

**RAPPORT DE
STAGE INGÉNIEUR**

**Mise en place d'un banc de test pour
caractériser chaque Virtuose 6D**

Auteur :
KOUTIT ABDELLAH

Entreprise :
HAPTION

Tuteur_Ecole :
M. ABDELHAMID CHRIETTE

Encadrant_Entreprise :
M. MOHAMAD DHAYBI

12 septembre 2023

Table des matières

Remerciements	4
1 Introduction	5
2 Entreprise : Haption	6
2.1 Organisation de l'entreprise	6
2.2 Présentation des produits	7
2.2.1 Le Virtuose	8
2.2.2 Le Scale1	9
3 Contexte du stage	10
3.1 Besoin de vérification en usine	10
3.1.1 Vérification des positions	10
3.1.2 Vérification des efforts	10
3.2 Besoin de vérification chez le client	11
3.2.1 Vérification des débattements	11
3.2.2 Vérification des efforts	11
4 Vérification des positions (Client)	11
4.1 Test JointsLimits	11
4.2 Butées logicielles	13
4.3 Test Stylo_Calibration	14
4.3.1 conception d'un support stylo	14
4.3.2 Optimisation :	16
4.4 Test Points	19
5 Vérification des positions (Usine)	21
5.1 Localisation par caméras	21
5.1.1 Calibration de la pièce	23
5.1.2 Calibration du corps	25
5.1.3 L'ajustement du corps	28
5.1.4 La sortie des données de mesure via Ethernet	30
5.2 Comparaison Caméras-Virtuose	32
5.2.1 Détermination de l'offset marqueur	33
5.2.2 Détermination de la matrice de transformation	35
5.2.3 Résultat de la comparaison	38

6 Vérification des effort	41
6.1 Vérification chez le client	41
6.2 Vérification en usine	41
6.2.1 Test ForceSensor	41
6.2.2 Résultats du test	43
7 Identification des coefficients de frottement	46
7.1 Simplification du problème :	46
7.2 Modèle dynamique inverse simplifié :	47
7.2.1 L'ajout du frottement	48
7.3 Identification des coefficients	49
7.3.1 Lissage des données	50
8 Conclusion	56
Annexe	57
Quelques fonctions utiles	57
Plan du banc de test	61

Table des figures

2.1 Vituose 6D	8
2.2 le Scale1	9
4.1 Joints Limits	12
4.2 Espace atteignable	12
4.3 butée : logicielle-mécanique	13
4.4 Poignée stylo	14
4.5 Connecteur	15
4.6 État finale poignée	15
4.7 Plan de câblage	16
4.8 Position end_effector par rapport aux CRI	18
4.9 déviation standard (m)	19
4.10 Output Test	20
5.1 Principe du tracking optique	22
5.2 Processus de pièce-calibration	24
5.3 Résultat de la pièce-calibration	25
5.4 Cible virtuose	26
5.5 Géométrie de la cible	27
5.6 Définition du système coordonnées corps	28

5.7 Ajustement du corps	30
5.8 DTrack2SDK outputs	32
5.9 Avant calibration	34
5.10 Après calibration	35
5.11 Avant calibration	37
5.12 Après calibration	38
5.13 erreurs en position (m)	39
5.14 Caméras vs Virtuose	40
6.1 TestBlockedMode	41
6.2 Poignée Capteur	42
6.3 Test ForceSensor	43
6.4 Force (N) : Vituose Vs Sensor	44
6.5 Erreur en force (N)	44
6.6 Torque (Nm) : Vituose Vs Sensor	45
6.7 Erreur en torque (Nm)	46
7.1 Pivot 1dof	47
7.2 frottement de Coulomb	49
7.3 Couple en fonction de l'angle	50
7.4 Données après lissage	51
7.5 Différence entre torque aller et retour	52
7.6 $d(\omega)$ en fonction du temps	53
7.7 Résultats d'identification	54
7.8 Résultats d'identification pour un high force	55

Remerciements

Au terme de cette expérience passionnante, je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué à la réalisation de ce stage. Le chemin parcouru n'aurait pas été possible sans leur soutien inestimable, leurs conseils éclairés et leur encouragement constant.

Tout d'abord, je tiens à remercier chaleureusement mon maître de stage, *M. Mohamad Dhaybi*, pour sa précieuse guidance tout au long de cette aventure professionnelle. Votre expertise, votre disponibilité et votre engagement à partager vos connaissances ont grandement enrichi cette expérience. Vos retours constructifs et vos encouragements ont été d'une importance capitale pour mon apprentissage.

Je souhaite également exprimer ma reconnaissance envers *M. François Louveau*, chef de l'équipe d'étude, pour avoir défini le sujet de stage et pour sa contribution à cette expérience. Votre vision et votre direction ont été essentielles pour orienter ce projet dans la bonne direction.

Je tiens également à exprimer ma reconnaissance envers toute l'équipe *HAPTION* pour m'avoir accueilli au sein de leur environnement de travail. Votre accueil chaleureux, votre esprit d'équipe et votre engagement envers l'excellence ont créé un cadre propice à l'apprentissage et à la découverte.

Un grand merci à mes collègues, pour leur camaraderie, leur collaboration et les échanges fructueux qui ont contribué à rendre cette expérience si enrichissante. Les discussions stimulantes et les partages d'expérience ont grandement contribué à mon développement professionnel.

Je n'oublie pas de remercier mes enseignants et professeurs, qui m'ont transmis les connaissances fondamentales qui ont servi de socle à ce stage. Votre enseignement a été une boussole tout au long de cette expérience.

Enfin, un immense merci à ma famille et à mes amis pour leur soutien indéfectible, leur compréhension et leurs encouragements constants. Votre appui a été ma source de motivation tout au long de cette aventure.

En somme, je suis profondément reconnaissant envers chacune des personnes qui ont contribué de près ou de loin à la réalisation de ce stage. Votre confiance, vos conseils et vos encouragements ont été les piliers qui ont soutenu ce projet. Cette expérience restera ancrée dans ma mémoire, et je suis impatient de poursuivre mon parcours professionnel en mettant en pratique les enseignements que j'ai acquis grâce à vous tous.

1 Introduction

Le domaine de la vérification et de la validation des dispositifs haptiques dans l'industrie est d'une importance cruciale pour garantir la performance, la fiabilité et la satisfaction des utilisateurs. Ce rapport de stage se plonge dans l'exploration approfondie des besoins de vérification, à la fois en usine et chez le client, en ce qui concerne les débattements et les efforts générés par les dispositifs haptiques. Ces dispositifs, tels que le Virtuose et le Scale1, offrent une expérience utilisateur immersive en recréant des sensations tactiles. Par conséquent, leur précision et leur conformité aux spécifications techniques sont essentielles pour assurer leur efficacité et leur adéquation à leur application.

Pour bien comprendre les enjeux de la vérification des dispositifs haptiques, il est nécessaire de se pencher sur la structure de l'entreprise Haption. Dans la première section, nous explorerons l'organisation de cette entreprise innovante, ainsi que la gamme de produits qu'elle propose. Cette présentation préalable est essentielle pour avoir une vue d'ensemble des outils et des technologies impliqués dans notre étude.

La deuxième section examinera le contexte général du stage. Il est important de souligner que la vérification des dispositifs haptiques doit être abordée à la fois en interne, en usine, et chez les clients utilisant ces dispositifs dans leur environnement. Notre rapport abordera donc les besoins de vérification à ces deux niveaux. Nous nous concentrerons sur la vérification des positions et des efforts, en soulignant les différentes exigences et méthodologies associées à chacun de ces aspects.

La troisième section de ce rapport se plongera dans les détails de la vérification des positions chez le client. Nous discuterons des tests spécifiques, tels que JointsLimits, les butées logicielles, et la calibration du stylo. L'objectif ici est de comprendre comment les dispositifs haptiques sont évalués en termes de positionnement, garantissant ainsi une précision maximale.

La quatrième section se concentrera sur la vérification des positions en usine. Cette étape est essentielle pour s'assurer que les dispositifs haptiques sont calibrés de manière optimale et qu'ils fonctionnent conformément aux spécifications. Nous explorerons des techniques telles que la localisation par caméras et la comparaison entre les données de caméras et celles du Virtuose pour valider la précision des positions.

La cinquième section abordera la vérification des efforts, à la fois chez le client et en usine. Il est crucial de garantir que les dispositifs haptiques génèrent les efforts appropriés pour créer des sensations tactiles réalistes. Nous discuterons des tests tels que ForceSensor et examinerons les résultats obtenus.

Enfin, dans la sixième section, nous aborderons un aspect plus complexe de la vérification, à savoir l'identification des coefficients de frottement. Cette étape est essentielle pour comprendre comment les forces sont générées et dissipées dans les dispositifs haptiques. Nous expliquerons en détail les méthodes utilisées pour déterminer ces coefficients, ainsi que leur importance pour la modélisation et la simulation des dispositifs haptiques.

En résumé, ce rapport se penche sur un aspect critique de l'industrie des dispositifs haptiques. Grâce à une compréhension approfondie des besoins de vérification en usine et chez le client, ainsi que des méthodologies spécifiques utilisées pour évaluer les positions et les efforts, nous espérons contribuer à l'amélioration de la qualité et de la performance des dispositifs haptiques, garantissant ainsi une expérience utilisateur exceptionnelle.

2 Entreprise : Haption

Haption est une entreprise spécialisée dans la conception et la fabrication de dispositifs de retour haptique pour les applications de réalité virtuelle (RV) et de réalité augmentée (RA). L'un de leurs projets phares est le Virtuose, un dispositif de retour haptique avancé qui offre une expérience immersive et réaliste aux utilisateurs.

2.1 *Organisation de l'entreprise*

L'entreprise est composée de différents départements, notamment le pôle de production et contrôle en mécanique et électronique, le bureau d'études (avec des parties dédiées à l'électronique, à la mécanique et à l'informatique) et le pôle commercial.

1. Bureau d'études - Mécanique :

- Responsable de la conception des pièces en 3D pour les bras Virtuose.
- Sélection des matériaux en fonction des exigences du client.
- Vérification des forces de frottement, des efforts, des résistances et des coûts.
- Objectif : répondre aux demandes du client en tenant compte des contraintes d'usinage et d'assemblage mécanique et électronique.

2. Production - mécanique :

- Chargé de la fabrication des systèmes mécaniques composant les bras Virtuose.

- Certaines pièces sont produites en interne, tandis que d'autres sont sous-traitées.
- Montage et équilibrage des axes des bras.

3. Bureau d'études - Électronique :

- Conception des systèmes embarqués, des plans de câblage et des cartes électroniques.
- Implémentation des conceptions dans les bras de robots.
- Activités simultanées avec le montage mécanique.
- Vérification pour détecter les anomalies potentielles dues à un mauvais montage, à une pièce défectueuse ou à une carte électronique défectueuse.

4. Production - Électronique :

- Connexions électriques et électroniques :
- Intégration harmonieuse des câblages avec les composants mécaniques et électroniques des bras.
- Contrôle de qualité et résolution des problèmes

5. Pôle informatique :

- Équipe de quatre développeurs chargés de la réalisation de logiciels pour optimiser l'utilisation des bras Virtuose.
- Développement d'API permettant la téléopération.
- Validation logicielle effectuée dans ce pôle.

6. Pôle commercial :

- Gestion des marchés en Europe, aux États-Unis et en Asie.
- Présentations et démonstrations lors de forums, expositions et chez les clients.
- Responsable des ventes des produits de la société Haption.

2.2 Présentation des produits

Haption propose des produits Hardware, une gamme d'interfaces haptiques : Une interface haptique est un système maître qui permet à son utilisateur d'interagir avec un système esclave dans un monde virtuel ou un monde réel à distance. Elle permet aux utilisateurs de concevoir, modeler et manipuler des objets dans un environnement extérieur (virtuel ou réel) avec un certain ressenti tactile et la perception kinesthésique, c'est le retour de force qui est utilisé le plus souvent comme retours sensoriels. Une telle interface est généralement constituée d'une structure mécanique articulée, similaire à celle des robots, actionnée par des moteurs et munie d'une poignée pour que l'opérateur puisse la manipuler.

2.2.1 *Le Virtuose*

Le Virtuose est un dispositif de retour haptique conçu pour simuler des sensations tactiles et haptiques réalistes lors de l'interaction avec des environnements virtuels. Il est composé d'un exosquelette haptique pour la main et le bras, qui permet aux utilisateurs de ressentir des forces, des résistances et des textures virtuelles.

Le Virtuose offre une large gamme d'applications dans divers domaines, tels que la formation médicale, la simulation de vol, la conception industrielle, la recherche en robotique et bien d'autres. Il permet aux utilisateurs de manipuler des objets virtuels avec une précision et une sensation de réalisme accrues, ce qui peut améliorer l'efficacité et l'immersion dans les applications de réalité virtuelle.



FIGURE 2.1 – Vituose 6D

Utilisations du Virtuose

Le Virtuose trouve des applications dans divers domaines, notamment :

- **Formation médicale** : Le Virtuose peut être utilisé pour simuler des procédures médicales, permettant aux étudiants en médecine de s'entraîner dans un environnement virtuel réaliste avant de passer à la pratique sur de vrais patients.

- **Simulation de vol :** Le Virtuose peut être utilisé pour simuler les sensations de pilotage d'un avion ou d'un hélicoptère, offrant aux pilotes en formation une expérience plus réaliste et immersive.
- **Conception industrielle :** Le Virtuose peut être utilisé pour manipuler des modèles 3D dans des environnements de conception industrielle, permettant aux ingénieurs de prototyper et de tester des produits virtuellement avant leur fabrication.
- **Recherche en robotique :** Le Virtuose peut être utilisé pour simuler les sensations de manipulation d'objets par des robots, ce qui permet aux chercheurs en robotique de développer et de tester des algorithmes de manipulation dans des environnements virtuels.

2.2.2 *Le Scale1*

Il s'agit de l'interface haptique avec le plus grand espace de travail du monde. Grâce à un porteur motorisé dont les dimensions sont adaptables (1 à 4 degrés de liberté), il peut couvrir une grande surface. « Le Scale1 se destine principalement à des applications liées à la maquette numérique en industrie et à des sujets de recherche liés à la réalité virtuelle. »



FIGURE 2.2 – le Scale1

Le Scale1 est utilisé principalement dans deux applications. D'un côté, les entreprises et industriels qui souhaitent simuler avec le plus de réalisme possible leurs opérations d'assemblages, processus de maintenance, de vérification d'accessibilités, ou encore études ergonomiques, en amont des phases de production et en lien avec les phases de prototypage. De l'autre, les laboratoires et universités qui s'équipent pour disposer d'un matériel haptique à la pointe

leur permettant d'interagir encore plus en profondeur avec les environnements immersifs et avancer les connaissances acquises sur les comportements humains, les sciences cognitives et autres sujets d'études utilisant la réalité virtuelle comme vecteur.

3 Contexte du stage

HAPTION commercialise des produits en France et à l'étranger. Les clients souhaitent vérifier les performances des produits au moment de leur réception. De façon symétrique, HAPTION a besoin de s'assurer que les produits expédiés répondent bien aux performances annoncées.

3.1 Besoin de vérification en usine

Il s'agit de faire une vérification approfondie afin de caractériser complètement le produit.

3.1.1 Vérification des positions

Le besoin est de vérifier que les débattements sont bien les bons. Il s'agit de valider :

- Que les débattements sont bien mesurés par rapport à la réalité (comparaison à une mesure externe)
- Que les débattements sont ceux annoncés dans les spécifications
- Que la calibration géométrique du dispositif est maîtrisée

3.1.2 Vérification des efforts

Au niveau des efforts, le besoin de vérification est multiple

- Vérifier que les efforts mesurés sont conformes à la réalité (comparaison à une mesure externe)
- Identifier les hystérésis en effort (assimilable à du frottement sec) en fonction de la charge
- Identifier les défauts de compensation de gravité par les ressorts
- Vérifier que les efforts nominaux sont atteints sans que les moteurs n'aient besoin d'un courant supérieur aux courants nominaux.
- Vérifier que les efforts maximaux peuvent bien être atteints.

3.2 **Besoin de vérification chez le client**

Il s'agit de permettre au client de tester les éléments principaux pour accepter la livraison du produit.

3.2.1 **Vérification des débattements**

Le besoin est de vérifier que les débattements sont bien conformes aux spécifications. L'utilisation des capteurs internes aux produits semble suffisante

3.2.2 **Vérification des efforts**

Le besoin est de valider que les efforts sont conformes à la réalité par une expérience simple à mettre en œuvre. Cette expérience devra induire que les efforts générés par le dispositif haptique soit proche des efforts nominaux (dans une position définie).

4 Vérification des positions (Client)

4.1 **Test JointsLimits**

Ce test permet de vérifier si les débattements articulaires sont bien conformes aux spécifications du cahier de charge du Virtuose et permet de retourner les valeurs "**upper Joint limit**" et "**lower Joint limit**" pour chaque articulation, ces valeur sont renseignées après dans le fichier *Virtuose_num.param*.

