



Développement d'un algorithme de suivi de particules pour l'ILC : outils de surveillance de qualité de données en ligne

Remi Ete

► To cite this version:

Remi Ete. Développement d'un algorithme de suivi de particules pour l'ILC : outils de surveillance de qualité de données en ligne. Physique Nucléaire Expérimentale [nucl-ex]. Université de Lyon, 2017. Français. | NNT : 2017LYSE1034 |.

HAL Id: tel-01579761

<https://tel.archives-ouvertes.fr/tel-01579761>

Submitted on 31 Aug 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° d'ordre NNT : 2017LYSE1035

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
l'Université Claude Bernard Lyon 1

École Doctorale ED52
Physique et Astrophysique de Lyon (PHAST)

Spécialité de doctorat : Physique expérimentale
Discipline : Physique des particules

Soutenue publiquement le 08/03/2017, par :
Rémi Ete

**Développement d'un algorithme de suivi de
particules pour l'ILC. Outils de surveillance
de qualité de données en ligne.**

Devant le jury composé de :

Mme AUGIER Corinne, Professeur des Universités, Université Lyon 1	Présidente
M. BRIENT Jean-Claude, Directeur de recherches, LLR	Rapporteur
M. BOUMEDIÈNE Djamel, Chargé de recherches, LPC	Rapporteur
Mme FOUZ Mary-Cruz, Directrice de recherche, CIEMAT	Examinatrice
M. GIRAUD Noël, Ingénieur de recherches, CNRS	Examineur
M. LAKTINEH Imad, Professeur des Universités, Université Lyon 1	Directeur de thèse
M. MIRABITO Laurent, Chargé de recherches, CNRS	Invité

Résumé

Après la découverte au *Large Hadron Collider* (LHC) d'un boson de Higgs compatible avec celui du modèle standard, les futurs projets de collisionneurs, tels que le Collisionneur Linéaire International ILC, proposés pour succéder au LHC, sont de plus en plus étudiés. Ces projets permettront d'effectuer des mesures de précision sur cette nouvelle particule mais aussi de révéler des phénomènes encore inconnus des physiciens.

Les deux détecteurs proposés pour être placés au point de collision de l'ILC, le Grand Détecteur International ILD et le Détecteur en Silicium SiD, seront des détecteurs généralistes, conçus pour permettre l'application des algorithmes de suivi de particules. Ces algorithmes, cœur des travaux de recherche présentés dans ce manuscrit, utilisent la granularité des calorimètres et des algorithmes de reconnaissance de formes complexes pour reconstruire les particules individuellement et ainsi obtenir les résolutions souhaitées en énergie. L'algorithme de suivi de particules PandoraPFA, le plus abouti à ce jour, a montré des améliorations significatives quant aux performances physiques atteignables en physique des hautes énergies. Cependant, cet algorithme est optimisé pour un calorimètre hadronique à lecture analogique avec des cellules d'une taille de $3 \times 3 \text{ cm} \times \text{cm}$.

Le calorimètre hadronique à lecture semi-digitale SDHCAL développé essentiellement à l'IPN de Lyon, fait partie des options pour le calorimètre hadronique du détecteur ILD. Un prototype d'un mètre cube constitué de 48 chambres à plaques résistives de verre (GRPC), a été construit et testé à de maintes reprises sur les lignes de faisceau au CERN. Mes travaux de recherches ont porté sur le développement d'un algorithme de suivi de particules dédié à ce calorimètre. Le principe de l'algorithme repose sur le développement des gerbes hadroniques qui a une topologie en arbre orienté. L'algorithme connecte les dépôts d'énergie dans les calorimètres en vue de reconstruire au mieux cette structure en arbre.

Afin de tester ce principe, une première implémentation a été développée pour le prototype physique du SDHCAL. La reconstruction des hadrons isolés est d'abord étudiée puis celle de deux hadrons proches afin de comprendre les performances de reconstruction lorsque deux particules sont proches. Dans ces deux études, les données récoltées lors des campagnes de test sur faisceau et celles générées par la simulation numérique du prototype utilisant Geant4 sont comparées. Une deuxième version a finalement été développée pour le détecteur ILD en tenant compte cette fois-ci de tous les sous-détecteurs. Des échantillons de simulation de boson Z^0 produisant deux jets issus d'une paire $q\bar{q}$ sont utilisés afin d'extraire les performances de cette implémentation.

En parallèle de ces travaux, un logiciel de surveillance de données en ligne nommé DQM4HEP a été développé. Un effort particulier a été mis sur la généralité du logiciel, en particulier l'indépendance du système vis-à-vis du type d'événement physique à traiter. Cette fonctionnalité permet au logiciel d'être réutilisable dans d'autres expériences. Celui-ci permet aussi l'inclusion d'analyses externes, que le physicien développe lui-même au sein d'une librairie. C'est dans ce cadre que des analyses spécifiques au SDHCAL ont été développées avec le format de données spécifique appelé LCIO et déployées lors de plusieurs tests sur faisceau, notamment lors d'un test sur faisceau combiné avec le calorimètre électromagnétique SiWECal proposé également pour ILD.

La prochaine étape concernant l'algorithme de suivi de particules sera de continuer d'améliorer les performances physiques. Concernant le logiciel de surveillance de données, son développement sera poursuivi afin d'être utilisé par d'autres expériences.

Remerciements

Je voudrais tout d'abord remercier mon directeur de thèse, Imad Laktineh, pour m'avoir offert ce sujet sur les techniques de PFA. Merci pour ton accompagnement qui n'a pas dû être simple étant donné un projet professionnel très variable au cours de cette thèse.

Mes seconds remerciements vont à Laurent, avec qui des discussions très éducatives ont permis de mettre en place le système de monitoring du SDHCAL. J'ai énormément appris en ta compagnie.

J'aimerais ensuite remercier mon jury de thèse : Corinne, pour avoir présidé dans une ambiance des plus détendues (merci aussi pour la correction autographe du manuscrit), les rapporteurs, Djamel et Jean-Claude, pour le temps qu'ils ont investi sur mon manuscrit et les examinateurs, Mary-Cruz, Noël et Laurent, pour leurs questions et remarques pertinentes durant la soutenance.

Je remercie le groupe ILC de l'IPNL dans lequel j'ai effectué mes travaux : Gérald pour tes discussions (souvent autour d'un café) et ta bonne humeur, Arnaud pour les concours de nommage de variables ridicules (ma préférée reste "G4Track gLeTrack;"), Antoine pour le partage qu'on a eu sur le DQM et la passion du gras, Guillaume pour ton humour plus que douteux qui a su égayer les longues nuits de testbeam ensemble (bonne chance pour la suite). Je remercie en plus du groupe ILC, Francois et Victor, pour les nuits de testbeam qui resteront, je pense, les meilleurs souvenirs de ma thèse à l'IPNL.

Je remercie ma famille qui a été là pendant les moments difficiles de cette thèse : mon père, ma mère, mes frères et mes beaux-parents, Paulo et Fred. Un remerciement spécial pour mon père pour la relecture de ce manuscrit et Paulo pour les virées en moto qui ont su me faire oublier les tracas du quotidien. Un grand merci à toute la famille pour l'organisation de notre mariage en Provence.

Je souhaiterais remercier tout particulièrement deux personnes qui ont grandement contribué à mon avancée dans mes études et ma vie professionnelle : Giampi et Stéphane. Vous m'avez tout deux offert l'opportunité de rentrer dans le milieu de la physique des particules au CPPM et au CERN, opportunité que j'ai su saisir au bon moment. Vous êtes et resterez deux personnes importantes pour moi.

Merci également à Fabienne pour ton temps et tes conseils en insertion professionnelle qui ont été plus que fructueux.

Mes derniers et plus importants remerciements vont à Marla, ma femme, qui m'a toujours soutenu pendant les moments difficiles. De longues années nous attendent tous les deux et je les sais d'avance remplies de bonheur étant données les 5 années que l'on a déjà passé ensemble.

Merci à tous.

Table des matières

1	Le contexte théorique	1
1	Le modèle standard de la physique des particules	2
2	Les particules élémentaires et leurs interactions	3
3	Mécanisme de brisure de symétrie	6
4	Au delà du modèle standard	7
2	Le collisionneur linéaire international ILC	9
1	Motivations et programme de physique de l'ILC	10
1.1	Motivation d'un collisionneur leptonique linéaire	10
1.2	Programme de physique de l'ILC	13
2	Le collisionneur linéaire international ILC	15
2.1	Le complexe d'accélérateur	15
2.2	Les détecteurs au point de collision	17
2.2.1	Le Grand Détecteur International ILD	19
2.2.2	Le Détecteur au Silicium SiD	28
3	Les autres projets de collisionneurs leptoniques	29
3	Le calorimètre hadronique semi-digital SDHCAL	31
1	Le phénomène de gerbe électromagnétique et hadronique	32
1.1	Interactions des particules dans la matière	32
1.2	Les gerbes électromagnétiques	34
1.3	Les gerbes hadroniques	34
2	La calorimétrie en physique des particules	36
3	Le prototype du SDHCAL	38
3.1	Description du prototype	38
3.2	Les chambres à plaques résistives	39
3.3	L'électronique de lecture	41
3.4	Le système d'acquisition	42
3.5	Simulation numérique du prototype	43
4	Les performances du SDHCAL	44
4.1	Reconstruction des événements physiques	45
4.2	Les performances du détecteur	46
4.2.1	Sélections de muons	46
4.2.2	Efficacité et multiplicité	49
4.3	Reconstruction de l'énergie des hadrons	49
4.3.1	Sélection des gerbes hadroniques	50

4.3.2	Calibration en fonction du temps	53
4.3.3	Estimation de l'énergie des hadrons	55
4.4	Des désaccords entre données et simulation	61
5	Conclusion	62
4	Les algorithmes de suivi de particules	65
1	Principe des algorithmes de suivi de particules	66
2	L'algorithme PandoraPFA	68
2.1	L'environnement de développement PandoraSDK	68
2.2	La fonction de calcul des pseudo-plans	70
2.3	Les algorithmes	71
2.4	Les performances	76
2.5	Les limites de PandoraPFA	77
3	ArborPFA : il était une fois, des arbres	79
5	L'algorithme ArborPFA pour le prototype du SDHCAL	83
1	Description de l'algorithme	84
2	Etude des hadrons isolés	93
2.1	Préparation des événements	93
2.2	Analyse des performances	93
3	Etude de séparation de deux hadrons proches	98
3.1	Superposition des événements	98
3.2	Analyse des performances	100
4	Les incertitudes systématiques	104
5	Conclusion	107
6	L'algorithme ArborPFA pour le détecteur ILD	109
1	Calibration en énergie des hadrons isolés	110
1.1	Calibration initiale	110
1.2	Correction inter-module	113
1.3	Correction angulaire	116
2	Les fonctions et les algorithmes ArborPFA	123
2.1	La fonction de calcul du champs magnétique	123
2.2	La fonction de calcul des pseudo-plans	124
2.3	Les fonctions d'identification de particules	124
2.4	Les fonctions de correction en énergie	126
2.5	L'ordonnancement des algorithmes	127
2.6	Les algorithmes de reconstruction	132
3	Les performances physiques de l'algorithme	147
4	Conclusion	151
7	Logiciel de surveillance de qualité de données en ligne	153
1	Introduction aux systèmes de surveillance en physique des hautes énergies	154
2	DQM4HEP : un environnement générique de surveillance de données en ligne	155

2.1	Vue d'ensemble du logiciel	155
2.2	Gestionnaire de <i>runs</i>	159
2.3	Lien vers les sources de données	160
2.4	Collecteur d'événements physiques	162
2.5	Modules d'analyse de données	164
2.5.1	Gestion des éléments de surveillance	167
2.5.2	La structure en cycle	168
2.6	Collecteurs d'éléments de surveillance	170
2.7	Les interfaces graphiques	171
2.7.1	Gestion des <i>runs</i>	171
2.7.2	Gestion des processus à distance	171
2.7.3	Interface de surveillance des données	174
3	Implémentation pour le prototype SDHCAL	177
3.1	Combinaison des détecteurs SiWECal et SDHCAL	177
3.2	Source de données	178
3.3	Les modules d'analyse	179
3.4	Déploiement du système et performances	184
4	Conclusion	188
Appendices		191
A Paramètres de l'algorithme ArborPFA pour le prototype SDHCAL		191
B Gestion des <i>plugins</i>		197
C Implémentation d'un module d'analyse de données		201
1	Le contexte : détection de muons cosmiques	201
2	Transfert des événements	202
3	Le module d'analyse	204
4	La configuration XML	206
Bibliographie		213
Index		215

Liste des figures

1.1	Les particules élémentaires du modèle standard réparties en familles et générations distinctes	2
1.2	Désintégration β^- mettant en jeu la production d'un anti-neutrino électronique	3
1.3	Production d'une paire quark anti-quark suite à une collision $e^+e^- \rightarrow q\bar{q}$. Une paire de <i>jets</i> est créée suite au processus d'hadronisation des quarks.	5
1.4	Diagramme de Feynman d'un échange de couleurs entre quarks par l'intermédiaire d'un gluon.	5
2.1	Comparaison des précisions attendues sur les facteurs κ_γ et κ_W au LHC, à l'ILC, au TLEP et à CLIC pour différentes versions des collisionneurs [6].	12
2.2	a) Diagramme de Feynman de la fusion Z^0 . b) Diagramme de Feynman de la fusion W^\pm . c) Diagramme de Feynman du <i>Higgsstrahlung</i> . d) Différentes sections efficaces impliquant un boson de Higgs dans l'état final [6].	14
2.3	Schéma du complexe d'accélération du collisionneur linéaire international [9].	15
2.4	Récapitulatif des paramètres de la machine ILC pour les trois énergies stantardes et l'amélioration à 1 <i>TeV</i> [4].	16
2.5	Schéma représentant le mécanisme de <i>push-pull</i> permettant d'intervertir les détecteurs ILD (à gauche) et SiD (à droite) au point de collision [3].	18
2.6	À gauche : vue globale du détecteur ILD. À droite : vue en coupe du détecteur ILD, détaillant les différents sous-détecteurs et certaines côtes [3].	19
2.7	Schéma illustrant la chambre à projection temporelle du détecteur ILD.	21
2.8	Schéma illustrant les différents types de calorimètres électromagnétiques et hadroniques éligibles à l'ILC.	24
2.9	Schéma illustrant les deux structures géométriques proposées pour le calorimètre hadronique dans la région du tonneau : la géométrie TESLA (à gauche) et la géométrie VIDEAU (à droite).	26
2.10	Schéma illustrant la structure des chambres à muons dans le tonneau (à gauche) et dans les bouchons (à droite) dans le détecteur ILD.	27
2.11	Le détecteur SiD. À gauche : vue globale du détecteur. À droite : vue coupe du détecteur ; code couleur : trajectographe (rouge), ECAL (vert), HCAL (violet) et chambres à muons (bleu.)	28

3.1	À gauche : Courbes de perte d'énergie pour différents matériaux traversés par des muons, des pions et des protons en fonction de leur impulsion. À droite : Courbes de perte d'énergie des électrons et des positons pour différents processus dans le plomb [38].	32
3.2	Sections efficaces de différents processus dans le plomb (à gauche) et le carbone (à droite) en fonction de l'énergie incidente du photon : $\sigma_{p.e}$ pour l'effet photoélectrique, $\sigma_{Compton}$ pour la diffusion Compton, $\sigma_{Rayleigh}$ pour la diffusion Rayleigh, κ_{nuc} pour l'interaction photonucléaire et κ_e pour la création de paires [38].	33
3.3	Schéma illustrant une gerbe électromagnétique initiée par un électron de haute énergie.	34
3.4	Schéma illustrant une gerbe hadronique. On y trouve quatre composantes énergétiques : l'énergie électromagnétique (rouge), l'énergie non-électromagnétique (noire), l'énergie invisible (verte) et l'énergie indétectable (bleue).	35
3.5	Photographie du prototype SDHCAL prise lors d'un test sur faisceau sur la ligne H2 du CERN.	38
3.6	Une gerbe hadronique d'un pion de 80 GeV enregistrée au SPS sur la ligne H6 au CERN, par le prototype SDHCAL. Les couleurs correspondent au seuils : premier seuil en vert, second seuil en bleu et troisième seuil en rouge. Le faisceau est dirigé suivant l'axe \vec{z}	39
3.7	Vue en coupe d'une chambre à plaques résistives en verre. Les dimensions sont celles utilisées par le prototype SDHCAL [8].	40
3.8	Photographie des carreaux en cuivre utilisée lors du développement de petites GRPCs. Ces carreaux sont imprimés sur l'autre face du circuit de lecture [8].	40
3.9	Efficacité des chambres 12 (a) et 33 (b) en fonction de la haute tension appliquée. Le plateau d'efficacité est atteint autour de 6.8 kV [8].	41
3.10	Modèles de simulation de la physique hadronique utilisés dans GEANT4 en fonction de l'énergie des particules.	44
3.11	(a) Distribution du temps des hits dans un événement RAMfull enregistré par le prototype du SDHCAL. Les bins verts correspondent aux événements physiques candidats. (b) Distribution du nombre de hits correspondant à du bruit dans un bin en temps (1 bin = 200 ns). Ces hits sont ceux qui ne sont pas sélectionnés par l'algorithme de groupement en temps <i>Trivent</i> (en bleu sur la figure (a)) [8].	45
3.12	Événement d'une gerbe hadronique enregistré dans le prototype du SDHCAL comportant des "trous" dans certains plans. Cet effet est corrigé lors de la reconstruction des événements physiques en agrandissant la fenêtre de temps d'un bin en temps supplémentaire.	46
3.13	Rapport transverse λ_{tr} en fonction du nombre de hits pour un échantillon de données à 30 GeV.	48
3.14	Distribution du nombre total de hits dans le calorimètre après sélection des muons pour une prise de données à 30 GeV.	48
3.15	(a) Efficacité et (b) multiplicité du détecteur, plan par plan [20].	49

3.16	Distribution du nombre total de hits dans le calorimètre après la suppression des muons pour une prise de données de pions de 30 <i>GeV</i> [16].	51
3.17	(a) Distribution du nombre de traces reconstruites (b) Distribution du premier plan d'interaction (c) Distribution du nombre de plans touchés. L'échantillon de données est issu d'une simulation d'électrons de 50 <i>GeV</i> [43].	51
3.18	Distribution du nombre de hits pour des échantillons de données de test sur faisceau de pions de (a) 10 <i>GeV</i> et (b) 50 <i>GeV</i> avant toutes coupures (en noir), après la suppression des muons (en rouge) et après suppression des gerbes électromagnétiques (en bleu) [43].	52
3.19	Nombre moyen de hits du troisième seuil pour des pions de 80 <i>GeV</i> en fonction du temps depuis le début du spill (a) avant et (b) après la correction temporelle. La droite en rouge est le résultat de l'ajustement d'un polynôme d'ordre 2 : $N(t) = p_0 + p_1.t + p_2.t^2$ [16].	54
3.20	Distributions du nombre de hits pour (a) le premier seuil, (b) le deuxième seuil et (c) le troisième seuil pour des pions de 70 <i>GeV</i> avant correction (en noir) et après correction (en rouge) en fonction du temps relatif au début du spill [43].	54
3.21	(a) Nombre de hits moyen en fonction de l'énergie du faisceau sur la ligne H6 du CERN. La ligne en noire est un ajustement linéaire effectué entre 5 et 20 <i>GeV</i> et la courbe en pointillé, son prolongement. (b) Déviation relative du nombre moyen de hits à l'ajustement linéaire en fonction de l'énergie du faisceau [16].	56
3.22	Distribution de l'énergie reconstruite des gerbes hadroniques pour une énergie de faisceau de (a) 20 <i>GeV</i> et (b) 40 <i>GeV</i> , avec l'estimateur en énergie binaire de l'équation 3.9 [43].	57
3.23	Énergie reconstruite moyenne (à gauche) et résolution relative (à droite) de l'énergie reconstruite des gerbes hadroniques en fonction de l'énergie du faisceau sur la ligne H6 du CERN. L'énergie reconstruite est estimée en utilisant l'information sur le nombre total de hits uniquement (équation 3.9) [16].	58
3.24	Nombre de hits moyen pour chacun des trois seuils du prototype du SDHCAL en fonction de l'énergie du faisceau. Les points en noir montrent la somme des trois contributions, soit le nombre de hits total moyen [16].	59
3.25	Évolution des trois paramètres α , β et γ de l'équation 3.13 en fonction du nombre total de hits après minimisation [15].	59
3.26	Énergie reconstruite moyenne (à gauche) et résolution relative (à droite) de l'énergie reconstruite des gerbes hadroniques en fonction de l'énergie du faisceau sur la ligne H6 du CERN. L'énergie reconstruite est estimée en utilisant l'information sur les trois seuils (équation 3.13) [16].	60
3.27	Résolution relative en énergie pour les modes binaires et multi-seuils du SDHCAL [16].	61

3.28	Nombre de hits moyen en fonction de l'énergie du faisceau pour les données (croix noires), (a) les listes physiques FTFP_BERT(_HP), QGSP_BERT(_HP) et (b) les listes physiques FTF_BIC, QGSP_FTFP_BERT et QGSP_BIC [43].	62
3.29	Valeur moyenne et déviation relative du profil latéral en fonction de l'énergie du faisceau pour les données (croix noires) et plusieurs listes physiques. La figure 3.29(a) montre aussi des résultats pour des protons [43].	63
4.1	Schémas illustrant la mesure de l'énergie des jets en exploitant a) la granularité des détecteurs et b) les algorithmes de reconnaissance de formes.	66
4.2	69
4.3	Découpage de la géométrie du détecteur en pseudo-plans dans a) la région du tonneau et b) la région des bouchons. Les lignes pleines indiquent les plans physiques du calorimètre électromagnétique et les lignes pointillées les pseudo-plans.	70
4.4	a) Les quatre formes de traces traitées en amont par l'algorithme PandoraPFA. b) Schéma illustrant le principe de l'algorithme principal de clustering de PandoraPFA. c) Principales formes d'associations topologiques entre clusters dans PandoraPFA.	72
4.5	d) e) et f) : Trois exemples de stratégies de re-clustering opérées par PandoraPFA ; g) Exemples de topologies traités par les algorithmes de suppression de fragments par PandoraPFA.	73
4.6	Performances de l'algorithme PandoraPFA sur des événements di-jets <i>uds</i> dans le modèle ILD_o1_v05 (AHCAL) : a) la résolution en énergie des jets en fonction du cosinus de l'angle entre un jet et l'axe du faisceau et b) la résolution en énergie des jets en fonction de l'énergie des jets.	77
4.7	Performances de l'algorithme PandoraPFA sur des événements di-jets <i>uds</i> dans le modèle ILD_o2_v05 (SDHCAL) : résolution en énergie des jets en fonction du cosinus de l'angle entre un jet et l'axe du faisceau	78
4.8	(a) Illustration du développement d'une gerbe hadronique initiée par un proton. (b) La même gerbe hadronique telle qu'elle pourrait être détectée dans un calorimètre à échantillonnage. Les flèches suggèrent une reconstruction des gerbes hadroniques suivant une topologie en arbre orientée.	79
5.1	(a) Exemple de groupement de hits (encadrés en vert) par plus proches voisins dans chaque plan physique, ici pour un plan particulier. (b) Exemple de séparation des groupes de hits et de création de vertex (encadrés en vert) à partir des groupes de hits de la figure (a).	84

5.2	Résultat de l'algorithme d'étiquetage des vertex appartenant au segment de traces. Les vertex étiquetés, en rouge, seront traités de manière particulière lors de la phase de création de clusters, notamment ceux appartenant aux segments de traces primaires, comme ceux entourés en noir.	85
5.3	(a) Zone locale de sélection des vertex étiquetés comme appartenant au segment de trace. (b) Résultat final de la connexion des vertex des segments de traces primaires.	86
5.4	(a) Résultat possible après l'exécution de l'algorithme de <i>création des connexions 1</i> . (b) Résultat possible après l'exécution de l'algorithme de <i>nettoyage des connexions 1</i>	87
5.5	Exemple d'application de l'algorithme de création de connexions 2. La connexion considérée est celle en vert. Deux angles solides vers l'avant et vers l'arrière sont utilisés pour créer de nouvelles connexions. Ces dernières sont schématisées en rouge sur la dernière étape à droite. . .	88
5.6	Exemple d'association entre clusters et traces. Deux cas sont considérés : 1) les particules avec pré-interactions et 2) les particules interagissant dans le calorimètre.	90
5.7	Exemple d'une interaction d'une particule neutre donnant naissance à plusieurs arbres dont les vertex racines sont proches (arbres vert, bleu et rouge). Un algorithme identifie ce type de topologie et assure la fusion de ces arbres au sein d'un même cluster.	90
5.8	Exemple de reconstruction finale d'une particule neutre (en bleu) et d'une particule chargée (en rouge) par l'algorithme ArborPFA dédié au SDHCAL.	92
5.9	Gerbe hadronique d'un pion chargé de 50 GeV dans le prototype du SDHCAL enregistré sur la H6 du SPS au CERN.	93
5.10	(a) Efficacité moyenne des hadrons isolés après reconstruction avec ArborPFA pour des données issues du test sur faisceau et de la simulation Geant4 pour plusieurs listes de physique. (b) Nombre moyen de hits du PFO chargé reconstruit par l'algorithme ArborPFA pour des données issues du test sur faisceau et de la simulation Geant4 pour les listes de physique FTFP_BERT_HP et FTF_BIC	94
5.11	Nombre moyen de PFOs reconstruits par l'algorithme ArborPFA pour des données issues du test sur faisceau et de la simulation Geant4 pour plusieurs listes de physique.	94
5.12	Distribution du nombre de PFOs pour des pions chargés (a) à 20 GeV et (b) à 70 GeV pour des données issues du test sur faisceau et de la simulation Geant4 pour plusieurs listes de physique.	95
5.13	Distribution du nombre de hits des PFOs reconstruits (a) à 20 GeV et (b) à 70 GeV.	95
5.14	Énergie reconstruite avant reconstruction (No PFA) et après reconstruction en utilisant les hits du PFO chargé pour (a) les données issues du test sur faisceau et (b) les données issues de la simulation Geant4 pour différentes simulations physiques (FTFP_BERT_HP et FTF_BIC). . .	96

5.15	Résolution en énergie avant reconstruction (No PFA) et après reconstruction en utilisant les hits du PFO chargé pour (a) les données issues du test sur faisceau et (b) les données issues de la simulation Geant4 pour différentes simulations physiques (FTFP_BERT_HP et FTF_BIC).	96
5.16	(b) Énergie reconstruite et (b) résolution en énergie après reconstruction en utilisant les hits du PFO chargé pour les données issues du test sur faisceau et les données issues de la simulation Geant4 pour différentes simulations physiques (FTFP_BERT_HP et FTF_BIC).	97
5.17	Distribution de l'énergie reconstruite du hadron neutre de 10 GeV avant superposition (en bleu) et après superposition avec un hadron chargé de 30 GeV pour une distance de séparation de 5 cm (en rouge) et 30 cm (en noir) pour (a) les données du test sur faisceau et (b) la simulation Geant4 avec la liste de physique FTF_BIC	99
5.18	Distribution du nombre de hits superposés pour un hadron neutre de 10 GeV et un hadron chargé de 30 GeV pour une distance de séparation de (a) 30 cm et (b) 5 cm.	101
5.19	Distribution du pourcentage du nombre total de hits superposés pour un hadron neutre de 10 GeV et un hadron chargé de 30 GeV pour une distance de séparation de (a) 30 cm et (b) 5 cm.	101
5.20	Nombre moyen de PFOs reconstruits par l'algorithme ArborPFA pour des données superposées issues de test sur faisceau et de la simulation Geant4 pour la liste de physique FTF_BIC	102
5.21	(a) Efficacité et (b) pureté du hadron neutre de 10 GeV après reconstruction.	102
5.22	(a) Probabilité de reconstruire au moins une particule neutre avec l'algorithme ArborPFA. (b) La différence moyenne entre l'énergie du hadron neutre après reconstruction et avant reconstruction pour les événements présentant au moins une particule neutre.	103
5.23	Différence moyenne entre l'énergie du hadron neutre avant et après reconstruction.	104
6.1	(a) Énergie reconstruite et (b) résolution en énergie des kaons neutres K_0^L après la calibration initiale à angles fixes ($\phi = 0$, $\theta = 1.5 \text{ rad}$).	112
6.2	Schéma illustrant le découpage du détecteur ILD en quatre régions : la zone 1) détectera les particules uniquement dans le tonneau central, la zone 2) dans le tonneau central et les bouchons, la zone 3) uniquement dans les bouchons et enfin la zone 4) dans les bouchons et les calorimètres à l'avant.	113
6.3	Schéma illustrant la séparation entre deux modules du calorimètre hadronique dans le tonneau central. Les cellules en rouge correspondent à celles utilisées lors de la correction en énergie inter-module.	114
6.4	(a) Énergie reconstruite et (b) résolution en énergie des kaons neutres K_0^L utilisant l'estimateur 6.1 en fonction de $\cos\theta$	114

6.5	Énergie moyenne déposée près des interstices inter-module dans le calorimètre hadronique dans la région du tonneau centrale en fonction de $\cos\theta$	115
6.6	(a) Énergie reconstruite et (b) résolution en énergie des kaons neutres K_0^L après la correction sur les interstices inter-module.	116
6.7	(a) Énergie reconstruite de kaons neutres K_0^L simulés en fonction de $\cos(\theta)$ et les ajustements linéaires dans les régions 1 et 3. (b) Évolution des paramètres e_0 , e_1 , b_0 et b_1 après ajustement des courbes en énergie de la figure (a).	118
6.8	Évolution des paramètres e_0 , e_1 , b_0 et b_1 après ajustement des courbes en énergie de la figure 6.7(a). Chacune des courbes associée aux quatre paramètres est ajustée avec (a) un polynôme d'ordre 1 et (b) un polynôme d'ordre 2 (voir table 6.1).	118
6.9	Fraction de l'énergie totale déposée par des kaons neutres K_0^L dans (a) le tonneau central et (b) les bouchons, en fonction de $\cos\theta$	119
6.10	Énergie reconstruite (en haut) et résolution en énergie (en bas) des kaons neutres K_0^L simulés en fonction de $\cos(\theta)$ après application de la correction inter-modules suivi de la correction angulaire en utilisant le polynôme d'ordre 1 (à gauche) et le polynôme d'ordre 2.	120
6.11	Ordonnancement des sous-algorithmes de reconstruction. Cinq phases se distinguent : i) préparation de l'événement, ii) reconstruction des photons, iii) <i>clustering</i> principal, iv) <i>re-clustering</i> et v) préparation et création des PFOs.	130
6.12	Schéma illustrant la stratégie de <i>split-and-merge</i> déployée avec les deux premiers algorithmes de <i>re-clustering</i> . Le cluster chargé de 12 <i>GeV</i> (en noir) est d'abord re-configuré pour obtenir un cluster neutre de 7 <i>GeV</i> . Le cluster chargé de 1 <i>GeV</i> (en rouge) est ensuite re-configuré et englobe cette fois-ci le cluster neutre de 7 <i>GeV</i>	131
6.13	Schéma illustrant les différentes étapes exécutées par l'outil de connexion des hits utilisant les traces issues du trajectographe comme référence.	133
6.14	Schéma illustrant la connexion des hits sans utiliser les traces issues du trajectographe.	134
6.15	Schéma illustrant la fusion des arbres au racines proches. Deux arbres ayant deux racines dans des pseudo-plans proches mais différents peuvent être fusionnés.	135
6.16	Schéma illustrant les variables topologiques mises en jeu lors de l'association entre traces et clusters. a) Sélection des clusters b) calcul du barycentre interne d'un cluster par rapport à une trace c) exemple de deux rejets et d'une compatibilité entre un trace et un cluster.	136
6.17	a) Les deux problèmes de topologie traités par l'algorithme d'association des fragments de traces, b) variables et coupures mises en jeu dans le premier problème de topologie (mip fragment) et c) variables et coupures mises en jeu dans le deuxième problème de topologie (broken track).	138

6.18	a) Quelques exemples de fragments formés après une phase de <i>clustering</i> et b) suppression des fragments et répartition des hits dans les clusters voisins.	140
6.19	a) Exemple d'association possible entre un cluster neutre et un cluster parent chargé. b) Exemple d'association impossible à cause d'un angle entre clusters trop important. c) Exemple d'association impossible à cause d'une différence en pseudo-plan invalide (trop importante ou négative). d) Exemple d'association impossible à cause d'une incompatibilité $E - p$	141
6.20	Exemples de problèmes topologiques traités par les algorithmes de <i>re-clustering</i> : a) <i>re-clustering</i> lors d'un excès d'énergie, b) <i>re-clustering</i> lors d'un déficit en énergie, c) <i>re-clustering</i> lors d'associations multiples.	143
6.21	Exemple d'un photon reconstruit comportant des hits sur les chemins d'une trace. Les hits en rouge sur le chemin de la trace sont supprimés du photon et remis à disposition pour un prochain <i>clustering</i>	147
6.22	Distribution de l'énergie totale reconstruite pour des échantillons di-jets de 91, 200, 360 et 500 <i>GeV</i> reconstruits avec (a) l'algorithme ArborPFA v02-05-01 et (b) une reconstruction parfaite.	148
6.23	Énergie moyenne et déviation à la linéarité du système di-jets <i>uds</i> en fonction de l'énergie du boson Z^0 virtuel pour différents types de reconstructions.	149
6.24	Résolution relative en énergie des jets en fonction de l'énergie du boson Z^0 pour différents types de reconstructions.	149
6.25	Résolution relative de l'énergie des jets en fonction de $\cos\theta$ pour un boson Z^0 virtuel ayant une énergie de (a) 91 <i>GeV</i> , (b) 200 <i>GeV</i> , (c) 360 <i>GeV</i> et (d) 500 <i>GeV</i> , pour différents types de reconstructions. Les figures (e) et (f) montrent la même variable pour différentes énergies de jets, respectivement après reconstruction avec l'algorithme ArborPFA et une reconstruction parfaite.	150
7.1	Architecture logicielle du système de surveillance des expériences (a) CMS et (b) ALICE. On y retrouve les principaux ingrédients des systèmes de surveillance de données en physique des hautes énergies.	154
7.2	Architecture globale du logiciel DQM4HEP : des sources de données provenant du système d'acquisition (gauche) au système de visualisation (droite).	156
7.3	Schéma illustrant l'organisation des paquets du logiciel DQM4HEP : les trois paquets principaux de DQM4HEP (en bleu), les paquets installés avec DQM4HEP (en vert) et les paquets externes (en rose).	157
7.4	Diagramme de classe UML illustrant l'organisation des classes relatives aux événements. La classe <code>DQMEventStreamer</code> définit une interface pour la sérialisation des objets de type <code>DQMEvent</code>	159
7.5	Schéma illustrant la reconstruction des événements à partir des données brutes provenant du système d'acquisition et leur traitement au sein du logiciel de surveillance.	161

7.6	Schéma illustrant les interactions entre un collecteur d'événements et ses clients. Deux type de clients : le <i>pusher</i> qui envoie les événements au collecteur et le client dit <i>de réception</i> qui envoie des requêtes d'accès. Les deux types de client utilisent la même API.	163
7.7	Schéma illustrant l'interface utilisateur des modules d'analyse de données en ligne (application <i>standalone</i>).	165
7.8	Schéma illustrant l'interface utilisateur des modules d'analyse de données en ligne (application analysant les données issues du collecteur d'événements).	165
7.9	Diagramme d'état UML de l'application des modules d'analyse en ligne : à gauche l'application dite <i>standalone</i> et à droite l'application analysant les données issues du collecteur d'événement.	166
7.10	Diagramme de classe UML illustrant l'organisation des classes relatives aux cycles. Les trois implémentations (en bas) constituent les trois types de cycles disponibles par défaut dans le logiciel.	169
7.11	Schéma illustrant les interactions entre un collecteur d'éléments de surveillance et ses clients. Deux type de clients : le <i>sender</i> qui transmet les éléments au collecteur et le client qui envoie des requêtes d'accès. .	170
7.12	Interface graphique cliente de gestion de <i>run</i>	172
7.13	Interface graphique cliente de gestion de processus. La figure montre le découpage de l'interface graphique en plusieurs zones distinctes. La table des processus montre trois serveurs (lyosdhcal10, lyosdhcal7 et lyosdhcal9) avec chacun une liste de processus que l'utilisateur contrôle à distance.	173
7.14	Interface graphique cliente vers les collecteur d'éléments de surveillance.	174
7.15	Interface graphique cliente de récupération et d'affichage d'éléments de surveillance. Les éléments de surveillance affichés dans la zone centrale (2) proviennent du module d'analyse de données brutes du SDHCAL (voir section 3.3). Chaque élément tracé est une carte de comptage du nombre de hits pour une chambre donnée du SDHCAL. La sixième carte de comptage de la deuxième ligne (chambre numéro 12) montre un déficit de hits dans la zone centrale. Ceci est dû à un flux trop important de gaz dans la chambre, problème qui a pu être corrigé par la suite.	175
7.16	Schéma illustrant l'architecture interne de l'interface graphique de surveillance des données.	176
7.17	Schéma de positionnement des calorimètres SiWECAL et SDHCAL lors du test en faisceau du mois de juin 2016 : en haut la vue Z-X, en bas la vue X-Y. Les dimensions ne sont pas à l'échelle.	178
7.18	Éléments de surveillance du module de Slow Control les plus vérifiés. (a) Lecture de la haute-tension par chambre GRPC. (b) Lecture du courant par chambre GRPC. (c) Évolution de la température autour des calorimètres en fonction du temps. (d) Évolution de la pression autour des calorimètres en fonction du temps.	180

7.19	(a) Profil X-Y du SDHCAL pour un événement de bruit. On peut y voir une des trois zones avec une série de voie allumées dû à un bruit électronique. (b) Gerbe hadronique issue d'un π^+ de 70 GeV interagissant dans le SDHCAL lors du test sur faisceau.	182
7.20	Éléments de surveillance du module d'analyse de données brutes les plus vérifiés. (a) Carte de comptage des hits en fonction du numéro de la DIF et de l'ASIC. (b) Taux d'occupation des ASICs en fonction du numéro de DIF.	182
7.21	Éléments de surveillance du module d'analyse du faisceau les plus vérifiés. (a) Distribution de la durée d'activité du faisceau (<i>spill</i>). (b) Distribution de la durée d'un cycle d'acquisition.	183
7.22	Carte de comptage de hits dans le SiWECal : (a) plan 6 et (b) plan 8.	184
7.23	Schéma illustrant le déploiement des processus sur les différents serveurs lors du test en faisceau combiné SiWECal et SDHCAL en juin 2016.	185

Liste des tableaux

2.1	Rapports de sections efficaces mesurées avec (σ_{pol}) et sans (σ_{unpol}) polarisation de faisceaux pour deux processus produisant un boson de Higgs : le Higgsstrahlung et la fusion WW , pour une énergie au centre de masse de $\sqrt{s} = 350 \text{ GeV}$ [6].	11
2.2	Précisions attendues sur la mesure des couplages au boson de Higgs. Pour le mode de désintégration invisible, les valeurs sont données avec un taux de confiance à 95%. Les quatre colonnes correspondent à : LHC avec 300 fb^{-1} , ILC à 250 GeV avec 250 fb^{-1} , ILC à 500 GeV avec 500 fb^{-1} , ILC à 1 TeV avec 1000 fb^{-1} [6].	12
2.3	Principaux processus physiques qui seront étudiés à l'ILC pour différentes énergies au centre de masse [6].	13
2.4	Table résumant les performances requises pour les sous-détecteurs des détecteurs ILD et SiD [3].	18
2.5	Table résumant les paramètres des différentes couches de détection du détecteur de vertex de l'ILD : le rayon du plan, la demi-longueur du plan, l'angle maximal du plan, la résolution spatiale des pixels et le temps de lecture des pixels.	20
2.6	Principaux paramètres de la TPC proposée pour équiper le détecteur ILD.	22
2.7	Table résumant les paramètres des trajectographes en silicium, complémentaires à la TPC et au détecteur de vertex du détecteur ILD.	23
2.8	Paramètres géométriques principaux du trajectographe en silicium du détecteur SiD.	28
3.1	Table des longueurs de radiation pour les matériaux les plus utilisés en calorimétrie [38].	35
3.2	Table des longueurs d'interaction pour les matériaux les plus utilisés en calorimétrie [38].	36
3.3	Table des résolutions en énergie de quelques calorimètres électromagnétiques en physique des particules. Les trois premiers sont des calorimètres homogènes et les quatre suivants sont des calorimètres à échantillonnage.	37
3.4	Table des résolutions en énergie de quelques calorimètres hadroniques en physique des particules. Tous les calorimètres sont à échantillonnage [38].	38
3.5	Efficacité de sélection des gerbes hadroniques en fonction de l'énergie de la particule incidente [43].	53

3.6	Nombre moyen de hits, écart type et résolution relative avant et après correction temporelle pour chaque seuil, pour des pions de 70 GeV [43].	55
4.1	Table résumant les performances requises pour la reconstruction des particules issues d'un jet de 100 GeV, dans un détecteur hautement granulaire (i.e type ILD ou CLIC).	67
5.1	Paramètres de l'algorithme utilisés pour l'évaluation des incertitudes systématiques.	105
5.2	Exemple d'incertitudes systématiques estimées pour deux points en particulier dans l'étude de séparation des gerbes proches pour les données de test sur faisceau : efficacité du hadron neutre de 10 GeV superposé avec un hadron chargé de 10 GeV (cas # 1) et 30 GeV (cas # 2) à 10 cm de séparation (voir figure 5.21(a)).	106
6.1	Valeurs de constantes de calibration à différentes étapes de la calibration effectuée avec des kaons longs K_0^L . La calibration initiale est réalisée avec des angles $\phi = 0$ et $\theta = 1.5 \text{ rad}$, alors que les corrections inter-modules et angulaire utilisent toute la gamme en angle θ .	122
6.2	Table résumant les valeurs de champs magnétique pour le détecteur ILD_o2_v05 utilisées en entrée du programme de reconstruction ArborPFA.	124
7.1	Tables des performances mémoire et CPU des différents processus déployés lors du test sur faisceau. Les valeurs minimales et maximales pour chaque variable sont indiquées respectivement en souligné et en gras.	187
7.2	Tables des performances réseaux des différents serveurs utilisés lors du test sur faisceau	187

Introduction

Après la découverte au Large Hadron Collider (LHC) d'un boson de Higgs compatible avec celui du modèle standard, les futurs projets de collisionneurs, tels que le Collisionneur Linéaire International ILC, proposés pour succéder au LHC, sont de plus en plus étudiés. Les deux détecteurs proposés pour être placés au point de collision de l'ILC, le Grand Détecteur International ILD et le Détecteur en Silicium SiD, seront des détecteurs généralistes, conçus pour permettre l'application des algorithmes de suivi de particules, principal sujet d'intérêt de cette thèse.

Le calorimètre hadronique à lecture semi-digitale SDHCAL développé essentiellement à l'IPN de Lyon, fait partie des options pour le calorimètre hadronique du détecteur ILD. Les travaux effectués dans cette thèse portent sur le développement d'un algorithme de suivi de particules basé sur la topologie en arbre des gerbes hadroniques. En parallèle, un logiciel de surveillance de données en ligne générique nommé DQM4HEP a été développé.

Le chapitre 1 présentera le cadre théorique de la physique des particules, le modèle standard. Les différents mécanismes d'interaction ainsi que les particules le constituant seront introduits. Les limites du modèle seront discutées afin de mieux comprendre les propositions de nouveaux projets d'accélérateurs par la communauté scientifique.

Le chapitre 2 présentera le Collisionneur Linéaire International ILC. Les différents détecteurs et sous-détecteurs proposés pour enregistrer les événements au point de collision seront décrits et plus particulièrement le Grand Détecteur International ILD. Nous nous intéresserons aussi brièvement à d'autres projets de collisionneurs proposés par la communauté scientifique afin de succéder au LHC.

Le chapitre 3 décrira les différentes caractéristiques du prototype de calorimètre hadronique semi-digital SDHCAL développé à l'IPNL et proposé pour équiper le calorimètre hadronique du détecteur ILD. Une comparaison des résultats entre la simulation Geant4 du prototype et les données obtenues lors de test en faisceau au CERN permettront d'extraire les performances du détecteur, nécessaires au développement de l'algorithme de suivi de particules .

Dans le chapitre 4, nous introduirons dans un premier temps les algorithmes de suivi de particules utilisés pour reconstruire les événements enregistrés dans les détecteurs. L'algorithme PandoraPFA, algorithme de suivi de particule le plus ma-

ture à ce jour, sera décrit en détail. Le principe de l'algorithme ArborPFA, principal fruit de ces travaux de recherche, sera exposé.

Le chapitre 5 fournira une description de l'algorithme de suivi de particules dédié au détecteur SDHCAL. Dans un premier temps, les performances de l'algorithme sur des hadrons chargés seuls seront extraites et comparées aux performances du détecteur. Dans un second temps, une étude visant à séparer un hadron neutre et un hadron chargé proches dans ce calorimètre, permettra de quantifier la part de confusion due à l'algorithme sur la reconstruction du hadron neutre.

Le chapitre 6 fournira une description complète de l'algorithme pour des détecteurs génériques du type ILD. Les différences avec la version dédiée au SDHCAL seront soulignées. Les portions d'algorithmes supplémentaires, telles que le *re-clustering* statistique, seront aussi exposées. L'application aux systèmes di-jet *uds* permettront d'extraire des performances sur la résolution en énergie des jets.

Le chapitre 7 sera dédié au développement d'un logiciel de *monitoring* de qualité de données en ligne. L'architecture logicielle et les fonctionnalités seront présentées. L'implémentation pour le détecteur SDHCAL dans une situation de test en faisceau sera ensuite détaillée. Des tests de performance mémoire et performance réseau ainsi que des cas pratiques d'utilisation lors des tests en faisceau seront présentés afin d'apprécier la qualité du logiciel.

Chapitre 1

Le contexte théorique

LES avancées dans le domaine de la physique des particules et plus particulièrement dans notre compréhension de la composition de la matière ont été considérables lors du siècle dernier. Les découvertes faites au LHC (Large Hadron Collider) depuis son démarrage en 2008 ont permis de valider l'existence d'une particule compatible avec le boson de Higgs en juillet 2012. La théorie sur laquelle reposent ces prédictions et découvertes n'est autre que le modèle standard. Dans ce chapitre nous présenterons les particules élémentaires ainsi que leurs interactions dans le cadre de ce modèle. Le mécanisme de brisure de symétrie donnant naissance à la masse des particules sera ensuite discuté. Nous terminerons ce chapitre en exposant quelques limites de cette théorie et ses possibles extensions.

Contenu

1	Le modèle standard de la physique des particules	2
2	Les particules élémentaires et leurs interactions	3
3	Mécanisme de brisure de symétrie	6
4	Au delà du modèle standard	7

1 Le modèle standard de la physique des particules

Le modèle standard est une théorie de jauge basée sur les groupes de symétrie $SU(3)_C \times SU(2)_L \times U(1)_Y$ décrivant les constituants élémentaires de la matière ainsi que leurs interactions.

Le groupe unitaire $U(1)$ décrit les interactions de type électromagnétique grâce à la théorie de jauge abélienne de l'électrodynamique quantique. Les groupes de symétrie $U(1)_Y \times SU(2)_L$ décrivent quant à eux la théorie électrofaible, unification de l'électrodynamique quantique et de l'interaction faible. Enfin, la chromodynamique quantique (QCD), basée sur le groupe non-abélien $SU(3)_C$, décrit l'interaction forte, responsable de la cohésion des noyaux dans les atomes.

La théorie présente plus de 20 paramètres physiques libres qui sont déterminés à l'aide de mesures expérimentales :

- trois constantes de couplages, une pour chaque interaction : α_e , α_w et α_s
- dix paramètres fixant la masse des particules : 6 pour les quarks m_q , 3 pour les leptons m_l et 1 pour le boson de Higgs m_H
- la constante d'auto-couplage du Higgs λ_H
- les quatre paramètres de la matrice de Cabibbo-Kobayashi-Maskawa (CKM) : les trois angles de mélanges θ_{12} , θ_{13} , θ_{23} et la phase de violation CP δ_{13}
- l'angle θ_c de la chromodynamique quantique
- les paramètres liés à la masses des neutrinos

Comme le montre la figure 1.1, les particules élémentaires du modèles standard sont réparties en plusieurs familles.

Particules de matière (fermions)			Particules d'interactions		boson de masse
	I	II	III		
QUARKS	2.4 MeV +2/3 1/2 u up	1.27 GeV +2/3 1/2 c charm	171.2 GeV +2/3 1/2 t top	0 0 1 γ photon	125 GeV 0 0 H boson de Higgs
	4.8 MeV -1/3 1/2 d down	104 GeV -1/3 1/2 s strange	4.2 GeV -1/3 1/2 b bottom	0 0 1 g gluon	
	<2.2 eV 0 1/2 ν_e neutrino électronique	<0.17 MeV 0 1/2 ν_μ neutrino muonique	<15.5 MeV 0 1/2 ν_τ neutrino tauique	91.2 GeV 0 1 Z^0 boson Z	
LEPTONS	511 KeV -1 1/2 e électron	105.7 MeV -1 1/2 μ muon	1.777 GeV -1 1/2 τ tau	80.4 GeV ± 1 1 W^\pm bosons W	

BOSONS DE JAUGE

nom
spin
charge électrique
masse
symbole

125 GeV
0
0
H
boson H

FIGURE 1.1 – Les particules élémentaires du modèle standard réparties en familles et générations distinctes

On trouve d'un coté les 12 fermions, particules de spin demi-entier, composés des quarks (en jaune) et des leptons (en vert). Les deux familles se décomposent

en trois générations de doublet de particules. La première génération constitue la matière stable et observable de notre univers avec les masses les plus faibles. Les deux autres générations sont composées de particules instables et plus massives, n'existant que durant des temps très courts lors de processus de hautes énergies. Vient s'ajouter à la famille des fermions, celle des bosons, vecteurs des interactions entre particules élémentaires. Un boson supplémentaire est également présent, le boson de Brout-Englert-Higgs, conséquence du mécanisme de brisure symétrie et donnant naissance à la masse des particules.

2 Les particules élémentaires et leurs interactions

Comme expliqué dans la section précédente, les particules au sein du modèle standard sont réparties en deux grandes familles : les fermions et les bosons, les fermions étant répartis en deux sous catégories : les leptons et les quarks.

Les leptons, particules de spin $1/2$, sont organisés en trois générations de doublets de particules. L'électron, particule de matière de charge -1 , fut découvert en 1897 par les physiciens Joseph John Thomson, John Townsend et Harold A. Wilson lors de travaux visant à appliquer un champ électrique dans un tube cathodique pour mesurer la charge des "*corpuscules*" s'échappant de celui-ci. L'électron fut la première particule élémentaire mise en évidence par les physiciens. Il faudra attendre le XX^e siècle pour découvrir ses homologues chargés et plus massifs : le muon et le tau.

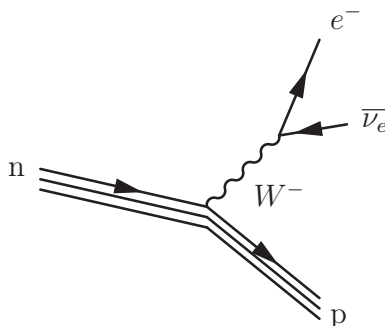


FIGURE 1.2 – Désintégration β^- mettant en jeu la production d'un anti-neutrino électronique

A chacune de ces particules chargées est associée une particule neutre quasiment sans masse, les neutrinos, n'interagissant que par interaction faible. Leur existence est postulée par Wolfgang Ernst Pauli en 1930 pour satisfaire les lois de conservation de l'énergie et de l'impulsion lors de la désintégration β^- dont le diagramme de Feynman est illustré sur la figure 1.2. L'anti-neutrino $\bar{\nu}_e$ accompagnant la formation d'un électron lors de la désintégration du neutron est découvert expérimentalement en 1956 auprès d'un réacteur nucléaire aux États-Unis. Les neutrinos muoniques et tauiques ne seront découverts quant à eux que dans les années 1962

et 2000 au Laboratoire national de Brookhaven et dans l'expérience DONUT à Fermilab respectivement.

L'expérience Super-Kamiokande au Japon mit ensuite en évidence en 1998 le phénomène d'oscillation des neutrinos permettant à ceux-ci de changer de saveur (par exemple de ν_e en ν_μ) avec le temps, montrant ainsi que les neutrinos ont une masse faible mais non nulle. Cette découverte valut le prix Nobel de Physique en 2002 au physicien Masatoshi Koshiba.

Les quarks forment la seconde famille de fermions au sein du modèle standard. Tout comme les leptons, cette famille est répartie en trois générations de doublets d'isospin faible. Les quarks sont les particules élémentaires composant les hadrons. Dans la première génération, on trouve les quarks les plus légers u et d formant les protons ou les neutrons dans les noyaux des atomes. Les deux familles suivantes contiennent les quarks c et s puis b et t , ce dernier étant à ce jour la particule la plus massive du modèle standard.

Historiquement, c'est en 1964 que Murray Gell-Mann suggère que le nombre de particules connues à ce jour pouvaient être expliqué comme une combinaison de trois particules fondamentales de charge $-1/3$ et $2/3$ appelées *quarks*¹. Plus tard viendra la théorie de la chromodynamique quantique décrivant ces particules et l'interaction forte, maintenant comprise dans le modèle standard. Dans cette théorie, les quarks interagissent par l'intermédiaire de 8 bosons vecteurs appelés gluons, sans masse ni charge². Un nouveau nombre quantique est également introduit, la couleur, qui peut prendre trois *valeurs* : rouge, vert ou bleu. La couleur permet d'expliquer l'existence de certains états liés comme par exemple celle du baryon Δ^{++} composé de trois quarks u tous de couleur différente et que le principe d'exclusion de Pauli devrait rejeter en l'absence de couleur. Une autre spécificité de cette théorie est le confinement de la couleur au sein des hadrons. Elle implique que les quarks ne peuvent exister dans un état non-lié mais sous forme de paires quarks anti-quark appelées *mésons* ou de triplet de quarks appelés *baryons*.

A haute énergie, lorsque des quarks sont produits, d'autres quarks sont spontanément créés pour former plusieurs états liés : ce sont les processus de *fragmentation* et de *hadronisation*. Il est à l'origine du phénomène de gerbes hadroniques observé dans les calorimètres ou encore des *jets* de hadrons suite aux collisions dans les accélérateurs de particules comme l'illustre le diagramme de la figure 1.3.

Les bosons vecteurs ou bosons *de jauge* sont les particules régissant les interactions entre particules.

Tout d'abord le photon ou γ , régit l'interaction électromagnétique entre particules chargées. Il a une charge nulle, une masse nulle et un spin de 1. Sa masse nulle lui confère, en vertu du principe d'incertitude d'Heisenberg, une portée d'interaction infinie mais dont l'intensité varie comme $1/r^2$ (r étant la distance entre deux

1. Le terme *quark* provient d'une phrase du roman *Finnegans Wake* de James Joyce : **Three Quarks for Muster Mark !**

2. Les quarks sont aussi soumis aux interactions électromagnétique et faible comme les leptons.

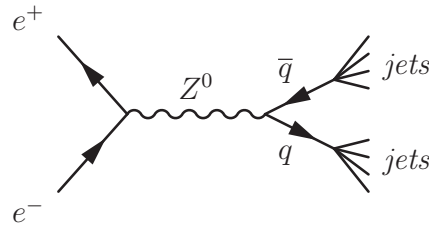


FIGURE 1.3 – Production d'une paire quark anti-quark suite à une collision $e^+e^- \rightarrow q\bar{q}$. Une paire de jets est créée suite au processus d'hadronisation des quarks.

particules chargées).

Pour l'interaction faible, trois bosons vecteurs régissent les interactions entre particules : les bosons W^\pm et Z^0 de masses respectives $m_{W^\pm}=80.385 \pm 0.015 \text{ GeV}$ et $m_{Z^0}=91.188 \pm 0.002 \text{ GeV}$ [38]. Leurs masses non nulles confèrent à l'interaction faible une portée très courte de l'ordre de $\sim 10^{-17}m$. Comme le montre la figure 1.2, la désintégration du neutron en proton met en jeu un boson W^- mais ne change pas la saveur des leptons, propriété commune de toutes les interactions de type faible. Cependant, elle permet de changer la saveur des quarks, ce qui explique le changement de saveur du quark u du neutron en quark d du proton lors de cette interaction.

Finalement, l'interaction forte dénombre 8 bosons de propriétés identiques appelés *gluons*. Du point de vue de la chromodynamique quantique, qui est basée sur le groupe de symétrie $SU(3)_C$, il devrait exister 9 gluons formés des différentes combinaisons de couleurs et anti-couleurs de la théorie. Du fait que les états de couleur neutre ne peuvent interagir par interaction forte, une relation entre trois états restreint alors le nombre de gluons à 8. Les propriétés du gluon sont similaires à celles du photon, il est de charge nulle, de masse nulle et de spin 1, mais la portée de l'interaction n'est pas infinie à cause du phénomène de confinement des quarks au sein des hadrons. Le gluon permet l'échange de couleur entre quarks comme le montre le diagramme de la figure 1.4.

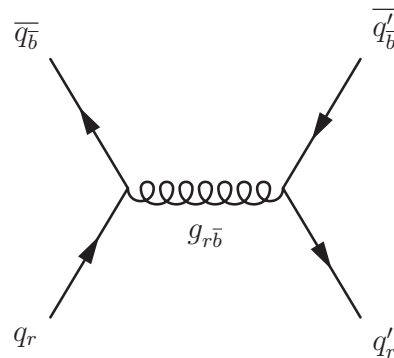


FIGURE 1.4 – Diagramme de Feynman d'un échange de couleurs entre quarks par l'intermédiaire d'un gluon.

3 Mécanisme de brisure de symétrie

A l'origine, le modèle de Glashow-Weinberg-Salam unifiant les interactions électromagnétique et faible, prédisait l'existence de toutes les particules du modèle électrofaible dont les bosons d'interaction γ , W^\pm et Z^0 . Cependant, il ne permettait pas d'expliquer la masse non-nulle des bosons W^\pm et Z^0 ainsi que celle des fermions.

C'est en 1964 qu'est introduit le mécanisme de Brout-Englert-Higgs, brisant spontanément la symétrie du Lagrangien grâce à l'introduction d'un champ scalaire dont la valeur moyenne dans le vide est non-nulle. Le potentiel renormalisable le plus simple brisant la symétrie peut s'écrire sous la forme :

$$V(\phi^\dagger\phi) = \mu^2\phi^\dagger\phi + \lambda(\phi^\dagger\phi)^2 \quad (1.1)$$

où le potentiel ϕ est un doublet de $SU(2)$ appelé doublet de Higgs dont le minimum local dans le vide est donné par :

$$\phi_0 = \sqrt{\frac{-\mu^2}{2\lambda}} \quad (1.2)$$

En introduisant ce potentiel dans le Lagrangien du modèle standard, il est possible de déterminer l'expression de toutes les masses des particules du modèle standard. Cependant, un terme de masse supplémentaire apparaît prédisant l'existence d'un boson de masse non-nulle et de spin 0 : le boson de Higgs. L'expression de sa masse est prédite mais sa valeur ne peut être estimée qu'à partir de mesures expérimentales.

En juillet 2012, les expériences CMS et ATLAS du collisionneur LHC, mesurent un excès de masse compris entre 125 et 127 GeV mettant en évidence l'existence d'une nouvelle particule compatible avec le boson de Higgs du modèle standard [1]. Cependant la source d'erreur la plus importante sur cette mesure réside dans le manque de statistique [31]. L'augmentation de l'énergie au centre de masse ainsi que de la luminosité au LHC lors des prises de données suivantes a permis d'augmenter la statistique et de fournir une meilleure précision sur cette mesure. Les augmentations suivantes permettront aussi de mesurer d'autres propriétés jusqu'alors inaccessibles de ce boson, notamment sa constante d'auto-couplage λ_H , prédite par le modèle standard.

Cette découverte motive aussi la construction d'autres collisionneurs tels que le Collisionneur Linéaire International (ILC) [9], véritable usine à boson de Higgs et dont la netteté des événements enregistrés améliorera la mesure de ces propriétés. D'autres projets sont aussi à l'étude et proposent des techniques d'accélération de particules différentes (linéaire ou circulaire), des gammes d'énergie différentes ou encore des particules différentes dans l'état initial de collision (hadron ou leptons). Parmi ces projets, on peut citer le Collisionneur Linéaire Compact (CLIC) [32], le Future Collisionneur Circulaire (FCC)[10] ou le Collisionneur Circulaire Electron-Positron (CEPC) [21].

4 Au delà du modèle standard

Le modèle standard est sans doute, avec la relativité générale, l'une des théories les mieux vérifiées de la physique moderne. Cependant, ce modèle ne permet pas d'expliquer certains phénomènes expérimentaux. Les plus fréquemment cités sont les suivants :

La masse des neutrinos

Le mécanisme de brisure de symétrie permet de donner une masse à certaines particules. Dans ce modèle, les neutrinos ont une masse nulle. Cependant, le phénomène d'oscillation des neutrinos, qui a été mesuré par certaines expériences comme Super-Kamiokande [24], montre que les neutrinos doivent avoir une masse non-nulle. Des termes de masse peuvent être introduits à la main dans le Lagrangien du modèle standard donnant ainsi naissance à la masse des neutrinos.

L'asymétrie matière-antimatière

Le modèle standard suppose que la matière et l'antimatière ont été produites en quantités égales lors du Big Bang, l'Univers dans son état initial ne supposant aucune asymétrie entre les deux. Aucun mécanisme ne permet pour le moment de rendre compte de cette asymétrie pourtant existante.

La gravité

La gravitation ne fait pas partie du modèle standard. Des théories alternatives telles que la théorie des cordes ou la théorie de supergravité tentent de les unifier. Cependant, étant donné les énergies et les conditions pour tester ces théories, les technologies permettant de les vérifier expérimentalement sont inaccessibles à ce jour.

La matière noire et l'énergie noire

La matière décrite par le modèle standard permet d'expliquer environ 4% de la matière observable dans notre univers. Parmi les 96% restants, se trouverait 69% d'énergie noire, sorte de densité d'énergie constante pour le vide. Les 27% restants sont attribués à la matière noire n'interagissant que par interaction faible. Or, aucune particule du modèle standard ne se présente comme bonne candidate à la matière noire.

Parmi les théories permettant des unifications ou des extensions du modèle standard, on trouve la supersymétrie (SuSy) [22], introduisant des particules super-partenaires à chaque fermion et boson. A chaque fermion est associé un partenaire bosonique et de même à chaque boson est associé un partenaire fermionique. La supersymétrie prédit entre autre l'existence d'une particule neutre appelé *neutralino* qui pourrait être un bon candidat pour la matière noire. Cette théorie permettrait également de résoudre le problème de hiérarchie des masses des particules du modèle standard. D'autres modèles sont aussi à l'étude et pourront être validés (ou invalidés) par les futures prises de données du LHC et par les projets de futurs collisionneurs de plus grande précision.

Chapitre 2

Le collisionneur linéaire international ILC

BIEN que le modèle standard ait montré de nombreux succès, des limites subsistent et de nouvelles mesures pourraient encore le mettre à l'épreuve. Les mesures effectuées par le LHC ont permis de mettre en évidence un boson compatible avec le boson de Higgs du modèle standard. Des projets de collisionneurs leptoniques sont en cours de développement et permettraient d'affiner certaines mesures déjà effectuées et de faire de nouvelles découvertes. Parmi ces projets, le Collisionneur Linéaire International (ILC) est à ce jour le plus abouti. Un calorimètre hadronique à lecture semi-digitale a été développé pour équiper l'un des deux détecteurs de l'ILC et fera l'objet d'un chapitre dédié. Après avoir exposé les motivations d'un collisionneur leptonique pour la recherche en physique des hautes énergies ainsi que le programme de physique de l'ILC, nous détaillerons le complexe d'accélération et les détecteurs qui l'équiperont. Un accent sera mis sur l'ILD, principal détecteur utilisé pour ce travail de recherche. Un point sera fait sur les autres projets de collisionneur leptonique en fin de chapitre.

Contenu

1	Motivations et programme de physique de l'ILC	10
1.1	Motivation d'un collisionneur leptonique linéaire	10
1.2	Programme de physique de l'ILC	13
2	Le collisionneur linéaire international ILC	15
2.1	Le complexe d'accélérateur	15
2.2	Les détecteurs au point de collision	17
3	Les autres projets de collisionneurs leptoniques	29

1 Motivations et programme de physique de l'ILC

1.1 Motivation d'un collisionneur leptonique linéaire

En physique des hautes énergies, les accélérateurs de particules sont utilisés pour faire collisionner des particules à très hautes énergies. Ils permettent ainsi d'accéder à de nouvelles particules secondaires de plus faibles énergies. Deux types de collisionneurs se distinguent : les machines dites de "découverte" comme le LHC utilisant des hadrons lors de collisions et les machines dites de "précision", comme le LEP, qui utilisent des leptons.

Les collisionneurs hadroniques permettent d'atteindre des gammes d'énergies généralement supérieures à celles des collisionneurs leptoniques. Les hadrons n'étant pas des particules élémentaires mais constitués de quarks, seule une partie des composants et donc une fraction de l'énergie des hadrons, sera mise en jeu lors d'une collision. C'est le cas au LHC où plusieurs réactions sont possibles entre les protons : la fusion de gluons, la collision quark-quark ou l'interaction quark-gluon. Cependant, la réaction finale et son énergie mise en jeu ne sont pas connues à l'avance mais uniquement avec une certaine probabilité.

Dans un collisionneur, les particules sont accélérées sous forme de paquets, séparés par un temps très court, formant ainsi un faisceau. Deux faisceaux sont accélérés en sens inverse pouvant donner lieu à des collisions. Suivant les caractéristiques de ces faisceaux, plusieurs particules d'un même paquet peuvent interagir en même temps. Ce phénomène est connu sous le nom d'empilement et produit des particules secondaires supplémentaires, rendant ainsi les mesures résultantes moins précises. L'empilement peut aussi être généré par des paquets provenant de collisions antérieures ou postérieures à un événement. Le nombre moyen d'interactions par croisement de faisceaux a été estimé à 21 dans le détecteur CMS en 2012 [14]. Ce nombre pourrait augmenter jusqu'à 140 interactions après une amélioration du LHC en HL-LHC (High Luminosity Large Hadron Collider) .

La configuration circulaire ou linéaire des accélérateurs est aussi un paramètre important. Les accélérateurs linéaires avec des leptons ne permettent d'atteindre des énergies aussi élevées que pour les accélérateurs circulaires utilisant des hadrons. En accélérant une particule chargée jusqu'à des énergies ultra-relativistes sur une trajectoire courbée, celle-ci émet un rayonnement appelé *rayonnement synchrotron*. L'énergie dissipée par ce rayonnement, proportionnelle à $1/m^4$ et à $1/R^2$ (m étant la masse de la particule et R le rayon de courbure de sa trajectoire), explique l'utilisation des hadrons et de grands rayons de courbure par les accélérateurs circulaires tels que le LHC.

La forte radiation émise par les collisionneurs hadroniques n'est pas non plus négligeable. En effet, les différents sous-détecteurs et surtout ceux placés près du point de collision, doivent présenter une forte résistance à la radiation. Durant l'amélioration du LHC en HL-LHC, certains détecteurs comme le trajectographe ou le calorimètre électromagnétique du détecteur CMS devront être remplacés, suite à des dégâts trop importants dû à cette radiation. Des expériences telles que le GIF++ [39] étudient l'effet du vieillissement des détecteurs sur leur efficacité en les soumet-

tant à une radiation élevée et permettent de mieux choisir les matériaux constituant les détecteurs en fonction du taux de radiation attendu sur le point de collision.

De leur côté, les collisionneurs leptoniques ne présentent que très peu d'empilement [6]. Une partie de cette empilement est émise près du faisceau et est due aux diffusions Bhabha ($e^+e^- \rightarrow e^+e^-$). Du fait de la nature élémentaire des particules primaires, l'énergie au centre de masse est beaucoup moins dispersée. Il sera surtout possible de choisir la valeur de l'énergie du faisceau, permettant ainsi de sélectionner les phénomènes à étudier.

Pour atteindre de hautes énergies avec un collisionneur leptonique, deux choix s'imposent. Le premier serait un collisionneur leptonique linéaire, sans rayonnement synchrotron. Du fait de sa nature linéaire, il ne serait possible d'accélérer les particules qu'une seule fois avant de les faire collisionner. Les particules ne participant pas à la collision seraient alors perdues. Le gradient d'accélération du complexe devra se montrer aussi très important pour ne pas construire une expérience avec des dimensions surréalistes.

Processus	P_{e^-}	P_{e^+}	$\sigma_{pol}/\sigma_{unpol}$
$e^+e^- \rightarrow ZH$	0.8	0	~ 1.13
$e^+e^- \rightarrow ZH$	0.8	0.3	~ 1.44
$e^+e^- \rightarrow ZH$	0.8	0.55	~ 1.71
$e^+e^- \rightarrow H\nu_e\bar{\nu}_e$	0.8	0	~ 1.90
$e^+e^- \rightarrow H\nu_e\bar{\nu}_e$	0.8	0.3	~ 2.40
$e^+e^- \rightarrow H\nu_e\bar{\nu}_e$	0.8	0.55	~ 2.95

TABLE 2.1 – *Rapports de sections efficaces mesurées avec (σ_{pol}) et sans (σ_{unpol}) polarisation de faisceaux pour deux processus produisant un boson de Higgs : le Higgsstrahlung et la fusion WW , pour une énergie au centre de masse de $\sqrt{s} = 350 \text{ GeV}$ [6].*

Le deuxième choix serait de proposer un collisionneur circulaire avec un très grand périmètre pour minimiser les pertes d'énergies par rayonnement synchrotron. Les projets CEPC (Circular Electron Positron Collider) [21] et FCC (Futur Circular Collider) [10] proposeraient des complexes d'accélération de l'ordre de 50 à 70 km et 80 à 100 km de périmètre respectivement pour atteindre des énergies au centre de masse de l'ordre de 400 GeV au maximum. Ces machines présenteraient au moins l'avantage de pouvoir être changées en collisionneur circulaire hadronique pouvant atteindre des énergies au centre de masse de l'ordre de 100 TeV !

La dernière particularité intéressante des collisionneurs leptoniques est la possibilité de polariser les faisceaux de particules. La polarisation d'un faisceau correspond au taux de particules dont le sens de propagation des particules est aligné avec leur spin. L'hélicité des électrons et positons est fortement corrélée au spin de la particule échangée lors des processus de diffusions et d'annihilations. En polarisant un faisceau de particules, certaines sections efficaces pourraient augmenter considérablement, ou de même, être fortement réduites (i.e pour un bruit de fond donné). A titre d'exemple, la table 2.1 présente différents rapports de sections efficaces avec et sans polarisation des faisceaux, pour deux processus produisant un boson de Higgs.

Mode	LHC	ILC(250)	ILC(500)	ILC(1000)
WW	4.1 %	1.9 %	0.24	0.17 %
ZZ	4.5 %	0.44 %	0.30 %	0.27 %
$b\bar{b}$	13.6 %	2.7 %	0.94 %	0.69 %
gg	8.9 %	4.0 %	2.0 %	1.4 %
$\gamma\gamma$	7.8 %	4.9 %	4.3 %	3.3 %
$\tau^+\tau^-$	11.4 %	3.3 %	1.9 %	1.4 %
$c\bar{c}$	-	4.7 %	2.5 %	2.1 %
$t\bar{t}$	15.6 %	14.2 %	9.3 %	3.7 %
$\mu^+\mu^-$	-	-	-	16 %
$self$	-	-	104 %	26 %
$BR(invis.)$	< 9 %	< 0.44 %	< 0.30 %	< 0.26 %
$\Gamma_T(h)$	20.3 %	4.8 %	1.6 %	1.2 %

TABLE 2.2 – Précisions attendues sur la mesure des couplages au boson de Higgs. Pour le mode de désintégration invisible, les valeurs sont données avec un taux de confiance à 95%. Les quatre colonnes correspondent à : LHC avec $300 fb^{-1}$, ILC à 250 GeV avec $250 fb^{-1}$, ILC à 500 GeV avec $500 fb^{-1}$, ILC à 1 TeV avec $1000 fb^{-1}$ [6].

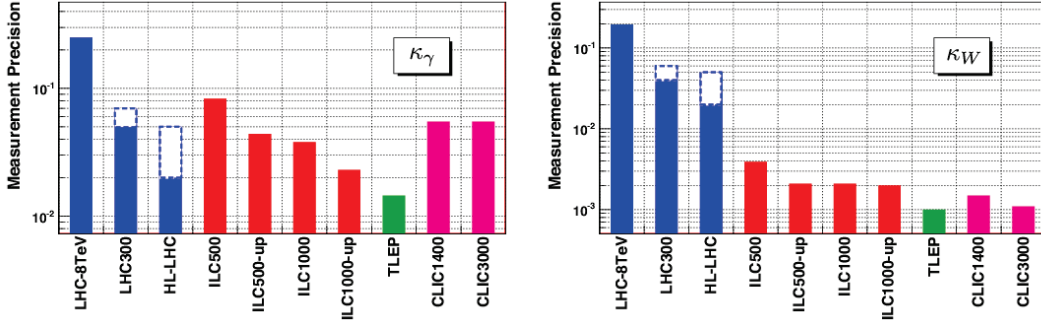


FIGURE 2.1 – Comparaison des précisions attendues sur les facteurs κ_γ et κ_W au LHC, à l'ILC, au TLEP et à CLIC pour différentes versions des collisionneurs [6].

La table 2.2 présente une comparaison des précisions attendues sur la mesure des couplages au boson de Higgs entre les mesures effectuées au LHC et à l'ILC avec trois points d'énergie. La figure 2.1 montre une comparaison de la précision attendue sur les déviations au modèle standard de la mesure de ces couplages, définies par les facteurs κ_A :

$$\kappa_A = \frac{g_{HA\bar{A}}}{g_{HA\bar{A},SM}} \quad (2.1)$$

où $g_{HA\bar{A}}$ correspond aux valeurs de couplage entre le boson de Higgs et les particules du modèle standard et $g_{HA\bar{A},SM}$ aux valeurs de couplages prédites par le modèle standard. L'environnement propre et l'énergie au centre de masse connue précisément à l'ILC, feront de ce collisionneur un digne successeur du LHC et permettront de tester le modèle standard avec efficacité.

1.2 Programme de physique de l'ILC

Énergie	Réaction	Mesure physique
91 GeV	$e^+e^- \rightarrow Z$	Mesure de précision électrofaible
160 GeV	$e^+e^- \rightarrow WW$	Masse du boson W
250 GeV	$e^+e^- \rightarrow Zh$	Couplage du Higgs
350 – 400 GeV	$e^+e^- \rightarrow t\bar{t}$	Couplages et masse du quark top
	$e^+e^- \rightarrow WW$	Couplages du W
	$e^+e^- \rightarrow \nu\bar{\nu}h$	Couplages du Higgs
500 GeV	$e^+e^- \rightarrow f\bar{f}$	Recherche d'un boson Z'
	$e^+e^- \rightarrow t\bar{t}h$	Couplages du Higgs au quark top
	$e^+e^- \rightarrow Zhh$	Auto-couplage du Higgs
	$e^+e^- \rightarrow \tilde{\chi}\tilde{\chi}$	Recherche de supersymétrie
	$e^+e^- \rightarrow AH, H^+H^-$	Recherche de nouveaux états du Higgs
700 – 1000 GeV	$e^+e^- \rightarrow \nu\bar{\nu}hh$	Auto-couplage du Higgs
	$e^+e^- \rightarrow \nu\bar{\nu}VV$	Secteur du Higgs composite
	$e^+e^- \rightarrow \nu\bar{\nu}t\bar{t}$	Secteur du Higgs composite et quark top
	$e^+e^- \rightarrow \tilde{t}\tilde{t}^*$	Recherche de supersymétrie

TABLE 2.3 – Principaux processus physiques qui seront étudiés à l'ILC pour différentes énergies au centre de masse [6].

La table 2.3 résume les principaux processus qui seront étudiés à l'ILC pour différentes énergies au centre de masse. L'énergie des deux faisceaux peut être réglée pour scanner toute les énergies jusqu'à l'énergie maximale de l'ILC, fixée à 1 TeV si les deux bras d'accélération sont rallongés (voir section 2.1). Ce programme physique, défini dans [6], comprendrait les énergies suivantes :

- **91 GeV** : Cette énergie correspond à la masse du boson Z . Son étude permettra d'améliorer la précision sur la mesure de l'angle de Weinberg θ_W et des constantes de couplages au boson Z .
- **160 GeV** : Cette énergie correspond au seuil de la réaction $e^+e^- \rightarrow W^+W^-$. Ce canal physique permettra de mesurer la masse du boson W avec une précision de l'ordre du MeV .
- **250 GeV** : Le processus $e^+e^- \rightarrow ZH$ (diagramme de la figure 2.2(c)) est la principale étude à cette énergie. La figure 2.2(d) montre différentes sections efficaces impliquant un boson de Higgs dans l'état final. À 250 GeV, on voit que ce processus est dominant. De plus, en étudiant les produits de désintégration du boson Z uniquement, il est possible d'étudier le boson de Higgs indépendamment du modèle théorique.
- **350-400 GeV** : Le seuil de production d'une paire de quark top se trouve aux alentours de 350 GeV. Une précision de l'ordre de 100 MeV peut être atteinte sur la mesure de sa masse. Cette mesure permettra aussi de contraindre plus finement le modèle standard et d'autres modèles d'unification. Comme le montre la figure 2.2(d), la réaction $e^+e^- \rightarrow \nu\bar{\nu}H$ est dominante sur cette

gamme d'énergie. Une mesure de la section efficace de ce processus permettrait d'estimer le couplage hWW . Sachant que cette section efficace augmente approximativement comme $\log(E_{CM}/m_h)$, une statistique plus importante serait accessible à plus haute énergie. Finalement, la mesure des couplages du boson W à haute énergie permettrait de découvrir de nouveaux phénomènes, ces couplages étant très sensibles à des modifications à plus haute énergie.

- **500 GeV** : Cette énergie correspond à l'énergie et à la luminosité nominale à l'ILC. Les réactions du type $e^+e^- \rightarrow f\bar{f}$ seront utiles pour sonder la structure interne des quarks et des leptons, ou encore étudier des modèles au delà du modèle standard. Les canaux $e^+e^- \rightarrow t\bar{t}H$ et $e^+e^- \rightarrow ZHH$ permettront quant à eux de mesurer le couplage entre le boson de Higgs et le quark top et l'auto-couplage du boson de Higgs.
- **700-1000 GeV** : En doublant la taille des accélérateurs, il sera possible d'effectuer de nouvelles mesures de l'auto-couplage du boson de Higgs ou du couplage entre le boson de Higgs et le quark top.

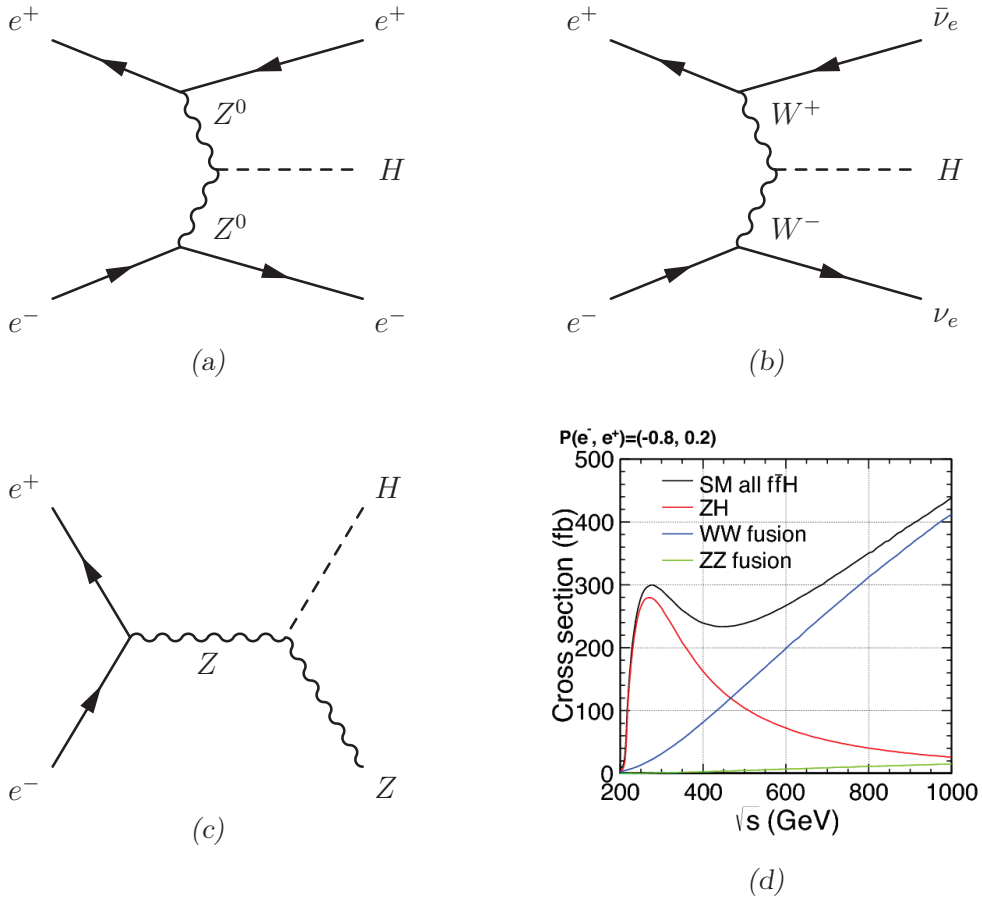


FIGURE 2.2 – a) Diagramme de Feynman de la fusion Z^0 . b) Diagramme de Feynman de la fusion W^\pm . c) Diagramme de Feynman du Higgsstrahlung. d) Différentes sections efficaces impliquant un boson de Higgs dans l'état final [6].

