EPM-RT-2019-01

CONCEPTION D'UN ALGORITHME DE FUSION DE DONNÉES POUR L'INTÉGRATION DE MESURES D'ACCÉLÉROMÈTRE ET DE GYROSCOPE POUR L'ESTIMATION D'ORIENTATION DE MEMBRE SUPÉRIEUR

Nelson Giraldo
Institut de génie biomédical
École Polytechnique de Montréal

Janvier 2019

Poly

ÉCOLE POLYTECHNIQUE MONTRÉAL

EPM-RT-2019-01

Conception d'un algorithme de fusion de données pour l'intégration de mesures d'accéléromètre et de gyroscope pour l'estimation d'orientation du membre supérieur.

Nelson Giraldo
Institut de génie biomédical
École Polytechnique de Montréal

Janvier 2019

©2019
Nelson Giraldo
Tous droits réservés

Dépôt légal :
Bibliothèque nationale du Québec, 2019
Bibliothèque nationale du Canada, 2019

EPM-RT-2019-01
Conception d'un algorithme de fusion de données pour l'intégration de mesures d'accéléromètre et de gyroscope pour l'estimation d'orientation du membre supérieur.
par : Nelson Giraldo
Institut de génie biomédical
École Polytechnique de Montréal

Toute reproduction de ce document à des fins d'étude personnelle ou de recherche est autorisée à la condition que la citation ci-dessus y soit mentionnée.

Tout autre usage doit faire l'objet d'une autorisation écrite des auteurs. Les demandes peuvent être adressées directement aux auteurs (consulter le bottin sur le site http://www.polymtl.ca/) ou par l'entremise de la Bibliothèque :

École Polytechnique de Montréal
Bibliothèque – Service du Prêt entre bibliothèques
Case postale 6079, Succursale « Centre-Ville »
Montréal (Québec)
Canada H3C 3A7

Téléphone : (514) 340-4846
Télécopie : (514) 340-4026
Courrier électronique : biblio.sfd@courriel.polymtl.ca

Ce rapport technique peut-être repéré par auteur et par titre dans le catalogue de la Bibliothèque :
http://vsmart.biblio.polymtl.ca/epmvw/Vubis.csp?Profile=DEFAULT&OpacLanguage=fre

Résumé

Les capteurs inertiels sont devenus largement disponibles et à faible coût. Les mesures peuvent être acquises à haute fréquence et intégrées pour obtenir des estimations d'orientation. Dans ce projet, un système de 9 capteurs inertiels Delsys® TrignoMT est utilisé pour estimer l'orientation. Les estimations du gyroscope sont exactes à court terme mais sont affectées par une dérivation d'intégration sur des intervalles plus longs. Les données de l'accéléromètre sont soumises à un bruit élevé et sont sensibles aux vibrations. Nous utilisons donc un algorithme de fusion de données pour combiner les données des deux capteurs et obtenir une estimation d'orientation plus précis

L'intégration des mesures se fait après l'acquisition et les mouvements suivis sont cycliques, nous utilisons donc une approche simple pour intégrer à la fois les données de l'accéléromètre et du gyroscope pour obtenir l'orientation des membres supérieurs. L'orientation initiale est calculée à l'aide des données de l'accéléromètre et les changements d'orientation ultérieurs sont estimés à l'aide des données du gyroscope.

L'algorithme de fusion des données s'est révélé fiable pour estimer l'orientation des membres supérieurs chez les sujets testés en donnant des estimations d'orientation qui étaient visuellement cohérentes avec les mouvements des bras du sujet testé.

GBM6903 – PROJET DE MAÎTRISE EN INGÉNIERIE III

Présenté à Maxime Raison

Rapport du Projet REAplan

Par Nelson Giraldo 1071268

Le 24 janvier 2019

Table des matières

Table des matières ____ *i*

Liste des figures ____ *iii*

Mise en contexte ____ *1*

Bras robotisé REAplan d'Axinesis ____ 2

Introduction ____ *3*

Projet antérieur ____ *3*

Calibration post-acquisition ____ *4*

Calibration du magnétomètre ____ 4

Calibration de l'accéléromètre ____ 8

Calibration du gyroscope ____ 9

Système Delsys® ____ *10*

Magnétomètre ____ 10

Accéléromètre ____ 10

Gyroscope ____ 10

Comparaison Accéléromètre vs Gyroscope ____ 11

Filtre Kalman ____ *11*

Phase de prédiction ____ 12

Phase de mise-à-jour ____ 12

Équations du filtre Kalman ____ 12

Version Extended ____ 13

Méthodologie ____ *13*

Protocole d'acquisition des données ____ 13

Séquence d'orientation : les angles d'Euler ZYX ____ 14

Systèmes de coordonnées (Delsys® vs Global) ____ 15

Filtre Kalman ____ 16

Acquisition des données ____ 18

Essais pour déterminer précision ____ 18

Calcul des angles du membre supérieur _____ **19**

Résultats _____ ***20***

Essais pour évaluer la précision _____ **20**

Discussion _____ ***29***

Conclusion _____ ***30***

Références _____ ***32***

Annexe _____ ***36***

Code Matlab _____ **36**

Liste des figures

Figure 1 - Exemple de calibration avec données en ellipsoïde avant calibration (rouge) et une sphère unitaire de données après calibration (bleu). 8
Figure 2 - Positionnement des centrales inertielles sur le patient. 14
Figure 3 - Représentation des angles d'Euler et leur axe associé. 15
Figure 4 - Robot REAplan avec système de coordonnées global. (Droits d'auteur : CUSL / H. Depasse) 15
Figure 5 - Système de coordonnées des centrales inertielles de Delsys® 16
Figure 6 - Plans anatomiques. 19
Figure 7 - Test de rotation de l'angle Yaw. (a) rotation autour de l'axe des z positif; (b) rotation autour de l'axe des z négatif. 21
Figure 8 - Test de rotation de l'angle Pitch. (a) rotation autour de l'axe des y positif; (b) rotation autour de l'axe des y négatif. 21
Figure 9 - Test de rotation de l'angle Pitch. (a) rotation autour de l'axe des y positif; (b) rotation autour de l'axe des y négatif. 22
Figure 10 - Test de mouvement Flexion-Extension du coude. 22
Figure 11 - Test de mouvement Abduction-Adduction de l'épaule. 23
Figure 12 - Test de mouvement Flexion-Extension de l'épaule. 23
Figure 13 - Test de mouvement Rotation interne-externe de l'épaule. 24
Figure 14 - Comparaison de l'angle Pitch pour IMU #2 pendant l'exercice cible, à partir des données du gyroscope (bleu), accéléromètre (rouge), et l'estimation du filtre EKF (vert). 25
Figure 15 - Comparaison de l'angle Roll pour IMU #2 pendant l'exercice cible, à partir des données du gyroscope (bleu), accéléromètre (rouge), et l'estimation du filtre EKF (vert). 25
Figure 16 - Comparaison de l'angle Pitch pour IMU #6 pendant l'exercice cible, à partir des données du gyroscope (bleu), accéléromètre (rouge), et l'estimation du filtre EKF (vert). 26
Figure 17 - Comparaison de l'angle Roll pour IMU #6 pendant l'exercice cible, à partir des données du gyroscope (bleu), accéléromètre (rouge), et l'estimation du filtre EKF (vert). 26
Figure 18 - Angles du membre supérieur pour l'exercice Cible. (a) Flexion-Extension du coude; (b) Abduction-Adduction de l'épaule; (c) Flexion-Extension de l'épaule; (d) Rotation interne-externe de l'épaule. 27
Figure 19 - Angles du membre supérieur pour l'exercice Carré. (a) Flexion-Extension du coude; (b) Abduction-Adduction de l'épaule; (c) Flexion-Extension de l'épaule; (d) Rotation interne-externe de l'épaule. 27
Figure 20 - Angles du membre supérieur pour l'exercice Cercle. (a) Flexion-Extension du coude; (b) Abduction-Adduction de l'épaule; (c) Flexion-Extension de l'épaule; (d) Rotation interne-externe de l'épaule. 28
Figure 21 - Angles du membre supérieur pour l'exercice Droite. (a) Flexion-Extension du coude; (b) Abduction-Adduction de l'épaule; (c) Flexion-Extension de l'épaule; (d) Rotation interne-externe de l'épaule. 28

Mise en contexte

La quantification des forces musculaires est la clé du suivi en réadaptation des troubles musculosquelettiques, un grand défi depuis Weber et Weber, 1836 [1]. La difficulté vient du fait que les articulations sont activées par une multitude de muscles, conduisant à une infinité de forces musculaires possibles. Pour résoudre ce problème et calculer les forces musculaires individuelles, le laboratoire du Centre de Recherche Marie-Enfant (CRME) a développé en 2011 une nouvelle méthode basée sur la modélisation dynamique multicorps, qui modélise le comportement dynamique des corps rigides ou flexibles interconnectés. Pour cette méthode, nous avons obtenu un brevet [2].

À partir de ce résultat, nous avons simplifié le processus de calcul de façon significative. Et, nous avons proposé une fonction de coût originale et innovatrice pour calculer les forces musculaires directement à partir des données cinématiques avec une simplicité comparable à celle de la fonction de coût de Crowninshield [3], la plus couramment utilisée dans le monde. En plus, notre fonction à l'avantage de prendre en considération la coactivation musculaire, ce qui est plus cohérant avec l'activité musculaire [4], [5]. Le modèle dynamique ostéoarticulaire du membre supérieur que nous avons développé a reçu un prix [6] et a été déclaré comme le modèle reproduisant le plus fidèlement le mouvement réel du membre supérieur [7], [8], [9].

Nous avons quantifié la fiabilité de nos mesures et calculs, en particulier les couples articulaires [10], les données électromyographiques brutes [11] et les forces isométriques maximales [12] des membres supérieurs, afin de prendre en considération les transferts cliniques. Aujourd'hui, cette expertise est utilisée de façon concrète avec plusieurs cliniques partenaires à travers le monde pour l'évaluation des fonctions motrices dans diverses populations : pour les enfants avec paralysie cérébrale au CHU Ste-Justine de Montréal [13], [14], [15], pour l'évaluation chirurgicale de l'épaule à l'Hôpital Sacré-Cœur de Montréal [16], [17], et pour les adolescents scoliotiques au CHU St-Luc en Belgique [18].

Pour les membres supérieurs, aucune recommandation clinique concernant le niveau d'assistance n'existe à ce jour à notre connaissance. Dans ce contexte, les mesures cinématiques et électromyographiques de laboratoire de mouvement, ainsi que les indicateurs quantifiés qui en découlent (angles, couples et puissances articulaires [19], [20], [21], co-activations musculaires [15], etc.) apparaissent comme des indicateurs ayant le potentiel de répondre à ces hypothèses et font intégralement partie de l'expertise de l'équipe de ce mandat [19], [20], [21], [14]. Il serait donc pertinent d'obtenir ces indicateurs chez les enfants atteints de paralysie cérébrale, équipés de capteurs électromyographiques et effectuant leurs exercices avec le bras robotisé REAplan immergé dans l'environnement d'un système de capture de mouvement, afin d'analyser la corrélation entre ces indicateurs et les niveaux d'assistance du REAplan.

À partir des mesures cinématiques et électromyographiques, les indicateurs biomécaniques suivants pourront être quantifiés :

- Angles, couples et puissances articulaires [19], [20], [21].
- Co-activations musculaires [14].

Bras robotisé REAplan d'Axinesis

Les dispositifs robotiques de réadaptation sont aujourd'hui reconnus pour améliorer la fonction motrice des patients, en augmentant la durée des exercices, de plus en plus sous forme de « jeux sérieux », ainsi que leur variété et leur adaptation au patient. L'application la plus générale des dispositifs robotiques de réadaptation concerne les patients après un accident vasculaire cérébral [22], [23].

Mais récemment, des études ont montré l'amélioration de la fonction motrice du membre supérieur chez des enfants atteints de paralysie cérébrale à l'aide du REAplan [24] (Figure 1). Le REAplan [25], [26] est un dispositif à retour haptique, c.à.d. qu'il est en mesure d'adapter automatiquement et en temps-réel son niveau assistance – formalisé par une raideur de ressort ramenant le préhenseur sur la trajectoire souhaitée selon le jeu – du membre supérieur suivant les efforts de l'utilisateur sur son préhenseur. Le REAplan se vend régulièrement en Europe, et le Centre de Réadaptation Marie-Enfant est le premier en Amérique du Nord à s'être doté (récemment) du REAplan.

Aux membres inférieurs, des études longitudinales avec des dispositifs de réadaptation à la marche tels que le Lokomat (HOCOMA, Suisse) ont indiqué que le niveau d'assistance est graduellement diminué au fur et à mesure que le nombre de séances de réadaptation augmente [27].

La collaboration avec Axinesis contribue à améliorer leur bras robotisé de réadaptation REAplan en leur permettant de recevoir une rétroaction quantitative de notre part sur la pertinence de l'utilisation du REAplan avec des nouvelles populations, surtout les patients avec des troubles neuromusculaires [28], [3], [29].

Introduction

Dans le cadre d'un projet de recherche d'une étudiante au doctorat, ce projet de maîtrise vise à venir en aide au niveau de l'acquisition, la calibration et la fusion des données de centrales inertielles pour obtenir une estimation des angles du membre supérieur lors d'exercices de réhabilitation sur le robot REAplan® de *Axinesis Rehabilitation Technologies*. Il fait suite au projet de maîtrise de Catherine Séguin [30] à l'automne 2017.

Le projet a lieu à l'école Victor-Doré, école spécialisée qui accueille des élèves de 4 à 13 ans qui présentent des déficiences motrices ou organiques graves. L'espace restreint disponible pour l'installation limite les choix de technologies. Il n'est donc pas possible d'utiliser un système d'acquisition par caméra tel Vicon ou BTS. C'est plutôt le système Delsys® Trigno™ avec des centrales inertielles sans fils qui est utilisé.

Le premier objectif du présent projet est de développer et d'implémenter un algorithme de fusion de données de centrales inertielles pour obtenir un estimé d'orientation du membre supérieur. Le deuxième objectif est de calculer les angles du membre supérieur lors de l'exécution de 4 exercices de réhabilitations ainsi que la vitesse de mouvement.

Ce rapport présente d'abord un survol du projet de C. Séguin suivi d'une présentation du système Delsys® qui est utilisé pour faire l'acquisition des données et des différents types de capteurs. Ensuite, le filtre Kalman sera expliqué en détail avant de passer à la méthodologie du présent projet. Finalement, les résultats sont présentés et analysés dans la discussion et une courte conclusion revient sur l'ensemble du projet et expose quelques recommandations.

Projet antérieur

Avant d'entamer ce projet, un survol du projet de C. Séguin [30] est nécessaire, ainsi que de son code Matlab.

La première partie de ce projet consiste à trouver une méthode d'estimation des angles du membre supérieur qui donne des bons résultats et qui soit reproductible. Après analyse du travail de El-Gohary et al. [31] et du travail de Luis Antonio Pereira de Lima (2015) (code Matlab disponible en annexe du rapport [30]), elle a conclu qu'une meilleure méthode est l'utilisation des angles d'Euler et un filtre complémentaire pour fusionner les données des capteurs.

Deux méthodes sont également présentées pour calculer l'angle *Yaw*, dont la deuxième a été jugée meilleure. La première se base sur le travail de Z. Tang et al. [32] et utilise un filtre Kalman avec compensation pour la fusion des données. La deuxième méthode utilise les données de l'accéléromètre pour calculer l'angle *Yaw* et un filtre complémentaire pour la fusion des données. Par contre, cette dernière n'utilise pas les données du magnétomètre.

Selon la littérature, il n'est pas possible de calculer l'angle *Yaw* uniquement avec l'accéléromètre. Étant donné que l'accélération gravitationnelle est unidirectionnelle, seulement les angles absolus de *Pitch* et *Roll* sont calculables. Et le gyroscope ne mesure que le taux de variation de cet angle. Sans connaître l'angle initiale, les données du gyroscope limitent le calcul à un angle relatif. Donc, l'utilisation de cette 2e méthode semble erronée.

Après vérification du code de son projet et du protocole d'acquisition, on remarque que la calibration des capteurs n'a pas été effectuée. Sans celle-ci, les données des capteurs présentent des erreurs majeures, particulièrement en ce qui concerne le magnétomètre. C'est ce manque dans le protocole qui semble expliquer l'inefficacité de la première méthode utilisée pour le calcul de l'angle *Yaw*.

Puisque l'acquisition des données de certains patients a déjà été faite, il est important de trouver une solution pour calibrer les données (post-acquisition) du magnétomètre et les inclure dans l'algorithme d'estimation des angles du membre supérieur.

Calibration post-acquisition

Le processus de calibration consiste à calculer les paramètres qui permettront d'ajuster les valeurs acquises par les capteurs pour correspondre le plus possible à une référence ou à une valeur connue. Par exemple, tenir une centrale inertielle immobile et à plat (axe des z vers le haut) devrait permettre d'obtenir une valeur de zéro pour le gyroscope (les 3 axes) et une valeur de 1 g pour l'axe des z et de zéro pour les 2 autres axes de l'accéléromètre. Ces paramètres sont par la suite utilisés pour corriger les données qui sont acquises avant de les afficher. Il est donc possible de faire la calibration ultérieurement, en autant qu'on puisse retrouver les mêmes conditions d'acquisition pour le calcul des paramètres et que l'acquisition inclut des données qui ont une référence connue.

Calibration du magnétomètre

Dans le cas du magnétomètre, la référence est le champ magnétique terrestre. Celui-ci est connu et des cartes indiquant les valeurs nécessaires pour décrire l'orientation et l'amplitude du champ magnétique terrestre en tout point sont facilement disponibles en ligne. Ces valeurs incluent les angles d'inclinaison et de déclination ainsi que son amplitude. Puisque l'intérêt présent est de connaître la direction du champ magnétique et l'orientation de la centrale inertielle par rapport à celle-ci, seules les valeurs des angles sont nécessaires. L'amplitude du champ peut être normalisé à 1.

