



IPSA (Institut Polytechnique des Sciences Avancées)
64 Bvd de Brandebourg - 94200 Ivry-sur-Seine
40 Bvd de la Marquette - 31000 Toulouse



IN2P3
Les deux infinis

CPPM (Centre de Physique des Particules de
Marseille)
163 Avenue de Luminy, 13009 Marseille



Mémoire de Stage de Fin d'Études

En vue de l'obtention du diplôme

INGÉNIEUR IPSA

Filière : Espace, Lanceurs et Satellites

Option : Systèmes Embarqués

Mise en place du contrôle-commande et du monitoring d'un banc de test interférométrique pour la mission LISA de l'ESA

Organisme d'accueil : Centre de Physique des Particules de Marseille

Dates du stage : 03 Février 2020 – 31 Juillet 2020

Réalisé par :

M. Alexandre CONDETTE
Aéro 5 ELSS
Promotion 2020

Sous la direction de :

Mme. Aurélia SECROUN (CPPM)
M. Fabrice LAMAREILLE (IPSA)

Date de rendu du rapport
07 Septembre 2020

Année Universitaire
2019/2020

Remerciements

Il me sera très difficile de remercier tout le monde, car c'est grâce à l'aide de nombreuses personnes que j'ai pu mener ce stage à son terme, dans les meilleures conditions possibles.

Je voudrais tout d'abord remercier grandement ma responsable de stage, Aurélia **SECROUN** pour toute son aide.

Je suis ravi d'avoir pu travailler en sa compagnie, car outre son appui scientifique, elle a toujours été là pour me soutenir et me conseiller au cours de ce stage tout d'abord, et par la suite lors de ma recherche de thèse.

Je remercie également messieurs Thomas **ZERGUERRAS** et Hubert **HALLOIN** du laboratoire Astroparticules et Cosmologie de Paris pour toutes nos discussions, tous les conseils qu'ils ont su me donner au cours de la réalisation de ce projet, pour le temps qu'ils ont consacré avec moi pour effectuer des tests lors de mes visites à Paris.

Julien **ZOUBIAN** a également pris du temps pour m'aider en programmation avec l'utilisation de GitLab, ainsi que pour la mise en place de la machine de monitoring et je l'en remercie, ainsi que Adrien **RIVIERE**. Tous deux ont été réellement d'une aide précieuse.

Il m'est impossible d'oublier Kélian **SOMMER**, stagiaire de 4e année, avec qui j'ai pu partager environ deux mois de stage, qui ont été très productifs, car nous formions une bonne équipe, et avions la même façon de travailler

Je remercie également toutes les personnes avec qui j'ai eu l'occasion d'intéragir au cours de ces 6 mois de stage, Dominique **FOUCHEZ** responsable du groupe de cosmologie RENOIR au CPPM, Stéphanie **ESCOFFIER**, William **GILLARD**, Pierre **PRAT**, Samuel **SUBE**, Ouali **ACEF**, Rémi **GRANELLI**, Nathalie **BESSON** l'ensemble du personnel du CPPM, de l'APC et autres laboratoires avec lesquels j'ai pu travailler.

Enfin mes derniers remerciements iront à Fabrice **LAMAREILLE**, professeur à l'IPSA de Toulouse, qui à su prendre du temps, non seulement en m'encadrant durant ces 6 mois, mais également en m'aidant tout au long de ce stage et lors de ma recherche de thèse. Il m'a également fait répéter mes soutenances pour l'Ecole Doctorale, m'a fourni un certain nombre de lettres de recommandations, et a tout fait pour m'aider, m'a soutenu et surtout supporté dans tout ce que j'ai entrepris au cours de ma scolarité.

Avertissement

Ce stage, comme de nombreux autres, a été perturbé par la crise sanitaire mondiale et la mise en place d'un confinement. Une partie de ce dernier a donc été effectuée en télé-travail, du 16 Mars au 15 Juin.

Cette situation exceptionnelle a donc ralenti dans une certaine mesure la réalisation du banc de test, et par conséquent du monitoring, puisque nous ne pouvions malheureusement avoir accès à un certain nombre d'équipements.

Néanmoins, il m'a été possible de continuer à travailler à distances et donc d'approfondir la réalisation de la chaîne d'acquisition d'un équipement en particulier. Tout cela sera développé plus en détails dans le corps de ce mémoire.

Table des matières

I Le CPPM : Un acteur important au sein de grandes collaborations de recherche internationales	2
1 La recherche au CPPM, à la croisée des deux infinis	2
2 Le groupe de cosmologie RENOIR	6
II LISA, le défi technique et scientifique d'un interféromètre spatial	9
1 Un interféromètre spatial, pour la détection d'ondes gravitationnelles	9
1.1 Les Ondes gravitationnelles	9
1.2 La détection interférométrique	10
1.3 LISA, Laser Interferometer Space Antenna	11
2 Le Banc de Test AIVT	13
III Instrumentation, Monitoring et Contrôle-commande ; Mes contributions au projet LISA	16
1 Le monitoring, une découverte de l'instrumentation pour moi	16
2 Le choix du langage de programmation et la conception des drivers	19
3 Le stockage et la visualisation des données	26
4 QtDesigner – Une interface pour les drivers	30
IV Dernière Visite à l'APC et Bilan	38
1 Le Phasemètre Faire la mesure du signal interférométrique	38
1.1 La Méthode de Heinzel	38
1.2 Lire les données	42
2 Le test global des drivers	43

Fiche de Synthèse

Auteur : Alexandre CONDETTE – Aéro 5 ELSS

Sujet du stage

Définition et mise en œuvre du contrôle-commande d'un banc interférométrique pour la mesure des ondes gravitationnelles dans le cadre de la mission spatiale LISA de l'ESA

Objectifs

- Réalisation de drivers pour chacun des instruments du banc
- Réalisation d'une interface graphique de contrôle
- Réalisation d'une interface graphique de monitoring
- Mise en place de la structure globale du code

Client principal

CPPM, APC, LAM, Consortium LISA France, ESA

Outils utilisés

Python (Inclus dans environnement Conda), Qt-Designer, Programmation en C, Linux (Ubuntu), Grafana, InfluxDB, GitLab, CATIA, LaTex

Etudes réalisées

Etude des données du phasemètre, dans le domaine temporel et conversion dans le domaine fréquentiel via les méthodes usuelles de transformées de Fourier et via la méthode de Heinzel

Résultats

Drivers et base de données fonctionnels, Interface Homme-Machine fonctionnelle mais à améliorer (optimisation, placements des outils, fonctions internes), Interface de monitoring à compléter au fur et à mesure des besoins de chacun.

Drivers en python des phasemètres non fonctionnels et impossibilité technique de modifier les codes C .

Explications des écarts possibles

Le manque de temps et les problèmes liés au télé-travail peuvent expliquer que certains points ne soient pas terminés aujourd'hui.

Quant au phasemètre, l'explication est simple : le code C est très difficile à comprendre pour une personne qui n'est pas habitué à la programmation en C, et est trop complexe pour être modifié par un stagiaire je pense. Le code python des phasemètres n'est également pas fonctionnel à cause de problèmes de compatibilités réseau posés par le phasemètre en lui même.

Difficultés rencontrées

- Difficultés d'accès aux équipements (Confinement)
- Une majeure partie de la programmation se faisait "à l'aveugle" car les équipements se trouvent à Paris
- Découverte de l'utilisation de certains logiciels (Grafana, InfluxDB)
- Problèmes de compatibilités entre les OS (différentes versions de Linux utilisées)
- Compréhension du code C du phasemètre

Travaux à poursuivre

- Finir les drivers pour les quelques instruments restant
- Améliorer les scripts de test
- Faire fonctionner au moins 2 phasemètres synchronisés
- Terminer les onglets Grafana
- Traiter les données

Introduction

Dans le cadre de ma cinquième et dernière année à l'**Institut Polytechnique des Sciences Avancées (IPSA)**, j'ai eu l'occasion d'effectuer mon stage de fin d'études (d'une durée de 6 mois, du 3 Février au 31 Juillet 2020), au sein du groupe de cosmologie ReNOIR¹ du **Centre de Physique des Particules de Marseille (CPPM)**, laboratoire relevant de l'Institut National de Physique Nucléaire et de Physique des Particules (IN2P3) institut dépendant du Centre National de la Recherche Scientifique (CNRS).

Mon stage a été encadré par madame Aurélia SECROUN, ingénierie de recherche au CPPM, travaillant sur les missions Euclid et LISA² de l'Agence Spatiale Européenne (ESA), ainsi que par monsieur Fabrice LAMAREILLE, professeur en Sciences Aéronautiques et Spatiales à l'IPSA de Toulouse.

Le sujet de mon stage portait sur la "**Définition et mise en œuvre du contrôle-commande du banc de test AIVT³ pour la mesure des ondes gravitationnelles dans le cadre de la mission spatiale LISA de l'ESA**". Un sujet très enrichissant, que j'ai eu la chance de pouvoir choisir, et surtout obtenir, aboutissement de mon envie depuis de longues années de travailler dans un domaine proche de la recherche en astrophysique, en cosmologie ou en instrumentation. Une des autres raisons qui ont motivé mon choix a été la configuration du CPPM, me permettant de côtoyer ces deux aspects.

Le sujet portant sur les ondes gravitationnelles a également motivé mon choix. Ces événements, dont l'existence a été prouvée très récemment par les travaux de détecteurs au sol (LIGO⁴ et Virgo) vont ouvrir de nombreuses nouvelles perspectives pour l'astronomie et l'astrophysique de demain. De plus, LISA constitue un réel défi technique et scientifique dans ce domaine, mais nous reviendrons sur ce point ultérieurement au cours de ce rapport.

Les caractéristiques de ce détecteur spatial, l'ampleur du projet qu'il représente, mais également le fait d'appartenir à l'une des nombreuses contributions nationales et internationales que LISA suscite (comme la majorité des projets spatiaux) voilà ce qui a motivé mon choix, que je ne regrette pas.

J'ai donc pu travailler en tant qu'ingénieur stagiaire dans un secteur d'activité renommé en France : La recherche.

Notons que la recherche en France correspond à 2,2% de part du PIB (soit environ 49.8 milliards d'euros) et la France se place au 6e rang mondial en part de publication scientifique⁵. De plus le CNRS est également le "premier organisme de recherche au monde en nombre de publications scientifiques" d'après le site [Campus France](#), qui précise d'ailleurs que les chercheurs français sont reconnus dans le monde entier pour leurs travaux. Cela fut donc un réel plaisir de travailler pour cet organisme, d'être membre d'un consortium d'excellents laboratoires français.

Celà a également été l'occasion pour moi d'avancer sur de nombreux points, aussi bien personnellement, que professionnellement, de m'améliorer en programmation, de découvrir le fonctionnement d'un laboratoire de recherche, le déroulement d'une mission spatiale, la mise en place d'un banc de test... Découvertes que je vais vous partager au cours de ce rapport, que je diviserai en quatre parties. Dans un premier temps une présentation du laboratoire, du groupe de cosmologie et de l'ensemble des personnels avec lesquels j'ai eu la chance de travailler. Ensuite l'explication de la mission LISA, son but, les résultats attendus et mon rôle au sein de cette dernière. En 3e partie de ce rapport, le contenu effectif de mon travail sera exposé, et enfin dans les dernières pages de ce mémoire, je dresserai un bilan de ces 24 semaines de stage, conclusion de ma scolarité ipsalienne.

Dans un premier temps, nous allons donc présenter plus en détails le CPPM, le groupe RENOIR, ainsi que tous les autres laboratoires et toutes les personnes avec lesquelles j'ai intéragi au cours de ces 6 derniers mois.

¹ReNOIR : Recherche de l'énergie NOIR

²LISA : Laser Interferometer Space Antenna

³Assemblage, Integration, Validation and Test

⁴Laser Interferometer Gravitational-Wave Observatory

⁵Les données citées ici sont tirées du site du ministère de l'enseignement supérieur de la recherche et de l'innovation : <https://www.enseignementsup-recherche.gouv.fr/pid25351/chiffres-cles-de-la-recherche.html>

Part I

Le CPPM : Un acteur important au sein de grandes collaborations de recherche internationales

1 La recherche au CPPM, à la croisée des deux infinis

"La recherche au CPPM et au sein de l'IN2P3 permet de lier les deux infinis. L'infiniment petit afin de comprendre les briques élémentaires de la matière, mais aussi de l'infiniment grand, l'Univers et son histoire"

Cristinel DIACONNUS, Directeur du CPPM

Créé en 1983 et implanté sur plus de 4 500 m² en plein cœur du parc national des Calanques de Marseille sur le campus universitaire de Luminy, le CPPM est une unité mixte de Recherche (UMR 7346), signifiant l'association d'un ou plusieurs laboratoires de recherche d'un établissement d'enseignement supérieur ou d'un organisme de recherche avec le CNRS⁶. Dans le cas du CPPM, ce dernier relève de l'IN2P3, et est sous tutelle de l'Institut National de Physique nucléaire et de physique des particules du CNRS et de l'Université d'Aix-Marseille. C'est donc un laboratoire de physique fondamentale expérimentale, participant, comme le prouve mon expérience personnelle, de façon active à la formation des jeunes chercheurs par la recherche elle-même, et à l'enseignement universitaire dont il valorise chacun des résultats.



Figure 1: Cristinel DIACONNUS, Directeur du CPPM



Figure 2: Le CPPM, au coeur du parc national des Calanques de Marseille

Riche de quelques 180 employés (composés de chercheurs, ingénieurs, personnels techniques et administratifs, doctorants, post-doctorants, stagiaires), le CPPM est placé sous la direction de monsieur Cristinel DIACONNUS et de son équipe. D'après le rapport d'activités 2019 du CPPM voici la répartition du personnel au sein du laboratoire :

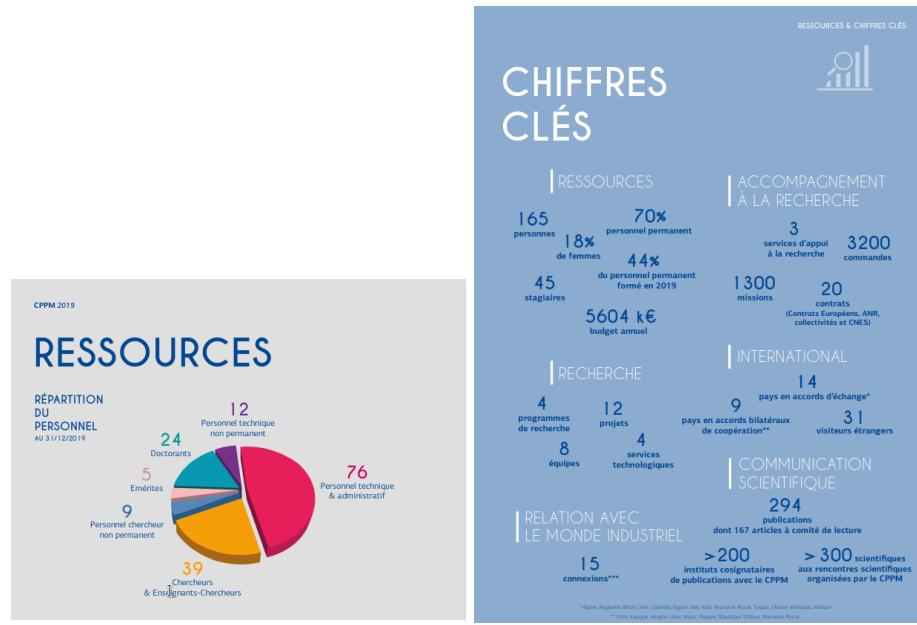


Figure 3: Ressources & Chiffres clés issus du rapport d'activités 2019 du CPPM - Auteure : Magali Damoisseaux

L'organisation au CPPM peut être divisée en plusieurs "poles" différents : - L'aspect technique sous la direction de Rémy Potheau, qui regroupe les services Electronique, Informatique, Patrimoine et Logistique...
 - Les services administratifs sous la responsabilité de madame Marie-Thérèse Donel
 - Les programmes scientifiques

Les principaux axes de recherche scientifiques du CPPM se situent à la croisée des deux infinis : L'étude de l'infiniment petit et de l'infiniment grand.

La physique des particules : Infiniment petit au travers de la physique des particules et des constituants élémentaires de la matière à l'exploration du cosmos. Le laboratoire marseillais conçoit et réalise des systèmes de détection à la pointe de la haute technologie et pouvant opérer dans des milieux aussi extrêmes que les fonds sous-marins, l'espace ou les entrailles de notre planète. Parmi ses principaux travaux on peut citer notamment les recherches sur le Boson de Higgs. Le CPPM ayant d'ailleurs participé à deux expériences placées au sein du Large Hadron Collider (LHC⁶) du CERN⁷ à Genève, avec notamment ATLAS, qui a permis en 2012 la découverte du boson de Higgs qui est, rappelons-le, une particule élémentaire dont l'existence permet d'expliquer la cassure de l'interaction unifiée électrofaible en deux interactions distinctes. Elle permet également d'expliquer pourquoi certaines particules possèdent une masse alors que d'autres n'en possèdent pas. Cette découverte a été récompensée en 2013 par le prix Nobel de Physique. Le Boson de Higgs est une des clefs de voûte de la physique des particules et ouvre la voie à la découverte future d'une nouvelle physique comme la matière noire ou la supersymétrie. L'équipe Atlas représente une part importante du CPPM avec 45 personnes la constituant depuis 1994, le CPPM a notamment participé à la conception de plusieurs éléments d'ATLAS comme le calorimètre électromagnétique ou le détecteur à pixel. Une autre équipe travaille également sur l'expérience LHCb au LHC, tandis qu'une troisième équipe se charge de l'expérience Belle II, installée au sein d'un accélérateur à électrons au Japon.

La cosmologie observationnelle : L'étude de l'infiniment grand : l'univers et son évolution, sa nature et sa dynamique. Sous branche de l'astrophysique étudiant la cosmologie par observation, elle mesure les grandeurs physiques des différents paramètres cosmologiques. Cette partie est le rôle de l'équipe RENOIR ; la partie I.2 de ce rapport y étant dédiée, nous ne nous y attarderons pas plus ici.

⁶Grand Collisionneur de Hadrons ; un hadron désigne un constituant d'un atome subissant l'interaction forte, comme par exemple un proton ou un neutron

⁷Conseil Européen pour la Recherche Nucléaire

Les astroparticules : L'étude de l'infiniment grand par l'infiniment petit, l'étude des phénomènes les plus violents de l'Univers et des particules de hautes énergies comme les neutrinos, particules engendrées par des réactions nucléaires produites par les supernovaes, voire même issues du Big Bang, voyageant à travers la matière à une vitesse proche de celle de la lumière mais n'intéragissant que très rarement avec elle. Ces neutrinos peuvent être détectés grâce à l'effet Cherenkov (émission lumineuse produite lorsqu'une particule chargée traverse un milieu transparent à une vitesse supérieure à la vitesse de la lumière dans ce milieu)⁸.



Figure 4: Un des éléments de KM3Net

L'équipe Neutrinos du CPPM s'implique dans plusieurs détecteurs à neutrinos comme ANTARES et KM3Net des détecteurs sous-marins.

Le CPPM travaille également sur le CTA (Cherenkov Telescope Array), qui est une nouvelle génération de télescopes gamma qui seront basés au sol et travailleront dans une gamme d'énergie gigantesque. La construction de cet immense observatoire astronomique, qui sera composé de pas moins de 118 télescopes de trois types différents, partagés entre un site situé aux îles canaries axé principalement sur l'étude des objets extragalactiques aux plus basses énergies possibles, et un site au Chili couvrant lui toute la gamme des énergies en se concentrant sur les sources galactiques, est censée s'achever en 2025. et sur le projet HESS pour la mesure des photons de haute énergie . (Nous savons par exemple des astronomes ont détecté récemment des photons à l'énergie record de 450 billions d'électrons-volts, émanant de la nébuleuse du crabe.

L'organigramme ci-dessous résume cette partie de présentation du CPPM.

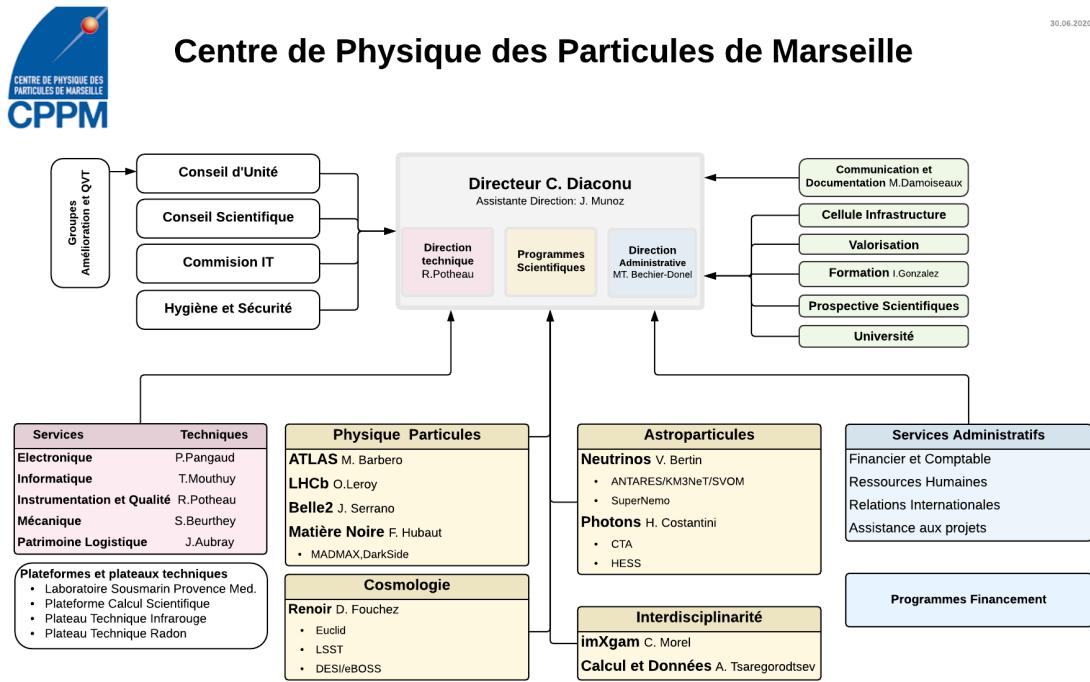


Figure 5: Organigramme du CPPM - Source : Site web du CPPM

J'ai pu rencontrer de nombreuses personnes au cours de mon stage, m'entretenir avec elle, et également discuter avec le directeur, notamment grâce à l'accueil réservé aux stagiaires et tout autre nouvel arrivant dans ce laboratoire, avec visite du site, discussion avec le service informatique ainsi qu'avec le service communication, avec le directeur...

⁸Source : https://www.laradioactivite.com/site/pages/Effet_Cherenkov.htm

De plus, les lundis de chaque semaine, était organisé au CPPM un séminaire, dans l'amphithéâtre, traitant sur divers sujets scientifiques (Ondes Gravitationnelles, Electronique, Matière noire, Centre de Calcul IN2P3...)

2 Le groupe de cosmologie RENOIR

L'équipe dont j'ai pu faire partie au cours de mon stage au CPPM se nomme le groupe de cosmologie RENOIR (pour Recherche Energie NOIR). Celui-ci, comme son nom l'indique, s'intéresse aux questions importantes de la cosmologie actuelle, notamment le phénomène d'expansion de l'Univers prédit en 1998 par Saul Perlmutter, Adam Riess et Brian Schmidt (Prix Nobel de physique 2011), phénomène s'expliquant notamment grâce à la présence d'énergie noire (DE⁹) dans la cosmologie. Les origines de l'énergie et de la matière noire représentent une des questions importantes de la physique actuelle car ces deux éléments encore non-identifiés, représentent ensemble plus de 95% des travaux de recherche au sein de ce groupe s'orientent également vers la recherche de la matière noire (DM¹⁰) qui pourrait être soit une nouvelle forme d'énergie, une constante cosmologique ou une déviation de la théorie de gravitation.

Le groupe est composé d'une vingtaine de personnes, incluant des chercheurs, ingénieurs, informaticiens, doctorants, et stagiaires.



Figure 6: Photographie du groupe Renoir

L'énergie noire, qui composerait plus de 70% de la masse de l'Univers, est une vraie question actuelle de la recherche en cosmologie et c'est pour cela que de très nombreuses expériences sont mises en place pour l'étudier, et ainsi pouvoir contraindre les paramètres cosmologiques (c'est à dire des paramètres qui permettent de décrire notre Univers, son comportement ; on peut par exemple citer le paramètre de Hubble-Lemaître H₀, décrivant l'expansion de l'Univers en donnant la vitesse d'une galaxie en fonction de sa distance).

Le CPPM et le groupe RENOIR sont impliqués dans de grandes expériences, en cours ou sur le point de commencer, permettant ou qui permettront d'étudier ce phénomène d'expansion. Ces projets demandent la mise en place d'instruments spécifiques destinés à ce genre de mesure.

Les objectifs étant de parvenir à mesurer les différents paramètres cosmologiques, ce qui permettrait de comprendre l'accélération soudaine de l'expansion de l'univers, le groupe de recherches Renoir s'est principalement concentré sur deux sondes et sur les mesures de grandes structures. Des observations ont notamment donné lieu à des analyses photométriques et spectroscopiques poussées, donnant les toutes premières indications précises des paramètres cosmologiques.

