

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

Tiago Aparecido Teixeira

**Implementação de métodos esparsos para deconvolução on-line de sinais no calorímetro
hadrônico do ATLAS**

Juiz de Fora
2022

Tiago Aparecido Teixeira

**Implementação de métodos esparsos para deconvolução on-line de sinais no calorímetro
hadrônico do ATLAS**

Tese apresentada ao Programa de Pós-Graduação
em Engenharia Elétrica da Universidade Federal
de Juiz de Fora como requisito parcial à obtenção
do título de Doutor em Engenharia Elétrica. Área
de concentração: Sistemas Eletrônicos

Orientador: Prof. Dr. Luciano Manhães de Andrade Filho

Juiz de Fora
2022

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF com
os dados fornecidos pelo(a) autor(a)

Teixeira, Tiago Aparecido.

Implementação de métodos esparsos para deconvolução on-line de sinais no calorímetro hadrônico do ATLAS / Tiago Aparecido Teixeira. – 2022.

231 f. : il.

Orientador: Luciano Manhães de Andrade Filho

Tese (Doutorado) – Universidade Federal de Juiz de Fora, Faculdade de Engenharia Elétrica. Programa de Pós-Graduação em Engenharia Elétrica, 2022.

1. Deconvolução. 2. Calorimetria. 3. Reconstrução de Energia. 4. Processamento online. 5. FPGA. I. de Andrade Filho, Luciano Manhães, orient. II. Título.

Tiago Aparecido Teixeira

Implementação de métodos esparsos para deconvolução on-line de sinais no calorímetro hadrônico do ATLAS

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Sistemas Eletrônicos

Aprovada em 12 de Novembro de 2021

BANCA EXAMINADORA

Luciano M. de A. Filho

Prof. Dr. Luciano Manhães de Andrade Filho - Orientador
Universidade Federal de Juiz de Fora

Carlos Augusto Duque

Professor Dr. Carlos Augusto Duque
Universidade Federal de Juiz de Fora

Álvaro A. M. de Medeiros

Professor Dr. Álvaro Augusto Machado de Medeiros
Universidade Federal de Juiz de Fora

Farias

Professor Dr. Paulo César Machado de Abreu Farias
Universidade Federal da Bahia

Bernardo Sotto-Maior Peralva

Professor Dr. Bernardo Sotto-Maior Peralva
Universidade do Estado do Rio de Janeiro

À minha família e
aos meus amigos.

AGRADECIMENTOS

Este trabalho só se tornou possível devido ao grande orientador que me aceitou nesta empreitada. Fico sem palavras para descrever o quanto foi importante tudo o que ele se dispôs a fazer por mim todo o tempo. Brinco até que não posso divulgar muito para não causar inveja a meus pares. Sô Luciano Manhães, este título é seu, meu amigo.

Agradeço muito a meus pais, por terem me dado todas as bases para que eu pudesse chegar até aqui e por suportarem todos os meus momentos de ausência, necessários para que este trabalho pudesse ficar pronto.

Ao meu grande amigo, Jacimar Tavares, por ficar me enchendo até eu criar coragem para esta empreitada.

O caminhar na realização deste meu sonho se tornou mais leve e agradável quando eu conheci o amor da minha vida: Luiza Leonel você trouxe luz e paz ao meu coração e esteve comigo nos momentos mais difíceis da reta final, o que tornou a chegada mais prazerosa e significativa. Muito obrigado por tudo, meu AMOR.

Tenho muito o que agradecer aos colegas do doutorado: ao João Paulo Duarte, à Elaine, enfim, a todos que de alguma forma sempre estiveram dispostos a me ajudar, seja com alguma explicação de alguma fórmula, seja com uma ida ao Cultural.

RESUMO

O acelerador de partículas, conhecido como *large hadron collider* entrou em operação em 2008 e, desde então, vem passando por um processo de constantes atualizações, em que parâmetros como a energia das colisões e a quantidade de partículas por *bunch* são aumentadas. Esses aumentos vêm afetando todos os detectores, dentre eles, o detector do experimento ATLAS. Nesse experimento, o sistema de *trigger* é onde os efeitos podem ser sentidos de forma mais contundente, pois há um aumento de ocorrência de colisões adjacentes, produzindo o efeito conhecido como *pile-up* (sobreposição de sinais), que dificulta o processo de seleção de eventos de interesse. Por isso, métodos de deconvolução *on-line* estão sendo propostos para lidar com esse problema. A maioria desses métodos são baseados em filtros *finite impulse response*, por já possuírem uma estrutura que permite a implementação *on-line*. Em contrapartida, o *large hadron collider* opera com trens de *bunches*, onde são intercaladas uma sequência temporal com colisões e outra sequência temporal sem colisões. Tal característica pode ser interpretada como janelas de aquisição de tamanho fixo, que permitem o uso de métodos iterativos (não causais) de deconvolução. Tais métodos tendem a apresentar resultados melhores que os baseados em filtros *finite impulse response* usados atualmente, por ser possível o uso de conhecimento especialista e algoritmos não causais. Para isso, neste trabalho, são propostos métodos iterativos de deconvolução baseados em teorias modernas de representação esparsa de dados e o seu desempenho é comparado com o desempenho dos métodos baseados em filtros *finite impulse response*. Dentre os métodos analisados destaca-se o *separable surrogate functionals*, com proposta de sua implementação em *field programmable gate array*, cujos resultados apontam a possibilidade da execução de várias instâncias desse método em paralelo, em uma única *field programmable gate array*, permitindo então o processamento *on-line* requerido no primeiro nível de *trigger* do ATLAS.

Palavras-chave: Calorimetria. Deconvolução. *Field Programmable Gate Array*. Processamento *online*. Reconstrução de Energia.

ABSTRACT

The particle accelerator, known as the large hadron collider went into operation in 2008 and, since then, has been undergoing a process of constant updating, where parameters such as collision energy and the amount of particles per bunch are increased. These increasing have affected all detectors, including the detectors in the ATLAS experiment. In this experiment, the trigger system is where the effects can be felt most forcefully, because there is an increasing in the occurrence of adjacent collisions, producing the effect known as pile-up (signals overlapping), that make the process difficult to selecting events of interest. Therefore, deconvolution on-line methods are being proposed to deal with this problem. Most of these methods are based on finite impulse response filter, because they already have a structure that allows the on-line implementation. In contrast, large hadron collider operates with train of bunches, where a time sequence with collision and another collision-less time sequence are interleaved. This feature can be interpreted as a fixed size acquisition windows, allowing the usage of iterative deconvolution methods. Such methods tend to present better results than the methods currently in use, based on finite impulse response filter, because it is possible to use calorimetry specific knowledge and non-causal algorithms. For this purpose, this work proposes iterative methods of deconvolution, based on modern theories of sparse data representation, and their performance is compared with the performance of methods based on finite impulse response filter. Among the analyzed methods, the Separable Surrogate Functionals stands out, being thus proposed a form of implementation of this method in field programmable gate array, whose results indicate the possibility of implementation of several instances of this method in parallel, in a single field programmable gate array, allowing then the on-line processing required in the first level trigger of ATLAS.

Keywords: Calorimetry. Deconvolution. Field Programmable Gate Array. Online Processing. Energy Reconstruction.

LISTA DE ILUSTRAÇÕES

Quadro 1 - Ordem de grandeza da energia mínima necessária e as ferramentas utilizadas para explorar distâncias Δx	28
Quadro 2 - Características dos aceleradores mais recentes	31
Quadro 3 - Resumo da evolução do ATLAS e do <i>Large Hadron Collider</i>	43
Quadro 4 - Configuração recomendada para cada uma das variações do método SSF . .	95
Figura 1 - Infográfico contendo as partículas fundamentais e forças previstas no Modelo Padrão.	139
Figura 2 - Técnicas para construção de aceleradores lineares	140
Figura 3 - Vista plana de um cyclotron clássico	140
Figura 4 - Esquema de um protonsynchrotron	141
Figura 5 - Diagrama esquemático de um síncrotron com foco em gradiente alternado . .	141
Figura 6 - Localização do Conseil Européen pour la Recherche Nucléaire na fronteira franco-suíça	142
Fotografia 1 - Large Hadron Collider no Conseil Européen pour la Recherche Nucléaire	142
Figura 7 - O complexo do LHC com todos os seus detectores e sub-dectores	143
Figura 8 - Plano de corte do Large Hadron Collider	143
Figura 9 - Cronologia das atualizações do Large Hadron Collider	144
Figura 10 - Trem de bunch do <i>Large Hadron Collider</i> com intervalos para o tempo de subida do kicker e intervalo de abortar	145
Figura 11 - Esquema de enchimento de 2556 <i>bunches</i> usado para o feixe 1 do LHC em 2017 para a operação física de prótons	145
Figura 12 - Encontro de feixes de longo alcance na operação do <i>Large Hadron Collider</i> , 2017	146
Gráfico 1 - Padrão de Colisão run 281411	146
Gráfico 2 - Padrão de Colisão run 283780	147
Gráfico 3 - Padrão de Colisão run 300908	147
Gráfico 4 - Padrão de Colisão run 328263	148
Gráfico 5 - Padrão de Colisão run 335177	148
Gráfico 6 - Padrão de Colisão run 349977	149
Figura 13 - Diagrama esquemático do sistema magnético do ATLAS	149
Figura 14 - Vista detalhada do detector ATLAS	150
Figura 15 - Sistemas de coordenadas cartesianas do ATLAS	151
Figura 16 - <i>Inner Detector</i>	152
Figura 17 - ATLAS subsistema de múons	152
Figura 18 - Calorímetro de argônio líquido.	153
Figura 19 - Calorímetro de Telhas envolvendo o calorímetro de argônio líquido	153
Figura 20 - Os quatro momentos de medições, para um grande range de energias de partículas	154

Figura 21 - Esquema da disposição das células num plano paralelo ao eixo do feixe, mostrando apenas o lado positivo η (o detector é simétrico em relação a $\eta = 0$)	154
Figura 22 - O barril central e as duas tampas.	155
Figura 23 - Desenho de um módulo do TileCal	155
Fotografia 2 - Módulos dos barris estendidos sendo instalados no Conseil Européen pour la Recherche Nucléaire	156
Gráfico 7 - Formato de um pulso característico do TileCal	156
Diagrama 1 - Diagrama funcional do sistema de seleção de eventos e aquisição de dados do ATLAS	157
Diagrama 2 - Componentes do sistema de <i>trigger</i> Level 1	157
Diagrama 3 - Sistema de <i>trigger</i> L1 do TileCal	158
Diagrama 4 - Cadeia de sinais do TileCal	158
Diagrama 5 - Esboço da eletrônica atual de leitura de <i>front-end</i> do TileCal	159
Figura 24 - Diagramas das três opções de Front-End Board	159
Diagrama 6 - Esboço da eletrônica de leitura do TileCal após o <i>upgrade</i> da Fase II	159
Figura 25 - Cobertura dos ângulos $\eta \times \phi$ dos módulos ROD e TilePPR.	160
Diagrama 7 - Blocos de um projeto TilePPR candidato final	160
Gráfico 8 - Simulação da ocorrência de um <i>pile-up</i> registrado pelo TileCal	161
Diagrama 8 - Forma de um Filtro FIR utilizando <i>Least Squares</i>	161
Gráfico 9 - Performance de diferentes estratégias de filtros usados no FCal1 em função de energia transversa	162
Figura 26 - TileCal modelado como sistema LTI no tempo discreto.	163
Diagrama 9 - Esquema representativo da modelagem inversa	163
Gráfico 10 - Expressão gráfica da função de encolhimento e a versão modificada proposta para promover a reconstrução positiva da energia	164
Diagrama 10 - Estrutura do SSF modificado	164
Diagrama 11 - Fluxo de dados em Separable surrogate functionals e Parallel-coordinate-descent	165
Gráfico 11 - Exemplo de mudanças de pedestal para luminosidades uniformes e realistas em diferentes regiões do calorímetro	166
Gráfico 12 - Relação da ordem do Filtro FDIP e o valor de RMSr para cada taxa de ocupação	167
Gráfico 13 - Valores de RMSr obtidos com a variação dos valores de v para o método SSF, de acordo com as taxas de ocupação	167
Diagrama 12 - Estrutura do SSFi modificado	168
Gráfico 14 - Curvas ROC dos métodos Filtro Casado, Filtro FDIP, DS e GD, para taxa de ocupação de 1%	169
Gráfico 15 - Curvas ROC dos métodos Filtro Casado, Filtro FDIP, DS e GD, para taxas de ocupação de 5% e 10%	170

Gráfico 16 - Curvas ROC dos métodos Filtro Casado, Filtro FDIP, DS e GD, para taxas de ocupação de 20% e 30%	171
Gráfico 17 - Curvas ROC dos métodos Filtro Casado, Filtro FDIP, DS e GD, para taxas de ocupação de 40% e 50%	172
Gráfico 18 - Curvas ROC dos métodos Filtro Casado, Filtro FDIP, DS e GD, para taxas de ocupação de 60% e 90%	173
Gráfico 19 - Variação do RMSr da aplicação dos métodos Filtro Casado, Filtro FDIP, DS e GD, de acordo com as taxas de ocupação	174
Gráfico 20 - Curvas ROC dos métodos da família <i>Greedy</i> , para taxa de ocupação de 1% .	175
Gráfico 21 - Curvas ROC dos métodos da família <i>Greedy</i> , para taxas de ocupação de 5% e 10%	176
Gráfico 22 - Curvas ROC dos métodos da família <i>Greedy</i> , para taxas de ocupação de 20% e 30%	177
Gráfico 23 - Curvas ROC dos métodos da família <i>Greedy</i> , para taxas de ocupação de 40% e 50%	178
Gráfico 24 - Curvas ROC dos métodos da família <i>Greedy</i> , para taxas de ocupação de 60% e 90%	179
Gráfico 25 - Variação do RMSr obtido da aplicação dos métodos <i>Greedy</i> , de acordo com as taxas de ocupação	180
Gráfico 26 - Valores de RMSr, obtidos com a variação dos valores de λ para o método SSF, com taxas de ocupação de 1% e 5%	181
Gráfico 27 - Valores de RMSr, obtidos com a variação dos valores de λ para o método SSF, com taxas de ocupação de 10% e 20%	182
Gráfico 28 - Valores de RMSr, obtidos com a variação dos valores de λ para o método SSF, com taxas de ocupação de 30% e 40%	183
Gráfico 29 - Valores de RMSr, obtidos com a variação dos valores de λ para o método SSF, com taxas de ocupação de 50% e 60%	184
Gráfico 30 - Valores de RMSr, obtidos com a variação dos valores de λ para o método SSF, com taxa de ocupação de 90%	185
Gráfico 31 - Valores médios de RMSr obtidos com a variação das opções de λ para o método SSF, de acordo com as taxas de ocupação	185
Gráfico 32 - Valores de μ para o método SSF, com taxas de ocupação de 1% e 5%	186
Gráfico 33 - Valores de μ para o método SSF, com taxas de ocupação de 10% e 20%	187
Gráfico 34 - Valores de μ para o método SSF, com taxas de ocupação de 30% e 40%	188
Gráfico 35 - Valores de μ para o método SSF, com taxas de ocupação de 50% e 60%	189
Gráfico 36 - Valores de μ para o método SSF, com taxa de ocupação de 90%	190
Gráfico 37 - Valores médios de μ obtidos para o método SSF, de acordo com as taxas de ocupação	190

Gráfico 38 - Valores de RMSr obtidos com as variações de μ , para o método SSF, com taxa de ocupação de 1%	191
Gráfico 39 - Valores de RMSr obtidos com as variações de μ , para o método SSF, com taxas de ocupação de 5% e 10%	192
Gráfico 40 - Valores de RMSr obtidos com as variações de μ , para o método SSF, com taxas de ocupação de 20% e 30%	193
Gráfico 41 - Valores de RMSr obtidos com as variações de μ , para o método SSF, com taxas de ocupação de 40% e 50%	194
Gráfico 42 - Valores de RMSr obtidos com as variações de μ , para o método SSF, com taxas de ocupação de 60% e 90%	195
Gráfico 43 - Valores de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o método SSFi, com taxas de ocupação de 1% e 5%	196
Gráfico 44 - Valores de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o método SSFi, com taxas de ocupação de 10% e 20%	197
Gráfico 45 - Valores de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o método SSFi, com taxas de ocupação de 30% e 40%	198
Gráfico 46 - Valores de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o método SSFi, com taxas de ocupação de 50% e 60%	199
Gráfico 47 - Valores de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o método SSFi, com taxa de ocupação de 90%.	200
Gráfico 48 - Valores médios de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o variação SSFi, de acordo com as taxas de ocupação	200
Gráfico 49 - Curvas ROC dos métodos da família <i>Shrinkage</i> , sem inicialização prévia, para taxa de ocupação de 1%	201
Gráfico 50 - Curvas ROC dos métodos da família <i>Shrinkage</i> , sem inicialização prévia, para taxas de ocupação de 5% e 10%	202
Gráfico 51 - Curvas ROC dos métodos da família <i>Shrinkage</i> , sem inicialização prévia, para taxas de ocupação de 20% e 30%	203
Gráfico 52 - Curvas ROC dos métodos da família <i>Shrinkage</i> , sem inicialização prévia, para taxas de ocupação de 40% e 50%	204
Gráfico 53 - Curvas ROC dos métodos da família <i>Shrinkage</i> , sem inicialização prévia, para taxas de ocupação de 60% e 90%	205
Gráfico 54 - Variação do RMSr da aplicação dos métodos <i>Shrinkage</i> , sem inicialização prévia, de acordo com as taxas de ocupação	206
Gráfico 55 - Curvas ROC dos métodos da família <i>Shrinkage</i> , com inicialização prévia, para taxa de ocupação de 1%	207
Gráfico 56 - Curvas ROC dos métodos da família <i>Shrinkage</i> , com inicialização prévia, para taxas de ocupação de 5% e 10%	208

Gráfico 57 - Curvas ROC dos métodos da família <i>Shrinkage</i> , com inicialização prévia, para taxas de ocupação de 20% e 30%	209
Gráfico 58 - Curvas ROC dos métodos da família <i>Shrinkage</i> , com inicialização prévia, para taxas de ocupação de 40% e 50%	210
Gráfico 59 - Curvas ROC dos métodos da família <i>Shrinkage</i> , com inicialização prévia, para taxas de ocupação de 60% e 90%	211
Gráfico 60 - Variação do RMSr da aplicação dos métodos <i>Shrinkage</i> , com inicialização prévia, de acordo com as taxas de ocupação	212
Gráfico 61 - Curvas ROC dos métodos que apresentaram os menores resultados de RMS da diferença do erro, dentre os de mesma classificação, para taxa de ocupação de 1% .	213
Gráfico 62 - Curvas ROC dos métodos que apresentaram os menores resultados de RMS da diferença do erro, dentre os de mesma classificação, para taxas de ocupação de 5% e 10%	214
Gráfico 63 - Curvas ROC dos métodos que apresentaram os menores resultados de RMS da diferença do erro, dentre os de mesma classificação, para taxas de ocupação de 20% e 30%	215
Gráfico 64 - Curvas ROC dos métodos que apresentaram os menores resultados de RMS da diferença do erro, dentre os de mesma classificação, para taxas de ocupação de 40% e 50%	216
Gráfico 65 - Curvas ROC dos métodos que apresentaram os menores resultados de RMS da diferença do erro, dentre os de mesma classificação, para taxas de ocupação de 60% e 90%.	217
Gráfico 66 - Agrupamento das variações do método SSF que apresentaram os menores resultados de RMS da diferença do erro, para taxa de ocupação de 1%	218
Gráfico 67 - Agrupamento das variações do método SSF que apresentaram os menores resultados de RMS da diferença do erro, para taxas de ocupação de 5% e 10%	219
Gráfico 68 - Agrupamento das variações do método SSF que apresentaram os menores resultados de RMS da diferença do erro, para taxas de ocupação de 20% e 30%	220
Gráfico 69 - Agrupamento das variações do método SSF que apresentaram os menores resultados de RMS da diferença do erro, para taxas de ocupação de 40% e 50%	221
Gráfico 70 - Agrupamento das variações do método SSF que apresentaram os menores resultados de RMS da diferença do erro, para taxas de ocupação de 60% e 90%.	222
Diagrama 13 - Circuito completo implementado	223
Figura 27 - Representação gráfica do mapeamento da matriz H	224
Figura 28 - Densidade da matriz H	224
Figura 29 - Representação gráfica do mapeamento da matriz de autocorrelação A	225
Figura 30 - Densidade da matriz de autocorrelação A	225
Figura 31 - Densidade da matriz \hat{H}^+	226
Diagrama 14 - Esquemático representativo do circuito SIPO	226

Diagrama 15 - Esquemático representativo do circuito PISO	227
Diagrama 16 - Core do circuito do método SSF implementados neste trabalho	227

LISTA DE TABELAS

Tabela 1 - Comparaçāo entre os mētodos Filtro Casado, Filtro FDIP, DS e GD, valores em MeV	86
Tabela 2 - Comparaçāo entre os mētodos <i>Greedy</i>	87
Tabela 3 - Comparaçāo entre os valores de λ	90
Tabela 4 - Comparaçāo entre os valores de μ	91
Tabela 5 - Comparaçāo entre quantidades de elementos em H^+	93
Tabela 6 - Comparaçāo entre valores de ν	94
Tabela 7 - Comparaçāo entre os mētodos <i>Shrinkage</i> , sem inicializaçāo prévia do vetor y_0	98
Tabela 8 - Comparaçāo entre os mētodos <i>Shrinkage</i> , com inicializaçāo prévia do vetor y_0	99
Tabela 9 - Valores de RMSr dos dois melhores mētodos de cada uma das categorias estudadas	100
Tabela 10 - Referênciā para implementaçāo em <i>hardware</i>	103
Tabela 11 - Conjunto das possibilidades de configurações de construçāo de circuitos	106
Tabela 12 - Configurações obtidas para o circuito do mētodo SSF	107
Tabela 13 - Resultados da implementaçāo em hardware da melhor configuração para se obter o menor valor de RMSr	123
Tabela 14 - Resultados da implementaçāo em hardware da melhor configuração para se obter o menor circuito	123

LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analog-to-Digital Converter</i>
ALICE	<i>A Large Ion Collider Experiment</i>
ASIC	<i>Applications Specific Integrated Circuit</i>
ATCA	<i>Advance Telecommunications Computing Architecture</i>
ATLAS	<i>A Toroidal LHC Apparatus</i>
BC	<i>Bunch-Crossing</i>
BCID	<i>Bunch-Crossing IDentification</i>
BE	<i>Back-End</i>
CDF	<i>Collider Detector at Fermilab</i>
CERN	<i>Conseil Européen pour la Recherche Nucléaire</i>
CIS	<i>Charge Injection System</i>
CMS	<i>Compact Muon Solenoid</i>
COF	<i>Constrained Optimal Filter</i>
CP	Carga-Paridade
CPM	<i>Compact-Processing Modules</i>
CTP	<i>Central Trigger Processor</i>
D0	<i>DØexperiment</i>
DAQ	<i>Data Acquisition System</i>
DB	<i>Daughter Board</i>
DCS	<i>Detector Control System</i>
DM	<i>Deconvolution Matrix</i>
DP	<i>Detection Probability</i>
DS	<i>Dantzig-Selector</i>
DSP	<i>Digital Signal Processor</i>
EB	<i>Extended Barrel</i>
EF	<i>Event Filter</i>
EM	<i>Electromagnetic Calorimeter</i>
FA	<i>False Alarm</i>
FATALIC	<i>Front-end for ATLAS TileCal Integrated Circuit</i>
FDIP	Filtro FIR de Deconvolução por modelagem Inversa com respostas Positivas
FE	<i>Front-End</i>
FEB	<i>Front-End Board</i>
FIR	<i>Finite Impulse Response</i>
FPGA	<i>Field Programmable Gate Array</i>
FTK	<i>Fast TracK finder</i>
GD	Gradiente Descendente
HEC	<i>Hadronic End Cap</i>
HERA	<i>Hadron-Elektron-Ringanlage</i>
HL-LHC	<i>High Luminosity LHC</i>
HLT	<i>High Level Trigger</i>

IB	<i>Iteration Block</i>
IIR	<i>Infinite Impulse Response</i>
IRLS	<i>Iterative-Reweighted Least-Squares</i>
LAr	<i>Liquid Argonic</i>
LArg	<i>Liquid Argonic Calorimeter</i>
LB	<i>Long Barrel</i>
LEP	<i>Large Electron-Positron Collider</i>
LHC	<i>Large Hadron Collider</i>
LHCb	<i>Large Hadron Collider beauty</i>
LHCf	<i>Large Hadron Collider forward</i>
LP	<i>Linear Program</i>
LSq	<i>Least Square</i>
LS	<i>Line Search</i>
LSB	<i>Least Significant Bit</i>
LS-OMP	<i>Least-Square Orthogonal-Matching-Pursuit</i>
LS1	<i>First Long Shutdown</i>
LS2	<i>Second Long Shutdown</i>
LS3	<i>Third Long Shutdown</i>
LTI	<i>Linear Time-Invariant</i>
LVL1	<i>Level 1 - Primeiro Nível de Seleção de Eventos</i>
LVL2	<i>Level 2 - Segundo Nível de Seleção de Eventos</i>
MB	<i>Main Board</i>
MF	<i>Matched Filter</i>
MP	<i>Matching Pursuit</i>
nMCM	<i>new Multi-Chip Module</i>
OF	<i>Optimal Filter</i>
OMP	<i>Orthogonal-Matching-Pursuit</i>
PISO	<i>Paralel Input / Serial Output</i>
PCD	<i>Parallel-Coordinate-Descent</i>
PMT	<i>Photo Multiplier Tube</i>
PS	<i>Proton Synchrotron</i>
PU	<i>Processing Unit</i>
QIE	<i>Charge Integrator and Encoder</i>
RE	<i>Representação Esparsa</i>
RMS	<i>Root Mean Square</i>
RMSr	<i>RMS recuperado</i>
ROB	<i>Read Out Buffers</i>
ROC	<i>Receiver Operating Characteristic</i>
ROD	<i>Read Out Driver</i>
RoI	<i>Region of Interest</i>
RTM	<i>ATCA Rear Transition Module</i>
SIPO	<i>Serial Input / Paralel Output</i>

SCT	<i>Semi-Conductor Tracker</i>
SLC	<i>Stanford Linear Collider</i>
SNR	<i>Signal-to-Noise Ratio</i>
SR	<i>Sparse Representation</i>
SSF	<i>Separable Surrogate Functionals</i>
SSFi	<i>SSF with vector y_0 pre-initialized</i>
SSFls	<i>SSF plus Least Square</i>
SSFils	<i>SSFi+LS with vector y_0 pre-initialized</i>
SSFlsc	<i>SSF plus LS with a ν constant</i>
SSFilsc	<i>SSFi+LS with a ν constant and vector y_0 pre-initialized</i>
StD	<i>Standard Deviation</i>
StOMP	<i>Stage-wise Orthogonal-Matching-Pursuit</i>
TDAQ	<i>Trigger and Data Acquisition</i>
TDC	<i>Time Digital Conversor</i>
TileCal	<i>Tile Calorimeter</i>
TilePPr	<i>Tile PreProcessors</i>
TOTEM	<i>TOtal Elastic and diffractive cross section Measurement</i>
TRT	<i>Transition Radiation Tracker</i>
TT	<i>Trigger Tower</i>
WMP	<i>Weak Matching Pursuit</i>

SUMÁRIO

1	INTRODUÇÃO	20
1.1	MOTIVAÇÃO	21
1.2	OBJETIVOS	22
1.3	METODOLOGIA	23
1.4	ORGANIZAÇÃO DO TEXTO	23
2	O ACELERADOR LARGE HADRON COLLIDER E O EXPERIMENTO ATLAS	24
2.1	O ÁTOMO	24
2.2	O MODELO PADRÃO	26
2.3	ACELERADORES DE PARTÍCULAS	27
2.3.1	Aceleradores com alvo fixo e colisões de feixes	29
2.3.2	Aceleradores lineares e circulares	29
2.3.2.1	<i>Aceleradores lineares</i>	29
2.3.2.2	<i>Aceleradores circulares</i>	30
2.4	LUMINOSIDADE	30
2.5	DETECTORES DE PARTÍCULAS E SEUS SISTEMAS DE TRIGGER	31
2.6	O CONSEIL EUROPÉEN POUR LA RECHERCHE NUCLÉAIRE	32
2.7	O Large Hadron Collider	33
2.7.1	Cronologia das atualizações do Large Hadron Collider	33
2.7.2	Trem de <i>bunches</i>	34
2.7.3	Padrões de Colisão do <i>Large Hadron Collider</i>	36
2.8	O EXPERIMENTO ATLAS	37
2.8.1	Sistema de coordenadas do ATLAS	37
2.8.2	Composição do detector ATLAS	38
2.9	O CALORÍMETRO HADRÔNICO DO ATLAS	39
2.9.1	O sistema de seleção de eventos e aquisição de dados do ATLAS	40
2.9.1.1	<i>O primeiro nível de trigger do TileCal</i>	41
2.9.1.2	<i>Cronologia das atualizações do nível 1 de trigger do TileCal</i>	42
2.9.2	A eletrônica do <i>Trigger and Data Acquisition</i> do TileCal	42
2.9.3	A eletrônica do experimento ATLAS na Fase II de <i>upgrade</i>	43
2.10	CONCLUSÃO DA SEÇÃO 2	45
3	REVISÃO DOS MÉTODOS DE RECONSTRUÇÃO DE ENERGIA	46
3.1	ESTRATÉGIAS ENVOLVENDO FILTROS FIR	46
3.2	CONVOLUÇÃO E DECONVOLUÇÃO	48
3.3	DECONVOLUÇÃO ON-LINE POR FILTRO FIR	50
3.3.1	Filtragem adaptativa e modelagem inversa	50
3.3.1.1	<i>Modelagem inversa</i>	51

3.3.1.2	<i>Estimação utilizando least squares</i>	52
3.4	FILTRO FIR DESENVOLVIDO	53
3.5	GRADIENTE DESCENDENTE	54
3.6	CONCLUSÃO DA SEÇÃO 3	56
4	REPRESENTAÇÃO ESPARSA PARA IMPLEMENTAÇÃO ON-LINE	57
4.1	REPRESENTAÇÃO ESPARSA	57
4.1.1	Métodos para representação esparsa	59
4.2	MÉTODOS GREEDY	59
4.2.1	Matching-pursuit	60
4.2.2	Orthogonal-matching-pursuit	61
4.2.3	Least-squares orthogonal-matching-pursuit	63
4.3	MÉTODOS DE RESOLUÇÃO POR PROGRAMAÇÃO LINEAR	65
4.3.1	Dantzig-selector	65
4.4	MÉTODOS SHRINKAGE	67
4.4.1	Separable surrogate functionals	68
4.4.1.1	<i>Separable surrogate functionals line search</i>	71
4.4.1.2	<i>Separable surrogate functionals line search com um valor de v pré-definido</i>	72
4.4.2	Pré-inicialização do vetor y_0	74
4.4.3	Parallel-coordinate-descent	75
4.5	CONCLUSÃO DA SEÇÃO 4	79
5	ANÁLISES DOS MÉTODOS APRESENTADOS	80
5.1	BANCO DE DADOS DAS SIMULAÇÕES	80
5.1.1	Ruído incoerente	81
5.1.2	O pulso característico do TileCal e o desvio de fase	82
5.1.3	Geração do banco de dados para as análises	82
5.1.3.1	<i>Toy Monte Carlo</i>	83
5.2	OS MÉTODOS QUE JÁ FORAM PROPOSTOS	84
5.3	MÉTODOS GREEDY	87
5.4	MÉTODOS SHRINKAGE	88
5.4.1	Análises de parâmetros e configurações	89
5.4.1.1	<i>Análise do parâmetro λ</i>	89
5.4.1.2	<i>Análise do parâmetro μ</i>	90
5.4.1.3	<i>Análise dos parâmetros ω e ς</i>	91
5.4.1.4	<i>Análise do parâmetro v</i>	93
5.4.1.5	<i>Configurações recomendadas para implementação em hardware</i>	95
5.4.2	Análises de curvas ROC	97
5.5	COMPARATIVO ENTRE OS MÉTODOS	99
5.5.1	Comparativo em relação à complexidade de implementação em hardware	101
5.6	CONCLUSÃO DA SEÇÃO 5	103

6	IMPLEMENTAÇÃO DA VARIAÇÃO SSFi DO MÉTODO SSF ORIGINAL	104
6.1	O PROCESSO DE QUANTIZAÇÃO	104
6.2	DIAGRAMAÇÃO DO CIRCUITO	108
6.2.1	Estrutura física do circuito	108
6.2.2	Comportamento do circuito	109
6.3	DESENVOLVIMENTO DO CIRCUITO COMPLETO	110
6.3.1	Adaptação de matrizes	111
6.3.2	Circuitos auxiliares	113
6.3.2.1	<i>Primeiro pipeline</i>	113
6.3.2.2	<i>Segundo pipeline</i>	116
6.3.2.3	<i>Terceiro pipeline</i>	119
6.3.3	Círculo principal	119
6.4	RESULTADOS	122
6.5	CONCLUSÃO DA SEÇÃO 6	124
7	CONCLUSÃO E PRÓXIMOS PASSOS	125
7.1	PRÓXIMOS PASSOS	126
	REFERÊNCIAS	128
	APÊNDICE A – Ilustrações da seção 2	139
	APÊNDICE B – Ilustrações da seção 3	161
	APÊNDICE C – Ilustrações da seção 4	164
	APÊNDICE D – Ilustrações da seção 5	166
	APÊNDICE E – Ilustrações da seção 6	223
	APÊNDICE F – Produção Bibliográfica	228

1 INTRODUÇÃO

A Engenharia Elétrica vem evoluindo a uma velocidade crescente, sobretudo na área da instrumentação, em que projetos com muitos sensores em paralelo geram uma grande quantidade de dados a serem processados, a uma taxa cada vez maior. O desafio resultante do processamento de uma grande quantidade simultânea de dados (*big data*) tornou-se um dos pilares da engenharia moderna, refletindo no desenvolvimento de novas tecnologias.

Essas evoluções constantes da Engenharia Elétrica vêm colaborar com inúmeros desafios, dentre eles os da Física de Altas Energias¹ (PERKINS, 2000), fomentando assim o desenvolvimento de experimentos cada vez mais complexos, a exemplo dos executados no *Large Hadron Collider* (LHC) (CERN, 2014), o principal colisor de partículas² instalado no *Conseil Européen pour la Recherche Nucléaire* (CERN) (ANTHONY, 2014). Tal colisor é considerado a maior máquina já construída pela humanidade, até então, contendo cerca de 27 km de circunferência e encontra-se instalado a 100 m de profundidade na fronteira franco-suíça.

Como o aumento da complexidade dos experimentos e do consequente número de sensores é constante, ocorre uma maior taxa de aquisição, surgindo a necessidade de atualizações frequentes, abrindo, assim, espaço para pesquisas contínuas, sempre em busca de novas tecnologias para situações cada vez mais desafiadoras. Dentro do contexto de constantes atualizações, que buscam preparar o caminho para a prova experimental da matéria escura, por exemplo, que ainda permanecem apenas no campo da teoria, o LHC está se preparando para uma nova fase de operação.

O experimento *A Toroidal LHC ApparatuS* (ATLAS)³ (AAD *et al.*, 2008a), um dos experimentos presentes no LHC e foco desse trabalho, também será impactado por essas atualizações. Foi neste experimento, e também no *Compact Muon Solenoid* (CMS) (AGGLETON; COLLABORATION, 2017), onde o Bóson de Higgs foi comprovado experimentalmente em 2012, gerando, assim, o Prêmio Nobel de Física , em 2013 (THE. . . , 2014).

O detector do experimento ATLAS possui dois subdettores que foram desenvolvidos para medir a energia das partículas. O que ocupa a porção mais interna é o eletromagnético, também conhecido como calorímetro de argônio líquido - *Liquid Argonic Calorimeter* (LArg) (ALEKSA *et al.*, 2013), responsável por medir a energia das partículas que interagem de forma eletromagnética (partículas leves e carregadas, como fôtons e elétrons) oriundas das colisões. O segundo calorímetro do experimento ATLAS é o calorímetro hadrônico, ou calorímetro de telhas *Tile Calorimeter* (TileCal) (SOLODKOV, 2017), que mede a energia dos hâdrons que atravessam o calorímetro eletromagnético, por sua menor densidade.

¹Com o advento dos grandes aceleradores no cenário da Física mundial, boa parte das pesquisas na área da Física das Partículas se direcionou para a área que atualmente, designamos de Física das Altas Energias, pois para o estudo de partículas cada vez menores é necessário o uso de energias de colisão cada vez maiores.

²Um colisor de partículas é um instrumento utilizado em Física de Altas Energias, para a aceleração e colisão de partículas entre obstáculos, de forma a quebrar as partículas em partículas menores, possibilitando assim o estudo dos limites da matéria.

³O experimento ATLAS é um dos quatro experimentos que compõem o LHC. Cada experimento é instalado em um ponto de colisão no anel que integra o LHC. Tal experimento será melhor detalhado na próxima seção.

As interações nos subdetectores do ATLAS chegam a gerar um fluxo de dados de cerca de 60 TB/s resultantes da colisão de dois feixes de prótons, com 2.808 grupos (*bunches*) de 10^{11} partículas cada um, que viajam em direções opostas, a uma velocidade de 99.9998% da velocidade da luz e colidem a uma taxa de 40MHz (KLOUS, 2010). Para processar estes dados, o ATLAS utiliza um sistema de *trigger* que seleciona os eventos que devem ser armazenados e os que podem ser ignorados. Esse sistema de *trigger* é, de fato, um sistema de filtragem *on-line* composto por dois níveis.

Este trabalho está inserido, mais especificamente, no primeiro nível da filtragem no que diz respeito à informação de energia das partículas que interagem com o TileCal. Neste primeiro nível de *trigger*, a solução deve ser implementada em *hardware* devido à alta taxa de eventos (40MHz). É neste nível de *trigger* que o sinal elétrico proveniente das células dos calorímetros (cerca de 10.000 células no caso do TileCal) (VALERO; COLLABORATION, 2017) é digitalizado a uma taxa de 40MHz e processado para estimar a energia depositada em cada célula.

A eletrônica de *front-end* produz um pulso elétrico conformado com duração temporal de 150ns, enquanto a taxa de colisão do LHC tem duração temporal de 25ns (AAD *et al.*, 2008a). Observa-se que essa diferença nos tempos de resposta do calorímetro em relação a taxa de colisão do LHC, possibilita a ocorrência da sobreposição de amostras de eventos subsequentes, ou seja, o empilhamento de sinais, *pile-up* (CLELAND; STERN, 1994). É, também, nesse nível que ocorrerá o maior impacto proveniente do acréscimo de probabilidade de colisões, previsto para ocorrer com as atualizações projetadas para os próximos anos. Com isso, é necessário encontrar métodos mais robustos e eficientes para operarem nesse novo ambiente com alta taxa de eventos com ocorrência de *pile-up*.

1.1 MOTIVAÇÃO

O aumento da energia das colisões e da quantidade de partículas por *bunch*, que vêm acontecendo no acelerador de partículas LHC favorecem o crescimento de eventos de *pile-up*. Isso se dá em decorrência das constantes atualizações nos últimos anos, fazendo com que os algoritmos atualmente em uso percam parte da eficiência de reconstrução da energia das partículas. É necessário, portanto, estudar outros métodos que sejam capazes de lidar com esse novo cenário.

Na área de processamento digital de sinais em sistemas lineares invariantes no tempo *Linear Time-Invariant* (LTI), a convolução pode ser modelada como uma relação linear entre a entrada e a saída. Conhecendo a resposta ao impulso é possível calcular a saída de um sistema LTI para qualquer sinal de entrada (MITRA, 2010).

Neste trabalho, o TileCal foi modelado como um sistema LTI no tempo discreto, de modo que a sua resposta ao impulso englobará toda a cadeia de processamento interno do TileCal. A partir dessa modelagem, a literatura atual disponibiliza trabalhos que propõem o uso de deconvolução, para serem utilizados de forma *on-line*, principalmente baseadas em Filtros

Finite Impulse Response (FIR) (MITRA, 2010).

Discretos métodos não causais de deconvolução que utilizam algoritmos iterativos e que trabalham a estimativa de forma simultânea, com várias amostras dentro de uma mesma janela são, também, encontrados na literatura de sistemas. Tais métodos tendem a apresentar melhores resultados em relação aos baseados em Filtros FIR (DUARTE, 2015). Todas essas pesquisas possuem implementações que necessitam de um tempo maior de processamento e, por esse motivo, são ideais para serem utilizadas de forma *off-line* (BARBOSA, 2017). Dentre esses métodos, uma categoria que vem apresentando ótima relação custo-benefício é a daqueles baseados em representação esparsa de dados.

Uma das características do LHC, que será explorada nesse trabalho, diz respeito a forma como ele foi projetado para evitar o efeito, descrito por (HERR, 1996), conhecido como PACMAN, ou seja, nas regiões onde as colisões de *bunches* produzem iterações de longo alcance, tais iterações podem interferir em outras iterações que ocorrem na mesma região. Nessas situações, esses *bunches* que produziram iterações de longo alcance são chamados de *bunches* PACMAN. O LHC, então, envia uma sequência de pacotes de prótons (*bunches*), espaçados de 25ns no tempo entre cada *bunch*, seguido de um período de *bunches* vazios, para se evitar o efeito PACMAN, nessa configuração. Esse padrão faz com que o sinal adquirido na eletrônica das células dos calorímetros seja interpretado como uma janela fixa de aquisição, onde o processo de deconvolução deve ser executado, seguido de um período sem sinal.

Essa configuração, além de impossibilitar o efeito PACMAN pode, ainda, impedir outro efeito: o de borda, que é quando as caldas de sinais que não ficaram completamente inseridas em uma única janela acabam por interferir com a próxima janela (DUARTE *et al.*, 2016). Uma vez definidos os tamanhos da janela contendo sinais e do intervalo sem sinais, esse último pode ser considerado como possuidor apenas de valores zero, ignorando as caldas residuais dos sinais. Portanto, baseando-se nesse configuração do LHC, torna-se viável a implementação de métodos iterativos de deconvolução em janelas no ambiente *on-line* do sistema de *trigger*, desde que o processamento seja rápido o suficiente para que o processo termine antes da próxima janela de *bunches*.

Sendo assim, este trabalho propõe o uso de métodos janelados baseados em representação esparsa de dados, de forma a possibilitar a deconvolução *on-line*, dos sinais oriundos das células de leitura do TileCal, para serem implementados no *hardware* do primeiro nível de *trigger* do detector do experimento ATLAS.

1.2 OBJETIVOS

Este trabalho tem como principal objetivo a implementação em *hardware* de um método *on-line* de estimativa de energia para calorimetria, que seja robusto em cenários de empilhamento de sinais e que apresente melhores resultados que os obtidos pelos métodos atualmente em uso, concomitantemente. Para tal, métodos baseados em representação esparsa de dados foram projetados, modificados e tiveram seus resultados comparados de acordo com critérios de seleção

previamente definidos.

Entre outros objetivos complementares, estão: a adaptação de métodos janelados iterativos, para serem implementados em um ambiente *free-running* e a análise da capacidade de implantação de diversos canais em paralelo nos dispositivos *Field Programmable Gate Array* (FPGA), conforme Meyer-Baese (2007).

1.3 METODOLOGIA

A metodologia aplicada na condução deste trabalho inclui o teste de algoritmos de representação esparsa de dados das famílias *Greedy* e *Shrinkage* (ELAD, 2010), de forma a implementar, em *hardware* reconfigurável, (FPGA) aquele mais adequado aos critérios de: menor complexidade de implementação em *hardware*, menor erro de detecção e estimação em relação aos obtidos com o algoritmo atualmente em uso e latência compatível com as exigências do sistema de *trigger L1* do ATLAS.

1.4 ORGANIZAÇÃO DO TEXTO

Esta tese foi escrita apresentando a seguinte organização:

- a. seção 2: apresentação do ambiente no qual este trabalho foi realizado, com destaque no experimento ATLAS e seu sistema de calorimetria, percorrendo um breve histórico do desenvolvimento do conhecimento humano desde os primeiros pensamentos a respeito do que seria um átomo, chegando até a eletrônica do experimento ATLAS;
- b. seção 3: exposição do desenvolvimento teórico e matemático do método atualmente em uso no primeiro nível do detector TileCal, do experimento ATLAS, e do desenvolvimento teórico matemático de métodos que vêm sendo propostos pela colaboração brasileira no ATLAS;
- c. seção 4: desenvolvimento teórico matemático semelhante ao executado na seção anterior, com a diferença de que aqui os métodos apresentados fazem parte da nova proposta deste trabalho, sendo tais métodos focados em representação esparsa, com algoritmos das famílias *Greedy* e *Shrinkage* e uma adaptação para uso *on-line* do algoritmo *Dantzig-Selector* (DS) descrito por Barbosa (2017). Tal estudo teve como objetivo a implementação *on-line* de um dos métodos já propostos de representação esparsa para uso *off-line*;
- d. seção 5: análise dos métodos apresentados nas seções 3 e 4, formando o embasamento matemático que servirá como ponto de partida ao que será realizado na seção 6;
- e. seção 6: apresentação da implementação do método que obteve a melhor aderência aos objetivos deste trabalho, dentre os estudados e analisados nesta tese, a partir dos resultados de detecção e estimação e com os critérios de seleção mais bem detalhados;
- f. seção 7: finalmente, é apresentada a conclusão deste trabalho.

2 O ACELERADOR LARGE HADRON COLLIDER E O EXPERIMENTO ATLAS

Nesta seção será apresentada uma breve introdução à história da construção do conhecimento humano na área de Física de Altas Energias, tendo como foco principal o detalhamento do experimento ATLAS, principalmente em relação ao seu sistema de *trigger*.

2.1 O ÁTOMO

O interesse e o esforço por compreender as leis naturais que regiam o mundo ocorrem desde o início da história humana, com registros encontrados em todas as civilizações antigas, da China ao Egito passando pela Mesopotâmia (BRENNAN, 2012) e chegando até a Grécia antiga. É na Grécia Antiga, pelos esforços dos filósofos naturais, que surge a primeira tentativa de compreender as leis naturais de forma sistematizada, com o intuito de racionalizar o mundo sem recorrer à intervenção divina.

Nesse contexto, na região da atual Turquia, nasce Tales de Mileto (aproximadamente 624 a.C. até aproximadamente 546 a.C.). Tales seria a primeira pessoa à qual se pode chamar de cientista e filósofo e é tido como aquele que levou a matemática e a astronomia, aprendidas em seus estudos no Egito, à Grécia, (ROONEY, 2013).

Por volta do século V a.C., Leucipo – também de Mileto como Tales – e seu aluno Demócrito – de Abdera (460-370 a.C.) – formulam as primeiras hipóteses sobre os componentes essenciais da matéria (BERRYMAN, 2016), incluindo a ideia inicial da infinitude e indivisibilidade do átomo.

Segundo Aristóteles (384-322 a.C.), eles, os atomistas, retomam dos Eleatas a ideia do Ser para conceder o elemento constitutivo do mundo: o átomo (SANTOS, 2009). O átomo, aqui, é um resumo do Ser de Parmênides e, assim, é engendrado e imperecível, mas reiterado até ao infinito sob uma grande variedade de formas geométricas.

Além do atomismo grego, há também o atomismo indiano, porém não há um consenso entre os historiadores de qual teria sido a fonte primordial, pois o principal expoente indiano foi o filósofo Kanada (Kashyapa) que pode ter vivido no século VI a.C. (portanto anterior aos gregos) ou no século II a.C. (portanto posterior aos gregos), esta indefinição de datas, tão díspares, fomenta a dúvida quanto a originalidade da teoria (ROONEY, 2013).

A teoria dos átomos de Kanada complementou a teoria elementar uma vez que ele propôs cinco tipos diferentes de átomo, um para cada um dos cinco elementos que compunham o modelo indiano da matéria; fogo, água, terra, ar e éter, o mesmo que no modelo de Aristóteles. Os átomos (ou paramanu) são atraídos uns para os outros e todos se agrupam. (ROONEY, 2013, p. 32).

Ambos os atomismos, grego e indiano, foram retomados pelos primeiros estudiosos árabes, aproximadamente 1500 anos depois dos primeiros estudos. Após traduzidos e comentados, a versão islâmica foi a que obteve maior sucesso, baseada no trabalho de Asharite de al-Ghazali

(1058-1111). Para ele, os átomos são as únicas coisas materiais eternas e tudo mais dura apenas um instante e é considerado "accidental" (GRIFFEL, 2016).

Nicholas de Autrecourt (c.1298-c.1369), possuidor da visão de que a matéria seria baseada em partículas minúsculas chamadas de "corpúsculos", fez o debate ferver em Paris, na época em que a capital Francesa era o centro intelectual da Europa. Autrecourt acreditava que toda matéria, espaço e tempo eram compostos por átomos, pontos e instantes e que toda mudança nada mais é que o resultado da reorganização de átomos (THIJSSEN, 2016).

No século XVII, uma variante do atomismo inicial tornou-se popular contando com o apoio do químico Robert Boyle (1627-1691), do filósofo Pierre Gassendi (1592-1655) e de Isaac Newton (1642-1727), dentre outros. Tal variante ficou conhecida como "corpuscularianismo" em que, diferentemente do atomismo original, os corpúsculos não precisam ser indivisíveis (FISHER, 2014).

O atomismo renasce quando, em 1649, Pierre Gassendi (1592-1655) publica sua teoria a respeito, explicitando que tudo o que acontecia decorria do movimento e da interação de partículas minutias, que seguiam as leis naturais, excluindo os seres pensantes do seu esquema. A teoria de Gassendi não alçou grande influência, por sofrer uma oposição direta de Descartes às suas ideias. Em 1661, o atomismo retoma sua força quando Robert Boyle publica *The Sceptical Chymist* (BOYLE, 2007), descrevendo um universo formado totalmente por átomos e aglomerados de átomos, todos em movimento contínuo (MACINTOSH; ANSTEY, 2014).

O primeiro trabalho moderno sobre química, produzido por Antoine Lavoisier (1743-1794), em 1789, incluía 33 elementos, incluindo luz e "calórico", que se pensava ser um fluido produtor de perda ou ganho de calor por meio do movimento. O químico inglês John Dalton (1766-1844) continuou a desenvolver as ideias de Lavoisier após sua morte e, em um trabalho publicado em 1808, estabeleceu as bases da moderna teoria atômica (THORPE, 1808).

Em 1827, Robert Brown (1773-1858), botânico escocês, obtém a primeira evidência visual indicando que a matéria é composta de partículas minúsculas. Enquanto examinava minúsculos grãos de pólen na água com o microscópio, Brown notou que eles se moviam constantemente, como se alguma coisa invisível os batesse. Mesmo não conseguindo explicar o que viu, hoje chamamos esse fenômeno de movimento browniano (ROONEY, 2013).

Em 1905 Albert Einstein (1879-1955) cria um modelo matemático para explicar o movimento browniano, tornando-se a base para as primeiras estimativas do tamanho das moléculas. Em 1926, o Prêmio Nobel de Física é destinado ao francês Jean Baptiste Perrin (1870-1942), por ter conseguido, em 1908, medir o tamanho de uma molécula de água usando o modelo matemático concebido por Einstein e o trabalho realizado por James Clerk Maxwell (1831-1879) (BRENNAN, 2012).

A indivisibilidade do átomo já estava sendo questionada, mesmo antes da prova definitiva de sua existência. Ainda em 1897, o elétron era descoberto pelo físico Joseph John Thomson (1856-1940). Ao estudar os raios X e catódicos, Thomson identifica partículas de massa aproximadamente 1.800 vezes menores que a do átomo mais leve (LIMA, 2007).

Em 1911, Ernest Rutherford (1871-1937) mostrou que o átomo aparentava ser como um sistema solar, constituindo-se de um núcleo de cargas positivas (prótons, por ele descobertos em 1918) e de elétrons girando em seu redor, como planetas. Esse modelo foi aperfeiçoado por Niels Bohr (1885-1962), em 1913, de tal forma que os elétrons conseguiam permanecer em órbita, ao invés de vagarem pelo espaço fora do núcleo, seguindo qualquer trajetória (ROONEY, 2013; VIEIRA, 2015).

A busca por mais partículas atômicas, ocorrida no início dos anos de 1930, mobilizou nomes como Walter Bothe (1891-1957), Irène Joliot-Curie (1897-1956, filha de Marie e Pierre Curie) e seu marido Frédéric Joliot-Curie (1900-1958) levando-os a descobrir, em janeiro de 1932, que um tipo de radiação produzida pela radiação de partículas alfa no berílio expulsava "algo" de outros elementos. Esse "algo" foi chamado de nêutron por James Chadwick (1881-1974), em fevereiro de 1932, após repetir os experimentos dos Joliot-Curie e concluir que esse outro tipo de partícula subatômica, não possuía carga (VIEIRA, 2015).

Com o objetivo de explicar o que poderia manter as partículas positivas e neutras unidas no núcleo atômico, Hideki Yukawa (1907-1981) sugeriu, em 1934, a existência da força nuclear forte, que seria carregada por partículas chamadas mésons. Daí, o verdadeiro mediador da força nuclear forte descoberto, em 1947, por três físicos – o inglês Cecil Powell (1903-1969), o brasileiro César Lattes (1924-2005) e o italiano Giuseppe Occhialini (1907-1993) – vem a ser o méson π , ou píon (BRENNAN, 2012).

Até 1953, os neutrinos, partícula sugerida inicialmente por Wolfgang Pauli (1900-1958) em 1930, nomeada por Enrico Fermi (1901-1954) em 1933, e detectada por Frederick Reines (1918-1998) e Clyde Cowan (1919-1974), não haviam saído da teoria (VIEIRA, 2015).

2.2 O MODELO PADRÃO

O Modelo Padrão é a teoria que descreve as partículas fundamentais (indivisíveis) e suas interações, de modo a formar a estrutura da matéria visível. Formulado entre 1970 e 1973, o modelo explica cerca de 5% da matéria conhecida do universo e não engloba a força gravitacional. Os outros 95% são conhecidos como matéria escura por não terem sido, ainda, detectados diretamente pelos instrumentos atualmente desenvolvidos (VIEIRA, 2015).

Conforme pode ser visualizado na Figura 1 (Apêndice A), o Modelo Padrão possui três gerações: a primeira geração é de todas as partículas, das quais toda a matéria detectável diretamente é composta; as partículas das outras duas gerações são instáveis, ou seja, decaem rapidamente para a primeira geração, necessitando da realização de experimentos de alta energia para o seu estudo direto (MOREIRA, 2009).

As partículas que não possuem estrutura interna são conhecidas como férmons e se dividem em duas classes. A primeira classe é a dos léptons – como elétron, muon, tau e seus respectivos neutrinos – e a segunda é a dos quarks, que se desdobra em seis tipos, a saber: *up*, *down*, *charm*, *strange*, *bottom* e *top*. Os hadrons possuem estrutura interna por serem formados por quarks e antiquarks e se subdividem em bárions, que são formados por três quarks (antiquark),

e em mésons, formados por um quark (antiquark) (BRENNAN, 2012). O elétron é a partícula mais popularmente conhecida, com destaque para o lépton. Entre os hadrons, os mais conhecidos são o próton e o nêutron. O próton é formado por dois quarks up e um down e o nêutron formado por dois quarks down e um up. Em um nível mais fundamental da matéria, os férmons são partículas com spin fracionário e obedecem ao princípio da exclusão de Pauli, ou seja, férmons idênticos não podem compartilhar do mesmo estado quântico (MOREIRA, 2009), fato importantíssimo para formação de átomos e moléculas.

Quanto às quatro forças conhecidas, o Modelo Padrão descreve três:

- a. a força forte: com os glúons como mediadores, aglutinando os quarks nos núcleos atômicos;
- b. a força fraca: é mediada pelos bósons W e Z, responsáveis por algumas formas de decaimento radioativo;
- c. o eletromagnetismo: os fótons são os mediadores e responsáveis por incorporar os átomos em moléculas e as moléculas em matéria macroscópica.

Já a quarta força, a gravidade, é postulada pelos grávitons, mas, até o momento, esse tipo de partícula permanece no campo da possibilidade teórica, ainda não sendo observado (BRENNAN, 2012).

No que se refere ao bóson de Higgs, como ficou conhecido, foi previsto teoricamente em 1964 por Peter Higgs e comprovado experimentalmente em 2012, após a identificação de um decaimento em massa na região entre 124 e 126 GeV, com desvio padrão (σ) de 2,9 e de 3,1, respectivamente (AAD *et al.*, 2012).

2.3 ACELERADORES DE PARTÍCULAS

Os aceleradores de partículas são os instrumentos específicos mais complexos desenvolvidos para possibilitar o estudo da estrutura da matéria. Tais instrumentos resultam de vários anos de estudos dentro da física e possibilitaram a criação da Física de Altas Energias. Assim, uma vez que se utilizam instrumentos, cada vez mais complexos, é possível a descoberta de partículas cada vez menores, à medida que a energia das colisões é aumentada.

O desenvolvimento do primeiro acelerador de partículas é atribuído a Rutherford, como parte das várias técnicas e ferramentas experimentais que ele vinha desenvolvendo em seu estudo sobre a estrutura do átomo, ainda na primeira década de 1900. Esse aparelho era capaz de dirigir um feixe de partículas alfa de uma fonte radioativa até um alvo e era do tipo linear (BRENNAN, 2012). Já o primeiro acelerador de partículas circular (ciclotron) foi construído por Ernest Lawrence (1901-1958) em 1931, rendendo-lhe o Nobel de Física de 1939 e media apenas 13cm de diâmetro (VIEIRA, 2015).

O acelerador de partículas é, portanto, um equipamento construído para acelerar e aumentar a energia de um feixe de partículas, por meio da geração de campos elétricos que as aceleram e campos magnéticos suficientemente fortes para orientar e focalizar tais feixes de partículas (CERN, 2012). Ao se estudar objetos cada vez menores, os limites da mecânica newtoniana são ultrapassados e, a partir desse ponto, as leis da mecânica quântica passam a operar.

Dentro dessa visão, podemos comparar os aceleradores de partículas aos grandes microscópios. Uma vez regida pelas leis da mecânica quântica, a partícula passa, agora, a ser vista como uma entidade dual.

Essa dualidade denomina-se partícula-onda e muitos aspectos do seu comportamento dual só podem ser descritos em termos de probabilidades. Ou seja, essa entidade dual é modelada como sendo inherentemente espalhada, não podendo ser descrita como um ponto (mecânica newtoniana) com posição e velocidade determinadas.

Visualizando essa entidade dual como uma nuvem de probabilidades, com suas dimensões comparáveis ao comprimento de onda, a partícula fisicamente visível pode ser encontrada em algum ponto dessa nuvem de probabilidades. Portanto, ao se reduzir o comprimento de onda reduz-se, também, o volume da nuvem que, para tal, é necessário aumentar o momento da partícula e, portanto sua energia. Dessa forma, objetos com dimensões cada vez menores podem ser estudados. Portanto, ao aumentar a energia do experimento, podemos enxergar objetos cada vez menores (PEREIRA, 2014).

Segundo o princípio da incerteza de Werner Heisenberg (1901-1976), estabelecido em 1927, ao serem considerados a posição x , o momento p_x e as suas incertezas respectivas, como Δx e Δp_x , eles não podem ser simultaneamente conhecidos com uma precisão melhor que $\Delta x \Delta p_x \approx \hbar/2$, onde Δx e Δp_x são o desvio padrão da posição e do momento, respectivamente. Para se obter uma relação com a energia basta efetuar uma multiplicação com c , $\Delta x \Delta E \approx \hbar/2$.

Para explorar dimensões da ordem de Δx , é necessária uma energia de $E(MeV) \approx 2\Delta E(MeV) \approx c/\Delta \approx 1,973 \times 10^{11}(MeVcm)/\Delta x(cm)$. O Quadro 1 lista algumas ordens de grandeza da magnitude mínima de energia necessária para explorar algumas dimensões Δx (BRAIBANT; GIACOMELLI; SPURIO, 2011).

Quadro 1 - Ordem de grandeza da energia mínima necessária e as ferramentas utilizadas para explorar distâncias Δx

$\Delta x cm$	E	Ferramentas Utilizadas
10^{-5}	2eV	Microscópios
10^{-8}	2keV	Raios-X
10^{-11}	2MeV	Raios- γ
10^{-14}	2GeV	Aceleradores de Partículas
10^{-16}	200GeV	Aceleradores de Partículas
10^{-17}	2TeV	Aceleradores de Partículas

Fonte: Braibant, Giacomelli e Spurio (2011, p. 46, tradução nossa).

Outro uso de aceleradores de partículas está ligado à criação de partículas, pois segundo os estudos de relatividade de Albert Einstein (1879-1955) em 1905, quando se chega à equação $E = mc^2$, descobre-se que energia e matéria podem ser convertidas uma em outra. Dessa forma,

os aceleradores podem criar partículas fazendo com que outras duas partículas se choquem, possibilitando assim o estudo das partículas fundamentais (ROONEY, 2013).

2.3.1 Aceleradores com alvo fixo e colisores de feixes

Os aceleradores com alvo fixo lançam um conjunto de partículas contra um alvo fixo promovendo, assim, uma colisão entre ambos. O espalhamento das partículas resultantes dessa colisão fornece informações importantes sobre a estrutura do feixe e do alvo.

Por sua vez, os colisores de feixes são aceleradores de partículas que promovem a colisão entre dois feixes de partículas, que são acelerados e direcionados para colidirem com igual momento e sinais opostos, em pontos específicos, utilizando melhor a energia envolvida na colisão. O uso de um acelerador circular é essencial para se obter altas energias (BRAIBANT; GIACOMELLI; SPURIO, 2011).

2.3.2 Aceleradores lineares e circulares

A composição básica de um acelerador inclui: uma fonte de íons, um campo acelerador e um campo guia que força a partícula a se mover em uma órbita bem definida. A partícula acelerada caminha em uma região de alto-vácuo e, portanto, as colisões das partículas com moléculas de gás residual são reduzidas, diminuindo a perda de energia ao longo do caminho. A fonte de íons se encontra em uma pequena cavidade mantida a baixa pressão. Nessa câmara, uma descarga elétrica ioniza continuamente gás hidrogênio e um campo elétrico é utilizado para orientar os elétrons e prótons às próximas etapas do acelerador.

2.3.2.1 Aceleradores lineares

Existem duas técnicas distintas, a depender do tipo de partícula que será utilizada (prótons ou elétrons, por exemplo), para a construção dos aceleradores lineares. Os aceleradores lineares de elétrons, Figura 2a (Apêndice A), possuem eletrodos de mesmo comprimento, uma vez que mesmo a baixas energias os elétrons viajam na velocidade da luz (BRAIBANT; GIACOMELLI; SPURIO, 2011), e são muito utilizados na medicina, por exemplo, em radioterapias que utilizam aceleradores lineares de feixe de elétrons e fôtons de alta energia no tratamento do câncer. De outra forma, os aceleradores lineares de prótons de baixa energia são constituídos de um conjunto de eletrodos cilíndricos conectados entre si e uma tensão oscilante com frequência constante, como mostrado na Figura 2b (Apêndice A). Nessa configuração, os prótons podem atingir energias de até 20MeV em aproximadamente 70 metros. Os prótons estão em uma região livre de campo elétrico dentro dos tubos, se movendo a uma velocidade constante, enquanto que entre tubos vizinhos, os prótons são acelerados por um campo elétrico. Se o tubo escolhido possuir um comprimento em que o tempo necessário para a partícula cruzá-lo seja igual à metade do período do oscilador, os prótons são acelerados uma segunda vez quando passarem para o próximo tubo e, assim, sucessivamente serão acelerados. Portanto, os comprimentos dos tubos

devem ir aumentando levando-se em conta o aumento da velocidade do próton (PEREIRA, 2014).

2.3.2.2 Aceleradores circulares

Os aceleradores circulares, se comparados aos aceleradores de alvo fixo, têm a desvantagem de serem menos versáteis e capazes de produzir um número menor de interações por unidade de tempo (BRAIBANT; GIACOMELLI; SPURIO, 2011). Podem ser classificados como:

- a. cyclotrons: nestes dispositivos, comumente chamados ciclotrons clássicos, o campo elétrico de aceleração oscila a uma frequência fixa, e o campo magnético orientador tem uma intensidade fixa. Um exemplo de um ciclotron clássico pode ser visualizado na Figura 3 (Apêndice A) (SUTTON, 2013);
- b. synchrocyclotrons: os ciclotrons nos quais a frequência da tensão de aceleração é alterada à medida que as partículas são aceleradas são chamados de synchrocyclotrons, ciclotrons modulados em frequência, ou phasotrons. A modulação faz com que as partículas não saiam de fase com a tensão de aceleração. Deste modo, o aumento de massa relativística não impõe um limite à energia. Além disso, a focagem magnética pode tornar-se mais forte, de modo que o campo magnético não precise ser moldado com tanta precisão (Figura 4 Apêndice A) (SUTTON, 2013).
- c. synchrotrons: à medida que as partículas em um synchrotron são aceleradas a força do campo magnético é aumentada para manter o raio da órbita aproximadamente constante. Esta técnica tem a vantagem de que o ímã necessário para formar as órbitas de partículas é muito menor que o necessário em um ciclotron para produzir as mesmas energias de partículas, Figura 5 (Apêndice A) (SUTTON, 2013). Ainda em relação a Figura 5 (Apêndice A) (SUTTON, 2013), as partículas são injetadas no anel síncrotron (mostrado à esquerda) com suas energias já elevadas por um acelerador linear. Eles são ainda mais acelerados em torno do síncrotron por uma série de eletroímãs, cujos campos aplicados ficam mais fortes à medida que a velocidade das partículas aumenta. O feixe de partículas é focalizado pelas pontas dos polos, mostradas em corte transversal à direita. Pontas com seção cruzada c-d focam o raio na direção radial, enquanto pontas com seção transversal a-b focam na direção vertical

2.4 LUMINOSIDADE

A luminosidade \mathcal{L} de um colisor é definida como o número que, multiplicado pela seção transversal total de um dado processo físico (σ), fornece o número total de colisões por unidade de tempo N, isto é, $N = \mathcal{L}\sigma$.

A luminosidade pode ser expressa através dos parâmetros do acelerador. Considerando um acelerador com raio R, onde: n_p é o número de partículas em cada pacote circulando em uma direção; N_p o número de pacotes; n_a o número de partículas em sentido oposto; N_a o número de pacotes de partículas em direção oposta; e r a média do raio transversal de cada pacote. Então, a luminosidade pode ser descrita pela equação (1) (BRAIBANT; GIACOMELLI; SPURIO, 2011),

sendo f a frequência de circulação dos pacotes no acelerador, $f = 1/\tau$, onde τ é o período de revolução, $\tau = \frac{2\pi R}{c}$. O fator G leva em conta o comprimento finito de um pacote e usualmente é $G \approx 1$.

$$\mathcal{L} = \frac{fn_p n_a N_p N_a G}{4\pi r^2} \quad (1)$$

Em relação a um colisor circular ultrarrelativista, que possua feixes com dimensões iguais em ambas as direções transversais a luminosidade \mathcal{L} é igual a (SHILTSEV *et al.*, 2008):

$$\mathcal{L} = \gamma f_0 \cdot \frac{N_1 N_2 N_B}{4\pi \epsilon \beta^*} \cdot H \quad (2)$$

onde: γ denota o fator relativista gama; f_0 a frequência de revolução; N_1 é a representação dos *bunches* do feixe 1; N_2 é a representação dos *bunches* do feixe 2; N_B é o número total de *bunches*; ϵ é a emitância normalizada, que é relativa ao *Root Mean Square* (RMS) transverso do feixe de tamanho σ no ponto de interação (IP), através da função beta no IP β^* , via $\epsilon = \gamma \sigma^2 / \beta^*$; finalmente, H que é um fator de redução geométrico (menor que uma unidade).

Em geral, a maioria dos eventos produzidos nas interações, os chamados eventos de "Minimum Bias" não são de interesse (BRAIBANT; GIACOMELLI; SPURIO, 2011). Atingir uma alta luminosidade é, por essa razão, um dos principais objetivos dos aceleradores de partículas, muitas vezes tão importante quanto atingir um elevado nível de energia.

O Quadro 2 mostra alguns dos principais aceleradores construídos nas últimas décadas e suas respectivas características como: energias no centro de massa, tipo de feixes e luminosidade (BRAIBANT; GIACOMELLI; SPURIO, 2011; PEREIRA, 2014).

Quadro 2 - Características dos aceleradores mais recentes

Acelerador	Ano	Feixes	Energia (GeV)	$\mathcal{L} cm^{-2}s^{-1}$
SPPS(CERN)	1981	pp	630 - 900	6×10^{30}
Tevatron(FNAL)	1987	pp	1800 - 2000	10^{31-32}
SLC(SLAC)	1989	e^+e^-	90	10^{31-32}
LEP(CERN)	1989	e^+e^-	90 - 200	10^{31-32}
HERA(DESY)	1992	ep	300	10^{31-32}
RHIC(BNL)	2000	pp, AA	200 - 500	10^{32}
LHC(CERN)	2010-2012	pp, pA, AA	7K - 8K	$\sim 0.76 \times 10^{34}$
SLHC(CERN)	2015	pp, pA, AA	14K	10^{34}

Fonte: Pereira (2014, p. 11).

2.5 DETECTORES DE PARTÍCULAS E SEUS SISTEMAS DE TRIGGER

Os detectores de partículas são instrumentos responsáveis por detectar as partículas e as subpartículas geradas das colisões entre as partículas aceleradas e estão instalados dentro dos aceleradores de partículas. O sistema de *trigger*, nos primeiros detectores, estimava a energia

das partículas de forma analógica e utilizava capacitores para tal finalidade. Tais capacitores permitiam que o sistema subsequente identificasse os pulsos conformados, ao atingirem os patamares definidos . Exemplos de detectores que se utilizavam de tal arquitetura podem ser encontrados em *Large Electron-Positron Collider* (LEP) (RÜDEN, 1985), Tevatron (AMIDEI *et al.*, 1988), *Stanford Linear Collider* (SLC) (BRIGGS *et al.*, 1989) e *Hadron-Elektron-Ringanlage* (HERA) (ABT; COLLABORATION, 1997; CARLIN *et al.*, 1995).

Com o avanço da eletrônica digital, possibilitando uma expansão na largura do pulso, e com o surgimento de um novo tipo de problema, o empilhamento de sinais adjacentes, ou *pile-up*, identificado inicialmente em (CLELAND; STERN, 1994), começam a surgir estudos de outras técnicas de detecção de sinais, visando solucionar o *pile-up*, conforme citado por Bystricky *et al.* (2004).

As soluções mais comumente adotadas para a aquisição de dados no primeiro nível de *trigger* são as baseadas no uso de filtros FIR. Essas soluções começam a ganhar destaque, apesar da realização de estudos comparativos comprovados em Fermilab (2002), para a adoção de alguns algoritmos, tais como deconvolução linear, detecção de picos e de médias, além de filtro FIR seguido por detector de picos, por exemplo.

É nesse cenário que os experimentos do LHC como, o *Large Hadron Collider beauty* (LHCb) (ALVES; ANDRADE-FILHO; COLLABORATION, 2008; BOURGEOIS; FITZ-PATRICK; STAHL, 2018; COLLABORATION, 2003), o *A Large Ion Collider Experiment* (ALICE) (KRIVDA *et al.*, 2007; LIJIAO, 2011; RONCHETTI *et al.*, 2012) e o CMS (AGGLETON; COLLABORATION, 2017; ROSE, 2009) são projetados e construídos com uma eletrônica capaz de lidar com esses problemas. Essa eletrônica vem evoluindo à medida em que novos estudos são feitos e novos desafios são traçados (COLLABORATION, 2017b; AMBLER; VACHON, 2018).

2.6 O CONSEIL EUROPÉEN POUR LA RECHERCHE NUCLÉAIRE

A primeira proposição oficial da criação de um laboratório de pesquisa nuclear que pudesse reunir a Europa pós-guerra foi feita por Louis de Broglie, em 1949. A ideia não foi muito bem-vista, inicialmente, mas o *Conseil Européen pour la Recherche Nucléaire* (CERN) nasce oficialmente, em 1954, com doze países fundadores: Bélgica, Dinamarca, França, Grécia, Itália, Noruega, Holanda, República Federal da Alemanha, Reino Unido, Suécia, Suíça e Iugoslávia (ANTHONY, 2014; CERN, 2012).

Localizado a noroeste de Genebra, na fronteira da França com a Suíça (Figura 6 Apêndice A (SUTTON, 2013)), o CERN é um dos maiores e mais respeitados centros de pesquisa científicas no mundo. Em 2016, o CERN englobava 22 países-membros, quase 12.000 cientistas de 70 países e 105 nacionalidades, mais de 2.500 funcionários além de 49 petabytes de dados gerados somente no ano de 2017 (PRALAVORIO, 2017a).

O objetivo primário do CERN é estudar o modelo padrão em um regime de maior energia e explorar duas questões fundamentais da física: a composição da matéria e as forças que a

mantém unida. Para alcançar esse objetivo, dezenas de estudos e experimentos são realizados nos laboratórios do CERN, com destaque para os que envolvem a Física de Altas Energias e fazem uso dos aceleradores de partículas lá construídos.

2.7 O Large Hadron Collider

O maior e mais poderoso acelerador de partículas do mundo, o *Large Hadron Collider* (LHC), a última aquisição do CERN, entrou em operação em 10 de setembro de 2008 (Fotografia 1 Apêndice A). É formado por um anel de 27 quilômetros de magnetos supercondutores, instalados à 100 metros de profundidade na fronteira franco-suíça, e conta com uma estrutura de aceleradores para aumentar a energia das partículas ao longo do caminho (CERN, 2014).

O complexo do LHC (Figuras 7 e 8 Apêndice A) possui seis detectores em pontos de colisão ao longo de sua circunferência, de forma que as colisões ocorridas possam ser reconstruídas. Tais detectores são o *A Toroidal LHC ApparatuS* (ATLAS), o *Compact Muon Solenoid* (CMS), o *A Large Ion Collider Experiment* (ALICE), o *Large Hadron Collider beauty* (LHCb), o *TOTal Elastic and diffractive cross section Measurement* (TOTEM) e o *Large Hadron Collider forward* (LHCf).

Os dois maiores experimentos de propósito geral, do LHC, são o ATLAS e o CMS, construídos para estudar a colisão entre prótons ($p - p$) e entre íons pesados $A - A$ (AAD *et al.*, 2008a; ARMSTRONG *et al.*, 1994; CHATRCHYAN *et al.*, 2008). O Bóson de Higgs foi detectado por eles no ano de 2012, cumprindo assim seu principal objetivo. Tal partícula elementar faz parte do Modelo Padrão e foi prevista por Peter Higgs ainda em 1964 (AAD *et al.*, 2012).

Os detectores ALICE e LHCb são considerados de médio porte e com propósitos mais específicos. O ALICE é especializado no estudo da formação do plasma de quarks e glúons (AAMODT *et al.*, 2008) e o LHCb é dedicado ao estudo de violação de Carga-Paridade (CP) e outros fenômenos raros, oriundos do decaimento do *méson-B* (ALVES JR. *et al.*, 2008).

Os experimentos TOTEM e LHCf são tidos como de pequeno porte e dedicados à física projetiva (*forward*) de prótons e íons pesados (ADRIANI *et al.*, 2006; BERARDI *et al.*, 2004).

Dentro do acelerador do LHC, dois feixes de prótons contendo 2.808 grupos (*bunches*) de 10^{11} partículas por grupo viajam em direções opostas em uma velocidade de 99,9998% da velocidade da luz, colidindo a uma taxa de 40MHz e gerando um fluxo de dados de 60TB/s (KLOUS, 2010) no sistema de aquisição do experimento ATLAS. Os dois tubos onde as colisões ocorrem são mantidos em ultra-vácuo (CERN, 2010) e um forte campo magnético guia os feixes por toda a extensão do colisor. Esses eletromagnets supercondutores são resfriados a $-271,3^{\circ}C$, temperatura menor que a do espaço exterior que é de $-270,5^{\circ}C$.

2.7.1 Cronologia das atualizações do Large Hadron Collider

A Figura 9 (Apêndice A) traça o desenvolvimento cronológico do LHC, desde a sua criação até a atualização para o *High Luminosity LHC* (HL-LHC). A evolução desse experimento

percorreu um período de 15 anos, entre 2011 e 2026 (HILUMI, 2020), que está descrito a seguir.

Em 2011, na *Run 1*, o LHC já trabalhava com uma luminosidade de $3 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$ e energia no centro de massa de 7TeV. No ano de 2012, passou a trabalhar com energia de 8TeV no centro de massa. O primeiro longo desligamento do LHC *First Long Shutdown* (LS1) ocorreu entre 2013 e 2014, para a atualização de alguns de seus componentes. No ano de 2015, voltou a operar com energia de 13TeV no centro de massa, dando início à *Run 2*. Em 2016, atingiu uma luminosidade de $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ e terminou o ano com $1,4 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ (PANDOLFI, 2016). A luminosidade alcançou o dobro do valor nominal planejado para essa fase, atingindo $2,06 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$, em 2017 (PRALAVORIO, 2017b).

Antes do início da *Run 3*, entre 2019 e 2021, foi dado andamento ao segundo grande desligamento, *Second Long Shutdown* (LS2), do LHC. Os experimentos ALICE, ATLAS e CMS passarão por atualizações que irão prepará-los para a retomada da operação em 2022, iniciando a *Run 3*. O LHC deverá operar com energia de 14TeV no centro de massa e uma luminosidade em torno de $2,5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ (HILUMI, 2020).

O início do terceiro longo desligamento do LHC, *Third Long Shutdown* (LS3), está programado para ocorrer entre 2024 e 2026. Servirá tanto para a realização de atualizações significativas nos experimentos ATLAS e CMS, quanto para aprontá-lo para o início da operação do HL-LHC, previsto para 2026. A energia no centro de massa deverá ficar em torno de 14TeV e a luminosidade projetada poderá alcançar de 5 até $7 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ (HILUMI, 2020).

2.7.2 Trem de *bunches*

Logo após a introdução do conceito de colisores, há mais de 40 anos, reconheceu-se que o efeito causado pelo encontro de feixes, devido à interação eletromagnética inerente aos *bunches* de partículas carregadas colidindo, pode limitar seriamente a luminosidade do colisor, conforme pode ser verificado em Shiltsev *et al.* (2008). A forma reduzida de uma derivação da Equação (2) (Equação 3) pode ser encontrada em Giovannozzi e Veken (2018a). Ela fornece o embasamento necessário e detalhado a respeito dos motivos e das finalidades que tornam a utilização de trens de *bunches* para a operação de colisores de partículas ser tão necessária. Nessa forma reduzida, a dependência da intensidade total dos feixes colidindo é destacada, enquanto as outras quantidades são incluídas no termo Ξ .

$$\mathcal{L} = \Xi N_1 N_2, \quad \Xi = \frac{\gamma f_0}{4\pi\epsilon\beta^* N_B} H \quad (3)$$

Uma das formas de possibilitar as colisões entre feixes de partículas pode ser encontrada no colisor Tevatron (Estados Unidos), onde feixes de prótons e antiprotons compartilham o mesmo tubo de feixe e abertura de ímã e, a fim de reduzir o número de interações frontais, os feixes são colocados em órbitas separadas em toda parte, exceto nos detectores *Collider Detector at Fermilab* (CDF) e *DØ experiment* (D0) IPs, onde são utilizados separadores eletrostáticos.

A interação de longo alcance entre feixes separados pode ser derivada da Equação 4, se

$r >> \sigma$. O efeito de escalas, como $(E, B) N_2/r$, além de ser não-linear, varia de grupo para grupo, o que o torna difícil de tratar (SHILTSEV *et al.*, 2008).

$$(E_r, B_\theta) \propto \frac{2eN_2}{r} \cdot \left(1 - \exp(-r^2/2\sigma^2)\right) \quad (4)$$

Na Run I do Tevatron, que decorreu entre agosto de 1992 e fevereiro de 1996, foram utilizados seis *bunches* em cada feixe, em órbitas separadas. Até 1992 foram instalados 22 separadores eletrostáticos destinados a atenuar as limitações dos encontros de feixes. Já para a Run II alcançou-se a operação de 36 *bunches*, em cada feixe (SHILTSEV, 2014).

No LHC, cada um dos dois feixes consiste em um trem de *bunches* (HERR, 1996). Estes *bunches* são espaçados por 25ns dando um número potencial de 3.564 *bunches* presentes ao mesmo tempo no anel de 27km do LHC. No entanto, o tempo de subida finito dos dispositivos de injeção e extração dos injetores requer pequenos intervalos sem *bunches* no trem (PETTERSSON; LEFÈVRE, 1995). Outra grande folga é necessária para o sistema de abortamento, resultando em um total de 2.835 *bunches* por feixe. O esquema de enchimento é mostrado na Figura 10 (Apêndice A).

Normalmente, um *bunch* em um feixe encontra sempre outro *bunch* em todas as interações de frente e de longo alcance. *Bunches* próximos dos intervalos encontrará os *bunches* correspondentes na colisão frontal, mas podem encontrar um intervalo nos pontos de colisão parasitária. Esses *bunches* têm, portanto, um esquema de colisão irregular com menos interações de longo alcance, onde os *bunches* próximos às lacunas são os mais irregulares. Eles se perdem primeiro e as lacunas são aumentadas, levando outros *bunches* a se tornarem irregulares. O efeito de perda de *bunches* nas lacunas é, geralmente, designado como efeito PACMAN, e os *bunches* chamados de *bunches* PACMAN.

Naturalmente, nem todas as interações de longo alcance são equivalentes quando as regiões de interação são diferentes (por exemplo, inserções β baixas e β altas). Os *bunches* PACMAN recebem sua "condição PACMAN" em uma região onde o efeito PACMAN de longo prazo é mais elevado do que o efeito PACMAN (por exemplo, alta inserção β) são menos afetadas do que outros *bunches* PACMAN que recebem suas colisões perdidas em uma região mais "destrutiva" (por exemplo, baixa inserção β).

A análise da modelagem sobre os dados obtidos na Run 1 do LHC desenvolvida por Giovannozzi e Veken (2018a) e aplicada em Giovannozzi e Veken (2018b) confirmou a imprecisão na reprodução dos dados do LHC, utilizando apenas os dados armazenados. Os modelos propostos mostraram um notável poder de reproduzir e descrever os comportamentos observados de luminosidade, em função do tempo e da luminosidade integrada, que embasam a busca por melhorias dos filtros utilizados.

Lotes oriundos do *Proton Synchrotron* (PS) de 48 *bunches* foram unidos para formar trens de 96 *bunches* (2 lotes PS) ou 144 *bunches* (3 lotes PS), para a operação de física de prótons de 2017 (HOSTETTLER *et al.*, 2018). O preenchimento do LHC no regime consistiu em 15 trens com 144 *bunches*, intercalados com 4 trens de 96 *bunches*. Acompanhado de 12

trens sem colisão "*bunches* de testemunhas", isto totalizou 2.556 *bunches* por feixe no LHC (Figura 11 Apêndice A).

O número de encontros de *bunches* de longo alcance, por feixe, no espaço de deriva em torno do ATLAS e do CMS, para este esquema de enchimento, é mostrado na Figura 12 (Apêndice A). O diferente número de encontros de feixes de longo alcance é esperado para produzir uma diferença em órbitas fechadas, devido aos efeitos do PACMAN descritos em Herr (1996).

Estudos contendo o tempo de duração e os tipos de *bunches*, além de outras informações, podem ser encontrados em Jowett *et al.* (2017), para o LHC e em Shiltsev *et al.* (2005), para o Tevatron. Uma análise já voltada para as atualizações previstas no HL-LHC é identificada em Fox *et al.* (2018).

2.7.3 Padrões de Colisão do *Large Hadron Collider*

O LHC vem testando vários padrões de *bunches* de colisões entre os feixes de prótons, desde o início de sua operação, (PETTERSSON; LEFÈVRE, 1995). Tais padrões consistem em uma quantidade de *slots* de tempo com colisões, seguidos de uma segunda quantidade de *slots* de tempo sem colisões.

A representação gráfica utilizada, nessa seção, para representar os padrões de colisão adotados no LHC é composta por dois tipos de gráficos. O primeiro gráfico contém uma linha em azul-claro, onde, no *slot* de tempo que ocorreu uma colisão, a linha é plotada no valor 1 e no *slot* de tempo onde não houve colisão a linha é plotada no valor 0. No segundo gráfico, plotado em seguida, a quantidade de *slots* de tempo onde ocorreram colisões é apresentada em vermelho, e a quantidade de *slots* de tempo onde não houveram colisões, em azul.

Alguns exemplos de padrões adotados a partir da Run 2 em 2015 podem ser visualizados nos Gráficos 1 e 2 (Apêndice A), onde se pode observar que, com apenas alguns dias de diferença, os padrões 72b39e (sendo 72 *slots* de tempo, *bunches*, com colisões e 39 sem colisões, *empty*) e 36b9e foram testados, nas *runs*⁴ 281411 e 283780, respectivamente.

Continuando os testes de padrões de colisões, em 2016 (Gráfico 3 Apêndice A), nota-se que, além do padrão principal na *run* amostrada, há um segundo padrão; neste caso, com um intervalo de tempo equivalente a 58 ou 59 *slots* sem colisões a cada 8 sequências do padrão principal, 72b37e.

Em 2017, o padrão 48b7e vinha sendo o mais testado (Gráfico 4 Apêndice A), até a ocorrência do problema que ficou conhecido com o nome de 16L2 (KAHLE; STEERENBERG; PANDOLFI, 2017). Tal nome se refere a um grupo, ou célula, composto por três dipolos, um quadripolo e alguns ímãs de correção, situados a 16 células à esquerda do ponto 2 do LHC. Na ocasião, a maioria dos depósitos de energia podia ser rastreado até essa célula, o que levantou a hipótese da presença de gás nos tubos de vácuo.

Carver *et al.* (2018) e Damerau *et al.* (2018) detalham mais a respeito desse problema (16L2) que levou o LHC a testar um padrão de colisões baseado em 8b4e. Significa que, de

⁴Neste contexto uma *run* é uma tomada de dados do LHC (CAMPLANI, 2019).

acordo com Wenninger e Pandolfi (2017), para cada trem de 8 *bunches* um trem de 4 *bunches* era inserido sem colisões. Esse padrão pode ser visualizado no Gráfico 5 (Apêndice A), que é uma pequena amostragem dos dados originalmente obtidos da *run* 335177, de forma que o gráfico gerado fosse compreensível.

Os feixes eram gerados de forma contínua, espaçados em 25ns , antes da adoção desse padrão de colisões. Esse novo padrão impactou no número de feixes, fazendo cair de 2.556 feixes, alcançados em julho, para algo em torno de 1.920 feixes. Além disso, tornou a operação mais suave, de forma que quase não se observou mais nenhum despejo associado com $16L2$, desde que os *bunches* gerados não possuíssem mais de $1,1 \times 10^{11}$ prótons. Nessa configuração, a redução do desempenho é aceitável podendo, inclusive, alcançar a meta ambiciosa de 45fb^{-1} de luminosidade⁵ integrada para 2017 (WENNINGER; PANDOLFI, 2017).

Com a resolução do problema associado ao $16L2$, ainda em 2017 (CARVER *et al.*, 2018; DAMERAU *et al.*, 2018), no ano seguinte o padrão $48b7e$ voltou a ser o mais adotado, conforme pode ser visto no Gráfico 6 (Apêndice A) referente a *run* 349977.

2.8 O EXPERIMENTO ATLAS

A colaboração ATLAS envolve 38 países, 174 institutos e conta com mais de 3.000 cientistas. O projeto do detector foi pensado para estudar a maior quantidade possível de fenômenos físicos, passíveis de serem gerados em colisões no LHC (COLLABORATION, 2018; EVANS; BRYANT, 2008), desde a busca pelo bóson de Higgs até dimensões extras e partículas que possam constituir a matéria escura.

O detector ATLAS possui uma geometria cilíndrica e cobre um ângulo sólido próximo a 4π . O ponto de colisão das partículas é rodeado por detectores de trajetórias (coletivamente referidos como detectores internos), localizados dentro de um solenoide supercondutor que provê um campo magnético axial de 2 Teslas⁶, seguido por um sistema de calorímetros e um espectrômetro de mûons (CERN, 2018). Na Figura 14 (Apêndice A) é mostrado o *layout* geral do detector ATLAS.

2.8.1 Sistema de coordenadas do ATLAS

O sistema de coordenadas do ATLAS é um sistema cartesiano que segue a regra da mão direita. A definição do sistema segue a seguinte forma (CERN, 1999; CERN, 2018):

- A origem é o ponto de interação O.
- O eixo-Z é o eixo do acelerador LHC.
- O eixo-X é horizontal, e aponta para o centro do anel do LHC.

⁵O femtobarn inverso (fb^{-1}) é uma medida de eventos de colisão de partículas por femtobarn; uma medida do número de colisões e da quantidade de dados coletados. Um femtobarn inverso corresponde a aproximadamente 100 trilhões (10^{12}) de colisões prótons-prótons. Durante um período de tempo, dois feixes de partículas com uma área transversal, medidas em femtobarns, são direcionadas para colidir. O número total de colisões é diretamente proporcional à luminosidade das colisões medidas ao longo deste tempo (CERN, 2019).

⁶Esse campo é utilizado para curvar a trajetória das partículas, de forma a poder medir a sua relação carga/massa.

- d. O eixo-Y é perpendicular ao eixo-X e ao eixo-Z.

Uma ilustração desse sistema de coordenadas pode ser visualizado na Figura 15 (Apêndice A).

O uso de coordenadas cilíndricas é comum em experimentos de física de partículas. As novas coordenadas podem ser obtidas a partir de transformações não lineares dos eixos x , y e z , da seguinte forma:

$$\phi = \arctan\left(\frac{x}{y}\right) \quad (5)$$

$$\theta = \arctan\left(\frac{x}{z}\right) \quad (6)$$

$$\eta = -\log\left(\arctan\left(\frac{\phi}{2}\right)\right) \quad (7)$$

A coordenada ϕ representa o ângulo de rotação e η , também conhecida como pseudorapidez, representa a direção de projeção das partículas, após a colisão.

2.8.2 Composição do detector ATLAS

O detector é composto por diversos subsistemas, cada um com características e funcionalidades diferentes, a saber:

- a. *Solenoid e Toroid Magnets* – sistema magnético desenvolvido para contribuir na medição do momento das partículas (Figura 13 Apêndice A) (AIRAPETIAN, 1999);
- b. *Inner Detector* – esse termo se refere ao conjunto composto por três subdetectores: um *Silicon Pixel Detector*, um *Silicon Microstrip Detector* e um *Transition Radiation Tracker*. A camada mais interna, recentemente instalada, de sensores de pixel já estava operacional pela primeira vez durante a retomada de dados na Run 2 em 2015. Esses detectores internos proveem medidas precisas de partículas rastreadas em um intervalo de pseudo-rapidez com $|\eta| < 2,5$. Também são organizados em uma forma geométrica coaxial ao redor do eixo do feixe (CERN, 2018), conforme pode ser visto na Figura 16 (a e b) (Apêndice A) (AAD *et al.*, 2008b);
- c. *Muon Spectrometer* – é composto por câmaras de rastreamento de alta precisão para medir a deflexão de múons em um campo magnético gerado por três grandes toroides supercondutores, que estão dispostos em uma bobina de simetria azimutal óctupla em torno dos calorímetros. As câmaras de alta precisão cobrem um intervalo de $|\eta| < 2,7$. O sistema de *trigger* de múons cobre o intervalo de $|\eta| < 2,4$ (AIRAPETIAN, 1999; CERN, 2018) (Figura 17 Apêndice A).
- d. *Calorímetros Eletromagnético e Hadronico* – são subdetectores desenvolvidos para medir a energia das partículas, que interagem de forma eletromagnética e hadrônica, respectivamente.

O Calorímetro Eletromagnético, ou LArg, é do tipo de amostragem⁷ e utiliza o Argônio Líquido como material amostrador (ativo) além de eletrodos de chumbo como material absorvedor (passivo). Os eletrodos estão posicionados na forma de um acordeon (Figura 18 Apêndice A) e mostra a energia da partícula através da ionização das células do calorímetro, sendo capaz de cobrir $|\eta| < 3,2$ (ALEKSA *et al.*, 2013).

O Calorímetro Hadrônico é também conhecido como Calorímetro de Telhas (*Tile Calorimeter* (TileCal)). É projetado para identificar e medir, principalmente, a energia dos jatos hadrônicos gerados pelas partículas resultantes das colisões. A forma com que essa leitura é realizada baseia-se em uma técnica de amostragem com telhas cintilantes, intercaladas com lâminas de ferro absorvedor e fibras ópticas capazes de diminuir a frequência dos fótons, para que possam ser lidos pela eletrônica de aquisição (Figura 19 Apêndice A). Este calorímetro será mais bem detalhado na próxima subseção.

2.9 O CALORÍMETRO HADRÔNICO DO ATLAS

O objetivo, aqui, é apresentar mais detalhes a respeito do Calorímetro Hadrônico (TileCal) do experimento ATLAS, citado na Subseção 2.8.2, detalhando, principalmente, o sistema de *Trigger and Data Acquisition* (TDAQ), foco deste trabalho.

Calorímetros devem fornecer boa contenção para chuveiros eletromagnéticos e hadrônicos e também devem limitar o impacto no sistema de mûon. Portanto, a profundidade do calorímetro é uma consideração de projeto importante.

Um exemplo das possibilidades de contenções presentes no ATLAS pode ser visto na Figura 20 (Apêndice A). Nela, constata-se o porquê da presença do TileCal englobando o LArg e como os hâdrons (prótons e neutrons) interagem com o TileCal.

O TileCal opera medindo a energia dos hâdrons, partículas constituídas de quarks e glúons – como os prótons, nêutrons, pions e kaons, por exemplo. O TileCal fornece, ainda, a medida indireta da presença de partículas que não interagem, como os neutrinos (DUARTE, 2015), através da energia faltante para a conservação do momento linear.

