
Rapport de Stage de Fin d'Etudes

Préparé pour : Luc PENAZZI, Eric CAMPO, Yoann CHARLON

Préparé par : Elodie BOUZBIB

Promotion 2015, Domaine IMAS

Date : du 28/02/2015 au 28/08/2015

École des Mines d'Albi - LAAS-CNRS



**Semelle intelligente pour les personnes
âgées :**

**Mesure de la variation du poids et comptage
de pas**

Remerciements

Les **nouvelles technologies** font à présent partie de notre vie. Les ingénieurs développent de nombreux objets pour renforcer notre bien être, et désormais il est possible de **suivre son état de santé**, afin de **prévenir d'éventuels problèmes**. Depuis plusieurs années, je souhaite travailler dans le domaine de **l'ingénierie biomédicale**. En effet, la médecine est un domaine fascinant, en constante évolution. Après un stage d'assistant ingénieur dans le domaine des diagnostics avancés, qui contenait beaucoup de chimie – qui n'est pas mon domaine de prédilection- j'ai eu la chance de réaliser mon stage au LAAS-CNRS, dans le domaine de la **domotique pour la santé**.

Etant fascinée par la **micro-électronique**, et possédant un grand intérêt pour la **programmation**, ce stage était la combinaison parfaite de mes attentes en terme de travail. Je tiens premièrement à remercier le docteur Julien Reboud, qui depuis mon stage en Écosse, m'avait conseillée de postuler au LAAS, sachant que j'y trouverai mon bonheur. Je souhaite ensuite remercier le professeur Eric Campo, qui dès le mois de Septembre, a accepté de porter de l'attention à mon profil, bien que mon cursus scolaire ne corresponde pas exactement à celui de ses stagiaires habituels. Je le remercie ensuite évidemment de m'avoir fait confiance, et de m'avoir encouragée tout au long de mon stage.

Je voudrais tout particulièrement remercier le docteur Yoann Charlon, qui m'a **transmis son savoir** en microélectronique avec **patience**, mais surtout qui a pris en compte toutes mes propositions au long du projet. Ce sujet étant le prolongement de son travail de thèse, cela devait lui tenir à cœur d'avoir une personne compétente pour l'aider à avancer dans son travail. J'espère avoir répondu à toutes ses attentes. Je le remercie aussi d'avoir participé à ma soutenance, et de **m'avoir épaulée et soutenue tout au long du stage**. Je souhaite moi aussi devenir docteur, et j'espère pouvoir développer un sujet aussi intéressant, avec autant de clarté que lui. Merci aussi pour ses conseils en termes de recherche de thèse, et son côté humain.

Enfin, je remercie Luc Penazzi, mon tuteur-école, qui m'a encadrée toute l'année dans sa majeure à **l'Ecole des Mines d'Albi**, et qui a tout de même accepté de m'encadrer quelques mois de plus pour ce stage de fin d'études. Merci pour ses conseils, tout au long de l'année, Il m'a soutenue lorsque je lui parlais d'ingénierie biomédicale – malgré mon cursus orienté matériaux aéronautiques- ou d'idée de thèse. Il a fait preuve d'une grande sagesse, patience, et sympathie. J'espère que je lui laisserai un bon souvenir, malgré mon tempérament parfois trempé !

Grâce à ce stage, je peux confirmer mes **ambitions de carrière de chercheuse**. Je tiens donc à remercier tous ceux qui m'ont entraînée à choisir cette voie, et soutenue dans cette direction. Je suis heureuse de pouvoir sortir de l'Ecole, diplômée, et avec une **idée précise de la vie que je souhaite mener dans ma carrière professionnelle**.

Table des Matières

Remerciements.....	2
Résumé	4
Abstract.....	5
Introduction	6
Première Partie : Rapport d'Action	7
Objet de la mission	7
Les résultats	9
Justification des résultats	13
Perspectives et Suites	14
Deuxième Partie : Rapport d'Étude	15
Présentation de mon TFE – Travail de Fin d'Études	15
I. Présentation du LAAS-CNRS.....	15
II. Présentation du Contexte de Stage	16
III. Finalités et But de ma mission.....	17
Déroulement de l'étude.....	19
I. Mise à jour du comparatif des méthodes de classification dans le cadre de la mesure du poids embarqué.....	19
1. Méthode discriminative ou générative ?	19
2. Comparaison des méthodes existantes, en fonction de la précision des résultats	21
3. Exploitation des résultats vers une nouvelle méthode phénoménologique efficace.....	29
4. Optimisation du code embarqué.....	31
II. Algorithmes.....	34
1. Mesure de la variation du poids embarqué.....	34
2. Détection de pas : Amélioration de l'existant.....	37
Conclusion.....	41
I. Bilan du projet	41
II. Synthèse des résultats	41
III. Suite à donner au projet, perspectives	42
Troisième Partie : Bilan personnel de mon projet professionnel	44
Bibliographie	46
Table des Illustrations et des Annexes	48
Annexes.....	50

Résumé

La santé, devenue un secteur de recherche majeur, commence peu à peu à s'adapter aux nouvelles technologies, et à profiter du développement des objets **connectés**. Nous faisons face à la fois à une augmentation des personnes voulant **suivre par eux-mêmes leur état de santé** sans avoir à faire appel à des médecins, mais également à un vieillissement démographique accéléré. Afin de pallier aux besoins des personnes âgées, et de juger leur forme physique, le groupe N2IS du LAAS-CNRS mène un projet national (RESPECT) de conception et d'évaluation d'un système de suivi discret **via une semelle intelligente**.

La mesure de l'activité motrice est un indicateur pour le praticien de l'état de santé de la personne âgée. L'objectif du projet est de développer une instrumentation médicale pour mettre en place un suivi de l'activité à domicile. La semelle doit alors être capable de suivre plusieurs paramètres liés à la marche ainsi que la **perte de poids**.

Lors de mon stage, je dois mettre au point l'algorithme de mesure de la variation de poids embarqué dans la semelle. En cas de variation de poids, une application Android embarquée dans la semelle devra être capable de le communiquer à un médecin, afin qu'il anticipe les éventuels problèmes de santé de son patient.

Mots-clés : domotique médicale, semelle intelligente, microprocesseur, programmation, électronique embarquée, actimétrie, communication sans-fil, mesure du poids, marche

Abstract

Healthcare has become a major field of scientific research, and is beginning to merge with new technologies, to become **connected**. We are facing a huge increase of people who want to **self-evaluate** their abilities, without having to meet a doctor, but also an important demographic aging.

In order to judge the **elderly health**, researches at the N2IS laboratory – at LAAS-CNRS, have lead to the conception of an **automated tracking object**, discreet but still **accurate, shaped into a smart insole**, and called RESPECT.

Measurement of **motor activity** indicates to a doctor if the health of one of his elderly patient is decreasing. The main objective of my placement is to develop a medical instrumentation to monitor someone's activity at home. Thus, the smart insole must be able to track some walking criteria, but also to notice a loss of weight.

During my placement, I have to design an efficient embedded algorithm that describes a **variation of weight** in the insole. If someone's weight is slightly decreasing, the Android application that is linked to the insole has to communicate it to a doctor, for him to **prevent his patient** from possible health issues.

Keywords: home medicine automation, smart insole, microprocessor, programming, wireless communication, embedded electronics, actimetry, weighing measurement, walking

Introduction

La **surveillance des personnes âgées à domicile** est un objectif bien identifié dans la politique de santé publique actuelle qui tend à privilégier le maintien au domicile et la **prévention de la perte d'autonomie**. Elle s'impose notamment par le rallongement de la durée de vie et le **vieillessement relatif de nos populations**. Elle est aussi une stratégie qui devrait apporter un plus grand confort aux patients concernés, en leur permettant de rester dans leur lieu de vie habituel et contribuer à une meilleure maîtrise des dépenses globales de santé. La **domotique médicale**, par le biais des outils technologiques, peut permettre un suivi afin de **détecter toute détérioration de l'état de santé et de porter assistance**. Il est possible, par le caractère ergonomique et transparent des technologies récentes de s'assurer une acceptabilité optimisée en contexte d'usage.

La **prévention de la dépendance** et le maintien au domicile des personnes âgées fragiles est un défi de taille. La fragilité est un concept émergent qui traduit un risque élevé de décès, et plus généralement de dépendance. Des corrélations entre la fragilité et certains paramètres sont actuellement bien établis : vitesse de marche, perte involontaire de poids, activité spontanée... Ainsi, l'évaluation et le suivi de la marche sont des éléments clés potentiellement prédictifs d'évolution morbide chez le sujet âgé. Les critères de Fried sont actuellement largement utilisés dans les études cliniques pour caractériser les **patients fragiles**. Ils reposent essentiellement sur l'analyse de la composante physique de la fragilité.

Le suivi des patients âgés fragiles se fait actuellement lors de consultations gériatriques hospitalières et donc hors du contexte de vie et de manière discontinue – tous les 3 mois. On ne sait rien du sujet entre ces intervalles, et on ne dispose pas de données dynamiques. Il y a un **réel intérêt quant au suivi sur le lieu de vie**, en contexte réel et dynamique. Cela permet de connaître la cinétique d'évolution, mais également d'**accéder à une étude plus centrée entre le patient et son médecin traitant**, notamment en **zones sous médicalisées**.

Les intérêts d'une mesure automatique à domicile sont :

- de pouvoir être répétée et donc de réaliser un **suivi longitudinal** sur le moyen et le long terme,
- d'être assurée de réaliser des mesures systématiques dans des conditions comparables et écologiques,
- de pouvoir détecter des variations significatives au plus tôt.

Ainsi un suivi personnalisé à domicile de personnes fragiles devrait permettre d'adapter la **prise en charge en fonction de l'évolution des critères de fragilité** de manière plus précoce et ainsi de **prévenir la dépendance** afin de permettre aux personnes âgées fragiles de **rester le plus longtemps à domicile**.

Dans ce contexte, le LAAS-CNRS développe un projet de recherche avec pour objectif de **concevoir une semelle de chaussure intelligente** pour suivre la marche des personnes âgées à domicile, mais aussi leur variation de poids. Je développerai dans une première partie le contexte du stage, puis expliquerai les étapes déterminantes de mon projet, à travers des **classifications d'activité**, puis l'élaboration de **plusieurs algorithmes**, définissant l'activité et la perte de poids, et finalement leur **mise en œuvre et les tests réalisés**.

Première Partie : Rapport d'Action

Objet de la mission

Nous faisons face actuellement à un **vieillesse démographique** général, entraînant une augmentation des handicaps et une hausse de personnes âgées vivant seules (95% des plus de 75 ans en France).

Les **nouvelles technologies permettent aujourd'hui de proposer des solutions innovantes de suivi de santé à domicile**. Le groupe N2IS du LAAS-CNRS travaille sur une solution innovante de suivi de la marche dont le but est d'améliorer la qualité de vie des seniors, à travers une semelle intelligente connectée.

L'âge, les pathologies, l'environnement et le milieu social sont des facteurs qui peuvent entraîner une **vulnérabilité** des personnes âgées. Lorsque ces facteurs se dégradent, la sensibilité aux situations de stress s'en suit, ce qui peut entraîner une **perte d'autonomie** de la personne (syndrome de la fragilité). Les **critères physiologiques de Fried**, permettent d'identifier une personne fragile (voir page 16).

La fragilité est définie comme un état de pré-dépendance mais elle **réversible à l'aide généralement d'un programme de remise en forme basé sur la marche**. L'objectif de la semelle instrumentée est de proposer un système de suivi de la marche dont les données d'activité seront accessibles par le médecin et la personne sur une application Web. L'objectif est de créer un système de prévention, de renforcement de l'autonomie, et d'encouragement à l'activité, de « **coaching motivationnel** ».

La semelle devra relever les paramètres suivants : la vitesse et la cadence de marche, le nombre de pas, la distance parcourue, les périodes d'activité, ainsi que la **variation du poids**.

Les principales **contraintes** de conception sont la **simplicité de mise en œuvre**, la **robustesse**, la **fiabilité**. Nous souhaitons créer un système discret, confortable, et **bon marché**. Aussi, ce système doit être connecté et intelligent pour permettre un suivi de santé distant et faire de la prévention. Le schéma de la figure 1 présente l'architecture générale du système RESPECT.

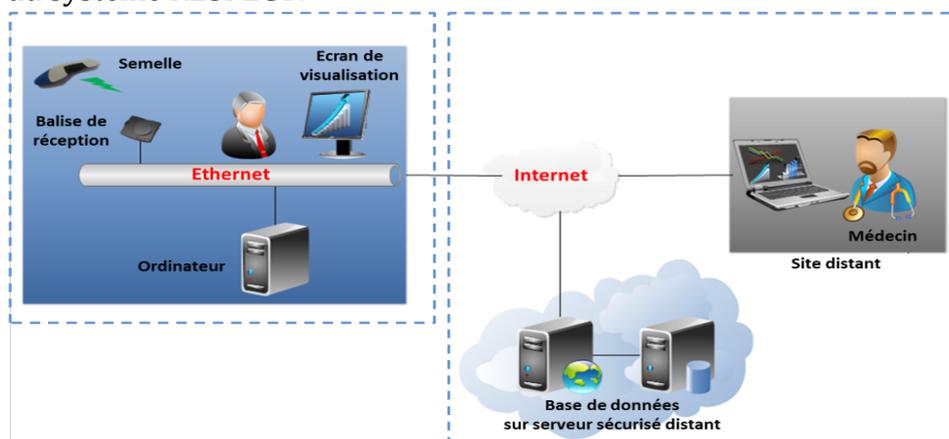


Figure 1 - L'architecture du système opérationnel

La perte involontaire de poids est un indicateur de perte d'autonomie (critère de Fried). Nous voulons donc développer un algorithme, répétable et précis, mesurant la variation

du poids. Seul le LAAS-CNRS s'est intéressé à cette mesure de manière embarquée, c'est donc une caractéristique innovante de notre semelle par rapport à l'existant.

L'objectif de ma mission sera de répondre à plusieurs questions : quand voulons nous relever le poids ? Quelle caractéristique de ce poids sera la plus significative ? Avons-nous besoin d'utiliser à la fois un capteur au niveau du talon et un capteur au niveau du métatarse, ou un seul d'entre eux suffira ?

Ces deux positions (figure 2) ont été définies comme les deux points forts en pression lors de la marche. C'est à cet endroit que, nous positionnerons des capteurs de pression pour mesurer le poids. Le module électronique de traitement des données sera intégré dans la semelle et placé sous la voûte plantaire, où la sensibilité est faible, pour ne pas gêner la personne.

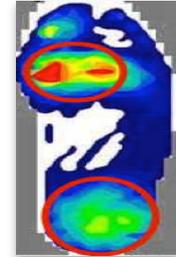


Figure 2 - Points forts en pression, positions des capteurs

La problématique principale de ce stage est centrée autour de la **mesure de la variation du poids**, L'objectif du système de pesée est d'alerter le médecin en cas de perte de poids jugée importante (5 à 10 % du poids total de la personne). Une alerte doit être déclenchée lors du dépassement d'un seuil qui doit être configurable et défini par le médecin traitant (prise en charge personnalisée). Pour accomplir cet objectif, le système devra être capable de détecter une variation de 3 kg.

Afin de réaliser cette mesure, nous souhaitons relever le poids de la personne dans des conditions de marche identiques, c'est à dire lors d'**une marche à vitesse normale sur du plat**. Pour cela, nous proposons de mettre en place une méthode de classification des activités de marche afin de détecter cette condition.

Cette méthode devra respecter la contrainte de simplicité de mise en œuvre, car le programme est destiné à être embarqué dans la semelle. Sa **reproductibilité** sera par ailleurs primordiale, puisque **l'algorithme sera utilisé plusieurs fois par jour**, afin de proposer une **mesure la plus précise possible**. Il faudra également **limiter la consommation énergétique** du système et donc ne pas utiliser une méthode trop gourmande en ressource informatique.

Dans un second temps, je proposerai une optimisation de l'algorithme de détection de pas existant pour en améliorer sa précision. Je présenterai dans ce rapport la mise en œuvre de cet algorithme de détection de pas.

Mon étude sera principalement centrée sur de la modélisation sous MatLab, puis de la programmation en langage C, ainsi que des périodes de tests en embarqué.

Le principal objectif de ma mission était donc de réaliser une **classification des activités** efficace, et d'implémenter une fonction de variation de poids qui lui soit liée. Néanmoins, comme tout projet de recherche, j'ai du réaliser des **tâches transverses**, comme une **étude bibliographique** ou encore de l'**optimisation de codes existants**.

Les résultats

L'étude bibliographique que j'ai initialement réalisée concernait en priorité les méthodes de classifications existantes. En effet, j'avais besoin de trouver une méthode à la fois simple et efficace, afin de déterminer les différentes activités d'une personne, à l'aide du matériel à ma disposition : une carte de développement multi-capteurs communicant en Bluetooth.

Classification des activités et choix d'une méthode

Il existe deux types de classifications, supervisée ou non-supervisée. J'ai choisi de ne m'intéresser qu'aux classifications supervisées, qui permettent, à partir d'une plage de données, d'évaluer les distributions d'une classe sans la mesurer explicitement, mais en se basant sur des valeurs lui étant liées. En effet, puisque je disposais d'une première base de données et des classes que je souhaitais obtenir – marche, escalier, course, à différentes vitesses, il me suffisait d'observer les erreurs de prédiction des différentes méthodes afin de choisir la plus efficace.

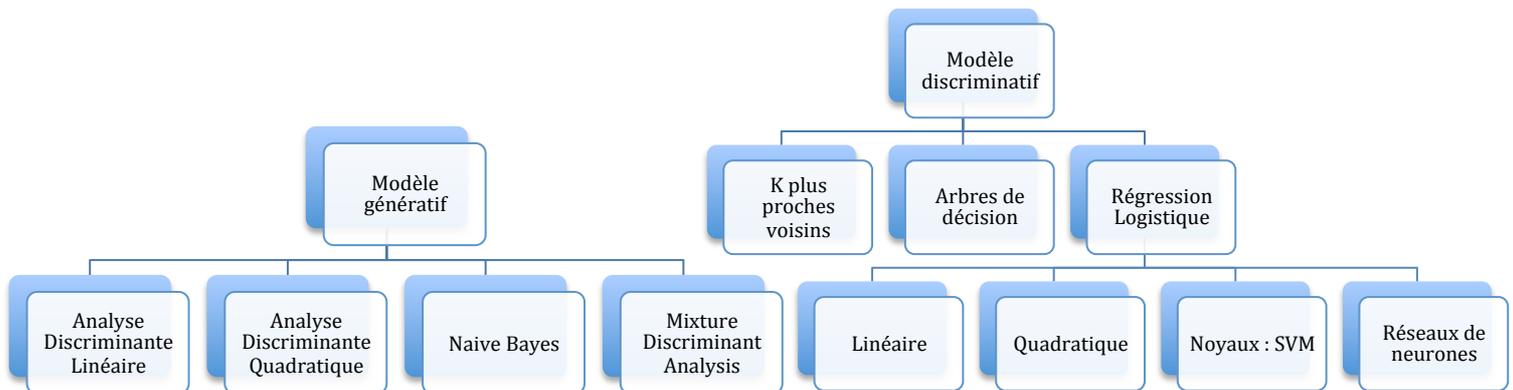


Figure 3 – Les différents types de classifications supervisées

J'ai implémenté et testé sous MatLab toutes ces méthodes de classification supervisée, dont les descriptifs et résultats, en fonction de leur précision, sont présentés à partir de la page 20. Malgré la grande efficacité de plusieurs méthodes, leur mise en œuvre semblait assez compromise, car l'objectif de cette classification reste *in fine* de l'implémenter dans la carte de développement. Cela nous impose en conséquence des contraintes, en consommation énergétique notamment.

J'ai néanmoins remarqué la redondance de certains critères dans ces méthodes, tels que la moyenne ou l'écart-type des données. J'ai ainsi choisi de calculer la moyenne et la variance des amplitudes des accélérations lors d'une activité, données facilement relevables grâce à l'accéléromètre présent dans la carte de développement. Aussi, toujours dans un objectif d'optimisation de l'espace et de la consommation dans la carte, j'ai préféré calculer les carrés de cette amplitude, afin de ne pas faire face à des fonctions « racine ». J'ai alors défini une variable, Var2, « variance des carrés des amplitudes des accélérations », telle que : $Var2 = \sigma_{ai2}^4 = \frac{1}{2w+1} \sum_{j=i-w}^{i+w} (a_{j2} - \bar{a}_{j2})^2$.

