ESTUDOS SOBRE A IMPLEMENTAÇÃO *ONLINE* PARA ESTIMAÇÃO DE ENERGIA NO TILECAL EM CENÁRIOS DE ALTA LUMINOSIDADE

Marcos Vinícius Teixeira*, Luciano M. de A. Filho*, Augusto Santiago Cerqueira*, José Manuel Seixas†, João Paulo B. da S. Duarte*

*UFJF – Universidade Federal de Juiz de Fora – PPEE
Campus Universitário, Plataforma 5 – Martelos – Juiz de Fora/MG – CEP:36036-030

Emails: teixeira.marcosv@gmail.com, luciano.ma.filho@gmail.com, augusto.s.cerqueira@gmail.com, seixas@lps.ufrj.br, joao.duarte@engenharia.ufjf.br

Abstract— This work aims at the study of techniques for online energy estimation in the ATLAS hadronic Calorimeter (TileCal) on the LHC collider. During further periods of the LHC operation, signals coming from adjacent collisions will be observed within the same window, producing a signal superposition. In this environment, the energy reconstruction method **COF** (Constrained Optimal Filter) outperforms the algorithm currently implemented in the system. However, the **COF** method requires an inversion of matrices and its online implementation is not feasible. To avoid such inversion of matrices, this work presents iteractive methods to implement the **COF**, resulting in simple mathematical operations. Based on the Gradient Descent, the results demonstrate that the algorithms are capable of estimating the amplitude of the superimposed signals with efficiency similar to **COF**.

Keywords— Energy Estimation, Best Linear Unbiased Estimator, Optimal Filter, Signal PileUp, Iterative optimization

Resumo— Este trabalho tem como objetivo o estudo de técnicas para a estimação da amplitude de sinais no calorímetro de telhas (TileCal) do ATLAS no LHC em cenários de alta luminosidade. Em alta luminosidade, sinais provenientes de colisões adjacentes são observados, ocasionando o efeito de empilhamento de sinais. Neste ambiente, o método **COF** (do inglês, *Constrained Optimal Filter*), apresenta desempenho superior ao algoritmo atualmente implementado no sistema. Entretanto, o **COF** requer a inversão de matrizes para o cálculo da pseudo-inversa de uma matriz de convolução, dificultando sua implementação *online*. Para evitar a inversão de matrizes, este trabalho apresenta métodos interativos, para adaptação do **COF**, que resultam em operações matemáticas simples. Baseados no Gradiente Descendente, os resultados demonstraram que os algoritmos são capazes de estimar a amplitude de sinais empilhados, além do sinal de interesse com eficiência similar ao **COF**.

Palavras-chave— Estimação de Energia, Melhor Estimador Linear Imparcial, Filtragem Ótima, Empilhamento de Sinais, Métodos Iterativos.

1 Introdução

O *Large Hadron Collider* (LHC) (L. Evans, 2008) é o maior colisionador de partículas do mundo e opera acelerando feixes de prótons em sentidos opostos, onde cada feixe é formado por milhares de pacotes de prótons. Os pacotes, então, colidem a uma taxa constante de 40 MHz no ponto de interesse dos detectores posicionados ao longo do acelerador.

O **ATLAS** (do inglês, *A Toroidal LHC ApparatuS*) (Collaboration, 2008) é o maior detector do LHC. É composto por diversos sub-detectores, onde cada um se dedica a medir propriedades específicas das partículas incidentes. Dentre estes destacam-se os calorímetros, responsáveis por absorver e amostrar, com precisão, a energia das partículas incidentes (Wigmans, 2000). O calorímetro hadrônico de Telhas do **ATLAS**, o TileCal (do inglês, Tile Calorimeter), ambiente de pesquisa deste trabalho, está passando por um processo de atualização devido ao aumento do número de interações por colisão (alta luminosidade) no **LHC**. Com este aumento, mais partículas atingirão os calorímetros do **ATLAS**, comprometendo o desempenho do algoritmo atualmente empregado no TileCal. Chamado de **OF** (do inglês, *Optimal Filter*) (Fullana, 2006), este algoritmo estima, de forma online, a energia em um determinado canal a partir da estimação da amplitude de um pulso proporcional a energia depositada.

