

UE PRO

Projet d'Application - Recherche

Par 146 – Benchmark et analyse de données de capteurs de mouvement

Rapport partiel d'avancement

Elèves :

Fernanda IROKAWA

Pedro PINHEIRO

Encadrants:

Romain Vuillemot

Stéphane Derrode

Contents

1.	Introduction	4
1.1.	Contexte.....	4
1.2.	Objectifs.....	5
1.2.1	Prise en main des capteurs de mouvement.....	6
1.2.2.	Enregistrer les mouvements avec différents capteurs et les représenter graphiquement pour des trajectoires générées	7
1.2.3.	Enregistrer les mouvements avec différents capteurs et les représenter graphiquement pour une trajectoire réelle.....	10
1.2.4.	Création d'une vidéo avec les données synchronisées d'une capture	12
2.	Gestion de projet.....	13
2.1.	Planning	13
2.2.	État d'avancement du projet.....	13
3.	Le projet	14
3.1.	La problématique des trajectoires.....	14
3.2.	Les capteurs de mouvement	15
3.2.1.	Motion Capture	15
3.2.2.	Accéléromètres	17
3.2.3.	Caméra vidéo	19
3.3.	Une première analyse.....	20
3.3.1.	Génération de trajectoires « aléatoires »	20
3.3.2.	Méthodes de synchronisation.....	21
3.3.3.	Autres méthodes de synchronisation	27
3.4.	Capture de mouvements.....	27

3.4.1. Calibration	27
3.4.2. Acquisition de données.....	27
3.4.3. Exportation des données	28
3.4.4. Analyse de l'expérimentation	28
4. Conclusion	28
5. Bibliographie	28
6. Annexes.....	29
6.1. Diagramme Gantt.....	29
6.2. Tutoriel Accéléromètres	31
6.2.1. Calibration	31
6.2.2. Utilisation	47

1. Introduction

1.1. Contexte

Aujourd’hui, nous disposons de plusieurs méthodes d’enregistrement de mouvements et de trajectoires, qui fournissent différents types d’information concernant ces trajectoires. Par exemple, les téléphones portables qui ont le GPS intégré et fournissent la position géographique de son utilisateur, les accéléromètres qui enregistrent l’accélération de mouvements, les gyroscopes qui fournissent les changements d’orientation, et encore plusieurs autres.

Dans le cas de l’enregistrement d’une trajectoire en utilisant différents capteurs, un problème fondamental qui apparaît est relié à l’assimilation de ces différentes données. Qu’il soit les différentes dimensions des données (position dans l’espace, accélération selon les trois axes) ou même l’écart temporel qui provient des différentes fréquences de capture, une synchronisation de ces données est nécessaire.

Nous nous intéressons finalement à la problématique de la synchronisation de données des différents capteurs : le motion capture (MoCap), l’accéléromètre et la caméra vidéo. Ce type d’étude a déjà été réalisé par Zhou et De la Torre [1], mais en s’intéressant à la synchronisation d’un même mouvement avec différentes capteurs réalisée par différents personnes. Pour cela, ils s’appuient sur différents méthodes de synchronisation, spatiale et temporelle, et explorent plus précisément le « Generalized Canonical Time Warping ».

Pour notre projet, nous envisageons réaliser la synchronisation des données d’un mouvement simple capturé avec les différents capteurs. Pour cela, il est nécessaire une étude préalable des méthodes de synchronisation existants, qu’ils soient temporels, spatiaux ou le deux.

1.2. Objectifs

SMART est une technique de description d'objectifs qui permet de les définir de façon plus claire et simple, en considérant toujours la réalisabilité et le temps nécessaire pour les atteindre.

Le mot SMART est en réalité un acronyme, où chaque lettre représente un indicateur. Les indicateurs sont différents pour chaque type de projet. Dans le cadre de notre projet, nous avons utilisé les indicateurs comme présentés ci-dessous. Pour chaque indicateur, on se pose une question pour savoir si l'objectif est bien défini. Étant répondues les 5 questions posées, on peut dire qu'on a un objectif SMART.

Le "S" représente le mot "Spécifique" : l'objectif doit être clair et bien expliqué, mais aussi simples de façon qu'il puisse être compris par des différents publics. Il faut, donc répondre la question : "Quel est notre objectif SMART?".

Le "M" représente le mot "Mesurable" : le progrès d'objectif doit être mesurable. Il faut, donc choisir les outils ou les aspects que peuvent montrer l'avancement d'objectif. La question pour cet indicateur est, alors : "Comment allons-nous mesurer nos progrès?".

Le "A" représente le mot "Atteignable" : Il faut qu'on sache les moyens qu'on peut utiliser pour atteindre l'objectif. C'est pourquoi, la question à se poser est : "Comment allons-nous atteindre notre objectif?".

Le "R" représente le mot "Réaliste" : Il doit être possible d'atteindre l'objectif. La question à répondre est, alors : "Comment savons-nous qu'il s'agit là d'un objectif réaliste?".

Le "T" représente le mot "Temporellement défini" : l'objectif doit être défini dans une période de temps. En conséquence, la question pour cet indicateur est : "Quand comptons-nous avoir atteint cet objectif?".

On a divisé notre projet en quatre parties principales. Chaque partie a été séparée en sous-objectifs. Dans les tableaux ci-dessous, on présente nos réponses pour chaque sous-tâche pour qu'elle soit SMART. La lettre correspond à la question répondue entre les cinq déjà présentées.

1.2.1 Prise en main des capteurs de mouvement

1.2.1.1. Motion Capture (MoCap)

S	Dans cette étape, notre objectif est prendre en main l'utilisation de l'appareil de motion capture situé dans la salle Amigo. Cela implique : maîtriser l'installation, le calibrage et l'utilisation des équipements et l'exportation de données.
M	On peut mesurer l'avancement à partir d'acquisitions des données de mouvements simples ainsi que la rédaction d'une méthodologie pour l'acquisition de ces données.
A	On peut atteindre notre objectif en suivant un tutoriel déjà rédigé par les anciennes élèves et en faisant plusieurs essais pour s'habituer.
R	On sait que notre objectif est réaliste et réalisable, car le MoCap est une machine qui a été déjà utilisée et maîtrisée par d'autres élèves.
T	Cet objectif doit être complété dans les deux premières semaines du projet.

1.2.1.2. Accéléromètres

S	De façon similaire à la prise en main du MoCap, on veut prendre en main l'utilisation des accéléromètres. Cela implique, une fois de plus, maîtriser
---	--

	<p>l'installation, le calibrage, l'utilisation des équipements et l'exportation de données.</p>
M	<p>On peut, une nouvelle fois mesurer notre progrès à partir d'acquisitions des données de mouvements simples ainsi que la rédaction d'un tutoriel pour la calibration et utilisation des accéléromètres.</p>
A	<p>Pour atteindre cet objectif, on peut utiliser les logiciels et tutoriels en anglais disponibles dans le site web de l'entreprise qui produit les accéléromètre [1]. Pour la calibration, on utilise le logiciel "Shimmer 9Dof Calibration". Pour les autres processus (utilisation et exportation), on utilise le logiciel "Consensys PRO".</p>
R	<p>Les logiciels et les tutoriels disponibles expliquent de façon claire et simple les étapes de fonctionnement des accéléromètres et rendent son utilisation plus simple. C'est pourquoi, on sait que notre objectif est réalisable.</p>
T	<p>Cette tâche doit être finie pendant les trois semaines après le début du projet.</p>

1.2.2. Enregistrer les mouvements avec différents capteurs et les représenter graphiquement pour des trajectoires générées

1.2.2.1. Génération de trajectoires aléatoires pour réaliser une première synchronisation

S	<p>Pendant deuxième partie de notre projet, on vise, tout d'abord, à réaliser la génération computationnelle de trajectoires simples pour faire une première synchronisation avant de passer à l'acquisition d'une trajectoire réelle.</p>
---	--

M	Dans cette étape, on peut mesurer notre progrès en regardant l'avancement du fonctionnement du code python créé ainsi que les graphiques générés.
A	Pour réaliser cette tâche, on crée un code python, avec lequel on peut générer graphiquement les trajectoires simples qui simulent l'enregistrement réel.
R	La génération d'une trajectoire est assez simple avec un code python. Cette simplicité indique la réalisabilité de l'objectif.
T	Cet objectif doit être atteint pendant la première semaine après la prise en main des capteurs.

1.2.2.2. Synchronisation des données générées en python

S	Après la génération de trajectoires, on vise à réaliser la synchronisation, aussi avec un code en python, des trajectoires simples générées.
M	Le fonctionnement du code python et les graphiques de synchronisation indiquent un progrès dans l'objectif.
A	Pour faire la synchronisation, on essaie de regarder les points d'intérêt des trajectoires (une altération de direction ou un changement de rythme par exemple), le taux d'erreur entre la trajectoire et sa référence ainsi que faire la correction des bruits (signal parasite qui génère des points qui ne correspondent pas à la trajectoire attendue). Pour cela, on utilise différents méthodes de synchronisation (Dynamical Time Warping par exemple).

R	Les algorithmes de synchronisation de données ont déjà été mis en place par des chercheurs. Cela montre que l'objectif est réaliste et réalisable.
T	Ce sous-objectif doit être réalisé pendant les trois semaines qui suivent la génération des trajectoires.

1.2.2.3. Comparaison des différents méthodes de synchronisation

S	Dans cette étape, notre objectif est de comparer les résultats obtenus pour différents types de synchronisation de trajectoires.
M	L'avancement peut être indiqué par le nombre de cas analysés.
A	Pour comparer les méthodes utilisées, on regarde les résultats graphiques pour voir quelle méthode de synchronisation a été plus adaptée pour chaque type de trajectoire analysée.
R	La qualité des méthodes sera visible dans les graphiques. L'objectif est donc réaliste.
T	Étant donné que nous sommes deux personnes dans le PAr, il n'y a pas besoin d'avoir fini toutes les synchronisations pour commencer à les analyser. Alors, cette étape sera effectuée pendant la première semaine après avoir finir l'implémentation de chaque méthode synchronisation de trajectoire.

1.2.2.4. Test de la robustesse

S	Ici, on essaye de comparer les résultats obtenus pour la synchronisation de trajectoires plus complexes par rapport à ceux des trajectoires simples.
M	Les nombre de cas analysés et un rapport montrant l'analyse des données sont des indicateurs de progrès.
A	Pour ce test, on génère des trajectoires plus complexes et on regarde les résultats graphiques pour voir quelles méthodes ont été les plus affectées par la complexité de la trajectoire analysée. On regarde aussi si la complexité change quelle est la méthode la plus adapté pour chaque type de trajectoire.
R	À nouveau, la qualité des méthodes sera visible dans les graphiques et L'objectif est donc réaliste.
T	Les méthodes étant déjà codés en python, il faut juste simuler les trajectoires plus complexes. Par conséquente, ces tests seront réalisés pendant la première semaine après avoir finir la comparaison des méthodes de synchronisation des trajectoires.

1.2.3. Enregistrer les mouvements avec différents capteurs et les représenter graphiquement pour une trajectoire réelle

1.2.3.1. Enregistrement de données d'une trajectoire réelle

S	Après les processus avec les trajectoires générées, on vise à enregistrer les données d'une trajectoire réelle avec différents capteurs.
---	--

M	La quantité des données enregistrées indiquent le progrès d'objectif.
A	En utilisant les capteurs disponibles (accéléromètre et MoCap jusqu'au moment), on peut enregistrer les données pour une trajectoire réelle avec la méthodologie créée en 1.2.1.1.
R	On sait que le projet est réalisable, puisqu'on a déjà enregistré les données de trajectoires pendant l'étape de prise en main des capteurs.
T	Cette étape ne dépend que de l'étape de prise en main des capteurs. Alors, elle peut être faite pendant la première semaine après la prise en main des appareils.

1.2.3.2. Synchronisation des données pour la trajectoire réelle

S	Cette étape a pour but de réaliser l'exportation et la synchronisation des données enregistrées pour les trajectoires réelles.
M	Le fonctionnement code en python capable d'analyser et synchroniser les données indique notre progrès.
A	Après les avoir enregistré, on exporte les données référentes à chaque capteur et on les synchronise en utilisant un code python et les méthodes déjà utilisées et codés en python. On utilise la méthode plus adapté pour la trajectoire réelle réalisée selon les analyses faites pour les trajectoires générées.
R	La synchronisation est faite de la même façon que pour la trajectoire générée, donc c'est réalisable.

T	Étant donné que les codes pour la synchronisation sont déjà faites, on effectue cette tâche pendant la première semaine après l'analyse des méthodes pour les trajectoires générées.
---	--

1.2.4. Création d'une vidéo avec les données synchronisées d'une capture

S	Pour montrer nos résultats, on vise à faire une vidéo montrant les données enregistrées par les capteurs déjà synchronisées et avec la trajectoire réelle (enregistrée par une caméra vidéo).
M	Notre avancement sera mesuré par le fonctionnement d'un code python capable de reproduire les graphiques des données synchronisés et par la production de la vidéo lui-même.
A	Pour accomplir cet objectif, on utilise un code python pour reproduire les graphiques synchronisées et on ajoute l'enregistrement fait avec la caméra vidéo pour montrer le mouvement de façon plus visible.
R	Une expérience similaire a été déjà réalisée par Zhou et De la Torre. Cela montre que c'est réalisable.
T	Cette étape sera réalisée pendant les 2 semaines après la synchronisation des données des trajectoires réelles.

2. Gestion de projet

2.1. Planning

Établir les objectifs d'une façon « smart » permet de construire un planning pour le projet. Ce planning suit la durée définie pour chaque objectif au début, et respecte aussi les conditions d'antériorité nécessaires à chaque tâche (comme par exemple, la réalisation de la synchronisation d'une trajectoire réelle ne peut pas être faite sans avoir capturé les données correspondantes).

Le planning est présenté sous la forme d'un diagramme Gantt (en annexe), mis à jour après la réalisation de chaque tâche et en informant la période qu'elle a été réalisé. Cela nous permet de suivre l'avancement du projet ainsi que notre retard par rapport à ce qui a été établi.

2.2. État d'avancement du projet

La définition des objectifs du projet et la reformulation du sujet nous ont pris beaucoup de temps en ce qui concerne le temps total du projet. Aujourd'hui nous nous trouvons à la moitié du projet (3 mois sont déjà passés) et par rapport à notre planning, nous sommes quelques semaines en retard.