Dans le cas de ce projet, l'endroit où a lieu les acquisitions, soit l'école Victor-Doré, l'angle de déclinaison est de 14°23.76' ouest et l'angle d'inclinaison est de 70.03°, selon [33] et [34]. Ces valeurs changent régulièrement. Il faut donc les vérifier occasionnellement et remettre à jour la calibration si nécessaire.

Une recherche de la littérature a permis de trouver les travaux de Kok et al. ([35], [36], et [37]) sur la calibration du magnétomètre. L'algorithme développé formule le problème de calibration en tant que problème d'estimation de l'orientation du capteur en présence de paramètres inconnus (θ). Si l'orientation du capteur est connue (orientation de référence et paramètres du champ magnétique terrestre) il est possible de faire correspondre les données acquissent aux données de référence. Les données de l'accéléromètre et du gyroscope sont utilisées pour aider dans la calibration du magnétomètre. Cette formulation nécessite la résolution d'un problème d'optimisation non convexe pour obtenir l'estimation de vraisemblance maximale (*Maximum Likelihood* ou ML) des paramètres de calibration.

Modèle du magnétomètre

Le modèle idéal du magnétomètre est décrit par l'équation suivante :

$$\mathrm{h_{m,t}} = \mathrm{R_t^{bn} m^n} \tag{1}$$

Où $h_{m,t}$: vecteur des mesures du magnétomètre
R_t^{bn} : matrice de rotation représentant l'orientation entre les 2 systèmes de coordonnées
m^n : vecteur du champ magnétique local normalisé

Cette équation ne tient pas compte des perturbations causées par la présence de matériaux ferromagnétiques ou des appareils électromagnétiques à proximité. Ces perturbations ont deux types d'effets sur le champ magnétique : le décalage dû à une magnétisation permanente des matériaux ferromagnétiques (*hard iron effect*) qui est représenté par le vecteur $\mathrm{o_{hi}}$, et la déformation due à un champ magnétique externe (*soft iron effect*) qui est représentée par la matrice $\mathrm{C_{si}}$.

L'équation ne tient pas compte non plus de la présence d'erreurs des capteurs, soit la non-orthogonalité entre les 3 axes ($\mathrm{C_{no}}$), la présence d'un décalage à zéro ($\mathrm{o_{zb}}$), la différence de sensibilité des 3 axes ($\mathrm{C_{sc}}$), et le désalignement entre le magnétomètre et les autres capteurs ($\mathrm{R^{b_m b}}$).

Alors, en tenant compte de la présence de toutes ces perturbations et erreurs, le modèle du magnétomètre s'écrit comme suit :

$$\mathrm{h_{m,t} = C_{sc}C_{no}(C_{si}R^{b_m b}R_t^{bn}m^n + o_{hi}) + o_{zb}} \tag{2}$$

En distribuant et en combinant les matrices, il est possible de décrire les perturbations par une matrice de distorsion de 3x3 (D) et un vecteur de décalage de 3x1 (o).

$$\mathrm{D = C_{sc}C_{no}C_{si}R^{b_m b}} \tag{3}$$

$$\mathrm{o = C_{sc}C_{no}o_{hi} + o_{zb}} \tag{4}$$

L'équation représentant les mesures du magnétomètre peut maintenant s'écrire comme suit :

$$\mathrm{y_{m,t} = DR_t^{bn}m^n + o + e_{m,t}} \tag{5}$$

Où $e_{m,t}$ représente l'erreur de mesure (bruit gaussien) du magnétomètre.

Il est important de noter que 2 hypothèses sont formulées pour dériver le modèle du magnétomètre.

1. La matrice D et le vecteur o sont constants et ne varie pas dans le temps.
2. Le champ magnétique local m^n est constant

Modèles du gyroscope et de l'accéléromètre

Les modèles du gyroscope (6) et de l'accéléromètre (7) sont modelés comme suit.

$$\mathrm{y_{\omega,t} = \omega_t + \delta_\omega + e_{\omega,t}} \tag{6}$$

$$\mathrm{y_{a,t} \approx -R_t^{bn}g^n + e_{a,t}} \tag{7}$$

Où ω_t : vecteur des rotations angulaires
δ_ω : vecteur des décalages du gyroscope
$e_{\omega,t}$: erreur de mesure (bruit gaussien) du gyroscope
$\mathrm{g^n}$: accélération gravitationnelle
$e_{a,t}$: erreur de mesure (bruit gaussien) de l'accéléromètre

Formulation du problème

Le problème de calibration du magnétomètre est formulé comme un problème de vraisemblance maximale (ML). Les paramètres θ sont trouvés en maximisant la probabilité d'obtenir l'orientation du magnétomètre. Ce problème peut également être posé en minimisant l'erreur entre l'estimé de l'orientation obtenu à partir des paramètres et l'orientation recherchée. Le problème d'optimisation s'écrit comme suit :

$$\min_{\theta \in \Theta} \quad \frac{1}{2}\sum_{t=1}^{N} \left|\left|y_t - \hat{y}_{t|t-1}(\theta)\right|\right|^2_{S_t^{-1}(\theta)} + \log\det S_t(\theta) \tag{8}$$

Qui peut être résolu pour les paramètres inconnus (θ). Mais puisque ce problème est non convexe, des bonnes valeurs initiales pour θ sont nécessaires.

En plus de la matrice de distorsion (D) et le vecteur de décalage (o), le champ magnétique local m^n est en général inconnu dans un environnement intérieur. Ensuite, les mesures du gyroscope (6) sont corrompues par un décalage δ_ω et doit être considérée comme un inconnu. Finalement, les matrices de covariances Σ_ω, Σ_a, et Σ_m sont également inconnues. La liste des paramètres se résume donc à :

$$D \in \mathbb{R}^{3x3} \tag{9}$$
$$o \in \mathbb{R}^3 \tag{10}$$
$$m^n \in \{\mathbb{R}^3 : \left|\left|m^n\right|\right|_2^2 = 1, m_x^n > 0, m_y^n = 0\} \tag{11}$$
$$\delta_\omega \in \mathbb{R}^3 \tag{12}$$
$$\Sigma_\omega \in \{\mathbb{R}^{3x3} : \Sigma_\omega \succeq 0, \Sigma_\omega = \Sigma_\omega^T\} \tag{13}$$
$$\Sigma_a \in \{\mathbb{R}^{3x3} : \Sigma_a \succeq 0, \Sigma_a = \Sigma_a^T\} \tag{14}$$
$$\Sigma_m \in \{\mathbb{R}^{3x3} : \Sigma_m \succeq 0, \Sigma_m = \Sigma_m^T\} \tag{15}$$

Pour réduire le nombre de paramètres, il est possible de décrire les matrices (13) à (15) par leur factorisation de Cholesky ce qui réduit à 6 paramètres chaque matrice. De plus, en choisissant un système de coordonnées de façon à ce que l'axe des x pointe vers le nord magnétique, il est possible de représenter (11) par sa composante verticale m_z^n (16) ou en considérant l'angle d'inclinaison φ (17), ce qui réduit sa représentation à un seul paramètre.

$$m^n = (\sqrt{1-(m_z^n)^2} \;\; 0 \;\; m_z^n)^T \tag{16}$$

$$m^n = (\cos\varphi \;\; 0 \;\; -\sin\varphi)^T \tag{17}$$

En tout, le vecteur θ est représenté par 34 paramètres.

Algorithme de calibration

Tel que mentionné, il est important d'avoir des bonnes conditions initiales des paramètres pour obtenir une bonne optimisation. La première étape de l'algorithme consiste donc à déterminer les valeurs initiales des paramètres.

Pour les paramètres de (12) à (15), l'acquisition d'une série de données lorsque les capteurs sont immobiles est suffisante pour calculer les valeurs initiales. Il est également possible d'utiliser les connaissances antérieures des capteurs pour déterminer les valeurs initiales de ces paramètres. Ensuite, une optimisation ellipsoïdale en utilisant les données du magnétomètre est effectuée pour déterminer

les valeurs initiales des paramètres (9) et (10). Ceci fait correspondre les données ellipsoïdales provenant du magnétomètre sur une sphère unitaire, mais ne peut déterminer la rotation R_D de la sphère. En utilisant la propriété du produit scalaire entre deux vecteurs qui demeure constant même après une rotation, il est possible de trouver la rotation de la sphère. Les 2 vecteurs considérés sont m^n et le vecteur $v^n = (0\ 0\ 1)^T$.

$$m_z^n = m^n v^n \tag{18}$$

Alors, on suppose que le produit scalaire de ces vecteurs dans le système de coordonnées des centrales inertielles est également constant. En appliquant une rotation R_t^{bn} pour passer des coordonnées globales (n) vers les coordonnées locales (b), on obtient.

$$v_t^b = R_t^{bn} v^n \quad \text{et} \quad m_t^b = R_t^{bn} m^n \tag{19}$$

Et en utilisant l'équation (5), on peut décrire le champ magnétique dans les coordonnées locales comme suit.

$$m_t^b = R_D^T D_0^{-1}(y_{m,t} - \hat{o}_0) \tag{20}$$

Ce qui conduit au problème de minimisation suivant :

$$\min_{R_D, m_{z,0}^n} \quad \frac{1}{2}\sum_{t=1}^{N} ||m_{z,0}^n - (v^n)^T R_t^{nb} R_D^T D_0^{-1}(y_{m,t} - \hat{o}_0)||_2^2 \tag{21}$$

$$\text{tel que} \quad R_D \in SO(3)$$

Ce qui permet d'obtenir les valeurs initiales de (9) et de (11).

La deuxième partie de l'algorithme consiste à optimiser les valeurs des paramètres. Tout d'abord, un filtre Extended Kalman (EKF), sans les données du magnétomètre qui ne sont pas encore calibrées, est utilisé pour estimer l'orientation des capteurs. Le filtre EKF donne comme résultat une estimation de l'orientation $\hat{y}_{t|t-1}(\theta)$ pour tout temps t, ainsi qu'une estimation de la matrice de covariance de l'état, $S_t(\theta)$. Ces estimations sont ensuite utilisées pour évaluer la fonction (8). Qui est utilisée pour déterminer les nouvelles valeurs des paramètres θ en calculant le gradient numérique de cette fonction.

Les étapes de la deuxième partie de l'algorithme sont répétées jusqu'à convergence des résultats.

L'algorithme complet est disponible dans l'article de Kok et al. [35]. Pour tous les détails veuillez-vous référer à l'article.

Résultats de la calibration

Malgré les multiples tentatives, il n'a pas été possible d'obtenir des résultats acceptables avec cette méthode. La Figure 1 montre un exemple des résultats recherchés. Ces résultats n'ont pas été obtenu. Il est donc évident que suite à la calibration, les données du magnétomètre n'ont pas été calibrées de façon à obtenir une sphère unitaire.

Plusieurs raisons peuvent expliquer l'échec de la calibration des données du magnétomètre post-acquisition. Tout d'abord, il est possible que l'implémentation de l'algorithme sur Matlab ne soit pas conforme au travail de Kok et al. [35]. Après plusieurs mois à étudier l'article et les différents concepts

nécessaire à la formulation de l'algorithme, je crois avoir assez bien compris la matière pour faire l'implémentation. Une difficulté rencontrée est l'estimation des valeurs initiales des paramètres, particulièrement l'optimisation de la fonction (21). Le travail de Kok et al. [36] a fait la démonstration de différentes façons de formuler l'orientation de la matrice, que ce soit par la matrice de rotation R_t^{bn}, les quaternions, les angles d'Euler ou bien la déviation par rapport à un point de linéarisation représentée par un vecteur à 3 paramètres. Aucune de ses méthodes de représentations a abouties à de meilleurs résultats.

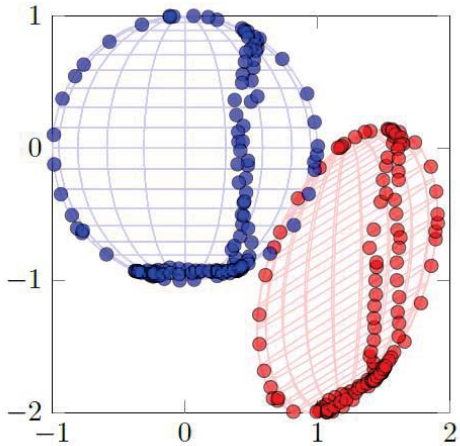


Figure 1 - Exemple de calibration avec données en ellipsoïde avant calibration (rouge) et une sphère unitaire de données après calibration (bleu).

Une autre possibilité nous ramène à aux hypothèses de départ pour la formulation du modèle du magnétomètre. Il est possible que les perturbations magnétiques ne soient pas constantes dans le temps. De plus, la deuxième hypothèse déclare que le champ magnétique local m^n est constant. Compte tenu que le local où a lieu les acquisitions est très limité en espace et qu'il contient plusieurs ordinateurs, le robot REAplan®, ainsi que des cabinets en métal, il est très probable que cette hypothèse soit fausse. La présence de toutes ces perturbations rend la modélisation du magnétomètre ainsi que l'estimation des paramètres de calibration très difficile.

Il a donc été déterminé de poursuivre le projet en utilisant seulement les données du gyroscope et de l'accéléromètre pour déterminer l'orientation des centrales inertielles. Des solutions sont donc nécessaires pour déterminer les angles Yaw et sont étudiées plus loin dans ce rapport.

Calibration de l'accéléromètre

La calibration de l'accéléromètre suit la méthode décrite par [38], qui est une calibration à six paramètres. Les six paramètres correspondre aux trois gains et aux trois valeurs de décalage, un de chaque pour les trois axes de l'accéléromètre. L'équation qui lie les 6 paramètres aux données acquises de l'accéléromètre est la suivante :

$$\mathrm{A_{cal}} = \begin{bmatrix} \mathrm{A_{cal,x}} \\ \mathrm{A_{cal,y}} \\ \mathrm{A_{cal,z}} \end{bmatrix} = \mathrm{WA_{acc}} + \mathrm{V} = \begin{bmatrix} \mathrm{W_{xx}} & 0 & 0 \\ 0 & \mathrm{W_{yy}} & 0 \\ 0 & 0 & \mathrm{W_{zz}} \end{bmatrix} \begin{bmatrix} \mathrm{A_{acc,x}} \\ \mathrm{A_{acc,y}} \\ \mathrm{A_{acc,z}} \end{bmatrix} + \begin{bmatrix} \mathrm{V_x} \\ \mathrm{V_y} \\ \mathrm{V_z} \end{bmatrix} \tag{22}$$

Où A_{acc} représente les valeurs acquises directement de l'accéléromètre avant calibration. La matrice W contient les gains et le vecteur V les valeurs de décalage.

En séparant les l'équation (22) en ses composantes x, y et z on obtient 3 équations avec 6 inconnus. Il est donc nécessaire d'obtenir 6 acquisitions pour résoudre le système d'équations, soit 2 par axe. Les deux acquisitions par axe doivent être fait de façon à obtenir +1 g et -1 g selon l'axe en question. Si $A_{acc,x,1}$ est l'acquisition lorsqu'on obtient +1 g sur l'axe des x et que $A_{acc,x,2}$ est l'acquisition lorsqu'on obtient -1 g, alors l'équation (22) selon sa composante x devient :

$$\mathrm{W_{xx}A_{acc,x,1} + V_x = 1} \qquad \mathrm{W_{xx}A_{acc,x,2} + V_x = -1} \tag{23}$$

En isolant les paramètres, et en répétant l'exercice avec chaque axe, on obtient les équations suivantes :

$$\begin{aligned} W_{xx} &= \frac{2}{\left(A_{acc,x,1} - A_{acc,x,2}\right)} & W_{yy} &= \frac{2}{\left(A_{acc,y,1} - A_{acc,y,2}\right)} & W_{zz} &= \frac{2}{\left(A_{acc,z,1} - A_{acc,z,2}\right)} \\ \mathrm{V_x} &= \frac{-\left(\mathrm{A_{acc,x,1} + A_{acc,x,2}}\right)}{\left(\mathrm{A_{acc,x,1} - A_{acc,x,2}}\right)} & \mathrm{V_y} &= \frac{-\left(\mathrm{A_{acc,y,1} + A_{acc,y,2}}\right)}{\left(\mathrm{A_{acc,y,1} - A_{acc,y,2}}\right)} & \mathrm{V_z} &= \frac{-\left(\mathrm{A_{acc,z,1} + A_{acc,z,2}}\right)}{\left(\mathrm{A_{acc,z,1} - A_{acc,z,2}}\right)} \end{aligned} \tag{24}$$

Les capteurs doivent être placés à plat selon chaque axe pour obtenir ±1 g. Puisque ce n'est pas le cas lors de l'acquisition des exercices, il n'est pas possible de se fier sur les données déjà acquises. Pour ces acquisitions, la calibration de l'accéléromètre se fait post-acquisition avec un essai où les capteurs sont placés à plat sur chacune de ses faces pendant 10 s. Ensuite, un échantillon d'environ 5 s est utilisé pour calculer la moyenne de chaque mesure et ainsi éliminé le bruit. Ce sont ces mesures qui sont finalement insérées dans les équations (24) pour calculer les paramètres de calibration de l'accéléromètre.

Calibration du gyroscope

La calibration du gyroscope est la plus simple. Peu importe l'orientation du capteur, les mesures doivent être égales à zéro lorsque celui-ci est immobile. La calibration peut être effectuée avec la même acquisition que la calibration de l'accéléromètre. Il suffit de calculer la moyenne pour chaque axe lorsque le capteur est immobile et utiliser ces valeurs comme valeurs de décalage pour ajuster les données du gyroscope. Par contre, après plusieurs essais sur différents jours on remarque que les valeurs de décalage varient d'une journée à une autre.