Em relação às suas proporções físicas, o TileCal possui um *Long Barrel* (LB) com 5,64 metros de comprimento, ocupando uma região entre $0 < |\eta| < 1,0$ de seção hadrônica, e dois *Extended Barrel* (EB) com 2,91 metros de comprimento cada, abrangendo uma região entre $0,8 < |\eta| < 1,7$ da seção hadrônica (Figura 21 Apêndice A).

Toda essa estrutura é montada numa forma cilíndrica com raio interno de 2,28m e externo de 4,23m (Figura 22 Apêndice A) (CARRIO *et al.*, 2013).

Os dispositivos cintiladores, em forma de telha, com aproximadamente 3mm de espessura, são posicionados perpendicularmente ao feixe de colisão e são separados por placas de aço de 4 a 5mm de espessura, formando, assim, uma estrutura de arranjo quase periódico de ferro cintilador (Figura 23 Apêndice A).

⁷Do inglês *Sampling Calorimeters*. Tais calorímetros medem parte da energia das partículas e inferem seu valor total após um processo de calibração.

Radialmente, o TileCal é segmentado em três camadas, conforme pode ser verificado na seção transversal exibida na Figura 21 (Apêndice A), cuja granularidade no sentido do ângulo η é mantida constante em 0, 1 (ABDALLAH; COLLABORATION, 2016). Quando uma partícula atravessa o material denso do TileCal, as subpartículas geradas interagem com os cintiladores e, então, a luz é coletada por fibras óticas nas duas extremidades da telha, provocando uma redundância na coleta de dados. Tais fibras óticas são estrategicamente agrupadas e acopladas a tubos fotomultiplicadores *Photo Multiplier Tube* (PMT), formando uma célula. Um exemplo desse agrupamento pode ser visualizado na Fotografia 2 (Apêndice A).

O sinal luminoso é convertido em sinal elétrico nos PMTs. O sinal elétrico é transmitido para um circuito de condicionamento e amplificação de sinal analógico, gerando um pulso estável de forma "fixa" com amplitude proporcional à energia depositada nas células (ANDRADE-FILHO, 2009). O pulso característico tem duração de 7 *Bunch-Crossing* (BC)s (Gráfico 7 Apêndice A) com distância BC de 25ns de duração, entre cada. A taxa de digitalização dos pulsos, sendo a taxa de colisão do LHC sincronizada com a taxa de digitalização, é de 40MHz (PERALVA *et al.*, 2013).

Como o principal objetivo do TileCal é contribuir para a reconstrução da energia dos jatos produzidos pelas interações próton-próton e auxiliar no cálculo do momento transverso faltante, o calorímetro deve contar com uma boa resolução em energia por toda sua dimensão radial, uma boa linearidade e uniformidade nas direções transversais e longitudinais. Para tal, a construção de cada barril é composta por 64 módulos, onde cada módulo realiza a cobertura de um ângulo de 5,6° no eixo azimutal. Em cada módulo é fixado um suporte onde ocorre toda a eletrônica de leitura.

Essa eletrônica de leitura inclui circuitos de *front-end* e digitalizadores de sinais – cerca de 10.000 canais – providos por ela. Os circuitos eletrônicos se localizam na parte externa do calorímetro, onde são agrupados em blocos de 48 pares de PMTs (ANDRADE-FILHO, 2009; DUARTE, 2015).

2.9.1 O sistema de seleção de eventos e aquisição de dados do ATLAS

O uso de sistemas de seleção de eventos (ou sistemas de filtragem) *on-line* é fundamental em ambientes que trabalham com altas taxas de eventos e onde se deseja separar o sinal de interesse dos demais sinais gerados pelo experimento (ou seja, ruído de fundo). Além disso, o sistema deve operar com restrições no tempo de processamento em ambientes que geram altas taxas de eventos. O processo de filtragem pode se tornar ainda mais complexo em casos específicos como, por exemplo, nos colisões de partículas, onde os eventos gerados podem requerer uma quantidade elevada de memória, sendo necessária uma discriminação *on-line* de alta velocidade (BARBOSA, 2017).

O sistema de aquisição de dados do ATLAS operou com sucesso durante o período entre 2009 e início de 2013, conhecido como *Run 1*. Selecionando eventos com alta eficiência energética por meio da energia de centro de massa (\sqrt{s}) de até 8TeV, satisfez, assim, uma grande

gama de processos físicos, incluindo a busca de partículas em escala de TeV (CZODROWSKI, 2015).

Na *Run 2* houve uma expansão aproximada de duas vezes na energia de colisão, atingindo 13TeV. A luminosidade também sofreu um fator adicional de 2,5 a 3, passando de $8 \times 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$ para algo entre $1 \text{ e } 2 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$. O número de interações por BC (μ_{peak}) passou de 40 na *Run 1* para cerca de 50 na *Run 2*. Além disso, pela primeira vez o *bunch spacing* passou de 50ns para 25ns (CZODROWSKI, 2015; NAKAHAMA, 2015). No entanto, a luminosidade atingiu $2,06 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$, ainda em 2017 (PRALAVORIO, 2017a), número superior ao dobro do projetado para a fase, aumentando as possibilidades de empilhamentos de sinais, pela maior taxa de ocupação do canal causada pelo aumento da luminosidade.

O sistema de seleção de eventos e aquisição de dados do detector ATLAS, (TDAQ) atual, ilustrado no Diagrama 1 (Apêndice A), faz uso de um sistema de *trigger* de multicamadas com o intuito de reduzir a taxa de eventos de 40MHz para algo em torno de 1,5kHz possibilitando, assim, que a informação completa dos eventos resultantes possam ser escritos em disco (ABBOTT *et al.*, 2015; NAKAHAMA, 2015).

A cada 25ns, cerca de 1,5MB são capturados à taxa de 40MHz, gerando um fluxo de dados de 60TB/s. Reduzindo-se para até 1,5kHz, são salvos em disco cerca de 1,5GB/s, o que diminui cerca de 4×10^{-4} . Essa taxa é dominada pela parte inelástica da seção transversal total, que é de aproximadamente 70MB. As seções transversais de muitos dos processos interessantes da física estão em um fator de 10^6 abaixo desse valor (ABOLINS, 2008).

2.9.1.1 O primeiro nível de trigger do TileCal

O primeiro nível de *trigger* (*Level-1* L1) é implementado em *hardware* e utiliza toda uma eletrônica customizada e executada em FPGAs, que seleciona previamente os eventos. Este nível reduz o fluxo de eventos de 40MHz para 100kHz com uma latência máxima permitida de $2,5\mu\text{s}$. Nestes eventos são selecionadas as *Region of Interest* (RoI) (ABBOTT *et al.*, 2015; CZODROWSKI, 2015). O Diagrama 2 (Apêndice A) apresenta maiores detalhes do L1.

Os eventos selecionados pelo primeiro nível são processados pelo *High Level Trigger* (HLT), que provê acesso a informações com resolução total do detector. Neste nível, um *hardware* de *Fast Track finder* (FTK) provê dados para o HLT nas taxas do L1.

No segundo nível de *trigger* (*Level-2* L2), o HLT é composto por um *cluster* de computadores que rodam algoritmos mais complexos que no L1 com resolução total. Os algoritmos são processados com um tempo médio por evento na ordem de 0,2s. Nesse HLT, a taxa do fluxo de dados vindos do L1 passa dos 100kHz para algo em torno de 1,5kHz, portanto, os dados para processamento *offline* já gravados em disco a uma taxa próxima de 1,5Gb/s (ABBOTT *et al.*, 2015; ABOLINS, 2008; CZODROWSKI, 2015).

A associação dos pulsos identificados em sua torre de *trigger* (TT) com o BC correspondente (do inglês *Bunch-Crossing IDentification* (BCID)) é um dos requisitos cruciais do L1 do TileCal. Trata-se de um passo delicado dado que, na TT, o sinal se estende por vários

BCs. O ideal é que o BCID associe um único valor de energia digitalizado para um amplo pulso identificado na TT e o envie para a central de processamento de *trigger*, em um intervalo de tempo correspondente ao pico do pulso (ASMAN *et al.*, 2012). Conforme apresentado no Diagrama 3 (Apêndice A), a seguir, o BCID fica instalado no nível 1 de *trigger*, mais especificamente no módulo de preprocessamento que está ligado à *Read Out Driver* (ROD) (ASMAN *et al.*, 2012).

2.9.1.2 Cronologia das atualizações do nível 1 de trigger do TileCal

Durante a *Run 1* do LHC – entre 2011 e 2012 – o Filtro Casado foi utilizado no *Level 1* do sistema de *trigger* do TileCal (PALKA, 2016), além de implementada a FPGA Xilinx XC2V1500 (GREEN *et al.*, 2003), em substituição ao *Optimal Filter* (OF) (FULLANA *et al.*, 2005) aplicado até então.

Entre as *Runs 1 e 2* do LHC – de 2013 a 2014 – ocorreu o primeiro grande desligamento do LHC LS1, período que ficou conhecido como Fase 0 de atualização do ATLAS. Foram implementados os Filtros FIR autocorrelacionados (PALKA, 2016) nas FPGAs Virtex 7 690T e Spartan-6 (ANDREI, 2012) ambas da Xilinx, no *Level 1* do sistema de *trigger* do TileCal.

A Fase I de atualização dos experimentos ATLAS e CMS ocorreu entre 2019 e 2021, quando o acelerador LHC foi novamente desligado durante o segundo grande desligamento LS2. Porém, neste período, não houve grandes atualizações como as que ocorreram na Fase 0. A maior parte das atualizações deve ficar para o LS3, que deve começar em 2024, junto com a Fase II de atualizações dos experimentos ATLAS e CMS.

Para a Fase II de atualização dos experimentos ATLAS e CMS existem várias propostas para a utilização de filtros baseados em FIR (DUARTE *et al.*, 2016), que devem ser implementados utilizando a FPGA Xilinx Kintex UltraScale+ XCKU115-1FLVB1760C (ARGOS, 2017) no *Level 1* do sistema de *trigger* do TileCal. O algoritmo desenvolvido neste trabalho deve ser executado nessa FPGA.

Um resumo dos principais pontos dessa cronologia é apresentado no Quadro 3, onde há uma correlação com a cronologia de atualizações do próprio LHC.

2.9.2 A eletrônica do *Trigger and Data Acquisition* do TileCal

A eletrônica do *Trigger and Data Acquisition* do TileCal é dividida em duas partes: a primeira é composta pela eletrônica de *Front-End* (FE) e de digitalização, situadas em módulos ainda dentro do detector ATLAS, nas chamadas gavetas (*drawers*). Tais gavetas são removíveis e podem ser inseridas em um suporte, facilitando sua manutenção (BIOT, 2009). A segunda parte, chamada de eletrônica de *Back-End* (BE), tem como principal componente o ROD, responsável por preprocessar e reunir os dados vindos da eletrônica de FE, quando um evento específico é selecionado pelo primeiro nível de *trigger*. Assim sendo, este processamento é feito a uma taxa média de 100kHz, enviando esses dados para os *Read Out Buffers* (ROB), no segundo nível do sistema de *trigger* (CERQUEIRA, 2012).

Quadro 3 - Resumo da evolução do ATLAS e do *Large Hadron Collider*

Atualização	LHC			HL-LHC
Runs	Run 1	Run 2	Run 3	Run 4 em diante
$\mathcal{L} cm^{-2}s^{-1}$)	3×10^{33}	$2,06 \times 10^{34}$	$2,5 \times 10^{34}$	de 5 até 7×10^{34}
TileCal LVL1 FPGA(Xilinx)	XC2V1500	Spartan-6 e Virtex 7 690T		Kintex UltraScale+ XCKU115
Filtro	Casado	FIR autocorrelacionado		Em estudo

Nota(s): As informações contidas neste quadro se basearam em dados compilados de Andrei (2012), Argos (2017), Hilumi (2020) e Palka (2016).

Fonte: Elaborado pelo autor (2021).

O Diagrama 4 (Apêndice A) mostra a atual cadeia de sinais do TileCal. Nela, a luz produzida pela energia depositada no detector é coletada por fibras óticas e enviadas para um *mixer*, onde várias fibras são agrupadas formando uma célula do detector (CERQUEIRA, 2015).

Os pulsos analógicos dos PMT são modelados (circuito de *shaper*), divididos em dois ramos (ganho alto e baixo, com uma taxa de ganho de 64:1, respectivamente) e amostrados a cada 25ns, com conversores analógico-digital de 10 bits (HARKUSHA; COLLABORATION, 2016) (*Analog-to-Digital Converter* (ADC)). As amostras digitais são armazenadas em memórias de *pipeline*, durante a latência do *trigger* L1 ($2,5\mu s$), no esquema de processamento de sinal do TileCal atual (Diagrama 5 Apêndice A).

Os sinais analógicos do PMT são somados e transmitidos para o L1 do calorímetro, simultaneamente, formando as torres de *trigger*. As amostras digitais dos eventos selecionados pelo *trigger* L1 são transmitidas para os ROD, localizados no sistema de *back-end*, a uma taxa média máxima de $100 kHz$. Os valores de energia, localização no tempo (BCID) e qualidade da medida são reconstruídos nas RODs, para cada canal, e são usados na cadeia de decisões do HLT, bem como no processamento de dados *off-line* (MLYNARIKOVA, 2017; SOLODKOV, 2017).

2.9.3 A eletrônica do experimento ATLAS na Fase II de *upgrade*

O *upgrade* do ATLAS para a Fase II que entrará em operação a partir de 2024, acomodará o sistema de detecção e aquisição de dados para o HL-LHC. Na arquitetura de leitura do TileCal, as atuais gavetas serão divididas em quatro minigavetas (*minidrawers*), operacionalmente independentes entre si. Isso reduzirá o número de conexões e falhas na transmissão de energia e dos dados. Cada minigaveta conterá 12 PMTs, cada um com sua *Front-End Board* (FEB), uma *Main Board* (MB), *Daughter Board* (DB) e *High Voltage board* (MLYNARIKOVA, 2017; SOLODKOV, 2017).

Em relação às FEBs, três diferentes opções estão sendo desenvolvidas e atualmente em teste (Figura 24 Apêndice A):

- a. a opção 3-em-1 é uma versão melhorada da atual placa de FE, é baseada em componentes discretos e provê uma melhor linearidade com menos ruído. Os sinais de 12 PMTs são coletados a uma frequência de amostragem de 40MHz pela MB, que acomoda 24 ADCs de 12 bits, e quatro FPGAs de controle (TANG *et al.*, 2015). O ruído eletrônico é menos de 1 ADC em ambos os ganhos. Os dados são transferidos a uma DB em comum;
- b. a *Charge Integrator and Encoder* (QIE) é um *Applications Specific Integrated Circuit* (ASIC) customizado, que inclui um divisor de corrente em quatro ganhos seguidos por um integrador fechado e um ADC *on-board* de 7 bits. Ele não realiza modelagem de pulsos e, portanto, minimiza os problemas de *pile-up* e permite medir os pulsos integrais do PMT. Ainda, provê digitalização em ponto flutuante da carga de entrada a cada 25ns (40MHz). O ASIC também inclui um *Time Digital Conversor* (TDC) de 5 bits de limite fixo e sua versão QIE12 é adaptada especificamente para o TileCal. Tem cerca de 18 bits de faixa dinâmica para medições de carga e ruído eletrônico de, aproximadamente, 1,6 contagens ADC no maior ganho. Esse ASIC fornece informações digitais, o que significa que a MB correspondente é responsável apenas por coletar e rotear os sinais para a DB em comum (SOLODKOV, 2017);
- c. o *Front-end for ATLAS TileCal Integrated Circuit* (FATALIC) também é baseado em um chip ASIC, mas, ao contrário da opção QIE, o desenvolvimento do FATALIC foi iniciado pela comunidade TileCal a partir do zero. Ele inclui o transportador de corrente seguido por um integrador de corrente com três ganhos diferentes ($1\times$, $8\times$, $64\times$). Os sinais do integrador são digitalizados com ADCs de 12 bits, dispostos em *pipeline* e inclusos no ASIC. O nível de ruído esperado na versão 5 (FATALIC5) é de cerca de 2,5 contagens ADC no maior ganho. Os dados digitalizados são coletados no MB, que transfere os dados para a DB em comum subsequentemente (MLYNARIKOVA, 2017).

A DB é a interface de comunicação *on-detector* entre o FE do TileCal e a sala de controle. O *off-detector Tile PreProcessors* (TilePPr) envia *Detector Control System* (DCS) e dados de controle de *run* via *links* ópticos de 4,8Gbps para as DBs. Estas, por sua vez, enviam dados de leitura e DCS para os TilePPRs via *links*, em turnos, rodando a 9,6Gbps. As atuais DB são baseadas na FPGA Xilinx Kintex-7. Os *links* ópticos são implementados utilizando o modulador baseado no Molex QSFP+ e tem uma medição de *Bit Error Rate* por volta de 10^{-18} (SOLODKOV, 2017).

O esquema da nova cadeia eletrônica é exibido no Diagrama 6 (Apêndice A), onde a FEB e a MB juntas proverão o condicionamento, amplificação e digitalização do sinal analógico vindo do PMT. A MB entregará dados digitalizados para a DB e também será responsável pelo controle e monitoramento das FEBs e pela distribuição de baixa voltagem na minigaveta. A principal diferença entre este sistema e o sistema antigo, cuja aplicação é de suma importância para a tese em questão, é que todos os dados gerados no detector e não apenas os selecionados pelo L1 (100kHz) serão transferidos para o TilePPr na taxa nominal do LHC (40MHz).

A Figura 25 (Apêndice A), representa a cobertura dos ângulos $\eta \times \phi$ das antigas ROD e o

novo esquema para a Fase-II, o TilePPr. Na mesma Figura, as RODs foram mapeadas pelo ponto de vista das RoI, enquanto que o mapeamento dos TilePPr foram otimizados pela perspectiva do *trigger* (COLLABORATION, 2017a).

Todos os dados gerados no detector, para cada BC, serão transferidos para o (TilePPr), fora do detector (ARGOS; VALERO, 2015) e fornecerão informações do chamado *trigger Level-0* do ATLAS. O TilePPr representará, assim, a interface entre os sistemas de aquisição, disparo e controle de dados mais a eletrônica do detector. Os dados recebidos pelo TilePPr, vindos de cada BC, são utilizados para reconstruir a energia depositada em cada uma das 10.000 células do TileCal. Anteriormente, isso era feito pelas RODs a 100kHz, somente com os dados recortados pelo L1, fornecendo assim as informações dos BCID direto do TilePPr (VALERO; COLLABORATION, 2017). O novo algoritmo de reconstrução de energia deverá operar, portanto, de forma interrupta, para todos os canais do TileCal, simultaneamente.

A principal proposta deste trabalho é desenvolver métodos de reconstrução *free-running*, a serem implementados no novo TilePPr, baseados na teoria da esparsidade, por serem mais eficientes que os atuais que utilizam Filtros FIR. A implementação proposta será instalada nas FPGAs modelo XCKU115-1FLVB1760C (ARGOS, 2017), presentes nos módulos TilePPr da Figura 6 (Apêndice A) e mais bem detalhado no Diagrama 7 (Apêndice A) (COLLABORATION, 2017a). No Diagrama 7 (Apêndice A) é apresentado o esquemático de uma placa-mãe do tipo *Advance Telecommunications Computing Architecture* (ATCA) contendo quatro Compact-Processing Modules (CPM)s, mais uma interface TDAQ implementada como uma *ATCA Rear Transition Module* (RTM) (SOLODKOV, 2017).

Antes de apresentar os detalhes do projeto e sua implementação, a próxima seção traz mais informações a respeito dos métodos atualmente usados e dos problemas relacionados à sua operação na Fase II de *upgrade* do LHC.

2.10 CONCLUSÃO DA SEÇÃO 2

Uma vez apresentada a contextualização histórica, ambiental e até mesmo eletrônica deste trabalho, o desenvolvimento teórico/matemático dos métodos em uso e dos já propostos por nosso grupo no ATLAS será apresentado a seguir, como é o caso do método Filtro FIR, autocorrelacionado e citado na Subsubseção 2.9.1.2.

3 REVISÃO DOS MÉTODOS DE RECONSTRUÇÃO DE ENERGIA

Esta seção apresenta uma revisão dos métodos de estimação de energia de partículas no primeiro nível de *trigger*, atualmente em uso, e dos métodos que vêm sendo propostos nos últimos anos para a operação na Fase II de atualizações do TileCal. Tais métodos servem de base para os estudos realizados e descritos mais à frente, permitindo a avaliação de todos os métodos estudados neste trabalho.

A possibilidade de ocorrências de *pile-up* de sinais, oriundos da digitalização promovida pela eletrônica de FE, será cada vez mais frequente com o *upgrade* do TileCal projetado para entrar em operação na Fase II de atualizações. Um exemplo de *pile-up* entre dois sinais pode ser visualizado no Gráfico 8 (Apêndice B), onde uma colisão sensibilizou uma determinada célula do TileCal gerando um sinal e, 50ns depois, ocorre uma segunda colisão que sensibilizou a mesma célula. Como estes eventos ocorreram em um período inferior ao necessário para identificar um pulso conformado (150ns), forma-se um sinal resultante, que é o efeito dos outros dois sinais empilhados. Como a eletrônica de *front-end* também será atualizada na Fase II, abre-se a possibilidade de se implementar, em hardware, métodos mais robustos contra esse efeito. Com isso, este trabalho focou na busca por alternativas viáveis dentro do novo cenário que está sendo construído para a Fase II.

Uma das características dos BC do LHC é que são gerados sempre seguindo algum tipo de padrão de colisão (ARDUINI *et al.*, 2007) e que podem ser considerados como janelas de *bunches* cheias, seguidas de uma pequena sequência de *bunches* vazios. O que se buscou, nesta pesquisa, foram métodos que pudessem ser implementados levando em conta essa característica específica. Assim, o objetivo principal deste trabalho é a recuperação de um sinal impulsivo em cada *bunch* e, para tal, na próxima subseção, será analisado o método atualmente em uso. Mais à frente, serão tratadas algumas alternativas já propostas.

3.1 ESTRATÉGIAS ENVOLVENDO FILTROS FIR

Daniells (2016) diz que a seleção correta dos coeficientes de um Filtro FIR é crucial para otimizar a eficiência da detecção do BCID da torre de *trigger* e a resolução da energia transversa (E_T). Neste trabalho, três estratégias diferentes de filtros FIR são consideradas como de fácil implementação no *new Multi-Chip Module* (nMCM). Dentre essas três estratégias de filtros FIR, citadas por Daniells (2016), a mais simples de todas é um Filtro FIR *trivial*, já que usa apenas um único multiplicador para passar efetivamente por todos os dados do ADC, inalterando-os. Isto significa que apenas os acréscimos *in-time* estão incluídos, ajudando a minimizar os acréscimos de *pile-up*. No entanto, o filtro se torna suscetível aos ruídos brancos incoerentes que, de outra forma, cancelariam por aproximação muitas amostras independentes.

A segunda abordagem mais simples, ainda segundo Daniells (2016), é o Filtro Casado adotada durante a *Run 1*. Nessa abordagem os coeficientes são escolhidos para serem proporcionais à amplitude do pulso do calorímetro em cada amostra. É a configuração ótima dos coeficientes

de filtragem para ruído branco (GARVEY; REES, 1996). Porém, para se garantir a efetividade dessa solução, é necessário que seja anexado um circuito de detecção de picos na saída do Filtro Casado, para que seja possível encontrar o BC correto onde houve deposição de energia.

Por fim, e igualmente segundo Daniells (2016), os filtros de autocorrelação aproveitam a correlação cruzada de um sinal feita entre eles em diferentes amostras e são eficazes na identificação da presença de sinais periódicos, que são obscurecidos pelo ruído. A estrutura da correlação dependerá da forma do pulso e do espaçamento do *bunch*, enquanto a força da correlação dependerá das condições de *pile-up*. É possível, portanto, produzir coeficientes de autocorrelação otimizados para um conjunto específico dessas condições.

Uma forma de se calcular a matriz de autocorrelação do ruído dos coeficientes de correlação linear é utilizando o método de correlação de Pearson da amostra (CHATTERJEE; HADI, 2013), descrito a seguir:

$$R_{ij} = \frac{n \sum S_i S_j - \sum S_i \sum S_j}{\sqrt{\left(n \sum S_i^2 - (\sum S_i)^2\right) \left(n \sum S_j^2 - (\sum S_j)^2\right)}} \quad (8)$$

onde: S_i e S_j são valores de ADC de amostras de dados de posições particulares na janela de amostragem e n é o número de janelas de amostragem consideradas. A matriz é determinada por seleção randômica, independente das janelas de dados de ADC, centradas em um BC preenchido, para uma simulação incluindo apenas ruído térmico e de *pile-up*.

O conjunto de coeficientes de autocorrelação \mathbf{a} é extraído através da multiplicação da inversa da matriz de correlação do ruído pela forma normalizada do pulso \mathbf{g} (CLELAND; STERN, 1994), como se vê em:

$$\mathbf{a} = \mathbf{R}^{-1} \cdot \mathbf{g} \quad (9)$$

Os elementos diagonais mostram a correlação perfeita entre uma amostra e ela mesma, enquanto a correlação entre diferentes elementos externos à diagonal se torna mais fraca na medida em que são separados. Eventualmente, mostram a anticorrelação como o resultado da entrada de um pulso fora dessa diagonal. A força da correlação, para amostras separadas pela mesma distância, é muito semelhante.

As contribuições das somas do Filtro FIR, em ambos os lados da amostra central, são pesadas negativamente, de forma irrefutável, pelos coeficientes. O filtro, quando comparado a outros filtros, é menos dependente do nível do pedestal, devido à menor soma de coeficientes, sendo uma vantagem significativa em áreas que experimentam grandes deslocamentos de pedestal (DANIELLS, 2016). A *performance* desses três tipos de filtros é demonstrada no Gráfico 9 (Apêndice B), para a torre de *trigger* FCal1. O Filtro de Autocorrelação demonstrou significante melhoria na eficiência de detecção de BCID em ambiente de alto *pile-up* e alcançou a sua eficiência em aproximadamente 12 GeV, quando relacionado a outros filtros. O Filtro FIR *trivial* apresentou melhor *performance* que o Filtro Casado utilizado durante a *Run 1*. Isso se

deve à maior significância do ruído térmico nessa região do calorímetro, quando comparado ao ruído de *pile-up* e, portanto, ao Filtro Casado. O Filtro Casado, por sua vez, é otimizado para ruído branco não possuindo, assim, uma boa *performance*.

A resposta dada pelos diferentes filtros é dominada pela sua sensibilidade ao deslocamento do pedestal e é relacionada à soma dos coeficientes utilizados. O Filtro de Autocorrelação tem, como resultado do maior coeficiente negativo usado, a maior independência do pedestal. Finalmente, o Filtro de Autocorrelação também tem a melhor resolução. Diante do mencionado, o Filtro de Autocorrelação prova ter a melhor *performance* em cada aspecto, oferecendo um aperfeiçoamento expressivo sobre a *performance* da *Run 1* em um ambiente dominado pelo ruído de *pile-up* (DANIELLS, 2016).

3.2 CONVOLUÇÃO E DECONVOLUÇÃO

A mudança da forma original do pulso característico do sinal capturado em determinado canal é efeito do empilhamento de sinais, conforme demonstrado no Gráfico 8 (Apêndice B). Com isso, a observação do sinal resultante de um determinado BC fica comprometida. A mudança das características do sinal, portanto, pode afetar a eficiência da detecção empregada atualmente, à medida que as atualizações da Fase II avançam, uma vez que o formato do sinal $h[n]$ ficará cada vez menos evidenciado. O trabalho desenvolvido por Duarte (2015) apresenta uma solução moldada pela teoria da convolução, buscando obter uma resposta para o problema *pile-up*. Segundo Bracewell (1999), a convolução pode ser definida como um operador linear que, ao ser aplicado em duas dadas funções, mensura a área subentendida do produto destas funções em relação ao deslocamento existente entre elas.

No contexto do processamento digital de sinais em sistemas lineares invariados (*Linear Time-Invariant – LTI*) no tempo discreto, no qual os trabalhos de Duarte (2015) e este trabalho se desenvolveram, a convolução pode ser modelada como uma relação entre a entrada e a saída do sistema. Sua caracterização pode ser definida pela resposta dada ao impulso, de acordo com as consequências das propriedades de um sistema LTI. É possível, portanto, calcular a saída de um sistema para qualquer entrada conhecendo a resposta ao impulso (MITRA, 2010), de tal modo que essa relação possa se desenvolver considerando $h[n]$ como a resposta ao impulso de um sistema LTI no tempo discreto, obtida da aplicação da entrada $\delta[n]$, sendo $\delta[n]$ a representação da distribuição delta de Dirac (RIFFE, 2018) e, portanto, possuindo as seguintes propriedades:

- a. $\delta[n] = 0$ para todo $n \neq 0$;
- b. $\sum_{i=-\infty}^{\infty} \delta[i] = 1$;

Como o sistema é LTI no tempo discreto, ao se calcular a resposta ao impulso de $h[n]$ para a entrada $x[n]$, a resposta para $\delta[n - 1]$ é igual a $h[n - 1]$. Sendo assim, as respostas a $\delta[n + 1], \delta[n - 2], \delta[n - 3], \dots, \delta[n - k]$ são, respectivamente, $h[n + 1], h[n - 2], h[n - 3], \dots, h[n - k]$. A título de exemplo, e de acordo com Duarte (2015), a resposta do sistema linear a uma entrada $x[n] = b_0\delta[n + 2] + b_1\delta[n + 1] + \dots + b_{k+1}\delta[n - k + 1] + b_k\delta[n - k]$ se

resume a:

$$r[n] = b_0 h[n+2] + b_1 h[n+1] + \dots + b_{k+1} h[n-k+1] + b_k h[n-k] \quad (10)$$

Para uma entrada aleatória $x[n]$ o resultado obtido pode ser expresso como uma combinação linear dos pesos b_k , que são atrasados por sequências de amostras unitárias, conforme pode ser visualizado na Equação 11:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n-k] \quad (11)$$

onde os coeficientes $x[k]$ denotam especificamente o valor da k -ésima amostra da sequência $x[n]$. A resposta do sistema LTI no tempo discreto para a sequência $x[k] \delta[n-k]$ é $x[k] h[n-k]$. Assim, a resposta $y[n]$ do sistema LTI no tempo discreto para $x[n]$ é dada por:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] \quad (12)$$

Logo, a Equação 12 pode ser reescrita conforme a Equação 13:

$$y[n] = \sum_{k=-\infty}^{\infty} x[n-k] h[k] \quad (13)$$

Mitra (2010) chama as Equações 12 e 13 de somatório de convolução das sequências $x[n]$ e $h[n]$ e podem ser representadas na forma resumida como na Equação 14, onde \circledast representa o somatório de convolução.

$$y[n] = x[n] \circledast h[n] \quad (14)$$

O TileCal foi modelado como um sistema LTI no tempo discreto tanto neste trabalho quanto no de Duarte (2015). Sua resposta ao impulso será considerada então por $h[n]$, que englobará toda a cadeia de processamento interno do TileCal. Dentro dessa abordagem, $h[n]$ torna-se o próprio pulso característico do TileCal e $x[n]$ passa a ser interpretado, convenientemente, como o valor da energia depositada em cada BC a ser determinado. Essa interpretação é possibilitada em função de a taxa de colisão do LHC e o *clock* de digitalização dos pulsos serem de mesma frequência ($40MHz$) e síncronos, de modo que a cada amostra digitalizada do *front-end* possa ser associado um BC específico. A Figura 26 (Apêndice B) representa esta abordagem de forma gráfica, onde $w[n]$ é modelado como um ruído eletrônico de fundo que adicionado ao vetor $y[n]$ forma o vetor $r[n]$.

Desta forma, a base do problema a ser solucionado por este trabalho é alterada para recuperar $x[n]$, a partir do conhecimento prévio de $h[n]$ e $r[n]$. Com isso, o modelo dado pela Figura 26 (Apêndice B) faz com que a questão de estimar a energia por BC, para cada uma das

células do TileCal, se resuma a implementar um processo *on-line* de deconvolução.

3.3 DECONVOLUÇÃO ON-LINE POR FILTRO FIR

Segundo Duarte, Andrade Filho e Seixas (2017), existem propostas que modelam toda a cadeia eletrônica do calorímetro como sendo um canal de comunicação. Seguindo essa linha, pode-se considerar a estimativa de energia como o resultado da equalização (ou deconvolução) desse canal (ANDRADE-FILHO *et al.*, 2015). A partir dessa visão, pode-se considerar as técnicas de deconvolução propostas em Duarte *et al.* (2016). Assim, são obtidas as bases que viabilizam a realização de um projeto de Filtro FIR através de uma abordagem estocástica baseada na teoria de filtragem adaptativa, ao invés da utilização de uma abordagem determinística.

A natureza estocástica existente na obtenção da resposta ao impulso do calorímetro fornece o suporte necessário para a utilização dessa abordagem. O TileCal possui um circuito de *shaper* que permite uma boa aproximação determinística para o sinal obtido por ele. Mesmo assim, esse sinal obtido possui algumas estocasticidades de difícil modelagem, tais como: deformações do pulso característico com relação à amplitude, saturação, amostragem no pico (desvios de fase), adição de ruído coerente do detector, entre outros.

Para a aplicação dessa técnica de abordagem estocástica é necessário apenas o conhecimento prévio de um conjunto de desenvolvimento, no qual devem ser conhecidas amostras de entrada e de saída do sistema. Utilizando-se de simulações de Monte Carlo do detector, se torna possível obter estes conjuntos de amostras, tornando viável e interessante a utilização dessa técnica na estimativa da energia por BC.

3.3.1 Filtragem adaptativa e modelagem inversa

Como uma tomada de dados no LHC pode levar dezenas de horas, a luminosidade tende a decrescer com o tempo (STEERENBERG, 2018), chegando à metade do seu valor no final do período de aquisição. Considerando que os filtros adaptativos são interessantes para aplicações em sistemas variantes no tempo, cogitar uma abordagem adaptativa para filtros de deconvolução, que permita obter uma relação de dependência entre os seus coeficientes e a luminosidade instantânea, é de grande interesse investigativo.

Segundo Haykin (2013), um filtro adaptativo possui uma adequada habilidade de operar em um ambiente desconhecido e acompanhar as variações estatísticas no tempo, tornando-os dispositivos poderosos para aplicações em processamento de sinais e aplicações de controle. A aplicação de filtros adaptativos é vasta e está inserida em diversas áreas (por exemplo: comunicações, radares, sonares e sismologia) (HAYKIN, 2013). Tais áreas possuem ao menos uma característica em comum, ou seja, um sinal de entrada e uma resposta desejada que é utilizada para controlar os valores de uma configuração de coeficientes ajustáveis do filtro adaptativo além de ser empregada para estimar um erro.

Esses coeficientes ajustáveis podem ser utilizados para promover o ajuste de pesos, refletir coeficientes, realizar a rotação de parâmetros, etc. Porém, a diferença essencial entre as

várias aplicações de filtros adaptativos (como nos sistemas de identificação, modelagem inversa, predição e cancelamento de interferências) nasce da maneira pela qual é extraída a resposta desejada (DUARTE, 2015). Neste estudo, a modelagem inversa é a que corresponde ao tema de deconvolução em questão.

3.3.1.1 Modelagem inversa

A função da aplicação de filtros adaptativos é a produção de um modelo inverso de um sistema desconhecido, neste contexto. O ideal, no caso de um sistema linear, é que o modelo inverso tenha uma função de transferência equivalente à função de transferência do sistema inverso da planta desconhecida. Na prática, porém, e de forma geral, é desenvolvido um Filtro FIR que aproxima a resposta do sistema inverso, fundamentando-se em algum critério de otimização. Tal critério costuma ser a minimização do erro quadrático médio (RMS) entre o valor desejado e a saída do filtro (KAY, 1993).

Na modelagem inversa, é possível se projetar um Filtro FIR que se aproxima da deconvolução de $h[n]$ a partir do conhecimento prévio de uma entrada $x[n]$ do sistema – aqui gerada por um *Toy Simulator* (a ser mais bem detalhado mais adiante) – e do sinal observado na saída do sistema $r[n]$. O Diagrama 9 (Apêndice B) ilustra essa modelagem. Nota-se que o erro entre $x[n]$ atrasado e a saída do filtro de deconvolução é usado no projeto dos coeficientes deste último.

Os filtros adaptativos são, normalmente, voltados para aplicações em tempo real. Esse modelo em tempo real, porém, gera a desvantagem de requerimento excessivo na elaboração e no custo do *hardware* onde o mesmo é projetado (HAYKIN, 2013). O esquema apresentado na Figura 9 (Apêndice B) exemplifica um processo de filtragem adaptativa em tempo real onde, em cada obtenção de uma nova amostra, os pesos do filtro são atualizados (DUARTE, 2015).

Esse processo, no entanto, pode ser implementado em janelas, de tal forma que o sistema inverso possa ser determinado utilizando um conjunto de desenvolvimento gerado por simulação, que se compõe por um vetor de valores desejados x e um vetor contendo valores da saída do sistema, r . Esse procedimento pode ser simplificado quando o tamanho do vetor r equivale à ordem do filtro adaptativo. Dessa maneira, a determinação dos coeficientes do filtro de deconvolução, representado aqui por \hat{g} , com x e r considerados vetores, resulta em uma questão de estimação de parâmetros.

Existem diversos estimadores, normalmente divididos em ótimos e subótimos (KAY, 1993). Porém, o único estimador no qual nenhuma hipótese probabilística sobre os dados é necessária para o projeto de estimação resultar em um modelo linear é o sub-ótimo *Least Square* (LS). A fundamentação neste estimador passa a ser adequada, quando a partir de $x[n]$ se consegue chegar em $r[n]$ sem precisar conhecer o sistema, utilizando um certo atraso nas amostras de entrada para o cálculo do erro, baseado na minimização do RMS, assumindo o projeto do filtro adaptativo da Figura 9 (Apêndice B) (DUARTE, 2015).

3.3.1.2 Estimação utilizando least squares

A estimação utilizando o método do *Least Square* (LS), resume-se na escolha dos parâmetros $\hat{\mathbf{g}}$ que minimizam a Equação 15.

$$J(\hat{\mathbf{g}}) = \sum_{n=0}^{W-1} (x[n] - v[n])^2 \quad (15)$$

Essa equação é o critério de erro adotado pelo estimador, também conhecida como função custo J . Uma representação visual pode ser conferida no Diagrama 8 (Apêndice B).

Na Equação 15: W é o número de amostras coletadas; x é um vetor de tamanho W , que representa amostras consecutivas do sinal desejado; e v , de mesmo tamanho, depende linearmente do vetor r e $\hat{\mathbf{g}}$ (de tamanho U), através de um modelo adequado para o sistema desconhecido (DUARTE, 2015). Note-se a representação da Equação 16:

$$J(\hat{\mathbf{g}}) = \sum_{n=0}^{W-1} (x[n] - (\hat{\mathbf{g}}' \otimes r))^2 \quad (16)$$

A função custo pode ser reescrita em função de $\hat{\mathbf{g}}$ sob uma forma matricial. Desse modo, a dimensão de $\hat{\mathbf{g}}$ deve ser pré-definida (ordem do filtro U). Denota-se, portanto, que $\hat{\mathbf{g}}$ possui dimensão $U \times 1$ e \mathbf{R} é uma matriz de observação de dimensão $W \times U$ ($W > U$). A matriz \mathbf{R} é formada pelo deslocamento das amostras de $r[n]$ observadas, onde a primeira coluna é formada pelos primeiros W valores de $r[n]$, a segunda pelo deslocamento de $r[n]$ de uma amostra e, assim, sucessivamente. O número de colunas deve ser compatível com a dimensão da ordem U escolhida. Assim, o vetor $v[n]$ pode ser representado em função da matriz \mathbf{R} . Para um exemplo de $U = 5$ e $W = 8$ tem-se a Equação 17.

$$\begin{bmatrix} r_1 & 0 & 0 & 0 & 0 \\ r_2 & r_1 & 0 & 0 & 0 \\ r_3 & r_2 & r_1 & 0 & 0 \\ r_4 & r_3 & r_2 & r_1 & 0 \\ r_5 & r_4 & r_3 & r_2 & r_1 \\ r_6 & r_5 & r_4 & r_3 & r_2 \\ r_7 & r_6 & r_5 & r_4 & r_3 \\ r_8 & r_7 & r_6 & r_5 & r_4 \end{bmatrix} \times \begin{bmatrix} \hat{g}_1 \\ \hat{g}_2 \\ \hat{g}_3 \\ \hat{g}_4 \\ \hat{g}_5 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix} \quad (17)$$

A função custo da Equação 15, reescrita na forma matricial em função de $\hat{\mathbf{g}}$, é dada pela Equação 18:

$$J(\hat{\mathbf{g}}) = (\mathbf{x} - \mathbf{R}\hat{\mathbf{g}})^T(\mathbf{x} - \mathbf{R}\hat{\mathbf{g}}) \quad (18)$$

Distribuindo a Equação 18, obtém-se a Equação 19 e, minimizando a função de custo J , obtendo a sua derivada, tem-se a Equação 20. Fazendo o gradiente da derivada igual a zero

chega-se à Equação 21:

$$J(\hat{\mathbf{g}}) = \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{R}\hat{\mathbf{g}} + \hat{\mathbf{g}}^T \mathbf{R}^T \mathbf{R}\hat{\mathbf{g}} \quad (19)$$

$$\frac{\partial J(\hat{\mathbf{g}})}{\partial \hat{\mathbf{g}}} = -2\mathbf{R}^T \mathbf{x} + 2\mathbf{R}^T \mathbf{R}\hat{\mathbf{g}} = 0 \quad (20)$$

$$2\mathbf{R}^T \mathbf{x} = 2\mathbf{R}^T \mathbf{R}\hat{\mathbf{g}} \quad (21)$$

Assumindo que $\mathbf{R}^T \mathbf{R}$ garanta a invertibilidade, $\hat{\mathbf{g}}$ pode ser estimado conforme a Equação 22. Essa equação mostra que a relação entre os pesos de $\hat{\mathbf{g}}$ e o sinal de entrada é dada pela pseudoinversa da matriz de observação \mathbf{R} (KAY, 1993) multiplicado pelas amostras consecutivas do sinal desejado $x[n]$.

$$\hat{\mathbf{g}} = (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{R}^T \mathbf{x} \quad (22)$$

O vetor de parâmetros $\hat{\mathbf{g}}$ estimado, corresponde à própria resposta ao impulso de um sistema inverso ao sistema desconhecido. Sendo assim, esse vetor de coeficientes pode ser interpretado como um Filtro FIR de ordem ($U - 1$) que deconvolui o sistema desconhecido (DUARTE, 2015).

Ao considerar, por conveniência, \mathbf{r} como um vetor de sinal recebido do TileCal e \mathbf{x} como um vetor de energias por BC de uma determinada célula do TileCal através de simulação, $\hat{\mathbf{g}}$ se torna um Filtro FIR que deconvolui o sinal recebido do TileCal, após sua implementação de maneira *on-line*. Assim, a energia por BC pode ser estimada diretamente na saída do filtro.

3.4 FILTRO FIR DESENVOLVIDO

De forma a propor um meio de comparação entre o Filtro Casado utilizado até a Fase I de atualizações do TileCal e o Filtro FIR baseado em deconvolução apontado por Duarte (2015), foi desenvolvida uma adaptação no algoritmo do Filtro FIR original. Tal adaptação foi nomeada de *Filtro FIR de Deconvolução por modelagem Inversa com respostas Positivas* (FDIP).

O Filtro FDIP implementado neste trabalho pode ser visualizado no Algoritmo 1, onde uma modificação que impede que todos os valores gerados sejam negativos também pode ser notada. Tal alteração se fez necessária por não ser de interesse tratar valores negativos para a energia obtida do TileCal e para facilitar a comparação destes valores de RMS obtidos entre os diversos métodos analisados neste trabalho, seguindo o critério de minimização do RMS entre os dados e o modelo.

Algoritmo 1: IMPLEMENTAÇÃO DO FILTRO FDIP

Entrada: Sinal x' , sinal r' , sinal r e ordem do Filtro

Saída: sinal \hat{x}

```

1 início
2    $G \leftarrow \text{ inicializa}$ 
3    $G' \leftarrow \text{ inicializa}$ 
4    $C \leftarrow \text{cov}(Y)$ 
5    $W \leftarrow \frac{hC^{-1}}{(hC^{-1})h^T}$ 
6    $z \leftarrow Wr$ 
7   para  $it \leftarrow (it + 1)$  até  $ordem$  faca
8      $G \leftarrow \text{circshift}(r, it - 1)$ 
9      $G' \leftarrow \text{circshift}(x', it - 1)$ 
10   fim
11    $y = G'(:, \text{arredondaAcima}(ordem/2))$ 
12    $\theta \leftarrow G^+y$ 
13    $[dimX, dimY] \leftarrow \text{dimensao}(G)$ 
14    $z \leftarrow \theta \otimes r'$ 
15    $\hat{x} \leftarrow z(\text{arredondaAbaixo}(dimY/2) : \text{tamanho} - \text{arredondaAcima}(dimY/2))$ 
16 fim
17 retorna  $\hat{x}$ 

```

3.5 GRADIENTE DESCENDENTE

Diferentemente da abordagem adotada até agora, os estudos dos próximos métodos buscarão obter os valores do vetor x diretamente dentro uma janela amostral de pulsos vindos do TileCal, ao invés de se obter os coeficientes de um Filtro FIR. O primeiro método a utilizar essa abordagem será o *Gradiente Descendente* (GD).

Duarte (2015) descreve que o GD é um método que parte da minimização do RMS. O processo se inicia minimizando a Equação 23:

$$J(\hat{x}) = (r - \mathbf{H}\hat{x})^T(r - \mathbf{H}\hat{x}) \quad (23)$$

onde: r representa o vetor de amostras recebidas e $\mathbf{H}\hat{x}$ representa um modelo linear para a convolução (Equação 24). O objetivo é estimar diretamente os valores do vetor x que minimizam a função de custo J . Analogamente à Equação 22, a minimização resultará na estimação da pseudoinversa da matriz de convolução \mathbf{H} (TEIXEIRA; ANDRADE-FILHO; PERALVA, 2014), que pode ser visualizada na Equação 25:

$$x = \mathbf{H}y \quad (24)$$

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{r} \quad (25)$$

Uma outra forma de executar a minimização da Equação 23 foi desenvolvida para esta tese, assim como ocorreu em Duarte (2015). Aqui não se usa o recurso de inversão de matrizes, porque os valores de $\hat{\mathbf{x}}$ são estimados e devem ser atualizados na direção contrária ao gradiente da função de custo J . Para tal, pequenos passos são dados de acordo com o índice i . Essa abordagem é conhecida na literatura como *Gradiente Descendente* (GD) e sua formulação é dada por:

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + \mu \mathbf{H}^T (\mathbf{r} - \mathbf{H} \hat{\mathbf{x}}_i) \quad (26)$$

Segundo Duarte (2015), a equação geral do GD é análoga à equação do método de deconvolução de Van Cittert (TEIXEIRA; FILHO, 2021). O desenvolvimento desse método de deconvolução pode ser aferido nesses dois trabalhos. Continuando em Duarte (2015), um fator de relaxamento μ otimizado para a equação geral do GD é apresentado, onde sua taxa de convergência é melhorada e cujo desenvolvimento completo pode ser aferido neste trabalho. O resultado final pode ser conferido através da Equação 27. Duarte *et al.* (2016) mencionam que pode ser encontrado, na literatura, um limite superior para μ , tal que seu valor ultrapasse o maior autovalor da matriz obtida por $\mathbf{H}^T \mathbf{H}$ dividido por dois.

$$\mu = \frac{\varphi^T \varphi}{\varphi^T (\mathbf{H}^T \mathbf{H}) \varphi} \quad (27)$$

onde: φ é equivalente a $(\mathbf{r} - \mathbf{H} \hat{\mathbf{x}}_i)$. O desenvolvimento completo da Equação 27 pode ser verificado em Duarte (2015). Destarte, o pseudocódigo que implementa as Equações 26 e 27 pode ser averiguado no Algoritmo 2 e a Equação 27 executada na linha 13.

Algoritmo 2: IMPLEMENTAÇÃO DO GRADIENTE DESCENDENTE

Entrada: Tamanho da amostra, tamanho da janela, número de iterações, matriz H e sinal r

Saída: sinal \hat{x}

```

1 início
2    $A \leftarrow H^T H$ 
3    $W \leftarrow \text{inicializa}$ 
4    $S \leftarrow \text{inicializa}$ 
5    $\hat{x} \leftarrow \text{inicializa}$ 
6   para  $it \leftarrow (it + 1)$  até  $\text{quantidadeDeAmostras}$  faz
7      $\text{passo} \leftarrow ((it * \text{janela}) - (\text{janela} - 1))$ 
8      $W \leftarrow r(\text{passo} : \text{passo} + (\text{janela} - 1))$ 
9      $S \leftarrow W^T$ 
10     $H^T r \leftarrow H^T W^T$ 
11    para  $iti \leftarrow (iti + 1)$  até  $\text{númeroDeIterações}$  faz
12       $B \leftarrow H^T r - (AS)$ 
13       $\mu \leftarrow B^T B / B^T AB$ 
14       $S \leftarrow (S + \mu B)$ 
15    fim
16     $\hat{x}(\text{passo} : \text{passo} + (\text{janela} - 1)) \leftarrow S$ 
17  fim
18 fim
19 retorna  $\hat{x}$ 

```

3.6 CONCLUSÃO DA SEÇÃO 3

Dentre os métodos já estudados por nosso grupo de pesquisa e para fins de análises comparativas, foram considerados os métodos: Filtro Casado, *Filtro FIR de Deconvolução por modelagem Inversa com respostas Positivas* (FDIP) e *Gradiente Descendente* (GD), pois assim pode-se observar a evolução de métodos que vão desde o uso na Fase I (Filtro Casado), passando por métodos atualmente em uso (baseados em Filtro FIR como o Filtro de Autocorrelação⁸) e propostas realizadas pelo grupo de pesquisa no qual este trabalho está inserido (GD).

Após a apresentação do desenvolvimento teórico/matemático dos métodos até aqui estudados pelo grupo de pesquisa e do método atualmente em uso no TileCal, a próxima seção foca no estudo teórico/matemático dos novos métodos mencionados neste trabalho, além da adaptação para uso *on-line* do método *Dantzig-Selector* (DS), já pesquisado pelo grupo.

⁸Sendo o filtro aqui considerado o Filtro FDIP, que apesar de ser uma proposta do nosso grupo é uma opção com uma estrutura bastante semelhante ao Filtro de Autocorrelação.

4 REPRESENTAÇÃO ESPARSA PARA IMPLEMENTAÇÃO ON-LINE

Trabalhos anteriores realizados pelo grupo de pesquisa, no qual este se insere, apontam que os métodos iterativos têm desempenho superior ao das técnicas baseadas em Filtros FIR (DUARTE, 2015; DUARTE *et al.*, 2016), no que diz respeito à detecção e acurácia da informação de energia reconstruída. O custo computacional é alto significando, portanto, um desafio para a proposta de algoritmos a serem utilizados em ambiente embarcado.

Nesse contexto, teorias modernas de *Representação Esparsa* (RE) de dados buscam obter soluções mais eficientes (ELAD, 2010) em termos de implementação. Nesta seção, o desenvolvimento teórico/matemático de métodos de RE de dados será estudado, formando a base teórica necessária para a construção de algoritmos iterativos de deconvolução baseados em RE, visando a reconstrução *online* de energia no Calorímetro Hadrônico do ATLAS.

4.1 REPRESENTAÇÃO ESPARSA

Ao considerar um vetor $\mathbf{r} \in \mathbb{R}^n$, $\mathbf{r} = [\alpha_1 \cdots \alpha_n]^T$, o sinal representado por ele será exatamente ou estritamente esparso se a maioria de seus elementos forem iguais a zero. Ou seja, se o suporte $\Lambda(\mathbf{r}) = \{1 \leq i \leq n | \alpha_i \neq 0\}$ for de cardinalidade $i \ll n$ (STARCK; MURTAGH; FADILI, 2010). Portanto, um sinal é dito i -esparso quando possui, no máximo, i elementos não nulos. Caso o sinal não seja esparso, ele ainda pode possuir uma RE em um domínio de transformada apropriado (STARCK; MURTAGH; FADILI, 2010). Genericamente, é possível representar o sinal \mathbf{r} como combinações lineares de N formas elementares, tais que:

$$\mathbf{r} = \mathbf{H}\mathbf{x} = \sum_{i=1}^n \mathbf{v}_i \alpha_i \quad (28)$$

onde: α_i são os coeficientes de representação de \mathbf{r} na matriz dicionário \mathbf{H}^9 , sendo $\mathbf{H} = [\mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_n]$ (uma matriz $M \times N$). Com isso, caso seja possível representar \mathbf{r} usando poucos elementos de \mathbf{H} tem-se uma RE do sinal \mathbf{r} .

Conforme se pode verificar em Silva (2016), a utilização de técnicas de RE de sinais em dicionários redundantes (base não-ortogonal) vem aumentando recentemente. Tais técnicas buscam solucionar problemas através da representação mais compacta (esparsa) de um sinal, em termos de uma combinação linear de elementos pertencentes a um dicionário redundante. Elas apresentam desempenho superior por a sua capacidade de modelagem, quando comparadas às transformadas ortogonais (HUANG; AVIYENTE, 2007).

A obtenção da RE de um sinal \mathbf{r} , consiste em computar os coeficientes da sua representação (vetor \mathbf{x}) e uma matriz dicionário \mathbf{H} que seja redundante. Ou seja, possui mais elementos (como colunas, por exemplo) do que o necessário para representar o sinal. Dessa forma, o sistema $\mathbf{r} = \mathbf{H}\mathbf{x}$ (Equação 24) faz com que o tamanho do vetor \mathbf{r} seja maior que o do vetor \mathbf{x} , tornando-o um sistema linear indeterminado, o que significa dizer que possui infinitas soluções. Portanto,

⁹No contexto da Representação Esparsa a matriz de convolução \mathbf{H} é chamada de matriz dicionário.

aquela que apresentar o menor número de coeficientes não nulos é a representação desejada de \mathbf{r} .

Encontrar a solução mais esparsa de um sistema linear indeterminado é um problema de otimização baseado em restrições impostas por uma função custo. Uma função custo que favorece a esparsidade é a norma ℓ_0 do vetor de coeficientes \mathbf{x} , dando origem ao problema (P_0) mostrado na Equação 29 ou, quando na presença de ruído, ao problema $(P_{0,\epsilon})$ mostrado na Equação 30.

$$(P_0) : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ sujeito a } \mathbf{r} = \mathbf{Hx} \quad (29)$$

$$(P_{0,\epsilon}) : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ sujeito a } \|\mathbf{r} - \mathbf{Hx}\|_2 \leq \epsilon \quad (30)$$

em que: $\|\mathbf{x}\|_0$ representa a norma ℓ_0 do vetor \mathbf{x} , que é o número de elementos não nulos deste vetor.

Generalizando, a norma ℓ_q é dada pela Equação 31:

$$\ell_q = \sum_{i=1}^n |\mathbf{x}_i|^q \quad (31)$$

Os problemas (P_0) e $(P_{0,\epsilon})$ não possuem solução de maneira direta. Esse problema é, geralmente, dividido em duas partes: a primeira é determinar o suporte de \mathbf{x} (sendo os índices dos elementos não nulos); e a segunda, logo depois, determinar o valor dos elementos do suporte. Dessa forma, uma maneira de resolver o problema é focar no suporte, consciente de que, posteriormente, os valores dos elementos não nulos de \mathbf{x} podem ser encontrados via Regressão Linear. Essa linha de raciocínio é originária dos métodos conhecidos como *Greedy* (vorazes ou gulosos) (ELAD, 2010).

O problema mostrado na Equação 30 (com maior relaxamento) é mais interessante tanto em aplicações de *denoising* (remoção de ruído), quanto em aplicações de compactação, porque algumas informações presentes no sinal, como o ruído, por exemplo, não são desejadas em sua representação.

A utilização da norma ℓ_1 , como função de custo é outra forma, também muito difundida, para obtenção da esparsidade. Nela, pode-se indicar que a minimização da norma ℓ_q , para $q \leq 1$ tende a esparsidade (ELAD, 2010). Assim, o problema a ser resolvido torna-se (P_1) (Equação 32), ou em sua forma mais relaxada, $(P_{1,\epsilon})$, visto na Equação 33:

$$(P_1) : \min_{\mathbf{x}} \|\mathbf{x}\|_1^1 \text{ sujeito a } \mathbf{r} = \mathbf{Hx} \quad (32)$$

$$(P_{1,\epsilon}) : \min_{\mathbf{x}} \|\mathbf{x}\|_1^1 \text{ sujeito a } \|\mathbf{r} - \mathbf{Hx}\|_2 \leq \epsilon \quad (33)$$

em que: $\|\mathbf{x}\|_1^1$ representa a norma ℓ_1 , do vetor \mathbf{x} , que é a soma dos valores absolutos dos coeficientes deste vetor.

O problema (P_1, ϵ) , mostrado na Equação 33, pode ser escrito como sendo de otimização, com uma função custo linear e com restrições quadráticas. Esse tipo de problema é bem solucionado com métodos provenientes da teoria de otimização atual. Em algumas situações, esses métodos de otimização de propósito geral podem se mostrar muito lentos e algumas técnicas especiais são necessárias para solucionar essa questão. Por exemplo: a escolha de um multiplicador de Lagrange λ apropriado (STARCK; MURTAGH; FADILI, 2010) pode transformar o problema (P_1, ϵ) no problema de otimização sem restrições, como mostrado na Equação 34:

$$(Q_1^\lambda) : \min_{\mathbf{x}} \lambda \|\mathbf{x}\|_1^1 + \frac{1}{2} \|\mathbf{r} - \mathbf{Hx}\|_2^2 \quad (34)$$

em que: o multiplicador de Lagrange λ é função de \mathbf{H} , \mathbf{r} e ϵ . Nota-se, nessa função custo, a presença das normas ℓ_2 e ℓ_1 . A primeira lida com a representação dos dados em meio ao ruído, enquanto a segunda busca a esparsidade. O multiplicador de Lagrange atua como peso para ponderar entre as duas normas e é obtido experimentalmente através da escolha do desvio ϵ .

Alguns métodos especializados para a minimização de (Q_1^λ) , baseados no processo de redução iterativa, serão discutidos a seguir. Também serão analisados métodos *Greedy* (ELAD, 2010), que abandonam a busca exaustiva e trabalham com uma série de atualizações termo a termo, adicionando elementos iterativamente para reconstruir uma aproximação contendo o reduzido número de elementos não nulos.

4.1.1 Métodos para representação esparsa

A busca exaustiva seria uma maneira de se encontrar a solução para (P_0) . No entanto, essa solução se mostra inviável em quase todas as situações. Uma suposição, como exemplo: a matriz \mathbf{H} possua N linhas e M colunas e que o vetor \mathbf{r} seja uma combinação linear de, no máximo, i_0 elementos dessa matriz, seria necessário enumerar todas as possíveis combinações de i_0 elementos de \mathbf{H} e testá-las. Para isso, são necessárias $O(m_{i_0}n_0^2)$ flops, o que torna esse método extremamente lento (ELAD, 2010).

Para solucionar esse tipo de problema, diversos métodos vêm sendo desenvolvidos. Esses métodos podem ser classificados em três principais classes:

- os métodos *Greedy*, que têm como objetivo minimizar as funções (P_0) ou $(P_{0,\epsilon})$;
- os métodos que relaxam a norma ℓ_0 e têm como objetivo minimizar as funções (P_1) ou $(P_{1,\epsilon})$; e
- os métodos iterativos de redução (*Iterative Shrinkage Algorithm*) que minimizam uma forma mais geral de (Q_1^λ) (Equação 34).

4.2 MÉTODOS GREEDY

Um método *Greedy* abandona a busca exaustiva e trabalha com uma série de atualizações, termo a termo, com a aproximação $\mathbf{x}^0 = 0$; além disso, adiciona elementos iterativamente para

reconstruir uma aproximação x_i , contendo i elementos não nulos. Um método dessa família é o OMP (PATI; REZAIIFAR; KRISHNAPRASAD, 1993), muito usado em processamento de sinais.

Neste estudo, serão analisados três métodos *Greedy*, sendo que todos são variações de um mesmo método conhecido como *Matching Pursuit* (MP) (WU; WANG, 2012): o primeiro método é o que dará origem aos outros dois métodos, ou seja, o método MP; o segundo método será a versão ortogonal do método MP, o método *Orthogonal-Matching-Pursuit* (OMP) (CAI; WANG, 2011); e o terceiro método será o método *Least-Square Orthogonal-Matching-Pursuit* (LS-OMP) (DUMITRESCU, 2013), onde é acrescentado uma etapa de *Least Square* ao método OMP.

Antes de prosseguir com as definições dos métodos de *Greedy*, é importante que o termo *spark* seja definido. Segundo Elad (2010), o *spark* da matriz \mathbf{H} – termo cunhado e definido em 2003, por Donoho e Elad (2003) – é uma propriedade chave, crucial para o estudo da singularidade. O *spark* é uma forma de caracterizar o espaço nulo de uma matriz \mathbf{H} usando a norma ℓ_0 . Para tal segue-se a definição: "Definição 2.2: O spark de uma determinada matriz \mathbf{H} é o menor número de colunas de \mathbf{H} que são linearmente dependentes" (ELAD, 2010, p. 23).

4.2.1 Matching-pursuit

A ideia básica por trás dos métodos que variam a partir do MP (WU; WANG, 2012) parte da suposição de que, havendo uma matriz \mathbf{H} que tenha $\text{spark}(\mathbf{H}) > 2$ e o problema de otimização (P_0) tenha valor $\text{val}(P_0) = 1$ (na solução ótima), então \mathbf{r} é um múltiplo escalar de alguma coluna da matriz \mathbf{H} e essa solução é conhecida por ser única. Pode-se identificar essa coluna aplicando M testes - um por coluna de \mathbf{H} . O j -ésimo teste pode ser feito pela minimização $\epsilon(j) = \|\mathbf{h}_j z_j - \mathbf{r}\|_2^2$, levando a $z_j = \mathbf{h}_j^T \mathbf{r} / \|\mathbf{h}_j\|_2^2$. Reinserindo na expressão de erro, o erro se torna:

$$\begin{aligned} \epsilon(j) &= \min_{z_j} \|\mathbf{h}_j z_j - \mathbf{r}\|_2^2 = \left\| \frac{\mathbf{h}_j^T \mathbf{r}}{\|\mathbf{h}_j\|_2^2} \mathbf{h}_j - \mathbf{r} \right\|_2^2 \\ &= \|\mathbf{r}\|_2^2 - 2 \frac{(\mathbf{h}_j^T \mathbf{r})^2}{\|\mathbf{h}_j\|_2^2} + \frac{(\mathbf{h}_j^T \mathbf{r})^2}{\|\mathbf{h}_j\|_2^2} \\ &= \|\mathbf{r}\|_2^2 - \frac{(\mathbf{h}_j^T \mathbf{r})^2}{\|\mathbf{h}_j\|_2^2}. \end{aligned} \quad (35)$$

Se o erro for zero, a solução foi encontrada.

Como a esparsidade da matriz de convolução \mathbf{H} , aqui utilizada, aumenta de acordo com o número considerado de amostras do sinal, então, o sistema possui uma solução $\hat{\mathbf{x}}$ que satisfaz $\|\hat{\mathbf{x}}\|_0 < \frac{\text{spark}(\mathbf{H})}{2}$ (ELAD, 2010).

Por isso, no estágio de atualização da solução provisória, os coeficientes que já fazem

parte do suporte S_{i-1} são mantidos inalterados e o novo coeficiente, referente à j_0 da matriz \mathbf{H} é o resíduo r_{i-1} . Essa diferença faz do MP (WU; WANG, 2012) um método computacionalmente de menor custo, porém, menos preciso que o OMP.

Com base no trabalho de Elad (2010), desenvolveu-se uma implementação do método MP, cujo pseudocódigo se encontra no Algoritmo 3:

Algoritmo 3: IMPLEMENTAÇÃO DO MATCHED PURSUIT

Entrada: Tamanho da amostra, tamanho da janela, patamar, matriz \mathbf{H} e sinal r

Saída: sinal \hat{x}

```

1 início
2    $M \leftarrow \text{inicializa}$ 
3    $\hat{x} \leftarrow \text{inicializa}$ 
4   para  $it \leftarrow (it + 1)$  até quantidadeDeAmostras faz
5      $\text{passo} \leftarrow ((it * \text{janela}) - (\text{janela} - 1))$ 
6      $s \leftarrow r(\text{passo} : \text{passo} + (\text{janela} - 1))$ 
7      $y \leftarrow s$ 
8     enquanto  $(y^T y) > \text{patamar}$  faz
9        $Z \leftarrow \text{abs}(H^T y)$ 
10       $\text{posZ} \leftarrow \text{procura}(Z == \text{maximo}(Z), 1)$ 
11       $\text{temp} \leftarrow H(:, \text{posZ})^T y$ 
12       $M(\text{posZ}) \leftarrow M(\text{posZ}) + \text{temp}$ 
13       $y \leftarrow y - H(:, \text{posZ}) * \text{temp}$ 
14    fim
15     $\hat{x}(\text{passo} : \text{passo} + (\text{janela} - 1)) \leftarrow M^T$ 
16  fim
17 fim
18 retorna  $\hat{x}$ 

```

Existem, ainda, diversas outras modificações com o intuito de simplificar ou melhorar o desempenho desses métodos, mas que não serão mencionadas neste trabalho.

4.2.2 Orthogonal-matching-pursuit

O OMP (CAI; WANG, 2011) é um método *Greedy* em que a cada iteração seleciona-se o elemento do dicionário que possui maior projeção ortogonal no sinal residual (assumindo que os elementos do dicionário estejam normalizados). O elemento selecionado é adicionado ao suporte da solução e o coeficiente de todos os elementos do vetor são atualizados via Regressão Linear (SEBER, 2003).

Na Equação 36, pode-se notar que a forma na qual a fase de varredura fornece os valores de erro é muito semelhante ao que se pode ver na Equação 35. Os produtos internos entre o resíduo r_{i-1} e cada elemento (coluna) \mathbf{h}_j da matriz \mathbf{H} são calculados no estágio de varredura. O índice j_0 do elemento \mathbf{h}_j que apresentou maior valor de produto interno é adicionado ao suporte no estágio de atualização do suporte. A atualização da solução provisória é feita via *Least Square* (LS) (MILLER, 2006), utilizando somente os elementos da matriz \mathbf{H} relativos ao

suporte de \mathbf{x} . Por fim, o novo resíduo \mathbf{r}_i é calculado e o critério de parada é avaliado.

$$\begin{aligned}
 \epsilon(j) = \min_{\mathbf{z}_j} \|\mathbf{h}_j \mathbf{z}_j - \mathbf{r}_{i-1}\|_2^2 &= \left\| \frac{\mathbf{h}_j^T \mathbf{r}_{i-1}}{\|\mathbf{h}_j\|_2^2} \mathbf{h}_j - \mathbf{r}_{i-1} \right\|_2^2 \\
 &= \|\mathbf{r}_{i-1}\|_2^2 - 2 \frac{(\mathbf{h}_j^T \mathbf{r}_{i-1})^2}{\|\mathbf{h}_j\|_2^2} + \frac{(\mathbf{h}_j^T \mathbf{r}_{i-1})^2}{\|\mathbf{h}_j\|_2^2} \\
 &= \|\mathbf{r}_{i-1}\|_2^2 - \frac{(\mathbf{h}_j^T \mathbf{r}_{i-1})^2}{\|\mathbf{h}_j\|_2^2}.
 \end{aligned} \tag{36}$$

A obtenção dos coeficientes no estágio de atualização da solução provisória é feita através da minimização da expressão $\|\mathbf{H}_{S_i} - \mathbf{x}_{S_i} - \mathbf{r}\|_2^2$ em que \mathbf{x}_{S_i} é a parte não nula do vetor \mathbf{x} e \mathbf{H}_{S_i} é uma matriz formada somente pelas colunas de \mathbf{H} que foram escolhidas até o momento. A solução desse problema é obtida igualando-se a derivada dessa expressão quadrática a zero, como mostrado na Equação 37 (ELAD, 2010).

$$\mathbf{H}_{S_i}(\mathbf{H}_{S_i} - \mathbf{x}_{S_i} - \mathbf{r}) = -\mathbf{H}_{S_i}\mathbf{r}_i = 0 \tag{37}$$

em que $\mathbf{r}_i = \mathbf{r} - \mathbf{H}\mathbf{x}_i$.

A Equação 37 mostra que as colunas da matriz \mathbf{H} que fazem parte do suporte S_i são, necessariamente, ortogonais ao resíduo \mathbf{r}_i e, portanto, não serão escolhidas novamente nas futuras iterações. Esse processo de ortogonalização é que nomeia o método. A complexidade computacional do método OMP, supondo que a solução possua i_0 elementos, é igual a $O(k_0 mn)$ flops, o que, em geral, é bem menor que as $O(m_{i_0} n k_0^2)$ flops necessárias para a busca exaustiva pela solução.

Baseado na Equação 36 e no trabalho de Elad (2010), foi desenvolvido o Algoritmo 4 adaptando os dados de entrada utilizados nesta tese à forma como o método atua sobre os mesmos.

Algoritmo 4: IMPLEMENTAÇÃO DO ORTHOGONAL MATCHING PURSUIT

Entrada: Tamanho da amostra, tamanho da janela, patamar, matriz \mathbf{H} e sinal \mathbf{r}

Saída: sinal $\hat{\mathbf{x}}$

```

1 início
2    $O \leftarrow \text{inicializa}$ 
3    $\hat{\mathbf{x}} \leftarrow \text{inicializa}$ 
4   para  $it \leftarrow (it + 1)$  até  $\text{quantidadeDeAmostras}$  faça
5      $\text{passo} \leftarrow ((it * \text{janela}) - (\text{janela} - 1))$ 
6      $s \leftarrow \mathbf{r}(\text{passo} : \text{passo} + (\text{janela} - 1))$ 
7      $y \leftarrow s$ 
8      $pos \leftarrow \text{inicializa}$ 
9     enquanto  $(y^T y) > \text{patamar}$  faça
10        $Z \leftarrow \text{abs}(\mathbf{H}^T y)$ 
11        $posZ \leftarrow \text{procura}(Z == \text{maximo}(Z), 1)$ 
12        $pos \leftarrow \text{sorteia}(pos, posZ[1])$ 
13        $y \leftarrow s - \mathbf{H}(:, pos)\mathbf{H}(:, pos)^+s$ 
14     fim
15      $O(pos) \leftarrow \mathbf{H}(:, pos)^+s$ 
16      $\hat{\mathbf{x}}(\text{passo} : \text{passo} + (\text{janela} - 1)) \leftarrow O^T$ 
17   fim
18 fim
19 retorna  $\hat{\mathbf{x}}$ 

```

4.2.3 Least-squares orthogonal-matching-pursuit

O LS-OMP (DUMITRESCU, 2013) é uma variação mais complexa e com um comportamento pouco melhor que a variação OMP (CAI; WANG, 2011), pois cada um dos testes no estágio de varredura é feito como LS completos. Assim, considera todas as colunas acumuladas e a candidata juntas, resolvendo todos os coeficientes de uma só vez. Os passos LS podem ser tornados eficientes usando a Equação 38 (ELAD, 2010).

$$\begin{bmatrix} \mathbf{M} & \mathbf{r} \\ \mathbf{r}^T & c \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{M}^{-1} + p\mathbf{M}^{-1}\mathbf{r}\mathbf{r}^T\mathbf{M}^{-1} & -p\mathbf{M}^{-1}\mathbf{r} \\ -p\mathbf{r}^T\mathbf{M}^{-1} & p \end{bmatrix} \quad (38)$$

onde: $p = 1/(c - \mathbf{r}^T \mathbf{M}^{-1} \mathbf{r})$. Denotando a submatriz que reúne as $i - 1$ colunas selecionadas de \mathbf{H} como \mathbf{H}_s , o problema LS que deve ser solucionado é dado pela Equação 39:

$$\min_{\mathbf{Y}_s, \mathbf{Z}} \left\| [\mathbf{H}_s \mathbf{h}_i] \begin{bmatrix} \mathbf{Y}_s \\ \mathbf{Z} \end{bmatrix} - \mathbf{r} \right\|_2^2. \quad (39)$$

O Algoritmo 5, foi utilizado para implementar o LS-OMP neste trabalho.

Algoritmo 5: IMPLEMENTAÇÃO DO LEAST-SQUARE ORTHOGONAL MATCHINGPURSUIT

Entrada: Tamanho da amostra, tamanho da janela, número de bunches com colisão, patamar, matriz H e sinal r

Saída: sinal \hat{x}

```

1 início
2    $L \leftarrow \text{inicializa}$ 
3    $\hat{x} \leftarrow \text{inicializa}$ 
4   para  $it \leftarrow (it + 1)$  até  $\text{quantidadeDeAmostras}$  faça
5      $passo \leftarrow ((it * \text{janela}) - (\text{janela} - 1))$ 
6      $s \leftarrow r(passo : passo + (\text{janela} - 1))$ 
7      $y \leftarrow s$ 
8      $pos \leftarrow \text{inicializa}$ 
9     enquanto  $(y^T y) > \text{patamar}$  faça
10        $Z \leftarrow \text{inicializa}$ 
11       para  $iti \leftarrow (iti + 1)$  até  $\text{númeroDeBunchesComColisão}$  faça
12          $posTemp \leftarrow [pos, iti]$ 
13          $yTemp \leftarrow s - H(:, posTemp)H(:, posTemp)^+s$ 
14          $Z(iti) \leftarrow yTemp^T yTemp$ 
15       fim
16        $posZ \leftarrow \text{procura}(Z == \text{minimo}(Z), 1)$ 
17        $pos \leftarrow \text{sorteia}(pos, posZ[1])$ 
18        $y \leftarrow s - H(:, pos)H(:, pos)^+s$ 
19     fim
20      $L(pos) \leftarrow H(:, pos)^+s$ 
21      $\hat{x}(passo : passo + (\text{janela} - 1)) \leftarrow L^T$ 
22   fim
23 fim
24 retorna  $\hat{x}$ 

```

A solução é dada zerando-se a derivada da Equação 39,

$$\begin{bmatrix} \mathbf{H}_S^T \\ \mathbf{h}_i^T \end{bmatrix} [\mathbf{H}_S \mathbf{h}_i] \begin{bmatrix} \mathbf{Y}_S \\ Z \end{bmatrix} - \begin{bmatrix} \mathbf{H}_S^T \\ \mathbf{h}_i^T \end{bmatrix} \mathbf{r} = 0, \quad (40)$$

levando a

$$\begin{bmatrix} \mathbf{Y}_S \\ Z \end{bmatrix}_{opt} = \begin{bmatrix} \mathbf{H}_S^T \mathbf{H}_S & \mathbf{H}_S^T \mathbf{h}_i \\ \mathbf{h}_i^T \mathbf{H}_S & \|\mathbf{h}_i\|_2^2 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{H}_S^T \mathbf{r} \\ \mathbf{h}_i^T \mathbf{r} \end{bmatrix}. \quad (41)$$

Assim, com base na Equação 38, se a matriz $(\mathbf{H}_s^T \mathbf{H}_s)^{-1}$ for armazenada, então a Equação 41 se torna fácil e leva a uma atualização dessa matriz para i colunas. Uma vez encontrada a solução, o erro residual deve ser avaliado e a coluna \mathbf{h}_i que leva ao menor erro deve ser escolhida. Cabe observar que, se \mathbf{h}_i for ortogonal às colunas de \mathbf{H}_s , a matriz a ser invertida acima torna-se bloco-diagonal e o resultado seria $\mathbf{x}_s = \mathbf{H}_s^+ \mathbf{r}$ e $Z = \mathbf{h}_i^T \mathbf{r} / \|\mathbf{h}\|_2^2$. Isso significa que os coeficientes da solução anterior permanecem intactos e um novo não-zero é introduzido com um coeficiente Z (ELAD, 2010).

4.3 MÉTODOS DE RESOLUÇÃO POR PROGRAMAÇÃO LINEAR

Nesta seção, será estudada uma alternativa de resolução da Equação 29 (e também para a Equação 30) por *Linear Program* (LP) (CANDES; TAO, 2007), apesar de esse tipo de solução necessitar de um grande poder computacional para ser implementado. Barbosa (2017), no entanto, demonstrou resultados muito promissores para a sua aplicação em processamento *off-line*.

4.3.1 Dantzig-selector

O que se apresenta, aqui, é uma alternativa competitiva ao OMP o *Dantzig-Selector* (DS). Este algoritmo foi proposto em 2007 por Candes e Tao (2007) e nomeado em tributo ao pai do algoritmo *Simplex*. Tal algoritmo é especialista em resolver problemas se utilizando de LP.

O método DS já foi estudado por Barbosa (2017) para aplicação *off-line* em calorimetria, apresentando bons resultados. Agora, serão propostas alterações nessa versão, de modo que ela possa ser utilizada em um ambiente *on-line*. Para tal, utilizaram-se as definições dadas por Elad (2010) como base, onde:

Considerando o seguinte problema: um vetor esparso \mathbf{x}_0 é multiplicado por uma matriz \mathbf{H} de tamanho $N \times M$, onde, em particular, N pode ser menor que M ; que, daqui por diante, as colunas de \mathbf{H} são ℓ_2 -normalizadas; que o resultado está contaminado por um ruído aditivo Gaussiano branco \mathbf{e} ¹⁰, com entradas i.i.d.¹¹ $e_i \sim \mathcal{N}(0, \sigma^2)$; e que $\mathbf{r} = \mathbf{H}\mathbf{x}_0 + \mathbf{e}$. Objetiva-se: estimar de forma confiável \mathbf{x}_0 .

Utilizando ideias de aproximação de resoluções definidas por Elad (2010), pode ser natural propor o estimador na forma da Equação 42:

$$(P_0^\epsilon) \hat{\mathbf{x}}_0^\epsilon = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ sujeito a } \|\mathbf{r} - \mathbf{H}\mathbf{x}\|_2 \leq \epsilon \quad (42)$$

onde: ϵ escolhido é proporcional a $\sqrt{n}\sigma$, o referido como (P_0^ϵ) e a substituição de ℓ_0 pela norma- ℓ_1 , que leva ao algoritmo *Basis Pursuit Denoising* (GILL; WANG; MOLNAR, 2011)

¹⁰Podemos assumir esse "ruído" como sendo um vetor determinístico de energia limitada $\|\mathbf{e}\|_2 = \epsilon$.

¹¹Em teoria das probabilidades e estatística, uma sequência ou outra coleção de variáveis aleatórias é independente e identicamente distribuída se cada variável aleatória tiver a mesma distribuição de probabilidade das outras e todas forem mutuamente independentes, sendo **i.i.d.** a abreviatura para tal característica.

como opção à complexidade da resolução desse problema (Equação 43).

$$(P_1^\epsilon) \hat{\mathbf{x}}_1^\epsilon = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ sujeito a } \|\mathbf{r} - \mathbf{Hx}\|_2 \leq \epsilon \quad (43)$$

Tornando a descrever o *Dantzig-Selector* (DS), que é uma alternativa muito interessante para (P_1^ϵ) , Candes e Tao (2007) propuseram estimar \mathbf{x}_0 pela resolução do problema (Equação 44).

$$(P_{DS}^\lambda) \hat{\mathbf{x}}_{DS}^\lambda = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ sujeito a } \|\mathbf{H}^T(\mathbf{r} - \mathbf{Hx})\|_\infty \leq \lambda\sigma \quad (44)$$

Como pode ser visto, a restrição foi alterada. A exigência original $\|\mathbf{r} - \mathbf{Hx}\|_2$ faz sentido, uma vez que se espera que o resíduo seja de uma norma limitada. A alternativa proposta é para que DS compute o resíduo $\hat{\mathbf{r}} = \mathbf{r} - \mathbf{Hx}$ e, então, avalie seus produtos internos com as colunas de \mathbf{H} . Estes produtos internos formam várias combinações ponderadas das entradas em $\hat{\mathbf{r}}$ e espera-se que todas elas sejam pequenas. A restrição prossegue, assim, exigindo que todos eles estejam abaixo de alguns limiares $\lambda\sigma$.

O problema colocado na Equação 44 tem uma forma de programação linear. Isto é facilmente observado ao adicionar uma variável auxiliar $\mathbf{u} \geq |\mathbf{x}|$, reformulando o problema como (Equação 45):

$$\min_{\mathbf{u}, \mathbf{x}} \mathbf{1}^T \mathbf{u} \text{ sujeito a } \left\{ \begin{array}{l} -\mathbf{u} \leq \mathbf{x} \leq \mathbf{u} \\ \mathbf{e} \\ -\sigma\lambda\mathbf{1} \leq \mathbf{H}^T(\mathbf{r} - \mathbf{Hx}) \leq \sigma\lambda\mathbf{1} \end{array} \right\} \quad (45)$$

Todos os termos (penalidade e restrições) são lineares em relação às incógnitas e, portanto, este é um problema clássico de LP, para o qual os resolvedores específicos podem ser adaptados (por exemplo, um algoritmo de *shrinkage* iterativo) (CHEN; YANG, 2017). Como se trata de um problema convexo, a obtenção do minimizador global está ao alcance e este estimador é, portanto, bastante atrativo.

Outra forma de converter DS em uma forma de LP é dividir o \mathbf{x} desconhecido em dois vetores que contêm suas entradas positivas e negativas: $\mathbf{x} = \mathbf{u} - \mathbf{v}$. Então, o problema pode ser reescrito como (Equação 46):

$$\min_{\mathbf{u}, \mathbf{v}} \mathbf{1}^T \mathbf{u} + \mathbf{1}^T \mathbf{v} \text{ sujeito a } \left\{ \begin{array}{l} \mathbf{u} \geq 0 \quad \mathbf{v} \geq 0 \\ \mathbf{e} \\ -\sigma\lambda\mathbf{1} \leq \mathbf{H}^T(\mathbf{r} - \mathbf{Hu} + \mathbf{Hv}) \leq \sigma\lambda\mathbf{1} \end{array} \right\} \quad (46)$$

Partindo da Equação 44, tomado como base o desenvolvimento do algoritmo apresentado por Barbosa (2017), fazendo as adaptações necessárias para transformar em *on-line* a versão originalmente *off-line* – notando que este algoritmo de Barbosa (2017) é originalmente baseado no algoritmo descrito em Elad (2010) – chega-se ao Algoritmo 6:

Algoritmo 6: IMPLEMENTAÇÃO DO DANTZIG-SELECTOR

Entrada: Tamanho da amostra, tamanho da janela, matriz \mathbf{H} e sinal \mathbf{r}

Saída: sinal $\hat{\mathbf{x}}$

```

1 início
2    $A \leftarrow H^T H$ 
3    $AA \leftarrow [A, -A; -A, A]$ 
4    $uns \leftarrow inicializa$ 
5    $\hat{x} \leftarrow inicializa$ 
6   para  $it \leftarrow (it + 1)$  até  $quantidadeDeAmostras$  faça
7      $passo \leftarrow ((it * janela) - (janela - 1))$ 
8      $s \leftarrow r(passo : passo + (janela - 1))$ 
9      $bb \leftarrow [(H^T * s) + uns * .12; -(H^T * s) + uns * .12]$ 
10     $um \leftarrow [uns; uns * 9]$ 
11     $res \leftarrow programacaoLinear(um, AA, bb)$ 
12     $saida \leftarrow res[: janela]$ 
13     $\hat{x}(passo : passo + (janela - 1)) \leftarrow saida$ 
14  fim
15 fim
16 retorna  $\hat{x}$ 

```

4.4 MÉTODOS SHRINKAGE

Segundo Elad (2010), existem diferentes formas de se desenvolver Métodos de Encolhimento Iterativo (*Iterative-Shrinkage Algorithms*) (CHEN; YANG, 2017), sendo que a primeira proposta explícita é a de técnicas semelhantes originadas por Starck, Murtagii e Bijaoui (1995), no contexto de imagem *deblurring*, usando *wavelet*.

A técnica de limiarização ou *Shrinkage* desenvolvida por Donoho (1995) é uma técnica simples, porém muito eficiente em aplicações de remoção de ruído. Avanços recentes mostram que essas técnicas de *shrinkage* auxiliam na busca por uma RE de um determinado sinal. Os métodos baseados nessa técnica formam a família de métodos *Shrinkage*, ou Algoritmos Iterativos de Redução.

Os métodos aqui estudados têm o objetivo de minimizar uma função do tipo que se encontra presente na Equação 47, que é uma forma generalizada da Equação 34:

$$f(\mathbf{x}) = \lambda \mathbf{1}^T \rho(\mathbf{x}) + \frac{1}{2} \|\mathbf{r} - \mathbf{Hx}\|_2^2 \quad (47)$$

Nessa equação, a função $\rho(\mathbf{x})$ opera ponto a ponto no vetor \mathbf{x} e deve ser escolhida de forma a promover a esparsidade. Um exemplo: para $\rho(\mathbf{x}) = |\mathbf{x}|^P$, obtém-se $\mathbf{1}^T \rho(\mathbf{x}) = \|\mathbf{x}\|_P^P$, o que dá a liberdade de se escolher qualquer valor de p que se achar adequado (ELAD, 2010).

Diversos métodos iterativos têm sido desenvolvidos para solucionar o problema proposto na Equação 47. De forma simplificada, cada iteração desses métodos consiste em uma multiplicação pela matriz \mathbf{H} e sua transposta, seguida por uma etapa de limiarização do resultado obtido. Para que o procedimento da obtenção de um método que minimize $f(\mathbf{x})$

descrito na Equação 47, para o caso em que \mathbf{H} é uma matriz unitária ($\mathbf{H}\mathbf{H}^T = \mathbf{I}$), $f(\mathbf{x})$ e explorando o fato de que a norma- ℓ_2 é unitariamente invariável, essa equação pode ser facilmente manipulada de forma a dar origem a um conjunto de tarefas de otimização unidimensionais e independentes. Com essa premissa, a função $f(\mathbf{x})$ pode ser reescrita da forma como se apresenta na Equação 48 (ELAD, 2010):

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{2} \|\mathbf{r} - \mathbf{Hx}\|_2^2 + \lambda \mathbf{1}^T \rho(\mathbf{x}) \\ &= \frac{1}{2} \|\mathbf{H}(\mathbf{H}^T \mathbf{r} - \mathbf{x})\|_2^2 + \lambda \mathbf{1}^T \rho(\mathbf{x}) \\ &= \frac{1}{2} \|\mathbf{H}^T \mathbf{r} - \mathbf{x}\|_2^2 + \lambda \mathbf{1}^T \rho(\mathbf{x}) \end{aligned} \quad (48)$$

Fazendo $\mathbf{H}^T \mathbf{r} = \mathbf{x}_0$, chega-se à Equação 49:

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{2} \|\mathbf{x}_0 - \mathbf{x}\|_2^2 + \lambda \mathbf{1}^T \rho(\mathbf{x}) \\ &= \sum_{i=1}^m \left[\frac{1}{2} (\mathbf{x}_0[i] - \mathbf{x}[i])^2 + \lambda \rho(\mathbf{x}[i]) \right] \\ &= \sum_{i=1}^m g(\mathbf{x}[i], \mathbf{x}_0[i]) \end{aligned} \quad (49)$$

A minimização de uma função escalar do tipo $g(\mathbf{x}, \mathbf{h}) = 1/2(\mathbf{x} - \mathbf{h})^2 - \lambda \rho(\mathbf{x})$, em relação a \mathbf{x} , pode ser solucionada analiticamente para alguns casos e numericamente para outros, dando origem à expressão $\hat{\mathbf{x}}_{opt} = S_{\rho, \lambda}(\mathbf{h})$. A função $S_{\rho, \lambda}(\mathbf{h})$ mapeia os elementos próximos à origem em zero ($S_{\rho, \lambda}(\mathbf{h}) = 0$, se $|\mathbf{h}| \leq T$) e os outros elementos são *shrinked*. Portanto, essa função é conhecida como uma função de *shrinkage* (ELAD, 2010).

O que este trabalho propõe é uma modificação na função de encolhimento, a fim de promover a reconstrução positiva da energia, conforme mostrado no Gráfico 10 (Apêndice C).

Pode-se partir do mesmo princípio para obter a função de *shrinkage*, quando a matriz \mathbf{H} deixa de ser unitária e quadrada. Porém, várias iterações deverão ser feitas, para se obter o resultado desejado. Outro método desenvolvido para tratar de casos como esse é o método que se utiliza de funções substitutivas (*surrogate*), que está descrito a seguir.

4.4.1 Separable surrogate functionals

Considerando a função original da Equação 47 (ELAD, 2010) e baseando-se no trabalho de Daubechies, Defrise e Mol (2004), adiciona-se o seguinte termo (Equação 50):

$$h(\mathbf{x}, \mathbf{x}_0) = \frac{c}{2} \|\mathbf{x} - \mathbf{x}_0\|_2^2 - \frac{1}{2} \|\mathbf{Hx} - \mathbf{Hx}_0\|_2^2 \quad (50)$$

O parâmetro c será escolhido de tal forma que a função $h(\cdot)$ seja estritamente convexa, implicando na necessidade de que seu Hessiano seja positivo-definido, $c\mathbf{I} - \mathbf{H}^T \mathbf{H} > 0^{12}$. Isso é satisfeito pela escolha $c > \|\mathbf{H}^T \mathbf{H}\|_2 = \lambda_{max}(\mathbf{H}^T \mathbf{H})$. Assim, essa nova função objetiva

¹²O símbolo *succeeds* > é utilizado em relações onde haja uma preferência completa e transitiva de X sobre Y, quando aplicado no cálculo matricial onde estabeleça relação com zero, como no exemplo dado por Elad (2010), define que a primeira parte deve resultar em algo que seja positivo-definido.

(Equação 51) (ELAD, 2010) é:

$$\tilde{f}(\mathbf{x}) = \frac{1}{2} \|\mathbf{r} - \mathbf{Hx}\|_2^2 + \lambda \mathbf{1}^T \rho(\mathbf{x}) + \frac{c}{2} \|\mathbf{x} - \mathbf{x}_0\|_2^2 - \frac{1}{2} \|\mathbf{Hx} - \mathbf{Hx}_0\|_2^2 \quad (51)$$

a função substituta que será usada no método proposto. Como pode ser visto a seguir, o fato de o termo $\|\mathbf{Hx}\|_2^2$ cair na nova função torna a tarefa de minimização muito mais simples e eficaz. Abrindo os vários termos da Equação 51 e prosseguindo com a sua reorganização, obtém-se uma nova expressão no formato da Equação 52:

$$\begin{aligned} \tilde{f}(\mathbf{x}) &= \frac{1}{2} \|\mathbf{r}\|_2^2 + \frac{1}{2} \|\mathbf{Hx}_0\|_2^2 + \frac{c}{2} \|\mathbf{x}_0\|_2^2 - \mathbf{r}^T \mathbf{Hx} + \lambda \mathbf{1}^T \rho(\mathbf{x}) \\ &\quad + \frac{c}{2} \|\mathbf{x}\|_2^2 - c \mathbf{x}^T \mathbf{x}_0 + \mathbf{x}^T \mathbf{H}^T \mathbf{Hx}_0 \\ &= Const_1 - \mathbf{x}^T [\mathbf{H}^T (\mathbf{r} - \mathbf{Hx}_0) + c \mathbf{x}_0] + \lambda \mathbf{1}^T \rho(\mathbf{x}) + \frac{c}{2} \|\mathbf{x}\|_2^2 \end{aligned} \quad (52)$$

A constante na Equação 52 contém todos os termos que são dependentes de \mathbf{r} e de \mathbf{x}_0 . Usando a definição que aparece na Equação 53:

$$\mathbf{v}_0 = \frac{1}{c} \mathbf{H}^T (\mathbf{r} - \mathbf{Hx}_0) + \mathbf{x}_0 \quad (53)$$

pode-se reescrever $\tilde{f}(\mathbf{x})$ (depois de dividir por c , que não possui impacto na meta de minimização) como (Equação 54):

$$\begin{aligned} \tilde{f}(\mathbf{x}) &= Const_2 - \mathbf{x}^T \mathbf{v}_0 + \frac{\lambda}{c} \mathbf{1}^T \rho(\mathbf{x}) + \frac{1}{2} \|\mathbf{x}\|_2^2 \\ &= Const_3 + \frac{\lambda}{c} \mathbf{1}^T \rho(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{v}_0\|_2^2 \end{aligned} \quad (54)$$

Como se vê, a solução dada a essa função de penalidade é descrita na Equação 55.

$$\mathbf{x}_{opt} = S_{\rho, \lambda/c}(\mathbf{v}_0) = S_{\rho, \lambda/c} \left(\frac{1}{c} \mathbf{H}^T (\mathbf{r} - \mathbf{Hx}_0) + \mathbf{x}_0 \right) \quad (55)$$

Isso é o minimizador global da função $\tilde{f}(\mathbf{x})$ na Equação 51. Até o momento, conseguiu-se converter a função original f em uma nova função \tilde{f} , para a qual pode-se obter uma expressão em formato fechado para seu minimizador global. Essa mudança da função objetivo depende da escolha do vetor \mathbf{x}_0 .

A ideia central de usar a função substituta é minimizar a função f de modo iterativo, produzindo a sequência de resultados $\{\mathbf{x}_i\}_k$, onde na $(k+1)$ -ésima iteração se minimiza \tilde{f} com a atribuição $\mathbf{x}_0 = \mathbf{x}_k$ (ELAD, 2010). A sequência de soluções $\{\mathbf{x}_i\}_k$ é comprovadamente convergente para o minimizador (local) da função original f . Assim, o método proposto é

simplesmente uma aplicação iterativa da Equação 56 na forma:

$$\mathbf{x}_{i+1} = S_{\rho, \lambda/c} \left(\frac{1}{c} \mathbf{H}^T (\mathbf{r} - \mathbf{H}\mathbf{x}_i) + \mathbf{x}_i \right) \quad (56)$$

Esse método é referido como sendo *Separable Surrogate Functionals* (ELAD, 2010), havendo a possibilidade da adição de uma busca em linha. Ao se escolher um valor de $\rho = 1$ ele é reduzido a uma versão plana do método *Separable Surrogate Functionals* (SSF), desenvolvida neste trabalho.