O método é ótimo para ambientes onde o ruído de fundo pode ser modelado por uma distribuição gaussiana e o sinal não varia o seu formato e a sua fase. Porém, em cenários de alta luminosidade, sinais provenientes de colisões adjacentes serão acumulados em uma mesma janela de 150 ns, fazendo com que o **OF** deixe de ser ótimo, reduzindo assim seu desempenho. Portanto, um novo algoritmo para estimação deve ser desenvolvido.

Proposto pela colaboração brasileira, um algoritmo mais eficiente foi implementado *offline*. Denominado **COF** (do inglês, *Constrained Optimal Filter*) (L. Filho, 2012), este vem obtendo excelentes resultados no cenário de alta luminosidade, estimando, além da amplitude do pulso de interesse, a amplitude de cada componente sobreposta. No entanto, o **COF** apresenta elevada complexidade o que dificulta

sua implementação dentro dos requisitos de tempo de processamento exigidos nos níveis de filtragem de eventos. Tendo em vista a importância da estimação da energia para calorimetria e da relevância desta informação para o **ATLAS**, este trabalho tem como objetivo avaliar métodos de implementação do **COF** para processamento *online*.

Assim, neste trabalho, foram realizados estudos que visam implementar o **COF** para processamento *online*. O **COF** requer a inversão de matrizes para o cálculo da pseudo-inversa de uma matriz de convolução. Este processo permite deconvoluir os sinais sobrepostos, possibilitando recuperar a amplitude dos mesmos. A necessidade de realizar um procedimento de inversão de matrizes ou a possibilidade de utilização de bancos de filtros para cada combinação de sinais empilhados, apresenta elevado custo para implementação *online*. Nesse sentido, o objeto de contribuição deste trabalho é realizar esta inversão através da solução de um sistema de equações lineares, a ser resolvido de forma iterativa, sequencial, viabilizando a implementação para processamento *online* e, permitindo assim, a reconstrução de energia em ambiente de alta luminosidade.

2 Métodos de Reconstrução de Energia

A seguir, descreve-se a estimativa da amplitude de um modelo genérico.

2.1 BLUE para um vetor de parâmetros

O **BLUE** (do inglês, *Best Linear Unbiased Estimator*) (Kay, 1993) restringi-se a um estimador linear, de modo a determinar os pesos ótimos de um filtro que minimiza a variância da estimativa. Neste, as amostras do sinal de entrada digitalizado é dado por $\mathbf{r} = \mathbf{Ga} + \mathbf{w}$, onde $\mathbf{a}$ é o vetor coluna que contém as amplitudes dos sinais sobrepostos. Cada coluna da matriz $\mathbf{G}$ representa as amostras dos sinais de referência, cujas amplitudes estão contidas no vetor $\mathbf{a}$. A componente $\mathbf{w}$ trata-se do ruído aditivo WG (do inglês, *White-Gaussian*) com média zero. Sendo assim, a equação geral do método **BLUE** pode ser definida como o produto interno entre os pesos do filtro e as amostras do sinal de entrada ($\hat{\mathbf{a}} = \mathbf{H^T}\,\mathbf{r}$), onde $\hat{\mathbf{a}}$ é a estimativa resultante do vetor de amplitudes e $\mathbf{H}$ representa a matriz de coeficientes do banco de filtros lineares a ser determinado. Para que os pesos do **BLUE** sejam encontrados, é imposta a imparcialidade na estimativa de $\hat{\mathbf{a}}$. Assim, para um estimador linear não-tendencioso, temos que $E\{\hat{\mathbf{a}}\} = \mathbf{H^T}\,E(\mathbf{r}) = \mathbf{a}$.

Seja o vetor $\mathbf{w}$ um ruído aditivo com média zero, então o valor esperado da estimativa resultante do vetor de amplitudes pode ser reescrita como $E\{\hat{\mathbf{a}}\} = \mathbf{H^T\,Ga} = \mathbf{a}$. No método **BLUE**, seja N o número de sinais sobrepostos, então N restrições são inseridas no processo de minimização de cada componente. Logo, para que $E\{\hat{\mathbf{a}}\}$ seja igual a $\mathbf{a}$, o produto entre a matriz dos pesos do **BLUE** $\mathbf{H}$ e a matriz do pulso de referência $\mathbf{G}$ deve ser igual a matriz identidade $\mathbf{I}$, como segue $\mathbf{H^T\,G} = \mathbf{I}$.