Lors des exercices, il y a un moment dans le cycle de mouvement de l'exercice de cible où le patient ne bouge pas alors que le patient attend le « GO » pour repartir le prochain mouvement. Il a donc été décidé de calculer la valeur de décalage à ce moment-là. Il y a en tout une dizaine de ces moments dans l'acquisition de l'exercice de cible. La moyenne de ces moments est utilisée comme valeur de décalage, et ce pour chaque axe.

$$\mathrm{U_x = moyenne\left(G_{gyr,x}\right)} \tag{25}$$

La calibration du gyroscope se fait donc en utilisant les données déjà acquises. Les équations qui relient les valeurs de décalages, pour chaque axe, et les données du gyroscope sont les suivantes :

$$\begin{aligned} G_{cal,x} &= G_{gyr,x} + U_x \\ \mathrm{G_{cal,y} = G_{gyr,y} + U_y} \\ G_{cal,z} &= G_{gyr,z} + U_z \end{aligned} \tag{26}$$

Système Delsys®

Le système Delsys® Trigno™ est composé de centrales inertielles qui contiennent chacun un accéléromètre (ACC) triaxial, un gyroscope (GYR) triaxial et un magnétomètre (MAG) triaxial. Chaque capteur peut donc prendre des mesures tridimensionnelles, soit l'accélération linéaire et gravitationnelle (gravité), la vitesse rotationnelle (degrés/s) autour de chaque axe et la force magnétique (μT). Les centrales inertielles contiennent également une électromyographie (EMG) mais dans le cadre de ce projet, il n'est pas utilisé.

Magnétomètre

Le magnétomètre fonctionne selon le principe de l'effet de Hall ce qui conduit à la génération d'un différentiel de voltage en présence d'un champ magnétique. Avec un magnétomètre triaxial, on peut donc mesurer l'amplitude et calculer la direction du champ magnétique local, qui inclut le champ magnétique terrestre ainsi que tout champ magnétique produit par les matériaux ferromagnétiques et les appareils électroniques à proximité. On cherche surtout à mesurer la direction du champ magnétique terrestre pour nous aider à préciser la direction de l'orientation de la centrale inertielle, communément appelé le *heading* en anglais ou l'angle *Yaw*.

Puisque le magnétomètre est très sensible aux perturbations magnétiques, il est important qu'il soit calibré régulièrement pour l'environnement dans lequel il est utilisé. Il est nécessaire de connaître l'angle de déclination et l'angle d'inclinaison du champ magnétique terrestre. Le premier est l'angle entre le nord géographique et le nord magnétique et varie d'un endroit à un autre. Le deuxième est l'angle entre le plan horizontal et la direction du champ magnétique. Cette valeur varie entre -90° (pôle sud magnétique) et +90° (pôle nord magnétique), et est égal à 0° à l'équateur magnétique. Ces informations, représentées sous forme de cartes magnétiques, sont actuellement disponibles en ligne. Les données obtenues du magnétomètre sont 3 vecteurs, un par axe, en microtesla (µT).

Accéléromètre

L'accéléromètre fonctionne selon le principe de l'effet piézoélectrique et donc génère une charge électrique en présence d'une contrainte mécanique. Dans ce cas, cette contrainte est générée par une masse en accélération. En mesurant cette charge électrique dans chaque direction, on peut calculer la direction de déplacement linéaire de la centrale inertielle. L'accélération gravitationnelle, qui est unidirectionnelle, a le même effet sur l'accéléromètre et peut nous aider à calculer les angles d'inclinaison de la centrale inertielle, soit les angles *Roll* et *Pitch*.

Par contre, puisque l'accéléromètre ne fait pas la différence entre une accélération linéaire et l'accélération gravitationnelle, une accélération linéaire trop grande peut induire des erreurs dans le calcul des angles d'inclinaison. De plus, l'accéléromètre est très sensible aux vibrations. Les données obtenues de l'accéléromètre sont 3 vecteurs, un par axe, en unité de gravité (g), où 1 g = 9.81 m/s^2.

Gyroscope

Le gyroscope fonctionne avec l'effet d'accélération Coriolis et l'effet piézoélectrique, et génère une charge électrique en présence d'une contrainte mécanique lorsque la centrale inertielle est en rotation. Le gyroscope triaxial nous permet de mesurer la rotation autour de chaque axe individuellement. Les données obtenues du gyroscope sont 3 vecteurs, un par axe, en degrés par seconde (deg/s). L'intégration nécessaire pour calculer l'angle de rotation à partir des données mesurées ajoute une constante d'intégration qui conduit à une dérive ou décalage, et une erreur, des résultats.

Comparaison Accéléromètre vs Gyroscope

Les limites de l'accéléromètre et du gyroscope ont été brièvement exposées dans les paragraphes précédents. Mais pour bien comprendre l'utilisation du filtre Kalman, voici une comparaison approfondie de ces deux types de capteurs.

Comme mentionné, l'accéléromètre est très sensible aux vibrations ce qui génère des signaux très bruités. De plus, l'impossibilité de différencier entre les différents types d'accélérations (gravitationnelle vs linéaire) introduit d'autres erreurs dans les signaux et les calculs qui en découlent. Par contre, lorsque l'accélération linéaire est minimale, il est possible d'obtenir une estimation de l'orientation absolue de la centrale inertielle avec peu d'erreur en utilisant l'accélération gravitationnelle. Et plus important, l'accéléromètre maintien cette précision à long terme. Mais compte tenu de l'aspect unidirectionnelle de la gravité (vertical), il n'est pas possible de calculer l'angle *Yaw* de l'orientation de la centrale inertielle, également appelé le *heading*.

Pour ce qui est du gyroscope, la plus grande limite est au niveau de l'intégration nécessaire pour calculer l'angle d'orientation de la centrale inertielle. Malgré la calibration et l'ajout d'une période de repos en début d'acquisition pour bien déterminer le décalage, la dérive peut tout de même varier, et en fait s'accumule, en cours d'acquisition lorsque celle-ci dure trop longtemps. Cette durée est estimée à environ 1 ou 2 minutes. Et puisque le gyroscope mesure le taux de variation d'angles, soit la vitesse de rotation angulaire, il n'est pas possible d'obtenir l'orientation absolue de la centrale inertielle en utilisant uniquement le gyroscope. Par contre, si on connait l'orientation de départ et que l'acquisition est de courte durée, le gyroscope génère des mesures précises et robuste avec peu de bruits et ce pour les 3 axes et par conséquent pour les 3 angles d'Euleur, soit le *Yaw*, le *Pitch* et le *Roll*.

En comparant les deux types de capteurs, une certaine complémentarité est visible. Du point de vue de la précision, le gyroscope est précis à court terme et l'accéléromètre à long terme. En matière de robustesse au bruit, le gyroscope l'est et l'accéléromètre ne l'est pas. À partir des mesures de l'accéléromètre il est possible de calculer l'orientation absolue et les mesures du gyroscope donnent une orientation relative.

Il est donc naturel de vouloir combiner les deux séries de données et, en gardant les points forts de chacun, obtenir une nouvelle série de données qui est plus précis et surtout plus fiable. C'est exactement ce que fait le filtre Kalman.

Filtre Kalman

Le filtre Kalman existe depuis les années 1950 mais demeure encore aujourd'hui un des algorithmes les plus utilisés pour la fusion de données. Sa popularité a continué à grandir avec la disponibilité de capteurs compacts de haute qualité et à bas prix. Parmi ses applications il y a le système de positionnement global (GPS), l'industrie aéronautique et plus récemment les drones.

La principale utilité d'un filtre Kalman est de fusionner plusieurs séries de données provenant de sources différentes, avec incertitudes variées, résultant en une série de données plus précise et plus fiable que les données originales.

Le filtre Kalman est un algorithme à deux étapes, la première étape en est une de prédiction et la deuxième de mise-à-jour. C'est un algorithme récursif qui utilise les états actuels et précédents du système, pour calculer le gain Kalman (K) et estimer l'état suivant en fonction des incertitudes de chaque source de données.

Il existe plusieurs variations du filtre Kalman dont les plus utilisées sont sa version originale, la version *extended* (EKF) et la version *unscented* (UKF). La version originale s'applique particulièrement aux systèmes linéaires, tandis que les versions EKF et UKF s'appliquent aux systèmes non-linéaires.

Le plus difficile dans l'implémentation d'un filtre Kalman est l'estimation des matrices de covariances des incertitudes du processus (Q) et des mesures (R). Plusieurs techniques existent pour estimer ces matrices, mais une approche pratique et simple consiste à calculer les covariances des mesures dans des conditions connues, telle que lorsque les capteurs sont immobiles. Dans ce cas, l'accélération et la rotation angulaire des capteurs doivent être à zéro.

Phase de prédiction

La phase de prédiction utilise les équations du modèle d'état du système, (27) et (29), pour calculer les estimations. L'équation (27) inclut les données du gyroscope (ω_k) comme entrées pour estimer le vecteur d'état actuel (x_{k+1}). L'équation (28) utilise la matrice Q, indiquant le niveau d'incertitude du modèle, pour estimer la matrice de covariance d'erreur de l'état actuel (P_{k+1}).

Phase de mise-à-jour

La première partie de la phase de mise à jour consiste à calculer le gain Kalman (K) à partir de la matrice R et de la matrice P, voir équations (30) et (31). C'est une représentation de l'incertitude totale du système incluant toutes les sources de données. Le gain est utilisé dans (32) pour mettre à jour x_{k+1} et dans (33) pour mettre à jour P_{k+1}, en multipliant par la différence e_k (29) entre la mesure (z_k) et l'estimation calculée à partir de l'état actuel. L'amplitude de K varie d'une itération à l'autre. Plus le gain est élevé, plus le filtre ajuste la prédiction pour correspondre avec la mesure lors de la mise à jour. Et plus le gain est faible, plus le filtre fait confiance à la prédiction.

Équations du filtre Kalman

Les équations du filtre Kalman sont résumé ici. L'indice « k+1|k » indique que l'estimation de l'état « k+1 » est basée sur l'état précédent « k ». Tandis que l'indice « k+1|k+1 » indique une mise à jour basée sur l'état actuel « k+1 ».

$$x_{k+1|k} = Fx_{k|k} + B\omega_k \tag{27}$$

$$P_{k+1|k} = FP_{k|k}F^T + Q \tag{28}$$

$$e_k = z_k - Hx_{k+1|k} \tag{29}$$

$$S_k = HP_{k+1|k}H^T + R \tag{30}$$

$$K_k = P_{k+1|k}H^T(S)^{-1} \tag{31}$$

$$x_{k+1|k+1} = x_{k+1|k} + Ke_k \tag{32}$$

$$P_{k+1|k+1} = P_{k+1|k} - KHP_{k+1|k} \tag{33}$$

Où : x_k : vecteur d'état du système
F : matrice de transition de l'état
B : matrice de contrôle des entrées de données
u_k : vecteur de contrôle (gyroscope)
P_k : matrice de covariance de l'incertitude de l'état
Q : matrice de covariance de l'incertitude du processus
z_k : vecteur des mesures (accéléromètre)
H : matrice de transition entre les états et les mesures
R : matrice de covariance de l'incertitude des mesures
K_k : gain Kalman

Version Extended

La version *extended* est simplement une extension du filtre Kalman pour les systèmes non-linéaires. C'est-à-dire que les équations du modèle et/ou de sortie contiennent des fonctions non-linéaires. Alors, au lieu d'utiliser les équations (27) et (29), le EFK utilise les équations suivantes :

$$\mathrm{x}_{k+1|k} = \mathrm{f}\left(\mathrm{x}_{k|k}, \omega_k\right) + \mathrm{e}_{\omega,k} \tag{34}$$

$$\mathrm{e}_k = \mathrm{z}_k - \mathrm{h}\left(\mathrm{x}_{k+1|k}\right) + \mathrm{e}_{a,k} \tag{35}$$

Afin de linéariser les équations à chaque itération, des matrices Jacobiennes sont calculées.

$$\mathrm{F}_k = \frac{\partial \mathrm{f}}{\partial \mathrm{x}_{k|k}} \tag{36}$$

$$\mathrm{H}_k = \frac{\partial \mathrm{h}}{\partial \mathrm{x}_{k+1|k}} \tag{37}$$

Ces matrices sont insérées dans les autres équations de la même façon que les matrices F et H le sont dans la version originale du filtre Kalman. Mise à part ces différences, le EKF est identique à sa version originale.

Méthodologie

Protocole d'acquisition des données

Tel que mentionné, le système sans fil Delsys® Trigno™ est utilisé pour faire l'acquisition des données. Neuf (9) centrales inertielles sont apposées sur un patient et associées à un muscle, voir le Tableau 1 pour la liste des muscles associés au centrales inertielles et la Figure 2 pour la localisation de chaque centrale inertielle. Le patient est ensuite positionné devant le REAplan® et la main du membre supérieur qui est sous étude est attaché au bras du robot.

Chaque patient effectue 4 exercices avec des mouvements cycliques (carré, cercle, ligne droite, cible) avec l'aide du robot REAplan®. Le mouvement cyclique est répété 10 fois pour chaque exercice.

Tableau 1 : Muscles associés aux centrales inertielles

IMU	Muscle	IMU	Muscle	IMU	Muscle
1	Lat. Deltoid	4	Triceps B.	7	Pectoralis M.
2	Biceps B.	5	Brachialis	8	Ant. Deltoid
3	Infraspinatus	6	Brachioradialis	9	Post. Deltoid

Figure 2 - Positionnement des centrales inertielles sur le patient.

Séquence d'orientation : les angles d'Euler ZYX

L'orientation des centrales inertielles et du membre supérieur est décrite par les angles d'Euler *Yaw*, *Pitch* et *Roll*, dans cet ordre. Cette séquence est la plus fréquemment utilisée dans l'industrie aérospatial dans la navigation. La séquence décrit une orientation suite à une combination de 3 rotations, une autour de chaque axe. En commençant par l'axe des z, soit l'angle *Yaw*, aussi appelé le *Heading* ou *Bearing*. Ensuite, une rotation autour de l'axe des y (*Pitch*) et enfin avec une rotation autour de l'axe des x (*Roll*). Ces 2 dernières rotations ensemble constituent l'inclinaison. La Figure 3 montre les angles et leur axe associé. La direction de la flèche indique une rotation positive.

Le seul inconvénient de l'utilisation des angles d'Euler pour décrire l'orientation est la possibilité d'être en position de blocage de cardan ou « *Gimbal Lock* ». C'est lorsqu'on perd un degré de liberté quand le plan perpendiculaire à deux des axes sont coplanaires. Ceci survient par exemple lorsque l'angle *Pitch* est de ±90°. D'autres méthodes pour décrire l'orientation sont disponibles comme les quaternions, la matrice de rotation, et la déviation par rapport à un point de linéarisation représentée par un vecteur à 3 paramètres. Chacun a ses avantages et inconvénients. Le choix de procéder avec les angles d'Euler s'est

fait car cette façon de faire est facile à comprendre et à visualiser. De plus, les contraintes de mouvement du patient rendent presque impossible de se mettre en position de blocage de cardan.

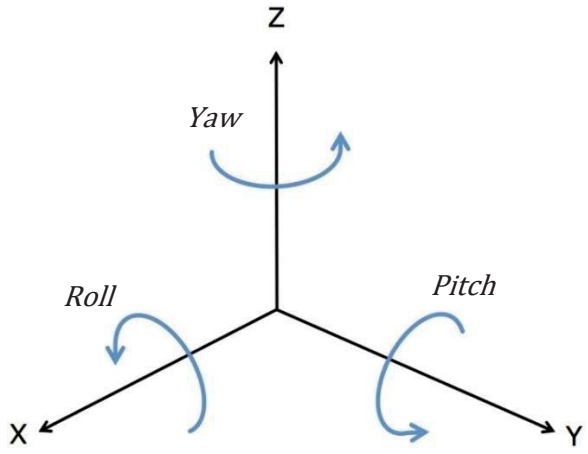


Figure 3 - Représentation des angles d'Euler et leur axe associé.

Systèmes de coordonnées (Delsys® vs Global)

Le système de coordonnées global a été déterminé avec l'axe des x vers l'avant du patient (perpendiculaire et vers l'écran du robot REAplan), l'axe des y vers la gauche et l'axe des z vers le haut. Ce système correspond également au choix de la séquence d'orientation des angles d'Euler ZYX, et par le fait même permet de mieux visualiser la description de l'orientation par ces angles.

Figure 4 - Robot REAplan avec système de coordonnées global. (Droits d'auteur : CUSL / H. Depasse)

Le système de coordonnées des centrales inertielles utilisé est celui déterminé par le manufacturier et est illustré à la Figure 5.

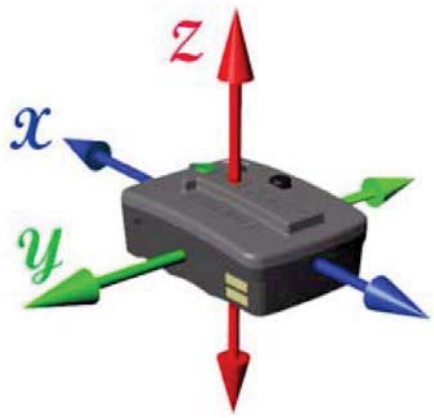


Figure 5 - Système de coordonnées des centrales inertielles de Delsys®

Filtre Kalman

Équations du système d'état

Étant donné les conditions d'acquisition avec des patients qui ont des difficultés motrices, incluant un manque de contrôle musculaire et des mouvements spastiques, on prend les données du gyroscope comme entrée de contrôle pour le modèle d'état du système. Le gyroscope étant plus robuste aux vibrations et aux bruits, il permet d'obtenir de meilleurs résultats. À l'intérieur du filtre Kalman, ces données sont fusionnées avec celles de l'accéléromètre pour réduire l'incertitude de l'acquisition et obtenir de meilleures estimations de l'orientation des centrales inertielles.

