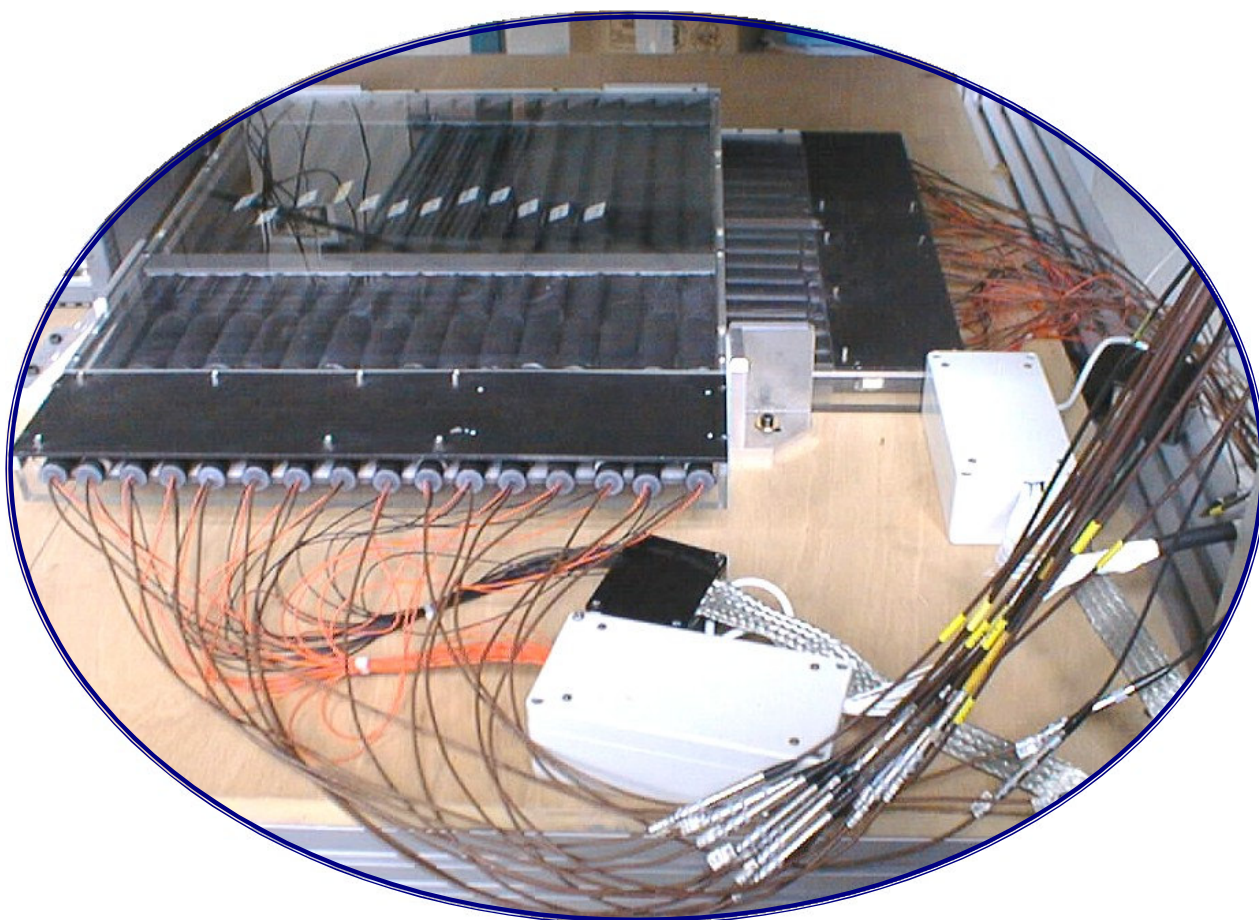


Vincent CHEVALIER

Mise en œuvre d'un banc cosmique



**Stage de DESS « Physique des capteurs et instrumentation »
Du 8 mars au 9 juillet 2004**

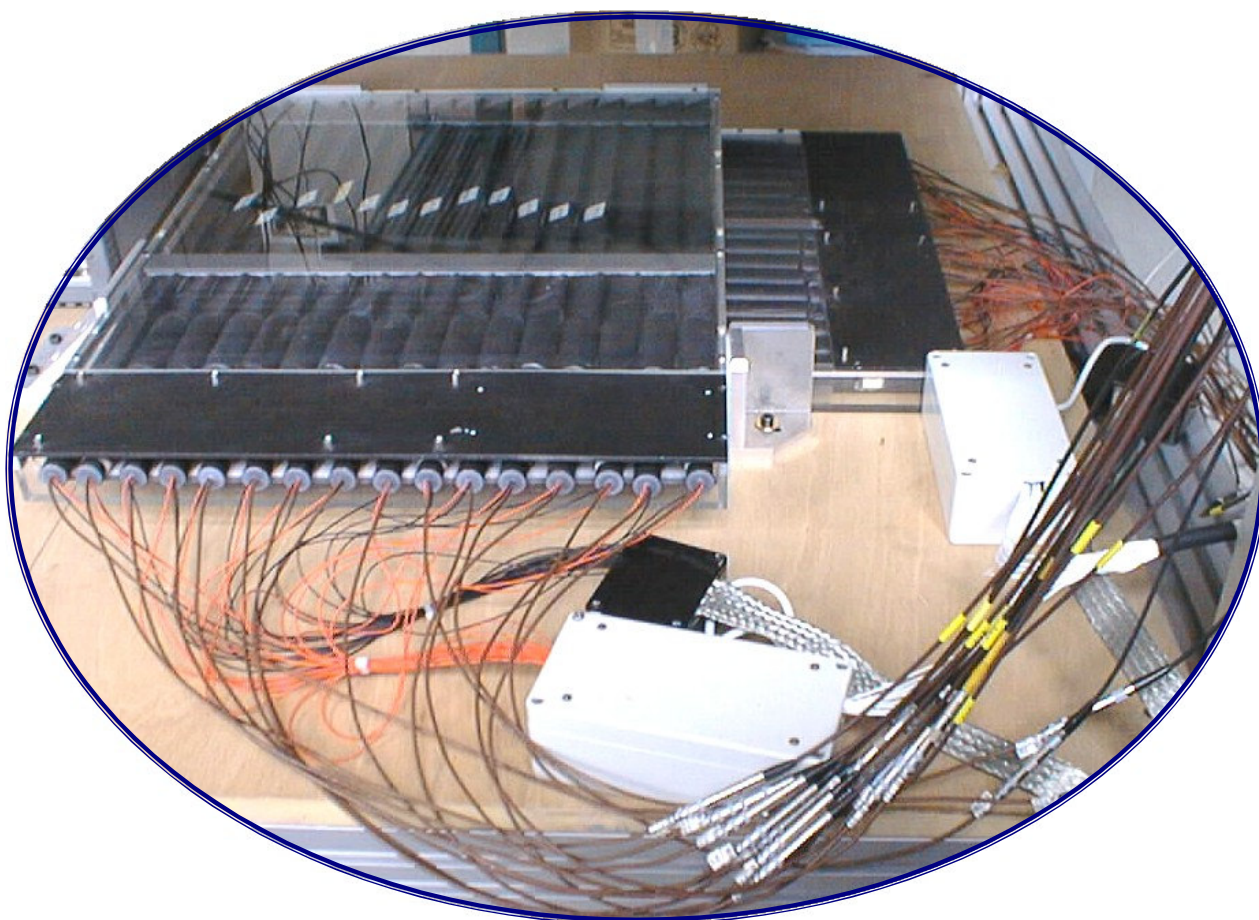
Responsables de stage :

**DESS :
Jack CARIOU**

**LLR :
Jean-Charles VANEL**

Vincent CHEVALIER

Mise en œuvre d'un banc cosmique



**Stage de DESS « Physique des capteurs et instrumentation »
Du 8 mars au 9 juillet 2004**

Responsables de stage :

**DESS :
Jack CARIOU**

**LLR :
Jean-Charles VANEL**

SOMMAIRE

INTRODUCTION	1
PRESENTATION DU LABORATOIRE	2
I. Le Centre National de la Recherche Scientifique (CNRS)	2
Ses activités	2
Son fonctionnement.....	3
II. L'Institut National de Physique Nucléaire et de Physique des Particules (IN ₂ P ₃).....	4
Ses missions	4
Ses laboratoires	5
III. Le Laboratoire Leprince-Ringuet (LLR).....	6
Son personnel	6
Ses activités	7
LES COLLISIONNEURS ET LE PROJET TESLA	9
I. Généralités.....	9
Principe d'un collisionneur	9
Ordres de grandeurs	10
II. Choix du type d'accélérateur.....	10
Selon ses caractéristiques	10
Cas du projet TESLA	12
III. Les détecteurs associés	13
Géométrie des détecteurs	13
Description d'un détecteur cylindrique	14
IV. En résumé.....	15
ANALYSES ET REALISATIONS	16
I. Le sujet de stage	16
Le calorimètre électromagnétique (Ecal pour Electromagnetic calorimeter)	17
Présentation du banc de test	18
II. Réalisations électriques et mécaniques (hard).....	22
Les photomultiplicateurs et scintillateurs.....	22
Adaptation de la carte « trigger »	25
Utilisation des résultats	27
III. Réalisations informatiques (soft)	28
Logiciels utilisés.....	28
Organigrammes des programmes.....	29
Vérification des voies.....	30
Modélisation du banc cosmique	31
Simulation Monte Carlo	35
Traitement des données du PCB	42
CONCLUSION	44
BIBLIOGRAPHIE	45
ANNEXES	46

TABLE DES ILLUSTRATIONS

Fig 1 : Répartition des effectifs au LLR.....	6
Fig 2 : Pertes par rayonnement dans les accélérateurs circulaires	11
Fig 3 : Représentation simplifiée de l'accélérateur linéaire	12
Fig 4 : Élément supraconducteur d'une ligne accélératrice	12
Fig 5 : Détecteur conique	13
Fig 6 : Détecteur cylindrique.....	13
Fig 7 : Vue schématique d'un détecteur typique.....	14
Fig 8 : Interaction de différentes particules avec les divers sous-ensembles d'un détecteur ...	14
Fig 9 : Trajectoire d'une particule dans un champ magnétique	14
Fig 10 : En fonction de sa charge	14
Fig 11 : En fonction de sa vitesse.....	14
Fig 12 : Représentation d'un wafer	17
Fig 13 : Une carte PCB avec ses 6 wafers.....	17
Fig 14 : Schématisation de la structure du banc cosmique	18
Fig 15 : Représentation d'un photomultiplicateur avec soin scintillateur.....	19
Fig 17 : Représentation temporelle sans le trigger	20
Fig 18 : Histogramme sans le trigger	20
Fig 19 : Représentation temporelle avec le trigger	21
Fig 20 : Histogramme avec le trigger	21
Fig 21 : Schématisation simplifiée (en regroupant les voies) de la carte trigger	21
Fig 22 : L'alimentation et une de ses cartes	22
Fig 23 : Face avant du DSA	23
Fig 24 : Visualisation d'un signal à l'aide du DSA.....	23
Fig 25 : Résultat du programme d'analyse de diaphonie	25
Fig 26 : Coupe d'un plan de voies.....	27
Fig 27 : Réponse de chacune des voies sur une acquisition de 24 H	30
Fig 28 : Localisation sur deux dimensions.....	31
Fig 29 : Localisation du point de passage dans le plan de détection.....	31
Fig 30 : Vue de l'angle incident part rapport aux plans de localisation	32
Fig 31 : Matrice de pads sur un PCB	33
Fig 32 : Détections au niveau du PCB	34
Fig 33 : Représentation de l'angle solide sur une demi-sphère.....	36
Fig 34 : Angle incident.....	37
Fig 35 : Densité de probabilité	37
Fig 36 : Fonction de répartition.....	37
Fig 37 : Répartition de ϕ	37
Fig 38 : Répartition de θ en $\cos$	37
Fig 39 : Répartition de θ en $\cos^2$	37
Fig 40 : Angle sur le plan de détection (x y).....	38
Fig 41 : Angle d'incidence de la particule	38
Fig 42 : Répartitions des angles incidents avec 8 voies de câblées par plan.....	39
Fig 43 : Répartitions réelles et en simulation	40
Fig 44 : 24 H de fonctionnement avec 8 voies	40
Fig 45 : 24 H de fonctionnement avec 16 voies	40
Fig 46 : Signal du MIP	42
Fig 47 : Cas 1 : 9 pads	42
Fig 48 : Cas 2 : 12 pads	42
Fig 49 : Cas 3 : 16 pads	42

Remerciements

Ces quelques mois passés au LLR m'ont permis de me faire une vision concrète de la recherche au sein d'un laboratoire dépendant du CNRS. Cette expérience m'a permis non seulement de participer à la réalisation et à l'avancement d'un grand projet, mais également de rencontrer des personnes riches en connaissances et compétentes.

Je remercie tout d'abord Henry Videau, directeur du laboratoire pour son accueil, ainsi qu'Arkli Karar, responsable du groupe électronique et Jean-charles Vanel mon responsable de stage pour les conseils apportés et disponibilités qu'ils ont su m'accorder.

De plus je tiens également à remercier l'ensemble des personnes, que ce soit celles avec qui j'ai pu travailler sur ce projet ou celles rencontrées au laboratoire, aussi bien les Chercheurs que les Ingénieurs, Techniciens et Thésards, d'avoir accepté autant que possible de dévoiler les travaux qu'ils exécutaient avec tant de convictions et de passions.

Introduction

La formation « DESS Physique des capteurs et instrumentation » de Brest inclut un stage de fin d'études, d'une durée de 4 mois, que j'ai souhaité effectuer dans le laboratoire « Louis Leprince Ringuet » sur le site de l'Ecole Polytechnique. J'ai donc intégré, avec pour encadrant Jean-Charles Vanel, le service électronique qui est dirigé par Akli Karar.

Les expériences d'astrophysique et de physique des particules mènent à réaliser des détecteurs de grande taille et de grande complexité. Elles se déroulent en trois phases ; la conception, la réalisation et l'exploitation du détecteur. Les deux premières étapes, également réalisées dans le laboratoire, nécessitent des moyens techniques importants dans les domaines mécanique, électronique et informatique.

Mon travail a consisté à mettre en œuvre un banc cosmique. Ce banc devra permettre d'étalonner les détecteurs en silicium avant de les assembler dans un nouveau type de calorimètre électromagnétique. Un calorimètre est un appareil pouvant identifier les particules et les directions suivies par ces dernières, généralement utilisé comme détecteur en physique des particules pour les collisions à très haute énergie.

Dans le cadre de mon projet, j'ai tout d'abord mis en fonctionnement et étalonné l'ensemble des voies. Ce travail associé à la modélisation physique du banc a permis puis par la suite, de procéder à l'interprétation des résultats en vue de leur amélioration.

Par la suite, la mise en œuvre d'une simulation du type « Monté Carlo » des rayons cosmique m'a permis d'appréhender l'influence des paramètres du banc et donc d'en déduire certains impératifs et recommandations d'utilisation.

Dans ce rapport, après une présentation du laboratoire, je vais détailler le projet TESLA avant de passer à mon travail d'analyse, de conception et d'interprétation.

Présentation du Laboratoire

Le Laboratoire Leprince-Ringuet (LLR) est une unité mixte de recherche (UMR 7638) de l'Institut National de Physique Nucléaire et de Physique des Particules (IN₂P₃) du Centre National de la Recherche Scientifique (CNRS) et de l'Ecole Polytechnique. Il est implanté sur le site de l'Ecole Polytechnique à Palaiseau (91). Avant de détailler les missions du laboratoire Leprince Ringuet il est important de voir les structures qui encadrent les travaux du LLR (c'est-à-dire le CNRS et l' IN₂P₃).

I. LE CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE (CNRS)

Le Centre national de la recherche scientifique est un organisme public de recherche fondamentale (Etablissement public à caractère scientifique et technologique, placé sous la tutelle du Ministre chargé de la Recherche). Il produit du savoir et met ce savoir au service de la société.

Avec 26 000 personnes (dont 11 600 chercheurs et 14 400 ingénieurs, techniciens et administratifs (ITA)), un budget qui s'élève à 2 214 millions d'euros HT pour l'année 2004 et une implantation sur l'ensemble du territoire national, le CNRS exerce son activité dans tous les champs de la connaissance, en s'appuyant sur 1260 unités de recherche et de service.

Ses activités

Le CNRS est présent dans toutes les disciplines majeures, regroupées au sein de huit départements scientifiques :

- Physique nucléaire et corpusculaire (PNC),
- Sciences physiques et mathématiques (SPM),
- Sciences et technologies de l'information et de la communication (STIC),
- Sciences pour l'ingénieur (SPI),
- Sciences chimiques (SC),
- Sciences de l'univers (SDU),
- Sciences de la vie (SDV),
- Sciences de l'homme et de la société (SHS),

Il regroupe également deux instituts nationaux :

- L'institut national de physique nucléaire et de physique des particules (IN₂P₃),
- L'institut national des sciences de l'univers (INSU).

Le CNRS développe des collaborations entre spécialistes de différentes disciplines et avec les universités. Cela permet ainsi d'ouvrir de nouveaux champs d'investigations répondant aux besoins de la société. Des actions interdisciplinaires de recherche sont notamment menées dans les domaines suivants :

- « Le Vivant et ses enjeux sociaux »,
- « Information, communication et connaissance »,
- « Environnement, énergie et développement durable »,
- « Nanosciences, nanotechnologies et matériaux »,
- « Astroparticules : des particules à l'Univers ».

Son fonctionnement

Les créations de nouveaux laboratoires du CNRS sont décidées pour quatre ans après évaluation du projet de recherche par le comité national. A la fin de cette période, le Comité national (instance collective chargée d'évaluer les chercheurs) donne son avis sur l'éventuelle prolongation de l'unité de recherche.

C'est donc le comité national, renforcé par ses atouts tels que la représentabilité, la crédibilité et l'indépendance, qui a le rôle central dans la création, la restructuration et la fermeture des laboratoires.

Ce sont ainsi 18 délégations en région qui assurent une gestion directe et locale des laboratoires. Ils entretiennent également les liens avec les partenaires locaux et les collectivités territoriales.

Le dynamisme du CNRS se traduit par de nombreux partenariats :

- 1 256 unités de recherche et de service dont 85 % en partenariat avec l'Enseignement supérieur et les autres organismes de recherche français;
- 4181 brevets actifs, 497 licences et 2 800 contrats industriels en cours avec les entreprises;
- plus de 100 créations d'entreprises à partir de laboratoires CNRS depuis 1999;
- 5 000 stagiaires étrangers accueillis dans les laboratoires,
- 81 accords de coopération avec plus de 50 pays,
- 197 programmes internationaux de coopération scientifique,
- 17 laboratoires européens associés et 4 groupements de recherche européens,
- 13 jumelages de laboratoires et 6 laboratoires mixtes internationaux (Grenoble, Tokyo, Princeton, Singapour, Santiago, Dalian en Chine);
- de nombreuses représentations à Bonn, Bruxelles, Johannesburg, Moscou, Pékin, Santiago du Chili, Tokyo, Tunis, Washington, et une antenne à Hanoi.

II. L'INSTITUT NATIONAL DE PHYSIQUE NUCLEAIRE ET DE PHYSIQUE DES PARTICULES (IN₂P₃)

Créé en 1971, l'Institut National de Physique Nucléaire et de Physique des Particules (IN₂P₃) est un institut du CNRS dont la mission est de promouvoir et de fédérer les activités de recherche dans les domaines de la physique.

Ses missions

La recherche fondamentale

Les expériences de recherche fondamentale effectuées au sein de l'Institut visent à identifier les constituants fondamentaux de la matière et à comprendre les interactions entre ces différents éléments. Ces expériences doivent permettre de déterminer l'origine des masses des particules et de comprendre le comportement de certains constituants élémentaires, allant des particules composites jusqu'aux noyaux.

Les particules et les noyaux jouent un rôle primordial dans l'évolution de l'Univers, aussi bien pour le big-bang que pour les phénomènes violents libérant de très hautes énergies dans le cosmos. La connaissance de l'infiniment petit, acquise au sein de l'Institut, s'avère être très précieuse pour la compréhension de phénomènes astrophysiques et cosmologiques.

Pour réaliser ces expériences, les laboratoires de l'IN₂P₃ développent des détecteurs et des accélérateurs, véritables outils de base. Ce sont d'ailleurs les performances de ces détecteurs et accélérateurs qui permettent de déterminer les progrès effectués dans la discipline. La création de ces outils nécessite la mise au point de techniques de pointe qui peuvent trouver des applications bien au-delà du champ de la physique nucléaire et de la physique des particules.

La mise en œuvre de grands programmes implique un partage des tâches et des échanges constants au sein de collaborations nationales et internationales. Ainsi la concertation et l'interaction sont deux aspects qui permettent de définir l'activité de recherche des laboratoires de l'IN₂P₃.

L'interdisciplinarité

Des liens privilégiés sont noués entre les chercheurs de l'IN₂P₃ et d'autres secteurs de la recherche tels que :

- l'astrophysique (aussi bien en nucléaire qu'en particulaire) et la cosmologie;
- la chimie, dans les programmes de chimie nucléaire et de radiochimie ;
- la physique du solide et celle des matériaux ;
- la biologie par l'utilisation de l'imagerie médicale et des accélérateurs à des fins thérapeutiques.

La communication des résultats

En qualité d'institut de recherche fondamentale, l'IN₂P₃ se doit d'apporter ses connaissances pour la résolution des problèmes de la société. Il est notamment concerné par la question du devenir des déchets radioactifs.

A la pointe de la technologie, il se doit également de faire bénéficier le monde de l'entreprise de son expertise. Pour cela il diffuse vers l'industrie les ressources technologiques qu'il a su développer pour la construction des grands instruments ou le traitement et la transmission des masses de données issues des expériences.

Composé uniquement de laboratoires étroitement associés à l'Université, l'IN₂P₃ accueille régulièrement des étudiants préparant une thèse, et est un lieu de formation de haut niveau à la physique fondamentale et à la haute technologie.

Ses laboratoires

L'IN₂P₃ est un institut regroupant divers laboratoires répartis sur toute la France :

- Centre de calcul de Lyon
- GANIL Caen
- IPN Orsay
- ISN Grenoble
- LAL Orsay
- LPC Clermont-Ferrand
- LPNHE Paris VI-VII
- SUBATECH Nantes
- Musée Curie et archives de l'institut du radium
- Groupe d'Astroparticules de Montpellier
- GDR Phénomènes cosmiques de haute énergie
- APC Paris Rive Gauche
- CPPM Marseille
- CSNSM Orsay
- IPN Lyon
- IReS Strasbourg
- LAPP Annecy
- LPC Caen
- PCC Collège de France
- **LLR Polytechnique**
- LSM Modane
- Virgo
- GDR SUSY
- Programmes

J'ai effectué mon stage dans le laboratoire LLR Polytechnique, dont je vais maintenant vous faire une présentation plus détaillée.

III. LE LABORATOIRE LEPRINCE-RINGUET (LLR)

Le laboratoire Louis Leprince Ringuet est situé dans l'enceinte de l'Ecole Polytechnique à Palaiseau. De 1974 à 2001 il a été connu sous le nom de Laboratoire de Physique Nucléaire des Hautes Energies (LPNHE-X). Ensuite, le physicien Louis Leprince Ringuet, chercheur et professeur à l'Ecole Polytechnique lui céda son nom.

Le programme de recherche du laboratoire porte en premier lieu, sur la physique des particules auprès des accélérateurs du CERN (Suisse), de DESY (Allemagne), de SLAC (Californie, USA) et de PHENIX (Brookhaven, USA). De plus, depuis 1990, un programme d'astrophysique a été ouvert, d'abord au mont Whipple (Arizona, USA), puis sur le site de Thémis dans les Pyrénées. En parallèle, des activités pluridisciplinaires se créent en fonction des synergies associées aux divers programmes.

Ces développements ne sont possibles que grâce à l'efficacité des groupes techniques et administratifs. Ces groupes représentent 60% du personnel du laboratoire. Chaque service met à la disposition de chaque expérience le nombre de personnes nécessaire, en fonction de l'urgence des besoins. Le rattachement principal de chaque ingénieur ou technicien se fait par l'intermédiaire du groupe technique. Cet aspect permet, d'une part d'optimiser les tâches, et d'autre part de développer la technicité de chacun.

Son personnel

Actuellement, le laboratoire est composé de 103 personnes réparties de la manière suivante :

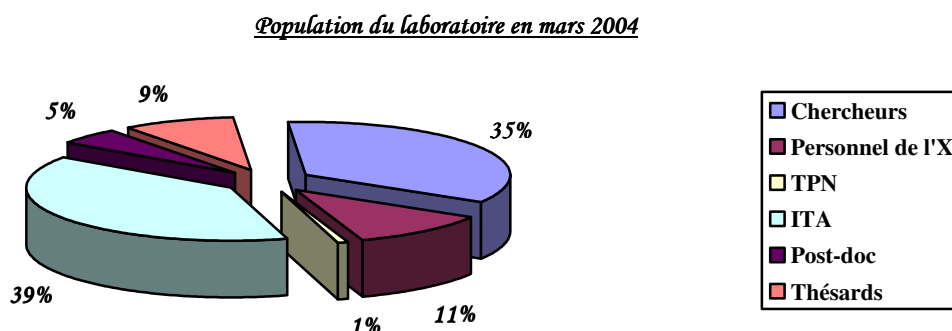


Fig 1 : Répartition des effectifs au LLR

Ainsi ce sont les chercheurs et les ITA (Ingénieurs-Techniciens-Administratifs) qui occupent la grande majorité du laboratoire. L'échange avec l'Ecole Polytechnique permet de disposer d'un personnel supplémentaire et de thésards. En contrepartie certains chercheurs dispensent des cours sur le site de l'école.

Ses activités

Le groupe de physique

Le domaine d'étude du LLR-Polytechnique et les applications qui en découlent appartiennent à la physique des particules et à l'astrophysique. En parallèle, des travaux d'envergure pluridisciplinaires (souvent liés aux compétences des laboratoires voisins sur l'Ecole Polytechnique) sont traités.

A chacun de ces domaines sont associés des chercheurs, des techniciens et des ingénieurs qui s'unissent pour travailler autour d'un même projet. Ainsi, actuellement, que ce soit en physique des particules ou en astrophysique, plusieurs projets sont menés conjointement.

Les projets menés en Physique des particules :

Nom du projet	Descriptif /Objectif du projet
ALEPH	Etude des annihilations entre e^+ et e^- de 90 à 200 GeV
BABAR	Mesure de la Violation de CP dans le système
CMS	Interaction proton-proton à 14 TeV
GEANT 4	Simulation Monte-Carlo pour la physique des particules
H1	Interactions électrons-protons à 27,5 +820 GeV
NA38/NA50	A la recherche du « plasma » de quarks et de gluons
PHENIX	Pour une confirmation de l'existence du plasma de quarks et de gluons
RD36	Calorimétrie Shashlik pour CMS
TESLA	Physique et détecteur sur le futur collisionneur linéaire à électrons

Les projets menés en Astrophysique :

Nom du projet	Descriptif /Objectif du projet
Artémis	Recherche d'antimatière dans les rayons cosmiques de hautes énergies
CAT	Télescope atmosphérique de Cerenkov
CELESTE	Détection de photons cosmiques dans un domaine d'énergie encore inexploré
HESS	Grand télescope de gamme stéréoscopique en Namibie

Les expériences pluridisciplinaires :

Nom du projet	Descriptif /Objectif du projet
ALP	Accélération de Particules par Laser
Métallocènes	Développement de photocathodes solides ou gazeuses
Expérience	Recherche d'effets non liés à l'électrodynamique quantique dans la diffusion photon-photon dans le domaine optique

Le groupe électronique

Le groupe d'électronique comprend 17 ingénieurs et techniciens permanents. Deux spécialistes sont chargés de la conception des cartes et deux électroniciens assurent le câblage des prototypes et des petites séries produites au laboratoire.

La première mission du groupe est de répondre aux demandes des physiciens que ce soit pour la préparation de nouvelles expériences ou la modification d'expériences en cours. Il assure également la conception, la réalisation et la maintenance des systèmes électroniques nécessaires aux expériences.

Parmi les domaines techniques abordés, les plus importants sont :

- l'étude de l'instrumentation au niveau des détecteurs de particules ;
- le traitement analogique rapide et à bas bruit ;
- le contrôle d'équipements instrumentaux et l'acquisition de données par ordinateur ;
- la transmission de données à haute vitesse sur fibres optiques ;
- la définition et la mise en œuvre de bancs de mesures et de tests ;
- les systèmes de déclenchement sélectifs rapides (« Trigger »).

La seconde mission du groupe est de se maintenir au plus haut niveau de compétence possible, face à ce qui se fait dans les autres laboratoires et dans l'industrie. Cette mission difficile est accomplie au contact de collaborateurs et concurrents internationaux, grâce à la prise de responsabilités dans des projets de pointe et grâce à « la veille technologique ».

Pour remplir ses missions, le groupe dispose de puissants moyens de conception électronique, principalement basés sur un réseau de stations de travail SUN, mais aussi sur PC. Les logiciels de conception utilisés dans le service proviennent des sociétés Cadence, Compass, Vantage, Xilinx et Altera.

Le groupe mécanique

Le groupe mécanique du laboratoire comprend 19 ingénieurs et techniciens permanents. Il est structuré en quatre sous-groupes que sont : le groupe projet, le bureau d'études, l'atelier et le service de fonctionnement général.

Pour répondre aux besoins des groupes de physique, le service de mécanique intervient dans les conceptions, calculs, mises au point, commandes et suivis dans l'industrie. Il effectue également les montages sur site et les mises au point des appareillages. Le groupe se répartit par équipes dans les expériences en utilisant les capacités du bureau d'étude et de l'atelier.

Le groupe informatique

Actuellement, tous les secteurs d'activité du laboratoire utilisent quotidiennement des moyens informatiques : les chercheurs et informaticiens pour les développements de logiciels généraux ou liés aux expériences, les groupes d'électronique et de mécanique avec les techniques variées de CAO ainsi que les services administratifs avec les outils de bureautique sur micro-ordinateurs.

Les collisionneurs et Le projet TESLA

Le collisionneur linéaire Tesla devrait permettre d'atteindre une énergie de 4 à 5 fois supérieure à celle du collisionneur LEP (Large Electron-Positron collider) du Cern et ainsi d'étudier avec une grande précision le mécanisme qui donne leurs masses aux particules élémentaires. Selon le scénario attendu, le LHC (Large Hadron Collider), prochain collisionneur proton-proton du Cern, devrait permettre de découvrir le boson de Higgs, responsable de ce mécanisme. Avec Tesla, instrument d'investigation complémentaire au LHC, il devrait être ensuite possible de compléter ces mesures c'est-à-dire d'évaluer avec précision l'ensemble des caractéristiques de cette particule d'importance exceptionnelle et de détecter d'éventuelles déviations révélatrices de phénomènes nouveaux prédits dans certaines théories (dimensions supplémentaires, théories de cordes, super-symétrie).

De manière à pouvoir mieux assimiler les caractéristiques associées au projet TESLA, il paraît indispensable de présenter les moyens techniques utilisés actuellement. Les pages suivantes vont me permettre de présenter les caractéristiques de l'accélérateur ainsi que celles associées aux détecteurs. Je détaillerais également le fonctionnement du calorimètre électromagnétique, autour duquel repose le sujet du stage.

I. GENERALITES

Principe d'un collisionneur

La plupart du temps, les physiciens veulent étudier des particules massives et instables, qui ont seulement une existence éphémère. Or l'environnement dans lequel nous nous situons est constitué de particules de masse très faibles. De manière à résoudre ce problème, l'utilisation de l'égalité d'Albert Einstein ($E=mc^2$, où E est l'énergie, m la masse et c la vitesse de la lumière), permet de considérer la masse comme une forme d'énergie.

Ainsi, pour produire des particules de forte masse, il suffira de donner beaucoup d'énergie cinétique à des particules de faible masse, grâce à un accélérateur, et de faire rentrer les particules en collision. Durant cette collision, l'énergie cinétique des particules est convertie en énergie de masse pour former des particules massives. C'est par ce procédé que la création des particules instables et massives ainsi que l'étude de leurs propriétés peuvent être réalisées.