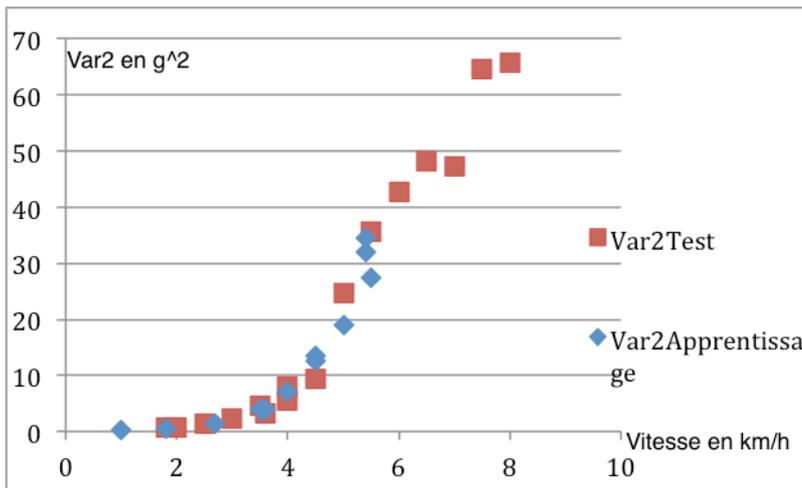
En calculant Var2 sur l'ensemble des fichiers de ma base de données, j'ai réalisé un arbre de classification (méthode et calculs décrits page 29). Les combinaisons contenues dans les embranchements correspondent alors aux carrés des amplitudes des accélérations, de leur moyenne, et leur Var2. Les résultats sont donnés dans le tableau 1 ci-après.

Classe	Lent (plat)	Normal (plat)	Rapide (plat)	Montée	Descente
Course	159	647	104,3		
Marche	0,5	3,8	34,4	34,3	110
Escalier				22,4	18,1

Tableau 1 – Variances des amplitudes au carré en fonction des phases de marche : en montée, descente, et sur du plat à 3 vitesses différentes, en g²

Nous pouvons remarquer, grâce au graphique ci-contre, que **la vitesse de marche est fortement corrélée avec la variance des carrés des amplitudes des accélérations**. Aussi, la variance varie énormément entre une « marche normale sur du plat », et une marche en montée, en descente, ou dans des escaliers, comme le montre le Tableau 1.

En calibrant la semelle pour chaque individu, nous pourrions alors **déterminer un intervalle de Var2** qui définira une phase de marche normale sur du plat.



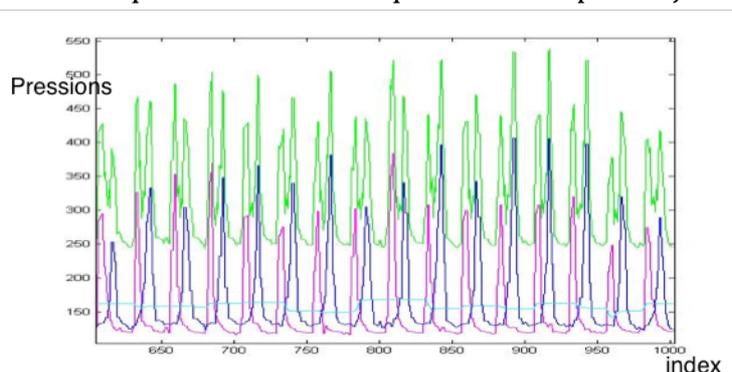
Vitesse (km/h)	1,8	2	2,5	3	3,5	3,6	4	4	4,5	5	5,5	6	6,5	7	7,5	8
Var2 (g ²)	0,78	0,68	1,34	2,43	4,53	3,23	5,60	8,06	9,32	24,7	35,7	42,8	48,2	47,2	64,5	65,7

Figure 4 – Variances des carrés des amplitudes en g² en fonction des vitesses de marche en km/h, en phases d'apprentissage et de test

Dès que l'individu entrera dans cet intervalle, nous vérifierons les conditions de marche à vitesse normale sur du plat. Ainsi, nous commencerons alors à relever les variations de poids, grâce à un nouvel algorithme dont la description est donnée ci-après.

Mesure de la variation du poids embarqué

Lors de cette étude, il m'a été demandé de définir si l'utilisation d'un ou de deux capteurs de pression était plus appropriée à la mesure de la variation de poids, et de définir quelles caractéristiques de ces capteurs je souhaitais relever.



Les données brutes des capteurs de pression embarqués dans la semelle ont été relevées lors d'une marche en continue.

Figure 5 - Variation de la pression sur les capteurs, en continu, en fonction des index - pression au niveau du talon en rose, pression à l'avant en bleu, et somme des deux en vert, moyenne des maxima en cyan

Comme nous pouvons le remarquer les maxima respectifs à l'avant et à l'arrière sont en légèrement décalés, mais ont des valeurs assez similaires. J'ai donc calculé les moyennes des maxima, sur une fenêtre glissante de 25 données, afin de juger de leur monotonie. Il était difficile de trouver un autre critère aussi stable, par exemple en n'utilisant qu'un seul capteur. En utilisant les maxima des deux capteurs, nous retrouvons une des valeurs qui semble la plus caractéristique du poids, car elle témoigne d'un pied posé à plat, pendant la marche.

De plus, j'ai jugé qu'il était plus **rigoureux** de relever les **pressions en continu de chaque activité**, et d'en calculer les moyennes, afin de détecter une **variation** d'une journée sur l'autre. Cela permet de ne pas prendre en compte les petites variations au cours de la journée. Nous avons choisi de relever les moyennes et moyennes des maxima de pressions, comme le montrent les graphiques ci-dessous. Aussi, nous donnons dans le tableau 2 un récapitulatif des variations de pression, en pourcentages, en fonction des poids utilisés pendant des tests.

Index (abscisses de la figure)	Poids supplémentaire (porté)	Moyennes des maxima	Moyennes des moyennes
1 :1890	-	-	-
1890 :3750	+2,6%	+2,7%	+3,9%
3750 :5640	+4,5%	+4,4%	+5,5%
5640 :7470	+5,4%	+4,8%	+5,8%
7470 :7870	+7,2%	+7,3%	+7,0%

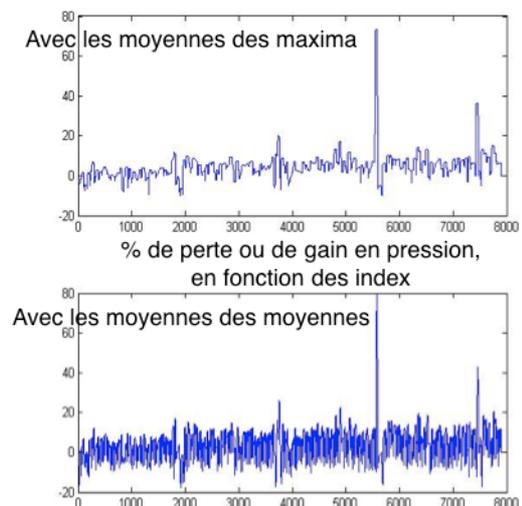


Tableau 2 - Evolution en pourcentages des pressions moyennes et de leurs maxima en fonction des poids appliqués - Evolutions des Moyennes des maxima en haut, moyenne des moyennes en bas

Les variations de quelques pourcentages peuvent être notées sur le graphique des évolutions des moyennes des maxima, alors que nous avons plus de difficultés à remarquer les variations sur les moyennes des moyennes.

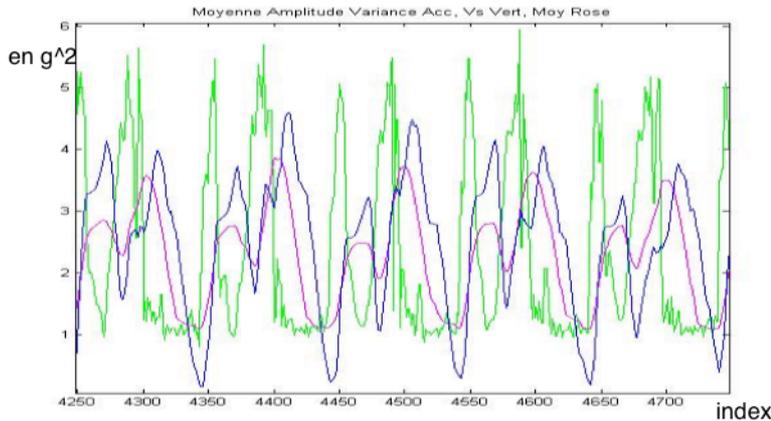
Dès qu'une personne rentre dans son intervalle respectif de Var2 (marche normale sur du plat), je souhaite relever le **maximum de pression de chaque foulée à l'aide d'une fenêtre glissante de 25 échantillons**, puis les moyenner pour chaque activité de marche. En fin de journée, on réalise alors une **moyenne de ces moyennes de maxima**, que nous stockons, sur le serveur de données.

Les résultats sur le serveur seront conservés chaque jour, puis moyennés par semaine. De cette manière, il sera alors possible d'observer les variations d'une semaine sur l'autre et d'alerter le médecin en cas de variation jugée importante (5% du poids de la personne). La classification des activités ainsi que cet algorithme de mesure des

variations de poids constituaient à l'origine l'intégralité de mon stage. Néanmoins, je me sentais capable d'améliorer l'actuel compteur de pas, déjà programmé dans la semelle. Je vais donc à présent décrire ce nouvel algorithme.

Amélioration de l'algorithme « compteur de pas »

La première version de l'algorithme de comptage des pas reposait sur des seuils numériques et la détection de **fronts montants et descendants** de l'amplitude de l'accélération. Des seuils numériques prédéfinis ne peuvent pas s'adapter à tout type de démarche. J'ai donc décidé de développer une méthode plus **astucieuse**.



Comme nous pouvons le remarquer sur la figure 6, Vs2, carré des amplitudes des accélérations, atteint des valeurs minimales de manière périodique, lorsque le pied est à plat sur le sol. C'est le **signe d'une nouvelle foulée**.

Figure 6 - Graphique représentant la marche, avec Vs2 en vert, Moy2 en rose, et Var2 en bleu, en g², en fonction du temps (10 index = 100ms)

J'ai alors décidé de calculer une nouvelle variance, **sans fenêtre glissante**, que j'ai nommée **Var3** :

$$\begin{aligned} Var3_j &= (a_{j2} - \overline{a_{j2}})^2 \\ &= (Vs2 - Moy2)^2 \end{aligned}$$

En effet, si nous calculons la **variance à chaque valeur** - et non pas sur une fenêtre glissante, nous pouvons remarquer que **Var3 croise Moy2 dès que l'on a une nouvelle foulée**.

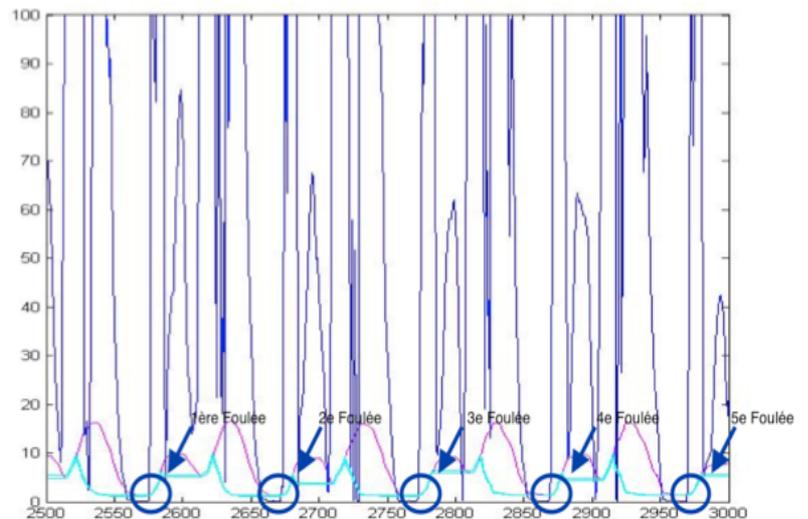


Figure 7 - Mise en évidence des foulées, avec Var3 en bleu, Moy2 en rose, et les minima et seuils de 10% des minima en cyan, en g²

Afin d'optimiser le code en embarqué, j'ai choisi de m'intéresser aux données brutes relevées par l'accéléromètre de la carte de développement. J'ai donc modélisé ma recherche de nouveau critère en calculant les parties entières de Vs2, Moy2 (dont les définitions se situent page 30) et Var3. Comme nous le remarquons sur le graphique de la figure 8 les échelles changent alors considérablement.

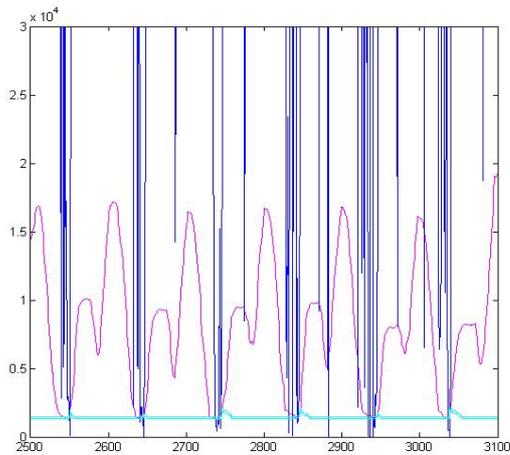


Figure 8 - Modélisation de Var3 et Moy2, données brutes du microprocesseur sans conversion en g

Aussi, nous pouvons noter que l'utilisation de délais de temps devient aussi inutile, puisque Var3 et Moy2 ne se croisent pas lorsque nous arrêtons de marcher.

En conclusion, ce nouveau compteur possède de grands avantages. Premièrement, il ne possède **plus de seuils numériques quant aux variances**, et donc une **liberté de démarche**, mais **aussi quant aux délais de temps**. En effet, pendant les périodes de pause, Var3 et Moy2 ne se croisent pas. Il est donc

inutile d'ajouter des délais de temps quant aux « faux rebonds » ou au manque d'activité par exemple.

Justification des résultats

J'ai commencé mon projet par un état de l'art, puis je me suis concentrée sur les méthodes de classifications couramment utilisées, car je savais qu'elles seraient l'étape la plus difficile à réaliser. Après une étude sur une petite base de données, j'ai décidé d'en réaliser une plus grande, afin de pouvoir confirmer mes différentes suppositions quant à l'utilisation d'une méthode. En la combinant avec les contraintes liées à l'embarqué, la consommation énergétique de la carte de développement et son volume mémoire, j'ai donc trouvé une méthode efficace, grâce à la définition de Var2, variance des carrés des amplitudes des accélérations de la marche.

A partir de ma base de données, et des tracés des courbes de marche que je réalisais, j'ai proposé une méthode de détection des pas. J'ai ensuite pu **programmer la carte de développement à ma disposition**, afin d'y implémenter ses nouveaux codes. A ce stade de mon stage, je teste encore mes codes en embarqué, en testant la semelle, la carte, et une tablette possédant l'application finale.

Mon travail peut se résumer à travers les trois tâches et leurs résultats décrits ci-après :

- **Classification**

On calcule à partir des données d'accélération fournies par l'accéléromètre, le carré du vecteur somme, sa moyenne sur une fenêtre glissante de 25 données, ainsi que sa variance. Un **intervalle de cette variance** pourra ensuite être défini, pendant une période de calibration, pour déterminer une **phase de « marche normale sur du plat »** de la personne à suivre.

Par exemple, supposons qu'une personne marche en moyenne à 4 km/h, il suffira juste de relever Var2 au moment de la calibration, en prenant 10% en dessous et au-dessus de la « vitesse normale ». Une fois situés dans l'intervalle de Var2 calculé, nous commencerons alors à relever les variations de poids.

- **Mesure de la variation du poids**

Lorsque nous marchons dans l'intervalle de fonctionnement de Var2, nous mesurons alors la pression appliquée à la fois au talon et à l'avant du pied. **Deux capteurs sont nécessaires**, pour augmenter la précision de la mesure. Aussi, en relevant les maxima sur des fenêtres glissantes, puis en les moyennant, nous avons une idée de la **pression appliquée lorsque tout le pied est à plat**, ce qui est plus représentatif du poids que seulement le talon ou le métatarse. Chaque jour, la **moyenne des moyennes des maxima** est réalisée et stockée sur le serveur, puis comparée aux données précédentes. Avoir la possibilité de remarquer une perte de 5% du poids sur une semaine permettrait de prévenir le médecin pour une prise en charge de la personne.

- **Amélioration du compteur de pas**

A présent, il n'est **plus nécessaire** d'utiliser des **seuils numériques** dans le code embarqué, afin de mesurer le nombre de pas réalisés. Une caractéristique de la marche, repérée dans mes graphiques, est le **croisement entre Var3**, variance du carré du vecteur somme à chaque donnée, **et Moy2**, la moyenne de ce même carré. Ces deux données ne se croisent qu'une seule fois lorsque nous atteignons un minimum de la moyenne de l'accélération, c'est-à-dire lorsque le pied est à plat et l'accéléromètre contre le sol (**nouvelle foulée**). L'**ajout de délais de temps** en cas de faux rebond ou d'inactivité est alors inutile, puisque sans mouvement, ces données ne se croisent jamais. Le nouveau code permet une **réelle optimisation** de l'existant.

Perspectives et Suites

Les algorithmes développés devraient permettre de suivre l'activité de marche et la variation de poids afin de mettre en place un suivi de personnes fragiles à domicile et faire de la prévention. La robustesse des algorithmes doit dorénavant être testée en conditions réelles. Bien que je teste dès à présent mes codes en embarqué et en conditions réelles, ce n'est pour le moment pas similaire aux conditions de marche d'une personne âgée. Il sera donc nécessaire de **tester les nouveaux algorithmes** sur un panel de personnes de plus de 65 ans. Nous commencerons par des personnes « **robustes** », et après validation, nous pourrions passer aux patients « **fragiles** ».

En outre, certaines fonctions peuvent être ajoutées à la semelle, cette fois-ci dans un cadre de la **surveillance et la détection de dangers**.

En effet, pendant mon stage, de nouvelles perspectives se sont ouvertes. En effet, nous souhaitons à présent ajouter une fonction de **détection de chute** à la semelle. Il permettra de **lancer une alerte** le cas échéant. Les chutes peuvent causer des pertes de connaissance, des fractures... Il est donc très important de les détecter. Des robots du bâtiment ADREAM du LAAS sont conçus pour aider les personnes à se relever en cas de chute, mais ces robots sont encore à l'état de prototypes. La nouvelle semelle aura pour objectif d'appeler à l'aide si une chute survient.

Lors de mon dernier mois de stage, je poursuivrai les tests de la semelle et travaillerai sur le repérage de **différents paramètres** liés à la chute. Ces résultats pourront alors être utilisés à l'avenir, à travers un projet plus complexe que le projet actuel.

Deuxième Partie : Rapport d'Étude

Présentation de mon TFE – Travail de Fin d'Études

I. Présentation du LAAS-CNRS

Le LAAS – **Laboratoire d'Analyse et d'Architecture des Systèmes**, est un centre de recherche, créé en 1968 et situé à Toulouse, il a pour objectifs de **modéliser, concevoir, maîtriser** des systèmes complexes, en interaction avec d'autres systèmes ou avec l'Homme. Il est composé de 22 équipes de recherche, soit 640 personnes dont environ 500 chercheurs, rassemblés autour de quatre disciplines primordiales : la **robotique, l'automatique, l'informatique**, et les **micro nano-systèmes**.



Figure 9 - Le LAAS-CNRS

Le LAAS-CNRS se divise en trois axes stratégiques : **ALIVE, Synergy** et **ADREAM**.

ALIVE consiste au développement de systèmes et dispositifs à l'interface avec les molécules, cellules et tissus du vivant. Cet axe concerne les sciences de l'ingénieur, du vivant et de l'environnement. Synergy a été récemment créé afin de concevoir des systèmes pour une gestion intelligente de l'énergie, et devenir un acteur de la transition énergétique, du composant aux réseaux électriques intelligents.

Enfin, ADREAM, l'axe en lien avec mon stage, représente la **mise en réseau massive d'objets et d'agents intelligents**, afin de développer les domaines de réseaux, de la robotique, des logiciels, pour anticiper le futur à travers la conception d'outils et d'expérimentations.

L'équipe dans laquelle j'ai été intégrée est **N2IS -Nano-Ingénierie et Intégration des Systèmes**. L'objectif est de mettre en œuvre de nouveaux concepts et méthodes pour le diagnostic de proximité, à l'aide de nouveaux microsystèmes intégrés dédiés à des réseaux distribués pour la surveillance des structures en avionique, et la **surveillance des personnes âgées**, dans le domaine de la santé.



