

Dissertação de
Mestrado

Comissionamento e Análise do Feixe de Teste do Experimento MINER ν A

David Alejandro Martinez Caicedo

Centro Brasileiro de Pesquisas Físicas

Rio de Janeiro, Julho de 2010

*Dedicado a mi Madre,
a mi tía Kathy
y a la memoria de mi abuela Olga.*

Agradecimentos

Començando mil gracias a Deus.

Ao meu orientador, Dr. Hélio da Motta, pelo constante apoio, atenção, dedicação e amizade, pelos muitos conselhos e ensinamentos recebidos durante a mestrado e pela oportunidade que me deu para participar neste projeto do Fermilab.

A todos os pesquisadores do CBPF, em especial aos professores Gilvan Alves, José Helayel, Fernando Dantes Nobre.

A toda a colaboração do experimento MINER ν A, em especial ao Dr. Jorge Morfin, ao Dr David Schmitz pelas acertadas sugestões para a melhoria de minha análise, ao Dr. Bruno Gobbi pela sua grande sabedoria para resolver problemas, e ao Dr. Richard Gran por seus valiosos conselhos na hora de interpretar e analisar os resultados.

À CAPES pela bolsa concedida durante o Mestrado.

A todo o pessoal administrativo do CBPF, em especial à Bethe e Ricardo.

A minha família por todo o apoio que me deu em todos os momentos. Com todo o amor para minha mãe Olga, para minha segunda mãe Kathy, e também a minhas irmãs Mafe e Sara, e meu primo Jorge Daniel. A meu tio Villa, como sempre seus grandes conselhos e sabedoria, mais que um tio um amigo.

À minha segunda família e grandes amigos Ruth y Jose, obrigado por todos os seus conselhos e sua grande paciência para suportar-me e ajudar-me sempre para decidir o melhor. Muito Obrigado por tudo FISHYS e, claro, muito obrigado a Jose que teve uma grande paciência para ensinar-me muitas coisas de C++ e ROOT, que usei para poder desenvolver muito de minhas análises de dados.

A Arturo o grande amigo de viagens e, claro, de PARRANDAS!

A meu grande PARCERO EL MAESTRO Alex Tapia, sua esposa e filha, já muitos anos de amizade e sempre disponível para debater física e falar muita CARRETA e acho que seguiremos assim por muito mais tempo!!!

Ao Richar, Diana e VALE!! uma grande família e grandes amigos. Aos colegas e amigos do CBPF, Dr. Cesar Castromonte por seus valiosos conselhos e ajuda na minha estadia no Fermilab, Elvis, Marcos, Santiago. A meus amigos do Fermilab, Marcos Alania um grande PARCERO, Noemi a CHICA DA PUC e, em especial, a Lee Patrick por sua paciência para explicar muitos detalhes tanto de software como de hardware durante o desenvolvimento do projeto da linha de feixe e por sua grande qualidade humana.

A meus profesores da Universidad de Nariño, Colômbia, Juan Florez e Alberto Quijano.

A meus amigos da Edwin Insuasty, Alejandro Perez, William Erazo.

Agradecimientos

Comenzando doy mil gracias a Dios.

A mi orientador el Dr Helio da Motta, por el constante apoyo, atencion, dedicacion y amistad, por los muchos consejos y conocimientos recibidos durante la maestria y la oportunidad que me dio de participar en este proyecto en Fermilab.

A todos los investigadores del CBPF, en especial a los profesores Gilvan Alves, José Helayel, Fernando Dantes Nobre.

A toda la colaboración MINERVA , en especial al Dr. Jorge Morfin, al Dr. David Schmitz por las acertadas sugerencias para la mejora de mi análisis al Dr. Bruno Gobbi por sus valiosas charlas y su gran sabiduria a la hora de resolver problemas, y al Dr. Richard Gran por sus valiosos consejos a la hora de interpretar y analizar resultados.

A CAPES por la beca concedida durante la Maestría.

A todo el personal administrativo del CBPF, en especial a Bethe y Ricardo.

Quisiera agradecer a mi familia por todo el apoyo que me brindaron en todo momento. Con todo el amor a mi madre Olga, a mi segunda madre mi tia Kathy, tambien a mis hermanas Mafe y Sara y a mi primo Jorge Daniel. A mi tio Villa, como siempre sus grandes consejos y sabiduria, mas que un tio un gran amigo.

A mi segunda familia y grandes amigos Ruth y Jose, gracias por todos sus consejos y su gran paciencia para soportarme y ayudarme siempre para decidir lo mejor. Muchas gracias por todo FISHYS!, y claro muchas gracias a Jose, que tuvo una gran paciencia para enseñarme muchas cosas de C++ y ROOT, las cuales aplicaria para poder desarrollar muchos de mis analisis de datos. A Arturo el gran amigo de viajes, y claro de PARRANDAS!

A mi gran PARCERO EL MAESTRO Alex Tapia, su esposa e hija, ya muchos años de amistad y siempre ahí para debatir fisica y hablar bastante CARRETA! y creo que seguiremos así por mucho mas tiempo!!!.

A Richar, Diana y VALE!! una gran familia, y grandes amigos!! A mis colegas y amigos del CBPF, Dr. Cesar Castromonte por su valiosos consejos y ayuda en mi estancia en Fermilab, Elvis, Marcos, Santiago. A mis amigos de Fermilab, Marcos Alania un gran PARCERO, Noemi la CHICA DE LA PUC, y en especial a Lee Patrick por su paciencia para explicar muchos detalles tanto de software como de hardware durante el desarrollo del proyecto del test beam y por su gran calidad humana.

A mis profesores de la Universidad de Nario, Juan Florez, Alberto Quijano!!.

A mis amigos de la Universida Edwin Insuasty, Alejandro Perez, William Erazo y Alex Mera.

Resumo

O experimento MINER ν A dispõe de um detector de neutrinos altamente segmentado e de grande precisão e registrará uma alta estatística de eventos (mais de 13 milhões em 4 anos de coleta de dados). O MINER ν A utiliza a linha de neutrinos muônicos NuMI do FERMILAB. O detector permitirá o estudo de interações neutrino-nucleon em grande detalhe. Além disso, o detector do MINER ν A possui um alvo com diferentes materiais, permitindo, pela primeira vez, o estudo de efeitos nucleares em interações de neutrinos. Para teste e caracterização de seu detector, o experimento utiliza uma versão em menor escala de seu detector que é exposto a um feixe de hádrons especialmente projetado e construído para este fim. Apresentamos aqui os trabalhos realizados no experimento MINER ν A que resultaram na montagem, teste e análise desta linha de feixe.

Abstract

The MINER ν A experiment has a highly segmented and high precision neutrino detector able to record events with high statistic (over 13 millions in a four year run). MINER ν A uses FERMILAB NuMI beamline. The detector will allow for a detailed study of neutrino-nucleon interactions. Moreover, the detector has a target with different materials allowing, for the first time, for the study of nuclear effects in neutrino interactions. MINER ν A employs a small scale version of its detector for test and characterization of its detector. This test detector is exposed to a specially designed and built hadron beam. In this dissertation we present the work done at the MINER ν A experiment that resulted in the assembling, test and analysis of this beamline.

Colaboração MINERνA

I. Albayrak⁷, L. Aliaga¹⁶, M. Andrews⁵, S. Avvakumov¹⁷, B. Baldin⁵, J. L. Bazo¹⁶, D. B. Beringer²², G. Blazey¹³, A. Bodek¹⁷, D. Boehnlein⁵, S. Boyd¹⁵, R. Bradford¹⁷, W. K. Brooks¹⁰, A. Bruell¹⁰, D. Buchholtz¹⁴, H. Budd¹⁷, A. Butkevich⁸, D.A.M. Caicedo², D. Casper³, C. Castromonte², D. Cherdack²⁰, M. E. Christy⁷, J. Chvojka¹⁷, M. A. C. Cummings¹³, D. S. Damiani²², I. Danko¹⁵, M. Day¹⁷, D. Drakoulakos¹, S. Dytman¹⁵, R. Ent¹⁰, G. A. Fiorentini², J. Felix⁶, R. Flight¹⁷, A. Gago¹⁶, H. Gallagher²⁰, D. Gaskell¹⁰, R. Gilman¹⁸, C. Gingu⁵, C. Glasshausser¹⁸, R. Gran¹², N. Grossman⁵, B. Gobbi¹⁴, D. A. Harris⁵, J. Hobbs¹⁴, X. Jiang¹⁸, A. Jeffers¹⁴, T. Kafka²⁰, V. Kuznetsov¹⁴, C. E. Keppel⁷, J. Kilmer⁵, S. M. S. Kim¹⁵, S. Kopp¹⁹, M. A. Kordosky²², M. Kostin⁵, S. Kulagin⁸, G. Kumbartzki¹⁸, H. Lee¹⁷, A. G. Leister²², L. Loiacano¹⁹, A. Loveridge¹⁴, S. Manly¹⁷, E. Maher¹¹, W. A. Mann²⁰, K. McFarland¹⁷, A. McGowan¹⁷, D. Meekins¹⁰, W. Melnitchouk¹⁰, A. Mislevic¹⁷, G. Moreno⁶, J. G. Morfin⁵, H. da Motta², D. Naples¹⁵, J. K. Nelson²², G. Niculescu⁹, I. Niculescu⁹, R. Ochoa²¹, W. Oliver²⁰, J. Olsen⁵, J. L. Palomino², V. Paolone¹⁵, J. Park¹⁷, L. Patrick¹⁴, E. Paschos⁴, G. Perdue¹⁷, O. Pereyra²¹, A. Pla-Dalmau⁵, M. Proga¹⁹, R. Ransome¹⁸, M. Reyes⁶, P. Rubinov⁵, V. Rykalin¹³, C. Rude¹², W. Sakumoto¹⁷, P. Shanahan⁵, H. Schellman¹⁴, R. M. Schneider²², E. Schulte¹⁸, C. Simon³, J. Solano²¹, P. Spentzouris⁵, P. Stamoulis¹, J. Steinman¹⁷, J. Tatar³, V. Tvaskis⁷, G. Tzanakos¹, M. Vaz², S. Wood¹⁰, G. Zavala⁶, B. Ziemer³, M. Zois¹

University of Athens, Athens, Greece¹

Centro Brasileiro de Pesquisas Fisicas, Rio de Janeiro, Brazil²

University of California, Irvine, California³

University of Dortmund, Dortmund, Germany⁴

Fermi National Accelerator Laboratory, Batavia, Illinois⁵

Universidad de Guanajuato, Guanajuato, Mexico⁶

Hampton University, Hampton, Virginia⁷

Institute for Nuclear Research, Moscow, Russia⁸

James Madison University, Harrisonburg, Virginia⁹

Jefferson Lab, Newport News, Virginia¹⁰

Massachusetts College of Liberal Arts, North Adams, Massachusetts¹¹

University of Minnesota Duluth, Duluth, Minnesota¹²

Northern Illinois University, DeKalb, Illinois¹³

Northwestern University, Evanston, IL¹⁴

University of Pittsburgh, Pittsburgh, Pennsylvania¹⁵

Pontificia Universidad Catolica del Peru, Lima, Peru¹⁶

University of Rochester, Rochester, New York¹⁷

Rutgers University, New Brunswick, New Jersey¹⁸

University of Texas-Austin, Texas¹⁹

Tufts University, Medford, Massachusetts²⁰

Universidad Nacional de Ingenieria, Lima, Peru²¹

The College of William and Mary College, Williamsburg, Virginia²²

Conteúdo

Agradecimentos	ii
Agradecimientos	iii
Resumo	iv
Abstract	v
Colaboração MINER ν A	vi
Lista de Figuras	x
Lista de Tabelas	xv
1 Introdução	2
2 Física do Neutrino	3
2.1 História	3
2.2 Propriedades dos neutrinos	5
2.3 Experimentos realizados	5
2.3.1 Neutrinos atmosféricos	5
2.3.2 Neutrinos solares	7
2.3.3 Experimentos com aceleradores	7
2.3.4 Experimentos com reatores	8
2.4 O Modelo Padrão	8
2.5 Neutrinos no Modelo Padrão	11
2.5.1 Adicionando massas de neutrinos ao Modelo Padrão	13
2.5.2 Adicionando a mistura de neutrinos ao Modelo Padrão	13
3 O experimento MINERνA	17
3.1 Feixes de neutrinos em aceleradores	17

3.2	O feixe NuMI	19
3.3	Descrição geral	21
3.4	Detectores de partículas	27
3.4.1	Perda de energia de partículas carregadas	27
3.4.2	Interação de fótons	29
3.5	Os dispositivos fotossensores	30
3.6	Os elementos cintiladores	31
3.7	Calorímetros	33
3.7.1	Calorímetro eletromagnético	33
3.7.2	Calorímetro hadrônico	34
3.8	Alvos nucleares	34
3.9	Eletrônica e aquisição de dados (DAQ)	36
3.9.1	Hardware do DAQ	36
3.9.2	<i>Software</i> de Aquisição de dados (DAQ)	37
3.10	Parâmetros do detector do MINER ν A	38
3.11	Testes	39
3.11.1	Mapeador	39
3.11.2	Teste das fotomultiplicadoras	40
3.11.3	Detector de Teste	40
3.11.4	Protótipo de rastreamento	41
3.12	Fase de Testes	41
3.13	O detector completo	42
3.14	Perspectivas do experimento MINER ν A	42
3.14.1	Espalhamento quase-elástico	46
3.14.2	Produção ressonante de píons	47
3.14.3	Produção coerente de píons	48
3.14.4	Dependência em A da seção de choque coerente	48
3.14.5	Produção de partículas estranhas e charmosas	49
3.15	Tomada de Dados	50
4	O feixe de Teste do MINERνA	55
4.1	Introdução	55
4.2	O feixe de mésons MB - Feixe Secundario	56
4.3	O Feixe Terciário	57
4.3.1	O Alvo de Cobre	59
4.3.2	O Colimador de Aço	59

4.3.3	Os Magnetos	60
4.3.4	Câmaras de fios	63
4.3.5	Hodoscópios	65
5	Análise do Feixe de Teste	67
5.1	Geração do Gatilho e Medida de Tempo de voo	67
5.2	Identificação das Partículas do Feixe	68
5.3	Configuração do Feixe	68
5.4	Resposta das Câmaras de Fios	70
5.5	Distribuição dos Momenta	71
5.6	Calibração pelo Cálculo de Massa do Píon	75
5.7	Eficiência das Câmaras de Fios	77
5.8	Resolução do Tempo de Voo	78
5.9	Sinal Duplo no Sistema de Tempo de Voo	79
5.10	Teste Geral do Sistema	79
5.11	Simulação do Detector de Teste do MINER ν A	81
6	Conclusões	84
A	Teste de MAPMT	86
A.1	Teste dos Fotomultiplicadores	86
A.1.1	Caixa dos PMT	86
A.1.2	Descrição das Montagens de Testes	87
A.1.3	Testes de Diafonia dos MAPMT	88
B	Códigos Fontes Desenvolvidos	91
B.1	Algoritmo para identificar canais mortos	91
B.1.1	deadchannels.C	91
B.2	O algoritmo basic stats (eficiência das câmaras)	93
B.2.1	basicstats.cxx	93
B.3	O algoritmo basic histos (ocupação das câmaras)	97
B.3.1	basichistos.cxx	97
B.4	O algoritmo wchits (visualização de ocupação das câmaras)	102
B.4.1	wchits.C	102
B.5	O algoritmo adaptado para o Event Display da simulação	105
B.5.1	EventDisplay.h	105
B.5.2	EventDisplay.C	111
B.6	Algoritmo do campo magnético	126

B.6.1	fitquadratic.cpp	126
B.7	O algoritmo usado para encontrar o valor médio do ruído de fundo do campo magnetico	128
B.7.1	zerosaverage.py	128
B.8	O algoritmo para fazer análise preliminar dos dados do campo magnético	130
B.8.1	pythonanalysis.py	130
B.9	Um dos algoritmos para reconstrução das trajetórias	132
B.9.1	TFullTrack.cxx	132
B.10	Algoritmo para visualização 3D das trajetórias	147
B.10.1	TBeamline.cxx	147

Bibliografia **149**

Lista de Figuras

2.1	Diagramas de Feynman representando contribuições de corrente neutra (direita) e corrente carregada (esquerda) ao processo de espalhamento $\nu_l + l \rightarrow \nu_l + l$, onde l representa qualquer uma das três famílias de léptons.	10
2.2	Regras de Feynman para vértices de interações electrofracas em termos de spinores de Dirac e de suas projeções ψ_L e ψ_R	12
3.1	Fração de π^+ escapando de um alvo espesso de grafite (partículas terciárias) como função do momentum do próton incidente, p_0 . O gráfico da esquerda mostra a fração de terciários para alvo com 2,0 comprimentos de interação para π^+ com momentum $p_z > 0,5 \text{ GeV}/c$; $5,0 \text{ GeV}/c$, etc. O gráfico da direita mostra a fração total de terciários para alvos de diferentes comprimento de interação, $0,5 \lambda$, $1,0 \lambda$ e $2,0 \lambda$. Figura extraída de [42]	18
3.2	Energia do próton incidente versus massa atômica do alvo nuclear para os cinco feixes de neutrinos em aceleradores existentes no mundo	19
3.3	Principais componentes da linha de feixe NuMI. Mésons positivos, produzidos no alvo, são focalizados por dois cones magnéticos e decaem em uma região de 675 m mantida em vácuo, produzindo, predominantemente, ν_μ . Os hádrons restantes são bloqueados em um absorvedor de feixe colocado ao final da região de decaimento.	20
3.4	Diferentes configurações do feixe NuMI. O espectro do feixe NuMI pode ser alterado movendo-se o alvo e/ou a segunda corneta focalizadora.	21
3.5	Vista do MINER ν A montado em sua estrutura de suporte.	22
3.6	Vista de cima do MINER ν A. Os MAPMT destacam-se no topo do detector.	22
3.7	Vista esquemática lateral do detector do MINER ν A	24
3.8	Vista de um módulo ativo do detector. A figura à direita mostra os três planos de fibras cintiladoras U,V e X deslocados de ± 60 graus.	25
3.9	Vista do detector do MINER ν A ilustrando forma e tamanho. Podemos ver os Alvos de Ferro e Chumbo	25

3.10	Vista de um módulo ativo do detector. A figura à direita mostra a estrutura deste tipo de módulo, UX ou VX.	26
3.11	Vista de um módulo do calorímetro eletromagnético. A figura à direita mostra a estrutura do calorímetro, alternando capas de chumbo com capas cintiladoras.	26
3.12	Vista de um módulo do calorímetro hadrônico. A figura à direita mostra a estrutura do calorímetro, alternando capas de ferro com capas cintiladoras.	27
3.13	Poder de parada para μ atravessando cobre em função do momentum. Extraído de [33]	28
3.14	Fração de energia perdida pelos elétrons e pósitrons em função da energia. Extraído de [47].	30
3.15	Seções de choque para canais de interação de fótons com a matéria. Extraído de [47]	31
3.16	Corte transversal de um prisma triangular cintilador empregado no <i>Inner Detector</i> . Nota-se o furo por onde a fibra óptica é inserida.	32
3.17	Prismas triangulares cintiladores arranjados de forma a constituírem um plano cintilador. Cada prisma corresponde a um canal. Cada prisma tem uma fibra óptica inserida ao longo de seu comprimento.	32
3.18	Vista de corte de um plano montado com prismas triangulares cintiladores.	32
3.19	Alvos nucleares: Vermelho = Fe, Cinza = Pb, Preto = C	35
3.20	Alvo de água do experimento MINER ν A	35
3.21	Diagrama esquemático do sistema de aquisição de dados do MINER ν A	38
3.22	Mapeador.	39
3.23	Montagem de teste das MAPMT.	40
3.24	O detector de testes do MINER ν A.	41
3.25	Esquema do detector do MINER ν A mostrando a parte que corresponde ao protótipo de rastreamento	42
3.26	Eventos com topologias diferentes podem ser distinguidos graças à capacidade de reconstrução do detector do MINER ν A. Resultados obtidos com simulação.	45
3.27	Medidas atuais de seção de choque para vários processos. Mostramos a região de energia acessível a diversos experimentos atuais e futuros. Observa-se a faixa de energia de atuação do MINER ν A. Gráfico extraído de [45]	46
3.28	superior: medidas atuais da seção de choque quase-elástica. A linha pontilhada em azul emprega um modelo de gás de Fermi. inferior: Previsão dos resultados de medida da seção de choque quase-elástica esperada pelo MINER ν A (apenas erros estatísticos).	47

3.29	Comparação entre os resultados existentes para produção coerente de píons e os resultados esperados pelo experimento MINER ν A.	48
3.30	Seção de choque coerente como função de A. A região sombreada indica a faixa de medidas anteriores. As cruces são as predições do modelo de Rein-Sehgal para espalhamento por C, Fe e Pb e os círculos são as previsões do modelo de Paschos-Kartavtsev. O experimento MINER ν A pode resolver as diferenças entre os diversos modelos.	49
3.31	Dois eventos obtidos com 25 módulos do detector MINER ν A (20% do detector) durante tomada de dados de teste no feixe NuMI.	51
3.32	Candidato de múons. A trajetória observada atravessa todo o detector e origina-se de fora do detector (vindo da esquerda). É um sinal compatível com o que seria deixado no detector por um múon viajando na mesma direção do feixe. Evento observado com um detector com 64 módulos.	52
3.33	Um evento onde uma das partículas resultantes viaja na direção contrária ao feixe. O neutrino penetra no detector vindo pela esquerda e interage no centro do detector (o ponto de onde vemos brotar várias trajetórias.	52
3.34	Evento obtido na tomada de dados com o detector completo do MINER ν A candidato a $\nu_\mu + n \rightarrow p\mu^- \pi^0$	53
3.35	Candidato π_0	54
3.36	Chuveiro electromagnético	54
4.1	Píons e prótons esperados. Simulação de MC usando NueGen.	55
4.2	Área de teste de múons no Fermilab mostrando as áreas experimentais utilizadas para o feixe de teste do MINER ν A	57
4.3	Área de teste de múons no Fermilab mostrando os componentes da linha de feixe para o detector de teste MINER ν A	58
4.4	Vistas da linha de feixe de teste.	59
4.5	Taxa esperada de partículas na saída do colimador para 15 milhões de π no alvo de Cu. Valores obtidos a partir de simulação.	60
4.6	Um dos magnetos empregados na linha de feixe. Observa-se o sistema empregado para mapeamento do campo magnético consistindo de uma grade de posições, sensor de prova e espaçador.	61
4.7	Monitoramento da temperatura dos magnetos da linha do feixe. À esquerda resultado obtido com simulação e, à direita, resultado obtido pelo sistema ACNET	62
4.8	Campo magnético B_y em um dos magnetos. À esquerda vemos o resultado de simulações e à direita o campo medido no processo de mapeamento. As polaridades para simulação e medidas são opostas.	63

4.9	Esquerda:Diagrama esquemático do sistema de leitura das câmaras de fios; Direita: eletrônica de leitura das câmaras.	64
4.10	Teste com SR-90 do plano XP da câmara 1. As figuras mostram histogramas da atividade nos canais para duas posições diferentes da fonte de raios β . Os detalhes na parte de baixo mostram os canais 51, 53 e 59 mortos.	65
4.11	A fonte de Sr-90 está posicionada proximo ao centro do plano. Observa-se, entretanto, atividade entre os fios 305 e 320, indicando canais quentes.	66
4.12	Distribuição XY, em milímetros, da associação de planos U e V da câmara 1. O gráfico da esquerda considera o fio 230 como central enquanto no gráfico da direita o fio 200 é tomado como central.	66
5.1	Diagrama lógico do gatilho	67
5.2	Distribuições de tempo de voo para quatro diferentes configurações de feixe. . .	69
5.3	Ocupação de canais das câmaras de fio. O eixo vertical corresponde a cada um dos 320 canais de cada câmara e o eixo horizontal divide-se entre os planos X,U,V e XP	70
5.4	Distribuição espacial em corte transversal do feixe como registrado pelas câmaras de fio	71
5.5	Rootupla gerada e histograma mostrando o número de canais ativados por evento para a câmara 1.	72
5.6	Reconstrução de trajetórias. Visão 2D e 3D de eventos com hits e sem hits nas camaras de fios.	73
5.7	Ângulo de deflexão e momentum calculados a partir do momentum transversal .	74
5.8	Distância percorrida e momentum determinados a partir do melhor ajuste da trajetória.	74
5.9	Tempo de voo (TOF) em função do momentum. Esquerda: simulação ; Direita: Dados.	75
5.10	Esquerda: cortes fiduciais nas câmaras de fios. Direita: distribuição de momentum versus TOF depois de aplicado cortes fiduciais	76
5.11	Distribuição da massa do pión para diferentes valores de distância percorrida pelos pions.	77
5.12	Verificação da resolução do tempo de voo. O gráfico da esquerda mostra as medidas do tempo de voo no início do processo e a figura da direita apresenta o resultado após os trabalhos de otimização realizados. A variável Sigma indica a resolução. O eixo horizontal apresenta apenas a diferença das medidas de tempo entre os conjuntos de hodoscópios com origem da contagem de tempo arbitrária.	79
5.13	Diagrama das conexões eletrônicas feitas para a linha de feixe.	80

5.14	Conexões eletrônicas feitas para a linha de feixe de teste	81
5.15	Ocupação das câmaras com todo o sistema montado. Observa-se vários canais com ocupação zero.	82
5.16	Ocupação das câmaras após operações de ajustes na eletrônica da linha de feixe	82
5.17	Rootupla gerada com a nova versão da simulação implementada para detector de teste do MINER ν A	83
5.18	<i>Event Display</i> mostrando um pión de 600 MeV/c gerado com simulação do detector de teste do MINER ν A com geometria de 10 módulos de rastreamento, 20 módulos de calorímetro eletromagnético e 10 módulos de calorímetro hadrônico.	83
A.1	Esquerda: Vista da fotomultiplicadora montada na caixa. Nota-se as fibras ópticas (à esquerda) e os cabos de sinal (à direita). Direita: Convenção de numeração dos elementos	87
A.2	Esquerda: Estrutura do sistema de testes usado. Direita: Foto da placa FEB. . .	87
A.3	Face de um MAPMT de 64 elementos. Os elementos são numerados de 1 a 64 começando no canto superior esquerdo. Os elementos 10, 14, 28, 51 e 55 são marcados como iluminados.	89
A.4	Distribuições de pedestais	89
A.5	Distribuições de contagem em teste de diafonia. Apenas os elementos 10, 14, 28, 51 e 55 estão iluminados. O elemento 37 apresenta sinal indicando diafonia entre ele e o seu vizinho 28.	90

Lista de Tabelas

2.1	Partículas que compõem o modelo padrão de interações fundamentais da natureza [33]	9
3.1	Quantidade de eventos de corrente carregada nos alvos nucleares	35
3.2	Algumas das exigências e parâmetros para o projeto da eletrônica do MINER ν A	36
3.3	Parâmetros do detector do MINER ν A.	39
3.4	Medidas atuais de corrente neutra.	43
3.5	Número de eventos em diferentes tópicos (valores aproximados)	44
3.6	Incertezas nas medidas das seções de choque atuais e as esperadas após o experimento MINER ν A.	47
3.7	Amostras esperadas pelo experimento MINER ν A para diversos eventos de produção exclusiva de partículas estranha durante uma tomada de dados de 4 anos	50
4.1	Características das câmaras de fios. Cada plano tem 320 canais efetivos.	63
5.1	Tempo de voo das partículas do feixe para momentum de 0,5 GeV/c e distância de 6 m entre os hodoscópios	68
5.2	Calibração da linha de feixe empregando a determinação da massa do pión	77

Capítulo 1

Introdução

A física de neutrinos é um dos tópicos mais interessantes em física de partículas. Atualmente há evidências de que os neutrinos possuem massa não nula [1] e podem mudar de sabor [2]. A mudança de tipo de neutrino (conhecida como oscilação de neutrinos) foi observada em experimentos com neutrinos solares, neutrinos atmosféricos e em experimento com aceleradores.

Esta tese é sobre meu trabalho realizado no experimento MINER ν A, que é um experimento montado na linha de feixe NuMI do FERMILAB empregando um feixe intenso de neutrinos. Em particular atuamos no grupo de responsável pela montagem e teste de uma linha de feixe empregada para testes e calibração. Realizamos trabalhos em *software* e *hardware* com vistas a instalação e teste da linha de feixe especial empregada para testes do detector.

A tese é dividida da seguinte forma:

O Capítulo 2 introduz aspectos básicos da física de neutrinos. Descreve a história do neutrino desde sua concepção teórica até os dias atuais, apresentando, de forma sucinta, uma série de experimentos realizados.

O Capítulo 3 descreve a linha de feixe de neutrinos NuMI, a física por trás dos detectores de partículas e o experimento MINER ν A.

O Capítulo 4 descreve o feixe de teste, seus componentes e origem. Apresentamos os testes realizados com cada elemento.

O Capítulo 5 descreve os testes e análises realizados com o feixe de teste e a verificação de seu funcionamento.

O Capítulo 6 faz um sumário e apresenta as conclusões desta dissertação.

Acrescentamos dois apêndices: o apêndice A descreve o trabalho de teste das fotomultiplicadoras multi-anôdo do experimento; o apêndice B apresenta os principais códigos desenvolvidos neste trabalho.

Capítulo 2

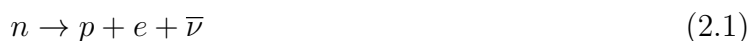
Física do Neutrino

2.1 História

Em 1911, um experimento realizado por von Bayer, Otto Hahn e Lise Meitner [3] sugeriu que a energia do elétron emitido no decaimento β apresentava um espectro contínuo ao invés de discreto. Isto estava em contradição com a lei de conservação da energia já que, aparentemente, energia era perdida durante o processo de decaimento.

Em 1927, Ellis e Wooster [4] estabeleceram, sem sombra de dúvidas, que o espectro do decaimento β é, de fato, contínuo.

Em decaimentos radioativos, os núcleos dos átomos se transformam em núcleos diferentes quando o nêutron decai em um próton com emissão de elétrons:



Sem o antineutrino, a conservação de energia requer que o elétron e o próton tenham, em conjunto, a energia do nêutron. Cada elétron deve, por isso, ser produzido com energia fixa, enquanto os experimentos mostraram que os elétrons não são mono-energéticos, mas apresentam-se em uma faixa contínua de energia. Esta faixa de energia corresponde exatamente ao estado final de três partículas que podem compartilhar energia, satisfazendo as leis de conservação.

Numa carta para a conferência de físicos em Tübingen (Alemanha) em 1930 [5] Wolfgang Pauli dirigiu-se aos *Liebe Radioaktive Damen und Herren*¹. Ele propõe um férmion que chamou de nêutron com massa da ordem de grandeza da massa do elétron e com carga elétrica zero que também seria emitido no decaimento β .

¹Prezados senhores e senhoras radioativos.

Quando Chadwick descobriu o nêutron, como o conhecemos hoje [6], em 1932, Fermi chamou de neutrino a partícula de Pauli, para diferenciá-la do nêutron pesado de Chadwick. A partir da comparação das formas observadas dos espectros, Fermi [7] e Perrin [8] postularam, em 1933, cada um, o neutrino sem massa.

Em 1934, Fermi [9] desenvolveu formalmente a teoria do decaimento β no âmbito da eletrodinâmica quântica de Dirac, Heisenberg e Pauli. Somente em 1956 Reines e Cowan [10] observaram diretamente o neutrino. Eles observaram a reação



empregando um reator como fonte de anti-neutrinos de 1 MeV e um alvo de cloreto de cádmio e água. O detector tinha um grande volume e o reator fornecia um fluxo intenso de neutrinos, uma necessidade devida à baixíssima secção de choque deste processo.

Em 1958, Goldhaber determinou a helicidade do neutrino [11] verificando serem eles sempre levógiros. Em 1959 Davis demonstrou que ν podiam ser distinguidos de $\bar{\nu}$ [12]. Em 1960 um experimento conduzido por Leon Lederman et. al. no acelerador AGS de Brookhaven [13] detectou um segundo tipo de neutrino: o ν_μ .

Em 1973, um experimento na câmara de bolhas Gargamelle do CERN descobriu interações de neutrinos na corrente neutra [14].

Experimentos com neutrinos solares tiveram início em 1968 por Davis [15] e foi observada uma discrepância entre as previsões teóricas e o fluxo de neutrinos solares observado. Esta discrepância ficou conhecida como o problema do neutrino solar.

Experimentos realizados com neutrinos atmosféricos também apresentaram uma discrepância entre o fluxo de neutrinos observados e o fluxo esperado. Os experimentos observaram no fluxo de neutrinos atmosféricos, que os neutrinos muônicos aparentemente desaparecem em comprimentos de propagação da ordem de centenas de quilômetros. Experimentos que mediram o fluxo de neutrinos solares acharam resultados que sugeriam o desaparecimento de neutrinos eletrônicos, na propagação dentro do Sol ou entre o Sol e a Terra.

Este resultado levou Gribov e Pontecorvo [16], em 1968, a descreverem a oscilação de sabores caso os neutrinos tenham massa. O “desaparecimento” de neutrinos atmosféricos (ν_μ) e neutrinos solares (ν_e) é fácil de explicar em termos de oscilação de neutrinos, a menos que pelo menos dois neutrinos tenham massa e se misturem. Isto não foi previsto pelo modelo padrão.

Em 1989, apenas alguns meses após o início da tomada de dados com o novo acelerador LEP do CERN, estudos da vida média do bóson Z revelaram a existência de três famílias de neutrinos [17, 18, 19, 20]. O terceiro neutrino, entretanto, só foi observado diretamente em 2000

pelo experimento DONUT [21].

Foi uma longa jornada desde os tempos pioneiros até o presente quando obteve-se a prova de que os neutrinos têm massa. Um amplo caminho no qual teóricos e experimentais caminharam de mãos dadas para mostrar a primeira evidência da física além do modelo padrão.

2.2 Propriedades dos neutrinos

Várias propriedades dos neutrinos já foram medidas, por exemplo:

- O experimento clássico de Wu et al. [22] em 1957 determinou que as interações fracas violam a conservação de paridade. Este resultado, aplicado a neutrinos sem massa leva a conclusão de que neutrinos devem ser polarizados, com helicidade +1 ou -1. Entretanto, em 1958, Goldhaber et al. [11] mediram a helicidade do neutrino e observaram que apenas neutrinos levógiros participam da interação fraca;
- Há apenas três tipos de neutrinos com acoplamento padrão ao bóson Z e com massas menores que $45 \text{ GeV}/c^2$ [17, 18, 19, 20];
- Neutrinos satisfazem à conservação de número leptônico, o que significa que um neutrino está sempre associado a seu lépton carregado.
- As massas dos neutrinos não são conhecidas. Embora tenham sido tradicionalmente consideradas como zero, experimentos recentes indicam serem elas muito pequenas, porém não nulas. A observação de oscilações de neutrinos é um indicativo de sua massa não ser nula.

2.3 Experimentos realizados

Apresentamos nesta seção um resumo de vários experimentos realizados com neutrinos. Esta lista não se pretende completa, mas descreve experimentos importantes e cujos resultados foram fundamentais para a compreensão que hoje temos dos neutrinos.

2.3.1 Neutrinos atmosféricos

Neutrinos atmosféricos são produzidos quando partículas de raios cósmicos interagem na atmosfera superior, produzindo um chuva secundária de mésons cujos decaimentos produzem neutrinos. Vários experimentos fornecem evidências para oscilações de neutrinos atmosféricos.

Super-Kamiokande

Consiste de um grande detector Čerenkov de água localizado na mina Mozumi no Japão. O detector está situado a uma profundidade de 1.000 m correspondendo a 2.700 metros equivalentes de água² (mea). Sua massa total é de 50 kt. Consiste de um tanque cilíndrico cheio com 680 t de água. O experimento detecta e mede neutrinos a partir do deslocamento do elétron na reação de espalhamento elástico

$$\nu_x + e^- \rightarrow \nu_x + e^- \quad (2.3)$$

onde $x = e, \mu, \tau$. Embora sensível aos três sabores de neutrinos, a reação de espalhamento elástico é dominada pelo fluxo de ν_e . O experimento Super-Kamiokande [23] encontrou evidências de oscilações de neutrinos ao registrar um déficit de $\nu_\mu(\bar{\nu}_\mu)$ enquanto observava o fluxo esperado de $\nu_e(\bar{\nu}_e)$.

Soudan2

Trata-se de um detector consistindo de um calorímetro com capacidade de rastreamento que foi instalado a 2.090 mea em uma mina em Soudan, Minesota, EUA. Possuía massa total de 963 t e foi construído para buscar decaimento de núcleons.

O experimento Soudan2 [24] observou o mesmo efeito de oscilação de neutrinos, de forma consistente com os resultados do experimento Super-Kamiokande. Note-se que os dois experimentos são diferentes, empregando materiais diferentes, tecnologias de detecção diferentes e técnicas diferentes de análise, o que reduz drasticamente a probabilidade de que erros sistemáticos possam falsear o sinal de oscilação de neutrinos.

SNO

O Observatório de Neutrinos de Sudbury (SNO) é um detector Čerenkov de 1 kt de água pesada (D₂O) ultra pura localizado a uma profundidade equivalente a 6.010 mea na mina Sudbury, no Canadá. Sudbury mediu o fluxo de neutrinos solares [25].

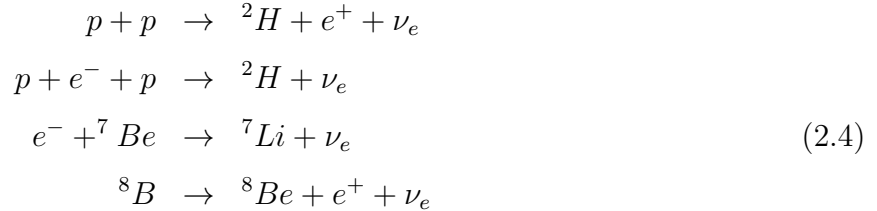
MACRO

Tratava-se de um experimento localizado no Laboratório de Gran Sasso, Itália. O detector estava colocado a 3.700 mea. MACRO tinha como objetivo buscar monopólos e múons vindos de fontes cósmicas. Para mais informação ver [31]

²Isto significa que a rocha envolvendo o detector equivale a colocar o detector sob 2.700 metros de água. É comum empregarmos esta unidade de medida equivalente de profundidade em física realizada em espaços subterrâneos.

2.3.2 Neutrinos solares

As reações de fusão nuclear no Sol produzem um fluxo intenso de neutrinos³. Essas reações podem ser representadas pelos processos:



O primeiro sinal persistente de oscilações de neutrinos foi observado em neutrinos solares. A detecção de neutrinos solares foi discutida, do ponto de vista experimental, por Davis [12]. Um número diferente de detectores persistentemente observava um número de neutrinos menor do que o esperado. Embora oscilação de neutrinos seja uma explicação possível para esta carência, não é a única. O número menor de neutrinos podia ser devido, por exemplo, a um modelo extremamente simples do Sol. Mohapatra [26] apresenta uma revisão histórica da situação experimental no final da década de 1990. O experimento SNO [25] observou uma taxa não nula de neutrinos não eletrônicos (ν_μ, ν_τ) oriundos do Sol. Como apenas ν_e são produzidos no Sol, isto constitui forte evidência para as oscilações de neutrinos. Evidências adicionais vieram do experimento KamLAND [27], no Japão, que emprega anti-neutrinos oriundos de um reator nuclear.

2.3.3 Experimentos com aceleradores

Experimentos deste tipo empregam um feixe de neutrinos produzidos em aceleradores. Neste arranjo o feixe original, em geral de p , colide com um alvo. Mésons resultantes da colisão são selecionados por intermédio de cones magnéticos e enviados em direção ao detector de neutrinos, posicionado a uma certa distância. O decaimento destes mésons produz um feixe de neutrinos.

O experimento K2K [28] emprega um feixe de ν_μ com energia média de 1,3 GeV e observou 28 eventos quando se esperava cerca de 38 eventos no caso de não ocorrerem oscilações.

O experimento MINOS [29] é um experimento de oscilações de neutrinos empregando o feixe de neutrinos NuMI do FERMILAB. Emprega dois detectores funcionalmente idênticos localizados a 1 km (o detector próximo) e a 750 km (o detector distante) da região onde os neutrinos são produzidos. O detector próximo mede a composição e o espectro do feixe de neutrinos enquanto o detector distante busca evidências de oscilações de neutrinos. MINOS observou um total de 215 eventos com neutrinos abaixo de 30 GeV comparado com $336,0 \pm$

³A energia média dos neutrinos que emergem do Sol é da ordem de 0,6 eV e dos atmosféricos e de 1 GeV.

$18,3 \pm 14,4$ eventos que seriam esperados caso não ocorressem oscilações. Para mais resultados recentes ver [30]

2.3.4 Experimentos com reatores

Vários experimentos foram realizados empregando reatores como fonte de anti-neutrinos. Um experimento empregando reator nuclear em 1956 propiciou a descoberta do neutrino [10]. Mais recentemente podemos citar o detector CHOOZ, na França. O detector tem um volume central de cintilador com massa de 5 t onde os $\bar{\nu}_e$ são detectados. Os resultados deste experimento [32] são consistentes com o desaparecimento de $\bar{\nu}_e$. O experimento Double CHOOZ segue ao experimento CHOOZ e emprega dois detectores idênticos, um posicionado a 400 m e outro a 1,05 km de distância dos núcleos dos reatores de Chooz. O experimento tem como objetivo por um valor não nulo do ângulo de mistura θ_{13} em um intervalo maior que o experimento CHOOZ.

2.4 O Modelo Padrão

O Modelo Padrão é a base teórica da física de partículas. Consiste da união do que aparentemente seriam três teorias diferentes: eletromagnetismo, interações fracas e interações fortes⁴. Eletromagnetismo e interações fracas foram unificados nas interações eletrofracas.

O Modelo Padrão descreve a natureza em termos de partículas elementares divididas entre as que constituem os blocos elementares de matéria (quarks e léptons) e as partículas mediadoras das interações, chamadas de bósons intermediários.

Os quarks apresentam um número quântico chamado “cor” que aparece em três variedades denominadas de *vermelho*, *azul* e *verde*⁵.

Os léptons não possuem “cor” e, portanto, não interagem fortemente. Enquanto cada um dos mediadores está associado a um tipo de interação: o fóton está associado com a interação eletromagnética; os bósons W^\pm e Z^0 estão associados à interação fraca e os glúons estão associados à interação forte. O Modelo Padrão tem sido bem sucedido na explicação da física de partículas, entretanto neutrinos com massa não nula podem ser considerados como evidência de uma física além do Modelo Padrão⁶.

A tabela 2.1 apresenta algumas propriedades das partículas do Modelo Padrão.

⁴A gravitação não faz parte do modelo padrão.

⁵O termo “cor” não apresenta qualquer relação com a definição usual de “cor”, a qual está ligada à luz visível.

⁶No Modelo Padrão não existe o neutrino destrógiro, o qual seria um dos mecanismos que permitiria geração de massa.

QUARKS		
Quarks	Massa	Carga elétrica
up (u)	1,7 a 3,3 MeV/c ²	$+\frac{2}{3}$
down (d)	4,1 a 5,8 MeV/c ²	$-\frac{1}{3}$
strange (s)	80 a 130 MeV/c ²	$-\frac{1}{3}$
charm (c)	1,18 a 1,34 GeV/c ²	$+\frac{2}{3}$
bottom (b)	4,19±0,18 GeV/c ²	$-\frac{1}{3}$
top (t)	172±0,9 GeV/c ²	$+\frac{2}{3}$
LÉPTONS		
Léptons	Massa	Carga elétrica
elétron (e)	0,510998910 ± 0,000000013 MeV/c ²	-1
neutrino eletrônico (ν_e)	< 2 eV/c ²	0
muon (μ)	105,6583668 ± 0,0000038 MeV/c ²	-1
neutrino muônico (ν_μ)	< 0,19 eV/c ²	0
tau (τ)	1776,82 ± 0,16 MeV/c ²	-1
neutrino tauônico (ν_τ)	< 18,2 eV/c ²	0
BÓSONS INTERMEDIÁRIOS		
Bósons	Massa	Carga elétrica
fóton(γ)	< 6×10^{-17} eV/c ²	0
W [±]	80,399 ± 0,023 GeV/c ²	±1
Z ⁰	91,1876 ± 0,0021 GeV/c ²	0
glúon (g)	0	0

Tabela 2.1: Partículas que compõem o modelo padrão de interações fundamentais da natureza [33]

Os neutrinos podem acoplar-se aos bósons portadores da interação fraca através de processos de corrente neutra (NC) ou corrente carregada (CC) como ilustrado na figura 2.1

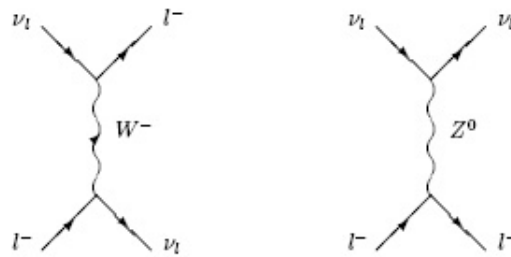


Figura 2.1: Diagramas de Feynman representando contribuições de corrente neutra (direita) e corrente carregada (esquerda) ao processo de espalhamento $\nu_l + l \rightarrow \nu_l + l$, onde l representa qualquer uma das três famílias de léptons.

Não há qualquer exigência intrínseca, no Modelo Padrão, que os neutrinos tenham massa zero. Entretanto, uma extensão do modelo é necessária para acomodar neutrinos com massa. Esta extensão pode ser conseguida pela inclusão de um neutrino destrógiro ν_R que não se acopla com os portadores da interação fraca, sendo, portanto, “estéril”. A busca por tais neutrinos “estéreis” é uma importante ferramenta de teste do modelo padrão. Até o momento tais neutrinos não foram encontrados.

Há várias questões abertas quanto à física de neutrinos. Por exemplo:

- Quais são as massas dos neutrinos?
- Qual é o padrão de mistura entre os diferentes tipos de neutrinos?
- Os neutrinos são as suas próprias antipartículas?
- Os neutrinos violam a simetria de CP?
- Existem neutrinos estéreis?
- O que os neutrinos podem nos dizer acerca dos modelos de nova física além do modelo padrão?
- Qual é o papel dos neutrinos na história cósmica ?

A resposta à maioria destas perguntas envolve o entendimento de como os neutrinos interagem com a matéria. A baixíssima secção de choque dos neutrinos faz com que hajam poucos dados e que a física dos neutrinos ainda seja pouco conhecida. O advento de feixes intensos de neutrinos, com energia e fluxo conhecidos e controlados, como o NuMI do

FERMILAB, abre a possibilidade para experimentos de neutrinos com alta estatística o que permitirá estudos detalhados e precisos de forma até hoje impossível.