Le moyen de test permet de déplacer doucement chaque articulation afin d'aller d'une butée à l'autre, d'enregistrer les valeurs limites dans un fichier *log_num.txt* et de faire un plot de l'espace de travail du virtuose correspondant aux valeurs trouvées par le biais du Modèle géométrique direct .

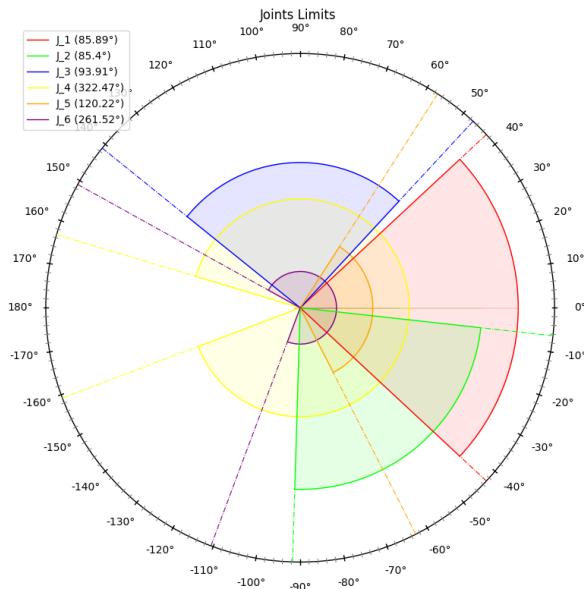


FIGURE 4.1 – Joints Limits

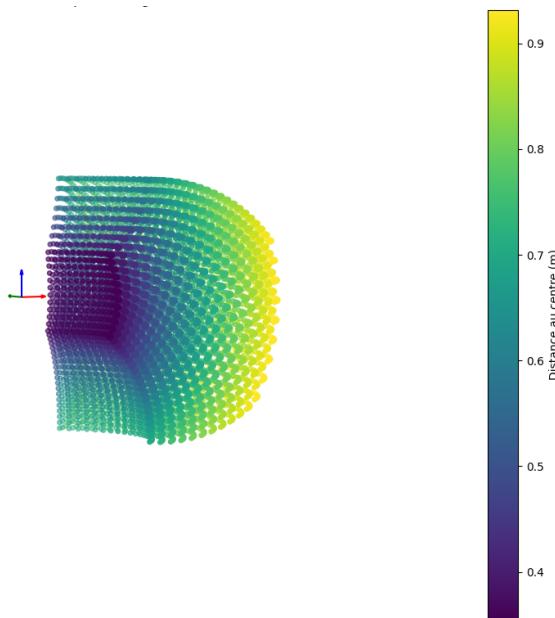


FIGURE 4.2 – Espace atteignable

Les limites articulaires trouvées représentent les butées logicielles : ce sont des limites prédéfinies pour que le virtuose n'aille pas sur les butées mécaniques et avoir tout le temps une sécurité contre la détérioration du système.

4.2 *Butées logicielles*

Afin de voir la différence entre les deux butées j'ai effectuer un test ou j'essaye d'enregistrer une trajectoire de l'effecteur en deux mode :

- **Transparent** : c'est un mode pour lequel on active le retour d'effort du virtuose avec des gains nulle en envoyant à chaque itération la position courante : dans ce cas les limites des articulations correspondent aux butées logicielles.
- **Sans retour d'effort** : c'est un mode où on n'appuie pas le bouton du retour d'effort et donc les moteurs sont pas alimentés : il y a pas de retour d'effort et les butées logicielles ne sont pas prises en compte, dans ce cas les limites articulaires représentent butées mécaniques.

Le test permet de retourner un plot des deux trajectoires enregistrées en animation virtuose :

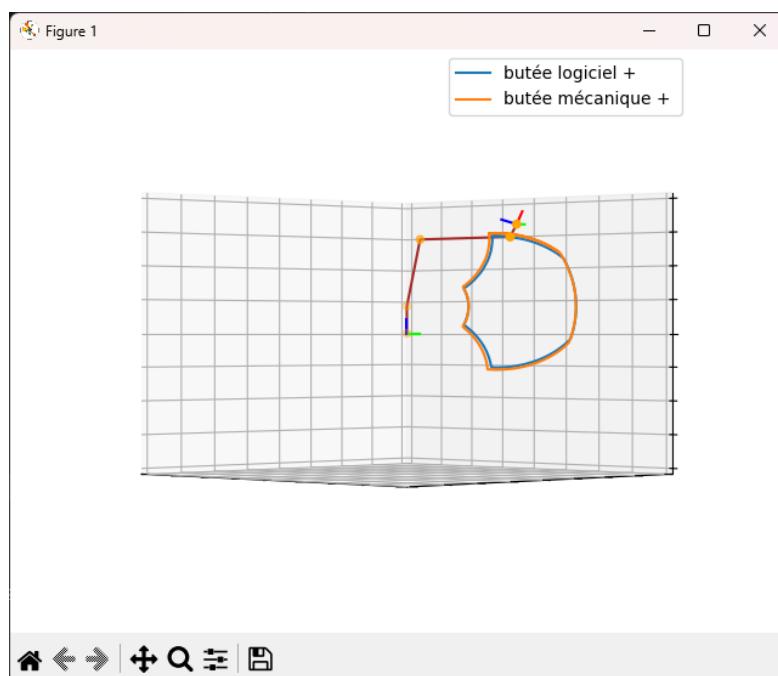


FIGURE 4.3 – butée : logicielle-mécanique

Ce test permet de voir le décalage entre butée logicielle et mécanique pour tous les articulation il faut que l'opérateur choisit la trajectoire adéquate pour rester toujours en butée des articulation, la trajectoire doit être simple à répéter afin de faire l'enregistrement pour les deux modes.

4.3 *Test Stylo_Calibration*

L'objectif de ce test est de déterminer la position de l'extrémité du stylo (effec- teur final) en relation avec le centre de rotation instantané des trois articulations finales.

Pour réaliser ce test, j'ai conçu une nouvelle poignée spéciale que j'utiliseraï dans mes expérimentations. Cette poignée de test sera également mise à disposition du client, permettant ainsi des vérifications chez lui en utilisant le dispositif virtuose.

4.3.1 *conception d'un support stylo*

La poignée est conçue pour fixer un stylo et initier l'enregistrement de chaque point en appuyant sur un bouton dédié à chaque point.

Ci-dessous, vous pouvez voir des illustrations de la poignée stylo :

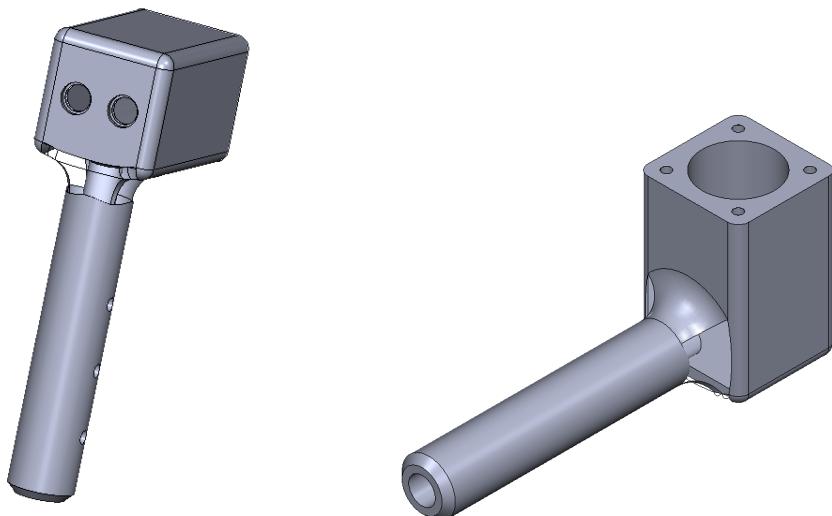


FIGURE 4.4 – Poignée stylo

Chaque poignée est capable de se connecter au dispositif virtuose par le biais d'un connecteur électronique, établissant ainsi une interface homme-machine à travers les boutons de la poignée.

J'ai inclus deux boutons poussoirs avec des LEDs intégrées dans la conception. Le premier bouton agit comme un bouton d'arrêt d'urgence ("homme mort"), tandis que le deuxième bouton est utilisé pour lancer et arrêter l'enregistrement des données.

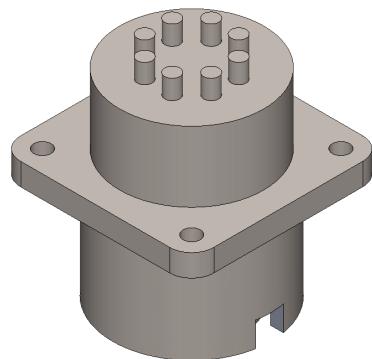
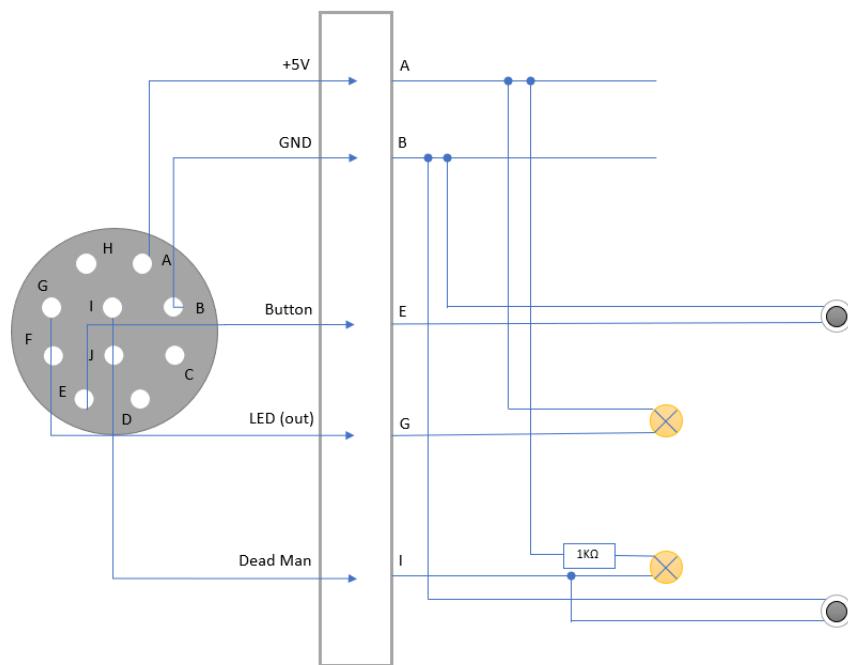


FIGURE 4.5 – Connecteur



FIGURE 4.6 – État finale poignée

**FIGURE 4.7 – Plan de câblage**

4.3.2 *Optimisation :*

L'objectif de cette optimisation est de déterminer les décalages (offsets) du stylo par rapport à l'effecteur spécifié dans le modèle géométrique préexistant dans le RaptorAPI. Cette optimisation vise à faciliter les tests en position cartésienne au niveau de la pointe du stylo par la suite.

Après avoir enregistré plusieurs configurations du robot en variant les positions des articulations de manière à ce que la pointe du stylo soit dirigée vers un même point sur la feuille, nous pourrons analyser les données obtenues.

Variable d'optimisation :

la variable d'optimisation sera le vecteur des coordonnées du end_effector dans le repère local du CRI .

$$U = [off_x, off_y, off_z]$$

Paramètres d'optimisation :

le modèle géométrique direct permet de donner la Position sous forme d'une variable **Displacement** constitué d'un vecteur coordonnés 3D et un quaternion pour définir la rotation de la Poignée :

$$Disp = [x, y, z, q_w, q_x, q_y, q_z]$$

le seul paramètre d'optimisation sera un fichier data qui contient les différents Displacements des nuages des points enregistrés.

Fonction objective :

la fonction objective pour ce problème d'optimisation passe par les étapes suivantes :

- lecture du paramètre d'optimisation *data.txt*
- conversion des displacements CRI aux coordonnées 3D end_effector qui correspondent à la variable d'optimisation *U*.
- Appliquer un clustering sur la nouvelle base de données pour avoir le centre de chaque nuage de points : cela pour éviter les effets de vibration poignée en appuyant le bouton d'enregistrement.
- la fonction retourne la dispersion par rapport au centre d'inertie des points trouvés c'est à dire la la norme des distances de chaque point au centre .

Minimiser la fonction objective permet de trouver le vecteur $U_{optimal}$ qui sera enregistré et utilisé après pour tous les test pour savoir la position du nouveau **end_effector**.

Validation :

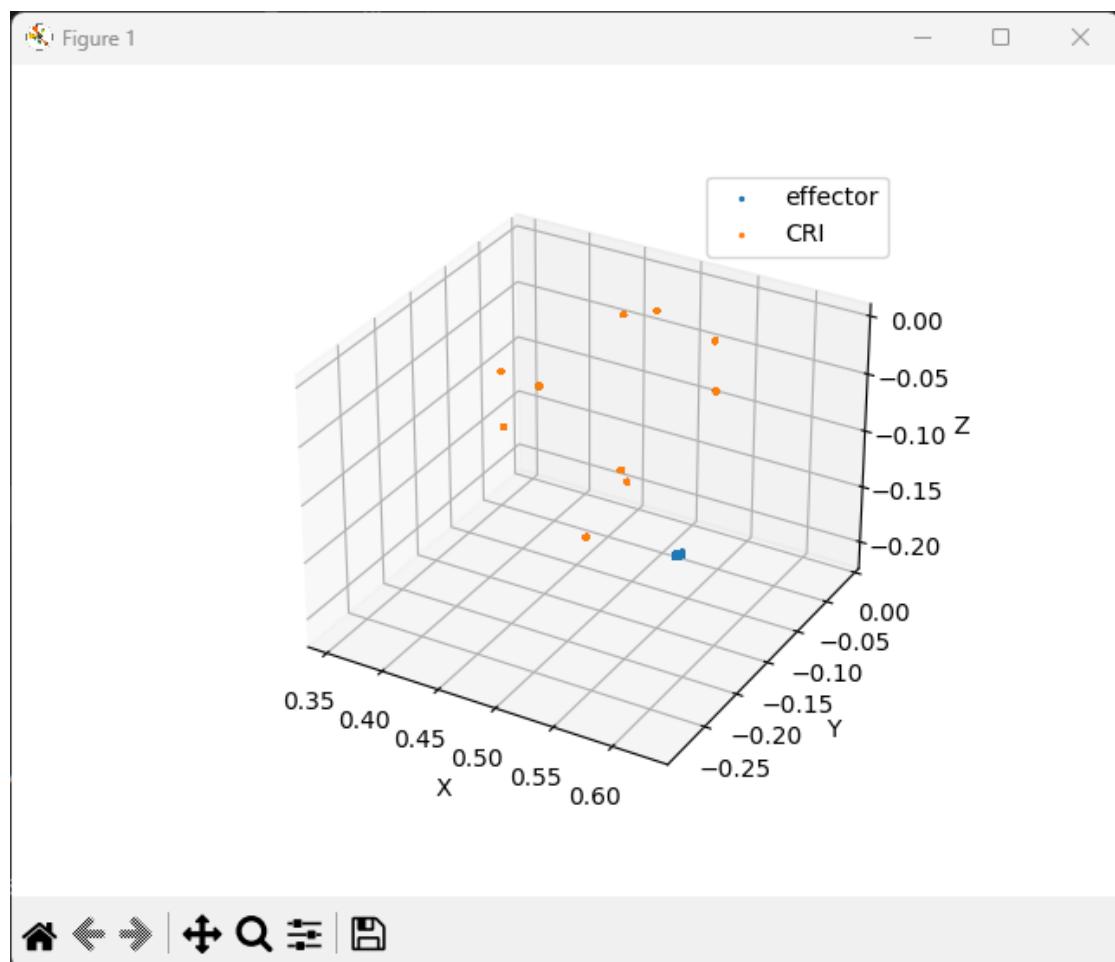


FIGURE 4.8 – Position end_effector par rapport aux CRI

L'optimisation a réussi à identifier les décalages adéquats, comme en témoigne le graphique qui démontre le centrage des points effecteurs sur un même emplacement. La figure ci-dessous confirme également la réussite de l'optimisation, car l'écart constaté est de l'ordre d'**1mm**

Cela indique que l'objectif de l'optimisation a été atteint avec succès, et que les ajustements apportés aux offsets ont conduit à l'alignement souhaité des points effecteurs, validant ainsi l'efficacité du processus d'optimisation.

