
LAIMI-UQAC

Laboratoire d'automatique et d'interaction

Proposition de projet

Étude de l'inclusion de la détection des activités humaines dans une cellule de travail hybride dédiée à la commande intelligente de robots via ROS en utilisant un réseau de capteurs portables sans fil

Titre	Proposition de projet : Étude de l'inclusion de la détection des activités humaines dans une cellule de travail hybride dédiée à la commande intelligente de robots via ROS en utilisant un réseau de capteurs portables sans fil
Nom de fichier	Proposition de projet.doc
Projet	P2015-02 Étude de l'inclusion de la détection des activités humaines dans une cellule de travail hybride dédiée à la commande intelligente de robots via ROS en utilisant un réseau de capteurs portables sans fil
Format	Word 2000
Date	2015/08/3
Famille	0001 Système qualité
Produit	0003 STB Étude des modèles d'interactions humain-robot
Version	1.0
Révision	.4
Statut	"Release" R

Le présent document est basé sur les normes en vigueur au Laboratoire LAIMI de l'Université du Québec à Chicoutimi.



UQAC

Laboratoire d'automatisme,
d'interaction 3D et multimodale intelligente
Université du Québec à Chicoutimi



Historique des modifications

Vers-Rev	Date	Auteur	Description de la modification



PRÉPARÉ PAR Nom/fonction	APPROUVÉ PAR: Nom/discipline	SCEAU SIGNATURE
Étienne Polesel _____ _____	ÉLECTRIQUE _____ _____	
_____ _____	_____ _____	
APPROUVÉ PAR : Martin Otis _____		
CHARGÉ DE PROJET		

Table des matières

1.0	Présentation du projet	8
1.1	Définition du sujet de projet d'application	8
1.2	Objectifs du projet	9
1.3	Méthodologie.....	10
1.4	Résultats escomptés	14
1.5	Échéancier	15
2.0	Revue de littérature	16
2.1	Les commandes utilisées pour piloter les robots	16
2.1.1	Les commandes visuelles	16
2.1.2	Les commandes vocales.....	16
2.1.3	Les commandes gestuelles	16
2.2	La coopération Humain/Robot dans un même espace de travail.....	17
2.2.1	Cobotique	17
2.2.2	Troisième main	17
2.3	Éviter l'apparition de problèmes de santé	17
2.3.1	Les blessures dues au travail.....	17
2.3.2	Les troubles musculo-squelettiques	18
2.4	Le retour haptique.....	18
3.0	Comparaison de différents microcontrôleurs et modules Wifi puis choix du meilleur élément	18
3.1	Étude des composants.....	19
3.1.1	La solution PIC24 + antenne.....	19
3.1.2	Composants Microcontrôleurs + Antennes	20
3.2	Comparaison.....	23
3.3	Choix effectué	25
3.4	Batterie et recharge sans fil.....	25
3.4.1	Batterie	25
3.4.2	Rechargement sans fil.....	25
4.0	Conception du circuit imprimé	28
4.1	Les composants.....	28
4.2	Les anciens composants.....	28
4.2.2	Les nouveaux composants	31
4.3	Plans schématiques	32

4.4	Le circuit imprimé.....	34
4.4.1	Conception et élaboration	34
4.4.2	Résultats obtenus	34
4.4.3	Directives à respecter	39
4.5	Conclusion.....	39
5.0	Basculement du mode « troisième main » au mode « évitement »	40
5.1	Trois sources de commande.....	40
5.1.1	La tête.....	40
5.1.2	Les mains	41
5.1.3	Les pieds	41
5.2	Les dictionnaires de mouvements détectables et utilisables	41
5.2.1	L'accéléromètre	42
5.2.2	Les capteurs de force	43
5.3	Utilisation des dictionnaires	45
5.4	Les premières mesures.....	46
5.5	La détection du mouvement.....	48
5.5.1	Les filtres	48
5.5.2	Les critères d'évaluations de la reconnaissance	49
5.6	Conclusion.....	52
6.0	Le pilotage du robot en mode « troisième main »	52
6.1	Le pilotage du robot	53
6.1.1	Le type de commande.....	53
6.1.2	Le signal utilisé.....	53
6.1.3	La commande du robot	54
6.2	L'application Android	54
6.3	Les résultats obtenus.....	55
6.4	Recommandations.....	55
6.5	Travail suggéré	55
7.0	Bibliographie	56
8.0	Annexes	57

8.1 Annexe 1 : Programme Matlab développé pour la détection du mouvement d'écrasement de cigarette :

57

8.2	Annexe 2 : Application Android pour la commande des moteurs à courant continu des articulation du robot ERICC	60
8.2.1	L'activité principale.....	60
8.2.2	Classe décrivant le comportement du IOIO	64
8.2.3	Manifeste de l'application	68
8.2.4	Gestion de la fenêtre graphique.....	68
8.3	Annexe 3 : Diagramme de Gantt du projet à l'achèvement du stage	72

Liste des figures

Figure 1	: Illustration du système d'acquisition et de traitement des données actuel	9
Figure 2	: Illustration d'un espace de travail hybride.....	10
Figure 3	: Illustration du système souhaité à la fin du projet	11
Figure 4	: Schéma illustrant l'architecture et le fonctionnement du système à développer.....	13
Figure 5	: Diagramme de Gantt de la réalisation du projet.....	15
Figure 6	: Schéma d'élaboration d'un système de rechargement sans fil pour le circuit imprimé inclus dans la semelle ..	26
Figure 7	: Alimentation réalisée pour l'application.....	27
Figure 8	: Caractéristique du baromètre (tension vs pression)	29
Figure 9	: Caractéristique de l'humidimètre (tension vs humidité relative)	30
Figure 10	: Schéma du routage des connecteurs, des composants d'alimentation et de la centrale inertielle utilisés dans le nouveau circuit imprimé.....	32
Figure 11	: Schéma du routage du microcontrôleur utilisé dans le nouveau circuit imprimé	33
Figure 12	: Couche supérieure	38
Figure 13	: Couche inférieure.....	38
Figure 14	: Position des capteurs de force par rapport au pied	43
Figure 15	: Schéma illustrant le mouvement permettant à l'opérateur de basculer le mode du robot entre "évitement de collision" et "troisième main"	45
Figure 16	: Courbes de relevé des Capteurs de force lors du test du mouvement d'écrasement de cigarette en position assise	46
Figure 17	: Courbes de relevé de l'accéléromètre lors du test du mouvement d'écrasement de cigarette en position assise	46
Figure 18	: Courbes de relevé des capteurs de forces lors du test du mouvement d'écrasement de cigarette en position debout.....	47
Figure 19	: Courbes de relevé de l'accéléromètre lors du test du mouvement d'écrasement de cigarette en position debout.....	47
Figure 20	: Relevés des capteurs de force filtrés.....	49
Figure 21	: Relevés de l'accéléromètre filtrés	49
Figure 22	: Relevés bruts des capteurs de la semelle lors d'une simulation de trois mouvements différents : la marche, l'écrasement avec l'avant du pied et l'écrasement avec l'arrière du pied.....	50
Figure 23	: Norme de l'accélération et capteurs de force après filtrage	51
Figure 24	: Résultats utilisés pour les tests de reconnaissance	51
Figure 25	: Résultat du test de reconnaissance du mouvement d'écrasement de cigarette	52
Figure 26	: Pilotage du bras robotisé ERICC grâce à l'accéléromètre d'un téléphone Android.....	53

Liste des tableaux

Tableau 1	: Comparaison entre les différents composants pouvant être utilisés dans le projet	24
Tableau 2	: Brochage des connecteurs du module WF121 en lien direct avec le circuit imprimé	36

Tableau 3 : Récapitulatif des brochages du circuit imprimé.....37

Tableau 4 : Prix des composants39

Tableau 5 : Mouvements simples du pied détectables par l'accéléromètre.....42

Tableau 6 : Dictionnaire de mouvements de rotation du pied ayant pour centre la première articulation des orteils.....43

Tableau 7 : Dictionnaire des signaux envoyés par les capteurs de force dans la semelle44



1.0 Présentation du projet

1.1 Définition du sujet de projet d'application

Les robots industriels sont majoritairement commandés en boucle fermée sans la perception de leur environnement. C'est pourquoi, afin d'assurer la sécurité du poste de travail, l'accès à l'espace de travail du robot est limité. Cependant, l'industrie automobile a investi des sommes majeures afin d'éliminer ces clôtures de sécurité qui réduisent le potentiel de productivité de ces robots. Dans un contexte sans clôture de sécurité, le robot doit percevoir son environnement et les opérateurs. Une technologie portable sera considérée afin de communiquer au robot la présence humaine sans l'usage de caméra. Cette technologie, nommée *Wireless Body Area Network (WBAN)*, devra être mise en œuvre dans une cellule de travail d'un robot.

Le LAIMI (Laboratoire d'Automatique, d'Interaction 3D et Multimodale Intelligente) de l'UQAC (Université du Québec à Chicoutimi) a créé, conçu et développé une nouvelle génération de vêtements instrumentés et intelligents. L'intégration de vêtements de sécurité intelligents favorisera l'étude et la compréhension du comportement humain de même que l'ergonomie de l'espace de travail afin de réduire le risque d'accidents avec le robot. D'ailleurs, ces accidents sont souvent occasionnés par une perte d'attention du travailleur, une méconnaissance du risque lié à l'utilisation des outils employés dans son milieu de travail ou par une réduction de la méfiance face à la machine due à une habitude de son utilisation quotidienne.

Actuellement, le LAIMI travaille sur différents projets, comptant parmi eux l'élaboration d'une semelle intérieure évaluant la posture de travail ou un casque de sécurité surveillant les activités et l'état de fatigue de l'utilisateur interagissant avec un robot démontré dans (Ping). Toutes ces applications utilisent une carte électronique conçue par le LAIMI comprenant un microcontrôleur PIC24 et un module Bluetooth. De plus, le laboratoire possède un automate APAX d'Adventech, un serveur PowerEdge de DELL, des capteurs Kinect et une antenne Wifi qui ont pour l'instant un potentiel inexploité faute de communication entre ces dispositifs et l'utilisation d'antenne Bluetooth. En effet, comme le montre la Figure 1, les capteurs utilisent une antenne Bluetooth pour communiquer avec une application Android qui traite, avec des algorithmes d'intelligence artificielle, les données reçues des vêtements de sécurité. Ces instruments, de même que les capteurs portables devront permettre la commande intelligente d'un robot industriel.

Dans une volonté de suivre les progrès technologiques, le LAIMI souhaite utiliser l'IoT, soit l'Internet of Things, qui consiste à rendre les objets intelligents dans une certaine mesure mais surtout leur donner la possibilité de se connecter au réseau internet et donc pouvoir être utilisés par n'importe quel dispositif disposant d'une connexion internet, que ce soit un ordinateur, une tablette, un téléphone intelligent ou un automate. L'utilisation de cette technologie permettrait de détecter les activités humaines sur l'électronique portable de manière à les transmettre à un automate qui commande un système de manufacture flexible.

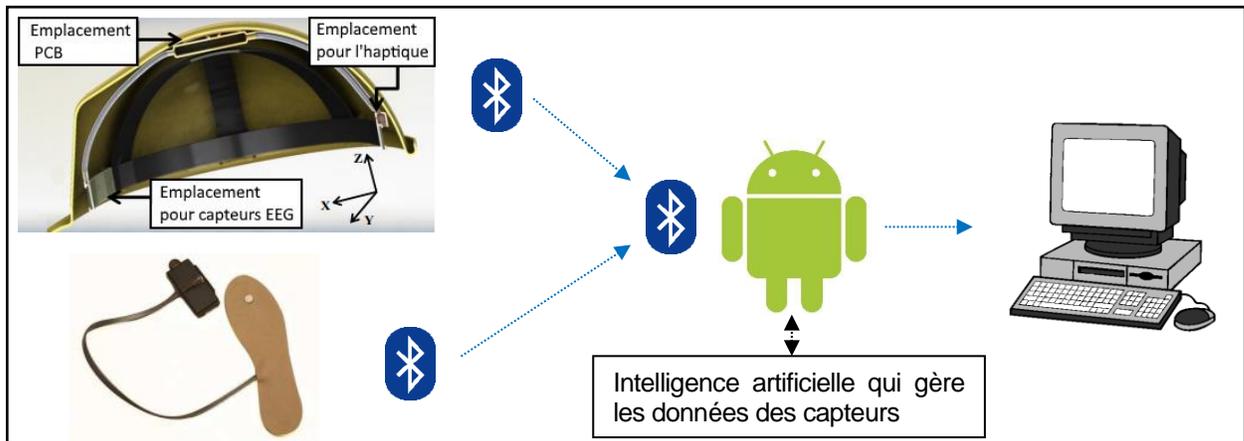


Figure 1 : Illustration du système d'acquisition et de traitement des données actuel

1.2 Objectifs du projet

L'industrie automobile est la plus grande utilisatrice de robots industriels et de machines-outils dans ses installations de fabrication ainsi que chez les fournisseurs d'équipements et d'accessoires. Cependant, ces robots ont leurs limites lorsque la tâche demandée nécessite un degré de perception, d'agilité ou de décision. La communauté scientifique propose des solutions de coopération et d'interaction entre l'humain et le robot afin de réunir les qualités de l'humain (décision, perception, résolution de problème) et du robot (force, endurance et précision) dans un même espace de travail. Néanmoins, bien que la sécurité entourant l'interaction robotique soit largement étudiée, sa mise en œuvre demeure une situation problématique lors de l'utilisation des robots dans le cadre de la santé et la sécurité des travailleurs. L'approche théorique vise donc à étudier l'interactivité humain-robot afin de réduire les symptômes moteurs et favoriser la réintégration des opérateurs ayant subi un accident par l'approche de la boucle perception-décision-action. La problématique est orientée vers l'inclusion de la détection des activités humaines et du positionnement du corps de l'humain dans la commande intelligente d'un robot industriel dans la cellule de travail hybride tel qu'illustré par la Figure 2.

Afin de résoudre cette problématique, l'objectif directeur de ce projet consiste à développer un système de commande qui inclut la plateforme Robot Operating System (Robot Operating System, 2015). Le post-traitement des données brutes (filtrage, segmentation, analyse statistiques) sera réalisé par chaque capteur disposé à l'intérieur des vêtements de protection individuels. Un système de transmission par des algorithmes d'intelligence artificielle est dédié à la détection des activités humaines, à la détection d'erreurs ou de comportements anormaux et à l'étude de la performance des activités ; la Figure 3 en suggère une première configuration. Le projet de recherche propose la conception, la réalisation et la mise en œuvre de la boucle d'interaction perception-décision-action dans une cellule de travail hybride semi-virtualisée afin d'entraîner des opérateurs.

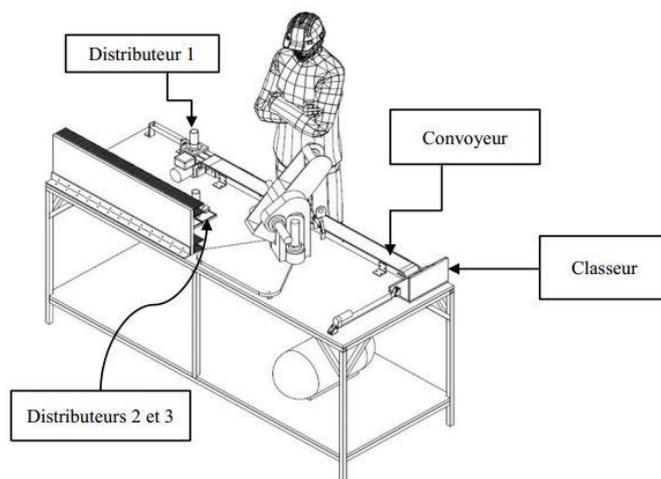


Figure 2 : Illustration d'un espace de travail hybride

1.3 Méthodologie

Habituellement, humains et robots ne sont pas autorisés à travailler ensemble dans un même espace de travail. Lorsqu'un humain est découvert dans l'espace atteignable par le robot, ce dernier est habituellement arrêté, mettant fin à la production. Cependant, à cause de nouveaux besoins, les industriels mènent des recherches pour développer une chaîne de production flexible et réactive afin de permettre une production en petites quantités ou s'adapter rapidement à des modifications des caractéristiques d'un produit (Meziane, 2015). La collaboration entre humain et robot partageant un même espace de travail (habituellement appelé cellule de travail hybride) augmenterait cette flexibilité. Cependant, la sécurité de l'humain dans le cas de la collaboration ou de l'interaction humain-robot est le principal problème.

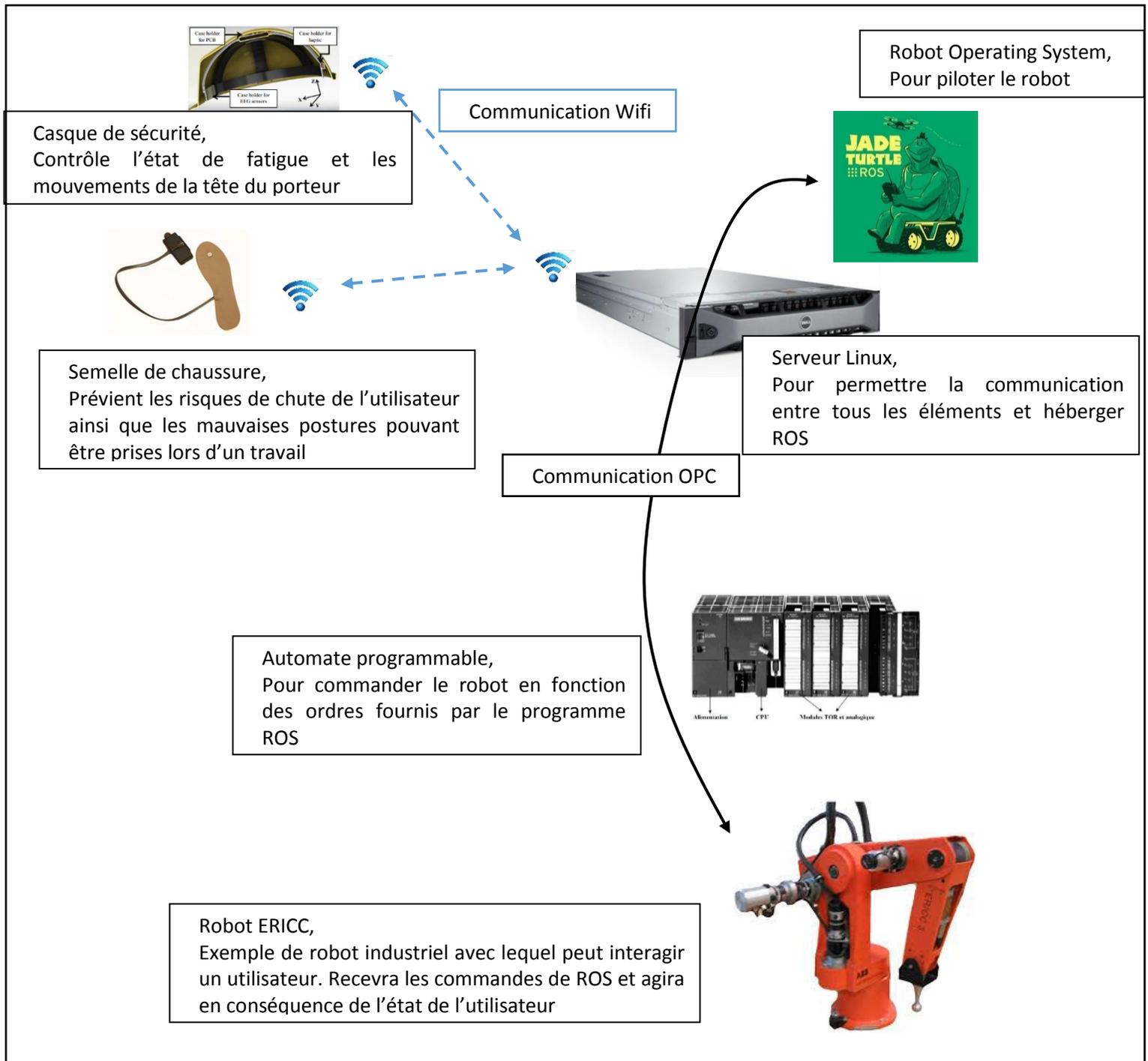


Figure 3 : Illustration du système souhaité à la fin du projet

Ce projet a pour but d'utiliser des capteurs pour inclure les activités humaines à l'intérieur de la boucle de contrôle du mouvement du robot afin d'augmenter la sécurité. Ces capteurs sont incorporés à de nouveaux vêtements intelligents afin d'améliorer la flexibilité dans une cellule de travail hybride. Le robot doit connaître l'action actuelle de l'humain (en utilisant l'intelligence artificielle) afin d'éviter une collision (planification de trajectoire) ou de changer des



paramètres de contrôle (comme l'amortissement) pour des interactions physiques (pHRI). La vision robotique utilisant des caméras ou de balayeurs laser 3D n'est pas utilisée dans ce projet car cette solution ne semble pas appropriée dans un environnement réel.

Dans le cadre de ce projet, on devra développer la fusion des données entre différents capteurs comme des capteurs de proximité ou de force, des accélérations des membres du corps humain via une centrale inertielle (comprenant accéléromètre, gyroscope et magnétomètre) ou encore d'un système de mesure d'un paramètre physiologique lié au stress, la fatigue et la concentration.