2.5 Neutrinos no Modelo Padrão

O Modelo Padrão das interações fracas e eletromagnéticas é baseado na quebra espontânea da simetria de calibre $SU(2)_L \otimes U(1)_Y$ e foi proposto pela primeira vez em 1967 por S. Weinberg [34] e A. Salam [35]. Na mecânica quântica relativística, os férmions de spin $\frac{1}{2}$ e massa m são descritos pela equação de Dirac:

$$(i\hbar\gamma^\mu d_\mu - mc)\psi = 0 \quad (2.5)$$

onde γ^μ são as matrizes de Dirac [36] e ψ é o campo spinorial de quatro componentes. Segundo a teoria das interações fracas, são as projeções quirais esquerda e direita do campo de partículas de spin $\frac{1}{2}$ que participam de interações na teoria de unificação eletrofraca:

$$\psi = \psi_L + \psi_R = \left(\frac{1 - \gamma^5}{2}\right)_L \psi + \left(\frac{1 + \gamma^5}{2}\right)_R \psi \quad (2.6)$$

Como os neutrinos interagem apenas fracamente e, no modelo, são considerados sem massa, os estados destrógiros do neutrino não estão incluídos no modelo padrão. Os doze férmions conhecidos [33] (ignorando suas antipartículas) são, então, descritos como seis dubletos levógiros fracos e nove singletos destrógiros

$$\begin{pmatrix} e \\ \nu_e \end{pmatrix}_L \quad \begin{pmatrix} \mu \\ \nu_\mu \end{pmatrix}_L \quad \begin{pmatrix} \tau \\ \nu_\tau \end{pmatrix}_L \quad e_R \quad \mu_R \quad \tau_R \quad (2.7)$$

$$\begin{pmatrix} u \\ d' \end{pmatrix}_L \quad \begin{pmatrix} c \\ s' \end{pmatrix}_L \quad \begin{pmatrix} t \\ b' \end{pmatrix}_L \quad u_R \quad d_R \quad c_R \quad s_R \quad t_R \quad b_R \quad (2.8)$$

o que permite uma formulação simétrica. Isto é ilustrado na tabela 2.2, onde os fatores básicos de vértice das interações eletrofracas são escritos usando espinores de Dirac e as suas projeções ψ_L , ψ_R definidas na equação 2.6

Os fatores de vértice na tabela 2.2 revelam várias características importantes. Primeiro, o fóton é acoplado aos estados levógiros e destrógiros de uma forma similar, como é esperado para a interação electromagnética. As interações fracas de corrente carregada, no entanto, têm a forma familiar (V-A) que escolhe ψ_L . As interações fracas de corrente neutra NC dependem dos férmions envolvidos, mas para os neutrinos com $c_\nu = c_A = \frac{1}{2}$, apenas μ_L está envolvido.

A interação dos quarks de CC adicionalmente inclui a matriz, U^{CKM} (matriz de mistura de Cabibbo-Kobayashi-Maskawa) que descreve a relação matemática entre os estados de quark

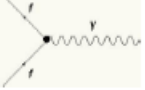
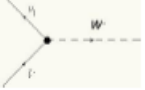

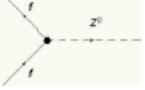
Interaction vertex	Dirac spinors	Chiral spinors
 QED fermions	$ig_e \bar{f} \gamma_\mu f$	$ig_e (\bar{f}_L \gamma_\mu f_L + \bar{f}_R \gamma_\mu f_R)$
 CC leptons	$\frac{-ig_w}{\sqrt{2}} \bar{\nu} \gamma_\mu \left(\frac{1 - \gamma^5}{2} \right) \ell$	$\frac{-ig_w}{\sqrt{2}} \bar{\nu}_L \gamma_\mu \ell_L$
 CC quarks	$\frac{-ig_w}{\sqrt{2}} \bar{q}_j \gamma_\mu \left(\frac{1 - \gamma^5}{2} \right) q_i U_{ij}^{CKM}$	$\frac{-ig_w}{\sqrt{2}} \bar{q}_{jL} \gamma_\mu q_{iL} U_{ij}^{CKM}$
 NC fermions	$\frac{-ig_z}{2} \bar{f} \gamma_\mu (c_V - c_A \gamma^5) f$	$\frac{-ig_z}{2} \bar{\nu}_L \gamma_\mu \nu_L$ for ν

Figura 2.2: Regras de Feynman para vértices de interações electrofracas em termos de spinores de Dirac e de suas projeções ψ_L e ψ_R

de massa definida (d, s, b), e os de sabor fraco definido (d', s', b'). Uma matriz equivalente de léptons, que relacione os estados observáveis fracos (ν_e, ν_μ, ν_τ), com os estados de massa dos neutrinos (ν_1, ν_2, ν_3) está visivelmente ausente neste formalismo.

Finalmente, um mecanismo de geração de massas de partículas é necessário. Não só é a massa da partícula uma realidade experimental com a qual uma teoria tem de lidar mas são as massas dos bósons de calibre W^\pm e Z^0 que fazem a força fraca ser de curtíssimo alcance ($\sim 10^{-18}m$) em relação ao alcance infinito da interação eletromagnética regida pelo fóton sem massa. As massas são geradas no modelo padrão através da quebra espontânea de simetria através do mecanismo de Higgs [37]. No modelo um dubleto complexo de campos escalares é introduzido [38]:

$$\phi \equiv \begin{pmatrix} \phi^1 \\ \phi^0 \end{pmatrix} \quad (2.9)$$

o que leva a termos adicionais de interação de Yukawa no lagrangiano do modelo padrão para cada família de léptons⁷

$$-\mathcal{L}_{Yukawa} = \sum_{\alpha, e, \mu, \tau} G_\alpha [\bar{\nu}_L \phi^1 l_R + \bar{l}_L \phi^0 l_R] + h.c \quad (2.10)$$

onde G_α são as novas constantes adimensionais de acoplamento (os índices de sabores ν_α e l_α são deixados de fora para maior clareza). Após a quebra espontânea de simetria, o valor

⁷Há termos similares para os quarks mas nos restringimos aqui os léptons porque estamos interessados apenas nas massas dos neutrinos.

esperado do campo de Higgs no vácuo torna-se $\langle \phi^+ \rangle = 0$ e $\langle \phi^0 \rangle = \frac{v}{\sqrt{2}}$, onde $v \simeq 246$ GeV, deixando os neutrinos sem massa e os léptons carregados com termos de massa de Dirac,

$$-\mathcal{L}_D = \sum_{l_\alpha, e, \mu, \tau} (m_D^l) \bar{l}_L l_R + h.c. \quad (2.11)$$

com um único parâmetro $m_D^l = G_l v / \sqrt{2}$ para cada lépton. Atualmente, as intensidades do acoplamento G_L não são determinadas pela teoria e só podem ser medidas experimentalmente.

2.5.1 Adicionando massas de neutrinos ao Modelo Padrão

Talvez a forma mais óbvia de dar massa aos neutrinos no modelo padrão seja adicionar o campo de neutrino ν_R destrógiro ao modelo o que, automaticamente, cria termos de massa de Dirac para cada neutrino ν , como o que ocorre com a equação 2.11 para os léptons carregados [39]

$$-\mathcal{L}_D = \sum_{\nu} (m_D^\nu) \bar{\nu}_L \nu_R + h.c. \quad (2.12)$$

Como os neutrinos são partículas neutras, um tipo diferente de termo de massa, conhecido como termo de massa de Majorana, também pode ser construído unicamente a partir de ν_L ou ν_R

$$-\mathcal{L}_{M_L} = \sum_{\nu} \frac{m_L^\nu}{2} \overline{(\nu_L)^C} \nu_L + h.c., \quad -\mathcal{L}_{M_R} = \sum_{\nu} \frac{m_R^\nu}{2} \overline{(\nu_R)^C} \nu_R + h.c. \quad (2.13)$$

onde $m_{L,R}^\nu$, são parâmetros com unidades de massa e $(\nu)^C = C(\bar{\nu})^T$, onde C é o operador de conjugação de carga. Os termos de massa de Majorana, portanto, envolvem a conversão de um neutrino em sua antipartícula o que seria proibido para outros férmions devido à violação da conservação da carga. Cabe à experimentação verificar se os termos da massa de Majorana existem. Se o neutrino é sua própria antipartícula então estes termos quase certamente existem. Se os neutrinos têm massa e se pode demonstrar que os termos de Dirac não existem, então os termos de Majorana são necessários.

2.5.2 Adicionando a mistura de neutrinos ao Modelo Padrão

Pelo fato de neutrinos interagirem apenas fracamente, não são os estados de massa que se pode acessar diretamente. Em vez disso, conhecemos os sabores $\nu_\alpha = (\nu_e, \nu_\mu, \nu_\tau, \dots)$ do neutrino através da identificação do lépton parceiro carregado $l_\alpha = (e, \mu, \tau, \dots)$ observado na interação fraca de Corrente Carregada ($\nu_\alpha A \rightarrow l_\alpha A'$)

$$-\mathcal{L}_{CC} = \sum_{\alpha, e, \mu, \tau} \frac{g}{\sqrt{2}} (\bar{\nu}_L \gamma_\mu l_L W_\mu^+ + \bar{l}_L \gamma_\mu \nu_L W_\mu^-), \quad (2.14)$$

onde, em benefício da clareza, mais uma vez deixamos de fora os subíndices de sabor. No entanto, uma propagação do neutrino através do espaço (por exemplo, desde a sua produção

até sua detecção) é ditada pelo Hamiltoniano livre cujos autoestados são estados de massa definida $\nu_i = (\nu_1, \nu_2, \nu_3, \dots)$ e cuja evolução temporal é descrita pela equação de Schrödinger:⁸

$$i \frac{d}{dt} |\nu_i(t)\rangle = E_i |\nu_i(t)\rangle \quad (2.15)$$

$$= \left(\sqrt{m_i^2 + p_i^2} \right) |\nu_i(t)\rangle \simeq \left(p_i + \frac{m_i^2}{2p_i} \right) |\nu_i(t)\rangle \approx \left(E_i + \frac{m_i^2}{2E_i} \right) |\nu_i(t)\rangle, \quad (2.16)$$

onde, nos últimos cálculos, supomos que $m_i \ll p_i$ e, portanto, $p_i \approx E_i$ para qualquer energia razoável do neutrino usada em um experimento⁹. A solução trivial para a equação 2.16 é, simplesmente

$$|\nu_i(t)\rangle = e^{-i(E_i + m_i^2/2E_i)t} |\nu_i(0)\rangle \quad (2.17)$$

Precisamos, agora, introduzir uma descrição de como os autoestados fracos $|\nu_\alpha\rangle$, que participam da interação, relacionam-se aos autoestados de massa $|\nu_i\rangle$, que se propagam através do espaço. Introduzimos uma matriz unitária de mistura U tal que

$$|\nu_i\rangle = \sum_{\alpha} U_{\alpha i} |\nu_\alpha\rangle, \quad (2.18)$$

$$|\nu_\alpha\rangle = \sum_i U_{\alpha i}^* |\nu_i\rangle, \quad (2.19)$$

que implica que um neutrino produzido como um autoestado de sabor $|\nu_\alpha\rangle$ é a superposição quântica de autoestados de massa com amplitudes determinadas pelos elementos de U . Os autoestados de massa que contribuem de forma coerente para um feixe experimental são aqueles com uma energia comum [13] de modo que a equação 2.17 implica que cada estado de massa se propaga com uma única fase complexa que depende de m_i^2 . Além disso, como o neutrino é ultra-relativista, podemos considerar a distância de propagação ao invés do tempo, $L \ll t$ (para $c = 1$). Assim, um neutrino produzido como autoestado fraco é, na distância L

$$|\nu_\alpha(L)\rangle = \sum_i U_{\alpha i}^* e^{-i(m_i^2/2E)L} \quad (2.20)$$

Portanto, a probabilidade de um neutrino ser produzido como um autoestado fraco α e ser detectado como um autoestado fraco β a uma distância L é dada por

⁸em toda esta seção empregamos unidades naturais onde $\hbar = c = 1$.

⁹para $m < 1$ eV e $E \sim 1$ GeV, $\gamma \geq 10^9$

$$P(\nu_\alpha \rightarrow \nu_\beta) = |\langle \nu_\beta | \nu_\alpha(L) \rangle|^2 = \left| \sum_i U_{\alpha i}^* e^{-i(m_i^2 L/2E)} U_{\beta i} \right|^2 \quad (2.21)$$

$$= \delta_{\alpha\beta} - 4 \sum_{i>j} R(U_{\alpha i}^* U_{\beta i} U_{\alpha j} U_{\beta j}^*) \sin^2 \left(\Delta m_{ij}^2 \frac{L}{4E} \right) \quad (2.22)$$

$$+ 2 \sum_{i>j} I(U_{\alpha i}^* U_{\beta i} U_{\alpha j} U_{\beta j}^*) \sin \left(\Delta m_{ij}^2 \frac{L}{2E} \right) \quad (2.23)$$

Onde $\Delta m_{ij} \equiv m_j^2 - m_i^2$ é a diferença entre os quadrados das massas dos autoestados $|\nu_j\rangle$ e $|\nu_i\rangle$. A forma senoidal da probabilidade de transmutação deu origem à denominação de “oscilação de neutrinos” a este efeito. Há várias observações sobre o fenômeno de oscilações de neutrinos como descrito pela equação 2.23:

- Se os neutrinos não têm massas degeneradas de forma que todos $\Delta m^2 = 0$, então a equação 2.23 reduz-se a $\delta_{\alpha\beta}$ e os neutrinos não podem alterar o sabor através de oscilações. Por outro lado, se os neutrinos oscilam, então isso indica que uma ou mais massas dos neutrinos são necessariamente diferente de zero e não idênticas.
- Se a matriz de mistura de neutrinos é diagonal, de tal forma que os autoestados de massa não se misturam, então a equação 2.23 uma vez mais se reduz a $\delta_{\alpha\beta}$ e a mudança de sabor através das oscilações não pode ocorrer.
- Para determinar a probabilidade de oscilação de antineutrinos devemos substituir a matriz de mistura por sua matriz complexa conjugada mudando para (-) o sinal do terceiro termo. Como $\bar{\nu}_\alpha \rightarrow \bar{\nu}_\beta$ é a imagem especular em CP de $\nu_\alpha \rightarrow \nu_\beta$ então $P(\nu_\alpha \rightarrow \nu_\beta) \neq P(\bar{\nu}_\alpha \rightarrow \bar{\nu}_\beta)$ seria evidência de violação CP no sector leptônico.
- Oscilações só são sensíveis às diferenças de massa e, portanto, não é possível determinar as massas absolutas dos neutrinos. No entanto, pode ser possível determinar o sinal de Δm^2 comparando as probabilidades de oscilação de neutrino e de antineutrino para neutrinos passando através de matéria densa [40].
- Incluindo as demais potências de \hbar e c e fazendo a conversão de $m(eV)$, $L(km)$ e $E(GeV)$ para unidades experimentais úteis, os argumentos das funções trigonométricas na equação 2.23 tornam-se, respectivamente:

$$\sin^2 \left(1,267 \frac{\Delta m^2 L}{E} \right), \quad \sin \left(2,534 \frac{\Delta m^2 L}{E} \right) \quad (2.24)$$

- Os parâmetros L e E determinam os valores de Δm^2 para os quais as oscilações podem ser observadas experimentalmente¹⁰. Se $(1, 267\Delta m^2 L/E) \ll 1$ então o efeito seria insignificante. Efeitos das oscilações são maiores para valores de $(1, 267\Delta m^2 L/E)$ de $\mathcal{O}(1)$.

¹⁰As diferenças de massa Δm^2 são de ordem de $1eV^2$, as distâncias de oscilação são de ordem de quilômetros e as energias são de ordem de GeV

Capítulo 3

O experimento MINER ν A

3.1 Feixes de neutrinos em aceleradores

Desde o primeiro uso de um feixe de neutrinos [41] em 1962, muitas linhas de feixe de neutrino em aceleradores foram construídas. Desde o ano 2000 cinco linhas de feixe de neutrinos entraram em funcionamento ou estão sendo construídas: duas no Japão (KEK e JPARC), duas nos Estados Unidos (BNB e NuMI, ambos no FERMILAB) e uma na Europa (o SPS do CERN). Neste trabalho empregamos o feixe NUMI do FERMILAB e para o detector MINER ν A que sera descrito nas seguintes seções.

Cada feixe de neutrinos tem sua própria configuração com sensibilidades e incertezas próprias. Basicamente, o feixe de neutrinos é produzido a partir do decaimento de mésons produzidos em interações de um feixe primário de hádrons (em geral prótons) com um alvo. O que determina o conjunto relevante de medidas dos hádrons é:

- Energia do próton primário (tipicamente entre 10 GeV e 500 GeV). O número médio de píons produzidos por próton incidente, $\langle n_\pi \rangle$, depende linearmente do momentum do próton incidente [42].
- Os alvos nucleares tendem a ser feitos de elementos mais leves (como Be, C e Al) devido à dificuldade de dissipação de calor em alvos de alto Z. Além disto as taxas de interações secundárias são também reduzidas em alvos de baixo Z.
- Os alvos são espessos (1 a 2 comprimentos de interação) para aumentar as taxas de reação do próton, mas buscando-se um compromisso para minimizar as interações secundárias. A importância das interações secundárias aumenta com o momentum do próton incidente figura 3.1.
- A produção de mésons relevantes fora do alvo é, geralmente, para a frente, mas pode

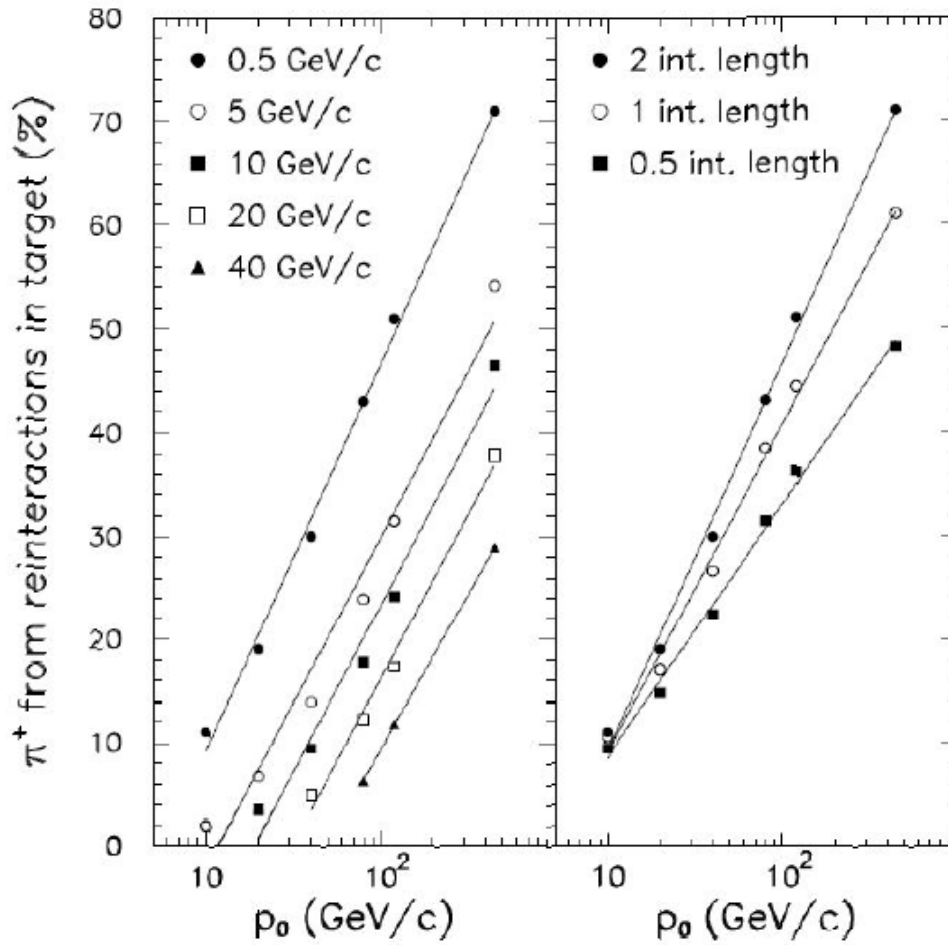


Figura 3.1: Fração de π^+ escapando de um alvo espesso de grafite (partículas terciárias) como função do momentum do próton incidente, p_0 . O gráfico da esquerda mostra a fração de terciários para alvo com 2,0 comprimentos de interação para π^+ com momentum $p_z > 0,5$ GeV/c; 5,0 GeV/c, etc. O gráfico da direita mostra a fração total de terciários para alvos de diferentes comprimento de interação, 0,5 λ , 1,0 λ e 2,0 λ . Figura extraída de [42]

se estender até 20 graus no sistema de referência do laboratório devido aos sistemas focalizadores.

- As medições da produção de π^+ , π^- , K^+ , K^- e K^0 são necessárias para que se entenda completamente os feixes de ν_μ , $\bar{\nu}_\mu$, ν_e , $\bar{\nu}_e$.

As combinações específicas da energia do feixe incidente de prótons e os alvos nucleares para os feixes de neutrinos existentes estão representados na figura 3.2.

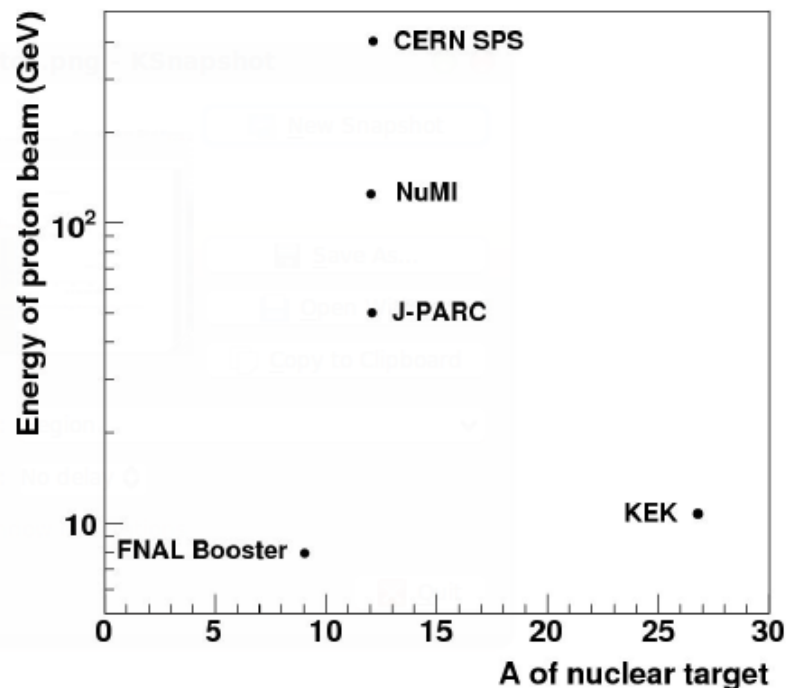


Figura 3.2: Energia do próton incidente versus massa atômica do alvo nuclear para os cinco feixes de neutrinos em aceleradores existentes no mundo

3.2 O feixe NuMI

A linha de feixe NuMI do FERMILAB foi projetada para produzir um feixe de neutrinos muônicos de alta intensidade. Um feixe de prótons de 120 GeV, extraído do anel de armazenagem do Main Injector, colide com um alvo de grafite produzindo várias partículas. Mésons carregados são focalizados por um sistema de magnetos focalizadores e, depois, decaem em uma área mantida em vácuo, produzindo neutrinos. A figura 3.3 mostra os principais componentes da linha de feixe NuMI. Uma descrição detalhada do feixe NuMI pode ser encontrada em [43] e [44].

A produção do feixe NuMI começa com a aceleração do feixe primário de prótons envolvendo três etapas: o LINAC, o “booster” e o “Main Injector”. O LINAC acelera os prótons até 400 MeV e os repassa ao “booster” que os acelera até 8 GeV. O “Main Injector” acelera os prótons até atingirem a energia de 120 GeV.

Os prótons são, então, extraídos e enviados em direção ao alvo de grafite. Todo o processo dura cerca de 1,9 s. As salvas de prótons resultantes contêm cerca de $2,5 \times 10^{13}$ prótons e tem duração de cerca de 9,7 μ s.

O alvo de grafite mede 0,95m de comprimento, é segmentado e refrigerado a água. O alvo tem espessura de apenas 6,4 mm, para minimizar reabsorção, e representa 1,9 comprimentos

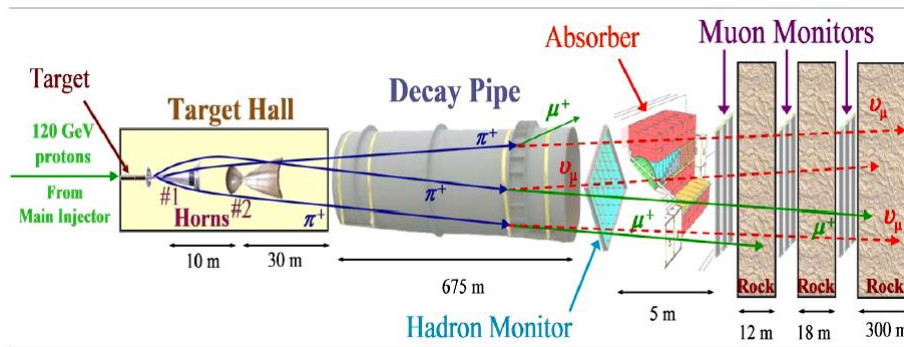


Figura 3.3: Principais componentes da linha de feixe NuMI. Mésons positivos, produzidos no alvo, são focalizados por dois cones magnéticos e decaem em uma região de 675 m mantida em vácuo, produzindo, predominantemente, ν_{μ} . Os hádrons restantes são bloqueados em um absorvedor de feixe colocado ao final da região de decaimento.

de interação hadrônica para maximizar a produção de mésons.

Os mésons positivos criados no alvo são focalizados por um par de dispositivos magnéticos de focalização chamados de cornetas (horns). As cornetas do NuMI são como lentes magnéticas parabólicas que criam um campo toroidal com valor de pico de 3 T. Elas são operadas em modo pulsado com uma corrente nominal de 200 kA e resfriadas por pulverização de água [43].

Após a focalização, os píons e káons entram em um cilindro, em vácuo, onde decaem produzindo um μ e um ν_{μ} . Esta região de decaimento consiste em um cilindro de 675 m de comprimento, 2 m de diâmetro e é mantido à pressão residual de, aproximadamente, 1 torr ou menor.

Hádrons que cheguem ao final desta região são parados por um absorvedor de feixe, que consiste de um núcleo de alumínio refrigerado à água envolvido por uma camada de blocos de aço e uma camada externa de concreto¹.

O feixe de neutrinos resultante consiste de 97,8% ν_{μ} e possui poucos $\bar{\nu}_{\mu}$ (1,8%) e ν_e (0,4%), estes últimos tendo como origem o decaimento do muon.

Uma das características do feixe NuMI está na capacidade de ajuste do espectro de energia do feixe de neutrinos. A figura 3.4 mostra três diferentes configurações possíveis: baixa energia (LE), média energia (ME) e alta energia (HE).

As diferentes energias são obtidas movendo-se o alvo e/ou a segunda corneta focalizadora, de forma similar ao movimento de lentes em um sistema óptico. Para que isto possa ocorrer, o alvo está montado em um sistema de trilhos que permite movê-lo ao longo do feixe por uma distância de até 2,5 m. Com isto, píons e káons de *momenta* diferentes são selecionados e focalizados na região de decaimento, resultando em diferentes espectros de energia dos neutrinos.

¹Os múons, porventura ainda existentes, são parados nos 300 m de rocha que separam a área experimental do absorvedor.

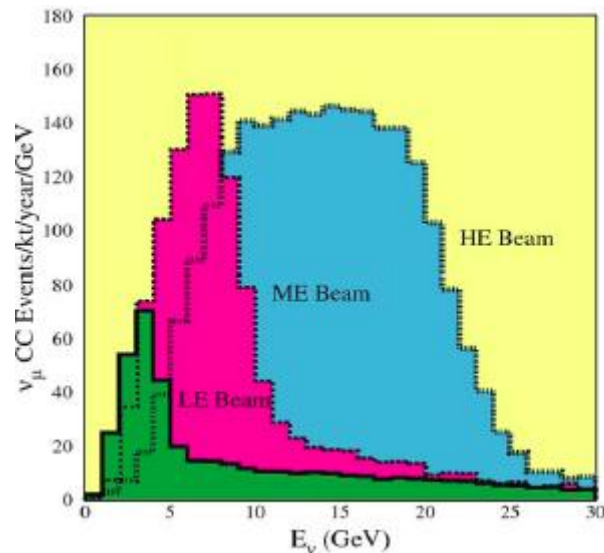


Figura 3.4: Diferentes configurações do feixe NuMI. O espectro do feixe NuMI pode ser alterado movendo-se o alvo e/ou a segunda corneta focalizadora.

A polarização das cornetas pode mudar de tal maneira que os mésons focalizados sejam de sinal oposto, permitindo ao NUMI produzir tanto feixe de neutrinos como de anti-neutrinos.

A linha NuMI fornece um feixe intenso de neutrinos para o experimento de oscilação de neutrinos MINOS. O *Near Detector* do MINOS² está instalado em uma grande área subterrânea (a 100 m de profundidade) construída no FERMILAB. Esta área possui espaço suficiente para que seja instalado um experimento de espalhamento ν e $\bar{\nu}$ -nucleon de alta estatística e alta resolução. O experimento MINER ν A, que é objeto desta tese, tem seu detector instalado imediatamente à montante do *Near Detector* do MINOS e medirá secções de choque de neutrinos e verificará os efeitos nucleares com a dependência do número atômico. A alta intensidade do feixe NuMI permite que possamos investigar pela primeira vez diversos aspectos de uma grande variedade de assuntos importantes na física de neutrinos (ou aumentar significativamente o conhecimento em áreas até aqui pouco investigadas ou com pouca estatística).

As figuras 3.5 e 3.6 mostram o detector MINER ν A.

3.3 Descrição geral

Para realizar todos os estudos a que se propõe, o detector do MINER ν A precisa estabelecer um novo paradigma em termos de experimento de neutrinos³. Com estados finais tão variados como reações inelásticas com alta multiplicidade, produção coerente de píons e espalhamento

²nome usado por ser o detector mais próximo ao início do feixe NuMI

³Uma apresentação mais detalhada do detector pode ser encontrada em [46].

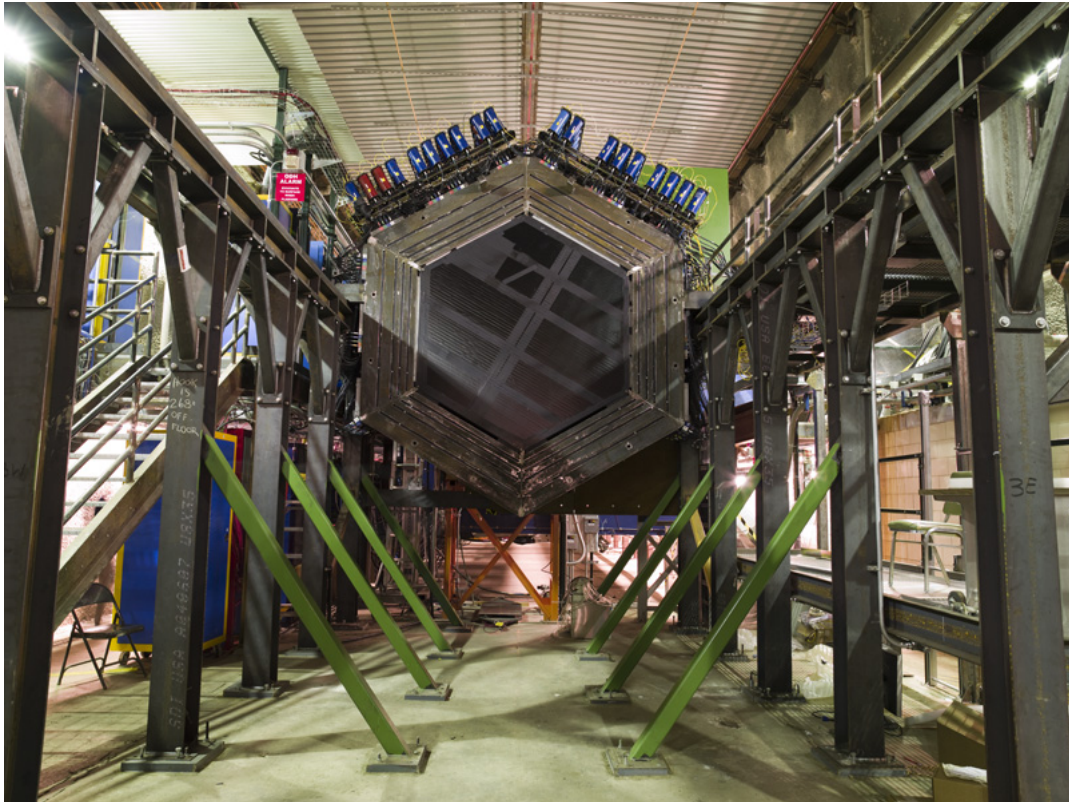


Figura 3.5: Vista do MINER ν A montado em sua estrutura de suporte.

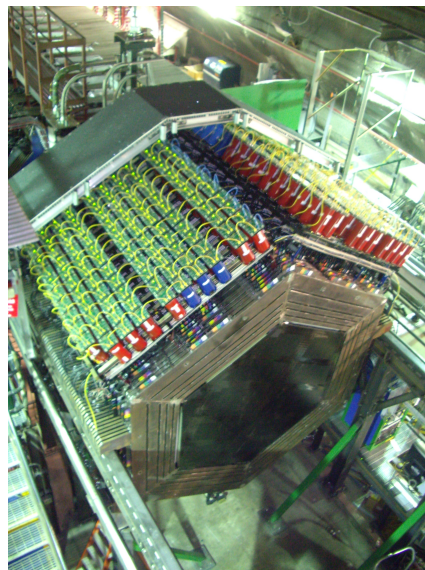


Figura 3.6: Vista de cima do MINER ν A. Os MAPMT destacam-se no topo do detector.

quase-elástico⁴ de neutrino, o detector é um híbrido de um calorímetro tradicional com um

⁴Em física de neutrinos o termo quase-elástico é empregado para descrever um processo no qual um neutrino interage com um núcleon e produz um lépton carregado no estado final. O termo refere-se ao fato de que o neutrino do estado inicial transforma-se em um lépton diferente e há apenas um único núcleon no estado final.

detector ativo⁵ altamente segmentado. O detector deve ser capaz de:

- Identificar múons e medir seus *momenta* com grande precisão.
- Identificar hádrons e π^0 individuais e medir seus *momenta*.
- Medir a energia de chuviros hadrônicos e electromagnéticos com uma melhor precisão.
- Minimizar a “confusão” entre eventos de corrente-neutra e de corrente-carregada.
- Acomodar diversos alvos nucleares.

Estes objetivos podem ser alcançados com um detector/alvo compacto consistindo, basicamente, de barras cintiladoras. O núcleo do detector MINER ν A consiste de um detector constituído de barras de cintilador sólido. Este detector plástico interior, conhecido como *Inner Detector* (ID) serve como o volume primário onde um rastreamento preciso de partículas é possível. Um material de baixa densidade e amostragem fina assegura que algumas das medidas mais difíceis possam ser feitas. Isto inclui, entre outras, detecção de prótons de recuo em eventos quase-elástico e identificação de partículas através de dE/dx . A figura 3.7 mostra uma vista lateral do detector.

O detector central é cercado, por todos os lados, pelos calorímetros electromagnético, hadrônico e um identificador de múons. O detector tem forma aproximadamente hexagonal com secção reta variando de 3,35 m a 4,10 m. O comprimento é de 5,9 m. A massa total de cintiladores é de cerca de 6 toneladas. À montante do detector encontram-se os alvos nucleares consistindo de, aproximadamente, 0,7 kg de Fe, 850 kg de Pb, 200 kg de He líquido e 600 kg de H_2O . O detector apresenta significativa granularidade, o que assegura precisão na reconstrução de vértices graças ao uso de cintiladores plásticos em forma de prismas triangulares com 3,3 cm de base, 1,7 cm de altura e comprimento de cerca de 4,0 m, com uma fibra óptica inserida em sua base para leitura do sinal. Barras semelhantes são empregadas nos calorímetros.

Nas baixas energias necessárias para estudo de seções de choque, muitos eventos contêm partículas que se diringem para as laterais e para trás.

Assim, os detectores de amostragem estendem-se para os lados, e mesmo para trás, do detector. Múons energéticos penetram no Detector Próximo do MINOS onde seus *momenta* podem ser medidos. Assim, o detector próximo do MINOS opera como um detector auxiliar do MINER ν A.

Exceto pelo veto à montante, o Detector do MINER ν A é segmentado transversalmente no ID com planos de barras cintiladoras sólidas. Por construção e conveniência, um plano do

⁵Chamamos de detector ativo quando a mesma estrutura exerce as funções de alvo e detecção simultaneamente.

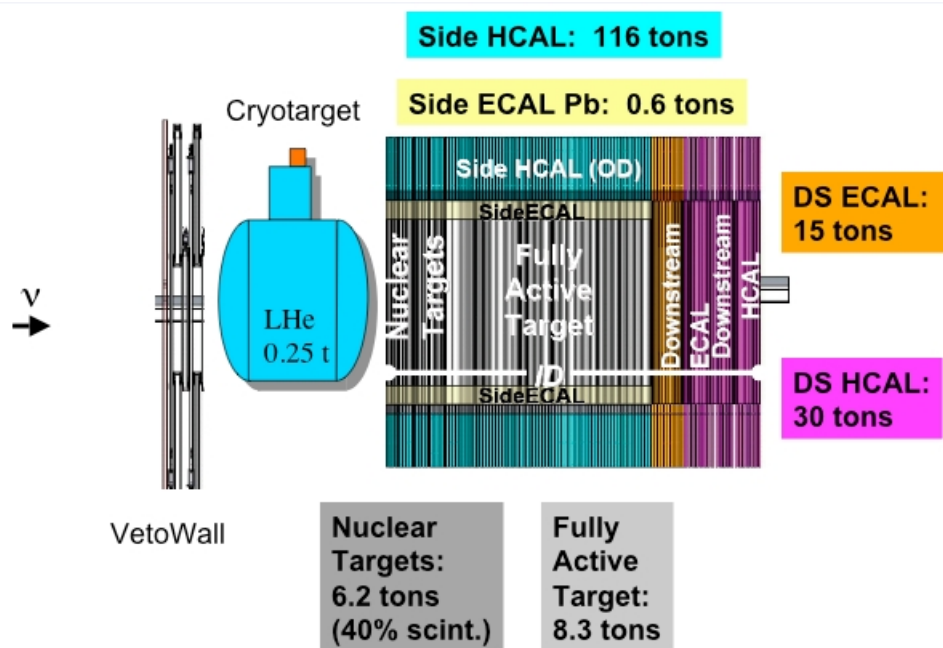


Figura 3.7: Vista esquemática lateral do detector do MINER ν A

MINER ν A incorpora as estruturas do *Inner Detector* (ID) e do *Outer Detector* (OD), além da estrutura externa de suporte. Dois planos de cintilador são montados em uma estrutura, chamada de módulo como ilustra na figura 3.8. Há 3 distintas orientações no ID, deslocadas de 60° e chamadas X, U e V. As diferentes orientações permitem uma reconstrução tridimensional de trajetórias e o deslocamento de 60 graus faz do hexágono a forma natural da seção reta transversal do detector.

A figura 3.9 mostra uma vista de frente do detector olhando-se na direção de onde vem o feixe. A figura humana ilustra as dimensões do detector.

O centro do detector é o *Inner Detector* (ID) (totalmente ativo), cujo núcleo plástico representa o volume efetivo para a maioria das análises do MINER ν A. A figura 3.10 ilustra os módulos da região ativa central. Detectores calorimétricos na região central do detector são construídos com a inserção de absorvedores entre planos adjacentes. A figura 3.11 ilustra um módulo do calorímetro eletromagnético.

Placas de chumbo de 0,2 cm de espessura são inseridas entre os planos de cintilador e na frente de cada módulo, para atuar como calorímetro eletromagnético lateral. Isto representa a maior parte do comprimento do detector, e o *Outer Detector* (OD) representa a maior parte da massa do detector.

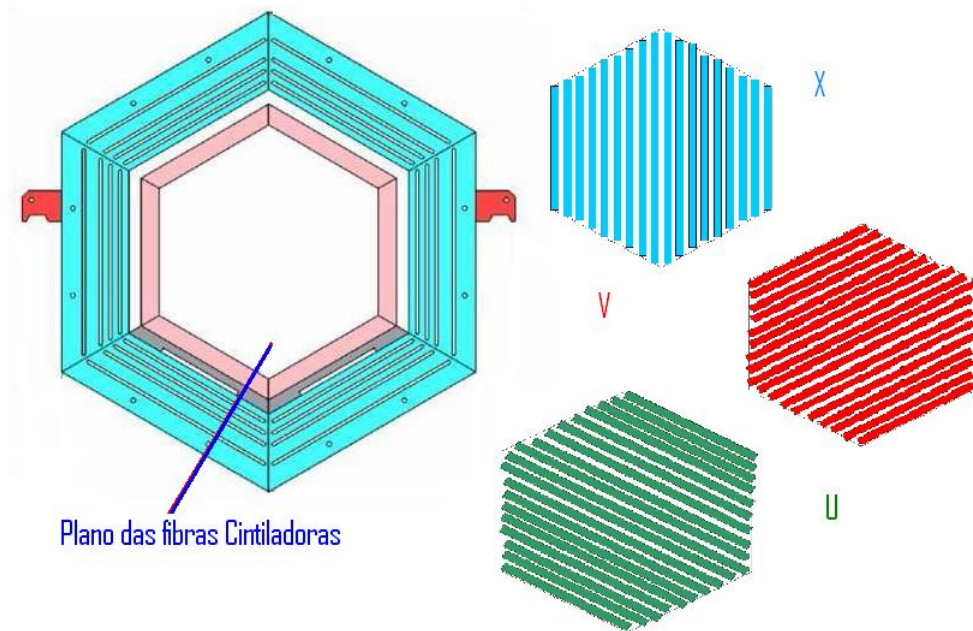


Figura 3.8: Vista de um módulo ativo do detector. A figura à direita mostra os três planos de fibras cintiladoras U,V e X deslocados de ± 60 graus.



Figura 3.9: Vista do detector do MINER ν A ilustrando forma e tamanho. Podemos ver os Alvos de Ferro e Chumbo

O Inner Detector é envolvido por sistemas de absorvedores e cintiladores que fazem o *Outer Detector* (OD). O Outer Detector compõe-se de seis “torres” (um sexto do hexágono).

A parte mais à jusante do detector tem os calorímetros hadrônicos (HCALs) com absorvedores de 1 polegada por plano cintilador como mostrado na figura 3.12. Os calorímetros

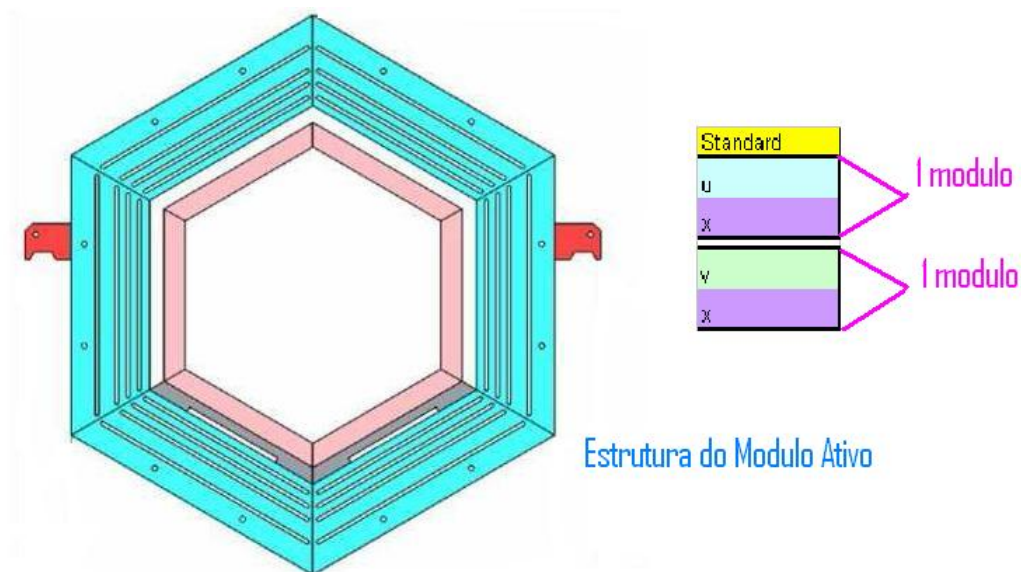


Figura 3.10: Vista de um módulo ativo do detector. A figura à direita mostra a estrutura deste tipo de módulo, UX ou VX.

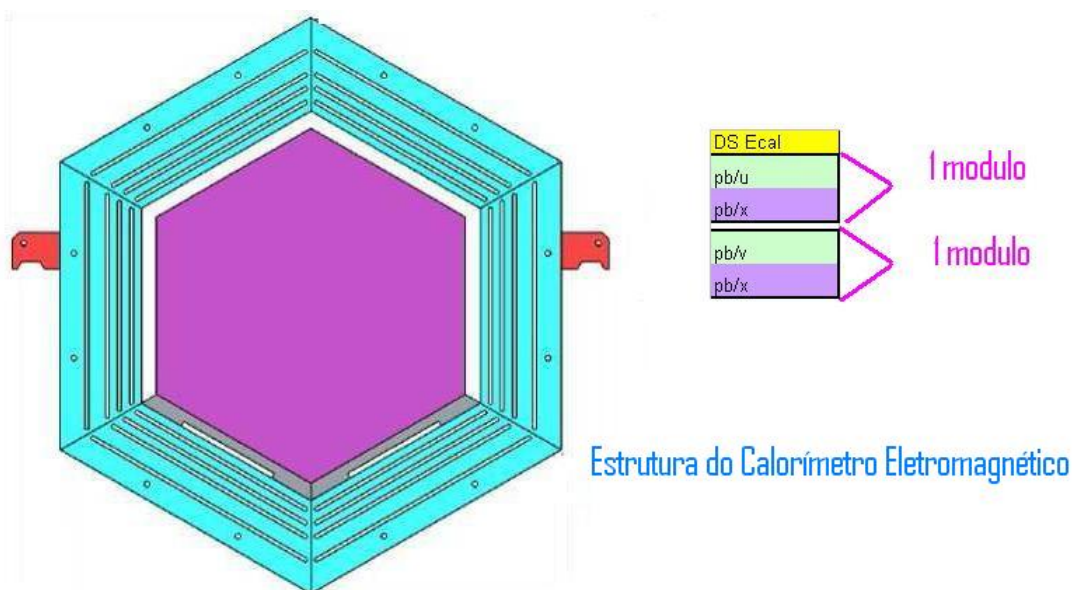


Figura 3.11: Vista de um módulo do calorímetro eletromagnético. A figura à direita mostra a estrutura do calorímetro, alternando capas de chumbo com capas cintiladoras.

eletromagnéticos possuem absorvedores de Pb de 0,2 cm por plano. A alta granularidade do Calorímetro Electromagnético (ECAL) assegura resolução de energia de fótons e elétrons além

de permitir a medida da direção de ambos.