Reorganizando a Equação 56, como na Equação 57, temos μ como o tamanho do passo em direção ao mínimo. A opção da escolha de um μ ótimo no passo *Line Search* (LS) (MORÉ; THUENTE, 1994) acelera a convergência do método (LUENBERGER; YE, 2008), ao custo de se aumentar significativamente a complexidade computacional do método, já que envolve multiplicação de vetores e matrizes de ordem elevada. Duarte *et al.* (2016) citam que um limite superior para μ , tal que seu valor não ultrapasse o maior autovalor da matriz obtida por $\mathbf{H}^T \mathbf{H}$ e dividido por dois (Equação 57), pode ser encontrado na literatura.

$$\mathbf{x}_{i+1} = S_\lambda \left(\mathbf{x}_i + \mu [\mathbf{H}^T (\mathbf{r} - \mathbf{H}\mathbf{x}_i)] \right) \quad (57)$$

Distribuindo a Equação 57 obtém-se a Equação 58:

$$\mathbf{x}_{i+1} = S_\lambda \left(\mathbf{x}_i + \mu [\mathbf{H}^T \mathbf{r} - \mathbf{H}^T \mathbf{H} \mathbf{x}_i] \right) \quad (58)$$

Fazendo $\mathbf{H}^T \mathbf{H} = \mathbf{A}$, que forma a matriz de autocorrelação \mathbf{A} (TEIXEIRA; ANDRADE-FILHO; PERALVA, 2014), chega-se à Equação 59:

$$\mathbf{x}_{i+1} = S_\lambda \left(\mathbf{x}_i + \mu \begin{bmatrix} \underbrace{\mathbf{H}^T \mathbf{r}}_{\text{constante}} & - & \underbrace{\mathbf{A} \mathbf{x}_i}_{\substack{\text{varia em} \\ \text{cada iteração}}} \end{bmatrix} \right) \quad (59)$$

onde: a parte constante da Equação 59 ($\mathbf{H}^T \mathbf{r}$) pode ser calculada uma única vez na inicialização do método, enquanto a parte variável da Equação 59 ($\mathbf{A} \mathbf{x}_i$) é recalculada a cada iteração.

O Diagrama 10 (Apêndice C) representa a diagramação da Equação 59, conforme implementado neste trabalho.

A versão do SSF implementada nesta pesquisa pode ser visualizada no Algoritmo 7. É uma versão adaptada de Elad (2010), para o uso em janelas amostrais e para ignorar valores negativos, convertendo-os para zero.

Algoritmo 7: IMPLEMENTAÇÃO DO SEPARABLE SURROGATE FUNCTIONALS

Entrada: Tamanho da amostra, tamanho da janela, número de iterações, matriz H e sinal r

Saída: sinal \hat{x}

```

1 início
2    $A \leftarrow H^T H$ 
3    $\hat{x} \leftarrow inicializa$ 
4   para  $it \leftarrow (it + 1)$  até quantidadeDeAmostras faça
5      $passo \leftarrow ((it * janela) - (janela - 1))$ 
6      $s \leftarrow r(passo : passo + (janela - 1))$ 
7      $H^T s \leftarrow H^T * s$ 
8      $y_0 \leftarrow s(3 : bunch + 3)$ 
9     para  $i \leftarrow (i + 1)$  até númeroDeIterações faça
10     $B \leftarrow H^T s - (Ay_i)$ 
11     $y_i \leftarrow (y_i + \mu B) - \lambda$ 
12     $y_i(y_i < 0) \leftarrow 0$ 
13  fim
14   $y(y < 0) \leftarrow 0$ 
15   $\hat{x}(passo : passo + (janela - 1)) \leftarrow y$ 
16 fim
17 fim
18 retorna  $\hat{x}$ 

```

4.4.1.1 Separable surrogate functionals line search

Uma vez definida a base matemática que leva à implementação do SSF, tal qual a realizada neste trabalho, continuou-se o estudo de alternativas que pudesse apresentar melhores resultados. Com isso chegou-se à inclusão de uma busca em linha (*Line Search (LS)*) na implementação do método SSF original. Tal estudo é encorajado por Elad (2010), que apresenta as duas versões de SSF em sua obra.

O pseudocódigo contendo a versão do *SSF plus Least Square* (SSF+LS) se encontra no Algoritmo 8.

Algoritmo 8: IMPLEMENTAÇÃO DO SEPARABLE SURROGATE FUNCTIONALS COM LINE SEARCH

Entrada: Tamanho da amostra, tamanho da janela, número de iterações, matriz H e sinal r

Saída: sinal \hat{x}

```

1 início
2    $A \leftarrow H^T H$ 
3    $\hat{x} \leftarrow \text{inicializa}$ 
4    $\kappa \leftarrow \text{diagonal}(A)^{-1}$ 
5   para  $it \leftarrow (it + 1)$  até  $\text{quantidadeDeAmostras}$  faca
6      $\text{passo} \leftarrow ((it * \text{janela}) - (\text{janela} - 1))$ 
7      $s \leftarrow r(\text{passo} : \text{passo} + (\text{janela} - 1))$ 
8      $H^T s \leftarrow H^T * s$ 
9      $y_0 \leftarrow s(3 : bunch + 3)$ 
10    para  $i \leftarrow (i + 1)$  até  $\text{númeroDeIterações}$  faca
11       $B \leftarrow H^T s - (Ay_i)$ 
12       $\text{resíduo} \leftarrow (y_i + \kappa B) - \lambda$ 
13       $\text{resíduo}(\text{resíduo} < 0) \leftarrow 0$ 
14       $\text{resíduo} = \text{resíduo} - y_i$ 
15       $y_i \leftarrow y_i + (\mu * \text{resíduo})$ 
16    fim
17     $y(y < 0) \leftarrow 0$ 
18     $\hat{x}(\text{passo} : \text{passo} + (\text{janela} - 1)) \leftarrow y$ 
19  fim
20 fim
21 retorna  $\hat{x}$ 

```

4.4.1.2 Separable surrogate functionals line search com um valor de ν pré-definido

Uma terceira modificação realizada no método SSF original de Elad (2010) foi testada e implementada, como pode ser visto na Seção 5. Essa nova variação do método SSF possui um determinado valor de ν fixo aplicado na etapa de LS da variação SSF+LS. A utilização de um valor fixo para ν atua como uma constante que tange o limite do ponto de convergência da variação SSF+LS. Esse valor de ν é, ainda, escolhido para cada uma das taxas de ocupação utilizadas neste estudo. O Algoritmo 9 foi o utilizado como base, para tal escolha.

A essa terceira variação do método SSF convencionou-se chamar de *SSF plus LS with a ν constant* (SSF+LS ν) e está inteiramente desenvolvida durante os estudos que geraram esta tese. O pseudocódigo que contém essa variação SSF+LS ν se encontra no Algoritmo 10.

Algoritmo 9: MONTAGEM DA LISTA DE VALORES DE v

Entrada: Valor inicial de i e de j
Saída: Lista de valores de v

```

1 início
2   |   valoresDe v  $\leftarrow$  inicializa
3   |   para  $i \leftarrow 1$  até 101 faca
4   |   |   para  $j \leftarrow 1$  até 101 faca
5   |   |   |   se  $i <> j$  então
6   |   |   |   |    $v \leftarrow i/j$ 
7   |   |   |   |   se  $v$  não está em valoresDe v então
8   |   |   |   |   |   valoresDe v é acrecido de  $v$ 
9   |   |   |   |   fim
10  |   |   |   fim
11  |   |   fim
12  |   fim
13 fim

```

Algoritmo 10: IMPLEMENTAÇÃO DO SEPARABLE SURROGATE FUNCTIONALS COM LINE SEARCH E UMA CONSTANTE ν PRÉ-DEFINIDA

Entrada: Tamanho da amostra, tamanho da janela, número de iterações, matriz H e sinal r
Saída: sinal \hat{x}

```

1 início
2   |    $A \leftarrow H^T H$ 
3   |    $\hat{x} \leftarrow$  inicializa
4   |    $\nu \leftarrow$  buscarConstante(ocupação)
5   |   para  $it \leftarrow (it + 1)$  até quantidadeDeAmostras faca
6   |   |    $passo \leftarrow ((it * janela) - (janela - 1))$ 
7   |   |    $s \leftarrow r(passo : passo + (janela - 1))$ 
8   |   |    $H^T s \leftarrow H^T * s$ 
9   |   |    $y_0 \leftarrow s(3 : bunch + 3)$ 
10  |   |   para  $i \leftarrow (i + 1)$  até númeroDeIterações faca
11  |   |   |    $B \leftarrow H^T s - (Ay_i)$ 
12  |   |   |    $resíduo \leftarrow ((y_i + B) * \nu) - (\nu * \lambda)$ 
13  |   |   |    $resíduo(resíduo < 0) \leftarrow 0$ 
14  |   |   |    $resíduo = resíduo - y_i$ 
15  |   |   |    $y_i \leftarrow y_i + (\mu * resíduo)$ 
16  |   |   fim
17  |   |    $y(y < 0) \leftarrow 0$ 
18  |   |    $\hat{x}(passo : passo + (janela - 1)) \leftarrow y$ 
19  |   fim
20 fim
21 retorna  $\hat{x}$ 

```

4.4.2 Pré-inicialização do vetor y_0

O vetor y_0 é inicializado com os valores da respectiva janela de aquisição de dados (vetor s , representa uma janela do vetor r), antes do processo iterativo ser realizado de fato, no método definido pela Equação 59. Uma forma alternativa para a inicialização do vetor y_0 , através de um pré-processamento a ser realizado antes da etapa iterativa do algoritmo (AGUIAR *et al.*, 2019), será descrita nesta subseção.

A forma mais simples e usual (ELAD, 2010) de se obter uma solução aproximada de x é através da minimização da sua norma euclidiana quadrática (WAGNER, 2013), respeitando a restrição de manter o modelo. Tal simplicidade pode ser explicada porque a matriz H da Equação 24 não é uma matriz quadrada e, portanto, não possui inversa. Isso pode ser resumido pelo problema P_2 (Equação 60):

$$(P_2) : \min_x \|x\|_2^2 \text{ sujeito a } r = Hx \quad (60)$$

A resolução desse problema está na utilização de multiplicadores de Lagrange e seu respectivo Lagrangiano é definido pela expressão contida na Equação 61.

$$\mathcal{L}(x) = \|x\|_2^2 + \lambda^T(Hx - r) \quad (61)$$

Derivando $\mathcal{L}(x)$ em relação a x , tem-se (Equação 62):

$$\frac{\partial \mathcal{L}(x)}{\partial x} = 2x + H^T\lambda \quad (62)$$

cuja melhor solução para x , obtida através de cálculos matriciais é (Equação 63):

$$\hat{x}_{opt} = -\frac{1}{2}H^T\lambda \quad (63)$$

substituindo a solução na restrição que foi definida no problema (P_2) , tem-se a Equação 64:

$$H\hat{x}_{opt} = -\frac{1}{2}HH^T\lambda = r \Rightarrow \lambda = -2(HH^T)^{-1}r \quad (64)$$

e, por fim, substituindo λ na expressão de \hat{x}_{opt} , encontrada na Equação 63, é possível se chegar à Equação 65:

$$\hat{x}_{opt} = -\frac{1}{2}H^T\lambda = H^T(HH^T)^{-1}r = H^+r \quad (65)$$

onde: H^+ é definida como a pseudo-inversa de H (TEIXEIRA; ANDRADE-FILHO; PERALVA, 2014).

A proposta deste trabalho é inicializar o vetor x_0 como H^+s , uma vez que a matriz de convolução H é previamente conhecida em problemas de deconvolução. Esse pré-processamento tende a reduzir consideravelmente o número de iterações do método ao qual ele for aplicado. Essa inicialização será aplicada em todas as variações do método SSF, aqui estudadas, conforme

se apresenta na Seção 5.

O trecho de código escrito em pseucódigo no Algoritmo 11 (linhas 3, 4 e 7) será incluído nas implementações algorítmicas das variações do método SSF, aqui estudadas, para possibilitar a inclusão da inicialização do vetor \mathbf{x}_0 .

Algoritmo 11: INICIALIZAÇÃO DO VETOR y_0

Entrada: Tamanho da amostra, tamanho da janela, número de iterações, matriz \mathbf{H} e sinal \mathbf{r}

Saída: sinal $\hat{\mathbf{x}}$

```

1 início
2   :
3    $H^+ \leftarrow H^T * A^{-1}$ 
4    $\hat{H}^+ \leftarrow H^+(abs(valorElementos) > \varsigma)$ 
5   para  $it \leftarrow (it + 1)$  até  $quantidadeDeAmostras$  faz
6     :
7      $y_0 \leftarrow H^+ * s$ 
8     para  $i \leftarrow (i + 1)$  até  $númeroDeIterações$  faz
9       :
10      fim
11    :
12  fim
13 fim
14 retorna  $\hat{\mathbf{x}}$ 

```

4.4.3 Parallel-coordinate-descent

O algoritmo *Parallel-Coordinate-Descent* (PCD), desenvolvido por Elad, Matalon e Zibulevsky (2006) será descrito nesta subseção. Sua construção se inicia de uma simples decida coordenada, seguida pela mesclagem de um conjunto de passos de descida em um único passo mais fácil e levando ao método *Parallel-Coordinate-Descent Iterative-Shrinkage*. O raciocínio aqui desenvolvido pode lembrar o mesmo utilizado no algoritmo BCR de Sardy, Bruce e Tseng (2000), que lidou com um caso especial onde \mathbf{H} era uma união de matrizes unitárias.

Considerando a função original da Equação 47 (ELAD, 2010), é proposto um algoritmo de descida coordenada, que atualize uma entrada por vez em \mathbf{x} , enquanto mantém os demais elementos intocados. Uma sequência dessas rodadas de M passos (endereçando cada uma das entradas em $\mathbf{x} \in \mathbb{R}^M$) é necessariamente convergente. Bastante interessante, conforme será demonstrado, é que cada um desses passos é obtido por *shrinkage*. Assumindo que a solução atual é \mathbf{x}_0 , o que se pretende é atualizar a i -ésima entrada ao redor desse valor atual de $\mathbf{x}_0[i]$, provocando uma função unidimensional no formato da Equação 66:

$$g(z) = \frac{1}{2} \| \mathbf{r} - \mathbf{H}\mathbf{x}_0 - \mathbf{h}_i(z - \mathbf{x}_0[i]) \|_2^2 + \lambda\rho(z) \quad (66)$$

O vetor \mathbf{h}_i é a i-ésima coluna em \mathbf{H} . O termo $\mathbf{h}_i(z - \mathbf{x}_0[i])$ remove o efeito do valor antigo e adiciona um novo valor. Escrevendo de forma diferente e denotando $\tilde{\mathbf{r}} = \mathbf{r} - \mathbf{H}\mathbf{x}_0 + \mathbf{x}_0[i]\mathbf{h}_i$, transforma-se na função a seguir (Equação 67):

$$\begin{aligned}
g(z) &= \frac{1}{2} \|\tilde{\mathbf{r}} - \mathbf{h}_i z\|_2^2 + \lambda \rho(z) \\
&= \frac{1}{2} \|\tilde{\mathbf{r}}\|_2^2 - \tilde{\mathbf{r}}^T \mathbf{h}_i z + \frac{1}{2} \|\mathbf{h}_i\|_2^2 \cdot z^2 + \lambda \rho(z) \\
&= \|\mathbf{h}_i\|_2^2 \left(\frac{\|\tilde{\mathbf{r}}\|_2^2}{2 \|\mathbf{h}_i\|_2^2} - \frac{\mathbf{h}_i^T \tilde{\mathbf{r}}}{\|\mathbf{h}_i\|_2^2} z + \frac{z^2}{2} + \frac{\lambda}{\|\mathbf{h}_i\|_2^2} \rho(z) \right) \\
&= \|\mathbf{h}_i\|_2^2 \left(\frac{1}{2} \left(z - \frac{\mathbf{h}_i^T \tilde{\mathbf{r}}}{\|\mathbf{h}_i\|_2^2} \right)^2 + \frac{\lambda}{\|\mathbf{h}_i\|_2^2} \rho(z) \right) + Const
\end{aligned} \tag{67}$$

A função unidimensional obtida na Equação 66 tem, exatamente, a mesma forma da Equação 68, para casos unitários, nos quais a computação *shrinkage* provê o minimizador. Assim, a solução para se obter o z ótimo é dado pela Equação 69 (ELAD, 2010).

$$\begin{aligned}
f(\mathbf{x}) &= \frac{1}{2} \|\mathbf{x}_0 - \mathbf{x}\|_2^2 + \lambda \mathbf{1}^T \rho(\mathbf{x}) \\
&= \sum_{i=1}^m \left[\frac{1}{2} (\mathbf{x}_0[i] - \mathbf{x}[i])^2 + \lambda \rho(\mathbf{x}[i]) \right] \\
&= \sum_{i=1}^m g(\mathbf{x}[i], \mathbf{x}_0[i])
\end{aligned} \tag{68}$$

A minimização da função escalar da forma $g(\mathbf{x}, \mathbf{h}) = 0.5(\mathbf{x} - \mathbf{h})^2 + \lambda \rho(\mathbf{x})$, com respeito à \mathbf{x} , requer tanto zerar o gradiente (no caso da suavização $\rho(\mathbf{x})$), quanto mostrar o subgradiente de g contendo zero, para o caso não-diferencial.

$$\begin{aligned}
\mathbf{z}_{opt} &= S_{\rho, \lambda / \|\mathbf{h}_i\|_2^2} \left(\frac{\mathbf{h}_i^T \tilde{\mathbf{r}}}{\|\mathbf{h}_i\|_2^2} \right) \\
&= S_{\rho, \lambda / \|\mathbf{h}_i\|_2^2} \left(\frac{1}{\|\mathbf{h}_i\|_2^2} \mathbf{h}_i^T (\mathbf{r} - \mathbf{H}\mathbf{x}_0) + \mathbf{x}_0[i] \right)
\end{aligned} \tag{69}$$

Esse algoritmo deve funcionar bem em casos com poucas dimensões, no entanto, ele é impraticável em problemas geralmente com muitas dimensões, por exemplo: quando a matriz \mathbf{H} não é informada explicitamente e somente multiplicações por ela e sua adjunta são possíveis. Assim, como esse algoritmo requer uma extração de colunas de \mathbf{H} uma a uma, ele se torna proibitivo (ELAD, 2010).

Considerando uma modificação nesse método que supere o problema colocado e confiando

na propriedade de que "ao minimizar uma função, se houver várias direções de descida, então qualquer combinação não negativa delas é também uma direção de descida" (ELAD, 2010, p. 122), o que se propõe é uma combinação linear simples do conjunto de todos os passos acima na Equação 69. Como cada uma dessas etapas manipula uma entrada no vetor de destino, pode-se escrever essa soma como na Equação 70.

$$\begin{aligned} \mathbf{v}_0 &= \sum_{i=1}^m \mathbf{e}_i \cdot S_{\rho, \lambda / \|\mathbf{h}_i\|_2^2} \left(\frac{1}{\|\mathbf{h}_i\|_2^2} \mathbf{h}_i^T (\mathbf{r} - \mathbf{Hx}_0) + \mathbf{x}_0[i] \right) = \\ &= \begin{bmatrix} S_{\rho, \lambda / \|\mathbf{h}_1\|_2^2} \left(\frac{1}{\|\mathbf{h}_1\|_2^2} \mathbf{h}_1^T (\mathbf{r} - \mathbf{Hx}_0) + \mathbf{x}_0[1] \right) \\ \vdots \\ S_{\rho, \lambda / \|\mathbf{h}_m\|_2^2} \left(\frac{1}{\|\mathbf{h}_m\|_2^2} \mathbf{h}_m^T (\mathbf{r} - \mathbf{Hx}_0) + \mathbf{x}_0[m] \right) \end{bmatrix} \quad (70) \end{aligned}$$

Na expressão anterior (Equação 70), \mathbf{e}_i é um vetor m -dimensional trivial, com zeros em todas as entradas e 1 na i -ésima entrada. Reescrevendo essa expressão tem-se a seguinte fórmula simplificada (Equação 71):

$$\mathbf{v}_0 = S_{\rho, \text{diag}(\mathbf{H}^T \mathbf{H})^{-1} \lambda} (\text{diag}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T (\mathbf{r} - \mathbf{Hx}_0) + \mathbf{x}_0) \quad (71)$$

O termo $\text{diag}(\mathbf{H}^T \mathbf{H})$ consiste nas normas das colunas do dicionário de \mathbf{H} . Ele é usado tanto para ponderação do erro de retroprojeção $\mathbf{H}^T (\mathbf{r} - \mathbf{Hx}_0)$, como para o operador de *shrinkage* (ELAD, 2010). Esse conjunto de pesos pode ser computado *off-line*, antes do algoritmo começar. Observa-se que a fórmula obtida não exige uma extração de colunas de \mathbf{H} , como na Equação 69. Para além, a operação necessária é uma multiplicação direta de \mathbf{H} e seus adjuntos por vetores, como visto na Equação 56.

Enquanto cada uma das direções CD tem a decida garantida, sua combinação linear não é necessariamente descendente sem um dimensionamento adequado. Assim, considerou-se essa direção e realizou-se uma pesquisa em linha (*Line Search - LS*) ao longo dela. Isso significa que o algoritmo iterativo atual tem a forma apresentada pela Equação 72 (ELAD, 2010).

$$\begin{aligned} \mathbf{x}_{i+1} &= \mathbf{x}_i + \mu (\mathbf{s}_i - \mathbf{x}_i) = \\ &= \mathbf{x}_i + \mu (S_{\rho, \text{diag}(\mathbf{H}^T \mathbf{H})^{-1} \lambda} (\text{diag}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T (\mathbf{r} - \mathbf{Hx}_i) + \mathbf{x}_i) - \mathbf{x}_i) \end{aligned} \quad (72)$$

Fazendo $\text{diag}(\mathbf{H}^T \mathbf{H})^{-1} = \mathbf{c}$, tem-se (Equação 73).

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mu(S_{\rho, c\lambda}(\mathbf{c}\mathbf{H}^T(\mathbf{r} - \mathbf{H}\mathbf{x}_i) + \mathbf{x}_i) - \mathbf{x}_i) \quad (73)$$

Distribuindo a Equação 73 obtém-se a Equação 74.

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mu(S_{\rho, c\lambda}(\mathbf{c}\mathbf{H}^T\mathbf{r} - \mathbf{c}\mathbf{H}^T\mathbf{H}\mathbf{x}_i + \mathbf{x}_i) - \mathbf{x}_i) \quad (74)$$

Fazendo, $\mathbf{H}^T \mathbf{H} = \mathbf{A}$, para formar a matriz de autocorrelação \mathbf{A} , e anulando o vetor constante \mathbf{c} nos dois lados da subtração, chega-se à Equação 75:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mu \left(S_{\rho, c\lambda} \begin{bmatrix} \underbrace{\mathbf{H}^T \mathbf{r}}_{\text{constante}} & - & \underbrace{\mathbf{A}\mathbf{x}_i + \mathbf{x}_i}_{\substack{\text{varia em} \\ \text{cada iteração}}} \end{bmatrix} - \mathbf{x}_i \right) \quad (75)$$

onde: a parte constante da Equação 75 ($\mathbf{H}^T \mathbf{r}$) pode ser calculada uma única vez na inicialização do método, enquanto a parte variável da Equação 75 ($\mathbf{A}\mathbf{x}_i + \mathbf{x}_i$) é recalculada a cada iteração. O vetor \mathbf{c} é obtido da diagonal de uma matriz cujos valores são constantes, para este trabalho, formando assim uma diagonal contendo todos os valores iguais. Optou-se, portanto, pela substituição desse vetor por um único valor, agora representado pela constante κ .

Com μ escolhido pelo algoritmo de LS de alguma forma, significa que é obrigatória a realização da otimização da função unidimensional (Equação 76):

$$h(\mu) = \frac{1}{2} \|\mathbf{r} - \mathbf{H}(\mathbf{x}_i + \mu(\mathbf{s}_i - \mathbf{x}_i))\|_2^2 + \lambda \mathbf{1}^T \rho(\mathbf{x}_i + \mu(\mathbf{s}_i - \mathbf{x}_i)) \quad (76)$$

que requer apenas mais duas multiplicações de \mathbf{H} por \mathbf{s}_i e \mathbf{x}_i . Este processo global será referido a seguir como o algoritmo *Parallel-Coordinate-Descent* (ELAD, 2010).

Como se pode ver, comparado com o SSF na Equação 56, o método aqui derivado é diferente em duas questões:

- as normas das menores partes de \mathbf{H} desempenham um papel importante na ponderação do erro retrocedido, enquanto que o algoritmo anterior usa uma constante;
- o novo algoritmo requer uma pesquisa em linha (LS) para obter uma descida.

Nas ilustrações a e b do Diagrama 11 (Apêndice C) pode-se observar como essas diferenças são sutis.

Mantendo a base de construção dos algoritmos utilizada neste trabalho, observa-se no Algoritmo 12 a implementação da Equação 75. Esse Algoritmo já está adaptado para o uso em janelas amostrais e ignorando os valores negativos, substituindo-os por zero.

Algoritmo 12: IMPLEMENTAÇÃO DO PARALLEL COORDINATE DESCENT

Entrada: Tamanho da amostra, tamanho da janela, número de iterações, matriz H e sinal r

Saída: sinal \hat{x}

```

1 início
2    $A \leftarrow H^T H$ 
3    $\hat{x} \leftarrow inicializa$ 
4    $\kappa \leftarrow diagonal(A)^{-1}$ 
5   para  $it \leftarrow (it + 1)$  até quantidadeDeAmostras faça
6      $passo \leftarrow ((it * janela) - (janela - 1))$ 
7      $s \leftarrow r(passo : passo + (janela - 1))$ 
8      $H^T s \leftarrow H^T * s$ 
9      $y_0 \leftarrow s(3 : bunch + 3)$ 
10    para  $i \leftarrow (i + 1)$  até númeroDeIterações faça
11       $B \leftarrow H^T s - (Ay_i)$ 
12       $resíduo \leftarrow y_i + B - (\kappa * \lambda)$ 
13       $resíduo(resíduo < 0) \leftarrow 0$ 
14       $resíduo = resíduo - y_i$ 
15       $y_i \leftarrow y_i + (\mu * resíduo)$ 
16    fim
17     $y(y < 0) \leftarrow 0$ 
18     $\hat{x}(passo : passo + (janela - 1)) \leftarrow y$ 
19  fim
20 fim
21 retorna  $\hat{x}$ 

```

4.5 CONCLUSÃO DA SEÇÃO 4

Uma vez que o desenvolvimento teórico/matemático dos novos métodos estudados já foram apresentados, este trabalho prossegue com a apresentação de análises comparativas entre eles, formando a base para os demais desenvolvimentos.

5 ANÁLISES DOS MÉTODOS APRESENTADOS

Uma vez concluído o embasamento teórico/matemático dos métodos apresentados nas seções anteriores, cabe, agora, focar na aplicação desses métodos em um conjunto de dados de simulação, pois o nível de empilhamento de sinais estudado nesta Tese só estará presente depois da entrada em operação da Fase II de atualização do experimento ATLAS. Portanto, ainda não possuímos os dados reais a serem trabalhados. Com isso, nesta seção, serão realizadas as análises dos métodos estudados até aqui, de forma a embasar a implementação do método mais bem adaptado ao ambiente no qual este estudo está inserido.

Um *Toy Monte Carlo Simulation* foi utilizado para que tais análises fossem realizadas. Ele é capaz de simular sinais com diversas características, inclusive as de empilhamento de sinais (*pile-up*), fornecendo, assim, comportamentos similares aos que podem ser encontrados em Monte Carlo para o TileCal (CHAPMAN, 2011). Um maior controle sobre as mais importantes características do sinal possibilita uma avaliação melhor dos métodos.

A utilização desse *Toy* também se justifica porque, até o momento, o TileCal não possui um simulador completo da nova eletrônica *free-running*, a ser implantada durante a Fase II de atualização do experimento ATLAS. Desse modo tornou-se imprescindível a utilização de um simulador que possibilitasse o nível de controle das características dos sinais utilizados nesta pesquisa, situação proporcionada pelo *Toy Monte Carlo Simulation*.

5.1 BANCO DE DADOS DAS SIMULAÇÕES

O desempenho, sob diversas condições, dos métodos estudados foi considerado, para favorecer a análise do seu comportamento em relação a dois aspectos:

- a. detecção – através da análise das curvas *Receiver Operating Characteristic* (ROC) (FAW-CETT, 2006), geradas a partir do desenho dos valores de falsos alarmes (do inglês *False Alarm* (FA)) no eixo x e dos valores de probabilidade de detecção (do inglês *Detection Probability* (DP)) no eixo y , sendo tais valores variando de 0 até 1. A análise da relação entre FA e DP possibilita saber o quanto o método em análise é eficiente em detectar quais BC tiveram deposição de energia;
- b. estimativa – efetuada a partir da análise do valor do *Root Mean Square* (RMS) do erro entre o sinal alvo (vetor \mathbf{x}) e o sinal recuperado pelo método em análise (vetor $\hat{\mathbf{x}}$), sinal esse recuperado a partir do sinal medido (vetor \mathbf{r}). Dessa forma, é possível aferir a eficiência do método em análise, em relação a sua assertividade de estimativa da energia depositada em cada uma das células do calorímetro.

A diferença do valor de RMS da distribuição de erro entre o sinal alvo e o sinal recuperado, a partir do sinal medido, foi aqui nomeada de *RMS recuperado* (RMSr), para fins de abreviação do significado completo (ou seja, RMSr é igual ao valor da diferença do valor de RMS da distribuição de erro entre o sinal alvo e o sinal recuperado, a partir do sinal medido). A fórmula para o cálculo do RMS pode ser visualizada na Equação 77 e, conforme descrito nas seções

anteriores, o sinal recuperado é representado pelo vetor \hat{x} e o sinal alvo é representado pelo vetor x . O número de pulsos amostrais está representado por Υ .

$$RMSError = \sqrt{\frac{\sum_{i=0}^{\Upsilon-1} (\hat{x}[i] - x[i])^2}{\Upsilon}} \quad (77)$$

Os bancos de dados gerados para a avaliação desses dois aspectos possuem sinais com características comuns, como: os pulsos característicos do TileCal, o valor médio da distribuição exponencial (PERALVA *et al.*, 2013) – na qual é realizado o sorteio da amplitude dos pulsos característicos – e a natureza do ruído de fundo.

Tais sinais são diferenciados por suas taxas de ocupação, que podem variar de 1% até 90% (neste trabalho as variações testadas foram: 1%, 5%, 10%, 20%, 30%, 40%, 50%, 60% e 90%). Nesse contexto, a taxa de ocupação pode ser entendida como a probabilidade de ocorrer a deposição de energia em uma determinada célula, no momento da colisão de duas partículas. Uma taxa de ocupação de 50% equivale dizer que, a cada duas colisões ocorreu uma deposição de energia em determinada célula.

A geração de um banco de dados de sinais, que possuísse essa variação de taxas de ocupação, foi necessária para a avaliação dos métodos em diferentes cenários de ocorrências de *pile-up*. O aumento da taxa de ocupação está diretamente relacionado com o aumento de luminosidade do canal e é inversamente relacionado com a esparsidade. O aumento da luminosidade está previsto para as atualizações que ocorrerão nos próximos anos no LHC (CERN, 2014). Com o uso desse conjunto de taxas de ocupação é possível a avaliação de métodos, que, ao mesmo tempo, sejam tanto tolerantes ao efeito de *pile-up* quanto robustos a variações de esparsidade.

5.1.1 Ruído incoerente

Existem duas fontes principais que contribuem para o ruído incoerente: o ruído térmico e o ruído de *pile-up*. O ruído térmico tem um espectro branco e depende das características do detector e da eletrônica de processamento de sinais, como amplificadores e cabos. O ruído de *pile-up* é o resultado do bombardeio constante de células do calorímetro por processos físicos, ocorridos com seções transversais altas, devido às múltiplas interações que ocorrem em cada BC (DANIELLS, 2016).

Outra fonte de ruído decorre da quantificação analógica dos sinais do calorímetro, que introduz um erro devido ao truncamento realizado pelos ADCs. Esse ruído de quantização tem uma distribuição uniforme de E_T e é calculado para ser de aproximadamente 75MeV, dado pela regra *Least Significant Bit* (LSB)/ $\sqrt{12}$, para contagens com um LSB de 0 : 25GeV, (Bennett, 1948).

Neste projeto, apenas os ruídos térmico e de *pile-up* foram considerados, pois os outros tipos de ruído, incluindo o ruído coerente e *crosstalk*, são de importância secundária para os fins

desta investigação. Portanto, não são considerados em decorrência de suas magnitudes serem pequenas.

Um *Toy Simulation*, chamado ToyMC, foi criado para modelar as condições de ruído esperadas para uma célula, que seja atingida por jatos de energia em qualquer ângulo $|\eta|$. Isso permite explorar adequadamente uma ampla gama de capacidades possíveis com o *new Multi-Chip Module* (nMCM). O simulador ToyMC modela o histórico completo das entradas de sinal analógico em circulação no LHC. Essa modelagem acontece com uma frequência de amostragem igual à frequência de BC. O simulador também é capaz de produzir os esquemas esperados de enchimento de *bunches* e perfis de luminosidade de *bunches*, para simular com precisão os efeitos do *pile-up in-time* e *out-of-time* sobre o desempenho do nMCM.

Esse ToyMC será mais bem detalhado, a seguir, e será utilizado para a geração dos dados para as análises realizadas nesta seção.

5.1.2 O pulso característico do TileCal e o desvio de fase

O pulso característico do TileCal corresponde à resposta impulsiva, que é digitalizada com sete amostras (amplitude máxima unitária), totalizando 150ns de amostragem a uma taxa de 40MHz, ou 25ns por amostra (AAD *et al.*, 2008a). É necessário que haja uma calibração para cada canal, porque pode ocorrer uma variação desta resposta, de canal para canal. No Gráfico 7 (Apêndice A) estão representadas as sete amostras do pulso característico de um determinado canal do TileCal, sendo a amostra central correspondente ao pico, sem desvio de fase e normalizadas. Os pontos intermediários são obtidos a partir da interpolação das sete amostras.

A análise de cada evento de colisão considerou, também, a diferença nos tempos de voo - *time of flight* (ToF) – das partículas (CLÉMENT *et al.*, 2008) , permitindo observar erros de sincronismo na digitalização. Tais erros são provocados pelo desvio de fase no momento da aquisição do sinal. Aqui, considerou-se a possibilidade desse tipo de erro com uma distribuição uniforme em uma faixa de $\pm 1\text{ns}$, para o sinal oriundo do TileCal.

5.1.3 Geração do banco de dados para as análises

O banco de dados utilizado nas análises realizadas é gerado seguindo os passos descritos abaixo:

- a. geração de Y amostras consecutivas de ruído branco e gaussiano com $\sigma = 1,3 \text{ ADC counts}$, representando o ruído eletrônico digitalizado de um canal do TileCal a 40MHz (PERALVA *et al.*, 2013);
- b. realização de um sorteio de posições (dos BC), onde ocorreram as deposições de energia, baseado na taxa de ocupação desejada. Por exemplo: uma taxa de ocupação de 50% significa que uma a cada duas colisões, em média, terão deposição de energia na célula em análise;
- c. realização de um sorteio das amplitudes (valor de energia) em uma distribuição exponencial.

O valor médio dessa distribuição é de 30 unidades de conversão do conversor ADC (DUAARTE, 2015). Esse valor foi retirado de um *fitting* exponencial feito na distribuição da deposição de energia no TileCal. Assim, a média da distribuição influencia diretamente na relação sinal-ruído;

- d. realização de um sorteio para a simulação do desvio de fase do pulso a ser gerado, utilizando uma distribuição uniforme de ± 1 ns (CLÉMENT *et al.*, 2008), uma vez já sorteados a amplitude da energia e os BC;
- e. ajuste do simulador utilizado para um padrão de colisão equivalente a $48b7e$, ou seja: a cada 48 colisões, onde cada intervalo de tempo equivale a 25ns, outros 7 intervalos de tempo (também de 25ns cada) ficavam vazios sem nenhuma colisão e, portanto, um valor zero era atribuído a cada um desses espaços sem colisão (DAMERAU *et al.*, 2018). Esse padrão foi utilizado por ser o mais presente nas colisões ocorridas em 2018, antes do desligamento para as manutenções preparativas para a Fase II de atualizações do ATLAS, e deve continuar a ser utilizado nas próximas tomadas de dados;
- f. sobreposição dos pulsos característicos, com suas amostras centrais localizadas nos BC sorteados e com as respectivas energias e desvio de fase;
- g. conversão das métricas utilizadas, passando de contagens de ADC (*ADC counts*) para Megaelétrons-volts (MeV) do TileCal, multiplicando os valores em *ADCcounts* por 12 (CLEMENT; KLIMEK, 2011; KLIMEK, 2012), para aproxima-las das métricas utilizadas no TileCal.

Além disso, a eletrônica do TileCal possibilita que sinais de até 10bits (HARKUSHYA; COLLABORATION, 2016) sejam gerados, com valores variando de -511 a 512. A parte negativa é pouco utilizada, pois ela só faz parte do ruído eletrônico. Por isso, os valores finais são arredondados para representação em números inteiros, de ponto fixo.

Optou-se por utilizar 1.820 janelas amostrais de 55 pulsos cada ($48b7e$). Com isso, cada método analisado foi aplicado a um sinal finito composto por $Y = 100.100$ pulsos.

A sequência com o valor correto das energias sorteadas é chamada de sinal alvo (vetor x). O sinal gerado pelo *ToyMC Simulation* utilizado é chamado de sinal medido (vetor r). Há, ainda, um terceiro sinal que é o sinal recuperado pelos métodos analisados a partir do sinal medido. Esse sinal recuperado é nomeado de acordo com o método em análise (vetor \hat{x}).

5.1.3.1 Toy Monte Carlo

A base teórica do ToyMC utilizado nesta obra, pode ser encontrada no trabalho de Daniells (2016). Seu desenvolvimento é fruto dos mesmos trabalhos anteriores realizados pelo grupo de pesquisa a que ela está inserida, com grande contribuição dos trabalhos de Duarte (2015) e Peralva (2012).

O ToyMC permite a simulação de várias características dos sinais, conforme indica Daniells (2016). A vantagem dessa simulação sobre os dados experimentais reais é que torna-se fácil a produção de grandes amostras estatísticas sob condições específicas, para comparar com

precisão o desempenho de diferentes configurações e características do sinal.

O ToyMC que foi implementado adiciona um sinal impulsivo em cada BC preenchido, com uma amplitude escolhida aleatoriamente por distribuições de amostragem baseadas na luminosidade do *bunch*. A forma do pulso do calorímetro, para essa célula, é multiplicada por essa amplitude e o resultado é adicionado ao pedestal, onde o pico do pulso é centrado no BC corrente. O processo é repetido para construção do sinal de entrada analógica de uma circulação completa no LHC e para ser processado pela emulação do sinal (DANIELLS, 2016).

O ruído de *pile-up* simulado com um espaçamento de 25ns foi medido no ToyMC, a partir do desvio-padrão do sinal analógico de entrada obtido, para os cruzamentos de um *bunch* cheio, considerando os efeitos do deslocamento médio da linha de base.

O formato do pulso usado pelo ToyMC, segundo Daniells (2016), foi adquirido por um osciloscópio com taxa de amostragem de 1GHz, para modelar corretamente os efeitos de desvio de fase esperados nos dados. Os sinais são registrados com uma precisão de 1ns, que permite a modelagem detalhada do *jitter*. O ToyMC pressupõe que as amplitudes de pulso serão dimensionadas linearmente com a energia, sem a ocorrência de saturação. Ainda segundo (DANIELLS, 2016), usar um esquema de preenchimento representativo do LHC na simulação é um requisito importante, a fim de modelar adequadamente os efeitos do empilhamento, como resultado dos comprimentos finitos do trem de *bunches* e do espaçamento entre eles. O Gráfico 6 (Apêndice A) mostra o esquema de enchimento escolhido para estudo no ToyMC, que se baseia na estrutura geral do trem utilizada em 2018. Trens de 48 *bunches* são agrupados em lotes de três ou dois e separados por um pequeno espaço de 7 *bunches*. Cada lote é separado por um longo intervalo de 31 *bunches* sem colisão no padrão 3-4-2 do número de trens de *bunches*, até que o intervalo de abortamento seja atingido (KOSTOGLOU *et al.*, 2020).

A combinação das formas de pulso do calorímetro e do esquema de enchimento resulta em deslocamentos na linha de base média, dentro de um trem de *bunches*, em relação ao valor nominal do pedestal. Esses deslocamentos são proporcionais ao número médio de interações por BC.

Os deslocamentos médios de pedestal para as células do TileCal, em diferentes regiões do calorímetro, são mostrados no Gráfico 11 (Apêndice D), para luminosidades uniformes e realistas de *bunches* correspondentes a $\langle \mu \rangle = 40$ (DANIELLS, 2016), para um trem de *bunches* com colisão seguindo após um longo intervalo de *bunches* sem colisão e, ainda, um segundo trem que segue após um curto intervalo de *bunches* sem colisão. No ToyMC o pedestal nominal é de 31,5 ADC *counts* sem a presença de colisões.

5.2 OS MÉTODOS QUE JÁ FORAM PROPOSTOS

As análises presentes nesta subseção dizem respeito aos métodos já estudados em trabalhos anteriores (BALABRAM-FILHO, 2014; BARBOSA, 2012; BARBOSA, 2017; DUARTE, 2015; PERALVA, 2012; PEREIRA, 2014), servindo como sustentáculo para a comparação das análises dos métodos aqui estudados.

O filtro casado (PALKA, 2016) seguido de um detector de picos foi utilizado durante a *Run 1* do LHC, entre 2011 e 2012, no *Level 1* do sistema de *trigger* do TileCal, conforme já esclarecido na Subsubseção 2.9.1.2. Foi analisado, aqui, para que uma linha comparativa em relação à evolução dos métodos em uso no *Level 1* do sistema de *trigger* do TileCal pudesse ser traçada.

Já o filtro FDIP é uma evolução dos filtros baseados em FIR, cujo uso iniciou-se entre as *Runs 1 e 2* do LHC – de 2013 a 2014 – e cujos testes continuam, inclusive, para a Fase II de atualizações dos experimentos ATLAS e CMS (ANDREI, 2012; DUARTE *et al.*, 2016; PALKA, 2016). A determinação dos coeficientes do filtro FDIP utilizado se deu a partir do estudo realizado entre a relação do número total de coeficientes em teste e o menor valor para RMS resultante da diferença entre o sinal obtido na saída do filtro e o sinal de controle¹³, sem a adição de ruído. Os resultados podem ser observados no Gráfico 12 (Apêndice D), para cada uma das taxas de ocupação consideradas neste trabalho.

Continuando a análise do Gráfico 12 (Apêndice D), observa-se que a partir de uma ordem inferior a 20 há uma estabilidade nos valores de RMSr. Mesmo assim optou-se por se utilizar a ordem 26, para as comparações realizadas utilizando o Filtro FDIP, garantindo o menor valor de RMSr alcançável pelo Filtro FDIP, independente da taxa de ocupação considerada.

Diferentemente dos filtros Casado e FDIP, que foram projetados para utilização no *Level 1* do sistema de *trigger* do TileCal, o método DS foi idealizado inicialmente para ser utilizado no *Level 2* (BARBOSA, 2017) do sistema de *trigger* do TileCal. Neste caso, foi adaptado para ser aplicado no *Level 1*. Apesar de esse método possuir uma alta complexidade computacional, ele será usado como comparação aos métodos esparsos propostos nesta tese.

O método GD, foi analisado por ser um método já proposto pelo grupo de pesquisa relacionado a este estudo e também pelo viés de transição apresentado entre os métodos já propostos e os novos métodos sugeridos. Esse método possui características comuns a ambos, como a utilização da teoria de convolução – também utilizada no Filtro FDIP – e o janelamento de sinais utilizado pelos novos métodos aqui apresentados. Ele foi aplicado 330 vezes no sinal aqui trabalhado, para que as análises fossem realizadas. Essas aplicações equivalem a iterações do algoritmo, ou seja: foram necessárias cerca de 330 iterações do método GD, para que os resultados apresentados fossem obtidos (DUARTE *et al.*, 2019).

O sinal referido nesta e nas próximas seções é o sinal obtido pelo ToyMC (subsubseção 5.1.3.1), simulando o sinal obtido pelo TileCal, conforme a configuração definida na subseção 5.1.3.

A eficiência da detecção dos métodos analisados pode ser comparada através das análises das curvas ROC, presentes no Apêndice D, na Seção D.1. Essas curvas ROC foram obtidas através da variação do patamar e, conforme se verifica, a eficiência de detecção do DS supera a dos demais métodos.

Ademais, à medida que a taxa de ocupação aumenta, os resultados de detecção do Filtro

¹³Métrica essa cujo acrônimo RMSr é utilizado nesse trabalho.

FDIP e do GD se aproximam do obtido com o DS. É a maior tolerância ao efeito de *pile-up* desses métodos, em relação ao Filtro Casado, que permite essa aproximação.

Ainda a respeito das curvas ROC, pode-se notar que os gráficos referentes ao Filtro Casado possuem baixos valores de DP. Isso se deve ao uso do detector de picos, que seleciona apenas a amostra central e, à medida que a taxa de ocupação e a incidência de *pile-up* aumentam, o detector de picos começa a falhar, refletindo em uma diminuição ainda maior dos valores de DP. Conforme as imagens do Gráfico 19 (Apêndice D na Seção D.1) demonstram, existe um valor de patamar onde se pode alcançar um valor mínimo de RMSr. Tal ponto é o de maior eficiência do método em análise e encontra-se destacado nas imagens.

Na Tabela 1, verificam-se todos os valores de RMSr onde o patamar é zero, pois, apesar de ser necessária a variação de patamar para se obter as curvas ROC, os métodos aqui considerados não possuem a definição de patamares em seus algoritmos. O valor de patamar igual a zero foi utilizado, porque patamares inferiores a zero podem considerar valores negativos de energia, o que não faz parte dos valores realísticos de medida de energia.

Tabela 1 - Comparaçāo entre os mētods Filtro Casado, Filtro FDIP, DS e GD, valoress em MeV

Taxa de ocupação	MF	FDIP	DS	GD
1%	14,19	19,32	9,75	24,48
5%	25,24	26,87	13,21	27,10
10%	34,91	30,94	16,63	30,05
20%	56,47	36,64	22,92	35,30
30%	79,54	40,44	28,32	39,29
40%	102,73	43,65	33,86	42,99
50%	125,62	46,13	39,08	45,96
60%	148,49	48,49	44,89	48,68
90%	212,86	54,15	59,26	55,64

Nota(s): Valores em MeV.

Fonte: Elaborada pelo autor (2021).

O mētido DS é mais tolerante ao efeito de *pile-up*, uma vez que obtêm os menores valoress de RMSr, quando relacionado a todos os outros mētods considerados (Tabela 1). O filtro FDIP consegue um valor de RMSr menor que o obtido pelo mētido DS, somente quando a taxa de ocupação atinge 90%. Aliás, o mētido DS chega a ser pior que o mētido GD, inclusive, à taxa de ocupação de 90%. O mētido DS, portanto, só apresenta desempenho melhor que o Filtro Casado. Este fato é esperado, pois DS é um mētido baseado na esparsidade dos dados, e a esparsidade diminui em valoress elevados de ocupação do canal.

5.3 MÉTODOS GREEDY

A análise dos métodos *Greedy*, estudados na Seção 4, será feita comparando-os com as análises efetuadas na seção 5.2. É a mesma metodologia já adotada anteriormente e que será utilizada em todo este trabalho.

A escolha de um patamar de resíduo é intrínseco aos métodos *Greedy* e, por isso, foi aqui analisado para a construção das métricas desses métodos. Além disso, o sinal trabalhado aqui é o mesmo da seção 5.2, conforme a configuração definida na Subseção 5.1.3.

A partir das análises das curvas ROC, presentes no Apêndice D na Seção D.2, pode-se comparar a eficiência da detecção dos métodos *Greedy*, confrontando-os entre si. Cabe lembrar que, no caso dos métodos *Greedy*, o patamar utilizado é o patamar do resíduo do próprio método em análise. Nos gráficos do Apêndice D na Seção D.2 pode ser observado que, ainda para baixas taxas de ocupação, o método LS-OMP já começa a se destacar dos métodos *Greedy*. Essa distância aumenta à medida que a taxa de ocupação aumenta com o método LS-OMP, destacando-se cada vez mais dos outros métodos *Greedy*, aqui analisados.

A curva ROC dos métodos *Greedy* se deforma à medida que vai ocorrendo a diminuição da esparsidade, à proporção que a taxa de ocupação aumenta. Isso evidencia as falhas em tais métodos, por serem fortemente dependentes da esparsidade. Do ponto de vista prático, porém, a incidência de altas taxas de ocupação (acima de 30%) são observadas em situações muito específicas, que ocorrem com uma certa raridade e, por isso, ainda é viável o uso desses métodos.

Em relação a análise da estimativa da energia, via utilização do RMSr, pode-se observar nas imagens do Gráfico 25 (Apêndice D na Seção D.2) a existência de um ponto em destaque, onde o valor de patamar do resíduo aponta para o menor valor de RMSr (destacado na Tabela 2). Trata-se de um ponto específico para cada uma das taxas de ocupação aqui utilizadas.

Tabela 2 - Comparação entre os métodos *Greedy*

Taxa de ocupação	MP		OMP		LS-OMP	
	Patamar	RMSr	Patamar	RMSr	Patamar	RMSr
1%	106,65	6,67	101,82	3,26	111,92	2,85
5%	74,94	15,37	97,48	8,77	108,66	7,28
10%	53,66	21,34	85,69	13,95	115,10	11,80
20%	44,89	29,39	82,26	23,27	108,66	19,36
30%	39,79	35,99	69,97	30,79	92,95	27,12
40%	33,94	42,81	69,97	37,26	85,69	33,44
50%	26,83	50,09	62,35	43,24	75,89	39,91
60%	29,39	59,90	56,28	48,17	68,93	45,66
90%	29,39	101,29	0,0	57,67	0,0	57,67

Nota(s): valores em MeV.

Fonte: Elaborada pelo autor (2021).

Mesmo com a baixa tolerância dos métodos *Greedy* à diminuição da esparsidade, para essa métrica, o método que mais se destaca para altas taxas de ocupação e, portanto, para um ambiente de maior incidência de *pile-up* (menor esparsidade) é o LS-OMP. Na Tabela 2 são visualizados os pontos de patamar de resíduo onde foram registrados os menores valores de RMSr, em cada uma das taxas de ocupação aqui estudadas. Na mesma tabela, também se percebe que o método LS-OMP foi o que obteve os menores valores de RMSr, dentre todos os outros métodos *Greedy* considerados.

5.4 MÉTODOS SHRINKAGE

A análise foco desta seção são os métodos *Shrinkage*, mencionados na Seção 4. Para proceder com essa análise, em primeiro lugar, foi aprofundado o estudo de parâmetros do método SSF e de parâmetros específicos às variações aqui estudadas. São eles:

- a. λ é o multiplicador de Lagrange, presente na Equação 34, responsável pelo equilíbrio das normas ℓ_2 e ℓ_1 , onde a primeira lida com a representação dos dados em meio ao ruído (ℓ_2), enquanto a segunda busca a esparsidade (ℓ_1) (ELAD, 2010). Pode-se notar que, ao parâmetro λ tender a zero, há uma predominância da norma ℓ_2 , tornando assim o método mais tolerante ao ruído e, consequentemente, ao efeito de *pile-up*;
- b. μ esse parâmetro representa o tamanho do passo em direção ao mínimo e está presente na Equação 57. O que se buscou foi a definição do melhor valor para esse parâmetro;
- c. ν é o parâmetro específico da variação SSF+LS ν , onde ele atua como uma constante que tange o limite do ponto de convergência da variação SSF+LS;
- d. ω representa o número de elementos não nulos da matriz \mathbf{H}^+ (formada pela pseudo-inversa da matriz \mathbf{H}). O número de elementos é obtido a partir da aplicação de um patamar de corte nos valores dos elementos presentes na matriz \mathbf{H}^+ , gerando assim uma matriz $\hat{\mathbf{H}}^+$ e que foi nomeado de $\varsigma^{(14)}$.

Em segundo lugar, foi feito um estudo comparativo entre todas as variações do método SSF e, também, com o método PCD. As variações do método SSF são:

- a. o próprio método SSF, conforme definido por Elad (2010);
- b. a variação *SSF plus Least Square* (SSF+LS);
- c. a variação *SSF plus LS with a ν constant* (SSF+LS ν);
- d. e, ainda, uma pré-inicialização do vetor \mathbf{y}_0 , conforme definida na Subseção 4.4.2, aplicada em cada uma das variações definidas acima.

¹⁴Conforme Aguiar *et al.* (2019), existe a possibilidade de se considerar apenas alguns dos elementos presentes na matriz \mathbf{H}^+ e, ainda assim, garantir uma convergência satisfatória dos valores de RMSr. A matriz $\hat{\mathbf{H}}^+$ é utilizada na inicialização do vetor \mathbf{y}_0 através da multiplicação da matriz $\hat{\mathbf{H}}^+$ pelo vetor \mathbf{s} , ou seja: $\mathbf{y}_0 = \hat{\mathbf{H}}^+ \mathbf{s}$, onde o vetor \mathbf{s} representa cada uma das janelas do vetor \mathbf{r} .

5.4.1 Análises de parâmetros e configurações

Para um estudo mais aprofundado dos métodos *Shrinkage* e das variações do método SSF, foi necessário iniciar pela análise detalhada dos parâmetros λ e μ , presentes no método SSF original e em todas as suas variações, além dos parâmetros ν e ω presentes em algumas das variações do método SSF. Esse procedimento difere dos demais que vêm sendo desenvolvidos e que foram iniciados com o estudo das curvas ROC. As conclusões dessas pesquisas detalhadas serão propagadas para as variações aqui consideradas, visando encontrar valores ótimos que possibilitem uma implementação futura de alguma de suas das variações em *hardware*.

Além da busca de um valor ótimo para cada um dos parâmetros considerados, buscou-se a possibilidade de um valor aproximado de cada um dos parâmetros, de forma a possibilitar a construção de um *hardware* menos complexo. Cada variação do método SSF, além do método PCD, foi aplicada entre 1 e 330 vezes (ou seja, entre 1 e 330 iterações) no mesmo sinal (vetor r) utilizado nas seções anteriores e cujas características foram definidas na Subseção 5.1.3. O resultado exibido nas tabelas é o resultado obtido para 330 iterações do método *Shrinkage* analisado, quando não for especificado o contrário.

5.4.1.1 Análise do parâmetro λ

O parâmetro λ é o multiplicador de Lagrange responsável pelo equilíbrio das normas ℓ_2 e ℓ_1 , conforme mencionado em Elad (2010).

As variações realizadas no parâmetro λ apresentadas no Apêndice D na Subseção D.3.1 indicam que tal parâmetro tende a zero, sem causar maiores danos nos resultados obtidos com o método SSF. Essa tendência favorece a norma ℓ_2 , que lida com a representação dos dados em meio ao ruído e, assim, em um ambiente de maior concentração de *pile-up*.

Também pode-se verificar nos gráficos do Apêndice D na Subseção D.3.1, que as imagens à esquerda representam os valores de RMSr obtidos em cada iteração (aplicação do método SSF variando de 1 até 330) para valores de λ variando de -2,0 até 2,0; com passos de 0,1 em 0,1. Os gráficos à direita representam os valores médios de RMSr para os valores de λ variando de -2,0 até 2,0; com passos de 0,1 em 0,1.

Os valores médios de RMSr foram agrupados no Gráfico 31 (Apêndice D na Subseção D.3.1), que demonstra um valor de RMSr para cada uma das taxas de ocupação utilizadas, facilitando aferir a tendência da convergência para zero do valor de λ , principalmente para médias e altas taxas de ocupação. Nessa figura, um ponto identifica o valor mínimo de RMSr em cada uma das taxas de ocupação.

A Tabela 3 apresenta um comparativo entre os valores de RMSr, de cada um dos valores ótimos de λ para uma dada taxa de ocupação, os valores de RMSr de $\lambda = 0$, também detalhados por taxa de ocupação e, ainda, uma coluna com a diferença percentual entre os valores de RMSr calculada. Os valores de RMSr apresentados foram obtidos em 330 iterações do método SSF e não correspondem aos valores médios presentes nos gráficos constantes nesta subseção. Na

Tabela 3, nota-se que um valor zero para λ só possui um impacto negativo quanto se considera uma taxa de ocupação muito baixa (de 1%), para as 330 iterações do método SSF sobre o sinal aqui trabalhado.

Tabela 3 - Comparaçāo entre os valores de λ

Taxa de ocupação	Melhor valor de λ	RMSr do melhor λ	RMSr para $\lambda = \text{zero}$	Diferença entre valores de λ (%)
1%	0,6	4,30	5,64	31,27
5%	0,4	9,83	10,15	3,34
10%	0,2	13,86	13,95	0,65
20%	0,0	20,79	20,79	0,00
30%	0,0	26,29	26,29	0,00
40%	0,0	31,42	31,42	0,00
50%	0,0	36,16	36,16	0,00
60%	0,0	40,39	40,39	0,00
90%	0,0	51,75	51,75	0,00

Fonte: Elaborada pelo autor (2021).

5.4.1.2 Análise do parâmetro μ

Uma vez fixado em zero o valor do parâmetro λ , pode-se iniciar a análise do parâmetro μ , que representa o tamanho do passo em direção ao mínimo e está presente na Equação 57. No estudo de Elad (2010), a etapa de LS é opcional e existe a possibilidade de escolha de um valor de μ que minimize o valor real da função, mas não há uma definição clara de como se chega a esse valor de μ . Para tal, buscou-se uma forma de encontrar o valor de μ em Duarte (2015).

Com base no trabalho de Duarte (2015), considerou-se a fórmula presente na Equação 27 utilizada pelo método GD e implementada no Algoritmo 2 na linha 13. Lembrando que φ é equivalente a $(\mathbf{r} - \mathbf{H}\hat{\mathbf{x}}_{it})$ (DUARTE, 2015). Porém, tal cálculo tornaria inviável a implementação em *hardware* de qualquer variação do método SSF que pudesse ser considerada. Conforme se verifica a partir dos gráficos presentes no Apêndice D, na Subseção D.3.2, ele tende a convergir para um valor que pode ser fixado.

Ainda em relação aos gráficos do Apêndice D, na Subseção D.3.2, as imagens à esquerda representam os valores individuais de μ , obtidos para cada uma das 1.820 janelas amostrais, em cada uma das 330 iterações (o método SSF foi aplicado de 1 a 330 vezes, em cada uma das janelas). As imagens à direita representam os valores médios de μ obtidos em cada iteração. Com isso, a partir de uma determinada quantidade de iterações o valor de μ se estabiliza.

É possível ver um resumo dos valores médios de μ , obtidos para todas as taxas de ocupação, aqui considerada, no Gráfico 37 (Apêndice D na Subseção D.3.2). Verifica-se que, apesar de serem valores diferentes, todos tendem a convergir para uma determinada estabilidade, independente da taxa de ocupação analisada.

Os gráficos presentes no Apêndice D na Subseção D.3.3, revelam que o valor 0,25 converge para um valor mínimo de RMSr e facilita sua implementação em *hardware*. Esse valor é equivalente a um deslocamento de dois *bits* à direita, ou $>> 2$. Além disso, os quatro valores avaliados para μ foram: um dinâmico (de acordo com a Equação 27) e três fixos, a fim de se evitar o cálculo da Equação 27 em hardware. Os três fixos foram escolhidos dentro do intervalo dos valores obtidos através da abordagem dinâmica, de modo que não se exigisse um circuito de multiplicação, em uma representação numérica de pontos fixos, para qualquer destes valores em hardware (apenas operação de mudança de bit) (MILNE, 2016). Como o valor $\mu = 0,5$ era muito alto, o algoritmo acabou por se revelar instável. No outro extremo, para $\mu = 0,125$, a convergência foi muito lenta. A melhor solução encontrada foi com $\mu = 0,25$, que produziu uma avaliação de desempenho semelhante ao μ dinâmico.

A convergência para o valor de $\mu = 0,25$ foi validada pela Tabela 4, onde fez-se um comparativo percentual entre o valor de RMSr obtido a partir da aplicação de um μ gerado dinamicamente – conforme a Equação 27 – e o valor de RMSr obtido a partir da aplicação de $\mu = 0,25$. Nota-se que a diferença percentual entre ambos é mínima, sobretudo se comparado ao ganho obtido na diminuição da complexidade do circuito que será gerado a partir da fixação do valor de μ .

Tabela 4 - Comparaçāo entre os valores de μ

Taxa de ocupação	RMSr para μ dinâmico	RMSr para $\mu = 0,25$	Diferença entre valores de μ (%)
1%	5,64	5,64	0,00
5%	10,15	10,16	0,03
10%	13,95	13,96	0,07
20%	20,79	20,85	0,27
30%	26,29	26,43	0,55
40%	31,42	31,68	0,81
50%	36,16	36,57	1,14
60%	40,39	41,01	1,54
90%	51,75	52,63	1,70

Fonte: Elaborada pelo autor (2021).

5.4.1.3 Análise dos parâmetros ω e ς

A análise de ω , para a formação da matriz \hat{H}^+ , utilizada na inicialização prévia do vetor y_0 , em $y_0 = \hat{H}^+ s$ foi possível a partir da fixação dos parâmetros μ e λ em 0,25 e 0,0, respectivamente. Sem essa inicialização prévia, o vetor y_0 é formado pelos elementos pertencentes ao BC onde houve colisões, ou seja, pelos 48 *bunches* de uma janela de $48b7e$, onde os 7 *bunches empty* são ignorados. A análise se apoiou, também, nos resultados publicados por Aguiar *et al.* (2019), que demonstram ser possível uma convergência dos valores de RMSr,

através da aplicação de algum valor de ς ⁽¹⁵⁾, mesmo que apenas alguns valores de ω ⁽¹⁶⁾ sejam mantidos.

A partir de um dado valor de ς , valores inferiores são zerados, o que cria um determinado grau de esparsidade na matriz \mathbf{H}^+ original, e possibilita a construção de circuitos menos complexos e menores, sem aumentos significativos nos valores de RMSr obtidos. Ao se considerar um sinal contendo o padrão $48b7e$, a matriz \mathbf{H}^+ conterá 2.592 valores únicos, tornando inviável a sua implementação em *hardware*. A análise gráfica desse parâmetro pode ser conferida no Apêndice D na Subseção D.3.4, indicando que mesmo possuindo um valor de ω muito pequeno há uma convergência para valores estáveis de RMSr, ainda em poucas iterações, o que torna mais fácil a implementação do mesmo em *hardware*.

Assim como nos demais, pode-se constatar nos gráficos do Apêndice D na Subseção D.3.4, que as imagens à esquerda representam os valores de RMSr, obtidos em cada iteração (variando de 1 até 330 a aplicação do método SSF). Os valores de RMSr são calculados para a variação de valores de ω , onde os mesmos variam de 0 até 2.592, de forma distributiva por amostragem aleatória, com um total de 40 amostras. As imagens à direita representam os valores médios de RMSr, para ω variando de 0 até 2.592, também de forma distributiva por amostragem aleatória, com um total de 40 amostras.

Os valores médios de RMSr de 330 iterações do algoritmo, em relação ao valor de ω , e também em relação a cada uma das taxas de ocupação utilizadas, foram agrupados no Gráfico 48 (Apêndice D na Subseção D.3.4), facilitando, assim, aferir a tendência na obtenção de valores estáveis de RMSr, quando ω possui um valor a partir de 140 (ς abaixo de 0,9).

A Tabela 5 foi criada a partir da tendência aferida até aqui e ao publicado em Aguiar *et al.* (2019). Nela é possível identificar o valor de ω para que o valor de RMSr convirja a um valor mínimo de RMSr, em relação aos outros valores de RMSr, obtidos a partir de vários valores distintos de ω . Na mesma tabela, é possível averiguar a diferença percentual entre os valores de RMSr, obtidos no melhor caso (menor valor de RMSr em relação aos demais) e o valor de RMSr quando considerado $\omega = 127$ ($\varsigma = 1,4$). Percebe-se que, quanto maior a taxa de ocupação, maior é a diferença percentual. Mesmo assim, essa diferença percentual não passa de 1% e, para uma taxa de ocupação de 90%, a utilização da matriz $\hat{\mathbf{H}}^+$ com $\omega = 127$ chega a ser melhor do que a da matriz \mathbf{H}^+ completa, com $\omega = 2.592$.

¹⁵Como dito no início da Seção 5.4, ς representa o patamar de corte, que aplicado aos valores dos elementos da matriz \mathbf{H}^+ geram ω .

¹⁶De igual forma, no início da Seção 5.4, ω representa o número de elementos não nulos da matriz \mathbf{H}^+ .

Tabela 5 - Comparaçao entre quantidades de elementos em $\hat{\mathbf{H}}^+$

Taxa de ocupação	ω	RMSr	$\omega = 127$	Diferença %
1%	180	5,64	5,64	0,00
5%	1128	10,15	10,15	0,02
10%	986	13,94	13,95	0,03
20%	2592	20,73	20,76	0,15
30%	816	26,15	26,22	0,26
40%	986	31,23	31,37	0,43
50%	2592	35,91	36,12	0,58
60%	1148	40,11	40,46	0,88
90%	2592	51,97	51,84	-0,25

Fonte: Elaborada pelo autor (2021).

A ocorrência de uma taxa de ocupação muito alta (acima de 40%) é rara no LHC, apesar de o aumento da diferença percentual ocorrer à medida que ela aumenta. A utilização de $\omega = 127$, para a representação da matriz $\hat{\mathbf{H}}^+$, torna viável a construção de um circuito que o comporte, diferentemente do que ocorreria com a utilização da matriz \mathbf{H}^+ de forma completa com todos os seus 2.592 elementos, pois tornaria inviável a construção de um circuito que comportasse tal matriz.

5.4.1.4 Análise do parâmetro ν

Os parâmetros examinados, até então, foram aplicados de forma cumulativa na versão original do método SSF. Partiu-se da análise do parâmetro λ do método SSF, originalmente definido por Elad (2010), na sua versão com o valor de λ fixado em zero, para se analisar o parâmetro μ . Assim, após aplicar o parâmetro λ fixado em zero e o parâmetro μ fixado em 0,25 no método SSF, iniciou-se a análise de ω , que culminou com a geração de uma variação do método SSF chamada *SSF with vector y_0 pre-initialized* (SSFi). O estudo desse parâmetro embasou a criação de variações com a inicialização prévia do vetor y_0 . Tais variações serão fundamentadas nas outras propostas de variações do método SSF já apresentadas. Nas variações em que houver a inicialização prévia do vetor y_0 , elas serão identificadas com a letra **i**, inserida após a sua respectiva sigla.

Buscou-se, ainda, a definição de um valor de ν , para ser aplicado na variação do método SSF+LS, formando assim uma terceira variação do método SSF: a variação *SSF plus LS with a ν constant* (SSF+LS ν). Para tal foi utilizado o Algoritmo 9, onde foram gerados 6.086 valores de ν entre 0,01 e 100, de forma a se encontrar um valor de ν que pudesse gerar um menor valor de RMSr, em relação à variação SSF+LS do método SSF. Essa busca se deu para cada uma das taxas de ocupação aqui utilizadas.

No Gráfico 13 (Apêndice D) são visualizados os 6.086 valores de RMSr, obtidos para

cada um dos 6.086 valores de ν analisados. Verificam-se, também, pontos em destaque, onde os valores de ν apresentam os menores valores de RMSr. Esses 6.086 pontos foram gerados para cada uma das taxas de ocupação consideradas, ou seja, para cada taxa de ocupação a ser considerada 6.086 valores diferentes de ν foram gerados.

No Gráfico 13 (Apêndice D), também, há um grande adensamento de pontos entre a faixa de zero a quatro, com a predominância dos menores valores de RMSr ocorrendo entre zero e dois, para todas as taxas de ocupação consideradas. Esse adensamento de pontos (entre 0 e 4) pode ser explicado pela estrutura do Algoritmo 9, que tende a gerar valores menores que 10. O fato de os menores valores de RMSr ocorrerem predominantemente entre zero e dois demonstra a fragilidade dessa variação SSF+LS ν , pois alterações mínimas no valor de ν (variações na terceira ou quarta casa decimal) podem causar uma grande variação nos valores de RMSr obtidos.

Os valores de ν , que obtiveram os menores valores de RMSr para cada uma das taxas de ocupação consideradas podem ser verificados na Tabela 6. Demonstra-se, também, a tentativa de utilização de um valor de ν que fosse de fácil implementação em *hardware*, uma vez que pudesse ser multiplicado por uma determinada constante (como 65, por exemplo) e, em seguida, um deslocamento de bits que pudesse reduzir os efeitos dessa multiplicação para algo próximo ao alcançado pelos melhores valores de ν encontrados.

Tabela 6 - Comparaçāo entre valores de ν

Taxa de ocupação	Melhor valor de ν	RMSr do melhor ν	RMSr para $\nu = 1,01562$	Diferença entre valores de ν (%)
1%	1,04918	4,89	5,22	6,37
5%	1,04167	7,84	8,70	9,88
10%	1,03659	11,13	12,03	7,49
20%	1,02041	18,42	18,54	0,65
30%	1,01149	24,78	24,95	0,69
40%	1,01010	30,76	31,95	3,74
50%	1,01010	36,41	38,84	6,24
60%	1,01010	42,13	46,23	8,86
90%	1,01010	58,74	69,18	15,08

Fonte: Elaborada pelo autor (2021).

No esforço para se fixar um valor constante para ν , tentou-se a utilização de um valor constante equivalente ao deslocamento à direita de 6bits do valor 65 ($65 >> 6$), gerando, assim, um valor constante de 1,015625. Tal valor é muito próximo aos melhores valores, dentre os 6.086 testados, encontrados para ν e é de fácil implementação em um *hardware* de ponto fixo, com precisão de, pelo menos, 6bits para a parte decimal.

A Tabela 6 apresenta, ainda, uma coluna com os valores de RMSr obtidos com a aplicação da constante 1,015625 e uma outra coluna com a diferença percentual entre os valores de RMSr, obtidos entre os menores de RMSr, e os valores de RMSr obtidos com essa constante proposta.

Essa última coluna demonstra que, mesmo para taxas de ocupação onde os valores de ν se repetem, há uma grande diferença percentual em relação a utilização da constante proposta.

Outra questão a ser notada é a de que, para valores de taxa de ocupação entre 20% e 40%, o uso da constante proposta aparenta ser viável, pois essa variação nas taxas de ocupação é muito próxima da realidade de operação do L1 do *trigger* do TileCal.

5.4.1.5 Configurações recomendadas para implementação em hardware

Uma síntese das configurações recomendadas para utilização em futuras implementações em *hardware* pode ser visualizada no Quadro 4, já que os quatro parâmetros utilizados em todas as variações do método SSF, aqui referenciadas, foram investigados.

Quadro 4 - Configuração recomendada para cada uma das variações do método SSF

Constante	Valor	Variações do método SSF					
		SSF¹⁷	SSFI_s	SSFI_{sc}	SSFI_{sci}	SSFI_{si}	SSFI
λ	0,0	✓	✓	✓	✓	✓	✓
μ	0,25	✓	✓	✓	✓	✓	✓
$\omega (\varsigma)$	127 (1,4)				✓	✓	✓
ν	1,01562			✓	✓		

Fonte: Elaborado pelo autor (2021).

As configurações registradas no Quadro 4 foram usadas parcialmente na geração dos resultados obtidos pelas análises de curva ROC, apresentadas na próxima subseção (Subseção 5.4.2). Apenas as configurações definidas para os parâmetros λ e μ foram seguidas na integralidade, para a obtenção dos resultados gerados por todas as variações do método SSF. Em relação ao parâmetro ν considerou-se o resultado que obteve o menor valor de RMSr para cada taxa de ocupação considerada. Quanto a ω , a matriz foi utilizada de forma integral na geração dos resultados das variações onde houver a inicialização prévia do vetor y_0 , com todos os seus 2.592 elementos. Ainda em relação à ω , um estudo mais aprofundado consta na Subseção 5.5.1.

A fórmula originalmente escrita na Equação 59 pode ser reescrita como a Equação 78, a partir da configuração proposta no Quadro 4. As letras representativas das matrizes e vetores envolvidos foram substituídas pelas mesmas letras presentes nos algoritmos implementados, aqui,

¹⁷Esse é o método SSF original disponível em Elad (2010).

em pseudo-código e no código em Python (MCKINNEY, 2017), nessa nova equação reescrita.

$$\mathbf{y}_{i+1} = S_0^+ \left(\mathbf{y}_i + 0.25 \begin{bmatrix} \underbrace{\mathbf{H}^T \mathbf{s}}_{\text{constante}} & - & \underbrace{\mathbf{A} \mathbf{y}_i}_{\substack{\text{varia em} \\ \text{cada iteração}}} \end{bmatrix} \right) \quad (78)$$

onde: \mathbf{y}_0 é formado pela multiplicação entre $\hat{\mathbf{H}}^+$, que representa a matriz \mathbf{H}^+ após a aplicação de algum valor de ζ , para a redução de ω dessa matriz; e o vetor \mathbf{s} , que representa cada uma das janelas do vetor \mathbf{r} , contendo todos os elementos da janela amostral e não somente a parte da janela amostral onde ocorreram de fato as colisões.

Resumindo as principais adaptações realizadas na Equação 59 tem-se:

- a. a fixação de um valor igual a zero para a constante λ , determinada pelas análises realizadas na Subseção 5.4.1.1. Essa escolha elimina a necessidade de construção de circuitos adicionais de comparação;
- b. a fixação de um valor igual a 0,25 para a constante μ . Essa fixação se deu a partir das análises realizadas na Subseção 5.4.1.2, além do fato de que ela evita a criação de um circuito adicional de multiplicação (apenas deslocamento de bits são necessários);
- c. a fixação de uma matriz \mathbf{A} resultante da multiplicação da matriz \mathbf{H} por sua transposta ($\mathbf{A} = \mathbf{H}^T \mathbf{H}$), eliminando a necessidade desse cálculo em hardware;
- d. o janelamento dos vetores \mathbf{r} e \mathbf{x} , criando sub-vetores \mathbf{s} e \mathbf{y} , respectivamente, possibilitado pelas características do próprio LHC em gerar trem de colisões em blocos de 48 colisões consecutivas, seguido de 7 colisões "vazias" (período sem colisões), conforme definido por Kostoglou *et al.* (2020). Esse janelamento permitiu a aplicação do método em partes menores, exigindo uma menor capacidade computacional na resolução do problema;
- e. a inicialização do vetor \mathbf{y}_0 através da multiplicação do vetor \mathbf{s} pela pseudoinversa da matriz \mathbf{H} , chamada de matriz \mathbf{H}^+ – análise realizada na Subseção 5.4.1.3. Essa inicialização diminui consideravelmente o número de iterações subsequentes necessárias;
- f. modificação na Função de Shrinkage (denominada S_λ^+) para gerar somente valores não-negativos.

As modificações realizadas refletiram, de modo consequente, no Diagrama 10 (Apêndice C), gerando o Diagrama 12 (Apêndice D).

O Algoritmo 7, também sofre as devidas alterações tornando-se o Algoritmo 13. Para se chegar às variações SSF+LS e SSF+LS ν basta substituir as linhas de 12 até 14 do Algoritmo 13 pelas linhas de 12 até 15 do Algoritmo 8 obtendo-se o SSF+LS, ou pelas linhas de 12 até 15 do Algoritmo 10 obtendo-se o SSF+LS ν .

Algoritmo 13: IMPLEMENTAÇÃO DO SSF SEGUNDO CONFIGURAÇÕES DA TABELA 4

Entrada: Tamanho da amostra, tamanho da janela, número de iterações, matriz H e sinal r

Saída: sinal \hat{x}

```

1 início
2    $A \leftarrow H^T H$ 
3    $H^+ \leftarrow H^T * A^{-1}$ 
4    $\hat{H}^+ \leftarrow H^+(abs(valorElementos) > \varsigma)$ 
5    $\hat{x} \leftarrow inicializa$ 
6   para  $it \leftarrow (it + 1)$  até  $quantidadeDeAmostras$  faz
7      $passo \leftarrow ((it * janela) - (janela - 1))$ 
8      $s \leftarrow r(passo : passo + (janela - 1))$ 
9      $H^T s \leftarrow H^T * s$ 
10     $y_0 \leftarrow \hat{H}^+ * s$ 
11    para  $i \leftarrow (i + 1)$  até  $númeroDeIterações$  faz
12       $B \leftarrow H^T s - (Ay_i)$ 
13       $y_i \leftarrow (y_i + 0.25B)$ 
14       $y_i(y_i < 0) \leftarrow 0$ 
15    fim
16     $\hat{x}(passo : passo + (janela - 1)) \leftarrow y$ 
17  fim
18 fim
19 retorna  $\hat{x}$ 

```

5.4.2 Análises de curvas ROC

Uma vez que todos os parâmetros e variações a serem aplicados no método SSF foram fixados, desviou-se o foco da investigação para a análise das curvas ROC, assim como suas variações, dos métodos *Shrinkage*.

No Apêndice D na Subseção D.3.5, se encontram as curvas ROC tanto para as variações do método SSF, quanto do método PCD, ambos sem a inicialização prévia do vetor y_0 . Nessas curvas pode-se observar que a variação SSF+LS ν apresenta uma leve vantagem em relação às outras variações do método SSF, fracassando para o método SSF original apenas quando a taxa de ocupação fica em 90%. Nota-se, ainda, que todas as outras variações do método SSF apresentam resultados muito próximos entre si e em relação ao método PCD.

O mesmo comportamento pode ser observado no Apêndice D na Subseção D.3.6, para todos os métodos de *Shrinkage* aqui considerados. Onde agora está a inicialização prévia do vetor y_0 , mantendo o comportamento anterior, a variação *SSFi+LS with a ν constant and vector y_0 pre-initialized* (SSFi+LS ν) apresenta uma leve vantagem em relação às outras variações

do método SSF e do método PCD, perdendo para os mesmos apenas quando a taxa de ocupação fica em 90%.

Para os métodos *Shrinkage*, incluindo as variações do método SSF, o valor de patamar, representado por ϵ_0 , é a energia do resíduo permitido na reconstrução final. Como tal resíduo possui dimensão de $(MeV)^2$, foi considerado, aqui, o valor de patamar já calculado como $\sqrt{(\epsilon_0)}$, tornando os valores representados compatíveis com os valores calculados para os demais métodos já estudados.

Nas imagens presentes nos Gráficos 54 e 60 (Apêndice D na Subseção D.3.5 e D.3.6), há um valor de $\sqrt{(\epsilon_0)}$ em cada uma das taxas de ocupação, que corresponde ao menor valor de RMSr para aquela taxa de ocupação. Isso ocorre tanto para o método PCD, quanto para as variações do método SSF aqui consideradas. O mesmo comportamento é observado com ou sem a inicialização prévia do vetor y_0 .

Ao analisar os dados da Tabela 7, constata-se que, para taxas de ocupação de 1% até 40%, a variação SSF+LS ν obteve os menores valores de RMSr, em relação às outras variações do método SSF e ao método PCD. Para taxas de ocupação acima de 40%, o método PCD apresentou os menores valores de RMSr. A Tabela 7 se refere aos métodos onde não foi aplicada a inicialização prévia do vetor y_0 .

Tabela 7 - Comparação entre os métodos *Shrinkage*,
sem inicialização prévia do vetor y_0

Taxa de ocupação	SSF	SSF+LS	SSF+LS com ν	PCD
1%	5,64	5,64	4,89	5,64
5%	10,16	10,19	7,84	10,16
10%	13,96	14,06	11,13	13,95
20%	20,85	21,22	18,42	20,80
30%	26,43	27,29	24,78	26,30
40%	31,68	32,99	30,76	31,44
50%	36,57	38,51	36,41	36,18
60%	41,01	43,72	42,13	40,41
90%	52,63	56,94	58,74	51,75

Nota(s): valores em MeV.

Fonte: Elaborada pelo autor (2021).

Em relação aos métodos que contaram com a inicialização prévia do vetor y_0 , foram encontrados os dados incluídos na Tabela 8. Analisando-os, verificou-se comportamento semelhante ao apresentado em relação aos métodos que não contaram com a inicialização prévia do vetor y_0 . Para taxas de ocupação variando de 1% até 40% a variação SSFi+LS ν obteve os menores valores de RMSr, em relação as outras variações do método SSF e ao método PCD. Para taxas de ocupação acima de 40% a variação *SSFi+LS with vector y_0 pre-initialized* (SSFi+LS)

apresentou os menores valores de RMSr.

Tabela 8 - Comparação entre os métodos *Shrinkage*, com inicialização prévia do vetor y_0

Taxa de ocupação	SSFi	SSFi+LS	SSFi+LS com ν	PCDi
1%	5,64	5,64	4,89	5,64
5%	10,15	10,15	7,85	10,15
10%	13,94	13,94	11,13	13,94
20%	20,73	20,71	18,44	20,73
30%	26,15	26,10	24,85	26,17
40%	31,23	31,14	31,00	31,27
50%	35,91	35,77	37,11	35,97
60%	40,11	39,94	43,70	40,19
90%	51,97	51,87	67,36	52,05

Nota(s): valores em MeV.

Fonte: Elaborada pelo autor (2021).

Na aplicação dos métodos *Shrinkage* não há definição de um patamar de corte que deva ser utilizado, conceito que está definido apenas para os métodos *Greedy* e foi utilizado, aqui, apenas na montagem das curvas ROC. Portanto, para a construção das Tabelas 7 e 8 foi considerado o patamar zero.

5.5 COMPARATIVO ENTRE OS MÉTODOS

O objetivo principal dessa seção é fazer uma análise comparativa entre os métodos que obtiveram os melhores resultados, dentre os de suas respectivas categorias, conforme detalhado a seguir. Considera-se, também, o grau de complexidade da implementação dos métodos que obtiverem os menores valores de RMSr em *hardware*, possibilitando, assim, a escolha de algum método que possa alcançar, dentre os métodos em avaliação, um equilíbrio entre o menor valor de RMSr e uma menor complexidade de implementação em *hardware*.

Os critérios utilizados para a escolha de cada um dos métodos, de forma a possibilitar uma análise comparativa entre eles foram:

- Métodos já propostos – método DS, por este apresentar os menores valores de RMSr, em relação aos métodos da mesma categoria; filtro FDIP, por ser um método de referência entre os métodos já propostos;
- Métodos *Greedy* – o método LS-OMP, por apresentar os menores valores de RMSr, em relação aos métodos *Greedy*; e o método OMP, por apresentar resultados muito próximos aos alcançados com o LS-OMP;
- Métodos *Shrinkage* – entre os métodos e variações de métodos *Shrinkage*, que não possuem a aplicação da inicialização prévia do vetor y_0 , a variação SSF+LS ν foi a selecionada.

Além dessa variação, mas dentro dessa categoria, o método PCD original (conforme descrito em Elad (2010)) também foi selecionado. Esses dois métodos foram selecionados por apresentarem valores muito próximos de RMSr entre si, e também, por serem os menores valores de RMSr, entre os métodos da mesma categoria. Já entre os métodos e variações de métodos *Shrinkage*, que possuem a aplicação da inicialização prévia do vetor y_0 , a variação SSFi+LS ν foi selecionada em conjunto com a variação SSFi+LS, pois ambas possuem os menores valores de RMSr, dentre os de mesma categoria. Ainda dentro dessa categoria, será analisado na Subseção 5.5.1 o método SSFi, pois o mesmo apresenta valores de RMSr muito próximos ao SSFi+LS.

Apenas os métodos que obtiveram os melhores resultados de estimação em cada uma das categorias analisadas, foram selecionados para terem suas curvas ROC comparadas entre si. As curvas ROC estão presentes no Apêndice D na Subseção D.4.1. Pode ser observado que as variações SSFi+LS ν e SSFi+LS do método SSF obtiveram os melhores resultados de detecção, quando considerados os demais métodos.

Analizando tais gráficos, pode-se notar que os métodos DS e LS-OMP se alternam com o método LS-OMP, apresentando os melhores resultados de detecção, quando aplicado nas baixas taxas de ocupação e o método DS apresenta os melhores resultados de detecção quando aplicado nas médias e altas taxas de ocupação. Mesmo assim, ambos perdem em detecção para as variações do método SSF, sendo a variação SSF+LS ν a que obteve os melhores resultados entre os demais métodos, fracassando apenas em relação à variação SSFi+LS, quando a taxa de ocupação é de 90%.

No Apêndice D os gráficos FDIP, DS (Gráfico 19) OMP, LS-OMP (Gráfico 25), SSF+LS, PCD, (Gráfico 54), SSFi+LS, SSFi+LS ν (Gráfico 60) mostram as imagens de RMSr para cada um dos métodos, aqui considerados, na ordem definida no início dessa seção. Na Tabela 9, os valores numéricos presentes nesses gráficos de RMSr podem ser visualizados.

Tabela 9 - Valores de RMSr dos dois melhores métodos de cada uma das categorias estudadas

Taxa de ocupação	FDIP	DS	OMP	LS-OMP	SSF+LS com ν	PCD	SSFi+LS	SSFi+LS com ν
1%	19,32	9,75	3,26	2,85	4,89	5,64	5,64	4,89
5%	26,87	13,21	8,77	7,28	7,84	10,16	10,15	7,85
10%	30,94	16,63	13,95	11,80	11,13	13,95	13,94	11,13
20%	36,64	22,92	23,27	19,36	18,42	20,80	20,71	18,44
30%	40,44	28,32	30,79	27,12	24,78	26,30	26,10	24,85
40%	43,65	33,86	37,26	33,44	30,76	31,44	31,14	31,00
50%	46,13	39,08	43,24	39,91	36,41	36,18	35,77	37,11
60%	48,49	44,89	48,17	45,66	42,13	40,41	39,94	43,70
90%	54,15	59,26	57,67	57,67	58,74	51,75	51,87	67,36

Nota(s): valores em MeV.

Fonte: Elaborada pelo autor (2021).

Uma análise mais atenta da Tabela 9, considerando apenas os menores valores de RMSr, sugere:

- a. que para um ambiente extremamente esparsos, onde a taxa de ocupação fosse de 1%, o método LS-OMP seria a escolha ideal;
- b. que no outro extremo, onde a taxa de ocupação fosse de 90%, ou seja, um ambiente com baixíssima esparsidade, o método PCD seria a opção ideal;
- c. que em um ambiente mais próximo da realidade do L1 do *trigger* do TileCal, onde a taxa de ocupação varia de 20% a 40% a variação SSF+LS ν , seria a alternativa ideal.

Porém, como não é apenas o menor valor de RMSr a única métrica que está sendo considerada, uma análise comparativa mais detalhada será apresentada mais à frente. Na Subseção 5.5.1, serão consideradas todas às métricas utilizadas, aqui, de forma a possibilitar uma escolha mais assertiva do método que deverá ser implementado em *hardware*, levando em consideração.

5.5.1 Comparativo em relação à complexidade de implementação em hardware

Uma vez encontrada a variação SSF+LS ν do método SSF e sendo a variação que apresentou os menores valores de RMSr, para a maioria das taxas de ocupação estudadas, efetuou-se uma análise comparativa entre ela e a variação SSFi, com duas diferentes opções de quantidades de ω , presentes na pseudo-inversa da matriz \mathbf{H} , a matriz \mathbf{H}^+ . Em uma variação, a matriz \mathbf{H}^+ possui todos os seus 2.592 elementos e, em outra, foi aplicado um valor de ς , considerando apenas elementos cujos valores estavam em uma faixa $\varsigma \pm 1,4$. Com isso, a matriz \mathbf{H}^+ passou a possuir $\omega = 127$.

A busca por uma alternativa à variação que obteve o menor valor de RMSr, em relação a todos os métodos e variações, foi motivada com base na dificuldade de implementação em *hardware* da variação SSF+LS ν . A versão que obteve os menores valores de RMSr contou com um valor de ν ótimo, para cada uma das taxas de ocupação analisadas, conforme se verificou na Tabela 6.

As imagens presentes no Apêndice D na Subseção D.4.2 apresentam o comparativo entre as variações aqui consideradas, sendo que nesses gráficos foram colocados os valores obtidos com o método Filtro FDIP, de forma a servir como referência para uma avaliação mais precisa dessas variações do método SSF. Nota-se a ocorrência de uma marca a cada 55 iterações, cujo objetivo duplo é:

- a. facilitar a visualização do número de iterações necessários para cada variação, até a estabilização do valor de RMSr;
- b. conhecer o número de iterações necessários à obtenção de um valor de RMSr menor que o obtido pelo método FDIP de referência.

Observando os referidos gráficos, têm-se que, quando aplicado um processamento prévio do vetor y_0 , o método SSF original (conforme definido em Elad (2010)) tende a necessitar de um número menor de iterações para atingir um ponto de estabilidade de valores mínimos de RMSr.

Esses valores estáveis são menores que os apresentados pelo método FDIP e ocorrem antes de se atingir a estabilidade dos valores gerados pela variação SSF+LS ν . Essa última variação foi a que obteve os melhores resultados entre todos os métodos e variações testadas. Os valores alcançados são menores que os obtidos pelo método FDIP, mesmo que a estabilidade não garanta valores de RMSr menores que os obtidos pela variação SSF+LS ν . Assim, esse resultado exige menos iterações para ser atingido e acaba por garantir um *hardware* que exija menos recursos em sua fabricação.

Dando continuidade à análise dos gráficos, a variação do SSF+LS ν necessita de um número maior de iterações para atingir valores mínimos de RMSr, que sejam menores que os valores apresentados pelo método FDIP. À medida que a taxa de ocupação aumenta, ocorre uma instabilidade nos valores mínimos de RMSr, chegando a elevar o valor do RMSr conforme o aumenta o número de iterações. Essa instabilidade é notada para as taxas de ocupação acima de 50%.

Diante da análise das variações do método SSF, conclui-se que a variação SSFi, utilizando uma matriz $\hat{\mathbf{H}}^+$ com $\omega = 127$, será o método a ser implementado em *hardware*, por apresentar uma convergência mais rápida de valores mínimos de RMSr, que sejam menores que os valores de RMSr apresentados pelo método FDIP. Isso se deve ao mesmo atingir um equilíbrio melhor na relação valor mínimo de RMSr e menor complexidade de implementação em *hardware*. A implementação da variação SSFi, utilizando uma matriz $\hat{\mathbf{H}}^+$ com $\omega = 127$, será o assunto tratado na próxima seção.

Cabe ressaltar que, a variação SSFi será implementada porque é menos complexa que a variação SSF+LS ν e por necessitar de um número menor de iterações para convergir, para um valor mínimo de RMSr. Assim, pode possibilitar a construção de um *hardware* menor e menos complexo e, ainda, que funcione em uma frequência maior, mesmo que para isso seja necessário descartar o método que apresente o menor valor de RMSr. Esse método considerado apresentaria um valor de RMSr bem menor que o valor obtido pelo Filtro FDIP.

Mesmo com o processamento prévio do vetor y_0 essa variação SSFi é bem menos complexa de ser implementada em *hardware* que a variação que obteve os menores valores de RMSr (SSF+LS ν), para a maioria das taxas de ocupação consideradas.

Como parâmetro de avaliação do *hardware* a ser implementado, ele deve apresentar valores de RMSr próximos aos alcançados pela variação SSFi e a matriz $\hat{\mathbf{H}}^+$ com $\omega = 127$ será executada em ponto flutuante. Os valores de RMSr obtidos por essa variação estão detalhados na Tabela 10 e a configuração utilizada nessa geração estão definidas na Tabela 4.

Tabela 10 - Referência para implementação em
hardware

SSFi com matriz \hat{H}^+ com $\omega = 127$			
Iterações	55	110	165
RMSr (ponto flutuante)	29.41	27.09	26.51

Fonte: Elaborada pelo autor (2021).

5.6 CONCLUSÃO DA SEÇÃO 5

Em consonância com o que foi analisado nas seções anteriores, optou-se por implementar a variação SSFi do método SSF original (de acordo com o apresentado por Elad (2010)) em *hardware*. Essa implementação se dará seguindo as especificações do ambiente onde esses métodos serão utilizados e a facilidade de implementação fornecida pelo método SSF. Portanto, sua implementação em *hardware* será estudada na próxima seção.

6 IMPLEMENTAÇÃO DA VARIAÇÃO SSFi DO MÉTODO SSF ORIGINAL

A implementação apresentada nesta seção é a extensão da implementação originalmente realizada e publicada no Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT) 2018 (TEIXEIRA *et al.*, 2018), acrescida da inicialização do vetor y_0 , inicialmente apresentada por Aguiar *et al.* (2019) e posteriormente estendida por Teixeira *et al.* (2020) e (TEIXEIRA; ANDRADE; SEIXAS, 2021).

Nas Seções 3 e 4, consta uma revisão teórica/matemática dos métodos estudados em trabalhos anteriores a este, além de terem sido exploradas duas famílias de métodos baseadas em representação esparsa de dados: *Greedy* e *Shrinkage*. A variação SSF+LS ν foi a que alcançou os menores valores de RMSr, em relação aos demais métodos e variações. Optou-se, portanto, por implementar em *hardware* a variação SSFi, conforme detalhado na Subseção 5.5.1. A escolha se baseou na maior aceitação dos métodos estudados com a implementação em FPGA (PARAB; GAD; NAIK, 2018), propósito deste trabalho, de forma a se construir um circuito que possibilite sua execução de forma ininterrupta (*free-running*), a uma taxa de 40MHz, considerando que este *hardware* contará apenas com operações entre números inteiros, de ponto fixo.

Dentro desse cenário, a variação SSFi se mostrou a mais aderente, mesmo que os melhores resultados obtidos fossem alcançados somente após a adição da inicialização prévia do vetor y_0 no circuito original do método SSF, ressaltando-se que o *hardware* final obtido não sofreu nenhum impacto significativo, devido as otimizações nele realizadas. Para tal, foi necessário definir diferentes quantizações, para diferentes partes do circuito.