Ce qui est recherché est l'orientation des centrales inertielles. Cette orientation est décrite par les angles d'Euler *Yaw*, *Pitch* et *Roll*. Ce sont donc ces trois valeurs qui composent le vecteur d'état x du système.

$$\mathrm{x_k} = [\mathrm{Yaw} \quad \mathrm{Pitch} \quad \mathrm{Roll}]^{\mathrm{T}} \tag{38}$$

Le gyroscope produit des mesures ω_k qui équivalent au taux de variation de rotation (deg/s) autour de chaque axe. Une intégration est nécessaire pour obtenir l'angle (deg) après un temps t (s). La période d'échantillonnage est de T = 0.00675 s. Et les équations du système s'écrivent comme suit :

$$\begin{aligned} Yaw_k &= Yaw_{k-1} + T\omega_{z,k} \\ \mathrm{Pitch_k} &= \mathrm{Pitch_{k-1}} + \mathrm{T}\omega_{\mathrm{y,k}} \\ Roll_k &= Roll_{k-1} + T\omega_{x,k} \end{aligned} \tag{39}$$

Ce qui donne comme équation du système, en forme de modèle d'état :

$$\mathrm{x_k} = \mathrm{Fx_{k-1}} + \mathrm{B}\omega_{\mathrm{k}} \tag{40}$$

Où $\quad F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}, \quad \omega_k = \begin{bmatrix} G_z \\ G_y \\ G_x \end{bmatrix}$

Le vecteur de sortie y_k représente les données mesurées qui servent de comparaison avec l'estimé de l'état du système. Dans ce cas-ci, ce sont les données de l'accéléromètres qui sont les sorties.

$$\mathrm{y_k} = [\mathrm{A_x} \quad \mathrm{A_y} \quad \mathrm{A_z}]^{\mathrm{T}} \tag{41}$$

La relation non-linéaire entre le vecteur d'état et la sortie est donnée par l'équation suivante :

$$\mathrm{y_k} = \mathrm{h}(\mathrm{x_k}) + \mathrm{e_k} \tag{42}$$

Avec le choix de la séquence des angles d'Euler ZYX et selon [39], les fonctions d'observation sont :

$$\mathrm{h}(\mathrm{x_k}) = \begin{bmatrix} -\sin(\mathrm{Pitch_k}) \\ \cos(\mathrm{Pitch_k})\sin(\mathrm{Roll_k}) \\ \cos(\mathrm{Pitch_k})\cos(\mathrm{Roll_k}) \end{bmatrix} \tag{43}$$

Et en calculant le gradient de $h(x_k)$ par rapport avec vecteur d'état, on obtient :

$$\mathrm{H} = \begin{bmatrix} 0 & -\cos(\mathrm{Pitch_k}) & 0 \\ 0 & -\sin(\mathrm{Pitch_k})\sin(\mathrm{Roll_k}) & \cos(\mathrm{Pitch_k})\cos(\mathrm{Roll_k}) \\ 0 & -\sin(\mathrm{Pitch_k})\cos(\mathrm{Roll_k}) & -\cos(\mathrm{Pitch_k})\sin(\mathrm{Roll_k}) \end{bmatrix} \tag{44}$$

Valeurs initiales

Les valeurs initiales du vecteur d'état sont calculées à partir de 2 sources. Pour le *Yaw*, l'estimation vient de la position initiale du patient. Puisque seulement le magnétomètre est en mesure de donner une estimation de l'angle *Yaw*, mais que ces données ne sont pas calibrées, on décide de faire une estimation à partir de la position initiale du patient. Chaque patient est assis devant le robot REAplan® et attaché au bras du robot avec le coude qui repose sur la table. Il n'est donc pas irréaliste que tous les patients aient une position similaire au début de chaque exercice. Pour les angles *Pitch* et *Roll*, l'estimation initiale vient des données de l'accéléromètre qui peuvent être utilisées pour calculer l'angle absolu.

$$\mathrm{Pitch_0} = \tan^{-1}\left(\frac{-\mathrm{A_{x,0}}}{\sqrt{\mathrm{A_{y,0}^2} + \mathrm{A_{z,0}^2}}}\right) \tag{45}$$

$$\mathrm{Roll_0} = \tan^{-1}\left(\frac{\mathrm{A_{y,0}}}{\mathrm{A_{z,0}}}\right) \tag{46}$$

La matrice de covariance P_k représente l'incertitude de l'état du modèle où les valeurs de la diagonale représentent l'incertitude des variables du vecteur d'état. Puisque nous savons très peu sur le système et sa précision pour le moment, il est difficile de déterminer des valeurs justes pour cette matrice. Par contre, un des avantages du filtre Kalman est la rapidité avec laquelle il s'ajuste au fur et à mesure des itérations. Si le système est bien défini, la matrice de covariance P_k est rapidement ajustée pour bien représenter l'incertitude du système. Dans la littérature ([32]), si le système n'est pas très bien connu, il est suggéré de définir la matrice de covariance initiale P_0 en tant que matrice identité.

Dans ce cas-ci, l'angle *Yaw* est une estimation grossière selon la position initiale du patient. L'erreur est estimée à 20° ou ±10°. En se fiant sur la variance de l'accéléromètre, les erreurs sur les angles *Pitch* et *Roll* se calcule à environ 0.0015 deg^2. La matrice de covariance de l'état initiale est la suivante :

$$P_0 = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 0.0015 & 0 \\ 0 & 0 & 0.0015 \end{bmatrix} \tag{47}$$

Estimation des matrices Q et R

La matrice Q représente l'incertitude de notre système, qui est représenté par les mesures du gyroscope. Une méthode d'évaluation du bruit produit par le système est simplement de faire une acquisition lorsque les centrales inertielles sont immobiles et calculer la variance du signal pour chaque axe. C'est ce qui est fait, mais les valeurs obtenues sont légèrement élevées ce qui augmente l'incertitude des données du gyroscope, par rapport à l'accéléromètre. Après plusieurs essais et erreurs, la matrice choisie est la suivante :

$$Q = 0.01 * \begin{bmatrix} \mathrm{var}(G_z) & 0 & 0 \\ 0 & \mathrm{var}(G_y) & 0 \\ 0 & 0 & \mathrm{var}(G_x) \end{bmatrix} \tag{48}$$

La matrice R représente l'incertitude des mesures de l'accéléromètre. La même méthode utilisée pour définir Q est d'abord utilisée pour obtenir la matrice de covariance R. Mais les valeurs obtenues sont très petites, surtout par rapport aux valeurs obtenues pour la matrice Q. Cette différence donne un plus grand poids aux données de l'accéléromètre lors de l'estimation, et ce n'est pas ce qu'on veut. Un ajustement est donc fait pour redéfinir les valeurs de la diagonale de la matrice R. Après plusieurs essais et erreurs, la matrice choisie est la suivante :

$$R = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \tag{49}$$

Acquisition des données

Pré-Calibration

Le système Delsys® possède un logiciel d'acquisition qui inclut un protocole de calibration préacquisition. Lorsque le protocole de calibration est suivi, ceci a pour effet de tenir compte des perturbations magnétiques présents dans l'environnement à proximité, des décalages de l'accéléromètre et du gyroscope, ainsi que du décalage du gyroscope. Les décalages de l'accéléromètre et du gyroscope peuvent varier d'une utilisation à une autre et il va de même pour la présence de perturbations magnétiques. Il est donc très important d'effectuer la calibration du système régulièrement. Selon le manufacturier, il est recommandé de l'effectuer au début de chaque séance ou au début de chaque journée de travail.

Préfiltre

La calibration préacquisition élimine une bonne partie de erreurs du système, mais n'a pas d'effet sur le bruit blanc (gaussien) présent dans les mesures causées par les vibrations. Une fois les données acquises, il est important d'appliquer un filtre passe-bas pour éliminer le plus possible de bruit.

Essais pour déterminer précision

Pour assurer une précision adéquate de l'algorithme, plusieurs essais sont effectués.

Le premier consiste à placer une centrale inertielle à plat, axe des z positif vers le haut, et de faire une rotation de 90° en sens horaire jusqu'à atteindre un tour complet, soit 4 rotations pour atteindre 360°.

Après chaque rotation, faire une pause de 10 secondes. Répéter l'essai avec l'axe des y positif vers le haut et l'axe des x positif vers le haut. Cet essai est utilisé pour évaluer le système de coordonnées des centrales inertielles, c'est-à-dire de s'assurer que les axes sont bien placés. Dans ce cas, l'axe des x doit être parallèle avec la flèche sur la centrale inertielle. L'axe des y pointe vers la gauche lorsque la centrale inertielle est à plat, et l'axe des z pointe vers le haut. Cet essai sert également à évaluer la précision du calcul de l'angle de rotation des centrales inertielles lors d'une rotation autour d'un seul axe.

La deuxième série d'essais consiste à effectuer différents mouvements pour évaluer la précision du calcul des angles recherchés du membre supérieur. En tout, quatre essais sont effectués, soit un pour chaque angle. Chaque essai est développé pour isoler un mouvement en particulier. Les essais sont choisis pour évaluer l'algorithme. Pour ces essais seulement deux centrales inertielles sont utilisées, soit celles pour le biceps (IMU #2) et le brachioradialis (IMU #6). Ces deux centrales inertielles sont celles utilisées dans l'algorithme pour calculer les angles.

Lors de tous les essais, les premiers mouvements sont fait sans pause suivi de plusieurs mouvements avec une pause de 3-4 secondes pour évaluer la stabilité de l'algorithme lorsque le patient bouge peu ou pas du tout.

Calcul des angles du membre supérieur

Pour calculer les angles du membre supérieur, le produit scalaire est utilisé. En considérant deux vecteurs, a et b, on obtient l'angle entre les deux vecteurs en appliquant le produit scalaire.

$$\mathrm{a} \cdot \mathrm{b} = |\mathrm{a}||\mathrm{b}| \cos(\alpha) \tag{50}$$

Ensuite, l'angle α est isolé.

$$\alpha = \cos^{-1}\left(\frac{\mathrm{a} \cdot \mathrm{b}}{|\mathrm{a}||\mathrm{b}|}\right) \tag{51}$$

Le choix des deux vecteurs se fait selon l'angle recherché. Les quatre angles sont la flexion-extension du coude (eFE), l'abduction-adduction de l'épaule (sAA), la flexion-extension de l'épaule (sFE), et la rotation interne-externe de l'épaule (sIE). Les angles recherchés, à l'exception de eFE, sont situés sur un des trois plans anatomiques (Figure 6), soit le plan frontal, le plan sagittal ou le plan transversal.

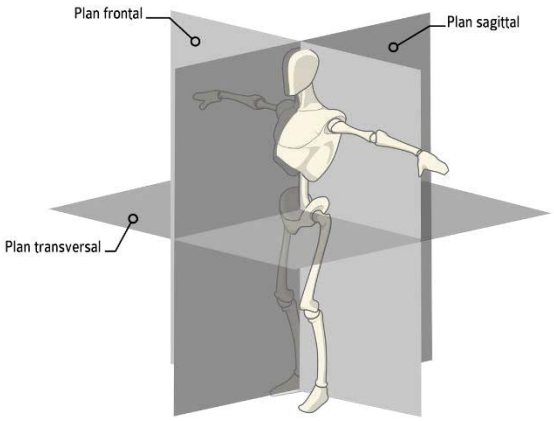


Figure 6 - Plans anatomiques.

Compte tenu de ces informations et des conditions d'acquisition où le patient demeure assis en regardant toujours dans la même direction avec son avant-bras attaché au robot REAplan®, le choix des vecteurs est décrit dans le Tableau 2, ainsi que le plan anatomique associé. Pour calculer l'angle de flexion-extension du coude, les 3 dimensions des vecteurs sont utilisées dans le produit scalaire. Pour les trois autres angles, le calcul se fait en projetant les deux vecteurs sur le plan associé et en appliquant le produit scalaire sur le vecteur à 2 dimensions résultant.

Tableau 2 - Vecteurs et plans associés aux calculs des angles du membre supérieur.

Angle	**Vecteur 1**	**Vecteur 2**	**Plan anatomique**
eFE	Axe x IMU 6 (brachioradialis)	Axe x IMU 2 (biceps)	N/A
sAA	Axe z syst. coord. global	Axe x IMU 2 (biceps)	Frontal
sFE	Axe z syst. coord. global	Axe x IMU 2 (biceps)	Sagittal
sIE	Axe x syst. coord. global	Axe x IMU 6 (brachioradialis)	Transversal

Résultats

Essais pour évaluer la précision

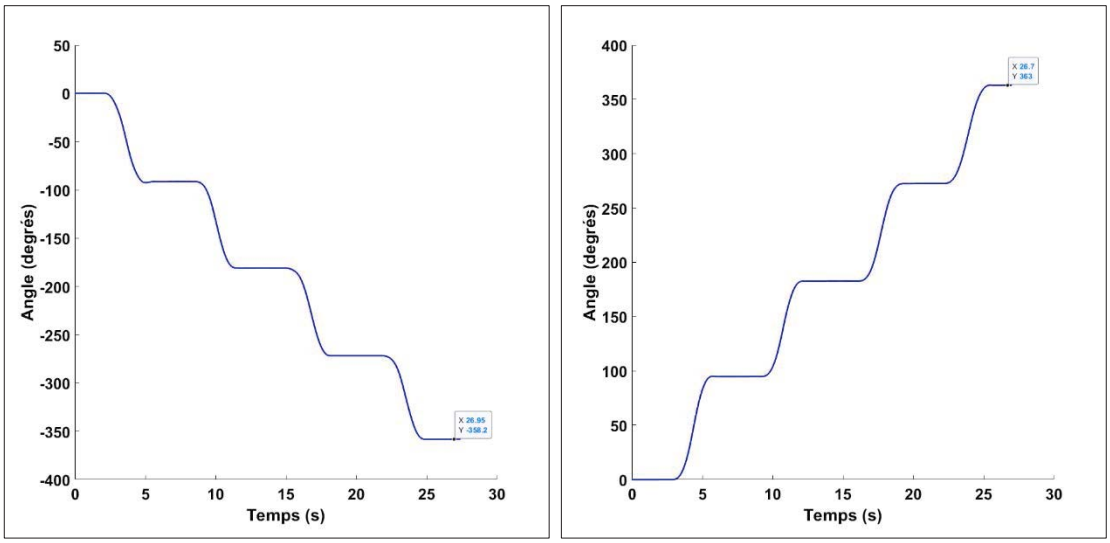
Test de rotation

La Figure 7 montre les résultats de l'angle *Yaw*. Avec l'axe des z positif vers le haut, après une rotation complète (sens horaire) on obtient un angle de -358.2°, et avec l'axe des z négatif vers le haut, on obtient un angle de 363°.

Le tableau suivant résume les résultats des tests de rotation de la Figure 7 à la Figure 9.

Tableau 3 - Résultats des tests de rotation.

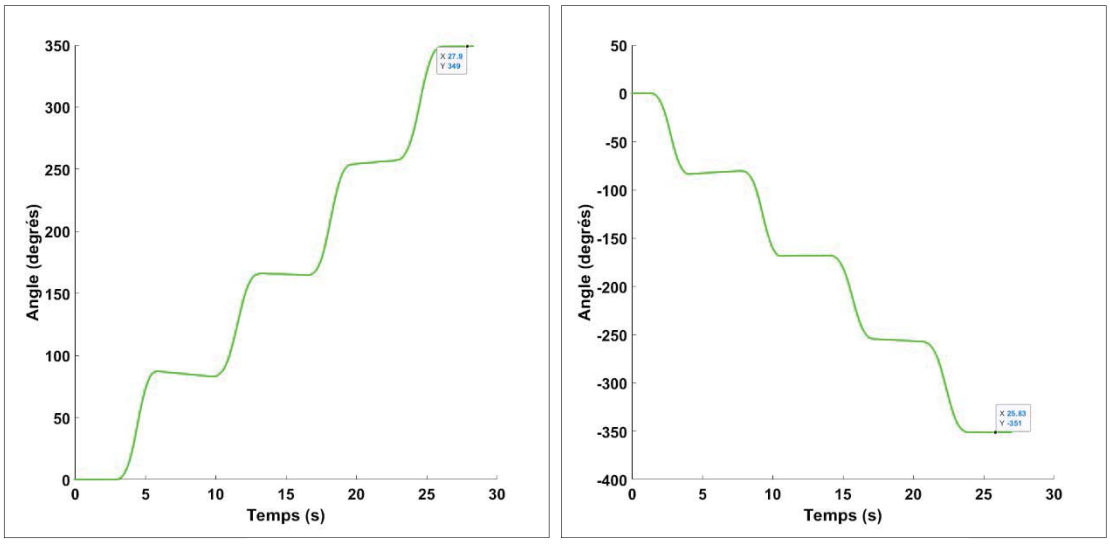
Axe de rotation	Axe positif	Axe négatif	Intervalle d'erreur	Différence des 2 rotations
z	-358.2°	363°	±3°	4.8°
y	349°	-351°	±11°	2°
x	-356.5°	358.8°	±3.5°	2.3°



(a) (b)

Figure 7 - Test de rotation de l'angle Yaw. (a) rotation autour de l'axe des z positif; (b) rotation autour de l'axe des z négatif.

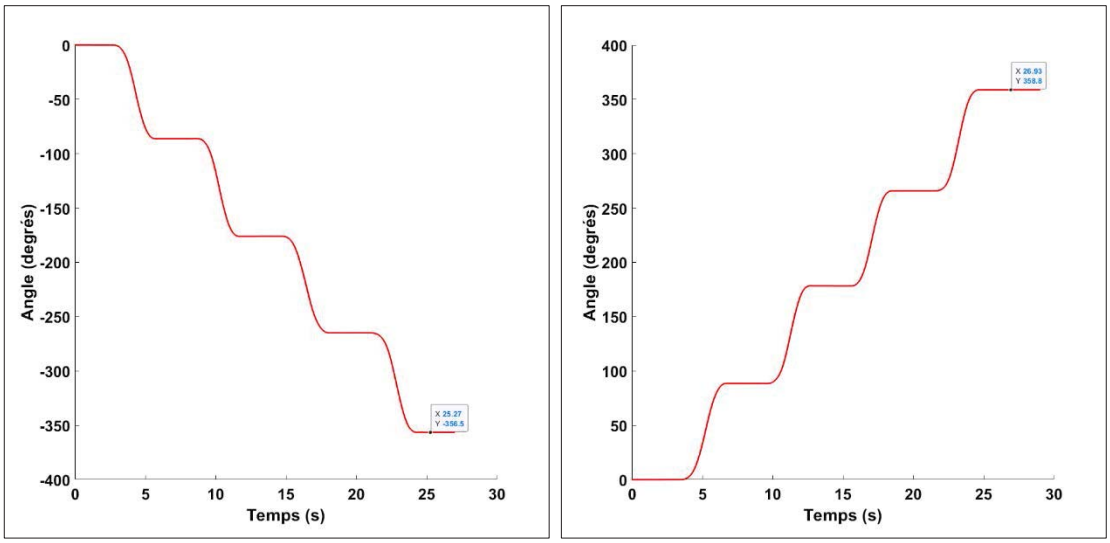
La Figure 8 montre les résultats de l'angle *Pitch*, où on obtient un angle de 349° lorsque l'axe des y positif est vers le haut, et de -351° lorsque l'axe des y négatif est vers le haut.



