

Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

**Validação do Sistema de Filtragem Online de um Detector
de Partículas de Altas Energias**

Autor:

Daniilo Enoque Ferreira de Lima

Orientador:

Prof. José Manoel de Seixas, D. Sc.

Orientador:

Denis Oliveira Damazio, D. Sc.

Examinador:

Prof. Basílio de Bragança Pereira, Ph. D.

Examinadora:

Prof^ª. Érica Ribeiro Polycarpo Macedo, D. Sc.

Examinador:

Prof. Gelson Vieira Mendonça, Ph. D.

Examinador:

Prof. Luiz Wagner Pereira Biscainho, D. Sc.

Examinador:

Rodrigo Coura Torres, M. Sc.

DEL

Abril de 2009

AGRADECIMENTO

Aos meus pais, Angela e Enoque, que sempre me apoiaram nos desvios e dúvidas no caminho; e ao meu irmão, Diogo, que, felizmente, me desviou dele muitas vezes. Aos meus bons amigos, que me ajudaram mesmo quando eu recusei o auxílio. A André e Gabi, em especial, pelas sugestões e discussões. Aos meus orientadores e amigos, Denis e Seixas, que suportaram nossas infindáveis discussões em todas as preciosas vezes em que discordamos. Aos colegas e professores, do LPS e fora dele, que contribuíram em cada passo do nosso aprendizado.

RESUMO

O Centro Europeu de Pesquisa Nuclear (CERN) tem seu foco no acelerador de partículas *Large Hadron Collider* (LHC). Seu objetivo é colidir feixes de partículas em altas energias. Vários detectores estarão prontos para medir as características das partículas após a colisão, que serão usadas para verificar a validade das novas teorias de física de partículas.

O ATLAS, maior destes detectores, entrará em operação em breve e deve ter todos seus subsistemas operando perfeitamente para os ambiciosos objetivos dos físicos que analisarão a informação obtida. O Sistema de Filtragem de Alto Nível do ATLAS foi desenvolvido em *software* para rejeitar informação irrelevante, excluindo o ruído da eletrônica e acontecimentos físicos já bem conhecidos. O presente trabalho tem como objetivo validar o funcionamento do Sistema de Filtragem de Alto Nível do ATLAS utilizando um procedimento sistemático. Tanto a verificação do funcionamento correto do *software*, quanto da performance física esperada são realizadas a cada nova versão disponível do sistema. Um sistema semi-automático de controle de qualidade também é implementado, utilizando métodos estatísticos para comparar os resultados da última versão do sistema com versões anteriores. O sistema desenvolvido envia mensagens eletrônicas mostrando a discrepância entre os resultados obtidos.

Palavras-Chave: CERN, ATLAS, física, *software*, validação, desempenho.

ABSTRACT

The European Center for Nuclear Research (CERN) has its focus on the particle accelerator Large Hadron Collider (LHC). Its objective is to collide particle beams at very high energies. Many assembled detectors measure the particles' characteristics after the collisions, which shall be used to verify the validity of the new theories of the particle physics.

ATLAS, the biggest of these detectors, will start its operations soon and it should have all of its subsystems working perfectly, so that the physicists can analyse the information correctly. The High Level Trigger System at ATLAS has been developed in software with the purpose of rejecting irrelevant information, such as electronics noise and well-known physics events.

The main goal of the present work is to validate the operation of the High Level Trigger System at ATLAS by using a systematic procedure. At each new version of the system both the software structure and the physics performance are validated. A semi-automatic system of quality control is also implemented, using statistical methods to compare the results from the last version of the system with previous ones. The developed system sends emails to the developers, showing the discrepancies between the results found.

Key-words: CERN, ATLAS, physics, software, validation, performance.

SIGLAS

ALICE - *A Large Ion Collider*
ATLAS - *A Toroidal LHC AparatuS*
ATN - *ATLAS Testing Nightly*
BNL - *Brookhaven National Laboratory*
BS - *ByteStream*
CERN - *Centro Europeu de Pesquisa Nuclear*
CMS - *Compact Muon Solenoid*
CVS - *Concurrent Versions System*
ESD - *Event Summary Data*
GCC - *GNU C Compiler*
GDB - *GNU DeBugger*
GIF - *Graphics Interchange Format*
GNU - *GNU is Not Unix*
HLT - *High Level Trigger* (Sistema de Filtragem de Alto Nível)
LAr - *Liquid Argon*
LHC - *Large Hadron Collider*
LHCb - *Large Hadron Collider beauty*
LHCf - *Large Hadron Collider forward*
NICOS - *Nightly COntrol System*
PHP - *PHP: Hypertext Processor*
PS - *Proton Synchroton*
RDO - *Raw Data Object*
RTT - *Run Time Tester*
SCT - *SemiConductor Tracker*
SPS - *Super Proton Synchroton*
SVN - *SubVersioN*
TileCal - *Tile Calorimeter* (Calorímetro de Telhas)
TOTEM - *TOTAL Elastic and diffractive cross section Measurement*
TRT - *Transistion Radiation Tracker*
XML - *Extensible Markup Language*

WWW - *World Wide Web*

Sumário

Lista de Figuras	ix
1 Introdução	1
1.1 Contexto	1
1.2 Motivação	2
1.3 Projeto desenvolvido	3
1.4 Organização do documento	4
2 Física Experimental de Altas Energias e o Detector ATLAS	5
2.1 Física das partículas elementares	7
2.2 O detector ATLAS	10
2.3 Processamento Offline	14
3 O Sistema de Filtragem do Detector ATLAS	21
3.1 Sistema de Filtragem	21
3.1.1 O canal elétron/fóton	26
3.1.2 O ruído para o canal elétron/fóton	27
3.2 Infra-estrutura do Sistema de Filtragem de Alto Nível	28
3.3 Validação do Sistema de Filtragem de Alto Nível	30
4 O projeto TrigEgammaValidation	35
4.1 Enfoque de Projeto	35
4.2 Implementação da infra-estrutura e o ambiente Run Time Tester .	37
4.3 Conjuntos de dados simulados	39
4.4 Algoritmos	41
4.4.1 Implementação em C++	43

4.5	Validação semi-automática através de métodos estatísticos	45
4.5.1	Teste de Kolmogorov-Smirnov	46
4.5.2	Divergência de Kullback-Leibler	46
4.5.3	Divergência de Jensen-Shannon	48
4.5.4	Negentropia quadrática	49
4.6	Sensibilidade das métricas	49
4.7	Resultados	52
4.7.1	Remoção de calibração fina na análise do sistema de calorimetria pelo segundo nível de filtragem	53
4.7.2	Assimetria na coordenada ϕ	57
4.7.3	Falha na detecção de um elétron	59
4.8	Obtenção de resultados via interface Web	62
4.9	Utilização da rede Grid para testes em versões estáveis do Sistema de Filtragem	67
5	Conclusões	75
	Bibliografia	77
A	Produção Científica	81

Lista de Figuras

2.1	Gráfico esquemático do LHC na fronteira franco-suíça.	6
2.2	Gráfico esquemático do detector ATLAS.	11
2.3	Sistema de coordenadas no detector ATLAS.	11
2.4	Células da seção eletromagnética do Calorímetro de Argônio Líquido no ATLAS.	13
2.5	Princípio de funcionamento do Calorímetro de Telhas no ATLAS. . .	15
2.6	Um corte transversal dos calorímetros eletromagnéticos do ATLAS. .	16
2.7	Interação das partículas com o detector ATLAS.	17
2.8	Pequena parte da infraestrutura disponibilizada no Athena.	19
3.1	Gráfico esquemático do Sistema de Filtragem do ATLAS.	23
3.2	Seção de choque e taxa de eventos detectados para a luminosidade nominal do LHC.	25
3.3	Organização dos algoritmos típicos no sistema de filtragem para o canal elétron/fóton.	31
3.4	Resultado da compilação e <i>link-edição</i> dos pacotes no projeto Atlas-Trigger.	33
4.1	O enfoque do projeto está em verificar as alterações do <i>software</i> à medida que este evolui da versão X para a versão $X + 1$	36
4.2	Estrutura geral dos trabalhos de teste do TrigEgammaValidation no RTT.	38
4.3	Estrutura de classes implementada para os algoritmos de validação em C++.	44
4.4	Efeito da adição de uma amostra gaussiana em cada uma das raias dos histograma nas métricas, para vários níveis de ruído.	50

4.5	Efeito da multiplicação de uma amostra gaussiana em cada uma das raias dos histograma nas métricas, para vários níveis de ruído.	51
4.6	Pdf da energia transversa estimada no segundo nível de filtragem, calculada pelo algoritmo de validação. A linha contínua representa a pdf após alterações e a linha tracejada representa a pdf antes das alterações.	54
4.7	Pdf do valor absoluto da coordenada η estimada no segundo nível de filtragem. Não ocorre nenhuma alteração após mudanças.	55
4.8	Histograma da massa invariante do bóson Z reconstruída utilizando apenas a informação de energia, η e ϕ do primeiro passo do segundo nível de filtragem.	56
4.9	Desvio relativo entre a variável ϕ estimada pelo algoritmo T2CaloEgamma_eGamma e a coordenada ϕ real antes da alteração.	58
4.10	Desvio relativo entre a variável ϕ estimada pelo algoritmo T2CaloEgamma_eGamma e a coordenada ϕ real após a alteração.	59
4.11	Eficiência em relação à informação da simulação, após a filtragem pelo segundo nível (apenas passo que utiliza informação de calorimetria). Efeito da mudança aplicada em ϕ	60
4.12	Massa invariante do bóson Z reconstruída com a informação do Filtro de Eventos. A linha contínua (sempre em zero) mostra o histograma após a alteração. A linha tracejada mostra o histograma antes da alteração. Foi assumida a hipótese de o decaimento ocorrer no centro do detector ATLAS.	61
4.13	Momento transverso estimado por TrigL2IDSCAN_eGamma	62
4.14	Histograma de referência (antes da alteração) do momento transverso estimado pelo algoritmo TrigL2IDSCAN_eGamma , no nível 2.	63
4.15	Interface Web mostrando histogramas gerados para conjuntos simulados de Z decaindo em um par elétron-pósitron após a filtragem do nível 2.	65
4.16	Interface Web mostrando curvas de eficiências medidas para conjuntos simulados de Z decaindo em um par elétron-pósitron após a filtragem do nível 2.	66

4.17 Interface Web mostrando o resultado da comparação entre histogramas para conjuntos simulados de Z decaindo em um par elétron-pósitron.	68
4.18 Envio de trabalhos à rede Grid através do Pathena.	70
4.19 Envio de trabalhos e pós-processamento através do Psequencer.	70
4.20 Envio de trabalhos e pós-processamento através do Psequencer para o TrigEgammaValidation.	71
4.21 Interface Web mostrando os histogramas gerados no Grid para conjuntos de elétrons puros.	73
4.22 Interface Web mostrando o gráfico de eficiência gerado no Grid para conjuntos de elétrons puros.	74

Capítulo 1

Introdução

Neste capítulo, o propósito básico do projeto e uma visão geral de seu contexto são apresentadas de forma bastante concisa. Descreve-se também o projeto em linhas gerais e a organização deste documento.

1.1 Contexto

Em aplicações onde o desenvolvimento de *software* tem um papel fundamental, ter uma boa visão global do sistema é importante para a gerência do projeto, permitindo que todos os seus componentes realizem as tarefas desejadas. Este aspecto é intensificado durante o desenvolvimento de sistemas de média e larga escala, onde o número de desenvolvedores e a complexidade do projeto podem resultar em efeitos inesperados. Em sistemas de maior escala, a complexidade é, frequentemente, incrementada pela relação entre sub-módulos, particularmente em um sistema multi-propósito com um grande número de componentes.

No caso de um sistema como o descrito anteriormente, a melhor forma de acompanhar a evolução do projeto para impedir falhas de forma eficiente é através de uma metodologia de validação sistemática. Isto é, deve-se verificar cada nova versão do projeto com o objetivo de validar as características mais essenciais do *software*.

Apesar de a validação realizada pelos desenvolvedores ser crucial, pode-se otimizar o processo de detecção de falhas utilizando métodos semi-automáticos. Neste caso, utiliza-se um sistema de validação que verifica características importantes do

projeto em cada nova versão desenvolvida, informando aos desenvolvedores caso um comportamento inesperado seja encontrado. Mesmo neste caso, a interação com os desenvolvedores e peritos do projeto é importante para detectar exatamente a causa do problema e corrigi-lo.

1.2 Motivação

O acelerador de partículas *Large Hadron Collider* (LHC), está localizado na fronteira entre a França e a Suíça, e é um dos principais focos do Centro Europeu de Pesquisas Físicas (CERN). Neste acelerador, feixes de prótons serão acelerados em direções opostas, colidindo em certos pontos onde há detectores prontos para medir o resultado das colisões.

Os detectores, projetados com o intuito de medir os resultados das colisões no LHC, têm o propósito de comprovar ou rejeitar, através de suas medidas, as novas teorias da física de partículas. O maior destes detectores é o ATLAS (*A Toroidal LHC Apparatus*), cujo propósito principal é determinar a existência e a massa de uma partícula denominada bóson de Higgs.

Devido à baixa probabilidade de ocorrência do fenômeno de interesse, é necessário colidir uma grande quantidade de prótons com alta frequência, visando à acumular uma estatística razoável. Pelo mesmo motivo, uma boa parte da informação obtida será pouco relevante para as análises físicas. Há também uma quantidade significativa de ruído proveniente do sistema de aquisição de dados.

Tendo em vista que a taxa de dados produzida no detector ATLAS é bastante grande, o espaço necessário para armazená-los em mídia permanente seria insustentável com as tecnologias atuais. Mesmo se fosse possível armazenar a informação adquirida, também não há tecnologia disponível para processar todos os dados obtidos em detalhes.

Um Sistema de Filtragem *Online* foi desenvolvido com o objetivo de descartar dados que não têm interesse na análise física. Este sistema é dividido em três níveis, dos quais o primeiro é implementado em *hardware* e os níveis seguintes são implementados em *software*. A parcela do Sistema de Filtragem implementada em *software* denomina-se Sistema de Filtragem de Alto Nível.

O *software* do Sistema de Filtragem de Alto Nível é implementado por variados grupos de desenvolvedores com conhecimentos e culturas diferentes, em um ambiente multinacional. O número de desenvolvedores do Sistema de Filtragem de Alto Nível é também bastante grande, uma vez que o ATLAS é um projeto de larga escala (36 países, 1800 pesquisadores). Além dos próprios detalhes de funcionamento dos três níveis do Sistema de Filtragem, há vários subsistemas que devem operar integrados de forma consistente.

1.3 Projeto desenvolvido

Tendo em vista a dimensão e complexidade do Sistema de Filtragem de Alto Nível, será utilizado um sistema de validação semi-automático para rastrear as mudanças na implementação do *software*, detectando erros de forma mais eficiente e sistemática.

Neste projeto, foi desenvolvido um módulo que pode ser posto em execução agregado ao *software* do Sistema de Filtragem. Pode-se utilizá-lo para medir características cruciais das análises, servindo como indicadores do desempenho do sistema que se deseja validar.

As características analisadas são constituídas, principalmente, pelas variáveis de decisão do sistema de filtragem. O sistema de filtragem utiliza tais variáveis para determinar se o evento deve ser armazenado, o que torna estas grandezas extremamente importantes para seu funcionamento.

Para alertar os desenvolvedores de forma semiautomática, um conjunto de métricas estatísticas foram utilizadas para comparar a informação obtida através destes indicadores com dados análogos obtidos em uma versão anterior do *software*. Após este processamento, mensagens eletrônicas são enviadas à equipe de validação com o resultado dos testes, para que uma análise mais detalhada seja realizada.

Também foi desenvolvido, como parte deste conjunto de ferramentas de validação, uma interface *Web* que permite o acesso mais fácil e eficiente aos resultados.

1.4 Organização do documento

Este capítulo introduz o projeto de forma concisa, mostrando os problemas que se deseja resolver, o contexto em que este se insere, a proposta de projeto e os detalhes da organização deste documento.

O Capítulo 2 explica, em linhas gerais, os aspectos importantes da física envolvida e o funcionamento do detector ATLAS, bem como o processamento realizado com os dados obtidos.

O Capítulo 3 entra em mais detalhes sobre o funcionamento do sistema de filtragem. Alguns detalhes específicos das análises utilizadas na detecção de elétrons e fótons são descritos. Também discutem-se alguns detalhes da infra-estrutura do sistema de filtragem e os métodos comuns de validação.

O Capítulo 4 discute a implementação deste projeto. A implementação e a infra-estrutura utilizada no mesmo é descrita, além de alguns resultados e testes. Mostra-se, também, uma forma mais acessível de apresentar os resultados, desenvolvida neste projeto, além de uma extensão do projeto para um ambiente de processamento distribuído.

O Capítulo 5 resume os aspectos discutidos neste documento, e o Apêndice A mostra a produção científica realizada durante o desenvolvimento do projeto.

Capítulo 2

Física Experimental de Altas Energias e o Detector ATLAS

O Centro Europeu de Pesquisa Nuclear (CERN) [1] é um dos maiores laboratórios de pesquisa em física de partículas. O atual foco do CERN é o acelerador circular *Large Hadron Collider* (LHC) [2], localizado a aproximadamente 100 m abaixo da superfície, na fronteira entre França e Suíça.

A ideia dos experimentos é colidir dois feixes de prótons e analisar os resultados das colisões. Cada feixe será acelerado no *Proton Synchrotron* (PS) e enviado, em seguida, para o *Super Proton Synchrotron* (SPS). O LHC receberá as partículas com uma energia inicial e as acelerará em seu túnel de 27 km de extensão até atingirem uma velocidade suficientemente grande. Após tal aceleração, os feixes serão desviados por campos magnéticos para entrarem em colisão em um dos pontos no LHC, onde há detectores prontos para medir informações sobre os acontecimentos físicos resultantes do processo. As quatro maiores experiências no LHC são o ATLAS¹, o CMS², o ALICE³, o LHCb⁴. Os detectores de menor escala são o TOTEM⁵ e o LHCf⁶. Um gráfico esquemático do LHC está representado na figura 2.1.

¹A Toroidal LHC ApparatuS

²Compact Muon Solenoid

³A Large Ion Collider

⁴Large Hadron Collider beauty

⁵TOTAL Elastic and diffractive cross section Measurement

⁶Large Hadron Collider forward

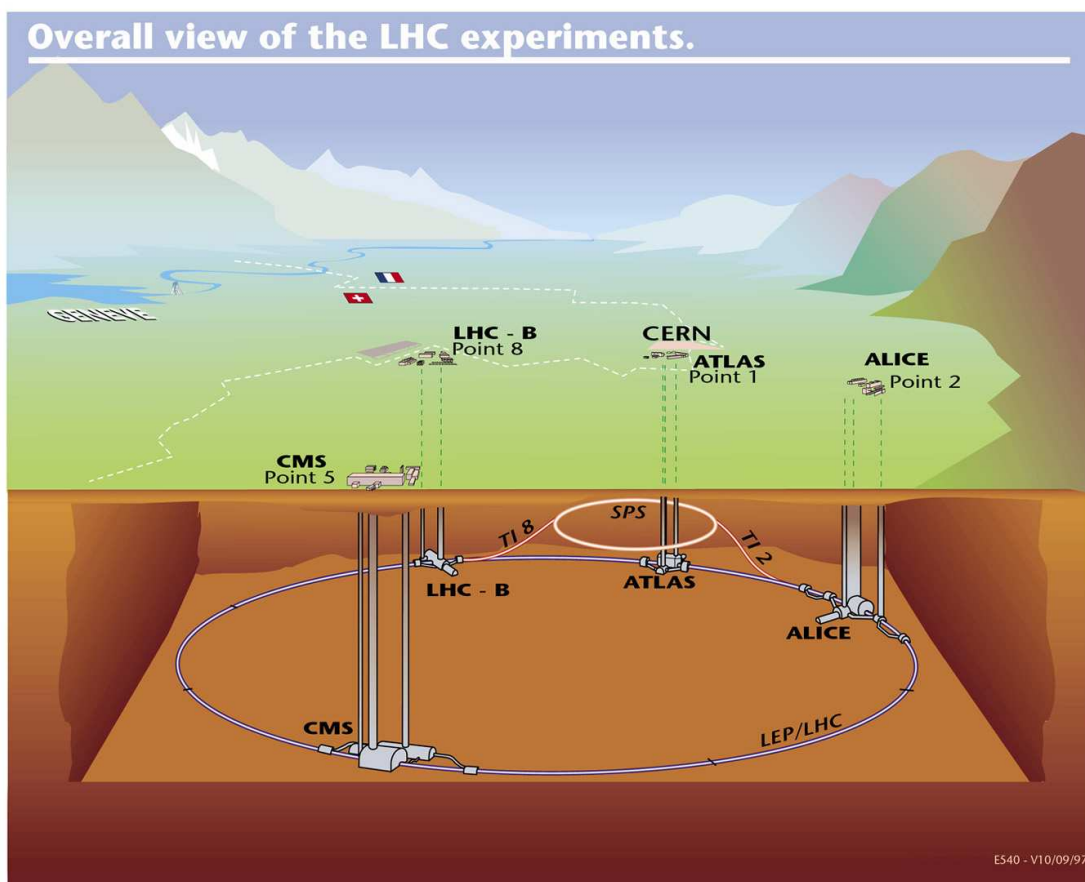


Figura 2.1: Gráfico esquemático do LHC na fronteira franco-suíça. Extraído de [3].

A cada 25 ns ocorre um evento físico, no qual as características das partículas resultantes da colisão devem ser estimadas da forma mais acurada possível. A **luminosidade** dos feixes no LHC, isto é, o fluxo de partículas⁷, é $1 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ [4]. Com esta luminosidade máxima, foi previsto que, a cada colisão de feixes (isto é, a cada 25 ns), em média, 23 partículas serão geradas e medidas pelos detectores.

Este projeto concentra-se no ATLAS [4] [5], o maior detector do LHC, onde espera-se encontrar evidências para comprovar ou invalidar as novas teorias da física de partículas. Algumas dessas teorias prevêm a existência de uma nova partícula denominada **bóson de Higgs**, que, se existir, será observada muito raramente.

2.1 Física das partículas elementares

Antes de examinar detalhadamente o funcionamento do ATLAS, deve-se entender seus objetivos mais nitidamente. Uma compreensão da física [6] sendo estudada e da história que a levou ao atual estudo das propriedades da matéria é, portanto, apropriada.

A mecânica clássica de Newton serve bem ao seu propósito para a descrição dos movimentos de corpos, assim como a eletrodinâmica, com as equações de Maxwell, descreve muito bem os campos eletromagnéticos e suas interações com a matéria. No século XX, porém, ambas teorias foram desafiadas em pontos distintos, quando novas experiências mostraram que estas eram incompletas.

As primeiras tentativas para descrever a matéria levaram ao modelo atômico de Bohr, que era inconsistente com a descrição dada pela mecânica newtoniana para o movimento das partículas. De acordo com este modelo, a matéria é constituída de átomos, que, por sua vez, contêm um núcleo maciço com prótons e nêutrons, além de elétrons orbitando ao redor do núcleo através de uma interação eletromagnética. A descrição clássica para o movimento das partículas previa que o elétron colidiria com o núcleo em poucos instantes, levando ao colapso do átomo.

Em paralelo, a Teoria Eletromagnética também era inconsistente com a mecânica clássica de Newton, uma vez que as equações de Maxwell mostram que a velocidade de propagação das ondas eletromagnéticas é a mesma em todos os referenciais.

⁷Número de partículas atravessando o detector por unidade de área, por unidade de tempo.

Ambas as inconsistências foram removidas com as novas teorias desenvolvidas. A Teoria da Relatividade Restrita de Einstein modifica as leis de Newton para que estejam de acordo com a Eletrodinâmica Clássica, e seu sucesso experimental e teórico foi verificado em pouco tempo. Defende-se que a velocidade de propagação das ondas eletromagnéticas é a mesma em todos os referenciais. Pode-se mostrar que as leis da Mecânica Relativística reduzem-se à Mecânica Clássica de Newton quando se observam partículas movendo-se em velocidades muito inferiores à da luz e, ainda assim, mantêm-se compatíveis com os experimentos e com a Teoria Eletromagnética Clássica.

Ainda de acordo com a Teoria da Relatividade Restrita, massa e energia são formas diferentes de se observar as mesmas grandezas físicas. Desta forma, uma partícula pode transformar sua energia em massa e vice-versa, gerando novas partículas, mas obedecendo sempre o princípio de Conservação de Energia, já que a massa da nova partícula gerada é apenas uma forma de energia.

Em paralelo, a Mecânica Quântica foi criada especificamente para descrever a interação das partículas no interior do átomo, explicando acuradamente os níveis de energia dos elétrons nos átomos e os vários experimentos realizados com partículas elementares. Tal teoria reduz-se à Mecânica Newtoniana quando a massa e dimensão das partículas são muito maiores que as massas e dimensões contidas, por exemplo, no átomo.

Apesar de duas descrições satisfatórias para as novas evidências terem sido encontradas, a Teoria da Relatividade Restrita e a Mecânica Quântica eram descorrelacionadas. Várias tentativas surgiram para unir as duas idéias, à medida que novos experimentos observam diferentes propriedades das partículas e campos.

O Modelo Padrão da Física de Partículas [7] [8] é o modelo mais aceito para unificar as idéias existentes e descrever as partículas. De acordo com este modelo, há um conjunto de partículas elementares e a interação entre estas é descrita através de trocas de bósons, cujo propósito é descrever os campos das diversas forças de interação.

A Tabela 2.1 mostra as partículas elementares previstas pelo Modelo Padrão.

Além destas partículas, também foi verificada a existência das respectivas anti-partículas, isto é, partículas que têm todas as características em comum com

Tabela 2.1: Partículas elementares previstas no Modelo Padrão

Três gerações de férmions				Bósons	
	I	II	III	Força e distância de interação	
Quarks	u up	c charm	t top	γ fóton	Eletromagnetismo (∞)
	d down	s strange	b bottom	g glúons	Interações fortes ($10^{-15}m$)
Léptons	e elétron	μ múon	τ táon	Z^0 Z⁰	Interações fracas ($10^{-18}m$)
	ν_e neutrino do elétron	ν_μ neutrino do múon	ν_τ neutrino do táon	W^\pm W⁺/W⁻	

estas, mas com números quânticos, como carga, invertidos. Portanto há também o *anti-up*, *anti-down*, etc. Por motivos históricos, a anti-partícula do elétron denomina-se pósitron.

A interação entre os *quarks* e léptons dá-se através de trocas de partículas mediadoras, ou bósons. O Modelo Padrão descreve três formas de interação: a força eletromagnética, a força forte e a força fraca. O fóton é o mediador de interação eletromagnética; os glúons mediam a interação forte; o Z , W^+ e W^- mediam as interações fracas.

Não se observam os *quarks* isoladamente, pois eles têm um tempo de vida muito curto e tendem a decair rapidamente em elementos mais estáveis. Nesta situação, a energia dos *quarks* é transferida para as novas partículas, algo possibilitado pela Teoria da Relatividade Restrita, que estabelece que massa é apenas uma forma de energia. Os léptons, por outro lado, são mais estáveis, particularmente o elétron, encontrado nos átomos. Apesar de instáveis isoladamente, pode-se encontrar os quarks agrupados. O próton, por exemplo, é constituído de dois *quarks up* e um *quark down*.

O Modelo Padrão inclui o mecanismo de Higgs para explicar como as partículas ganham massa. Esse mecanismo, no entanto, exige a existência de uma partícula ainda não verificada experimentalmente: o **bóson de Higgs**.