Une nouvelle boucle de rétroaction devra être conçue ainsi qu'un module d'intelligence artificielle afin d'ajuster la trajectoire ou les paramètres de contrôle du robot, donnant naissance à une boucle d'interaction. Une des difficultés est de formuler le calcul de la trajectoire en temps réel en utilisant les données du module d'intelligence artificielle (IA). Cette IA devra fournir des informations sur l'activité humaine, les gestes et mouvements pouvant être identifiés, comme des intentions humaines, en utilisant des algorithmes comme les HMM (*Hidden Markov Model*) avec transformation de domaine (dont celles des ondelettes discrètes ou de Haar) ou encore les réseaux de neurones récurrents. Ces intentions devront être considérées dans le calcul de la trajectoire du robot, que l'humain soit en contact physique avec ou pas. Une autre difficulté sera l'élaboration de la transition de contrôle entre les mouvements libres et les mouvements contraints. Tout ce fonctionnement est illustré par la Figure 4.

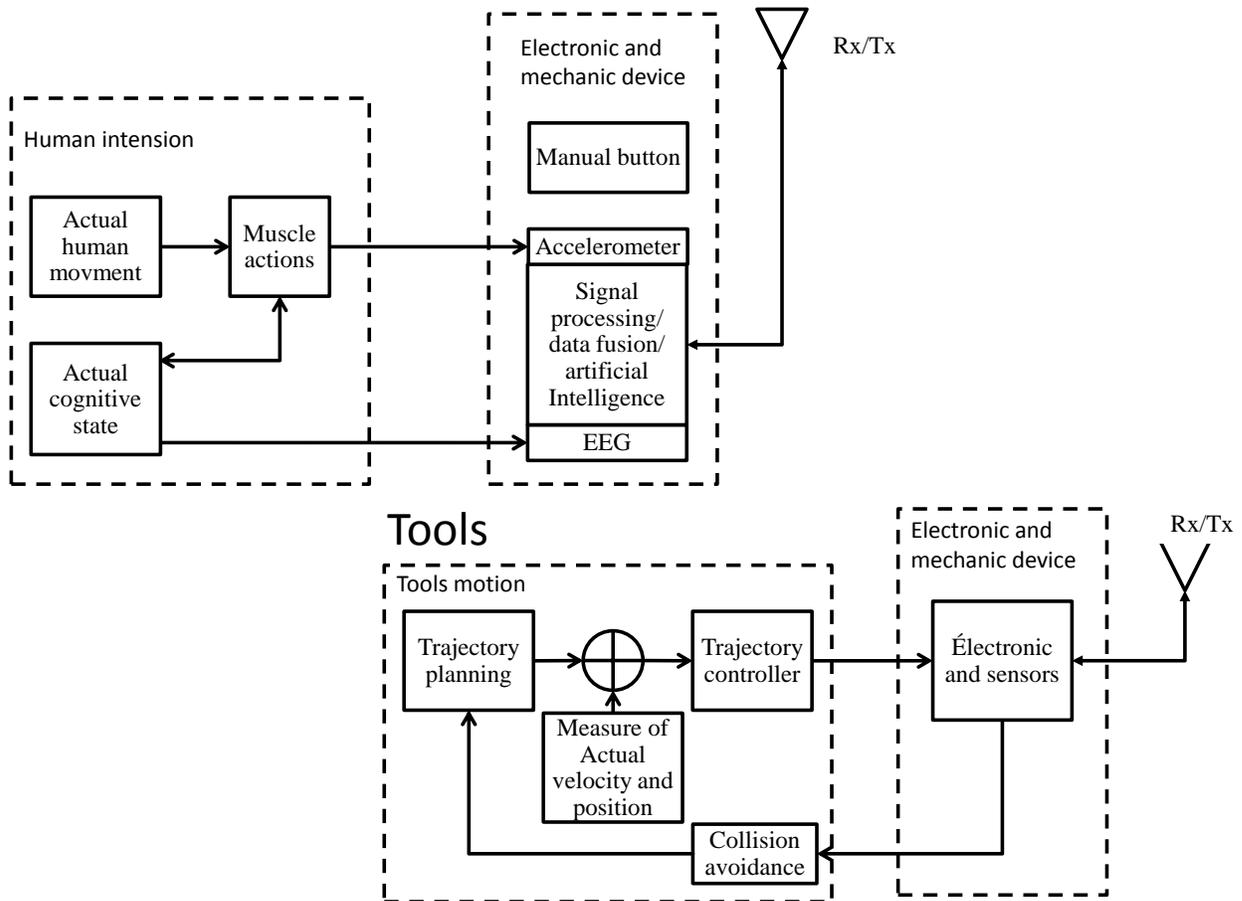


Figure 4 : Schéma illustrant l'architecture et le fonctionnement du système à développer

Un des principaux défis sera de trouver une méthode d'analyse des performances humaines dans des activités de production afin d'en définir ses caractéristiques et son impact. D'ailleurs, des semelles instrumentées ont démontré la capacité de mesurer la posture par une mesure de la distribution des forces au sol après chaque répétition d'une même tâche, comme démontré par Eya Barkallah (Barkallah, Otis, Ngomo, & Heraud, 2015).

En effet, des postures inadéquates pourraient rapidement mener à des troubles musculo-squelettiques. D'autre part, les états mentaux ont un impact important sur la santé et la sécurité (Ping, 2015). A ce sujet, il est connu que l'électroencéphalographie est un système potentiellement intéressant .Il est largement utilisé dans la commande d'application informatique et dans la commande de robots mobiles. Cependant, il a une grande difficulté de calibration et pourrait ne pas fonctionner dans des applications industrielles où les champs électromagnétiques sont présents. Pour cette raison, d'autres systèmes de mesures sont investigués dont la combinaison de la conductance de surface de la peau (réponse électro-dermique) ainsi que sa réponse potentielle. Ces capteurs seront inclus dans un casque de sécurité **Erreur ! Source du renvoi introuvable.** afin de transmettre les informations liées à l'attention ou la concentration de l'opérateur vers le robot. Un niveau de risque sera calculé en temps-réel dans le vêtement de sécurité de manière à transmettre des signaux tactiles d'avertissement. Plus particulièrement, ce thème de recherche vise les cinq étapes suivantes :

	6MIG899 – Projet d'application	Rapport final	Étienne Polesel	1.0.0-R
---	--------------------------------	---------------	-----------------	---------

- Comparer différentes possibilités quant à l'installation d'un module Wifi et de son utilisation. Par exemple, est-il possible de remplacer le microcontrôleur PIC par un module IoT ou alors devons nous garder ce microcontrôleur et ajouter le module Wifi.
- Proposer une communication entre le serveur et l'automate installés au LAIMI, comparer et tester différentes méthodes comme par exemple le protocole OPC (OLE (Object Linking and Embedding) for Process Control) (OpenOPC for Python, 2015) ou CORBA (Common Object Request Broker Architecture (Orfali & Harkey, 1997) pour ne citer que ces deux procédés qui ont l'avantage d'être multiplateforme.
- Réaliser le post-traitement par le module du capteur afin de le rendre intelligent.
- Programmation de ROS en considérant les mesures.
- Afin d'élaborer la technologie, je participerai à l'élaboration d'un scénario d'entraînement préalable pour l'opérateur favorisant une réduction des situations de stress, d'inconfort ou de fatigue.

Afin de pouvoir vérifier et contrôler l'avancée du projet, ces étapes ont été planifiées à l'aide du logiciel MSProject. Le diagramme de Gantt résultant est présenté par la Figure 5.

1.4 Résultats escomptés

Ces travaux de recherche auront une incidence directe sur l'avancement des connaissances dans plusieurs domaines en émergence dont l'interaction humain robot utilisant des scénarii réalistes lors de l'entraînement des opérateurs où il y a des problématiques de symptômes réduisant leurs capacités motrice et fonctionnelle. À la fin de ce projet, ces travaux de recherche fourniront un instrument d'automatisation des évaluations d'un ordonnancement d'une tâche industrielle pour la commande intelligente d'un robot industriel en utilisant des capteurs portables sans fils.

1.5 Échéancier

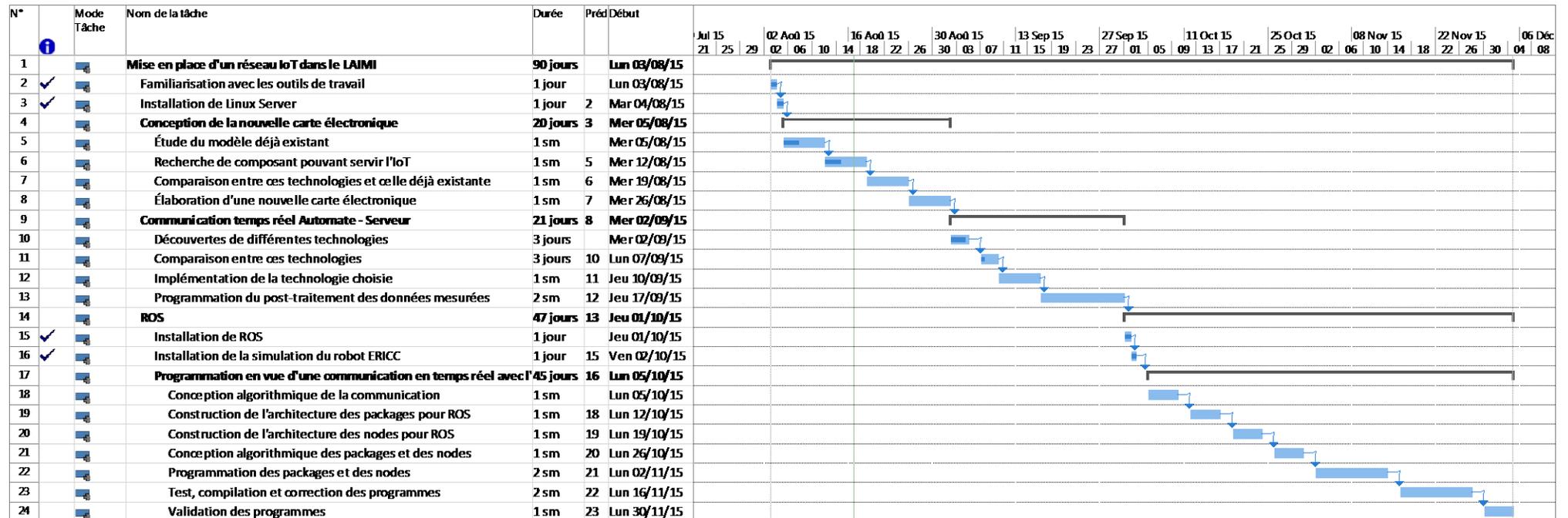


Figure 5 : Diagramme de Gantt de la réalisation du projet



2.0 Revue de littérature

2.1 Les commandes utilisées pour piloter les robots

2.1.1 Les commandes visuelles

Dans beaucoup de systèmes actuels, les robots sont commandés par l'opérateur grâce à des capteurs visuels : caméras ou Kinect pour ne citer qu'eux. L'avantage de ce type de commande est qu'elle permet à la fois au robot d'obéir aux ordres de l'opérateur mais aussi de connaître en permanence la position de son opérateur et donc de ne pas le blesser.

Le problème de ce genre de technologie est qu'en situation de cellule de travail hybride, ce type de technologie voit son efficacité diminuer. En effet, dans ce type de poste de travail, opérateur et robot doivent travailler de concert sur une même tâche, donc l'opérateur ne peut pas en permanence surveiller le robot ou sa propre posture. Il est possible que les gestes commandant le robot puissent correspondre à une position que doit prendre l'opérateur pour accomplir sa tâche. Si cette position est mal interprétée, alors le robot effectuera une action qui risque de blesser l'opérateur. C'est pour cela qu'il est nécessaire de rendre les robots assez intelligents pour reconnaître les bons signes au bon moment.

2.1.2 Les commandes vocales

Ce type de commande est aussi largement répandu car la voix est le premier outil de communication des êtres humains, il est donc compréhensible que ce système ait été utilisé pour piloter un robot. Les systèmes de reconnaissance vocale sont assez développés pour commander les robots avec des ordres simples et précis mais ces ordres sont encore du niveau de langage que l'on peut avoir avec un chien : « avance », « recule », « prend la pièce A ».

Même si des recherches sont organisées pour améliorer le vocabulaire des robots, cette technologie est à la fois coûteuse et demande un environnement de travail calme voire peu bruyant pour que l'opérateur n'ait pas à hurler pour se faire comprendre. De plus, la voix peut changer en cas de maladie ou sous le coup de l'émotion par exemple. Si la voix s'en trouve trop déformée, alors le robot ne la reconnaît plus et ne peut plus effectuer son travail. Ensuite ce genre de procédé nécessite une programmation de plusieurs langues, et nous savons très bien que dans chaque langue, il y a des accents différents qui pourraient rendre un ordre incompréhensible pour le robot.

2.1.3 Les commandes gestuelles

Ici, il ne s'agit pas de commandes perçues par le robot via une caméra ou tout autre appareil visuel mais de commandes captées par des capteurs sur l'opérateur, comme des accéléromètres, des centrales inertielles ou des gyroscopes. L'utilisation de ces capteurs se montre pertinente dans les cas où les types de commandes précédemment cités font défaut : dans les milieux bruyants où lorsque la visibilité peut être perturbée. En plus, ce sont des capteurs fiables pour la plupart, demandant peu d'entretien et peu de programmation pour le robot puisque les signaux sont déjà sous forme analogique ou numérique (il n'est pas nécessaire de convertir des images ou de créer d'immenses banques de données).



Cependant, ce type de capteur ne se situe pas sur le robot mais sur l'opérateur et peut donc entraver sa liberté de mouvement : avoir vingt fils qui sortent du dos limite forcément vos mouvements. Ainsi, il peut être nécessaire de développer des systèmes de communication avec le moins de fils possible (en utilisant le multiplexage par exemple) voire de la communication sans fil (comme les technologies RFID, Bluetooth ou Wifi). En plus de cela, le robot ne connaît virtuellement que les points fournis par ces capteurs. Supposons que seules les mains soient équipées de tels capteurs, où se situe le reste du corps ? Il faut donc créer tout un système de repères pour permettre au robot d'éviter de collisionner l'être humain pendant sa tâche.

2.2 La coopération Humain/Robot dans un même espace de travail

2.2.1 Cobotique

Ce terme, « cobot », est un néologisme inventé par Michael Peshkin et J. Edward Colgate (Peshkin & Colgate, 1999) qui est la contraction des deux mots anglais : « collaborative » et « robot » et qui définit les robots qui interagissent directement avec l'humain et partagent avec celui-ci la même aire de travail. Les cobots sont différents des robots téléguidés dans la mesure où ces derniers sont uniquement contrôlés par l'être humain. Une autre grande différence avec des robots intelligents construits avant le développement de ce concept est l'utilisation de la réalité virtuelle pour l'exécution de tâches. Ainsi, il est possible pour les humains de réaliser, grâce au cobot, des tâches qui demandent un niveau de force, d'endurance ou de précision trop élevé pour un être humain, et ainsi d'éviter l'apparition de TMS (Troubles Musculo-Squelettiques) chez l'opérateur.

Depuis l'apparition de ce principe, des applications concrètes ont été développées comme par exemple l'utilisation d'un cobot dans le cadre d'opérations chirurgicales (Riwan, et al., 2011) en particulier sur le rachis (Riwan, et al., 2011).

2.2.2 Troisième main

Les robots de type « troisième main » ont pour rôle d'assister l'opérateur dans son travail ou sa tâche en la lui facilitant. Ce type de robot se distingue du premier dans le cadre où le cobot est une extension de l'opérateur et l'aide à combler une lacune alors que la troisième main assiste l'opérateur dans une tâche qu'il sache faire. Ainsi, un robot troisième main va donner l'outil nécessaire à l'opérateur pour qu'il réalise sa tâche puis reprendra l'outil et en fournira un autre, en fonction du comportement de l'utilisateur. C'est ce procédé qui a été développé par (Eweron, et al., 2015).

2.3 Éviter l'apparition de problèmes de santé

Il est toujours regrettable qu'un être humain se blesse au travail, que ce soit une blessure aiguë (subie intensément sur une courte période) ou chronique (subie sur une longue période ou de façon répétitive).

2.3.1 Les blessures dues au travail

D'après la CSST (Commission de la Santé et de la Sécurité du Travail), en 2013 au Québec, 83458 personnes ont été victimes d'un accident de travail et 63 en sont décédées. Dans certains domaines industriels, il est fortement probable que l'utilisation de robots aurait pu éviter ces accidents. En effet, avec sa force, le robot ne perdrait pas le



contrôle d'un outil dangereux et donc on pourrait limiter la quantité de blessures dues à l'inattention ou à une perte de contrôle. C'est pour cela que la première tâche qu'un robot doit réaliser est la protection de l'être humain.

2.3.2 Les troubles musculo-squelettiques

L'utilisation des robots à aujourd'hui pour but de diminuer le risque de survenue de blessures chez les opérateurs. Les TMS (Troubles Musculo-Squelettiques) sont des maladies apparaissant lors de la répétition prolongée d'un même mouvement ou encore l'exécution d'un mauvais geste pour faire une tâche (comme par exemple se pencher en avant pour ramasser quelque chose de lourd au sol plutôt que de se tenir droit et descendre sur ses genoux).

Ainsi de plus en plus de robots sont utilisés pour limiter ces risques (Reinhart & Spillner) et certains chercheurs développent des simulations afin de prévoir les comportements des opérateurs à leur tâche afin de régler le robot au mieux pour que ce soit lui, et non l'opérateur, qui effectue la tâche la plus pénible (Maurice, Measson, Padois, & Bidaud, 2012).

2.4 Le retour haptique

Le principe du retour haptique est d'utiliser le sens du toucher pour transmettre à l'homme des informations venant de son environnement. Dans notre cas, un tel système permettrait d'avertir l'opérateur de l'approche d'un robot et la direction depuis laquelle il arrive. Pour cela, plusieurs axes de recherches dans différents domaines ont été réalisées : afin de permettre à des personnes amputées de retrouver les sensations par exemple, de communiquer via un réseau d'ordinateur (Rovers & van Essen, 2006) ou comme expliqué précédemment, afin de prévenir un danger et connaître sa direction afin de l'éviter.

Dans la littérature, il est possible de trouver des études de semelles (Velázquez, Bazan, & Varona, Wearable Electronic Tactile Display for the Foot, 2013) et de comportements de personnes vis-à-vis de la détection grâce aux sens du pied.

3.0 Comparaison de différents microcontrôleurs et modules Wifi puis choix du meilleur élément

Dans le cadre de ce projet, différents microcontrôleurs et modules Wifi ont été comparés et sélectionnés afin de permettre la modification du circuit imprimé du LAIMI ou la conception d'un nouveau, et ce en fonction du choix effectué.

Afin de choisir le dispositif convenant le mieux à notre application, nous nous appuyerons sur les critères de sélection suivants :

- Au moins 6 entrées/sorties ADC (*Analog to Digital Converter*), afin de connecter les différents capteurs utilisés par d'autres applications
- Une faible consommation d'énergie, afin de pouvoir fonctionner sur batterie le plus longtemps possible
- Des entrées SPI/I²C, afin de pouvoir connecter et utiliser différents capteurs ou dispositifs utilisant ces protocoles de communication
- Des sorties digitales, afin de pouvoir générer des signaux PWM pour contrôler des moteurs

	6MIG899 – Projet d'application	Rapport final	Étienne Polesel	1.0.0-R
---	--------------------------------	---------------	-----------------	---------

- Un port UART ou équivalent pour pouvoir utiliser un module Wifi différent du microcontrôleur ou une antenne si le module Wifi est inclus dans le microcontrôleur
- La facilité de soudure ou d'installation ainsi que de programmation

Les dispositifs comparés sont les suivants :

- ESP8266-1 / ESP8266-12 (microcontrôleur + Wifi)
- SESUB-PAN-T2541 de TDK (Bluetooth)
- RN171 / RN 1723 de Microchip (microcontrôleur + Wifi)
- LBWA1ZV1CD-716 de Murata (microcontrôleur + Wifi)
- WF121 de BlueGiga (microcontrôleur + Wifi + coexistence Bluetooth)

Parmi ces composants, certains ne sont que des microcontrôleurs comme le PIC24 ou que des modules Wifi comme l'ESP8266-1, alors que d'autres sont des tout-en-un comme les RN171 et 1723, le LBWA1ZV12CD-716, l'ESP8266-12 ou le WF121.

3.1 Étude des composants

3.1.1 La solution PIC24 + antenne

La version actuelle est constituée d'un PIC24 auquel nous pouvons ajouter différents capteurs et un module Bluetooth. Le problème de cette version est que le module Bluetooth est la récupération d'une version commerciale de clé USB Bluetooth qui servait à l'origine à équiper un ordinateur d'une antenne Bluetooth. Il est évident qu'une telle solution n'est pas viable sur le long terme. Ainsi, nous nous intéresserons dans cette partie du rapport à une technologie qui nous permettrait de garder le circuit imprimé actuel en ne faisant que quelques modifications mineures afin d'y intégrer une antenne, qu'elle soit Bluetooth ou Wifi.

3.1.1.1 ESP8266-1/ESP8266-12

Ainsi, parmi les modules Wifi existants, nous avons choisi d'étudier l'ESP8266-1, version comprenant 8 connecteurs ainsi que l'ESP8266-12, avec 16 connecteurs. Ces modules ont l'avantage d'être petits et facile à installer. Il suffit de relier les masses, relier le VCC et les ports UART pour qu'ils soient opérationnels. De plus, ils possèdent leurs propres adresses internet et les protocoles nécessaires à leur utilisation. Ils sont légers, 3g, petits, 21.1mm de long pour 13.2 de large, et possèdent plusieurs ports GPIO (2 pour la version 3 et 10 pour la 12). Ils peuvent être aussi connectés à un ordinateur via un câble FTDI afin d'y installer des mises à jours. Il est toujours possible de piloter ces modules possédant un petit microprocesseur de 32 bits grâce aux commandes AT, très fortement documentées sur internet. Cependant, ces modules consomment énormément d'énergie et nécessitent une alimentation supplémentaire, ou en tout cas, doivent être directement connectés à la batterie. D'autres avantages à leur accorder sont qu'ils ne sont vraiment pas chers (entre 5 et 6\$ l'unité en fonction du fournisseur pour la version 3) et qu'ils peuvent être utilisés avec un Arduino, permettant ainsi de faire des tests sans risquer d'abimer de matériel.