On pourra citer par exemple les missions eBOSS¹¹/DESI¹², missions consistant à mesurer le spectre de nombreuses galaxies afin de contraindre les propriétés de l'énergie noire, en utilisant plusieurs types de sondes cosmiques comme par exemple les Oscillations Acoustiques Baryoniques (BAO¹³) pour eBOSS : Ondes de pressions qui se sont formées dans le plasma primordial (entre le Big-Bang et 380 000 ans après ce dernier),

⁹DE : Dark Energy

¹⁰DM : Dark Matter

¹¹eBOSS : Extended Baryon Oscillation Spectroscopic Survey

¹²Dark Energy Spectroscopic Instrument

¹³BAO : Baryonic Acoustic Oscillations

et qui ont créée des fluctuations de densité dans ce dernier. On peut retrouver aujourd’hui ces traces dans le fond diffus cosmologique (CMB¹⁴), comme une zone de surdensité au sein des galaxies à environ 150 Mpsec¹⁵.

Le groupe participe aussi à de grandes collaborations internationales, comme par exemple la mission Euclid de l’ESA : un télescope spatial, qui travaillera dans le domaine du visible et du proche infrarouge afin d’effectuer des mesures de BAO, mais également mesurer d’autres sondes cosmologiques comme les phénomènes de cisaillement gravitationnel (strong/weak lensing), c’est à dire la déviation des rayons lumineux qui nous arrivent par exemple d’une galaxie, à cause de la présence d’un corps massif entre la source lumineuse et nous, le champ gravitationnel de ce corps va avoir pour effet de déformer l’espace-temps autour de lui et ainsi dévier les rayons lumineux, ce qui aura - normalement pour conséquence, une image déformée de la galaxie pour un observateur terrestre.

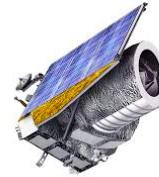


Figure 7: Vue d’artiste d’Euclid

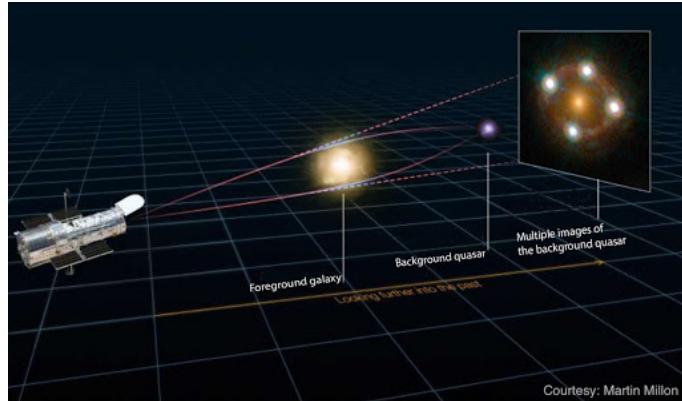


Figure 8: Illustration du phénomène de lentillage gravitationnel

Sur Euclid le groupe est notamment responsable de la caractérisation des détecteurs infrarouge du spectro-photomètre, le NISP, mais également la partie « segment-sol » d’Euclid, c’est à dire la phase de traitement et de distribution des données à chacun des pays participant ; Le CPPM se charge principalement de la production de simulations, notamment sur le NISP.

Enfin, le CPPM travaille également sur le projet LSST le Legacy Survey of Space and Time, télescope du Vera-Rubin observatory au Chili, qui est un important projet de télescope de 8m basé au sol et couvrant la totalité du ciel grâce à plusieurs bandes photométriques, ce qui permet d’obtenir un champ d’observation extrêmement large et optimisé pour la mesure d’autres sondes cosmologiques comme les supernovae ou également des mesures de cisaillement. Ce télescope hyper puissant est capable de révéler en l’espace de 15 secondes les objets célestes 15 millions de fois moins lumineux que l’objet le plus faible que l’œil nu peut percevoir. Sur ce projet, le CPPM est impliqué dans la construction du chargeur de fibres, mais également travaille sur d’autres aspects, comme par exemple le groupe RENOIR travaillant également sur la préparation des analyses, de la calibrations (avec notamment la participation à d’autres



Figure 9: Vue sous-projets comme StarDICE, pour la calibration photométrique du LSST). En plus d’être impliqué technique à tous ces projets, le groupe RENOIR travaille également sur le traitement des données générées par ces différentes expériences.

Ces différents projets permettent de réaliser des mesures de plusieurs sondes et apporteront d’importantes et surtout très précises informations sur les origines et les interprétations du phénomène d’accélération de l’Univers. Renoir se charge pour cela de développer les outils qui permettront d’étudier et expliquer afin de comprendre mieux la nature de l’énergie noire qui est à la base de cette accélération.

¹⁴CMB : Cosmic Microwave Background

¹⁵Mpsec = Méga-parsec ; 1 Parsec = $\frac{648000}{\pi}$ Unités Astronomiques ($\simeq 3.086 \times 10^{16} m$)

Enfin, un autre axe de recherche de la cosmologie actuelle, qui est amené à se développer dans les années à venir est également traité au sein du groupe RENOIR, du moins d'un côté technique pour l'instant, il s'agit du sujet des ondes gravitationnelles, avec une participation à une mission internationale portant le nom de mission LISA, sujet principal de mon stage dont je vais pouvoir vous parler plus en détails maintenant...

Part II

LISA, le défi technique et scientifique d'un interféromètre spatial

1 Un interféromètre spatial, pour la détection d'ondes gravitationnelles

1.1 Les Ondes gravitationnelles

Phénomène prédit par Albert Einstein dans la théorie de la relativité générale, une onde gravitationnelle est comme son nom l'indique, une onde qui se propage, une déformation de la courbure de l'espace-temps, que l'on pourrait comparer par analogie, à une vague à la surface de l'eau.

Ce genre de phénomène peut se produire lorsque deux objets très massifs (par exemple deux trous noirs) sont proches, si proches qu'ils peuvent entrer en collision et fusionner ; Comme une pierre que l'on aurait jetée au milieu d'un lac. Une onde va alors se créer au point de collision des trous noirs, et se propager dans l'espace-temps, à la vitesse de la lumière et ainsi nous parvenir après de longues années.

La convergence en spirale de deux trous noirs a provoqué ces ondes gravitationnelles.

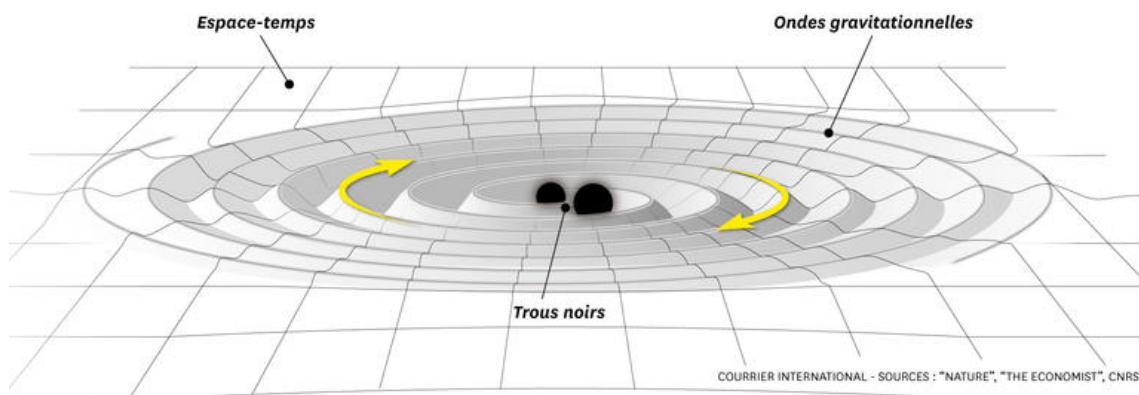


Figure 10: Illustration de la déformation de l'espace-temps à proximité d'un binaire de trous-noir

L'étude des ondes gravitationnelles, qui n'interagissent pas avec la matière, pourrait constituer une nouvelle sonde cosmologique, notamment pour l'étude des phénomènes de trous noirs, mais également celle de l'univers à grande échelle (étude des larges structures, contraintes des paramètres cosmologiques comme la constante de Hubble). Une onde gravitationnelle permettant de mesurer directement la dilatation de l'espace-temps. Elles permettraient également de par leur nature, d'étudier la jeunesse de l'Univers, les ondes gravitationnelles originelles créées par le Big Bang pouvant hypothétiquement être captées par des détecteurs terrestres mais surtout spatiaux.

L'existence des ondes gravitationnelles a été prouvée en 2015 par les travaux conjoints de deux détecteurs au sol : le Laser Interferometer Gravitational-Wave Observatory (LIGO) aux Etats-Unis et le Virgo en Europe (Italie).



Figure 11: Les interféromètres au sol, LIGO à gauche et Virgo à droite

Le point commun que l'on peut remarquer entre ces deux détecteurs, sont leurs deux long bras orthogonaux de 4 km de long pour LIGO et de 3 km de long pour Virgo. Ces bras étant les constituants principaux du principe de détection des ondes gravitationnelles : l'interféromètre, basé sur le principe d'interféromètre de Michelson.

1.2 La détection interférométrique

Le principe est relativement simple du moins en théorie ; un faisceau laser unique, de fréquence bien connue est divisé par une lame séparatrice en 2 faisceaux orthogonaux, qui vont se propager dans les bras des interféromètres. Ces bras sont en réalité des cavités de Fabry-Pérot, c'est à dire un système constitué de miroirs à haut coefficient de réflexion, permettant de pouvoir artificiellement augmenter la longueur de chaque bras, car comme nous le verrons ultérieurement, la sensibilité de l'interféromètre est directement liée à leur longueur.

Le signal est ensuite réfléchi au sein de chaque bras sur un miroir et renvoyé vers un détecteur, à savoir un ensemble de photodiodes. Les signaux des deux bras se recombinent et produisent une figure d'interférence, c'est ce qu'on appelle l'interférométrie homodyne de phase, c'est à dire que l'on a utilisé un seul laser, ou deux lasers de même fréquence, et que l'on obtient grâce à ce procédé une figure interférométrique sur l'amplitude.

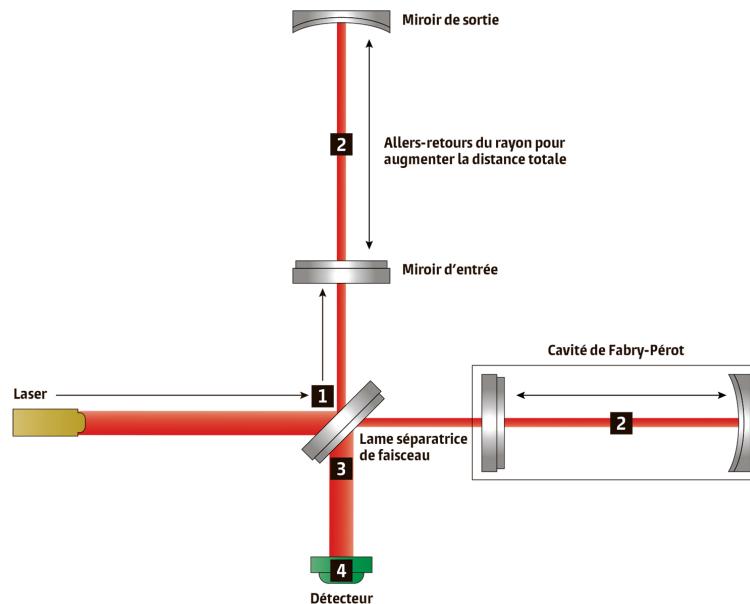


Figure 12: Principe de fonctionnement d'un interféromètre de Michelson, comme LIGO/Virgo

Le passage d'une onde gravitationnelle va avoir pour conséquence d' agir sur l'espace-temps au sein des détecteurs (le dilater sur un des bras, le comprimer sur l'autre) ce qui va donc modifier la figure d'interférence

au niveau des photorécepteurs, comme on peut le voir sur les courbes ci-dessous, avec à gauche une simulation de ce que l'on devrait obtenir (la déformation observée par rapport au temps) et à droite, les données réellement captées par LIGO en 2015.

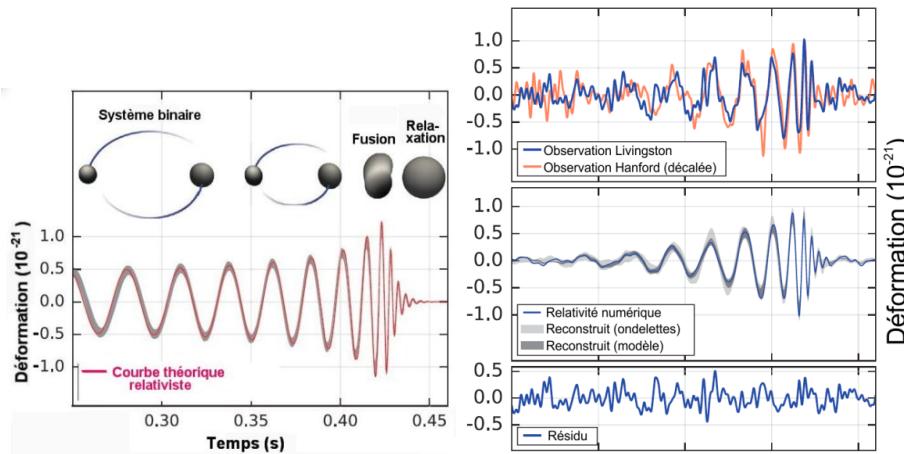


Figure 13: La détection d'une onde gravitationnelle, à gauche un modèle prévisionnel et à droite la détection effective d'une onde par LIGO en 2015

Comme indiqué plus haut, la taille des bras de l'interféromètre influe grandement sur l'amplitude des ondes qu'il peut détecter. En effet, si notre interféromètre possède des bras de longueur l , et qu'une onde d'amplitude h (qui est très petite) passe dans l'un d'eux, la variation du chemin optique δL vaut alors :

$$\delta L = L \times k \times h$$

Avec k un facteur de forme $<= 1$

Pour détecter des ondes moins amples, se trouvant à des longueurs d'ondes plus basses (LIGO et Virgo sont sensibles entre des fréquences de 1 Hz à 10 kHz). Il faut donc construire des interféromètres plus grands. Cependant sur Terre, plusieurs facteurs sont limitant pour construire des interféromètres géants :

- La courbure de la Terre : cette dernière à pour effet, puisque le laser ne va pas suivre la courbure de la Terre (il s'agit de photons, qui voyagent en ligne droite), le fait qu'au-delà d'une certaine distance, le laser ne se reflètera pas sur le miroir et donc ne sera pas capté par le détecteur.

- Le bruit sismique : vibrations du sol, activités humaines etc., qui génèrent énormément de bruit à basses fréquences. Pour pallier à ces deux problèmes, une solution « évidente » s'impose : utiliser l'espace, ce qui est le but de la mission LISA, un interféromètre spatial.

1.3 LISA, Laser Interferometer Space Antenna

Avec LISA, les contraintes de distances sont complètement abolies, ainsi que celles liées au bruit sismique. En effet LISA sera une constellation de 3 satellites, voyageant en triangle équilatéral en orbite héliocentrique, suivant la Terre lors de sa rotation autour du Soleil avec un retard d'environ 20° (50 millions de kilomètres) ; la formation sera sur un plan incliné à 60° par rapport à l'écliptique (le plan de rotation de la Terre par rapport au Soleil).

Chacun des bras de LISA (il y en aura 6 au total, 2 pour chaque « cotés » du triangle) mesurera 2,5 millions de kilomètres permettant à LISA d'obtenir une sensibilité égale à LIGO ou Virgo, mais sur une plage de fréquence totalement différente, puisque LISA travaillera entre 0,1 mHz et 1 Hz.

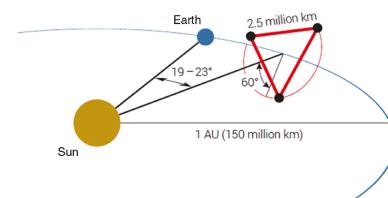


Figure 14: Représentation de la constellation LISA

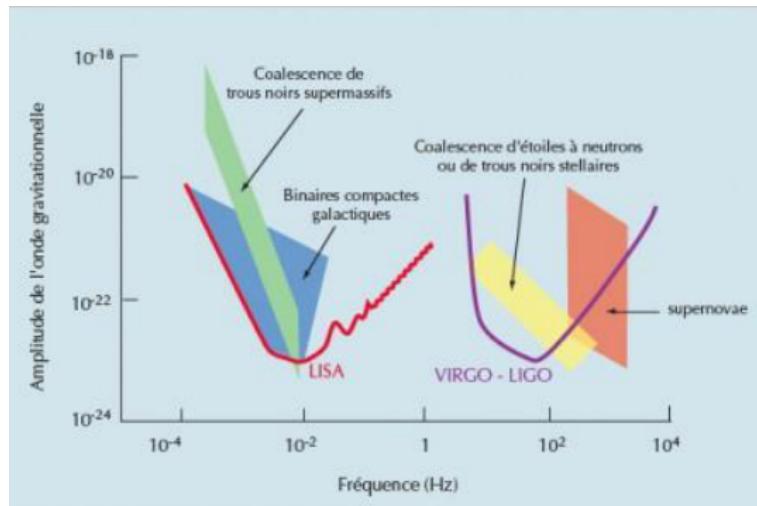


Figure 15: Sensibilité de LISA et des interféromètres au sol en fonction de leur bande de fréquence

Mais pour la capacité à détecter ces ondes gravitationnelles avec la précision requise qui est de $10pm/\sqrt{Hz}$ (soit pouvoir détecter une variation de 10 pm sur 2,5 millions de kilomètres, sur toute la bande de fréquence de LISA) , cela relève d'un véritable défi technique, puisque devant traiter un certain nombre de bruits différents autre que les bruits terrestres (on pourra notamment parler pour LISA du bruit thermique, qui est le problème majeur des photorécepteurs, bruit induit par l'électronique...) devant également utiliser de nouvelles techniques de détection, comme l'interférométrie hétérodyne de phase (utilisation de deux lasers de fréquences différents mais proches, produisant une interférence de phase, moins sensible à l'environnement, plus précis et efficace à basses fréquences).

Le problème se posant aujourd’hui est que ces techniques n’ont pas encore toute été démontrées, de même que de nombreux autres éléments du projet comme les photorécepteurs.

C'est pour cela qu'en ce moment même est mis en place un banc de test AIVT au laboratoire Astroparticules et Cosmologie de Paris, afin de démontrer la faisabilité du projet et permettant d'effectuer des premiers tests sur de nouveaux composants.

2 Le Banc de Test AIVT

De nombreux acteurs sont impliqués dans la réalisation de la mission LISA, et la France ne fait pas exception. Le diagramme ci-dessous montre les implication de chaque pays dans le projet.

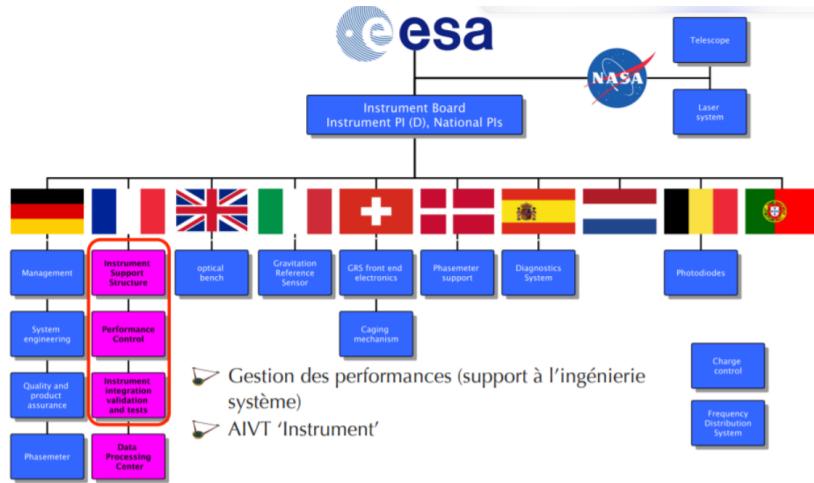


Figure 16: Organigramme représentant l'implication de chaque pays membre de l'ESA au coeur de la mission LISA

Comme le montre ce diagramme, le consortium LISA France, composé d'environ 150 membres répartis dans 15 laboratoires, a en charge principalement des activités de type "AIVT" (Assembly, Integration, Validation and Test) et mon travail s'inscrit dans la contribution LIG, qui se focalise sur l'instrument en lui-même, une contribution dirigée par le laboratoire AstroParticules et Cosmologie de Paris (APC) et co-dirigé par le laboratoire ARTEMIS de l'Observatoire de la Côte d'Azur.

La première étape du projet est la réalisation d'un banc de test interférométrique dont la réalisation devrait être terminée fin 2020 / début 2021. Un projet dans lequel sont impliqués de nombreux laboratoires :

- APC (Paris) : Direction du projet | Banc de test
- ARTEMIS (Nice) : Photorécepteurs + Diffusion
- CEA/IRFU (Paris) : Chaîne de lecture | Modèle thermo-mécanique
- CPPM (Marseille) : Monitoring & Contrôle-Commande + Caractérisation des Photodiodes LISA
- Institut Fresnel (Marseille) : Etudes de diffusion
- LAM (Marseille) : Environnement
- SYRTE (Paris) : Réalisation du laser

Le but de ce banc de test est de démontrer la faisabilité du projet, et la possibilité d'atteindre la précision de nombreux instruments, notamment la précision requise sur les phasemètres, la possibilité de détecter un signal de l'ordre de précision requis sur LISA ($10\text{pm}/\sqrt{\text{Hz}}$).

Ce banc de test sera composé d'un certain nombre d'éléments comme le montre le schéma ci-dessous :

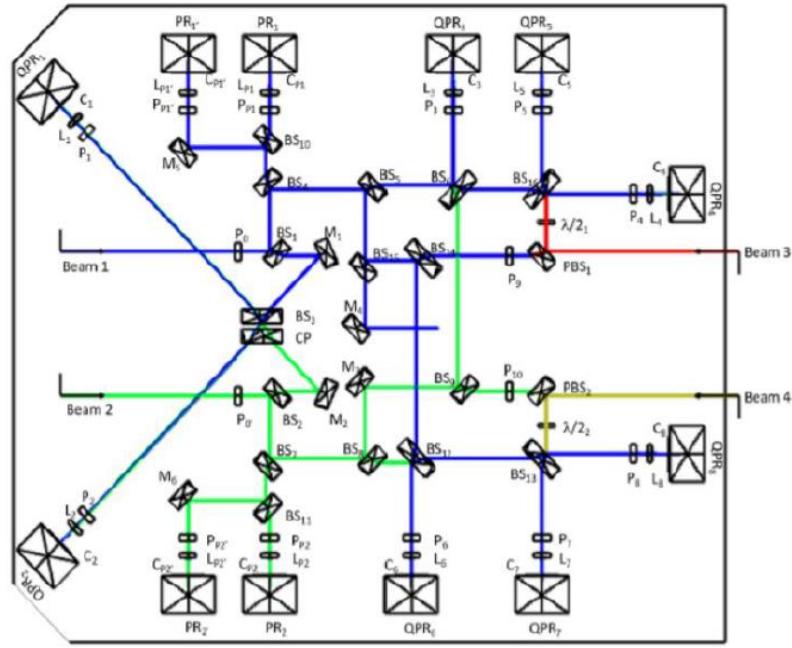


Figure 17: Représentation du banc de test interférométrique de LISA

Nous pouvons notamment voir sur ce schéma, les traits bleus et verts représentant des interféromètres à bras égaux (mêmes distances parcourues par les lasers) permettant de faire la calibration du banc de test, l'alignement des injecteurs des lasers et permettant ainsi d'obtenir un signal interférométrique de référence. La partie gauche du banc de test elle correspond à des interféromètres à bras inégaux, permettant de "simuler" le télescope de LISA, le MOSA¹⁶.

Le banc de test actuellement en cours de réalisation à l'APC est un montage optique "classique" que l'on nomme MIFO pour "Metallic Interferometer", étant réalisé en Invar : un alliage de fer et de nickel au coefficient de dilatation très faible. Matériaux qui est d'ailleurs également très utilisé en aéronautique, pour la réalisation des moules de pièces composites.

Parmis les composants de ce banc de test nous pouvons citer :

- Des photorecepteurs simples
- Des photodiodes à quadrant
- Des résistances chauffantes
- Des capteurs de températures
- Les injecteurs des lasers

Tous ces éléments seront placés dans une enceinte sous vide et régulée thermiquement.

Une seconde batterie de tests sera effectuée au sein du Laboratoire d'Astrophysique de Marseille (LAM) sur une deuxième version du banc interférométrique, réalisée cette fois en Zerodur, un verre au coefficient de dilatation presque nul et extrêmement homogène, ce qui permettra d'obtenir des résultats encore plus précis, d'autant plus que pour cette série de tests, le banc sera placé au sein de la cuve ERIOS¹⁷ permettant la caractérisation, l'étalonnage spectrophotométrique et les réglages optiques sous vide secondaire (pression comprise entre 10^{-3} et 10^{-5} mbar) froid suivant les normes ISO 8 et ISO 5)

¹⁶MOSA : Movable Sub-Optical Assembly : Télescope qui captera le signal laser d'un satellite de LISA au sein d'un autre.