Figure 10 - L'axe ADREAM en illustrations

II. Présentation du Contexte de Stage

Nous faisons face actuellement à un **vieillesse démographique** général, entraînant une augmentation des handicaps et une hausse de personnes âgées vivant seules. C'est notamment le cas de 95% des plus de 75 ans en France par exemple.

Les **nouvelles technologies permettent aujourd'hui de proposer des solutions innovantes de suivi de santé à domicile (domotique médicale)**. Le groupe N2IS du LAAS-CNRS travaille sur une solution innovante de suivi de la marche dont le but est d'améliorer la qualité de vie des seniors, à travers une semelle intelligente et connectée.

L'âge, les pathologies, l'environnement et le social sont des facteurs qui peuvent entraîner une **vulnérabilité** des personnes âgées. Lorsque ces facteurs se dégradent, la sensibilité aux situations de stress s'en suit, ce qui peut entraîner une **perte d'autonomie** de la personne (syndrome de la fragilité). La fragilité est liée à un risque élevée de chute, de perte d'autonomie, d'hospitalisation et décès.

Le syndrome de la fragilité est défini par les **critères de Fried** :

- **Faible vitesse de marche** : comme nous pouvons le voir sur la figure 11, la fragilité est liée avec une vitesse moyenne inférieure à 1m/s,
- **Diminution de l'activité physique**
- **Perte de poids involontaire**
- **Épuisement, faible endurance, manque d'énergie, baisse de la dépense calorique hebdomadaire**
- **Faible force de préhension**

La présence d'un ou deux critères, définit une personne **pré-fragiles**, 3 critères ou plus correspondent à un état de **fragilité**.



Figure 11 - Les vitesses de marche selon les critères de Fried



En France, 50% des plus de 65ans sont pré-fragiles, et 15% sont considérés fragiles.

La fragilité est définie comme un état de pré-dépendance mais elle **réversible à l'aide généralement d'un programme de remise en forme basé sur la marche**. L'objectif de la semelle instrumentée est de proposer un système de suivi de la marche dont les données d'activité seront accessibles par le médecin et la personne à travers une application Web. L'objectif est de créer un système de prévention, de renforcement de l'autonomie, et d'encouragement à l'activité, de « **coaching motivationnel** ».

La semelle devra relever les paramètres suivants : la vitesse et la cadence de marche, le nombre de pas, la distance parcourue, les périodes d'activité, ainsi que la **variation du poids**.

Les principales **contraintes** de conception sont la **simplicité de mise en œuvre**, la **robustesse**, la **fiabilité**. Nous souhaitons créer un système discret, confortable, et **bon marché**. Aussi, ce système doit être connecté et intelligent pour permettre un suivi de santé distant et faire de la prévention.

III. Finalités et But de ma mission

Mon travail de stage se situe dans le prolongement du travail de thèse de Yoann Charlon axé sur la conception et le développement d'une semelle intelligente. Actuellement, une mesure de l'activité de marche a été programmée dans le microprocesseur, mais la fonction de mesure du poids embarqué est toujours absente. L'**objectif principal** de ce stage est donc de participer au **développement électronique du système embarqué** en ajoutant une **fonction « variation de poids »**, avec une caractérisation en situation réelle d'utilisation.

On peut séparer les tâches à effectuer en deux parties :

Partie Recherche

➤ **État de l'art**

Sur des systèmes existants récents de mesure de l'actimétrie et de la pression plantaire, sur les méthodes de classifications, sur les compositions des semelles.

➤ **Choisir une classification adaptée**

En effet, nous devons réaliser une classification, afin de détecter les conditions : marche à vitesse normale sur du plat. Cela permettra ensuite de mesurer les pressions plantaires dans des conditions à chaque fois identiques afin d'évaluer la variation de poids.

➤ **Mettre en œuvre cette classification avec MatLab**

➤ **Caractériser une fonction « variation de poids »**

➤ **Justifier le choix d'un ou deux capteurs de pression**

On souhaite en effet limiter le nombre de capteurs et de fonctions dans la semelle, car chaque ajout correspond à une augmentation du coût et une diminution de la place.

➤ **Modéliser des chutes**

Je dois comparer les chutes, vers l'avant, l'arrière, la gauche, la droite, afin de déterminer si leur détection est réalisable.

Partie Développement

➤ **Prendre en main la carte de développement conçue par le laboratoire**

➤ **Programmer un premier code en langage C – Récupérer les données en pression et accélération via USB pour traitement avec MatLab**

Afin de réaliser une base de données, concernant les variations de poids – marche sur tapis roulant avec des poids dans mon sac à dos, et aidant à valider les classifications. On veut retracer des courbes de marche plus précises.

➤ **Développer sous MatLab les fonctions « variation de poids » et « compteur de pas »**

➤ **Programmer un code de « variation de poids »**

Il faut alors prendre en main l'accéléromètre et les capteurs de pression.

➤ **Améliorer l'existant quant à l'actimétrie**

Les codes existants étant peu précis et fonctions de seuils numériques, j'ai décidé d'améliorer ce code et de le remplacer par un nouveau que j'ai conçu. Il faut évidemment optimiser la place et la consommation énergétique de la carte de développement.

➤ **Réaliser les tests en embarqué**

➤ **Valider des solutions**

Le matériel à ma disposition (figure 12) est une carte de développement, différents types de semelles, une tablette, et MatLab et ses toolboxes.

La carte de développement utilisée a été conçue par le LAAS-CNRS dans le but de mettre au point l'électronique de la semelle intelligente ainsi que de tester les algorithmes à implémenter. Le microprocesseur utilisé est le nRF51822, ce SoC – *System on Chip*, comprend un émetteur-récepteur basse consommation à 2.4 GHz. L'accéléromètre choisi est l'ADXL362, un nano accéléromètre 3 axes haute performance et ultra basse consommation. Les semelles, étant utilisées avec la carte de développement possèdent deux capteurs de pression, à l'avant du pied au niveau du métatarse, et l'autre au niveau du talon.

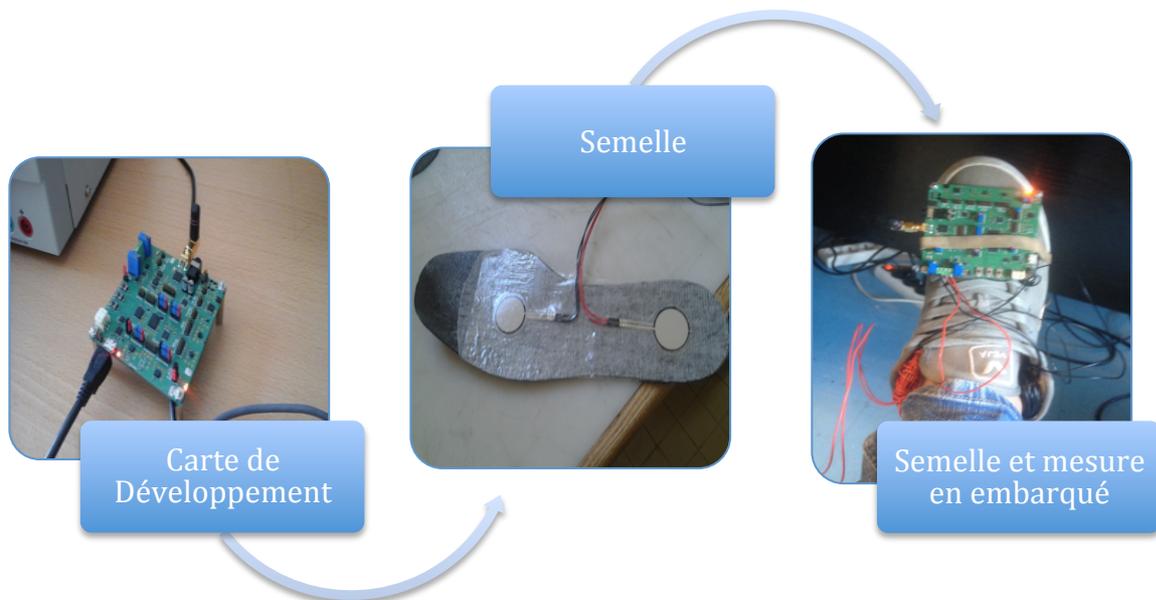


Figure 12 - Matériel à ma disposition

Déroulement de l'étude

Mon stage a consisté en trois étapes distinctes. Premièrement, l'étude comparative des différentes méthodes de classification disponibles, puis la mise en œuvre de la méthode choisie dans la carte de développement. Ensuite, les méthodes de classification devant aider à déterminer lors de quelle phase relever les variations de pression, la seconde étape est donc de développer un algorithme de mesure des variations de poids. Enfin, j'ai aussi développé un algorithme, de comptage de pas, afin d'améliorer le code existant.

I. Mise à jour du comparatif des méthodes de classification dans le cadre de la mesure du poids embarqué

Dans le cadre de la **mesure du poids embarqué**, nous souhaitons réaliser une **classification des activités** de l'utilisateur, puisque cette mesure, pour être fiable, doit être réalisée dans des conditions de marche classique : **marche sur du plat à vitesse normale**. En effet, celui-ci différera si nous décidons de relever le poids du patient lorsqu'il est en train de monter des escaliers, en train de courir, ou pendant la marche Il faut donc réaliser un algorithme basé sur la reconnaissance des activités : marche ou autre.

De nombreuses méthodes existent en matière de classification. Est-il plus judicieux d'utiliser des méthodes génératives ou discriminantes ? Quelle est la différence entre ces deux types ? Les paramètres à prendre en compte lors de cette étude comparative seront idéalement la simplicité de mise en œuvre de la méthode et son efficacité – dans ce genre d'études, les 99% de fiabilité sont visés^[9]. Sa **reproductibilité** est par ailleurs primordiale, puisque **l'algorithme sera utilisé plusieurs fois par jour**, afin d'avoir une **mesure la plus précise possible**. Il faut également **limiter la consommation énergétique** du système et donc ne pas utiliser une méthode trop gourmande en ressource informatique.

L'utilisation d'une méthode supervisée est justifiée par le fait que les modélisations des classifications sont réalisées sur une première base de données, pour ensuite être reproduites en embarqué. On souhaite définir une méthode appropriée sur ordinateur afin de l'adapter ensuite aux données réelles dans le microprocesseur.

Dans cette étude, nous n'évaluerons que des **méthodes supervisées**, dont le rôle est « prédictif ». En effet, à partir d'une première plage de données, d'apprentissage, nous pourrons alors évaluer la distribution d'une classe sans la mesurer directement, mais en se basant sur des valeurs lui étant liées. L'objectif sera alors de **minimiser l'erreur de prédiction**. L'approche discriminative modélisera directement une règle de classification $P(Y|X)$, Y étant la sortie et X l'entrée (ou prédicteurs, attributs), alors que l'approche générative sera une recherche de la modélisation de la distribution jointe $P(X, Y)$. La déduction de la règle de classification s'en déduit ensuite.

1. Méthode discriminative ou générative ?

Dans cette partie, nous allons décrire les avantages de chaque méthode, pour déterminer quelle sera la plus efficace lors de la modélisation des activités. Comme le

montre la figure 13, de nombreuses méthodes existent pour classifier correctement. Rappelons que l'objectif en classification est la minimisation du coût logistique, qui correspond aussi à la maximisation de vraisemblance conditionnelle.

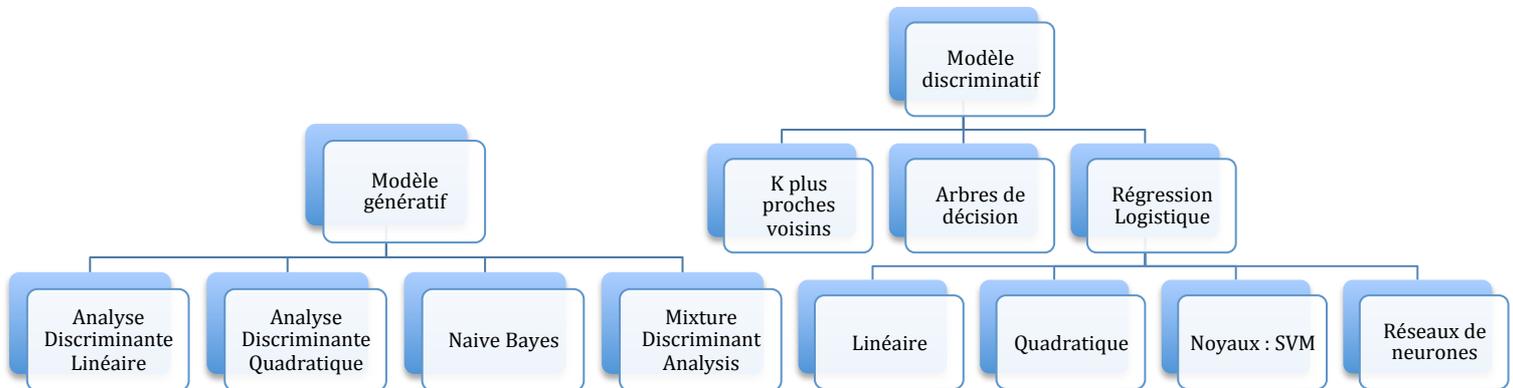


Figure 13 - Les différents types de classifications supervisées

Premièrement, intéressons nous aux régressions logistiques, représentant la majorité des modèles discriminatifs. Pour obtenir une règle de décision non-linéaire avec cette méthode, les variables X sont transformées en $\tilde{X} = \varphi(X)$, à partir desquelles une méthode de classification linéaire est appliquée. Pour un réseau de neurones par exemple, φ sera un ensemble de fonctions logistiques appliquées aux entrées, dont les poids seront obtenus par minimisation d'une des fonctions de coût. Les intérêts de ces méthodes sont leur efficacité lorsque X a une distribution inconnue et difficilement modélisable. Elles fournissent de bons résultats pour des classifications de textes ou des catégorisations d'objets, néanmoins les modèles génératifs ont aussi des potentiels discriminatifs non-négligeables, avec de vraies modélisations probabilistes.

La majorité des modèles génératifs peuvent être sous forme graphique, en alliant les variables par graphe de dépendance, avec un sens probabiliste des arcs. La modélisation générative permet de trouver une structure pour la distribution jointe des entrées X et sorties Y . Un modèle génératif est par nature plus contraint qu'un modèle conditionnel, puisqu'en approche discriminative, les hypothèses portent uniquement sur les $P(Y|X)$, et dans le cas génératif, les hypothèses portent sur la distribution de toutes les entrées (figure 14).

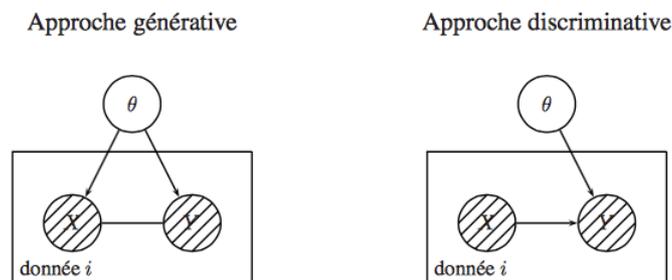


Figure 14 - Schématisation des deux approches

Un autre avantage de l'estimation générative est que les paramètres des classes peuvent être estimés de manière indépendante, ce qui simplifie le problème de maximisation. Ces méthodes sont souples, simples et parcimonieuses. Le taux d'erreur est généralement plus faible avec une estimation générative. Néanmoins, il est préférable de le vérifier par

soi-même plutôt qu'en utilisant des cas généraux. C'est pourquoi nous allons, dans la partie suivante, comparer les différentes méthodes à notre disposition (figure 15).

Dans [10], un exemple de classification de séries temporelles est étudié. Cet exemple est intéressant puisque dans le cadre de la mesure du poids, on souhaite caractériser les mouvements des patients lorsqu'ils sont en activité, c'est-à-dire au cours du temps.

Une application « naïve » de l'approche discriminative – qui définit une distribution des sorties conditionnellement aux entrées et paramètres, serait d'appliquer une méthode de classification quelconque sur X . On ne prendrait alors pas en compte l'aspect temporel. Avec une approche générative, on tentera de trouver un modèle adapté à la série $\{X_t\}_t$, pour chaque classe possible. On en déduira ensuite le classifieur. Les résultats seront bien meilleurs.

Par définition, une méthode d'apprentissage génératif est définie par une phase d'apprentissage, consistant à rechercher un estimateur $\hat{\theta}$ de θ , à partir des données d'apprentissage, puis une phase de test. La variable à prédire Y est discrète et définit l'index de la classe. A partir d'une donnée X , la valeur y est prédite avec une probabilité :

$$p(Y = k | X = x ; \hat{\theta}) = \frac{\pi_k f_k(x; \hat{\theta})}{\sum_{l=1}^K \pi_l f_l(x; \hat{\theta})}$$

pour $k=1..K$, $\pi_k = P(Y=k)$, f_k la densité de la k -ième classe.

La notion d'apprentissage correspond à l'estimation du paramètre θ , pour satisfaire au mieux les objectifs du modélisateur, c'est-à-dire approche par $P(Y|X ; \theta)$ la distribution conditionnelle $P(Y|X)$. L'estimateur est la solution du problème d'optimisation, le maximum de vraisemblance.

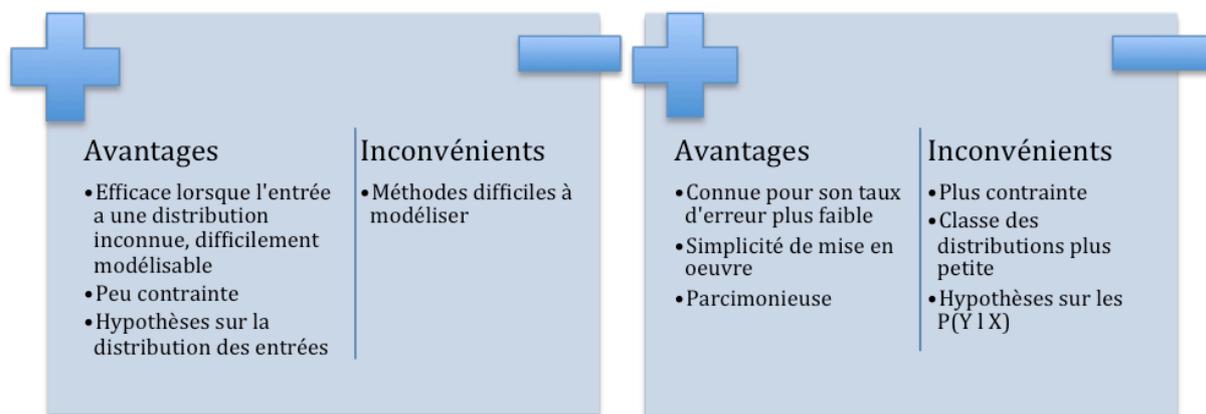


Figure 15 - Avantages et Inconvénients des méthodes discriminative (à gauche) et générative (à droite)

2. Comparaison des méthodes existantes, en fonction de la précision des résultats

La classification et distinction des activités ont été les premières tâches que j'ai réalisées durant ce stage. J'ai utilisé une base de données qui avait été mise au point par un précédent stagiaire avec la carte de développement.

La mise en données et les méthodes à modéliser sous MatLab

Dans chaque classification, la mise en données se fait selon deux phases, celle d'apprentissage, et celle de test. Dans notre modélisation, chacune de ces phases était séparée en trois sous-parties - marche, course, escalier, elles-mêmes séparées en trois sous-parties : durées des cycles d'activités, maxima, aires. Nous avons alors un vecteur X d'apprentissage, un vecteur Y représentant les classes, et un vecteur Z de test (figure 16).