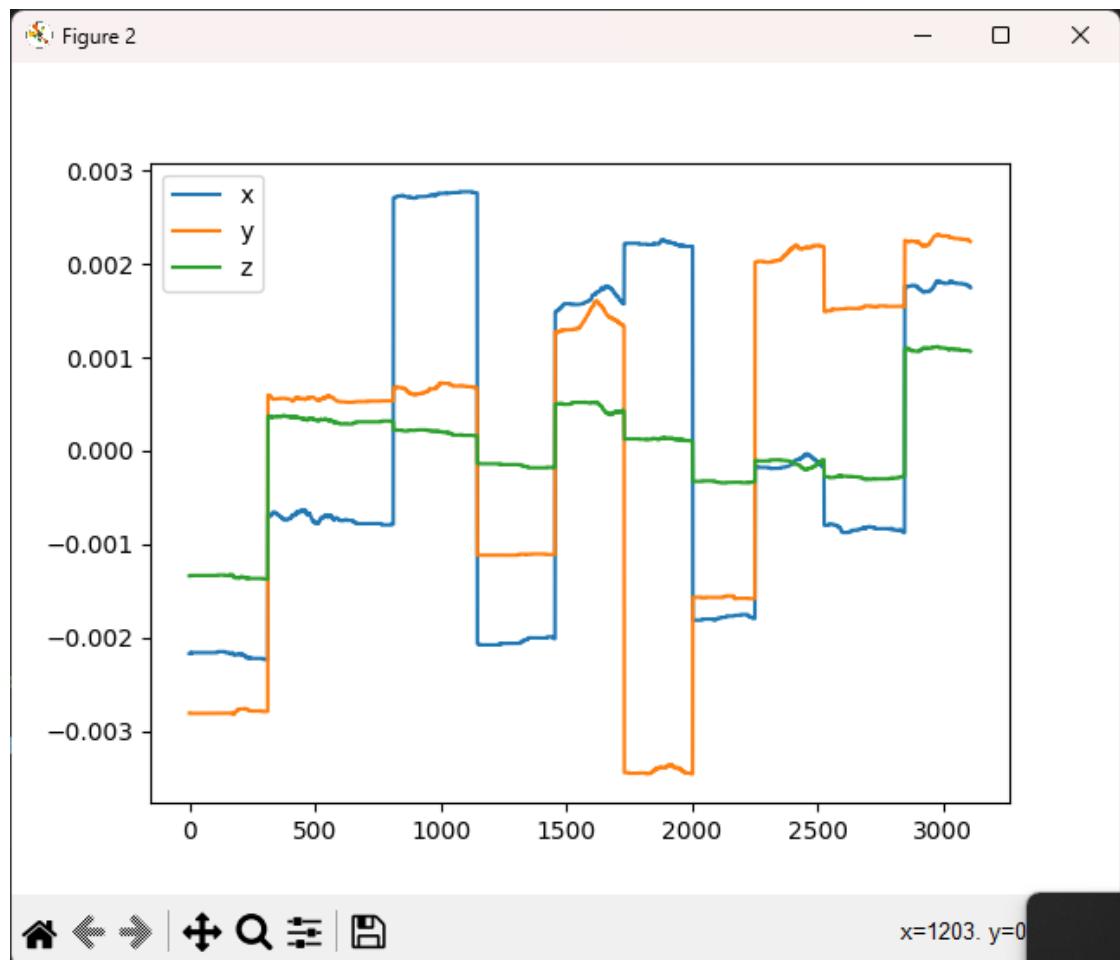


FIGURE 4.9 – déviation standard (m)

4.4 Test Points

Le but de ce test est de vérifier la précision des positions cartésiennes estimées par le système virtuose en les comparant à des mesures obtenues à l'aide de capteurs externes. Cette comparaison vise à évaluer l'alignement entre les capteurs internes du système et les capteurs externes. Un capteur externe, par exemple un instrument de mesure de distance, sera utilisé pour enregistrer plusieurs points sur une feuille. Ensuite, les distances entre ces points seront comparées pour évaluer la conformité des positions cartésiennes estimées.

Pour faciliter cette comparaison, nous effectuerons des mesures de distance, car celles-ci ne dépendent pas du repère choisi. Cette approche est logique étant donné que le repère lié au système virtuose diffère du repère de la feuille.

Le test mettra en œuvre un processus similaire à celui du regroupement (clustering) précédemment utilisé. Cependant, cette fois-ci, il enregistrera des points différents pour permettre la mesure des distances entre eux. Ensuite, le test déterminera les centres de chaque nuage de points et utilisera ces points centraux comme référence pour former un polygone constitué par ces points, en prenant en compte les distances entre les segments. Ce polygone sera ensuite présenté à l'opérateur pour qu'il effectue la comparaison avec les positions estimées par le système virtuose.

En résumé, ce test permettra de vérifier l'exactitude des positions cartésiennes estimées en les comparant à des mesures de distance obtenues à l'aide de capteurs externes. Le processus inclura l'enregistrement de points, le calcul de distances et la formation d'un polygone de référence pour la comparaison finale.

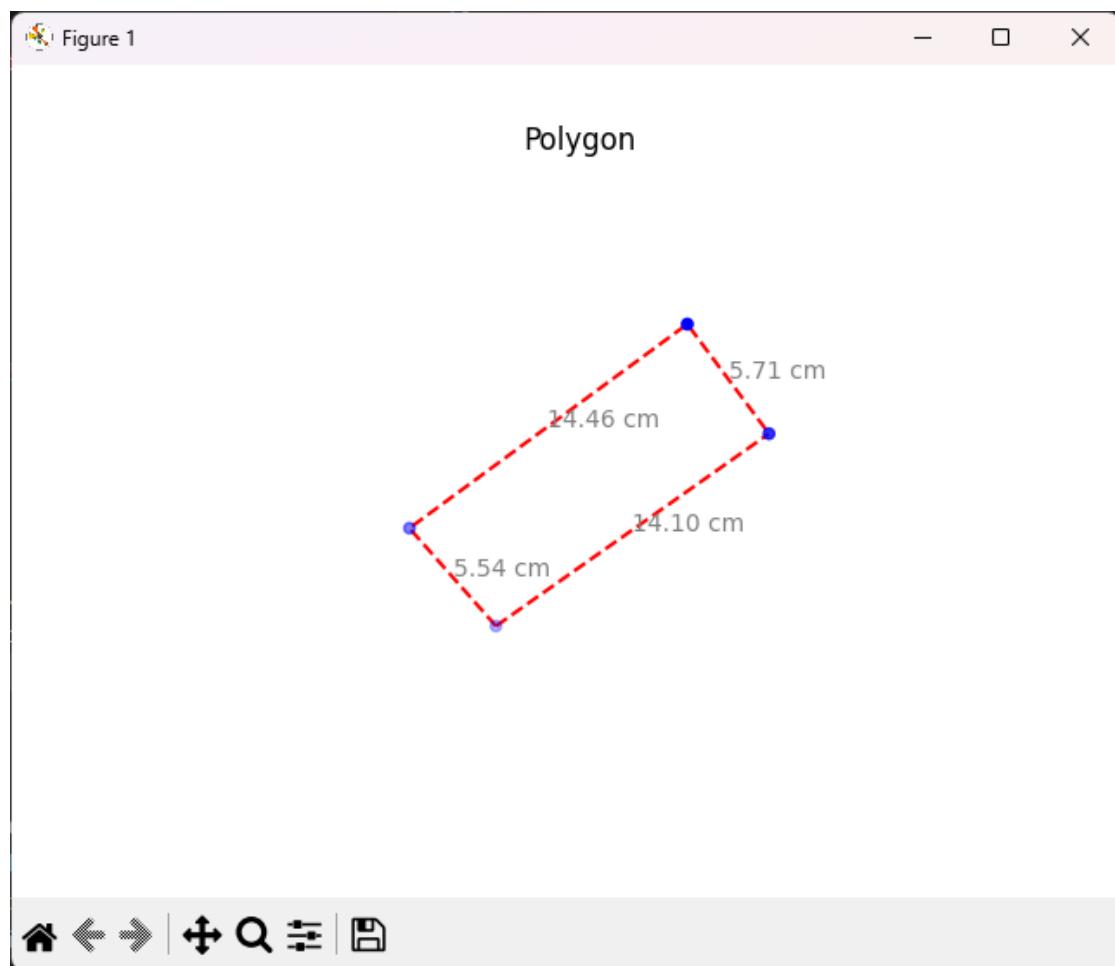


FIGURE 4.10 – Output Test

Pour construire le polygone convexe j'ai utilisé la méthode de Jarvis qui s'explique comme suit :

1. Trouver le point le plus à gauche de l'ensemble de points en cherchant celui avec la plus petite valeur x (en cas d'égalité, choisissez celui avec la plus petite valeur y).
2. Commencer avec ce point le plus à gauche et parcourir les autres points pour trouver le prochain point dans le polygone. Sélectionner le point ayant l'angle le plus petit et le plus à gauche par rapport au point actuel.
3. Répéter l'étape 2 jusqu'à revenir au point de départ, en ajoutant chaque nouveau point au polygone.
4. Former le polygone en utilisant les points trouvés dans l'ordre où ils ont été sélectionnés.

5 Vérification des positions (Usine)

Afin d'effectuer la vérification des positions en usine il nous a fallu de trouver un outil de localisation pour l'utiliser comme un capteur externe au virtuose.

5.1 *Localisation par caméras*

Les systèmes de suivi ART sont des systèmes de suivi optique infrarouge (IR). Nous considérons le "suivi" comme la mesure de la position des objets ou des sujets qui se déplacent dans un espace défini. Ces objets ou sujets à suivre doivent être équipés de marqueurs individuels ou d'agencements rigides de marqueurs (corps rigides ou cibles). La position et/ou l'orientation de ces corps rigides peuvent être mesurées. Si seule la position spatiale (X, Y, Z) est mesurée, on parle de "trois degrés de liberté" (3DL) de suivi. La mesure simultanée de la position et de l'orientation (trois coordonnées angulaires indépendantes) est appelée "six degrés de liberté" (6DL) de suivi. Des marqueurs individuels sont suffisants si seules les coordonnées 3DL sont nécessaires. Pour le suivi en 6DL, un corps rigide est obligatoire. Les marqueurs passifs sont recouverts d'un matériau rétro-réfléchissant - ils agissent comme des réflecteurs de lumière. Les émetteurs de lumière actifs (par exemple, basés sur des LED infrarouges) sont appelés marqueurs actifs.

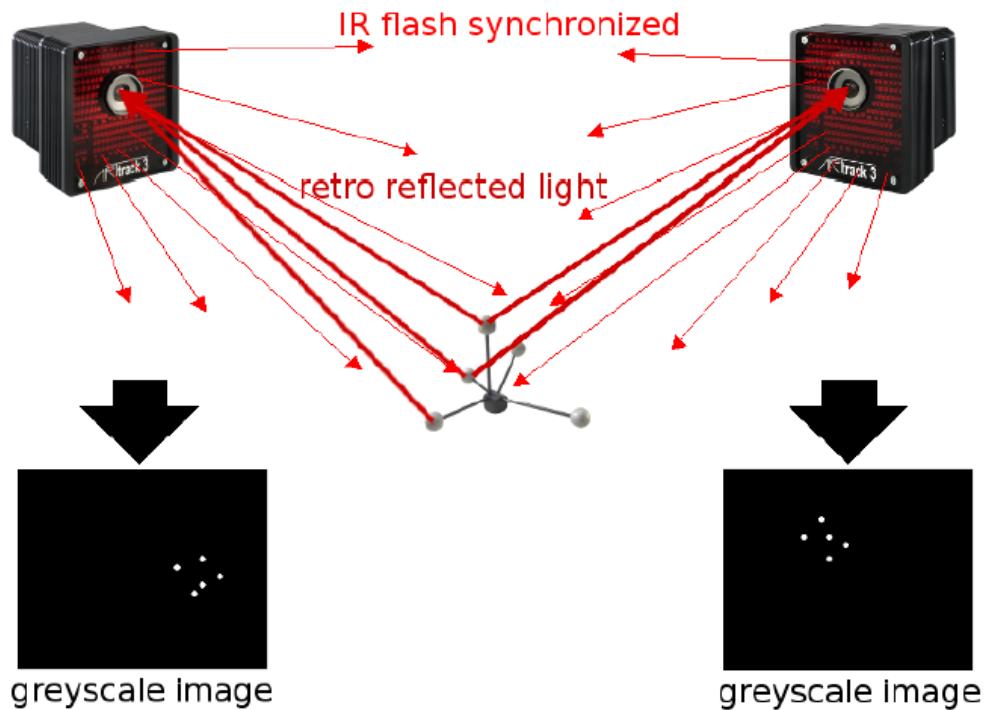


FIGURE 5.1 – Principe du tracking optique

Les caméras envoient des flashes infrarouges synchronisés qui sont réfléchis vers l'objectif par le matériau rétro-réfléchissant qui recouvre les marqueurs de la cible. Les caméras de suivi intelligentes, qui balayent un certain volume, détectent le rayonnement infrarouge réfléchi par les marqueurs et créent une image en niveaux de gris basée sur le rayonnement infrarouge reçu. Pendant le prétraitement, la caméra calcule avec une grande précision les positions en 2D des marqueurs en utilisant la reconnaissance de motifs. Une précision moyenne en 2D de 0,04 pixel (déviation maximale de 0,1 pixel en 2D) est standard dans les caméras de suivi ART. Ensuite, les données en 2D sont envoyées au contrôleur via Ethernet. Maintenant, le contrôleur doit calculer les données en 3DL ou 6DL. La base de ce calcul est que les champs de vision des caméras se chevauchent. DTrack2 calcule le chemin des rayons optiques des caméras vers les marqueurs et fournit les intersections des rayons en coordonnées tridimensionnelles. Ces intersections sont les positions des marqueurs. La position et l'orientation des caméras sont connues grâce à l'étalonnage de la pièce. Lors de l'étalonnage du corps, DTrack2 identifie certains agencements de marqueurs comme des corps rigides. Sur cette base, DTrack2 est capable de calculer les données en 6DL et

connaît enfin la position et l'orientation de la cible, et donc de l'objet ou du sujet à suivre. Dans les systèmes de suivi optique, il faut savoir que le suivi n'est possible que tant que la cible se trouve dans la plage de suivi des caméras et n'est pas occultée par d'autres objets ou par l'objet à suivre. Plus en détail, au moins quatre marqueurs d'une cible doivent être visibles pendant au moins deux caméras pour permettre le suivi.

5.1.1 Calibration de la pièce

Avant de pouvoir commencer le suivi, une calibration de la pièce doit être réalisée. Pendant cette calibration, le système identifie la position des caméras à l'intérieur du volume de mesure et détermine le système de coordonnées tridimensionnel. La calibration de la pièce est toujours la première étape à effectuer. Sans elle, la calibration du corps et le suivi ne seront pas possibles.

La calibration de la pièce doit être réalisée :

- Après la première configuration du système,
- Après tout changement de position des caméras infrarouges (re-calibration de la pièce),
- Après tout changement du nombre de caméras,
- Avant une calibration du corps (lorsque le système a été initialement configuré).

En principe, il est recommandé de toujours effectuer une calibration de la pièce après une certaine période d'utilisation du système, en particulier si la configuration du système ne exclut pas les mouvements des caméras au fil du temps (ces mouvements de caméra peuvent également être des dérives thermiques!).

Comment se fait la calibration ?

Déplacez doucement la baguette à l'intérieur du volume de mesure afin de générer un nuage de points virtuel. Ce nuage de points doit remplir au moins environ les deux tiers du volume de mesure. Il est utilisé pour le calcul des positions des caméras infrarouges, donc déplacer la baguette dans un volume très restreint réduira la précision de la calibration. Il faut trouver un compromis entre (1) des mouvements trop amples qui entraînent souvent l'échec de la calibration de la pièce, et (2) des mouvements trop limités qui conduisent à une

calibration de la pièce valide mais peu précise. Évitez les mouvements rapides et agités.