Há várias outras questões que o Modelo Padrão não consegue explicar de primeiros princípios. Outras teorias, ao tentar explicar essas questões, introduzem a existência de novos bósons de Higgs e ainda novas partículas. No Modelo Padrão, no entanto, as massas são parâmetros que devem ser medidos experimentalmente. Não há uma explicação mais fundamental para a enorme diferença encontrada entre as massas das diferentes partículas.

Desta forma, uma medida da massa do bóson de Higgs é essencial para determinar quais das teorias estão incorretas, mantendo apenas as teorias que explicam todos os detalhes verificados experimentalmente. É importante notar que o bóson de Higgs tem um tempo de vida muito curto e rapidamente decai em outras partículas, de forma que sua detecção é apenas possível de forma indireta.

2.2 O detector ATLAS

O ATLAS é um detector cilíndrico projetado para cobrir a maior área possível. Os feixes de partículas atravessam seu eixo de simetria, colidindo em seu centro. A Figura 2.2 mostra um desenho esquemático do ATLAS e de seus subdetectores. A parte central do detector é denominada **barril** e os discos que limitam suas extremidades são denominados **tampas**. Os vários subsistemas, posicionados no barril do detector ou nas tampas, medem as características das partículas que emergem dos eventos de colisão.

Para identificar posições no ATLAS, foi criado um sistema de coordenadas próprio com origem no ponto de colisão dos feixes. A direção do feixe é o eixo z , e o plano $x - y$ é o plano transversal à direção do feixe, com o sentido positivo de x apontando para o centro do anel do LHC e o sentido positivo do eixo y apontando para cima. A Figura 2.3 mostra o sistema de coordenadas no detector.

O ângulo azimutal ϕ é medido ao redor do eixo z , no plano $x-y$ e o ângulo polar θ é medido a partir do eixo z , no plano $z-y$. Costuma-se usar a **pseudo-rapidez** em vez do ângulo polar, que é definida como $\eta = -\ln(\text{tg}(\frac{\theta}{2}))$. A distância no espaço

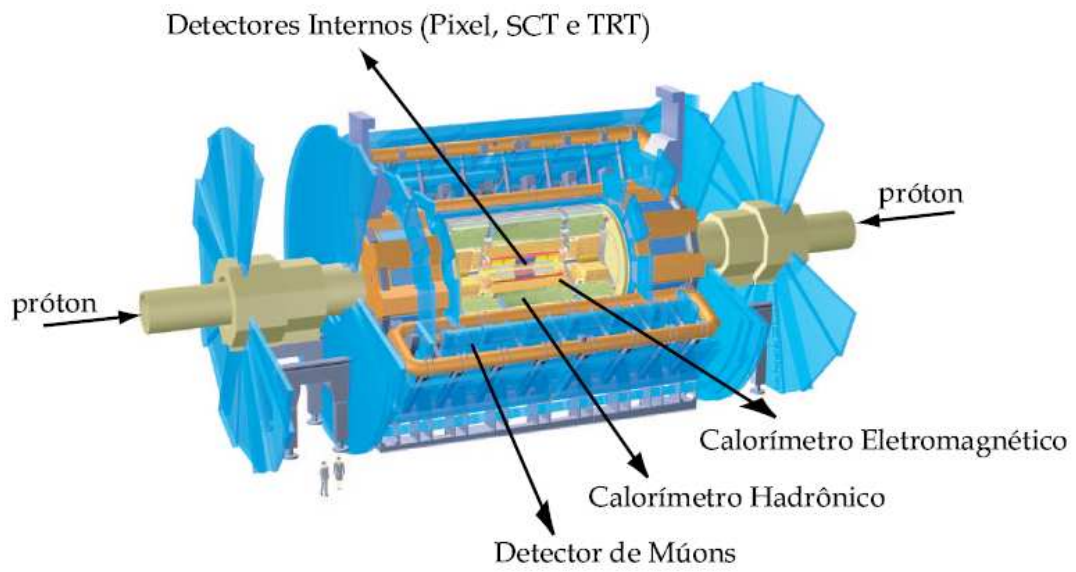


Figura 2.2: Gráfico esquemático do detector ATLAS. Extraído de [9].

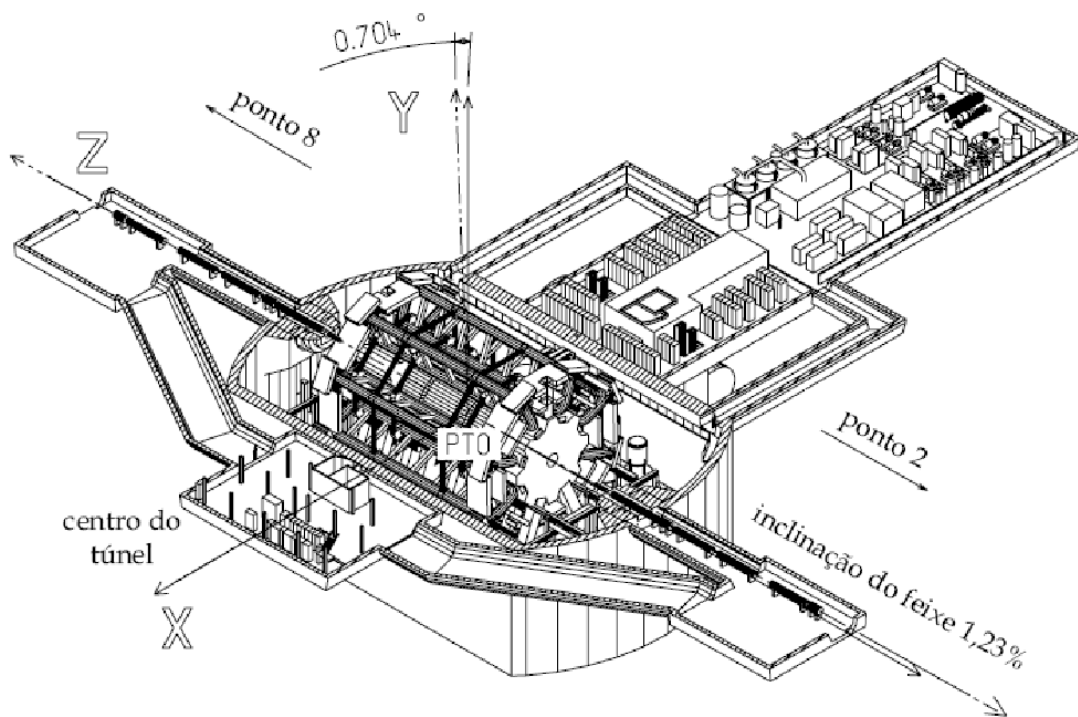


Figura 2.3: Sistema de coordenadas no detector ATLAS. Extraído de [9].

angular $\eta \times \phi$ é definida como $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$. Frequentemente, utilizam-se projeções de quantidades físicas no plano $x-y$. São exemplos, a energia e momento estimados de partículas detectadas. Costuma-se utilizar a energia transversa e o momento transverso, que são apenas as projeções destas quantidades no plano $x-y$.

O Detector Interno é o primeiro subsistema a interagir com as partículas provenientes da colisão, e tem como objetivo reconstruir a trajetória das partículas que atravessam o detector. O Detector Interno é composto pelo Detector de Píxeis (*Pixel Detector*), o SCT (*SemiConductor Tracker*) e o TRT (*Transition Radiation Tracker*) e está imerso em um campo magnético de 2 Tesla, gerado pelo Solenóide Central. Os subdetectores do Detector Interno medem pontos por onde as partículas passam e tentam reconstruir suas trajetórias. Devido ao campo magnético, as partículas carregadas têm uma trajetória curva, permitindo realizar medidas de suas cargas elétricas, momentos e os pontos de onde as partículas emergem, após a colisão, isto é, seus **vértices**.

Para reconstruir os momentos e os traços das partículas com alta resolução, são necessárias muitas medidas de pontos nas trajetórias. O SCT é um detector de traços baseado em semicondutores que utiliza pastilhas e pontos (píxeis) microscópicos de silício, fornecendo uma boa medida dos vértices das partículas. O TRT proporciona uma detecção de vértices com um menor custo por ponto e, combinado com o SCT, fornece uma boa resolução na medida da trajetória

Os calorímetros são os próximos detectores a interagirem com as partículas, e têm como propósito medir a energia destas. Há dois sistemas de calorimetria no ATLAS: o Calorímetro de Argônio Líquido [10] e o Calorímetro de Telhas [11]. O Calorímetro de Argônio Líquido é utilizado no barril e nas tampas. No barril, apenas como calorímetro eletromagnético e nas tampas para detectar elétrons, fótons e hádrons. A Figura 2.4 mostra a estrutura da parte eletromagnética do calorímetro de Argônio Líquido. Já o Calorímetro de Telhas cobre apenas o barril e mede a energia de partículas hadrônicas, como nêutron, prótons e píons.

Os Calorímetros de Argônio Líquido utilizam Argônio Líquido como material absorvedor e eletrodos de chumbo para captar íons formados durante a interação da partícula com o detector. Os eletrodos estão disponíveis em forma de acordeão, com o objetivo de cobrir maximamente as partículas em todas a faixa em ϕ .

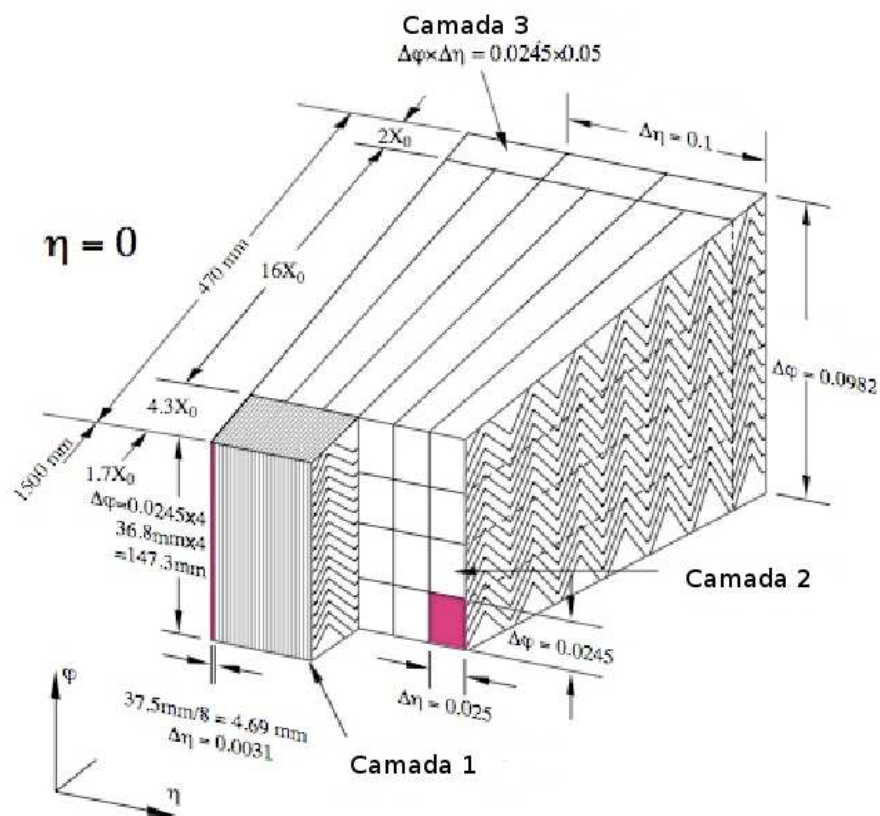


Figura 2.4: Células da seção eletromagnética do Calorímetro de Argônio Líquido no ATLAS. Extraído de [10].

O Calorímetro de Telhas utiliza uma liga de aço como material absorvedor e telhas cintilantes para amostrar a energia da cascata gerada. A Figura 2.5 mostra o princípio de funcionamento do Calorímetro de Telhas. O sinal luminoso gerado nas telhas cintilantes é convertido em sinal elétrico através de fotomultiplicadores e, a seguir, é enviado para a eletrônica de leitura para o processamento seguinte. No caso do Calorímetro de Telhas, ocorre dupla leitura do sinal luminoso, criando uma redundância do sinal, muito útil em caso de falhas na leitura devido a problemas ou envelhecimento.

A Figura 2.6 mostra um corte transversal nos calorímetros eletromagnéticos do ATLAS. Nota-se que há uma parte, em $\eta \approx 1,3$, entre o barril e o calorímetro eletromagnético da tampa, que está descoberta. Esta região é necessária para que os cabos dos Detectores Internos possam passar. Há cintiladores que estimam a energia perdida na medida desta região.

Externo a ambos calorímetros, está o Espectrômetro de Múons, imerso em um campo magnético e desenvolvido para medir o momento dos múons que não interagiram com os outros detectores. Seu funcionamento baseia-se na deflexão de múons em toróides com núcleo de ar e um detector de traços bastante preciso.

As partículas eletromagnéticas incluem elétrons, pósitrons e fótons, e são extremamente importantes para as análises físicas. Partículas hadrônicas compreendem todas as partículas que interagem através de força forte. A distinção entre tais partículas tem importância fundamental nos algoritmos utilizados na análise física. A Figura 2.7 mostra quais partículas interagem com quais detectores no ATLAS. Essa informação é bastante útil para a discriminação das partículas detectadas.

2.3 Processamento Offline

Os sistemas do ATLAS foram desenvolvidos visando a reconstruir todas as características das partículas criadas nas colisões. A tarefa mais importante após a aquisição dos sinais provenientes de todos os subsistemas é, portanto, a **reconstrução** das características físicas de todas as partículas detectadas. Todas as partículas devem ser identificadas e deve-se extrair, para cada uma, suas características cinemáticas.

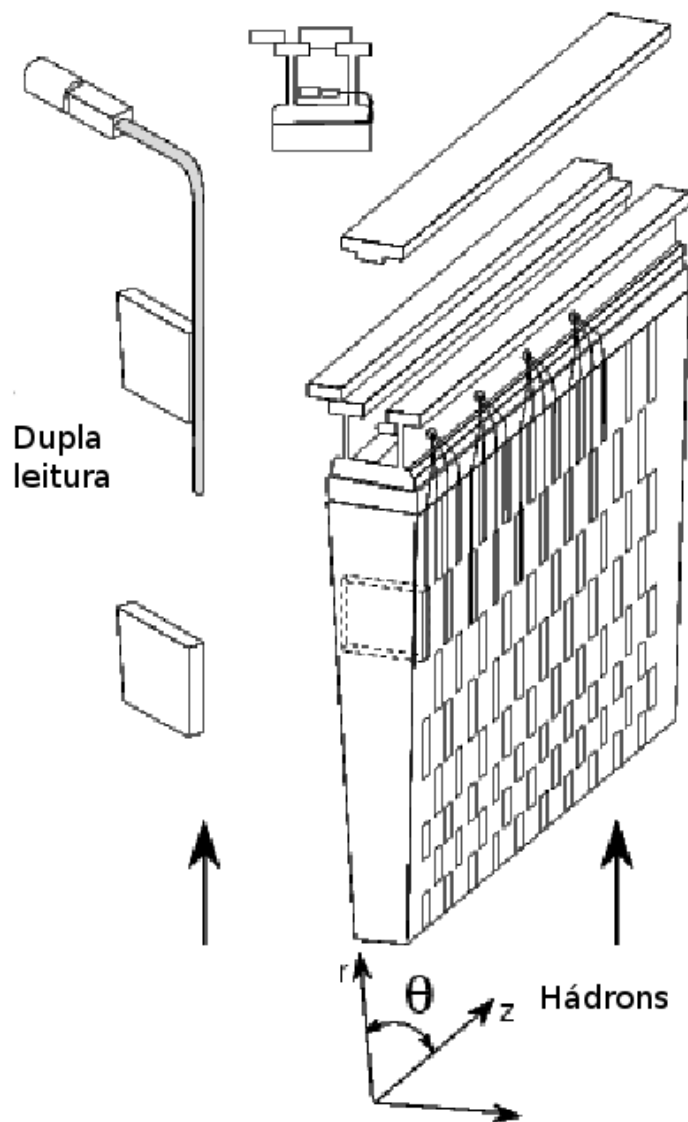


Figura 2.5: Princípio de funcionamento do Calorímetro de Telhas no ATLAS. Extraído de [11].

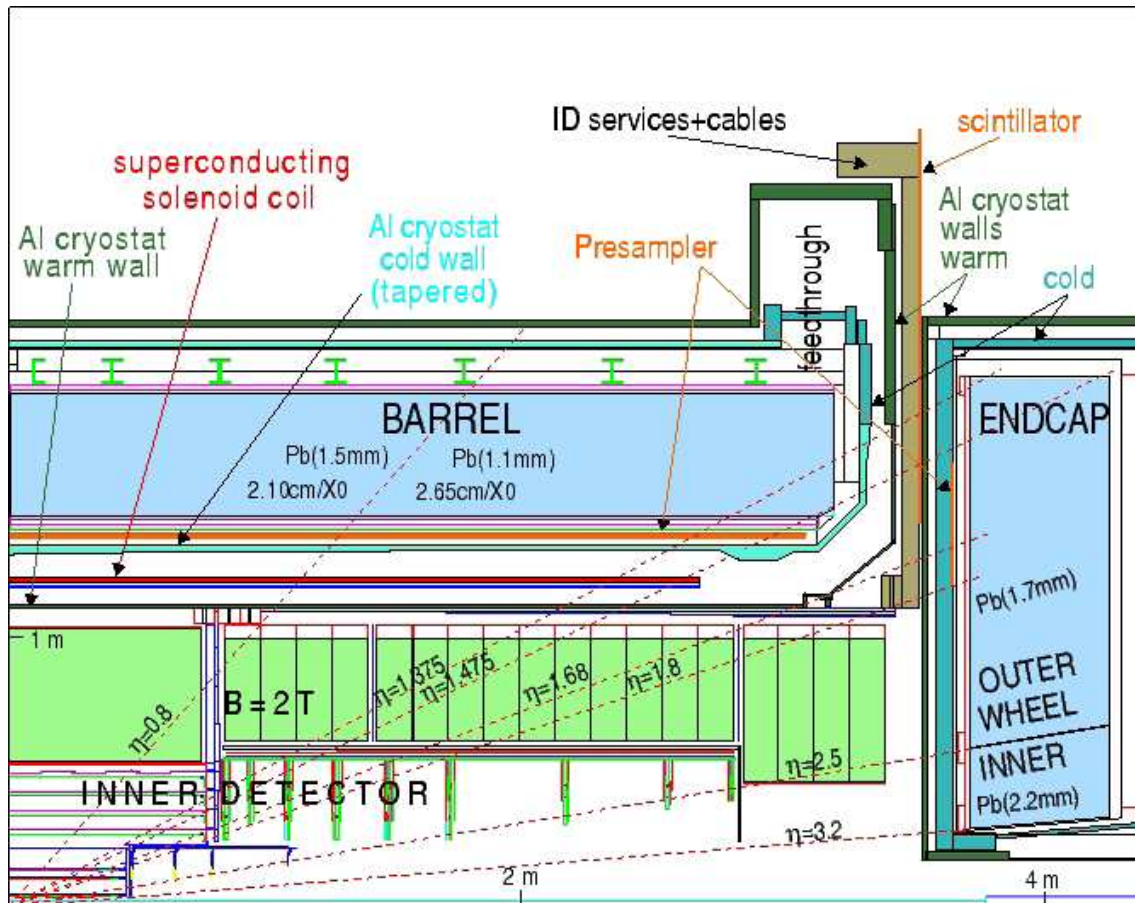


Figura 2.6: Um corte transversal dos calorímetros eletromagnéticos do ATLAS. Extraído de [12].

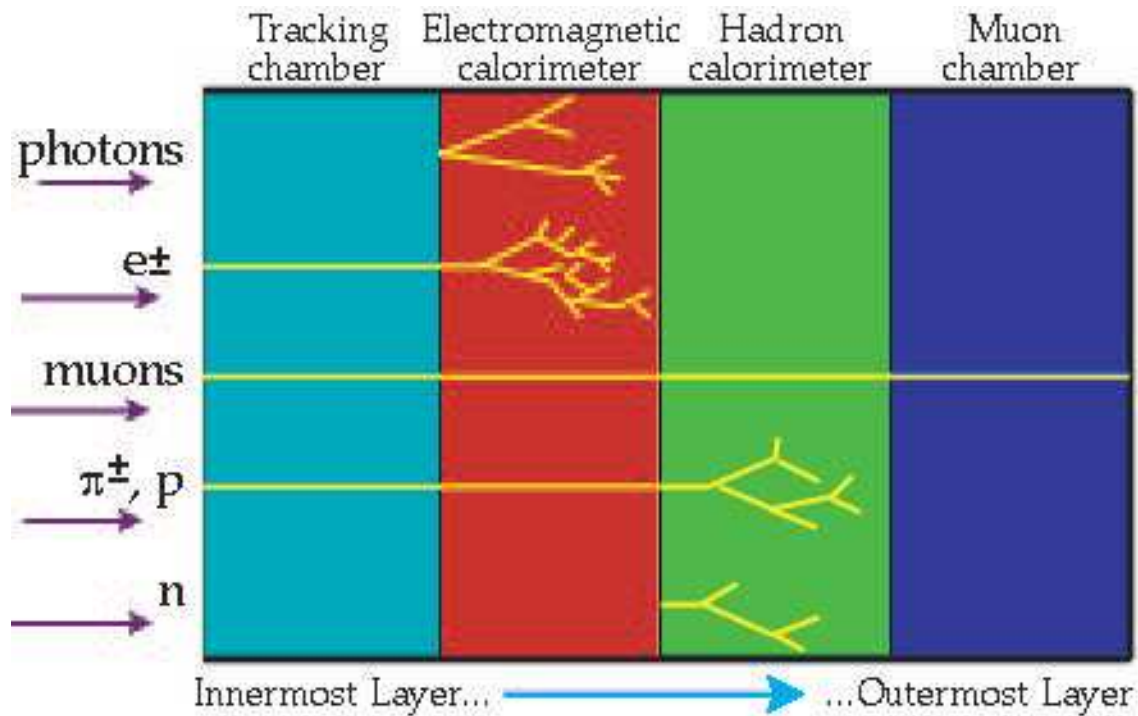


Figura 2.7: Interação das partículas com o detector ATLAS. Extraído de [9].

Após a identificação de cada partícula, vários pesquisadores tentarão desvendar os processos pelos quais tais partículas foram geradas antes de sua detecção. A informação que se deseja encontrar para comprovar ou rejeitar as novas teorias não está nas medidas diretas das partículas que interagem com o detector, mas nas estimativas indiretas do mecanismo pelo qual tais partículas foram geradas. O bóson de Higgs, por exemplo, tem um tempo de vida muito curto, sendo apenas possível detectá-lo (caso exista) indiretamente através das partículas nas quais ele decai naturalmente.

Com o intuito de determinar as características básicas de todas as partículas medidas, a Colaboração ATLAS desenvolveu o **Software de Reconstrução *Offline* do ATLAS** utilizando as linguagens de programação C++ e Python. O *software* de reconstrução utiliza uma infra-estrutura comum denominada **Athena**.

O mecanismo de funcionamento do *software* de reconstrução é bastante simples, e baseia-se na noção de **algoritmo**. Para cada evento, ou seja, para cada colisão entre os feixes de prótons, uma lista determinada de algoritmos são executados, em uma ordem específica, utilizando a informação obtida pelo detector ou gerada pelos algoritmos precedentes para realizar qualquer processamento. A ordem na qual os

algoritmos são executados fica especificada em um arquivo de configuração escrito em Python, denominado *job options* (ou **opções de trabalho**, em português). No que concerne a implementação, todos os algoritmos herdam da classe C++ **Algorithm**.

Existem outras entidades no *software*, como ferramentas que realizam tarefas ou cálculos frequentes (**Tools**) e serviços que funcionam como suporte para os algoritmos (**Services**). Apesar de toda a infraestrutura disponível a idéia essencial é executar um conjunto de algoritmos em sequência.

Vários algoritmos foram criados para realizar cálculos simples utilizando informações obtidas por algoritmos executados anteriormente: a combinação destes permite identificar as partículas e suas características com boa performance. A Figura 2.8 mostra uma pequena parte da infraestrutura disponível no Athena com este fim. Nota-se que cada algoritmo tem acesso a um conjunto de serviços que lhe permite obter várias informações. O Serviço de Mensagem lhe permite enviar mensagens para o registro do trabalho de forma organizada. Um outro serviço permite o acesso às informações disponíveis no *job options* de configuração do trabalho. Existem outros serviços com diversos fins, visando a servir de suporte aos algoritmos.

Uma das necessidades fundamentais dos algoritmos é enviar dados para o algoritmo que será executado logo a seguir. Para suprir esta necessidade, o conceito de Armazenamento Transitório de Dados foi criado. O objetivo de um serviço que implementa o Armazenamento Transitório de Dados é armazenar um objeto C++ em uma região de memória e permitir que outros algoritmos compartilhem a informação publicada. O Serviço de Dados do Evento, implementado na classe C++ **StoreGateSvc**, permite essa troca de informações de maneira simplificada. O Serviço de Dados do Detector e o Serviço de Histograma armazenam, respectivamente, características do detector e histogramas gerados pelos algoritmos, permitindo a publicação de outros tipos de dados, utilizando o mesmo conceito.

A informação gerada pelos algoritmos, armazenada através dos Serviços de Armazenamento Transitório de Dados, pode ser escrita em mídia permanente para consulta e análise posterior. Há vários formatos de arquivos disponíveis para o armazenamento destes dados. Para os histogramas gerados em análises, arquivos no formato próprio do sistema de análise estatística ROOT [13] são utilizados. No

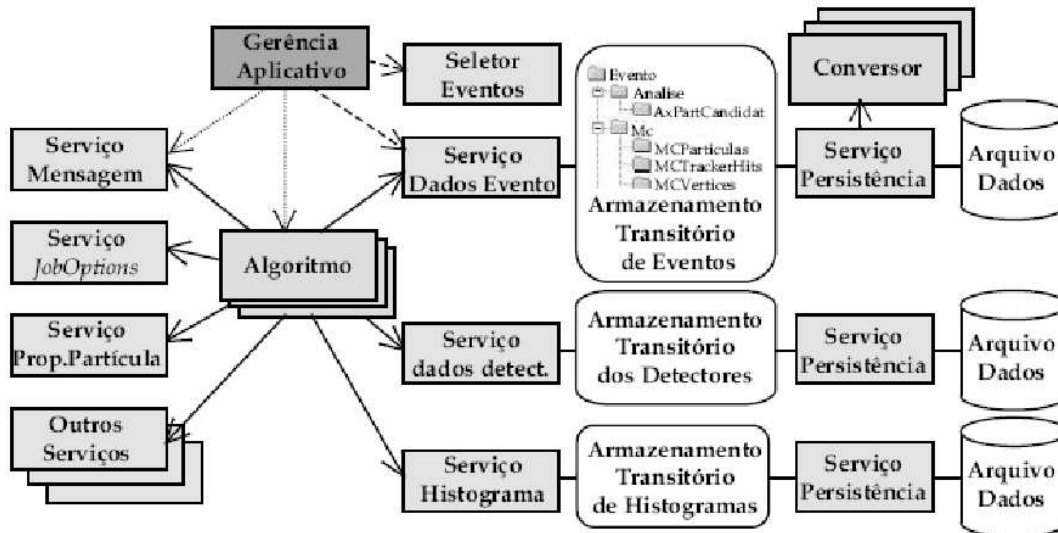


Figura 2.8: Pequena parte da infraestrutura disponibilizada no Athena. Extraído de [9].

entanto, visando a armazenar dados mais complexos, vários tipos de arquivos baseados no formato ROOT foram criados. Arquivos *Event Summary Data* (ESD)⁸ são utilizados para armazenar o resultado do processamento dos algoritmos de reconstrução. Arquivos *Raw Data Object* (RDO)⁹ são frequentemente utilizados para testes e contêm toda a informação bruta gerada pelo detector, antes de qualquer processamento.