	6MIG899 – Projet d'application	Rapport final	Étienne Polesel	1.0.0-R
---	--------------------------------	---------------	-----------------	---------

Ces modules ne peuvent être utilisés que comme une antenne à ajouter au système déjà existant car ils ne possèdent pas assez de ports pour correspondre aux exigences du problème. L'utilisation de ces modules permettrait donc de réutiliser le PIC24 et il suffit de les souder sur la plaque existante pour l'utiliser. Cependant, il faudrait gérer l'alimentation de ces modules, ce qui risque de compliquer un peu leur implantation.

3.1.2 Composants Microcontrôleurs + Antennes

Beaucoup d'objets soumis à comparaison dans ce rapport sont en lien direct avec l'IoT (*Internet of Things*) et possèdent donc un module de communication Bluetooth ou Wifi en plus d'un microcontrôleur. Deux choix s'offrent à nous : le premier consisterait à ignorer la partie microcontrôleur des modules et n'utiliser que la partie communication, mais il s'agirait d'un gaspillage du potentiel de ces modules, quant au second, il consiste à repenser le circuit imprimé du LAIMI, voire en concevoir un nouveau, qui permettrait l'utilisation à la fois du module de communication et du microcontrôleur inclus dans ces modules.

Les modules vont être présentés en fonction de leur mode de communication. Dans chacune des descriptions, nous présenterons d'abord les généralités du module, puis les caractéristiques de la partie communicante et enfin, celles du microcontrôleur.

3.1.2.1 Bluetooth

Dans un premier temps, le circuit fut conçu pour fonctionner avec une antenne Bluetooth. Nous pouvons garder cette technologie car elle nous permet de ne modifier qu'une toute petite partie du circuit et des programmes déjà existants. Nous ne verrons pas les modules venant d'une clé USB comme c'est actuellement le cas. Cela se justifie par le fait que ce sont des produits finis et pas forcement adaptés à notre application à cause, par exemple de la taille ou de la consommation en électricité ; car rappelons le, le PIC24 fonctionne avec une tension de 3.3V alors qu'un module Bluetooth conçu pour un ordinateur sera alimenté par 5V.

SESUB-PAN-T2541

D'abord, il s'agit d'un dispositif prêt à être utilisé, remarquable pour sa petite taille : 5.6 par 4.6mm pour 1 d'épaisseur avec une tension d'alimentation de 3.3V. Un des inconvénients pour inclure ce module est qu'il se soude par en dessous, compliquant l'installation de celui-ci sur le circuit imprimé, mais pour sa décharge, les connecteurs intérieurs ne sont que des masses et l'ensemble des connecteurs essentiels à son fonctionnement se situe à la périphérie du module.

Quant au microcontrôleur de ce dispositif, il s'agit d'un 8051 Core possédant 256KB de mémoire flash et de 8KB de RAM. Ce microcontrôleur est complété par 8 ADC de 12 bits sur les ports P1.x. Quant à la communication, elle peut être UART ou SPI, les deux pouvant être simultanées puisqu'elles n'utilisent pas les mêmes ports et il existe la possibilité d'utiliser une horloge de 32kHz intégrée à ce module

Pour ce qui est de l'antenne, elle est libre de conception, de sorte que nous pouvons soit la graver sur la piste, soit utiliser un fil.

	6MIG899 – Projet d'application	Rapport final	Étienne Polesel	1.0.0-R
---	--------------------------------	---------------	-----------------	---------

3.1.2.2 Wifi

Afin de permettre à ce circuit de communiquer, l'option choisie est une antenne Bluetooth mais le LAIMI souhaite utiliser une antenne Wifi. En effet, le Wifi a l'avantage d'être universel et peut s'exporter facilement. De plus, sa portée est supérieure.

RN171 / RN 1723

Ces modules sont sensiblement les mêmes, à ceci près que la version 1723 possède un mode de veille (le *Doze Mode*) et un certificat de sécurité (le TLS 1.0, 1.1 et 1.2) supplémentaires. Je me baserai donc sur cette version pour la comparaison avec d'autres modules.

Ce module est plutôt de grande taille (26.67mm x 17.78mm x 3.18) et possède son propre régulateur de tension, un cristal et tout le nécessaire pour son utilisation rapide et facile. En plus, il possède un mode veille très basse consommation (4µA). 49 ports sont disponibles dont 10 entrées/sortie GPIO et 8 entrées ADC de 14 bits sur une échelle de 0 à 400mV (ils peuvent aller jusqu'à 1.2V mais saturent à 0.4), apportant une résolution de 12µV.

Quant au microprocesseur, il s'agit d'un système 32bits accompagné d'une mémoire ROM de 2MB et d'une RAM de 128KB. Mais il est possible de le relier avec une carte mémoire flash pour augmenter ces capacités.

Quant au Wifi, les quatre protocoles de sécurité les plus courants sont présents : le WEP-128, le WPA-PSK (TKIP), le WPA2-PSK et le TLS 1.0, 1.1 et 1.2. La fréquence est de 2.412 à 2.462 GHz avec les modulations 802.11b/g. La transmission de données aérienne est de 1-11 Mbps pour le 802.11b et de 6-54 Mbps pour le 802.11g. L'antenne peut être gravée sur le circuit ou être un simple fil.

LBWA1ZV1CD-716

Ce module est lui aussi de toute petite taille (10mm x 7.9mm x 1.25mm) et se soude par en-dessous comme le SESUB-PAN-T2541 mais les broches intérieures ne sont pas toutes de masses, même s'il y en a beaucoup. Il possède 76 broches dont 25 masses, 23 ports I/O sélectionnables, et la possibilité de connecter un phototransistor pour recevoir des données BlinkUp ainsi qu'une LED bicolore de fonctionnement. Parmi les 23 I/O, 11 sont des ADC 12 bits, 2 des DAC, 7 des PWM, il y a 5 voies UART, 2 I²C et 2 SPI en plus d'une voie SPI dédiée. Évidemment, certains ports sont utilisés plusieurs fois mais ils restent suffisants pour notre application. Quant aux conditions d'utilisation du module, il est conseillé de lui appliquer une tension de 3.3V régulée. Ce module consomme beaucoup de courant : en condition d'utilisation normale avec le Wifi d'actif, il consomme 80mA (consommation qui peut monter jusqu'à 250mA) et en veille, 6µA. Les ports I/O supportent les 3.3V.

Le microcontrôleur est un STM32F405 de STMicroelectronics avec un processeur ARM Cortex-M4, système 32 bits.

Le module Wifi de ce dispositif est un BCM43362 de Broadcom. L'antenne associée à ce dispositif peut être imprimée sur circuit ou être un simple fil. Pour les protocoles Wifi, les modulations disponibles sont 802.11b/g/n, fonctionnant toutes aux fréquences de 2400-2483.5MHz et dont le débit est de 11, 54 et 65 Mbps respectivement. Quant aux protocoles de sécurité, l'AES, le TKIP, le WAP et la WPA2 sont supportés par ce dispositif.

	6MIG899 – Projet d'application	Rapport final	Étienne Polesel	1.0.0-R
---	--------------------------------	---------------	-----------------	---------

WF121-A-V2

Bien que de taille respectable, il est dans la même gamme que l'ESP8266-12. Bien qu'il y ait des broches en dessous du module, elles ne sont pas utilisées, il s'agit des broches pour le réglage en usine. Il possède 55 broches dont 5 masses et 38 GPIO programmables. Parmi ces GPIO, nous pouvons noter la présence de 10 ADC, 4 UART, 3 I²C, 2 SPI, 1 USB, 1 Ethernet, tous les ports nécessaires à la programmation ICSP du microcontrôleur et au débogage du module Wifi. Par rapport à son alimentation, il faut lui appliquer une tension de 3.3V que ce soit pour le microcontrôleur ou pour l'antenne. Il est assez gourmand en courant en cas de fonctionnement intense (jusqu'à 400 mA) mais en moyenne, il ne consomme que 100mA pour le CPU et la consommation du module Wifi reste dans la moyenne des autres modules.

Le microcontrôleur associé à ce module est composé d'un processeur MIPS4K de 80 MHz, d'une mémoire SRAM de 128kB et d'une mémoire flash de 512kB.

Le module Wifi possède tous les protocoles nécessaires au bon fonctionnement du module et ce afin d'assurer l'aspect prêt-à-l'emploi de ce module. Il peut supporter entre 1 et 11 canaux de communication et peut utiliser les standards 802.11 b/g/n avec une fréquence de fonctionnement de 2412 à 2472 MHz. En plus, il est possible de faire coexister Wi-Fi et Bluetooth. En effet, le module peut se voir greffer un module Bluetooth et gérer la communication et l'ordre de priorité entre le Wi-Fi et le Bluetooth.

3.2 Comparaison

		ESP8266-1	ESP8266-12	SESUB-PAN-T2541	RN171	RN1723	LBWA1ZV1C D-716	WF121	
Caractéristiques physiques	Dimension (Long. x Larg. x H en mm)	17.45 x 12.2 x ?	24.0 x 16.0 x ?	5.6 x 4.6 x 1.0	26.67 x 17.78 x 3.18	26.67 x 17.78 x 3.18	10 x 7.9 x 1.25	26.2 x 15.4 x 2.1	
Caractéristiques électriques	Courant d'alimentation	Maximal (mA)	500	500	20.6	120	120	250	400
		Minimal (mode veille) (µA)	Inconnu	Inconnu	1.2	4	4	6	10
	Tension d'alimentation (V)	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
	Consommation en courant pour la transmission (mA)	215	215		190	120		143	
	Consommation en courant pour la réception (mA)	62	62		40	40		127	
Gestion d'alimentation		Non	Non	Non	Oui	Oui	Oui	Non	
Mode de communication	Technologie	Wifi	Wifi	Bluetooth	Wifi	Wifi	Wifi	Wifi	
	Modulation	802.11 b/g/n	802.11 b/g/n		802.11 b/g	802.11 b/g	802.11b/g/n	802.11 b/g/n	
	Protocoles de sécurité	Inconnu	Inconnu	Inconnu	WEP-128, WPA-PSK (TKIP), WPA2-PSK (AES)	WEP-128, WPA-PSK (TKIP), WPA2-PSK (AES), TLS 1.0, 1.1 et 1.2	Inconnu		
Connectique	Nombre de broche	8	16	36 + 35 (masses)	49	49	76	55	
	Type des connections	Ronds percés	Ronds percés	Pastilles ronde	Demi-ronds percés	Demis-ronds percés	Pastilles carrées	Demi-ronds percés	
	Taille des connecteurs	Inconnu	Inconnu	0.28 mm	0.05 mm	0.05 mm	0.8*0.8mm 0.35*0.8mm 0.65*0.8mm	0.9*1.0mm 2*0.9mm 2.6*6.1mm	

Périphériques	ADC	Nombre	0	1	8	8	8	10	10
		Type	Ne s'applique pas	Inconnu	12 bits	14 bits	14 bits	12 bits	10 bits
		Tension	NAP	Inconnu	Inconnu	400mV	400mV	3.3V	3.3V
	PWM		non	non	non	non	non	Oui (7)	Programmables
	Horloge	Présence	non	non	oui	oui	oui	non	Oui
Fréquence (kHz)				32 768	Inconnu	Inconnu		26 000	
Budget	Fournis par Digkey.ca		non	oui	oui	oui	oui	oui	oui
	Autre fournisseur		Sparkfun						
	Prix à l'unité (dollar canadien)		6.95	11.34	27.97	37.59	23.91	29.66	42.62
Développement	Possibilité d'implanter un code		Oui	Oui	Oui	Non	Non	Oui	Oui
	Langage de programmation		Luna/AT/C	Luna/AT/C	C			Squirrel	C
	IDE		Arduino IDE	Arduino IDE	BLE-Stack SDK			ElectricImp IDE	Possibilité de le connecter en ICSP
	Configuration					Commandes ASCII via Wifi ou UART	Commandes ASCII via Wifi ou UART		

Tableau 1 : Comparaison entre les différents composants pouvant être utilisés dans le projet



3.3 Choix effectué

À la suite de la comparaison des différents modules, le WF121 semble être le candidat offrant les meilleurs compromis. Tout d'abord, il s'agit d'un dispositif prêt-à-l'emploi comprenant à la fois un microcontrôleur, une antenne Wifi et un dispositif de coexistence Bluetooth. De plus, il possède un nombre suffisant d'ADC pour notre application et les signaux PWM pourront être générés à partir de l'horloge incluse dans le module. En plus de cela, il est facilement programmable avec le matériel du laboratoire puisque nous pourrions réutiliser le programmeur/débogueur de Microchip pour programmer et déboguer ce module. Cependant, son prix à l'unité reste élevé mais diminue lors d'une commande en plus grandes quantités (42.62\$ à l'unité mais 25.94 par 500) chez Digikey en tout cas.

3.4 Batterie et recharge sans fil

3.4.1 Batterie

Du fait de sa consommation relativement faible, il est envisageable d'alimenter ce système grâce à une batterie. Cependant, ce module n'en possède pas, ni ne possède de système de protection de batterie. Nous proposons donc la possibilité d'ajouter au circuit imprimé contenant ce module, un dispositif de rechargement de batterie. Le LAIMI possède déjà un tel dispositif utilisé par d'autres circuits créés par le laboratoire dont nous pourrions nous inspirer afin de l'intégrer sur ce circuit.

De plus, comme il est prévu d'utiliser des batteries Lithium-Polymères, il faut prévoir une protection en courant pour celles-ci. Dans cette catégorie de batterie, il existe deux modèles : avec et sans protection. Afin de pouvoir utiliser indifféremment ces deux types de batterie, il est prévu d'équiper le circuit imprimé d'un dispositif de protection.

3.4.2 Rechargement sans fil

Toujours dans l'optique de rendre ce système le plus portable possible, la solution du rechargement sans fil de la batterie a été envisagée. Nous ne nous attarderons pas sur la théorie permettant l'utilisation de cette technologie mais sur son utilisation pratique dans notre système. Rappelons tout de même que le principe est d'alimenter un solénoïde avec un courant alternatif afin qu'il produise un champ magnétique, champs qui est récupéré par un autre solénoïde qui crée alors du courant, lequel courant est utilisé, dans notre cas, pour recharger la batterie alimentant notre système.

Il existe déjà sur le marché des systèmes préconstruits permettant la réalisation d'un tel système mais cela implique d'ajouter un circuit imprimé dans le système alors que nous souhaitons rendre le système le plus petit possible. Une telle solution est présentée par la Figure 6.

Le PCB gérant l'utilisation de la batterie est composé de deux éléments distincts mais d'égale importance : le chargeur de la batterie et la protection en courant.

Le chargeur de batterie (dans le cas de la Figure 6, il s'agit du TP4056) est un dispositif utilisé afin de protéger une batterie de type LiPo (Lithium – Polymères) d'une surcharge ou d'une surdécharge. La protection en courant sert à faire en sorte que lorsque du courant est amené au système pendant un temps de rechargement, ce courant soit utilisé par le système et recharge la batterie. Si aucun courant n'est apporté au système, alors c'est la batterie qui sert de source d'énergie.

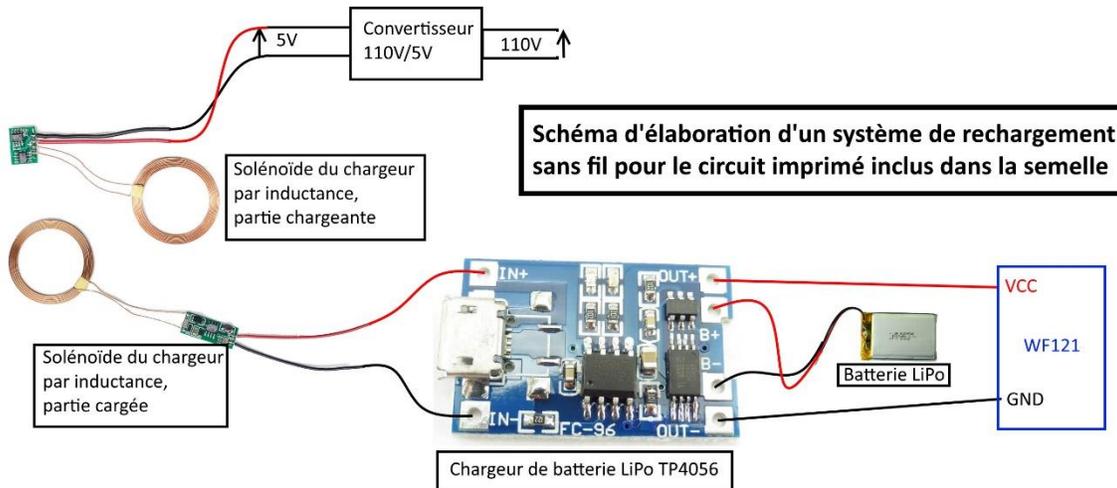


Figure 6 : Schéma d'élaboration d'un système de rechargement sans fil pour le circuit imprimé inclus dans la semelle

Le dispositif présenté par la Figure 6 est un dispositif clé en main. Afin de dépendre au minimum de matériel extérieur, nous avons envisagé de concevoir notre propre système de rechargement sans-fil. Actuellement, il y a deux solutions pour le convertisseur 110V/5V : une alimentation d'ordinateur ou un petit transformateur 5V qu'on trouve facilement dans le commerce. Ces deux solutions offrent l'avantage de fournir une tension régulée de 5V qui permet d'alimenter la partie émettrice du module de recharge. La distinction se trouve dans la puissance qui peut être fournie et dans la possibilité de connexion : alors que le chargeur fournit une puissance de quelques Watts (suffisante pour un ou deux chargeurs), l'alimentation d'ordinateur peut fournir jusqu'à 600W de puissance, ce qui offre la possibilité de charger de nombreux modules en même temps, de plus, elle pourrait être intégrée à l'armoire qui héberge de serveur du LAIMI et offrir un système primaire de ventilation.

À cette source d'alimentation est connectée la partie émettrice d'un module de recharge sans fil trouvé dans le commerce. Il nous serait possible de développer notre propre système mais cela fera l'objet de recherches ultérieures. Le signal émis par le solénoïde de l'émetteur est récupéré par un autre solénoïde dont la forme peut être adaptée à nos besoins. Pour l'instant, le choix a été fait d'utiliser un solénoïde de même diamètre intérieur et d'un nombre de spires identique au solénoïde de l'émetteur. Un autre modèle devra être conçu pour correspondre au mieux à l'emplacement qu'il occupera dans la semelle.

La tension aux bornes de ce solénoïde est alternative. Or le système est alimenté par une tension de 3.3V continue. Il faut donc rendre le signal continu et diminuer sa valeur. Pour ce faire, nous utilisons un pont de diodes, un condensateur pour lisser la tension, une diode Zener de 5,1V pour éviter les surtensions qui pourraient endommager le dernier composant de cette chaîne : le gestionnaire de charge de la batterie. La partie alimentation du circuit imprimé est présentée Figure 7.

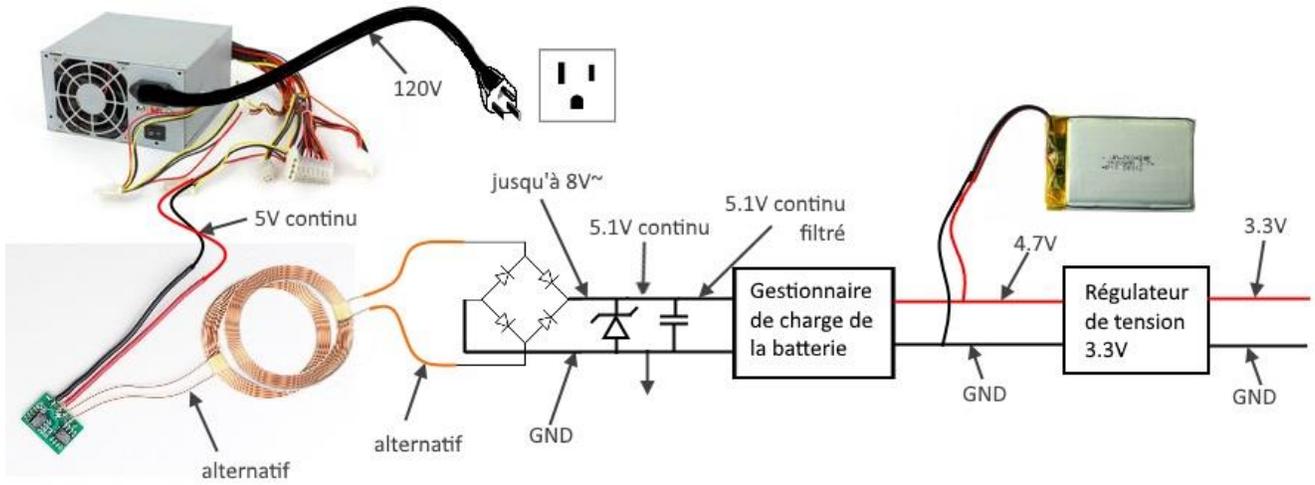


Figure 7 : Alimentation réalisée pour l'application



4.0 Conception du circuit imprimé

Après avoir déterminé qu'il était possible de remplacer le microprocesseur PIC24FJ256GB210 par un module combinant Wi-Fi et microcontrôleur. Ce nouveau module est le modèle WF121 de BlueGiga. Rappelons que ce module répond à tous les critères de sélections qui sont :

- Programmation en ICSP
- Au moins 6 entrées analogiques pour les dispositifs suivants : l'humidimètre, le baromètre, 4 capteurs de forces
- Un port I²C pour l'IMU
- Au moins 3 PWM pour des vibrateurs
- Fonctionnement à 3.3V
- Basse consommation d'énergie

De plus, d'autres critères se greffent pour la conception de ce nouveau circuit imprimé :

- Rechargement sans fil
- Le plus petit possible (la taille idéale serait de 24*36mm maximum)
- Coûtant moins de 100\$

Nous développerons les choix qui ont été faits ainsi que la conception réalisée à l'aide du logiciel *Altium*. Dans un premier temps nous verrons les composants choisis, puis nous verrons les branchements réalisés dans *Altium*. Ensuite nous verrons le circuit imprimé conçu et justifierons les choix entrepris et enfin nous rédigerons les directives d'utilisation de ce nouveau circuit et les modifications qu'il pourrait être nécessaire d'apporter à ce circuit.