6.1 O PROCESSO DE QUANTIZAÇÃO

A necessidade de se definir quantos *bits* deveriam ser deslocados à esquerda ou à direita, se deveu para possibilitar a construção de um circuito em ponto fixo, que mantivesse um elevado grau de precisão, em relação aos resultados obtidos nas simulações realizadas em ponto flutuante. Daí a importância de se focar no desenvolvimento de um circuito que fosse capaz de ser executado na mesma velocidade de *clock* do LHC, ou seja, 40MHz e que ao mesmo tempo ocupasse o mínimo possível de elementos lógicos, permitindo a construção de múltiplas instâncias em uma única placa de FPGA, de forma a viabilizar a execução em paralelo de tais instâncias. Ao processo de definição do quantitativo de *bits* de deslocamento deu-se o nome de *quantização* e, durante o desenvolvimento do circuito, observou-se ser imprescindível a utilização de valores de quantização específicos para cada parte do circuito. Destacam-se:

- a. uma quantidade de *bits* de deslocamento à esquerda na entrada do circuito, para a conversão dos valores fracionários em valores inteiros. A escolha dessa quantidade deveria considerar o menor valor possível, de forma a garantir a precisão dos resultados, sem impactar o tamanho final do circuito;
- b. uma segunda quantidade de *bits* de deslocamento, que servissem para os deslocamentos à esquerda e à direita, de forma a garantir o alinhamento dos valores obtidos com as somas e

- multiplicações necessárias à implementação do circuito como um todo;
- quantidades específicas de *bits* de deslocamento a serem utilizadas para cada uma das constantes representativas das matrizes, de forma a possibilitar que a realização dos cálculos matriciais fossem realizados de forma linear.

Todos os deslocamentos de bits à esquerda serviram para aumentar a precisão dos cálculos realizados com números decimais, ou seja, em um sistema $Qm.n$. Os valores aqui representados como deslocamentos à esquerda servem à parte decimal representada pela letra n , em um sistema $Qm.n$. Os deslocamentos à direita servem para realinhamentos dos bits da parte inteira, necessários para os cálculos realizados em várias partes dos circuitos, no sistema $Qm.n$. Os deslocamentos à direita dizem respeito ao m .

Com as variações dos valores das quantizações, foram obtidos resultados com precisões diferenciadas e com tamanhos de circuitos gerados igualmente variados, fornecendo uma gama de possibilidades de circuitos, a serem utilizados de acordo com os critérios de eficiência de resultados *versus* tamanhos e velocidades de operação. Foram construídas implementações em Python (MCKINNEY, 2017) da variação SSFi, selecionada na Seção 5, para auxiliar na obtenção das quantizações ideais a cada parte do circuito. Tais implementações foram realizadas simulando a construção em ponto fixo do equivalente em Verilog, postergando a implementação em Verilog até serem encontrados os valores das quantizações que, durante as simulações, obtivessem os melhores resultados. Dois critérios principais foram adotados:

- obter a diferença do valor de RMSr entre o sinal alvo (vetor \mathbf{x}) e o sinal obtido na saída do circuito (vetor $\hat{\mathbf{x}}$), devendo tal diferença ser a menor possível, em relação a diferença do valor de RMSr do sinal alvo e do sinal obtido, quando considerada a versão em ponto flutuante do método em análise;
- obter a diferença do valor de RMSr entre o sinal alvo e o sinal obtido na saída do circuito, devendo tal diferença ser a menor possível, em relação a diferença do valor de RMSr do sinal alvo e do sinal obtido, quando utilizada a versão em ponto flutuante de um Filtro FDIP de ordem 26.

Para encontrar os valores de quantização que satisfizessem esses dois critérios, a versão em ponto fixo da variação SSFi foi exposta a um total de 8.448 variações, em quatro métricas avaliadas:

- Número de blocos*: é o número de blocos do método em análise em que cada um varia de tamanho de acordo com o tamanho da janela amostral, podendo ser alcançada, assim, a quantidade de iterações equivalentes, através da multiplicação do número de blocos pelo tamanho da janela amostral. Os detalhes do processo de conversão do tamanho da janela amostral em iterações equivalentes serão elucidados adiante (Seção 6.2);
- Precisão das constantes da matriz H* : essa métrica representa a quantidade de *bits* deslocados à esquerda, utilizados para a conversão em números inteiros dos valores em ponto flutuante das constantes da matriz H . Para se alcançar os mesmos efeitos

nas constantes da matriz de autocorrelação A ¹⁸, basta acrescentar 5 bits nos valores de deslocamento à esquerda da matriz H ;

- c. *Precisão das constantes da matriz \hat{H}^+* : essa métrica corresponde à quantidade de bits deslocados à esquerda e utilizados para a conversão em números inteiros dos valores em ponto flutuante das constantes da pseudo-inversa da matriz H . O fator ς já foi aplicado, conforme descrito na Subseção 5.5.1. É representada por \hat{H}^+ ;
- d. *Ganho*: é o número de bits de deslocamento à esquerda. Tal ganho é aplicado nos valores de entrada vindos de conversores ADC, instalados em cada um dos canais de leitura do TileCal, fazendo com que os valores originalmente de 10 bits (HARKUSHA; COLLABORATION, 2016) do sinal passassem a ser de 10 bits + ganho. Esse valor de ganho é usado, também, para o alinhamento de bits em outras partes do circuito.

A Tabela 11 apresenta um resumo dessas métricas:

Tabela 11 - Conjunto das possibilidades de configurações de construção de circuitos

Métrica	Início	Fim	Passo	Variações
Número de blocos	1	6	tamanho da janela	6
Precisão da matriz H	5	20	1	16
Precisão da matriz H^+	8	64	8	8
Ganho (bits)	5	15	1	11
			Total	8.448

Nota(s): Os valores das constantes das matrizes H e H^+ estão em bits.

Fonte: Elaborada pelo autor (2021).

Para se obter uma configuração que pudesse apresentar uma menor necessidade de elementos lógicos em sua implementação, uma segunda possibilidade de implementação mais eficiente em relação a um Filtro FDIP de ordem 26 e, ainda, com a menor quantidade de elementos lógicos possíveis em sua implementação foi necessário:

- a. fixar os valores de $\lambda = 0,0$ e $\mu = 0,25$, conforme estudado na Seção 5;
- b. calcular a diferença de RMSr entre o sinal alvo (vetor x) e o sinal recuperado (vetor \hat{x}), a partir do sinal medido (vetor r). A recuperação do vetor \hat{x} se deu utilizando a versão em ponto flutuante da variação SSFi;
- c. calcular, de igual forma, a diferença de RMSr entre o sinal alvo e o sinal obtido da aplicação da versão em ponto flutuante de um Filtro FDIP, de ordem 26, no sinal medido;
- d. calcular, ainda, a diferença de RMSr entre o sinal alvo e o sinal recuperado, agora pela versão em ponto fixo da variação SSFi, a partir do sinal medido, para cada variação de quantização realizada;

¹⁸Conforme definido para a Equação 59, a matriz de autocorrelação A é formada pela multiplicação da matriz H por sua transposta, sendo $A = H * H^T$.

- e. aferir a corretude¹⁹ do circuito implementado, comparando-se os resultados obtidos no passo ii e iv e, também, as diferenças de eficiência de cada variação de quantização, em relação a versão em ponto flutuante da variação SSFi;
- f. construir opções de quantizações aderentes aos diferentes cenários de eficiência × recursos de *hardware* disponíveis com os resultados obtidos nos passos iii e iv.

As características dos sinais utilizados nessa implementação foram as mesmas definidas na Seção 5.1, ou seja, 1.820 janelas de $48b7e$, totalizando $\Upsilon = 100.100$ pulsos amostrais. Foi fixado o percentual de 30% de taxa de ocupação para o sinal alvo, pois esse é um valor de taxa de ocupação importante a ser analisado para a Fase II de atualizações do TileCal. O sinal medido foi obtido seguindo o descrito na Seção 5.1. Nesse sinal com ruído foi aplicado um filtro FDIP de ordem 26, obtendo 40,44 MeV de RMSr²⁰, sendo esse valor utilizado para a avaliação dos outros métodos.

Na Tabela 12, encontram-se as configurações que levaram à obtenção dos menores resultados de RMSr, para cada uma das variações testadas na variação SSFi. Também pode-se conferir as configurações mínimas necessárias, para se obter um valor de RMSr mais próximo ao obtido por um Filtro FDIP de ordem 26.

Tabela 12 - Configurações obtidas para o circuito do método SSF

Métrica	Menor RMSr	Menor RMSr que FDIP ²¹
RMS (MeV)	26,95	33,28
Interações	165	55
Precisão da matriz H	7	5
Precisão da matriz H^+	16	8
Ganho na entrada (bits)	10	5

Nota(s): Os valores das constantes das matrizes H e H^+ estão em *bits*.

Fonte: Elaborada pelo autor (2021).

Os circuitos simulados em Python, foram sintetizados e implementados utilizando-se tanto o Intel Quartus Prime Lite Edition versão 19.1 (PARAB; GAD; NAIK, 2018), quanto o Xilinx Vivado Webpack 2019.1 (CHURIWALA, 2016), garantindo a compatibilidade com as principais placas FPGAs presentes no mercado. A latência do circuito gerado foi aferida utilizando o Xilinx Vivado Webpack 2019.1 e a simulação do circuito sintetizado e implementado nas duas plataformas foi realizada através do ModelSim Altera Starter Edition 10.1b.

No final desta seção, serão apresentados os resultados obtidos na construção desses circuitos em *hardware*, tais como, o número de elementos lógicos necessários para sua construção

¹⁹Na Ciência da computação teórica, a **corretude** de um algoritmo pode ser afirmada quando se diz que o algoritmo é correto com respeito à determinada especificação, do inglês *correctitude*.

²⁰Valor para 30% de taxa de ocupação disponível na Tabela 1

²¹O valor do RMSr do Filtro FDIP considerado é o mesmo para a taxa de ocupação de 30% disponível na Tabela 1, ou seja, 40,44.

e a taxa de operação. Esses resultados foram obtidos com a placa de FPGA Xilinx (XILINX, 2019) Kintex UltraScale+ KCU116, modelo XCKU5P-FFVB676-2-E (ULTRASCALE, 2019), com 216.960 LUTs e 433.920 Ffs. Essa placa está disponível na versão Vivado Webpack 2019.1 (CHURIWALA, 2016), para uso sem custos. Ressalta-se que a FPGA a ser utilizada, de fato, no primeiro nível de *trigger* do ATLAS deverá ser a Xilinx Kintex UltraScale+ XCKU115-1FLVB1760C (ARGOS, 2017), com aproximadamente 1,45 milhões de elementos lógicos, frente aos 0,6 milhões de elementos lógicos disponíveis na placa da versão gratuita do Vivado utilizado.

6.2 DIAGRAMAÇÃO DO CIRCUITO

O diagrama do circuito descrito, a seguir, tem como objetivo demonstrar como será processado um vetor r de tamanho infinito, obtido de forma contínua, *free-running*, gerando ao final do circuito um vetor \hat{x} também infinito, porém com um atraso Δ em relação ao vetor de entrada. Parte desse atraso diz respeito ao janelamento do vetor r e cada janela desse vetor é aqui nomeada como vetor s . O vetor \hat{x} é o resultado da recuperação do vetor x convoluído com os ruídos da eletrônica do TileCal, além de outros ruídos intrínsecos ao sistema, conforme detalhado na Subseção 5.1.3.1. O Diagrama 13 (Apêndice E) apresenta os blocos do circuito completo implementado. Esse diagrama foi construído de forma a poder ser utilizado na representação de vários métodos, a serem implementados no futuro, uma vez que basta substituir o bloco $y_{i+1} + \mu(f - Ay_i)$ pela fórmula do método a ser utilizado.

Os cálculos matriciais realizados por esse circuito são executados de forma linear, sem a utilização de um circuito de *clock*, e podem ser facilmente identificados no Diagrama 13 (Apêndice E) por apresentarem preenchimento na cor cinza. Tais cálculos serão detalhados a seguir. Os blocos sem preenchimento possuem registradores sincronizados com o *clock* de 40MHz do LHC.

O circuito do Diagrama 13 (Apêndice E) será apresentado sobre duas óticas distintas e complementares, sendo uma sobre o aspecto físico do circuito e outra sobre o seu comportamento lógico, nas próximas duas subseções.

6.2.1 Estrutura física do circuito

Os dados de entrada, vindos de um conversor ADC instalado em um canal de leitura do TileCal, apresentam-se de forma serial a uma taxa de 40MHz, sincronizados com a frequência de colisões do LHC (vetor r , *free-running*). Esses dados são paralelizados em janelas (de mesmo tamanho do trem de *bunches* do LHC), pelo conversor *Serial Input / Parallel Output* (SIPO), para a formação do vetor s (um vetor s para cada janela do vetor r). O vetor s recém-formado e já em paralelo, serve de entrada para dois circuitos diferentes, na saída do conversor SIPO, onde:

- é multiplicado pela pseudo-inversa da matriz H , com o fator ς já aplicado, conforme descrito na Subseção 5.5.1, aqui representada como \hat{H}^+ , formando um vetor y_0 resultante de \hat{H}^+s . Esse novo vetor é armazenado em um banco de registradores chamado *REG bank IN*;
- é multiplicado pela matriz H , formando um vetor f , resultante de $H^T s$, que é armazenado

em um banco de registradores chamado *REG bank fix*.

Os cálculos desses vetores f e y_0 ocorrem apenas uma vez, logo após a formação do vetor s . O vetor f é mantido constante durante todas as iterações do circuito.

Os dados armazenados no banco de registradores *fix* (*REG bank fix*), são utilizados na parte constante da Equação 59, ou seja, $H^T s$. Essa parte da Equação computa o produto entre o vetor de entrada e a matriz H otimizada. Os dados armazenados no banco de registradores *IN* (*REG bank IN*) são repassados, já com ganho, para a entrada de um circuito Multiplexador (*MUX*), na primeira iteração do circuito (porta a do *MUX*).

Como os blocos destacados em cinza são executados de forma combinacional em um único ciclo de *clock*, os dados que saem do *MUX* serão registrados no banco de registradores de saída (*REG bank OUT*), mas após passar por $\iota \times$ *Iteration Block* (IB), onde ι representa a quantidade de IB. A cada ciclo de *clock* e enquanto o conversor SIPO prepara a próxima janela de amostras de sinais, o circuito *MUX*, chaveado para a porta b , realimenta o circuito de iterações, a partir dos dados armazenados no *REG bank OUT*. Significa que, para uma janela amostral composta por um padrão $48b7e$, o circuito de iterações é alimentado pela porta a do *MUX* $1 \times$ e realimentado pela porta b do *MUX* $54 \times$, totalizando uma janela amostral de tamanho 55 (*window size* = 55). Ao fazer esse processo uma quantidade $\iota \times$ tem-se o equivalente a $\iota \times$ *window size* iterações. Por fim, um circuito conversor *Parallel Input / Serial Output* (PISO) serializa os dados de saída e remove o ganho dado na entrada inicial (porta a do *MUX*), de tal forma que os dados são apresentados, também em sequência, na saída do circuito.

Como todo o procedimento descrito é executado de forma síncrona, do ponto de vista exterior, esse processamento se comporta como um sistema *free-running* com uma latência igual a $2 \times$ *window size*, para *window size* = 55 e *clock* de $40MHz$ (25ns por amostra), mais 2 ciclos de *clock*, necessários na inicialização do circuito. A latência total fica em torno de $2 \times 55 \times 25ns + (2 \times 25ns)$, ou $2,8\mu s$. Essa latência obtida está acima da capacidade atual de processamento, de $2,5\mu$ (CZODROWSKI, 2015), que passará a ser de $10\mu s$ (SUMMERS *et al.*, 2020) para a Fase II. Esse tempo total de processamento não será exclusivo para esse circuito, no entanto menos de um terço dos $10\mu s$ será necessário para a operação do circuito.

6.2.2 Comportamento do circuito

O circuito se comporta como uma arquitetura *pipeline* de três estágios, durante sua execução:

- a. o primeiro estágio do *pipeline* comprehende o circuito SIPO, o bloco que computa as operações de $f = H^T s$ e o pré-processamento de y_0 . O circuito SIPO constrói uma janela de 55 amostras (o vetor s) armazenando as mais recentes n amostras consecutivas de $r[n]$. Quando os registradores internos do circuito SIPO estão totalmente ocupados, ambos os vetores, f e y_0 , são calculados. Na próxima borda de subida do *clock*, os 48 elementos resultantes, para ambos os vetores, são armazenados em bancos de registradores de entrada. Este processo é repetido para cada janela (ciclos de 55 *clocks*), sem interrupção;

- b. a segunda etapa do *pipeline* compreende os blocos entre os bancos registradores de entrada e de saída. Os blocos centrais, que estão na cor cinza, são responsáveis por executar uma iteração SSFi. Essa etapa possui um caminho de retroalimentação sincronizado com o *clock* de dados (40 MHz), a fim de computar várias iterações antes que a próxima janela seja entregue pela primeira etapa do *pipeline*. O vetor reconstruído resultante ($\hat{\mathbf{y}}_i$) é atualizado no banco de registradores de saída para cada ciclo de *clock*;
- c. a terceira e última etapa é executada pelo bloco do circuito PISO. Esse circuito serializa o vetor reconstruído presente no banco de registradores de saída, entregando a informação de forma *free-running*, com um atraso Δ , formando, assim, o vetor $\hat{\mathbf{x}}[n - \Delta]$ (que é o vetor $\hat{\mathbf{x}}$ *free-running* com atraso Δ), sincronizado com o *clock* do ADC (40 MHz). Dessa forma, o processamento em janela do sistema proposto torna-se transparente.

O sistema opera em uma arquitetura de *pipeline* em três estágios, como descrito anteriormente. Assim, três janelas consecutivas são avaliadas simultaneamente, uma em cada etapa. A primeira fase abriga a janela mais recente e envia os dados em paralelo para a fase seguinte.

As iterações do SSFi são implementadas até a segunda etapa do *pipeline*. No primeiro *clock*, o multiplexador é comutado para o registro de entrada. O valor inicial do algoritmo, desse modo, é \mathbf{y}_0 , como esperado. Esse sinal se propaga através dos blocos que computam uma iteração SSFi. Para os próximos 54 ciclos de *clock* consecutivos, o multiplexador aponta para o registro de saída, que atualiza o vetor reconstruído, a cada ciclo de *clock*, em um fluxo de dados de *feedback*. Nessa etapa, se apenas um bloco de iteração SSFi for implementado, é possível computar 55 iterações antes que a primeira etapa envie uma nova janela. Portanto, uma cascata de ι blocos permite a execução de $55 \times \iota$ iterações.

A última etapa, simplesmente, serializa a mais recente janela reconstruída entregue pela segunda etapa. Como consequência dessa arquitetura proposta, o atraso de resposta Δ é fixado em duas vezes o tamanho da janela ($2 \times 55 = 110$), independentemente do número de iterações (que é um múltiplo de 55, para o padrão de colisão igual a $48b7e$).

6.3 DESENVOLVIMENTO DO CIRCUITO COMPLETO

Nesta seção, será desenvolvida a adaptação da versão em Python (MCKINNEY, 2017) da variação SSFi, para Verilog (PALNITKAR, 2003) e a linguagem utilizada na construção dos circuitos em FPGA (PARAB; GAD; NAIK, 2018), destacando os pontos onde foram necessários adaptações matemáticas. Os esquemáticos dos circuitos servem de base para o desenvolvimento de circuitos com diferentes tamanhos de janelas amostrais. Cabe lembrar que todos os resultados das métricas analisadas foram obtidos através da implementação de circuitos para uma janela amostral de $48b7e$.

6.3.1 Adaptação de matrizes

A matriz H , presente na primeira linha do Algoritmo 13, é formada pelas sete amostras de referência do sinal do TileCal se deslocando a cada coluna. Tais valores se repetem em uma diagonal, ficando os outros valores iguais a zero. Baseado nessa característica, cada um dos valores foi transformado em um parâmetro de configuração no código gerado em Verilog e, mapeando o uso de cada valor no cálculo matricial, pode-se substituir os cálculos matriciais por operações de soma e de multiplicação otimizadas, de forma a não efetuar cálculos com valores iguais a zero, transformando, assim, um cálculo matricial em um cálculo linear.

Uma representação gráfica desse mapeamento pode ser visualizada na Figura 27 (Apêndice E). O trecho do código Verilog, onde os valores não zero são mapeados para parâmetros, pode ser visualizado no Algoritmo 14, onde os valores em ponto flutuante foram deslocados à esquerda 17 bits, formando às constantes representativas da matriz H . Todos esses valores são armazenados em parâmetros com 1 bit a mais para o sinal.

Algoritmo 14: Mapeamento dos valores da Matriz H em parâmetros Verilog, deslocamento à esquerda de 17 bits, ou $<< 17$

```

1      :
2  #((
3      parameter signed [18:0] h00 = 19'd3,           // 0,00002304
4      parameter signed [18:0] h01 = 19'd2258,        // 0,0172264
5      parameter signed [18:0] h02 = 19'd59303,       // 0,452445
6      parameter signed [18:0] h03 = 19'd131072,      // 1,0
7      parameter signed [18:0] h04 = 19'd73834,       // 0,563307
8      parameter signed [18:0] h05 = 19'd19572,       // 0,149335
9      parameter signed [18:0] h06 = 19'd5552         // 0,0423598
10 )
11   :

```

Considerando uma matriz H , compatível com o padrão de colisões adotado pelo LHC até o final de 2018 (ou seja, $48b7e$), que possui 2.592 elementos, sendo apenas 336 não nulos, tem-se uma taxa de esparsidade na ordem de 87,03%. Na Figura 28 (Apêndice E) pode-se observar graficamente a distribuição dessa densidade.

Várias otimizações foram realizadas, para reduzir recálculos desnecessários. A de maior impacto foi a criação de uma matriz de autocorrelação A , que é o resultado da multiplicação da matriz H por sua transposta. Tal cálculo faz parte da primeira linha do Algoritmo 13 e a mesma técnica aplicada na matriz H também foi aplicada na matriz de autocorrelação A , por se tratar de uma matriz *Toeplitz* (BÖTTCHER; GRUDSKY, 2000), ou seja, além dos valores se repetirem em uma diagonal também são espelhados, em relação à diagonal principal. A representação gráfica

do mapeamento da matriz de autocorrelação A pode ser visualizada na Figura 29 (Apêndice E) e o trecho de código Verilog equivalente está no Algoritmo 15. Nota-se, na Figura 29 (Apêndice E), que a diagonal principal foi destacada, de forma a facilitar a visualização do espelhamento dos valores.

Algoritmo 15: Mapeamento dos valores da Matriz de Autocorrelação A em parâmetros Verilog, deslocamento à esquerda de 20 *bits*, ou $<< 20$.

```

1      :
2  #((
3  parameter signed [21:0] a00 = 22'd1621532, // 1,54641329854284
4  parameter signed [21:0] a01 = 22'd1168107, // 1,11399364702225
5  parameter signed [21:0] a02 = 22'd466930, // 0,44529883180639
6  parameter signed [21:0] a03 = 22'd125465, // 0,11965246577979
7  parameter signed [21:0] a04 = 22'd22808, // 0,02175096274828
8  parameter signed [21:0] a05 = 22'd769, // 0,00073314753712
9  parameter signed [21:0] a06 = 22'd1, // 0,00000097597
10 )
11      :

```

Ainda para a matriz de autocorrelação A , notou-se, experimentalmente, que os deslocamentos de *bits* necessários para os cálculos eram equivalentes aos deslocamentos efetuados na matriz H , acrescidos de 5 *bits*. Como a matriz H precisou de um deslocamento de 17 *bits* à esquerda, a matriz de autocorrelação A precisou de um deslocamento de 22 *bits* também à esquerda. Tais valores foram armazenados em uma variável contendo um 1 *bit* extra para o sinal. Na análise da densidade dessa matriz, segundo o padrão de colisão de 48b7e, constata-se que ela possui 2.304 elementos, sendo 582 não nulos, perfazendo assim uma taxa de esparsidade da ordem de 74,74%, conforme pode ser conferida graficamente na Figura 30 (Apêndice E).

A matriz H^+ (pseudo-inversa da matriz H) possui 2.592 elementos não nulos e distintos entre si, o que tornaria inviável a sua utilização em um circuito de *hardware*, nos moldes do objetivo desse trabalho. Conforme a análise realizada na Subseção 5.5, porém a possibilidade de se aplicar um fator de redução da densidade na matriz H^+ , de forma a possibilitar o seu uso na construção do circuito é concebível.

Seguindo as conclusões obtidas na Subseção 5.5.1, ao aplicar um fator de redução de densidade, de forma a manter apenas os valores na faixa de $\pm 1,4$ ($\varsigma = 1,4$), apenas 127 elementos não nulos continuam fazendo parte da matriz H^+ ($\omega = 127$). Com isso, ela sai de uma taxa de esparsidade da ordem de 0% para uma taxa de esparsidade da ordem de 95,10%, considerando os 2.592 elementos obtidos de uma configuração compatível com o padrão de colisão 48b7e. Na Figura 31 (Apêndice E), pode-se visualizar graficamente essa densidade.

O Algoritmo 16 contém uma amostragem do mapeamento da matriz H^+ , com $\omega = 127$

(matriz $\hat{\mathbf{H}}^+$) nos parâmetros que serão utilizados como constantes pelo circuito que realizará os cálculos de forma linear. Na construção do circuito, os parâmetros relativos à matriz $\hat{\mathbf{H}}^+$ foram nomeados com a letra "b", assim como no código originalmente escrito em Python a matriz $\hat{\mathbf{H}}^+$ é nomeada como matriz \mathbf{B} .

Algoritmo 16: Mapeamento dos valores da Matriz $\hat{\mathbf{H}}^+$ em parâmetros Verilog,
deslocamento à esquerda de 17 bits, ou << 17

```

1      :
2  #((
3 parameter signed [18:0] b00 = 19'd54218, // 1.654605047455801
4 parameter signed [18:0] b01 = 19'd67530, // 2.06085849765563
5 parameter signed [18:0] b02 = 19'd73529, // 2.243925545422725
6      :
7 parameter signed [18:0] b124 = 19'd74757, // 2.28139331602365
8 parameter signed [18:0] b125 = 19'd70646, // 2.15595549569686
9 parameter signed [18:0] b126 = 19'd58536, // 1.78638806348108
10 )
11      :

```

Ainda no Algoritmo 16, observa-se que a quantidade de *bits* utilizada nos parâmetros não é suficiente para a conversão total dos valores em ponto flutuante dos elementos da matriz $\hat{\mathbf{H}}^+$, em seu equivalente em ponto fixo, deslocado à esquerda uma determinada quantidade de *bits*. Foi o suficiente, porém, para se atingir o valor de RMSr desejado.

6.3.2 Circuitos auxiliares

Fazem parte dos circuitos auxiliares dessa implementação os módulos:

- Serial Input / Parallel Output* (SIPO)
- Cálculo linear de $\mathbf{H}^T \mathbf{s}$;
- Cálculo linear de $\hat{\mathbf{H}}^+ \mathbf{s}$;
- Círculo Multiplexador *MUX*;
- Parallel Input / Serial Output* (PISO).

Para possibilitar um melhor entendimento dos circuitos auxiliares aqui implementados, optou-se por localizá-los de acordo com o comportamento do circuito completo. Tal comportamento foi detalhado na Subseção 6.2.2.

6.3.2.1 Primeiro pipeline

Como o ambiente onde o método será implementado é um ambiente *free running* e, também, como o próprio LHC se utiliza de uma estrutura de trens de *bunches*, os sinais vindos

da eletrônica do sistema de aquisição de dados do TileCal já chegam ao circuito em forma de janelas. Por isso, a parte do código do Algoritmo 13, entre as linhas 7 e 8, são equivalentes às linhas em Verilog do módulo SIPO, cujo trecho de código pode ser visualizado no Algoritmo 17.

Algoritmo 17: Trecho de código do circuito SIPO, com destaque para o alinhamento da janela em relação ao tamanho do trem de *bunches* do LHC, utilizando-se para isso um contador e um sinal de *enable*

```

1      :
2      reg signed [10:0] r1 [54:0];
3      reg [6:0] cont = 0;
4
5      assign en = cont == 55;
6
7      always @ (posedge clk)
8      begin
9          if (cont <= 54)
10             cont <= cont + 1;
11         else
12             cont <= 1;
13         if (en == 1'b1)
14             begin
15                 r1[ 0] <= x;
16                 r1[ 1] <= r[ 0];
17                 :
18                 r1[53] <= r[52];
19                 r1[54] <= r[53];
20             end
21         end
22     :

```

Ao se considerar um padrão de colisão equivalente a $48b7e$, o bloco SIPO possuirá dois bancos com 55 registradores cada. Esse número de registradores é definido na construção do circuito, de acordo com o padrão de colisões do LHC ($48b7e$, $8b4e$, etc.). O primeiro banco é configurado como registradores de deslocamento, de modo a permitir a paralelização dos dados. Completado um ciclo de deslocamento de 55 amostras, elas são copiadas para um novo banco de registradores, que mantêm os dados registrados por 55 ciclos de *clock*, chamado *Input Registers Bank*.

A janela amostral formada por esse banco de registradores é controlada por sinais de

enable, através de um bloco contador chamado de *Enable Generator*. Com isso, no fim da contagem de 55 ciclos de *clock* o *Input Registers Bank* recebe uma nova janela amostral. Com essa configuração de dois bancos de registradores, o segundo banco mantém os dados da janela atual intactos durante 55 ciclos de *clock*, para o processamento das iterações, enquanto o primeiro banco monta a próxima janela a ser processada. O *Input Registers Bank* alimenta dois outros circuitos auxiliares, como o circuito responsável pelo cálculo presente na linha 9 do Algoritmo 13 e o circuito responsável pelo cálculo presente na linha 10 do Algoritmo 13. Ambos os circuitos realizam seus cálculos no primeiro ciclo de *clock*, após a montagem da janela amostral no primeiro *pipeline*. Os resultados desses cálculos alimentarão os circuitos do segundo *pipeline*. A representação esquemática desses circuitos pode ser constatada no Diagrama 14 (Apêndice E).

A criação do circuito $\mathbf{H}^T s$ foi necessária, porque os cálculos que ele efetua são reutilizados dentro das iterações do circuito principal e, ainda, devido às otimizações efetuadas para o cálculo matricial (Subseção 6.3.1). Esse circuito $\mathbf{H}^T s$ acabou se transformando em uma série de filtros FIR de ordem 7 e a quantidade de Filtros FIR é definida de acordo com a quantidade de *bunches* onde ocorreram colisões na janela amostral.

Considerando o padrão de colisões 48b7e utilizado, o circuito $\mathbf{H}^T s$ é formado por 48 Filtros FIR de ordem 7, conforme se pode verificar no Algoritmo 18. Para o padrão 8b4e seriam necessários 8b Filtros FIR (ou simplesmente 8 Filtros FIR de ordem 7). Os resultados obtidos por esse circuito $\mathbf{H}^T s$ são armazenados em um banco de registradores chamado *REG Bank fix*, possibilitando o seu uso pelos outros circuitos do segundo *pipeline*.

Algoritmo 18: Trecho do código do circuito $\mathbf{H}^T s$, apresentando uma série de Filtros FIR

```

1      :
2 assign outHr00 = (h00 * in00) + ... + (h05 * in05) + (h06 * in06);
3 assign outHr01 = (h00 * in01) + ... + (h05 * in06) + (h06 * in07);
4 assign outHr02 = (h00 * in02) + ... + (h05 * in07) + (h06 * in08);
5      :
6 assign outHr45 = (h00 * in45) + ... + (h05 * in50) + (h06 * in51);
7 assign outHr46 = (h00 * in46) + ... + (h05 * in51) + (h06 * in52);
8 assign outHr47 = (h00 * in47) + ... + (h05 * in52) + (h06 * in53);
9      :

```

Em relação ao circuito necessário para a implementação do cálculo $\mathbf{H}^+ s$, não foi possível utilizar-se de filtros FIR equivalentes, por conta de a matriz $\hat{\mathbf{H}}^+$ aplicada na construção desse circuito possuir 127 elementos distintos entre sim ($\omega = 127$). Assim, foi necessária a criação de 127 parâmetros para o armazenamento de seus valores e, como consequência da disposição desses elementos na matriz, os cálculos realizados por esse circuito apresentaram um formato de

filtros FIR de coeficientes assimétricos. Um trecho do código Verilog, do circuito, utilizado no cálculo \hat{H}^+s , pode ser encontrado no Algoritmo 19, onde a calibração do alinhamento de *bits* à direita garante que o resultado apresente um determinado ganho. O resultado é armazenado em *REG Bank IN*, que está ligado na entrada da porta *a* do circuito de *MUX* acoplado no início do segundo *pipeline*.

Algoritmo 19: Trecho do código do circuito \hat{H}^+s , utilizando um cálculo matricial otimizado.

```

1      :
2 assign outBs00 = 0;
3 assign outBs01 = ((b00 * s[4])) >>> align;
4      :
5 assign outBs04 = ((b04 * s[7]) + (b05 * s[8])) >>> align;
6 assign outBs05 = ((b06 * s[7]) + (b07 * s[8]) + (b08 * s[9])) >>> align;
7      :
8 assign outBs43 = ((b120 * s[45]) + (b121 * s[46]) + (b122 * s[47])) >>> align;
9 assign outBs44 = ((b123 * s[46]) + (b124 * s[47])) >>> align;
10     :
11 assign outBs46 = ((b126 * s[49])) >>> align;
12 assign outBs47 = 0;
13     :

```

6.3.2.2 Segundo pipeline

Na interface entre o primeiro e o segundo *pipeline*, há um circuito de *MUX* cujo trecho do código pode ser visualizado no Algoritmo 20. Sua porta *a* recebe os dados vindos do primeiro *pipeline*, já com um ganho pré-determinado, e sua porta *b* é voltada aos circuitos presentes no segundo *pipeline*.

Algoritmo 20: Trecho do código do circuito de MUX, que controla a origem dos dados a serem processados pelo restante do algoritmo, podendo ser do SIPO ou do banco de registradores

```

1      :
2      always @ (*)
3      begin
4          if (en == 1'b1)
5              begin
6                  c00 <= a00;
7                  c01 <= a01;
8                  :
9                  c46 <= a46;
10                 c47 <= a47;
11             end
12         else
13             begin
14                 c00 <= b00;
15                 c01 <= b01;
16                 :
17                 c46 <= b46;
18                 c47 <= b47;
19             end
20         end
21     :

```

Para o segundo *pipeline*, um circuito auxiliar específico foi desenvolvido para efetuar o cálculo do produto do sinal de entrada com a matriz de autocorrelação A na forma otimizada. Um trecho do código desse circuito pode ser visualizado no Algoritmo 21.

Algoritmo 21: Cálculo matricial otimizado utilizando a matriz A

```

1      :
2 assign Ax[ 0] = (a00 * in00) + ... + (a06 * in06);
3 assign Ax[ 1] = (a00 * in01) + (a01 * (in00 + in02)) + ... + (a06 * in07);
4 assign Ax[ 2] = (a00 * in02) + (a01 * (in01 + in03)) + ... + (a06 * in08);
5      :
6 assign Ax[45] = (a00 * in45) + (a01 * (in44 + in46)) + ... + (a06 * in39);
7 assign Ax[46] = (a00 * in46) + (a01 * (in45 + in47)) + ... + (a06 * in40);
8 assign Ax[47] = (a00 * in47) + ... + (a06 * in41);
9      :

```

Nota-se que, diferentemente do que ocorre com a utilização das otimizações aplicadas na matriz H , os cálculos envolvendo a matriz de autocorrelação A consideraram a sua simetria. Com isso, ela pode ser vista como uma série de filtros FIR de coeficientes simétricos (coeficientes espelhados), possibilitando então o uso de uma estrutura de processamento onde um único coeficiente, que é refletido, é usado uma única vez para diminuir o número de multiplicações. Essa estrutura é usada em filtros FIR de Fase Linear (MITRA, 2010). Esse circuito é o responsável pelo cálculo da multiplicação entre parênteses na linha 12 do Algoritmo 13.

Um último circuito auxiliar que merece destaque, é o responsável pelo *loop* interno presente entre as linhas 11 e 15 do Algoritmo 13. Esse circuito executa uma determinada quantidade de blocos do circuito principal e, a cada execução, um número equivalente de iterações é processado pelo algoritmo principal. Assim, para $\iota = 3$ blocos de uma janela de tamanho 55, são executadas 165 iterações equivalentes pelo algoritmo principal. Um trecho desse código pode ser visualizado no Algoritmo 22, onde se pode notar que ao trocar *Algorithm* por SSFi ou por outro método qualquer, esse outro método passa a ser utilizado.

Algoritmo 22: Trecho do código do *loop* de execução da variação SSFi

```

1      :
2  genvar itr;
3  generate
4    for (itr = 1; itr <= totalITR; itr = itr+1)
5    begin: block
6      Algorithm#.bits(bits)) Algorithm(
7        w_HTs[0], w_HTs[1], w_HTs[2], ..., w_HTs[46], w_HTs[47],
8        x_algo[0][itr-1], x_algo[1][itr-1], ..., x_algo[47][itr-1],
9        x_algo[0][itr], x_algo[1][itr], ..., x_algo[47][itr]);
10   end
11 endgenerate
12 :

```

6.3.2.3 Terceiro pipeline

Outro circuito responsável pela gestão do enjanelamento de sinais é o conversor PISO, cujo esquemático pode ser visualizado no Diagrama 15 (Apêndice E). Como o Algoritmo PISO é muito semelhante ao Algoritmo 17, optou-se por não apresentá-lo neste trabalho. Esse circuito recria a janela de forma semelhante ao executado na linha 16 do Algoritmo 13.

O bloco PISO comprehende 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 55 ciclos de *clock*. O Diagrama 15 (Apêndice E) mostra os blocos desse circuito, onde a função de *load* é implementada através de multiplexadores.

6.3.3 Circuito principal

Maiores detalhes a respeito do circuito SSF implementado neste trabalho serão apresentados nesta subseção.

Os circuitos apresentados até aqui podem ser utilizados para a implementação de vários métodos *Shrinkage*. O gráfico de fluxo de uma iteração do método SSF está apresentado no Diagrama 16 (Apêndice E). Dessa forma, basta trocar a indicação do circuito principal a ser utilizado, para conduzir os mais variados testes.

Vale ressaltar que, apesar dessa implementação dizer respeito a variação SSFi, ela se destaca do método SSF apenas na inicialização prévia do vetor y_0 . Essa alteração no vetor serve, apenas, como uma primeira entrada para o método SSF, conforme originalmente divulgado

por Elad (2010). Com isso, o circuito principal é, de fato, o SSF que contou com um circuito auxiliar responsável pela inicialização do vetor y_0 (Subseção 6.3.2, mais especificamente Algoritmo 19).

Conforme pode ser verificado no circuito principal do método (Diagrama 16 Apêndice E), ele possui duas fontes de entrada de sinais, sendo a primeira vinda do circuito auxiliar $H^T s$. Como dito anteriormente, esse circuito é uma série de filtros FIR, aplicados ao sinal originalmente vindo do sistema de aquisição de dados. A segunda fonte de entrada de dados vem da seleção feita pelo circuito de *MUX*, pela porta c , sincronizado com o *Enable Generator*, acoplado ao circuito SIPO. A entrada do *MUX* pode ser tanto o sinal vindo diretamente do sistema de aquisição de dados – porta a conectada ao circuito SIPO – quanto o sinal vindo do banco de registradores acoplado ao último bloco do circuito principal, porta b conectada ao circuito PISO. Ambas as conexões são intermediadas por bancos de registradores. Os passos seguidos pelo circuito SSF são:

- a. execução da multiplicação matricial otimizada, apresentada no Algoritmo 21, ao sinal vindo do circuito de *MUX*;
- b. aplicação de um deslocamento de *bits* à esquerda, para o alinhamento de *bits*, necessário para o próximo passo, ao sinal vindo de $H^T s$;
- c. cálculo da diferença dos resultados dos dois passos anteriores (Algoritmo 23), vetor B presente na linha 12 do Algoritmo 13;

Algoritmo 23: Cálculo do vetor B

```

1          :
2
3      assign B[ 0] = (H^Ts00 << (bits - 5)) - Ax[ 0];
4      assign B[ 1] = (H^Ts01 << (bits - 5)) - Ax[ 1];
5          :
6      assign B[46] = (H^Ts46 << (bits - 5)) - Ax[46];
7      assign B[47] = (H^Ts47 << (bits - 5)) - Ax[47];
8          :

```

- d. deslocamento dessa diferença à direita, o equivalente a multiplicação pelo valor de μ , ou seja, $>>> \mu$;
- e. após o deslocamento anterior, o resultado é somado ao sinal vindo do *MUX*, deslocado à esquerda uma determinada quantidade de *bits*, conforme Algoritmo 24;

Algoritmo 24: Cálculo do vetor auxiliar xB , para o método SSF

```

1      :
2      assign xB[ 0] = (in00 << align) + (B[ 0] >>> mu);
3      assign xB[ 1] = (in01 << align) + (B[ 1] >>> mu);
4      :
5      assign xB[46] = (in46 << align) + (B[46] >>> mu);
6      assign xB[47] = (in47 << align) + (B[47] >>> mu);
7      :

```

- f. a soma anterior é diminuída o valor de λ , já deslocado à esquerda uma determinada quantidade de *bits* e o resultado dessa subtração é, então, deslocado à direita (Algoritmo 25), finalizando assim os cálculos da linha 13 do Algoritmo 13 ²². No Diagrama 16 (Apêndice E) esse passo aparece circulado por uma linha tracejada, indicando que ele pode ser ignorado a depender do valor de λ ;

Algoritmo 25: Cálculo do vetor auxiliar tmp , para o método SSF

```

1      :
2 assign tmp[0] = (xB[ 0] - (lambda << (align-gain-1))) >>> align;
3 assign tmp[1] = (xB[ 1] - (lambda << (align-gain-1))) >>> align;
4      :
5 assign tmp[6] = (xB[46] - (lambda << (align-gain-1))) >>> align;
6 assign tmp[7] = (xB[47] - (lambda << (align-gain-1))) >>> align;
7      :

```

- g. Então, um conhecimento especialista zera os valores negativos, Algoritmo 26, linha 14 do Algoritmo 13, finalizando assim o circuito SSF.

²²De acordo com a Seção 5, o valor de λ tende a zero, alterando o código Verilog de "assign tmp[i] = (xB[i] - (λ << (align-gain-1))) >>> align;" para apenas "assign tmp[i] = xB[i] >>> align;". De qualquer forma, o algoritmo em Python é capaz de gerar o circuito para o método SSF, prevendo o cenário onde o valor de λ não seria zero.

Algoritmo 26: Aplicação do conhecimento especialista, no método SSF

```

1      :
2      assign out00 = tmp[ 0][bits] == 1 ? 0 : tmp[ 0];
3      assign out01 = tmp[ 1][bits] == 1 ? 0 : tmp[ 1];
4      :
5      assign out46 = tmp[46][bits] == 1 ? 0 : tmp[46];
6      assign out47 = tmp[47][bits] == 1 ? 0 : tmp[47];
7      :

```

6.4 RESULTADOS

Para a geração dos resultados, os circuitos foram concebidos a partir das configurações presentes na Tabela 12 e construídos, de forma automática, em Verilog, utilizando-se uma ferramenta desenvolvida em Python, especialmente projetada para esse fim. O uso dessa ferramenta permite a construção de circuitos para qualquer variação das métricas utilizadas, incluindo o padrão de colisões utilizado pelo LHC. Ou seja, é possível criar um circuito específico para qualquer um dos 10.000 canais do TileCal e adaptado a qualquer padrão de colisão do LHC.

O sinal utilizado para os testes de execução dos circuitos criados foi o mesmo para a geração da Tabela 12, descrito na Seção 6.1. Foram avaliados os resultados de reconstrução de energia através da minimização do erro médio quadrático da distribuição de erro entre a diferença do sinal alvo e o sinal recuperado pelo circuito construído (aqui chamado RMSr). Esse sinal foi simulado de um sinal recuperado, a partir de um conversor ADC, instalado em um canal de leitura do TileCal. As características simuladas desse sinal são as mesmas definidas na Seção 5.1.

Uma vez definidos o sinal a ser utilizado para testar os circuitos, as configurações dos circuitos que seriam testadas e a frequência máxima de operação do circuito, efetuou-se a construção dos circuitos obtendo, assim, a quantidade de elementos lógicos necessários à construção de cada um. Todos esses valores estão agrupados de acordo com:

- a. na Tabela 13 se encontram os resultados obtidos da aplicação da configuração que obteve os menores resultados de RMSr, de acordo com a simulação realizada em Python de um circuito em ponto fixo. Tal configuração está definida na coluna *Melhor Configuração* da Tabela 12;

Tabela 13 - Resultados da implementação em hardware da melhor configuração para se obter o menor valor de RMSr

SSFi com matriz \hat{H}^+ com $\omega = 127$			
Iterações	55	110	165
RMSr (ponto flutuante)	29,41	27,09	26,51
RMSr (ponto fixo)	30,87	28,69	28,13
MHz	29,04	34,86	39,11
LUT	31.378	59.927	87.160
FF	2.097	2.098	2.097
DSP	435	503	571
IO	23	23	23

Fonte: Elaborada pelo autor (2021).

- a. na Tabela 14 encontram-se os resultados obtidos da aplicação da configuração que obteve os menores resultados de RMSr, apenas em relação ao Filtro FDIP de ordem 26 em ponto flutuante, de acordo com a simulação realizada em Python de um circuito em ponto fixo. Tal configuração está definida na coluna *Melhor apenas que FDIP ponto flutuante* da Tabela 12.

Tabela 14 - Resultados da implementação em hardware da melhor configuração para se obter o menor circuito

SSFi com matriz \hat{H}^+ com $\omega = 127$			
Iterações	55	110	165
RMSr (ponto flutuante)	29,41	27,09	26,51
RMSr (ponto fixo)	32,42	30,58	30,15
MHz	27,99	34,47	39,49
LUT	26.026	47.753	69.702
FF	2.360	2.360	2.360
DSP	386	454	522
IO	23	23	23

Fonte: Elaborada pelo autor (2021).

Os circuitos foram avaliados seguindo dois critérios:

- a. inicialmente, o circuito deveria apresentar o menor valor de RMSr, em relação ao valor de RMSr obtido com um código implementado em Python em ponto flutuante; e
- b. o circuito deveria apresentar um RMSr igual ou menor ao obtido pela implementação em ponto flutuante do filtro FDIP de ordem 26.

O tempo disponível para a operação do algoritmo, considerando uma janela composta por 48 *bunches* com colisões e 7 *bunches* sem colisões, é de 55 ciclos de *clock*, sendo esse *clock* sincronizado com o LHC a uma taxa de 40MHz, ou 25ns entre colisões, permitindo assim o

seu uso no primeiro nível de *trigger* do ATLAS. Mesmo o circuito operando a essa taxa de 40MHz, em uma janela de tamanho 55 a latência final fica em torno de $2 \times 55 \times 25\text{ns}$, mais a inicialização do circuito, ou $2 \times 25\text{ns}$, totalizando $2,8\mu\text{s}$, como se pode observar nas Tabelas 13 e 14. Tanto as versões dos circuitos que apresentam os menores valores de RMS da distribuição de erro, quanto as que apresentam os menores gastos de elementos lógicos, em sua construção, conseguem operar dentro da latência máxima permitida, que será de $10\mu\text{s}$ (SUMMERS *et al.*, 2020) para a Fase II.

6.5 CONCLUSÃO DA SEÇÃO 6

Conforme desenvolvido, nesta última seção, a implementação da variação SSFi se deu por ter apresentado os melhores resultados de detecção e estimação, conforme análises realizadas na seção anterior. Na próxima seção, será apresentada a conclusão geral deste trabalho e a proposição de trabalhos futuros.

7 CONCLUSÃO E PRÓXIMOS PASSOS

Este trabalho teve como principal objetivo a implementação de um método *on-line* de estimativa de energia para calorimetria em *hardware* reconfigurável (FPGA). Um método que fosse robusto em cenários de empilhamento de sinais, ao mesmo tempo que apresentasse melhores resultados que os obtidos pelos métodos atualmente em uso. Destarte, métodos baseados em representação esparsa de dados foram projetados, modificados e tiveram seus resultados comparados, de acordo com critérios de seleção previamente definidos, para operarem no ambiente *on-line* do ATLAS.

A metodologia aplicada na condução deste trabalho incluiu o teste de métodos de representação esparsa de dados das famílias *Greedy* e *Shrinkage*, de forma a implementar em FPGA aquele que melhor se adequou aos critérios de: menor complexidade de implementação em *hardware*; apresentou o menor erro de detecção e estimativa, em relação aos obtidos com o método atualmente em uso; apresentou latência compatível com as exigências do sistema de trigger L1 do ATLAS.

Os métodos da família *Greedy* abandonam a busca exaustiva e trabalham com uma série de atualizações termo a termo, adicionando elementos iterativamente, para reconstruir uma aproximação contendo um reduzido número de elementos não nulos. Tais métodos têm como objetivo minimizar as funções (P_0) ou $(P_{0,\epsilon})$. Dentre os métodos da família *Greedy*, foram analisados os métodos baseados no *Matching Pursuit*, a saber, a própria versão MP, a versão ortogonal (*Orthogonal-Matching-Pursuit*) e a *least square* (*Least-Square Orthogonal-Matching-Pursuit*). A ideia básica por trás dos métodos que variam a partir do MP, parte da suposição de que, havendo uma matriz \mathbf{A} que tenha $\text{spark}(\mathbf{A}) > 2$, o problema de otimização (P_0) tenha valor $\text{val}(P_0) = 1$ (na solução ótima).

Em relação à família de métodos *Shrinkage*, pode-se dizer que a técnica de limiarização, ou *Shrinkage*, desenvolvida por Donoho (1995), é uma técnica simples, porém muito eficiente em aplicações de remoção de ruído. Avanços recentes mostram que essas técnicas de *Shrinkage* auxiliam na busca por uma representação esparsa de um determinado sinal. Os métodos baseados nessa técnica formam a família de métodos de *Shrinkage*, ou Algoritmos Iterativos de Redução.

Os métodos aqui estudados têm o objetivo de minimizar uma função do tipo presente na Equação 47, que é uma forma generalizada da Equação 34. Uma proposta de desenvolvimento dessa equação (Subseção 4.4.1) chega a uma aplicação iterativa da Equação 56 e, assim, ao método *Separable Surrogate Functionals*. Todos os métodos apresentados neste trabalho, obtiveram melhor desempenho, tanto em relação aos atualmente utilizados baseados em Filtro Casado, quanto aos métodos propostos em trabalhos anteriores baseados em Filtros FIR.

Conforme pode ser conferido na Tabela 9, o método que obteve os menores valores de RMS da distribuição de erro foi o método LS-OMP, para baixas taxas de ocupação (até 20% de taxa de ocupação). Para as altas taxas de ocupação (taxas de ocupação acima de 20%), o método SSF obteve os menores valores de RMS da distribuição de erro, sendo que, em relação à análise

das curvas ROC, o método SSF obteve os melhores resultados para uma baixa faixa de *False Alarm* (até 20%).

Seguindo as especificações do ambiente onde esses métodos serão utilizados e a facilidade de implementação fornecida pelo método SSF, na Seção 6 foi efetuada sua implementação em *hardware*. Tomando como base as métricas obtidas dos métodos da família *Greedy*, recomenda-se que, em trabalhos futuros, um método híbrido seja desenvolvido, aproveitando a parte de detecção, principalmente do MP, que possui uma matemática mais aderente ao ambiente de ponto fixo.

Foi necessário, porém, desenvolver todo um circuito digital capaz de operar esses métodos iterativos em ambiente *on-line*, a uma taxa de $40MHz$. Tal circuito foi construído com sucesso em um ambiente simulado, utilizando a FPGA Xilinx Zynq UltraScale+ XCZU7EV-FFVC1156-2-E, disponível na versão Vivado Webpack 2018.3, mostrando resultados satisfatórios para a implementação *on-line* nessa FPGA, que é um modelo bem mais modesto que o modelo que será utilizado no ambiente real. Sendo assim, os testes indicam que a implementação no ambiente real tende também a ser um sucesso.

O circuito operando à taxa de $40MHz$, em uma janela de tamanho 55, a latência final fica em torno de $2 \times 55 \times 25ns$, mais a inicialização do circuito, ou $2 \times 25ns$, totalizando $2,8\mu s$, acima do limite da latência máxima atual de $2,5\mu s$ exigida por esse nível de *trigger*. Essa latência, porém, representa menos da metade da latência máxima permitida para a Fase II, que será de $6\mu s$, tornando concreta a proposta desse circuito para a implementação no primeiro nível de *trigger* do ATLAS já na Fase II. E ainda, como se pode observar nas Tabelas 13 e 14, tanto as versões dos circuitos que apresentam os menores valores de RMS da distribuição de erro, quanto as que apresentam os menores gastos de elementos lógicos em sua construção conseguem operar dentro da latência máxima permitida para a Fase II.

7.1 PRÓXIMOS PASSOS

Para os próximos passos serão consideradas, ainda, as propostas definidas a seguir.

- Avaliação de outros métodos da família Shrinkage*: serão avaliados ainda outros métodos da família *Shrinkage*, como:

- *Iterative-Reweighted Least-Squares* (IRLS), que converte normas não- ℓ_2 , em norma ℓ_2 , pela introdução de ponderações;
- *Stage-wise Orthogonal-Matching-Pursuit* (StOMP) que é especialmente adaptado para casos em que a matriz \mathbf{A} é aleatória, como em detecção compactada, embora possa ser usada em outros casos com algum sucesso;

- Uso de simulação oficial do ATLAS*: os sinais a serem utilizados nas validações dos métodos serão obtidos de simulações oficiais do ATLAS, utilizando-se do simulador AREUS (HORN, 2018; MADYSA, 2018), em substituição ao *Toy Simulator* atualmente utilizado neste trabalho;

- Uso de um processador em FPGA projetado para essa tarefa*: está sendo projetado um

processador em FPGA específico para ser utilizado na resolução dos cálculos matriciais presentes neste trabalho. Tal processador operará com uma alta frequência de *clock*, compatível com as FPGAs mais modernas, além de operar com números em ponto flutuante. Com esse processador especializado em cálculos matriciais, todos os cálculos que atualmente estão sendo feitos em paralelo serão executados em um processador sequencial, em uma frequência de *clock* bem maior, diminuindo em muito a quantidade de elementos lógicos utilizados;

- d. *Avaliação global do método na FPGA utilizado na fase II:* o método SSF e os outros métodos a serem estudados serão implementados na FPGA, que está definida para entrar em operação junto com a Fase II de atualizações. Com isso, tais métodos devem ser testados na FPGA Xilinx Kintex UltraScale+ XCKU115-1FLVB1760C, onde os dados de entrada são alimentados por canais de fibra óptica. A FPGA possui 1,45 milhão de elementos lógicos, de acordo com os dados da Tabela 13. O valor deverá ser suficiente para suportar cerca de 12 canais, em paralelo, da versão do método SSF projetado para rodar o equivalente a 165 iterações, que é o que possui o melhor equilíbrio entre as métricas avaliadas. A quantidade de canais é a mesma disponível em cada uma das *minidrawers* projetadas para a Fase II. Com isso, os dados serão processados a $40MHz$ por cada uma das 12 instâncias da implementação do método SSF, simultaneamente. No final serão reempacotados e enviados, também via fibra óptica, para o primeiro nível de *trigger* do ATLAS.

REFERÊNCIAS

- AAD, G. *et al.* The atlas experiment at the cern large hadron collider. *JINST*, v. 3, p. S08003, 2008.
- AAD, G. *et al.* Atlas pixel detector electronics and sensors. *JINST*, v. 3, p. P07007, 2008.
- AAD, G. *et al.* Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Phys. Lett.*, B716, p. 1–29, 2012.
- AAMODT, K. *et al.* The alice experiment at the cern lhc. *JINST*, v. 3, p. S08002, 2008.
- ABBOTT, B. *et al.* *The Evolution of the Region of Interest Builder for the ATLAS Experiment at CERN*. Geneva, 2015.
- ABDALLAH, J.; COLLABORATION, A. The Laser calibration of the ATLAS Tile Calorimeter during the LHC run 1. The Laser calibration of the Atlas Tile Calorimeter during the LHC run 1. *JINST*, v. 11, n. arXiv:1608.02791, p. T10005. 31 p, Aug 2016. Disponível em: <<https://arxiv.org/pdf/1608.02791.pdf>>.
- ABOLINS, M. A. The ATLAS trigger: Commissioning with cosmic rays. *J. Phys. Conf. Ser.*, v. 119, p. 022014, 2008.
- ABT, I.; COLLABORATION, H. The h1 detector at HERA. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Elsevier BV, v. 386, n. 2-3, p. 310–347, Feb 1997.
- ADRIANI, O. *et al.* Technical design report of the lhcf experiment: Measurement of photons and neutral pions in the very forward region of lhc. *CERN Document Server*, 2006.
- AGGLETON, R.; COLLABORATION, C. An fpga based track finder for the l1 trigger of the cms experiment at the high luminosity lhc. *Journal of Instrumentation*, v. 12, n. 12, p. P12019, 2017.
- AGUIAR, M. S. *et al.* Implementação em fpga de um método iterativo de deconvolução para operar no sistema de trigger do experimento atlas. In: *XXII Encontro Nacional de Modelagem Computacional e X Encontro de Ciência e Tecnologia de Materiais*. Juiz de Fora: Essentia Editora, 2019.
- AGUIAR, M. S. *et al.* Arquitetura multi-core de processadores reconfiguráveis para reconstrução online de energia no calorímetro hadrônico do atlas. In: *XXIII Congresso Brasileiro de Automática (CBA 2020)*. Porto Alegre, Brazil: [s.n.], 2020.
- AIRAPETIAN, A. Atlas: Detector and physics performance technical design report. volume 1. *CERN Document Server*, 1999.
- ALEKSA, M. *et al.* Atlas liquid argon calorimeter phase-i upgrade technical design report. *CERN Document Server*, 2013.
- ALVES, A. A.; ANDRADE-FILHO, L. M.; COLLABORATION, L. The LHCb detector at the LHC. *Journal of Instrumentation*, IOP Publishing, Bristol, v. 3, n. 08, p. S08005–S08005, Aug 2008.
- ALVES JR., A. A. *et al.* The lhcb detector at the lhc. *JINST*, v. 3, p. S08005, 2008.

- AMBLER, A.; VACHON, B. *Performance and optimization of analog and digital filtering algorithms for the Phase-II upgrade of the Hadronic End-Cap Calorimeter*. Tese (Doutorado) — McGill University, 2018.
- AMIDEI, D. *et al.* A two level fastbus based trigger system for CDF. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Elsevier BV, v. 269, n. 1, p. 51–62, Jun 1988.
- ANDRADE-FILHO, L. M. d. *Detecção e Reconstrução de Raios Cósmicos usando Calorimetria de Altas Energias*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, mar 2009.
- ANDRADE-FILHO, L. M. de *et al.* Calorimeter response deconvolution for energy estimation in high-luminosity conditions. *IEEE Transactions on Nuclear Science, Oak Ridge*, v. 62, n. 6, p. 3265–3273, Dec 2015. ISSN 0018-9499.
- ANDREI, V. *The upgrade of the PreProcessor system of the ATLAS level-1 calorimeter trigger*. Geneva, 2012.
- ANTHONY, K. Celebrating the first of a kind. *CERN Document Server*, set. 2014.
- ARDUINI, G. *et al.* Alternative bunch filling schemes for the LHC. [S.l.], 2007.
- ARGOS, F. C. Activities summary. TileCal Project meeting. 2017.
- ARGOS, F. C.; VALERO, A. *Performance of the Tile PreProcessor Demonstrator for the ATLAS Tile Calorimeter Phase II Upgrade*. Geneva, 2015.
- ARMSTRONG, W. W. *et al.* Atlas: Technical proposal for a general-purpose p p experiment at the large hadron collider at cern. *CERN Document Server*, 1994.
- ASMAN, B. *et al.* The atlas level-1 calorimeter trigger: Preprocessor implementation and performance. *Journal of Instrumentation*, v. 7, p. P12008, 12 2012. Disponível em: <<http://dx.doi.org/10.1088/1748-0221/7/12/P12008>>.
- BALABRAM-FILHO, L. E. *Fator de Qualidade para um Calorímetro de Altas Energias Operando com Alta Taxa de Eventos*. Dissertação (Mestrado) — UFRJ/COPPE, 2014.
- BARBOSA, D. P. *Estudo de Técnicas de Otimização para reconstrução de Energia de jatos no Primeiro Nível de Seleção de Eventos do Experimento ATLAS*. Dissertação (Mestrado) — UFJF, 2012.
- BARBOSA, D. P. *Estimação de Energia para Calorimetria em Física de Altas Energias Baseada em Representação Esparsa*. Tese (Doutorado) — Universidade Federal de Juiz de Fora, 2017.
- Bennett, W. R. Spectra of quantized signals. *The Bell System Technical Journal*, v. 27, n. 3, p. 446–472, July 1948. ISSN 0005-8580.
- BERARDI, V. *et al.* Totem: Technical design report. total cross section, elastic scattering and diffraction dissociation at the large hadron collider at cern. *CERN Document Server*, 2004.
- BERRYMAN, S. Leucippus. In: ZALTA, E. N. (Ed.). *The Stanford Encyclopedia of Philosophy*. Winter 2016. Stanford: Metaphysics Research Lab, Stanford University, 2016.
- BIOT, J. A. V. *Tilecal Read-Out Drivers Production and Firmware Developments*. Tese (Doutorado) — Universitat de València, 2009.

- BÖTTCHER, A.; GRUDSKY, S. M. *Toeplitz Matrices, Asymptotic Linear Algebra and Functional Analysis*. New Delhi: Hindustan Book Agency, 2000.
- BOURGEOIS, D.; FITZPATRICK, C.; STAHL, S. Using holistic event information in the trigger. *arXiv*, 2018.
- BOYLE, R. *The Sceptical Chymist*. London: Kessinger Publishing, LLC, 2007. ISBN 0548010676.
- BRACEWELL, R. *The Fourier Transform & Its Applications*. New York: McGraw-Hill Science/Engineering/Math, 1999. ISBN 978-0073039381.
- BRAIBANT, S.; GIACOMELLI, G.; SPURIO, M. *Particles and Fundamental Interactions*. New York: Springer-Verlag GmbH, 2011. ISBN 9400724632.
- BRENNAN, R. P. *Gigantes da Física*. Rio de Janeiro: Jorge Zahar Editor Ltda., 2012.
- BRICE, M. Lhc restart run 2. *CERN Document Server*, abr. 2015. Disponível em: <<https://cds.cern.ch/record/2006741>>.
- BRIGGS, D. *et al.* A calorimeter software trigger for the mark II detector at SLC. *Computer Physics Communications*, Elsevier BV, v. 57, n. 1-3, p. 273–277, Dec 1989.
- BYSTRICKY, J. *et al.* Algorithms and architecture for the l1 calorimeter trigger at dø run IIb. *IEEE Transactions on Nuclear Science*, Institute of Electrical and Electronics Engineers (IEEE), v. 51, n. 3, p. 351–355, Jun 2004.
- CAI, T. T.; WANG, L. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information Theory*, v. 57, n. 7, p. 4680–4688, 2011.
- CAMPLANI, A. *ATLAS Trigger and Data Acquisition Upgrades for the High Luminosity LHC*. Geneva, 2019.
- CANDES, E.; TAO, T. The dantzig selector: Statistical estimation when p is much larger than n. *Ann. Statist.*, The Institute of Mathematical Statistics, v. 35, n. 6, p. 2313–2351, 12 2007.
- CARLIN, R. *et al.* Experience with the ZEUS trigger system. *Nuclear Physics B - Proceedings Supplements*, Elsevier BV, v. 44, n. 1-3, p. 430–434, Nov 1995.
- CARRIO, F. *et al.* *Design of an FPGA-based embedded system for the ATLAS Tile Calorimeter front-end electronics test-bench*. Geneva, 2013. Disponível em: <https://cds.cern.ch/record/1627588/files/1748-0221_9_03_C03023.pdf>.
- CARVER, L. R. *et al.* The Ghost Train Instability (MDs 2066 and 2936). *CERN Document Server*, Feb 2018.
- CERN. Extended barrels modules 0 installation at CERN. CERN Document Server. 1998. Disponível em: <<http://cds.cern.ch/record/40173>>.
- CERN. *ATLAS technical coordination: Technical Design Report*. Geneva: CERN, 1999. (Technical Design Report ATLAS). Disponível em: <<https://edms.cern.ch/document/103032/1>>.
- CERN. The particle suppliers. les fournisseurs de particules. *CERN Bulletin*, p. 3, mar. 2010.
- CERN. About cern. *CERN Document Server*, jan. 2012.

- CERN. The large hadron collider. *CERN Document Server*, 2014.
- CERN. *Observation of electroweak $W^\pm Z$ boson pair production in association with two jets in pp collisions at $\sqrt{s} = 13\text{TeV}$ with the ATLAS Detector*. Geneva, 2018.
- CERN. Inverse femtobarn. Accessed: 2020-02-20. 2019. Disponível em: <<https://writing-guidelines.web.cern.ch/entries/inverse-femtobarn>>.
- CERQUEIRA, A. *Tile Calorimeter Upgrade Program for the Luminosity Increasing at the LHC*. [S.I.], 2015. Disponível em: <<https://arxiv.org/abs/1509.08994>>.
- CERQUEIRA, A. S. Upgrade for the atlas tile calorimeter readout electronics at the high luminosity lhc. *Journal of Physics: Conference Series*, v. 404, n. 1, p. 012060, 2012.
- CHAPMAN, J. ATLAS simulation computing performance and pile-up simulation in ATLAS. *LPCC*, 2011.
- CHATRCHYAN, S. *et al.* The cms experiment at the cern lhc. *JINST*, v. 3, p. S08004, 2008.
- CHATTERJEE, S.; HADI, A. S. *Regression Analysis by Example*. New Jersey: Wiley–Blackwell, 2013. ISBN 9781118456248.
- CHEN, S.; YANG, M. An adaptive fast iterative shrinkage threshold algorithm. In: *2017 29th Chinese Control And Decision Conference (CCDC)*. [S.I.: s.n.], 2017. p. 2190–2194.
- CHURIWALA, S. *Designing with Xilinx FPGAs: Using Vivado*. 1st. ed. New York: Springer Publishing Company, Incorporated, 2016. ISBN 3319424378, 9783319424378.
- CLELAND, W.; STERN, E. Signal processing considerations for liquid ionization calorimeters in a high rate environment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Elsevier BV, v. 338, n. 2-3, p. 467–497, Jan 1994.
- CLEMENT, C.; KLIMEK, P. *Identification of Pile-up Using the Quality Factor of Pulse Shapes in the ATLAS Tile Calorimeter*. Geneva, 2011. Disponível em: <<http://cds.cern.ch/record/1398262>>.
- CLÉMENT, C. *et al.* *Time Calibration of the ATLAS Hadronic Tile Calorimeter using the Laser System*. Geneva, 2008.
- COLLABORATION, A. *Technical Design Report for the Phase-II Upgrade of the ATLAS Tile Calorimeter*. Geneva, 2017.
- COLLABORATION, C. *The Phase-2 Upgrade of the CMS L1 Trigger Interim Technical Design Report*. Geneva, 2017.
- COLLABORATION, L. LHCb trigger system technical design report. *CERN Document Server*, 2003.
- COLLABORATION, L. Lhc machine outreach. On-line available. 2018.
- COLLABORATION, T. A. Operation of the ATLAS trigger system in run 2. *Journal of Instrumentation*, IOP Publishing, v. 15, n. 10, p. P10004–P10004, oct 2020. Disponível em: <<https://doi.org/10.1088/1748-0221/15/10/p10004>>.
- CZODROWSKI, P. *The ATLAS Trigger System: Ready for Run 2*. Geneva, 2015.

- DAMERAU, H. *et al.* RF Manipulations for Special LHC-Type Beams in the CERN PS. *CERN Document Server*, p. WEPAF063. 4 p, 2018.
- DANIELLS, A. C. *Pile-Up Suppression in the ATLAS Level 1 Calorimeter Trigger and Searches for Higgs Boson Pair Production*. Tese (Doutorado) — University of Birmingham, 2016. Disponível em: <http://www.ep.ph.bham.ac.uk/publications/thesis/acd_thesis.pdf>.
- DAUBECHIES, I.; DEFRISE, M.; MOL, C. D. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, Wiley Online Library, v. 57, n. 11, p. 1413–1457, 2004.
- DONOHO, D. L. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, v. 41, n. 3, p. 613–627, May 1995. ISSN 0018-9448.
- DONOHO, D. L.; ELAD, M. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, National Academy of Sciences, v. 100, n. 5, p. 2197–2202, 2003. ISSN 0027-8424.
- DUARTE, J. *et al.* Online energy reconstruction for calorimeters under high pile-up conditions using deconvolutional techniques. *Journal of Instrumentation*, IOP Publishing, v. 14, n. 12, p. P12017–P12017, dec 2019.
- DUARTE, J. a. P. B. d. S. *Estudo de Técnicas de Deconvolução para Reconstrução de Energia online no Calorímetro Hadrônico do ATLAS*. Dissertação (Mestrado) — UFJF, 2015.
- DUARTE, J. P. B. S.; Andrade Filho, L. M.; SEIXAS, J. M. Representação esparsa aplicada à estimativa online de energia para um calorímetro submetido a uma alta taxa de eventos. In: SBRT. *XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*. São Pedro - SP, 2017.
- DUARTE, J. P. B. S. *et al.* Técnicas de deconvolução aplicadas à estimativa de energia online em calorimetria de altas energias em condições de alta taxa de eventos. In: *XXI Congresso Brasileiro de Automática*. Vitória - ES: [s.n.], 2016.
- DUMITRESCU, B. Sparse total least squares: Analysis and greedy algorithms. *Linear Algebra and its Applications*, Elsevier BV, v. 438, n. 6, p. 2661–2674, Mar 2013.
- ELAD, M. *Sparse and Redundant Representations*. New York: Springer New York, 2010.
- ELAD, M.; MATALON, B.; ZIBULEVSKY, M. Image denoising with shrinkage and redundant representations. In: *IEEE Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. [S.I.], 2006. v. 2, p. 1924–1931.
- EVANS, L.; BRYANT, P. Lhc machine. *JINST*, v. 3, p. S08001, 2008.
- FAWCETT, T. An introduction to ROC analysis. *Pattern Recognition Letters*, Elsevier BV, v. 27, n. 8, p. 861–874, jun 2006.
- FERMILAB, D. C. *DØ Run IIb Upgrade Technical Design Report*. Batavia, 2002.
- FIELDS, S.; CARVER, L. Analyzing the lhc magnet quenches. *IEEE Spectrum: Technology, Engineering, and Science News*, nov. 2017. Disponível em: <<https://spectrum.ieee.org/analyzing-the-lhc-magnet-quenches>>.

- FISHER, S. Pierre gassendi. In: ZALTA, E. N. (Ed.). *The Stanford Encyclopedia of Philosophy*. Spring 2014. Stanford: Metaphysics Research Lab, Stanford University, 2014.
- FOX, J. *et al.* Control of Intra-Bunch Vertical Motion in the SPS with GHz Bandwidth Feedback. *J. Phys. : Conf. Ser.*, v. 1067, n. 7, p. WEPAL079. 6 p, 2018.
- FULLANA, E. *et al.* *Optimal Filtering in the ATLAS Hadronic Tile Calorimeter*. Geneva, 2005.
- GARVEY, J.; REES, D. *Bunch Crossing Identification for the ATLAS Level-1 Calorimeter Trigger*. [S.I.], 1996.
- GILL, P. R.; WANG, A.; MOLNAR, A. The in-crowd algorithm for fast basis pursuit denoising. *IEEE Transactions on Signal Processing*, IEEE, v. 59, n. 10, p. 4595–4605, 2011.
- GIOVANNOZZI, M.; VEKEN, F. F. Van der. Description of the luminosity evolution for the CERN LHC including dynamic aperture effects. Part I: the model. *Nucl. Instrum. Meth.*, A905, p. 171–179, 2018.
- GIOVANNOZZI, M.; VEKEN, F. F. Van der. Description of the luminosity evolution for the CERN LHC including dynamic aperture effects. Part II: application to Run 1 data. *Nucl. Instrum. Meth.*, A908, p. 1–9, 2018.
- GOODSON, J. J. *Search for Supersymmetry in States with Large Missing Transverse Momentum and Three Leptons including a Z-Boson*. Tese (Doutorado) — Stony Brook University, May 2012. Presented 17 Apr 2012. Disponível em: <<http://cds.cern.ch/record/1449722>>.
- GREEN, B. *et al.* *ATLAS Trigger/DAQ RobIn Prototype*. Geneva, 2003.
- GRIFFEL, F. Al-ghazali. In: ZALTA, E. N. (Ed.). *The Stanford Encyclopedia of Philosophy*. Winter 2016. Stanford: Metaphysics Research Lab, Stanford University, 2016.
- HARKUSHA, S.; COLLABORATION, A. T. *Upgrade of the ATLAS hadronic Tile calorimeter for the High luminosity LHC*. Geneva, 2016.
- HAYKIN, S. *Adaptative Filter Theory*. 5th. NJ, USA: Prentice Hall, 2013. ISBN 978-0132671453.
- HERR, W. *Effects of PACMAN bunches in the LHC*. Geneva, 1996.
- HILUMI, P. HL-LHC Industry: Industry Relations and Procurement Website for the HL-LHC project. Accessed: 2020-09-08. 2020. Disponível em: <<https://hilumilhc.web.cern.ch/content/hl-lhc-project>>.
- HORN, P. AREUS - a software framework for the ATLAS Readout Electronics Upgrade Simulation. *CERN Document Server*, Apr 2018.
- HOSTETTLER, M. *et al.* Luminosity Scans for Beam Diagnostics. *Phys. Rev. Accel. Beams*, v. 21, n. arXiv:1804.10099, p. 102801. 10 p, Apr 2018.
- HUANG, K.; AVIYENTE, S. Sparse representation for signal classification. In: *Advances in neural information processing systems*. [S.I.: s.n.], 2007. p. 609–616.
- JOWETT, J. *et al.* Lifetime of Asymmetric Colliding Beams in the LHC. *CERN Document Server*, p. TUPVA013. 4 p, 2017.

- KAHLE, K.; STEERENBERG, R.; PANDOLFI, S. Lhc report: Something in the nothing. *CERN Document Server*, 2017.
- KAY, S. M. *Fundamentals of Statistical Signal Processing, Volume I*. NJ, USA: Prentice Hall, 1993. ISBN 0133457117.
- KLIMEK, P. Signal reconstruction performance with the ATLAS hadronic tile calorimeter. *Journal of Physics: Conference Series*, IOP Publishing, v. 404, p. 012046, dec 2012.
- KLOUS, S. *Event streaming in the online system: Real-Time Organization of ATLAS Data*. Geneva, 2010.
- KOSTOGLOU, S. et al. *Impact of the 50 Hz harmonics on the beam evolution of the Large Hadron Collider*. [S.I.], 2020.
- KRIVDA, M. et al. The ALICE trigger electronics. *CERN Document Server*, p. 5 p, 2007.
- LIJIAO, L. *L0/L1 trigger generation by the ALICE PHOS detector*. Tese (Doutorado) — Bergen U., 2011.
- LIMA, L. C. *História da Física*. 2007.
- LUENBERGER, D. G.; YE, Y. *Linear and Nonlinear Programming*. New York: Springer US, 2008.
- MACINTOSH, J. J.; ANSTEY, P. Robert boyle. In: ZALTA, E. N. (Ed.). *The Stanford Encyclopedia of Philosophy*. Fall 2014. Stanford: Metaphysics Research Lab, Stanford University, 2014.
- MADYSA, N. *AREUS - a software framework for ATLAS Readout Electronics Upgrade Simulation*. Geneva, 2018.
- MCKINNEY, W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 2 edition. ed. MA, Boston: O'Reilly Media, 2017. ISBN 978-1491957660.
- MEYER-BAESE, U. *Digital Signal Processing with Field Programmable Gate Arrays*. 3. ed. New York: Springer Berlin Heidelberg, 2007. ISBN 978-3-540-72613-5.
- MILLER, S. J. The method of least squares. *Mathematics Department Brown University*, v. 8, p. 1–7, 2006.
- MILNE, W. E. *Numerical Calculus (Princeton Legacy Library)*. New Jersey: Princeton University Press, 2016. ISBN 0691653488.
- MITRA, S. K. *Digital Signal Processing : A Computer-Based Approach*. New York: McGraw-Hill Higher Education, 2010. ISBN 978-0073380490.
- MLYNARIKOVA, M. *Upgrade of the ATLAS hadronic Tile calorimeter for the High luminosity LHC*. [S.I.], 2017. Disponível em: <<https://arxiv.org/abs/1709.00099>>.
- MOBS, E. The CERN accelerator complex. Complexe des accélérateurs du CERN. jul. 2016. General Photo. Disponível em: <<http://cds.cern.ch/record/2197559>>.
- MORÉ, J. J.; THUENTE, D. J. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software (TOMS)*, ACM New York, NY, USA, v. 20, n. 3, p. 286–307, 1994.

- MOREIRA, M. A. O modelo padrão da física de partículas. *Revista Brasileira de Ensino de Física*, 2009.
- MOUCHE, P. Overall view of the LHC. Vue d'ensemble du LHC. *CERN Document Server*, Jun 2014. General Photo. Disponível em: <<https://cds.cern.ch/record/1708847>>.
- MURK, S. *Trigger Algorithms and Monte Carlo Event Generation for Dijet Searches in ATLAS and Beyond*. Dissertação (Mestrado) — Lund University, 2016. Disponível em: <<http://lup.lub.lu.se/student-papers/record/8902969/file/8902970.pdf>>.
- NAKAHAMA, Y. *The ATLAS Trigger System: Ready for Run-2*. Geneva, 2015.
- PALKA, M. Operation of the enhanced ATLAS first level calorimeter trigger at the start of run-2. *Journal of Instrumentation*, IOP Publishing, v. 11, n. 02, p. C02059–C02059, feb 2016. Disponível em: <<http://dx.doi.org/10.1088/1748-0221/11/02/C02059>>.
- PALNITKAR, S. *Verilog HDL*. 2 edition. ed. NJ, USA: Prentice Hall, 2003. ISBN 978-0130449115.
- PANDOLFI, S. Lhc report: end of 2016 proton-proton operation. *CERN Document Server*, out. 2016.
- PARAB, J. S.; GAD, R. S.; NAIK, G. M. Getting hands on altera® quartus® ii software. In: _____. *Hands-on Experience with Altera FPGA Development Boards*. New Delhi: Springer India, 2018. p. 19–37. ISBN 978-81-322-3769-3.
- PATI, Y. C.; REZAIIFAR, R.; KRISHNAPRASAD, P. S. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*. [S.l.: s.n.], 1993. p. 40–44 vol.1. ISSN 1058-6393.
- PEQUENAO, J. Computer generated image of the atlas calorimeter. *CERN Document Server*. 2008. Disponível em: <<https://cds.cern.ch/images/CERN-GE-0803015-01>>.
- PEQUENAO, J. Computer generated image of the atlas inner detector. *CERN Document Server*. 2008. Disponível em: <<https://cds.cern.ch/images/CERN-GE-0803014-01>>.
- PEQUENAO, J. Computer generated image of the atlas liquid argon. *CERN Document Server*. 2008. Disponível em: <<https://cds.cern.ch/images/CERN-GE-0803016-01>>.
- PEQUENAO, J. Computer generated image of the atlas muons subsystem. *CERN Document Server*. 2008. Disponível em: <<https://cds.cern.ch/images/CERN-GE-0803017-01>>.
- PEQUENAO, J. Computer generated image of the whole atlas detector. *CERN Document Server*. 2008. Disponível em: <<https://cds.cern.ch/record/1095924>>.
- PERALVA, B. S.-M. *Detecção de sinais e estimativa de energia para calorimetria de altas energias*. Dissertação (Mestrado) — UFJF, 2012.
- PERALVA, B. S. M. et al. *The TileCal Energy Reconstruction for Collision Data Using the Matched Filter*. Geneva, 2013.
- PEREIRA, R. A. *Estimação de Energia em um Calorímetro Finamente Segmentado*. Dissertação (Mestrado) — UFRJ/COPPE, 2014.

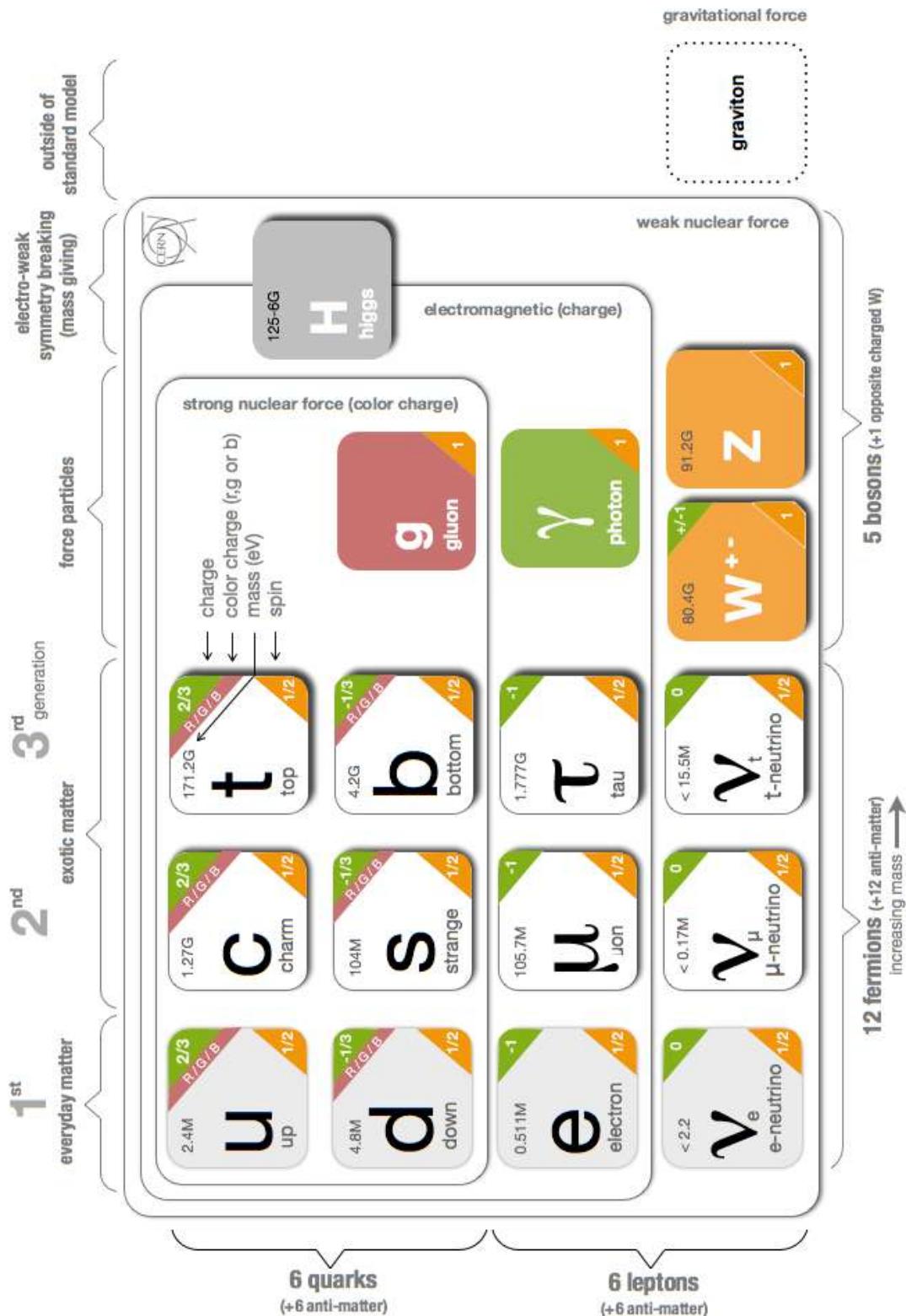
- PERKINS, D. Introduction to high energy physics. *Cambridge University Press*, 2000.
- PETTERSSON, T. S.; LEFÈVRE, P. The Large Hadron Collider: Conceptual design. *CERN Document Server*, 1995.
- PRALAVORIO, C. The new annual report is available. *CERN Document Server*, jun. 2017.
- PRALAVORIO, C. Record luminosity: well done lhc. *CERN Document Server*, nov. 2017.
- PURCELL, A. Go on a particle quest at the first cern webfest. le premier webfest du cern se lance à la conquête des particules. *CERN Publications*, p. 10, ago. 2012. Printed version. Disponível em: <<http://cds.cern.ch/record/1474832/files/2012-35-36-E-web.pdf>>.
- RIFFE, D. M. *The Dirac Delta Function*. 2018. Accessed: 2020-02-20. Disponível em: <http://www.physics.usu.edu/riffe/3750/lecture_notes.htm>.
- RONCHETTI, F. et al. The ALICE electromagnetic calorimeter high level triggers. *J. Phys. Conf. Ser.*, v. 396, p. 012045, 2012.
- ROONEY, A. *A História da Física*. São Paulo: M. Books do Brasil Editora Ltda, 2013.
- ROSE, A. *The Level-1 Trigger of the CMS experiment at the LHC and the Super-LHC*. Tese (Doutorado) — Imperial Coll., London, 2009.
- RÜDEN, W. von. Trigger and data-acquisition plans for the LEP experiments. *CERN Document Server*, 1985.
- SANTOS, J. T. Presença da identidade eleática na filosofia grega clássica. *Journal of Ancient Philosophy*, III, 2009. Issue 2.
- SARDY, S.; BRUCE, A. G.; TSENG, P. Block coordinate relaxation methods for nonparametric wavelet denoising. *Journal of Computational and Graphical Statistics*, JSTOR, v. 9, n. 2, p. 361, Jun 2000.
- SEBER, G. A. F. *Linear regression analysis*. Hoboken, N.J: Wiley-Interscience, 2003. ISBN 978-0471415404.
- SHILTSEV, V. Overview of beam-beam effects in the Tevatron. In: *Proceedings, ICFA Mini-Workshop on Beam-Beam Effects in Hadron Colliders (BB2013)*: CERN, Geneva, Switzerland, March 18-22 2013. [S.l.: s.n.], 2014. p. 11–18.
- SHILTSEV, V. et al. Beam-beam effects in the tevatron. *Phys. Rev. ST Accel. Beams*, American Physical Society, v. 8, p. 101001, Oct 2005.
- SHILTSEV, V. et al. Experimental Studies of Compensation of Beam-Beam Effects with Tevatron Electron Lenses. *New J. Phys.*, v. 10, p. 043042, 2008.
- SILVA, L. R. M. *Estudo e Implementação de Técnicas de Detecção e Compressão de Distúrbios Elétricos*. Tese (Doutorado) — Universidade Federal de Juiz de Fora, 2016.
- SOLODKOV, A. *Upgrade of the ATLAS hadronic Tile Calorimeter for the High luminosity LHC*. Geneva, 2017. Disponível em: <<http://cds.cern.ch/record/2264924>>.
- STARCK, J.-L.; MURTAGH, F.; FADILI, J. M. *Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity*. New York: Cambridge University Press, 2010. ISBN 978-0521119139.

- STARCK, J.-L.; MURTAGII, F.; BIJAOUI, A. Multiresolution support applied to image filtering and restoration. *Graphical Models and Image Processing*, Elsevier BV, v. 57, n. 5, p. 420–431, Sep 1995.
- STEERENBERG, R. *LHC Report: The LHC is full.* [S.I.], 2018.
- SUMMERS, S. *et al.* Fast inference of Boosted Decision Trees in FPGAs for particle physics. *JINST*, v. 15, n. arXiv:2002.02534. 05, p. P05026. 14 p, Feb 2020.
- SUTTON, C. *Particle accelerator.* 2013. Accessed: 2020-02-20. Disponível em: <<https://www.britannica.com/technology/particle-accelerator>>.
- TANG, F. *et al.* Upgrade Analog Readout and Digitizing System for ATLAS TileCal Demonstrator. *IEEE Trans. Nucl. Sci.*, v. 62, n. 3, p. 1045–1049, 2015.
- TEIXEIRA, M. V.; ANDRADE-FILHO, L. M.; PERALVA, B. S. Reconstrução online para calorímetros operando em condições de altas luminosidades. In: *XX Congresso Brasileiro de Automática*. Belo Horizonte - MG: [s.n.], 2014.
- TEIXEIRA, T.; ANDRADE, L.; SEIXAS, J. de. Sparse deconvolution methods for online energy estimation in calorimeters operating in high luminosity conditions. *Journal of Instrumentation*, IOP Publishing, v. 16, n. 09, p. P09008, sep 2021. Disponível em: <<https://doi.org/10.1088/1748-0221/16/09/p09008>>.
- TEIXEIRA, T. A. *et al.* Método iterativo de representação esparsa implementado em FPGA para aplicação em calorimetria. In: *XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBRT 2020)*. Florianópolis, Brazil: [s.n.], 2020.
- TEIXEIRA, T. A. *et al.* Implementação em fpga de um método recursivo de deconvolução aplicado em calorímetros operando a alta taxa de eventos. In: SBRT. *XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*. Campina Grande - PB, 2018.
- TEIXEIRA, T. A.; FILHO, L. M. A.; SEIXAS, J. M. Implementação de métodos iterativos de deconvolução para processamento on-line no calorímetro hadrônico do atlas. In: SBF. *XLII RTFNB - XL ENFPC 2019*. Campos do Jordão, 2019.
- TEIXEIRA, T. A.; FILHO, L. M. A.; SEIXAS, J. M. Métodos de representação esparsa implementados de forma iterativa para a realização de deconvolução de sinais obtidos com o calorímetro hadrônico do atlas. In: SBF. *XLIII RTFNB - XLI ENFPC 2021*. OnLine, 2021.
- TEIXEIRA, T. A.; FILHO, L. M. de A. Linear equation system solver circuit for streaming processing in fpga. *Journal of Control, Automation and Electrical Systems*, oct 2021. ISSN 2195-3899. Disponível em: <<https://doi.org/10.1007/s40313-021-00842-6>>.
- THE Higgs boson. 2014. Accessed: 2020-02-20. Disponível em: <<https://cds.cern.ch/record/1998491>>.
- THIJSSEN, H. Nicholas of autrecourt. In: ZALTA, E. N. (Ed.). *The Stanford Encyclopedia of Philosophy*. Spring 2016. Stanford: Metaphysics Research Lab, Stanford University, 2016.
- THORPE, J. D. T. S. T. E. *A new system of chemical philosophy*. London: Printed by S. Russell for R. Bickerstaff, London, 1808.

- ULTRASCALE. *UltraScale and UltraScale+ FPGAs Packaging and Pinouts*. [S.l.], 2019. Accessed: 2020-03-25. Disponível em: <https://www.xilinx.com/support/documentation/user_guides/ug575-ultrascale-pkg-pinout.pdf>.
- VALERO, A.; COLLABORATION, A. T. *Data acquisition and processing in the ATLAS tile calorimeter phase-II upgrade demonstrator*. Geneva, 2017.
- VAZQUEZ, W. P. *Functional diagram of the ATLAS Trigger and Data Acquisition system in Run 2 showing expected peak rates and bandwidths through each component*. [S.l.], 2017. Disponível em: <<https://cds.cern.ch/record/2244345/files/ATL-DAQ-PROC-2017-007.pdf>>.
- VIEIRA, C. L. *História da Física*. Rio de Janeiro: CBPF, 2015.
- WAGNER, K. *Proportionate-type normalized least mean square algorithms*. London, U.K. Hoboken, N.J: ISTE Wiley, 2013. ISBN 978-1848214705.
- WENNINGER, J.; PANDOLFI, S. Lhc report: operation with holes. *CERN Document Server*, 2017.
- WU, H.; WANG, S. Adaptive sparsity matching pursuit algorithm for sparse reconstruction. *IEEE Signal Processing Letters*, v. 19, n. 8, p. 471–474, 2012.
- XILINX. *Xilinx Web Site*. [S.l.], 2019. Accessed: 2020-03-25. Disponível em: <<https://www.xilinx.com/>>.
- XU, C.; AISSAOUI, I.; JACQUEY, S. Algebraic analysis of the van cittert iterative method of deconvolution with a general relaxation factor. *Journal of the Optical Society of America A*, The Optical Society, v. 11, n. 11, p. 2804, Nov 1994.

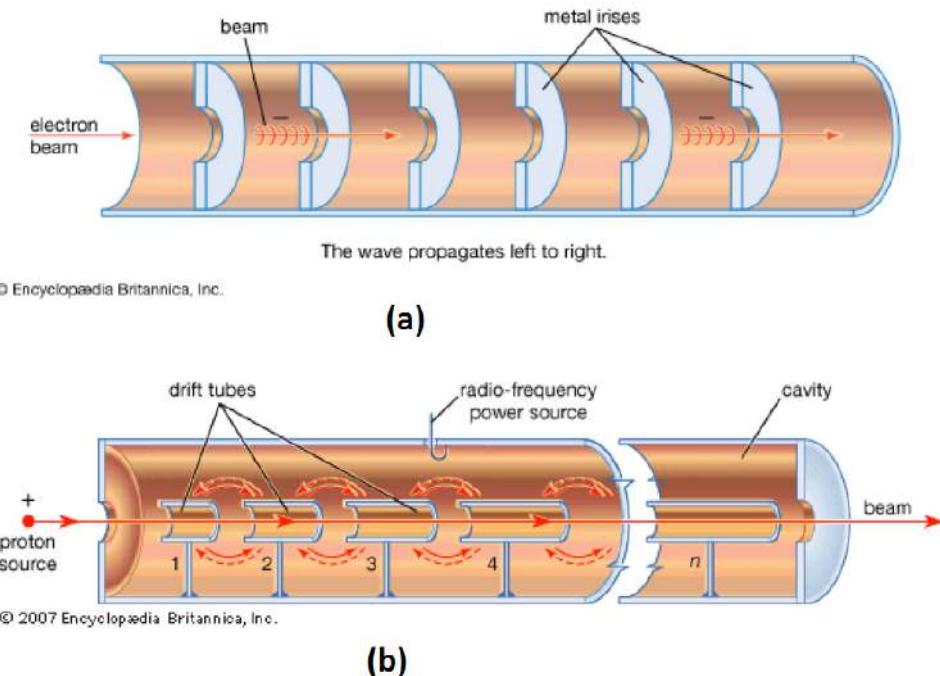
APÊNDICE A – Ilustrações da seção 2

Figura 1 - Infográfico contendo as partículas fundamentais e forças previstas no Modelo Padrão



Fonte: Purcell (2012, p. 9).