Desta forma, o valor esperado da estimativa resultante do vetor de amplitudes $E\{\hat{\mathbf{a}}\}$ será igual ao próprio vetor de constantes $\mathbf{a}$ e, dado que o vetor $\mathbf{a}$ representa a amplitude real das amostras do sinal de entrada $\mathbf{r}$, a propriedade de imparcialidade apresentada é atendida. Uma vez determinada a restrição para pesos do **BLUE**, o objetivo passa a ser encontrar a variância dos estimadores, dada por $var\{\hat{\mathbf{a}}\} = \mathbf{H^T C\,H}$, onde $\mathbf{C}$ é a matriz de covariância do ruído.

A etapa seguinte consiste em aplicar o método dos multiplicadores de Lagrange (Bertsekas, 1996), de forma a minimizar a variância dado as restrições. Como N restrições são inseridas no processo de minimização de cada componente, então N^2 multiplicadores são utilizados no método. O método de Lagrange, formado por uma combinação linear, determina o ponto mínimo da variância da estimativa da amplitude condicionada à restrição imposta. Isto é feito derivando a função de Lagrange e igualando-a a zero. Em seguida, as etapas da formulação matemática consistem em isolar variáveis e substituí-las para se determinar os pesos do **BLUE** e, conseqüentemente, a estimativa resultante ótima do vetor de amplitudes. Este procedimento é descrito em (Kay, 1993). Logo, os pesos ótimos do **BLUE** para um vetor de parâmetros são encontrados através da Equação (1):

$$\mathbf{H_{otimo}} = \left(\mathbf{G^T\,C^{-1}G}\right)^{-1}\mathbf{C^{-1}G} \tag{1}$$

Substituindo a Equação (1) em $\hat{\mathbf{a}} = \mathbf{H^T\,r}$, a estimativa ótima da amplitude do sinal é determinada por meio da Equação (2):

$$\hat{\mathbf{a}} = \left(\mathbf{G^T\,C^{-1}G}\right)^{-1}\mathbf{G^T C^{-1}r} \tag{2}$$

2.2 OF com restrições adicionais - COF

Para o caso em que se deseja estimar apenas a informação relevante para o sistema de *trigger* do TileCal, ou seja, apenas o pulso central da janela, os pesos de um filtro linear seriam dados pela Equação (1), substituindo a matriz $\mathbf{G}$ pelo vetor $\mathbf{g}$ com as amostras do sinal de referência do TileCal. Portanto, na ausência de *pileup*, e assumindo que o ruído eletrônico é gaussiano branco (WG), a Equação (1) pode ser reescrita conforme a Equação (3)

$$\mathbf{h_{otimo}} = \frac{\mathbf{g}}{\|\mathbf{g}\|^{\mathbf{2}}} \tag{3}$$

Para o caso particular em que se observa *pileup* em algum BC (do inglês, *Bunch-Crossings*) e assumindo que o ruído eletrônico é WG, a Equação (1) pode ser simplificada, conforme é apresentado pela Equação (4):

$$\mathbf{H_{otimo}} = \left(\mathbf{G^T}\,\mathbf{G}\right)^{-1}\mathbf{G} \tag{4}$$

onde $\mathbf{G}$ contém os sinais de referência relacionados aos BC onde o *pile-up* foi observado. O processo de detecção de *pileup*, para os 7 BC centrais, também pode ser executado com a ajuda da Equação (2). No caso especial em que se deseja estimar todas as sete componentes sobrepostas, por ser uma matriz quadrada, podemos reescrever a Equação (4) reduzindo-a, como mostra a Equação (5). Desta forma, a inversa da matriz $\mathbf{G^T}$ é, portanto, chamada de matriz de deconvolução **(DM)**.

$$\mathbf{H_{otimo}} = \left(\mathbf{G^T}\right)^{-1} \tag{5}$$

O processo de deconvolução indica a existência de informações de *pile-up*. Estas informações são então comparadas com um limiar. Se as informações de *pile-up* forem maiores do que este limiar, então o **COF** será capaz de reconstruir a energia de cada componente detectada com o mínimo de restrições.