4.1 Les composants

Ce nouveau circuit imprimé utilise des composants déjà utilisés ou possédés par le laboratoire pour la plupart. La première raison est d'ordre financière, il n'y aura pas à commander de nouveaux composants. La seconde est pratique, il est ainsi possible de s'appuyer sur le circuit déjà conçu afin de limiter le nombre d'erreurs potentielles qui pourraient apparaître parce que le composant est inconnu par les membres du laboratoire. Les seuls composants complètement nouveaux sont le module Wifi WF121, et deux dispositifs pour la protection de la batterie LiPo (Lithium Polymères) utilisée pour ce dispositif.

4.2 Les anciens composants

Parmi ces composants, nous retrouvons le baromètre, l'humidimètre et la centrale inertielle. Nous retrouvons aussi le gestionnaire de charge de la batterie et des connecteurs pour des capteurs analogiques entre autres.



4.2.1.1 Le baromètre, ASB1.200VRTNH19

Il s'agit d'un baromètre analogique du constructeur TDK. La seule particularité est la présence d'un condensateur entre les bornes d'alimentation. Il se connecte via quatre bornes mais seulement trois sont utilisées :

- V_{supply} pour l'alimentation en 3.3V
- GND, la masse
- V_{out} , la tension en sortie du capteur.

La pression est donnée par la formule suivante :

$$p_{meas} = \frac{p_{r,max} - p_{r,min}}{V_{FS}} * \left(\frac{V_{meas}}{V_{Supply}} - V_{out,min} \right) + p_{r,min}$$

Avec pour valeur :

V_{meas} : tension mesurée sur la borne V_{out}

p_{meas} : pression mesurée par le capteur

$V_{out,min} = V_{out}(p_{r,min})$

$p_{r,max} = 1.2$ bar ou 17.4 psi

$V_{FS} = V_{out}(p_{r,max}) - V_{out}(p_{r,min})$

$p_{r,min} = 0.2$ bar ou 2.9 psi

La Figure 8 montre la caractéristique de ce capteur.

Output characteristics

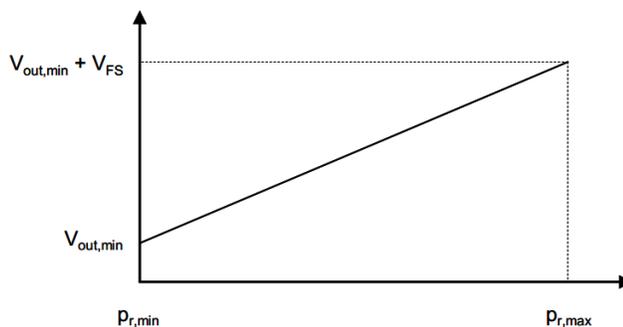


Figure 8 : Caractéristique du baromètre (tension vs pression)

4.2.1.2 L'humidimètre, HIH5030

Ce capteur comporte quatre bornes mais seulement trois sont utilisées : les deux bornes d'alimentation et la tension fournies par le capteur. La Figure 9 montre la réponse normale du capteur à 25°C et alimenté en 3.3V.



HIH-5030/5031 Series

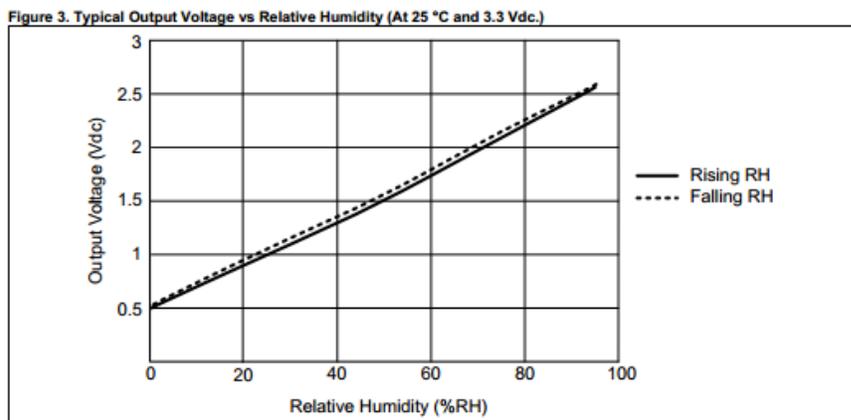


Figure 9 : Caractéristique de l'humidimètre (tension vs humidité relative)

Ces deux composants n'ont pas été repris dans la nouvelle version du circuit imprimé car il a été calculé que les variations de pressions détectables par le baromètre étaient trop importantes par rapport à celles qu'il fallait mesurer (le changement d'altitude détectable est de l'ordre de la centaine de mètres) et puis, l'humidimètre ne fournit pas d'informations pertinentes pour l'utilisation que nous allons faire de ce circuit imprimé.

4.2.1.3 La centrale inertielle, MPU9150

Ce composant est constitué de plusieurs dispositifs : un accéléromètre trois axes, un gyroscope trois axes, un magnétomètre mais aussi un thermomètre. Il est possible soit de prélever les valeurs fournies par chaque composant directement à partir des sorties analogiques, soit d'utiliser une communication I²C, ce qui nous intéresse tout particulièrement puisque le nombre d'entrées analogiques est assez limité sur le WF121. Nous ne décrivons pas ici les règles et protocoles utilisés pour communiquer via le port I²C.

4.2.1.4 Le chargeur de batterie LiPo, MCP73831

Ce dispositif est un gestionnaire linéaire de charge de batterie LiPo à une cellule. Nous attirons l'attention de l'utilisateur sur la valeur de la résistance R_{prog} . Cette résistance permet de contrôler le courant utilisé par ce dispositif pour charger la batterie et il faut utiliser la formule suivante pour calculer sa valeur :

$$I_{REG} = \frac{1000}{R_{PROG}}$$

R_{PROG} est calculé en k Ω et I_{REG} (en mA) correspond à l'intensité délivrée par la batterie en une heure.

4.2.1.5 Le régulateur de tension 3.3V, CAT6219

Ce régulateur sert à fournir en permanence une tension de 3.3V, quelle que soit la tension à l'entrée de ce dispositif comprise bien entendu dans l'intervalle accepté par le dispositif (2.3 à 5.5V).



4.2.2 Les nouveaux composants

Parmi les nouveaux composants, nous pouvons compter le module Wi-Fi bien entendu mais aussi un dispositif de protection en courant pour batterie LiPo. Ce dernier est obsolète pour les batteries qui en disposent mais essentiel pour celles qui n'en ont pas. Ainsi, nous avons agrandi le choix de batteries que nous pouvons utiliser pour alimenter le circuit électronique.

4.2.2.1 Le module Wi-Fi, WF121

Nous présenterons ici les branchements qui ont été effectués sur les bornes du module afin de répondre aux critères énoncés plus haut dans ce document.

Nous utilisons donc les deux ports I²C, le port SPI, six entrées analogiques, six GPIO (trois seuls fournissant 3.3V et trois connectés aux grilles de transistors MOSFET permettant de fournir du courant provenant directement de la batterie) et les ports de la programmation ICSP ainsi que les connecteurs d'un éventuel module Bluetooth supplémentaire. De plus, il est possible d'utiliser la fonction USB de ce dispositif et de le recharger grâce à ce port.



4.3 Plans schématiques

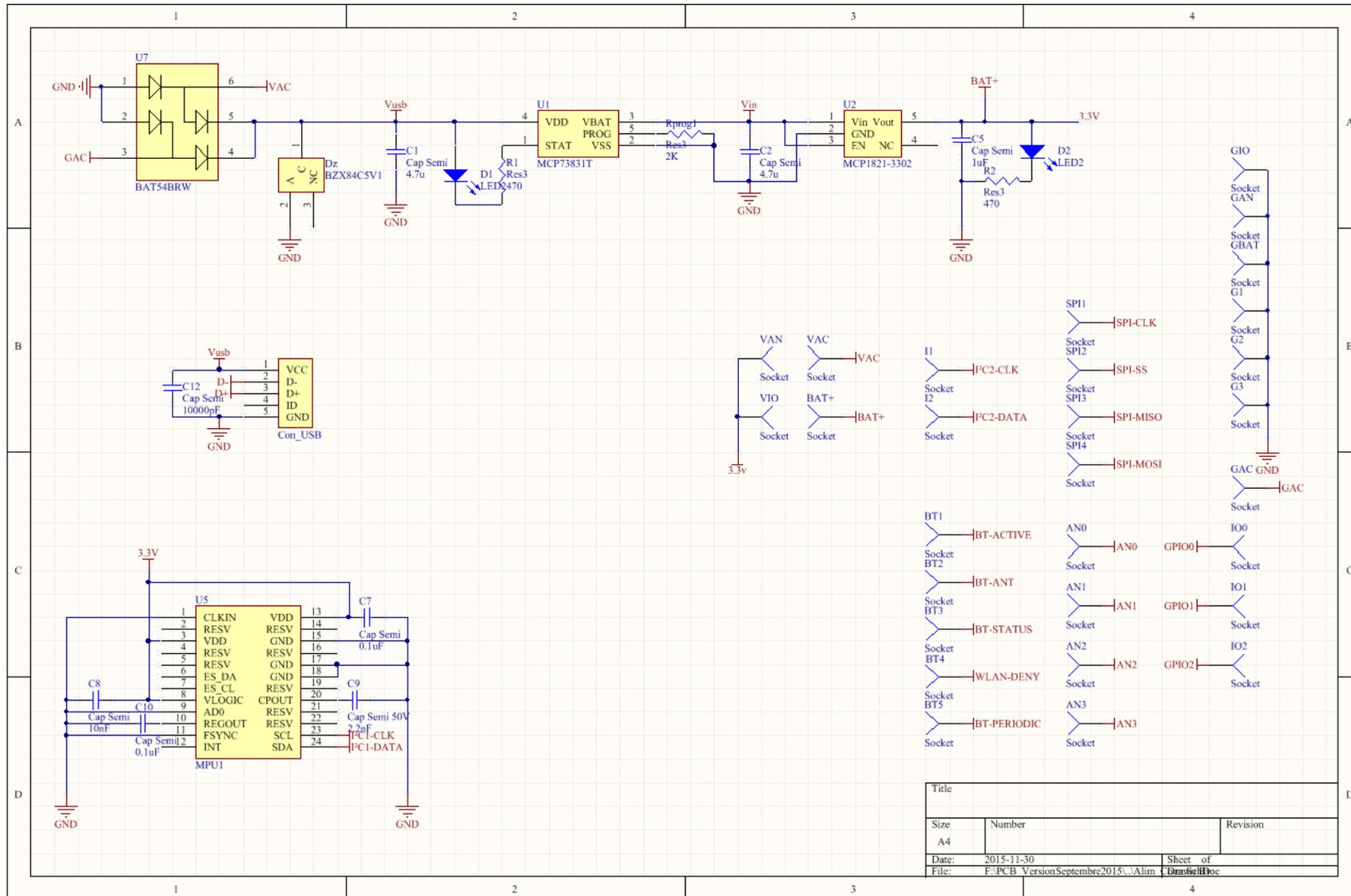


Figure 10 : Schéma du routage des connecteurs, des composants d'alimentation et de la centrale inertielle utilisés dans le nouveau circuit imprimé

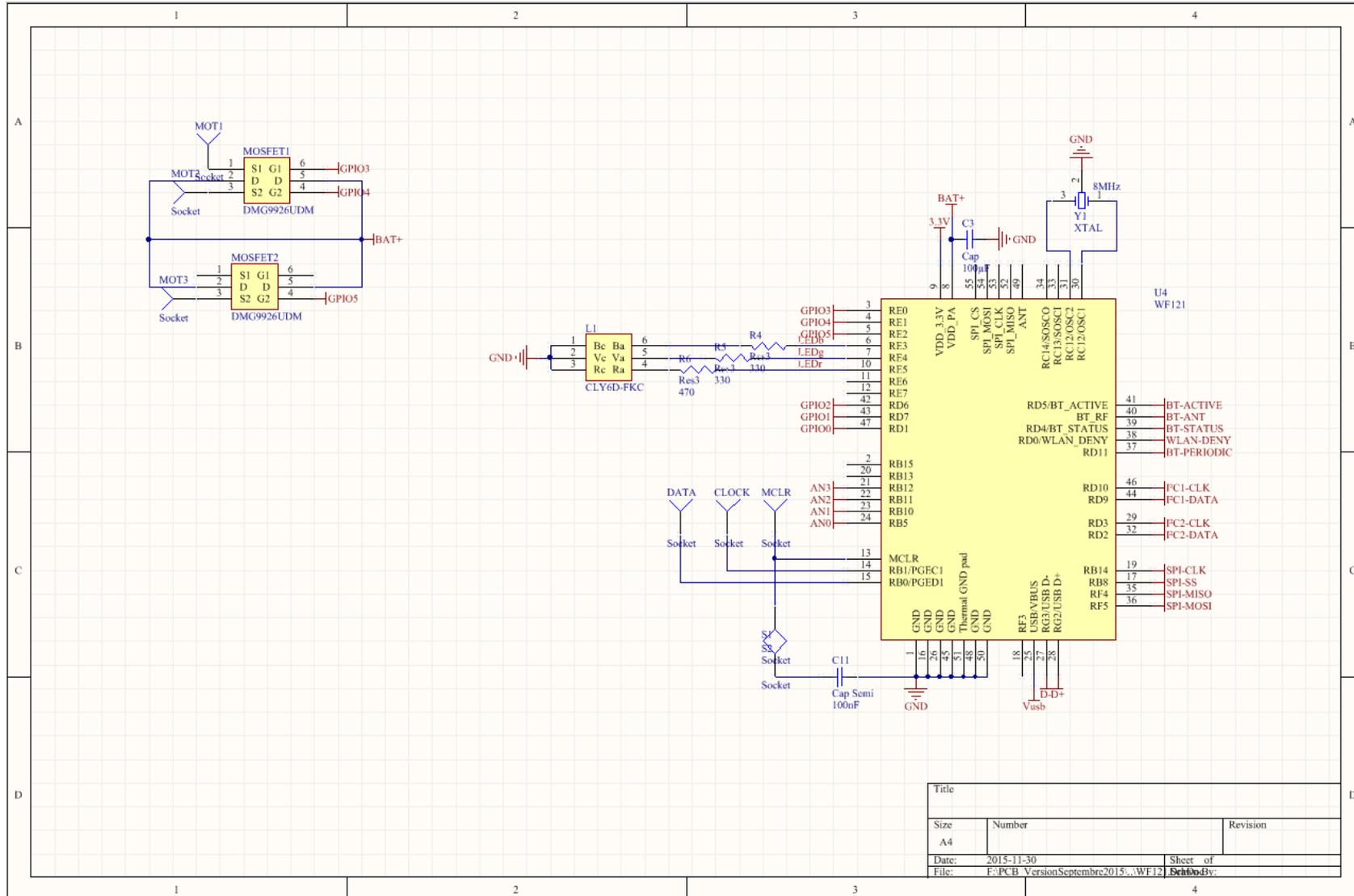


Figure 11 : Schéma du routage du microcontrôleur utilisé dans le nouveau circuit imprimé



Dans la partie haute du schéma présenté Figure 10, nous pouvons voir les composants gérant la charge de la batterie. En dessous, sur la gauche se trouvent tous les capteurs et leurs branchements. Sur la droite, se situent les différents connecteurs utilisés par d'autres éléments non représentés ici car n'appartenant pas au circuit imprimé. Il s'agit des capteurs de force et des vibreurs. La Figure 11 présente le module Wi-Fi et ces branchements. Les broches sont rangées en fonction de leur fonction.

Chaque composant ainsi que ses empreintes a été conçu et implanté dans *Altium* à partir des informations obtenues des datasheets.

4.4 Le circuit imprimé

4.4.1 Conception et élaboration

Le routage programmé et calculé par *Altium* n'était pas satisfaisant alors il a été réalisé étape par étape dans une recherche d'optimisation d'utilisation. Ainsi, les ports ont été regroupés selon leur rôle et chacun possède sa propre source d'alimentation. Afin de minimiser la quantité de cuivre à éliminer des plaques, des plans de masse ont été créés sur chaque face. Évidemment, afin d'éviter un effet condensateur de ses plans de masse à travers l'isolant de la plaque du circuit, un certain nombre de vias ont été ajoutés.

4.4.2 Résultats obtenus

4.4.2.1 Le circuit imprimé

Sur la face supérieure illustrée Figure 12 se trouvent les capteurs analogiques ainsi que le module Wi-Fi. Sur la face inférieure, Figure 13, se trouvent les composants de la gestion de la batterie. La taille finale de ce circuit imprimé est de 28*38mm, ce qui est exactement la taille du circuit existant. De nombreuses connections sont utilisées vers des dispositifs extérieurs au circuit (comme des moteurs, des capteurs de forces, un module Bluetooth et des ports de communication SPI et I²C). Rappelons que ce dispositif doit pouvoir se glisser dans un casque de sécurité ou une semelle de chaussure.

4.4.2.2 Consommation électrique

Nous considérerons ici la consommation électrique du circuit imprimé à différents modes de fonctionnement : le mode veille et le mode plein fonctionnement. En mode veille, l'ensemble des dispositifs consommerait 120µA dont 110 seraient utilisés par le module WF121. Cette valeur est prise lorsque la communication Wifi est en veille, ce qui s'effectue automatiquement lorsqu'il n'y a pas de transfert de données via l'antenne d'après la datasheet. En plein fonctionnement, l'ensemble des dispositifs consommerait 105 mA dont 100 pour le WF121 en fonctionnement moyen avec des pics de consommation pouvant atteindre 400mA lors de la transmission de données. Le LAIMI possède actuellement des batteries Lithium-Polymères de 400mAh et de 4.7V. Ainsi, nous pouvons prévoir une durée de vie en fonctionnement de 4h environ.



4.4.2.3 Brochage

Les brochages en lien avec le circuit imprimé et le module WF121 sont décrits dans le Tableau 2. Le Tableau 3 décrit quant à lui le brochage du circuit imprimé.

N° de port	Nom	Fonction	Sortie circuit imprimé	Rôle	
1, 16, 26, 45, 48, 50	GND	Masse	Gbat	Masse pour la batterie	
			G1	Masse pour la recharge sans fil	
			G2	Masse pour les capteurs analogiques	
			G3	Masse pour les GPIO	
			GMOT1, GMOT2, GMOT3	Masse pour les moteurs	
51	GND therm	Masse	GND therm	Dissipation de chaleur du module	
8	VDD_PA	Alimentation pour l'antenne	3.3V	Alimentation du module	
9	VDD_3.3V	Alimentation du module			
3	RE0	GPIO	MOT1	Sortie MOSFET	
4	RE1		MOT2		
5	RE2		MOT3		
6	RE3			GPIO pour LED bleue	
7	RE4			GPIO pour LED verte	
10	RE5			GPIO pour LED rouge	
11	RE6				
12	RE7				
42	RD6			GPIO2	Entrée/sortie
43	RD7			GPIO1	
47	RD1			GPIO0	
2	RB15	Convertisseur analogique numérique		Inutilisés	
20	RB13				
21	RB12		AN3	Entrées analogiques pour dispositifs analogiques	
22	RB11		AN2		
23	RB10		AN1		
24	RB5		AN0		
13	MCLR	Programmation ICSP du module	MCLR	Ces trois broches sont à connecter au programmeur Pickit de Microchip.	