En ce qui concerne nos objectifs, le Gantt en annexe montre ce qui nous avons déjà réalisée par rapport aux objectifs et délais définis dans le planning. Même si quelques tâches sont en retard, d'autres dont la réalisation était prévue pour les mois de février et mars ont été déjà faites ou commencées, comme par exemple une capture de mouvement a été réalisée, ainsi que le vidéo de synchronisation qu'on souhaite réaliser à la fin a aussi déjà été commencé.

3. Le projet

3.1. La problématique des trajectoires

L'avancement technologique a permis le développement des techniques de capture de mouvement. Toutes les données qui peuvent être enregistrées servent à mieux comprendre la mobilité et le mouvement de différents objets. Y. Zheng [2] a réalisé un aperçu de la problématique de l'exploitation de données de trajectoires aujourd'hui (Figure 3-1).

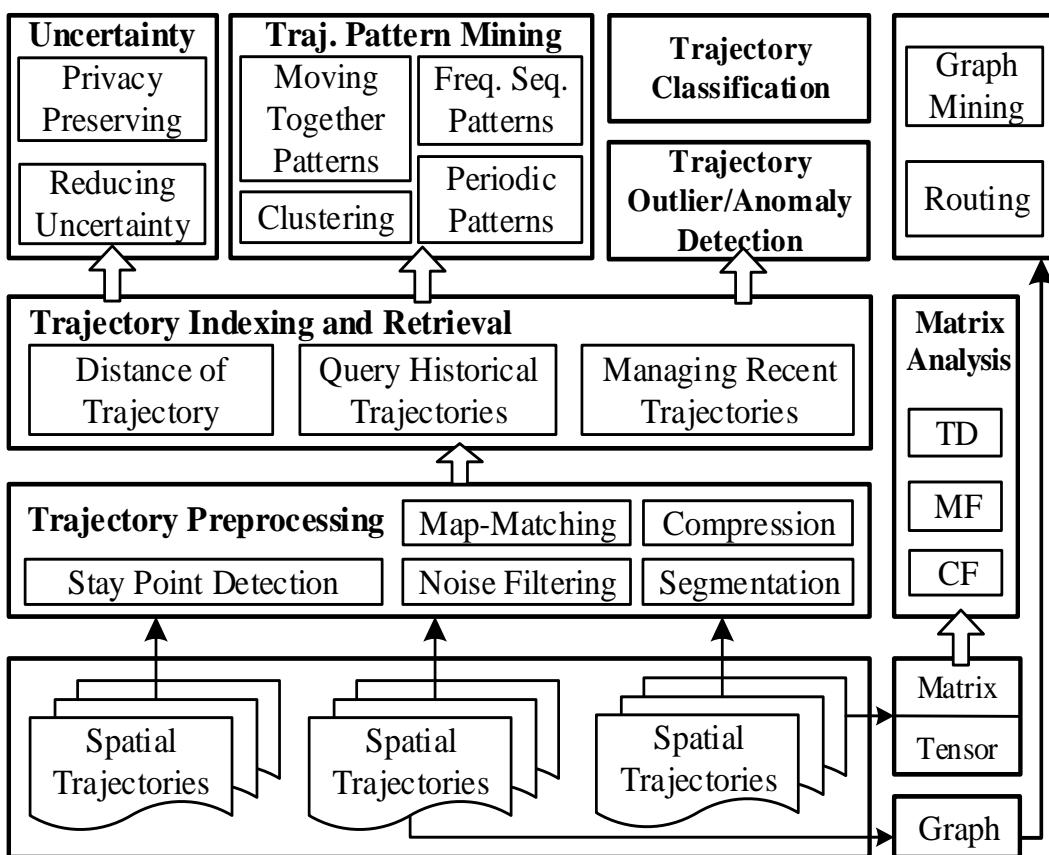


Figure 3-1 : « Le paradigme de l'exploitation de données de trajectoires » [2]

La problématique se développe autour de trois grands axes : le premier est le *prétraitement des données*, qui comprend par exemple le filtrage du bruit, la taille des données acquises. Ensuite, *l'indexage et la récupération de la trajectoire* concerne les méthodes analyse et manipulation des données. Finalement, le

troisième axe consiste au traitement porté aux trajectoires après l'exécution des deux premières étapes. Cela concerne l'incertitude lié à la discréétisation d'une trajectoire (il n'est pas possible de savoir la position d'un objet en mouvement entre deux points consécutifs acquis), l'analyse de motifs qui peuvent apparaître dans une trajectoire (et qui peuvent servir à la classer), l'étude des anomalies dans les trajectoires.

Tout cela représente aujourd'hui l'état de l'art de ce vaste domaine d'étude.

3.2. Les capteurs de mouvement

Dans le cadre de ce projet, on mettra en place l'utilisation de trois capteurs. Dans cette section, on explique la démarche de fonctionnement de ces capteurs ainsi que leurs avantages et désavantages.

3.2.1. Motion Capture

3.2.1.1. La démarche

La technique de motion capture est la plus utilisée pour faire les animations et les effets spéciaux dans les films. Cette technique fonctionne en utilisant les différents capteurs, qui enregistrent les données des trajectoires et leur transmettent à un logiciel capable d'interpréter ces données brutes et reconstruire les trajectoires 3D en temps réel dans un environnement virtuel. Cette technologie est utilisée pour l'étude des mouvements humains, mais aussi dans les domaines de la santé et de l'anatomie.

La capture peut être faite de différentes façons (optique, gyroscopique, mécanique ou magnétique). Dans ce rapport on ne s'intéressera qu'à la capture optique, étant laquelle nous avons disponible dans la salle amigo et aussi la plus utilisée.

La capture optique, à son tour, peut être faite de deux façons (avec marqueurs et sans marqueurs). La technologie utilisée par le Kinect par exemple n'utilise pas des marqueurs. L'appareil de motion capture situé dans la salle AMIGO, utilise les

marqueurs pour capturer les données et, par conséquent, on ne parlera que de ce type de capture.

Ce type de capture est fait en plaçant des marqueurs sur la personne ou l'objet qui ira réaliser le mouvement (Figure 3-2). Ces marqueurs sont capables de réfléchir le rayonnement infrarouge. Les caméras émettent un rayonnement infrarouge et captent sa réflexion. Comme les caméras ne sont sensibles qu'à les longueurs d'onde dans le domaine infrarouge, les caméras n'enregistrent pas l'image visible par l'être humain, mais si la réflexion des marqueurs sur forme de points. Il faut dire que les marqueurs sont de format arrondi pour que la réflexion soit faite pour n'importe quel angle d'incidence des rayonnements.



Figure 3-2 : Capture de mouvement en utilisant le Motion Capture [3]

Vu qu'une caméra toute seule n'est pas capable d'observer 3 dimensions, il faut au moins deux caméras pour que le logiciel puisse calculer la position du point dans l'espace par triangulation (il y en a six dans la salle AMIGO). Pour la rotation il faut au moins deux capteurs qui doivent être liés manuellement à partir du logiciel. Ces deux points seront donc traités comme une trace dans la visualisation virtuelle. Cela permet le logiciel de faire le calcul et la représentation de la rotation dans l'environnement virtuel.

3.2.1.2. Avantages

L'appareil de capture de mouvement optique a une calibration facile avec une croix de calibration. Il faut juste mouvementer la croix dans la salle. Aussi, on a une représentation compréhensible et virtuelle de la trajectoire en temps réel. L'exportation des données enregistrées peut être faite de façon très simple avec le logiciel.

3.2.1.3. Désavantages

L'appareil présente des désavantages liés à l'optique. L'infrarouge n'est pas capable de traverser les objets et alors nous avons la possibilité de cacher un ou plus marqueurs pendant quelques instantes. Les marqueurs doivent, alors, être placés de façon à qu'ils soient visibles durant toute l'acquisition. Aussi, la capture peut être perturbée à cause des points parasites dus aux objets qui sont aussi capables de réfléchir l'infrarouge.

Un deuxième problème est lié à l'espace d'utilisation de l'appareil. Le mouvement doit être fait dans l'espace délimitée par la vision des caméras. L'appareil est, donc, fixé et limite les types de mouvement que peuvent être fait.

Un troisième problème est le prix des appareils qui est très élevé. Celui de la salle AMIGO vaut au tour de 40.000 € par exemple.

3.2.2. Accéléromètres

3.2.2.1. La démarche

Quand on parle d'un accéléromètre, en réalité, on parle d'un système trois accéléromètres. Chacun de ses accéléromètres mesure l'accélération linéaire dans une des trois axes de directions orthogonaux.

Le principe peut être illustré par un système masse-ressort. Si on fait un mouvement dans un système de ce type, la masse essaie de rester dans sa position initiale à cause de l'inertie. Le ressort est alors allongé ou comprimé et par la loi fondamentale

de la dynamique pour une ressort, $F = K \cdot x$, mais aussi $F = m \cdot a$. Alors, si on sait la différence de position de la masse (Δx), la masse m et la constante de rappel du ressort K , on peut calculer le module et la direction de l'accélération dans une axe. Avec trois systèmes, on peut observer l'accélération dans les trois dimensions et, en observant les changements d'accélération dans les 3 axes, on peut avoir la rotation d'accéléromètre.

Dans notre projet, on utilise l'accéléromètre Shimmer3 (Figure 3-3). Pour exporter et pour traiter les données pour cet accéléromètre de façon simple et pour configurer le Shimmer3 (changer la fréquence d'enregistrement par exemple), on peut utiliser le logiciel Consensys PRO [4]. Pour la calibration des accéléromètres on utilise le logiciel Shimmer 9DoF Calibration [5].



Figure 3-3 : Accéléromètre du type Shimmer3 [4]

3.2.2.2. Avantages

Le logiciel Consensys PRO pour l'accéléromètre Shimmer3 permet d'exporter et traiter les données de façon simple et rapide. L'accéléromètre est portable et donc permet d'analyser mouvements sans une contrainte d'espace. La calibration est aussi simple et rapide avec le logiciel Shimmer 9DoF. Il est beaucoup moins cher que les appareils MoCap.

3.2.2.3. Désavantages

L'accéléromètre ne donne pas directement la position spatiale. Pour cela, il faut utiliser des algorithmes qui essaient de calculer cette position en utilisant les accélérations, ce qui normalement ne donne pas des résultats précis.

3.2.3. Caméra vidéo

3.2.3.1. La démarche

Les lentilles sont l'ouverture pour les informations. Elles capturent les grandes portions de lumière qui contiennent les informations de l'image. La couleur est formée à partir de l'intensité des trois couleurs primaires (bleu, vert et rouge) dans la lumière.

Le Charged Couple Device (CDD) est constitué de 300 mil à 500 mil diodes sensibles à la lumière (photosites). Chacun de ces microcarrés contient une couleur pour former l'image. Le CDD transforme, alors, la quantité de lumière qu'il reçoit pour chaque photosite en charges électriques. Ces charges sont envoyées, une par une, à un capteur qui enregistre l'image complète. Pendant ce processus, la prochaine image est capturée. Ces étapes sont répétées jusqu'à l'arrêt du vidéo. Les images sont enregistrées en séquence pour après être regroupées pour donner la sensation de mouvement.

3.2.3.2. Avantages

La caméra vidéo présente la trajectoire de la manière la plus compréhensible pour l'être-humain. Le prix varie mais est aussi beaucoup moins cher que pour le MoCap.

3.2.3.3. Désavantages

La caméra vidéo présente la trajectoire de la manière la plus compréhensible pour l'être-humain. Le prix varie mais est aussi beaucoup moins cher que pour le MoCap.

3.3. Une première analyse

L'étude de trajectoires enregistrées par les capteurs de mouvement s'appuie sur une étude théorique préalable. Les données récupérées ne sont pas forcément enregistrées avec une même fréquence ni une même dimension. Pour pouvoir étudier le cas réel de trajectoires capturées, nous nous intéressons d'abord à l'étude de la synchronisation de trajectoires générées à partir d'un programme informatique.

3.3.1. Génération de trajectoires « aléatoires »

Nous nous intéressons à la génération de la trajectoire d'un stylo qui dessine un carré (ou même d'une voiture qui tourne en rond autour d'un bloc) et qui va être enregistré par un capteur de mouvement. Pour cela, nous avons une trajectoire de référence (la trajectoire « réelle ») et la trajectoire enregistrée par le capteur. Un capteur de mouvement aura certainement des incertitudes plus ou moins importantes dépendant de l'appareil utilisé.

La génération des trajectoires est réalisée de façon à avoir une trajectoire de référence et une trajectoire dite « aléatoire », à laquelle les incertitudes propres à un capteur de mouvement sont ajoutées sous la forme d'une distribution de probabilité uniforme. Le choix de ce type de distribution de probabilité a été fait à cause de sa simplicité, mais il serait possible d'utiliser autres distributions que s'approchent plus de ce qu'on observe normalement, comme une distribution gaussienne.

Les deux trajectoires générées pour étudier les méthodes de synchronisation ont différentes types de segmentation et la présence (ou pas) de l'incertitude (Figure 3-4).



Figure 3-4 : Représentation graphique des trajectoires de référence (en bleu) et l'aleatoire (en orange) générées

3.3.2. Méthodes de synchronisation

Les méthodes de synchronisation sont plusieurs disponibles dans plusieurs bibliographies. Pour notre projet, nous nous concentrerons surtout sur quatre différentes méthodes de synchronisation [6]–[8].

3.3.2.1. Dynamic Time Warping

Le Dynamic Time Warping (DTW) [6], [9], [10] a été étudié en plusieurs cas de synchronisation de trajectoires. Il est une méthode simple qui normalement est incorporé à d'autres méthodes de synchronisation plus puissants et plus complexes. En dehors du contexte de trajectoires, il a été déjà utilisé pour réaliser la reconnaissance vocale, la correspondance de signatures. Il consiste à synchroniser deux séries de données en utilisant la distance minimale entre chaque point de chaque trajectoire (Figure 3-4).