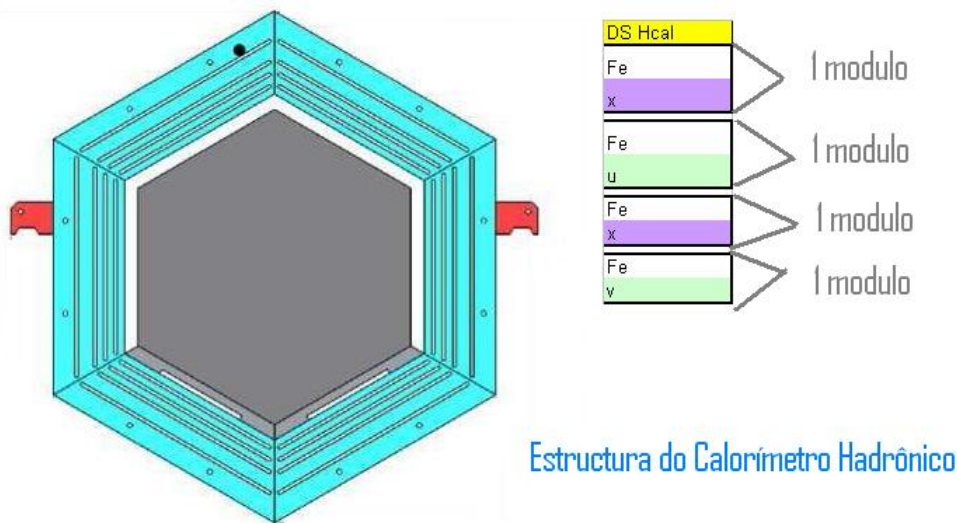


Figura 3.12: Vista de um módulo do calorímetro hadrônico. A figura à direita mostra a estrutura do calorímetro, alternando capas de ferro com capas cintiladoras.

Por fim, a região assinalada como Alvos Nucleares (*nuclear targets*) na figura 3.7 contém absorvedores colocados entre alvos ativos de forma que é possível o estudo de eventos em diferentes alvos nucleares. Estes alvos são descritos na seção 3.7.

3.4 Detectores de partículas

A fim de compreender a física dos detectores de partículas precisamos de um conhecimento adequado da interação das partículas com a matéria. Ao atravessar a matéria uma partícula interage com seus constituintes microscópicos e a interação específica depende tanto das propriedades inerentes da partícula (carga, massa, spin, sabor, etc.) quanto de sua energia. Por exemplo, fótons de algumas centenas de keV chocando-se com uma parede de concreto interagem primariamente com os elétrons atômicos componentes da parede.

3.4.1 Perda de energia de partículas carregadas

Partículas carregadas atravessando a matéria perdem energia, principalmente, por excitação, ionização ou Bremsstrahlung. O poder de parada é uma quantidade útil para descrever a perda de energia em um material e é definida como a perda média de energia por unidade de comprimento:

$$\left\langle \frac{-dE}{d\chi} \right\rangle = \left\langle \frac{-dE}{d(\rho x)} \right\rangle \quad (3.1)$$

A figura 3.13 mostra o poder de parada para μ^+ atravessando o cobre como função de $\beta\gamma$ e processo predominante em cada região de energia.

Excitação e região de ionização

Uma partícula carregada, com uma energia relativística moderada, perde energia principalmente por interações electromagnéticas com os elétrons atômicos.

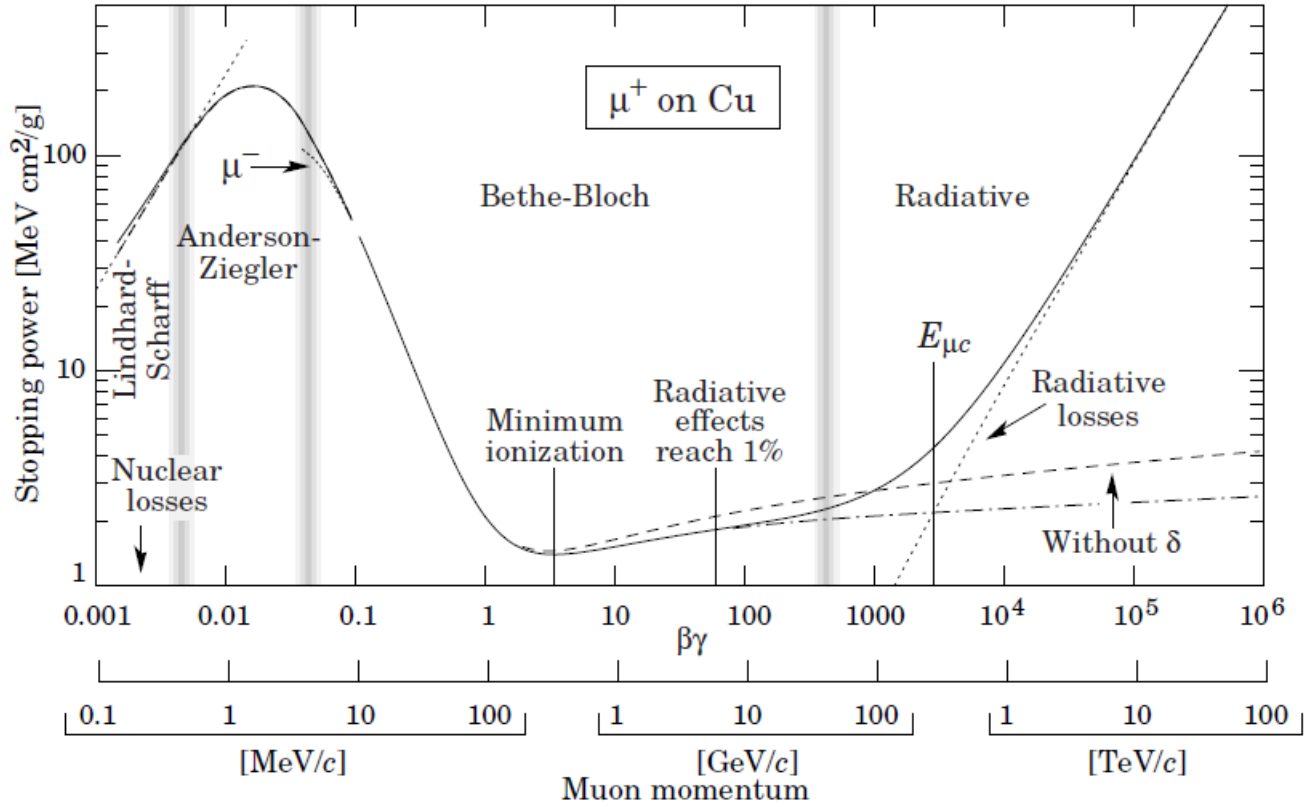


Figura 3.13: Poder de parada para μ atravessando cobre em função do momentum. Extraído de [33]

Quando a partícula incidente não é um elétron a perda média de energia é descrita pelo formalismo de ionização de Bethe-Bloch [33]

$$\langle -dE/dx \rangle = K \frac{Z}{A} \frac{z^2}{\beta^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 T_{max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right] \quad (3.2)$$

Onde:

K	$4\pi N_A r_e^2 m_e c^2 = 0,307075 \text{ MeV cm}^2 \text{ g}^{-1}$
z	Carga da partícula incidente
$Z(A)$	Número Atômico (massa) do material
m_e	massa do elétron
I	Energia de excitação média $\approx 16Z^{0,9} \text{ eV}$, $Z > 1$
T_{max}	energia cinética máxima que pode ser transferida a um elétron livre em uma colisão
$\delta(\beta\gamma)$	correção de efeito da densidade

A equação de Bethe-Bloch é muito útil na física de partículas pelo fato de que maioria das interações das partículas com os detectores ocorre nos regimes de energia de validade desta equação. Como a equação 3.2 depende exclusivamente de β , esta deposição de energia é frequentemente usada para identificação de partículas de baixo momentum⁶.

Região Radioativa

Em altas energias as partículas diminuem sua velocidade, devido ao campo electromagnético dos constituintes dos átomos, emitindo fótons. Esta desaceleração é chamada Bremsstrahlung e é o canal dominante em altas energias onde o campo electromagnético é mais forte como pode ser visto na figura 3.14.

Além disso, é de particular importância para os elétrons que a secção de choque do processo é proporcional ao inverso da massa da partícula espalhada. Bremsstrahlung e a produção de pares levam ao desenvolvimento de chuviscos electromagnéticos que é um fenômeno muito utilizado em calorimetria.

3.4.2 Interação de fótons

Fótons interagem com a materia através dos seguintes canais:

Efeito Fotoelétrico

No efeito fotoelétrico, o fóton perde energia na ionização dos elétrons atômicos. Esta seção de choque aumenta drasticamente quando é atingido o potencial de ionização de cada camada atômica (K, L, \dots). Este canal domina em baixas energias como pode ser visto na ver figura 3.15.

Efeito Compton

⁶A equação de Bethe-Bloch precisa receber alguns termos extras para compensar efeitos de baixa energia e para o caso em que a partícula incidente um elétron.

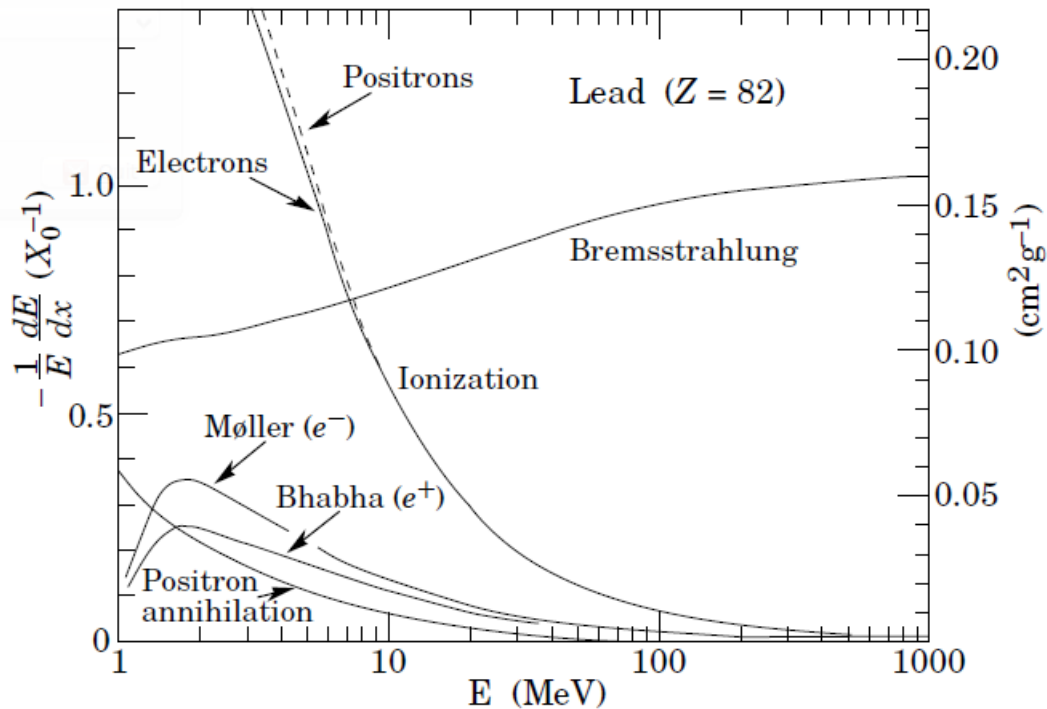


Figura 3.14: Fração de energia perdida pelos elétrons e pósitrons em função da energia. Extraído de [47].

Espalhamento inelástico domina em energias médias, como visto na figura 3.15. O fóton incidente transfere parte da sua energia para os elétrons atômicos mais externos. Uma descrição deste efeito foi fornecido por A. Compton no século dezenove e é compactada na formula de Klein-Nishina onde assumimos que o elétron está em repouso.

Produção de pares

Para energias maiores que 1,2 MeV, o fóton perde energia⁷ na maior parte por produção de pares $e^+ e^-$. Veja [33] para referência a fundo.

3.5 Os dispositivos fotossensores

O custo maior do detector do MINER ν A vem dos dispositivos fotossensores e eletrônica associada. No MINER ν A o sinal de luz é suficiente para empregar fotocatodos com eficiência quântica de 15% e fibras de diâmetro de 1,2 mm. MINER ν A emprega fotomultiplicadoras

⁷Este valor de energia é imposto pela massa das partículas produzidas.

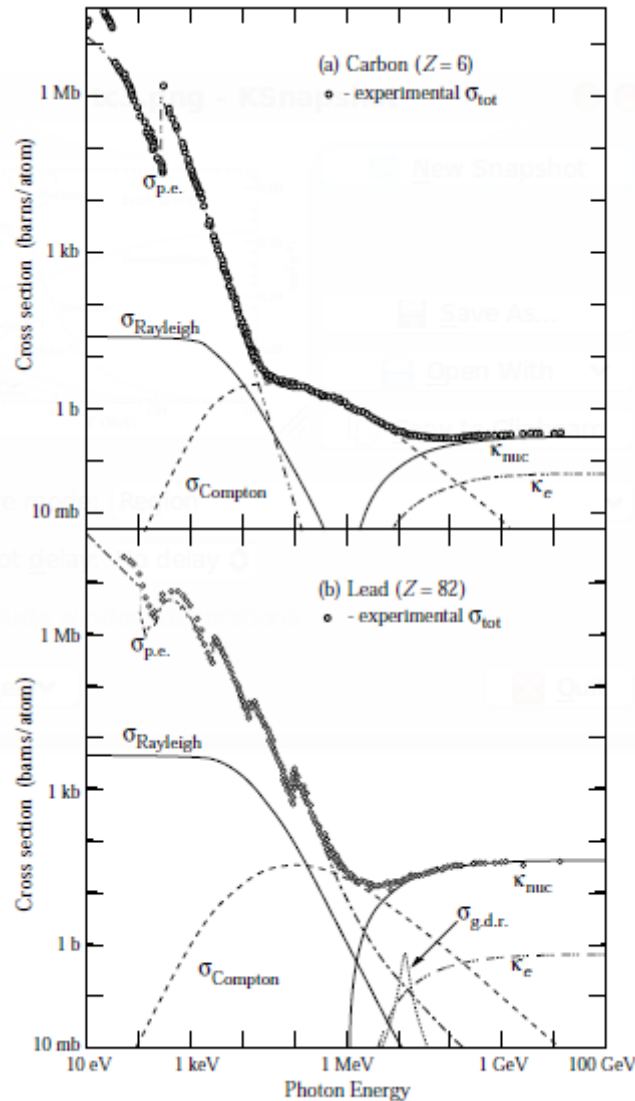


Figura 3.15: Seções de choque para canais de interação de fótons com a matéria. Extraído de [47]

multi-anodo HAMAMATSU H8804 com 64 canais.

3.6 Os elementos cintiladores

No *Inner Detector* (ID), os elementos sensíveis do MINER ν A são cintiladores em forma de prismas triangulares, com 1,7 cm de altura por 3,3 cm de base, feitos de poliestireno (Dow 663) e dopados com POP (1% por peso) e POPOP (0,03% por peso) revestidos com uma cobertura reflectiva de TiO_2 com fibras óticas alteradoras de comprimento de onda (WLS) (Kurraray Y11 com dopante 175 ppm) de 1,2 mm incrustadas (figura 3.16). As fibras WLS são

levadas a conectores óticos nas extremidades dos módulos e fibras óticas claras levam a luz até fotomultiplicadoras multianodo. Para melhorar a resolução de coordenadas, estes elementos triangulares são montados em planos (figura 3.17 e figura 3.18); isto permite repartir carga entre tiras de cintiladores vizinhas num simples plano para se obter, por interpolação, a coordenada.

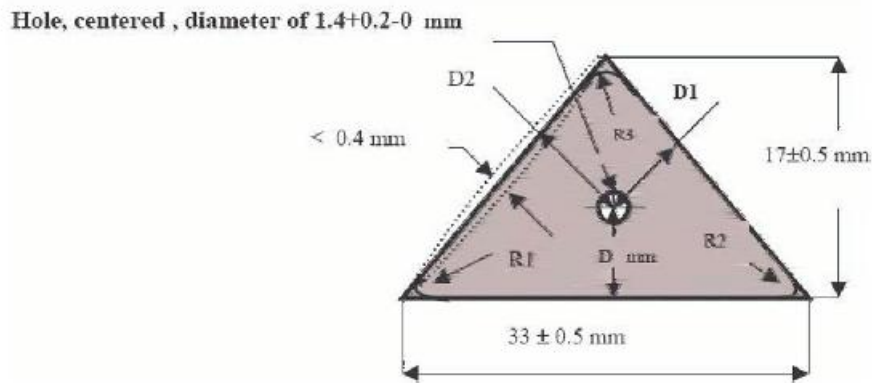


Figura 3.16: Corte transversal de um prisma triangular cintilador empregado no *Inner Detector*. Nota-se o furo por onde a fibra ótica é inserida.

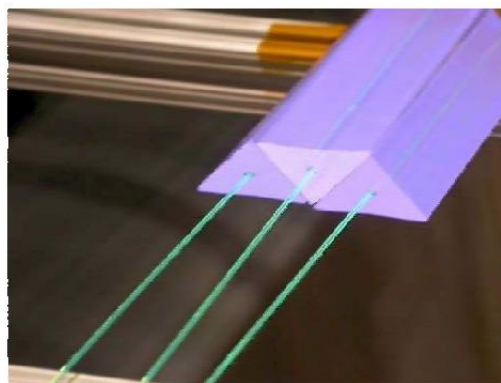


Figura 3.17: Prismas triangulares cintiladores arranjados de forma a constituírem um plano cintilador. Cada prisma corresponde a um canal. Cada prisma tem uma fibra ótica inserida ao longo de seu comprimento.

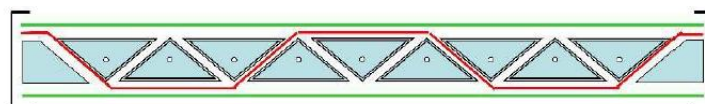


Figura 3.18: Vista de corte de um plano montado com prismas triangulares cintiladores.

3.7 Calorímetros

Os objetivos do MINER ν A requerem a realização de medidas de energias de partículas carregadas (p , π^\pm , K^\pm , μ^\pm) e neutras (π^0 , K^0) com energias de até poucos GeV. Para atingir estes objetivos, o MINER ν A possui os seguintes elementos: um conjunto de chumbo e cintilador alternados à jusante e envolvendo o detector ativo para calorimetria electromagnética, um conjunto de aço e cintilador alternados à jusante e na parte exterior do detector para calorimetria hadrônica, e placas de chumbo, aço e carbono à montante do detector central para calorimetria electromagnética e hadrônica. Os calorímetros estão concebidos de forma a que chuveiros originados na região central sejam completamente contidos no sistema.

3.7.1 Calorímetro eletromagnético

A detecção de fótons de alta energia dá-se através do processo de produção de pares que leva a um chuveiro de e^+ , e^- e γ . Como a seção de choque de produção de pares é proporcional a Z^2 , folhas de chumbo são, em geral, utilizadas para produção de um chuveiro de tamanho razoável. O comprimento característico dos chuveiros varia com a energia. Porém, para fótons de até uns poucos GeV, como esperado em nosso experimento, 99% da energia estará contida em 4 cm de Pb (correspondente a 7 comprimentos de radiação).

O calorímetro electromagnético à jusante consiste de 20 camadas de Pb de 2 mm de espessura, alternadas com camadas de cintiladores feitas a partir dos cintiladores em forma de prisma triangular já mencionados. A resolução de energia esperada é de $6\%/\sqrt{E}$, com E dado em GeV. A figura 3.11 ilustra um módulo do ECAL. O calorímetro lateral é similar. Folhas de chumbo, também de 2 mm de espessura, são alternadas com camadas de cintiladores. Fótons que penetrem o calorímetro em ângulos de até 25° em relação à linha do feixe, são absorvidos. Para ângulos maiores eles não serão totalmente contidos, mas penetrarão no calorímetro hadrônico externo, onde o chuveiro restante está totalmente contido.

Como o objetivo principal das placas de Pb, Fe e C à montante é servir como alvo, a calorimetria nesta região não é tão eficiente quanto nos módulos à jusante e nas laterais. A amostragem é mais grosseira pois as placas de Pb, Fe e C são mais espessas. O arranjo dos alvos implica que o número de comprimentos de interação visto pelo chuveiro nesta região, antes de escapar do detector, varia entre 5 e 10. Entretanto, uma vez que os fótons nesta direção terão energia muito menor, os chuveiros que se iniciem na região central também estarão totalmente contidos.

3.7.2 Calorímetro hadrônico

O calorímetro hadrônico à jusante está posicionado logo após o calorímetro eletromagnético e consiste de 20 camadas de ferro de 2,54 mm de espessura alternadas com placas cintiladoras. A ação combinada de 4 cm de Pb e 50 cm de Fe para múons de até 600 MeV e prótons de até cerca de 800 MeV. O comprimento de interação nuclear do Fe é 16 cm. Assim, prótons (ou píons) de maior energia também serão, provavelmente, detidos. A figura 3.12 ilustra um módulo do HCAL. O calorímetro hadrônico lateral consiste de uma placa de ferro de 55,9 cm de espessura, com 5 aberturas, cada uma com 2,5 cm, preenchidas com cintiladores. A espessura total de ferro é 43,4 cm, que é suficiente para parar prótons de até 750 MeV que penetrem a 90° e prótons de até cerca de 1 GeV que penetrem a 30°.

A resolução do calorímetro hadrônico é esperada ser cerca $50\%\sqrt{E}$ para hádrons com energia acima de 1 GeV. A resolução para partículas de menor energia é esperada ser 50% ou menos, dependendo da energia.

3.8 Alvos nucleares

Os alvos nucleares do MINER ν A consistem de carbono, ferro e chumbo além de He e H_2O . O Ferro foi escolhido por ser um material barato e por ser um absorvedor comum em experimentos com neutrinos, como o MINOS. Chumbo é o material nuclear de Z mais alto que é de fácil obtenção. Idealmente os alvos nucleares deveriam ser feitos de muitos alvos finos com várias camadas rastreadoras entre cada alvo de forma a determinar-se a multiplicidade de estados finais e a quantidade de energia de cada uma das partículas de baixa energia. Alguns fatores, entretanto, limitam o número e tamanho dos alvos assim como o número de camadas rastreadoras. A resolução espacial intrínseca do detector é da ordem de 1 cm na direção do feixe. Assim, alvos mais finos seriam ineficientes. Placas com espessura maior que 2,5 cm (como as usadas no MINOS) não permitiriam uma melhora significativa do conhecimento do espectro de partículas de baixa energia, que é um dos objetivos do MINER ν A. Para aumentar a estatística deveríamos ter cerca de 1 tonelada de cada alvo. Restrições práticas, entretanto, limitam o tamanho do alvo que podemos implementar.

O MINER ν A decidiu implementar um alvo seguindo o seguinte esquema onde cada F indica uma estrutura XU ou XV (um par FF é um conjunto XUXV): FF Pb/Fe [alvo 1] FFFF Pb/Fe [alvo 2] FFFF Pb/Fe/C [alvo 3] FFFF Pb[alvo 4] FFFF Pb/Fe [alvo 5]. Alvos 1 e 2 têm 60% de Fe e 40% de Pb representando cerca 230 kg de Fe e de Pb em cada alvo. O alvo 3 tem 50% de C, 30% de Fe e 20% de Pb representando cerca de 140 kg de C e 110 kg de Fe e 110 kg de Pb. Alvo 4 é de Pb puro com massa total de 170 kg. Alvo 5 tem Pb e Fe, sendo 60% de Fe e 40%

Material do Alvo	Mass (ton)	Amostra de corrente carregada
Hélio	0,25	0,6 M
Hidrocarbono	3	8,6 M
Carbono	0,6	1,4 M
Ferro	1	2,9 M
Aço	1	2,9 M
Água	0,6	1,4 M

Tabela 3.1: Quantidade de eventos de corrente carregada nos alvos nucleares

de Pb, cada um com 115 kg. No total, temos 685 kg de Fe e 855 kg de Pb. O número total de eventos que vão ocorrer nos alvos nucleares de CC (corrente carregada) é de: 2 milhões para o Fe, 2,5 milhões para o Pb e 400.000 para o C. Ilustrações da configuração dos alvos podem ser vistas na figura 3.19. O alvo de água é visto na figura 3.20. Consiste de uma caixa de formato hexagonal, de mesma dimensões que o detector, posicionada imediatamente antes dos demais alvos. O alvo de He será instalado imediatamente à montante do detector, como ilustrado na figura 3.7.

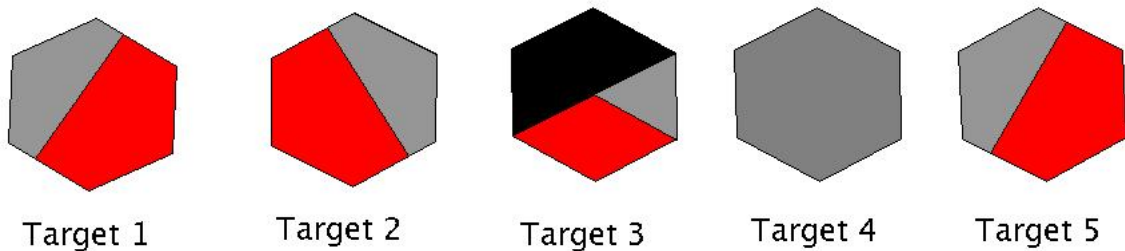


Figura 3.19: Alvos nucleares: Vermelho = Fe, Cinza = Pb, Preto = C

Figura 3.20: Alvo de água do experimento MINER ν A .

3.9 Eletrônica e aquisição de dados (DAQ)

A tabela 3.2 agrupa e resume as exigências para a eletrônica do MINER ν A. Estas exigências são motivadas pelos seguintes objetivos:

- Resolução espacial fina explorando o compartilhamento de luz entre barras cintiladoras vizinhas;
- Identificação de π^\pm e p através da informação de dE/dx ;
- Reconhecimento de padrões eficientes, empregando timing para identificar a direção da trajetória e separar interações que ocorram dentro do mesmo ciclo (*spill*);
- Habilidade para identificar partículas estranhas e decaimento de múons empregando técnicas de coincidência;
- Tempo morto desprezível dentro de um ciclo (*spill*).

A taxa média de dados esperada para o MINER ν A é de aproximadamente 100 kBytes/s.

Parâmetro	valor
ciclo (<i>spill</i>)	12 μ s
Tempo de repetição	>1,9 s
Número de canais	30.972
Ocupação por ciclo	2%
Variação de ganho do detector fotossensível	4,5 dB
Resolução temporal	3 ns

Tabela 3.2: Algumas das exigências e parâmetros para o projeto da eletrônica do MINER ν A

As exigências para o DAQ do MINER ν A são modestas devido à taxa relativamente baixa de eventos. O tempo previsível do feixe torna um sistema de gatilho complexo desnecessário. No lugar de um tal sistema, simplesmente temos um sinal de *gate* que se abre imediatamente antes da chegada do feixe e todas as informações de carga e tempo de todo o detector são simplesmente registradas após o término do *spill*.

3.9.1 Hardware do DAQ

Os elementos ativos do detector do MINER ν A têm seus sinais enviados para fotomultiplicadoras multianodo (MAPMT) de 64 canais. A informação de amplitude e tempo dos sinais das MAPMT é digitalizada pela eletrônica e guardada para leitura pelo sistema de aquisição

de dados (DAQ). Cada placa eletrônica de leitura (FEB) é conectada a uma única fotomultiplicadora.

Grupos de até 10 FEB são lidos e o resultado enviado para um módulo controlador de leitura de bastido (CROC - *Crate Read-Out Controller*) montado em um bastidor VME. Cada CROC pode acomodar 4 cadeias de leitura de FEB. Um total de 12 CROCs são necessários para todo o detector do MINER ν A. Os bastidores VME também alojam um módulo de interface do CROC (CRIM *CROC Interface Module*), um módulo de tempo (MTM - *MINER ν A Timing Module*) e uma fonte de 48 V. Não há CPU nos bastidores VME.

Durante a tomada de dados, o sistema DAQ funciona durante toda a salva de neutrinos. Ao final do período de 10 μ s o DAQ lê todos os canais que apresentem sinal acima de um certo pedestal. Mesmo com alta taxa de ocupação, o número total de *bytes* lidos a cada salva de neutrinos é menor que 200 kB com supressão de zeros (1 Mb sem supressão de zeros). O tempo morto é desprezível.

A fonte de alimentação de 48 V alimenta as bases das fotomultiplicadoras. MINER ν A emprega o mesmo *hardware* para aquisição de dados e para o sistema de controle do detector (DCS - *Detector Control System*). A mesma conexão que faz a leitura dos FEB também fornece o canal de comunicação e controle do detector como, por exemplo, o controle das voltagens dos MAPMT. Os computadores principais para o DAQ e para o sistema de controle lento (o sistema de controle e monitoramento de variáveis cujo valor sofra alterações lentas) localizam-se próximos da eletrônica VME conectados à rede do FERMILAB através de duas linhas de alta velocidade TCP/IP. Uma servidora de duas CPU realiza estas tarefas tendo uma CPU dedicada à aquisição de dados e a outra ao controle e monitoramento. As máquinas do sistema DAQ rodam no linux e encontra-se entre os computadores do DAQ e a rede geral.

3.9.2 *Software de Aquisição de dados (DAQ)*

O software do MINER ν A é montado no arcabouço GAUDI empregando a infraestrutura cliente-servidor GAUCHO, originalmente desenvolvidos para a colaboração LHCb. A taxa média de dados esperada para eventos sem supressão de dados é de apenas 100 kB/s e uma janela de dois segundos é disponível após cada salva de neutrinos de 10 μ s. A alta previsibilidade do feixe torna desnecessário o estabelecimento de um sistema complicado de gatilho. No lugar de tal sistema, o MINER ν A simplesmente ativa a eletrônica do detector e toda informação é lida ao final de cada salva. O sistema de controle lento é, igualmente, simples tendo cada MAPMT sua própria fonte de alta tensão local e os módulos FEB fazem a leitura das altas tensões, temperaturas e outros parâmetros empregados para monitoração e controle. Um diagrama esquemático é visto na figura 3.21.

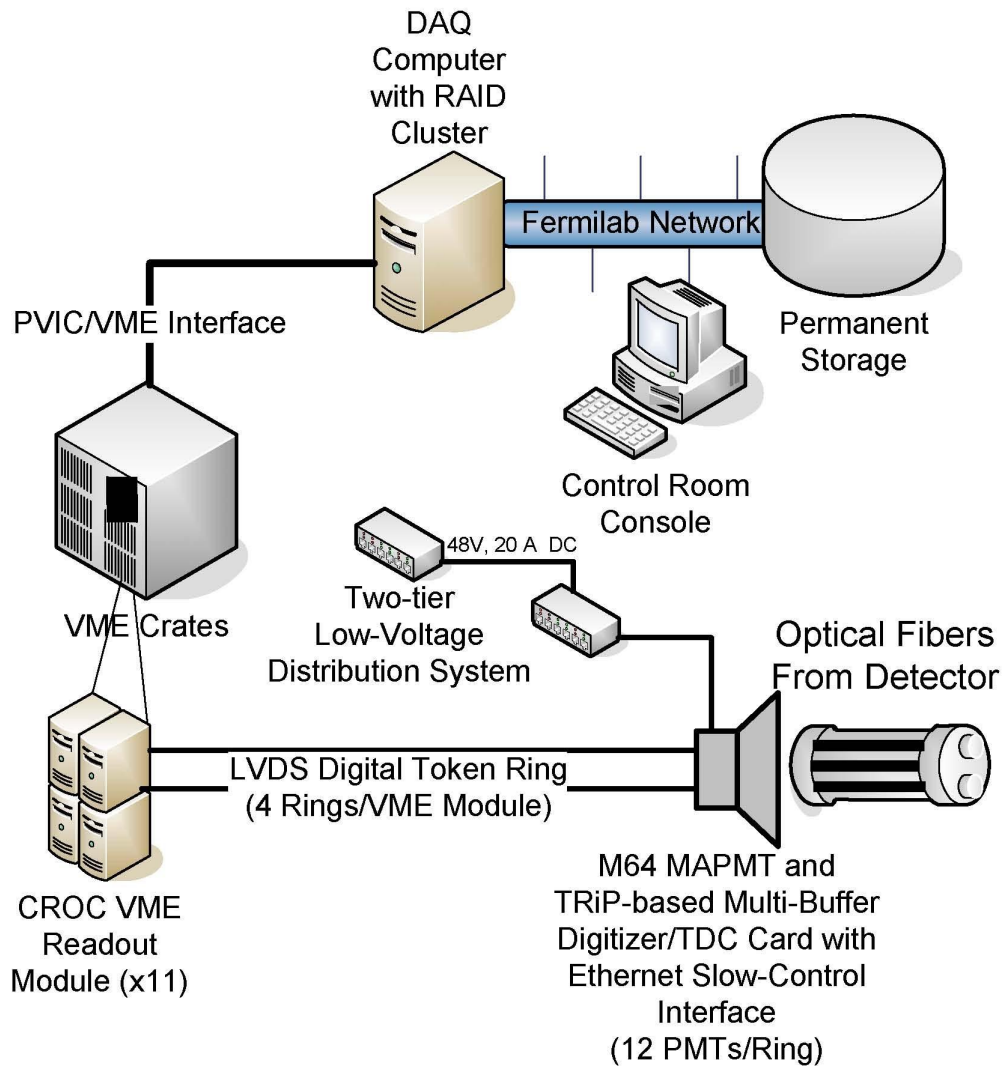


Figura 3.21: Diagrama esquemático do sistema de aquisição de dados do MINER ν A

3.10 Parâmetros do detector do MINER ν A

A tabela 3.3 apresenta o número total de canais do detector do MINER ν A em seus subdetectores. Os canais são dominados pela granularidade nos alvos de plástico, Pb e Fe. Quanto a massa, o domínio cabe ao *Outer Detector(OD)*. O MINER ν A emprega 19,2 toneladas de cintiladores, 93 km de fibras ópticas cintilantes e outros 43 km de fibras ópticas claras entre o detector e 587 fotomultiplicadoras multianodos de 64 canais cada.

Sub-detector	ID	OD	massa (toneladas)
Alvo ativo e ECAL	15.360	5.760	14,6
ECAL à montante (Alvo de Pb)	3.072	1.152	3,5
HCAL à montante (Alvo de Fe)	1.536	576	7,0
ECAL à jusante	2.560	960	19,8
HCAL à jusante	2.560	960	26,4
Veto	426	n.a.	15,1
OD envolvendo as regiões do alvo	n.a.	n.a.	126,5
Total	25.514	8.408	211,3

Tabela 3.3: Parâmetros do detector do MINER ν A.

3.11 Testes

Vários subprojetos ocorrem paralelo à construção do detector do MINER ν A. Realçamos aqui dois deles: o mapeador de módulos e o teste das fotomultiplicadoras MAPMT.

3.11.1 Mapeador

O mapeador é um sistema que varre um módulo cintilador do detector com duas fontes radioativas e lê a resposta da fotomultiplicadora. Os resultados obtidos em cada pedaço do módulo durante cada passagem das fontes radioativas são digitalizados e armazenados. O DAQ incumbe-se de acumular os dados e de controlar o movimento das fontes. Este processo permite-nos encontrar variações ao longo das tiras cintilantes e realizar um mapeamento completo de cada um dos módulos.

A figura 3.22 ilustra o mapeador isolado e sobreposto a um módulo.

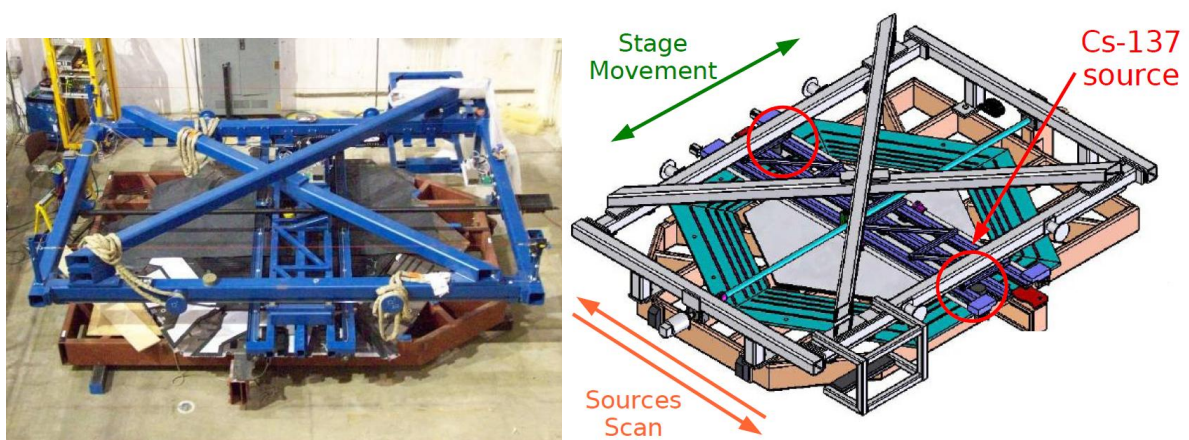


Figura 3.22: Mapeador.

3.11.2 Teste das fotomultiplicadoras

Grupos de MAPT passam por testes intensivos onde pulsos de luz são enviados a cada um dos elementos da fotomultiplicadora em diversas configurações. Os sinais das fotomultiplicadoras são digitalizados e armazenados. Estes dados são usados para levantamento completo das características de todas as MAPMT usadas no detector. Este sistema permite determinamos parâmetros como ganho, diafonia, linearidade e interferência entre canais. A estrutura montada permite o teste simultâneo de 4 fotomultiplicadoras.

A figura 3.23 ilustra a estrutura empregada para os testes das MAPMT. Parte de nosso trabalho consistiu na realização de testes de operação dos MAPMT e encontra-se descrito no Apêndice A.



Figura 3.23: Montagem de teste das MAPMT.

3.11.3 Detector de Teste

A função deste detector é fornecer uma descrição clara da resposta do detector do MINER ν A a partículas específicas. É fundamental para a calibração do detector do MINER ν A. Emprega um feixe de hádrons com energia entre 200 MeV e 1 GeV.

Este detector é uma réplica em pequena escala do MINER ν A, sem os alvos nucleares e construído com peças mais curtas dos mesmos cintiladores empregados no MINER ν A, empregando as mesmas MAPMT e eletrônica. O objetivo principal é verificar como o MINER ν A responde a partículas individuais, especialmente prótons e píons. Isto inclui informação sobre a energia total de chuveiros hadrônicos, dE/dx ao longo das trajetórias, resolução de energia para estas quantidades, topologia da formação dos chuveiros e identificação de partículas. As informações obtidas por este detector de teste servirão de entrada para ajuste da simulação do detector (que emprega GEANT4) e para a calibração da energia durante a fase de reconstrução. É, também, importante para o desenvolvimento de padrões de reconhecimento e medida das eficiências.

Os padrões de espalhamento e absorção de píons em núcleos não são bem modelados nas simulações de Monte Carlo atuais.

O detector de teste do MINER ν A contém 40 planos com orientação XUXV como no detector completo do MINER ν A. A figura 3.24 ilustra o detector de teste. As tomadas de dados no detector de teste começaram em abril de 2010.

O feixe empregado neste detector e sua caracterização constituem a parte principal desta tese e são descritos nos capítulos 4 e 5.

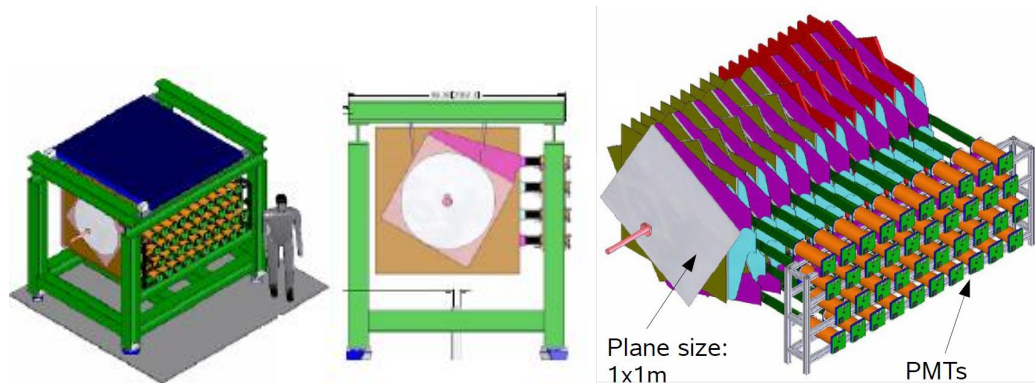


Figura 3.24: O detector de testes do MINER ν A.

3.11.4 Protótipo de rastreamento

O protótipo de rastreamento consiste em 20% do detector completo, possuindo calorimetria e a parte central do detector. A figura 3.25 ilustra este protótipo. Várias tomadas de dados empregando múons cósmicos são realizadas com este protótipo com vistas à realização de testes de operação do detector. O protótipo permite ao experimento MINER ν A verificar a operação e funcionamento do detector antes da montagem final do detector completo com testes do *hardware*, DAQ e reconstrução de eventos.

3.12 Fase de Testes

A fase de testes com o prototipo do experimento MINER ν A começou em outubro de 2008 e estendeu-se até o segundo semestre de 2009. O protótipo de rastreamento e o detector de prova usaram, cada um, 100 MAPMT implicando em eventos de cerca de 200 kB. A taxa de eventos para o detector de prova é de cerca de 10 Hz (muito maior que a taxa de eventos do detector completo que será da orde de 0,5 Hz). Na fase de testes coletamos cerca de 250 GB por semana⁸ o que demandou um espaço em disco de 10 TB até o final de 2009.

⁸Convém observar que isto é a quantidade que o D0/CDF escrevem por hora.

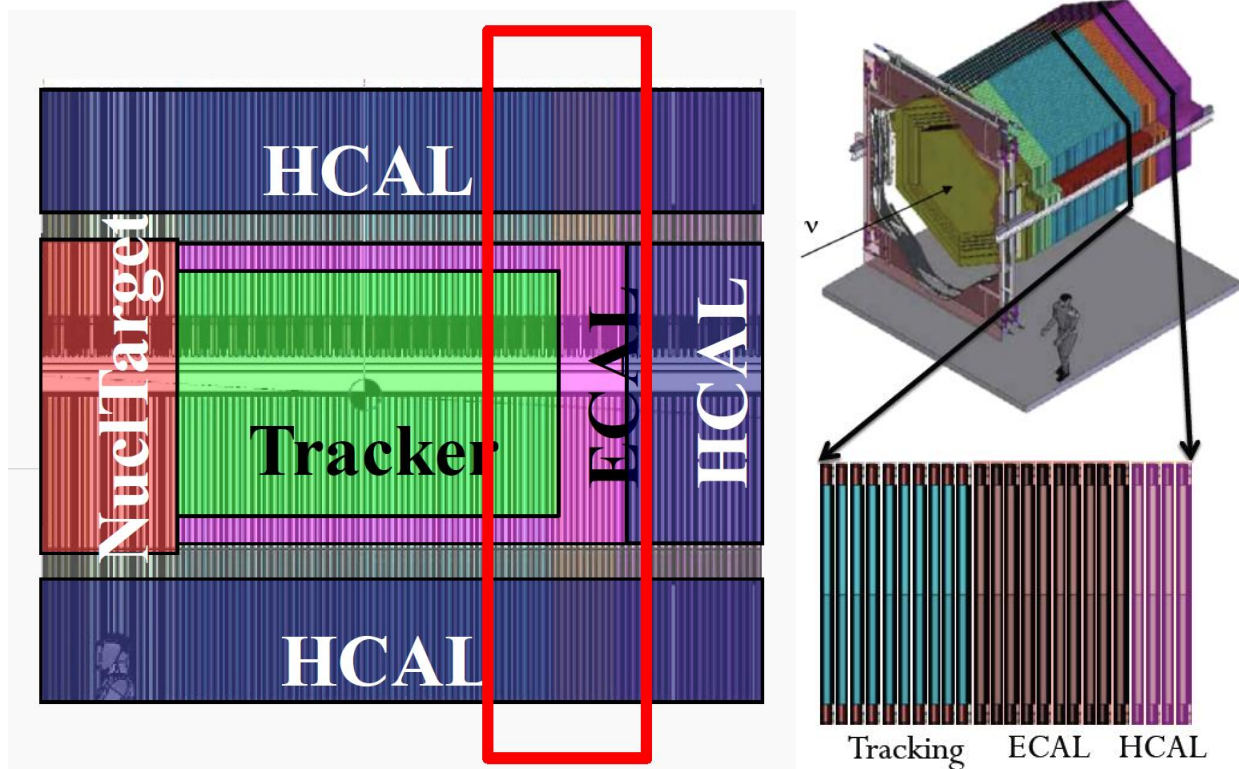


Figura 3.25: Esquema do detector do MINER ν A mostrando a parte que corresponde ao protótipo de rastreamento

3.13 O detector completo

O experimento completo emprega cerca de 500 MAPMT. Um evento sem supressão de zeros é da ordem de 1 MB. A maioria dos eventos, entretanto, será obtido com supressão de zeros e é da ordem de 100 kB ocorrendo a cada 2 segundos. A taxa de eventos encontra-se, então, entre 5 GB e 10 GB por dia correspondendo a algo entre 1,5 TB e 3 TB por ano. Em quatro anos de tomada de dados o MINER ν A acumulará cerca de 6 TB a 12 TB de dados. Tudo isto será mantido em discos rígidos.

O experimento MINER ν A trabalhará em associação com o experimento MINOS de forma a ter acesso a alguns de seus dados que serão empregados na detecção de múons.

3.14 Perspectivas do experimento MINER ν A

Na faixa de 1 a 10 GeV dados relevantes vêm de experimentos com câmaras de bolhas entre 1960 e 1980. A câmara de bolhas Gargamelle do Argonne, de 12 pés, a câmara de 7 pés de Brookhaven, BEBC no CERN, SKAT em Serpukov e a câmara de 15 pés do FERMILAB estudaram interações de neutrinos e anti-neutrinos com núcleos.

Apesar da baixa estatística, a excelente resolução de imagem das câmaras de bolhas tornou possível a realização de uma grande gama de experimentos. No entanto os dados dos experimentos com câmara de bolhas, além de poucos, não cobrem todo o espectro de energias dos neutrinos, alvos nucleares e tipos de neutrinos ($\nu/\bar{\nu}$) necessário para um entendimento completo das interações de neutrinos. A tabela 3.4 resume a situação experimental daqueles projetos.

Experimento	Ano	Reação	Medida	Eventos	Referência
Gargamelle	1977	$\nu/\bar{\nu}$ -propano/freon	semi-inclusiva	ν :1.061	[49]
	1977	$\nu/\bar{\nu}$ -propano/freon	produção de π	$\bar{\nu}$:1.200	
Gargamelle	1978	$\bar{\nu}$ -propano/freon	$\bar{\nu}(\pi^0)$	139	[50]
	1978	$\bar{\nu}$ -propano/freon	$\bar{\nu}(\pi^-)$	73	
Gargamelle	1978	ν -propano/freon	$\nu p \rightarrow \nu p \pi^0$	240	[51]
	1978	ν -propano/freon	$\nu p \rightarrow \nu p \pi^+$	104	
	1978	ν -propano/freon	$\nu p \rightarrow \nu n \pi^0$	31	
	1978	ν -propano/freon	$\nu p \rightarrow \nu n \pi^+$	94	
Gargamelle	1979	$\nu/\bar{\nu}$ -propano/freon	$\nu(1\pi^0)$	178	[52]
	1979	$\nu/\bar{\nu}$ -propano/freon	$\bar{\nu}(1\pi^0)$	139	
BNL-Counter	1977	$\nu/\bar{\nu}$ -Al/C	$\nu(\pi^0)$	204	[53]
	1977	$\nu/\bar{\nu}$ -Al/C	$\bar{\nu}(\pi^0)$	22	
ANL-12	1974	ν -D ₂ / ν -H ₂	$\nu p \rightarrow \nu n \pi^+$	8	[54]
	1974	ν -D ₂ / ν -H ₂	$\nu p \rightarrow \nu p \pi^0$	18	
ANL-12	1980	ν -D ₂	$\nu n \rightarrow \nu p \pi^-$?	[55]
ANL-12	1981	ν -D ₂	$\nu n \rightarrow \nu p \pi^-$?	[56]
	1981	ν -D ₂	$\nu p \rightarrow \nu p \pi^0$	8	
	1981	ν -D ₂	$\nu p \rightarrow \nu p \pi^+$	22	
BNL-7	1981	ν -D ₂	$\nu n \rightarrow \nu p \pi^-$	200	[57]

Tabela 3.4: Medidas atuais de corrente neutra.