14	PGEC1		CLOCK	Attention, afin de fonctionner correctement, il faut s'assurer de déconnecter le condensateur, sinon le programmeur ne fonctionnera pas.
15	PGED1		DATA	
18	RF3	Ports pour connexion USB	Inutilisé	
25	USB/VBUS		Tension fournie par le port USB	
27	USB D-		Transmission de données par le port USB	
28	USB D+			
17	RB8	Communication SPI	SPI2	Broche de sélection
19	RB14		SPI1	Horloge
35	RF4		SPI3	Entrée maître – Sotie esclave
36	RF5		SPI4	Sorite maître – Entrée esclave
29	RD3	Communication I ² C avec la centrale inertielle IMU	Horloge	
32	RD2		Données	
44	RD9	Communication I ² C	I2	Données
46	RD10		I1	Horloge
37	RD11	Ports pour ajouter un module de communication <i>Bluetooth</i>	BT5	Cf. la datasheet du module WF121 de BlueGiga pour plus d'information
38	WLAN_DENY		BT4	
39	BT_STATUS		B3	
40	BT_RF		BT2	
41	BT_ACTIVE		BT1	
49	ANT	Antenne WiFi	Inutilisé	
30	OSC1	Cristal externe pour cadencer la communication USB ou UART	Utilisés	
31	OSC2			
33	SOSCI	Cristal externe pour augmenter la précision de l'horloge de veille	Inutilisés	
34	SOSCO			
52	SPI_MISO	Communication SPI pour les réglages d'usines	Inutilisés	
53	SPI_CLK			
54	SPI_MOSI			
55	SPI_CS			

Tableau 2 : Brochage des connecteurs du module WF121 en lien direct avec le circuit imprimé



Nom	Rôle	Fonction
VAC	Borne positive du module de rechargement sans fil (5V)	Alimentation depuis le dispositif de recharge sans fil
GAC	Borne négative du module de rechargement sans fil	
BAT+	Borne positive de la batterie LiPo	Alimentation depuis la batterie
GBAT	Borne négative de la batterie LiPo	
MOT1	Borne positive d'un moteur à courant continu (5V, 20A max)	Branchement du premier moteur
G1	Borne négative d'un moteur à courant continu (5V, 20A max)	
MOT2	Borne positive d'un moteur à courant continu (5V, 20A max)	Branchement du second moteur
G2	Borne négative d'un moteur à courant continu (5V, 20A max)	
MOT3	Borne positive d'un moteur à courant continu (5V, 20A max)	Branchement du troisième moteur
G3	Borne négative d'un moteur à courant continu (5V, 20A max)	
VAN	Borne d'alimentation positive d'un dispositif analogique (+3.3V)	Alimentation et ports analogiques
AN0	Entrée analogique	
AN1	Entrée analogique	
AN2	Entrée analogique	
AN3	Entrée analogique	
GAN	Borne d'alimentation négative d'un dispositif analogique (0V)	Alimentation et entrées/sorties directes sur le module WF121
VIO	Borne d'alimentation positive d'un dispositif numérique (+3.3V)	
GPIO0	Entrée/sortie numérique	
GPIO1	Entrée/sortie numérique	
GPIO2	Entrée/sortie numérique	
GIO	Borne d'alimentation négative d'un dispositif numérique (0V)	Ports du module Bluetooth. Voir la datasheet du module WF121 pour plus de détail.
BT1	Bornes Bluetooth Voir tableau 1 pour plus de renseignements	
BT2		
BT3		
BT4		
BT5		
I1	Bornes I ² C Voir tableau 1 pour plus de renseignements	Ports de communication I ² C
I2		
SPI1	Bornes SPI Voir tableau 1 pour plus de renseignements	Ports de communication SPI
SPI2		
SPI3		
SPI4		
DATA	Bornes ICSP Voir tableau 1 pour plus de renseignements	Programmation du module en ICSP via le Pickit de <i>Microchip</i>
CLOCK		
MCLR		
S1	Saut pour le débogage ICSP	Ces deux connecteurs doivent être soudés ensemble en fonctionnement normal et dessoudés lors d'un débogage
S2		

Tableau 3 : Récapitulatif des brochages du circuit imprimé



4.4.2.4 Circuit imprimé conçu

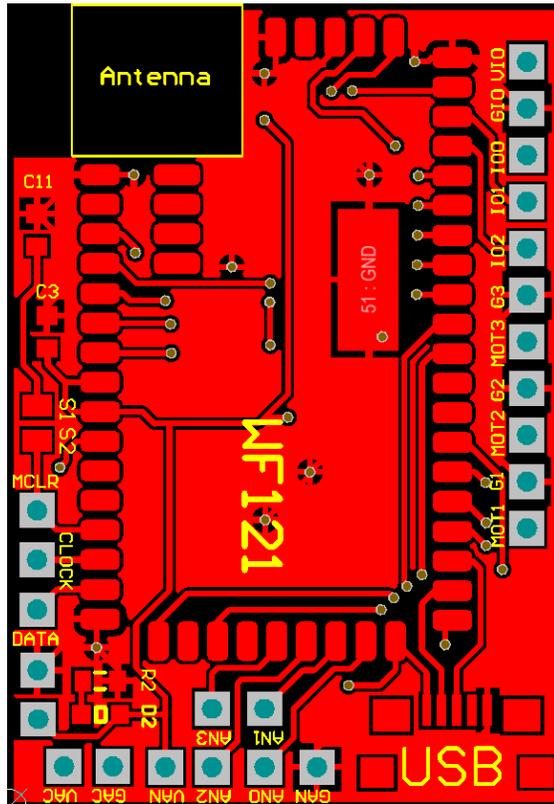


Figure 12 : Couche supérieure

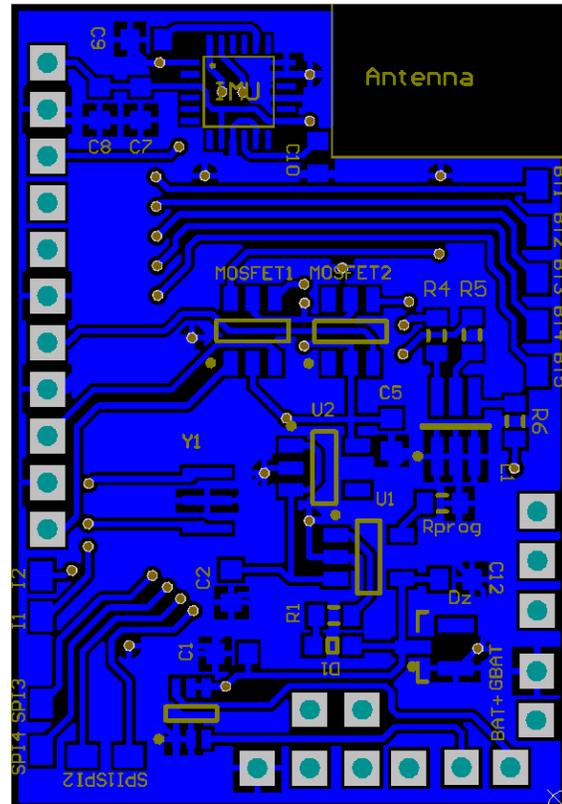


Figure 13 : Couche inférieure

La Figure 12 présente la couche supérieure du circuit imprimé. Elle supporte le module WF121, le connecteur USB et le saut pour la connexion du condensateur en fonctionnement normal. La Figure 13 présente quant à elle la face inférieure du circuit qui comporte le système de puissance et d'alimentation, les connecteurs de batterie et des ports de communication SPI, I²C et Bluetooth. Elle porte aussi la centrale inertielle symbolisée par « IMU ».

4.4.2.5 Prix des composants

Le Tableau 4 liste les composants et leur prix. Les prix sont exprimés en dollars canadien, hors taxe et correspondent à la valeur des composants commandés chez Digikey sur leur site internet à la date du 28 novembre 2015. Seul le MPU-9150 (la centrale inertielle) correspond à une estimation car non en stock sur Digikey.

Tous ces composants sont comptés hors taxe. Il faut y ajouter l'étain pour la soudure et la valeur du circuit imprimé lui-même (qui vaut 0.62\$CAD si réalisé par l'université). Le prix total de ce module est donc inférieur à 100 \$CAD, seuil maximum fixé pour le projet.

Composant	Description	Prix	Quantité	Prix
-----------	-------------	------	----------	------



		unitaire		total
WF121	Module intégrant microcontrôleur et antenne WiFi	41.18	1	41.18
MCP73831T-2ACI	Gestionnaire de charge par régulation de courant de batteries LiPo à une cellule de 4.2V.	0.80	1	0.8
CAT6219-330	Régulateur de tension 3.3V	1.32	1	1.32
BAT54BRW	Composant de 4 diodes utilisé pour faire un pont de diode	.65	1	.65
MPU-9150	Centrale inertielle	15	1	15
DMG9926	Double MOSFET N à drain commun	.72	2	1.44
BZX84C5V1	Diode zener de 5,1V	.23	1	.23
A118106CT	Connecteur micro USB	4.07	1	4.07
CSTCE8M00G55A	Résonateur de 8.00MHz avec condensateur intégrés	.75	1	.75
SMLP36RGB2W3	LED tricolor rouge-vert-bleu	1.65	1	1.65
SML-D1	LED monochrome	.49	2	.98
Résistances	Pour ces composants, voir les valeurs dans les schémas. Une valeur moyenne a été prise car il est possible de faire de grosses réductions de prix sur Digikey.	.02	6	.12
Condensateurs		.1	11	1.2
Prix total hors taxe :				69.39

Tableau 4 : Prix des composants

4.4.3 Directives à respecter

Les principales directives concernant ce circuit concernent la programmation via le Pickit de Microchip.

Afin de programmer le microcontrôleur via le Pickit, il faut s'assurer que les bornes S1 et S2 près des connecteurs ICSP soient dessoudées pour déconnecter le condensateur C11 lorsqu'on utilise le Pickit. Sinon, il faut souder ces deux broches ensemble pour connecter le condensateur.

4.5 Conclusion

La conception de ce circuit imprimé n'a pas posé de difficulté particulière mis à part sur deux points : l'utilisation ou non de résistance sur le port MCRL et l'utilisation d'un cristal externe pour l'horloge du microprocesseur intégré. Sur le premier point, la question a été discutée avec un technicien et après relecture de la datasheet, il est apparu que les résistances sont en interne, dans le module WF121. Quant au second point, il est expressément indiqué que des cristaux ne sont nécessaires que dans deux situations : la nécessité d'une plus grande précision pour l'horloge gérant la veille des appareils (le Wi-Fi et le processeur) et si le port USB est utilisé ou que le débit du port UART est supérieur à 115 kbps. Comme nous ne sommes dans aucune de ces situations, aucun cristal n'a été utilisé. Ce choix a été confirmé par le document cité précédemment car il répète ces directives.

Le circuit a été long à réaliser parce qu'il fallait créer le bloc schématique et l'empreinte de chaque composant à partir des datasheets de chacun. Mais grâce aux documents de la version actuelle du circuit utilisée par le LAIMI, le



temps de la recherche des capteurs a été sauvegardé puisque nous avons pu connaître les capteurs utilisés par le laboratoire.

5.0 Basculement du mode « troisième main » au mode « évitement »

Actuellement sont développés pour le milieu industriel des robots programmés pour l'évitement de collision avec l'opérateur dans le cadre d'espaces de travail hybride. Rappelons que le but de ce genre d'espace de travail et de faire collaborer pour une même tâche un robot et un opérateur. Un autre axe de développement de robots est suivi dans le même but : les robots « troisième main ». Ce type de robot, abondant dans les entreprises automobiles par exemple, sont très pratiques voire essentiels pour l'assemblage de différentes pièces ou parties ensemble. Alors que les premiers priorisent l'évitement et les seconds l'obéissance, le basculement d'un mode à l'autre peut se révéler pertinent. Ainsi, l'opérateur n'aurait pas à se soucier des manœuvres du robot quand celui-ci doit amener la pièce en mode « évitement de collision », puisque le robot éviterait à tout prix son opérateur, et l'on n'utiliserait le robot en troisième main que lorsque c'est nécessaire, au moment de l'assemblage. En plus, le mode troisième main permettrait à l'opérateur de placer le robot dans une autre position, plus ergonomique pour lui que la position actuelle du robot ou encore d'effectuer des tâches non programmées pour le robot que celui-ci pourrait apprendre et effectuer plus tard.

Afin de pouvoir basculer d'un mode de fonctionnement à l'autre (« troisième main » et « évitement de collision »), l'opérateur devrait effectuer une manœuvre détectable par le robot. Actuellement, le LAIMI possède des capteurs retournant des informations depuis la tête grâce au casque de sécurité, depuis les mains grâce à des gants ou depuis les pieds grâce aux semelles.

Ainsi, nous comptabilisons trois sources de commandes dont nous allons dans une première partie explorer les possibilités. Dans une seconde partie, nous développerons un dictionnaire des différents mouvements réalisables et mesurable et enfin, nous verrons l'étude réalisée sur le mouvement de l'écrasement de cigarette.

5.1 Trois sources de commande

Dans cette partie seront établis les avantages et les inconvénients de chacune des sources de commande que sont la tête, les mains et les pieds.

5.1.1 La tête

Il serait logique de penser que la tête pourrait être le centre névralgique des commandes exécutées par l'opérateur à l'intention du robot. En effet, cette partie du corps est celle comportant le plus de capteurs (électroencéphalogramme, accéléromètres, gyroscope, magnétomètre pour les principaux) et il est naturel d'utiliser la tête pour communiquer, ne serait-ce que par la voix ou par ses mouvements. Cependant, cette source représente aussi le désavantage d'être déjà très utilisée pour la communication entre êtres humains et les mêmes signes n'ont pas la même signification entre différentes cultures. Par exemple, pour faire « oui » de la tête, on hoche celle-ci de haut en bas en Europe et en Amérique, mais en Inde, on la bascule de gauche à droite. Un autre désavantage, et non des moindres, est qu'il ne faut pas que l'opérateur bouge la tête pendant des heures afin de ne pas envoyer au robot de fausses commandes qui pourraient le mettre lui ou ses collègues en danger. Une telle pratique ferait apparaître chez les



utilisateurs des crampes musculaires ou des maux et traumatismes que nous souhaitons éviter. L'utilisation de la voix peut paraître pertinent mais ce procédé nécessite un environnement calme et que la voix de l'utilisateur soit constante, or la voix peut se fatiguer et se modifier au point qu'il soit alors impossible pour le robot de la reconnaître.

5.1.2 Les mains

Les mains sont un des organes les plus utilisés consciemment dans la vie de tous les jours et elles peuvent être utilisées pour communiquer lorsque la voix ne suffit pas. Cependant, puisque il est prévu d'utiliser ce mode d'interaction à un poste de travail hybride, leur utilisation peut se révéler peu pertinente en tant que vecteur de commande pour le robot. En effet, puisque les mains servent dans ce cas à la manipulation d'objets, d'outils ou de composants, il se peut qu'un geste programmé corresponde à une position nécessaire à la réalisation d'une tâche, entraînant ainsi une réponse du robot qui ne correspond pas à celle souhaitée par l'opérateur. De plus, communiquer avec les mains impose à l'utilisateur de laisser en suspens la tâche qu'il est en train d'accomplir et cela ne peut pas être toujours possible. Dans le cas qui nous concerne, l'utilisation des mains comme point de repère semble plus pertinent que comme vecteur de commande.

5.1.3 Les pieds

Il ne reste ainsi plus que les pieds et les jambes pour permettre à l'opérateur de commander le robot avec lequel il interagit. En regardant de plus près les littératures et les technologies déjà existantes, rares sont celles mettant en œuvre les pieds pour commander même si depuis quelques années, des recherches dans cette voie ont été menées (Menelas & Otis, 2013). Le principal inconvénient à l'utilisation des pieds pour la commande vient du fait que ce n'est pas un organe souple comme la tête ou les mains et qu'ils nous servent à nous tenir debout. De plus, commander avec le pied et souvent associé avec une posture où l'un des pieds est en l'air alors que l'utilisateur tient en équilibre sur l'autre pied, ce que certaines personnes sont incapables de faire à cause de problèmes d'équilibre par exemple. Mais nous avons développé une semelle constituée de quatre capteurs de force disposés de telle sorte qu'il est possible d'associer à une combinaison de ces capteurs des ordres simples à faire exécuter par le robot. De plus, l'utilisation des pieds comme vecteur de commande permet de laisser libres la tête et les mains pour une utilisation tout à fait normale pour l'utilisateur, et ainsi rendre plus intuitive la collaboration entre l'homme et la machine.

C'est donc ce dernier vecteur que nous pensons utiliser et que nous présenterons dans la suite de ce document.

5.2 Les dictionnaires de mouvements détectables et utilisables

Avec la semelle viennent différentes possibilités de détection de mouvements des pieds de l'opérateur. Rappelons donc brièvement ce qui constitue les capteurs et les actionneurs utilisables dans la semelle développée :

- L'accéléromètre qui permet la détection de tout changement de position du pied de l'opérateur
- Les quatre capteurs de force disposés aux quatre « coins » de la semelle : à gauche et à droite du talon et de part et d'autre du premier métatarsien, à la base des phalanges des orteils



Ainsi, à partir de ces composants, il est possible de définir un dictionnaire de mouvements réalisables par l'opérateur. C'est ce que nous verrons dans cette partie.

5.2.1 L'accéléromètre

L'accéléromètre nous donne la possibilité d'observer et d'utiliser les mouvements du pied pour établir des commandes pour le robot. Cette détection est possible car le dispositif ne se trouve pas au niveau de la cheville et donc nous pouvons mesurer les rotations en plus des translations. Nous assumons ici que les translations ne sont pas de vraies translations mais des rotations dont le centre est suffisamment loin pour permettre de faire cette approximation. Cela se justifie par le fait que chez le sujet debout, les rotations ont leur centre au niveau de la hanche et que lorsqu'il est assis, ce centre se trouve au niveau du genou. Le Tableau 5 liste les différents mouvements détectables par l'accéléromètre qui sont des mouvements facilement réalisables le pied en l'air alors que le Tableau 6 montre des mouvements plus facilement réalisables avec le pied au sol.

Mouvements de rotation du pied ayant pour centre la cheville			Mouvements simples de translation du pied (ne nécessitant qu'une intervention minimale de la jambe)		
Adduction (vers l'intérieur) / Abduction (vers l'extérieur)		Mouvement de rotation dans le plan (xOy) du pied ayant pour centre de rotation le talon	Translation de gauche à droite		Mouvement de translation selon l'axe x
Dorsiflexion (vers le bas) / Extension (vers le haut)		Mouvement de rotation dans le plan (yOz) du pied ayant pour centre de rotation la cheville	Translation d'avant en arrière		Mouvement de translation selon l'axe y
Varus (vers l'intérieur) / Valgus (vers l'extérieur)		Mouvement de rotation dans le plan (xOz) du pied ayant pour centre de rotation la cheville	Translation de haut en bas		Mouvement de translation de haut en bas

Tableau 5 : Mouvements simples du pied détectables par l'accéléromètre

Mouvements complexes de rotation du pied ayant pour centre de rotation l'articulation de la première phalange des orteils



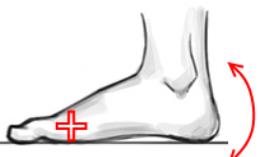
Rotation de gauche à droite		Mouvement de rotation dans le plan (xOy) du pied ayant pour centre de rotation l'articulation de la première phalange des orteils
Rotation de haut en bas		Mouvement de rotation dans le plan (yOz) du pied ayant pour centre de rotation l'articulation de la première phalange des orteils

Tableau 6 : Dictionnaire de mouvements de rotation du pied ayant pour centre la première articulation des orteils

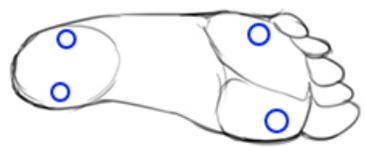
5.2.2 Les capteurs de force

Les quatre capteurs de force disposés dans la semelle permettent quant à eux de détecter la position du pied par rapport au sol. En effet, comme le montre la Figure 14, les capteurs sont disposés aux quatre coins de la semelle et nous permettent donc de savoir quelle est l'orientation du pied. La contrainte d'utilisation de ces capteurs est que le pied doit être en contact avec le sol pour qu'ils puissent être utilisés. Autrement, la seule information fournie est : « le pied est en l'air ».



Figure 14 : Position des capteurs de force par rapport au pied

Le Tableau 7 liste les combinaisons pouvant être renvoyées par les capteurs. Ne sont traitées ici que les combinaisons jugées réalistes et demandant peu d'effort d'apprentissage à l'opérateur.

Signaux envoyés par les capteurs (cercle bleu : inactif, rond bleu : actif)	Signification du signal
	Les quatre capteurs sont inactifs donc le pied est en l'air.



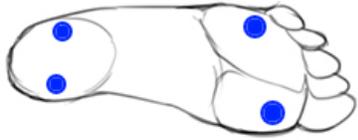
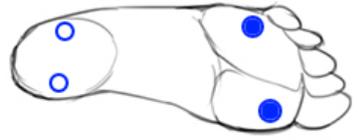
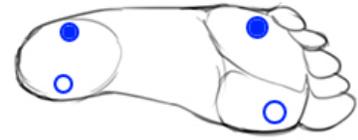
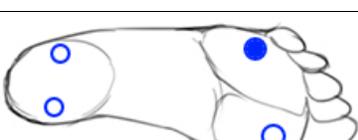
	Les quatre capteurs sont actifs donc le pied est au sol à plat.
	Les capteurs de devant sont actifs donc le pied est incliné vers l'avant
	Les capteurs de l'extérieur sont actifs donc le pied est incliné vers l'extérieur
	Les capteurs de derrière sont actifs donc le pied est incliné vers l'arrière
	Les capteurs de l'intérieur sont actifs donc le pied est incliné vers l'intérieur
	Le capteur extérieur avant est actif donc le pied est incliné vers l'avant sur le côté extérieur
	Le capteur intérieur avant est actif donc le pied est incliné vers l'avant sur le côté intérieur

Tableau 7 : Dictionnaire des signaux envoyés par les capteurs de force dans la semelle

À partir de ces différents dictionnaires, il est possible d'utiliser directement ces signaux de base ou de réaliser des combinaisons entre les signaux de l'accéléromètre et des capteurs.