O arquivo RDO é bastante flexível e pode armazenar todos os tipos de dados necessários para a reconstrução, análises e testes, mas contêm muita informação redundante e costumam ser bastante grandes. Por este motivo, foi criado um formato próprio para armazenar a informação obtida diretamente do detector, denominado **ByteStream**. O acesso aos dados do ByteStream é bem mais eficiente do que no caso da leitura da informação de arquivos RDO.

Portanto, a informação proveniente do detector será armazenada no formato ByteStream, que tem informações específicas ao detector ATLAS apenas. O sistema de reconstrução *offline* lerá os dados dos arquivos ByteStream e gerará um arquivo ESD, contendo o resultado das análises. Arquivos RDOs são frequentemente utilizados para representar dados simulados, uma vez que podem armazenar a informação

⁸Sumário de Dados de Evento, em português.

⁹Dados de Objeto Básicos, em português.

referente ao processo de simulação, ou seja, quais as partículas simuladas em cada evento.

Capítulo 3

O Sistema de Filtragem do Detector ATLAS

Este capítulo explica em mais detalhes o funcionamento do sistema de filtragem do detector ATLAS, com uma ênfase no canal elétron/fóton, onde este trabalho se insere.

3.1 Sistema de Filtragem

Em princípio, os requisitos para a análise física residem apenas nos dados providos pelo detector e em um sistema para processar os sinais após sua aquisição, identificando as características essenciais das partículas. No entanto, as limitações tecnológicas impuseram uma restrição à implementação deste modelo.

Com a colisão de feixes ocorrendo a cada 25 ns, a taxa de eventos é de 40 MHz com 23 partículas sendo geradas, em média, e interagindo com o detector na região do barril. O detector vai ainda encontrar uma quantidade muito grande de ruído da eletrônica e interagir com muitas partículas que envolvem processos já bem conhecidos da física de partículas, o que também forma o ruído de fundo da experiência. Vale também notar que os eventos de interesse, como o bóson de Higgs, ocorrerão muito raramente. Com a quantidade de informação que deve ser armazenada por evento em tais condições (~ 60 TB/s), torna-se inviável armazenar todas as informações disponíveis nos canais de leitura do detector.

Um sistema de filtragem [14] foi desenvolvido para eliminar informações irre-

levantantes ao processamento *offline*, limitando a taxa de ruído de fundo, isto é, eventos que não são interessantes para a análise física. Mesmo com este sistema de filtragem, a probabilidade de ocorrência dos eventos de interesse é muito baixa. Portanto, é fundamental que o sistema de filtragem tenha alta eficiência para a detecção de eventos interessantes e alta rejeição ao ruído de fundo, garantindo uma quantidade de dados suficiente e não-polarizada para análise posterior.

O sistema de filtragem (*Trigger System* em inglês) é dividido em três níveis sequenciais. O primeiro nível varre o detector procurando, apenas na informação provida pelos calorímetros e detectores de múons, regiões onde alguma informação possa ser interessante. Este primeiro passo no sistema de filtragem usa a informação do detector com granularidade menos fina. Nos calorímetros, apenas regiões de $(\Delta\eta; \Delta\phi) = (0, 4; 0, 4)$ são analisadas, sendo consideradas regiões interessantes se a energia estimada em algumas células for maior que um limiar definido. Devido à alta taxa de eventos, o primeiro nível é implementado em *hardware* utilizando FPGA e ASIC. As regiões marcadas como interessantes, as Regiões de Interesse, ou RoI (*Regions of Interest*), são enviadas para serem processadas pelo segundo nível de filtragem.

Os segundo e terceiro níveis de filtragem são geralmente referidos como **Sistema de Filtragem de Alto Nível**, ou *High Level Trigger System* (HLT). Eles são implementados em *software* e analisam a informação fornecida pelo nível anterior com maior detalhe. A implementação destes dois níveis do sistema de filtragem foi feita de tal forma que é possível executá-los na infra-estrutura do ambiente *offline*, ou seja, no Athena, da mesma forma que o sistema de reconstrução de eventos. Aproveitando essa característica, boa parte do último nível de filtragem, frequentemente referido como Filtro de Eventos ou *Event Filter*, foi desenvolvida com base nos algoritmos de reconstrução *offline*. Certas partes do Filtro de Eventos são realmente algoritmos de reconstrução executados no ambiente *online*, durante a aquisição de eventos, com simplificações.

Um desenho esquemático do Sistema de Filtragem no ATLAS está representado na Figura 3.1, onde observa-se o fluxo dos sinais obtido no detector. A informação é armazenada de forma analógica em *Shift Capacitor Arrays* (SCA) até ser aprovada pelo primeiro nível. Se ocorrer uma aprovação pelo primeiro nível, os

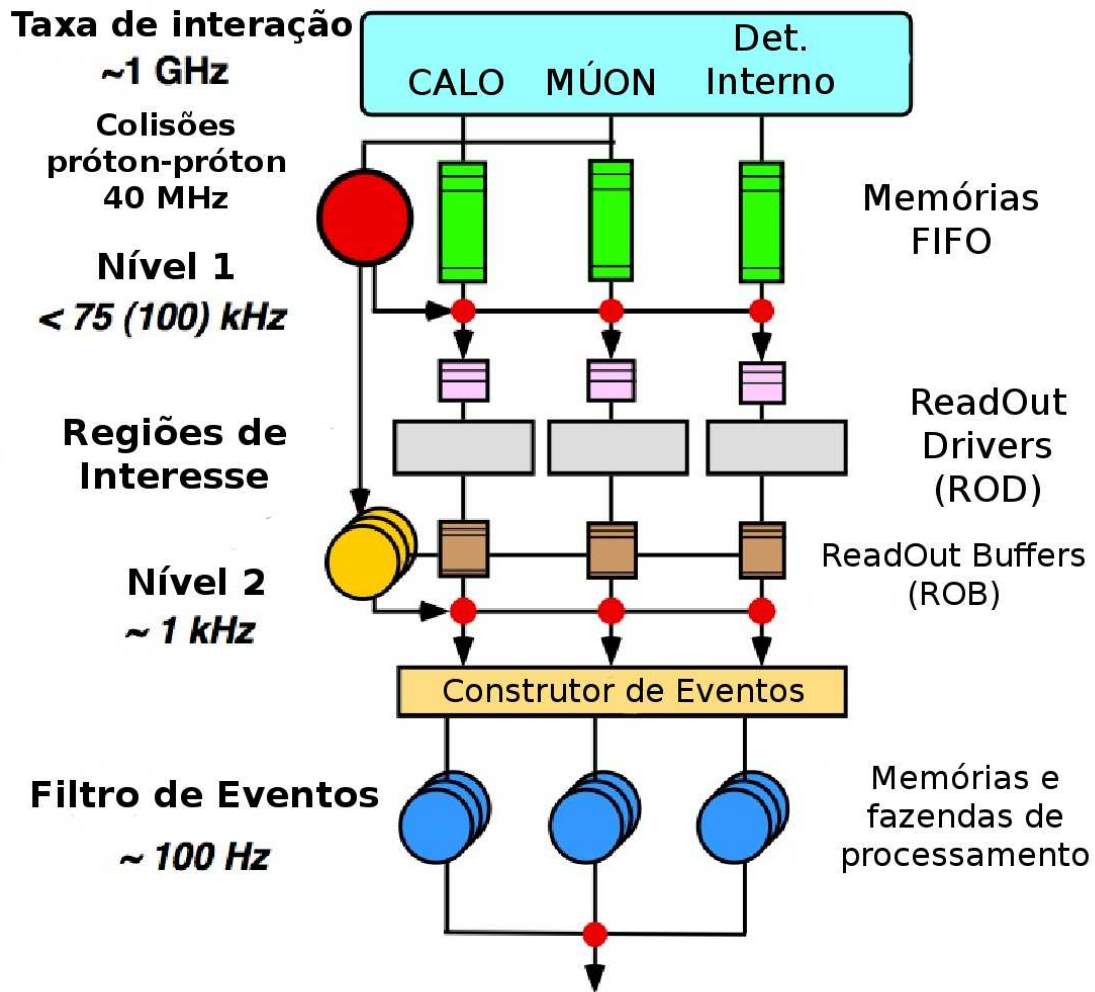


Figura 3.1: Gráfico esquemático do sistema de Filtragem do ATLAS.

dados são digitalizados e transferidos para os *ReadOut Drivers*.

Os *ReadOut Drivers* são dispositivos implementados em *hardware* que executam um processamento digital do sinal e o enviam para uma fazenda de computadores dedicados ao sistema de filtragem de alto nível. A informação proveniente dos *ReadOut Drivers* é armazenada em memórias, denominadas *ReadOut Buffers*, de onde pode ser obtida pelos segundo ou terceiro níveis de filtragem sob demanda.

O Filtro de Eventos analisa apenas os eventos aceitos pelo segundo nível de filtragem, tomando a decisão de armazená-los ou não. Os eventos aceitos são armazenados em mídia permanente para futuras análises no ambiente de reconstrução *offline*.

É importante lembrar que há uma limitação no tempo de execução de cada nível do sistema de filtragem, visto que os eventos a serem analisados são arma-

zenados em uma fila na memória para aguardar processamento e serão eliminados caso a fila esteja cheia. A limitação no tempo de processamento do segundo nível de filtragem é de ~ 40 ms, enquanto o Filtro de Eventos tem por volta de ~ 1 s para tomar uma decisão sobre a aceitação do evento.

Para entender melhor os algoritmos que tomam a decisão no sistema de filtragem, é preciso conhecer melhor os fenômenos físicos que ocorrerão no detector ATLAS. A Figura 3.2 mostra a taxa de ocorrência de variados tipos de eventos físicos para a luminosidade nominal do LHC. A abscissa mostra a massa das partículas detectadas no ATLAS para certo evento físico e a ordenada mostra a frequência na qual estes eventos ocorrerão.

Nesta figura, observa-se que os eventos adquiridos estão dominados por eventos de **jetos** na faixa de 50 GeV a 300 GeV, que têm pouca importância na física que está sendo estudada. As taxas de aquisição dos decaimentos dos bósons Z , W^+ e W^- estão mostradas em várias faixas de massa e são particularmente interessantes para realizar estudos de calibração no Sistema de Calorimetria. Na região de 100 GeV, por exemplo, encontra-se o decaimento do bóson Z em dois léptons. Eventos evidenciando decaimentos do bóson de Higgs ocorrem na faixa de frequência abaixo de milihertz, sendo muito mais difíceis de serem encontrados.

Existem vários algoritmos que realizam análises no sistema de filtragem de alto nível, dependendo da configuração do sistema de aquisição de dados. Anterior à aquisição de dados, um conjunto de canais de interesse são selecionados, ou seja, decide-se quais tipos de partículas devem ser armazenadas. Pode-se, por exemplo, utilizar apenas algoritmos que procurem por fótons, se é desejado analisar o decaimento do bóson de Higgs em dois fótons, e utilizar essa informação para estimar a massa do bóson de Higgs. A configuração dos algoritmos executados e, portanto, das partículas sendo pesquisadas é determinado pelo **Menu do Sistema de Filtragem**.

O menu configurado determina, não apenas o conjunto de algoritmos que serão executados, mas também a rigidez da seleção aplicada. Os algoritmos operam aplicando cortes com base nas características dos dados obtidos. Pode-se controlar a eficiência da filtragem e a taxa de eventos indesejados, alterando os cortes aplicados pelos algoritmos. Com seleções mais brandas, aumenta-se a taxa, mas garante-

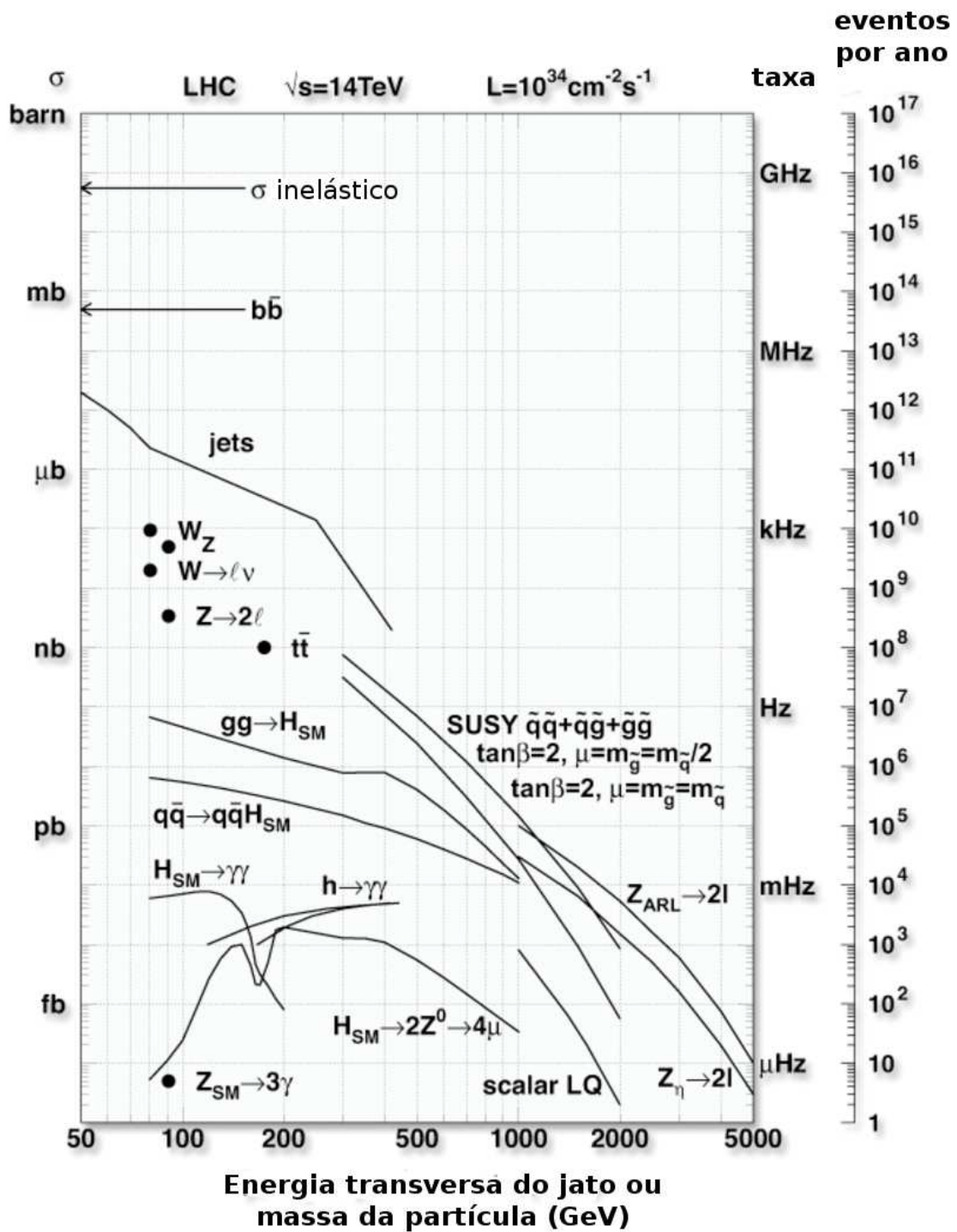


Figura 3.2: Seção de choque e taxa de eventos detectados para a luminosidade nominal do LHC. Extraído de [15].

se que haverá partículas de interesse selecionadas (se existirem). Por outro lado, pode-se aplicar seleções mais rígidas, diminuindo o ruído, mas reduzindo também a estatística obtida.

Após a seleção de granularidade menos fina do nível um, algoritmos específicos do nível dois são executados, com plena granularidade, para cada tipo de corte aplicado no primeiro nível, de forma a detectar apenas as partículas de interesse.

3.1.1 O canal elétron/fóton

No LHC, partículas decaem rapidamente em outras mais estáveis, de tal modo que há um interesse grande na detecção de partículas com maior tempo de vida. Um dos canais importantes envolve a detecção de elétrons, pósitrons e fótons. Este canal de estudo é constituído por algoritmos específicos baseados nas características destas partículas. Costuma-se denominar este canal de estudo **egamma**¹ [8].

O pacote de algoritmos de reconstrução *offline*, no canal Egamma, chama-se **egammaRec**, parte deste sendo utilizado no Filtro de Eventos para a discriminação e extração de características.

O funcionamento dos algoritmos que detectam elétrons baseia-se, em primeira instância, na informação de energia fornecida pelos calorímetros, pois um elétron depositará apenas energia nos calorímetros eletromagnéticos, e não nos sistemas de calorimetria hadrônica. Utilizando-se também os sistemas do detector interno e o campo magnético gerado pelos toróides, é possível estimar o momento do elétron e sua razão carga-massa, permitindo sua identificação.

Os fótons não deixam um traço no detector interno, mas sua detecção pode ser feita com base no sistema de calorimetria. Em altas energias ocorre um fenómeno denominado **bremstrahlung** [8], no qual um elétron, ao sofrer aceleração ou desaceleração, emite fótons. Este fenómeno pode ocorrer devido à interação do elétron com os sistemas de calorimetria e é um fenómeno que não pode ser descartado e pode até mesmo ser usado para a identificação de elétrons e fótons.

Um conjunto grande de outras partículas decaem em elétrons ou fótons, de

¹O nome é utilizado apenas na comunidade do ATLAS e vem do símbolo atribuído às partículas. O elétron é escrito e^- (e^+ para o pósitron) e o fóton indica-se pela letra grega γ .

forma que pode-se detectar os elétrons e utilizar esta informação para reconstruir as características das partículas originais que decaíram em elétrons. Uma dessas partículas é o Z , que decai com certa probabilidade em dois elétrons. Medindo o momento e energia de cada elétron, pode-se estimar estas mesmas características do Z , assim como da partícula que o gerou.

A partícula Z já é bem conhecida pelos resultados de outros experimentos, o que nos permite tirar várias conclusões sobre os elétrons nos quais ela decai, assim como sobre os sistema de detecção. O decaimento de Z em elétrons é frequentemente utilizado para verificar o funcionamento dos calorímetros e sua resolução em energia.

Se for comprovada a existência do bóson de Higgs, uma das previsões para seu decaimento é em dois fótons. De outra forma, a detecção de dois fótons vindo de um mesmo ponto poderia indicar a existência do bóson de Higgs, o que mostra que a detecção de fótons é bastante importante. Vários conjuntos simulados deste evento físico estão disponíveis e é bastante importante verificar o funcionamento do sistema de filtragem e dos calorímetros com estes eventos.

3.1.2 O ruído para o canal elétron/fóton

Os eventos que deseja-se observar ocorrem em energia bastante alta, mas estão sempre poluídos com eventos muito mais freqüentes e ruído da eletrônica dos sistemas de detecção.

O ruído da eletrônica é mais acentuado na região de baixas energias. Justo por este motivo, a forma mais simples de eliminá-lo é aplicar um corte em energia, aceitando eventos com energia maior que um certo patamar.

Em baixas energias² também ocorrem outros fenômenos já bem conhecidos, como o decaimento da partícula J/Ψ em elétrons. Tais eventos podem ser interessantes para auxiliar na verificação do funcionamento dos calorímetros, mas não há grande interesse em mantê-los se o objetivo é comprovar as novas teorias da física de partículas.

Apesar deste ruído em baixas energias, uma das principais fontes de ruído no canal Egama são jatos [16]. Jatos não são partículas, mas uma cascata de partículas geradas por um fenômeno denominado hadronização. Jatos podem ocor-

²Menos de 10 GeV.

rer em altas energias e são detectados com frequência muito maior do que os canais de interesse, dificultando a análise física dos eventos.

3.2 Infra-estrutura do Sistema de Filtragem de Alto Nível

Para melhor compreender os modos pelos quais o sistema de filtragem de alto nível pode falhar, é fundamental entender seu funcionamento em detalhes. O sistema de filtragem compartilha muitos conceitos com o sistema de reconstrução *offline*. Dois deles são o conceito de Armazenamento Transitório de Dados e de algoritmo. Para entender melhor o funcionamento do sistema, é necessário compreender um conjunto de conceitos específicos.

O Sistema de Filtragem de Alto Nível é construído sobre a arquitetura do sistema de reconstrução *offline*, adicionando características próprias e restringindo a execução de acordo com os requisitos do projeto. Sua implementação baseia-se no conceito de **TriggerElement**, que representa um elemento que deve ser processado. Quando o primeiro nível de filtragem detecta uma região interessante, um novo TriggerElement é criado, associado ao corte implementado pelo primeiro nível. Ao TriggerElement podem ser anexados outros objetos, representando dados que podem ou devem ser processados. Também é possível criar uma hierarquia entre os TriggerElements: se um TriggerElement foi processado com sucesso, a consequência é a criação e a necessidade de processamento de outro TriggerElement. Esta hierarquia é utilizada para representar, na estrutura de dados, os vários níveis de processamento existentes no sistema de filtragem.

Para formalizar a configuração do processamento, cria-se a noção de **cadeia**. Uma cadeia representa a detecção de um sinal procurado no segundo nível ou Filtro de Eventos. Para que uma cadeia seja **aceita**, isto é, para que se entenda que o sinal pesquisado foi encontrado, um conjunto de hipóteses devem ser verdadeiras. Estas hipóteses denominam-se **assinaturas**. Para, afinal, verificar se uma assinatura foi encontrada, uma ou mais **sequências** de algoritmos devem ser executadas e todas devem garantir que a hipótese (assinatura) levantada é verdadeira.

O *Steering* é o sistema que gerencia o processamento do sistema de filtragem,

ativando a execução dos vários passos de processamento. O *Steering* obtém as regiões de interesse assinaladas pelo primeiro nível de filtragem e cria novos TriggerElements para cada um dos cortes satisfeitos. A cada TriggerElement, associam-se objetos, identificando a posição do detector que causou sua geração.

Uma vez compreendidos estes elementos básicos para o funcionamento do sistema de filtragem, pode-se analisar seu funcionamento geral. O TriggerElement gerado pelo nível 1 é enviado à cadeia de nível 2 correspondente. Para verificar se a cadeia de nível 2 foi aceita, o Steering tenta verificar se as assinaturas correspondentes são verdadeiras. Para isto, as sequências de algoritmos associadas à assinatura são executadas, dando as hipóteses como verdadeiras ou falsas.

Cada sequência de algoritmos é sempre constituída por um ou mais algoritmos de extração de características (*Feature EXtraction Algorithm* ou *FEX Algorithm*) seguidos de nenhum ou um algoritmo de teste de hipóteses (*HYPOTHESIS Algorithm* ou *HYPO Algorithm*). Com esta arquitetura, pode-se realizar um processamento inicial aos dados recebidos do primeiro nível e depois aplicar diferentes decisões, dependendo da partícula que deseja-se manter. Uma vez que o processamento inicial não é executado mais de uma vez para a mesma RoI, este método economiza tempo de execução e uso de memória.

A Figura 3.3 mostra uma cadeia típica³ no sistema de filtragem para o canal elétron/fóton. A Região de Interesse é detectada pelo primeiro nível de filtragem e alimenta os algoritmos de análise do segundo nível e do filtro de eventos.

A cadeia associada ao segundo nível de filtragem, representada no retângulo no centro da figura, é dividida em três passos de filtragem. O primeiro passo contém dois algoritmos, como mostrado no retângulo de bordas arredondadas ao topo: **T2CaloEgamma_eGamma** e **L2CaloHypo_e5_medium**. O primeiro algoritmo extrai características úteis à classificação utilizando apenas a informação do sistema de calorimetria e o segundo toma decisões baseado na informação calculada pelo algoritmo anterior. Um exemplo de informação estimada no algoritmo T2CaloEgamma_eGamma é a energia eletromagnética que a partícula depositou nos calorímetros. O algoritmo L2CaloHypo_e5_medium utiliza esta característica para

³Nominalmente, as cadeias L2_e5_medium e EF_e5_medium, que verificam a hipótese da existência de elétrons com energia transversa maior do que 5 GeV.

rejeitar regiões de interesse com baixa deposição de energia eletromagnética nos calorímetros.

O passo seguinte contém apenas algoritmos de extração de características que utilizam somente a informação dos detectores internos. Para este passo, os algoritmos **TrigSiTrack_eGamma** e **TrigIDSCAN_eGamma** são executados para calcular os traços gerados nos detectores internos. Nenhuma região de interesse é rejeitada neste passo, visto que não há algoritmo de teste de hipótese.

O terceiro passo combina a informação obtida no primeiro passo e a informação extraída no segundo passo com o algoritmo **L2IDCaloFex_all**. Esta informação é utilizada para realizar um novo corte no algoritmo de teste de hipóteses **L2IDCaloHypo_e5_medium**. Este último algoritmo, no nível 2, aceita apenas partículas para as quais foram detectadas energias e traços consistentes com a hipótese de a partícula ser um elétron. Isto é válido, pois um elétron é uma partícula que será detectada pelo sistema de calorimetria e pelos detectores de traço.

O filtro de eventos está representado ao lado do segundo nível de filtragem, na Figura 3.3. Este também é dividido em três passos, que utilizam a mesma lógica que a utilizada no segundo nível. O primeiro passo utiliza apenas a informação de calorimetria. O passo seguinte é composto por vários algoritmos (não é possível mostrar todos no gráfico) que utilizam apenas a informação dos detectores de traço. O último passo combina estas informações. Apesar de utilizarem a mesma lógica, o filtro de eventos faz uma pesquisa mais detalhada e calibrações mais finas na informação provida pelos sub-sistemas do detector.

3.3 Validação do Sistema de Filtragem de Alto Nível

Uma vez que é possível executar o sistema de filtragem no ambiente Athena, é útil reutilizar este ambiente para realizar testes no *software*, validando o código e o algoritmo sem toda a infraestrutura do sistema de aquisição de dados. Aproveita-se ainda a própria estrutura de validação do *software* de reconstrução *offline*, evitando a duplicação de esforços.