3 Propostas para Implementação *Online*

Para implementar o **COF** iterativamente, partimos da função custo para minimizar o erro médio quadrático entre o sinal recebido $\mathbf{r}$ e a projeção das amplitudes estimadas $\hat{\mathbf{a}}$ na matriz dada pelos sinais de referência $\mathbf{G}$.

3.1 Gradiente Descendente - GD

$$J\left(\hat{a}\right) \;=\; \|\mathbf{G}\hat{\mathbf{a}} \;-\; \mathbf{r}\|^{\mathbf{2}} \tag{6}$$

Atualizando-se os valores em $\hat{\mathbf{a}}$ na direção contrária ao gradiente de J, em pequenos passos, obtém-se os estimadores de amplitude $\hat{\mathbf{a}}$ que miniminizam o erro médio quadrático. Este algoritmo é conhecido como Gradiente Descendente (Luenberger, 2008). Assim, derivando J em relação a $\hat{\mathbf{a}}$, o processo iterativo na determinação de $\hat{\mathbf{a}}$ é implementado da seguinte forma,

$$\hat{\mathbf{a}}[n+1] = \hat{\mathbf{a}}[n] - \mu\mathbf{G}\left(\mathbf{G}\hat{\mathbf{a}}[n] - \mathbf{r}\right) \tag{7}$$

onde μ e a taxa de evolução do processo iterativo. O valor inicial de $\hat{\mathbf{a}}[\mathbf{0}]$ é obtido com o processo de deconvolução dado pela Equação (5).

3.2 Gradiente Descendente com Convergência Dinâmica - GDD

No método **GDD**, o processo iterativo na determinação de $\hat{\mathbf{a}}$ é implementado de forma análoga à Equação (7). Entretanto, no presente método, determina-se o valor de μ capaz de obter o menor valor possível de J na direção de propagação a cada iteração. Assim, partindo da Equação (7) e assumindo o vetor direção $\mathbf{d}$ equivalente à $\mathbf{G^T}\left(\mathbf{r} - \mathbf{G}\hat{\mathbf{a}}[k]\right)$, determina-se a equação geral do presente método,

$$\hat{\mathbf{a}}[k+1] = \hat{\mathbf{a}}[k] + \mu\,\mathbf{d} \tag{8}$$

Definida a Equação (8), os passos seguintes da formulação matemática consistem em determinar a taxa de convergência dinâmica μ. Para tal, inicialmente reescrevemos a Equação (6), conforme apresenta-se a Equação (9),

$$J\left(\hat{\mathbf{a}}\right) = \left(\mathbf{r} \;-\; \mathbf{G}\hat{\mathbf{a}}[k+1]\right)^{\mathbf{T}}\left(\mathbf{r} \;-\; \mathbf{G}\hat{\mathbf{a}}[k+1]\right) \tag{9}$$

onde $\hat{\mathbf{a}}[k+1]$ representa o vetor de amplitudes da próxima iteração na direção de propagação. Em seguida, substitui-se a Equação (8) em $\hat{\mathbf{a}}[k+1]$ fazendo com que o termo μ a ser determinado esteja presente na Equação (9).

Determinado J em termos de μ, os próximos passos consistem no processo de minimização em relação a μ. Logo, derivando, igualado a zero, e em seguida isolando os demais termos em função da taxa de convergência μ dinâmica, encontramos a Equação (10),

$$\mu = \frac{\mathbf{d^T d}}{\mathbf{d^T}\left(\mathbf{G^T G}\right)\mathbf{d}}. \tag{10}$$

Este procedimento também é apresentado em (Teixeira, 2014). Substituindo (10) em (8), encontramos a formulação para estimação de sinais, expressa por,

$$\hat{\mathbf{a}}[k+1] = \hat{\mathbf{a}}[k] + \frac{\mathbf{d^T d}}{\mathbf{d^T}\left(\mathbf{G^T G}\right)\mathbf{d}}\mathbf{d}[k]. \tag{11}$$

3.3 Gradiente Conjugado - GC

O método do Gradiente Conjugado é um método iterativo para resolver um sistema específico de equações lineares,

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}, \tag{12}$$

onde $\mathbf{A}$ é uma matriz quadrada p x p, simétrica e definida positiva, $\mathbf{x}$ e $\mathbf{b}$ são vetores coluna. Neste sentido, uma solução encontrada para resolver o sistema seria simplesmente considerar a matriz $\mathbf{A}$ inversa, ou seja, $\mathbf{x} = \mathbf{A^{-1}}\,\mathbf{b}$.