Ordres de grandeurs

Plus la longueur d'onde utilisée pour sonder est courte, plus le nombre d'informations est élevé. Une particule permet de sonder des distances de l'ordre de grandeur de sa longueur d'onde. Pour sonder des échelles plus petites, la longueur d'onde de l'échantillon doit être également plus petite.

En physique des hautes énergies le théorème suivant est appliqué : « L'impulsion (ou quantité de mouvement d'une particule est inversement proportionnelle à sa longueur d'onde ». Les accélérateurs de particules permettent donc, non seulement d'augmenter la vitesse des particules, mais également de diminuer la longueur d'onde de ces particules.

C'est pour cette raison que l'on choisira un appareil de visualisation particulier (téléscope, microscope électronique ou accélérateur) en fonction de la dimension de l'objet à visualiser (galaxie, cellule, atome ou noyau).

Grâce aux vitesses atteintes dans l'accélérateur linéaire de TESLA, les qualités de visualisation seront exceptionnelles et permettront d'observer des détails plus fins encore que ceux qui ont été visualisés jusqu'à présent.

II. CHOIX DU TYPE D'ACCELERATEUR

Selon ses caractéristiques

Il y a différentes façons de concevoir ces accélérateurs, chacune avec ses avantages et ses inconvénients. Voici une liste succincte des choix de conception des principaux accélérateurs.

Le type de collision produit

- **avec une cible fixe** : une particule est projetée sur une cible immobile. Dans les expériences sur cible fixe, une particule chargée, comme un électron ou un proton, passe dans un accélérateur et vient frapper une cible qui peut être un solide, un liquide ou un gaz. Un détecteur détermine les propriétés telles que la charge, la masse ou l'impulsion des particules résultantes.
- **avec collision de deux faisceaux envoyés l'un contre l'autre** : Au cours de ces expériences, deux faisceaux de particules de haute énergie se croisent et entrent en collision en certains points déterminés. L'avantage de cette méthode réside dans le fait de disposer de deux faisceaux d'énergie cinétique (énergie associée à la vitesse) non nulle, l'énergie disponible est plus élevée que lors d'une expérience utilisant un faisceau envoyé sur une cible fixe. De plus, la probabilité de formation de particules de grande masse est elle aussi plus importante. Les particules utilisées ayant une grande impulsion, elles ont alors des longueurs d'onde courtes et font d'excellentes sondes.

Sa forme

Tous les accélérateurs sont soit linéaires, soit circulaires, la différence réside dans le fait que, soit la particule est éjectée hors de l'accélérateur (accélérateur linéaire), soit elle décrit une trajectoire circulaire en recevant de l'énergie à chaque tour (accélérateur circulaire).

Que l'accélérateur soit linéaire ou circulaire, l'utilisation d'un champ électrique s'impose pour accélérer les particules. De plus, dans un accélérateur circulaire, c'est l'introduction d'un champ magnétique (grâce à des aimants supraconducteurs), qui va permettre aux particules de garder une trajectoire circulaire.

Avantages/inconvénients des accélérateurs linéaires et circulaires

L'avantage d'un accélérateur circulaire sur un accélérateur linéaire réside dans le fait que les particules prennent à chaque tour un peu plus d'énergie. On obtient donc des particules très énergétiques sans avoir une longueur d'appareil démentielle. De plus, la probabilité de collision entre les faisceaux est augmentée si l'on considère le nombre de tours effectués.

D'un autre côté, les accélérateurs linéaires sont beaucoup plus simples à construire car ils ne requièrent pas de gros aimants pour maintenir la trajectoire circulaire. Les accélérateurs linéaires demandent des cavités supraconductrices de grand diamètre pour donner suffisamment d'énergie aux particules, et sont également plus coûteux.

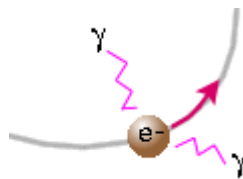


Fig 2 : Pertes par rayonnement dans les accélérateurs circulaires

Un autre problème que les physiciens doivent considérer est qu'une particule chargée et accélérée émet de l'énergie sous forme de rayonnement. A grande énergie, la perte par rayonnement est beaucoup plus importante pour une accélération circulaire que pour une accélération linéaire. De plus, la perte par rayonnement est beaucoup plus importante pour les électrons accélérés que pour les protons qui sont lourds. Les électrons et les anti-électrons (positons) ne peuvent être amenés à de hautes énergies que dans les accélérateurs linéaires, ou les accélérateurs circulaires de grand diamètre.

A titre d'exemple, le grand accélérateur en construction au CERN, le LHC ('Large Hardon Collider') est chargé d'accélérer des protons jusqu'à 14 TeV (Téra-électronVolt soit 10^{12} eV). Il est circulaire et a une circonférence de 27 Km. Il dissipe cependant une puissance de 3.6 kW. Or, comme les protons sont des particules lourdes, il est possible d'utiliser un accélérateur circulaire. Pour le projet TESLA, les particules accélérées étant légères (électrons et positons), ce choix ne peut être envisagé.

Cas du projet TESLA

Le projet TESLA (inspiré du nom du physicien Yougoslave) signifie TeV-Energy Superconducting Linear Accelerator, c'est-à-dire « accélérateur linéaire supraconducteur atteignant une énergie de l'ordre du Terra-eV (10^{12} eV) ». C'est donc dans l'optique d'atteindre une très haute énergie que le choix de l'accélérateur linéaire a été fait. Les pertes engendrées par les accélérateurs circulaires ne permettent pas d'atteindre une telle valeur.

Ainsi, TESLA est un projet d'accélérateur linéaire à électron, fonctionnant dans la gamme d'énergie allant de 91 GeV jusqu'à une énergie d'environ 1000 GeV. Dans ce projet, les cavités accélératrices sont supraconductrices.

D'autre part, et ce toujours afin d'atteindre une énergie qui soit la plus élevée possible, la collision se fait entre deux faisceaux de particules envoyés l'un contre l'autre.

Description de l'accélérateur

Les deux faisceaux entrant en collision sont constitués pour l'un d'électrons et pour l'autre de positrons. Chacun des faisceaux est accéléré par paquet de 10^{10} électrons (e^-) ou de positons (e^+) d'environ quelques nanomètres d'épaisseur dans un accélérateur linéaire de 17 Km. La taille du paquet d'électrons ou de positons dépend de l'anneau de création à l'amont de chacun des deux accélérateurs.

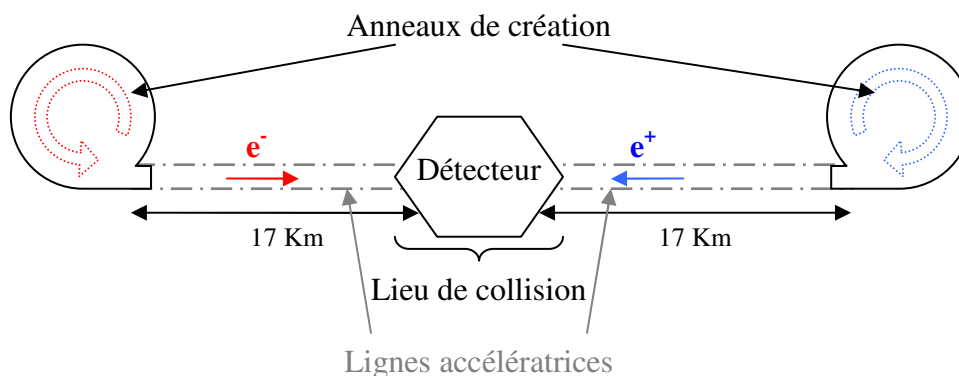


Fig 3 : Représentation simplifiée de l'accélérateur linéaire



Fig 4 : Élément supraconducteur d'une ligne accélératrice

A l'heure actuelle le site d'implantation du projet TESLA n'a pas encore été choisi ; de nombreux pays espèrent encore voir le projet s'implanter sur leur territoire d'ici 2015, date d'implantation prévue.

III. LES DETECTEURS ASSOCIES

Lorsqu'un accélérateur a cédé de l'énergie aux particules, elles peuvent alors entrer en collision entres-elles. Chacune de ces collisions est appelée « évènement ». Le but est d'isoler chaque évènement, d'en recueillir les caractéristiques et de vérifier si la transformation des particules concorde avec la théorie testée. Chaque évènement est très complexe car un grand nombre de particules sont produites simultanément. La plupart de ces particules ont des durées de vie si courtes qu'elles ne peuvent parcourir que de très faibles distances avant de se désintégrer en d'autres particules et ne sont donc pas détectables.

Les physiciens des particules étudient les particules issues de la désintégration de particules hypothétiques et en déduisent ainsi l'existence de ces dernières. Pour chercher ces diverses particules et les produits de leur désintégration, des détecteurs complexes constitués de sous-détecteurs ont été conçus. Chacun de ces sous-détecteurs permet distinguer les particules ou de réaliser des mesures d'énergies, d'impulsions, de trajectoire de ces dernières. Une fois tous ces éléments regroupés dans un détecteur, une analyse détaillée des données permettra d'isoler une particule individuelle de la multitude de particules impliquées dans une collision.

Pour chaque évènement, une vaste quantité de données issues des détecteurs sont enregistrées en vue d'être interprétées par des programmes d'analyse informatique.

Géométrie des détecteurs

Une attention aussi importante peut être accordée aux évènements ayant lieu avant la collision qu'à ceux qui se déroulent à l'instant suivant la collision. Pour mesurer ces évènements des détecteurs sont placés dans la région qui sera « arrosée » par des particules à la suite de la collision. Les détecteurs sont construits de différentes façons selon le type de collisions à étudier.

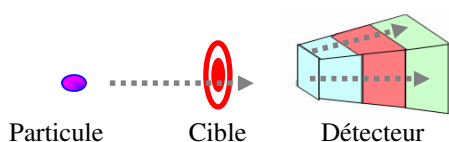


Fig 5 : Détecteur conique

- **Cible fixe** : les particules sont produites face à la cible et les détecteurs en forme de cônes sont placés derrière cette cible.

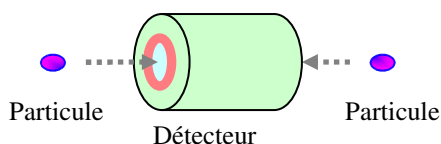


Fig 6 : Détecteur cylindrique

- **Collision de faisceaux** : les particules sont émises dans toutes les directions, le détecteur sera donc de forme sphérique, ou plus généralement cylindrique.

Le projet TESLA étudie les particules émises après collision de faisceaux, c'est pourquoi le détecteur utilisé sera cylindrique.

Description d'un détecteur cylindrique

Les détecteurs modernes sont formés de différentes parties qui testent les différents aspects d'un événement (collision). Ces événements sont prédéterminés de façon à ce que le maximum d'informations utiles puisse être traité.

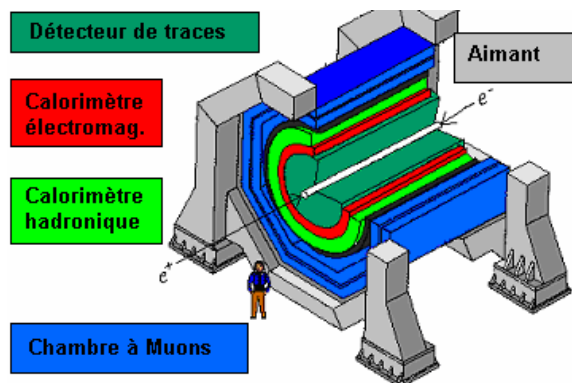


Fig 7 : Vue schématique d'un détecteur typique

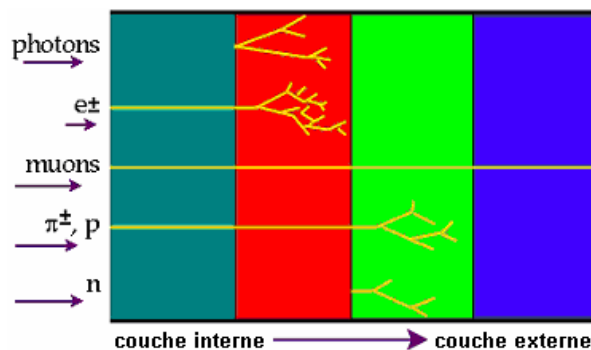


Fig 8 : Interaction de différentes particules avec les divers sous-ensembles d'un détecteur

Les détecteurs sont constitués de nombreux sous-ensembles qui testent chacun une propriété spécifique des particules. Ces sous-ensembles sont empilés de façon à ce que toute particule traverse successivement les couches internes pour se diriger vers les couches externes. Une particule n'est pas observable jusqu'à ce qu'elle interagisse avec le détecteur de façon mesurable, ou se désintègre en particules elles-mêmes détectables.

Caractérisation des particules traversant le détecteur

Une fonction importante d'un détecteur est la mesure de la charge et de l'impulsion de la particule. C'est pour cette raison que les parties internes du détecteur sont placées dans un champ magnétique intense (en particulier pour le détecteur de traces). On peut facilement identifier les particules chargées par leur trace, celles des particules positives ou négatives s'infléchissant en sens opposé dans un même champ magnétique.

Fig 9 : Trajectoire d'une particule dans un champ magnétique

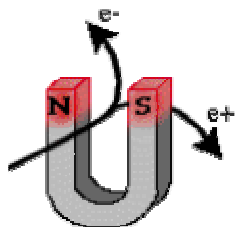


Fig 10 : En fonction de sa charge

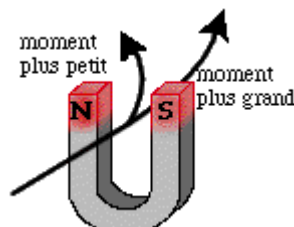


Fig 11 : En fonction de sa vitesse

Les particules ayant une vitesse importante ont une trajectoire moins courbée dans un champ magnétique que celles ayant une plus petite impulsion. Cela est dû au fait qu'une particule de grande impulsion passe moins de temps dans le champ magnétique (ou qu'elle ait une plus grande inertie) qu'une particule de faible impulsion. On peut ainsi calculer l'impulsion de chaque particule.

Plus de précisions sur le calorimètre électromagnétique

Dans le cas du projet TESLA, le futur calorimètre électromagnétique sera constitué d'un empilement de couches de tungstène et de plaques de détecteur en silicium. Les photons, électrons et positons vont interagir avec les noyaux de tungstène au passage à travers ces plaques. Il y aura alors formation d'une gerbe électromagnétique. Un signal électronique provenant de l'épaisseur en silicium va alors permettre de détecter le passage d'une particule. Une pixellisation du détecteur en silicium permettra également de disposer d'une localisation de ce passage à travers le plan formé par le détecteur en silicium. Une superposition de couches au niveau de détecteurs en silicium permettra d'obtenir une représentation tridimensionnelle associée au passage d'une particule.

Il est à noter qu'un facteur très important à ce niveau est la minimisation de l'épaisseur de chacune des couches de silicium. C'est pour cette raison qu'il est utilisé du tungstène, élément de très forte densité, pour les couches de séparation.

IV. EN RESUME

En résumé, un accélérateur prend des particules, les accélère en utilisant des champs électriques et magnétiques, et les fait entrer en collision soit avec une cible fixe soit avec d'autres particules.

Les détecteurs entourant le point de collision reçoivent des millions d'informations associées à un événement. L'ensemble des informations relatives à un événement est mémorisé.

L'enregistrement et le traitement des données s'effectuent grâce à des ordinateurs. Ce traitement des données permet de représenter les trajectoires probables des particules, leur désintégration, et les anomalies par rapport au comportement attendu.

C'est ainsi, par l'analyse de tels événements que l'on a pu prouver la validité du modèle standard. Un accélérateur permet donc d'espionner la structure des particules. L'ordinateur tient un grand rôle dans la prise de mesures, mais c'est à l'homme d'interpréter les données alors obtenues.

Les résultats expérimentaux ont donné de nombreux arguments en faveur du modèle standard, mais il faut reconnaître qu'il reste encore beaucoup de chose à découvrir et à apprendre. Les physiciens sont en particulier ennuyés par les neutrinos, la matière noire, et l'intégration de la gravité dans le modèle standard. A l'avenir, il faudra concevoir de nouvelles expériences pour tester les théories élaborées pour expliquer ces phénomènes. Un tel projet aura donc pour but de contribuer à la découverte de nouvelles connaissances.

Les lignes générales du projet TESLA étant présentées, passons maintenant à la présentation du sujet de stage et des travaux que j'ai pu réaliser au sein du laboratoire.

Analyses et réalisations

I. LE SUJET DE STAGE

Contribution à la mise en œuvre d'un hodoscope de rayons cosmiques à base de scintillateurs et de photomultiplicateurs pour la calibration d'un détecteur de particules ionisantes en silicium.

L'origine de ce stage est la mise en œuvre d'un appareil de mesure permettant de tester et d'étalonner un prototype détecteur en silicium associé au calorimètre électromagnétique.

Dans un premier temps, il est indispensable d'assimiler le fonctionnement d'un calorimètre électromagnétique, l'appareil va donner la trajectoire et l'énergie des particules qui le traversent.

Lorsqu'une particule traverse le calorimètre électromagnétique, elle crée des paires électron-trou par excitation (ionisation) du matériau (silicium). Lorsque qu'une différence de potentiel est appliquée aux bornes de chaque pixel (cellule du détecteur), cette information devient mesurable. Le phénomène se reproduit dans chaque pixel traversé par la particule et permet ainsi de reconstituer la trajectoire de la particule en fonction des pixels touchés.

Le second point à prendre en compte est que pour effectuer une mesure correcte des données relatives aux passages de particules il s'avère primordial d'étalonner chacune des cellules, c'est-à-dire connaître l'énergie déposée dans un pixel pour une amplitude mesurée en sortie de l'électronique. C'est pour réaliser cette tâche que l'hodoscope (instrument identifiant le passage d'une particule chargée) à voies de photomultiplicateurs et scintillateurs s'avère utile. Dans la mesure où il n'y a pas d'interaction entre la charge de la particule et un champ magnétique intense, la trajectoire d'une particule est une droite. Ainsi pour déterminer cette trajectoire dans l'espace, il suffira de disposer de deux localisations spatiales de cette même particule (localisations effectuées grâce aux voies de détections).

Pour étalonner le calorimètre électromagnétique, il est indispensable de disposer d'une référence d'énergie. Plutôt que d'utiliser une source radioactive dont la dégradation va générer plusieurs types de rayonnements de trajectoires difficiles à identifier, il a été choisi d'utiliser des particules cosmiques. Ces particules ou MIP (Minimum Ionizing Particles) venant de l'espace ont la propriété de toujours déposer en moyenne une quantité d'énergie d'un même ordre de grandeur, en traversant une même épaisseur de matière. Cette source est considérée comme constante et va ainsi permettre de réaliser une calibration des détecteurs.

Une présentation du sujet de stage va impliquer celle du détecteur en silicium ainsi que celle du banc de test tout en détaillant l'ensemble des éléments le constituant.

Le calorimètre électromagnétique (Ecal pour Electromagnetic calorimeter)

Le calorimètre recherché doit avoir une fine granularité avec une grande segmentation de manière à avoir la meilleure résolution possible en d'énergie et en direction. Placé près du faisceau de particules, l'Ecal devra donc être très compact. Un tel calorimètre a une structure comparable à un sandwich de Tungstène-Silicium. Pour cette raison, ce dispositif s'appelle un « calorimètre traqueur ». Il jouera un rôle essentiel dans la mise en évidence du boson de Higgs par sa désintégration en deux photons; ce qui nécessite d'excellentes résolutions angulaire et en énergie. Cette résolution angulaire sera obtenue par une superposition de détecteurs silicium (30 couches) divisées en pixels de 1cm^2 .

Le but du stage est de pouvoir étalonner chaque détecteur en silicium du prototype de calorimètre électromagnétique, grâce à l'utilisation d'une source naturelle constituée par les rayons cosmiques (Cf. annexe).

Description d'un détecteur silicium

C'est la pixellisation qui va permettre de définir la taille de la surface de détection. Ici le pixel élémentaire sera constitué par un pad de 1 cm^2 (1 cm sur 1 cm). Cette surface élémentaire correspond aux exigences de résolutions, considérées pour le projet TESLA.

Un wafer est en réalité une matrice 36 (6*6) pads. Un espace d' 1mm de protection électronique est attribué au pourtour de chaque wafer, ce qui mène à une taille totale de 6.2 cm * 6.2 cm. Son épaisseur, optimisée pour l'obtention du signal de détection est de $525\text{ }\mu\text{m}$.

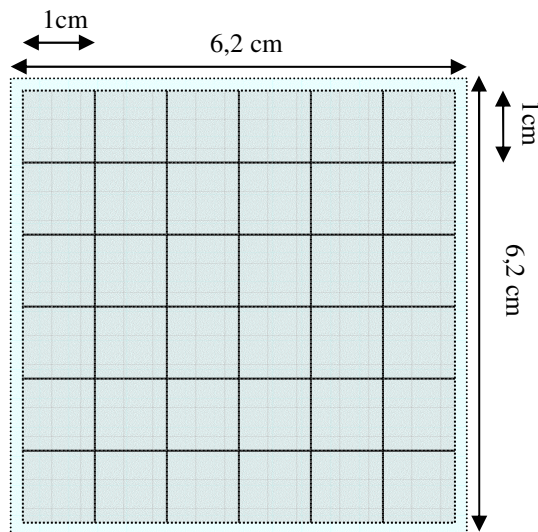


Fig 12 : Représentation d'un wafer

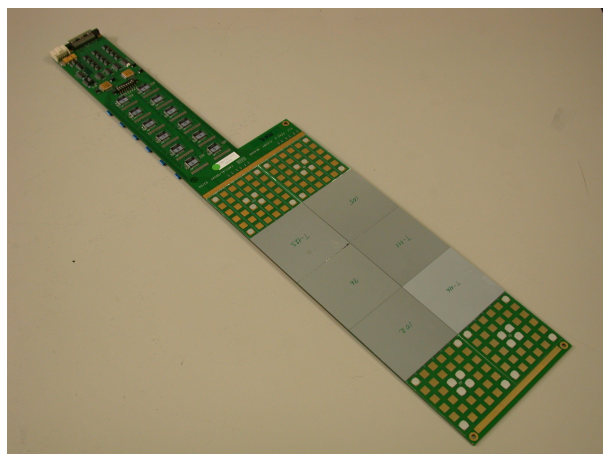


Fig 13 : Une carte PCB avec ses 6 wafers

La liaison électronique, qui va permettre de réaliser le transfert d'information entre le détecteur silicium et le circuit imprimé, est établie grâce à un procédé de collage avec une colle conductrice.

Une carte électronique (PCB pour Printed Circuit Board) va permettre de regrouper 6 wafers (3*2) côte à côte et de relever les signaux issus de chacun des pads à l'aide d'une électronique dite de « Front End ».

Présentation du banc de test

Le but du banc de test est de pouvoir réaliser l'étalonnage de détecteurs en silicium à partir de l'utilisation des rayons cosmiques, dont la localisation et l'angle d'impact avec le détecteur sont identifiés.

Pour cela le projet repose sur de nombreuses connaissances physiques, il nécessite une structure mécanique bien définie et fonctionne grâce à des cartes électroniques, complémentées par un traitement de données informatique.

Sa structure mécanique

Construit au sein du groupe de mécanique, le banc repose avant tout sur une structure rigide en aluminium et en bois, dont voici les caractéristiques physiques :

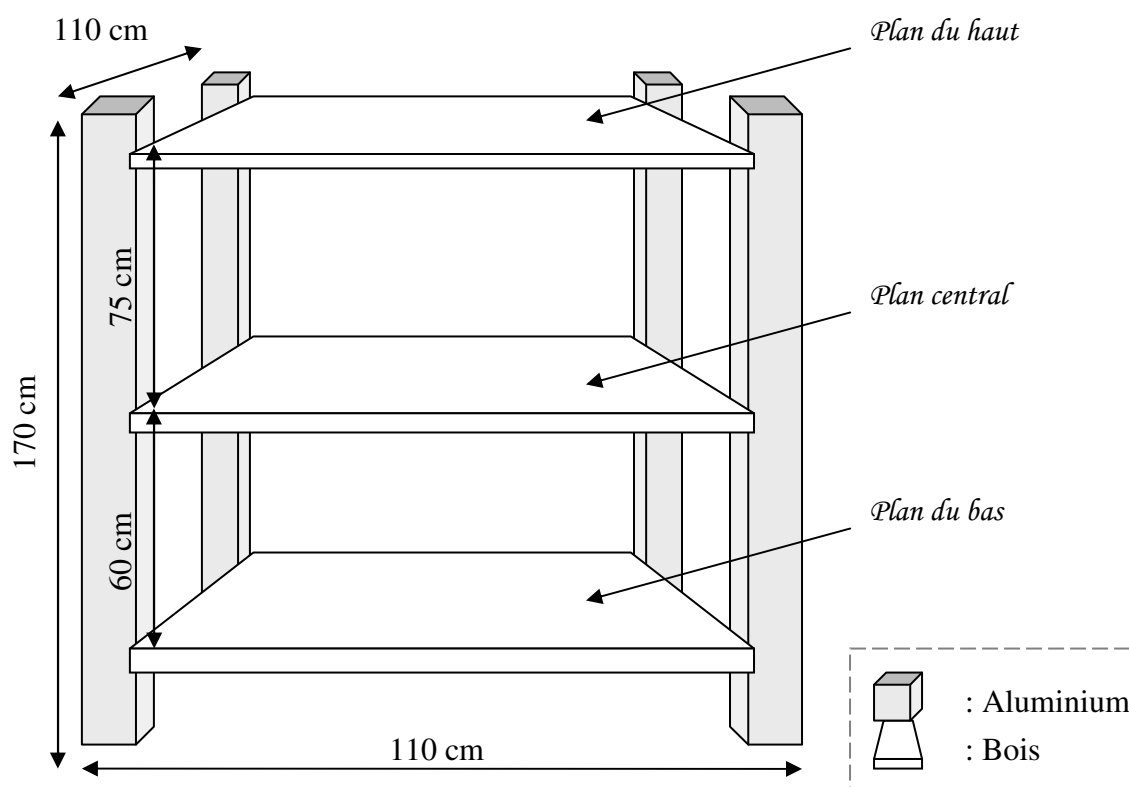


Fig 14 : Schématisation de la structure du banc cosmique

La structure de l'hodoscope à elle seule est relativement volumineuse (120 cm * 120 cm * 170 cm), mais pour le fonctionnement final de l'ensemble, de nombreux appareils (alimentations des photomultiplicateurs et cartes électroniques, ainsi que le matériel d'acquisition) lui sont également associés.

Les plans de photomultiplicateurs

L'hodoscope va fonctionner grâce à quatre plans, chacun constitué de 16 photomultiplicateurs (PM) associés chacun de manière individuelle un barreau de scintillateur couvrant une largeur de 2,75cm. L'utilisation de ces 16 voies placées côte à côte permettra de surveiller une grande surface avec une résolution spatiale égale à la largeur d'un scintillateur. Le fonctionnement du photomultiplicateur est détaillé plus précisément en annexe.

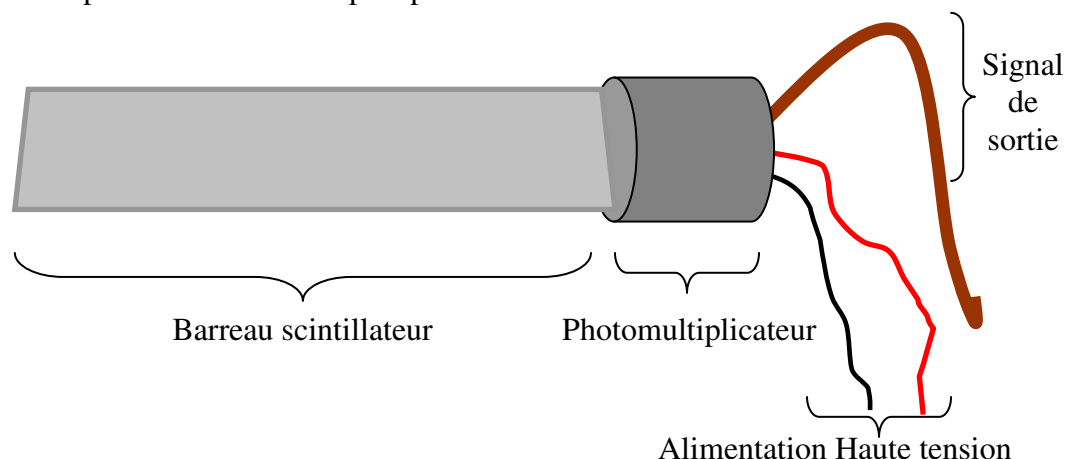


Fig 15 : Représentation d'un photomultiplicateur avec son scintillateur

Lors du passage d'une particule à travers le barreau du scintillateur, il y aura génération de photons qui vont se propager jusqu'au photomultiplicateur. Cela se traduira par la création d'un signal sur la voie de sortie du photomultiplicateur considéré.

Le couplage de plans

Deux plans seront dédiés à la localisation respectivement en X et Y d'une particule. Ainsi deux couples de plans permettront de localiser le passage d'une particule à deux hauteurs différentes (plan haut et bas de l'hodoscope).

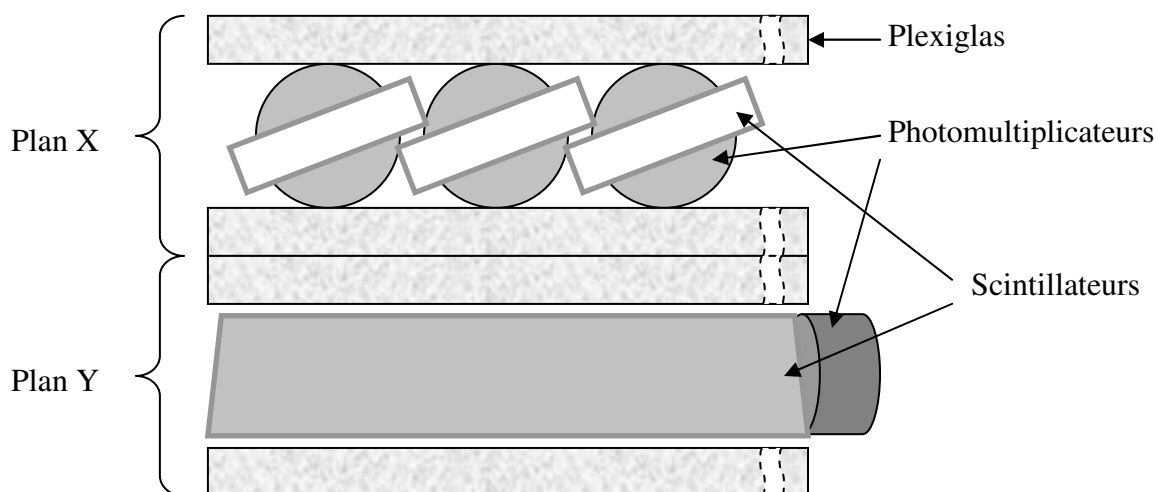


Fig 16 : Coupe d'un couple de plans

La carte de trigger

L'utilisation de la carte trigger a pour but de détecter le passage d'une particule à travers les quatre plans et permet ainsi mémoriser les signaux obtenus sur les PM à cet instant donné. Le but de la carte est de générer un évènement (signal de trigger), lorsque chacun des plans a au minimum une de ses voies touché. Cette carte électronique, développée par un précédent stagiaire, a ainsi pu être testée en fonctionnement réel (chacune des 64 voies reliée à un PM).

Utilité d'un tel système

Les signaux issus du détecteur sont relativement faibles face à ceux du bruit. Si l'on souhaite visualiser le signal utile il faut se concentrer sur un espace de temps réduit, mais cela n'est pas simple lorsque le signal a une très faible fréquence d'apparition. De plus l'aspect aléatoire de ces deux types de signaux rend difficile la détection du signal recherché. C'est pour cela qu'il faut chercher continuellement le signal utile dans les données brutes. Le trigger va effectuer cette action et générer un signal de déclenchement lorsqu'un signal est reconnu comme étant celui attendu.