A utilização de uma infraestrutura comum tem muitas outras vantagens

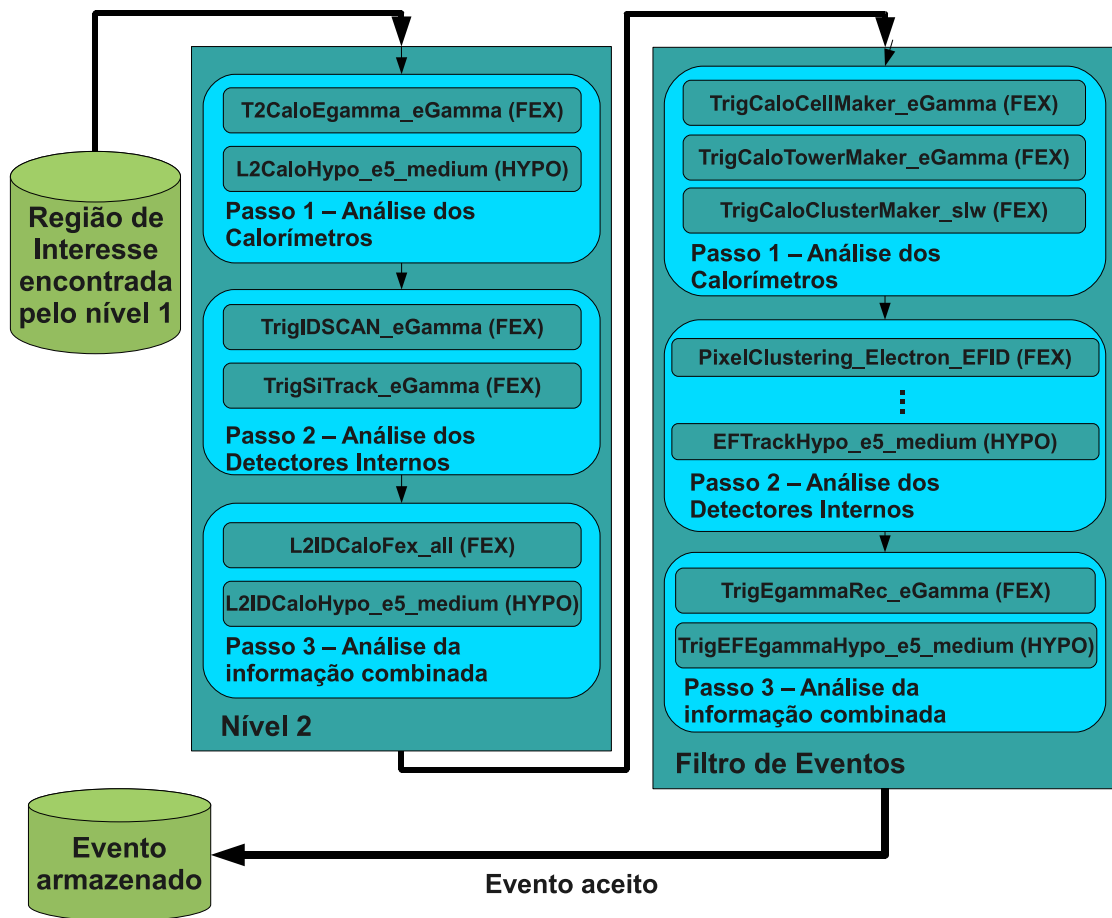


Figura 3.3: Organização dos algoritmos típicos no sistema de filtragem para o canal elétron/fóton.

mesmo fora da validação, como, por exemplo, o uso da mesma estrutura de desenvolvimento que o *software* de reconstrução, permitindo a exportação de idéias livremente. Justamente tal flexibilidade permitiu o compartilhamento de boa parte do *software* no Filtro de Eventos, como foi explicado anteriormente.

Antes de compreender a estrutura da validação, é útil entender como as novas versões do *software* são criadas, tanto no sistema de filtragem quanto no sistema de reconstrução *offline*, pois esse mecanismo é bastante útil na identificação e rápida correção de problemas.

O *software*⁴ é dividido em pacotes individuais com tarefas específicas e versões próprias evoluindo de acordo com o seu desenvolvimento e sendo controladas pelo sistema de controle de versão CVS⁵ (*Concurrent Version System*) [18]. Todas as noites uma nova versão dos **projetos**, compostos por vários pacotes de *software*, é compilada para vários ramos (*branches*) do *software*, cada um com um objetivo específico. Há, por exemplo, um ramo de validação (isto é, a ser validado) e outro de desenvolvimento que é mais estável.

Cada nova versão dos projetos é compilada com versões particulares dos pacotes que os compõem, determinadas pelos desenvolvedores dos pacotes. Todas as noites a nova versão do *software* denomina-se *rel_n*, sendo *n* um número inteiro de 0 a 6. Portanto, há uma versão do *software* para cada dia da semana. Ao chegar-se em *rel_6*, remove-se a versão anterior de *rel_0* e uma nova *rel_0* é criada, prosseguindo-se assim por toda a semana. No momento em que se julga que o *software* está estável, uma nova versão permanente do projeto é criada, contendo as versões dos pacotes individuais da compilação noturna mais estável. Esta versão permanente não é sobre-escrita na semana seguinte.

A determinação das versões a serem utilizadas para cada pacote na compilação noturna do projeto é feita pelos desenvolvedores utilizando o sistema Tag Collector [19], através de uma interface Web. Cada mudança na versão corrente deve ser justificada, facilitando o rastreamento de problemas, caso ocorram.

⁴Para a explicação que segue, entende-se *software* como qualquer parte do sistema de reconstrução *offline* ou do sistema de filtragem, uma vez que ambos utilizam o mesmo procedimento.

⁵Planeja-se uma mudança no sistema de controle de versão no ano de 2009 para a ferramenta Subversion [17].

Package Name	Container	Build	Checkreq Test	Unit Test	Manager(s)
AtlasTriggerRelease	N/A	●	N/A	FAIL	emil.obreshkov at cern.ch,david.quarrie at cern.ch
TrigCaloRec	Trigger/TrigAlgorithms	●	N/A	N/A	cibran.santamarina.rios at cern.ch,xella at nbi.dk,delsart at ips.umontreal.ca,fer
TrigCaloRinger	Trigger/TrigAlgorithms	●	N/A	N/A	rodrigo.torres at cern.ch,daniloefl at gmail.com
TrigEFMissingET	Trigger/TrigAlgorithms	●	N/A	N/A	diego.casadei at cern.ch,kyle.cranmer at cern.ch,xiaowen.lei at cern.ch,ignacio.a
TrigT2CaloCommon	Trigger/TrigAlgorithms	●	N/A	N/A	xin.wu at physics.unige.ch,pilar.casado at cern.ch,kyle.cranmer at cern.ch,damaz
TrigT2CaloCosmic	Trigger/TrigAlgorithms	●	N/A	N/A	damazio at mail.cern.ch
TrigT2CaloEgamma	Trigger/TrigAlgorithms	●	N/A	N/A	xin.wu at physics.unige.ch,damazio at mail.cern.ch,carlos.osuna.escamilla at cer
TrigT2CaloJet	Trigger/TrigAlgorithms	●	N/A	N/A	kyle.cranmer at cern.ch,xin.wu at physics.unige.ch,ignacio.aracena at cern.ch,till
TrigT2CaloTau	Trigger/TrigAlgorithms	●	N/A	N/A	olga.igonkina at cern.ch,carlos.osuna.escamilla at cern.ch,pilar.casado at cern.ch
TrigT2MinBias	Trigger/TrigAlgorithms	●	N/A	N/A	w.bell at cern.ch,regina.kwee at cern.ch
TrigValAlgs	Trigger/TrigValidation	●	N/A	N/A	goncalo at pp.rhul.ac.uk,strom at physics.uoregon.edu
TrigmuIso	Trigger/TrigAlgorithms	●	N/A	N/A	stefano.giagu at cern.ch
TrigNavigation	Trigger/TrigEvent	●	N/A	FAIL	tomasz.bold at cern.ch,carlo.schiavi at ge.infn.it
TrigSteering	Trigger/TrigSteer	●	N/A	FAIL	nicolas.berger at cern.ch,till.eifert at cern.ch
TrigTimeAlgs	Trigger/TrigTools	●	N/A	FAIL	j.t.m.baines at rl.ac.uk,tomasz.bold at cern.ch
AnalysisTriggerAlgs	PhysicsAnalysis/AnalysisTrigger	●	N/A	N/A	wolfgang.ehrenfeld at cern.ch,tmaeno at bnl.gov,atw at hep.ph.bham.ac.uk,mon
AtlasTriggerRunTime	N/A	●	N/A	N/A	emil.obreshkov at cern.ch,david.quarrie at cern.ch

Figura 3.4: Resultado da compilação e *link-edição* dos pacotes no projeto AtlasTrigger.

A compilação noturna dos projetos é feita pelo *Nightly Control System* (NICOS) [20] automaticamente, e o resultado da compilação é mostrado em um sistema Web para monitoração dos desenvolvedores. Como os projetos são muito complexos, os desenvolvedores podem testar se suas modificações estão de acordo com o funcionamento de um pacote, mas podem não prever suas consequências em outros, causando, no caso mais grave, erros de compilação. Os registros de compilação podem ser monitorados na interface Web do NICOS, permitindo identificar a causa do problema.

A Figura 3.4 mostra o resultado da compilação de cada pacote do HLT em uma das versões noturnas. O nome do pacote é identificado na primeira coluna, à esquerda, onde se pode seguir o *hyperlink* para obter os registros de compilação e *link-edição*. A segunda coluna indica o diretório onde o pacote está localizado, que tem fins organizacionais, apenas. A coluna a seguir contém um círculo, cuja cor comunica se houve erros de compilação, avisos ou nenhum problema foi detectado neste estágio. As duas colunas seguintes indicam os resultados de testes que verificam a dependência entre pacotes. A última coluna, à direita, contém o nome dos desenvolvedores ou gerentes deste pacote, que devem ser contactados em caso de problemas.

Cada projeto é constituído por um conjunto de pacotes, como os mostrados na Figura 3.4. Os pacotes ilustrados na figura correspondem ao projeto AtlasTrigger,

que contém pacotes destinados ao funcionamento do sistema de filtragem de alto nível. Outros projetos têm objetivos diferentes, como o projeto AtlasSimulation, que mantém o *software* destinado à simulação de eventos no detector.

Após a compilação com sucesso dos projetos, pequenos testes programados pelos desenvolvedores são executados. Os testes *Atlas Testing Nightly* (ATN) são executados logo após a compilação e também têm os resultados acessíveis na interface Web do NICOS. Estes testes podem executar quaisquer comandos que se deseje, permitindo comparar histogramas e resultados em seguida. Para os pacotes do projeto AtlasTrigger, costuma-se executar o ambiente de reconstrução Athena com apenas 10 eventos de colisão de dados simulados e avaliar histogramas com informações dos algoritmos de extração de características e os registros de texto emitidos pelo *software* em modo de alta verbosidade.

Apenas com a comparação dos registros de texto⁶ do *software* pode-se determinar minúsculas mudanças no mesmo. Determina-se, em seguida, se o efeito da mudança era esperado ou considerado danoso ao projeto.

A comparação feita com apenas 10 eventos detecta muitas mudanças, mas também é necessário um meio de realizar a comparação com mais dados, verificando tendências que possam ter sido inseridas na execução do *software*. A análise da resposta física também é interessante e deve ser executada sistematicamente. O *Run Time Tester* (RTT) [21] é uma infra-estrutura destinada a executar, a cada nova versão do software, um conjunto de trabalhos com configurações específicas, analisando muito mais eventos, geralmente com dados simulados.

Assim que um problema é detectado, a ferramenta Web Savannah [22] é utilizada para assinalar o problema a um dos desenvolvedores, que deve detectar a causa e resolver o problema, ou assinalar o problema a outro desenvolvedor com maior compreensão do problema. Esta ferramenta Web permite uma comunicação entre os desenvolvedores e é uma forma eficiente de organizar a detecção de falhas, mantendo os registros para referência futura. Ferramentas públicas para depuração de *software*, como o GDB [23] e as ferramentas Valgrind [24] são muito utilizadas para encontrar a causa de falhas, quando estas são detectadas.

⁶Mensagens de depuração emitidas pelos algoritmos individualmente.

Capítulo 4

O projeto TrigEgammaValidation

Para validar o funcionamento dos algoritmos do sistema de filtragem de alto nível, foi desenvolvida uma ferramenta para gerar histogramas, buscando evidências eventuais de problemas sérios no *software* e na performance física. Este pacote de teste, o TrigEgammaValidation, é executado diariamente a cada nova versão do Sistema de Filtragem, através da arquitetura do RTT (*Run Time Tester*).

4.1 Enfoque de Projeto

O objetivo do projeto é, em essência, comparar duas versões do sistema de filtragem, isto é, medir as consequências das alterações aplicadas no *software*, verificando possíveis polarizações estatísticas, erros na análise, falhas de implementação ou qualquer outra inconsistência. A ação do sistema de filtragem é melhor verificada observando-se o efeito do sistema em conjuntos de dados simulados, onde o efeito desejado é bem conhecido¹.

Visando a rastrear mudanças no *software* da forma mais detalhada possível, é útil analisar a resposta do sistema após cada mudança de versão. Desta forma, seria possível saber exatamente em qual versão uma mudança crucial ocorreu, permitindo a busca das mudanças na implementação que causaram o referido efeito, valendo-se dos registros disponíveis através da ferramenta CVS.

¹Dados produzidos a partir de raios cósmicos durante o teste do detector estão disponíveis. Estes dados, no entanto, não reproduzem condições analisadas pelos algoritmos do Sistema de Filtragem no canal elétron/fóton.

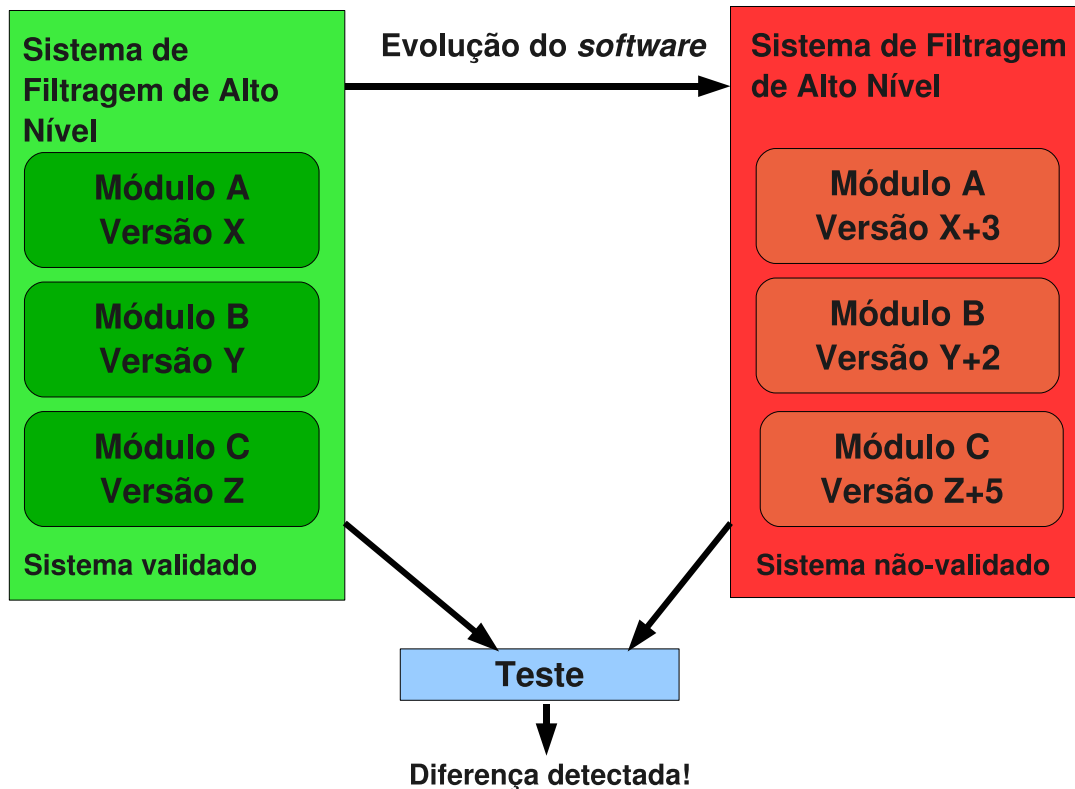


Figura 4.1: O enfoque do projeto está em verificar as alterações do *software* à medida que este evolui da versão X para a versão $X + 1$.

A Figura 4.1 mostra, de forma esquemática, o efeito de uma evolução no *software*. Esta figura mostra duas versões do sistema de filtragem, lado a lado. Cada versão do sistema é composta por módulos com funções específicas. Após algumas alterações nos módulos que o compõem, uma nova versão do sistema de filtragem é criada e precisa ser validada. Supondo que a versão à esquerda está bem validada, deseja-se verificar se a nova versão, à direita, funciona como esperado através de testes que meçam e apontem as diferenças no *software*.

Para verificar o funcionamento do Sistema de Filtragem de Alto Nível, utiliza-se o ambiente de Reconstrução *Offline*, já que é possível executar o sistema de filtragem acoplado a este ambiente. Em toda nova versão do sistema de filtragem, os vários **procedimentos de teste** são executados, cada um consistindo na execução do HLT e um ou mais algoritmos desenvolvidos no pacote de validação, além de, possivelmente, executarem-se os algoritmos de Reconstrução *Offline*. Os algoritmos obtêm a informação gerada pela Reconstrução *Offline*, pelo Sistema de Filtragem e

a informação dos eventos físicos simulados com todos os detalhes do verdadeiro processo (quando disponíveis) e comparam todos os dados disponíveis com os obtidos em uma versão anterior. Toda esta informação está disponível através do Armazenamento Transitório de Dados do Evento (utilizando-se o serviço **StoreGateSvc**) e é utilizada pelos algoritmos desenvolvidos.

Uma vez obtida, tal informação é utilizada para gerar histogramas de variáveis relevantes ao Sistema de Filtragem e correlacioná-las em gráficos dispostos na forma de histogramas. Não ocorrendo alterações na informação utilizada como referência, os histogramas de uma versão alteram-se com relação à outra apenas devido a mudanças no Sistema de Filtragem, permitindo um rastreamento mais rápido das mudanças no *software* sendo validado.

Para monitorar qualquer alteração no sistema de forma semiautomática, meios estatísticos são utilizados. Notando-se diferenças grandes entre histogramas de versões anteriores e uma versão mais recente, os desenvolvedores podem ser notificados para que se verifique a causa da discrepância.

4.2 Implementação da infra-estrutura e o ambiente Run Time Tester

A infra-estrutura do RTT [25] permite a configuração de um conjunto de trabalhos de testes que serão executados para cada nova versão, utilizando um arquivo XML dentro do pacote TrigEgammaValidation. Tal arquivo de configuração determina tanto as opções de teste que serão utilizadas para executar o **Athena**, quanto os scripts de pós-processamento necessários para gerar os resultados finais, que podem ser escritos em Python ou em forma de uma macro para o sistema de análise estatística do ROOT [13]. Além da forma do processamento que deve ser realizado, a configuração também ajusta os conjuntos de dados simulados que devem ser processados pelo trabalho de teste do RTT.

O procedimento de teste está representado na Figura 4.2. É executado um trabalho de reconstrução na infra-estrutura Athena para um conjunto de dados simulados, armazenado em formato *Raw Data Format* (RDO) ou no formato de

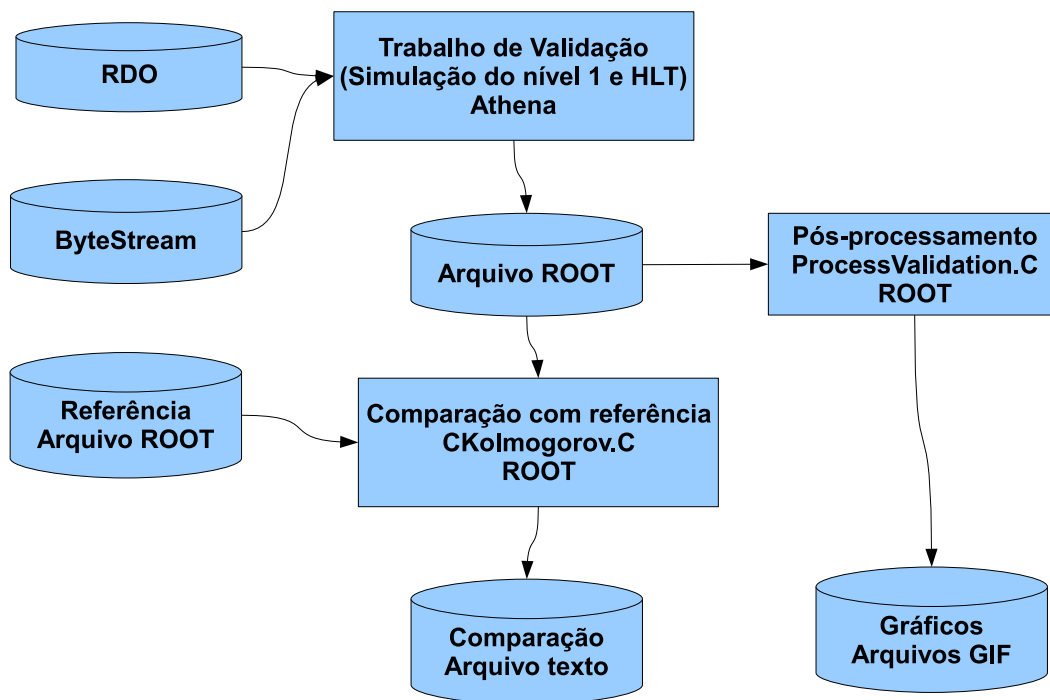


Figura 4.2: Estrutura geral dos trabalhos de teste do TrigEgammaValidation no RTT.

representação de dados criado pela colaboração ATLAS, *ByteStream*². Enquanto a execução do HLT utilizando dados simulados armazenados em arquivos RDO valida o funcionamento dos algoritmos no sistema de filtragem, ela não verifica o correto funcionamento do sistema de leitura de dados a partir de *ByteStream*, que será o formato efetivamente utilizado pela comunidade quando dados reais forem obtidos. Por outro lado, não é possível obter informações acerca da simulação quando se lê os arquivos no formato *ByteStream*, uma vez que estes não contêm este tipo de informação³.

Após a execução do trabalho de reconstrução e execução do HLT, é gerado um arquivo no formato ROOT contendo histogramas de variáveis interessantes à validação do sistema de filtragem. O passo seguinte é a execução do *script* **ProcessValidation.C** no sistema de análise estatística ROOT, que lê os histogramas, formata-os e gera arquivos de imagens GIF para apresentação ao usuário.

O próximo passo consiste na execução do *script* **CKolmogorov.C**, que compara todos os histogramas disponíveis no arquivo ROOT utilizando várias métricas estatísticas e disponibilizando o resultado em um arquivo de texto **results.txt**. Finalmente, o *script* Python **didItFail.py** verifica os resultados das métricas estatísticas no arquivo **results.txt** e envia esta informação para os desenvolvedores do TrigEgammaValidation através de correio eletrônico.

A infraestrutura RTT trata da execução desta série de passos. Os algoritmos de validação desenvolvidos para que sejam utilizados no passo de execução do Athena serão discutidos em detalhes a seguir. As métricas estatísticas utilizadas no *script* **CKolmogorov.C** também serão descritas mais detalhadamente.

4.3 Conjuntos de dados simulados

Os conjuntos de dados utilizados para realizar os testes foram escolhidos por terem relevância na verificação do funcionamento dos subsistemas do ATLAS. Todos incluem eventos físicos já bem estabelecidos por experimentos anteriores,

²Ver seção 2.3 para mais detalhes sobre os formatos de arquivos utilizados.

³Arquivos no formato *ByteStream* foram especificados para armazenar dados reais adquiridos pelo detector ATLAS, não sendo preparados para portar informações sobre simulações.

especialmente conjuntos simulados para situações de interesse, nas quais espera-se determinado comportamento do sistema de filtragem.

Um dos conjuntos de simulação utilizados consiste de eventos físicos de elétrons com energias diversas interagindo em direções quaisquer com o detector, aos quais costuma-se chamar de *single electrons*. Tais eventos físicos são completamente artificiais, pois não consistem de colisões⁴.

Apesar de o conjunto de dados ser bastante artificial, tais simulações são bastante úteis para analisar apenas a resposta do sistema de filtragem à interação do elétron com o detector. Pode-se, por exemplo, medir a eficiência do sistema de filtragem para detecção de elétrons nas diferentes posições do detector e para diferentes faixas de energia. Além da eficiência, é importante saber como reconstruir a energia dos elétrons a partir dos dados obtidos e medir a resolução dos calorímetros.

Outro conjunto de dados que fornece uma informação bastante importante é a simulação do decaimento de uma partícula Z em dois elétrons. Como foi comentado antes, a partícula Z já foi detectada antes em outros experimentos e conhece-se bastante sobre ela. Sendo uma partícula instável, ela decai em dois elétrons e espera-se detectá-la, após o Filtro de Eventos, com uma frequência da ordem de 1 Hz. O interesse neste particular decaimento do bóson Z vem da possibilidade de usá-lo para medir a eficiência do sistema de filtragem em relação à Reconstrução *Offline* durante a aquisição real de eventos, utilizando uma técnica denominada *Tag and Probe* [26]. Há também a possibilidade do decaimento do bóson de Higgs em dois bósons Z , que decairão, em seguida, em elétrons.

O decaimento de um bóson de Higgs em dois fótons é um conjunto de dados de simulação que tem particular interesse, pois um dos objetivos do experimento é justamente detectar o bóson de Higgs. Fótons não têm carga como o elétron, e, portanto, não sofrem interação com o campo magnético. Consequentemente, nenhuma medida nos sistemas de detecção de traço é exigida para a aceitação de um fóton pelo HLT.

Para verificar a resposta do sistema de filtragem ao ruído de fundo provo-

⁴Sempre que houver uma colisão entre os dois feixes de prótons, haverá pelo menos duas partículas, de forma que a quantidade de movimento do sistema seja conservada. Portanto, é impossível que haja apenas um único elétron em um evento.

cado por jatos duplos, uma simulação deste conjunto é utilizada. Pode-se utilizá-la para estimar a taxa de rejeição dos eventos pelo sistema de filtragem. Este último conjunto é bastante importante, visto que o sistema de filtragem tem como objetivo final otimizar a eficiência mantendo a taxa de eventos no sistema de aquisição abaixo do limite projetado.

4.4 Algoritmos

Cada conjunto de dados descrito tem características distintas, as quais devem ser verificadas de diferentes formas. Para cada conjunto, foi desenvolvido um algoritmo específico que obtém as informações calculadas pelo sistema de filtragem, disponíveis no Repositório de Dados Transitórios do Evento (**StoreGateSvc**) e aplica os cortes dos algoritmos de hipótese para a detecção de um elétron.

O ponto crucial é verificar se a informação extraída pelos algoritmos de extração de características do sistema de filtragem está correta e medir a eficiência dessa filtragem artificial, que esperamos ser bem próxima da eficiência medida pelo sistema de filtragem real.

Com estes algoritmos de validação, não se verifica a acurácia dos algoritmos de teste de hipóteses, que são os que realmente aplicam os cortes, mas valida-se os algoritmos de extração de características, que são bem mais complexos.

Dois tipos de algoritmos foram desenvolvidos. Em um caso compara-se a informação do sistema de filtragem com os acontecimentos gerados (a **Verdade**, isto é, a informação gerada na simulação MonteCarlo), obtida através dos metadados gerados durante a simulação. Em um segundo caso, compara-se o resultado da filtragem com a reconstrução *offline* dos eventos, sobre a qual assume-se que seja a forma ótima de realizar a reconstrução com a informação disponível com colisões.

Em alguns formatos de persistência, como os arquivos *Raw Data Object* (RDO) e *Event Summary Data* (ESD)⁵, pode-se armazenar a informação da simulação. Alguns algoritmos desenvolvidos utilizam esta informação para realizar uma comparação entre as partículas detectadas pelo Sistema de Filtragem e as partículas simuladas. Os algoritmos que realizam esta comparação são executados

⁵Ver Seção 2.3 para mais detalhes.

no Athena, como descrito na Seção 4.2, denominando-se **ZeeValidation**, **HggValidation** e **SingleElectronValidation**. Em todos os algoritmos mencionados, a idéia básica é a mesma: verifica-se a posição onde o elétron simulado interage com o detector (η e ϕ) e sua verdadeira energia transversa (E_T), comparando ambas com a estimativa do Sistema de Filtragem. Uma vez descoberta a associação entre o elétron verdadeiro e o detectado pelo primeiro nível do Sistema de Filtragem, pode-se prosseguir na análise para os níveis seguintes, realizando esta mesma comparação para cada análise feita no Sistema de Filtragem.

Os algoritmos de teste aplicam os mesmos cortes esperados pelos algoritmos de hipóteses reais para a assinatura de elétrons `e22i_tight`, que procura elétrons com energia transversa maior que 22 GeV, isolados de outras partículas. O corte `e22i_tight` foi escolhido, por ser um corte extremamente importante na análise física para a seleção de elétrons. As variáveis de corte podem, no entanto, ser alteradas, mudando a configuração do trabalho de teste.