Entretanto, o cálculo da inversa retornaria ao problema apresentado pelo método **COF**, dado que sua implementação para processamento *online* não é otimizada. Uma outra solução é propor $\mathbf{x}$ como sendo uma combinação linear de uma base vetorial $\mathbf{p}_i$ pré-determinada, como mostra a Equação (13), e encontrar uma solução para determinar α_i,

$$\mathbf{x} = \sum_{i=1}^{p} \alpha_i \ \mathbf{p}_i. \tag{13}$$

Neste caso, substituindo a Equação (13) na Equação (12), encontramos a Equação (14) para $\mathbf{A}$ simétrica e definida positiva,

$$\mathbf{b} = \mathbf{A}\mathbf{x} = \sum_{i=1}^{p} \alpha_i \ \mathbf{A} \ \mathbf{p}_i. \tag{14}$$

Multiplicando ambos os lados da Equação (12) por $\mathbf{p}_k$, então podemos reescrever a Equação (14), conforme apresenta a Equação (15),

$$\mathbf{p_k^T b} \ = \ \mathbf{p_k^T} \sum_{i=1}^{p} \alpha_i \ \mathbf{A} \ \mathbf{p}_i. \tag{15}$$

Novamente, sabendo que $\mathbf{A}$ é uma matriz simétrica e definida positiva, então podemos reescrever a Equação (15), conforme apresenta a Equação (16),

$$\mathbf{p_k^T b} \ = \ \sum_{i=1}^{p} \alpha_i \ \mathbf{p_k^T} \ \mathbf{A} \ \mathbf{p}_i. \tag{16}$$

A Equação (16) pode ser extremamente simplificada se nos restringirmos a uma base tal que $\mathbf{p_k^T A p}_i = 0$, para $k \neq i$. Logo, através desse produto, os vetores $\mathbf{p}_k$ e $\mathbf{p}_i$ dizem-se ser A-ortogonais ou conjugados. Assim, podemos reescrever a Equação (16) conforme a Equação (17),

$$\mathbf{p_k^T b} \ = \ \alpha_k \ \mathbf{p_k^T} \ \mathbf{A} \ \mathbf{p}_k. \tag{17}$$

Finalmente, para que o ponderador α_k do sistema linear seja encontrado a partir da propriedade imposta, basta resolver a Equação (17) isolando-o, como mostra a Equação (18). Uma outra vantagem em restringir as componentes da base a serem A-ortogonais ($\mathbf{p_k^T A p}_i = 0$) é que os mesmos podem ser encontrados iterativamente usando de decomposição de Gram-Schmidt, como será visto adiante. Este fato reflete uma estreita relação desse método com o **GDD**.

$$\alpha_k = \frac{\mathbf{p_k^T b}}{\mathbf{p_k^T} \ \mathbf{A} \ \mathbf{p}_k} \tag{18}$$

3.3.1 Adaptação do GC para o COF

Em nosso caso particular, o sistema linear a ser otimizado é dado pela Equação (19),

$$\mathbf{G\,a} = \mathbf{r}, \tag{19}$$

onde $\mathbf{G}$ contém os sinais de referência relacionados aos BCs onde o empilhamento de sinais foram observados. Neste caso, a matriz $\mathbf{G}$ não é quadrada nem simétrica já que o empilhamento de sinais ocorre aleatoriamente. Porém, multiplicando à direita ambos os lados da Equação (19) por $\mathbf{G^T}$, como mostra a Equação (20),

$$\mathbf{G^T G\,a} = \mathbf{G^T\,r}, \tag{20}$$

e relacionando a Equação (20) com a Equação (12), temos que $\mathbf{A} = \mathbf{G^T\,G}$ e $\mathbf{b} = \mathbf{G^T\,r}$, de modo que a Equação (18) pode ser reescrita para este caso em particular, como mostra a Equação (21),

$$\alpha_k = \frac{\mathbf{p_k^T G^T\,r}}{\mathbf{p_k^T\ G^T\,G\ p}_k}. \tag{21}$$

Assim, de acordo com a Equação (21), para usar o método **GC** no mesmo caso, devemos primeiro projetar o vetor de entrada $\mathbf{r}$ na matriz $\mathbf{G}$ e considerar $\mathbf{A} = \mathbf{G^T\,G}$. Desta forma, conclui-se que os sistemas de equações lineares, (12) e (19), podem ser resolvidos através da Equação (13). Neste caso, reescrevendo a Equação (13), temos

$$\mathbf{x} = \alpha_1\,\mathbf{p}_1 \ + \ \alpha_2\,\mathbf{p}_2 \ + \ ... \ + \alpha_k \ \mathbf{p}_p, \tag{22}$$

onde p é o número de sinais empilhados.