Le programme réel qui sera mis en application sera amené à changer suivant les nouvelles découvertes et nouvelles mesures qui seront effectuées par le LHC à 14 TeV . Mais même dans le cas de nouvelles découvertes, les mesures prévues par l'ILC restent intéressantes du point de vue de leur précision, une part importante de ces mesures étant liées au boson de Higgs.

2 Le collisionneur linéaire international ILC

Le collisionneur linéaire international ILC est un projet de collisionneur linéaire électron-positon. Son énergie au centre de masse serait comprise entre 91 et 500 GeV dans un premier temps avec la possibilité d'étendre les accélérateurs pour atteindre une énergie de 1 TeV . Projet initié au début des années 2000 par la collaboration TESLA, un premier aperçu des paramètres de la machine ne tardera pas (2003 puis 2006). Ces paramètres sont, bien entendu, mis à jour régulièrement en fonction des découvertes au LHC et par d'autres expériences. Un rapport de conception technique a été publié en juin 2013 [6] en 5 volumes décrivant le projet dans le moindre détail. Le projet aurait de forte chance d'être hébergé au Japon sur le site de Iwate, sous la montagne de Kitakami au nord de Tokyo.

2.1 Le complexe d'accélérateur

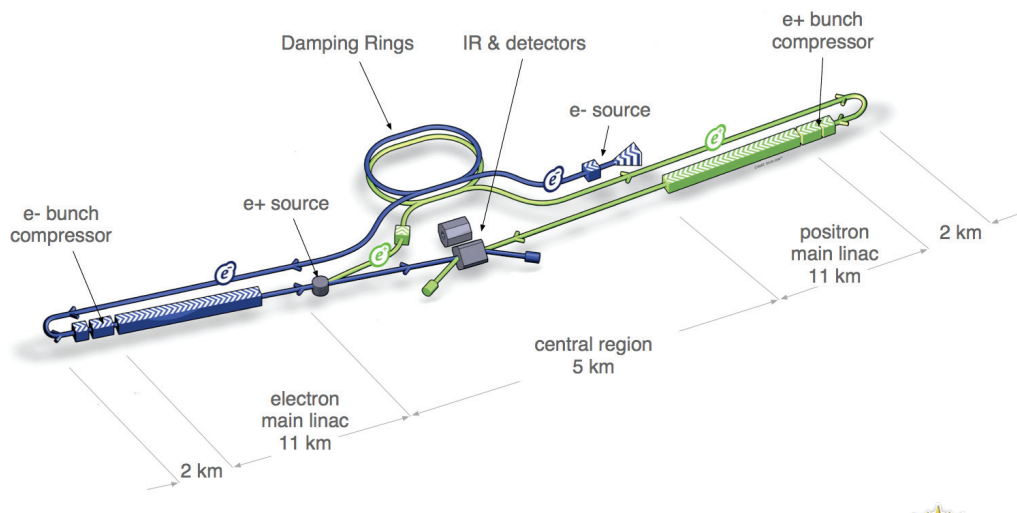


FIGURE 2.3 – Schéma du complexe d'accélération du collisionneur linéaire international [9].

La figure 2.3 illustre le schéma du complexe d'accélération de l'ILC. On y retrouve les différentes sous-parties composant le complexe [9] :

- une source d'électrons polarisés extraits d'une photocathode DC [36]

	Machine standard 500 GeV				Amélioration 1 TeV			
Énergie au centre de masse	E_{CM}	GeV	250	350	500	1000		
Taux de collisions	f_{rep}	Hz	5	5	5	4		
Nombre de paquets	n_b		1312	1312	1312	2450		
Nombre de particules par paquet	N	$\times 10^{10}$	2	2	2	2.74		
Séparation entre les paquets	δt_b	ns	554	554	554	366		
Courant de pulsation	I_{beam}	mA	5.8	5.8	5.8	7.6		
Gradient d'accélération	G_a	$MV.m^{-1}$	14.7	21.4	31.5	38.2		
Puissance moyenne du faisceau	P_{beam}	MW	2.9	7.3	10.5	27.2		
Polarisation des électrons	P_-	%	80	80	80	80		
Polarisation des positons	P_+	%	30	30	30	20		
Étalement en énergie des électrons	$\Delta p/p$	%	0.190	0.158	0.124	0.083		
Étalement en énergie des positons	$\Delta p/p$	%	0.152	0.100	0.070	0.043		
Longueur des paquets	σ_z	mm	0.3	0.3	0.3	0.250		
Taille horizontale du faisceau au point de collision	σ_x^*	nm	729.0	683.5	474	481		
Taille verticale du faisceau au point de collision	σ_y^*	nm	7.7	5.9	5.9	2.8		
Luminosité	L	$\times 10^{34} cm^{-2} s^{-1}$	0.75	1.0	1.8	3.6		

FIGURE 2.4 – Récapitulatif des paramètres de la machine ILC pour les trois énergies stantardes et l'amélioration à 1 TeV [4].

- une source de positons polarisés obtenus par conversion d'un photon de haute énergie, lui-même obtenu d'une paire électron-positon convertie en faisant passer des électrons accélérés dans un onduleur
- deux anneaux de stockage pour les électrons et positons de 5 GeV, dans un tunnel commun de 3,2 km de circonférence
- deux compresseurs de paquet de particules précèdent les *linacs* principaux, un pour les électrons et un pour les positons
- deux accélérateurs linéaires (*linac*) principaux, utilisant chacun deux cavités radiofréquences supraconductrice de 1.3 GHz fournissant un gradient moyen d'accélération de 31.5 MV.m⁻¹ d'une pulsation de 1.6 ms
- deux systèmes de distribution de faisceaux de 2,2 km de long, permettant des collisions avec un angle de 14 mrad en un unique point qui pourra être occupé par deux détecteurs dans une configuration dite "*push-pull*".

La table 2.4 présente quant à elle les principaux paramètres de la machine ILC. Certains de ces paramètres ont été sujets à optimisation alors que d'autres ont été fixés par contrainte. Par exemple, la séparation entre les paquets et le nombre de particules par paquet sont limités par un temps d'injection et un temps d'extraction réalistes. De même, les paramètres reliés au paquet de particules sont limités par les contraintes au point d'interaction (i.e stabilité des faisceaux ou réduction du bruit fond).

Pour une énergie au centre de masse de 500 GeV, le taux de collision est estimé à 5 Hz, ce qui, comparé au 40 MHz du LHC à son démarrage, est très faible. Cela correspond à un faisceau actif pendant 0.95 ms toute les 200 ms, les paquets de particules étant espacés de 554 ns. Le nombre de paquets de particules (1312) et le nombre de particules par paquet (2×10^{10}) restent aussi très faibles comparés au LHC. L'accélérateur pourra, dans sa version à 500 GeV, délivrer une luminosité d'environ $2 \times 10^{34} \text{cm}^{-2} \text{s}^{-1}$ (jusqu'à $3.6 \times 10^{34} \text{cm}^{-2} \text{s}^{-1}$ à 1 TeV).

2.2 Les détecteurs au point de collision

Les détecteurs conçus pour l'ILC doivent pouvoir satisfaire d'une part le programme physique exposé dans la section 1.2 et d'autre part les contraintes du complexe d'accélération, par exemple le bruit de fond près de la ligne de faisceau. Deux détecteurs ont été proposés afin d'éviter d'éventuels biais expérimentaux et confirmer les mesures et découvertes : ILD et SiD. Ils seront installés sur une plateforme amovible afin d'alterner les prises de données. Ce mécanisme, dit "*push-pull*", permettra à l'un des deux détecteurs d'être placé sur la ligne de faisceau pendant que l'autre sera placé en maintenance. Le temps de transition devra être de l'ordre de la journée pour pouvoir maximiser la luminosité intégrée de l'ILC.

Ces deux détecteurs sont conçus pour permettre l'application de l'algorithme de suivi de particules ou *Particle Flow Algorithm* (PFA). Cet algorithme consiste à utiliser le sous-détecteur le plus approprié pour mesurer l'énergie des particules dans les jets. Le chapitre 4 dédié aux algorithmes de suivi de particules présentera les différentes implémentations logicielles de ce concept.

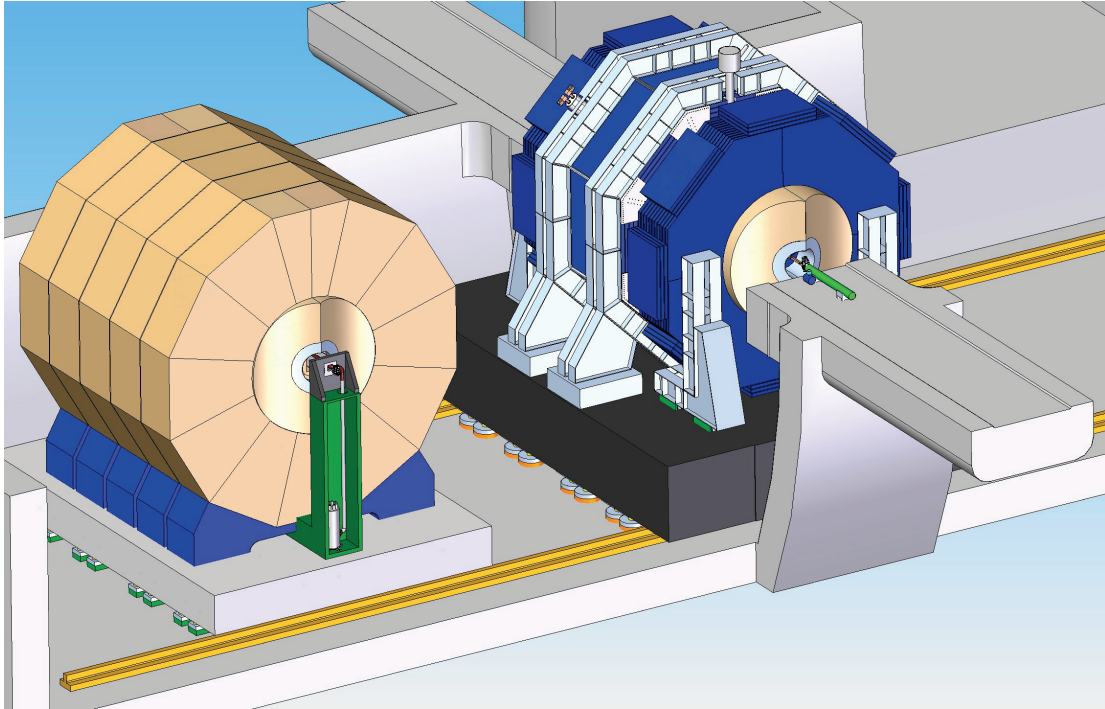


FIGURE 2.5 – Schéma représentant le mécanisme de *push-pull* permettant d'intervertir les détecteurs ILD (à gauche) et SiD (à droite) au point de collision [3].

Détecteur	Mesure	Performance requise
Trajectographe	$\delta(\frac{1}{p})$	$8.10^{-5} (\text{GeV}/c)^{-1}$
Chambres à muons		
Trajectographe + Calorimétrie	$\frac{\Delta E}{E}$	3-4 %
Vertex	Résolution spatiale	$< 3 \mu m$
	Budget matière	$< 0.15 \% X_0/\text{layer}$
	Rayon première couche	$\simeq 1.6 \text{ cm}$

TABLE 2.4 – Table résumant les performances requises pour les sous-détecteurs des détecteurs ILD et SiD [3].

La version de l'algorithme de suivi de particules utilisée à l'ILC requiert une grande granularité dans les calorimètres afin d'identifier correctement les dépôts d'énergie des différentes particules. Cependant, une grande granularité entraîne un nombre important de voies et donc une consommation électrique importante, ce qui pourrait induire un bruit non négligeable. L'utilisation d'un système de refroidissement augmenterait le budget matériel dans les différents sous-détecteurs et impacterait les performances de la reconstruction. Afin de limiter la consommation d'électricité, les différents sous-détecteurs utiliseront une alimentation pulsée

qui permettra d'éteindre les composants électroniques consommant trop d'électricité au moment où les faisceaux de particules sont éteints, soit durant 199.05 ms sans croisement. Le trajectographe devra quant à lui présenter un budget matériel très faible afin d'éviter d'éventuelles interactions avant l'entrée des particules dans les calorimètres.

Les détecteurs **ILD** (**I**nternational **L**arge **D**etector) et **SiD** (**S**ilicon **D**etector) sont des détecteurs généralistes de physique des particules avec une structure en couche d'oignon. On y trouve un détecteur de vertex, un trajectographe, un calorimètre électromagnétique, un calorimètre hadronique et des chambres à muons. Leurs structures génériques, la grande granularité des calorimètres et l'utilisation de l'algorithme de suivi de particule, permettront d'obtenir des performances inégalées quant à la résolution en énergie des jets et donc de mener à bien les différentes mesures exposées dans la section 1.2.

Afin de réaliser le programme physique de l'ILC, des performances minimales ont été fixées pour certains des sous détecteurs. Ces performances sont résumées dans la table 2.4.

2.2.1 Le Grand Détecteur International ILD

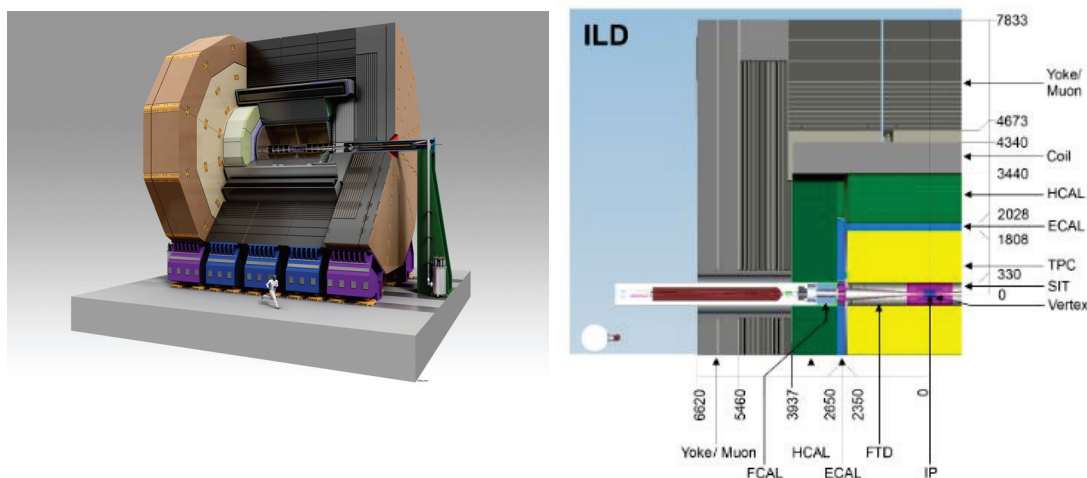


FIGURE 2.6 – À gauche : vue globale du détecteur *ILD*. À droite : vue en coupe du détecteur *ILD*, détaillant les différents sous-détecteurs et certaines côtes [3].

Le détecteur **ILD** (**I**nternational **L**arge **D**etector) a été imaginé pour réaliser le programme physique de l'ILC tout en respectant les contraintes techniques exposées précédemment. La figure 2.6 présente une vue globale de ce détecteur (à gauche) et une vue en coupe détaillant les côtes de différents sous-détecteurs (à droite). Ces deux vues sont extraites du rapport de conception technique (TDR, Technical Design Report) publié en 2013 [3]. Les différentes technologies et dimensions des sous-détecteurs présentées dans ce rapport constituent une référence plutôt mature pour l'ILC.

La description qui suit dans cette section, détaillera chacun des sous-détecteurs de l'ILD, comme présentés dans le rapport de conception technique [3]. Des études d'optimisation sont toujours en cours afin d'obtenir les meilleures performances physiques en fonction des différents paramètres des sous-détecteurs. Ces optimisations seront elles aussi présentées avec les différents sous-détecteurs.

Au niveau de sa conception globale, le détecteur ILD est composé d'un détecteur de vertex, d'un trajectographe et de calorimètres plongés dans un champ magnétique de $3.5 T$. Les chambres à muons sont elles insérées dans une culasse utilisée pour obtenir un retour de champ vers l'intérieur du détecteur. Le champ sera donc moins important dans cette région.

Le détecteur de vertex

Le détecteur de vertex est le détecteur le plus proche du point de collision. Il sert à reconstruire les vertex primaires issus de la collision et les vertex secondaires des différentes désintégrations sous-jacentes des particules à court temps de vie. C'est par exemple le cas des mésons B_0 ($c\tau = 455.4 \mu m$) et D_0 ($c\tau = 122.9 \mu m$) [38].

Le détecteur de vertex de référence dans l'ILD serait constitué de 3 couches cylindriques disposant sur chacune des 2 faces d'un détecteur à silicium, le tout pour obtenir jusqu'à 6 points de mesure par particule chargée. La table 2.5 résume les différents paramètres proposés pour les différents plans du détecteur de vertex de l'ILD [3]. Le premier plan de la première couche se situe à un rayon de 16 mm du point d'interaction et possède une résolution spatiale de $2.8 \mu m$, ce qui remplit bien les conditions de performances requises pour un détecteur de vertex à l'ILC. Le budget matériel de $\sim 0.15\% X_0/plan$ remplit lui aussi les pré-requis. La première couche du détecteur de vertex devra cependant être capable de résister à une dose de $1 kGy/an$.

	R (mm)	$ z $ (mm)	$ \cos(\theta) $	σ (μm)	Temps de lecture (μm)
Plan 1	16	62.5	0.97	2.8	50
Plan 2	18	62.5	0.96	6	10
Plan 3	37	125	0.96	4	100
Plan 4	39	125	0.95	4	100
Plan 5	58	125	0.91	4	100
Plan 6	60	125	0.90	4	100

TABLE 2.5 – Table résumant les paramètres des différentes couches de détection du détecteur de vertex de l'ILD : le rayon du plan, la demi-longueur du plan, l'angle maximal du plan, la résolution spatiale des pixels et le temps de lecture des pixels.

Trois options ont été développées comme candidates au détecteur de vertex de l'ILD : les capteurs CMOS (**CPS** : **CMOS Pixel Sensors**), les capteurs CCD

(**FPCCD** : **F**ine **P**ixel **C**CD) et les capteurs DEPFET (**DEPFET** : **D**epleted **f**ield **e**ffect **t**ransistor).

La chambre à projection temporelle

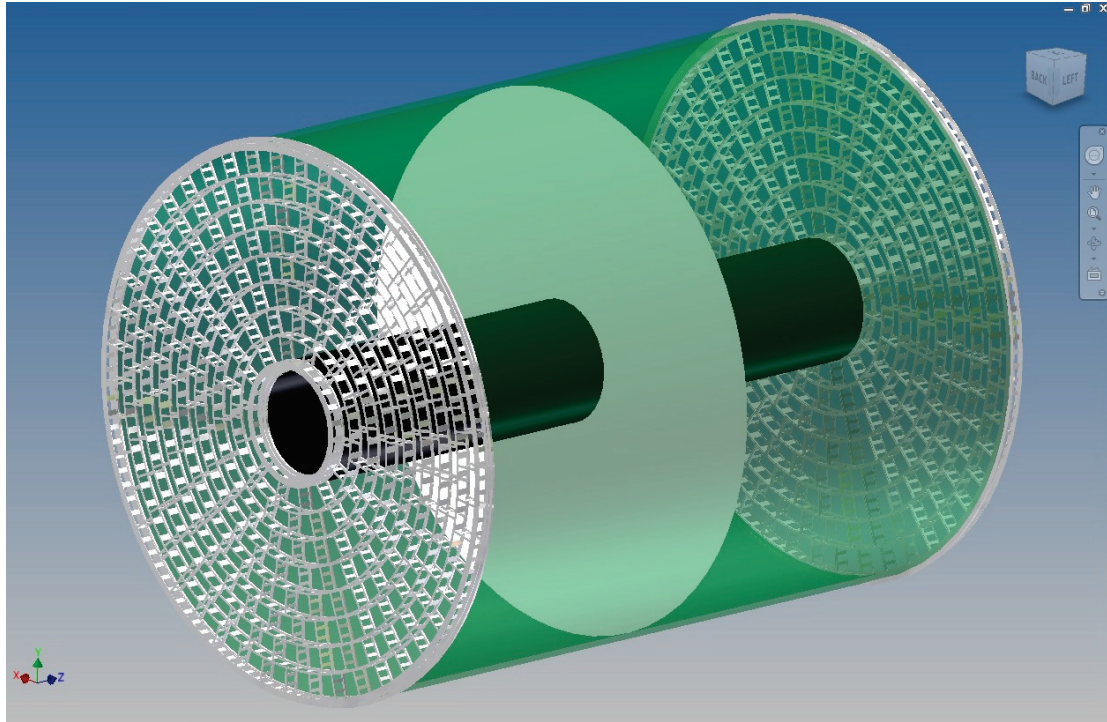


FIGURE 2.7 – Schéma illustrant la chambre à projection temporelle du détecteur ILC.

Principal trajectographe du détecteur ILC, la TPC (**T**ime **P**rojection **C**hamber) reconstruit les traces des particules chargées qui la traversent. Ce détecteur est principalement constitué d'une chambre remplie de gaz que les particules chargées ionisent sur leurs passages. Un champ électrique appliqué dans toute la chambre et parallèle à l'axe du faisceau (\vec{z}), permet de rabattre les charges sur les bouchons. En arrivant aux extrémités des bouchons, ces charges sont multipliées et lues par un détecteur.

La figure 2.7 illustre la chambre à projection temporelle avec les deux zones actives (en blanc) de part et d'autre du détecteur dans les bouchons. Deux options ont été proposées pour équiper ces deux zones : des détecteurs GEM (**G**az **E**lectron **M**ultiplier) ou Micromegas. Dans les deux cas, la lecture des signaux s'effectuera sur des carreaux de cuivre de $1 \times 6 \text{ mm}^2$ avec un total de 10^6 voies de lecture de chaque côté de la TPC. Les informations spatiales (r, ϕ, z) et temporelles de chaque point couplées à la vitesse de dérive des charges dans le gaz permettent de reconstruire individuellement chaque point dans la TPC. Des algorithmes de reconnaissances de

Paramètre			
Paramètres géométriques	r_{in}	r_{out}	z
	329 mm	1808 mm	± 2350 mm
Couverture angulaire	jusqu'à $\cos\theta \simeq 0.98$		
Budget matériel de la TPC	$\simeq 0.05 X_0$ dans la direction du tonneau $< 0.25 X_0$ par détecteur par bouchon		
Nombre de voies de lecture	$\simeq 1 - 2 \times 10^6$ par bouchon		
Taille des carreaux de lecture	$\simeq 1 \times 6$ mm ²		
σ_{point} en $r\phi$	$\simeq 60 - 100$ μm		
σ_{point} en rz	$\simeq 0.4 - 1.4$ μm		
Résolution en dE/dX	5 %		
Résolution en impulsion	$\delta(1/p_t) \simeq 10^{-4} (\text{GeV}.c)^{-1}$		

TABLE 2.6 – Principaux paramètres de la TPC proposée pour équiper le détecteur ILD.

formes sont ensuite appliqués sur l'ensemble des points pour reconstruire les traces laissées par les particules chargées.

La chambre à projection temporelle est un détecteur qui permet l'application de l'algorithme de suivi de particules. En effet, son budget matière est très faible ce qui limite l'interaction des particules avant les calorimètres. Comparée à d'autres trajectographes, et notamment ceux en silicium, la TPC possède une moins bonne résolution spatiale sur la position des points reconstruits. Ceci est néanmoins compensé par le nombre de points plus important que peut laisser une trace dans une TPC. Un aspect important des TPC est aussi la possibilité de mesurer la perte d'énergie dE/dX laissée par chacune des particules dans le gaz. Cette information est utilisée par les techniques d'identification de particules après la reconstruction de l'événement. La table 2.6 présente les principaux paramètres de la TPC proposée pour équiper le détecteur ILD.

Les trajectographes complémentaires

Quatre détecteurs en silicium viendront compléter le système de trajectographie de l'ILD :

- le SIT (Silicon Internal Tracker), placé entre le détecteur de vertex et la TPC dans le tonneau, permet d'améliorer la reconstruction des traces entre ces deux détecteurs
- le SET (Silicon External Tracker), placé entre la TPC et le calorimètre électromagnétique dans le tonneau, permet d'améliorer la précision sur le point d'entrée des traces dans les calorimètres

SIT (bandes croisées, double-face, silicium)					
Géométrie			Caractéristiques		Matériau
R [mm]	Z [mm]	$\cos\theta$	Résolution $R - \phi$ [μm]	Temps [ns]	X_0 [%]
153	368	0.910	$R : \sigma = 7.0$	307.7	0.65
300	644	0.902	$z : \sigma = 50.0$	$\sigma = 80.0$	0.65
SET (bandes croisées, double-face, silicium)					
Géométrie			Caractéristiques		Matériau
R [mm]	Z [mm]	$\cos\theta$	Résolution $R - \phi$ [μm]	Temps [ns]	X_0 [%]
1811	2350	0.789	$R : \sigma = 7.0$	307.7	0.65
ETD (bandes croisées, face unique, silicium)					
Géométrie			Caractéristiques		Matériau
R [mm]	Z [mm]	$\cos\theta$	Résolution $R - \phi$ [μm]		X_0 [%]
419.3-1822.7	2420	0.985-0.799	$x : \sigma = 7.0$		0.65
FTD (pixels pour les deux premiers disques, bandes pour les autres, silicium)					
Géométrie			Caractéristiques		Matériau
R [mm]	Z [mm]	$\cos\theta$	Résolution $R - \phi$ [μm]		RL [%]
39-134	220	0.985-0.802			0.25-0.5
49.6-164	371.3	0.991-0.914		$\sigma = 3 - 6$	0.25-0.5
70.1-308	644.9	0.994-0.902			0.65
100.3-309	1046.1	0.994-0.959			0.65
130.4-309	1447.3	0.995-0.998		$\sigma = 7.0$	0.65
160.5-309	1848.5	0.996-0.986			0.65
190.5-309	2250	0.996-0.990			0.65

TABLE 2.7 – Table résumant les paramètres des trajectographes en silicium, complémentaires à la TPC et au détecteur de vertex du détecteur ILD.

- l’ETD (Endcap Tracking Detector), placé entre la TPC et le calorimètre électromagnétique dans les bouchons, jouera le même rôle que le SET
- le FTD (Forward Tracking Detector), placé le long de la ligne de faisceau de part et d’autre du détecteur de vertex, permet d’étendre la couverture angulaire de la TPC pour des angles faibles ($\cos\theta > 0.802$)

Pour les détecteurs SIT, SET et ETD, des bandes croisées de silicium de $10\text{ cm} \times 50\ \mu\text{m}$ assurent la détection des particules chargées tandis que des pixels en silicium seront utilisés pour le FTD. Ces détecteurs sont aussi utilisés pour leur très bonne résolution en temps et servent à horodater les traces et à les assigner à un paquet de particules précis.

La table 2.7 résume les paramètres de ces quatre détecteurs.

Les calorimètres

Les calorimètres que l’on trouve traditionnellement en physique des particules possèdent une faible segmentation transverse et bien souvent une segmentation longitudinale très faible, voir absente (une seule couche de détection). La performance

clé de ces calorimètres réside cependant dans leur résolution en énergie.

Les calorimètres de l'ILC sont conçus pour permettre l'application d'un algorithme de suivi de particule. Un important pré-requis est une grande segmentation transverse et longitudinale pour pouvoir identifier de façon très précise les dépôts d'énergie de chaque particule. Même si la segmentation est un paramètre important pour les calorimètres de l'ILC, la résolution en énergie doit elle aussi être correcte. Les techniques de compensation logicielles utilisant la topologie des gerbes électromagnétiques et hadroniques, rendu exploitable grâce à la grande granularité des calorimètres, permettront d'améliorer ces résolutions. La grande segmentation transverse des calorimètres a fait opter pour des calorimètres à échantillonnage, qu'ils soient électromagnétiques ou hadroniques. Au niveau des matériaux, le tungstène sera utilisé en tant qu'absorbeur pour les calorimètres électromagnétiques, sa longueur de radiation élevée ($X_0 = 0.35 \text{ cm}$) permettant de contenir en grande partie le développement des gerbes électromagnétiques. Son faible rayon de Molière (voir chapitre 3 section 1.2) permettra de mieux séparer les photons et les électrons proches des autres particules. L'acier sera quant à lui utilisé pour l'absorbeur des calorimètres hadroniques pour sa longueur d'interaction élevée ($\lambda_0 = 16.77 \text{ cm}$).

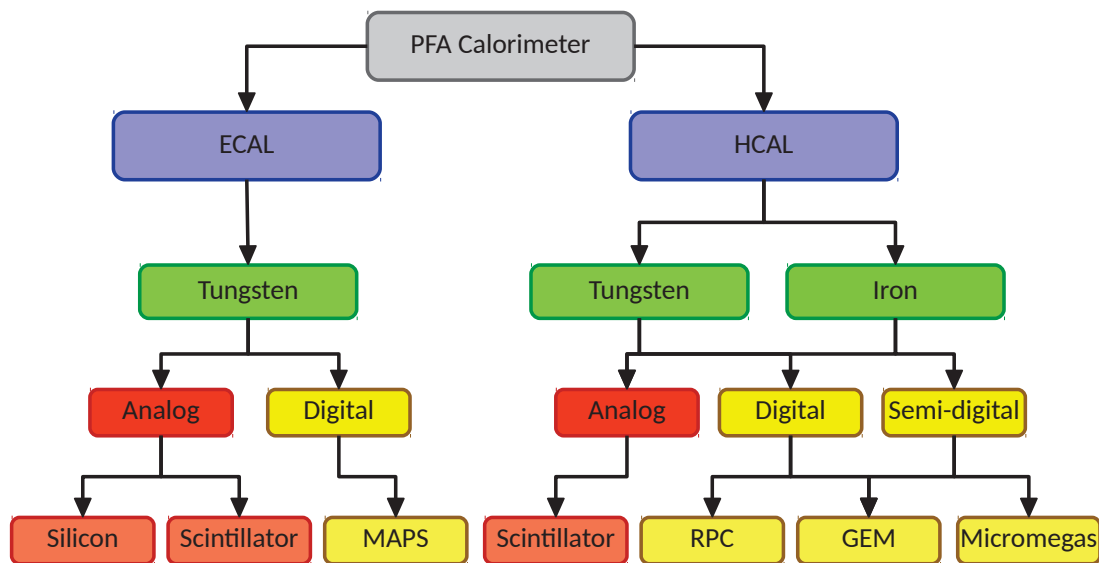


FIGURE 2.8 – Schéma illustrant les différents types de calorimètres électromagnétiques et hadroniques éligibles à l'ILC.

La figure 2.8 illustre les différentes options de calorimètres électromagnétiques et hadroniques pour équiper le système calorimétrique des détecteurs de l'ILC.

Le calorimètre électromagnétique

Trois options sont proposées pour équiper le calorimètre électromagnétique de l'ILD. Toutes trois ont cependant le même rôle : identifier les photons et les électrons et mesurer leurs énergie. La capacité à séparer les photons d'autres particules proches

sera aussi d'une grande importance. La grande granularité du ECAL permettra aussi de séparer les hadrons proches, dont les gerbes continueront la plupart du temps dans le calorimètre hadronique. Les trois options sont les suivantes :

- le **SiECal** utilise des carreaux de silicium d'une segmentation transverse de $5 \times 5 \text{ mm}^2$ et d'une épaisseur de $330 \text{ }\mu\text{m}$ pour une lecture analogique. Le nombre total de voies électroniques avec les dimensions prévues pour l'ILD s'élèverait à environ 100 millions. Afin de tester et valider ce détecteur, un prototype physique a été construit et testé sur faisceau [18]. Fait de 30 couches d'une aire active de $18 \times 18 \text{ cm}^2$ et d'une granularité de $10 \times 10 \text{ mm}^2$ ($\sim 10k$ voies), les résultats pour ce prototype montrent une résolution en énergie de $\sigma_E/E = 16.6\%/\sqrt{E(\text{GeV})} \oplus 1.1\%$.
- le **ScECal** utilise des bandes croisées de plastique scintillant d'une taille de $5 \times 45 \text{ mm}^2$ pour une lecture analogique. Le fait de croiser ces bandes permet de réduire grandement le nombre de voies électroniques utilisées tout en obtenant une segmentation de $5 \times 5 \text{ mm}^2$. L'identification des dépôts sera cependant plus complexe et devra faire appel à un algorithme de reconstruction appelé SSA (*Strip Splitting Algorithm*). Un prototype physique pour ce détecteur a aussi été construit et testé sur faisceau [23]. Il est composé de 30 couches contenant chacune 72 bandes de taille $10 \times 45 \text{ mm}^2$. Les résultats montrent une résolution en énergie de $\sigma_E/E = 12.9\%/\sqrt{E(\text{GeV})} \oplus 1.2\%$ sur des électrons de $2 - 32 \text{ GeV}$.
- l'option **MAPS** (*Monolithic Active Pixel Sensor*) utilise quand à lui des pixels en silicium de taille $50 \times 50 \text{ }\mu\text{m}^2$ pour une lecture digitale. Aucun prototype n'a encore été construit à ce jour mais des simulations **Geant4** montrent des résolutions en énergie allant jusqu'à $\sigma_E/E = 12.8\%/\sqrt{E(\text{GeV})} \oplus 1.3\%$ [37].

Les paramètres principaux du calorimètre électromagnétique qui sera installé dans l'ILD sont encore en cours d'optimisation. Parmi ces paramètres, on peut citer le rayon interne du calorimètre, le nombre de couches ou le nombres de bandes de scintillateur. Les performances extraites en faisant varier ces paramètres seront discutées dans la section 1 du chapitre 4 dédié aux algorithmes de suivi de particules.

Le calorimètre hadronique

Le but du calorimètre hadronique dans l'ILD est de séparer les dépôts d'énergie des hadrons neutres et chargés. L'identification des hadrons neutres constitue le principal défi dans les techniques de reconstruction par suivi de particules. Une granularité élevée permettra de séparer ces dépôts et de mesurer leur énergie, l'énergie des hadrons chargés étant principalement évaluée par le trajectographe après reconstruction. Le rapport de conception technique propose d'équiper l'ILD avec un calorimètre hadronique à échantillonnage de 48 couches avec des absorbeurs en acier. Les deux options sont les suivantes :

- le **AHCAL** (*Analog Hadron Calorimeter*) qui utilise des tuiles scintillantes en plastique avec une segmentation de $30 \times 30 \text{ mm}^2$ pour une lecture analogique.

Un prototype physique a été construit et testé de nombreuses fois sur faisceau. Le prototype adopte cependant une segmentation transverse particulière avec des tuiles de $30 \times 30 \text{ mm}^2$ dans la région centrale (10×10 cellules), puis $60 \times 60 \text{ mm}^2$ et $120 \times 120 \text{ mm}^2$ en allant vers l'extérieur, le tout développé sur 38 couches pour un total de près de 300k cellules. La résolution en énergie atteinte est de $\sigma_E/E = 57\%/\sqrt{E(\text{GeV})} \oplus 1.6\%$ et peut descendre jusqu'à $\sigma_E/E = 44.3\%/\sqrt{E(\text{GeV})} \oplus 1.8\%$ avec des techniques de compensation logicielle.

- le **SDHCAL** (Semi-Digital Hadron Calorimeter) utilisant des chambres à plaque résistives de verre (GRPC) avec une segmentation de $10 \times 10 \text{ mm}^2$ pour une lecture semi-digitale. Une description plus détaillée sera présentée dans le chapitre 3.

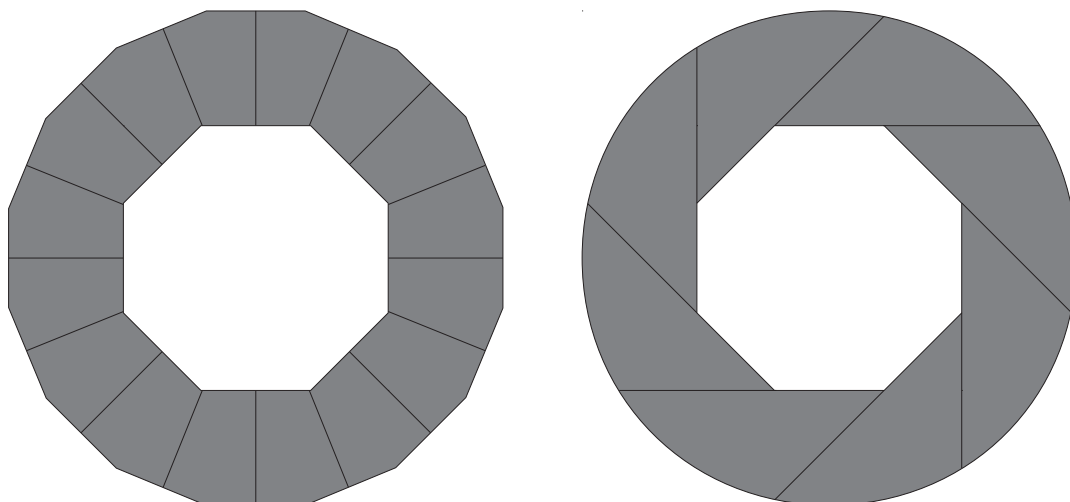


FIGURE 2.9 – Schéma illustrant les deux structures géométriques proposées pour le calorimètre hadronique dans la région du tonneau : la géométrie TESLA (à gauche) et la géométrie VIDEAU (à droite).

Deux structures géométriques ont été proposées pour le calorimètre hadronique dans la région du tonneau. La figure 2.9 illustre ces deux géométries. La première (à gauche), connue sous le nom de géométrie TESLA, est divisée en 16 modules suivant ϕ et conçue pour accueillir le calorimètre AHCAL. Cette structure est divisée en deux parties, de part et d'autre du faisceau selon z , pour un total de 32 modules. La deuxième (à droite), connue sous le nom de géométrie VIDEAU, est divisée en 8 modules se chevauchant et conçue pour accueillir le calorimètre SDHCAL. Cette structure est, elle, divisée en 5 le long du faisceau, pour un total de 40 modules indépendants. A noter que cette dernière est optimisée pour réduire les zones non-instrumentées que les particules pourraient traverser.

Les calorimètres vers l'avant

Trois calorimètres supplémentaires placés autour de la ligne de faisceau compléteront ceux installés dans les bouchons :

- le LumiCal, qui mesurera la luminosité
- le BeamCal, qui mesurera le bruit photon-photon issu de la ligne de faisceau
- le LHCAL, qui comblera l'espace non instrumenté entre la ligne de faisceau et le calorimètre hadronique dans les bouchons.

Les chambres à muons

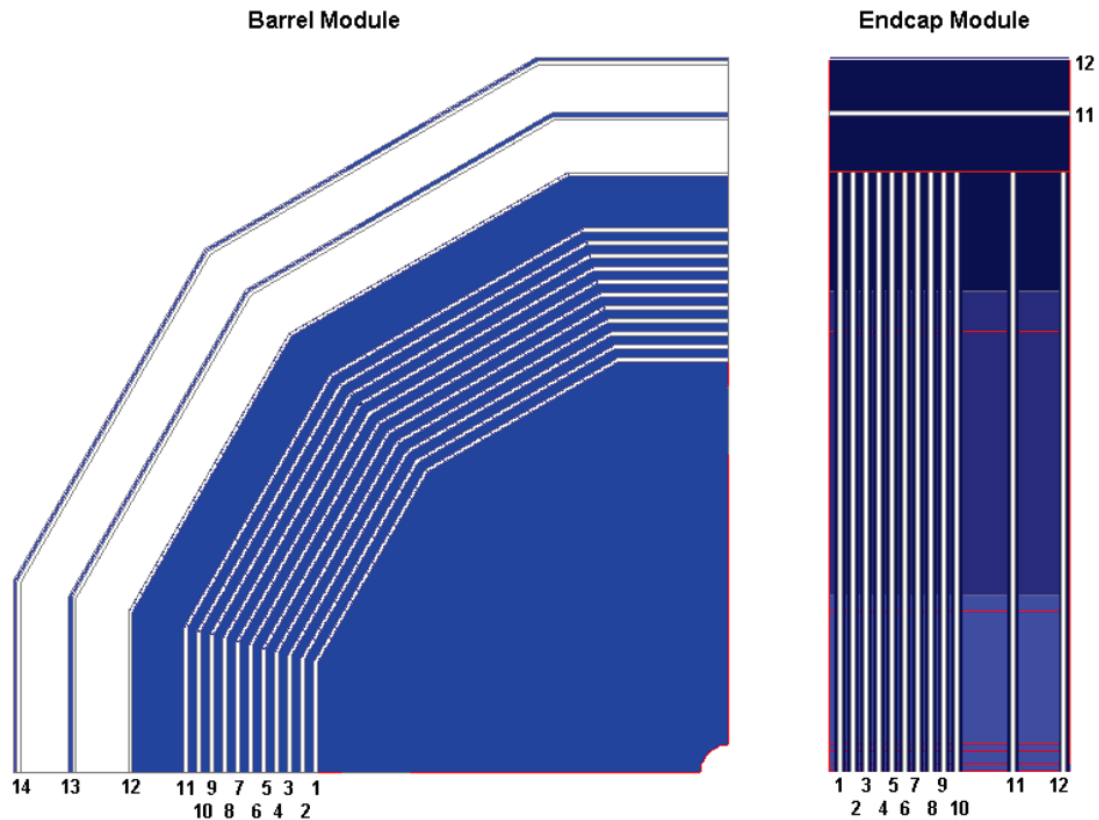


FIGURE 2.10 – Schéma illustrant la structure des chambres à muons dans le tonneau (à gauche) et dans les bouchons (à droite) dans le détecteur ILC.

Les chambres à muons seront placées à l'extérieur du détecteur après l'aimant ($\sim 2\lambda_I$), dans une culasse en fer, permettant le retour de champ. Ils serviront à reconstruire la trace des muons et à mesurer leur impulsion mais aussi à mesurer l'énergie qui pourrait s'échapper du calorimètre hadronique (*Tail Catcher*). Au niveau des technologies candidates, on trouve les RPCs (*Resistive Plate Chamber*), très souvent utilisées pour la reconstruction des muons dans les autres expériences

de physique des particules ou encore des bandes scintillantes couplées à des photomultiplicateurs en silicium. La figure 2.10 illustre la structure des chambres à muons dans le détecteur ILD.

2.2.2 Le Détecteur au Silicium SiD

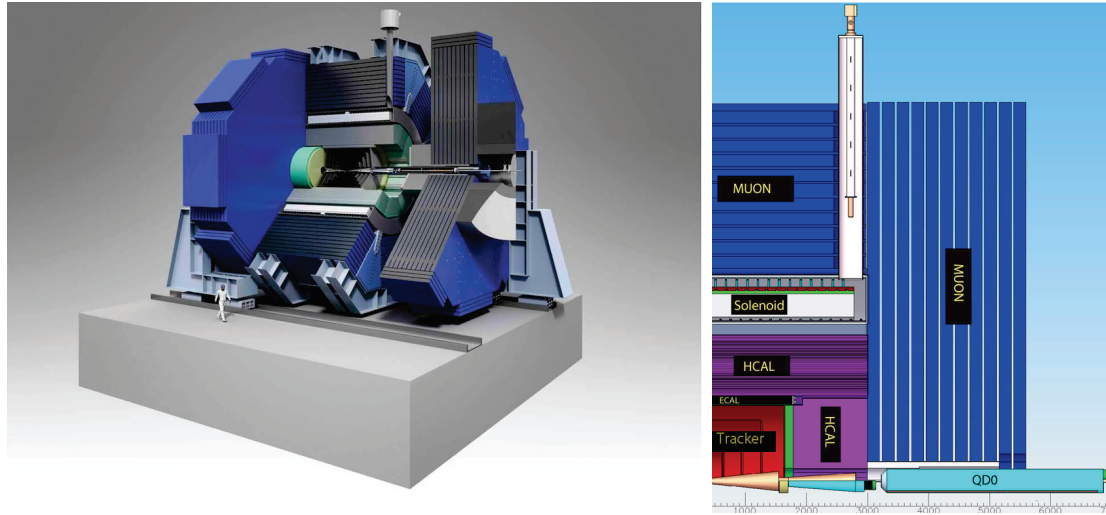


FIGURE 2.11 – Le détecteur SiD. À gauche : vue globale du détecteur. À droite : vue coupe du détecteur ; code couleur : trajectographe (rouge), ECAL (vert), HCAL (violet) et chambres à muons (bleu.)

Tonneau	R (cm)	Couverture des capteurs (cm)	Nombre de modules en ϕ	Nombre de modules en z
Couche 1	21.95	11.6	20	13
Couche 2	46.95	147.3	38	17
Couche 3	71.95	200.1	58	23
Couche 4	96.95	251.8	80	29
Couche 5	121.95	304.5	102	35
Disques (bouchons)	z_{inner} (cm)	R_{inner} (cm)	R_{outer} (cm)	Nombre de modules
Disque 1	78.89	20.89	48.8	96
Disque 2	107.5	20.89	75.14	238
Disque 3	135.55	20.89	100.31	438
Disque 4	164.09	20.89	125.36	662

TABLE 2.8 – Paramètres géométriques principaux du trajectographe en silicium du détecteur SiD.

Le Détecteur au Silicium (SiD) sera le deuxième détecteur placé sur la ligne de faisceau en alternance avec le détecteur ILD. Il est lui aussi conçu pour appliquer

un algorithme de suivi de particules pour la reconstruction. On retrouvera donc le même type de technologies côté calorimétrie.

Deux différences sont cependant importantes entre ces deux détecteurs. Premièrement, le trajectographe n'est pas une TPC mais un trajectographe en silicium, composé de 5 couches concentriques dans le tonneau et 4 disques par bouchon. Les paramètres de ce trajectographe sont résumés dans la table 2.8 [3]. À noter que ce détecteur est moins sensible au bruit venant de la ligne faisceau que la TPC du détecteur ILD. Cependant, le rayon externe du trajectographe est plus faible (1219.5 mm contre 1808 mm pour la TPC de l'ILD) et le budget matière est plus important ($0.1 - 0.25 X_0$) que celui du trajectographe de l'ILD. Ce qui amène à la deuxième différence avec l'ILD : un champ magnétique plus élevé, de 5T (contre 3.5 à 4 T pour l'ILD), pour accentuer la courbure des particules chargées et ainsi contrebalancer le rayon plus faible.

3 Les autres projets de collisionneurs leptoniques

Le collisionneur linéaire international ILC n'est pas le seul projet de succession au LHC. D'autres projets de collisionneurs leptoniques pourraient venir préciser les mesures faites au LHC et tester le modèle standard :

- le *Collisionneur Linéaire Compact (CLIC)*, est un projet de collisionneur linéaire électron-positon proposé par le CERN. Il est conçu pour atteindre une énergie au centre de masse entre 1 et 3 TeV. Les technologies utilisées dans le complexe d'accélération pour atteindre de telles énergies sont très différentes de celles utilisées dans l'ILC, les cavités supraconductrices ne suffisant pas à fournir le gradient moyen d'accélération de 100 MV/m nécessaire. La technologie développée par le projet CLIC utiliserait deux faisceaux, le premier, de faible intensité, étant utilisé comme apport d'énergie au second faisceau. Cette technique permettrait d'atteindre des énergies jusqu'à 5 TeV. La longueur du collisionneur est alors estimée à environ 43km pour une version à 3 TeV. Un seul détecteur serait placé au point de collision, avec le même type de sous-détecteurs que pour l'ILC : grande granularité dans les calorimètres et permettant l'application des algorithmes de suivi de particules.
- le *Future Circular Collider (FCC)*, est un projet de collisionneur circulaire aussi proposé par le CERN. Avec une circonférence d'environ 100 km, ce collisionneur propose trois concepts. Le premier serait de collisionner des électrons et des positons (FCC-ee) avec une énergie au centre de masse entre 90 et 350 GeV avec le même programme physique que l'ILC pour ces énergies. Cependant, le fait de construire un tunnel circulaire permet aussi d'effectuer des collisions hadroniques. C'est le deuxième scénario proposé : deux faisceaux de protons avec une énergie au centre de masse inégalée de l'ordre de 100 TeV collisionneraient pour explorer la physique au delà de l'échelle du TeV. Le troisième concept propose des collisions inélastiques protons/électrons à 50 TeV/60 GeV pour étudier les interactions quark-gluon et chercher une éventuelle sous-structure aux particules élémentaires existantes. Les dé-

tecteurs au point de collision sont encore à l'étude et ne seront pas forcément conçus pour l'application d'un algorithme de suivi de particules.

- le *Circular Electron Positron Collider* (**CEPC**) est un projet de collisionneur circulaire électron-positon, de 50 à 70 km de circonférence, développé par la Chine. Avec une énergie au centre de masse de 250 GeV, son principal intérêt serait de fournir des bosons de Higgs à leur seuil de production (*Higgs factory*). L'utilisation d'une ligne circulaire fournirait plusieurs points de collisions et donc plusieurs installations de détecteurs. Ces derniers seraient aussi conçus, comme pour l'ILC, avec des calorimètres hautement granulaires pour l'application d'un algorithme de suivi de particules.

Chapitre 3

Le calorimètre hadronique semi-digital SDHCAL

AN calorimètre hadronique avec une lecture semi-digitale a été proposé pour équiper le détecteur ILD de l'expérience ILC. Un prototype a été construit, en grande partie à Lyon, d'une taille d'environ un mètre cube et exposé à différents faisceaux de particules au CERN sur les lignes H2 du SPS et H6 du SPS ainsi qu'au PS. Ce chapitre a pour but de décrire ce prototype et de discuter les performances atteintes lors des tests en faisceau. Après une présentation du phénomène de cascade électromagnétique et hadronique, une revue des différents types de calorimètres en physique des hautes énergies sera faite. Le prototype technologique du SDHCAL sera décrit ainsi que la simulation numérique basée sur *Geant4* et utilisée pour la comparaison avec les données. Ses performances seront finalement exposées.

Contenu

1	Le phénomène de gerbe électromagnétique et hadronique	32
1.1	Interactions des particules dans la matière	32
1.2	Les gerbes électromagnétiques	34
1.3	Les gerbes hadroniques	34
2	La calorimétrie en physique des particules	36
3	Le prototype du SDHCAL	38
3.1	Description du prototype	38
3.2	Les chambres à plaques résistives	39
3.3	L'électronique de lecture	41
3.4	Le système d'acquisition	42
3.5	Simulation numérique du prototype	43
4	Les performances du SDHCAL	44
4.1	Reconstruction des événements physiques	45
4.2	Les performances du détecteur	46
4.3	Reconstruction de l'énergie des hadrons	49
4.4	Des désaccords entre données et simulation	61
5	Conclusion	62

1 Le phénomène de gerbe électromagnétique et hadronique

1.1 Interactions des particules dans la matière

Lors de la traversée d'une particule dans la matière, celle-ci peut interagir avec le milieu. Différents processus peuvent causer cette interaction. Dans le cas d'un processus non-destructif, la particule perdra nécessairement de l'énergie. Dans le cas de particules chargées, la perte d'énergie moyenne est donnée par la formule de Bethe-Block [38] :

$$\left\langle -\frac{dE}{dx} \right\rangle = K z^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \frac{2m_e c^2 \beta^2 \gamma^2 T_{max}}{I^2} - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right]. \quad (3.1)$$

avec z le nombre de charge de la particule, m_e la masse de l'électron, Z le numéro atomique de la cible dans le milieu, A sa masse atomique (en $g.mol^{-1}$), T_{max} l'énergie cinétique maximale que peut emporter un électron après une collision, I le potentiel d'excitation moyen du matériau et δ un facteur multiplicatif prenant en compte la densité du matériau traversé. Les autres sont des constantes : K vaut $4\pi N_A r_e^2 m_e c^2$ et le facteur γ relativiste vaut $\frac{1}{\sqrt{1-\beta^2}}$ avec $\beta = \frac{v}{c}$. Son domaine de validité s'étend sur $1 \leq \gamma\beta \leq 1000$ avec une erreur inférieure à 1%.

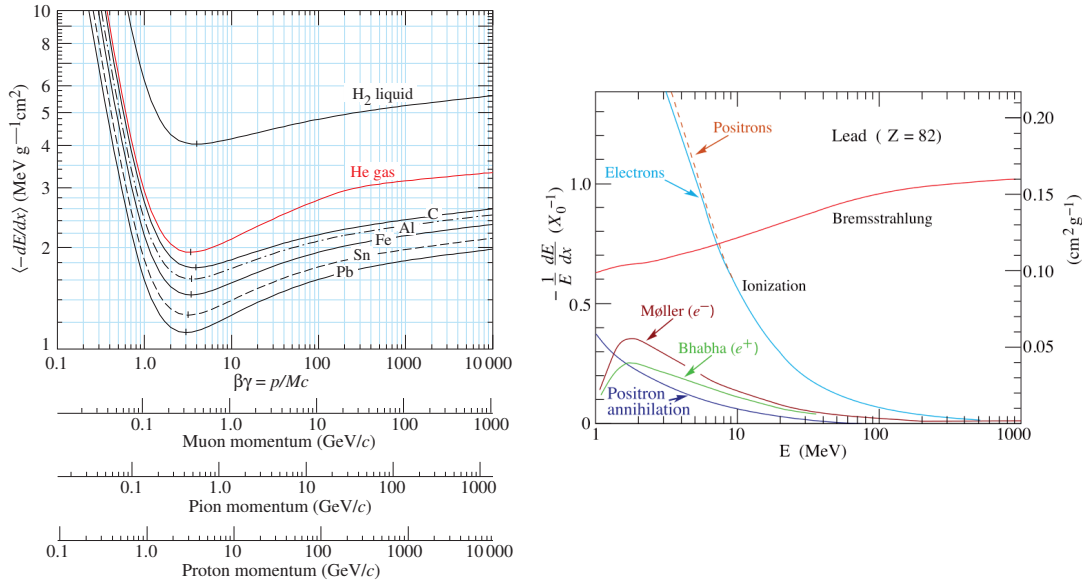


FIGURE 3.1 – À gauche : Courbes de perte d'énergie pour différents matériaux traversés par des muons, des pions et des protons en fonction de leur impulsion. À droite : Courbes de perte d'énergie des électrons et des positons pour différents processus dans le plomb [38].

Les électrons et les positons interagissent de manière légèrement différente dans les matériaux. Des diffusions peuvent avoir lieu (diffusion Møller pour les électrons

et diffusion Bhabha pour les positons) ainsi que des annihilations pour les positons de basse énergie ($E < 100 \text{ MeV}$) donnant naissance à deux photons ayant chacun une énergie $E_\gamma = 511 \text{ keV}$. La figure 3.1 à gauche montre quelques exemples de courbe de perte d'énergie dans différents matériaux pour les muons, les pions et les protons en fonction de $\beta\gamma$ (sans Bremsstrahlung). La figure 3.1 à droite montre quant à elle l'énergie perdue par les électrons et les positons dans le plomb pour différents processus.

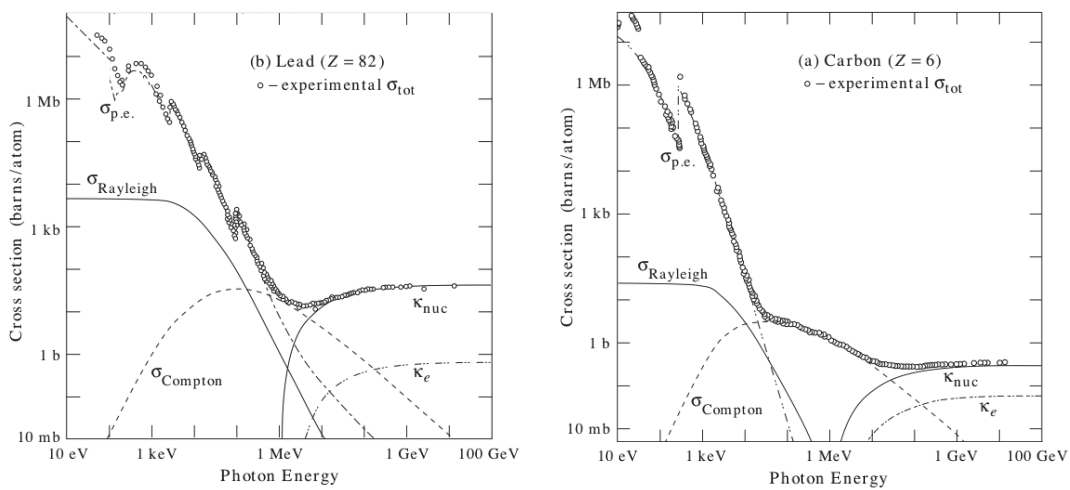


FIGURE 3.2 – Sections efficaces de différents processus dans le plomb (à gauche) et le carbone (à droite) en fonction de l'énergie incidente du photon : $\sigma_{p.e.}$ pour l'effet photoélectrique, $\sigma_{Compton}$ pour la diffusion Compton, $\sigma_{Rayleigh}$ pour la diffusion Rayleigh, κ_{nuc} pour l'interaction photo-nucléaire et κ_e pour la création de paires [38].

Les photons peuvent interagir de manière destructive ou non-destructive. Les diffusions Rayleigh et Compton sont toutes deux des interactions non-destructives qui amènent le photon incident à dévier de sa trajectoire avec (Compton) ou sans (Rayleigh) perte d'énergie. L'effet photoélectrique convertit quant à lui le photon incident en un électron issu du nuage électronique du matériau (interaction destructive). Les deux autres interactions possibles sont l'interaction photonucléaire et la création de paires. Le premier est une absorption du photon par le noyau du matériau (proton, neutron ou fission nucléaire) tandis que l'autre convertit le photon en une paire électron-positon. Ces processus sont plus ou moins importants en fonction de l'énergie incidente du photon et du matériau. La figure 3.2 montre les sections efficaces de chacun de ces processus en fonction de l'énergie incidente du photon, dans le plomb (à gauche) et le carbone (à droite).

Les hadrons chargés interagissent, en plus de l'ionisation décrite précédemment, de manière forte avec le noyau du matériau et créent de nouvelles particules. Ces dernières peuvent être issues de la fission des noyaux ou de hadrons instables issus de cette fission par exemple. Dans ce cas, la désexcitation pourra s'accompagner de γ ou de particules α . Certaines de ces particules peuvent aussi se désintégrer de manière électromagnétique, comme le π_0 ($\sim 98.8\%$ de chance en $\pi_0 \rightarrow \gamma\gamma$) qui ont un temps de vie très court ($\tau_{\pi_0} = 8.52 \pm 0.18 \times 10^{-17} \text{ s}$).

Les hadrons neutres, tels que les neutrons, interagissent uniquement de manière forte comme décrit précédemment. Les neutrons peuvent, en plus des interactions fortes, être capturés par les noyaux des atomes du matériau. La désexcitation du noyau se fera en émettant un γ .

1.2 Les gerbes électromagnétiques

Les cascades électromagnétiques sont principalement initiées par des électrons, des positons ou des photons. La figure 3.3 illustre le principe d'une gerbe électromagnétique. On y voit un électron rayonner un photon, qui est ensuite converti en paire e^+e^- , etc ... Les photons continuent à se convertir en paire e^+e^- tant que leur énergie est supérieure à 1022 keV . En dessous de cette énergie, ils subiront des processus de basse énergie tels que la diffusion Compton ou encore la diffusion Rayleigh, etc... et finiront par être absorbés par le milieu. Les pions neutres initient aussi des gerbes électromagnétiques, étant donné que le processus principal de désintégration est $\pi_0 \rightarrow \gamma\gamma$

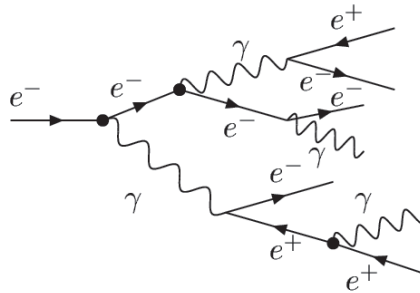


FIGURE 3.3 – Schéma illustrant une gerbe électromagnétique initiée par un électron de haute énergie.

La longueur de radiation X_0 quantifie la longueur moyenne pour qu'un électron dépose 63% de son énergie dans un matériau (souvent exprimé en $g.cm^{-2}$). Par conséquent, chaque matériau possède sa propre longueur de radiation. La table 3.1 donne les longueurs de radiation des matériaux les plus fréquemment utilisés dans la calorimétrie.

1.3 Les gerbes hadroniques

Les gerbes hadroniques sont initiées par l'interaction des hadrons de hautes énergies avec la matière. Ce phénomène est par exemple produit dans les jets dans les expériences de physique de hautes énergies ou encore dans l'atmosphère lorsqu'un rayon cosmique y interagit. Suite à la première interaction du hadron avec la matière, ce dernier va créer de nouvelles particules secondaires qui interagiront à leur tour

saire pour qu'un hadron de haute énergie déclenche une cascade (souvent exprimé en $g.cm^{-2}$). Cette longueur varie en fonction des matériaux mais aussi du hadron incident.

Matériau	Densité ($g.cm^{-3}$)	λ_I ($g.cm^{-2}$)	λ_I (cm)
Carbone (graphite)	2.210	117.8	53.30
Aluminium	2.699	136.7	50.64
Fer	7.874	169.8	20.42
Cuivre	8.960	165.9	18.51
Tungstène	19.30	218.7	11.33
Plomb	11.35	226.2	19.93
Uranium	18.95	235.3	12.42

TABLE 3.2 – *Table des longueurs d'interaction pour les matériaux les plus utilisés en calorimétrie [38].*

La table 3.2 donne les longueurs d'interaction des matériaux les plus fréquemment utilisés dans la calorimétrie.

2 La calorimétrie en physique des particules

Les calorimètres en physique des particules sont des détecteurs utilisés pour mesurer l'énergie des particules chargées et neutres, de manière destructive. Deux types de calorimètres sont utilisés : les calorimètres à échantillonnage ou les calorimètres homogènes. Les premiers voient leurs parties actives séparées par des couches d'absorbeur alors que les deuxièmes voient leur parties actives et leurs absorbeurs confondus. Dans les deux cas, les calorimètres sont des milieux denses, destinés à forcer l'interaction des particules et à créer des gerbes électromagnétiques ou hadroniques. Étant donné les propriétés relativement différentes des gerbes électromagnétiques et hadroniques, deux types de calorimètres leur seront dédiés. Les calorimètres électromagnétiques mesureront l'énergie des photons, des électrons et des positons alors que les calorimètres hadroniques mesureront l'énergie des hadrons. Afin de mesurer au mieux l'énergie de ces particules, leurs cascades devront être au maximum contenues à l'intérieur du calorimètre. La longueur totale de radiation X_0 des calorimètres électromagnétiques et la longueur totale d'interaction λ_I des calorimètres hadroniques seront des paramètres importants. Malheureusement, leur profondeur ne peut être trop grande en raison du coût. Elle devra être optimisée en fonction de l'échelle d'énergie des particules incidentes et des caractéristiques des matériaux utilisés pour construire le calorimètre en question. Généralement, les calorimètres électromagnétiques comptent entre 15 et 30 longueurs de radiation contre 5 à 8 longueurs d'interaction pour les calorimètres hadroniques.

Les performances d'un calorimètre standard sont données par sa résolution en

énergie :

$$\frac{\sigma_E}{E} = \frac{a}{\sqrt{E}} \oplus b \oplus \frac{c}{E} \quad (3.2)$$

où a est le terme stochastique lié aux fluctuations dans les gerbes électromagnétiques et hadroniques, b est le terme asymptotique lié à la saturation du calorimètre à haute énergie ou à l'énergie qui fuit du calorimètre et c est le terme de bruit lié à la calibration et à l'uniformité du détecteur. Les trois termes dans l'équation 3.2 sont ajoutés en quadrature. La fraction d'échantillonnage, définie comme la fraction d'énergie déposée dans les absorbeurs par rapport à l'énergie totale déposée, dominera la résolution en énergie des calorimètres à échantillonnage et sera généralement moins bonne que pour les calorimètres homogènes.

Côté calorimétrie électromagnétique, la résolution en énergie est généralement meilleure étant donnée la fluctuation moins importante au sein des gerbes électromagnétiques. Les versions homogènes utilisent généralement des cristaux scintillants ou des détecteurs en verre riche en plomb alors que les versions à échantillonnage utiliseront par exemple pour la partie active des tuiles en plastique scintillantes ou de l'argon liquide. Les absorbeurs varient aussi d'un calorimètre à l'autre ; les plus utilisés étant l'uranium (ZEUS) ou le plomb (CDF, H1, ATLAS). Le tungstène fait aussi partie des absorbeurs possibles puisqu'il est prévu d'être utilisé comme absorbeur pour le calorimètre électromagnétique de l'ILD.

Côté calorimétrie hadronique, les résolutions en énergie sont moins bonnes, étant données les fluctuations plus importantes attendues dans les cascades hadroniques. Le type de calorimètre traditionnellement utilisé sera celui à échantillonnage. On trouvera principalement pour la partie active des tuiles scintillantes ou des détecteurs gazeux, type chambres à fils (MWPC) ou chambre à plaques résistives (RPC). Les absorbeurs sont généralement de l'uranium ou du fer.

Les tables 3.3 et 3.4 montrent quelques-unes des technologies et leur performances, dans les secteurs de la calorimétrie électromagnétique (table 3.3) et de la calorimétrie hadronique (table 3.4 [38].)

Technologie (Expérience)	Profondeur	Résolution en énergie	Date
$PbWO_4$ (CMS)	$25X_0$	$3\%/\sqrt{E} \oplus 0.5\% \oplus 0.2\%/E$	1997
$Bi_4Ge_3O_{12}$ (L3)	$22X_0$	$2\%/\sqrt{E} \oplus 0.7\%$	1993
$CsI(Tl)$ (BELLE)	$16X_0$	1.7% pour $E_\gamma > 3.5 GeV$	1998
Scintillateur/ U (ZEUS)	$20 - 30X_0$	$18\%/\sqrt{E}$	1988
Scintillateur/ Pb (CDF)	$18X_0$	$13.5\%/\sqrt{E}$	1988
Argon liquide/ Pb (H1)	$20 - 30X_0$	$12\%/\sqrt{E} \oplus 1\%$	1998
Argon liquide/ Pb (ATLAS)	$25X_0$	$10\%/\sqrt{E} \oplus 0.4\% \oplus 0.3/E$	1996

TABLE 3.3 – Table des résolutions en énergie de quelques calorimètres électromagnétiques en physique des particules. Les trois premiers sont des calorimètres homogènes et les quatre suivants sont des calorimètres à échantillonnage.

Technologie (Expérience)	Profondeur	Résolution en énergie	Date
Scintillateur/ U (ZEUS)	$5\lambda_I$	$35\%/\sqrt{E}$	1991
Scintillateur/ Fe (ATLAS)	$7\lambda_I$	$47\%/\sqrt{E} \oplus 2.2\%$	1994
Chambres à fils/ U (L3)	$6.5\lambda_I$	$55\%/\sqrt{E} \oplus 5\%$	1990
Chambres à fils/ Fe (ALEPH)	$7.16\lambda_I$	$80\%/\sqrt{E}$	1990

TABLE 3.4 – Table des résolutions en énergie de quelques calorimètres hadroniques en physique des particules. Tous les calorimètres sont à échantillonnage [38].

3 Le prototype du SDHCAL

3.1 Description du prototype

La collaboration CALICE a développé un calorimètre hadronique avec une lecture semi-digitale [8]. C'est un calorimètre à échantillonnage conçu pour permettre l'application des algorithmes de suivi de particules (PFA). Ses différentes caractéristiques respectent les exigences attendues pour un calorimètre hadronique à l'ILC (voir section 2.2.1 du chapitre 2). Un prototype de ce calorimètre a été construit en 2011, principalement à l'Institut de Physique Nucléaire de Lyon, avec une taille d'environ $1.3 m^3$.

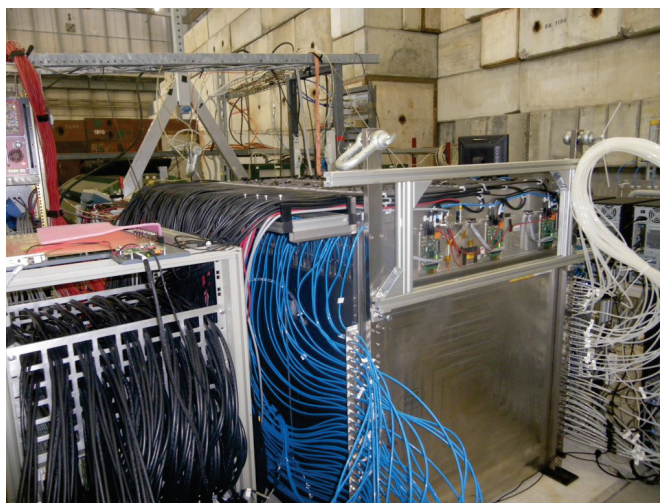


FIGURE 3.5 – Photographie du prototype SDHCAL prise lors d'un test sur faisceau sur la ligne H2 du CERN.

La partie active est composée de chambres à plaques résistives de verre (GRPC) d'une taille de $1 \times 1 m^2$ qui sont insérées dans des cassettes d'aciers de $2 \times 0.25 cm$ d'épaisseur. Chaque chambre (1 cassette + 1 GRPC) est insérée dans une structure auto-portante, entre deux plaques d'acier de $1.5 cm$ d'épaisseur chacune, pour un total de $2 cm$ d'absorbeur à chaque plan. Une couche supplémentaire d'acier précède la première GRPC du calorimètre ($1.75 cm$). Les dimensions du calorimètre sont au

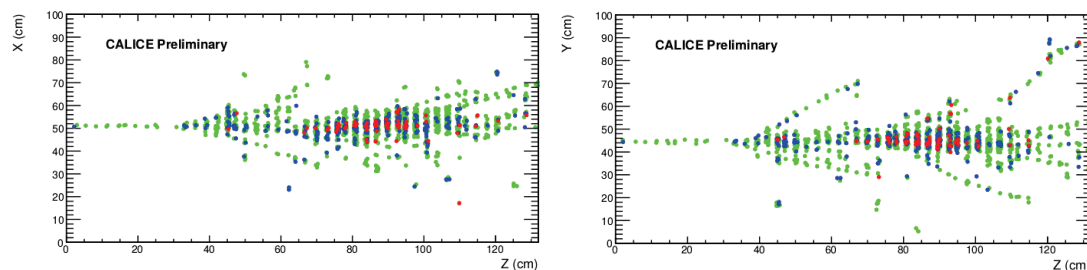


FIGURE 3.6 – Une gerbe hadronique d’un pion de 80 GeV enregistrée au SPS sur la ligne H6 au CERN, par le prototype SDHCAL. Les couleurs correspondent aux seuils : premier seuil en vert, second seuil en bleu et troisième seuil en rouge. Le faisceau est dirigé suivant l’axe \vec{z} .

total $1 \times 1 \times 1.3 \text{ m}^3$, pour un total de $6\lambda_I$. La figure 3.5 montre une photographie du prototype lors d’une campagne de prise de données sur faisceau au CERN.

Les principaux intérêts de ce calorimètre résident dans sa granularité élevée de $1 \times 1 \text{ cm}^2$ et sa lecture semi-digitale. Le mode de lecture semi-digitale, en opposition au mode de lecture digitale, permet d’échantillonner la charge sur plusieurs seuils (ici 3 seuils). Ces seuils permettent, non pas de mesurer directement l’énergie laissée par les particules incidentes traversant la cellule, mais de donner une estimation du nombre de particules qui la franchissent.

La figure 3.6 montre une interaction dans les plans XoZ et YoZ d’un pion de 80 GeV dans le SDHCAL enregistrée lors d’un test sur faisceau. Les trois couleurs correspondent aux différents seuils appliqués sur chaque cellule : vert pour le seuil 1, bleu pour le seuil 2 et rouge pour le seuil 3. On voit tout de suite l’intérêt d’utiliser trois seuils ici : la partie dense, et donc la plus énergétique, de la gerbe, contient en moyenne des seuils plus élevés. Cette information sera par la suite utilisée pour reconstruire l’énergie des particules dans le calorimètre et pour compenser la saturation à haute énergie. Grâce à la granularité élevée du détecteur, il est aussi possible de distinguer des segments de traces laissés par des particules issues de la cascade hadronique mais n’interagissant que plus tardivement. Ce genre d’information sera aussi utilisé lors du traitement de données et de la reconstruction par suivi de particules.

Nous allons maintenant détailler le fonctionnement des chambres à plaques résistives utilisées dans le SDHCAL. L’électronique d’acquisition reliant les GRPCs et le système d’acquisition seront ensuite décrits. Finalement, le fonctionnement du système d’acquisition, dans sa dernière version connue, sera développé.

3.2 Les chambres à plaques résistives

La figure 3.7 montre une vue en coupe d’une chambre à plaques résistives en verre utilisée par le SDHCAL. Les chambres à plaques résistives de verre sont des détecteurs gazeux composés d’une cathode et d’une anode en verre, de résistivité $\rho = 10^{12} \Omega.cm$ et de respectivement 0.7 et 1.1 mm d’épaisseur. Elles sont séparées

par une fine couche de gaz de 1.2 mm d'épaisseur dont l'uniformité est garantie par des billes en céramique collées entre les deux électrodes. Le mélange de gaz utilisé est le suivant : 93% de $C_2H_2F_4$ (*TFE*), 5% de CO_2 et 2% de SF_6 . Le *TFE* est la gaz qui fournit les électrons de l'avalanche. Le CO_2 absorbe les photons issus des désexcitations des molécules et limite ainsi le déclenchement des avalanches secondaires intempestives (*quencher*). Le SF_6 permet quant à lui d'absorber les électrons. En faible quantité dans le mélange il permettra de réduire la largeur des avalanches électroniques et de réduire le bruit thermique.

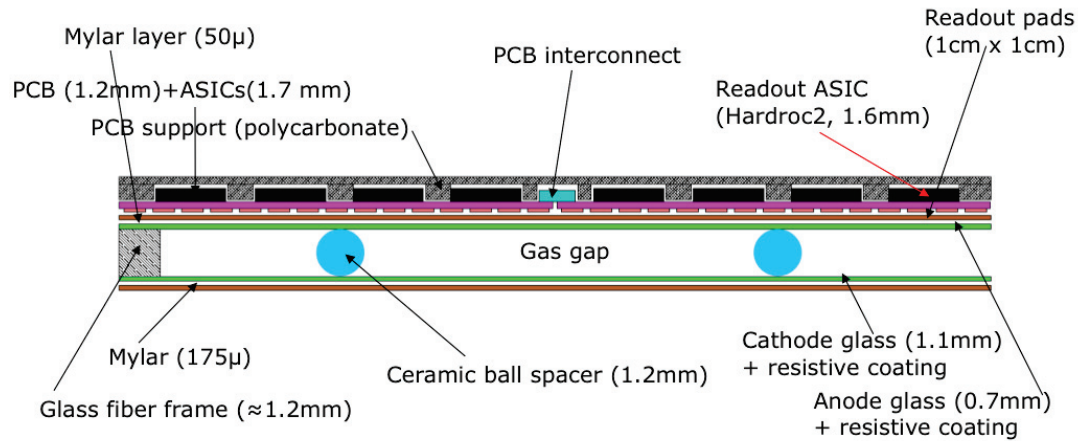


FIGURE 3.7 – Vue en coupe d'une chambre à plaques résistives en verre. Les dimensions sont celles utilisées par le prototype SDHCAL [8].

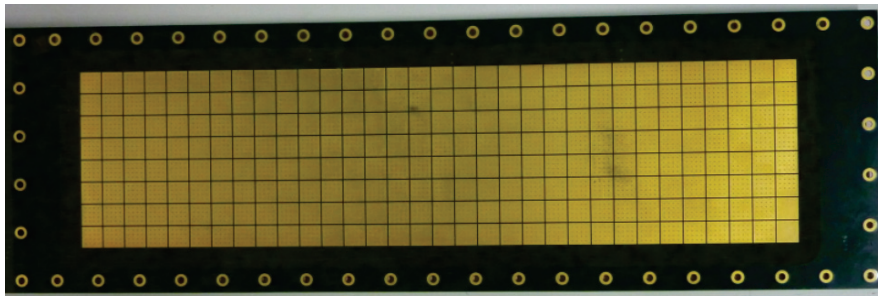


FIGURE 3.8 – Photographie des carreaux en cuivre utilisés lors du développement de petites GRPCs. Ces carreaux sont imprimés sur l'autre face du circuit de lecture [8].

La peinture résistive appliquée sur la surface des électrodes permet d'y appliquer une haute tension. La haute tension appliquée est déterminée en fonction de l'efficacité du détecteur. Les figures 3.9(a) et 3.9(b) montrent l'efficacité de deux chambres GRPCs en fonction de la haute tension appliquée. La haute tension nominale doit se situer sur le plateau, ici, à partir de 6.8 kV. La haute tension appliquée a été fixée à 6.9 kV pour les différentes prises de données au SPS.

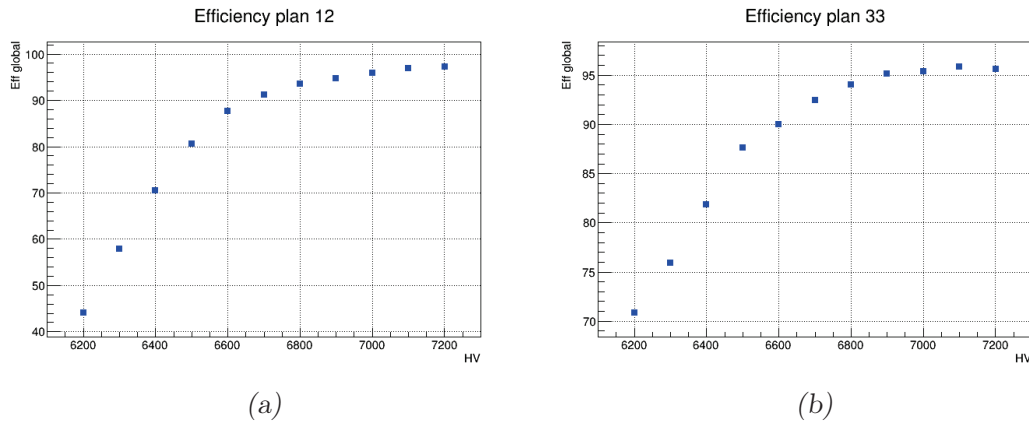


FIGURE 3.9 – Efficacité des chambres 12 (a) et 33 (b) en fonction de la haute tension appliquée. Le plateau d’efficacité est atteint autour de 6.8 kV [8].

Directement imprimés sur une face du circuit de lecture, des carreaux en cuivre d’une taille de 1 cm^2 permettent de collecter la charge issue des avalanches électroniques dans le gaz (voir figure 3.8). Leur nombre dans chaque GRPC s’élève à 96×96 cellules, soit 9216 cellules, pour un total de plus de 440000 cellules dans les 48 couches du calorimètre. Pour isoler ces dernières, une fine feuille de Mylar de $50 \mu\text{m}$ d’épaisseur est placée entre l’anode et les carreaux en cuivre.

3.3 L’électronique de lecture

Les carreaux en cuivre des GRPCs sont imprimés sur une des faces du circuit de lecture. De l’autre côté, le circuit est composé de 6 *ASUs* (*Active Unit Sensors*) soudés entre eux. Sur chacune de ces unités sont placés 24 *ASICs* (*Application Specific Integrated Circuit*) HARDDROC2. Chacun de ces ASICs collecte les signaux issues de 64 carreaux en cuivre. Ces signaux sont directement mis en forme et transmis à trois discriminateurs pour déterminer le seuil du signal de chacune des voies. Les trois seuils des différents discriminateurs sont fixés par logiciel. Dans le cas du prototype du SDHCAL, ces seuils sont fixés respectivement à 114 *fC*, 5 *pC* et 15 *pC* pour le premier, deuxième et troisième seuils.

Le tout est piloté par 3 DIFs (*Detector Interfaces Board*) qui sont connectées au 144 ASICs de la chambre. Ces cartes assurent le lien entre les signaux collectés par les ASICs et le système d’acquisition. Ce dernier permet de paramétrer l’ensemble de l’électronique et de les piloter lors des acquisitions.