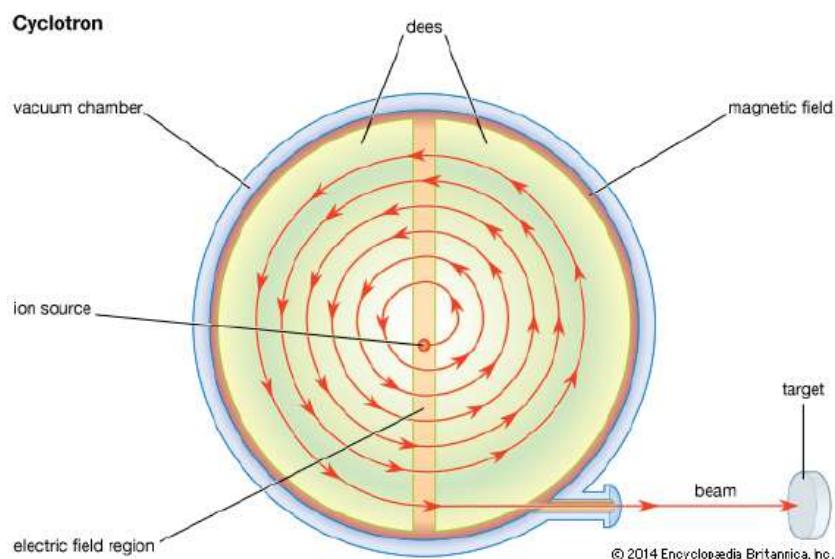
Figura 2 - Técnicas para construção de aceleradores lineares



Nota(s): (a) Câmara de aceleração de um acelerador de elétrons linear (b) Diagrama esquemático de um acelerador linear de ressonância de prótons.

Fonte: Sutton (2013).²³

Figura 3 - Vista plana de um cyclotron clássico.

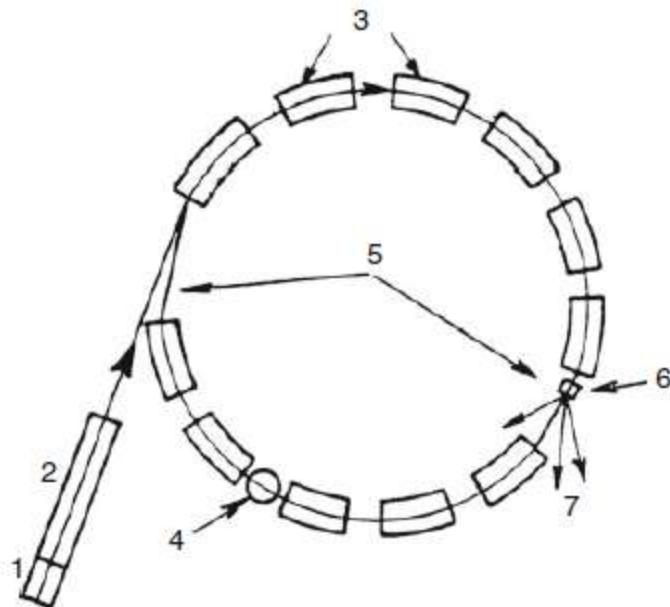


Fonte: Sutton (2013)²⁴.

²³Disponível em: <https://www.britannica.com/technology/particle-accelerator>

²⁴Disponível em: <https://www.britannica.com/technology/particle-accelerator>

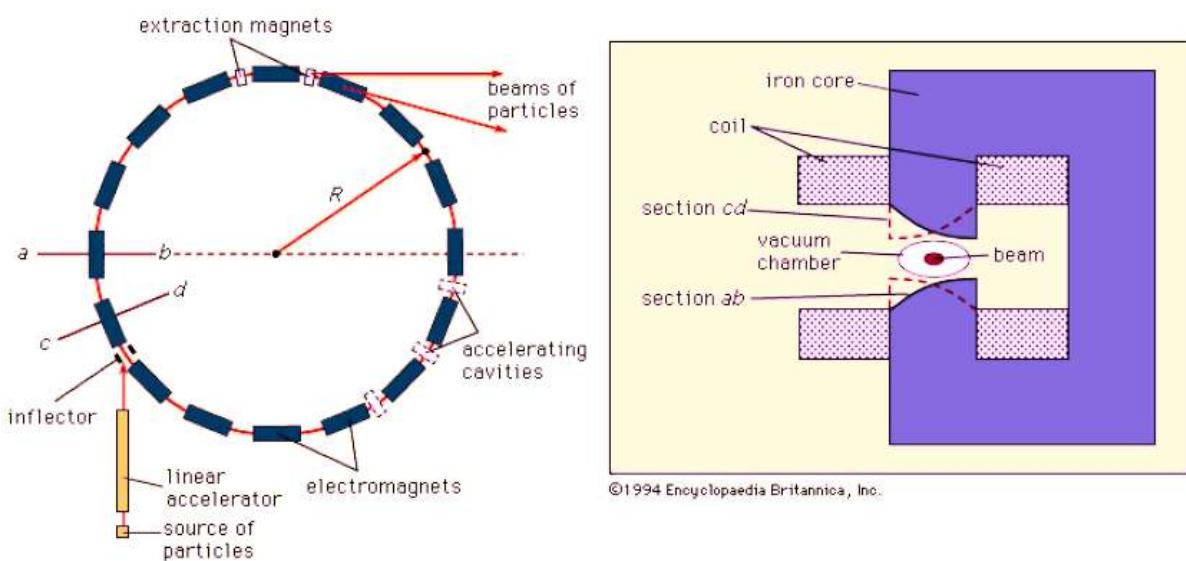
Figura 4 - Esquema de um protonsynchrotron



Nota(s): O pré-acelerador é um acelerador eletroestático (1) seguido por um acelerador linear (2). No anel principal encontra-se: (3) Os imãs, (4) uma cavidade de aceleração e (5) as "seções retas" nas quais (6) os alvos são instalados, dos quais (7) as linhas de feixes secundárias são produzidas para os experimentos.

Fonte: Braibant, Giacomelli e Spurio (2011, p. 51).

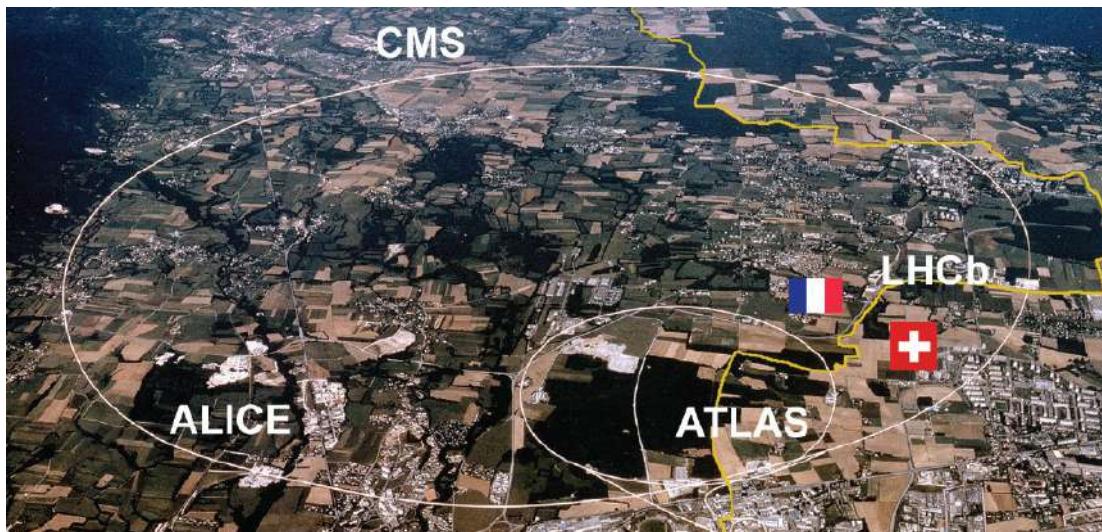
Figura 5 - Diagrama esquemático de um síncrotron com foco em gradiente alternado



Fonte: Sutton (2013)²⁵.

²⁵Disponível em: <https://www.britannica.com/technology/particle-accelerator>

Figura 6 - Localização do Conseil Européen pour la Recherche Nucléaire na fronteira franco-suíça



Fonte: Fields e Carver (2017)²⁶.

Fotografia 1 - Large Hadron Collider no Conseil Européen pour la Recherche Nucléaire



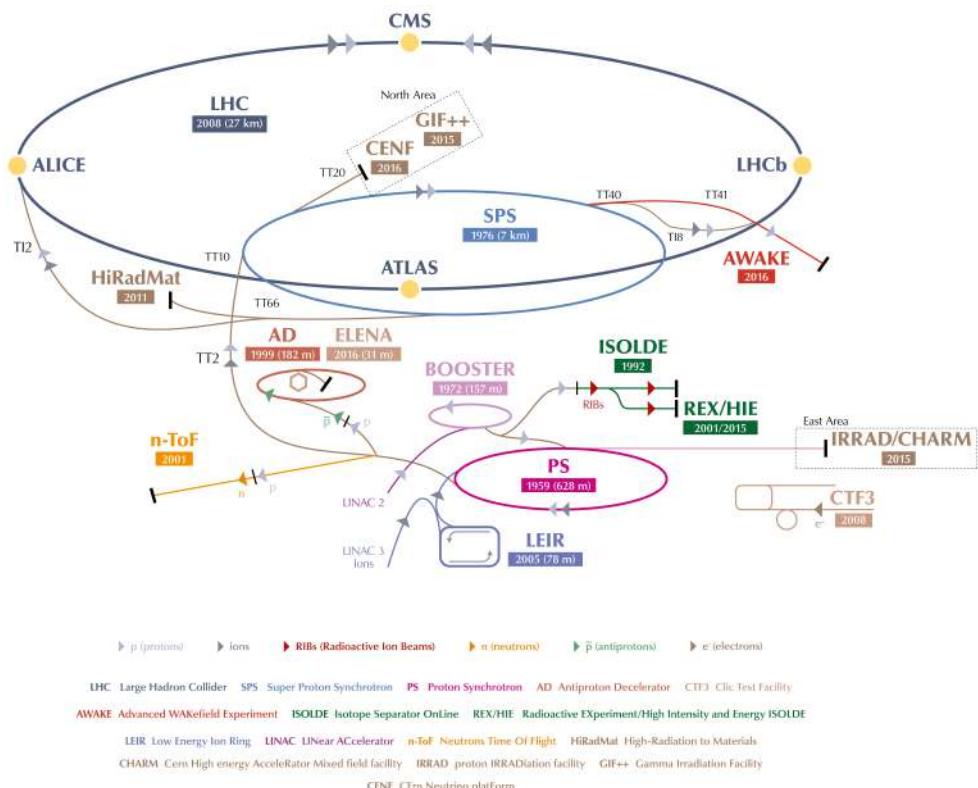
Nota(s): O hub LHC no Centro de Controle do CERN foi embalado com operadores do LHC para o reinício do acelerador após um desligamento de dois anos.

Fonte: Brice (2015)²⁷.

²⁶Disponível em: <https://spectrum.ieee.org/analyzing-the-lhc-magnet-quenches>

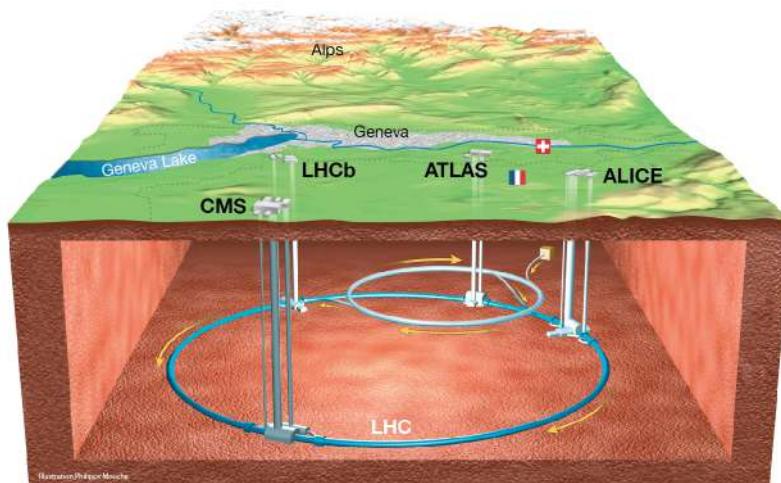
²⁷Disponível em: <https://cds.cern.ch/record/2006741>

Figura 7 - O complexo do LHC com todos os seus detectores e sub-dectores



© CERN, 2016. All rights reserved.

Figura 8 - Plano de corte do Large Hadron Collider

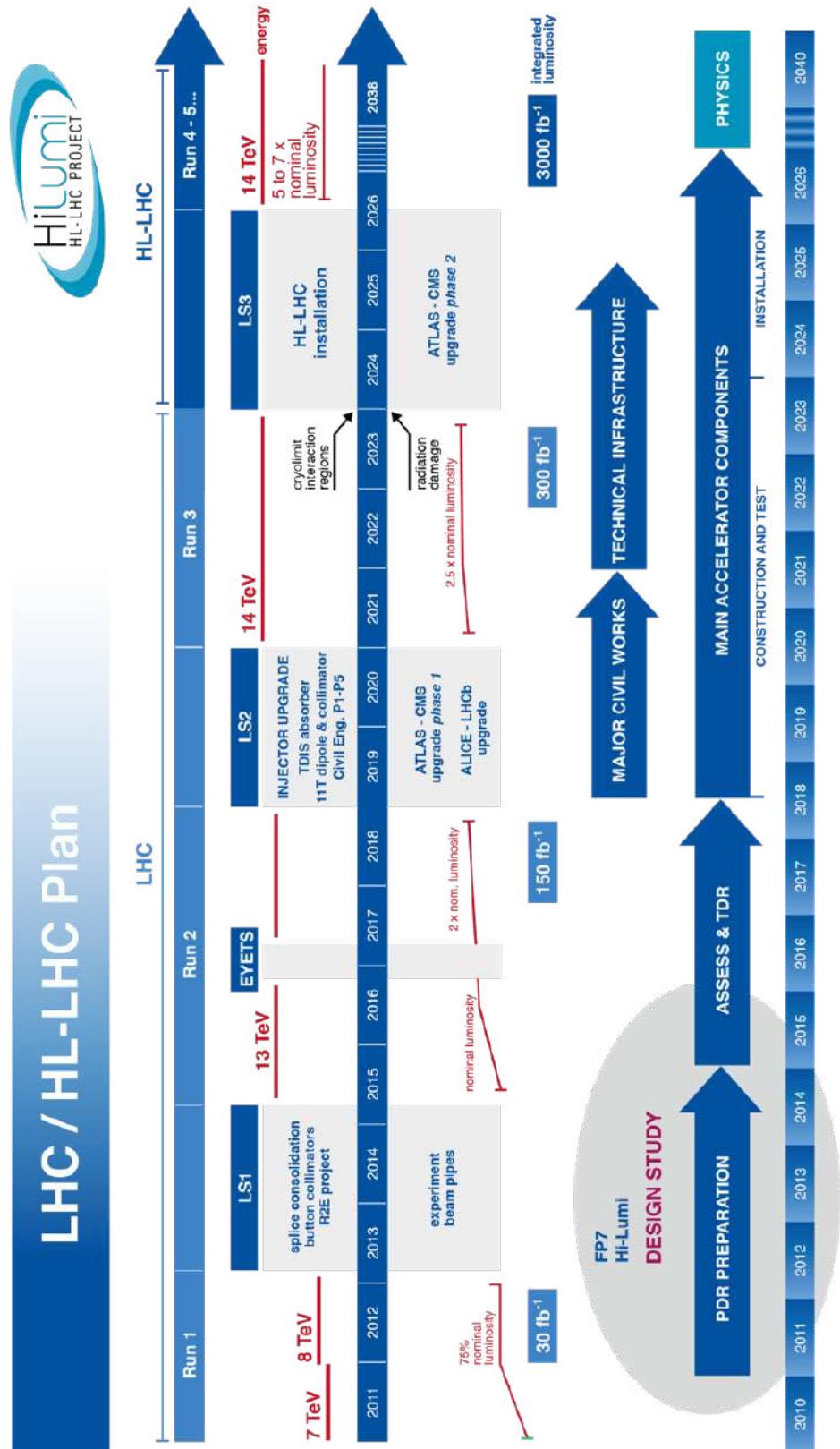


Fonte: Mouche (2014)²⁹.

²⁸Disponível em: <https://cds.cern.ch/record/2197559>

²⁹Disponível em: <https://cds.cern.ch/record/1708847>

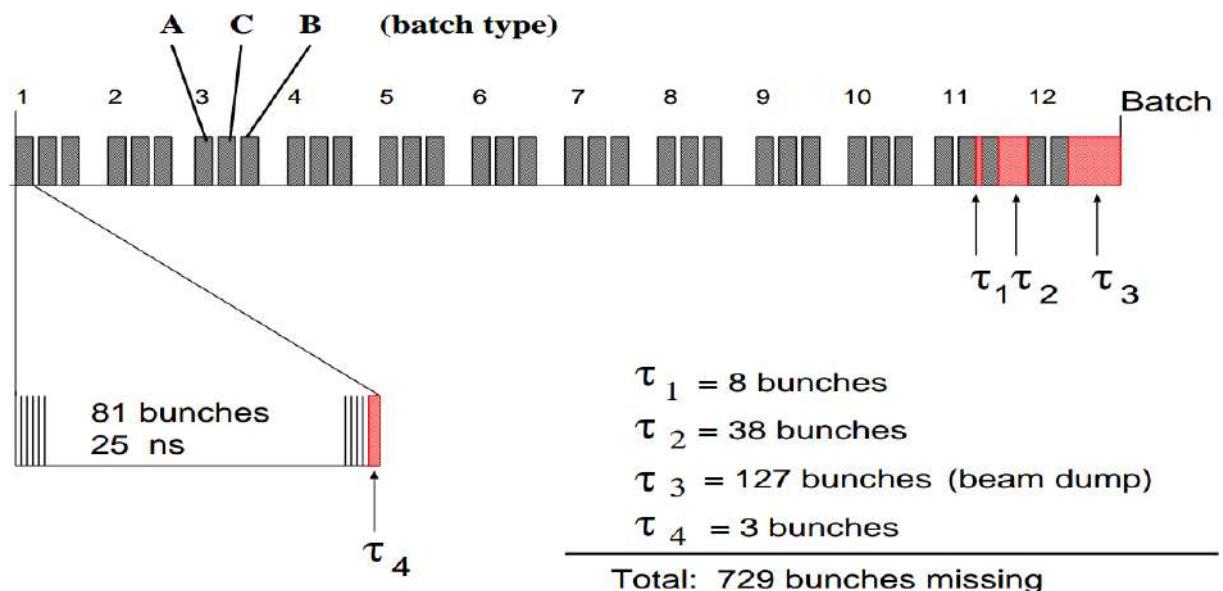
Figura 9 - Cronologia das atualizações do Large Hadron Collider



Fonte: Hilumi (2020)³⁰.

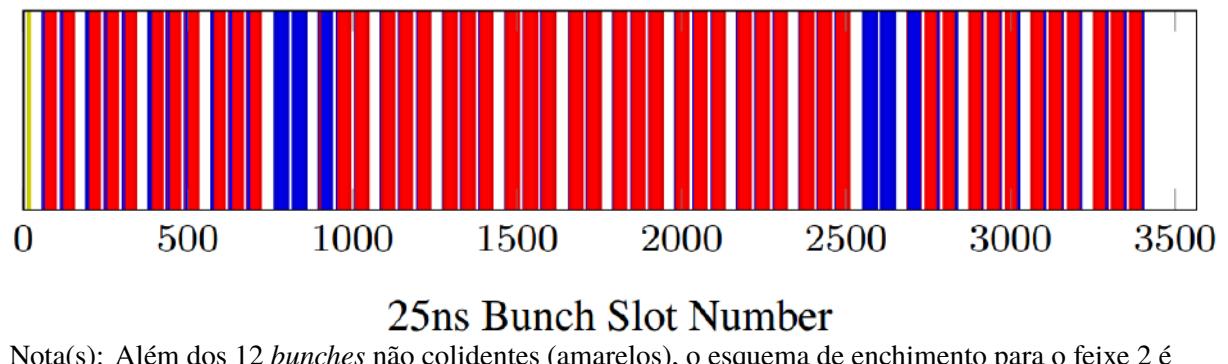
³⁰Disponível em: <https://hilumilhc.web.cern.ch/content/hl-lhc-project>

Figura 10 - Trem de bunch do *Large Hadron Collider* com intervalos para o tempo de subida do kicker e intervalo de abortar



Fonte: Herr (1996, p. 4).

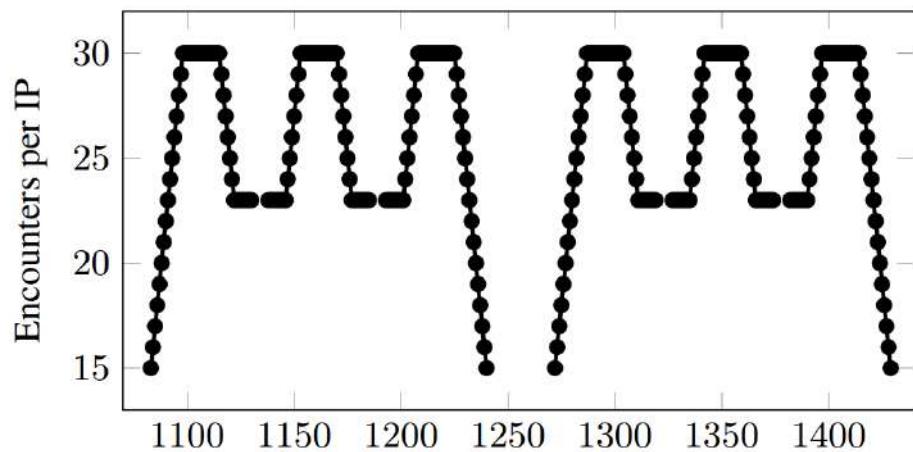
Figura 11 - Esquema de enchimento de 2556 *bunches* usado para o feixe 1 do LHC em 2017 para a operação física de prótons



Nota(s): Além dos 12 *bunches* não colidentes (amarelos), o esquema de enchimento para o feixe 2 é idêntico. Os *bunches* em azul não colidem nos experimentos ALICE ou LHCb.

Fonte: Hostettler *et al.* (2018, p. 7).

Figura 12 - Encontro de feixes de longo alcance na operação do *Large Hadron Collider*, 2017

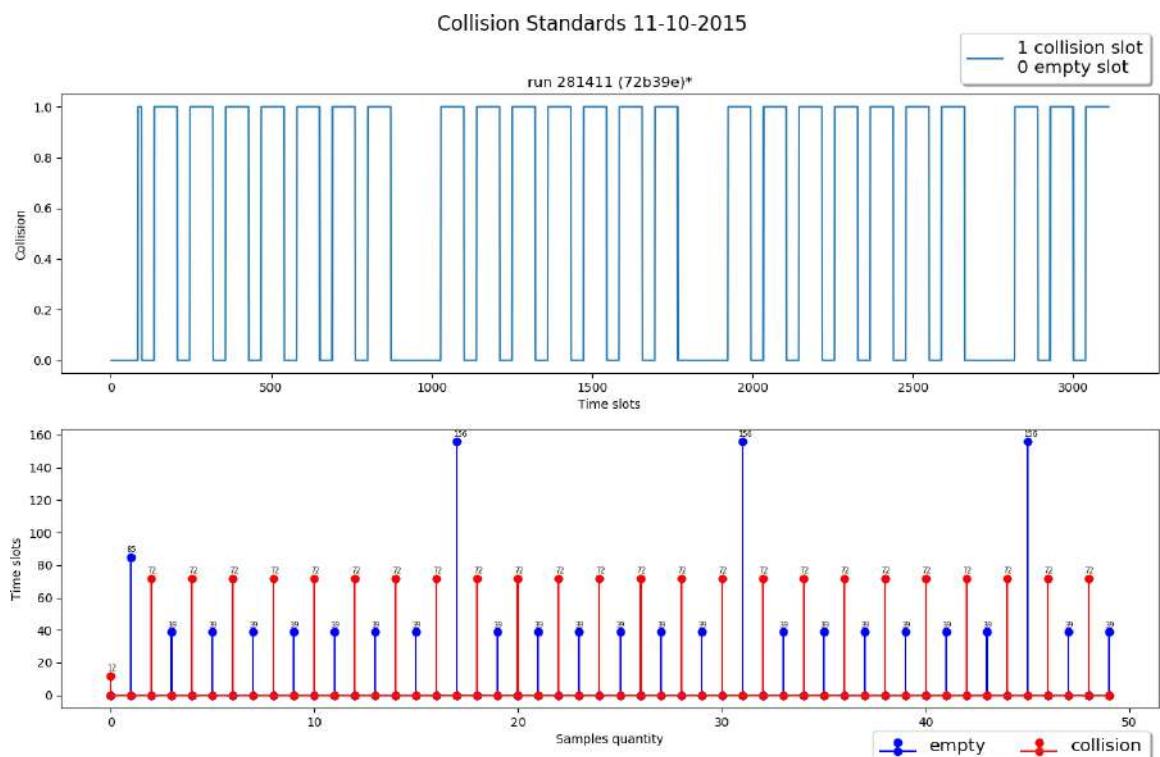


25ns Bunch Slot Number

Nota(s): Número de encontros de feixes de longo alcance no espaço de deriva e triplos internos em torno do ATLAS e CMS para dois trens de 2556 *bunches* do esquema de preenchimento dos feixes usados na operação do LHC em 2017. O padrão é idêntico para todos os trens.

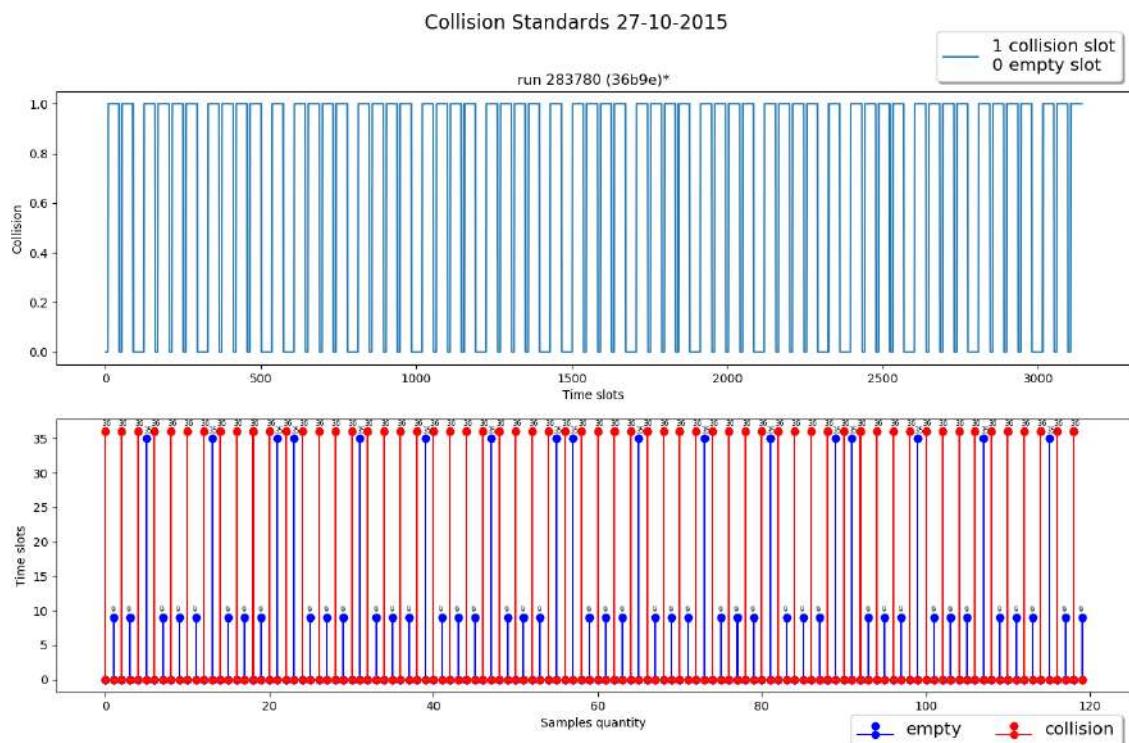
Fonte: Hostettler *et al.* (2018, p. 7).

Gráfico 1 - Padrão de colisão run 281411



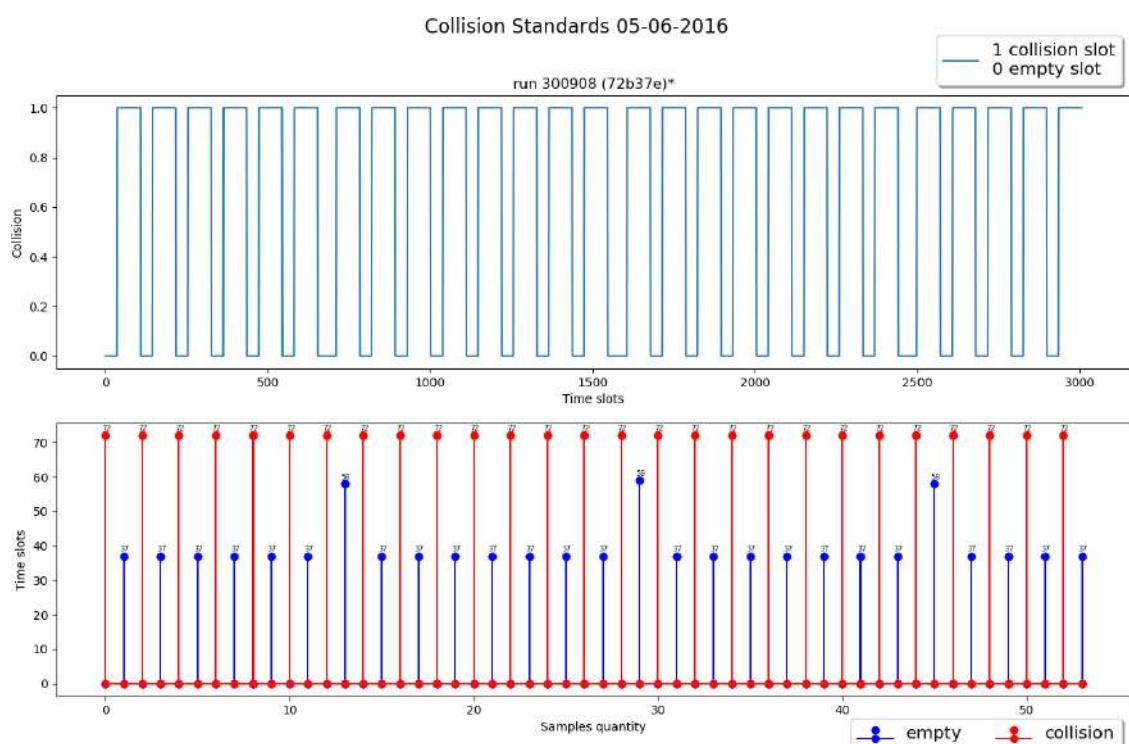
Fonte: Elaborado pelo autor (2021).

Gráfico 2 - Padrão de colisão run 283780



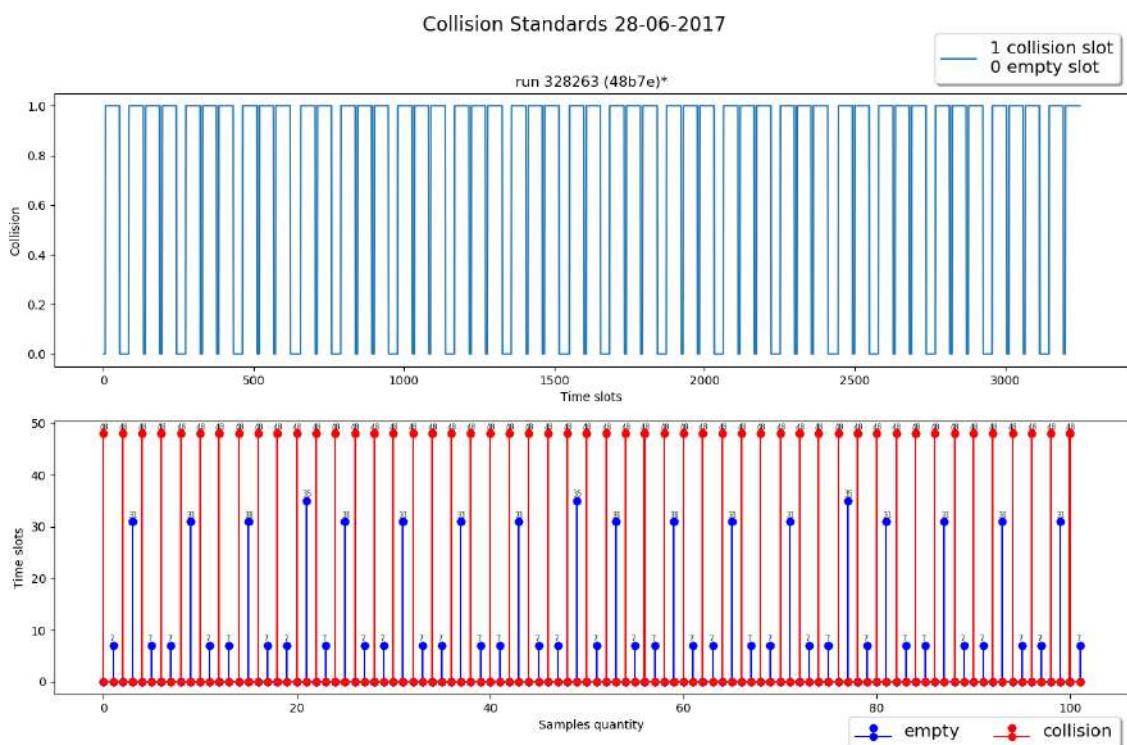
Fonte: Elaborado pelo autor (2021).

Gráfico 3 - Padrão de colisão run 300908



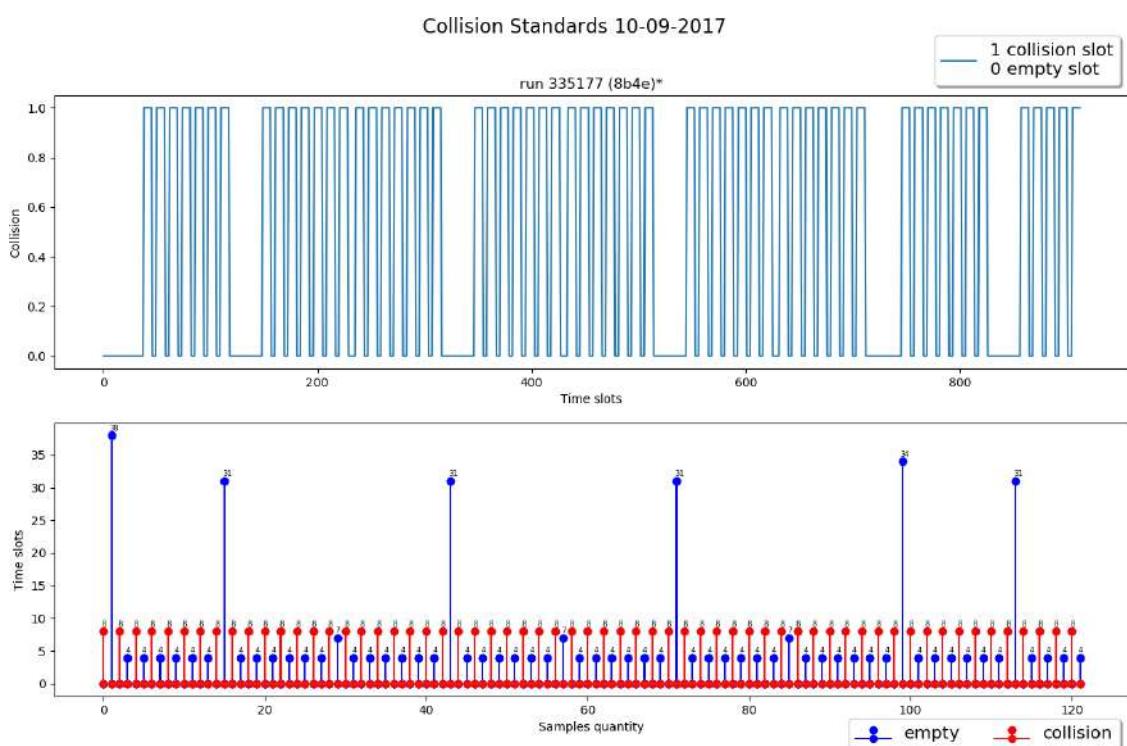
Fonte: Elaborado pelo autor (2021).

Gráfico 4 - Padrão de colisão run 328263



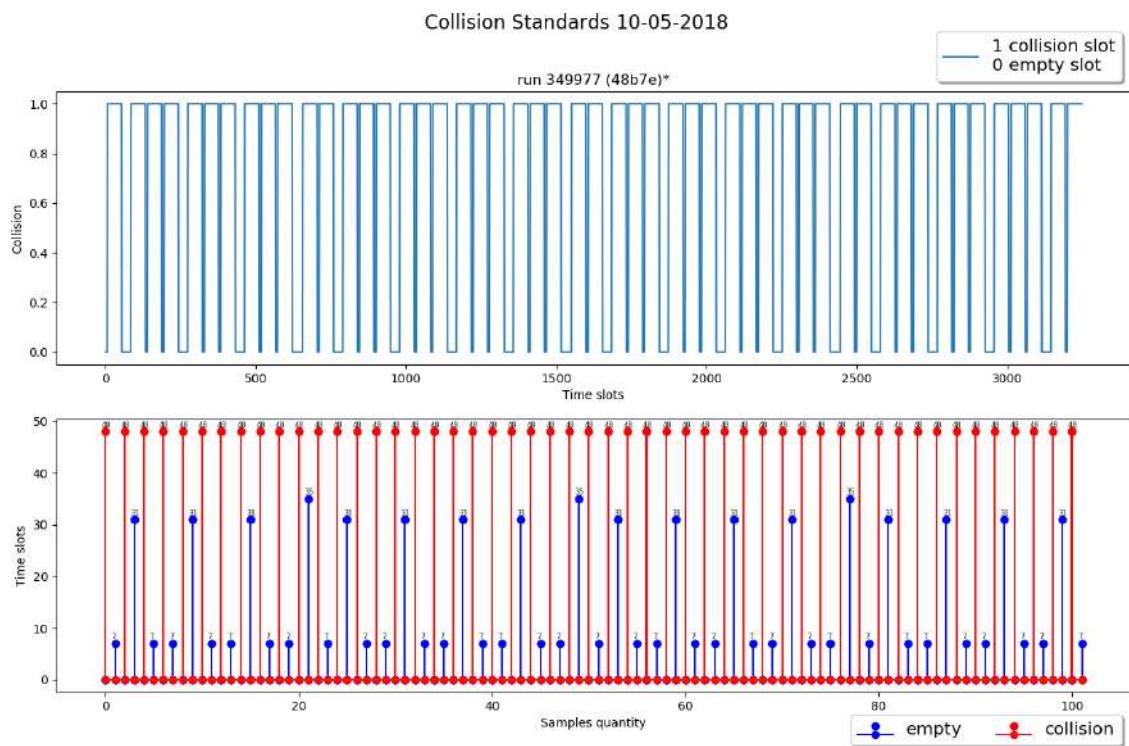
Fonte: Elaborada pelo autor (2021).

Gráfico 5 - Padrão de Colisão run 335177



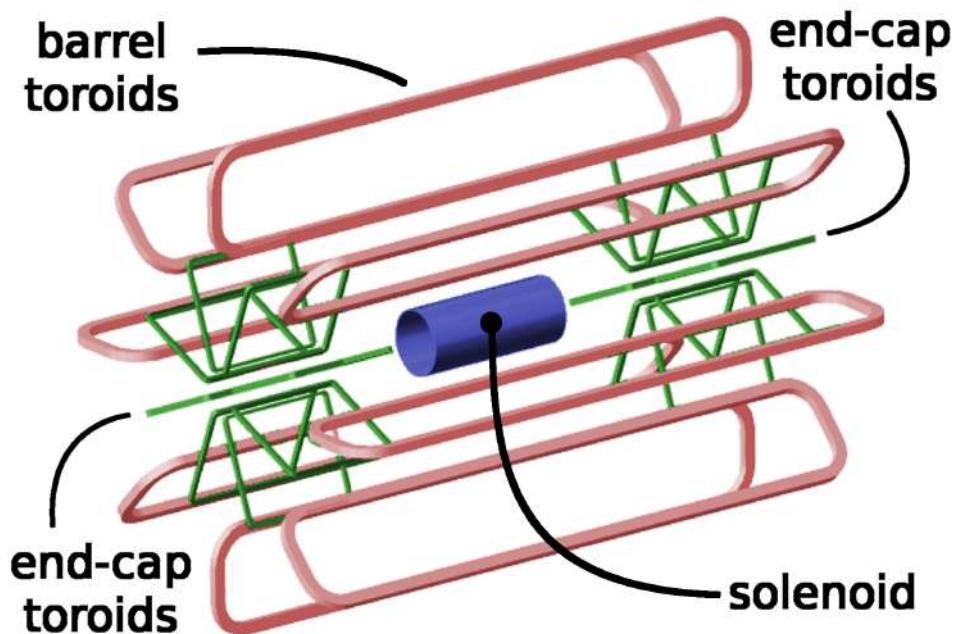
Fonte: Elaborado pelo autor (2021).

Gráfico 6 - Padrão de colisão run 349977



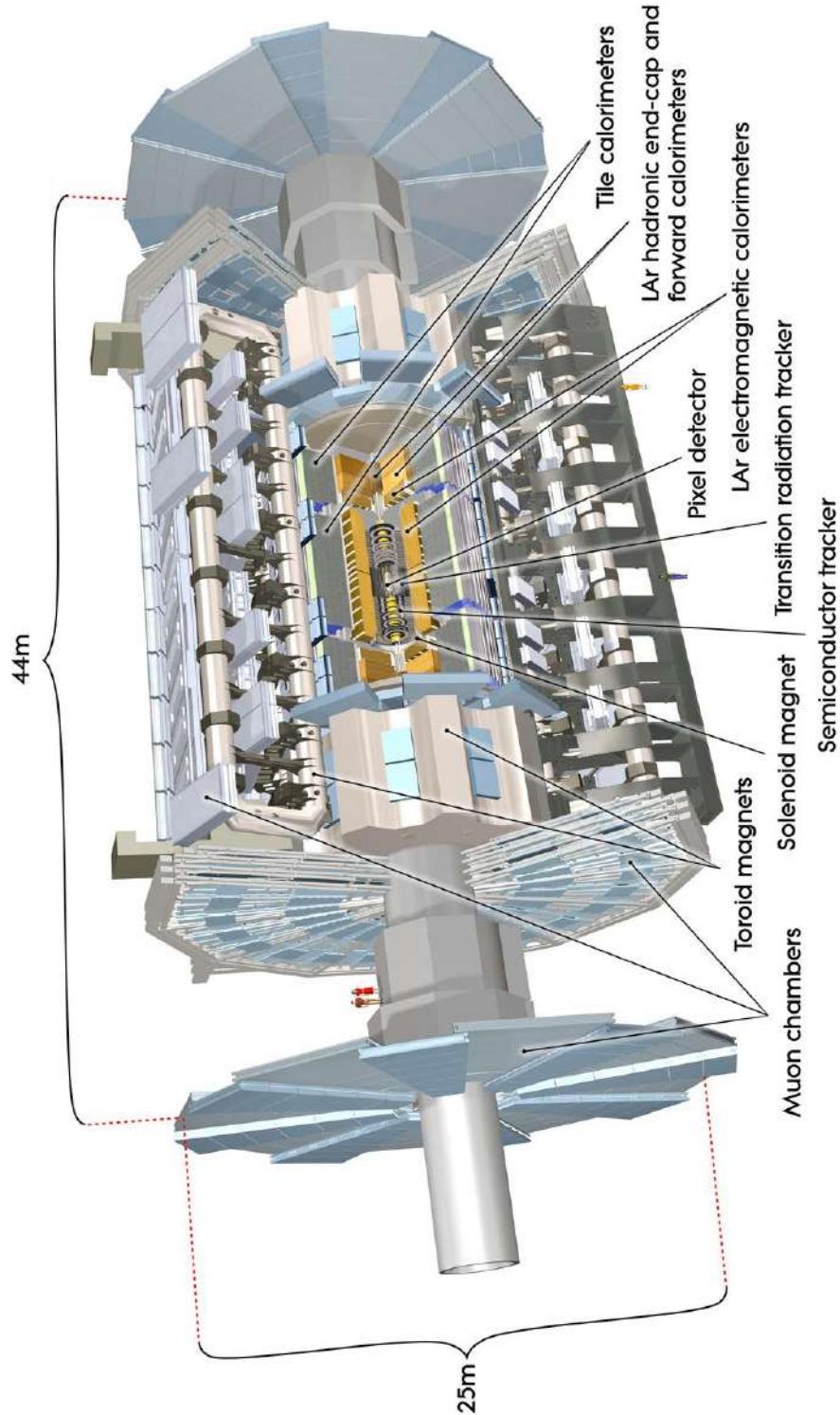
Fonte: Elaborado pelo autor (2021).

Figura 13 - Diagrama esquemático do sistema magnético do ATLAS



Fonte: Goodson (2012, p. 49).

Figura 14 - Vista detalhada do detector ATLAS

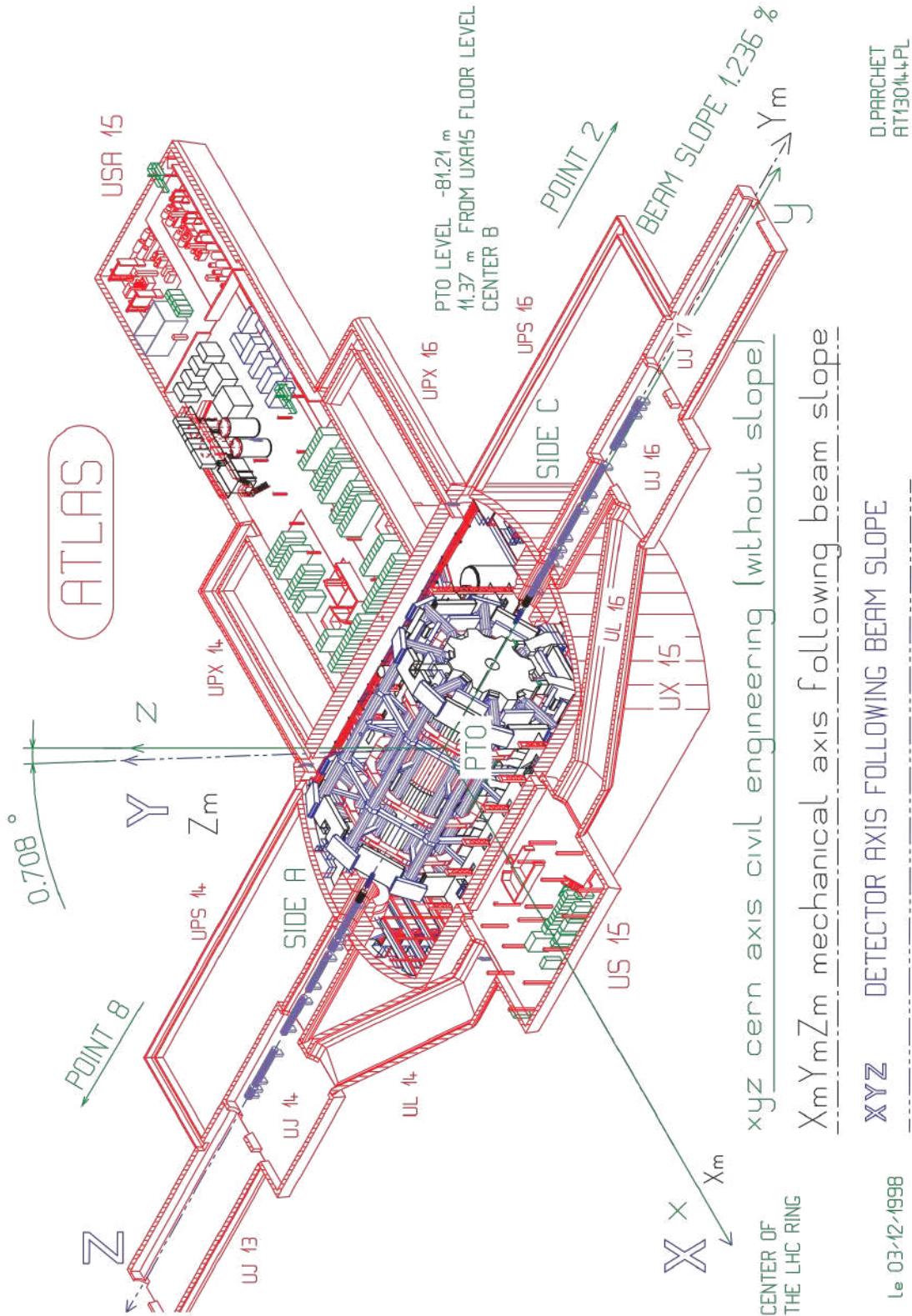


Nota(s): As dimensões do detector são de 25m de altura e 44m de comprimento. O peso total do detector é de aproximadamente 7.000 toneladas.

Fonte: Pequenao (2008e)³¹.

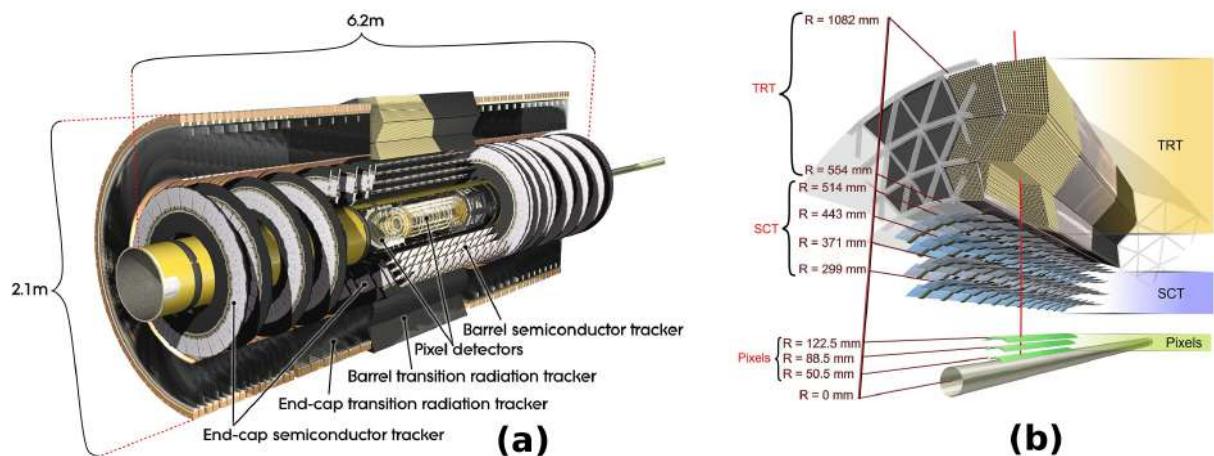
³¹Disponível em: <https://cds.cern.ch/record/1095924>

Figura 15 - Sistemas de coordenadas cartesianas do ATLAS



Fonte: CERN (1999, p. 92).

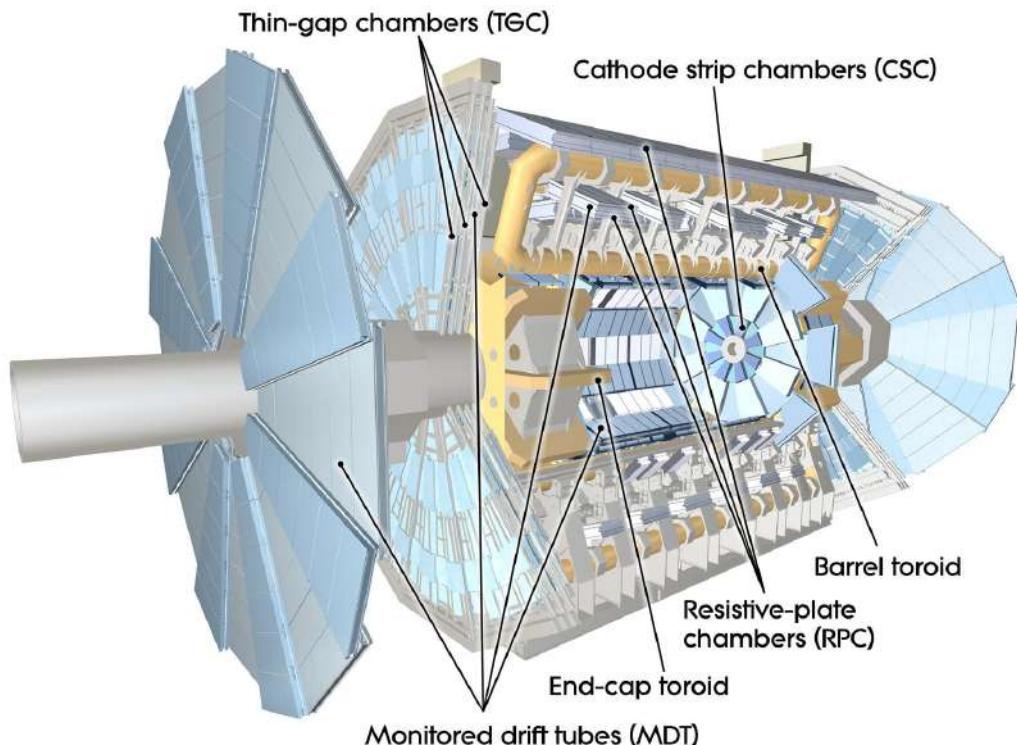
Figura 16 - Inner Detector



Legenda: (a) Conjunto de detectores que formam o ATLAS Inner Detector; (b) Detalhe da organização em camadas dos detectores presentes em (a).

Fonte: Pequenao (2008b)³².

Figura 17 - ATLAS subsistema de múons

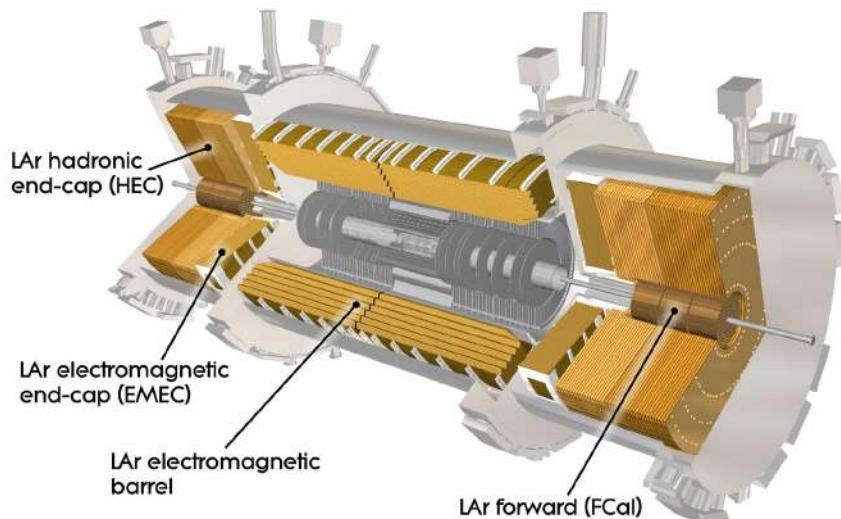


Fonte: Pequenao (2008d)³³.

³²Disponível em: <https://cds.cern.ch/images/CERN-GE-0803014-01>

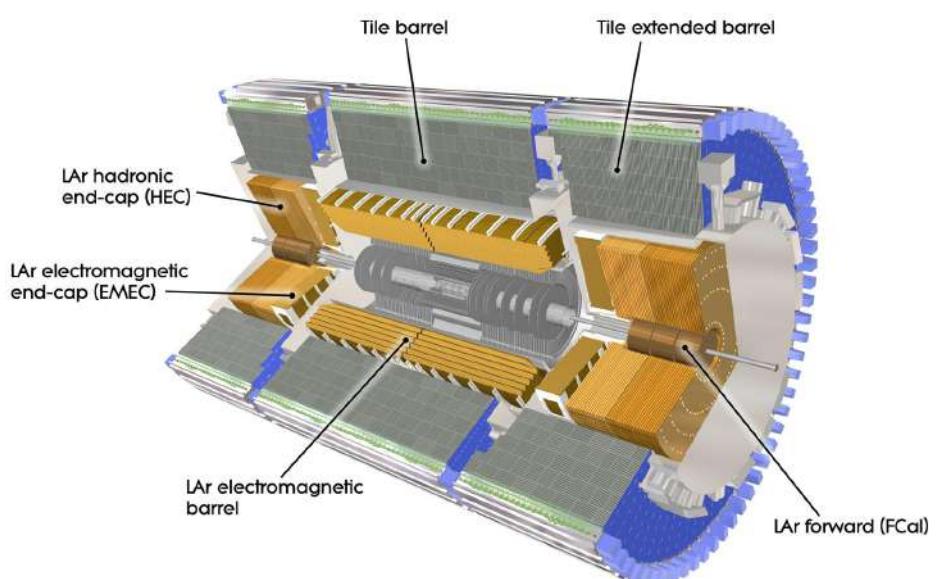
³³Disponível em: <https://cds.cern.ch/images/CERN-GE-0803017-01>

Figura 18 - Calorímetro de argônio líquido



Fonte: Pequenao (2008c)³⁴.

Figura 19 - Calorímetro de telhas envolvendo o calorímetro de argônio líquido

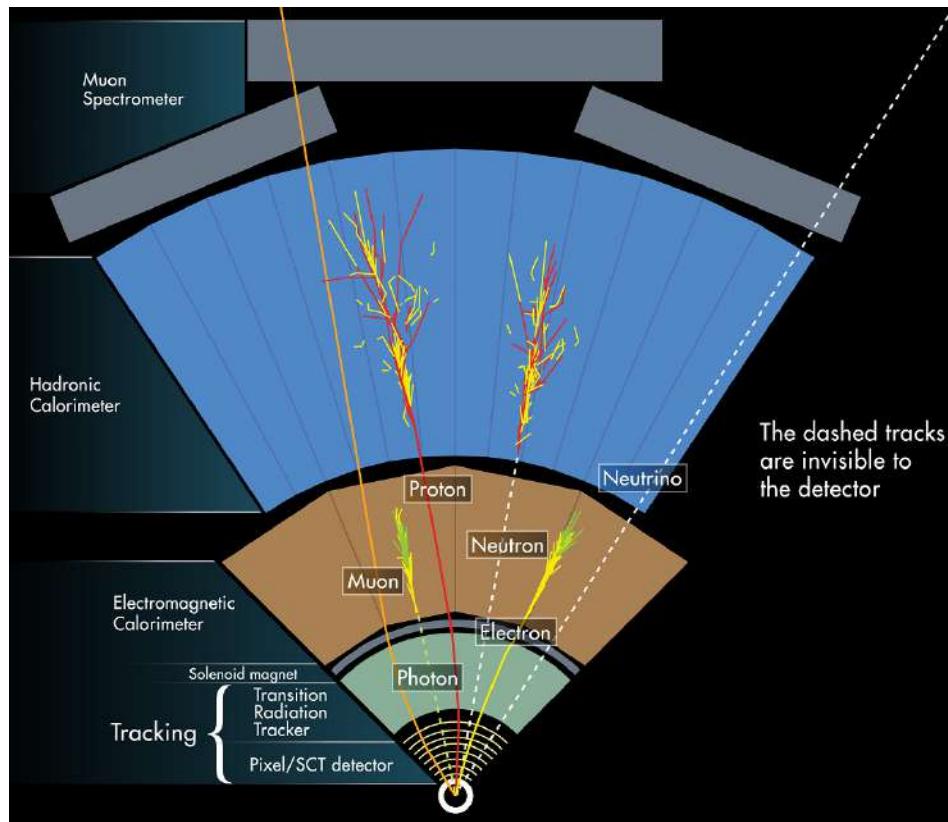


Fonte: Pequenao (2008a)³⁵.

³⁴Disponível em: <https://cds.cern.ch/images/CERN-GE-0803016-01>

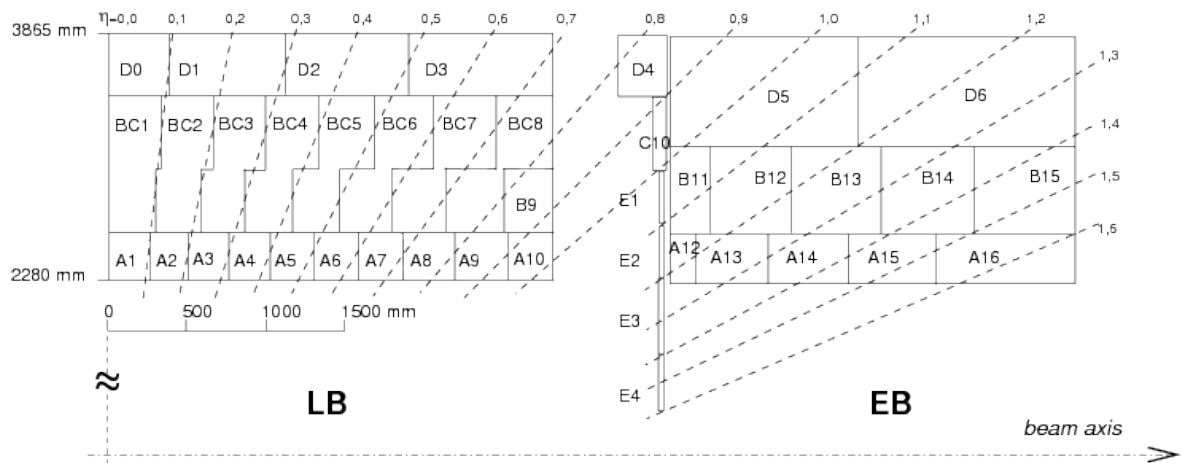
³⁵Disponível em: <https://cds.cern.ch/images/CERN-GE-0803015-01>

Figura 20 - Os quatro momentos de medições, para um grande range de energias de partículas.



Fonte: Murk (2016, p. 13).

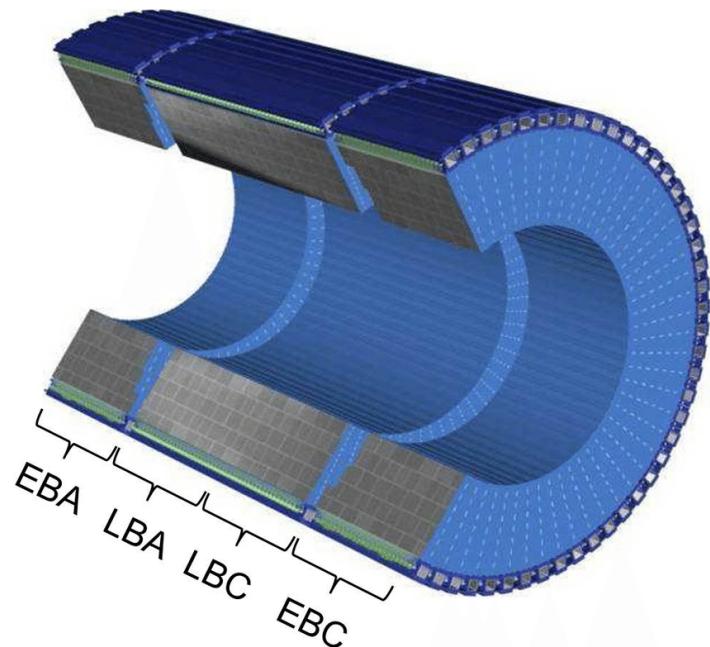
Figura 21 - Esquema da disposição das células num plano paralelo ao eixo do feixe, mostrando apenas o lado positivo η (o detector é simétrico em relação a $\eta = 0$)



Nota(s): As três camadas radiais (A, BC e D) também são visíveis. Cintiladores especiais, chamados de células *gap* (E1 e E2) e *crack* (E3 e E4), estão localizados entre o barril e a tampa da extremidade.

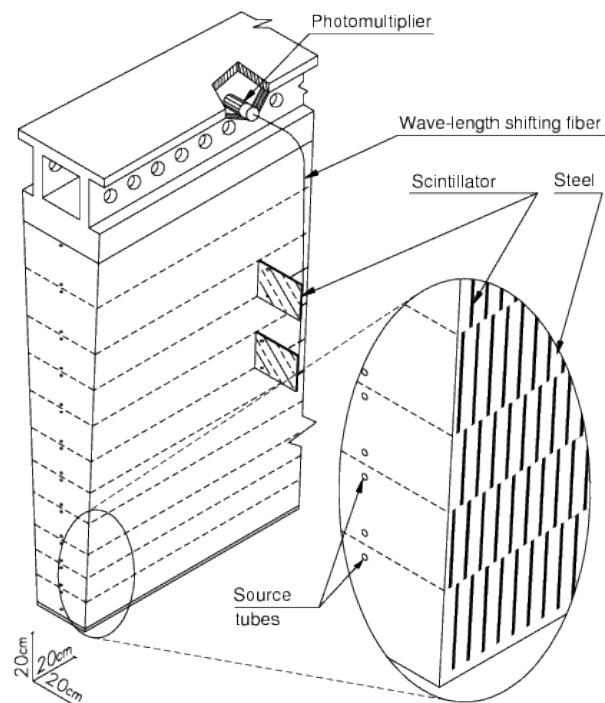
Fonte: Abdallah e Collaboration (2016, p. 4).

Figura 22 - O barril central e as duas tampas



Fonte: Carrio *et al.* (2013, p. 2).

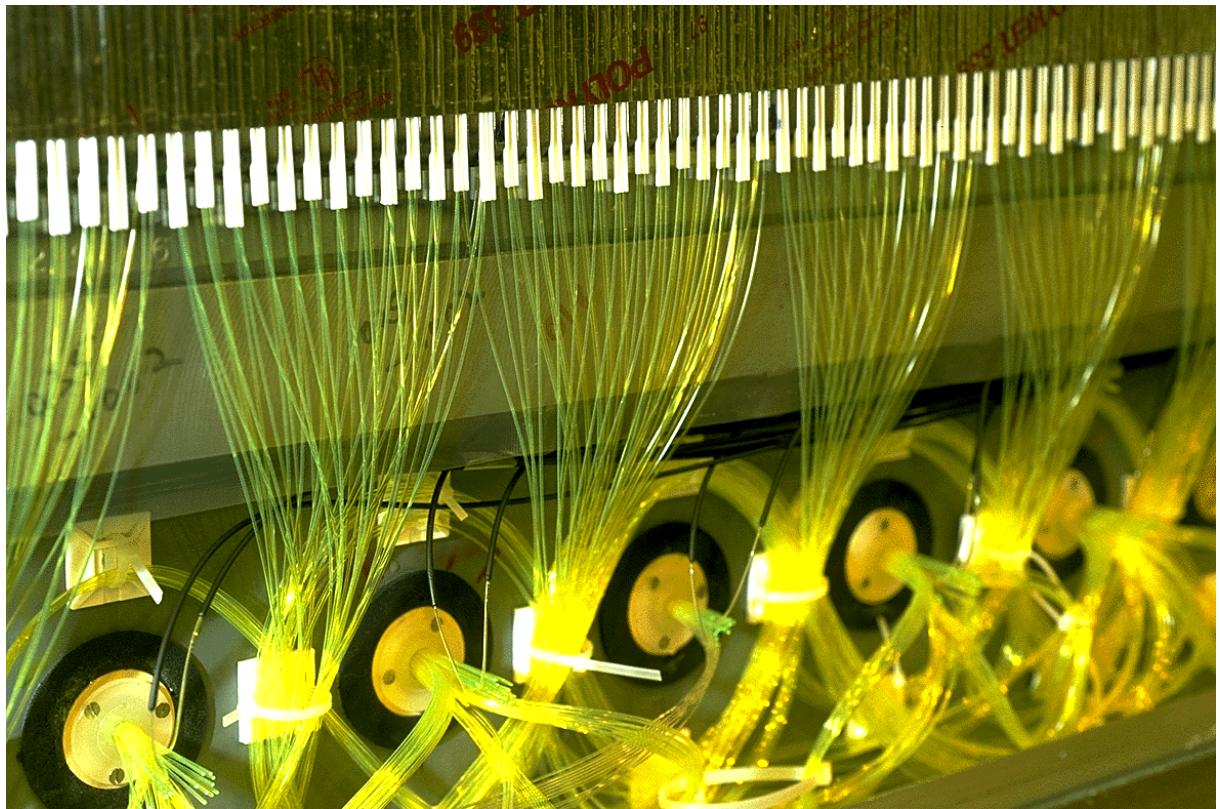
Figura 23 - Desenho de um módulo do TileCal



Nota(s): A figura mostra o caminho de leitura de luz entre as telhas cintilantes e os tubos fotomultiplicadores (PMTs).

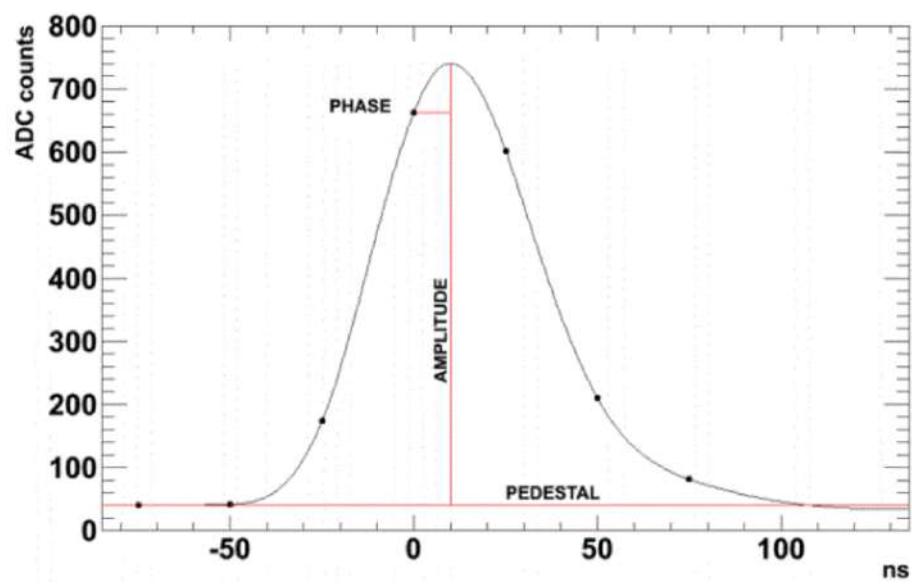
Fonte: Abdallah e Collaboration (2016, p. 4).

Fotografia 2 - Módulos dos barris estendidos sendo instalados no Conseil Européen pour la Recherche Nucléaire



Fonte: CERN (1998)³⁶.

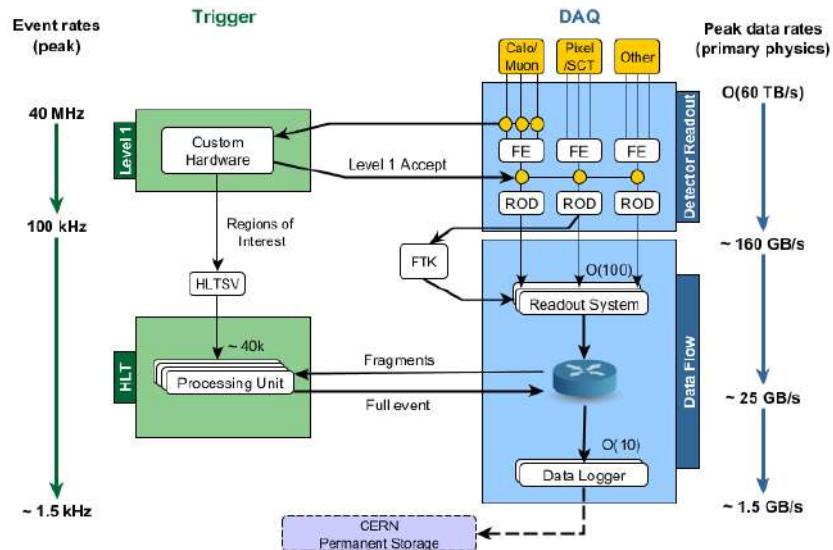
Gráfico 7 - Formato de um pulso característico do TileCal



Fonte: Peralva *et al.* (2013, p. 2).

³⁶Disponível em: <http://cds.cern.ch/record/40173>

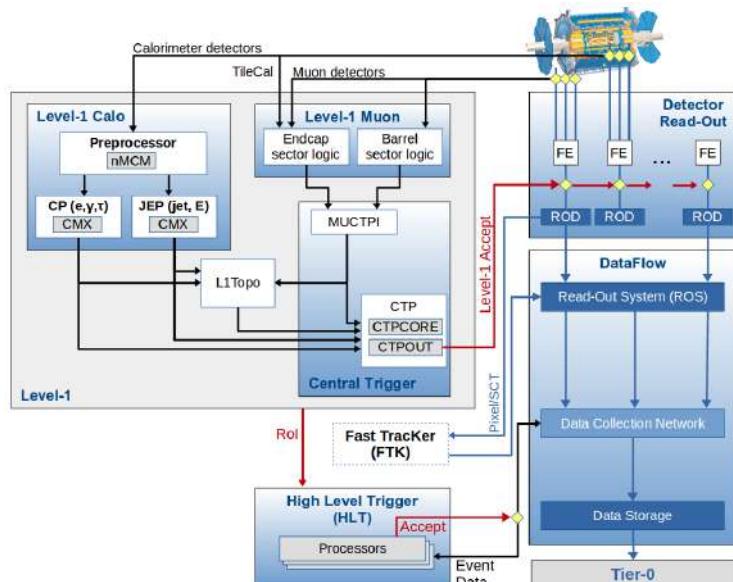
Diagrama 1 - Diagrama funcional do sistema de seleção de eventos e aquisição de dados do ATLAS



Nota(s): O diagrama mostra as taxas de pico esperadas e as larguras de banda em cada componente.

Fonte: Vazquez (2017, p. 3).

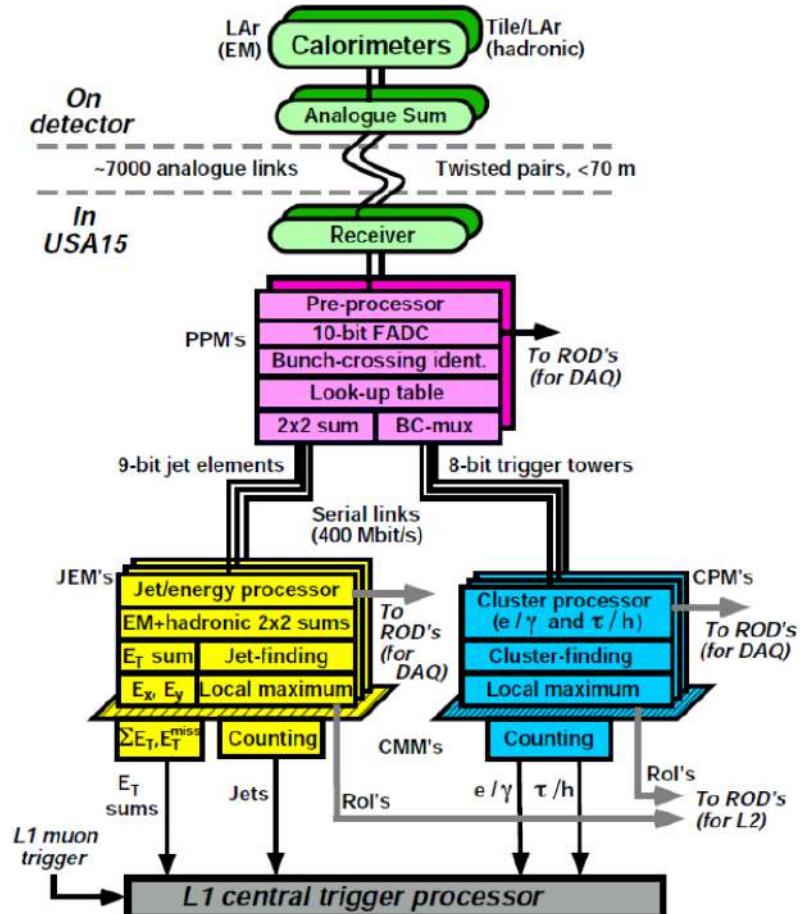
Diagrama 2 - Componentes do sistema de *trigger* Level 1



Nota(s): Esquema do sistema TDAQ do ATLAS, com foco específico dado aos componentes do sistema de *trigger* L1.

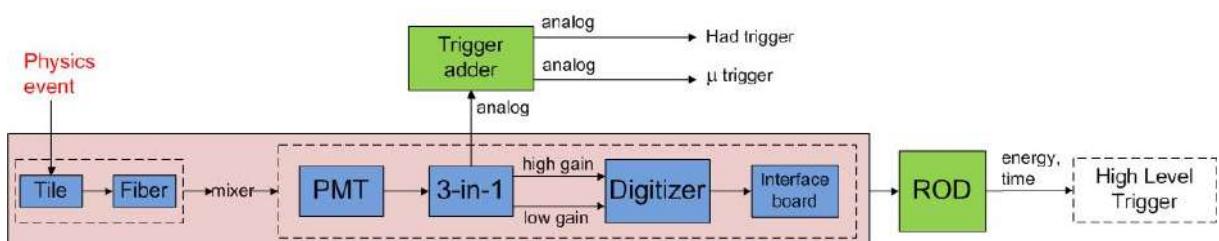
Fonte: Collaboration (2020, p. 3).

Diagrama 3 - Sistema de trigger L1 do TileCal



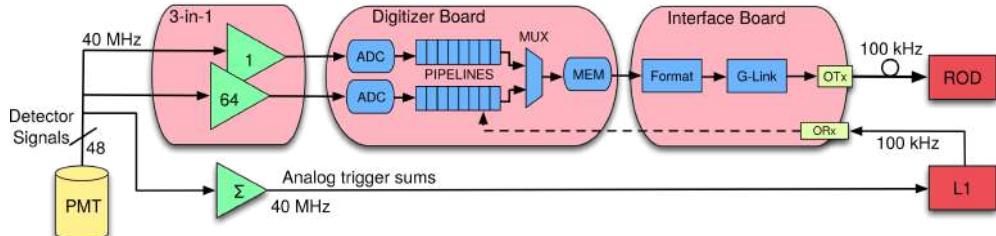
Fonte: Asman *et al.* (2012, p. 3).

Diagrama 4 - Cadeia de sinais do TileCal



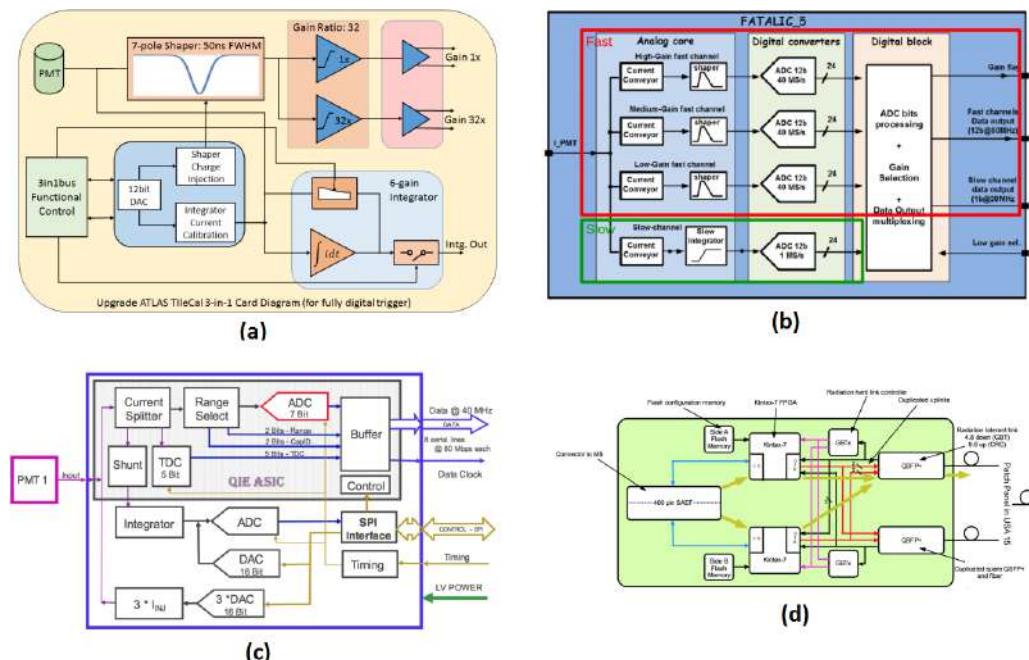
Fonte: Cerqueira (2015, p. 3).

Diagrama 5 - Esboço da eletrônica atual de leitura de *front-end* do TileCal



Fonte: Solodkov (2017, p. 2).

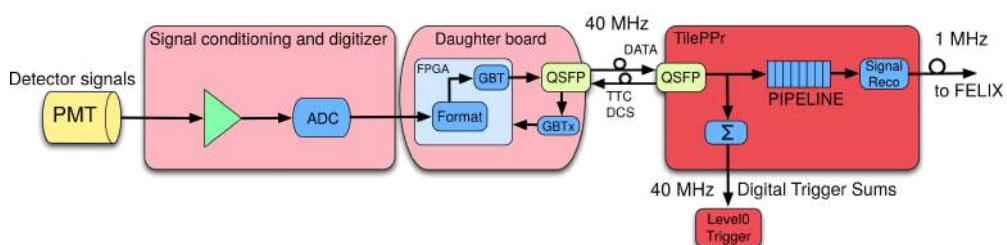
Figura 24 - Diagramas das três opções de Front-End Board



Legenda: (a) A placa de *front-end* 3-em-1, (b) A placa de *front-end* QIE, (c) Arquitetura básica do circuito FATALIC5; e (d) A organização da DB.

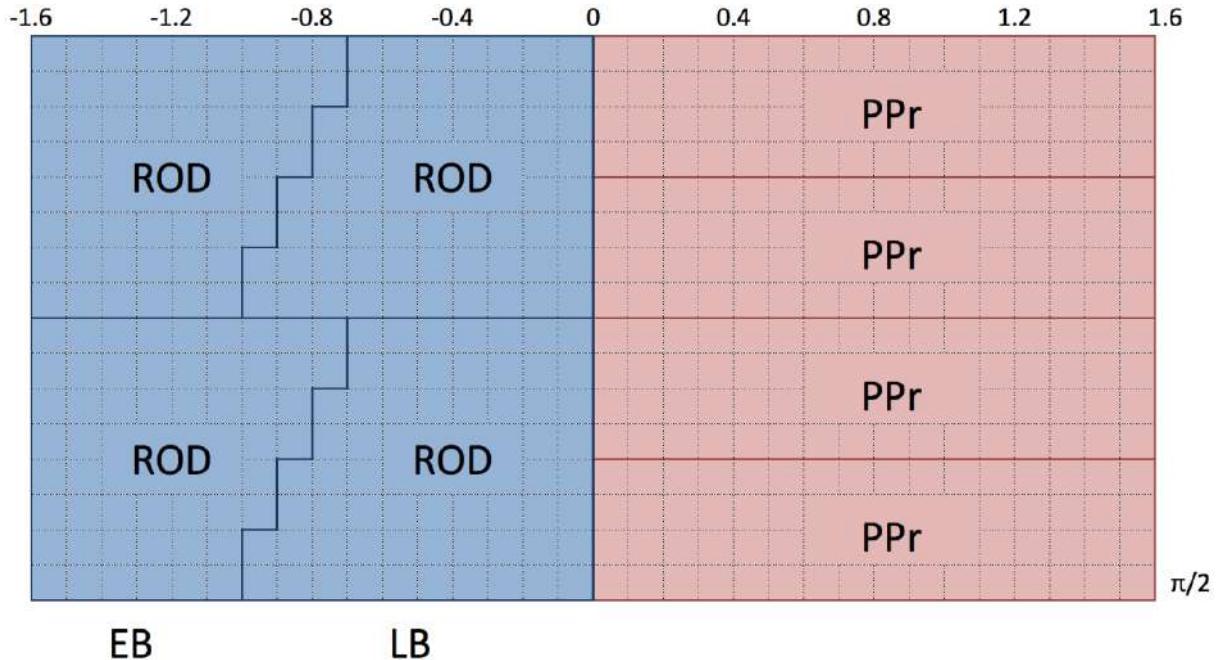
Fonte: Figura adaptada de Solodkov (2017, p. 3).

Diagrama 6 - Esboço da eletrônica de leitura do TileCal após o *upgrade* da Fase II



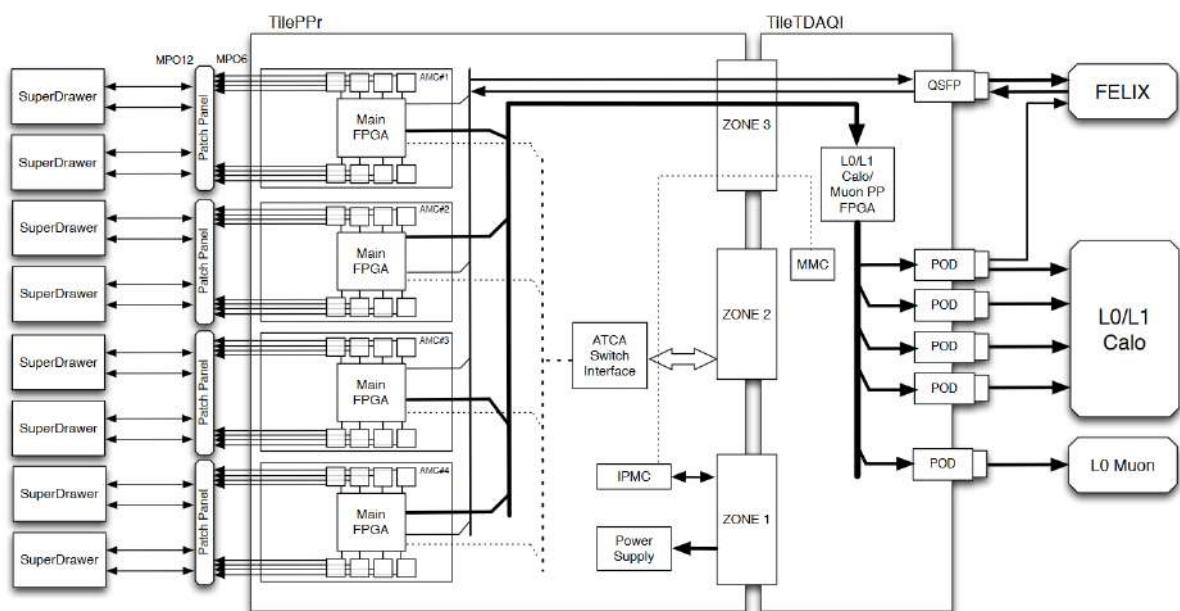
Fonte: Mlynarikova (2017, p. 1).

Figura 25 - Cobertura dos ângulos $\eta \times \phi$ dos módulos ROD e TilePPr



Fonte: Collaboration (2017a, p. 150).

Diagrama 7 - Blocos de um projeto TilePPr candidato final

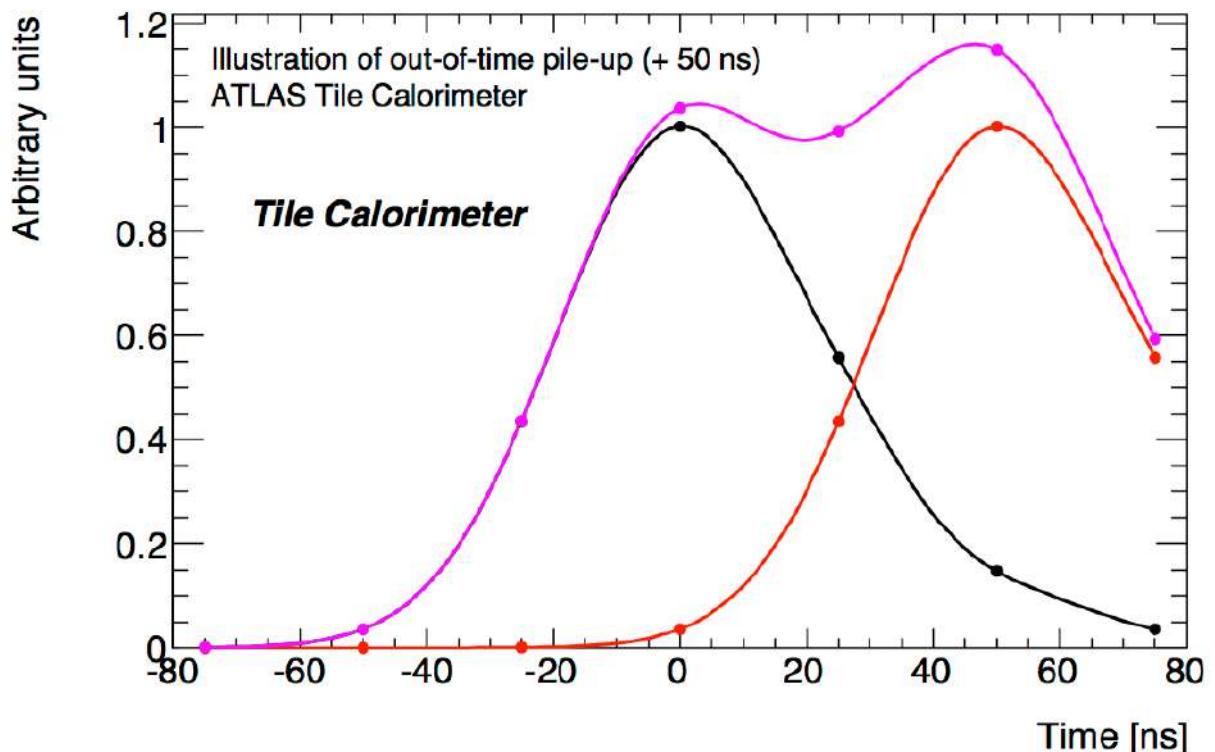


Nota(s): Um sistema PPr completo capaz de processar os dados de 8 *super-drawers*, consistindo em uma placa-mãe ATCA com 4 CPMs mais um TDAQi RTM.

Fonte: Collaboration (2017a, p. 144).

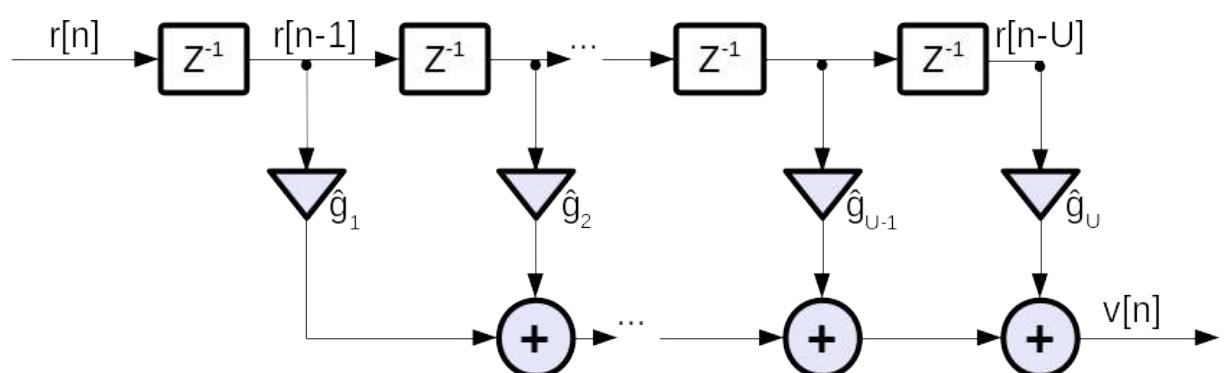
APÊNDICE B – Ilustrações da seção 3

Gráfico 8 - Simulação da ocorrência de um *pile-up* registrado pelo TileCal



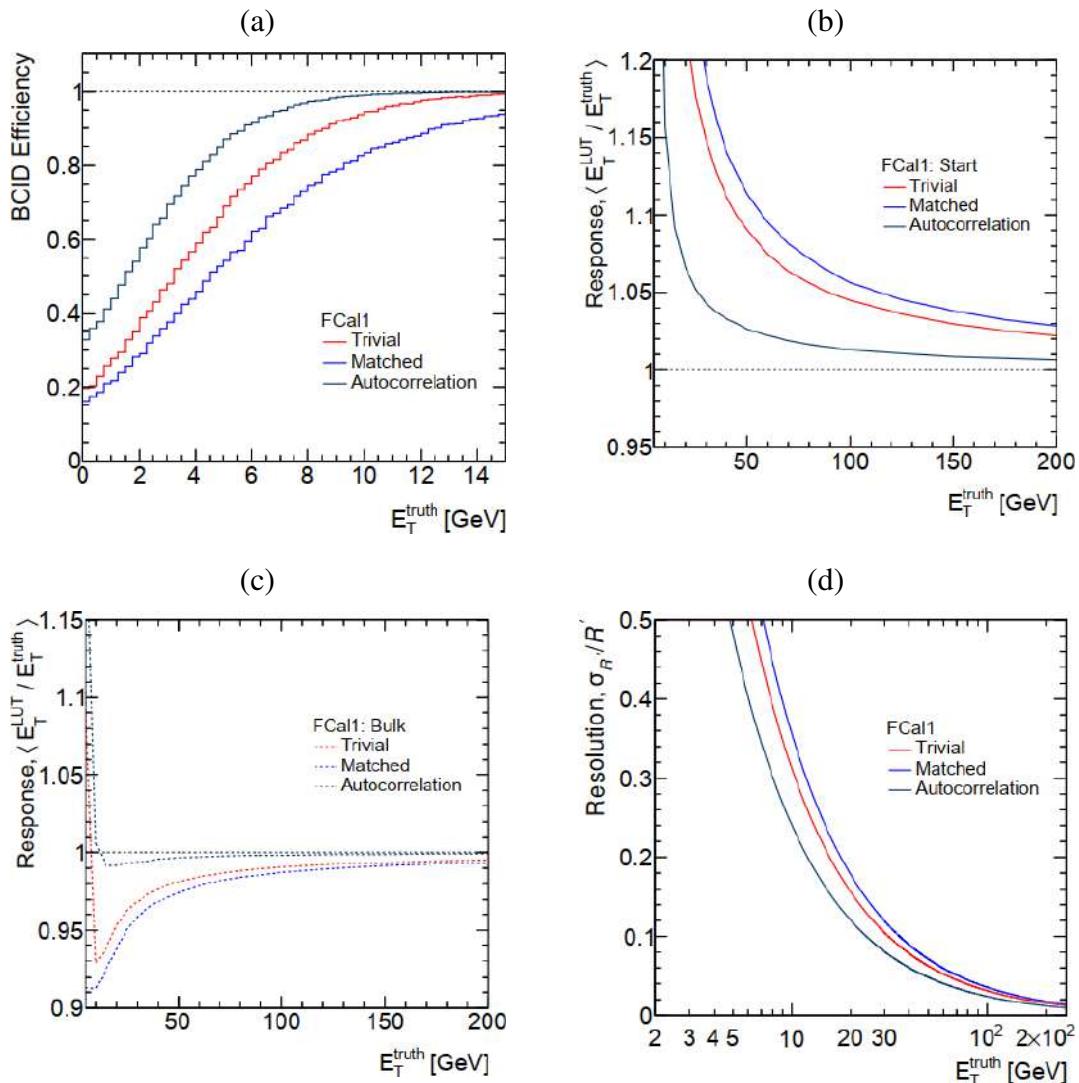
Fonte: Clement e Klimek (2011, p. 3).