O detector MINER ν A está completo e a tomada de dados começou em março de 2010 e terá uma duração prevista de quatro anos. Neste período o experimento deverá obter cerca de 14,5 milhões de eventos de corrente carregada (CC). A tabela 3.5 apresenta o número esperado de eventos em diversos tópicos.

O experimento MINER ν A usará, de preferência, a configuração LE do feixe NuMI por cerca de dois anos até Março de 2012 junto com o experimento MINOS e depois do shutdown de 2012 usará a configuração ME junto com o experimento NO ν A.

O conteúdo de ν_e no feixe NuMI é estimado ser da ordem de apenas 1% do fluxo. Uma

Processo	Total	CC/t	CN/t
Quase-elástico	0,8 M	0,27 M	0,09 M
Produção ressonante	1,7 M	0,53 M	0,16 M
Transição: Ressonância para DIS	2,1 M	0,67 M	0,21 M
DIS, Funções de Estrutura e PDF em alto x	4,3 M	1,37 M	0,4 M
Produção coerente de π	0,13 M	0,09 M	0,04 M
Produção de hádron charmosos e estranhos	>240 k		
Distribuição de Parton generalizada	na ordem de 10 k eventos		
Efeitos nucleares	He: 0,6 M, C: 0,4 M, Fe: 2,0 M e Pb: 2,5 M		

Tabela 3.5: Número de eventos em diferentes tópicos (valores aproximados)

importante função do MINER ν A será fornecer uma medida do fluxo e espectro de energia de ν_e no feixe NuMI mais acurada do que a que é possível ser obtida pelo experimento MINOS.

O MINER ν A permitirá medidas mais precisas da seção de choque em baixas energias. Para calcular a seção de choque total temos que considerar as contribuições; $\sigma_{TOT} = \sigma_{QE} + \sigma_{RES} + \sigma_{DIS}$ onde as reações que contribuem em cada seção de choque são :

$$\sigma_{QE}: \text{quase-elástico} \rightarrow \nu(\bar{\nu})n(p) \rightarrow \mu^-(\mu^+)p(n)$$

$$\sigma_{RES}: \text{ressonância} \rightarrow \nu N \rightarrow \mu N^*$$

$$\sigma_{DIS}: \text{espalhamento profundamente inelástico} \rightarrow \nu N \rightarrow \mu X$$

Com os eventos quase elásticos tem-se a perspectiva de calcular, com maior precisão, $\sigma(E_\nu)$ e $d\sigma/dQ^2$, que são importantes para o estudo da oscilação de neutrinos, a determinação do fator forma do vetor axial F_A particularmente em altas Q^2 , e o estudo do espalhamento próton intra nuclear e a dependência com o número de massa A (alvos de C, Fe, e Pb).

A alta granularidade do detector e sua capacidade de rastreamento permitem a identificação de eventos com diversas topologias. A figura 3.26 ilustra a capacidade de identificação de eventos pelo detector. Nesta figura observamos três topologias diferentes e como elas se diferenciam no detector. Em 3.15 apresentamos diversos eventos coletados em tomada de dados que exemplificam esta capacidade do MINER ν A .

Um experimento de espalhamento de neutrinos no feixe NuMI é uma oportunidade excepcional para o estudo de vários tópicos de física com precisão até agora não obtida. Muitos destes tópicos não foram estudados de forma sistemática e outros apresentam poucos resultados com grandes erros estatísticos e sistemáticos. Alguns dos assuntos possíveis de serem estudados pelo experimento MINER ν A são:

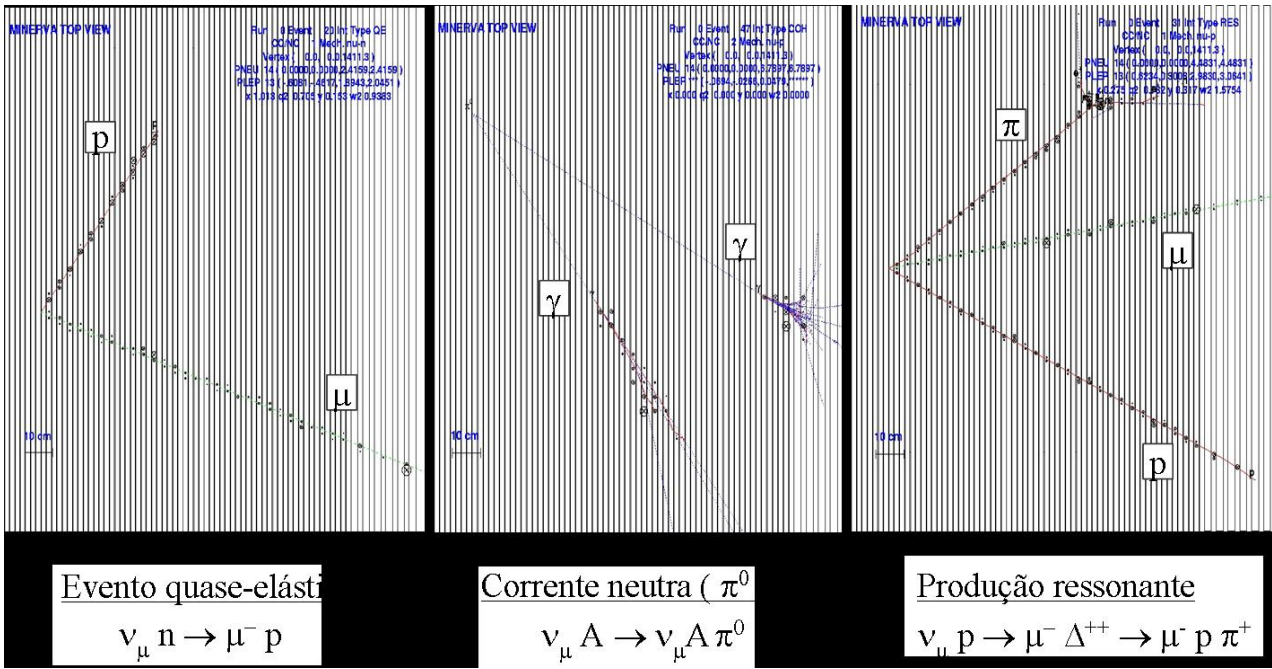


Figura 3.26: Eventos com topologias diferentes podem ser distinguidos graças à capacidade de reconstrução do detector do MINER ν A. Resultados obtidos com simulação.

- Medida precisa da seção de choque quase elástica neutrino-núcleon, incluindo a dependência em E_{ν} e em q^2 .
- Determinação da seção de choque de produção de píons (simples e dupla) para interações em corrente carregada e corrente neutra.
- Medida precisa da produção coerente de píons com particular atenção à dependência com o número atômico A do núcleo.
- Exame dos efeitos nucleares nas interações de neutrinos.
- Estudos da física nuclear para a qual reações de neutrinos fornecem informações complementares aos estudos realizados na mesma faixa cinemática.

Estes itens são, por si só, importante tópicos de pesquisas. Além disto, são essenciais para a minimização dos erros sistemáticos nos experimentos de oscilação de neutrinos.

A figura 3.27 mostra as medidas de seção de choque realizadas até o momento e a faixa de energia de diversos experimentos. Note-se a abrangência da região de energia acessível ao MINER ν A.

Os dados mostrados na figura 3.27 são de experimentos das décadas de 70 e 80 do século passado, com baixa estatística e altos erros sistemáticos. O experimento MINER ν A reduzirá as incertezas destas medidas como ilustrado na tabela 3.6.

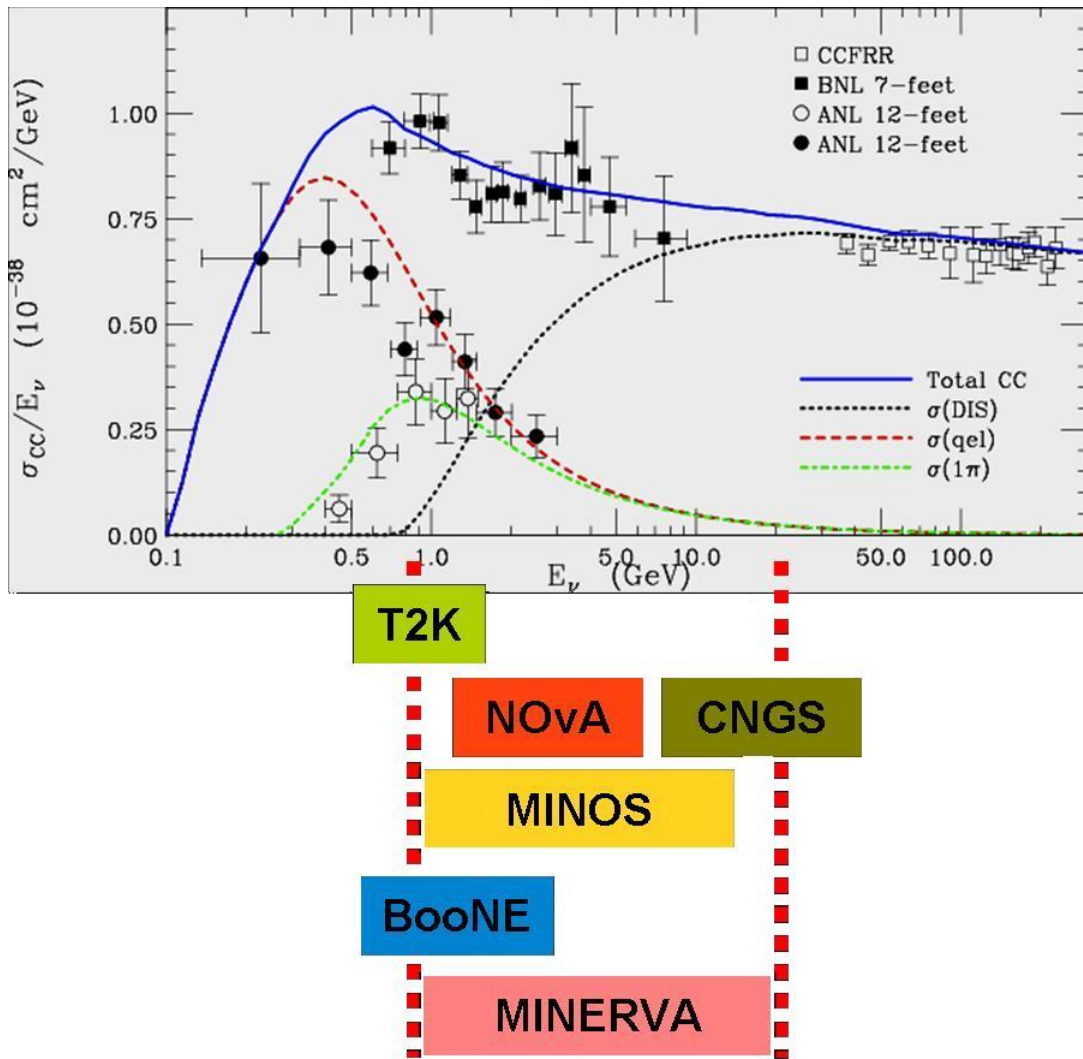


Figura 3.27: Medidas atuais de seção de choque para vários processos. Mostramos a região de energia acessível a diversos experimentos atuais e futuros. Observa-se a faixa de energia de atuação do MINER ν A. Gráfico extraído de [45]

3.14.1 Espalhamento quase-elástico

Espalhamento quase-elástico domina as interações ν -A na região de $E_\nu > 2$ GeV. A medida precisa da seção de choque para esta reação, incluindo a dependência com a energia e com a variação do alvo nuclear, é fundamental para os experimentos sobre oscilação de neutrinos.

O gráfico da parte superior da figura 3.28 sumariza o conhecimento atual da seção de choque quase-elástica. Este gráfico mostra que as medidas atuais tem erros muito grandes em toda a faixa de energias do MINER ν A. O gráfico inferior da figura mostra as medidas esperadas pelo MINER ν A.

Incertezas estimadas para a seção de choque		
Processo	Atual	Após MINER ν A
Quase-elástico	20 %	5 %
Produção ressonante	40 %	5/10 % (CC/CN)
DIS	20 %	5 %
Produção coerente de π	100 %	20 %

Tabela 3.6: Incertezas nas medidas das seções de choque atuais e as esperadas após o experimento MINER ν A.

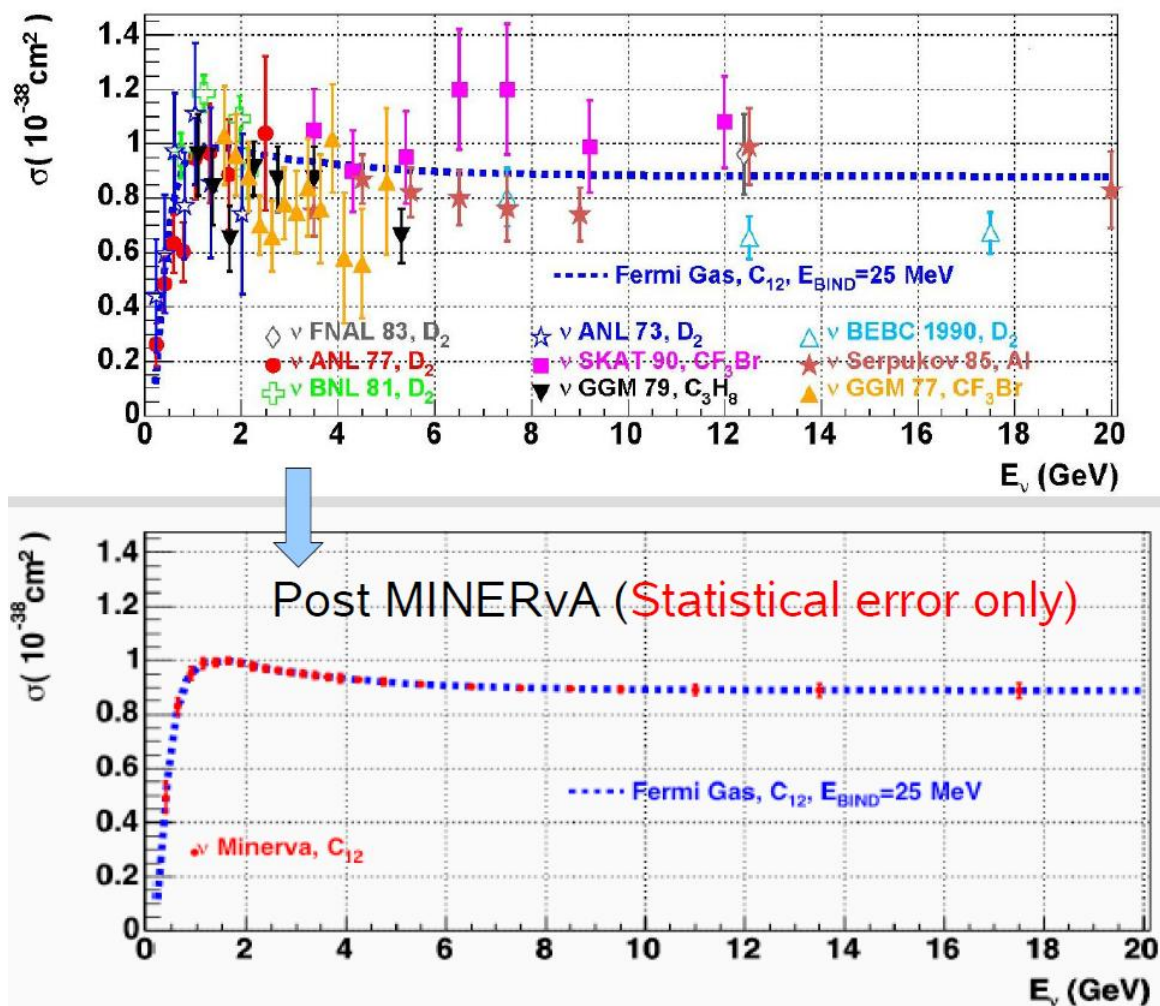


Figura 3.28: superior: medidas atuais da seção de choque quase-elástica. A linha pontilhada em azul emprega um modelo de gás de Fermi. inferior: Previsão dos resultados de medida da seção de choque quase-elástica esperada pelo MINER ν A (apenas erros estatísticos).

3.14.2 Produção ressonante de píons

A grande estatística, variedade de alvos nucleares, baixo patamar de detecção e excelente sistema de rastreamento e calorimetria permitirão que o experimento MINER ν A faça melhorias

significativas nas medidas da produção ressonante de píons em interações de neutrinos. Estas medidas imporão limites às seções de choque, seções de choque relativas, à forma das distribuições de Q^2 e permitirão as primeiras comparações diretas das interações de neutrinos em alvos diferentes.

3.14.3 Produção coerente de píons

As altas taxas do experimento MINER ν A, os diversos alvos nucleares, a alta granularidade do detector, a grande capacidade de reconstrução de trajetórias e a boa calorimetria do detector tornam possíveis o estudo de espalhamento coerente neutrino-núcleo em corrente carregada e em corrente neutra com precisão até então inatingível.

A figura 3.29 mostra os resultados atuais para produção coerente de píons junto com os resultados esperados pelo MINER ν A.

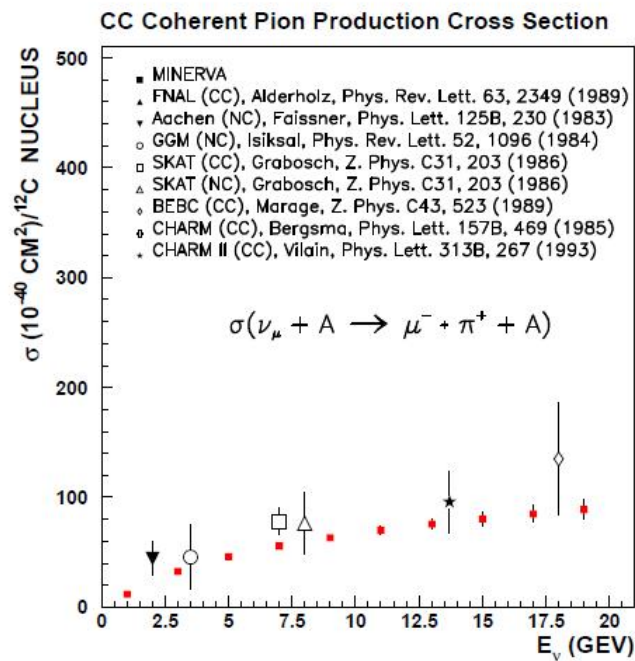


Figura 3.29: Comparação entre os resultados existentes para produção coerente de píons e os resultados esperados pelo experimento MINER ν A.

3.14.4 Dependência em A da seção de choque coerente

MINER ν A comparará as taxas de produção de seção de choque coerente para o carbono e para o chumbo. O cálculo da dependência em A da seção de choque depende, principalmente, do modelo empregado para as interações hádron-núcleo. A medida desta dependência serve,

portanto, como teste destes modelos. Esta comparação ainda não foi realizada. O modelo de Rein- Seghal e o modelo de Paschos, por exemplo, prevêm os valores de 0,223 e 0,259, respectivamente, para a razão entre a seção de choque de produção em corrente neutra ν -C e ν -Pb. A figura 3.30 mostra a dependência em A prevista por ambos os modelos. O experimento MINER ν A pode resolver as diferenças entre os modelos.

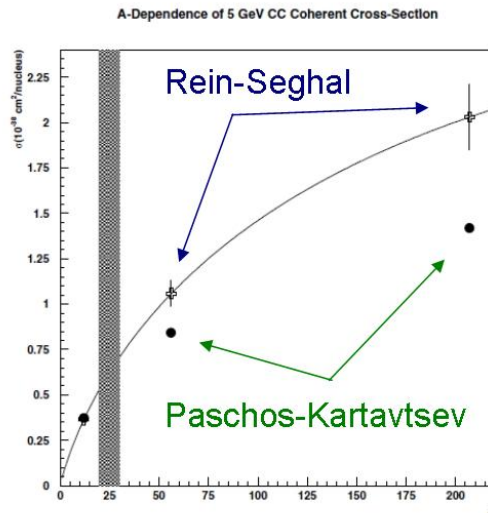


Figura 3.30: Seção de choque coerente como função de A. A região sombreada indica a faixa de medidas anteriores. As cruzes são as previsões do modelo de Rein-Sehgal para espalhamento por C, Fe e Pb e os círculos são as previsões do modelo de Paschos-Kartavtsev. O experimento MINER ν A pode resolver as diferenças entre os diversos modelos.

3.14.5 Produção de partículas estranhas e charmosas

O experimento MINER ν A realizará os primeiros estudos com alta estatística da produção exclusiva de partículas estranhas por neutrinos. A tabela 3.7 sumariza o tamanho da amostra esperada em diversos canais durante 4 anos de tomada de dados. As seções de choque obtidas terão impacto em outras áreas da física de partículas. O programa de física do MINER ν A inclui, também, a busca de novos processos como reações de mudança de estranheza em reações de corrente neutra e ressonâncias bariônicas incomuns. Uma tomada de dados prolongada do feixe NuMi com antineutrinos facilitará estudos de produção de híperons Λ , Σ e Y^* além de permitir novas medidas dos elementos da matriz Cabibbo Kobayashi Maskawa.

Um maior detalhamento pode ser encontrado em [46].

Tipo de reação	Canal exclusivo	N° de eventos
$\Delta S = 0$ CC	$\nu_{\mu}n \rightarrow \mu^{-}K^{+}\Lambda^0$	23.100
	$\nu_{\mu}n \rightarrow \mu^{-}\pi^0K^{+}\Lambda^0$	20.400
	$\nu_{\mu}n \rightarrow \mu^{-}\pi^{+}K^0\Lambda^0$	13.800
	$\nu_{\mu}n \rightarrow \mu^{-}K^{-}K^{+}p$	11.200
	$\nu_{\mu}p \rightarrow \mu^{-}K^0K^{+}\pi^0p$	3.300
$\Delta S = 1$ CC	$\nu_{\mu}p \rightarrow \mu^{-}K^{+}p$	34.900
	$\nu_{\mu}n \rightarrow \mu^{-}K^0p$	5.200
	$\nu_{\mu}n \rightarrow \mu^{-}\pi^{+}K^0$	4.600
$\Delta S = 0$ NC	$\nu_{\mu}p \rightarrow \nu K^{+}\Lambda^0$	7.900
	$\nu_{\mu}n \rightarrow \nu K^0\Lambda^0$	2.400
	$\nu_{\mu}n \rightarrow \nu K^0\Lambda^0$	6.100

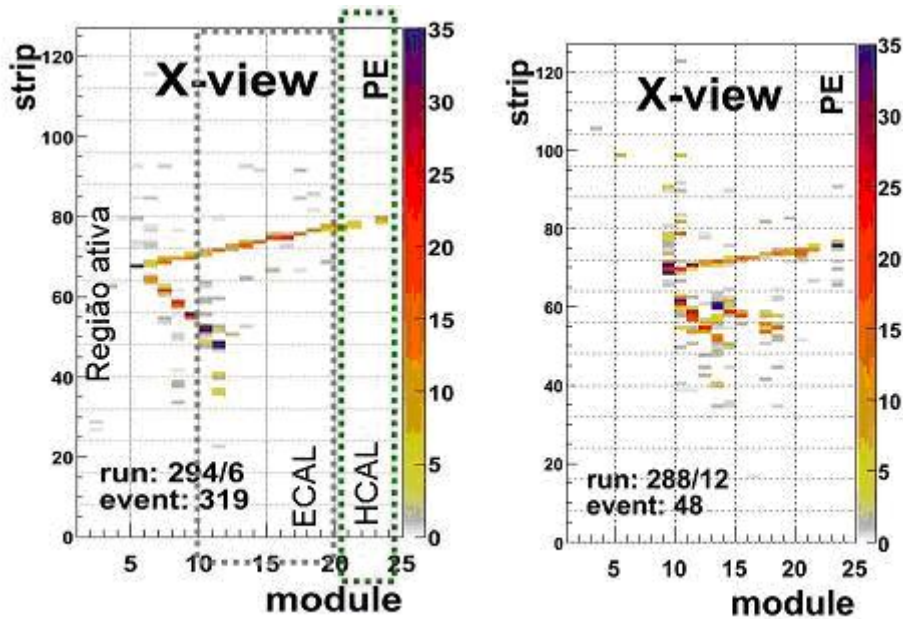
Tabela 3.7: Amostras esperadas pelo experimento MINER ν A para diversos eventos de produção exclusiva de partículas estranha durante uma tomada de dados de 4 anos

3.15 Tomada de Dados

Em abril de 2009 o experimento MINER ν A iniciou uma fase de tomada de dados com o detector de teste, que consiste de 24 módulos do detector. Esta tomada de dados empregou o feixe NuMI e serviu para teste de todo o *hardwaree software* do experimento.

A figura 3.31 apresenta dois eventos reais, obtidos durante a tomada de dados de teste. A figura à esquerda mostra um evento quase-elástico onde observamos a ocorrência de um múon (a linha que avança além dos limites do detector) e um próton (a linha que avança por uma certa distância dentro do detector até ser completamente absorvida). A figura da direita apresenta um evento com ocorrência de um π^0 (observe os dois chuveiros apontando para um vértice, que são originários dos dois fótons resultantes do decaimento do π^0). Em ambas as figuras observamos que o vértice primário ocorre já bem dentro do detector (o feixe de neutrinos penetra no detector vindo pela esquerda). Na figura da esquerda observamos um evento quase-elástico (a linha que se prolonga para além do detector é consistente com um múon e a linha curta é consistente com um próton. Nota-se o aumento na energia depositada à medida que o suposto próton atravessa o detector). Na figura da direita observamos a capacidade do detector do MINER ν A identificar π^0 (os dois chuveiros observados em direção à parte direita inferior da figura são compatíveis com o sinal deixado por dois fótons resultantes do decaimento de um π^0).

À medida que avançava a montagem do detector, mais módulos eram instalados. A tomada de dados de teste ocorria, então, com diversas configurações do detector. Este processo servia para que a montagem pudesse ser sistematicamente testada.



Candidato quase elastico

candidato π^0

Figura 3.31: Dois eventos obtidos com 25 módulos do detector MINER ν A (20% do detector) durante tomada de dados de teste no feixe NuMI.

A figura 3.32 mostra um evento candidato a múon registrado quando o detector possui 64 módulos já instalados (conhecido como *frozen detector*). Observamos uma trajetória que atravessa todo o detector vindo de fora deste. Tal trajetória é compatível com um múon que venha acompanhando a linha de feixe. A figura 3.33 mostra um evento em que uma das partículas resultantes da interação viaja na direção contrária a do feixe. Na figura o feixe de neutrinos vem da esquerda e observamos uma interação ocorrendo dentro do detector. A maioria das partículas resultantes segue na direção frontal.

Em março de 2010 iniciou-se a tomada de dados já com o detector completo. A figura 3.34 mostra um evento candidato a $\nu_\mu + n \rightarrow p\mu^- \pi^0$. Observa-se o chuveiro iniciando-se no calorímetro eletromagnético. Observa-se um traço percorrendo todo o detector (compatível com um μ^-), um traço curto (compatível com um p) e um chuveiro que começa afastado do vértice (notado mais facilmente na vista V) que é compatível com o π^0 que decai em $\gamma\gamma$. Nas figuras o neutrino penetra o detector pela esquerda.

A figura 3.35 mostra outro evento obtido com o detector completo onde observamos várias trajetórias partindo de um mesmo ponto (próximo ao módulo 55) e dois chuveiros partindo de um mesmo ponto por volta do módulo 80 e *strip* 35. Isto é compatível com um π^0 oriundo do vértice e decaindo em $\gamma\gamma$ neste ponto.

Por fim mostramos a figura 3.36 na qual observamos claramente dois chuveiros que se

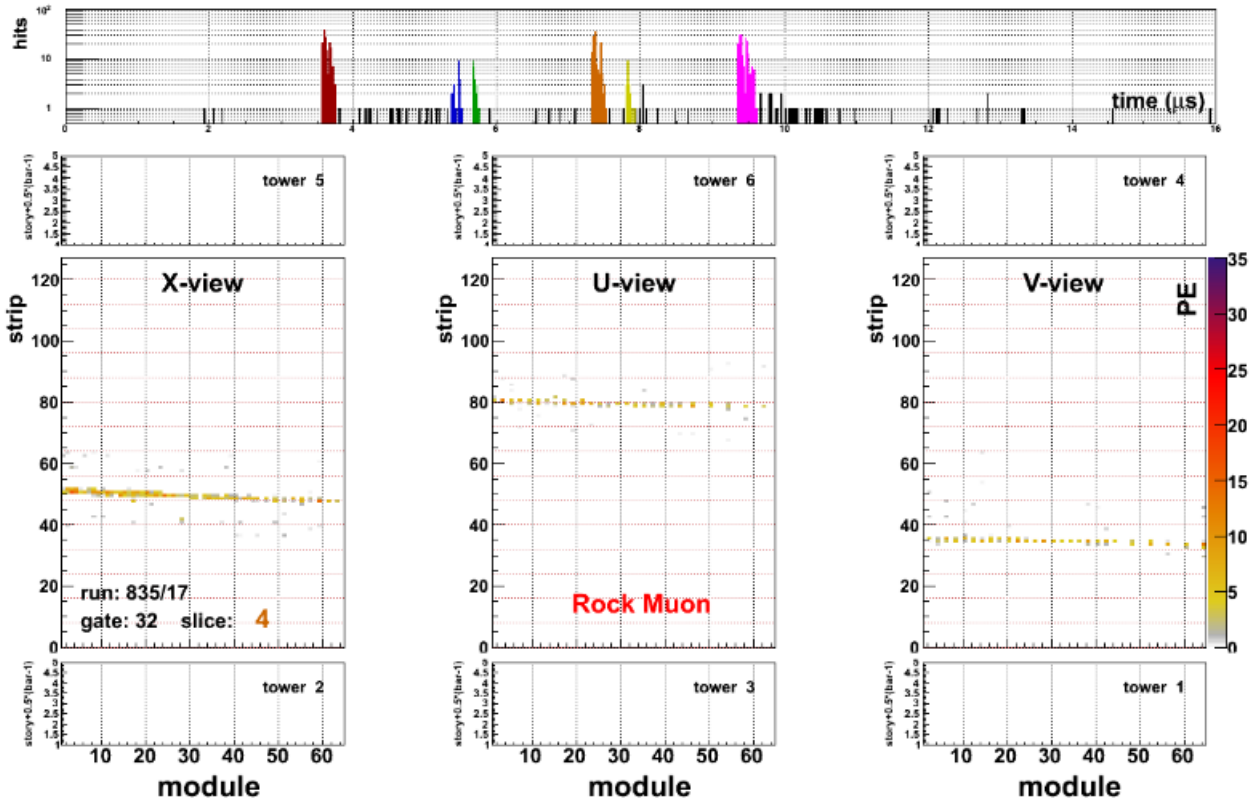


Figura 3.32: Candidato de múons. A trajetória observada atravessa todo o detector e origina-se de fora do detector (vindo da esquerda). É um sinal compatível com o que seria deixado no detector por um múon viajando na mesma direção do feixe. Evento observado com um detector com 64 módulos.

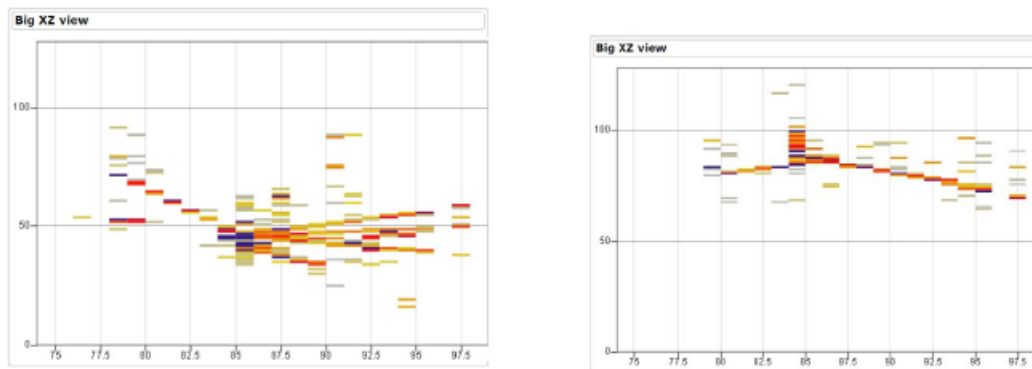


Figura 3.33: Um evento onde uma das partículas resultantes viaja na direção contrária ao feixe. O neutrino penetra no detector vindo pela esquerda e interage no centro do detector (o ponto de onde vemos brotar várias trajetórias).

desenvolvem dentro do detector.

O detector do experimento MINER ν A está completamente montado e a tomada de

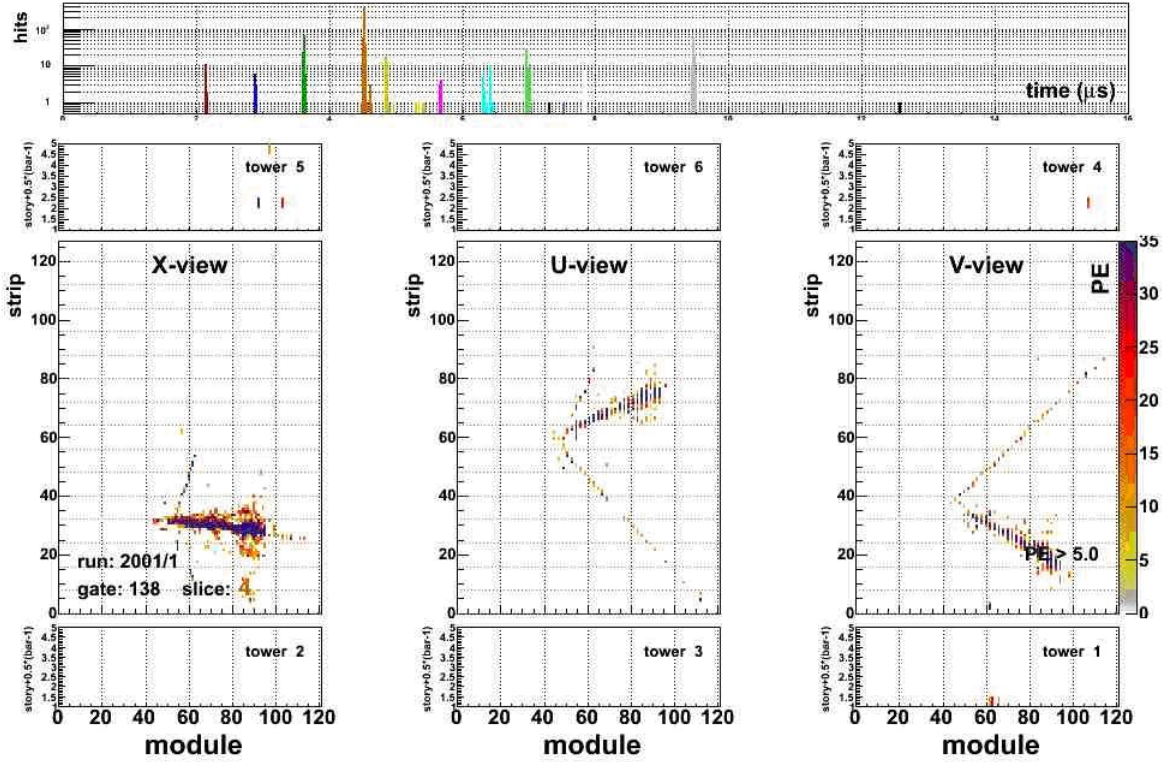


Figura 3.34: Evento obtido na tomada de dados com o detector completo do MINER ν A candidato a $\nu_\mu + n \rightarrow p\mu^- \pi^0$.

dados está em andamento. Todos os sistemas funcionam dentro dos parâmetros planejados e esperados. A caracterização completa da operação e resposta do detector encontra-se em andamento juntamente com o desenvolvimento dos *softwares* de reconstrução e análise. A boa caracterização do detector depende, em grande parte, do emprego do detector de teste, como descrito em 3.11.3. Nos capítulos 4 e 5 descrevemos nosso trabalho na montagem e teste do feixe empregado com este detector, o que constitui a parte principal desta dissertação.

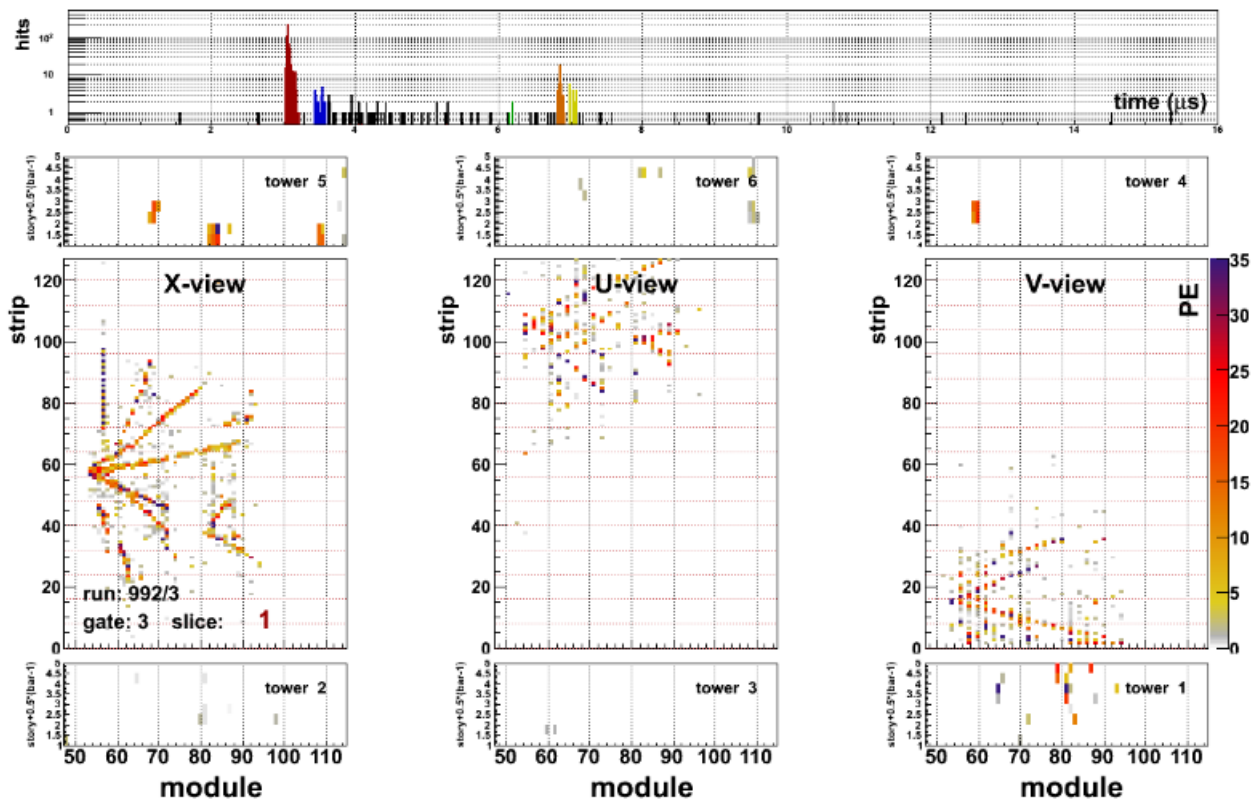


Figura 3.35: Candidato π_0

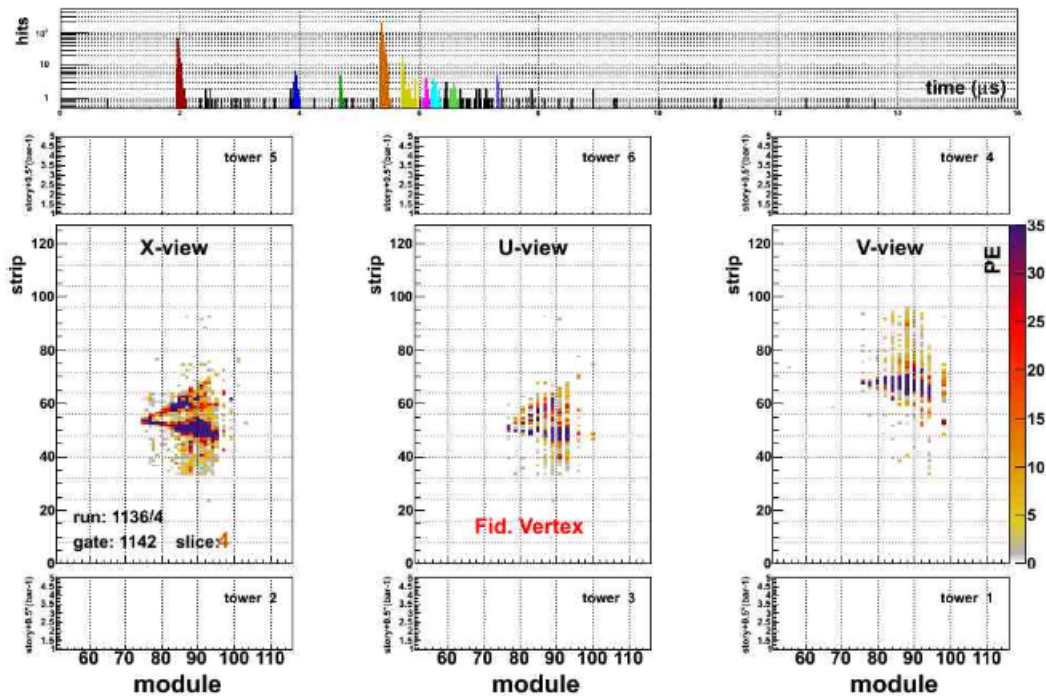


Figura 3.36: Chuveiro electromagnético

Capítulo 4

O feixe de Teste do MINER ν A

4.1 Introdução

Conforme descrito em 3.11.3 um detector do MINER ν A em menor escala é empregado para calibração e caracterização da resposta do detector do MINER ν A . O uso deste detector de prova requer a montagem de um experimento completo empregando um feixe de hádrons. Este experimento permite-nos medir a resposta do detector e o funcionamento do sistema de fonte de aquisição de dados (DAQ).

O detector de prova tem dimensões de 1m x 1 m e é composto de 40 módulos removíveis com Fe e Pb com as mesmas características do *Inner Detector* (ID) . Uma simulação empregando o *software* NUGEN mostra que as distribuições de energia de hádrons provenientes de eventos quase elásticos (QEL), ressonantes e espalhamento inelástico profundo (DIS) têm um máximo ao redor de 0,5 GeV para píons e prótons conforme mostrado na figura 4.1. Assim, é necessário que se estabeleça um feixe de hádrons, conhecido como feixe de teste (*test beam* TB), que atinja estas energias.

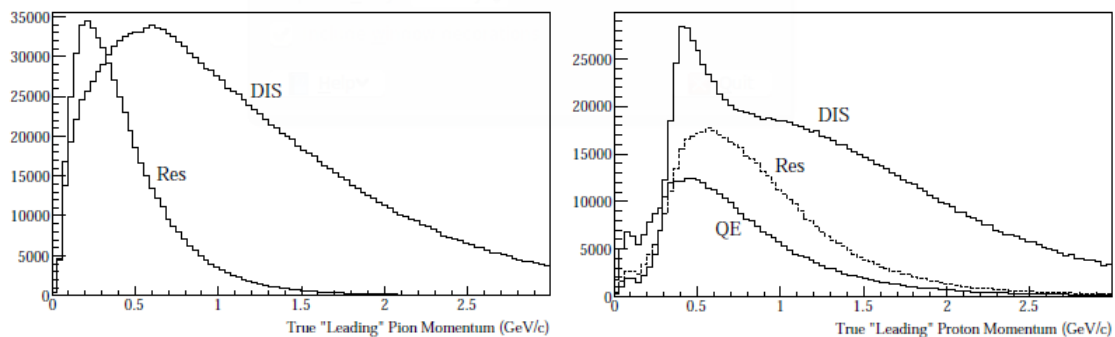


Figura 4.1: Píons e prótons esperados. Simulação de MC usando NueGen.

O feixe de teste TB é gerado a partir de um feixe de mésons. Estes feixe, por sua vez,

é produzido a partir de um feixe de prótons de 120 GeV. O processo emprega os seguintes aceleradores.

- **Cockroft-Walton:** neste acelerador, átomos de hidrogênio são ionizados pela adição de elétrons (formando íons de H^-) e acelerados eletrostaticamente até a energia de 750 keV.
- **LINAC:** ao sair do Cockroft-Walton os íons negativos entram em um acelerador linear de aproximadamente 150 m de comprimento conhecido como LINAC (*Linear Accelerator*) onde são acelerados até 400 MeV. No LINAC a aceleração é feita por uma série de cavidades de RF que definem, também, a estrutura temporal do feixe. Ao final do LINAC os íons passam por uma fina lâmina de chumbo que remove os elétrons permitindo que apenas os prótons continuem.
- **Booster:** os prótons provenientes do LINAC entram em um pequeno síncrotron conhecido como *Booster*. O Booster acelera os pacotes de prótons¹ até a energia de 8 GeV.
- **Injetor Principal MI:** o MI é um síncrotron cerca de sete vezes maior que o Booster que recebe os pacotes de prótons a 8 GeV e os acelera até 120 GeV. O MI também gerencia a injeção de p no Tevatron, o principal acelerador do FERMILAB².
- **Distribuidor (*switchyard*):** parte dos prótons são enviados pelo MI para um distribuidor que os distribui para duas linhas de feixe voltadas a experimentos de alvo fixo: a linha de feixe NuMI (onde é produzido o feixe de neutrinos utilizado pelo experimento MINER ν A e que é descrito em 3.2) e a linha de feixe de mésons Meson Beam (MB).