Ce signal de déclenchement permettra par la suite de limiter considérablement le nombre de mesures à mémoriser, tout en augmentant de manière inversement proportionnelle la probabilité d'obtention du signal attendu (ici un signal reflétant le passage d'une particule).

Ce principe, souvent utilisé dans le domaine de la physique nucléaire et de l'astrophysique peut être présenté de la manière suivante :

Fonctionnement en continu (sans le « trigger »)

Dans notre cas, il est pratiquement impossible de discerner le signal, très faible face au bruit. Cela est dû au fait que le signal que l'on souhaite traiter a une fréquence d'apparition très faible et représente donc une infime partie du temps d'acquisition.

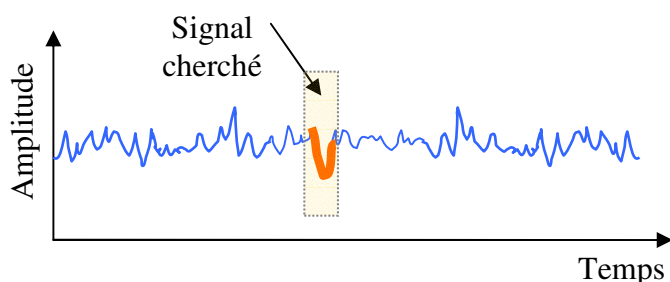


Fig 17 : Représentation temporelle sans le trigger

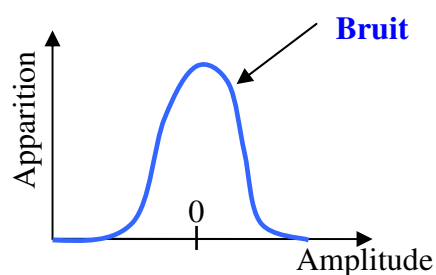


Fig 18 : Histogramme sans le trigger

La problématique est de pouvoir faire accroître l'importance du signal de façon à ce qu'il ne soit plus « noyé » par le bruit. Pour cela il faut mettre en œuvre un système capable de concentrer les mesures sur des petites portions de temps où la probabilité d'obtenir un évènement est pratiquement certaine. D'où l'utilité de la carte de trigger.

Fonctionnement par intermittence (avec le « trigger »)

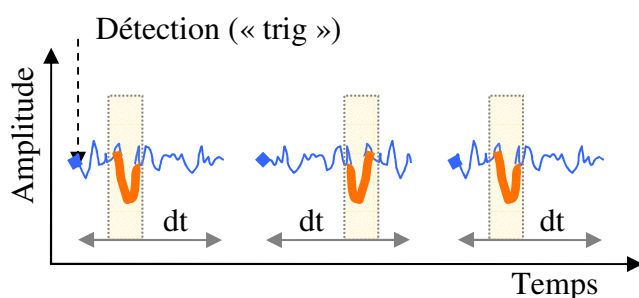


Fig 19 : Représentation temporelle avec le trigger

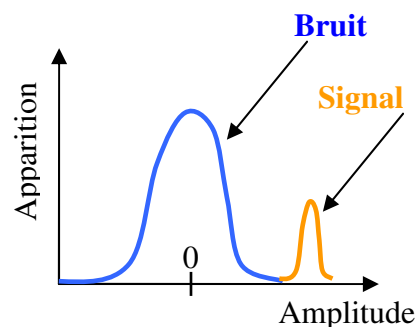


Fig 20 : Histogramme avec le trigger

Le signal prend de l'importance face au bruit et il devient possible d'identifier ses caractéristiques. Ainsi l'on atteint le but visé qui est d'augmenter le rapport *signal/bruit* en terme d'apparition.

Principe de la carte

Les signaux issus des plans de PM arrivent sur la carte « trigger » et lorsqu'une détection a lieu simultanément sur au moins une voie de chacun des quatre plans, un signal « trigger » est généré. Ce signal ainsi que l'ensemble des signaux numérisés (provenant des 64 voies) sont transmis à un PC via une liaison USB.

En complément, un signal de trigger rapide (norme NIM) est également envoyé au système testé.

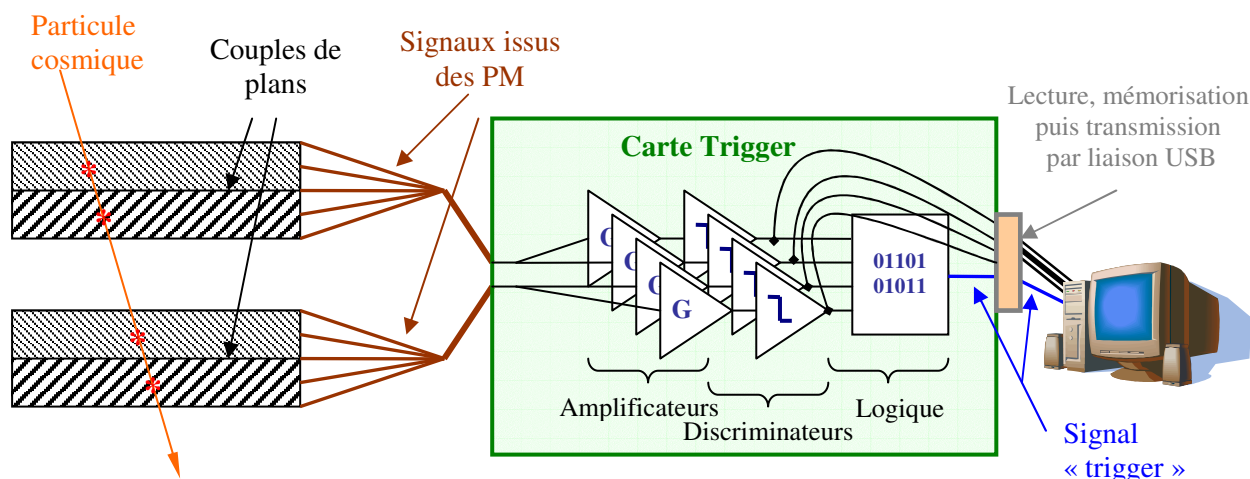


Fig 21 : Schématisation simplifiée (en regroupant les voies) de la carte trigger

Fonctionnement attendu

Dans la réalité, les signaux issus des PM ne nécessitent pas d'être amplifiés par contre le réglage du seuil pour le discriminateur (comparateur) sera déterminant quant au nombre de détections.

Le temps de réponse du trigger doit être inférieur à 180 ns afin de permettre le déclenchement du dispositif testé.

II. REALISATIONS ELECTRIQUES ET MECANIQUES (HARD)

Les photomultiplicateurs et scintillateurs

Les photomultiplicateurs sont tous identiquement constitués, mais certains de leurs paramètres varient d'un PM à l'autre. Ces variations de caractéristique vont entraîner une nécessité de correction pour l'uniformisation de leur réponse lorsqu'un MIP traversera un scintillateur.

Afin de pouvoir utiliser les plans de l'hodoscope j'ai dû câbler chaque photomultiplicateur et m'assurer du bon fonctionnement de chacune des voies. Cette étape a été réalisée par l'analyse des signaux de réponse au passage aléatoire d'une particule. Par la suite, la vérification de la qualité du barreau de scintillateur a pu être établie grâce à une source radioactive de Strontium, placée au plus près possible du scintillateur. Enfin la dernière étape qui est l'étalonnage de l'ensemble des voies, a été établie suivant plusieurs méthodes.

Matériel utilisé

Alimentation haute tension Caen (SY 2527 Universal Multichannel Power Supply System)

Chaque photomultiplicateur nécessite une alimentation en tension continue négative élevée (environ 1400 V ici). De manière à pouvoir relever des signaux de sortie sur un ensemble de 64 voies de photomultiplicateurs, chacune d'entre elles doit être alimentée indépendamment si l'on souhaite pouvoir les étalonner correctement.

Nous avons utilisé une alimentation très haute tension de chez Caen pour réaliser cette opération d'alimentation des différentes voies.

Les caractéristiques de cette alimentation couplée avec deux cartes internes (A193 32AN) :



Fig 22 : L'alimentation et une de ses cartes

- Possibilité de relier un clavier et un écran,
- Alimentation de tension continue,
- Activation des voies par groupement de 8,
- Chaque tension est ajustable individuellement et peut prendre 2 valeurs suivant une condition d'entrée,
- Contrôle de tension sur chaque voie,
- Contrôle d'intensité sur chaque carte,
- Réglage des croissances et décroissances de tensions et d'intensités,
- Chacune des cartes dispose de 48 voies.

Toutes ces fonctionnalités se sont avérées utiles pour l'analyse et le paramétrage des tensions d'alimentations.

Digital Signal Analyser (DSA 602 A)



Fig 23 : Face avant du DSA

Sur cette visualisation à l'écran de l'oscilloscope, apparaît ici le signal créé par un photomultiplicateur lors du passage d'un MIP à travers le barreau de scintillateur. La forme caractéristique du pulse de l'ordre de quelques nanosecondes et d'une amplitude voisine de 600 mV, apparaît clairement.

Par la suite se sont les fonctions de calcul de moyennes et d'écart type qui se sont avérées utiles pour l'étalonnage des différentes voies de photomultiplicateurs, couplés aux scintillateurs.

Le DSA est un oscilloscope numérique avec les caractéristiques suivantes :

- bande passante de 1 GHz,
- écran tactile,
- rack pouvant recevoir 3 cartes.

Il dispose de deux cartes d'acquisition :

- 11A72 : 2 voies 1GHz avec entrée sur 50 Ω ,
- 11A32 : 2 voies 400MHz avec entrée sur 50 Ω ou 1M Ω .

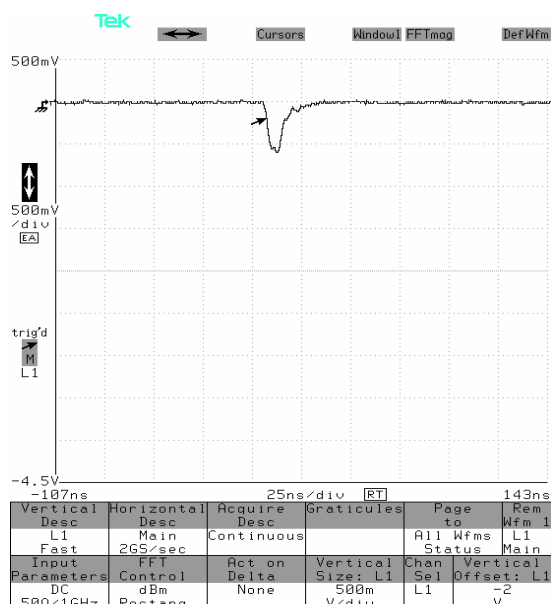


Fig 24 : Visualisation d'un signal à l'aide du DSA

La source radioactive : Sr90

Le strontium (Sr) se trouve à l'état naturel en tant qu'élément non radioactif. Le strontium⁹⁰ est l'isotope radioactif le plus important dans l'environnement. Il émet une particule bêta sans rayonnement gamma car il se dégrade en Yttrium⁹⁰ (aussi radioactif). Sr⁹⁰ a une demi-vie de 29 ans, c'est-à-dire qu'après 29 ans son activité sera divisée par deux. Sr⁹⁰ est un sous-produit de la fission de l'uranium et du plutonium dans des réacteurs, armes nucléaires.

Cette source radioactive va permettre d'étalonner les différentes voies de l'hodoscope. Son support est un coffre de plomb pourvu d'une fente qui permet la création d'un rayonnement directionnel.

Les propriétés de ce rayonnement dans nos scintillateurs sont proches de celles des rayons cosmiques. De plus, ce signal est rapidement atténué, même dans l'air et ne se propage donc pas sur une grande distance.

Cette source sera donc rapprochée au maximum de la voie de détection lors de l'étalonnage.

Vérification du fonctionnement des PM

La vérification du bon fonctionnement de chacune des voies (Photomultiplicateur + scintillateur) a été réalisée sur le DSA, en injectant une tension voisine de 1200 V grâce à l'alimentation « Caen ». Cette première approche a permis de visualiser la réponse de chacune des voies au passage de particules cosmiques. La fonction trigger de l'oscilloscope s'est alors avérée utile en ce qui concerne la mesure des amplitudes sur les signaux.

Les mesures ont révélé que les caractéristiques varient pour chacun des PM. Certaines contraintes environnementales tels que la distance du PM par rapport à un plan de masse peuvent influencer sur les résultats. Un autre aspect est que pour obtenir une stabilité de fonctionnement, les PM doivent être alimentés quelques dizaines de minutes avant de faire les mesures.

Les scintillateurs

De la même manière, la qualité des scintillateurs (de part leur isolation photonique, la qualité de leur découpe) et surtout du collage entre le scintillateur et le photomultiplicateur entraînent une grande diversité dans les résultats. Ainsi certains scintillateurs moins bien isolés se révèlent être sensibles à la lumière du jour. Il a également été possible de visualiser une disparité sur les propagations dans le barreau de scintillateur, grâce à la source radioactive.

Etalonnage des voies

De manière à ce que chacune des voies réponde similairement à une même détection, il a été indispensable d'étalonner chacune des voies. Cette uniformisation s'est réalisée en plusieurs étapes d'ajustement des tensions d'alimentation de chacun des 64 photomultiplicateurs.

Sans la source (rayons cosmiques + bruit)

En utilisant un seuil de détection voisin de 150 mV (celui de la carte de détection « trigger »), nous avons grossièrement étalonné les différentes voies de façon à ce que l'on ait un nombre de détections voisin de 320 par minute car :

$$surface[m^2] * temps[s] * angleS[sr] * 70[m^{-2}.s^{-1}.sr^{-1}] = 319 \text{ particules / min}$$

La valeur de $70 m^{-2}.s^{-1}.sr^{-1}$ est donnée pour le passage de particules cosmiques (cf. annexe).

avec $surface = 2.75 cm * 44 cm$, $temps = 60 s$ et $angleS = 2\pi sr$

Avec la source (radiations + rayons cosmiques + bruit)

Par la suite, une optimisation plus fine de la tension d'alimentation a été possible grâce à l'utilisation des fonctions d'écart type et de moyenne du signal de sortie sur le DSA. Avec un seuil de trigger à 150 mV, la mesure de la valeur moyenne du pic atteint permet d'étalonner la voie. Ainsi lorsque l'on place la source sur le barreau de scintillateur, les détections radioactives seront beaucoup plus fréquentes que les cosmiques. L'ajustement de la valeur moyenne à une valeur identique pour chacune des voies va donc constituer l'étalonnage des plans.

Adaptation de la carte « trigger »

Exemple de résultats obtenu à partir du programme « correlation.m » sur une journée de mesure.

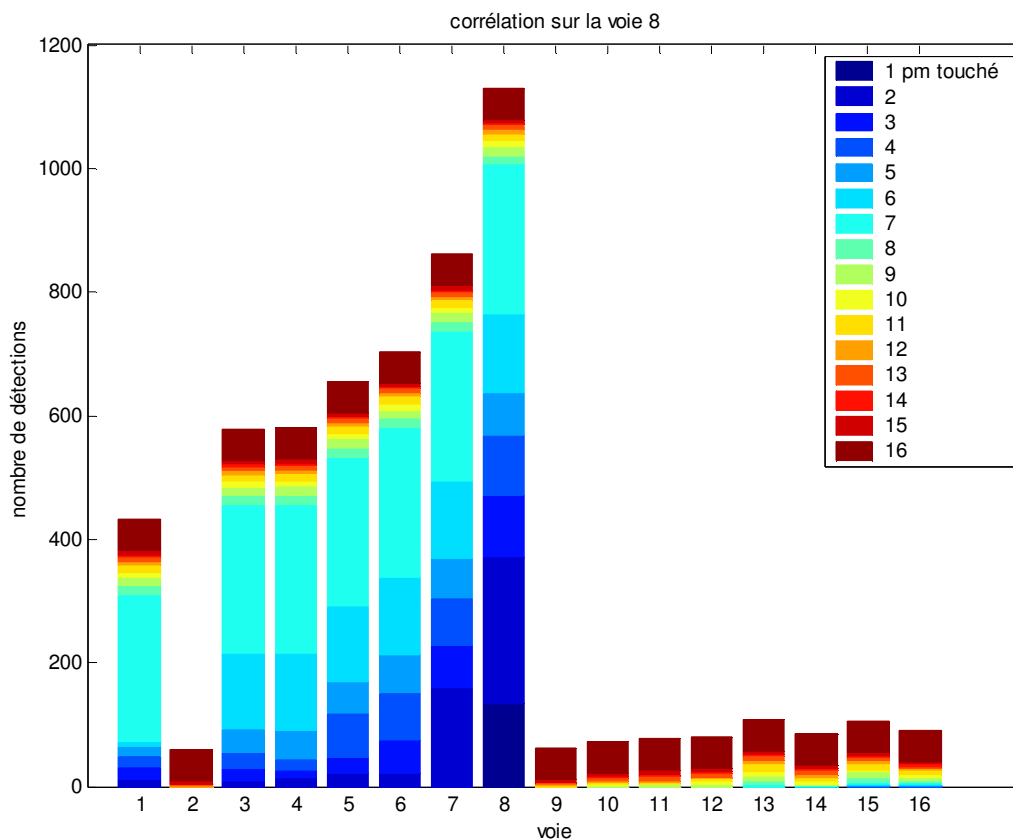


Fig 25 : Résultat du programme d'analyse de diaphonie

Ici nous étudions la huitième voie. La voie 2 est court-circuitée par une charge de 50 ohms placée à l'extrémité d'un câble de 8 ns et la voie 1 est connectée via un transformateur. Les voies 9 à 15 en adaptation d'impédance et la voie 16 reste « en l'air ».

Deux phénomènes perturbateurs apparaissent : le dialogue entre les différentes voies et l'apparition de différence de potentiel entre les deux plans de masses.

Le cross-talk

La propagation du signal par le phénomène de cross-talk (diaphonie) apparaît clairement par l'importance croissante des détections multiples sur les voies qui avoisinent la voie n°8.

Une solution pour résoudre ce problème aurait été de diminuer le gain des PM (baisser leur alimentation) mais en réalité cela n'aurait pas apporté grand-chose. En effet le nombre de détection aurait diminué à moins de rabaisser le seuil de déclenchement des discriminateurs, mais on serait revenu au problème précédent.

La mise en place d'un pont diviseur de tension aurait contribué également au même résultat.

Utilisation d'une diode

Pour résoudre le problème de cross-talk l'utilisation d'une diode pour écrêter le signal a été envisagée.

Dans ce cas j'ai dans un premier temps tracé les caractéristiques de plusieurs diodes : 1n4148 1n4151 et une diode Zéner de tension zéner de 2,7 V (la plus faible valeur disponible au laboratoire).

Les caractéristiques de ces diodes correspondent à la réalité en ce qui concerne la réponse à un signal variant lentement. Par contre si l'on applique un pulse issu d'un PM à une diode, le résultat fait apparaître un temps d'adaptation avant lequel la diode se comporte d'avantage comme un condensateur. Ce temps d'adaptation correspond à la caractéristique T_{rr} (Time of Reverse Recovery) figurant dans certaines documentations techniques.

Une recherche plus approfondie de diode à faible T_{rr} n'a pas permis de trouver de diode ayant des caractéristiques meilleures que celles de la 1n4151 (T_{rr} de 2ns) même parmi les diodes schottky.

La seconde étape qui est celle de l'utilisation de la CAO avec le logiciel Pspice n'a pas permis de trouver solution car il s'est avéré que le T_{rr} n'était pas pris en compte dans le modèle de la diode, ce qui correspond alors à un T_{rr} de 0. Autrement dit le défaut que nous souhaitons éliminer n'apparaissait pas dans les résultats de la simulation. Toutefois cette étape a permis de voir qu'en simulation la diode apporte bien l'effet attendu.

Afin de tenter de réduire le T_{rr} , l'utilisation d'un condensateur en parallèle à la diode a été testé mais ne s'est pas révélé très efficace.

Le problème de masses

Le fonctionnement simultané de plusieurs cartes et appareils interconnectés entre eux fait apparaître de nombreux problèmes de différence de potentiels entre les différents plans de masses. Pour réduire l'effet causé par l'apparition de ces perturbations, plusieurs solutions ont été traitées.

Utilisation d'un transformateur

L'utilisation d'un transformateur permet de limiter le nombre de mauvaises détections, mais cet effet n'est pas aussi puissant que nous pouvions l'attendre. De plus il faudrait câbler un transformateur sur chacune des 64 voies, ce que mènerait à refaire la carte.

Lien entre les différents plans de masses

La masse de l'alimentation Hautes Tensions a été reliée à la structure en aluminium de l'hodoscope. Puis nous avons relié chacun des groupements de masse des plans à la structure. Par la suite, nous avons relié ensemble les masses de la carte de trigger et celles des photomultiplicateurs. Les effets de ce câblage sont relativement réduits face à ce que l'on pouvait s'attendre comme améliorations.

Utilisation des résultats

Les résultats issus de la carte trigger devraient, en théorie, permettre d'identifier systématiquement le chemin suivi par la particule. Or ce n'est pas toujours le cas ; il arrive parfois que certaines informations (lorsque plusieurs voies sont en simultané sur un même plan) ne puisse mener à une localisation. Cependant il existe un cas de figure que nous nommerons « double détection » ne suivant pas cette loi.

La détection d'une particule

Une détection au niveau d'un plan s'effectue lors du passage d'une particule au travers d'un barreau de scintillateur. Comme l'on peut le voir sur la figure 25, la disposition physique des barreaux de scintillateurs est telle que l'on peut avoir deux détections simultanées sur un même plan. Ce cas de figure va être nommé « double détection ».

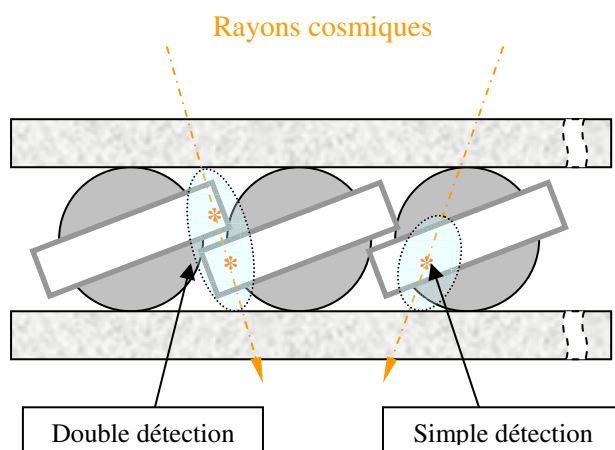


Fig 26 : Coupe d'un plan de voies

Une double détection va permettre d'obtenir une marge d'erreur plus faible sur la localisation de cette détection au niveau du plan considéré, et va donc se répercuter sur la localisation au niveau du couple de plan ainsi que sur la localisation sur le plan de détection.

Globalement, d'après les observations effectuées sur l'ensemble des résultats extraits des acquisitions, les doubles détections occupent environ 10 % des localisations correctement identifiées. L'on pourrait penser que quelques-uns de ces événements, perçus comme des doubles détections sont causés par un phénomène de cross-talk, mais en réalité il est très rare que le cross-talk ne se propage que sur une seule des voies voisines.

III. REALISATIONS INFORMATIQUES (SOFT)

Logiciels utilisés

Les logiciels utilisés dans les réalisations informatiques ont été choisis en fonction des disponibilités au laboratoire.

Excel

Dans un premier temps, le travail d'analyse des données a été effectué sous Excel, mais l'utilisation de ce logiciel ne s'est pas avérée adaptée face à la puissance de calculs nécessaire. Le fichier de données tend rapidement à devenir très volumineux, ce qui mène à un « bug » du PC. Cette utilisation d'Excel même si elle a mené à l'utilisation d'une solution plus adaptée s'est avérée intéressante pour la vérification du fonctionnement correct de chacune des voies et a permis d'amorcer le développement des codes de traitement des données.

Matlab

Matlab s'est présenté comme étant une solution adaptée au problème et d'une efficacité incomparable face à Excel.

De plus de nombreuses options et fonctionnalités du logiciel nous ont permis d'obtenir des résultats directement interprétables, grâce notamment aux diverses représentations graphiques disponibles.

Programmation sous Matlab

Dans un premier temps, pour un programme d'analyse de données, il est indispensable de charger les données, dans notre cas, il s'agira de lire un fichier texte et de charger les données qu'il contient dans des matrices qui seront traitées par la suite dans Matlab.

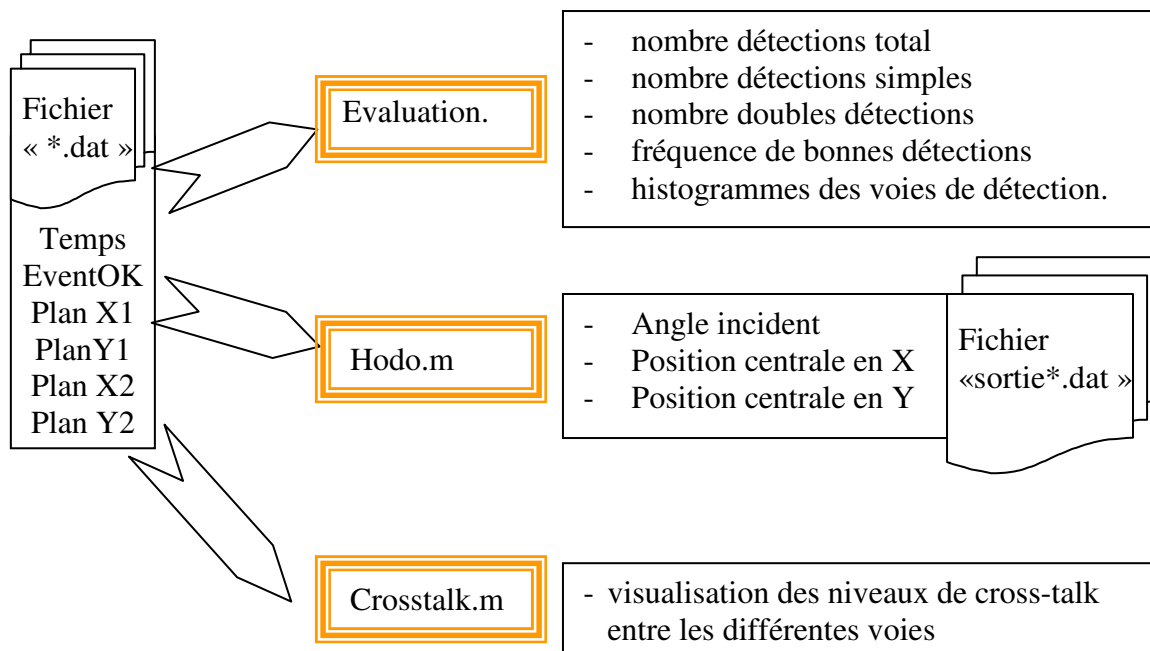
Dans un second temps, il y a une étape de calcul, qui va mener au calcul des éléments désirés.

La dernière étape est l'affichage des résultats à l'écran, soit en format de sortie texte dans la fenêtre d'exécution, soit par l'ouverture de figures sur lesquelles seront placés les courbes, histogrammes ou autres représentations.

Dans le cas d'un programme de simulation, la problématique est de renseigner au préalable l'ensemble des paramètres en conditions d'entrées. Par la suite, la modélisation du phénomène physique doit mener à l'obtention d'un résultat qui pourra par la suite suivre l'étape d'affichage.

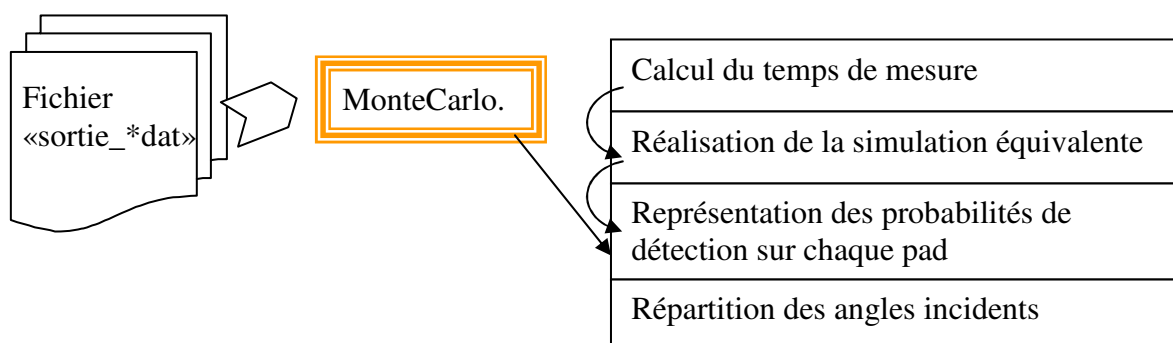
Organigrammes des programmes

Programmes d'interprétation des données



Ces trois programmes permettent, pour le premier (eval.m), d'évaluer le bon fonctionnement du système de détection, pour le second (hodo.m), de générer un fichier avec les informations qui vont nous servir pour l'étalonnage des matrices de silicium, et pour le dernier (crosstalk.m) de représenter les problèmes de dialogue entre les différentes voies.

Programme d'interprétation associé à la simulation



Ce programme de simulation (MonteCarlo.m) va dans un premier temps permettre procéder à la validation de chacune de ces étapes de traitement grâce à un rapprochement de ces données à celles obtenues dans acquisitions réelles. Par la suite, il va permettre, par ajustement de ces paramètres de procéder à l'ajustement optimal de certaines des caractéristiques physiques du banc cosmique.

Vérification des voies

De façon à pouvoir effectuer rapidement une surveillance de l'ensemble des voies du banc cosmique, l'utilisation d'un programme (verification.m) s'avère être une solution rapide et efficace.

Pour cela, le programme va traiter un fichier d'acquisition, effectuer un certain nombre de tests et afficher les résultats relatifs à chacune des voies de chaque plan. La figure 26 correspond au résultat obtenu pour une utilisation en câblant 8 voies sur chacun des plans.

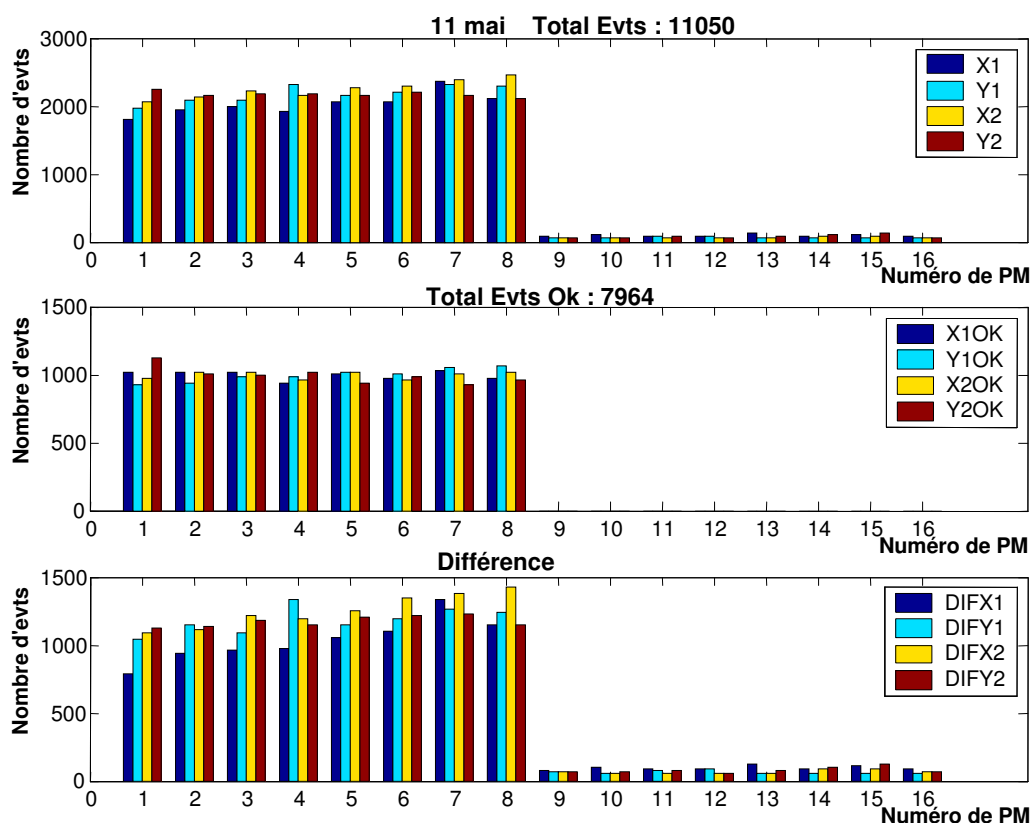


Fig 27 : Réponse de chacune des voies sur une acquisition de 24 H

Cette première étape permet de vérifier que chacune des voies sont correctement câblées et étalonnées. Par ailleurs nous discernons également les détections correctes des mauvaises détections. Dans ce cas, le taux de bonnes détections par PM est voisin de 50%.