(a) (b)

Figure 8 - Test de rotation de l'angle Pitch. (a) rotation autour de l'axe des y positif; (b) rotation autour de l'axe des y négatif.

La Figure 9 montre les résultats de l'angle *Roll*. Avec l'axe des x positif vers le haut on obtient un angle de -356.5°, et de 358.8° lorsque l'axe des x négatif est vers le haut.



(a) (b)

Figure 9 - Test de rotation de l'angle Pitch. (a) rotation autour de l'axe des y positif; (b) rotation autour de l'axe des y négatif.

Dans tous essais, on observe que chaque plateau se situe à intervalle d'environ 90°, ce qui est le résultat recherché.

Tests de mouvement

La Figure 10 à la Figure 13 montrent les résultats des tests de mouvement.

Flexion-Extension du coude

L'angle calculé pour cet essai est celui entre la bras et l'avant-bras. L'essai commence avec le bras en pleine extension (180°) le long du corps, suivi de plusieurs flexion-extension (FE) du coude à environ 90°.

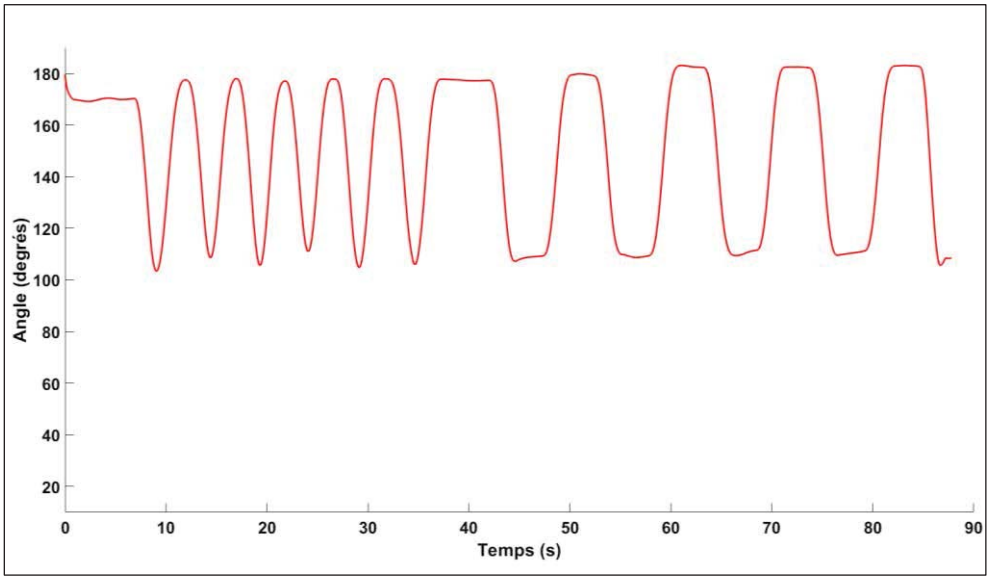


Figure 10 - Test de mouvement Flexion-Extension du coude.

Abduction-Adduction de l'épaule

Cet essai commence avec le bras en pleine extension au niveau du coude et placé le long du corps (0°), suivi de plusieurs abduction-adduction (AA) de l'épaule à 90°.

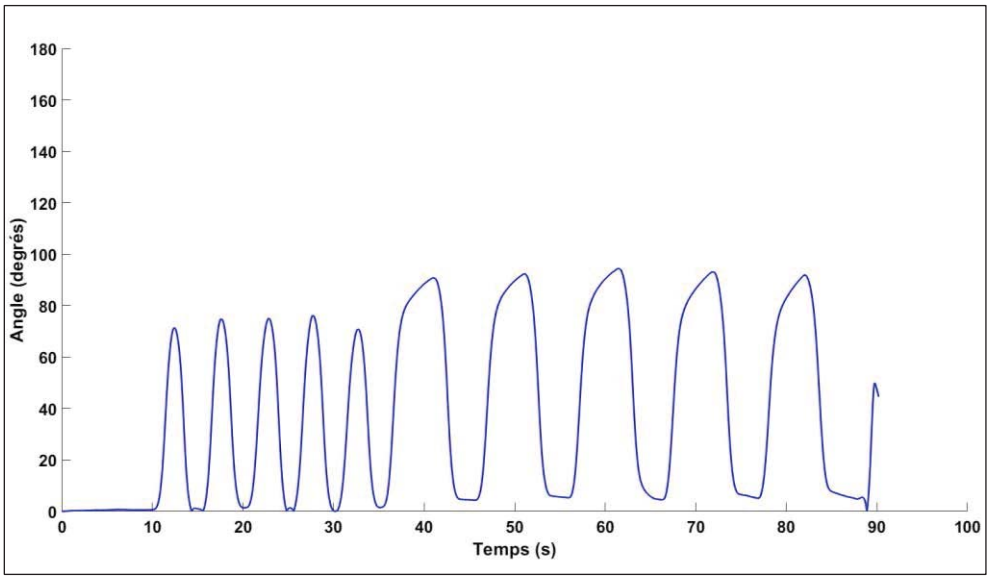


Figure 11 - Test de mouvement Abduction-Adduction de l'épaule.

Flexion-Extension de l'épaule

Cet essai commence avec le bras en pleine extension au niveau du coude et placé le long du corps (0°), suivi de plusieurs flexion-extension (FE) de l'épaule à 90°.

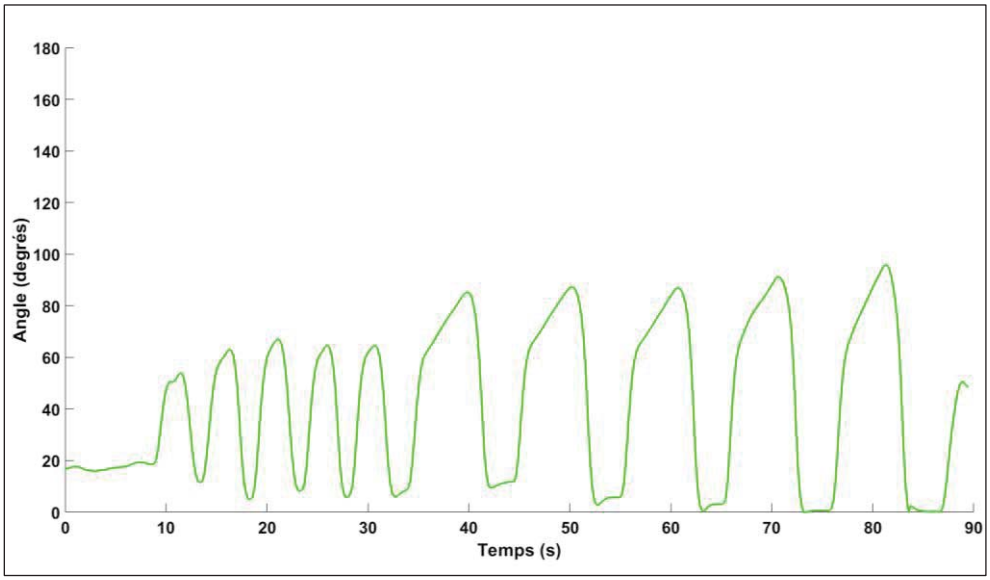


Figure 12 - Test de mouvement Flexion-Extension de l'épaule.

Rotation interne-externe de l'épaule

Cet essai commence avec une flexion du coude de 90° et l'avant-bras vers l'avant, suivi de plusieurs rotation interne-externe (IE) de l'épaule à environ ±45°. Dans ce cas-ci, en plus d'une pause après chaque rotation, une pause additionnelle est ajoutée au point de départ, soit à 0°.

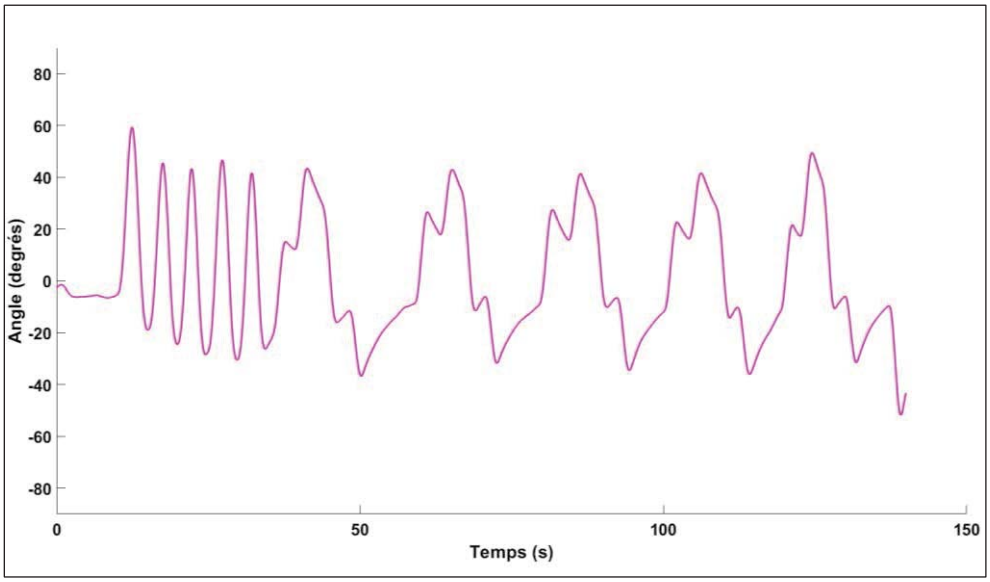


Figure 13 - Test de mouvement Rotation interne-externe de l'épaule.

Patient test

Les résultats qui sont exposés ici proviennent d'un essai avec un patient test qui n'entre pas dans le protocole de sélection pour l'étude. Ce test avec un patient d'âge adulte et sans déficience physique ou mentale est utilisé pour évaluer l'algorithme. Tout d'abord, la Figure 14 à la Figure 17 démontre l'effet du EKF sur les angles d'Euler. Ici on ne montre que les angles *Pitch* (Figure 14 et Figure 16) et *Roll* (Figure 15 et Figure 17), car l'angle *Yaw* est calculé simplement avec le gyroscope et donc le EKF n'a pas d'effet sur son estimation. Les courbes de la Figure 14 et la Figure 15 sont tirées de l'estimation des angles de la centrale inertielle #2 lors de l'exercice cible. Les courbes de la Figure 16 et Figure 17 sont tirées de la centrale inertielle #6. Ces deux centrales sont choisies car ce sont elles qui sont utilisées pour calculer les angles du membre supérieur.

Dans tous les cas, l'estimation à partir des données du gyroscope apparaît en bleu, l'estimation à partir des données de l'accéléromètre en rouge et l'estimation après l'application du filtre *Extended* Kalman apparaît en vert.

Angles d'Euler

La Figure 14 à la Figure 17 montrent les résultats obtenus des angles *Pitch* et *Roll* en comparant l'estimation de l'accéléromètre, du gyroscope et l'estimation obtenue avec le filtre Extended Kalman.

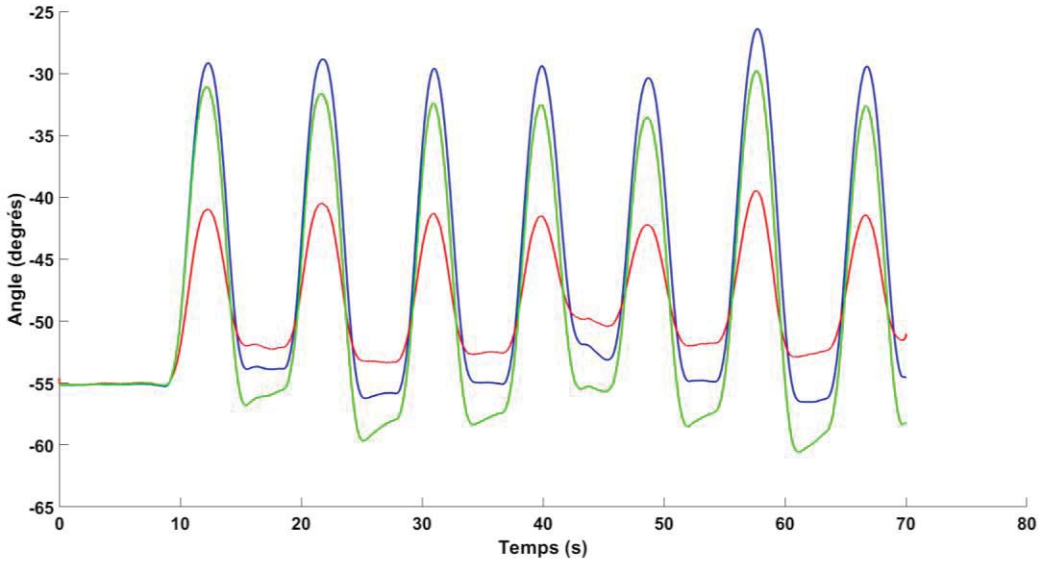


Figure 14 - Comparaison de l'angle Pitch pour IMU #2 pendant l'exercice cible, à partir des données du gyroscope (bleu), accéléromètre (rouge), et l'estimation du filtre EKF (vert).

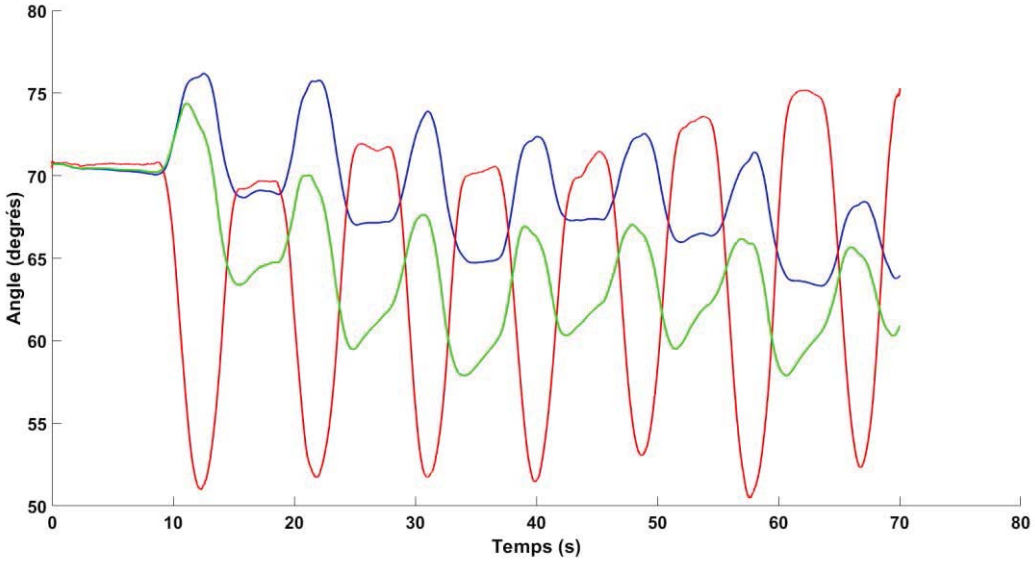


Figure 15 - Comparaison de l'angle Roll pour IMU #2 pendant l'exercice cible, à partir des données du gyroscope (bleu), accéléromètre (rouge), et l'estimation du filtre EKF (vert).

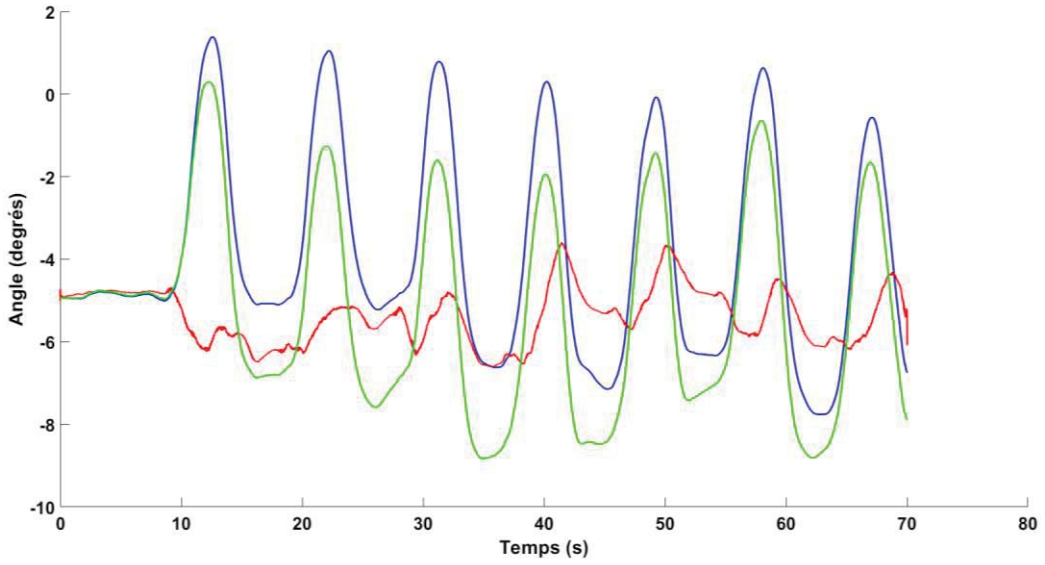


Figure 16 - Comparaison de l'angle Pitch pour IMU #6 pendant l'exercice cible, à partir des données du gyroscope (bleu), accéléromètre (rouge), et l'estimation du filtre EKF (vert).