4.2 O feixe de mésons MB - Feixe Secundario

Os prótons de 120 GeV do *Main Injector* (MI) colidem com um alvo de Al de 30 cm de espessura. As partículas resultantes são desviadas por um magneto que faz a seleção de momenta e produz o MB. Há 5 áreas experimentais onde este feixe pode ser usado. Uma delas é a área de teste de mésons (*Meson Test Hall*, (MTest), onde experimentos diversos podem estudar a resposta de seus detectores a um feixe monoenergético de hádrons. Um novo feixe hadrônico de baixa energia, feixe terciário, ajustado para a produção de energias na faixa de centenas de MeV, foi estabelecido no MTest para teste do detector do experimento MINER ν A . A figura 4.2 mostra uma planta baixa da área experimental destacando a área empregada para o teste do

¹Circulam 84 pacotes de prótons espaçados por 19 ns

²O MI também gerencia a produção de \bar{p} , que são mantidos em um anel de acumulação, e a sua posterior injeção do Tevatron

MINER ν A . O comissionamento e análise de dados deste novo feixe é o meu trabalho principal no experimento e é apresentado nesta dissertação

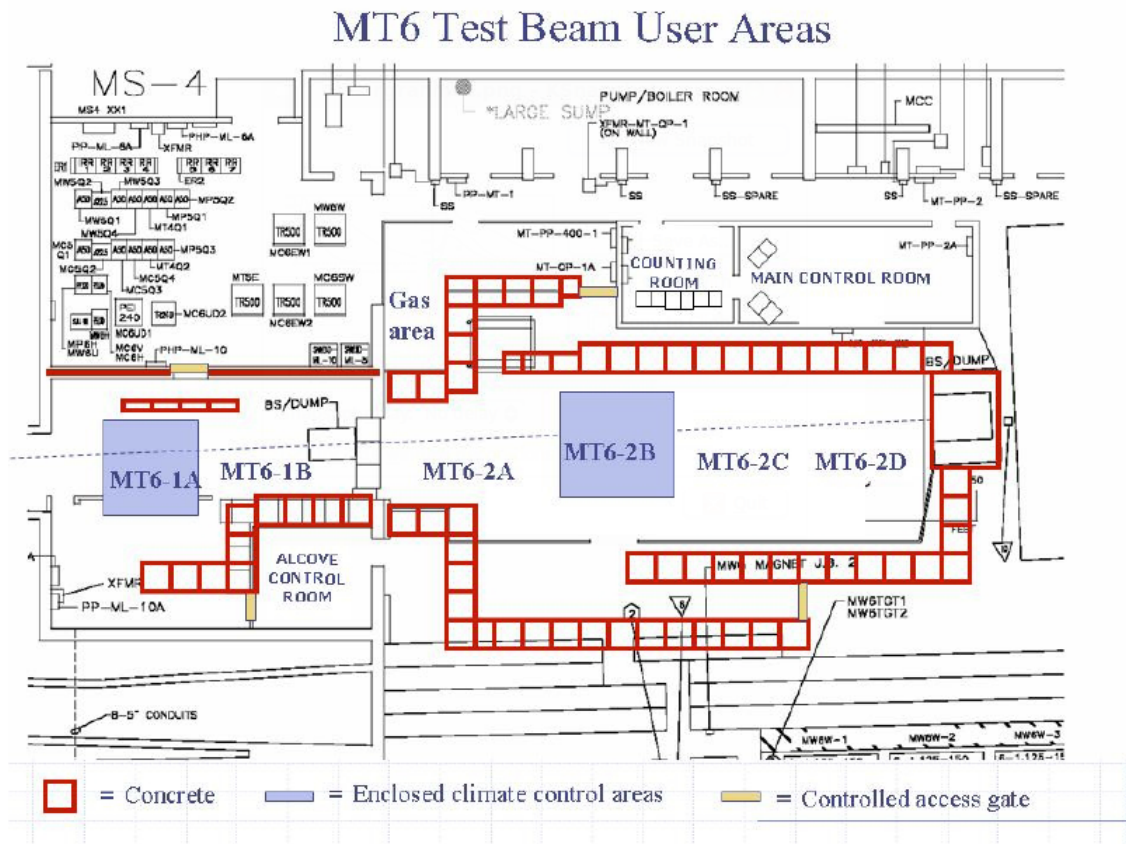


Figura 4.2: Área de teste de múons no Fermilab mostrando as áreas experimentais utilizadas para o feixe de teste do MINER ν A .

4.3 O Feixe Terciário

O feixe terciário deve atender às seguintes características:

- Fornecer π com momentum entre 300 MeV/c e 1.500 MeV/c.
- Ter uma taxa de eventos compatível com um DAQ lento, como o do MINER ν A .
- Informar ao DAQ as características específicas da partícula.

Durante meu trabalho estabelecemos um feixe misto para satisfazer todos estes requisitos³.

³Embora o desejado fosse um feixe exclusivamente de π , na prática o feixe produzido possui também e^+ , e^- e p que são tratados como contaminação.

Após a colisão do feixe secundário (prótons de 120 GeV/c) com o alvo de Al os prótons remanescentes são absorvidos de forma que nenhum dispositivo sensível ao longo do detector seja exposto aos prótons de 120 GeV/c. Um feixe resultante de píons de 16 GeV/c é enviado para área MT6-2A onde nosso feixe de teste é produzido⁴. A figura 4.3 mostra a estrutura da linha de feixe terciário obtido a partir de um feixe de píons de 16 GeV/c e a figura 4.4 mostra a linha de feixe de teste como montada para os testes do detector do MINER ν A .

Os componentes da linha de feixe que instalamos são:

- Um alvo de cobre;
- Um colimador de aço;
- Dois dipolos magnéticos;
- Quatro câmaras de fios: duas antes dos magnetos e duas depois;
- Oito hodoscópios : dois após o colomimador e seis depois da quarta câmara de fios.

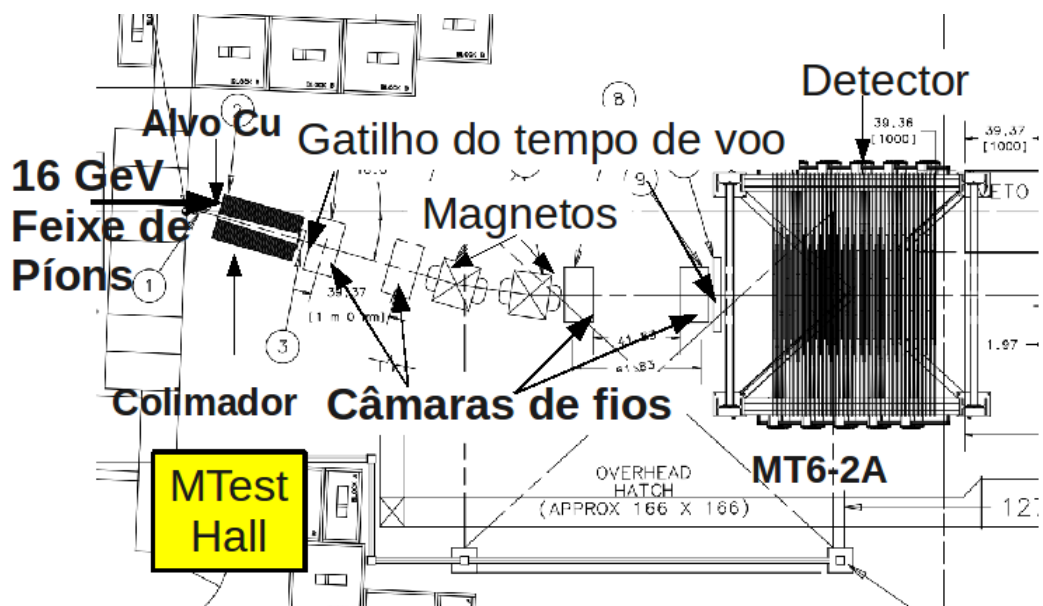


Figura 4.3: Área de teste de múons no Fermilab mostrando os componentes da linha de feixe para o detector de teste MINER ν A .

⁴Vários testes, descritos em 5.3, foram realizados para decidirmos por um feixe de 16 GeV/c.

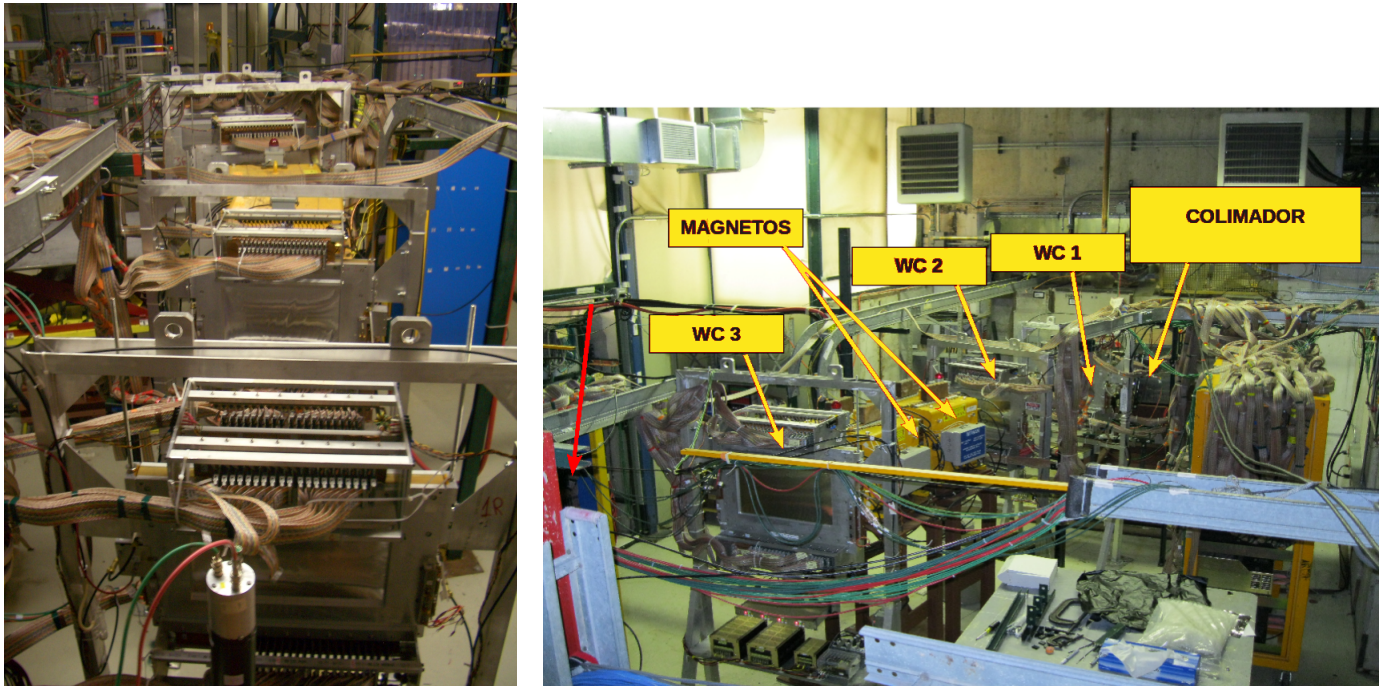


Figura 4.4: Vistas da linha de feixe de teste.

4.3.1 O Alvo de Cobre

O cobre foi escolhido como alvo devido à sua compacidade e baixa atividade. Pares e^- e e^+ energéticos vêm de chuveiros electromagnéticos originados de um γ . Os π e os prótons vêm de excitação nuclear.

Análise realizada em [58] mostra que um alvo de cobre com $1,4 \lambda_I$, correspondendo a uma espessura de 21,1 cm, é adequado para produção do feixe exigido.

4.3.2 O Colimador de Aço

Um colimador de aço posicionado logo após o alvo de Cu, conforme mostrado na figura 4.3, seleciona as partículas que saem a um ângulo de 16 graus⁵ com o feixe de π .

A figura 4.5 mostra o espectro de partículas esperado na saída do colimador para um feixe de π de 16 GeV/c no alvo de Cu obtido a partir de simulação. A figura mostra que o feixe terciário apresenta preponderância de π com pico ao redor de 450 MeV/c. Para energias abaixo de 200 MeV/c prevalecem e^+ e e^- . Prótons representam cerca de um quinto dos π . A geometria e projeto da linha de feixe fazem com que muito poucos e^+ , e^- e p cheguem ao detector de teste.

⁵Este valor é escolhido para atender as restrições de espaço e ao poder de deflexão dos magnetos.

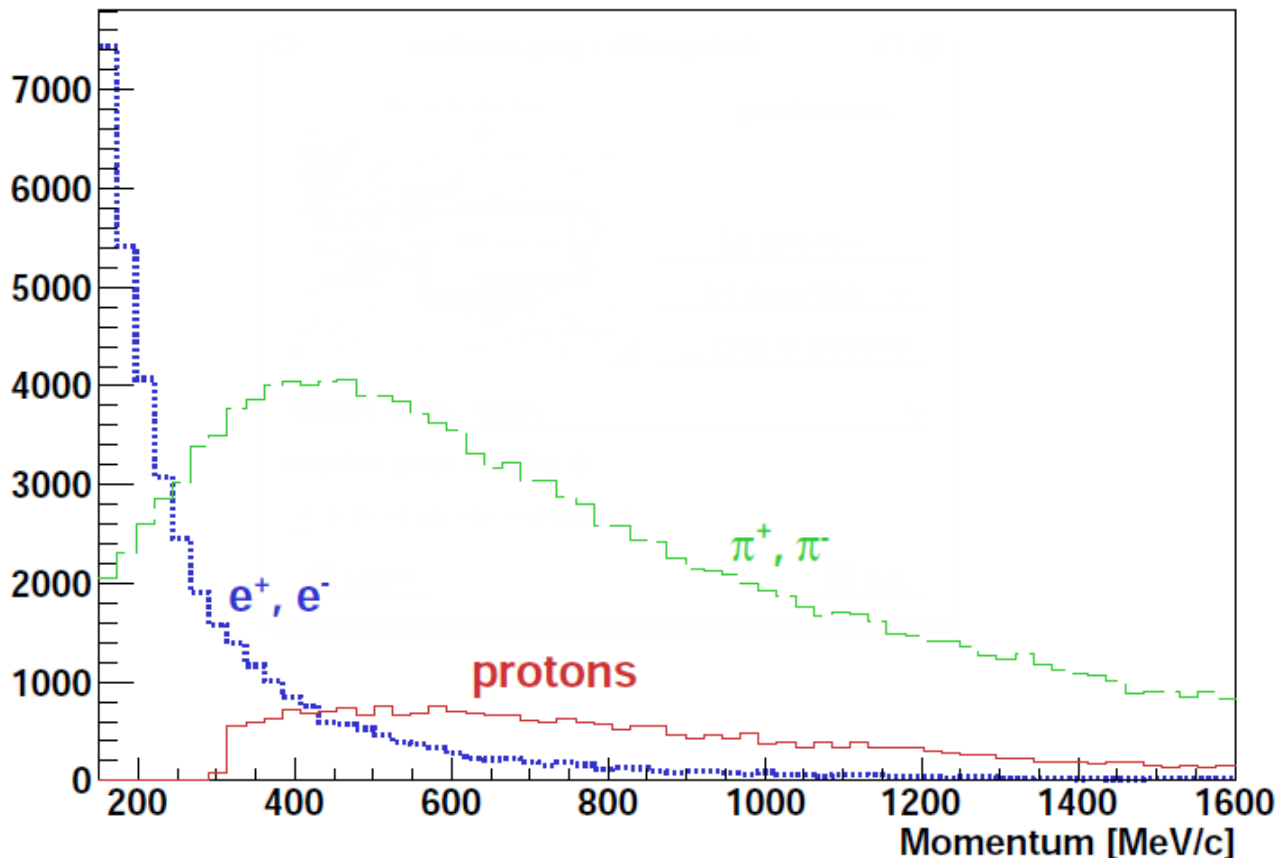


Figura 4.5: Taxa esperada de partículas na saída do colimador para 15 milhões de π no alvo de Cu. Valores obtidos a partir de simulação.

4.3.3 Os Magnetos

Quando o feixe sai do colimador contém várias espécies de partículas com *momenta* diversos, como mostrado na figura 4.5. Assim, os elementos da linha de feixe que se seguem ao colimador precisam desviar o feixe permitindo que o usuário faça uma seleção de partículas e momento. Isto é conseguido com o emprego de um sistema de dipolos magnéticos, para a seleção de momento, e um sistema do tempo de voo para a identificação das partículas.

Empregamos uma dupla de magnetos⁶ dipolares refrigerados a ar com abertura de cerca de 30 cm por 15 cm. A figura 4.6 mostra um dos magnetos. Uma simulação realizada empregando o programa SUPERFISH mostra que os magnetos podem desenvolver um campo magnético constante de 3,4 kGauss com corrente de 100 A por um longo período sem superaquecimento. Isto foi posteriormente confirmado pelo sistema ACNET⁷ que permite monitorar variáveis como

⁶Estes magnetos foram usados anteriormente no experimento *debuncher*

⁷Accelerator Control NETWORK é um sistema empregado no FERMILAB para monitoramento *online* de

temperatura e corrente. A figura 4.7 permite uma comparação entre os resultados da simulação e os dados obtidos com o ACNET para a temperatura dos magnetos em um intervalo de tempo de 3 horas.

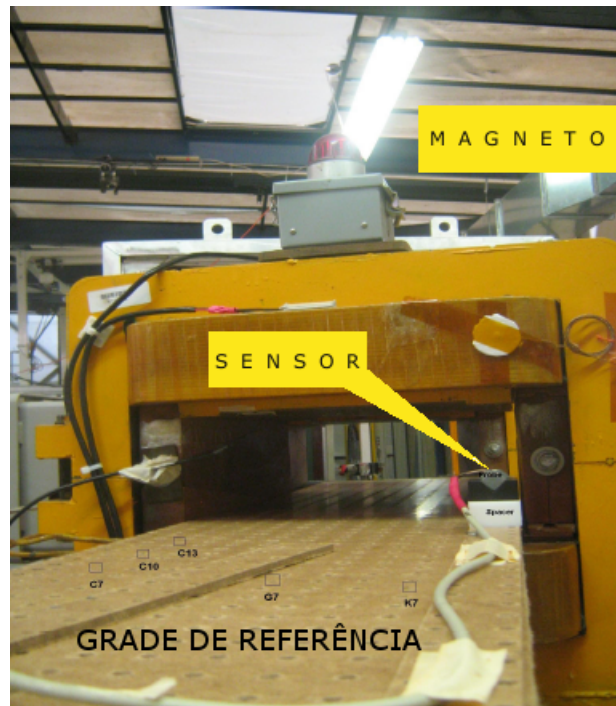


Figura 4.6: Um dos magnetos empregados na linha de feixe. Observa-se o sistema empregado para mapeamento do campo magnético consistindo de uma grade de posições, sensor de prova e espaçador.

Os dois magnetos têm campo de 0,34 T de intensidade cujo mapeamento foi realizado empregando um sensor capaz de medir a intensidade do campo magnético nas 3 dimensões espaciais (B_x, B_y, B_z). Na figura 4.6 observamos o sensor e o sistema de posicionamento e referência empregado. O mapeamento foi realizado varrendo-se todo o interior da abertura dos magnetos pela alteração sucessiva da posição do sensor no sistema de referência. O programa de aquisição de dados foi desenvolvido na plataforma LabView e um código⁸ em Python foi desenvolvido para a análise e tratamento dos dados.

Os magnetos foram alimentados por uma corrente de 100 A durante 5 segundos⁹. Os dados incluem um registro de tempo, a leitura da posição no sistema de referência e o tempo de cada pulso.

diversas variáveis do acelerador e detectores associados.

⁸Vários códigos foram desenvolvidos durante o trabalho descrito nesta dissertação. Os principais códigos podem ser encontrados em B

⁹A corrente consistia de uma rampa de 0 A a 100 A em cerca de 1 s, após o qual mantinha-se estável por 5 segundos repetindo-se a cada minuto

Conversores analógico-digital (ADC) são empregados para leitura dos sinais do sensor dos quais subtrai-se os pedestais obtidos com a realização de medidas na ausência de corrente. Os valores são convertidos para Gauss na razão de 1 Gauss para cada contagem de 209 do ADC¹⁰. O erro na medida é tomado como $\frac{\sigma}{\sqrt{N}}$ onde N é o número de medidas e σ seu desvio padrão.

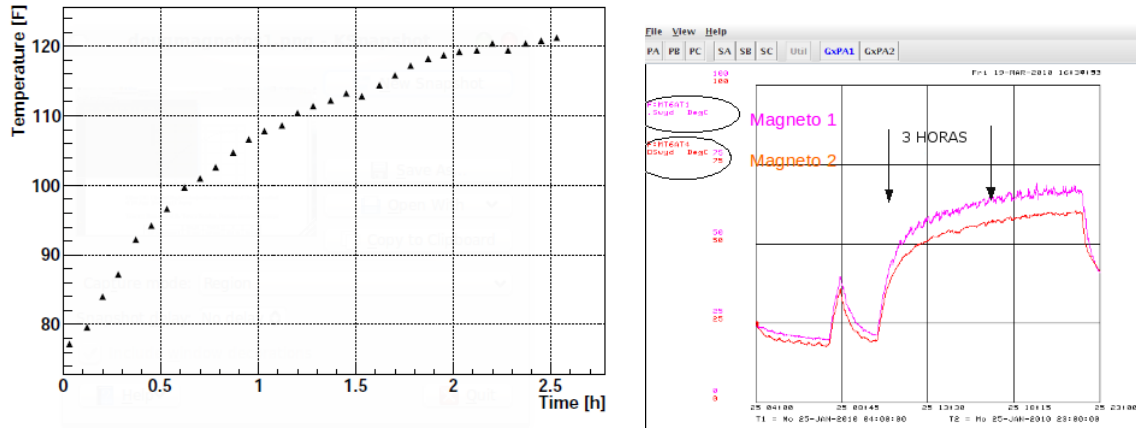


Figura 4.7: Monitoramento da temperatura dos magnetos da linha do feixe. À esquerda resultado obtido com simulação e, à direita, resultado obtido pelo sistema ACNET

Por construção e facilidade, a origem é escolhida como sendo o centro do magneto a ser analisado e tem o eixo z na direção do feixe, o eixo x à direita e o eixo y na vertical apontando para o chão (sistema destrógiro). Neste sistema de coordenadas, a componente das medições do sensor de prova tem uma grande componente principal com valor correspondente a 3.300 Gauss de um campo apontando para baixo, o que implica em desviar para a esquerda (direção x negativo) a trajetória de partículas positivas. Com relação ao método empregado, devemos chamar a atenção para os seguintes pontos:

- Apenas a componente principal do campo magnético (neste caso B_y) está sendo usada pelos programas de reconstrução de trajetórias e de análise.
- Possíveis variações térmicas de componentes dos magnetos não são consideradas.
- Pedacos de metal foram instalados para resfriamento das bobinas, o que pode alterar ligeiramente o campo magnético.

A figura 4.8 mostra resultados para B_z e B_y obtidos em simulações e os medidos.

¹⁰Valores do conversão do equipamento usado

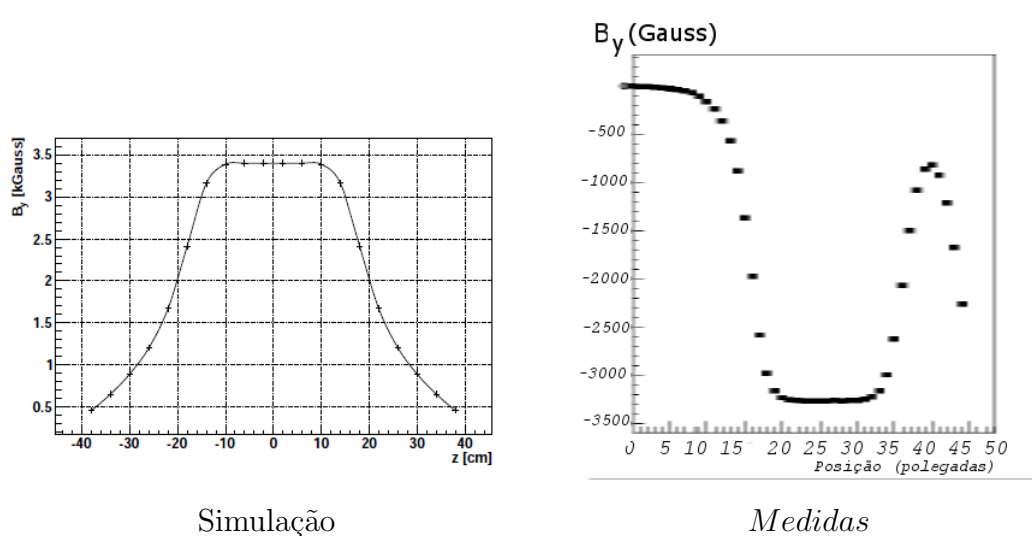


Figura 4.8: Campo magnético B_y em um dos magnetos. À esquerda vemos o resultado de simulações e à direita o campo medido no processo de mapeamento. As polaridades para simulação e medidas são opostas.

4.3.4 Câmaras de fios

Empregamos câmaras de fio originalmente usadas no experimento Hypercp [59] como elementos de rastreamento do feixe. Estas câmaras são formadas por quatro planos (X-U-V-XP) com aproximadamente um milímetro de espaçamento entre fios¹¹. As vistas verticais (X e XP) estão deslocadas de 0,5 mm e os planos U e V fazem um ângulo de $\pm \arctan(1/2)$ com a vertical. Uma resolução de 0,5 mm pode, então, ser obtida. As câmaras tem alimentação típica de 2.200 V e possuem uma atmosfera de 80 % Ar e 20 % CO₂. O sistema de leitura foi desenvolvido e testado, inicialmente, com raios cósmicos para testar cada etapa da eletrônica constituinte das câmaras. A tabela 4.1 apresenta as características principais das câmaras de fios e a figura 4.9 mostra o diagrama do sistema de leitura das câmaras de fio.

Câmaras	Abertura Interna	distância entre os fios	diâmetro dos fios
1	45,72 cm x 25,40 cm	1,016 mm	12,5 μm
2	45,72 cm x 25,40 cm	1,016 mm	12,5 μm
3	55,88 cm x 30,48 cm	1,270 mm	12,5 μm
4	55,88 cm x 30,48 cm	1,270 mm	12,5 μm

Tabela 4.1: Características das câmaras de fios. Cada plano tem 320 canais efetivos.

Antes da montagem da linha de feixe realizamos diversos testes com as câmaras de fio para

¹¹Apenas 320 canais de cada plano foram instrumentados.

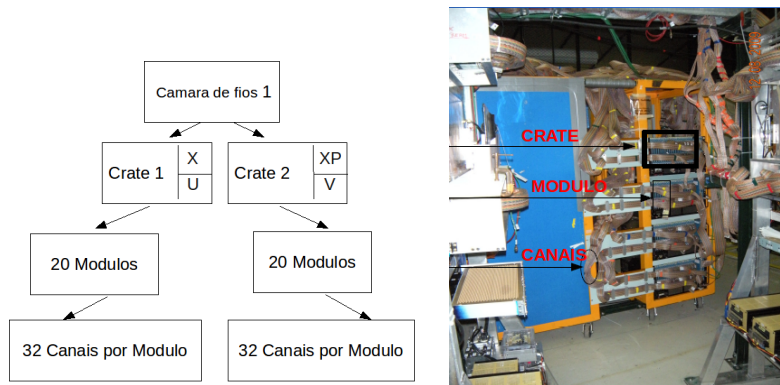


Figura 4.9: Esquerda: Diagrama esquemático do sistema de leitura das câmaras de fios; Direita: eletrônica de leitura das câmaras.

verificar, principalmente:

- canais mortos;
- canais quentes;
- patamar dos discriminadores;
- alta tensão dos fios.

Para a realização destes testes empregamos uma fonte radioativa consistindo de Sr-90, que emite raios β de 0,546 MeV, que era posicionada sobre a superfície de cada câmara em passos de 1 polegada permanecendo 3 minutos em cada posição.

A figura 4.10 apresenta um resultado para o plano XP da câmara de fios número 1 para duas posições da fonte de Sr-90. Na figura, os histogramas na parte de cima apresentam todos os fios da câmara enquanto na parte de baixo destacamos uma região em particular. Observa-se que os canais correspondentes aos fios 51, 53 e 57 estão mortos, não apresentando qualquer sinal.

A figura 4.11 mostra outro tipo de problema que foi observado no plano U da câmara 1. Na figura observamos 16 canais quentes (fios 305 a 320). Os fios das câmaras são agrupados em grupos de 16 e levados aos módulos préamplificadores. Neste caso específico o problema deveu-se a uma conexão defeituosa do cartão que contém os préamplificadores destes canais.

Os testes com a fonte de SR-90 serviram, também, para identificar os fios centrais dos planos V e U. Para este estudo a fonte de Sr-90 é posicionada no centro geométrico da câmara e a posição XY é computada. A figura 4.12 mostra resultados para os planos U e V da câmara 1. No gráfico da esquerda, o fio 230 dos planos U e V é considerado como central. A posição XY obtida é $x = 1,083$ mm e $y = -78,05$ mm. O gráfico da direita mostra que quando o fio

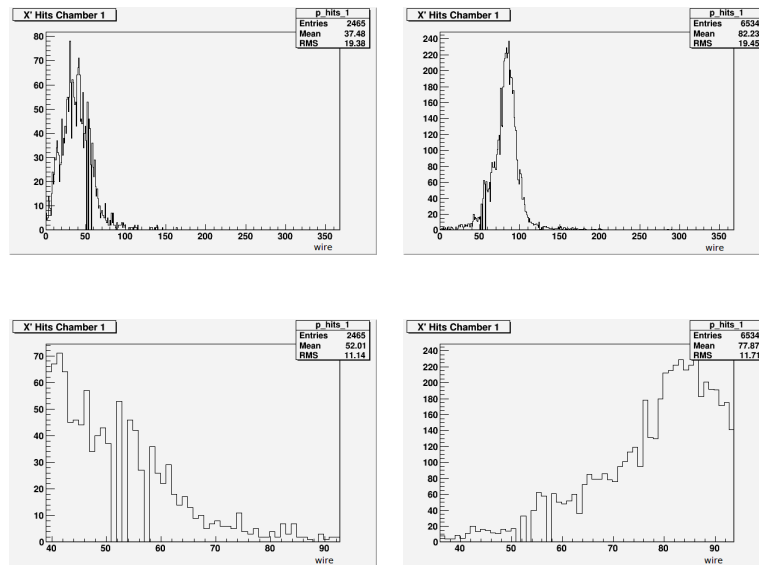


Figura 4.10: Teste com SR-90 do plano XP da câmara 1. As figuras mostram histogramas da atividade nos canais para duas posições diferentes da fonte de raios β . Os detalhes na parte de baixo mostram os canais 51, 53 e 59 mortos.

200 passa a ser considerado como fio central passamos a obter $x = 1,083$ mm e $y = -5,347$ mm. Este trabalho foi realizado para todas as câmaras para determinação dos fios centrais, informação esta que foi incluída nos arquivos de configuração da linha de feixe e necessária para os programas de reconstrução.

4.3.5 Hodoscópios

Os oito hodoscópios consistem de placa cintiladora associada a uma fotomultiplicadora e estão agrupados de maneira a formarem dois conjuntos de medidas separados por cerca de 6,0 m. Em seguida ao colimador temos um par de hodoscópios TOF1-TOF2 arrumados de forma a suas placas cintiladoras se sobreporem. Após a câmara 4 temos os demais seis hodoscópios agrupados em pares TOF3-TOF4, TOF5-TOF6 e TOF7-TOF8 cada um tendo as placas cintiladoras se sobrepondo. Estes três conjuntos são posicionados um acima do outro de forma a constituírem uma parede de hodoscópios com três setores: inferior, médio e superior.

Cada hodoscópio foi testado para vazamento de luz e teve suas altas tensões ajustadas para desempenho eficiente. O conjunto de hodoscópios é empregado para estabelecimento do gatilho e para medida do tempo de voo das partículas do feixe como descrito mais adiante neste trabalho.

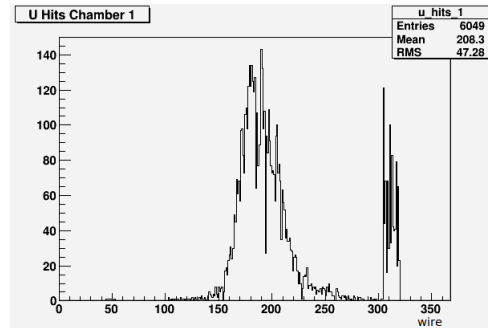


Figura 4.11: A fonte de Sr-90 está posicionada próximo ao centro do plano. Observa-se, entretanto, atividade entre os fios 305 e 320, indicando canais quentes.

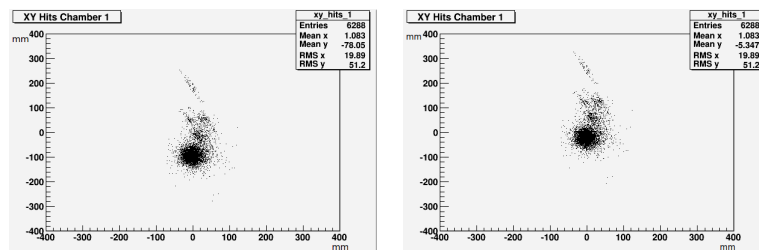


Figura 4.12: Distribuição XY, em milímetros, da associação de planos U e V da câmara 1. O gráfico da esquerda considera o fio 230 como central enquanto no gráfico da direita o fio 200 é tomado como central.

Capítulo 5

Análise do Feixe de Teste

5.1 Geração do Gatilho e Medida de Tempo de voo

Nosso feixe utiliza os sinais dos hodoscópios para geração do gatilho do detector de teste e para identificação das partículas do feixe.

Após discriminação, o sinal das fotomultiplicadoras são enviados para a eletrônica de gatilho e de determinação do tempo de voo (TOF). O gatilho é produzido simplesmente pela coincidência dos sinais dos hodoscópios de juzante e de montante como ilustrado na figura 5.1 indicando que uma partícula carregada passou pelos dois conjuntos.

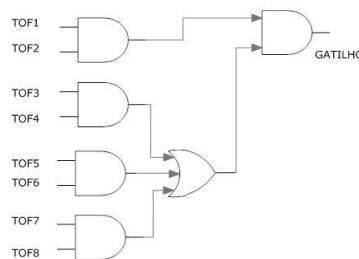


Figura 5.1: Diagrama lógico do gatilho

O sinal de gatilho é enviado para um gerador de pulsos que gera, então, o gatilho do sistema.

O sinal dos hodoscópios são enviados para conversores analógico-digitais (TDC) que, disparado pelo gatilho, registra o tempo de todos os sinais dos hodoscópios. A combinação destas medidas permite determinar o tempo que a partícula do feixe leva para percorrer o percurso entre o par TOF1-TOF2 e o segundo conjunto de hodoscópios¹.

¹Antes de todos os trabalhos com a eletrônica todos os sinais foram sincronizadas.

5.2 Identificação das Partículas do Feixe

A combinação do colimador e magnetos resulta em um feixe de 0,5 GeV/c constituídos de π , p , e^+ e e^- como descrito no capítulo 4. A diferença de massa das partículas implica em tempo de voo diferente entre os dois conjuntos de hodoscópios e, portanto, a diferença entre os tempos de voo permite a identificação das partículas desde que a medida do tempo de voo tenha a resolução adequada. A tabela 5.1 mostra o tempo de voo típico de cada partícula do feixe.

Partícula	Massa (MeV)	Tempo de voo (ns)
p	938,0	47,03
π	140,0	20,090
e^+, e^-	0,5	20,002

Tabela 5.1: Tempo de voo das partículas do feixe para momentum de 0,5 GeV/c e distância de 6 m entre os hodoscópios

A identificação da partícula é obtida combinando-se o tempo de voo com o momentum que é calculado a partir do desvio sofrido pela partícula ao passar pelos magnetos. Na seção 5.8 descrevemos o trabalho realizado para determinação e otimização da resolução da medida do tempo de voo.

5.3 Configuração do Feixe

O FERMILAB pode fornecer-nos feixes de píons de diversas energias para a geração de nosso feixe de teste. Realizamos vários testes com diferentes configurações buscando a que melhor serve a nosso propósito. Cada feixe secundário fornecido pelo FERMILAB possui estrutura diferente o que afeta nosso feixe final. Para cada configuração medimos o tempo de voo das partículas de nosso feixe e selecionamos a configuração que produz um feixe mais limpo. Nosso objetivo é dispor de um feixe preponderantemente de píons de 0,5 GeV/c. Mostramos quatro configurações tentadas e, para cada uma delas, medimos o tempo de voo das partículas mostrados na figura 5.2.

- Píons de 8 GeV/c sem cortes nas câmaras de fios e com os magnetos ligados. Observando o histograma verificamos que para um total de 969 eventos registrados 441 aparecem no pico principal e 528 nos picos extras. Estes picos extras são associados à forma como o Tevatron envia o feixe de prótons de 120 GeV/c. Os picos secundários constituem um background em nosso feixe.

- Píons de 8 GeV/c com cortes nas câmaras de fios e magnetos desligados. Agora consideramos apenas eventos onde temos um mínimo de três planos em cada câmara de fios. A distribuição vista na figura 5.2 mostra que, para a mesma tomada de dados, temos 200 eventos no pico principal e 81 eventos nos pacotes adjacentes.
- Píons de 16 GeV/c com corte nas câmaras e com os magnetos ligados. Observamos que de um total de 484 eventos registrados apenas 5% estão nos pacotes secundários.
- Píons de 16 GeV/c com corte nas câmaras e com os magnetos desligados. Registramos 1.137 eventos nesta tomada de dados sendo que cerca de 10 % aparecem nos pacotes secundários.

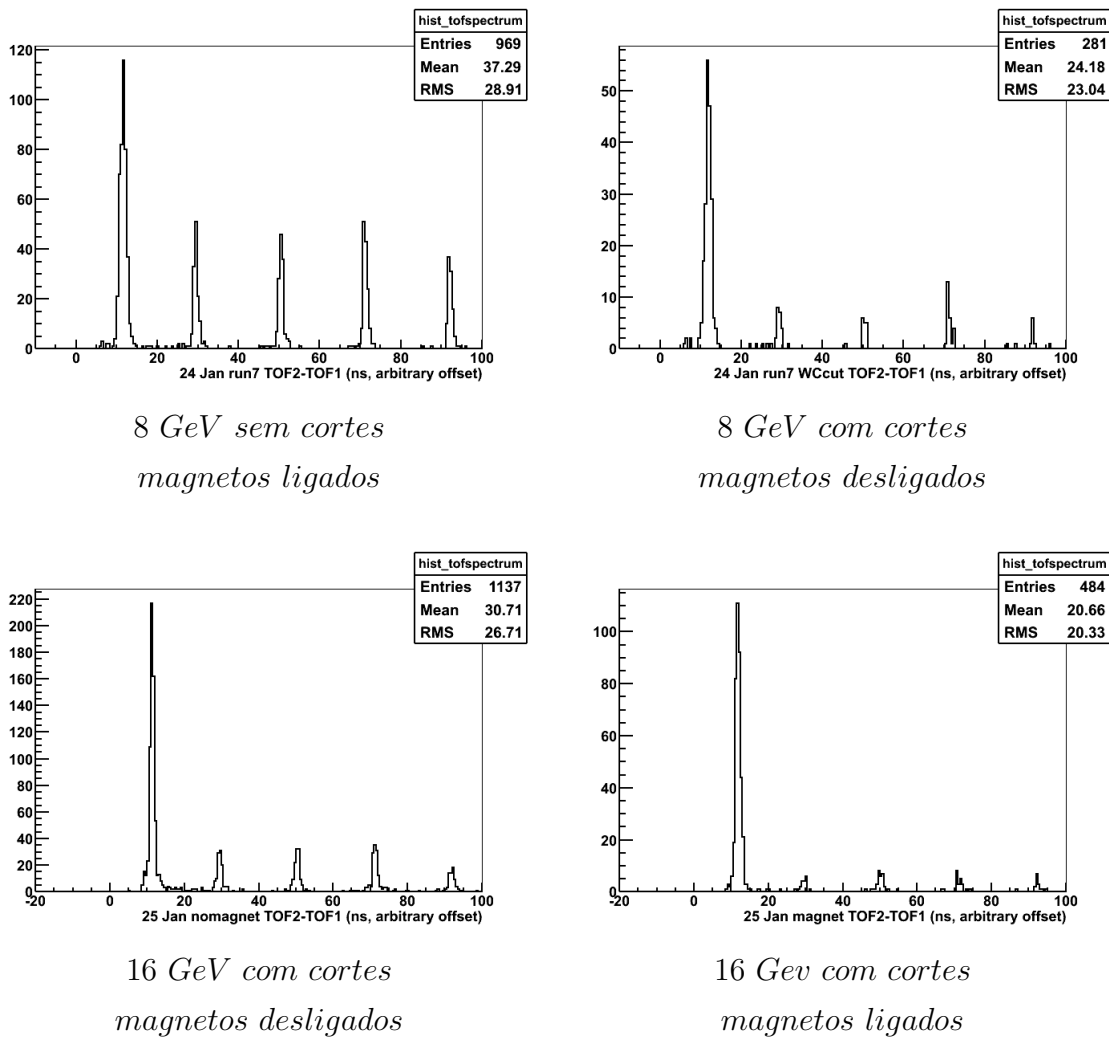


Figura 5.2: Distribuições de tempo de voo para quatro diferentes configurações de feixe.

A melhor configuração é um feixe de píons de 16 GeV, por resultar em mais eventos no pico principal, com poucos eventos em pacotes secundários.

5.4 Resposta das Câmaras de Fios

Desenvolvemos um *software* para leitura e processamento em tempo real dos dados provenientes do DAQ e registrar os canais das câmaras que apresentam sinal. Determina-se, então, o número de canais ativos para cada plano de cada câmara e as posições espaciais X,Y correspondentes. A figura 5.3 apresenta a ocupação das câmaras e a figura 5.4 mostra a posição XY do feixe como determinado pelas câmaras.

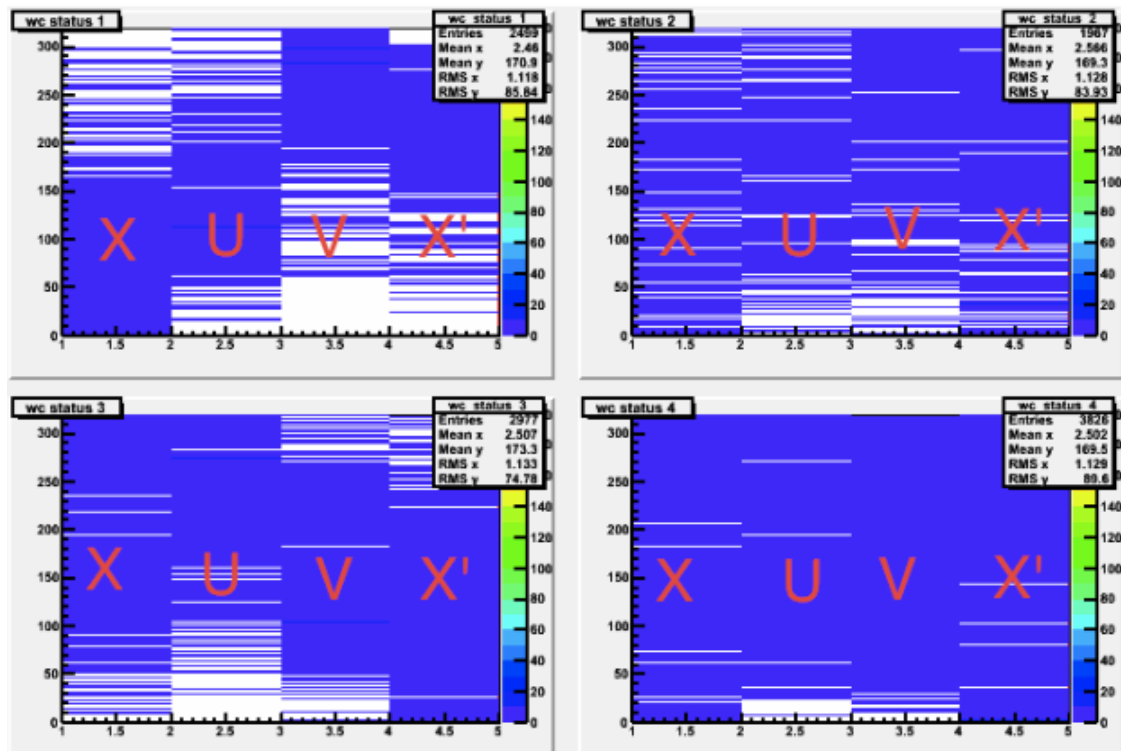


Figura 5.3: Ocupação de canais das câmaras de fio. O eixo vertical corresponde a cada um dos 320 canais de cada câmara e o eixo horizontal divide-se entre os planos X,U,V e XP

Este programa permite a monitoração *online* do sistema e identificação imediata de problemas não apenas nas câmaras como em qualquer parte da linha de feixe.

A distribuição da posição do feixe observada na câmara 1 (ver figura 5.4) que se encontra logo após o colimador mostra uma forma retangular de aproximadamente 15 cm X 7,5 cm que corresponde à abertura do colimador. As câmaras que se seguem mostram a dispersão sofrida pelo feixe. As câmaras 3 e 4 mostram a deformação sofrida pelo feixe devido à passagem pelos magnetos.

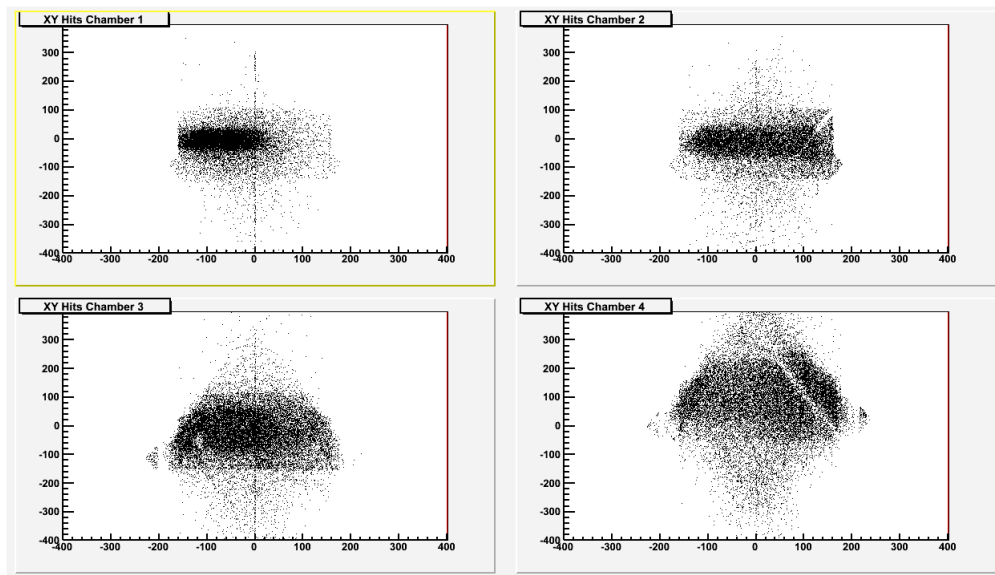


Figura 5.4: Distribuição espacial em corte transversal do feixe como registrado pelas câmaras de fio

5.5 Distribuição dos Momenta

Uma primeira estimativa do momentum das partículas do feixe é realizado segundo os passos seguintes;

- Cálculo das posições XYZ em cada câmara;
- Construção de dois segmentos de reta: um entre a câmara 1 e a câmara 2 e outro entre as câmaras 3 e 4;
- Cálculo do ângulo entre os dois segmentos;
- Empregando o valor do campo magnético estimamos o momento da partícula compatível com o desvio sofrido pela partícula.