Les mauvaises détections sont causées par le phénomène de cross-talk (diaphonie), mais ce dernier voit son ampleur croître proportionnellement à l'amplitude du signal qui le crée. Ainsi pour une amplitude très élevées, l'ensemble des voies vont être touchées par le phénomène et seront ainsi considérées comme touchées par le passage d'une particule.

Pour une tension moins élevée, le cross-talk aura tendance à se propager sur l'ensemble des voies alimentées en tension.

Nous avons pu remarquer que le système se comportait relativement mieux lorsque toutes les voies sont câblées ; ainsi de 78% de détections correctes on est passé à 90%

Modélisation du banc cosmique

Voici une brève présentation des méthodes utilisées pour les localisations et détermination des angles dans les programmes effectués sur Matlab (Cf. annexes).

Localisations sur les couples de plans

La localisation du passage d'une particule se fait en fonction du scintillateur touché et de sa largeur (2,75cm), sur chacun des plans. La marge d'erreur permise est alors de l'ordre d'une demie largeur de scintillateur, sur chacun des plans.

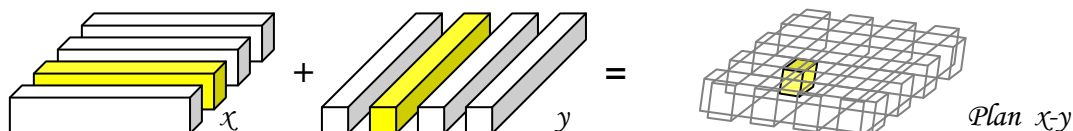


Fig 28 : Localisation sur deux dimensions

Ainsi pour donner une localisation bidimensionnelle du point de passage d'une particule sur un plan, le couplage de deux plans orientés suivant deux axes (x et y) il suffit de disposer d'une localisation sur chacun des plans. A noter que l'on néglige l'écart dû à l'épaisseur d'un plan, très faible par rapport à la distance entre les deux couples de plans.

Localisation sur le plan central

En disposant de deux localisations de passages de particules à deux hauteurs différentes (direction z), l'on obtient donc la trajectoire rectiligne suivie par la particule. Il est alors facilement réalisable de localiser le passage de cette particule à une autre hauteur.

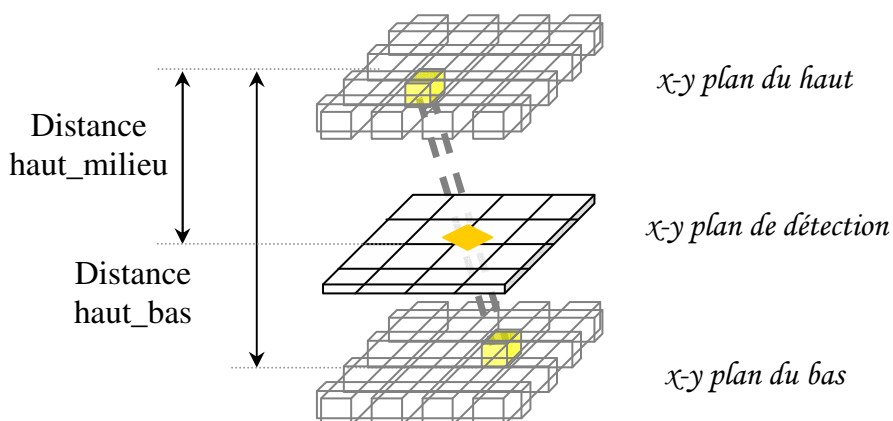


Fig 29 : Localisation du point de passage dans le plan de détection

Pour avoir la localisation des coordonnées sur le plan de détection, on utilise le théorème de Thalès. Les formules obtenues sont alors :

$$\varphi_{\text{déte}} = \varphi_{\text{haut}} + \Delta\varphi \times \frac{\text{Dist}_{\text{haut_milieu}}}{\text{Dist}_{\text{haut_bas}}}$$

avec $\varphi, \text{position} \in (x, y)$ et $\Delta\varphi = (\varphi_{\text{bas}} - \varphi_{\text{haut}})$

Calcul de l'angle incident

Grâce aux localisations sur les deux plans, il est possible également de déterminer l'angle incident avec lequel la particule va traverser le plan du détecteur ; cet angle incident sera identique pour chacun des plans.

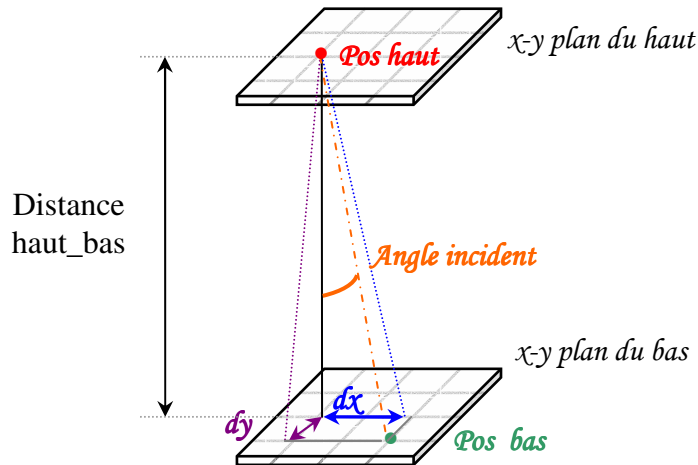


Fig 30 : Vue de l'angle incident part rapport aux plans de localisation

On obtient alors :

$$\tan(\text{Angle incident}) = \frac{\sqrt{(\Delta x)^2 + (\Delta y)^2}}{\text{Dist haut_bas}}$$

soit :

$$\text{Angle incident} = \tan^{-1} \left(\frac{\sqrt{(\Delta x)^2 + (\Delta y)^2}}{\text{Dist haut_bas}} \right)$$

Plus les plans seront rapprochés et plus les angles pourront prendre une valeur élevée. De la même façon une augmentation du nombre de photomultiplicateurs va contribuer à une augmentation de la surface de détection et également celle de la plage de valeur pour les angles.

L'angle maximal mesurable sera celui formé par une particule passant par la diagonale du détecteur.

$$\text{Angle incident max} = \tan^{-1} \left(\frac{\sqrt{(\text{Nb}rePM_x - 1)^2 + (\text{Nb}rePM_y - 1)^2}}{\text{Dist haut_bas}} \times \text{Largeur_Scintillateur} \right)$$

ce qui donne avec une distance entre le haut et le bas de 135 cm :

- 8 photomultiplicateurs en x et en y : 11.40°,
- 16 photomultiplicateurs en x et en y : 23.37°.

Modélisation d'une carte PCB

Afin de pouvoir avoir une représentation des détections finales sur le PCB, il a été nécessaire de le modéliser de façon à ce que chaque pixel (pad) le constituant soit localisé précisément par rapport aux autres.

Le PCB dispose de six wafer, de 36 pads chacun. Afin de grouper les résultats, chaque pad va constituer une cellule de matrice, comme l'illustre la figure suivante :

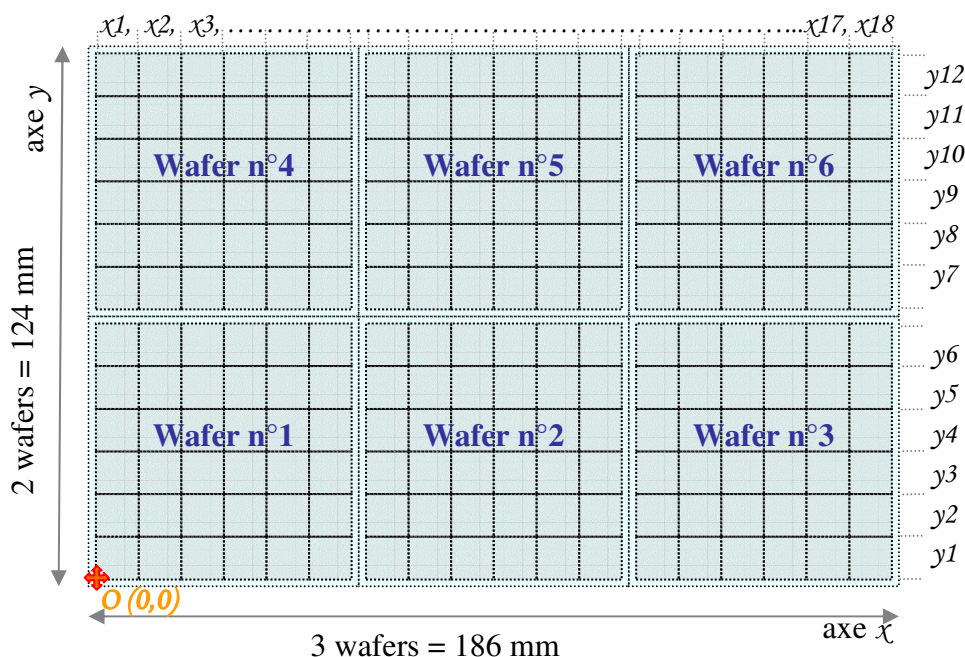


Fig 31 : Matrice de pads sur un PCB

La localisation dans le plan de la carte PCB s'effectuera simplement grâce à la pixellisation par pad. De plus, il devient ainsi aisé de regrouper les 36 pads de chaque Wafer sous Matlab.

La carte PCB, localisée sur le plan de détection doit être judicieusement placée car en effet c'est sa position qui va jouer le plus grand rôle quand au nombre de particules identifiées qui la traverseront.

Ainsi (cf. figures 42, 43 et 44), il sera inévitable de placer le centre du PCB au niveau du centre de la projection des voies câblées sur le plan de détection si l'on souhaite obtenir le taux maximal de détections. Cela se traduira systématiquement par un changement de repère, aussi bien en x qu'en y .

Programme réalisé

Voici les lignes de commandes utilisées pour le changement de repère :

```
% Coordonnées sur plan détection (origine hodo)
XD = x1+dx*B*rap;
YD = y1+dy*B*rap;
% Coordonnées sur plan détection (origine PCB)
X = XD - (dim(2,1)-LongCarte)/2;
Y = YD - (dim(2,2)-LargCarte)/2;
```

La localisation sur les pads :

```

if (X>0 & X<LongCarte & Y>0 & Y<LargCarte ) % sur le PCB
    danspcb = danspcb + 1;           % comptage des évènements
    Dist = -0.5;
    % pour les X
    PixelX=0;
    PixelY=0;
    for wafer = 1:3,
        for pad = 1:6,
            Dist = Dist + 1;
            if (abs(Dist-X)<0.5)
                PixelX = 6*(wafer-1)+pad;
            end
        end
        Dist = Dist + 0.2;
    end
    % Pour les Y
    Dist = -0.5;
    for wafer = 1:2,
        for pad = 1:6,
            Dist = Dist + 1;
            if (abs(Dist-Y)<0.5)
                PixelY = 6*(wafer-1)+pad;
            end
        end
        Dist = Dist + 0.2;
    end
    if (PixelX ~= 0 & PixelY ~= 0) % localisation sur un pad
        pcb(PixelX,PixelY) = pcb(PixelX,PixelY) + 1;
    end
end
end

```

Visualisation des résultats

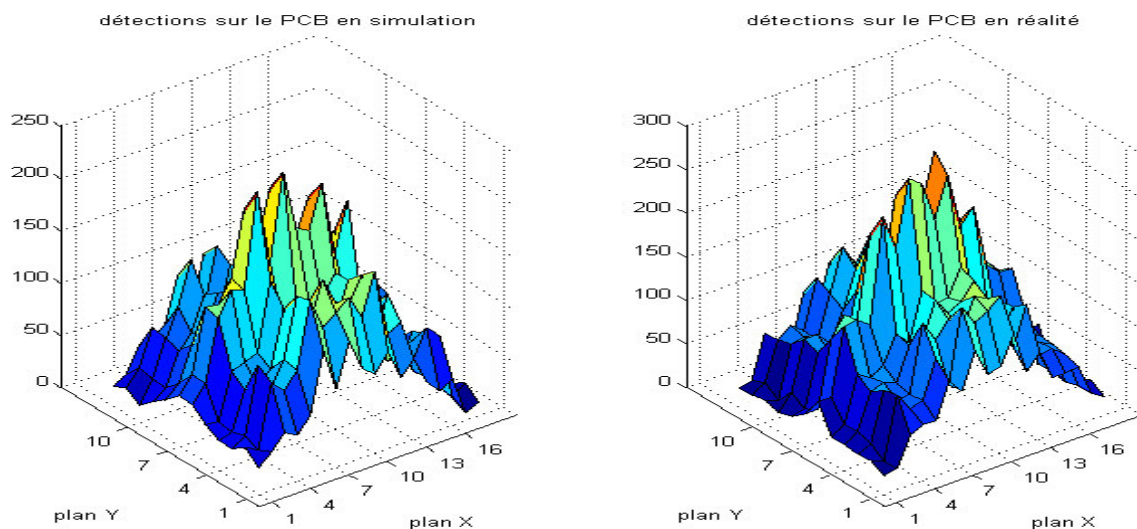


Fig 32 : Détections au niveau du PCB

La représentation de matrices par la fonction « surf » permet de visualiser directement, sous la forme d'une image 3D, la répartition des détections au niveau de chaque pad du PCB. Ces résultats théoriques et pratiques permettent de valider la cohérence du modèle Monte-Carlo que je vais maintenant vous présenter.

Simulation Monte Carlo

La simulation «Monte Carlo », également connue sous le nom de « simulation stochastique » est une méthode d'étude numérique bien adaptée aux rayons cosmiques.

Principe général

L'aspect stochastique de la méthode Monte Carlo peut être uniquement une astuce pour traiter un problème purement déterministe ou bien provenir vraiment d'une modélisation stochastique d'un problème sur lequel on dispose d'une information lacunaire.

Toute réponse donnée par une méthode Monte Carlo est associée à une incertitude de nature statistique. C'est-à-dire qu'il sera possible de lier une probabilité donnée à un résultat encadré par un certain intervalle de confiance.

De la même façon que l'on exige que les expériences physiques soient reproductibles pour être homologuées, les expériences Monte Carlo doivent remplir certaines conditions de reproductibilité.

Selon la complexité du problème à étudier, il est possible d'utiliser des méthodes Monte Carlo adaptées. Deux grandes familles de simulations se distinguent :

- les simulations statiques (ou directe) pour des problèmes de complexité réduite,
- les simulations dynamiques (ou corrélées) pour des problèmes de grande complexité.

Voici une liste de problèmes-type qui peuvent être traités par simulation Monte Carlo directe :

- simulation de fonctionnement (rayonnement cosmique, radioactivité, files d'attente, ou encore diffusion neutronique dans des centrales nucléaires) ;
- anticipation (modèles boursiers ou sociologiques) ;
- test d'hypothèses extrêmes (fissures dans des structures architecturales, cascade des pannes) ;
- cryptographie (codes asymétriques et signature électronique);
- calcul numérique (calcul d'intégrales multidimensionnelles, intégration stochastique, équations différentielles stochastiques).

Application à une particule cosmique

Nous savons que le flux de rayonnement cosmique autour de la direction verticale est de $70 \text{ m}^{-2} \text{ s}^{-1} \text{ stéradian}^{-1}$ au niveau de la mer (cf. annexe). Si l'on souhaite intégrer un tel flux dans une simulation logicielle, il est nécessaire de passer par l'utilisation de la méthode « Monte Carlo ».

Avant de présenter les méthodes utilisées pour déterminer le nombre de passages de particules et répartir aléatoirement les angles des passages de particules, la compréhension du concept de l'angle solide est nécessaire. C'est pourquoi je vais en réaliser une brève présentation.

L'angle solide

La notion d'angle solide permet de préciser la répartition d'un flux dans l'espace. L'angle solide décrit par une direction passant par le point S de l'espace correspond à l'aire de projection de la surface créée par les deux angles θ et φ sur la sphère unitaire centrée en S. Il s'exprime en stéradian (sr).

Dans le cas d'une surface infinitésimale dA , on utilise une approximation de l'angle solide $d\omega$ constitué par dA .

Ce qui donne en considérant que l'angle à 0° est celui d'une particule dont la trajectoire est parallèle à l'axe azimutal de la sphère :

$$d\omega = \frac{dA}{r^2} = \sin \theta \, d\theta \, d\varphi \quad (\text{Cf. représentation de l'angle solide}).$$

On en déduit alors que l'angle solide associé à une sphère est :

$$\begin{aligned} S &= \int_0^{2\pi} \int_0^\pi \sin \theta \, d\theta \, d\varphi = \int_0^{2\pi} [-\cos \theta]_0^\pi \, d\varphi = \int_0^{2\pi} -(-1-1) \, d\varphi \\ &= \int_0^{2\pi} 2 \, d\varphi = [2\varphi]_0^{2\pi} = 4\pi \end{aligned}$$

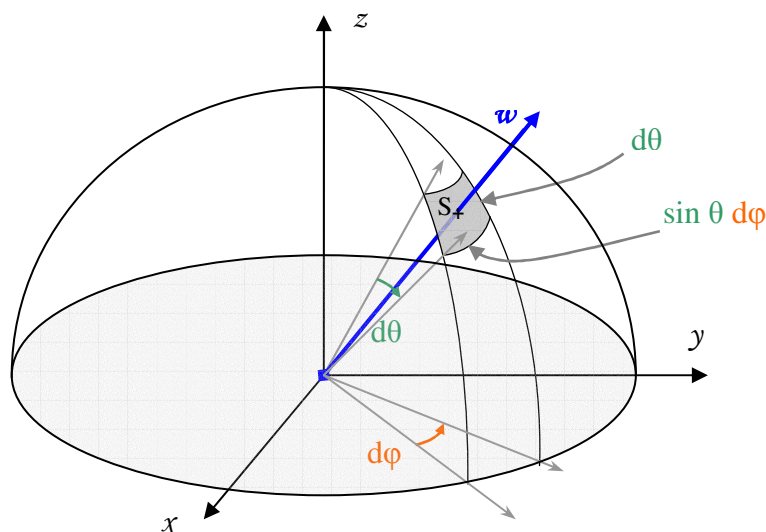


Fig 33 : Représentation de l'angle solide sur une demi-sphère

Le flux incident, Φ_e couvert par cet espace peut permettre de caractériser soit l'éclairement énergétique (cas d'une source) ou alors l'excitance (dans notre cas où l'on n'émet pas).

Ces deux caractéristiques sont en effet calculées de la même façon :

$$E = M = \frac{d\Phi_e}{dA}.$$

La répartition aléatoire des angles

La création d'angles aléatoirement répartis est en réalité plus complexe que ce que l'on pourrait attendre ; elle repose plus précisément sur une répartition aléatoire sur une sphère. Ce point ne doit surtout pas être négligé car il va influencer directement sur les résultats de la simulation « Monte Carlo ». En voici les définitions pour les angles :

- L'angle projeté sur le plan horizontal : φ , sera réparti uniformément entre 0 et 2π .
- L'angle incident : θ , sera réparti entre 0 et $\pi/2$ de manière non uniforme car comme nous l'avons vu précédemment la grandeur de l'élément de surface $d\omega$ varie proportionnellement à $\sin \theta$.

Fonction de répartition : $F_x(x) = \sin x$

Densité de probabilité : $f_x(x) = (\sin \theta)' = \cos x$

Pour obtenir une distribution en $\sin \theta$, à partir d'une distribution uniforme, il faut faire un tirage uniforme en $\cos \theta$.

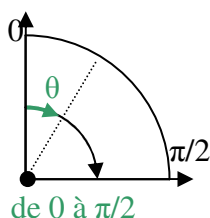


Fig 34 : Angle incident

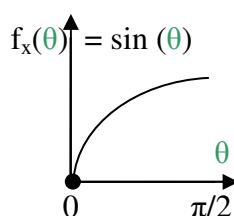


Fig 35 : Densité de probabilité

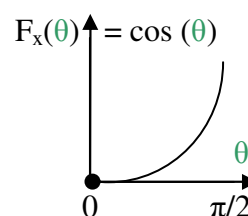


Fig 36 : Fonction de répartition

De plus il faut également tenir compte de l'effet d'atténuation du flux par le passage à travers l'atmosphère qui suit également une loi en $\cos \theta$.

On aura donc une répartition suivant une loi en $\cos^2 \theta$ pour θ compris entre 0 et $\pi/2$.

La forme de la fonction de répartition de x étant connue, il devient possible de la générer à partir d'une variable aléatoire (y) uniformément répartie entre 0 et 1:

$$F_x(x) = y = \cos^2 x$$

d'où :

$$\sqrt{y} = |\cos x|$$

$$\text{or } \sqrt{y} = \cos x$$

entre 0 et $\pi/2$ ce qui donne :

$$x = \arccos(\sqrt{y})$$

Voici une représentation des répartitions de θ et φ sur 40 valeurs, obtenue en 60000 passages :

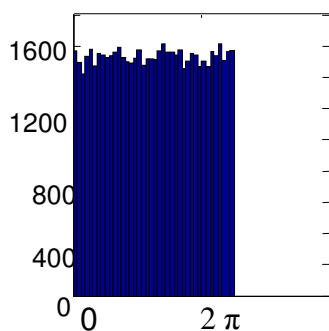


Fig 37 : Répartition de φ

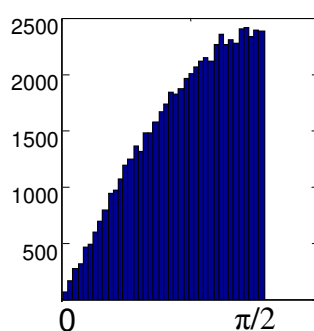


Fig 38 : Répartition de θ en $\cos$

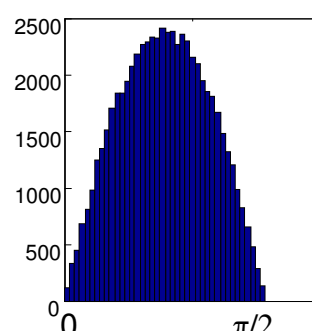


Fig 39 : Répartition de θ en $\cos^2$

L'effet de l'atmosphère va impliquer une diminution progressive du nombre de passage de particules en fonction de l'angle incident de cette dernière.

La détermination du nombre de passages de particules

Pour déterminer le nombre de passages de particules correspondant à une surface et à un temps donné, nous utilisons la répartition des rayons cosmiques sur une demi sphère en procédant par deux intégrations angulaires :

Parcours de l'ensemble de la demi sphère

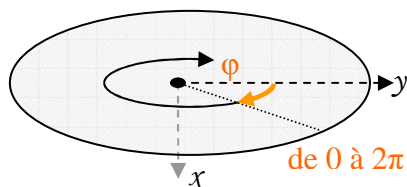


Fig 40 : Angle sur le plan de détection (x y)

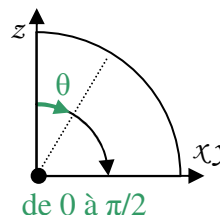


Fig 41 : Angle d'incidence de la particule

En intégrant sur les deux angles (celui du plan de détection et l'incident), on obtient :

$$S = \int_0^{2\pi} \int_0^{\pi/2} \cos \theta \, d\theta \, d\varphi = 2\pi$$

En y appliquant l'effet de l'atmosphère (Cf. plus haut) on obtiendra :

$$S = \int_0^{2\pi} \int_0^{\pi/2} \cos^2 \theta \, d\theta \, d\varphi = \pi$$

Pour obtenir le nombre de passages de particules il suffira alors d'appliquer :

$$\text{NombreParticules} = \Phi_{\text{cosmique}} [m^{-2} \cdot s^{-1} \cdot sr^{-1}] \times \text{Temps} [s] \times \text{Surface} [s] \times S [sr]$$

avec $S = \pi \, sr$

et Φ_{cosmique} : le flux de rayonnement cosmique égal à $70 \, m^{-2} \cdot s^{-1} \cdot sr^{-1}$.

Réalisation du programme de simulation

Utilisation de $\tan \theta$ comme module et de φ pour les positions en x et en y sur le plan 2 :

```
NbreCoups=round(temps*dim(1,1)*dim(1,2)*pi*flux) % passages à simuler

for (i=1:NbreCoups),
    alea=rand(1,4); % génération de 4 valeurs aléatoires
    x1=alea(1)*(dim(1,1)); % position x en cm sur le plan 1
    y1=alea(2)*(dim(1,2)); % position y en cm sur le plan 2
    theta = acos(sqrt(alea(1,3))); % thêta en cos²
    phi = alea(1,4)*2*pi; % phi uniforme

% calcul des coordonnées sur le plan 2
    x2=distance*tan(theta)*sin(phi); % position x en cm
    y2=distance*tan(theta)*cos(phi); % position y en cm

...
end
```

Résultats obtenus

La réalisation d'une telle simulation, même si elle paraît relativement simple s'est avérée assez complexe par la prise en compte de la notion d'angle solide. Cependant l'exécution du programme de simulation a permis de vérifier une bonne corrélation entre le nombre de détections de passages de particules en simulation et celui obtenu en réalité. De plus, une étude plus approfondie des répartitions de passages de particules a permis de vérifier que la répartition des angles en simulation coïncident avec celle issues des mesures. Ce n'est qu'en couplant ces deux aspects (spatial et fréquentiel) que nous avons pu valider la cohérence de la simulation du passage des particules sur le banc cosmique.

Répartition des Angles

Dans le cas d'une détection de particule localisée, voici les résultats obtenus, pour 8 voies câblées, avec une précision de l'ordre du dixième de degré:

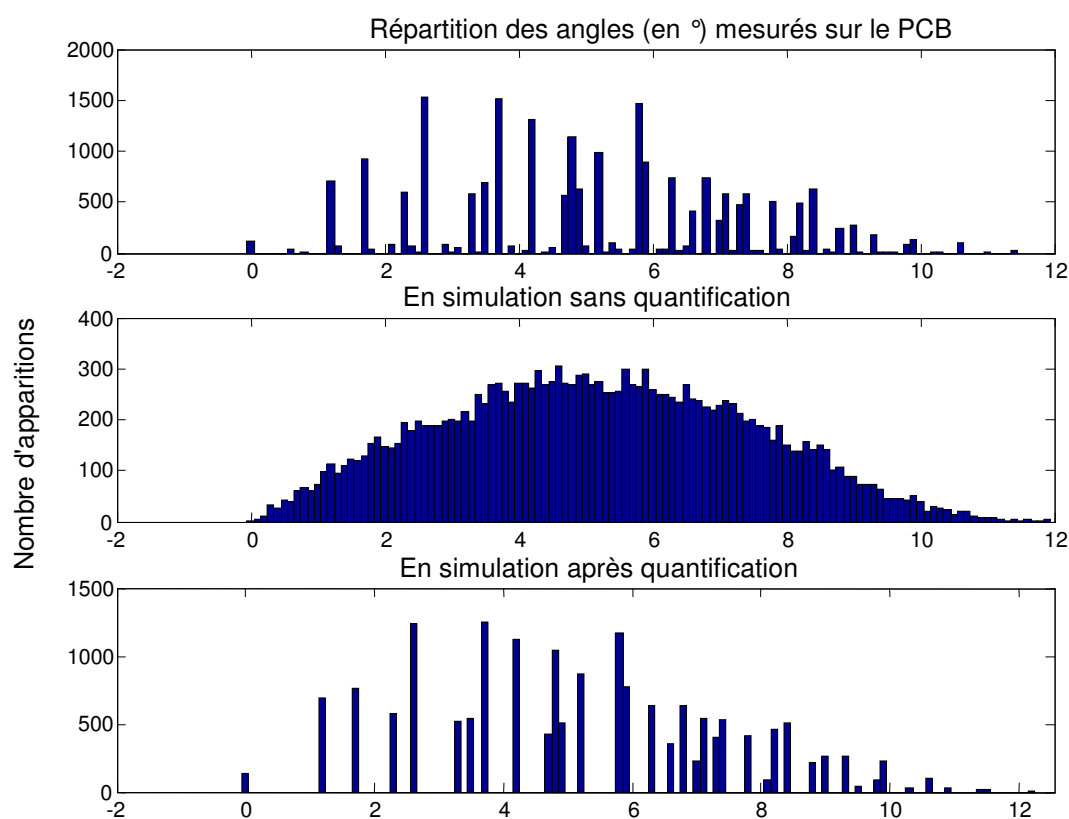


Fig 42 : Répartitions des angles incidents avec 8 voies de câblées par plan

Les mesures tiennent compte des doubles détections c'est pour cela que l'histogramme obtenu comporte quelques barres supplémentaires, mais dans la mesure où ces doubles détections sont minoritaires, l'on considère alors que la simulation et la réalité sont en parfaite coïncidence. Ainsi grâce à la deuxième courbe, nous obtenons une visualisation très proche de celle des rayons cosmiques passant par le PCB. De plus, l'effet de la quantification causé par les voies ressort clairement sur les ensembles de valeurs pouvant être prises par les angles incidents.

Répartitions sur le plan de détection

Afin de valider le modèle, il est également nécessaire de vérifier sa cohérence en terme de répartition spatiale. Pour cela nous prenons en compte le plan de détection.

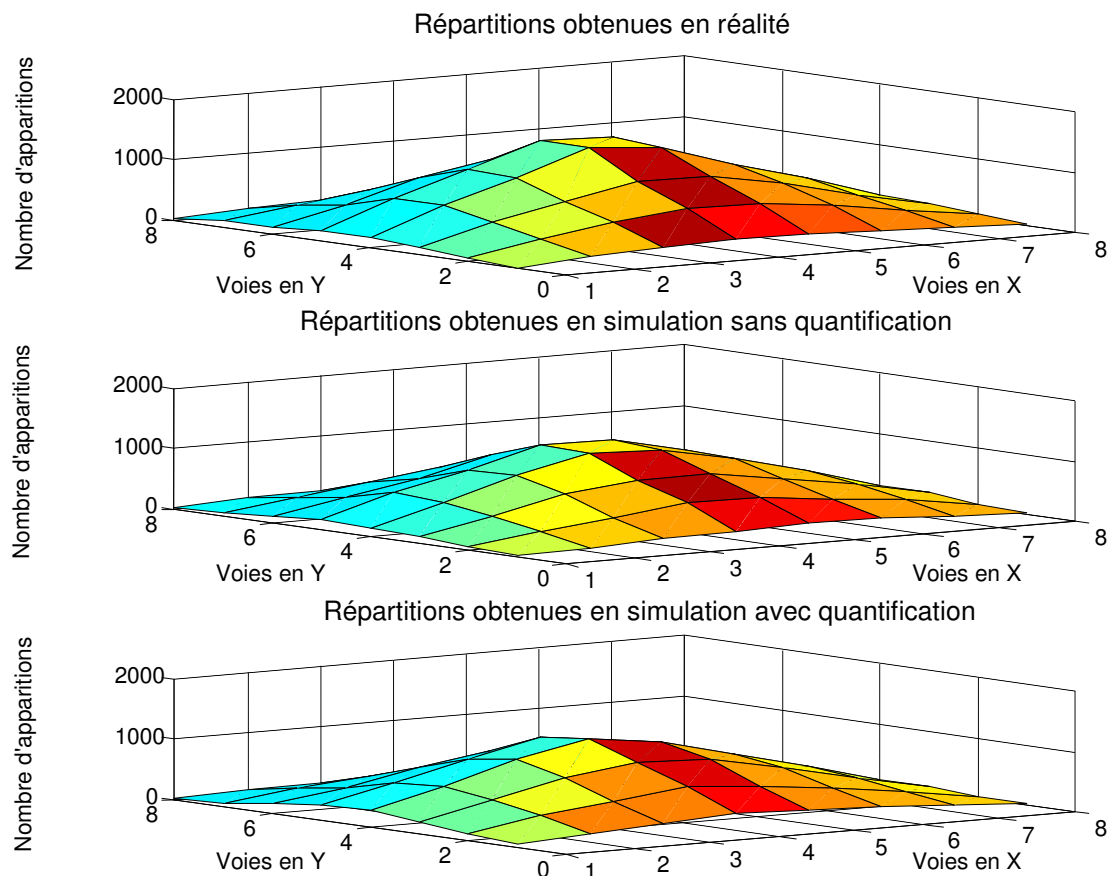


Fig 43 : Répartitions réelles et en simulation

Il est clairement perceptible que pour obtenir le maximum de détections identifiées par le banc, il est indispensable de placer le détecteur sur la partie centrale du plan de détection.