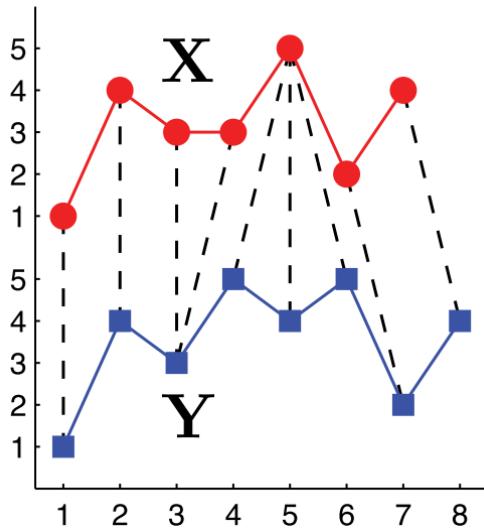


Figure 3-5 : deux trajectoires X et Y. Les lignes pointillées représentent la synchronisation en utilisant la méthode DTW [6]

Le principe

A partir de deux séries temporelles $X = [x_1, \dots, x_{nx}] \in \Re^{2 \times nx}$ et $Y = [y_1, \dots, y_{ny}] \in \Re^{2 \times ny}$, la synchronisation des données doit minimiser la distance entre les points alignés :

$$\min(J_{dtw}(P)) = \sum_{t=1}^m \|x_{p_t}^x - y_{p_t}^y\|^2$$

Où P est la matrice de correspondance $P = [p^x, p^y]^T \in \Re^{2 \times m}$ et m est le nombre d'indices nécessaires pour réaliser la correspondance entre les séries. Les deux vecteurs qui composent cette matrice de correspondance sont $p_x \in \{1 : n_x\}^{m \times 1}$ et $p_y \in \{1 : n_y\}^{m \times 1}$. La matrice de correspondance P doit satisfaire les contraintes :

- Condition de frontière : $p_1 = [1, 1]^T$ et $p_m = [n_x, n_y]^T$, c'est-à-dire, le premier et le dernier frame des séries doivent être alignés
- Condition de continuité : $0 \leq p_t - p_{t-1} \leq 1$
- Condition de monotonie : $t_1 \geq t_2 \Rightarrow p_{t_1}^x \geq p_{t_2}^x$ et $p_{t_1}^y \geq p_{t_2}^y$

Le i_{eme} frame de la série X sera aligné au j_{eme} frame de la série Y s'il existe un p_t tel que $p_t = [p_t^x, p_t^y] = [i, j]$.

Implémentation

Pour pouvoir implémenter la méthode du DTW, il était nécessaire d'envisager quelles structures seraient fondamentales pour résoudre le problème. La synchronisation des données est faite en trouvant le chemin optimal (par rapport à la distance entre les points de chaque série) pour arriver du premier point au dernier. La Figure 3-6 montre une représentation du chemin qu'on veut trouver (en rouge). En effet, à chaque pair (i, j) de cette matrice, on peut trouver la distance entre les deux points x_i et y_j . Le chemin va donc passer par les points où cette distance est minimale, en respectant toujours le sens qu'il peut suivre (soit à gauche, soit en bas ou soit à gauche et en bas – Figure 3-7).

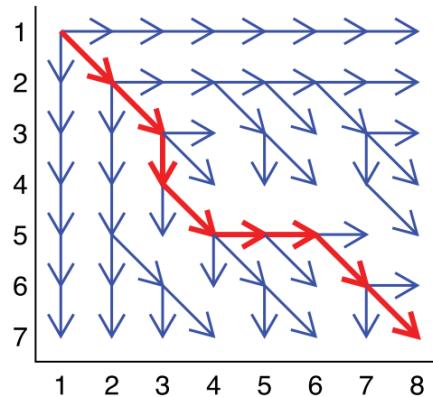


Figure 3-6 : représentation graphique méthode du DTW pour trouver le chemin optimal de synchronisation des deux séries temporelles X et Y [6]

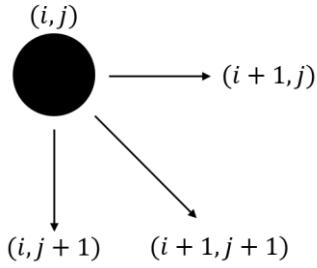


Figure 3-7 : représentation des chemins possibles pour un point (i,j)

L'implémentation passe donc par une matrice de distances où chaque élément (i, j) de la matrice correspond à la distance entre ces deux points x_i et y_j (Figure 3-8). On peut aussi retrouver les données synchronisées en partant du dernier élément de la matrice représentée par la Figure 3-7, c'est-à-dire, le point $\{n_x, n_y\}$ (condition de frontière) et en réalisant le chemin inverse pour y arriver.

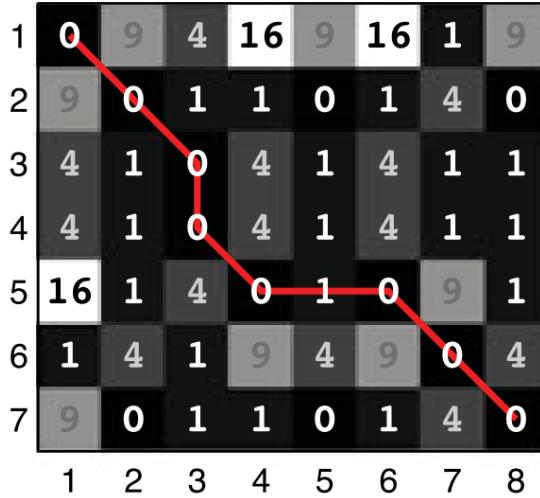


Figure 3-8 : Représentation de la matrice de distances pour deux trajectoires X et Y [6]

La représentation graphique de cette synchronisation (Figure 3-9) permet de montrer l'association entre ces points et sert aussi à savoir si tous les conditions aux limites ont été respectées. En effet, à chaque point p_t , l'élément (p_t^x, p_t^y) de la matrice représenté dans la figure assume la valeur 1. Ce type d'illustration sera utilisé pour une analyse de nos résultats.

1	1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0
5	0	0	0	0	1	1	1	0	0
6	0	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	0	1
	1	2	3	4	5	6	7	8	9

Figure 3-9 : Représentation de la matrice de correspondance [3]

Résultats et analyse

En utilisant l'algorithme du DTW, les deux trajectoires générées en 3.3.1 (représentées graphiquement par la Figure 3-4) ont été alignées. On voit que la synchronisation tend vers la diagonale qui lie le premier (p_1) et le dernier point (p_m). De plus, on obtient une matrice de correspondance de taille $m = 124$. Cela veut dire qu'à chaque point de la trajectoire de référence on associe un et seulement un point de la trajectoire « aléatoire ». Par contre, les points de cette dernière s'associent à un ou plusieurs points de la référence. Il est possible d'observer cela à partir de la largeur du rectangle en blanc à chaque ligne de la Figure 3-10. Ce résultat est attendu car la trajectoire « aléatoire » a été créée en utilisant les points de la référence.

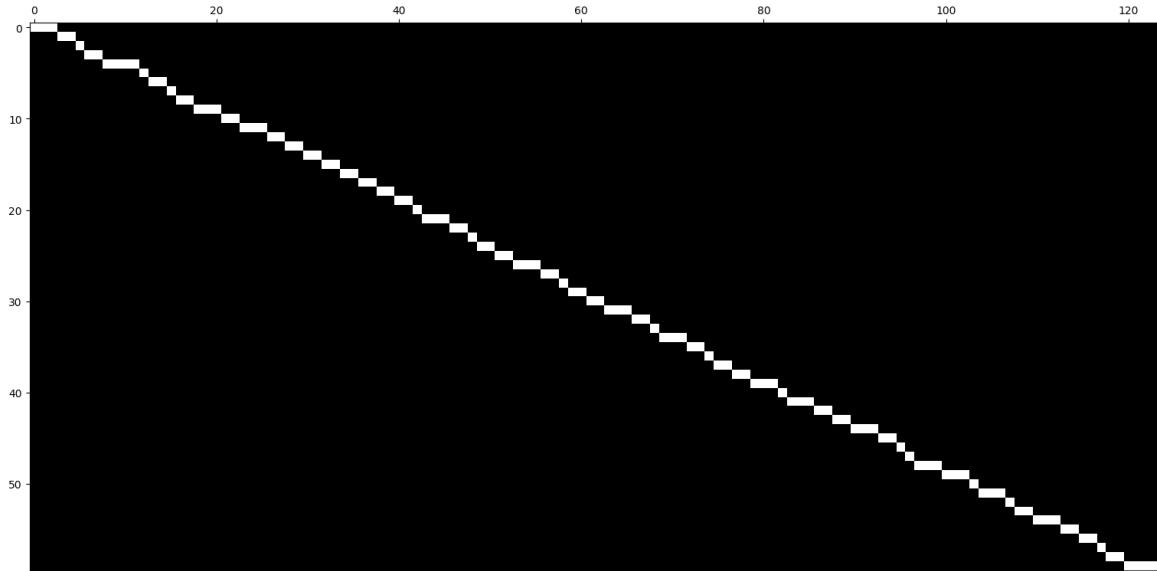


Figure 3-10 : Résultat graphique de la synchronisation de la trajectoire de référence avec la trajectoire "aléatoire"

Bien que cette méthode permette de synchroniser les deux trajectoires qui ont été générées à ce but, la méthode du DTW présente plusieurs limitations que peuvent devenir contraignantes à la synchronisation de données provenues des différents capteurs [5]. Comme le DTW ne permet pas de réaliser une pondération des données et ne peut donc être utilisé dans la synchronisation de données multimodales. De plus, comme il s'appuie dans la programmation dynamique, lorsqu'on augmente la taille des données à analyser, le temps nécessaire à la synchronisation augmente de manière quadratique. Finalement, les conditions aux limites imposées (comme la condition de frontière) font le chemin optimal trop « rigide », ce qui peut ne pas convenir quand on désire synchroniser des trajectoires qui sont écartés dans le temps.

Cette méthode, très classique et simple, est cependant peu adaptée à la problématique traitée dans ce projet. Les trajectoires dont la synchronisation est souhaitée ne présentent pas la même dimension et ne sont pas non plus enregistrées avec la même fréquence d'acquisition et/ou le même instant de départ. Il est

nécessaire, donc, d'utiliser des méthodes plus poussés et plus adaptés à notre contexte.

3.3.3. Autres méthodes de synchronisation

La méthode à utiliser pour la synchronisation des données des captures de mouvement avec le Motion Capture, l'accéléromètre et la caméra vidéo doit donc être adaptée à la multimodalité des capteurs. Trois méthodes sont encore à étudier [6]–[8]. Ils sont basés sur le DTW[6], [7] et sur la corrélation croisée [8].

3.4. Capture de mouvements

Pour la capture de mouvements, nous avons créé une méthodologie avec différents étapes. Ici, la méthodologie est utilisée pour la capture du mouvement d'un stylo qui dessine un rectangle. Il faut savoir que, pour la capture d'une trajectoire plus complexe, quelques étapes de la méthodologie pourront être adaptées.

3.4.1. Calibration

- Calibration des six caméras du MOCAP en utilisant la croix de calibration.
- Calibration d'un capteur Shimmer3 (Accéléromètre) en utilisant le logiciel Shimmer 9DoF Calibration.

3.4.2. Acquisition de données

- En posant un réflecteur et un capteur Shimmer sur un stylo, capturer les données, avec les deux capteurs de façon simultanée, du mouvement du stylo sur la feuille de dessin (il est important de maintenir le stylo perpendiculaire par rapport à la feuille pour que le mouvement des capteurs soit seulement dans un plan parallèle au plan de la feuille de dessin).
- Réaliser le mouvement du stylo de façon qu'il soit visible pour toutes les six caméras pour assurer la bonne reconstruction du mouvement MOCAP.

3.4.3. Exportation des données

- Exporter les données acquises avec les deux capteurs dans le format .csv (lisible par Excel par exemple).

3.4.4. Analyse de l'expérimentation

- Comparer les deux trajectoires reconstruites pour analyser la précision de ces trajectoires par rapport à la réalité marquée sur la feuille de dessin utilisée.

4. Conclusion

Aujourd’hui le projet se trouve en retard par rapport au planning établi. Comme il nous reste seulement 3 mois de projet, il faut réussir à rattraper ce retard pour pouvoir atteindre tous les objectifs définis.

Ce projet a pour but l’étude de trajectoires de mouvement générées à partir de capteurs de mouvement dont on dispose à la salle Amigo. Cette analyse passe d’abord par l’étude de trajectoires synthétiques. Les résultats obtenus pourront alors être exploités pour de trajectoires réelles. Plusieurs méthodes existent pour réaliser ce type d’alignement de données, et nous avons déjà réalisé une première analyse en utilisant le DTW. Cette méthode ne semblant pas idéal dans le contexte de notre projet, d’autres méthodes sont encore à étudier.

5. Bibliographie

- [1] « Wearable Sensor Technology | Wireless IMU | ECG | EMG | GSR ». [En ligne]. Disponible sur: <http://www.shimmersensing.com/>. [Consulté le: 18-janv-2018].
- [2] Y. Zheng, « Trajectory Data Mining: An Overview », *ACM Trans Intell Syst Technol*, vol. 6, n° 3, p. 29:1–29:41, mai 2015.
- [3] H. Josiński, D. Kostrzewska, A. Michalczuk, et A. Świtoński, « The Expanded Invasive Weed Optimization Metaheuristic for Solving Continuous and Discrete Optimization Problems », *ScientificWorldJournal*, vol. 2014, p. 831691, mars 2014.