Dans ce prototype, le nombre de voies est déjà assez important, avec plus de 440000 canaux de lecture. Le calorimètre de l’ILD comportera plus de 70 millions de cellules de 1 cm^2 . Comme mentionné dans la section 2.2 du chapitre 2, ce grand nombre de canaux entraîne une consommation électrique élevée et donc une énergie dissipée importante. Un système de refroidissement deviendrait alors indispensable. Cependant, ces systèmes ont une taille assez imposante et un budget matière non-négligeable. Cette solution n’est pas envisageable pour l’application des algorithmes

de suivi de particules, puisque ces derniers requièrent que les zones non instrumentées aient un budget matière faible pour optimiser la résolution en énergie des jets. Dans le chapitre 2 dédié au projet de l'ILC, nous avons vu que la structure du faisceau était de 0.95 *ms* de période active sur les 200 *ms*. Cette structure permet d'éteindre les composants électroniques les plus consommateurs en énergie grâce à une alimentation pulsée durant la période sans faisceau. Ceci aura pour effet de réduire la dissipation thermique dans les différents sous-détecteurs et de réduire le bruit. Un système de refroidissement plus simple, par exemple par eau, avec un budget matière faible, pourra être mis en place pour assurer la stabilité des détecteurs durant la prise de données.

Le prototype du SDHCAL intègre une alimentation pulsée au sein de son électronique. Elle permet d'éteindre les composants analogiques des ASICs pendant les périodes sans prise de données. Le SDHCAL a été le premier détecteur à tester cette alimentation pulsée au sein d'une chambre GRPC placée dans un champs magnétique [13]. Ce système a aussi été utilisé lors des différentes prises de données du prototype au SPS. Un simple système de refroidissement a cependant été utilisé pour assurer le bon fonctionnement du détecteur tout au long des prises de données. Celui-ci est composé de plaques de cuivres fixées sur les deux cotés du prototype dans lesquelles circulent de l'eau refroidie à 10°C.

3.4 Le système d'acquisition

Les différentes cartes présentées dans la section précédente sont synchronisées à l'aide de cartes externes nommées *concentrateurs*. Les temps dans les ASICs et dans les DIFs sont échantillonnés par pas de 200 *ns* et synchronisés à l'aide de DCC (*Data Concentrator Card*). Les DCCs sont elles-mêmes synchronisées par une SDCC (*Synchronous Data Concentrator Card*). Au démarrage d'une acquisition, les horloges internes des ASICs sont remises à zéro. Au cours d'une acquisition, si un des canaux des ASICs dépasse le premier seuil, un événement est stocké dans la mémoire interne de l'ASIC à laquelle il est relié. Lorsque la limite de la mémoire d'un seul canal est atteinte (127 événements), l'acquisition est arrêtée et la mémoire de la totalité des ASICs du détecteur est lue par les DIFs et ensuite transmise aux ordinateurs chargés de collecter les données. On appelle un événement RAMfull la totalité des événements lus par les ASICs depuis le début d'une phase d'acquisition. Une fois la totalité des données transmises, un nouveau cycle d'acquisition démarre. Chacun des événements contient les informations suivantes :

- le numéro de l'événement RAMfull
- le temps absolu (par pas de 200 *ns*) au moment du signal RAMfull depuis le début de la prise de données
- Pour chaque hit :
 - le numéro de la DIF (8 bits),
 - le numéro de l'ASIC (6 bits),
 - le numéro du canal (6 bits),

- le temps depuis le début du cycle d’acquisition, par pas de 200 *ns* (64 bits),
- le seuil déclenché (2 bits).

Ce mode de fonctionnement a l’avantage de ne pas utiliser de déclenchement externe. Une fois une acquisition lancée, les cycles se succèdent jusqu’à la fin de l’acquisition. De ce fait, les événements RAMfull contiendront aussi des muons cosmiques et toutes sortes de particules issues du faisceau. Le bruit du détecteur sera aussi enregistré dans les événements. Cependant, avec ce mode de fonctionnement, si un ou plusieurs canaux électroniques sont bruyants, les cycles d’acquisitions deviennent alors très courts et empêchent le bon déroulement de la prise de données.

Ce mode de fonctionnement rend obligatoire une procédure de reconstruction des événements physiques qui sera détaillée dans la section 4.1.

3.5 Simulation numérique du prototype

La simulation numérique du prototype physique du SDHCAL est une étape importante pour la validation des données enregistrées lors des tests sur faisceau. La compatibilité entre les données et la simulation est aussi la première étape avant une simulation complète des détecteurs dans l’ILD.

La première partie de cette simulation, réalisée avec le logiciel **GEANT4**, simule le passage des particules à travers les matériaux du détecteur. La deuxième partie est la digitalisation, développée dans un programme séparé. Cette dernière permet d’obtenir une structure de données comparable à celle enregistrées lors des tests sur faisceau à partir des données issues de la simulation.

Le logiciel de simulation **GEANT4** décrit la géométrie des détecteurs avec une grande précision ainsi que le passage des particules à travers ces derniers. La composition chimique et les densités des matériaux sont utilisées dans les calculs de sections efficaces dans **GEANT4**. La physique est simulée à l’aide de listes de physique définie par une liste de processus et une gamme d’énergie pour laquelle ces processus sont valables. Pour simuler les phénomènes physiques, différents modèles sont combinés pour couvrir toute la gamme d’énergie désirée. Le logiciel **GEANT4** définit un certain nombre de listes physiques et leurs transitions entre les différents modèles. La figure 3.10 montre les modèles de simulation de la physique hadronique utilisés dans **GEANT4** en fonction de l’énergie des particules.

Les listes de physique **FTFP_BERT** et **QGSP_BERT** sont deux listes usuellement utilisés dans le domaine de la calorimétrie à haute énergie. Une option pour ces deux listes permet de simuler avec une meilleure précision les interactions des neutrons avec une énergie inférieure à 20 *MeV*. Elle se nommeront alors respectivement **FTFP_BERT_HP** et **QGSP_BERT_HP**. La liste **FTF_BIC** sera également utilisée pour la comparaison aux données de test sur faisceau.

Le passage des particules à travers la matière est réalisé par une méthode d’élément fini. Cela signifie que pour une particule donnée, sa trajectoire et les processus mis en jeu lors de son passage sont calculés pas à pas par le programme sur des petits segments. Le programme de simulation du prototype du SDHCAL sauvegarde les segments produits par les particules chargées dans la couche de gaz de chaque

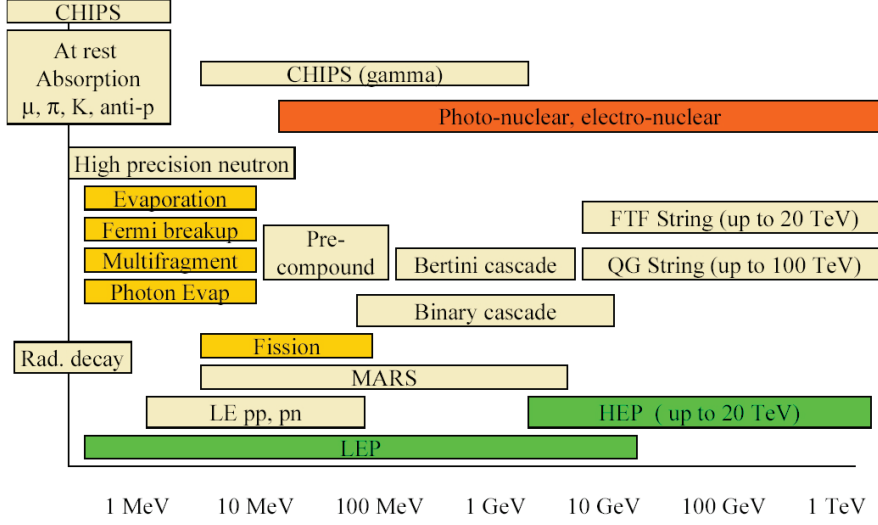


FIGURE 3.10 – Modèles de simulation de la physique hadronique utilisés dans GEANT4 en fonction de l'énergie des particules.

chambre. Leurs passages pourraient déclencher la simulation de plusieurs avalanches électroniques, phénomène qui n'est pas pris en charge par GEANT4. La simulation de ces avalanches est traitée dans un programme séparé appelé **SimDigital**, développé par la collaboration SDHCAL [20]. Ce programme permet d'estimer la répartition de la charge induite sur les carreaux de cuivre de GRPCs et de créer des hits au dessus d'un certain seuil, de la même façon que dans les ASICs du prototype.

Pour chaque segment sauvegardé en sortie de la simulation, la charge induite est aléatoirement répartie sur les cellules environnantes suivant une distribution de Polya :

$$P(q; \theta, \bar{q}) = \left[\frac{q}{\bar{q}} (1 + \theta) \right]^\theta e^{-\frac{q}{\bar{q}}(1+\theta)} \quad (3.3)$$

avec \bar{q} la valeur moyenne de la charge en pC et θ un paramètre libre lié à la largeur de la distribution. Des corrections dues à l'efficacité du détecteur en conditions réelles sont également apportées afin de se rapprocher au mieux des données enregistrées lors des différents tests sur faisceau. Une description plus détaillée de l'algorithme de digitalisation est disponible dans [20]. Ce travail de recherche étant néanmoins basé sur cette simulation, ses performances seront appréciées dans la section 4 de ce chapitre.

4 Les performances du SDHCAL

Le prototype physique du SDHCAL ainsi que sa simulation numérique ont été décrits dans les sections précédentes. Le prototype a été soumis à différents types de faisceau de particules sur les lignes H2 et H6 au CERN afin de valider son fonctionnement. Nous allons voir comment les données récoltées permettent de

reconstruire l'énergie des particules et de décrire la topologie des gerbes hadroniques suivant plusieurs méthodes.

4.1 Reconstruction des événements physiques

Au vu de la façon dont sont enregistrées les données par l'électronique du prototype, un découpage des événements RAMfull est nécessaire afin d'analyser les données particule par particule. Différents types de données sont enregistrées dans ces événements RAMfull : des muons cosmiques, des muons du faisceau, des gerbes hadroniques, des gerbes électromagnétiques et du bruit. Les événements physiques dus aux particules sont séparés du bruit en utilisant un algorithme de groupement temporel des hits appelé *Trivent* [8].

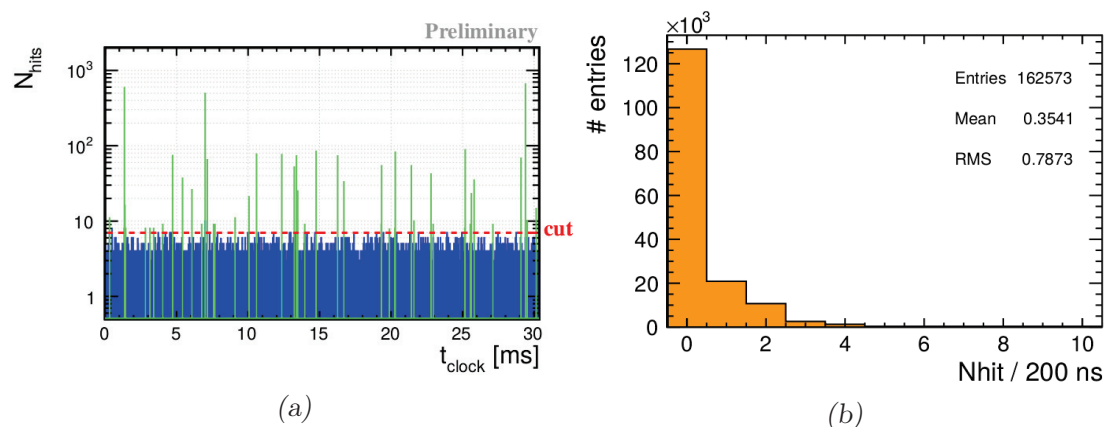


FIGURE 3.11 – (a) Distribution du temps des hits dans un événement RAMfull enregistré par le prototype du SDHCAL. Les bins verts correspondent aux événements physiques candidats. (b) Distribution du nombre de hits correspondant à du bruit dans un bin en temps (1 bin = 200 ns). Ces hits sont ceux qui ne sont pas sélectionnés par l'algorithme de groupement en temps *Trivent* (en bleu sur la figure (a)) [8].

La distribution de l'horodatage des hits est échantillonnée par pas de 200 ns grâce à l'horloge interne des ASICs. La figure 3.11(a) montre un exemple de spectre en temps d'un événement RAMfull enregistré par le prototype SDHCAL. Les potentiels événements physiques sont ceux dont le nombre de hits dans un bin de temps ($\Delta t = 200$ ns) dépasse une certaine valeur ($N_{hit} > 7$ sur la figure 3.11(a)). Parmi ces bins candidats, les maxima locaux sont identifiés ($t = t_0$) et des événements physiques sont créés. Enfin, les hits des bins adjacents ($t = t_0 \pm 200$ ns) sont également ajoutés à l'événement. Il a cependant été observé que pour un faible nombre d'événements, certaines DIFs pouvaient se désynchroniser des autres d'un coup d'horloge. Les gerbes apparaissent alors "trouées", comme sur la figure 3.12. La fenêtre en temps additionnelle est finalement agrandie d'un bin de temps supplémentaire afin de récupérer ces hits manquants ($t = t_0 \pm 2 \times 200$ ns). Une fois l'événement total reconstruit, les hits sont convertis dans un format plus approprié pour l'analyse des

données. La position des hits en mm est reconstruite à partir des identifiants de DIF, ASIC et canal de chacun d'eux.



FIGURE 3.12 – Événement d'une gerbe hadronique enregistré dans le prototype du SDHCAL comportant des "trous" dans certains plans. Cet effet est corrigé lors de la reconstruction des événements physiques en agrandissant la fenêtre de temps d'un bin en temps supplémentaire.

Les hits qui ne sont pas associés à un événement physique sont considérés comme du bruit. La figure 3.11(b) montre la distribution du nombre des hits de bruit par bin en temps. Sa valeur moyenne de $0.35 \text{ hits} / 200 \text{ ns}$ permet d'estimer un bruit moyen de 1.75 hits par événement physique reconstruit.

A noter que l'algorithme `Trivent` a été généralisé pour reconstruire des événements physiques issus de plusieurs détecteurs, si la structure de l'horodatage des données respectives le permet. Cette généralisation sera notamment utilisée par le système de surveillance de la combinaison des détecteurs SiWECal et SDHCAL lors d'un test sur faisceau présenté dans la section 3 du chapitre 7.

4.2 Les performances du détecteur

Les deux quantités importantes qui caractérisent les GRPCs du SDHCAL sont l'efficacité et la multiplicité. La multiplicité correspond au nombre de canaux allumés lorsqu'une MIP (Minimum Ionising Particle) a été signalée. Pour calculer ces deux variables, les muons sont utilisés. Les autres quantités qui caractérisent le SDHCAL dans son ensemble sont reliées à l'énergie. Le nombre de hits est sans doute une des quantités les plus importantes, puisque l'estimateur d'énergie en dépendra directement.

4.2.1 Sélections de muons

Afin d'étudier l'efficacité et la multiplicité de chaque GRPC du SDHCAL, on utilise des événements contenant des muons. Ces derniers doivent d'abord être sélectionnés parmi les événements physiques reconstruits. Durant la prise de données, aucun détecteur Cherenkov n'a été utilisé et en conséquence, les particules ne peuvent

être directement identifiées. Pour cela, nous nous aiderons de variables topologiques spécifiques à la signature de chaque type de particule.

La première étape de la sélection consiste en une analyse en composante principale (PCA) et permet d'extraire plusieurs variables intéressantes. La PCA consiste en la diagonalisation de la matrice de covariance. Dans notre cas, les éléments de celle-ci sont déterminés par l'équation suivante :

$$M_{ij} = \sum_k^N (q_{ik} - \bar{q}_i)(q_{jk} - \bar{q}_j) \quad (3.4)$$

avec q_{ik} les coordonnées des positions des hits, \bar{q}_i les coordonnées barycentriques des hits de l'événement et N le nombre total de hits. Cette matrice est donc une matrice 3×3 . Après la diagonalisation de cette matrice, les valeurs propres λ_i donnent les directions principales de la particule dans le calorimètre. La valeur propre λ_1 donne la direction principale alors que les valeurs propres λ_2 et λ_3 correspondent aux directions principales transverses. On définit le rapport transverse de la façon suivante :

$$\lambda_{tr} = \frac{\sqrt{\lambda_2^2 + \lambda_3^2}}{\lambda_1} \quad (3.5)$$

Ce rapport caractérise l'extension de la particule dans l'espace. Un faible rapport transverse sera caractéristique d'une trace dans le détecteur. On peut en plus corrélérer cette variable au nombre total de hits dans le détecteur. La figure 3.13 montre le rapport transverse en fonction du nombre total de hits N_{hit} dans le détecteur pour un échantillon de données de test sur faisceau à 30 GeV. La première région avec $N_{hit} < 200$ correspond donc aux muons de faisceau et aux muons cosmiques alors que la deuxième région avec $N_{hit} > 200$ correspond aux gerbes hadroniques. La suppression des événements avec $\lambda_{tr} > 0.05$ permet en plus de conserver une partie importante des muons et de rejeter la plupart des gerbes hadroniques.

La deuxième sélection appliquée pour isoler les traces dans le calorimètre est liée à l'ajustement linéaire de la trace. Des amas sont d'abord créés à partir des hits voisins dans chaque plan. Les événements avec un nombre d'amas faible sont rejetés ($N_{amas} < 5$). La trajectoire de la particule est ajustée avec une régression linéaire. La qualité de l'ajustement est estimée avec un χ^2 :

$$\chi^2 = \frac{1}{N_{amas}} \sum_i^{N_{amas}} \left(\frac{d_i}{\sigma_i} \right)^2 \quad (3.6)$$

avec d_i la distance entre les barycentres des amas des hits et la droite définie par les coefficients obtenus par la régression linéaire et σ_i l'erreur sur la position de l'amas i . Les événements avec $\chi^2 > 100$ sont rejetés. La simulation permet de confirmer la qualité de l'ensemble de ces coupures avec une efficacité proche de 96%. La figure 3.14 montre la distribution du nombre total de hits dans le calorimètre après cette sélection pour un échantillon de données de test sur faisceau à 30 GeV. Les deux pics correspondent aux muons cosmiques (plus faible nombre de hits) et aux muons de faisceau (plus grand nombre de hits). A noter que ces mêmes sélections

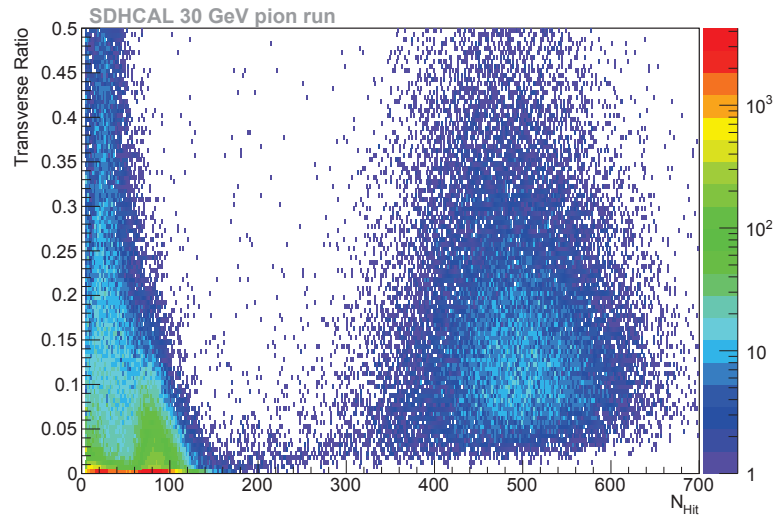


FIGURE 3.13 – Rapport transverse λ_{tr} en fonction du nombre de hits pour un échantillon de données à 30 GeV.

sont appliquées sur les événements simulés afin de ne pas introduire de biais sur les différentes mesures résultantes.

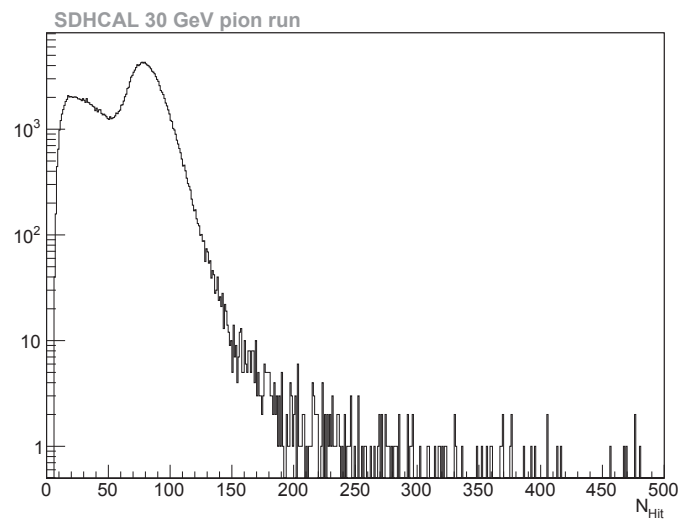


FIGURE 3.14 – Distribution du nombre total de hits dans le calorimètre après sélection des muons pour une prise de données à 30 GeV.

4.2.2 Efficacité et multiplicité

À partir des traces sélectionnées par la méthode précédente, l'efficacité et la multiplicité dans chaque plan sont évalués.

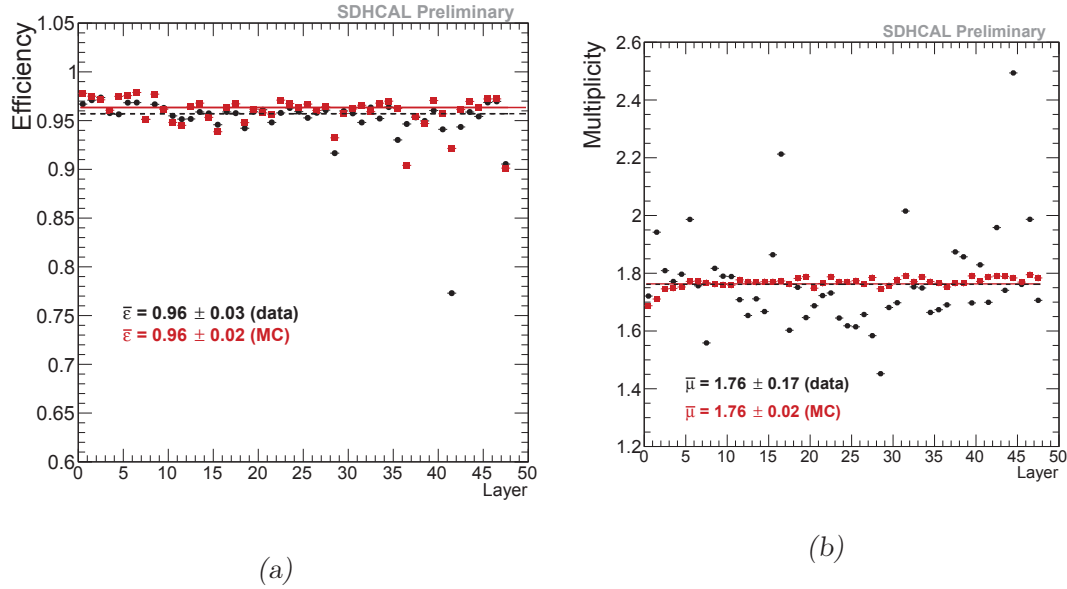


FIGURE 3.15 – (a) Efficacité et (b) multiplicité du détecteur, plan par plan [20].

Pour l'efficacité, on détermine, à partir de la droite définie par les coefficients obtenues par la régression linéaire, le point d'impact dans chaque plan. Un plan est considéré comme efficace si un amas est trouvé à une distance inférieure à 5 cm du point d'impact. La figure 3.15(a) montre l'efficacité moyenne du détecteur, plan par plan. L'efficacité moyenne est estimée à $96 \pm 0.03\%$. La faible efficacité du plan 42 ($\epsilon_{42} \simeq 77\%$) s'explique par le fait qu'une carte électronique (sur 3) de la chambre n'était pas en marche au moment de la prise de données.

La multiplicité est quant à elle évaluée par la taille de l'amas trouvé lors du calcul de l'efficacité. La figure 3.15(b) montre la multiplicité moyenne du détecteur, plan par plan. La multiplicité moyenne sur l'ensemble du détecteur est estimée à 1.76 ± 0.17 . La dispersion plus importante de la multiplicité dans les données de test sur faisceau peut s'expliquer par des différences d'uniformité dans chaque chambre du détecteur, en raison de la résistivité de la peinture appliquée sur les électrodes.

Pour les deux quantités, les incertitudes correspondent aux écarts types dans chaque plan.

4.3 Reconstruction de l'énergie des hadrons

Afin d'évaluer l'énergie des gerbes hadroniques, une sélection est d'abord appliquée. Elle permet de supprimer les muons et les gerbes électromagnétiques. Une fois cette sélection effectuée, différents estimateurs d'énergie seront appliqués et comparés sur ces événements.

4.3.1 Sélection des gerbes hadroniques

Contamination par les protons

La ligne H6 du SPS est contaminée par des protons. Les gerbes hadroniques qu'ils produisent sont peu différentes des gerbes hadroniques initiées par les pions. La différence se fait en réalité au niveau de la longueur d'interaction des protons qui est légèrement plus faible. En moyenne, les protons interagissent de manière forte plus rapidement que les pions. Les gerbes hadroniques issues des protons démarrant ainsi plus tôt et le nombre total de hits se verra donc augmenté car les gerbes produites par les protons seront plus contenues.

La collaboration ATLAS a mesuré la contamination des protons sur la ligne H6 [2]. La fraction du nombre de protons dans le faisceau de pions est estimée à 0.45 ± 0.12 à 50 GeV et 0.61 ± 0.06 à 100 GeV . Les événements contenant des protons ne seront pas traités différemment des pions par la suite. Cependant, cette contamination ne sera pas présente sur la ligne H2 puisque le faisceau est chargé négativement.

Suppression des muons

Les sélections suivantes permettent de supprimer une part importante des muons de faisceau et des cosmiques :

- une coupure sur le rapport $\frac{N_{hit}}{N_{plans}} > 3$, avec N_{plans} le nombre de plans contenant au moins 1 hit,
- une coupure sur le rapport transverse $\lambda_{tr} > 0.01$, avec λ_{tr} défini par l'équation 3.5,
- une coupure sur le rapport $\frac{N_{IP}}{N_{plans}} > 0.1$, avec N_{IP} le nombre de plans pour lesquels l'écart-type de la position des hits est supérieur à 5 cm . Cette dernière coupure permet de supprimer les muons radiatifs pour lesquels une partie de la trace est plus large dans les plans où la radiation est émise.

Le résultat de cette sélection sur une prise de données de pions de 30 GeV est montré sur la figure 3.16.

Suppression des gerbes électromagnétiques

Afin de supprimer les électrons des événements physiques reconstruits, au moins une des trois coupures suivantes doit être vérifiée :

- une coupure sur le nombre de traces reconstruites $N_{trk} > 0$ en utilisant la technique de la transformée de Hough [17],
- une coupure sur le plan de démarrage de la gerbe $P_{start} \geq 5$,
- une coupure sur le nombre de plans contenant au moins 1 hit $N_{plan} > 30$, les gerbes électromagnétiques étant très compactes et s'arrêtant avant pour la gamme d'énergie considérée.

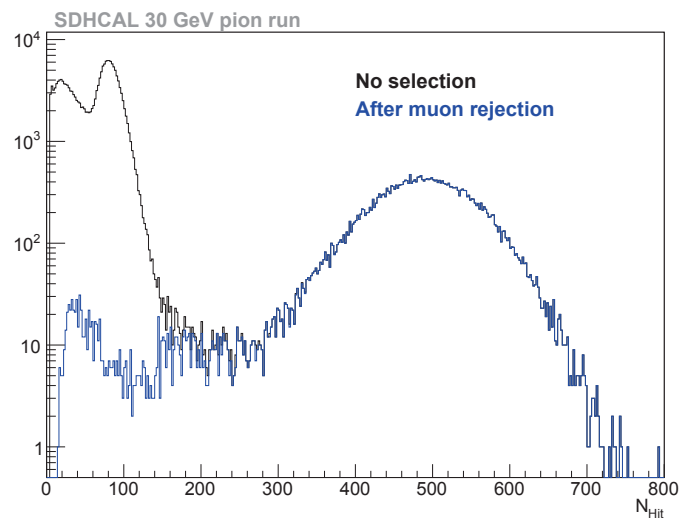


FIGURE 3.16 – Distribution du nombre total de hits dans le calorimètre après la suppression des muons pour une prise de données de pions de 30 GeV [16].

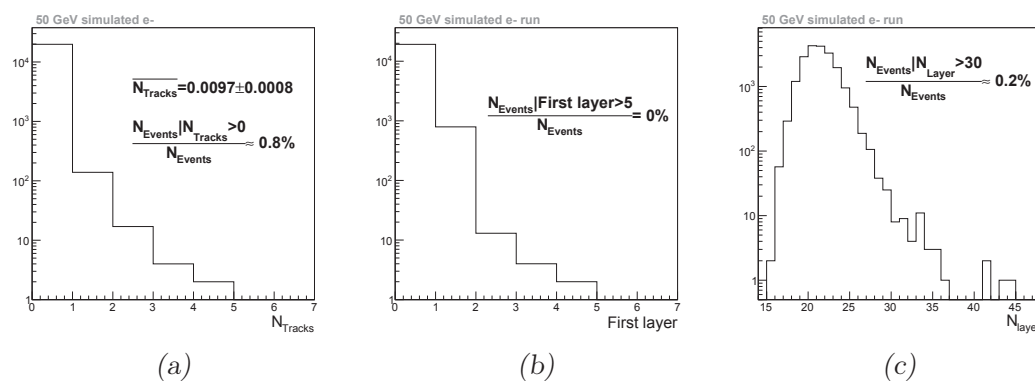


FIGURE 3.17 – (a) Distribution du nombre de traces reconstruites (b) Distribution du premier plan d'interaction (c) Distribution du nombre de plans touchés. L'échantillon de données est issu d'une simulation d'électrons de 50 GeV [43].

La figure 3.17 montre la distribution de ces trois variables pour un échantillon de simulation d'électrons de 50 GeV. Chacune des trois figures montre également la proportion d'événements non rejetés par les coupures définies précédemment. Ces coupures sont très efficaces et permettent de supprimer une très grande partie des gerbes électromagnétiques. Cependant, afin de limiter cette contamination, une fine couche de plomb de 8 mm d'épaisseur était insérée sur la ligne de faisceau en amont pour arrêter une part importante des électrons.

Coupages additionnelles

Malgré l'application des coupures précédentes, les échantillons de données sont toujours contaminés dans la région de faible nombre de hits. Le reste des événements est issu d'interactions en amont du calorimètre donnant naissance à plusieurs topologies d'événement. Les trois coupures suivantes traitent trois topologies d'événement différentes :

- les particules neutres peuvent s'identifier en vérifiant que les 5 premiers plans du calorimètre contiennent moins de 4 hits. A noter que cette coupure supprime aussi beaucoup de muons cosmiques,
- les particules avec un angle important par rapport à l'axe du faisceau. Cet angle θ_{shower} est déterminé en ajustant une droite avec tous les hits de l'événement. La coupure appliquée est $\theta_{shower} > 25^\circ$. Cet angle est dû à de possibles interactions en amont du calorimètre qui pourraient dévier la particule ou aux muons cosmiques.
- les événements avec plusieurs particules. Ces particules sont créées par des interactions de la particule incidente en amont du calorimètre et sont dispersées dans les premiers plans. La dispersion des hits dans les 5 premiers plans est calculée. Si 4 de ces plans contiennent des amas de hits séparés de plus de 5 cm, l'événement est étiqueté comme contenant plusieurs particules et supprimé de l'échantillon.

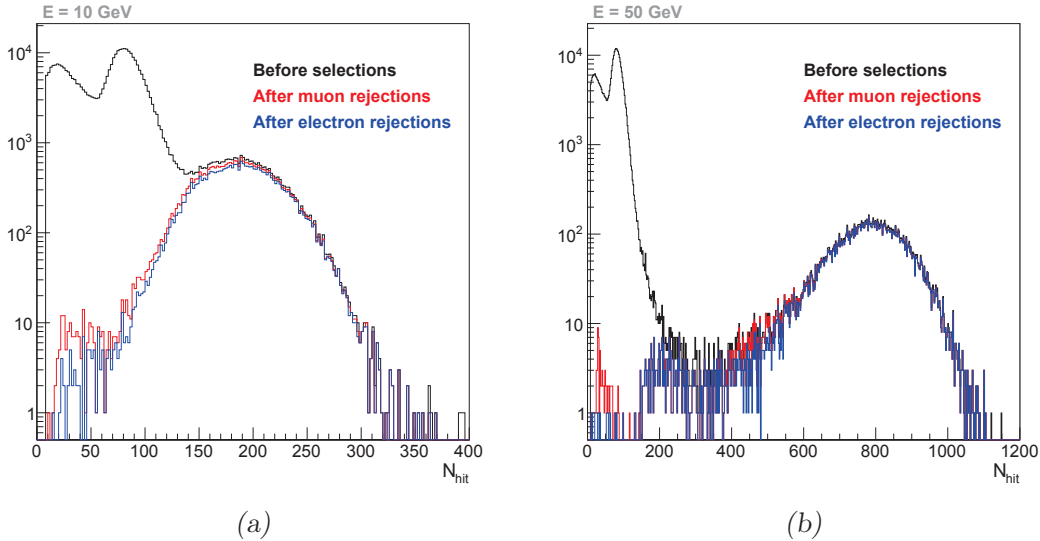


FIGURE 3.18 – Distribution du nombre de hits pour des échantillons de données de test sur faisceau de pions de (a) 10 GeV et (b) 50 GeV avant toutes coupures (en noir), après la suppression des muons (en rouge) et après suppression des gerbes électromagnétiques (en bleu) [43].

Énergie	Efficacité
5 GeV	57.9 ± 0.3%
10 GeV	85.8 ± 0.2%
15 GeV	90.5 ± 0.2%
20 GeV	92.3 ± 0.2%
25 GeV	93.9 ± 0.2%
30 GeV	94.3 ± 0.2%
40 GeV	95.1 ± 0.3%
50 GeV	94.9 ± 0.2%
60 GeV	95.1 ± 0.2%
70 GeV	94.5 ± 0.2%
80 GeV	94.3 ± 0.2%

TABLE 3.5 – Efficacité de sélection des gerbes hadroniques en fonction de l'énergie de la particule incidente [43].

Toutes ces coupures permettent de filtrer au mieux les événements et de ne garder que les gerbes hadroniques avec un minimum de biais. La figure 3.18 montre la distribution du nombre de hits avant toutes coupures (en noir), après suppression des muons (en rouge) et après suppression des électrons (en bleu) pour deux échantillons de pions à 10 GeV et 50 GeV.

La table 3.5 détaille l'efficacité de sélection des gerbes hadroniques évaluée avec la simulation en fonction de l'énergie de la particule incidente. L'efficacité est satisfaisante sur toute la gamme en énergie, excepté à 5 GeV. Pour cette énergie, les gerbes hadroniques sont rejetées à 21% par les coupures sur les muons, à 15% par les coupures sur les électrons et 11% par les coupures sur l'angle de la gerbe.

4.3.2 Calibration en fonction du temps

La figure 3.19(a) montre le nombre moyen de hits de seuil 3 pour des pions de 80 GeV en fonction du temps depuis le début du spill. On voit que ce nombre moyen diminue avec le temps. Cette diminution est due à la haute résistivité volumique du verre utilisée dans les GRPCs du SDHCAL ($\rho = 10^{12} \Omega.cm$) qui diminue l'efficacité de détection lorsque le flux de particules est élevé [28]. Cet effet est encore plus important au cœur des gerbes hadroniques ou électromagnétiques où se trouve un nombre important de particules secondaires. On s'attend donc à ce que cette diminution s'intensifie avec l'énergie de la particule incidente étant donné que la densité de particules secondaires augmentera au sein de la gerbe.

Ce nombre de hits moyen peut se corriger en ajustant un polynôme d'ordre 2 pour chacun des seuils et en ajoutant artificiellement des hits à chacune des gerbes en fonction de leur temps d'occurrence dans le spill de la façon suivante :

$$N_{i,corr} = N_i - (p_{1,i}.t + p_{2,i}.t^2) \quad (3.7)$$

avec N_i le nombre de hits pour le seuil i de la gerbe, $p_{1,i}$, $p_{2,i}$ les coefficients

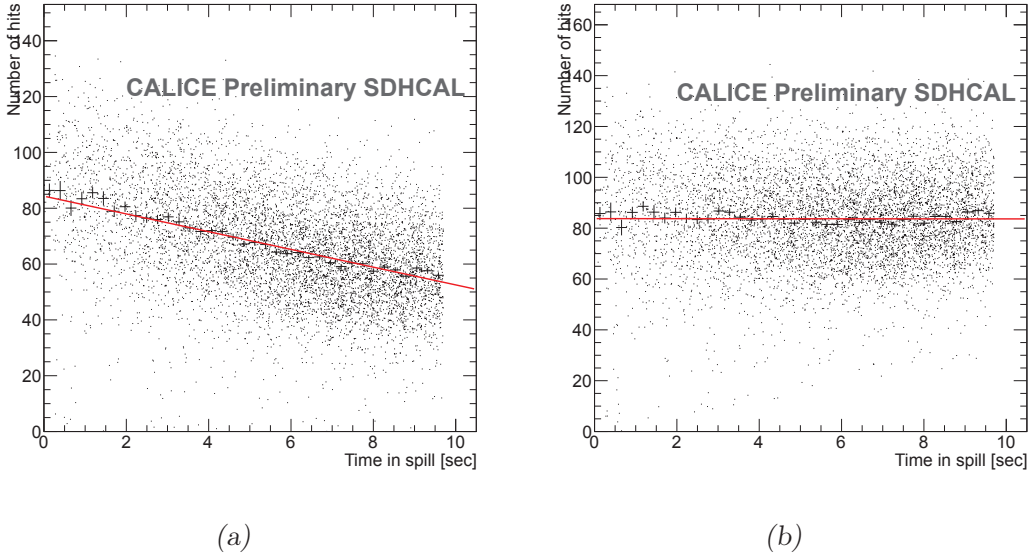


FIGURE 3.19 – Nombre moyen de hits du troisième seuil pour des pions de 80 GeV en fonction du temps depuis le début du spill (a) avant et (b) après la correction temporelle. La droite en rouge est le résultat de l’ajustement d’un polynôme d’ordre 2 : $N(t) = p_0 + p_1.t + p_2.t^2$ [16].

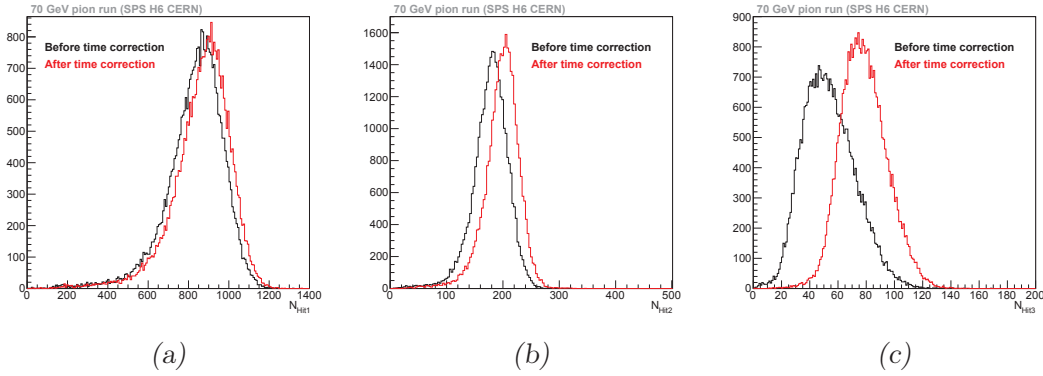


FIGURE 3.20 – Distributions du nombre de hits pour (a) le premier seuil, (b) le deuxième seuil et (c) le troisième seuil pour des pions de 70 GeV avant correction (en noir) et après correction (en rouge) en fonction du temps relatif au début du spill [43].

issus de l’ajustement du polynôme d’ordre 2 et t le temps relatif depuis le début le spill. Le nombre de hits pour les trois différents seuils pour des pions de 70 GeV avant et après cette correction est montré sur la figure 3.20. La figure 3.19(b) montre le nombre moyen de hits du troisième seuil pour des pions de 80 GeV en fonction du temps depuis le début du spill après la correction temporelle. On voit que ce nombre reste stable en fonction du temps. La table 3.6 montre le nombre moyen de

	Seuil 1		Seuil 2		Seuil 3	
	Avant correction	Après correction	Avant correction	Après correction	Avant correction	Après correction
\bar{N}	832.3 ± 0.78	865.1 ± 0.78	177.2 ± 0.17	197.7 ± 0.17	53.2 ± 0.10	78.4 ± 0.09
σ_N	142.8 ± 0.55	142.4 ± 0.55	31.8 ± 0.12	31.1 ± 0.18	18.4 ± 0.07	16.3 ± 0.06
$\frac{\sigma_N}{\bar{N}}$	0.172 ± 0.001	0.165 ± 0.001	0.179 ± 0.001	0.157 ± 0.001	0.346 ± 0.002	0.208 ± 0.001

TABLE 3.6 – Nombre moyen de hits, écart type et résolution relative avant et après correction temporelle pour chaque seuil, pour des pions de 70 GeV [43].

hits, l'écart type et la résolution relative avant et après correction temporelle pour chaque seuil, pour des pions de 70 GeV. On observe finalement une augmentation du nombre de hits ce qui a pour effet d'améliorer la résolution relative du nombre de hits. Cette amélioration permettra d'obtenir une meilleure résolution en énergie.

Il est important de noter que cette correction temporelle corrige artificiellement le nombre de hits. Aucun hit n'est réellement créé au sein de l'événement. Cette correction ne pourra pas être prise en compte lors de la reconstruction par la méthode de suivi de particules. Cependant, les résultats sur la reconstruction en énergie des gerbes hadroniques qui suivent tiennent compte de cette correction. Des résultats proches de ceux tenant compte de cette correction peuvent être obtenus en sélectionnant uniquement les événements contenus dans les premières secondes de chaque spill. Dans ce cas, une grande partie de la statistique sera perdue. Cette sélection supplémentaire sera effectuée en amont de la reconstruction.

Dans le cadre du détecteur ILD, cette correction ne s'avèrera pas nécessaire. En effet, durant la prise de données avec le prototype SDHCAL, le faisceau était focalisé au centre du détecteur afin d'enregistrer au mieux la totalité des gerbes hadroniques. Dans le détecteur ILD, la probabilité que deux particules issues de deux événements consécutifs soient enregistrées au même endroit dans le SDHCAL reste faible.

4.3.3 Estimation de l'énergie des hadrons

Les sélections décrites précédemment sont appliquées afin de sélectionner les gerbes hadroniques. La propriété principale extraite par le SDHCAL pour chacune des gerbes hadroniques est le nombre de hits.

La figure 3.21 montre le nombre de hits moyen en fonction de l'énergie du faisceau. Un ajustement linéaire est réalisé (en noir) de 5 à 20 GeV. En étendant cet ajustement aux énergies supérieures (en pointillé), on remarque que la réponse du SDHCAL sature au fur-et-à-mesure que l'énergie augmente. Ceci s'observe particulièrement bien sur la déviation à la linéarité représentée en dessous sur la même figure. Un léger sursaut s'observe à 60 GeV et peut probablement s'expliquer par la contamination par les protons, comme expliqué dans la section précédente.

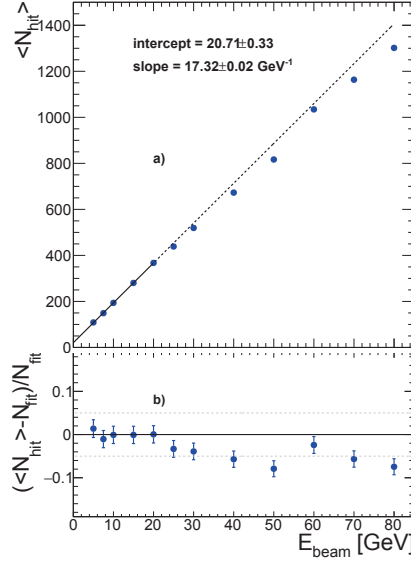


FIGURE 3.21 – (a) Nombre de hits moyen en fonction de l'énergie du faisceau sur la ligne H6 du CERN. La ligne en noire est un ajustement linéaire effectué entre 5 et 20 GeV et la courbe en pointillé, son prolongement. (b) Déviation relative du nombre moyen de hits à l'ajustement linéaire en fonction de l'énergie du faisceau [16].

Estimateurs d'énergie des hadrons

Comme expliqué précédemment, la propriété principale du SDHCAL est le nombre de hits. Cette information seule permet dans un premier temps de définir des estimateurs pour reconstruire l'énergie des hadrons (mode binaire). Le premier estimateur d'énergie que nous décrirons utilise uniquement le nombre de hits déposés dans le SDHCAL par la particule incidente :

$$E_{rec} = A.N_{hit} \quad (3.8)$$

avec A une constante à déterminer à partir des données. Le phénomène de saturation ne permet cependant pas d'obtenir une linéarité suffisante au delà de 30 GeV. Un deuxième estimateur d'énergie permet de restaurer la linéarité à haute énergie :

$$E_{rec} = A.N_{hit} + B.N_{hit}^2 + C.N_{hit}^3 \quad (3.9)$$

avec A , B et C des constantes déterminées en minimisant le χ^2 suivant sur toute la gamme en énergie disponible :

$$\chi^2 = \frac{1}{N_{evt}} \cdot \sum_{i=1}^{N_{evt}} \frac{(E_{beam,i} - E_{rec,i})^2}{E_{beam,i}} \quad (3.10)$$

avec N_{evt} le nombre total d'événements sur toute la gamme en énergie, $E_{beam,i}$ l'énergie du faisceau à chaque valeur en énergie pour la particule i et $E_{rec,i}$ l'énergie reconstruite pour la particule i . La figure 3.22 montre la distribution de l'énergie reconstruite pour deux énergies (20 et 40 GeV). La valeur moyenne de l'énergie reconstruite \bar{E}_{rec} et la résolution en énergie σ_E sont extraites par deux ajustements successifs de deux fonctions gaussiennes. Le premier ajustement est réalisé sur toute la distribution et permet d'extraire une première valeur de la moyenne $\bar{E}_{rec,1}$ et de la largeur $\sigma_{E,1}$ de la distribution. Ce premier ajustement permet de définir l'intervalle du second ajustement comme $[\bar{E}_{rec,1}-1.5.\sigma_{E,1}, \bar{E}_{rec,1}+1.5.\sigma_{E,1}]$. Ce dernier donnera les valeurs finales de l'énergie reconstruite \bar{E}_{rec} et de la résolution σ_E pour chaque valeur d'énergie du faisceau. Cette méthode permet de s'affranchir des queues de distribution présentes à basse énergie, dues aux hadrons qui interagissent tardivement dans le détecteur laissant ainsi s'échapper une part importante de l'énergie du détecteur.

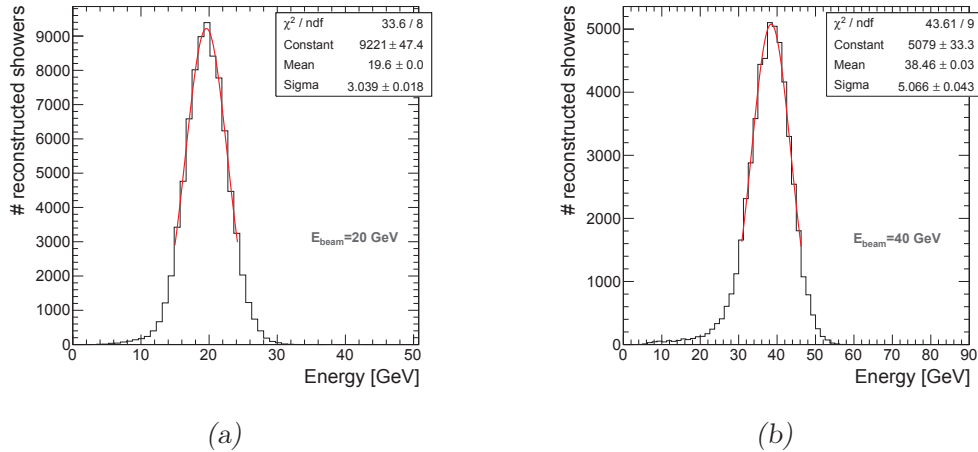


FIGURE 3.22 – Distribution de l'énergie reconstruite des gerbes hadroniques pour une énergie de faisceau de (a) 20 GeV et (b) 40 GeV, avec l'estimateur en énergie binaire de l'équation 3.9 [43].

La figure 3.23 (à gauche) montre l'énergie reconstruite moyenne et la déviation relative définie comme :

$$\frac{\Delta E}{E} = \frac{E_{rec} - E_{beam}}{E_{beam}} \quad (3.11)$$

en fonction de l'énergie du faisceau sur la ligne H6 du CERN. On voit que la linéarité est contenue dans un intervalle de $\pm 5\%$ sauf à basse énergie, à 5 GeV, où l'on observe une déviation relative de l'ordre de 11%. La figure 3.23 (à droite) présente quant à elle la résolution relative ($\frac{\sigma_E}{E_{rec}}$) en fonction de l'énergie du faisceau.

La figure 3.24 montre le nombre de hits moyen pour chacun des trois seuils du prototype du SDHCAL ainsi que la somme des trois contributions, en fonction de l'énergie du faisceau. Nous rappelons ici que la lecture semi-digitale constitue

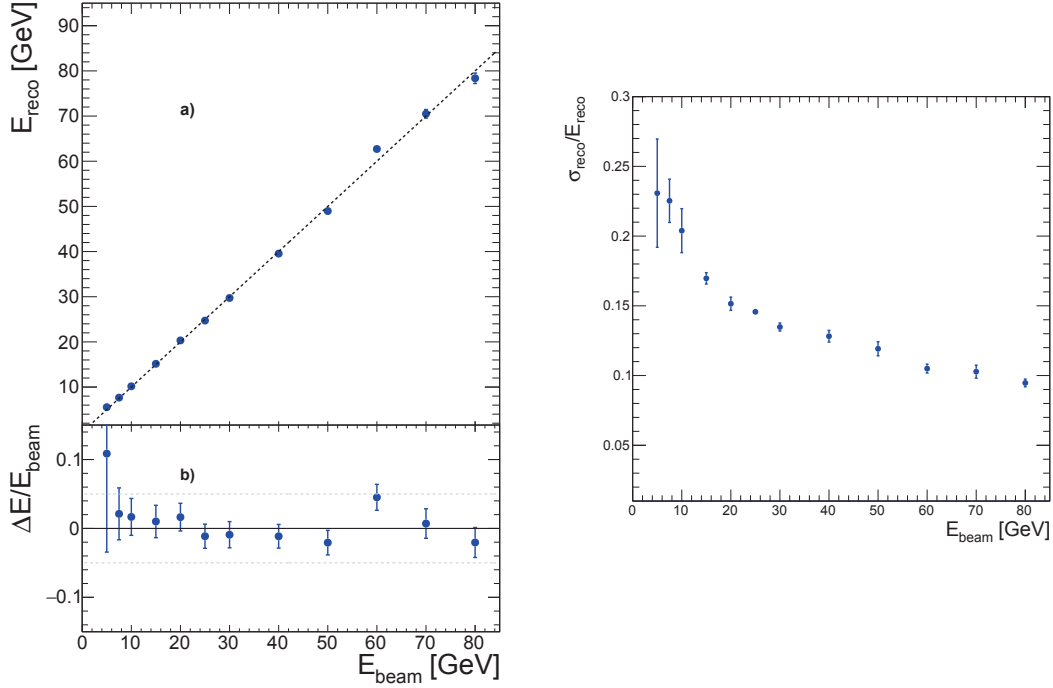


FIGURE 3.23 – Énergie reconstruite moyenne (à gauche) et résolution relative (à droite) de l'énergie reconstruite des gerbes hadroniques en fonction de l'énergie du faisceau sur la ligne H6 du CERN. L'énergie reconstruite est estimée en utilisant l'information sur le nombre total de hits uniquement (équation 3.9) [16].

une des spécificités du SDHCAL. En effet, plus la densité de particules secondaires ionisantes traversant la couche de gaz est élevée, plus le seuil déclenché sera haut. Le premier estimateur en énergie qui est développé est le suivant :

$$E_{rec} = \alpha \cdot N_1 + \beta \cdot N_2 + \gamma \cdot N_3 \quad (3.12)$$

avec N_1 , N_2 et N_3 , le nombre de hits pour chaque seuil. Cependant, le fait d'avoir ces trois coefficients constants ne permet pas d'obtenir une bonne description de l'énergie reconstruite sur une large gamme en énergie. Ces trois coefficients sont finalement paramétrés en fonction du nombre total de hits. Une fonction quadratique du nombre total de hits s'est avérée donner les meilleurs résultats [15]. Cet estimateur s'exprime finalement sous la forme :

$$E_{rec} = \alpha(N_{hit}) \cdot N_1 + \beta(N_{hit}) \cdot N_2 + \gamma(N_{hit}) \cdot N_3 \quad (3.13)$$

$$= (\alpha_1 + \alpha_2 \cdot N_{hit} + \alpha_3 \cdot N_{hit}^2) \cdot N_1 \quad (3.14)$$

$$+ (\beta_1 + \beta_2 \cdot N_{hit} + \beta_3 \cdot N_{hit}^2) \cdot N_2 \quad (3.15)$$

$$+ (\gamma_1 + \gamma_2 \cdot N_{hit} + \gamma_3 \cdot N_{hit}^2) \cdot N_3 \quad (3.16)$$

avec α_1 , α_2 , α_3 , β_1 , β_2 , β_3 , γ_1 , γ_2 et γ_3 neuf constantes à déterminer en minimisant le χ^2 donné par l'équation 3.10.

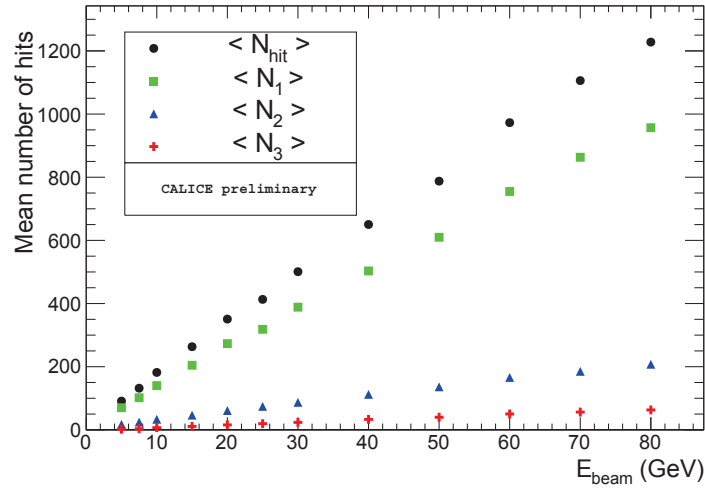


FIGURE 3.24 – Nombre de hits moyen pour chacun des trois seuils du prototype du SDHCAL en fonction de l'énergie du faisceau. Les points en noir montrent la somme des trois contributions, soit le nombre de hits total moyen [16].

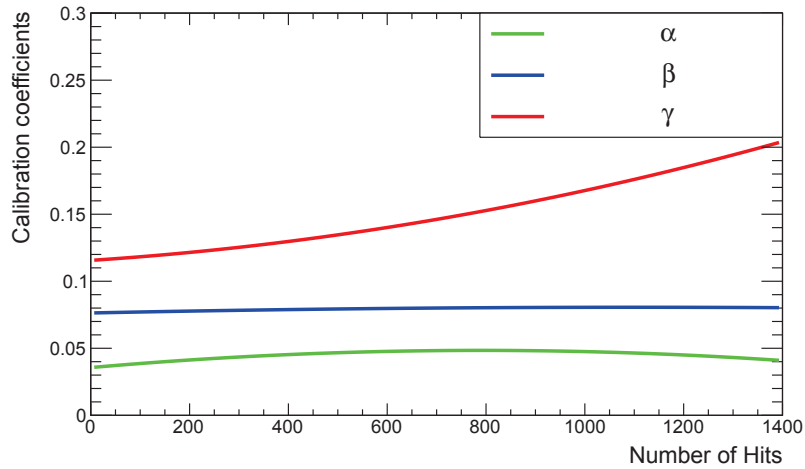


FIGURE 3.25 – Évolution des trois paramètres α , β et γ de l'équation 3.13 en fonction du nombre total de hits après minimisation [15].

Les trois fonctions α , β et γ sont tracées en fonction du nombre de hits après minimisation du χ^2 sur la figure 3.25. L'évolution du coefficient γ , correspondant au troisième seuil, montre l'importance de traiter les seuils lors de l'estimation de l'énergie des gerbes hadronique. Ceci est particulièrement vrai à haute énergie où le phénomène de saturation est accentué comme le montre l'énergie reconstruite sur la figure 3.26 (à gauche). Ces valeurs moyennes sont elles aussi extraites à l'aide deux ajustements successifs de fonction gaussienne. La déviation relative est en dessous

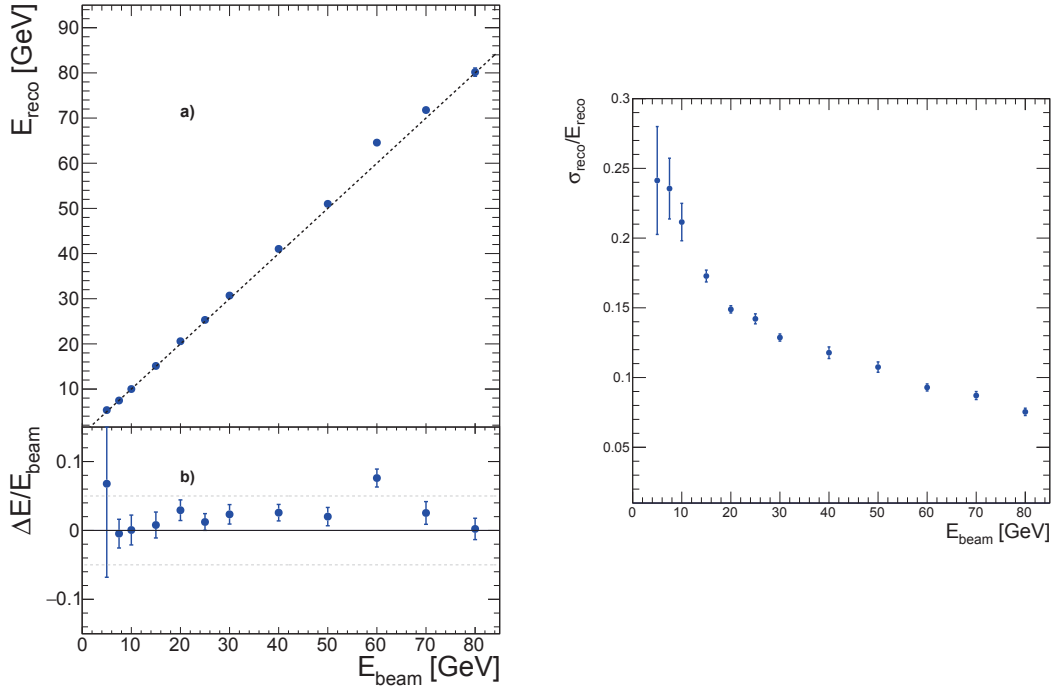


FIGURE 3.26 – Énergie reconstruite moyenne (à gauche) et résolution relative (à droite) de l'énergie reconstruite des gerbes hadroniques en fonction de l'énergie du faisceau sur la ligne H6 du CERN. L'énergie reconstruite est estimée en utilisant l'information sur les trois seuils (équation 3.13) [16].

de 5% sauf pour deux points en énergie. Le point en énergie à 5 GeV montre une grande incertitude alors que pour le point en énergie à 60 GeV , l'augmentation en énergie peut s'expliquer par la contamination des protons qui augmente le nombre de hits du SDHCAL.

La figure 3.26 (à droite) montre la résolution relative en fonction de l'énergie du faisceau. On voit que cette résolution descend jusqu'à 7.5% à 80 GeV ce qui constitue un résultat prometteur. La figure 3.27 montre un comparatif de la résolution relative en énergie entre le mode binaire (équation 3.9) et le mode multi-seuils (équation 3.13). On voit ici clairement l'intérêt d'utiliser l'information des trois seuils pour calculer l'énergie des gerbes hadroniques.

D'autres méthodes de compensation logicielles ont également été développées afin d'améliorer la résolution en énergie. On citera notamment la Transformée de Hough qui permet d'identifier les segments de trace dans les gerbes hadroniques et d'assigner un poids différent aux hits appartenant à ces traces [17] mais aussi les techniques d'apprentissage telles que le réseau de neurones [34].

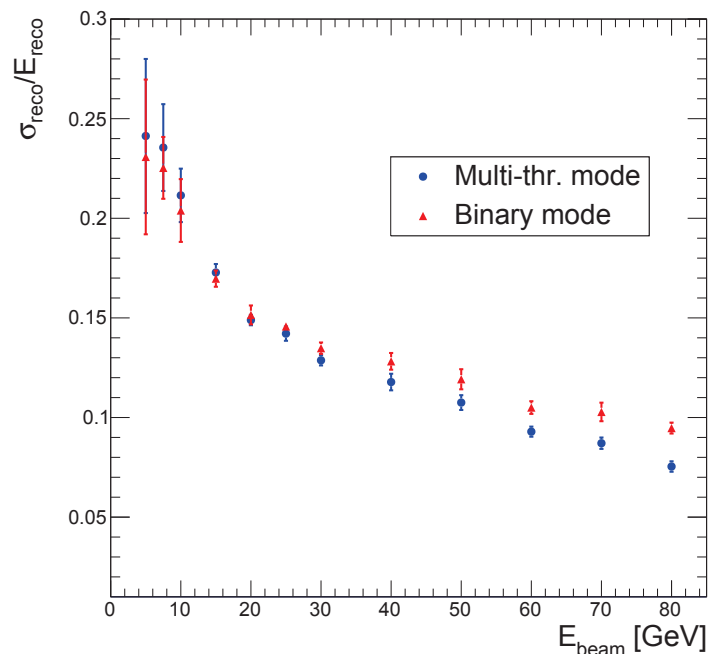


FIGURE 3.27 – Résolution relative en énergie pour les modes binaires et multi-seuils du SDHCAL [16].

4.4 Des désaccords entre données et simulation

Les résultats de la simulation du prototype SDHCAL et des tests sur faisceau ont été présentés dans les sections précédentes. La figure 3.28 présente le nombre de hits moyen en fonction de l'énergie du faisceau pour les données et différentes listes physiques. On peut observer qu'à partir de 50 GeV , le nombre de hits moyen est plus élevé dans les données. Ce phénomène s'accroît avec l'énergie du faisceau avec en moyenne 15% de déviation relative à 80 GeV (par rapport aux données) pour toute les listes de physique. La contamination ne permet cependant pas d'expliquer cette différence.

Une étude des différentes variables topologiques des gerbes hadroniques a été menée afin de caractériser les gerbes hadroniques [43]. Une variable particulièrement importante dans la suite de ces travaux de recherches émerge de cette étude : la valeur moyenne des profils latéraux des gerbes hadroniques. La figure 3.29 montre cette variable et sa déviation relative en fonction de l'énergie du faisceau pour les données et différentes listes physiques. On observe un désaccord entre les données et les différentes simulations.

Comme nous allons le voir, ces désaccords seront considérés lors de l'étude des performances de l'algorithme ArborPFA pour le prototype du SDHCAL, notamment lorsque deux gerbes hadroniques proches seront superposées l'une à côté de l'autre. Dans ce cas, la différence au niveau des profils latéraux aura un impact sur

le contact entre les gerbes et donc l'efficacité de séparation des contributions des gerbes hadroniques par l'algorithme de reconstruction (voir section 3 du chapitre 4).

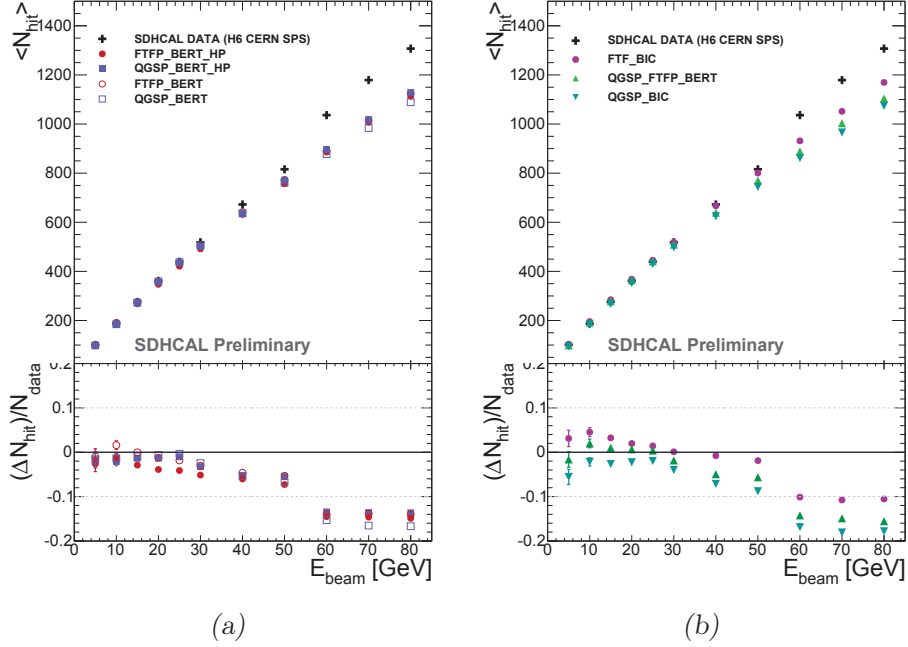


FIGURE 3.28 – Nombre de hits moyen en fonction de l'énergie du faisceau pour les données (croix noires), (a) les listes physiques FTFP_BERT(_HP), QGSP_BERT(_HP) et (b) les listes physiques FTF_BIC, QGSP_FTFP_BERT et QGSP_BIC [43].

5 Conclusion

Un prototype de calorimètre à lecture semi-digitale a été développé principalement à l'Institut de Physique Nucléaire de Lyon. D'une taille d'environ $1.3 m^3$, il est composé de 48 chambres à plaques résistives de verre, intercalées de couches d'absorbéur en acier. Les signaux issus des avalanches électroniques sont collectés sur des carreaux en cuivre d'une taille de $1 \times 1 cm^2$. L'électronique de lecture, composée de 144 ASICs et 3 DIFs par chambre, se charge de transmettre les données au système d'acquisition pour être écrites sur disque. Les signaux mis en forme par les ASICs sont échantillonnés suivant trois seuils de charge quantifiant la densité de particules traversant la couche de gaz. Une simulation numérique basée sur GEANT4 a aussi été développée, ainsi qu'un programme de digitalisation afin d'obtenir des données comparables à celles enregistrées lors des différents tests sur faisceau du prototype physique SDHCAL.

Afin d'extraire les performances du détecteur, une sélection des gerbes hadronique est opérée, et ce, à l'aide de variables topologiques. La sélection des muons a permis d'extraire une efficacité moyenne de 96% sur l'ensemble du détecteur, ainsi

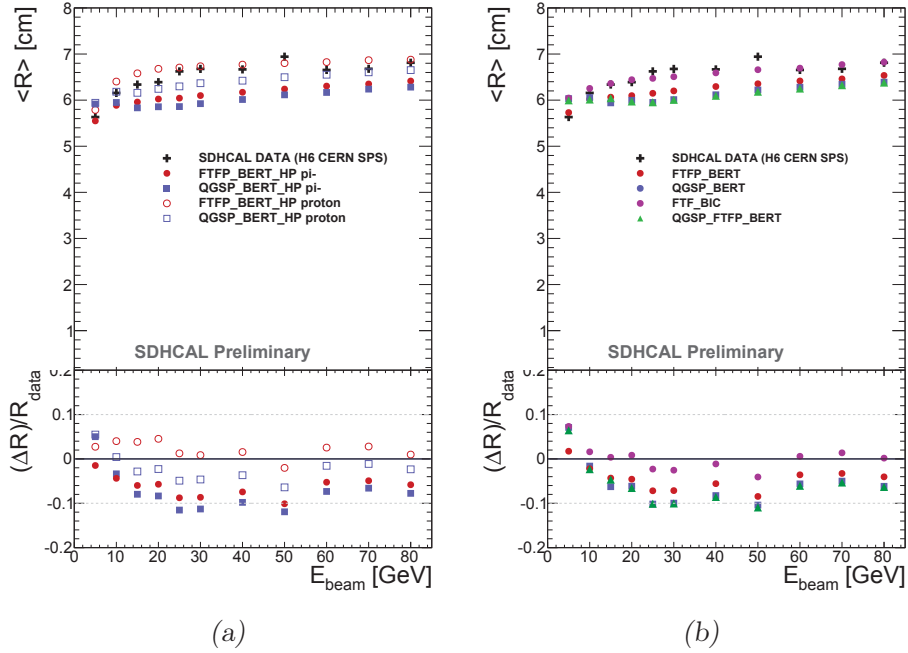


FIGURE 3.29 – Valeur moyenne et déviation relative du profil latéral en fonction de l'énergie du faisceau pour les données (croix noires) et plusieurs listes physiques. La figure 3.29(a) montre aussi des résultats pour des protons [43].

qu'une multiplicité moyenne de 1.76. Pour reconstruire l'énergie des hadrons, différents estimateurs ont été testés, tous utilisant le nombre de total de hits comme variable principale. L'utilisation des trois seuils dans le calcul de l'énergie a notamment permis de lutter contre la saturation à haute énergie et d'obtenir une meilleure linéarité à basse énergie. Une chute du nombre de hits en fonction du temps de l'événement au sein du *spill* due à la saturation du détecteur a été observée. Des corrections supplémentaires ont été apportées à l'estimateur en énergie pour corriger cette chute. Les résultats ont montré une bonne linéarité en énergie reconstruite avec une déviation globalement inférieure à 5%. La résolution relative en énergie quant à elle descend à 7.5% à 80 GeV. Des désaccords entre la simulation et les données ont aussi été mis en évidence au niveau du nombre total de hits et des profils transverses des gerbes hadroniques.

Chapitre 4

Les algorithmes de suivi de particules

CE chapitre est dédié aux algorithmes de suivi de particules, utilisés dans le cadre de la reconstruction des événements dans les détecteurs à grande granularité. Le premier algorithme de suivi de particules à avoir été développé est PandoraPFA. Il a montré qu'il pouvait atteindre les exigences en résolution en énergie des jets souhaitées ($\sigma_E/E \sim 3\%$). Cependant, son implémentation a été optimisée pour un calorimètre hadronique à lecture analogique avec des cellules d'une taille de $3 \times 3 \text{ cm}^2$. Une alternative, appelée ArborPFA, a été conçue afin de prendre en compte la grande granularité du SDHCAL et sa lecture semi-digitale. Après avoir introduit le principe des algorithmes de suivi de particules, l'algorithme PandoraPFA sera détaillé. Ses performances et ses limitations seront discutées. Le principe de l'algorithme ArborPFA sera exposé en fin de chapitre. Les différentes implémentations dédiées au prototype du SDHCAL et au détecteur ILD feront l'objet de chapitres séparés (chapitres 5 et 6).

Contenu

1	Principe des algorithmes de suivi de particules	66
2	L'algorithme PandoraPFA	68
2.1	L'environnement de développement PandoraSDK	68
2.2	La fonction de calcul des pseudo-plans	70
2.3	Les algorithmes	71
2.4	Les performances	76
2.5	Les limites de PandoraPFA	77
3	ArborPFA : il était une fois, des arbres	79

1 Principe des algorithmes de suivi de particules

Le concept des algorithmes de suivi de particules a été introduit dans les années 2000 dans le rapport du projet TESLA [5]. L'idée est d'utiliser le sous-détecteur le plus approprié pour effectuer la mesure en énergie la plus précise possible de chacune des particules. Par exemple, le trajectographe fournira une meilleure précision que les calorimètres sur la mesure de l'énergie des particules chargées. Le calorimètre électromagnétique sera un choix judicieux pour la mesure de l'énergie des photons. On peut distinguer deux concepts d'algorithmes. L'approche *PFA* ou *PFlow* (*Particle Flow Algorithm* ou *Particle Flow*) se différencie du *EFA* ou *EFlow* (*Energy Flow Algorithm* ou *Energy Flow*) par l'utilisation de la topologie en plus de l'information sur l'énergie dans les calorimètres. L'utilisation de la topologie dans la reconstruction nécessite des calorimètres à grande granularité. C'est ce qui opposera l'approche du *EFlow* utilisé par la collaboration ALEPH [12], à l'approche du *PFlow* utilisée par l'ILC, CLIC ou au CEPC et que nous discuterons par la suite dans ce chapitre.

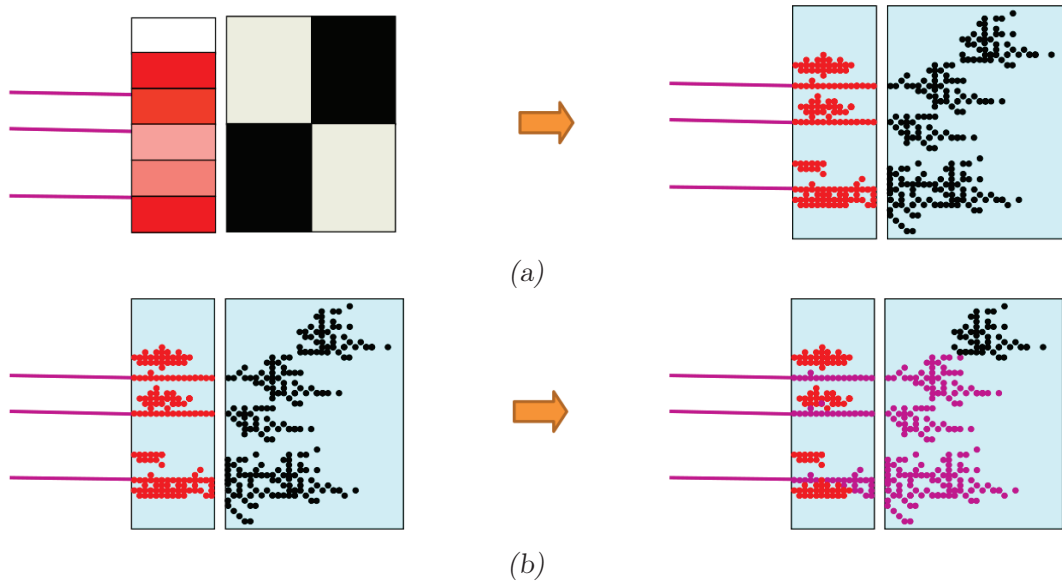


FIGURE 4.1 – Schémas illustrant la mesure de l'énergie des jets en exploitant a) la granularité des détecteurs et b) les algorithmes de reconnaissance de formes.

La figure 4.1(a) oppose les détecteurs de faible granularité (à gauche) aux détecteurs de haute granularité (à droite) comme ceux que l'on peut trouver à l'ILC. La figure 4.1(b) oppose quant à elle la mesure traditionnelle de l'énergie des jets (à gauche) en comptabilisant l'énergie électromagnétique (en rouge) et hadronique (en noire) contenue dans un jet, aux techniques utilisant des algorithmes de reconnaissance de formes (à droite) comme les algorithmes de suivi de particules, permettant de reconstruire les particules individuellement et ainsi comptabiliser trois énergies

au sein d'un jet :

$$E_{jet} = E_{track} + E_{\gamma} + E_n \quad (4.1)$$

avec E_{track} l'énergie des particules chargées (en rose), E_{γ} l'énergie des photons (en rouge) et E_n l'énergie des hadrons neutres (en noir).

La table 4.1 montre la fraction d'énergie issue d'un jet de 100 GeV pour différents types de particules. La reconstruction des particules chargées ($\sim 65\%$) est principalement guidée par la mesure de l'impulsion de la trace dans le trajectographe et par la topologie des gerbes dans les calorimètres. Les photons ($\sim 25\%$) eux ne bénéficient pas d'une mesure de l'impulsion dans le trajectographe, mais leur topologie très caractéristique rend leur identification relativement moins complexe. Le principal défi est alors de reconstruire correctement les hadrons neutres ($\sim 10\%$), parfois en contact avec d'autres particules dans les calorimètres au sein d'un jet. L'utilisation de la topologie est pour cela un atout majeur, puisqu'elle permettra de mieux séparer les dépôts d'énergie appartenant aux hadrons neutres des autres particules. Même si la part d'énergie déposée par les hadrons neutres est plus faible que pour les autres types de particules, ceux-ci pourraient les contaminer et dégrader la résolution en énergie des jets. Les algorithmes devront donc utiliser au maximum la granularité des détecteurs pour appliquer des méthodes de reconnaissance de formes capables de distinguer les différentes gerbes et de les associer entre elles si besoin est.

	Fraction en énergie	Sous-détecteur optimal	Performance attendues
Particules chargées	65%	Trajectographe	$\delta(\frac{1}{p}) = 8.10^{-5} (GeV/c)^{-1}$
Photons	25%	ECAL	$\frac{\sigma_E}{E} = 12\%/\sqrt{E}$
Hadrons neutres	10%	ECAL+HCAL	$\frac{\sigma_E}{E} < 60\%/\sqrt{E}$

TABLE 4.1 – Table résumant les performances requises pour la reconstruction des particules issues d'un jet de 100 GeV, dans un détecteur hautement granulaire (i.e type ILD ou CLIC).

L'application d'un PFA dans ce type de détecteurs nécessite aussi un budget matière faible dans le trajectographe pour éviter des pertes d'énergies trop importantes et des interactions supplémentaires qui pourraient compliquer la tâche de l'algorithme. Les calorimètres devront contenir les gerbes électromagnétiques et hadroniques et présenter le moins possible de zones mortes afin de mesurer au mieux l'énergie des particules.

En résumé, les techniques de suivi de particules sont une combinaison de détecteurs hautement granulaires et d'algorithmes de reconnaissances de formes adaptés à la topologie des gerbes électromagnétiques et hadroniques observées dans ces détecteurs. L'implémentation de ces algorithmes sera plus ou moins spécifique en fonction

de la granularité des détecteurs et donc de la topologie des gerbes. Certains pourront, suivant leur généricité, être réutilisés par différentes expériences.

Plusieurs algorithmes de suivi de particules ont été développés par différentes collaborations. L’algorithme PandoraPFA, le plus abouti de ces algorithmes à ce jour, a principalement été développé pour les détecteurs ILD, SiD et CLIC. D’autres algorithmes de suivi de particules ont été conçus mais sont plus spécifiques aux détecteurs pour lesquels ils ont été développés. Parmi eux, on peut bien sûr citer le PFlow du détecteur CMS au LHC ou encore l’algorithme IowaPFA développé pour le détecteur SiD de l’ILC, comme alternative à l’algorithme PandoraPFA.

2 L’algorithme PandoraPFA

PandoraPFA est un algorithme de suivi de particules écrit en C++, conçu par le laboratoire Cavendish à Cambridge [44]. Cet algorithme est composé de plusieurs sous-algorithmes traitant chacun un problème de reconnaissance de formes particulier dans les calorimètres. L’algorithme de clustering principal est basé sur une forme de cône, qui agglomère récursivement les hits pour former des amas. La spécificité de l’algorithme PandoraPFA réside cependant dans le principe du *re-clustering* statistique. Les particules chargées présentant une différence importante entre l’impulsion de la trace associée et de l’énergie mesurée dans les calorimètres font l’objet d’une analyse plus poussée. L’algorithme de clustering principal est alors exécuté plusieurs fois avec des paramètres différents afin de trouver la meilleure compatibilité entre l’impulsion de la trace et l’énergie mesurée dans les calorimètres.

L’algorithme PandoraPFA a particulièrement évolué depuis la première publication de ses performances en 2009 [44]. Historiquement, le code originel de l’algorithme PandoraPFA permettait d’obtenir les résolutions en énergie des jets attendues à l’ILC pour le détecteur ILD, c’est-à-dire de l’ordre de 3–4%. Cependant, son architecture ne prévoyait pas d’espace mémoire suffisant pour reconstruire certains événements issus du détecteur CLIC. La réécriture complète de l’algorithme original a permis de le séparer en trois parties : un environnement de développement générique pour les algorithmes de suivi de particules (PandoraSDK), l’implémentation des algorithmes du code originel (LCContent) et un processeur MARLIN [25] qui charge le contenu algorithmique et les événements à reconstruire (MarlinPandora).