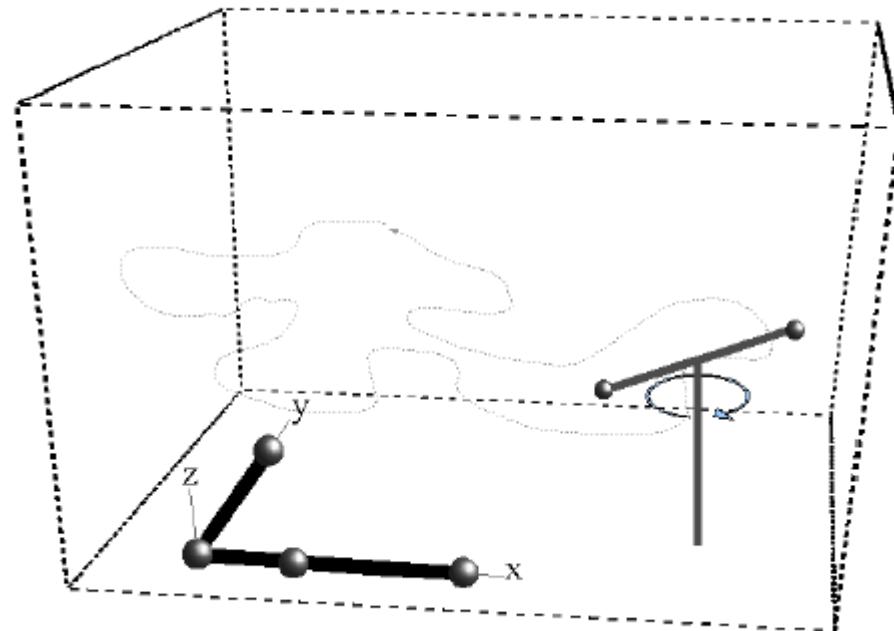


FIGURE 5.2 – Processus de pièce-calibration

Après une calibration réussie de la pièce, la fenêtre d'informations de DTrack2 affiche les résultats de la calibration. Cette fenêtre montre les résidus moyens pour chaque caméra (ici, le "résidu" correspond au résidu moyen des rayons pendant la détection des marqueurs), ainsi que l'écart moyen ("résidu de la baguette") et l'écart maximum ("plage de la baguette") de la longueur de la baguette pendant le processus de calibration. Ces valeurs dépendent de la géométrie du système et ne donnent des informations sur la qualité de la calibration qu'à un utilisateur expérimenté. La valeur "Images utilisées" représente le pourcentage de données valides (c'est-à-dire utilisées pour la calibration de la pièce) pour chaque caméra. Elle devrait être aussi élevée que possible pour toutes les caméras. Des valeurs inférieures à 50% indiquent une mauvaise qualité de calibration de la pièce. Le nombre d'images valides devrait être supérieur à 70% pour chaque caméra. La calibration de la pièce est confirmée (c'est-à-dire que les données sont enregistrées) en appuyant sur le bouton OK

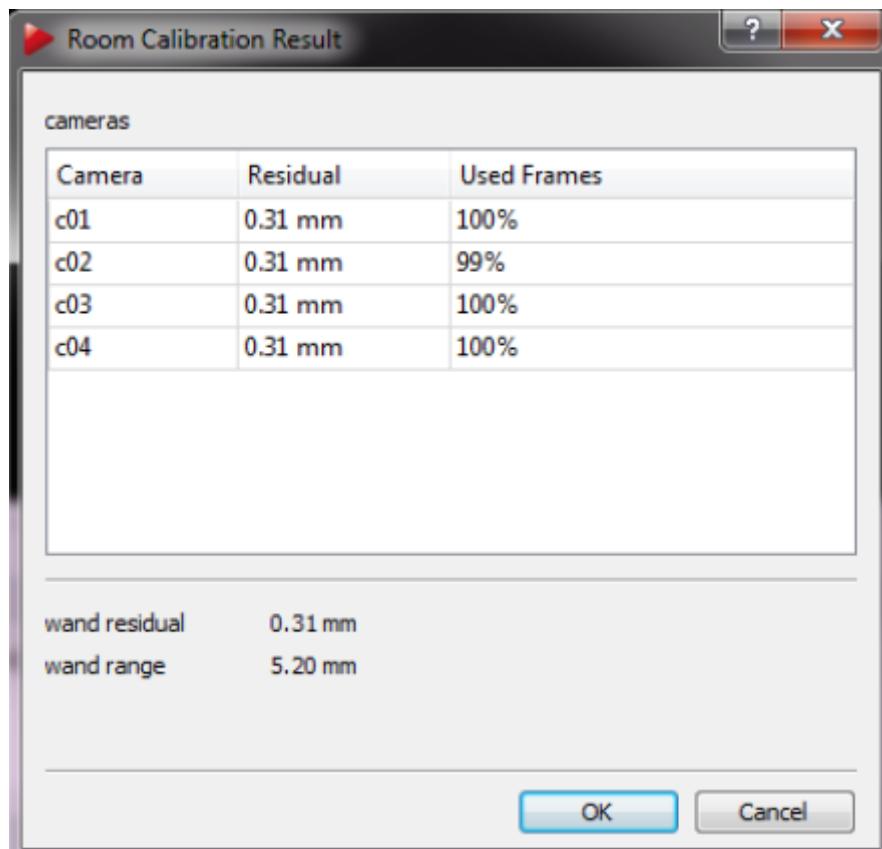


FIGURE 5.3 – Résultat de la pièce-calibration

5.1.2 *Calibration du corps*

Le processus d'enseignement de la géométrie d'une cible au système de suivi est appelé calibration du corps. Pour effectuer une calibration du corps, la cible (corps rigide) à calibrer doit être dans le champ de vision des caméras infrarouges.



FIGURE 5.4 – Cible virtuose

Tout d'abord, la cible à calibrer doit être sélectionnée dans la liste de sélection du corps. Par défaut, le "type" est défini sur "standard", ce qui permet de calibrer une cible en 6 DOF avec des marqueurs sphériques, plats ou actifs. Si vous utilisez une cible composée de marqueurs en forme d'anneau, vous devez sélectionner le "type" en fonction de la géométrie de la cible :

- "5DOF" : une cible composée de marqueurs sphériques alignés le long d'un axe principal. La rotation autour de cet axe ne peut pas être détectée par le système, vous perdez donc une dimension de liberté. C'est pourquoi on l'appelle une cible en 5DL.
- "5DOF (cylindER)" : une cible composée de marqueurs en forme d'anneau alignés le long d'un axe principal.
- "2x 5DOF (cylindER)" : une cible en forme de L composée de marqueurs en forme d'anneau qui fournit des données en 6DL. Sélectionnez le type approprié en fonction de la géométrie de votre cible afin d'effectuer la calibration du corps correctement.

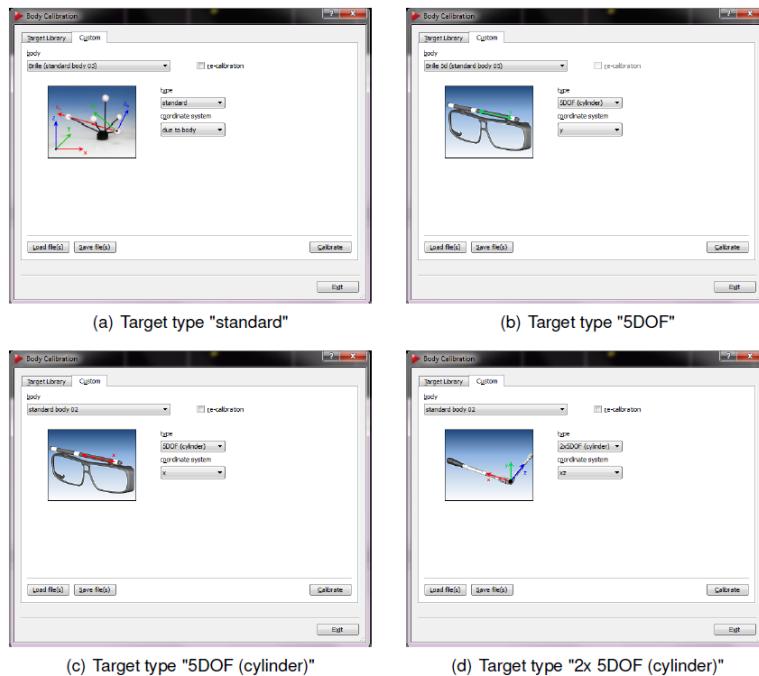


FIGURE 5.5 – Géométrie de la cible

Pour la cible virtuose j'ai utilisé la géométrie standard, donc j'ai du choisir entre différent type de calibration : '[due to body](#)', '[due to room](#)' or '[due to room \(zero in marker\)](#)'.

La différence entre ces types de calibration réside dans l'orientation du système de coordonnées du corps par rapport au corps lui-même. Pendant la calibration du corps, DTrack2 définit un système de coordonnées local (système de coordonnées du corps) pour chaque cible.

Body calibration setting due to body détermine le système de coordonnées du corps en se basant sur un ensemble de règles :

1. Recherchez la plus grande distance entre deux marqueurs du corps rigide. Ces deux marqueurs (#1 et #2) définiront l'axe X.
2. Recherchez un troisième marqueur (#3) qui a la plus petite distance par rapport à l'un des deux marqueurs #1 et #2. Le marqueur qui a la plus petite distance par rapport au marqueur #3 devient le marqueur #1. Il définit l'origine des coordonnées. L'autre marqueur devient le marqueur #2. L'axe X positif est dirigé du marqueur #1 vers le marqueur #2.

3. Le marqueur #3 définit le plan X/Y.
4. L'axe Z est déjà défini par ces règles, ce qui donne un système de coordonnées à main droite.

Body calibration setting due to room L'origine du système de coordonnées du corps est définie comme le centre (centre de gravité) de tous les marqueurs constituant le corps rigide. Les axes du système de coordonnées du corps sont parallèles aux axes du système de coordonnées de la pièce au début de la calibration du corps. Autrement dit, le résultat d'une calibration du corps dépendra de la position angulaire de la cible pendant la calibration. Une mesure en 6DL, après une calibration sans avoir déplacé le corps, donnera les coordonnées angulaires $0^\circ / 0^\circ / 0^\circ$. Si la cible a été déplacée pendant la calibration, la position angulaire de la cible au début de la calibration sera prise en compte.

Body calibration setting due to room (zero in marker) La méthode utilisée est une combinaison des deux premières méthodes. La direction des axes du système de coordonnées du corps est réglée parallèlement au système de coordonnées de la pièce au moment de la calibration du corps, comme cela est fait pour la configuration due à la pièce. L'origine du système de coordonnées du corps est définie au centre de gravité de tous les marqueurs constituant le corps rigide.

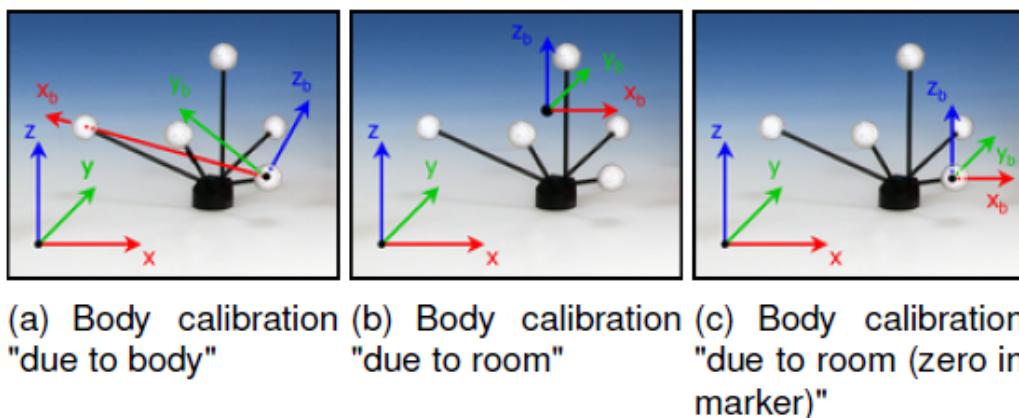


FIGURE 5.6 – Définition du système de coordonnées corps

5.1.3 L'ajustement du corps

La fonction d'ajustement du corps est un outil permettant de manipuler les données de calibration des corps rigides, c'est-à-dire que le système de

coordonnées du corps peut être modifié par rapport aux marqueurs du corps. Cet outil intègre directement dans DTrack2 la fonctionnalité de l'outil séparé précédemment appelé "ABCMan".

Les principales fonctionnalités de l'ajustement du corps sont les suivantes :

- Ajuster le système de coordonnées du corps de chaque cible à l'intérieur de l'interface utilisateur de DTrack2.
- Mode en ligne : la représentation de la cible dans la vue graphique peut être déplacée en déplaçant la cible réelle.
- Ajuster les axes du système de coordonnées du corps aux axes du système de coordonnées de la pièce sans modifier l'origine.
- Utiliser des systèmes de coordonnées prédéfinis pour le corps.
- Déplacer l'origine du système de coordonnées du corps.
- Définir l'origine du système de coordonnées du corps sur un marqueur sélectionné.

Ces fonctionnalités permettent de manipuler et d'ajuster les données de calibration du corps, offrant ainsi une plus grande flexibilité dans le suivi et l'alignement des systèmes de coordonnées.

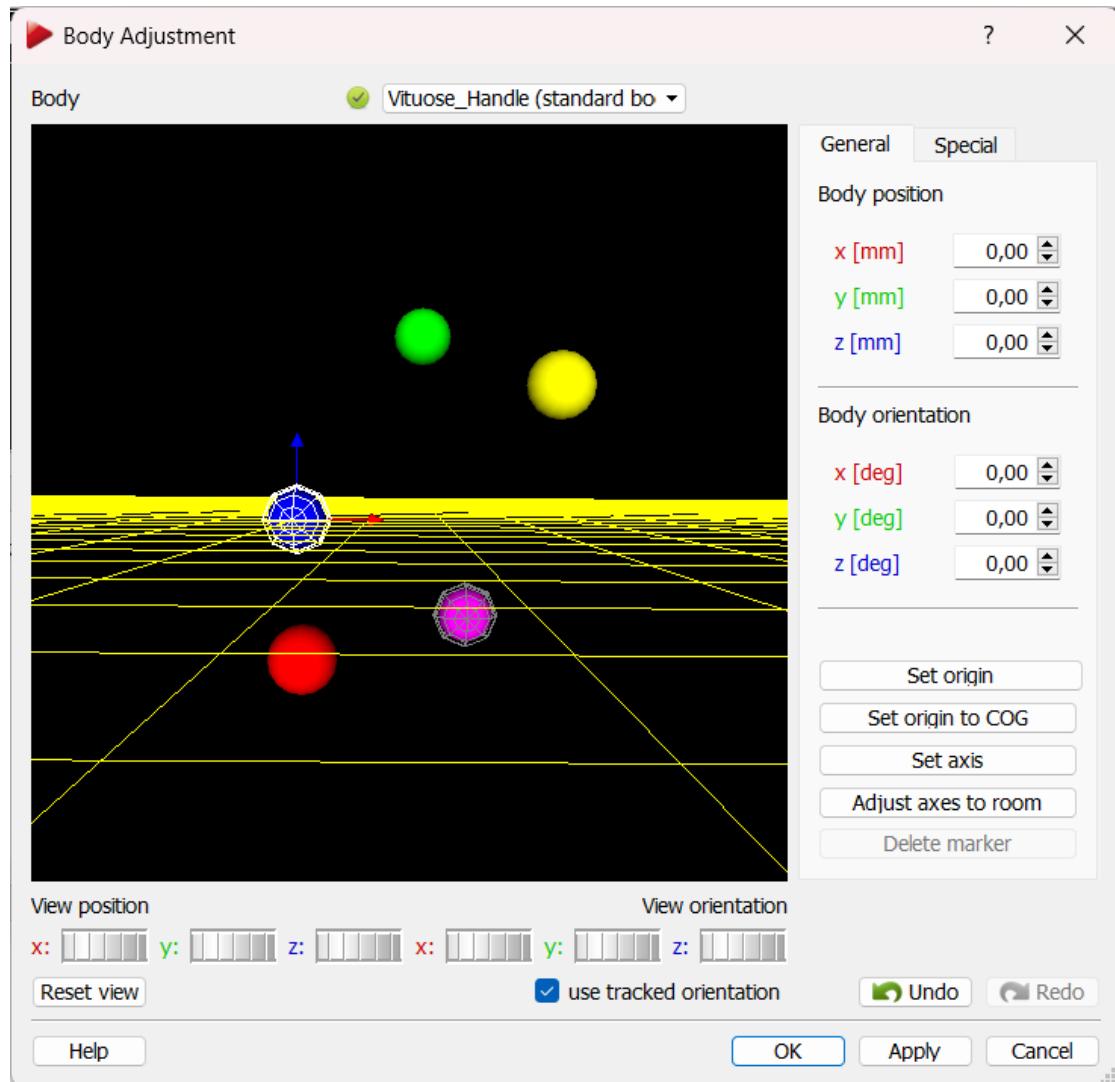


FIGURE 5.7 – Ajustement du corps

5.1.4 La sortie des données de mesure via Ethernet

DTrack2 utilise Ethernet (datagrammes UDP/IP) pour envoyer les données de mesure à d'autres applications. L'adresse IP et le port de l'application (et de l'ordinateur sur lequel elle s'exécute) peuvent être configurés dans les paramètres de sortie.

Chaque datagramme contient tous les résultats d'une seule mesure, codés au format ASCII. Un datagramme est envoyé après chaque mesure des caméras, c'est-à-dire à la fréquence de synchronisation.

Le "diviseur d'envoi de données" dans les paramètres de sortie offre la possibilité de diminuer la fréquence de sortie des données ($f_{sortie} = f_{synchronisation} / d_{diviseur}$). Cela permet de réduire la quantité de données envoyées, ce qui peut être utile pour des applications qui n'ont pas besoin de données à une fréquence élevée ou qui ont des limitations de bande passante.

Toutes les données sont exprimées en millimètres (mm) pour les mesures de position et en degrés (deg) pour les mesures angulaires.

ART fournit un code source d'échantillon gratuit appelé "DTrack2SDK" en C++ pour recevoir et analyser les données de sortie. Veuillez contacter ART pour l'obtenir.

Un datagramme UDP au format ASCII contient plusieurs lignes séparées par CR/LF (hex 0D 0A). Chaque ligne contient des données d'un type spécifique et commence par un identifiant. Dans les paramètres de sortie, vous pouvez configurer quelles lignes ou types de données doivent être inclus dans la sortie :

- Position 3D : Cette ligne contient les données de position 3D des marqueurs ou des corps rigides, exprimées en millimètres.
- Orientation : Cette ligne contient les données d'orientation des corps rigides, exprimées en degrés.
- Degrés de liberté (DOF) : Cette ligne indique si les mesures sont en 3DL (trois degrés de liberté) ou en 6DL (six degrés de liberté).
- Données brutes : Cette ligne contient les données brutes non traitées des capteurs, telles que les intensités infrarouges ou les coordonnées de pixel.