Le programme de simulation permet également de prendre en compte un grand nombre de paramètres tels que le nombre de voies :

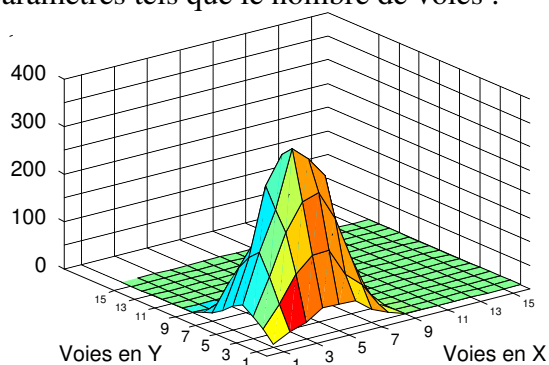


Fig 44 : 24 H de fonctionnement avec 8 voies

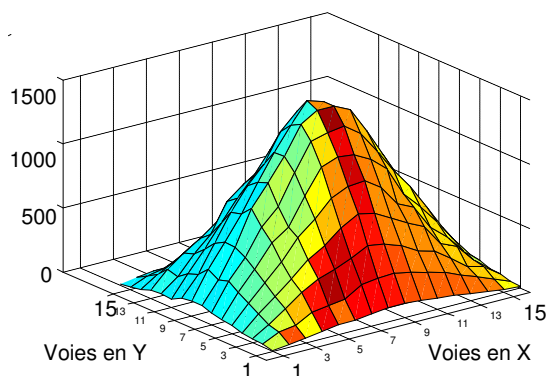


Fig 45 : 24 H de fonctionnement avec 16 voies

Optimisation des détections

L'analyse des résultats de simulation a permis de déterminer l'influence de certains paramètres tels que les tensions d'alimentation et les seuils de détection. De plus, en disposant d'une simulation cohérente, il est possible de vérifier facilement et rapidement l'efficacité d'un réglage physique (distances, nombre de voies...) au niveau du banc de test.

Ajustement de la tension d'entrée

Afin de diminuer le nombre de mauvaises détections la tension d'alimentation a été baissée de 30 V pour chaque PM. Cela a en effet permis de réduire quelque peu le taux de détections erronées, mais en contrepartie le nombre total de détection a chuté à un tel point que cette solution s'est présentée comme désavantageuse.

Extension de 8 à 16 voies

La carte « trigger », initialement prévue pour 16 voies par plan a dans un premier temps, en raison des moyens techniques été réduite à 8 voies. Par la suite, lorsque l'ensemble des câbles supplémentaires nous est parvenu, nous avons pu constater une amélioration des résultats ; le taux de bonnes détections sur chacune des voies est globalement passé de 50% à 70%.

Ecart entre les plans de localisation et de détection

Pour avoir un nombre plus élevé de détections, il est possible de rapprocher les deux couples de plans de localisation, cette étape a pour effets de diminuer la précision quand aux angles et de prendre en compte des angles de valeurs plus élevées. Or, en raison de l'effet d'atténuation atmosphérique sur les particules cosmiques, la probabilité d'obtenir un angle de forte valeur décroît d'autant plus grande que cette valeur est (répartition suivant une loi en $\cos^2$).

Le fait de rapprocher les couples de plans de détections n'aura aucun effet sur la précision de la localisation du passage de la particule.

De plus, il est important de savoir que la zone utile de l'hodoscope est plus grande que celle d'un PCB regroupant 6 wafers.

De manière à avoir une répartition plus homogène, il est également possible de rapprocher le plan de détection de l'un des plans de localisation. Cependant cette pratique ne doit pas être abusive car elle entraîne également une diminution du nombre de détections du fait d'une répartition plus homogène des détections identifiées.

Traitement des données du PCB

Chaque pixel de chaque wafer du PCB renvoie un signal électrique qui est mémorisé à chaque coup de « trigger ». Ainsi en plaçant la carte PCB au centre du plan de détection la probabilité d'obtention de détections sur un des pads de la carte sera maximale.

Soustraction du bruit

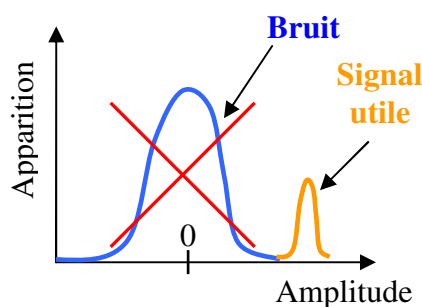


Fig 46 : Signal du MIP

L'on souhaite ne visualiser que le signal utile c'est-à-dire celui du MIP. Pour cela il faut identifier les caractéristiques du bruit sur chacun des pad (moyenne, écart-type (σ)) et ne prendre en compte que les résultats où l'amplitude relevée sera supérieure à une certaine valeur (nous avons choisi la valeur de 4σ qui s'adapte parfaitement à notre cas).

La difficulté dans la détermination des caractéristiques du bruit est de s'assurer que l'on intègre seulement les cas où le rayon cosmique ne passe pas par le pad considéré.

Augmentation du rapport Signal/Bruit

Il est possible de sélectionner les signaux suivant la localisation de leur point d'impact sur la carte PCB. Ainsi la probabilité de détection est accrue et le signal du MIP devient plus important face au bruit.

On a alors :

$$\frac{\text{signal}}{\text{bruit}} = \frac{\text{surface active}}{\text{surface scrutée}} = \frac{1 \text{ wafer}}{\text{nombre de wafers dans la zone retenue}} \quad (\text{celui touché})$$

La taille de la zone retenue dépend de la pixellisation en terme de barreaux scintillateurs. En effet la largeur de ces dernier étant de 27,5 mm la marge d'erreur est égale à la moitié de cette valeur c'est-à-dire 13.75 mm. La détermination du point d'impact sur le PCB ne pourra par conséquent pas se limiter à un seul pad, mais s'étendre aux pads les plus contigus au point de passage déterminé (9, 12 ou 16 pads suivant la localisation).

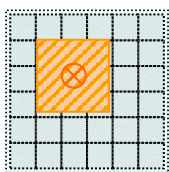


Fig 47 : Cas 1 : 9 pads

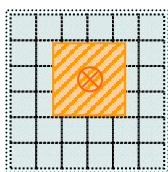


Fig 48 : Cas 2 : 12 pads

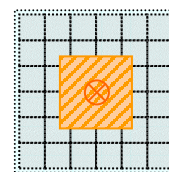


Fig 49 : Cas 3 : 16 pads

Utilisation de « fit »

De manière à améliorer l'extrapolation des paramètres d'une « landau », courbe caractéristique propre au signal des MIP, un ajustement (fit) peut être réalisé. Cette méthode permet d'approximer les valeurs caractéristiques obtenues en mesures, tout en se rapprochant au maximum de la courbe théorique.

Ainsi les résultats obtenus permettent de déterminer plus clairement les caractéristiques du MIP, (valeur moyenne, écart-type). L'ensemble de ces résultats, mis en concordance avec les localisations et angles incidents associés aux particules cosmiques va rendre possible l'étalonnage des matrices de wafers (car en connaissant l'angle incident on peut en déduire la quantité de matière traversée, donc l'énergie déposée par la particule).

Pour mettre en œuvre cette partie, il a choisi d'utiliser une « toolbox », disponible seulement sur des versions de Matlab plus récentes que celle que j'utilisais. C'est donc Franck, fortement impliqué sur la partie électronique du projet qui se penche déjà sur ce problème et va ainsi poursuivre ce travail.

Conclusion

Au terme de ce stage, je peux dire que, même s'ils ont évolués au fil de l'avancement, les objectifs visés ont été atteints et le banc cosmique est actuellement fonctionnel.

Les connaissances théoriques et pratiques, acquises au DESS « Physique des Capteurs et Instrumentation », même si ma formation initiale était faible en Physique, m'ont aidées à m'intégrer dans le milieu professionnel et à maîtriser le sujet.

Ce stage d'analyse, de modélisation et de réalisation a constitué une riche expérience, tant sur le plan technique que sur le plan humain et relationnel.

Sur le plan technique et scientifique, j'ai pu augmenter mes connaissances en Physique théorique et découvrir des démarches scientifiques ainsi que des méthodes de programmation et de simulation tel que « Monte Carlo », qui me seront très probablement utiles à l'avenir.

Sur le plan humain, au-delà de mon travail et du bon accueil qui m'a été fait dans le laboratoire, j'ai pu découvrir que l'importance de la communication et des échanges entre services, au sein d'un projet était tout aussi importante que dans les très grandes entreprises. J'ai également été touché par la passion des chercheurs pour la physique et leur dévouement dans la quête de la recherche de l'origine de la vie par l'étude de l'infiniment petit.

D'un autre point de vue, mes attentes personnelles qui étaient de travailler au sein d'une entité et d'en apprécier ses particularités humaines, techniques et organisationnelles, ont été atteinte. De plus cela m'a permis d'appréhender mon avenir professionnel.

Je vais maintenant pouvoir rendre la main à Franck et Simon qui travaillent sur le projet, respectivement pour la partie électronique et informatique et continuer mon chemin en souhaitant de tout cœur que ma contribution à ce projet aura permis de faire avancer la science.

BIBLIOGRAPHIE

Livres

- « Radiation detection and measurement » *Glenn F.KNOLL (1979)*
« Techniques for Nuclear and Particles Physics Experiments » *William R. Leo*
« Photomultiplicateurs » *Philips Composants (1990)*
« Probability, Random Variables, and Stochastic Processes » *PAPOULIS*
« Compatibilité électromagnétique » *ALAIN CHAROY*

ET SUR INTERNET

Informations sur les structures

<http://polywww.in2p3.fr/>

<http://www.in2p3.fr/>

<http://www.cnrs.fr/>

Informations sur le projet TESLA

<http://polywww.in2p3.fr/flc/>

<http://polywww.in2p3.fr/flc/calice.html>

<http://pg.infn.it/calor2004/program/>

Informations complémentaires

<http://caeau4.in2p3.fr/detecteurs/>

http://www.lal.in2p3.fr/CPEP/atom_fund.html

<http://cdfinfo.in2p3.fr/Experiences/cosmics.html>

http://www.ens-lyon.fr/DSM/magistere/projets_biblio/2003/danache/

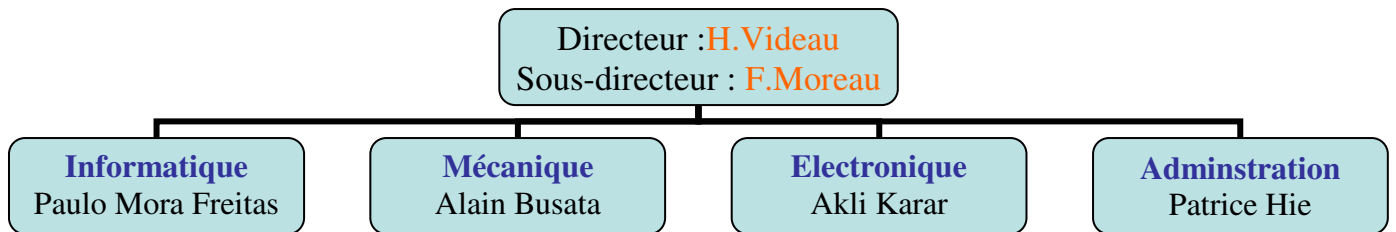
<http://www.sr90.net/>

<http://pdg.lbl.gov/>

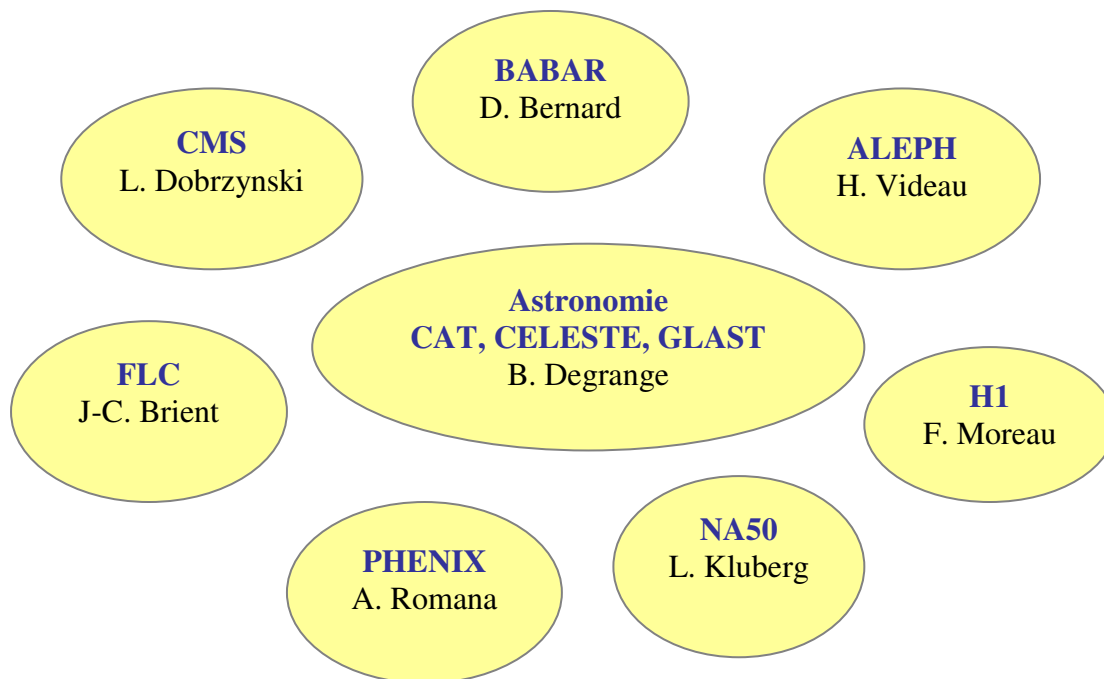
Annexes

1. **Organigramme du LLR**
2. **Les rayons cosmiques**
3. **Le photomultiplicateur**
4. **Programmes sous Matlab**
 - a) **LectureHodo.m**
 - b) **AngleSolideTest.m**
 - c) **MonteCarloVerif.m**
 - d) **Data.m**
 - e) **MoteCarloPCB.m**
 - f) **FinProgramme.m**
 - g) **SelFichier.m**
 - h) **Verif_Soft.m**
 - i) **Evaluation.m**
 - j) **Pcb_hodo.m**

1) Organigramme du LLR



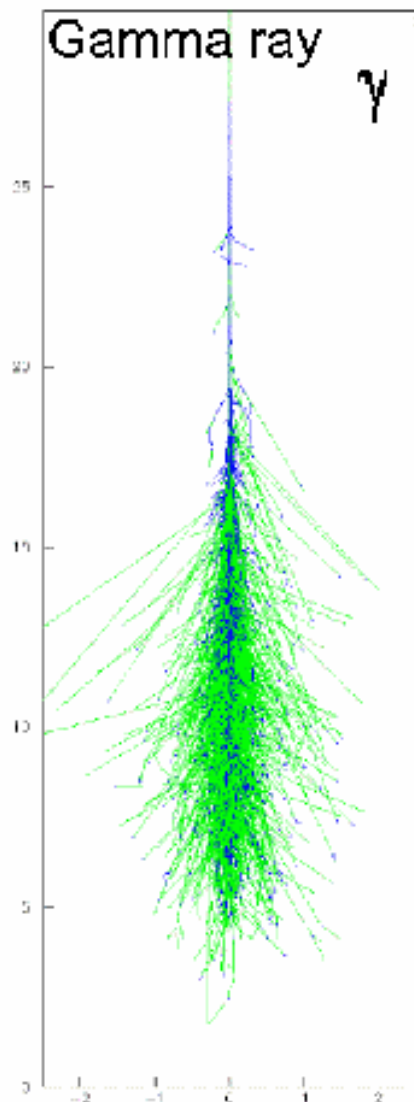
Au niveau des expériences :



2) Les rayons cosmiques

La physique des particules cosmiques est déjà étudiée depuis un certain temps par la communauté scientifique et pourtant, leur origine exacte est encore inconnue. Les énergies mesurées s'étendent sur plus de 30 ordres de grandeur, on a déjà détecté des particules avec des énergies extrêmes pouvant égaler celle d'une balle de tennis au service (10^{20} eV).

Certaines particules cosmiques, en pénétrant dans l'atmosphère, interagissent violemment avec les molécules d'air. Cette première interaction donne naissance à des dizaines de particules qui se partagent l'énergie initiale. Ces particules vont, soit se désintégrer, soit interagir de nouveau avec les molécules d'air, donnant chacune naissance à plusieurs autres particules. Ce processus va se dérouler de proche en proche dans toute l'atmosphère donnant naissance à une véritable cascade de particules élémentaires : une gerbe atmosphérique.



Création d'une gerbe atmosphérique
(échelle Kilométrique)

Comme vous pouvez le voir sur cette simulation, ce n'est qu'une partie de cette gerbe atmosphérique (et non la particule primaire désintégrée lors du premier choc) qui va atteindre le sol. La taille et la densité de particules qui la constituent dépendent de l'énergie de la particule primaire. Des gerbes peuvent contenir des milliards de particules et s'étaler sur plusieurs kilomètres carrés comme l'illustre la représentation ci-contre.

On peut alors tenter de mesurer le passage des particules qui forment ces gerbes atmosphériques en plaçant des détecteurs au sol.

Le rayonnement cosmique primaire se compose de protons, photons, Noyaux et de Neutrinos. Au niveau du sol, se sont des muons cosmiques qui passent.

Le flux moyen de rayonnement cosmique supérieur à 1 GeV autour de la direction verticale est de $70 \text{ m}^{-2} \text{ s}^{-1} \text{ stéradian}^{-1}$ au niveau de la mer.

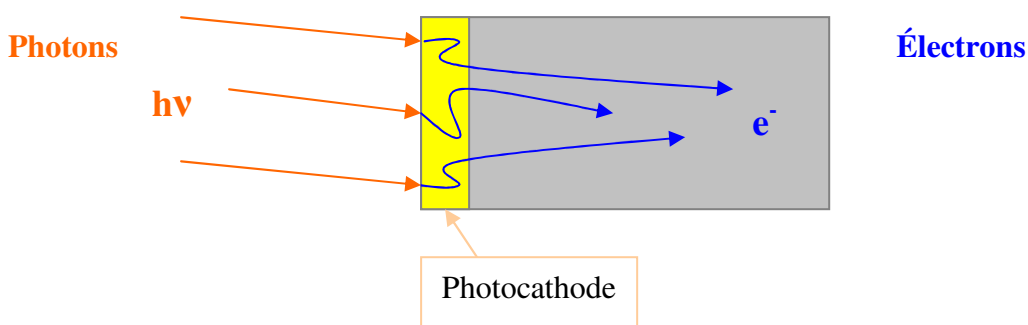
3) Le photomultiplicateur

Principe de fonctionnement

Le photomultiplicateur a la propriété de convertir la lumière (photons) en signal électrique. Son principe de fonctionnement, (une cathode photoémissive associée à des multiplicateurs d'électrons puis à des collecteurs) est connu depuis le début des années 20. Cependant ce n'est que vers la fin des années trente que le photomultiplicateur devient un outil de laboratoire. Dans les années cinquante grâce à l'essor de la physique nucléaire et de la physique atomique il devient de plus en plus commun. Aujourd'hui on l'utilise aussi bien dans la recherche que dans l'industrie que ce soit dans le domaine nucléaire, spatial, médical ou pour la télémétrie de précision.

Mode de fonctionnement

En physique nucléaire un photomultiplicateur est le plus souvent couplé à un scintillateur pour capter la lumière émise par ce dernier. Les rayons lumineux seront guidés, par un guide de lumière, c'est-à-dire un « tunnel transparent » aux parois réfléchissantes, vers la photocathode du PM.



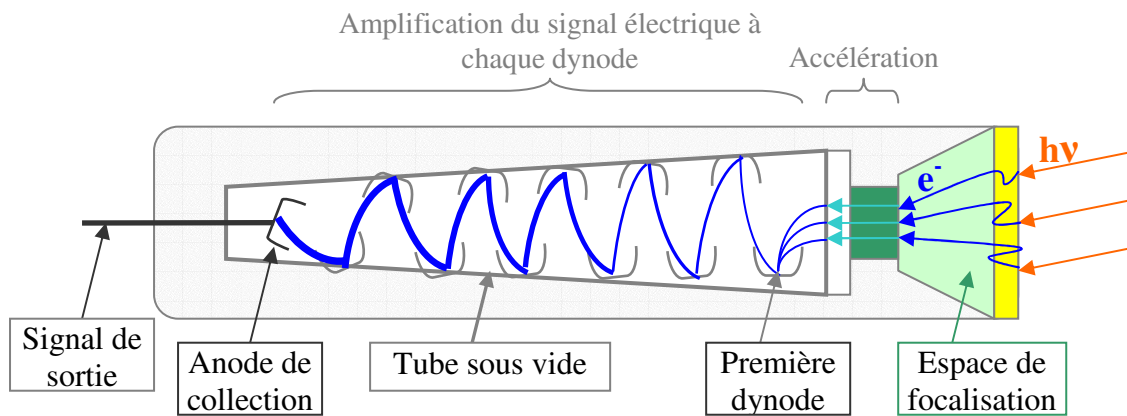
Principe de la photocathode

La photocathode

La photocathode fonctionne par effet photoélectrique, ce qui signifie qu'un photon (particule de lumière) va exciter suffisamment un atome de la photocathode pour lui arracher un électron. Cet électron va transiter à travers la photocathode pour sortir ensuite dans l'espace de focalisation (où il règne le vide). La couche de matière qui sert de photocathode est très faible (quelques centaines de nanomètres au maximum), elle est donc déposée sur une couche de verre qui sert de point d'entrée au photomultiplicateur.

Les photocathodes sont des matériaux à faible affinité électronique. Ce ne sont donc pas les atomes qui sont excités mais les électrons de la bande de valence, qui par l'énergie du photon incident vont « sauter » sur la bande de conduction et ensuite quitter le matériau.

On caractérise une photocathode par son efficacité quantique, c'est le nombre d'électrons émis pour n photons incidents. Cette efficacité quantique dépend essentiellement de la qualité associée à la photocathode.



Vision schématique d'un Photomultiplicateur

Collection

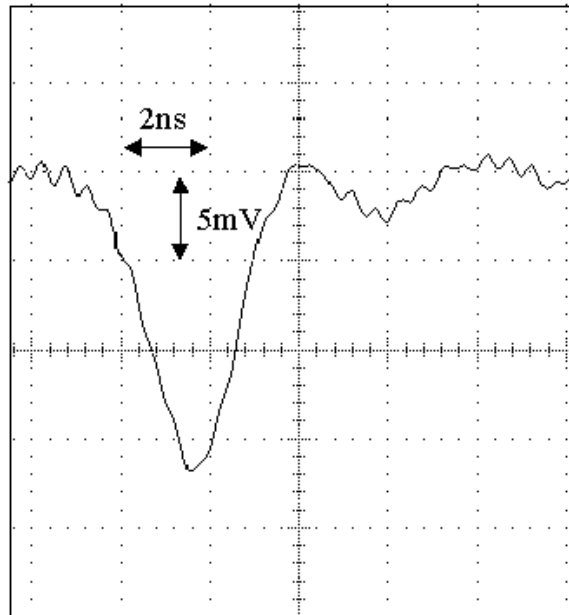
Une fois que les électrons sont émis de la photocathode, ils sont dirigés vers la première dynode. C'est le rôle de l'espace de focalisation et d'accélération. À l'aide d'électrodes mises à un certain potentiel, des lignes de champ sont créées, et les électrons qui sortent de la photocathode vont suivre ces lignes de champ pour finir sur la première dynode. L'efficacité de cette partie du PM, que l'on appelle l'optique d'entrée, est le plus souvent supérieure à 80 % (i.e. pour 10 électrons émis par la photocathode, plus de huit atteignent la première dynode).

Multiplication

Après avoir quitté l'optique d'entrée, les électrons atteignent la première dynode avec une énergie de quelques centaines d'électrons volts. Quand les électrons touchent la dynode, ils provoquent un mécanisme appelé émission secondaire, qui ressemble à l'effet photoélectrique mais avec des électrons comme particule incidente. L'électron qui arrive ainsi avec quelques centaines d'eV va générer quelques dizaines d'électrons de beaucoup plus faible énergie, qui par la différence de potentiel entre la première dynode et la seconde, iront (en accélérant) vers la seconde dynode pour provoquer de nouveau le même mécanisme. En répétant ce mécanisme le long des différents étages de dynodes on peut multiplier les 4 ou 5 premiers photons émis par la photocathode en quelques millions.

Les matériaux qui composent les dynodes sont bien sûr choisis pour leur qualité de photoémission, ils permettent généralement de créer une vingtaine d'électrons. Bien évidemment il y a une efficacité de collection entre chaque dynode mais elle est suffisamment grande pour que l'efficacité de l'étage, c'est-à-dire le nombre d'électrons créés, multiplié par l'efficacité de collection de l'espace inter dynode, soit largement supérieure à 1. Admettons qu'au final, avec les deux effets on gagne un facteur 5 après chaque dynode, si on en a 10, le gain total est 5^{10} soit 10 millions. C'est à dire pour un électron qui sort de la photocathode, 10^7 arrivent sur l'anode.

Exemple de signal en sortie du Photomultiplicateur



Visualisation à l'écran de l'oscilloscope

Ceci est le signal électrique à la sortie de l'anode de collection pour un photoélectron. C'est le signal que l'on obtient lorsque l'on a eu un seul électron qui a été émis par la photocathode. Ce type d'émission apparaît grâce à l'agitation thermique, c'est-à-dire sans particule incidente.

Il permet de voir que pour un électron, au début de la chaîne de multiplication on finit par un signal d'une vingtaine de millivolts.

On peut noter la rapidité du signal, 4 nanosecondes en tout. Le PM utilisé pour cette mesure est un PM réglé pour favoriser le temps de montée du signal (le plus court possible). On aurait pu régler le PM autrement pour favoriser la multiplication d'électrons et obtenir ainsi un signal plus élevé (un facteur 2), mais le temps de montée du signal aurait été plus grand. Le choix entre ces deux réglages dépend de l'utilisation voulue pour le PM. Pour faire des mesures de temps de grandes précisions (quelques ns) on utilisera le premier réglage, pour détecter des particules laissant peu de lumière dans le scintillateur, on utilisera le second.