Pode-se traçar um paralelo entre esta equação e a Equação (8) do método **GD**. Para tal, nota-se que o resultado obtido com a Equação (8), após k iterações, é um valor muito próximo de $\hat{\mathbf{a}}$ dado pelo método **COF**, e pode ser escrito, por extenso, como,

$$\hat{\mathbf{a}} \cong \hat{\mathbf{a}}[0] \ + \ \mu_1\mathbf{d}_1 \ + \ \mu_2\mathbf{d}_2 \ + \ ... \ + \mu_k\mathbf{d}_k, \tag{23}$$

onde $\hat{\mathbf{a}}[0]$ é o valor inicial obtido pelo **DM** (5). Porém inicializando-se com um vetor nulo $\hat{\mathbf{a}}[0] = 0$ e comparando-se a Equação (22) e (23) termo a termo, chegamos as seguintes conclusões: i) Se forçarmos o resíduo $\mathbf{d}$ a ser A-ortogonal a cada iteração, ou seja, $\mathbf{d_k^T\,A\,d_{k+1}^T} = 0$, o método convergirá perfeitamente para o **COF**, após p iterações, onde p é o número de sinais de referência contribuindo com o empilhamento de sinais (número de colunas de $\mathbf{G}$). ii) Nesse caso, os p fatores de convergência dinâmica μ_k são dados

diretamente pela Equação (21). iii) Não se faz necessário utilizar o valor do **DM** como ponto de partida. iv) O processo de se obter direções A-ortogonais a cada iteração é computado com a decomposição de Gran-Smidth.

Uma vez que se provou ser possível encontrar a solução em p iterações, minimizando em cada passo ao longo das direções, só é necessário encontrar essas direções. Para tal, nessa primeira etapa, é possível gerar vetores conjugados a partir de vetores $\mathbf{v}_k$ linearmente independentes através do processo de conjugação de Gram-Schmidt. Assim, fazendo $\mathbf{p}_1$ igual a $\mathbf{d}_1$, podemos calcular as demais direções conjugadas $\mathbf{p}_k$ através da combinação linear representada pela Equação (24),

$$\mathbf{p}_k = \mathbf{d}_k \; + \; \sum_{j=0}^{k\text{-}1} \beta_{k,j} \; \mathbf{p}_j \;\; para \; k > 2, \qquad (24)$$

onde $\beta_{k,j}$ é um escalar a ser determinado. Tal escalar tem a função de ajustar a direção de $\mathbf{d}_k$ de modo a torná-lo conjugado às direções anteriores. Para isso, transpomos a Equação (24) e multiplicamos à direita por $\mathbf{A}\mathbf{p}_i$, em ambos os lados, para encontramos a Equação (25),

$$\mathbf{p}_k^{\mathbf{T}}\mathbf{A}\mathbf{p}_i \;\; = \;\; \mathbf{d}_k^{\mathbf{T}}\mathbf{A}\mathbf{p}_i \; + \; \sum_{j=0}^{k\text{-}1} \beta_{k,j} \; \mathbf{p}_j^{\mathbf{T}}\mathbf{A}\mathbf{p}_i. \qquad (25)$$

Impondo a condição $\mathbf{p}_k^{\mathbf{T}}\mathbf{A}\mathbf{p}_i = 0$, para $i = k+1$, podemos reescrever a Equação (25) como segue,

$$0 \;\; = \;\; \mathbf{d}_k^{\mathbf{T}}\mathbf{A}\mathbf{p}_j \; + \; \beta_{k,j} \; \mathbf{p}_j^{\mathbf{T}}\mathbf{A}\mathbf{p}_j. \qquad (26)$$