¹⁷ERIOS : Etalonnage Réglage et Intégration pour l'Optique Spatiale



Figure 18: La cuve ERIOS du LAM - Site du LAM

ISO 3	ISO 4	ISO 5	ISO 6	ISO 7	ISO 8
Microélectronique : Fabrication de semi-conducteurs circuits intégrés avec des géométries submicroniques.					
		Chimie fine Pharmaceutique : Médicaments injectables, Production d'implants prothèses chirurgicales, Conditionnements buvables, Gélules.			
		Supports d'information : Fabrication de films plastiques, Cassettes vidéo, Disques durs micropographies.			
			Électronique / Opto-électronique Matériel optique de haute précision. Assemblage de micro supports.		
			Micro-Mécanique : Appareillage de mesure, Roulement Optique, Robinetterie, Instrumentation de bord.		
			Industries Agro-Alimentaires : Plats cuisinés, Boissons, Industrie de la viande, Conditionnement.		
				Spatial : Assemblage et intégration de satellites, Fabrication de miroirs.	
				Automobile : Cabines de peinture, Equipements électriques.	
					Hydraulique et Pneumatique : Assemblage de composants.

Figure 19: Les normes ISO pour les salles blanches

Nous avons vu il y a quelques instants que sur ce banc de test se trouveront deux types de photodiodes : Des photodiodes simples et des photodiodes à cadrants ; Ces composants sont des éléments importants du montage, qu'il nous faut monitorer (monitorer chacun des 4 cadrans des photodiodes à quadrant, mais également faire des corrélations de cadrans deux à deux) afin d'obtenir plusieurs mesures, notamment les mesures de phase issues des phasemètres.

Il nous faudra aussi pouvoir contrôler la température, le réglage des lasers etc. Mais comment faire cela dans une cuve fermée me direz-vous ? Et bien il s'agit là de tout le principe du monitoring et du contrôle-commande, sujet principal de mon stage et dont nous allons parler dans la suite de ce rapport.

Part III

Instrumentation, Monitoring et Contrôle-commande ; Mes contributions au projet LISA

1 Le monitoring, une découverte de l'instrumentation pour moi

Dans ces deux dernières parties de rapport, je vais donc détailler au mieux toutes mes activités lors de ce stage de fin d'études.

Dans ces deux dernières parties de rapport, je vais donc tenter au mieux de détailler toutes mes activités lors de ce stage de fin d'études, qui m'a permis de me conforter dans mes ambitions professionnelles futures en me permettant par exemple d'apprendre à maîtriser l'utilisation d'une plateforme permettant de partager les données et les codes communs avec l'ensemble de toute une équipe, d'obtenir une version de gestion de ces codes optimale et de savoir également créer des alertes de correction permettant d'intervenir en cas de bug informatique par exemple. Moi qui était déjà passionné de programmation au préalable ais eu encore grâce à ces travaux la possibilité d'améliorer et approfondir mes connaissances dans ce domaine.

Comme vu dans les parties précédentes, ma mission au CPPM et en particulier au sein du projet LISA a été la mise en place du monitoring et du contrôle-commande du banc de test AIVT, un rôle que je qualifierai de central, car devant m'occuper de la gestion du banc, mais également de la récupération des signaux de tous les instruments autour de ce dernier, cette position m'a permis d'être à l'interface de tous les laboratoires cités précédemment, qui contribuent d'une manière où d'une autre à ce montage optique.

Ma principale mission a donc été de communiquer avec chaque acteur, afin de définir un cahier des charges et ainsi pouvoir faire plusieurs choix techniques pour la mise en place de toute l'interface de monitoring et de C&C (Contrôle-Commande).

L'illustration ci-dessous résume l'ensemble des éléments que j'ai pu identifier comme faisant partie de cette interface.

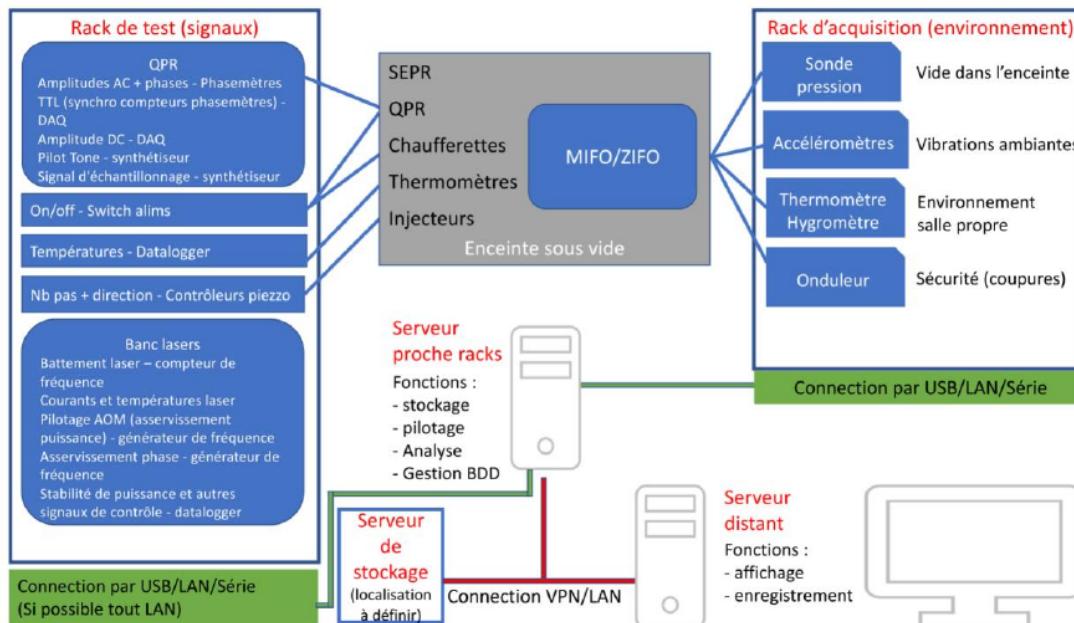


Figure 20: Représentation de l'ensemble des signaux à montrer - Aurélia Secroun & Alexandre CONDETTE

Comme nous pouvons le voir sur cette illustration, deux types de signaux sont à considérer, les signaux "Test" correspondant aux mesures effectuées par les instruments du banc, notamment sur les lasers, ainsi que des signaux d'"environnement" permettant de monitorer également les conditions d'acquisition.

Toutes ces données sont ensuite envoyées vers un serveur proche de la manipulation, serveur que j'ai eu la chance de pouvoir configurer avec certains collègues, sous un système d'exploitation Linux (Ubuntu 20.04), et dont j'ai également participé à la commande, en fonction des spécificités requises pour notre banc de test, notamment les capacités de calculs et de mémoire (par exemple, nous avons décidé la mise en place de 192 Giga-octets de mémoire vive (RAM), et une capacité de stockage de 30 Téra-octet répartie sur plusieurs disques en parallèles).

Ce serveur servira à acquérir les données, au stockage de ces dernières (en données brutes et en données traitées) et servira également au pilotage du banc de test, avec la mise en place d'une interface de commande.

Nos moyens de monitoring et de C&C se trouveront donc sur ce serveur, qui sera accessible depuis n'importe quel autre ordinateur (à l'APC ou à distance, via un VPN) pour la partie monitoring, permettant ainsi à chacun de pouvoir accéder en temps réel aux données utiles pour eux. La partie commande quant à elle n'est accessible que depuis le serveur en lui-même directement par l'interface que nous détaillerons plus tard, soit via une machine dédiée au monitoring, placée en salle de contrôle. Cette machine dotée de 4 écrans 27 pouces et d'une excellente carte graphique, nous permet d'afficher notre interface de monitoring, mais également, via un accès en bureau à distance au serveur (VNC¹⁸ : X2GO), d'agir sur quelques fonctionnalités de l'interface de commande.

J'ai donc pu au cours de mon stage, effectuer plusieurs visites à Paris, à l'APC, ainsi qu'au SYRTE afin de voir la réalisation du laser par Ouali ACEF et discuter avec les responsables du projet à l'APC, Thomas ZERGUERRAS et Hubert HALLOIN.

De mes discussions avec chacun, j'ai pu identifier un certain nombre de contraintes pour la réalisation du monitoring, qui m'ont permis de faire des choix techniques, voici quelques contraintes majeures :

- Le nombre de signaux : 250 signaux à monitorer quasiment en continu
- La rapidité d'acquisition : La plupart des signaux sont assez lents, car acquis à une fréquence de 1 Hz, néanmoins, les signaux issus des phasemètres sont beaucoup plus rapides car ayant une fréquence de 400 Hz, et devant en plus faire l'objet d'un traitement en aval. Il faut donc être capable de gérer ces signaux.
- La multitude d'instruments à gérer : Une quinzaine d'instruments en simultané, avec des modes de communications différents (Ethernet, USB, GPIB)
- Le fait de devoir agir sur ces instruments également.

Toutes ces contraintes ont été à l'origine des choix suivants.

Dans tous les cas, la logique reste la même, nous avons un instrument, prenons par exemple les contrôleurs des moteurs, des modules Newport 8742 (cf photo), cet instrument est relié au banc interférométrique, et fait des mesures sur ce dernier (dans certains cas il agit aussi sur certains composants, comme les contrôleurs, qui agissent sur les injecteurs des lasers). Cet instrument doit ensuite être relié à la machine (le serveur proche du banc), mais la machine n'est pas capable de communiquer directement avec l'instrument, il faut donc utiliser un "driver" pour faire l'interface entre les commandes de l'ordinateur et les instruments afin de pouvoir faire en sorte que l'ordinateur puisse lire les données envoyées par l'instrument, et dans un second temps, que nous puissions agir sur les réglages de cet instrument directement via notre ordinateur (par exemple changer la position d'un moteur).

Ces drivers permettent donc de faire le lien entre l'instrument physique, et la machine qui permet de contrôler ces instruments. Au sein du serveur, les données sont stockées sur des disques durs (HDD) et celles qui ont besoin d'être traitées (par exemple les signaux des phasemètres dont nous reparlerons plus tard) sont envoyées sur un SSD pour le traitement via d'autres scripts (Python / Matlab).

Sur le serveur se trouve également l'interface de commande, qui permet plusieurs choses :

- Lancer l'acquisition des données

¹⁸VNC : Virtual Network Computing : Service d'accès bureau à distance

- Lancer la communication avec chacun des instruments indépendamment
- Contrôler le réglage des injecteurs
- Agir sur les générateurs de tension / voltage

Nous reparlerons ultérieurement de cette interface.

Les drivers ont également un autre rôle, ils font le lien entre la machine, et le stockage en Base de Données des informations issues des instruments. Les données stockées sur les disques sont envoyées vers une base de données.

Cette base de données transmettra les renseignements qu'elle aura reçus à une interface de visualisation placée sur l'ordinateur de monitoring qui sera accessible via n'importe quel ordinateur, grâce un accès sécurisé en VPN¹⁹, et qui va également gérer des alertes (seuils, coupure de courant) et est capable de s'auto-actualiser à intervalles réguliers, en allant chercher les données dans la base.

Le schéma ci-dessous résume cette vision du monitoring.

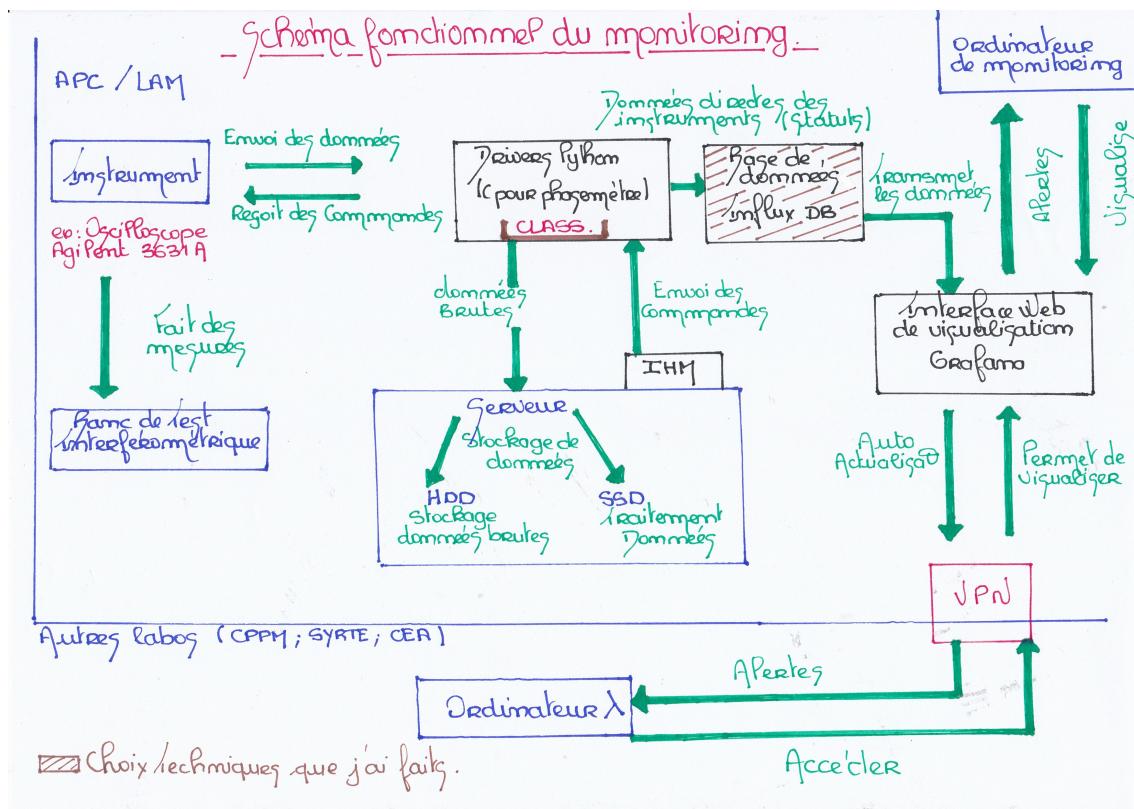


Figure 21: Schéma fonctionnel du Monitoring dans son ensemble

Nous allons maintenant détailler chacun des éléments dans les sous parties suivantes

¹⁹VPN : Virtual Private Network

2 Le choix du langage de programmation et la conception des drivers



Figure 22: Logo de Python

Le premier choix technique que j'ai été amené à faire à été le langage de programmation à utiliser pour la réalisation des drivers.

J'ai fait le choix d'utiliser la programmation en python pour de nombreuses raisons. La première étant que c'est le langage dans lequel je me sens le plus à l'aise, et dont nous avons eu le plus recours au cours de notre scolarité à l'IPSA, c'était déjà un élément en faveur de Python.

De plus, Python est un langage de programmation haut niveau, et très pratique pour la programmation orientée objet comme c'est le cas ici. Ce stage m'a d'ailleurs permis de réellement comprendre ce que signifie le terme "programmation orientée objet", car ici il fallait programmer pour communiquer avec un instrument physique, un objet, et ce fut une façon assez nouvelle pour moi d'appréhender la programmation.

Python est également un langage de plus en plus répandu dans de nombreux domaines, notamment dans le secteur du spatial.
De plus, grâce à ses nombreux gestionnaires de paquets et ses librairies développées par toute la communauté, Python permet d'avoir des modules pour la communication série (USB/GPIB avec PyVisa), la communication éthernet (Paquet Socket intégré à Python), la réalisation de base de données (MySQL, Influxdb), et également la lecture, la sauvegarde des données (JSON, HDF5).

De plus Python est Open source et multiplateforme, donc les codes sont facilement transposables d'un ordinateur à un autre.

Pour faciliter cette portabilité du code, j'ai crée un environnement virtuel sous Anaconda (conda 4.8.3, python version 3.7.2) appellé LISA_MONITORING.

Cet environnement à une utilité cruciale dans ce genre de projet, au lieu d'installer les paquets sur la machine, on les installe dans l'environnement et tous nos programmes fonctionneront dans cet environnement, ce qui permettra d'éviter des problèmes de versionnement, mais également la portabilité du code, car au lieu d'avoir à installer tous les paquets un par un en faisant attention aux versions lorsque l'on porte le programme, il suffira d'installer l'environnement.

Comment ça fonctionne ? : On crée l'environnement grâce à une commande spécifique.

Dans un terminal (Ici Ubuntu 20.04)

```
conda create --name LISA_MONITORING --file=requirements.txt
```

Le document *requirements.txt* est un fichier regroupant toutes les librairies utilisées dans l'environnement (un exemple de document de ce type est disponible en Annexe).

Voici une liste non-exhaustive de modules utilisés lors de mon stage : Pyvisa, Influxdb, JSON, csv, socket, logging, doctest, os...

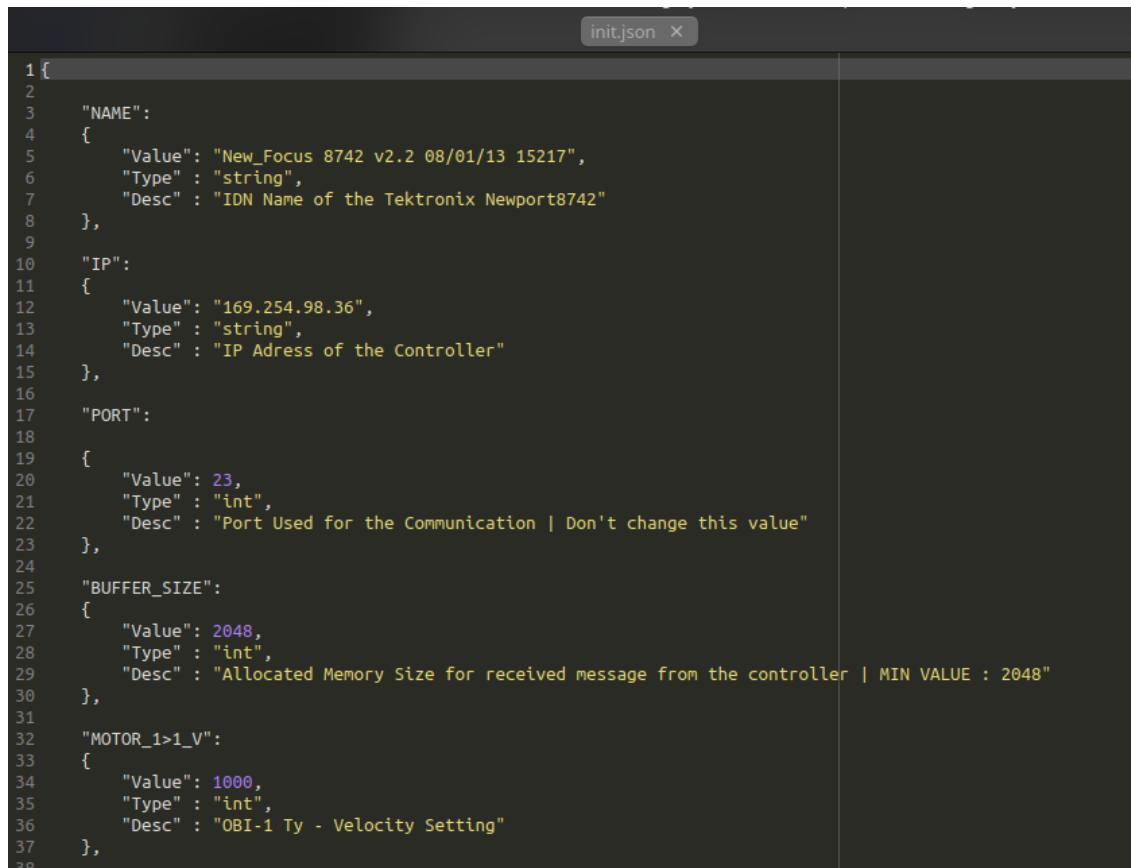
Dans tous les cas, ces drivers sont constitués de la même manière : Une classe Python portant le nom de l'équipement (ex: Newport8742) suivie d'une fonction d'initialisation, comme ceci :

L'élément que je prends ici en exemple, le contrôleur Newport8742 est le mieux placé pour illustrer mon stage, car ce module a été le seul auquel j'avais accès lors de la crise sanitaire et du confinement, j'ai du focaliser mon travail sur cet élément, et la réalisation de toute la chaîne de mesure, en partant des moteurs, jusqu'à la lecture sur l'interface de monitoring sachant également que cet élément a besoin d'être contrôlé via l'interface Homme-Machine.

```
class Newport8742(object):
    def __init__(self):
```

Ensuite le code permet d'initialiser la communication avec l'instrument via soit une communication série, soit une communication éthernet avec un socket.

Cette fonction d'initialisation va également aller lire les informations de réglages prédéfinies dans un fichier de configuration au format .json comme ceci :



```

1 [
2   "NAME":
3     {
4       "Value": "New_Focus 8742 v2.2 08/01/13 15217",
5       "Type" : "string",
6       "Desc" : "IDN Name of the Tektronix Newport8742"
7     },
8   "IP":
9     {
10      "Value": "169.254.98.36",
11      "Type" : "string",
12      "Desc" : "IP Adress of the Controller"
13    },
14   "PORT":
15     {
16       "Value": 23,
17       "Type" : "int",
18       "Desc" : "Port Used for the Communication | Don't change this value"
19     },
20   "BUFFER_SIZE":
21     {
22       "Value": 2048,
23       "Type" : "int",
24       "Desc" : "Allocated Memory Size for received message from the controller | MIN VALUE : 2048"
25     },
26   "MOTOR_1>1_V":
27     {
28       "Value": 1000,
29       "Type" : "int",
30       "Desc" : "OBI-1 Ty - Velocity Setting"
31     },
32   ],
33 ],
34 ],
35 ],
36 ],
37 ],
38 ],
39 ]

```

Figure 23: Fichier de configuration en JSON - init.json du contrôleur des moteurs

Celà permet à un utilisateur de fixer quelques données au démarrage simplement, sans avoir à ouvrir le code python et risquer de modifier des lignes de codes critiques.

Chaque driver ensuite possède des fonctions qui sont propres à chaque instrument. Ces instructions sont, la plupart du temps, en langage SCPI²⁰, voici un exemple avec la fonction pour changer la vitesse d'un moteur, implémentée dans le driver Newport8742 :

²⁰SCPI : Standard Commands for Programmable Instruments

```

871
872     # Command to the controller
873     self.SOCKET.send("{xx}VA{nn}\r\n"
874         .format(xx=Motor, nn=Value)
875         .encode())
876
877     self.Check_Errors('Set_Velocity')
878
879     if self.ERROR_STATE is False: # No Error
880         # log record + info in the console
881
882     logger.info("{Motor_ID} : "
883                 "VELOCITY SET at {Velocity} steps/s"
884                 .format(Motor_ID=Motors_Correspondances[Motor], Velocity=Value))
885
886     print("NEWPORT 8742 : INFO : {Motor_ID} : "
887           " VELOCITY SET at {Velocity} steps/s"
888           .format(Motor_ID=Motors_Correspondances[Motor], Velocity=Value))
889
890     InjectorsData['Motor{Motor_ID}Vel'.format(Motor_ID=Motor)] = str(Value)
891     self.To_DB(InjectorsData)

```

Figure 24: Extrait de code : Commande SCPI, Logfile et Ecriture BDD

Les ligne 873 à 875 composent la commande envoyée au contrôleur. Les éléments notés "xx" et "nn" correspondant respectivement à l'identifiant du moteur et à la valeur de vitesse désirée, sont des éléments passés en arguments au driver, via l'interface graphique, nous verrons ultérieurement de quelle manière. La commande SCPI est donc *xxVAnn* commande implémentée dans le contrôleur par le constructeur. Elle est envoyée en ajoutant un retour chariot et un saut de ligne, et convertie au format binaire (*encode()*).

Toutes les fonctions des drivers vont ensuite vérifier que la commande est valide, car à une autre méthode implémentée dans le driver :

```
def Check_Errors(Function):
```

Les erreurs possibles sont soit données par le constructeur, soit des limites fixées dans le code (comme par exemple une valeur limite d'accélération).

On peut ensuite voir dans cet extrait de code que si aucune erreur n'est détectée (ligne 879), on rempli un fichier "historique" ou *logfile*, chaque driver possède 3 types de fichiers de log :

- INFO : Pour les données utiles mais non critiques (par exemple la modification valide d'un paramètre)
- WARNING : Pour avertir d'une modification critique, mais non dangereuse pour l'expérience (Exemple : Valeur de vitesse nulle, Valeur dépassant un certain seuil)
- ERRORS : Modification critique et dangereuse (Exemple : Forçage d'une valeur hors-limite, Déconnexion de l'appareil)

Enfin dans les dernières lignes de code on voit la communication des données dans la base en direct (ligne 890).

Une instance de celle-ci étant toujours active, et une classe python dédiée permettant d'y écrire des données.

La particularité de chaque driver est que chaque fonction se teste seule et se documente seule.

Celà est possible grâce à deux paquets de Python :

Doctest : Un paquet permettant d'effectuer des tests de codes.

Prenons un exemple simple, je veux créer une fonction "Hello" qui prends en argument mon prénom "Alexandre", cette fonction sera alors :

```
def Hello(Name) :
    print("Hello ", Name)
```

Basique me direz-vous, *Hello(Alexandre)* renverra "Hello Alexandre", mais comment être sûr que la fonction renverra exactement ce que je souhaite ? C'est la l'utilité de doctest. Les fonctions de nos drivers étant plutôt basiques (changer des valeurs, lire des données), on peut tester nos codes sur des exemples

connus.

Reprenons, je sais que mon code doit me renvoyer "Hello Alexandre", avec doctest, qui est une librairie à importer dans Python, je peux vérifier cela :

```
import doctest

def Hello(Name) :
    """
        .. doctest::
        >>>Hello ("Alexandre")
        Hello Alexandre
    """
    print ("Hello ", Name)

if __name__ == "__main__":
    doctest.testmod( verbose=True)
```

Par l'ajout de ces 4 lignes, le test sera effectué. C'est ce qu'on appelle en python, une docstring. La librairie doctest va lire la docstring et effectuer les actions derrière les chevrons (ici donc Hello(Alexandre)) et attendre comme résultat, ce que nous lui précisons en dessous.