5.3 Utilisation des dictionnaires

À partir des dictionnaires définis précédemment, il est possible d'utiliser ces mouvements pour donner des ordres au robot. C'est ce qui a été fait lors de la conception d'une version précédente de la semelle (Menelas & Otis, 2013) pour le cas de l'accéléromètre. Un autre dictionnaire a été réalisé par M. Ping Li (Ping, Safety analysis using a smart safety helmet embedded with IMU and EEG sensors applied in industrial facility, 2015) pour la détection des mouvements de la tête.

Dans le présent document, nous allons nous intéresser à un mouvement permettant à l'opérateur de faire basculer le robot entre les modes « évitement de collision » et « troisième main ». Le mouvement qui sera considéré ici est celui de l'écrasement de cigarette. Ce mouvement est composé d'un basculement vers l'avant du pied et d'une série de deux ou trois rotations puis le pied est à nouveau posé à plat au sol, comme illustré par la Figure 15.

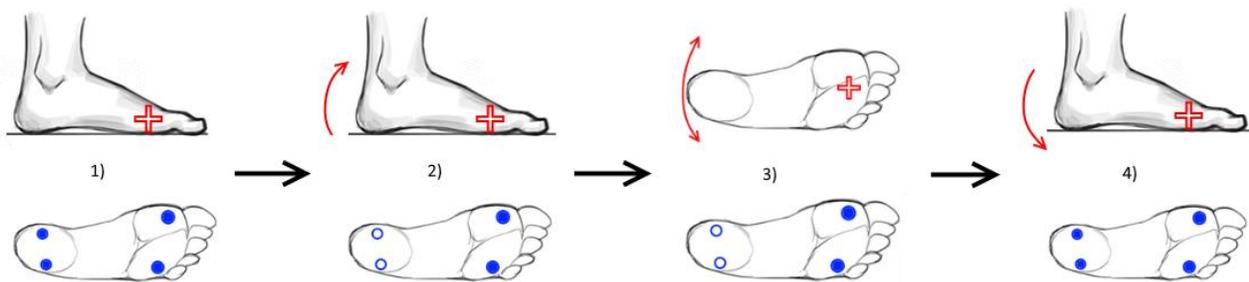


Figure 15 : Schéma illustrant le mouvement permettant à l'opérateur de basculer le mode du robot entre "évitement de collision" et "troisième main"

Ce mouvement possède plusieurs avantages. Le premier est qu'il est connu de tout le monde même s'il tend à disparaître du comportement des fumeurs puisqu'ils utilisent des cendriers et qu'il sera nécessaire aux non-fumeurs d'en faire l'apprentissage facilement. Le deuxième est qu'il demande peu d'efforts, il n'est pas nécessaire d'être champion olympique de gymnastique pour pouvoir l'effectuer. Le troisième est que l'opérateur garde le pied au sol, de ce fait, même les personnes victimes de troubles de l'équilibre peuvent faire ce mouvement en toute sécurité. Enfin, il s'agit d'un mouvement qui ne peut pas être accidentel, en effet, il ne peut pas être confondu avec une montée ou descente d'escalier par exemple et nécessite une volonté d'action de l'opérateur pour être effectué. L'inconvénient qui peut apparaître lors de l'utilisation de ce mouvement est que l'opérateur ne réussisse pas à maintenir le contact avec les deux capteurs du bout du pied, il s'agit donc d'un problème mais qui peut être corrigé facilement en demandant à l'opérateur de faire son mouvement au mieux ou plus difficilement en tenant compte dans la programmation de la détection de ce mouvement.



5.4 Les premières mesures

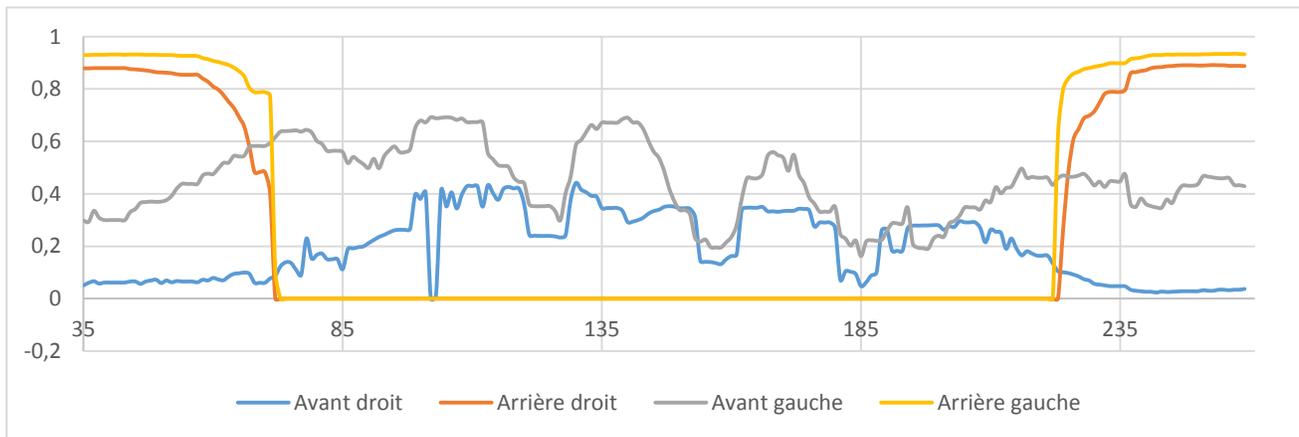


Figure 16 : Courbes de relevé des Capteurs de force lors du test du mouvement d'écrasement de cigarette en position assise

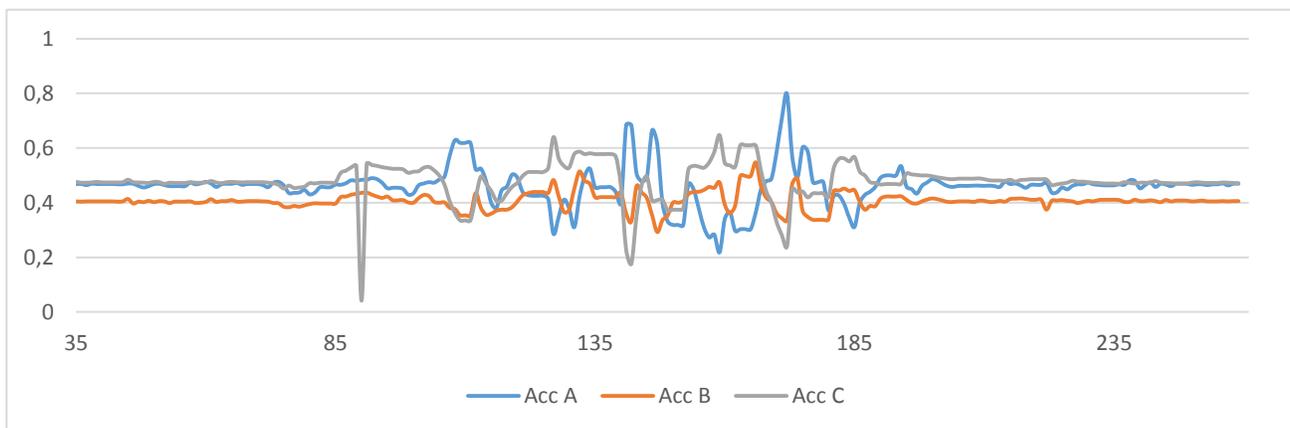


Figure 17 : Courbes de relevé de l'accéléromètre lors du test du mouvement d'écrasement de cigarette en position assise

Les mesures présentées ici correspondent à un mouvement d'écrasement de cigarette effectué par moi-même chaussé au pied droit avec la semelle développée par le LAIMI en position assise. Douze autres tests ont été réalisés à différentes vitesses et nombres de rotations et le relevé est présenté en 8.1 mais un seul mouvement a été représenté ici. La Figure 16 montre le comportement des capteurs de force alors que la Figure 17 montre celui de l'accéléromètre.

Il est possible de constater sur ces graphiques le même comportement que celui prévu : les deux capteurs du talon n'envoient plus de signal et l'oscillation de la cheville est détectée par l'accéléromètre.

Il est aussi possible de voir les mêmes oscillations sur les capteurs de l'avant du pied, qui correspondent en fait à un relâchement de la pression exercée sur le sol pendant les mouvements de rotation. Il pourrait donc être tentant d'utiliser ces variations pour détecter les mouvements de rotation mais il est préférable de regarder le comportement de



l'accéléromètre. En effet, il s'agit ici de ma propre façon d'écraser une cigarette et il est parfaitement logique de supposer qu'une autre personne pourra mettre plus ou moins de force que moi sur les capteurs. De plus, ces mesures ont été réalisées alors que j'étais assis.

L'allure des courbes en position debout, présentées aux Figure 18 et Figure 19, est différente pour le capteur de force avant gauche notamment. En effet, en position assise, seule la force employée par l'utilisateur est détectée par les capteurs alors qu'en position debout, la force correspond au poids de la personne. Elle est donc plus importante en amplitude et est plus régulière car l'utilisateur s'appuie sur cette partie du pied pour maintenir son équilibre.

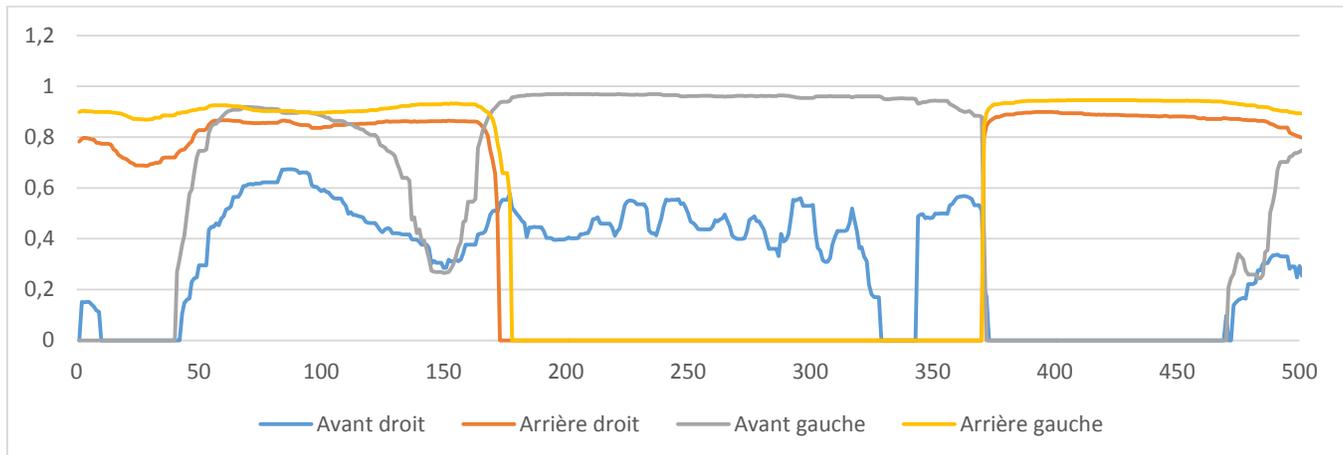


Figure 18 : Courbes de relevé des capteurs de forces lors du test du mouvement d'écrasement de cigarette en position debout

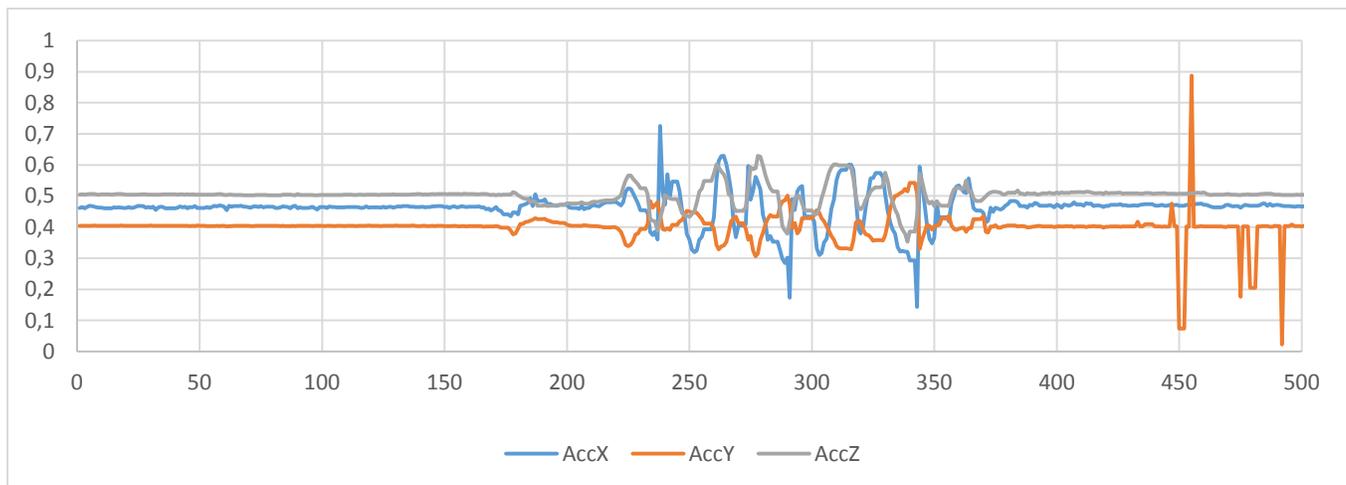


Figure 19 : Courbes de relevé de l'accéléromètre lors du test du mouvement d'écrasement de cigarette en position debout

Les variations visibles dans les relevés des capteurs de force avant le point 170 correspondent à une reprise d'équilibre sur le pied. C'est pour cela que les capteurs de force enregistrent une variation mais pas l'accéléromètre.



Quant à l'accéléromètre, on peut constater que les courbes sont plus irrégulières que prévu. Il est possible de remarquer que tous les axes répondent à la variation de l'accélération alors qu'on pourrait s'attendre à ce qu'un seul de ces axes montre une variation. Cela est dû probablement au fait que le capteur n'est pas parfaitement aligné avec le sol mais aussi à une variation de la hauteur de la cheville par rapport au sol pendant le mouvement. J'ai pu avoir un mouvement contenu dans un plan oblique plutôt que parallèle au sol. Mais les données restent tout de même exploitables pour la détection du mouvement.

5.5 La détection du mouvement

Elle s'effectue après avoir filtré les mesures de capteurs. Chaque capteur nécessite son propre type de filtre : les signaux des capteurs de forces sont filtrés par un filtre passe-bas afin de ne conserver que la composante continue du signal alors que le signal de l'accéléromètre est filtré par un passe-bande afin d'éliminer les bruits et la composante continue. Les caractéristiques de chacun des filtres seront présentées dans un premier point. Une fois filtrés, plusieurs tests sont effectués à partir de ces signaux, c'est ce que nous développerons en second point dans cette partie.

5.5.1 Les filtres

Les fréquences de coupure des filtres utilisés sont situées dans le domaine des basses fréquences. En effet, les mouvements à détecter, comme la marche ou l'écrasement, sont de basse fréquence parce que c'est le domaine de fréquence de l'homme. Pour la marche, la fréquence est de 1 voire 2 Hz alors que pour le mouvement d'écrasement, elle est de 3 à 5 Hz en fonction de la vigueur mise dans ce mouvement. Mais en aucun cas la fréquence des signaux que nous cherchons à analyser ne dépasserait les 10Hz. La fréquence d'échantillonnage des capteurs est de 100 Hz, largement suffisante pour notre application.

Donc, le filtre passe-bas a une fréquence de coupure de 3 Hz et les fréquences de coupure du filtre passe-bande sont de 1 et 5 Hz. Les relevés filtrés sont montrés Figure 20 et Figure 21.

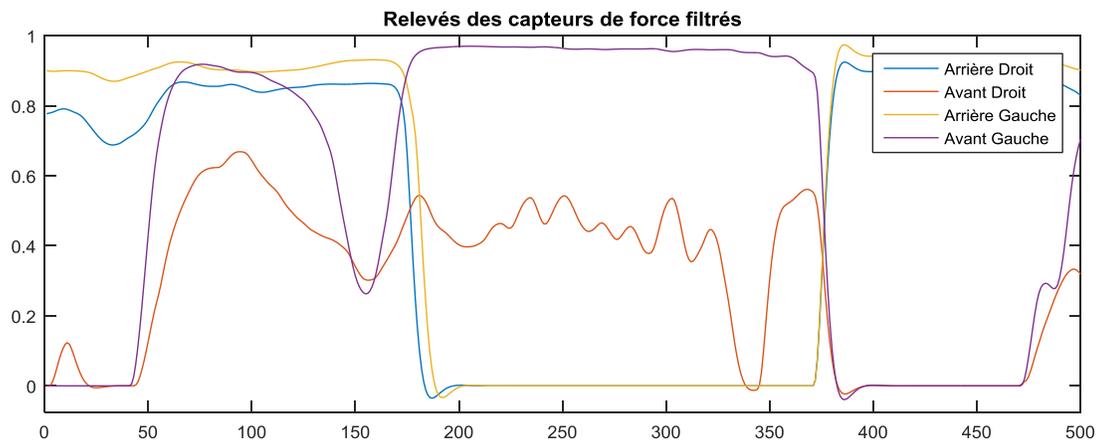


Figure 20 : Relevés des capteurs de force filtrés

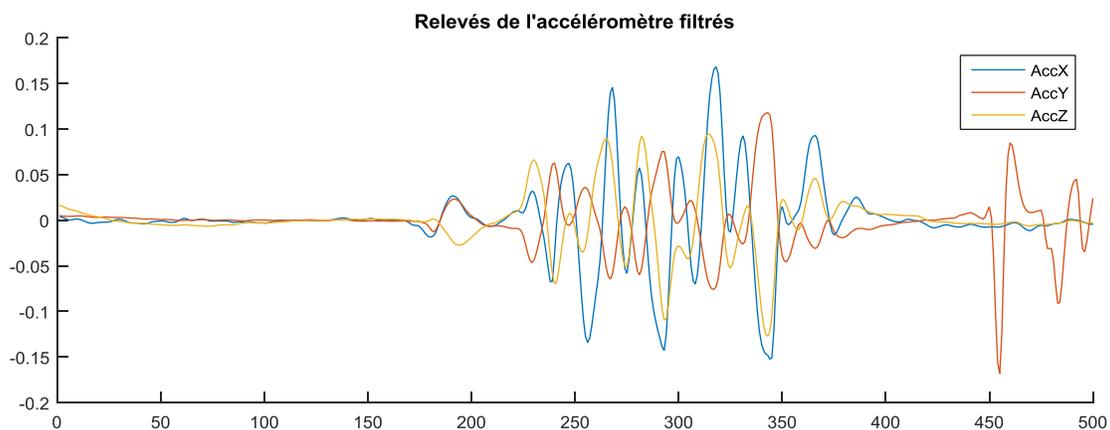


Figure 21 : Relevés de l'accéléromètre filtrés

5.5.2 Les critères d'évaluations de la reconnaissance

La détection du mouvement se compose de plusieurs tests : l'état des capteurs de force, l'analyse fréquentielle des capteurs et de la norme de l'accéléromètre.

Afin de correspondre au mouvement, les tests à vérifier sont les suivants :

- Tests booléens sur la valeur de la moyenne sur 180 points consécutifs des capteurs de force avant gauche, arrière droit et arrière gauche. Le capteur avant droit n'a pas été pris en compte car les différences entre la marche et le mouvement d'écrasement ne sont pas assez significatifs pour être utilisés. La moyenne a été utilisée parce que l'utilisation des valeurs filtrées donne un résultat positif pour les mouvements de la marche, il s'agit donc d'un faux-positif. La longueur de 180 points correspond à la longueur du modèle utilisé pour l'accéléromètre et il s'agit du modèle le plus long, cette valeur est donc valeur de référence.



- Tests booléens sur le résultat de la corrélation croisée entre le modèle et le calcul depuis les relevés de la norme de l'accéléromètre. Ce modèle est inspiré de la modulation d'amplitude. Il s'agit donc d'un produit entre un sinus et un cosinus selon l'Équation 1 :

$$\text{Équation 1 : modèle} = -.05 * \left(\sin \left(2 * \pi * \frac{t}{1.5} \right) + 1 \right) * \left(\cos \left(2 * \pi * \frac{t}{18} \right) + 1 \right)$$

- Tests booléens sur le résultat de la corrélation croisée entre les FFT (*Fast Fourier Transform*) du modèle et du calcul depuis les relevés de la norme de l'accéléromètre.

Afin de tester la robustesse du programme développé, une série de mesure a été réalisée. Celle-ci est constituée successivement de cinq pas de marche, un mouvement d'écrasement de cigarette, trois autres pas, un nouveau mouvement d'écrasement de cigarette, cinq pas, un mouvement d'écrasement avec le talon au sol et l'avant du pied soulevé, trois pas, un mouvement d'écrasement de cigarette et pour finir un pas. Le relevé des capteurs est présenté à la Figure 22.