Para o algoritmo **SingleElectronValidation**, comparam-se os elétrons detectados pelo Sistema de Filtragem com as informações obtidas sobre a simulação de elétrons. O algoritmo **ZeeValidation**, por outro lado, procura, na informação da simulação, apenas elétrons resultantes do decaimento de bósons Z. Da mesma forma, o algoritmo **HggValidation**, procura, na informação da Verdade, fótons resultantes de um decaimento do bóson de Higgs. Em todos os casos, o procedimento, após encontrar-se a partícula de referência, é o mesmo.

Quando a análise realiza a comparação com o código *offline*, o processo é idêntico, mas a referência não é a Verdade. O objeto `egammaContainer`, disponível no Repositório Transiente de Dados do Evento, sumariza toda a informação extraída pelos algoritmos de processamento *offline* para partículas eletromagnéticas. Os algoritmos disponíveis para este fim são o **ZeeOfflineValidation** e o **SingleElectronOfflineValidation**.

Além de verificar a boa resposta do sistema de filtragem ao sinal procurado, é preciso verificar se a taxa de ruído está aceitável. Para este fim, o algoritmo **JetValidation** conta a quantidade de eventos aceitos erroneamente pelos níveis de filtragem. Esse algoritmo deve ser usado com um conjunto de dados contendo apenas eventos de ruído e, neste caso, foram utilizadas simulações de jatos duplos

de 17 GeV de energia (média).

Um dos resultados da física quântica é que a taxa de eventos detectados pode ser estimada a partir da quantidade de eventos aceita, número total de eventos, luminosidade e seção de corte, como se pode ver na equação 4.1 [6] [27]. Com as duas últimas quantidades conhecidas para o conjunto de dados e as duas primeiras sendo calculadas na execução do algoritmo, pode-se ter uma estimativa da taxa de eventos em Hertz para essa assinatura. Este resultado é utilizado pelo algoritmo **JetValidation** para estimar a taxa de aquisição de amostras de ruído.

$$\text{Taxa de eventos} = \frac{N_{\text{após a filtragem}} \times \mathbb{L} \times \sigma}{N_{\text{antes da filtragem}}}, \quad (4.1)$$

onde:

$N_{\text{após a filtragem}}$ é o número de eventos após filtragem;

$N_{\text{antes da filtragem}}$ é o número de eventos antes da filtragem;

\mathbb{L} é a luminosidade;

σ é a seção de corte.

4.4.1 Implementação em C++

Os algoritmos descritos foram implementados em C++ na estrutura de classes mostrada na Figura 4.3, visando a evitar a repetição de código e sistematizar o tratamento de erros.

A classe **IValidationAlgo** é a classe-pai de todos os algoritmos desenvolvidos e implementa um sistema de tratamento de erros baseado em envio de **exceções** em C++. Esta classe é uma interface abstrata que pode detectar falhas no código e enviar mensagens de erro mais sugestivas, facilitando a depuração da implementação. A generalidade de sua implementação permite que qualquer algoritmo de validação deste projeto, ou avanço futuro deste, possa usufruir da sua estrutura.

A classe **FEXValidationAlgo** provê um conjunto de métodos frequentemente utilizados no pós-processamento de histogramas e no processamento dos eventos sob análise, particularmente para os algoritmos desenvolvidos neste projeto. Os algoritmos **HggValidation**, **SingleElectronValidation**, **SingleElectronOfflineValidation**, **ZeeOfflineValidation**, **ZeeValidation** e **JetValidation** são implementações particulares das classes abstratas mencionadas, que utilizam as fer-

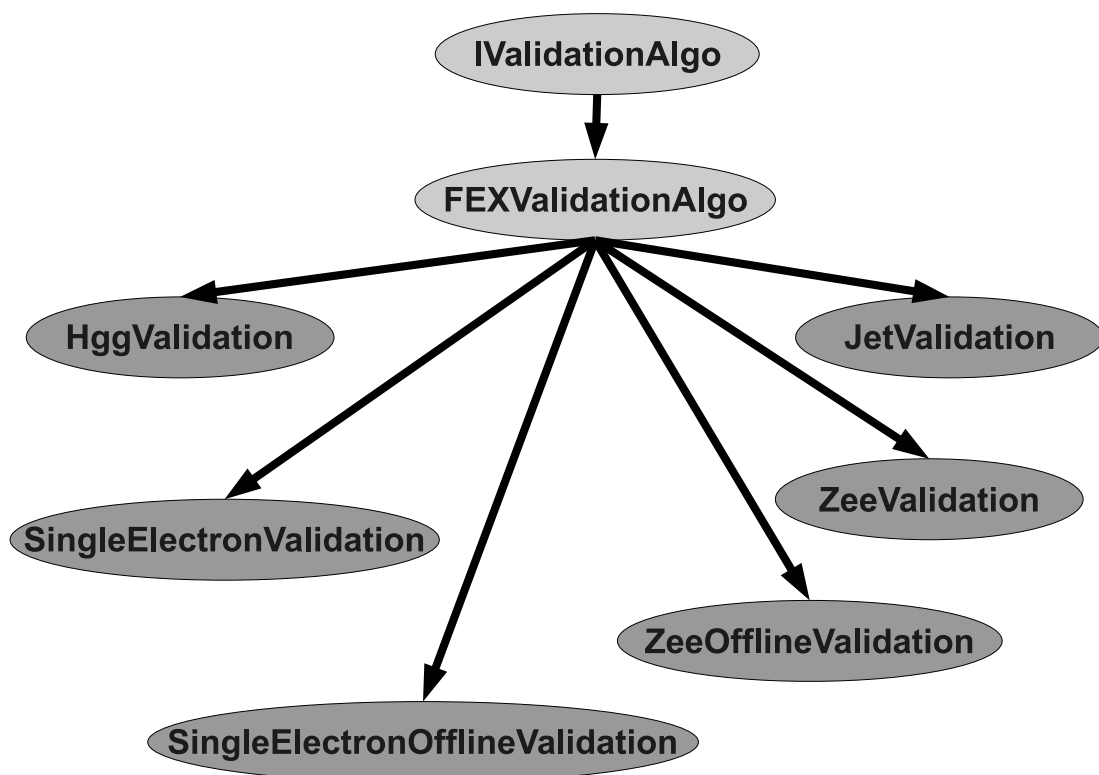


Figura 4.3: Estrutura de classes implementada para os algoritmos de validação em C++.

ramentas gerais providas pelas classes **FEXValidationAlgo** e **IValidationAlgo**, disponíveis através de herança.

Erros detectados nos algoritmos de validação são capturados pelo sistema de detecção de erros da classe **IValidationAlgo** e causam o término do processamento de eventos com uma mensagem de erro, facilitando a depuração do código.

4.5 Validação semi-automática através de métodos estatísticos

A primeira validação dos histogramas é sempre realizada por um perito no Sistema de Filtragem. A equipe de validação do sistema de filtragem, no canal Egamma, verifica estes histogramas, procurando inconsistências. Apesar da validação realizada por esta equipe, é útil ter um sistema que a auxilie a encontrar erros de forma mais rápida. Vários métodos estatísticos estão disponíveis para este fim.

O processo de validação é separado em passos, como descrito na Figura 4.2. O *script* **ProcessValidation.C** gera figuras específicas ao algoritmo executado, para facilitar a comparação dos validadores. Após a geração de figuras, o *script* **CKolmogorov.C** é executado no ambiente ROOT. Este script compara cada um dos histogramas gerados com histogramas gerados em uma versão anterior, previamente validada de forma subjetiva ou através deste mesmo método, gerando um número que indica o nível de similaridade entre os histogramas. Em algum ponto deve ocorrer uma validação subjetiva dos histogramas gerados pelos peritos no sistema de filtragem. Após este passo, pode-se utilizar o método descrito para comparar os resultados dos algoritmos de validação objetivamente.

Para formalizar a comparação entre os histogramas, formularemos o propósito dos métodos que seguem. Sejam $\hat{p}_A(x)$ e $\hat{p}_B(x)$ estimativas das funções de densidade de probabilidade das variáveis estocásticas independentes A e B . Deseja-se saber se as pdfs $\hat{p}_A(x)$ e $\hat{p}_B(x)$ são idênticas. O problema consiste, portanto, em verificar a hipótese nula:

$$H_0 : \hat{p}_A(x) = \hat{p}_B(x), \forall x \in \mathbb{R}. \quad (4.2)$$

4.5.1 Teste de Kolmogorov-Smirnov

O teste de Kolmogorov-Smirnov [28] compara duas funções de densidade de probabilidade (pdf) medindo a distância entre suas funções de distribuição cumulativa (cdf). A hipótese nula 4.2 é, portanto, verificada sob a estatística de Kolmogorov-Smirnov, descrita na equação:

$$\begin{aligned} D_{KS}(\hat{p}_A, \hat{p}_B) &= \max_x \left\| \int_{-\infty}^x (\hat{p}_A(t) - \hat{p}_B(t)) dt \right\| \\ &= D_{KS}(A, B) = \max_x \left\| \hat{P}_A(x) - \hat{P}_B(x) \right\| \end{aligned} \quad (4.3)$$

onde:

$$\begin{aligned} \hat{P}_A(x) &= \int_{-\infty}^x \hat{p}_A(t) dt \\ \hat{P}_B(x) &= \int_{-\infty}^x \hat{p}_B(t) dt \end{aligned}$$

Pode-se calcular a probabilidade de a hipótese nula ser satisfeita, usando a distribuição de Kolmogorov dada por:

$$P(z) = 2 \sum_{j=1}^{\infty} (-1)^{j-1} \exp(-2j^2 z^2) \quad (4.4)$$

com $z = D_{KS}(A, B)$.

O teste de Kolmogorov-Smirnov foi provado sob a hipótese de a distribuição de referência ser contínua. Testes empíricos [13] mostram que, para um número de amostras muito grande, pode-se obter uma boa aproximação de uma distribuição contínua e espera-se um resultado com alto grau de confiança do teste de Kolmogorov-Smirnov.

4.5.2 Divergência de Kullback-Leibler

A divergência de Kullback-Leibler [29] é uma métrica estatística com o propósito de medir a **distância** entre duas distribuições. Podemos usá-la, portanto, para testar a hipótese nula 4.2.

Um conceito importante na divergência de Kullback-Leibler é a entropia contida na variável estocástica. A entropia mede a quantidade de aleatoriedade em uma

distribuição e está definida na equação abaixo, para uma variável aleatória discreta X :

$$H(X) = - \sum_i P(X = a_i) \log P(X = a_i). \quad (4.5)$$

A entropia diferencial é uma extensão do conceito de entropia para variáveis contínuas e está definida na equação a seguir para uma variável aleatória contínua Y :

$$H(Y) = - \int_{-\infty}^{\infty} p_Y(\xi) \log(p_Y(\xi)) d\xi. \quad (4.6)$$

O conceito de entropia diferencial pode ser estendido para uma variável aleatória vetorial. Neste caso, toma a forma:

$$H(\vec{Y}) = - \int p_Y(\vec{\xi}) \log p_Y(\vec{\xi}) d\vec{\xi}. \quad (4.7)$$

O conceito de entropia mútua é definida em 4.8 e fornece um indicador da quantidade de informação contida em variáveis correlacionadas. Pensando em termos de representação da informação, a entropia mútua indica quanta compactação pode ser obtida para codificar uma variável aleatória vetorial, aproveitando a informação contida mutuamente nas variáveis. Nota-se que a entropia mútua é obtida removendo-se a quantidade de informação contida em cada variável X_1, X_2, \dots, X_n individualmente da entropia diferencial para a variável vetorial, o que deixa clara a interpretação deste conceito:

$$I(X_1, X_2, \dots, X_n) = \sum_{i=1}^n H(X_i) - H(\vec{X}), \quad (4.8)$$

onde: $\vec{X} = \begin{pmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{pmatrix}$.

A entropia mútua pode ser interpretada de maneira diferente, utilizando-se a definição de divergência de Kullback-Leibler, formulada na equação:

$$D_{KL}(\hat{p}_A, \hat{p}_B) = \int_{-\infty}^{\infty} \hat{p}_A(\xi) \log \frac{\hat{p}_A(\xi)}{\hat{p}_B(\xi)} d\xi, \quad (4.9)$$

onde: \hat{p}_A e \hat{p}_B são pdfs (estimadas) para A e B .

A divergência de Kullback-Leibler fornece uma noção de distância entre as pdfs \hat{p}_A e \hat{p}_B ⁶ [30]. Pode-se utilizar a divergência descrita em 4.9 para medir a independência entre as variáveis aleatórias de uma distribuição vetorial. Se as variáveis aleatórias X_1 e X_2 forem independentes, espera-se que seja possível fatorar a distribuição conjunta em $p_{X_1}p_{X_2}$. Portanto, uma medida da independência ou, de outra forma, dependência mútua entre X_1 e X_2 é dada pela distância entre $p_{\vec{X}}$ e $p_{X_1}p_{X_2}$, onde $\vec{X} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$.

Nota-se portanto que a divergência de Kullback-Leibler mede consistentemente a distância entre distribuições, dando-nos o resultado quantitativo e qualitativo que desejamos. Apesar das vantagens do método, esta medida tem a desvantagem de ser assimétrica, o que é algo não-intuitivo na interpretação dos resultados. Um método para simetrizar a métrica de Kullback-Leibler é definir a Divergência Simetrizada de Kullback-Leibler [31] como na equação:

$$D_{SKL}(\hat{p}_A, \hat{p}_B) = D_{KL}(\hat{p}_A, \hat{p}_B) + D_{KL}(\hat{p}_B, \hat{p}_A) \quad (4.10)$$

É possível mostrar facilmente que a métrica descrita em 4.10 é simétrica e não-negativa [31]. Pode-se verificar a hipótese nula em 4.2 definindo um valor máximo para o valor da divergência de Kullback-Leibler. Esta escolha é bastante complexa, pois a divergência de Kullback-Leibler não é limitada.

4.5.3 Divergência de Jensen-Shannon

Apesar das vantagens da divergência Simetrizada de Kullback-Leibler, não é possível determinar uma cota superior para seu valor em uma distribuição completamente geral. A Divergência de Jensen-Shannon [32], definida como:

$$D_{JS}(\hat{p}_A, \hat{p}_B) = \frac{D_{KL}(\hat{p}_A, \hat{p}_M) + D_{KL}(\hat{p}_B, \hat{p}_M)}{2} \quad (4.11)$$

onde $\hat{p}_M = \frac{\hat{p}_A + \hat{p}_B}{2}$, é simétrica, limitada e não-negativa. Pode-se mostrar ainda que esta métrica tem uma cota superior em 1, ou seja, $D_{JS}(\hat{p}_A, \hat{p}_B) \in [0, 1]$ [32].

⁶Note que não se fala da distância entre as variáveis aleatórias A e B , mas entre suas distribuições, que podem até mesmo ser as mesmas, apesar de A e B serem independentes.

Torna-se mais fácil verificar a hipótese nula 4.2 através da divergência de Jensen-Shannon, uma vez que esta é limitada. Um corte em certo valor da divergência pode funcionar como fator de decisão.

4.5.4 Negentropia quadrática

Um método mais simples de comparar as distribuições é construir uma métrica análoga à métrica L_2 utilizada em espaços euclidianos. A métrica L_2 é definida como:

$$D_{L_2} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$$

$$D_{L_2}(a, b) = \sqrt{\sum_i (a_i - b_i)^2}$$

$$a_i, b_i \in \mathbb{R}^n. \quad (4.12)$$

Observando a forma de 4.12, podemos construir uma forma análoga para as distribuições \hat{p}_A e \hat{p}_B como:

$$D_{NQ}(\hat{p}_A, \hat{p}_B) = \int_{-\infty}^{\infty} (\hat{p}_A(\xi) - \hat{p}_B(\xi))^2 d\xi, \quad (4.13)$$

que chamamos de Negentropia Quadrática.

A métrica descrita em 4.13 é não-negativa e simples. Um corte no valor desta métrica pode ser o fator de decisão para verificar a comprovação ou rejeição da hipótese nula 4.2.

4.6 Sensibilidade das métricas

Para verificar como as métricas descritas se comportam quando uma mudança ocorre nos histogramas, foi introduzida uma alteração específica em histogramas obtidos através da versão estável 14.5.0 do Athena e os valores das métricas foram comparados.

A metodologia utilizada consiste em alterar cada raia h_i do histograma H , para gerar os novas raias h'_i do histograma alterado H' . Seja $G(\mu, \sigma)$ uma amostra de uma variável aleatória gaussiana de média μ e desvio padrão σ . As duas formas utilizadas para alterar o histograma H são:

1. O novo valor de cada raia será $h'_i = h_i + G(0, h_i \times N)$.

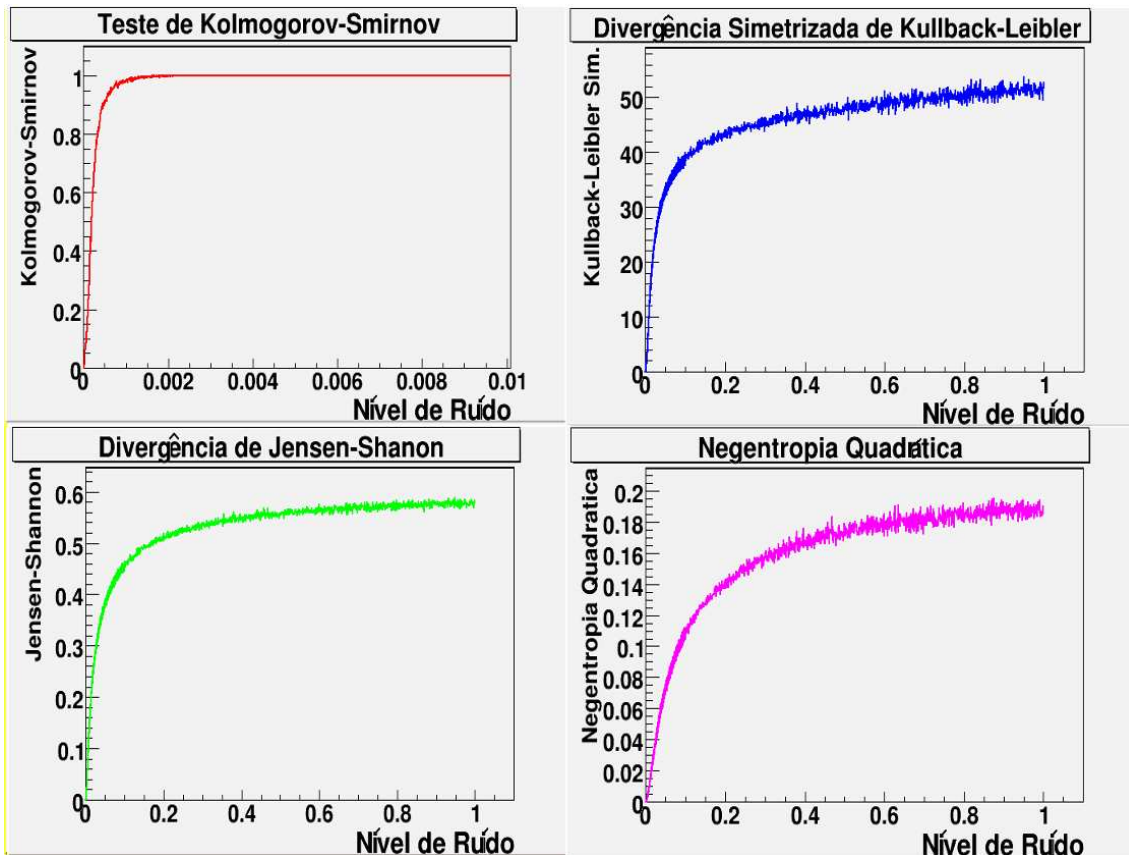


Figura 4.4: Efeito da adição de uma amostra gaussiana em cada uma das raia dos histograma nas métricas, para vários níveis de ruído.

2. O novo valor de cada raia será $h'_i = h_i \times G(1, h_i \times N)$.

O valor de N será denominado **nível de ruído** e indica o nível de influência das alterações aplicadas. A idéia das alterações é perturbar os histogramas de modo a simular uma mudança no *software*. No primeiro caso, adiciona-se, em cada raia, uma perturbação gaussiana. Pode-se controlar o nível de perturbação alterando o valor de N . No segundo caso, a perturbação é aplicada de forma multiplicativa.

Os testes foram realizados para vários valores de N . Para obter uma medida comum da **qualidade** de cada métrica, uma média dos resultados de cada métrica entre os histogramas utilizados foi calculada. A Figura 4.4 mostra o comportamento das métricas após a alteração descrita no item 1 anteriormente e a Figura 4.5, o efeito da alteração descrita no item 2. Nestas figuras, o valor zero indica máxima semelhança entre o histograma original e a versão alterada.

Note que teste de Kolmogorov-Smirnov tem uma resposta muito rápida logo que o **nível de ruído** aumenta e estabiliza em seu valor máximo (1) para ambos os

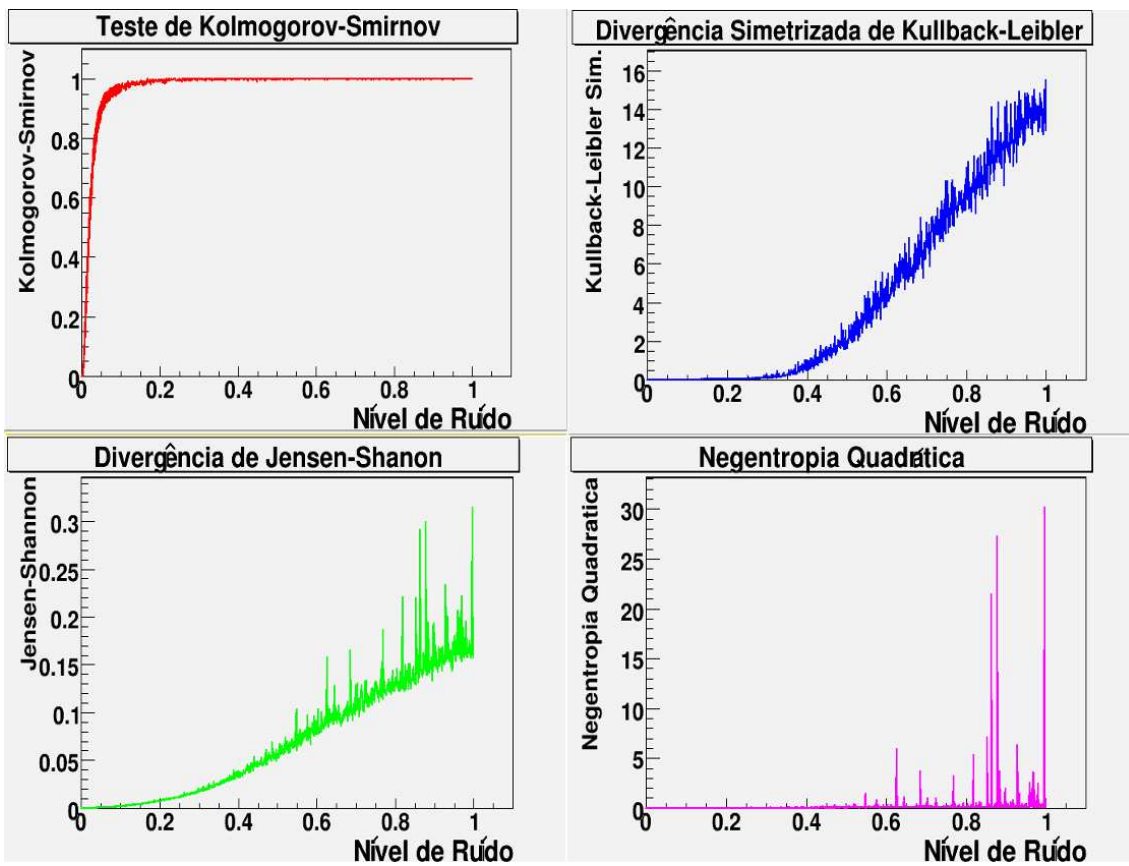


Figura 4.5: Efeito da multiplicação de uma amostra gaussiana em cada uma das raias dos histograma nas métricas, para vários níveis de ruído.

casos. Para o caso multiplicativo, a resposta de todas as métricas é bem mais lenta. A Negentropia Quadrática comporta-se de forma não esperada quando ocorre multiplicação pelas amostras gaussianas. Para o caso aditivo, a Negentropia Quadrática é pouco sensível às mudanças.

Também verificamos que, apesar da diferença em escala nas ordenadas, a divergência de Jensen-Shannon e a divergência simetrizada de Kullback-Leibler se comportam de forma semelhante, ou seja, a forma da resposta é bastante similar. Para o caso aditivo, a divergência de Jensen-Shannon se estabiliza próxima ao seu valor máximo (1). Aplicando-se, por exemplo, um corte no valor de 0,8 no caso aditivo para a divergência de Jensen-Shannon, pode-se rejeitar as diferenças entre histogramas que correspondem a alterações maiores que 20% de ruído.

4.7 Resultados

Visando a verificar o efeito de falhas no sistema de filtragem em casos reais, podemos criar problemas no sistema de filtragem e verificar como as métricas são alteradas. Os testes reproduzidos nesta seção foram baseados em problemas reais no sistema de filtragem.

A ideia consiste em executar o sistema de filtragem antes e após uma modificação possivelmente danosa, gerando histogramas, através do projeto desenvolvido, em cada execução. As métricas analisadas seriam, então, utilizadas para medir quantitativamente as diferenças causadas nos histogramas.

Um aspecto importante neste projeto é a **subjetividade**⁷ dos resultados. Apesar de ser possível analisar os histogramas apresentados de forma objetiva, apoiando-se nas teorias físicas, é bastante difícil ter uma avaliação precisa do comportamento dos histogramas apresentados devido à complexidade das análises. Nestes casos um conhecimento da física envolvida é utilizado para investigar o efeito das mudanças aplicadas, servindo como base para verificar o funcionamento das métricas estatísticas.

⁷O termo é usado com uma licença, uma vez que todas as análises são objetivas. A subjetividade mencionada aqui diz respeito à dificuldade de se quantificar objetivamente os resultados obtidos.

4.7.1 Remoção de calibração fina na análise do sistema de calorimetria pelo segundo nível de filtragem

Um primeiro teste foi a remoção de uma calibração feita implementada pelo algoritmo **T2CaloEgamma_eGamma**. Esta calibração utiliza constantes empíricas, calculadas durante alguns testes, para otimizar a estimativa da energia depositada no sistema de calorimetria eletromagnético⁸. Esperamos, portanto, verificar uma mudança (possivelmente pequena) nas energias estimadas no segundo nível de filtragem e, conseqüentemente, uma mudança em características importantes para os passos de processamento seguintes. Para realizar o teste, utilizaremos um conjunto de dados simulados correspondentes a eventos do bóson Z decaindo em elétrons.

A Figura 4.6, gerada pelos algoritmos de validação, mostra uma estimativa da pdf da energia transversa calculada pelo algoritmo **T2CaloEgamma_eGamma**⁹. A linha contínua, sempre em zero, mostra o histograma após a remoção da calibração, e a linha tracejada mostra a referência. A linha vertical tracejada mostra o valor de energia transversa onde é aplicado o corte pelo algoritmo de teste de hipótese. Verificamos que há uma diferença significativa na energia estimada e que esta diferença corresponderá a uma redução na eficiência de filtragem, uma vez que mais eventos estão na parte anterior ao corte após a mudança.

A Figura 4.7 mostra o valor absoluto da variável η . Novamente, a linha contínua representa a pdf após as mudanças, e a linha tracejada, que representa a referência, está sobreposta pela linha contínua. Uma vez que a calibração mencionada altera apenas o cálculo de energia, a posição da partícula estimada pelo algoritmo não é afetada, como se pode verificar no gráfico.