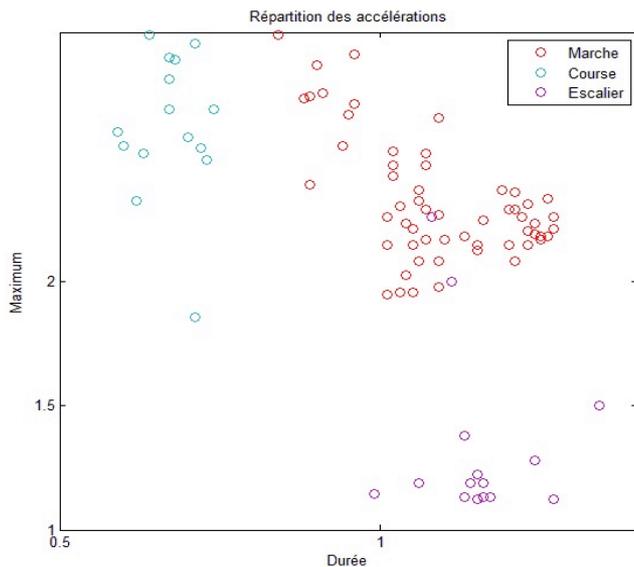


Figure 16 - Répartition des Données d'accélération en phase d'Apprentissage

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$$

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix}$$

$$Course = \begin{cases} Lente : 25m - 10km/h \\ Normale : 25m - 15 km/h \\ Rapide : 25m - 20km/h \end{cases}$$

$$Escalier = \begin{cases} Descente : 4 étages \\ Montée : 4 étages \end{cases}$$

$$Marche = \begin{cases} Lente : 3km/h \\ Normale : 5km/h \\ Rapide : 7km/h \\ Montée : 100m - 5km/h \\ Descente : 100m - 5km/h \end{cases}$$

$$X_i = [Durée Acc_i \quad Max Acc_i \quad Aire Acc_i \quad Durée Press_i \quad Max Press_i \quad Aire Press_i]$$

Enfin, chaque indice correspond à une activité différente. Le premier est la marche, le second la course, et le troisième les escaliers.

Présentation des méthodes utilisées

a. Analyse Discriminante Linéaire – LDA

Avec une méthode générative comme la LDA, pour conserver la paramétrisation générative, on estimera les moyennes et les variances, de manière discriminative. La solution ne sera pas unique, mais toutes les solutions maximiseront la vraisemblance conditionnelle, et donneront la même règle de classification.

Objectif : Pour chaque classe à identifier, on calcule une fonction linéaire des attributs. A partir des données d'apprentissage, on retourne une analyse de classification discriminante, puis on prédit les classes de Z, c'est-à-dire de la phase de test. Puisque nous savons quel résultat nous souhaitons, il est ensuite facile de déterminer l'erreur, à partir des prédictes.

Avec la méthode LDA, la fonction de la classe qui donne le score le plus élevé représente la classe prédite, c'est-à-dire : Si $P(i, j) == \max(P(i, :)) \Rightarrow label(i) = j-1$.

Exemple :

$$P(i,j) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0,005 & 0,995 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0,25 & 0,75 \end{pmatrix} \rightarrow label(i) = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 2 \\ 2 \end{pmatrix}$$

Une qualité de LDA est qu'elle ne nécessite pas de multiples passages sur les données pour optimiser les solutions. Elle gère naturellement avec plus de deux classes et peut fournir des estimations de probabilité pour chacune des catégories candidates. Elle trouve une transformation linéaire, dite « discriminante », des deux prédicteurs X et Y, donnant un nouvel ensemble de valeurs transformées plus précis qu'avec un seul prédicteur. On a alors une solution de la forme $Z = C_1X + C_2Y$.

La LDA permet de trouver le meilleur espace pour classer les données (figure 17). Cela peut être comparé à une diagonalisation de matrice. En effet, on cherche le meilleur espace afin de réaliser la meilleure projection possible. En algèbre linéaire, cela serait l'équivalent de la base des vecteurs propres, dans laquelle la matrice sera alors facile à manipuler. Les valeurs propres de la matrice $P(i, j)$ ci-dessus seraient alors de 0, 1 et 2, dans notre cas d'étude.

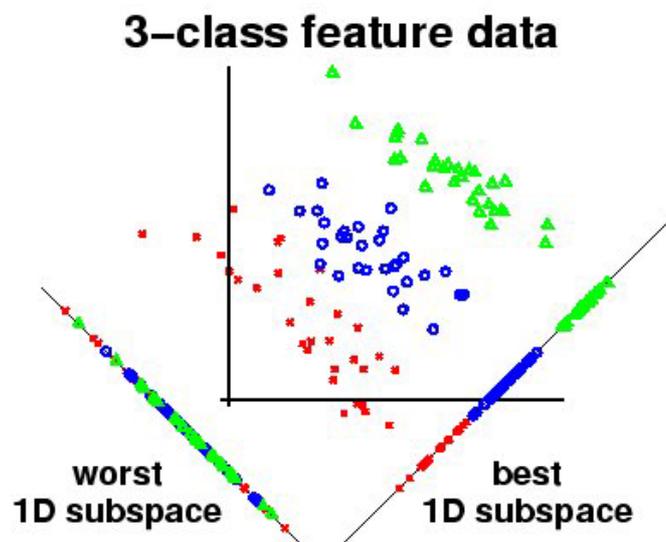


Figure 17 - Illustration du fonctionnement de LDA

b. Analyse Discriminante Quadratique – QDA

Evidemment, le fonctionnement de la méthode QDA (figure 19) est strictement le même que pour la LDA (figure 18), à la différence de la fonction des attributs, qui n'est donc plus linéaire, mais quadratique.

On obtient alors une modélisation plus précise, puisque nous utilisons alors un polynôme de second degré pour déterminer Z en fonction des prédicteurs.

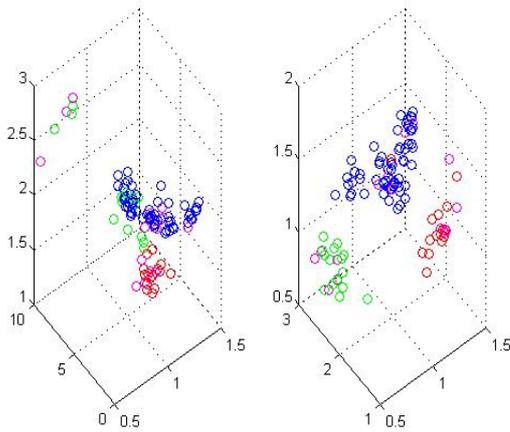


Figure 18 - LDA

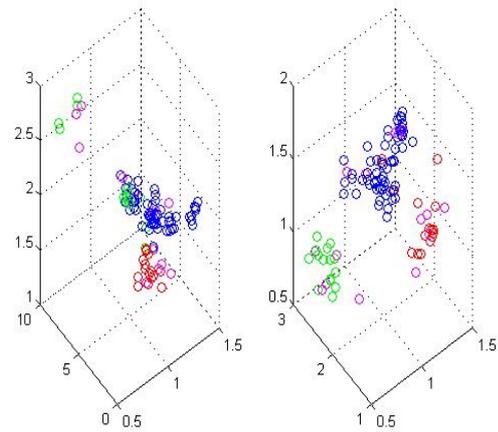


Figure 19 - QDA

c. Analyse Discriminante par Mélange – MDA

De même que précédemment, nous obtenons Z avec une fonction des attributs. Cette fois-ci, nous n'avons plus de fonction linéaire ou quadratique, on modélisera les densités des groupes $f_k(x) = p(X = x|Y = k)$ par un mélange de gaussiennes multivariées. Un logiciel – Mixmod, a été développé par le CNRS de Franche-Comté, modélisant la MDA. Il propose différents critères afin de regrouper les classes, ainsi que différents algorithmes, qui peuvent être combinés afin d'obtenir des résultats optimaux (figures 20 à 22).

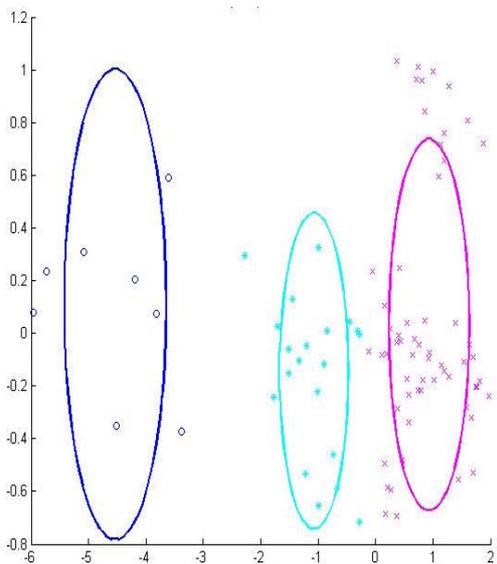


Figure 20 - Critère ICL, Integrated Completed Likelihood

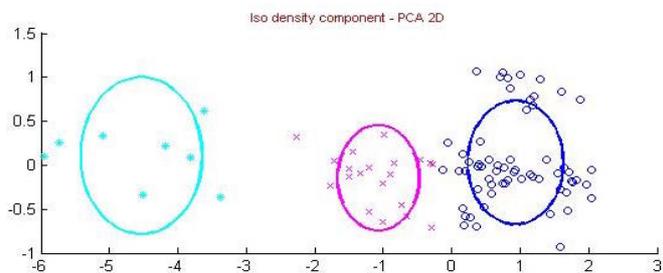


Figure 21 - Critère BIC, Bayesian Inferior Criterion

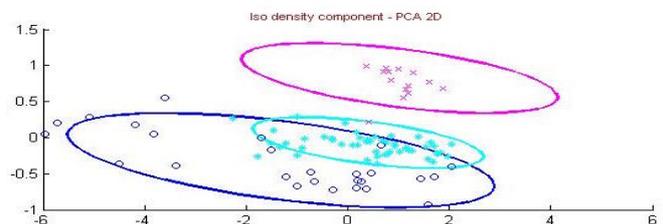


Figure 22 - Critère NEC - Critère d'entropie

Dans [10], il est démontré que MDA est meilleur que LDA, et capable de trouver une frontière de classification très proche de la frontière optimale de Bayes - que nous verrons dans la Classification Naïve Bayésienne ci-après. En effet, lorsque LDA donne de bonnes performances, les taux de classification de MDA sont comparables, mais lorsque

LDA est inefficace, typiquement pour des problèmes de classification non-linéaires, MDA permet une amélioration considérable des performances. En très grande dimension, MDA a des performances limitées, car les données se trouvent généralement dans un sous-espace de dimension réduite.

d. Classification Naïve Bayésienne

La classification naïve bayésienne est un type de classification bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance - dite naïve, des hypothèses. Elle met en œuvre un classifieur bayésien naïf, appartenant à la famille des classifieurs linéaires. En termes simples, un classifieur bayésien naïf suppose que l'existence d'une caractéristique pour une classe est indépendante de l'existence d'autres caractéristiques. Le modèle probabiliste pour un classifieur est le modèle conditionnel $P(C|F_1, \dots, F_n)$ où C est une variable de classe dépendante dont les instances ou classes sont peu nombreuses, conditionnée par plusieurs variables caractéristiques F_1, \dots, F_n .

$$\text{classifieur}(f_1, \dots, f_n) = \underset{c}{\operatorname{argmax}} P(C = c) \prod_{i=1}^n P(F_i = f_i | C = c)$$

Les modèles probabilistes ainsi décrits sont plus faciles à manipuler, puisqu'ils peuvent être factorisés par l'antérieure probabilité à priori de C , $P(C)$, et les lois de probabilités indépendantes $P(F_i|C)$. S'il existe k classes pour C et si le modèle pour chaque fonction $P(F_i | C = c)$ peut être exprimé selon r paramètres, alors le modèle bayésien naïf correspondant dépend de $(k - 1) + nrk$ paramètres.

Sur MatLab, la classification naïve bayésienne assigne une nouvelle observation de la classe la plus probable, en supposant que les fonctions sont conditionnellement indépendantes. **La fonction fitNaiveBayes s'adapte aux données de l'apprentissage, en associant à chaque distribution un vecteur contenant sa moyenne et son écart-type.** Elle calcule ensuite la probabilité des ordres des classes à postériori, pour les données du test. Dans les graphiques 23 et 24, les contours de Gauss sont aussi tracés. Nous pouvons alors faire le lien avec la MDA utilisée précédemment, qui utilise des mélanges de Gaussiennes.

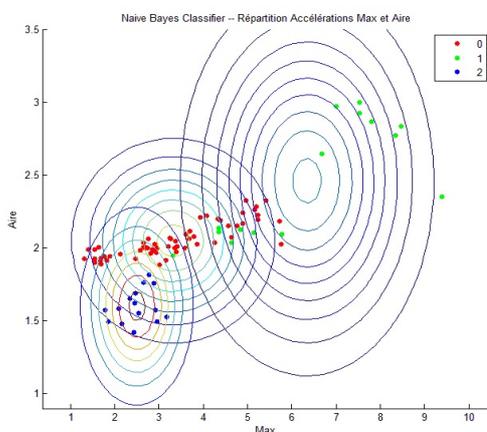


Figure 23 - Classification Naive Bayésienne, Données d'accélération

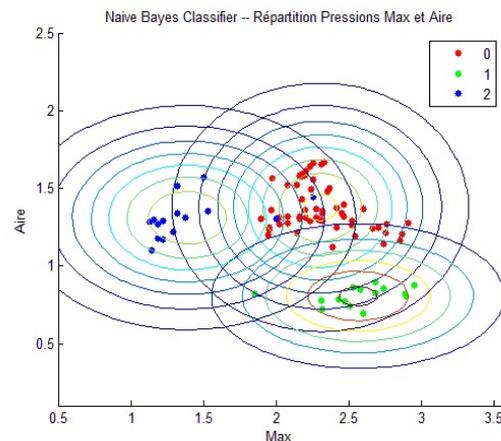


Figure 24 - Classification Naive Bayésienne, Données de pression

e. Séparateurs à Vastes Marges, Machines à Vecteurs de Support – SVM

Objectif : Le SVM va changer de base, pour trouver un nouveau repère dans lequel la « frontière séparatrice » est optimale, à partir des données d'apprentissage. Dans ce contexte, une marge est la distance entre la frontière de séparation et les échantillons les plus proches, aussi appelés vecteurs support. La frontière séparatrice est la frontière qui maximise la marge (figure 25). On peut rejeter les points trop proches des frontières pour obtenir une meilleure précision. Le classifieur permet de résoudre ce problème comme un problème d'optimisation quadratique.

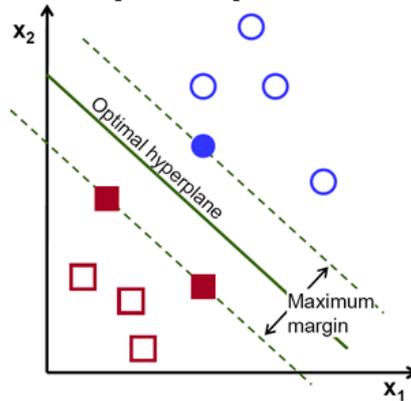


Figure 25 - Les Marges et la Frontière Séparatrice

Si les données ne sont pas linéairement séparables, on transforme alors l'espace de représentation des données d'entrées en un espace de plus grande dimension, dans lequel il existe une séparatrice linéaire. L'avantage est que cette technique, dite de « kernel trick », ne nécessite pas le calcul de la transformation, et qu'elle respecte la condition du théorème de Mercer : « Une fonction à noyau continue, symétrique, semi-définie positive, peut s'exprimer comme un produit scalaire dans un espace de plus grande dimension » (figure 26).

En considérant l'analogie avec l'algèbre linéaire, que j'avais précédemment réalisée, nous pouvons dire que cette technique permet, en changeant de base, de trouver une matrice diagonale, et une fois le noyau et les valeurs propres trouvés, nous n'avons pas besoin de connaître la matrice de changement de repère pour effectuer nos manipulations.

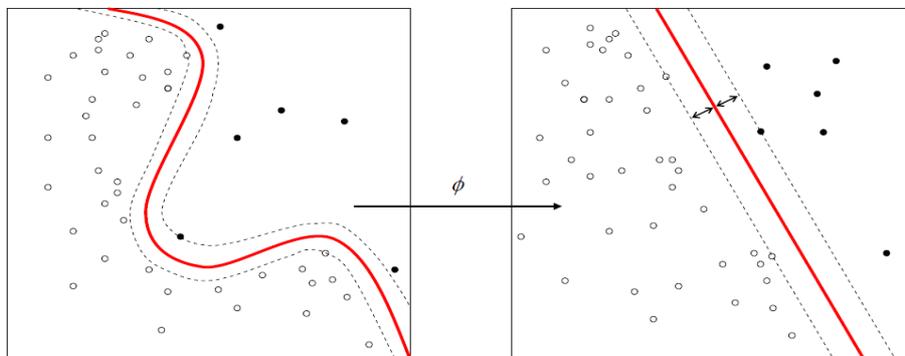


Figure 26 - La transformation de l'espace de représentation

Pour la classification, le système [8] utilise un SVM, basée sur deux scalaires représentant la pression et l'inertie, et une séquence temporelle de scalaires représentant ces deux mêmes caractéristiques. Les deux premiers scalaires évoqués

peuvent être obtenus en filtrant les données mesurées ou en les combinant sur le temps, par exemple dans un filtre moyenneur. Les activités sont classées comme « assis », « debout », ou « marchant ». Elles peuvent aussi comprendre la course ou d'autres activités sportives. La marche correspond à une grande pression moyenne et une grande magnitude d'accélération. Rester debout correspond à une grande pression moyenne avec une faible magnitude d'accélération. Enfin, être assis correspond à une faible ou moyenne pression, avec une faible magnitude d'accélération. En fonction du lieu, la pression moyenne peut être mesurée pour les deux pieds ou un seul.

f. Les réseaux de neurones

Les réseaux de neurones sont une méthode de classification très efficace. Ils sont utilisés dans la SmartShoe par exemple [9]. La méthode de classification utilisée dans cet article rejette les points étant trop près des limites de décision du classifieur, jugés incertains. Théoriquement, sur 1000 points, 200 pourront être écartés, et les 800 restants auront une probabilité de 0,99 d'être placés dans la bonne classe. Un epoch est un segment de 2 secondes de test dans cet article, c'est un échantillon.

La SmartShoe considérée dans cette étude contient six capteurs dans chaque chaussure, dont cinq de pression dans la semelle et un accéléromètre au niveau du talon (figure 27).

Elle étudie différents paramètres : les moyennes, les minima, les écart-types, les entropies, les variances, les maximums, le nombre de passages à la moyenne – *NMC number of mean crossings*, les écarts absolus moyens et le ratio entre les racines carrées des coefficients d'ondelettes dans les directions antérieure/postérieure et dans la direction verticale et le ratio des maximums des accélérations et des pressions, ainsi que la valeur de la pression au moment de l'accélération maximale.

Avec un MLP, le rejet de certaines valeurs est basé sur l'estimation de la fiabilité de la classification. Idéalement, seul le neurone associé à la classe à laquelle appartient un epoch donné aura la valeur 1, alors que les autres vaudront 0.

Changer les seuils de rejet modifie le nombre d'epochs rejetés, et la précision de la classification. Le MLP après rejet permet de dépasser les 99% de précision. Avec des seuils à 0,95, plus de 50% des epochs ont été rejetés.

Cette méthode peut être appréciable quant à l'étude de différentes postures et activités. Le rejet de certains epochs peut avantager l'étude posturale de patients atteints de Parkinson par exemple, dont certains mouvements n'ont pas besoin d'être classés. La méthode de rejet montre donc une potentielle amélioration dans la recherche biomédicale, où la précision est primordiale et où les échantillons sont nombreux. En dépit de son efficacité, il demeure difficile de réaliser un MLP en C : on cherche une méthode devant allier performance, coût, reproductibilité et simplicité de mise en œuvre. La SmartShoe combine plus de capteurs, de calculs lourds, et sera en conséquence plus difficile à mettre en œuvre mais aussi bien plus chère.

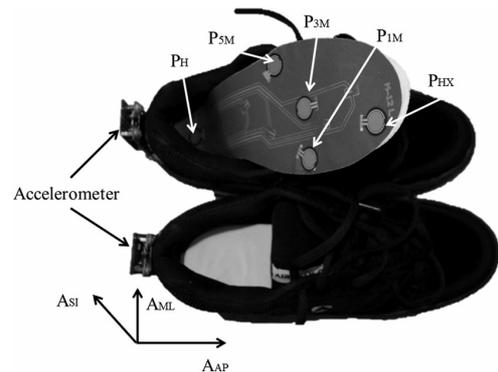


Figure 27 - Paire de chaussures SmartShoe équipées des capteurs

g. Calcul des distances pondérées

Pour chaque activité, on admet connaître l'écart-type et la moyenne. Pour chaque caractéristique de l'activité inconnue Z, on mesure la distance entre celle-ci et les caractéristiques correspondantes de chaque activité connue. On divise ensuite cette distance par le double de l'écart-type correspondant, pour **prendre en compte la dispersion**. Pour reconnaître l'activité, on prend la distance minimale, comme pour l'Analyse Discriminante Linéaire, mais avec la fonction « min » au lieu de « max ».

h. K-plus proches voisins

La méthode des K-plus proches voisins est une méthode d'apprentissage supervisée. Dans ce cadre, on dispose d'une base de données d'apprentissage constituée de N couples "entrées-sorties". Pour estimer la sortie associée à une nouvelle entrée X, la méthode des K-plus proches voisins consiste à prendre en compte, de façon identique, les k échantillons d'apprentissage dont l'entrée est la plus proche de X. Dans un problème de classification, on retiendra la classe la plus représentée parmi les k sorties associées aux k entrées les plus proches de la nouvelle entrée X.

i. Arbres de classification et de régression

Dans les structures d'arbre, **les feuilles représentent les valeurs de la variable cible et les embranchements correspondent à des combinaisons de variables d'entrée qui mènent à ces valeurs**. Le logiciel MatLab retourne un arbre de régression ou de classification, basé sur les prédicteurs et la réponse Y. On prédit ensuite les classes auxquelles appartiennent les données de Z.