Vous pouvez sélectionner les lignes ou les types de données à inclure dans la sortie en fonction de vos besoins spécifiques en utilisant les options de configuration dans les paramètres de sortie de DTrack2.

Identifier	Type of data	enable/disable in
fr	frame counter	always enabled
ts	timestamp	<i>Settings → Output ('ts')</i>
6dcal	additional informations	<i>Settings → Output ('6dcal')</i>
6d/6di	standard bodies (6DOF)	<i>Settings → Output ('6d / 6di')</i>
3d	additional markers (3DOF)	<i>Settings → Output ('3d')</i>
6df/6df2	Flysticks (6DOF + buttons)	<i>Settings → Output ('6df / 6df2')</i>
6dmt	Measurement Tools (6DOF + tip trafo)	<i>Settings → Output ('6dmt')</i>
6dmtr	Measurement Tool references (6DOF)	<i>Settings → Output ('6dmtr')</i>
6dmt2	Measurement Tool with sphere tip (6DOF + tip trafo + sphere radius)	<i>Settings → Output ('6dmt2')</i>
gl/glcal	Fingertracking hands (6DOF + fingers)	<i>Settings → Output ('gl / glcal')</i>

FIGURE 5.8 – DTrack2SDK outputs

L'identifiant "6d" fait référence aux données de mesure des corps standard en 6DL (six degrés de liberté) qui sont suivis par le système de suivi. Les corps en 6DL comprennent tous les corps en 6DL, à l'exception des Flysticks et des Outils de mesure. Les corps qui ne sont pas suivis par le système à ce moment-là n'apparaissent pas dans la sortie.

Les données de chaque corps suivi sont regroupées en blocs entre crochets [] et contiennent les informations suivantes :

$$[id\ qu][s_x\ s_y\ s_z\ \theta\ \phi\ \psi][b_0\ b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8]$$

1. Numéro d'identification (id, à partir de 0) et valeur de qualité (qu, non utilisée).
2. Position (s_x, s_y, s_z) et angles d'orientation (θ, ϕ, ψ).
3. Matrice de rotation (bi) de l'orientation du corps.

La matrice de rotation R est donnée par :

$$R = \begin{bmatrix} b_0 & b_3 & b_6 \\ b_1 & b_4 & b_7 \\ b_2 & b_5 & b_8 \end{bmatrix}$$

5.2 Comparaison Caméras-Virtuose

Afin de faire la comparaison en temps réelle c'est à dire avoir une trame du virtuose en même temps qu'une trame des caméras j'ai du inclure le module DTrack2SDK dans le CMakeLists de mon Projet TestBench, et j'appelle la fonction

GetCartesianPose lorsque le Dtrack a bien envoyé une trame de valeur, puis j'enregistre La trajectoire Caméras et la trajectoire Virtuose dans des fichiers afin de traiter ces données après.

Calibration Caméras-Virtuose :

Après avoir enregistré la même trajectoire de deux points de vue différent, et pour faire une comparaison il faut que ces deux soient exprimées en même point et dans la même base et donc il faut une calibration caméras-virtuose. La calibration doit se faire afin de :

- Déterminer la position du marqueur par rapport au point CRI : le point référence du virtuose.
- Déterminer la matrice de transformation entre la base caméras et la base virtuose.

5.2.1 Détermination de l'offset marqueur

Variable d'optimisation :

la variable d'optimisation sera les coordonnées du marqueur dans le repère local du CRI .

$$U = [off_x, off_y, off_z]$$

Fonction objective :

la fonction objective pour ce problème d'optimisation passe par les étapes suivantes :

- lecture des paramètre d'optimisation *data_caméras.txt* et *data_virtuose.txt*
- conversion des déplacements CRI aux coordonnées 3D marqueur qui correspondent à la variable d'optimisation *U*.
- calcul des distances de chaque point de chaque base de données à tous les autres points de la même base de données
- la fonction retourne la norme de la différence entre toutes les distances calculées

Minimiser la fonction objective permet de trouver le vecteur $U_{optimal}$ qui permet de coïncider les distances calculées pour les deux bases de données j'ai

choisit l'optimisation sur les distance puisque j'ai pas les même repères pour les deux datas et la distance n'en dépendait pas.

Validation :

Afin de valider l'optimisation j'essaie de tracer la distance parcourue par rapport au point initiale d'enregistrement.

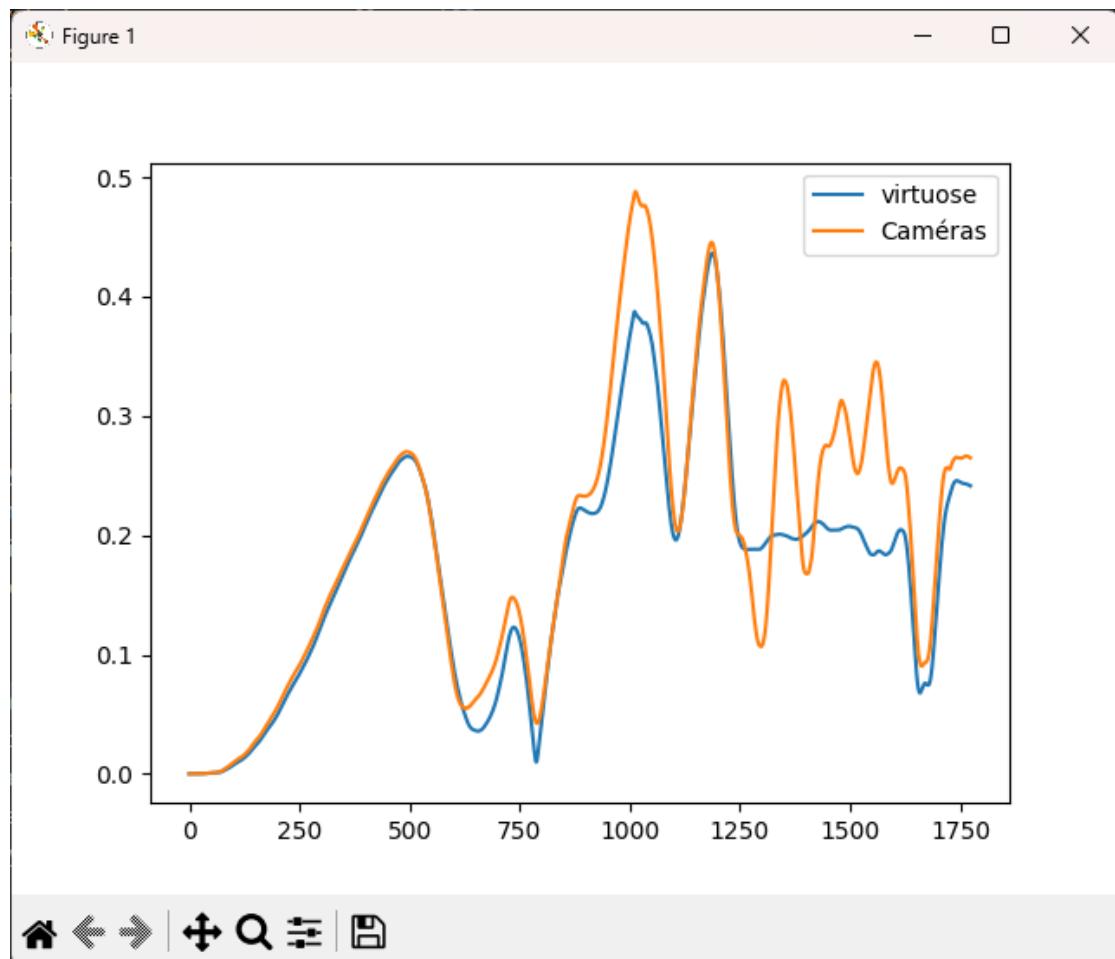


FIGURE 5.9 – Avant calibration

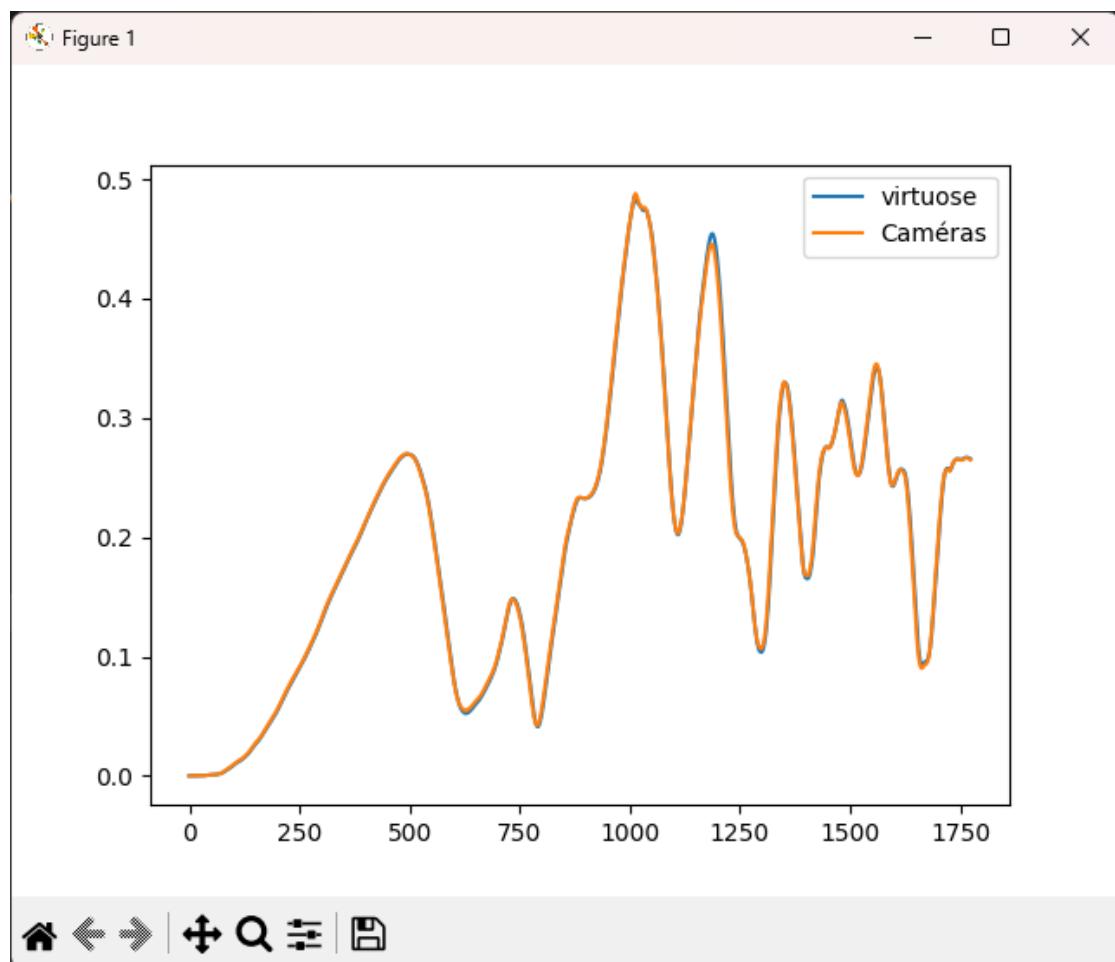


FIGURE 5.10 – Après calibration

On enregistre la valeur trouvée par l'optimisation et on va l'utiliser jusqu'à la fin de l'étude comparative.

5.2.2 *Détermination de la matrice de transformation*

Variable d'optimisation :

la variable d'optimisation sera un vecteur qui regroupe les paramètres nécessaires à la définition de la transformation qui ramène la trajectoire caméras dans la base virtuose.

$$U = [t_x, t_y, t_z, r_x, r_y, r_z]$$

Fonction objective :

la fonction objective pour ce problème d'optimisation passe par les étapes suivantes :

- lecture des paramètre d'optimisation *data_cameras.txt* et *data_virtuose.txt*
- conversion des déplacements CRI aux coordonnées 3D marqueur qui correspondent à la variable *marqueur_offset* déjà trouvée *U*.
- transformation de la trajectoire caméras par la variable d'optimisation *U*
- la fonction retourne la norme de la différence entre les deux nouvelles trajectoires

Minimiser la fonction objective permet de trouver le vecteur $U_{optimal}$ qui permet de coïncider les deux courbes et donc trouver la matrice de transformation correspondante.

Validation :

Vecteur de transformation :

```
[ 0.89818771 -0.30949991 -1.20420788  0.01535296  0.00696676 -1.8799201 ]
```

Matrice de transformation :

```
[[ -0.30421667  0.9524557 -0.01674386  0.89818771]
 [-0.95257739 -0.30429009 -0.00196515 -0.30949991]
 [-0.0069667   0.01535199  0.99985788 -1.20420788]
 [ 0.          0.          0.          1.        ]]
```

Offset marqueur :

```
[0.02793294 0.0604684 0.15933805]
```

On remarque que le vecteur de transformation présente une légère rotation par rapport à x et y, ce qui est normal puisque le virtuose est posé sur une table qui a une inclinaison par rapport au sol où j'ai effectué la calibration de la pièce.

Afin de valider l'optimisation j'essaie de tracer les deux trajectoires après transformation et voir si elles coïncident ou pas.