2.1 L’environnement de développement PandoraSDK

PandoraSDK est un environnement de développement d’algorithmes de reconnaissance de formes conçus pour la physique des hautes énergies. La figure 4.2 illustre les interactions entre l’environnement de développement et l’utilisateur. D’un côté, à gauche, une application charge tous les détails relatifs aux détecteurs et les algorithmes choisis pour reconstruire les événements dans l’environnement PandoraSDK. Ce dernier, au milieu, fournit le contenu nécessaire à la reconstruction physique. Enfin, à droite, c’est la partie qui implémente les algorithmes de suivi de particules. C’est ici que le physicien posera sa propre logique au travers de ses algorithmes. Des APIs sont fournis à l’utilisateur pour simplifier et accélérer le développement

des algorithmes de reconstruction. L'utilisateur peut fournir en plus de ses propres algorithmes :

- une fonction d'identification de particules (pour les muons, les photons et les électrons)
- une fonction pour calculer le champ magnétique en tout point
- des fonctions de correction d'énergie, pour corriger l'énergie électromagnétique et hadronique des clusters
- une fonction de calcul des pseudo-plans (voir section 2.2)
- une fonction de calcul de profil de gerbes (position du début de la gerbe, profil longitudinal, profil transversal)

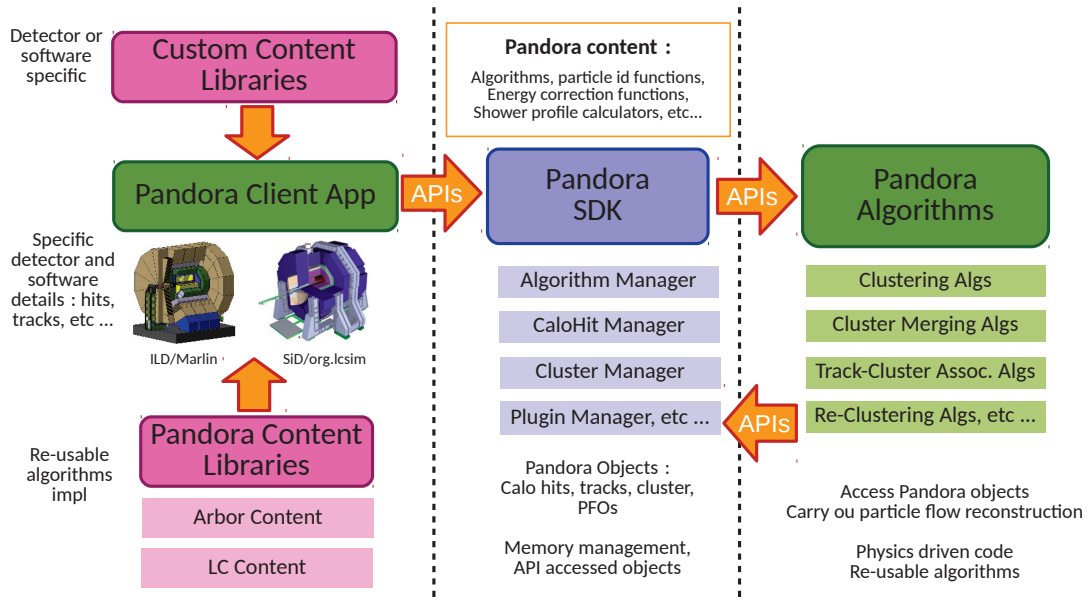


FIGURE 4.2

L'environnement contrôle les accès mémoires pour l'utilisateur, ce qui assure leur intégrité tout le long de l'exécution des algorithmes. Parmi les fonctionnalités intéressantes, on trouve une gestion des objets Pandora sous forme de listes, permettant de traiter un problème de reconnaissance de forme localement, par exemple, sur une liste particulière de clusters. Des fonctionnalités plus avancées permettent de garder en mémoire les associations entre les clusters reconstruits dans les calorimètres et les trace issues du trajectographe. Une procédure de *re-clustering* permettra de reconsidérer la reconstruction d'un ou plusieurs clusters en récupérant leurs hits et en exécutant plusieurs algorithmes localement. Cela permet d'affiner la reconstruction des particules suivant certains critères physiques importants (compatibilité entre trace et cluster associés, critères topologique, etc ...).

Cet environnement a été utilisé pour réarranger le code de PandoraPFA mais a aussi servi à implémenter des algorithmes de reconnaissance de formes pour d'autres

expériences, notamment la reconstruction dans la TPC à argon liquide de la collaboration MicroBooNE (LArContent) [19]. Il sera par la suite utilisé pour développer les sous-algorithmes de l'algorithme ArborPFA (ArborContent), principal fruit de ces travaux de recherche.

2.2 La fonction de calcul des pseudo-plans

La fonction de calcul des pseudo-plans est utilisée pour abstraire la géométrie du détecteur, en le segmentant en une succession de couches concentriques, du point d'interaction vers l'extérieur. Elle est particulièrement utile dans certaines régions où l'ordonnancement des plans physiques des calorimètres ne reflète plus la profondeur dans les calorimètres. Par exemple, dans la zone extérieure du calorimètre hadronique dans les bouchons, le plan physique 1 est situé juste après le plan physique 29 du calorimètre électromagnétique du tonneau central.

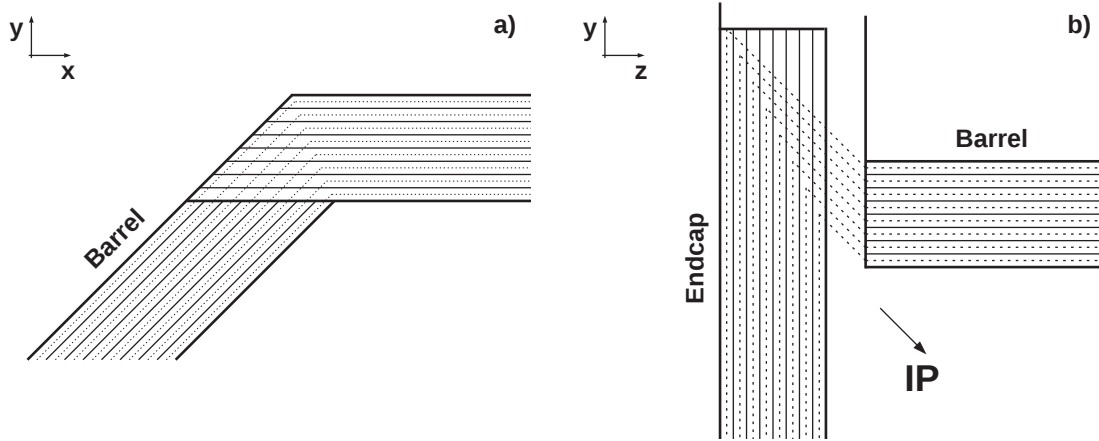


FIGURE 4.3 – Découpage de la géométrie du détecteur en pseudo-plans dans a) la région du tonneau et b) la région des bouchons. Les lignes pleines indiquent les plans physiques du calorimètre électromagnétique et les lignes pointillées les pseudo-plans.

La figure 4.3 illustre le découpage de la géométrie du détecteur en pseudo-plans dans (a) la région du tonneau et (b) des bouchons pour les premiers plans du calorimètre électromagnétique. Le tonneau est découpé suivant l'angle ϕ en segments de même taille et la distance entre les pseudo-plans suit simplement celle des plans physiques. En revanche, le découpage des bouchons en pseudo-plans est un peu plus complexe. Le découpage dans la partie $R < R_{ECAL}$ (où R_{ECAL} est le rayon interne du calorimètre électromagnétique dans le tonneau central) suit l'ordre des plans physiques comme pour le tonneau. La partie $R > R_{ECAL}$ utilise quant à elle la projection vers le point d'interaction de telle sorte que la numérotation des pseudo-plans soit fortement liée à la profondeur dans les calorimètres. Cela permettra d'ordonner les hits dans les calorimètres, non pas par détecteur (ECAL, HCAL, tonneau ou bouchon) mais suivant l'ordre logique du développement des gerbes. Ce découpage est aussi valable dans les chambres à muons. Le pseudo-plan 0 est utilisé

pour la projection des traces sur la face avant du calorimètre électromagnétique. La numérotation dans les calorimètres commencera donc par 1.

2.3 Les algorithmes

De façon générale et quelque soit le domaine, tout algorithme possède une ou plusieurs entrées et une ou plusieurs sorties. L'algorithme PandoraPFA requiert en entrée une liste de hits dans les calorimètres, une liste de trace dans le trajectographe et un jeu de paramètres pour configurer les algorithmes. Il fournit en sortie une liste de particules reconstruites contenant chacune une liste de clusters dans les calorimètres, une liste de traces du trajectographe (pour les particules chargées) et des méta-données relatives à la particule (code pdg, charge, masse, énergie, etc ...). À chaque hit et chaque trace peuvent aussi être associées des méta-données Monte Carlo issues de la simulation. Ces méta-données, disponibles dans les particules reconstruites en bout de chaîne, permettront d'effectuer des comparaisons avant et après reconstruction pour évaluer les performances de l'algorithme.

Une fois les entrées chargées dans l'environnement Pandora, l'algorithme se déroule de la façon suivante :

Préparation de l'événement

Quatre formes de traces sont repérées (voir figure 4.4 (a)) et traitées en amont. Seules les traces atteignant les calorimètres et ayant une énergie suffisamment élevée pour créer des hits dans les calorimètres seront utilisées durant la phase de clustering et d'associations topologiques. Diverses quantités, telles que l'isolation ou l'étiquetage des MIPs, sont calculées pour chaque hit.

Clustering principal

L'algorithme principal de clustering est un algorithme itératif, balayant les calorimètres pseudo-plan par pseudo-plan, du point d'interaction vers l'extérieur, à la recherche de hits contenus dans un cône pour former des clusters (voir figure 4.4 (b) pour une illustration). Les points de départ (ou *seeds*) des clusters sont donnés soit par la projection des traces disponibles dans le trajectographe sur la face avant du calorimètre électromagnétique (cas des hits en rouge sur la figure) soit par un hit non aggloméré dans un pseudo-plan (cas des hits en bleu). La direction du cône d'agglomération est donné soit par le vecteur d'impulsion de la trace soit par un ajustement linéaire en 3D sur la position des hits du cluster (pas de pondération sur les positions). Cet algorithme produit en sortie une liste de clusters reconstruits.

Associations topologiques

La philosophie de PandoraPFA repose sur le principe "il est plus simple d'associer des objets que de les dissocier". En effet, il est plus simple d'identifier deux structures à associer suivant des critères topologiques que d'analyser une structure complexe que l'on voudrait dissocier en plusieurs. L'algorithme de clustering discuté précédemment est donc conçu pour reconstruire au mieux

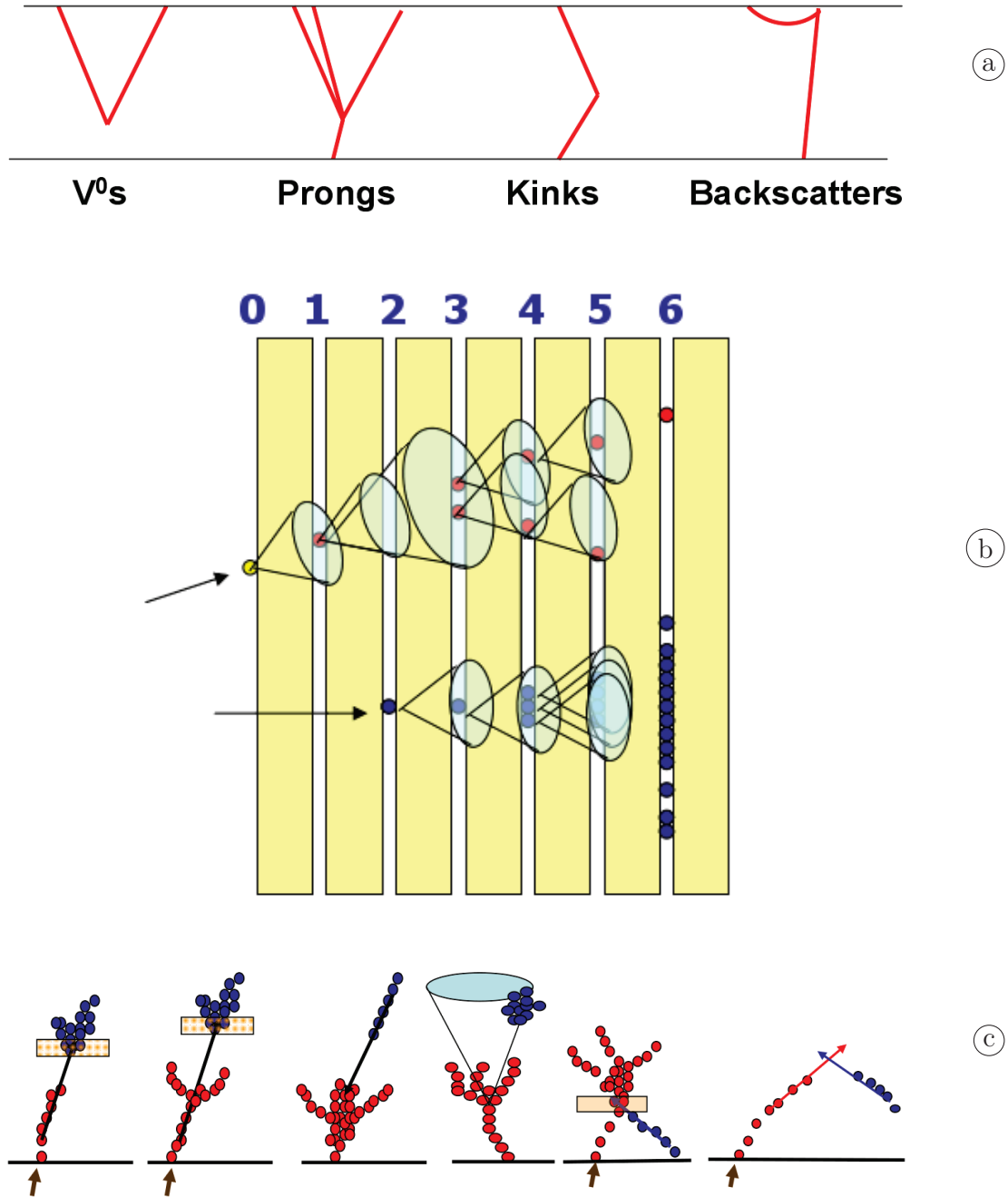


FIGURE 4.4 – a) Les quatre formes de traces traitées en amont par l’algorithme PandoraPFA. b) Schéma illustrant le principe de l’algorithme principal de clustering de PandoraPFA. c) Principales formes d’associations topologiques entre clusters dans PandoraPFA.

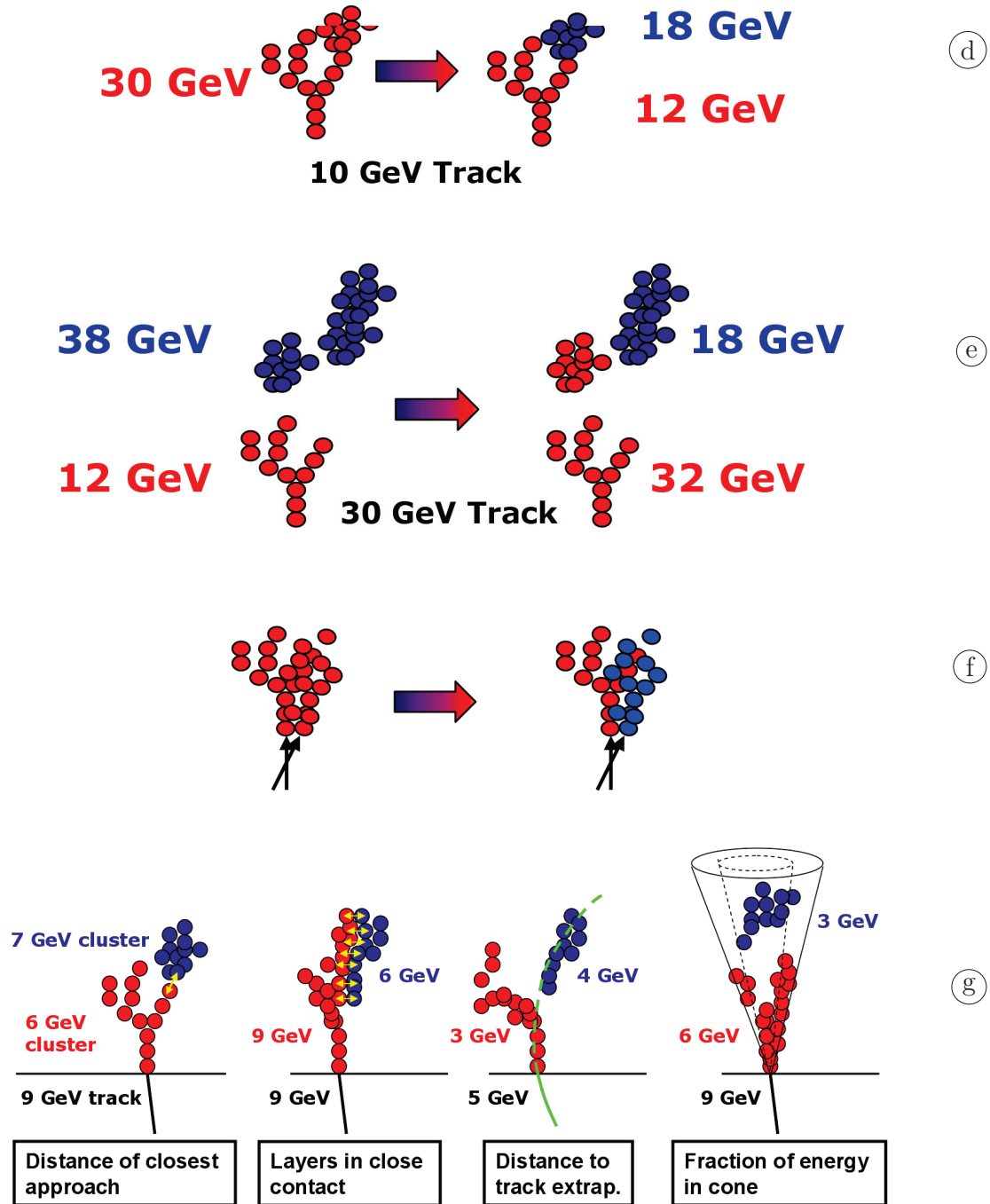


FIGURE 4.5 – d) e) et f) : Trois exemples de stratégies de re-clustering opérées par PandoraPFA ; g) Exemples de topologies traités par les algorithmes de suppression de fragments par PandoraPFA.

les clusters mais avec des paramètres qui favorisent la création de petites structures. La série d'algorithmes qui suit, permet, suivant des critères topologiques et des critères énergétiques, d'associer les traces du trajectographe avec les clusters et les clusters entre eux.

La toute première étape est d'associer les traces du trajectographe avec les clusters. La distance entre les premiers hits d'un cluster et la différence entre l'impulsion de la trace et l'énergie du cluster sont les principaux critères de validité pour une association. On définit la compatibilité $E - p$ entre une trace et un cluster c par la variable χ_c :

$$\chi_c = \frac{E_c - p_{track}}{\sigma_p}, \quad \text{avec } \sigma_p = a_h \cdot \sqrt{p_{track}} \quad (4.2)$$

avec E_c l'énergie du cluster c , p_{track} l'impulsion de la trace et a_h le terme stochastique de la résolution en énergie du calorimètre hadronique (voir section 2 du chapitre 3). Une compatibilité négative sera signe d'une énergie manquante dans le cluster par rapport à la trace et au contraire une compatibilité positive sera signe d'une énergie additionnelle dans le cluster. Certains algorithmes d'association tenteront d'associer un cluster fils avec un cluster parent en vérifiant si la compatibilité $E - p$ s'améliore après la fusion des clusters. Ceci se traduit par le test suivant :

$$\chi_p^2 - \chi_{p+f}^2 \quad \begin{cases} > 0 & \text{association possible} & (1) \\ < 0 & \text{association rejetée.} & (2) \end{cases} \quad (4.3)$$

avec χ_p la compatibilité du cluster parent avec sa trace et χ_{p+f} la compatibilité de la somme des clusters parent et fils avec cette même trace. Pour le cas (1), on parlera par la suite *d'amélioration de la compatibilité $E - p$* .

La figure 4.4 (c) illustre quelques exemples d'associations topologiques que l'on peut trouver dans cette série d'algorithmes. Les deux premiers cas utilisent la direction du cluster parent (en rouge) pour chercher des associations évidentes avec un cluster fils proche (en bleu). Le troisième, au contraire, considère la direction du cluster fils pour tester une fusion possible avec cluster parent proche. Le quatrième utilise ici un cône, dont la base pointe sur le début de la gerbe et dont la direction est donnée par la composante principale du cluster, pour chercher une fusion avec des potentiels clusters fils améliorant la compatibilité $E - p$. Le cinquième traite les cas de rétrodiffusion au sein des gerbes hadronique alors que le dernier tente de reconstituer un segment de trace courbée dans les calorimètres qui a été brisé du à un angle trop important. Il est à noter que toutes ces méthodes de reconnaissance de formes topologiques sont rendues possible grâce à la grande granularité des calorimètres.

Le re-clustering

Arrivé à ce stade, les particules sont reconstruites et la plupart du temps, de façon satisfaisante. Cependant, certains cas peuvent se montrer plus compliqués, les paramètres de l'algorithme de clustering ne pouvant satisfaire toute

les topologies. Les erreurs les plus fréquentes commises par la reconstruction viennent de la confusion engendrée par des particules en contact dans les calorimètres. Cette source peut provenir des hadrons neutres ou peut être due à un environnement très dense dans les jets de hautes énergies. Le principal estimateur utilisé par PandoraPFA pour juger de la qualité de la reconstruction pour chaque particule chargée est sa compatibilité $E - p$. Diverses stratégies sont employées pour traiter les différentes topologies problématiques. Dans tous les cas, la stratégie globale employée par PandoraPFA est celle décrite par l'algorithme 1. Elle peut se résumer en une optimisation de la comptabilité $E - p$ pour chaque particule chargée en exécutant l'algorithme de clustering et les associations topologiques plusieurs fois, avec des paramètres différents.

Algorithm 1 Pseudo-code décrivant un algorithme de re-clustering.

```

1: procedure RECLUSTERING
2:   Get current cluster list
3:   for all clusters do
4:     if track-cluster not compatible then
5:       Extract Cluster CaloHits
6:       for all Re-Clustering algorithms do
7:         Form new Clusters with CaloHits
8:         if Compatibility improved then
9:           Update best compatibility
10:        end if
11:       end for
12:       Load best Cluster list with best compatibility
13:       Remove all other Clusters
14:     end if
15:   end for
16: end procedure

```

La figure 4.5 (d) montre une première topologie problématique avec un surplus d'énergie dans un cluster en comparaison à l'impulsion de la trace. Le problème topologique (e) présente un cluster avec de l'énergie manquante par rapport à la trace. Afin de contrebalancer ce manque, il devient nécessaire de chercher des hits supplémentaires dans les clusters neutres environnants. Le troisième problème topologique (f) peut se rencontrer lorsque deux traces ont été associées à un seul cluster. Dans ce cas, l'utilisation d'un critère de proximité entre chaque hit et une trace permettra de séparer le cluster en plusieurs contributions, sans utiliser la stratégie décrite par l'algorithme 1.

Suppression des fragments

Après une optimisation poussée au cas par cas pour chaque particule chargée, certaines particules peuvent avoir été reconstruites avec une bonne compatibilité $E - p$ mais avec quelques fragments qui ne sont pas associés. Une série finale d'algorithmes permettra d'identifier des fragments proches de particules

chargées montrant des signes d'associations évidents. Ces algorithmes tenteront d'associer ces fragments fils à des clusters parents en gardant une bonne compatibilité $E-p$ même si celle-ci n'est pas optimale. La figure 4.5 (g) montre quelques exemples d'associations, utilisant les critères principaux habituels : la proximité spatiale et la compatibilité $E-p$.

Création des particules reconstruites

La dernière étape de PandoraPFA consiste en la création des particules reconstruites ou PFOs (Particle Flow Objects). À ce stade, la fonction d'identification de particule est utilisée pour assigner un type de particule (électrons, photons, muons, hadrons chargés ou neutres) à chaque cluster. Un PFO est constitué d'une liste de clusters, de traces et de méta-données relatives à la particule reconstruite. Les signatures laissées par les traces (voir figure 4.4 (a)) sont utilisées pour créer des PFOs plus complexes composés par exemple de deux traces et leurs clusters associés, dans le cas d'une signature en V^0 . À noter qu'un PFO peut aussi être composé uniquement de traces si celles-ci ne sont pas associées à des clusters dans les calorimètres.

2.4 Les performances

La principale observable étudiée pour l'estimation des performances d'un algorithme de suivi de particule est la résolution en énergie des jets σ_j/E_j . Dans le cadre de l'ILC, cette résolution doit être de l'ordre de $30\%/\sqrt{E}$, afin de garantir une bonne identification des bosons Z^0 et W^\pm [11]. L'algorithme PandoraPFA a été appliqué sur des événements di-jets issus de bosons Z^0 virtuels en paires $q\bar{q}$ (uds), dans le détecteur ILD, afin d'évaluer ses performances. Les bosons Z^0 sont générés à l'origine des coordonnées, avec une énergie de 91, 200, 360 et 500 GeV . Les quarks sont émis avec la même énergie (la moitié de l'énergie du Z^0 chacun) et une impulsion opposée.

Comme il a été discuté dans le chapitre 2, plusieurs calorimètres électromagnétiques et hadroniques sont candidats pour équiper le détecteur ILD. Dans une première version, dénotée ILD_o1_v05, le détecteur ILD est équipé d'un calorimètre électromagnétique en silicium (SiECAL), d'une granularité de $5 \times 5 \text{ mm}^2$ et d'un calorimètre hadronique analogique avec des scintillateurs en plastique (AHCAL), d'une granularité de $30 \times 30 \text{ mm}^2$. La deuxième version, dénotée ILD_o2_v05, verra le détecteur ILD équipé du même calorimètre électromagnétique mais d'un calorimètre hadronique à lecture semi-digitale avec des chambres à plaques résistives en verre (SDHCAL), d'une granularité de $10 \times 10 \text{ mm}^2$. Dans les deux cas, les paramètres de tous les autres détecteurs sont les mêmes que ceux présentés dans la section 2.2.1 du chapitre 2.

La résolution en énergie des jets est calculée en utilisant les grandeurs RMS_{90} et $Mean_{90}$ définis comme l'écart-type et la moyenne de la distribution en énergie contenant 90% de la statistique. Cette définition permet d'être relativement peu

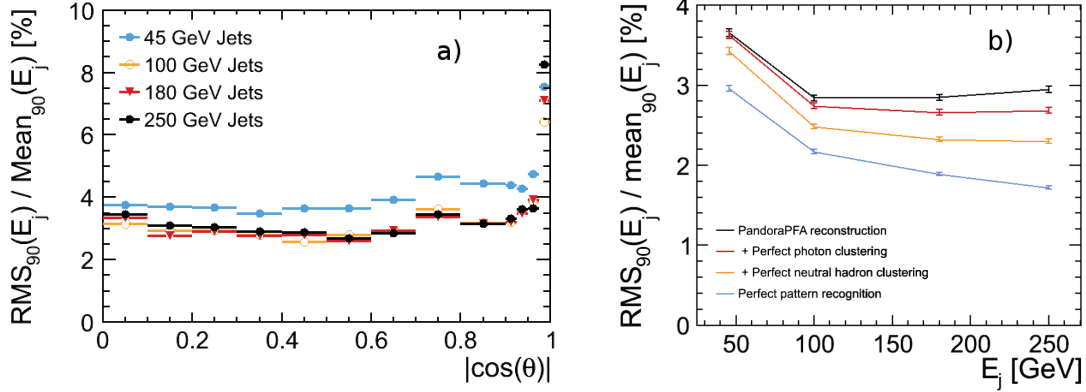


FIGURE 4.6 – Performances de l'algorithme PandoraPFA sur des événements di-jets *uds* dans le modèle *ILD_o1_v05* (AHCAL) : a) la résolution en énergie des jets en fonction du cosinus de l'angle entre un jet et l'axe du faisceau et b) la résolution en énergie des jets en fonction de l'énergie des jets.

influencé par les queues de distribution [44]. La grandeur $|\cos\theta|$ est définie comme :

$$|\cos\theta| = \frac{1}{E_{tot}} \sum_q \frac{|p_{z,q}| \cdot E_q}{\|\vec{p}_q\|} \quad (4.4)$$

avec \vec{p}_q l'impulsion du quark q , E_q l'énergie du quark q , $p_{z,q}$ la coordonnée du quark q suivant l'axe du faisceau et E_{tot} l'énergie totale du système di-jets. Les paramètres géométriques du détecteur ILD donnent alors $|\cos\theta| > 0.7$ pour les bouchons et $|\cos\theta| < 0.7$ pour la région du tonneau.

La figure 4.6 montre les performances de l'algorithme PandoraPFA sur des événements di-jets *uds* avec le modèle *ILD_o1_v05* [44]. La figure 4.6 a) montre que la résolution en énergie des jets dans la région du tonneau est inférieure à 4%. La dégradation de la résolution à presque 5% pour $|\cos\theta| = 0.7$ correspond à la région intermédiaire entre le tonneau et les bouchons, dans laquelle une part de l'énergie des jets est perdue dans les parties non-instrumentées du détecteur. La figure 4.6 b) montre quant à elle la résolution en énergie des jets en fonction de l'énergie des jets. L'algorithme PandoraPFA donne la possibilité de reconstruire en amont certaines particules en utilisant les informations Monte Carlo. L'algorithme est ensuite exécuté sur l'événement restant. Cela permet d'étudier les différentes parts de confusion dues à certains types de particules afin de mieux comprendre les points forts et les points faibles de l'algorithme. Cette figure montre la reconstruction standard de l'algorithme (en noir), une reconstruction parfaite des photons (en rouge), une reconstruction parfaite des photons et hadrons neutres (en jaune) et une reconstruction parfaite de l'événement (en bleu).

2.5 Les limites de PandoraPFA

Nous venons de voir les performances de l'algorithme PandoraPFA pour le modèle *ILD_v05_o1* avec un calorimètre hadronique analogique. La résolution en

énergie des jets atteint la valeur requise par l'ILC de 3%. Cependant l'algorithme PandoraPFA a été développé pour un calorimètre hadronique analogique avec une segmentation de $30 \times 30 \text{ mm}^2$. L'implémentation actuelle requiert qu'avec chaque hit dans les calorimètres soit fournie une énergie calibrée. Même si l'environnement PandoraSDK laisse la place pour une option de calorimètre digital, celle-ci n'est pas traitée dans l'algorithme PandoraPFA. L'option semi-digitale est quant à elle un peu plus complexe à traiter. Nous avons vu au chapitre 3 que le calorimètre hadronique semi-digitale comportait des avantages à utiliser trois seuils pour la reconstruction en énergie, particulièrement à haute énergie. Cette utilisation des trois seuils a été exploitée en utilisant une forme non-linéaire, que nous rappelons ici :

$$\begin{aligned}
 E &= (\alpha_1 + \alpha_2 \cdot N_{hit} + \alpha_3 \cdot N_{hit}^2) \cdot N_1 \\
 &+ (\beta_1 + \beta_2 \cdot N_{hit} + \beta_3 \cdot N_{hit}^2) \cdot N_2 \\
 &+ (\gamma_1 + \gamma_2 \cdot N_{hit} + \gamma_3 \cdot N_{hit}^2) \cdot N_3
 \end{aligned} \tag{4.5}$$

avec N_{hit} le nombre total de hits pour 1 PFO. En terme d'algorithmie, cela implique que pour deux clusters d'énergie E_1 et E_2 , leurs énergies ne peuvent être additionnées de manière linéaire :

$$E_{1,2} \neq E_1 + E_2 \tag{4.6}$$

Ceci est aussi vrai lors de l'ajout d'un hit à un cluster. Dans l'algorithme PandoraPFA, la reconstruction de l'énergie est supposée linéaire et la majorité des algorithmes effectuant des fusions de clusters ou des ajouts de hits à un cluster font cette supposition.

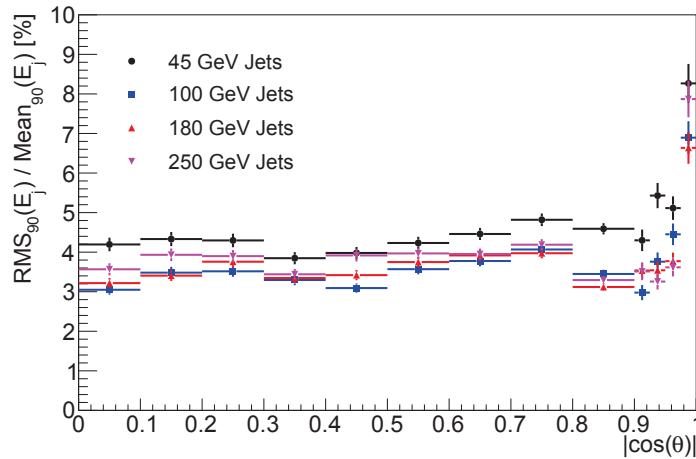


FIGURE 4.7 – Performances de l'algorithme PandoraPFA sur des événements di-jets uds dans le modèle ILD_o2_v05 (SDHCAL) : résolution en énergie des jets en fonction du cosinus de l'angle entre un jet et l'axe du faisceau

Des études ont cependant été faites [42] avec l'algorithme PandoraPFA et le modèle ILD_v05_o2 (SDHCAL) en utilisant une reconstruction en énergie avec une

forme linéaire, en assignant à chaque hit une valeur en énergie. La résolution en énergie des jets est montrée sur la figure 4.7. Ces performances sont légèrement moins bonnes que celles obtenues pour le modèle ILD_v05_o1 avec un calorimètre hadronique analogique. Ces résultats suggèrent que i) l'algorithme doit être adapté au cas semi-digital en modifiant le code source ou ii) un autre algorithme plus adapté au cas semi-digital doit servir d'alternative.

Basé sur cette deuxième solution, un algorithme de suivi de particules exploitant toutes les capacités du calorimètre hadronique à lecture semi-digitale (trois seuils et granularité plus élevée) a été développé.

3 ArborPFA : il était une fois, des arbres ...

L'approche de l'algorithme ArborPFA n'est pas récente. Le concept a été introduit par la collaboration ALEPH dans la mesure de la polarisation du τ [29]. Le principe a ensuite été dérivé pour des détecteurs à grande granularité pour l'ILC [41].

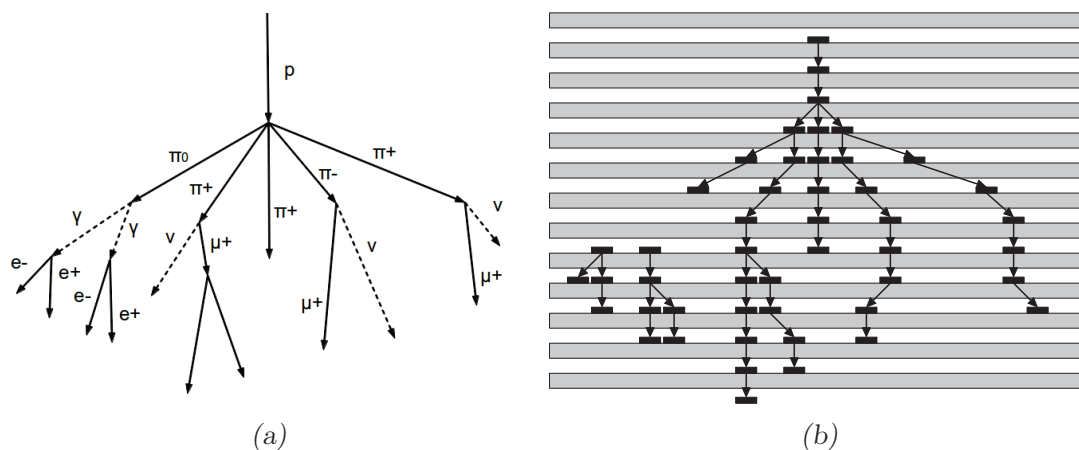


FIGURE 4.8 – (a) Illustration du développement d'une gerbe hadronique initiée par un proton. (b) La même gerbe hadronique telle qu'elle pourrait être détectée dans un calorimètre à échantillonnage. Les flèches suggèrent une reconstruction des gerbes hadroniques suivant une topologie en arbre orientée.

L'idée principale est basée sur le fait que les gerbes électromagnétiques et hadroniques suivent un développement suivant une topologie en arbre orientée. La figure 4.8(a) illustre le développement d'une gerbe hadronique initiée par un proton. La figure 4.8(b) illustre la même gerbe mais telle qu'elle serait détectée dans un calorimètre à échantillonnage à grande granularité. Les carrés noirs représentent les cellules touchées suite au développement de la gerbe hadronique. Le but de l'algorithme ArborPFA sera de connecter ces cellules pour obtenir des structures suivant chacune une topologie en arbre orientée (voir page 81 pour une définition exacte). Les flèches (ou connecteurs) tracées en supplément sur la figure 4.8(b) montrent un résultat possible de la reconstruction de telles structures après l'application de l'algorithme

ArborPFA. Ces connecteurs ne peuvent bien sûr lier que deux cellules allumées par des particules chargées. Par exemple, les deux petites structures à gauche sur la figure 4.8(b) sont issues de deux photons et seront reconstruites comme deux arbres à part. Des algorithmes d'associations, tels que ceux implémentés par l'algorithme PandoraPFA (voir section 2.3), seront nécessaires afin d'identifier et d'associer ce type de structures à d'autres. Une grande granularité dans le calorimètre hadronique permettra de mieux discerner les traces laissées dans le calorimètre et donc d'utiliser au mieux les arguments topologiques.

Cette approche a été choisie puisqu'elle reflète le mieux la physique sous-jacente des gerbes électromagnétiques et hadroniques. De ce fait, des arguments très proches de la physique seront utilisés pour opérer la reconstruction.

Afin de valider l'idée principale de cet algorithme, une première version a été développée pour le prototype du calorimètre hadronique à lecture semi-digitale. Cette version, exposée dans le chapitre 5, évaluera les performances de l'algorithme sur des pions chargés isolés et permettra de mieux comprendre son comportement. Une deuxième étude visant à séparer un pion chargé proche d'un hadron neutre évaluera les capacités de l'algorithme à séparer deux particules proches. Pour ces deux analyses, une étude systématique basée sur les variations de certains paramètres de l'algorithme a été réalisée afin d'évaluer avec plus de finesse les erreurs sur ces performances.

Une seconde version de l'algorithme, présentée dans le chapitre 6, a ensuite été développée pour un détecteur complet de physique des particules (trajectographe + ECal + HCal) et testée sur le détecteur ILD dans sa version ILD_v05_o2 avec un calorimètre hadronique semi-digital. Les performances de l'algorithme seront extraites en utilisant des événements de type di-jets, comme dans le cas de l'algorithme PandoraPFA, et seront comparées aux performances de ce dernier.

Pour les deux versions, l'algorithme utilise certaines définitions communes, à savoir :

- le pseudo-plan. Il représente la même quantité que celle décrite dans la section 2.2 pour l'algorithme PandoraPFA. Dans la première version de l'algorithme pour le prototype SDHCAL, la numérotation des pseudo-plans suivra l'ordre naturel des plans physiques, alors que pour la deuxième version, le calcul des pseudo-plans sera le même que celui utilisé dans l'algorithme PandoraPFA (voir section 2.2). On définit la direction *vers l'avant* comme l'ordre croissant des pseudo-plans et *vers l'arrière* comme l'ordre décroissant des pseudo-plans.
- le vertex. Il représente un point dans l'espace que l'on peut relier par un ou plusieurs connecteurs (un sommet en théorie des graphes). Dans la première version de l'algorithme ArborPFA pour le prototype SDHCAL, un groupe d'un ou plusieurs hits dans un même plan physique sera utilisé pour définir un point (voir la section 1 du chapitre 5). La deuxième version de l'algorithme pour les détecteurs complets verra la définition du vertex restreinte à un seul hit dans les calorimètres (voir la section 2 du chapitre 6).
- le connecteur. Il connecte deux vertex dans une direction donnée (une arrête en théorie des graphes). On parlera par la suite de connecteur dirigé vers

l'avant (vers l'arrière) s'il pointe vers un vertex dont le pseudo-plan est plus grand (plus petit) que celui dont il part. Deux vertex contenus dans un même pseudo-plan ne peuvent être connectés.

- un arbre. Un arbre dans le cadre de l'algorithme ArborPFA est défini comme étant, au sens de la théorie des graphes, un arbre enraciné avec les propriétés suivantes :
 - il est connexe, c'est à dire que quels que soient les vertex v_i et v_j de l'arbre, il existe une chaîne du vertex v_i vers le vertex v_j
 - il a un unique vertex sans prédécesseur, c'est-à-dire sans connexion vers l'arrière. Ce vertex est défini comme la racine, ou *seed*, de l'arbre.
 - tous les vertex, hormis la racine, possèdent uniquement un seul prédécesseur, c'est-à-dire une seule connexion vers l'arrière.

Les vertex ne possédant pas de successeur, c'est-à-dire qui n'ont pas de connexion vers l'avant, sont appelés des feuilles ou *leafs*.

- un cluster. Le cluster, ou amas, est composé d'un ou plusieurs arbres.
- une particule reconstruite ou PFO (pour **P**article **F**low **O**bject), est définie de la même manière que pour l'algorithme PandoraPFA, c'est-à-dire une liste de clusters, optionnellement une liste de traces dans le trajectographe et des métadonnées relatives à la particule reconstruite.

L'implémentation d'un algorithme de clustering en utilisant des arbres comporte de nombreux avantages. Les algorithmes de parcours d'arbre en profondeur ou en largeur de la théorie des graphes permettent de naviguer de façon récursive au sein d'un arbre. Leur utilisation sera très avantageuse lors de la création des arbres ou pour extraire des variables topologiques utiles lors de la phase d'association de clusters.

Chapitre 5

L'algorithme ArborPFA pour le prototype du SDHCAL

DANS le chapitre précédent, nous avons introduit les algorithmes de suivi de particules et le principe général de l'algorithme ArborPFA. Ce chapitre présente l'implémentation de cet algorithme dédiée au prototype du SDHCAL. Dans un premier temps, les algorithmes qui le composent seront détaillés. Une première étude sur des hadrons isolés permettant de comprendre les performances de l'algorithme, sera présentée. Une seconde étude visant à séparer deux hadrons proches dans le SDHCAL sera finalement exposée. Pour ces deux analyses, une étude systématique permettant d'évaluer avec plus de finesse les erreurs sur les performances sera décrite en fin de chapitre.

Contenu

1	Description de l'algorithme	84
2	Etude des hadrons isolés	93
2.1	Préparation des événements	93
2.2	Analyse des performances	93
3	Etude de séparation de deux hadrons proches	98
3.1	Superposition des événements	98
3.2	Analyse des performances	100
4	Les incertitudes systématiques	104
5	Conclusion	107

1 Description de l'algorithme

L'algorithme est globalement séparé en trois parties : la première partie est une phase de pre-clustering, la deuxième sera dédiée à la création des clusters puis dans la dernière partie, les clusters seront associés aux traces puis entre eux pour former des particules reconstruites. Les descriptions ne mentionnent pas les valeurs des paramètres des différents algorithmes. Les paramètres les plus importants sont donnés dans l'annexe A.

Création des vertex

Quand une particule traverse un détecteur gazeux, plusieurs cellules peuvent être allumées dans un plan. Ce phénomène est connu sous le nom de multiplicité et a été décrit dans le chapitre 3. Dans le cas d'une seule MIP laissant plusieurs hits sur son passage, il est illogique de considérer 4 vertex à connecter par la suite, étant donné qu'ils sont issus de la même particule. Le but de cet algorithme est de grouper ce type de hits ensemble pour former des vertex séparés. Les autres types de hits seront ensuite considérés comme des vertex singuliers.

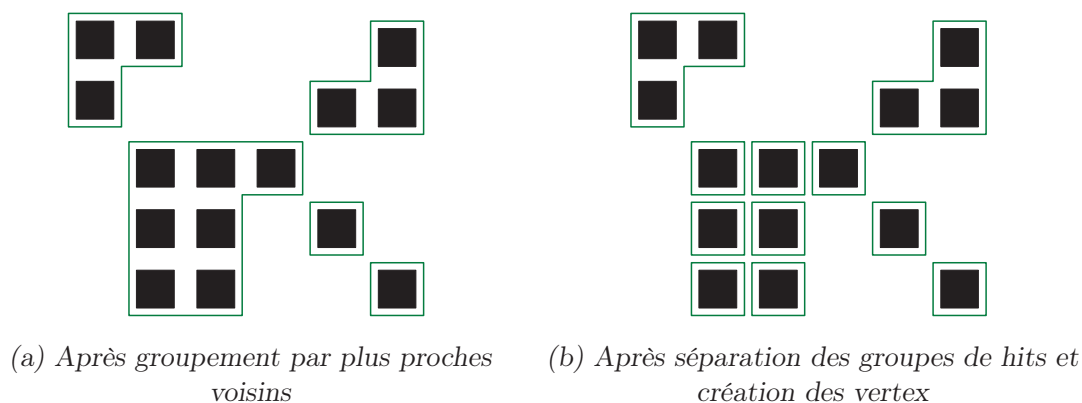


FIGURE 5.1 – (a) Exemple de groupement de hits (encadrés en vert) par plus proches voisins dans chaque plan physique, ici pour un plan particulier. (b) Exemple de séparation des groupes de hits et de création de vertex (encadrés en vert) à partir des groupes de hits de la figure (a).

Cet algorithme commence par former des groupes de hits en utilisant un algorithme de plus proches voisins, comme l'illustre la figure 5.1(a). Pour une particule seule (sans radiation) traversant une couche de gaz, la multiplicité maximale observée doit être de 4, si la particule traverse près de la jointure de quatre cellules. Ainsi, si un groupe est formé de quatre hits ou moins, il forme un vertex. Ceci correspond au cas des MIPs ou encore des hits isolés, dus par exemple à des neutrons lents. Dans le cas contraire, le groupe est brisé et chaque hit est considéré comme un vertex, comme illustré par la figure 5.1(b). C'est généralement le cas au cœur des gerbes où le nombre de hits voisins est plus important.

Étiquetage des vertex appartenant au segment de traces

Cet algorithme a pour but d'identifier et d'étiqueter les vertex susceptibles

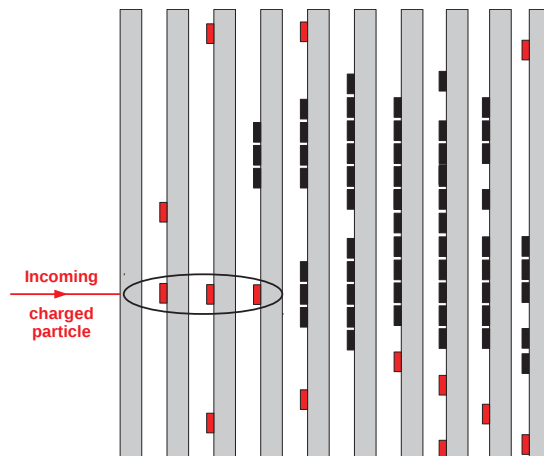


FIGURE 5.2 – Résultat de l'algorithme d'étiquetage des vertex appartenant au segment de traces. Les vertex étiquetés, en rouge, seront traités de manière particulière lors de la phase de création de clusters, notamment ceux appartenant aux segments de traces primaires, comme ceux entourés en noir.

d'appartenir au segment de traces traversant le calorimètre, comme les MIPs. Ces vertex seront traités de façon particulière dans la phase création de clusters. Pour chaque vertex, un comptage du nombre de vertex environnants dans le même pseudo-plan, à une distance maximale Δ_{mip} , est effectué. Si ce nombre n'excède pas $N_{mip, cut}$, le vertex considéré est étiqueté comme appartenant possiblement à un segment de trace. La figure 5.2 montre deux gerbes hadronique dans un calorimètre à échantillonnage. Les vertex étiquetés sont illustrés en rouge. Ces vertex seront principalement utilisés pour connecter les vertex appartenant aux segments de traces primaires, comme ceux entourés sur la figure 5.2.

La deuxième phase est celle de la création de clusters. Le but est de connecter les vertex pour former des arbres. Cette phase est divisée en plusieurs sous-algorithmes qui alternent la création de connexions et la formation d'arbres en supprimant certaines de ces connexions. À la fin de l'exécution de ces algorithmes, les arbres créés sont convertis en clusters, chacun contenant exactement un arbre.

Connexion des segments de traces primaires

Cet algorithme crée des connexions entre les vertex issus de chaque trace primaire des particules chargées, comme ceux entourés en noir sur la figure 5.2. La reconstruction de ces segments de traces est particulièrement importante par la suite pour l'association entre les clusters et les traces issues du trajectographe.

La réutilisation des algorithmes de l'environnement PandoraSDK est très utile dans le cadre de cet algorithme. En effet, il est possible de créer une sous-

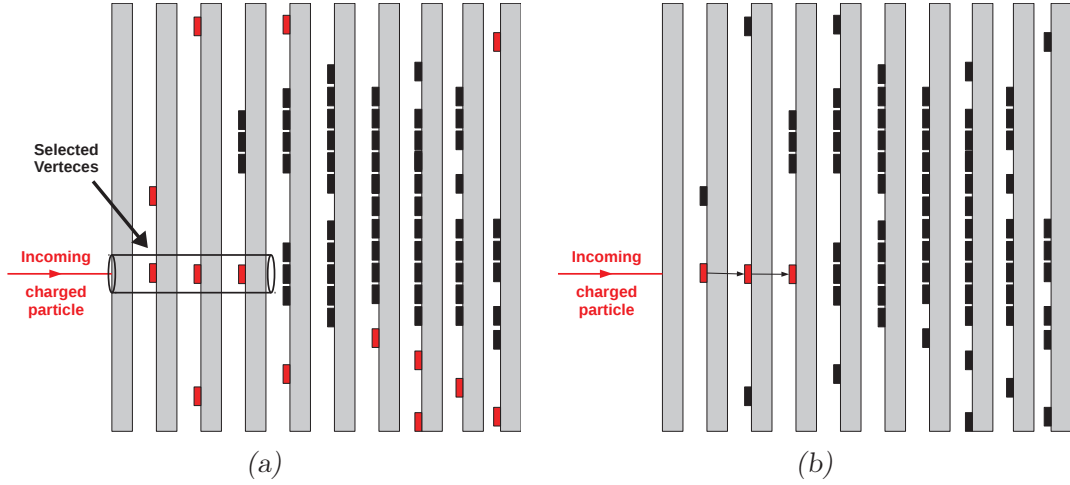


FIGURE 5.3 – (a) Zone locale de sélection des vertex étiquetés comme appartenant au segment de trace. (b) Résultat final de la connexion des vertex des segments de traces primaires.

liste de vertex particuliers et d'exécuter des algorithmes uniquement sur cette sous-liste. C'est ce que nous allons faire ici en créant une sous-liste de vertex composée uniquement de ceux étiquetés dans l'algorithme précédent et qui se trouvent sur la trajectoire des traces du trajectographe dans le calorimètre, et ce, jusqu'au début de la gerbe. La zone de sélection est montrée sur la figure 5.3(a). Deux algorithmes sont ensuite exécutés localement sur ces vertex pour i) créer des connexions entre ces vertex et ii) garder les connexions optimales pour former des arbres. Ces deux algorithmes sont décrits ci-après (*Création des connexions 1* et *Nettoyage des connexions 1*) et seront aussi utilisés dans la phase de formation des arbres principaux. Le résultat final de l'algorithme est montré sur la figure 5.3(b).

Création des connexions 1

A partir d'une sélection de vertex, des connexions sont créées entre chaque vertex et les vertex avoisinants. Pour chaque vertex, on cherche d'autres vertex dans les N_{pl} pseudo-plans suivants, à une distance maximale Δ_{max} , puis on crée une connexion entre eux. Un résultat possible de cet algorithme exécuté sur tous les vertex est illustré sur la figure 5.4(a).

Nettoyage des connexions 1

Cet algorithme suppose que des connexions ont été créées auparavant, par exemple, après l'algorithme de *création des connexions 1*. Son but est de nettoyer les connexions pour former des arbres. Pour cela, on définit la direction de référence \vec{C}_{ref} d'un vertex comme :

$$\vec{C}_{ref} = w_{bck} \cdot \sum_{\sigma} \sum_b \vec{c}_{b,\sigma} - w_{fwd} \cdot \sum_{\delta} \sum_f \vec{c}_{f,\delta} \quad (5.1)$$

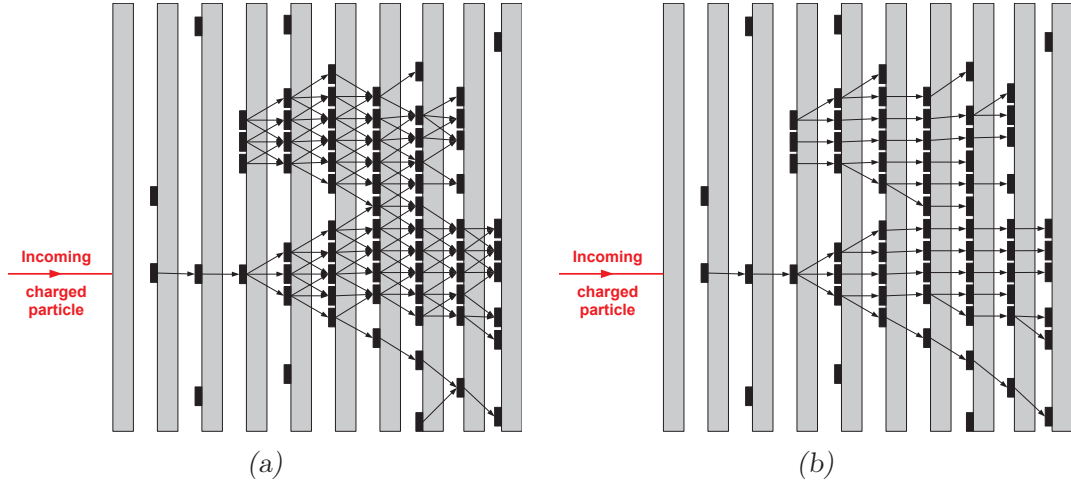


FIGURE 5.4 – (a) Résultat possible après l'exécution de l'algorithme de création des connexions 1. (b) Résultat possible après l'exécution de l'algorithme de nettoyage des connexions 1.

avec :

- w_{bck} (resp. w_{fwd}) un poids global, positif, assigné aux connexions vers l'arrière (resp. vers l'avant),
- $\vec{c}_{b,\sigma}$ (resp. $\vec{c}_{f,\delta}$), la direction d'un connecteur vers l'arrière (resp. vers l'avant), à une profondeur σ (resp. δ) du vertex considéré.

Plus simplement, ce vecteur de référence est une somme pondérée des directions des connexions liées au vertex considéré. Afin de construire une structure en arbre, chaque vertex doit garder au plus une seule connexion vers l'arrière. Ce vecteur de référence indique la direction vers l'arrière la plus probable pour garder cette unique direction. Afin de décider quelle connexion doit être gardée, on définit le paramètre d'ordre κ pour chaque connexion vers l'arrière comme :

$$\kappa = \left(\frac{\theta}{\pi}\right)^{p_\theta} \cdot \left(\frac{\Delta}{\Delta_{max}}\right)^{p_\Delta} \quad (5.2)$$

avec :

- θ l'angle entre la direction de référence du vertex considéré et la direction de la connexion vers l'arrière,
- Δ la distance entre les deux vertex connectés,
- Δ_{max} la distance maximale de connexion (coupure),
- p_θ (resp. p_Δ) un paramètre de puissance pour l'angle normalisé (resp. la distance normalisée).

Le paramètre κ quantifie l'alignement avec le vecteur de référence et est défini sur $[0, 1]$. Plus ce paramètre sera petit, plus le connecteur considéré sera aligné

avec le vecteur de référence. Les paramètres p_θ et p_Δ permettent de pondérer l'angle ou la distance de façon plus ou moins important et de façon non-linéaire. L'algorithme calcule la direction de référence \vec{C}_{ref} pour chaque vertex ainsi que le paramètre d'ordre κ pour chacune de ses connexions vers l'arrière. La connexion avec le paramètre d'ordre κ le plus petit est choisie pour former l'arbre. La suppression des autres connexions vers l'arrière est effectuée à la fin de l'algorithme, de sorte à ce que tous les connecteurs contribuent à l'évaluation de la direction de référence de chaque vertex.

Au terme de l'algorithme, différentes structures en arbre sont formées. Les figures 5.4(a) et 5.4(b) montrent un exemple de résultat possible de l'exécution de l'algorithme de *création des connexions 1* et de l'algorithme de *nettoyage des connexions 1*.

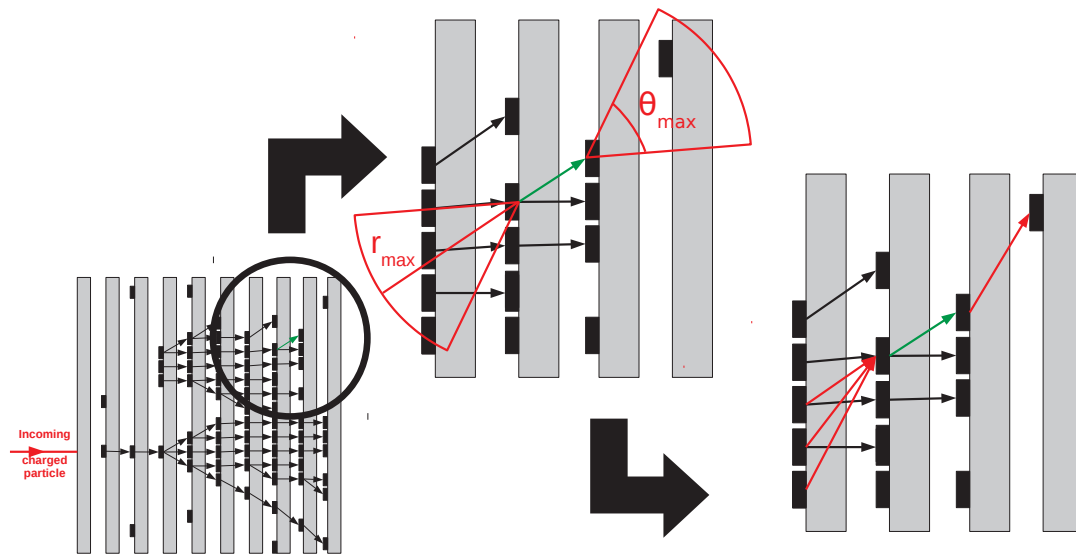


FIGURE 5.5 – Exemple d'application de l'algorithme de création de connexions 2. La connexion considérée est celle en vert. Deux angles solides vers l'avant et vers l'arrière sont utilisés pour créer de nouvelles connexions. Ces dernières sont schématisées en rouge sur la dernière étape à droite.

Création des connexions 2

Cet algorithme suppose que différentes structures en arbre ont déjà été créées en amont. Son but est de créer de nouvelles connexions à partir de celles déjà existantes et de favoriser un alignement entre les connexions au sein des gerbes. Pour chaque connexion existante, d'autres connexions sont créées en considérant les vertex de part et d'autre du connecteur. Vers l'avant comme vers l'arrière, on cherche d'autres vertex dans la direction du connecteur, dans un angle solide d'ouverture angulaire θ_{max} et de rayon maximum r_{max} , puis on crée de nouvelles connexions. La figure 5.5 illustre à partir d'un connecteur en particulier (en vert), la création de nouvelles connexions (en rouge sur le dernier schéma à droite).

Nettoyage des connexions 2

Cette deuxième version du nettoyage des connexions est assez similaire à la première. Sa seule différence réside dans l'ordonnancement du nettoyage. Dans la première version, il n'y avait aucun ordonnancement et les connexions à supprimer étaient retirées en fin d'algorithme. Ici, le nettoyage s'effectuera pseudo-plan par pseudo-plan en partant du plus grand. Une fois les connexions d'un pseudo-plan nettoyées, les connexions à supprimer sont retirées avant de passer au pseudo-plan suivant. En procédant ainsi, le calcul de la direction de référence prendra en compte uniquement des connecteurs "propres" vers l'avant. En combinant cet effet avec un ajustement des paramètres de l'équation 5.1 avec $w_{bck} < w_{fwd}$, $\sigma = 1$ et $\delta > 1$, il sera possible d'accentuer l'alignement avec les connexions "propres" vers l'avant. Le résultat final de cet algorithme est le même que la première version : un jeu de structure en arbre.

Création des clusters

À partir des structures en arbre produites par une série d'algorithmes de création de connexions et de nettoyage de connexions, chaque vertex racine est identifié et un cluster est créé à partir d'une seule structure en arbre en suivant récursivement les vertex jusqu'aux vertex feuilles.

Association entre traces et clusters

L'association entre les traces et les clusters est effectuée en utilisant l'information sur la direction et l'impulsion de la trace ainsi que l'énergie du cluster. La définition de la compatibilité $E - p$ définie dans la section 2.3 pour l'algorithme PandoraPFA sera utilisée dans cette algorithme.

Deux cas sont considérés lors de cette association :

- la particule a interagi en amont du calorimètre ou dans le premier absorbeur. Dans ce cas, le nombre de vertex racines se trouvant dans les N_{pl} premiers pseudo-plans et à une distance maximale $\Delta_{tc,1}$ de la projection de la trace sur la face avant du calorimètre dépasse une certaine valeur N_v . Ces vertex racines sont alors triés par distance croissante à cette projection. Les clusters associés à ces vertex racines sont progressivement associés à la trace jusqu'à ce que la compatibilité $E - p$ soit optimale. Ces clusters sont finalement fusionnés ensemble dans un nouveau cluster qui sera associé à la trace. Un exemple est illustré dans le cas ① de la figure 5.6.
- la particule a continué son trajet dans le calorimètre et a laissé un segment de trace primaire. Dans ce cas le nombre de vertex racine ne dépasse pas N_v dans les N_{pl} premiers pseudo-plans à une distance maximale $\Delta_{tc,1}$ (condition inverse du premier cas). Sachant qu'un seul segment de trace primaire est attendue, une seconde coupure sur la distance maximale $\Delta_{tc,2}$ ($\Delta_{tc,2} < \Delta_{tc,1}$) est appliquée afin de réduire la confusion pour de faibles distances de séparation entre des particules proches. Les procédures d'association et de fusion appliquées sont les mêmes que pour le premier cas.

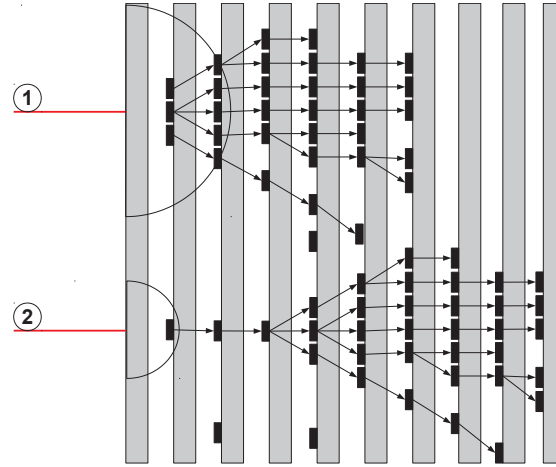


FIGURE 5.6 – Exemple d'association entre clusters et traces. Deux cas sont considérés : 1) les particules avec pré-interactions et 2) les particules interagissant dans le calorimètre.

Un exemple est illustré dans le cas (2) sur la figure 5.6.

Fusion des arbres neutres

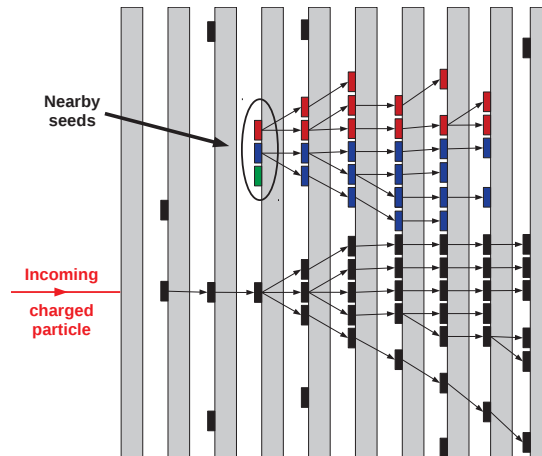


FIGURE 5.7 – Exemple d'une interaction d'une particule neutre donnant naissance à plusieurs arbres dont les vertex racines sont proches (arbres vert, bleu et rouge). Un algorithme identifie ce type de topologie et assure la fusion de ces arbres au sein d'un même cluster.

Cet algorithme est dédié aux interactions des particules neutres pour lesquelles le premier pseudo-plan d'interaction contient plusieurs vertex racines.

Un exemple est illustré sur la figure 5.7. On peut y voir trois arbres (rouge, vert et bleu) dont les vertex racines sont proches et devraient en réalité appartenir à un seul cluster. L'algorithme identifie les vertex racines dans les mêmes pseudo-plans et fusionne les arbres au sein d'un même cluster si la distance les séparant est inférieure à $\Delta_{pl,v}$.

Association d'arbres pointant

Dans cet algorithme, certains clusters neutres (sans association de trace) vont être fusionnés à des clusters chargés ou neutres suivant des critères topologiques et de compatibilité $E - p$. Ces clusters sont sélectionnés suivant des critères topologiques. Leur taille doit être supérieure à N_v vertex et doit s'étendre sur au moins N_{pl} pseudo-plans contigus. Tous ces clusters sont considérés comme potentiels parents mais seuls les clusters neutres sont considérés comme potentiels fils. Un ajustement linéaire de la position des vertex en 3 dimensions de chaque cluster est d'abord effectué. La droite résultante de l'ajustement définit l'axe du cluster. On définit le pseudo-plan interne pl_{inner} d'un cluster comme son pseudo-plan le plus en amont. Les potentiels clusters parents sont ensuite triés par pseudo-plan interne décroissant, donc du plus grand au plus petit. Pour chaque potentiel cluster fils f , on cherche un potentiel cluster parent p tel que $pl_{p,inner} < pl_{f,inner}$, par deux méthodes possibles :

1. la méthode de projection barycentrique. On définit la distance de projection barycentrique d_{proj} comme la distance minimale entre le barycentre du cluster parent et l'axe du cluster fils. On définit d'autre part l'angle $\theta_{p,f}$ entre les deux clusters (parent et fils) comme l'angle entre leurs axes respectifs. Cette méthode vise alors à trouver la distance de projection barycentrique minimale parmi tous les potentiels clusters parents telle que $d_{proj} < d_{proj,max}$ et $\theta_{p,f} < \theta_{p,f,max}$.
2. La méthode de croisement des axes : on définit la distance de croisement des axes d_{cross} comme la distance minimale entre les axes de deux clusters (d.c.a). On définit d'autre part la distance minimale d'approche entre les deux clusters d_{close} comme la plus courte distance entre un vertex du cluster parent et l'axe du cluster fils. Cette deuxième méthode vise alors à trouver le cluster parent ayant la distance de croisement d'axe d_{cross} minimale telle que $d_{cross} < d_{cross,max}$ et $d_{close} < d_{close,max}$.

A ce stade, la meilleure possibilité pour un cluster parent doit être évaluée. Plusieurs choix sont possibles en fonction du résultat des deux méthodes précédentes. i) Les deux méthodes n'ont pas trouvé de cluster parent possible. Dans ce cas, aucune action particulière n'est effectuée. ii) Une des deux méthodes a trouvé un cluster parent. Le cluster meilleur parent est alors désigné par cette méthode. iii) Les deux méthodes ont trouvé deux clusters parent différents. La distance barycentrique la plus courte entre le cluster fils et les deux clusters parents permettra de prendre la décision finale. Si un cluster parent a été désigné, on distingue deux cas :

- le cluster parent est un cluster neutre. La fusion est alors immédiatement opérée.

- le cluster parent est un cluster chargé. La fusion des deux clusters est opérée uniquement si la compatibilité $E - p$ après fusion s'améliore.

Fusion des petits arbres neutres

Conçu pour être exécuté en bout de chaîne, cet algorithme suppose que la partie principale des gerbes a été reconstruite en amont. Seulement de petites structures en arbres et les vertex isolés qui entourent les gerbes n'ont pas encore été associés. La taille des arbres est ici le critère essentiellement utilisé pour identifier ces structures. Si leurs taille n'excède pas N_v vertex alors ces structures sont fusionnées dans le cluster le plus proche (ayant une taille $N > N_v$). La distance entre la petite structure et le cluster est calculée comme étant la différence de distance cartésienne entre le barycentre des deux objets.

Création des particules reconstruites

La création des particules reconstruites est la dernière étape de l'algorithme. Cet algorithme est le même que celui utilisé par PandoraPFA à la fin de la reconstruction pour la création de PFOs (voir section 2.3). La figure 5.8 montre un résultat final possible de cette implémentation de l'algorithme ArborPFA.

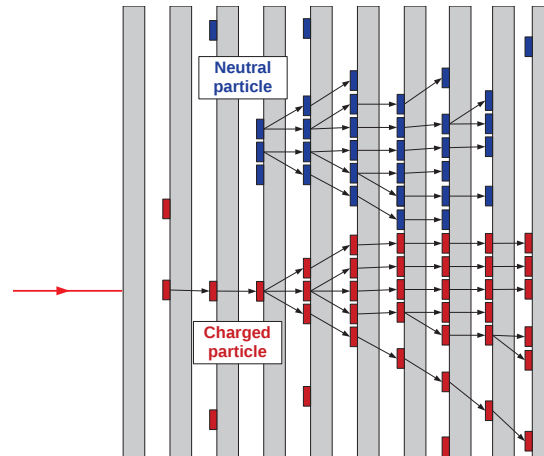


FIGURE 5.8 – Exemple de reconstruction finale d'une particule neutre (en bleu) et d'une particule chargée (en rouge) par l'algorithme ArborPFA dédié au SDHCAL.

2 Etude des hadrons isolés

2.1 Préparation des événements

Afin d'émuler correctement un hadron chargé dans le programme de reconstruction et sachant qu'aucun trajectographe n'a été installé lors du test sur faisceau, une trace est générée en amont du calorimètre. La topologie de la gerbe hadronique dans le calorimètre est utilisée pour trouver le point d'entrée. Le barycentre global des positions des hits de la gerbe hadronique est calculé dans les directions \vec{e}_x et \vec{e}_y (directions transverses au faisceau). Un nouveau barycentre est déterminé en utilisant uniquement les 4 premiers plans dans une région de 8×8 cellules dans le plan transverse, autour du premier barycentre. Ce dernier barycentre définit le point d'entrée de la gerbe hadronique. À partir de ce point, une trace est générée le long de la ligne de faisceau (dans la direction \vec{e}_z) avec l'impulsion égale à celle du faisceau.

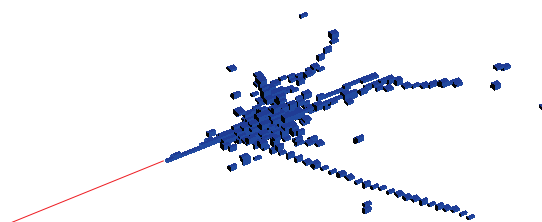


FIGURE 5.9 – Gerbe hadronique d'un pion chargé de 50 GeV dans le prototype du SDHCAL enregistré sur la H6 du SPS au CERN.

La trace et les hits du calorimètre sont ensuite chargés dans l'environnement PandoraSDK avec la géométrie du SDHCAL. Les algorithmes de ArborPFA sont ensuite exécutés. Un pion chargé de 50 GeV après le chargement dans l'environnement PandoraSDK est montré sur la figure 5.9.

2.2 Analyse des performances

On définit l'efficacité ϵ_s comme la fraction du nombre de hits de la particule chargée reconstruite par le programme ArborPFA par rapport au nombre total de hits en entrée. La figure 5.10(a) montre l'efficacité moyenne des pions isolés en fonction de l'énergie du faisceau. Cette efficacité est supérieure à 95% sur toute la gamme d'énergie, aussi bien pour la simulation que pour les données de test sur faisceau.

La figure 5.10(b) montre le nombre moyen de hits assignés au PFO chargé après reconstruction. Nous avons déjà remarqué un désaccord entre les données et la simulation sur le nombre de hits moyen au chapitre 3. Ce désaccord s'observe ici aussi à partir de 50 GeV, avec un nombre moyen de hits dans le PFO chargé plus faible dans la simulation par rapport au données.

La figure 5.11 montre le nombre moyen de PFOs reconstruits. Ce nombre moyen reste très proche de 1 PFO à 10 GeV et monte jusqu'à 1.35 PFOs à 80 GeV pour les données de test sur faisceau. Pour la simulation, cette augmentation

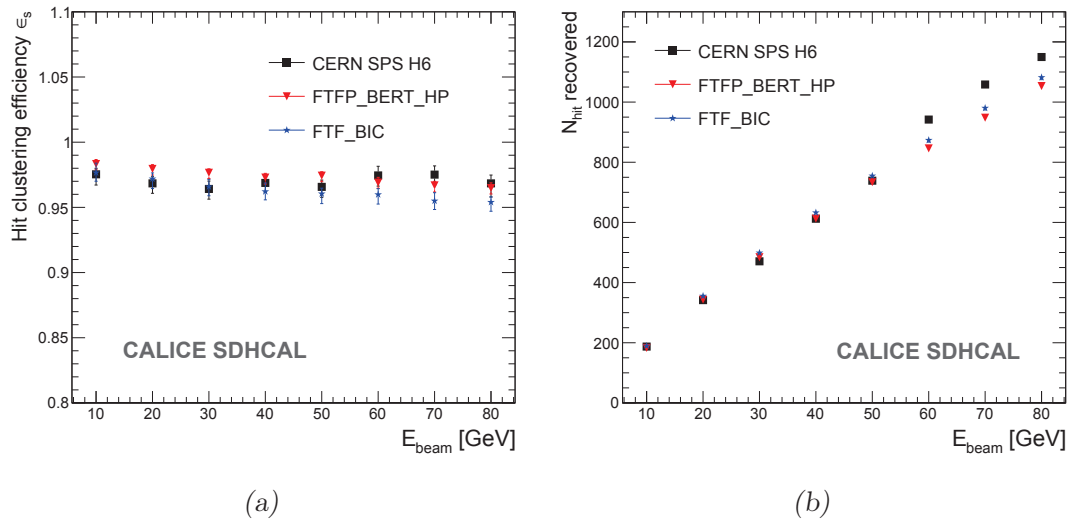


FIGURE 5.10 – (a) Efficacité moyenne des hadrons isolés après reconstruction avec ArborPFA pour des données issues du test sur faisceau et de la simulation *Geant4* pour plusieurs listes de physique. (b) Nombre moyen de hits du PFO chargé reconstruit par l’algorithme ArborPFA pour des données issues du test sur faisceau et de la simulation *Geant4* pour les listes de physique FTFP_BERT_HP et FTF_BIC.

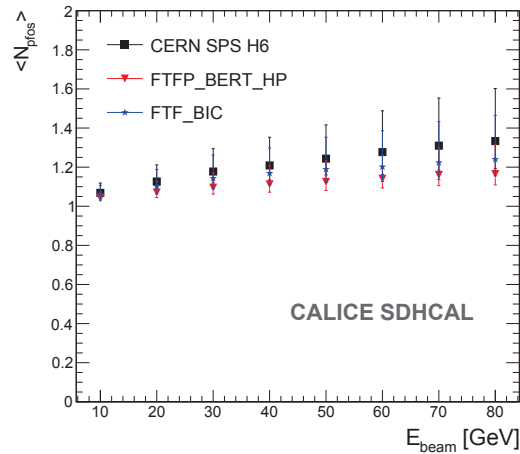


FIGURE 5.11 – Nombre moyen de PFOs reconstruits par l’algorithme ArborPFA pour des données issues du test sur faisceau et de la simulation *Geant4* pour plusieurs listes de physique.

est moins prononcée. À 80 *GeV*, le nombre de PFOs moyen monte à 1.25 PFOs pour la liste FTF_BIC et à 1.15 pour la liste FTFP_BERT_HP. Les figures 5.12(a) et 5.12(b) montrent les distributions du nombre de PFOs reconstruits à 20 *GeV* et 70 *GeV*. La différence est plus visible à haute énergie, où la fraction des événements reconstruisant 2 PFOs est plus importante pour les données de test sur faisceau.

La figure 5.13(b) montre la distribution du nombre de hits de chacun des PFOs reconstruits. On voit que ces PFOs additionnels produits à haute énergie sont en réalité des petits fragments qui n'ont pas été associés au PFO chargé principal. On en déduit que la probabilité de reconstruire un deuxième PFO ayant un faible nombre de hits dans les données de test sur faisceau est plus importante.

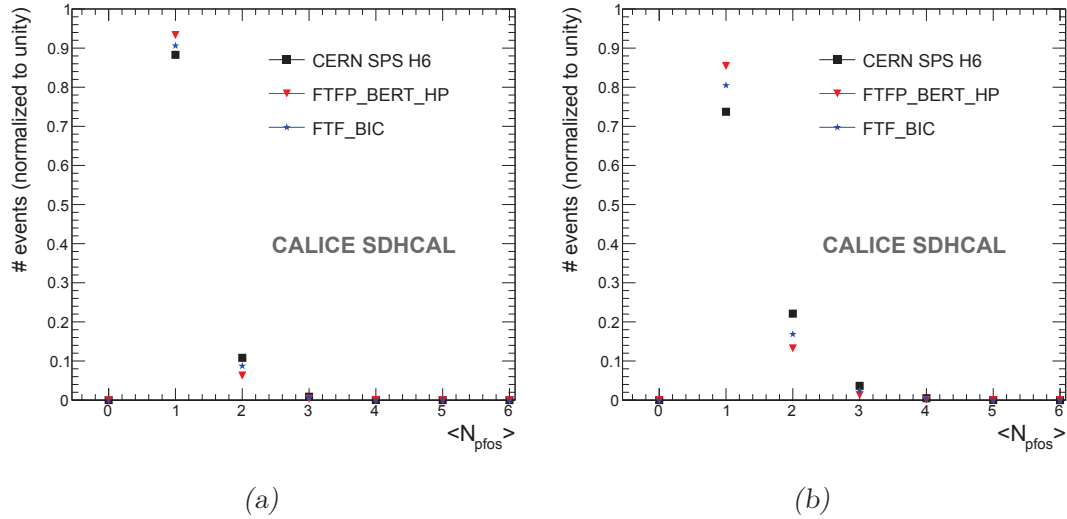


FIGURE 5.12 – Distribution du nombre de PFOs pour des pions chargés (a) à 20 GeV et (b) à 70 GeV pour des données issues du test sur faisceau et de la simulation *Geant4* pour plusieurs listes de physique.

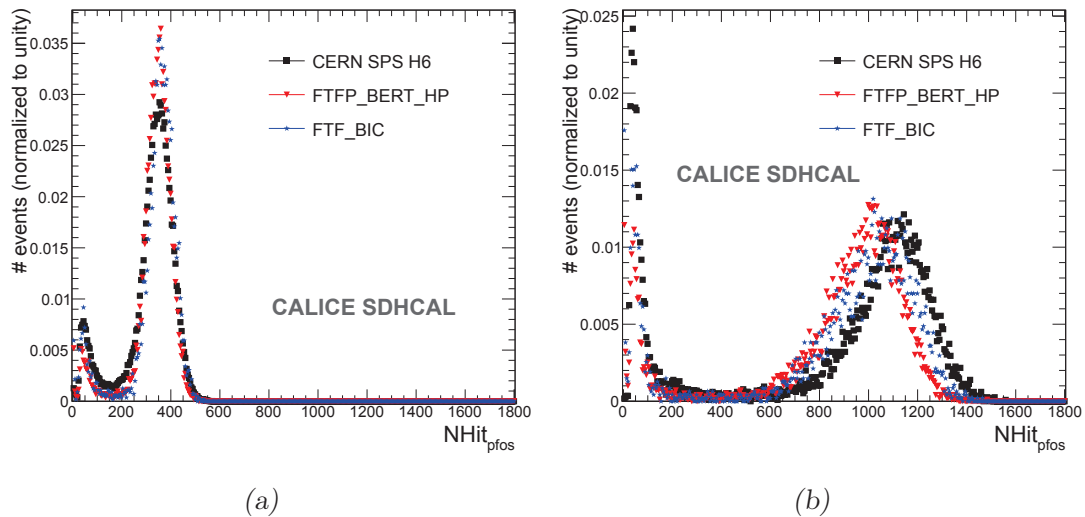


FIGURE 5.13 – Distribution du nombre de hits des PFOs reconstruits (a) à 20 GeV et (b) à 70 GeV.

Les figures 5.14(a), 5.14(b) et 5.16(a) montre l'énergie reconstruite du PFO chargé reconstruit pour différents types de données. Les figures 5.15(a), 5.15(b) et

5.16(b) montrent quant à elles la résolution en énergie du PFO chargé reconstruit.