Diagrama 8 - Forma de um Filtro FIR utilizando *Least Squares*



Fonte: Elaborada pelo autor (2021).

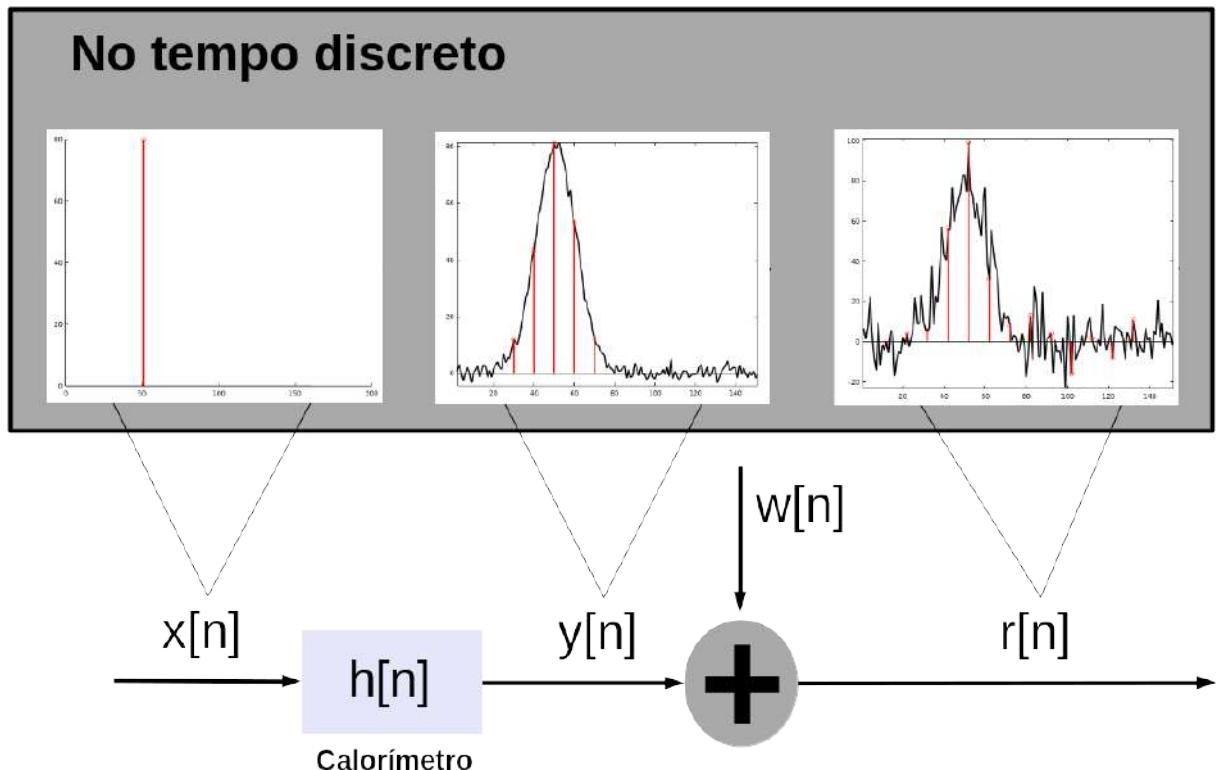
Gráfico 9 - Performance de diferentes estratégias de filtros usados no FCal1 em função de energia transversa



Legenda: (a) mostra a eficiência de detecção de BCID para a identificação correta de sinais; (b) e (c) mostram a resposta a eventos no início e no volume de trens de *bunches*, respectivamente; (d) mostra a resolução fracionada.

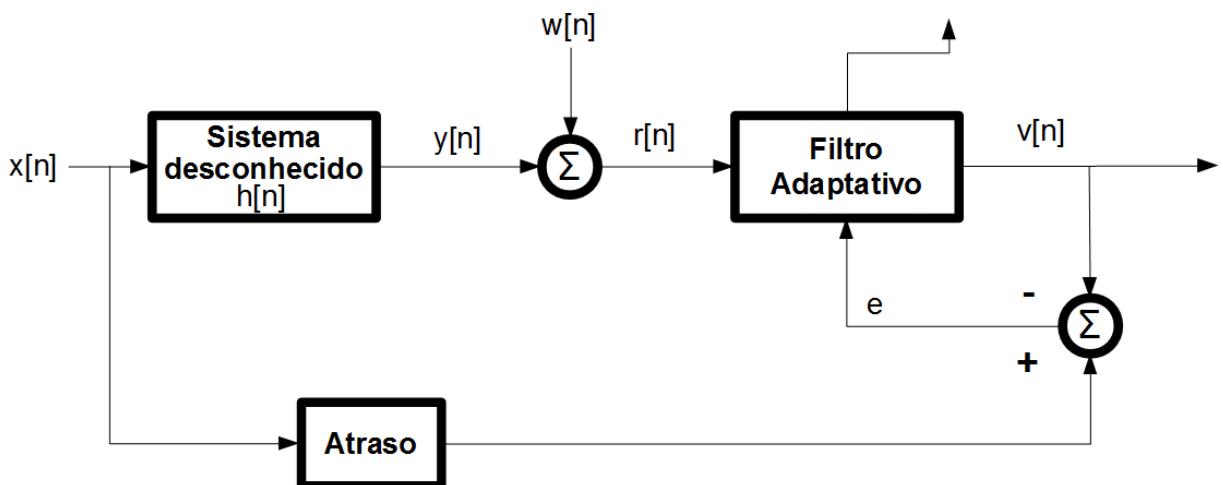
Fonte: Daniells (2016, p. 89).

Figura 26 - TileCal modelado como sistema LTI no tempo discreto



Fonte: Elaborada pelo autor (2021).

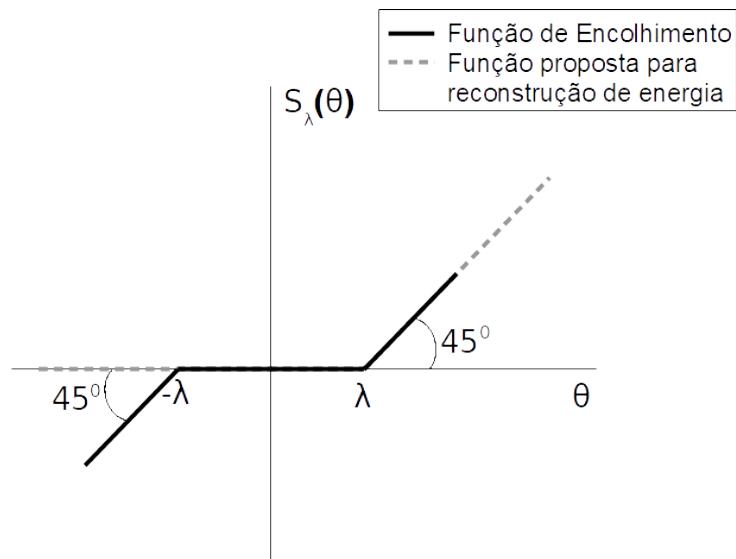
Diagrama 9 - Esquema representativo da modelagem inversa



Fonte: Elaborada pelo autor (2021).

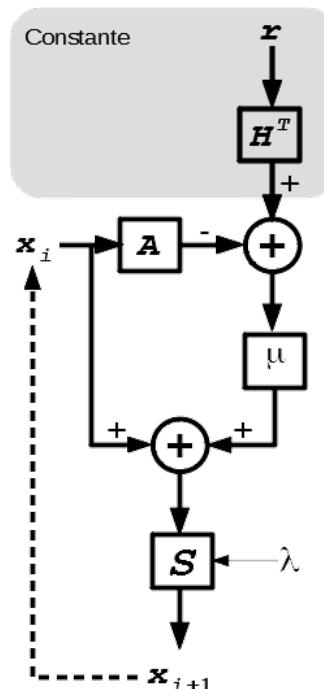
APÊNDICE C – Ilustrações da seção 4

Gráfico 10 - Expressão gráfica da função de encolhimento e a versão modificada proposta para promover a reconstrução positiva da energia



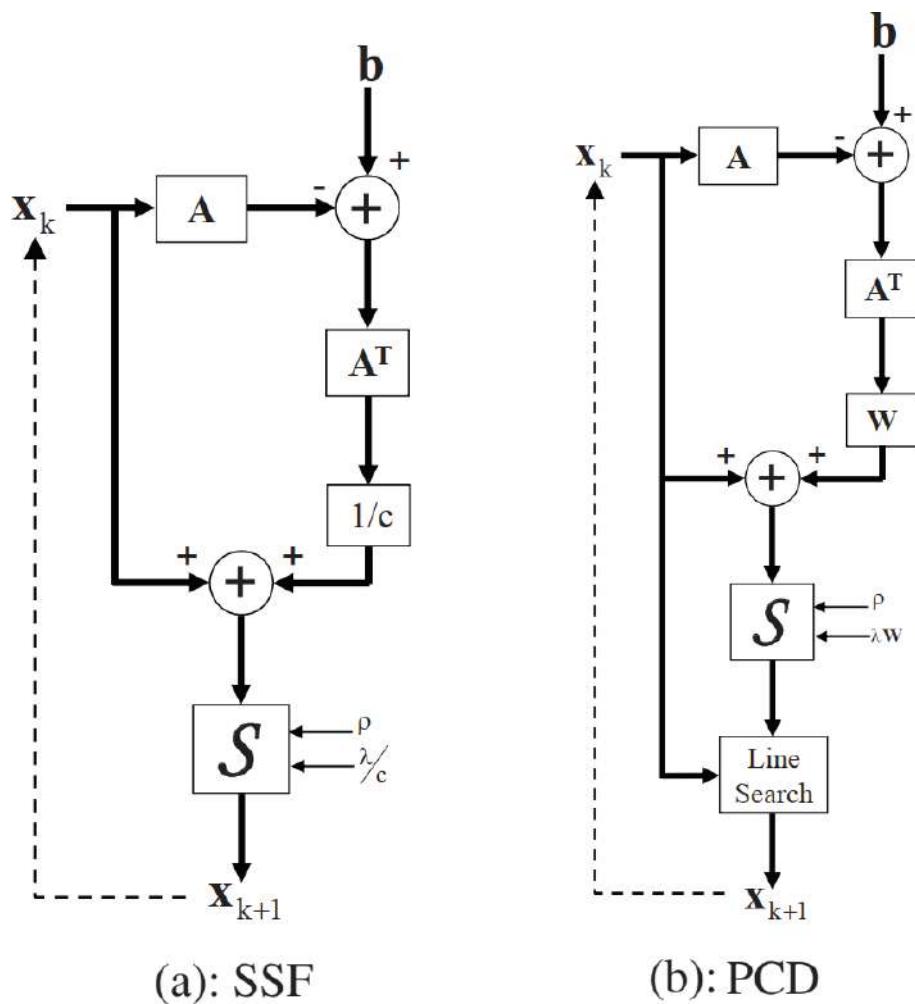
Fonte: Elaborada pelo autor (2021).

Diagrama 10 - Estrutura do SSF modificado



Fonte: Elaborada pelo autor (2021).

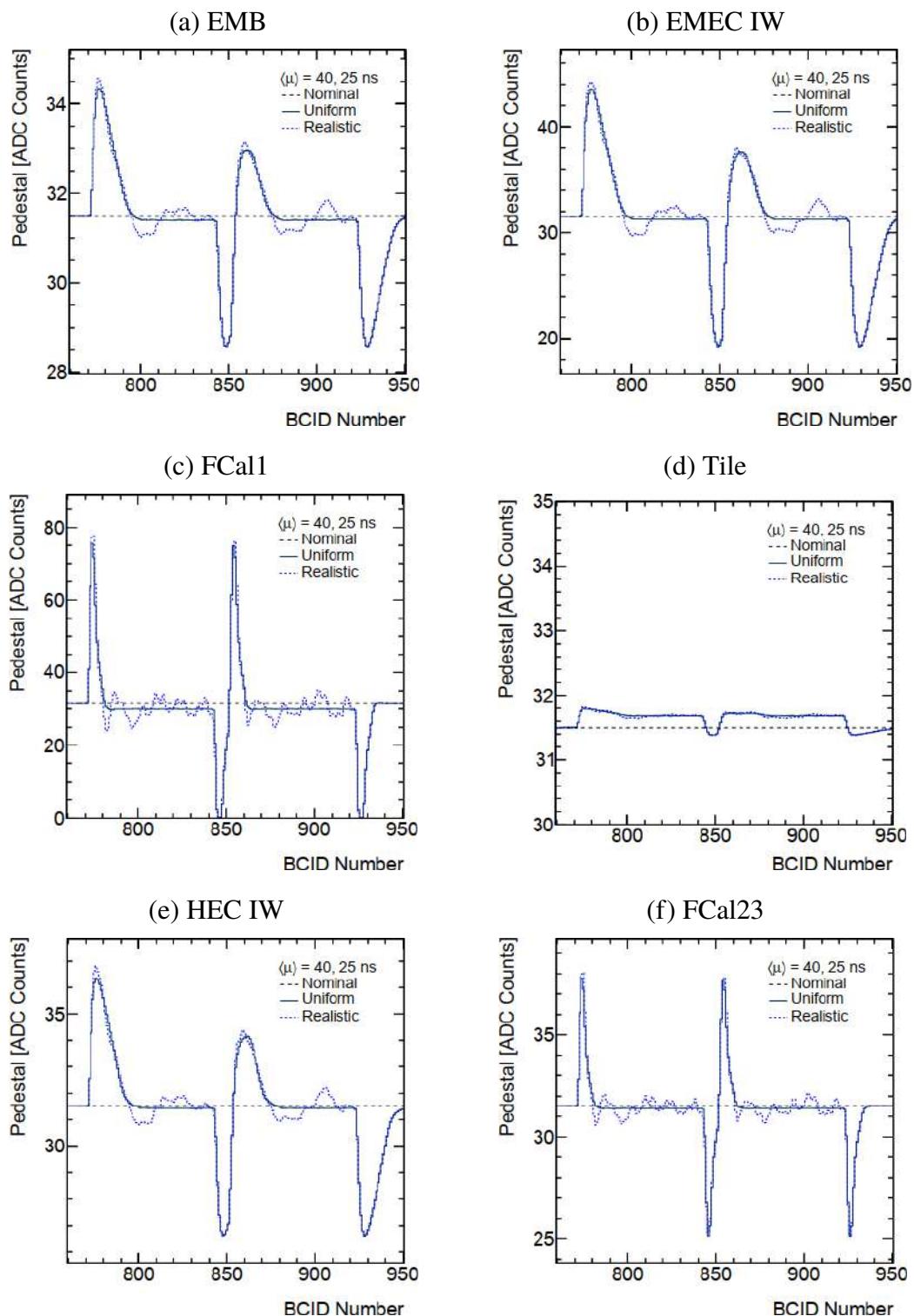
Diagrama 11 - Fluxo de dados em Separable surrogate functionals e Parallel-coordinate-descent



Fonte: Elaborada pelo autor (2021).

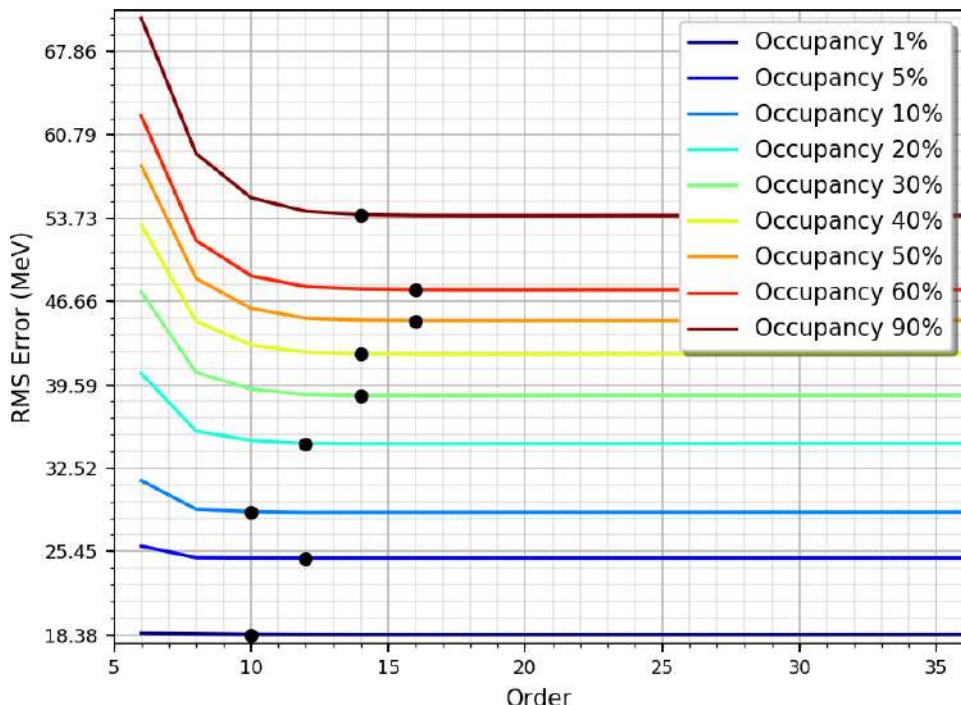
APÊNDICE D – Ilustrações da seção 5

Gráfico 11 - Exemplos de mudanças de pedestal para luminosidades uniformes e realistas em diferentes regiões do calorímetro



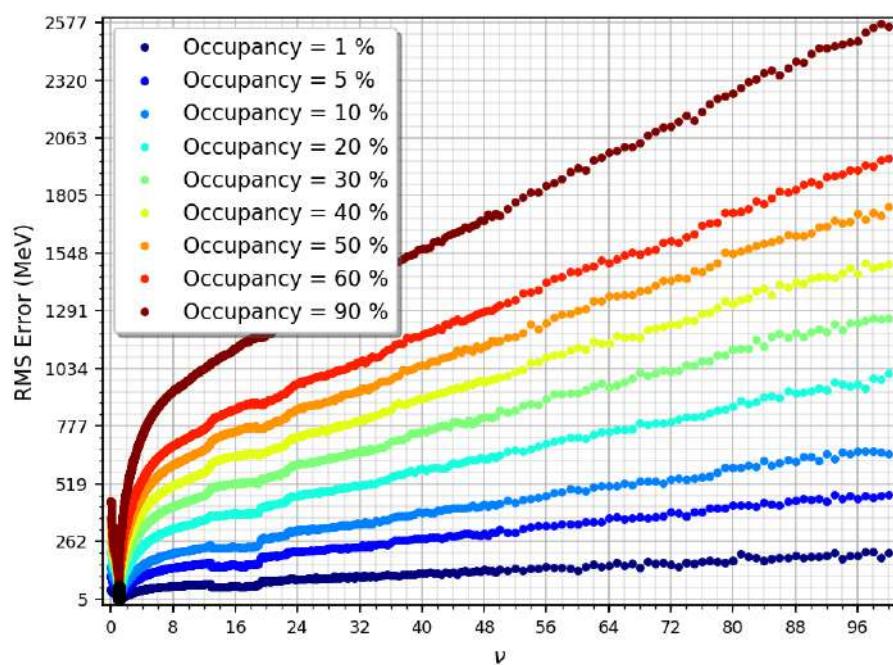
Fonte: Adaptado de Daniells (2016, p. 82).

Gráfico 12 - Relação da ordem do Filtro FDIP e o valor de RMSr para cada taxa de ocupação



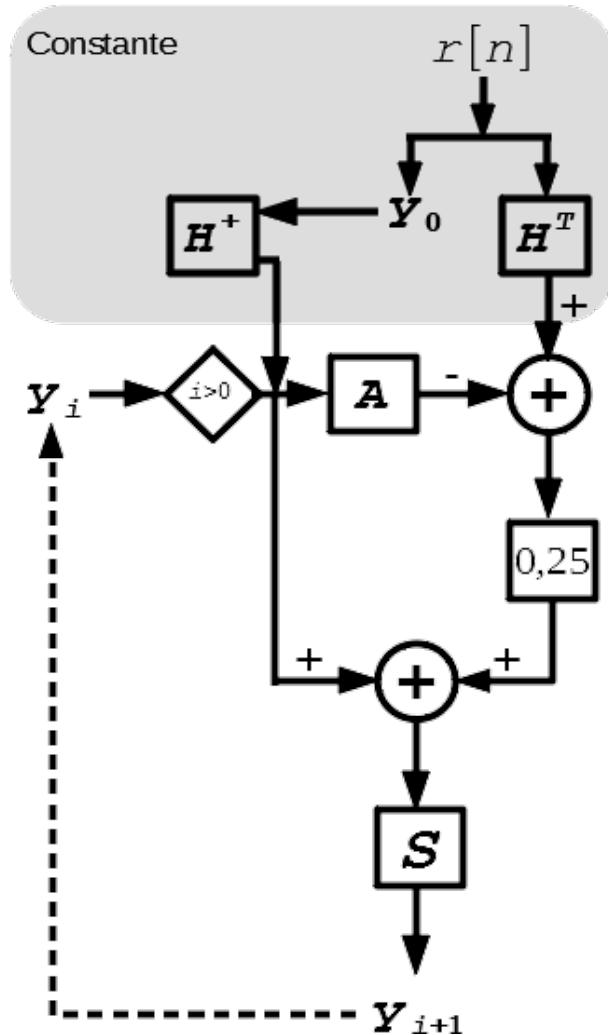
Fonte: Elaborado pelo autor (2021).

Gráfico 13 - Valores de RMSr obtidos com a variação dos valores de ν para o método SSF, de acordo com as taxas de ocupação



Fonte: Elaborado pelo autor (2021).

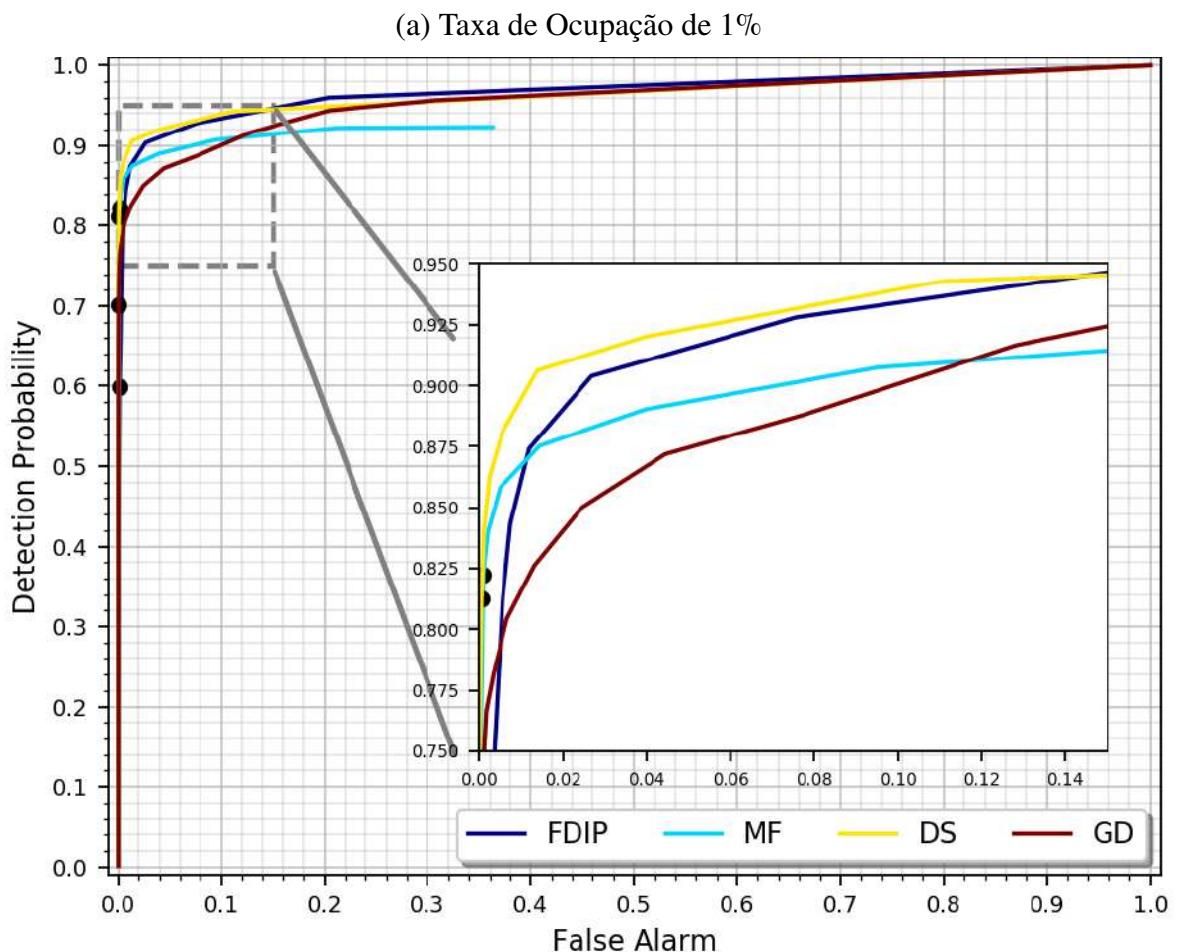
Diagrama 12 - Estrutura do SSFi modificado



Fonte: Elaborado pelo autor (2021).

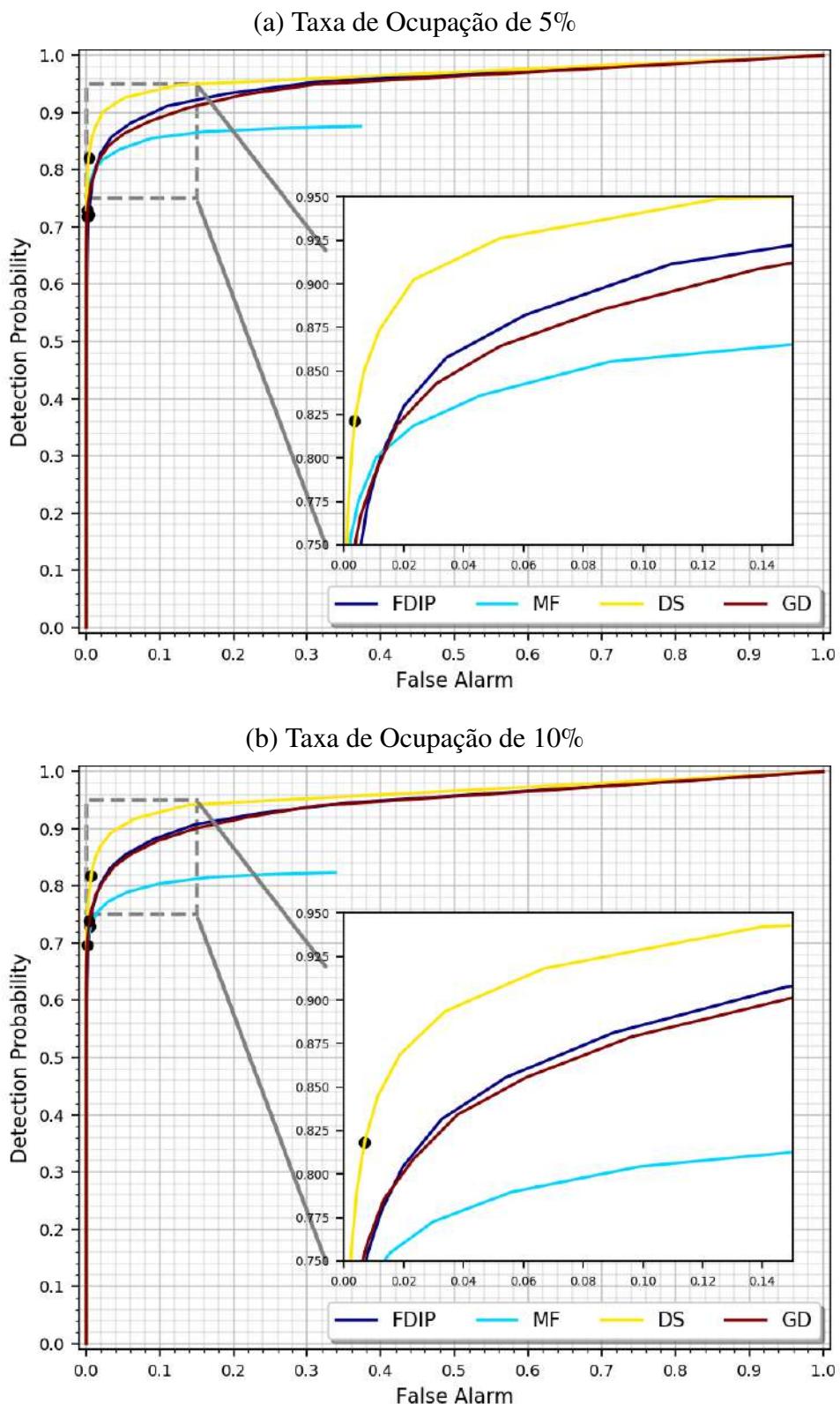
D.1 Detalhamento Gráfico dos Métodos já Propostos

Gráfico 14 - Curvas ROC dos métodos Filtro Casado, Filtro FDIP, DS e GD, para taxa de ocupação de 1%



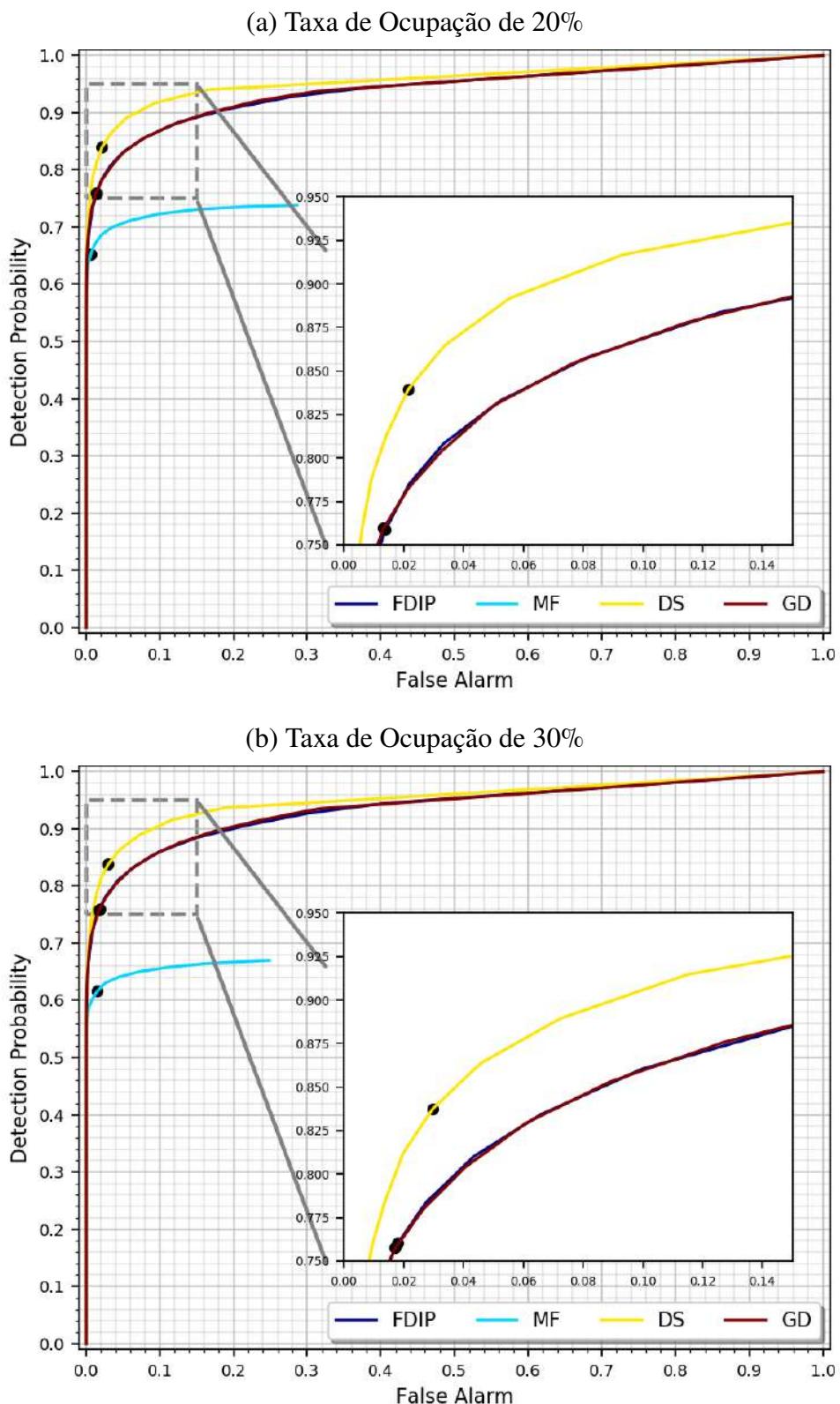
Fonte: Elaborada pelo autor (2021).

Gráfico 15 - Curvas ROC dos métodos Filtro Casado, Filtro FDIP, DS e GD, para taxas de ocupação de 5% e 10%



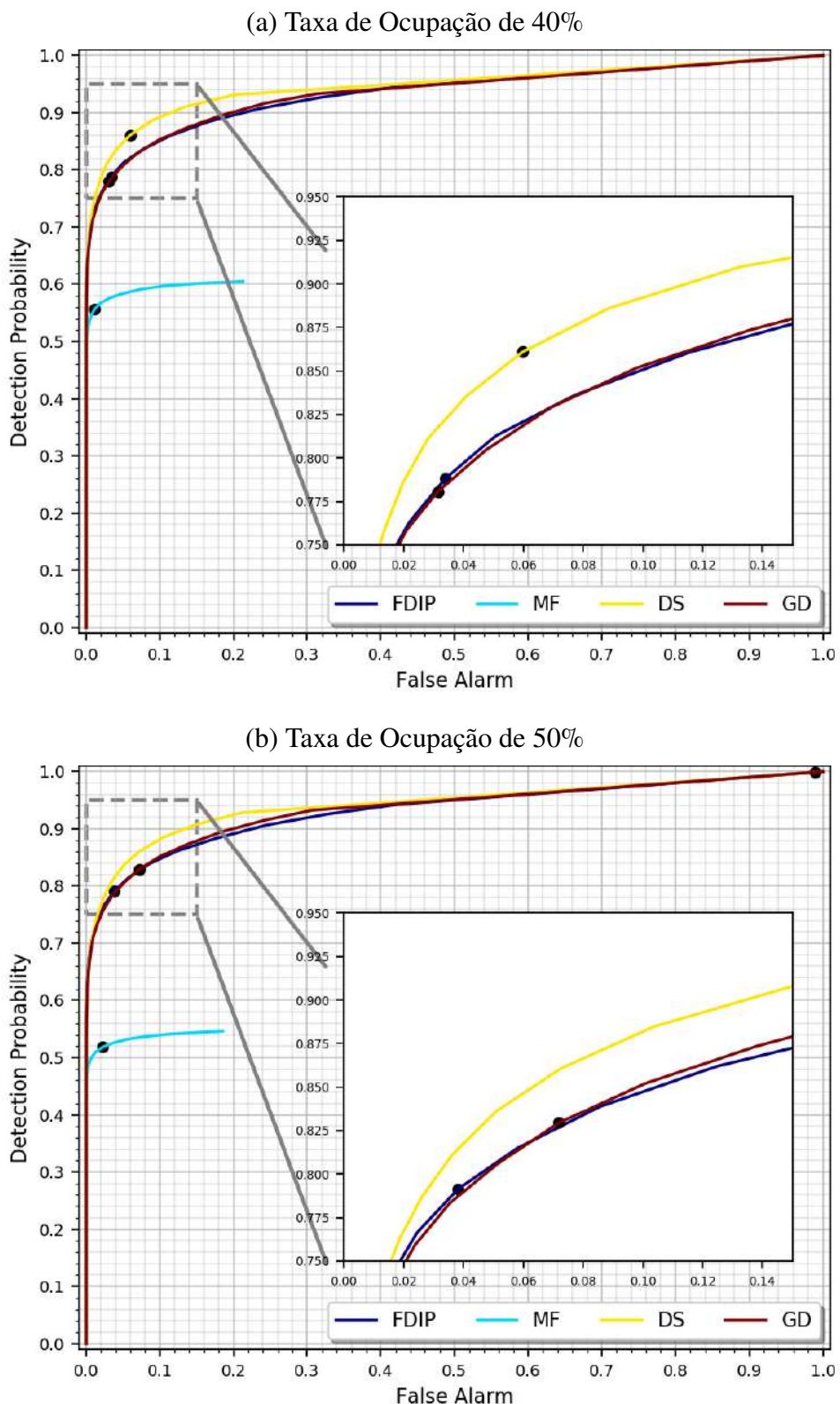
Fonte: Elaborada pelo autor (2021).

Gráfico 16 - Curvas ROC dos métodos Filtro Casado, Filtro FDIP, DS e GD, para taxas de ocupação de 20% e 30%.



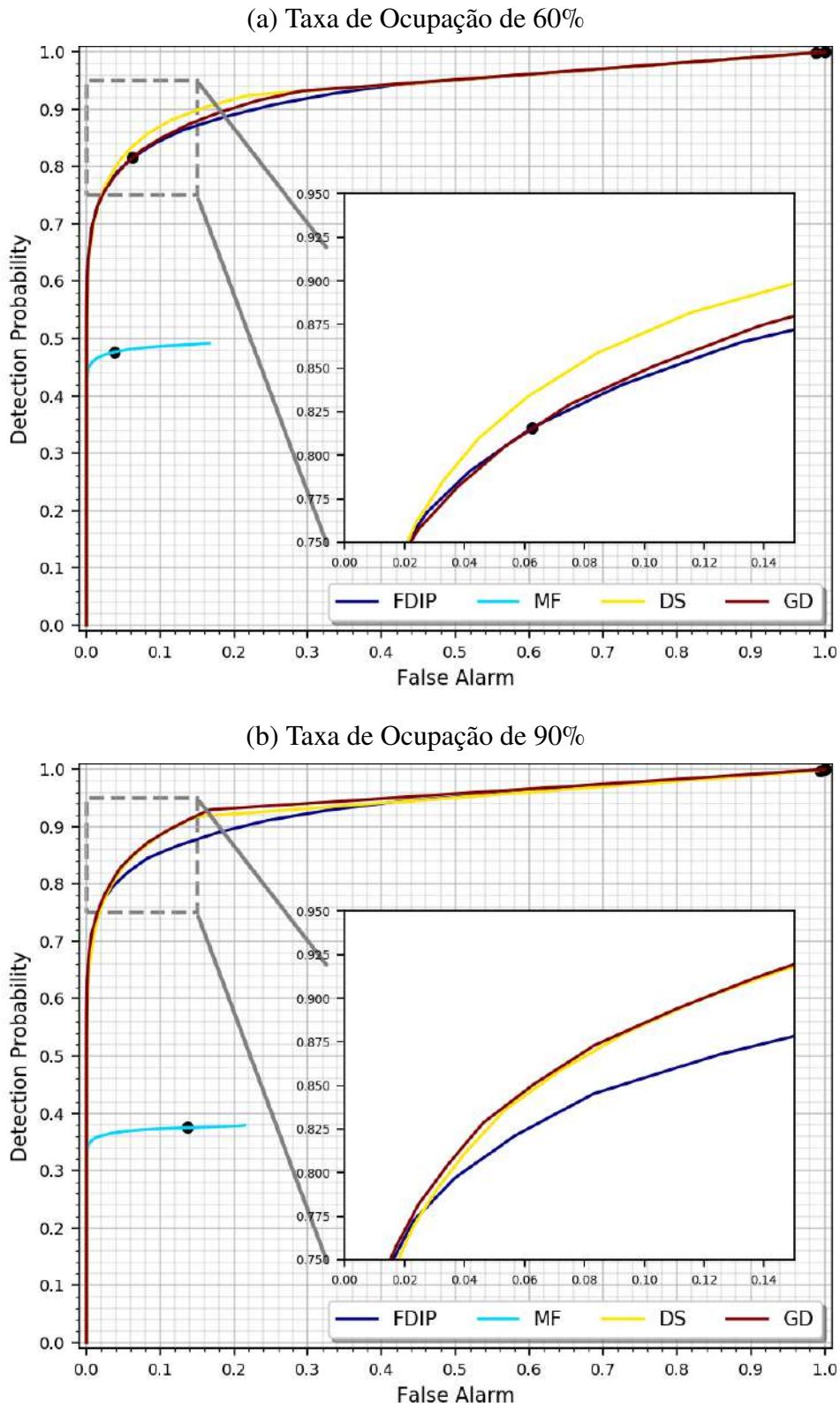
Fonte: Elaborada pelo autor (2021).

Gráfico 17 - Curvas ROC dos métodos Filtro Casado, Filtro FDIP, DS e GD, para taxas de ocupação de 40% e 50%.



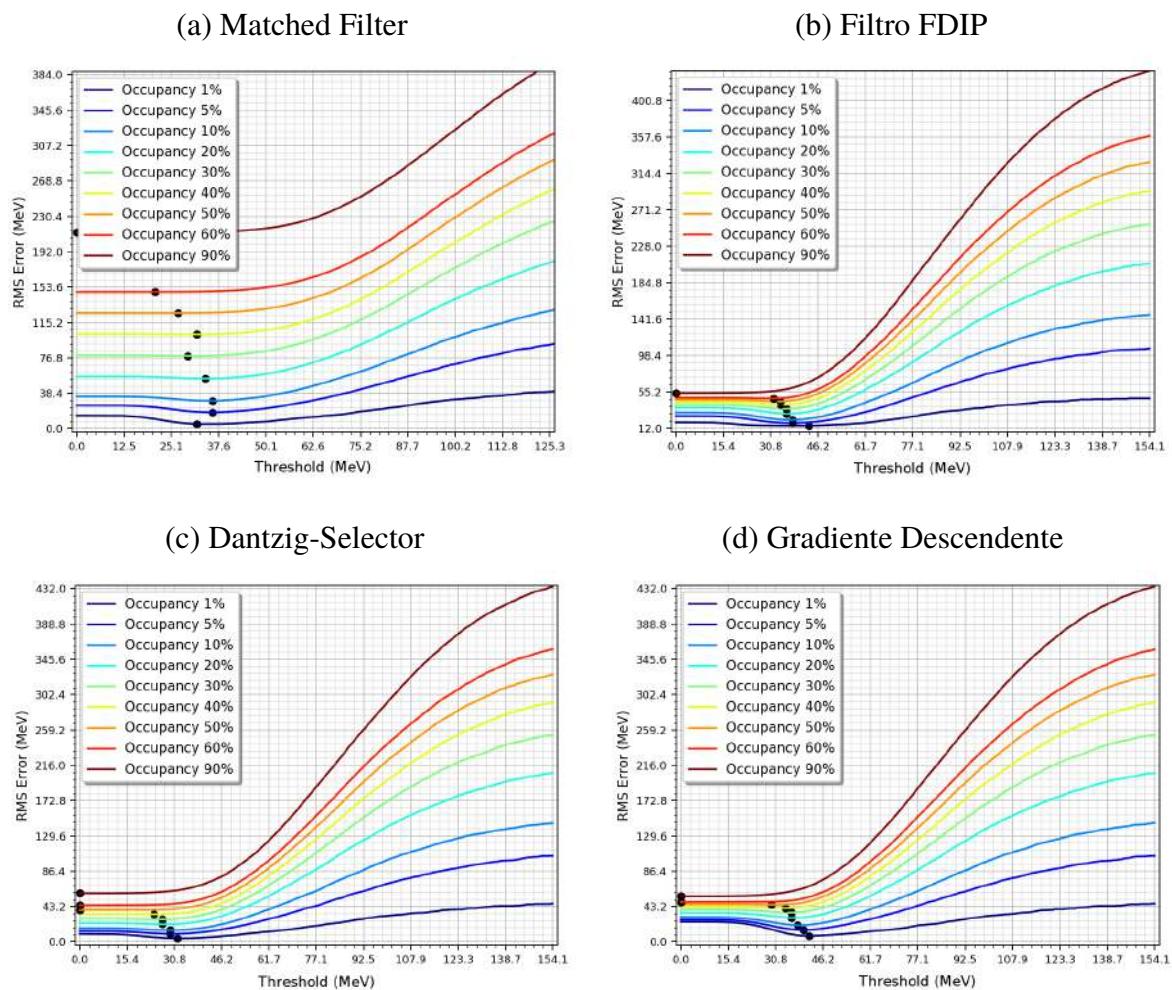
Fonte: Elaborada pelo autor (2021).

Gráfico 18 - Curvas ROC dos métodos Filtro Casado, Filtro FDIP, DS e GD, para taxas de ocupação de 60% e 90%.



Fonte: Elaborada pelo autor (2021).

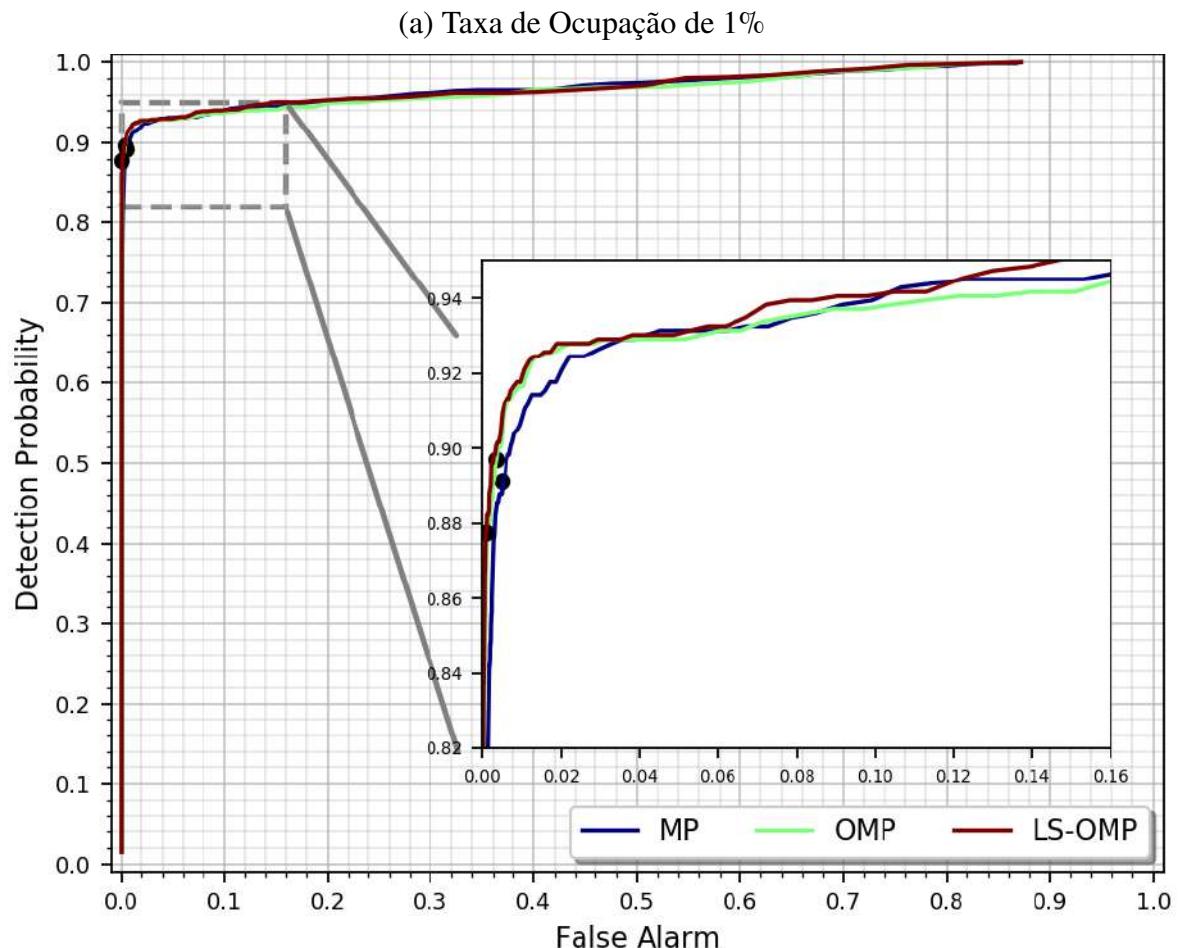
Gráfico 19 - Variação do RMSr da aplicação dos métodos Filtro Casado, Filtro FDIP, DS e GD, de acordo com as taxas de ocupação



Fonte: Elaborado pelo autor (2021).

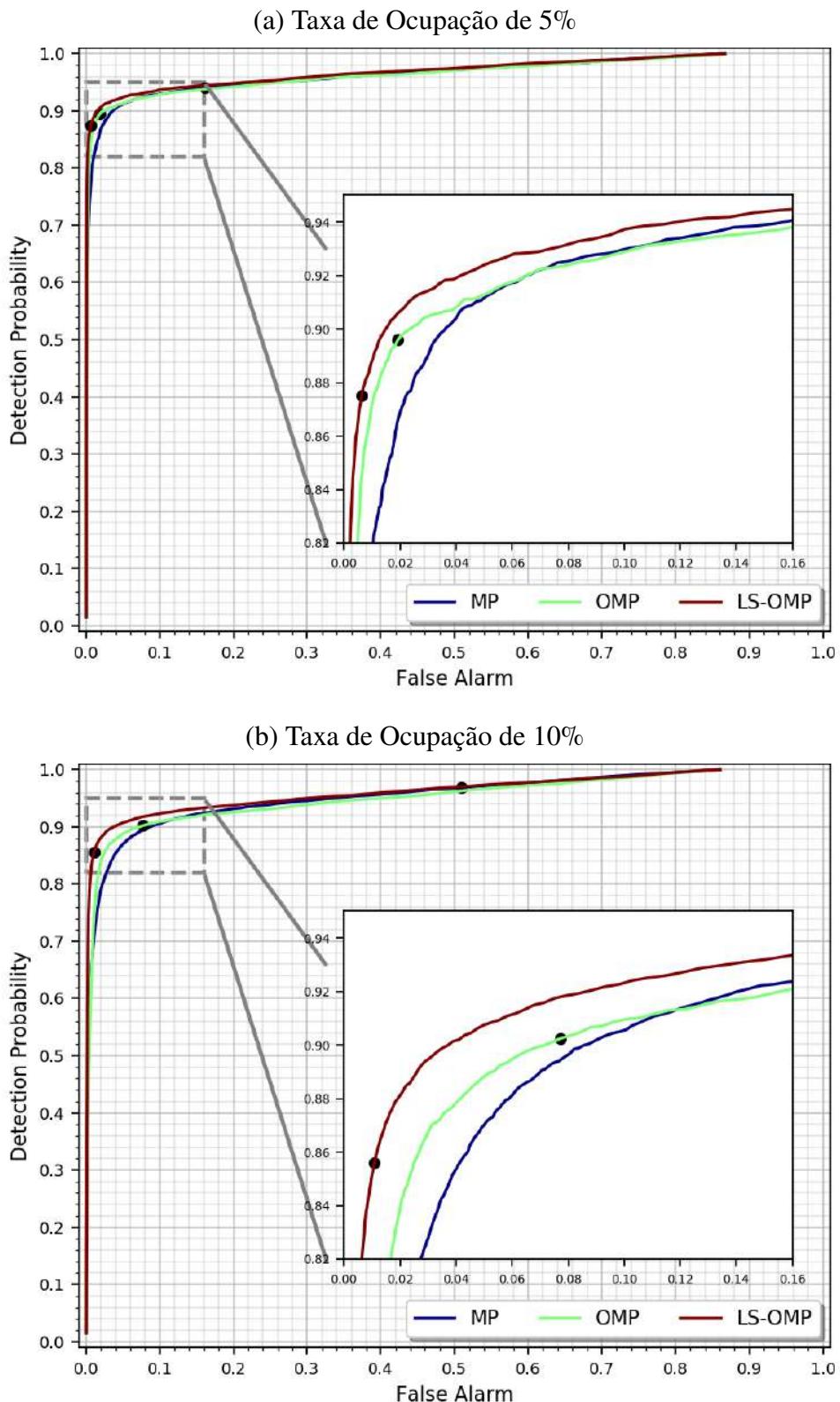
D.2 Detalhamento Gráfico dos Métodos da Família Greedy

Gráfico 20 - Curvas ROC dos métodos da família *Greedy*, para taxa de ocupação de 1%



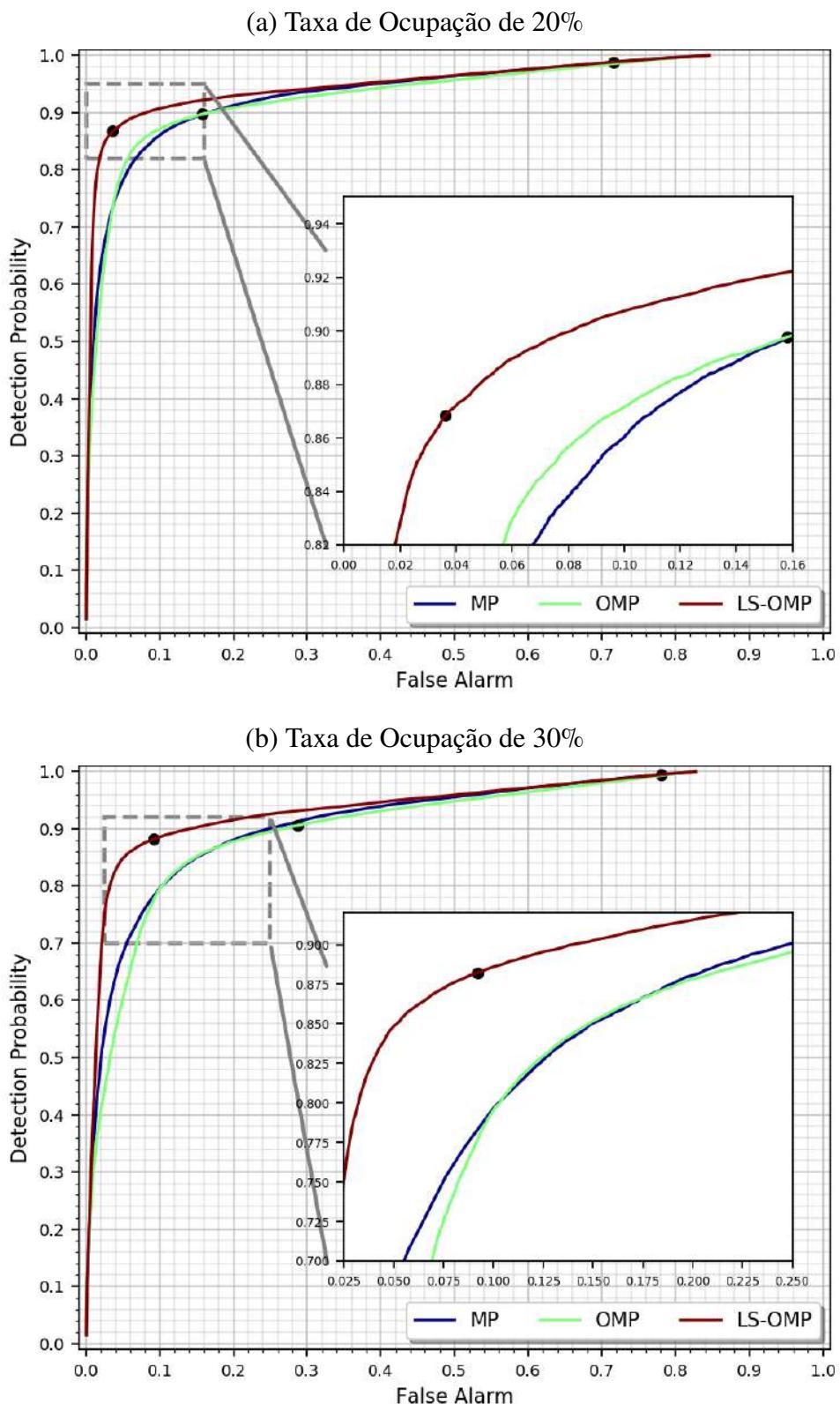
Fonte: Elaborada pelo autor (2021).

Gráfico 21 - Curvas ROC dos métodos da família *Greedy*, para taxas de ocupação de 5% e 10%



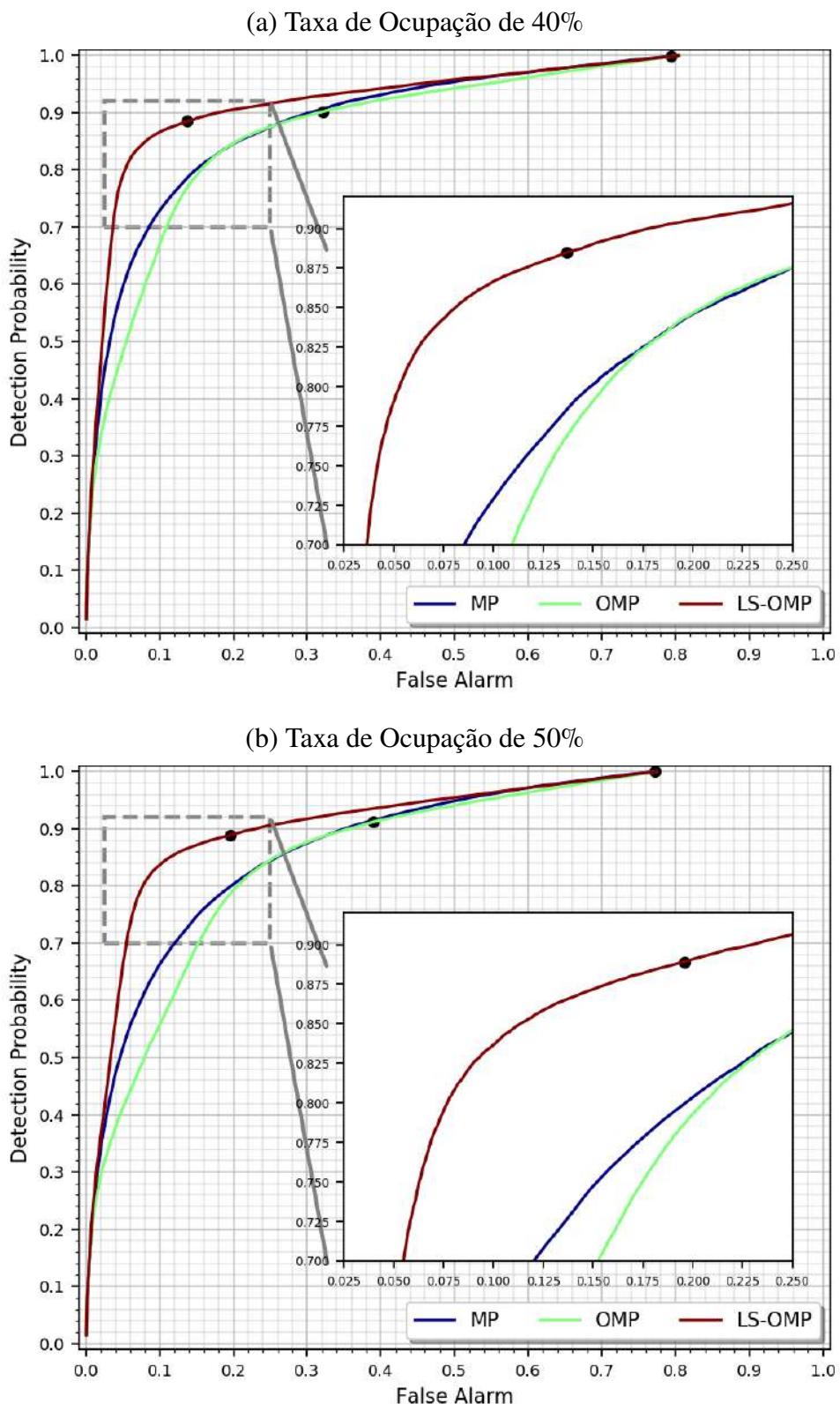
Fonte: Elaborada pelo autor (2021).

Gráfico 22 - Curvas ROC dos métodos da família *Greedy*, para taxas de ocupação de 20% e 30%



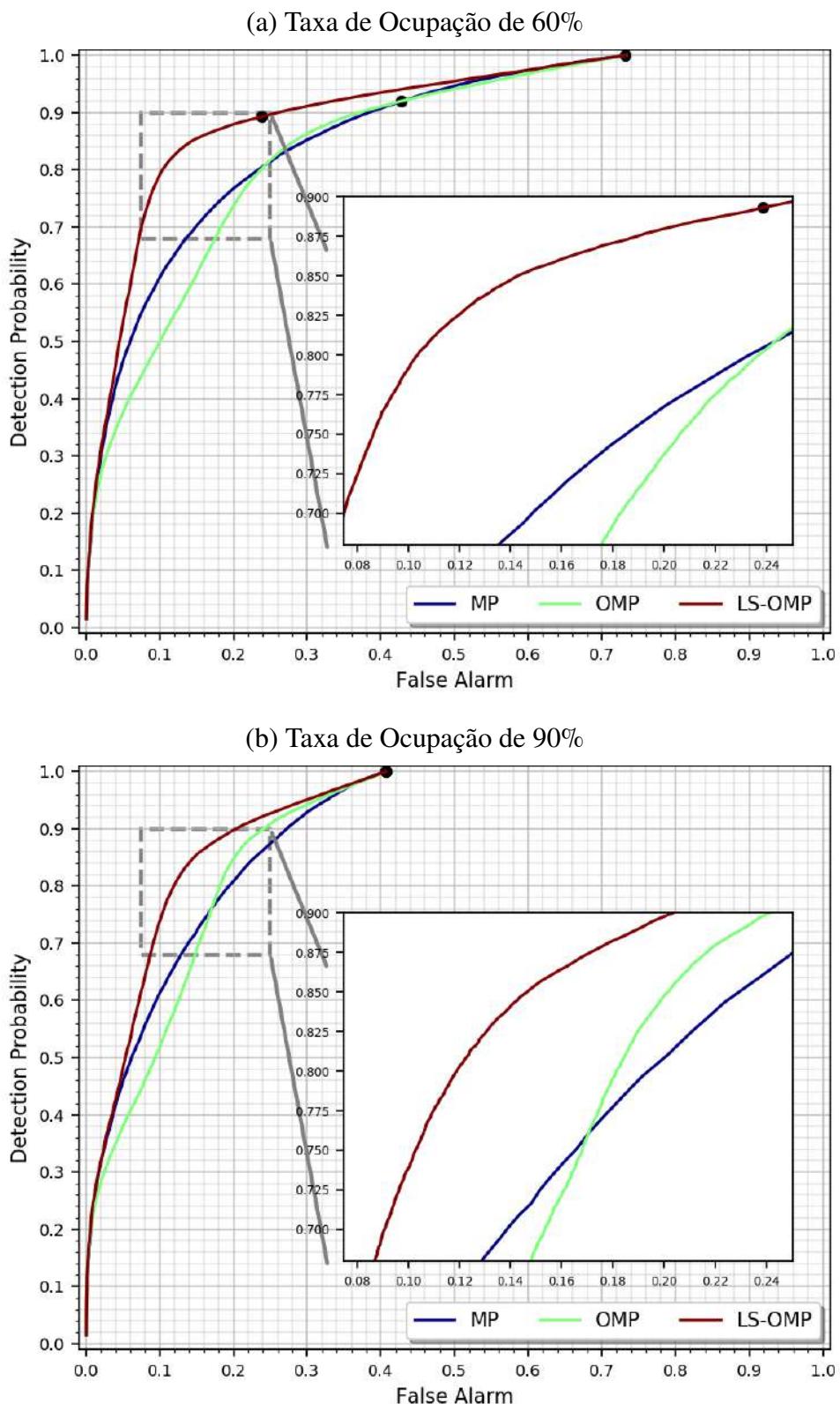
Fonte: Elaborada pelo autor (2021).

Gráfico 23 - Curvas ROC dos métodos da família *Greedy*, para taxas de ocupação de 40% e 50%



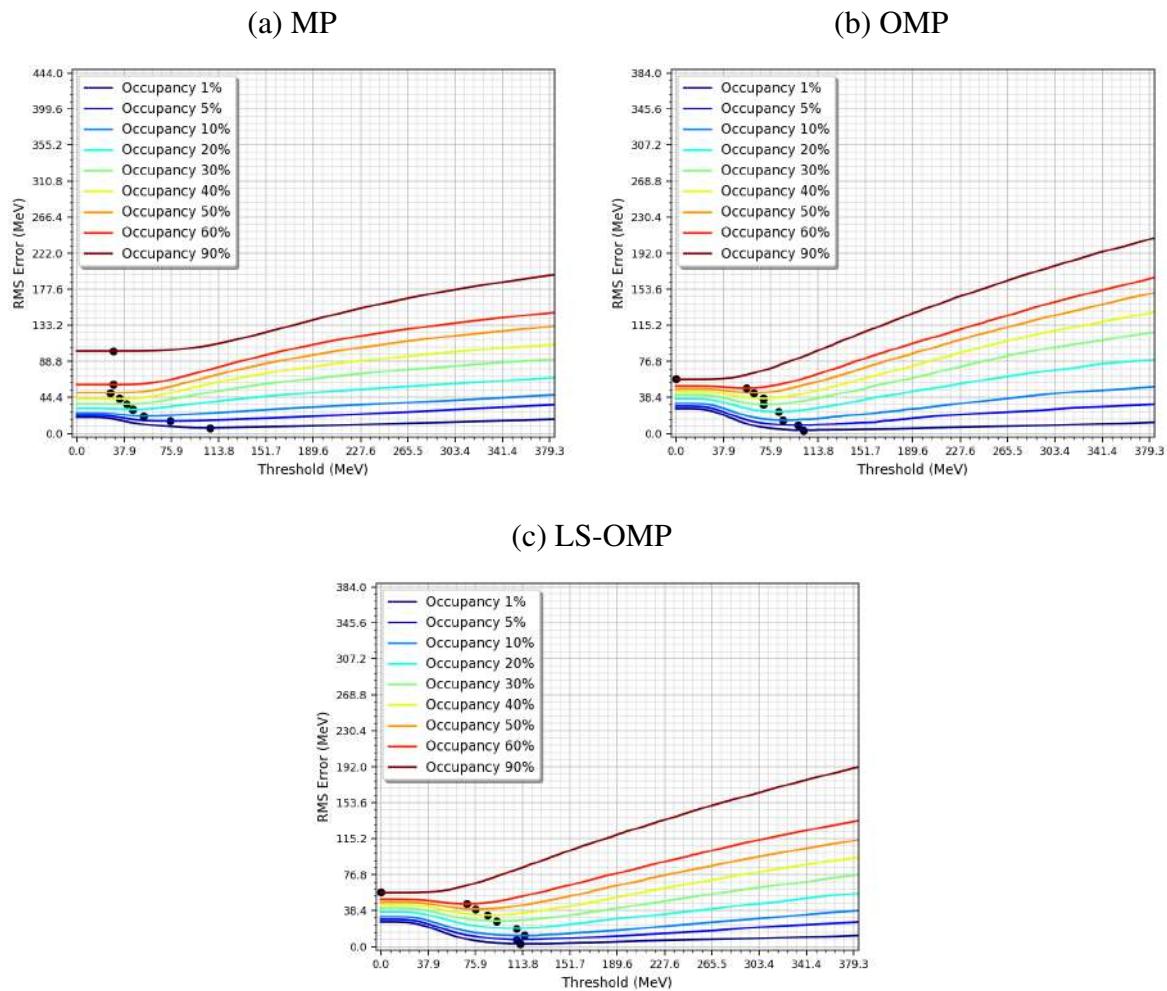
Fonte: Elaborada pelo autor (2021).

Gráfico 24 - Curvas ROC dos métodos da família *Greedy*, para taxas de ocupação de 60% e 90%



Fonte: Elaborada pelo autor (2021).

Gráfico 25 - Variação do RMSr obtido da aplicação dos métodos *Greedy*, de acordo com as taxas de ocupação



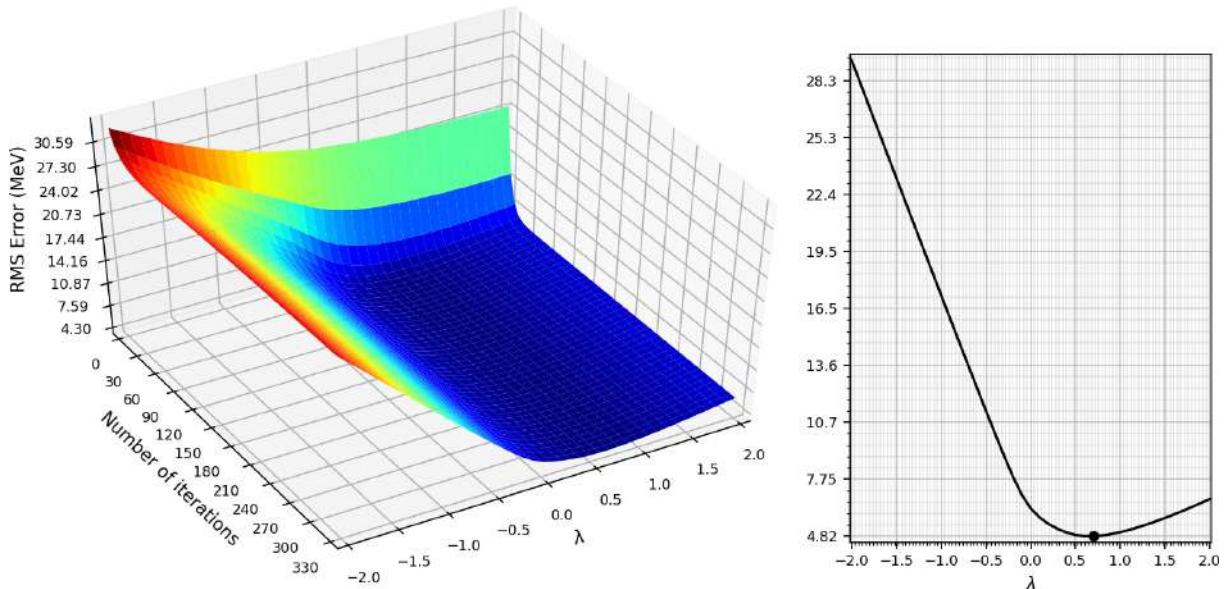
Fonte: Elaborado pelo autor (2021).

D.3 Detalhamento Gráfico dos Métodos da Família Shrinkage

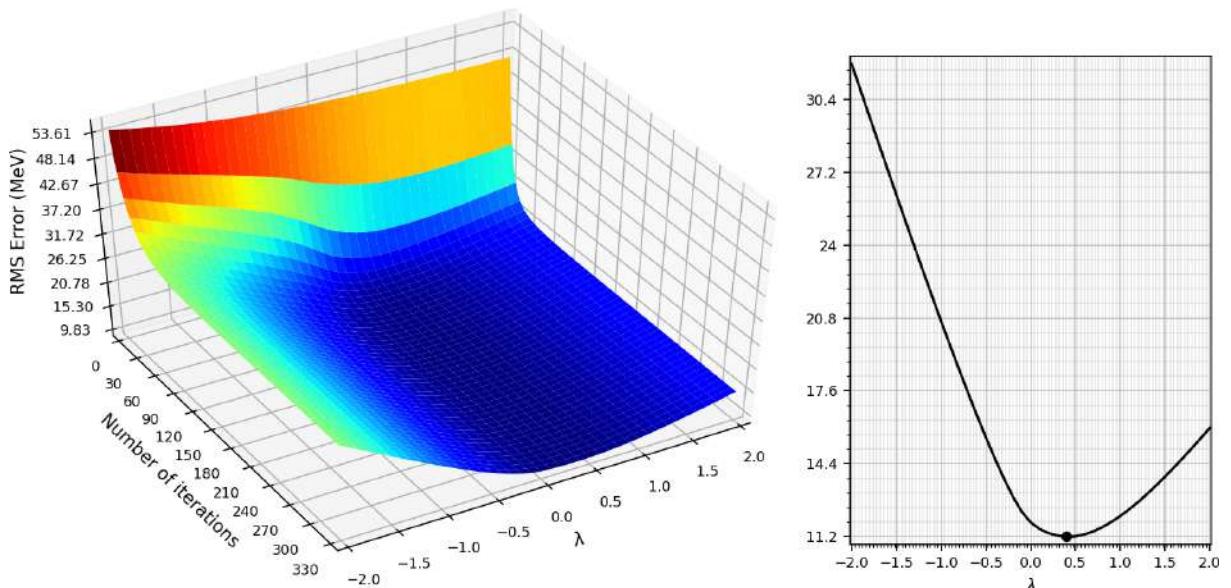
D.3.1 Valores de RMSr de λ

Gráfico 26 - Valores de RMSr, obtidos com a variação dos valores de λ para o método SSF, com taxas de ocupação de 1% e 5%

(a) Taxa de Ocupação de 1%



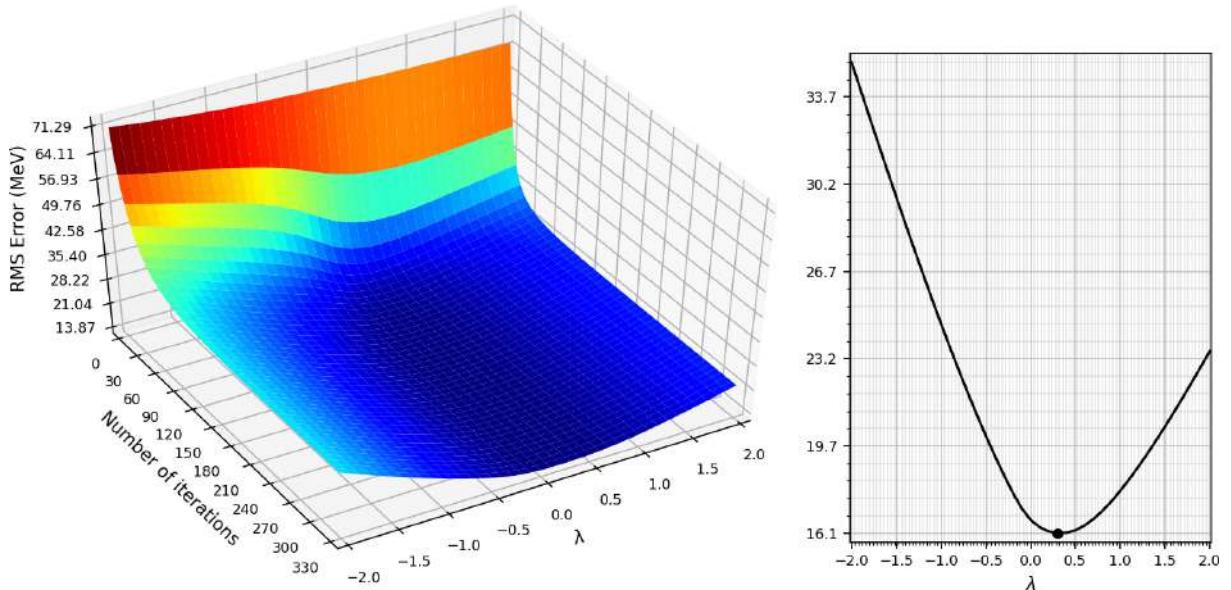
(b) Taxa de Ocupação de 5%



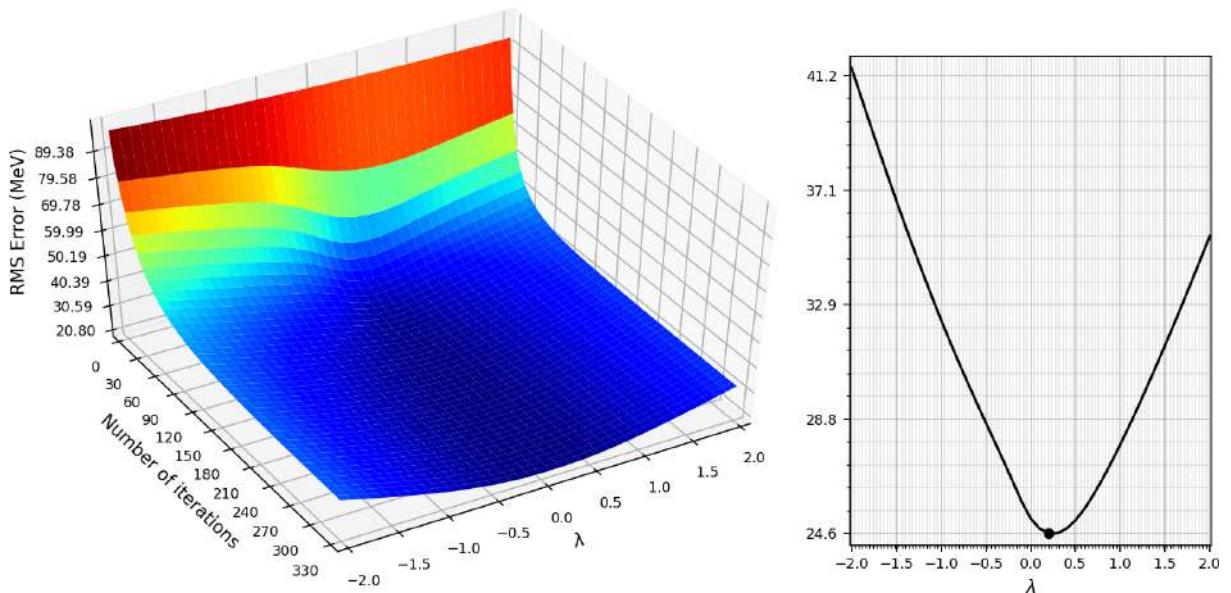
Fonte: Elaborada pelo autor (2021).

Gráfico 27 - Valores de RMSr, obtidos com a variação dos valores de λ para o método SSF, com taxas de ocupação de 10% e 20%.

(a) Taxa de Ocupação de 10%



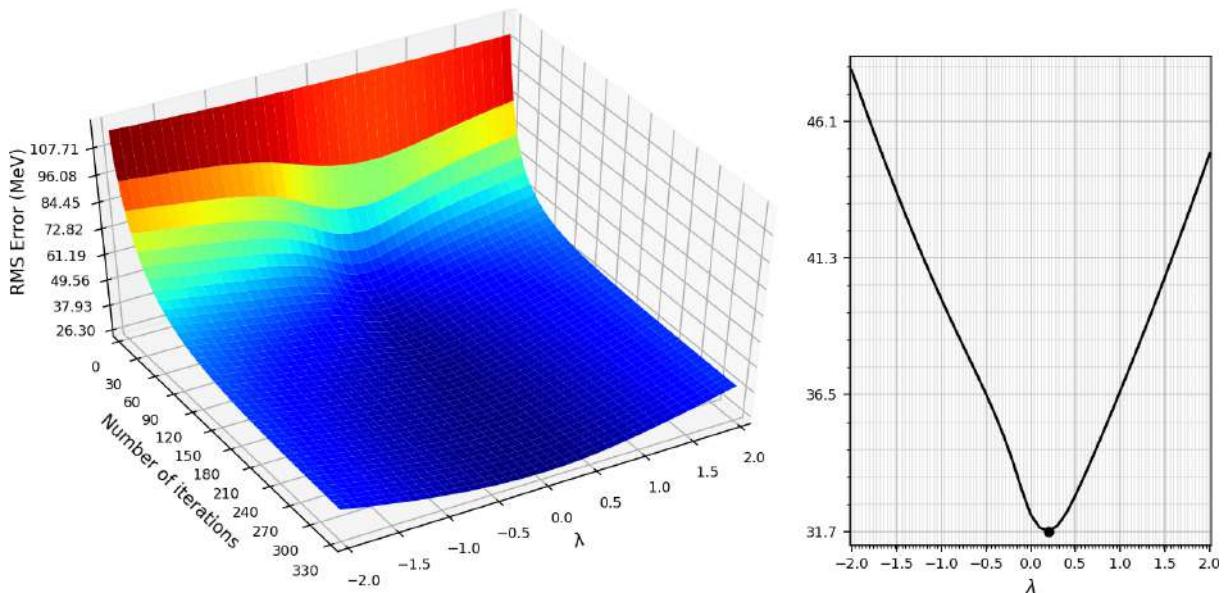
(b) Taxa de Ocupação de 20%



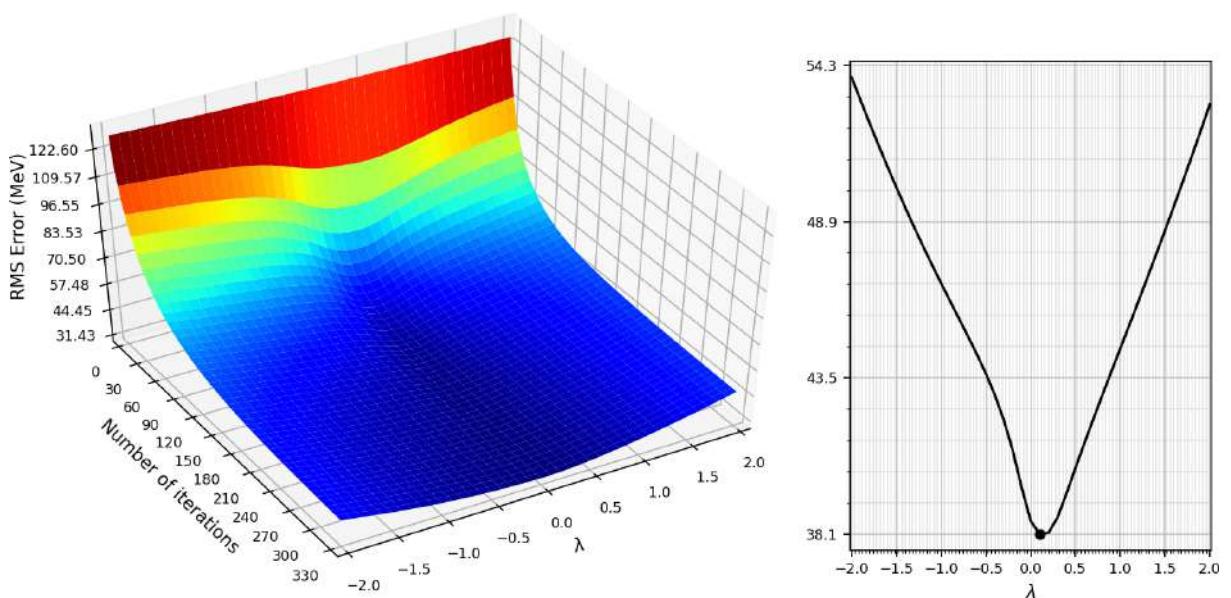
Fonte: Elaborada pelo autor (2021).

Gráfico 28 - Valores de RMSr, obtidos com a variação dos valores de λ para o método SSF, com taxas de ocupação de 30% e 40%.

(a) Taxa de Ocupação de 30%



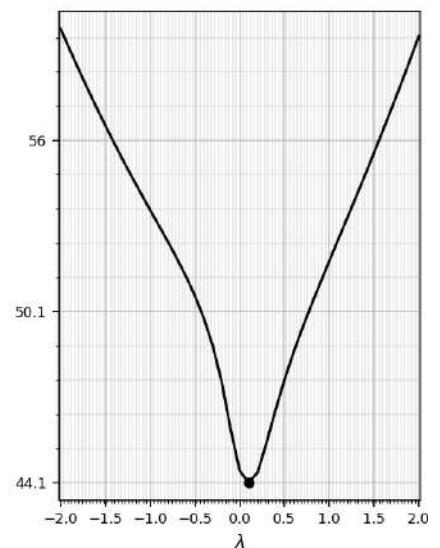
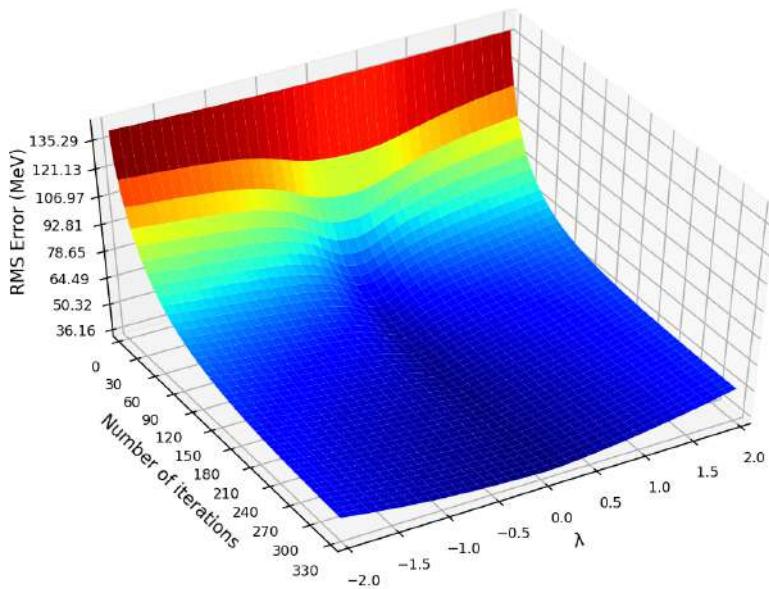
(b) Taxa de Ocupação de 40%



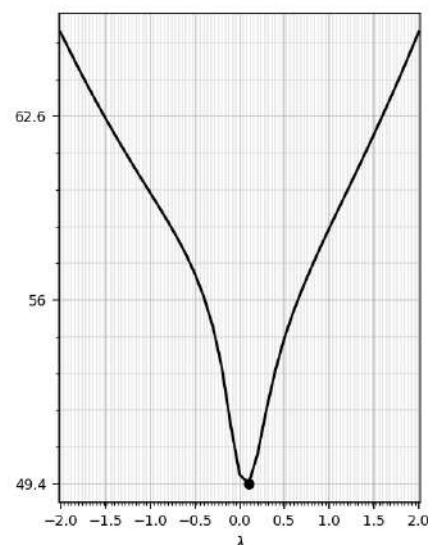
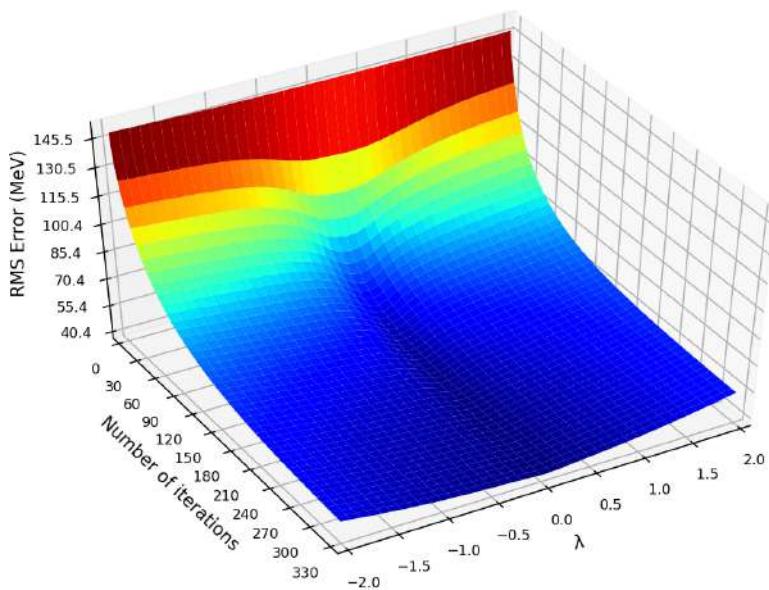
Fonte: Elaborada pelo autor (2021).

Gráfico 29 - Valores de RMSr, obtidos com a variação dos valores de λ para o método SSF, com taxas de ocupação de 50% e 60%

(a) Taxa de Ocupação de 50%

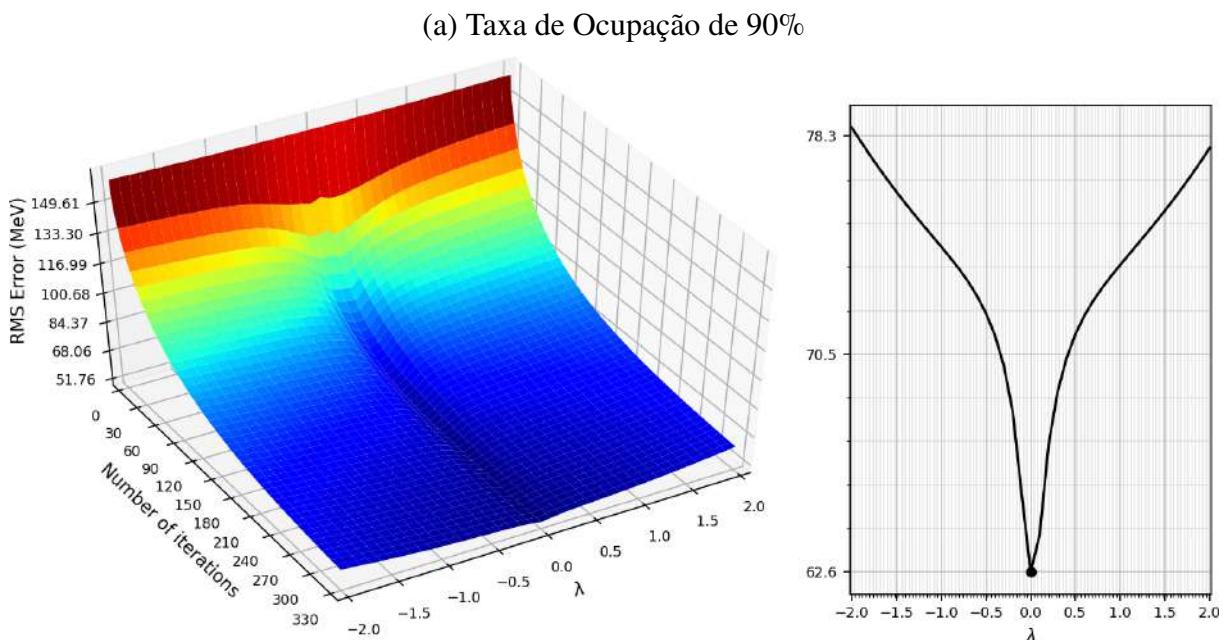


(b) Taxa de Ocupação de 60%



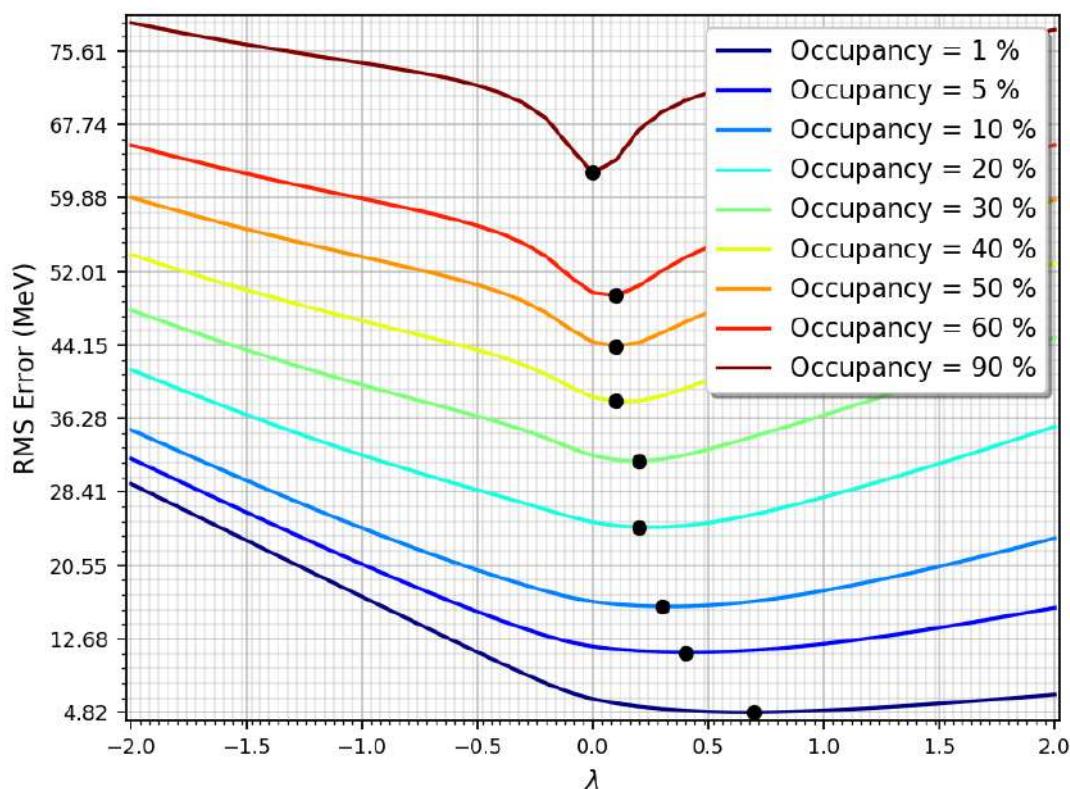
Fonte: Elaborada pelo autor (2021).

Gráfico 30 - Valores de RMSr, obtidos com a variação dos valores de λ para o método SSF, com taxa de ocupação de 90%



Fonte: Elaborada pelo autor (2021).