Finalmente, para que $\beta_{k,j}$ seja determinado, basta isolarmos o termo na Equação (26). Desta forma, encontramos a Equação (27),

$$\beta_{k,j} \;\; = \;\; - \frac{\mathbf{d}_k^{\mathbf{T}}\mathbf{A}\mathbf{p}_j}{\mathbf{p}_j^{\mathbf{T}}\mathbf{A}\mathbf{p}_j}. \qquad (27)$$

Até esta etapa, podemos sumarizar o processo de adaptação do algoritmo **GC** para a determinação das amplitudes dadas pelo **COF**, como; i) Inicializando com todas as amplitudes igual a zero, encontramos a direção da primeira propagação $\mathbf{d}_1$ e seu passo μ_1 de acordo com a a Equação (10). ii) A direção da próxima iteração $\mathbf{d}_k$ é calculada normalmente como no algoritmo **GD**, porém um ajuste fino nesta direção, usando a Equação (24) é necessário para garantir que a direção atual seja conjugada às direções anteriores em relação a $\mathbf{G}^{\mathbf{T}}\mathbf{G}$. iii) Após p iterações, obtém-se o valor exato dado pelo **COF**.

O cálculo de $\beta_{k,j}$ necessita do armazenamento dos vetores $\mathbf{d}_k$ das direções anteriores o que gera operações adicionais a serem executadas pelo presente método. No entanto, ao se gerar um conjunto de vetores conjugados, pode-se calcular um novo vetor $\mathbf{p}_k$ usando apenas o vetor anterior $\mathbf{p}_{k-1}$, o que exige pouco armazenamento e cálculo. Assim, a computação dos vetores conjugados dado pela Equação (24) pode ser simplificada usando algumas manipulações matemáticas. Tais simplificações são detalhadas em (Wigmans, 2000), e comprovam que a forma mais eficiente para o cálculo de β é dado por,

$$\beta_k \;\; = \;\; \frac{\mathbf{e}_k^{\mathbf{T}}\mathbf{A}\mathbf{p}_{k-1}}{\mathbf{p}_{k-1}^{\mathbf{T}}\mathbf{A}\mathbf{p}_{k-1}}. \qquad (28)$$

assim como a Equação (18), pode ser reescrita conforme a Equação (29) para o cálculo de α

$$\mu_k \;\; = \;\; \frac{\mathbf{e}_k^{\mathbf{T}}\mathbf{e}_k}{\mathbf{p}_k^{\mathbf{T}}\mathbf{G}^{\mathbf{T}}\mathbf{G}\mathbf{p}_k} \qquad (29)$$

4 Simulação dos Métodos Propostos

Já conhecido a eficiência do método *offline* e de forma compará-lo com os métodos propostos, aqui, busca-se encontrar o menor desvio do erro, de forma à concluir o quão longe a estimativa dos métodos iterativos estarão da estimativa **COF**, em porcentagem. Para tal, inicialmente foi gerado um vetor linha com aproximadamente 100.000 amostras, onde cada uma das amostras é equivalente a energia absorvida para uma determinada célula do calorímetro, após o evento de colisão. Em seguida, são sobrepostos sinais de referência do TileCal com uma distribuição de amplitude dada por uma exponencial e com o valor médio igual a 30 contagens de ADC para simular a energia depositada. A sequência de colisões é então dividida em janelas de 7 amostras, gerando vetores de entrada para a simulação. Esta sequência é preenchida aleatóriamente para um determinado fator de ocupância, que representa a porcentagem (10% e 20%) de colisões que realmente atingiram uma determinada célula do calorímetro. Após determinado os vetores de entrada um ruído WG de 1 contagem de ADC é adicionado, representando o ruído eletrônico.

A Figura (1) demonstra o resultado da simulação para o método **GD** para uma ocupância de 20%. Nota-se que são necessárias no mínimo 20 iterações, para se obter desvio inferior a 1% entre o **GD** e o **COF**, considerando μ igual a 0.4. Logo, a partir dos valores determinados em simulação para a taxa de convergência (μ) e número de interações, o método **GD** e capaz de determinar a estimativa da amplitude de sinais sobrepostos com precisão, se comparado com o **COF**.