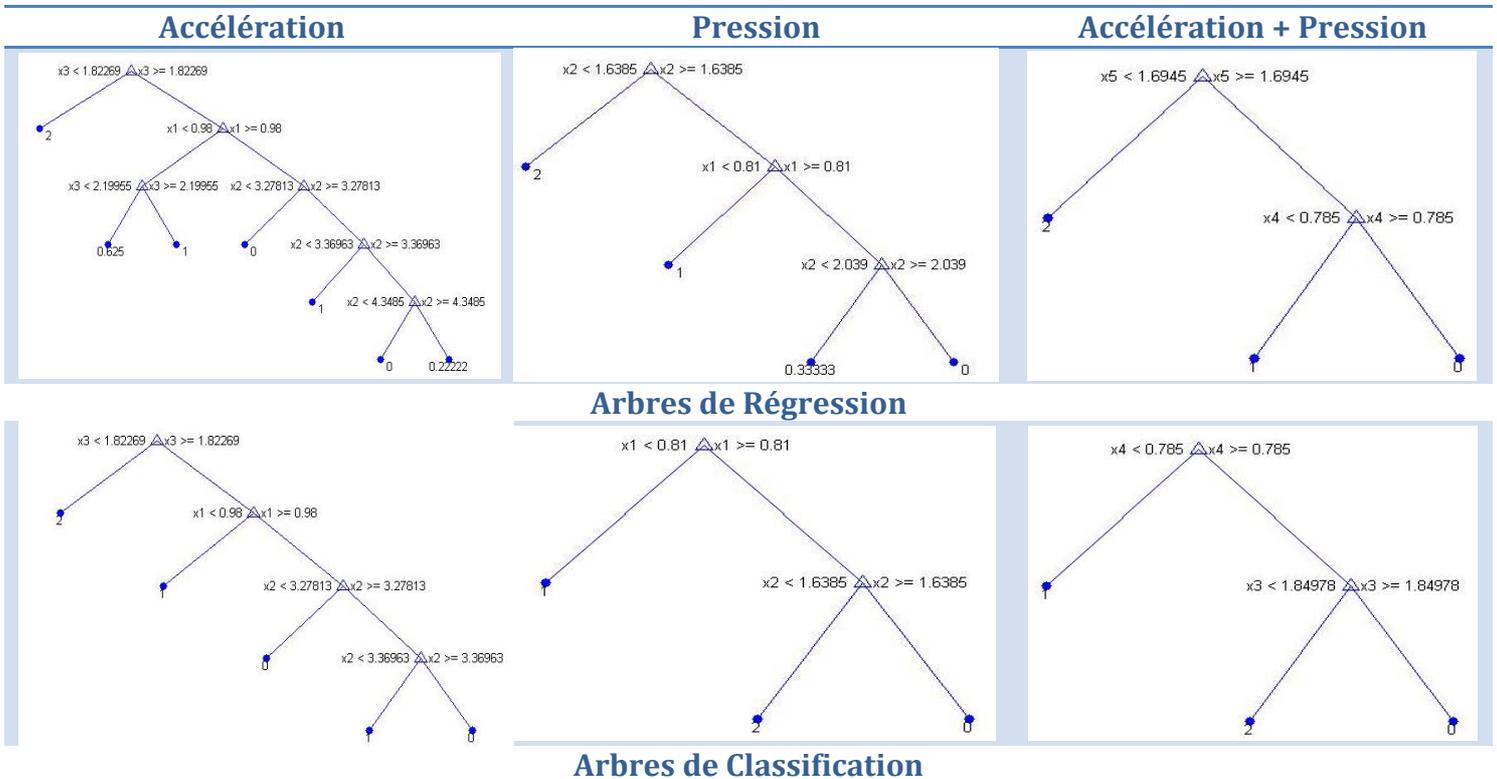


Figure 28 - Différents Arbres de Régression et Classification

Premiers résultats quant à la précision des différentes méthodes

Toutes les méthodes citées précédemment ont été testées sous MatLab, en recourant à un certain nombre d'essais consécutifs, afin d'obtenir un maximum de précision quant à leurs résultats. Le tableau 3 donne un comparatif des taux d'erreurs de chaque méthode, pour des données exclusivement en pression, en accélération, ou combinées.

Méthodes	Accélération + Pression	Accélération	Pression
LDA lin	5,53%	23,00%	3,19%
LDA quad	8,59%	23,55%	3,77%
LDA custom	9,31%	23,00%	3,19%
Naive Bayes	5,21%	21,04%	3,67%
SVM lin	16,05%	47,64%	8,48%
SVM quad	75,43%	50,53%	69,56%
SVM rbf	7,84%	23,49%	3,36%
Distances Pondérées	5,25%	15,99%	3,64%
Arbre Reg	7,72%	29,40%	4,83%
Arbre class	5,32%	17,43%	2,92%
K voisins	8,84%	21,89%	3,72%

Tableau 3 - Comparatif des taux d'erreurs de chaque méthode sous MatLab

Nous pouvons remarquer que certaines méthodes sont très efficaces, nous laissant avec environ 95% de précision. Aussi, nous pouvons remarquer que les méthodes les plus précises combinant accélération et pression, le sont aussi indépendamment.

D'après les premiers essais, la durée des cycles de marche, leur valeur maximale ainsi que leur aire sont de bonnes caractéristiques. Afin d'économiser de l'énergie au niveau du système embarqué, j'ai proposé une simplification au niveau du traitement des données. **Ayant remarqué la redondance du critère des écart-types et des moyennes**, dans des méthodes efficaces telles que la **classification naïve bayésienne**, ou la **méthode des distances pondérées**, j'ai décidé de calculer les écart-types et moyennes des différentes étapes de marche. J'ai ainsi proposé un **arbre de classification**, utilisant l'écart-type de l'amplitude des accélérations pour définir un type de marche et une vitesse.

3. Exploitation des résultats vers une nouvelle méthode phénoménologique efficace

Précédemment, nous avons pu remarquer la **précision** de chaque méthode, un critère important pour la sélection de cette dernière. Néanmoins, nous devons aussi prendre en compte la **simplicité de mise en œuvre** de la méthode, car celle-ci sera ensuite implémentée en C dans le microprocesseur. La méthode qui suit a été choisie dans cette optique. Elle utilisera les critères cités ci-dessus, tels que les écarts-types et les moyennes des amplitudes des accélérations de la marche.

Les caractéristiques utilisées pour l'acquisition des foulées et des cadences, sont l'amplitude de l'accélération, aussi appelée Vecteur Somme V_s , sa moyenne et sa variance.

- Calcul du **vecteur somme**, amplitude de l'accélération :

$$V_s = a_i = \sqrt{a_{xi}^2 + a_{yi}^2 + a_{zi}^2}$$

- Calcul de la **moyenne** sur une fenêtre glissante de $w = 25$ données, appliquée à l'amplitude de l'accélération a_q :

$$Moy = \bar{a}_j = \frac{1}{2w + 1} \sum_{q=i-w}^{i+w} a_q$$

- Calcul de la **variance sur une fenêtre glissante** :

$$Var = \sigma_{ai}^2 = \frac{1}{2w + 1} \sum_{j=i-w}^{i+w} (a_j - \bar{a}_j)^2 = \frac{1}{2w + 1} \sum_{j=i-w}^{i+w} (V_s - Moy)^2$$

Avec ces caractéristiques, j'ai réalisé une base de données afin d'identifier chaque phase de marche.

Nous pouvons observer dans le tableau 4 les différences de variance des amplitudes des accélérations, en fonction des phases de marche, puis des différentes vitesses de marche. Chaque variance est respectivement associée à une vitesse.

Classe	Lent (plat)	Normal (plat)	Rapide (plat)	Montée	Descente
Course	2,46	5,36	8,04		
Marche	0,05	0,26	1,06	1,01	2,09
Escalier				0,54	0,61

Tableau 4 - Variances en fonction des phases de marche : en montée, descente, et sur du plat à 3 vitesses différentes, en g

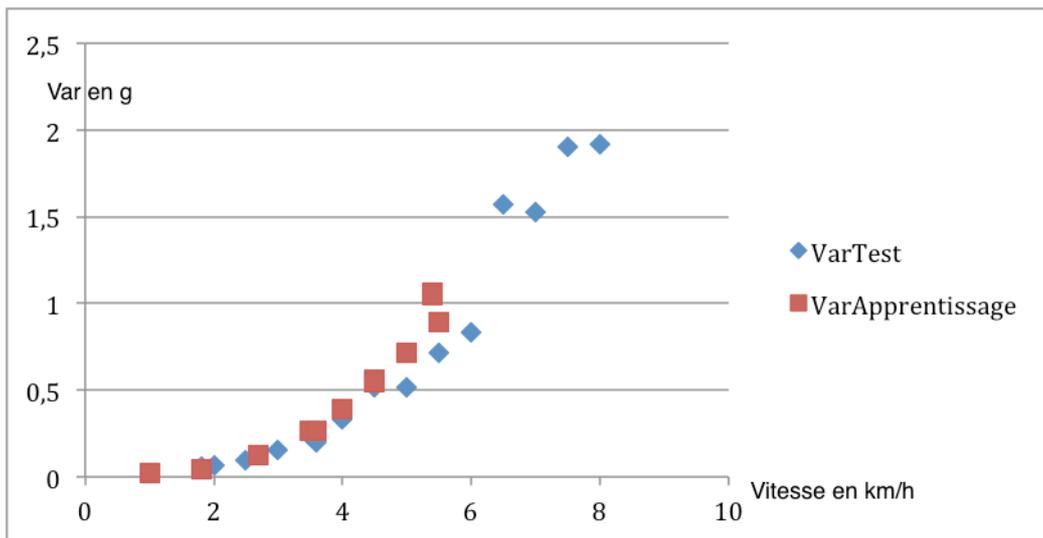


Figure 29 - Variances en fonction des vitesses de marche en km/h, en phases d'apprentissage et de test, en g

Nous pouvons remarquer, grâce à la figure 29 ci-dessus, que **la vitesse de marche est fortement corrélée avec la variance des amplitudes des accélérations**. Nous

pouvons ainsi définir un intervalle de Var pour la phase « marche normale », nous pourrions retrouver cette phase de marche dès que Var sera dans cet intervalle. De plus, grâce au tableau 4, nous remarquons que lorsque nous marchons en montée, en descente, voire dans des escaliers ou que nous courons, les valeurs sont très éloignées de la « marche à vitesse normale sur du plat ».

4. Optimisation du code embarqué

Afin d'alléger les calculs et donc d'**économiser l'énergie** embarquée de la semelle, mais aussi d'augmenter la plage de fonctionnement de la **méthode des « variances »**, j'ai proposé de relever le carré du Vecteur Somme, plutôt que sa définition habituelle. En effet, en langage C, la bibliothèque <math.h>, contenant notamment la fonction « racine carrée », utilisée par V_s prend beaucoup de place, et peut nécessiter beaucoup de temps de calcul.

Ainsi, mes calculs sont devenus :

- Calcul du **carré du vecteur somme**, carré de l'amplitude de l'accélération :

$$V_{s2} = a_{i2} = a_{xi}^2 + a_{yi}^2 + a_{zi}^2$$

- Calcul de la **moyenne** sur une fenêtre glissante de $w = 25$ données, **appliquée au carré de l'amplitude** de l'accélération a_q :

$$Moy2 = \bar{a}_{j2} = \frac{1}{2w + 1} \sum_{q=i-w}^{i+w} a_{q2}$$

- Calcul de la **variance des amplitudes au carré** sur une fenêtre glissante :

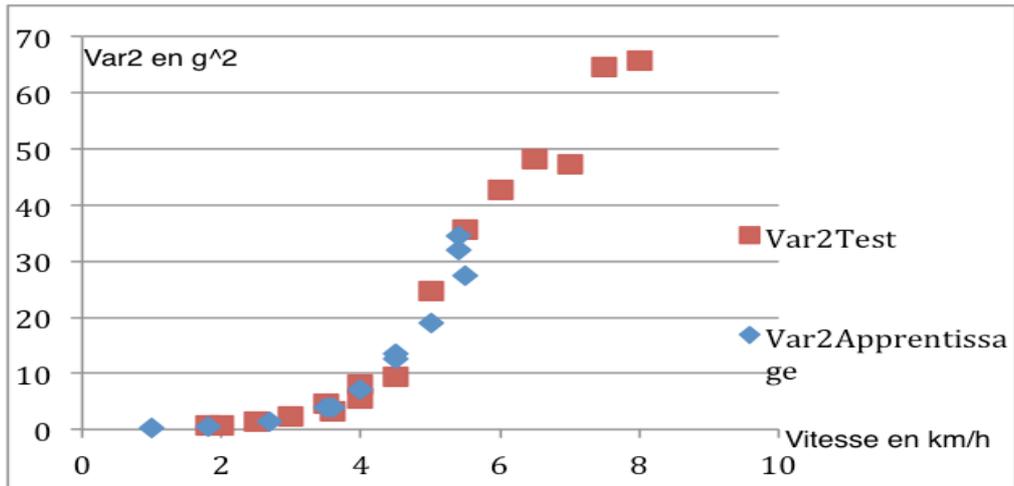
$$Var2 = \sigma_{ai2}^4 = \frac{1}{2w + 1} \sum_{j=i-w}^{i+w} (a_{j2} - \bar{a}_{j2})^2$$

$$\text{soit } Var2 = \frac{1}{2w + 1} \sum_{j=i-w}^{i+w} (V_{s2} - Moy2)^2 = \frac{1}{2w + 1} \sum_{j=i-w}^{i+w} (V_s^2 - Moy2)^2$$

Les variances obtenues après ces modifications sont présentées dans le tableau 5.

Classe	Lent (plat)	Normal (plat)	Rapide (plat)	Montée	Descente
Course	159	647	1043		
Marche	0,5	3,8	34,4	34,3	110
Escalier				22,4	18,1

Tableau 5 - Variances des amplitudes au carré en fonction des phases de marche : en montée, descente, sur du plat à 3 vitesses différentes, en g^2



Vitesse (km/h)	1,8	2	2,5	3	3,5	3,6	4	4	4,5	5	5,5	6	6,5	7	7,5	8
Var2 (g ²)	0,78	0,68	1,34	2,43	4,53	3,23	5,60	8,06	9,32	24,7	35,7	42,8	48,2	47,2	64,5	65,7

Figure 30 - Variances des carrés des amplitudes en fonction des vitesses de marche en km/h, en phases d'apprentissage et de test, en g²

Cette méthode est simple à mettre en œuvre, en effet il suffira de définir le calcul de Var2 sur une fenêtre glissante, et de demander l'acquisition des données de pression, concernant la variation de poids, lorsque Var2 sera compris dans un intervalle optimal (figure 30).

J'ai testé cette méthode sur tapis roulant, en faisant varier les vitesses de marche. Sur le graphique figure 31, nous pouvons remarquer les différences de Var2, représentée en bleu. J'ai marché à 4km/h, puis 6, puis 2, 8 et enfin je suis retournée en conditions normales, à 4km/h. Var2 varie considérablement.

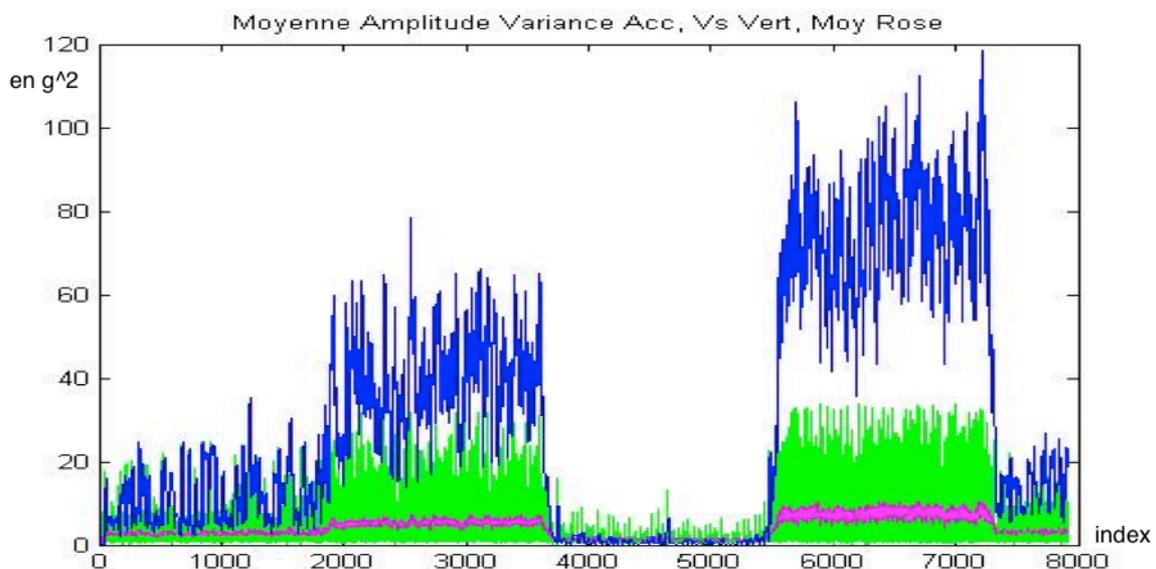


Figure 31 - Evolution des Var2 (Bleu), Moy2 (Rose) et Vs2 (Vert) en fonction des vitesses de marche (index 0 à 2000 : 4km/h, 2000 à 4000 : 6 km/h, 4000 à 5500 : 2km/h, 5500 à 7500 : 8 km/h, 7500 à 8000 : 4km/h) en g²

En annexe se trouvent différents graphiques concernant cette marche, justifiant notamment le choix de Var2 face à Var. Une partie du code est montrée dans la figure 32.

```

UINT32 M = m-25;
//Remise à 52 de m
if (awake) // Mise en marche de la carte - l'ADXL (accéléromètre, repère du mouvement)
{
    m++;
    if (m>250)
    {
        m=52;
    }

    Vs[m] = (acceleration.x*acceleration.x + acceleration.y*acceleration.y +acceleration.z*acceleration.z);
    // Définition du vecteur somme

    if (m > 51)
    {
        for (j=M; j<m; j++)
        {
            Moy[m] += Vs[j]/26; // Puis de la moyenne sur une fenetre glissante de 25 valeurs
        }

        for (i=M; i<m; i++)
        {
            Var2[m] += ((Vs[i] - Moy[i])*(Vs[i] - Moy[i]))/26624; // Puis du Var2, ATTENTION à penser à la conversion en g
            // 26624 = 26 (fenêtre glissante) * 32 * 32 (LSB)
        }
        if ((Var2[m] <=15) && (Var2[m] >= 3))
        {
            pressok = QuatreDGetCurrentPression(&pression); //fonction démarrant la prise de mesure en pression;
        }
    }
}
}

```

Figure 32 - Code de Var2, fonction getAcc_Moy

La fonction est appelée en continu dans `main.c`.

En conclusion de cette partie, cette méthode consiste en un arbre de classification. Les combinaisons contenues dans les embranchements correspondent aux carrés des vecteurs somme, de leur moyenne, et leur Var2. Elle allie **précision, simplicité de mise en œuvre, faible coût**, et faible **consommation énergétique**. Dès l'acquisition d'une semelle par une personne âgée, il leur est demandé de marcher à un rythme normal et rapide, afin de calibrer la semelle. **A l'aide de Var2 lors de cette étape de calibration, nous pourrions alors déterminer un intervalle de deux variances, avec plus ou moins 10% de marge par exemple**. Cet intervalle permettra la reconnaissance d'une activité de marche normale sur du plat, et par la suite la mesure du poids et sa variation.