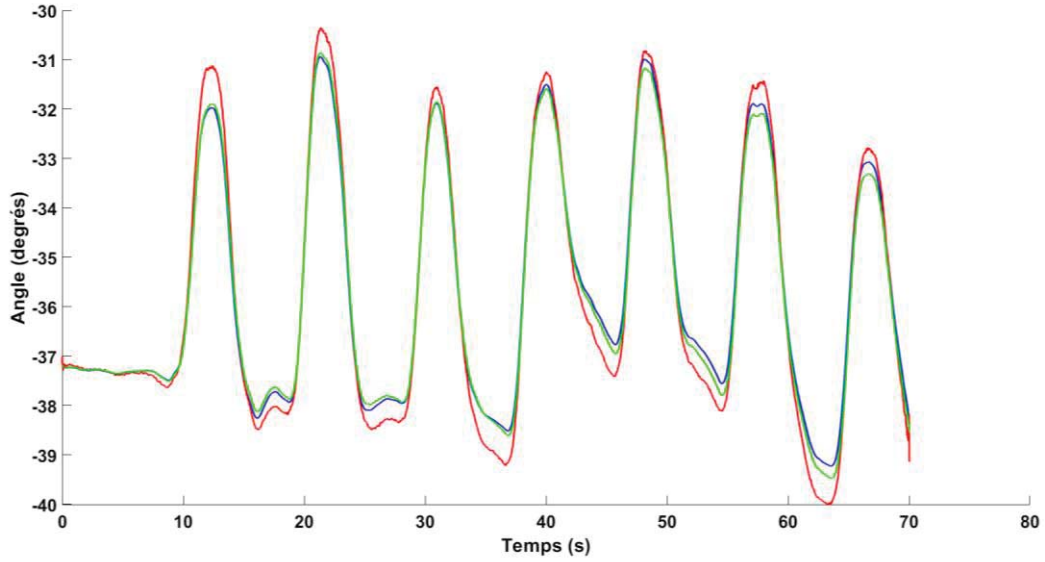


Figure 17 - Comparaison de l'angle Roll pour IMU #6 pendant l'exercice cible, à partir des données du gyroscope (bleu), accéléromètre (rouge), et l'estimation du filtre EKF (vert).

Exercices

La Figure 18 à la Figure 21 montrent les résultats des quatre angles recherchés pour le patient test. Dans chaque figure, on montre une échelle standard de 0° à 180° pour mieux comparer l'amplitude du chaque mouvement séparément entre eux.

Cible

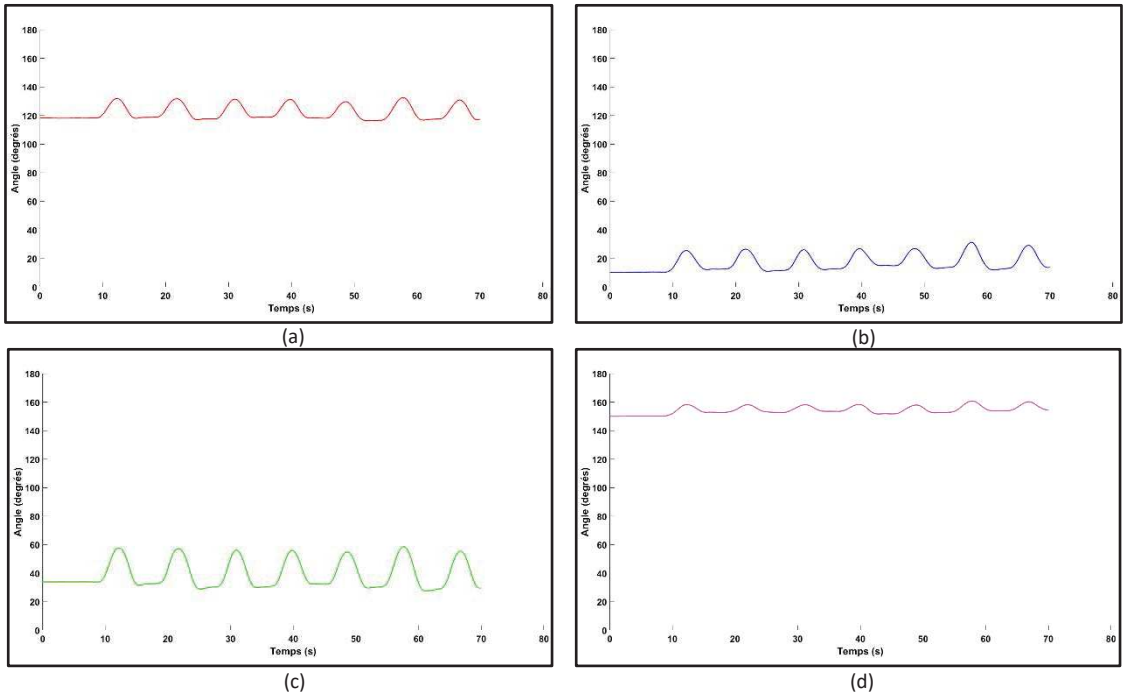


Figure 18 - Angles du membre supérieur pour l'exercice Cible. (a) Flexion-Extension du coude; (b) Abduction-Adduction de l'épaule; (c) Flexion-Extension de l'épaule; (d) Rotation interne-externe de l'épaule.

Carré

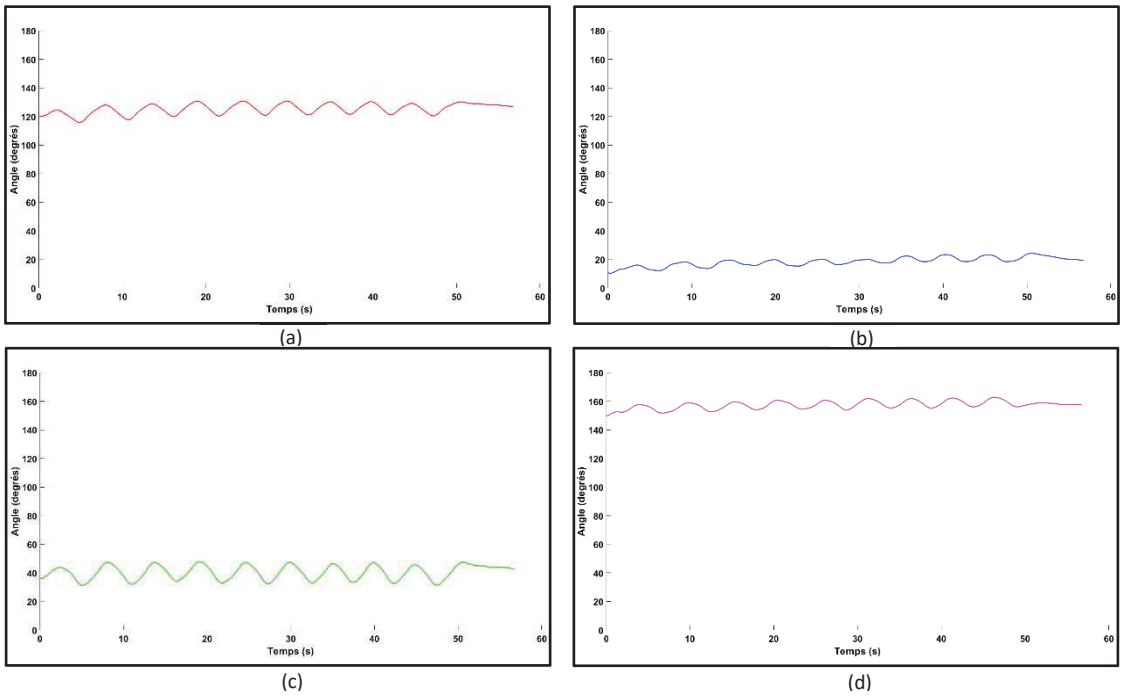


Figure 19 - Angles du membre supérieur pour l'exercice Carré. (a) Flexion-Extension du coude; (b) Abduction-Adduction de l'épaule; (c) Flexion-Extension de l'épaule; (d) Rotation interne-externe de l'épaule.

Cercle

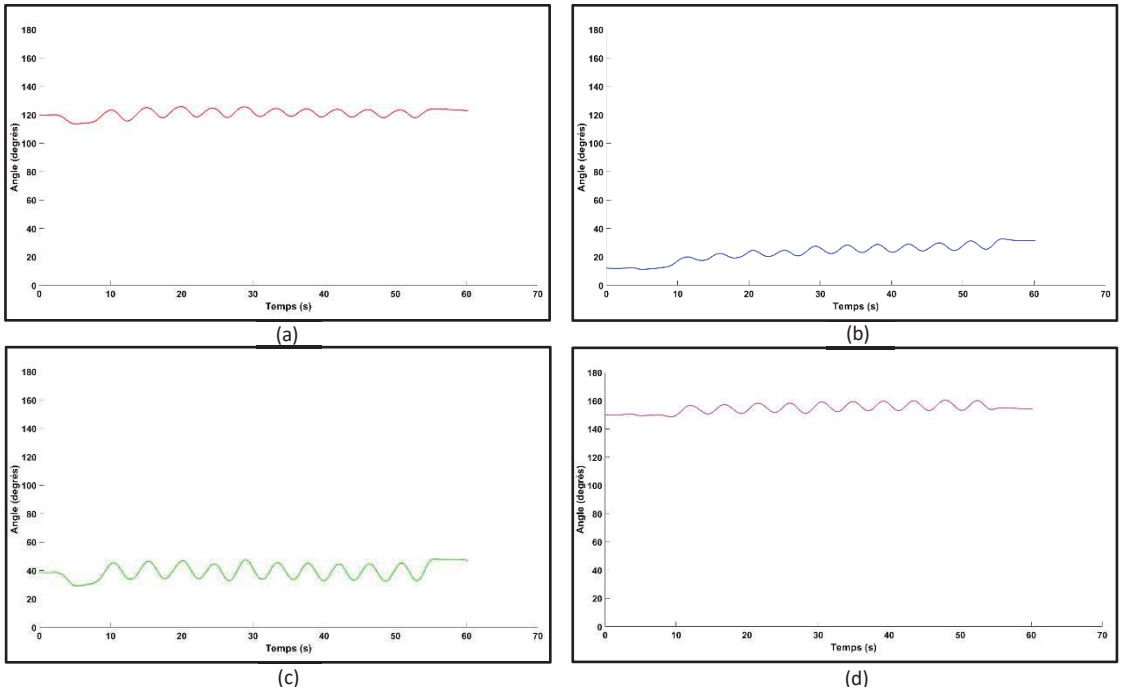


Figure 20 - Angles du membre supérieur pour l'exercice Cercle. (a) Flexion-Extension du coude; (b) Abduction-Adduction de l'épaule; (c) Flexion-Extension de l'épaule; (d) Rotation interne-externe de l'épaule.

Droite

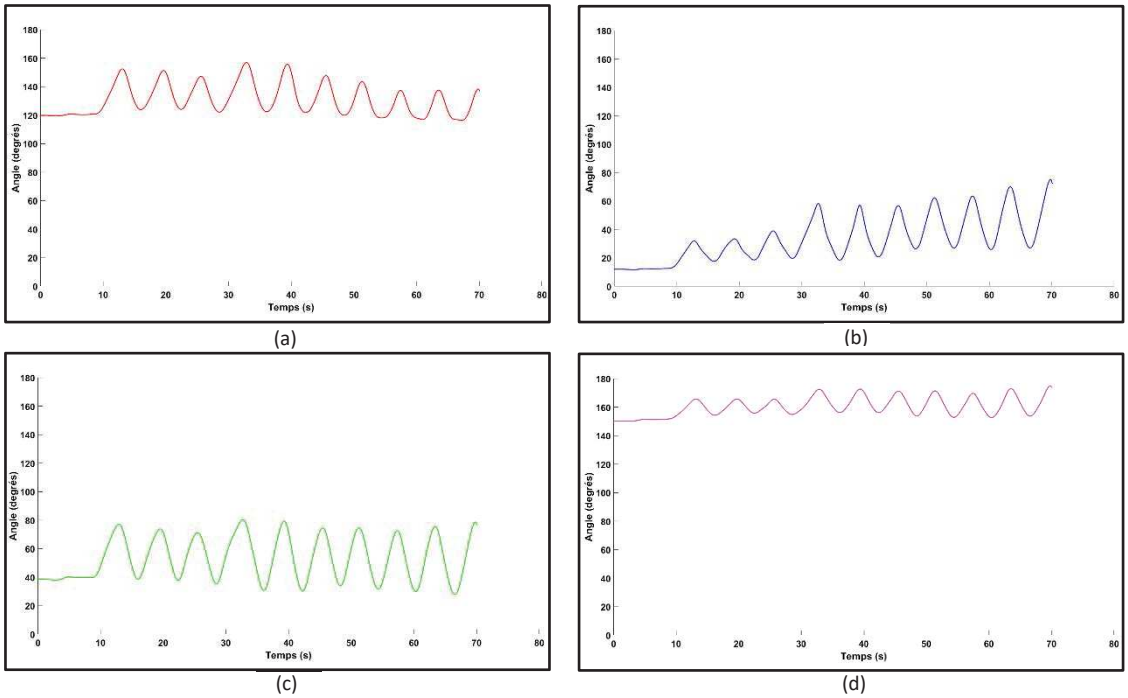


Figure 21 - Angles du membre supérieur pour l'exercice Droite. (a) Flexion-Extension du coude; (b) Abduction-Adduction de l'épaule; (c) Flexion-Extension de l'épaule; (d) Rotation interne-externe de l'épaule.

Discussion

Une fois l'algorithme implanté, la première étape est d'évaluer sa précision. Dans ce projet, avec l'espace limité où a lieu l'acquisition, il est difficile d'installer un autre système pour évaluer l'orientation des centrales inertielles ou du membre supérieur et ainsi obtenir des valeurs de comparaison. De plus, l'élimination des données du magnétomètre pour manque de calibration réduit considérablement notre précision pour l'estimation d'un des trois angles (*Yaw*) pour estimer l'orientation. Par contre, certaines solutions ont permis d'obtenir des résultats très satisfaisants.

D'après les résultats dans la section précédente, on observe tout d'abord une excellente précision dans les essais de rotation. À l'exception des rotations autour de l'axe des y, la précision demeure à l'intérieur de l'intervalle d'erreur de ±3.5°. Dans le cas de l'axe des y, la précision est moins grande avec un intervalle d'erreur de ±11°. Ceci est grandement dû au fait que les centrales inertielles n'ont pas une forme rectangulaire, surtout sur les côtés perpendiculaires à l'axe des y. Ces côtés sont légèrement inclinés, ce qui explique que la rotation est plus petite qu'en réalité. Une partie de la rotation est observée sur l'axe des x qui est en légère rotation dès le départ de l'essai. Par contre, on observe que la différence entre l'erreur de rotation autour de l'axe positif par rapport à l'axe négatif est de 2°, ce qui est similaire à l'intervalle d'erreur des deux autres axes, qui se situe à 4.8° et 2.3°, respectivement pour l'axe des z et l'axe des x. Cette dernière observation sert à déterminer la précision lorsqu'un mouvement de va-et-vient est effectué. Si on part de 0° et qu'on effectue une rotation suivie d'une rotation inverse pour revenir au point de départ, il est raisonnable de penser qu'on obtienne une erreur de moins de 4.8°.

La deuxième série d'essais (Figure 10 à Figure 13) pour évaluer la précision lors de mouvement de bras pour déterminer les angles du membre supérieur a également aboutie à des résultats satisfaisants. Par contre, dans cette série d'essai il n'est pas possible de déterminer une précision. Les mouvements ont été effectués sans points de référence pour s'assurer de faire des rotation précises de 90°. Les résultats sont plutôt évalués de façon visuelle.

Dans tous les cas, on obtient des résultants visuellement satisfaisants autant au niveau de l'amplitude du mouvement que de la stabilité de l'intervalle. C'est-à-dire qu'on observe, par exemple pour la FE du coude, que l'intervalle du mouvement se situe entre 90° et 180°. Dans le cas de l'essai pour le mouvement de rotation interne-externe de l'épaule (Figure 13), on observe une petite dérive lorsque le bras est immobilisé. Ceci est visible dans la deuxième moitié du graphique. Les plateaux devraient être plat, mais on observe bien une dérive qui varie entre 5° et 10° selon la longueur de la pause.

Les résultats du filtre EKF semblent être satisfaisants. Encore une fois, il n'est pas possible d'évaluer la précision des résultats car aucune référence n'est disponible pour effectuer une comparaison. On observe que le filtre EKF réagit bien lorsque les deux séries de données (accéléromètre et gyroscope) sont similaires, comme c'est le cas à la Figure 14 et la Figure 17. À la Figure 16, on observe que l'estimation du EKF conserve l'amplitude des données du gyroscope mais a dévié pour suivre plutôt les résultats de l'accéléromètre. La seule préoccupation est lorsque le calcul des angles à partir des données de l'accéléromètre et du gyroscope ne concordent pas au niveau de la direction de rotation, comme c'est le cas à la Figure 15. Aucune raison n'a été trouvé pour expliquer la divergence entre ces deux résultats. Le filtre donne priorité aux résultats du gyroscope pour ce qui est de la direction de rotation, ce qui est ce qu'on recherche car le gyroscope est supposé avoir une meilleure précision à court terme. Par contre, dans ce cas particulier, on voit qu'il y a une dérive dans l'estimation de l'angle après le filtre EKF.

Les derniers résultats montrent les angles obtenus lors de l'exécution des mouvements pour les quatre exercices, soit la cible, le carré, le cercle et la droite. Ces résultats sont démontrés de la Figure 18 à la Figure 21. On observe que les amplitudes pour les mouvements des exercices du carré et du cercle sont relativement petites. Ce qui est un résultat attendu compte tenu que ces exercices sont restreints au niveau du mouvement. Le contour du carré et du cercle sont relativement petits. De plus, on observe que l'intervalle des angles obtenus demeure constant tout le long du mouvement, à l'exception de l'abduction-adduction du mouvement du cercle qui semble avoir une petite dérive.