Uma vez que é possível obter a energia e a posição de interação com o sistema de calorimetria dos elétrons neste conjunto simulado, pode-se utilizar esta informação para medir a massa da partícula Z que neles decaiu. A Figura 4.8 mos-

⁸Esta calibração foi removida nas últimas versões estáveis do sistema de filtragem, sendo substituída por outros algoritmos de calibração mais precisos. Na versão do sistema utilizada para realizar estes testes, esta calibração ainda está presente.

⁹A posição detectada no sistema de filtragem é correlacionada com a informação da simulação no algoritmo de validação, portanto apenas verdadeiros elétrons constituem os eventos que geraram este gráfico.

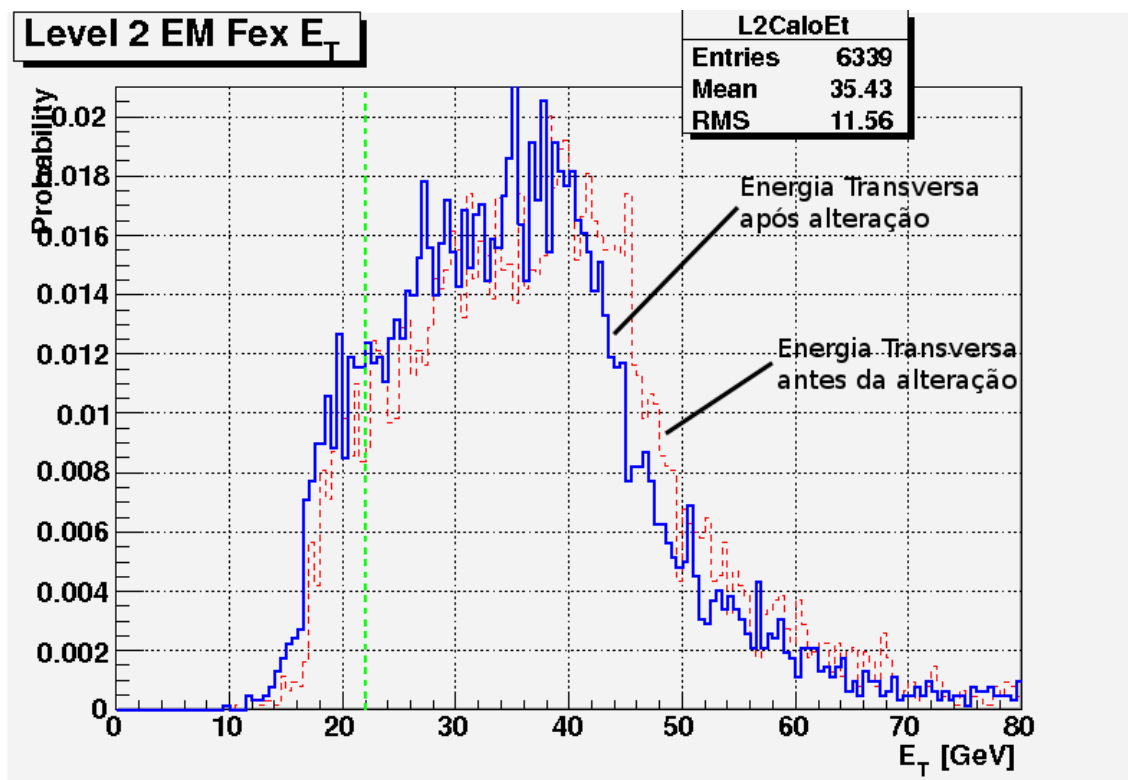


Figura 4.6: Pdf da energia transversa estimada no segundo nível de filtragem, calculada pelo algoritmo de validação. A linha contínua representa a pdf após alterações e a linha tracejada representa a pdf antes das alterações.

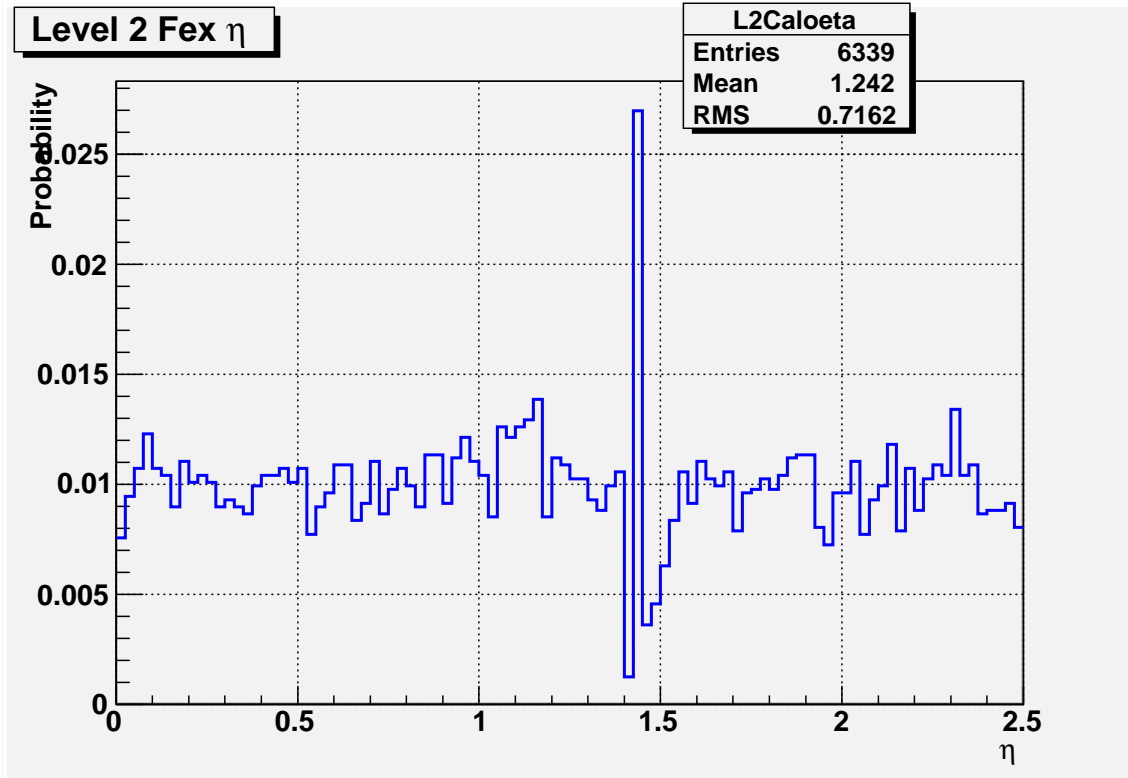


Figura 4.7: Pdf do valor absoluto da coordenada η estimada no segundo nível de filtragem. Não ocorre nenhuma alteração após mudanças.

tra a massa invariante do bóson Z para o conjunto amostral analisado. Neste cálculo foi assumido que a posição onde ocorreu o decaimento do bóson Z era o centro¹⁰ do detector ATLAS. Nota-se que há uma clara diferença nas curvas em linha contínua (após a mudança) e em linha tracejada (referência). A massa reconstruída no gráfico está menor após a remoção da calibração, indicando que a energia estimada para os elétrons foi reduzida.

Podemos, de posse destes resultados, averiguar as medidas realizadas utilizando as métricas estatísticas. A Tabela 4.1 lista a comparação entre os histogramas apresentados. Todas as métricas foram ajustadas para resultar zero em caso de máxima semelhança entre os histogramas.

Podemos verificar na tabela 4.1 que o teste de Kolmogorov-Smirnov é bem mais sensível a variações¹¹. Nenhuma diferença é detectada no histograma da coordenada η , como era esperado. Da mesma forma, a maior diferença é apontada

¹⁰Posição, no eixo de simetria do detector, onde se espera que ocorra a colisão.

¹¹O valor máximo para a discrepância entre histogramas no teste de Kolmogorov-Smirnov é 1.

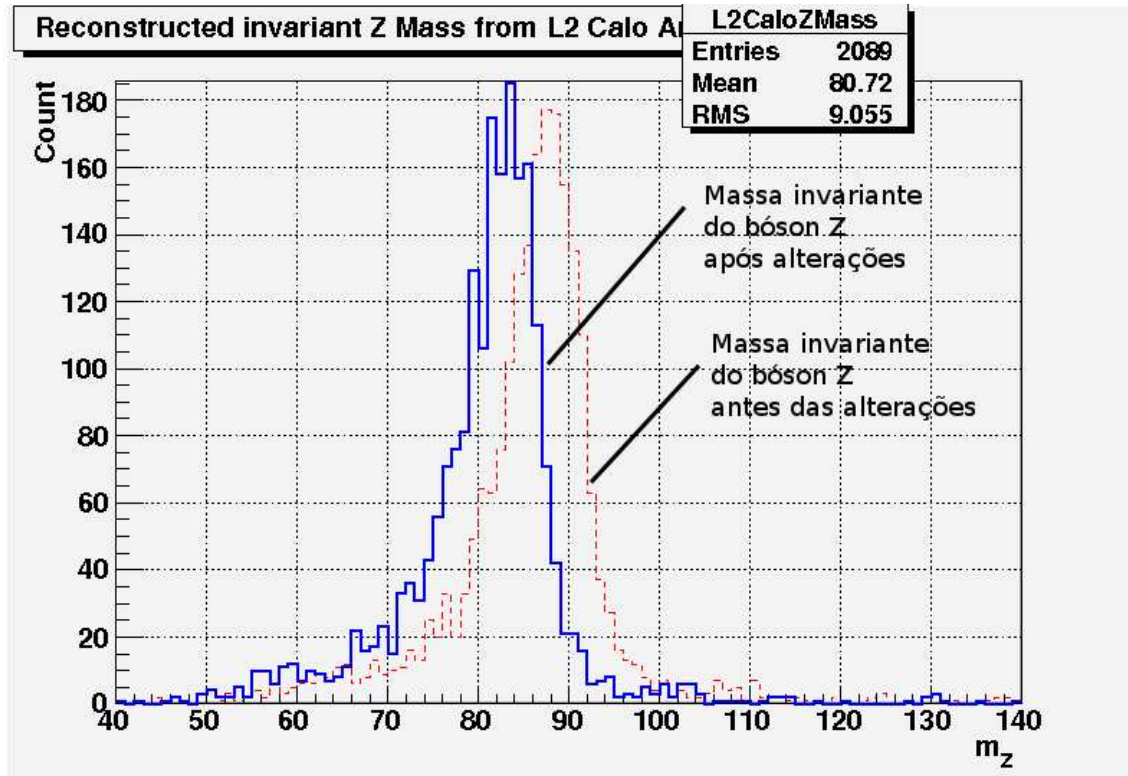


Figura 4.8: Histograma da massa invariante do bóson Z reconstruída utilizando apenas a informação de energia, η e ϕ estimada no algoritmo **T2CaloEgamma_eGamma**. O gráfico em linha contínua mostra o histograma após alterações e o gráfico em linha tracejada mostra o histograma antes da alteração no *software*.

Tabela 4.1: Medida das métricas para uma mudança na calibração da energia.

Histograma	Teste de Kolmogorov-Smirnov	Divergência de Jensen-Shannon	Divergência Simetrizada de Kullback-Leibler	Negentropia Quadrática
Energia Transversal	1,000	$1,769 \times 10^{-2}$	$1,609 \times 10^{-2}$	$7,160 \times 10^{-4}$
η	0,000	0,000	0,000	0,000
Massa do bóson Z	1,000	$1,669 \times 10^{-2}$	2,891	$2,845 \times 10^{-2}$

para o histograma de massa invariante do bóson Z, no qual, visualmente, a diferença é maior. O teste de Jensen-Shannon tem valor máximo em 1, como o teste de Kolmogorov-Smirnov, mas não é tão sensível a alterações.

4.7.2 Assimetria na coordenada ϕ

Uma outra possibilidade de erro é o cálculo incorreto da coordenada ϕ . Isto pode acontecer devido a falhas na configuração do sistema de filtragem, e causa um tratamento assimétrico da informação fornecida pelo detector. Este efeito já foi observado com um conjunto de dados simulado, em decorrência de um erro na simulação dos eventos.

Para o teste corrente, este efeito foi simulado alterando-se o valor da coordenada ϕ a cada dois eventos na informação do segundo nível de filtragem, usando-se o valor absoluto desta variável, ao invés do seu valor real. Espera-se uma redução na eficiência do sistema de filtragem na região de ϕ negativo, pois a associação feita entre o segundo nível e o primeiro será, conseqüentemente, feita de forma incorreta. Neste teste, foram utilizadas as mesmas condições que as da sub-seção anterior. O conjunto simulado utilizado foi do bóson Z decaindo em elétrons.

A Figura 4.9¹² mostra um histograma do desvio relativo entre a variável ϕ calculada no segundo nível de filtragem e a posição real onde ocorreu interação com o detector (obtida a partir da informação de simulação) antes da alteração realizada na variável ϕ .

A Figura 4.10 representa a mesma variável após a alteração. Nota-se que a forma da distribuição não se altera muito, mas uma análise mais detalhada mostra que a contagem de eventos, na ordenada, foi reduzida. O ponto de máximo antes da mudança é ~ 2200 , e após é ~ 1600 . O número de entradas nos histogramas, indicado na caixa superior à direita, no campo *Entries*, também foi drasticamente alterado¹³. Esta alteração ocorre pois os outros eventos têm um desvio relativo bastante grande (em módulo), fora da faixa de valores representada no histograma. Isto é, os eventos afetados pela alteração têm desvio relativo em ϕ maior que 100%,

¹²A Colaboração ATLAS costuma referir informalmente a este desvio relativo como **resolução**; por isto o termo foi utilizado.

¹³Utilizou-se o mesmo conjunto amostral em ambos os casos.

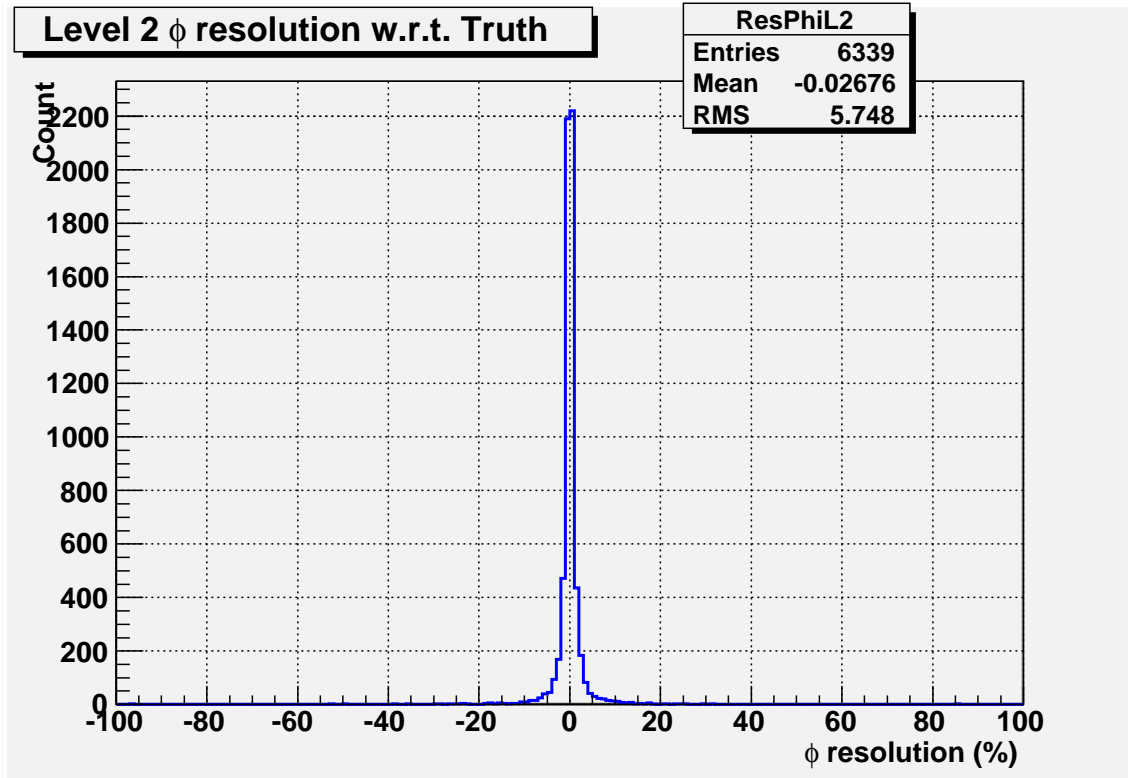


Figura 4.9: Desvio relativo entre a variável ϕ estimada pelo algoritmo **T2CaloEgamma_eGamma** e a coordenada ϕ real antes da alteração.

em módulo.

A Figura 4.11 mostra no gráfico superior à esquerda a eficiência do sistema de filtragem em diferentes regiões da coordenada η . Nota-se uma queda em eficiência em torno de $\eta \sim 1, 3$; isto ocorre porque esta região não está completamente coberta pelo sistema de calorimetria. O gráfico inferior mostra a eficiência para diferentes faixas de energia transversa, e o gráfico no topo à direita mostra a eficiência em relação à coordenada ϕ . Ocorre uma grave queda em eficiência na região de ϕ negativo, como era esperado.

Os gráficos de eficiência para energia transversa, η e ϕ , são gerados dividindo-se raia-a-raia os respectivos histogramas destas variáveis após serem aceitas pelo segundo nível de filtragem pelo histograma das mesmas informado pela Verdade. Como os gráficos mostrados não podem ser compreendidos como pdfs, a comparação utilizando métodos estatísticos é feita entre os histogramas de energia, η e ϕ , após a filtragem.

A Tabela 4.2 mostra o valor das métricas para os histogramas mencionados.

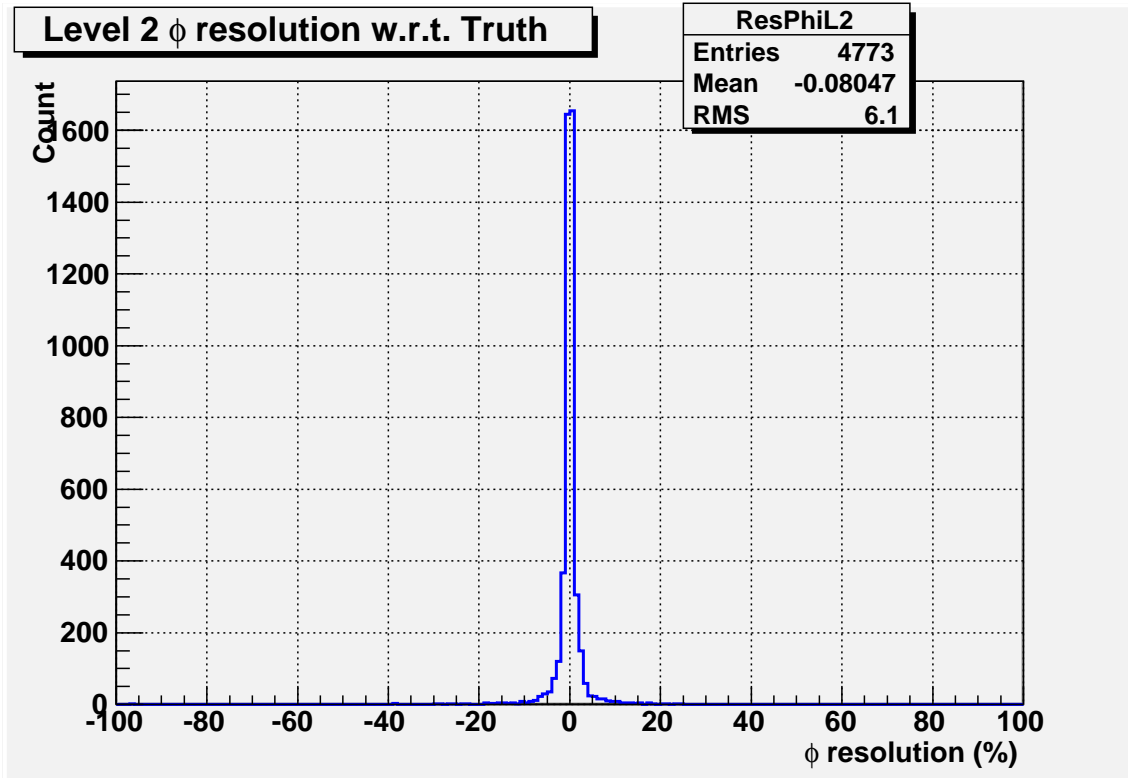


Figura 4.10: Desvio relativo entre a variável ϕ estimada pelo algoritmo **T2CaloEgamma_eGamma** e a coordenada ϕ real após a alteração.

Apesar da diferença na estatística, a forma dos histogramas de desvio relativo em ϕ se altera pouco; portanto, após a normalização, as métricas não detectam bem a diferença. Por outro lado, essa diferença é bastante clara no histograma da coordenada ϕ após a filtragem no nível 2, uma vez que foi a diferença neste histograma que afetou diretamente a eficiência em ϕ mencionada anteriormente. Novamente, o teste de Kolmogorov-Smirnov é mais sensível no caso do histograma de ϕ , onde o valor do teste é 1, indicando máxima discrepância. Por outro lado, a divergência de Jensen-Shannon e a divergência de Kullback-Leibler detectam mudanças no histograma de desvio relativo em ϕ com mais sensibilidade.

4.7.3 Falha na detecção de um elétron

Outra possível falha possível é a não detecção de uma das partículas. Um erro na infraestrutura do sistema de filtragem poderia ter esta consequência, que tem efeito particularmente grave em conjuntos de dados simulados onde há um decaimento em dois elétrons. Para o teste que será discutido, foi utilizado o mesmo

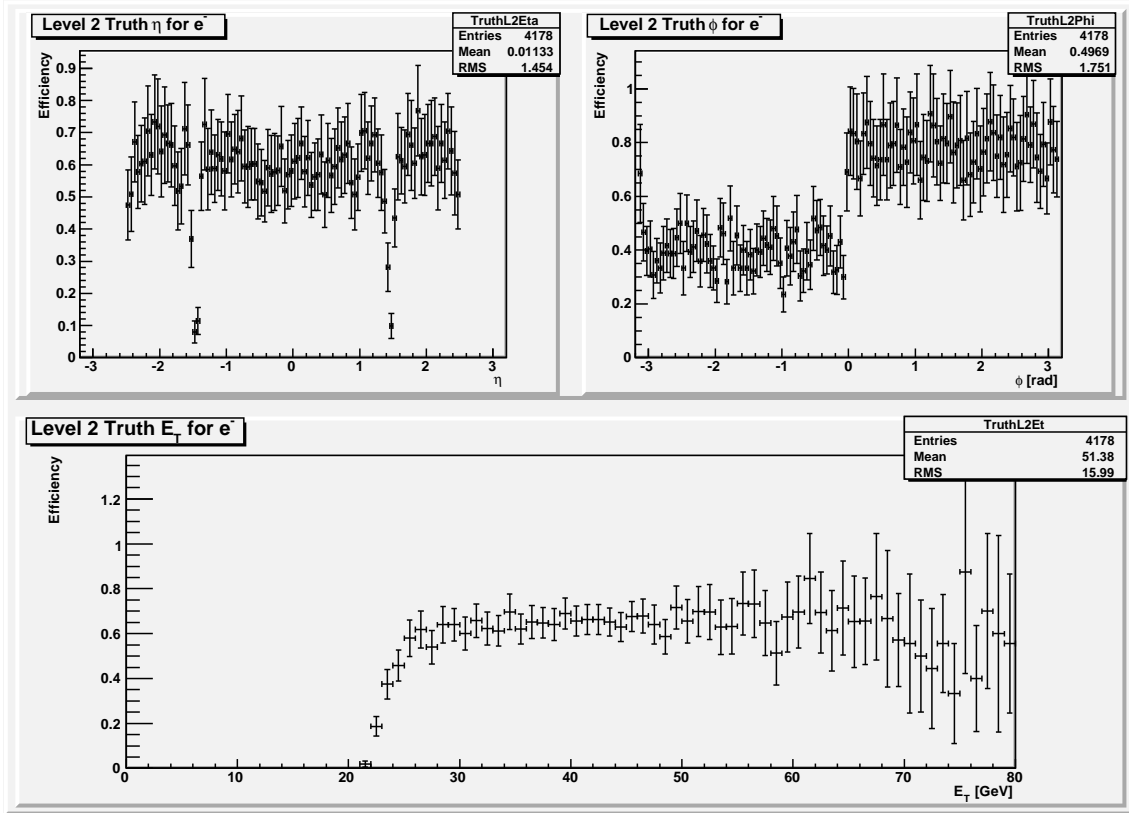


Figura 4.11: Eficiência em relação à informação da simulação, após a filtragem pelo segundo nível (apenas passo que utiliza informação de calorimetria). Efeito da mudança aplicada em ϕ .

Tabela 4.2: Medida das métricas para uma mudança no sinal de ϕ no segundo nível de filtragem.

Descrição do histograma	Teste de Kolmogorov-Smirnov	Divergência de Jensen-Shannon	Divergência Simetrizada de Kullback-Leibler	Negentropia Quadrática
Desvio em ϕ	$6,999 \times 10^{-6}$	$2,907 \times 10^{-3}$	$6,549 \times 10^{-2}$	$1,064 \times 10^{-4}$
ϕ após nível 2	1,000	$2,224 \times 10^{-2}$	$8,975 \times 10^{-2}$	$9,426 \times 10^{-4}$

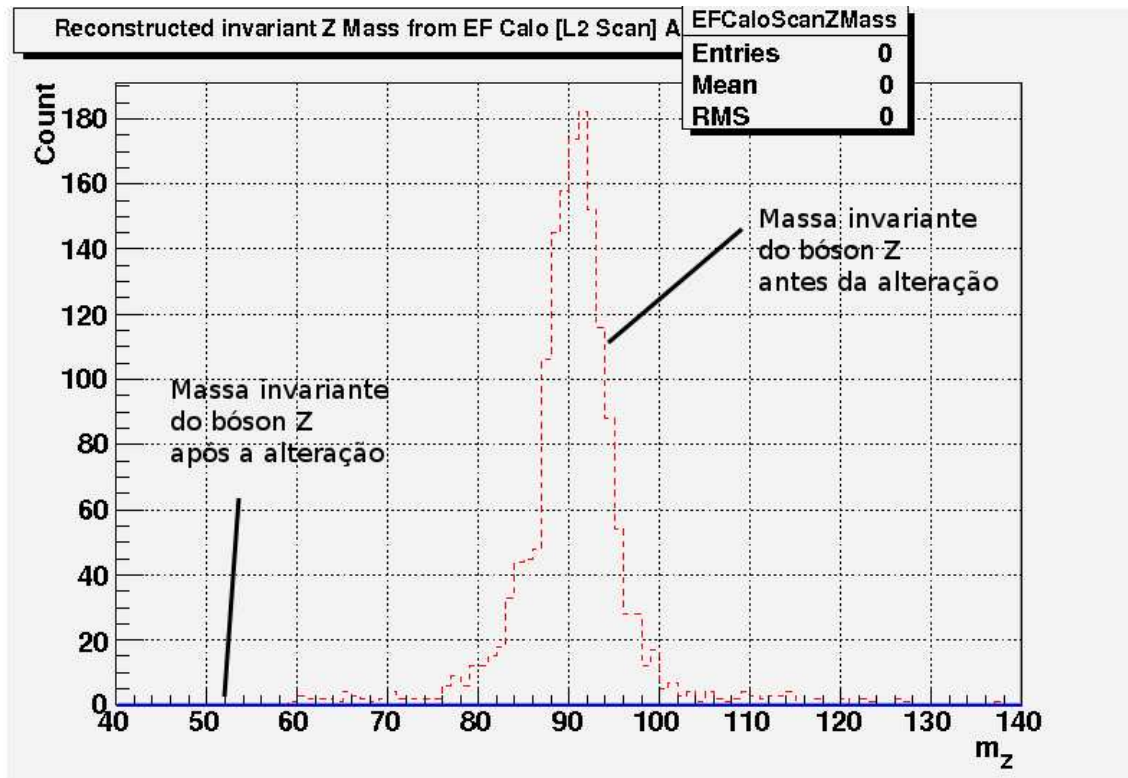


Figura 4.12: Massa invariante do bóson Z reconstruída com a informação do Filtro de Eventos. A linha contínua (sempre em zero) mostra o histograma após a alteração. A linha tracejada mostra o histograma antes da alteração. Foi assumida a hipótese de o decaimento ocorrer no centro do detector ATLAS.

conjunto de dados simulados que nas subseções anteriores, no qual ocorre um decaimento do bóson Z em elétrons.