II. Algorithmes

A présent, nous allons étudier dans cette seconde partie les différents algorithmes que j'ai développés, afin de mesurer la variation du poids ou le nombre de pas réalisés par une personne, en embarqué. Les algorithmes seront développés sous MatLab puis codés en C dans la carte de développement.

1. Mesure de la variation du poids embarqué

La perte involontaire de poids étant un des critères de Fried dans la mesure de la fragilité d'une personne âgée, nous voulons donc développer un algorithme efficace, répété plusieurs fois par jour – par souci de précision, mesurant la variation du poids. Seul le LAAS-CNRS s'est intéressé à ce critère, c'est donc une caractéristique innovante de notre semelle par rapport à l'existant.

L'objectif de ma mission sera de répondre à plusieurs questions : quand voulons nous relever le poids ? Quelle caractéristique de ce poids sera la plus significative ? Avons-nous besoin d'utiliser à la fois un capteur au niveau du talon et un capteur au niveau du métatarse, ou un seul d'entre eux suffira ?

Note : Ces deux positions (figure 33) ont été définies comme les deux points forts en pression lors de la marche. Aussi, la sensibilité étant très faible au niveau de l'intérieur de la voûte plantaire, c'est à cet endroit que, stratégiquement, nous positionnerons le module à l'avenir.
(source : www.novelusa.com)

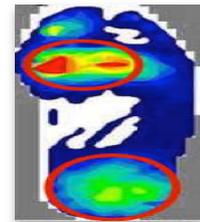


Figure 33 - Positions des capteurs de pression

Proposition

Dans un premier temps, j'ai relevé les pressions en continu pendant des périodes de marche stables, afin de trouver un critère facilement repérable, pertinent et représentatif de la pression appliquée aux différents capteurs.

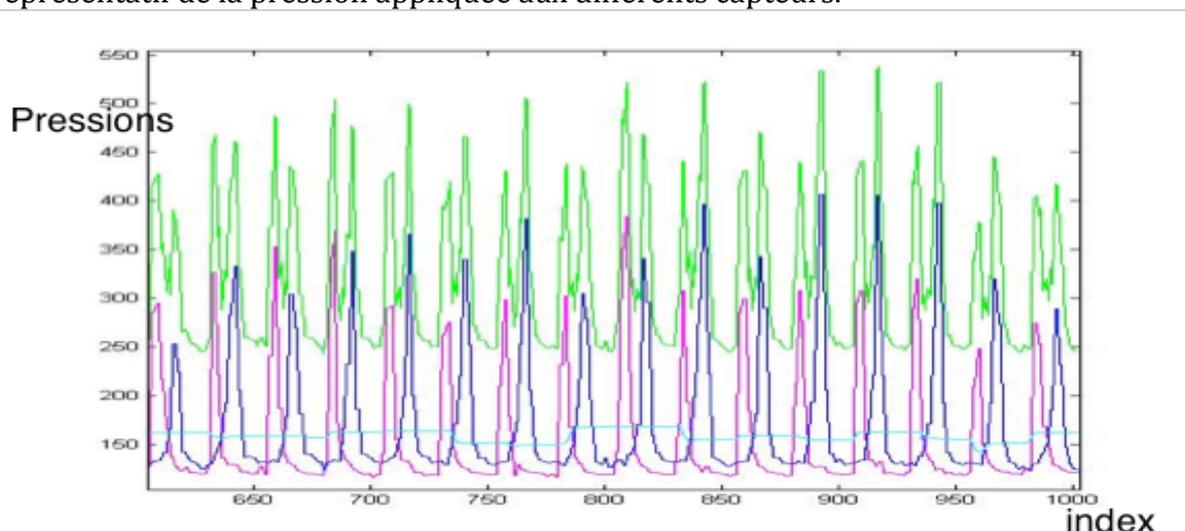


Figure 34 - Variation de la pression sur les capteurs, en continu, en fonction des index - pression au niveau du talon en rose, pression à l'avant en bleu, et somme des deux en vert, moyenne des maxima en cyan

Comme nous pouvons le remarquer sur la figure 34 représentant une marche en continu, sans variation de poids, les maxima respectifs à l'avant et à l'arrière sont en décalés, mais ont des valeurs assez proches. C'est pourquoi j'ai tracé les moyennes des maxima, calculées sur une fenêtre glissante de 25 données, afin de juger de leur stabilité. En relevant seulement les moyennes ou les moyennes des maxima sur un seul capteur, il était difficile de trouver un critère aussi stable, une courbe aussi monotone.

De plus, j'ai jugé qu'il était **rigoureux** de relever les **pressions en continu**, et d'en calculer les moyennes, car on pourrait facilement détecter une **variation de quelques pourcentages** de ces valeurs. En fin de journée, on réalise une moyenne de ces moyennes, ce qui permet de ne pas tenir compte de légères variations pouvant avoir lieu.

Évidemment, lorsque l'on évoque la notion de « pression en continu », c'est en considérant que nous sommes dans une marche normale sur du plat, c'est-à-dire avec un Var2 compris dans un certain intervalle défini au préalable.

La résultante des forces verticales sous la surface du pied varie au cours du temps et des étapes de la phase d'appui du cycle de marche comme le montre la figure 35 (a). L'amplitude de ces forces est en corrélation avec la vitesse de marche et la longueur de la foulée. Ainsi, la résultante des forces sous la surface du pied, équivaut au poids de la personne multiplié par un coefficient compris entre 0 et 1,25 au cours d'une marche normale (1,5 m/s). Le poids du corps peut être multiplié par un coefficient allant jusqu'à 2,5 pendant la course (6 m/s) comme le montre la figure 35 (b).

Ainsi, la vitesse de marche, la longueur de la foulée et les phases de la marche doivent être connues pour réaliser une mesure de poids réaliste en dynamique. La classification des activités, dépendant des vitesses et définissant les phases de marche, permet donc d'avoir une valeur corrélée avec le poids.

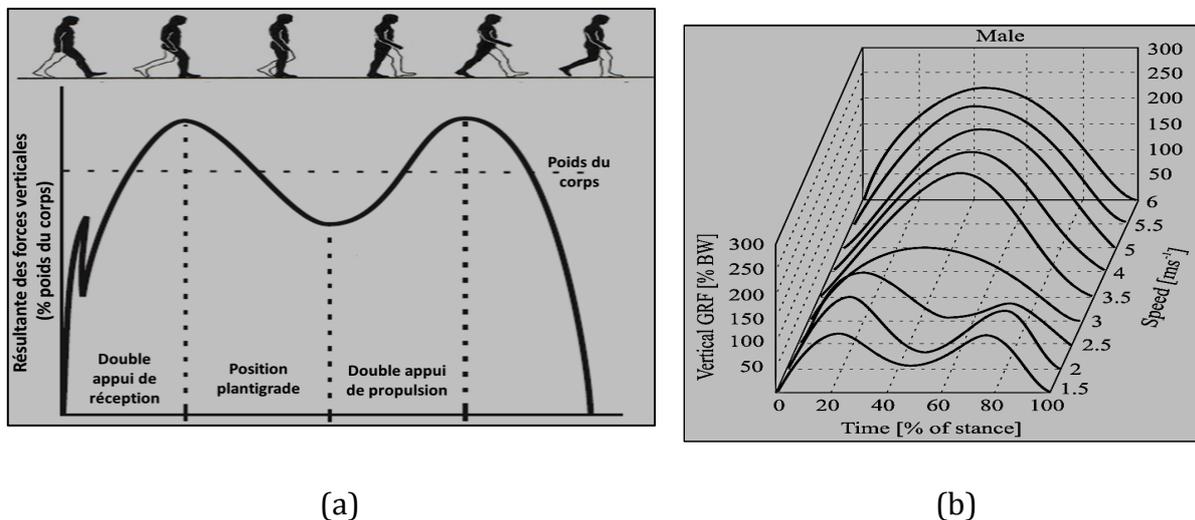


Figure 35 - Résultante des forces verticales en fonction des étapes de la phase d'appui (a) et résultante des forces verticales en fonction de la vitesse (b)

La valeur résultant des forces verticales sera enregistrée sur le serveur de l'application. Chaque jour, une valeur, moyenne des moyennes ou moyennes des maxima, sera retenue, et pourra être comparée aux jours suivants. Nous pourrions donc stocker des valeurs sur de longues durées, et repérer des variations de poids sur plusieurs semaines voire mois.

Tests

J'ai réalisé dans un premier temps une base de données avec différents poids portés dans un sac à dos. Je me suis mise dans des conditions de marche normale, sur le tapis roulant, à 4km/h.

Sur le graphique suivant (figure 36), nous pouvons remarquer les différences de pression lorsque nous ajoutons du poids à chaque arrêt. Premièrement, je ne porte rien, puis 2,6%, 4,5%, 5,4% et enfin 7,2% de mon poids – à l'aide d'haltères.

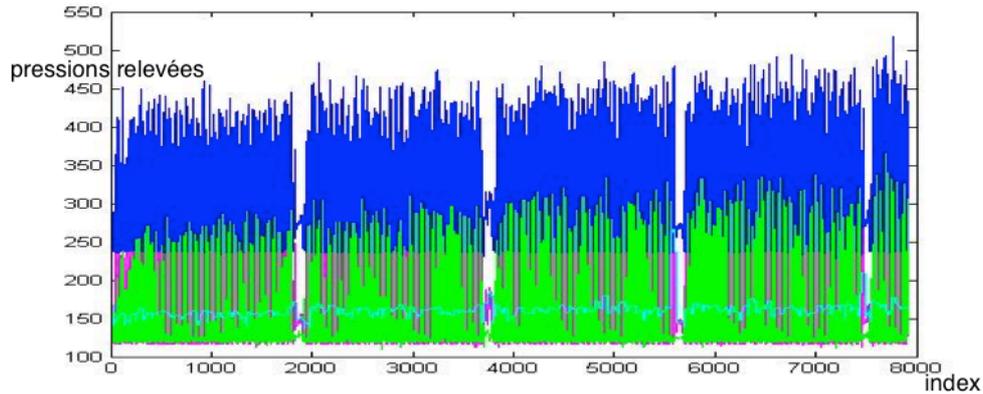


Figure 36 - Variation de la pression en fonction des poids appliqués

Le tableau 6 récapitule des variations de pression, en pourcentages, en fonction des poids que j'ai portés.

Index (abscisses de la figure)	Poids supplémentaire (porté)	Moyennes des maxima	Moyennes des moyennes
1 :1890	-	-	-
1890 :3750	+2,6%	+2,7%	+3,9%
3750 :5640	+4,5%	+4,4%	+5,5%
5640 :7470	+5,4%	+4,8%	+5,8%
7470 :7870	+7,2%	+7,3%	+7,0%

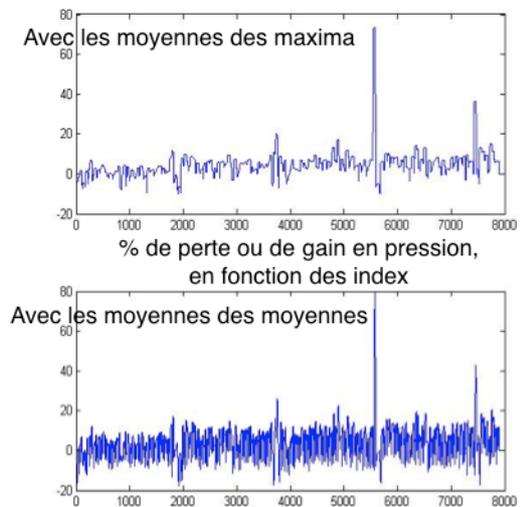


Tableau 6 - Evolution en pourcentages des pressions moyennes et de leurs maxima en fonction des poids appliqués – Moyennes des maxima en haut, moyenne des moyennes en bas

Nous remarquons sur le tableau précédent que les augmentations en pourcentages des moyennes des maxima sont proches de celles des poids appliqués en conditions réelles. Nous pouvons aussi remarquer que les variations des moyennes des moyennes sont plus élevées pour cet essai. Or, lors des essais suivants, nous avons toujours trouvé des

valeurs proches pour les maxima, tandis que les résultats quant aux moyennes étaient très variables.

Les capteurs étant situés sous le pied, la pression sera alors maximale lorsque le pied sera à plat. En utilisant une fenêtre glissante, nous sommes capables de conserver cette valeur en mémoire sur un cycle entier. Elle témoignera du poids appliqué, et sera relativement constante. C'est pourquoi je décide de relever les moyennes des maxima en pression. **Les variations de quelques pourcentages pourront être notées**, alors que nous avons plus de difficultés à remarquer des variations sur les moyennes des moyennes. En effet, les moyennes des moyennes suivent le signal entier, c'est-à-dire qu'elles décrivent tout le cycle de la marche.

Solution retenue

Dès qu'une personne rentre dans son intervalle respectif de Var_2 , on souhaite relever le **maximum de pression sur une fenêtre glissante de 25 données**, puis le moyenniser. Celui-ci est calculé pendant la marche, et dès que la carte de développement se met en veille, sa valeur moyenne est alors **stockée sur le serveur**, avec l'actimétrie. A chaque activité réalisée dans des conditions jugées « normales », nous relevons une nouvelle valeur. En fin de journée, on réalise alors une **moyenne de ces moyennes de maxima**, que nous stockons, sur le serveur. Les codes en C se situent en annexe.

Les résultats sur le serveur seront conservés chaque jour, semaine, mois. Leur comparaison sera facile à réaliser, à l'aide d'un calcul de pourcentage de ces pressions – et par extension, du poids. Ensuite, dès que les valeurs obtenues varieront sur le serveur, une alerte pourra être envoyée à l'application « Respect » ainsi qu'au médecin. Ce dernier pourra demander à son patient de se peser, ou de venir le consulter. **Les risques d'être confronté à une perte involontaire de poids seront alors diminués**, car repérés à temps.

2. Détection de pas : Amélioration de l'existant

L'application Respect et sa semelle associée ont été développées avant mon arrivée. L'actimétrie était alors déjà programmée, et prête pour une période de tests. Néanmoins, cette dernière reposait sur une idée de seuils numériques, définissant des **fronts montants et descendants** de l'amplitude de l'accélération. Un problème auquel nous aurions pu être confrontés est l'adaptation à tout type de démarche. En effet, l'algorithme ne pourra être modifié en fonction de chaque patient, et les seuils numériques définissent une démarche. Des périodes de délais entre chaque foulée étaient aussi définies, et ne pouvaient être **compatibles avec une marche dans des escaliers ou une course**.

Souhaitant **affiner les seuils numériques** de son algorithme, il m'a été demandé de retracer des courbes de marche, comme celle présentée figure 37.

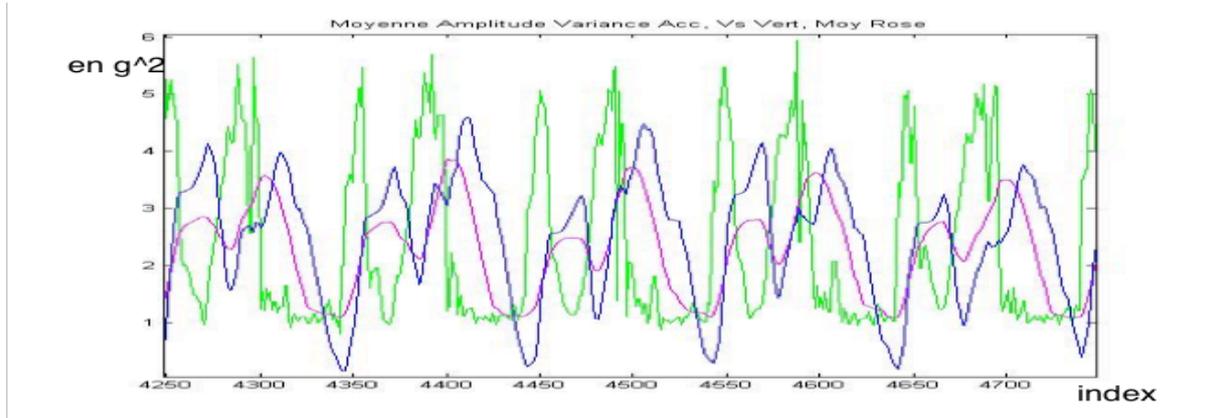


Figure 37 - Graphique représentant la marche, avec Vs2 en vert, Moy2 en rose, et Var2 en bleu, en g², en fonction du temps (10 index = 100ms)

Comme nous pouvons le remarquer sur ce graphique, Vs2 atteint des valeurs minimales pendant une certaine période. Elle est le **signe d'une nouvelle foulée**, lorsque le pied est à plat sur le sol, puisque l'accéléromètre est sous la voûte plantaire.

Proposition

Sur la figure précédente, qui a été réalisée à 2km/h, et qui a donc un Var2 assez faible, nous pouvons remarquer qu'à chaque fois que nous avons une nouvelle foulée, Var2 et Moy2 se croisent. Aussi, moyennes et variances sont de la même échelle, mais comme nous l'avons vu précédemment, Var2 varie considérablement lorsque nous passons à de la marche rapide, ou de la course. C'est pourquoi j'ai décidé de calculer une nouvelle variance, **sans fenêtre glissante**, que j'ai nommée **Var3** :

$$Var3_j = (a_{j2} - \overline{a_{j2}})^2 = (Vs2 - Moy2)^2$$

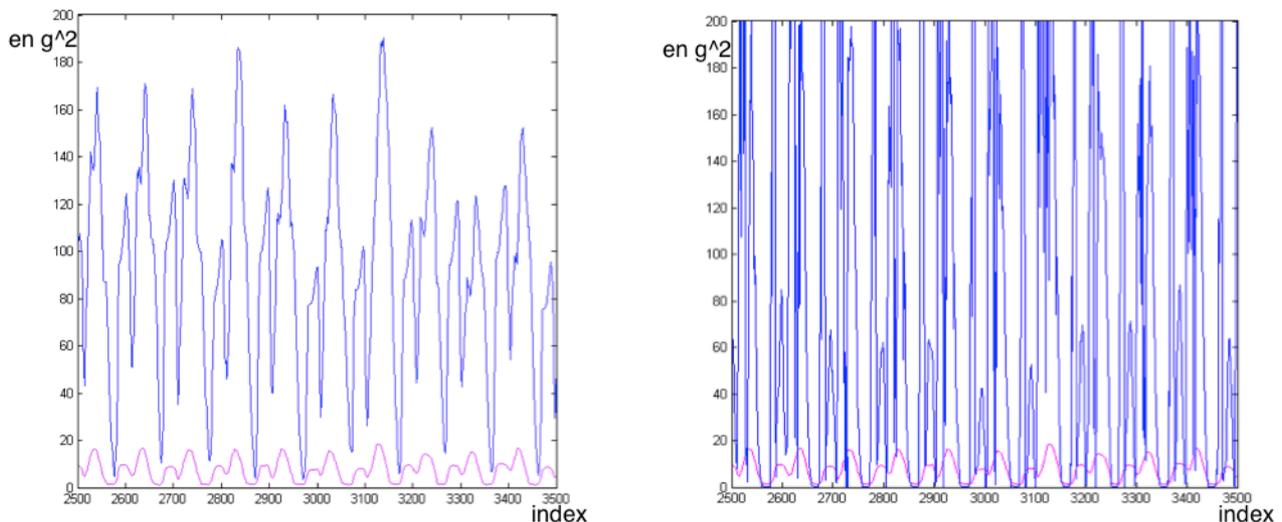


Figure 38 - Comparaison de l'utilisation de Var2 (à gauche) et Var3 (à droite), en g², Var2 et Var3 en bleu, Moy2 en rose

Nous remarquons sur les graphiques figure 38 que Var2 est trop haut pour croiser Moy2, comme c'était le cas dans la première figure. Néanmoins, si nous calculons la **variance à chaque valeur** – et non pas sur une fenêtre glissante, nous pouvons remarquer que **Var3 croise Moy2 dès que l'on a une nouvelle foulée**. Cela n'arrive

que pour **un seul index** lorsqu'on se situe dans un seuil minimal de moyenne, c'est-à-dire pour un nouveau pas.