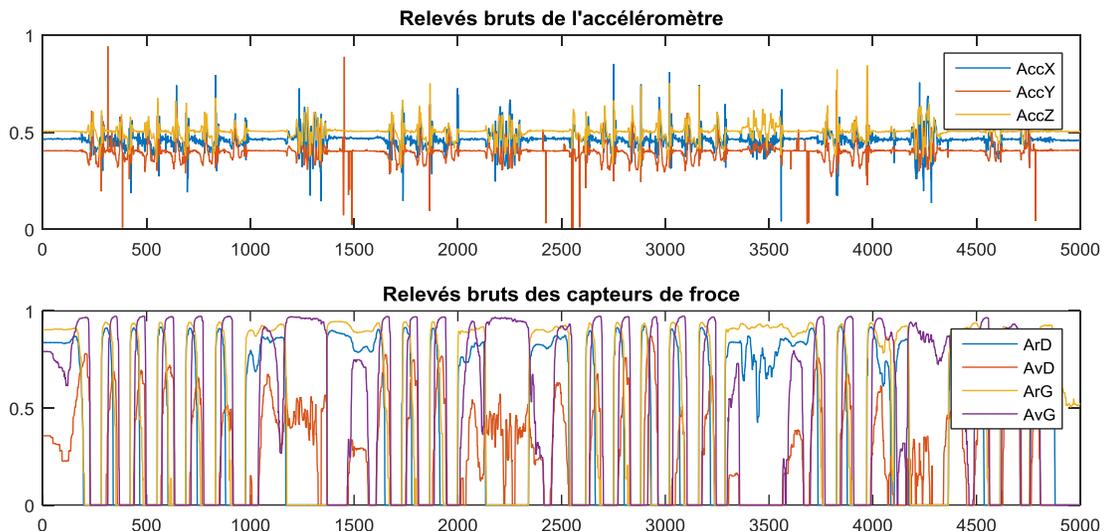


Figure 22 : Relevés bruts des capteurs de la semelle lors d'une simulation de trois mouvements différents : la marche, l'écrasement avec l'avant du pied et l'écrasement avec l'arrière du pied.

À partir de ces relevés, il est possible de partitionner l'enregistrement en intervalles contenant chacun un type de mouvement :

- Marche : [200;950], [1600;2075], [2500;3325], [3670;4030] et [4407;4880]
- Écrasement arrière : [3325;3600]
- Écrasement avant : [1100;1400], [2075;2400] et [4150;4400]

Une fois ces signaux filtrés et la norme de l'accélération calculée, nous obtenons les relevés présentés par la figure.

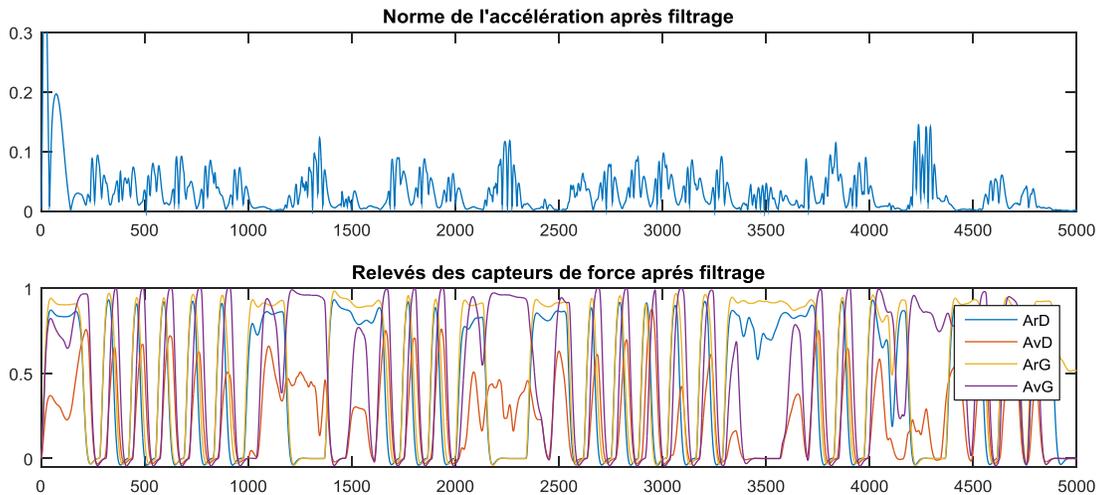


Figure 23 : Norme de l'accélération et capteurs de force après filtrage

Ensuite, afin de simuler le comportement du système en développement, le programme parcourt ces relevés points par points en analysant une fenêtre de 180 points (1.8 s) avec les tests expliqués précédemment. À chaque analyse de la corrélation croisée, le maximum est enregistré et c'est sur ce résultat que nous effectuons le test booléen. La Figure 24 montre les résultats utilisés pour établir les seuils de ces tests.

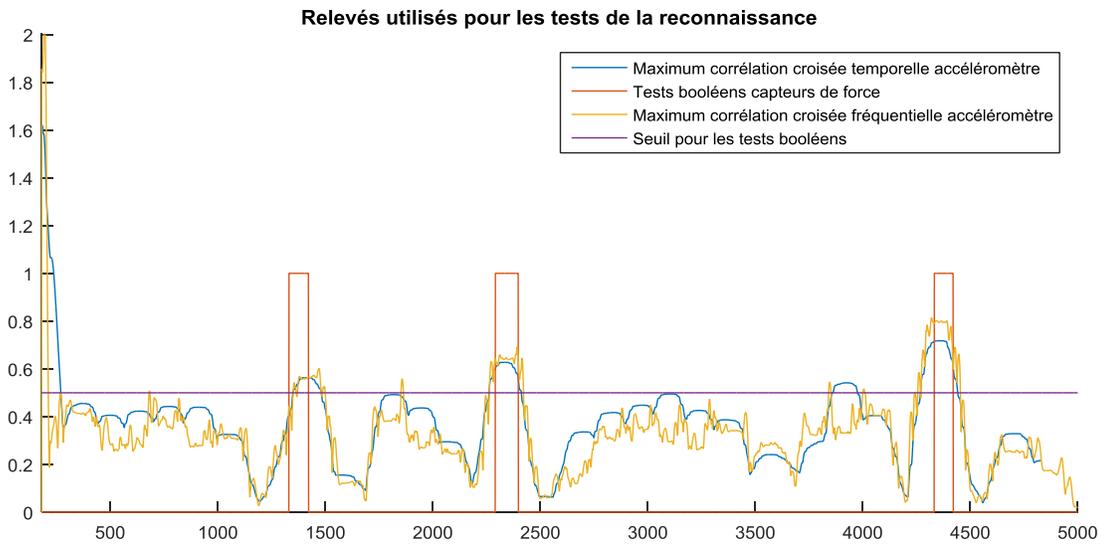


Figure 24 : Résultats utilisés pour les tests de reconnaissance

À partir de ce dernier relevé, il est aisé de voir qu'avec un seuil de 0.5 environ les tests booléens donneront une valeur vraie pour la comparaison entre les maxima des corrélations et ce seuil.

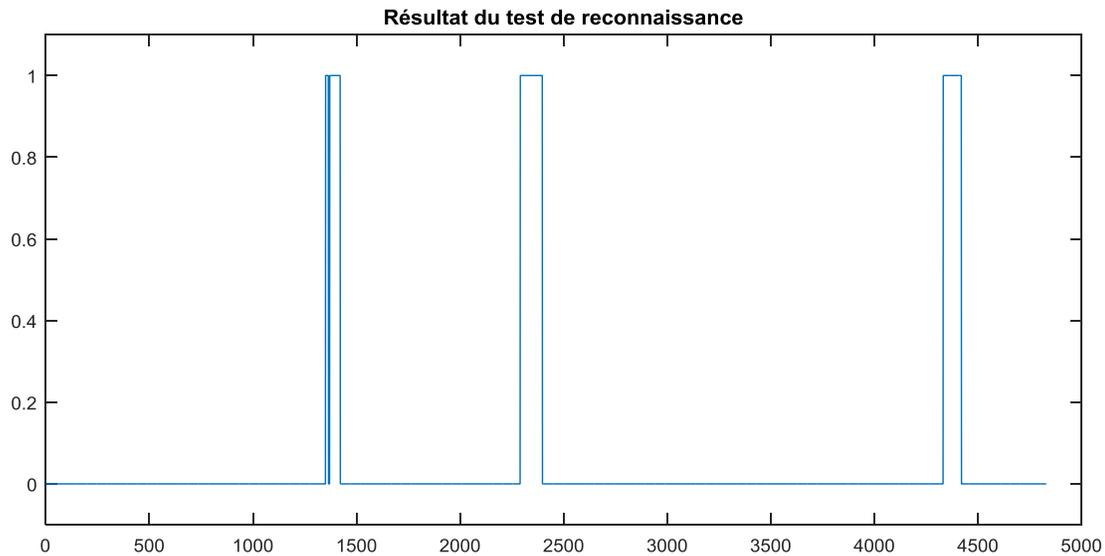


Figure 25 : Résultat du test de reconnaissance du mouvement d'écrasement de cigarette

Après avoir réalisé les tests, les intervalles de reconnaissances sont les suivants : [1350;1365], [1370;1422], [2290;2397] et [4332;4420]. Nous pouvons constater un faux négatif sur l'intervalle [1365;1370]. En réexaminant les relevés servant à réaliser ce test, il est possible de constater qu'il s'agit d'une irrégularité dans la courbe du relevé des maxima de la corrélation croisée de l'analyse fréquentielle de la norme de l'accéléromètre. En descendant ce seuil à une valeur de 0.48, il n'y a plus de faux négatif.

5.6 Conclusion

Après avoir étudié les différentes sources de commandes disponibles pour l'opérateur sans que cela ne soit une gêne dans son travail, nous en sommes arrivés à la conclusion que le pied et ces mouvements sont disponibles et assez variés pour satisfaire les besoins de notre application. Ensuite, des mesures ont été effectuées afin de voir les caractéristiques utiles pour notre application d'un mouvement particulier, celui de l'écrasement de cigarette, et des modèles ont été développés pour détecter ce mouvement. Les tests sur le maximum de la corrélation croisée ont été choisis pour la plupart des analyses car c'est une valeur facile à obtenir et à programmer. En effet, il s'agit ici d'une simulation qui a pour but de nous aider dans le développement d'un programme qui effectuera lui-même les tests et détectera de lui-même le mouvement.

6.0 Le pilotage du robot en mode « troisième main »

Dans cette partie, nous développerons l'étude de la communication d'un ordre à un robot depuis la semelle instrumentée. Cependant, afin de pouvoir tester cette solution sur le robot ERICC, une application Android a été développée et c'est un module IOIO qui commande le robot, comme illustré Figure 26. Nous décrivons d'abord les choix effectués pour le pilotage du robot, ensuite l'application Android développée, les résultats obtenus et enfin les recommandations et améliorations applicables à cette partie du projet.

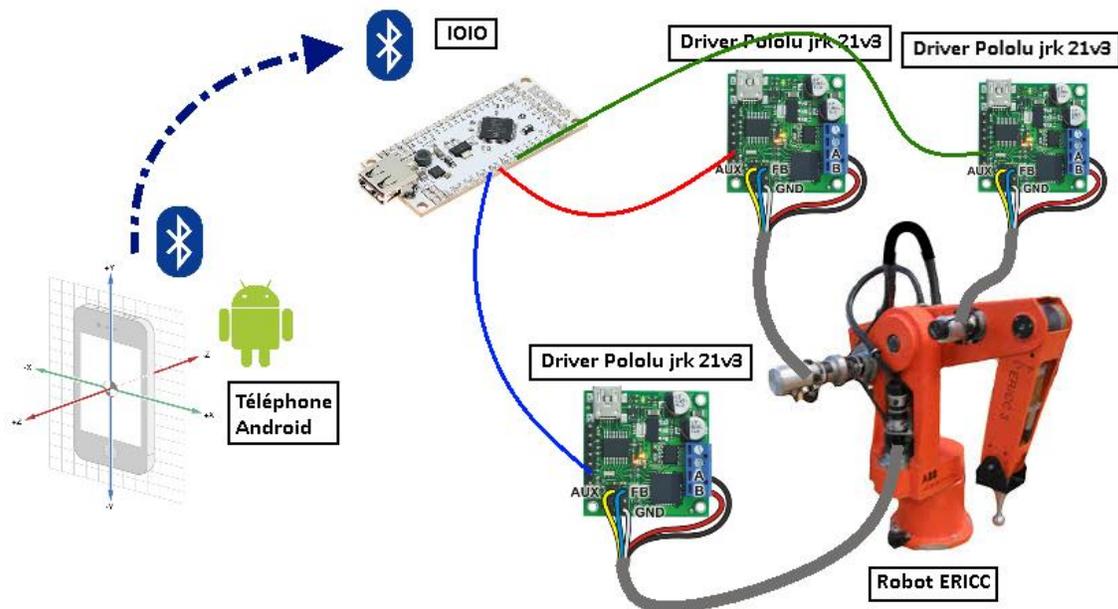


Figure 26 : Pilotage du bras robotisé ERICC grâce à l'accéléromètre d'un téléphone Android

6.1 Le pilotage du robot

Dans cette partie seront développés le type de la commande du robot, le type de signal utilisé pour le commander et la commande réalisée.

6.1.1 Le type de commande

À l'heure actuelle, le robot ERICC est composé de quatre moteurs à courant continu commandant les articulations de lacet, de l'épaule, du coude et du poignet. Chacun de ces moteurs est commandé par un pilote jrk21v3 du constructeur Pololu. L'avantage de ce pilote est qu'il permet de commander les moteurs à courant continu grâce à différents types de signaux, que ce soit analogique, TTL (numérique) ou PWM. Les étudiants travaillant précédemment sur ce robot ont choisi de générer un signal PWM pour le pilotage des moteurs du robot. Dans le cadre de ce projet, il s'est avéré être plus simple de travailler avec un signal analogique.

6.1.2 Le signal utilisé

Dans les faits, ce n'est pas un véritable signal analogique dans la mesure où le signal généré par le module IOIO est un signal de type PWM dont nous faisons varier le rapport cyclique, il est donc reçu par le pilote comme un signal analogique. Cette solution a été choisie car elle représente l'avantage d'être simple à mettre en œuvre sur le module IOIO. En effet, il est possible de modifier directement le rapport cyclique du signal carré périodique produit par le module alors que la génération d'un signal PWM à proprement parler aurait exigé la création de ce signal par le programmeur et de traiter directement avec l'horloge du microcontrôleur, ce qui est nettement plus difficile.

En modifiant le rapport cyclique, nous obtenons ainsi un signal de faible amplitude lorsque le rapport est proche de 0% et un signal de forte amplitude lorsque le rapport est proche de 100%. Le réglage des paramètres dans le pilote



des modules Pololu permet d'utiliser toute la plage de l'amplitude du signal pour toute la plage du mouvement voulu pour le robot.

6.1.3 La commande du robot

Afin de générer le signal de la commande du robot, nous utilisons une règle de proportionnalité entre l'angle d'inclinaison du téléphone, sur lequel est installé l'application, et le rapport cyclique. Le repère tridimensionnel des téléphones Android ainsi que l'amplitude de la rotation selon chaque axe sont normés ; ainsi, il est facile d'utiliser une simple règle de proportionnalité entre l'inclinaison du téléphone et le rapport cyclique du signal produit par le module IOIO.

Deux solutions se sont présentées lors de la conception de cette application :

- La première consiste à relier directement le rapport cyclique à l'inclinaison du téléphone. Les avantages de cette solution sont sa simplicité d'implantation et le caractère intuitif de son utilisation : à une inclinaison donnée du téléphone correspond une position. En revanche elle oblige l'opérateur à être très attentif sur sa position car l'inclinaison du téléphone ne doit pas changer lorsqu'on passe d'un axe de travail à l'autre.
- La seconde consiste à additionner un gain proportionnel à l'inclinaison du téléphone à la valeur actuelle du rapport cyclique. L'avantage de cette solution est qu'elle permet d'utiliser le téléphone même avec de petites valeurs d'inclinaison. Cependant, il faut régler ce gain afin que le mouvement ne soit ni trop rapide, ni trop lent.

Dans notre application, la première solution a été utilisée pour effectuer les réglages sur les pilotes des moteurs alors que la seconde correspond à l'utilisation finale car elle permet de piloter le robot avec précision, même avec une très petite variation d'angle.

6.2 L'application Android

L'application développée dans cette partie du projet, dont le code source se trouve en 8.2, a pour rôle de récupérer les valeurs de l'accélération de pesanteur terrestre mesurées par l'accéléromètre du téléphone, de les convertir en angle de rotation, d'utiliser ces angles pour calculer le gain à ajouter (ou retrancher) au rapport cyclique et de faire produire par le module IOIO un signal PWM dont le rapport cyclique aura été calculé. L'équation utilisée est la suivante :

$$DutyCycle = \frac{\theta}{2 * \pi * k} + DutyCycle$$

Où θ est la valeur d'angle fournie de l'accélération par rapport au repère de l'accéléromètre du téléphone en radian, k est un gain à ajuster en fonction de la vitesse de déplacement voulue tel que plus k est grand, plus le système est lent et $DutyCycle$ est le rapport cyclique du signal PWM généré par le IOIO.

Cette application a pour squelette une autre application développée précédemment par les membres du LAIMI est et donc largement améliorable.



6.3 Les résultats obtenus

Pour faire fonctionner le robot à partir du module IOIO, il faut connecter les ports 10, 11 et 12 du module aux bornes RX des pilotes Pololu du moteur de lacet, de l'épaule et du coude respectivement.

L'utilisation de cette application a donc permis de contrôler les moteurs du robot après un nombre important de réglages. Comme nous avons quatre moteurs à commander et seulement deux axes de commande exploitables, il a été décidé de ne piloter que trois des quatre moteurs : lacet, épaule et coude. Un des axes commande le moteur de lacet alors que l'autre commande les moteurs de l'épaule et du coude.

6.4 Recommandations

Il est fortement conseillé d'alimenter le module via une source externe et de placer le robot proche de la position initiale programmée par l'application.

6.5 Travail suggéré

Afin d'améliorer l'application, il est envisageable d'approfondir le travail sur l'interface graphique et de finaliser l'implémentation de la lecture des potentiomètres des moteurs afin de connaître leur position et ainsi terminer l'élaboration de la commande du robot grâce au signal de l'accéléromètre.



7.0 Bibliographie

- Barkallah, E., Otis, M. J.-D., Ngomo, S., & Heraud, M. (2015). Measuring Operator's pain : toward Evaluating Musculoskeletal disorder at work. *Cogent*. (2015, Août 17). *OPC DataHub*. Récupéré sur <http://opcdatahub.com/Home.html>
- Eweron, M., Neumann, G., Lioutikov, R., Ben Amor, H., Peters, J., & Maeda, G. (2015). Modeling Spatio-Temporal Variability in Human-Robot Interaction with Probabilistic Movement Primitives. *Workshop on Machine Learning for Social Robotics, ICRA*.
- Geib, J.-M., Gransart, C., & Merle, P. (1999). *Corba : Des concepts à la pratique* (éd. 2e). Paris: Dunod.
- Maurice, P., Measson, Y., Padois, V., & Bidaud, P. (2012). Assessment of physical exposure to musculoskeletal risks in collaborative robotics using dynamic simulation. *HAL*.
- Menelas, B.-A. J., & Otis, M. J.-D. (2013). Use of the foot in 3Dinteractions with mobile devices.
- Meziane, R. (2015). *Partage des activités de production dans un environnement dynamique en collaboration humain-robot*.
OpenOPC for Python. (2015, Août 17). Récupéré sur <http://openopc.sourceforge.net/>
- Orfali, R., & Harkey, D. (1997). *Client/Server Programming with Java and CORBA*. John Wiley & Sons, inc.
- Peshkin, M., & Colgate, J. E. (1999). Cobots. (M. U. Press, Éd.) *Industrial Robot: An International Journal*, 26(5), pp. 335-341. Consulté le 11 5, 2015
- Ping, L. (2015, août 12). *Safety analysis using a smart safety helmet embedded with IMU and EEG sensors applied in industrial favity*.
- Ping, L., Meziane, R., Otis, M. J.-D., Ezzaidi, H., & Cardou, P. (2014, Octobre 16-18). A smart Safety helmet using IMU ans EEG sensors for worker fatigue detection. *Robotique and Sensors Environments (ROSE), 2014 IEEE International Symposium on*, pp. 55-60. Consulté le août 17, 2015, sur <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6952983>
- Reinhart, G., & Spillner, R. S. (s.d.). Approach of Applying Human-Robot-InteractionTechnologies to Assist Workers with Musculoskeletal Disorders in Production.
- Riwan, A., Giudicelli, B., Taha, F., Lazennec, J.-Y., Sabhani, A., Kilian, P., . . . Lavallée, S. (2011). Projet Surgicobot : robot d'assistance à la chirurgie du rachis. *SciVerse ScienceDirect Journals*, 32(2), pp. 130-134. doi:10.1016/j.irbm.2011.01.006
- Robot Operating System. (2015, Août 17). *Wiki ROS*. Récupéré sur <http://wiki.ros.org/>
- Rovers, A., & van Essen, H. (2006). Guidelines for haptic interpersonal communication applications: an exploration of foot interaction styles. *Virtual Reality*, 177-191.
- Velázquez, R., & Pissaloux, E. E. (2008). Tactile Displays in Human-Machine Interaction: Four Case Studies. *The International Journal of Virtual Reality*, 7(2), pp. 51-58.
- Velázquez, R., Bazan, O., & Varona, J. (2013). Wearable Electronic Tactile Display for the Foot.