Estes cálculos requerem o conhecimento da localização espacial de todos os componentes da linha de feixe (câmaras e magnetos). Estes dados são obtidos através de levantamento de dimensões e demarcações realizado por equipe de agrimensores do FERMILAB e incluídos em um arquivo de configuração que é lido pelos programas de processamento e análise. Este arquivo de configuração também contém o modelo dos campos magnéticos² obtidos no levantamento

²Empregamos nesta fase do trabalho apenas a componente vertical dos campos magnéticos que são responsáveis pela maior contribuição aos desvios das partículas. As componentes transversais representam menos de 5%. Seus efeitos, entretanto, poderão ser considerados em um futuro refinamento do trabalho e, se necessário, facilmente incluídos no arquivo de configuração.

descrito no capítulo 4

A figura 5.5 mostra a distribuição de canais ativados (*hits*) para a câmara 1. A grande maioria dos eventos resulta em até cinco canais ativados.

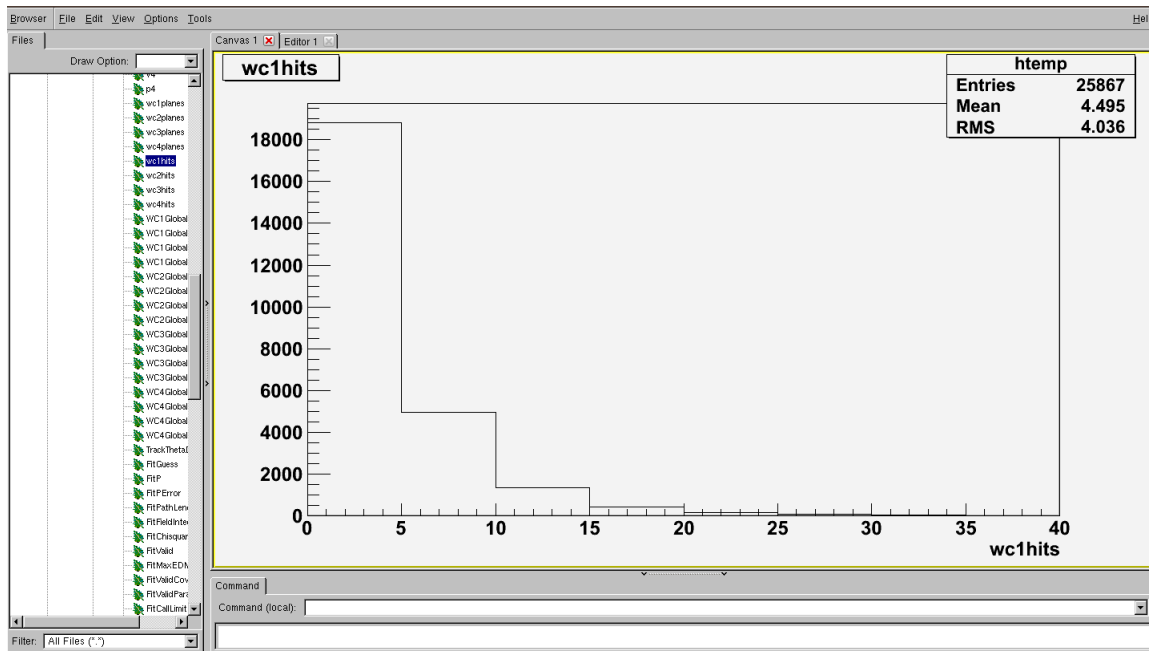


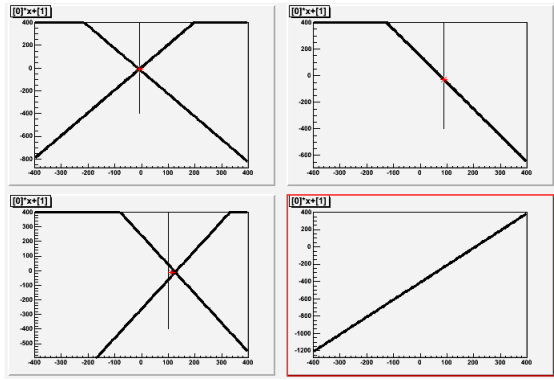
Figura 5.5: Rootupla gerada e histograma mostrando o número de canais ativados por evento para a câmara 1.

Um algoritmo de Runge-Kutta é empregado para computação da passagem da partícula pelos campos magnéticos e o pacote MINUIT 2 é usado para ajustar a melhor trajetória da partícula obtendo-se o desvio sofrido e o momentum correspondente.

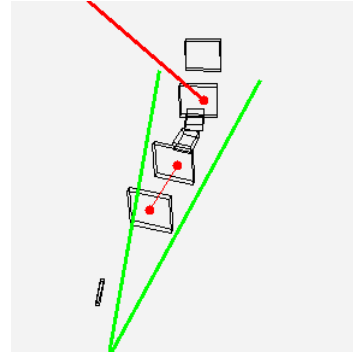
A figura 5.6 mostra duas situações distintas. Na parte superior temos um evento onde podemos reconstruir coordenadas nas câmaras 1, 2 e 3 (três canais ativos nas câmaras 1 e 3 e dois canais ativos na câmara 3). Entretanto, a câmara 4 tem apenas um canal ativo o que não permite a determinação de nenhuma coordenada XY nesta posição³ nesta posição. Neste evento podemos reconstruir um segmento de trajetória entre as câmaras 1 e 2 mas não temos o segundo segmento após os magnetos. A trajetória não pode ser reconstruída, como visto na representação tridimensional à direita.

Na parte inferior da figura temos um evento onde podemos reconstruir coordenadas nas quatro câmaras. Por conseguinte o programa reconstrói uma trajetória completa, como visto na imagem tridimensional à direita.

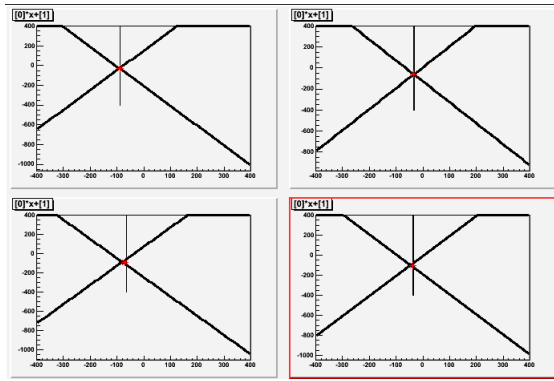
³A coordenada Z é dada, em primeira aproximação, pela posição da câmara determinada pelas medidas dos agrimensores.



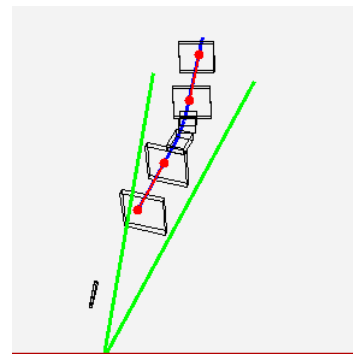
Coordenadas XY de trajetoria obtidas nas camaras 1, 2 e 3 apenas



Trajectoria nao pode ser reconstruida



Coordenadas XY reconstruidas em todas as camaras



Trajectoria reconstruida

Figura 5.6: Reconstrução de trajetórias. Visão 2D e 3D de eventos com hits e sem hits nas camaras de fios.

Por geometria e pelo projeto da linha de feixe, o ângulo de deflexao deve estar perto de 15 graus, mesmo para coincidências acidentais. Então, a primeira estimativa do momentum P é dada por

$$P = \frac{P_t}{\sin(\theta)} \quad (5.1)$$

onde θ é o ângulo entre os segmentos reconstruídos pelas câmaras de fios, e P_t é o momentum transversal para o qual fazemos uso do valor fixo de 100 MeV/c, que foi obtido a partir de simulações do campo magnetico dos magnetos usados.

Uma primeira estimativa da distribuição de momentum pode ser vista na figura 5.7, onde observamos um pico em aproximadamente 0,4 GeV/c. A figura 5.8 apresenta a distribuição de *momenta*, obtidos a partir do ajuste feito com o Minuit 2. Esta distribuição tem um erro esperado entre 5 % e 10 % devido, principalmente, a incertezas no mapeamento do campo

magnético. Uma melhoria da medida do campo magnético deve significar uma redução dos erros cerca de 2 %.

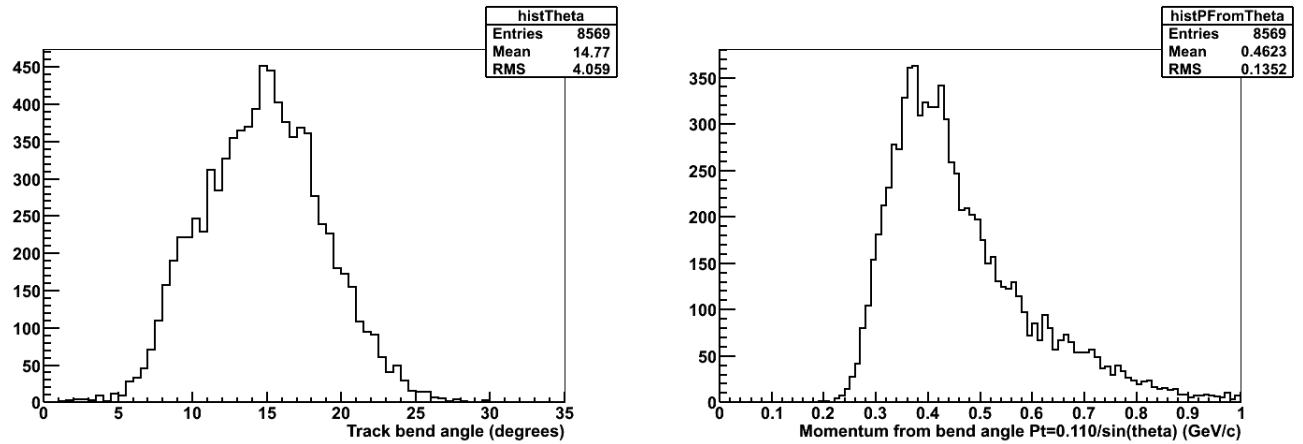


Figura 5.7: Ângulo de deflexão e momentum calculados a partir do momentum transversal

A figura 5.8 apresenta a distribuição da distância percorrida pela partícula entre os hodoscópios os momenta obtidos a partir da reconstrução da trajetória obtida pelo melhor fit, como descrito anteriormente. Esta distribuição tem um erro que pode oscilar no intervalo de 5 % a 10 % devido, principalmente, a falhas na determinação do campo magnético.

O código por nós desenvolvido tem, também, a capacidade de fornecer informação sobre o comprimento da trajetória da partícula. O erro nesta variável está ao redor de 1 %.

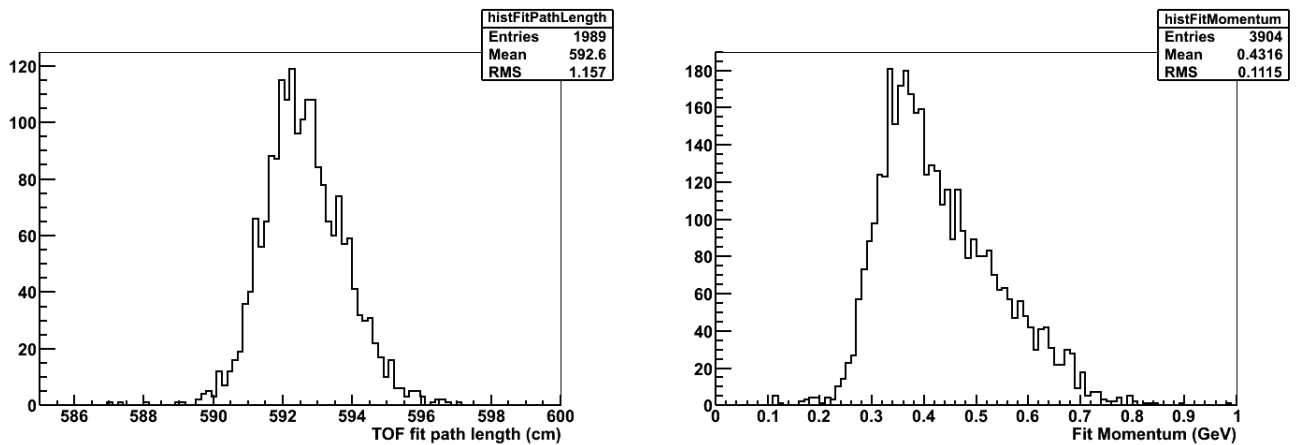


Figura 5.8: Distância percorrida e momentum determinados a partir do melhor ajuste da trajetória.

A figura 5.9 permite uma comparação entre os resultados obtidos em simulações e a partir de tomada de dados.

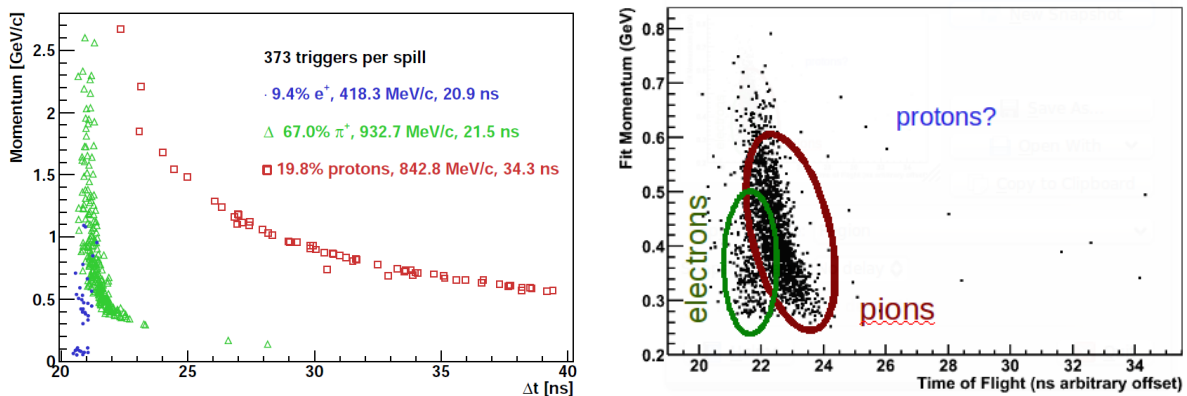


Figura 5.9: Tempo de voo (TOF) em função do momentum. Esquerda: simulação ; Direita: Dados.

Na figura obtida para os dados reais só temos partículas com momentum de até aproximadamente 800 MeV/c devido às dimensões e posicionamento de nossas câmaras pois partículas com maior momentum sofrem um desvio menor e não passam pela parte ativa da câmara 4.

Aos dados aqui apresentados faltam muitos outros cortes e critérios de qualidade que são feitos com vários códigos que foram desenvolvidos para este fim. A figura 5.10, por exemplo, apresenta a mesma distribuição anterior mas com um corte fiducial sobre regiões nas 4 câmara de fios onde encontramos a maior qualidade de *hits* para que sejam empregados na reconstrução. Isto é feito a partir das distribuições das posições (X,Y) nas câmaras indicado na figura. O pico dos píons termina 2 nanosegundos depois do pico dos elétrons, o que concorda com o obtido na simulação.

Em resumo, dos 7.800 eventos com sinais de TOF válidos, e um número mínimo de *hits* nas câmaras de fios, 1.927 foram selecionados como eventos reconstruíveis e com ajustes válidos sendo 1.859 no pico principal de nossa distribuição de tempo de voo e 68 nos pacotes adjacentes, durante 6 horas de tomada de dados. Fazendo os cortes fiducial temos 1.093 eventos no pico principal da distribuição de tempo de voo, e 14 nos pacotes adjacentes. O objetivo foi sempre reduzir o *background* representado por estes pacotes fora do pico principal

5.6 Calibração pelo Cálculo de Massa do Píon

Para calibração e ajuste de nossa linha de feixe calculamos a massa do π a partir dos dados obtidos durante uma tomada de dados. Comparando o valor obtido com o valor conhecido,

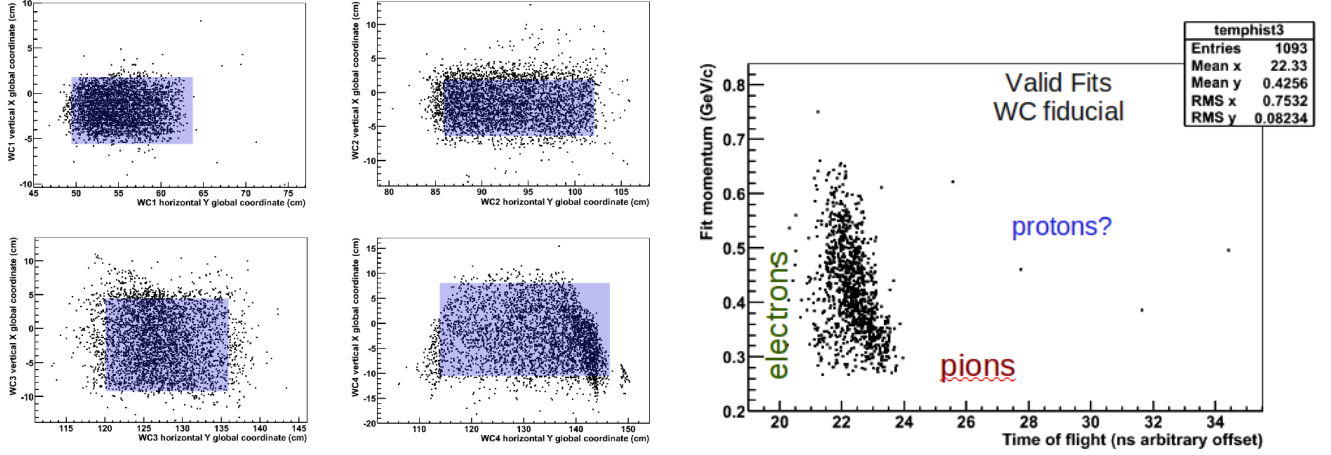


Figura 5.10: Esquerda: cortes fiduciais nas câmaras de fios. Direita: distribuição de momentum versus TOF depois de aplicado cortes fiduciais

podemos verificar a operação de nosso sistema, calibrá-lo e realizar os ajustes necessários.

A massa do π pode ser determinada a partir da medida do momento e da velocidade da partícula através das expressões:

$$m = \sqrt{\frac{p^2(1 - \beta^2)}{\beta^2}} \quad (5.2)$$

$$\beta = \frac{v}{c} \quad (5.3)$$

onde p é o momentum, v é a velocidade da partícula e c é a velocidade da luz.

A velocidade da partícula é determinada como sendo

$$v = \frac{LOF}{TOF} \quad (5.4)$$

com LOF sendo a distância percorrida e TOF o tempo de voo.

As seguintes variáveis afetam a determinação da massa:

- a distância entre os hodoscópios e as câmaras de fio;
- o valor do campo magnético nos magnetos;

Em nosso trabalho calculamos a massa dos pions do feixe para vários valores destas variáveis e empregando alguns cortes de seleção.

A figura 5.11 ilustra o resultado deste processo. Nela vemos tres conjuntos de gráficos obtidos em três situações diferentes: o superior mostra a relação entre β e o momentum da partícula do feixe; o inferior mostra a massa calculada para a partícula. O par da esquerda mostra a situação

antes de qualquer ajuste ou corte. No centro temos a situação com o corte nos momentos e β e uma variação da distância entre hodoscópios e câmaras. Observa-se que a distribuição de massa tem sua média próxima ao valor da massa conhecida do π embora ainda com um desvio padrão bastante grande. Nos gráficos da direita temos outro conjunto de corte β -momentum e diferente valor para a distância entre hodoscópio e câmaras. Esta figura mostra a situação final que fornece os valores de todos os parâmetros a serem usados e incluídos nos programas de análise. A tabela 5.2 apresenta um resumo deste estudo.

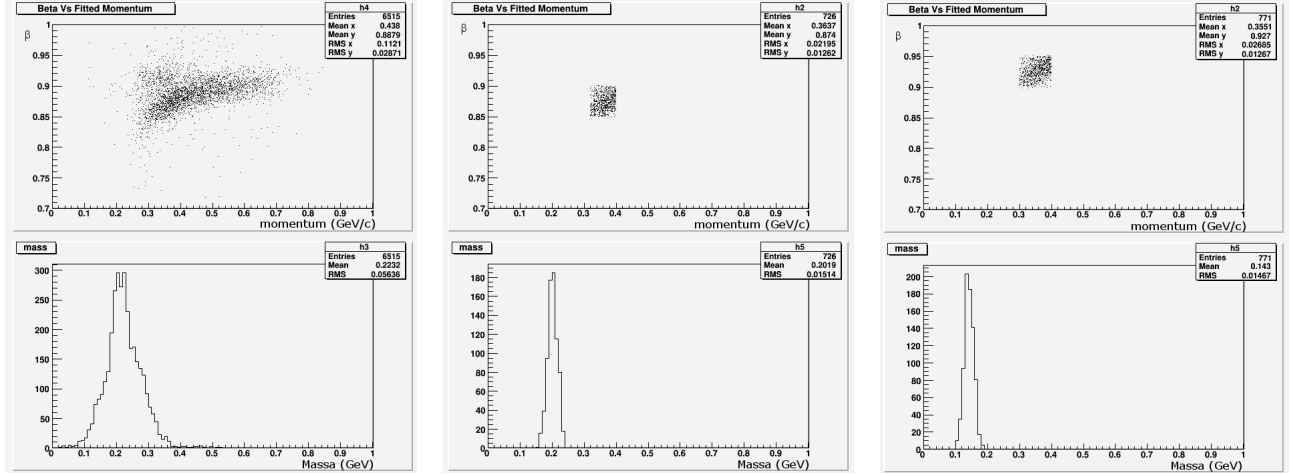


Figura 5.11: Distribuição da massa do pión para diferentes valores de distância percorrida pelos pions.

	Antes de cortes		Depois de cortes	
distância entre hodoscópios e câmaras (cm)	massa (GeV)	RMS	massa (GeV)	RMS
45	0,223	0,056	0,201	0,015
30	0,165	0,045	0,151	0,016
40	0,143	0,044	0,143	0,014

Tabela 5.2: Calibração da linha de feixe empregando a determinação da massa do pión

A distância entre os hodoscópios e as câmaras foi inicialmente determinada através de uma medida simples com uma fita métrica como sendo de 45 ± 1 cm. O processo descrito acima indicou o valor a ser usado como sendo cerca de 40 cm.

5.7 Eficiência das Câmaras de Fios

A eficiência de uma câmara é definida como:

$$\epsilon = \frac{NE}{NT} \quad (5.5)$$

onde o NE é o número de eventos lidos pela câmara de fios e NT é o número total de gatilhos usados.

A eficiência é calculada para diversos gatilhos e com diversas exigências sobre os sinais das câmaras como, por exemplo, número de *hits* por câmara, espaçamento entre os *hits*. A eficiência das câmaras influencia diretamente a reconstrução de trajetórias e determinação dos momenta que é o objetivo dos sistema de câmaras.

Desenvolvemos um código para leitura dos sinais das câmaras e cálculo das eficiências para diversas combinações de gatilho e critérios de seleção de sinal com vista a otimizar a eficiência. Inicialmente a eficiência das câmaras era de cerca de 50% passando para cerca de 90% após o trabalho de otimização. A configuração ótima foi, então, implementada.

5.8 Resolução do Tempo de Voo

Conforme mostrado na seção 5.2 a identificação das partículas do feixe depende da resolução da medida do tempo de voo. Para discriminar p das demais partículas do feixe (píons e elétrons) o sistema precisa ser capaz de distinguir entre 400 ns e 20 ns. Entretanto, a discriminação entre pión e elétrons exige a capacidade de distinguir entre 20,09 ns e 20,002 ns, ou seja cerca de 90 ps de resolução.

A partir de dados obtidos em uma tomada de dados verificamos ser a resolução da medida de tempo de voo da ordem de 400 ps, o que não é suficiente para discriminação entre π e elétrons. A resolução de tempo depende principalmente da velocidade de resposta das fotomultiplicadoras e dos discriminadores. Realizamos, então, um trabalho de otimização da operação destes dispositivos variando as altas voltagens de alimentação das fotomultiplicadoras e os patamares dos discriminadores. Este trabalho foi realizado a partir de tomada de dados empregando raios cósmicos, cada uma com duração de seis horas, durante três dias empregando um código que desenvolvemos para a análise destes dados e verificação da resolução alcançada.

A figura 5.12 mostra a distribuição do tempo de voo no início de nosso trabalho e após os ajustes realizados.

No momento, a resolução alcançada encontra-se na faixa de 230 ps o que não é, ainda, suficiente para a discriminação $\pi - e$. Entretanto, como visto na seção 4.3.2, há grande prevalência de π na região de trabalho de nosso feixe. A melhora da resolução será tentada com a substituição das fotomultiplicadoras por outras mais rápidas. No momento os elétrons são considerados como *background* de nosso feixe e a sua identificação pode ser obtida pela

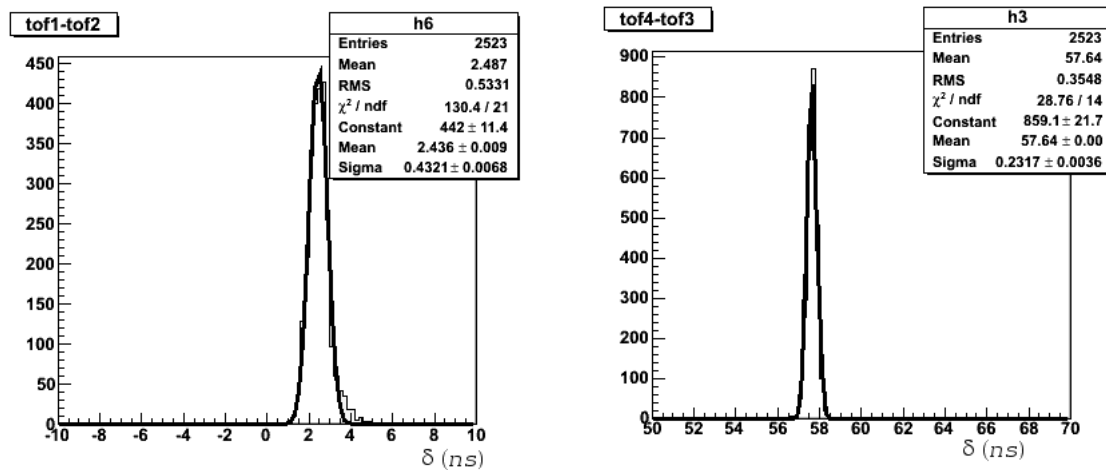


Figura 5.12: Verificação da resolução do tempo de voo. O gráfico da esquerda mostra as medidas do tempo de voo no início do processo e a figura da direita apresenta o resultado após os trabalhos de otimização realizados. A variável Sigma indica a resolução. O eixo horizontal apresenta apenas a diferença das medidas de tempo entre os conjuntos de hodoscópios com origem da contagem de tempo arbitrária.

análise dos dados obtidos do detector de teste pois píons e elétrons deixam sinais marcadamente diferentes no detector, como verificado em simulações.

5.9 Sinal Duplo no Sistema de Tempo de Voo

Uma partícula ao atravessar nossa linha de feixe deixa sinal no par de hodoscópios TOF1-TOF2 e em um dos três pares de hodoscópios TOF3-TOF4 ou TOF5-TOF6 ou TOF7-TOF8. Com os dados provenientes do sistema de aquisição DAQ, especificamente dos dados do nosso sistema de tempo de voo, verificamos que para aproximadamente 1% do total dos dados há sinal em dois pares dos hodoscópios à jusante das câmaras. Várias podem ser as causas deste efeito, como a passagem de duas partículas no feixe na mesma janela de tempo ou partículas decorrentes de interações ao longo da linha. Desenvolvemos um código para identificar estes eventos a partir da distribuição de contagens dos TDC. Esta informação pode ser usada para eliminação destes eventos, seja *offline* ou pela geração de um veto ao gatilho.

5.10 Teste Geral do Sistema

Para verificar o funcionamento de todo o sistema, realizamos diversas tomadas de dados empregando o gatilho, os sinais dos TDC, a identificação das partículas do feixe, e cálculo dos

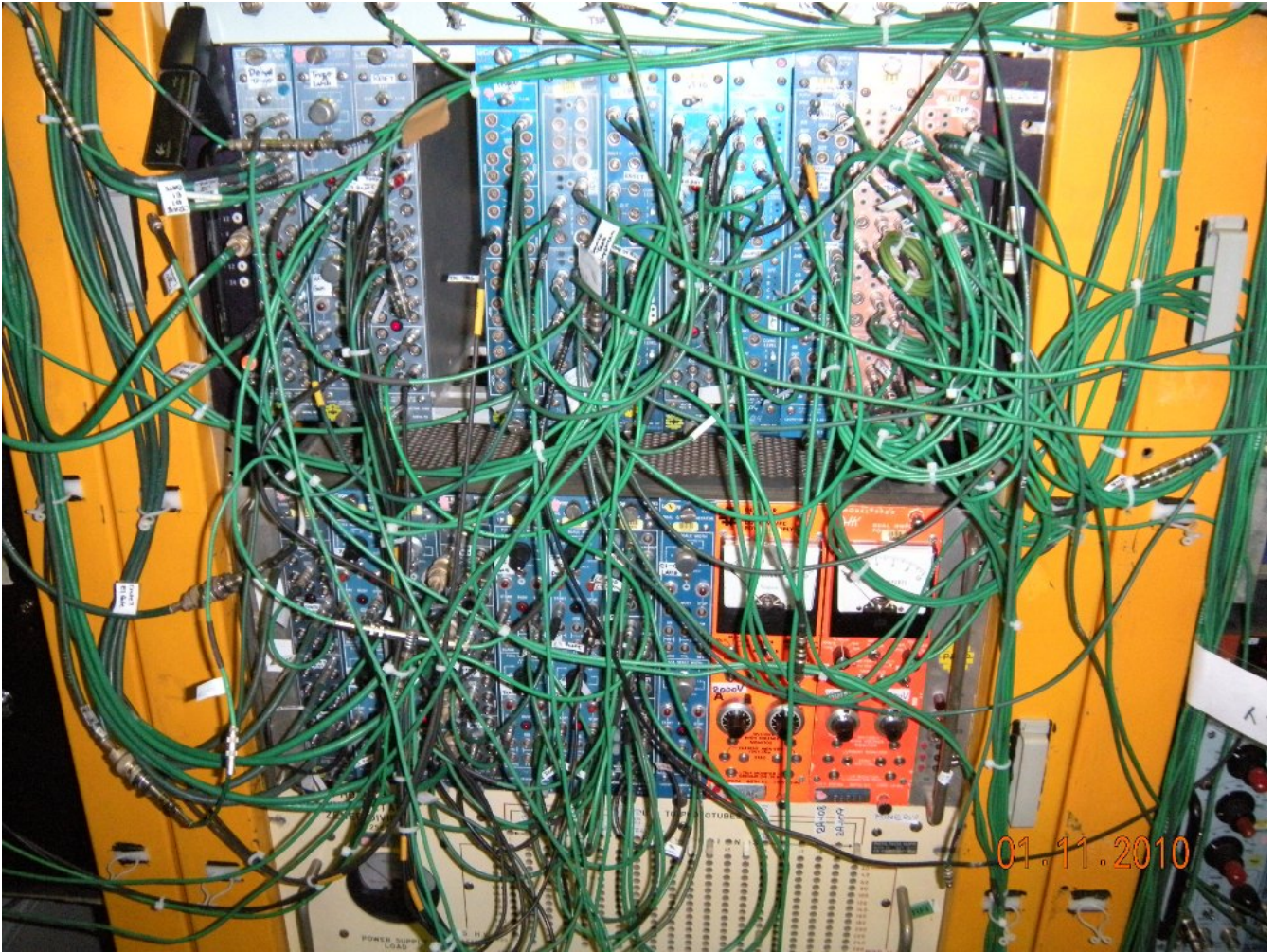


Figura 5.14: Conexões eletrônicas feitas para a linha de feixe de teste

5.11 Simulação do Detector de Teste do MINER ν A

Adaptamos o código de simulação do detector completo do MINER ν A para a simulação do detector de teste. O objetivo principal foi permitir um acesso rápido do usuário às diversas variáveis envolvidas e, então, relacioná-las de maneira rápida e eficiente. Nesta idéia de simulação o usuário pode alterar a geometria do detector, as partículas de entrada e todos demais parâmetros a partir das folhas de uma árvore de uma *rootupla*. A figura 5.17 mostra a estrutura da *rootupla* gerada para a simulação do detector de teste.

Igualmente, adaptamos o mostrador de eventos (*event display*) do experimento para uso com o detector de teste. Nesta versão o usuário pode manipular parâmetros como o número de fotoelétrons, realizar cortes nesta variável e ver o resultado no mostrador de eventos, além de guardá-los em arquivos de imagem instantaneamente.

A figura 5.18 mostra um exemplo no qual empregamos um pión simulado de 600 MeV/c em uma geometria do detector consistindo de 10 módulos de rastreamento, 20 módulos de

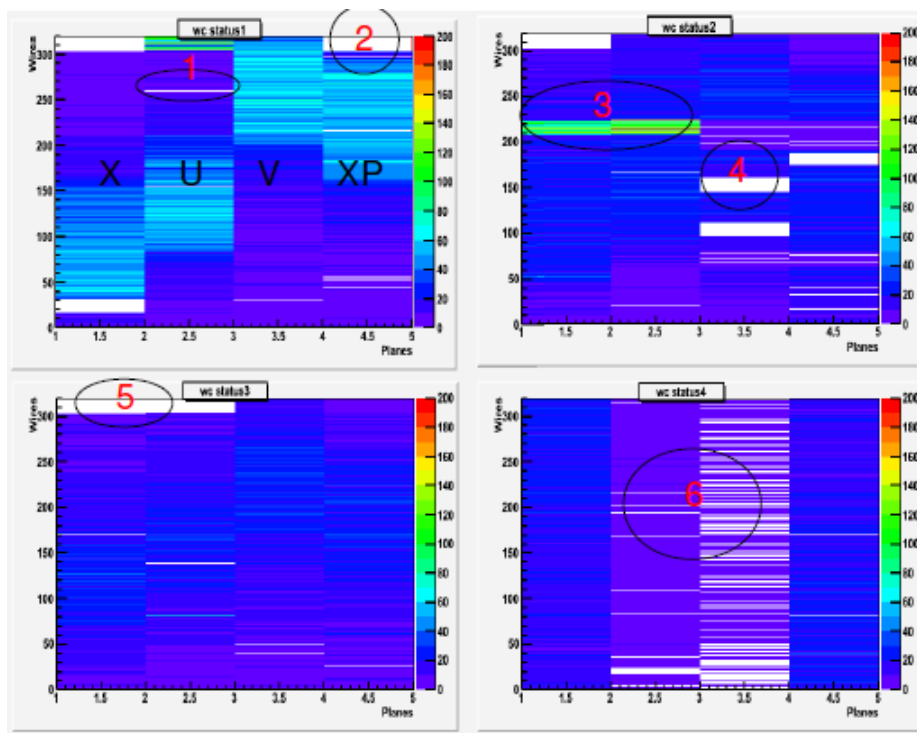


Figura 5.15: Ocupação das câmaras com todo o sistema montado. Observa-se vários canais com ocupação zero.

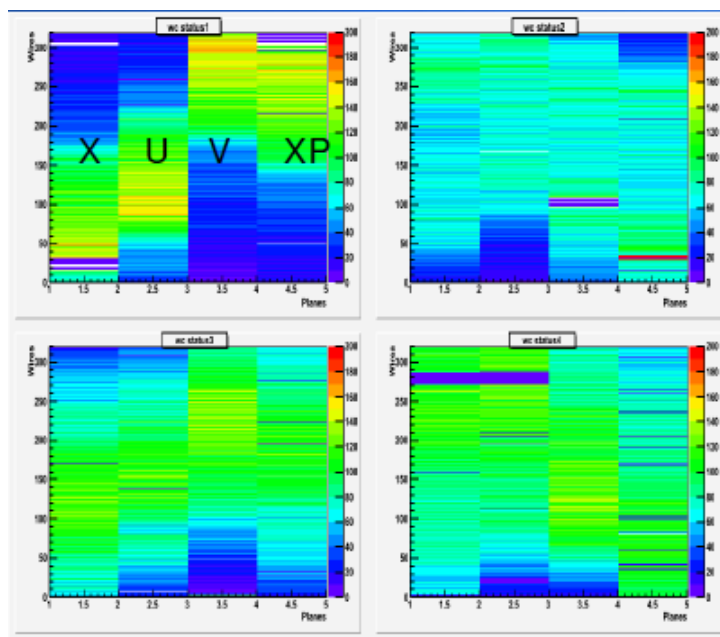


Figura 5.16: Ocupação das câmaras após operações de ajustes na eletrônica da linha de feixe calorímetro eletromagnético e 10 módulos de calorímetro hadrônico.

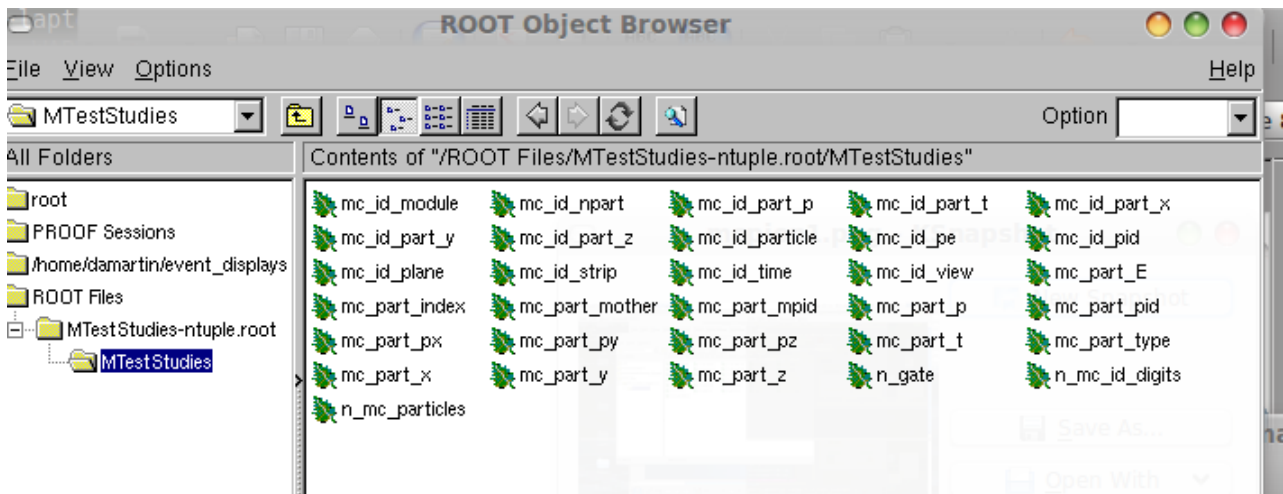


Figura 5.17: Rootupla gerada com a nova versão da simulação implementada para detector de teste do MINER ν A .

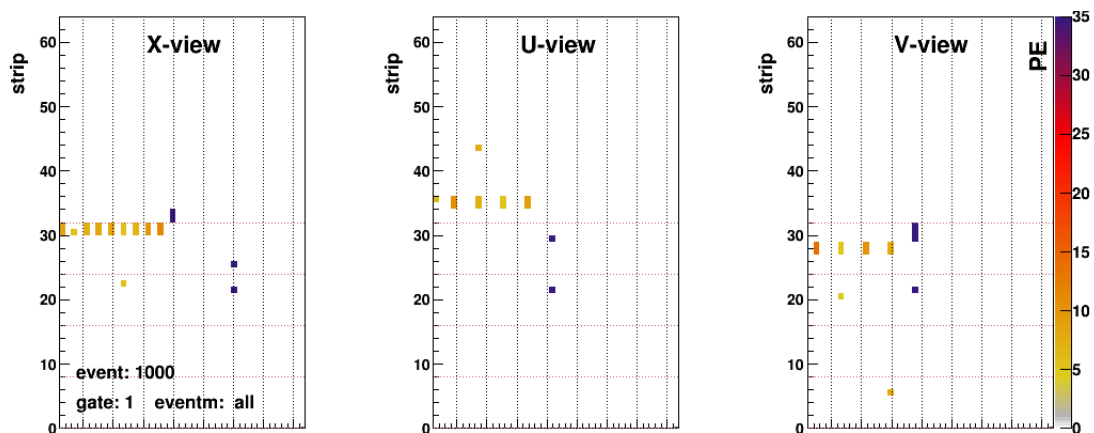


Figura 5.18: *Event Display* mostrando um pión de 600 MeV/c gerado com simulação do detector de teste do MINER ν A com geometria de 10 módulos de rastreamento, 20 módulos de calorímetro eletromagnético e 10 módulos de calorímetro hadrônico.

Capítulo 6

Conclusões

O experimento MINER ν A emprega uma versão em menor escala de seu detector para estudar o comportamento e resposta do detector. Para isto necessita-se de um feixe de hádrons construído especialmente para esta finalidade.

Trabalhei no grupo do feixe de teste onde montamos completamente a linha de feixe de teste, testamos todos os seus componentes individualmente, integramos as partes e testamos seu desempenho como um sistema. Em resumo:

- Determinamos a melhor configuração do feixe fornecido pelo Fermilab para atender às nossas necessidades.
- Montamos, instrumentamos e testamos todas as câmaras de fios tornando-as operacionais e determinamos seus parâmetros de operação.
- Montamos o sistema de hodoscópios empregados para geração do gatilho e na medida do tempo de voo das partículas do feixe.
- Tornamos operacionais os magnetos da linha de feixe realizando o mapeamento do campo magnético.
- Desenvolvemos os códigos empregados para o sistema de DAQ e para leitura de todos os componentes.
- Desenvolvemos o processo de identificação das partículas do feixe.
- Desenvolvemos o código para determinação dos *momenta* das partículas do feixe, otimizando os parâmetros empregados.
- Montamos toda a eletrônica do sistema da linha de feixe, testamos e otimizamos sua operação.

Como resultado deste trabalho a linha de feixe de teste entrou em operação em abril de 2010. O detector de teste foi instalado e o feixe vem sendo empregado para os estudos do comportamento do detector do MINER ν A . Estes estudos são fundamentais para a análise dos dados do experimento, pois permitem modelar corretamente a resposta do detector.

O experimento MINER ν A já encontra-se na fase de tomada de dados e os códigos empregados na reconstrução e análise vão incorporar os resultados obtidos com o detector de teste. O conjunto linha de feixe de teste mais detector de teste consiste, por si só, em um experimento de física experimental de altas energias. O trabalho aqui descrito me permitiu um contato direto com todos os estágios de desenvolvimento, construção e operação de um experimento.

No momento, continuam, no FERMILAB, os trabalhos de aprimoramento da linha de feixe buscando-se, principalmente, o aumento da resolução na medida do tempo de voo.

Como continuação deste trabalho participarei diretamente da tomada de dados do experimento MINER ν A e trabalharei no desenvolvimento de códigos de análise e na análise de dados que está em seus estágios iniciais. Neste contexto, o trabalho no estabelecimento da linha de feixe de teste fornece excelente base para o trabalho que se segue.

Apêndice A

Teste de MAPMT

A.1 Teste dos Fotomultiplicadores

As fotomultiplicadoras constituem um dos dispositivos principais do detector e o objetivo dos teste é verificar a operação de cada uma delas identificando problemas para que possam ser reparados antes da instalação final no detector. Empregamos as fotomultiplicadoras multi-anodo de 64 canais Hamamatsu H8804 (MAPMT ou simplesmente PMT) como o elemento fotossensor básico do detector, que transforma luz produzida pela passagem de partículas ionizadas através do cintilador em pulsos elétricos os quais são medidos e armazenados pela eletrônica e o DAQ. A eletrônica conectada aos MAPMT deve fornecer a tensão requerida pelos dispositivos (aproximadamente 800 V no catodo), registrar os sinais dos MAPMT e fornecer quaisquer outros sinais de controle porventura necessários. Os MPMT são montados no centro de um cilindro de aço que fornece um ambiente isolado de luz e de campos magnéticos. Esta estrutura é conhecida como PMT box, e é mostrada nas figuras A.1. Em uma das extremidades chegam as fibras ópticas que trazem os sinais de luz e na outra encontra-se a base fornecedora das altas voltagens (HV).

A.1.1 Caixa dos PMT

A caixa do MAPMT (PMT box) é feita de aço de 2,36 mm de espessura. O MAPMT, seus suportes, a estrutura de plástico para conexão das fibras (*cookie*) são montados em uma estrutura de suporte que é, então, inserida axialmente na embalagem de aço (ver figura A.1). As caixas são montadas em estruturas instaladas nas superfícies superiores do detector. A figura A.1 mostra, à direita, os elementos (*pixels*) e o mapeamento das fibras ópticas.

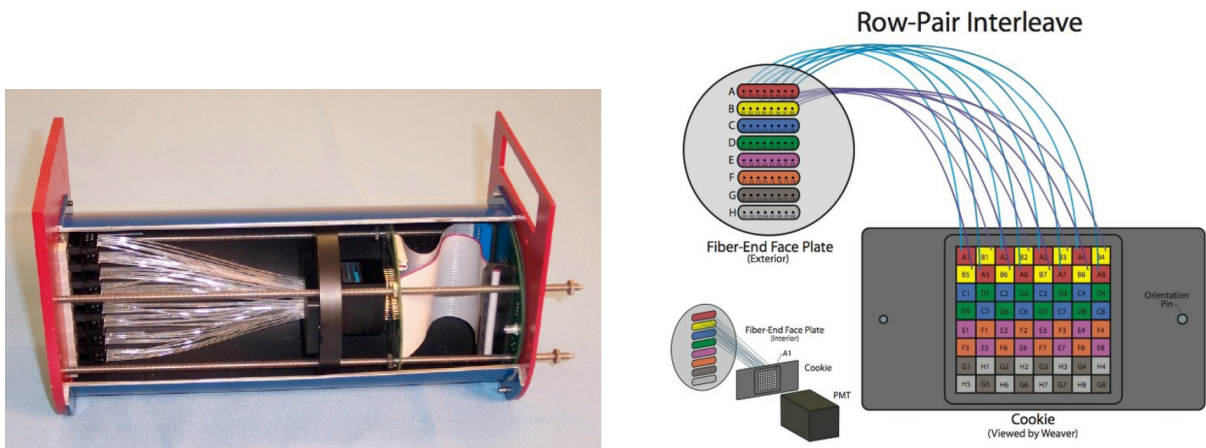


Figura A.1: Esquerda: Vista da fotomultiplicadora montada na caixa. Nota-se as fibras ópticas (à esquerda) e os cabos de sinal (à direita). Direita: Convenção de numeração dos elementos

A.1.2 Descrição das Montagens de Testes

Para fazer os testes das fotomultiplicadoras montamos o sistema ilustrado na figura A.2 cujos componentes são:

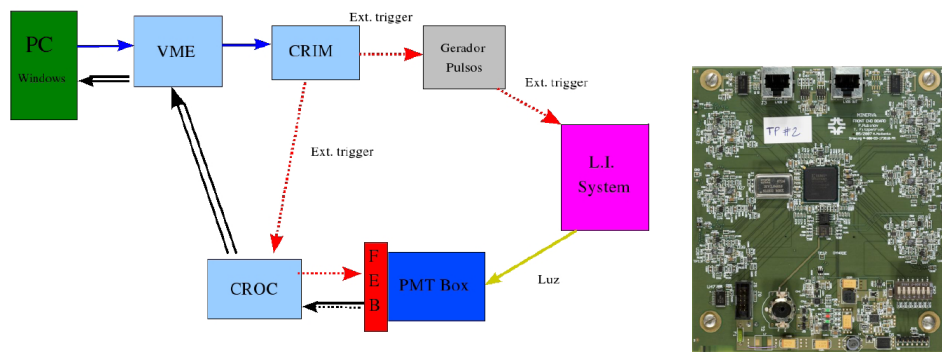


Figura A.2: Esquerda: Estrutura do sistema de testes usado. Direita: Foto da placa FEB.