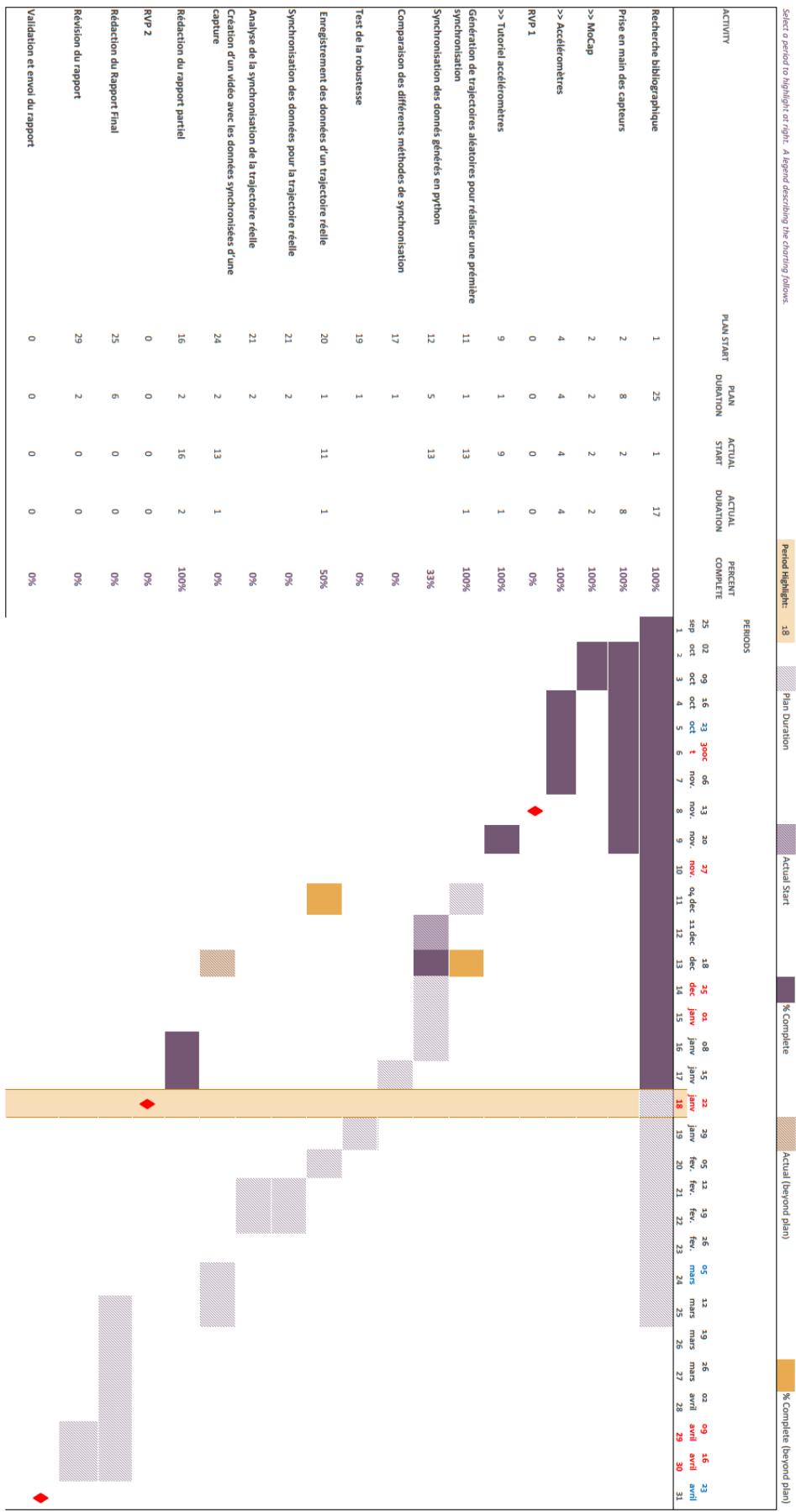
- [4] « ConsensysPRO Software | Collect and analyze biometric and motion data from Shimmer sensors. » [En ligne]. Disponible sur: <http://www.shimmersensing.com/products/consensys>. [Consulté le: 18-janv-2018].
- [5] « Shimmer 9DoF Calibration Application | IMU Calibration ». [En ligne]. Disponible sur: <http://www.shimmersensing.com/products/shimmer-9dof-calibration>. [Consulté le: 18-janv-2018].
- [6] F. Zhou et F. D. la Torre, « Generalized Canonical Time Warping », *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, n° 2, p. 279-294, févr. 2016.
- [7] S. Sankararaman, P. K. Agarwal, T. Mølhave, J. Pan, et A. P. Boedihardjo, « Model-driven Matching and Segmentation of Trajectories », in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, New York, NY, USA, 2013, p. 234–243.
- [8] L. Fridman, D. E. Brown, W. Angell, I. Abdić, B. Reimer, et H. Y. Noh, « Automated synchronization of driving data using vibration and steering events », *Pattern Recognit. Lett.*, vol. 75, n° Supplement C, p. 9-15, mai 2016.
- [9] F. Zhou et F. De la Torre, « Canonical Time Warping for Alignment of Human Behavior », in *Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference*, 2009, p. 2286-2294.
- [10] F. Zhou et F. D. la Torre, « Generalized time warping for multi-modal alignment of human motion », in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, p. 1282-1289.

6. Annexes

6.1. Diagramme Gantt

Project Planner

Select a period to highlight at right. A legend describing the charting follows.



6.2. Tutoriel Accéléromètres

NB : Ce tutoriel sert à l'utilisation du shimmer3, les autres versions ne seront pas adaptées.

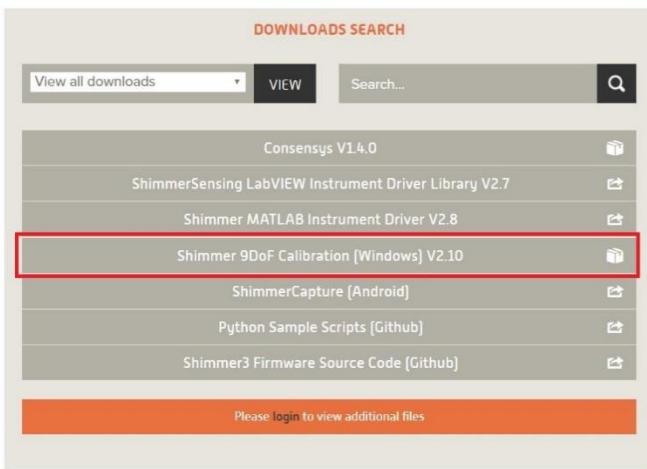
Pour pouvoir utiliser les accéléromètres, vous devez avoir installé le programme *Consensys* sur votre ordinateur (disponible pour téléchargement [ici](#)). Utilisez le [User Guide](#) pour l'installer.

Branchez le câble de la base à une prise et ensuite branchez le câble à la base (suivez bien cette ordre). Branchez la base à l'ordinateur avec le câble USB. Pour allumer un des accéléromètres, appuyez sur le bouton orange.

NB : Pour utiliser plusieurs capteurs, il faut avoir une licence pour le ConsensysPRO.

6.2.1. Calibration

Téléchargez le programme *shimmer 9DOF calibration*. La dernière version est disponible sur le site [Shimmer Sensor](#). Un document sur la calibration (en anglais) est aussi disponible pour téléchargement.



Il est recommandé d'aussi suivre la [vidéo](#) avec la démonstration de la calibration pour mieux suivre ce tutoriel.

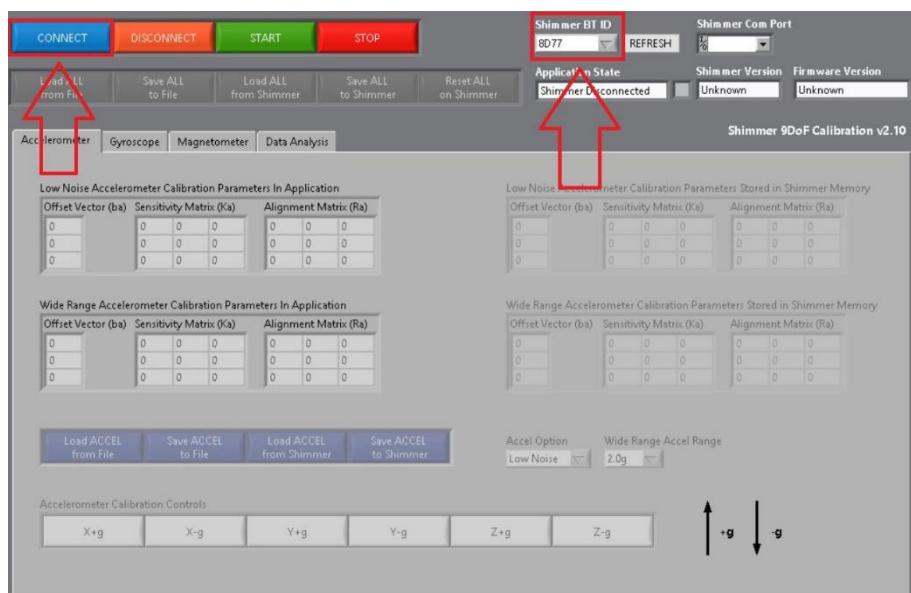
La calibration doit être faite pour chaque capteur séparément.

Avec le programme *shimmer 9DOF calibration* installé, allumez le Bluetooth de votre ordinateur et connectez-le au capteur que vous irez calibrer (PIN : 1234). Il faut que le capteur soit allumé pour pouvoir le connecter. Vous devez laisser le capteur connecté à votre ordinateur pendant tout le processus de calibration. Le capteur doit apparaître dans "Appareils Bluetooth disponibles" comme "Shimmer3-XXXX" (ou XXXX sera l'ID de votre capteur, par exemple "Shimmer3-8D77").

Posez le capteur sur le "*Shimmer3 Calibration Stand*".



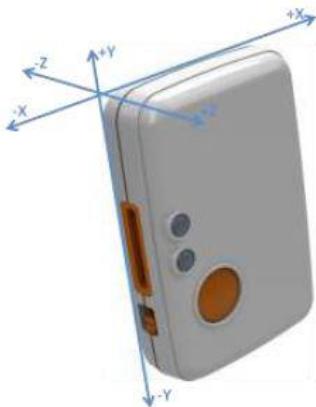
Ouvrez le programme *shimmer 9DOF calibration*. Sélectionnez l'ID du capteur et cliquez sur "**Connect**".



Il va apparaître "Shimmer Connected" sur "Application State" si la connexion est réussite.

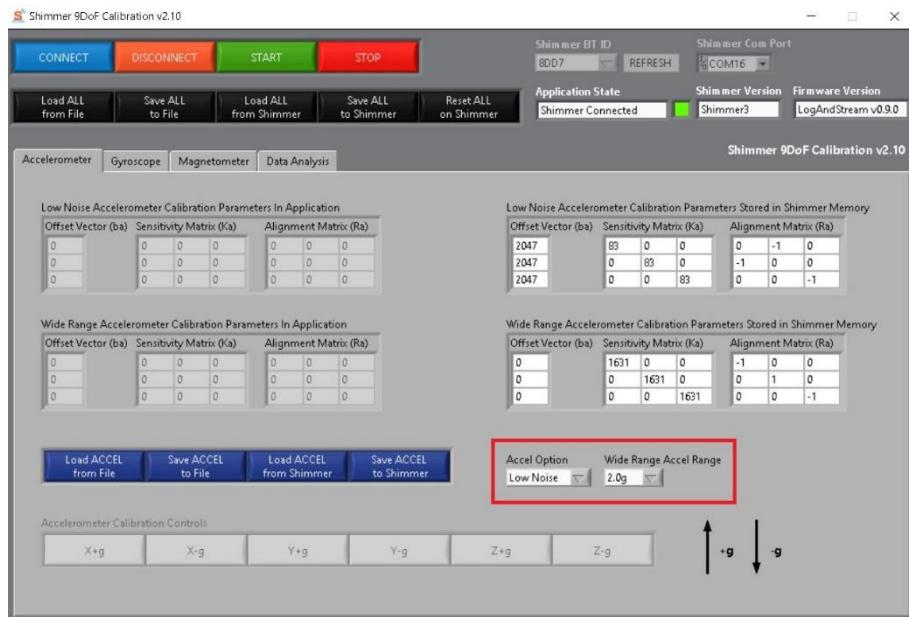


Pour calibrer, il faut définir les axes du capteur. Dans notre cas, nous utiliserons la convention de l'image ci-dessous.



6.2.1.1. Calibration de l'accéléromètre

Vous pouvez choisir le type d'accélération et la gamme d'accélération (cela va dépendre de l'application envisagée pour l'accéléromètre). Dans ce tutoriel, on choisit "Low noise" et "2.0g".



Pour commencer la calibration, vous devez choisir une surface plane. Cliquez sur **START** (en vert) pour lancer le transfert de données du shimmer vers l'ordinateur.
Placez votre axe x vers le haut.



Appuyez sur **X+g** et attendez jusqu'à que le "wait" disparaît.

Tournez le shimmer de façon a placer l'axe x vers le bas. Appuyez sur **X-g** et attendez jusqu'à que le "**wait**" disparaît.



Positionnez l'axe y vers le haut. Appuyez dur "**Y+g**" et attendez jusqu'à que le "**wait**" disparaît.



Tournez le shimmer de façon à placer l'axe y vers le bas. Appuyez sur **Y-g** et attendez jusqu'à que le "**wait**" disparaît.



Positionnez l'axe z vers le haut. Appuyez sur "**Z+g**" et attendez jusqu'à que le "**wait**" disparaît.



Tournez le shimmer de façon à placer l'axe z vers le bas. Appuyez sur **Z-g** et attendez jusqu'à que le "**wait**" disparaît.



Vous pouvez répéter autant de fois que vous voulez chaque axe. Après avoir suivi cette méthode pour les trois axes, cliquez sur "**STOP**" (en rouge).

Normalement vous devez obtenir un tableau comme celui-ci :

Shimmer 9DoF Calibration v2.10

CONNECT DISCONNECT START STOP

Shimmer BT ID: 80D7 Shimmer Com Port: COM16

Load ALL from File Save ALL to File Load ALL from Shimmer Save ALL to Shimmer Reset ALL on Shimmer Application State: Shimmer Connected Shimmer Version: Shimmer3 Firmware Version: LogAndStream v0.9.0

Accelerometer Gyroscope Magnetometer Data Analysis

Shimmer 9DoF Calibration v2.10

Low Noise Accelerometer Calibration Parameters In Application			Low Noise Accelerometer Calibration Parameters Stored in Shimmer Memory		
Offset Vector (ba)	Sensitivity Matrix (Ka)	Alignment Matrix (Ra)	Offset Vector (ba)	Sensitivity Matrix (Ka)	Alignment Matrix (Ra)
2016	82 0 0	0 -1 0	2047	83 0 0	0 -1 0
1990	0 82 0	-1 0 -0,01	2047	0 83 0	-1 0 0
2030	0 0 83	0,01 -0 -1	2047	0 0 83	0 0 -1

Wide Range Accelerometer Calibration Parameters In Application			Wide Range Accelerometer Calibration Parameters Stored in Shimmer Memory		
Offset Vector (ba)	Sensitivity Matrix (Ka)	Alignment Matrix (Ra)	Offset Vector (ba)	Sensitivity Matrix (Ka)	Alignment Matrix (Ra)
0	0 0 0	0 0 0	0	1631 0 0	-1 0 0
0	0 0 0	0 0 0	0	0 1631 0	0 1 0
0	0 0 0	0 0 0	0	0 0 1631	0 0 -1

Load ACCEL from File Save ACCEL to File Load ACCEL from Shimmer Save ACCEL to Shimmer

Accel Option: Low Noise Wide Range Accel Range: 2,0g

Accelerometer Calibration Control: X+g X-g Y+g Y-g Z+g Z-g

Diagram showing the orientation of the sensor during calibration: Up arrow labeled '+g', Down arrow labeled '-g'.