Deux possibilités s'offrent alors :

- 1) Le test est validé, dans ce cas encore une fois deux possibilités :
- a) La lecture n'a pas été demandée (doctest.testmod(verbose=False)), dans ce cas aucun message n'est renvoyé.
- b) La lecture a été demandée (doctest.testmod(verbose=True)), dans ce cas le programme nous renvoie un message pour indiquer que le test est valide :

```
In [6]: runfile('/home/condette/Bureau/sanstitre0.py', wdir='/home/condette/Bureau')
Trying:
    Hello("Alexandre")
Expecting:
    Hello Alexandre
ok
1 items had no tests:
    __main__
1 items passed all tests:
    1 tests in __main__.Hello
1 tests in 2 items.
1 passed and 0 failed.
Test passed.
```

Figure 25: Message de test Doctest réussi

- 2) Le test a échoué, dans ce cas, nos avons quoi qu'il en soit, un message d'erreur qui s'affiche, indiquant la ligne de l'erreur, ce qui était attendu et la réponse effective du programme.

```
In [9]: runfile('/home/condette/Bureau/sanstitre0.py', wdir='/home/condette/Bureau')
*****
File "/home/condette/Bureau/sanstitre0.py", line 4, in __main__.Hello
Failed example:
    Hello("Aelxandre")
Expected:
    Hello Alexandre
Got:
    Hello Aelxandre
*****
1 items had failures:
  1 of  1 in __main__.Hello
***Test Failed*** 1 failures.
```

Figure 26: Message de test Doctest ayant échoué

Ces docstrings sont également très utiles pour notre seconde librairie :

Sphinx : Qui permet de générer de la documentation automatiquement. Ces docstrings doivent respecter un format d'écriture, le RST²¹, voici toujours avec notre exemple Hello Alexandre, ce que celà donnerait :

```
import doctest

def Hello(Name) :
    """
        Ceci est un programme de test

        :param Name: Nom de la personne qui utilise le code
        :type Name: str

        .. doctest::
            >>>Hello("Alexandre")
            Hello Alexandre
    """
    print("Hello", Name)

if __name__ == "__main__":
    doctest.testmod(verbose=True)
```

Dans ces éléments on voit que l'on donne le nom du paramètre, ici 'Name', et on peut lui adjoindre une courte description.

Ensuite on donne le type du paramètre, ici une chaîne de caractères donc *str*.
On peut ensuite donner un exemple.

Une fois celà fait, il faut sauvegarder le programme et grâce à une ligne de commande, la documentation se générera au format html automatiquement, comme ceci :

²¹RST : ReStructured Text

Test module ¶

`Test.Hello(Name)`

[source]

Ceci est un programme de test

Parameters: `Name (str)` – Nom de la personne qui utilise le code

Voici la fonction de test, qui est valide.

Example

```
>>> Hello("Alexandre")
Hello Alexandre
```

©2020, Alexandre CONDETTE . | Powered by [Sphinx 3.1.2](#) & [Alabaster 0.7.12](#) | [Page source](#)

Figure 27: Exemple de documentation auto-générée avec Sphinx

Notre Doctest sert également d'exemple pour cette documentation.

Un exemple de documentation complète est présente en annexe du rapport.

Pour conclure cette partie, je vous propose deux illustrations, la première représentant l'arborescence globale pour un driver avec l'architecture que j'ai mise en place, et la seconde illustrant le fonctionnement global.

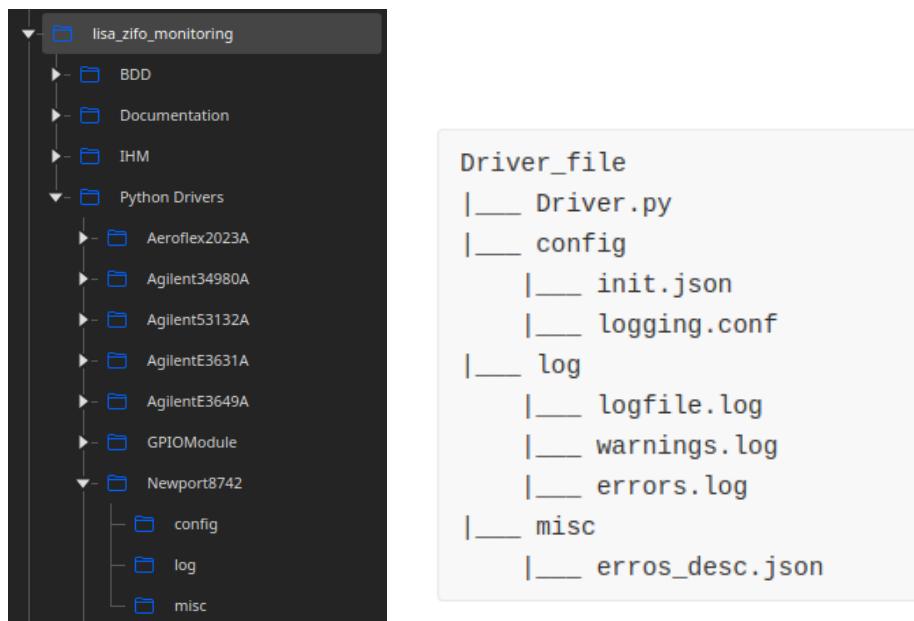


Figure 28: Arborescence des fichiers du projet de monitoring

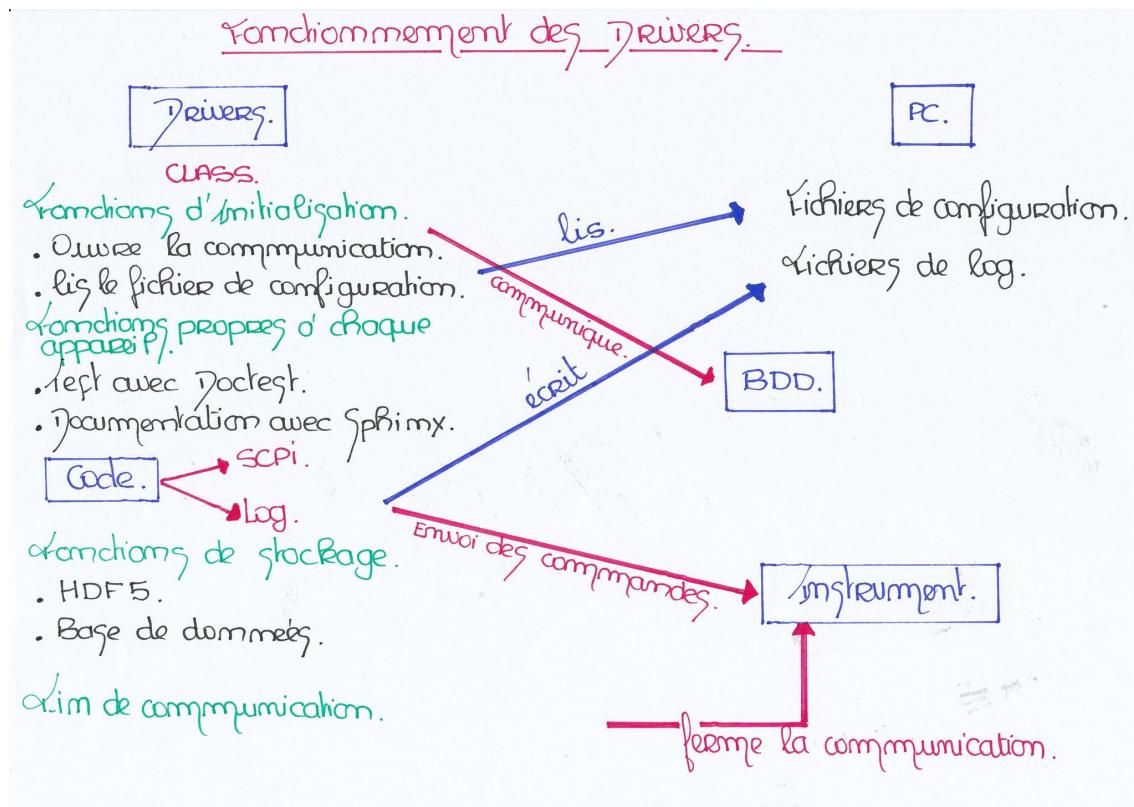


Figure 29: Fonctionnement global des drivers

3 Le stockage et la visualisation des données

Maintenant que nous avons vu le fonctionnement des drivers, nous allons nous intéresser au second choix technique que j'ai fait, quant au stockage et la visualisation des données.

Comme nous en avons déjà parlé dans les parties précédentes, les données sont stockées sous deux formes, en données brutes et sous forme de base de données où données non traitées et traitées cohabitent.

Pour l'enregistrement sous forme de fichier, le format d'enregistrement que j'ai décidé d'utiliser est le format hdf5 ou Hierarchical Data Format, permettant comme son nom l'indique, de structurer des fichiers et données en grande quantité, comme celà est le cas sur le banc de test de LISA (environ 30 Téraoctets de données sur 12 mois de test).

Celà permet de stocker les données sous forme d'arborescences, on peut également "imbriquer" les fichiers hdf5 les uns dans les autres, permettant d'obtenir une très bonne hiérarchisation des données.

Celà est possible grâce à une bibliothèque nommée *h5py*.

Cette partie bien que primordiale, n'est pas la plus intéressante. Le plus crucial pour le monitoring est la mise en place de base de données.

La base de données que j'ai décidé d'utiliser pour ce projet est une base InfluxDB qui présente de nombreux avantages par rapport à notre cahier des charges.



Dans un premier temps InfluxDB permet une capacité d'écriture à haute fréquence, suffisamment pour correspondre à la fréquence d'écriture des phasemètre (400 Hz). C'est également un système de base de données qui a déjà fait ses preuves, qui a été utilisé sur d'autres missions comme Euclid, ou au CERN.

Figure 30: Logo d'InfluxDB

C'est également un système open-source et compatible avec le logiciel de monitoring que j'ai décidé d'utiliser : Grafana.

InfluxDB possède également sa propre librairie python, se qui le rend facilement implantable, comme le montre le code que j'ai réalisé pour mettre en place la base.

Ce code est disponible en annexe de ce rapport et est une classe Python, qui comme les drivers est composée d'une fonction d'initialisation, qui crée la base. Cette dernière est créée sur l'adresse locale de notre ordinateur *127.0.0.1* sur le port 8086. Elle est donc accessible en web, via l'interface de visualisation d'InfluxDB. Mais cette dernière ne nous intéresse pas.

La fonction crée la base qui se nomme "*LISA_DB*", cette base est ensuite protégée par nom d'utilisateur et mot de passe.

La classe contient ensuite une méthode, qui s'appelle

```
def Write_in_DB(self, Data, Table):
```

Cette fonction permet de convertir des données *Data* entrée sous forme de json, composée de 3 champs à savoir *measurement*, *field* et *tags* sous un format stockable par InfluxDB, dans les tables de la base, nous avons décidé de créer une table par "type" de données (par exemple une table pour les températures, une pour la position des moteurs).

C'est cette fonction qui est appellée régulièrement dans les drivers pour transmettre les données acquises.

En discussion avec l'APC, nous avons décidé d'utiliser une seule base de données pour le moment, mais à l'avenir deux bases seront utilisées, pour distinguer données "brutes" des données traitées.

Comme dit en introduction de cette partie, InfluxDB sert de base à notre logiciel de monitoring : Grafana.

Grafana est également un logiciel opensource, qui permet via la création de "Dashboards", qui s'appuie donc sur une base de données, dans notre cas InfluxDB d'afficher nos données d'une manière plutôt optimisée.

Mais pourquoi utilisé Grafana plutôt qu'une autre interface ? Est bien car Grafana possède une spécificité : l'accès à la visualisation des données via une interface Web. Cet accès est sécurisé et on peut y accéder via 3 types de comptes que j'ai définis : Un compte Administrateur, Un compte Editeur et un compte Visualisateur. Le compte administrateur est dédié à la ou les personne(s) qui programment, afin d'apporter les modifications sur les "dashboards", comme modifier les requêtes SQL vers InfluxDB ; Le mode Editeur permet de déplacer les dashboards, tandis que le mode Viewer permet uniquement de visualiser les données.

Grafana possède également toute une batterie de plugins permettant d'avoir toutes les possibilités dont nous avons besoin sur LISA (Graphique temporels, fréquentiels, radar)

Voici des exemples de rendus Grafana sur le monitoring de LISA :



Figure 31: Logo de Grafana

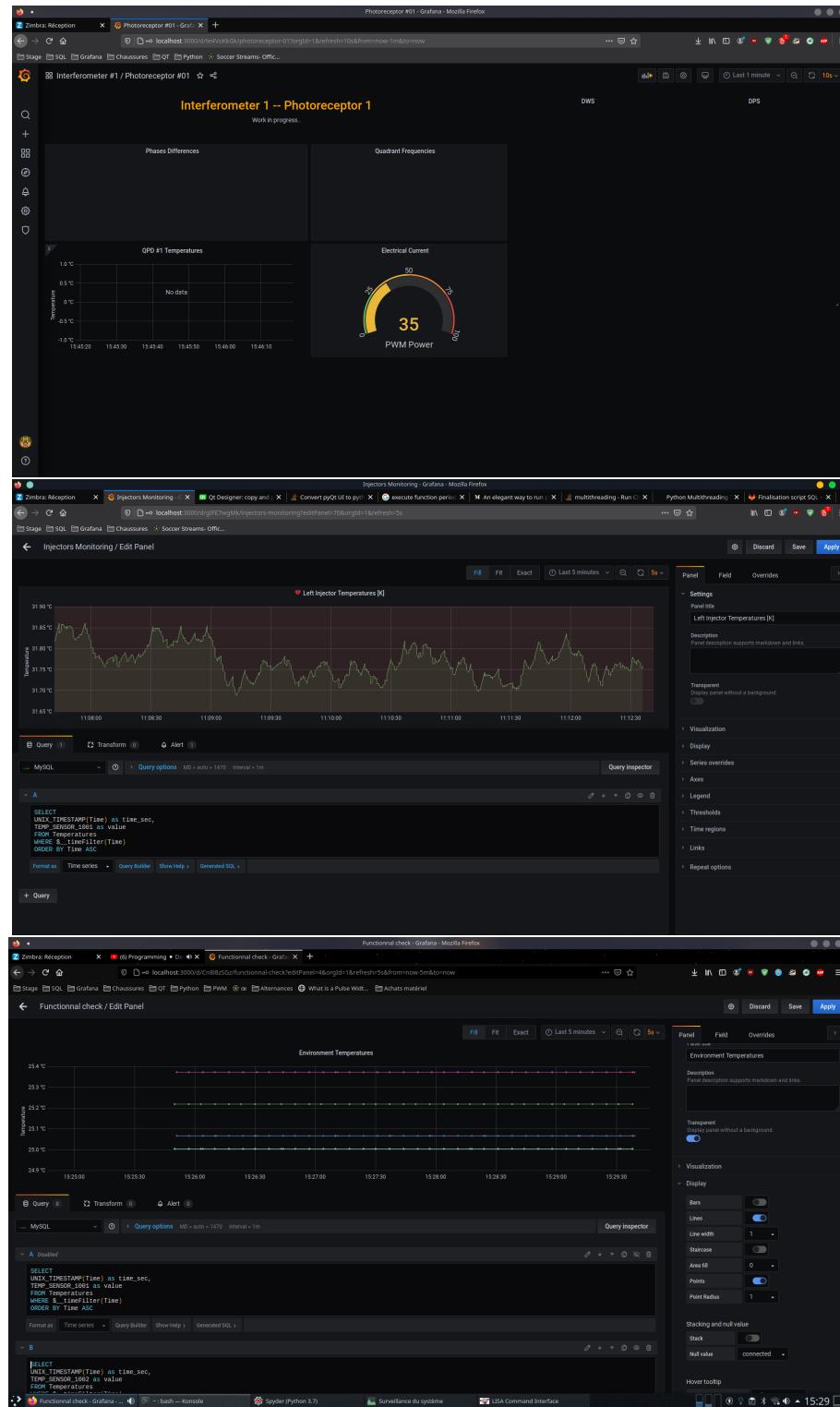


Figure 32: Extraits de la page Grafana de Monitoring du banc LISA

Un autre avantage majeur de Grafana est la gestion des alertes. Il est possible d'implémenter des seuils sur chacun des éléments de dashboards, si ce seuil est dépassé, Grafana génère une alerte, qui est envoyée par mail et par sms à un administrateur.

J'ai commencé à mettre en place ce système, et Julien Zoubian s'est chargé de la création d'un compte mail propre à la machine pour l'envoi des mails d'alertes.

La mise en page Grafana est amenée à être modifiée, en fonction des besoins de chacun.

L'interface sera accessible sur la machine de monitoring à 4 écrans, permettant de visualiser un certain nombre de données simultanément.

L'interface de monitoring se compose donc de 12 dashboards pour les photorécepteurs (1 par photorécepteur), 1 pour les photodiodes simples, 1 pour les moteurs, 1 pour le monitoring du laser, 1 pour les données environnementales et un pour la visualisation des connexions. Comme nous pouvons le voir sur cette image :

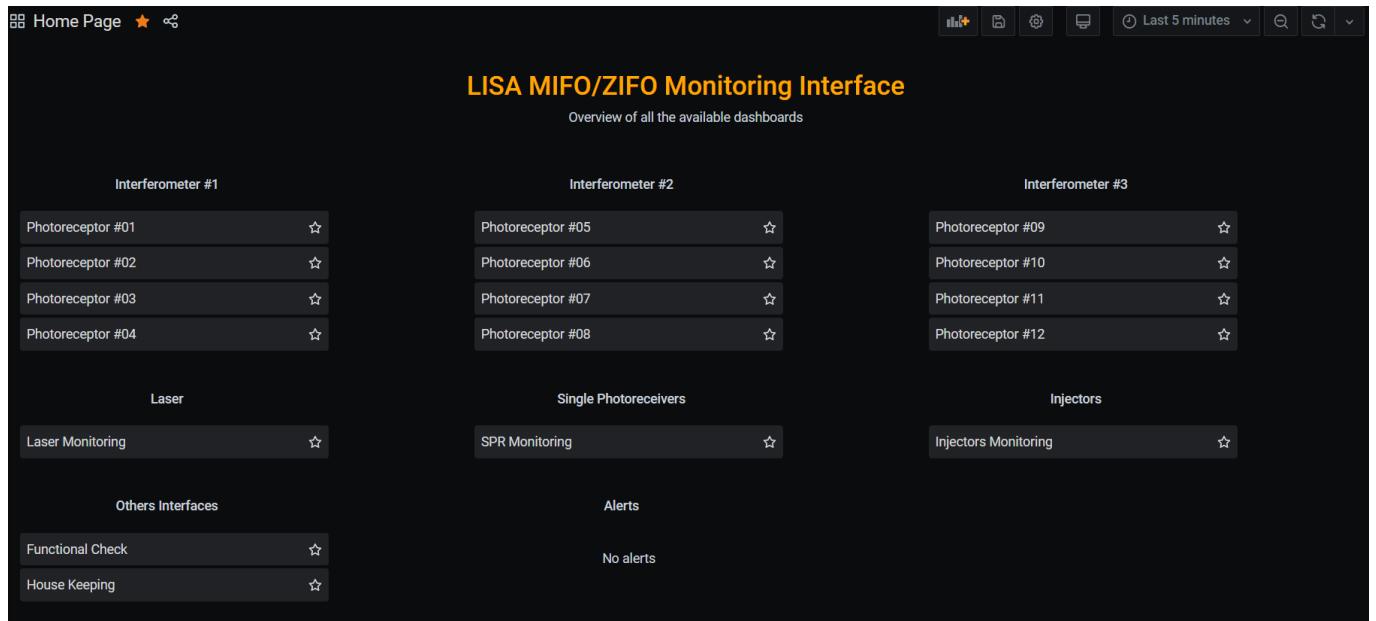


Figure 33: Page d'accueil de Grafana - LISA

Cette interface est également liée à l'interface de commande, via les drivers. Interface de commande dont nous allons maintenant voir les spécificités.

4 QtDesigner – Une interface pour les drivers

Une autre partie importante de mon travail a été la réalisation d'une interface de commande, afin de pouvoir agir sur certains équipements comme les contrôleurs des moteurs.

Cette interface est la possibilité de faire le lien entre l'utilisateur et les drivers créés précédemment. Pour la réaliser, j'ai décidé d'utiliser le programme QtDesigner, permettant de créer ses propres outils d'interface (comme des boutons, des zones de saisie de texte etc.). Cet outil est extrêmement pratique car il évite des tâches fastidieuses comme avec tkinter intégré à Python, où il faut placer et dimensionner ses "widgets" via le code. QtCreator avec QtDesigner intégré sur distribution Linux, permet de faire tout cela via une interface graphique, et est donc plus facile à mettre en oeuvre.

Je remercie également Aurélia Secroun de m'avoir aidé à prendre en main ce logiciel.

Un autre atout majeur de QtDesigner est qu'il permet de directement faire le lien entre l'interface est des signaux Python, ces derniers pouvant être associés à des fonctions python.

Comme pour l'explication, je vais faire comme pour la conception des drivers, partir d'un exemple simple, et ensuite je vous montrerai le résultat de l'interface graphique que j'ai créée.

Voici comment se présente l'interface :

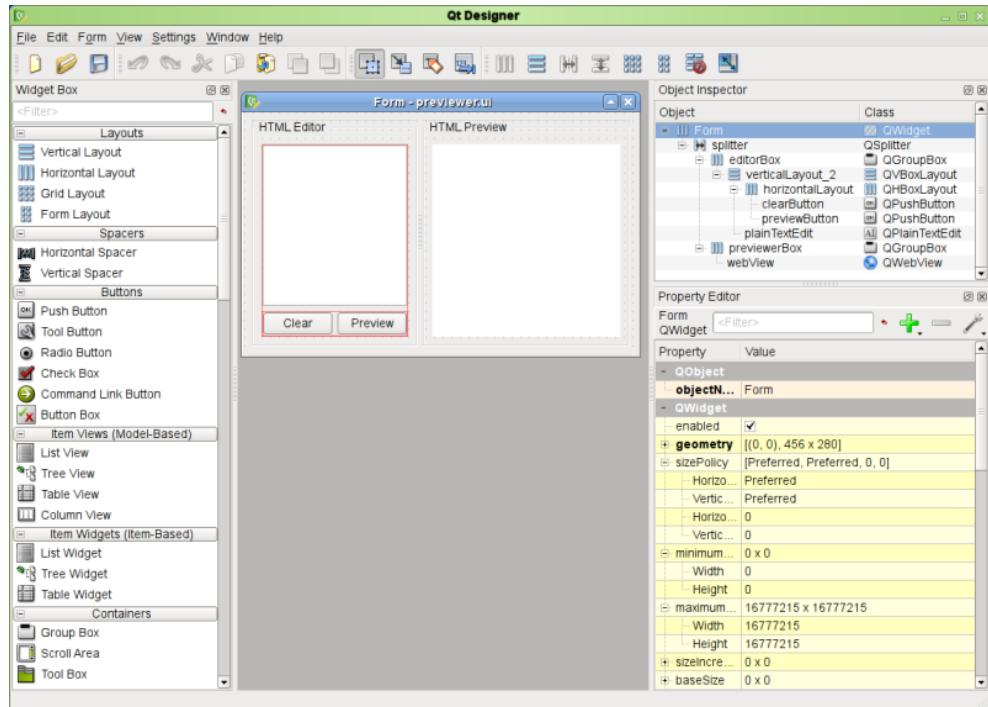


Figure 34: Page d'accueil de QtDesigner

La réalisation d'un bouton est donc extrêmement optimisée, au lieu de 4 ou 5 lignes de codes à réaliser, il suffit de "glisser-déposer" l'outil "Push Button".

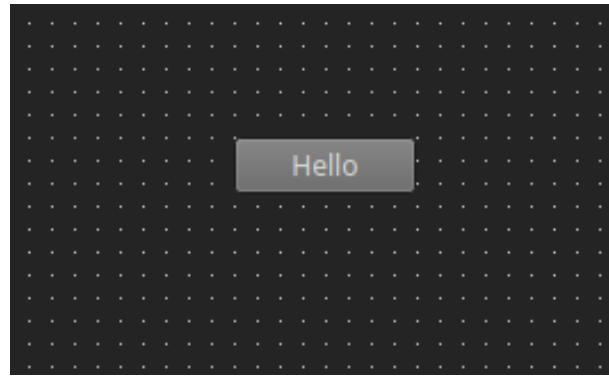


Figure 35: Pushbutton dans Qt

Voici donc notre bouton "Hello" qui va faire appel à notre fonction "Hello" de la section 3.1. Pour ce faire il suffit de créer un signal et de lui donner le nom de notre fonction, ici "Hello()".

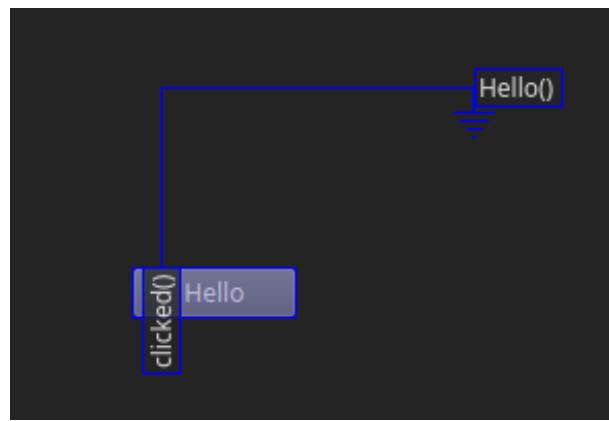


Figure 36: Pushbutton dans Qt

QtDesigner nous fourni ainsi un fichier .GUI qui est ensuite exportable en .py ce fichier .py constituera donc ce qu'on pourrait qualifier de "frontend" de notre application, donc l'ensemble des wigdets que l'on a placés.