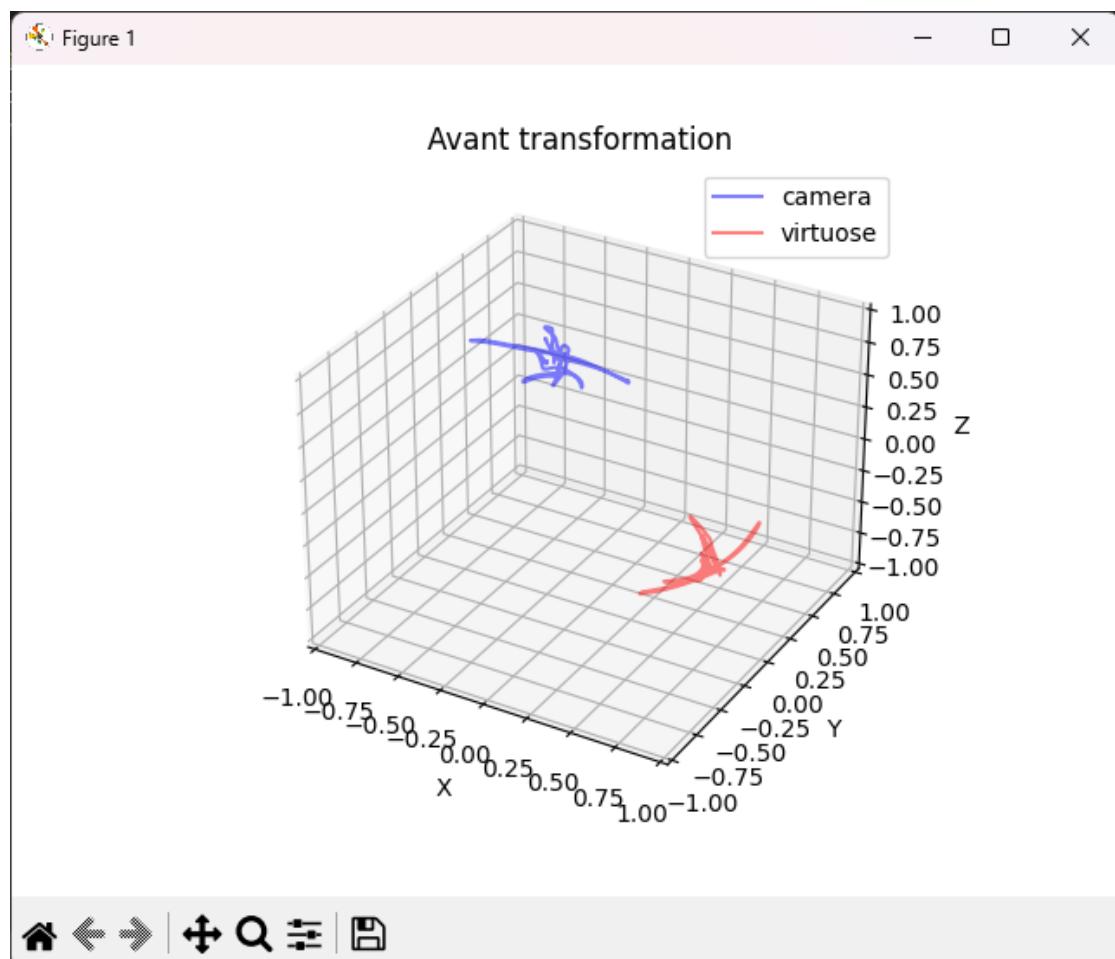


FIGURE 5.11 – Avant calibration

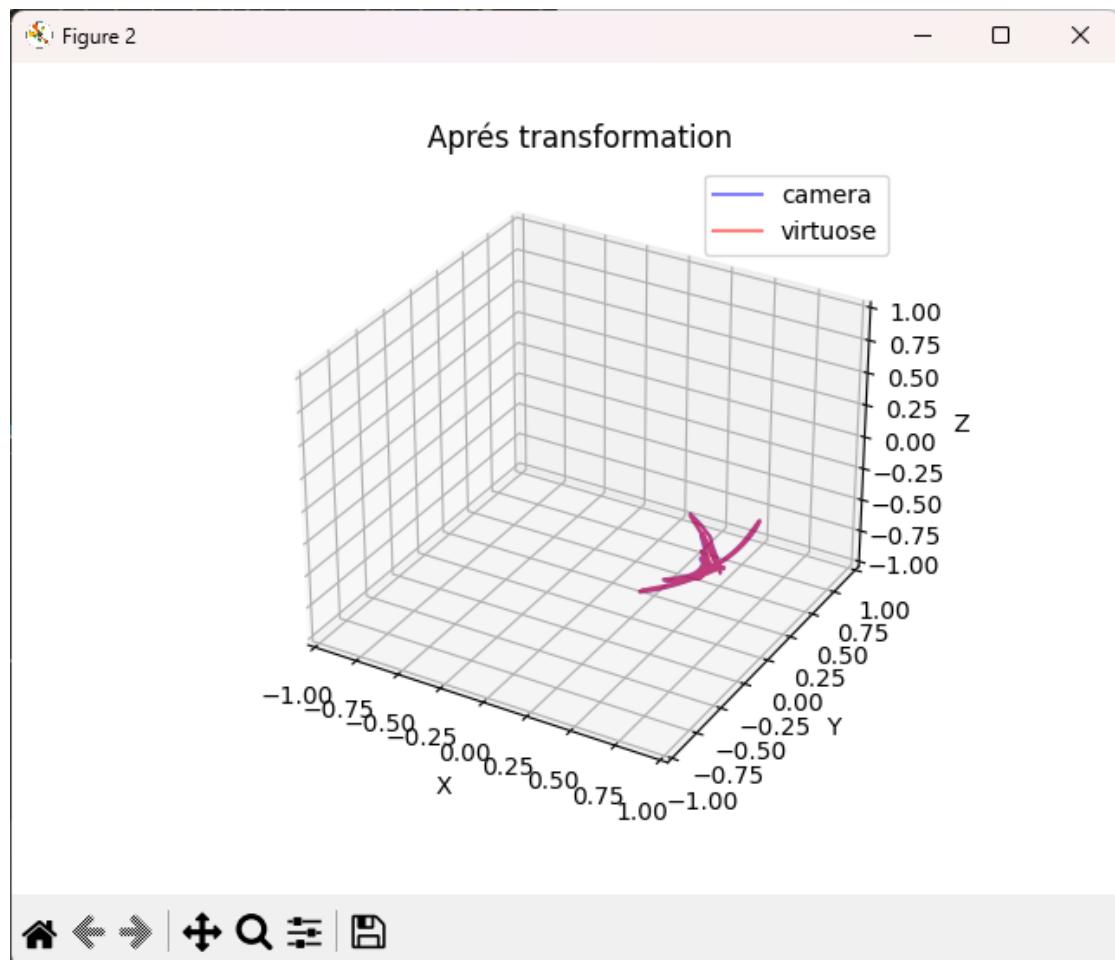


FIGURE 5.12 – Après calibration

5.2.3 Résultat de la comparaison

Après avoir calibrer les caméras avec la position virtuose on peut enregistrer d'autre trajectoires et essayer d'évaluer la précision en position du virtuose : le but de tous ce qui était fait avant.

Pour cela je calcule l'erreur en position entre les deux trajectoires et je calcule le déviation standard de l'erreur.

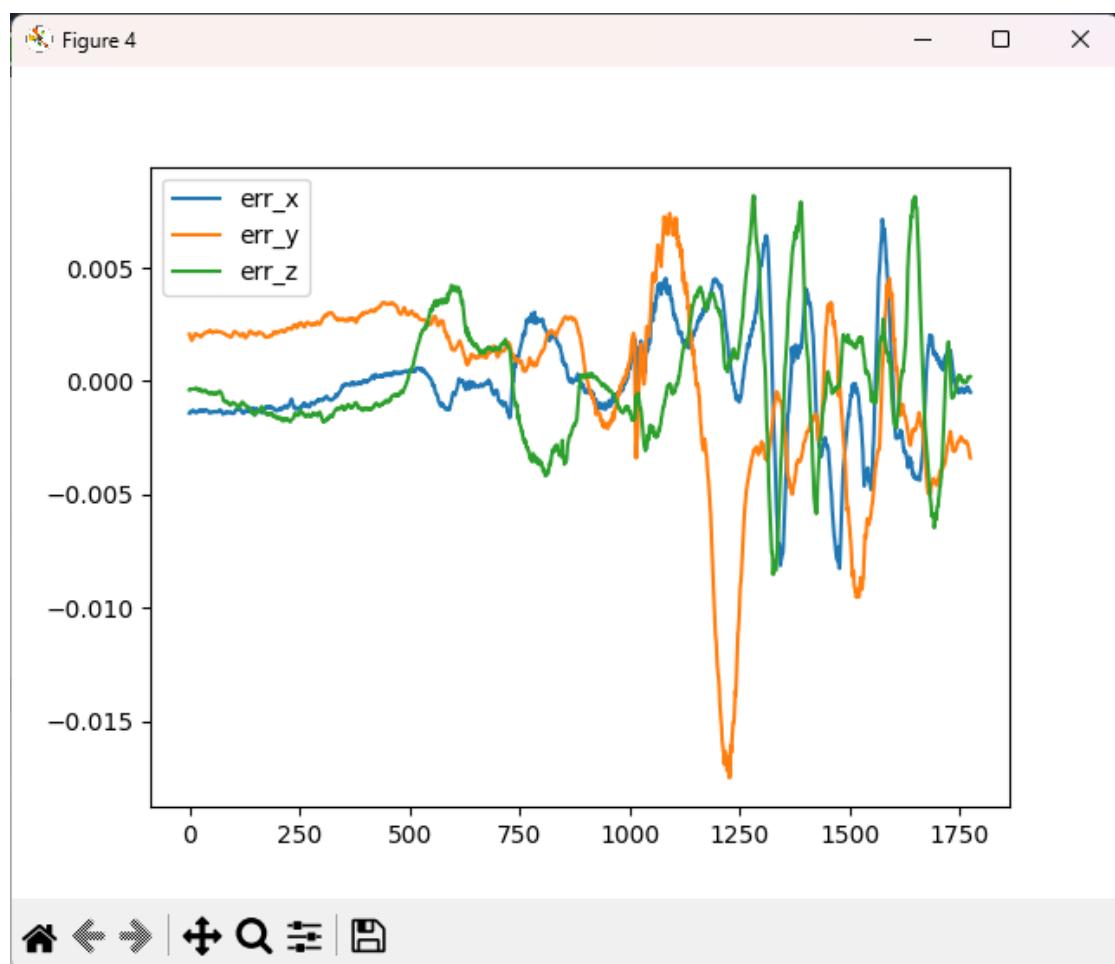


FIGURE 5.13 – erreurs en position (m)

Déviation standard des erreurs :

```
std_x  =  0.002312072341372838 (m)
std_y  =  0.00406994566918269 (m)
std_z  =  0.0026478624234321777 (m)
```

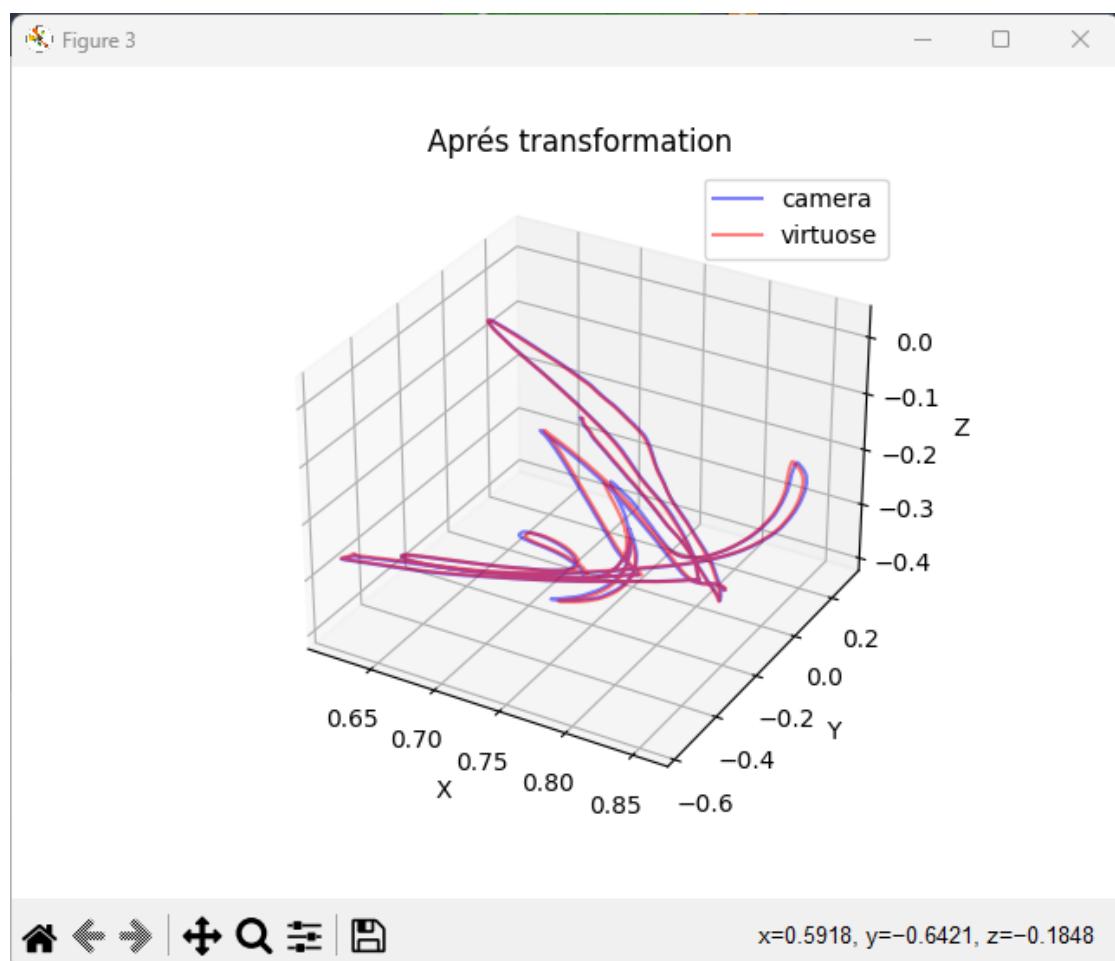


FIGURE 5.14 – Caméras vs Virtuose

6 Vérification des effort

6.1 Vérification chez le client

Afin de valider la conformité de la mesure des efforts par le Virtuose à la réalité pour le client on a prévu de faire un test **TestBlockedMode** qui maintient le dispositif dans une positions fixe et permet d'affiche à l'utilisateur les valeurs des effort mesurées, et en accrochant des poids au point TCP l'utilisateur peut comparer la mesure avec la masse réelles.

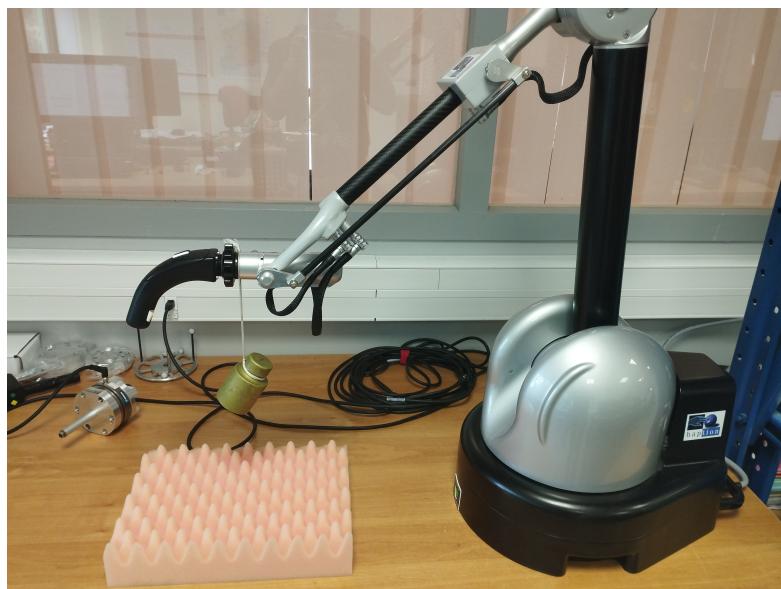


FIGURE 6.1 – TestBlockedMode

6.2 Vérification en usine

6.2.1 Test ForceSensor

Dans le cadre de la Vérification Usine, j'ai entrepris de vérifier la précision en effort en utilisant un capteur d'effort six axes . Ce capteur a été fixé au bout de la poignée d'un robot Virtuose, ce qui nous a permis de mesurer les forces et les moments appliqués en temps réelle .

La première étape consistait à concevoir un connecteur sur mesure afin de fixer le capteur d'effort de manière fiable et sécurisée à la poignée du Virtuose. Cette conception était cruciale pour garantir que le capteur était correctement

positionné et aligné avec le robot, afin d'obtenir des mesures précises et représentatives des forces exercées.



FIGURE 6.2 – Poignée Capteur

Pour effectuer les mesures en temps réel, j'ai développé un test ForceSensor qui utilisait un socket pour établir une communication bidirectionnelle avec le capteur d'effort. Grâce à cette communication, nous avons pu obtenir les valeurs du capteur de manière continue pendant que la Raptor API est connecté au Virtuose.



FIGURE 6.3 – Test ForceSensor

Pour obtenir les valeurs du capteur dans le repère du Virtuose, nous avons effectué une transformation des coordonnées. Pour cela, nous avons utilisé la rotation de la poignée du Virtuose, exprimée en quaternions, afin de convertir les valeurs du capteur de sa base native à celle du robot Virtuose. Cette opération nous a permis d'exprimer les forces et les moments mesurés par le capteur dans la même référence que celle du robot, facilitant ainsi la comparaison et l'analyse des données.

Grâce à cette méthodologie, nous avons pu réaliser des tests de vérification de la précision en effort du robot Virtuose dans des conditions réelles. Les données recueillies ont été analysées en détail pour évaluer la conformité du robot par rapport aux spécifications de précision en effort requises. Cette approche a été essentielle pour s'assurer du bon fonctionnement et de la sécurité du système de manipulation robotique. Les résultats obtenus ont été documentés et contribueront à améliorer les performances du robot Virtuose dans des applications de manipulation exigeantes.

6.2.2 Résultats du test

Pour le test j'ai essayé d'appliquer des efforts dans toutes les directions afin d'avoir une idée générale sur la précision effort .

Forces :

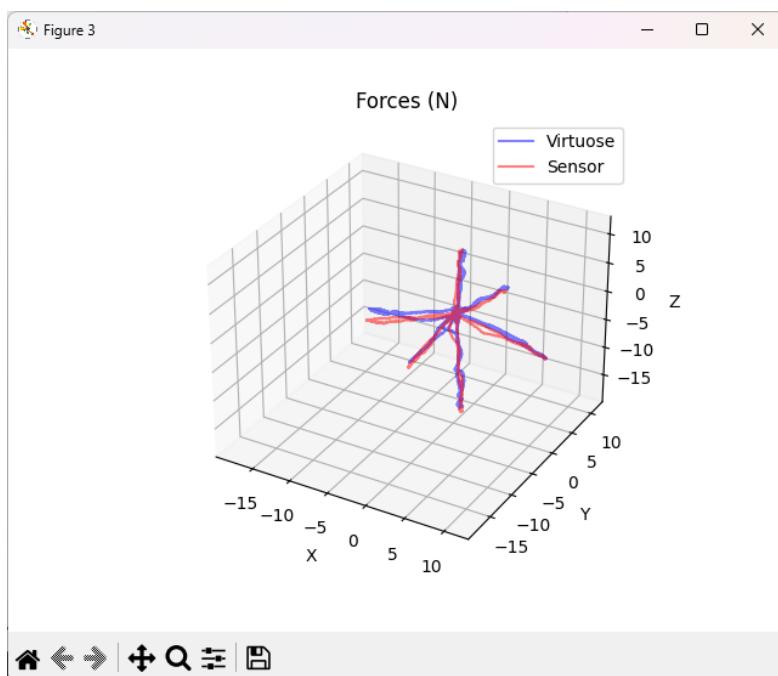


FIGURE 6.4 – Force (N) : Virtuose Vs Sensor

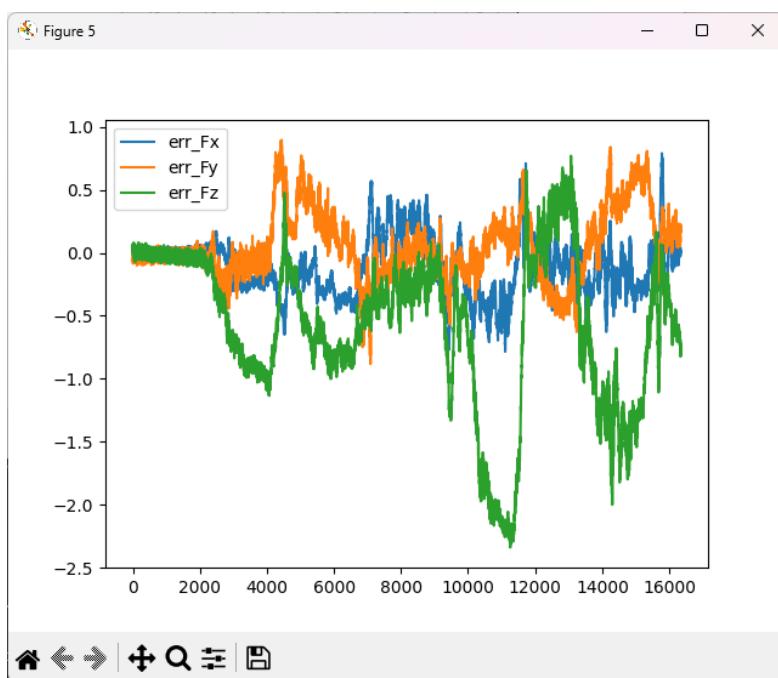


FIGURE 6.5 – Erreur en force (N)