Si le tableau en bleu, la calibration n'a pas encore été transmise au shimmer. Il faut donc appuyer sur "**Save ACCEL to Shimmer**". Vous pouvez aussi enregistrer

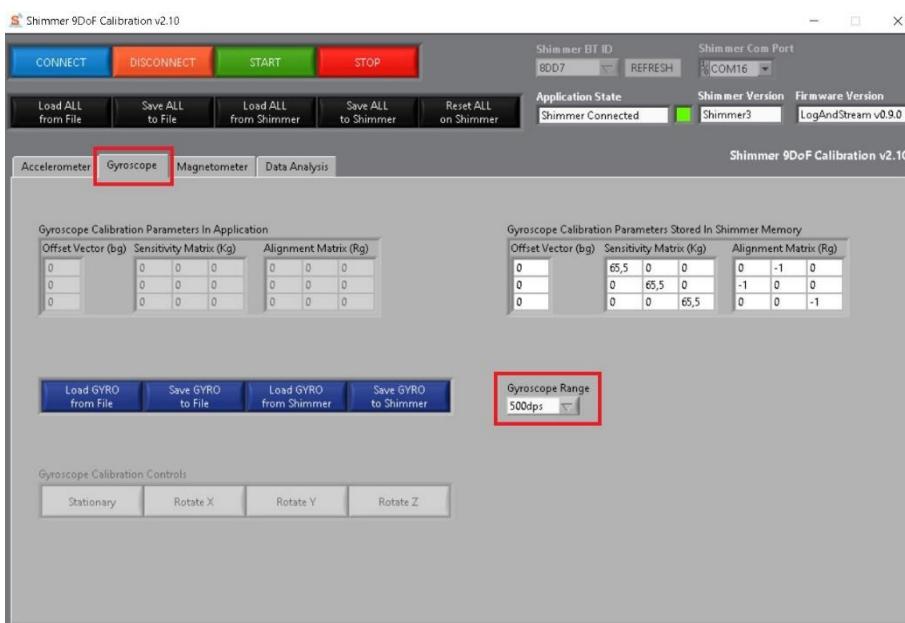
votre calibration sur votre ordinateur en cliquant sur "**Save ACCEL to file**" (il est recommandé de le faire au cas où la calibration est perdue à cause d'un imprévu).

Après avoir appuyé sur "Save ACCEL to Shimmer" le tableau doit être gris.

6.2.1.2. Calibration du gyroscope

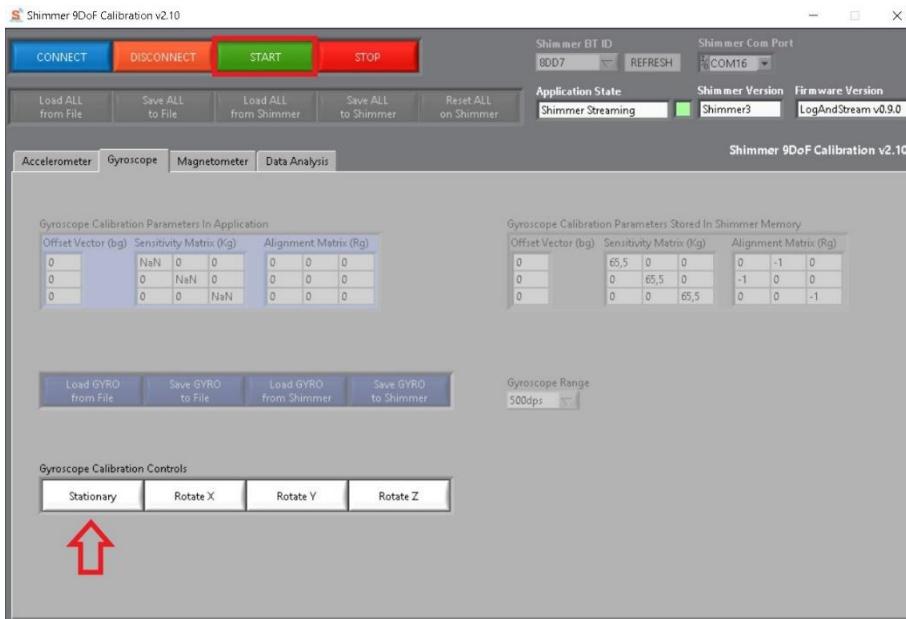
Sélectionnez l'onglet "Gyroscope".

Vous pouvez aussi sélectionner la gamme du gyroscope selon votre application.



Cliquez sur **START**.

Posez le shimmer sur une surface plane. Cliquez sur **stationnaire**. Ne touchez pas au shimmer ni à la surface, il ne doit avoir aucun mouvement ni vibration. Attendez jusqu'à la disparition du **wait**.



Maintenant on va commencer à rotationner le shimmer.

Il est recommandé d'utiliser un objet ou une surface de référence pour appuyer le shimmer pendant la calibration.

Pour l'axe x, vous devez faire attention de tourner le shimmer dans le sens de l'horloge, avec l'axe x qui est directionné vers l'horloge.

*botar imagem do eixo x entrando no rélojio

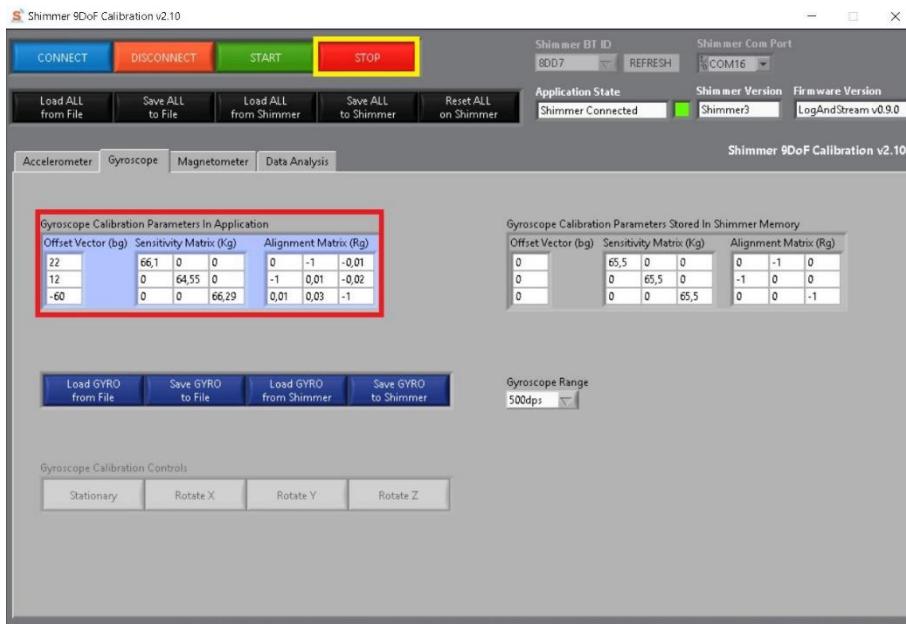
Avant commencer la rotation, cliquez sur "**Rotation X**". Vous devez faire une rotation complète (360 degrés) en vous garantissant de placer le shimmer exactement dans la même position du début. Pour cela, vous pouvez l'appuyer sur la surface de référence de façon à simplifier le travail. Vous pouvez déplacer linéairement le shimmer, mais vous ne devez pas le tourner dans les autres axes.

Pour mieux expliquer comment faire la rotation du shimmer, une [vidéo démonstrative](#) est disponible.

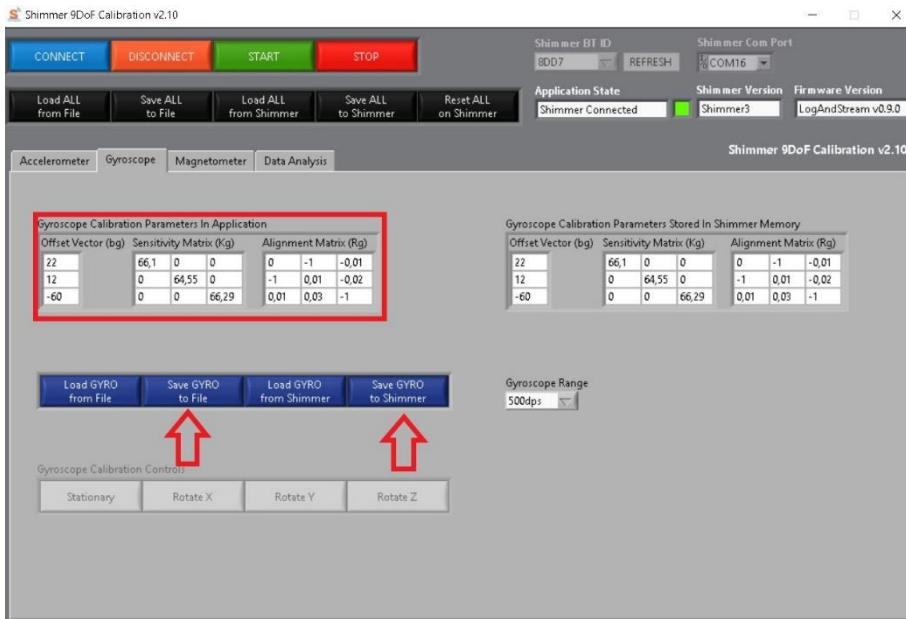
Après rotationner le shimmer, cliquez sur **Press to Stop**.

Vous devez répéter la rotation pour les trois axes. Vous pouvez réaliser les rotations dans l'ordre que vous souhaitez et autant de fois que vous trouvez pertinent.

Cliquez sur **STOP** en rouge pour arrêter la transmission de données.



Après avoir fait la rotation pour les trois axes, cliquez sur "**Save GIRO to Shimmer**" pour envoyer la calibration du gyroscope vers le shimmer. Le tableau de paramètres de calibration doit passer de bleu à gris. De plus, si vous voulez enregistrer la calibration sur votre ordinateur, cliquez sur "**Save GIRO to file**".



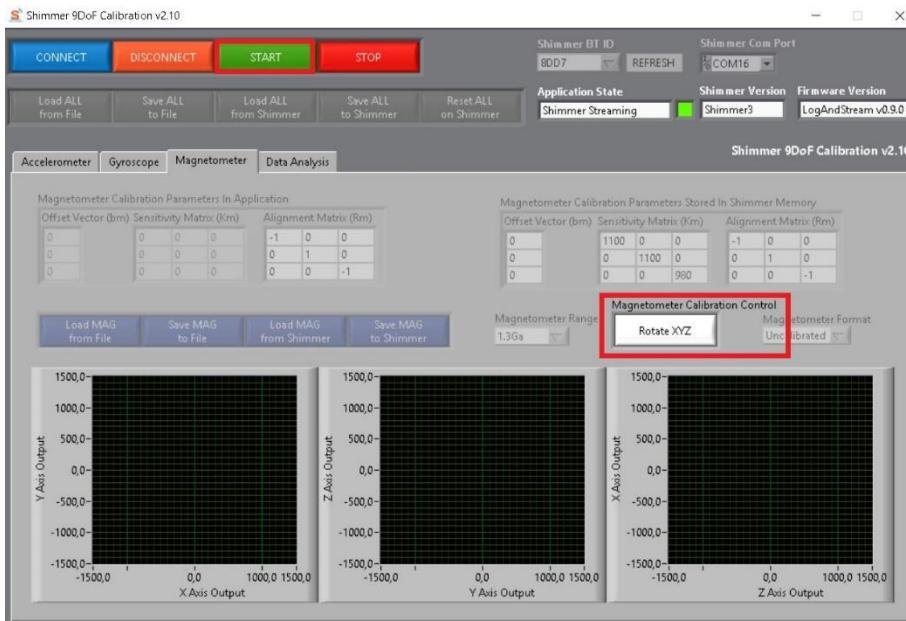
6.2.1.3. Calibration du magnétomètre

Sélectionnez l'onglet "Magnetometer".

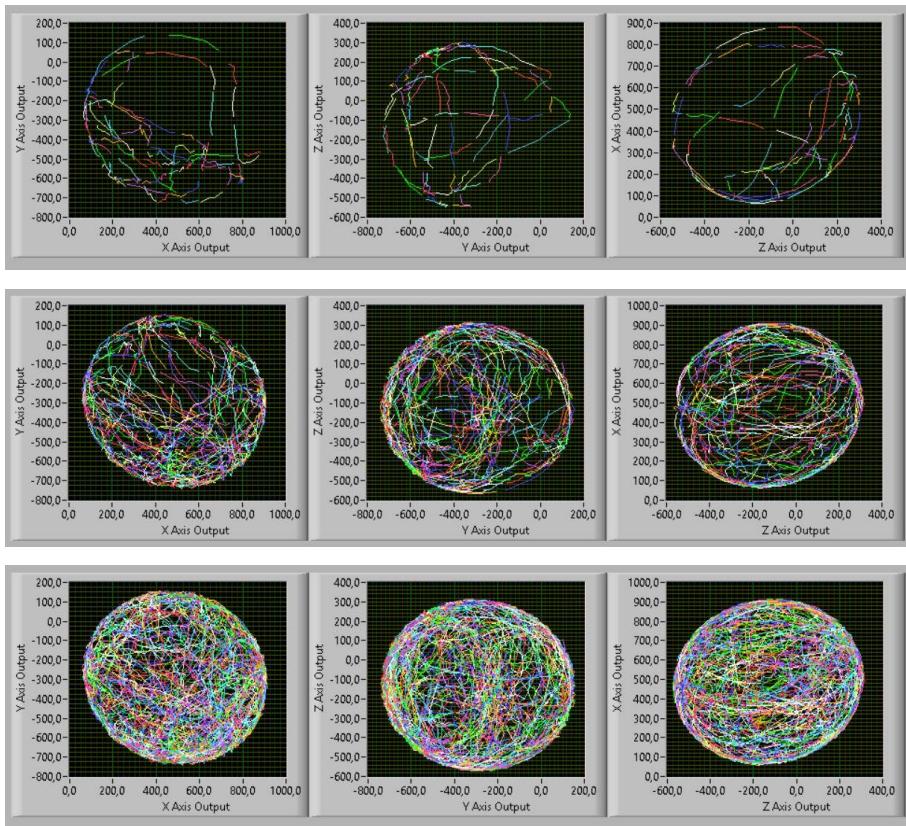
Vous pouvez aussi sélectionner la gamme du magnétomètre selon votre application.



Cliquez sur **START**. Cliquez sur **Rotation XYZ**.



Tournez le shimmer aléatoirement dans toutes les directions possibles. Vous devez obtenir progressivement dans les trois graphiques qui apparaissent quelque chose qui ressemble de plus en plus à une sphère. Vous pouvez vous arrêter quand les trois graphiques semblent beaucoup à une sphère (cf. images ci-dessous).