Gráfico 31 - Valores médios de RMSr obtidos com a variação das opções de λ para o método SSF, de acordo com as taxas de ocupação

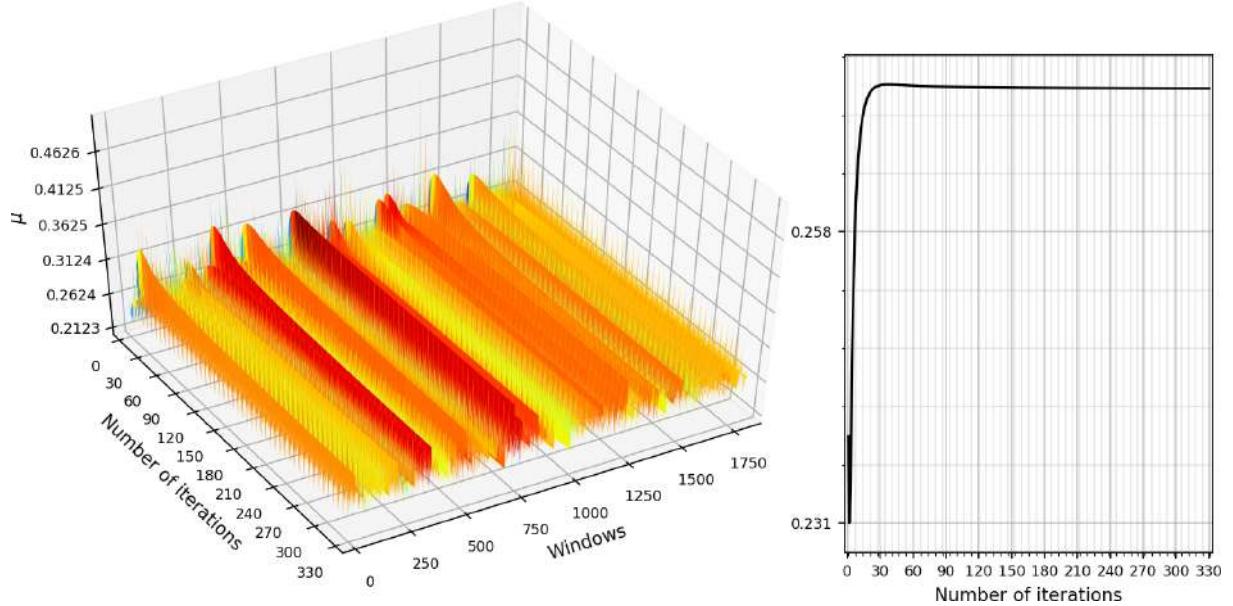


Fonte: Elaborado pelo autor (2021).

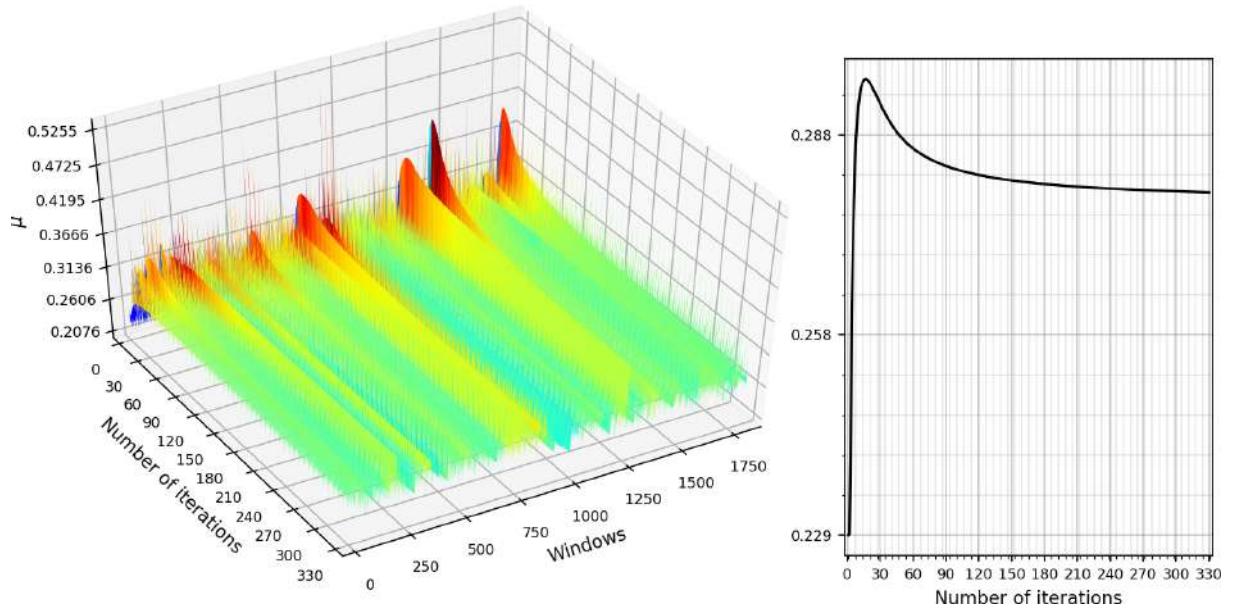
D.3.2 Valores de μ em 3D

Gráfico 32 - Valores de μ para o método SSF, com taxas de ocupação de 1% e 5%

(a) Taxa de Ocupação de 1%



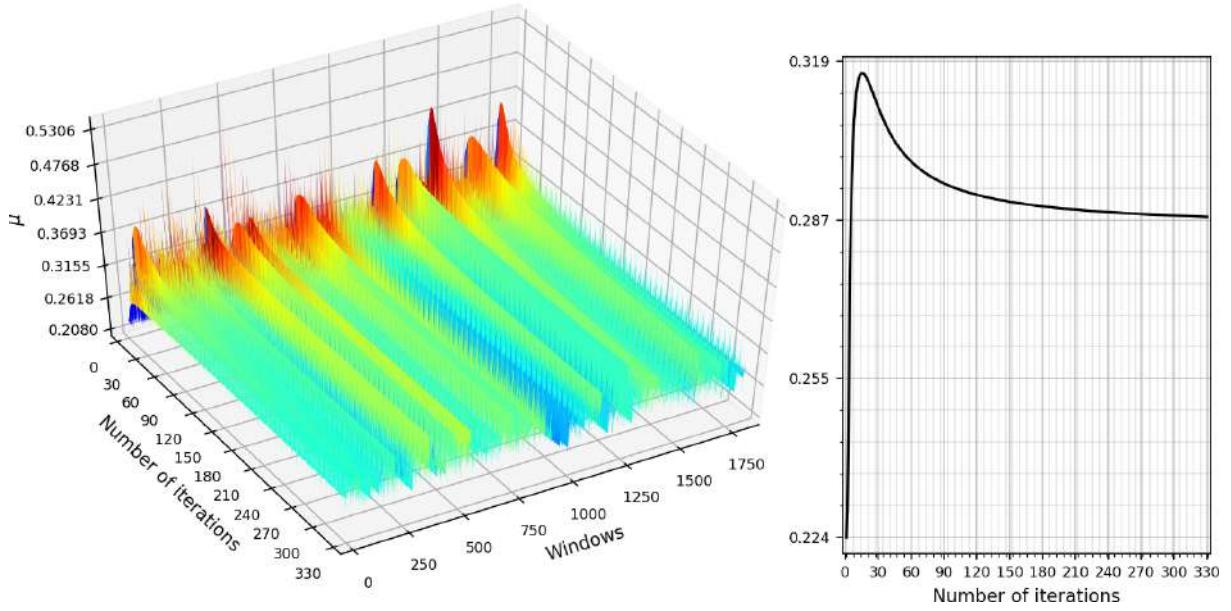
(b) Taxa de Ocupação de 5%



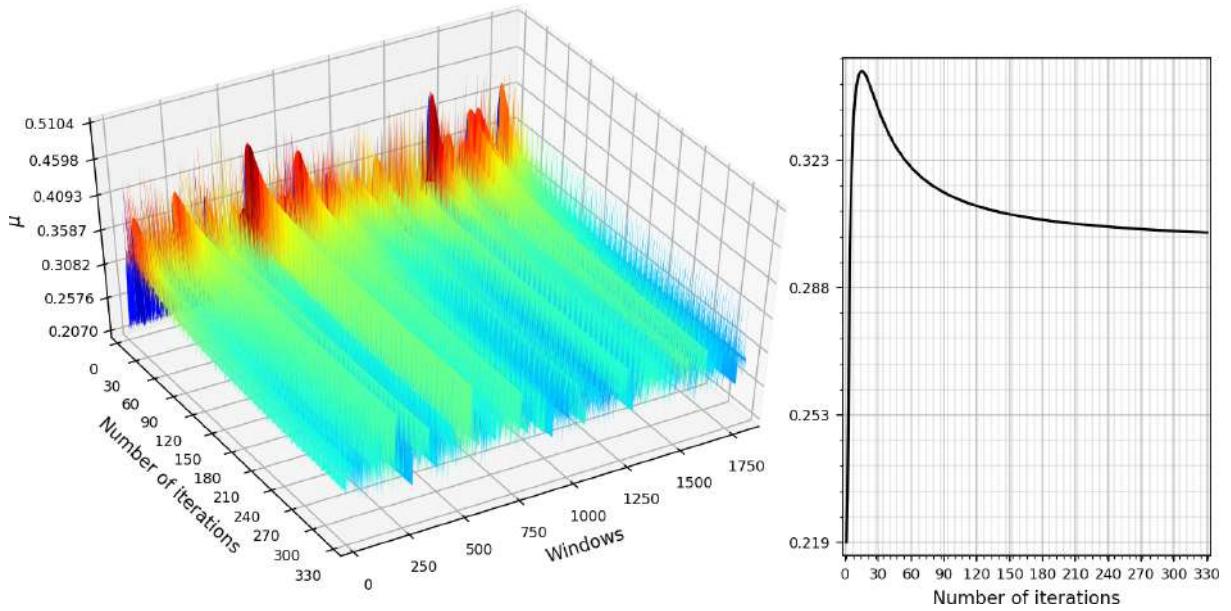
Fonte: Elaborada pelo autor (2021).

Gráfico 33 - Valores de μ para o método SSF, com taxas de ocupação de 10% e 20%

(a) Taxa de Ocupação de 10%



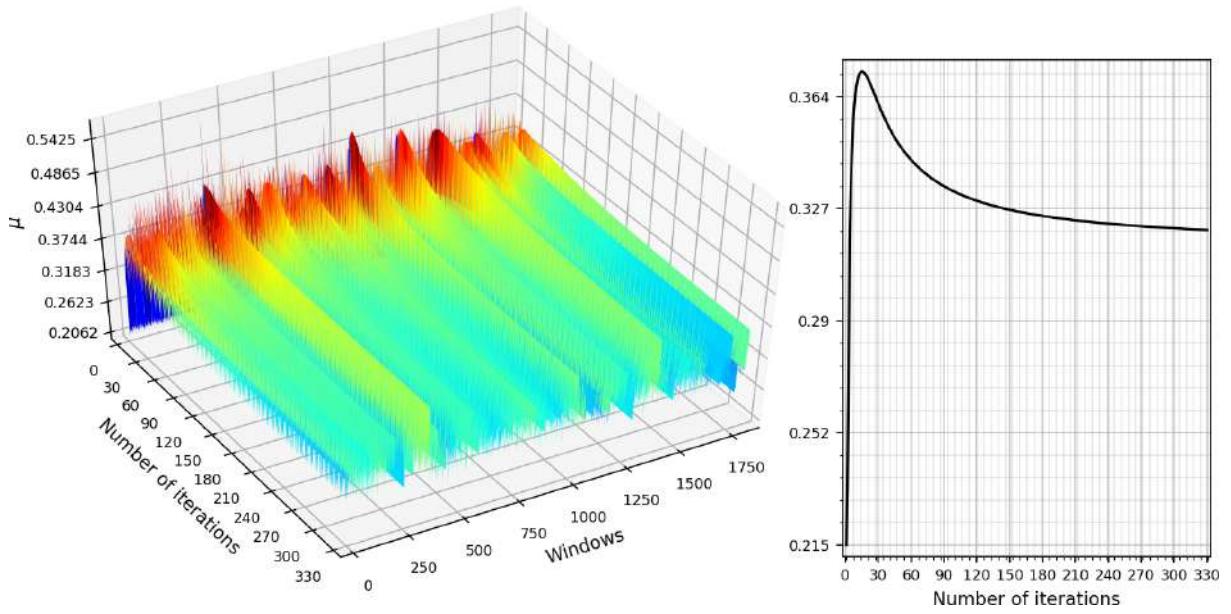
(b) Taxa de Ocupação de 20%



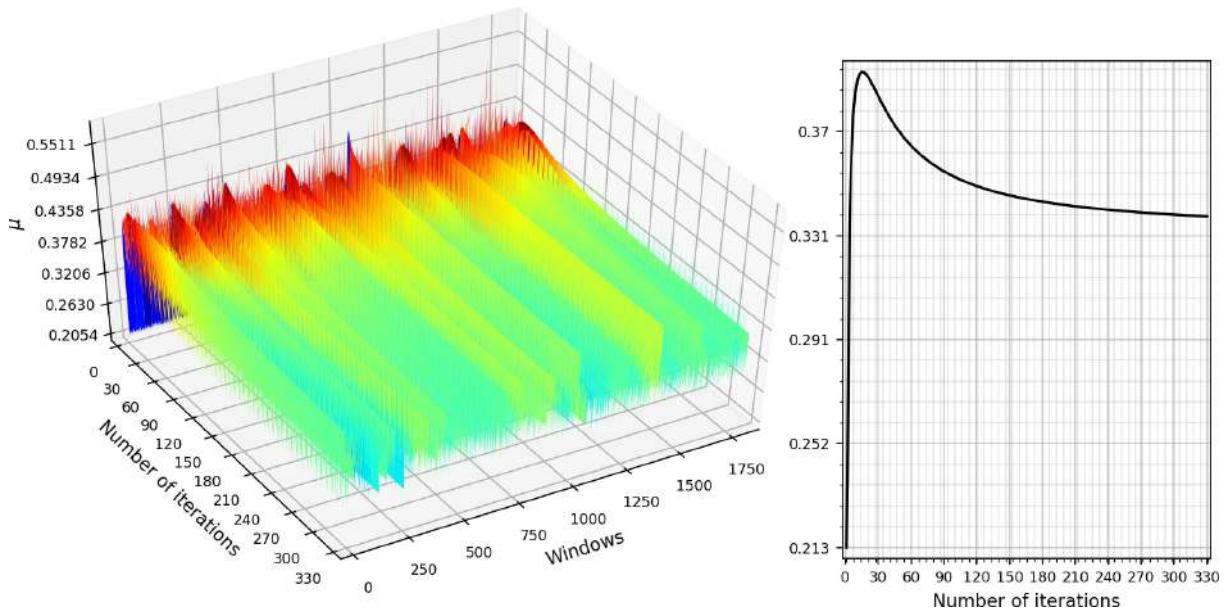
Fonte: Elaborada pelo autor (2021).

Gráfico 34 - Valores de μ para o método SSF, com taxas de ocupação de 30% e 40%

(a) Taxa de Ocupação de 30%



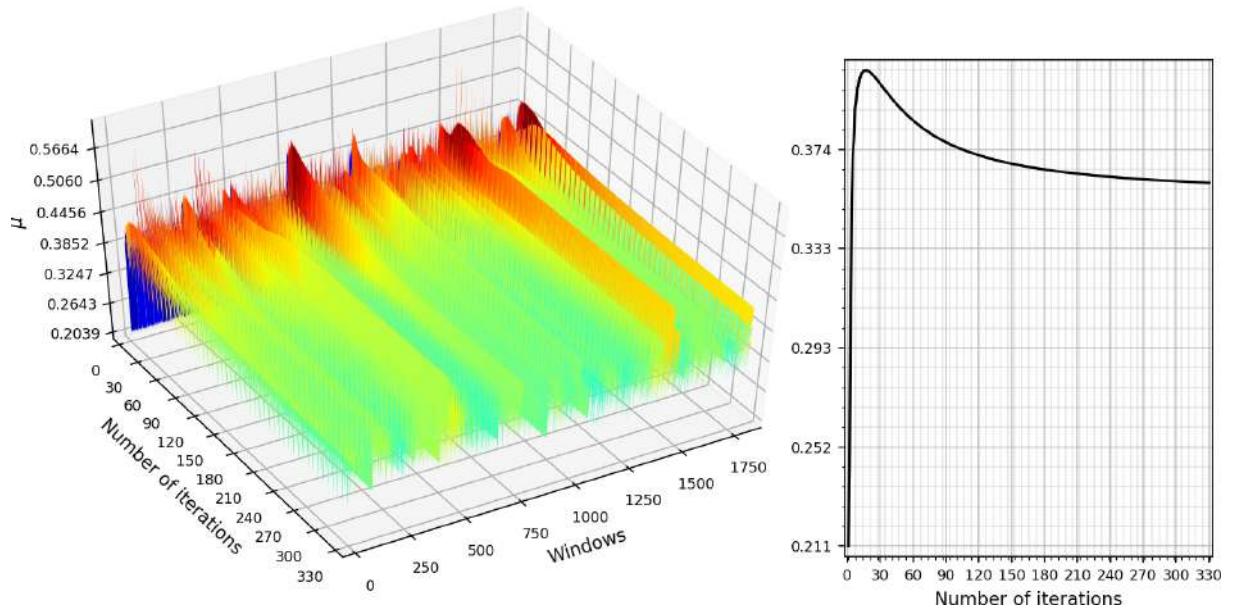
(b) Taxa de Ocupação de 40%



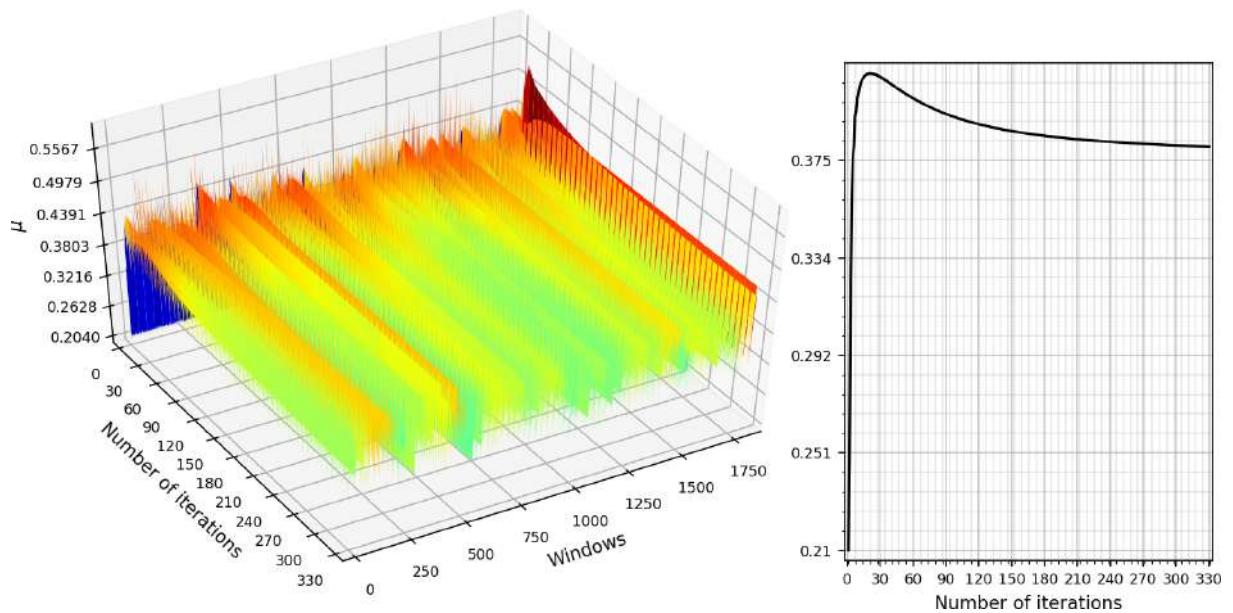
Fonte: Elaborada pelo autor (2021).

Gráfico 35 - Valores de μ para o método SSF, com taxas de ocupação de 50% e 60%.

(a) Taxa de Ocupação de 50%



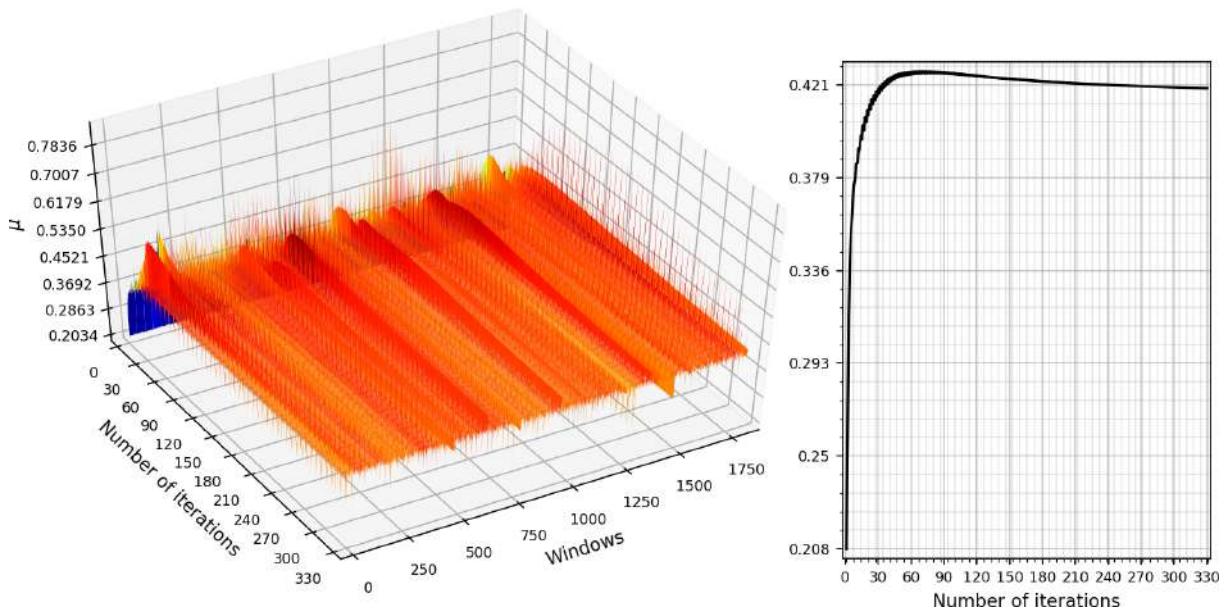
(b) Taxa de Ocupação de 60%



Fonte: Elaborada pelo autor (2021).

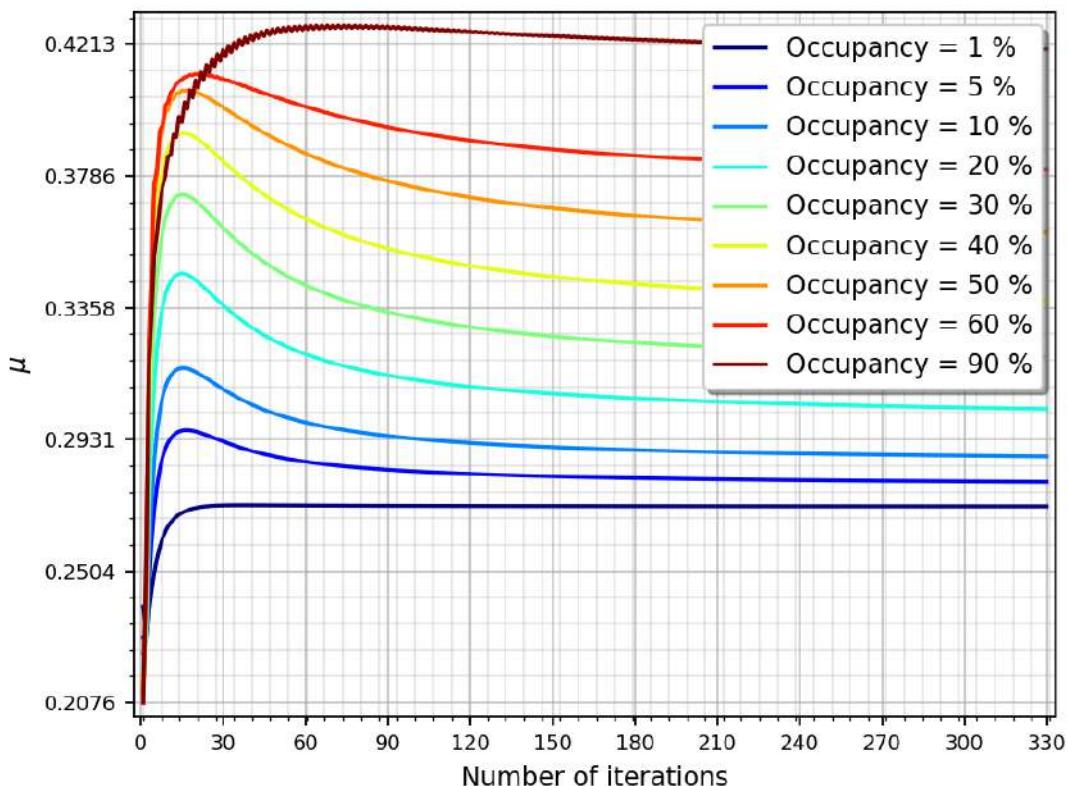
Gráfico 36 - Valores de μ para o método SSF, com taxa de ocupação de 90%

(a) Taxa de Ocupação de 90%



Fonte: Elaborada pelo autor (2021).

Gráfico 37 - Valores médios de μ obtidos para o método SSF, de acordo com as taxas de ocupação

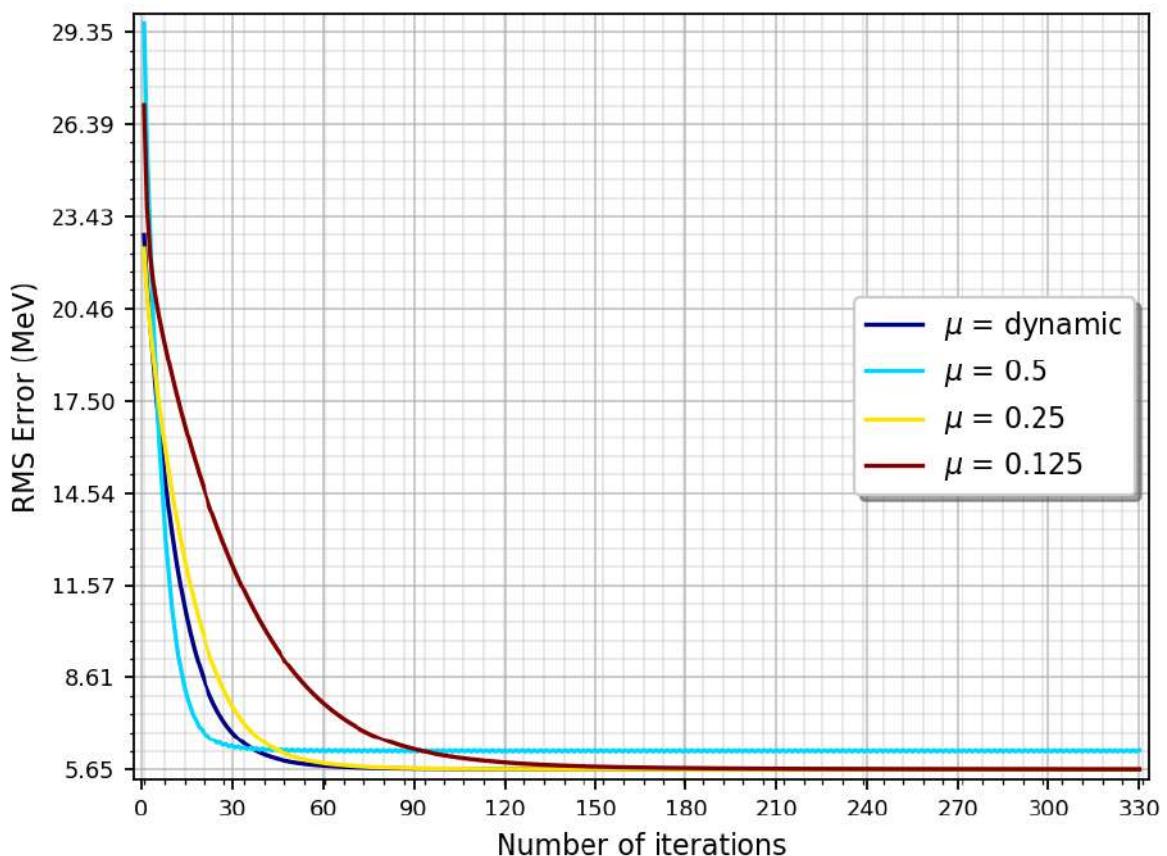


Fonte: Elaborado pelo autor (2021).

D.3.3 Valores de μ médios em 2D

Gráfico 38 - Valores de RMSr obtidos com as variações de μ , para o método SSF, com taxa de ocupação de 1%

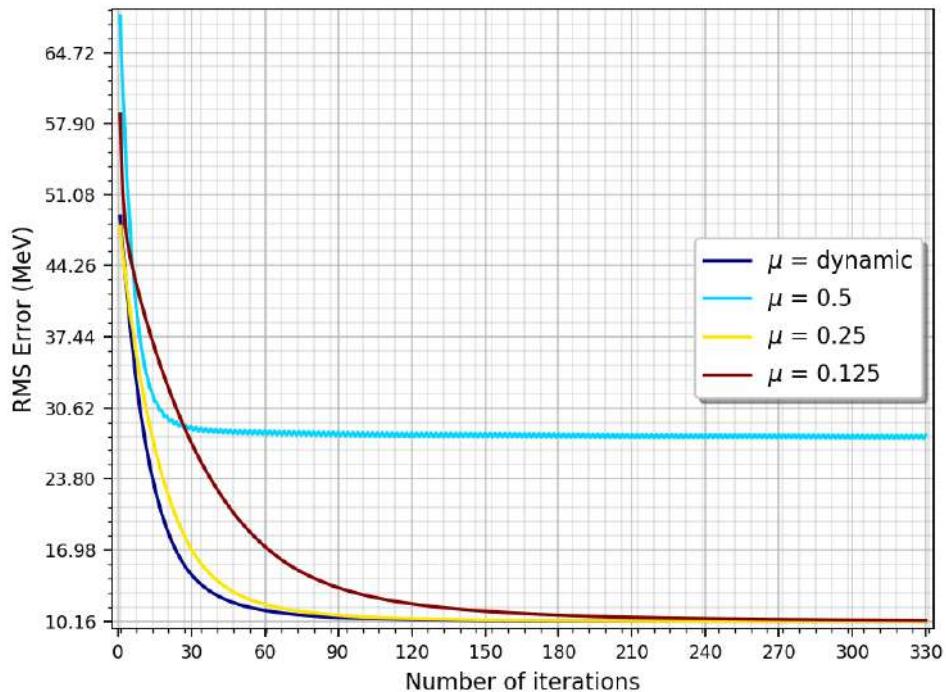
(a) Taxa de Ocupação de 1%



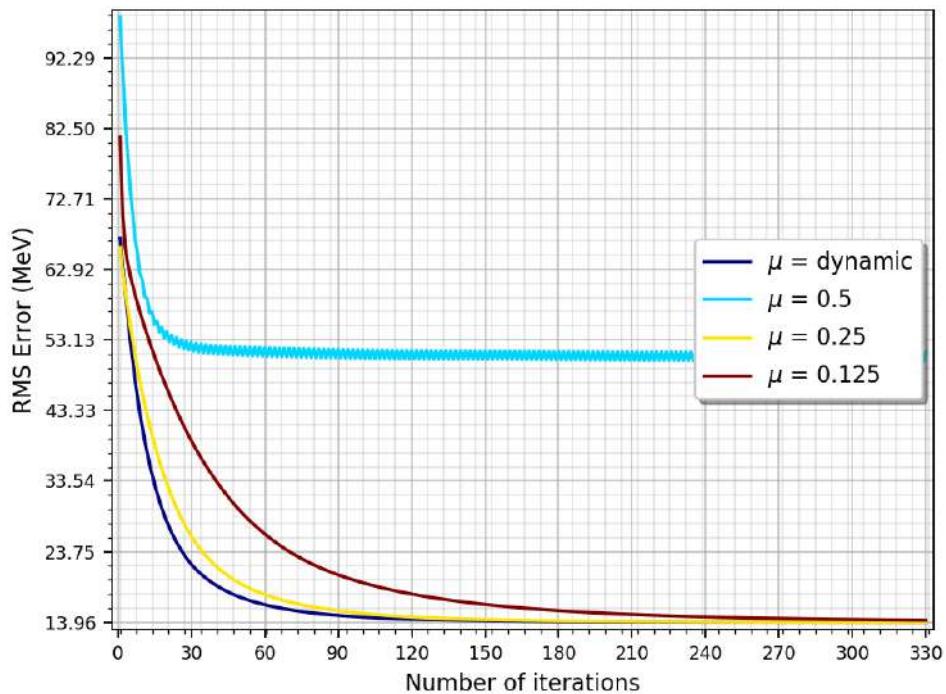
Fonte: Elaborada pelo autor (2021).

Gráfico 39 - Valores de RMSr obtidos com as variações de μ , para o método SSF, com taxas de ocupação de 5% e 10%

(a) Taxa de Ocupação de 5%



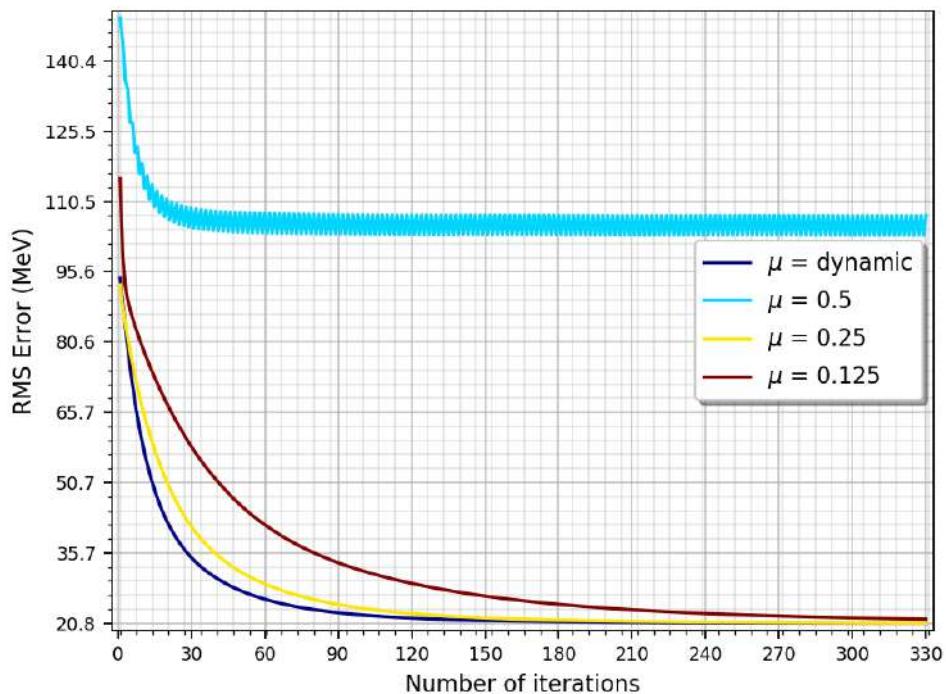
(b) Taxa de Ocupação de 10%



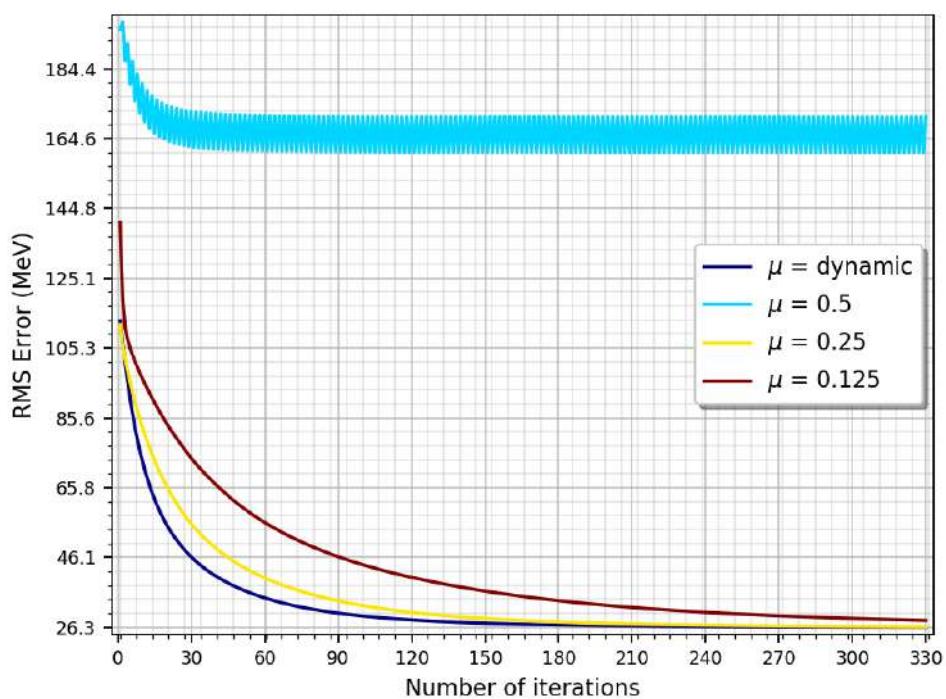
Fonte: Elaborada pelo autor (2021).

Gráfico 40 - Valores de RMSr obtidos com as variações de μ , para o método SSF, com taxas de ocupação de 20% e 30%

(a) Taxa de Ocupação de 20%



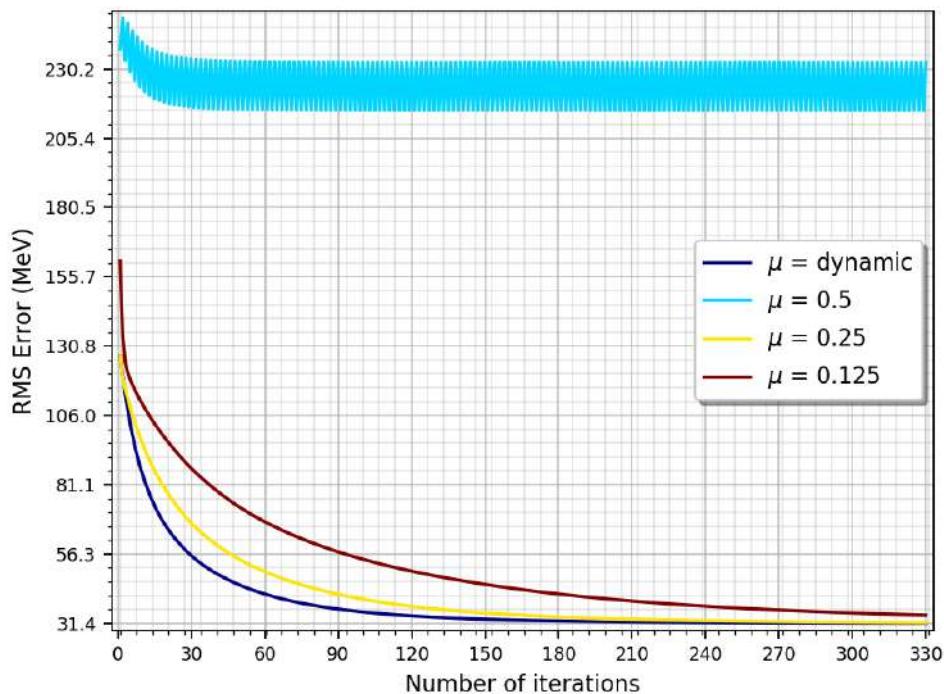
(b) Taxa de Ocupação de 30%



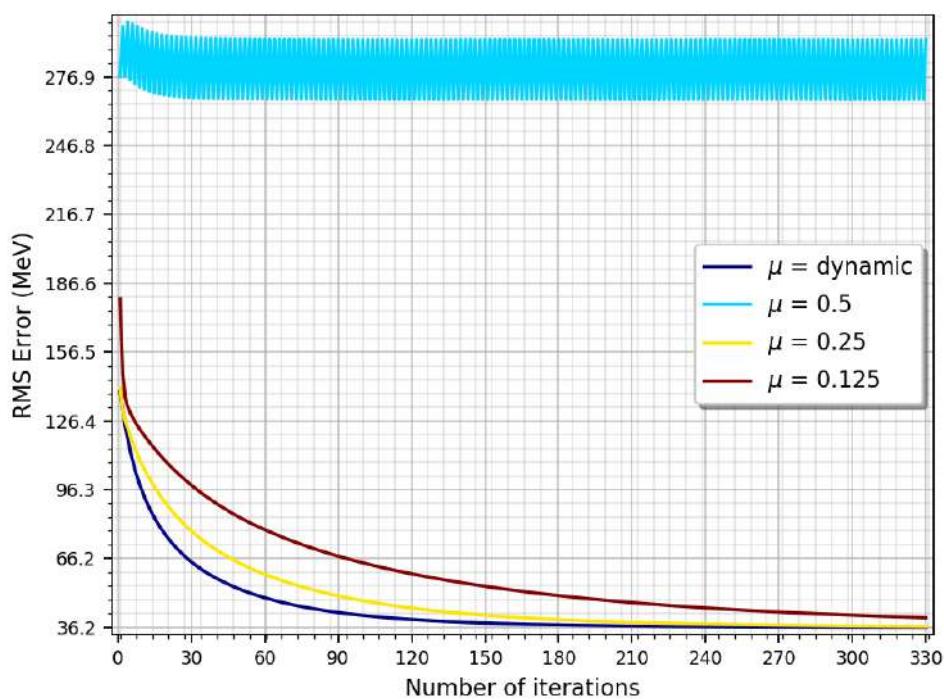
Fonte: Elaborada pelo autor (2021).

Gráfico 41 - Valores de RMSr obtidos com as variações de μ , para o método SSF, com taxas de ocupação de 40% e 50%

(a) Taxa de Ocupação de 40%



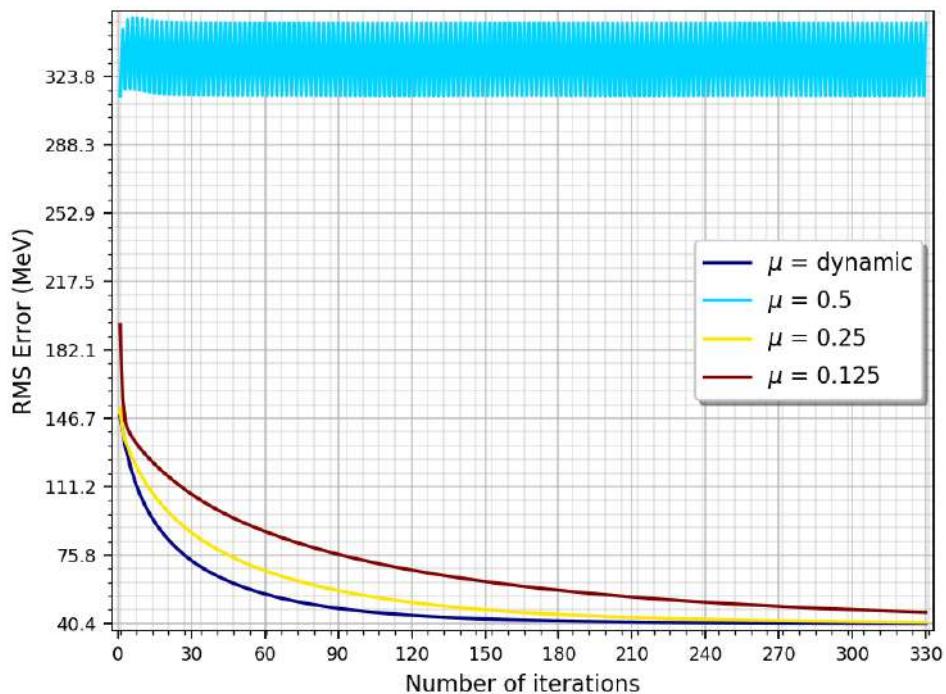
(b) Taxa de Ocupação de 50%



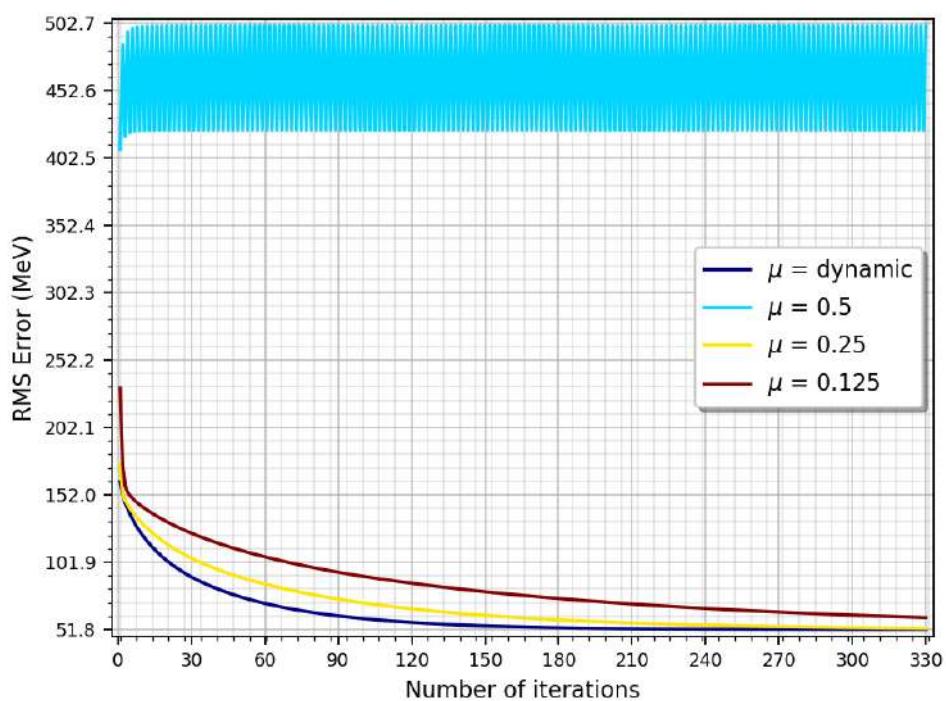
Fonte: Elaborada pelo autor (2021).

Gráfico 42 - Valores de RMSr obtidos com as variações de μ , para o método SSF, com taxas de ocupação de 60% e 90%

(a) Taxa de Ocupação de 60%



(b) Taxa de Ocupação de 90%

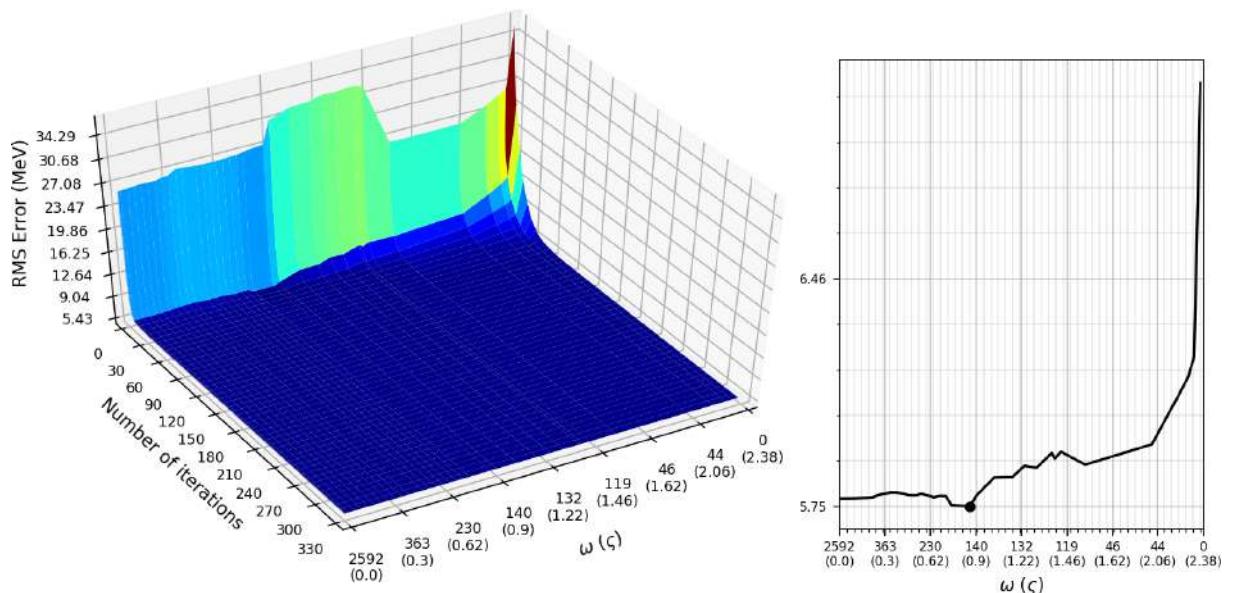


Fonte: Elaborada pelo autor (2021).

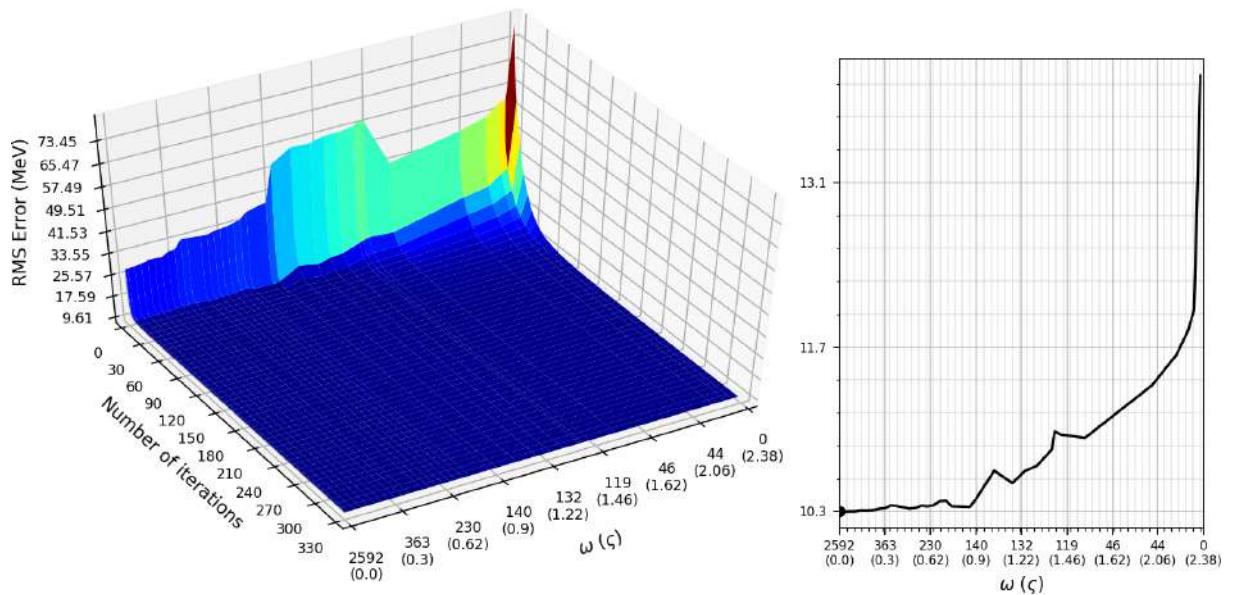
D.3.4 Valores de RMSr para variações do número de elementos da pseudo-inversa da matriz H

Gráfico 43 - Valores de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o método SSFi, com taxas de ocupação de 1% e 5%

(a) Taxa de Ocupação de 1%



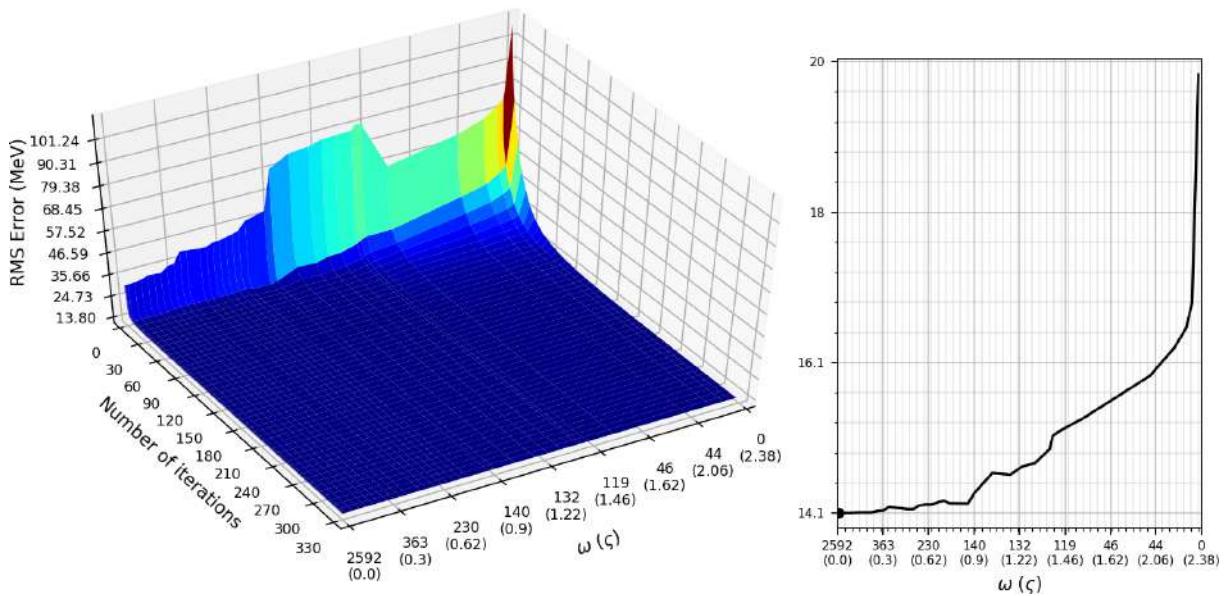
(b) Taxa de Ocupação de 5%



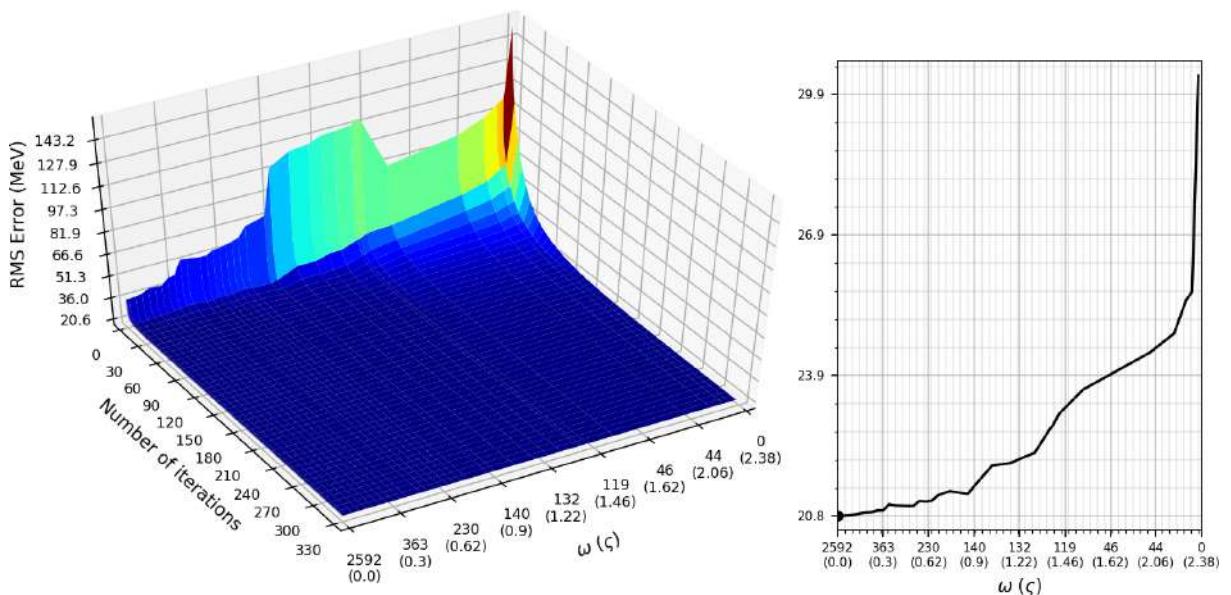
Fonte: Elaborada pelo autor (2021).

Gráfico 44 - Valores de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o método SSFi, com taxas de ocupação de 10% e 20%

(a) Taxa de Ocupação de 10%



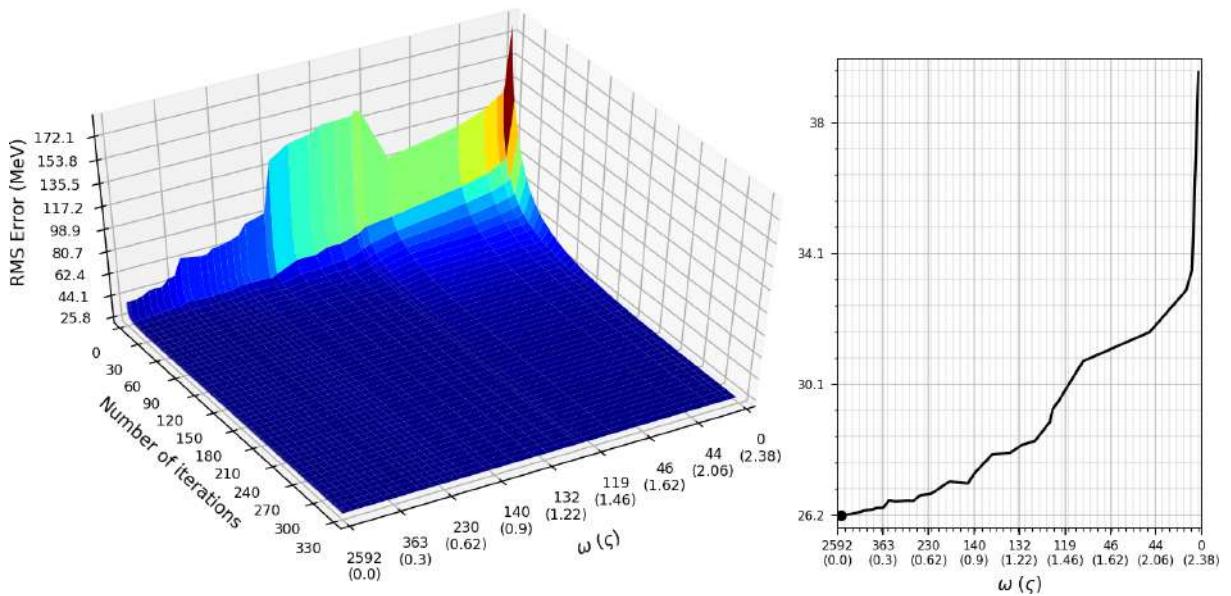
(b) Taxa de Ocupação de 20%



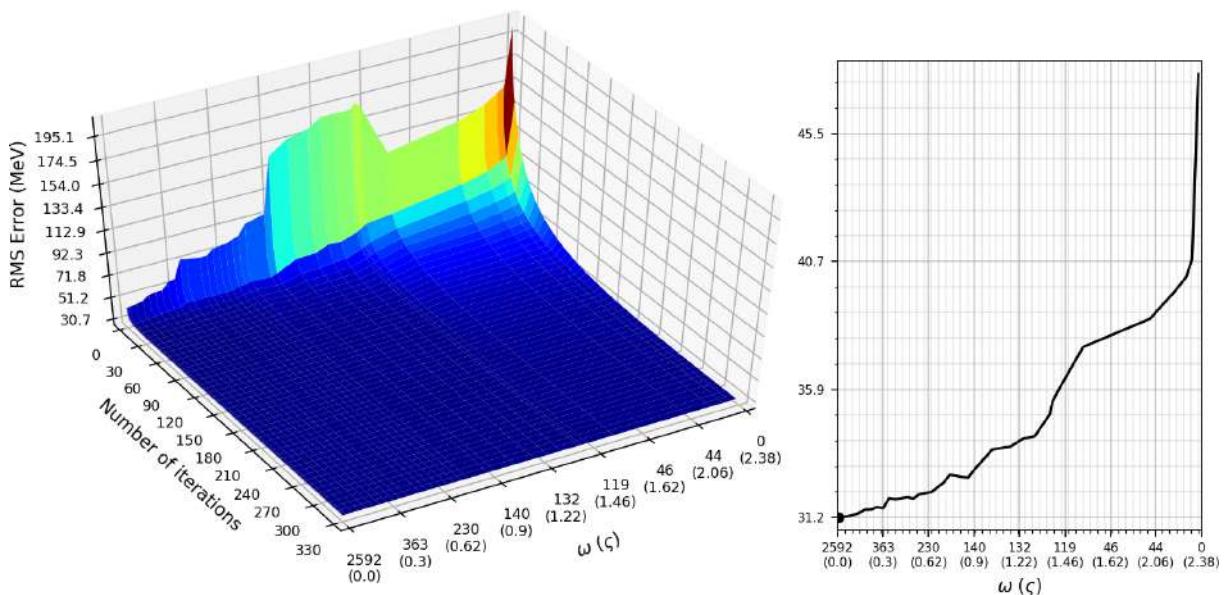
Fonte: Elaborada pelo autor (2021).

Gráfico 45 - Valores de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o método SSFi, com taxas de ocupação de 30% e 40%

(a) Taxa de Ocupação de 30%



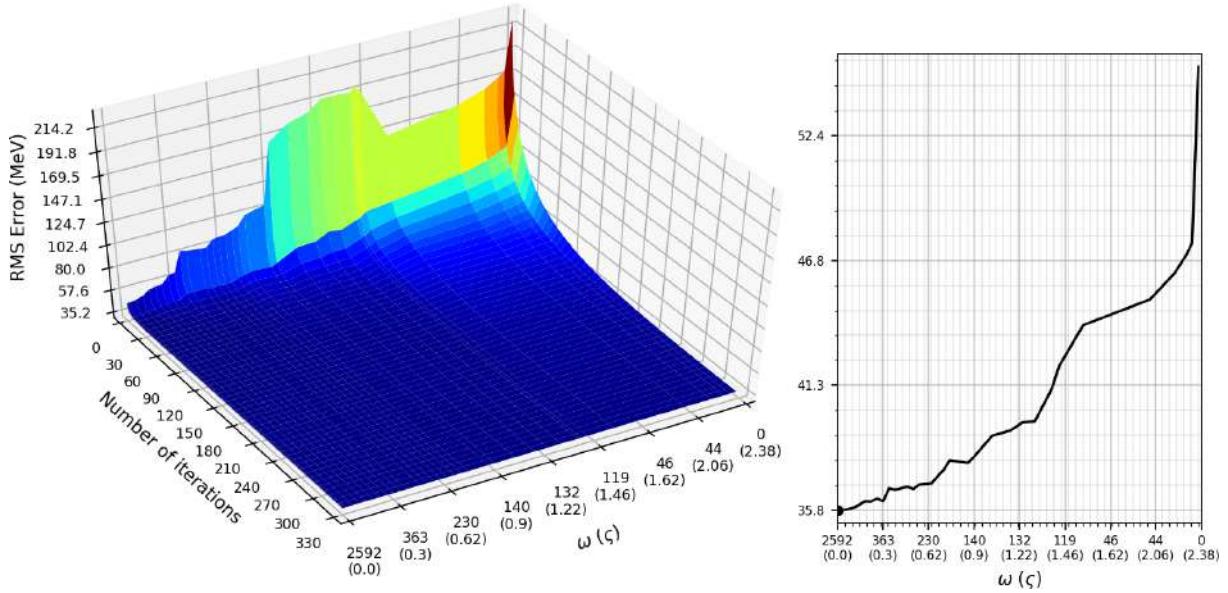
(b) Taxa de Ocupação de 40%



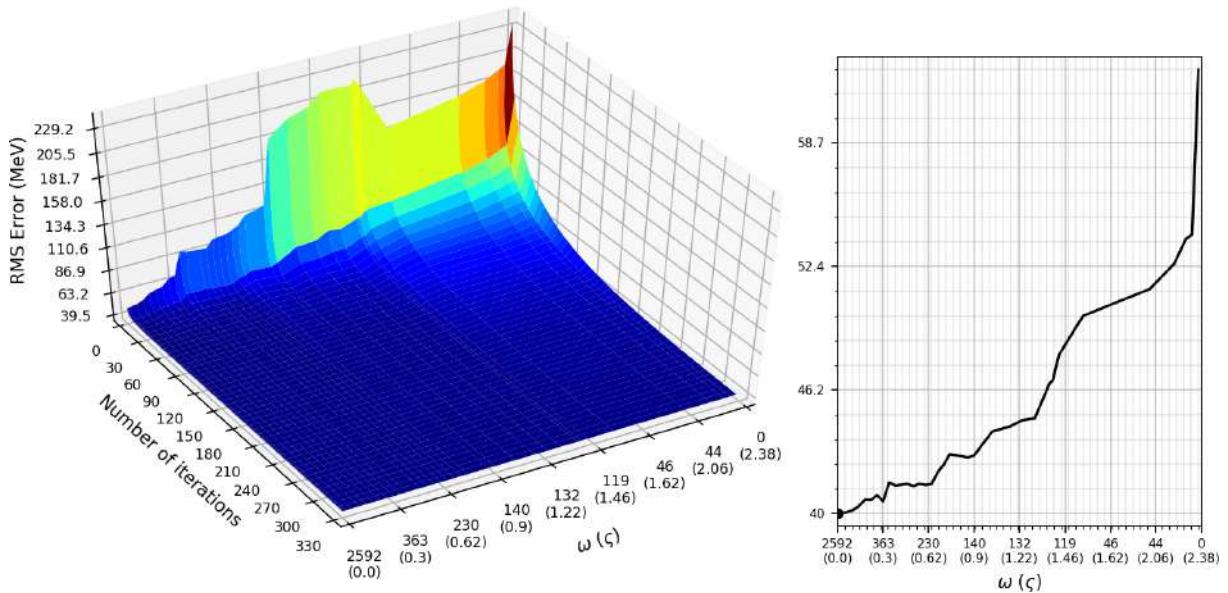
Fonte: Elaborada pelo autor (2021).

Gráfico 46 - Valores de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o método SSFi, com taxas de ocupação de 50% e 60%

(a) Taxa de Ocupação de 50%



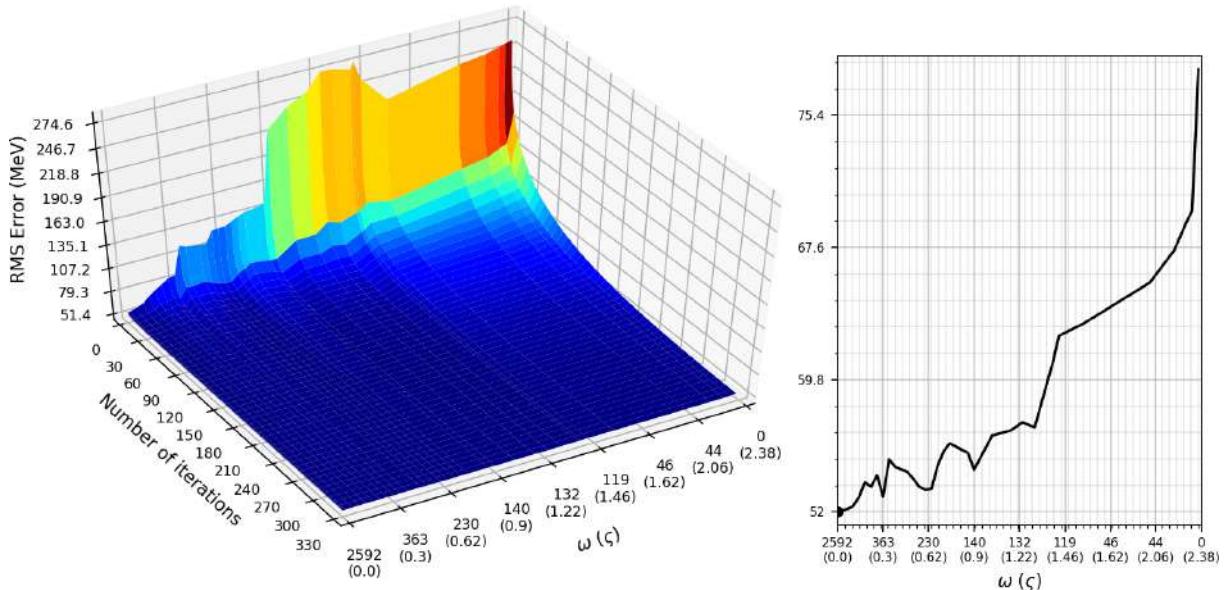
(b) Taxa de Ocupação de 60%



Fonte: Elaborada pelo autor (2021).

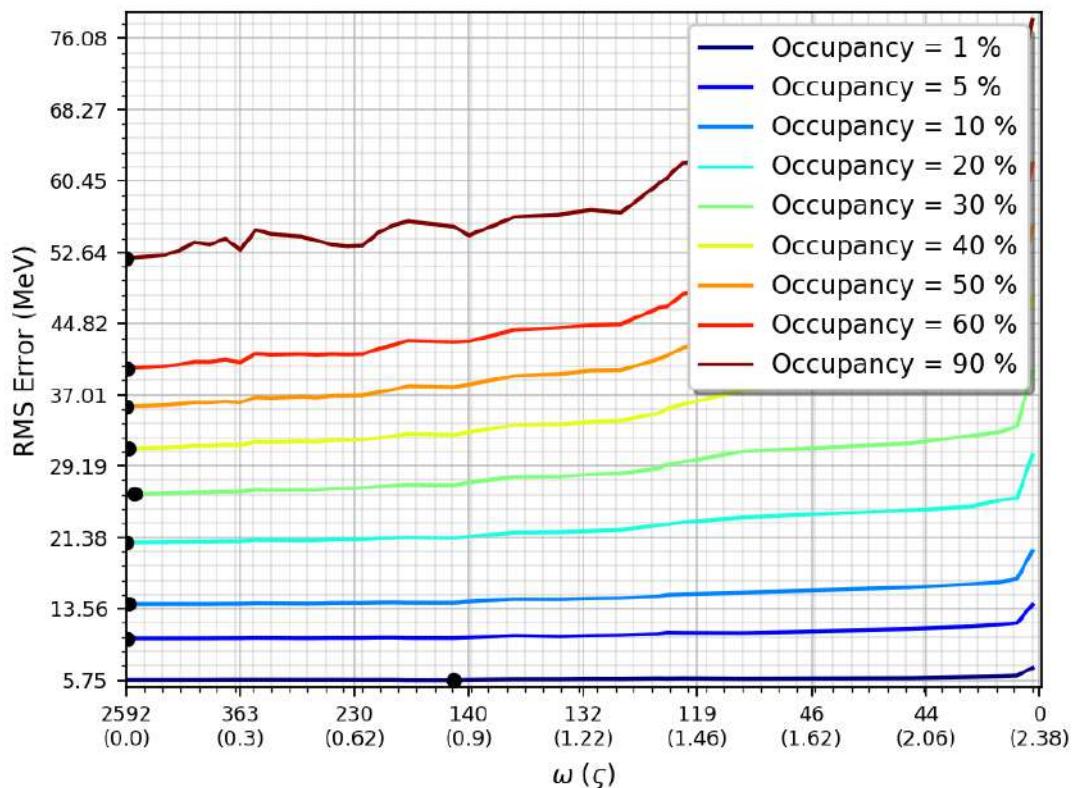
Gráfico 47 - Valores de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o método SSFi, com taxa de ocupação de 90%.

(a) Taxa de Ocupação de 90%



Fonte: Elaborada pelo autor (2021).

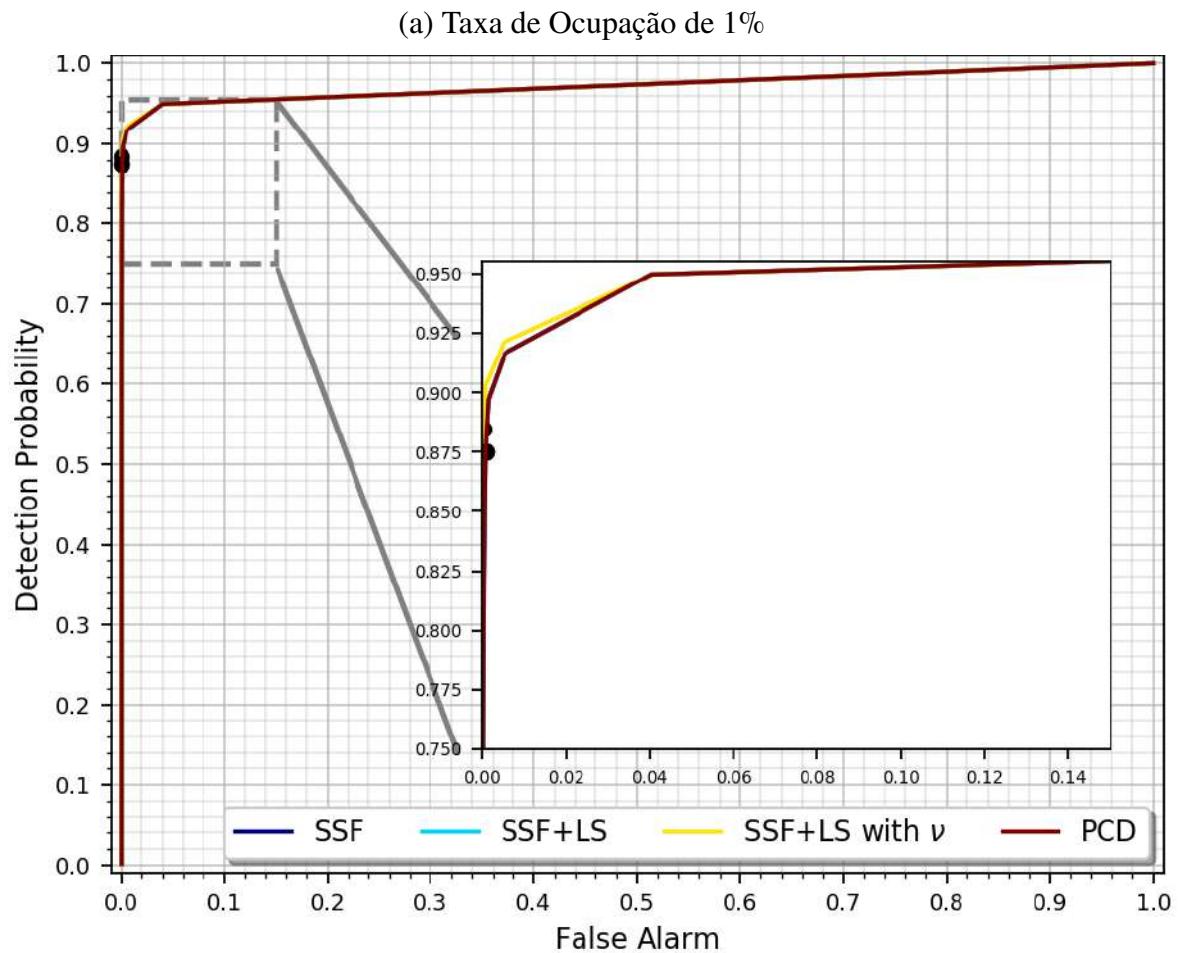
Gráfico 48 - Valores médios de RMSr, obtidos com a variação do número de elementos da matriz H^+ para o variação SSFi, de acordo com as taxas de ocupação



Fonte: Elaborado pelo autor (2021).

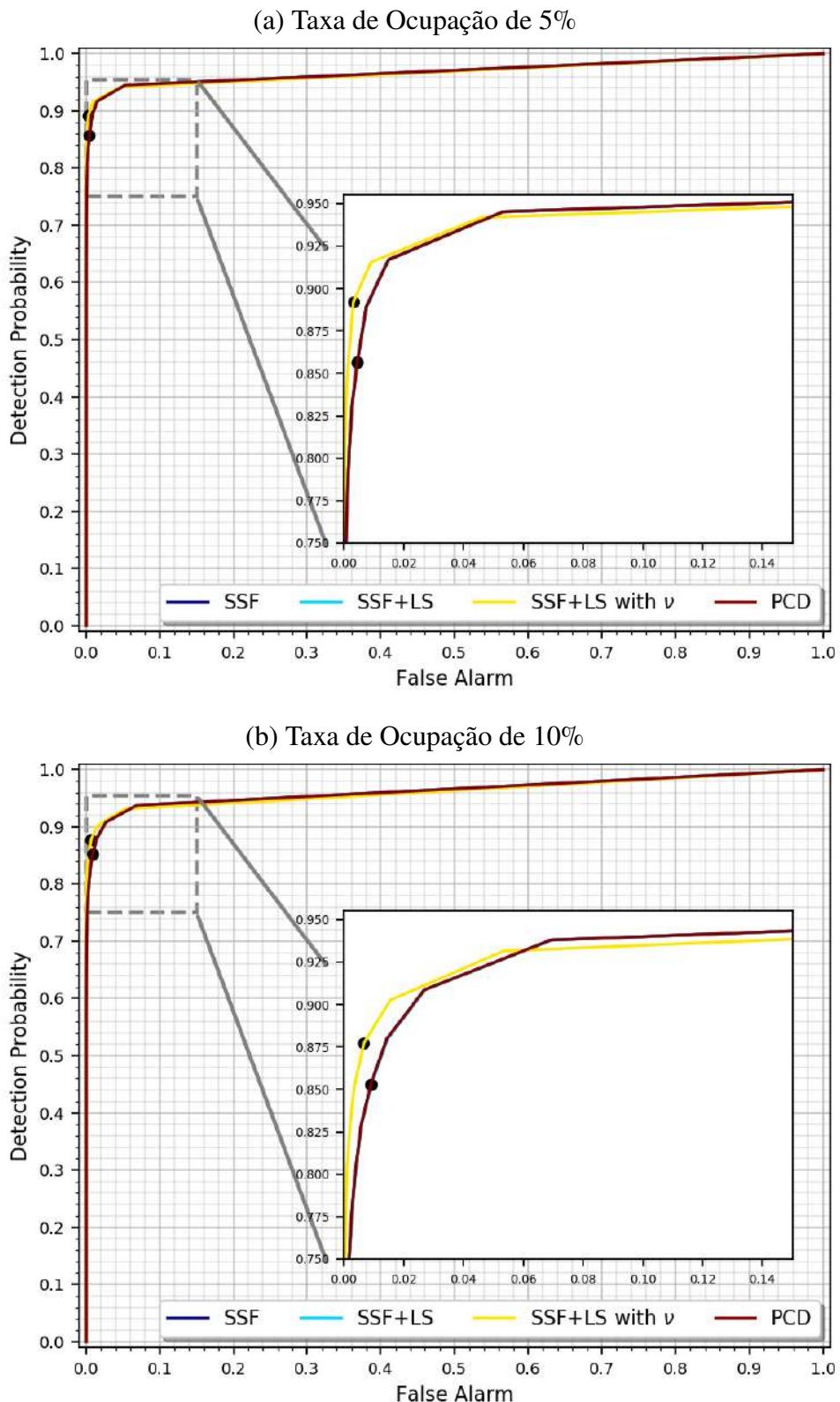
D.3.5 Curvas ROC de métodos *Shrinkage*, sem inicialização prévia

Gráfico 49 - Curvas ROC dos métodos da família *Shrinkage*, sem inicialização prévia, para taxa de ocupação de 1%



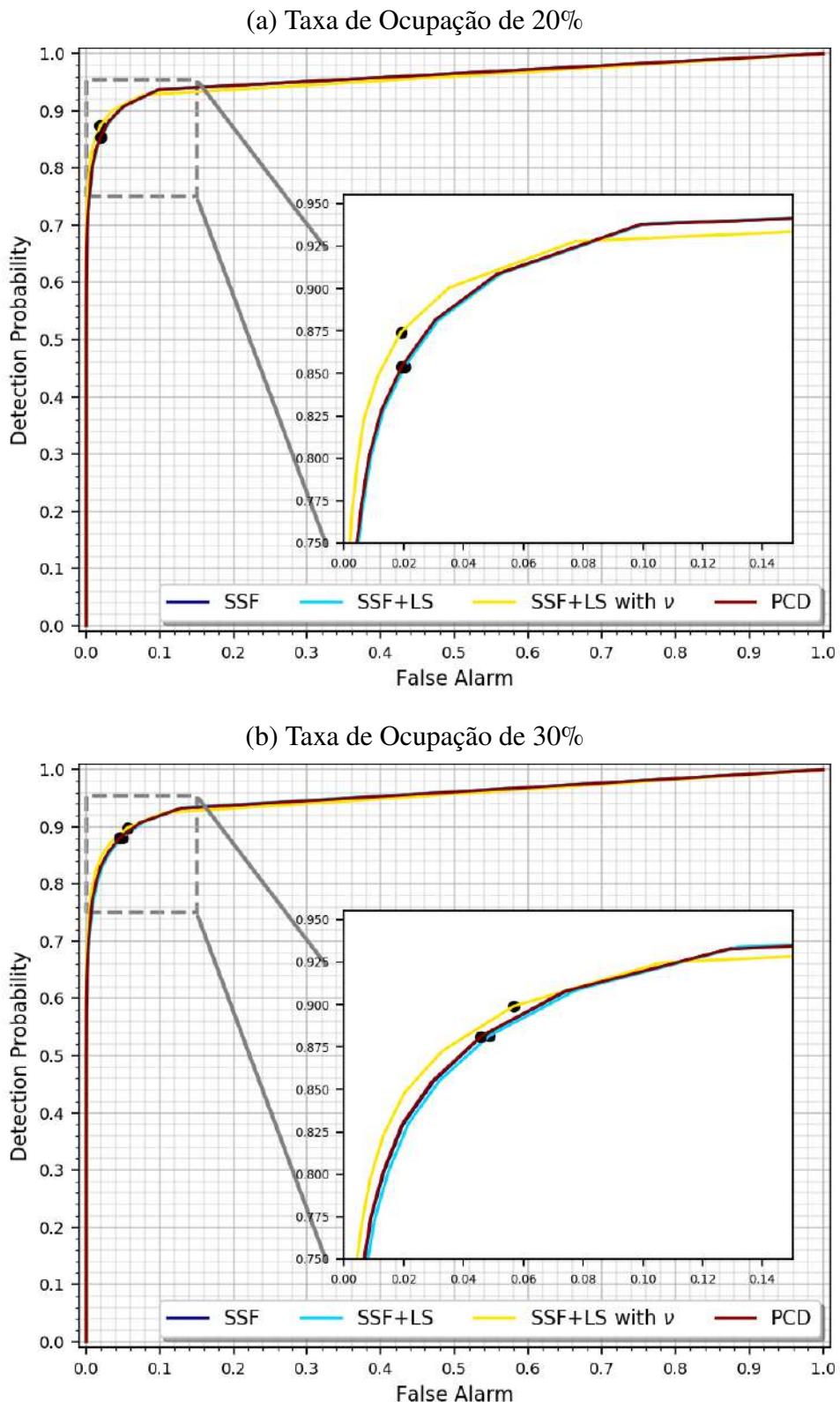
Fonte: Elaborada pelo autor (2021).

Gráfico 50 - Curvas ROC dos métodos da família *Shrinkage*, sem inicialização prévia, para taxas de ocupação de 5% e 10%



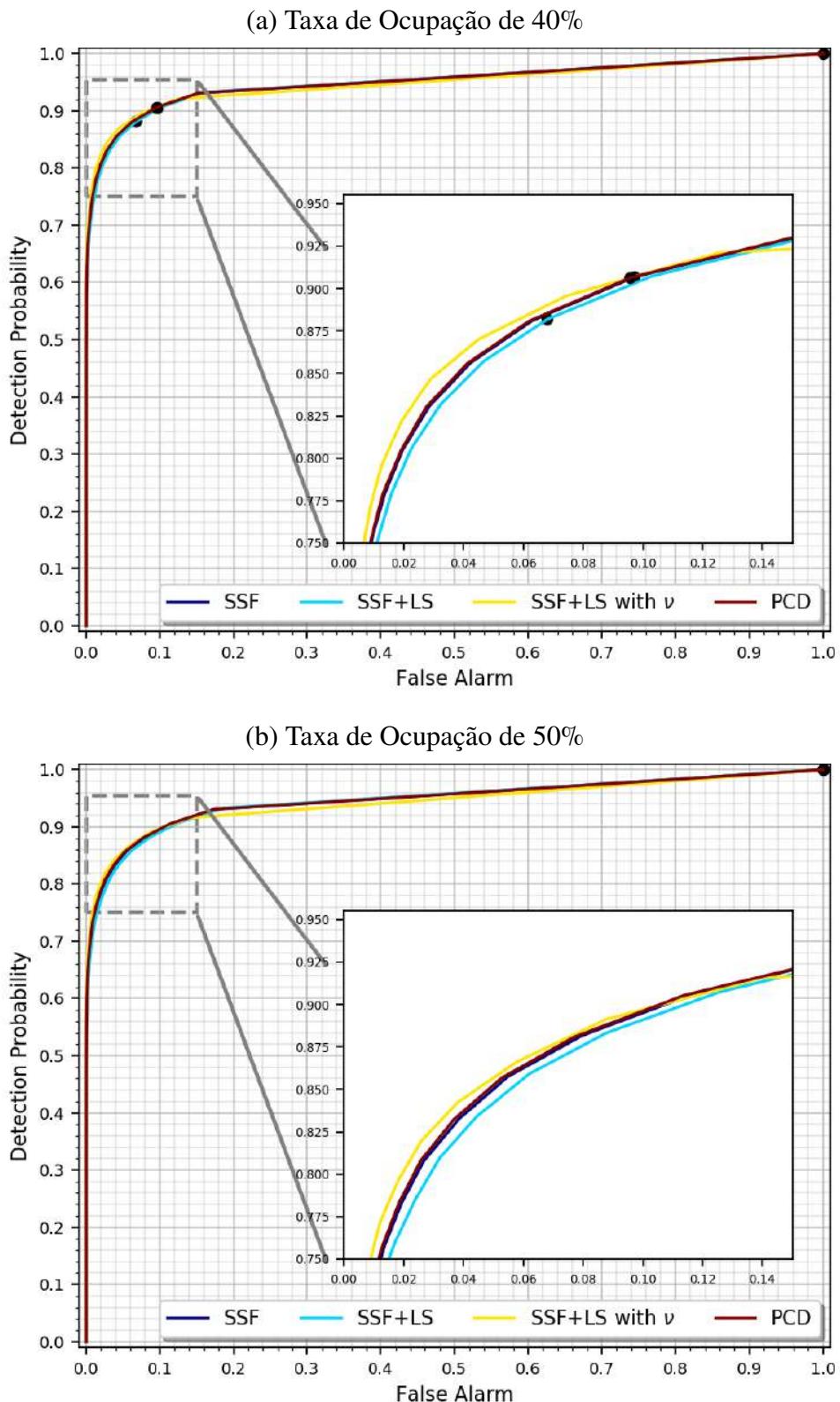
Fonte: Elaborada pelo autor (2021).

Gráfico 51 - Curvas ROC dos métodos da família *Shrinkage*, sem inicialização prévia, para taxas de ocupação de 20% e 30%



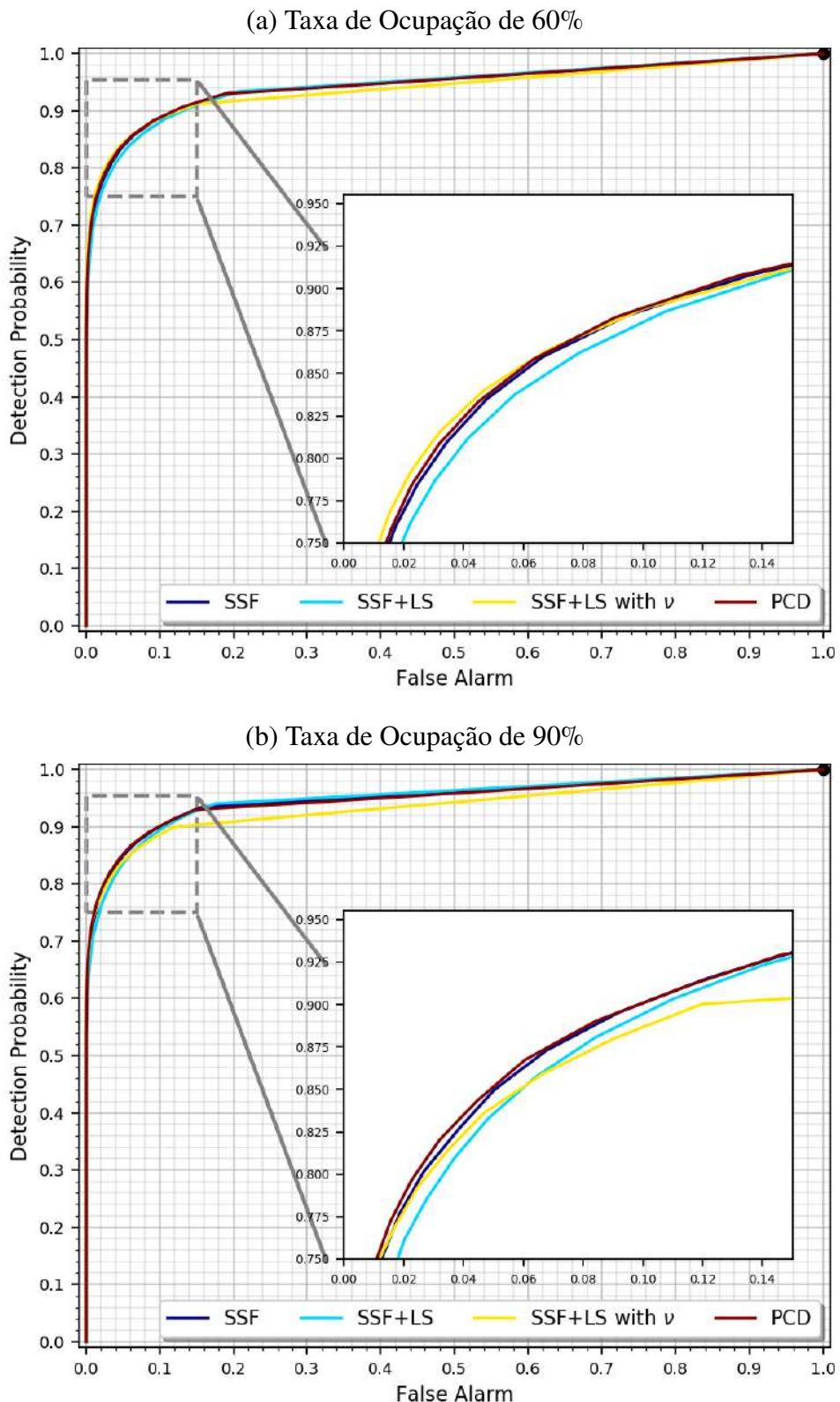
Fonte: Elaborada pelo autor (2021).

Gráfico 52 - Curvas ROC dos métodos da família *Shrinkage*, sem inicialização prévia, para taxas de ocupação de 40% e 50%



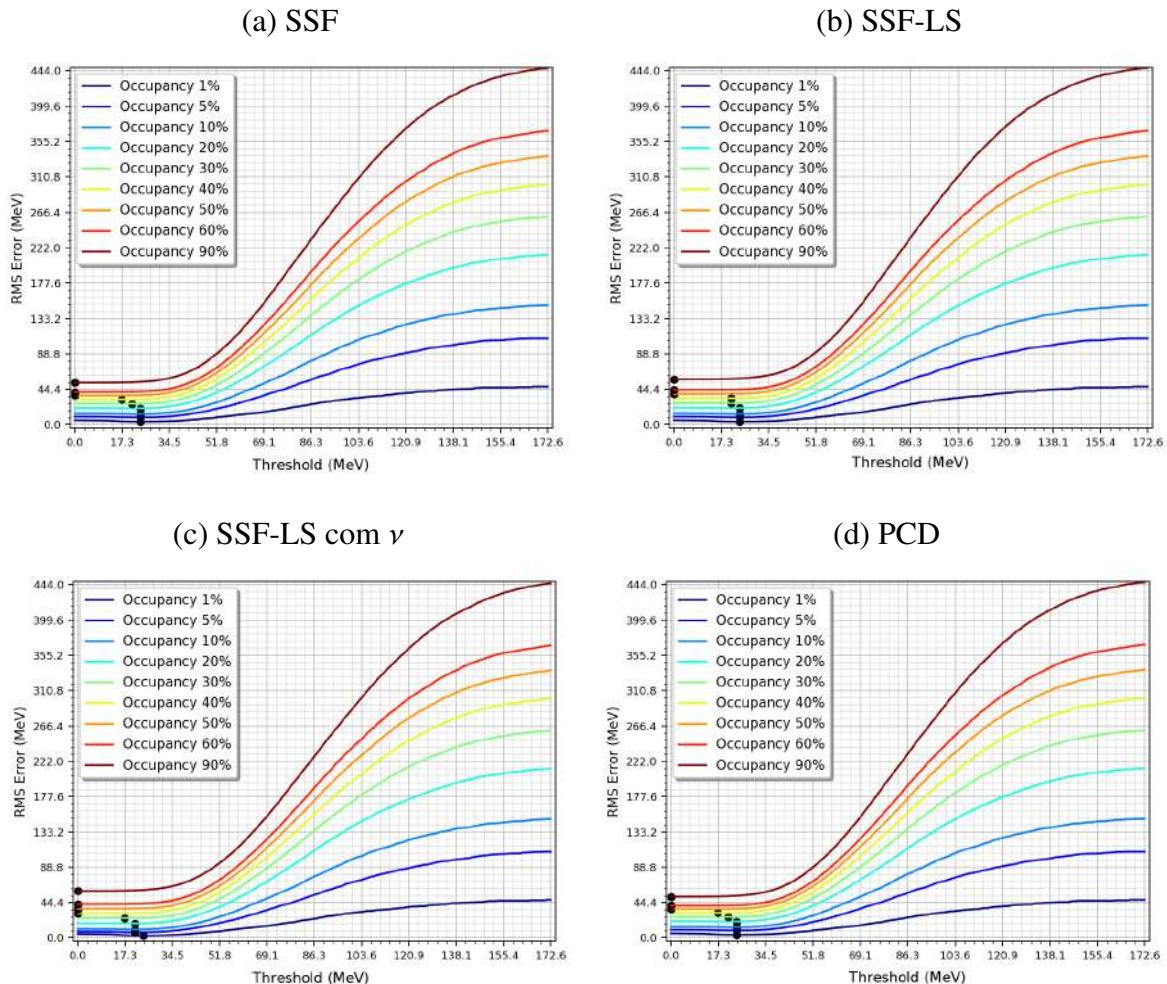
Fonte: Elaborada pelo autor (2021).