Dans ce cas, il est important de préciser que nous utilisons des câbles d'impédance de 50 Ω . En effet, le Photomultiplicateur est une source de courant. L'intégrale du pulse, divisée par 50 Ω permet donc d'obtenir la charge fournie par le PM. Si le signal correspond à 1 photon-électron, il sera alors possible de remonter directement jusqu'au gain du photomultiplicateur.

```
%*****V***Chevalier***2004***%
%* LectureHodo.m *
%-----*
%* Lecture d'un fichier du type *.dat *
%* Le fichier devra etre placé dans le répertoire courant *
%* Permet de vérifier le bon fonctionnement de chaque voie *
%*****

%*****
% initialisation
%*****

clear all
close all
clc
help LectureHodo

%*****
% Programme
%*****

NomFichier='WE_8mai.dat'
NomFichier=SelFichier('dat');

% lecture du fichier
types=(' %f %s ');
for i=1:4 , % 4 plans
    for j=1:16, % 16 PM
        types=strcat(types,'%ld');
    end
end
% type donnera:      '%f %s   %ld%d%d%d%d%d%d%d%d%d%d%d%d%d%d
                        %ld%d%d%d%d%d%d%d%d%d%d%d%d%d%d%d
                        %ld%d%d%d%d%d%d%d%d%d%d%d%d%d%d
                        %ld%d%d%d%d%d%d%d%d%d%d%d%d%d%d%'

fprintf('\nLecture du fichier');
[time,EvtOk,...
X1(:,1),X1(:,2),X1(:,3),X1(:,4),X1(:,5),X1(:,6),X1(:,7),X1(:,8),...
X1(:,9),X1(:,10),X1(:,11),X1(:,12),X1(:,13),X1(:,14),X1(:,15),X1(:,16),...
Y1(:,1),Y1(:,2),Y1(:,3),Y1(:,4),Y1(:,5),Y1(:,6),Y1(:,7),Y1(:,8),...
Y1(:,9),Y1(:,10),Y1(:,11),Y1(:,12),Y1(:,13),Y1(:,14),Y1(:,15),Y1(:,16),...
X2(:,1),X2(:,2),X2(:,3),X2(:,4),X2(:,5),X2(:,6),X2(:,7),X2(:,8),...
X2(:,9),X2(:,10),X2(:,11),X2(:,12),X2(:,13),X2(:,14),X2(:,15),X2(:,16),...
Y2(:,1),Y2(:,2),Y2(:,3),Y2(:,4),Y2(:,5),Y2(:,6),Y2(:,7),Y2(:,8),...
Y2(:,9),Y2(:,10),Y2(:,11),Y2(:,12),Y2(:,13),Y2(:,14),Y2(:,15),Y2(:,16),...
] = textread(strcat(NomFichier,'.dat'),types ,'headerlines',8);

%Nombre Total evts
TotalEvt = length(time);
fprintf('\nLe nombre total d'évènements est : %d',TotalEvt)

%Temps moyen de lecture entre deux evts
Meantime = mean(time); % calcul de la moyenne de temps entre deux évènements
TotalTime = sum(time); % temps total de l'acquisition
NbrHeure = TotalTime/3600; % temps total en heures
fprintf('\nLe temps moyen entre deux Evts est : %f \nTemps total : %f heure(s)',Meantime,NbrHeure)

%Calcul du age evt Ok
NbrEvtOk=0; % initialisation du compteur d'évènements Ok
for i=1:length(EvtOk), % parcours de l'ensemble des événements
    if strcmp(EvtOk(i),'OK') % si l'évènement est Ok
        NbrEvtOk=NbrEvtOk+1; % incrémentation du compteur d'évènements Ok
    end
end
fprintf('\nLe nombre d'evt OK est : %d soit %f %%.',NbrEvtOk,NbrEvtOk/TotalEvt*100)
fprintf('\nLe temps moyen entre deux Evts OK est %f',TotalTime/NbrEvtOk)

%Nombre total d'évènements par plan
ToX1 = sum(X1(:,:)); % plan X du haut
ToX2 = sum(X2(:,:)); % plan X du bas
ToY1 = sum(Y1(:,:)); % plan Y du haut
ToY2 = sum(Y2(:,:)); % phan Y du bas
```

```
% colormap(cool)
figure,
subplot(3,1,1);
bar([ToX1',ToY1',ToX2',ToY2'],1,'grouped');
legend('X1','Y1','X2','Y2');
set(gca,'XTick',[0:1:16]);
title(strcat(NomFichier,sprintf('    Total Evts : %d',TotalEvt)), 'FontWeight','bold');
xlabel('Numéro de PM','FontWeight','bold');
ylabel('Nombre d''evts','FontWeight','bold');

%Distrib Event OK
X1_OK=0;           % initialisation du compteur plan X du haut
Y1_OK=0;           % ... plan Y du haut
X2_OK=0;           % ... plan X du bas
Y2_OK=0;           % ... plan Y du bas
for i=1:length(EvtOk), % parcour de l'ensemble des évènements
    if strcmp(EvtOk(i),'OK') % détection simple
        X1_OK = X1_OK + X1(i,:); % comptage du passage sur le plan X du haut
        Y1_OK = Y1_OK + Y1(i,:); % ... plan Y du haut
        X2_OK = X2_OK + X2(i,:); % ... plan X du bas
        Y2_OK = Y2_OK + Y2(i,:); % ... plan Y du bas
    end
end

subplot(3,1,2);
bar([X1_OK',Y1_OK',X2_OK',Y2_OK'],1,'grouped');
legend('X1OK','Y1OK','X2OK','Y2OK');
set(gca,'XTick',[0:1:16]);
title(sprintf('Total Evts Ok : %d',NbrEvtOk), 'FontWeight','bold');
xlabel('Numéro de PM','FontWeight','bold');
ylabel('Nombre d''evts','FontWeight','bold');

% calcul des différences (doubles détections ou erreurs)
DifX1 = ToX1-X1_OK; % au niveau du plan X du haut
DifY1 = ToY1-Y1_OK; % ... plan Y du haut
DifX2 = ToX2-X2_OK; % ... plan X du bas
DifY2 = ToY2-Y2_OK; % ... plan Y du bas

subplot(3,1,3);
bar([DifX1',DifY1',DifX2',DifY2'],1,'grouped');
legend('DIFX1','DIFY1','DIFX2','DIFY2');
set(gca,'XTick',[0:1:16]);
title('Différence','FontWeight','bold');
xlabel('Numéro de PM','FontWeight','bold');
ylabel('Nombre d''evts','FontWeight','bold');

% calcul de pourcentages de bonnes détections
for i=1:16,
    rapX1(i)=100*X1_OK(i)/ToX1(i); % plan X du haut
    rapY1(i)=100*Y1_OK(i)/ToY1(i); % plan Y du haut
    rapX2(i)=100*X2_OK(i)/ToX2(i); % plan X du bas
    rapY2(i)=100*Y2_OK(i)/ToY2(i); % plan Y du bas
end

figure,
bar([rapX1',rapY1',rapX2',rapY2'],1,'grouped');
legend('plan X1','plan Y1','plan X2','plan Y2');
set(gca,'XTick',[0:1:16]);
title(strcat(NomFichier,sprintf('%d événements sur %.1f heures',TotalEvt,TotalTime/3600)), 'FontWeight','bold');
xlabel('Numéro de PM','FontWeight','bold');
ylabel('% d''événements ok','FontWeight','bold');

FinProgramme
```

```
*****V*Chevalier***2004***
%*      AngleSolideTest.m      *
%*-----*
%* Visualisation des répartitions sur les angles théta et phi      *
%* Sert surtout pour les illustrations du rapport      *
%*****

clear all      % suppression des données chargées
close all      % fermeture des fenêtres d'affichage graphique
clc

help AngleSolideTest.m

%*****
% parametre de simulation
%*****
heure=0 ;
min=24;
sec=0;
pts= 1; % nombre de barres pour les histogrammes
        % nombre de valeurs pour la répartition des angles

while pts < 2

    text={'Entrez le nombre de bean pour la représentation:'};
    def={'40'};
    dlgTitle='Détail en degrés';
    lineNo=1;
    answer=inputdlg(text,dlgTitle,lineNo,def);
    pts=str2num(char(answer));
end

pts

%*****
% parametres figés
%*****

largeurPM=2.75;      % largeur d'un barreau de scintillateur en cm
flux=70*(0.01^2);    % flux moyende muons par cm^-2 s^-1 sr^-1
NbrePM=16            % nombre de PM par plan
nbvoies=[NbrePM NbrePM; NbrePM NbrePM ];    % [x1 y1; x2 y2]
dim=nbvoies*largeurPM; % dimension des plans en cm

%*****
% simulation
%*****

% calcul du nombre de passages
temps=3600*heure+60*min+sec;      % on ramène le temps en secondes
NbreCoups=round(temps*dim(1,1)*dim(1,2)*pi*flux) %calcul du nombre de passages à simuler

fprintf('simulation de %d h %2d min %2d sec\n',heure,min,sec);

phi=zeros(NbreCoups,1);      % initialisation de la matrice phi
theta=zeros(NbreCoups,1);    % initialisation de la matrice théta
thetal=zeros(NbreCoups,1);   % initialisation de la matrice théta1

% création des passages
fprintf('max : %5d\n',round(NbreCoups/1000));
for (i=1:1:NbreCoups),      % parcours de l'ensemble des pasages à simuler

    if(mod(i,5000)==0)
        fprintf('\t%d',i/1000);      % affichage pour visualiser que le programme tourne
    end
    if mod(i,50000)==0
        fprintf('\n');
    end

    alea=rand(1,4);      % génération de 4 variables aléatoires
    x1=alea(1)*(dim(1,1)); % localisation en x sur le plan du haut en cm
```

```
y1=alea(2)*(dim(1,2)); % localisation en y sur le plan du haut en cm
theta(i) = acos(alea(1,3)); % theta en considérant que l'angle solide
theta1(i) = acos(sqrt(alea(1,3))); % theta en y appliquant l'effet de l'atmosphère
phi(i) = alea(1,4)*2*pi; % phi uniformément répartis entre 0 et 2pi

% dxB=distance*tan(theta)*sin(phi(i)); % cm
% dyB=distance*tan(theta)*cos(phi(i)); % cm

end

% *****
% affichage des résultats
% *****

figure, % ouverture d'une fenêtre

subplot(1,3,1); % première partie de la fenêtre
hist(phi,pts); % phi
title('phi');
hold on % superposition des affichages

subplot(1,3,2); % seconde partie de la fenêtre
hist(theta,pts); % théta sans l'atmosphère
title('théta en cos');

subplot(1,3,3); % troisième partie de la fenêtre
hist(theta1,pts); % théta avec l'atmosphère
title('théta en cos²');

FinProgramme
```

```
%*****V*Chevalier***2004***
%* MonteCarloVerif.m *
%*-----*
%* Vérification du fonctionnement de la simulation Monte Carlo *
%* Resultats ramenés sur la zone de détection centrale *
%* Le fichier *_Hodo ne doit pas contenir de mauvaise détection *
%* Le fichier '*._Hodo' est créé par le prog 'evaluation.m' *
%*****

%*****
% initialisation
%*****

clear all
close all
clc
help MonteCarloVerif

%*****
% Paramètres du banc
%*****
distance_h_b=75; % (cm)
distance_h_b=135; % (cm)
largeurPM=2.75; % largeur d'un barreau de scintillateur en cm
NbrePM=16;

%NomFichier='essai 07-06-2004'
NomFichier=SelFichier('_Hodo');

% VARIABLES DE MODELISATION
largeurPM=2.75; %
err=largeurPM/2; % erreur max de quantification due aux scintillateurs
flux=70*(0.01^2); % flux moyende muons par cm^-2 s^-1 sr^-1
NbrePM=16 % nombre de voies par plan
nbvoies=[NbrePM NbrePM; NbrePM NbrePM ] % [x1 y1; x2 y2]
dim=nbvoies*largeurPM; % dimensions en cm
Pas=largeurPM;
Nbbin=NbrePM;
repartb=zeros(Nbbin);
repartbq=zeros(Nbbin+1);

repart=zeros(Nbbin);
repartm=zeros(Nbbin);
repartmq=zeros(Nbbin);

PCBreel=zeros(12,18);
pcb=zeros(12,18);
pcbm=zeros(12,18);

LongCarte = 18.4;
LarCarte = 12.2;

distance_h_b=135 % en cm
distance_h_m=75; % en cm
rap=distance_h_m/distance_h_b % calcul du rapport

coincidence= 0;
danspcb=0;
danspcbm=0;
danspcbrel=0;
detectreel=0;
detectsimu=0;

% LECTURE DES DONNEES

[temps,posX,posY,angleincident]=textread(strcat(NomFichier,'*_Hodo'),' %f %f %f %f');
NbLignes=size(temps,1)
tempstotal=sum(sum(temps));
heure= (tempstotal-mod(tempstotal,3600))/3600;
min=(tempstotal-3600*heure-mod(tempstotal,60))/60;
sec=round(tempstotal-3600*heure-60*min);
angle=zeros(NbLignes,1); % initialisation de la matrice des angles
```

```
sprintf('La durée de la mesure est de %d heures, %d min et %d sec',heure, min, sec);
```

```
% MESURES REELES
```

```
for i=1:NbLignes, % ensemble des resultats (que des détections valides)  
% détections pouvant toucher le PCB
```

```
% mesures ramenées au plan du PCB
```

```
X = posX(i) - (dim(2,1)-LongCarte)/2;
```

```
Y = posY(i) - (dim(2,2)-LarCarte)/2;
```

```
if (X>-err & X<LongCarte+err & Y>-err & Y<LarCarte+err )
```

```
    danspcbreel = danspcbreel + 1;
```

```
    Dist = -0.5;
```

```
    % pour les X
```

```
    PixelX=0;
```

```
    PixelY=0;
```

```
    for wafer = 1:3,
```

```
        for pad = 1:6,
```

```
            Dist = Dist + 1;
```

```
            if ((Dist-X)>-0.5 & (Dist-X)<=0.5)
```

```
                PixelX = 6*(wafer-1)+pad;
```

```
            end
```

```
        end
```

```
        Dist = Dist + 0.2;
```

```
    end
```

```
    % Pour les Y
```

```
    Dist = -0.5;
```

```
    for wafer = 1:2,
```

```
        for pad = 1:6,
```

```
            Dist = Dist + 1;
```

```
            if ((Dist-Y)>-0.5 & (Dist-Y)<=0.5)
```

```
                PixelY = 6*(wafer-1)+pad;
```

```
            end
```

```
        end
```

```
        Dist = Dist + 0.2;
```

```
    end
```

```
    if (PixelX ~= 0 & PixelY ~= 0)
```

```
        detectreel=detectreel+1;
```

```
        PCBreel(PixelY,PixelX) = PCBreel(PixelY,PixelX) + 1;
```

```
        angle(detectreel)=angleincident(i);
```

```
    end
```

```
end
```

```
repart(round(0.5+(posX(i))/Pas),round(0.5+(posY(i))/Pas))=...
```

```
repart(round(0.5+(posX(i))/Pas),round(0.5+(posY(i))/Pas))+1;
```

```
end
```

```
% SIMULATION
```

```
temps_simu=temptotal;
```

```
NbreCoups=round(temps_simu*dim(1,1)*dim(1,2)*pi*flux) %calcul du nombre de passages à simuler
```

```
anglesimu=zeros(NbreCoups,1); % initialisation de la matrice des angles
```

```
anglesimunum=zeros(NbreCoups,1); % initialisation de la matrice des angles
```

```
for i=1:1:NbreCoups, % parcours de l'ensemble des passages à simuler
```

```
    if(mod(i,10000)==0)
```

```
        clc
```

```
        sprintf('ligne %d sur %d',i,NbreCoups)
```

```
        % affichage pour visualiser que le programme tourne
```

```
    end
```

```
    alea=rand(1,4);
```

```
    x1=alea(1)*(dim(1,1)); % cm
```

```
    y1=alea(2)*(dim(1,2)); % cm
```

```
    theta = acos(sqrt(alea(1,3)));
```

```
    phi = alea(1,4)*2*pi;
```

```
    dxB=distance_h_b*tan(theta)*sin(phi); % delta x entre les deux plans en cm
```

```
dyB=distance_h_b*tan(theta)*cos(phi);    % delta y entre les deux plans en cm

X2=0;
if (0<(x1+dxB)) & ((x1+dxB)<dim(2,1)) & (0<(y1+dyB)) & ((y1+dyB)<dim(2,2)) % la détection est ok
    coincidence=coincidence+1;

repartm(round(0.5+(x1+dxB*rap)/Pas),round(0.5+(y1+dyB*rap)/Pas))=...
repartm(round(0.5+(x1+dxB*rap)/Pas),round(0.5+(y1+dyB*rap)/Pas))+1;

% SANS QUANTIFICATION

% Coordonnees sur plan détection (origine hodo)
XD = x1+dxB*rap;
YD = y1+dyB*rap;
% Coordonnees sur plan détection (origine PCB)
X = XD - (dim(2,1)-LongCarte)/2;
Y = YD - (dim(2,2)-LarCarte)/2;

if (X>0-err & X<LongCarte+err & Y>0-err & Y<LarCarte+err ) % sur le PCB
    danspcb = danspcb + 1;
    Dist = -0.5;
    % pour les X
    PixelX=0;
    PixelY=0;
    for wafer = 1:3,
        for pad = 1:6,
            Dist = Dist + 1;
            if ((Dist-X)>-0.5 & (Dist-X)<=0.5)
                PixelX = 6*(wafer-1)+pad;
            end
        end
        Dist = Dist + 0.2;
    end
    % Pour les Y
    Dist = -0.5;
    for wafer = 1:2,
        for pad = 1:6,
            Dist = Dist + 1;

            if ((Dist-Y)>-0.5 & (Dist-Y)<=0.5)
                PixelY = 6*(wafer-1)+pad;
            end
        end
        Dist = Dist + 0.2;
    end
    if (PixelX ~= 0 & PixelY ~= 0)
        pcb(PixelY,PixelX) = pcb(PixelY,PixelX) + 1;
    end
end

% QUANTIFICATION PAR PRISE EN COMPTE DES VOIES
% coordonnées de la détection en haut
X1= largeurPM*round((x1+0.5)/largeurPM)-largeurPM/2;    % localisation par rapport à la voie
Y1= largeurPM*round((y1+0.5)/largeurPM)-largeurPM/2;    % ...
% coordonnées de la détection en bas
X2= largeurPM*round((x1+dxB+0.5)/largeurPM)-largeurPM/2;    % ...
Y2= largeurPM*round((y1+dyB+0.5)/largeurPM)-largeurPM/2;    % ...
% écart entre les 2 coordonnées
dXB= X2-X1;    % delta X
dYB= Y2-Y1;    % delta Y

% Coordonnees sur plan détection (origine hodo)
XD = X1+dXB*rap;
YD = Y1+dYB*rap;

% Coordonnees sur plan détection (origine PCB)
Xm=XD - (dim(2,1)-LongCarte)/2;
Ym=YD - (dim(2,2)-LarCarte)/2;
```

```
% Repartition sur plan Detection en pixel FM
if and(XD>0, YD>0)
    repartmq(round(0.5+XD/Pas),round(0.5+YD/Pas))=...
    repartmq(round(0.5+XD/Pas),round(0.5+YD/Pas))+1;
end

% détections pouvant toucher le PCB
if (Xm>-err & Xm<LongCarte+err & Ym>-err & Ym<LarCarte+err )
    danspcbm = danspcbm + 1;
    % pour les X
    Dist = -0.5;
    PixelX=0;
    PixelY=0;
    for wafer = 1:3,
        for pad = 1:6,
            Dist = Dist + 1;
            if ((Dist-Xm)>-0.5 & (Dist-Xm)<=0.5)
                PixelX = 6*(wafer-1)+pad;
            end
        end
        Dist = Dist + 0.2;
    end
    % Pour les Y
    Dist = -0.5;
    for wafer = 1:2,
        for pad = 1:6,
            Dist = Dist + 1;

            if ((Dist-Ym)>-0.5 & (Dist-Ym)<=0.5 )
                PixelY = 6*(wafer-1)+pad;
            end
        end
        Dist = Dist + 0.2;
    end
    if (PixelX ~= 0 & PixelY ~= 0)
        pcbm(PixelY,PixelX) = pcbm(PixelY,PixelX) + 1;
        detectsimu=detectsimu+1;
        anglesimu(detectsimu)=theta*180/pi; % angle en degrés
        anglesimunum(detectsimu)=atan(sqrt(dXB*dXB+dYB*dYB)/distance_h_b)*180/pi;
    end
end
end
end

for i=1:3,
    for j=1:2,
        matricreel(i,j)=sum(sum(PCBreel(1+(j-1)*6:j*6,1+(i-1)*6:i*6)));
        matric(i,j)=sum(sum(pcb(1+(j-1)*6:j*6,1+(i-1)*6:i*6)));
        matricl(i,j)=sum(sum(pcbm(1+(j-1)*6:j*6,1+(i-1)*6:i*6)));
    end
end
end
sprintf('\nEN REALITE')
NbLignes
danspcbreeel
PCBreel'
sum(sum(PCBreel))
matricreel

sprintf('\nEN SIMULATION')

NbreCoups
coincidence
danspcb
pcb'
sum(sum(pcb))
matric

sprintf('\nETAPE DE QUANTIFICATION PAR PM')
danspcbm
pcbm'
sum(sum(pcbm))
matricl
```

```
figure,
subplot(3,1,1)
surfl(pcb)
xlabel('plan X');
ylabel('plan Y');
title('détections sur le PCB en simulation');
set(gca,'xtick',[1:1:18]);
set(gca,'ytick',[1:1:12]);
subplot(3,1,2)
surfl(pcbm)
xlabel('plan X');
ylabel('plan Y');
title('détections sur le PCB avec l''effet des voies en simulation');
set(gca,'xtick',[1:1:18]);
set(gca,'ytick',[1:1:12]);
subplot(3,1,3)
surfl(PCBreel)
xlabel('plan X');
ylabel('plan Y');
title('détections sur le PCB en réalité');
set(gca,'xtick',[1:1:18]);
set(gca,'ytick',[1:1:12]);

decalnum=zeros(12,18);
decalreel=zeros(12,18);
decalreelq=zeros(12,18);

for i=1:12,
    for j=1:18,

        val=pcb(i,j)/pcbm(i,j);
        if val>=1
            decalnum(i,j)= round(100*val);
        else
            decalnum(i,j)=-round(100/val);
        end

        val=PCBreel(i,j)/pcbm(i,j);
        if val>=1
            decalreelq(i,j)= round(100*val);
        else
            decalreelq(i,j)=-round(100/val);
        end

        val=PCBreel(i,j)/pcb(i,j);
        if val>=1
            decalreel(i,j)= round(100*val);
        else
            decalreel(i,j)=-round(100/val);
        end
    end
end

figure,
subplot(3,1,1)
surfl(decalnum)
xlabel('pads en X');
ylabel('pads en Y');
zlabel(' %%');
title(sprintf('écart de quantification ( de -%d %% à +%d %%)',...
    max(max(-decalnum)),max(max(decalnum))));
set(gca,'xtick',[1:1:18]);
set(gca,'ytick',[1:1:12]);

subplot(3,1,2)
surfl(decalreelq)
xlabel('pads en X');
ylabel('pads en Y');
zlabel(' %%');
```

```
title(sprintf('écart entre la simulation quantifiée et la réalité ( de -%d %% à +%d %%)',...
    max(max(-decalreelq)),max(max(decalreelq))));
set(gca,'xtick',[1:1:18]);
set(gca,'ytick',[1:1:12]);

subplot(3,1,3)
surfl(decalreel)
xlabel('pads en X');
ylabel('pads en Y');
zlabel(' %%');
title(sprintf('écart entre la simulation non quantifiée et la réalité ( de -%d %% à +%d %%)',...
    max(max(-decalreel)),max(max(decalreel))));
set(gca,'xtick',[1:1:18]);
set(gca,'ytick',[1:1:12]);

decalnum'
decalreelq'
decalreel'
```

% en bonus, pour les illustrations du rapport

```
figure(10)
subplot(3,1,1)
surfl(repart);
title('répartition mesurées');
zlabel('nombre d''apparitions')
ylabel('voies en Y')
xlabel('voies en X')
subplot(3,1,2)
surfl(repartm);
title('répartitions simulées');
zlabel('nombre d''apparitions')
ylabel('voies en Y')
xlabel('voies en X')
subplot(3,1,3)
surfl(repartmq);
title('répartitions simulées et quantifiées');
zlabel('nombre d''apparitions')
ylabel('voies en Y')
xlabel('voies en X')

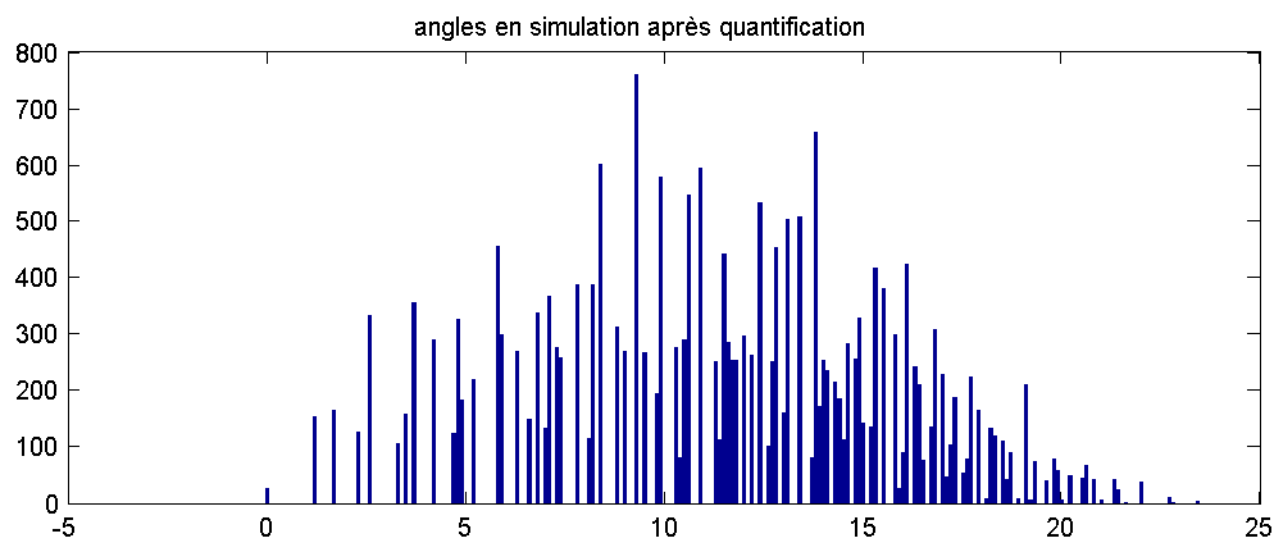
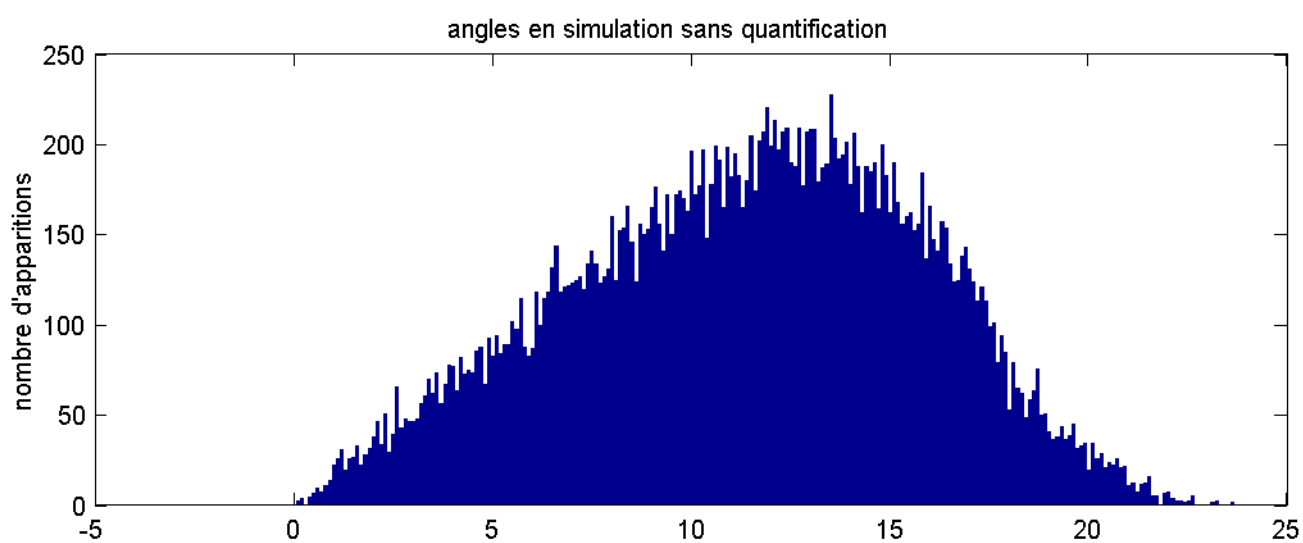
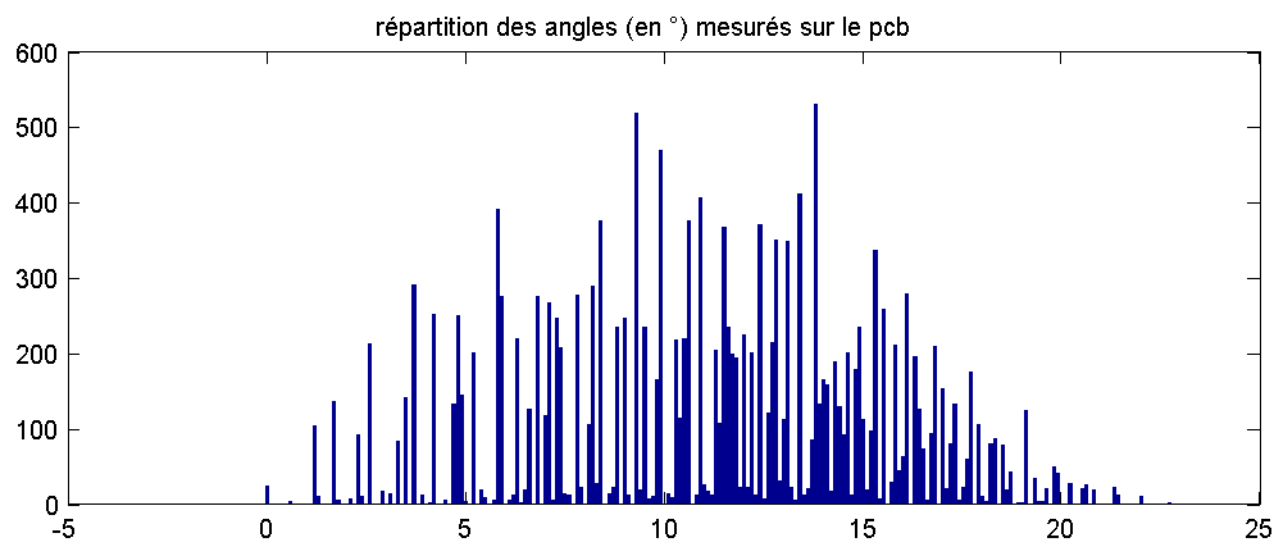
figure(11)
subplot(3,1,1)
hist(angle(1:detectreel),[0:0.1:max(angle(1:detectreel))]);
title('répartition des angles (en °) mesurés sur le pcb');

subplot(3,1,2)
hist(anglesimu(1:detectsimu),[0:0.1:max(anglesimu(1:detectsimu))]);
title('angles en simulation sans quantification');
ylabel('nombre d''apparitions')
subplot(3,1,3)
hist(anglesimunum(1:detectsimu),[0:0.1:max(anglesimunum(1:detectsimu))]);
title('angles en simulation après quantification');

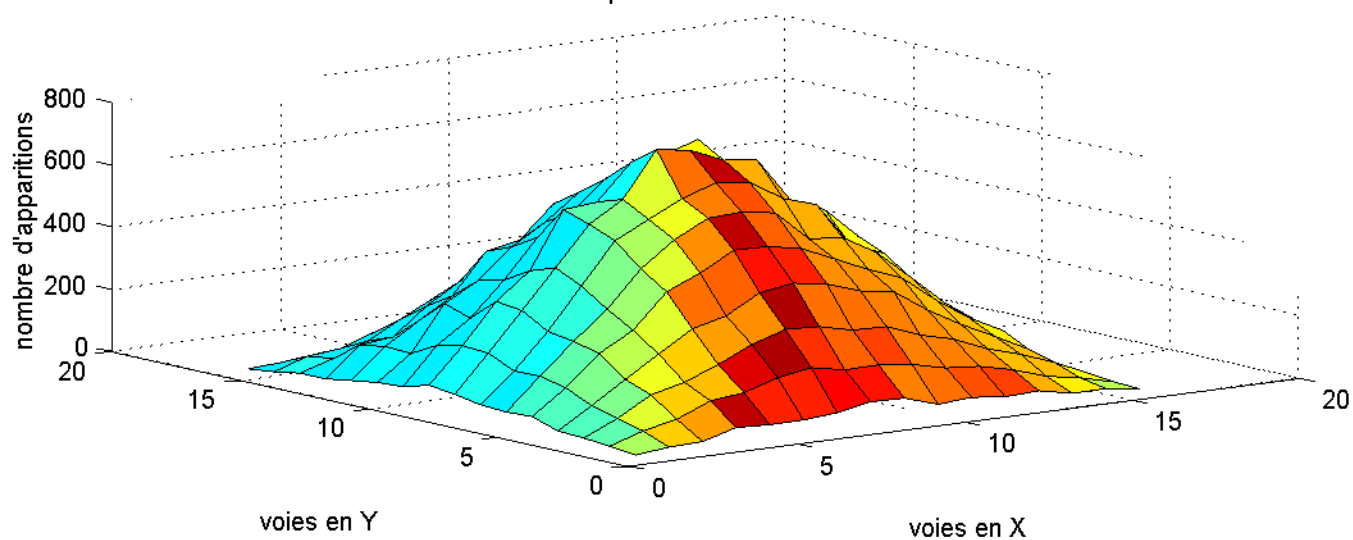
figure(12)
subplot(2,1,1)
%surf(pcbm)
pcolor(pcbm);
colorbar;
xlabel('plan X');
ylabel('plan Y');
title('détections sur le PCB en simulation');
set(gca,'xtick',[1:3:18]);
set(gca,'ytick',[1:3:12]);
```

```
subplot(2,1,2)
%surf(PCBreel)
pcolor(PCBreel);
colorbar;
xlabel('plan X');
ylabel('plan Y');
title('détections sur le PCB en réalité');
set(gca,'xtick',[1:3:18]);
set(gca,'ytick',[1:3:12]);
```