Reprendons ensuite notre fonction "Hello()" qu'il faut maintenant modifier légèrement pour l'intégrer à notre application.

```

1 import doctest
2 import app as App
3 import sys
4
5 from PyQt5.QtWidgets import QApplication, QMainWindow
6
7 class HelloApp(QMainWindow):
8     """
9         The purpose of this class is to create the graphical user interface for the LISA MIFO / ZIFO test bench.
10    It will also make the link between the .ui and the backend functions.
11    """
12
13
14    def __init__(self, parent=None):
15
16        # Create the interface and all the widgets defined in "Control.ui.py"
17        super().__init__(parent)
18        self.ui = App.Ui_Dialog() # Create an instance of the interface
19        self.ui.setupUi(self) # Create all the buttons, labels and other Qt objects
20
21    def Hello(self, Name) :
22        """
23            Ceci est un programme de test
24
25            :param Name: Nom de la personne qui utilise le code
26            :type Name: str
27
28            Voici la fonction de test, qui est valide.
29
30            **Example**
31            :::
32            >>> Hello("Alexandre")
33            Hello Alexandre
34
35        """
36        Name = 'IPSA'
37        print("Hello", Name)
38
39 if __name__ == "__main__":
40    doctest.testmod(verbose=False)
41
42    app = QApplication.instance()
43    if app is None:
44        app = QApplication(sys.argv)
45    main_window = HelloApp()
46    main_window.show()
47
48    try:
49        app.exec_()
50    except KeyboardInterrupt():
51        print(0)
52        app.quit()
53

```

Figure 37: Code permettant de lier l'application à la fonction

Ces quelques modifications permettent, lorsque l'on appuie sur le bouton d'automatiquement lancer la fonction "Hello(IPSA)".

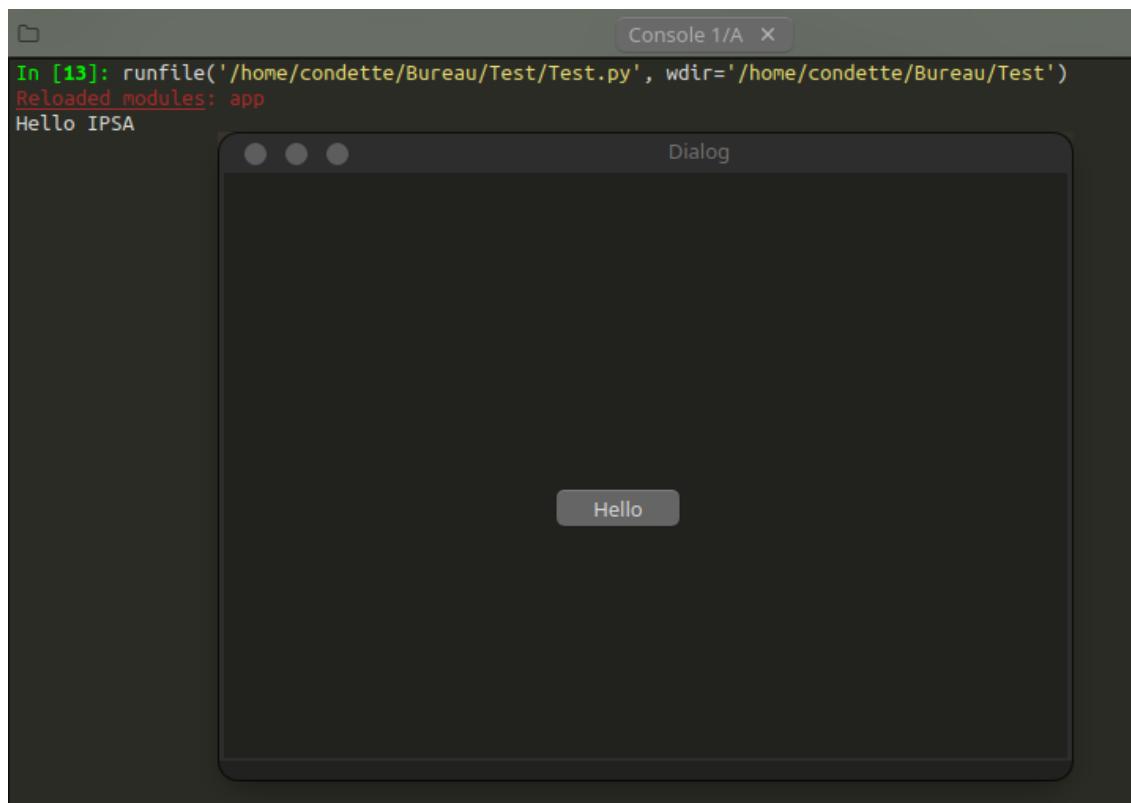


Figure 38: Résultat

Voilà pour la base du fonctionnement de Qt, mais ce qui nous intéresse ici, c'est la manière dont j'ai réalisé cette interface. Dont le rôle principal est le contrôle des injecteurs des moteurs. Cette interface permet également de démarrer la communication avec chacun des instruments du banc individuellement, de lancer et d'arrêter l'acquisition des données et voici à quoi elle ressemble :

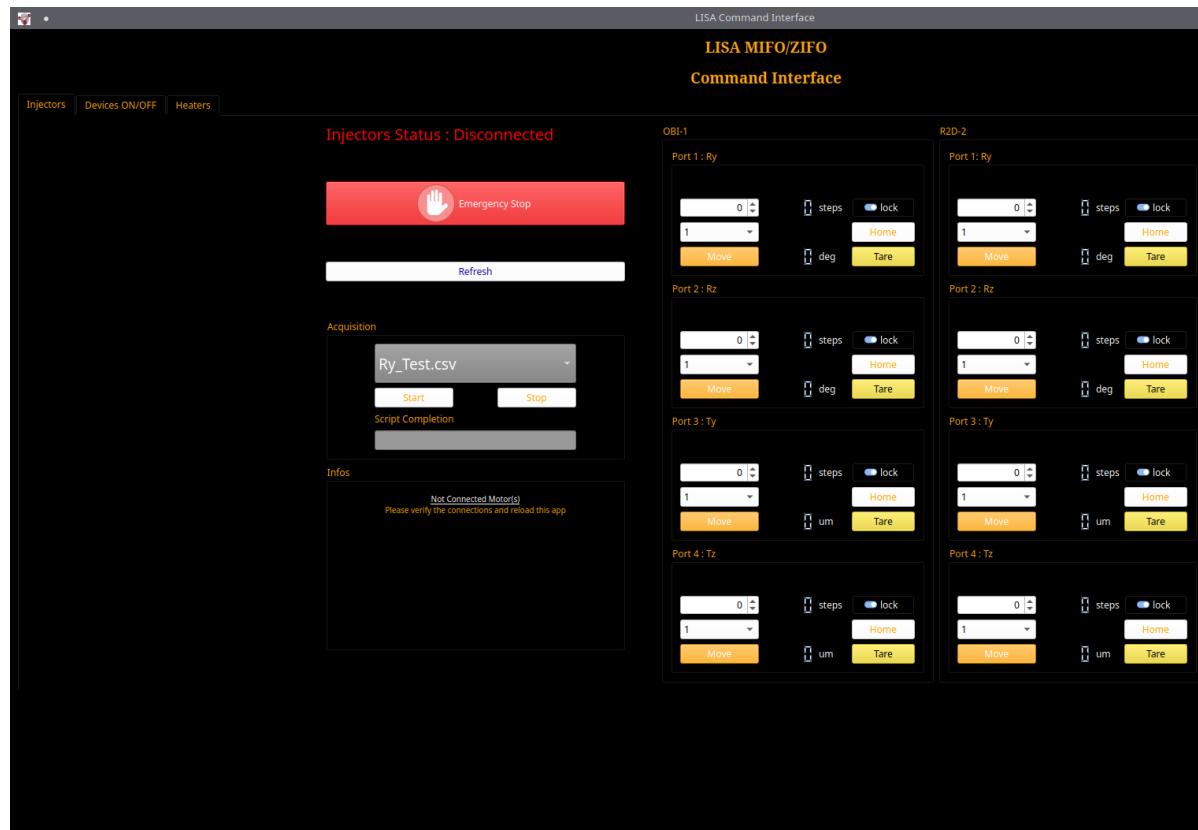


Figure 39: Interface de commande des moteurs - Alexandre CONDETTE

On voit donc qu'à la fin de mon stage cette interface est composée de 3 onglets, dont nous observons ici le premier, consacré aux injecteurs, réalisé avec la supervision de l'APC (Hubert Halloin, Thomas Zerguerras) et du CEA qui s'occupe de la réalisation des injecteurs. Cette partie de l'interface permet donc d'indiquer au moteur la position absolue où l'on souhaite qu'il se rende et la taille des pas du moteur qu'il devra effectuer (ici suivant les recommandations du CEA j'ai fixé comme valeurs de pas 1, 10, 100 et 1000).

Chaque moteur (il y a 4 moteurs (2 Rotations, Ry, Rz et 2 Translations Ty Tz) sur chacun des deux injecteurs OBI-1 et R2-D2) possède également un bouton de retour à sa position d'origine "Home" ainsi qu'un bouton "Tare" pour définir une nouvelle position "zéro".

La position actuelle est indiquée en nombre de pas et en microns ou en degrés suivant la nature du moteur. Cette conversion est à affiner en fonction des études du CEA sur la longueur des pas de chaque moteur, de plus ces derniers ne possèdent pas d'encodeurs, cette position est approximative, mais assez précise pour rester valable sur l'ensemble de la manipulation.

Pour ne pas perturber le fonctionnement des contrôleurs, j'ai ajouté des sécurités, empêchant de déplacer 2 moteurs simultanément, car cela pourrait causer des interférences sur les contrôleurs et provoquer une mise en sécurité du système.

Toutes les informations de positions sont également transmises à l'interface de monitoring Grafana. Par exemple :

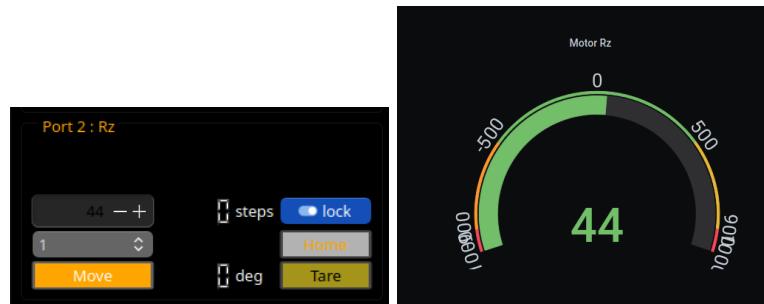


Figure 40: La liaison IHM - Grafana

Le rendu n'est pas des meilleurs suite à l'utilisation d'une autre version d'Ubuntu pour la visualisation

L'interface des moteurs permet également de lire et d'exécuter des scripts de test, permettant d'automatiser les déplacements des moteurs (pour leur faire faire un déplacement sinusoïdal par exemple) cette fonctionnalité est opérationnelle mais mériterait une amélioration (car pour l'instant elle ne permet que de faire des déplacements linéaires basiques).

Le deuxième onglet de cette interface est donc l'onglet qui nous permet d'initier la communication avec chacun des appareils :

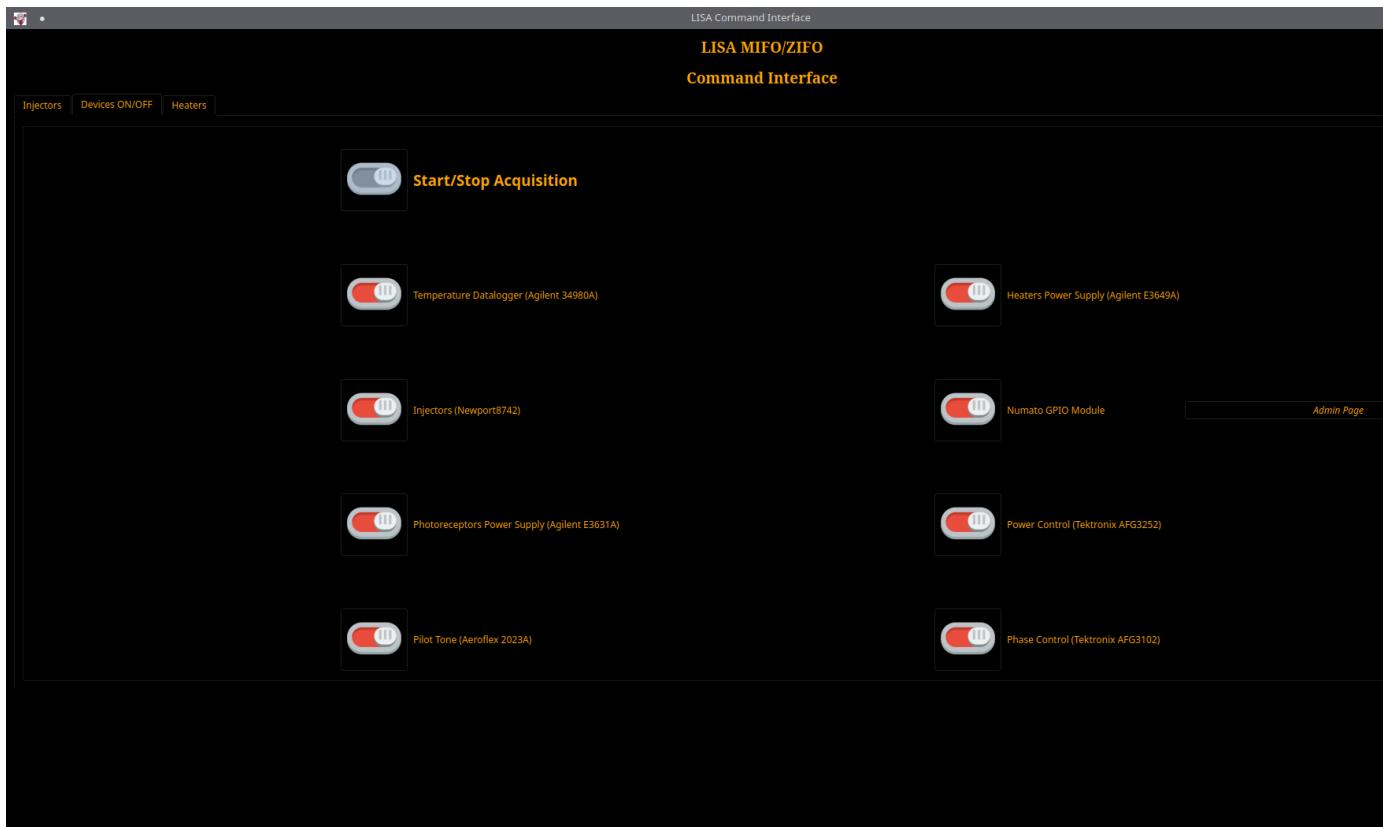


Figure 41: Page de commande des connexions

Cette page est assez simple, derrière chaque bouton, se trouve juste la fonction d'initialisation de chaque classe d'appareil. Par exemple pour le Newport8742, le bouton Newport8742 est relié à la fonction :

```
import Newport8742 as NP
MyNewport = NP.Newport8742()
```

Et de même pour chacun des appareils.

Enfin la dernière fonctionnalité de cette interface est le contrôle des résistances chauffantes du banc de test, afin de pouvoir faire des essais thermiques sur le banc et étudier la réponse du signal interférométrique à des fluctuations de températures.

Ces résistances chauffante sont alimentées en 0 et 13V grâce à un système de PWM²², le principe est simple, nous avons une alimentation qui fonctionne en "tout ou rien", qui délivre donc soit 0V soit 13V, pour obtenir toutes les valeurs de tensions entre ces deux valeurs on utilise donc le PWM, on met le signal à 1 (1 équivalent à la position 13V) à une fréquence fixe, en contrôlant le rapport cyclique (la période pendant laquelle le PWM est en position "1" sur la période totale). La valeur moyenne de tension sur un cycle étant alors :

$$V_M = V_{Max} * \eta$$

Avec $V_{Max} = 13V$

$\eta = T_{haut}/T$ le rapport cyclique (la proportion de temps en état 1 sur la période)

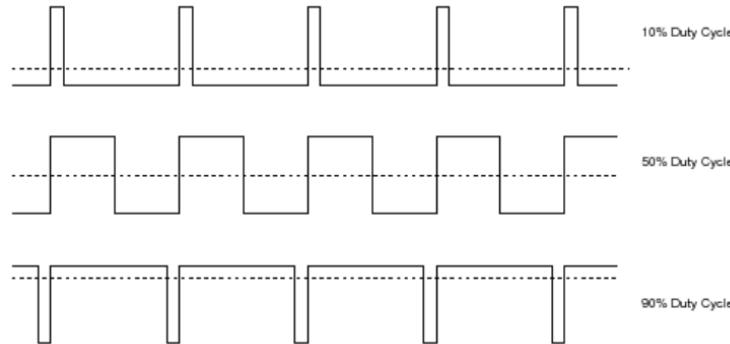


Figure 42: Illustration du PWM

Cette partie de l'interface permet donc de fixer la valeur du rapport cyclique soit la tension appliquée, et ainsi modifier l'intensité de la résistance chauffante et contrôler sa température

²²PWM : Pulse Width Modulation

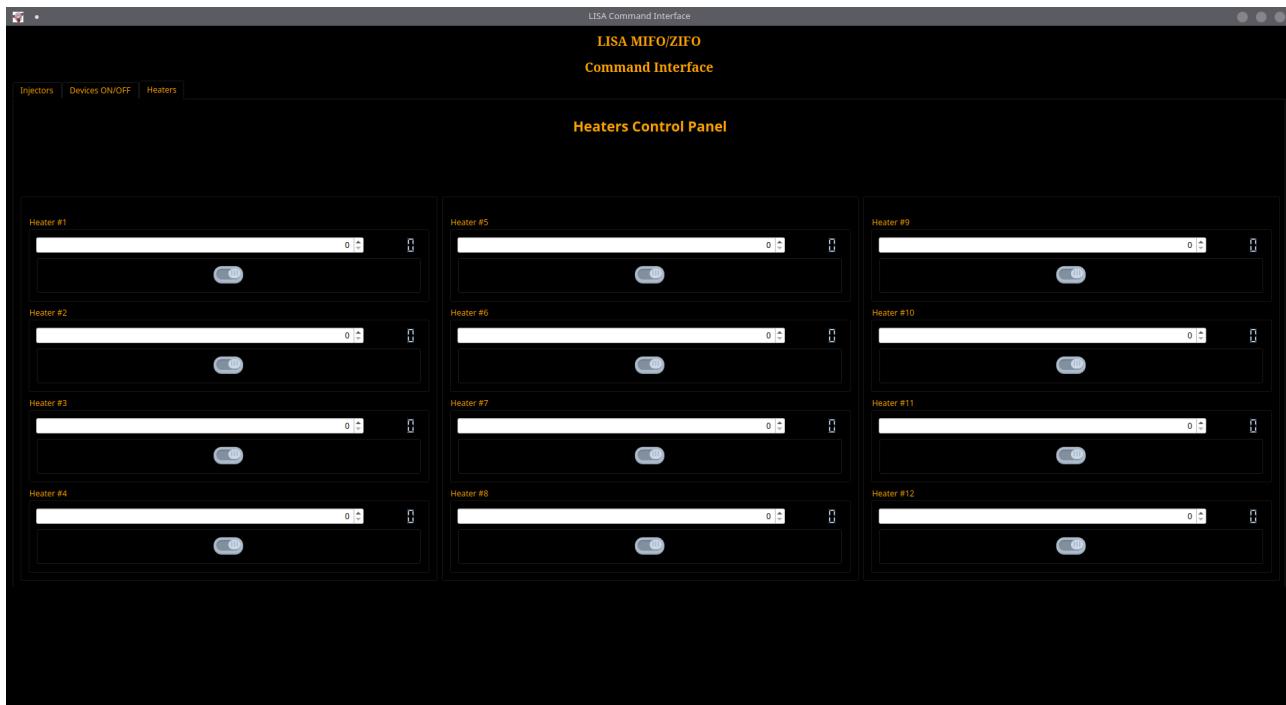


Figure 43: Page de réglage des résistances chauffantes

Maintenant que nous avons vu tous les éléments que j'ai mis en place, il est temps de faire un bilan, et j'ai pu moi même en faire un lors de mon avant dernière semaine de stage, en me rendant à l'APC pour tester les drivers et l'interface sur tous les appareils présents la bas. Car oui je ne l'ai pas encore précisé, mais tous les instruments contrôlés, à l'exception des contrôleurs des moteurs, se trouvaient à l'APC à Paris. Il a donc fallu faire le développement à distance, via un accès VPN. une contrainte supplémentaire.

Part IV

Dernière Visite à l'APC et Bilan

1 Le Phasemètre | Faire la mesure du signal interférométrique

1.1 La Méthode de Heinzel

Un des éléments les plus problématiques de ce banc de test a été le phasemètre du banc. Un instrument développé par l'APC suivant un modèle de l'AEI²³. Ce module est basé sur une carte FPGA²⁴ Xilinx Virtex-6 ML605 et possède actuellement un code en C permettant de récupérer les données sous un format binaire, dans un domaine temporel. Voici à quoi ressemble ce phasemètre :



Figure 44: Phasemetre APC-LISA

Le système est basé sur des boucles d'asservissement proportionnelles-intégrale (PI) et son code est écrit en VHDL²⁵ (Code AEI)

²³Albert Einstein Institute

²⁴FPGA : Field-Programmable Gate Array

²⁵VH(SIC)DL : Very High Speed Integrated Circuit Hardware Description Language

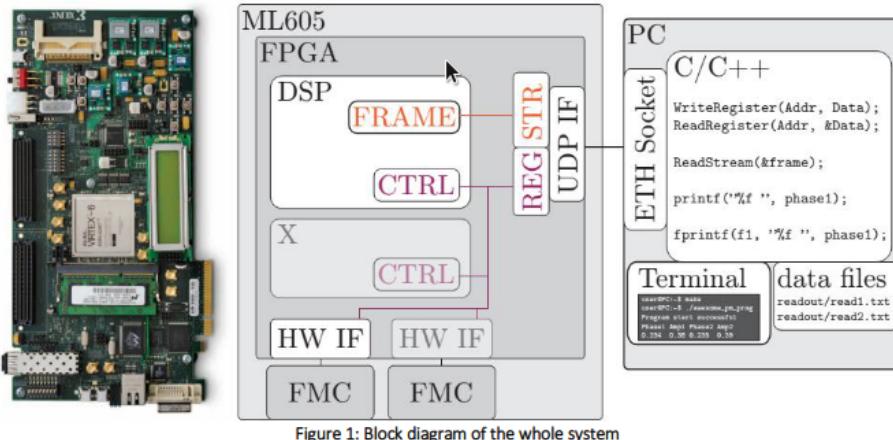


Figure 45: Phasemetre APC-LISA - Schéma bloc

Et voici les données telles qu'elles sont reçues sur l'ordinateur via le code C, le phasemètre n'étant relié à aucun photodétecteur pour le moment, on observe uniquement le bruit de phase et notamment un bruit très particulier le "jitter", bruit induit par les convertisseur analogiques/numériques²⁶.

```
int meta[1] freq_a[Hz] freq_b[Hz] freq_c[Hz] freq_d[Hz] freq_e[Hz] freq_f[Hz] freq_g[Hz] freq_h[Hz]
i_b[V] i_c[V] i_d[V] i_e[V] i_g[V] i_h[V] q_c[V] q_d[V] q_e[V] q_f[V] q_g[V] q_h[V]
563740672 +10713889.55936186574399 +7857247.71870134584606 +10713837.45585267804563 +10713837.45585267804563
+10714649.39396664313972 +10714649.39396664313972 +10714114.81512999907136 +10714114.81512999907136
+0.00024176729599 +0.00026391188259 +0.00026391188259 +0.00017009096551 +0.00017009096551
+0.00015744828942 +0.00015744828942 +0.00000076429276 +0.00000034632016 +0.00000043389537 +0.00000043389537
-0.00000054933542 -0.00000054933542 -0.00000073642792 -0.00000073642792
563806208 +10714773.65773769468069 +7857331.57236096728593 +10714026.58504103124142 +10714026.58504103124142
+10714106.14704394526780 +10714106.14704394526780 +10713631.81649684906006 +10713631.81649684906006
+0.00009911523673 +0.00013250129603 +0.00015011187506 +0.00015011187506 +0.00015104335686 +0.00015104335686
+0.00017246743831 +0.00017246743831 -0.00000061700718 -0.000000008757521 +0.000000124197574 +0.000000124197574
+0.00000050554782 +0.00000050554782 -0.00000022291872 -0.00000022291872
563871744 +10713985.09194768592715 +7857464.62242491170764 +10713969.03143412806094 +10713969.03143412806094
+10714546.94492545910180 +10714546.94492545910180 +10713977.68877490796149 +10713977.68877490796149
+0.00018378056345 +0.00024209371269 +0.00026340235409 +0.00026340235409 +0.00017025815455 +0.00017025815455
+0.00015622621715 +0.00015622621715 +0.00000001990346 +0.00000001990346 +0.000000127382127 +0.000000127382127
-0.0000003385877 -0.0000003385877 -0.00000078419622 -0.00000078419622
563937280 +10714683.23412499763072 +7857206.58661475125700 +10714273.03710219636559 +10714273.03710219636559
+10714305.63559857010841 +10714305.63559857010841 +10713673.46199217624962 +10713673.46199217624962
+0.00010078712714 +0.00013282771723 +0.00014821308523 +0.00014821308523 +0.00015105927963 +0.00015105927963
+0.00017173499108 +0.00017173499108 -0.0000007165245 -0.0000007165245 +0.000000113051638 +0.000000113051638
+0.00000069264031 +0.00000069264031 +0.00000077623484 +0.00000077623484
```

Figure 46: Données binaires du phasemètre lues en fichier texte

Deux mission principale m'étaient confiées sur cette instrument, la première était au niveau du traitement des données. En effet, les données reçues par le phasemètre sont dans le domaine temporel, et on souhaiterait les obtenir dans le domaine fréquentiel afin d'en calculer la DSA²⁷, c'est à dire la racine carrée de la DSP²⁸ représentant la répartition sur le domaine de fréquence de LISA [0.1 mHz - 1Hz] de l'amplitude pour la DSA et donc de la puissance pour la DSP, du signal.