Gráfico 53 - Curvas ROC dos métodos da família *Shrinkage*, sem inicialização prévia, para taxas de ocupação de 60% e 90%



Fonte: Elaborada pelo autor (2021).

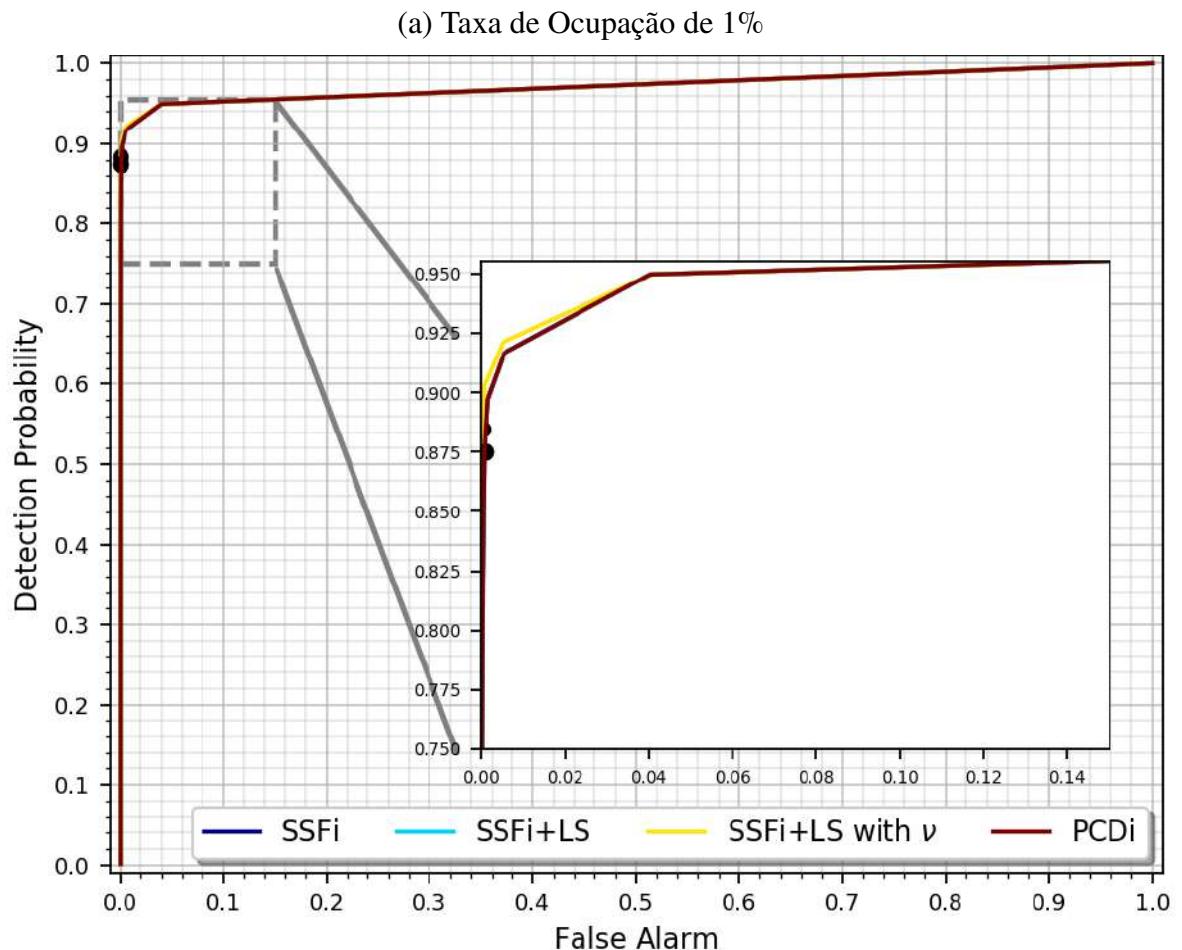
Gráfico 54 - Variação do RMSr da aplicação dos métodos *Shrinkage*, sem inicialização prévia, de acordo com as taxas de ocupação



Fonte: Elaborado pelo autor (2021).

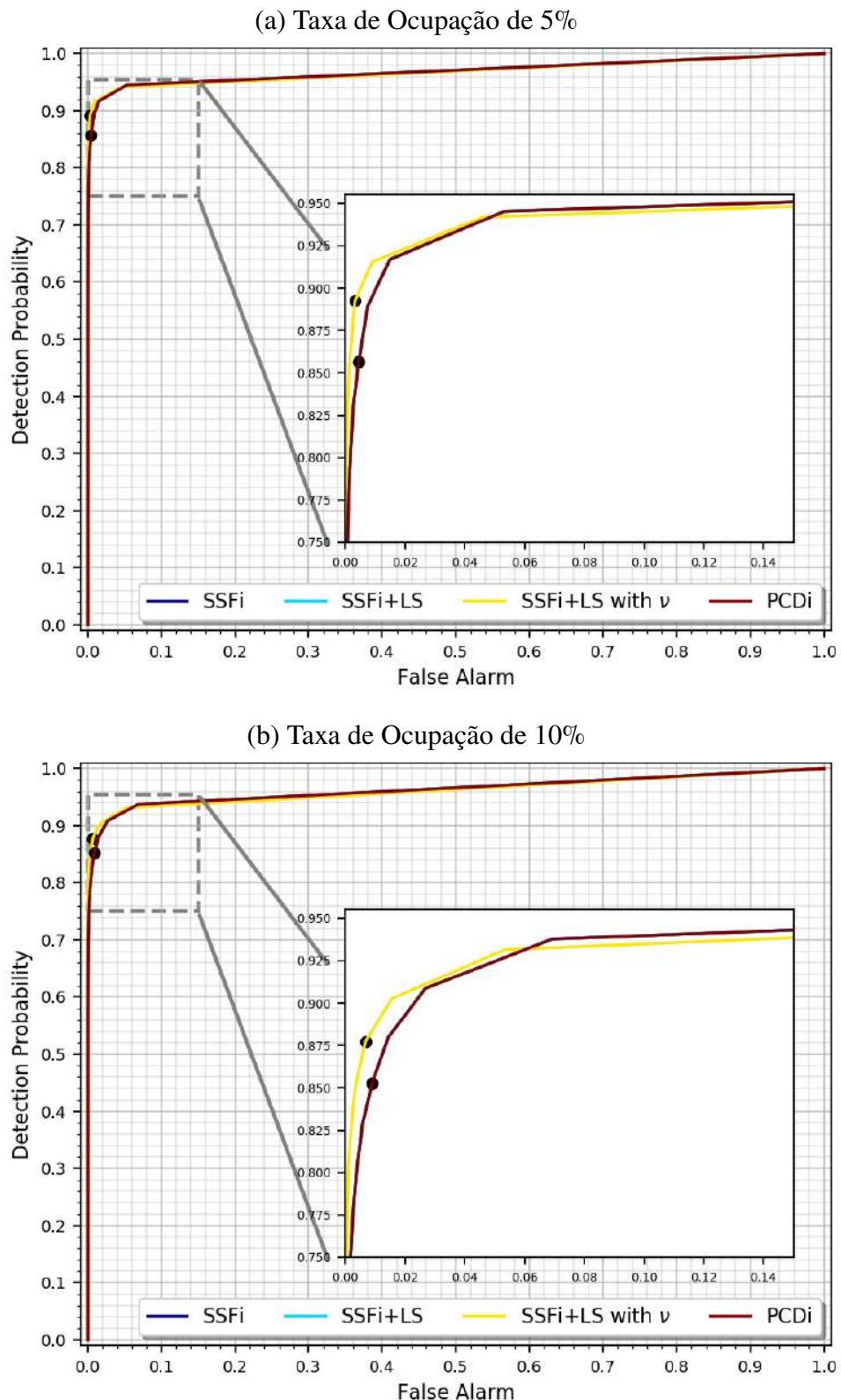
D.3.6 Curvas ROC de métodos *Shrinkage*, com inicialização prévia

Gráfico 55 - Curvas ROC dos métodos da família *Shrinkage*, com inicialização prévia, para taxa de ocupação de 1%



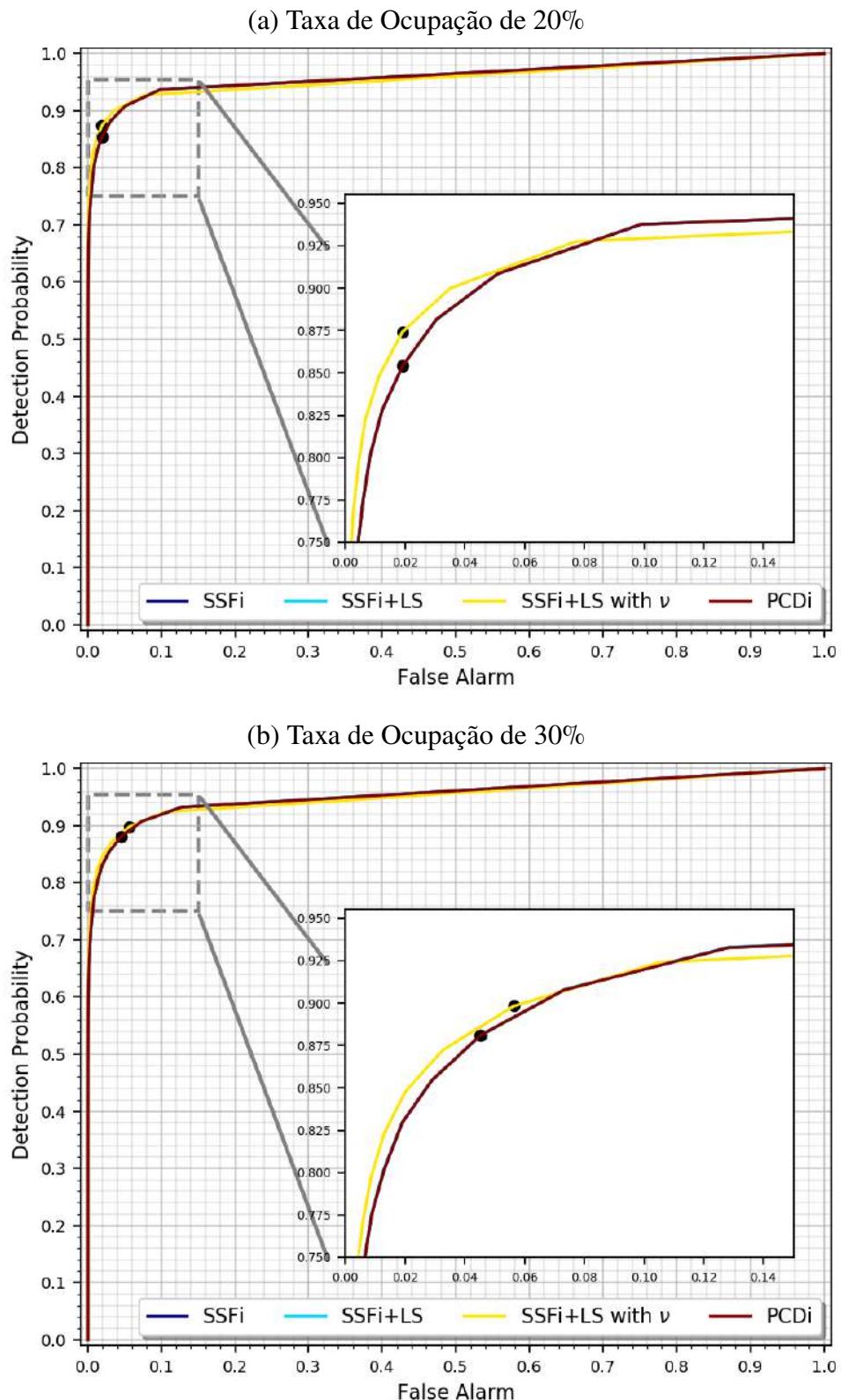
Fonte: Elaborada pelo autor (2021).

Gráfico 56 - Curvas ROC dos métodos da família *Shrinkage*, com inicialização prévia, para taxas de ocupação de 5% e 10%



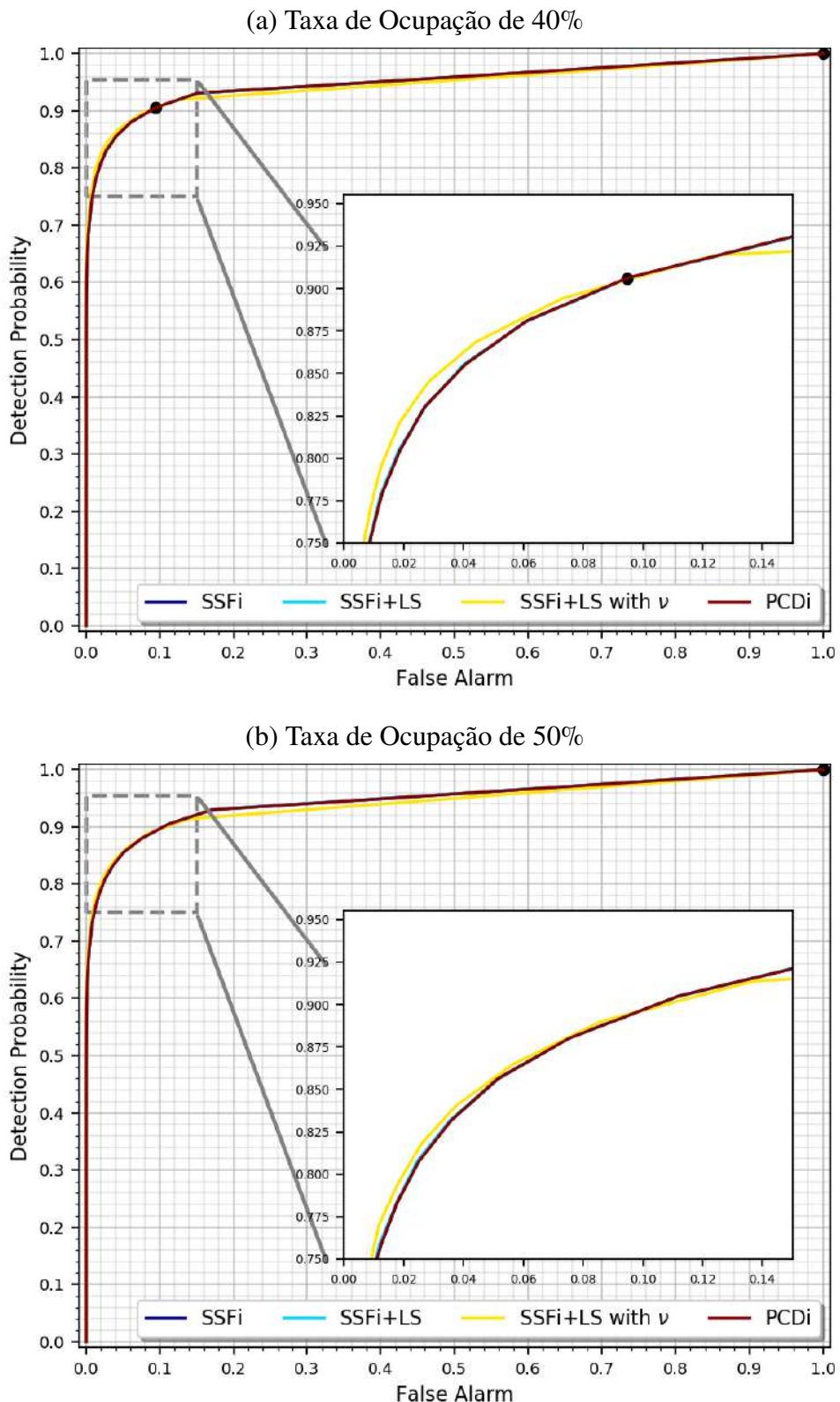
Fonte: Elaborada pelo autor (2021).

Gráfico 57 - Curvas ROC dos métodos da família *Shrinkage*, com inicialização prévia, para taxas de ocupação de 20% e 30%



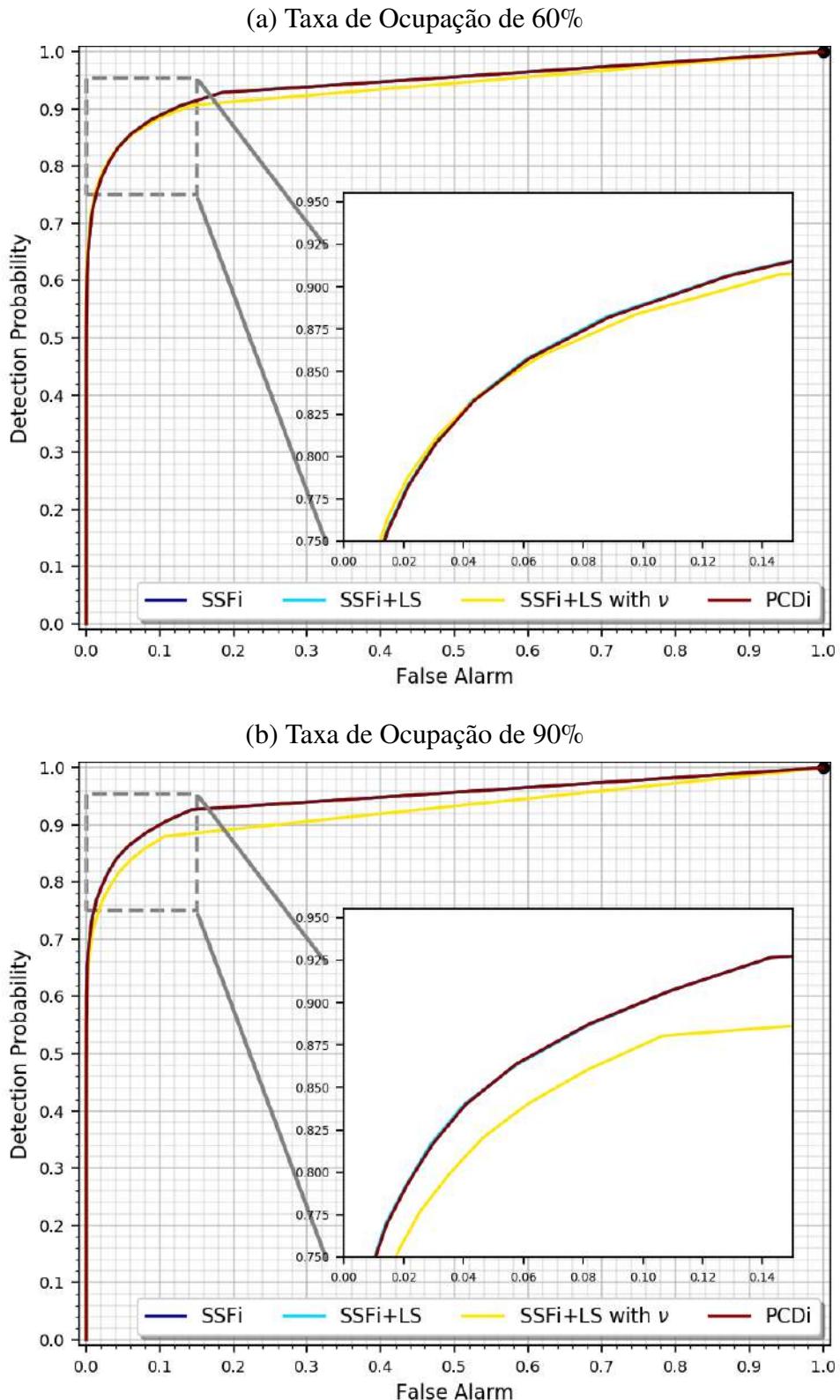
Fonte: Elaborada pelo autor (2021).

Gráfico 58 - Curvas ROC dos métodos da família *Shrinkage*, com inicialização prévia, para taxas de ocupação de 40% e 50%



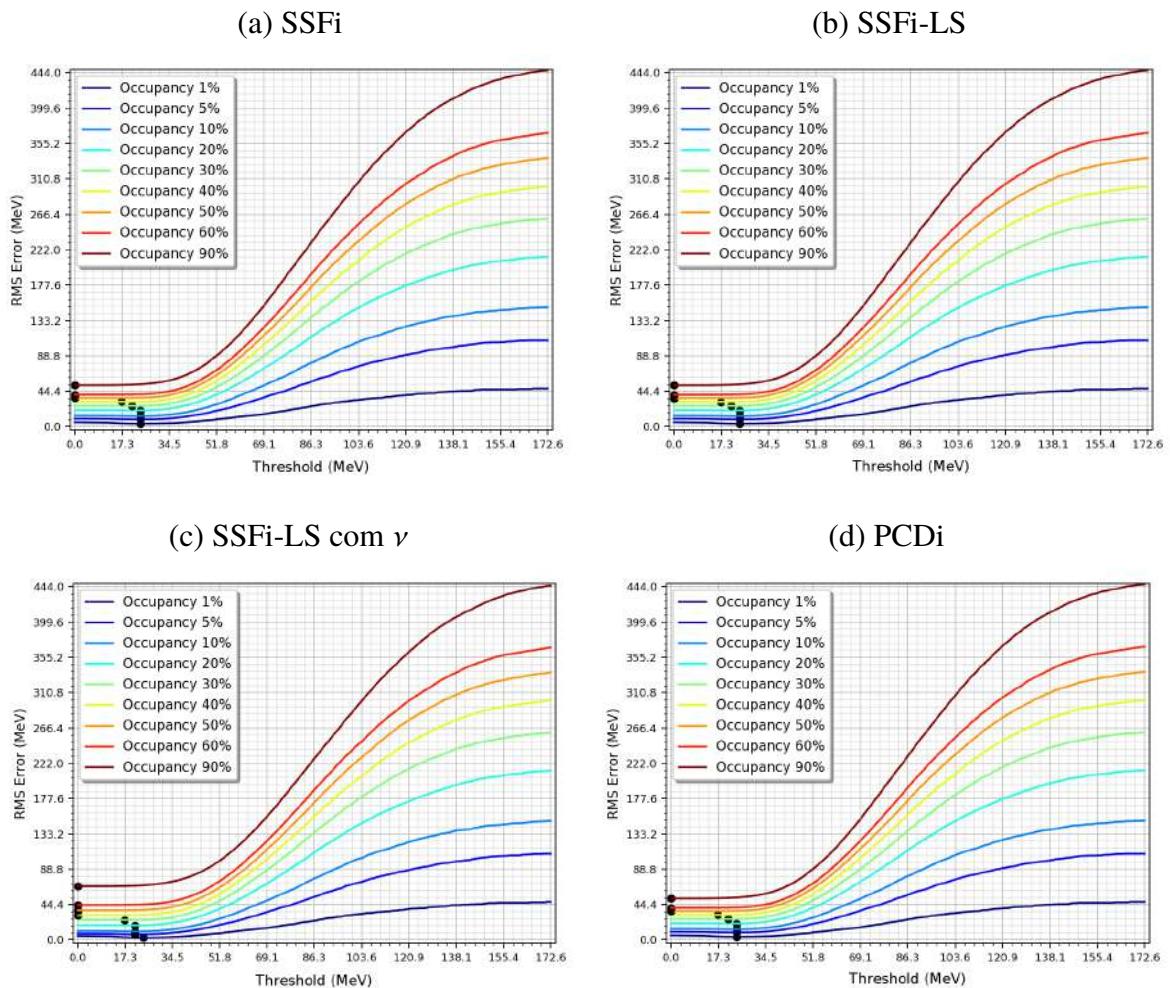
Fonte: Elaborada pelo autor (2021).

Gráfico 59 - Curvas ROC dos métodos da família *Shrinkage*, com inicialização prévia, para taxas de ocupação de 60% e 90%



Fonte: Elaborada pelo autor (2021).

Gráfico 60 - Variação do RMSr da aplicação dos métodos *Shrinkage*, com inicialização prévia, de acordo com as taxas de ocupação

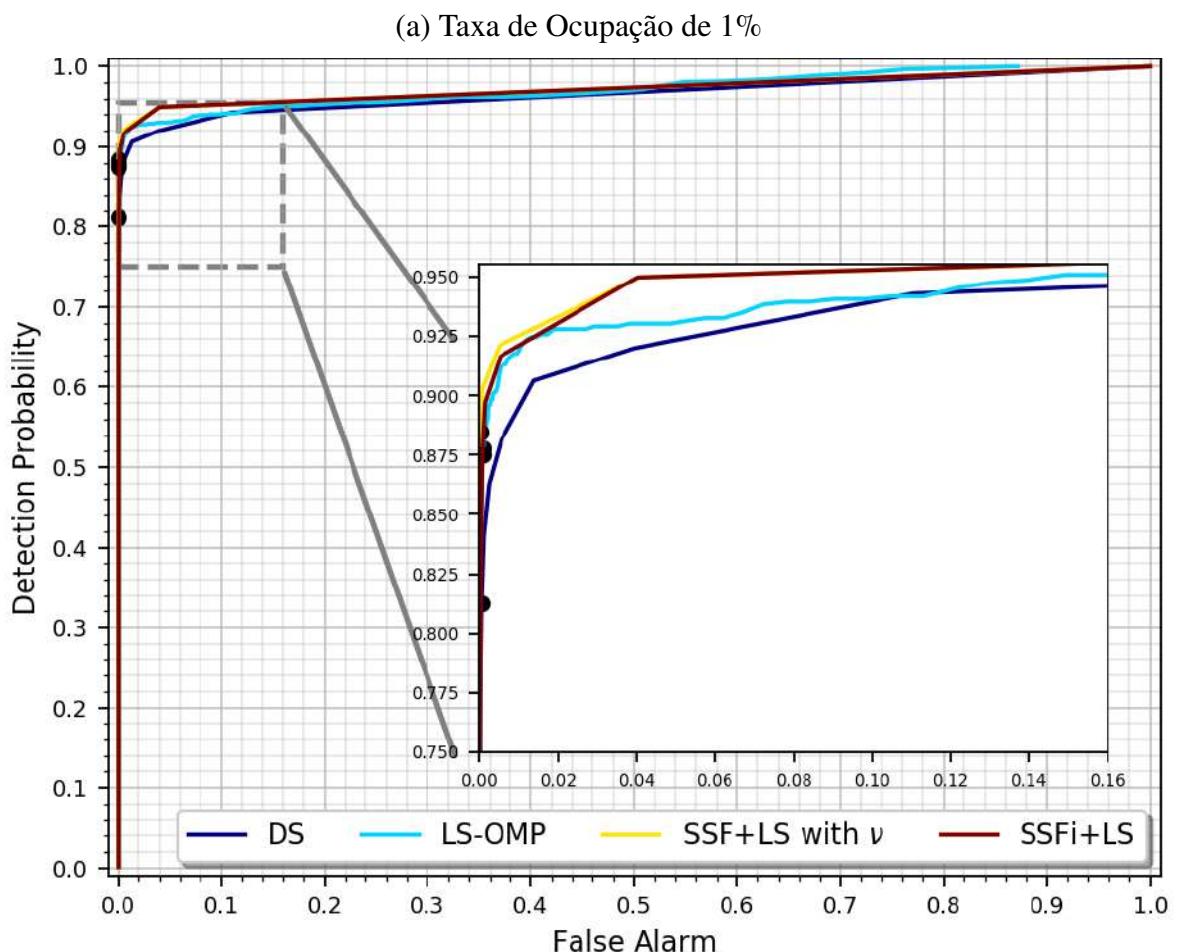


Fonte: Elaborado pelo autor (2021).

D.4 Detalhamento Gráfico Comparativo entre os Métodos

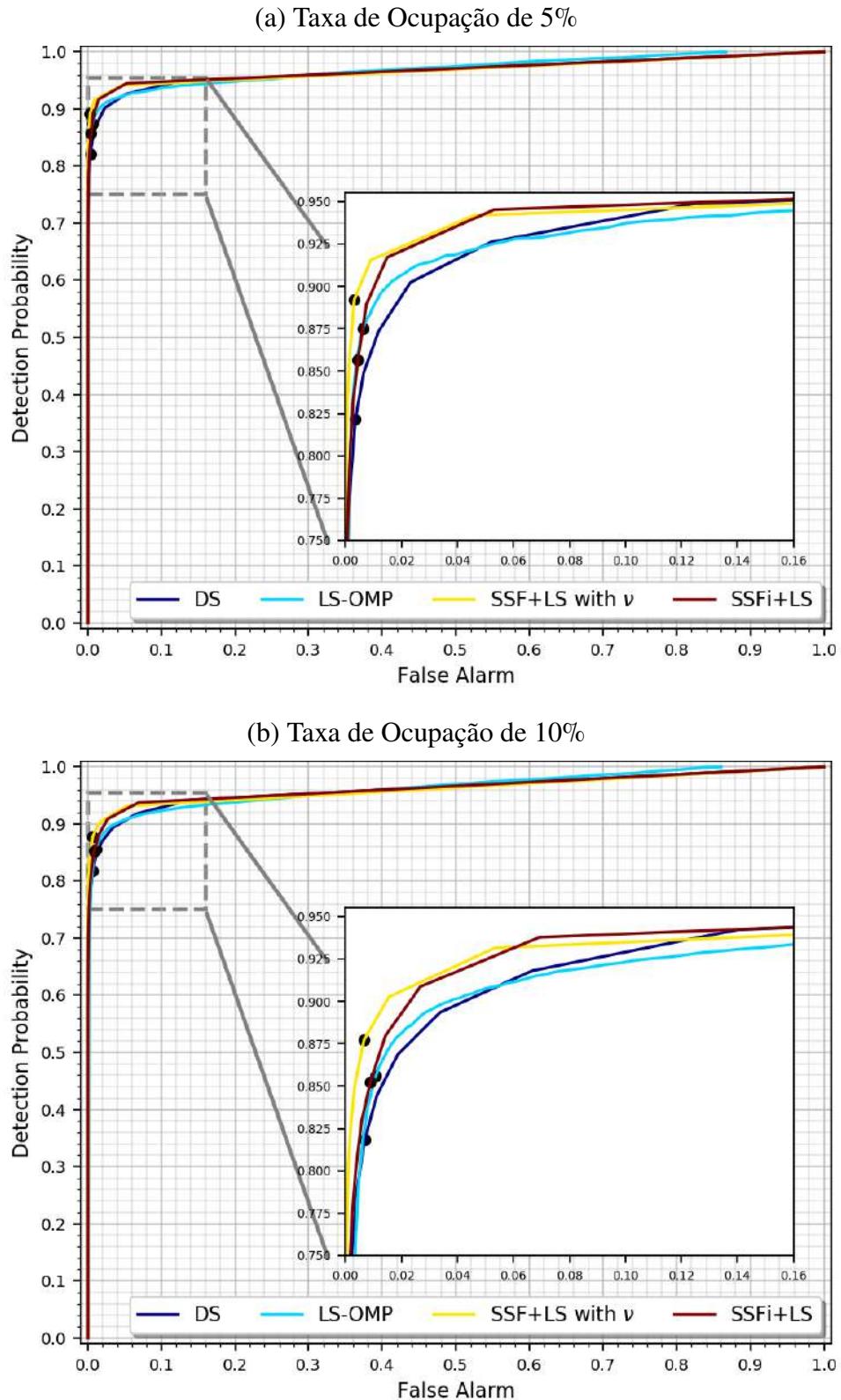
D.4.1 Comparativo de Curvas ROC

Gráfico 61 - Curvas ROC dos métodos que apresentaram os menores resultados de RMS da diferença do erro, dentre os de mesma classificação, para taxa de ocupação de 1%



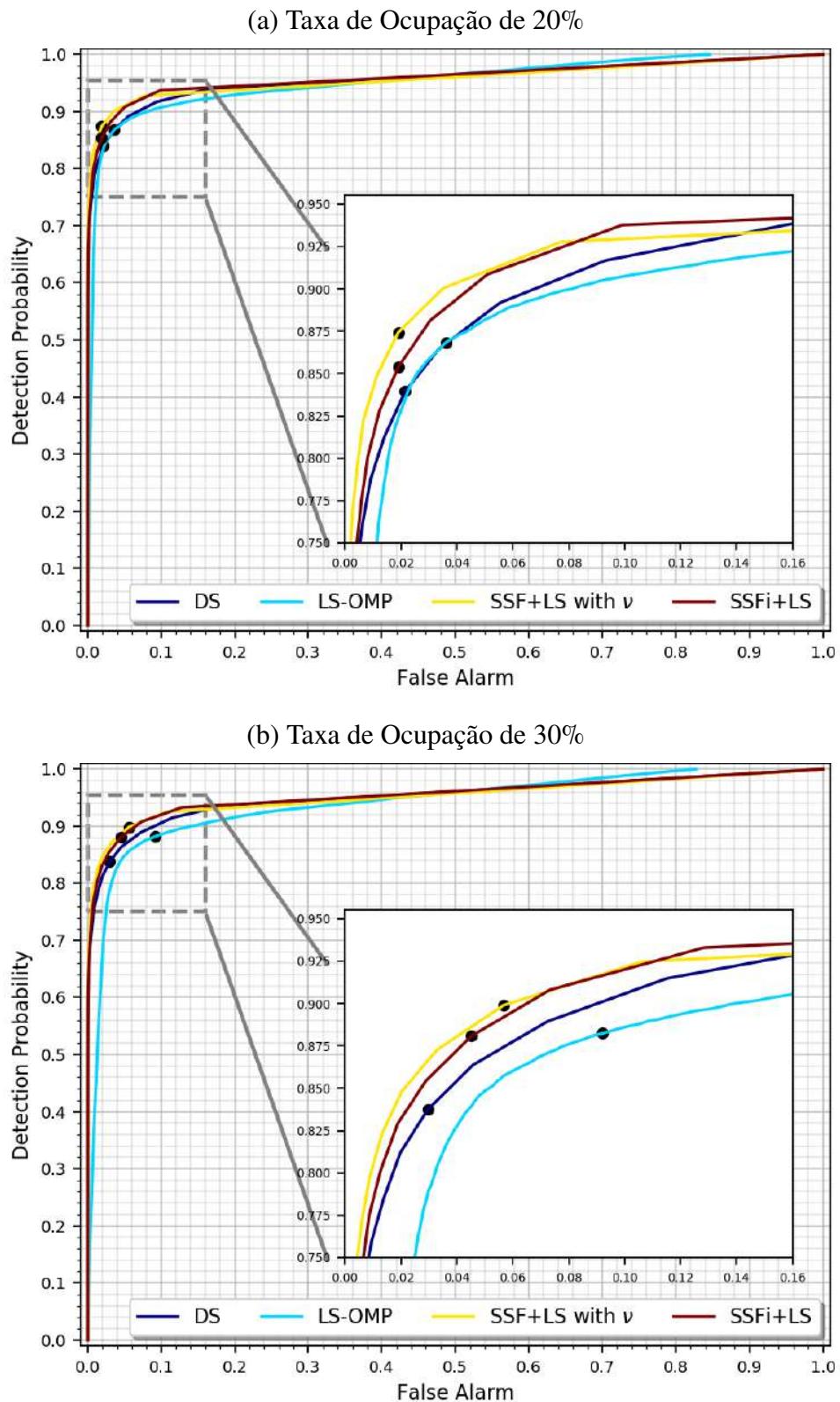
Fonte: Elaborada pelo autor (2021).

Gráfico 62 - Curvas ROC dos métodos que apresentaram os menores resultados de RMS da diferença do erro, dentre os de mesma classificação, para taxas de ocupação de 5% e 10%



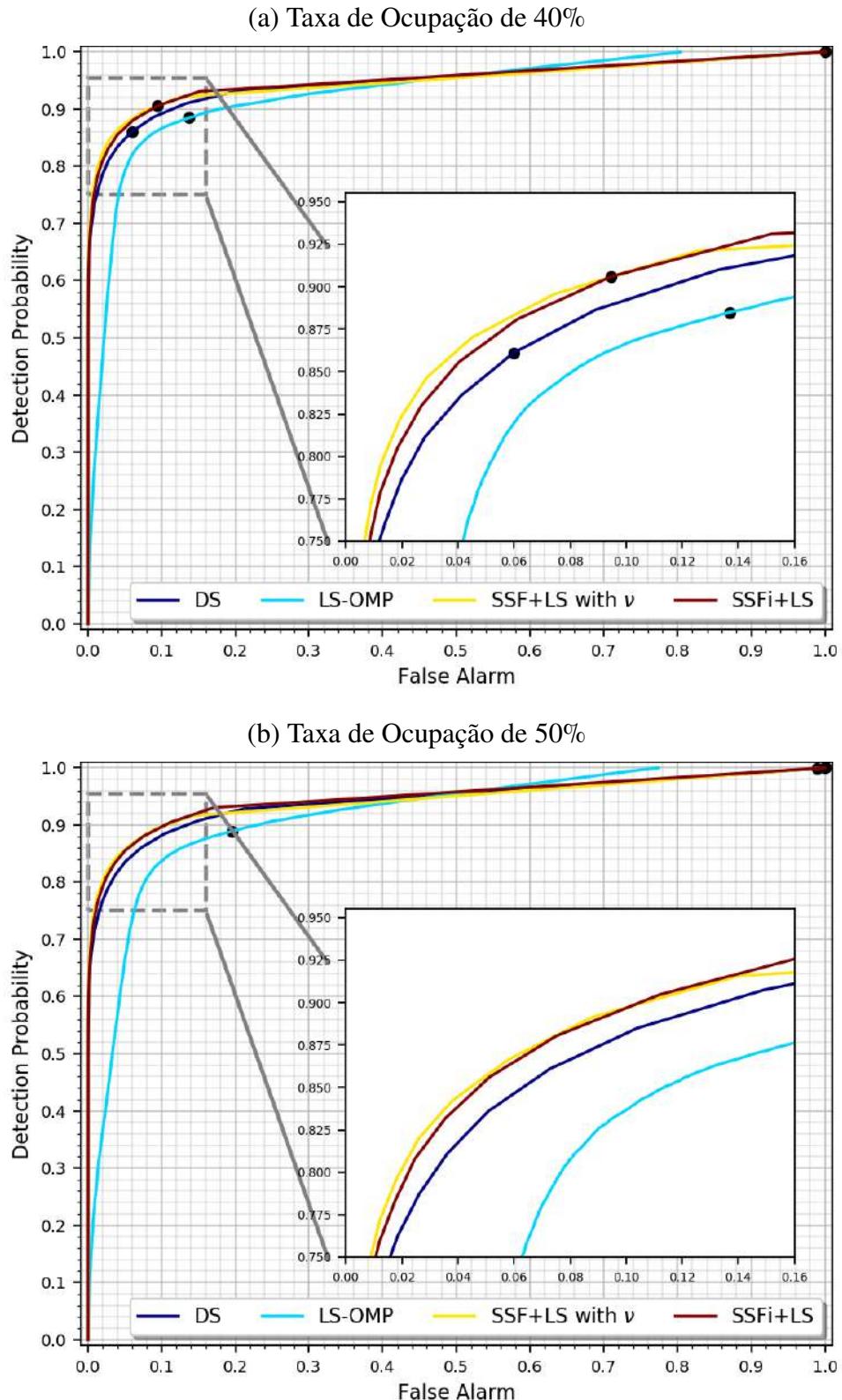
Fonte: Elaborada pelo autor (2021).

Gráfico 63 - Curvas ROC dos métodos que apresentaram os menores resultados de RMS da diferença do erro, dentre os de mesma classificação, para taxas de ocupação de 20% e 30%



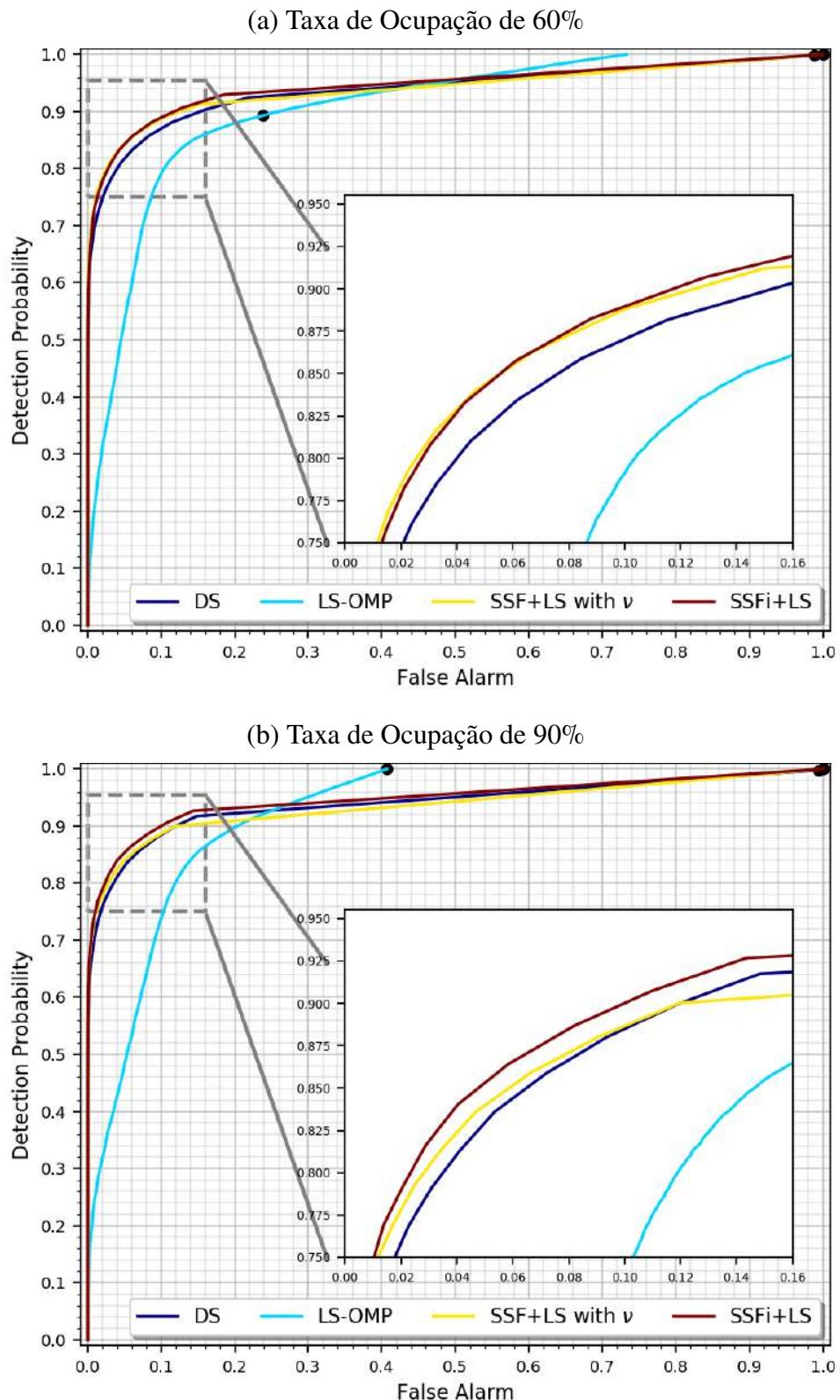
Fonte: Elaborada pelo autor (2021).

Gráfico 64 - Curvas ROC dos métodos que apresentaram os menores resultados de RMS da diferença do erro, dentre os de mesma classificação, para taxas de ocupação de 40% e 50%.



Fonte: Elaborada pelo autor (2021).

Gráfico 65 - Curvas ROC dos métodos que apresentaram os menores resultados de RMS da diferença do erro, dentre os de mesma classificação, para taxas de ocupação de 60% e 90%.

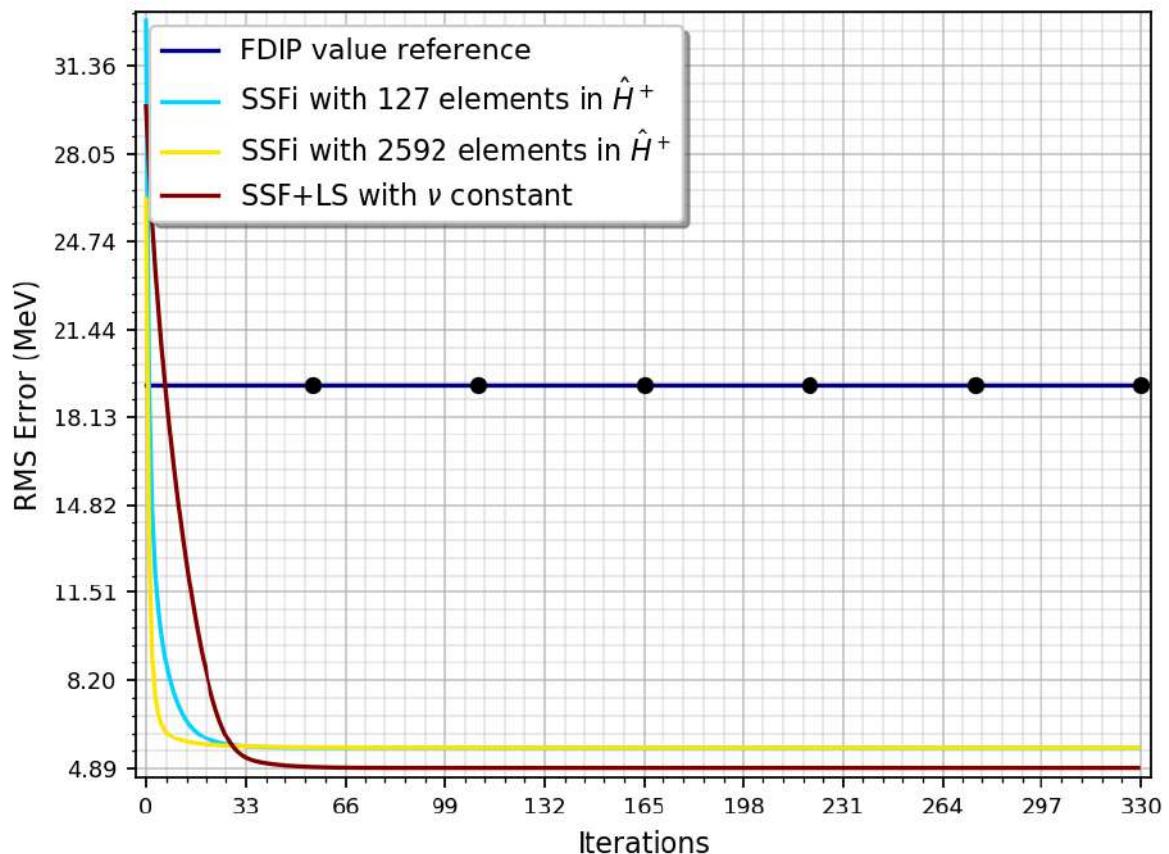


Fonte: Elaborada pelo autor (2021).

D.4.2 Comparativo de Variações do Método SSF

Gráfico 66 - Agrupamento das variações do método SSF que apresentaram os menores resultados de RMS da diferença do erro, para taxa de ocupação de 1%

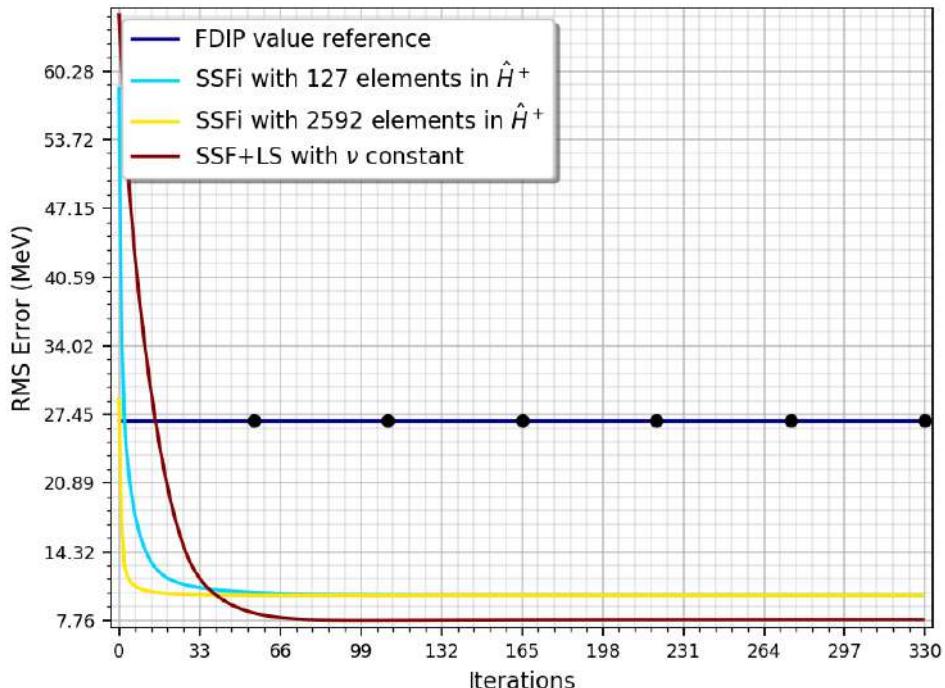
(a) Taxa de Ocupação de 1%



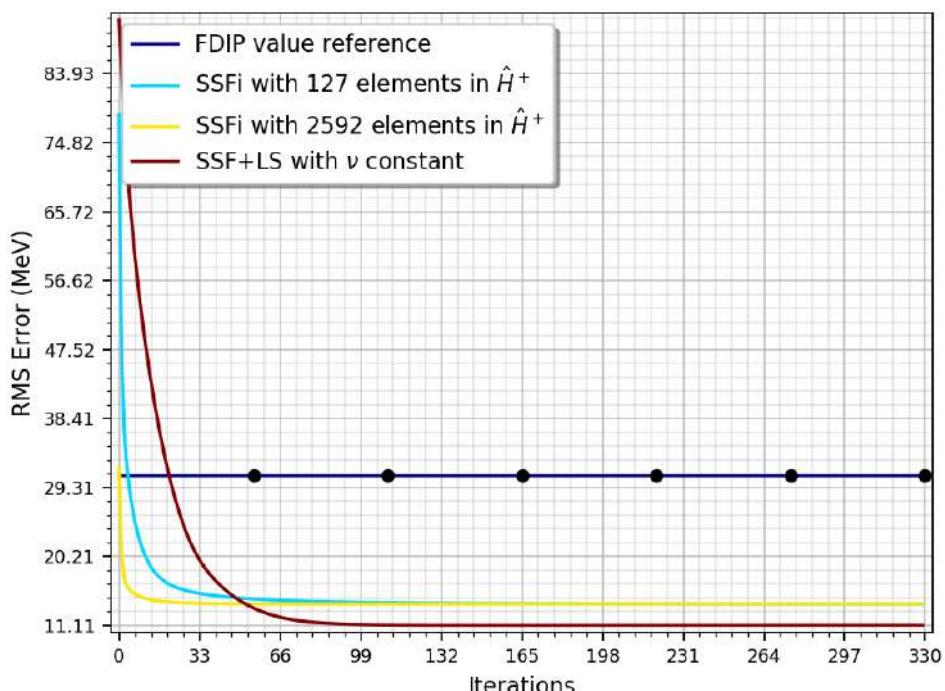
Fonte: Elaborada pelo autor (2021).

Gráfico 67 - Agrupamento das variações do método SSF que apresentaram os menores resultados de RMS da diferença do erro, para taxas de ocupação de 5% e 10%

(a) Taxa de Ocupação de 5%



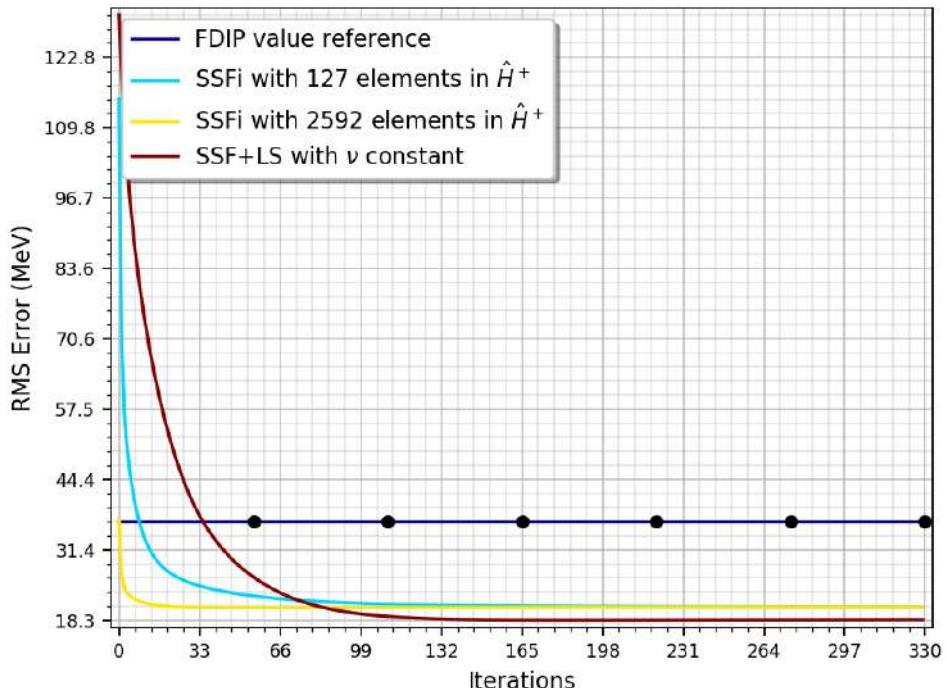
(b) Taxa de Ocupação de 10%



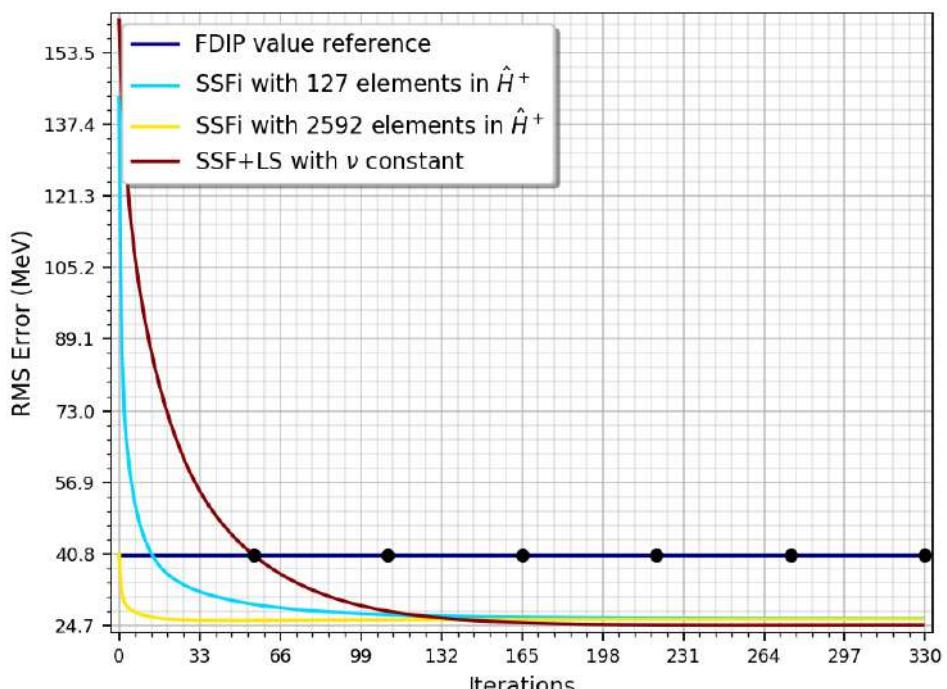
Fonte: Elaborada pelo autor (2021).

Gráfico 68 - Agrupamento das variações do método SSF que apresentaram os menores resultados de RMS da diferença do erro, para taxas de ocupação de 20% e 30%

(a) Taxa de Ocupação de 20%



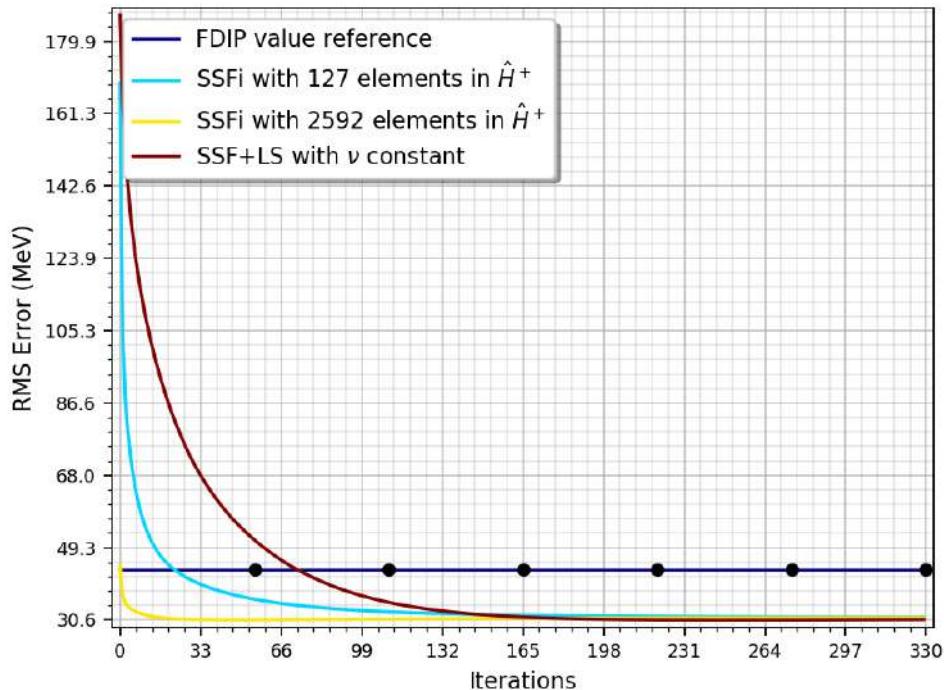
(b) Taxa de Ocupação de 30%



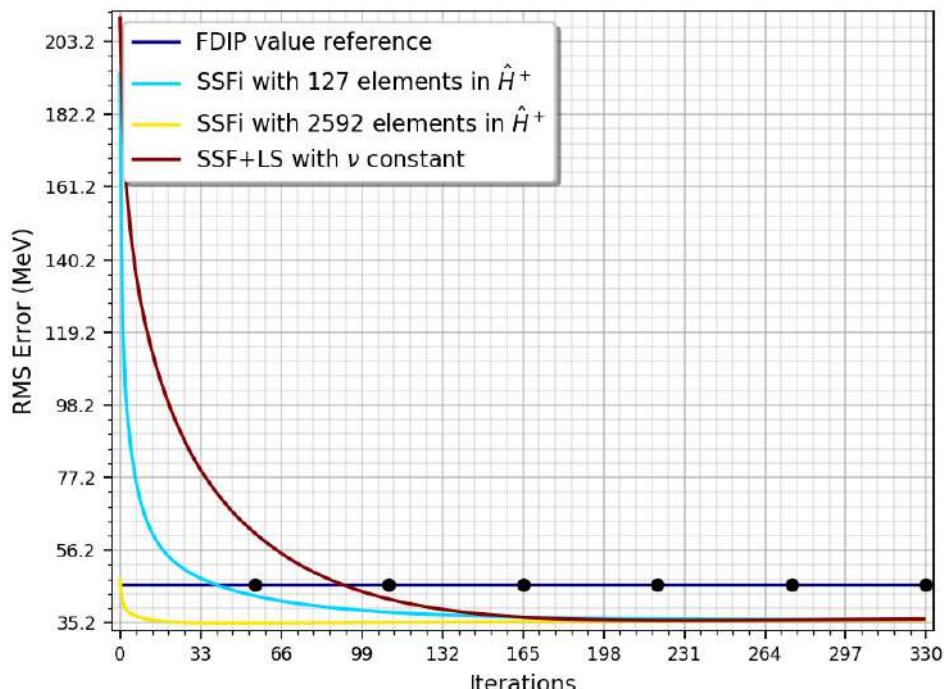
Fonte: Elaborada pelo autor (2021).

Gráfico 69 - Agrupamento das variações do método SSF que apresentaram os menores resultados de RMS da diferença do erro, para taxas de ocupação de 40% e 50%

(a) Taxa de Ocupação de 40%



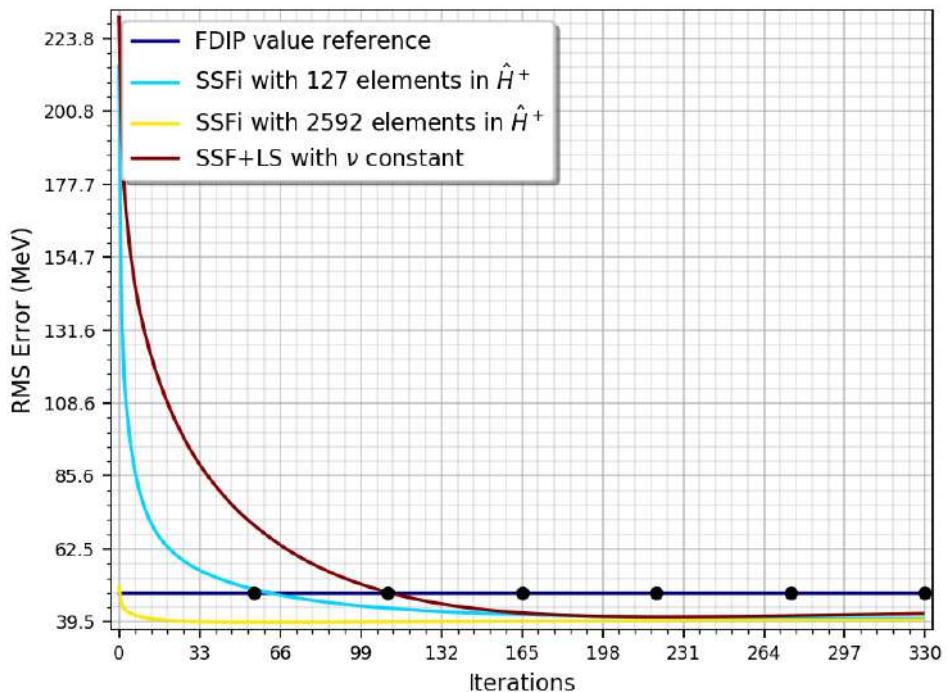
(b) Taxa de Ocupação de 50%



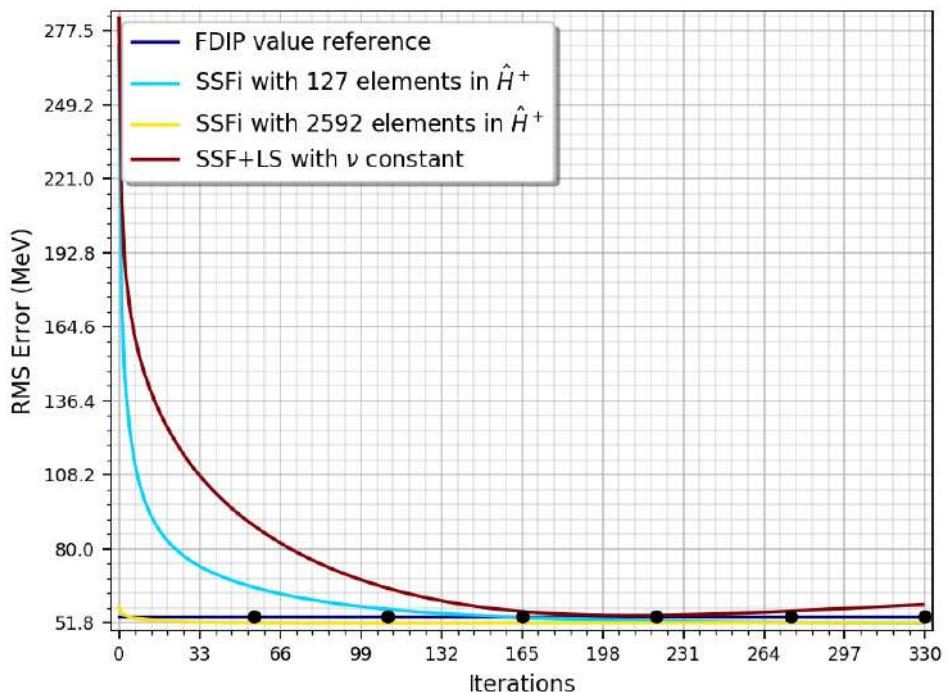
Fonte: Elaborada pelo autor (2021).

Gráfico 70 - Agrupamento das variações do método SSF que apresentaram os menores resultados de RMS da diferença do erro, para taxas de ocupação de 60% e 90%.

(a) Taxa de Ocupação de 60%



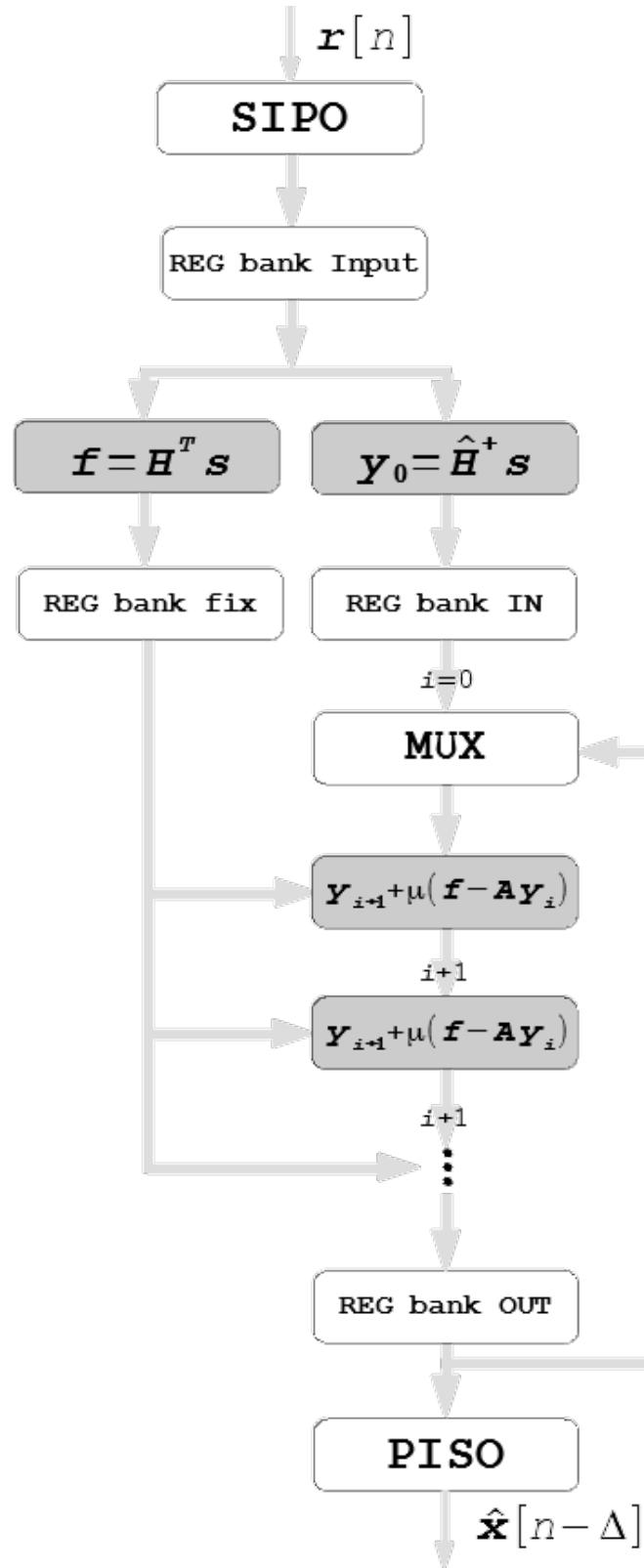
(b) Taxa de Ocupação de 90%



Fonte: Elaborada pelo autor (2021).

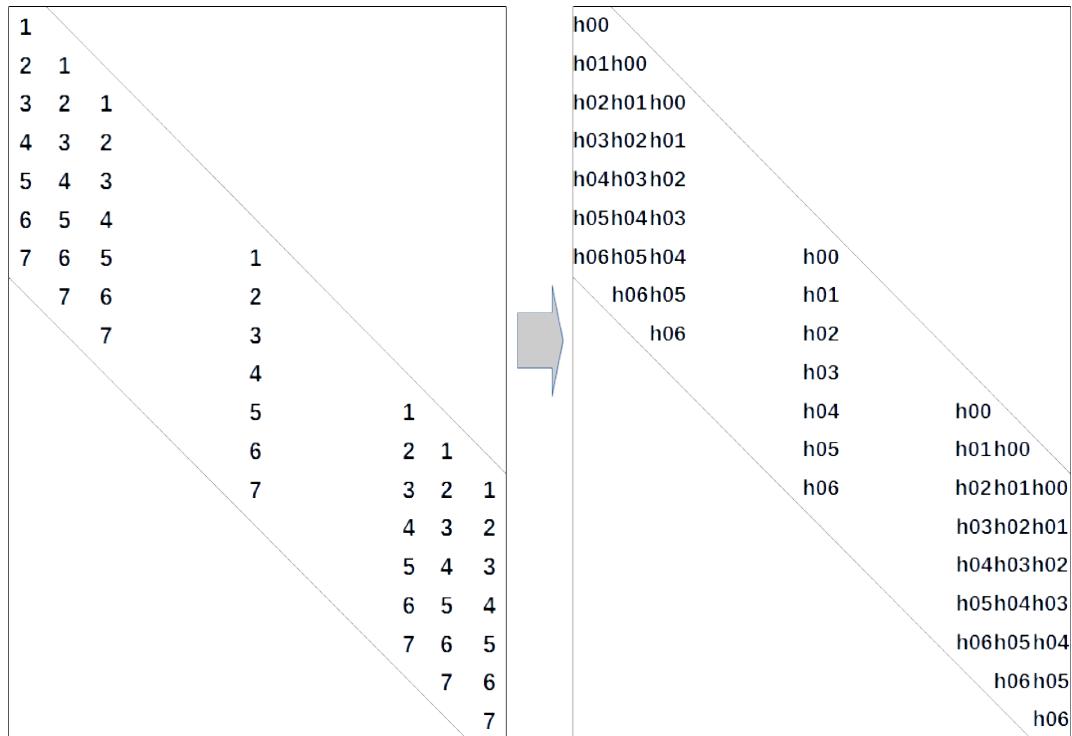
APÊNDICE E – Ilustrações da seção 6

Diagrama 13 - Circuito completo implementado



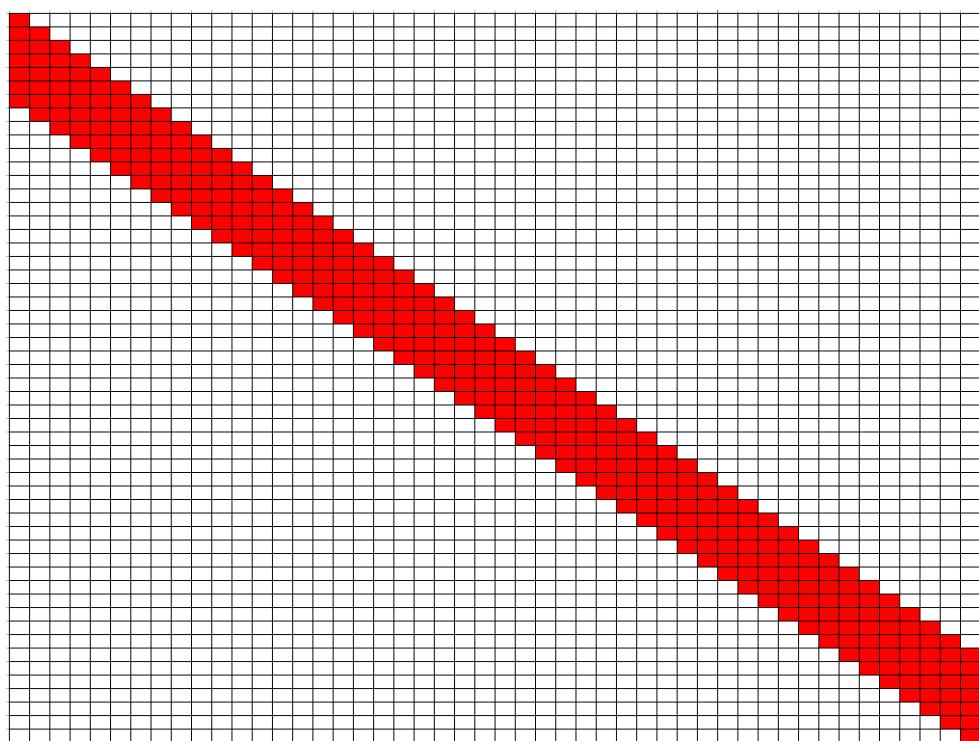
Fonte: Elaborado pelo autor (2021).

Figura 27 - Representação gráfica do mapeamento da matriz H



Fonte: Elaborada pelo autor (2021).

Figura 28 - Densidade da matriz H



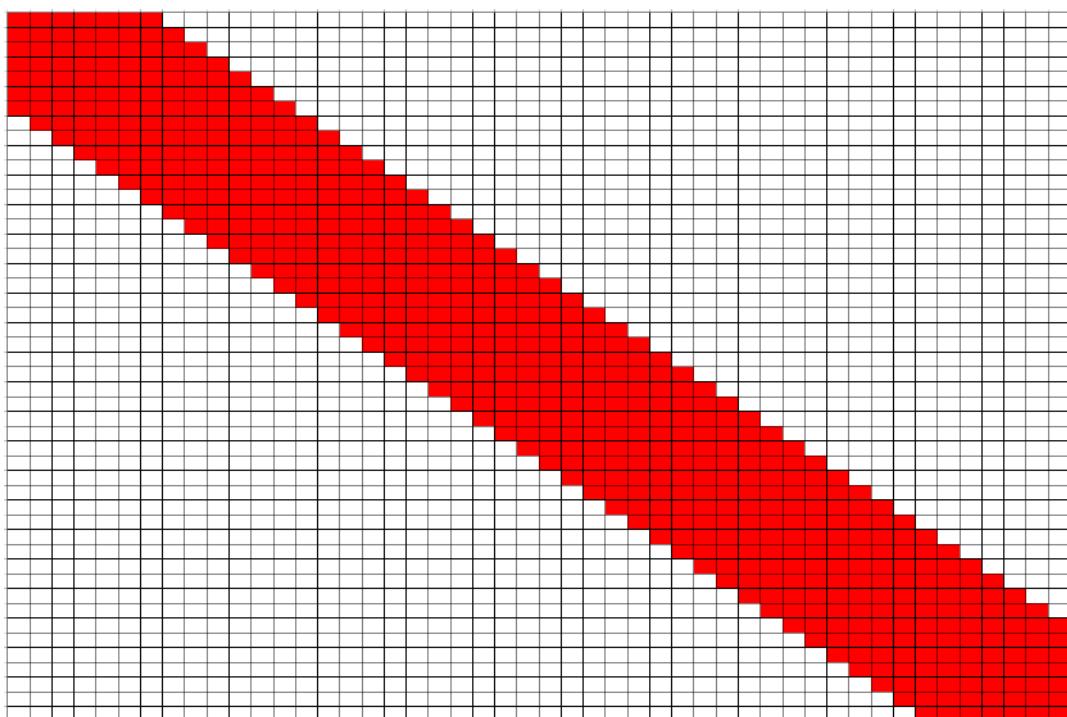
Fonte: Elaborada pelo autor (2021).

Figura 29 - Representação gráfica do mapeamento da matriz de autocorrelação A

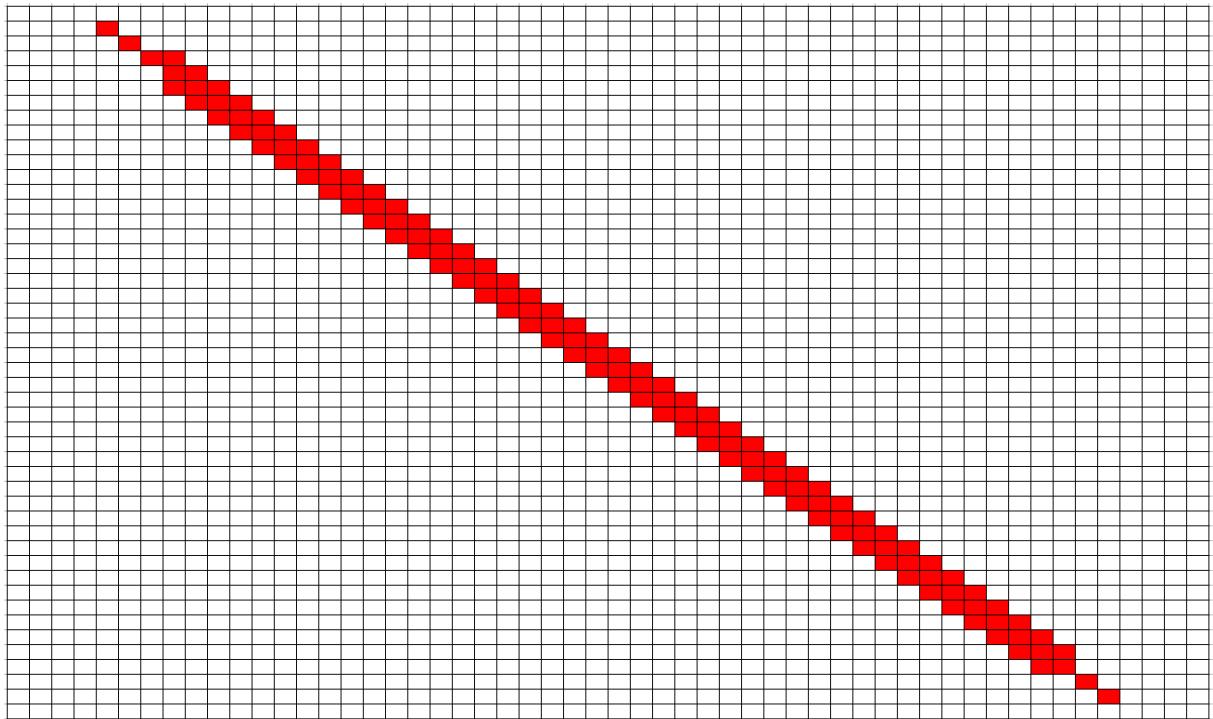
a00 a01 a02	a05 a06
a01 a00 a01	a04 a05 a06
a02 a01 a00	a03 a04 a05
a03 a02 a01	a02 a03 a04
a04 a03 a02	a01 a02 a03
a05 a04 a03	a00 a01 a02
a06 a05 a04	a01 a00 a01
a06 a05 a05	a02 a01 a00
a06	a03 a02 a01
	a04 a03 a02
	a05 a04 a03
	a06 a05 a04
	a06 a05
	a06
	a06 a05 a04
	a06 a05 a05
	a06 a01
	a02 a01 a00
	a02 a01 a00
	a03 a02 a01
	a04 a03 a02
	a05 a04 a03
	a06 a05 a04
	a06 a05 a05
	a02 a01 a00

Fonte: Elaborada pelo autor (2021).

Figura 30 - Densidade da matriz de autocorrelação A

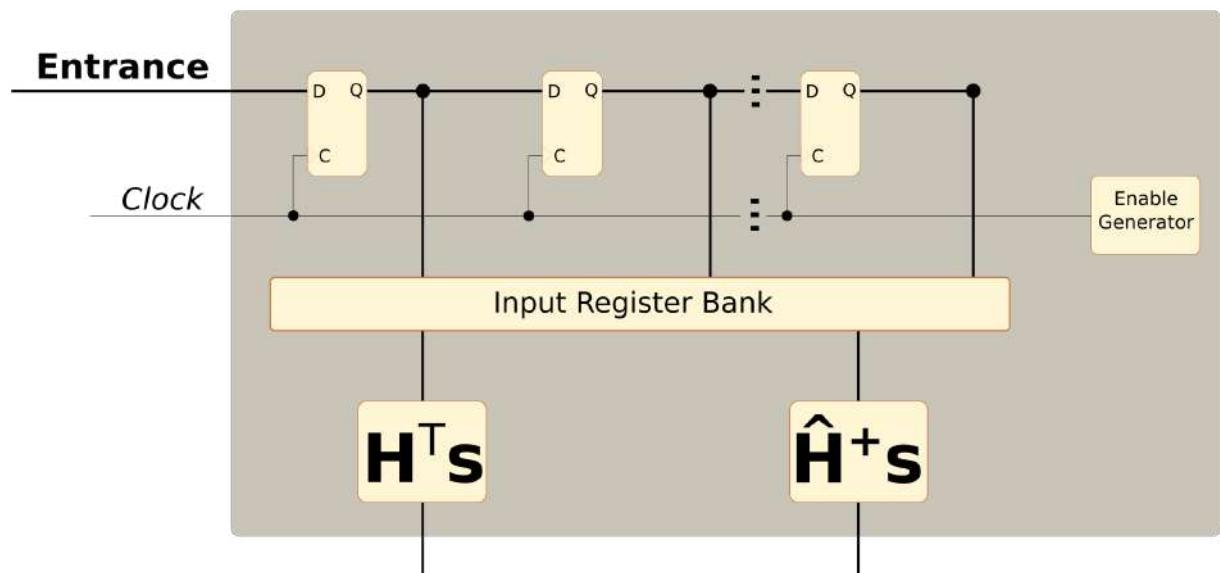


Fonte: Elaborada pelo autor (2021).

Figura 31 - Densidade da matriz \hat{H}^+ 

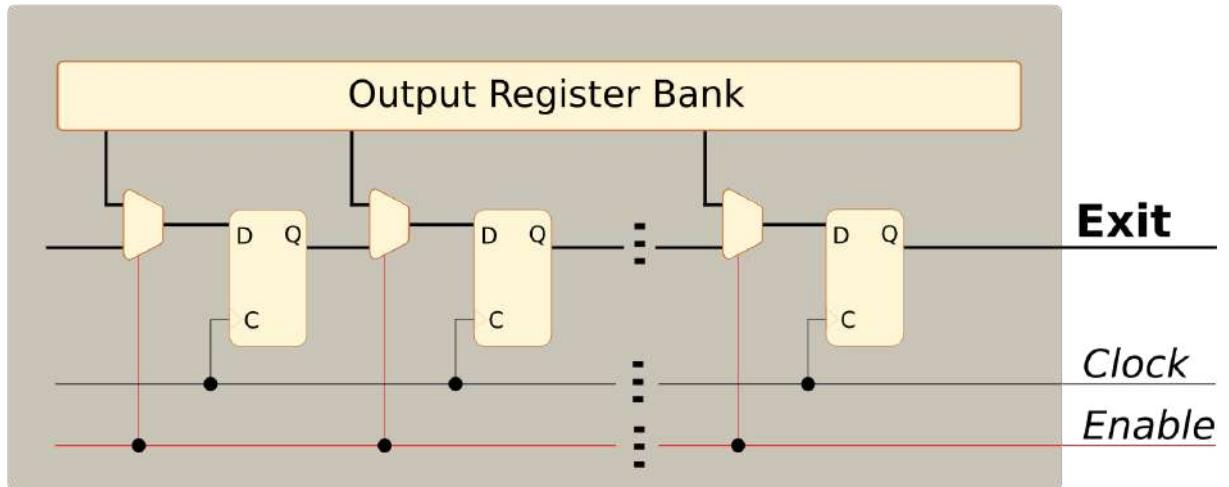
Fonte: Elaborada pelo autor (2021).

Diagrama 14 - Esquemático representativo do circuito SIPO



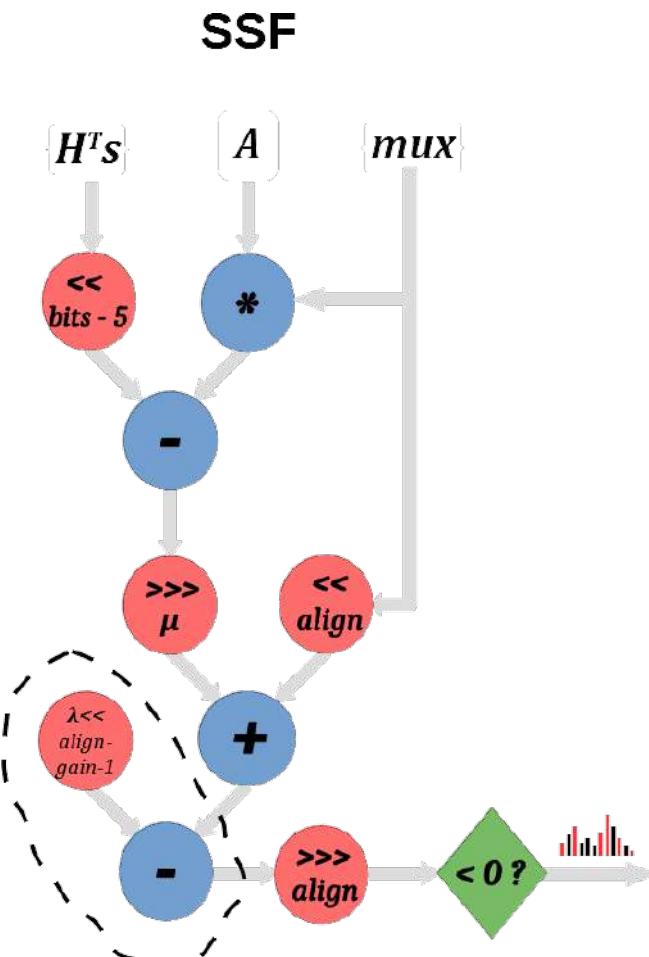
Fonte: Elaborado pelo autor (2021).

Diagrama 15 - Esquemático representativo do circuito PISO



Fonte: Elaborado pelo autor (2021).

Diagrama 16 - Core do circuito do método SSF implementados neste trabalho



Fonte: Elaborado pelo autor (2021).

APÊNDICE F – Produção Bibliográfica

F.1 Artigos Publicados em Congressos Nacionais

TEIXEIRA, T. A. *et al.* Implementação em fpga de um método recursivo de deconvolução aplicado em calorímetros operando a alta taxa de eventos. In: SBRT. *XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*. Campina Grande - PB, 2018.

Resumo: 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.

TEIXEIRA, T. A.; FILHO, L. M. A.; SEIXAS, J. M. Implementação de métodos iterativos de deconvolução para processamento on-line no calorímetro hadrônico do atlas. In: SBF. *XLII RTFNB - XL ENFPC 2019*. Campos do Jordão, 2019.

Resumo: O acelerador de partículas LHC entrou em operação em 2008 e desde então, vem passando por um processo de constantes atualizações, em que parâmetros como a energia das colisões e a quantidade de partículas por *bunch* são aumentadas. Esses aumentos vêm afetando todos os detectores, dentre eles o detector do experimento ATLAS. Nesse experimento, o sistema de *trigger* é onde os efeitos podem ser sentidos de forma mais contundente, pois há um aumento de ocorrência de colisões adjacentes, produzindo o efeito conhecido como *pile-up*, que dificultam o processo de seleção de eventos de interesse. Por isso, métodos de deconvolução *on-line* estão sendo propostos para lidar com esse problema. A maioria desses métodos são baseados em Filtros FIR, por esses já possuírem uma estrutura que permite a implementação *on-line*. Em contrapartida, o LHC opera com trens de *bunches*, onde são intercaladas uma sequência temporal com colisões e outra sequência temporal sem colisões. Tal característica pode ser interpretada como janelas de aquisição de tamanho fixo, nos permitindo o uso de métodos iterativos de deconvolução. Tais métodos tendem a apresentar resultados melhores que os métodos atualmente em uso, baseados em Filtros FIR, por ser possível o uso de conhecimento especialista e algoritmos não causais. Para tal, neste trabalho são propostos métodos iterativos de deconvolução baseados em teorias modernas de representação esparsa de dados e o seu desempenho é comparado com o desempenho dos métodos baseados em Filtro FIR. Dentre os métodos analisados destaca-se o *Separable Surrogate Functionals* (SSF), sendo assim proposta uma forma de implementação desse método em FPGA, cujos resultados apontam a possibilidade da implementação de várias instâncias desse método em paralelo, em uma única FPGA, permitindo então o processamento paralelo requerido no primeiro nível de *trigger* do ATLAS.

AGUIAR, M. S. *et al.* Implementação em fpga de um método iterativo de deconvolução para operar no sistema de trigger do experimento atlas. In: *XXII Encontro Nacional de Modelagem Computacional e X Encontro de Ciência e Tecnologia de Materiais*. Juiz de Fora: Essentia Editora, 2019.

Resumo: Aceleradores de partículas, como o LHC, em Genebra, na Suíça, colidem partículas subatômicas cujos subprodutos (partículas fundamentais da natureza) são analisados por detectores ao redor do ponto de colisão. O LHC vem passando por um processo de atualização, em que parâmetros como a energia das colisões e a quantidade de partículas por *bunch* são aumentados, impactando diretamente nos sistemas de instrumentação dos detectores, como, por exemplo, o Experimento ATLAS presente neste acelerador. Este aumento na probabilidade de ocorrência de colisões adjacentes produz um efeito conhecido como *pile-up* (empilhamento de sinais) na eletrônica de leitura dos sub-detectores do ATLAS. Para lidar com este novo desafio, métodos iterativos de deconvolução de sinais, baseados em Representação Esparsa de Dados, como o SSF, vêm sendo propostos, apresentando resultados satisfatórios. No entanto, tais métodos apresentam um alto custo computacional, de modo que o principal desafio atual é o desenvolvimento de algoritmos capazes de operar de forma online. Este trabalho apresenta um aprimoramento proposto ao SSF, permitindo reduzir a quantidade de iterações deste algoritmo em uma ordem de grandeza, facilitando, assim, a sua implementação online.

AGUIAR, M. S. *et al.* Arquitetura multi-core de processadores reconfiguráveis para reconstrução online de energia no calorímetro hadrônico do atlas. In: *XXIII Congresso Brasileiro de Automática (CBA 2020)*. Porto Alegre, Brazil: [s.n.], 2020.

Resumo: Calorímetros são sistemas usados para medir a energia de partículas fundamentais que atravessam o seu material. Para tal, pulsos elétricos são gerados por sensores posicionados ao longo do material absorvedor do calorímetro. Técnicas de processamento digital de sinais são empregadas para detectar e estimar parâmetros destes pulsos de forma a inferir a energia das partículas. Tais técnicas, quando implementadas online, necessitam ser de baixa complexidade para que seja possível a sua implementação em hardware dedicado. No entanto, técnicas baseadas em teoria de Representação Esparsa (RE) de dados vêm se destacando quanto à eficiência de reconstrução, mas, devido ao seu alto custo computacional, ainda não são utilizadas como uma opção para processamento online. Neste trabalho, é apresentada a customização de um processador e sua utilização em uma arquitetura multi-core em FPGA, possibilitando a implementação online de técnicas baseadas em RE. Para demonstrar seu funcionamento, foi utilizado o calorímetro hadrônico do Experimento ATLAS como ambiente de aplicação.

TEIXEIRA, T. A. *et al.* Método iterativo de representação esparsa implementado em FPGA para aplicação em calorimetria. In: *XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2020)*. Florianópolis, Brazil: [s.n.], 2020.

Resumo: 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 estimativa da amplitude dos sinais gerados nessas colisões não é tolerante a esse efeito, este trabalho propõe a implementação em *hardware* de um método iterativo baseado em uma variante do método *Separable Surrogate Functionals* 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 a sua implementação *online*.

F.2 Artigos Aceitos em Congressos Nacionais

TEIXEIRA, T. A.; FILHO, L. M. A.; SEIXAS, J. M. Métodos de representação esparsa implementados de forma iterativa para a realização de deconvolução de sinais obtidos com o calorímetro hadrônico do atlas. In: SBF. *XLIII RTFNB - XLI ENFPC 2021*. OnLine, 2021.

Resumo: A reconstrução de energia em calorímetros que operam em colisões de partículas que possuem uma alta luminosidade tem se tornado um grande desafio. Nesse cenário, pulsos advindos da saída frontal dos calorímetros se sobrepõem uns aos outros (efeito de *pile-up*), comprometendo a estimativa da energia quando não é realizado nenhum pré-processamento para a distribuição do sinal. Recentemente, métodos baseados na deconvolução do sinal foram propostos para reconstruções tanto *on-line* como *off-line*. Para o processamento *on-line*, restrições relativas ao processamento rápido, requisitos de memória e implementação de custos limitam o desempenho geral. A reconstrução *off-line* permite o uso da teoria da Representação Esparsa (RE) para implementar métodos sofisticados de Deconvolução Iterativa (DI). Este artigo apresenta métodos de DI baseados em algoritmos RE cujo custo computacional é efetivo para a implementação *on-line*. Usando dados de simulação, as técnicas atuais foram comparadas com as propostas de RE para validação de desempenho em ambientes *on-line*. A análise mostrou que, apesar do maior custo computacional, quando comparado aos métodos padrão, o aumento do desempenho pode justificar o uso das técnicas propostas, em particular para o Separable Surrogate Functional (SSF). que se mostra viável para implementação em FPGAs modernos.

F.3 Artigos Publicados em Periódicos

TEIXEIRA, T.; ANDRADE, L.; SEIXAS, J. de. Sparse deconvolution methods for online energy estimation in calorimeters operating in high luminosity conditions. *Journal of Instrumentation*, IOP Publishing, v. 16, n. 09, p. P09008, sep 2021. Disponível em: <<https://doi.org/10.1088/1748-0221/16/09/p09008>>.

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 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.

TEIXEIRA, T. A.; FILHO, L. M. de A. Linear equation system solver circuit for streaming processing in fpga. *Journal of Control, Automation and Electrical Systems*, oct 2021. ISSN 2195-3899. Disponível em: <<https://doi.org/10.1007/s40313-021-00842-6>>.

Abstract: The solution of Linear Equation System (LES) has numerous applications in digital signal processing. However, due to its high computational cost, it becomes difficult to implement it in high-speed free-running (streaming) processing applications. This work presents an implementation, in FPGA, of a digital circuit for LES resolution, using the Gradient Descent method, capable of operating online. Such implementation brings a new perspective of execution of iterative algorithms, based on matrix calculations, in embedded processing for high speed (tens of MHz) uninterrupted data-flow. In other words, the proposed method acts as a real-time processing block (at each clock, a new data enters and another one leaves), allowing the implementation of matrix calculations in a transparent way for continuous flow systems. In addition to proposing an implementation architecture, this work discusses the relationship between the number of iterations, the amount of resources used and the allowable acquisition rate. As a case study, the method is applied to a channel equalization problem.