Dans le cas de l'exercice de la cible, on observe une plus grande amplitude des angles obtenus, ce qui concorde avec l'exercice car le mouvement que doit effectuer le patient est plus grand. Et cette différence est encore plus marquée pour l'exercice de la droite pour lequel le mouvement est moins restreint par rapport aux autres exercices. Encore une fois, les intervalles, de l'amplitude des angles, demeurent constantes tout le long des exercices, à l'exception de l'angle d'abduction-adduction de l'exercice de la droite. En plus de dériver, l'amplitude semble augmenter au fur et à mesure de l'exercice.

Même si en général les résultats semblent très satisfaisants visuellement et concordent avec les exercices exécutés, il est difficile de confirmer avec beaucoup de précision si ces résultats sont justes. Il est nécessaire et important d'avoir une référence pour comparer les résultats. Mais compte tenu des essais effectués pour valider l'algorithme et déterminer une certaine précision, on est confiant que l'algorithme fonctionne et donne des bons résultats.

Le deuxième objectif incluait le calcul de la vitesse des mouvements. Cette partie des objectifs a été beaucoup plus ardu. Le rapport signal sur bruit était trop faible pour obtenir des résultats satisfaisants. Lors de l'exécution des mouvements, c'est principalement l'avant-bras qui bouge. Par contre, seulement une des 9 centrales inertielles est située sur l'avant-bras et celle-ci est placée à proximité de l'articulation du coude, ce qui réduit sensiblement son mouvement. En conséquence, les données obtenues des capteurs de cette centrale inertielle sont de moins bonnes qualités que celles qu'on obtiendrait d'une centrale inertielle située plus proche de la main qui effectue les plus grands mouvements en amplitude.

Conclusion

L'objectif principal de ce projet était de développer et d'implémenter un algorithme de fusion de données de centrales inertielles pour l'estimation d'orientation du membre supérieur. Après une première analyse du projet et des contraintes auxquelles on devait faire face, la première étape était de trouver une méthode de calibration post-acquisition des données du magnétomètre. Ces données étaient les seules qui permettaient d'obtenir une estimation de l'angle *Yaw*. Sans cet angle, l'estimation de l'orientation devenait plus compliquée. Malgré plusieurs tentatives d'appliquer la méthode de M. Kok et al. ([35]), il n'a pas été possible d'obtenir la calibration du magnétomètre. Après réflexion et compte tenu des conditions d'acquisition et des contraints au niveau des mouvements des patients, il a été décidé d'estimer visuellement les angles *Yaw* initiaux. Les résultats semblent prometteurs malgré tout.

Certains essais sont effectués pour déterminer la précision de l'algorithme et les résultats obtenus sont pour la plupart inférieurs à ±5°. Puisqu'aucune référence n'est disponible pour comparer objectivement les résultats de l'estimation des angles lors des exercices, une analyse visuelle est effectuée. Encore une

fois, les résultats sont très satisfaisants et plus important ces résultats concordent avec les mouvements effectués lors des exercices.

Le deuxième objectif était de calculer les angles du membre supérieur lors de l'exécution de 4 exercices de réhabilitations ainsi que la vitesse de mouvement. La première partie de cet objectif, soit le calcul des angles, était très simple à exécuter une fois le premier objectif atteint. Par contre, il n'a pas été possible de calculer de la vitesse des mouvements. Il est donc suggéré d'ajouter une centrale inertielle au niveau du poignet ou sur la main. Et il n'est pas nécessaire que cette dernière soit attachée directement sur la peau du patient. L'intérêt pour cette centrale inertielle est de mesurer les accélérations avec une meilleure précision car c'est le point qui bouge le plus et qui suit exactement les contours et lignes lors des exercices.

Le manque de référence pour bien comparer les résultats serait un autre ajout important à faire pour la suite du projet. Même s'il est difficile, voire impossible, d'installer un système d'acquisition par caméra comme le système Vicon dans le local où a lieu les acquisitions, il doit exister d'autres options pour mieux déterminer la précision de l'algorithme et des centrales inertielles. Par exemple, un bras robotisé ou mécanique auquel on fixe les centrales inertielles et pour lequel on est capable de déterminer avec précision les angles des différents liens. Ces robots seraient utilisés pour évaluer la précision de l'algorithme de fusion de données.

Au cours de développement de ce projet, une autre suggestion a été faite pour améliorer le protocole d'acquisition. Une pause d'environ 10 s doit être ajoutée au début et la fin de chaque acquisition pendant laquelle le patient demeure immobile. Cette pause sert à bien calibrer le gyroscope qui est le capteur le plus important et le plus fiable pour l'estimation de l'orientation. Il a été possible de calibrer le gyroscope même sans les pauses. Par contre, l'ajout de cette pause simplifierait grandement la calibration et l'algorithme. Il ne serait plus nécessaire d'avoir l'implication du chercheur pour déterminer les intervalles du signal à considérer pour la calibration.

L'objectif principal de ce projet a été atteint. Pour la suite du projet, il reste à implémenter les suggestions décrites ici et ajuster le code Matlab pour inclure les données du magnétomètre calibrées.

Références

[1] W. Weber et E. Weber, Mechanik Der Menschlichen Gehwekzeuge: Eine Anatomisch-Physiologische Untersuchung 1., Göttingen: Dietrich, 1836.

[2] M. Raison, «Portable and clinical tool for dynamic quantification of the muscles forces performed by the human body.». États-Unis Brevet EFS-ID-12883045, 31 05 2013.

[3] R. Crowninshield et R. Brand, «A physiologically based criterion of muscle force prediction in locomotion.,» *J Biomech,* vol. 14, n° %111, pp. 793-801, 1981.

[4] J. Wen*, M. Raison et S. Achiche, «Using a cost function based on kinematics and electromyographic data to quantify muscle forces.,» *Journal of Biomechanics,* vol. 80, pp. 151-158, 2018.

[5] J. Wen*, S. Achiche et M. Raison, «Evaluating muscle forces with a novel cost function that considers co-contractions.,» chez *ECCOMAS Thematic Conference on Multibody Dynamics*, Duisburg, Germany, 2019.

[6] D. Haering*, M. Raison et M. Begon, «Measurement and Description of Three-Dimensional Shoulder Range of Motion With Degrees of Freedom Interactions.,» *Journal of Biomechanical Engineering,* vol. 136, n° %18, p. 84502, 2014.

[7] M. Laitenberger*, M. Raison, D. Perié-Curnier et M. Begon, «Refinement of the upper limb joint kinematics and dynamics using a subject-specific closed-loop forearm model.,» *Multibody System Dynamics,* vol. 33, n° %14, pp. 413-438, 2015.

[8] M. Laitenberger*, M. Raison, G. Guillaume*, D. Périé-Curnier, F. D. Maso* et M. Begon, «Forearm modeling refinement: sensitivity on the upper limb osteo-articular kinematics and dynamics.,» chez *11th World Congress on Computational Mechanics (WCCM)*, Barcelona, Espagne, 2014.

[9] M. Laitenberger*, M. Begon, G. Gaudet*, D. Périé-Curnier, F. D. Maso* et M. Raison, «Impact of the forearm modeling refinement on the upper limb joint kinematics and dynamics.,» chez *3rd Internat. Multibody System Dynamics (IMSD) Conference*, Busan, Corée, 2014.

[10] L. Ballaz*, M. Raison, C. Detrembleur, G. G. * et M. Lemay., «Joint torque variability and repeatability during cyclic flexion-extension of the elbow.,» *BMC Sports Science, Medicine and Rehabilitation,* vol. 8, n° %18, 2016.

[11] G. Gaudet*, M. Raison, F. D. Maso, S. Achiche et M. Begon, «Intra- and intersession reliability of surface electromyography on muscles actuating the forearm during maximum voluntary contractions.,» *Journal of Applied Biomechanics,* vol. 32, n° %16, pp. 558-570, 2016.

[12] F. Dal Maso*, M. Begon et M. Raison, «Methodology to Customize Maximal Isometric Forces for Hill-Type Muscle Models.,» *Journal of applied biomechanics,* vol. 33, n° %11, pp. 80-86, 2017.

[13] S. Raouafi*, S. Achiche, M. Begon, A. Sarcher* et M. Raison, «Classification of upper limb disability levels of children with spastic unilateral cerebral palsy using K-means algorithm.,» *Medical and Biological Engineering and Computing,* vol. 56, n° %11, pp. 80-86, 2017.

[14] A. Sarcher*, M. Raison, M. Lemay, F. Leboeuf, K. Trudel, P. Mathieu et L. Ballaz, «Relation between muscular activity and range of motion during active elbow movement in children with spastic hemiplegia cerebral palsy.,» *Clinical Biomechanics,* vol. 30, n° %11, pp. 86-94, 2015.

[15] A. Sarcher*, M. Raison, R. Gross, L. Ballaz, F. Leboeuf, S. Brochard et B. Perrouin-Verbe, «Distinction between pathological and functional co-activation during active elbow extension in children with spastic hemiplegic cerebral palsy.,» chez *24th Annual Meeting of the European Society of Movement Analysis in Adults and Children (ESMAC)*, Heidelberg, Allemagne, 2015.

[16] J. Clément*, M. Raison et D. Rouleau, «Reproducibility analysis of upper limbs reachable workspace, and effects of acquisition protocol, sex and hand dominancy.,» *Journal of biomechanics,* vol. 68, pp. 58-64, 2018.

[17] J. Clement*, J. Menard, M. Raison, J. Dumais, L. Dubois et D. Rouleau, «Three-dimensional analysis of the locked position in patients with recurrent shoulder instability.,» *Journal of Shoulder Elbow Surgery,* vol. 26, n° %13, pp. 536-543, 2017.

[18] B. Samadi*, S. Achiche, P. P.-L. Cedoz, J. Zaidi, M. Mousny, P. Mahaudens, C. Detrembleur et M. Raison, «Identification of the Cobb angle in adolescents with idiopathic scoliosis based on a non-invasive analysis of gait.,» *Spine,* 2018.

[19] M. Yazji, M. Raison, C. Aubin, H. Labelle, C. Detrembleur, P. Mahaudens et M. Mousny, «Are the mediolateral joint forces in the lower limbs different between scoliotic and healthy subjects during gait?,» *Scoliosis,* vol. 10, n° %1Suppl 2:S3, 2015.

[20] M. Laitenberger, M. Raison, D. Périé-Curnier et M. Begon, «Refinement of the upper limb joint kinematics and dynamics using a subject-specific closed-loop forearm model.,» *Multibody System Dynamics,* vol. 33, n° %14, 2015.

[21] B. Samadi, M. Raison, A. Parent, L. Ballaz et S. Achiche, «Custom sizing of lower limb exoskeleton actuators based on dynamic modeling of the human body: Case study of children with cerebral palsy.,» chez *IEEE international Conference on Technologies for Practical Robot Applications*, Boston, Massachusetts, 2015.

[22] S. Dehem, M. Gilliaux, T. Lejeune, C. Detrembleur, G. D., J. Sapin, M. Vanderwegen et G. Stoquart, «Assessment of upper limb spasticity in stroke patients using the robotic device REAplan.,» *J Rehabil Med.,* vol. 49, n° %17, pp. 565-571, 2017.

[23] M. Gilliaux, T. Lejeune, C. Detrembleur, J. Sapin, B. Dehez, C. Selves et G. Stoquart, «Using the robotic device REAplan as a valid, reliable, and sensitive tool to quantify upper limb impairments in stroke patients.,» *J Rehabil Med. ,* vol. 46, n° %12, pp. 117-125, 2014.

[24] M. Gilliaux, A. Renders, D. Dispa, D. Holvoet, J. Sapin, B. Dehez, C. Detrembleur, T. Lejeune et G. Stoquart, «Upper limb robot-assisted therapy in cerebral palsy: a single-blind randomized controlled trial.,» *Neurorehabil Neural Repair,* vol. 29, n° %12, pp. 183-192, 2015.

[25] «Site Web du REAplan,» Axinesis, 2018. [En ligne]. Available: www.axinesis.com/.

[26] M. Gilliaux, T. Lejeune, J. Sapin, B. Dehez, G. Stoquart et C. Detrembleur, «Age Effects on Upper Limb Kinematics Assessed by the REAplan Robot in Healthy Subjects Aged 3 to 93 Years.,» *Ann Biomed Eng.,* vol. 44, n° %14, pp. 1124-1133, 2016.

[27] D. Varoqui, X. Niu et M. Mirbagheri, «Ankle voluntary movement enhancement following robotic-assisted locomotor training in spinal cord injury.,» *J Neuroeng Rehabil.,* vol. 11, n° %11, p. 46, 2014.

[28] S. Raouafi*, S. Achiche et M. Raison, «Development of a system for the assessment of upper limb motor function impairment using sEMG and IMU sensors.,» *Engineering Applications of Artificial Intelligence,* 2018.

[29] E. Forster et e. al., «Extension of a state-of-the-art optimization criterion to predict co-contraction.,» *J Biomech.,* vol. 14, n° %111, pp. 577-581, 2004.

[30] C. Séguin, «GBM6904 - Projet de maîtrise en ingénierie III,» École Polytechnique de Montréal, Montréal, 2017.

[31] M. El-Gohary, L. Holmstrom, J. Huisinga, E. King, J. McNames et F. Horak, «Upper Limb Joint Angle Tracking with Inertial Sensors,» chez *33rd Annual International Conference of the IEEE EMBS*, Boston, 2011.

[32] Z. Tang, M. Sekine, T. Tamura, N. Tanaka, M. Yoshida et W. Chen, «Measurement and Estimation of 3D Orientation using Magnetic and Inertial Sensors,» *Advanced Biomedical Engineering,* vol. 4, pp. 135-143, 2015.

[33] «Calculateur de déclinaison magnétique,» Gouvernement du Canada, 13 07 2018. [En ligne]. Available: http://geomag.nrcan.gc.ca/calc/mdcal-fr.php. [Accès le 15 05 2018].

[34] «Find the magnetic declination at your location,» Magnetic-declination.com, 2018. [En ligne]. Available: http://www.magnetic-declination.com/. [Accès le 15 05 2018].

[35] M. Kok et T. B. Schon, «Magnetometer calibration using inertial sensors,» *IEEE Sensors Journal,* vol. 16, n° %114, pp. 5679-5689, 2016.

[36] M. Kok, J. D. Holy et T. B. Schon, «Using Inertial Sensors for Position and Orientation Estimation,» *Foundations and Trends in Signal Processing,* vol. 11, n° %11-2, pp. 1-153, 2017.

[37] M. Kok, J. D. Holy, T. B. Schon, F. Gustafsson et H. Luinge, «Calibration of a magnetometer in combination with inertial sensors,» chez *2012 15th International Conference on Information Fusion*, Singapore, Singapore, 2012.

[38] M. Pedley, «AN4399 : High-Precision Calibration of a Three-Axis Accelerometer,» Freescale Semiconductor [Application Notes], Austin, Texas, États-Unis, 2015.

[39] M. Pedley, «AN3461 : Tilt Sensing Using a Three-Axis Accelerometer,» Freescale Semiconductor [Application Notes], Austin, Texas, États-Unis, 2013.

[40] R. Faragher, «Understanding the basis of the Kalman filter via a simple and intuitive derivation,» *Signal Processing Magazine [Lecture Notes],* pp. 128-132, 2012.

[41] M. B. Rhudy, R. A. Salguero et K. Holappa, «A Kalman Filtering Tutorial for Undergraduate Students,» *International Journal of Computer Science & Engineering Survey,* vol. 8, n° %11, pp. 1-18, 2017.

[42] Delsys Incorporated, «Trigno Wireless System User's Guide,» Delsys Incorporated, Natick, Massachusetts, États-Unis, 2013.