Déviation standard des erreurs :

```
std_Fx = 0.2219901092199246 (N)
std_Fy = 0.26947154456226824 (N)
std_Fz = 0.6135555620658395 (N)
```

Torques :

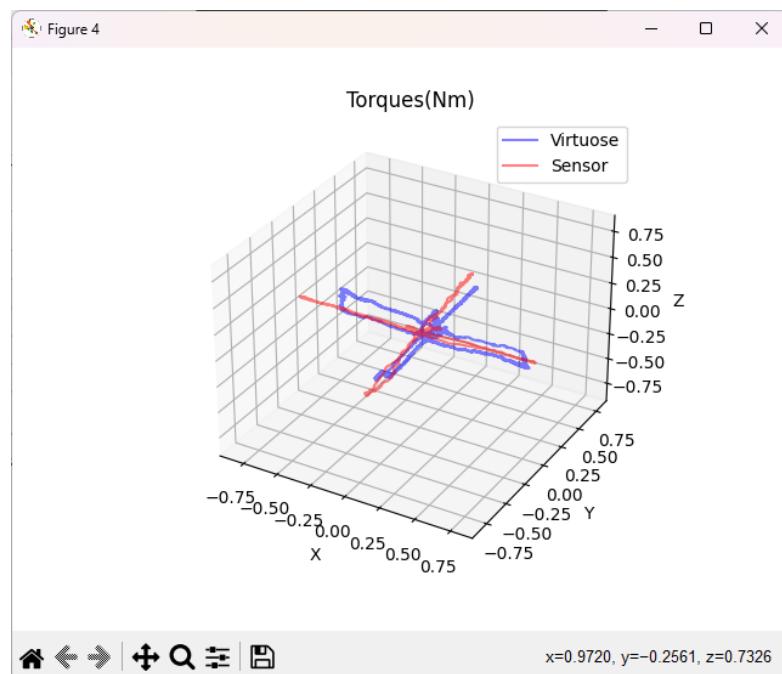


FIGURE 6.6 – Torque (Nm) : Vituose Vs Sensor

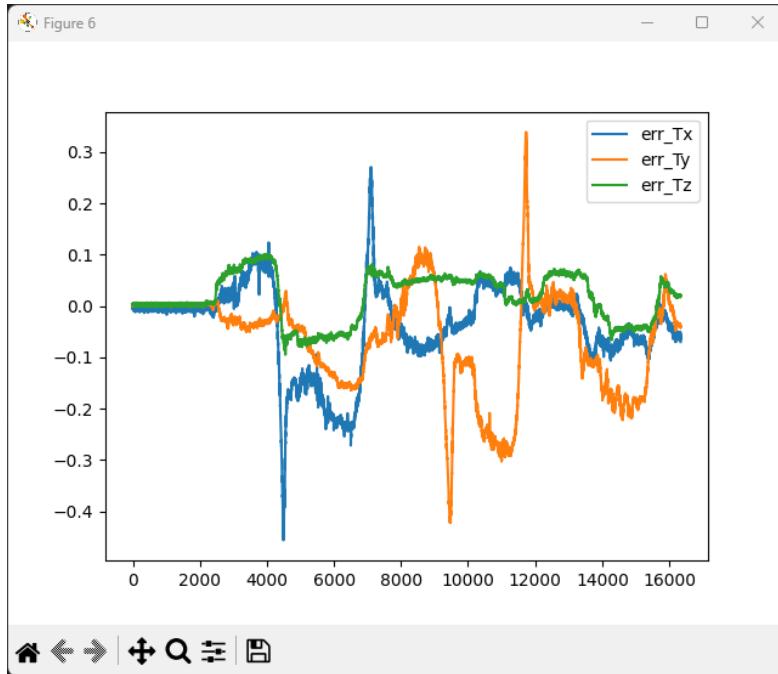


FIGURE 6.7 – Erreur en torque (Nm)

Déviation standard des erreurs :

```
std_Tx = 0.08466452288368423 (Nm)
std_Ty = 0.09863234989948261 (Nm)
std_Tz = 0.04679310033510127 (Nm)
```

7 Identification des coefficients de frottement

7.1 Simplification du problème :

Pour cette partie afin de déterminer les coefficient de frottement on va considérer que le Virtuose est un simple bras manipulateur à un seul degré de liberté θ , une masse m , une inertie J , et son centre d'inertie G est à l'extrémité du bras.

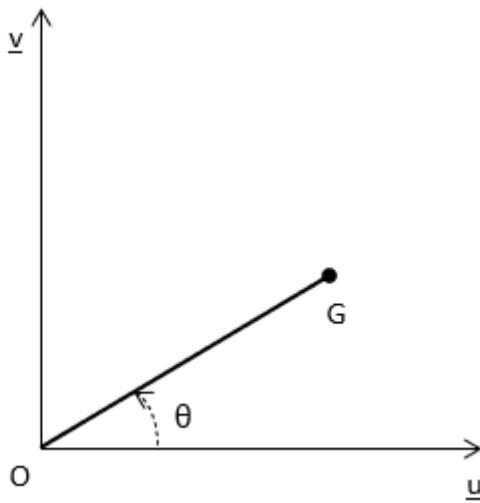


FIGURE 7.1 – Pivot 1dof

7.2 Modèle dynamique inverse simplifié :

Le modèle dynamique inverse revient à exprimer le couple Γ en fonction de l'angle d'articulation et ses dérivées sous la forme :

$$\Gamma = f(\theta, \dot{\theta}, \ddot{\theta}) \quad (1)$$

D'après l'équation de Lagrange on a :

$$\Gamma = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad (2)$$

où :

- $L = E_c - E_p$
- $\frac{d}{dt}$ représente la dérivée par rapport au temps,
- L est l'énergie cinétique du système,
- \dot{q}_i représente la dérivée temporelle de la variable généralisée q_i ,
- q_i représente les variables généralisées du système,
- Γ représente les forces généralisées appliquées sur le système.

L'équation de Lagrange exprime l'idée fondamentale que la somme des travaux des forces généralisées est égale au taux de variation de l'énergie cinétique du système.

En résolvant l'équation de Lagrange, on peut déterminer les équations du mouvement du système et les forces/couples appliqués en fonction des variables généralisées. Cela permet d'analyser et de prédire le comportement du système mécanique de manière mathématique et rigoureuse.

Pour notre simplification on aura :

— Détermination de E_p :

$$E_p = -m \cdot (\overrightarrow{OG} \cdot \vec{g})$$

$$E_p = -m \cdot l \cdot \left(\begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} \cdot \begin{bmatrix} g_u \\ g_v \end{bmatrix} \right)$$

$$E_p = -m \cdot l \cdot (g_u \cdot \cos(\theta) + g_v \cdot \sin(\theta))$$

— Détermination de E_c :

$$E_c = \frac{1}{2} J \dot{\theta}$$

Et donc :

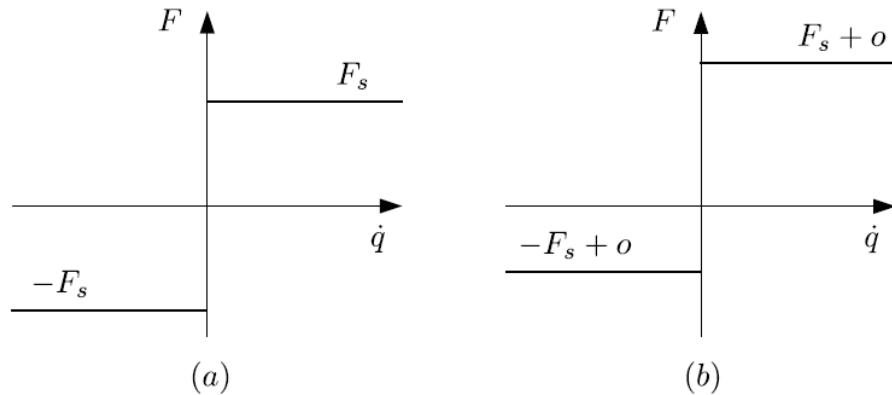
$$L = \frac{1}{2} J \dot{\theta} + m \cdot l \cdot (g_u \cdot \cos(\theta) + g_v \cdot \sin(\theta)) \quad (3)$$

D'où l'expression de Γ sans considération du frottement :

$$\Gamma = J \ddot{\theta} + m \cdot l \cdot (-g_u \cdot \sin(\theta) + g_v \cdot \cos(\theta)) \quad (4)$$

7.2.1 L'ajout du frottement

Maintenant, nous pouvons ajouter le frottement au modèle. Le frottement est un phénomène complexe qui se décompose en deux composantes physiquement distinctes : le frottement visqueux, qui est proportionnel à la vitesse, et le frottement de Coulomb, qui dépend uniquement de la direction du mouvement. Le frottement visqueux pour l'articulation i peut être exprimé comme $F_{vi} \cdot \dot{q}_i$. Dans le cas du frottement de Coulomb, la forme standard est $F_{si} \cdot \text{sgn}(q_i)$, ce qui correspond au frottement de Coulomb symétrique, comme dans la figure, cas (a). Lorsque le frottement de Coulomb n'est pas symétrique, la force de frottement est plus grande dans une direction que dans l'autre, comme dans la figure, cas (b). Ainsi, de manière générale, nous pouvons écrire la force de frottement comme $F_{si} \cdot \text{sgn}(q_i) + o_i$, où o_i est un décalage par rapport au cas symétrique.

**FIGURE 7.2 – frottement de Coulomb**

L'équation du couple devient alors :

$$\Gamma = J.\ddot{\theta} + m.l.(-g_u \cdot \sin(\theta) + g_v \cdot \cos(\theta)) + F_s \cdot \text{sgn}(\dot{\theta}) + o + F_v \cdot \dot{\theta} \quad (5)$$

7.3 Identification des coefficients

Afin de déterminer les coefficients de flottement F_s et F_v j'ai prévu de faire un test d'aller retour avec une vitesse constante pour chaque articulation et donc cela va me permettre d'enregistrer les valeurs du couple fourni à l'articulation pour chaque angle.

la commande que j'utilise en vrai n'est pas une commande en vitesse dans le sens absolue, puisque ça va dépendre du temps de réponse du contrôleur PD déjà implémenté dans le **RaptorAPI** pour le retour haptique du virtuose, et je commande que la position en angle du bras en l'incrémentant à chaque itération par la vitesse désirée multipliée par le temps de cycle.

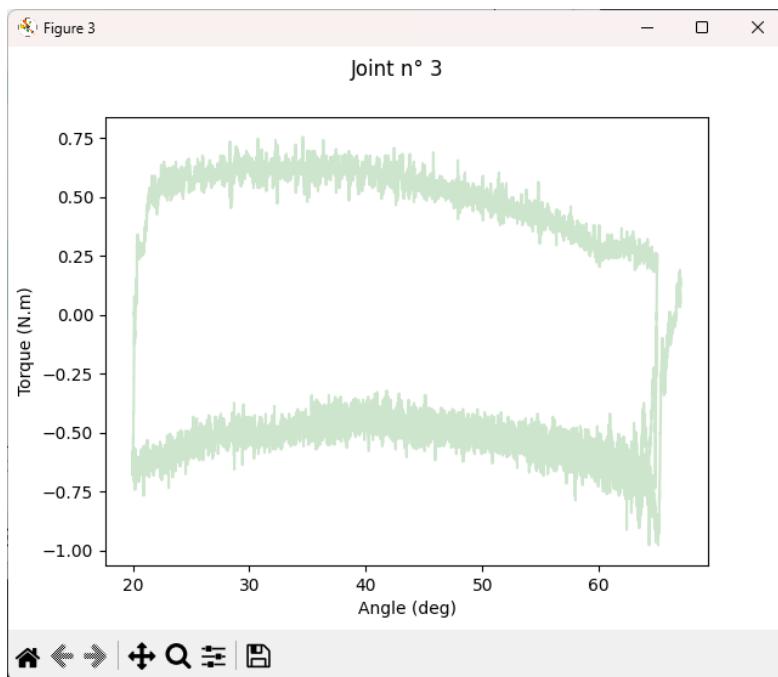


FIGURE 7.3 – Couple en fonction de l’angle

7.3.1 Lissage des données

Dans le contexte du traitement des données, le filtrage est une technique utilisée pour atténuer ou éliminer les composantes indésirables d'un signal, telles que le bruit ou les variations rapides. L'objectif du filtrage est de lisser le signal et de mettre en évidence les caractéristiques essentielles.

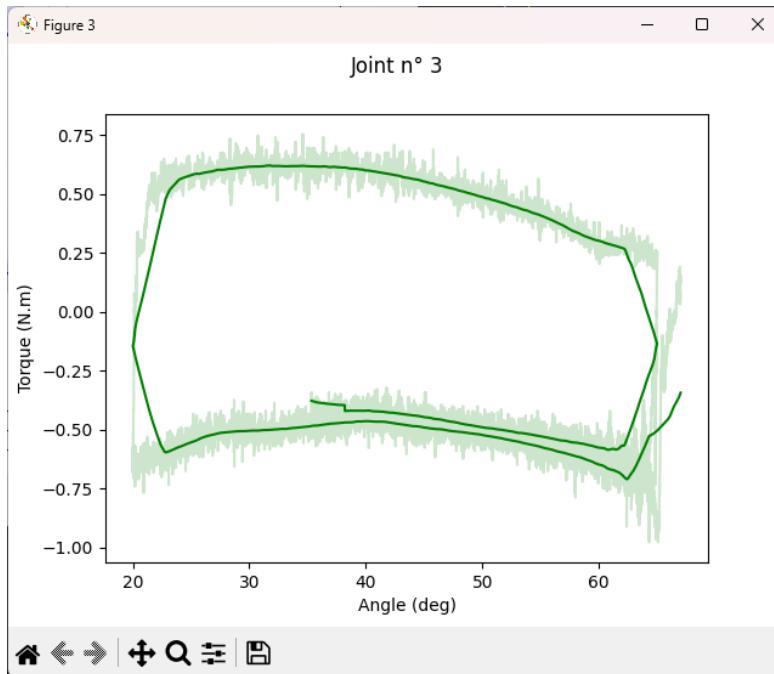


FIGURE 7.4 – Données après lissage

le programme de lissage que j'ai utilisé est le suivant :

- Nous parcourons chaque point du signal (les valeurs **Angle** et **Torque**)
- Pour chaque point, nous définissons une fenêtre glissante qui comprend un certain nombre de points adjacents de chaque côté du point central. La taille de cette fenêtre peut être ajustée en modifiant la variable 'taille_fenetre'.
- Nous calculons la valeur moyenne des points à l'intérieur de la fenêtre glissante. Cela est réalisé en prenant la somme des valeurs des points et en la divisant par le nombre total de points dans la fenêtre.
- La valeur moyenne calculée devient la nouvelle valeur pour le point central de la fenêtre glissante. Cela nous donne un signal filtré qui est lissé par rapport aux variations des points adjacents.

En appliquant cette opération pour chaque point du signal, nous obtenons un signal filtré qui est généralement moins bruité et plus lisse que le signal d'origine.

En utilisant la formule (5) on peut déduire la formule suivante pour notre test donc à vitesse constante et pour le même $\theta = \theta_c$:

$$d = \Gamma_{\theta_c}(\dot{\theta} = \omega) - \Gamma_{\theta_c}(\dot{\theta} = -\omega) = 2.(F_s + F_v \cdot \omega) \quad (6)$$

On va utiliser cette formule pour identifier la somme des deux coefficients et ça va correspondre à la différence entre le torque aller et le torque retour.