Através da Figura (2), é possível concluir que o **GDD** necessita, em média, de 9 e 13 iterações,

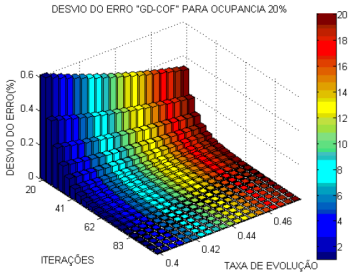

Figura 1: Desvio do erro entre o **GD** e o **COF**.

no mínimo, para um desvio inferior a 1% do **COF**, com ocupância de 10% e 20%, respectivamente. Este resultado comprova uma eficiência superior do **GDD** quando comparado ao **GD**, já que é capaz de determinar a estimativa da amplitude de sinais sobrepostos com precisão similar ao **COF** com um menor número de interações.

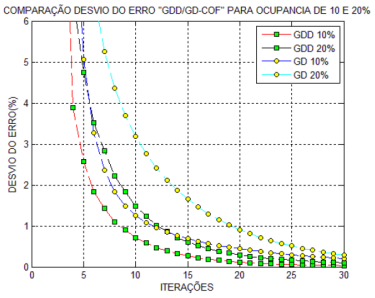

Figura 2: Comparação **GD**, **GDD** e **COF**.

Diferentemente do **GD** e **GDD**, através da Figura (3), é possível concluir que o **GC** é capaz de convergir para um desvio nulo com um total de 7 iterações, no máximo, como era esperado, já que o algoritmo irá convergir para o **COF** com um número finito de iterações. Em nosso caso particular, este número finito tem seu valor máximo dado pela quantidade máxima de sinais sobrepostos detectados dentro da janela de 150ns.

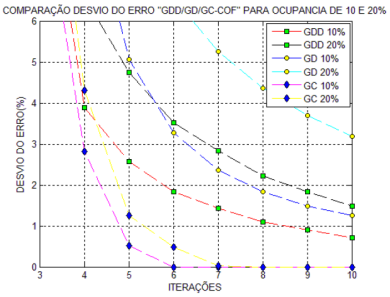

Figura 3: Comparação **GD**, **GDD**, **GC** e **COF**.

5 Conclusões e Trabalhos Futuros

Através deste trabalho foi possível comprovar matematicamente que o algoritmo **GD** evita a inversão de matrizes resultando em simples operações de soma e produto. No entanto, o **GD** necessita de um número elevado de iterações para convergência. Já o **GDD**, localiza o ponto de mínimo com menor número iterações se comparado com o **GD**, haja visto que a taxa de evolução é calculada adaptativamente. Este método resulta em simples operações de produto, soma e divisão entre escalares. Por fim, comprovou-se que **GC** executa poucas iterações sendo capaz de encontrar a amplitude de sinais com elevada precisão, superando os demais métodos. Este, converge para um número finito de iterações (p iterações), que correspondem exatamente a quantidades de sinais empilhados dentro de uma mesma janela. Também, resulta em simples operações de produto, soma e divisão entre escalares evitando a inversão de matrizes do **COF** e viabilizando sua implementação em dispositivos FPGA para processamento *on-line*.

Referências

Bertsekas, D. P. (1996). *Constrained Optimization and Lagrange Multiplier Methods*, 1st edn, Athena Scientific.

Collaboration, T. A. (2008). The atlas experiment at the cern large hadron collider, *Journal of Instrumentation* **3**(08003).

Fullana, E. (2006). Optimal filtering in the atlas hadronic tile calorimeter, *IEEE Transactions On Nuclear Science* **53**(4).

Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing – Estimation Theory*, 1st edn, Prentice Hall.

L. Evans, P. B. (2008). Lhc machine, *Journal of Instrumentation, JINST 3 S08001* .

L. Filho, A. Cerqueira, D. D. J. S. (2012). Calorimeter signal response deconvolution for online energy estimation in presence of pile-up, *Atl-tilecal-int-2012-008*, CERN Document Serve.

Luenberger, D. G., Y. (2008). *Linear and Nonlinear Programming*, 3rd edn, Reading, MA, Addison-Wesley.

Teixeira, M. V.; A. Filho, L. M. . P. B. S. (2014). Reconstrução online para calorímetros operando em condições de altas luminosidades, *Xx cba, 2014*, XX Congresso Brasileiro de Automática.

Wigmans, R. (2000). *Calorimetry: Energy Measurement in Particle Physics*, Oxford University Press.