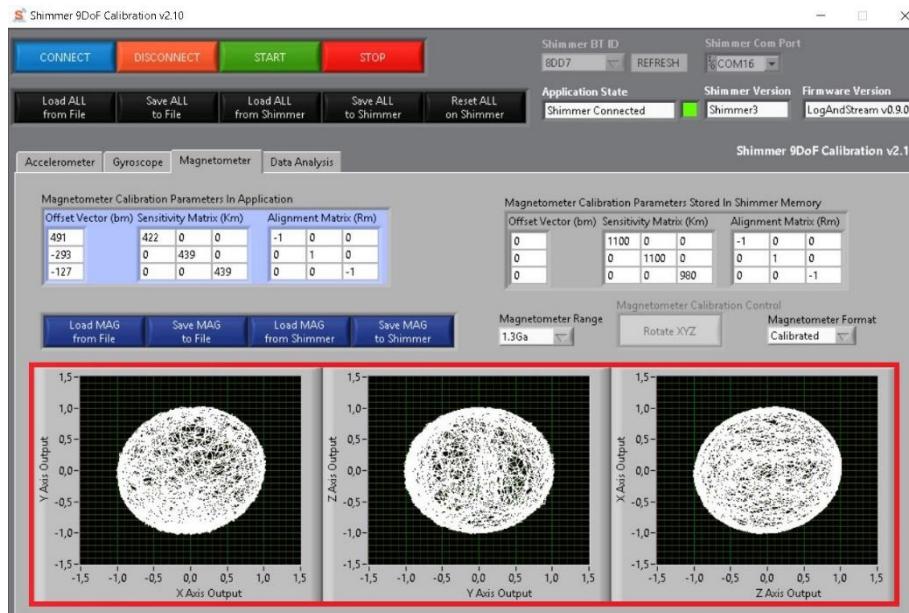
Quand vous trouvez que la calibration est suffisante, appuyez sur **Press to Stop**.

Cliquez sur **STOP**.

Pour vérifier la calibration, changez le "**Magnetometer format**" d'**Uncalibrated** par **Calibrated**.



Si la calibration est bien faite, les graphiques doivent être de sphères presque parfaits centrés en (0,0,0).



Cliquez sur "**Save MAG to Shimmer**" pour envoyer la calibration du magnétomètre vers le shimmer. Le tableau de paramètres de calibration doit passer de bleu à gris. De plus, si vous voulez enregistrer la calibration sur votre ordinateur, cliquez sur "**Save MAG to file**".

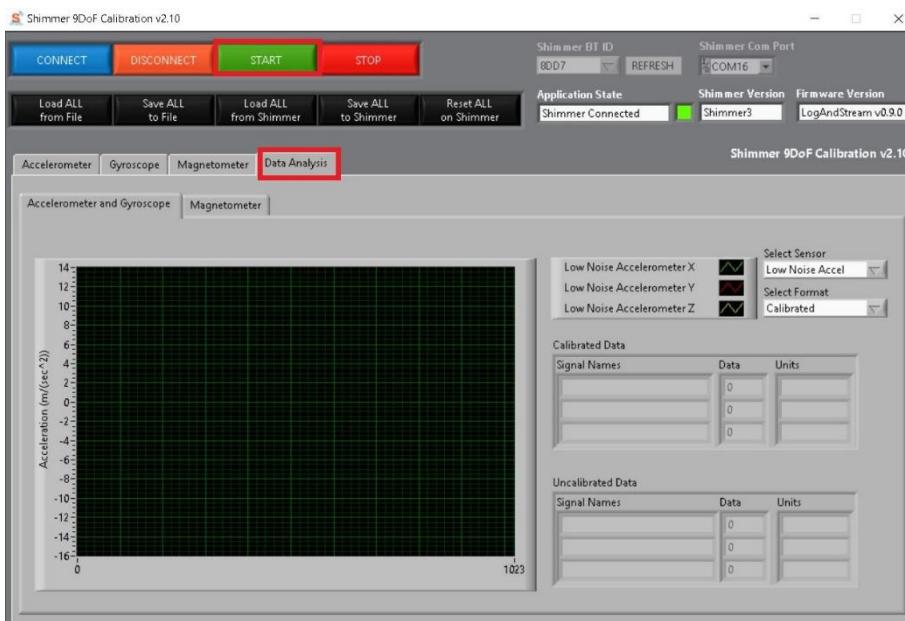
NB : Après avoir fait tous les étapes de calibration, vous pouvez appuyer sur **Save ALL to file** pour enregistrer la calibration complète sur votre ordinateur.

6.2.1.4. Data Analysis

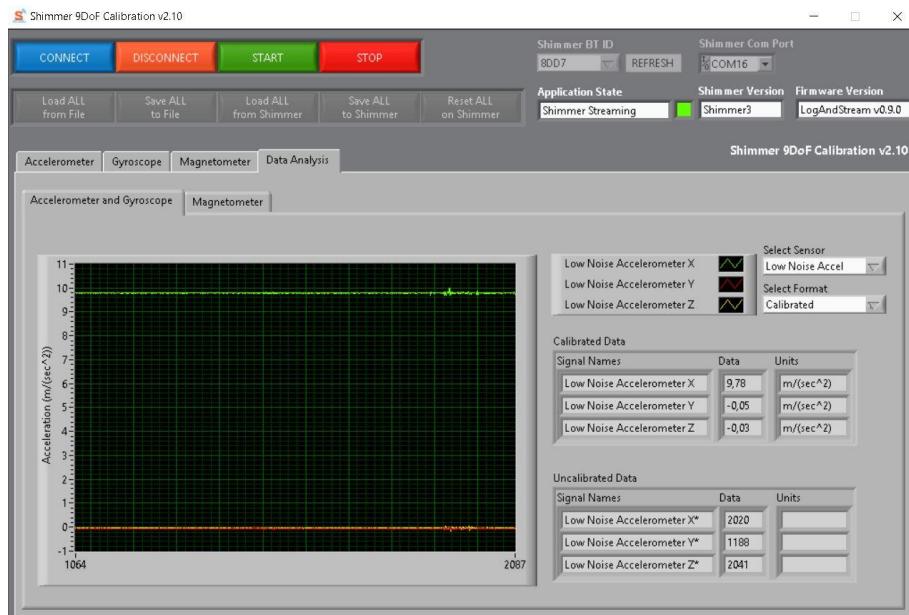
Maintenant, vous allez vérifier la justesse de la calibration. Sélectionnez l'onglet **Data Analysis**. Appuyez sur **START**.

Accéléromètre

Par défaut, vous devez avoir "Accelerometer" sélectionné sur "Select Sensor". S'il n'est pas le cas, changez-le par "Accelerometer". Posez le shimmer avec l'axe x vers le haut.



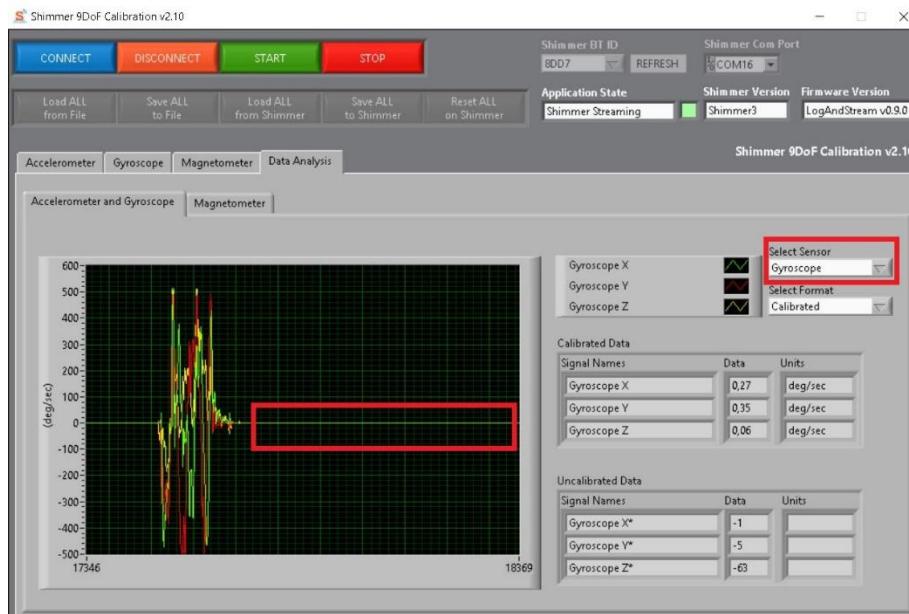
Vous devez observer dans le graphique que l'accélération pour cet axe est d'environ 10 m/s² (qui correspond à la pesanteur) et pour les axes y et z les accélérations sont nulles.



Répétez cette méthode de vérification de la pesanteur pour y et z.

Gyroscope

Sélectionnez le gyroscope sur "Select Sensor". Si la calibration est correcte, le signaux sur les graphiques doivent être nuls quand le shimmer est stationnaire.



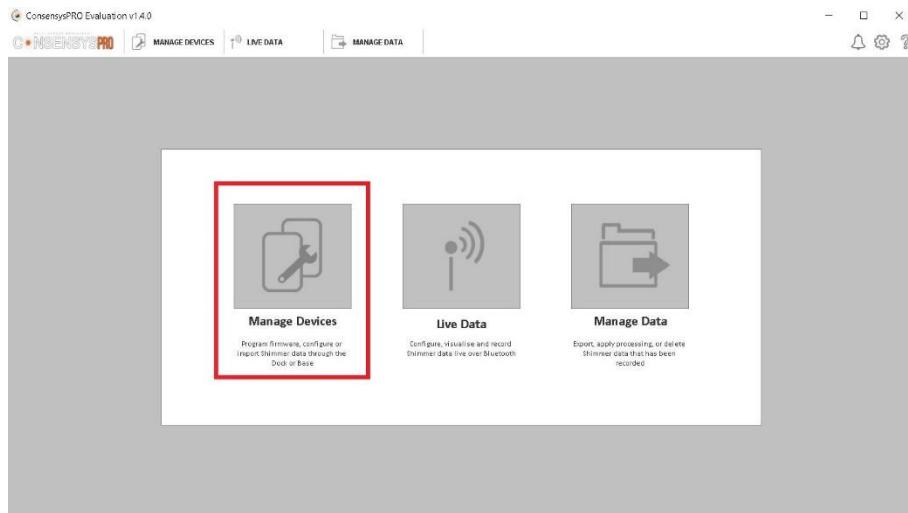
Magnetometer

Sélectionnez le magnétomètre sur "**select sensor**".

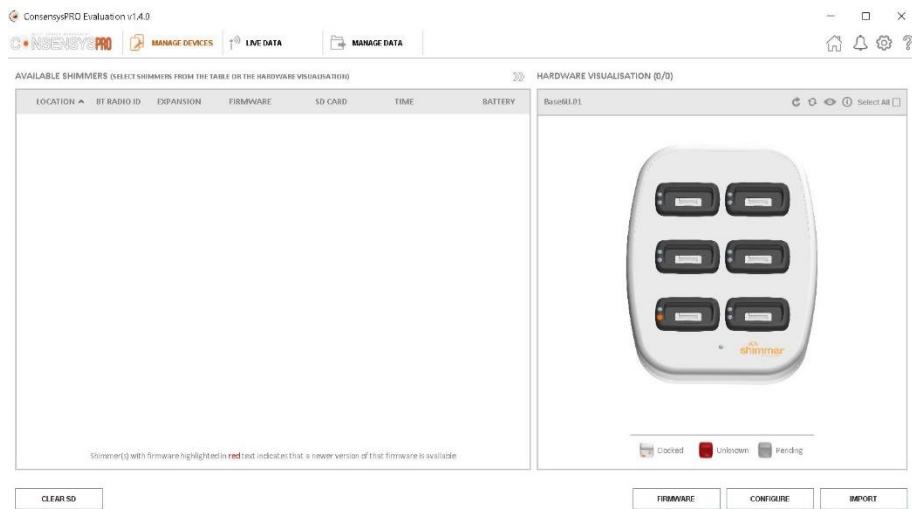
Dans le cas du magnétomètre, lorsque un des axes du shimmer est positionnée parallèlement au vecteur du champ magnétique il va mesurer 1 (avec fluctuations autour de cette valeur) dans cet axe et 0 dans les autres. Il faut remarquer que ce vecteur n'est pas parallèle à la surface de la planète, il y a un angle d'inclinaison qui existe.

6.2.2. Utilisation

Ouvrez le programme *Consensys*. Sur le menu qui apparaît, vous devez cliquer sur **Manage Devices**.



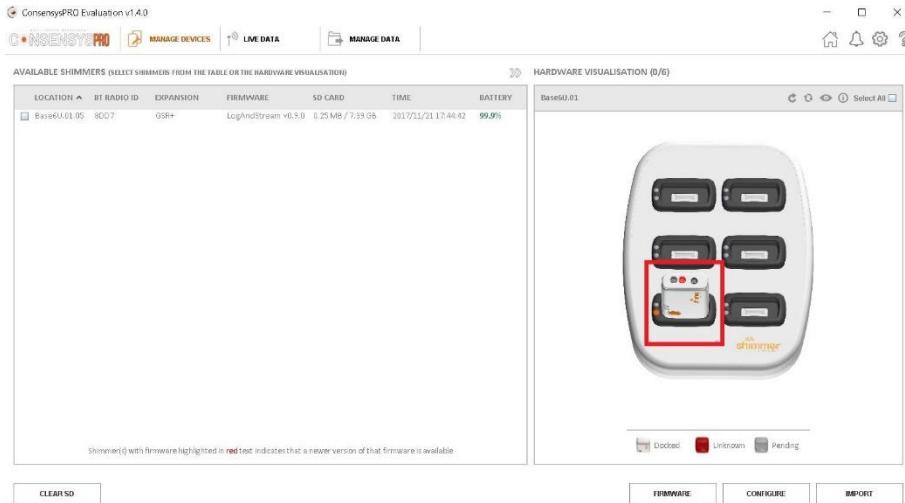
La fenêtre va ressembler à :



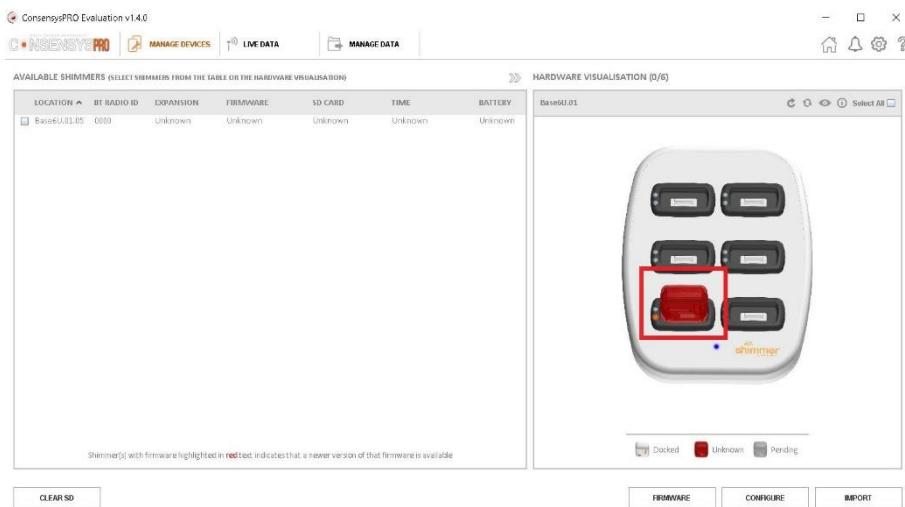
Connectez le shimmer que vous voulez utiliser à la base. Il doit être allumé avant d'être branché.