Neste caso, a consequência direta da ausência de uma das partículas é a incapacidade de se reconstruir a massa invariante da partícula Z. A Figura 4.12 mostra, na linha contínua, o histograma de massa invariante do bóson Z após a mudança. A linha tracejada é a referência. Neste caso, a diferença é completa e esperamos que não haja similaridade nenhuma, em todas as métricas.

A Figura 4.13 mostra o efeito da alteração no histograma de momento transversal calculado nas partículas. A linha contínua representa o histograma após a alteração, e a linha tracejada mostra a referência. Nota-se que a diferença na pdf é, visualmente, pequena. No entanto, o número total de entradas no histograma de referência, representado isoladamente na Figura 4.14, é bem maior (5477 eventos no histograma de referência, e apenas 3476 no histograma após alterações). Neste

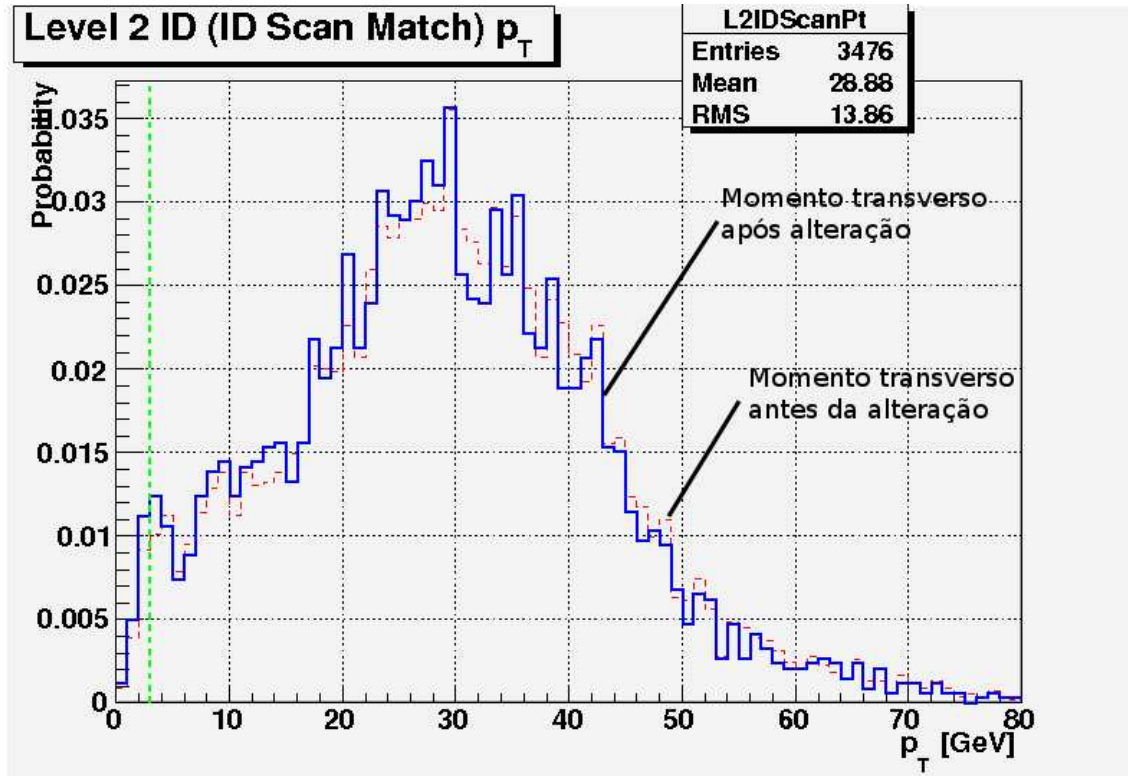


Figura 4.13: Momento transverso estimado pelo algoritmo `TrigL2IDSCAN_eGamma`, no nível 2. A linha tracejada mostra o histograma antes das alterações e a linha contínua mostra o histograma após alterações.

histograma, a pdf não foi alterada significativamente, mas a quantidade de dados foi alterada.

A Tabela 4.3 mostra os valores das métricas para os histogramas analisados. Podemos observar que todas as métricas acusam a diferença no histograma de massa invariante do bóson Z, como esperado. No caso do histograma de momento transverso, o teste de Kolmogorov-Smirnov apresenta a maior sensibilidade para as mudanças.

4.8 Obtenção de resultados via interface Web

Um sistema Web foi desenvolvido para apresentar os resultados da validação das versões noturnas do sistema de filtragem. A interface permite a visualização mais simples dos resultados para cada nova versão analisada pelo sistema de validação. A interface desenvolvida não precisa ser muito complexa e foi, portanto, desenvolvida

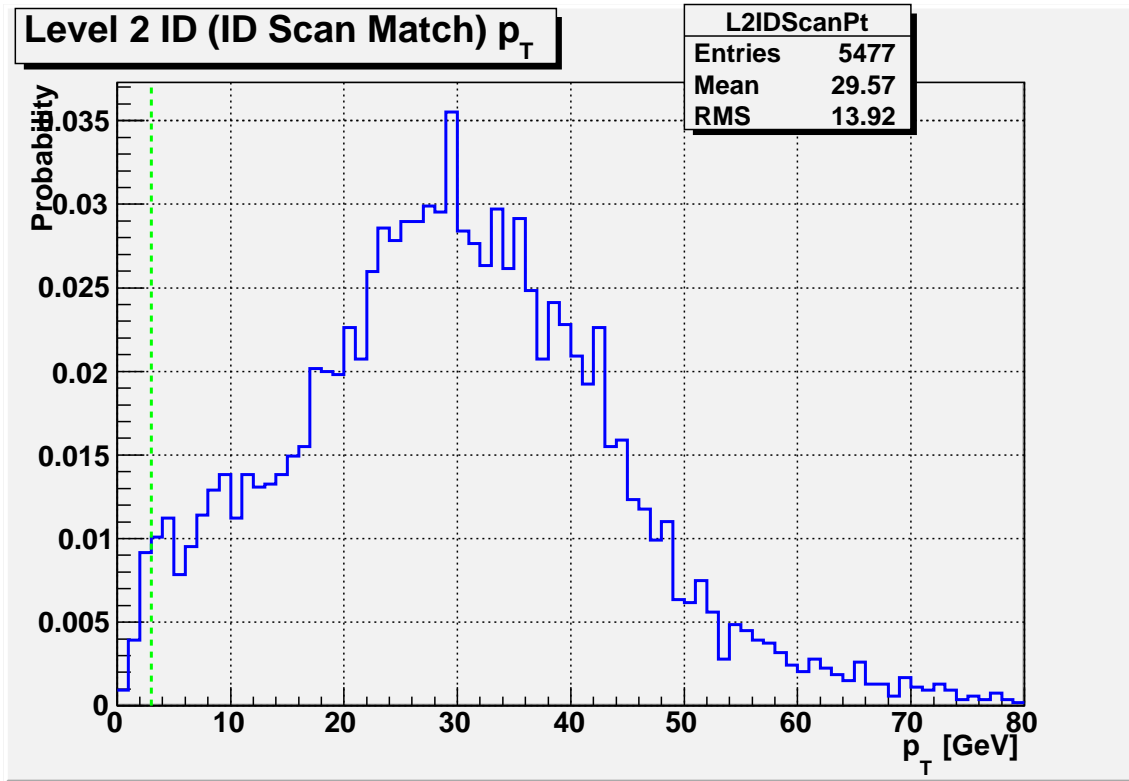


Figura 4.14: Histograma de referência (antes da alteração) do momento transverso estimado pelo algoritmo `TrigL2IDSCAN_eGamma`, no nível 2.

Tabela 4.3: Medida das métricas para a falha de detecção de uma das partículas em uma simulação do decaimento do bóson Z em um par elétron-pósitron. Os valores na primeira linha não são resultados das medidas estatísticas, uma vez que não se pode calculá-las em um histograma vazio; neste caso, os valores mostrados são retornados pelo *script* CKolmogorov.C para indicar uma falha.

Descrição do histograma	Teste de Kolmogorov-Smirnov	Divergência de Jensen-Shannon	Divergência Simetrizada de Kullback-Leibler	Negentropia Quadrática
Massa do bóson Z calculada com energia, η e ϕ do Filtro de Eventos	1,000	1,000	9999	9999
p_T calculado por <code>TrigL2IDSCAN</code>	$8,711 \times 10^{-1}$	$2,249 \times 10^{-3}$	$5,300 \times 10^{-2}$	$1,489 \times 10^{-4}$

utilizando a linguagem interpretada PHP [33]. Ela está disponível em <http://www.cern.ch/info-trigegammavalidation/>.

A Figura 4.15 mostra um histograma obtido na versão noturna de 08/02/2009 sobreposto com um histograma de referência (tracejado) após o corte do algoritmo de validação. À esquerda, pode-se ver um menu em árvore. O primeiro nível do menu contém um item que permite selecionar a versão do *software*, cujos resultados estão sendo apresentados (*List all releases*), um item com informações gerais sobre a eficiência (*Overall results*) e, a seguir, itens representando os diferentes procedimentos de teste realizados para esta versão. Após clicar no item $Z \rightarrow e + e$ *RDO w.r.t. Truth*, para visualizar os resultados para o procedimento de teste que utiliza dados simulados do decaimento do bóson Z, expande-se uma lista de sub-itens com os resultados da análise para cada parte do sistema de filtragem. No lado direito da figura, o primeiro histograma (*Level 2 (ID Scan Match) p_T*) mostra o valor do momento transversal, calculado no segundo nível de filtragem para o conjunto de dados de simulação. Abaixo deste histograma há outros histogramas disponíveis para consulta.

A Figura 4.16 mostra, em uma página com mesma estrutura, os gráficos de eficiência para a filtragem em relação às variáveis verdadeiras para os elétrons gerados na simulação. No lado direito da figura, verificamos três gráficos. A eficiência deste nível de filtragem para várias regiões na coordenada η está no canto superior esquerdo. A eficiência para vários valores da coordenada ϕ está no canto superior direito. Abaixo, há a eficiência para vários valores da energia transversal.

Devido à geometria simétrica do detetor, esperamos que a eficiência em relação à coordenada ϕ seja uniforme. Para a coordenada η , por outro lado, sabe-se que a região em torno de $|\eta| = 1,3$ não está completamente coberta pelos calorímetros, ou seja, a estimativa de energia nesta região não é tão boa quanto em outras partes do detetor, logo a eficiência nesta região deve sofrer um impacto negativo. Para o gráfico de eficiência em relação à energia transversal, esperamos uma resposta aproximadamente uniforme após o corte nesta variável, que ocorre em 22 GeV. Utilizando estas análises subjetivas, podemos utilizar a informação disponível no sistema Web para verificar erros no desempenho físico do *software*.

Nota-se na Figura 4.16 que as barras de erro são relativamente grandes. Isso

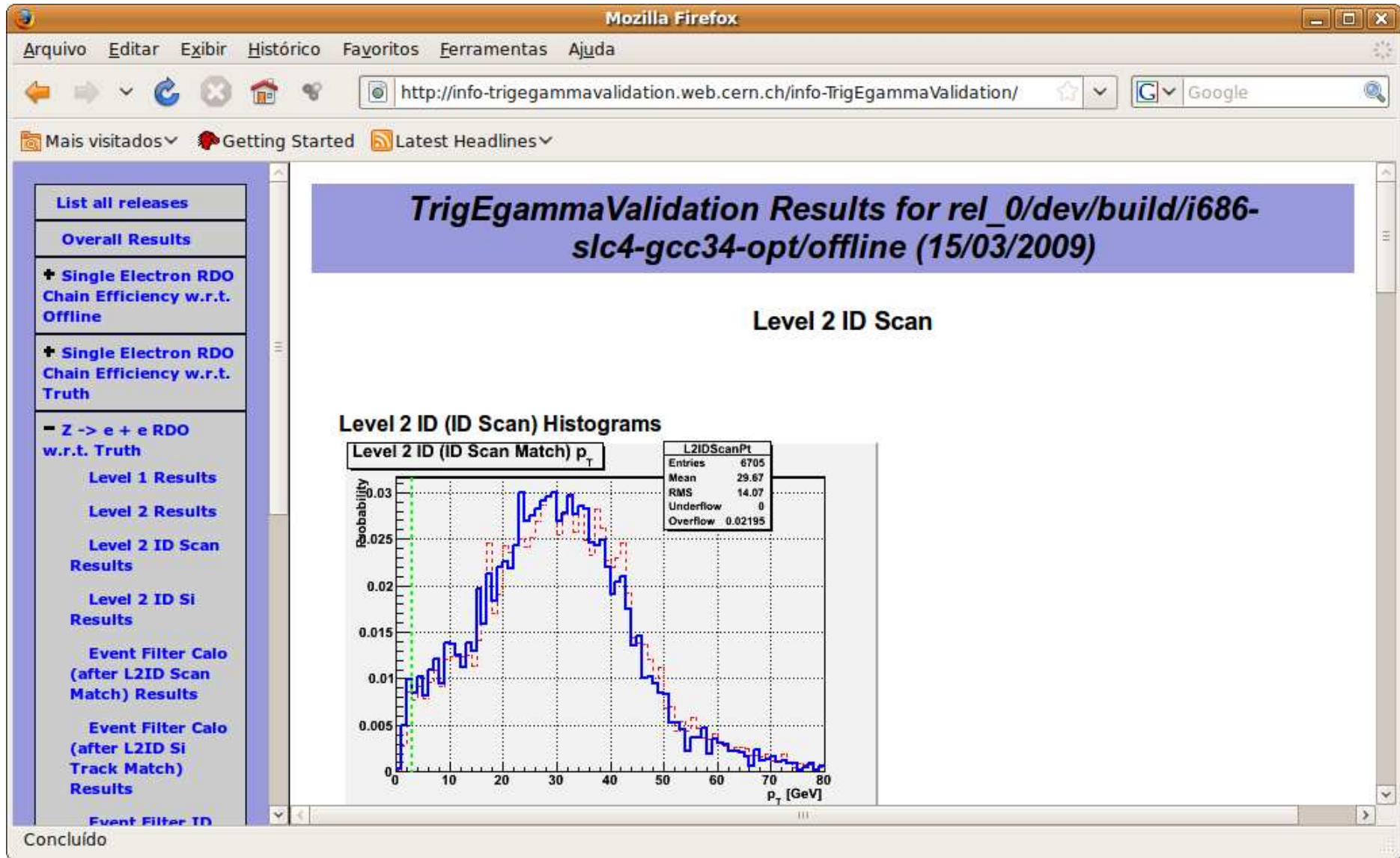


Figura 4.15: Interface Web mostrando histogramas gerados para conjuntos simulados de Z decaindo em um par elétron-pósitron após a filtragem do nível 2.

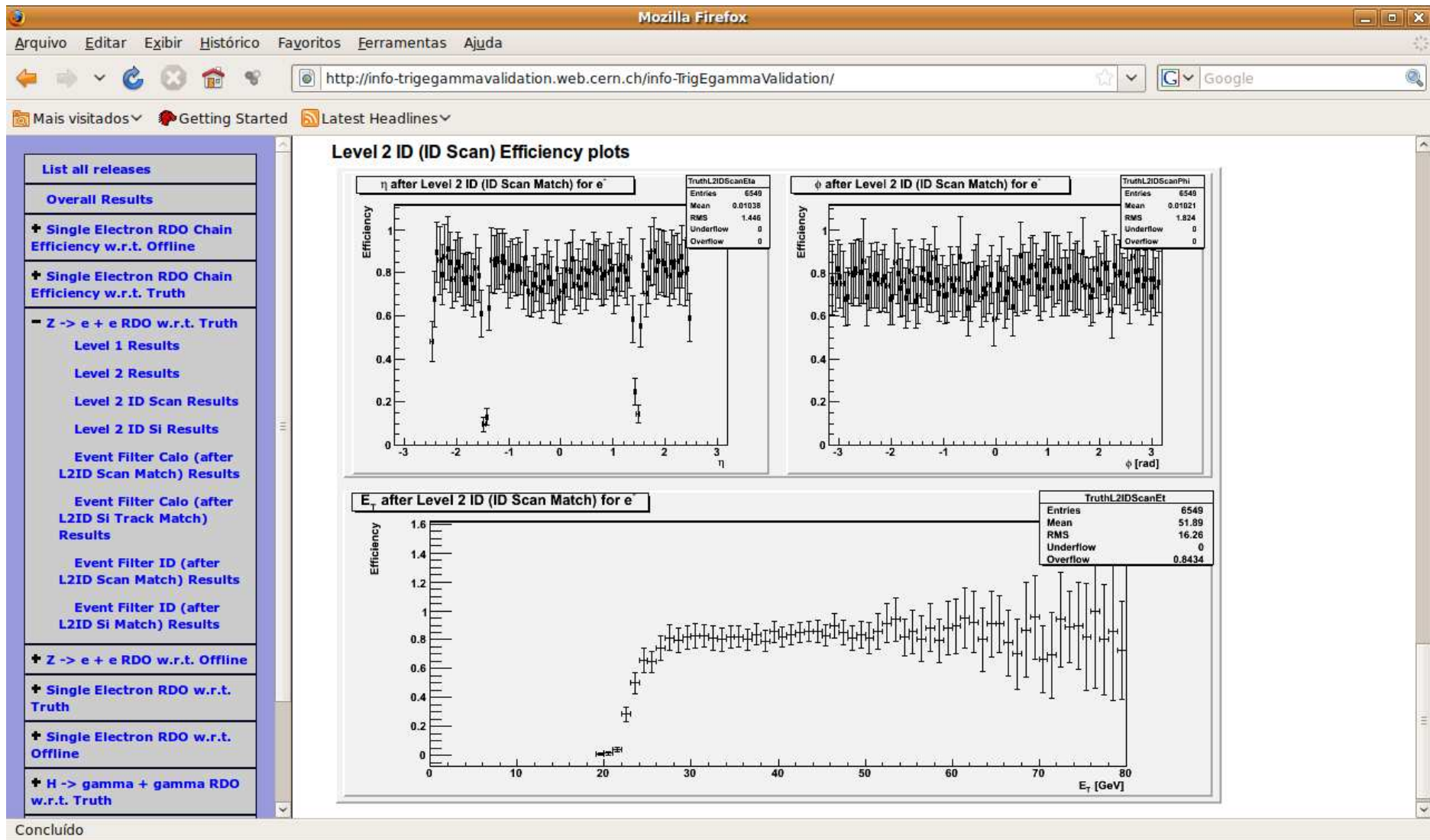


Figura 4.16: Interface Web mostrando curvas de eficiências medidas para conjuntos simulados de Z decaindo em um par elétron-pósitron após a filtragem do nível 2.

ocorre porque o trabalho de teste deve ser executado em, no máximo, um dia, pois no dia seguinte uma nova versão estará disponível e os recursos computacionais dedicados à validação serão direcionados para a versão seguinte. Desta forma, devido às limitações no tempo de execução dos algoritmos de validação, poucos eventos de dados de simulação são utilizados nos procedimentos de teste. O efeito é uma maior barra de erro.

A Figura 4.17 mostra uma página com os resultados do teste de comparação, utilizando cada uma das quatro métricas descritas anteriormente. A tabela à direita contém o nome do histograma comparado e o valor numérico de cada uma das métricas utilizadas na comparação. Valores do teste de Kolmogorov-Smirnov e da divergência de Jensen-Shannon estão atualmente sendo utilizados para realizar a validação semi-automática: se a medida de similaridade destes (onde 100% indica similaridade máxima) for menor que 90%, a equipe de validação é alertada. Estudos ainda estão sendo feitos para ajustar estes valores para um valor ótimo.

A configuração da interface é feita através de um arquivo XML, contendo a descrição dos menus e a caracterização das imagens. O arquivo XML é bastante versátil e permite a visualização de qualquer tipo de imagem e formatação de menus, além de ser extensível, caso novos procedimentos de testes sejam criados.

Para verificar quais são as versões e análises disponíveis do *software*, o sistema de páginas em PHP lista a árvore de diretórios gerada pela infraestrutura *Run Time Tester*, que tem uma estrutura específica para cada versão e análise. O arquivo de configuração XML indica o nome dos procedimentos de teste e figuras que devem ser mostrados e mapeia estes procedimentos e figuras para nomes mais claros durante a visualização. Cada página mostrada à direita das figuras é gerada por uma classe PHP cujo propósito é representar aquela estrutura, com os detalhes descritos no arquivo XML.

4.9 Utilização da rede Grid para testes em versões estáveis do Sistema de Filtragem

Pode-se notar que os gráficos na seção anterior têm barras de erro relativamente grandes. Para melhorar a quantidade de dados analisada, pode-se executar o

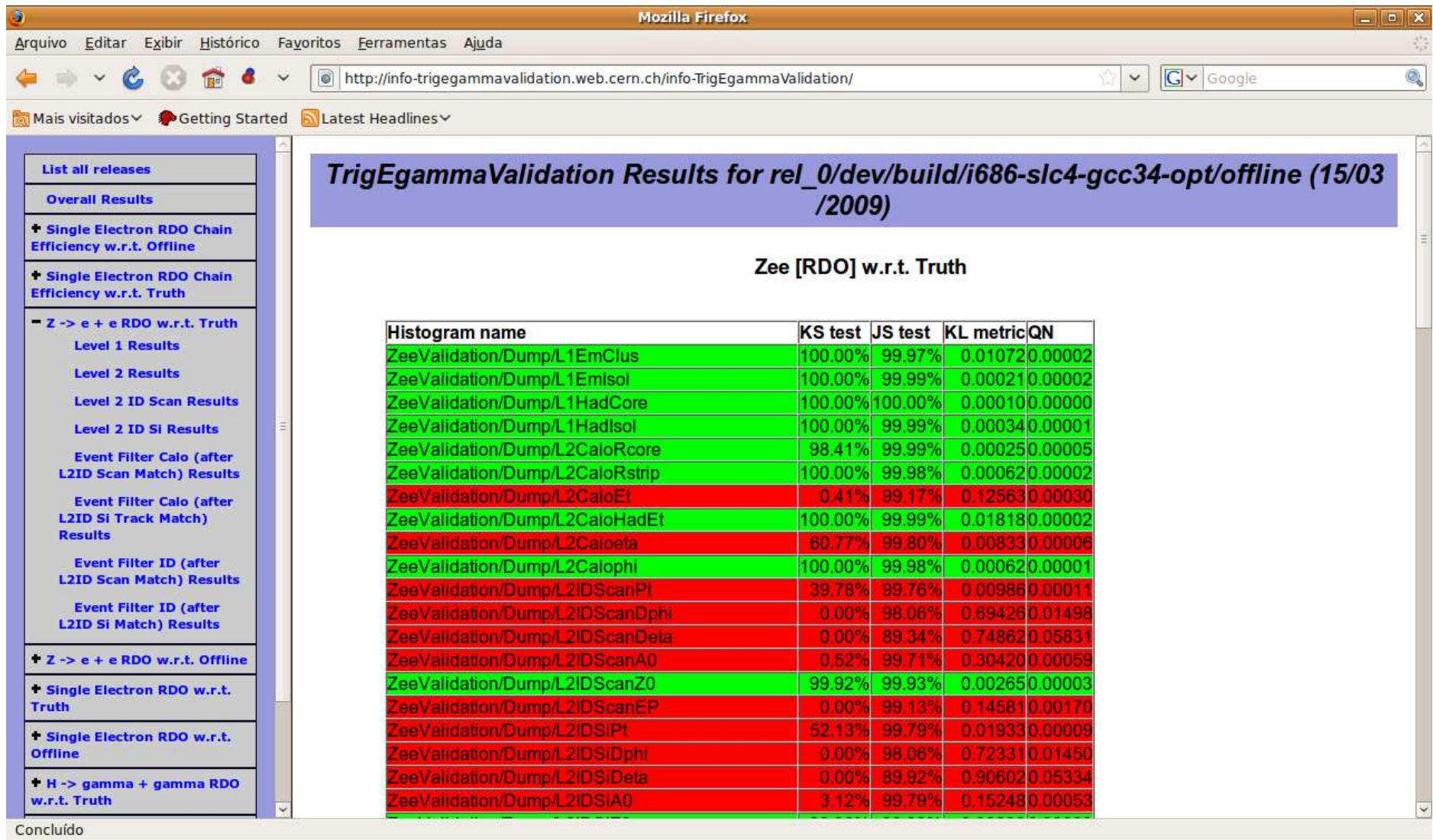


Figura 4.17: Interface Web mostrando o resultado da comparação entre histogramas para conjuntos simulados de Z decaindo em um elétron-pósitron.

ambiente de validação desenvolvido em um ambiente de Computação em Grelha [34] (rede Grid), que é constituído de centros computacionais ligados em várias partes do planeta, permitindo testes mais elaborados com análises que seriam bastante demoradas com os recursos computacionais tradicionais.

A infraestrutura desenvolvida utiliza as ferramentas Panda [35] para submeter os trabalhos de teste a qualquer rede Grid¹⁴. Utiliza-se a ferramenta **Pathena** para enviar os trabalhos de teste e gerar os histogramas para uma versão estável. Uma breve ilustração do funcionamento do Pathena é dada na Figura 4.18. O Pathena funciona gerando um pacote compactado da área de trabalho local contendo as modificações e opções de trabalho (*job options*) do usuário, durante o passo **buildJob**, ilustrado na figura. Em seguida, esse pacote fica disponível para ser executado em vários computadores conectados à rede Grid, no passo **runAthena** da Figura 4.18, cada um com uma cópia parcial ou total do conjunto de dados de entrada requisitado. Após a execução do Athena na rede Grid, os arquivos ROOT com os histogramas de saída ficam disponíveis em repositórios nesta rede, prontos para serem obtidos. Após o término da execução, uma carta eletrônica é enviada ao usuário que submeteu o processo, notificando do sucesso ou falha de execução.

Após a geração dos arquivos de saída na Grid¹⁵, é preciso obtê-los, uni-los e realizar o mesmo pós-processamento que se realiza na infraestrutura do *Run Time Tester*. Para automatizar o pós-processamento, a ferramenta **Psequencer** é utilizada. A Figura 4.19 mostra como o Psequencer processa os trabalhos. Nesta figura, o trabalho é enviado, utilizando o Pathena no passo **PANDA**. A seguir, o Psequencer aguarda uma notificação por email no passo **Notificação**, realizada automaticamente pelo Pathena quando o trabalho é finalizado. Quando a notificação é recebida, o Psequencer executa, no item **Passo 2**, o conjunto de ações descritas pelo usuário para o pós-processamento.

Para este trabalho em particular, a lógica implementada através do Psequencer está ilustrada na Figura 4.20. Notamos que a lógica de execução é bastante

¹⁴Desde que esta seja suportada pelas ferramentas Panda. Neste projeto foi utilizada com frequência a rede Grid BNL/USATLAS.