En observant mes courbes, j'ai ainsi réalisé ma première modélisation sous MatLab (figure 39). J'ai compté mes pas, et vérifié mes résultats.

```

87 - global Nbpas;
88 - Nbpas = 0;
89 - for m=1:Max3;
90 -     M2 = m+25;
91 -     M1 = m;
92 -     for n=M1:M2;
93 -         if Moy2(n) < min(Moy2(M1:M2));
94 -             Minacc(n) = Moy2(n);
95 -         else
96 -             Minacc(n) = min(Moy2(M1:M2));
97 -         end
98 -     end
99 - end
100 -
101 - Seuilacc = Minacc + 0.1.*Minacc;
102 -
103 - for t=1:Max-1
104 -     if Moy2(t) <= Seuilacc(t);
105 -         if Moy2(t+1) > Moy2(t);
106 -             if ( Moy2(t) >= Var3(t) ) && ( Moy2(t+1) <= Var3(t+1) );
107 -                 Nbpas = Nbpas + 1;
108 -             end
109 -         end
110 -     end
111 - end
    
```

Figure 39 - Code MatLab du compteur de pas

Alors que la précision était assez faible avec le précédent code du compteur de pas, la modélisation sous MatLab a démontré une **précision** souvent **supérieure à 95%** - ce qui est notre objectif.

Tests

Comme le montre le graphique de la figure 40, il est donc assez simple de détecter les pas grâce à ce code, **sans utiliser de seuils numériques**.

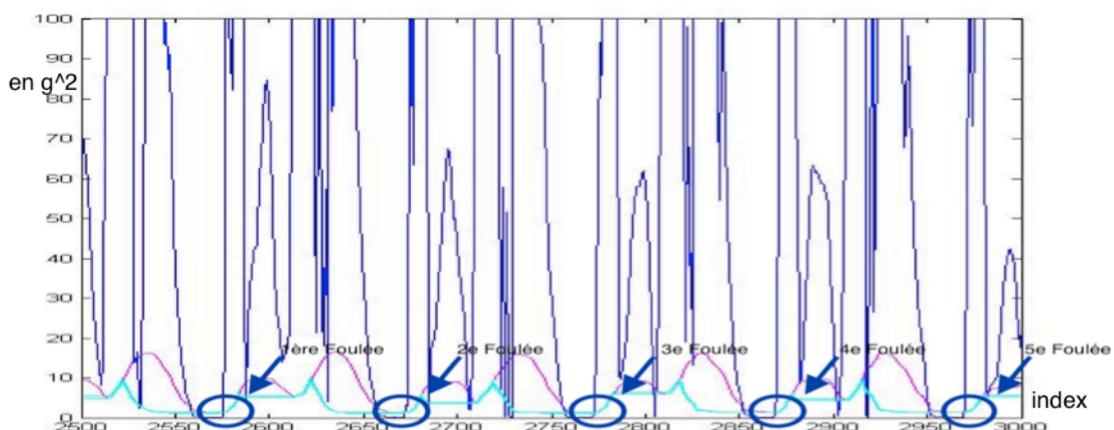


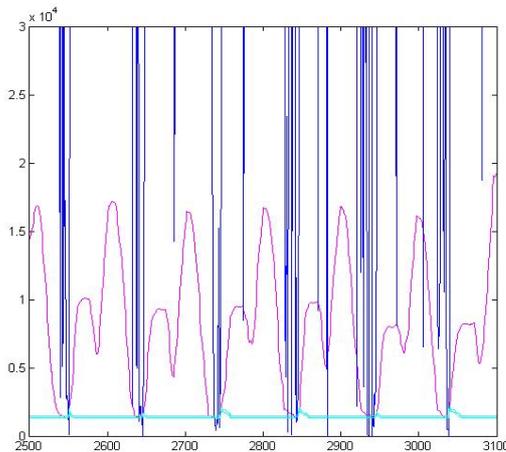
Figure 40 - Mise en évidence des foulées, avec Var3 en bleu, Moy2 en rose, et les minima et seuils de 10% des minima en cyan, en g^2

Néanmoins, les valeurs sont très précises, et l'utilisation de nombres à virgule est très limitée dans un microprocesseur, pour une question de consommation. En effet, pour la détection du poids, Var2 étant en général compris entre 3 et 15 g^2 , la précision à une

décimale près n'était pas nécessaire. En revanche, pour cette étude, je ne pouvais pas me permettre de convertir les données en g dans le microprocesseur. La conversion n'étant pas linéaire, j'ai donc refait une modélisation sous MatLab, afin de l'intégrer ensuite à la carte de développement.

Solution retenue

Afin d'optimiser le code en embarqué, j'ai observé mes courbes de marche, à la recherche d'un critère similaire. J'ai donc relevé Vs2, Moy2 et Var3 avec des parties entières, afin de me mettre dans des conditions similaires à celle dans la carte (figure 41).



```

74 - global Nbpas2;
75 - Nbpas2 = 0;
76
77 - for t=1:Max-1
78 -     if Moy2(t) <= Seuilacc(t);
79 -         if Moy2(t+1) > Seuilacc(t+1);
80 -             if Var3(t) <= Moy2(t) <= Var3(t+1);
81 -                 Nbpas2 = Nbpas2 + 1;
82 -             end
83 -         end
84 -     end
85 - end
74 - global Nbpas2;
75 - Nbpas2 = 0;
76
77 - for t=1:Max-1
78 -     if Moy2(t) <= Seuilacc(t);
79 -         if Moy2(t+1) > Seuilacc(t+1);
80 -             if Var3(t) <= Moy2(t) ^ Moy2(t) <= Var3(t+1);
81 -                 Nbpas2 = Nbpas2 + 1;
82 -             end
83 -         end
84 -     end
85 - end
    
```

Figure 41 - Modélisation de Var3 et Moy2, données brutes du microprocesseur sans conversion en g, et modélisation du code embarqué sous MatLab

Note : Sous MatLab, écrire $a < b < c$ correspond en réalité à $a < b \wedge b < c$, ie un « XOR », « OU EXCLUSIF ». Avec un « OU INCLUSIF », ou un « ET », les résultats ne sont étonnamment pas convaincants.

Le code en C se situe en annexe, mais reprend la même trame que la modélisation MatLab.

En conclusion, ce nouveau compteur possède de grands avantages (figure 42). Premièrement, il ne possède **plus de seuils numériques quant aux variances**, et donc une **liberté de démarche**, mais **aussi quant aux délais de temps**. En effet, pendant les périodes de pause, Var3 et Moy2 ne se croisent pas.

Il est donc inutile d'ajouter des délais de temps quant au manque d'activité par exemple. Néanmoins, nous devons encore tester l'ajout d'un petit délai anti-rebond, en cas de faux mouvement.

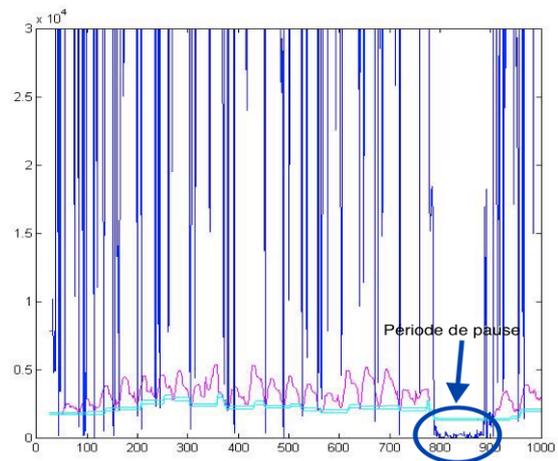


Figure 42 - Avantage du nouveau compteur de pas quant aux délais de temps (avec Var3 en bleu, Moy2 en rose, et données brutes du microprocesseur sans conversion)

Le code du compteur de pas a donc finalement été **optimisé** durant ce stage.

Conclusion

I. Bilan du projet

Le principal objectif de ma mission était de réaliser une **classification des activités** efficace, et d'implémenter une fonction de variation de poids qui lui soit liée. Néanmoins, comme tout projet de recherche, j'ai du réaliser des **tâches transverses**, comme une **étude bibliographique** ou encore de **l'optimisation de l'existant**.

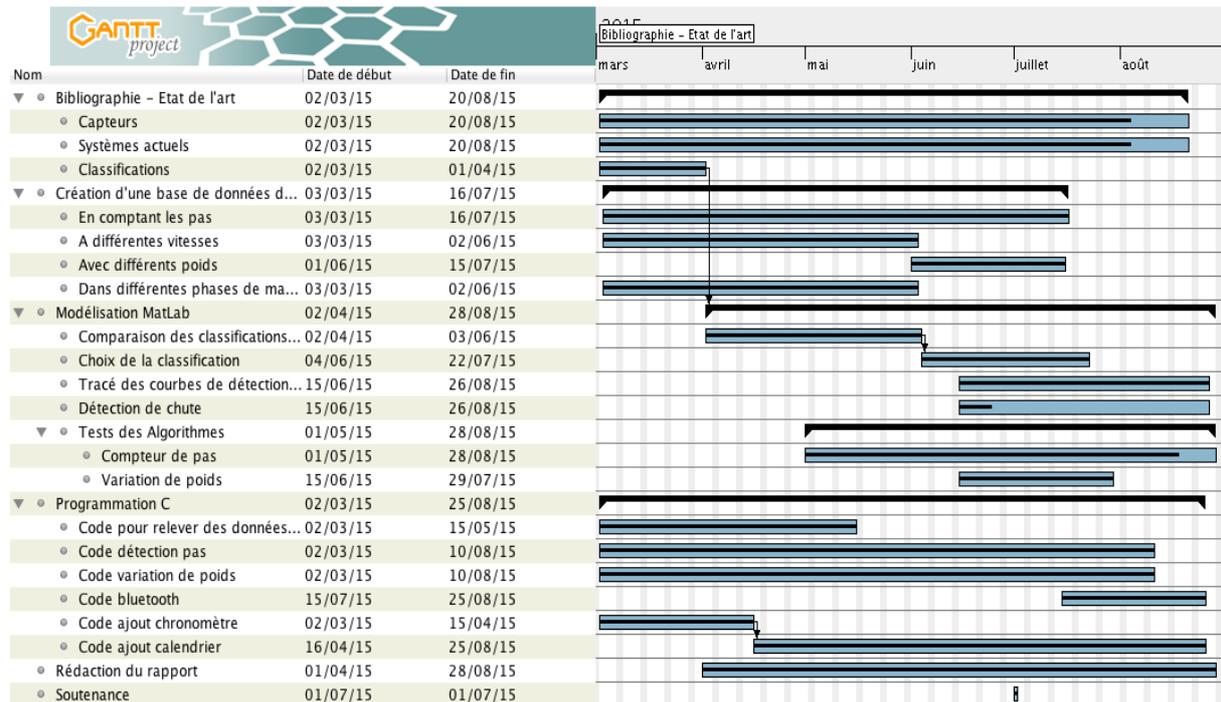


Figure 43 - GANTT de mon projet

J'ai rapidement réussi à prendre en main la carte de développement, afin d'obtenir mes bases de données et de pouvoir travailler sur les différentes fonctions qu'il m'était demandé de développer. Comme il est possible de le remarquer dans ce rapport, chaque tâche découlait naturellement d'une autre. En effet, chaque livrable m'a permis d'en réaliser un autre (figure 43).

II. Synthèse des résultats

Méthodologie

En effet, j'ai commencé mon projet par un état de l'art, puis me suis concentrée sur les classifications, car je savais qu'elles constituaient l'étape cruciale de mon stage. Après une étude sur une petite base de données, j'ai décidé d'agrandir cette dernière afin de pouvoir confirmer mes différentes suppositions quant aux choix d'une méthode. En la combinant avec les contraintes liées à l'embarqué, comme la consommation énergétique de la carte de développement, j'ai donc trouvé une méthode d'autant plus

efficace, à travers la définition de $Var2$, variance des carrés des amplitudes des accélérations de la marche.

A partir de cette nouvelle base de données, et des tracés des courbes de marche que j'ai réalisés, j'ai réussi à proposer une méthode de détection des pas. Après une période de tests en modélisation, j'ai ensuite pu **programmer la carte de développement**, afin d'y implémenter ces nouveaux codes. A ce stade de mon stage, je teste encore les codes en embarqué, en marchant avec la semelle, la carte, et une tablette possédant l'application « Respect ».

Récapitulatif des résultats

- **Classification**

On calcule à présent, à partir des données d'accélération fournies par l'accéléromètre, le carré du vecteur somme, sa moyenne sur une fenêtre glissante de 25 données, ainsi que sa variance. Un **intervalle de cette variance** pourra ensuite être défini, pendant une période de calibration, pour déterminer une **phase de « marche normale sur du plat »** de la personne à suivre.

Par exemple, supposons qu'une personne marche en moyenne à 4km/h, il suffira juste de relever $Var2$ au moment de la calibration, en prenant 10% en dessous et au-dessus de la « vitesse normale ». Une fois situés dans l'intervalle de $Var2$ calculé, nous commencerons alors à relever les variations de poids.

- **Mesure de la variation du poids**

Lorsque nous marchons dans l'intervalle de fonctionnement de $Var2$, nous mesurons alors la pression appliquée à la fois au talon et à l'avant du pied. **Deux capteurs sont nécessaires**, pour augmenter la précision de la mesure. Aussi, en relevant les maxima sur des fenêtres glissantes, puis en les moyennant, nous avons une idée de la **pression appliquée lorsque tout le pied est à plat**, ce qui est plus représentatif du poids que seulement le talon ou le métatarse. Chaque jour, la **moyenne des moyennes des maxima** est réalisée et stockée sur le serveur, puis comparée aux données précédentes. Avoir la possibilité de remarquer une perte de 5% du poids sur une semaine permettrait de prévenir le médecin pour une prise en charge de la personne.

- **Amélioration du compteur de pas**

A présent, il n'est **plus nécessaire** d'utiliser des **seuils numériques** dans le code en embarqué, afin de mesurer le nombre de pas réalisés. Une caractéristique de la marche, repérée dans mes graphiques, est le **croisement entre $Var3$** , variance du carré du vecteur somme à chaque donnée, **et $Moy2$** , la moyenne de ce même carré. Ces deux données ne se croisent qu'une seule fois lorsque nous atteignons un minimum de la moyenne de l'accélération, c'est-à-dire lorsque le pied est à plat et l'accéléromètre contre le sol (**nouvelle foulée**). L'**ajout de délais de temps** en cas de faux rebond ou d'inactivité est alors inutile, puisque sans mouvement, ces données ne se croisent jamais. Le nouveau code permet une **réelle optimisation** de l'existant.

III. Suite à donner au projet, perspectives

Les algorithmes développés devraient permettre de suivre l'activité de marche et la variation de poids afin de mettre en place un suivi de personnes fragiles à domicile et

faire de la prévention. La robustesse des algorithmes doit dorénavant être testée en conditions réelles. Bien que je teste à présent mes codes en embarqué, et en conditions réelles, je ne vérifie pas les conditions de marche d'une personne âgée. Il sera donc nécessaire de **tester les nouveaux algorithmes** sur un panel de personnes de plus de 65 ans. Nous commencerons par des personnes **robustes**, et après validation, pourrons passer aux personnes **fragiles**.

En outre, certaines fonctions peuvent être ajoutées à la semelle, cette fois-ci dans un cadre de la **surveillance et la détection de dangers**.

En effet, pendant mon stage, de nouvelles perspectives se sont ouvertes à la semelle. En effet, nous souhaitons à présent ajouter une **détection de chute** à la semelle. Elle permettra de **lancer une alerte** le cas échéant. Les chutes peuvent causer des pertes de connaissance, des fractures... Il est donc très important de les détecter. Des robots du bâtiment ADREAM au LAAS sont déjà conçus pour aider les personnes à se relever en cas de chute, mais sont encore au stade de prototypes. La nouvelle semelle aura pour objectif d'appeler à l'aide si une chute survient.

Lors de mon dernier mois de stage, je travaillerai sur les **différents critères** que nous pouvons repérer avec la semelle lors d'une chute. Ces résultats seront alors utilisés à l'avenir, à travers un projet plus complexe que l'actuelle semelle RESPECT.

Troisième Partie : Bilan personnel de mon projet professionnel

Le métier de chercheur est une profession qui soulevait beaucoup d'interrogations en moi lorsque j'étais en M1. Néanmoins, mon précédent stage m'avait énormément plu, et je souhaitais confirmer mes ambitions de chercheuse, avec un stage de longue durée, qui me permettrait d'atteindre tous les objectifs que l'on m'avait fixés, et ne pas avoir une fois encore la frustration de laisser le projet à un futur doctorant ou stagiaire.

Ce stage a répondu à mes attentes personnelles, car il m'a permis d'endosser le rôle de chercheur, et cette position m'a énormément plu. En effet, j'aime la science depuis longtemps, car j'aime la logique et ma grande satisfaction est de proposer des solutions qui fonctionnent. Lorsque l'on est chercheur, les résultats ne dépendent que de nous. Il n'y a pas d'horaires précis de travail, seulement, si l'on veut avancer, il ne tient qu'à nous de redoubler d'efforts. J'aime cette mentalité, et bien que la microélectronique et la programmation étaient pour moi des domaines inconnus, ce fut une expérience très enrichissante. Etant très curieuse, découvrir un nouveau domaine ne me dérangeait absolument pas, et mon adaptation fut en conséquence assez rapide. J'ai pu apprendre très rapidement, grâce à une motivation sans limites, qui est restée constante depuis le début de mon projet jusqu'à sa fin.

En M1, je doutais de mes capacités à réaliser une thèse. La quantité de travail pendant deux années de classes préparatoires était déjà considérable, et je ne savais pas si j'étais prête à m'investir dans une expérience similaire. Néanmoins, lorsque j'avais vu les doctorants du laboratoire, qui conservaient une vie sociale active, ma capacité d'adaptation à toutes sortes de sujet, ainsi que le bonheur qu'éprouvent les chercheurs à travers le sentiment d'accomplissement d'un projet, j'arrivais à me projeter dans quelques années, en tant que docteur. Cependant, les domaines du biomédical qui m'intéressent le plus sont plus portés sur de la robotique, des exosquelettes, voire la microélectronique. Avant de me lancer dans un doctorat de trois ans dans un domaine inconnu mais attrayant, je voulais tout d'abord déterminer si j'étais capable de réussir à réaliser un projet semblable. Ce stage avait un intitulé fascinant, je voulais déterminer si, sans compétences particulières pour ce dernier, en seulement six mois, j'allais réussir à le mener à bout. Selon moi, l'accomplissement de ce projet devait me prouver que j'étais enfin prête à réaliser un doctorat, que j'étais assez mûre et autonome pour cela.

À la fin de mon cursus à l'École des Mines d'Albi, je peux dire que j'ai beaucoup de connaissances, en modélisation notamment, et en matériaux. Mais le plus satisfaisant reste l'expérience que j'ai pu acquérir pendant mes stages. J'ai appris que je pouvais faire de la chimie dans l'ingénierie biomédicale, et m'en sortir. Cette année, j'ai enfin pu découvrir ce qui m'attirait tant depuis des années : la microélectronique, dans le cadre de la recherche en ingénierie biomédicale. Ce domaine est tellement vaste, que je suis réellement satisfaite d'avoir pu toucher à plusieurs de ses facettes, mais surtout passer le plus long de mes stages dans ce qui m'attirait le plus. En combinant à la fois mes

connaissances théoriques de l'école, et mes compétences pratiques acquises pendant mes stages, je pense réellement que ma carrière peut désormais commencer du bon pied.

La mentalité du chercheur, et l'atmosphère d'un centre de recherche sont particulièrement agréables à partager. J'aime l'autonomie, et la prise d'initiatives. Mon stage n'a duré que six mois, mais il me semble évident qu'un véritable chercheur doit adorer le sentiment de réussite de son projet lorsqu'il travaille dessus depuis longtemps. La publication doit être un véritable accomplissement. Ce sentiment est typiquement celui que je souhaite éprouver dans ma future vie professionnelle.