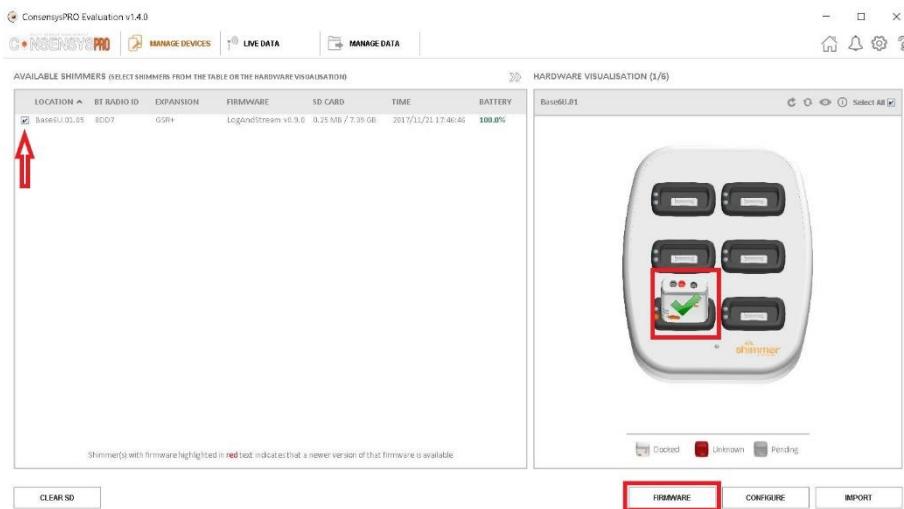
Attendez le shimmer apparaître sur l'image de la base. Cela veut dire que le programme a bien identifié le shimmer.



Quand le programme ne reconnaît pas le shimmer, un capteur en rouge apparaît au lieu de l'image du shimmer.

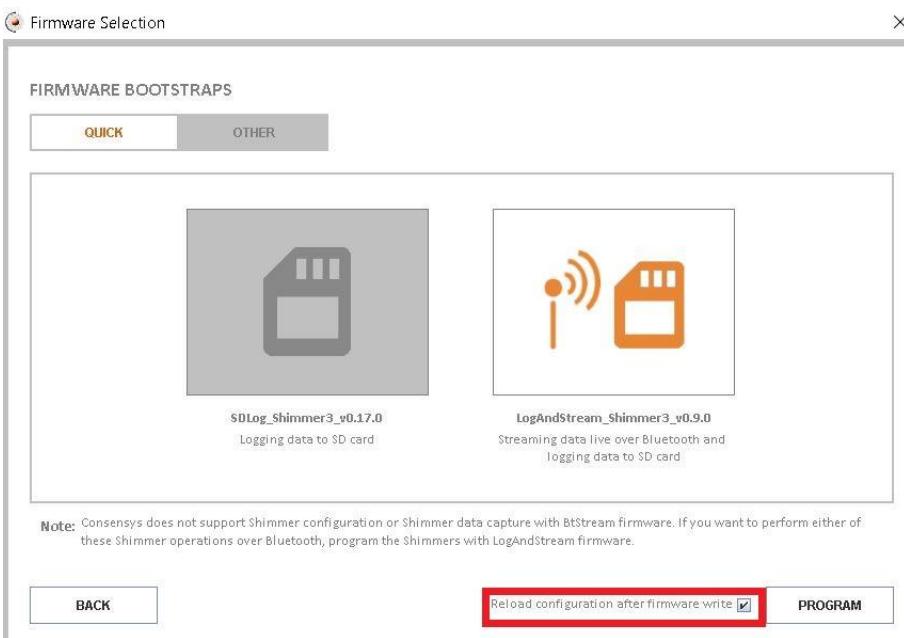


Couchez la case pour sélectionner tous les shimmers que vous voulez configurer.
Cliquez sur **FIRMWARE**.

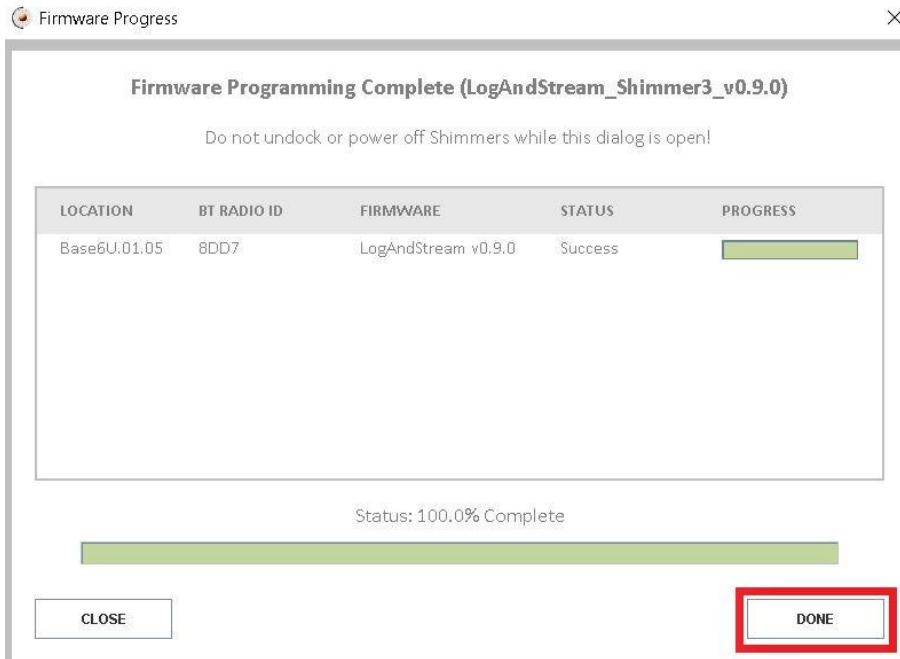


Une fenêtre va apparaître. Vous pouvez donc sélectionner la méthode de capture (enregistrement dans le SD card ou *streaming* par Bluetooth).

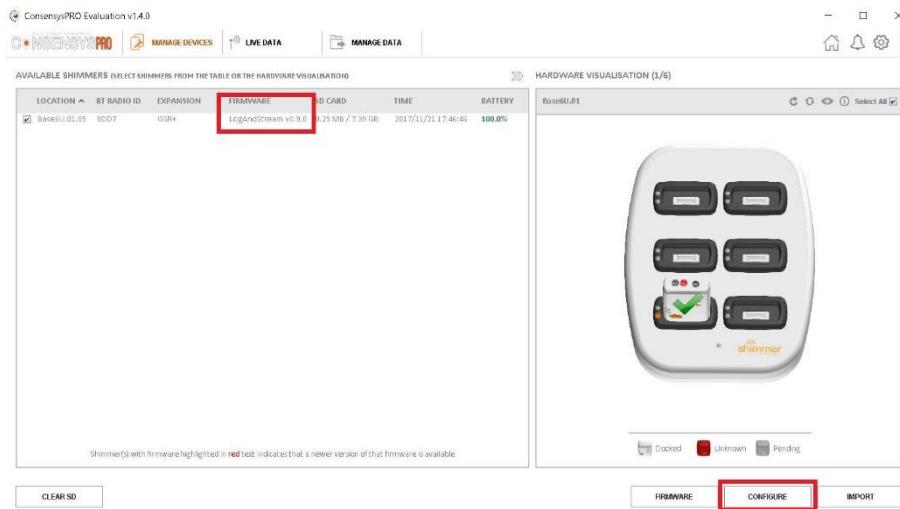
IMPORTANT ! Si votre shimmer est déjà calibré, cochez la case à côté de **PROGRAM**.



Cliquez sur **PROGRAM**. Une nouvelle fenêtre va apparaître. Attendez jusqu'à ce que le status soit 100% complet. Cliquez ensuite sur **DONE**.



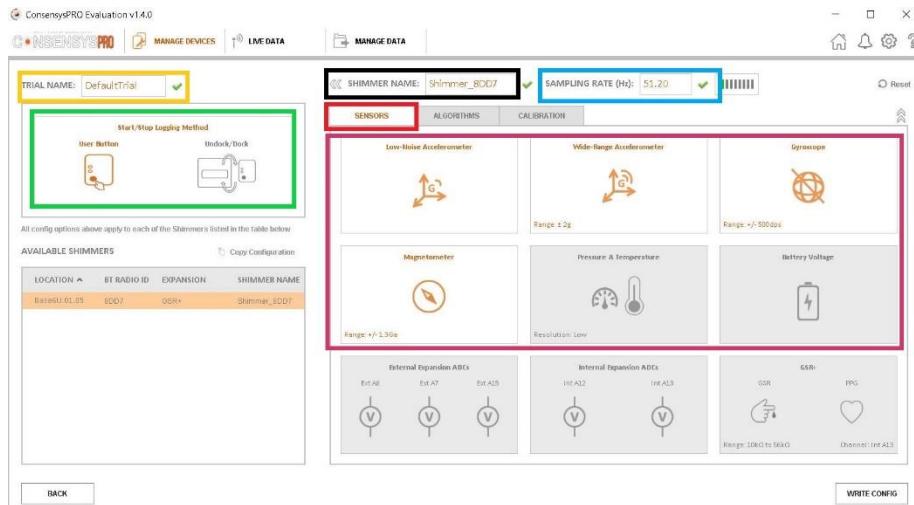
Vous pouvez vérifier si le firmware a été bien installé. Cliquez sur **CONFIGURE**.



Sur l'onglet **SENSORS** (en rouge) :

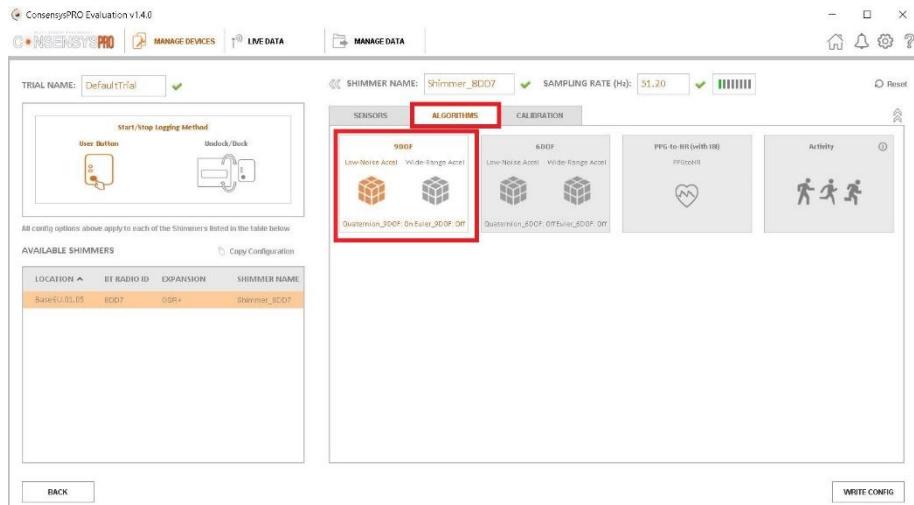
Vous pouvez insérer le nom de l'essai si vous voulez (en jaune). Vous devez sélectionner la méthode d'activation et d'arrêt de la capture de données (en vert). Vous pouvez aussi donner un nom pour le shimmer (en noir). Choisissez la fréquence

du shimmer (en bleu). Sélectionnez quel type de capteur vous voulez utiliser (accéléromètres, gyroscope, etc.) (en rose).



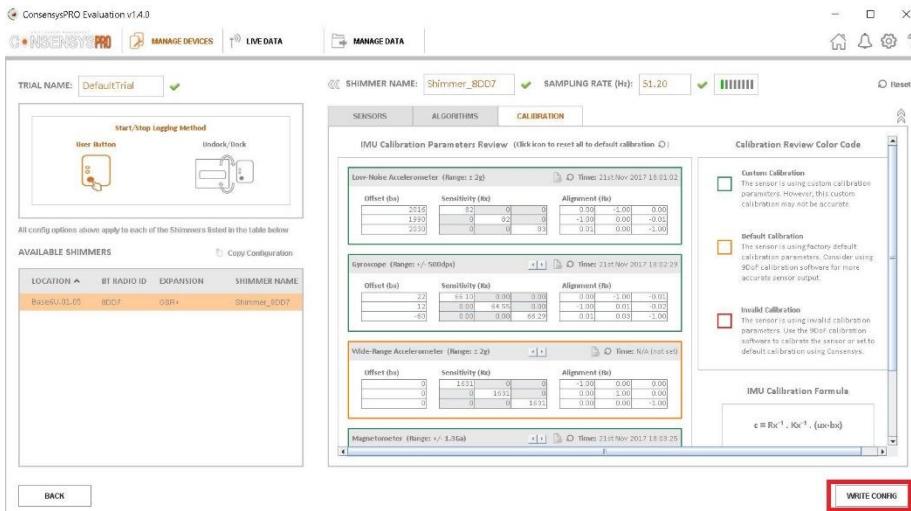
Sur l'onglet **ALGORITHMS** :

Sélectionnez l'algorithme que vous voulez utiliser. Il faut choisir 9DOF pour le Shimmer3 IMU.



Sur l'onglet **CALIBRATION** :

Vérifiez que la calibration correspond à celle réalisée auparavant. Cliquez sur **WRITE CONFIG**.