- FEB(Front End Board): Uma placa baseada sobre 64 canais para a fotomultiplicadora multi-anodo. Fornece uma alta tensão programável, controla o ganho(alto,medio ou baixo) dos conversores analógicos digitais(ADC) e serve como interface para o sistema de aquisição de dados (DAQ). A placa principal da electrónica (FEB) e mostrada na figura A.2
- CROC Chain Readout Controller: Responsável pelo sincronismo, controle e leitura dos dados para o FEB.

- CRIM Chain Readout Interface Module: Realiza várias funções de interface para o CROC. Este módulo é necessário devido a grande variedade de aplicações para as quais o CROC pode ser usado.
- Injeção de Luz (L.I. System): O sistema de injeção de luz é montado com dois diodos emissores de luz (LED).
- Gerador de pulsos: Gera os pulsos para ativação dos LEDs. O processo seguido em geral para aquisição de dados é mostrado na figura A.2 onde as setas indicam o sentido do processo de trabalho. A seta tracejada de cor vermelha indica informação da sincronização e a seta dupla de cor preta indica informação dos dados. O CRIM, como interface, administra a informação, fornece sincronismo para o gerador de pulsos e este envia o sinal para o sistema que gera a luz necessária para os testes.

Na figura A.2 as setas tracejadas de cor vermelha indicam informação da sincronização e a seta dupla de cor preta indica informação dos dados. O CRIM, como interface, administra a informação, fornece sincronismo para o gerador de pulsos que envia um sinal para o sistema que produz a luz necessária para os testes.

A.1.3 Testes de Diafonia dos MAPMT

Os testes de diafonia dos MAPMT seguem os passos seguintes:

- Registra-se 1.000 eventos de pedestal (sem injeção de luz). Isto permite registrar o sinal presente na ausência de luz e identificar a ocorrência de vazamento de luz.
- Produz-se um histograma para cada canal (elemento).

A figura A.3 mostra uma vista de frente de um MAPMT indicando os 64 elementos com os elementos 10, 14, 28, 51 e 55 marcados como iluminados. Um elemento fora das bordas da face da MAPMT possui 8 vizinhos. Para medirmos a diafonia de um elemento iluminado registramos a contagem obtida neste elemento e em todos os seus n vizinhos imediatos. A diafonia é calculada pela expressão

$$diafonia = \sum_{i=1}^n \frac{N_i^{light} - N_i^{pedestal}}{N_0^{light} - N_0^{pedestal}} \times 100 \quad (A.1)$$

Onde N_0^{light} é a contagem do elemento iluminado, $N_0^{pedestal}$ é a contagem de pedestal do elemento iluminado, N_i^{light} é a contagem do vizinho i do elemento iluminado e $N_i^{pedestal}$ é a contagem de pedestal do vizinho i do elemento iluminado.

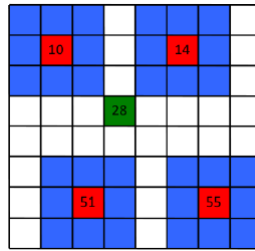


Figura A.3: Face de um MAPMT de 64 elementos. Os elementos são numerados de 1 a 64 começando no canto superior esquerdo. Os elementos 10, 14, 28, 51 e 55 são marcados como iluminados.

A diafonia total de um MAPMT é a soma da diafonia de todos os elementos. Fotomultiplicadoras com diafonia total igual ou menor que 8% são aceitas para instalação no detector. Quatro ou mais contagens 5σ acima do pedestal indica ocorrência de infiltração de luz.

A figura A.4 mostra os 64 histogramas de pedestais de uma MAPMT e a figura A.5 mostra os 64 histogramas obtidos em um teste de diafonia onde os elementos são iluminados segundo o esquema mostrado na figura A.3. Observa-se claramente sinal no elemento 37 (que não está iluminado) mas é vizinho de canto superior esquerdo do elemento 28 (que está iluminado) indicando forte diafonia entre estes elementos.



Figura A.4: Distribuições de pedestais



Figura A.5: Distribuições de contagem em teste de diafonia. Apenas os elementos 10, 14, 28, 51 e 55 estão iluminados. O elemento 37 apresenta sinal indicando diafonia entre ele e o seu vizinho 28.

O resultado das medidas de diafonia de todos os MAPMT são armazenados para uso nos programas de reconstrução e análise. Detalhes sobre o sistema de teste das MAPMT podem ser encontrados em [60].

Apêndice B

Códigos Fontes Desenvolvidos

B.1 Algoritmo para identificar canais mortos

B.1.1 deadchannels.C

```
//David Martinez , Dead Channels source testing
```

```
{
    TFile *rfile = new TFile("14-3.root","READ");
5
    //int  a,b,c,d;
    unsigned int i,j,k,l

10
    TH1I *xhits = (TH1I*)rfile->Get("xhits");
    for (i=0;i<368;i++){
        if (xhits->GetBinContent(i)==0){
            cout<<"dead channel X"<<i<<endl;
15
        }
    }
    TH1I *xphits = (TH1I*)rfile->Get("phits");
    for (j=0;j<368;j++){
        if (xphits->GetBinContent(j)==0){
20
            cout<<"dead channel XP"<<j<<endl;
        }
    }
    TH1I *uhits = (TH1I*)rfile->Get("uhits");
    for (k=0;k<368;k++){
25
        if (uhits->GetBinContent(k)==0){
            cout<<"dead channel U"<<k<<endl;
        }
    }
}
```

```
        }  
    }  
    TH1I *vhits = (TH1I*) rfile ->Get("vhits");  
30 for (l=0;l<368;l++){  
    if (vhits->GetBinContent(l)==0){  
    cout<<"dead channel V"<<l<<endl;  
    }  
    }  
35  
}
```

B.2 O algoritmo basic stats (eficiência das câmaras)

B.2.1 basicstats.cxx

```
// David Martinez 2010
// Lee Patrick
// Generate some basic statistics
// from a run
5

// std libraries
#include <iostream>
#include <fstream>
10 #include <cstring>
#include <unistd.h>
#include <stdio.h>
#include <vector>
#include <sstream>

15 // ROOT
#include <TApplication.h>
#include <TFile.h>
#include <TCanvas.h>
20 #include <TROOT.h>
#include <TH2F.h>
#include <TH1F.h>

// Custom
25 #include "TestBeam.h"
#include "TCrateEvents.h"
#include "TWCRawEvent.h"
#include "THalfTrack.h"
#include "TFullTrack.h"
30 #include "TBeamLine.h"
#include "bfield_defs.h"

#include "misc_funcs.h"

35 #define NCR 8 // Number fcrates
#define NWC 4 // Number of WCs

int main(int argc, char **argv){
```

```

40  if (argc < 2) {
        std::cout << "Usage: " << argv[0] << " input.txt"
            << std::endl;
        return 1;
45  }
    TH1F *htemp = new TH1F("htemp", "n_rawhits", 100, 0, 100);

    // Instantiate our Test Beam primitives
    loadTBConfig("config/config.pot");
50  TWCRawEvent wcev(argv[1], NCR, NWC);

    Float_t n_events=0;
    Float_t n_tracks=0;
    Float_t n_tracks1=0;

55  Float_t n_wc1=0;
    Float_t n_wc2=0;
    Float_t n_wc3=0;
    Float_t n_wc4=0;
60  Float_t n_wc5=0;
    Float_t n_wc6=0;
    Float_t n_wc7=0;
    Float_t n_wc8=0;

65  Float_t n_ups=0;
    Float_t n_dns=0;
    Float_t n_ups1=0;
    Float_t n_dns1=0;

70  bool wc1, wc2, wc3, wc4, wc5, wc6, wc7, wc8;

    //Loop over events
    while (wcev.FillNextEvent()) {
75  n_events++;

        wc1=wcev.GetWC(1).Empty();
        wc2=wcev.GetWC(2).Empty();
80  wc3=wcev.GetWC(3).Empty();
        wc4=wcev.GetWC(4).Empty();
        wc5=wcev.GetWC(1).Bad();
        wc6=wcev.GetWC(2).Bad();

```

```

    wc7=wcev.GetWC(3).Bad();
85    wc8=wcev.GetWC(4).Bad();

    if(!wc1){n_wc1++;}
90    if(!wc2){n_wc2++;}
    if(!wc3){n_wc3++;}
    if(!wc4){n_wc4++;}
    if(!wc5){n_wc5++;}
    if(!wc6){n_wc6++;}
95    if(!wc7){n_wc7++;}
    if(!wc8){n_wc8++;}

    if(!wc1 && !wc2){n_ups++;}
100    if(!wc3 && !wc4){n_dns++;}
    if(!wc5 && !wc6){n_ups1++;}
    if(!wc7 && !wc8){n_dns1++;}

105

    if(!wc1 && !wc2 && !wc3 && !wc4){
        n_tracks++;
    }

110    if(!wc5 && !wc6 && !wc7 && !wc8){
        n_tracks1++;
    }

}

115    std::cout << "WC1 Events: " << n_wc1 << std::endl;
    std::cout << "WC1 Events/Triggers: " << n_wc1/n_events << std::endl;

    std::cout << "WC2 Events: " << n_wc2 << std::endl;
120    std::cout << "WC2 Events/Triggers: " << n_wc2/n_events << std::endl;

    std::cout << "WC3 Events: " << n_wc3 << std::endl;
    std::cout << "WC3 Events/Triggers: " << n_wc3/n_events << std::endl;

125    std::cout << "WC4 Events: " << n_wc4 << std::endl;
    std::cout << "WC4 Events/Triggers: " << n_wc4/n_events << std::endl;

```

```

std::cout << "Potential Upstream Tracks: " << n_ups << std::endl;
std::cout << "Ups/Triggers: " << n_ups/n_events << std::endl;
130
std::cout << "Potential Dnstream Tracks: " << n_dns << std::endl;
std::cout << "Dns/Triggers: " << n_dns/n_events << std::endl;

std::cout << "Number of Triggers: " << n_events << std::endl;
135 std::cout << "Number of Potential Tracks: " << n_tracks << std::endl;
std::cout << "Tracks/Triggers: " << n_tracks/n_events << std::endl;
;

std::cout << "WC1 Events Filter: " << n_wc5 << std::
endl;
140 std::cout << "WC1 Events/Triggers: " << n_wc5/n_events << std::endl;

std::cout << "WC2 Events Filter: " << n_wc6 << std::endl;
std::cout << "WC2 Events/Triggers: " << n_wc6/n_events << std::endl;

145 std::cout << "WC3 Events Filter: " << n_wc7 << std::
endl;
std::cout << "WC3 Events/Triggers: " << n_wc7/n_events << std::endl;

std::cout << "WC4 Events Filter: " << n_wc8 << std::
endl;
std::cout << "WC4 Events/Triggers: " << n_wc8/n_events << std::endl;
150

std::cout << "Potential Upstream Tracks Filter: " << n_ups1 << std::
endl;
std::cout << "Ups/Triggers: " << n_ups1/n_events << std::endl;

std::cout << "Potential Dnstream Tracks Filter: " << n_dns1 << std::
endl;
155 std::cout << "Dns/Triggers: " << n_dns1/n_events << std::endl;

std::cout << "Number of Triggers: " << n_events << std::endl;
std::cout << "Number of Potential Tracks Filter: " << n_tracks1 << std::
endl;
std::cout << "Tracks/Triggers: " << n_tracks1/n_events << std::
endl;
160
}

```

B.3 O algoritmo basic histos (ocupação das câmaras)

B.3.1 basichistos.cxx

```
// Lee C. Patrick
// David Martinez

// Goes event by event, figures out if it
5 // is decent enough to fit, then fits.
// Also has some primitive event display

// std libraries
#include <iostream>
10 #include <fstream>
#include <cstring>
#include <unistd.h>
#include <stdio.h>
#include <vector>
15 #include <sstream>

// ROOT
#include <TApplication.h>
#include <TFile.h>
20 #include <TCanvas.h>
#include <TROOT.h>
#include <TH2F.h>
#include <TH1F.h>

25 // Custom
#include "TestBeam.h"
#include "TCrateEvents.h"
#include "TWCRawEvent.h"
#include "THalfTrack.h"
30 #include "TFullTrack.h"
#include "TBeamLine.h"

#include "misc_funcs.h"
35
#define NCR 8 // Number fcrates
#define NWC 4 // Number of WCs
```

```

40 int main(int argc, char **argv){

    if(argc<3){
        std::cout << "Usage: " << argv[0] << " input.txt output.root"
            << std::endl;
45     return 1;
    }

    // THIS PART IS CRUCIAL!!!
    // Loads in the configuration file
50 loadTBConfig(" config/config.pot");

    TApplication myapp("myapp",0,0);
    TFile *rfile = new TFile(argv[2],"RECREATE");

55 // Instantiate our Test Beam primitives
    TWCRawEvent wcev(argv[1], NCR, NWC);

    // Create Histograms
    TH1F *xhits_histos [NWC];
60 TH1F *phits_histos [NWC];
    TH1F *uhits_histos [NWC];
    TH1F *vhits_histos [NWC];

    TH2F *WCHitAccum [NWC];
65

    TH1F thetadeg(" thetadeg","Theta: Angle between input and output",91,0,90);
    TH1F pguess(" pguess","Momentum Guess",200,0,2.0);

    // Generate names for them -> Annoying
70 for (Int_t x=0; x<NWC; x++){

        std::string xname = "x_hits_";
        std::string pname = "p_hits_";
        std::string urname = "u_hits_";
75     std::string vname = "v_hits_";
        std::string xtitle = "X Hits Chamber ";
        std::string ptitle = "X' Hits Chamber ";
        std::string utitle = "U Hits Chamber ";
        std::string vtitle = "V Hits Chamber ";

80     std::string xytitle = "XY Hits Chamber ";
        std::string xyname = "xy_hits_";

```

```

std::stringstream num;
85 num << x+1;

xname+= num.str();
pname+= num.str();
urname+= num.str();
90 vname+= num.str();
xyrname+= num.str();

xtitle+= num.str();
ptitle+= num.str();
95 utitle+= num.str();
vtitle+= num.str();
xytitle+= num.str();

xhits_histos[x] = new TH1F(xname.c_str(),xtitle.c_str(),
100 369,0,368);
phits_histos[x] = new TH1F(pname.c_str(),ptitle.c_str(),
369,0,368);
uhits_histos[x] = new TH1F(urname.c_str(),utitle.c_str(),
369,0,368);
105 vhits_histos[x] = new TH1F(vname.c_str(),vtitle.c_str(),
369,0,368);
WCHitAccum[x] = new TH2F(xyrname.c_str(),xytitle.c_str(),
401,-400,400,
401,-400,400);
110 }

// Loop over events
while(wcev.FillNextEvent()){
115 //Loop over Wire Chambers in Event
// if( !wcev.GetWC(1).Bad() &&
// !wcev.GetWC(2).Bad() &&
// !wcev.GetWC(3).Bad() &&
120 // !wcev.GetWC(4).Bad()){
if(true){

for(Int_t i=0; i<NWC; i++){

125 TWireChamber wc=wcev.GetWC(i+1);
//wc.Plot();

```

```

//wc.Print();

130   if(!wc.Bad()){
        // Fill X Hits
        std::vector<Int_t> xhits=wc.GetXHits();
        for(UInt_t j=0; j<xhits.size(); j++){
            xhits_histos[i]->Fill(xhits[j]);
135     }
        // Fill X' Hits
        std::vector<Int_t> phits=wc.GetPHits();
        for(UInt_t j=0; j<phits.size(); j++){
            phits_histos[i]->Fill(phits[j]);
140     }
        // Fill U Hits
        std::vector<Int_t> uhits=wc.GetUHits();
        for(UInt_t j=0; j<uhits.size(); j++){
            uhits_histos[i]->Fill(uhits[j]);
145     }
        // Fill V Hits
        std::vector<Int_t> vhits=wc.GetVHits();
        for(UInt_t j=0; j<vhits.size(); j++){
            vhits_histos[i]->Fill(vhits[j]);
150     }
        // Fill XY Positions
        TVector3 hit=wc.GetWCHitLocal();
        WCHitAccum[i]->Fill(hit.X(),hit.Y());
    }
155 }

    THalfTrack intrack(wcev.GetWC(1),wcev.GetWC(2));
    THalfTrack outtrack(wcev.GetWC(3),wcev.GetWC(4));

160    TFullTrack ftrack(intrack,outtrack);
    ftrack.InitialGuess();
    thetadeg.Fill(ftrack.GetThetaDeg());
    pguess.Fill(ftrack.GetGuess_p());
}

165 }

    rfile ->Write();
    rfile ->Close();

170    return 0;

```

}

B.4 O algoritmo wchits (visualização de ocupação das câmaras)

B.4.1 wchits.C

```

//Use the output file from basic_histos.cxx
//David Martinez Jan2010

void wchits(char* filename) {
5     gROOT-> Reset();
        char name[50];
        sprintf(name,"%s.root",filename);
        TFile *rfile = new TFile(name,"READ");

10    //{
//TFile *rfile = new TFile("wc1_data_14.root","READ");
        TH1F *wccx[4];
        TH1F *wccy[4];
        TH1F *wccp[4];
15     TH1F *wccv[4];
        TH1F *wcc[4];
        TH1F *wd[4];
        for (Int_t i=1;i<5;i++){
            wccx[i]= (TH1F*) rfile ->Get(Form("x_hits_%d",i));
20
        };
        //gStyle->SetOptStat(kFALSE);
        //gStyle->SetOptTitle(kFALSE);
        //gStyle->SetOptStat(kTRUE);
25     gStyle->SetLineWidth(2);
        //gStyle->SetHistLineWidth(2);

        int myFavoriteFont = 62; //Choose 132 serif or 62 sans
        float myFavoriteFontSize = 0.04; //Choose 0.04 or 0.05
30     gStyle->SetLabelFont(myFavoriteFont,"XYZ");
        gStyle->SetLabelSize(myFavoriteFontSize,"XYZ");
        gStyle->SetTitleFont(myFavoriteFont,"XYZ");
        gStyle->SetTitleSize(myFavoriteFontSize,"XYZ");
        gStyle->SetTitleYOffset(0.80);
35

        gStyle->SetPadTopMargin(0.05);
        gStyle->SetPadBottomMargin(0.12);
        gStyle->SetPadRightMargin(0.1);

```

```

    gStyle->SetPadLeftMargin    (0.1);
40  gROOT->ForceStyle();
    gStyle->SetPalette(1,0);

    for (Int_t i=1;i<5;i++){
        wccy[i]= (TH1F*) rfile ->Get(Form(" u_hits_%d",i));
45  };

    for (Int_t i=1;i<5;i++){
        wccp[i]= (TH1F*) rfile ->Get(Form(" p_hits_%d",i));
50  };

    for (Int_t i=1;i<5;i++){
        wccv[i]= (TH1F*) rfile ->Get(Form(" v_hits_%d",i));
55  };

    for (Int_t i=1;i<5;i++){
        wcc[i]= (TH1F*) rfile ->Get(Form(" xy_hits_%d",i));
60  };
    for (Int_t i=1;i<5;i++){
        wd[i]= (TH1F*) rfile ->Get(Form(" wc_status_%d",i));
        };
65

    TCanvas *c1 = new TCanvas();
        c1->Divide(2,2);
70  for (Int_t i=1;i<5;i++){
        c1->cd(i);
        wccx[i]->Draw();};
    TCanvas *c2 = new TCanvas();
        c2->Divide(2,2);
75  for (Int_t i=1;i<5;i++){
        c2->cd(i);
        wccy[i]->Draw();};
    TCanvas *c3 = new TCanvas();
        c3->Divide(2,2);
80  for (Int_t i=1;i<5;i++){
        c3->cd(i);
        wccp[i]->Draw();};

```

```
TCanvas *c4 = new TCanvas();
    c4->Divide(2,2);
85 for (Int_t i=1;i<5;i++){
    c4->cd(i);
    wccv[i]->Draw();};

TCanvas *c5 = new TCanvas();
90 c5->Divide(2,2);
for (Int_t i=1;i<5;i++){
    c5->cd(i);
    wcc[i]->Draw();};

95 int MyMaximum = 200;

TCanvas *c6 = new TCanvas();
    c6->Divide(2,2);
    for (Int_t i=1;i<5;i++){
100 c6->cd(i);
        wd[i]->SetMaximum(MyMaximum);
        wd[i]->Draw("COLZ");};

105 }
```

B.5 O algoritmo adaptado para o Event Display da simulação

B.5.1 EventDisplay.h

```

#ifndef EventDisplay_h
#define EventDisplay_h

5 #include <TROOT.h>
#include <TChain.h>
#include <TFile.h>
#include <TH2.h>
#include <TStyle.h>
10 #include <TCanvas.h>
#include <iostream>
#include <vector>
#include <fstream>
#include <TStyle.h>
15 #include <TPostScript.h>
#include "TLegend.h"
#include "TPaletteAxis.h"
#include "TGaxis.h"

20

class EventDisplay {
public :
    TTree          *fChain;    //!

```

```

Float_t      mc_part_py[20000];    //[n_mc_particles]
40  Float_t      mc_part_pz[20000];    //[n_mc_particles]
Float_t      mc_part_E[20000];    //[n_mc_particles]
Float_t      mc_part_p[20000];    //[n_mc_particles]
Int_t        n_mc_id_digits;
Int_t        mc_id_strip[20000];    //[n_mc_id_digits]
45  Int_t        mc_id_plane[20000];    //[n_mc_id_digits]
Int_t        mc_id_module[20000];    //[n_mc_id_digits]
Int_t        mc_id_view[20000];    //[n_mc_id_digits]
Float_t      mc_id_pe[20000];    //[n_mc_id_digits]
Float_t      mc_id_time[20000];    //[n_mc_id_digits]
50  Int_t        mc_id_npart[20000];    //[n_mc_id_digits]
Int_t        mc_id_particle[20000];    //[n_mc_id_digits]
Int_t        mc_id_pid[20000];    //[n_mc_id_digits]
Float_t      mc_id_part_x[20000];    //[n_mc_id_digits]
Float_t      mc_id_part_y[20000];    //[n_mc_id_digits]
55  Float_t      mc_id_part_z[20000];    //[n_mc_id_digits]
Float_t      mc_id_part_t[20000];    //[n_mc_id_digits]
Float_t      mc_id_part_p[20000];    //[n_mc_id_digits]

// List of branches
60  TBranch      *b_n_gate;    //!
TBranch      *b_n_mc_particles;    //!
TBranch      *b_mc_part_index;    //!
TBranch      *b_mc_part_pid;    //!
TBranch      *b_mc_part_type;    //!
65  TBranch      *b_mc_part_mother;    //!
TBranch      *b_mc_part_mpid;    //!
TBranch      *b_mc_part_x;    //!
TBranch      *b_mc_part_y;    //!
TBranch      *b_mc_part_z;    //!
70  TBranch      *b_mc_part_t;    //!
TBranch      *b_mc_part_px;    //!
TBranch      *b_mc_part_py;    //!
TBranch      *b_mc_part_pz;    //!
TBranch      *b_mc_part_E;    //!
75  TBranch      *b_mc_part_p;    //!
TBranch      *b_n_mc_id_digits;    //!
TBranch      *b_mc_id_strip;    //!
TBranch      *b_mc_id_plane;    //!
TBranch      *b_mc_id_module;    //!
80  TBranch      *b_mc_id_view;    //!
TBranch      *b_mc_id_pe;    //!
TBranch      *b_mc_id_time;    //!

```

```

TBranch      *b_mc_id_npart;    //!<
TBranch      *b_mc_id_particle;  //!<
85 TBranch      *b_mc_id_pid;    //!<
TBranch      *b_mc_id_part_x;    //!<
TBranch      *b_mc_id_part_y;    //!<
TBranch      *b_mc_id_part_z;    //!<
TBranch      *b_mc_id_part_t;    //!<
90 TBranch      *b_mc_id_part_p;  //!<

EventDisplay(TTree *tree=0);
virtual ~EventDisplay();
virtual Int_t   Cut(Long64_t entry);
95 virtual Int_t   GetEntry(Long64_t entry);
virtual Long64_t LoadTree(Long64_t entry);
virtual void    Init(TTree *tree);
virtual void    Loop();
virtual Bool_t  Notify();
100 virtual void   Show(Long64_t entry = -1);
virtual void    LoopEventDisplay(int first_event = 1, int num_events = 10,
    int num_modules = 40, int first_module = 1,
                                bool mc_truth = false, bool show_clusters =
                                false, bool print_files = false, bool
                                pause_between = false,
                                double Cx = 1200, double Cy = 800 );

105 virtual void   PauseEventDisplay(double pe_cut = 0, int num_modules = 40,
    int first_module = 1, bool mc_truth = false,
                                bool show_clusters = false, bool
                                print_files = false, double Cx = 1200,
                                double Cy = 800 );

void          event_display( const int gate, const int eventm, double pe_cut = 0,
    double maxPE = 35,
                                bool mc_truth = false, bool show_clusters =
                                false, bool print_files = false,
110                                int num_modules = 40, int first_module = 1,
                                TCanvas *cEventDisplay = NULL,
                                double Cx = 1200, double Cy = 800 );

virtual void   set_color_pallette();
virtual int    pause( int &eventm, double &pe_cut );

115 private:

```

```

    char* words;
    char* words2;
120
};

#endif
125
#ifdef EventDisplay_cxx
EventDisplay::EventDisplay(char *filename)
{
    TFile *f = new TFile(filename);
130    TTree *tree = (TTree*)f->Get("MTestStudies");
    words = new char[1000];
    words2 = new char[1000];

135    Init(tree);
}

EventDisplay::~EventDisplay()
{
140    if (!fChain) return;
    delete fChain->GetCurrentFile();
}

Int_t EventDisplay::GetEntry(Long64_t entry)
145 {
    // Read contents of entry.
    if (!fChain) return 0;
    return fChain->GetEntry(entry);
}

150 Long64_t EventDisplay::LoadTree(Long64_t entry)
{
    // Set the environment to read one entry
    if (!fChain) return -5;
    Long64_t centry = fChain->LoadTree(entry);
155    if (centry < 0) return centry;
    if (!fChain->InheritsFrom(TChain::Class())) return centry;
    TChain *chain = (TChain*)fChain;
    if (chain->GetTreeNumber() != fChain) {
        fChain = chain->GetTreeNumber();
160        Notify();
    }
}

```

```

    return centry;
}

165 void EventDisplay::Init(TTree *tree)
    {
    // The Init() function is called when the selector needs to initialize
    // a new tree or chain. Typically here the branch addresses and branch
    // pointers of the tree will be set.
170 // It is normally not necessary to make changes to the generated
    // code, but the routine can be extended by the user if needed.
    // Init() will be called many times when running on PROOF
    // (once per file to be processed).

175 // Set branch addresses and branch pointers
    if (!tree) return;
    fChain = tree;
    fCurrent = -1;
    fChain->SetMakeClass(1);

180 fChain->SetBranchAddress(" n_gate", &n_gate, &b_n_gate);
    fChain->SetBranchAddress(" n_mc_particles", &n_mc_particles, &b_n_mc_particles
        );
    fChain->SetBranchAddress(" mc_part_index", mc_part_index, &b_mc_part_index);
    fChain->SetBranchAddress(" mc_part_pid", mc_part_pid, &b_mc_part_pid);
185 fChain->SetBranchAddress(" mc_part_type", mc_part_type, &b_mc_part_type);
    fChain->SetBranchAddress(" mc_part_mother", mc_part_mother, &b_mc_part_mother)
        ;
    fChain->SetBranchAddress(" mc_part_mpid", mc_part_mpid, &b_mc_part_mpid);
    fChain->SetBranchAddress(" mc_part_x", mc_part_x, &b_mc_part_x);
    fChain->SetBranchAddress(" mc_part_y", mc_part_y, &b_mc_part_y);
190 fChain->SetBranchAddress(" mc_part_z", mc_part_z, &b_mc_part_z);
    fChain->SetBranchAddress(" mc_part_t", mc_part_t, &b_mc_part_t);
    fChain->SetBranchAddress(" mc_part_px", mc_part_px, &b_mc_part_px);
    fChain->SetBranchAddress(" mc_part_py", mc_part_py, &b_mc_part_py);
    fChain->SetBranchAddress(" mc_part_pz", mc_part_pz, &b_mc_part_pz);
195 fChain->SetBranchAddress(" mc_part_E", mc_part_E, &b_mc_part_E);
    fChain->SetBranchAddress(" mc_part_p", mc_part_p, &b_mc_part_p);
    fChain->SetBranchAddress(" n_mc_id_digits", &n_mc_id_digits, &b_n_mc_id_digits
        );
    fChain->SetBranchAddress(" mc_id_strip", mc_id_strip, &b_mc_id_strip);
    fChain->SetBranchAddress(" mc_id_plane", mc_id_plane, &b_mc_id_plane);
200 fChain->SetBranchAddress(" mc_id_module", mc_id_module, &b_mc_id_module);
    fChain->SetBranchAddress(" mc_id_view", mc_id_view, &b_mc_id_view);
    fChain->SetBranchAddress(" mc_id_pe", mc_id_pe, &b_mc_id_pe);

```

```

fChain->SetBranchAddress("mc_id_time", mc_id_time, &b_mc_id_time);
fChain->SetBranchAddress("mc_id_npart", mc_id_npart, &b_mc_id_npart);
205 fChain->SetBranchAddress("mc_id_particle", mc_id_particle, &b_mc_id_particle)
    ;
fChain->SetBranchAddress("mc_id_pid", mc_id_pid, &b_mc_id_pid);
fChain->SetBranchAddress("mc_id_part_x", mc_id_part_x, &b_mc_id_part_x);
fChain->SetBranchAddress("mc_id_part_y", mc_id_part_y, &b_mc_id_part_y);
fChain->SetBranchAddress("mc_id_part_z", mc_id_part_z, &b_mc_id_part_z);
210 fChain->SetBranchAddress("mc_id_part_t", mc_id_part_t, &b_mc_id_part_t);
fChain->SetBranchAddress("mc_id_part_p", mc_id_part_p, &b_mc_id_part_p);
Notify();
}

215 Bool_t EventDisplay::Notify()
{
    // The Notify() function is called when a new file is opened. This
    // can be either for a new TTree in a TChain or when when a new TTree
    // is started when using PROOF. It is normally not necessary to make changes
220 // to the generated code, but the routine can be extended by the
    // user if needed. The return value is currently not used.

    return kTRUE;
}

225 void EventDisplay::Show(Long64_t entry)
{
    // Print contents of entry.
    // If entry is not specified, print current entry
230 if (!fChain) return;
    fChain->Show(entry);
}

Int_t EventDisplay::Cut(Long64_t entry)
{
235 // This function may be called from Loop.
    // returns 1 if entry is accepted.
    // returns -1 otherwise.
    return 1;
}

240 #endif // #ifdef EventDisplay_cxx

```

B.5.2 EventDisplay.C

```

//-----
//Original Code David Schmitz
//adapted and modifications by D. Martinez for the Mtest simulation studies
//
5
#define EventDisplay_cxx
#include "EventDisplay.h"
#include <TH2.h>
#include <TStyle.h>
10 #include <TCanvas.h>
#include <limits>

//=====
// LoopEventDisplays
15 //=====

void EventDisplay::LoopEventDisplay(int first_event , int num_events , int
    num_modules , int first_module , bool mc_truth , bool show_clusters ,
    bool print_files , bool pause_between , double
    Cx , double Cy ){

    cout << "#####" << endl;
20 cout << " Event Displays" << endl;
    cout << "#####" << endl;

    set_root_env();

25 if (fChain == 0) return;

    Long64_t nentries = fChain->GetEntriesFast();
    Long64_t nbytes = 0, nb = 0;

30 int eventm = 0;
    double pe_cut = 0;

    if( show_clusters == 1 )
        TCanvas *cEventDisplay = new TCanvas( "cEventDisplay", "cEventDisplay", Cx,
            Cy );
35 else
        TCanvas *cEventDisplay = new TCanvas( "cEventDisplay", "cEventDisplay", Cx,
            Cy );

```

```

//=====
40 //-- Loop over events in NTuple
//=====

if( first_event < 1 ) return;

45 for (Long64_t jentry=0; jentry<nentries;jentry++) {

    Long64_t ientry = LoadTree(jentry);
    if (ientry < 0) break;
    nb = fChain->GetEntry(jentry);    nbytes += nb;

50    const int gatenum = n_gate; cout << n_gate << " aqui estoy " << gatenum << "
        " << n_mc_particles<<endl;
    // const int gatenum = jentry;

        event_display(gatenum, eventm, pe_cut, 35, mc_truth, show_clusters,
            print_files, num_modules, first_module, cEventDisplay );

55    if( pause_between == 1 ){
        int next = pause( eventm, pe_cut );
        if( next > 0 )
            jentry = next-2;
60    else if( next == 0 ){;}
        else
            jentry = first_event+num_events-1;
    }
}

65 }

//=====
// PauseEventDisplays
70 //=====

void EventDisplay::PauseEventDisplay(double pe_cut, int num_modules, int
    first_module, bool mc_truth, bool show_clusters,
                                bool print_files, double Cx, double Cy ){

75    cout << "#####" << endl;
    cout << " Starting MTestStudies Event Display" << endl;
    cout << "#####" << endl;

    set_root_env();

```



```

80     if ( fChain == 0 ) return ;

        Long64_t nentries = fChain->GetEntriesFast();
        Long64_t nbytes = 0, nb = 0;
85
        int eventm = -1;
        int next = 1;
        int new_eventm = 1;
        double new_pe_cut = 0;
90     char filetype [3];

        if( show_clusters == 1 )
            TCanvas *cEventDisplay = new TCanvas( "cEventDisplay", "cEventDisplay", Cx,
                Cy );
        else
95         TCanvas *cEventDisplay = new TCanvas( "cEventDisplay", "cEventDisplay", Cx,
            Cy );

//=====
//-- Loop over events in NTuple
100 //=====

        for (Long64_t jentry=0; jentry<nentries; jentry++) {

            Long64_t ientry = LoadTree(jentry);
105         if ( ientry < 0 ) break;
            nb = fChain->GetEntry(jentry);   nbytes += nb;

            const int gatenum = n_gate;
            //const int gatenum = jentry;
110
            if( next != -2 && next != -3 ){
                event_display(gatenum, eventm, pe_cut, 35, mc_truth, show_clusters,
                    print_files, num_modules, first_module, cEventDisplay );
            }

115         next = pause( new_eventm, new_pe_cut );

            if( next == -2 || next == -3 ){

                sprintf( filetype, "gif" );
120         if( next == -3 ) sprintf( filetype, "jpg" );

```

```

if( eventm < 0 || eventm > 100000 ){
    if( pe_cut == 0 )
        sprintf(words," ${HOME}/event_displays/event_display_ev%d.%s", gatenum,
            filetype );
125     else
        sprintf(words," ${HOME}/event_displays/event_display_ev%d_c%2.1f.%s",
            gatenum, pe_cut, filetype );
    }else{
        if( pe_cut == 0 )
            sprintf(words," ${HOME}/event_displays/event_display_ev%d_sl%d.%s",
                gatenum, eventm, filetype );
130     else
        sprintf(words," ${HOME}/event_displays/event_display_ev%d_sl%d_c%2.1f.%
            s", gatenum, eventm, pe_cut, filetype );
    }

    cEventDisplay->Print(words, filetype);
135

    jentry--;
}
else{

140     if( next > 0 )
        jentry = next-2;
    else if( next == 0 ){;}
    else
        jentry = nentries-1;
145

    pe_cut = new_pe_cut;
    eventm = new_eventm;
}

150 }

}

155
//=====
// event_display
//=====

```

```

160 void EventDisplay::event_display( const int gate, const int eventm, double
    pe_cut, double maxPE,
                                bool mc_truth, bool show_clusters, bool
                                print_files, int num_modules,
                                int first_module, TCanvas *cEventDisplay, double
                                Cx, double Cy ){
cout << " dentro de event_display " << gate << endl;
    if( cEventDisplay == NULL ){
165     if( show_clusters == 1 )
        cEventDisplay = new TCanvas( "cEventDisplay", "cEventDisplay", Cx, Cy );
    else
        cEventDisplay = new TCanvas( "cEventDisplay", "cEventDisplay", Cx, Cy );
    }

170     int slice_color = kBlack;
    if( eventm == 1 ) slice_color = kRed+2;
    if( eventm == 2 ) slice_color = kBlue+1;
    if( eventm == 3 ) slice_color = kGreen+2;
175     if( eventm == 4 ) slice_color = kOrange+2;
    if( eventm == 5 ) slice_color = kYellow+1;
    if( eventm > 5 ) slice_color = eventm;

    cEventDisplay->Clear();
180     cEventDisplay->Divide(1,1);

    double idheight = 0.7;

    TPad *XUV = cEventDisplay->GetPad(1);
185     XUV->SetPad(0,0.1,1,0.7);

    if( show_clusters == 1 )
        XUV->Divide(6,1);
    else
190     XUV->Divide(3,1);

    //-----
195 // ID displays
    //-----

    XUV->cd(1);
    gPad->SetTopMargin(0.013);
200 gPad->SetBottomMargin(0.015);

```

```

gPad->SetGridx(); //cout << gate << endl; sprintf(words," hola%d",1);
TH2F *stripsX = new TH2F(" stripsX"," stripsX",num_modules,first_module ,
    first_module+num_modules+1,64,0,64);
//cout << " antes de sprintf " << words << endl;
sprintf( words, " mc_id_pe*(n_gate==%d && mc_id_module>0 && mc_id_pe>%f &&
    mc_id_view==1)", gate, pe_cut);
205 //sprintf( words2," hit_strip:hit_module-%d",first_module);
//cout << " antes de sprintf2 " << gate << endl;
sprintf( words2," mc_id_strip:mc_id_module");
MTestStudies->Project(" stripsX", words2, words);
stripsX->GetXaxis()->SetTitle(" module");
210 stripsX->GetXaxis()->SetTitleSize(0.0);
stripsX->GetXaxis()->SetTitleOffset(0.6);
stripsX->GetXaxis()->SetLabelSize(0.0);
stripsX->GetYaxis()->SetTitle(" strip ");
stripsX->GetYaxis()->SetTitleSize(0.06);
215 stripsX->GetYaxis()->SetTitleOffset(1.2);
stripsX->GetYaxis()->SetLabelSize(0.045);
stripsX->GetZaxis()->SetLabelSize(0.05);
stripsX->GetZaxis()->SetRangeUser(0,maxPE);
stripsX->GetZaxis()->SetTitle(" PE ");
220 stripsX->GetZaxis()->SetTitleSize(0.07);
stripsX->GetZaxis()->SetTitleOffset(-0.9);
stripsX->DrawCopy(" col");

for( int i = 0; i < 40; i++ ){
225   if( i%8 == 0 ){
       sprintf(words,"%d",i);
       TF1 *line = new TF1(" line",words,first_module,first_module+num_modules+1);
       line->SetLineWidth(1);
       line->SetLineStyle(3);
230   line->SetLineColor(46);
       line->DrawCopy(" same");
       delete line;
   }
}

235
if( show_clusters == 1 ){
    XUV->cd(2);
    gPad->SetTopMargin(0.013);
    gPad->SetBottomMargin(0.015);
240   if( num_modules <= 8 )
       TH2F *clustersX = new TH2F(" clustersX"," clustersX
           ",350,-450,-100,2000,-1100,1100);

```

```

else if( num_modules <= 12 )
    TH2F *clustersX = new TH2F(" clustersX", " clustersX
        ",550,-450,100,2000,-1100,1100);
else if( num_modules <= 16 )
245    TH2F *clustersX = new TH2F(" clustersX", " clustersX
        ",700,-450,250,2000,-1100,1100);
else
TH2F *clustersX = new TH2F(" clustersX", " clustersX",1000, 0,1200,
    1000,-1100,1100);
if( eventm < 0 || eventm > 10000 )
    sprintf( words, " n_gate==%d && mc_id_view==1", gate );
250 else
    sprintf( words, " n_gate==%d && mc_id_view==1 && mc_id_time==%d", gate,
        eventm );
MTestStudies->Project(" clustersX", " mc_id_part_z", words );
clustersX->SetMarkerStyle(20);
clustersX->SetMarkerSize(0.7);
255 clustersX->SetMarkerColor(14);
clustersX->GetXaxis()->SetTitle("Z (mm)");
clustersX->GetXaxis()->SetTitleSize(0.09);
clustersX->GetXaxis()->SetTitleOffset(0.6);
clustersX->GetXaxis()->SetLabelSize(0.05);
260 clustersX->GetYaxis()->SetTitle("X (mm) ");
clustersX->GetYaxis()->SetTitleSize(0.08);
clustersX->GetYaxis()->SetTitleOffset(0.9);
clustersX->GetYaxis()->SetLabelSize(0.04);
clustersX->DrawCopy();
265 }

add_plot_label( "X-view", 0.40, 0.9, 0.07, 1 );
sprintf( words, " event: %d", n_gate );
add_plot_label( words, 0.2, 0.13, 0.055, 1 );
270 if( eventm < 0 || eventm > 10000 ){
    sprintf( words, " gate: %d eventm: all", gate );
    add_plot_label( words, 0.2, 0.06, 0.055, 1 );
}
else{
275    sprintf( words, " gate: %d eventm:", gate );
    add_plot_label( words, 0.2, 0.06, 0.055, 1 );
    sprintf( words, "%d", eventm );
    add_plot_label( words, 0.63, 0.06, 0.07, slice_color );
}
280 if( show_clusters == 1 )

```

```

    XUV->cd(3);
else
    XUV->cd(2);
285  gPad->SetTopMargin(0.013);
    gPad->SetBottomMargin(0.015);
    gPad->SetGridx();
    TH2F *stripsU = new TH2F("stripsU","stripsU",num_modules,first_module,
        first_module+num_modules+1,64,0,64);
    if( eventm < 0 || eventm > 10000 )
290     sprintf( words, "mc_id_pe*(n_gate==%d && mc_id_module>0 && mc_id_pe>%f &&
        mc_id_view==2)", gate, pe_cut );
else
    sprintf( words, "mc_id_pe*(n_gate==%d && mc_id_module>0 && mc_id_pe>%f &&
        mc_id_view==2)", gate, pe_cut);
//sprintf( words2,"hit_strip:hit_module-%d",first_module);
    sprintf( words2,"mc_id_strip:mc_id_module");
295  MTestStudies->Project("stripsU", words2, words);
    stripsU->GetXaxis()->SetTitle("module");
    stripsU->GetXaxis()->SetTitleSize(0.0);
    stripsU->GetXaxis()->SetTitleOffset(0.6);
    stripsU->GetXaxis()->SetLabelSize(0.0);
300  stripsU->GetYaxis()->SetTitle("strip          ");
    stripsU->GetYaxis()->SetTitleSize(0.06);
    stripsU->GetYaxis()->SetTitleOffset(1.2);
    stripsU->GetYaxis()->SetLabelSize(0.045);
    stripsU->GetZaxis()->SetLabelSize(0.05);
305  stripsU->GetZaxis()->SetRangeUser(0,maxPE);
    stripsU->GetZaxis()->SetTitle("PE          ");
    stripsU->GetZaxis()->SetTitleSize(0.07);
    stripsU->GetZaxis()->SetTitleOffset(-0.9);
    stripsU->DrawCopy("col");

310  for( int i = 0; i < 40; i++ ){
        if( i%8 == 0 ){
            sprintf(words,"%d",i);
            TF1 *line = new TF1("line",words,first_module,first_module+num_modules+1);
315            line->SetLineWidth(1);
            line->SetLineStyle(3);
            line->SetLineColor(46);
            line->DrawCopy("same");
            delete line;
320        }
    }
}

```

```

if( show_clusters == 1 ){
  XUV->cd(4);
325  gPad->SetTopMargin(0.013);
  gPad->SetBottomMargin(0.015);
  if( num_modules <= 8 )
    TH2F *clustersU = new TH2F(" clustersU ", " clustersU
      ", 400, -450, -50, 2000, -1100, 1100);
  else if( num_modules <= 12 )
330  TH2F *clustersU = new TH2F(" clustersU ", " clustersU
      ", 500, -450, 50, 2000, -1100, 1100);
  else if( num_modules <= 16 )
    TH2F *clustersU = new TH2F(" clustersU ", " clustersU
      ", 550, -450, 200, 2000, -1100, 1100);
  else
    TH2F *clustersU = new TH2F(" clustersU ", " clustersU ", 1000, 0,
      1200, 1000, -1100, 1100);
335  if( eventm < 0 || eventm > 10000 )
    sprintf( words, "(n_gate==%d && mc_id_view==2)", gate );
  else
    sprintf( words, "(n_gate==%d && mc_id_view==2 && mc_id_time==%d)", gate,
      eventm );
  MTestStudies->Project(" clustersU ", " mc_id_part_z", words );
340  clustersU->SetMarkerStyle(20);
  clustersU->SetMarkerSize(0.7);
  clustersU->SetMarkerColor(14);
  clustersU->GetXaxis()->SetTitle("Z (mm)");
  clustersU->GetXaxis()->SetTitleSize(0.09);
345  clustersU->GetXaxis()->SetTitleOffset(0.6);
  clustersU->GetXaxis()->SetLabelSize(0.05);
  clustersU->GetYaxis()->SetTitle("U (mm) ");
  clustersU->GetYaxis()->SetTitleSize(0.08);
  clustersU->GetYaxis()->SetTitleOffset(0.9);
350  clustersU->GetYaxis()->SetLabelSize(0.04);
  clustersU->DrawCopy();

}

355  add_plot_label( "U-view", 0.40, 0.9, 0.07, 1 );

if( show_clusters == 1 )
  XUV->cd(5);
else
360  XUV->cd(3);
  gPad->SetTopMargin(0.013);

```

```

gPad->SetBottomMargin(0.015);
gPad->SetGridx();
TH2F *stripsV = new TH2F("stripsV","stripsV",num_modules,first_module,
    first_module+num_modules+1,64,0,64);
365 if( eventm < 0 || eventm > 10000 )
    sprintf( words, "mc_id_pe*(n_gate==%d && mc_id_module>0 && mc_id_pe>%f &&
        mc_id_view==3)", gate, pe_cut );
else
    sprintf( words, "mc_id_pe*(n_gate==%d && mc_id_module>0 && mc_id_pe>%f &&
        mc_id_view==3 )", gate, pe_cut);
//sprintf( words2,"hit_strip:hit_module-%d",first_module);
370 sprintf( words2,"mc_id_strip:mc_id_module");
MTestStudies->Project("stripsV", words2, words);
stripsV->GetXaxis()->SetTitle("module");
stripsV->GetXaxis()->SetTitleSize(0.0);
stripsV->GetXaxis()->SetTitleOffset(0.6);
375 stripsV->GetXaxis()->SetLabelSize(0.0);
stripsV->GetYaxis()->SetTitle("strip
    ");
stripsV->GetYaxis()->SetTitleSize(0.06);
stripsV->GetYaxis()->SetTitleOffset(1.2);
stripsV->GetYaxis()->SetLabelSize(0.045);
380 stripsV->GetZaxis()->SetLabelSize(0.05);
stripsV->GetZaxis()->SetRangeUser(0,maxPE);
stripsV->GetZaxis()->SetTitle("PE
    ");
stripsV->GetZaxis()->SetTitleSize(0.07);
stripsV->GetZaxis()->SetTitleOffset(-0.9);
385 stripsV->DrawCopy("colz");

for( int i = 0; i < 40; i++ ){
    if( i%8 == 0 ){
        sprintf(words,"%d",i);
390 TF1 *line = new TF1("line",words,first_module,first_module+num_modules+1);
        line->SetLineWidth(1);
        line->SetLineStyle(3);
        line->SetLineColor(46);
        line->DrawCopy("same");
395 delete line;
    }
}

if( show_clusters == 1 ){
400 XUV->cd(6);
    gPad->SetTopMargin(0.013);
    gPad->SetBottomMargin(0.015);