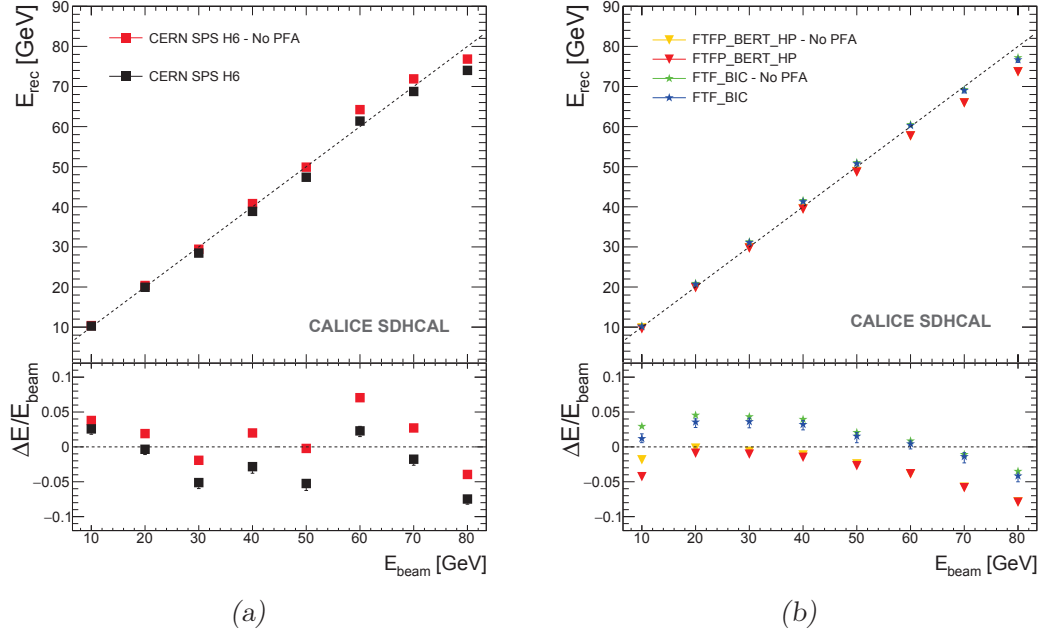


FIGURE 5.14 – Énergie reconstruite avant reconstruction (No PFA) et après reconstruction en utilisant les hits du PFO chargé pour (a) les données issues du test sur faisceau et (b) les données issues de la simulation *Geant4* pour différentes simulations physiques (*FTFP_BERT_HP* et *FTF_BIC*).

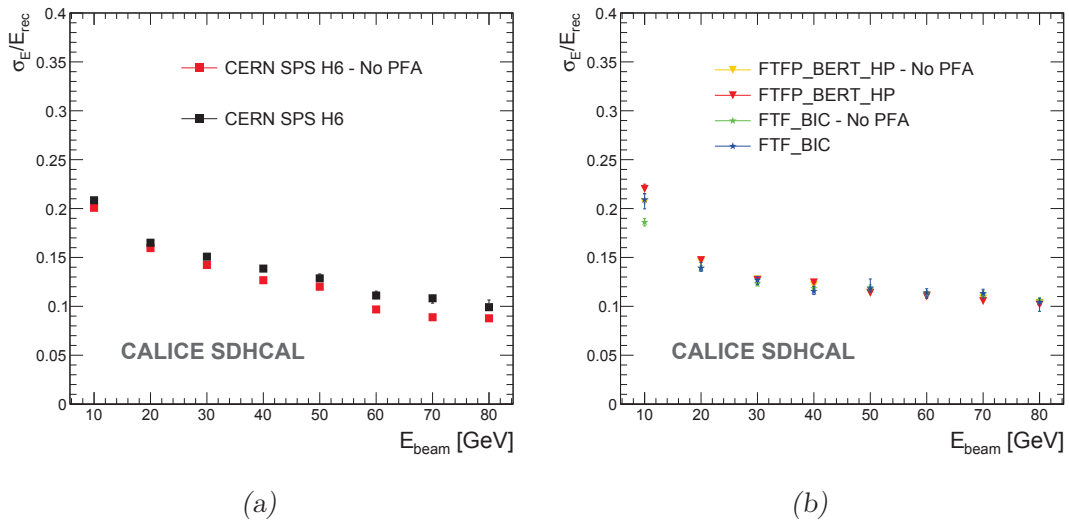


FIGURE 5.15 – Résolution en énergie avant reconstruction (No PFA) et après reconstruction en utilisant les hits du PFO chargé pour (a) les données issues du test sur faisceau et (b) les données issues de la simulation *Geant4* pour différentes simulations physiques (*FTFP_BERT_HP* et *FTF_BIC*).

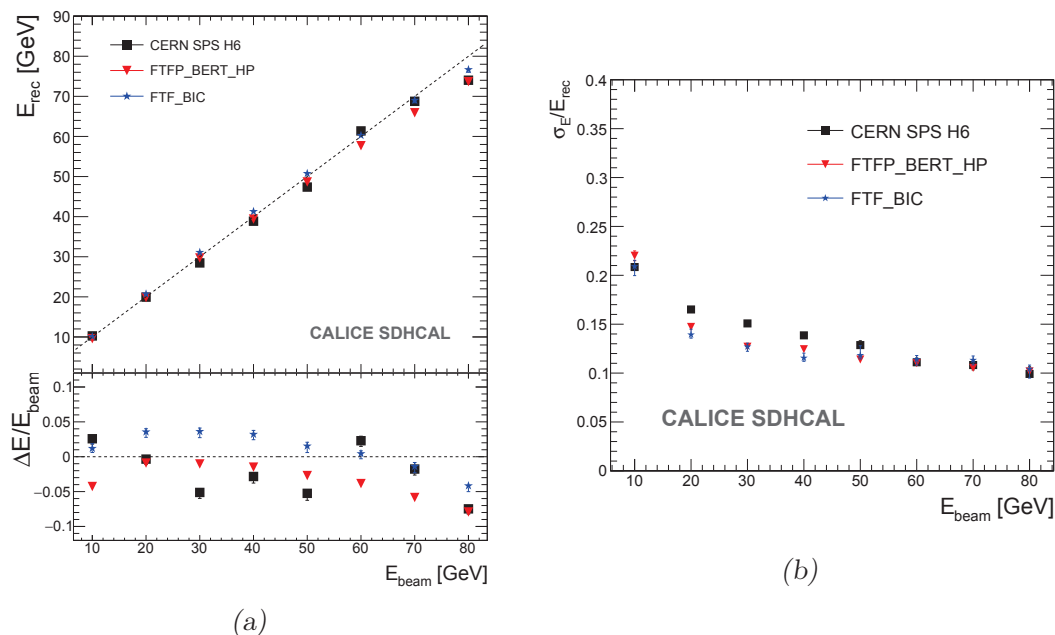


FIGURE 5.16 – (a) Énergie reconstruite et (b) résolution en énergie après reconstruction en utilisant les hits du PFO chargé pour les données issues du test sur faisceau et les données issues de la simulation *Geant4* pour différentes simulations physiques (FTFP_BERT_HP et FTF_BIC).

Chacun des points d'énergie reconstruite montrés sur ces figures est extrait de la manière suivante :

- un ajustement d'une fonction gaussienne est effectué sur la distribution d'énergie reconstruite. On obtient alors un premier jeu de paramètres $\mu_{E,1}$ et $\sigma_{E,1}$ pour l'énergie moyenne et la largeur de la distribution en énergie.
- un second ajustement est effectué sur l'intervalle $[\mu_{E,1}-1.5\cdot\sigma_{E,1}, \mu_{E,1}+1.5\cdot\sigma_{E,1}]$. De ce dernier on extrait un second jeu de paramètres finaux pour l'énergie moyenne reconstruite μ_E et σ_E . La déviation à la linéarité en énergie est alors calculée de la façon suivante :

$$\Delta E/E_{beam} = (\mu_E - E_{beam})/E_{beam} \quad (5.3)$$

L'efficacité montrée sur la figure 5.10(a) montre qu'en effet un faible pourcentage de hits manquent dans la particule reconstruite finale. Il est donc normal d'attendre une légère diminution de l'énergie reconstruite du PFO chargé comparé à l'énergie estimée lorsque l'on inclut tous les hits dans le calorimètre. Cette diminution n'affecte au final que très peu la résolution en énergie. Néanmoins, la déviation à la linéarité de l'énergie se dégrade légèrement avec un maximum de déviation à 7.5% à 80 GeV pour les données du test sur faisceau. Pour la simulation, on observe une légère différence au niveau de cette linéarité mais à basse énergie avec 4% de déviation à 10 GeV pour la simulation FTFP_BERT_HP. De façon globale, la déviation à la linéarité reste toujours inférieure à 8% sur toute la gamme en énergie.

Les incertitudes montrées sur chacune de ces figures comprennent les erreurs statistiques et les erreurs systématiques évaluées sur une variation des paramètres de l'algorithme. Cette estimation sera plus détaillée dans la section 4.

3 Etude de séparation de deux hadrons proches

La capacité d'un algorithme de suivi de particules à séparer des gerbes proches dans les calorimètres est un point clé dans la reconstruction des événements dans des détecteurs tels que ceux de l'ILC. Cette source de confusion vient principalement de la proximité d'un hadron neutre avec une autre particule chargée. Cette deuxième étude vise à étudier la capacité de l'algorithme ArborPFA à séparer un hadron neutre proche d'un hadron chargé.

Générer deux hadrons proches et différencier les dépôts en énergie dans la simulation est bien entendu possible. Cependant, il sera impossible d'obtenir le même type de données dans le cadre d'une prise de données réelles sur le prototype du SDHCAL. D'une part, d'un point de vue technique, on ne dispose pas de deux faisceaux de particules ni d'une intensité de faisceau suffisante pour obtenir deux particules dans un même événement. D'autre part, même si le dispositif le permettait, les dépôts en énergie laissés par les deux particules enregistrées ne seraient pas différenciables et il serait alors impossible d'estimer les performances de séparation de l'algorithme.

Afin de traiter de manière équivalente les données de tests sur faisceau et de simulations, deux hadrons issus du jeu de données précédemment traité sont superposés au sein d'un même événement. L'algorithme ArborPFA est ensuite exécuté sur ce nouvel événement. Un jeu de variables est ensuite extrait pour évaluer les performances de l'algorithme quant à la séparation des gerbes hadroniques.

3.1 Superposition des événements

Deux événements issus du jeu de données traité lors de l'analyse des hadrons isolés sont superposés au sein d'un même événement. Ce traitement est effectué aussi bien pour les données de tests sur faisceau que pour les simulations afin de tester de manière équivalente l'algorithme de reconstruction. Le jeu de données des tests sur faisceau ne comporte que des pions chargés. C'est pourquoi afin de superposer un hadron neutre avec un hadron chargé, le hadron neutre sera émulé par un hadron chargé dont les hits de la trace primaire dans le calorimètre auront été supprimés. Les énergies choisies pour les deux hadrons sont de 10 GeV pour la particule neutre et de 10 GeV et 30 GeV pour la particule chargée. Ces énergies correspondent aux énergies typiques des particules au sein d'un jet à l'ILC [33]. Plusieurs distances de séparation entre les deux hadrons ont été choisies, allant de 5 cm à 30 cm , par pas de 5 cm . Une distance de 30 cm entre les deux particules correspond à deux gerbes correctement séparées et identifiables du point de vue d'un algorithme de suivi de particules. Les distances les plus faibles, c'est-à-dire 5 cm ou 10 cm , correspondent à une superposition plus ou moins importante des gerbes hadroniques, leurs rayons

moyens dans la simulation étant d'environ 6 cm [20] et donc plus de difficultés à les séparer du point de vue de l'algorithme.

L'algorithme de superposition des gerbes se déroule de la façon suivante :

1. Le segment de trace primaire des deux pions est identifié à l'aide la transformé de Hough [17]. Cela permet d'identifier le point d'entrée des gerbes hadroniques dans le calorimètre. Si pour un des deux hadrons, aucune trace primaire n'est identifiée, les deux particules ne sont pas superposées.
2. Les hits appartenant au segment de trace primaire du pion de 10 GeV sont supprimés de l'événement, permettant ainsi d'émuler un hadron neutre.
3. Les deux gerbes sont re-centrées au centre du calorimètre dans les directions \vec{e}_x et \vec{e}_y (directions transverses). Aucun ajustement n'est fait dans la direction \vec{e}_z (ligne du faisceau). Ce recentrage est effectué par rapport à leurs point d'entrée dans le calorimètre.
4. Les deux gerbes sont déplacées dans la direction \vec{e}_x d'une distance $-d/2$ pour le hadron neutre et $+d/2$ pour le pion chargé, avec d la distance au centre du calorimètre en *cm*.

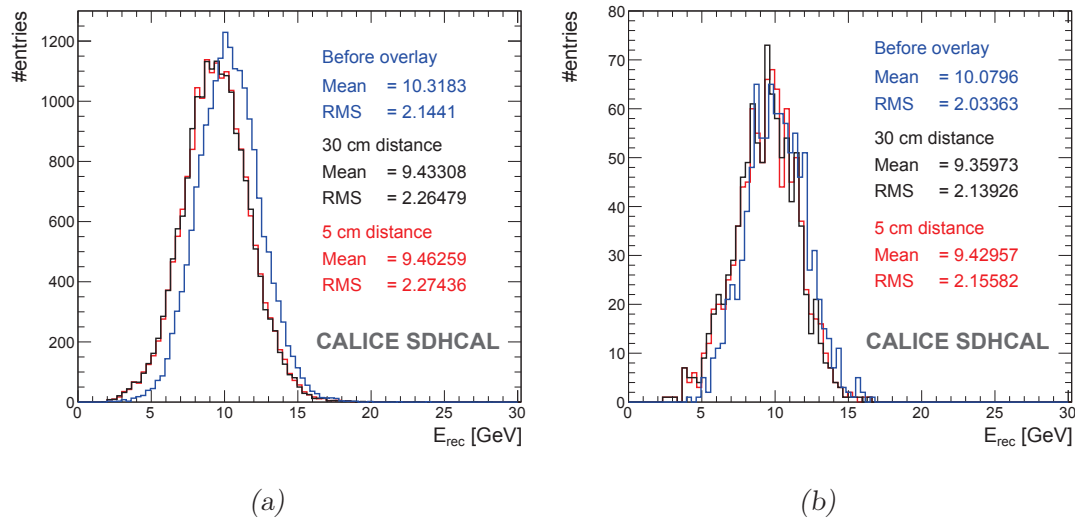


FIGURE 5.17 – Distribution de l'énergie reconstruite du hadron neutre de 10 GeV avant superposition (en bleu) et après superposition avec un hadron chargé de 30 GeV pour une distance de séparation de 5 cm (en rouge) et 30 cm (en noir) pour (a) les données du test sur faisceau et (b) la simulation *Geant4* avec la liste de physique *FTF_BIC*.

5. Les deux gerbes sont finalement superposées au sein du même événement. Lors de cette étape, un problème se pose. En superposant les hits un par un dans le calorimètre, certaines cellules peuvent se superposer. Avec un calorimètre digital, les hits seraient superposés sans problème particulier. Pour un calorimètre analogique, l'énergie du nouveau hit serait simplement la somme des énergies des deux hits. Dans notre cas, la lecture semi-digitale pose un problème, celui du calcul du nouveau seuil. En effet, dans les données de test

sur faisceau, l'information sur la charge déposée dans chaque cellule n'est pas disponible. La seule information dont nous disposons est la valeur des seuils des deux hits. L'approximation la plus intuitive est d'assigner le seuil le plus haut des deux hits au hit superposé. Les figures 5.17(a) et 5.17(b) montrent l'énergie reconstruite du hadron neutre de 10 GeV avant (en bleu) et après superposition avec un hadron chargé de 30 GeV, séparés par une distance de 5 cm (en rouge) et 30 cm (en noir). Cette configuration représente celle où le plus de hits superposés est attendu. Le cas de la séparation de 30 cm est pris comme référence pour comparaison, sachant que la part de hits superposés est négligeable (voir figures 5.18(a) et 5.19(a)). On voit que les distributions en énergie sont très peu décalées quand on passe de 30 cm de séparation à 5 cm, ce qui valide l'approximation du seuil le plus haut comme choix du nouveau seuil. Cette approximation n'aura pas d'impact sur la suite de l'étude. Les figures 5.18(b) et 5.19(b) montrent le nombre de hits superposés et le pourcentage du nombre de hits superposés pour une distance de 5 cm. On voit que le taux de hits superposés est finalement bas, même à faible distance. En revanche, la suppression des hits du segment de trace primaire du hadron neutre émulé a plus d'impact sur l'énergie reconstruite (figures 5.17(a) et 5.17(b) en bleu). L'énergie moyenne perdue par cette suppression est estimée à 0.9 GeV pour les données de test sur faisceau et de 0.7 GeV pour la simulation.

6. Les hits sont étiquetés par rapport à la gerbe à laquelle ils appartenaient, 1 ou 2. Les hits superposés sont étiquetés 3. Cette information permettra de faire des comparaisons avant et après reconstruction pour estimer les performances de l'algorithme.
7. Un nouvel événement est créé, contenant les deux gerbes superposées. Le point d'entrée de chacune des gerbes après leur décalage est stocké en information complémentaire dans l'événement.

3.2 Analyse des performances

La figure 5.20 montre le nombre moyen de PFOs après reconstruction en fonction de la distance entre les deux particules. Pour des grandes distances de séparation, ce nombre moyen tend vers la somme des moyennes du nombre de PFOs pour deux hadrons isolés. Ce nombre est stable en fonction de la distance jusqu'à 15 cm, puis descend jusqu'à environ 1.8 PFOs à 5 cm. Ceci est dû à une superposition des gerbes plus importante et donc des difficultés pour l'algorithme à discerner les différentes contributions de chacune des gerbes.

Pour quantifier la séparation des gerbes, on définit l'efficacité ϵ et la pureté ρ d'une des deux gerbes comme :

$$\epsilon = \frac{N_{good}}{N_{init,tot}} \quad (5.4)$$

$$\rho = \frac{N_{good}}{N_{rec,tot}} \quad (5.5)$$

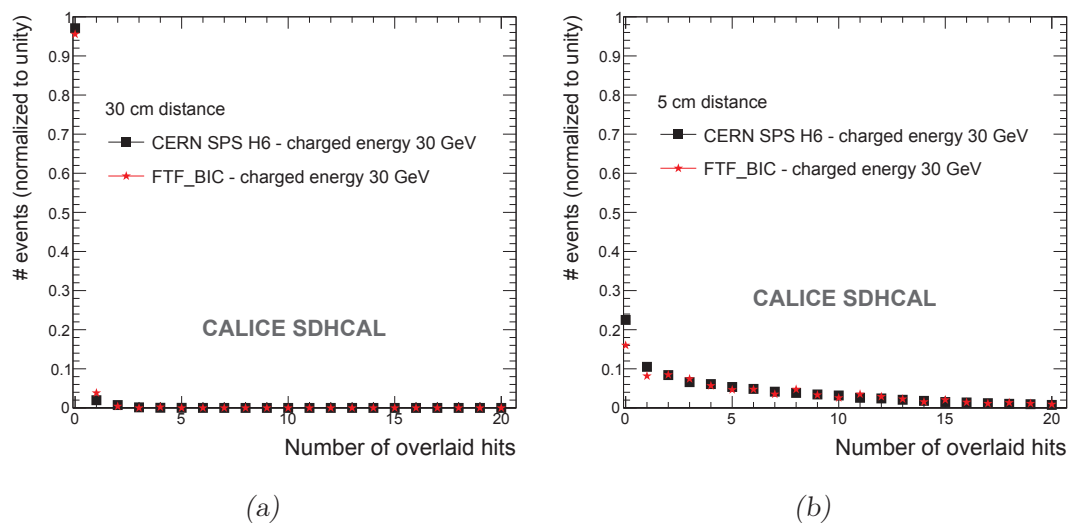


FIGURE 5.18 – Distribution du nombre de hits superposés pour un hadron neutre de 10 GeV et un hadron chargé de 30 GeV pour une distance de séparation de (a) 30 cm et (b) 5 cm.

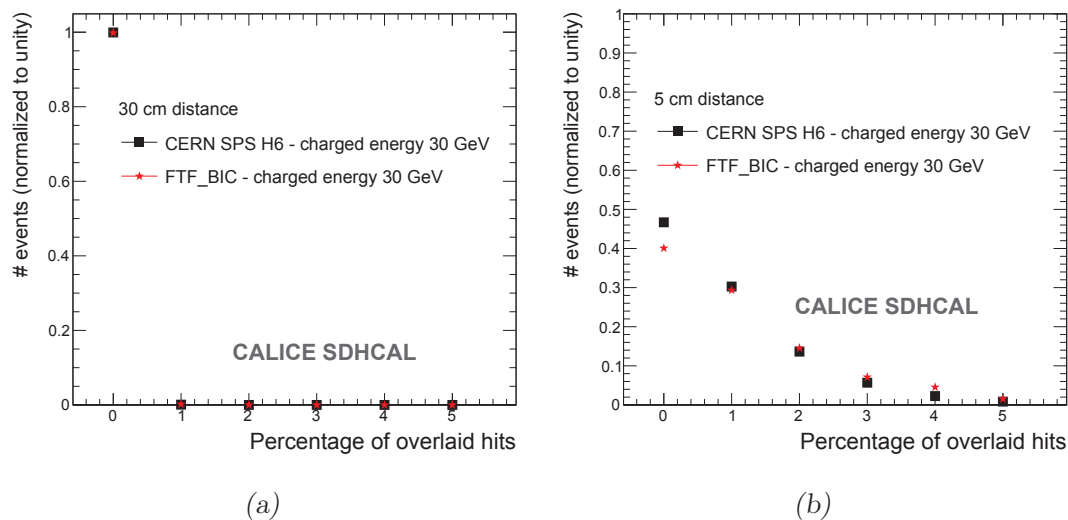


FIGURE 5.19 – Distribution du pourcentage du nombre total de hits superposés pour un hadron neutre de 10 GeV et un hadron chargé de 30 GeV pour une distance de séparation de (a) 30 cm et (b) 5 cm.

avec N_{good} le nombre de hits appartenant initialement à la particule considérée et correctement attaché après la reconstruction, $N_{ini,tot}$ le nombre total de hits de la particule avant la reconstruction, $N_{rec,tot}$ le nombre de hits de la particule reconstruite. Les figures 5.21(a) et 5.21(b) montrent respectivement l'efficacité et la pureté du hadron neutre reconstruit en fonction de la distance de séparation. De la même manière que pour le nombre moyen de PFOs, à faible distance, l'efficacité et la pureté diminuent avec la distance de séparation. À faibles distances, les hits

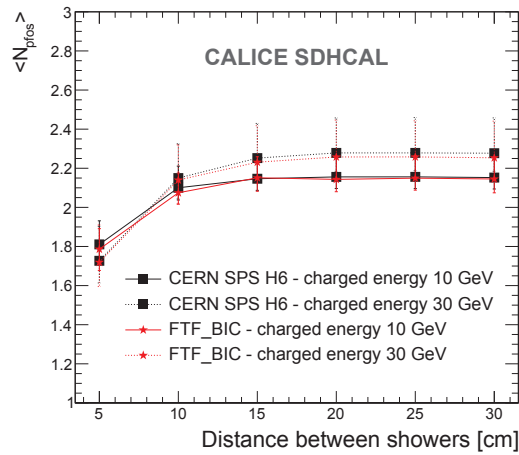


FIGURE 5.20 – Nombre moyen de PFOs reconstruits par l’algorithme ArborPFA pour des données superposées issues de test sur faisceau et de la simulation *Geant4* pour la liste de physique *FTF_BIC*.

d’une première gerbe sont assignés à la deuxième, ainsi la pureté et l’efficacité s’en voient impactées. À grandes distances, la pureté ne tend pas vers 100%. Ceci est dû à l’algorithme de *fusion des petits arbres neutres* qui utilise principalement un critère de distance pour fusionner les fragments, sans considérer l’énergie ou la taille des clusters. Sachant que le nombre de fragments neutres autour d’une particule isolée augmente avec son énergie, une part non-négligeable des hits de la particule chargée est assignée à la particule neutre, entraînant une diminution de sa pureté.

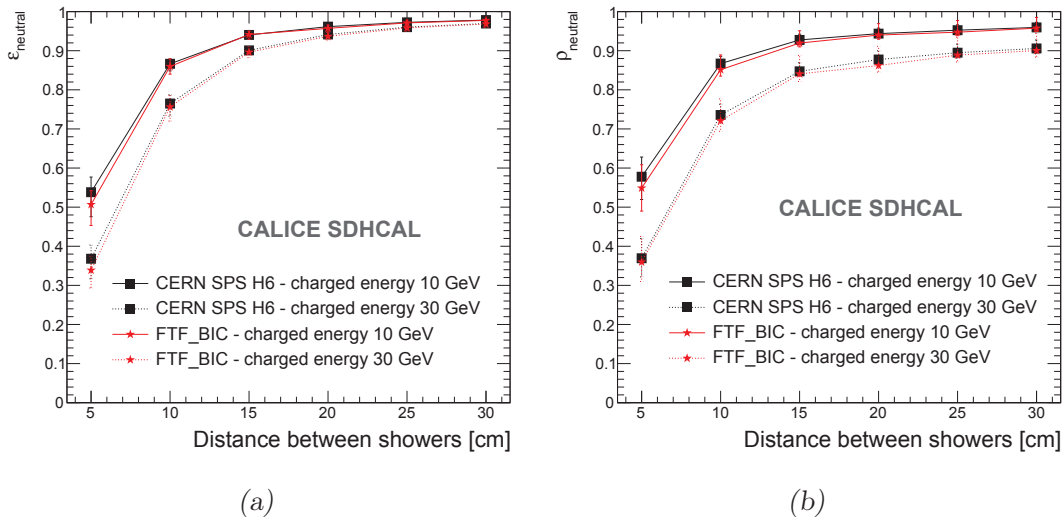


FIGURE 5.21 – (a) Efficacité et (b) pureté du hadron neutre de 10 GeV après reconstruction.

La figure 5.22(a) montre la fraction d’événements pour laquelle au moins un

hadron neutre a été reconstruit. Ici encore, ce nombre diminue avec la distance de séparation. Cette probabilité est stable et supérieure à 97% de 30 cm à 15 cm. La diminution à faibles distances dépend de l'énergie de la particule chargée. Plus cette énergie est grande, plus le nombre de hits augmente ainsi que le rayon de la gerbe et plus grande est la confusion. Il est donc normal d'observer une probabilité plus faible pour une particule chargée de 30 GeV que pour une particule chargée de 10 GeV. Cette probabilité chute à 73% pour une particule chargée de 10 GeV et 62% pour une particule chargée de 30 GeV.

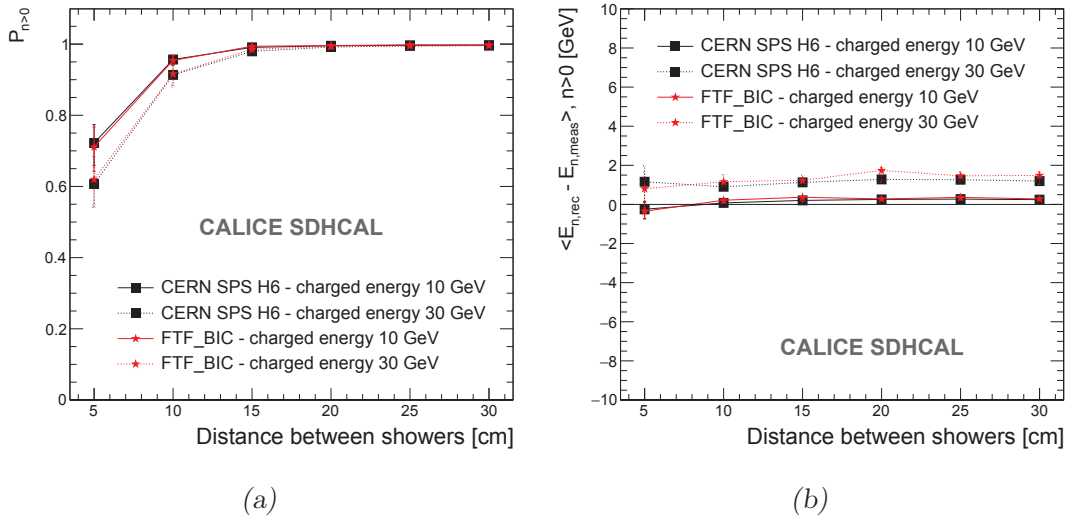


FIGURE 5.22 – (a) Probabilité de reconstruire au moins une particule neutre avec l'algorithme ArborPFA. (b) La différence moyenne entre l'énergie du hadron neutre après reconstruction et avant reconstruction pour les événements présentant au moins une particule neutre.

Les figures 5.23 et 5.22(b) montrent les performances de reconstruction en énergie du hadron neutre. La figure 5.23 montre la différence entre l'énergie neutre totale reconstruite et l'énergie estimée de la particule neutre initiale. Cette dernière estimation ne tient pas compte de l'énergie manquante due à la suppression des hits du segment de trace primaire. Pour la même raison que pour la non-convergence de la pureté à grande distance, l'énergie reconstruite moyenne neutre augmente avec l'énergie de la particule chargée. Une différence d'environ 1 GeV est observée entre le cas d'une particule chargée de 10 GeV et 30 GeV. La différence en énergie est constante jusqu'à 15 cm de séparation. Bien entendu, cette différence chute à faible distance pour atteindre environ 3 GeV. La figure 5.22(b) montre la même différence en énergie mais uniquement pour les événements où au moins une particule neutre a été reconstruite. On voit que globalement sur toute la gamme en distance de séparation, cette différence en énergie est constante ($\pm 5\%$), même à 5 cm de séparation. Cela signifie que si la particule neutre est reconstruite, dans ce cas son énergie est correctement reconstruite, montrant ainsi que la reconstruction à faibles distance a un comportement binaire : la particule neutre est soit correctement reconstruite, soit complètement fusionnée avec la particule chargée.

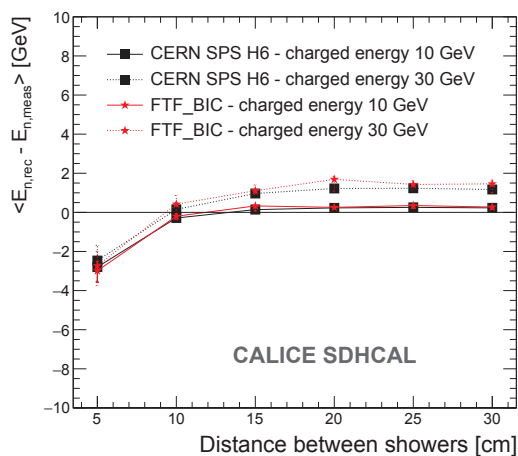


FIGURE 5.23 – Différence moyenne entre l'énergie du hadron neutre avant et après reconstruction.

4 Les incertitudes systématiques

Dans les résultats des sections 2 et 3, les incertitudes statistiques et systématiques ont été prises en compte. Ces deux sources ont été ajoutées en quadrature. Les incertitudes systématiques sont estimées en variant certains paramètres de l'algorithme jugés comme étant les plus importants, autour de leurs valeurs nominales. La table 5.1 résume le choix de ces paramètres, leurs valeurs nominales et les variations utilisées pour l'évaluation des incertitudes systématiques. Un exemple d'erreurs détaillées pour deux points en particulier est résumé dans la table 5.2.

Pour chacune des variations inférieures et supérieures des paramètres, l'algorithme complet est exécuté et un nouveau jeu de variables d'analyse est obtenu. Les différences avec les variables d'analyse évaluées aux valeurs nominales (sans variation) sont estimées et ajoutées en quadrature pour obtenir l'erreur systématique.

Paramètres	Valeur nominale	Variation inférieure	Variation supérieure
Distance de connexion 1	45 mm	40 mm	50 mm
Distance de connexion 2	65 mm	60 mm	70 mm
Angle de connexion	0.7 rad	0.6 rad	0.8 rad
Poids des connexions vers l'arrière 1	2	1	3
Poids des connexions vers l'avant 1	3	2	4
Poids des connexions vers l'arrière 2	1	0.01	2
Poids des connexions vers l'avant 2	5	4	6
Coupure sur la taille des fragments	20	15	25

TABLE 5.1 – Paramètres de l'algorithme utilisés pour l'évaluation des incertitudes systématiques.

Paramètre	Cas # 1		Cas # 2	
	Erreur inférieure	Erreur supérieure	Erreur inférieure	Erreur supérieure
Distance de connexion 1	$-4.6 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$	$5.1 \cdot 10^{-3}$	$7.9 \cdot 10^{-3}$
Distance de connexion 2	$-1.6 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$	$-3.1 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$
Angle de connexion	$3.2 \cdot 10^{-4}$	$-1.4 \cdot 10^{-4}$	$8.9 \cdot 10^{-4}$	$6.7 \cdot 10^{-4}$
Poids des connexions vers l'arrière 1	$1.5 \cdot 10^{-4}$	$1.6 \cdot 10^{-4}$	$4.4 \cdot 10^{-4}$	$-6.7 \cdot 10^{-4}$
Poids des connexions vers l'avant 1	$2.5 \cdot 10^{-4}$	$3.4 \cdot 10^{-4}$	$-8.9 \cdot 10^{-4}$	$6.7 \cdot 10^{-4}$
Poids des connexions vers l'arrière 2	$-1.3 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$	$-1.4 \cdot 10^{-3}$	$3.5 \cdot 10^{-3}$
Poids des connexions vers l'avant 2	$1.2 \cdot 10^{-4}$	$-3.0 \cdot 10^{-4}$	$-0.2 \cdot 10^{-4}$	$-1.3 \cdot 10^{-4}$
Coupure sur la taille des fragments	$-4.9 \cdot 10^{-3}$	$4.9 \cdot 10^{-3}$	$-5.5 \cdot 10^{-3}$	$5.5 \cdot 10^{-3}$
Total	0.866 ± 0.007 (stat) $\pm_{0.018}^{0.012}$ (syst)		0.764 ± 0.007 (stat) $\pm_{0.032}^{0.022}$ (syst)	

TABLE 5.2 – Exemple d'incertitudes systématiques estimées pour deux points en particulier dans l'étude de séparation des gerbes proches pour les données de test sur faisceau : efficacité du hadron neutre de 10 GeV superposé avec un hadron chargé de 10 GeV (cas # 1) et 30 GeV (cas # 2) à 10 cm de séparation (voir figure 5.21(a)).

5 Conclusion

Un algorithme appelé ArborPFA, utilisant la topologie en arbre orienté des gerbes hadroniques, a été développé. Afin de tester l'idée de cet algorithme, une première implémentation dédiée au prototype du SDHCAL a été exposée dans ce chapitre.

Les différents algorithmes ont dans un premier temps été décrits. Les performances de l'algorithme ont été extraites à partir d'échantillons de hadrons isolés dans le détecteur pour différentes énergies. L'efficacité de reconstruction s'est révélée être supérieure à 96% et l'énergie reconstruite très proche de celle estimée avant la reconstruction. Le nombre moyen de PFO reconstruits a montré une croissance linéaire de 1.1 PFOs à 10 *GeV* jusqu'à 1.35 PFOs à 80 *GeV* pour les données de test sur faisceau contre une croissance linéaire de 1.1 PFOs à 10 *GeV* jusqu'à 1.2 PFOs à 80 *GeV* pour les échantillons de simulation.

Une étude visant à séparer deux hadrons proches à différentes distances de séparation a ensuite été présentée. Deux hadrons issus de deux événements différents ont été superposés dans le même événement physique. Pour un des deux hadrons, les hits appartenant au segment de trace primaire ont été supprimés afin de simuler un hadron neutre. Cette étude a permis de mettre en évidence une très bonne séparation jusqu'à 10 cm de distance, avec une probabilité de reconstruire au moins un hadron neutre supérieure à 90%. En-deçà, le pouvoir de séparation diminue, du fait d'un contact important entre les deux gerbes hadroniques et aux confusions résultantes pour l'algorithme. L'analyse des résultats a aussi montré que lorsqu'au moins un hadron neutre était reconstruit son énergie était proche de celle avant la reconstruction ($\pm 5\%$), et ce, quelle que soit la distance de séparation entre les hadrons. Cela suggère que les algorithmes exécutés dans le cas d'une distance de séparation faible sont soit très efficaces ou inefficaces.

Chapitre 6

L'algorithme ArborPFA pour le détecteur ILD

LA première implémentation de l'algorithme ArborPFA dédiée au prototype du SDHCAL a permis d'apprécier les performances qu'il était possible d'atteindre avec ce type d'algorithme et de détecteur. Dans ce chapitre, nous allons décrire un algorithme similaire, dédié au détecteur ILD et tenant compte cette fois-ci de la présence de sous-détecteurs additionnels. Dans un premier temps, nous discuterons la calibration en énergie des hadrons isolés dans le détecteur ILD, nécessaire pour obtenir une bonne linéarité. Nous décrirons ensuite les différents algorithmes et fonctions additionnelles utilisés pour reconstruire les événements dans le détecteur ILD. Les performances de l'algorithme sur un système de di-jet uds issus d'un boson Z^0 virtuel seront finalement exposées en fin de chapitre.

Contenu

1	Calibration en énergie des hadrons isolés	110
1.1	Calibration initiale	110
1.2	Correction inter-module	113
1.3	Correction angulaire	116
2	Les fonctions et les algorithmes ArborPFA	123
2.1	La fonction de calcul du champs magnétique	123
2.2	La fonction de calcul des pseudo-plans	124
2.3	Les fonctions d'identification de particules	124
2.4	Les fonctions de correction en énergie	126
2.5	L'ordonnancement des algorithmes	127
2.6	Les algorithmes de reconstruction	132
3	Les performances physiques de l'algorithme	147
4	Conclusion	151

1 Calibration en énergie des hadrons isolés

Cette première section a pour but d'étudier la réponse en énergie des hadrons dans le détecteur ILD en fonction de leur angle d'incidence et de leur énergie. Les algorithmes de reconstruction prévus dans l'implémentation de ArborPFA pour le détecteur ILD faisant notamment usage de l'énergie des particules pour traiter de certaines situations, une bonne linéarité ainsi qu'une bonne résolution en énergie sont attendues afin de traiter au mieux ces situations sans ambiguïté et avec un minimum d'erreur.

Pour mener cette étude, nous utiliserons le logiciel de simulation Mokka 08-04 [35] avec le détecteur ILD_o2_v05 et la liste de physique hadronique QGSP_BERT. Les particules sont générées à l'aide d'un canon à particule positionné à l'origine des coordonnées et avec des énergies allant de 10 *GeV* à 80 *GeV* par pas de 10 *GeV*. Nous rappelons que pour le détecteur ILD_o2_v05, le tonneau central est divisé en 5 modules de largeurs égales. Entre chaque module du calorimètre hadronique un espace vide de 2 *mm* est présent. De plus, de part et d'autre de chaque module, une couche supplémentaire d'acier de 2 cm d'épaisseur vient fermer le module, rajoutant ainsi un budget matière non négligeable. On s'attend donc à observer une chute en énergie reconstruite pour des particules isolées traversant ces interstices. Cette observation peut aussi être faite dans la région intermédiaire entre le tonneau central et les bouchons où, en plus des espaces vides entre chaque sous-détecteur, des couches d'absorbeurs supplémentaires viennent fermer chacun des sous détecteurs.

1.1 Calibration initiale

La première étape de cette calibration en énergie consiste à extraire les constantes de calibration des différents calorimètres en générant des particules dans une région qui servira de référence par la suite. Cette région est choisie dans le tonneau central avec un angle $\phi=0$ et un angle $\theta=1.5 \text{ rad}$. Ce dernier angle est choisi légèrement inférieur à $\frac{\pi}{2}$ ($\simeq 1.5707$) afin d'éviter de rencontrer trop de matériaux sur le chemin de la particule, notamment dans l'électrode centrale dans la chambre à projection temporelle [3]. Cela permettra d'éviter une perte importante de l'énergie de la particule ou de produire des interactions en amont des calorimètres. La reconstruction des événements est réalisée à l'aide du logiciel ILCSOFT v01-17-06. La chaîne de reconstruction permet notamment d'extraire les traces dans le trajectographe et les hits digitalisés dans les calorimètres.

La calibration des calorimètres électromagnétique et hadronique consiste alors en l'évaluation d'un ensemble de constantes de calibration pour les particules électromagnétique et hadronique. Nous dénoterons C_e^e l'ensemble des constantes de calibration pour la réponse électromagnétique dans le calorimètre électromagnétique, C_h^e pour la réponse hadronique dans le calorimètre électromagnétique, C_e^h pour la réponse électromagnétique dans le calorimètre hadronique et C_h^h pour la réponse hadronique dans le calorimètre hadronique.

Les ensembles des constantes de calibration du calorimètre électromagnétique C_e^e et C_h^e ont été estimés en amont de la digitalisation. Les hits reconstruits après

digitalisation dans le calorimètre électromagnétique contiendront donc une énergie nominale (en GeV), auquel se verra multipliée une constante de calibration pour donner l'énergie calibrée, soit électromagnétique, soit hadronique (voir équation 6.2). Pour la réponse du calorimètre hadronique, nous supposons qu'au vu des énergies attendues à l'ILC pour des particules issues des jets, les particules électromagnétiques (photons, électrons et positons) seront contenues dans le calorimètre électromagnétique, ou déborderont très peu dans le calorimètre hadronique. Nous supposons par la suite et dans une première approximation, que l'ensemble de ces constantes de calibration C_e^h et C_h^h est équivalent.

Ces constantes seront finalement extraites en générant des kaons long K_L^0 et en utilisant les constantes de calibration C_h^e dans le calorimètre électromagnétique. L'algorithme ArborPFA décrit dans la section 2 est utilisé pour extraire les différents PFOs de l'événement.

De façon générale, l'énergie totale de toute particule reconstruite est donné par :

$$E_{rec,\rho} = E_{ecal,\rho} + E_{hcal} \quad (6.1)$$

où $E_{ecal,\rho}$ et E_{hcal} sont donnés par :

$$E_{ecal,\rho} = \sum_i c_\rho^e \cdot e_i \quad (6.2)$$

$$E_{hcal} = \alpha \cdot N_1 + \beta \cdot N_2 + \gamma \cdot N_3 \quad (6.3)$$

avec $\rho = e$ ou h dépendamment du type de la particule (électromagnétique ou hadronique), e_i l'énergie dans le calorimètre électromagnétique déterminée après digitalisation, c_ρ^e la constante de calibration pour les particules électromagnétiques ($\rho = e$) ou hadronique ($\rho = h$) dans le calorimètre électromagnétique, N_1, N_2, N_3 le nombre de hits pour chacun des trois seuils du SDHCAL et α, β, γ les constantes de calibration énergétique du SDHCAL pour chacun des trois seuils. Les valeurs des trois constantes α, β et γ sont d'abord données par la table 6.1 (pré-calibration).

Pour calibrer ces trois constantes, la somme de l'énergie totale de chaque PFO est utilisée, et ce, pour éviter de biaiser la calibration en cas de fragmentation de la particule incidente en plusieurs PFOs. La nouvelle valeur des constantes est obtenue en minimisant le χ^2 suivant sur toute la gamme en énergie :

$$\chi^2 = \frac{1}{N_{evt} - 1} \sum_{i=1}^{N_{evt}} \frac{(E_{rec,i} - E_{mc,i})^2}{E_{mc,i}} \quad (6.4)$$

avec N_{evt} le nombre total d'événements sur toute la gamme en énergie, $E_{mc,i}$ l'énergie Monte Carlo de la particule incidente et $E_{rec,i}$ la somme de l'énergie totale de chaque PFO reconstruit. La table 6.1 montre les valeurs des constantes de calibration utilisées pour le calorimètre électromagnétique (pré-calibration) et les valeurs des trois paramètres α, β et γ obtenues après minimisation du χ^2 (calibration initiale). La figure 6.1 montre l'énergie reconstruite (6.1(a)) et la résolution en énergie des kaons neutres K_0^L après la calibration initiale en fonction de l'énergie

simulée. Excepté à basse énergie ($\leq 10 \text{ GeV}$) la déviation à la linéarité est inférieure à 5%. Cette déviation reste tout de même inférieure à 10% à basse énergie. La résolution relative obtenue est de $\sigma_E/E = 57\%/\sqrt{E} \oplus 3\%$. La résolution relative est inférieure à 8% à 80 GeV (contre $\sim 10\%$ pour la simulation du prototype du SDHCAL seul) étant donnée l'utilisation du calorimètre électromagnétique et de sa meilleure résolution en énergie.

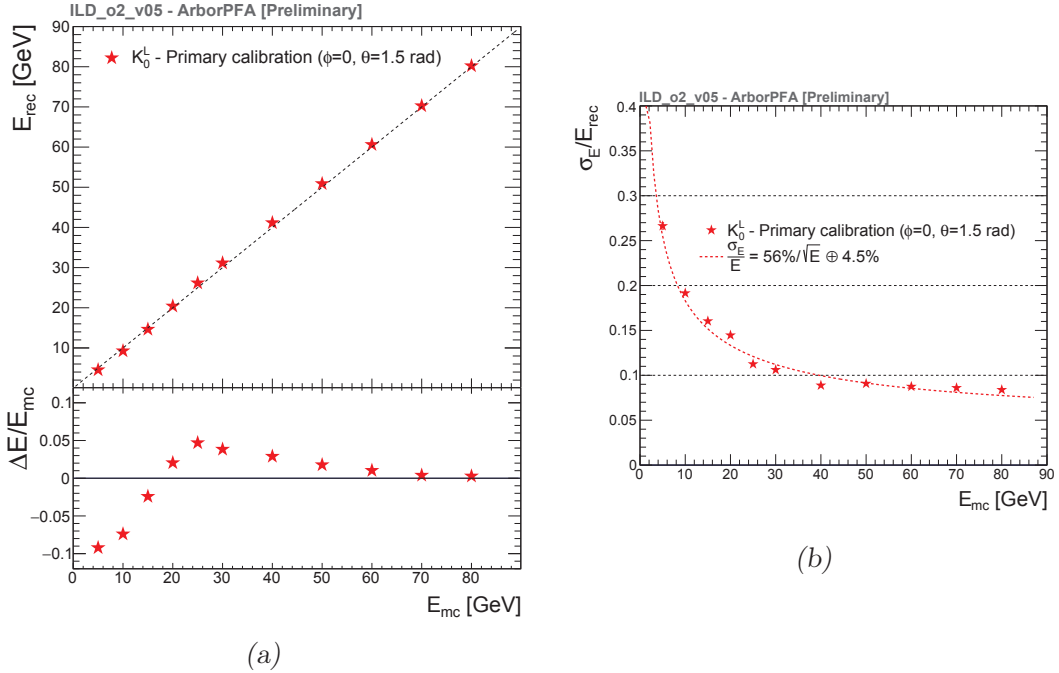


FIGURE 6.1 – (a) Énergie reconstruite et (b) résolution en énergie des kaons neutres K_0^L après la calibration initiale à angles fixes ($\phi = 0$, $\theta = 1.5 \text{ rad}$).

La figure 6.4(a) montre l'énergie moyenne reconstruite de kaons neutres K_0^L en fonction de $\cos\theta$. L'estimateur en énergie utilisé est celui décrit par l'équation 6.1. Comme le montre la figure 6.2, on peut distinguer quatre zones du détecteur en fonction de l'angle θ que l'on retrouve au niveau de l'énergie reconstruite.

La première zone de détection ($\cos(\theta) = [0, 0.57]$) présente uniquement des détecteurs dans le tonneau central. La troisième zone ($\cos(\theta) = [0.78, 0.98]$) présente uniquement des détecteurs dans les bouchons. La deuxième zone ($\cos(\theta) = [0.57, 0.78]$) est une zone mixte de détection tonneau central/bouchons. La quatrième zone ($\cos(\theta) = [0.98, 1]$) est majoritairement composée de calorimètres vers l'avant. L'énergie des particules contenues dans ces détecteurs sera supposée calibrée en amont étant donné que l'électronique de lecteur n'est pas celle du SDHCAL (électronique analogique) et ne fera pas l'objet de correction en énergie.

Les corrections en énergie qui vont être développées par la suite auront pour but final de restaurer au mieux la linéarité sur toute la gamme en angle θ , et ce, sur toute la gamme en énergie.

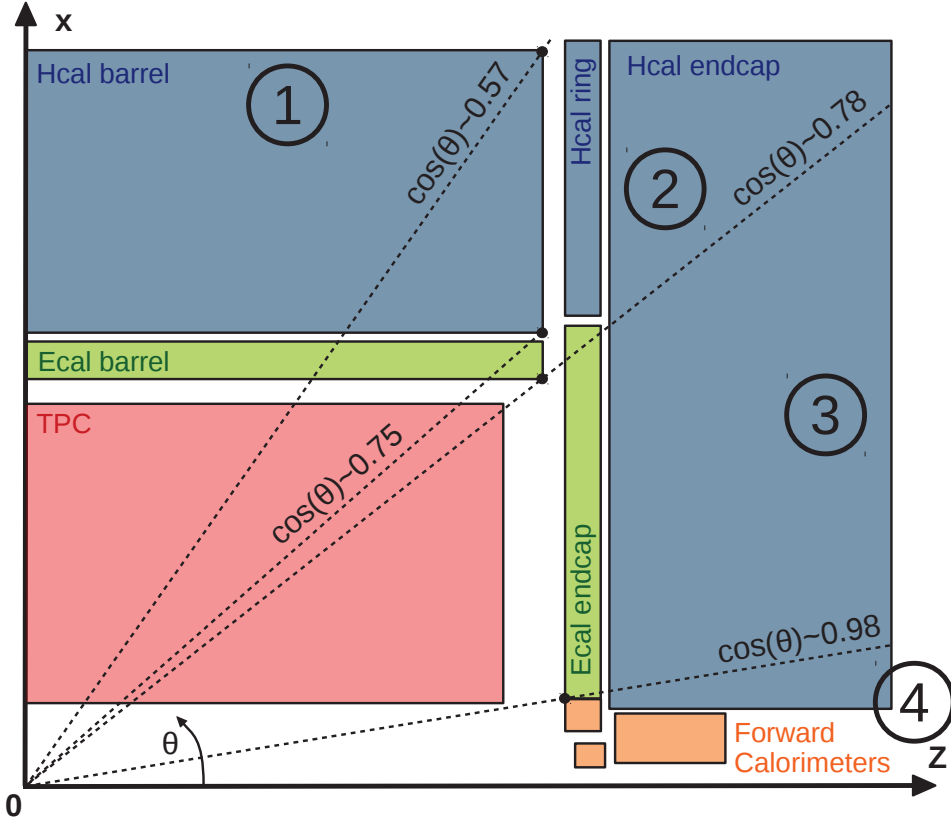


FIGURE 6.2 – Schéma illustrant le découpage du détecteur ILD en quatre régions : la zone 1) détectera les particules uniquement dans le tonneau central, la zone 2) dans le tonneau central et les bouchons, la zone 3) uniquement dans les bouchons et enfin la zone 4) dans les bouchons et les calorimètres à l'avant.

1.2 Correction inter-module

L'énergie reconstruite des kaons neutres K_0^L en fonction de $\cos\theta$ (voir figure 6.4(a)) montre deux régions où l'on observe une diminution en énergie. Ces deux régions sont situées autour des interstices inter-module du calorimètre hadronique à $z = \pm 485 \text{ mm}$ et $z = \pm 1423 \text{ mm}$, soit, étant donnée la géométrie du tonneau central, dans les deux intervalles $0.2 < |\cos\theta| < 0.23$ pour le premier interstice et $0.52 < |\cos\theta| < 0.57$ pour le deuxième.

On définit l'énergie déposée près des interstices par une particule donnée comme :

$$E_{gap} = \sum_i^{N_{hit}} \sum_g^{N_{gap}} e_i \epsilon_{g,i} \quad (6.5)$$

avec N_{hit} le nombre de hits dans le PFO, e_i l'énergie déposée dans la cellule i et $\epsilon_{g,i}$ la fonction définie par :

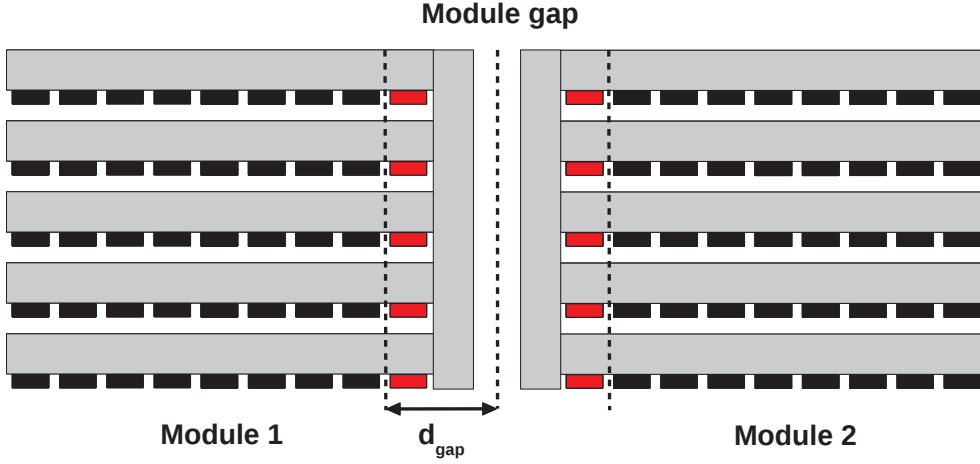


FIGURE 6.3 – Schéma illustrant la séparation entre deux modules du calorimètre hadronique dans le tonneau central. Les cellules en rouge correspondent à celles utilisées lors de la correction en énergie inter-module.

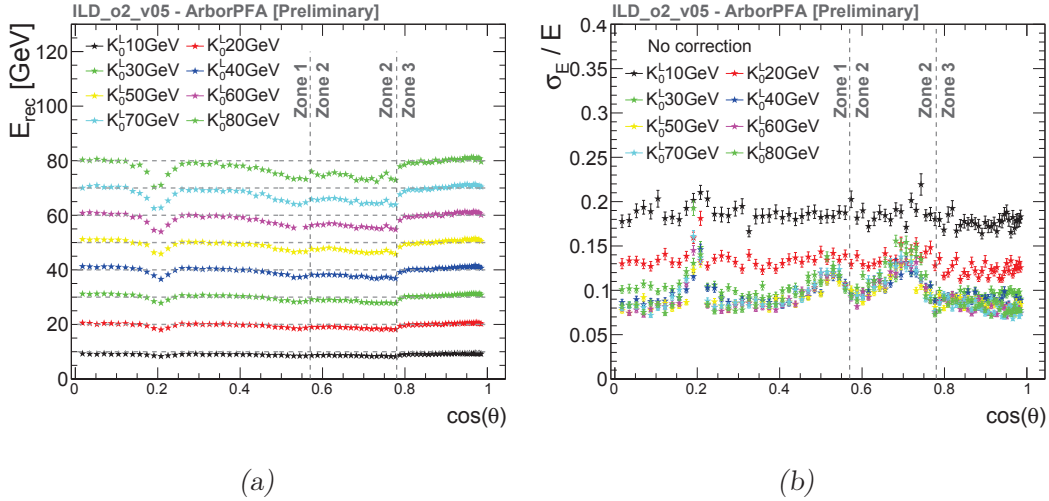


FIGURE 6.4 – (a) Énergie reconstruite et (b) résolution en énergie des kaons neutres K_0^L utilisant l'estimateur 6.1 en fonction de $\cos\theta$.

$$\epsilon_{g,i} = \begin{cases} 1 & |z_g - z_i| < d_{gap} \\ 0 & \text{sinon.} \end{cases}$$

avec z_g la position suivant l'axe \vec{e}_z du gap g , z_i la position suivant l'axe \vec{e}_z du hit i et d_{gap} une coupure sur la distance de séparation entre le hit et le gap considérés. Dans le calorimètre hadronique du tonneau central, cette distance est ajustée pour comptabiliser la première couche de cellules près des interstices. Le schéma illustré sur la figure 6.3 montre les différentes cellules (en rouge) pouvant être sélectionnées pour cette correction en énergie. Par exemple, pour une MIP laissant un hit par plan

et traversant un interstice, 2 hits, un de chaque coté de l'interstice, contribueront à l'évaluation de cette énergie.

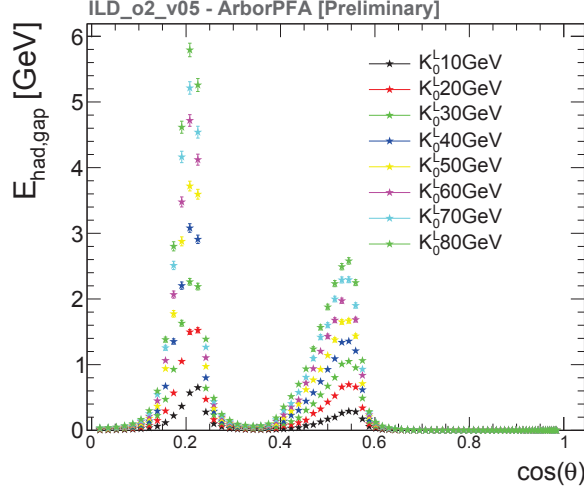


FIGURE 6.5 – Énergie moyenne déposée près des interstices inter-module dans le calorimètre hadronique dans la région du tonneau centrale en fonction de $\cos\theta$.

La figure 6.5 montre l'énergie déposée près des interstices du calorimètre hadronique dans la région du tonneau. On retrouve les deux courbes distinctes correspondant aux deux interstices inter-modules. On observe d'autre part que cette quantité d'énergie augmente avec l'énergie de la particule simulée. La différence en magnitude des deux pics s'explique par la différence en angle d'incidence de la particule simulé. En effet, pour le premier interstice, l'angle d'incidence est plus important. De ce fait, la surface à proximité des interstices pour une gerbe donnée s'en voit augmentée ainsi que le nombre de hits et donc au final, l'énergie comptabilisée.

Les deux couches d'absorbeurs refermant les modules dans le tonneau central peuvent être vues comme deux couches d'absorbeurs faisant partie du détecteur et la couche de vide entre les deux modules comme une zone instrumentée avec une efficacité nulle. L'idée de la correction que l'on veut appliquer ici, est de considérer que les déficits en énergie observés près des interstices peuvent se corriger en ajoutant, avec un certain facteur, l'énergie près des interstices. Nous proposons un premier estimateur en énergie de cette correction comme :

$$E_{rec,gap} = E_{rec,had} + \alpha_{gap} \cdot E_{gap} \quad (6.6)$$

avec $E_{rec,had}$ l'énergie hadronique reconstruite suivant l'estimateur donnée par la formule 6.1, α_{gap} une constante à ajuster et E_{gap} l'énergie près des interstices du calorimètre hadronique dans le tonneau central.

Au vu de la forme des distributions en énergie en fonction de $\cos\theta$, il devient complexe de définir un χ^2 à minimiser pour trouver une valeur de α_{gap} optimale. Cette constante est donc évaluée graphiquement en utilisant le maximum de la

distribution de l'énergie déposée près des interstices à 80 GeV et la valeur de l'énergie reconstruite au $\cos\theta$ correspondant. On trouve le maximum à $\cos\theta \sim 0.22$ avec une valeur $E_{gap} = 5.9$. On peut estimer l'énergie manquante pour une particule incidente de 80 GeV sur la figure 6.4(a) en se plaçant à $\cos\theta \sim 0.22$. On trouve une valeur de l'énergie manquante d'environ 9 GeV . On obtient alors pour la constante α_{gap} :

$$\alpha_{gap} \approx \frac{9}{5.9} \sim 1.5254 \quad (6.7)$$

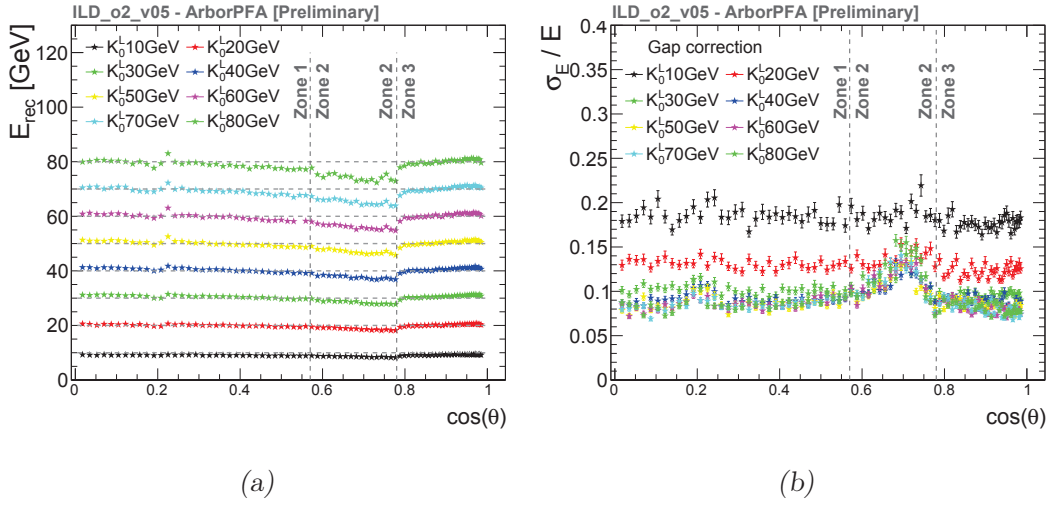


FIGURE 6.6 – (a) Énergie reconstruite et (b) résolution en énergie des kaons neutres K_0^L après la correction sur les interstices inter-module.

La figure 6.6(a) montre l'énergie reconstruite après l'application de cette correction. On peut voir que la correction est particulièrement efficace sur la région autour du deuxième interstice et permet de réajuster l'énergie reconstruite sur une pente qui décroît linéairement. L'effet de la correction sur le premier interstice est un peu moins efficace. Sur l'intervalle $0.14 < \cos\theta < 0.2$, la correction est sous-estimée tandis que sur l'intervalle $0.2 < \cos\theta < 0.24$ la correction est sur-estimée. Ceci est dû à la forme linéaire de l'estimateur 6.6 qui constitue une première approximation pour cette correction.

1.3 Correction angulaire

La deuxième correction en énergie, qui est une correction angulaire, permet de rétablir la linéarité dans les trois premières régions montrées par la figure 6.2. La correction en énergie sur les interstices inter-module est appliquée en amont de cette correction angulaire. La quatrième région ne sera pas considérée puisque, comme discuté précédemment, les technologies utilisées pour les calorimètres sont différentes de celles utilisées dans le tonneau central et les bouchons. Nous supposons la réponse de ces derniers détecteurs déjà calibrée en amont de cette correction.

La correction angulaire s'applique de la façon suivante. Pour chaque point en énergie considéré, une fonction linéaire est d'abord ajustée sur les régions 1 et 3 :

$$E_{barrel} = b_0 + b_1 \cdot \cos(\theta) \quad \text{dans la région 1} \quad (6.8)$$

$$E_{endcap} = e_0 + e_1 \cdot \cos(\theta) \quad \text{dans la région 3} \quad (6.9)$$

La figure 6.7(a) montre l'énergie reconstruite en fonction de $\cos\theta$ avec le résultat de chacun des ajustements pour les régions 1 et 3. Les figures 6.8(a) et 6.8(b) montrent quant à elle l'évolution des paramètres e_0 , e_1 , b_0 et b_1 en fonction de l'énergie simulée. On se propose d'ajuster chacune de ces courbes avec deux fonctions possibles, un polynôme d'ordre 1 ou un polynôme d'ordre 2, afin de les comparer.

Dans le premier cas, on a la paramétrisation suivante :

$$b_0 = b_{00} + b_{01} \cdot E_{mc} \quad (6.10)$$

$$b_1 = b_{10} + b_{11} \cdot E_{mc} \quad (6.11)$$

$$e_0 = e_{00} + e_{01} \cdot E_{mc} \quad (6.12)$$

$$e_1 = e_{10} + e_{11} \cdot E_{mc} \quad (6.13)$$

et pour le deuxième cas :

$$b_0 = b_{00} + b_{01} \cdot E_{mc} + b_{02} \cdot E_{mc}^2 \quad (6.14)$$

$$b_1 = b_{10} + b_{11} \cdot E_{mc} + b_{12} \cdot E_{mc}^2 \quad (6.15)$$

$$e_0 = e_{00} + e_{01} \cdot E_{mc} + e_{02} \cdot E_{mc}^2 \quad (6.16)$$

$$e_1 = e_{10} + e_{11} \cdot E_{mc} + e_{12} \cdot E_{mc}^2 \quad (6.17)$$

avec E_{mc} l'énergie simulée du kaon neutre K_0^L . L'extraction de ces paramètres nous permettra de calculer, en fonction de l'énergie reconstruite et de $\cos\theta$, la correction en énergie à apporter pour restaurer la linéarité dans les régions 1 et 3.

En effet, en réinjectant ces équations dans les équations 6.8 et 6.9, il est possible de retrouver les deux formes analytiques pour estimer la correction en énergie dans les régions 1 et 3. Dans le cas du polynôme d'ordre 1, on obtient :

$$E_1(\cos\theta, E_{rec}) = \frac{-b_{00} - b_{10} \cdot \cos\theta + E_{rec}}{b_{01} + b_{11} \cdot \cos\theta} \quad (6.18)$$

$$E_3(\cos\theta, E_{rec}) = \frac{-e_{00} - e_{10} \cdot \cos\theta + E_{rec}}{e_{01} + e_{11} \cdot \cos\theta} \quad (6.19)$$

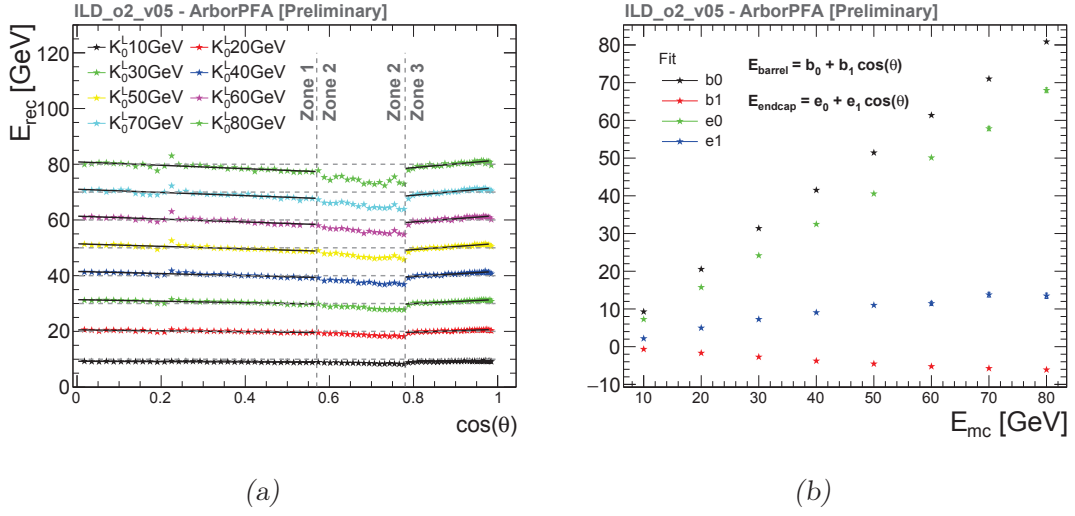


FIGURE 6.7 – (a) Énergie reconstruite de kaons neutres K_0^L simulés en fonction de $\cos(\theta)$ et les ajustements linéaires dans les régions 1 et 3. (b) Évolution des paramètres e_0 , e_1 , b_0 et b_1 après ajustement des courbes en énergie de la figure (a).

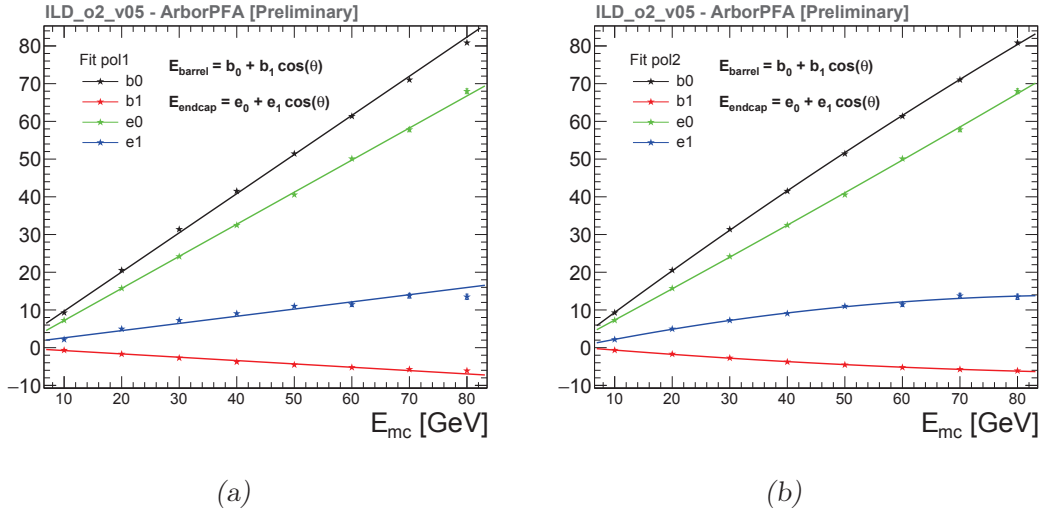


FIGURE 6.8 – Évolution des paramètres e_0 , e_1 , b_0 et b_1 après ajustement des courbes en énergie de la figure 6.7(a). Chacune des courbes associée aux quatre paramètres est ajustée avec (a) un polynôme d'ordre 1 et (b) un polynôme d'ordre 2 (voir table 6.1).

et pour le polynôme d'ordre 2 :

$$E_1(\cos\theta, E_{rec}) = \frac{-[b_{01} + b_{11} \cdot \cos\theta] + \sqrt{\Delta_1(E_{rec}, \cos\theta)}}{2 \cdot b_{02} + b_{12} \cdot \cos\theta} \quad (6.20)$$

$$E_3(\cos\theta, E_{rec}) = \frac{-[e_{01} + e_{11} \cdot \cos\theta] + \sqrt{\Delta_3(E_{rec}, \cos\theta)}}{2 \cdot e_{02} + e_{12} \cdot \cos\theta} \quad (6.21)$$

avec :

$$\Delta_1(E_{rec}, \cos\theta) = [b_{01} + b_{11} \cdot \cos\theta]^2 - 4[b_{02} + b_{12} \cdot \cos\theta][b_{00} + b_{10} \cdot \cos\theta - E_{rec}] \quad (6.22)$$

$$\Delta_3(E_{rec}, \cos\theta) = [e_{01} + e_{11} \cdot \cos\theta]^2 - 4[e_{02} + e_{12} \cdot \cos\theta][e_{00} + e_{10} \cdot \cos\theta - E_{rec}] \quad (6.23)$$

Dans la région intermédiaire 2, les particules peuvent tantôt être contenues dans le tonneau central, dans les bouchons ou être fractionnées dans les deux. Si la particule est totalement contenue dans la région du tonneau central, on pourra appliquer la correction en énergie donnée par l'équation 6.18 (resp. 6.20) spécifique à la région 1 et de même si la particule est totalement contenue dans les bouchons, la correction en énergie donnée par l'équation 6.19 (resp. 6.21) pourra être appliquée.

Le problème se pose lorsque la particule dépose une fraction de son énergie dans le tonneau et dans les bouchons. On définit la fraction f_{barrel} (resp. f_{endcap}) comme la fraction en énergie déposée de la particule reconstruite dans le tonneau central (resp. les bouchons). Les figures 6.9(a) et 6.9(b) montrent respectivement les fractions f_{barrel} et f_{endcap} en fonction de $\cos\theta$. L'évolution de ces fractions nous permet de définir un estimateur pour la correction en énergie dans la région intermédiaire qui sera aussi valable pour les régions 1 et 3 :

$$E(\cos\theta, E_{rec}) = f_{barrel} \cdot E_1(\cos\theta, E_{rec}) + f_{endcap} \cdot E_3(\cos\theta, E_{rec}) \quad (6.24)$$

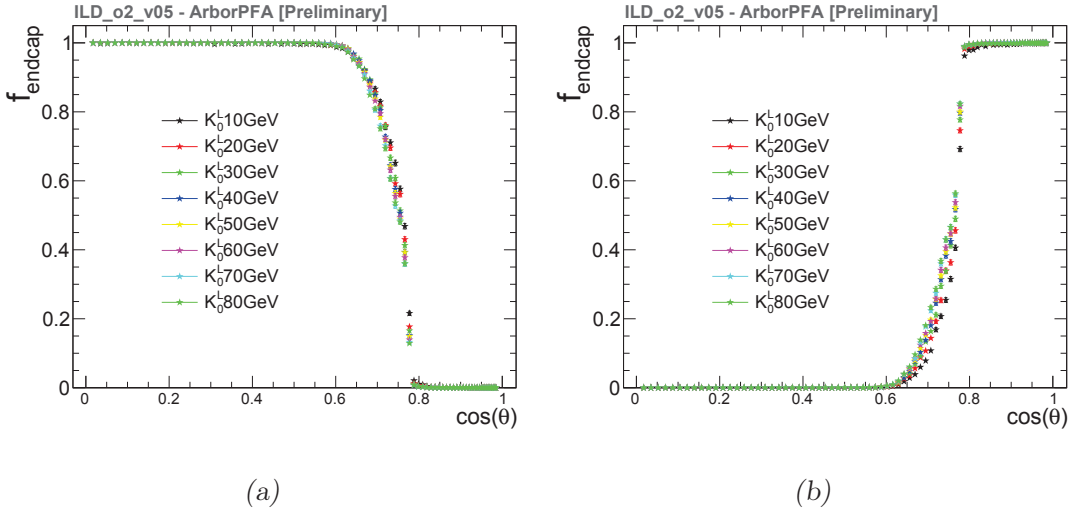


FIGURE 6.9 – Fraction de l'énergie totale déposée par des kaons neutres K_0^L dans (a) le tonneau central et (b) les bouchons, en fonction de $\cos\theta$.

Cette définition nous permet de retrouver la correction en énergie donnée par l'équation 6.18 (ou 6.20) dans la limite où $f_{barrel} \rightarrow 1$ et $f_{endcap} \rightarrow 0$ et de même, dans la limite où $f_{barrel} \rightarrow 0$ et $f_{endcap} \rightarrow 1$ de retrouver la correction en énergie donnée

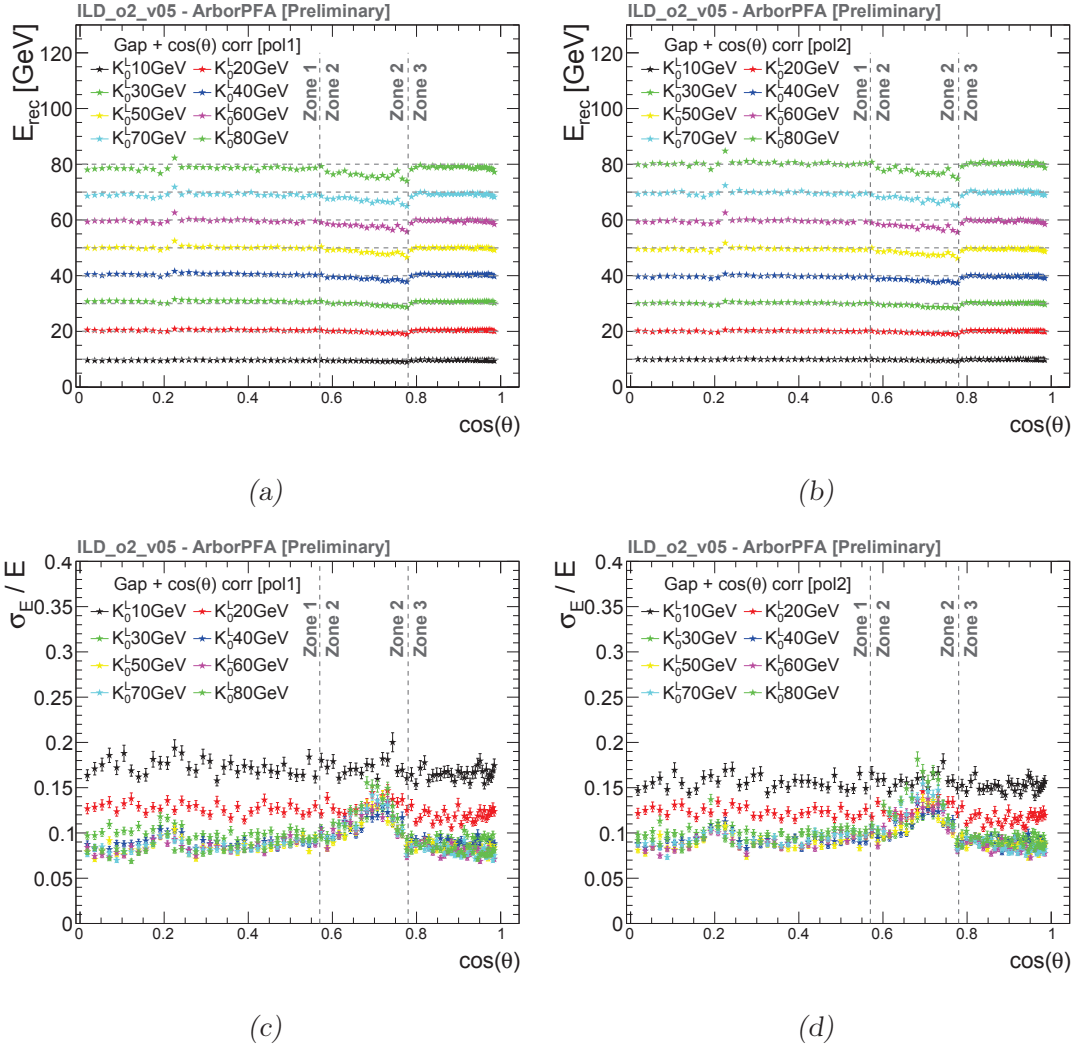


FIGURE 6.10 – Énergie reconstruite (en haut) et résolution en énergie (en bas) des kaons neutres K_0^L simulés en fonction de $\cos(\theta)$ après application de la correction inter-modules suivie de la correction angulaire en utilisant le polynôme d'ordre 1 (à gauche) et le polynôme d'ordre 2.

par l'équation 6.19 (ou 6.21). Elle permet d'autre part, d'assurer la continuité de la correction en énergie entre les zones 1 et 2 et les zones 2 et 3.

Les figures 6.10(b) et 6.10(a) montrent l'énergie reconstruite des kaons neutres K_0^L simulés en fonction de $\cos(\theta)$ après application de cette correction. On voit que l'énergie dans les régions 1 et 3 est correctement réajustée. La correction avec le polynôme d'ordre 2 permet cependant d'obtenir une meilleure linéarité. Les figures 6.10(c) et 6.10(d) montrent la résolution en énergie après l'application de cette correction. En utilisant le polynôme d'ordre 2, on observe une amélioration de la résolution à basse énergie et une légère détérioration à haute énergie dans la région 2.

La région 2 est quant à elle plus difficile à corriger étant donnée la part plus importante de zone non-équipée en détecteur, notamment dans la région $0.75 < \cos\theta < 0.78$, où les particules traversent un nombre important d'interstices et de zones non-instrumentées (voir figure 6.2). La correction permet néanmoins d'améliorer la linéarité, même si celle-ci n'est pas totalement réajustée. Ce dernier désaccord constituera la limite de cette correction et devra faire l'objet d'une étude plus poussée afin de mieux saisir les phénomènes mis en jeu dans cette région du détecteur et d'appliquer des corrections supplémentaires.

Calibration/Correction	Paramètre	Valeur	Erreur	Unité
Pré-calibration	c_e^e	1.007	-	-
	c_h^e	1.075	-	-
	α	0.0400452	-	GeV
	β	0.101322	-	GeV
	γ	0.39448	-	GeV
Calibration initiale ($\phi = 0, \theta = 1.5 \text{ rad}$)	α	0.0433(771)	$\pm 10^{-4}$	GeV
	β	0.0884(24)	$\pm 10^{-4}$	GeV
	γ	0.4573(53)	$\pm 10^{-4}$	GeV
Correction inter-modules ($\phi = 0, 0 < \theta < \frac{\pi}{2}$)	α_{gap}	1.5254	-	-
Correction angulaire ($\phi = 0, 0 < \theta < \frac{\pi}{2}$) polynôme d'ordre 1	b_{00}	-0.69744	± 0.0219633	GeV
	b_{01}	1.0391	± 0.000673522	-
	b_{10}	0.111512	± 0.0644432	GeV
	b_{11}	-0.0878576	± 0.00201236	-
	e_{00}	-1.32069	± 0.177342	GeV
	e_{01}	0.852075	± 0.00554714	-
	e_{10}	0.711654	± 0.197203	GeV
	e_{11}	0.190686	± 0.00614984	-
Correction angulaire ($\phi = 0, 0 < \theta < \frac{\pi}{2}$) polynôme d'ordre 2	b_{00}	-2.0695	± 0.037411	GeV
	b_{01}	1.15493	± 0.002649	-
	b_{02}	-0.00154133	$\pm 3.43796e - 05$	GeV ⁻¹
	b_{10}	0.574691	± 0.109654	GeV
	b_{11}	-0.127181	± 0.00779645	-
	b_{12}	0.000531069	± 0.000101724	GeV ⁻¹
	e_{00}	-0.913565	± 0.303824	GeV
	e_{01}	0.817793	± 0.021501	-
	e_{02}	0.000459511	± 0.000278442	GeV ⁻¹
	e_{10}	-0.837188	± 0.337783	GeV
	e_{11}	0.320991	± 0.0238778	-
	e_{12}	-0.00174508	± 0.000308989	GeV ⁻¹

TABLE 6.1 – Valeurs de constantes de calibration à différentes étapes de la calibration effectuée avec des kaons longs K_0^L . La calibration initiale est réalisée avec des angles $\phi = 0$ et $\theta = 1.5 \text{ rad}$, alors que les corrections inter-modules et angulaire utilisent toute la gamme en angle θ .