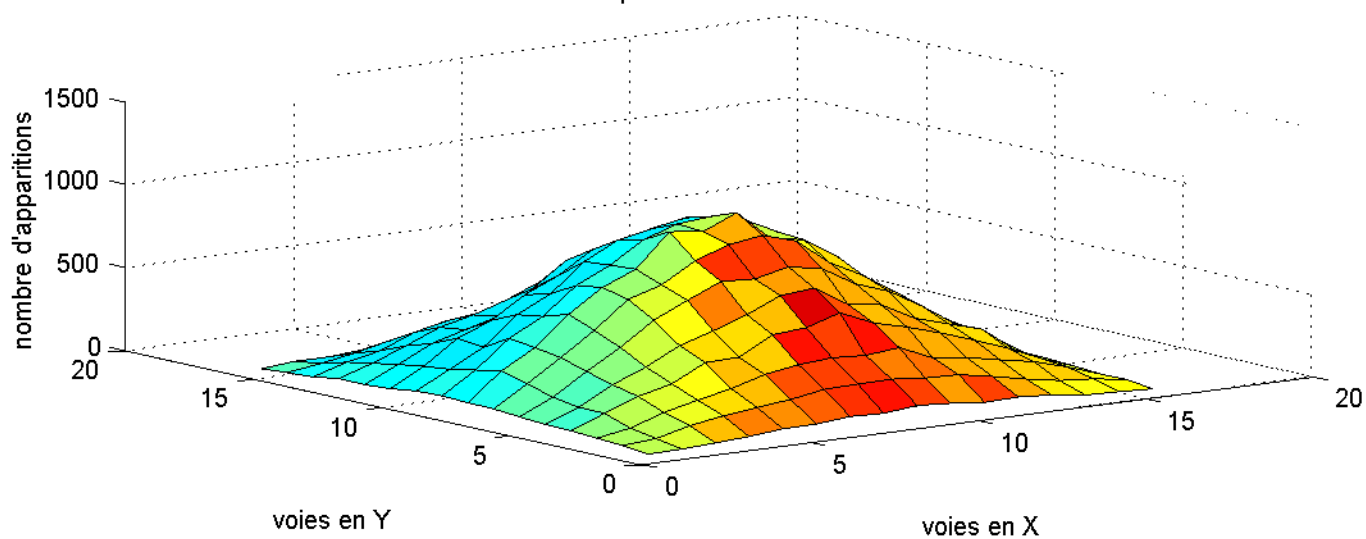
FinProgramme



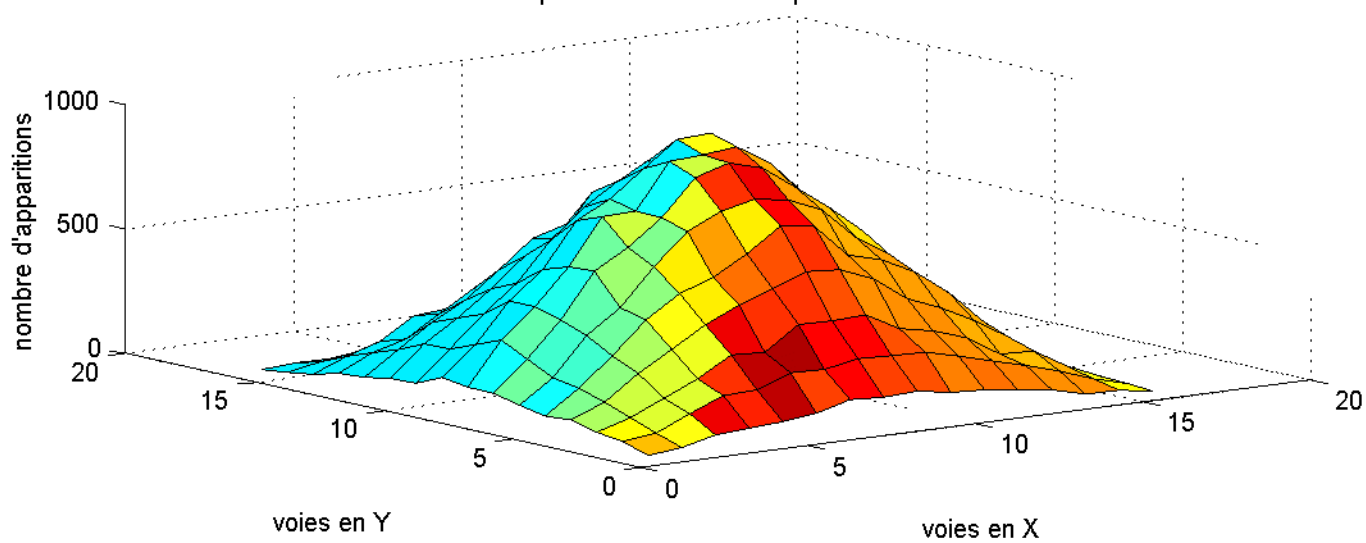
répartition mesurées



répartitions simulées



répartitions simulées et quantifiées



```
%*****V*Chevalier***2004***
%*      Data.m      *
%*-----*
%* Lecture d'un fichier du type 'SLAB#____.dat' *
%* Numero, Temps, posx, posy, angle, vfe l1à12,voies l1à18 *
%* Permet de visualiser l'ensemble des réponses sur chacun des pads *
%*****

%*****
% initialisation
%*****

clear all
close all
clc
help Data.m

%*****
% Programme
%*****

%NomFichier='Slab#0 13_40';
NomFichier=SelFichier('dat');

types=('%d %f %f %f %f');
for i=1:12 ,      % 6 VFE
    for j=1:18,      % 18 pads par VFE
        types=strcat(types,'%f');
    end
end

i=1;

[ Numero, Temps, posx, posy, angle ...
vfe(:,0+i,1),vfe(:,0+i,2),vfe(:,0+i,3),vfe(:,0+i,4),vfe(:,0+i,5),vfe(:,0+i,6),...
    vfe(:,0+i,7),vfe(:,0+i,8),vfe(:,0+i,9),vfe(:,0+i,10),vfe(:,0+i,11),vfe(:,0+i,12),...
    vfe(:,0+i,13),vfe(:,0+i,14),vfe(:,0+i,15),vfe(:,0+i,16),vfe(:,0+i,17),vfe(:,0+i,18),...
vfe(:,1+i,1),vfe(:,1+i,2),vfe(:,1+i,3),vfe(:,1+i,4),vfe(:,1+i,5),vfe(:,1+i,6),...
    vfe(:,1+i,7),vfe(:,1+i,8),vfe(:,1+i,9),vfe(:,1+i,10),vfe(:,1+i,11),vfe(:,1+i,12),...
    vfe(:,1+i,13),vfe(:,1+i,14),vfe(:,1+i,15),vfe(:,1+i,16),vfe(:,1+i,17),vfe(:,1+i,18),...
vfe(:,2+i,1),vfe(:,2+i,2),vfe(:,2+i,3),vfe(:,2+i,4),vfe(:,2+i,5),vfe(:,2+i,6),...
    vfe(:,2+i,7),vfe(:,2+i,8),vfe(:,2+i,9),vfe(:,2+i,10),vfe(:,2+i,11),vfe(:,2+i,12),...
    vfe(:,2+i,13),vfe(:,2+i,14),vfe(:,2+i,15),vfe(:,2+i,16),vfe(:,2+i,17),vfe(:,2+i,18),...
vfe(:,3+i,1),vfe(:,3+i,2),vfe(:,3+i,3),vfe(:,3+i,4),vfe(:,3+i,5),vfe(:,3+i,6),...
    vfe(:,3+i,7),vfe(:,3+i,8),vfe(:,3+i,9),vfe(:,3+i,10),vfe(:,3+i,11),vfe(:,3+i,12),...
    vfe(:,3+i,13),vfe(:,3+i,14),vfe(:,3+i,15),vfe(:,3+i,16),vfe(:,3+i,17),vfe(:,3+i,18),...
vfe(:,4+i,1),vfe(:,4+i,2),vfe(:,4+i,3),vfe(:,4+i,4),vfe(:,4+i,5),vfe(:,4+i,6),...
    vfe(:,4+i,7),vfe(:,4+i,8),vfe(:,4+i,9),vfe(:,4+i,10),vfe(:,4+i,11),vfe(:,4+i,12),...
    vfe(:,4+i,13),vfe(:,4+i,14),vfe(:,4+i,15),vfe(:,4+i,16),vfe(:,4+i,17),vfe(:,4+i,18),...
vfe(:,5+i,1),vfe(:,5+i,2),vfe(:,5+i,3),vfe(:,5+i,4),vfe(:,5+i,5),vfe(:,5+i,6),...
    vfe(:,5+i,7),vfe(:,5+i,8),vfe(:,5+i,9),vfe(:,5+i,10),vfe(:,5+i,11),vfe(:,5+i,12),...
    vfe(:,5+i,13),vfe(:,5+i,14),vfe(:,5+i,16),vfe(:,5+i,16),vfe(:,5+i,17),vfe(:,5+i,18),...
vfe(:,6+i,1),vfe(:,6+i,2),vfe(:,6+i,3),vfe(:,6+i,4),vfe(:,6+i,5),vfe(:,6+i,6),...
    vfe(:,6+i,7),vfe(:,6+i,8),vfe(:,6+i,9),vfe(:,6+i,10),vfe(:,6+i,11),vfe(:,6+i,12),...
    vfe(:,6+i,13),vfe(:,6+i,14),vfe(:,6+i,15),vfe(:,6+i,16),vfe(:,6+i,17),vfe(:,6+i,18),...
vfe(:,7+i,1),vfe(:,7+i,2),vfe(:,7+i,3),vfe(:,7+i,4),vfe(:,7+i,5),vfe(:,7+i,6),...
    vfe(:,7+i,7),vfe(:,7+i,8),vfe(:,7+i,9),vfe(:,7+i,10),vfe(:,7+i,11),vfe(:,7+i,12),...
    vfe(:,7+i,13),vfe(:,7+i,14),vfe(:,7+i,15),vfe(:,7+i,16),vfe(:,7+i,17),vfe(:,7+i,18),...
vfe(:,8+i,1),vfe(:,8+i,2),vfe(:,8+i,3),vfe(:,8+i,4),vfe(:,8+i,5),vfe(:,8+i,6),...
    vfe(:,8+i,7),vfe(:,8+i,8),vfe(:,8+i,9),vfe(:,8+i,10),vfe(:,8+i,11),vfe(:,8+i,12),...
    vfe(:,8+i,13),vfe(:,8+i,14),vfe(:,8+i,15),vfe(:,8+i,16),vfe(:,8+i,17),vfe(:,8+i,18),...
vfe(:,9+i,1),vfe(:,9+i,2),vfe(:,9+i,3),vfe(:,9+i,4),vfe(:,9+i,5),vfe(:,9+i,6),...
    vfe(:,9+i,7),vfe(:,9+i,8),vfe(:,9+i,9),vfe(:,9+i,10),vfe(:,9+i,11),vfe(:,9+i,12),...
    vfe(:,9+i,13),vfe(:,9+i,14),vfe(:,9+i,15),vfe(:,9+i,16),vfe(:,9+i,17),vfe(:,9+i,18),...
vfe(:,10+i,1),vfe(:,10+i,2),vfe(:,10+i,3),vfe(:,10+i,4),vfe(:,10+i,5),vfe(:,10+i,6),...
    vfe(:,10+i,7),vfe(:,10+i,8),vfe(:,10+i,9),vfe(:,10+i,10),vfe(:,10+i,11),vfe(:,10+i,12),...
    vfe(:,10+i,13),vfe(:,10+i,14),vfe(:,10+i,15),vfe(:,10+i,16),vfe(:,10+i,17),vfe(:,10+i,18),...
vfe(:,11+i,1),vfe(:,11+i,2),vfe(:,11+i,3),vfe(:,11+i,4),vfe(:,11+i,5),vfe(:,11+i,6),...
    vfe(:,11+i,7),vfe(:,11+i,8),vfe(:,11+i,9),vfe(:,11+i,10),vfe(:,11+i,11),vfe(:,11+i,12),...
    vfe(:,11+i,13),vfe(:,11+i,14),vfe(:,11+i,15),vfe(:,11+i,16),vfe(:,11+i,17),vfe(:,11+i,18),...
]=textread(strcat(NomFichier,'.dat'),types,'headerlines',13, 'commentstyle','c++');
```

NbLignes=length(Temps);

```
beanning=2048;

detect=zeros(12,18,2*beanning);
moyenne=zeros(12,18);
ecarttype=zeros(12,18);

for i=1:NbLignes,
    if(mod(i,1000)==0)
        clc
        sprintf('ligne %d sur %d',i,NbLignes)
    end

    for num_vfe=1:12, %12
        for pad=1:18,
            %moyenne(num_vfe,pad)=mean(vfe(:,num_vfe,pad)); %moyenne
            %ecarttype(num_vfe,pad)=std(vfe(:,num_vfe,pad)); %ecarttype
            bean=vfe(i,num_vfe,pad)+beanning+1;
            detect(num_vfe,pad,bean)=detect(num_vfe,pad,bean)+1;
        end
    end
end

for num_vfe=1:12
    a=zeros(18,2*beanning);
    clear text
    clear maxi

    for pad=1:18,
        a(pad,:)=detect(num_vfe,pad,:);
        maxi(pad)=1;
        for j=1:2*beanning
            if detect(num_vfe,pad,j)>detect(num_vfe,pad,maxi(pad))
                maxi(pad)=j;
            end
        end
        text(pad)=cellstr(sprintf('%2d %2.2f %4d',pad,std(a(pad,:)),maxi(pad)-beanning));
    end

    figure
    plot([-beanning:beanning-1],a(1:7,1:2*beanning),'-',...
        [-beanning:beanning-1],a(8:14,1:2*beanning),'.-',...
        [-beanning:beanning-1],a(15:18,1:2*beanning),'*-');

    title(sprintf('vfe numéro %d',num_vfe));
    legend(text);
    xlabel('beanning');
    ylabel('nombre de coups');
end
finprogramme;

for num_vfe=1:12, %12

a=zeros(18,beanning);
ampl=zeros(18,beanning);
    for pad=1:18,
        [ampl(pad,:) a(pad,:)]=hist(vfe(:,num_vfe,pad),beanning);
    end
    figure
    plot(a(1:8,:),ampl(1:8:,:),'-',...
        a(9:16,:),ampl(9:16:,:),'.-',...
        a(17:18,:),ampl(17:18:,:), '*-');
    title(sprintf('vfe numéro %d',num_vfe));
    legend(text);
    xlabel('beanning');
    ylabel('nombre de coups');
end

finprogramme;
```

```
%*****V*Chevalier***2004***
%*      MonteCarloPCB.m      *
%*-----*
%* Aplication de la simulation Monte Carlo      *
%* Et résultats en terme de voies PCB      *
%*****
```

```
clear all
close all
clc
```

```
help MonteCarloPCB
```

```
%*****
% parametre de simulation
%*****
heure=3 ;
min=0;
sec=0;
```

```
%*****
% parametres figés
%*****
```

```
largeurPM=2.75;      % en cm
flux=70*(0.01^2);    % flux moyende muons par cm^-2 s^-1 sr^-1
a=16
nbvoies=[a a; a a ] % [x1 y1; x2 y2]
dim=nbvoies*largeurPM;
Pas=largeurPM;
Nbbin=round(44/Pas);
```

```
LongCarte = 18.4;
LarCarte = 12.2;
```

```
distance_h_b=135;    % en cm
distance_h_m=75;     % en cm
rap=distance_h_m/distance_h_b % calcul du rapport
```

```
%*****
% programme
%*****
```

```
%initialisations
```

```
repartm=zeros(Nbbin);
repartmq=zeros(Nbbin);
```

```
pcb=zeros(18,12);
pcbm=zeros(18,12);
```

```
coincidence= 0;
danspcb=0;
danspcbm=0;
```

```
% Simulation
```

```
temps=3600*heure+60*min+sec;          % on ramène le temps en secondes
NbreCoups=round(temps*dim(1,1)*dim(1,2)*pi*flux) %calcul du nombre de passages à simuler
```

```
angle=zeros(NbreCoups,1); % initialisation de la matrice des angles
```

```
for (i=1:1:NbreCoups),          % parcours de l'ensemble des pasages à simuler

    if(mod(i,10000)==0)
        clc
        i          % affichage pour visualiser que le programme tourne
        NbreCoups
    end
```

```
alea=rand(1,4);
x1=alea(1)*(dim(1,1)); % cm
y1=alea(2)*(dim(1,2)); % cm
theta = acos(sqrt(alea(1,3)));
phi = alea(1,4)*2*pi;

dxB=distance_h_b*tan(theta)*sin(phi); % delta x entre les deux plans en cm
dyB=distance_h_b*tan(theta)*cos(phi); % delta y entre les deux plans en cm

X2=0;
if (0<(x1+dxB)) & ((x1+dxB)<dim(2,1)) & (0<(y1+dyB)) & ((y1+dyB)<dim(2,2)) % la détection est ok
    coincidence=coincidence+1;
    angle(coincidence)=theta*180/pi;

    repartm(round(0.5+(x1+dxB*rap)/Pas),round(0.5+(y1+dyB*rap)/Pas))=...
    repartm(round(0.5+(x1+dxB*rap)/Pas),round(0.5+(y1+dyB*rap)/Pas))+1;

% SANS QUANTIFICATION

% Coordonnees sur plan détection (origine hodo)
XD = x1+dxB*rap;
YD = y1+dyB*rap;
% Coordonnees sur plan détection (origine PCB)
X = XD - (dim(2,1)-LongCarte)/2;
Y = YD - (dim(2,2)-LarCarte)/2;

if (X>0 & X<LongCarte & Y>0 & Y<LarCarte) % sur le PCB
    danspcb = danspcb + 1;
    Dist = -0.5;
    % pour les X
    PixelX=0;
    PixelY=0;
    for wafer = 1:3,
        for pad = 1:6,
            Dist = Dist + 1;
            if (abs(Dist-X)<0.5)
                PixelX = 6*(wafer-1)+pad;
            end
        end
        Dist = Dist + 0.2;
    end
    % Pour les Y
    Dist = -0.5;
    for wafer = 1:2,
        for pad = 1:6,
            Dist = Dist + 1;

            if (abs(Dist-Y)<0.5)
                PixelY = 6*(wafer-1)+pad;
            end
        end
        Dist = Dist + 0.2;
    end
    if (PixelX ~= 0 & PixelY ~= 0)
        pcb(PixelX,PixelY) = pcb(PixelX,PixelY) + 1;
    end
end

% QUANTIFICATION PAR PRISE EN COMPTE DES VOIES
% coordonnées de la détection en haut
X1= largeurPM*round((x1+0.5)/largeurPM); % localisation par rapport à la voie
Y1= largeurPM*round((y1+0.5)/largeurPM); % ...
% coordonnées de la détection en bas
X2= largeurPM*round((x1+dxB+0.5)/largeurPM); % ...
Y2= largeurPM*round((y1+dyB+0.5)/largeurPM); % ...
% ecart entre les 2 coordonnées
dXB= X1-X2; % delta X
dYB= Y1-Y2; % delta Y

% Coordonnees sur plan détection (origine hodo)
XD = X1+dXB*rap;
YD = Y1+dYB*rap;
```

```
% Coordonnees sur plan detection (origine PCB)
Xm=XD - (dim(2,1)-LongCarte)/2;
Ym=YD - (dim(2,2)-LarCarte)/2;

% detections pouvant toucher le PCB
if (Xm>-largeurPM/2 & Xm<LongCarte+largeurPM/2 & Ym>-largeurPM/2 & Ym<LarCarte+largeurPM/2 )
    danspcbm = danspcbm + 1;
    % pour les X
    Dist = -0.5;
    PixelX=0;
    PixelY=0;
    for wafer = 1:3,
        for pad = 1:6,
            Dist = Dist + 1;
            if (abs(Dist-Xm)<0.5)
                PixelX = 6*(wafer-1)+pad;
            end
            Dist = Dist + 0.2;
        end
        % Pour les Y
        Dist = -0.5;
        for wafer = 1:2,
            for pad = 1:6,
                Dist = Dist + 1;

                if (abs(Dist-Ym)<0.5 )
                    PixelY = 6*(wafer-1)+pad;
                end
            end
            Dist = Dist + 0.2;
        end
        if (PixelX ~= 0 & PixelY ~= 0)
            pcbm(PixelX,PixelY) = pcbm(PixelX,PixelY) + 1;
        end
    end

% Repartition sur plan Detection en pixel PM
if and(and(XD>0,YD>0),and(XD<44,YD<44))
    repartmq(round(0.5+XD/Pas),round(0.5+YD/Pas))=...
    repartmq(round(0.5+XD/Pas),round(0.5+YD/Pas))+1;
end

end
end

for i=1:3,
    for j=1:2,
        matric(i,j)=sum(sum(pcb(1+(i-1)*6:i*6,1+(j-1)*6:j*6)));
        matric1(i,j)=sum(sum(pcbm(1+(i-1)*6:i*6,1+(j-1)*6:j*6)));
    end
end

%*****
% resultats
%*****

NbreCoups
coincidence
danspcb
pcb
sum(sum(pcb))
matric

sprintf('ETAPE DE QUANTIFICATION PAR PM');
    danspcbm
    pcbm
    sum(sum(pcbm))
    matric1

figure,
subplot(2,1,1)
surfl(pcb)
```

```
title('détectations sur le PCB');
subplot(2,1,2)
surfl(pcbm)
title('détectations sur le PCB avec l''effet des voies');

subplot(2,1,1)
surfl(repartm);
xlabel('plan X');
ylabel('plan Y');
title('Répartition au niveau du PCB sans quantification')
set(gca,'xtick',[1:1:16]);
set(gca,'ytick',[1:1:16]);
subplot(2,1,2)
surfl(repartmq);
xlabel('plan X');
ylabel('plan Y');
title('Répartition au niveau du PCB avec quantification')
set(gca,'xtick',[1:1:16]);
set(gca,'ytick',[1:1:16]);

figure,
surf(repartm);
xlabel('plan X');
ylabel('plan Y');
title(sprintf('Répartitions sur le plan central en %d heures de simulation',heure))
set(gca,'xtick',[1:2:16]);
set(gca,'ytick',[1:2:16]);

FinProgramme
```

```
%*****V*Chevalier**2004**  
%* FinProgramme.m *  
%*-----*  
%* Fonction envoyant un message à un pc distant *  
%* Le mom du Pc et le message peuvent etre modifiés *  
%*****
```

```
NomPC='polntdl';  
Message='Le programme a fini de tourner';  
fprintf('\n');  
eval(sprintf('dos('net send %s %s');',NomPC,Message));  
  
fprintf('Fin du programme\n');
```

```
function NomFichier = SelfFichier(ext)

%*****V*Chevalier***2004***
%* Fonction qui sélectionne un fichier          *
%* Dans le répertoire courant                  *
%* Avec une extension entrée comme paramètre   *
%*****

NomFichier='';
while (exist(strcat(NomFichier, '.', ext))~=2)
    fprintf('\n Fichiers disponibles dans le répertoire courant: \n');
    eval(strcat('dir *.', ext));
    NomFichier =input(sprintf('fichier .%s à ouvrir ouvrir : ', ext), 's');
    clc
end
break
```

```
%*****V*Chevalier***2004***
%*          Verif_Soft.m          *
%*-----*
%* Lecture d'un fichier du type Slab#0 xx_xx.hod          *
%* Le fichier devra etre placé dans le répertoire courant *
%* Permet de vérifier la cohérence dans les calculs d'angles et de positions *
%*****

%*****
% initialisation
%*****

clear all
close all
clc
help Verif_Soft

%*****
% Paramètres du banc
%*****
Ecart_h_m=[75,75];      %      [81.3, 77.1];      %[x y] (cm)
Ecart_h_b=[135,135];    %      [139.2, 130.8];    %[x y] (cm)
largeurPM=2.75;         %      (cm)
NbrePM=16;

%*****
% Programme
%*****

NomFichier=SelFichier('hod');

types=(' %d %f %f %f %f');
%-- donnees='[  Numero, temps, posx, posy, angle';
for i=1:4 ,      % 4 plans
    for j=1:16,   % 16 PM
        types=strcat(types, '%1d');
%--      if mod(i,2)
%--          donnees=strcat(donnees,sprintf(' %d(:,%d) ', (i==1)+2*(i==3),j));
%--      else
%--          donnees=strcat(donnees,sprintf(' %d(:,%d) ', (i==2)+2*(i==4),j));
%--      end
    end
end

%-- commande=strcat(donnees,' ]=textread(strcat(NomFichier, '.hod'),types,'headerlines',13, 'commentstyl
e','c++');');
fprintf('\nLecture du fichier\n')
%-- eval(commande);

[  Numero, temps, posx, posy, angle ...
    x1(:,1),x1(:,2),x1(:,3),x1(:,4),x1(:,5),x1(:,6),x1(:,7),x1(:,8),x1(:,9),x1(:,10),x1(:,11),x1(:,12),x1(:,
13),x1(:,14),x1(:,15),x1(:,16),...
    y1(:,1),y1(:,2),y1(:,3),y1(:,4),y1(:,5),y1(:,6),y1(:,7),y1(:,8),y1(:,9),y1(:,10),y1(:,11),y1(:,12),y1(:,
13),y1(:,14),y1(:,15),y1(:,16),...
    x2(:,1),x2(:,2),x2(:,3),x2(:,4),x2(:,5),x2(:,6),x2(:,7),x2(:,8),x2(:,9),x2(:,10),x2(:,11),x2(:,12),x2(:,
13),x2(:,14),x2(:,15),x2(:,16),...
    y2(:,1),y2(:,2),y2(:,3),y2(:,4),y2(:,5),y2(:,6),y2(:,7),y2(:,8),y2(:,9),y2(:,10),y2(:,11),y2(:,12),y2(:,
13),y2(:,14),y2(:,15),y2(:,16),...
]=textread(strcat(NomFichier, '.hod'),types,'headerlines',13, 'commentstyle','c++');

%*****
% creation et initialisation des matrices
%*****

NbLignes=size(temps,1)
XHAUT=zeros(1,NbLignes);
YHAUT=zeros(1,NbLignes);
XBAS=zeros(1,NbLignes);
YBAS=zeros(1,NbLignes);
Localisation=zeros(NbLignes,2,2);
LocCentre=zeros(NbLignes,2);
angleBIS=zeros(1,NbLignes);
angles=zeros(1,NbLignes);
```

```

%*****
% traitement
%*****

% calcul des évènement bons
NbEventOk=0;
DetectSimple=0;
EventOk=zeros(1,NbLignes);
corel=zeros(1,NbLignes);
for i=1:NbLignes %parcours de l'ensemble des résultats
    valid=0;

    if(mod(i,1000)==0)
        clc
        sprintf('ligne %d sur %d',i,NbLignes)
        % affichage pour visualiser que le programme tourne
    end

    if posx(i)>0 & posy(i)>0 %on n'a pas -1 dans les coordonnées de simon
        if (sum(x1(i,:))+sum(x2(i,:))+sum(y1(i,:))+sum(y2(i,:)))==4 %si on a un evènement ok
            DetectSimple=DetectSimple+1;
        end
        if (sum(x1(i,:))<=2 & (sum(x2(i,:))<=2 & (sum(y1(i,:))<=2 & (sum(y2(i,:))<=2) % max de 2 voies actives ✓
            par plan
                x1val=0;
                x2val=0;
                y1val=0;
                y2val=0;

                for m=1:(NbrePM),
                    if (x1(i,m)==1) % 1 PM à 1
                        if sum(x1(i,:))==1 % 1voie
                            x1val=m;
                        else % 2voies
                            if m~=16
                                if (x1(i,m+1)==1) %PM suivant à 1
                                    x1val=m+0.5;
                                end
                            end
                        end
                    end

                    if (x2(i,m)==1) % 1 PM à 1
                        if sum(x2(i,:))==1 % 1voie
                            x2val=m;
                        else % 2voies
                            if m~=16
                                if x2(i,m+1)==1 %PM suivant à 1
                                    x2val=m+0.5;
                                end
                            end
                        end
                    end
                end

                if (y1(i,m)==1) % 1 PM à 1
                    if sum(y1(i,:))==1 % 1voie
                        y1val=m;
                    else % 2voies
                        if m~=16
                            if y1(i,m+1)==1 %PM suivant à 1
                                y1val=m+0.5;
                            end
                        end
                    end
                end

                if (y2(i,m)==1) % 1 PM à 1
                    if sum(y2(i,:))==1 % 1voie
                        y2val=m;
                    else % 2voies
                        if m~=16
                            if y2(i,m+1)==1 %PM suivant à 1
                                y2val=m+0.5;
                            end
                        end
                    end
                end
            end
        end
    end
end
end

```

```

                                end
                        end
                end
        end
        if (x1val~=0) & (x2val~=0) & (y1val~=0) & (y2val~=0)
            NbEventOk=NbEventOk+1;
            numeroPM(NbEventOk,1,1)=x1val; % (n°, xy, hb)
            numeroPM(NbEventOk,2,1)=y1val; % (n°, xy, hb)
            numeroPM(NbEventOk,1,2)=x2val; % (n°, xy, hb)
            numeroPM(NbEventOk,2,2)=y2val; % (n°, xy, hb)
            valid=1;
        end
    end
    if valid==1
        %calcul des coordonnées en X et en Y
        for xy=1:2,                                %1 pour x et 2 pour y
            for hb=1:2,                                % 1 pour haut et 2 pour bas
                Localisation(NbEventOk,xy,hb)=largeurPM*numeroPM(NbEventOk,xy,hb)...
                    -largeurPM/2;
            end
            LocCentre(NbEventOk,xy)=Localisation(NbEventOk,xy,1)+ ...
                (Localisation(NbEventOk,xy,2)-Localisation(NbEventOk,xy,1))...
                *Ecart_h_m(xy)/Ecart_h_b(xy);
        end

        XHAUT(NbEventOk)=Localisation(NbEventOk,1,1);
        YHAUT(NbEventOk)=Localisation(NbEventOk,2,1);
        XBAS(NbEventOk)=Localisation(NbEventOk,1,2);
        YBAS(NbEventOk)=Localisation(NbEventOk,2,2);

        Posx(NbEventOk)=posx(i);
        Posy(NbEventOk)=posy(i);

                                %identification de l'angle
        angleBIS(NbEventOk)=angle(i);
        angles(NbEventOk)=atan(sqrt(...
            (Localisation(NbEventOk,1,2)-Localisation(NbEventOk,1,1))^2+...
            (Localisation(NbEventOk,2,2)-Localisation(NbEventOk,2,1))^2 ...
            )/mean(Ecart_h_b))*180/pi;

        Valid=0 ;
    end
end
corel(i)=NbEventOk; % permet de trouver la corrélation entre les numéros et les lignes
end

%*****
% affichage des résultats
%*****

fprintf('Il y a eu %d détections valides sur %d détections, soit %2.2f %%\n',...
        NbEventOk, NbLignes, 100*NbEventOk/NbLignes);

fprintf('Parmis ces détections %d sont simples, soit %2.2f %% des détections valides\n',...
        DetectSimple, 100*DetectSimple/NbEventOk);

fprintf('Et %d détections sont multiples, soit %2.2f %% des détections valides\n',...
        NbEventOk-DetectSimple, 100*(NbEventOk-DetectSimple)/NbEventOk);

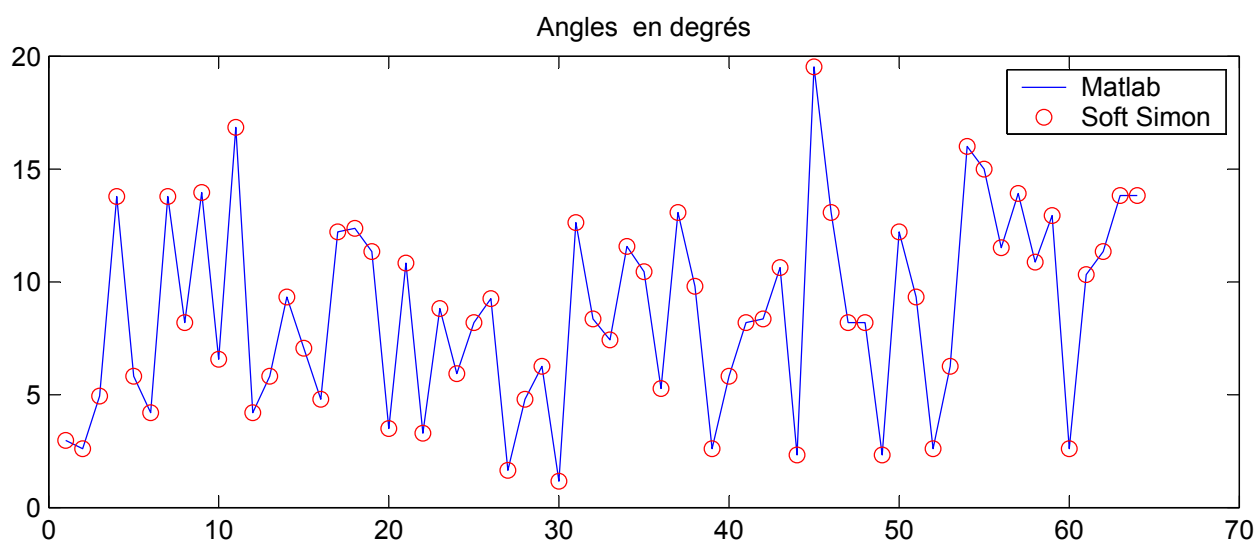
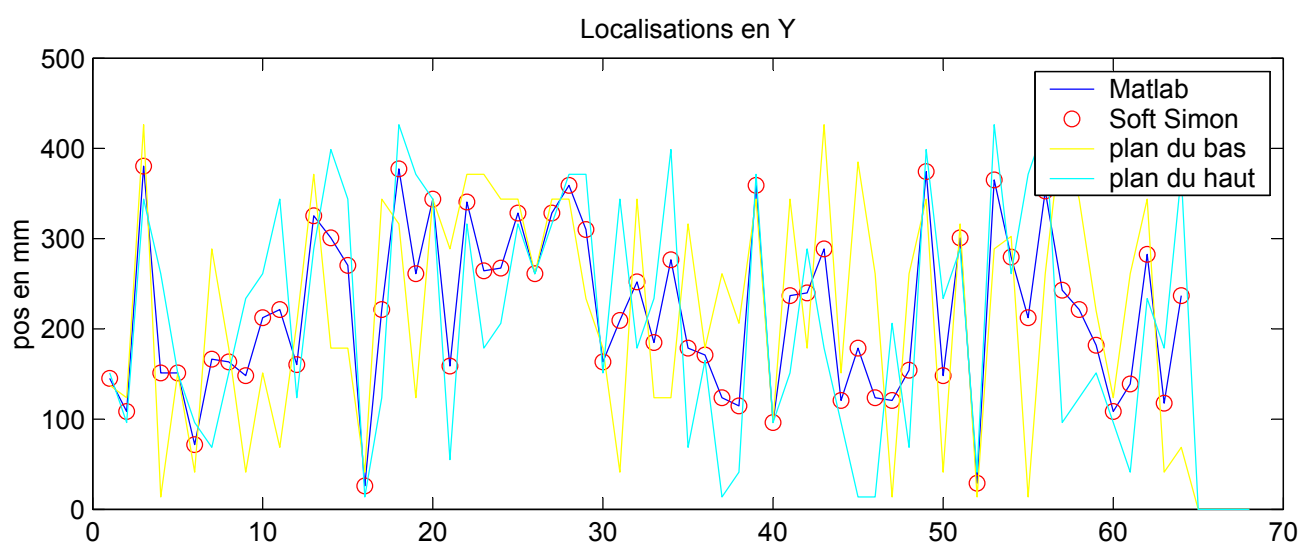
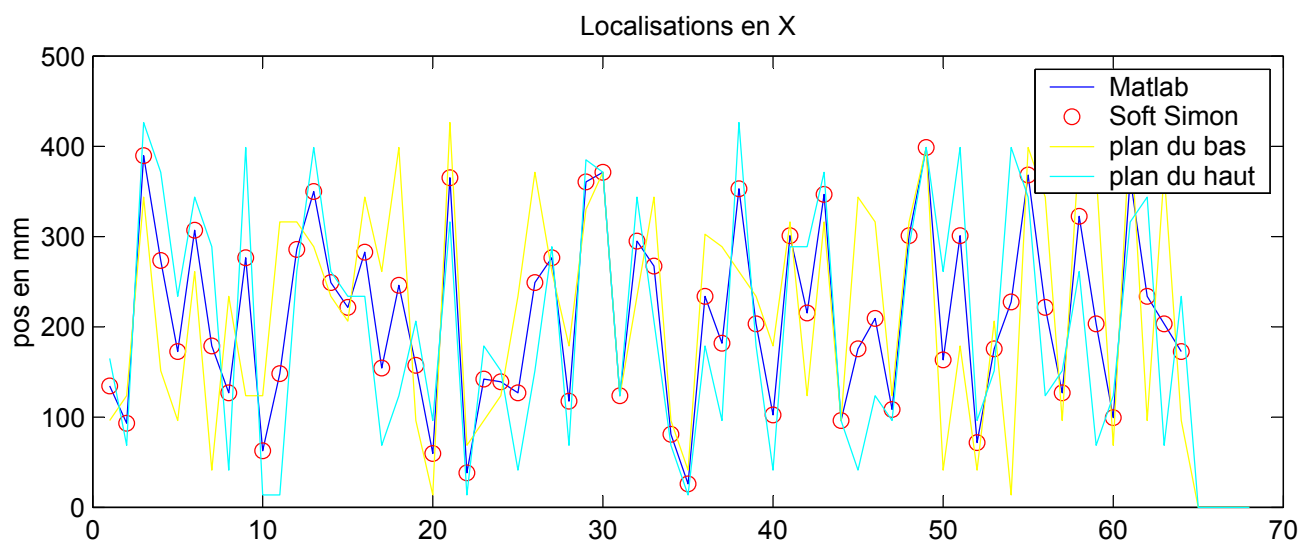
figure
subplot(3,1,1)
plot( LocCentre((1:NbEventOk),1)*10)
hold on
plot( Posx, 'ro')
plot(XHAUT*10,'y')
plot(XBAS*10,'c')
title 'Localisations en X'
ylabel 'pos en mm'
legend('Matlab','Soft Simon','plan du bas','plan du haut');

subplot(3,1,2)
plot( LocCentre((1:NbEventOk),2)*10)
hold on
plot( Posy, 'ro')
plot(YHAUT*10,'y')
plot(YBAS*10,'c')
```

```
title 'Localisations en Y'
ylabel 'pos en mm'
legend('Matlab', 'Soft Simon', 'plan du bas', 'plan du haut');

subplot(3,1,3)
plot(angles(1:NbEventOk)')
hold on
plot(angleBIS(1:NbEventOk) '*180/pi, 'ro')
title 'Angles en degrés'
legend('Matlab', 'Soft Simon');

finprogramme
```