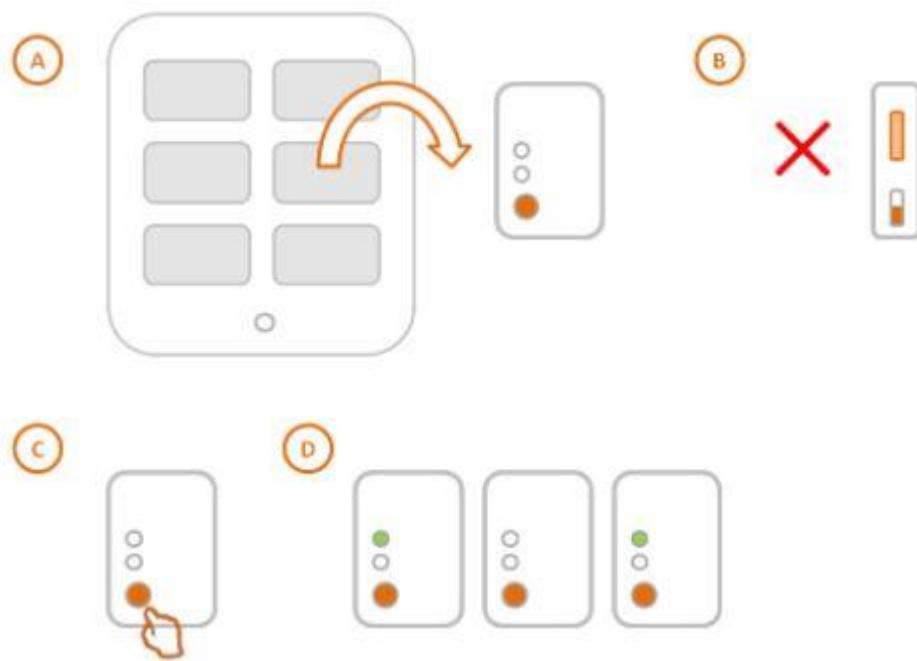


Attendez que la configuration soit complète et cliquez sur **NEXT**.



6.2.2.1. Capture

Déconnectez le shimmer de la base (sans l'éteindre). Appuyez sur le bouton orange pour lancer l'acquisition (le LED bleu va s'allumer à chaque seconde).



Pour arrêter l'acquisition, appuyez à nouveau sur le bouton orange. Le LED vert va s'allumer à chaque 2 secondes.

6.2.2.2. Importation et Exportation

Reconnectez le shimmer à la base. Cliquez sur **Import**.

LOCATION	BT RADIO ID	EXPANSION	FIRMWARE	SD CARD	TIME	BATTERY
<input checked="" type="checkbox"/> BaseU.01.05	ECDT	GSR+	LogonStream v0.0	0.25 MB / 7.39 GB	2017/11/21 17:46:46	100.0%

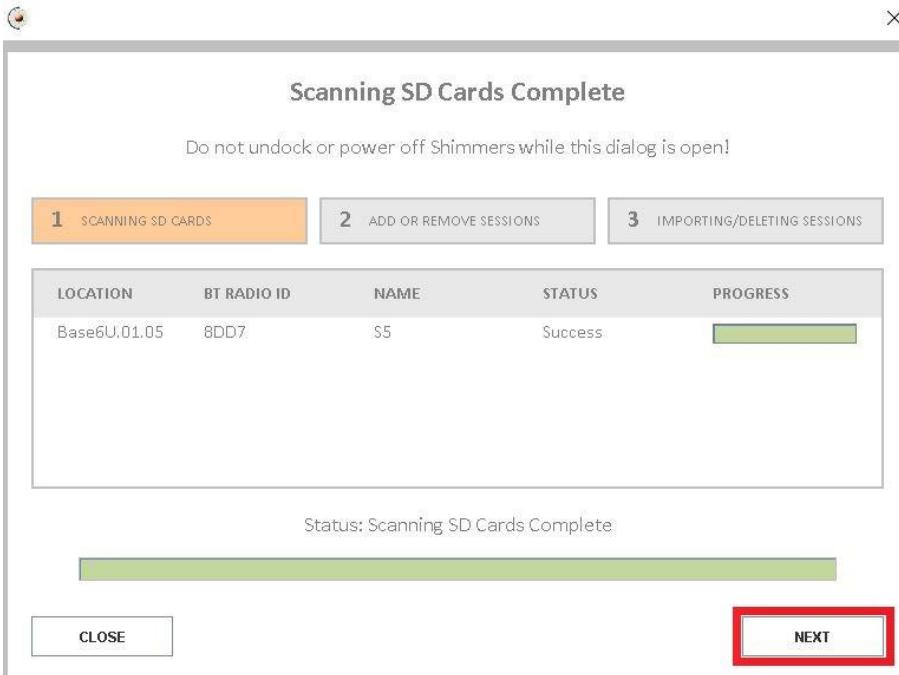
Shimmer(s) with firmware highlighted in red text indicate that a newer version of that firmware is available.

HARDWARE VISUALISATION (1/6)

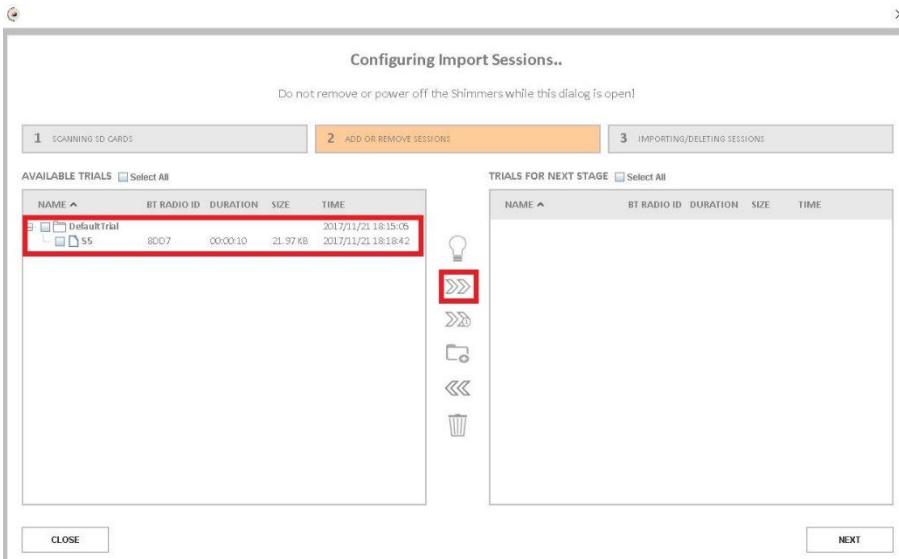
Docked Unknown Pending

CLEAR SD FIRMWARE CONFIGURE IMPORT

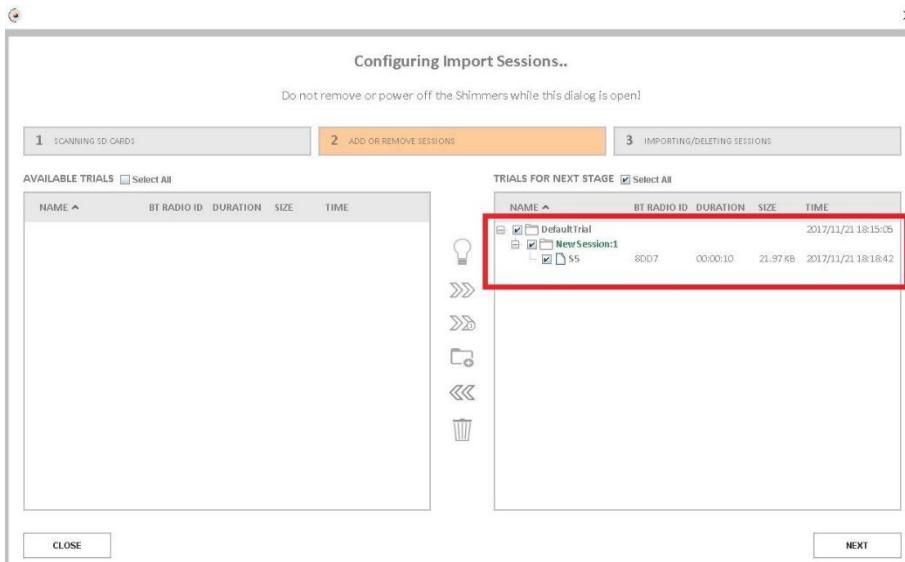
Attendez la fin du *Scanning* et ensuite cliquez sur **NEXT**.



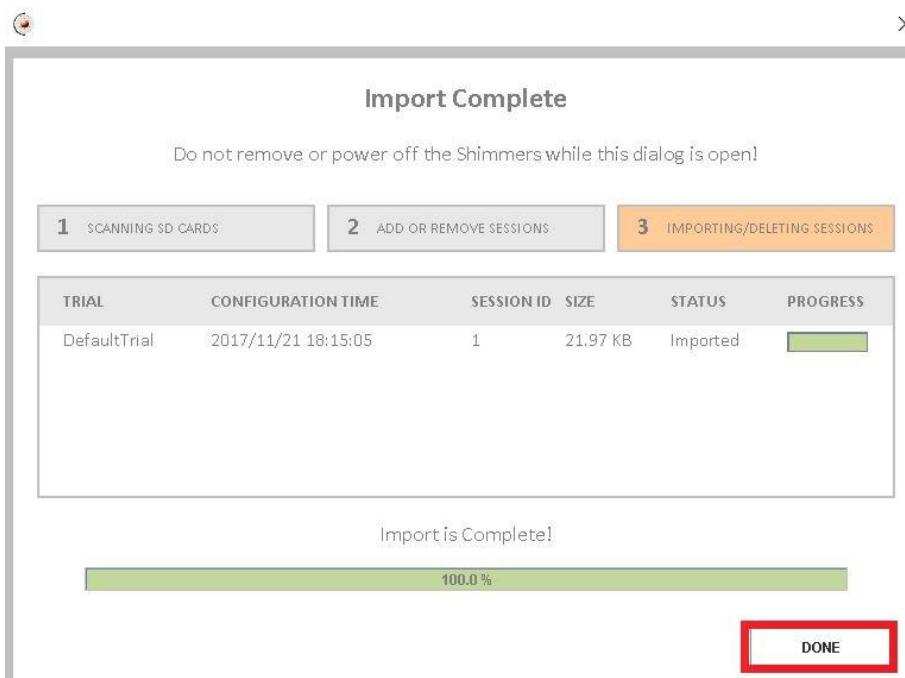
Sélectionnez l'essai que vous voulez importer du shimmer. Cliquez sur la flèche pour passer l'essai à la deuxième colonne.



Sélectionnez les données que vous voulez traiter dans la prochaine étape et puis cliquez sur **NEXT**.



Attendez la fin de l'importation et cliquez sur **DONE**.



Une nouvelle fenêtre s'ouvrira. Vous pouvez l'ouvrir aussi à partir du menu qui apparaît lors de l'ouverture du programme en cliquant sur *Manage data*.

Sélectionnez les données à exporter et puis cliquez sur la flèche en bas.

ConsensysPRO Evaluation v1.4.0

MANAGE DEVICES | LIVE DATA | MANAGE DATA

AVAILABLE DATA (SELECT A DATASET FROM THE TABLE BELOW)

NAME	SYNC	RTC	TIME	DURATION	SIZE	CONFIG	REPORTS
SampledCAPP			2015/06/23 16:17:26	00:02:00	1.87 MB		
SampledCQ			2015/06/23 16:26:46	00:02:00	1.00 MB		
SampledMG			2015/06/23 16:25:39	00:02:00	4.56 MB		
SampledOF_W			2015/06/26 17:58:14	00:01:00	7.98 MB		
SampledOF_R			2015/06/26 18:17:14	00:01:00	7.98 MB		
Sampledyc_SD			2015/06/09 21:58:32	00:02:25	32.77 MB		
SampledEvents			2016/04/07 09:15:52	00:01:35	4.49 MB		
SampledRep			2016/07/11 21:14:21	00:02:23	16.70 MB		
SampledOF_Y			2017/04/11 10:16:40	00:32:25	20.00 MB		
DefaultTrial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2017/11/21 18:15:05	00:00:30	23.00 KB		
SD Recording	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2017/11/21 18:18:43	00:00:10	21.97 KB		
Session 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2017/11/21 18:18:43	00:00:10	21.97 KB		
55 - 100.21Hz - 99.8%	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					

Channels with a * after their name have been calibrated using default calibration parameters

DATA DESCRIPTIONS (INSERT DESCRIPTIONS FOR TRIALS/SSESSIONS)

DefaultTrial - 2017/11/21 18:15:05

Insert Trial Description Here.

SD Recording - Session 1

Insert Session Description Here.

SAVE

DELETE EXPORT PROCESS

Vous devez maintenant choisir comment vous voulez exporter les données. Si vous choisissez "Formatted Local" sur "Timestamp Format", vous auriez vos données avec l'heure correspondante.

ConsensysPRO Evaluation v1.4.0

MANAGE DEVICES | LIVE DATA | MANAGE DATA

AVAILABLE DATA (SELECT A DATASET FROM THE TABLE BELOW)

NAME	SYNC	RTC	TIME	DURATION	SIZE	CONFIG	REPORTS
SampledCAPP			2015/06/23 16:17:26	00:02:00	1.87 MB		
SampledCQ			2015/06/23 16:26:46	00:02:00	1.00 MB		
SampledMG			2015/06/23 16:25:39	00:02:00	4.56 MB		
SampledOF_W			2015/06/26 17:58:14	00:01:00	7.98 MB		
SampledOF_R			2015/06/26 18:17:14	00:01:00	7.98 MB		
Sampledyc_SD			2015/06/09 21:58:32	00:02:25	32.77 MB		
SampledEvents			2016/04/07 09:15:52	00:01:35	4.49 MB		
SampledRep			2016/07/11 21:14:21	00:02:23	16.70 MB		
SampledOF_Y			2017/04/11 10:16:40	00:32:25	20.00 MB		
DefaultTrial	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2017/11/21 18:15:05	00:00:30	23.00 KB		
SD Recording	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2017/11/21 18:18:43	00:00:10	21.97 KB		
Session 1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2017/11/21 18:18:43	00:00:10	21.97 KB		
55 - 100.21Hz - 99.8%	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					

Channels with a * after their name have been calibrated using default calibration parameters

DATA DESCRIPTIONS (INSERT DESCRIPTIONS FOR TRIALS/SSESSIONS)

DefaultTrial - 2017/11/21 18:15:05

Insert Trial Description Here.

SD Recording - Session 1

Insert Session Description Here.

SAVE

DELETE EXPORT PROCESS

File Format: CSV
Timestamp Format: Calibrated

Cliquez sur **EXPORT** et enregistrez le fichier sur votre ordinateur.