Annexe

Code Matlab

REAPlan.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Fusion data algorithm for the estimation of orientation of the upper   %
%  limb using an extended Kalman filter.                                  %
%                                                                         %
%  GBM6903 - Projet de maîtrise en ingénierie III                         %
%                                                                         %
%  Creation: Nelson Giraldo, on 05.07.2018                                %
%  Last update: Nelson Giraldo, on 21.01.2019                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
close all
clc

% Period of acquisition
T = 0.00675;
% Reorganize list of filenames so "cible" file is first
fncible = 0;
while fncible == 0
    prompt = 'File name for target exercise must include "cible".  Press any key when
ready.';
    strSTART = input(prompt,'s');

    filename = uigetfile('*.csv','Select all 4 .csv files for
analysis','multiselect','on');
    fncible = 0;
    nbf = max(size(filename));
    for i=1:nbf
        if contains(filename{i},'cible')
            fncible = i;
            break;
        end
    end
end

fname = cell(size(filename));
fname{1} = filename{fncible};
fnindex = 2;
for j=1:max(size(filename))
    if j ~= fncible
        fname{fnindex} = filename{j};
        fnindex = fnindex + 1;
    end
end

% Extract data and realign coordinate systems
[allIMU,allIMUs] = extractdata(fname{1});
% Calculate calibration offsets for Gyroscope data
[offGx,offGy,offGz,varGx,varGy,varGz,X] = getGYROffsets(allIMUs);
```

```
% Load calibration gains and offsets for Accelerometer data
% Needs to be calibrated seperately
load('Cal_gen_ACC.mat');


%%
% Loop to analyse all 4 csv files containing data from all 4 exercises
for k=1:4
    % Open CSV file containing data on Acc, Gyro and Magà
    % Data for 'Cible' exercise has already been extracted from file
    if k ~= 1
        [allIMU,allIMUs] = extractdata(fname{k});
    end

    % Calculate number of IMUs
    nb_imu = size(allIMUs,2)/6;
    % Number of data points
    n = max(size(allIMUs));
    % Inital estimate of Yaw angle for IMUs
    Yaw0 = [-150;               % Calculated empirically for 1- Deltoid Lat
             165;               % Calculated empirically for 2- Biceps
              75;               % Calculated empirically for 3- Infraspinatus
            -170;               % Calculated empirically for 4- Triceps
             155;               % Calculated empirically for 5- Brachialis
            -150;               % Calculated empirically for 6- Brachioradialis
             135;               % Calculated empirically for 7- Pectoral
            -150;               % Calculated empirically for 8- Deltoid Ant
            -150];              % Calculated empirically for 9- Deltoid Post

    % Create variables
    % Variable initialization
    Angle = zeros(n,3,9);       % Angle vector = [ Yaw Pitch Roll ]
    Pitch = zeros(n,9);
    Roll = zeros(n,9);
    Angle_K = zeros(n,3,nb_imu);% Angle vector post Kalman Filter
    R = zeros(3,3,n,nb_imu);    % Rotation matrix for orientation
    allIMUs_cal = zeros(size(allIMUs));     % Calibrated data matrix

    % Loop to
    for imu=1:nb_imu
        IMU = 6*(imu-1);

        Ax = allIMUs(:,1+IMU);
        Ay = allIMUs(:,2+IMU);
        Az = allIMUs(:,3+IMU);
        Gx = allIMUs(:,4+IMU);
        Gy = allIMUs(:,5+IMU);
        Gz = allIMUs(:,6+IMU);

        % Calibrate Accelerometer data with gain (W) and offsets (V)
        A = W(:,:,imu)*[Ax Ay Az]';
        A(1,:) = A(1,:) + V(1,imu);
        A(2,:) = A(2,:) + V(1,imu);
        A(3,:) = A(3,:) + V(1,imu);
        A = A';

        % Calibrate Gyroscope data with offsets
        Gx = Gx - offGx(imu);
```

```
        Gy = Gy - offGy(imu);
        Gz = Gz - offGz(imu);

        % Calibrated data
        allIMUs_cal(:,1+IMU:6+IMU) = [A Gx Gy Gz];

        % Calculate euler angles of each IMU from GYR data (initial orientation from ACC
data)
        [Angle(:,:,imu),Pitch(:,imu),Roll(:,imu)] =
euler_angles(allIMUs_cal(:,1+IMU:6+IMU),Yaw0(imu),T,varGx(imu),varGy(imu),varGz(imu));

        % Estimate euler angles with Kalman Filter
        Angle_K(:,:,imu) =
EKF(allIMUs_cal(:,1+IMU:6+IMU),Angle(:,1,imu),varGz(imu),varGy(imu),varGx(imu),T);

        % Calculate orientation in matrix form (R) for each IMU
        R(:,:,:,imu) = eul2rotm(Angle_K(:,:,imu),2);
    end

    % Call to function armAngles
    [eFE,sAA,sFE,sIE] = armAngles(R);
    % Save results
    savefname = extractname(fname{k});
    saveresults([savefname,'_calibrated'],{'Ax','Ay','Az','Gx','Gy','Gz'},allIMUs_cal)
    saveresults([savefname,'_angles'],{'eFE','sAA','sFE','sIE'},[eFE,sAA,sFE,sIE])
    if k == 1
        save([savefname,'_Xdata.mat'],'X');
    end
end
```

extractdata.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Extract data from csv file containing acquisition data from Delsys     %
%  Trigno system.                                                          %
%  This function returns raw and smoothed data after reorienting the       %
%  sensor coordinate system to align with the world coordinate.            %
%                                                                          %
%  Creation: Nelson Giraldo, on 18.10.2018                                 %
%  Last update: Nelson Giraldo, on 15.01.2019                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [allIMU,allIMUs] = extractdata(filename)
    allData = importdata(filename,',',1);
    nb_imu = size(allData.data,2)/20;

    for i=1:nb_imu
        IMU = 20*(i-1);

        Ax = allData.data(:,4+IMU);
        Ay = allData.data(:,6+IMU);
        Az = allData.data(:,8+IMU);
        Gx = allData.data(:,10+IMU);
        Gy = allData.data(:,12+IMU);
        Gz = allData.data(:,14+IMU);

        % World (w) coordinate system vs IMU(imu) coordinate system
        % Specific to this projet
        % Zw = Zimu, Yw = Ximu, Xw = -Yimu
        data(:,6*i-5:6*i) = [-Ay Ax Az -Gy Gx Gz];
    end

    % Eliminate all zeros & NAN data
    [allIMU] = eliminateZerosNaN(data);

    % Smooth data curves
    allIMUs = zeros(size(allIMU));
    for t=1:size(allIMU,2)
        allIMUs(:,t) = smooth(allIMU(:,t),200);%0.05,'rloess');
    end
end
```

eliminateZerosNaN.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Eliminate leading and ending zeros/NaN data from acquisition data.     %
%  This function returns a new matrix without zeros or Nan that appear    %
%  across all data vectors.                                                %
%                                                                          %
%  Creation: Nelson Giraldo, on 18.10.2018                                 %
%  Last update: Nelson Giraldo, on 18.10.2018                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [data] = eliminateZerosNaN(allIMU)
% Eliminate all zeros & NAN data
    n = size(allIMU,1);
    debut = 1;

    for i=1:n
        if(any(isnan(allIMU(i,:))))      % returns 1 and continues loop if any element is
                                          a NaN
        elseif(all(allIMU(i,:)))         % returns 1 if all elements are nonzeros
            debut = i;                   % returns row where all elements are nonzeros and
                                          none are NaN
            break
        end
    end
    fin = n;
    for i=n:-1:debut
        if(any(isnan(allIMU(i,:))))      % returns 1 and continues loop if any element is
                                          a NaN
        elseif(all(allIMU(i,:)))         % returns 1 if all elements are nonzeros
            fin = i;                     % returns row where all elements are nonzeros and
                                          none are NaN
            break
        end
    end

    data = allIMU(debut:fin,:);
end
```

getGYROffsets.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Calculate offset vector for Gyroscope data from acquisition file.      %
%  This function returns 7 variables including offset annd variance for   %
%  each axis and each IMU, as well as the intervals used for the          %
%  calculation.                                                            %
%                                                                          %
%  The user is required to select intervals from the figure corresponding %
%  to periods where there is no rotation across all axes.                 %
%                                                                          %
%  Creation: Nelson Giraldo, on 28.09.2018                                 %
%  Last update: Nelson Giraldo, on 15.01.2019                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [mGx,mGy,mGz,varGx,varGy,varGz,Xa] = getGYROffsets(allIMUs)

    nb_imu = size(allIMUs,2)/6;

    % Initialize all variables
    mGx = zeros(nb_imu,1);
    mGy = mGx;
    mGz = mGx;

    varGx = zeros(nb_imu,1);
    varGy = mGx;
    varGz = mGx;

    % Loop through all IMUs
    for i=1:nb_imu
        % Select Gyroscope data
        Gx = allIMUs(:,i*6-2);
        Gy = allIMUs(:,i*6-1);
        Gz = allIMUs(:,i*6);
        % Plot data
        h = figure(i);
        hold on
        plot(Gx,'r')
        plot(Gy,'b')
        plot(Gz,'g')
        legend('Gx','Gy','Gz');
        % Prompt user to select intervals for calibration
        prompt = '\nSelect intervals for gyroscope offsets.\nPress any key to continue.';
        str = input(prompt,'s');

        % Loop through all intervals selected by user
        loop = 1;
        while loop
            % Retrieve input (intervals) from user
            [X,~] = ginput;
            nb = size(X,1);
            % Verify interval data includes two points, beginning and end
            if mod(nb,2) == 0
                m = nb/2;
                tGx = [];
                tGy = [];
                tGz = [];
```

```
            X = round(X);
            % Concatanate data for analysis
            for j=1:m
                tGx = [tGx;Gx(X(j*2-1):X(j*2))];
                tGy = [tGy;Gy(X(j*2-1):X(j*2))];
                tGz = [tGz;Gz(X(j*2-1):X(j*2))];
            end
            % Calculate offset
            mGx(i) = mean(tGx);
            mGy(i) = mean(tGy);
            mGz(i) = mean(tGz);
            % Calculate variance
            varGx(i) = var(tGx);
            varGy(i) = var(tGy);
            varGz(i) = var(tGz);
            loop = 0;
        end
        % Used for tests including only 2 IMUs
        if nb_imu == 2 || i == 2
            Xa = X;
        % Select Intervals from IMU #6 for return
        elseif i == 6
            Xa = X;
        end
        clear X
    end

    hold off
    close(h)
end
end
```

euler_angles.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Calculate Euler angles Yaw, Pitch and Roll from Accelerometer and      %
%  Gyroscope from acquisition data file.                                  %
%                                                                         %
%  This function returns a 3 column matrix of angles.                     %
%                                                                         %
%  Creation: Nelson Giraldo, on 03.09.2018                                %
%  Last update: Nelson Giraldo, on 06.12.2018                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Angle] = euler_angles(allIMU,Yaw0,T,VarGx,VarGy,VarGz)
    n = size(allIMU,1);

    Ax = allIMU(:,1);
    Ay = allIMU(:,2);
    Az = allIMU(:,3);
    Gx = allIMU(:,4);
    Gy = allIMU(:,5);
    Gz = allIMU(:,6);

    % Calculate Roll and Pitch angles from accelerometer data
    Pitch = atan2d(-Ax,sqrt(Ay.^2+Az.^2));
    Roll = atan2d(Ay,Az);

    % Calculate initial Pitch and Roll angles from a small sample at the
    % beginning of acquisition
    Pitch0 = mean(Pitch(25:75));
    Roll0 = mean(Roll(25:75));

    % Angle(frame,axis), where axis = [Yaw Pitch Roll]
    Angle = zeros(n,3);
    % Initialize first data point
    Angle(1,:) = [Yaw0 Pitch0 Roll0];
    % G = [Gz(:) Gy(:) Gx(:)];

    % Calculate Yaw, Pitch, Roll angles with GYR
    Gz(-2*VarGz<Gz & Gz<2*VarGz) = 0;
    Angle(:,1) = cumsum(Gz*T);
    Gy(-2*VarGy<Gy & Gy<2*VarGy) = 0;
    Angle(:,2) = cumsum(Gy*T);
    Gx(-2*VarGx<Gx & Gx<2*VarGx) = 0;
    Angle(:,3) = cumsum(Gx*T);
end
```

EKF.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Extended Kalman Filter                                                  %
%  Using only Accelerometer and Gyroscope                                  %
%                                                                          %
%  This function returns a 3 column matrix of angles.                      %
%                                                                          %
%  Variable 'data' must contain vectors of acquisition data from both      %
%  Accelerometer and Gyroscope.  Other input variables are:                %
%       Yaw : yaw angles calculated from Gyroscope                         %
%       varGz, varGy, varGx : variance of each Gyroscope axis              %
%       T : period of acquisition                                          %
%                                                                          %
%  Creation: Nelson Giraldo, on 12.11.2018                                 %
%  Last update: Nelson Giraldo, on 15.01.2019                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [angles] = EKF(data,Yaw,varGz,varGy,varGx,T)

    n = size(data,1);

    Ax = data(:,1);
    Ay = data(:,2);
    Az = data(:,3);
    Gx = data(:,4);
    Gy = data(:,5);
    Gz = data(:,6);

    Pitch = atan2d(-Ax,sqrt(Ay.^2+Az.^2));
    Roll = atan2d(Ay,Az);

    % state vector x = [Yaw Pitch Roll]'
    x_p = zeros(3,n);
    x_e = zeros(3,n);
    F = eye(3);             % Transition matrix
    B = T*eye(3);           % Input control matrix
    G = [Gz Gy Gx];         % Input vector

    % Initial values of state vector
    x_e(:,1) = [Yaw(1) Pitch(1) Roll(1)]';
    % Initial value of state model covariance matrix
    P_e = [100 0 0; 0 100 0; 0 0 225];
    % GYR covariance process noise matrix
    Q = [varGz 0 0; 0 varGy 0; 0 0 varGx];
    z = [Ax Ay Az]';

    % Observation functions
    h = [-sind(Pitch) cosd(Pitch).*sind(Roll) cosd(Pitch).*cosd(Roll)]';
    % Jacobian matrix of the observation functions
    sizeh = max(size(h));
    H = zeros(3,3,sizeh);
    for i =1:sizeh
        H(:,:,i) = [0   -cosd(Pitch(i))                    0
                    0   -sind(Pitch(i)).*sind(Roll(i))     cosd(Pitch(i)).*cosd(Roll(i))
                    0   -sind(Pitch(i)).*cosd(Roll(i))     -
cosd(Pitch(i)).*sind(Roll(i))];
    end
```

```
    % ACC covariance measurement noise matrix
    R = eye(3).*10^(-5);

    % EKF loop
    for t = 1:n
        x_p(:,t) = F*x_e(:,t) + B*G(t,:)';      % Predicted state vector
        P_p = F*P_e*F' + Q;                     % Predicted covariance matrix

        e_t = z(:,t) - h(:,t);%H*x_p(:,t);      % z(:,t) measured data from accelerometer
        S = H(:,:,t)*P_p*H(:,:,t)' + R;
        K = P_p*H(:,:,t)'*(S)^-1;               % Kalman gain

        x_e(:,t+1) = x_p(:,t) + K*e_t;          % Updated state vector
        P_e = P_p - K*H(:,:,t)*P_p;             % Updated covariance matrix
    end
    angles = x_e(:,1:end-1)';
end
```

armAngles.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Calculate upper limb angles                                            %
%    Elbow flexion-extension                                              %
%    Shoulder flexion-extention                                           %
%    Shoulder abduction-adduction                                         %
%    Shoulder internal-external rotation                                  %
%                                                                         %
%  This function returns a vector for each calculate angle                %
%                                                                         %
%  Creation: Nelson Giraldo, on 27.10.2018                                %
%  Last update: Nelson Giraldo, on 06.12.2018                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [eFE,sAA,sFE,sIE] = armAngles(R)
    % For reference, this is what each index of variable R refers to.
    % R(data point coord (1=x,2=y,3=z), axis(1=x,2=y,3=z), frame(n), IMU #(1-9))
    n = max(size(R));

    eFE = zeros(n,1);
    sAA = zeros(n,1);
    sFE = zeros(n,1);
    sIE = zeros(n,1);

    % Elbow flexion-extension angle
    for i=1:n
        a = -R(:,1,i,6);    % Brachioradialis X axis
        b = R(:,1,i,2);     % Biceps X axis
        eFE(i) = acosd(dot(a,b)/(norm(a)*norm(b)));
    end

    % Shoulder adduction-abduction angle
    c = [0 1];              % World coordinate system Z axis within YZ plane
    for j=1:n
        b = R(2:3,1,j,2);   % Biceps X axis
        sAA(j) = acosd(dot(c,b)/(norm(c)*norm(b)));
    end

    % Shoulder flexion-extension angle
    c = [0 1];              % World coordinate system Z axis within XZ plane
    for k=1:n
        b = [R(1,1,k,2) R(3,1,k,2)];    % Biceps X axis
        sFE(k) = acosd(dot(c,b)/(norm(c)*norm(b)));
    end

    % Shoulder internal-external rotation angle
    c = [1 0];              % World coordinate system X axis within XY plane
    for m=1:n
        b = R(1:2,1,m,6);   % Brachioradialis X axis
        sIE(m) = acosd(dot(c,b)/(norm(c)*norm(b)));
    end
end
```

getACCgainoffsets.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Calculate gain matrix and offset vector for Accelerometer data from    %
%  acquisition file.                                                      %
%  This function returns 2 variables including gain and offsets for each  %
%  axis and each IMU, as well as the intervals used for the calculation.  %
%                                                                         %
%  The user is required to select intervals from the figure corresponding %
%  to periods where there is no rotation across all axes.                 %
%                                                                         %
%  Creation: Nelson Giraldo, on 28.09.2018                                %
%  Last update: Nelson Giraldo, on 06.12.2018                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [W,V] = getACCgainsoffsets(allIMUs)

    % A = W*Acc + V = [3x3]*[3x1] + [3x10]
    % W is the gain calibration matrix
    % W = [Wxx 0 0; 0 Wyy 0; 0 0 Wzz]
    % V is the offset calibration matrix
    % V = [Vx Vy Vz]'

    % During selection of intervals, choose 2 points on the abscisse for
    % each test where the IMU is immobile and the axis is pointing up or down
    % in the order as follows:
    % +x-axis, -x-axis
    % +y-axis, -y-axis
    % +z-axis, -z-axis

    nb_imu = size(allIMUs,2)/6;

    W = zeros(3,3,nb_imu);
    V = zeros(3,nb_imu);

    for i=1:nb_imu
        Ax = allIMUs(:,i*6-5);
        Ay = allIMUs(:,i*6-4);
        Az = allIMUs(:,i*6-3);

        h = figure;
        hold on
        plot(Ax,'r')
        plot(Ay,'g')
        plot(Az,'b')
        legend('Ax','Ay','Az')
        hold off

        prompt = '\nSelect intervals for Accelerometer calibration in the following
order: \n     +x-axis, -x-axis \n     +y-axis, -y-axis \n     +z-axis, -z-axis\nPress any
key to continue';
        str = input(prompt,'s');

        loop = 1;
        while loop
            [X,~] = ginput;
            n = size(X,1);
            if n == 12
```

```
            loop = 0;
        else
            clear X;
        end
    end

    X = round(X);

    Ax0 = mean(Ax(X(1):X(2)));
    Ax1 = mean(Ax(X(3):X(4)));
    Ay0 = mean(Ay(X(5):X(6)));
    Ay1 = mean(Ay(X(7):X(8)));
    Az0 = mean(Az(X(9):X(10)));
    Az1 = mean(Az(X(11):X(12)));

    W(1,1,i) = 2/(Ax0 - Ax1);
    W(2,2,i) = 2/(Ay0 - Ay1);
    W(3,3,i) = 2/(Az0 - Az1);

    V(1,i) = -(Ax0 + Ax1)/(Ax0 - Ax1);
    V(2,i) = -(Ay0 + Ay1)/(Ay0 - Ay1);
    V(3,i) = -(Az0 + Az1)/(Az0 - Az1);

    close(h)
  end
end
```