Bibliographie

YOANN CHARLON, *Conception de dispositifs électroniques portés pour le suivi de l'état de santé des personnes âgées*, Thèse, 2014,

<https://tel.archives-ouvertes.fr/tel-01079731/document>

YOANN CHARLON, ERIC CAMPO, *A Smart Insole to Promote Healthy Aging for Frail Elderly Individuals: Specifications, Design, and Preliminary Results*, Article, 2015,

http://www.researchgate.net/publication/277933547_A_Smart_Insole_to_Promote_Healthy_Aging_for_Frail_Elderly_Individuals_Specifications_Design_and_Preliminary_Results

[1] BRIGHAM YOUNG UNIVERSITY, *Composite Material Used as a Strain Gauge*, brevet n°WO2014144532(A1), Septembre 2014,

http://wo.espacenet.com/publicationDetails/biblio?DB=wo.espacenet.com&II=0&ND=3&adjacent=true&locale=en_EP&FT=D&date=20140918&CC=WO&NR=2014144532A1&KC=A1

[2] ARVAI LAZLO, *Device for Monitoring Balance and a Method for Manufacturing thereof*, brevet n° WO2013038214(A2), Mars 2013,

http://wo.espacenet.com/publicationDetails/biblio?DB=wo.espacenet.com&II=0&ND=3&adjacent=true&locale=en_EP&FT=D&date=20130321&CC=WO&NR=2013038214A2&KC=A2

[3] Qi2 ELEMENTS II, LLC, *Foot-Mounted Sensor Systems for Tracking Body Movement*, brevet n°WO2014100045(A1), Juin 2014,

http://wo.espacenet.com/publicationDetails/biblio?DB=wo.espacenet.com&II=0&ND=3&adjacent=true&locale=en_EP&FT=D&date=20140626&CC=WO&NR=2014100045A1&KC=A1

[4] ASPHAHANI FAREID, *Portable system for analyzing human gait*, brevet n°US6836744 (B1), Décembre 2004,

<http://worldwide.espacenet.com/publicationDetails/biblio?CC=US&NR=6836744B1&KC=B1&FT=D>

[5] NOKIA CORP, *Directing Shoe Insole*, brevet n°WO2008151642(A1), Décembre 2008,

<http://worldwide.espacenet.com/publicationDetails/biblio?CC=WO&NR=2008151642A1&KC=A1&FT=D>

[6] SAZONOV, E.S., *Development of SmartStep : an insole-based physical activity monitor*, Article, Juillet 2013,

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6611221>

[7] SAZONOV, E.S., *SmartStep : a Fully Integrated, Low-Power Insole Monitor*, Article, Juin 2014,

<http://www.mdpi.com/2079-9292/3/2/381/htm>

[8] ETH ZURICH, *Human Motion Analysis Method and Device*, brevet n°WO2014154352(A1), Octobre 2014,

http://wo.espacenet.com/publicationDetails/biblio?DB=wo.espacenet.com&II=0&ND=3&adjacent=true&locale=en_EP&FT=D&date=20141002&CC=WO&NR=2014154352A1&KC=A1

[9] WENLONG TANG, *Highly Accurate Recognition of Human Postures and Activities Through Classification With Rejection*, Article, Janvier 2014,

<http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6693687>

[10] GUILLAUME BOUCHARD, *Les modèles génératifs en classification supervisée et applications à la catégorisation d'images et à la fiabilité industrielle*, Thèse, 2005,

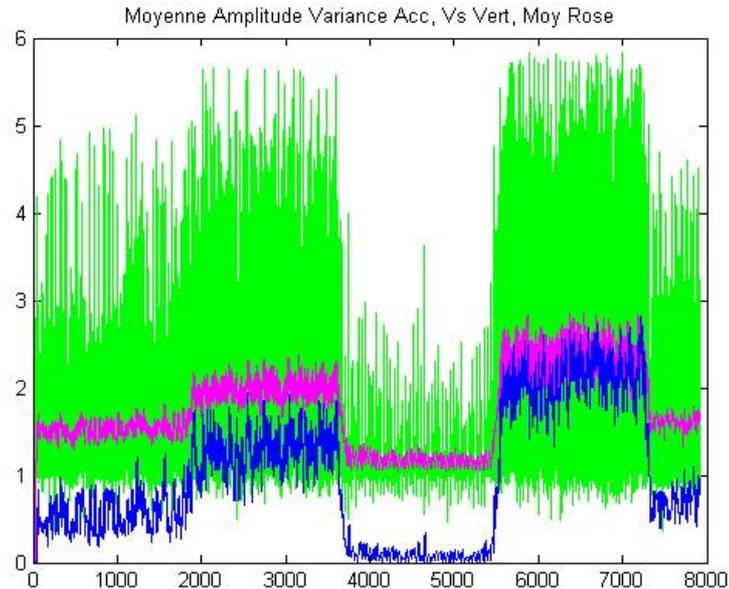
<http://www.math.u-psud.fr/select/pub/thesegeb.pdf>

Table des Illustrations et des Annexes

Tableau 1 – Variances des amplitudes au carré en fonction des phases de marche : sur du plat, en montée, descente, à 3 vitesses différentes, en g^2	10
Tableau 2 - Evolution en pourcentages des pressions moyennes et de leurs maxima en fonction des poids appliqués – Moyennes des maxima en haut, moyenne des moyennes en bas	Erreur ! Signet non défini.
Tableau 3 - Comparatif des taux d'erreurs de chaque méthode sous MatLab	29
Tableau 4 - Variances en fonction des phases de marche : sur du plat, en montée, descente, à 3 vitesses différentes, en g.....	30
Tableau 5 – Variances des amplitudes au carré en fonction des phases de marche : sur du plat, en montée, descente, à 3 vitesses différentes, en g^2	31
Figure 1 – L'architecture du système opérationnel	7
Figure 2 – Points forts en pression, positions des capteurs	8
Figure 3 – Les différents types de classifications supervisées	9
Figure 4 – Variances des carrés des amplitudes en g^2 en fonction des vitesses de marche en km/h, en phases d'apprentissage et de test	10
Figure 5 - Variation de la pression sur les capteurs, en continu, en fonction des index - pression au niveau du talon en rose, pression à l'avant en bleu, et somme des deux en vert, moyenne des maxima en cyan	10
Figure 6 - Graphique représentant la marche, avec Vs2 en vert, Moy2 en rose, et Var2 en bleu, en g^2 , en fonction du temps (10 index = 100ms).....	12
Figure 7 - Mise en évidence des foulées, avec Var3 en bleu, Moy2 en rose, et les minima et seuils de 10% des minima en cyan, en g^2	12
Figure 8 - Modélisation de Var3 et Moy2, données brutes du microprocesseur sans conversion en g.....	13
Figure 9 - Le LAAS-CNRS	15
Figure 10 - L'axe ADREAM en illustrations	15
Figure 12 - Les vitesses de marche selon les critères de Fried	16
Figure 12 - Matériel à ma disposition.....	18
Figure 13 - Les différents types de classifications supervisées	20
Figure 14 - Schématisation des deux approches	20
Figure 16 - Avantages et inconvénients des méthodes discriminatives (à gauche) et génératives (à droite)	21
Figure 17 - Répartition des Données d'accélération en phase d'Apprentissage	22
Figure 18 - Illustration du fonctionnement de LDA.....	23
Figure 19 - LDA	24
Figure 20 - QDA.....	24
Figure 21 - Critère ICL, Integrated Completed Likelihood.....	24
Figure 22 - Critère BIC, Bayesian Inferior Criterion	24
Figure 23 - Critère NEC – Critère d'entropie	24
Figure 24 - Classification Naive Bayésienne, Données d'accélération	25
Figure 25 - Classification Naive Bayésienne, Données de pression	25
Figure 26 - Les Marges et la Frontière Séparatrice	26
Figure 27 - La transformation de l'espace de représentation	26

Figure 28 - Paire de chaussures SmartShoe équipées des capteurs	27
Figure 29 - Différents Arbres de Régression et Classification	28
Figure 30 - Variances en fonction des vitesses de marche en km/h, en phases d'apprentissage et de test, en g	30
Figure 31 - Variances des carrés des amplitudes en fonction des vitesses de marche en km/h, en phases d'apprentissage et de test, en g ²	32
Figure 32 - Evolution des Var2 (Bleu), Moy2 (Rose) et Vs2 (Vert) en fonction des vitesses de marche (index 0 à 2000 : 4km/h, 2000 à 4000 : 6 km/h, 4000 à 5500 : 2km/h, 5500 à 7500 : 8 km/h, 7500 à 8000 : 4km/h) en g ²	32
Figure 33 - Code de Var2, fonction getAcc_Moy	33
Figure 34 - Positions des capteurs de pression	34
Figure 35 - Variation de la pression sur les capteurs, en continu, en fonction des index - pression au niveau du talon en rose, pression à l'avant en bleu, et somme des deux en vert, moyenne des maxima en cyan	34
Figure 36 - Résultante des forces verticales en fonction des étapes de la phase d'appui (a) et résultante des forces verticales en fonction de la vitesse (b).....	35
Figure 37 - Variation de la pression en fonction des poids appliqués	36
Figure 38 - Graphique représentant la marche, avec Vs2 en vert, Moy2 en rose, et Var2 en bleu, en g ² , en fonction du temps (10 index = 100ms).....	38
Figure 39 - Comparaison de l'utilisation de Var2 (à gauche) et Var3 (à droite), en g ² , Var2 et Var3 en bleu, Moy2 en rose.....	38
Figure 40 - Code MatLab du compteur de pas.....	39
Figure 41 - Mise en évidence des foulées, avec Var3 en bleu, Moy2 en rose, et les minima et seuils de 10% des minima en cyan, en g ²	39
Figure 42 - Avantage du nouveau compteur de pas quant aux délais de temps (avec Var3 en bleu, Moy2 en rose, et données brutes du microprocesseur sans conversion)	40
Figure 43 - GANTT de mon projet.....	41
Annexe 1 - Identification des différentes vitesses de marche à l'air de Var (bleu), Vs (vert) et Moy (rose), en g	50
Annexe 3 - Vs, Moy et Var des différentes vitesses de marche, en g.....	50
Annexe 4 - Partie du code C de la boucle principale du fichier <code>main.c</code> , appelant <code>getAcc_Moy</code> , fonction du calcul de Var2, <code>process_pressure</code> gérant les maxima et moyennes des pressions, et <code>step_detection</code> , le compteur de pas.....	51
Annexe 5 - Structure de l'Actimétrie.....	51
Annexe 6 - Code de la fonction <code>process_pressure</code> , le code gérant les maxima et moyennes des pressions	52
Annexe 7 - Code C de la fonction <code>step_detection</code> , le compteur de pas.....	53
Annexe 8 - Fonction <code>log_cadence_handler</code> , qui met à jour les données de la structure Actimétrie, appelée « actimétrie »	54
Annexe 9 - Envoi des valeurs de la structure actimétrie, dans la fonction <code>rscs</code> , "running speed and cadence service structure", définie avec le bluetooth.....	55

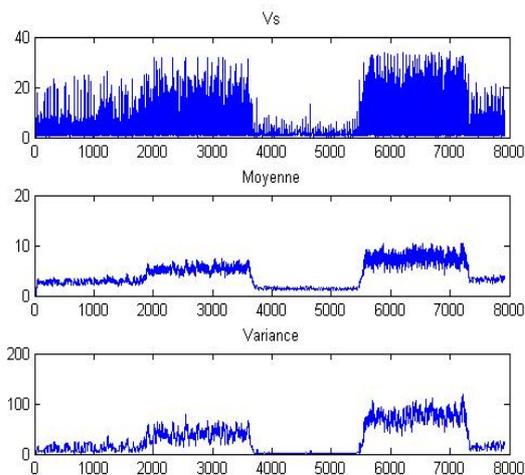
Annexes



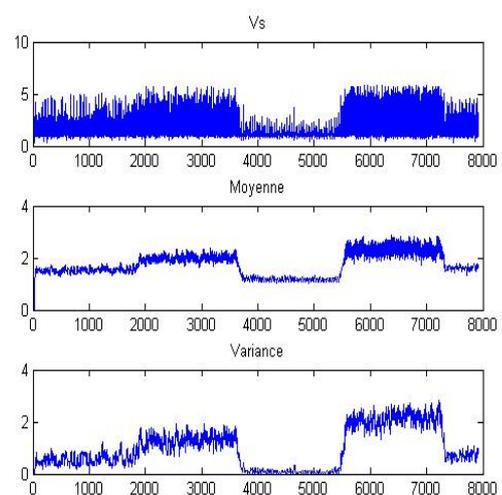
Annexe 1 - Identification des différentes vitesses de marche à l'air de Var (bleu), Vs (vert) et Moy (rose), en g

A l'aide de cette première annexe, nous remarquons que Var est efficace quant à l'identification d'une phase de marche à vitesse normale. Néanmoins, d'après les ordonnées, en g, nous remarquons que, bien que notables, ces différences seront difficiles à relever avec la carte de développement. En effet, l'utilisation de *float*, c'est-à-dire de nombres à virgule, est limitée et prend beaucoup d'espace dans le microprocesseur.

Nous pouvons étudier respectivement les Vs, Moy et Var sur les annexes suivantes, et notamment remarquer les différences d'échelles.



Annexe 2 - Vs2, Moy2 et Var2 des différentes vitesses de marche, en g²



Annexe 3 - Vs, Moy et Var des différentes vitesses de marche, en g

```
/*Boucle principale*/
while(1)
{
    /*Récupérations des mesures et calculs des données*/
    if(awake)
    {
        if(just_woke_up) /* Réveil */
        {
            just_woke_up = false;

            ADXL_FlushFIFO();
        }
        else
        {
            if(accok)
            {
                accok = false;
                step_detection();
                getAcc_Moy(); // C'est la fonction qui correspond au calcul de Var2
            }
            if (pressok)
            {
                pressok = false;
                process_pressure();
            }
        }
    }
    else
    {
        if(just_asleep)
        {
            just_asleep = false;
        }
    }
}
```

Annexe 4 - Partie du code C de la boucle principale du fichier **main.c**, appelant **getAcc_Moy**, fonction du calcul de Var2, **process_pressure** gérant les maxima et moyennes des pressions, et **step_detection**, le compteur de pas

```
struct StructActimetrie
{
    SYSTEMTIME time;

    UINT32 nb_pas;
    UINT32 walking_time; //en secondes
    UINT32 cadence_moy; // en pas/s

    UINT32 max_pression;
    UINT32 mean_pression;
};
```

Annexe 5 - Structure de l'Actimétrie

```

252 void process_pressure()
253 {
254     /*VARIABLES*/
255     extern UINT32 indexGetLog;
256     UINT32 Pres[indexGetLog];
257     extern BOOLEAN just_asleep;
258     UINT32 Max_Pres[indexGetLog];
259     char t;
260     UINT32 K = w-25;
261     static UINT32 count = 0, countmax = 0;
262     static UINT32 mean = 0, max = 0;
263
264     if (w>250)
265     {
266         w = 27;
267     }
268     w++;
269     Pres[w] = pression.avant + pression.talon; //Définition des pressions à relever
270     if(w>26)
271     {
272         for(t=K; t<w; t++) // On relève les maxima des pressions
273         {
274             if (Pres[w] > Pres[t])
275             {
276                 Max_Pres[w] = Pres[w];
277             }
278             if (Pres[t] >= Pres[w])
279             {
280                 Max_Pres[w] = Pres[t];
281             }
282             countmax++;
283             max += Max_Pres[w];
284         }
285         mean += Pres[w];
286         count ++;
287     }
288     /*Mise à jour*/
289     if(count >0 && countmax >0)
290     {
291         mean_pression += mean/count;
292         max_pression += max/countmax;
293         count_pression ++;
294     }
295     /* Remise à zéro */
296     max = 0;
297     mean = 0;
298     count = 0;
299     countmax=0;
300 }

```

Annexe 6 – Code de la fonction process_pressure, le code gérant les maxima et moyennes des pressions

Rapport de Stage de Fin d'Études

Semelle intelligente pour les personnes âgées

```
void step_detection()
{
    extern BOOLEAN accok;
    extern BOOLEAN awake;
    extern UINT32 indexGetLog;
    UINT32 Vs[indexGetLog];
    UINT32 Min[indexGetLog];

    UINT32 Var3[indexGetLog];
    UINT32 Seuil[indexGetLog];
    UINT32 Moy[indexGetLog];

    char j,i;

    UINT32 M = n-25;
    UINT32 N = n-50;

    if (awake)
    {
        n++;

        if (n>250) // On ne doit pas dépasser 255, car 256 = 2^8 -> Bit
        {
            n=52; //On redémarre directement à 52 pour pouvoir entrer dans la condition ci-dessous
        }

        Vs[n] = (acceleration.x*acceleration.x + acceleration.y*acceleration.y +acceleration.z*acceleration.z);
        // DBG_PRINT_ARG("Vs : %lu \n",Vs[m]);

        if (n > 51)
        {
            for (j=M; j<n; j++)
            {
                Moy[n] += Vs[j]/26; // Définition de Moy2
            }

            Var3[n] = ((Vs[n] - Moy[n])*(Vs[n] - Moy[n])); // Définition de Var3

            for (i=N; i<n; i++) // Définition du Minimum des Moyennes
            {
                if ((Moy[n] <= Moy[i]) && (Moy[N] >= Moy[n]))
                {
                    Min[n] = Moy[n];
                }
                if ((Moy[N] <= Moy[i]) && (Moy[n] >= Moy[N]))
                {
                    Min[n] = Moy[N];
                }
            }
            Seuil[n] = Min[n] + 0.1*Min[n]; //Et du seuil qui l'accompagne

            if (Moy[n-1] <= Seuil[n-1])
            {
                if (Moy[n] > Seuil[n])
                {
                    if ( (Var3[n-1] <= Moy[n-1]) ^ (Moy[n-1] <= Var3[n]))
                    {
                        CLEAR_LED2; //On allume une LED Orange à chaque nouvelle foulée
                        nb_pas +=2; // On compte +2 car on compte les foulées
                        //DBG_PRINT_ARG("Nbpas : %lu \n",nb_pas); // Sert afficher sur l'ordinateur
                    }
                    SET_LED2; //On éteint la LED Orange : la foulée est terminée
                }
            }
        }
    }
}
```

Annexe 7 - Code C de la fonction step_detection, le compteur de pas

```
143 void log_cadence_handler(void * p_context)
144 {
145     SYSTEMTIME time;
146
147     /*Heure du log*/
148     get_time(&time);
149
150     /*Mise à jour de l'actimétrie*/
151     actimetrie.time = time;
152     actimetrie.nb_pas = nb_pas;
153
154
155     if(count_pression>0)
156     {
157         actimetrie.max_pression = (max_pression/count_pression);
158         actimetrie.mean_pression = (mean_pression/count_pression);
159     }
160     else
161     {
162         actimetrie.max_pression = 0;
163         actimetrie.mean_pression = 0;
164     }
165
166     /*Si le système est configuré et il y a eu de l'activité*/
167     if( calendar_set && (actimetrie.nb_pas>0) )
168     {
169         framWakeup();
170
171         /*Log de l'actimétrie*/
172         loggerAddMes( (UINT8*)&actimetrie);
173
174         framSleep();
175     }
176
177     /*Remise à zéro des paramètres*/
178     max_pression = 0;
179     mean_pression = 0;
180     count_pression = 0;
181
182     m = 0;
183     n = 0;
184     w = 0;
185     new_data_available = true;
186
187 }
188
```

Annexe 8 - Fonction log_cadence_handler, qui met à jour les données de la structure Actimétrie, appelée « actimétrie »

On enverra alors la structure actimétrie à l'aide de la fonction send_log_cadence.

```
58 uint32_t send_log_cadence()
59 {
60     uint32_t err_code;
61
62     /* On vérifie qu'il reste des données à envoyer */
63     if(loggerGetNbMes() <= 0) return LOG_EMPTY_ERROR;
64
65     framWakeup();
66
67     /* Récupération de l'actimétrie */
68     loggerGetMes((UINT8*)&actimetrie), false);
69
70     rscs_meas.time.annees = actimetrie.time.wYear;
71     rscs_meas.time.mois = actimetrie.time.wMonth;
72     rscs_meas.time.jours = actimetrie.time.wDay;
73     rscs_meas.time.heures = actimetrie.time.wHour;
74     rscs_meas.time.minutes = actimetrie.time.wMinute;
75     rscs_meas.cadence = actimetrie.cadence_moy;
76     rscs_meas.duree = actimetrie.walking_time;
77     rscs_meas.nb_pas = actimetrie.nb_pas;
78
79     rscs_meas.pression_max = actimetrie.max_pression;
80     rscs_meas.pression_moyenne = actimetrie.mean_pression;
81
82     /* Envoi de l'actimétrie */
83     err_code = ble_rscs_measurement_send(&m_rscs, &rscs_meas);
84     if ((err_code != NRF_SUCCESS) &&
85         (err_code != NRF_ERROR_INVALID_STATE) &&
86         (err_code != BLE_ERROR_NO_TX_BUFFERS) &&
87         (err_code != BLE_ERROR_GATTS_SYS_ATTR_MISSING))
88     {
89         {
90             APP_ERROR_HANDLER(err_code);
91         }
92     }
```

Annexe 9 - Envoi des valeurs de la structure actimétrie, dans la fonction rscs, "running speed and cadence service structure", définie avec le bluetooth

Une fois les données chargées dans cette fonction, elles sont envoyées par bluetooth à l'application Android RESPECT.