Pour ce faire la méthode habituelle est d'utilisé les transformées de Fourier, puisque la DSP est la transformée de Fourier de la fonction d'autocorrélation... Pour simplification notons simplement que si s est notre signal, t le temps d'intégration de ce signal, sa DSP vaut :

$$DSP_s = |TF(s)|^2 * t$$

Où $TF(s)$ est la transfromée de Fourier du signal.

Python possèdant des bibliothèques pour faire celà seul, il nous permettrait de relativement facilement effectuer les transformations et estimer la DSP. Mais il y a un soucis.

Via les méthodes usuelles, à base fréquence le signal est relativement cohérent, car il y a peut de points, mais plus on augmente la fréquence, plus il y a de points, le temps d'intégration n'évoluant pas au fur et à

²⁶CAN : Convertisseur Analogique / Numérique

²⁷DSA : Densité Spectrale D'Amplitude

²⁸Densité Spectrale de Puissance

mesure, on obtient donc un signal complètement bruité dans la bande de fréquence qui nous intéresse comme le montre le graphique suivant :

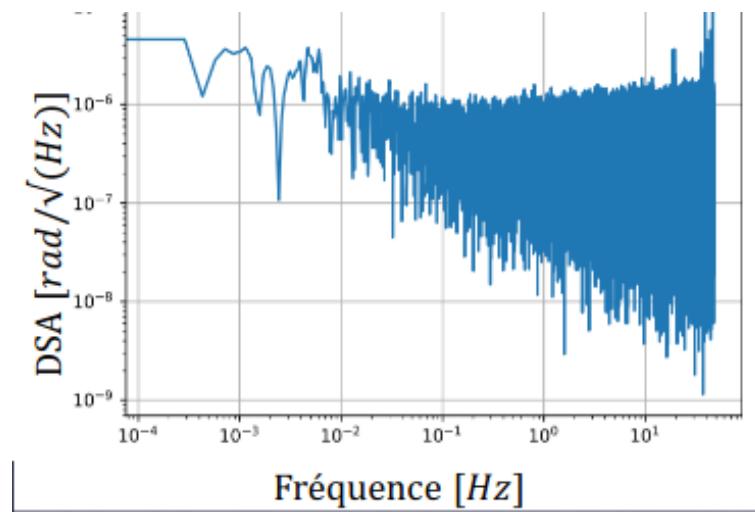


Figure 47: Bruit relatif du phasemètre - Méthode Usuelle

Une méthode a donc été développée pour contrer cet effet, une méthode basée sur la "Méthode de Welch" qui permet de faire la transformée de Fourier du signal par segment. Cette méthode modifiée s'appelle méthode de Heinzel, et est une méthode de Welch où la fréquence d'intégration varie en fonction du "segment", ainsi on a un temps d'intégration qui est optimisé pour chaque point du signal, permettant de réduire le "bruit" induit par la transformée de Fourier "Classique" comme nous l'avons vu précédemment.

Pour expliquer la méthode de Welch, il s'agit d'un algorithme :
On segmente notre signal, les segments peuvent et doivent se chevaucher pour réduire l'erreur.

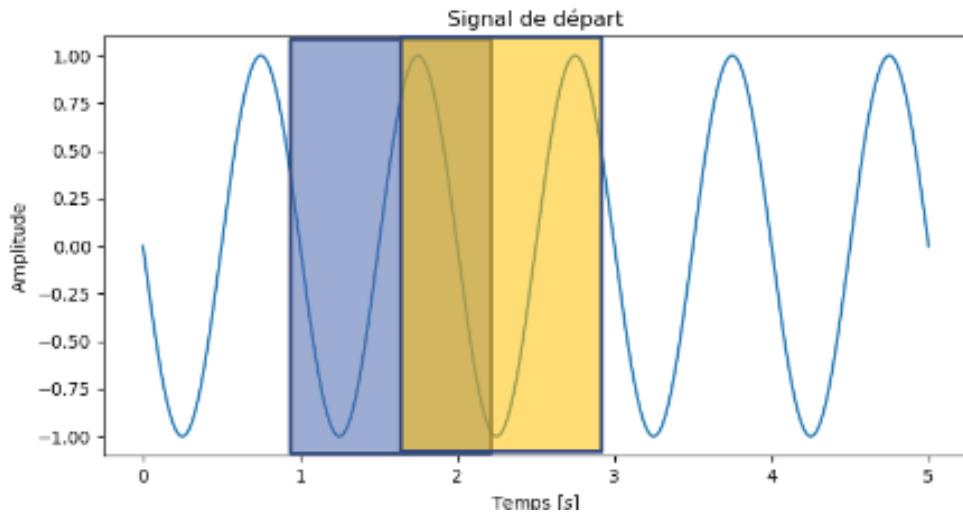


Figure 48: Méthode de Welch - Segmentation

On applique ensuite sur chaque segment un fenêtrage, par exemple un fenêtrage de type rectangulaire, pour isoler chaque partie du signal :

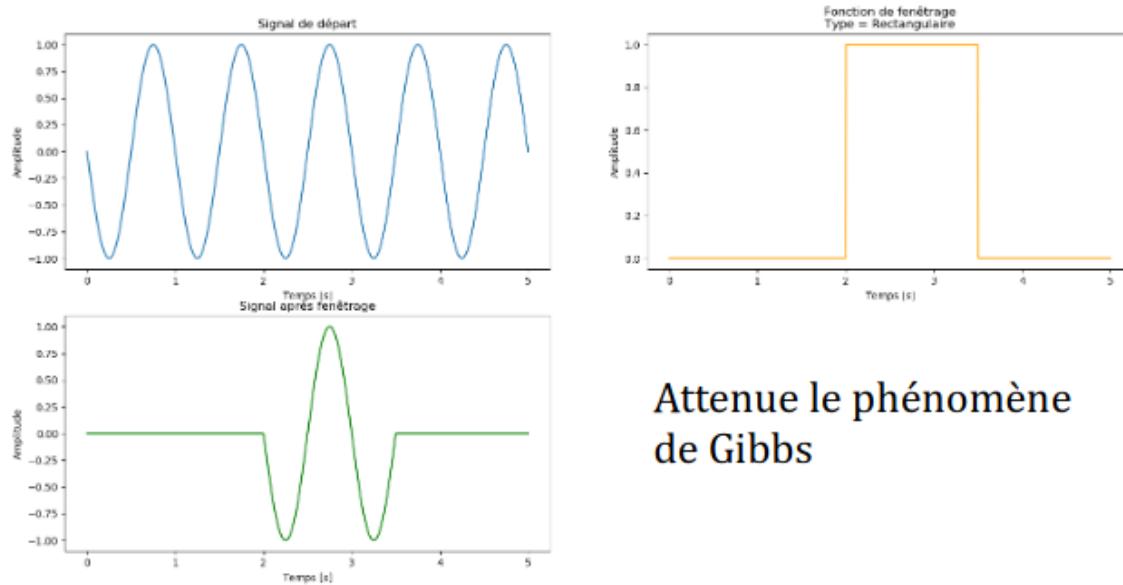


Figure 49: Méthode de Welch - Fenêtrage

Cela permet d'atténuer le phénomène de Gibbs, soit "l'effet de bords", des petites oscillations qui peuvent survenir sur le signal.

Ensuite sur chaque segment on fait la transformée de Fourier.
La moyenne des résultats permet d'estimer la DSP. Ensuite la DSA est calculée comme la racine carrée de cette DSP.

Il me fallait donc implémenter cette fonction en python, mais compte tenu des délais de mon stage je n'ai pas pu le faire avec python, néanmoins j'ai pu avec Matlab étudier quelques jeux de données avec cette méthode, et on voit que le signal obtenu est nettement plus lisible :

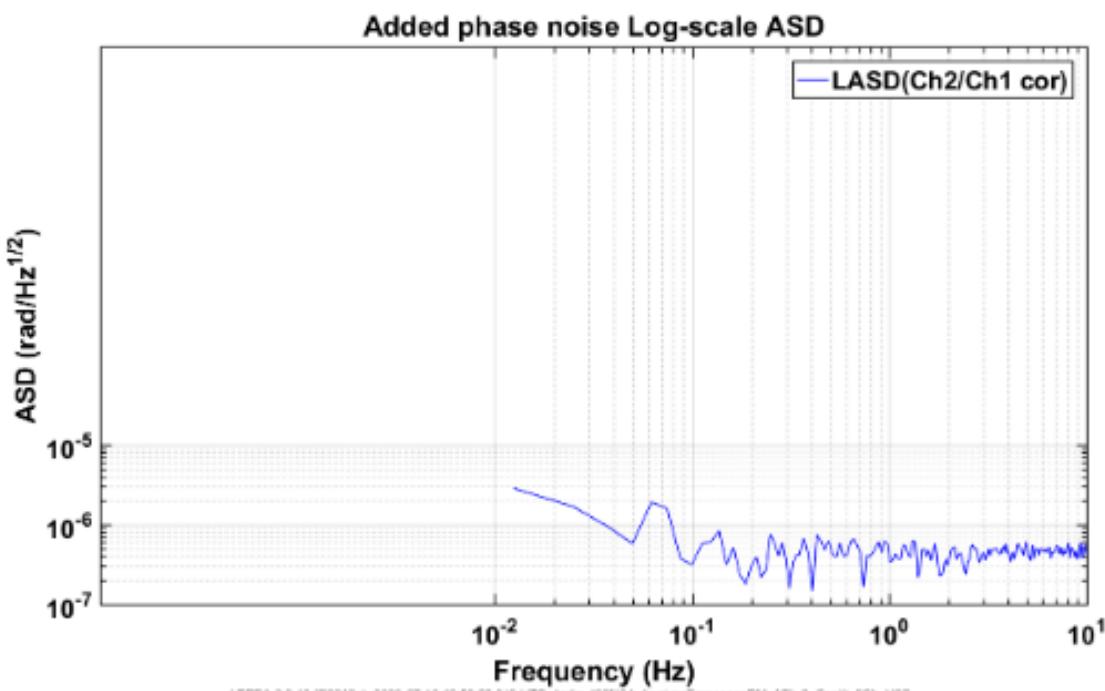


Figure 50: Bruit relatif du phasemètre - Méthode Heinzel Matlab
Le jeu de données est le même que celui utilisé sur la figure XX pour comparaison

1.2 Lire les données

La deuxième mission que j'avais sur ce phasemètre était :

- De lire le code C via Python
- Modifier le code afin de permettre l'utilisation de plusieurs phasemètre simultanément, et envoyer les données (en brut, donc format non binaire) dans la base InfluxDB.

Cette deuxième mission est à moitié réussi pour moi, car la première fonction est opérationnelle, il est possible de lire le code C via Python grâce à une technique dite de "wrapping" ; En d'autre terme, j'ai changé le mode de compilation du code C pour qu'il génère des fichiers "objets" (.o), ces fichiers sont ensuite lisibles par Python par une bibliothèque "*Cython*" qui permets d'exécuter ces fichiers comme des modules python à part entière. Encore une technique de programmation que j'ai apprise lors de ce stage.

Quant à la modification du code C, comme vous pouvez le voir sur la photographie suivante, il est constitué de nombreux sous-fichiers

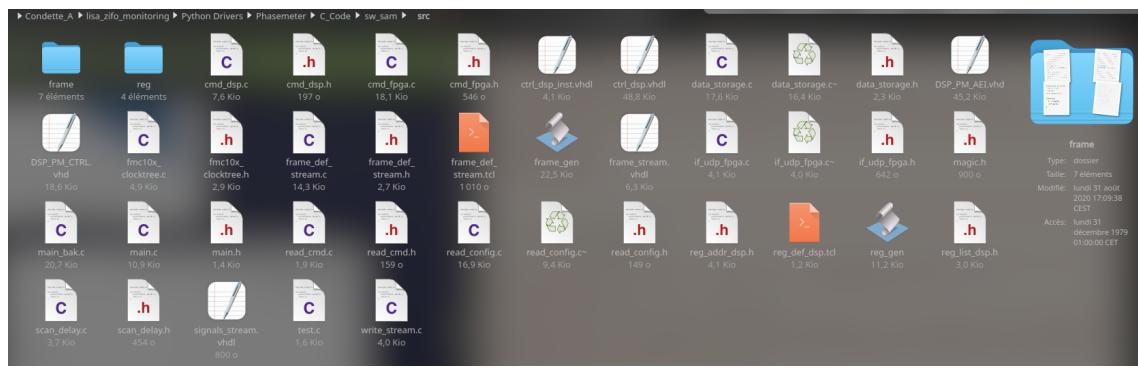


Figure 51: Fichiers du code C

Ces codes sont des codes tirés de l'AEI, et sont très mal commentés. Bien que messieurs Pierre Prat et Samuel Sube de l'APC et du CEA m'ont aidé à en comprendre une majorité, ils ont eux aussi des soucis de compréhension du code pour le moment, de plus il n'y a qu'un seul phasemètre de disponible actuellement.

Lors de ma dernière visite à l'APC en Juillet, j'avais emporté des codes pour tester une communication avec le phasemètre, mais des problèmes techniques liés notamment à l'OS (j'utilise Ubuntu 20.04, et le code du phasemètre a été réalisé sur CentOS7) et de paquets (notamment mon compilateur g++ qui ne fonctionnait plus), il ne nous a pas été possible de tester tout cela. Néanmoins Thomas Zerguerras à pu sur sa machine accéder au phasemètre et lire les données, ce qui reste encourageant.

Je suis néanmoins déçu de ne pas avoir pu travailler plus longuement sur ce point, assez complexe. Mais j'ai pu lors de mon dernier mois de stage avoir "sous ma responsabilité" un stagiaire de 4e année de l'ESTACA, Kélian Sommer, à qui j'ai pu transmettre tous les programmes et toute la méthodologie de travail pour poursuivre ce que j'avais commencé.

Nous avons pu ensemble tester nos codes sur de nombreux instruments à l'APC, ce que nous allons voir maintenant dans cette dernière partie, qui me permettra d'effectuer un bilan technique par la même occasion.

2 Le test global des drivers

Voici ce que je pourrais qualifier de concrétisation de mon stage ; La dernière visite à l'APC a été l'occasion pour Kélian (le stagiaire de l'ESTACA) et moi, de tester tous nos drivers, et la connexion entre l'IHM²⁹, les drivers, la base de données et Grafana.

Parmis les éléments à tester nous avions les suivants :

- Les contrôleurs des moteurs Newport8742
- Alimentation Agilent3631A
- Alimentation Agilent3649A
- Générateurs de fréquences tektronix AFG3102 / AFG3252
- Synthétiseur Aeroflex 2023A

Voici quelques photos pour vous représenter ces instruments

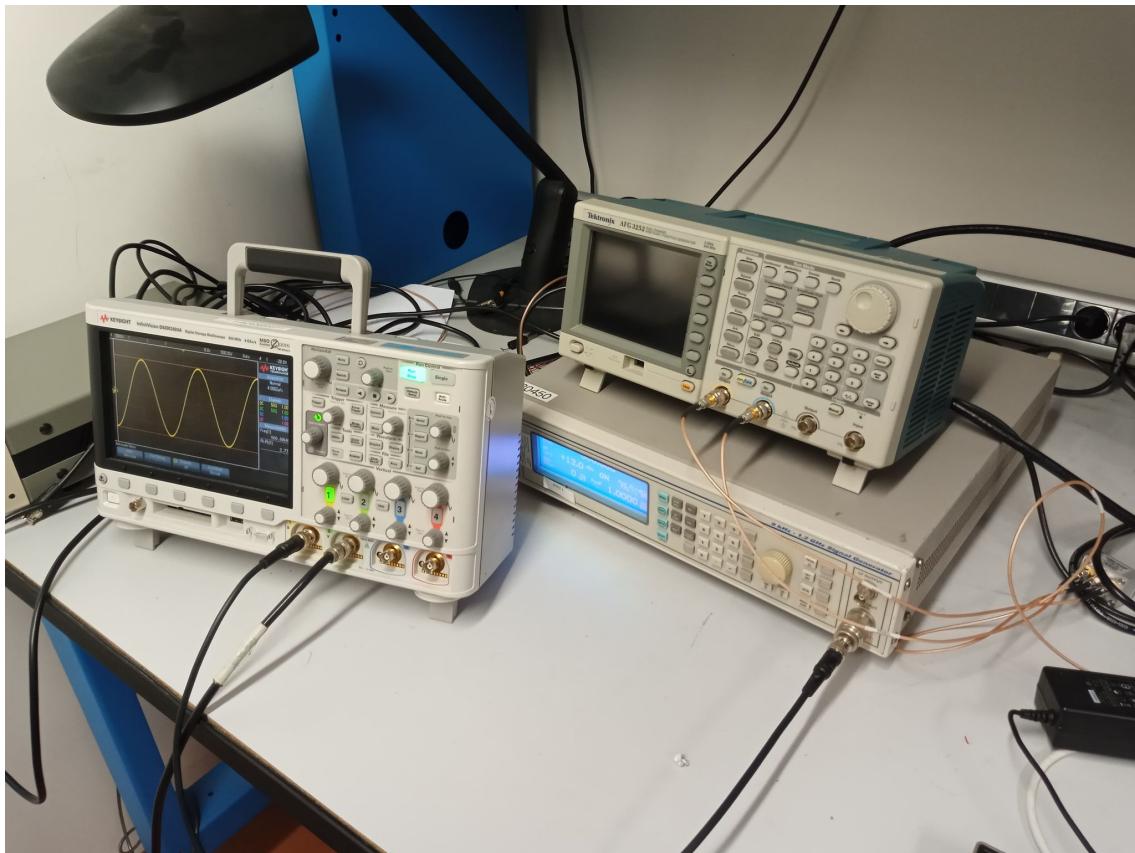


Figure 52: Un oscilloscope, un générateur de fréquence et un générateur de tension

²⁹IHM : Interface Homme-Machine



Figure 53: Un Générateur de Tension 2 sorties Agilent

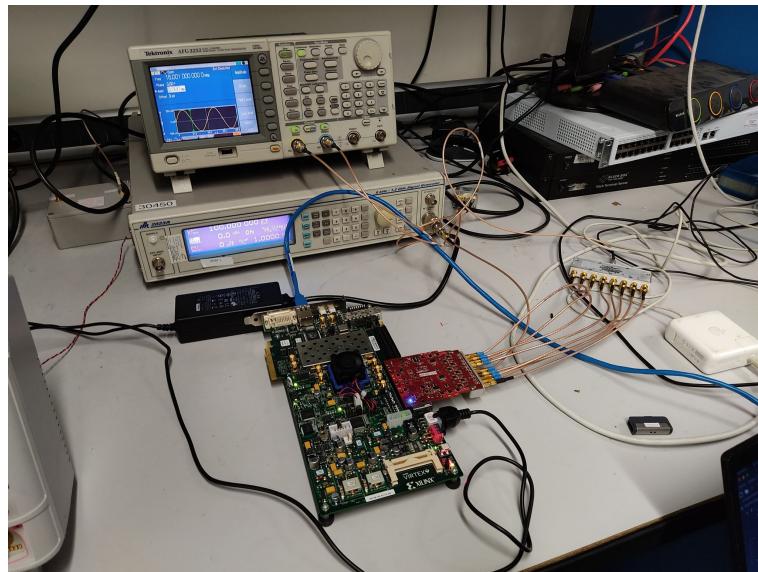


Figure 54: L'installation du phasemètre



Figure 55: Un générateur textronix



Figure 56: Une carte Numato 32 Sorties GPIO

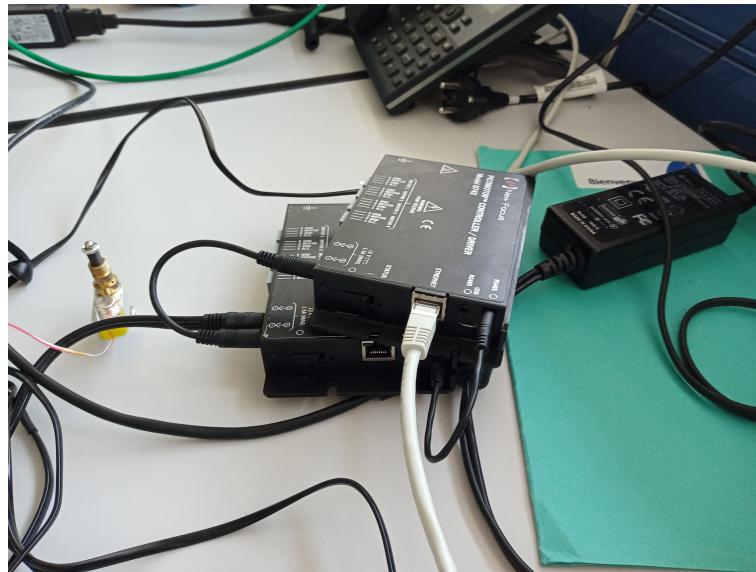


Figure 57: Les contrôleurs des moteurs Newport

Cette batterie de test a été très concluante pour nous, mettant en lumière quelques soucis que nous avions, notamment via les instruments Tektronix qui ne pouvait communiquer via PyVisa, mais via un autre langage, le VIX11, ne modifiant quasiment rien aux programmes si ce n'est la bibliothèque utilisée.

Nous avons pu entre autre faire des tests de commandes des équipements (par exemple fixer une valeur de tension de sortie aux alimentations, de lire les valeurs des sorties), pour les équipements reliés en LAN, tester la connexion éthernet en directe, puis via l'utilisation d'un switch lan D-LINK DES1005D.

Nous avons décidé avec Aurélia Secroun d'utiliser le maximum d'instruments possible en communication LAN, car plus fiable que de la communication Série GPIB ou USB, et permettant de plus grandes distances de câbles (au LAM, la manipulation sera à 7 mètres du serveur, à cette distance les câbles USB ne sont plus fiables), néanmoins quelques instruments anciens ne possède que des ports GPIB (cf. illustration ci-contre). Néanmoins nous avons pu récupérer à l'APC un convertisseur GPIB-LAN qu'ils ne réussissaient pas à faire fonctionner (à cause de problèmes de drivers) et grâce à l'aide de Julien Zoubian, à 3 nous avons pu régler ce problème et utiliser ce convertisseur, sur lequel les instruments en GPIB seront reliés, pour ensuite être connectés en Ethernet à notre serveur.

Le bilan que nous avons pu faire de ces 2 journées à l'APC a été très positif, tous les drivers sont testés, documentés et fonctionnels comme le montre le tableau récapitulatif ci-dessous :



Figure 58: Port GPIB

Composants/grandeurs	Appareil	Quantité	Connexion	Driver/librairie	Fréquence de lecture	Résolution	Nombre de connexions	Interaction	État d'avancement
Températures	AGILENT 34980A	1	LAN	Python + VISA	1Hz	6 chiffres %, 22 bits	24x PT10K + 24x NTC30k/50k	Lecture	Driver fonctionnel
Signaux DC (résistances)							40 protocoles = 24		A tester
Chaudronnage	NUMATO 32Ch annelés GPIBModule	1	LAN	Python + Test			12 différentes	ON/OFF	Driver fonctionnel
QFR + SPR							12 photorécepteurs		
Alimentation PB	AGILENT E3649A	1	GPIB	Python + VISA			4	Commande	Driver fonctionnel
Alimentation chaudières	AGILENT E3631A	1	GPIB	Python + VISA			12	Commande	Driver fonctionnel
Moteurs des injecteurs	NEWPORT 8742	2	LAN	Python (socket)			8 moteurs	Commande	Driver fonctionnel
Phase	Phasemeter APC	3	LAN	C (AEI)	400Hz	64 bits	8 voies, 2 phasemeters/phasemètre, 16 signaux trigger	Lecture	Fonctionne en C sous Ubuntu 18.04
Signal AC						54 bits	8phasemètre		A tester
Signal Tente	AERO-PLEX 2023A	2	GPIB	Python + VISA	Ponctuel	5 chiffres	24	Commande	Driver fonctionnel
Signal d'échantillonage							1		Driver fonctionnel
Asservissement en phase	TEKTRONIX AFG3102	1	LAN	Python + VIX11		24 bits	1	Commande	Driver modifié avec VIX11 à tester
Asservissement en puissance	TEKTRONIX AFG3252	1	LAN	Python + VIX11		24 bits	2	Commande	Driver modifié avec VIX11 à tester
Compteur de fréquence	PERIODICOM CNO-3P	1	GPIB	Python + VISA		64 bits	1	Lecture	A tester
	AGILABT 50132R	1	GPIB	Python + VISA		64 bits	1	Lecture	A tester

Figure 59: Tableau bilan

On voit clairement sur ce tableau les points qui restent à corriger, notamment sur le phasemètre et le compteur de fréquence qui lui est associé.