2 Les fonctions et les algorithmes ArborPFA

Comme discuté précédemment, l’environnement PandoraSDK permet de séparer l’implémentation des algorithmes de suivi de particules de leur utilisation par un code utilisateur. L’algorithme ArborPFA que nous allons détailler plus loin, a été développé avec cet environnement dans une librairie appelée *ArborContent*. L’implémentation qui va être présentée est relativement différente de celle discutée précédemment pour le prototype du SDHCAL (voir chapitre 5) même si le principe de la création d’arbres reste le même. Celle-ci doit maintenant tenir compte de plusieurs sous-détecteurs, avec une géométrie plus complexe, d’angles d’interaction de particules non perpendiculaires aux différents plans dans les calorimètres, etc ... La fonction de calcul des pseudo-plans jouera un rôle majeur dans l’implémentation des algorithmes, puisqu’elle permettra de s’affranchir des distances longitudinales qui peuvent être problématiques lorsque des particules traversent plusieurs sous détecteurs (voir plus bas). D’autre part, certains des sous-algorithmes qui vont être présentés, ont été pensés pour être réutilisés massivement et seront exécutés de nombreuses fois durant l’algorithme.

Nous commencerons par décrire les différentes fonctions utiles aux sous-algorithmes de suivi de particules : les fonctions de calcul de champs magnétique et des pseudo-plans ainsi que les fonctions d’identification de particules et de corrections en énergie. Les différents sous-algorithmes seront présentés un à un. La description de l’algorithme dans sa globalité, avec l’ordonnancement des sous-algorithmes, sera finalement présentés.

2.1 La fonction de calcul du champs magnétique

Cette fonction permet de retourner à tout moment dans un algorithme, le vecteur du champs magnétique en tout point. Cette fonction est très fortement dépendante du détecteur utilisé. Ici nous étudions le détecteur ILD_o2_v05. Dans le cadre des algorithmes de suivi de particules, la valeur du champ magnétique peut se simplifier contrairement au code de reconstruction de traces qui nécessite des cartes de champs très précises. Nous assignerons donc trois valeurs de champ magnétique dans trois régions différentes du détecteur. La table 6.2 résume les valeurs de champs magnétique utilisées en entrée du programme de reconstruction. La région centrale comprend tout le système trajectographique et les calorimètres électromagnétique et hadronique, aussi bien dans le tonneau central que dans les bouchons. Les deux autres régions extérieures concernent les chambres à muons dans le tonneau central et dans les bouchons, pour lesquelles deux valeurs du champ magnétique sont retenues. Dans tous les cas, le champ magnétique sera dirigé suivant l’axe \vec{e}_z .

Ces valeurs seront utilisées lors d’ajustement d’hélices sur les traces issues du trajectographe et serviront, par exemple, à calculer le point d’entrée des particules chargées dans le calorimètre électromagnétique ou différents calculs de distance entre un point et la trace considérée.

Dans la table 6.2, la valeur de z (4072 mm) correspond à la distance entre l’origine et la face avant de la première chambre à muons dans les bouchons alors

Région	Champs magnétique (\vec{e}_z)	Limite géométrique
Région centrale	3.5 T	$ z < 4072 \text{ mm}$ $r < 3790 \text{ mm}$
Tonneau muon	-1.5 T	$ z < 4072 \text{ mm}$ $r > 3790 \text{ mm}$
Bouchon muon	0.01 T	$ z > 4072 \text{ mm}$

TABLE 6.2 – Table résumant les valeurs de champs magnétique pour le détecteur *ILD_o2_v05* utilisées en entrée du programme de reconstruction *ArborPFA*.

que la valeur de r (3790 mm) correspond à la distance moyenne en r de la culasse utilisée pour le retour de champ.

Il est cependant important de noter qu'au vu de l'implémentation actuelle des différents sous algorithmes, seul la valeur du champ magnétique dans la zone centrale sera utilisée.

2.2 La fonction de calcul des pseudo-plans

La fonction de calcul des pseudo-plans utilisée est la même que celle utilisée dans l'implémentation proposée par *PandoraPFA* (voir 2.2). Elle permet de s'affranchir des contraintes géométriques, notamment des transitions entre sous-détecteurs dans la région intermédiaire dans les bouchons avec $r > 1847 \text{ mm}$ (rayon interne du calorimètre électromagnétique dans le tonneau central). Les distances entre les hits étant extrêmement variables dans cette région à cause des différentes transitions entre sous détecteurs, une différence entre deux pseudo-plans sera plus appropriée qu'une distance cartésienne pour estimer la distance entre deux hits.

2.3 Les fonctions d'identification de particules

Les fonctions d'identification de particules sont utilisées pour déterminer la nature des clusters reconstruits par les algorithmes. Ces fonctions sont appelées au sein des algorithmes afin de prendre des décisions quant à de possibles associations ou encore vis-à-vis de l'estimation de l'énergie des particules (électromagnétique ou hadronique).

Ainsi, quatre fonctions d'identification de particules peuvent être implémentées dans l'environnement *PandoraSDK* :

- identification des gerbes électromagnétiques
- identification des photons
- identification des électrons (et positons par extension)
- identification des muons

Tout autre particule ne passant aucun de ces tests d'identification pourra être considéré comme un hadron neutre ou chargé, selon la présence de traces associées dans le cluster. Ces fonctions ont été implémentées dans l'algorithme *ArborPFA*,

puisque l'application des corrections en énergie sur les clusters nécessite de connaître la nature du cluster (voir paragraphe sur les corrections en énergie plus bas). La fonction d'identification des gerbes électromagnétiques est généralement utilisée dans les deux fonctions d'identification des photons et des électrons pour identifier dans un premier temps un cluster de type électromagnétique et ensuite appliquer des coupures supplémentaires pour identifier les photons et les électrons. Elle est aussi fréquemment utilisée lors de l'application des corrections en énergie pour décider si un cluster est électromagnétique ou hadronique et appliquer la correction en énergie appropriée.

- Identification des gerbes électromagnétiques. On dénote par f_{em} la fraction d'énergie du cluster contenue dans le calorimètre électromagnétique, par $f_{em,last}$ la fraction en énergie électromagnétique du cluster contenue dans les derniers pseudo-plans [$pl > pl_{min}$] du calorimètre électromagnétique, par pl_{start} le pseudo-plan du début de la gerbe et enfin par pl_{ext} l'extension du cluster en nombre de pseudo-plans entre le début de la gerbe et le dernier pseudo-plan du cluster. Les coupures suivantes discriminent alors les gerbes électromagnétiques :
 1. $f_{em} < f_{em,min}$, $f_{em,last} > f_{em,last,max}$, $pl_{ext} > pl_{ext,max}$, avec $f_{em,min}$, $f_{em,last,max}$ et $pl_{ext,max}$ trois paramètres de coupures pour chacune des trois variables. Les gerbes électromagnétiques ont une grande probabilité d'être entièrement contenu dans le calorimètre électromagnétique, compte tenu des énergie attendues pour les particules individuelles au sein d'un jet à l'ILC [33]. Un débordement d'un certain pourcentage est toléré. Cette coupure permet aussi de discriminer les hadrons puisque ceux-ci continuent généralement leur développement dans le calorimètre hadronique.
 2. $pl_{start} > pl_{start,min}$. Étant donnée la longueur de radiation du calorimètre électromagnétique dans le détecteur ILD ($X_0 > 0.35\text{ cm}$), la probabilité qu'une particule électromagnétique démarre dans un pseudo-plan profond reste faible. Il peut arriver que le pseudo-plan de départ de la gerbe n'ait pas pu être trouvé (cas du muon non-radiatif par exemple). En cas d'échec, une coupure discriminante est appliquée.
- Identification des photons. Cette fonction d'identification utilise la fonction d'identification des gerbes électromagnétiques comme premier discriminant. La seconde étape est de différencier les photons des électrons. Pour cela on demande à ce qu'aucune trace ne soit associée au cluster. Une coupure supplémentaire est cependant appliquée pour différencier les photons des fragments hadroniques neutres dans le calorimètre hadronique. On dénote par \vec{p}_{cl} la direction du cluster évaluée par un ajustement linéaire sur la position de tous les hits du cluster entre le pseudo-plan de départ de la gerbe pl_{start} et le dernier pseudo-plan, par \vec{c}_{cl} le vecteur correspondant au centre de gravité du cluster et enfin par $\theta_{dir,c}$ l'angle entre le centre de gravité du cluster \vec{c}_{cl} et sa direction \vec{p}_{cl} . Le photon est discriminé si l'angle $\theta_{dir,c}$ est trop important. Des fragments hadroniques neutres secondaires peuvent être émis à partir de gerbes hadro-

niques primaires et comportent souvent un angle $\theta_{dir,c}$ plus important comparé à celui des photons émis depuis le point de collision.

- Identification des électrons. Plus simple encore que la fonction d'identification des photons, elle utilise la fonction d'identification des gerbes électromagnétique comme premier discriminant puis vérifie qu'au moins une trace est associé au cluster.
- Identification des muons. Dans la version actuelle de l'algorithme de reconstruction, aucun sous-algorithme ne traite de la reconstruction des muons en particulier. Les hits dans les chambres à muons ne sont donc pas utilisés. Cette fonction d'identification pour les muons tiendra compte de cette spécificité. Le premier critère pour identifier un muon est bien entendu la présence d'une trace associée au cluster. Les critères de sélection sont assez similaires à ceux utilisés lors de la sélection des gerbes hadroniques dans le prototype du SDHCAL (voir chapitre 3 section 4.3.1), à ceci près que les muons dans le détecteur ILD sont soumis à un champ magnétique non-nul et possèdent par définition une courbure non-nulle. Tout d'abord un nombre minimum de pseudo-plans dans le calorimètre électromagnétique et hadronique [$N_{pl} > N_{pl,min}$] est requis pour discriminer les pions ne démarrant pas de gerbe hadronique dans les calorimètres et terminant leurs trajectoires bien avant la fin du calorimètre hadronique. Un pseudo-plan est dit de haute densité dans le calorimètre électromagnétique (resp. hadronique) lorsque le nombre de hits non-isolés qu'il contient dépasse une certaine coupure [$N_{e,dens} > N_{e,dens,max}$] (resp. [$N_{h,dens} > N_{h,dens,max}$]). Un cluster est considéré comme un muon si, après les premières coupures, le nombre de pseudo-plans de haute densité reste bas [$N_{pl,dens} < N_{pl,dens,max}$].

2.4 Les fonctions de correction en énergie

En entrée du programme de reconstruction, chaque hit dans les calorimètres est créé avec 4 énergies différentes :

- l'énergie dite *d'entrée*, qui est simplement celle qui provient du hit après digitalisation,
- l'énergie équivalent à une MIP, correspondant à l'énergie moyenne déposée par un MIP dans le calorimètre en question (électromagnétique ou hadronique). Pour le calorimètre SDHCAL, cette énergie ne peut être calculée de manière simple. Ce champ sera toutefois rempli avec l'énergie calibrée équivalente au seuil le plus bas mais ne sera pas utilisée dans la version actuelle des algorithmes,
- l'énergie électromagnétique, correspondant à l'énergie calibrée avec une réponse électromagnétique,
- l'énergie hadronique, correspondant à l'énergie calibrée avec une réponse hadronique.

De la même façon que pour l'énergie électromagnétique/hadronique pour chaque hit, les fonctions de corrections en énergie pourront être de nature électromagnétique

ou hadronique. Cela est particulièrement important dans le cas des corrections qui ont été présentées précédemment (voir section 1) puisque que celles-ci concernent uniquement les hadrons. De ce fait, aucune correction en énergie ne sera apportée pour les particules électromagnétiques. Concernant les hadrons, une calibration primaire sera effectuée en amont pour définir l'énergie calibrée pour chaque seuil. Les corrections en énergie près des interstices inter-modules dans le tonneau central et en fonction de $\cos(\theta)$ seront appliquées.

D'autre part il sera possible, pour un cluster donné, de calculer 4 énergies différentes :

- l'énergie électromagnétique, calculée comme la somme de l'énergie électromagnétique de chaque hit contenu dans le cluster,
- l'énergie hadronique, calculée comme la somme de l'énergie hadronique de chaque hit contenu dans le cluster,
- l'énergie électromagnétique corrigée, utilisant les fonctions de corrections en énergie électromagnétique à partir de l'énergie électromagnétique du cluster,
- l'énergie hadronique corrigée, utilisant les fonctions de corrections en énergie hadronique à partir de l'énergie hadronique du cluster.

Les calculs en énergie corrigée demanderont plus de temps de calcul comparé aux deux premières fonctions qui pourront, dans un premier temps, donner une première estimation de l'énergie du cluster de façon plus rapide.

2.5 L'ordonnement des algorithmes

Comme expliqué précédemment, les sous-algorithmes implémentés dans l'environnement PandoraSDK sont réutilisables au sein de l'algorithme dans sa globalité. Certains des sous-algorithmes qui vont être présentés dans la section suivante traiteront d'un problème de topologie particulier alors que d'autres seront des algorithmes composites exécutant d'autres sous-algorithmes. Le but de cette section est de décrire et discuter l'ordonnement des différents sous-algorithmes de reconstruction.

L'algorithme est séparé en 5 parties. L'ordonnement des sous-algorithmes se déroule comme présenté sur la figure 6.11.

On trouve d'abord les algorithmes de préparation de l'événement (CaloHitPreparation et EventPreparation) qui ne sont exécutés qu'une seule fois au début de l'algorithme. Dans le premier, des propriétés sur les hits dans les calorimètres sont calculées. Le deuxième crée les listes de hits et de traces qui seront utilisés dans les phases de *clustering*.

Vient ensuite la reconstruction des photons qui permet de retirer les photons de l'événement pour pouvoir traiter ensuite uniquement les particules chargées (électrons/positons et hadrons chargés) et les hadrons neutres lors du *clustering* principal. Cet algorithme de reconstruction de photons est un algorithme composite formé d'un algorithme de *clustering* exécuté uniquement sur les hits des différents calorimètres électromagnétiques et d'une série d'algorithmes d'associations topologiques (voir section suivante). La reconstruction des photons est suivie d'un algorithme de suppression des hits appartenant aux photons sur le chemin des traces dans les

- ▶ EnergyExcessReclustering
 - Clustering algorithms :
 - ArborClustering1
 - ArborClustering2
 - ArborClustering3
 - ArborClustering4
 - ArborClustering5
 - ArborClustering6
 - ArborClustering7
 - ArborClustering8
 - ArborClustering9
 - ArborClustering10
 - ArborClustering11
 - ArborClustering12
 - TopologicalAssociations
 - TrackClusterAssociation
- ▶ MissingEnergyReclustering
 - Clustering algorithms :
 - ArborClustering13
 - ArborClustering14
 - ArborClustering15
 - ArborClustering16
 - ArborClustering17
 - ArborClustering18
 - ArborClustering19
 - ArborClustering20
 - ArborClustering21
 - ArborClustering22
 - ArborClustering23
 - ArborClustering24
 - TopologicalAssociations
 - TrackClusterAssociation
- ▶ SplitTrackReclustering

Algorithmes de
re-clustering

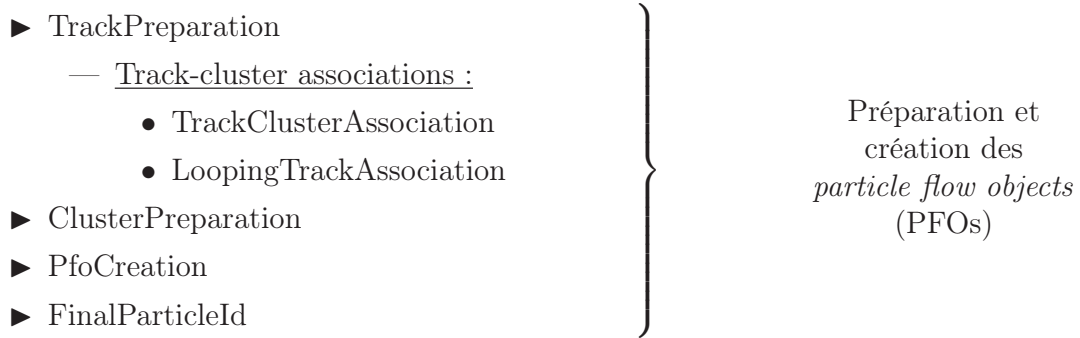


FIGURE 6.11 – Ordonnancement des sous-algorithmes de reconstruction. Cinq phases se distinguent : i) préparation de l'événement, ii) reconstruction des photons, iii) *clustering* principal, iv) *re-clustering* et v) préparation et création des PFOs.

calorimètres. La récupération de ces hits sera précieuse lors de la connexion des hits pour les particules chargées et évitera de fragmenter inutilement les clusters chargés.

La troisième partie de l'algorithme est responsable de la création des clusters primaires à partir des hits restant après la reconstruction des photons. Contrairement à l'algorithme de reconstruction des photons, tous les hits des différents calorimètres seront utilisés (électromagnétique et hadronique). En sortie de cet algorithme, on obtient une liste de clusters contenant des électrons/positons, des hadrons chargés et des hadrons neutres en plus de la liste des clusters de photons créée en amont.

La phase suivante consiste en l'optimisation de la phase de *clustering* primaire. Il est important de noter que seuls les clusters créés lors la phase de *clustering* primaire participeront à cette étape d'optimisation. Trois algorithmes de *re-clustering* sont exécutés ici. Leur ordre d'apparition est très important dans l'algorithme. Comme le montre l'ordonnancement, l'ordre est le suivant :

- EnergyExcessReclustering
- MissingEnergyReclustering
- SplitTrackReclustering

Le premier aura pour effet d'améliorer la compatibilité $E - p$ lorsqu'un excès d'énergie est observé dans un cluster chargé. Pour cela, le cluster sera dans la plupart du temps fragmenté en plusieurs clusters. Le deuxième aura aussi pour effet d'améliorer la compatibilité $E - p$ mais dans le cas où un déficit en énergie est observé dans un cluster chargé. Cependant, l'effet inverse se produira : le cluster aura tendance à absorber d'autres clusters dans son entourage. L'idée de cet ordonnancement est celle de la stratégie *split-and-merge* qui permettra dans un premier temps de fragmenter les clusters qui présentent un excès d'énergie puis de potentiellement réassigner ces même fragments à d'autres clusters présentant un déficit en énergie.

La figure 6.12 illustre cette stratégie avec deux clusters chargés candidats aux deux algorithmes de *re-clustering*. Le premier algorithme de *re-clustering* permet de récupérer la bonne compatibilité $E - p$ avec la trace de 6 GeV et produit un nouveau cluster de 7 GeV tandis que le deuxième algorithme utilisera ce dernier pour récupérer la bonne compatibilité $E - p$ avec la trace de 7 GeV .

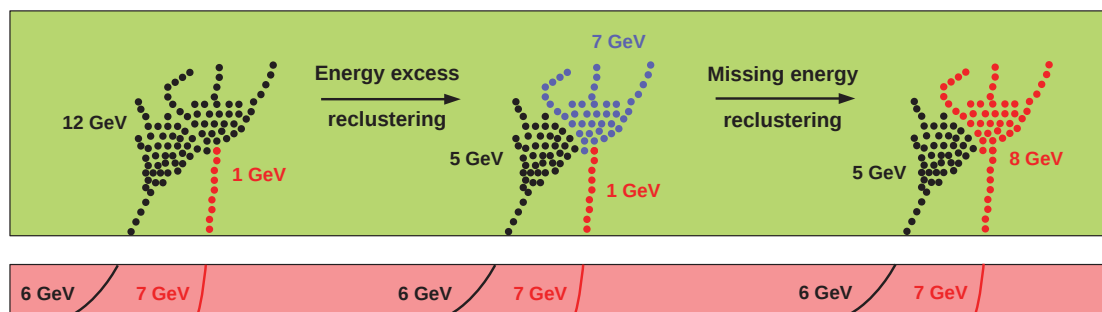


FIGURE 6.12 – Schéma illustrant la stratégie de *split-and-merge* déployée avec les deux premiers algorithmes de *re-clustering*. Le cluster chargé de 12 GeV (en noir) est d’abord re-configuré pour obtenir un cluster neutre de 7 GeV. Le cluster chargé de 1 GeV (en rouge) est ensuite re-configuré et englobe cette fois-ci le cluster neutre de 7 GeV.

Le troisième algorithme de *re-clustering* se placera en dernière position étant donné qu’il n’est pas directement impliqué dans cette stratégie. Son rôle est très spécifique et permet de traiter un problème particulier où plusieurs traces sont associées au même cluster et donc de séparer les différentes contributions dans les calorimètres pour chacune des traces associées.

Concernant les différents algorithmes de *clustering* utilisés dans les deux premiers algorithmes de *re-clustering*, l’ordonnement des sous-algorithmes est le même que pour l’algorithme de *clustering* principal, seuls les paramètres changent. Pour l’algorithme de *re-clustering EnergyExcessReclustering*, les paramètres varient de façon à ce que la taille du cluster considéré diminue à chaque algorithme de *clustering* exécuté. De la même façon, pour l’algorithme de *re-clustering MissingEnergyReclustering*, les paramètres varient de façon à ce que la taille du cluster considéré augmente au fur-et-à-mesure. Les paramètres concernés sont ceux des outils de connexion de hits (avec et sans trace) et souvent liés aux distances de connexion, soit :

- l’écart en pseudo-plan entre deux hits à connecter,
- la distance transverse de connexion,
- l’angle entre la position du hit et les connexions.

La dernière partie consiste en la préparation et la création des PFOs. Les traces et les clusters éligibles à la création de PFOs sont regroupés dans deux listes respectivement et les PFOs sont créés. A noter la présence de deux algorithmes d’association entre traces et clusters lors de la préparation des traces permettant de peaufiner les associations entre traces et clusters. Le dernier algorithme permet d’identifier les particules et de faire persister cette information dans les méta-données de chaque PFOs en sortie du programme de reconstruction.

2.6 Les algorithmes de reconstruction

Préparation des hits [CaloHitPreparation]

Cet algorithme a pour but de calculer certaines quantités et de les faire persister pour chaque hit de l'événement. Ces quantités sont les suivantes :

- La densité. Pour un hit i , la densité D_i est calculée de la manière suivante :

$$D_i = \frac{1}{n+1} \sum_i^n \frac{d_{i,j}}{d_{max}} \quad (6.25)$$

avec $d_{i,j}$ la distance entre les hits i et j appartenant au même pseudo-plan et dont la distance est inférieure à d_{max} et n le nombre de hits satisfaisant les deux conditions précédentes. La densité sera comprise entre 0 et 1, 0 signifiant que le hit i ne possède pas de voisin proche dans le même pseudo-plan et 1 que le hit i possède un grand nombre de voisins dans le même pseudo-plan.

- La variable d'isolation. On définit la distance d'isolation entre les hits i et j comme :

$$d_{i,j,isol} = \|\vec{p}_j - \vec{p}_i\| \sin(\theta_{i,j}) \quad (6.26)$$

avec \vec{p}_i la position du hit i et $\theta_{i,j}$ l'angle entre le vecteur \vec{p}_i et le vecteur $\vec{p}_j - \vec{p}_i$. On considère par la suite que le hit i n'est pas isolé si au moins N_{isol} hits ont une distance d'isolation inférieure à une coupure [$d_{i,j,isol} < d_{isol,max}$] et si la distance entre le hit i et une des traces de l'événement est inférieure à une certaine coupure [$\Delta_{track} < \Delta_{track,isol}$]. Cette dernière condition évitera de voir un hit appartenant à la trace primaire dans les calorimètres, étiqueté comme isolé. Les hits étiquetés comme isolés seront traités de manière particulière lors de la phase de *clustering* (voir "outils de connexions" plus bas).

Préparation de l'événement [EventPreparation]

Dans l'environnement PandoraSDK, le contenu de l'événement (hits, traces, clusters, PFOs, etc ...) est géré sous forme de listes que l'utilisateur peut créer, remplir, vider ou supprimer par le biais des algorithmes. Parmi ces listes, la *liste courante* sert de point d'accès central aux algorithmes. L'algorithme de préparation de l'événement choisit les traces et les hits qui seront utilisés lors de la phase de *clustering*.

Les hits de l'événement seront séparés en deux listes, une pour les calorimètres électromagnétique et hadronique et une pour les chambres à muons. La liste de hits des calorimètres sera enregistrée comme liste courante pour les prochains algorithmes. En ce qui concerne les traces, seules les traces pouvant être utilisées dans la phase de *clustering* seront ajoutées dans une liste séparée, liste qui sera enregistrée comme liste de trace courante par la suite. Les traces qui sont retenues, sont celles qui ne comportent pas de traces filles.

Outil de connexion des hits (avec trace) [TrackDrivenSeeding]

Cet outil de connexion de hits est utilisé dans les phases de *clustering*. Il permet de connecter les hits entre eux en utilisant les traces issues du trajectographe comme référence. La figure 6.13 montre les différentes étapes exécutées pour connecter les hits à partir des traces.

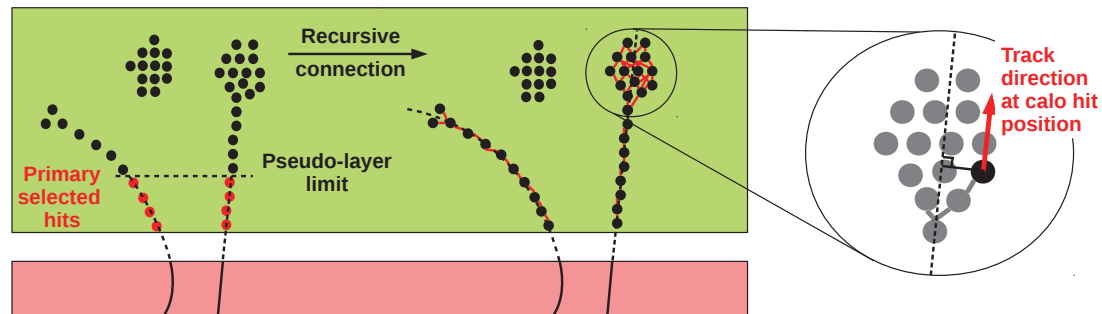


FIGURE 6.13 – Schéma illustrant les différentes étapes exécutées par l'outil de connexion des hits utilisant les traces issues du trajectographe comme référence.

La première phase consiste à trouver pour chaque trace des hits non isolés dans les N_{pl} premiers pseudo-plans du calorimètre électromagnétique, à une distance cartésienne maximale Δ_{track} de la trace considérée. Un exemple de sélection de hits est montré en rouge sur la figure 6.13 à gauche. La deuxième phase utilise ces hits comme point de départ et connecte récursivement les hits dans les plans suivants. Pour chaque hit considéré, la projection de la position du hit est calculée. La tangente à la trace en ce point définit alors la direction à suivre pour connecter les prochains hits. La figure 6.13 à droite montre un exemple de direction (flèche en rouge) suivi par un hit pour ses prochaines connexions. Une fois la direction calculée, une connexion ne peut s'effectuer uniquement qu'avec des hits dans les N_{pl} pseudo-plans suivants et uniquement si la distance transverse est inférieure à une certaine coupure Δ_{max} .

Cet outil de connexion n'était pas présent dans la version dédiée au prototype du SDHCAL. En effet, les particules dans le prototype du SDHCAL entraient avec un angle perpendiculaire aux différents plans et n'étaient pas plongées dans un champ magnétique. Cet outil de connexion permet en outre de connecter de façon efficace tous les hits appartenant aux particules chargées, particulièrement celles avec une faible impulsion (donc un grand rayon de courbure) dans un environnement plongé dans un champ magnétique non-nul. Cet outil est bien entendu entièrement dédié aux particules chargées.

Un exemple de résultat après connexion est montré sur la figure 6.13 au centre. On voit que la particule neutre au centre ne présente aucune connexion à la fin de l'algorithme.

Outil de connexion des hits (sans trace) [ConnectorSeeding]

Cet outil de connexion est très similaire à celui qui est appliqué lors de la reconstruction dans le prototype SDHCAL (*ConnectorSeeding1*). Il est dédié aux

particules neutres provenant du point de collision. La connexion des hits est basée sur la position du hit lui-même dans les calorimètres, position qui donne également une très bonne estimation de la direction de propagation du hit.

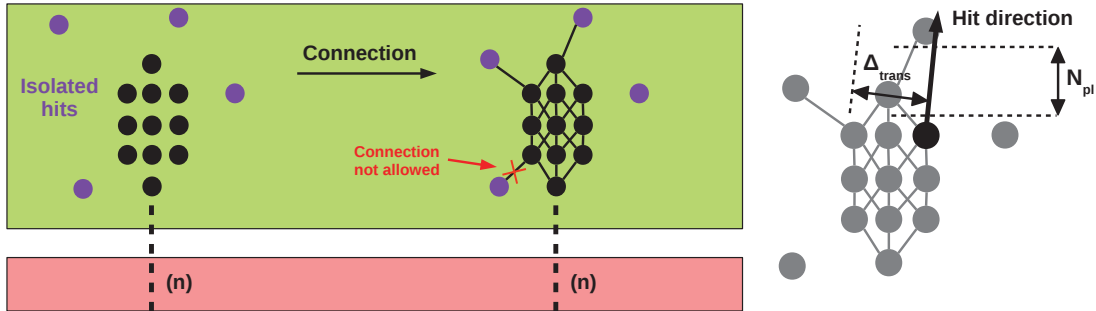


FIGURE 6.14 – Schéma illustrant la connexion des hits sans utiliser les traces issues du trajectographe.

Ainsi, un hit donné sera connecté avec d'autres hits dans les N_{pl} pseudo-plans suivants, à une distance transverse maximale de Δ_{trans} . L'étiquetage des hits isolés joue un rôle important ici, puisque un hit isolé pourra être connecté avec un hit dans les pseudo-plans précédents mais pas dans les pseudo-plans suivants. Ceci peut s'expliquer de façon physique : un hit très isolé ou un hit isolé gravitant autour d'une gerbe est généralement issu d'une particule en fin de chaîne de désintégration et ne donnera donc pas naissance à d'autres particules dans les prochains pseudo-plans.

La figure 6.14 montre un exemple de connexion de hits (en noir) avec quelques hits isolés (en mauve) gravitant autour de la particule neutre. Le schéma au centre montre les connexions possibles entre hits ainsi qu'une connexion impossible entre un hit isolé et un hit dans le pseudo-plan suivant. Le schéma de droite montre les différentes variables mises en jeu lors de la connexion d'un hit vers d'autres hits dans les pseudo-plans suivants.

Outil de nettoyage des connexions [ConnectorCleaning]

C'est l'outil final utilisé pour supprimer certaines connexions créées par les outils de connexions décrits précédemment et former des structures en arbre. Cet outil est en tout point similaire à celui implémenté dans la version dédiée au prototype SDHCAL (voir "*nettoyage des connexions 1*", page 86).

Fusion des arbres aux racines proches [ClosebySeedMerging]

Cet algorithme est équivalent à l'algorithme de "*fusion des arbres neutres*" déployé dans le cadre de la reconstruction dans le prototype du SDHCAL (voir page 90). Quelques changements ont néanmoins été apportés afin de tenir compte de la géométrie plus complexe du détecteur ILD.

Les différents clusters ayant des racines proches [$\Delta_{seed} < \Delta_{seed,max}$] et dont la différence en pseudo-plans est inférieure à une certaine coupure [$\Delta_{pl} < \Delta_{pl,max}$] sont fusionnés. Dans la version dédiée au prototype du SDHCAL, cette différence

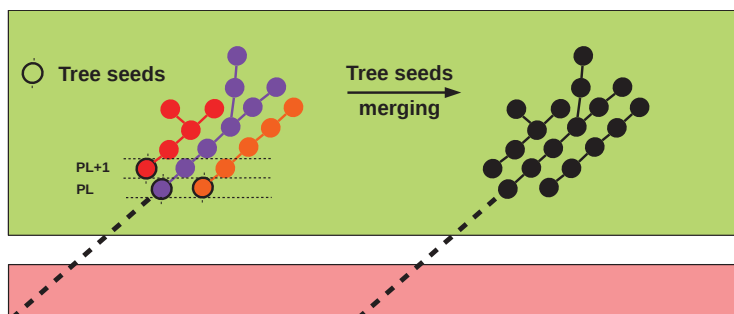


FIGURE 6.15 – Schéma illustrant la fusion des arbres au racines proches. Deux arbres ayant deux racines dans des pseudo-plans proches mais différents peuvent être fusionnés.

était fixée à 0 (même pseudo-plan). Ici, les particules neutres peuvent interagir avec un angle non-nul par rapport aux plans physiques des calorimètres. Cet angle non-nul a pour effet de parfois décaler ces racines d'un pseudo-plan. La figure 6.15 illustre un exemple où une particule neutre a interagi avec un angle important et où l'on peut voir plusieurs racines dans des pseudo-plans différents.

Association entre traces et clusters [TrackClusterAssociation]

Le but de cet algorithme est de fabriquer des associations entre les traces issues du trajectographe et les clusters reconstruits dans les calorimètres.

La première partie de cet algorithme consiste en une sélection des traces et des clusters potentiellement associables. Les traces éligibles sont celles atteignant les calorimètres et les clusters ceux dont le pseudo-plan interne l_{inner} est inférieur à une certaine coupure $l_{inner,max}$. Cette première sélection en amont permet de réduire le nombre de traces et de clusters lors des calculs d'association et donc de réduire la complexité de l'algorithme. La figure 6.16 a) illustre la sélection de deux clusters parmi les trois représentés.

La partie principale de cet algorithme consiste à associer à chaque trace le cluster le plus compatible. Il est aussi possible de ne pas associer de cluster à une trace si les contraintes ne le permettent pas. Enfin, il est possible suivant la configuration de l'algorithme, d'associer plusieurs traces à un seul cluster. Une seconde configuration importante au sein de l'algorithme est la possibilité d'utiliser ou non la compatibilité $E - p$ lors des calculs de contraintes entre une trace et un cluster. En effet, suivant l'étape d'appel de ce sous-algorithme dans l'algorithme global, les clusters pourraient ne pas être totalement reconstruits (fragmentations éventuelles, fusion des hits isolés, etc ...) et leurs énergies s'en verraient donc faussées. Pour éviter de mauvaises associations, des informations uniquement liées à la topologie des traces et des clusters peuvent être utilisées en premier lieu pour contraindre ces associations.

La recherche du meilleur cluster à associer à une trace se déroule en plusieurs étapes :

1. calcul des contraintes (voir plus bas) et de la compatibilité entre la trace et

chaque cluster,

2. coupure sur certaines contraintes afin de réduire la liste d'associations possibles. Le cluster avec la meilleure compatibilité (définition plus bas) parmi les restants est retenu,
3. dans le cas où un cluster compatible est retenu :
 - (a) si l'algorithme est configuré pour que l'association entre un cluster et plusieurs traces soit possible, alors une association entre le cluster et la trace est construite,
 - (b) dans le cas contraire, pour un cluster donné, la trace ayant la meilleure compatibilité $E - p$ est sélectionnée pour créer une association en fin d'algorithme.

On définit le barycentre interne \vec{p}_{inner} d'un cluster par rapport à une trace comme :

$$\vec{p}_{inner} = \frac{1}{\sum \epsilon_i} \cdot \sum_i^n \epsilon_i \cdot \vec{p}_i \quad (6.27)$$

avec n le nombre de hits dans le cluster, \vec{p}_i la position du hit i appartenant au cluster et ϵ_i valant 1 si les conditions suivantes sont satisfaites : le hit i doit être contenu dans un cône d'angle d'ouverture θ_{track} positionné au point d'entrée de la trace dans le calorimètre et dont la direction est donnée par l'impulsion en ce point, et 0 sinon. D'autre part le hit i devra être contenu dans les N_{pl} premiers pseudo-plans du cluster considéré. Dans nombre de cas ce barycentre est indéfini, puisque seuls les clusters situés près de la projection de la trace pour donner un barycentre défini. La figure 6.16 b) illustre la sélection et le calcul du barycentre interne d'un cluster par rapport à une trace.

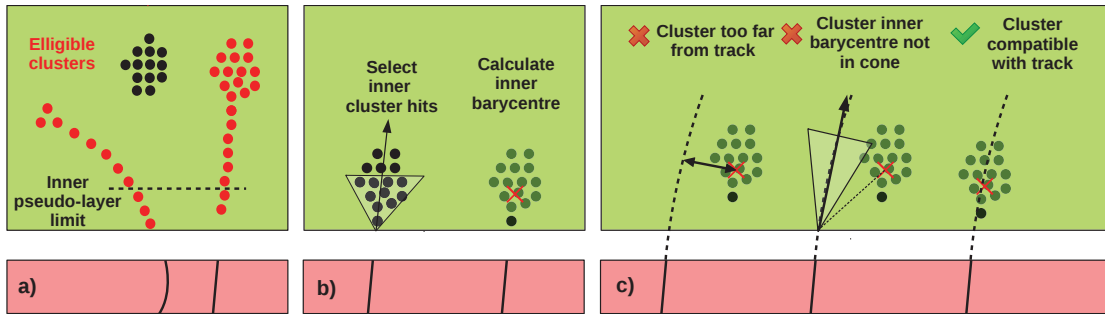


FIGURE 6.16 – Schéma illustrant les variables topologiques mises en jeu lors de l'association entre traces et clusters. a) Sélection des clusters b) calcul du barycentre interne d'un cluster par rapport à une trace c) exemple de deux rejets et d'une compatibilité entre un trace et un cluster.

On définit alors la distance de compatibilité entre une trace et un cluster $\Delta_{track,cluster}$ comme la plus courte distance entre la trace et le barycentre interne du cluster considéré. On définit d'autre part l'angle d'ouverture entre une trace et un cluster $\theta_{track,cluster}$ comme l'angle entre le barycentre interne \vec{p}_{inner} du cluster et la

direction de la trace au point d'entrée dans les calorimètres. Dans l'implémentation actuelle, on trouve dans la liste des contraintes :

- l'angle d'ouverture $\theta_{track,cluster}$
- la distance trace-cluster $\Delta_{track,cluster}$
- la compatibilité, définie comme :
 - la distance trace-cluster $\Delta_{track,cluster}$, si la configuration requiert uniquement des contraintes topologiques,
 - le produit de la compatibilité $E-p$ avec la distance trace-cluster $\Delta_{track,cluster}$, si la configuration requiert des contraintes énergétiques.

Parmi les coupures sur les contraintes, on trouve une coupure sur la distance trace-cluster [$\Delta_{track,cluster} < \Delta_{track,cluster,max}$] et une coupure sur l'angle d'ouverture [$\theta_{track,cluster} < \theta_{track,cluster,max}$]. La figure 6.16 c) illustre trois cas : le cas où le cluster est trop éloigné de la trace (à gauche), le cas où l'angle d'ouverture est trop grand (au centre) et le cas où le cluster et la trace sont compatibles pour une association (à droite).

Association des fragments de traces [MipFragmentMerging]

Cet algorithme permet de résoudre deux problèmes de topologie. Le premier concerne les fragments de traces issus des interactions dans les calorimètres généralement produites par des particules neutres de faible énergie qui voyagent dans les calorimètres et qui produisent, après une interaction, une particule chargée voyageant elle-même sur une courte distance. L'algorithme aura pour but d'identifier ces clusters et de les assigner à un cluster parent. Le second problème qui nous intéresse concerne les particules chargées produisant un segment de trace primaire qui s'interrompt avant de produire une gerbe éventuelle. Cette brisure peut être due à une inefficacité dans le calorimètre ou à une conversion prématurée en particule neutre. Cette seconde partie de l'algorithme tentera d'associer un ou plusieurs clusters se trouvant plus loin dans les calorimètres. La figure 6.17 a) montre deux exemples liés à ces deux problèmes de topologie.

Dans les deux cas, la première étape est d'identifier les clusters ayant la topologie d'une trace : une série de hits alignés les uns derrière les autres et sans gerbe résultante.

La première partie de l'algorithme vise à identifier ces segments de traces dans les calorimètres. Les critères suivants sont requis pour considérer un cluster comme un fragment de MIP :

- un nombre minimum de hits dans le cluster [$N_{hit} > N_{hit,min}$],
- le cluster ne doit pas être étiqueté comme un photon,
- le rapport transverse (voir équation 3.5) doit être faible [$\lambda_{tr} < \lambda_{tr,max}$],
- une extension minimale en nombre de pseudo-plans [$\Delta_{pl} > \Delta_{pl,min}$],
- un faible nombre de hits moyen par pseudo-plan [$N_{hit}/N_{pl} < N_{h,pl,max}$],
- un pourcentage faible de pseudo-plans ne contenant pas de hit [$N_{touch,pl}/N_{pl} > \epsilon_{mip}$],

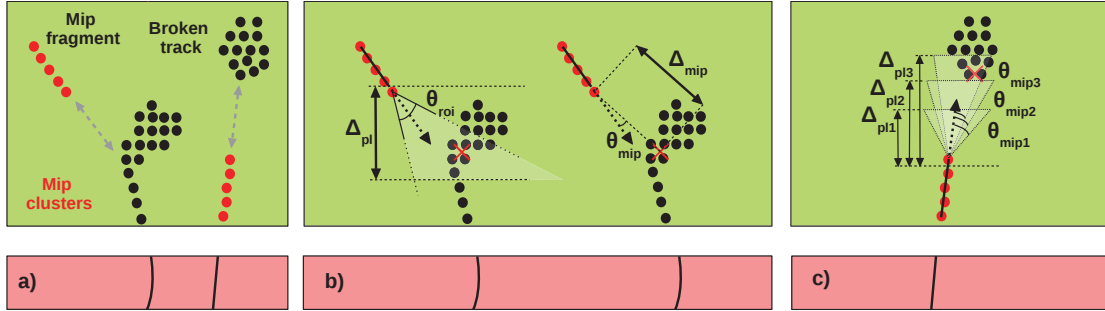


FIGURE 6.17 – a) Les deux problèmes de topologie traités par l’algorithme d’association des fragments de traces, b) variables et coupures mises en jeu dans le premier problème de topologie (mip fragment) et c) variables et coupures mises en jeu dans le deuxième problème de topologie (broken track).

- l’absence (la présence) de trace associée au cluster dans le cas du premier (second) problème de topologie.

Une fois un cluster MIP identifié, quelques propriétés générales peuvent être extraites :

- la direction de la MIP, calculée avec un ajustement linéaire (non-pondéré) sur la position de tous les hits du cluster,
- le barycentre interne, défini comme le barycentre des hits dans le pseudo-plan interne du cluster MIP,
- le barycentre externe, défini comme le barycentre des hits dans le pseudo-plan externe du cluster MIP.

Dans le premier problème de topologie, le but est de trouver un éventuel cluster parent dans lequel le cluster MIP sera fusionné. Quelques coupures sont d’abord appliquées pour déterminer si un cluster peut être considéré comme un parent éventuel :

- le cluster parent ne doit pas être étiqueté comme un photon,
- il doit avoir un nombre minimum de hits [$N_{hit} > N_{hit,min}$],
- il doit avoir un minimum d’énergie hadronique [$E_{had} > E_{had,min}$].

Pour un cluster MIP donné, le cluster parent dans lequel il doit être fusionné est déterminé de la manière suivante. Pour chaque cluster parent possible :

1. on calcule un barycentre avec tous les hits du cluster parent contenus dans une région définie par :
 - la direction du cluster MIP,
 - un angle d’ouverture maximal θ_{roi} ,
 - une différence du nombre de pseudo-plans Δ_{pl} entre chaque hit du cluster et le pseudo-plan du barycentre interne du cluster MIP.

Ces différentes quantités sont illustrées sur le schéma de gauche de la figure 6.17 b).

2. le nombre de hits utilisés pour le calcul de ce barycentre doit être supérieur à un certain nombre $[N_{hit,roi} > N_{hit,roi,min}]$ pour que le cluster parent soit éligible,
3. le barycentre du cluster parent doit être contenu dans un angle d'ouverture $\theta_{mip,max}$ et à une distance du barycentre interne du cluster MIP inférieure à une distance maximale $\Delta_{mip,max}$. Le schéma de droite de la figure 6.17 b) illustre ces deux variables.

Enfin, parmi tous les clusters parents, celui passant les coupures décrites précédemment et ayant la plus petite distance Δ_{mip} et le plus petit angle d'ouverture θ_{mip} est choisi.

Pour le deuxième problème de topologie, on cherche à associer un ou plusieurs fragments détachés de la particule se trouvant plus loin dans les calorimètres. Cela signifie que l'on s'attend aussi à une mauvaise compatibilité $E - p$ entre le cluster MIP et sa trace associée.

Pour un cluster MIP donné, on cherche à associer tout fragment respectant les conditions suivantes :

1. le pseudo-plan interne du fragment doit être supérieur au pseudo-plan externe du cluster MIP,
2. la compatibilité $E - p$ après association des deux clusters doit s'améliorer (voir équations 4.2 et 4.3)
3. le barycentre interne du fragment considéré est contenu dans l'une des trois régions définies par un angle d'ouverture θ_{mip} et une différence de pseudo-plan maximale de Δ_{pl} . Les trois régions définies auront trois valeurs de $\theta_{mip,i}$ et $\Delta_{pl,i}$ différentes de tel sorte que :

$$\theta_{mip,1} > \theta_{mip,2} > \theta_{mip,3} \quad (6.28)$$

$$\Delta_{pl,1} < \Delta_{pl,2} < \Delta_{pl,3} \quad (6.29)$$

Un exemple de définition de ces trois régions est illustré sur la figure 6.17 c).

Suppression et fusion des fragments [FragmentRemoval]

Après formation des clusters au sein de l'algorithme, des structures en arbre ne correspondant pas à des particules peuvent être formées. La figure 6.18 a) illustre quelques exemples de fragments que l'on peut retrouver après une phase de *clustering*.

Le but de cet algorithme est d'identifier ces fragments, de les supprimer et de répartir les hits issus de ces fragments dans les clusters avoisinants. Parmi les clusters disponibles, seuls ceux n'ayant pas de trace associée, comportant un faible nombre de hits $[N_{hit} < N_{hit,frag}]$ et ayant une faible énergie hadronique $[E_{had} < E_{had,max}]$, pourront potentiellement être considérés comme des fragments. Une sélection est ensuite appliquée sur les clusters restants. Les clusters satisfaisants au moins une des conditions suivantes sont considérés comme des fragments à supprimer :

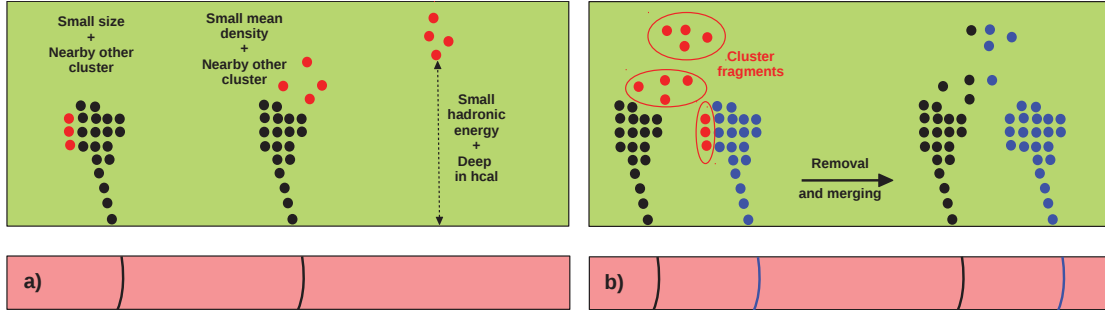


FIGURE 6.18 – a) Quelques exemples de fragments formés après une phase de *clustering* et b) suppression des fragments et répartition des hits dans les clusters voisins.

- les clusters contenus dans un seul pseudo-plan,
- les clusters contenant uniquement des hits du calorimètre hadronique et ayant une faible énergie hadronique [$E_{had} < E_{had,max,2}$],
- les clusters avec un nombre de hits très faible [$N_{hit} < N_{hit,rmvl}$] ou une énergie hadronique très faible [$E_{had} < E_{had,max,3}$]
- les clusters avec une densité moyenne (moyenne de la densité de tous ses hits) faible [$d < d_{frag}$] et proches des clusters non-fragments. Pour cela on compte le nombre de hits appartenant au fragment proche des autres clusters non-fragments (distance la plus courte parmi tous les hits d'un cluster). Si ce nombre de hits dépasse un certain nombre [$N_{hit,near} > N_{hit,min}$], alors on considère le fragment comme proche des autres clusters.

Une fois tous les fragments identifiés, les clusters sont supprimés et les hits sont répartis dans les clusters environnants suivant les mêmes critères utilisés par l'algorithme de "*fusion des hits entourant les clusters*" (voir plus loin).

Association des clusters pointant [PointingClusterAssociation]

Dans cet algorithme, nous traiterons des clusters fragments issus des gerbes hadroniques. Le but sera de définir une direction depuis laquelle ces fragments neutres proviennent et d'y trouver un cluster parent possible pour les y attacher. Dans cet algorithme nous traiterons uniquement le cas particulier où ces fragments sont issus de clusters chargés afin d'utiliser l'amélioration de la compatibilité $E - p$ comme critère final d'association. La figure 6.19 a) montre un exemple d'une particule neutre émise à partir d'une particule chargée et présentant tous les critères pour être fusionné avec cette dernière.

Parmi tous les clusters disponibles dans l'événement, une première sélection est effectuée. Seuls les clusters ayant au minimum N_{hit} et N_{pl} pseudo-plans touchés participeront à la suite de l'algorithme. Les photons seront aussi ignorés. La suite consiste à trouver, pour chaque cluster neutre, un potentiel cluster chargé dans lequel le cluster neutre sera fusionné si les contraintes topologiques et énergétiques le permettent.

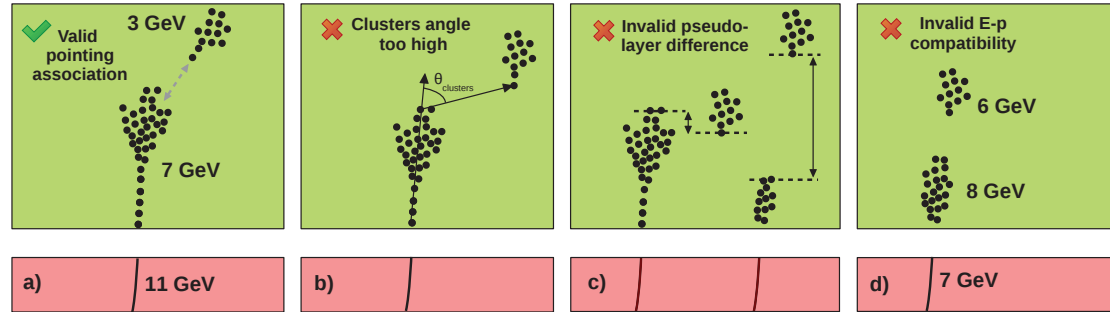


FIGURE 6.19 – a) Exemple d’association possible entre un cluster neutre et un cluster parent chargé. b) Exemple d’association impossible à cause d’un angle entre clusters trop important. c) Exemple d’association impossible à cause d’une différence en pseudo-plan invalide (trop importante ou négative). d) Exemple d’association impossible à cause d’une incompatibilité $E - p$.

On calcule d’abord pour chaque cluster quelques propriétés :

- la direction du cluster, définie par un ajustement linéaire sur les $N_{pl,fit}$ premiers pseudo-plans du cluster si celui-ci est neutre, sinon, sur tous les hits du cluster,
- le barycentre interne (externe), défini par le barycentre de la position des hits dans le premier (dernier) pseudo-plan du cluster,
- la position interne (externe), définie par la projection du barycentre interne (externe) sur la direction du cluster,
- le pseudo-plan interne (externe), défini par le pseudo-plan correspondant à la position interne (externe).

Des coupures sont appliquées sur le potentiel cluster parent :

- la différence en nombre de pseudo-plans entre le pseudo-plan interne du cluster neutre et le dernier pseudo-plan du cluster parent considéré ne doit pas excéder un certain nombre [$\Delta_{pl} < \Delta_{pl,max}$] (voir figure 6.19 c)),
- l’angle d’ouverture entre la direction du cluster neutre et la direction du cluster parent, défini par un ajustement linéaire sur la position de tous ses hits, ne doit pas excéder un certain angle [$\theta_{fit} < \theta_{fit,max}$],
- l’angle d’ouverture entre le vecteur liant la position externe du cluster parent et la position interne du cluster neutre, et la direction du cluster parent, ne doit pas excéder un certain angle [$\theta_{cluster} < \theta_{cluster,max}$] (voir figure 6.19 b)),
- la compatibilité $E - p$ après fusion des clusters doit soit s’améliorer, soit rester compatible [$\chi_{E-p}^2 < \chi_{E-p,max}^2$] (voir figure 6.19 d))

Parmi les clusters parents passant toute ces coupures, celui ayant le plus petit angle $\theta_{cluster}$ est choisi.

Fusion des hits entourant les clusters [SurroundingHitsMerging]

C'est généralement l'algorithme final dans la liste des algorithmes d'associations. Il permet d'associer les hits qui n'ont pas encore été assignés aux clusters à ces derniers. On définit la distance entre un hit et un cluster $\Delta_{h,c}$ comme la distance la plus courte entre tous les hits du cluster et le hit considéré. L'algorithme cherche alors la distance minimale entre un hit non-associé et tous les clusters disponibles afin de l'y associer. Une coupure est tout de même appliquée sur cette distance [$\Delta_{h,c} < \Delta_{h,c,max}$] afin d'éviter d'assigner aux clusters des hits trop éloignés.

Association des traces à grand rayon de courbure [LoopingTrackAssociation]

Algorithme d'association entre traces et clusters dédié aux traces avec un grand rayon de courbure. Ces traces particulières ont une impulsion faible et sont pour la plupart projetées sur la face avant des bouchons. Ces traces croisent parfois beaucoup plus d'interstices dans la région des calorimètres vers l'avant. Les premiers hits des clusters résultants se trouvent dans ce cas dans des couches plus profondes dans les calorimètres. Les associations entre ces traces et clusters sont parfois mal gérées par l'algorithme standard d'association entre traces et clusters.

Les traces considérées sont les suivantes :

- les traces ayant un grand rayon de courbure,
- les traces sans cluster associé,
- les traces projetées sur les bouchons.

Parmi les traces sélectionnées, chaque cluster sans trace associé est examiné afin de trouver celui avec la meilleure compatibilité. Pour chaque cluster et pour une trace donnée, plusieurs variables sont calculées :

- le vecteur position du barycentre interne \vec{p}_{cl} défini comme le barycentre des hits du cluster contenus dans le pseudo-plan interne,
- le point \vec{p}_{trk} défini comme le point de croisement de la trace avec le plan $z = \vec{p}_{cl} \cdot \vec{e}_z$,
- la distance transverse entre la trace et le cluster Δ_{XY} définie par :

$$\Delta_{XY} = \sqrt{(\vec{p}_{cl} \cdot \vec{e}_x - \vec{p}_{trk} \cdot \vec{e}_x)^2 + (\vec{p}_{cl} \cdot \vec{e}_y - \vec{p}_{trk} \cdot \vec{e}_y)^2} \quad (6.30)$$

- l'angle θ_{cl-trk} défini comme l'angle entre la direction de la trace au point \vec{p}_{trk} et la direction du cluster définie par un ajustement linéaire des hits du cluster dans les N_{pl} premiers pseudo-plans,

Les clusters avec un barycentre interne \vec{p}_{cl} non contenu dans les bouchons, avec une distance Δ_{XY} supérieure à une certaine distance [$\Delta_{XY} > \Delta_{XY,max}$] et un angle θ_{cl-trk} supérieur à une certaine coupure [$\theta_{cl-trk} > \theta_{cl-trk,max}$], ne sont pas considérés. Enfin parmi les clusters restants, celui avec la meilleure compatibilité $E-p$ est choisi pour être associé à la trace.

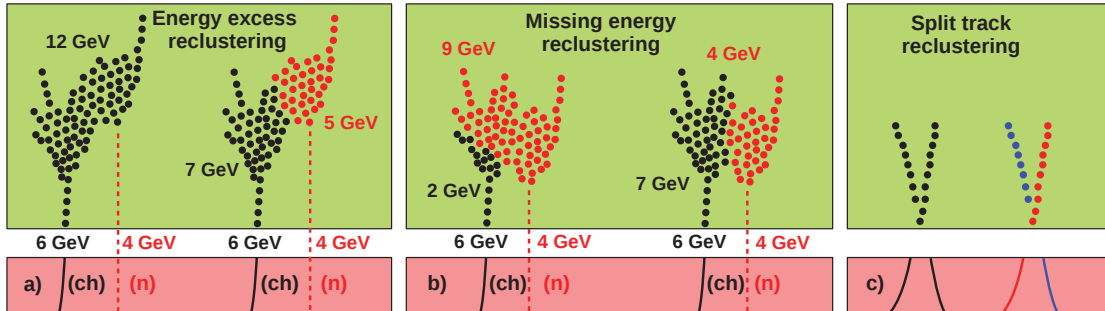


FIGURE 6.20 – Exemples de problèmes topologiques traités par les algorithmes de *re-clustering* : a) *re-clustering* lors d'un excès d'énergie, b) *re-clustering* lors d'un déficit en énergie, c) *re-clustering* lors d'associations multiples.

Re-clustering lors d'un excès d'énergie [EnergyExcessReclustering]

Après une première phase de *clustering*, l'algorithme de *clustering* ou les algorithmes d'associations topologiques ont pu échouer à reconstruire certaines particules dans les calorimètres. Ceci peut se produire au sein de jets assez denses dans lesquels les clusters reconstruits de chaque particule se trouvent en contact les uns avec les autres. Dans ce type de configuration, une part importante de l'énergie d'un cluster peut se retrouver assignée dans un autre. Dans le cas d'un hadron neutre, ce dernier pourrait aussi se retrouver complètement fusionné dans un autre cluster par inadvertance.

Certains des algorithmes de *re-clustering* qui vont être présentés sont basés, comme pour l'implémentation de PandoraPFA [44], sur une application itérative d'algorithmes de *clustering*, en variant les paramètres de ces derniers de sorte à produire des clusters avec des topologies différentes. A chaque itération, un estimateur détermine si la reconstruction opérée a amélioré la situation. La configuration pour laquelle l'estimateur s'est avérée minimum sera gardée et les clusters correspondants remplaceront ceux de la situation initiale. Cette procédure a été décrite dans la section 2.3 du chapitre 4 par le pseudo-code 1.

De façon générale et dans l'implémentation actuelle, les algorithmes de *re-clustering* ne traiteront pas les particules chargées sortant du détecteur. Ce type de topologie est synonyme d'énergie manquante dans le cluster. Le principal critère utilisé pour déclencher un *re-clustering* étant la compatibilité $E - p$, ce dernier se montrera invalide à chaque algorithme de *clustering* exécuté. De ce fait, la procédure de *re-clustering* ne sera pas concluante. Pire encore, des particules neutres à proximité du cluster sortant du détecteur pourraient être associées à ce dernier pour combler le manque d'énergie.

Le premier algorithme de *re-clustering* qui est présenté ici traite d'une situation particulière où une particule chargée montre un excès d'énergie par rapport à sa trace associé. On ne considérera pas le cas où plusieurs traces sont associées à un cluster. Ce cas sera traité en aval par un algorithme dédié (voir "*re-clustering* lors d'associations multiples" plus bas).

Pour chaque cluster dans la liste courante présentant un excès en énergie, une

série de plusieurs algorithmes de *clustering* est exécutée, suivie à chaque itération d'une série d'algorithmes d'associations topologiques. A chaque itération, parmi tous les clusters créés, celui étant associé avec la trace du cluster initial est identifié. La qualité de la reconstruction est estimée avec la compatibilité $E - p$ de ce nouveau cluster avec la trace initiale. Si cette compatibilité satisfait la relation $\chi_{Ep}^2 < \chi_{Ep,cut}^2$, la procédure de *re-clustering* pour ce cluster pourra s'arrêter et choisir cette configuration de nouveaux clusters comme étant la configuration finale. Cela permet, dans le cas où la reconstruction est considérée comme particulièrement bien opérée, d'éviter d'exécuter des itérations supplémentaires inutiles.

La figure 6.20 a) montre un exemple de cluster comportant un excès en énergie dû à la présence d'un hadron neutre dans son voisinage. Après cette procédure, on peut voir que la compatibilité $E - p$ de la particule chargée s'est améliorée et a permis de reconstruire le hadron neutre dans un cluster séparé.

Re-clustering lors d'un déficit en énergie [MissingEnergyReclustering]

La principe est ici très similaire à celui de l'algorithme précédent. Nous traiterons ici des clusters chargés montrant un déficit en énergie comparé à sa trace associée. Ici encore, nous ne traiterons pas les clusters chargés sortant du détecteur ou les clusters chargés ayant plusieurs traces associées. Les clusters chargés en déficit d'énergie sont généralement reconstruits avec des clusters neutres dans leur voisinage, justifiant justement ce déficit. Cela peut être dû à une particule neutre ayant récupéré une partie importante du cluster chargé. Cela peut aussi être dû à la fragmentation du cluster chargé en plusieurs clusters mais que les algorithmes d'associations topologiques n'ont pas pu résoudre.

En plus du cluster chargé déficitaire en énergie devront être ajoutés d'autres clusters voisins durant la procédure de *re-clustering*, afin de pouvoir récupérer une part de l'énergie manquante de ces derniers. La sélection des clusters voisins au cluster chargé considéré se fait suivant les critères suivants :

- le cluster voisin doit être neutre,
- la plus courte distance entre les hits des deux clusters doit être inférieure à une certaine coupure [$\Delta < \Delta_{max}$],
- dans le cas où le cluster neutre ne passe pas la précédente coupure, une seconde coupure est considérée. L'angle d'ouverture $\theta_{cluster}$ défini comme l'angle entre la direction du cluster chargé et le vecteur joignant les barycentres des deux clusters, doit être inférieur à une certaine coupure.

Si aucun cluster neutre n'est sélectionné, le *re-clustering* n'est pas déclenché puisque aucune énergie supplémentaire ne pourra améliorer la compatibilité $E - p$ du cluster chargé.

La suite se déroule comme dans l'algorithme de *re-clustering* précédent. Les clusters neutres sélectionnés et le cluster chargé sont utilisés dans la phase de *re-clustering* itératif. La compatibilité $E - p$ du cluster chargé est minimisée à chaque itération. A la fin du *re-clustering*, la meilleure configuration est sélectionnée. A noter que les clusters neutres correspondants sont réinjectés dans la liste courante et peuvent participer aux éventuels *re-clustering* des autres particules chargées.

La figure 6.20 b) montre un exemple de cluster en déficit d'énergie dû à la présence d'un hadron neutre à proximité. L'application de cet algorithme de *re-clustering* va permettre de récupérer la bonne énergie du cluster chargé mais aussi celle du hadron neutre.

Re-clustering lors d'associations multiples [SplitTrackReclustering]

Si la configuration des algorithmes d'association entre traces et clusters précédemment exécutés a permis des associations multiples entre traces et clusters, il se peut que certains clusters aient plusieurs traces associées. Cet algorithme est dédié à ce type de situation et a pour but de redistribuer les contributions du cluster considéré parmi les différentes traces associées.

La figure 6.20 c) montre un exemple de ce type de situation. La redistribution des hits du clusters est basée sur la distance entre les hits et les traces considérés. Le hit considéré sera re-distribué à la trace la plus proche.

Préparation des traces pour la création de PFOs [TrackPreparation]

Après toutes les étapes de *clustering*, lorsque l'on considère que les particules reconstruites finales peuvent être créées, deux listes de traces et de clusters sont produites pour préparer la création des PFOs par un algorithme dédié en aval (voir "*Création de PFOs*" plus bas).

Cet algorithme prépare les traces utilisées lors des phases de *clustering* en amont. Une liste d'algorithmes d'association entre traces et clusters peut être exécuté en tant que dernières associations avant la création de PFOs. Cette étape permettra par exemple de gérer les dernières associations qui n'ont pas pu être créées par les précédents algorithmes.

Les traces sélectionnées sont celles présentant les critères suivants :

- les traces ayant des clusters associés ou de faible énergie ne pouvant donner lieu à la création de clusters dans les calorimètres,
- les traces ayant des traces filles satisfaisant le premier critère.

Préparation des clusters pour la création de PFOs [ClusterPreparation]

Tout comme l'algorithme précédent, cet algorithme prépare les clusters produits par les phases de *clustering* précédentes pour la création de PFOs en aval. Les clusters sélectionnés sont simplement ceux n'appartenant pas à un PFO.

Création de PFOs [PfoCreation]

Étape finale de la reconstruction, cet algorithme crée les particules reconstruites finales.

A partir de la liste de clusters créée par l'algorithme de "*préparation des clusters*", les PFOs neutres sont créés avec les clusters ne possédant pas de trace associée, un cluster par PFO. Les PFOs chargés sont quant à eux créés à partir de la liste de traces créées à partir de l'algorithme de "*préparation des traces*" et de la même liste de clusters.

A partir d'une trace donnée, un PFO est créé, contenant la liste de ses traces parentes et de toutes les traces filles associées à ces traces parentes. Pour chacune des traces considérées, les éventuels clusters associés sont ajoutés au PFO.

Pour les PFOs neutres, des premières coupures de sélection sont appliquées :

- les PFOs dont les clusters contenus dans un seul pseudo-plan sont supprimés,
- les PFOs ayant une énergie trop faible sont supprimés. L'énergie est calculée suivant le résultat des fonctions d'identification de particules. L'énergie des photons est l'énergie électromagnétique corrigée et celle des hadrons neutres l'énergie hadronique corrigée,
- les PFOs avec un faible nombre de hits sont supprimés.

Identification finale des particules [FinalParticleId]

Après la création de PFOs, les fonctions d'identification de particules sont exécutées une à une afin d'identifier chacune des particules. L'information est enregistrée dans les méta-données de chaque PFO de façon persistante.

Algorithme de *clustering* ArborPFA [ArborClustering]

Cet algorithme est un algorithme composite. Il ne résout pas de problème de topologie en soi. Son but est d'exécuter une liste de sous-algorithmes de connexions et nettoyage de connexions de hits. Une fois cette série de sous-algorithmes exécutée, l'algorithme attend une série de hits connectés avec une topologie en arbre. Les différents arbres sont identifiés et un cluster est créé pour chacun.

La configuration de cet algorithme permet en outre :

- de spécifier le type de hits à utiliser lors des connexions : les hits du calorimètre électromagnétique, hadronique, des chambres à muons, ou les trois,
- de discriminer les arbres composés uniquement d'un seul hit (pas de connexion).

Algorithme de *clustering* parent [ClusteringParent]

Tout comme l'algorithme précédent, l'algorithme de *clustering* parent est un algorithme composite. Son but est de lancer un algorithme de *clustering* et une série d'algorithmes d'associations topologiques.

Reconstruction des photons [PhotonReconstruction]

Cet algorithme est en tout point similaire à l'algorithme de *clustering* parent. A la fin de l'algorithme, la fonction d'identification des photons est exécutée sur chaque cluster créé. Les clusters identifiés comme étant des photons sont sauvegardés dans une liste dédiée et les autres clusters sont supprimés. La stratégie qui sera déployée par la suite consistera à exécuter un algorithme de *clustering* ArborPFA uniquement sur les hits du calorimètre électromagnétique. Les photons qui ne sont pas reconstruits par cet algorithme seront reconstruits dans l'algorithme principal et auront encore une chance d'être identifiés en fin de reconstruction.

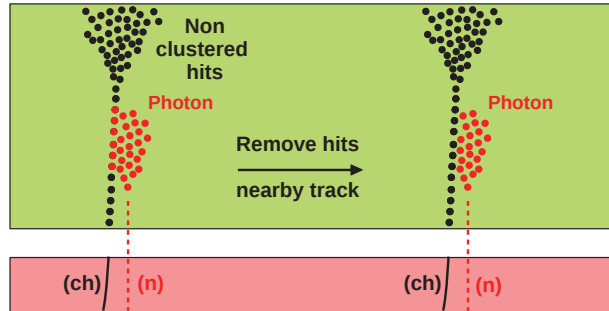


FIGURE 6.21 – Exemple d'un photon reconstruit comportant des hits sur les chemin d'une trace. Les hits en rouge sur le chemin de la trace sont supprimés du photon et remis à disposition pour un prochain *clustering*.

Suppression des hits sur le chemin des traces [NearbyTrackPhotonRemoval]

La figure 6.21 montre la situation traitée par cet algorithme. Lors de la reconstruction des photons, certains photons peuvent être reconstruits avec une particule chargée dans leur voisinage. Des hits appartenant notamment à la trace primaire de cette particule chargée, sont parfois assignés au photon. Ces hits sont particulièrement importants par la suite dans les algorithmes de clustering. Ils sont utilisés lors de la création des connexions en partant de la projection des traces. De ce fait, si les hits de la trace primaire sont assignés au photons, les connexions s'arrêteront avant d'arriver au début de la gerbe.

Le but de l'algorithme est de retirer les hits sur le chemin de toutes les traces disponibles dans l'événement appartenant aux photons. Les hits se trouvant à une distance inférieure à Δ_{track} d'une trace et appartenant à un photon sont supprimé du photon. Á la fin de l'algorithme, une coupure est appliquée sur l'énergie et sur le nombre de hits des photons. Si l'énergie électromagnétique est inférieure à E_{min} ou le nombre de hits inférieur à $N_{hit,min}$, alors le photon est supprimé.

La figure 6.21 montre un exemple où un photon reconstruit se voit retirer quelques hits près d'une trace. Ces hits permettront de reconstruire par la suite le cluster chargé voisin lors d'une phase de *clustering*.

3 Les performances physiques de l'algorithme

Tout comme pour l'algorithme PandoraPFA, les performances sont évaluées à partir d'événements di-jets produits à partir d'un boson Z^0 virtuel se désintégrant en une paire de quarks $q\bar{q}$. Les échantillons sont simulés grâce au logiciel de simulation numérique Mokka 08-04 [35] et reconstruits avec la chaîne de reconstruction du logiciel ILCSoft v01-17-06 [30]. La reconstruction finale est opérée grâce à l'algorithme de suivi de particules ArborPFA v02-05-01, dont le contenu algorithmique a été présenté dans la section précédente. L'extraction des performances est finalement réalisée avec le logiciel PandoraAnalysis v00-06.

Afin de mieux comprendre les performances extraites et les confusions pouvant

dégrader ces performances, il est possible de reconstruire certains types de particules en amont de l'algorithme de suivi de particules en utilisant les informations Monte Carlo et de reconstruire le reste de l'événement avec l'algorithme de suivi de particule. Par exemple, on peut demander à reconstruire les photons en amont. On parlera dans ce cas de "reconstruction parfaite des photons". Par extension, il sera aussi possible de reconstruire chaque événement avec les informations Monte Carlo de toutes les particules. On parlera dans ce cas de "reconstruction parfaite". Par la suite, les performances de l'algorithme seront comparées avec une "reconstruction parfaite de photons" et une "reconstruction parfaite". À noter que dans tous les cas, les mêmes corrections en énergie sont appliquées, aussi bien sur les particules parfaitement reconstruites que sur les particules reconstruites par ArborPFA.

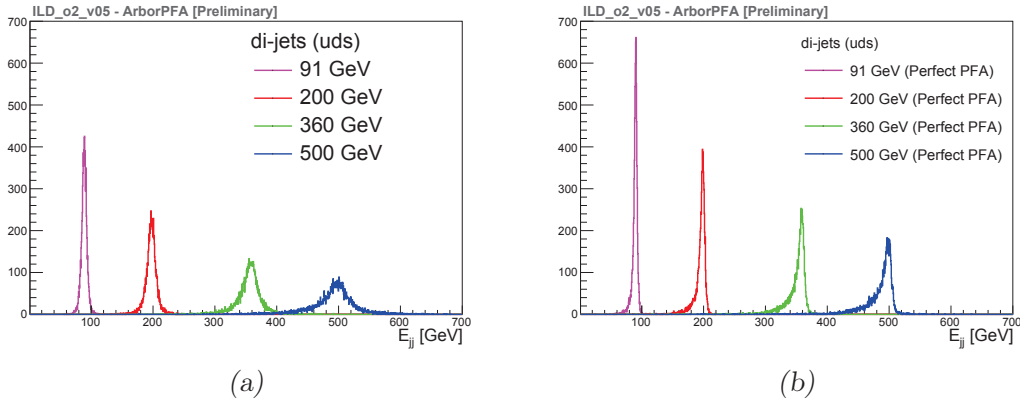


FIGURE 6.22 – Distribution de l'énergie totale reconstruite pour des échantillons di-jets de 91, 200, 360 et 500 GeV reconstruits avec (a) l'algorithme ArborPFA v02-05-01 et (b) une reconstruction parfaite.

Les figures 6.22(a) et 6.22(b) montrent respectivement l'énergie totale reconstruite avec l'algorithme ArborPFA et une reconstruction parfaite. Afin de mieux comparer ces distributions, la moyenne et la déviation à la linéarité de ces différentes distributions sont montrées sur la figure 6.23. Ces moyennes sont déterminées par la méthode du RMS_{90} . On y trouve aussi les énergies reconstruites moyennes pour la "reconstruction parfaite des photons".

La résolution relative de l'énergie des jets peut être déterminée à partir de la méthode du RMS_{90} en utilisant la distribution en énergie totale du système di-jets :

$$\frac{\sigma_j}{E_j} = \frac{\sigma_{jj}/\sqrt{2}}{E_{jj}/2} = \sqrt{2} \cdot \frac{\sigma_{jj}}{E_{jj}} \quad (6.31)$$

avec σ_{jj} la résolution en énergie totale et E_{jj} l'énergie totale du système di-jets, tout deux déterminés par la méthode du RMS_{90} . La figure 6.24 montre la résolution relative de l'énergie des jets en fonction de l'énergie du boson Z^0 virtuel.

Les différents panneaux de la figure 6.25 montrent la résolution relative de l'énergie des jets en fonction de $\cos\theta$. Ces résultats ne respectent pour le moment pas les performances de référence pour la reconstruction des jets dans l'ILC

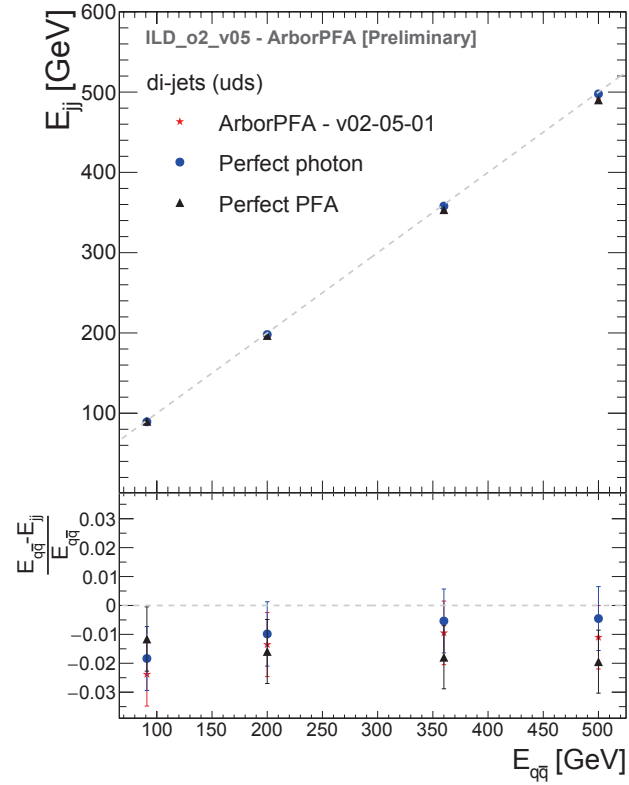


FIGURE 6.23 – Énergie moyenne et déviation à la linéarité du système di-jets uds en fonction de l'énergie du boson Z^0 virtuel pour différents types de reconstructions.

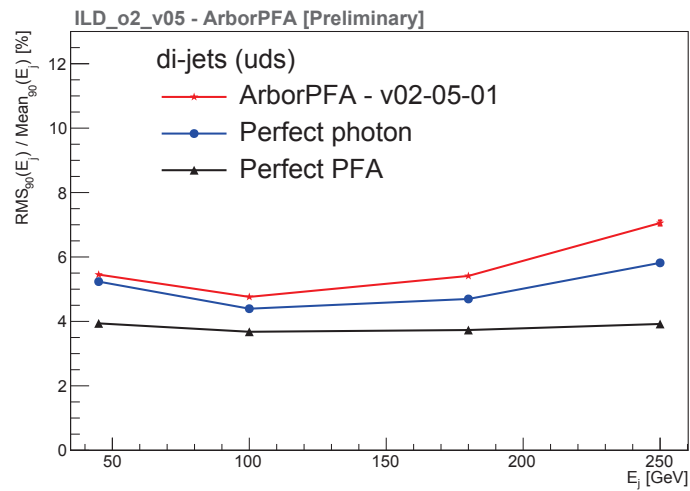


FIGURE 6.24 – Résolution relative en énergie des jets en fonction de l'énergie du boson Z^0 pour différents types de reconstructions.

($\sigma_j/E_j \sim 3 - 4\%$). Des améliorations pourraient être apportées aux algorithmes d'ArborPFA et aux différents calculs en énergie effectués au sein de l'algorithme.

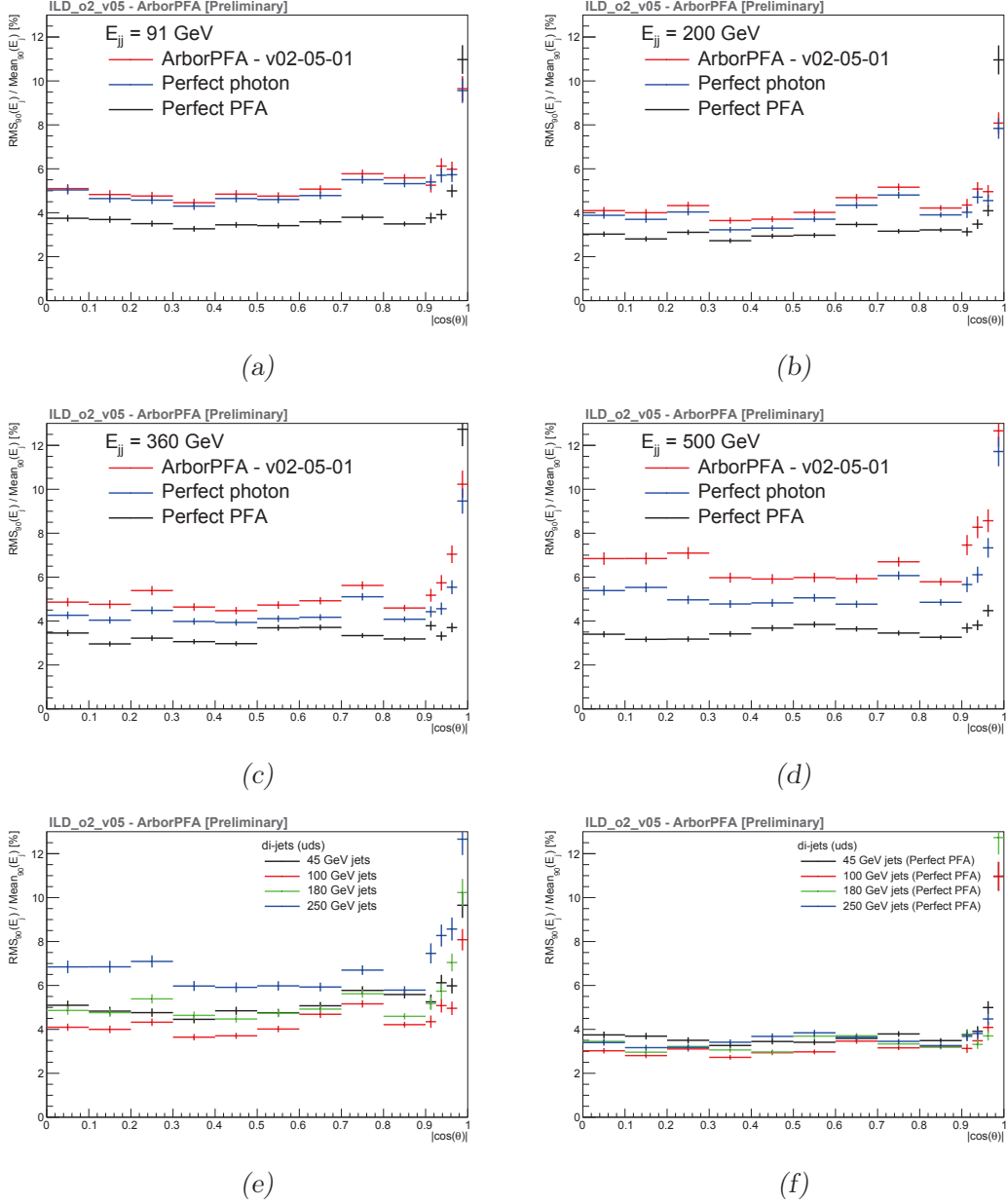


FIGURE 6.25 – Résolution relative de l'énergie des jets en fonction de $\cos\theta$ pour un boson Z^0 virtuel ayant une énergie de (a) 91 GeV, (b) 200 GeV, (c) 360 GeV et (d) 500 GeV, pour différents types de reconstructions. Les figures (e) et (f) montrent la même variable pour différentes énergies de jets, respectivement après reconstruction avec l'algorithme ArborPFA et une reconstruction parfaite.