```



```

    if( num_modules <= 8 )
        TH2F *clustersV = new TH2F(" clustersV", " clustersV
            ", 400, -450, -50, 2000, -1100, 1100);
405 else if( num_modules <= 12 )
        TH2F *clustersV = new TH2F(" clustersV", " clustersV
            ", 500, -450, 100, 2000, -1100, 1100);
    else if( num_modules <= 16 )
        TH2F *clustersV = new TH2F(" clustersV", " clustersV
            ", 550, -450, 250, 2000, -1100, 1100);
    else
410     TH2F *clustersV = new TH2F(" clustersV", " clustersV
            ", 1000, 0, 1200, 1000, -1100, 1100);
    if( eventm < 0 || eventm > 10000 )
        sprintf( words, "(n_gate==%d && mc_id_view==3)", gate );
    else
        sprintf( words, "(n_gate==%d && mc_id_view==3 && mc_id_time==%d)", gate,
            eventm );
415 MTestStudies->Project(" clustersV", " mc_id_part_z", words );
    clustersV->SetMarkerStyle(20);
    clustersV->SetMarkerSize(0.7);
    clustersV->SetMarkerColor(14);
    clustersV->GetXaxis()->SetTitle("Z (mm)");
420 clustersV->GetXaxis()->SetTitleSize(0.09);
    clustersV->GetXaxis()->SetTitleOffset(0.6);
    clustersV->GetXaxis()->SetLabelSize(0.05);
    clustersV->GetYaxis()->SetTitle("V (mm) ");
    clustersV->GetYaxis()->SetTitleSize(0.08);
425 clustersV->GetYaxis()->SetTitleOffset(0.9);
    clustersV->GetYaxis()->SetLabelSize(0.04);
    clustersV->DrawCopy();

}
430
add_plot_label( "V-view", 0.40, 0.9, 0.07, 1 );

if( pe_cut > 0 ){
    sprintf( words, "PE > %2.1f", pe_cut );
435 add_plot_label( words, 0.6, 0.15, 0.055, 1 );
}

if( print_files == 1 ){
    sprintf(words, "event_display_ev%d. gif", n_gate);
440 cEventDisplay->Print(words, " gif");
}

```

```

else
    cEventDisplay->Update();

445 //delete hitTime;
    delete stripsX;
    delete stripsU;
    delete stripsV;

450 if( show_clusters == 1 ){
        delete clustersX;
        delete clustersU;
        delete clustersV;
    }
455 }

//=====
// utility for easily adding Latex labels on plots
//=====

void EventDisplay::add_plot_label( char* label , double x, double y, double size ,
    int color ) {
465     TLatex *latex = new TLatex( x, y, label );
    latex->SetNDC();
    latex->SetTextSize( size );
    latex->SetTextColor( color );
470 latex->Draw();

}

//=====
475 // utility to set up basic root environment
//=====

void EventDisplay::set_root_env() {

480 //gStyle->SetPalette(1);
    gStyle->SetOptStat(0000);
    gStyle->SetOptFit(0000);
    gStyle->SetOptTitle(0);
    gStyle->SetCanvasColor(0);

```

```

485   gStyle->SetPadBorderMode(0);
      gStyle->SetFrameBorderMode(0);
      gStyle->SetCanvasBorderMode(0);
      gStyle->SetPadTopMargin(0.08);
      gStyle->SetPadBottomMargin(0.15);
490   gStyle->SetPadLeftMargin(0.15);
      gStyle->SetPadRightMargin(0.15);

      set_color_pallete();

495  }

//=====
// utility to set up color pallete
//=====

500  void EventDisplay::set_color_pallete(){

      //const Int_t NRGBs = 5;
      const Int_t NRGBs = 6;
505   const Int_t NCont = 150;

      //-- gray to red to purple scale
      Double_t stops[NRGBs] = { 0.00, 0.03, 0.12, 0.35, 0.70, 1.00 };
510   Double_t red[NRGBs] = { 0.95, 0.70, 0.87, 0.90, 1.00, 0.20 };
      Double_t green[NRGBs] = { 0.95, 0.70, 0.80, 0.50, 0.00, 0.10 };
      Double_t blue[NRGBs] = { 0.95, 0.70, 0.12, 0.00, 0.00, 0.50 };

      TColor::CreateGradientColorTable(NRGBs, stops, red, green, blue, NCont);
515   gStyle->SetNumberContours(NCont);

      }

//=====
520 // utility to pause between displays
//=====

      int EventDisplay::pause( int &eventm, double &pe_cut ){

525   const int length = 20;
      char junk[length];
      char evstr[length];
      char slstr[length];

```

```

char cutstr[length];
530 int del1 = length-1;
int del2 = length-1;
int i;

eventm = -1;
535 pe_cut = 0;

cout << "Options:" << endl;
cout << " RETURN for next spill" << endl;
cout << " Enter \'spill\' you wish to view" << endl;
540 cout << " Enter \'spill/slice\' you wish to view" << endl;
cout << " add a '-X' to any command to make a min pe cut on display" << endl;
cout << " -Ex. 131-1.5 or 250/3-2.5" << endl;
cout << " 'p' to print the current display to .gif" << endl;
cout << " 'j' to print the current display to .jpg" << endl;
545 cout << " 'q' to exit and make current display an active canvas" << endl;

cin.getline( junk, length, '\n');
cin.clear();

550
/-- loop over and find '/'
for( i = 0; i < length; i++){
    sprintf( words, "%c", junk[i] );
    if( strcmp( words, "/" ) == 0 ){
555        del1 = i;
        continue;
    }
}

560
/-- loop over and find 'c'
for( i = 0; i < length; i++){
    sprintf( words, "%c", junk[i] );
    if( strcmp( words, "-" ) == 0 ){
565        del2 = i;
        continue;
    }
}

/-- fill spill number before / or -
570 for( i = 0; i < del1 && i < del2; i++){
    evstr[i] = junk[i];
}

```

```
    //-- slice is between / and -
575   for( i = del1+1; i < del2; i++ ){
        slstr[i-(del1+1)] = junk[i];
    }

    //-- cut is after -
580   for( i = del2+1; i < length; i++ ){
        cutstr[i-(del2+1)] = junk[i];
    }

    if( evstr[0] == 'j' ) return -3;
585   if( evstr[0] == 'p' ) return -2;
    if( evstr[0] == 'q' ) return -1;

    int ev = atoi(evstr);
    if( del1 < length-1 ){
590     slice = atoi(slstr);
    }
    if( del2 < length-1 ){
        pe_cut = atof(cutstr);
    }
595   return ev;

}

600 //#####

    //
    // END
    //
    //#####

605 //
    //
```

B.6 Algoritmo do campo magnético

B.6.1 fitquadratic.cpp

```

Double_t quadratic(Double_t*x, Double_t*par)
{
    return par[0]*x[0]*x[0];
}

5
void fit_quadratic() {
    gROOT-> Reset();
    ifstream file("2Run.csv");
    //char line[128];
10    //char pmt[64];

    double Position[500];
    double Bx[500];
    double sBx[500];
15    double By[500];
    double sBy[500];
    double Bz[500];
    double sBz[500];
    int i = 0;
20

    //file.getline(line, 128);
    while(!file.eof()){

25
        file >> Position[i] >> Bx[i] >> By[i] >> Bz[i] >>sBy[i] >> sBy[i] >> sBz
            [i];
        // cout <<Position[i] << Bx[i] << By[i] << Bz[i] << endl;
        i++;

30    }
    //cout << endl;
    file.close();

35
    ///TCanvas *c1 = new TCanvas(" c1 ", " Graph Draw Options", 1400, 900 );
    ///c1->Divide(3, 1);
    ///c1->cd(1);
    TGraph *gr = new TGraph(500, Position, Bx);

```

```
40 gr->SetMarkerColor(1);
   gr->SetMarkerStyle(21);
   gr->GetXaxis()->SetTitle(" Position");
   gr->GetYaxis()->SetTitle(" B1 ( Gauss)");
   gr->GetXaxis()->CenterTitle();
45 gr->SetTitle(" B1 Vs Position");
   //gr->GetYaxis()->CenterTitle();
   gr->Draw(" AP");

       TF1 *p = new TF1(" quadratic",quadratic,0,10,1); // x in [0;10], 2
           parameters
50 //pois->SetParName(0," Const"); pois->SetParameter(0,800);
   //pois->SetParName(1," #mu"); pois->SetParameter(1,htemp->GetMean());

       // Fitting it
       gStyle->SetOptStat(0);
55 gStyle->SetOptFit(111);
       gr->Fit(p," R");
}
```

B.7 O algoritmo usado para encontrar o valor médio do ruído de fundo do campo magnetico

B.7.1 zerosaverage.py

```

#D. Martinez
import os.path
import glob
import sys
5 import string

## ----- ##
a = 0
rest = []
10 tmp = []
avx = 0
avy = 0
avz = 0

15 outfile = open('Runprobee.txt', 'w')

for fname in glob.glob('Post_firstrun_F_ZERO_10_16.csv'):
    if os.path.isfile(fname):
#        print fname
20
        infile = open(fname, 'r')
        data = infile.readlines()
        infile.close()
        a = len(data)
25 #avx = 0
#avy = 0
#avz = 0

for n in range(0, len(data)):
30 line=data[n].split()
# avx = 0
# avy = 0
# avz = 0
    date = line[0]
35 rest = line[1]

    tmp = rest.split(",")
    avx+=float(tmp[1])

```



```
    avy+=float(tmp[2])
40    avz+=float(tmp[3])

    # print avx/len(data),avy/len(data),avz/len(data)

    #s = "%s %s %s \n" % (avx,avy,avz)
45    #s = "%s %s %s \n" % (totx,toty,totz)
    #outfile.write(s)

    # print line[0], 'space',tmp[0]
totx = avx/a
50 toty = avy/a
totz = avz/a
print totx,toty,totz
s = "%s %s %s \n" % (totx,toty,totz)
outfile.write(s)
55
outfile.close()
```

B.8 O algoritmo para fazer análise preliminar dos dados do campo magnético

B.8.1 pythonanalysis.py

```

#David Martinez , Lee Patrick

#!/usr/bin/python

5 from ROOT import *

from array import *

file = open("combined")
10 lines = file.readlines()

date = array('c')
sec = array('f')
Bx = array('f')
15 By = array('f')
Bz = array('f')
temp = array('f')
rawtemp = array('l')

20 # Temp. Histogram
hist = TH1F("temperature", "Temperature", 2000, 26, 34)

for l in lines:
    p = l.strip().split(',')
25
    # Don't Really Care about Date
    # date.append(p[0].split()[0])

    # For ease of plotting convert this into an
30 # absolute second during the day
    time = p[0].split()[1]
    h, m, s = time.split(':')
    second = int(h)*3600 + int(m)*60 + float(s)
    sec.append(second)

35
    Bx.append(float(p[1]))
    By.append(float(p[2]))
    Bz.append(float(p[3]))

```

```
40     rawtemp.append( int(p[4]))
        hist.Fill( float(p[5]))

# Number of measurements
45 n = len(sec)

gr_Bx_t = TGraph(n, sec, Bx)
gr_Bx_t.SetTitle("Bx vs. t; t; Bx")

50 gr_By_t = TGraph(n, sec, By)
gr_By_t.SetTitle("By vs. t; t; By")

gr_Bz_t = TGraph(n, sec, Bz)
gr_Bz_t.SetTitle("Bz vs. t; t; Bz")
55

# Easiest to compare all three on one
# canvas (vertically)

60 c1 = TCanvas()
c1.Divide(1,3)
c1.cd(1)
gr_Bx_t.Draw("AP")
c1.cd(2)
65 gr_By_t.Draw("AP")
c1.cd(3)
gr_Bz_t.Draw("AP")

70 c2 = TCanvas()
hist.Draw()
```

B.9 Um dos algoritmos para reconstrução das trajetórias

B.9.1 TFullTrack.cxx

```

//
//
// TFullTrack
//Lee Patrick
5 // This is in a curious state... holds initial parameter
// guesses for input to fitter... the fitter will then
// fill in the track information. Lee needs to find a more
// elegant solution in the future— limited by the interface
// to TMinuit2
10
#include "TFullTrack.h"
#include "misc_funcs.h"
#include "bfield_defs.h"

15 #include <TPolyLine3D.h>
#include <TPolyLine.h>
#include <Minuit2/MnUserParameterState.h>
#include <Minuit2/FunctionMinimum.h>
#include <Minuit2/MnMachinePrecision.h>
20 #include <Minuit2/MnMigrad.h>
#include <Minuit2/MnPrint.h>

//Really needed only to plot.
#include <TCanvas.h>
25 #include <TROOT.h>
#include <TView.h>
#include <TView3D.h>
#include <TH1F.h>

30
ClassImp(TFullTrack);

TFullTrack::TFullTrack(THalfTrack in, THalfTrack out){

35   intrack=in;
   outtrack=out;

   convertmmtocm = 0.1;

```

```

convertcmtomm = 10.;
40 //errordef=1.0;

// Choose B Field here
BField_const=TestBeam::FT_BField_const;

45 stepsize=TestBeam::FT_stepsize;
nsteps=TestBeam::FT_nsteps;
ztol = TestBeam::TF_ztol;

InitialGuess();
50
PathLength = 0.0;
FieldIntegral = 0.0;
ChiSquare = 100000;

55 StraightX = -999.0;
StraightY = -999.0;

//BField=new BField_const_layer(370,484,-BField_const,0,0);
}
60
void TFullTrack::InitialGuess(void){
// Take a guess at the momentum under
// the assumption of a uniform B-Field
// and unit charge
65
//CheckStraight();

Double_t rad2deg=180./TMath::Pi();
//Double_t deg2rad=TMath::Pi()/180.;
70
TVector3 z(0,0,1);

TVector3 in =intrack.GetTrack();
TVector3 out=outtrack.GetTrack();
75
TVector3 ups_pos=intrack.GetWCDS().GetWCLocation();
TVector3 dns_pos=outtrack.GetWCUS().GetWCLocation();

Double_t zup= TestBeam::FT_zstart;
80 Double_t zdn= TestBeam::FT_zend;

Double_t th = TMath::Abs(in.Unit().Angle(z));

```

```

Double_t ph = TMath::Abs(out.Unit().Angle(z));

85  TVector3 entrypoint;
    entrypoint.SetX(ups_pos.X());
    entrypoint.SetY(ups_pos.Y()+(zup-ups_pos.Z())*TMath::Tan(th));
    entrypoint.SetZ(zup);

90  TVector3 exitpoint;
    exitpoint.SetX(dns_pos.X());
    exitpoint.SetY(dns_pos.Y()+(dns_pos.Z()-zdn)*TMath::Tan(ph));
    exitpoint.SetZ(zdn);

95  guess_cos=in.Unit();
    guess_q  =1.;

    // Have to propagate in forward from
    // WC2 to static B-Field and
100 // backward from WC3 to B-Field

    // Then need to Calculate radius , using
    // entry and exit points from static
    // B-Field

105 // for now just using WC2 and WC3
    // location

    //TVector3 entrypoint=intrack.GetWCDS().GetWCLocation();
110 //TVector3 exitpoint=outtrack.GetWCUS().GetWCLocation();

    Double_t R= (exitpoint.Y()-entrypoint.Y()) /
                (in.Unit().Y()-out.Unit().Y());

115 thetarad=out.Angle(in);
    thetadeg=thetarad*rad2deg;

    // Pay attention to units here!!!
    // p=.03BR where B=kGauss, R=meters
120 // p=.3BR where B=Tesla, R=meters
    // GeV/c
    //guess_p=.03*guess_q*BField_const*(.001*R)*1000;
    Double_t oldguess_p=0.3*guess_q*BField_const*R/1000;
    //guess_p=0.3*guess_q*BField_const*R/1000;

125 //Rik guesses. Take Pt(MeV)/sin(thetarad).

```

```

guess_p = (Double_t)0.100/(Double_t)sin(thetarad);

std::cout << "Radius Guess: " << R << std::endl;
130 std::cout << "Momentum Guess: " << guess_p << " old " << oldguess_p << " deg "
    << thetadeg << std::endl;

guess_orig = convertmmtocm*intrack.GetWCUS().GetWCHitGlobal();
135

}

void TFullTrack::PlotGuess(void){
140
    std::vector<Double_t> guess_pars;
    guess_pars.push_back(guess_q);
    guess_pars.push_back(guess_orig.X());
    guess_pars.push_back(guess_orig.Y());
145 guess_pars.push_back(guess_orig.Z());
    guess_pars.push_back(guess_cos.X());
    guess_pars.push_back(guess_cos.Y());
    guess_pars.push_back(guess_p);

150 std::cout << "FillTrack for guess " << std::endl;

    guess_track=FillTrack(guess_pars);

    TPolyLine3D *t = new TPolyLine3D(guess_track.size());
155 for (UInt_t i=0; i<guess_track.size(); i++){
        t->SetPoint(i, guess_track[i].X(),
                    guess_track[i].Y(),
                    guess_track[i].Z());
    }
160 t->SetLineColor(6);
    t->SetLineWidth(1);
    t->Draw("SAME");
}

165 void TFullTrack::Plot(void){
    TPolyLine3D *t = new TPolyLine3D(fit_track.size());
    for (UInt_t i=0; i<fit_track.size(); i++){
        t->SetPoint(i, fit_track[i].X(),
                    fit_track[i].Y(),

```

```

170         fit_track[i].Z());
    }
    t->SetLineColor(4);
    t->SetLineWidth(1);
    t->Draw("SAME");
175 }

void TFullTrack::Plot2DXZ(void){
    TPolyLine *txz = new TPolyLine(fit_track.size());
    for(UInt_t i=0; i<fit_track.size(); i++){
180     txz->SetPoint(i, fit_track[i].X(), fit_track[i].Z());
    }
    txz->SetLineColor(4);
    txz->SetLineWidth(1);
    txz->Draw("Same");
185 }

void TFullTrack::Plot2DYZ(void){
    TPolyLine *tyz = new TPolyLine(fit_track.size());
    for(UInt_t i=0; i<fit_track.size(); i++){
190     tyz->SetPoint(i, fit_track[i].Y(), fit_track[i].Z());
    }
    tyz->SetLineColor(4);
    tyz->SetLineWidth(1);
    tyz->Draw();
    }
195

void TFullTrack::PlotEvent2D(void){
    //TCanvas c1;
    //c1.Divide(1,2);
    //c1.cd(1);
200 Plot2DYZ();
    //c1.cd(2);
    Plot2DXZ();
    //c1.Update();
    // c1.WaitPrimitive();
205 }

void TFullTrack::PlotEvent(void){
    Plot();
    intrack.PlotEvent();
210 outtrack.PlotEvent();
    }

```



```

TFullTrack::~TFullTrack() {}
215 Double_t TFullTrack::GetThetaRad(void){return thetarad;}
Double_t TFullTrack::GetThetaDeg(void){return thetadeg;}

void TFullTrack::Fit(void){

220   CheckStraight();

   FitFCN.SetBField(BField);

   TWireChamber wc1=intrack.GetWCUS();

225   using namespace ROOT::Minuit2;
   //TVector3 startpos = convertmmtocm*intrack.GetWCUS().GetWCHitGlobal();

   FitFCN.AddWCPos(convertmmtocm*intrack.GetWCUS().GetWCHitGlobal());
230   FitFCN.AddWCPos(convertmmtocm*intrack.GetWCDS().GetWCHitGlobal());
   FitFCN.AddWCPos(convertmmtocm*outtrack.GetWCUS().GetWCHitGlobal());
   FitFCN.AddWCPos(convertmmtocm*outtrack.GetWCDS().GetWCHitGlobal());

   // Do it again for later, in FillTrack.
235   AddWCPos(convertmmtocm*intrack.GetWCUS().GetWCHitGlobal());
   AddWCPos(convertmmtocm*intrack.GetWCDS().GetWCHitGlobal());
   AddWCPos(convertmmtocm*outtrack.GetWCUS().GetWCHitGlobal());
   AddWCPos(convertmmtocm*outtrack.GetWCDS().GetWCHitGlobal());

240

   MnUserParameters upar;

   // Fixed Parameters for now
245   upar.Add(" charge",1.0);

   //upar.Add(" xpos", guess_orig.X(),0.1);
   //upar.Add(" ypos", guess_orig.Y(),0.1);
   //upar.Add(" zpos", guess_orig.Z(),0.1);

250   upar.Add(" xpos", guess_orig.X());
   upar.Add(" ypos", guess_orig.Y());

   // Z and Y starting positions are
255   // related by constraining them to be
   // on the plane in space defined
   // by Wire Chamber #1

```

```

upar.Add("zpos", guess_orig.Z());
260
//std::cout << "Measured Z: " << startpos.Z() << std::endl;

//Several ways I don't want to implement this right now.
// Variable with limits
265 // syntax is Add("name",value,(error?) lowerbound, upper bound
//upar.Add("cosx",guess_cos.X(),0.15,0.1,1.0);
//upar.Add("cosy",guess_cos.Y(),0.15,0.1,1.0);

// syntax for Add that fixes everything, not a free parameter.
270 //upar.Add("cosx",guess_cos.X());
//upar.Add("cosy",guess_cos.Y());

// syntax for Add that allows parameter to vary within errors.
//upar.Add("cosx",guess_cos.X(),1.0);
275 //upar.Add("cosy",guess_cos.Y(),1.0); //Lee had 0.15

// make the guess go right down the center line.
// this helps the fitter avoid areas where the B-field cuts off.
// this is not fully robust if we move WC4, of course.
280 upar.Add("cosx",0.0,0.15); // No vertical slope.
upar.Add("cosy",0.27,0.20); // Horizontal at 15 degrees.

// Variable, no limit -> should set some
upar.Add("p",guess_p,0.5*guess_p);
285

UInt_t MigradStrategy=1; //1 is default, 2 is "high" whatever.
MnMigrad migrad(FitFCN,upar,MigradStrategy);
Double_t fitTolerance = 0.1; //default is 0.1, when to stop trying harder.

290 // Rik hasn't figured out how to fix and release parameters yet.
// upar.Fix("cosx") and Release("cosx") is the wrong syntax.
// migrad.Fix("cosx") and Release("cosx") borks.

FunctionMinimum min = migrad(0,fitTolerance);
295

std::cout<<"minimum " << min << std::endl;

// Here extract any interesting information I want.
// For now, just grab the value of the minimized function.
300 // Later, add convergence and other misbehavior.
ChiSquare = min.Fval();

```

```

FitValid = (Int_t)min.IsValid();
FitMaxEDM = (Int_t)min.IsAboveMaxEdm();
FitCallLimit = (Int_t)min.HasReachedCallLimit();
305 FitValidCovar = (Int_t)min.HasValidCovariance();
FitValidParam = (Int_t)min.HasValidParameters();

for(Int_t i=0; i<7; i++){
    fitresults.push_back(min.UserParameters().Value(i));
310    fiterrors.push_back(min.UserParameters().Error(i));
}

std::cout << "FillTrack for fit " << GetFit_p() << " " << GetFit_pError() << "
    chisq " << ChiSquare
        << " valid " << (Int_t)min.IsValid()
315        << (Int_t)min.IsAboveMaxEdm()
        << (Int_t)min.HasPosDefCovar()
        << std::endl;
std::cout << "Fit results 0: "
        << fitresults[0] << " " << fiterrors[0] << " 1: "
320        << fitresults[1] << " " << fiterrors[1] << " 2: "
        << fitresults[2] << " " << fiterrors[2] << " 3: "
        << fitresults[3] << " " << fiterrors[3] << std::endl;
std::cout << "Fit results 4: " << fitresults[4] << " " << fiterrors[4] << "
    5: "
        << fitresults[5] << " " << fiterrors[5] << " 6: "
325        << fitresults[6] << " " << fiterrors[6] << std::endl;

fit_track=FillTrack(fitresults);

//Uncomment if you want to find special tracks to view.
330 //I don't bother drawing the magnets here.
//if(!FitValid){
//    PlotEvent();
//    gPad->WaitPrimitive();
//}

335 Int_t RunDangerousChisquareCheck=0;
if(RunDangerousChisquareCheck){
    // for fun, run FillTrack with values within errors of results,
    // use this to check the chisquare behavior incase you are suspicious.
340 // comment this out for production, because it will mess you up.
    std::cout << "DANGER turned on testing errors and chisquare."<< std::endl;
    std::cout << "NOT FOR PRODUCTION" << std::endl;
    std::cout << "calls to FillTrack overwrite member variables."<< std::endl;

```

```

345   for (Int_t j=0; j<21; j++){
        std::vector<Double_t> tempfitresults;
        std::vector<Double_t> tempfiterrors;
        for (Int_t i=0; i<6; i++){
            tempfitresults.push_back(min.UserParameters().Value(i));
350         tempfiterrors.push_back(min.UserParameters().Error(i));
        }
        Double_t fracerr = 0.1 * (Double_t)(j-10);
        tempfitresults.push_back(min.UserParameters().Value(6)
                                + fracerr*min.UserParameters().Error(6));
355         tempfiterrors.push_back(min.UserParameters().Error(6));

        std::vector<TVector3> temp_fit_track;

        temp_fit_track=FillTrack(tempfitresults);
360     }
    } // End use this to check the chisquare behavior incase you are suspicious.

}

365 void TFullTrack::AddWCPos(TVector3 pos){WCPos.push_back(pos);}

std::vector<TVector3> TFullTrack::FillTrack(std::vector<Double_t> par){
370     Double_t charge=par[0];
        Double_t cosz=TMath::Sqrt(1-par[4]*par[4]-par[5]*par[5]);

        //TVector3 WCPos = convertmmtocm*intrack.GetWCUS().GetWCLocation();
375     //Double_t WC1_Orient = intrack.GetWCUS().GetWCOrient();
        //WC1_Orient      = TMath::Abs(WC1_Orient);
        //Double_t zyslope = 1./TMath::Tan(WC1_Orient);
        //zyslope         = TMath::Abs(zyslope);
        //Double_t zstart = -1.*par[2]*zyslope +
380         //(WCPos.Z()+WCPos.Y()*zyslope);

        Double_t ivect [] = { par[1], par[2], par[3],
                             par[4], par[5], cosz, par[6]};
        Double_t ovect [7] = {0};

385     std::vector<TVector3> track;

```

```

// reset these private variables to default , just to be safe.
PathLength = 0.0;
390 FieldIntegral = 0.0;

// Before the loop , grab the WC4 hit z position ,
// so I can truncate the dl and bdl calculation.
// The 0.10 converts from mm to cm.
395 Double_t wc4hitz = convertmmtocm * outtrack.GetWCDS().GetWCHitGlobal().Z();
//Double_t wc1hitz = convertmmtocm * intrack.GetWCUS().GetWCHitGlobal().Z();
//std::cout << "wc4hitz " << wc4hitz << " wc1 " << wc1hitz << std::endl;

BField->GetField(par [1] , par [2] , par [3] );
400

// All the inputs and outputs are in centimeters.
// Most of the survey data is in mm, but its pedantically converted.
//std::cout << "input vector "
//   << ivect [0] << " " << ivect [1] << " " << ivect [2] << " " << ivect [3]
405 //   << " " << ivect [4] << " " << ivect [5] << " " << ivect [6] << std::endl;

// =====
// Take the input vector and backtrack to the xy plane constant z.
// this allows the PathLength variable to be a good quantity for TOF studies
410 // even though WC1 is oriented at an angle.
// Possible danger! This is approximate, add 5 cm to TGT-COLLCHANNELUPST
// Put that parameter into the .pot options file when you get a chance.
// The absolute distance might be off, but relative distances are accurate.
Double_t toflzpos = 128.5410 + 5.00;
415 Double_t thisdeltaz = toflzpos-ivect [2]; //dz, final - initial.
Double_t thisdeltax = thisdeltaz * ivect [3] / ivect [5];
Double_t thisdeltay = thisdeltaz * ivect [4] / ivect [5];
// Now this seeds the pathlength.
PathLength = TMath::Sqrt (thisdeltax*thisdeltax + thisdeltay*thisdeltay +
    thisdeltaz*thisdeltaz);
420

//std::cout << "firstpath " << PathLength << " " << toflzpos << " delta "
//   << thisdeltax << " " << thisdeltay << " " << thisdeltaz << std::endl;
// == end backtrack input vector for PathLength

425 // =====
// Declare variables for extracting the fit residuals in x and y
// For each wire chamber, get the signed x and y, plus (always +) d^2.
// But to get the signed, I need to track the minimum absolute value
// I could clean this out and save three arrays, but no harm, do it later.
430

```

```

Double_t mindist [N.CHAMBERS]={0};
Double_t minabsx [N.CHAMBERS]={0};
Double_t minabsy [N.CHAMBERS]={0};
Double_t savedx [N.CHAMBERS]={0};
435 Double_t savedy [N.CHAMBERS]={0};
Double_t distx [N.CHAMBERS]={0};
Double_t disty [N.CHAMBERS]={0};
Double_t dist [N.CHAMBERS]={0};

440 for (Int_t i=0; i<N.CHAMBERS; i++){
    mindist [i]=WCPos [i].Mag2 ();
    dist [i]=mindist [i];
    minabsx [i]=999.0;
    minabsy [i]=999.0;
445 savedx [i]=minabsx [i];
    savedy [i]=minabsy [i];
    distx [i]=minabsx [i];
    disty [i]=minabsy [i];
}

450 for (Int_t i=0; i<nsteps; i++){

    // First take a RungeKuttaStep, then calculate some stuff.
    // Here I don't bother with the adaptive step size.
455 // Maybe I should?
    RungeKuttaStep (charge, stepsize, ivect, ovect, BField);

    // Check if I should extract the position residuals yet.
    // Copied from the algorithm in the track fit algorithm.
460 for (UInt_t j=0; j<WCPos.size (); j++){
        if (TMath::Abs (WCPos [j].Z ()-ovect [2])<=ztol){
            // This calculated distance-squared, like the fitter uses.
            dist [j]=(WCPos [j].X ()-ovect [0]) *(WCPos [j].X ()-ovect [0]) +
                (WCPos [j].Y ()-ovect [1]) *(WCPos [j].Y ()-ovect [1]) +
465 (WCPos [j].Z ()-ovect [2]) *(WCPos [j].Z ()-ovect [2]);
            // These calculate the x and y distance.
            distx [j]=(WCPos [j].X ()-ovect [0]);
            disty [j]=(WCPos [j].Y ()-ovect [1]);

470 //std::cout << "zposdiff: " << j << " "
            // << TMath::Abs (WCPos [j].Z ()-ovect [2]) << " "
            // << ztol << " " << stepsize << " "
            // << WCPos [j].X () << std::endl;

```

```

475     if (mindist [j] > dist [j]) mindist [j] = dist [j];
        if (minabsx [j] > TMath::Abs (distx [j])) {
            minabsx [j] = TMath::Abs (distx [j]);
            savedx [j] = distx [j];
        }
480     if (minabsy [j] > TMath::Abs (disty [j])) {
            minabsy [j] = TMath::Abs (disty [j]);
            savedy [j] = disty [j];
        }
    }
485 }

// Here I should compute the additional quantities
// PathLength dl, FieldIntegral of b.dl

490 // Don't trust "stepsize", calculate dl explicitly. Why? Hmm.
// Get the change in position begin to end.
Double_t dx = ovect [0] - ivect [0];
Double_t dy = ovect [1] - ivect [1];
495 Double_t dz = ovect [2] - ivect [2];

// Get the midpoint position.
Double_t mx = (Double_t) 0.5 * (ovect [0] + ivect [0]);
Double_t my = (Double_t) 0.5 * (ovect [1] + ivect [1]);
500 Double_t mz = (Double_t) 0.5 * (ovect [2] + ivect [2]);

if (mz < wc4hitz) {
    // only increment up to the z location of the *hit* in wc4.
    // should be roughly 714.0 cm. TOF2 is || to WC4.
505 // treat the rest of it as starting from the origin ,
    // because TOF1 is || to the collimator , but a ~15 deg angle to WC1
    // let dl increment from zero , and subtract off ~194 cm.
    Double_t dl = TMath::Sqrt (dx*dx + dy*dy + dz*dz);
    PathLength += dl;
510 TVector3 field = BField->GetField (mx, my, mz);
    FieldIntegral += dl * field.x();
}

// Clean this out later. Used to check when we pass WC4.
515 // if (mz > wc4hitz - 1.0 && mz < wc4hitz + 1.0) {
//   std::cout << "track at WC4 " << convertmmtocm * outtrack.GetWCDS().
//     GetWCHitGlobal().X()
//   << " " << convertmmtocm * outtrack.GetWCDS().GetWCHitGlobal().Y()

```

```

//  << " " << convertmmtocm * outtrack.GetWCDS().GetWCHitGlobal().Z()
//  << " " << ovect[0] << " " << ovect[1] << " " << ovect[2] << std::endl;
520 //}

copyvect(ovect, ivect);
TVector3 pos(ovect[0], ovect[1], ovect[2]);
track.push_back(convertcmtomm*pos);
525 }

//std::cout << "output vector "
//  << ovect[0] << " " << ovect[1] << " " << ovect[2] << " " << ovect[3]
//  << " " << ovect[4] << " " << ovect[5] << " " << ovect[6] << std::endl;
530
for(UInt_t j=0; j<WCPos.size(); j++){
    FitResX[j] = savedx[j];
    FitResY[j] = savedy[j];
    FitResD2[j] = mindist[j];
535 }

std::cout << "residuals mindist ";
for(UInt_t j=0; j<WCPos.size(); j++){
    std::cout << mindist[j] << " ";
540 }
std::cout << std::endl;
std::cout << "residuals dx ";
for(UInt_t j=0; j<WCPos.size(); j++){
    std::cout << savedx[j] << " ";
545 }
std::cout << std::endl;
std::cout << "residuals dy ";
for(UInt_t j=0; j<WCPos.size(); j++){
    std::cout << savedy[j] << " " << FitResY[j] << " ";
550 }
std::cout << std::endl;

555 std::cout << "Total PathLength " << PathLength << " FieldIntegral " <<
    FieldIntegral << std::endl;
return track;
}

void TFullTrack::CheckStraight(void){
560 // checks to see if WC3 is in a straight line with WC2 and WC1 hits.

```



```

// this might be likely for high momentum particles going through,
// and indicates a robust form of accidental coincidence.
// Really it returns the residuals from such a check,
// you decide whats straight.
565
// By the way, this doesn't do what I think it should.
// Not many paths can go stright without clipping the magnet wall.

TVector3 z(0,0,1);
570
TVector3 in =intrack.GetTrack();
TVector3 out=outtrack.GetTrack();

TVector3 in_pos = convertmmtocm*intrack.GetWCUS().GetWCHitGlobal();
575 TVector3 wc3_pos = convertmmtocm*outtrack.GetWCUS().GetWCHitGlobal();

//track is a not-normalized vector between down and up.
Double_t extrapX = in_pos.X()+wc3_pos.Z()*in.X()/in.Z();
Double_t extrapY = in_pos.Y()+wc3_pos.Z()*in.Y()/in.Z();
580
//std::cout << "Check Straight at " << wc3_pos.Z() << " " << in_pos.Z()
// << " x "
// << extrapX << " " << wc3_pos.X()
// << " y " << extrapY << " " << wc3_pos.Y()
585 // << " sl " << in.X()/in.Z() << " " << in.Y()/in.Z()
// << std::endl;

// In principle , could even fit in B-field , right?

590 StraightX = extrapX - wc3_pos.X();
StraightY = extrapY - wc3_pos.Y();

return;
}
595

Double_t TFullTrack::GetGuess_p(void){return guess_p;}
Double_t TFullTrack::GetFit_p(void){return fitresults[6];}
Double_t TFullTrack::GetFit_pError(void){return fiterrors[6];}
600 Double_t TFullTrack::GetPathLength(void){return PathLength;}
Double_t TFullTrack::GetFieldIntegral(void){return FieldIntegral;}
Double_t TFullTrack::GetChisquare(void){return ChiSquare;}
Int_t TFullTrack::GetFitValid(void){return FitValid;}
Int_t TFullTrack::GetFitMaxEDM(void){return FitMaxEDM;}

```

```
605 Int_t TFullTrack::GetFitValidCovar(void){return FitValidCovar;}
Int_t TFullTrack::GetFitValidParam(void){return FitValidParam;}
Int_t TFullTrack::GetFitCallLimit(void){return FitCallLimit;}
Double_t TFullTrack::GetFitResX(Int_t whichwc){return FitResX[whichwc];}
Double_t TFullTrack::GetFitResY(Int_t whichwc){return FitResY[whichwc];}
610 Double_t TFullTrack::GetFitResD2(Int_t whichwc){return FitResD2[whichwc];}
Double_t TFullTrack::GetStraightX(void){return StraightX;}
Double_t TFullTrack::GetStraightY(void){return StraightY;}

615
void TFullTrack::SetBField(BFieldObj *theB){BField=theB;}


---


```

B.10 Algoritmo para visualização 3D das trajetórias

B.10.1 TBeamline.cxx

```

//
// TBeamLine
// Lee Patrick
// some minor modifications by D. Martinez
5
// Pretty much just draws non-data objects in the beamline

#include "TBeamLine.h"
#include "TestBeam.h"
10
#include <TPolyLine3D.h>
#include <TMarker3DBox.h>

15
ClassImp(TBeamLine);

TBeamLine::TBeamLine(void){

20   sb_st=TestBeam::BL_sb_st;
   sb_ed=TestBeam::BL_sb_ed;
   tb_st=TestBeam::BL_tb_st;
   tb_ed=TestBeam::BL_tb_ed;
   M1Pos=TestBeam::BL_M1Pos;
25   M2Pos=TestBeam::BL_M2Pos;
   targetpos=TestBeam::BL_targetpos;

}

30 void TBeamLine::Plot(void){

   // Plot Secondary Beamline Direction
   TPolyLine3D *sbeamline = new TPolyLine3D(2);
   sbeamline->SetPoint(0,sb_st.X(),
35   sb_st.Y(),
   sb_st.Z());
   sbeamline->SetPoint(1,sb_ed.X(),
   sb_ed.Y(),
   sb_ed.Z());
40   sbeamline->SetLineColor(3);

```


Bibliografia

- [1] Karagiorgi, G.A. Aguilar-Arevalo, J.M. Conrad, and M.H. Shaevitz. Leptonic CP violation studies at MiniBooNE in the (3+2) sterile neutrino oscillation hypothesis, *Phys Rev D* 75 (2007)1.
- [2] Z. Maki, M. Nakagawa, and S. Sakata, *Prog. Theor. Phys.* 28, 870 (1962).
- [3] von Bayer, O. Hahn, L. Meitner, *Phys. Zeitschrift*, 12, January, 1911, p. 378
- [4] C.D. Ellis, B.A. Wooster, The average energy of desintegration of Radium E, *Proc. Roy. Soc. A* 117(1927) 109-123
- [5] W. Pauli, carta enviada à Conferência de Tübingen, Dec. 1930.
- [6] J. Chadwick, Possible existence of a neutron, *Nature* 129 (1932)12.
- [7] E. Fermi, *Ricerca Scient.* **2** 12(1933).
- [8] F. Perrin, *Comptes rendues* **197**, 1624(1933).
- [9] E. Fermi, *Z. Physik*, **88** 161(1934).
- [10] F. Reines and C.L. Cowan, The neutrino, *Nature* 178 (1956)446.
- [11] M. Goldhaber, L. Grodzins, and A.W. Sunyar, Helicity of neutrinos, *Phys. Rev.* 109 (1958)1015.
- [12] R. Davis and D.S. Harmer, Attempt to Observe the $\text{Cl}^{37}(\bar{\nu}, e^{-})\text{Ar}^{37}$ Reaction Induced by Reactor Antineutrinos, *Bull. Am. Phys. Soc.* 4, (1959)217.
- [13] G. Danby et al., Observation of high-energy neutrino reactions and the existence of two kinds of neutrinos, *Phys. Rev. Lett.* 9 (1962)36.
- [14] Gargamelle Neutrino Collaboration: F.J. Hasert et al., Observation of neutrino-like interactions without muon or electron in the Gargamelle neutrino experiment, *Phys. Lett. B* **46** (1973)138.

- [15] R. Davis, D.S. Harner, and K.C. Hoffman, Search for neutrinos from the sun, *Phys. Rev. Lett.* **20** (1968)1205.
- [16] V.N. Gribov and B. Pontecorvo, *Phys. Lett B* **28**(1969)493.
- [17] ALEPH Collaboration: D. Decamp et al., Determination of the Number of Light Neutrino Species, *Phys. Lett. B* **231**(1989)519.
- [18] Delphi Collaboration: P.A. Aarnio et al., Measurement of the Mass and Width of the Z0 Particle from Multi-Hadronic Final States Produced in the e+ e- Annihilation, *Phys. Lett. B* **231** (1989)539.
- [19] L3 Collaboration: B. Adeva et al., A Determination of the Properties of the Neutral Intermediate Vector Boson Z0, *phys. Lett. B* **231** (1989)509.
- [20] OPAL Collaboration: M.Z. Akrawy et al., Measurement of the Z0 Mass and Width with the OPAL Detector at LEP, *Phys. Lett. B* **231** (1989)530.
- [21] DONUT Collaboration: T. Patzak, First direct observation of the tau neutrino, *Europhys. News* **32**, 56 (2001).
- [22] C.S. Wu et al., *Phys. Rev.* **105** (1957)1413.
- [23] Super-Kamiokande Collaboration: Y. Fukuda et al., Evidence for oscillation of atmospheric neutrinos, *Phys. Rev. Lett.* **81**, (1998)1562.
- [24] Soudan 2 Collaboration: M.C. Sanchez et al., Observation of atmospheric neutrino oscillations in Soudan 2, *Phys. Rev. D* **68** (2003)113004.
- [25] SNO Collaboration: Q.R. Ahmad et al., Direct evidence for neutrino flavor transformation from neutral-current interactions in the Sudbury Neutrino Observatory, *Phys. Rev. Lett.* **89**, (2002)01131.
- [26] R.N. Mohapatra e P.B. Pal, *Massive Neutrinos in Physics and Astrophysics*, World Scientific (1998)
- [27] KamLAND Collaboration: K. Eguchi et al., First results from KamLAND: Evidence for reactor anti-neutrino disappearance, *Phys. Rev. Lett.* **90**, (2003)021802.
- [28] K2K Collaboration: M.H. Ahn et al., Measurement of neutrino oscillation by the K2K experiment, *Phys. Rev. D* **74** (2006)072003.

- [29] MINOS Collaboration: D.G. Michael et al., Observation of Muon Neutrino Disappearance with the MINOS Detector in the NuMI Neutrino Beam, *Phys. Rev. Lett.* **97** (2006)191801.
- [30] <http://www-numi.fnal.gov/PublicInfo/forscientists.html>
- [31] G. Giacomelli, A. Margiotta, The MACRO Experiment at Gran Sasso, arXiv:0707.1691(2005).
- [32] M. Apollonio et al., Search for neutrino oscillations on a long base-line at the CHOOZ nuclear power station, *Eur. Phys. J. C* **27**, 331-374 (2003).
- [33] Particle Data Group, *J. Phys. G: Nucl. Part. Phys.* **33** (2010).
- [34] S. Weinberg, A Model Of Leptons, *Phys. Rev. Lett.* **19**, 1264 (1967)
- [35] A. Salam, Weak And Electromagnetic Interactions, publicado originalmente em Svartholm: Elementary Particle Theory, Proceedings Of The Nobel Symposium Held 1968 At Lerum, Sweden, Stockholm 1968, 367-377
- [36] D. Griffiths, Introduction to Elementary Particles, Wiley-VCH (2004).
- [37] P.W. Higgs, Broken Symetries And the Masses of Gauge Bosons, *Phys. Rev. Lett.* **13**, 508 (1964).
- [38] C. Quigg, Gauge Theories of the Strong, Weak and Electromagnetic Interactions, Addison Wesley Longman, Inc. (1997).
- [39] B. Kayser, Neutrino mass, mixing, and flavor change, [arXiv:hep-ph/0211134].
- [40] D. S. Ayres et al. [NOvA Collaboration], NOvA proposal to build a 30-kiloton off-axis detector to study neutrino oscillations in the Fermilab NuMI beamline, [arXiv:hep-ex/0503053].
- [41] G. Danby, J. M. Gaillard, K. Goulianos, L. M. Lederman, N. B. Mistry, M. Schwartz and J. Steinberger, Observation of high-energy neutrino reactions and the existence of two kinds of neutrinos, *Phys. Rev. Lett.* **9** (1962)36.
- [42] S. E. Kopp, Accelerator neutrino beams, *Phys. Rept.* **439** (2007)101 [arXiv:physics/0609129].
- [43] J. Hylen et al., NuMI Technical Design Handbook, Internal NuMI report (2003).
- [44] R. M. Zwaska, Accelerator Systems and Instrumentation for the NuMI Neutrino Beam, PhD thesis University of Texas at Austin, 2005.

- [45] Lipari, Lusignoli and Sartogo, Phys. Rev. Lett. **74**, (1995)4384)
- [46] Proposal to Perform a High-Statistics Neutrino Scattering Experiment Using a Fine-grained Detector in the NuMI Beam, <http://minerva.fnal.gov/proposal.pdf>
- [47] C. Amsler et al. Review of particle physics. Physics Letters B, 667:1, July 2008.
- [48] W.R. Leo, Techniques for Nuclear and Particle Physics Experiments, Springer-Verlag (1994)
- [49] H. Kluttig, J.G. Morfin e W.Van Doninck, Phys.Lett. B**71** (1977)446.
- [50] O. Erriquez et al., Phys. Lett. B **73** (1978)350.
- [51] W. Krenz et al.,Nucl. Phys. B**135** (1978)45.
- [52] M. Pohl et al., Phys. Lett. B**82** (1979)461.
- [53] W.-Y. Lee et al.,Phys. Rev. Lett. **38** (1977)202.
- [54] S.J. Barrish et al., Phys. Rev. Lett **33** (1974)448.
- [55] M. Derrick et al., Phys. Lett B **92** (1980)363.
- [56] M. Derrick et al., Phys. Lett B **23** (1981)569.
- [57] N.J. Baker et al., Phys. Rev. D **23** (1981)2495.
- [58] Pérez Lara,C.E., Design, Simulation, Commissioning and Test of the Tertiary Beam at Fermilab, Tese de Mestrado, Pontífica Universidad catolica,Lima, Peru (2010)
- [59] R.A. Burnstein et al., Hypercp: A high-rate spectrometer for the study of charged hyperon and kaon decays. Nuclear Instruments and Methods in Physics Research A, **541** (2005)516-565
- [60] J.L.P. Gallo, Testes de Hardware e Software para a Reconstrução de Trajetórias no Experimento MINERvA, tese de mestrado, CBPF (2008)