En plus des tests individuels, des test "chainés" ont été effectués par nos soins, c'est à dire que nous avons connecté 2 appareils ensemble, puis 3, 4 etc. jusqu'à ce que le maximum d'appareils soient reliés à notre PC (à nos ordinateur portable à ce moment là, donc des machines nettement moins puissantes que le serveur) et ces tests ce sont révélés concluants, chaque appareil étant reconnu et pouvant via le driver adéquat être lu, et si demandé, contrôlé.

Il manque encore les équipements liés au laser, ce dernier étant encore en cours de conception au SYRTE. Mais des équipements déjà présents à l'APC serviront pour ce dernier.

Voici d'ailleurs une photo du laser et de tout ses équipements :

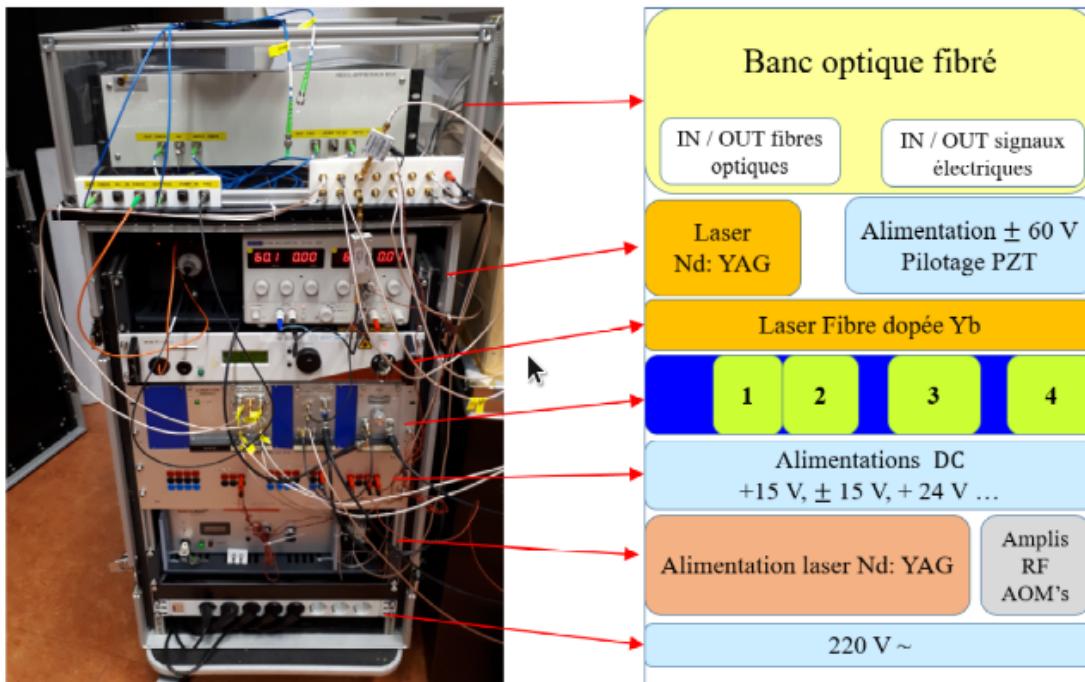


Figure 60: Vue de face du laser V1 du banc

Cette dernière visite à l'APC a donc été l'occasion pour moi de dresser un bilan de ce stage, qui bien qu'il ait été perturbé par tous les éléments que nous connaissons, est arrivé à son terme d'une très bonne manière, l'ensemble de la chaîne de mesure fonctionnant comme nous avons pu le voir via le driver Newport8742, depuis l'instrument physique, jusqu'à l'affichage du monitoring, en passant par la commande par l'IHM. Ensuite les autres drivers, qui sont conçus sur le même modèle, en apportant bien su chacun leurs spécificités (langages de communication différents, fonctions différentes forcément), ne pouvait que fonctionner en suivant la même logique.

En se déplaçant à l'APC nous avons également pu vérifier que nos codes pouvaient fonctionner aussi bien en connexion à distance (VPN) qu'en connexion directe LAN ou série. Pour la suite, il ne restera qu'à implémenter ces codes directement dans le serveur qui gérera la manipulation, et à y adjoindre les différents pipelines de traitement des données (notamment la méthode de Heinzel, implémentée en Python pour le phasemètre) et à tester les connexions VPN et VNC depuis Marseille sur ce serveur.

Quant à l'accès à l'interface de monitoring via le web, il est possible via un accès SSH sécurisé qui a été testé avec le service informatique du CPPM, et qui sera ensuite transmis au service informatique de l'APC.

Ayant participé à la caractérisation du serveur qui servira à la manipulation, j'aimerai beaucoup être capable un jour de le voir en fonctionnement, avec tous les codes implémentés en lui.

Pour information, tous les codes que j'ai développés sont disponibles sur le Gitlab IN2P3, un plateforme de mise en commun des codes, permettant de gérer les versions, de créer des "issues" en cas de problèmes sur un code, c'est un excellent outils pour le développement, que je continuerai d'utiliser à l'avenir. Ayant appris à m'en servir au cours de ce stage.

Le seul point sur lequel je suis un peu déçu de mon stage est donc que je n'ai pas pu aller au bout de la réalisation des codes pour le phasemètre, partie qui était la plus complexe, mais qui représentait le plus d'enjeux pour moi également.

Conclusion

Il est temps pour moi à présent de conclure ce rapport. Au risque de me répéter encore, je dirais, pour commencer, que ce stage aura été sans aucun doute une expérience et ce à bien des points de vue, expérience professionnelle, expérience humaine, expérience tout court... ce qui aura été très important et enrichissant pour moi.

Comme nous avons pu le voir tout au long de ce rapport, j'ai notamment, je pense, très largement amélioré mon niveau en programmation, en découvrant et en perfectionnant ma maîtrise des outils comme GitLab, QtDesigner, Grafana, la base InfluxDB, notamment déjà par l'aspect réellement pratique des codes que j'ai pu réaliser. C'était une véritable fierté personnelle déjà de pouvoir travailler dans un laboratoire tel que le CPPM, qui plus est sur un projet d'une telle envergure. Imaginez ma satisfaction la première fois qu'un de mes programmes est parvenu agir sur le contrôleur des moteurs !

J'ai évidemment ressenti la même satisfaction lorsque les actions réalisées sur un équipement devenaient grâce à mes programmes visibles sur l'interface Grafana.

Grâce à cette nouvelle expérience professionnelle, j'ai conforté mes ambitions futures et suis parvenu à m'orienter dans une direction qui me satisfait; j'ai reçu les bons enseignements, les bons conseils, et les bonnes aides me permettant de rechercher et obtenir une thèse dans un des observatoires les plus réputés de France.

Ces 6 mois ont également été l'occasion pour moi de nombreuses personnes sympathiques, à l'écoute, passionnées par leur métier et surtout très compétentes. Leur amitié, leur soutien et leurs conseils m'ont aidé et m'aideront encore à poursuivre mon chemin. Certes, je les ai déjà remerciées en tout début de rapport, mais il me paraît normal d'avoir encore une pensée pour chacun d'entre eux au moment de le conclure.

J'ai eu l'occasion de découvrir et d'approfondir de nombreuses choses diverses dans les domaines de la programmation, de l'ingénierie, la cosmologie, l'électronique, ou la physique des particules.

Moi qui suis d'un naturel assez réservé ne me suis rarement senti aussi bien et vite intégré et à l'aise dans une équipe car dès le premier jour, on m'a donné mes chances de faire mes preuves, on m'a fait confiance, et j'espère sincèrement aujourd'hui avoir réalisé ce qui était attendu de moi et n'avoir déçu personne.

Difficile bien sûr de ne pas souligner le contexte particulier dans lequel s'est déroulé une partie de ce stage à la saveur assez particulière. La crise sanitaire nous obligeant tous au confinement et en ce qui me concerne au télétravail, il est évident que de nombreuses choses étaient plus compliquées à mettre en place, surtout qu'il n'est pas évident de réaliser certains travaux à distance. Néanmoins, je considère cela également comme une expérience enrichissante, et je pense que c'est également cela m'a amené à faire encore davantage de mon mieux afin que l'ensemble de mes productions aussi bien au niveau des codes que des rapports ou des documentations soit le plus proche de ce qui était attendu d'un ingénieur de recherche.

Il a fallu travailler à distance, en faisant régulièrement des points par visio-conférences, échangeant nos travaux, nos questions, nos points de vue; ce qui a demandé une certaine adaptation non seulement de ma part mais également de tout le CPPM et de ma tutrice. Nous sommes parvenus à avancer et malgré tous ces fâcheux événements le travail a pu continuer dans de bonnes conditions.

Cette période m'a d'ailleurs contraint à mieux documenter tout ce que je réalisais, comme le font tous les ingénieurs. J'ai fait davantage de recherches, pris davantage de conseils, et suis parvenu à réaliser la partie du travail qui m'avait été confiée.

L'accueil que j'ai reçu au CPPM, le cadre de travail dans lequel je me trouvais, les rencontres que j'y ai faites au cours de ces quelques mois et les très nombreuses opportunités que l'on m'a données me permettent aujourd'hui d'appréhender avec confiance ma future vie professionnelle et envisager l'avenir sereinement.

Comme je l'ais déjà souligné, j'ai d'ailleurs d'ores et déjà l'opportunité aujourd'hui de pouvoir continuer à travailler sur le projet LISA, via un autre de ses aspects, en me focalisant sur l'instrument, le MOSA, grâce à une thèse que j'aurais prochainement l'occasion de pouvoir démarrer à l'Observatoire de la Côte d'Azur de Nice aux côtés de monsieur Michel Lintz.

La dernière partie de ce rapport faisait déjà plus ou moins office de conclusion technique, les codes jusqu'ici réalisés fonctionnant comme il était attendu. Il reste néanmoins beaucoup de travail avant d'arriver au banc de test final, que j'espère vraiment avoir l'occasion de pouvoir voir en action le moment venu, ce qui représentera une finalité à mon travail personnel sur le projet, bien qu'il n'en représente qu'une partie infime.

En ce qui les éléments perfectibles à la fin de mon stage, je pourrais citer l'optimisation de la base de données, la configuration complète du serveur, l'optimisation de l'interface de commande, avec l'ajout des scripts extrêmement complexe, et surtout, la mise en place de la gestion des différents phasemètres ce qui, comme nous avons pu le voir précédemment, pose le plus de problèmes, avec également la possibilité de réussir à synchroniser l'acquisition des données de chacun de ces instruments.

Je souhaite enfin une dernière fois remercier Aurélia Secroun, et tout le personnel du CPPM et des divers laboratoires avec lesquels j'ai travaillé. J'espère également avoir encore l'occasion de les rencontrer et pourquoi pas retravailler avec eux tout d'abord au travers de ma thèse.

Pour être franc, l'instrumentation était une voie qui ne me parlait pas tant que cela avant ce stage, et aujourd'hui, m'ayant fait une idée plus précise du sujet, je trouve cela extrêmement intéressant et enrichissant, notamment l'instrumentation pour le spatial, qui pour moi, est une traduction des spécialités pour lesquelles j'ai opté à l'IPSA, permettant de mêler à la fois programmation, électronique et astrophysique / et aussi cosmologie dans une certaine mesure.

En conclusion, j'ai effectué un excellent stage, dans une ambiance de travail agréable, au sein d'une équipe chaleureuse, dans un cadre idyllique, et entouré des attentions de chacun de mes collègues. J'ai eu l'honneur de travailler sur un projet qui n'en est bien évidemment encore qu'à ses débuts mais qui offre d'ores et déjà des perspectives d'évolutions immenses, et qui m'ont permis de m'améliorer autant personnellement d'un point de vue humain, qu'au niveau de mon sens de l'organisation de travail, de l'organisation de réunions, de gestion de ses dernières, et comme dit précédemment, sur mes capacités en programmation en Python, C. J'espère avoir l'occasion un jour de retravailler au CPPM. Et serai ravi que d'autres IPSAliens après moi puissent découvrir ce laboratoire, la mission LISA et le monitoring.

Liste des figures

1	Cristinel DIACONNU, Directeur du CPPM	2
2	Le CPPM, au coeur du parc national des Calanques de Marseille	2
3	Ressources & Chiffres clés issus du rapport d'activités 2019 du CPPM - Auteure : Magali Damoisseaux	3
4	Un des éléments de KM3Net	4
5	Organigramme du CPPM - Source : Site web du CPPM	4
6	Photographie du groupe Renoir	6
7	Vue d'artiste d'Euclid	7
8	Illustration du phénomène de lentillage gravitationnel	7
9	Vue d'artiste du LSST	7
10	Illustration de la déformation de l'espace-temps à proximité d'un binaire de trous-noir	9
11	Les interféromètres au sol, LIGO à gauche et Virgo à droite	10
12	Principe de fonctionnement d'un interféromètre de Michelson, comme LIGO/Virgo	10
13	La détection d'une onde gravitationnelle, à gauche un modèle prévisionnel et à droite la détection effective d'une onde par LIGO en 2015	11
14	Représentation de la constellation LISA	11
15	Sensibilité de LISA et des interféromètres au sol en fonction de leur bande de fréquence	12
16	Organigramme représentant l'implication de chaque pays membre de l'ESA au cœur de la mission LISA	13
17	Représentation du banc de test interférométrique de LISA	14
18	La cuve ERIOS du LAM - Site du LAM	15
19	Les normes ISO pour les salles blanches	15
20	Représentation de l'ensemble des signaux à monitorer - Aurélia Secroun & Alexandre CONDETTE	16
21	Schéma fonctionnel du Monitoring dans son ensemble	18
22	Logo de Python	19
23	Fichier de configuration en JSON - init.json du contrôleur des moteurs	20
24	Extrait de code : Commande SCPI, Logfile et Ecriture BDD	21
25	Message de test Doctest réussi	22
26	Message de test Doctest ayant échoué	23
27	Exemple de documentation auto-générée avec Sphinx	24
28	Arborescence des fichiers du projet de monitoring	24
29	Fonctionnement global des drivers	25
30	Logo d'InfluxDB	26
31	Logo de Grafana	27
32	Extraits de la page Grafana de Monitoring du banc LISA	28
33	Page d'accueil de Grafana - LISA	29
34	Page d'accueil de QtDesigner	30
35	QPushButton dans Qt	31
36	QPushButton dans Qt	31
37	Code permettant de lien l'application à la fonction	32
38	Résultat	33
39	Interface de commande des moteurs - Alexandre CONDETTE	34
40	La liaison IHM - Grafana	35
41	Page de commande des connexions	35
42	Illustration du PWM	36
43	Page de réglage des résistances chauffantes	37
44	Phasemetre APC-LISA	38
45	Phasemetre APC-LISA - Schéma bloc	39
46	Données binaires du phasemètre lues en fichier texte	39
47	Bruit relatif du phasemètre - Méthode Usuelle	40
48	Méthode de Welch - Segmentation	40
49	Méthode de Welch - Fenêtrage	41
50	Bruit relatif du phasemètre - Méthode Heinzel Matlab	41
51	Fichiers du code C	42
52	Un oscilloscope, un générateur de fréquence et un générateur de tension	43
53	Un Générateur de Tension 2 sorties Agilent	44

54	L'installation du phasemètre	44
55	Un générateur textronix	45
56	Une carte Numato 32 Sorties GPIO	45
57	Les contrôleurs des moteurs Newport	46
58	Port GPIB	46
59	Tableau bilan	46
60	Vue de face du laser V1 du banc	47

Table des Annexes

- i. Bibliographie
- ii. Webographie
- iii. Liste des sigles et abréviations
- iv. Glossaire
- v. Evaluation de Stage
- vi. Documentation SCPI
- vii. Exemple de requirements.txt
- viii. Extrait du code driver Newport8742
- ix. Documentation HTML du Newport8742
- ix. Code de la Base de données

Bibliographie

GUDRUN Wanner & all : *Methods for simulating the readout of lengths and angles in laser interferometers with Gaussian beams* , Optics Communication 11/07/2011

SECROUN Aurelia & KAJFASZ Eric : *LISA, Dernières nouvelles et coup d'oeil vers l'avenir* , Publication interne, 24/09/2019

HALLOIN Hubert : *LISA : principe de mesure et activités AIVT* , Publication interne, 18 mai 2017

HALLOIN Hubert : *L'Observatoire LISA* , Publication interne, 22 Septembre 2017

HALLOIN Hubert : *Proposal for a LISA Payload AIV Plan* Publiation interne LISA France, 09 Mai 2016

BARRANCO German Fernandez : *Photodetection in satellite laser interferometers* , Thèse de la Fakultat fur Elektrotechnick und Informatik, Hannover, 2017

BARRANCO German Fernandez & all : *Phase stability of photoreceivers in intersatellite laser interferometers* , Optic Express, Avril 2017

BARRANCO German Fernandez & all : *Spatially resolved photodiode response for simulating precise interferometers* , Applied Optics, Aout 2016

CERVANTES Felipe Guzman & all : *Characterization of photoreceivers for LISA*, Arxiv, 20 Mars 2012

MARTINO Joseph : *Budget de performance LISA et LISA Pathfinder*, LISA France, Juillet 2018

HALLOIN Hubert : *Cours LISA, bases d'interférométrie* , Publication interne, Juillet 2018

DIEKMANN Christian : *Developmnet of core elements for the LISA optical bench - Electro-Optical measurement systems and test devices* , Thèse de la Fakultat fur Mathematik und Physik Hannover, 2013

BUY Christelle : *Heterodyne interfeometry optical bench ZIFO test request* , LISA France, 26 Septembre 2019

ZERGUERRAS Thomas : *Technology items for the heterodyne interferometry optical bench for LISA MOSA AIVT activities* , LISA France, 14 Janvier 2020

Publication ESA : *LISA Metrology System, Final Report* , 2014

DANZMANN Karsten : *LISA, A proposal in response to ESA call for L3 mission concepts* , Publication AEI, 20 Janvier 2017

LISA Instrument Group : *LISA Payload Definition Document* , Publication ESA, 1 Mai 2018

GERBERDING Oliver : *Phase Readout for satellite interferometry*, Thèse de la Fakultat fur Mathematik und Physik Hannover, 2014

SCHWARZE Thomas : *Phase extraction for laser interferometry in space: phase readout schemes and optical testing*, Thèse de la QUEST-Leibniz-Forschungsschule Hannover, 2018

HALLOIN Hubert & all : *LISA : Hearing the heavens* , Séminaire au CPPM, 27 Mai 2019

DANZMANN Karsten & all : *LISA, Unveiling a hidden Universe* Publication ESA, Fevrier 2011

Trobs Michale & HEINZEL Gerhard : *Improved spectrum estimation from digitized time series on a logarithmic frequency axis* , Science Direct, 5 Decembre 2005

JIMENEZ Pablo Marin : *Development of a wide-bandwidth laser frequency noise measurement system*, Rapport de fin d'études de l'Universidad de Granada, Fevrier 2013

JIMENEZ Pablo Marin, SANTOS Cayetano & PRAT Pierre : *Phasemeter LISA Documentation*Publication interne, 08 Septembre 2015

PRAT Pierre & SUBE Samuel : *Mesures de bruit de phase, Phasemètre prototype LISA* , Publication interne 26 Septembre 2019

ASEF Ouali : *Manuel d'utilisation du banc Laser V1* , LISA France, 31 Janvier 2020

Webographie

Site du Ministère de l'Enseignement Supérieur, de la Recherche et de l'Innovation

Chiffres clés de la recherche en France : <https://www.enseignementsup-recherche.gouv.fr/pid25351/chiffres-cles-de-la-recherche.html>

Site de Campus France

*Chiffres-clés : L'importance de la recherche en France : <https://www.campusfrance.org/fr/importance-recherche-France>
Des Chercheurs Français exceptionnels :<https://www.campusfrance.org/fr/chercheurs-francais-exceptionnels>*

Site de LISA :

<https://www.elisascience.org>

Site du CPPM :

<https://www.cppm.in2p3.fr>

Site du groupe RENOIR :

<https://www.cppm.in2p3.fr/renoir/>

Site de l'APC:

http://www.apc.univ-paris7.fr/APC_CS/

Validation expérimentale des performances interférométrique de LISA : https://www.apc.univ-paris7.fr/APC_CS/fr/validation-experimentale-des-performances-interferometriques-de-lisa

Site du LAM

<https://www.lam.fr/>

LISA Nasa website

<https://lisa.nasa.gov/>

LISA - ESA

https://www.esa.int/Science_Exploration/Space_Science/LISA

Le Temps

L'interférométrie, l'art de mesurer des distances <https://www.letemps.ch/sciences/linterferometre-lart-mesurer-di>

Optique Ingénieur

Interférométrie hétérodyne http://www.optique-ingenieur.org/fr/cours/OPI_fr_M02_C08/co/Contenu_03.html

Interférométrie homodyne http://www.optique-ingenieur.org/fr/cours/OPI_fr_M02_C07/co/Grain_OPI_fr_M02_C07_2.html

Liste des sigles et abréviations

AIVT : Assemblage, Integration, Validation and Tests
APC : Laboratoire AstroParticules et Cosmologie (Paris)
BAO : Baryonic Acoustic Oscillations
CAN : Convertisseur Analogique-Numérique
C&C : Contrôle-Commande
CERN : Conseil Européen pour la Recherche Nucléaire
CMB : Cosmic Microwave Background
CNRS : Centre National de la Recherche Scientifique
CPPM : Centre de Physique des Particules de Marseille
DE : Dark Energy
DESI : Dark Energy Spectroscopic Instrument
DM : Dark Matter
eBOSS : extended Baryon Oscillation Spectroscopic Survey
ERIOS : Etalonnage Réglage et Intégration pour l'Optique Spatiale
ESA : European Space Agency
FPGA : Field Programmable Gate Array
IHM : Interface Homme Machine
IN2P3 : Institut National de Physique Nucléaire et de Physique des Particules
LHC : Large Hadron Collider
LIGO : Laser Interferometer Gravitational-Wave Observatory
LISA : Laser Interferometer Space Antenna
LSST : Legacy Survey of Space and Time
MOSA : Movable Sub-Optical Assembly
Mpsec : Méga-Parsec
NASA : National Aeronautic and Space Administration
PI : Proportionnel-Integral
PLL : Phase Locked Loop
PWM : Pulse Width Modulation
QPR : Quadrant Photoreceptor
ReNOIR : Recherche de l'énergie noire
RST : ReStructured Text
SCPI : Standard Commands for Programmable Instruments
SPR : Single Photoreceptor
TDI : Time Delay Interferometry
TTL : Transistor-Transistor Logic
VHDL : Very High Speed integrated Circuit Hardware Description Language
VNC : Virtual Network Computing
VPN : Virtual Private Netork

Glossaire

Astroparticules : Domaine de l'astrophysique qui étudie les particules d'origine cosmique présentes dans l'espace.

Boson : Particule Subatomique de spin entier (exemple : photon), ils agissent comme intermédiaires des interactions fondamentales (sauf le boson de Higgs)

Boson de Higgs : Particule élémentaire découverte au sein du LHC en 2012 et qui permet de conférer une masse non nulle aux boson W et Z (médiateurs de l'interaction faible)

Cosmologie : Domaine de l'astrophysique qui étudie l'origine, la nature, la structure et l'évolution de l'Univers

Drivers : Programme informatique destiné à permettre l'intéraction avec un périphérique

Energie Noire : Forme d'énergie hypothétique, invisible qui composerait 70% de la masse de l'Univers et souvent qualifiée de "force gravitationnelle répulsive", elle serait responsable de l'accélération de l'expansion de l'Univers

Grand Collisionneur de Hadrons : Le LHC est le plus puissant accélérateur de particules du monde, des protons sont accélérés en son sein. Le LHC a permis de confirmer l'existence du boson de Higgs

Hadron : Particule composée de particules subatomiques régie par l'intéraction forte (exemple un proton, composé de 2 quarks up et 1 quark down)

Interférométrie : Superposition d'ondes électromagnétiques provoquant un phénomène d'interférences afin d'en extraire des informations

Matière Noire : Matière hypothétique, qui composerait 27 % de notre Univers, et qui permet d'expliquer plusieurs phénomènes, notamment l'estimation de la masse des galaxies, ou la courbe des vitesses de rotation de ces dernières en fonction de la distance au centre galactique (Travaux de Vera Rubin).

Ondes Gravitationnelles : Courbure de l'espace-temps se propageant à la vitesse de la lumière, prédit par la théorie de la relativité générale d'Einstein en 1916, et dont l'existence a été démontrée en 2015.

Oscillations Acoustiques Baryoniques (BAO) : Ondes acoustiques qui se propageaient dans le plasma primordial (avant la recombinaison), ayant laissé des traces dans la structure actuelle de l'Univers.

Spin : Caractéristique quantique représentant le moment cinétique intrinsèque d'une particule en rotation sur elle-même

Supersymétrie : Modèle prédisant que chaque particule du Modèle Standard dispose d'un partenaire "supersymétrique", dont le spin diffère du sien de moitié ; cela permet d'expliquer la masse des particules.