4 Conclusion

Une seconde implémentation de l'algorithme ArborPFA a été développée pour le détecteur ILD_o2_v05 utilisant le SDHCAL comme calorimètre hadronique. Une calibration en énergie des hadrons isolés a, dans un premier temps, permis d'obtenir une linéarité et une résolution satisfaisantes pour la combinaison du calorimètre électromagnétique et du SDHCAL. Des chutes en énergie importantes ont cependant été observées près des interstices entre les différents modules du tonneau central du détecteur et ont donné lieu à une première correction en énergie. D'autre part, une deuxième décroissance en énergie a été observée en fonction de l'angle θ dans le détecteur, donnant finalement lieu à une deuxième correction en énergie. La linéarité et la résolution relative en énergie obtenues sont satisfaisantes dans la région du tonneau central et dans les bouchons mais restent encore insuffisantes dans la région intermédiaire entre ces deux parties du détecteur.

Les différents sous-algorithmes de cette deuxième version ont ensuite été détaillés. Parmi les grands changements avec la première version figurent notamment l'ajout d'algorithmes de *re-clustering* utilisés pour optimiser la compatibilité entre l'impulsion des traces et l'énergie de leurs clusters associés, l'ajout de fonctions d'identification de particules, ou encore d'algorithmes de connexions supplémentaires dédiés aux particules chargées dans un champ magnétique. Tout comme pour l'algorithme PandoraPFA, les performances ont été évaluées à partir d'échantillons de Z^0 virtuel se désintégrant en deux jets de même énergie et de directions opposées. Les résultats obtenus sur la résolution relative en énergie des jets ne respectent pour le moment pas les références de performances requises par l'ILC ($\sigma_j/E_j \sim 3-4\%$) et devront faire l'objet d'une étude plus approfondie.

Parmi les études supplémentaires, il serait intéressant d'estimer les efficacités de reconstruction et de séparation pour chaque type de particule. Ceci permettrait, par exemple, de mieux comprendre l'origine des confusions observées par les différents types d'algorithmes "parfaits" utilisés dans l'étude des di-jets. Les résultats présentés sur les corrections en énergies ont montré que les corrections appliquées n'étaient pas suffisantes, notamment dans la région de transition entre le tonneau central et les bouchons. Des corrections supplémentaires pourraient améliorer la résolution relative en énergie des jets dans cette région. Au niveau des algorithmes, certains cas d'associations topologiques sont encore absents de l'implémentation, comme par exemple la gestion de la rétrodiffusion au sein des gerbes hadroniques.

Chapitre 7

Logiciel de surveillance de qualité de données en ligne

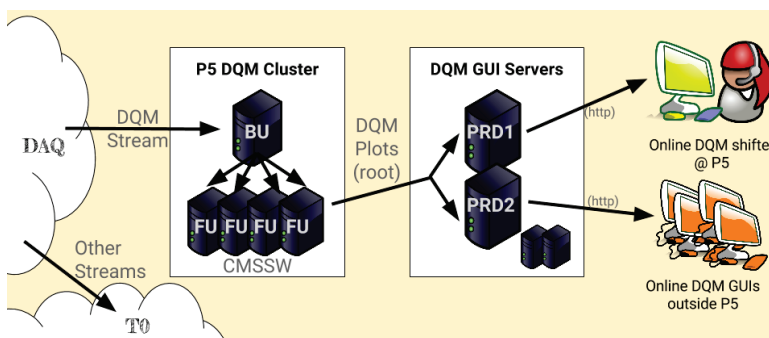
LES campagnes de prises de données du prototype CALICE SDHCAL au CERN au cours des dernières années ont montré des performances compatibles avec celles désirées pour un calorimètre hadronique dans l'ILD. Cependant, afin de s'assurer de la bonne qualité des données durant ces campagnes, il est nécessaire de mettre en place un système de surveillance du détecteur. C'est dans ce cadre que le logiciel DQM4HEP a été développé. Après une introduction aux systèmes de surveillance en physique des hautes énergies, l'architecture, les composants et le système de visualisation du logiciel DQM4HEP seront présentés. Enfin l'implémentation des analyses en ligne lors du test sur faisceau combiné du CALICE SDHCAL avec le CALICE SiW-ECal sera exposée.

Contenu

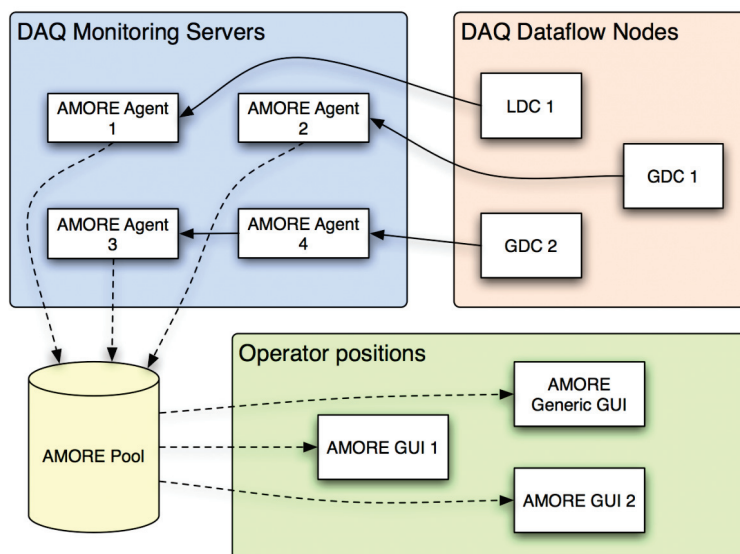
1	Introduction aux systèmes de surveillance en physique des hautes énergies	154
2	DQM4HEP : un environnement générique de surveillance de données en ligne	155
2.1	Vue d'ensemble du logiciel	155
2.2	Gestionnaire de <i>runs</i>	159
2.3	Lien vers les sources de données	160
2.4	Collecteur d'événements physiques	162
2.5	Modules d'analyse de données	164
2.6	Collecteurs d'éléments de surveillance	170
2.7	Les interfaces graphiques	171
3	Implémentation pour le prototype SDHCAL	177
3.1	Combinaison des détecteurs SiWECal et SDHCAL	177
3.2	Source de données	178
3.3	Les modules d'analyse	179
3.4	Déploiement du système et performances	184
4	Conclusion	188

1 Introduction aux systèmes de surveillance en physique des hautes énergies

Le suivi de la qualité des données en ligne lors d'une prise de données avec un détecteur est d'une importance cruciale. C'est une première étape vers la certification des données qui seront ensuite analysées hors ligne. De tels systèmes ont déjà été développés et notamment en physique des hautes énergies au LHC, où la surveillance de la qualité des données et du bon fonctionnement des détecteurs est primordiale. Ces systèmes doivent fournir les fonctionnalités suivantes i) fournir un sommaire sur le bon fonctionnement des détecteurs, ii) évaluer la qualité des données prises par ces détecteurs, iii) identifier rapidement des problèmes éventuels pour permettre aux opérateurs de les résoudre.



(a)



(b)

FIGURE 7.1 – Architecture logicielle du système de surveillance des expériences (a) CMS et (b) ALICE. On y retrouve les principaux ingrédients des systèmes de surveillance de données en physique des hautes énergies.

La plupart des expériences qui développent ce type de système présentent dans leurs logiciel un fort couplage à la structure des événements (*Event Data Model*). C'est par exemple le cas des expériences CMS et ALICE, qui reçoivent dans leur module d'analyse une structure de données propre à l'expérience. La figure 7.1 montre l'architecture logicielle des environnements d'analyse en ligne des expériences CMS et ALICE.

On y trouve les principaux ingrédients des systèmes de surveillance en physique des hautes énergies, à savoir :

- une interface vers le système d'acquisition,
- des applications d'analyse en ligne,
- des collecteurs d'histogrammes,
- des interfaces client graphiques/web de visualisation d'histogrammes.

Ces différentes parties sont généralement implémentées dans des processus indépendants et liées par des communications réseaux. Ce choix permet de répartir la charge de calcul sur les différentes machines et de ne pas impacter sur la vitesse de la prise de donnée par le système d'acquisition. Elle permet aussi de fournir la modularité nécessaire pour paralléliser les analyses de données sur plusieurs cœurs d'une seule machine et/ou sur plusieurs machines au sein d'un *cluster*. Le processus gérant la collection des histogrammes est une application serveur basée sur le paradigme publication/souscription. Cela permet à plusieurs clients de visualiser les histogrammes en même temps sur un ou plusieurs ordinateurs.

2 DQM4HEP : un environnement générique de surveillance de données en ligne

2.1 Vue d'ensemble du logiciel

Le logiciel implémente les fonctionnalités communes aux systèmes de surveillance de données en ligne. La figure 7.2 en montre l'architecture globale. On y retrouve certains ingrédients des systèmes de surveillance des expériences CMS et ALICE : distribution des événements, traitement des données, distribution des histogrammes et visualisation de ces derniers.

Deux processus importants assurent le lien avec le système d'acquisition (*DAQ*). Le processus responsable de la reconstruction des événements (*DQM Event builder*) permet de récupérer les données issues de l'acquisition. Il utilise la mémoire partagée comme espace d'échange de façon à ne pas perturber le système d'acquisition. Le deuxième processus important assure la réception des signaux de début et fin de *runs* (*Run control server*) envoyés par le système d'acquisition. Les signaux sont envoyés par message HTTP POST, les bibliothèques HTTP client étant standards dans la plupart des langages. Cela permet de limiter les dépendances logicielles dans le système d'acquisition vis-à-vis du système de surveillance.

L'ensemble des processus déployés pour une expérience donnée est géré à distance par des applications serveurs de gestion de processus (*Job control server*).

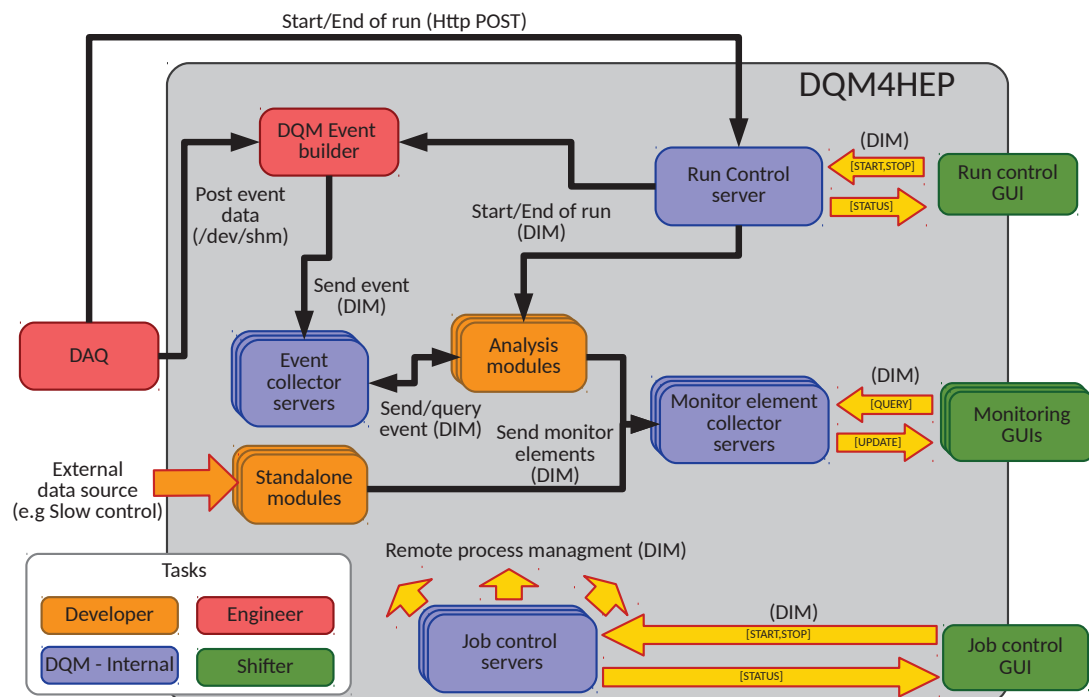


FIGURE 7.2 – Architecture globale du logiciel DQM4HEP : des sources de données provenant du système d'acquisition (gauche) au système de visualisation (droite).

Ces serveurs sont pilotés par l'intermédiaire d'une ou plusieurs application(s) graphique(s) (*Job control GUI*) permettant le démarrage, l'arrêt ou la récupération de l'état d'un processus sur un hôte donné en temps réel.

Le reste des applications assure la collection des données (*Event collector servers*), la collection des éléments de surveillance (*Monitor element collector servers*) et le traitement des données en ligne (*Analysis modules* et *Standalone modules*).

Le principe de ce logiciel est de fournir les fonctionnalités nécessaires aux physiciens pour analyser les données issues de leurs détecteurs en se focalisant uniquement sur l'analyse elle-même. Les seules parties qu'un développeur doit fournir pour faire fonctionner le système sont les suivantes :

- la définition du type d'événement à traiter,
- la conversion des données issues du système d'acquisition,
- l'implémentation des analyses de données en ligne.

Au niveau de la gestion des distributions, le logiciel est séparé en plusieurs sous-paquets. La figure 7.3 en montre l'organisation. Les fonctionnalités du système de surveillance sont implémentées dans les paquets DQMCore, DQMviz et DQM4ILC et dépendent de 6 autres paquets (dont deux optionnels) liés à la sérialisation des données et à l'implémentation des communications réseaux.

Coté sérialisation, on trouve :

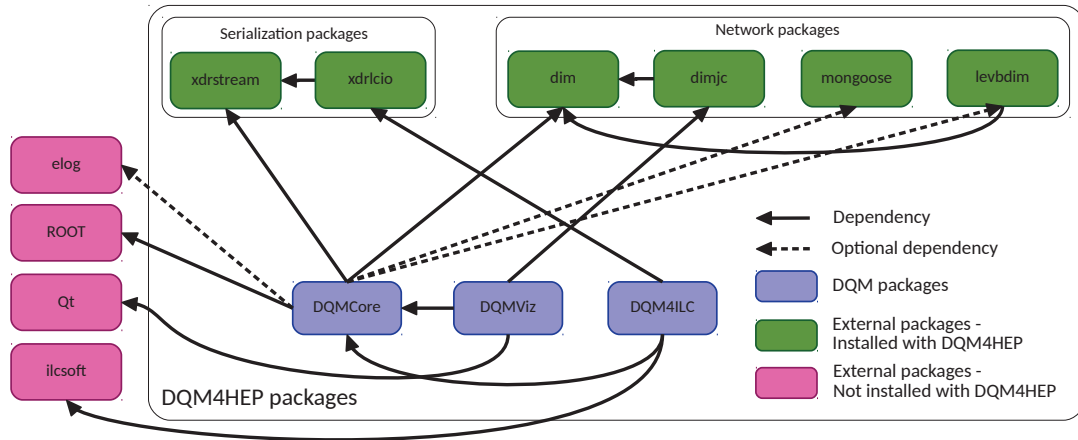


FIGURE 7.3 – Schéma illustrant l'organisation des paquets du logiciel DQM4HEP : les trois paquets principaux de DQM4HEP (en bleu), les paquets installés avec DQM4HEP (en vert) et les paquets externes (en rose).

- `xdrstream`, paquet fournissant des interfaces de sérialisation de données dans une représentation indépendante de l'architecture de la machine. Il permet aussi l'écriture et la lecture de données dans différents descripteurs de fichiers (mémoire tampon, fichier binaire, socket, etc...) de façon transparente pour l'utilisateur. Il est à noter que ce paquet a été développé pour les besoins du système DQM4HEP mais de manière suffisamment générique pour être réutilisable dans n'importe quel autre contexte.
- `xdricio`, paquet basé sur `xdrstream` implémentant la sérialisation des données au format LCIO [7], format de fichier utilisé par la collaboration LCC. Il sera utilisé pour sérialiser les événements physiques et les transmettre par communication réseau afin de traiter les événements issus du prototype technologique du SDHCAL (voir 3).

D'autre part, pour la partie réseau, on trouve les paquets suivants :

- DIM (*Distributed Information Management*), une librairie légère et portable développée par la collaboration DELPHI [27]. Elle fournit une couche de communication inter-processus via l'utilisation de socket TCP/IP. Elle sera utilisée pour gérer toutes les communications entre les processus au sein du logiciel DQM4HEP.
- DIMJC (*DIM Job Control*), basé sur DIM et fournissant un système de gestion de processus à distance. Très léger, le paquet consiste en un exécutable serveur créant des processus à la demande sur la machine sur laquelle il est exécuté et une interface client DIM envoyant les requêtes de création de ces processus à distance à partir d'un poste client. Les différentes applications de DQM4HEP fonctionnant toutes de façon indépendante et liées par réseau, il devient alors possible avec un tel système de superviser l'ensemble des processus lors d'un déploiement pour une expérience donnée.

- `levbdim` (optionnel), lui aussi basé sur DIM et fournissant une reconstruction des événements en utilisant la mémoire partagée. Ce paquet est utilisé par le système d'acquisition du prototype technologique du SDHCAL pour récolter les données brutes mais aussi pour transmettre les données au système de surveillance. Son implémentation sera discutée plus en détails dans la section 2.3.
- `mongoose` (optionnel), librairie cliente HTTP. Très légère, elle est considérée comme un standard dans ce domaine.

Le paquet `DQMCORE` implémente les fonctionnalités principales du logiciel de surveillance, à savoir :

- gestion des *plugins*,
- interface vers les événements physiques,
- gestion des éléments de surveillance et des tests de qualité,
- interface vers le système d'acquisition,
- applications d'analyse de données,
- collecteurs d'événements et d'éléments de surveillance,
- interface client vers les différents types de collecteurs,
- gestion des *runs*.

Le cœur du logiciel est basé sur le gestionnaire de *plugins*, fonctionnant sur la base de seulement deux classes et une macro pré-processeur. Ce sous-système du logiciel permet d'importer des classes utilisateurs à partir de bibliothèques externes. Plus de détails sur le fonctionnement de ce système sont donnés dans l'annexe B.

Une deuxième spécificité importante du logiciel est le modèle d'événements utilisé pour le traitement des données. Contrairement aux logiciels existants, DQM4HEP ne repose pas sur un modèle d'événements particulier. La figure 7.4 montre le diagramme de classe UML relatif au modèle d'événements. Deux interfaces abstraites permettent :

- d'englober un type d'événement utilisateur particulier (`DQMEVENTBASE<T>`)
- de définir la sérialisation pour ce type d'événements (`DQMEVENTSTREAMER`)

Cette architecture présente plusieurs avantages. Premièrement, elle permet de manipuler un événement sans connaître à priori son vrai type. L'utilisation de l'événement sous-jacent stocké dans l'objet qui l'englobe, se fait au moment de l'appel des méthodes. Deuxièmement, l'interface `DQMEVENTSTREAMER` fournit une fabrique [26] à événements et les fonctions de sérialisation pour un type d'événements donné. En combinant cette architecture avec le gestionnaire de *plugins*, il devient alors possible d'insérer dans le système tout type d'événements utilisateur, tout en définissant la façon dont ils doivent être écrits et lus par le système. C'est là où repose toute la généralité du logiciel DQM4HEP.

Le paquet `DQMVIS` regroupe les interfaces graphiques utilisateurs. Implémenté avec Qt [40] et basé sur `DQMCORE` et `dimjc`, il fournit trois interfaces graphiques :

- une interface client graphique vers un gestionnaire de *run*.

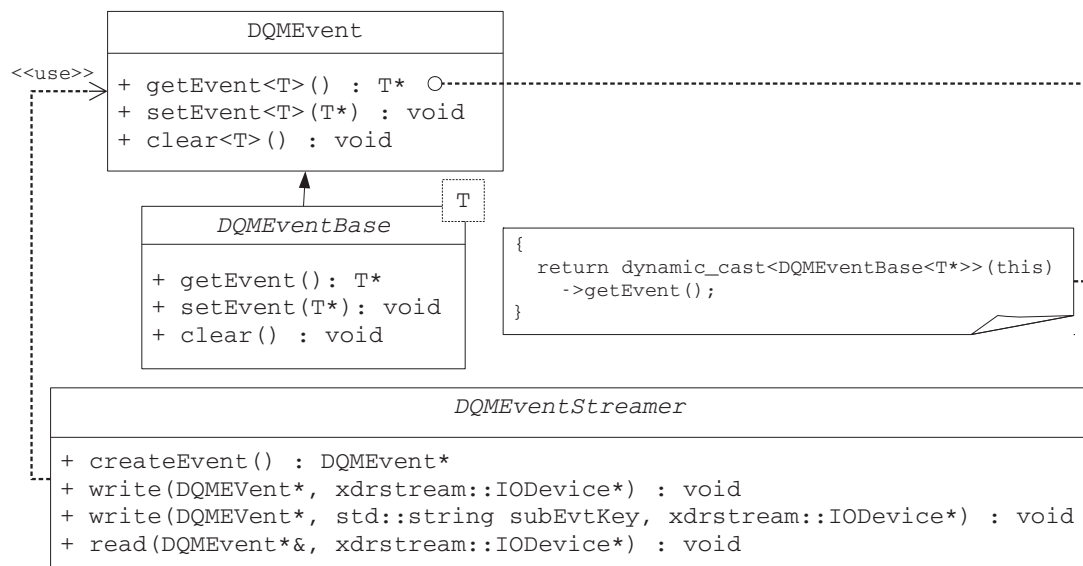


FIGURE 7.4 – Diagramme de classe UML illustrant l’organisation des classes relatives aux événements. La classe `DQMEventStreamer` définit une interface pour la sérialisation des objets de type `DQMEvent`.

- une interface client graphique vers les différents serveurs de gestion de processus.
- une interface client graphique d’affichage d’éléments de surveillance (histogrammes, scalaires, textes, etc ...).

Une description plus détaillée sera fournie dans la section 2.7.

Le troisième paquet relatif au système de surveillance, `DQM4ILC`, a été développé spécifiquement pour l’expérience ILC. Il fournit plusieurs implémentations d’interfaces de `DQM4HEP`, notamment la sérialisation des événements au format LCIO [7] au sein du logiciel, ainsi que quelques outils en ligne de commande. Un exécutable permettra notamment de publier des événements à partir d’un fichier LCIO vers un collecteur d’événements physiques.

L’ensemble de tous ces paquets est regroupé dans un méta-paquet gérant le processus d’installation avec leur dépendance : `DQM4HEP`.

2.2 Gestionnaire de *runs*

En physique des hautes énergies, on appelle *run* une période stable pendant laquelle le système d’acquisition enregistre des données. Le numéro de *run* identifie chaque prise de données. D’autres paramètres peuvent être utilisés pour identifier un *run*, tel que le nom du détecteur ou encore l’énergie du faisceau. Le gestionnaire de *run* développé dans `DQM4HEP`, est une application serveur permettant de récupérer, par plusieurs méthodes possibles, les signaux de début et fin de *run* ainsi que les paramètres associés provenant du système d’acquisition :

1. En utilisant un POST HTTP.
2. En utilisant l'interface client fournie par DQM4HEP.
3. En utilisant les outils de ligne de commande.
4. En utilisant la page web fournie par le serveur de gestion de *run*.

La première méthode a l'avantage de supprimer la dépendance de la librairie DQM4HEP dans l'application qui exécute le POST, contrairement à la deuxième pour laquelle des portions de code devront être implantées dans l'application. Les deuxième et troisième méthodes nécessitent quant à elles l'installation du logiciel DQM4HEP sur le PC exécutant la commande. La dernière méthode est très utile pour faire fonctionner le logiciel sans système d'acquisition lors des tests. Le client se connecte directement au gestionnaire de *run* par requête HTTP et le pilote via une page web. Cependant, dans un vrai contexte d'acquisition de données, il faudra copier les paramètres du *run* démarré par le système d'acquisition et en démarrer un spécifiquement pour les processus reliés au système de surveillance.

2.3 Lien vers les sources de données

Un pré-requis important pour les logiciels de surveillance en physique des hautes énergies est de ne pas impacter la vitesse d'enregistrement des données par la DAQ. Le système d'acquisition doit pouvoir recevoir des données issus des différents sous-détecteurs, les reconstruire, les écrire sur disque et les transmettre au système de surveillance de façon efficace sous peine de rater des événements physiques. Un tel système ne doit pas non plus dépendre du logiciel de surveillance utilisé lors d'une prise de données.

C'est dans cet esprit que le logiciel `levbdim` a été conçu. Basé sur DIM au niveau des communications réseau, il fournit les fonctionnalités nécessaires à la collecte des données depuis différentes sources en utilisant la mémoire partagée. La figure 7.5 illustre le flux de propagation des données, à partir des sources de données des différents sous-détecteurs jusqu'à leur envoi au système de surveillance en passant par leur écriture sur disque.

Pour `levbdim`, un système d'acquisition est composé d'un nombre fixe de source de données N_s qui publient leurs données de manière fréquente via l'interface réseau fournie par DIM. Ces données sont postées dans la mémoire partagée (`/dev/shm/levbdim/`) dès leur réception (DAQ, en haut sur la figure 7.5), un fichier par source de données. Les paquets de données sont alors identifiés à l'aide de trois identifiants :

- l'identifiant de source qui publie la donnée,
- l'identifiant du sous-détecteur qui publie la donnée,
- l'identifiant d'événement servant de clé commune pour identifier un événement complet.

Un programme de collecte des événements scanne en continu la mémoire partagée à la recherche des N_s sources de données ayant le même identifiant d'événements

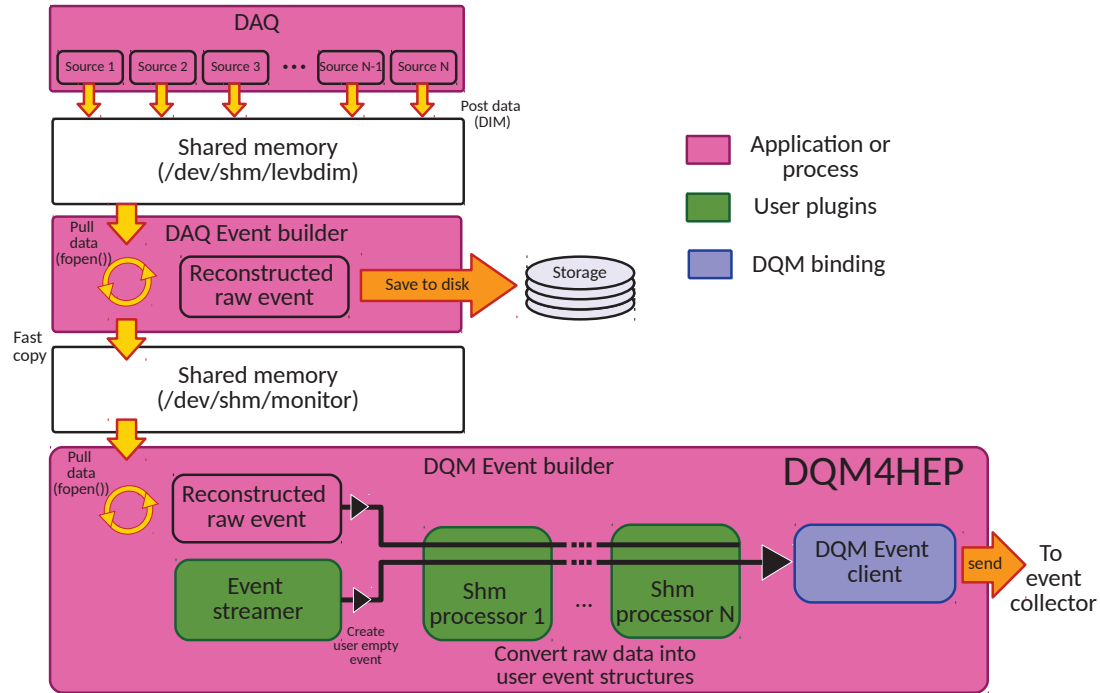


FIGURE 7.5 – Schéma illustrant la reconstruction des événements à partir des données brutes provenant du système d’acquisition et leur traitement au sein du logiciel de surveillance.

(*DAQ Event builder*). Une fois les N_s sources de données collectées pour un événement donné, le contenu de chaque fichier est passé à l’utilisateur sous forme de mémoire tampon pour qu’il soit traité. Dans le cas de la DAQ, ces données sont écrites sur disque avant d’être dupliquées de façon rapide dans un autre dossier de la mémoire partagée, appelé `/dev/shm/monitor/`. L’utilisation de la mémoire partagée et de la copie des fichiers gérée par la *libc* de chaque source de données permet alors d’optimiser leur temps d’échange entre le système d’acquisition et le système de surveillance.

C’est là que le logiciel de surveillance DQM4HEP entre en jeu. Une application similaire à celle de la DAQ collecte et reconstruit les événements issus de la mémoire partagée. Cette application est basée sur i) l’utilisation d’un `DQMEventStreamer` utilisé pour créer les événements et les écrire dans une mémoire tampon, ii) l’utilisation d’une liste ordonnée de modules de conversion des données (classe `DQMShmProcessors`) à partir des données brutes reçues par le système d’acquisition dans le format que l’utilisateur a défini dans i). La principale fonction de rappel de ces modules prend conjointement en paramètre la liste des données brutes et un événement utilisateur alloué par le `DQMEventStreamer` dans l’attente d’une conversion des données. Usuellement, il convient de définir une instance de module par détecteur, le décodage des données brutes y étant fortement lié. Une fois les données converties par l’utilisateur, elles sont envoyées aux collecteurs d’événements grâce à une interface client dédiée.

L'insertion du code utilisateur est réalisée par le gestionnaire de *plugins* en considérant les objets de type `DQMEventStreamer` et `DQMShmProcessor` comme des *plugins* compilés dans une ou plusieurs librairie(s) utilisateur. Cela apporte la souplesse nécessaire vis-à-vis d'un changement dans le code de conversion des données par l'utilisateur. L'application est quant à elle configurée à partir d'un fichier XML fournie par l'utilisateur.

2.4 Collecteur d'événements physiques

Dans le but d'analyser la grande quantité de données issue d'un système complexe de détecteurs, il est nécessaire d'exécuter plusieurs analyses différentes. Ces analyses peuvent devenir nombreuses si le complexe de détecteurs devient ... complexe ! Elles doivent être exécutées dans des processus séparés mais aussi sur différents ordinateurs pour répartir la charge de calcul. Pour que plusieurs processus puissent analyser des événements, un point d'accès commun aux données est nécessaire. Le paradigme de publication/souscription s'impose alors. La séparation des analyses sur plusieurs ordinateurs suggère une ou plusieurs application(s) centrale(s) de collecte d'événements communiquant sur l'interface réseau pour transmettre les événements aux analyses.

Le collecteur d'événements tel qu'il est implémenté dans le paquet `DQMCORE`, présente les fonctionnalités minimales pour assurer le lien entre le système d'acquisition et les analyses de données en ligne. La figure 7.6 illustre les interactions entre un collecteur d'événements et deux types de clients : le client dit *pusher* envoyant les événements aux collecteurs et le client dit de *réception*, exécutant des requêtes au collecteur pour récupérer les événements.

Les deux types de client utilisent la même interface pour communiquer avec le collecteur. Au niveau du client *pusher*, la seule fonction utilisée est la fonction `sendEvent()`, permettant tout simplement d'envoyer un événement au collecteur. La partie de l'interface relative au client de *réception* est un peu plus complexe. Pour récupérer les événements au collecteur, le client dispose de deux méthodes :

- La méthode dite de *requête*. Le client demande directement le dernier événement en date au collecteur, avec un délai maximum de N secondes pour la requête.
- La méthode dite de *mise à jour*. Le client souscrit au collecteur pour recevoir une mise à jour dès qu'un nouvel événement remplace l'ancien. Dans ce cas, un système de queue est utilisé du côté client pour stocker les événements reçus. Le client accède alors aux événements en récupérant le premier événement issu de la queue.

En utilisant la seconde méthode, il est aussi possible d'utiliser une fonction de rappel asynchrone pour récupérer l'événement directement au moment de la mise à jour. Cependant, la méthode de stockage via la queue comporte un avantage. Si le processus client est occupé à traiter l'événement, l'événement suivant sera ajouté dans la queue. Dans le cas de la fonction de rappel, la réception de l'événement sera mise en attente au niveau réseau, le temps que le processus client finisse de

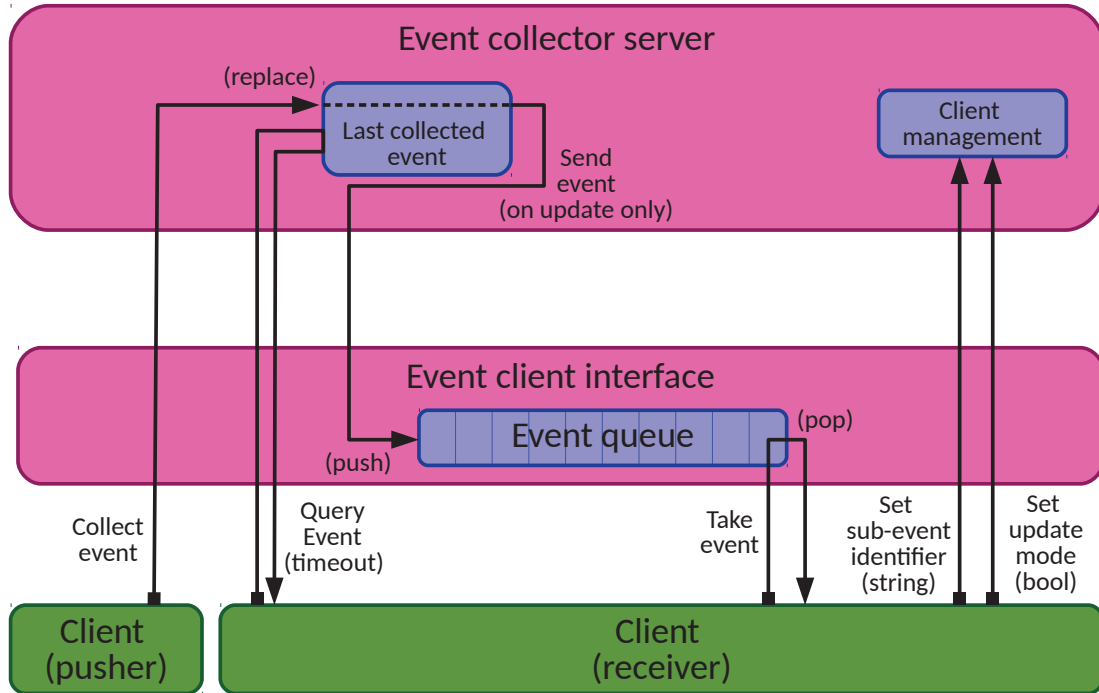


FIGURE 7.6 – Schéma illustrant les interactions entre un collecteur d'événements et ses clients. Deux type de clients : le *pusher* qui envoie les événements au collecteur et le client dit de *réception* qui envoie des requêtes d'accès. Les deux types de client utilisent la même API.

traiter le précédent. Ce système de queue est notamment très pratique en périodes de test sur faisceau, durant lesquelles le faisceau de particules incident ne fonctionne pas en continu mais par période de courte durée (\simeq quelques secondes), périodes elles-mêmes espacées de quelques dizaines de secondes. Les données prises durant la période de fonctionnement du faisceau se retrouve alors stockées dans la queue côté client et seront traitées en différé. La queue possède une taille limite, fixée par l'utilisateur, et permet d'éviter les débordements de mémoire. Quand la taille limite est atteinte et qu'un nouvel événement est inséré, l'événement en tête de queue est supprimé.

Une option supplémentaire pour les deux types de clients est aussi de pouvoir spécifier une clé de sous-événement. Si une clé est spécifiée, seule la sous-partie demandée sera renvoyée au client. Dans le cas de la première méthode de réception, la clé est spécifiée au moment de l'envoi de la requête. Dans l'autre cas, une fonction permet de spécifier au serveur la clé de sous-événement qui devra être utilisée de manière permanente au moment de la mise à jour. Par exemple, pour un événement contenant des informations de température, de pression, de courant et de tension, la clé "T:P" permettra de sérialiser uniquement la température et la pression dans l'événement avant l'envoi.

2.5 Modules d'analyse de données

Afin d'analyser les données provenant des détecteurs, un choix d'architecture multi-collecteurs d'événement et multi-applications d'analyse de données s'est imposé. Ces applications d'analyse sont conçues pour :

- recevoir des données,
- analyser et réduire la taille des données à quelques objets (histogrammes, scalaires, textes, etc ...),
- évaluer la qualité des données,
- alerter les utilisateurs d'un problème mineur ou critique durant la prise de données.

Les trois derniers points sont fortement reliés à la connaissance des détecteurs. Les analyses deviennent alors spécifiques pour chaque détecteur ou sous-détecteur du système. De telles applications doivent laisser la possibilité à l'utilisateur d'y insérer sa propre analyse. Cependant, effectuer un suivi des détecteurs à partir de la prise de données n'est pas toujours suffisant. Des informations, de type température ou pression par exemple, qui ne sont généralement pas jointes aux données, peuvent s'avérer cruciales pour ce suivi.

Ainsi deux interfaces utilisateurs ont été développées au sein de deux applications différentes. La première, encapsulée dans la classe `DQMAnalysisModule`, permet à l'utilisateur de recevoir les données provenant des détecteurs à partir d'un collecteur d'événements. La prise de données étant reliée au gestionnaire de *runs*, cette interface recevra aussi les signaux de début et de fin de *runs*. La deuxième interface, encapsulée dans la classe `DQMStandaloneModule`, se montre plus généraliste. L'utilisateur ne reçoit pas de données du détecteur, ni les signaux du gestionnaire de *runs*. Elle laisse la possibilité de choisir sa source de donnée à traiter. Elle est, par exemple, très bien conçue pour effectuer des relevés de température, de pression ou encore de haute-tension dans certains détecteurs.

Les figures 7.7 et 7.8 illustrent les différents composants des deux types d'applications présentes dans le logiciel. Dans le cas des modules d'analyse de données (figure 7.8), les signaux provenant du gestionnaire de *runs* et les événements issus du collecteur sont directement transmis par l'application au module utilisateur. Toute opération utilisateur, pour les deux types de module, s'effectuera par le biais d'une *API*, par exemple, pour exécuter des tests de qualité (voir section 2.5.1). Elle fournit des opérations utiles à l'utilisateur pour i) accéder à l'état de l'application, ii) créer, supprimer, lister ou naviguer dans une structure en répertoire, iii) allouer, supprimer ou accéder à des éléments de surveillances, iv), enregistrer, créer ou supprimer des tests de qualité, v) envoyer des alertes.

La figure 7.9 illustre, par un diagramme d'état UML, les différents étapes importantes des deux types d'applications lors de leur exécution. Une partie importante de ces étapes est commune : initialisation de l'application, structure en cycle (voir section 2.5.2), envoi des éléments de surveillance aux collecteurs, archivage optionnel des éléments et fin de l'application. Vient s'intercaler, pour l'application analysant les données du collecteur d'événement (diagramme de droite), la réception

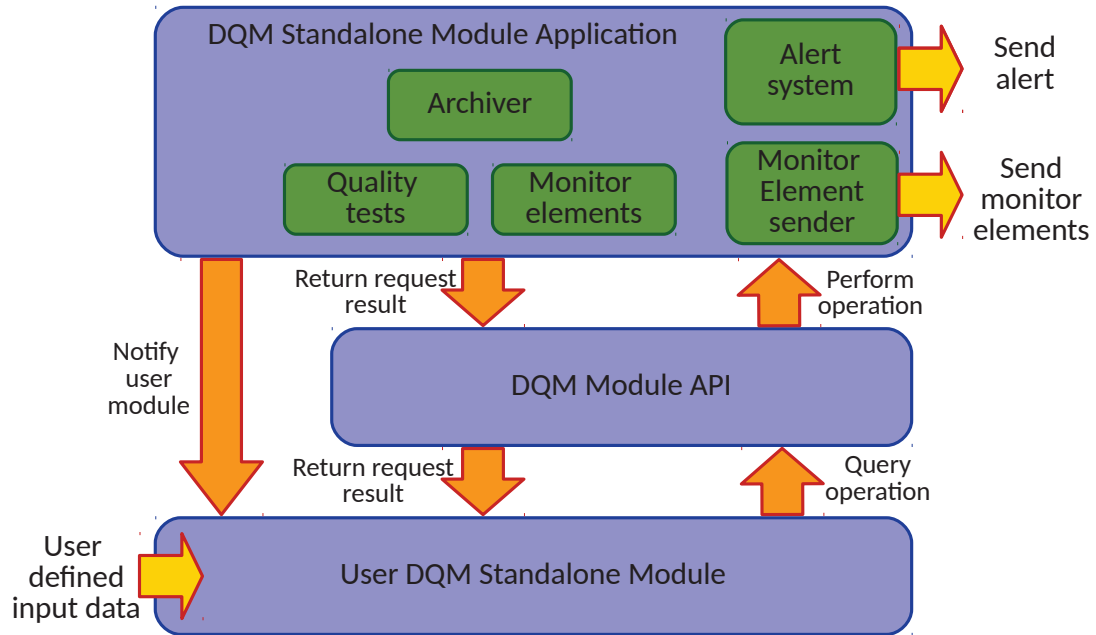


FIGURE 7.7 – Schéma illustrant l'interface utilisateur des modules d'analyse de données en ligne (application standalone).

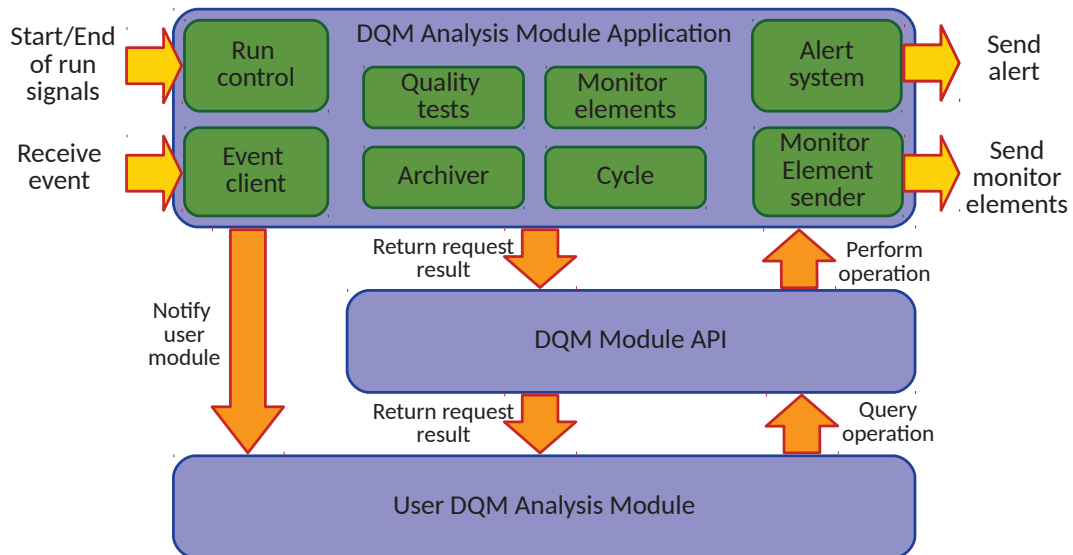


FIGURE 7.8 – Schéma illustrant l'interface utilisateur des modules d'analyse de données en ligne (application analysant les données issues du collecteur d'événements).

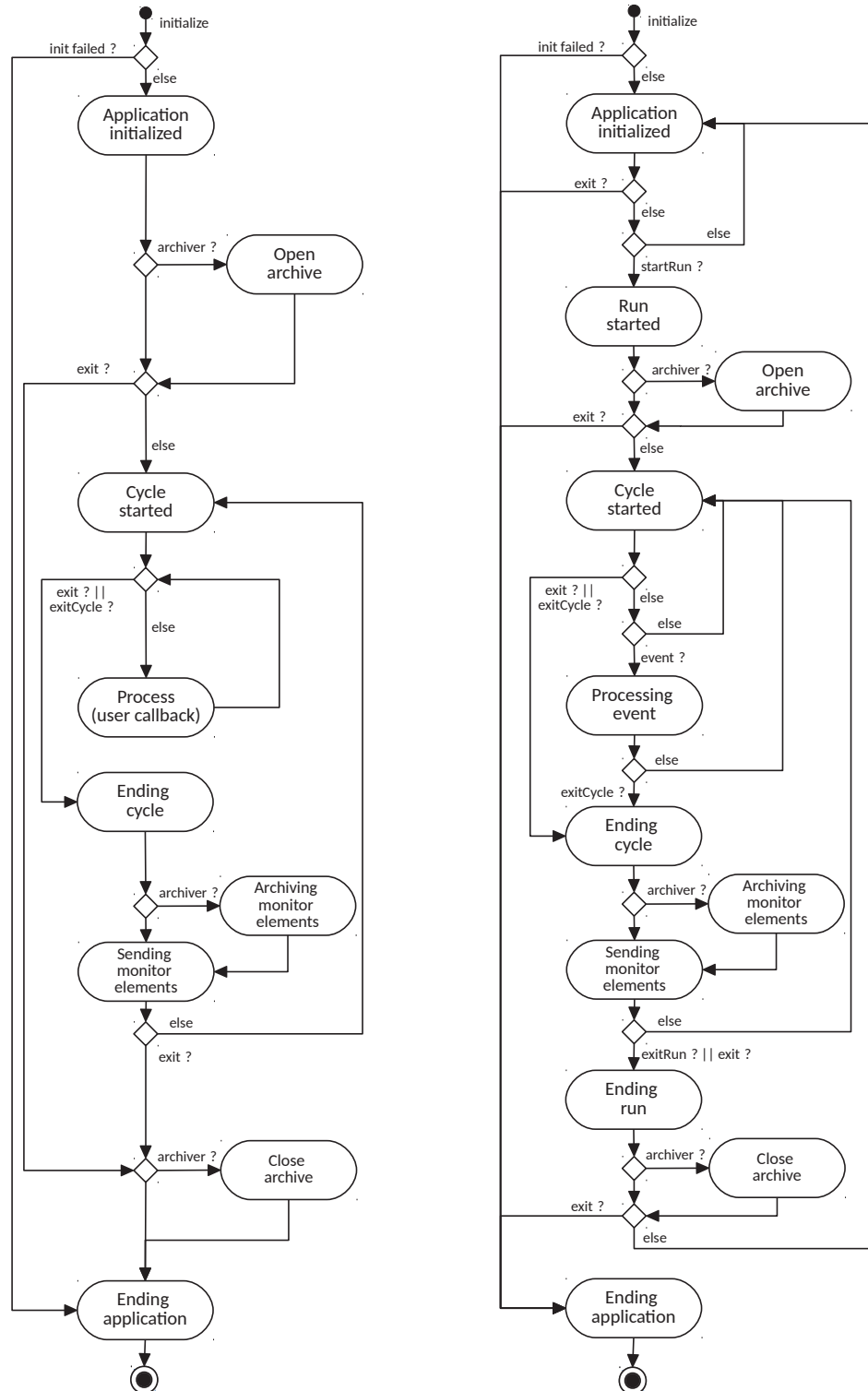


FIGURE 7.9 – Diagramme d'état UML de l'application des modules d'analyse en ligne : à gauche l'application dite *standalone* et à droite l'application analysant les données issues du collecteur d'événement.

des signaux de début et fin de *runs*. Pour l'application dite *standalone*, la fonction `process()` remplace la fonction `processEvent()` de l'application d'analyse d'événements. Cette fonction est une simple fonction de rappel dans laquelle l'utilisateur définit ce que bon lui semble.

Les deux applications sont configurées par un fichier XML contenant les paramètres internes pour l'application et les paramètres utilisateur nécessaires à l'analyse des données. Un exemple de module d'analyse de données avec un fichier de configuration est donné dans l'annexe C.

2.5.1 Gestion des éléments de surveillance

Comme énoncé précédemment, le but des applications d'analyse de données est de convertir un amas important de données en éléments de plus petite taille, par exemple des histogrammes, des graphes, des valeurs scalaires ou encore du texte. La classe `DQMMonitorElement` englobe toutes les propriétés d'un élément de surveillance :

- un nom,
- un type,
- un chemin dans une arborescence,
- le nom du module qui l'a alloué,
- une description,
- le nom du collecteur où il sera disponible,
- une police de remise à zéro,
- une qualité,
- une option spécifiant si l'élément doit être publié à la fin d'un cycle,
- le numéro du *run* en cours,
- un objet de type `TObject`,
- une option utilisée pour le tracé de l'élément,
- une liste de tests de qualité,
- la liste des résultats des tests de qualité.

Au sein des applications d'analyse, l'allocation d'un élément nécessite au minimum le *nom*, le *chemin* dans l'arborescence et un *titre*. Le *type* de l'élément est déterminé par la fonction qui alloue l'élément, par exemple `bookRealHistogram2D()`. Le nom, le chemin, le nom du module et du collecteur identifient de manière unique un élément aux yeux d'un utilisateur externe.

Les éléments sont stockés en interne dans une structure en dossier, permettant ainsi l'existence non-conflictuelle de deux éléments avec le même nom mais rangés dans deux dossiers différents. La *police de remise à zéro* permet quant à elle de remettre à zéro un élément de manière transparente par l'application. Cela peut être utile par exemple pour remplir un histogramme pendant un cycle et le remettre à zéro après l'envoi au collecteur. La *qualité* est une énumération identifiant la qualité de l'élément par l'utilisateur, allant de 1 pour "très mauvaise qualité" à 5 pour

"très bonne qualité" (0 pour un élément sans qualité particulière). L'objet de type `TObject`, classe de base de la plupart des objets `ROOT`, fournit un polymorphisme pour les différents types d'histogrammes, de graphes, etc ... Il permet aussi à l'utilisateur d'insérer son propre type d'objet héritant de l'interface `TObject`.

Tests de qualité

Les éléments de surveillance ont la possibilité d'enregistrer nominativement plusieurs tests de qualité. Ces derniers sont exécutés à la demande de l'utilisateur, par exemple en fin de cycle, et fournissent, un à un, une évaluation de la qualité de l'élément. Par exemple, pour une distribution en énergie attendue gaussienne dans un calorimètre, il serait possible d'ajuster une fonction gaussienne et d'évaluer la qualité de la distribution en fonction du χ^2 de l'ajustement. Un autre exemple, plus utilisé dans les grandes expériences telles que CMS ou ALICE, serait de comparer un élément à un autre élément servant de référence. Le test comparerait deux histogrammes en utilisant un test de Kolmogorov et évaluerait la qualité de l'élément en fonction du résultat du test statistique. Certains tests de qualité, comme l'ajustement de fonction, sont fournis par l'environnement de développement, mais d'autres peuvent y être insérés par l'utilisateur. Une fois un test de qualité exécuté, son résultat est stocké dans l'élément. À noter que l'élément et les résultats des tests de qualité associés sont envoyés ensemble au collecteur.

Archivage des éléments

Au lancement de l'application, l'utilisateur a la possibilité d'activer un archivage automatique des éléments de surveillance. Une archive (un fichier `ROOT`) est ouverte en début de *run* ou au démarrage de l'application, en fonction du type d'application. Les objets de type `TObject`, stockés dans chacun des éléments de surveillance, sont archivés à la fin de chaque cycle.

2.5.2 La structure en cycle

Comme le montre la figure 7.2, les applications d'analyse de données s'inscrivent dans un tout. D'un côté des données sont reçues, et d'un autre, les éléments de surveillance sont transmis au collecteur dédié, qui seront visualisés en bout de chaîne sur les postes clients.

Contrairement aux analyses effectuées hors-ligne, les données ne sont pas traitées de façon linéaire. Par exemple, sur la ligne du SPS au CERN, les particules arrivent sous forme de paquets durant entre 4 et 10 secondes appelés *spill*, eux-mêmes espacés d'un temps mort de 30 secondes. Cela signifie que pendant quelques secondes, un flot important de données sera traité par les analyses, suivi d'une seconde période pendant laquelle aucune donnée ne sera traitée. Une telle structure temporelle de traitement des événements impacte la conception des applications d'analyse. De même, la réception des éléments de surveillance sur les postes clients en bout de chaîne, ne peut se faire à chaque événement traité. Visualiser un histogramme qui est mis à jour toutes les 10 millisecondes ne comporte que très peu

d'intérêt. Sachant qu'une analyse fournit en général un nombre important d'histogrammes, il devient inconcevable d'en envoyer plusieurs centaines dans un aussi court laps de temps.

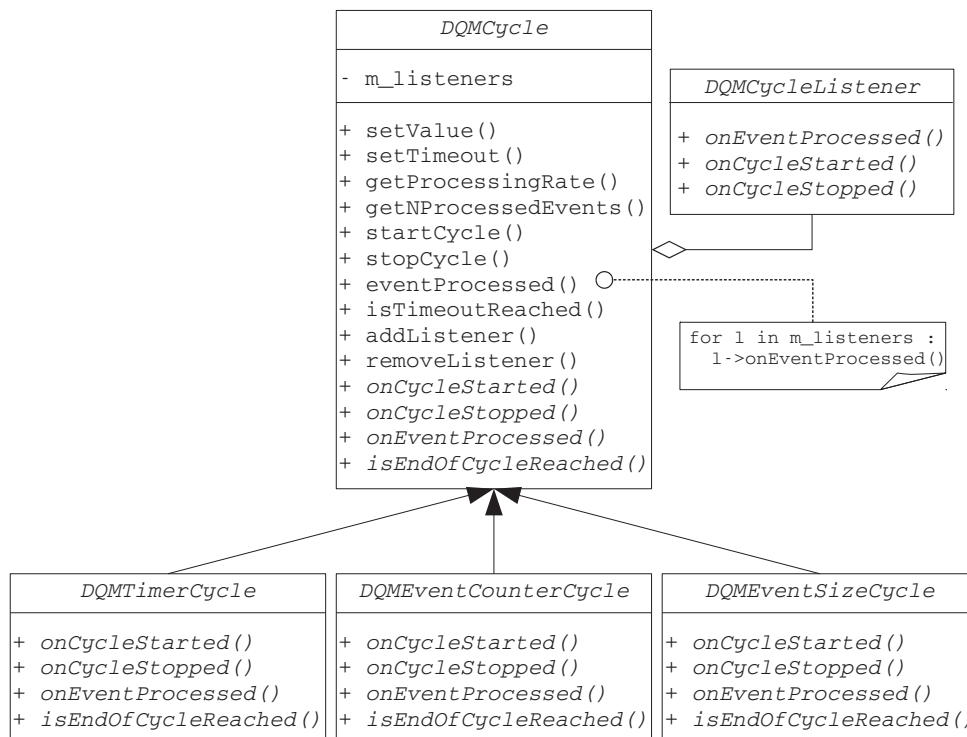


FIGURE 7.10 – Diagramme de classe UML illustrant l'organisation des classes relatives aux cycles. Les trois implémentations (en bas) constituent les trois types de cycles disponibles par défaut dans le logiciel.

Tout ceci suggère de structurer le traitement de données au sein de l'application sous forme périodique, appelé *cycle*. Au cours d'un cycle, l'application traite les données issues du collecteur d'événements. A la fin du cycle, la liste des éléments de surveillance déclarés par l'utilisateur comme transmissible est envoyée au collecteur dédié. Ceci permettra entre autre de réduire le temps passé à envoyer les éléments au collecteur.

La figure 7.10 montre les relations entre les différentes classes autour de la gestion des cycles. Le fonctionnement des cycles est encapsulé dans l'interface `DQMCycle`. Trois types de cycles sont fournis par le logiciel :

- `DQMTimerCycle` implémente un cycle basé sur une période de temps. Au bout de n secondes, le cycle se termine.
- `DQMEventCounterCycle` implémente un cycle basé sur un comptage d'événements. Au bout de n événements analysés, le cycle se termine.
- `DQMEventSizeCycle` implémente un cycle basé sur la taille du tampon mémoire utilisé pour stocker l'événement brut reçu (en octet). Au bout de n octets d'événements reçus, le cycle s'arrête.

Les cycles possèdent aussi un temps mort au bout duquel, si aucun événement n'est analysé, ils s'arrêtent. Les cycles au sein de l'application sont traités comme des *plugins* permettant à l'utilisateur de définir et d'insérer un nouveau type de cycle au sein de l'application. On pourrait par exemple imaginer un nouveau cycle comme étant une combinaison "OU" des trois cycles définis précédemment.

En outre, la structuration en cycles permet d'accumuler de la statistique pendant une période donnée. Cela permet le calcul de moyennes sur des distributions ou de faire des ajustements de fonctions sur cette période.

2.6 Collecteurs d'éléments de surveillance

Le collecteur d'éléments de surveillance assure le lien entre les modules qui produisent les éléments de surveillance, et les applications clientes qui veulent accéder à ces derniers. Cependant, étant donnée la quantité importante d'éléments pouvant provenir des modules, il n'est pas concevable de tous les collecter dans le même processus. De même, du point de vue du client, visualiser des centaines d'histogrammes en même temps n'a pas beaucoup de sens. Il n'est donc pas utile de collecter tous les éléments à la fin de chaque cycle des modules. Par conséquent, uniquement des informations succinctes seront collectées, telles que le nom, le chemin ou encore la description des éléments.

La figure 7.11 illustre les interactions entre un collecteur et deux interfaces clients.

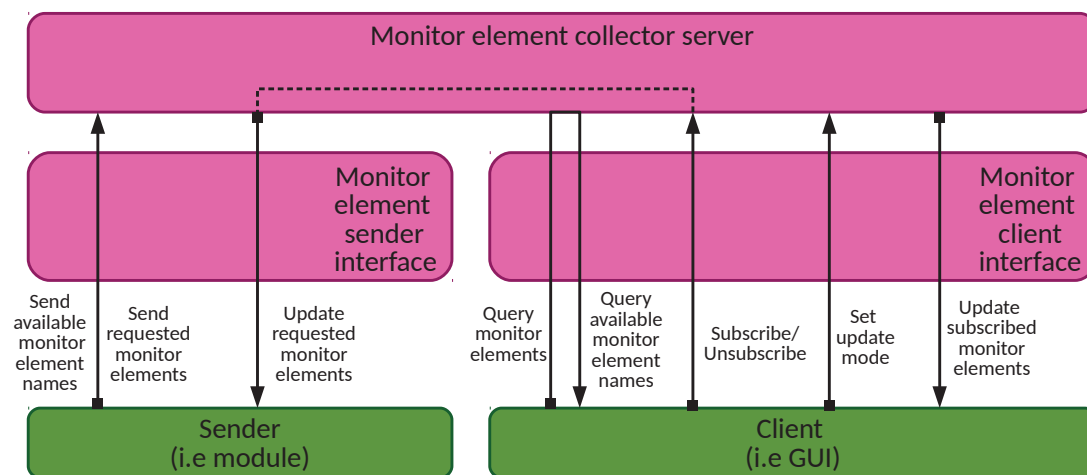


FIGURE 7.11 – Schéma illustrant les interactions entre un collecteur d'éléments de surveillance et ses clients. Deux type de clients : le sender qui transmet les éléments au collecteur et le client qui envoie des requêtes d'accès.

La première est utilisée dans les applications de modules (en bas à gauche) pour envoyer les éléments au collecteur. Une liste d'éléments souscrits par les clients est enregistrée dans l'interface de sorte que seuls les éléments apparaissant dans cette liste soient envoyés. La liste des éléments disponibles est renvoyée à chaque

fois qu'un nouvel élément est disponible. De même, pour informer que les éléments ne sont plus disponibles, par exemple lors de l'arrêt du module, une liste vide est renvoyée au collecteur.

La deuxième (en bas à droite) est une interface client, par exemple une interface graphique d'affichage, qui demande accès aux éléments par l'intermédiaire du collecteur. Grâce à cette interface, l'utilisateur peut souscrire à des éléments de son choix. La liste des éléments disponibles est accessible sur requête. Cette liste peut être filtrée en choisissant uniquement certaines caractéristiques des éléments, par exemple, tous les éléments contenant "calo" et de type histogramme à une dimension. Tout comme pour le collecteur d'événements, deux modes de fonctionnement sont possibles pour l'accès aux éléments. Après souscription, l'utilisateur peut décider de recevoir ces derniers lors de leur mise à jour dans le collecteur. Il peut décider à tout moment de stopper les mises à jour, tout en conservant ses souscriptions. Le deuxième mode fonctionnement n'est autre qu'un accès sur requête.

2.7 Les interfaces graphiques

Les trois interfaces graphiques clientes sont regroupées au sein du paquet `DQMviz`. La librairie graphique utilisée est Qt [40], librairie multi-plateforme, ayant fait ses preuves dans de nombreux projets. Elle est notamment compatible avec certains objets ROOT, tel que `TCanvas`, permettant de tracer des primitives issues du logiciel ROOT. Il est alors possible grâce à l'implémentation du système graphique `TQt` fournie par ROOT, d'exécuter une application Qt et de tracer des primitives ROOT. Cette implémentation sera utilisée pour l'interface de surveillance des données présentée dans la section 2.7.3.

2.7.1 Gestion des *runs*

L'interface graphique client pour la gestion des *runs* est une implémentation graphique de l'interface client fournie par le paquet `DQMCore`. Elle permet de piloter graphiquement le gestionnaire de *run* à distance et/ou d'accéder à son état en cours. La figure 7.12 montre l'interface graphique lors d'un *run* en cours.

2.7.2 Gestion des processus à distance

Implémentation graphique de l'interface client fournie par le paquet `dimjc`, cette interface graphique est plus que nécessaire lors d'un déploiement en situation réelle. Elle permet de contrôler une grande quantité de processus exécutés sur plusieurs hôtes en parallèle à partir d'une seule et unique fenêtre. Elle peut aussi être utilisée de façon plus générique, dans tout type de contexte puisque son implémentation à elle seule ne dépend pas du logiciel de surveillance mais uniquement de `dim` et `dimjc`.

L'interface permet de :

- visualiser en temps réel les méta-données relatives à une liste de processus,
- démarrer, arrêter, redémarrer un ou plusieurs processus,

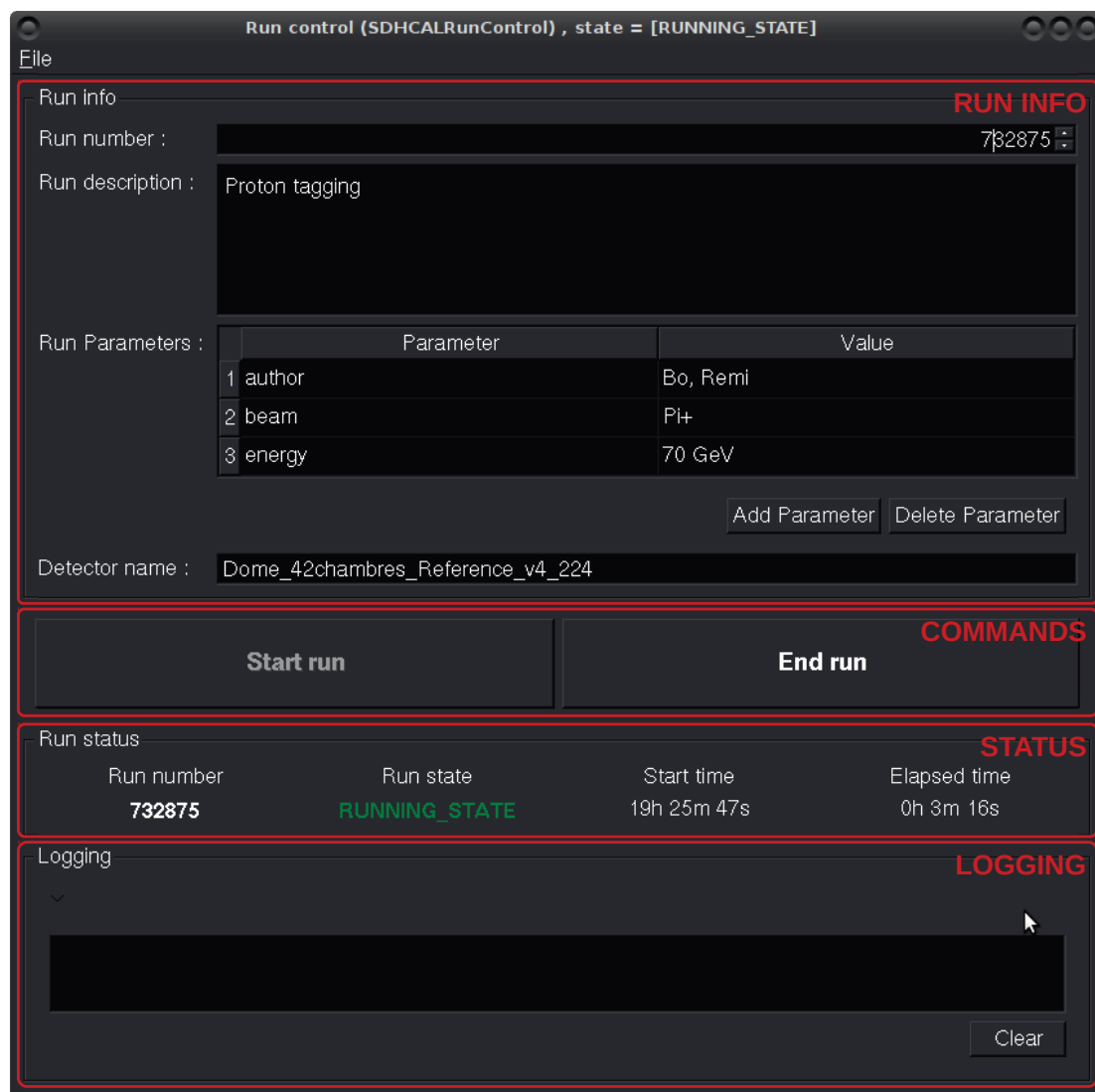


FIGURE 7.12 – Interface graphique cliente de gestion de run

Job interface

Stop Update period (secs) : **AUTOMATIC UPDATES**

Set Kill Method INT (Interrupt): 2 **KILL METHOD**

Job Control	Program Name	PID	Status
<input type="checkbox"/> lysdhal10	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_run_control_server	11045	S (sleeping)
<input type="checkbox"/> DQMRunControl	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_event_collector	12154	S (sleeping)
<input type="checkbox"/> EventCollector	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_event_collector	12157	S (sleeping)
<input type="checkbox"/> EventCollector_2	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_monitor_element_collector	12164	S (sleeping)
<input type="checkbox"/> MonitorElementCollector			
<input type="checkbox"/> lysdhal7			
<input type="checkbox"/> AsicAnalysisModule	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_analysis_module	31189	R (running)
<input type="checkbox"/> ECalAnalysisModule	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_analysis_module	31251	R (running)
<input type="checkbox"/> EventDisplayModule	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_analysis_module	31268	D (disk sleep)
<input type="checkbox"/> HoughTransformModule	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_analysis_module		X (dead)
<input type="checkbox"/> ParticleIDModule	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_analysis_module		X (dead)
<input type="checkbox"/> BeamAnalysisModule	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_analysis_module	31280	R (running)
<input type="checkbox"/> HitAnalysisModule	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_analysis_module	31292	D (disk sleep)
<input type="checkbox"/> RawAnalysisModule	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_analysis_module	31301	R (running)
<input type="checkbox"/> SlowControlModule	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_analysis_module	31308	R (running)
<input type="checkbox"/> lysdhal9			
<input checked="" type="checkbox"/> ShmDriver	/opt/dqmsoftware/dqm4hep/bin/dqm4hep_start_shm_driver	21002	S (sleeping)

Load file Reload file Open LogFile Update

ARGS and ENV

FIGURE 7.13 – Interface graphique cliente de gestion de processus. La figure montre le découpage de l'interface graphique en plusieurs zones distinctes. La table des processus montre trois serveurs (lysdhal10, lysdhal7 et lysdhal9) avec chacun une liste de processus que l'utilisateur contrôle à distance.

- lire les journaux de chaque processus en cours.

La figure 7.13 montre l'interface graphique de gestion de processus à distance pour trois serveurs et leurs processus associés.

2.7.3 Interface de surveillance des données

Cette interface graphique se situe en bout de chaîne de traitement de données du système de surveillance et constitue le rendu final du logiciel pour l'utilisateur. Sa principale fonction est de récupérer les éléments de surveillance de différents collecteurs et de les afficher. Pour cela un client graphique vers les collecteurs a été développé. Il permet à l'utilisateur de naviguer au sein des collecteurs disponibles et de construire des sélections d'éléments qu'il souhaite afficher par la suite.

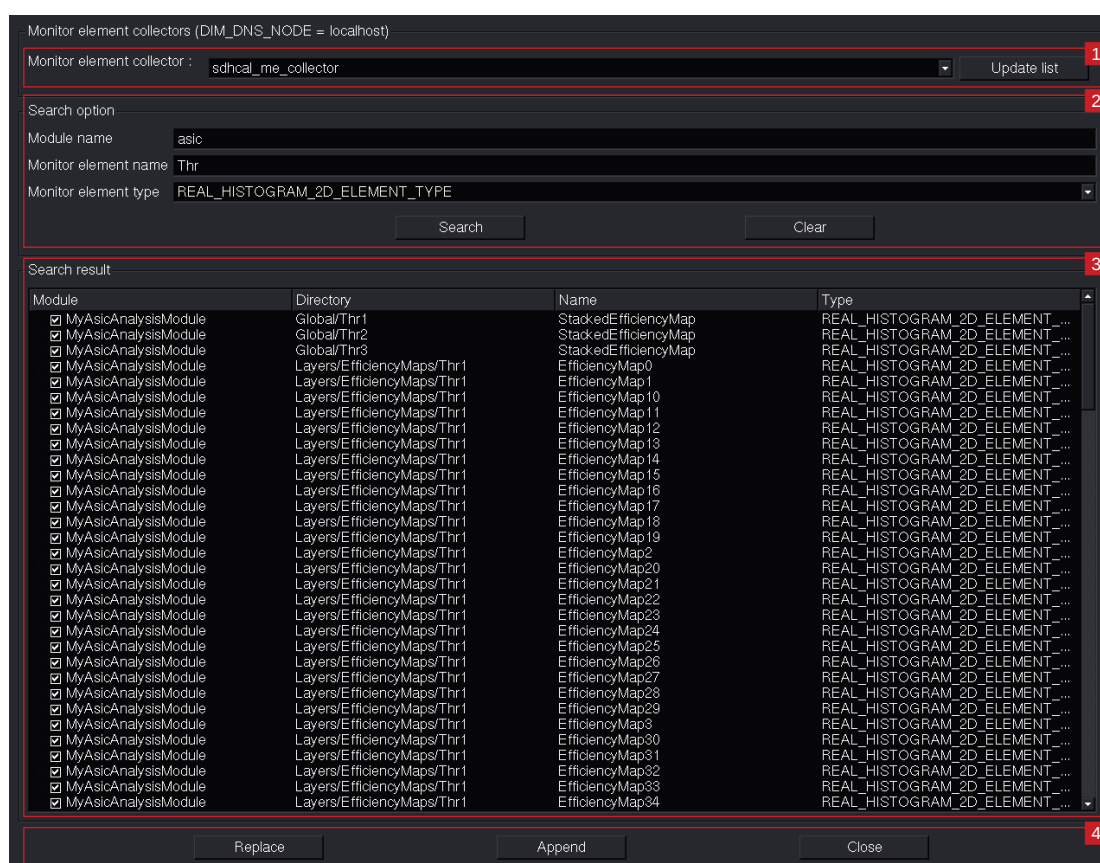


FIGURE 7.14 – Interface graphique cliente vers les collecteur d'éléments de surveillance.

La figure 7.14 montre l'interface graphique cliente vers les collecteurs d'éléments de surveillance, aussi appelé navigateur, avec un découpage des différentes zones graphiques :

1. sélection du collecteur d'élément de surveillance,

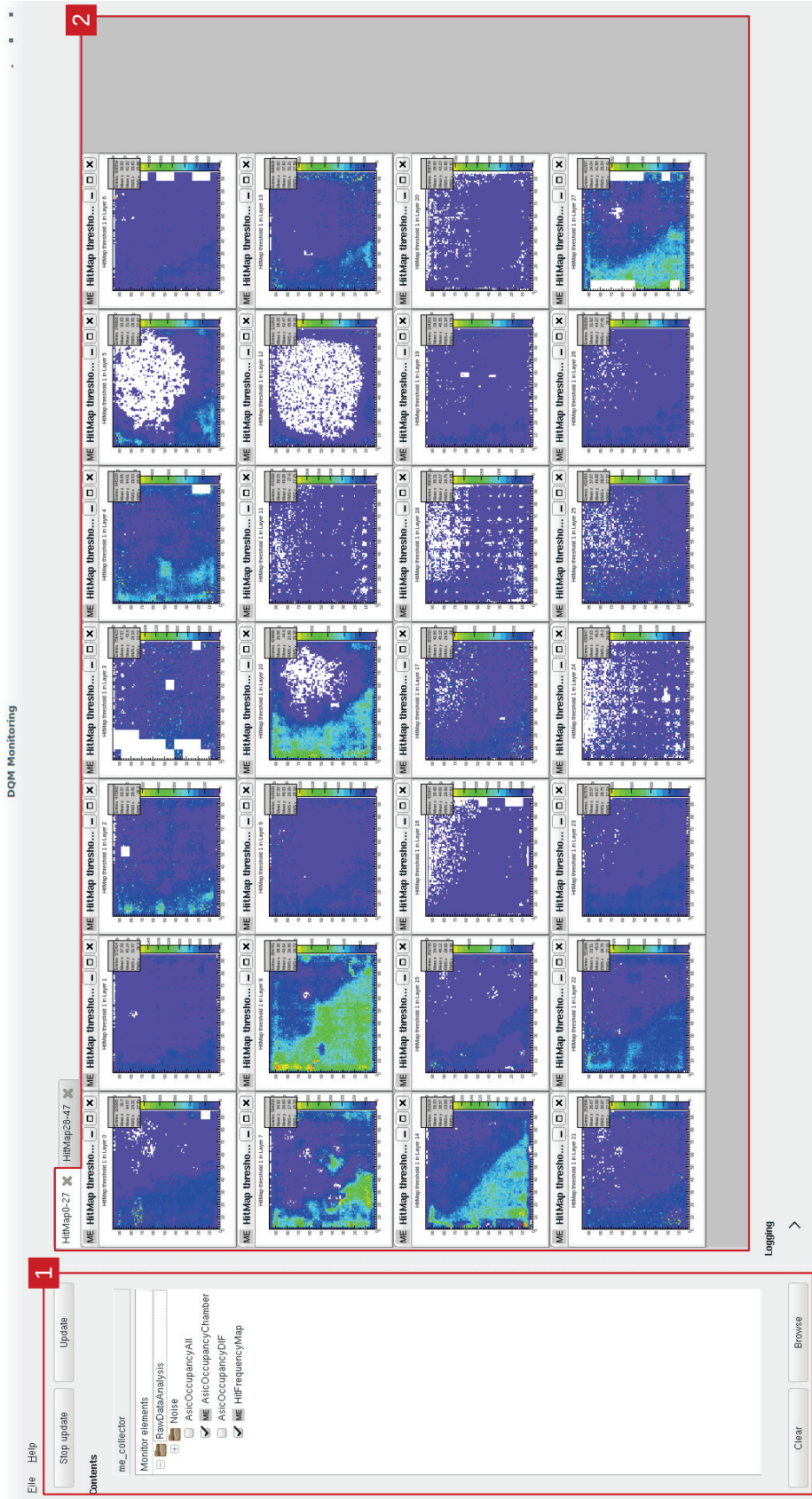


FIGURE 7.15 – Interface graphique cliente de récupération et d’affichage d’éléments de surveillance. Les éléments de surveillance affichés dans la zone centrale (2) proviennent du module d’analyse de données brutes du SDHCAL (voir section 3.3). Chaque élément tracé est une carte de comptage du nombre de hits pour une chambre donnée du SDHCAL. La sixième carte de comptage de la deuxième ligne (chambre numéro 12) montre un déficit de hits dans la zone centrale. Ceci est dû à un flux trop important de gaz dans la chambre, problème qui a pu être corrigé par la suite.

2. options de filtrage pour construire une sélection,
3. résultats de la recherche et construction d'une sélection (cocher/décocher),
4. ajout de la sélection dans l'interface graphique principale.

Au niveau de l'interface graphique principale, un soin particulier a été apporté à l'architecture. L'application du patron de conception MVC (Modèle-Vue-Contrôleur) a permis de séparer :

- La structuration des données au sein de l'application (modèle)
- La vue, c'est à dire l'interface graphique elle-même (vue)
- Le contrôle de l'application ou *backend*, qui implémente le fonctionnement de l'application même (contrôleur)

La figure 7.16 illustre les interactions principales entre chacun de ces trois composants.

Le modèle stocke les éléments en provenance des différents collecteurs avec le même type de stockage utilisé au sein des applications de modules (voir section 2.5). Quand un élément est ajouté, supprimé ou mis à jour, une notification est envoyée à la vue pour incorporer les changements. La liste de tous les éléments de surveillance disponibles au sein du modèle, constitue la sélection courante de l'application.

Le contrôleur sert de lien central entre la vue, le modèle et les différents collecteurs d'éléments de surveillance. Il utilise plusieurs interfaces client vers les collecteurs d'éléments de surveillance, une par collecteur. Toute requête ou opération en relation avec les collecteurs s'effectuera donc par le biais du contrôleur. De nouvelles connexions peuvent être ajoutées grâce au navigateur, décrit précédemment dans cette section.

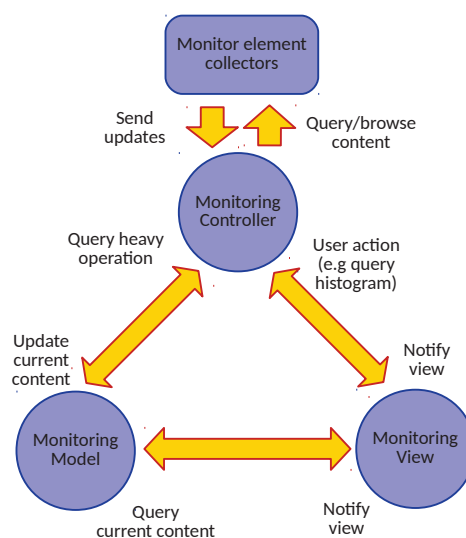


FIGURE 7.16 – Schéma illustrant l'architecture interne de l'interface graphique de surveillance des données.

La vue gère les interactions entre l'utilisateur et l'application. La figure 7.15 montre la fenêtre principale de l'application avec quelques éléments tracés en exemple. La zone 1 montre la sélection d'éléments de surveillance construite par l'utilisateur alors que la zone 2 fournit des aires de tracés organisées sous forme d'onglets, permettant de visualiser plusieurs primitives à la fois.

3 Implémentation pour le prototype SDHCAL

Comme mentionné dans la section 2.1, pour faire fonctionner le logiciel de surveillance de données, l'utilisateur doit fournir une certaine quantité de code. Ainsi, pour analyser les données qui proviennent des prototypes CALICE SDHCAL et CALICE SiWECAL lors des tests en faisceau, un jeu de bibliothèques a été développé. Elle fournissent les fonctionnalités pour i) convertir et trier les données brutes provenant des détecteurs et ii) analyser les données extraites. Les convertisseurs et les modules pour le détecteur SDHCAL qui vont être présentés, ont été développés dans un premier temps à partir de fichiers de données SDHCAL de tests précédents, en simulant le comportement d'une prise de données réelle. Pour le détecteur SiWECAL, la validation des convertisseurs de données et des modules d'analyse n'a pu s'effectuer qu'au moment de la prise de données lors du test en faisceau.

Après avoir détaillé l'implémentation des convertisseurs de données et des modules d'analyses, le déploiement global du logiciel sera présenté. Quelques estimations des performances mémoires et des performances réseau des processus mis en jeu dans ce déploiement seront finalement exposées.

3.1 Combinaison des détecteurs SiWECAL et SDHCAL

Lors du test en faisceau du mois de juin 2016, deux calorimètres ont été installés sur la ligne du SPS au CERN pour les soumettre à différents types de faisceaux de particules. Le but de ce test était cependant plus focalisé sur l'aspect ingénierie. En effet pour la première fois, deux calorimètres à grande granularité ont pu être testés avec un système d'acquisition combiné. Ce test a permis de valider d'une part la combinaison des deux systèmes d'acquisition mais aussi le système de surveillance des données pour un jeu de données combinées.