[illegible]

```
for i=1:NbLignes %parcours de l'ensemble des résultats
    if(mod(i,1000)==0)
        clc
        sprintf('ligne %d sur %d',i,NbLignes)
    end

    valid=0;
    if (sum(x1(i,:))+sum(x2(i,:))+sum(y1(i,:))+sum(y2(i,:)))==4 %si on a un evènement pouvant etre
valide %max de 2 voies actives
        DetectSimple=DetectSimple+1;
    end
    if (sum(x1(i,:))<=2 & (sum(x2(i,:))<=2 & (sum(y1(i,:))<=2 & (sum(y2(i,:))<=2) % max de 2 voies actives
par plan
        x1val=0;
        x2val=0;
        y1val=0;
        y2val=0;

        for m=1:(NbrePM),
            if (x1(i,m)==1) % 1 PM à 1
                if sum(x1(i,:))==1 % 1voie
                    x1val=m;
                else % 2voies
                    if m~=16
                        if (x1(i,m+1)==1) %PM suivant à 1
                            x1val=m+0.5;
                        end
                    end
                end
            end

            if (x2(i,m)==1) % 1 PM à 1
                if sum(x2(i,:))==1 % 1voie
                    x2val=m;
                else % 2voies
                    if m~=16
                        if x2(i,m+1)==1 %PM suivant à 1
                            x2val=m+0.5;
                        end
                    end
                end
            end

            if (y1(i,m)==1) % 1 PM à 1
                if sum(y1(i,:))==1 % 1voie
                    y1val=m;
                else % 2voies
                    if m~=16
                        if y1(i,m+1)==1 %PM suivant à 1
                            y1val=m+0.5;
                        end
                    end
                end
            end

            if (y2(i,m)==1) % 1 PM à 1
                if sum(y2(i,:))==1 % 1voie
                    y2val=m;
                else % 2voies
                    if m~=16
                        if y2(i,m+1)==1 %PM suivant à 1
                            y2val=m+0.5;
                        end
                    end
                end
            end

            if (x1val~=0) & (x2val~=0) & (y1val~=0) & (y2val~=0)
                NbEventOk=NbEventOk+1;
                numeroPM(NbEventOk,1,1)=x1val; % (n°,xy,hb)
                numeroPM(NbEventOk,2,1)=y1val; % (n°,xy,hb)
                numeroPM(NbEventOk,1,2)=x2val; % (n°,xy,hb)
                numeroPM(NbEventOk,2,2)=y2val; % (n°,xy,hb)
                valid=1;
            end
        end
    end
end
```

```
end
end
if valid==1
    %calcul des coordonnées en X et en Y
    for xy=1:2,
        %1 pour x et 2 pour y
        for hb=1:2,
            % 1 pour haut et 2 pour bas
            Localisation(NbEventOk,xy,hb)=largeurPM*numeroPM(NbEventOk,xy,hb)...
                -largeurPM/2;
        end
        LocCentre(NbEventOk,xy)=Localisation(NbEventOk,xy,1)+ ...
            (Localisation(NbEventOk,xy,2)-Localisation(NbEventOk,xy,1))...
            *Ecart_h_m(xy)/Ecart_h_b(xy);
    end
    %identification de l'angle
    angleBIS(NbEventOk)=angle(i);
    Angle(NbEventOk)=atan(sqrt(...
        (Localisation(NbEventOk,1,2)-Localisation(NbEventOk,1,1))^2+...
        (Localisation(NbEventOk,2,2)-Localisation(NbEventOk,2,1))^2 ...
    )/mean(Ecart_h_b))*180/pi;
    Valid=0 ;
end
if NbEventOk>0
    Temps(NbEventOk)=Temps(NbEventOk)+temps(i);
end
end

LocCentre(:,3)=Angle'; % LocCentre([X Y Angle])

fprintf('Il y a eu %d évènements OK sur %d détections soit %2.2f %%',...
        NbEventOk, NbLignes, 100*NbEventOk/NbLignes)
fprintf('Parmis ces détections %d sont multiples soit %2.2f %%',...
        NbEventOk-DetectSimple, 100*(NbEventOk-DetectSimple)/NbEventOk)

donnee=zeros(NbEventOk,5);
% FICHER DU TYPE HODO FUTUR
for i=1:NbEventOk,
    donnee(i,1)=Temps(i);
    donnee(i,2:4)=LocCentre(i,1:3);
end

figure,
subplot(2,1,1)
hist(LocCentre(:,1),44);
title 'répartition des détections en x';
subplot(2,1,2)
hist(LocCentre(:,2),44);
title 'répartition des détections en x';

fprintf('écriture du fichier %s._Hodo',NomFichier);
dlmwrite(strcat(NomFichier,'_Hodo'),donnee,'\t');
FinProgramme
```

```
% PCB_hodo.m
% nécessite les fichiers (xx.dat)
%               (xx.ACQ_PCBA-1)
% et            (xx.ACQ_PCBA-2)
% avec a variable prenant la valeur 1 ou 2

clear all
close all
clc
help PCB_hodo

%NomFichier='WeekEnd';
%NomFichier='5 mai soir cut';

NomFichier='';

% while (exist(strcat(NomFichier,'.dat'))~=2) | or(...
%       (exist(strcat(NomFichier,'.ACQ_PCB1-1'))~=2) | (exist(strcat(NomFichier,'.ACQ_PCB1-2'))~=2), ...
%       (exist(strcat(NomFichier,'.ACQ_PCB2-1'))~=2) | (exist(strcat(NomFichier,'.ACQ_PCB2-2'))~=2))
%   dir *.dat;
%   dir *.ACQ*;
%
%   NomFichier =input('fichier .dat à ouvrir : ','s');
%   clc
% end

%sans .dat
sansdat=0;
while and(...
    (exist(strcat(NomFichier,'.ACQ_PCB1-1'))~=2) | (exist(strcat(NomFichier,'.ACQ_PCB1-2'))~=2), ...
    (exist(strcat(NomFichier,'.ACQ_PCB2-1'))~=2) | (exist(strcat(NomFichier,'.ACQ_PCB2-2'))~=2))
    dir *.ACQ*;
    NomFichier =input('fichier .ACQ_PCBx-1 et .ACQ_PCBx-2 à ouvrir : ','s');
    clc
    sansdat=1;
end

button = questdlg('Quel PCB analyser ','Choix PCB','1','2','default')
switch button
    case '1'
        a=button;
    case '2'
        a=button;
    otherwise
        sprintf('Erreur dans la saisie du pcb')
        break
end

button = questdlg('Quel wafer analyser ',strcat('PCB',a),'1','6','default')
switch button
    case '1'
        wafersel=1;
    case '6'
        wafersel=6;
    otherwise
        sprintf('Erreur dans la saisie du wafer')
        break
end

%LocReseau='\\CosmicBench\VTC2\'
%LocReseau='';

if(sansdat~=1)
Ecart_h_m=[75,75]; % [81.3, 77.1]; % [x y]
Ecart_h_b=[135,135]; % [139.2, 130.8]; % [x y]
largeurPM=2.75;

NbrePM=8;

% lecture du fichier
```

```
types=('%f %s');  
for i=1:4 ,      % 4 plans  
    for j=1:16,   % 16 PM  
        types=strcat(types,'%ld');  
    end  
end  
% type donnera:      '%f %s   %ld%d%d%d%d%d%d%d%d%d%d%d%d%d%d' %ld%d%d%d%d%d%d%d%d%d%d%d%d%d'd  
                        %ld%d%d%d%d%d%d%d%d%d%d%d%d%d'd %ld%d%d%d%d%d%d%d%d%d%d%d%d'd  
                        %ld%d%d%d%d%d%d%d%d%d%d%d%d'd %ld%d%d%d%d%d%d%d%d%d%d%d'd'  
  
[temps, EventOk,...  
x1(:,1),x1(:,2),x1(:,3),x1(:,4),x1(:,5),x1(:,6),x1(:,7),x1(:,8),...  
x1(:,9),x1(:,10),x1(:,11),x1(:,12),x1(:,13),x1(:,14),x1(:,15),x1(:,16),...  
y1(:,1),y1(:,2),y1(:,3),y1(:,4),y1(:,5),y1(:,6),y1(:,7),y1(:,8),...  
y1(:,9),y1(:,10),y1(:,11),y1(:,12),y1(:,13),y1(:,14),y1(:,15),y1(:,16),...  
x2(:,1),x2(:,2),x2(:,3),x2(:,4),x2(:,5),x2(:,6),x2(:,7),x2(:,8),...  
x2(:,9),x2(:,10),x2(:,11),x2(:,12),x2(:,13),x2(:,14),x2(:,15),x2(:,16),...  
y2(:,1),y2(:,2),y2(:,3),y2(:,4),y2(:,5),y2(:,6),y2(:,7),y2(:,8),...  
y2(:,9),y2(:,10),y2(:,11),y2(:,12),y2(:,13),y2(:,14),y2(:,15),y2(:,16)...  
]=textread(strcat(NomFichier,'.dat'),types,'headerlines',8);  
  
NbLignes=size(temps,1)  
tempstotal=sum(sum(temps))  
temps_simu=tempstotal;  
  
numeroPM=zeros(NbLignes,2,2);  
LocCentre=zeros(NbLignes,4);  
Localisation=zeros(NbLignes,2,2);  
Angle=zeros(1,NbLignes);  
Temps=zeros(1,NbLignes);  
  
%break;  
% calcul des évènements bons  
NbEventOk=0;  
OkDouble=0;  
  
for i=1:NbLignes     %parcours de l'ensemble des résultats  
    if(mod(i,1000)==0)  
        clc  
        sprintf('ligne %d sur %d',i,NbLignes)  
    end  
  
    valid=0;  
    if strcmp(cellstr(EventOk(i)), 'OK')==1           %si on a un evènement ok  
        NbEventOk =NbEventOk+1;                       % comptage évènement ok  
        valid=1;  
                                                %identification des PM touchés  
        for k=1:NbrePM,  
            if x1(i,k)==1  
                numeroPM(NbEventOk,1,1)=k;          %(n°,xy,hb)  
            end  
            if x2(i,k)==1  
                numeroPM(NbEventOk,1,2)=k;          %(n°,xy,hb)  
            end  
            if y1(i,k)==1  
                numeroPM(NbEventOk,2,1)=k;          %(n°,xy,hb)  
            end  
            if y2(i,k)==1  
                numeroPM(NbEventOk,2,2)=k;          %(n°,xy,hb)  
            end  
        end  
    else  
        % cas 'NOK'  
        if (sum(x1(i,:))<=2) & (sum(x2(i,:))<=2) & (sum(y1(i,:))<=2) & (sum(y2(i,:))<=2) %2 max par PM  
            x1val=0;  
            x2val=0;  
            y1val=0;  
            y2val=0;  
  
            for m=(NbrePM-1),  
                if (x1(i,m)==1)                               % 1 PM à 1  
                    if sum(x1(i,:))==1                         % 1voie  
                        numeroPM(NbEventOk+1,1,1)=m;         %(n°,xy,hb)
```

```
x1val=1;
else % 2voies
    if x1(i,m+1)==1 %PM suivant à 1
        numeroPM(NbEventOk+1,1,1)=m+0.5;
        x1val=1;
    end
end
end

if (x2(i,m)==1) % 1 PM à 1
    if sum(x2(i,:))==1 % 1voie
        numeroPM(NbEventOk+1,2,1)=m; % (n°,xy,hb)
        x2val=1;
    else % 2voies
        if x2(i,m+1)==1 %PM suivant à 1
            numeroPM(NbEventOk+1,2,1)=m+0.5;
            x2val=1;
        end
    end
end
end

if (y1(i,m)==1) % 1 PM à 1
    if sum(y1(i,:))==1 % 1voie
        numeroPM(NbEventOk+1,1,2)=m; % (n°,xy,hb)
        y1val=1;
    else % 2voies
        if y1(i,m+1)==1 %PM suivant à 1
            numeroPM(NbEventOk+1,1,2)=m+0.5;
            y1val=1;
        end
    end
end
end

if (y2(i,m)==1) % 1 PM à 1
    if sum(y2(i,:))==1 % 1voie
        numeroPM(NbEventOk+1,2,2)=m; % (n°,xy,hb)
        y2val=1;
    else % 2voies
        if y2(i,m+1)==1 %PM suivant à 1
            numeroPM(NbEventOk+1,2,2)=m+0.5;
            y2val=1;
        end
    end
end
end

if (x1val==1) & (x2val==1) & (y1val==1) & (y2val==1)
    OkDouble=OkDouble+1;
    NbEventOk=NbEventOk+1;
    valid=1;
    LocCentre(NbEventOk,4)=i;
end
end
end

if valid==1

    %calcul des coordonnées en X et en Y
    for xy=1:2, %1 pour x et 2 pour y
        for hb=1:2, % 1 pour haut et 2 pour bas
            Localisation(NbEventOk,xy,hb)=largeurPM*numeroPM(NbEventOk,xy,hb)...
                -largeurPM/2;
        end
        LocCentre(NbEventOk,xy)=Localisation(NbEventOk,xy,1)+ ...
            (Localisation(NbEventOk,xy,2)-Localisation(NbEventOk,xy,1))...
            *Ecart_h_m(xy)/Ecart_h_b(xy);
    end

    %identification de l'angle
    Angle(NbEventOk)=atan(sqrt(...
        (Localisation(NbEventOk,1,2)-Localisation(NbEventOk,1,1))^2+...
        (Localisation(NbEventOk,2,2)-Localisation(NbEventOk,2,1))^2 ...
    )/mean(Ecart_h_b))*180/pi;
```

```
Valid=0;
end
if NbEventOk>0
    Temps(NbEventOk)=Temps(NbEventOk)+temps(i);
end
end

LocCentre(:,3)=Angle'; % LocCentre([X Y Angle])

sprintf('Il y a eu %d évènements OK sur %d détections soit %2.2f %%',...
        NbEventOk, NbLignes, 100*NbEventOk/NbLignes)
sprintf('Parmis ces détections %d sont multiples soit %2.2f %%',...
        OkDouble, 100*OkDouble/NbLignes)

% LECTURE DES DONNEES PCB

% répartition des détections

Pas=largeurPM;
Nbbin=round(44/Pas);
repart=zeros(Nbbin);
for i=1:NbEventOk,
    repart(round(0.5+LocCentre(i,1)/Pas),round(0.5+LocCentre(i,2)/Pas))=...
        repart(round(0.5+LocCentre(i,1)/Pas),round(0.5+LocCentre(i,2)/Pas))+1;
end

end

types=('d');
for i=1:6, % 6 VFE
    for j=1:18, % 18 pads par VFE
        types=strcat(types,'%f');
    end
end

i=1;

[ Numero,...
  vfe(:,0+i,1),vfe(:,0+i,2),vfe(:,0+i,3),vfe(:,0+i,4),vfe(:,0+i,5),vfe(:,0+i,6),...
    vfe(:,0+i,7),vfe(:,0+i,8),vfe(:,0+i,9),vfe(:,0+i,10),vfe(:,0+i,11),vfe(:,0+i,12),...
    vfe(:,0+i,13),vfe(:,0+i,14),vfe(:,0+i,15),vfe(:,0+i,16),vfe(:,0+i,17),vfe(:,0+i,18),...
  vfe(:,1+i,1),vfe(:,1+i,2),vfe(:,1+i,3),vfe(:,1+i,4),vfe(:,1+i,5),vfe(:,1+i,6),...
    vfe(:,1+i,7),vfe(:,1+i,8),vfe(:,1+i,9),vfe(:,1+i,10),vfe(:,1+i,11),vfe(:,1+i,12),...
    vfe(:,1+i,13),vfe(:,1+i,14),vfe(:,1+i,15),vfe(:,1+i,16),vfe(:,1+i,17),vfe(:,1+i,18),...
  vfe(:,2+i,1),vfe(:,2+i,2),vfe(:,2+i,3),vfe(:,2+i,4),vfe(:,2+i,5),vfe(:,2+i,6),...
    vfe(:,2+i,7),vfe(:,2+i,8),vfe(:,2+i,9),vfe(:,2+i,10),vfe(:,2+i,11),vfe(:,2+i,12),...
    vfe(:,2+i,13),vfe(:,2+i,14),vfe(:,2+i,15),vfe(:,2+i,16),vfe(:,2+i,17),vfe(:,2+i,18),...
  vfe(:,3+i,1),vfe(:,3+i,2),vfe(:,3+i,3),vfe(:,3+i,4),vfe(:,3+i,5),vfe(:,3+i,6),...
    vfe(:,3+i,7),vfe(:,3+i,8),vfe(:,3+i,9),vfe(:,3+i,10),vfe(:,3+i,11),vfe(:,3+i,12),...
    vfe(:,3+i,13),vfe(:,3+i,14),vfe(:,3+i,15),vfe(:,3+i,16),vfe(:,3+i,17),vfe(:,3+i,18),...
  vfe(:,4+i,1),vfe(:,4+i,2),vfe(:,4+i,3),vfe(:,4+i,4),vfe(:,4+i,5),vfe(:,4+i,6),...
    vfe(:,4+i,7),vfe(:,4+i,8),vfe(:,4+i,9),vfe(:,4+i,10),vfe(:,4+i,11),vfe(:,4+i,12),...
    vfe(:,4+i,13),vfe(:,4+i,14),vfe(:,4+i,15),vfe(:,4+i,16),vfe(:,4+i,17),vfe(:,4+i,18),...
  vfe(:,5+i,1),vfe(:,5+i,2),vfe(:,5+i,3),vfe(:,5+i,4),vfe(:,5+i,5),vfe(:,5+i,6),...
    vfe(:,5+i,7),vfe(:,5+i,8),vfe(:,5+i,9),vfe(:,5+i,10),vfe(:,5+i,11),vfe(:,5+i,12),...
    vfe(:,5+i,13),vfe(:,5+i,14),vfe(:,5+i,15),vfe(:,5+i,16),vfe(:,5+i,17),vfe(:,5+i,18),...
]=textread(strcat(NomFichier,'.acq_pcb',a,'-1'),types,'headerlines',10);

sprintf('pcb 1 lu')
[ Numerobis,...
  vfebis(:,0+i,1),vfebis(:,0+i,2),vfebis(:,0+i,3),vfebis(:,0+i,4),vfebis(:,0+i,5),vfebis(:,0+i,6),...
    vfebis(:,0+i,7),vfebis(:,0+i,8),vfebis(:,0+i,9),vfebis(:,0+i,10),vfebis(:,0+i,11),vfebis(:,0+i,12),.
  ..
    vfebis(:,0+i,13),vfebis(:,0+i,14),vfebis(:,0+i,15),vfebis(:,0+i,16),vfebis(:,0+i,17),vfebis(:,0+i,18
),...
  vfebis(:,1+i,1),vfebis(:,1+i,2),vfebis(:,1+i,3),vfebis(:,1+i,4),vfebis(:,1+i,5),vfebis(:,1+i,6),...
    vfebis(:,1+i,7),vfebis(:,1+i,8),vfebis(:,1+i,9),vfebis(:,1+i,10),vfebis(:,1+i,11),vfebis(:,1+i,12),.
  ..
    vfebis(:,1+i,13),vfebis(:,1+i,14),vfebis(:,1+i,15),vfebis(:,1+i,16),vfebis(:,1+i,17),vfebis(:,1+i,18
),...
  vfebis(:,2+i,1),vfebis(:,2+i,2),vfebis(:,2+i,3),vfebis(:,2+i,4),vfebis(:,2+i,5),vfebis(:,2+i,6),...
    vfebis(:,2+i,7),vfebis(:,2+i,8),vfebis(:,2+i,9),vfebis(:,2+i,10),vfebis(:,2+i,11),vfebis(:,2+i,12),.
  ..
    vfebis(:,2+i,13),vfebis(:,2+i,14),vfebis(:,2+i,15),vfebis(:,2+i,16),vfebis(:,2+i,17),vfebis(:,2+i,18
]
```

```
), ...
    vfebis(:,3+i,1),vfebis(:,3+i,2),vfebis(:,3+i,3),vfebis(:,3+i,4),vfebis(:,3+i,5),vfebis(:,3+i,6),...
    vfebis(:,3+i,7),vfebis(:,3+i,8),vfebis(:,3+i,9),vfebis(:,3+i,10),vfebis(:,3+i,11),vfebis(:,3+i,12),.
..
    vfebis(:,3+i,13),vfebis(:,3+i,14),vfebis(:,3+i,15),vfebis(:,3+i,16),vfebis(:,3+i,17),vfebis(:,3+i,18)
), ...
    vfebis(:,4+i,1),vfebis(:,4+i,2),vfebis(:,4+i,3),vfebis(:,4+i,4),vfebis(:,4+i,5),vfebis(:,4+i,6),...
    vfebis(:,4+i,7),vfebis(:,4+i,8),vfebis(:,4+i,9),vfebis(:,4+i,10),vfebis(:,4+i,11),vfebis(:,4+i,12),.
..
    vfebis(:,4+i,13),vfebis(:,4+i,14),vfebis(:,4+i,15),vfebis(:,4+i,16),vfebis(:,4+i,17),vfebis(:,4+i,18)
), ...
    vfebis(:,5+i,1),vfebis(:,5+i,2),vfebis(:,5+i,3),vfebis(:,5+i,4),vfebis(:,5+i,5),vfebis(:,5+i,6),...
    vfebis(:,5+i,7),vfebis(:,5+i,8),vfebis(:,5+i,9),vfebis(:,5+i,10),vfebis(:,5+i,11),vfebis(:,5+i,12),.
..
    vfebis(:,5+i,13),vfebis(:,5+i,14),vfebis(:,5+i,15),vfebis(:,5+i,16),vfebis(:,5+i,17),vfebis(:,5+i,18)
), ...
]=textread(strcat(NomFichier,'.acq_pcb',a,'-2'),types,'headerlines',10);
sprintf('pcb 2 lu')

Nblignes=min(length(Numerobis),length(Numero))
clear Numero
clear Numerobis
clear Temps
clear EventOk
clear Localisation
clear numeroPM
clear x1
clear x2
clear y1
clear y2
sprintf('rafraichissement de la mémoire')

Nblignes=round(Nblignes/2);

% rassemblement des données dans data
data= zeros(Nblignes,6*2*18);
for i=1:Nblignes, % nombre d'evenements
    for NumVFE=1:6,
        for pad=1:18,
            data(i,(NumVFE-1)*18+ pad)=vfe(i,NumVFE,pad);
        end
    end
    for NumVFE=7:12,
        for pad=1:18,
            data(i,(NumVFE-1)*18+ pad)=vfebis(i,NumVFE-6,pad);
        end
    end
end

% moyenne par fichier
moyVFE1_6=mean(mean(mean(vfe(:,1,:),2)))
moyVFE7_12=mean(mean(mean(vfebis(:,1,:),2)))
% clear vfe
% clear vfebis

% calcul des moyennes et des ecarts types pour chaque pad
Moyenne=zeros(1,12*18);
Moyenne=mean(data); %chaque pad est considéré (216 valeurs)
EcartType=zeros(1,12*18);
EcartType =std(data); % sur la zone analysée (216 valeurs)

for wafersel=1:6,

wafermin=wafersel; % inclut le wafer saisi
wafermax=wafersel; % inclut le wafer saisi
NbWAFER=wafermax-(wafermin-1);

figure(1);
subplot(2,2,1:2);
bar(1000*EcartType(1:12*18));
title (sprintf('%s PCB%s-1 VFE 1 à 12',NomFichier,a));
```

```
xlabel('Pads');
ylabel('Tension en mV');
axis([0 216 0 1]);

sprintf('Calcul de l''écart type du bruit')
%Calcul de l'écart type du bruit

DataBruit=zeros(Nblignes,36);
titi=0;
wafer=wafersel;
for i=1:Nblignes,
    %for wafer=1:6,
    mip=0;
    for pad=1:36;
        if data(i,pad+36*(wafer-1))-Moyenne(pad+36*(wafer-1))>2*EcartType(pad+36*(wafer-1)) %4*mean(Ecar
tTypeBruit( 36*(wafermin-1-5)+1:36*(wafermax-5))) % 4*EcartType(pad)
            mip=1; % mip détecté
        end
    end
    if mip==0 % pas de mip sur le wafer
        titi=titi+1;
        DataBruit(titi,:)=data(i,(wafer-1)*36+1:wafer*36);
    end
end
%end

MoyenneBruit=mean(DataBruit(1:titi,:));

sprintf('soustraction de la moyenne ')
%*****
% soustraction de la moyenne
for i=1:Nblignes,
    for pad=1:36;
        pc(i,pad)=data(i,pad+36*(wafer-1))-MoyenneBruit(pad);
    end
end

EcartTypeBruit =zeros(1,36);
EcartTypeBruit =std(pc);

%%%%%%%%%%%%%%ok

matrice1 = zeros(36*Nblignes,1);
matrice2 = zeros(36*Nblignes,1);
matrice3 = zeros(36*Nblignes,1);

sprintf('création de la matrice 2')
num=0;
tata=0;
toto=0;
for i=1:Nblignes,
    for pad=1:36, % 36 pads par wafer
        tata=tata+1;
        matrice1(tata)=pc(i,pad);
        if pc(i,pad)>4*EcartTypeBruit(pad)
            toto =toto+1;
            matrice2(toto)=pc(i,pad);
        end
    end
end
end

%figure(2);
subplot(2,2,3);
hist(matrice1(1:tata),1000);
title (sprintf('bruit + mip sur le wafer %d ',wafersel));
xlabel('Level (V)');
ylabel('Nbr Evt');
axis([-0.005 0.02 0 20]);

%figure(4);
```

```
subplot(2,2,4);
hist(matrice2(1:toto),1000);
title(sprintf('mip sur le wafer %d ',wafer));
xlabel('Level (V)');
ylabel('Nbr Evt');
axis([-0.005 0.02 0 20]);

% figure(3);
% %hist(matrice1,1000);
% stairs(hist(matrice1,100));
% axis ([-0.002 0.002 0 20000]);
% hold on;
% Esp = 0;
% Ect = 0.0002;
% X = -2:0.1:2;
% Y = 10000*sqrt(1/2/pi/Ect.^2)*exp(-(X-Esp).^2/(2*Ect.^2));
% plot(X,Y,'y');
% hold off;
end

% for i=1:min(size(Numerobis),size(Numero),size(temps,1)),
%     for j=0:5,
%         for k=1:18,
%             LocCentre(i,5+ j*18 + (k-1))=vfe(i,j+1,k);
%             LocCentre(i,5+ (j+6)*18 + (k-1))=vfebis(i,j+1,k);
%         end
%     end
% end

finprogramme
```

Vincent
CHEVALIER

Mise en œuvre d'un banc cosmique

Rapport de stage de fin d'études
DESS « Physique des capteurs et Instrumentation »
Université de Bretagne Occidentale UBO BREST

Résumé :

Ce stage au sein du laboratoire Louis Leprince Ringuet m'a permis de mettre en œuvre un hodoscope de rayons cosmiques à base de scintillateurs et de photomultiplicateurs pour la calibration d'un détecteur de particules ionisantes en silicium.

Je me suis tout d'abord attaché à câbler et étalonner les 64 voies de photomultiplicateurs couplés aux scintillateurs qui vont constituer le banc cosmique. Par la suite j'ai réalisé une modélisation physique du banc de test ainsi qu'une simulation « Monte-Carlo » sous Matlab.

Mots clés :

Physique des particules, Détecteur, Rayons cosmiques

Photomultiplicateur, Hodoscope, Banc de test

Modélisation, simulation Monte-Carlo

Keywords :

Particles physics, Detector, Cosmic rays

Photomultiplier, Hodoscope, Testbench

Modeling, Monte-Carlo simulation