Annexe I : Evaluation de stage

EVALUATION DU STAGE OU DE LA MISSION PAR L'ENTREPRISE TRAINEES PLACEMENT ASSESSMENT FORM									
 Date : 2 septembre 2020		Niveau du stagiaire 2ème cycle <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>4ème année</td> <td><input type="checkbox"/></td> <td>Sème année</td> <td><input checked="" type="checkbox"/></td> </tr> </table>				4ème année	<input type="checkbox"/>	Sème année	<input checked="" type="checkbox"/>
4ème année	<input type="checkbox"/>	Sème année	<input checked="" type="checkbox"/>						
NOM ET PRENOM DE L'ELEVE - STUDENT NAME NOM TUTEUR/REPRÉSENTANT(S) DE L'ENTREPRISE TUTOR OR COMPANY REPRESENTATIVE ENTREPRISE (si stage ou contrat) - COMPANY (if work placement or contract)		Alexandre Condette Aurélie Scroun CNRS/CPME							
Compétences mises en œuvre par le stagiaire <i>Competences put in practice by the trainee</i>		Oui Yes	Non No						
A répond à un besoin opérationnel et/ou fonctionnel par une solution technique <i>Has responded to an operational and/or functional need with a technical solution</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Has conceived or developed a system or equipment		<input checked="" type="checkbox"/>	<input type="checkbox"/>						
A réalisé un équipement ou un système - <i>Has built a system or an equipment</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>						
A conduit ou géré un projet, une étude - <i>Has lead or managed a project or a study</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>						
A eu des responsabilités en management - <i>Has had Management responsibilities</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>						
				note seul pour valider le stage fixée à : 12/20					
				Pensez à indiquer Votre NOTE GLOBALE	20/20				
Merci de cocher la case correspondante									
Comportement en situation de travail - Behaviour in the workplace		Insuffisant Insufficient	Moyen Average	Bien Good	Très bien Very Good				
Ponctualité, Assiduité - <i>Punctuality/ Attendance</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Respect des délais par rapport aux travaux demandés - <i>Ability to complete assignments to a deadline</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Curiosité intellectuelle - <i>Intellectual Curiosity</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Rapide, Efficacité - <i>Speed and efficiency</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Organisation, Méthode, Rigueur - <i>Organisation, Method, Rigour</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Motivation et investissement - <i>Motivation</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Aptitude à communiquer, sens du contact - <i>Interpersonal Relationships & Communication</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Esprit d'initiative - <i>Sense of initiative</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Respect des règles de sécurité et du règlement intérieur (si applicable) <i>Respect of Safety Rules (if applicable)</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Présentation, aisance - <i>Self Presentation, Confidence</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Niveau d'anglais - <i>Level of English</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Force de conviction - <i>Persuasiveness</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Dynamisme - <i>Dynamism</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Sens des responsabilités - <i>Sense of Responsibility</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Autonomie - <i>Autonomy</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Réalisation des reporting à sa hiérarchie - <i>Achievement of reporting to his superiors</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Participation et intégration dans l'équipe ou le service - <i>Team working ability</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Compétences Métiers / Vocational Competencies		Insuffisant Insufficient	Moyen Average	Bien Good	Très bien Very Good				
Niveau technique - <i>Technical level</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Propositions d'améliorations techniques - <i>Technical initiative and innovation</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Innovation et Créativité - <i>Innovation & Creativity</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Utilisation du outil informatique (Bureautique et CAD) - <i>Use of IT systems (Office and CAD)</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Détection des risques - <i>Risk Detection</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Esprit d'analyse - <i>Analysis capability</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Esprit de synthèse - <i>Ability to synthesize</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Connaissances théoriques - <i>Theoretical knowledge</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Réussite par rapport aux objectifs - <i>Success in achieving objectives set out</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
COMMENTAIRES (obligatoire) - Commentaries (compulsory)									
<p>Alexandre est un excellent étudiant qui a rapidement intégré son poste de stagiaire au sein du projet et aux difficultés rencontrées. Précisément, il s'est très vite intégré dans l'équipe "LISA" du CPME tout en assumant son rôle à l'interface des 5 laboratoires directement impliqués dans son travail. Il a vite compris les enjeux et pris en main son projet en proposant des solutions concrètes qui répondent bien au cahier des charges qu'il a contribué à construire par des discussions avec ses interlocuteurs. Il a fait preuve d'une très grande autonomie, qualité particulièrement appréciable pendant la période de confinement où les seuls contacts que nous avions étaient par visioconférence. Alexandre est à la fois travailleur, efficace et force de proposition. Il s'intéresse à tout. Son travail a abouti à la réalisation d'une interface homme-machine visant à piloter un banc de mesure interfémétrique très bas bruit, interface qui sera utilisée lors des tests effectifs réalisés par les équipes françaises l'année prochaine. Sa contribution est majeure pour le projet et le résultat clairement celui d'un ingénieur de recherche. Alexandre a également eu l'opportunité d'encadrer un étudiant de quatrième année de l'Estaca, qu'il a pris sous son aile, formé et à qui il a passé le relai pour la suite du projet.</p>									
Le sujet de stage était-il lié à la recherche ? <i>The internship's subject was it related to research?</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Si oui, le sujet du stage était-il en lien avec l'industrie ? <i>If yes, the internship's subject was related to the industry?</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Le mémoire de stage est-il confidentiel ? <i>The report is confidential?</i>		<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Suivi post stage (AERO 5 uniquement) - Work placement follow up (only ING3)									
Aptitude à occuper un poste d'ingénieur - <i>Out/Non?</i> oui, sans problème <i>Is the trainee able to occupy an engineering position (yes/no)?</i>									
De quelle nature (ex. ingénieur calcul, ingénieur d'affaires ...) ? <i>What type of position (eg. CAD engineer, business engineer)?</i> ingénieur de recherche <i>What type of position (eg. CAD engineer, business engineer)?</i>									
Si non, expliquer les raisons / <i>If not, explain the reasons:</i>									
Merci de renvoyer ce document dès que possible / Please return the completed document asap mail : neila.souna@ipsa.fr Adresse / Address : IPSA - 63 Boulevard de Brandebourg 94200 IVRY SUR SEINE									

Annexe II : Documentation SCPI -

Newfocus8742

6.2 Command List

The controller understands many commands. The following table lists all of them in alphabetical order. It also shows whether the command can be issued while motion of any Picomotor is in progress or otherwise.

Command	Description	Command executed when motion is in progress
*IDN?	Identification string query	◆
*RCL	Recall parameters	
*RST	Reset instrument	◆
AB	Abort motion	◆
AC	Set acceleration	◆
AC?	Get acceleration	◆
DH	Define home position	
DH?	Get home position	◆
MC	Motor check	
MD?	Get motion done status	◆
MV	Move indefinitely	
MV?	Get motion direction	◆
PA	Move to a target position	
PA?	Get destination position	◆
PR	Move relative	
PR?	Get destination position	◆
QM	Set motor type	◆
QM?	Get motor type	◆
RS	Reset the controller	◆
SA	Set controller address	◆
SA?	Get controller address	◆
SC	Scan RS-485 network	◆
SC?	Get RS-485 network controller addresses	◆
SD?	Get scan status	◆
SM	Save to non-volatile memory	◆
ST	Stop motion	◆
TB?	Get error message	◆

Annexe III : Exemple de requirements.txt

```
alabaster==0.7.12
anaconda-client==1.7.2
anaconda-navigator==1.9.12
anaconda-project==0.8.3
argh==0.26.2
asn1crypto==1.3.0
astroid @ file:///tmp/build/80754af9/astroid_1592495912941/work
astropy==4.0.1.post1
atomicwrites==1.4.0
attrrs==19.3.0
autopep8 @ file:///tmp/build/80754af9/autopep8_1592412889138/work
Babel==2.8.0
backcall==0.2.0
backports.functools-lru-cache==1.6.1
backports.shutil-get-terminal-size==1.0.0
backports.tempfile==1.0
backports.weakref==1.0.post1
beautifulsoup4==4.9.1
bitarray @ file:///tmp/build/80754af9/bitarray_1594753876414/work
bkcharts==0.2
bleach==3.1.5
bokeh @ file:///tmp/build/80754af9/bokeh_1593187602660/work
boto==2.49.0
Bottleneck==1.3.2
brotlipy==0.7.0
certifi==2020.6.20
cffi==1.14.0
chardet==3.0.4
click==7.1.2
cloudpickle @ file:///tmp/build/80754af9/cloudpickle_1594141588948/work
clyent==1.2.2
colorama==0.4.3
commonmark==0.9.1
conda==4.8.4
conda-build==3.20.1
conda-package-handling==1.7.0
conda-verify==3.4.2
contextlib2==0.6.0.post1
cryptography==2.9.2
cycler==0.10.0
Cython @ file:///tmp/build/80754af9/cython_1594831566883/work
cytoolz==0.10.1
dask @ file:///tmp/build/80754af9/dask-core_1594156306305/work
decorator==4.4.2
defusedxml==0.6.8
diff-match-patch @ file:///tmp/build/80754af9/diff-match-patch_1594828741838/work
distributed @ file:///tmp/build/80754af9/distributed_1594750293818/work
docutils==0.16
entrypoints==0.3
et-xmlfile==1.0.1
fastcache==1.1.0
filelock==3.0.12
flake8==3.8.3
Flask==1.1.2
fsspec==0.7.4
future==0.18.2
gevent @ file:///tmp/build/80754af9/gevent_1593009537125/work
glob2==0.7
gmpy2==2.0.8
greenlet==0.4.16
h5py==2.10.0
Heapdict==1.0.1
html5lib @ file:///tmp/build/80754af9/html5lib_1593446221756/work
idna @ file:///tmp/build/80754af9/idna_1593446292537/work
imageio @ file:///tmp/build/80754af9/imageio_1594161405741/work
imagesize==1.2.0
importlib-metadata @ file:///tmp/build/80754af9/importlib-metadata_1593446406207/work
influxdb==5.3.0
influxdb-client==1.10.0
intervaltree @ file:///tmp/build/80754af9/intervaltree_1594361675072/work
ipykernel @ file:///tmp/build/80754af9/ipykernel_1594753368510/work/dist/ipykernel-5.3.2-py3-none-any.whl
ipython @ file:///tmp/build/80754af9/ipython_1593447377063/work
ipython-genutils==0.2.0
```

Annexe IV : Extrait du code driver Newport8742

L'intégralité du code fait environ 32 pages, il est donc impensable de le recopier ici, néanmoins, voici la fonction d'initialisation :

```
# %% Initialisation Method
def __init__(self, initialisation=True):
    """
    All the parameters are initiated via a configuration file
    *config/init.json*
    This method will set all the parameters of the class Newport8742
    in accordance with this file
    Also, the accelerations and velocities values of all the picomotors
    will be set (after verification that the given values are coherent
    with the specifications of the device)
    """

    # MySQL DB initialisation
    self.DataBase = DB.MySQL_DB()
    # Create dataframe to store values that will be written to the database
    self.Injectors_df = pandas.DataFrame()

    # Parameters definition
    self.IP = init_param['IP']['Value'] # IP Address of the controller
    self.PORT = init_param['PORT']['Value'] # Port Value
    #                               23 here
    #                               don't change this value !

    self.BUFFER_SIZE = init_param['BUFFER_SIZE']['Value'] # Min Size of
    # Buffer : 2048

    self.SOCKET = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    # Communication link

    self.STOP_STATE = False

    # Initialise connexion
    try:
        # Ask a connection to the controller via TCP/IP
        self.SOCKET.connect((self.IP, self.PORT))

        # Confirmation of connection by the device
        self.SOCKET.recv(self.BUFFER_SIZE)
        self.CONNECTION = True

        # Query the ID of the used controller
        self.IDN = self.Identification()

        # Verify if connected to right device
        self.VerifyDevice()

        # Console print for dev only,
        # This line will be deleted when a (n+1) function will be developed
        print("CONNECTED TO {ID}".format(ID=self.IDN))

        logger.info("***** CONNECTION SUCCESSFUL"
                   " TO {ID}".format(ID=self.IDN))
        # Record of the connexion in the log file

        # Append data to the dataframe
        self.Injectors_df = self.Injectors_df.append({'Time': self.GET_TIME(), 'ConnexionStatus': 1},
                                                     ignore_index=True)
        # Write data to the database
        self.To_DB()

    except OSError:
        self.CONNECTION = False

        # Possible problems : - Device not connected
        #                      - Device Off
        logger.error('!!!! CONNECTION FAILED TO NEWPORT 8742 !!!!!')
        # Record of the error in the logfile
```

```

print("Controller not connected : CONNEXION FAILED TO NEWPORT 8742")

# Append data to the dataframe
self.injectors_df = pandas.DataFrame()
self.injectors_df = self.injectors_df.append({'Time': self.GET_TIME(), 'ConnexionStatus': 0},
ignore_index=True)
# Write data to the database
self.To_DB()

# =====#
# Initialisation of the other parameters :

if self.CONNECTION is True and initialisation is True:
    logger.info("\n***** NEWPORT 8742 : INITIALISATION PHASE *****\n")
    # For easiest read of the logfile

    for InjectorID in range(1, 3):
        for Motor_ID in range(1, 5):
            CompleteMotor_ID = str(InjectorID) + '>' + str(Motor_ID)
            # Check if the motors are connected
            self.SOCKET.send('{xx}QM?\r\n'.format(xx=CompleteMotor_ID).encode())
            time.sleep(0.2)
            Type = self.SOCKET.recv(self.BUFFER_SIZE).decode()

            if Type == '@\r\n':
                # If the motor is not connected -->
                # Display on Monitoring
                print('NEWPORT 8742 : ERROR : MOTOR {ID} NOT CONNECTED'.format(ID=CompleteMotor_ID))
                logger.error('NEWPORT 8742 : ERROR : MOTOR {ID} NOT CONNECTED'.format(ID=CompleteMotor_ID))

            else:
                # Motor Connected --> Check the current position
                self.Get_Position(CompleteMotor_ID, Display=False)

    # === Configuration of Acceleration Values ===

    Acceleration = init_param['MOTOR_{ID}_A'.format(ID=CompleteMotor_ID)]['Value']
    # For each motor, find the corresponding acceleration parameter
    # in the configuration file

    # Check if the Acceleration value is in the specifications
    # of the device

    if Acceleration > Acceleration_Max or Acceleration < Acceleration_Min:
        Acceleration = Acceleration_Default
        # Acceleration doesn't fit the spec -->
        # Set to a default value
        logger.warning("{ID} : "
                      "ACCELERATION SETTING OUT OF RANGE"
                      " - SET TO DEFAULT VALUE"
                      .format(ID=Motors_Correspondances[CompleteMotor_ID]))

    print("NEWPORT 8742 : WARNING : "
          "{ID} : "
          "ACCELERATION SETTING OUT OF RANGE - "
          "SET TO DEFAULT VALUE"
          .format(ID=Motors_Correspondances[CompleteMotor_ID]))

    self.Set_Acceleration(CompleteMotor_ID, Acceleration)
    # Use Method to set acceleration for each motor
    # and check if there are no errors

    # === Configuration of Velocity Values ===

    Velocity = init_param['MOTOR_{ID}_V'.format(ID=CompleteMotor_ID)]['Value']
    # For each motor, find the corresponding velocity parameter
    # in the configuration file

```

```

# Check if the velocity value is in the specifications
# of the device

if Velocity > Velocity_Max or Velocity < Velocity_Min:
    Velocity = str(Velocity_Default)
    # Velocity doesn't fit the spec -->
    # set to a default value

    logger.warning("{ID} : "
                  "VELOCITY SETTING OUT OF RANGE - "
                  "SET TO DEFAULT VALUE"
                  .format(ID=Motors_Correspondances[CompleteMotor_ID]))

    print("NEWPORT 8742 : WARNING : "
          "{ID} : "
          "VELOCITY SETTING OUT OF RANGE - "
          "SET TO DEFAULT VALUE"
          .format(ID=Motors_Correspondances[CompleteMotor_ID]))

    self.Set_Velocity(CompleteMotor_ID, Velocity)
    # Use Method to set velocity for each motor
    # and check if there are no errors

    # Verification that the initialisation is coherent

    # Check @1 - Buffer Size
    if self.BUFFER_SIZE < 2048:      # Buffer Size no sufficient
        self.BUFFER_SIZE = 2048

logger.info("\n===== NEWPORT 8742 : END OF INITIALISATION PHASE =====\n")
# print("\n===== NEWPORT 8742 : END OF INITIALISATION PHASE =====\n")
# For easiest read of the logfile

```


Annexe V : Documentation HTML du Newport8742

LISA Monitoring

Navigation

Contents:

I. Prerequisites
II. Newport 8742 - Driver
Appendices

Quick search

Go

Newport8742 module

`class Newport8742.Newport8742(initialisation=True)`

[source]

Bases: `object`

This class is a driver for the following device : Newport 8742 piezoelectrical controller

Developed for the LISA MIFO/ZIFO Monitoring By the Particle Physics Center of Marseille (CPPM)

Warning:

- This driver works only for 2 chained controllers Newport8742
- The order in which the controllers are connected is important, make sure to connect the controller whose IP is entered in the config file as 'master' (the only one connected in LAN)

Parameters:

- `IP (str)` – The IP Address of the controller, Default value : 169.254.98.36
- `PORT (int)` – The used port for communication, Default value : 23
- `BUFFER_SIZE (int)` – Size of the buffer for incoming message from the controller : Default value : 2048
- `SOCKET (socket.socket)` – A communication link (bidirectionnal [i.e read & write]) between the computer and the device - The communication is binary encoded
- `CONNECTION (bool)` – A boolean variable to know if the controller is connected or not (`True` = ** Connected**)

Example :

```
>>> MyController = Newport8742()
CONNECTED To New Focus 8742 v2.2 08/01/13 15217
NEWPORT 8742 : INFO : OBI-1 : Ry : ACCELERATION SET at 5000 steps
NEWPORT 8742 : INFO : OBI-1 : Ry : VELOCITY SET at 1000 steps/s
NEWPORT 8742 : WARNING : OBI-1 : Rz : ACCELERATION SETTING OUT OF
NEWPORT 8742 : INFO : OBI-1 : Rz : ACCELERATION SET at 10000 step
NEWPORT 8742 : INFO : OBI-1 : Rz : VELOCITY SET at 850 steps/s
NEWPORT 8742 : INFO : OBI-1 : Ty : ACCELERATION SET at 10000 step
NEWPORT 8742 : INFO : OBI-1 : Ty : VELOCITY SET at 600 steps/s
NEWPORT 8742 : INFO : OBI-1 : Tz : ACCELERATION SET at 100 steps/
NEWPORT 8742 : INFO : OBI-1 : Tz : VELOCITY SET at 1000 steps/s
NEWPORT 8742 : INFO : R2-D2 : Ry : ACCELERATION SET at 5000 steps
NEWPORT 8742 : INFO : R2-D2 : Ry : VELOCITY SET at 1000 steps/s
NEWPORT 8742 : WARNING : R2-D2 : Rz : ACCELERATION SETTING OUT OF
NEWPORT 8742 : INFO : R2-D2 : Rz : ACCELERATION SET at 10000 step
NEWPORT 8742 : INFO : R2-D2 : Rz : VELOCITY SET at 850 steps/s
NEWPORT 8742 : INFO : R2-D2 : Ty : ACCELERATION SET at 10000 step
NEWPORT 8742 : INFO : R2-D2 : Ty : VELOCITY SET at 600 steps/s
NEWPORT 8742 : INFO : R2-D2 : Tz : ACCELERATION SET at 100 steps/
NEWPORT 8742 : INFO : R2-D2 : Tz : VELOCITY SET at 1000 steps/s
>>> print(MyController.IP)
169.254.98.36
>>> print(MyController.PORT)
23
>>> MyController.BUFFER_SIZE
2048
>>> type(MyController.SOCKET)
<class 'socket.socket'>
>>> MyController.CONNECTION
True
```

Note:

All the parameters have to be given to the program via a configuration file : `config/init.json`

`Check_Error(Function, Display=True)`

[source]

This command is used to read the error code.

The error code is one numerical value up to three(3) digits long. (See Appendix for complete listing)

In general, non-axis specific errors numbers range from 0-99.

Axis-1 specific errors range from 100-199, Axis-2 errors range from 200-299 and so on

Parameters:

- `Function (str)` – The name of the function from which the check errors is launched, in order to know in the logfile from where came an error
- `Display (bool)` – If we want to display the error message, by default this variable is True but sometimes, we don't want to display the message (for example in case of double use of the method)

Returns: `self.ERROR_STATE` : A boolean variable in order to know if an error was detected (`True` = Error Detected)

Annexe V : Code de la base de données

```
# -*- coding: utf-8 -*-

"""
This module can be used to create and write in a InfluxDB Database
Version : 2.0.0

CPPM and LISA France property

Alexandre CONDETTE
condette@cppm.in2p3.fr

"""

# %% Modules Import

from influxdb import InfluxDBClient
import doctest

# %% Parameters and Variables

host = 'localhost'      # Host IP Adress for the DB
port = 8086               # Port used by the DB

user = 'LISA_Monitoring'
password = '8w8Cyd8X'

# %% Class Database

class LISA_DB(object):
    """
    A class to communicate with an InfluxDB Database

    For now the connexion is only on localhost:8086, but the adress must change in future versions
    """
    # % INIT
    def __init__(self):
        """
        Initiate the DataBase on a given host adress and port
        This function also create a defined admin user for the database
        """
        self.DB_State = False

        # Create the communication client with the Database
        try:
            self.client = InfluxDBClient(host, port, user, password)

            # Get the List of existing Database
            try:
                self.List_DB = self.client.get_list_database() # We list all the existing Database
                available on the network (at the given ip adress)

                for counter in range(len(self.List_DB)):

                    if self.List_DB[counter]['name'] == 'LISA_DB':
                        self.client.switch_database('LISA_DB') # Switch to our desired database
                        self.client.switch_user(user, password) # Switch to the desired user
                        self.DB_State = True
                        break

                    else:
                        continue

                # If The Database doesn't exist, we create it
                if self.DB_State is False:
                    self.Create_DB('LISA_DB', user, password)
                    self.client.switch_user(user, password)
                    self.DB_State = True

            except ConnectionError:
                print("Please run influxd.exe "
                      "before trying to connect to the database")
            except Exception:
                pass
        except:
            pass
    
```

```

print("Impossible to connect to the DB\r")
print("The database is not connected, "
      "please run influxd.exe before "
      "trying to connect to InfluxDB\n")

# %% Create_DB
def Create_DB(self, DB_name, username, user_password):
    """
    Create an InfluxDb database with a given User & Password

    :param DB_name: The name of the database you want to create
    :type DB_name: str

    :param username: The user ID, here user is 'LISA_MONITORING'
    :type username: str

    :param user_password: The password associated to the given user
    :type user_password: str
    """

    self.client.create_database(DB_name)    # Create the database
    self.client.create_user(username, user_password)  # Add a user in the DB
    self.client.switch_database(DB_name)    # Activate the DB

# %% Write in DB
def Write_in_DB(self, Data, Table):
    """
    Write a package of data into a table of our Database

    :param Data: Our data written in JSON style, 3 fields are mandatory :
    * measurement : The table
    * field : data name
    * tag : Additionnal tags
    :type Data: json

    :param Table: The name of the measurements table
    :type Table: str
    """

    global json_data

    # The datas send to the Database must respect the following syntax :
    json_data = [{ 'measurement': Table,
                  'fields': {},
                  'tags': {}
                }
    ]

    for Key, Value in Data.items():
        if type(Value) == dict: #
            for Key2, Value2 in Value.items():
                json_data[0]['fields'][ '{}'.format(Key, Key2)] = Value2
                json_data[0]['tags'][Key2] = Value2
                json_data[0]['measurement'] = Table

            self.client.write_points(json_data)    # Proper write of Data in the database

        else:
            json_data[0]['fields'][Key] = Value
            self.client.write_points(json_data)

```

Résumé

Ce document retranscrit l'ensemble de mon travail au sein du Centre de Physique des Particules de Marseille(CPPM), stage au cours duquel j'ai eu l'occasion de travailler au sein du collaboration nationale et internationale sur le projet du Laser Interferometer Space Antenna (LISA), une mission de l'ESA et de la NASA. Durant 6 mois j'ai pu programmer à haut niveau en Python pour la réalisation d'interface de contrôle-commande et de monitoring d'un banc de test interférométrique, afin de démontrer la faisabilité du projet, la possibilité d'atteindre la sensibilité requise sur les 3 phasemètres du banc, mais également simuler la détection d'une onde gravitationnelle par le LISA. Ce stage fut une excellente expérience à la fois professionnelle et humaine, qui m'a permis de voir le fonctionnement d'un laboratoire de recherche, mais également de découvrir le monitoring, l'instrumentation. J'ai l'intime conviction que ce type de stage est exactement ce que l'on demande d'un étudiant en 5e année de l'IPSA ayant suivi mes spécialités (Option Système, majeure Espace Lanceurs et Satellites) Ce document présente le CPPM, la mission LISA et explique je l'espère au mieux les travaux que j'ai pu effectuer au sein du groupe RENOIR du CPPM et au sein du consortium LISA France

Abstract

This document transcribes all of my work at the Centre de Physique des Particules de Marseille (CPPM), an internship during which I had the opportunity to work within the national and international collaboration on the Laser Interferometer Space Antenna (LISA) project, a mission of ESA and NASA. During 6 months I was able to program at a high level in Python for the realization of the control and monitoring interface of an interferometric test bench, in order to demonstrate the feasibility of the project, the possibility to reach the required sensitivity on the 3 phasemeters of the bench, but also to simulate the detection of a gravitational wave by the LISA. This internship was an excellent experience, both professional and human, which allowed me to see the functioning of a research laboratory, but also to discover the monitoring, the instrumentation. I am convinced that this type of internship is exactly what one would expect from a 5th year IPSA student who has followed my specialties (System Option, major in Launcher and Satellite Space). This document presents the CPPM, the LISA mission and explains, I hope, the work that I have been able to carry out within the RENOIR group of the CPPM and within the LISA France consortium.

Alexandre CONDETTE - IPSA Aéro 5
alexandre.condette@ipsa.fr – +336 49 76 79 07