¹⁵Geralmente, alguns trabalhos de teste precisam ser re-executados, pois ocorrem falhas na execução do Athena na rede Grid. Isto ocorre frequentemente devido ao tempo limite de obtenção dos arquivos de entrada, que estão armazenados em fita.

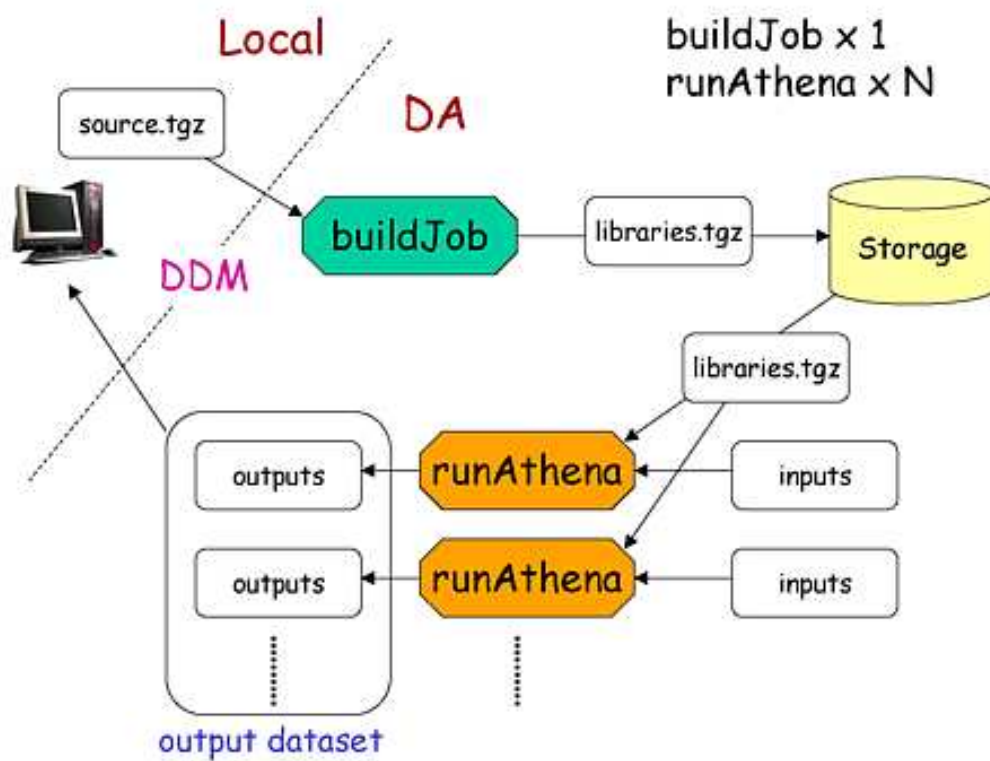


Figura 4.18: Envio de trabalhos à rede Grid através do Pathena.

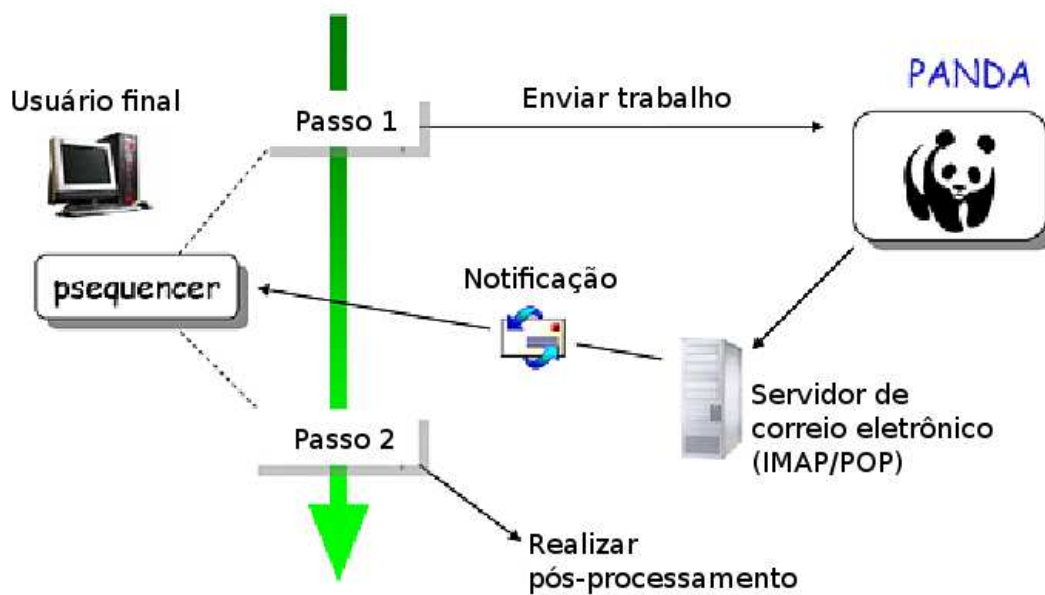


Figura 4.19: Envio de trabalhos e pós-processamento através do Psequencer.

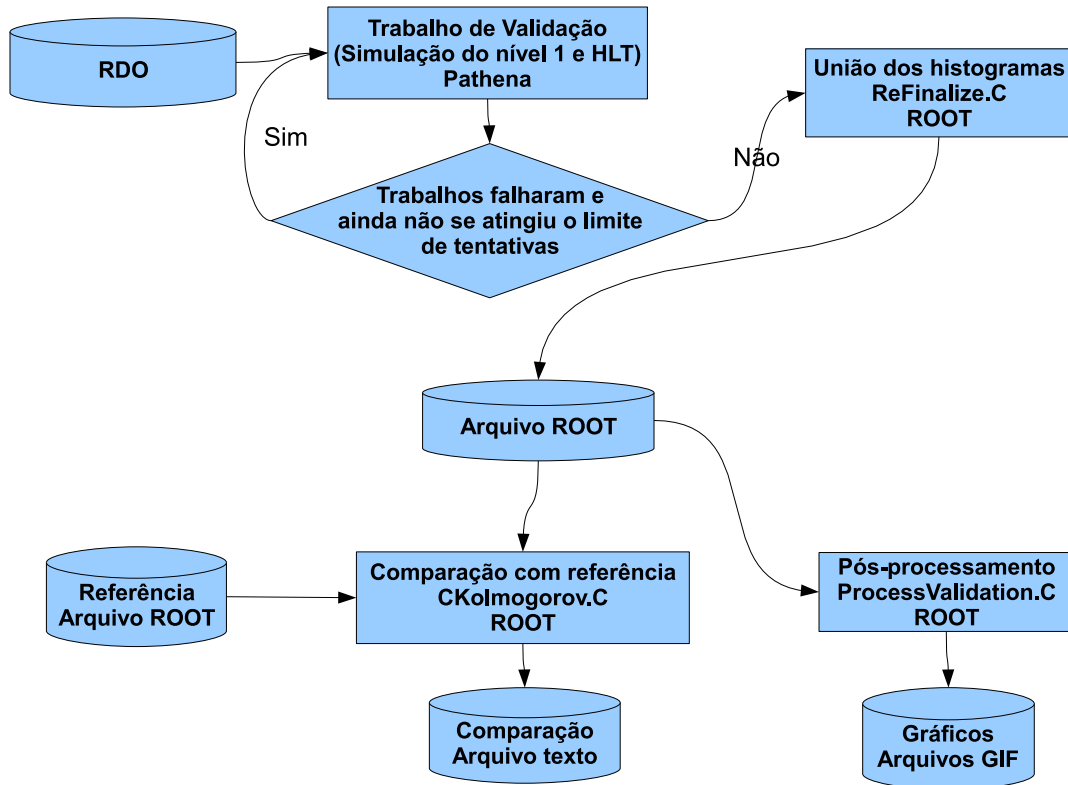


Figura 4.20: Envio de trabalhos e pós-processamento através do Psequencer para o TrigEgammaValidation.

similar à execução do trabalho na infraestrutura RTT. Há apenas um passo extra onde é verificado se todos os trabalhos foram finalizados corretamente e mais um processamento para unir todos os histogramas gerados utilizando o *script* **ReFinalize.C**.

A ferramenta Psequencer foi criada pelo desenvolvedor das ferramentas Panda devido a uma demanda para utilizá-la neste projeto, de forma a automatizar a execução do pós-processamento do procedimento de teste.

Apesar da utilidade de se executar estes procedimentos na rede Grid, as versões noturnas não estão disponíveis nos computadores conectados à rede. Portanto, este método é utilizado apenas para validar versões estáveis do *software* de filtragem do ATLAS.

Os resultados das comparações estão disponíveis em páginas Web semelhantes às disponíveis em versões noturnas, que são dedicadas à versões estáveis rolando na rede Grid. O local para acesso destes resultados é <http://www.cern.ch/>

`info-stabletrigegammavalidation/`. As Figuras 4.21 e 4.22 mostram resultados disponíveis nessa interface para elétrons puros simulados na versão estável 14.4.0. A estrutura da interface Web para o Grid é, essencialmente, a mesma que a interface Web para as versões noturnas. A única diferença na implementação é o tratamento da árvore de diretórios gerada após a geração do trabalho na rede Grid, que é mais simples que a árvore gerada pelo RTT.

A Figura 4.21 mostra os menus contendo itens gerais e itens relacionados aos procedimentos de teste, como no caso das versões noturnas. Da mesma forma, ao se clicar na descrição de um dos procedimentos de teste, um conjunto de sub-itens mostram os resultados para cada um dos níveis de filtragem. Do lado direito da figura, o primeiro histograma mostra o resultado da variável **Rcore**, utilizada no segundo nível de filtragem para determinar se a partícula é um elétron. Esta variável é calculada através de uma razão de energias das células da região de interesse, e espera-se que esteja bastante próxima do valor 1 para elétrons.

A Figura 4.22 mostra os gráficos de eficiência com a mesma forma que os gerados nas versões noturnas, após o Filtro de Eventos. Como no caso das versões noturnas, esperamos uma eficiência na coordenada ϕ uniforme e uma baixa eficiência na região da coordenada η em volta de $\|\eta\| = 1, 3$. A eficiência em energia deve ser aproximadamente uniforme após o corte de 22 GeV.

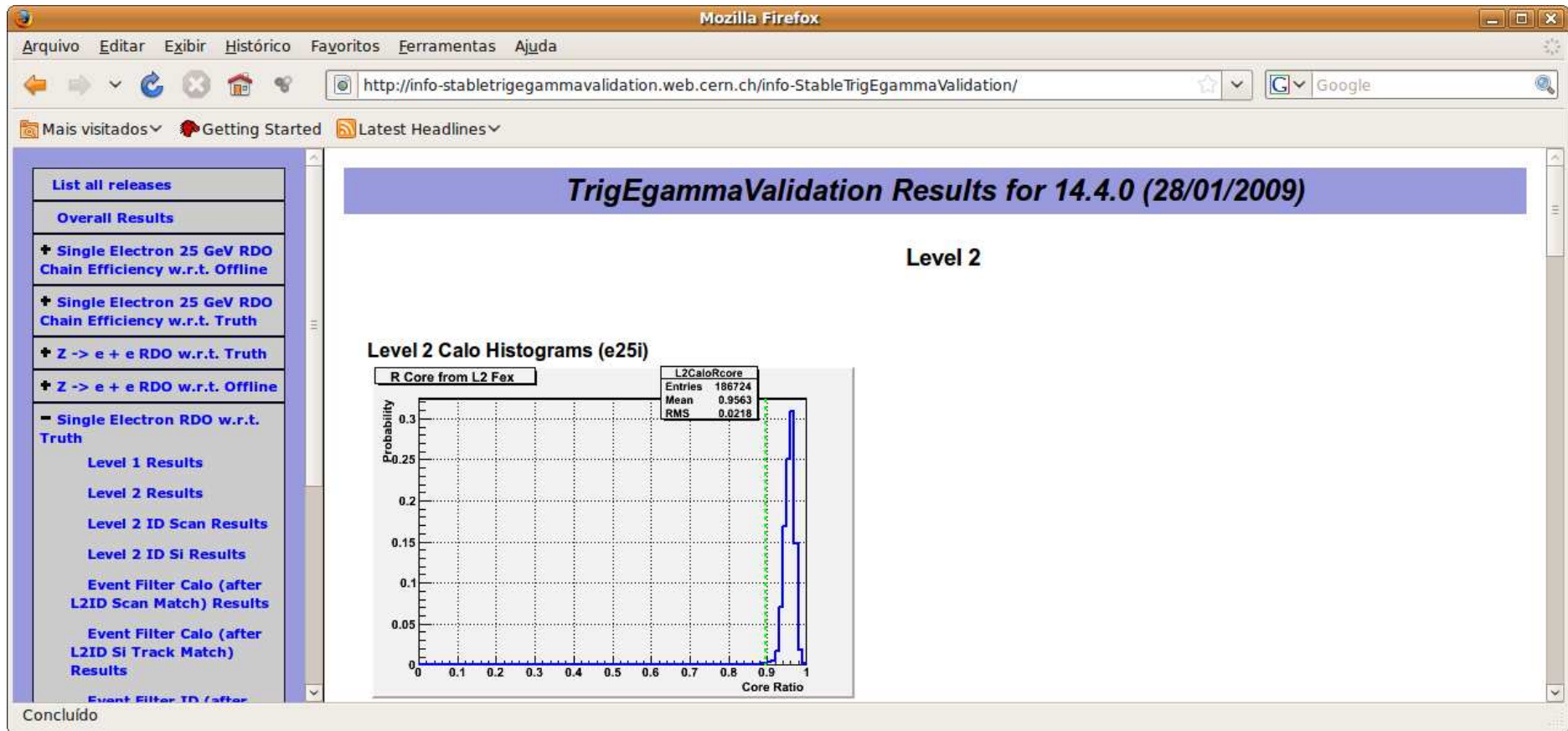


Figura 4.21: Interface Web mostrando os histogramas gerados no Grid para conjuntos de elétrons puros.

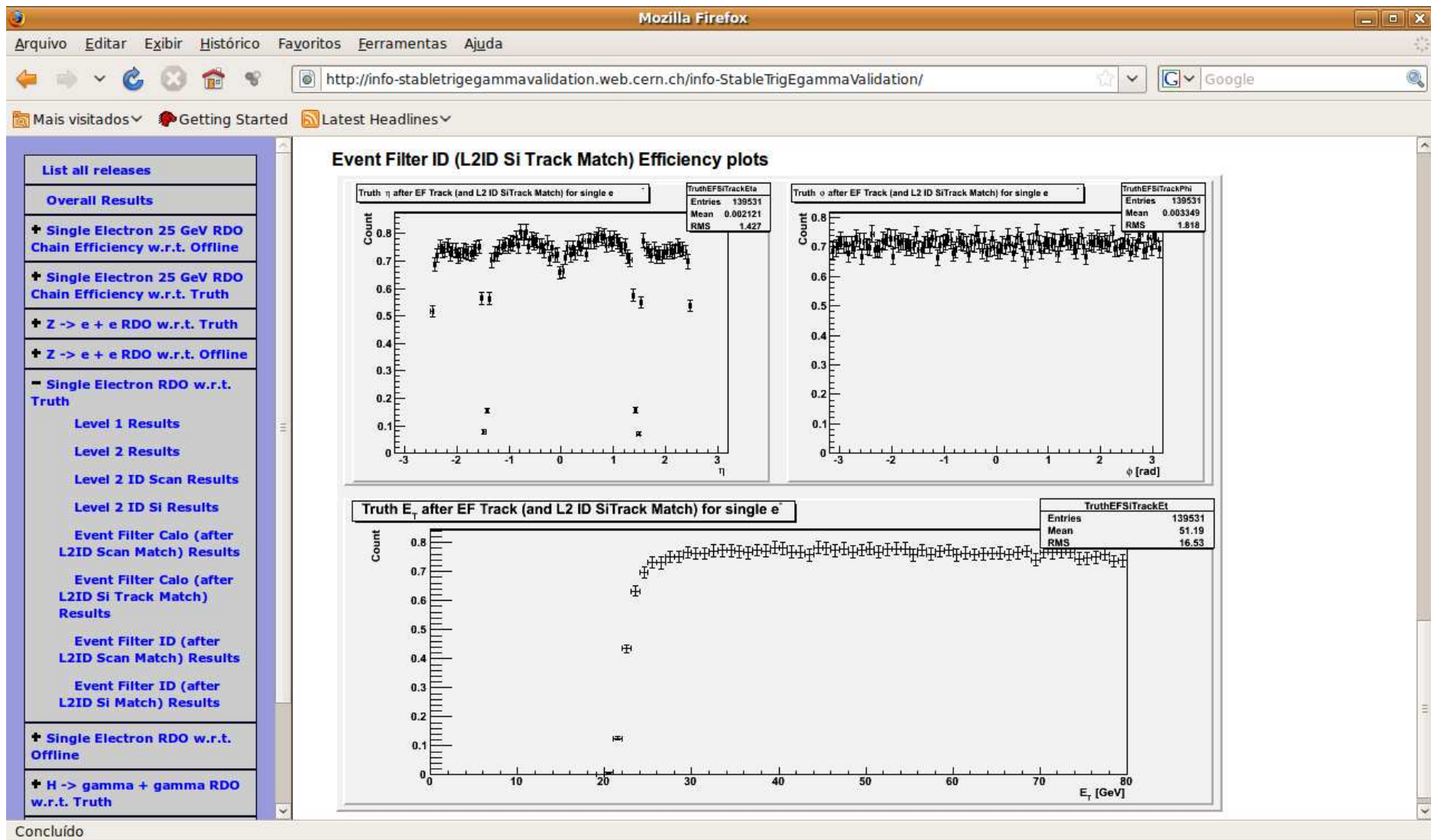


Figura 4.22: Interface Web mostrando o gráfico de eficiência gerado no Grid para conjuntos de elétrons puros.

Capítulo 5

Conclusões

Neste projeto, foi desenvolvido um sistema semiautomático de validação, que utiliza métricas estatísticas para detectar inconsistências no desempenho (eficiência e acurácia na extração de características) do *software* sob teste. No caso particular do sistema de filtragem do detector ATLAS do LHC, onde este projeto se insere, a validação depende significativamente de conhecimentos de física, que nos permitem detectar erros a partir de informações extraídas após o processamento de dados simulados.

O projeto utilizou a infraestrutura **Run Time Tester** (RTT), desenvolvida pela Colaboração ATLAS, que fornece uma interface de desenvolvimento mais simples para a validação. As análises desenvolvidas foram escritas na linguagem de programação C++ e utilizam a informação extraída pelo sistema de filtragem para realizar histogramas que podem ser úteis na detecção de falhas. Métricas estatísticas foram utilizadas para comparar os resultados entre diferentes versões do sistema de filtragem, visando a uma medida automática de falhas.

A sensibilidade das métricas utilizadas na validação foi estudada, mostrando que o teste de Kolmogorov-Smirnov é bem sensível a alterações nos histogramas sob validação. Também ficou claro que o comportamento da divergência de Jensen-Shannon é bastante similar ao da divergência simetrizada de Kullback-Leibler. Além disso, a Negentropia Quadrática pode não ter uma resposta contínua em certos casos (como foi o caso da multiplicação de uma variável gaussiana).

Os resultados mostram como as métricas podem ser utilizadas para detectar erros reais de forma eficiente. Eles sugerem que ao se observar as métricas se

pode apontar quais histogramas causaram a diferença rapidamente, permitindo um rastreamento de problemas mais eficiente. Os problemas analisados incluíram uma alteração no algoritmo de calibração utilizado para o cálculo de energia no nível 2, resposta assimétrica na coordenada ϕ e a falha na detecção de uma das partículas pelo sistema de filtragem. Em todos os casos, foi possível detectar as diferenças observando apenas as métricas.

Com o objetivo de facilitar a visualização dos resultados, uma interface Web que sumariza os resultados dos testes de validação realizados também está disponível. Esta interface permite verificar os resultados das comparações e os histogramas gerados de forma mais simples do que a interface RTT.

No caso de versões estáveis do *software* sob validação, também pode-se utilizar a rede Grid de processamento distribuído, na qual temos a vantagem de obter histogramas com muito mais dados e, portanto, uma validação mais precisa.

Muito do conhecimento e dos recursos desenvolvidos neste projeto foram utilizados para desenvolver também uma ferramenta de monitoração *online* para o sistema de filtragem no canal elétron/fóton. Neste caso, o propósito foi gerar alguns histogramas para verificar o funcionamento do sistema de filtragem em curto tempo, após a obtenção de dados reais. Este outro projeto foi implementado no pacote TrigHLTMonitoring e TrigOfflineMon, ambos disponíveis para a Colaboração ATLAS.

O Sistema de Filtragem opera em dois estágios. Este trabalho valida apenas o primeiro estágio, analisando características importantes na decisão do sistema. Trabalhos futuros estão sendo desenvolvidos com o objetivo de validar também o segundo estágio, onde a decisão é efetivamente aplicada. Também pretende-se realizar análises sistemáticas de decaimentos da partícula J/Ψ , o que será bastante útil para verificar a intercalibração dos subdetectores.

Apesar de este trabalho ter sido desenvolvido no âmbito do Sistema de Filtragem do detector ATLAS, a idéia pode ser facilmente estendida para outros ambientes, dentro ou fora do domínio da Física de Altas Energias.

Referências Bibliográficas

- [1] CERN, “CERN - European Organisation for Nuclear Research”, <http://www.cern.ch>. Acessado em 19 de dezembro de 2007.
- [2] CERN, “Página web do LHC”, <http://lhc.web.cern.ch/lhc>. Acessado em 19 de dezembro de 2007.
- [3] DOS ANJOS, A. R., **Sistema Online de Filtragem em um Ambiente com Alta Taxa de Eventos**. Tese (Doutorado), COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.
- [4] ATLAS Collaboration, AAD, G., et al., “The ATLAS Experiment at the CERN Large Hadron Collider”. Em: **ATLAS Detector Status and Physics Startup Plans**, 2008.
- [5] CERN, “ATLAS Collaboration”, <http://atlas.web.cern.ch/Atlas>. Acessado em 19 de dezembro de 2007.
- [6] FEYNMAN, R. P., LEIGHTON, R. B., SANDS, M., **The Feynman Lectures on Physics**. Addison Wesley, 1998.
- [7] GRIFFITHS, D. J., **Introduction to Elementary Particles**. John Wiley and Sons Inc., 1987.
- [8] The Particle Data Group, “The Particle Adventure”, <http://www.particleadventure.org>. Acessado em 19 de dezembro de 2007.
- [9] DA ROCHA GESUALDI MELLO, A., **Processamento Digital de Imagens para um Detector de Partículas de Alta Energia e Elevada Taxa de Eventos**. Tese (Doutorado), COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

- [10] ATLAS Collaboration, **Liquid Argon Calorimeter Technical Design Report**. CERN, 1996.
- [11] ATLAS Collaboration, **Tile Calorimeter Technical Design Report**. CERN, 1996.
- [12] ATLAS Collaboration, **ATLAS Detector and Physics Performance Technical Design Report**. CERN, 1999.
- [13] The ROOT developers, “Página web do Sistema ROOT”, <http://root.cern.ch/>. Acessado em 16 de janeiro de 2009.
- [14] ATLAS Collaboration, **ATLAS High-Level Trigger, Data Acquisition and Controls Technical Design Report**. CERN, 1999.
- [15] ATLAS Collaboration, “Data Preparation for the ATLAS High-Level Trigger Calorimeter Algorithms”, 2008. Nota interna publicada pela Colaboração ATLAS.
- [16] WIKIPEDIA, “Hadronization - Wikipedia, the free encyclopedia”, <http://en.wikipedia.org/wiki/Hadronization>. Acessado em 19 de dezembro de 2007.
- [17] CollabNet, Inc., “subversion.tigris.org”, <http://subversion.tigris.org/>. Acessado em 04 de abril de 2009.
- [18] Desenvolvedores da ferramenta CVS, “CVS - Open Source Version Control”, <http://www.nongnu.org/cvs/>. Acessado em 04 de abril de 2009.
- [19] ATLAS Software Infrastructure Team, “Tag Collector 2 developer page”, <http://atlastagcollector.in2p3.fr/>. Acessado em 04 de abril de 2009.
- [20] UNDRUS, A. E., “NICOS System of Nightly Builds for Distributed Development”. Em: **Proceedings of 2003 Conference for Computing in High-Energy and Nuclear Physics (CHEP 03)**, La Jolla, California, 2003.
- [21] SIMMONS, B., “The ATLAS RunTimeTester User Guide”, <http://www.hep.ucl.ac.uk/atlas/AtlasTesting/RTTUserGuide/RTTUserGuide.html>. Acessado em 04 de abril de 2009.

- [22] Free Software Foundation, Inc., “Welcome [Savannah]”, <http://savannah.gnu.org/>. Acessado em 04 de abril de 2009.
- [23] GNU, “Página web oficial do GDB”, <http://www.gnu.org/software/gdb/>. Acessado em 16 de janeiro de 2009.
- [24] Valgrind Developers, “Página web oficial da infraestrutura Valgrind”, <http://valgrind.org/>. Acessado em 16 de janeiro de 2009.
- [25] RTT, “Página web com documentação do sistema RTT”, <http://www.hep.ucl.ac.uk/atlas/AtlasTesting/RTTUserGuide/RTTUserGuide.html>. Acessado em 16 de janeiro de 2009.
- [26] CDF Collaboration, “First measurements of inclusive W and Z cross sections from Run II of the Fermilab Tevatron Collider”, **Physical Review Letters**, v. 94, 2005.
- [27] GRIFFITHS, D. J., **Introduction to Quantum Mechanics**. Prentice-Hall Inc., 1995.
- [28] JAMES, F., **Statistical Methods in Experimental Physics**. 2^a ed. World Scientific Publishing Co., 2006.
- [29] HYVÄRINEN, A., KARHUNEN, J., OJA, E., **Independent Component Analysis**. John Wiley & Sons Inc., 2001.
- [30] TUMMINELLO, M., LILLO, F., MANTEGNA, R. N., “Kullback-Leibler distance as a measure of the information filtered from multivariate data”, **Phys. Rev. E**, **031123**, v. 76, n. 3, pp. 12, Junho 2007.
- [31] JOHNSON, D. H., SINANOVIĆ, S., **Symmetrizing the Kullback-Leibler Distance**, Relatório, IEEE Transactions on Information Theory, 2001.
- [32] LIN, J., et al., “Divergence Measures Based on the Shannon Entropy”, **IEEE Transactions on Information Theory**, v. 37, n. 1, Janeiro 1995.
- [33] The PHP Group, “PHP: Hypertext Preprocessor”, <http://php.net/>. Acessado em 04 de abril de 2009.

- [34] UMIC - Agência para a Sociedade do conhecimento I.P., “Iniciativa Nacional GRID”, <http://www.gridcomputing.pt/>. Acessado em 09 de março de 2009.
- [35] MAENO, T., “PanDA: distributed production and distributed analysis system for ATLAS”, **Journal of Physics: Conference Series**, v. 119, n. 6, pp. 4, 2008.

Apêndice A

Produção Científica

- [1] ENOQUE FERREIRA DE LIMA, D., DE SEIXAS, J. M., OLIVEIRA DAMAZIO, D., et al., "Software Validation Infrastructure for the ATLAS High-Level Trigger", **XII Advanced Computing and Analysis Techniques in Physics Research**, PoS (ACAT08) 084, 2009.

- [2] ENOQUE FERREIRA DE LIMA, D., DE SEIXAS, J. M., OLIVEIRA DAMAZIO, D., "Electron and Photon Selection Validation Tool for the High-Level Trigger at ATLAS", Painel, **Workshop em Física Experimental de Altas Energias e Tecnologias Associadas: 20 anos de colaboração com o CERN**, Dezembro 2008.