La figure 7.17 montre un schéma de positionnement des deux calorimètres lors du test en faisceau. Le faisceau de particules incidentes entre d'abord dans le calorimètre électromagnétique pour ensuite passer dans le calorimètre hadronique.

Le calorimètre SiWECAL est un calorimètre électromagnétique à échantillonnage développé en grande partie à Paris au Laboratoire Leprince-Ringuet (LLR) et au Laboratoire de l'Accélérateur Linéaire (LAL). Chaque couche alterne une couche d'absorbeur de tungstène d'épaisseur variable suivie d'une couche active en silicium de 2.5 mm. Cette dernière couche est segmentée en 32 x 32 cellules de 0.5 x 0.5 cm^2 , pour un total de 1024 cellules par plan. Dans la version utilisée lors du test sur faisceau, 10 couches étaient entreposées, avec des emplacements pour insérer les couches d'absorbeur. Cependant, le second objectif du test sur faisceau était de tester ces 10 couches successives du calorimètre électromagnétique. Afin de pouvoir récupérer des données dans le détecteur SDHCAL sans interaction préalable des particules, les couches de tungstène ont été retirées.

Le prototype technologique du SDHCAL est quant à lui décrit au chapitre 3.

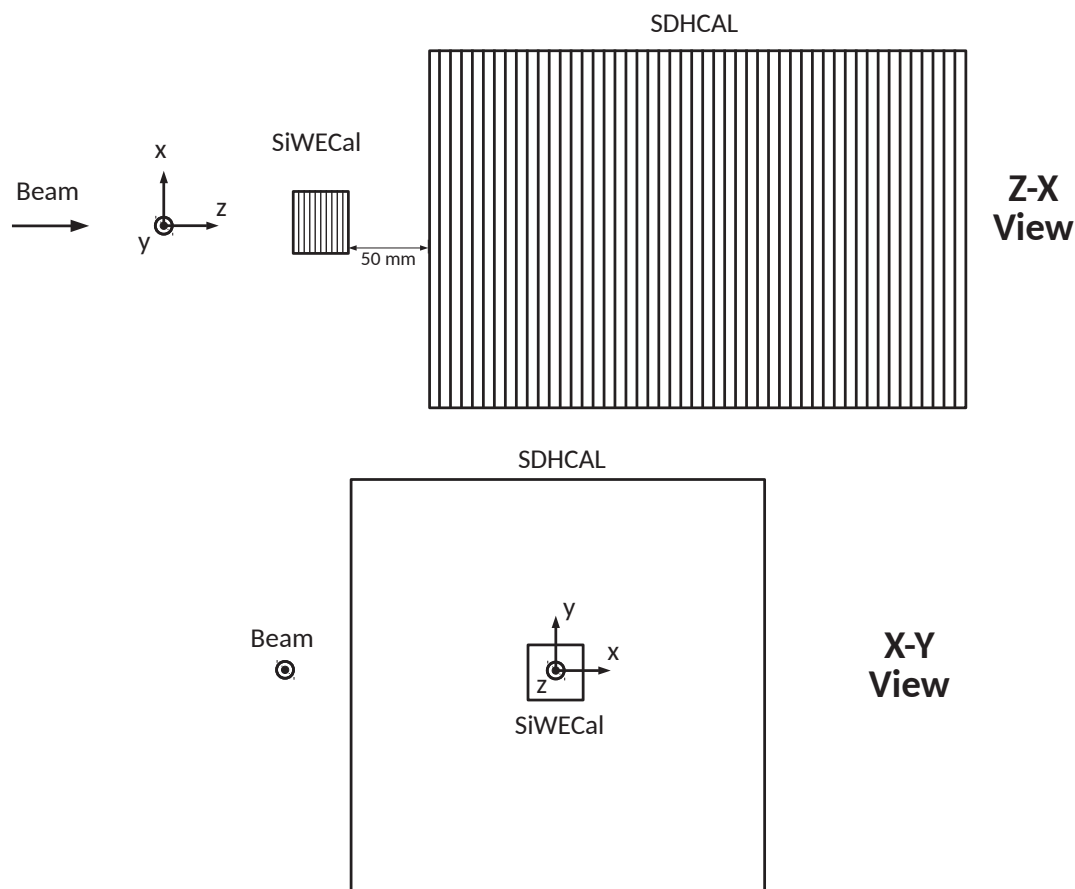


FIGURE 7.17 – Schéma de positionnement des calorimètres SiWECAL et SDHCAL lors du test en faisceau du mois de juin 2016 : en haut la vue Z-X, en bas la vue X-Y. Les dimensions ne sont pas à l'échelle.

3.2 Source de données

La reconstruction des données brutes se fait à l'aide du logiciel `levbdim`. Chaque DIF (voir chapitre 3) du SDHCAL, au nombre de 144 dans sa version la plus courante, est considéré comme une source de données. L'application DQM4HEP de reconstruction des événements attendra donc 144 sources de données. Chacune d'elle délivrera l'ensemble des données relatives à chaque voie au cours d'un cycle d'acquisition, sous forme d'une mémoire tampon. Une bibliothèque C++ fournie par la collaboration SDHCAL permettra par la suite de décoder le tampon mémoire de chaque DIF pour extraire les données.

En ce qui concerne le SiWECAL, le système d'acquisition combiné des deux détecteurs offre la possibilité de republier les données d'un détecteur dans l'autre détecteur. Entre d'autres termes, les données acquises par le SiWECAL seront renvoyées dans le système d'acquisition du SDHCAL. Ceci implique que les données publiées par le SDHCAL avec le logiciel `levbdim` comporteront aussi les données

du SiWECAL sous forme de source de données additionnelles. Au total seront publiés avec `levbdim`, 144 sources de données provenant de SDHCAL et 10 sources de données additionnelles provenant du SiWECAL.

Le format des événements utilisé pour traiter les données sera le format LCIO dont le *streamer* d'événements est fourni par le paquet `DQM4ILC`. Les modules de conversion de données convertiront donc des données brutes sous forme de mémoire tampon vers un format de type LCIO. Pour fournir aux analyses les données les plus complètes possibles, trois modules de conversion ont été développés.

EventInfoProcessor Exécuté en premier lors de la réception des données, il encode les données générales d'un événement : numéro d'événement, numéro du *run* en cours, nom du détecteur et horodatage.

SDHCALProcessor Ce module de conversion traite seulement les données du SDHCAL. A chaque événement, une collection de type `EVENT::CalorimeterHit` est créée, contenant les informations de chaque voie touchée au cours du cycle d'acquisition. Pour chacune des voies, on enregistre le numéro de la DIF, le numéro de l'ASIC, le numéro de la voie, l'identifiant de croisement de faisceau (BXID), le seuil (du mode semi-digital), les identifiants de cellule (I et J dans le plan et K pour le plan) et la position en *mm* par rapport à l'origine.

SiWECALProcessor Exécuté en dernier, il convertit les données brutes du ECAL dans le même format que le SDHCAL (collection de `EVENT::CalorimeterHit`). La seule différence notable, hormis la structure des données lors du décodage, est le champ "énergie" qui remplace le seuil dans le cas du SDHCAL. Plusieurs options permettent de choisir ce qui sera stocké dans le champ "énergie" de la structure LCIO :

1. pas de stockage d'énergie (mode digital).
2. la valeur de l'énergie est supposée calibrée dans les données brutes et est directement copiée dans la structure LCIO.
3. la valeur en ADC remplace la valeur de l'énergie. Elle pourra être traitée par les modules d'analyse après coup.
4. l'énergie est calibrée au moment de la conversion des données à partir de la valeur en ADC et des constantes de calibration issues de données externes (fichiers ROOT détaillés ou constantes de calibration moyennes).

Durant le test en faisceau combiné, les constantes de calibration n'étant pas disponibles, la valeur en ADC remplacera la valeur de l'énergie (choix 3).

3.3 Les modules d'analyse

Des modules d'analyse de données ont été développés pour les détecteurs SiWECAL et SDHCAL. Cependant, le traitement des données combinées n'étant pas encore prêt à cette période, des analyses séparées ont donc été implémentées.

Module de *slow control*

- Type de module : *Standalone*
- Type de données : *slow control*
- Description :

Ce module ne traite pas de données issues des collecteurs mais des données de type environnemental. Un capteur externe permet de récupérer la température (figure 7.18(c)) et la pression autour des détecteurs (figure 7.18(d)). Ces variables sont ensuite tracées dans des graphes dédiés. Au niveau du détecteur SDHCAL, les valeurs de la basse tension globale appliquée et lue sont aussi récupérées. Le même type de relevé est effectué chambre par chambre pour la haute-tension (figure 7.18(a)).

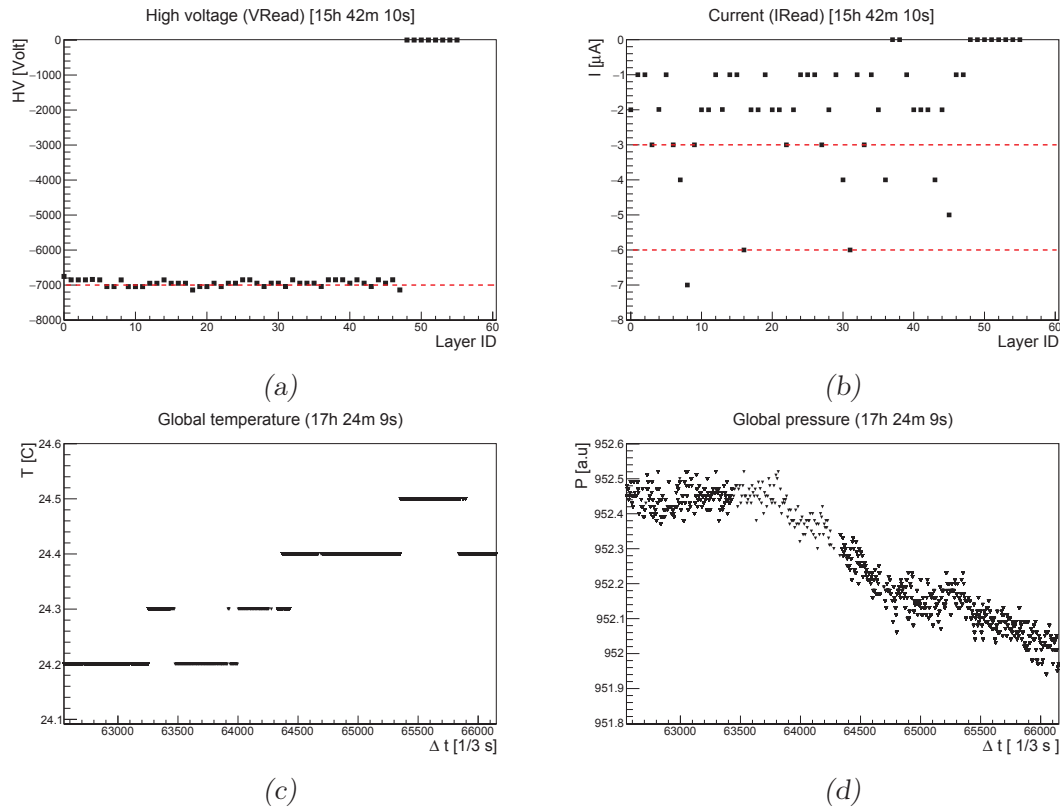


FIGURE 7.18 – Éléments de surveillance du module de *Slow Control* les plus vérifiés. (a) Lecture de la haute-tension par chambre GRPC. (b) Lecture du courant par chambre GRPC. (c) Évolution de la température autour des calorimètres en fonction du temps. (d) Évolution de la pression autour des calorimètres en fonction du temps.

La variable la plus intéressante et la plus vérifiée au cours du test en faisceau a sans doute été le courant dans chaque chambre (figure 7.18(b)). En effet, en fonction de la température et de la pression du gaz dans les chambres, les courants peuvent

fluctuer et impacter la détection des particules. Pour ne pas négliger la surveillance de cette donnée, des tests de qualité sur la valeur des courants ont été encodés. Quatre plages de valeurs de courant fixent la qualité des courants dans les chambres :

$0 \mu A \leq I \leq -3\mu A$	→ Très bonne qualité
$-3 \mu A < I \leq -6\mu A$	→ Bonne qualité
$-6 \mu A < I \leq -10\mu A$	→ Mauvaise qualité
$-10 \mu A < I$	→ Très mauvaise qualité

Le code couleur affiché sur l'interface graphique met rapidement en évidence un courant trop élevé en jaune (troisième plage) ou en rouge (quatrième plage). Dans ce cas, il est nécessaire de réajuster le flux de gaz global ou dans la chambre en question. Des graphes additionnels permettent aussi de suivre les variations de courant en fonction du temps.

Module d'*event display*

- Type de module : *Analyse*
- Type de données : Données SDHCAL
- Description :

Le principal but de ce module est de classer les événements par type de particule et d'afficher les événements en 3D ou dans différentes vues projetées. La classification s'opère grâce à des variables topologiques et permettent de sélectionner les muons cosmiques, les muons du faisceau, les particules électromagnétiques (e^\pm et γ), les hadrons chargés, les hadrons neutres et les événements de bruit de fond ou cohérents.

Pour chaque type de particule sont tracés :

- une vue 3D de l'événement,
- les profils Z-X, Z-Y et X-Y de l'événement (i.e figure 7.19(b)),
- les mêmes profils mais superposés sur une période d'un cycle.

Ce module a été particulièrement utile, lorsque après un redémarrage du faisceau sur la ligne du SPS, notre faisceau de pions était réapparu comme un faisceau d'électrons. Les détecteurs continuaient de prendre des données normalement, le flux de particules estimé était toujours le même, mais le dernier événement hadronique affiché sur l'interface graphique était resté le même et la vue 3D pour les électrons défilait. Cela nous a permis de réagir rapidement, d'arrêter notre prise de données et d'appeler la salle de contrôle pour obtenir plus d'informations.

Certains bruits électroniques déjà connus ont aussi pu être visualisés. Le profil X-Y de bruit, comme montré sur la figure 7.19(a), est un bruit récurrent. Il peut arriver qu'à la fin d'un cycle d'acquisition, la DIF qui envoie le signal de lecture à la MDCC déclenche un signal non voulu sur certaines voies. Ce type de bruit se repère aussi dans les distributions du nombre de hits ou d'énergie durant l'analyse hors ligne.

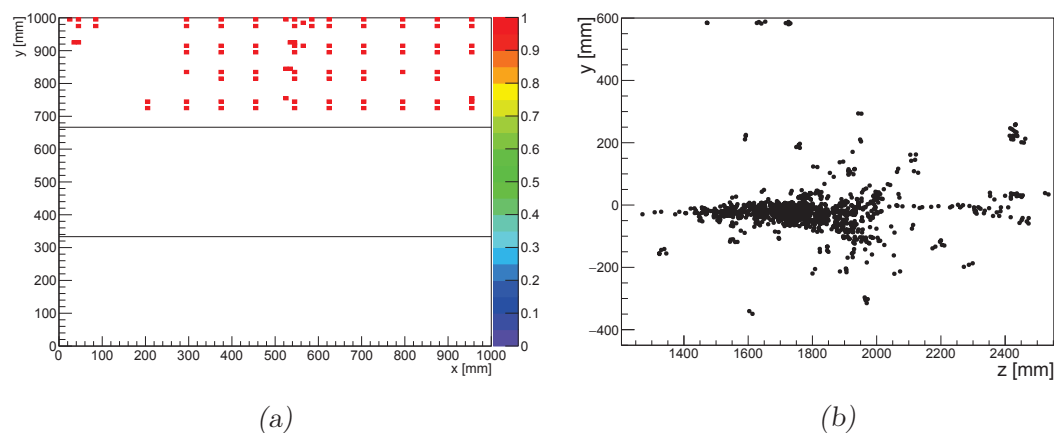


FIGURE 7.19 – (a) Profil X-Y du SDHCAL pour un événement de bruit. On peut y voir une des trois zones avec une série de voie allumées dû à un bruit électronique. (b) Gerbe hadronique issue d'un π^+ de 70 GeV interagissant dans le SDHCAL lors du test sur faisceau.

Module d'analyse des données brutes

- Type de module : *Analyse*
- Type de données : Données SDHCAL
- Description :

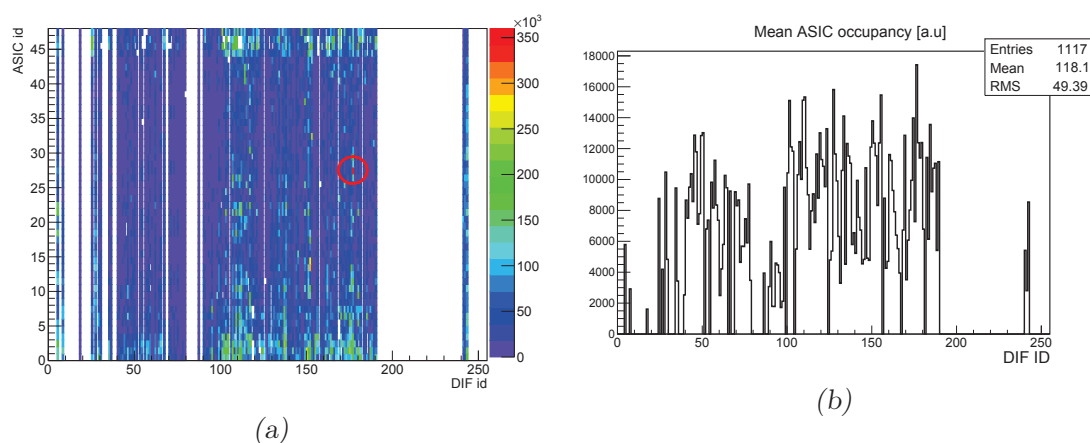


FIGURE 7.20 – Éléments de surveillance du module d'analyse de données brutes les plus vérifiés. (a) Carte de comptage des hits en fonction du numéro de la DIF et de l'ASIC. (b) Taux d'occupation des ASICs en fonction du numéro de DIF.

Ce module fournit un ensemble de cartes de comptage 2D de hits par plan dans le SDHCAL. Ces cartes sont remplies soit par cellule, soit par ASIC. On trouve aussi des graphes de corrélation, comme la carte de comptage des ASICs touchés

en fonction du numéro de la DIF, comme tracé sur la figure 7.20(a). Ce graphe permet de repérer rapidement des points chauds, comme celui entouré en rouge, où certaines puces déclenchent la fin d'un cycle d'acquisition bien trop souvent et/ou rapidement.

Parmi les graphes les plus visualisés, on y verra principalement toutes les cartes de comptage pour chaque plan (voir figure 7.15 de la section 2.7.3), le graphe de corrélation des puces électroniques (comptage DIF vs ASIC) ou encore le taux d'occupation des ASICs en fonction du numéro de la DIF (figure 7.20(b)).

Module d'analyse du faisceau

- Type de module : *Analyse*
- Type de données : Données SDHCAL
- Description :

Ce module, assez court au niveau de l'analyse, utilise les données SDHCAL pour remonter à différentes informations relatives à la structure du faisceau ou au système d'acquisition. Principalement, on retrouvera des informations en temps telles que la distribution de la durée des *spills* (figure 7.21(a)), la distribution de la durée d'un cycle d'acquisition (figure 7.21(b)) ou encore le temps mort entre deux *spills*.

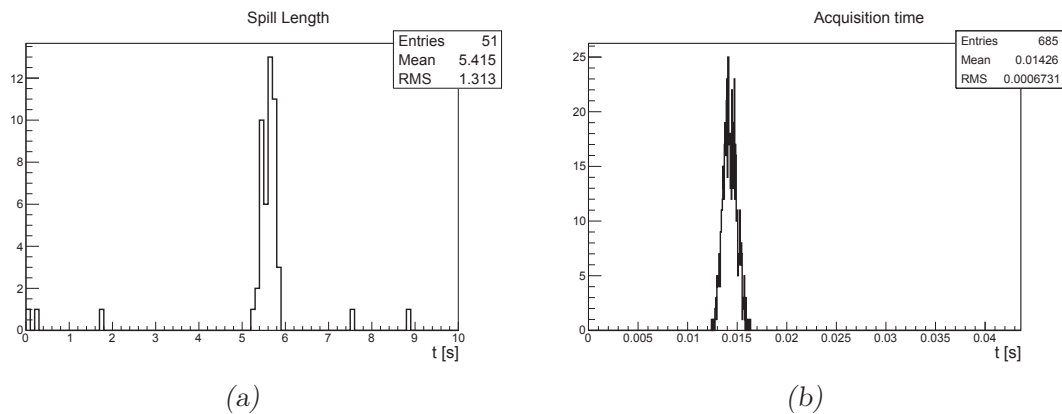


FIGURE 7.21 – Éléments de surveillance du module d'analyse du faisceau les plus vérifiés. (a) Distribution de la durée d'activité du faisceau (*spill*). (b) Distribution de la durée d'un cycle d'acquisition.

Ces informations permettent de mieux comprendre le comportement du système d'acquisition en fonction du réglage du faisceau en salle de contrôle.

Module d'analyse des données du ECAL

- Type de module : *Analyse*
- Type de données : Données SiWECal
- Description :

Ce module est entièrement dédié à l'analyse des données du calorimètre électromagnétique SiWECal. Des cartes de hits ou de comptage ADC pour chaque plan permettent d'identifier les zones de bruits, les zones mortes et le point d'entrée du faisceau. Une distribution en ADC globale a permis de confirmer que le ECal n'était pas calibré.

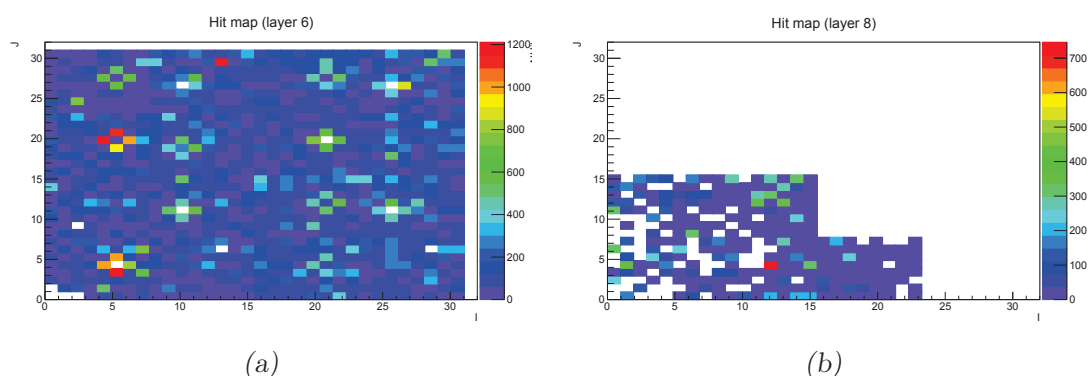


FIGURE 7.22 – Carte de comptage de hits dans le SiWECal : (a) plan 6 et (b) plan 8.

Module d'identification de particules

- Type de module : *Analyse*
- Type de données : Données SDHCAL
- Description :

Ce module classe les événements de la même manière que le module d'*event display* et propose un comptage de particules depuis le début d'un *run*. Des distributions annexes du nombre de *hits* dans le SDHCAL et par type de particules sont aussi fournies afin de confirmer l'identification des particules.

Sa principale utilité est de vérifier la statistique enregistrée lors d'un *run* pour un type de particule donné, mais aussi de vérifier la contamination par d'autres particules dans notre jeu de données.

3.4 Déploiement du système et performances

La figure 7.23 montre le déploiement des processus liés au système de surveillance pour les détecteurs SiWECal et SDHCAL. Quatre serveurs ont été utilisés pour ce déploiement. Le serveur du système d'acquisition, lyosdhcal9, a hébergé la reconstruction des données, tant pour l'écriture sur disque que pour le transfert des

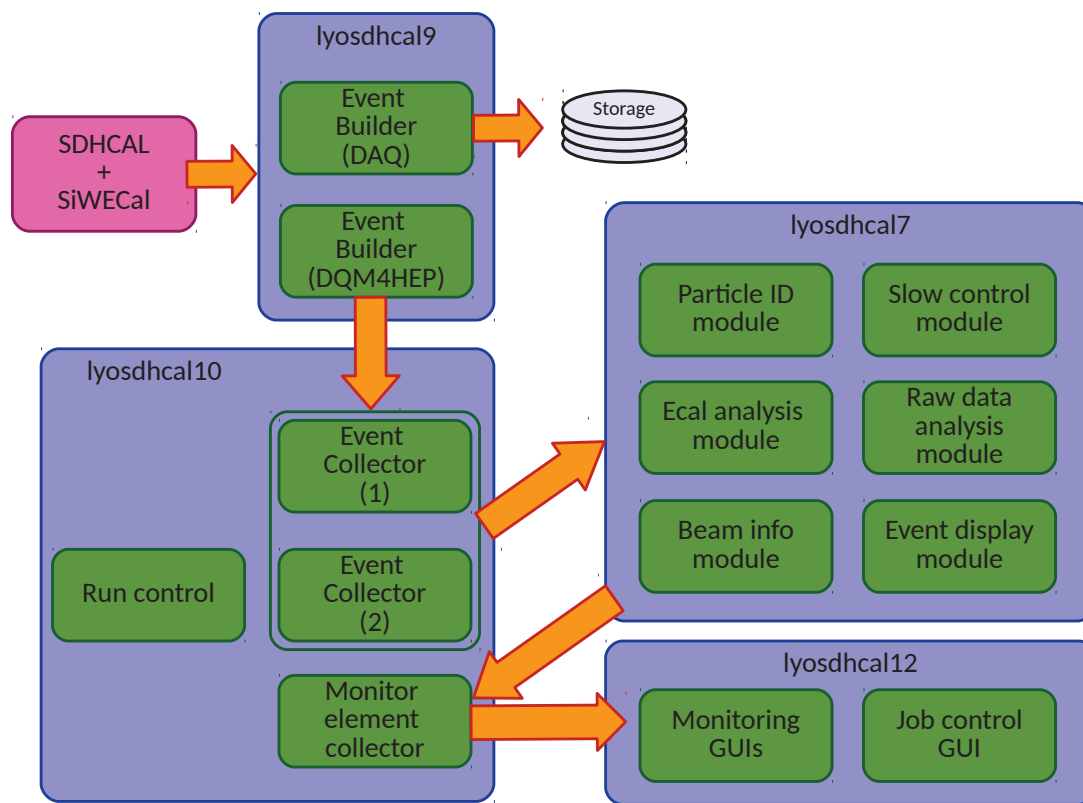


FIGURE 7.23 – Schéma illustrant le déploiement des processus sur les différents serveurs lors du test en faisceau combiné SiWECAL et SDHCAL en juin 2016.

événements au système de surveillance. Aucun autre processus n'a été exécuté sur ce serveur afin de laisser toute la puissance des CPUs pour le système d'acquisition.

Pour ce qui est des modules d'analyse, ils ont tous été installés sur un PC séparé, lyosdhcal7. Malgré la supposée lourde charge CPU des analyses, le seul processeur AMD (4 cœurs) du serveur a permis de tourner les analyses sans soucis tout le long du test sur faisceau, et ce, sans impacter leur vitesse de traitement.

Les applications de collecte et le gestionnaire de *run* ont été regroupés sur le serveur lyosdhcal10. Deux collecteurs d'événements ont été déployés pour contrebalancer la charge CPU des analyses et éviter de perdre des événements. Étant donné le faible nombre de clients d'éléments de surveillance, un seul collecteur d'éléments de surveillance a été déployé.

Les interfaces graphiques de gestion de processus et de surveillance ont été déployées sur le serveur lyosdhcal12, seul PC sur lequel sera branché un écran, hormis le PC d'acquisition. Leur nombre a varié en fonction des besoins des *shifters*.

Bilan du test sur faisceau

Au niveau de la détection de problèmes, le système de surveillance a parfaite-

ment joué son rôle. Les graphes les plus visualisés ont certainement été les graphes d'évolution des courants dans les GRPCs. Principalement vérifiés avant de démarrer une acquisition, ils permettaient de s'assurer que l'efficacité du détecteur était optimale. Le code couleur de l'interface graphique a permis de repérer rapidement les chambres à problème et de corriger les problèmes avant de démarrer les acquisitions.

Les cartes de comptage de hits par voie fournies par le module d'analyse de données brutes ont permis de rendre compte de certains effets dans les chambres. L'uniformité de la circulation de gaz est fortement corrélée aux cartes de comptage de hits. Une circulation de gaz trop importante gonflera les chambres en leur centre et verra l'efficacité de détection et donc le comptage de hits diminuer. Les cartes fournies par ce module ont donc permis de suivre indirectement la répartition du gaz dans chaque chambre au cours d'une acquisition. Les premières données reçues au démarrage de l'acquisition ont donc été capitales à ce niveau.

La distribution du temps d'acquisition a permis de repérer certaines puces trop bruyantes. En effet, quand une puce devient trop bruyante, le signal de fin d'acquisition se déclenche très vite. La distribution sera donc très piquée sur une valeur inférieure à la moyenne usuelle ($\simeq 15-18$ ms). Une période de 4 ou 5 *spills* suffisait en général à effectuer cette vérification.

Un autre exemple de l'utilisation du système de surveillance a été lors d'une acquisition avec un faisceau de pions, lorsque celui-ci s'est subitement arrêté suite à un incident technique sur la ligne. Les derniers événements affichés par le module d'*event display* s'était donc figé sur notre interface. Au redémarrage du faisceau, le flux de particule fourni par les logiciels du CERN était identique. Tout laissait supposer que les paramètres du faisceau étaient identiques. A la reprise de l'acquisition, l'identification des particules dans ce module ainsi que leur affichage sur différents graphes séparés a permis de montrer que notre faisceau était devenu un faisceau d'électrons. D'autre part le module d'identification de particules s'est avéré peu utile étant donné le but du test sur faisceau. Le module d'*event display* a de ce fait, et en quelque sorte, remplacé ce module.

Le module dédié au calorimètre électromagnétique n'a pas permis de repérer de problème mais a plus été utile à la compréhension de ce détecteur pour l'équipe du SDHCAL. Les différents graphes et les interprétations fournis par l'équipe du ECAL ont principalement étoffé notre connaissance de ce détecteur en situation de prise de données réelles.

L'interface de gestion des processus s'est montrée plus qu'indispensable. L'affichage des journaux des différents processus via cette interface a permis de résoudre de nombreux problèmes de développement tant au niveau de l'implémentation de la librairie pour le SDHCAL que pour l'environnement de développement de DQM4HEP. L'interface de surveillance n'a pas montré de limitation ni de problème particulier. La plupart des fonctionnalités implémentées dans l'interface ont été utilisées par les opérateurs.

Quelques performances

La table 7.1 montre les performances mémoire et CPU des différents processus

Processus	Mémoire virtuelle (KB)	Mémoire résiduelle (KB)	% Mémoire	% CPU
Slow control	619600	256194	3.23	19.75
Analyse ECal	<u>410477</u>	<u>89444</u>	<u>1.13</u>	7.35
Analyse données brutes	580559	221993	2.8	32.4
<i>Event display</i>	545670	237811	3	50.4
Collecteur d'éléments de surveillance	607924	305080	3.72	<u>5.05</u>
Collecteur d'événements physiques 1	558420	270784	3.3	13.57
Collecteur d'événements physiques 2	518524	252332	3.08	7.57
Gestionnaire de <i>run</i>	-	-	-	(0.03)
Convertisseurs SHM	1061870	638328	7.79	7.76

TABLE 7.1 – Tables des performances mémoire et CPU des différents processus déployés lors du test sur faisceau. Les valeurs minimales et maximales pour chaque variable sont indiquées respectivement en souligné et en gras.

Serveur/processus sortant		Serveur/processus entrant	Bande passante (MB/s)
lyosdhcal9/Convertisseurs	→	lyosdhcal10/Collecteurs d'événements physique	12
lyosdhcal10/Collecteurs d'événements physique	→	lyosdhcal7/Modules d'analyse de données	41
lyosdhcal7/Modules d'analyse de données	→	lyosdhcal10/Collecteur d'éléments de surveillance	12

TABLE 7.2 – Tables des performances réseaux des différents serveurs utilisés lors du test sur faisceau

mis en jeu dans le système de surveillance lors du test sur faisceau. Les statistiques sont obtenues en utilisant la commande `pidstat` lors d'une acquisition de pions de 60 GeV et moyennées sur 2 minutes. A noter que parmi les processus, le module d'identification de particules n'apparaît pas, étant donné que son déploiement ne s'est pas avéré nécessaire lors de ce test sur faisceau. Quant au gestionnaire de *run*, la commande `pidstat` n'a pas pu estimer certaines de ses statistiques. Mais étant donné son rôle, les valeurs attendues restent très faibles comparé aux autres processus mis en jeu dans ce déploiement.

La mémoire virtuelle représente la quantité totale de mémoire utilisée par le processus (mémoire RAM, mémoire partagée, disque, etc ...), par opposition à la mémoire résiduelle ou mémoire physique représentant la mémoire vive utilisée par le processus.

Les valeurs maximales de mémoire virtuelle, mémoire vive et pourcentage mémoire utilisés sont attribuées au processus de conversion des données brutes. En effet, en plus des données brutes reçues via la mémoire partagée, les événements sont convertis au format LCIO, augmentant de façon considérable l'usage de la mémoire. Le pourcentage de CPU maximal utilisé est quant à lui décerné au module d'analyse de données brutes, étant donné le nombre important de cartes de comptage à remplir.

Les valeurs minimales de mémoire sont attribuées au module d'analyse de données du ECal. Le faible nombre de couches composant le détecteur mais surtout le fait qu'une faible partie de ces couches étaient efficaces font que peu d'histogrammes sont finalement remplis. Les requêtes peu fréquentes au collecteur d'éléments de surveillance, malgré la collecte des éléments de surveillance envoyés par les modules, en font finalement le processus consommant le moins de CPU.

La table 7.2 montre quant à elle, et à titre informatif, les statistiques réseaux des différents serveurs obtenues sur une période d'un *spill* ($\simeq 40$ secondes).

4 Conclusion

Avec la complexification des détecteurs en physique des particules, les systèmes de surveillance de qualité de données sont plus qu'indispensables pour le bon déroulement des prises de données. Ces logiciels sont aussi un premier pas vers la certification des données avant une analyse hors ligne plus poussée. De tels systèmes ont déjà été bâtis par différentes expériences, notamment au LHC, où chacune des expériences a développé son propre système. Cependant, les systèmes existants sont fortement couplés au principe de fonctionnement du détecteur et à la structure des données sous-jacentes les rendant très difficilement exploitables par d'autres expériences telles que l'ILC.

Le logiciel DQM4HEP propose une implémentation de ce type de système. Il repose sur un système de *plugin* qui le rend hautement modulaire. Cette modularité permet notamment d'abstraire la gestion des événements physique au sein du système, et de ce fait, de proposer un environnement de développement générique pour les analyses de données en ligne. La séparation des différentes tâches définies par ce logiciel et leur lien via une couche réseau, en fait aussi un système robuste au changement. Une application de récupération et reconstruction de données en ligne donne la possibilité, grâce au système de *plugin*, de réinterpréter et convertir ces données dans un format défini par l'utilisateur. De même, des applications dédiées aux analyses de données utilisent ce système pour insérer des analyses utilisateurs. Elles permettent au physicien développeur de se focaliser sur l'analyse elle-même au moyen d'une interface dédiée et non sur la gestion d'un code complexe au sein d'un système complexe. Leur but est aussi de réduire la taille des données, souvent importante, à quelques éléments (histogrammes, graphes, scalaires, etc...) collectés

sur un serveur et mis à disposition d'autres utilisateurs. Enfin quelques outils spécifiques à l'expérience ILC ont été développés dans un paquet à part, DQM4ILC, fournissant notamment un *streamer* d'événements pour le format LCIO.

Les nombreux processus impliqués dans un déploiement par l'utilisateur sont entièrement gérés par le biais d'une interface graphique. Une fois le système complet installé sur un serveur, cette interface gèrera la création et l'arrêt des processus ainsi qu'un accès à leur état sur différentes machines depuis un poste client. Une seconde interface graphique a aussi été développée et est entièrement dédiée à la visualisation des éléments de surveillance produits par les analyses de données. Elle est implémenté pour assurer plusieurs connexions vers des serveurs de collecte d'éléments. Elle propose plusieurs modes de mise à jour, soit manuelle pour récupérer des éléments à la demande de l'utilisateur, soit en mode automatique lorsque les éléments sont mis à jour par l'analyse de données.

Le logiciel étant en réalité plus un environnement de développement, il est nécessaire de fournir un code minimal pour assurer son fonctionnement d'un bout à l'autre. Une implémentation spécifique aux prototypes SDHCAL et SiWECAL a été développée pour le test en faisceau combiné du mois de juin 2016. Elle propose une série de convertisseurs, changeant les données brutes des deux détecteurs vers le format LCIO ainsi que plusieurs analyses destinées à comprendre ces deux détecteurs. Le déploiement complet des différents processus a été exposé et un bilan positif des différents tests effectués a été établi. Les analyses ont permis de détecter rapidement certains problèmes avant ou pendant les prises de données. Les relevés de performance mémoire, CPU et réseau sont cohérents avec le rôle de chacune des applications et serveurs déployés.

DQM4HEP est en bonne voie mais des améliorations majeures pourraient être apportées. Le lourd déploiement du système effectué lors du test en faisceau suggère une application graphique supplémentaire, capable de donner en direct les performances du système déployé. Des questions comme "Mon collecteur d'événement reçoit-il des événements? Si oui, combien durant la dernière minute?" ne nécessiteraient pas de chercher dans les fichiers journaux du collecteur en question. D'autres améliorations mineures peuvent aussi être apportées telles que l'ajout de fonctionnalités au niveau des interfaces vers les différents collecteurs ou encore la possibilité de changer certains paramètres importants dans les modules d'analyse en cours d'exécution via l'interface réseau.

Annexe A

Paramètres de l’algorithme ArborPFA pour le prototype SDHCAL

Cette annexe a pour but de décrire les principaux paramètres de l’algorithme ArborPFA dédié au SDHCAL. Étant donné leur nombre important, les valeurs précisées dans cette annexe n’ont pas subi d’optimisation. Cependant, lors des différentes études présentées dans ce document, les résultats montrés présentaient des incertitudes systématiques calculées sur la base de variations de certains de ces paramètres. Le nom des paramètres est annoté avec le même symbole utilisé dans la description des algorithmes de la section 1 du chapitre 5.

Création des vertex

Paramètre	valeur
MaxVertexSize	4
IntraPseudoLayerDistance	11 mm

- MaxVertexSize
→ La taille maximale d’un groupe de hits pour former un vertex. Dans le cas contraire, chaque hit du groupe forme un vertex à part entière.
- IntraPseudoLayerDistance
→ La distance entre deux cellules voisines utilisées pour former des groupe de hits. Cette distance correspond à la distance entre deux cellules voisines dans le SDHCAL.

Étiquetage des vertex appartenant aux segment de traces

- MaxNNighbors ($N_{mip,cut}$)
→ Le nombre maximal de vertex voisins au sein d’un même pseudo-plan.
- IntraPseudoLayerNeighbourDistance (Δ_{mip})
→ La distance maximale entre deux vertex voisins lors du comptage des voisins

Paramètre	valeur
MaxNNeighbors ($N_{mip,cut}$)	6
IntraPseudoLayerNeighbourDistance (Δ_{mip})	50 mm

Connexion des segments de traces primaires

Paramètre	valeur
ConnectionDistance (Δ_{max})	110 mm
BackwardConnectorWeight (w_{bck})	2
ForwardConnectorWeight (w_{fwd})	3
OrderParameterAnglePower (p_θ)	1
OrderParameterDistancePower (p_Δ)	5
MaxNEmptyConsecutivePseudoLayers	3

- ConnectionDistance (Δ_{max})
→ La distance maximale entre deux vertex à connecter
- BackwardConnectorWeight (w_{bck})
→ Le poids assigné aux connexions vers l'arrière dans le calcul de la direction de référence
- ForwardConnectorWeight (w_{fwd})
→ Le poids assigné aux connexions vers l'avant dans le calcul de la direction de référence
- OrderParameterAnglePower (p_θ)
→ Le paramètre de puissance sur l'angle dans le calcul du paramètre κ lors du nettoyage des connexions
- OrderParameterDistancePower (p_Δ)
→ Le paramètre de puissance sur la distance dans le calcul du paramètre κ lors du nettoyage des connexions
- MaxNEmptyConsecutivePseudoLayers
→ Le nombre maximal de pseudo-plans consécutifs sans hit

Création des connexions 1

Paramètre	valeur
ConnectionDistance (Δ_{max})	45 mm

- ConnectionDistance
→ La distance maximale entre deux vertex à connecter

Nettoyage des connexions 1

Paramètre	valeur
BackwardConnectorWeight (w_{bck})	2
ForwardConnectorWeight (w_{fwd})	2
OrderParameterAnglePower (p_θ)	1
OrderParameterDistancePower (p_Δ)	5
ForwardReferenceDirectionDepth (δ)	1

- BackwardConnectorWeight (w_{bck})
→ Le poids assigné aux connexions vers l'arrière dans le calcul de la direction de référence
- ForwardConnectorWeight (w_{fwd})
→ Le poids assigné aux connexions vers l'avant dans le calcul de la direction de référence
- OrderParameterAnglePower (p_θ)
→ Le paramètre de puissance sur l'angle θ dans le calcul du paramètre κ
- OrderParameterDistancePower (p_Δ)
→ Le paramètre de puissance sur la distance Δ dans le calcul du paramètre κ
- ForwardReferenceDirectionDepth (δ)
→ La profondeur maximale vers l'avant utilisée dans le calcul de la direction de référence

Création des connexions 2

Paramètre	valeur
ConnectionDistance (r_{max})	65 mm
ConnectionAngle (θ_{max})	0.7 rad

- ConnectionDistance (r_{max})
→ La distance maximale entre deux vertex à connecter
- ConnectionAngle (θ_{max})
→ L'angle maximal entre deux connexions

Nettoyage des connexions 2

- BackwardConnectorWeight (w_{bck})
→ Le poids assigné aux connexions vers l'arrière dans le calcul de la direction de référence
- ForwardConnectorWeight (w_{fwd})
→ Le poids assigné aux connexions vers l'avant dans le calcul de la direction de référence

Paramètre	valeur
BackwardConnectorWeight (w_{bck})	0.1
ForwardConnectorWeight (w_{fwd})	5
OrderParameterAnglePower (p_θ)	1
OrderParameterDistancePower (p_Δ)	5
ForwardReferenceDirectionDepth (δ)	2

- OrderParameterAnglePower (p_θ)
→ Le paramètre de puissance sur l'angle θ dans le calcul du paramètre κ
- OrderParameterDistancePower (p_Δ)
→ Le paramètre de puissance sur la distance Δ dans le calcul du paramètre κ
- ForwardReferenceDirectionDepth (δ)
→ La profondeur maximale vers l'avant utilisée dans le calcul de la direction de référence

Association entre traces et clusters

Paramètre	valeur
TrackToClusterDistanceCut1 ($\Delta_{tc,1}$)	75 mm
TrackToClusterDistanceCut2 ($\Delta_{tc,2}$)	55 mm
FirstInteractingPseudoLayerNSeedCut (N_v)	15
TrackToClusterNPseudoLayersCut (N_{pl})	3
TrackClusterCompCut	3

- TrackToClusterDistanceCut1 ($\Delta_{tc,1}$)
→ La distance entre un vertex racine et la projection d'une trace sur la face avant du calorimètre. Cette distance est utilisée quand une pré-interaction a été détectée.
- TrackToClusterDistanceCut2 ($\Delta_{tc,2}$)
→ La distance réduite entre un vertex racine et la projection d'une trace sur la face avant du calorimètre. Cette distance est utilisée lorsque aucune pré-interaction n'a été détectée.
- FirstInteractingPseudoLayerNSeedCut (N_v)
→ Le nombre maximal de vertex racines à une distance $\Delta_{tc,1}$ à comptabiliser pour détecter une pré-interaction.
- TrackToClusterNPseudoLayersCut (N_{pl})
→ Le nombre maximal de pseudo-plans primaires à considérer pour une association.
- TrackClusterCompCut
→ La coupure sur la compatibilité $E-p$ avant d'associer une trace à un cluster.

Fusion des arbres neutres

Paramètre	valeur
SeedSeparationMerge ($\Delta_{pl,v}$)	25 mm

- SeedSeparationMerge ($\Delta_{pl,v}$)
→ La distance maximale entre vertex racines au sein d'un même pseudo-plan

Association d'arbres pointant

Paramètre	valeur
MinClusterSize (N_v)	4
MinNPseudoLayers (N_{pl})	4
FitToBarycentreDistanceCut ($d_{proj,max}$)	30 mm
FitToBarycentreAngleCut ($\theta_{p,f,max}$)	$\frac{\pi}{6}$ rad
FitToFitDistanceCut ($d_{cross,max}$)	20 mm
FitDistanceApproachCut ($d_{close,max}$)	20 mm
Chi2AssociationCut	1

- MinClusterSize (N_v)
→ La taille minimale d'un cluster (pré-sélection)
- MinNPseudoLayers (N_{pl})
→ Le nombre minimal de pseudo-plans sur lequel s'étend un cluster (pré-sélection)
- FitToBarycentreDistanceCut ($d_{proj,max}$)
→ La distance barycentrique maximale
- FitToBarycentreAngleCut ($\theta_{p,f,max}$)
→ L'angle maximal lors du calcul de la méthode barycentrique
- FitToFitDistanceCut ($d_{cross,max}$)
→ La distance de croisement des axes maximale
- FitDistanceApproachCut ($d_{close,max}$)
→ La distance minimale d'approche (dca) maximale
- Chi2AssociationCut
→ Coupure sur la compatibilité $E - p$ lors des associations entre clusters

Fusion des petits arbres neutres

Paramètre	valeur
MaximumDaughterSize (N_v)	20
LargeDistanceCut	1000 mm

- MaximumDaughterSize (N_v)
→ La taille maximale des petites arbres
- LargeDistanceCut
→ La distance maximale entre deux clusters

Annexe B

Gestion des *plugins*

Le but de cette annexe est d'expliquer le mécanisme d'insertion des *plugins* dans l'environnement DQM4HEP de montrer, par un exemple simple, comment un plugin est inséré dans un programme.

Le mécanisme des plugins constitue le cœur du logiciel DQM4HEP. Il permet d'insérer des objets (ou des classes plutôt) à l'intérieur d'un programme à partir d'un code extérieur au programme. Ce système repose sur deux classes `DQMPluginManager` et `DQMPlugin`, et une macro pré-processeur `DQM_PLUGIN_DECL`. La classe singleton [26] `DQMPluginManager` est la principale interface entre les *plugins* et le développeur. En interne, elle assure le chargement des *plugins* via des bibliothèques partagées ainsi que leur gestion au sein du programme appelant. Pour le développeur, cette classe est une interface de chargement et d'accès aux *plugins*. La classe `DQMPlugin` est utilisée en interne pour abstraire les *plugins* de façon transparente. Enfin, la macro `DQM_PLUGIN_DECL` est utilisée dans la bibliothèque externe utilisateur pour déclarer un plugin en particulier (voir portion de code B.1).

```
1 #define DQM_PLUGIN_DECL( ClassName , ClassStr )\  
2     class DQMPlugin_##ClassName :\  
3         public dqm4hep::DQMPlugin,\  
4         public ClassName\  
5     {\  
6     public:\  
7         DQMPlugin_##ClassName () :\  
8             dqm4hep::DQMPlugin(ClassStr), ClassName() {}\  
9         DQMPlugin *create() const {\  
10             return new DQMPlugin_##ClassName (false) ;\  
11         }\  
12     };\  
13     DQMPlugin_##ClassName instance_DQMPlugin_##ClassName;
```

Code B.1 – Macro pré-processeur permettant d'insérer un plugin de façon dynamique dans un programme.

Afin de mieux illustrer le fonctionnement de ce système, nous nous appuyerons sur des portions de codes très simples. Les portions de code B.2 et B.3 implémentent un *plugin* utilisateur dont la classe porte le nom `MyPluginClass` et qui sera inséré sous le nom "`MyPlugin`". La classe ne nécessite pas d'héritage puisque la macro d'insertion `DQM_PLUGIN_DECL` se charge de cette étape. Cela permet au développeur de

réduire les dépendances en héritage et de transformer sa classe en *plugin* DQM4HEP uniquement au moment de l'insertion de cette dernière.

```

1 // file MyPlugin.h
2
3 class MyPluginClass
4 {
5 public:
6     MyPluginClass();
7     void print() const;
8
9 private:
10    int m_anInt;
11 };

```

Code B.2 – Fichier *MyPlugin.h* contenant la déclaration de la classe *plugin* utilisateur.

```

1 // file MyPlugin.cc
2
3 #include <MyPlugin.h>
4 #include <dqm4hep/DQMPlugin.h>
5 #include <iostream>
6
7 // plugin insert in the framework
8 DQM_PLUGIN_DECL("MyPlugin", MyPluginClass)
9
10 MyPluginClass::MyPluginClass() :
11     m_myInt(42) {
12 }
13
14 void MyPluginClass::print() const {
15     std::cout << "integer : " << m_myInt << std::endl;
16 }

```

Code B.3 – Fichier *MyPlugin.cc* contenant l'implémentation de la classe *plugin* utilisateur et l'appel à la macro d'insertion du *plugin*.

```

1 {
2     // ...
3     DQMPluginManager *pMgr = DQMPluginManager::instance();
4     pMgr->loadlibrary("libMyPlugin.so");
5     // or export DQM4HEP_PLUGIN_DLL=libMyPlugin.so in a shell
6     // and pMgr->loadlibraries();
7
8     MyPluginClass *pPlugin =
9         pMgr->createPluginClass<MyPluginClass>("MyPlugin");
10    pPlugin->print();
11    delete pClass;
12    // ...
13 }

```

Code B.4 – Exemple de récupération de la classe *MyPluginClass* depuis un programme externe.

L'insertion du *plugin* s'effectue à la ligne 8, seule réelle dépendance du système de plugin DQM4HEP pour l'utilisateur, mis à part les fichiers à inclure. Ce code sera compilé au sein d'une librairie que l'on nommera pour l'exemple `libMyPlugin.so`. La portion de code B.4 montre un exemple de chargement de la librairie contenant notre *plugin*, depuis un code externe (i.e. un exécutable DQM4HEP). Le chargement de la librairie est effectué à la ligne 4 et la récupération de notre classe exemple à la ligne 9. La deuxième solution pour charger la librairie est d'utiliser la variable d'environnement `DQM4HEP_PLUGIN_DLL` en y exportant la liste des librairies, séparées par le symbole ':' et d'appeler la ligne 6 à la place. Cette méthode est appliquée au début de la plupart des exécutables fournis par le logiciel DQM4HEP.

Annexe C

Implémentation d'un module d'analyse de données

Pour analyser les événements provenant du système d'acquisition, l'utilisateur doit implémenter un module d'analyse de données en ligne à l'aide de la classe `DQMAalysisModule`. Le module sera compilé au sein d'une librairie partagée et chargé par une application dédiée. Pour mieux illustrer son implémentation, nous allons fournir un contexte de prise de données avec un simple détecteur et son format de données.

1 Le contexte : détection de muons cosmiques

Prenons par exemple un système de détection de muons cosmiques composé de deux scintillateurs plastiques avec des photo-multiplicateurs pour la lecture des signaux. Le système d'acquisition détecte un événement lorsque les deux scintillateurs ont détecté une particule en coïncidence. Le format de l'événement est défini avec la structure `Event_t` suivante :

```
1 // Event.h
2 struct Event_t {
3     uint32_t    m_adc1;
4     uint32_t    m_adc2;
5     uint64_t    m_time;
6 };
```

avec `m_adc1` le comptage ADC brut issu du photomultiplicateur 1, `m_adc2` le comptage ADC brut issu du photomultiplicateur 2 et `m_time` l'horodatage de l'événement en micro-secondes depuis le début de l'acquisition. Le module d'analyse recevra ces événements par l'intermédiaire d'un collecteur d'événements que l'on nommera "EVT_COL". Les signaux de *runs* émis par le système d'acquisition sont acheminés jusqu'à un gestionnaire de *run* que l'on nommera "RUN_CTRL". Le module d'analyse produira des éléments de surveillance qui seront envoyés au collecteur d'éléments de surveillance que l'on appellera "ME_COL".

2 Transfert des événements

Afin de transférer les événements du système d'acquisition vers le collecteur, puis du collecteur vers le module d'analyse, l'interface `DQMEventBase<T>` permettant de gérer notre structure `Event_t` est implementée ainsi que la classe de sérialisation correspondante, `EventStreamer_t`. L'interface d'événement est implémentée de la manière suivante :

```

1 // DQMEvent_t.h
2 #include <Event.h>
3
4 class DQMEvent_t : public DQMEventBase<Event_t>
5 {
6 public:
7     DQMEvent_t() : m_pEvent(0) {
8     }
9
10    ~DQMEvent_t() {
11        this->clear();
12    }
13
14    Event_t *getEvent() const {
15        return m_pEvent;
16    }
17
18    void setEvent(Event_t *pEvent) {
19        this->clear();
20        m_pEvent = pEvent;
21    }
22
23    void clear() {
24        if(0 != m_pEvent && this->isOwner())
25            delete m_pEvent;
26
27        m_pEvent = NULL;
28    }
29
30 private:
31     Event_t *m_pEvent;
32 };

```

La sérialisation est implémentée de la manière suivante :

```

1 // EventStreamer_t.h
2
3 class EventStreamer_t : public DQMEventStreamer
4 {
5 public:
6     DQMEvent *createEvent() const;
7     StatusCode write(const DQMEvent *const pEvent, IODevice *pDevice)
8     ;
9     StatusCode read(DQMEvent *&pEvent, IODevice *pDevice);
10    StatusCode write(const DQMEvent *const pObject, const string &
11        subEventIdentifier, IODevice *pDevice);

```

```
1 // EventStreamer_t.cc
2
3 #include <DQMEvent_t.h>
4 #include <EventStreamer_t.h>
5
6 // plugin insert in framework
7 DQM_PLUGIN_DECL(EventStreamer_t, "PMEventStreamer")
8
9 // create an event containing an Event_t structure
10 DQMEvent_t *EventStreamer_t::createEvent() const {
11
12     DQMEvent_t *pEvent = new DQMEvent_t();
13     pEvent->setEvent(new Event_t);
14
15     return pEvent;
16 }
17
18 // write the Event_t structure in the device
19 StatusCode EventStreamer_t::write(const DQMEvent *const pDQMEvent,
20     IODevice *pDevice) {
21
22     const Event_t *pEvent = pDQMEvent->getEvent<DQMGenericEvent>();
23
24     pDevice->write<uint32_t>(&pEvent->m_adc1);
25     pDevice->write<uint32_t>(&pEvent->m_adc2);
26     pDevice->write<uint64_t>(&pEvent->m_time);
27
28     return STATUS_CODE_SUCCESS;
29 }
30
31 // read the Event_t structure from the device
32 StatusCode EventStreamer_t::read(DQMEvent *&pEvent, IODevice *
33     pDevice) {
34
35     pEvent = this->createEvent();
36     Event_t *pLocalEvent = pDQMEvent->getEvent<Event_t>();
37
38     pDevice->read<uint32_t>(&pEvent->m_adc1);
39     pDevice->read<uint32_t>(&pEvent->m_adc2);
40     pDevice->read<uint64_t>(&pEvent->m_time);
41
42     return STATUS_CODE_SUCCESS;
43 }
44
45 // normally, write a part of the Event_t structure in the device
46 // but to simplify, write the whole event
47 StatusCode EventStreamer_t::write(const DQMEvent *const pEvent,
48     const string &subEventIdentifier, IODevice *pDevice) {
49
50     return this->write(pEvent, pDevice);
51 }
52 }
```

Le code de la sérialisation est très explicite : les trois champs de la structure `Event_t` sont lus ou écrits dans l'interface `IODevice`. Pour simplifier cet exemple,

l'écriture d'un sous-événement n'a pas été implémentée (ligne 45); un événement complet est écrit à la place. Tous ces fichiers sont compilés au sein d'une librairie que l'on nommera `libEventStreamer.so`. La classe qui sérialise les événements est déclarée en *plugin* (ligne 7) et sera utilisée pour décoder les événements à leur réception dans l'application.

3 Le module d'analyse

Le module d'analyse de données, que l'on appellera `PModule`, reçoit des événements de type `Event_t`, au fur et à mesure que des muons cosmiques traversent les scintillateurs. Le code en est donné ci-après :

```

1 // PModule.h
2 #include <dqm4hep/DQMAnalysisModule.h>
3
4 class PModule : public DQMAnalysisModule
5 {
6 public:
7     PModule();
8
9     // from DQMAnalysisModule
10    StatusCode initModule();
11    StatusCode readSettings(const TiXmlHandle xmlHandle);
12    StatusCode processEvent(DQMEvent *pEvent);
13    StatusCode startOfCycle();
14    StatusCode endOfCycle();
15    StatusCode startOfRun(DQMRun *pRun);
16    StatusCode endOfRun(DQMRun *pRun);
17    StatusCode endModule();
18
19 private:
20     DQMMonitorElementPtr    m_adcDistribution1;
21     DQMMonitorElementPtr    m_adcDistribution2;
22     DQMMonitorElementPtr    m_timeDistribution1;
23
24     bool m_doPrinting;
25 };

```

```

1 // PModule.cc
2
3 #include <PModule.h>
4
5 // -- dqm4hep headers
6 #include "dqm4hep/DQMMonitorElement.h"
7 #include "dqm4hep/DQMRun.h"
8 #include "dqm4hep/DQMXmlHelper.h"
9 #include "dqm4hep/DQMModuleApi.h"
10 #include "dqm4hep/DQMPlugin.h"
11
12 // plugin insert in framework
13 DQM_PLUGIN_DECL(PModule, "PModule")
14

```

```
15 PModule::PModule() {
16     setVersion(1, 0, 0);
17     setDetectorName("2PMs for muons");
18 }
19
20 // read some parameters and book histograms
21 StatusCode PModule::readSettings(const TiXmlHandle xmlHandle) {
22
23     // read a parameter
24     DQMXmlHelper::readParameterValue(xmlHandle, "DoPrinting",
25         m_doPrinting);
26
27     // book histograms
28     DQMXmlHelper::bookMonitorElement(this, xmlHandle,
29         "ADCDistribution", m_adcDistribution1, {"pm_id" : "1"});
30
31     DQMXmlHelper::bookMonitorElement(this, xmlHandle,
32         "ADCDistribution", m_adcDistribution2, {"pm_id" : "2"});
33
34     DQMXmlHelper::bookMonitorElement(this, xmlHandle,
35         "TimeDistribution", m_timeDistribution);
36
37     return STATUS_CODE_SUCCESS;
38 }
39
40 // init the module. Do some prints
41 StatusCode PModule::initModule() {
42
43     DQMModuleApi::cd(this);
44
45     // print directory structure of monitor elements in shell
46     if(m_doPrinting)
47         DQMModuleApi::ls(this, true);
48
49     return STATUS_CODE_SUCCESS;
50 }
51
52 StatusCode PModule::startOfRun(dqm4hep::DQMRun *const pRun) {
53     cout << "Start of run : " << *pRun << endl;
54     return STATUS_CODE_SUCCESS;
55 }
56
57 StatusCode PModule::startOfCycle() {
58     cout << "Starting new cycle !" << endl;
59     return STATUS_CODE_SUCCESS;
60 }
61
62 // get the real event implementation and fill booked histograms
63 StatusCode RandomModule::processEvent(DQMEvent *const pDQMEvent) {
64
65     Event_t *pEvent = pDQMEvent->getEvent<Event_t>();
66
67     m_adcDistribution1->get<TH1>()->Fill(pEvent->m_adc1);
68     m_adcDistribution2->get<TH1>()->Fill(pEvent->m_adc2);
```

```

68   m_timeDistribution->get<TH1>()->Fill(pEvent->m_time);
69
70   return STATUS_CODE_SUCCESS;
71 }
72
73 StatusCode PMModule::endOfCycle() {
74   return STATUS_CODE_SUCCESS;
75 }
76
77 StatusCode PMModule::endOfRun() {
78   cout << "End of run : " << *pRun << endl;
79   return STATUS_CODE_SUCCESS;
80 }
81
82 StatusCode PMModule::endModule() {
83   return STATUS_CODE_SUCCESS;
84 }

```

Deux fonctions sont principalement utilisées dans cet exemple. La fonction `readSettings()` permet de configurer le module à l'aide d'un objet représentant une balise XML. Dans cette fonction, un paramètre est lu et trois éléments de surveillance sont alloués sous les références "ADCDistribution" (pour deux éléments) et "TimeDistribution". La fonction `processEvent()` est appelée par l'application lorsque un événement est reçu et permet de l'analyser. L'analyse est ici assez courte : on remplit les trois histogrammes avec les trois champs fournis par la structure `Event_t`.

4 La configuration XML

Une exemple complet de configuration XML du module implémenté précédemment est donné ci-après :

Code C.1 – Fichier XML utilisé pour configurer une application de module d'analyse de données

```

<dqm4hep>
  <archiver open="true" directory="/tmp" />

  <network>
    <runcontrol type="DimRunControlClient" name="RUN_CTRL" />

    <eventcollector type="DimEventClient">
      <parameter name="collectorName"> EVT_COL </parameter>
      <parameter name="maxQueueSize"> 100 </parameter>
      <parameter name="streamerName"> PMSStreamer </parameter>
    </eventcollector>

    <monitorelementcollector name="ME_COL"/>
  </network>

```

```

<cycle type="EventCounterCycle" value="10" timeout="20" />

<module type="PModule" name="PhotoMultiModule">

  <parameter name="DoPrinting"> true </parameter>

  <monitorElement
    ID="ADCDistribution"
    name="ADCDistribution${pm_id}" path="/ADC/" type="TH1F"
    title="ADC distribution of PM ${pm_id}"
    nBins="200" min="0" max="2000"
    drawOption="B" resetPolicy="EndOfRun"/>

  <monitorElement
    ID="TimeDistribution"
    name="TimeDistribution" path="/Timing/" type="TH1F"
    title="Time distribution of muon detection \nfrom start of run
      (unit ms)"
    nBins="200000" min="0" max="200000"
    drawOption="B" resetPolicy="EndOfRun"/>

</module>

</dqm4hep>

```

La première balise `<archiver>` configure l'archivage des éléments de surveillance. Une archive sera écrite à chaque *run* dans le dossier `"/tmp"`. La série de balises suivantes contenues dans `<network>`, configure la partie réseau de l'application. La configuration comprend le gestionnaire de *run*, le collecteur d'événements et le collecteur d'éléments de surveillance. Dans l'interface vers le collecteur d'événements, on retrouve notre plugin pour la sérialisation des événements (`PMEventStreamer`). Le cycle (balise `<cycle>`) est configuré ici sur un comptage d'événements : au bout de 10 événements, le cycle se termine. Si aucun événement n'est reçu durant 20 secondes, le cycle se termine aussi.

La dernière partie de la configuration est dédiée au module utilisateur (balise `<module>`). Le type correspond au nom spécifié à l'enregistrement du plugin dans la librairie. Le nom sera celui qui apparaîtra sur toutes les interfaces graphiques en bout de chaîne. Le paramètre "DoPrinting" est ensuite lu par l'utilisateur (ligne 24 de l'implémentation du module). La configuration correspondant aux lignes 28, 31 et 34 de l'implémentation du module correspondent à la lecture des deux balises `<monitorElement>` restantes. La clé utilisée dans le code réfère à l'attribut ID de ces balises et doit être unique pour chaque élément de surveillance dans le fichier XML. Le reste des attributs de la balise `<monitorElement>` configure l'élément de surveillance en interne et est expliqué avec plus de détails dans la section 2.5.1 du chapitre 7. Le module est compilé au sein d'une librairie que l'on appellera

libPModule.so. Le fichier XML s'appellera `steeringFile.xml`.

L'application est finalement lancée de la façon suivante :

```
$ export DQM4HEP_PLUGIN_DLL=libEventStreamer.so:libPModule.so  
$ dqm4hep_start_analysis_module -f steeringFile.xml
```

Bibliographie

- [1] Precise determination of the mass of the Higgs boson and studies of the compatibility of its couplings with the standard model. Technical Report CMS-PAS-HIG-14-009, CERN, Geneva, 2014. URL <https://cds.cern.ch/record/1728249>.
- [2] E. Abat *et al.* Study of energy response and resolution of the ATLAS barrel calorimeter to hadrons of energies from 20-GeV to 350-GeV. *Nucl.Instrum.Meth.*, A621 :134–150, 2010. doi :10.1016/j.nima.2010.04.054.
- [3] Halina Abramowicz *et al.* The International Linear Collider Technical Design Report - Volume 4 : Detectors. 2013.
- [4] Chris Adolphsen, Maura Barone, Barry Barish, Karsten Buesser, Philip Burrows, *et al.* The International Linear Collider Technical Design Report - Volume 3.II : Accelerator Baseline Design. 2013.
- [5] G Alexander, T Behnke, S Bertolucci, RD Heuer, and R Settles. Tesla technical design report, part iv : A detector for tesla. *DESY-01-011, DESY-2001-011, DESY-01-011D, DESY-2001-011D, DESY-TESLA-2001-23, DESY-TESLA-FEL-2001-05, ECFA-2001-209*, 2001.
- [6] Tim and Barklow, Keisuke Fujii, Yuanning Gao, Andre Hoang, Shinya Kanemura, Jenny List, Heather E Logan, Andrei Nomerotski, Maxim Perelstein, Michael E Peskin, Roman Pöschl, Jürgen Reuter, *et al.* The International Linear Collider Technical Design Report - Volume 2 : Physics. Technical Report arXiv :1306.6352. ANL-HEP-TR-13-20. BNL-100603-2013-IR. IRFU-13-59. CERN-ATS-2013-037. Cockcroft-13-10. CLNS-13-2085. DESY-13-062. FERMILAB-TM-2554. IHEP-AC-ILC-2013-001. ILC-REPORT-2013-040. INFN-13-04-LNF. JAI-2013-001. JINR-E9-2013-35. JLAB-R-2013-01. KEK-Report-2013-1. KNU-CHEP-ILC-2013-1. LLNL-TR-635539. SLAC-R-1004. ILC-HiGrade-Report-2013-003, Geneva, Jun 2013. URL <http://cds.cern.ch/record/1601967>. Comments : See also <http://www.linearcollider.org/ILC/TDR> . The full list of signatories is inside the Report.
- [7] S. Aplin, J. Engels, F. Gaede, N. A. Graf, T. Johnson, and J. McCormick. Lcio : A persistency framework and event data model for hep. In *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2012 IEEE*, pages 2075–2079. Oct 2012. ISSN 1082-3654. doi :10.1109/NSSMIC.2012.6551478.
- [8] G. Baulieu, M. Bedjidian, K. Belkadhi, J. Berenguer, V. Boudry, P. Calabria, S. Callier, E. Calvo Almillo, S. Cap, L. Caponetto, C. Combaret, R. Cor-

- nat, *et al.* Construction and commissioning of a technological prototype of a high-granularity semi-digital hadronic calorimeter. *Journal of Instrumentation*, 10(10) :P10039, 2015. URL <http://stacks.iop.org/1748-0221/10/i=10/a=P10039>.
- [9] Ties Behnke, James E. Brau, Brian Foster, Juan Fuster, Mike Harrison, James McEwan Paterson, Michael Peskin, Marcel Stanitzki, Nicholas Walker, and Hitoshi Yamamoto. The International Linear Collider Technical Design Report - Volume 1 : Executive Summary. 2013.
- [10] M. Bicer, H. Duran Yildiz, I. Yildiz, G. Coignet, M. Delmastro, T. Alexopoulos, C Grojean, S. Antusch, T. Sen, H.-J. He, K. Potamianos, S. Haug, *et al.* First look at the physics case of tlep. *Journal of High Energy Physics*, 2014(1) :164, 2014. doi :10.1007/JHEP01(2014)164. URL <http://dx.doi.org/10.1007/JHEP01%282014%29164>.
- [11] J. C. Brient. Improving the jet reconstruction with the particle flow method : An introduction. In *Calorimetry in particle physics. Proceedings, 11th International Conference, CALOR 2004, Perugia, Italy, March 29-April 2, 2004*, pages 445–451. 2004.
- [12] D. Buskulic, D. Casper, I. De Bonis, D. Decamp, P. Ghez, C. Goy, J.P. Lees, M.N. Minard, P. Odier, B. Pietrzyk, F. Ariztizabal, M. Chmeissani, *et al.* Performance of the ALEPH detector at LEP. *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, 360 :481–506, 1995. URL <http://hal.in2p3.fr/in2p3-00002096>.
- [13] L Caponetto, C Combaret, C de la Taille, F Dulucq, R Kieffer, I Laktineh, N Lumb, L Mirabito, and N Seguin-Moreau. First test of a power-pulsed electronics system on a grpc detector in a 3-tesla magnetic field. *Journal of Instrumentation*, 7(04) :P04009, 2012. URL <http://stacks.iop.org/1748-0221/7/i=04/a=P04009>.
- [14] CMS Luminosity - Public Results. <https://twiki.cern.ch/twiki/bin/view/CMSPublic/LumiPublicResults>. Accessed : 2010-09-30.
- [15] The CALICE collaboration. First results of the CALICE SDHCAL Technological Prototype, CALICE Analysis note CAN-037. 2012.
- [16] The CALICE collaboration. First results of the CALICE SDHCAL Technological Prototype, CALICE Analysis note CAN-037b. 2012.
- [17] The CALICE collaboration. Tracking within Hadronic Showers in the SDHCAL prototype using Hough Transform Technique, CALICE Analysis note CAN-047. 2013.
- [18] The CALICE collaboration, J Repond, J Yu, C M Hawkes, Y Mikami, O Miller, N K Watson, J A Wilson, G Mavromanolakis, M A Thomson, D R Ward, W Yan, *et al.* Design and electronics commissioning of the physics prototype of a si-w electromagnetic calorimeter for the international linear collider. *Journal of Instrumentation*, 3(08) :P08001, 2008. URL <http://stacks.iop.org/1748-0221/3/i=08/a=P08001>.

- [19] The MicroBooNE Collaboration. The Pandora multi-algorithm approach to automated pattern recognition in LAr TPC detectors, MICROBOONE-NOTE-1015-PUB. 2016.
- [20] Z. Deng, Y. Li, Y. Wang, Q. Yue, Z. Yang, D. Boumediene, C. Carloganu, V. Français, G. Cho, D-W. Kim, S.C. Lee, W. Park, *et al.* Resistive plate chamber digitization in a hadronic shower environment. *Journal of Instrumentation*, 11(06) :P06014, 2016. URL <http://stacks.iop.org/1748-0221/11/i=06/a=P06014>.
- [21] Wang Dou, Gao Jie, Xiao Ming, Geng Hui-Ping, Guo Yuan-Yuan, Xu Shou-Yan, Wang Na, An Yu-Wen, Qin Qing, Xu Gang, and Wang Sheng. Optimization parameter design of a circular e+e- higgs factory. *Chinese Physics C*, 37(9) :097003, 2013. URL <http://stacks.iop.org/1674-1137/37/i=9/a=097003>.
- [22] Pierre Fayet. Supersymmetric theories of particles and interactions. *Physica Scripta*, 1987(T15) :46, 1987. URL <http://stacks.iop.org/1402-4896/1987/i=T15/a=006>.
- [23] K. Francis, J. Repond, J. Schlereth, J. Smith, L. Xia, E. Baldolemar, J. Li, S.T. Park, M. Sosebee, A.P. White, J. Yu, G. Eigen, *et al.* Performance of the first prototype of the CALICE scintillator strip electromagnetic calorimeter . *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, 763 :278 – 289, 2014. ISSN 0168-9002. doi :<http://dx.doi.org/10.1016/j.nima.2014.06.039>. URL <http://www.sciencedirect.com/science/article/pii/S0168900214007621>.
- [24] Y. Fukuda, T. Hayakawa, E. Ichihara, K. Inoue, K. Ishihara, H. Ishino, Y. Itow, T. Kajita, J. Kameda, S. Kasuga, K. Kobayashi, Y. Kobayashi, *et al.* Evidence for oscillation of atmospheric neutrinos. *Phys. Rev. Lett.*, 81 :1562–1567, Aug 1998. doi :10.1103/PhysRevLett.81.1562. URL <http://link.aps.org/doi/10.1103/PhysRevLett.81.1562>.
- [25] F. Gaede. Marlin and LCCD : Software tools for the ILC. *Nucl.Instrum.Meth.*, A559 :177–180, 2006. doi :10.1016/j.nima.2005.11.138.
- [26] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns : Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1 edition, November 1994. ISBN 0201633612. URL <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0201633612>.
- [27] Clara Gaspar, M Dönszelmann, and Ph Charpentier. Dim, a portable, light weight package for information publishing, data transfer and inter-process communication. *Computer physics communications*, 140(1) :102–109, 2001.
- [28] Y. Haddad, I. Laktineh, G. Grenier, N. Lumb, and S. Cauwenbergh. High rate resistive plate chamber for LHC detector upgrades . *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, 718(0) :424 – 426, 2013. ISSN 0168-9002. doi :<http://dx.doi.org/10.1016/j.nima.2012.11.029>. URL <http://www.sciencedirect.com>.

- com/science/article/pii/S0168900212013630. Proceedings of the 12th Pisa Meeting on Advanced Detectors La Biodola, Isola d'Elba, Italy, May 20 – 26, 2012.
- [29] A. Heister et al. Measurement of the tau polarisation at lep. *The European Physical Journal C - Particles and Fields*, 20(3) :401–430, 2001. ISSN 1434-6052. doi :10.1007/s100520100689. URL <http://dx.doi.org/10.1007/s100520100689>.
- [30] Ilcsoft homepage. <http://ilcsoft.desy.de>. Accessed : 2010-09-30.
- [31] V. Khachatryan, A. M. Sirunyan, A. Tumasyan, W. Adam, T. Bergauer, M. Dragicevic, J. Erö, C. Fabjan, M. Friedl, R. Frühwirth, V. M. Ghete, C. Hartl, *et al.* Observation of the diphoton decay of the higgs boson and measurement of its properties. *The European Physical Journal C*, 74(10) :3076, 2014. ISSN 1434-6052. doi :10.1140/epjc/s10052-014-3076-z. URL <http://dx.doi.org/10.1140/epjc/s10052-014-3076-z>.
- [32] Lucie Linssen, Akiya Miyamoto, Marcel Stanitzki, and Harry Weerts. Physics and Detectors at CLIC : CLIC Conceptual Design Report. 2012. doi :10.5170/CERN-2012-003.
- [33] Olga Lobban, Aravindhan Sriharan, and Richard Wigmans. On the energy measurement of hadron jets. *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, 495(2) :107 – 120, 2002. ISSN 0168-9002. doi :[http://dx.doi.org/10.1016/S0168-9002\(02\)01615-7](http://dx.doi.org/10.1016/S0168-9002(02)01615-7). URL <http://www.sciencedirect.com/science/article/pii/S0168900202016157>.
- [34] Sameh Mannai, Kais Manai, Eduardo Cortina, and Imad Laktineh. Energy reconstruction in a highly granularity semi-digital hadronic calorimeter. *Journal of Physics : Conference Series*, 664(7) :072033, 2015. URL <http://stacks.iop.org/1742-6596/664/i=7/a=072033>.
- [35] Mokka homepage. <https://11rforge.in2p3.fr/trac/Mokka/>. Accessed : 2010-09-30.
- [36] N Nishimori, R Nagai, M Yamamoto, Y Honda, T Miyajima, H Iijima, M Kuriki, M Kuwahara, S Okumi, T Nakanishi, and R Hajima. Development of a 500-kv photocathode dc gun for erls. *Journal of Physics : Conference Series*, 298(1) :012005, 2011. URL <http://stacks.iop.org/1742-6596/298/i=1/a=012005>.
- [37] G Nooren and E Rocco. A particle counting em calorimeter using maps. *Journal of Physics : Conference Series*, 587(1) :012061, 2015. URL <http://stacks.iop.org/1742-6596/587/i=1/a=012061>.
- [38] K.A. Olive and Particle Data Group. Review of particle physics. *Chinese Physics C*, 38(9) :090001, 2014. URL <http://stacks.iop.org/1674-1137/38/i=9/a=090001>.
- [39] Dorothea Pfeiffer, Georgi Gorine, Hans Reithler, Bartolomej Biskup, Ali Day, Adrian Fabich, Joffrey Germa, Roberto Guida, Martin Jaekel, and Federico

- Ravotti. The Radiation Field in the New Gamma Irradiation Facility (GIF++) at CERN. 2016.
- [40] Qt company. <http://www.qt.io>. V4.7, 2016.
- [41] Manqi Ruan and Henri Videau. Arbor, a new approach of the Particle Flow Algorithm. In *Proceedings, International Conference on Calorimetry for the High Energy Frontier (CHEF 2013) : Paris, France, April 22-25, 2013*, pages 316–324. 2013. URL <https://inspirehep.net/record/1286443/files/arXiv:1403.4784.pdf>.
- [42] A. Steen. *Étude des gerbes hadroniques à l'aide du prototype du calorimètre hadronique semi-digital et comparaison avec les modèles théoriques utilisés dans le logiciel GEANT4*. Ph.D. thesis, Université Claude Bernard Lyon 1, 2015.
- [43] Arnaud Steen. *Étude des gerbes hadroniques à l'aide du prototype du calorimètre hadronique semi-digital et comparaison avec les modèles théoriques utilisés dans le logiciel GEANT4*. Ph.D. thesis, 2015. URL <http://www.theses.fr/2015LY010230>. Thèse de doctorat dirigée par Laktineh, Imad et Grenier, Gérald Physique des particules Lyon 1 2015.
- [44] M.A. Thomson. Particle flow calorimetry and the pandorapfa algorithm. *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, 611(1) :25 – 40, 2009. ISSN 0168-9002. doi :<http://dx.doi.org/10.1016/j.nima.2009.09.009>. URL <http://www.sciencedirect.com/science/article/pii/S0168900209017264>.

Index

- A**
Arbre (graphe) 79, 81, 88
Archiveur 168
- B**
Bethe-Block (formule de) 32
- C**
Collecteur
 d'éléments de surveillance 170
 d'événements physiques 162
Collisionneurs
 CEPC 6, 30
 CLIC 6, 29
 FCC 6, 29
 HL-CEPC 11
 HL-FCC 11
 HL-LHC 10
 ILC 6, 10, 15
 LHC 6, 10
Compatibilité $E - p$ 89
Compatibilité $E - p$ 74
Convertisseur (*ShmProcessor*) .. 162, 179
Cycle, structure en
 EventCounterCycle 169
 EventSizeCycle 169
 TimerCycle 169
- D**
Data source 160, 178
Décteur
 CMS 10
 ILD 19
 SiD 28
dim 157
dimjc 157
DQM Event Data Model 158
DQM4HEP, paquet
 DQM4HEP 159
- DQM4ILC 159
DQMCore 158
DQMVis 158
DQMSDHCAL 177
- E**
Efficacité (SDHCAL) 49
Élément de surveillance 167
Empilement, phénomène d' 10
Énergie (estimateur) 56
Event builder 160
Event streamer 158
- G**
GRPC 39
- H**
Higgs, boson de 6
- I**
Interface graphique utilisateur (GUI)
 Job control 171
 Monitoring 174
 Run control 171
- L**
levbdim 158
- M**
Mécanisme de brisure de symétrie 6
Modèle standard
 Au delà 7
 Théorie 2
Module
 Analysis module 164
 Standalone module 164
Modules (DQMSDHCAL)
 Analyse faisceau 183
 Données brutes 182
 Event display 181

- Particle ID 184
 SiWECal 183
 Slow control 180
 mongoose 158
 Multiplicité (SDHCAL) 49
- P**
- PandoraSDK 68
 Particle Flow 17, 66
 Particle Flow (algorithme)
 ArborPFA (SDHCAL) 79, 84
 CMS 68
 IowaPFA 68
 PandoraPFA 68
 Particules élémentaires
 W^\pm 5
 Z^0 5
 Boson de Higgs 6
 Boson de jauge 4
 Électron 3
 Gluon 4, 5
 Leptons 3
 Muon 3
 Neutrinos 3
 Photon 4
 Quarks 4, 10
 Tau 3
 Performances (DQM4HEP)
 CPU 187
 Mémoire 187
 Réseaux 187
 Performances (PFA)
 ArborPFA (ILD) 147
 ArborPFA (SDHCAL) 93, 98
 PandoraPFA 76
 Polarisation (faisceau) 11
 Power pulsing 42
 Pseudo-plan 70
- R**
- Run control 159
- S**
- SDHCAL 26, 38
 Shared memory 160
 Simulation (SDHCAL) 43, 61
 SiWECal 25, 177
 Supersymétrie 7
 Symétrie , groupe de 2
- T**
- Test de qualité 168
 Test sur faisceau 177
- X**
- xdrlcio 157
 xdrstream 157