8.0 Annexes

8.1 Annexe 1 : Programme Matlab développé pour la détection du mouvement d'écrasement de cigarette :

```
close all;

%% Importation des données
if ~exist('DataTest','var')
    % Mouvement de marche sur plancher plat
    DataTest = dlmread('RawData0_F-1.txt',' ');
    DataTest = [DataTest(:,2:4) DataTest(:,6:9)];
end

%% Filtrage des données

Data = DataTest; % à changer en fonction des données à analyser

% Passe-bande pour les accéléromètres
[b1,a1] = butter(2,[1 5]/100,'bandpass');
% Passe-bas pour les capteurs de force
[b2,a2] = butter(2, 3/100,'low');

% Accéléromètre
DataOut1 = filter(b1,a1,Data(:,1));
DataOut2 = filter(b1,a1,Data(:,2));
DataOut3 = filter(b1,a1,Data(:,3));
% Capteur de force
DataOut4 = filter(b2,a2,Data(:,4));
DataOut5 = filter(b2,a2,Data(:,5));
DataOut6 = filter(b2,a2,Data(:,6));
DataOut7 = filter(b2,a2,Data(:,7));

%% Calcul norme accéléromètre
NormeAcc = sqrt(DataOut1.^2 + DataOut2.^2 + DataOut3.^2);

DataOut = [NormeAcc DataOut4 DataOut5 DataOut6 DataOut7];

%% Traitement de la norme de l'accéléromètre

% Temporel

% Élaboration du modèle
tmax=18;
t=0:.1:tmax;
mod = sin(2*pi*t/1.5)+1;
env = -cos(2*pi*t/18)+1;
modeleNA = .05*env.*mod;

% Calcul de la corrélation croisée entre le modèle et le relevé
temp = [zeros(1,181); modeleNA];
for i=181:length(NormeAcc)-180
    xco = xcorr(NormeAcc(i-180:i),modeleNA);
    temp(1,:) = NormeAcc(i-180:i);
    maxcorna(i) = max(xco);
end
figure(1)
subplot(2,2,1)
plot(maxcorna)
```



```
% Fréquentiel
maxcorfft=[];
for i=181:length(NormeAcc)
    fft1 = abs(fft(NormeAcc(i-180:i)));
    fft2 = abs(fft(modeleNA));
    xcorfft = xcorr(fft1(5:40),fft2(5:40));
    maxcorfft(i) = max(xcorfft)/20;%coefficient arbitraire pour mise à l'échelle
end
figure(1)
subplot(2,2,3)
plot(maxcorfft)

clear modele env mod tmax t

%% Capteurs de force

% Calcul de la moyenne sur la fenêtre
nbr=180;
for i = nbr+1:length(Data)
    for j=1:4
        moyfilt(i,j) = sum(DataOut(i-nbr:i,j+1))/nbr;
    end
end

boolmoy = zeros(length(DataOut),1);
for i =1:length(DataOut4)
    if (moyfilt(i,1)<.2) && (moyfilt(i,3)<.2) && (moyfilt(i,4)>.65)
        boolmoy(i) = 1;
    end
end
figure(1)
subplot(2,2,2)
plot(boolmoy)

%% Traçage des relevés
figure(1)
subplot(2,2,4)
hold on
plot(maxcorn)
plot(boolmoy)
plot(maxcorfft)
title('Relevés utilisés pour les tests de la reconnaissance')
axis([180 5000 0 2])
legend('Corrélation croisée temporelle accéléromètre',...
    'Tests booléens capteurs de force',...
    'Corrélation croisée fréquentielle accéléromètre')

figure(1)
subplot(211)
plot(Data(:,1:3))
title('Norme de l''accélération avant filtrage')
axis([0 5000 0 1])
subplot(212)
plot(Data(:,4:7))
axis([0 5000 0 1])
axis([0 5000 -.05 1])
legend('ArD','AvD','ArG','AvG')
title('Relevés des capteurs de force avant filtrage')

figure(2)
subplot(211)
```



```
plot(DataOut(:,1))
title('Norme de l'accélération après filtrage')
axis([0 5000 0 .3])
subplot(212)
plot(DataOut(:,2:5))
axis([0 5000 0 1])
axis([0 5000 -.05 1])
legend('ArD','AvD','ArG','AvG')
title('Relevés des capteurs de force après filtrage')

figure(3)
hold on
plot(maxcorna)
plot(boolmoy)
plot(maxcorfft)
plot(.48*ones(1,5000))
title('Relevés utilisés pour les tests de la reconnaissance')
axis([180 5000 0 2])
legend('Maximum corrélation croisée temporelle accéléromètre',...
       'Tests booléens capteurs de force',...
       'Maximum corrélation croisée fréquentielle accéléromètre','Seuil pour les tests booléens')

%% Test final pour la reconnaissance de mouvement

testfin=[];
for i=1:4830
    if(maxcorna(i)>.48) && (maxcorfft(i)>.48) && (boolmoy(i)==1)
        testfin(i)=1;
    else
        testfin(i)=0;
    end
end
figure(4)
plot(testfin)
axis([0 5000 -.1 1.1])
title('Résultat du test de reconnaissance')
```



8.2 Annexe 2 : Application Android pour la commande des moteurs à courant continu des articulation du robot ERICC

8.2.1 L'activité principale

```
package uqac.ca.pwmapp;

import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;

import ioio.lib.util.IOIOLooper;
import ioio.lib.util.android.IOIOActivity;

public class MainActivity extends IOIOActivity implements IOIOListener{
    /**
     * *****
     *                               Paramètres pour le IOIO
     * *****
     */

    // Génération de PWM
    private IOIOVibe vibe;
    private ToggleButton connectionButton;
    private double[] dutyCycle = {0.5,0.5,0.5};
    private boolean connected = false;
    // // Lecture analogique des potentiomètres des moteurs
    // private ToggleButton acquisitionButton;
    // private boolean acquisition;
    // // Fonction Replay
    // private ToggleButton replayButton;
    // private boolean replay;
    /**
     * *****
     *                               Paramètres pour l'affichage de l'orientation
     * *****
     */
    SensorManager manager;
    Sensor accelerometre;
    Sensor magnetometre;

    private TextView textviewa;
    private TextView textviewm;
    private TextView textviewo;
    private TextView textviewox;
    private TextView textviewoy;
    private TextView textviewoz;
    private TextView textviewdcX;
    private TextView textviewdcY;
    private TextView textviewdcZ;
    private TextView textviewfreq;

    float[] mAccelerometreValues = new float[3];
    float[] mMagnetometreValues = new float[3];
    // Matrice qui enregistre les informations de rotation fournies par le telephone
    float[] mRotationMatrix = new float[9];
    // Matrice qui enregistre les angles de rotations du telephone en radians
}
```



```
float[] orientationVals = new float[3];

@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);

    this.vibe = new IOIOVibe();
    this.vibe.addIOIOListener(this);

    textViewfreq=(TextView) findViewById(R.id.textView10);

    /*****
    /*          Gestion des capteurs pour la détection de la position          */
    *****/
    manager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    textviewa=(TextView) findViewById(R.id.textView1);
    textviewm=(TextView) findViewById(R.id.textView2);
    textviewo=(TextView) findViewById(R.id.textView3);
    textviewox=(TextView) findViewById(R.id.textView4);
    textviewoy=(TextView) findViewById(R.id.textView5);
    textviewoz=(TextView) findViewById(R.id.textView6);
    textviewdcX=(TextView) findViewById(R.id.textView7);
    textviewdcY=(TextView) findViewById(R.id.textView8);
    textviewdcZ=(TextView) findViewById(R.id.textView9);

    // Testons la présence de l'accélérateur
    if (manager.getSensorList(Sensor.TYPE_ACCELEROMETER).size() > 0)
    {
        accelerometre = manager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);

        textviewa.setText("Accélérateur présent...");
    }
    else
        textviewa.setText("Il n'y a pas d'accélérateur présent sur cet appareil !");

    // Testons la présence du magnétomètre
    if (manager.getSensorList(Sensor.TYPE_MAGNETIC_FIELD).size() > 0)
    {
        magnetometre = manager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);

        textviewm.setText("Magnétomètre présent...");
    }
    else
        textviewm.setText("Il n'y a pas de magnétomètre présent sur cet appareil !");

    this.connectionButton = (ToggleButton) findViewById(R.id.toggleButton);
    this.connectionButton.setChecked(false);
    this.connectionButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            connected = !connected;
            if(vibe != null) {
                vibe.toggleBluetooth();
            }
        }
    })
}
```



```
});  
  
//      this.acquisitionButton = (ToggleButton)findViewById(R.id.toggleButton2);  
//      this.acquisitionButton.setChecked(false);  
//      this.acquisitionButton.setOnClickListener(new View.OnClickListener() {  
//          @Override  
//          public void onClick(View view) {  
//              acquisition = !acquisition;  
//              if(vibe != null) {  
//                  vibe.toggleAcquisition();  
//              }  
//          }  
//      });  
//  
//      this.replayButton = (ToggleButton)findViewById(R.id.toggleButton3);  
//      this.replayButton.setChecked(false);  
//      this.replayButton.setOnClickListener(new View.OnClickListener() {  
//          @Override  
//          public void onClick(View view) {  
//              replay = !replay;  
//              if(vibe != null) {  
//                  vibe.toggleReplay();  
//              }  
//          }  
//      });  
//  
}   
  
private void changePattern()  
{  
    /*  
    ***** Calcul du dutyCycle à partir de l'orientation du téléphone *****  
    */  
  
    dutyCycle[0] = ((orientationVals[2]) / (2 * 3.141592*250) + dutyCycle[0]);  
    dutyCycle[1] = ((orientationVals[1]) / (2 * 3.141592*250) + dutyCycle[1]);  
    dutyCycle[2] = ((orientationVals[1]) / (2 * 3.141592*250) + dutyCycle[2]);  
  
    if(this.vibe != null) {  
        this.vibe.setPattern(dutyCycle);  
        textViewdcX.setText("Rapport Cyclique X : " + String.valueOf(dutyCycle[0]));  
        textViewdcY.setText("Rapport Cyclique Y : " + String.valueOf(dutyCycle[1]));  
        textViewdcZ.setText("Rapport Cyclique Z : " + String.valueOf(dutyCycle[2]));  
    }  
  
    @Override  
    protected IOIOLooper createIOIOLooper()  
    {  
        return this.vibe;  
    }  
  
    @Override  
    public void ioioConnected()  
    {  
        runOnUiThread(new Runnable() {  
            @Override  
            public void run() {  
                ((TextView)findViewById(R.id.ioioTV)).setText("IOIO Connected");  
            }  
        });  
    }  
}
```



```
@Override
public void ioioDisconnected()
{
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(MainActivity.this, "IOIO Disconnected", Toast.LENGTH_LONG).show();
            ((TextView) findViewById(R.id.ioioTV)).setText("IOIO disonnected");
        }
    });
}

// Sur le resume, installation du listener des capteurs
@Override
public void onResume ()
{
    super.onResume ();

    if (manager.getSensorList(Sensor.TYPE_ACCELEROMETER).size() > 0)
    {
        manager.registerListener(mySensorListener, accelerometre,
SensorManager.SENSOR_DELAY_NORMAL);
    }
    if (manager.getSensorList(Sensor.TYPE_MAGNETIC_FIELD).size() > 0)
    {
        manager.registerListener(mySensorListener, magnetometre,
SensorManager.SENSOR_DELAY_NORMAL);
    }
}

// Si notre activité est en pause, défaire le listener qui ne sert plus à rien...
@Override
protected void onPause() {

    super.onPause ();
    manager.unregisterListener(mySensorListener);

}

// Classe interne pour gérer le listener
SensorEventListener mySensorListener = new SensorEventListener() {

    @Override
    public void onAccuracyChanged(Sensor arg0, int arg1) {

    }

    @Override
    // Si l'accélérateur change, alors afficher les nouvelles valeurs...
    public void onSensorChanged(SensorEvent event) {

        int SensorType = event.sensor.getType();
        switch(SensorType) {
            case Sensor.TYPE_MAGNETIC_FIELD:
                // Mesure des valeurs du magnétomètre
                mMagnetometreValues = event.values;
                break;
            case Sensor.TYPE_ACCELEROMETER:
                // Mesure des valeurs de l'accéléromètre
                mAccelerometreValues = event.values;
                break;
            default:

```



```
        return;
    }

    SensorManager.getRotationMatrix(mRotationMatrix, null, mAccelerometreValues, mMagnetometreValues);
    SensorManager.getOrientation(mRotationMatrix, orientationVals);

    //      textviewox.setText("Xo:" + String.valueOf(orientationVals[0]));
    textviewoy.setText("Yo:" + String.valueOf(orientationVals[1]));
    textviewoz.setText("Zo:" + String.valueOf(orientationVals[2]));
    changePattern();
    }
};

@Override
protected void onDestroy() {
    super.onDestroy();
    //      vibe.onDestroy();
}
}
```

8.2.2 Classe décrivant le comportement du IOIO

```
package uqac.ca.pwmapp;

import android.os.Environment;
import android.util.Log;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;

import java.util.ArrayList;
import java.util.Timer;
import java.util.TimerTask;

import ioio.lib.api.AnalogInput;
import ioio.lib.api.IOIO;
import ioio.lib.api.PwmOutput;
import ioio.lib.api.exception.ConnectionLostException;
import ioio.lib.util.IOIOLooper;

public class IOIOVibe implements IOIOLooper
{
    // Génération PWM
    private PwmOutput vibex, vibey, vibez;
    private static int VIBE_PINx = 10;
    private static int VIBE_PINy = 11;
    private static int VIBE_PINz = 12;
    private static int VIBE_FREQ = 10000;
    private double[] dutyCycle = {0.5, 0.5, 0.5};
    // // Lecture capteurs analogiques
    // private AnalogInput[] input;
    // private int[] list_AnalogInput = {40, 41, 42, 43};
    // private StringBuffer buff;
    // private int compteur acquisition;
}
```



```
// private Timer mTimerWriteFile;
// // Gestion de fichier
// private String base_file_name = "Sortie_IOIOCollect_Mesure_v";
// private int version = -1;
// private File dir;
// private String tag = "IOIOCollectv2";

private ArrayList<IOIOListener> listeners = new ArrayList<IOIOListener>();

private boolean startBT = false; // Connection Bluetooth activée/désactivée
// private boolean startAc = false; // Acquisition activée/désactivée
// private boolean startRe = false; // Replay de trajectoire enregistrée

// private float[] tab;
// private int frequence = 100;
// private int msecacq = 0;

@Override
public void setup(IOIO ioio) throws ConnectionLostException, InterruptedException
{
    try {
        this.vibex.setDutyCycle((float) this.dutyCycle[0]);
        this.vibey.setDutyCycle((float) this.dutyCycle[1]);
        this.vibez.setDutyCycle((float) this.dutyCycle[2]);
        this.vibex = ioio.openPwmOutput(this.VIBE_PINx, this.VIBE_FREQ);
        this.vibey = ioio.openPwmOutput(this.VIBE_PINy, this.VIBE_FREQ);
        this.vibez = ioio.openPwmOutput(this.VIBE_PINz, this.VIBE_FREQ);

        for(IOIOListener l : this.listeners)
        {
            l.ioioConnected();
        }
    }

    // dir = Environment.getExternalStorageDirectory();
}
catch (Exception e)
{
    Log.e("IOIO_EX", "An exception in setup", e);
}

// input_ = new AnalogInput[list_AnalogInput.length];
//
// buff = new StringBuffer();
// //buff.setLength(max_buff_size);
// compteur_acquisition = 0;
// msecacq = 1000/frequence;
//
// mTimerWriteFile = new Timer();
// mTimerWriteFile.scheduleAtFixedRate(Inscription_Fichier, 400, 400);
//
// tab = new float[list_AnalogInput.length];
// for(int i=0; i < list_AnalogInput.length; i++){
//     input_[i] = ioio.openAnalogInput(list_AnalogInput[i]);
// }
//
// // buffer initialization
// for (int i = 0; i < list_AnalogInput.length; i++) {
//     input_[i].setBuffer(500);
// }
}

@Override
```



```
public void loop() throws ConnectionLostException, InterruptedException
{
    Thread.sleep(1);
    // acquisition a 1KHz

    if(this.startBT)
    {
        this.vibex.setDutyCycle((float) this.dutyCycle[0]);
        this.vibey.setDutyCycle((float) this.dutyCycle[1]);
        this.vibez.setDutyCycle((float) this.dutyCycle[2]);
    }

    //      if(this.startAc)
    //      {
    //          for (int i = 0; i < tab.length; i++)
    //          {
    //              tab[i] = input_[i].readBuffered();
    //          }
    //          buff.append(compteur_acquisition + "\t" + tab[0] + "\t" + tab[1] + "\t" + tab[2] +
    // "\t" + tab[3] + "\n");
    //          compteur_acquisition++;
    //      }

    }

    // Activation du Bluetooth
    public void toggleBluetooth()
    {
        this.startBT = !this.startBT;
    }

    // //Activation de l'acquisition
    // public void toggleAcquisition ()
    // {
    //     this.startAc = !this.startAc;
    //     if(startAc) {
    //         version++;
    //     }
    // }
    // //Activation du replay
    // public void toggleReplay ()
    // {
    //     this.startRe = !this.startRe;
    //     if(startRe) {
    //         version++;
    //     }
    // }

    public void setPattern(double[] dutyCycle)
    {
        this.dutyCycle = dutyCycle;
    }

    @Override
    public void disconnected()
    {
        for(IOIOListener l : this.listeners)
        {
            l.ioioDisconnected();
        }
    }
}
```



```
@Override
public void incompatible()
{
}

public void addIOIOListener(IOIOListener l)
{
    if(l != null)
    {
        this.listeners.add(l);
    }
}

// TimerTask Inscription_Fichier = new TimerTask() {
//
//     @Override
//     public void run() {
//
//         final int longueur = buff.length();
//
//         if (longueur > 0) {
//             WriteFile(base_file_name + String.valueOf(version) + ".txt", buff, longueur);
//             buff.delete(0, longueur);
//         }
//     }
// };

/*****
/*                               Écriture dans un fichier                               */
*****/
// public void WriteFile(String filename, StringBuffer b, int endindex) {
//
//     try {
//         File fOut = new File(dir, filename);
//         if (!fOut.exists()) {
//             try {
//                 fOut.createNewFile();
//             } catch (IOException e) {
//                 e.printStackTrace();
//             }
//         }
//         BufferedWriter buf = new BufferedWriter(new FileWriter(fOut, true));
//         buf.append(b.subSequence(0, endindex));
//         buf.flush();
//         buf.close();
//     } catch (FileNotFoundException e) {
//         Log.e(tag, e.toString());
//     } catch (IOException e) {
//         Log.e(tag, e.toString());
//     }
// }

// public void onDestroy(){
//     if (buff != null) {
//         WriteFile(base_file_name + String.valueOf(version) + ".txt", buff,
// compteur_acquisition);
//     }
// }
}
```



8.2.3 Manifeste de l'application

```
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="uqac.ca.pwmapp" >

    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

    <uses-feature
        android:name="android.hardware.sensor.accelerometer"
        android:required="true" />
    <uses-feature
        android:name="android.hardware.sensor.compass"
        android:required="true" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:screenOrientation="portrait"
            android:configChanges="orientation" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

8.2.4 Gestion de la fenêtre graphique

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Information Accéléromètre"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Information Magnétomètre"
```



```
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_below="@+id/textView1"
        android:layout_alignParentLeft="true"/>

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/activity_vertical_margin"
    android:text="Orientation (radian)"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_below="@+id/textView10"
    android:layout_alignParentLeft="true"/>

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="X : "
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_below="@+id/textView3"
    android:layout_alignParentLeft="true"/>

<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Y : "
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_below="@+id/textView4"
    android:layout_alignParentLeft="true"/>

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Z : "
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_below="@+id/textView5"
    android:layout_alignParentLeft="true"/>

<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/activity_vertical_margin"
    android:text="Rapport cyclique : "
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_below="@+id/textView6"
    android:layout_alignParentLeft="true"/>

<TextView
    android:id="@+id/textView8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Rapport cyclique : "
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_below="@+id/textView7"
    android:layout_alignParentLeft="true"/>

<TextView
    android:id="@+id/textView9"
```



```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Rapport cyclique : "
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_below="@+id/textView8"
        android:layout_alignParentLeft="true"/>

<TextView
    android:id="@+id/textView10"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Frequence PWM : 10kHz"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_below="@+id/textView2"
    android:layout_alignParentLeft="true"/>

<!--<RelativeLayout-->
    <!--android:orientation="horizontal"-->
    <!--android:layout_width="fill_parent"-->
    <!--android:layout_height="fill_parent"-->
    <!--android:layout_alignParentLeft="true"-->
    <!--android:id="@+id/linear"-->
    <!--android:layout_below="@+id/textView9"-->

    <TextView
        android:text="Statut"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/ioioTV"
        android:layout_alignParentBottom="true"
        android:layout_toRightOf="@+id/toggleButton"
        android:layout_marginLeft="25dp"
        android:layout_marginBottom="15dp" />

    <ToggleButton
        android:text="@string/connectButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/toggleButton"
        android:textOff="@string/connexionText"
        android:textOn="@string/disconnectString"
        android:layout_alignParentBottom="true"/>
    <!--android:layout_toLeftOf="@+id/toggleButton3" />-->

    <!--<ToggleButton-->
        <!--android:text="Acquisition"-->
        <!--android:layout_width="wrap_content"-->
        <!--android:layout_height="wrap_content"-->
        <!--android:id="@+id/toggleButton2"-->
        <!--android:textOff="@string/acquisitionRun"-->
        <!--android:textOn="@string/acquisitionStop"-->
        <!--android:layout_marginLeft="35dp"-->
        <!--android:layout_alignParentTop="true"-->
        <!--android:layout_alignParentLeft="true"-->
        <!--android:layout_marginTop="25dp" />-->

    <!--<ToggleButton-->
        <!--android:text="Replay"-->
        <!--android:layout_width="wrap_content"-->
        <!--android:layout_height="wrap_content"-->
        <!--android:id="@+id/toggleButton3"-->
        <!--android:textOff="@string/replayRun"-->
```



```
<!--android:textOn="@string/replayStop"-->  
<!--android:layout_alignTop="@+id/toggleButton2"-->  
<!--android:layout_toRightOf="@+id/toggleButton2" />-->  
<!--</RelativeLayout>-->
```

```
</RelativeLayout>
```



8.3 Annexe 3 : Diagramme de Gantt du projet à l'achèvement du stage