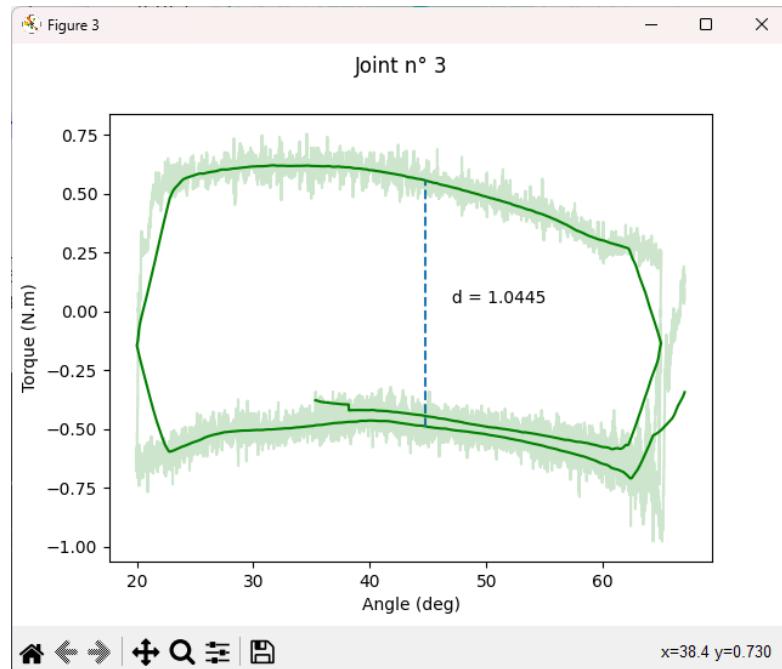


FIGURE 7.5 – Différence entre torque aller et retour

La fonction $d(\omega)$ présente une caractéristique affine. Par conséquent, en effectuant le test avec différentes vitesses, nous pouvons déduire les deux paramètres qui la définissent.

Le deuxième test que j'ai envisagé consiste à répéter la procédure d'aller-retour, tout en variant la vitesse de rotation à chaque itération. À chaque itération, je tracerai la relation entre le couple exercé et le temps de manière à obtenir une courbe de couple en fonction du temps.

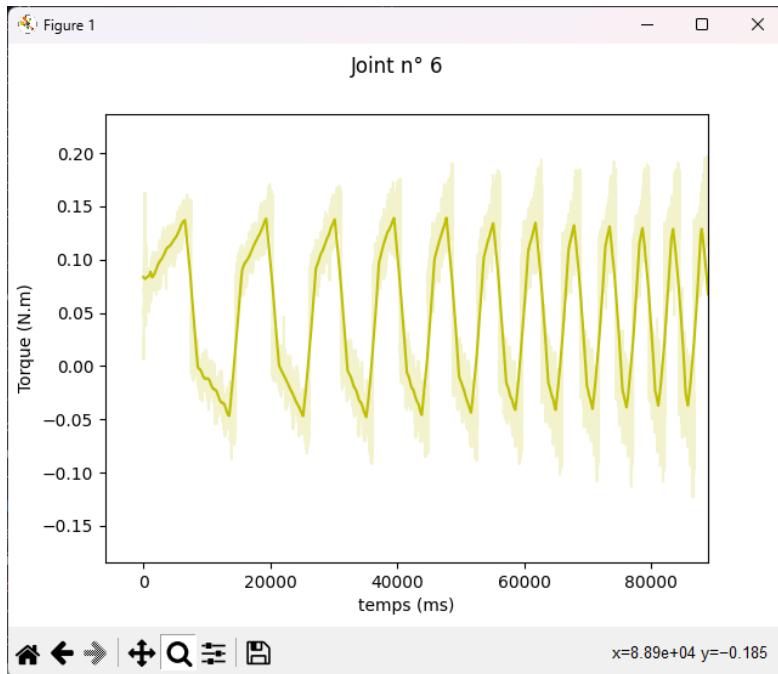


FIGURE 7.6 – $d(\omega)$ en fonction du temps

Effectivement, étant donné que les pics observés dans les données sont généralement proches de la même valeur, on peut supposer que le coefficient F_v (lié au frottement visqueux) est négligeable en comparaison avec F_s (le frottement sec de Coulomb). Dans ce cas, la différence calculée $d(\omega)$ deviendrait approximativement égale au double du frottement de Coulomb F_s . Cette simplification permet de simplifier l'analyse et de considérer principalement l'impact du frottement sec dans les interactions observées.

$$\forall \omega \quad F_s \approx \frac{d(\omega)}{2}$$

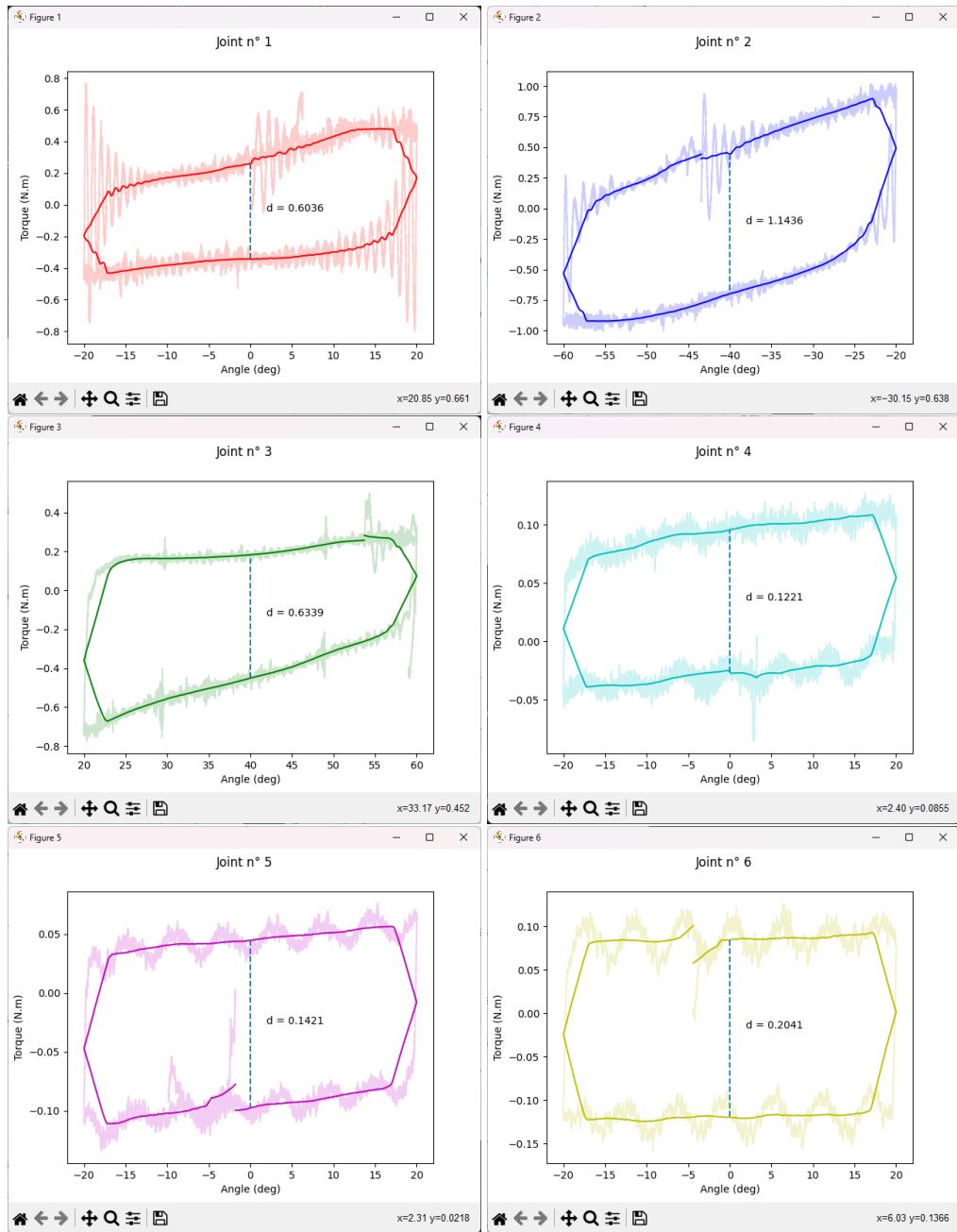


FIGURE 7.7 – Résultats d'identification

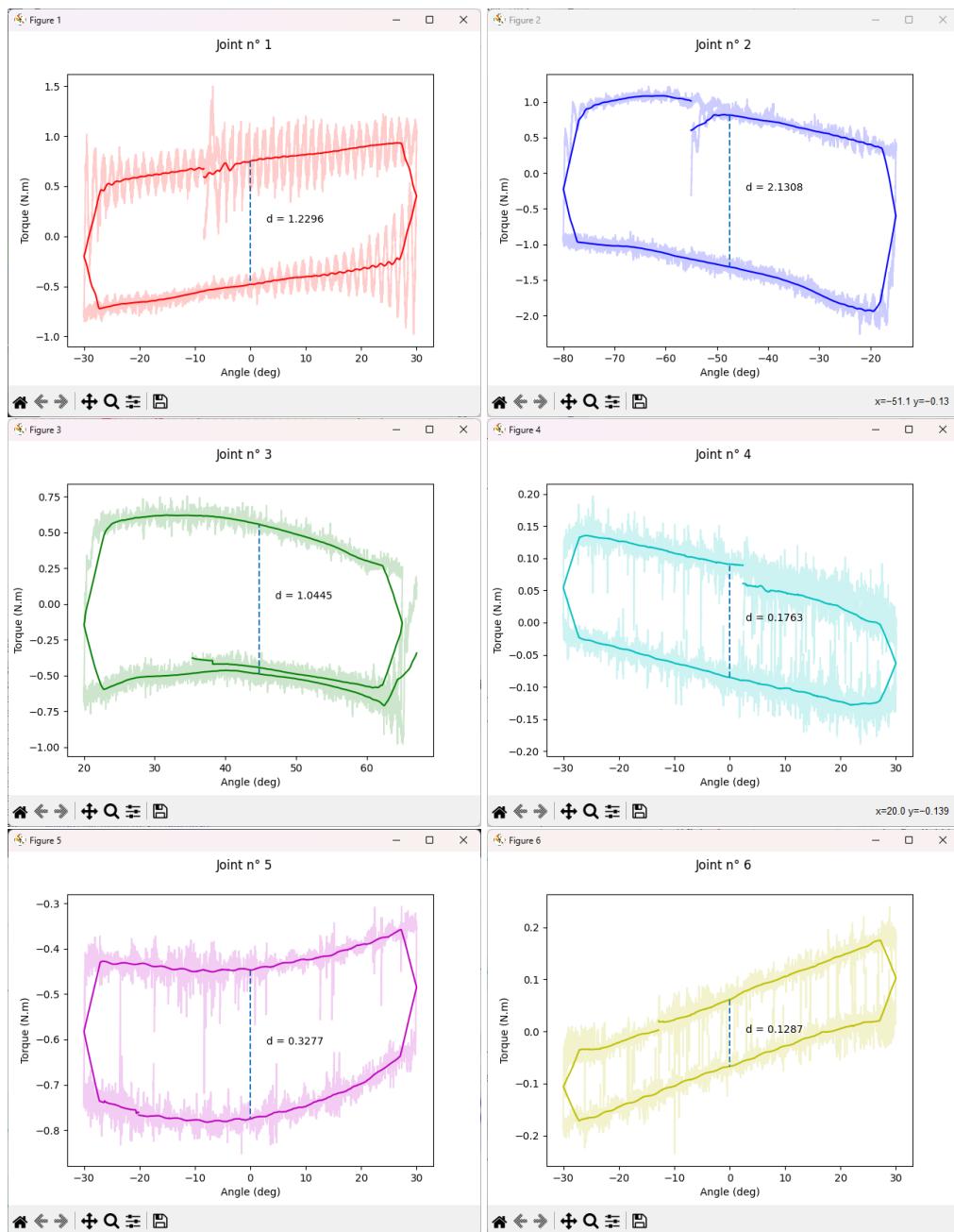


FIGURE 7.8 – Résultats d’identification pour un high force

8 Conclusion

Ce stage a représenté un voyage passionnant au cœur des défis cruciaux liés à la vérification et à la validation des dispositifs haptiques. Au fil de ces semaines de travail acharné, nous avons adopté une démarche méthodique pour explorer en profondeur les méthodes de vérification, aussi bien en usine qu'au près du client, concernant les débattements et les efforts générés par ces dispositifs. Ce rapport a mis en lumière plusieurs aspects essentiels de cette expérience.

L'originalité de notre démarche réside dans la combinaison de la recherche minutieuse et de l'application pratique. Nous avons entrepris une analyse exhaustive des besoins de vérification des débattements et des efforts, en soulignant l'importance cruciale de la précision, de la conformité aux spécifications et de la simplicité des méthodes utilisées. La recherche de solutions a été le fil conducteur de notre travail, de la calibration géométrique à l'expérimentation de validation chez le client. Cette approche holistique a permis d'aborder les problèmes sous différents angles et d'explorer diverses méthodologies pour y répondre.

Les perspectives découlant de cette expérience sont prometteuses. Les compétences techniques acquises dans la vérification des dispositifs haptiques sont inestimables, et elles sont directement applicables à l'industrie. La compréhension approfondie des exigences de vérification permettra de garantir que les produits haptiques répondent aux normes de qualité les plus strictes et offrent une expérience utilisateur immersive et satisfaisante. De plus, la recherche sur l'identification des coefficients de frottement ouvre la voie à des améliorations futures dans la modélisation et la simulation de ces dispositifs.

En dehors de la démarche professionnelle, ce stage a également été une expérience humaine enrichissante. La collaboration avec l'équipe d'Haption a été une source d'inspiration, et les échanges fructueux avec les collègues ont contribué à notre développement professionnel. Le contexte du travail au sein de cette entreprise innovante nous a offert une perspective précieuse sur le monde industriel.

En conclusion, ce stage a été bien plus qu'une simple expérience professionnelle. Il a été une aventure intellectuelle et humaine. Les compétences acquises, les défis relevés et les perspectives ouvertes font de cette expérience un chapitre mémorable de notre parcours. Elle renforce notre conviction que la vérification et la validation des dispositifs haptiques sont essentielles pour l'avenir de cette technologie immersive. Avec gratitude envers toutes les personnes impliquées, nous clôturons ce stage en regardant vers l'avenir avec un enthousiasme renouvelé pour les opportunités qui se présentent.

Annexe

Quelques fonctions utiles

```

# fonction pour trouver les coordonnees du nouveau
end_effector

def effector_coord(data=default_data, effector_coord_local=
default_effector_coord_local):
    x, y, z = [], [], []
    CRI_x, CRI_y, CRI_z = [], [], []

    5
    with open(sys.path[0]+'\..\..\log\'+data, 'r') as file:
        for line in file:
            x_, y_, z_, q_x_, q_y_, q_z_, q_w_ = map(float, line.
                split())
    10
    q=np.array([q_x_, q_y_, q_z_, q_w_])

    CRI = np.array([x_, y_, z_])
    matrice_rotation = quaternion_to_rotation_matrix(q)
    vecteur_translation = CRI.reshape(3, 1)
    15
    matrice_transformation = np.concatenate((
        matrice_rotation, vecteur_translation), axis=1)
    matrice_transformation = np.vstack((
        matrice_transformation, np.array([0, 0, 0, 1])))

    effector_coord_global = matrice_transformation @ np.
        append(effector_coord_local, 1)
    20
    x.append(effector_coord_global[0])
    y.append(effector_coord_global[1])
    z.append(effector_coord_global[2])

    25
    CRI_x.append(CRI[0])
    CRI_y.append(CRI[1])
    CRI_z.append(CRI[2])

    return x,y,z,CRI_x,CRI_y,CRI_z

```

```
from scipy.optimize import minimize
from sklearn.cluster import KMeans

5 #fonction objective pour l'optimisation du StyloCalibration
def f(X):

    effector_coord_local = np.array(X)

10
    x,y,z,_,_,_ = effector_coord(data, effector_coord_local)
    data_ = np.column_stack((x, y, z))
    kmeans = KMeans(n_clusters=1, n_init=10)
    kmeans.fit(data_)

15
    Obj_func = kmeans.inertia_

    print("Objective function :")
    print(Obj_func)

20
    return Obj_func

def optimize():

25     bounds = ((-1, 1), (-1, 1), (-1, 1)) # La plage pour
                                                les trois variables

        result = minimize(f, (175e-03, -0.25e-03, 109.9e-03),
                           method='SLSQP', bounds=bounds) # Minimisation

30     print(result)
     return result
```

```

# params contient le vecteur de translation plus le vecteur
# angle d'euler
def construct_transformation_matrix(params):
    # Exemple : Matrice de transformation avec translations
    # et rotations
    translation = np.array([[1, 0, 0, params[0]],
                           [0, 1, 0, params[1]],
                           [0, 0, 1, params[2]],
                           [0, 0, 0, 1]])

    # Construction de la matrice de rotation      partir des
    # angles de rotation
    10   rotation_x = np.array([[1, 0, 0, 0],
                               [0, np.cos(params[3]), -np.sin(
                                   params[3]), 0],
                               [0, np.sin(params[3]), np.cos(
                                   params[3]), 0],
                               [0, 0, 0, 1]])

    15   rotation_y = np.array([[np.cos(params[4]), 0, np.sin(
        params[4]), 0],
                               [0, 1, 0, 0],
                               [-np.sin(params[4]), 0, np.cos(
                                   params[4]), 0],
                               [0, 0, 0, 1]])

    20   rotation_z = np.array([[np.cos(params[5]), -np.sin(
        params[5]), 0, 0],
                               [np.sin(params[5]), np.cos(params
                                   [5]), 0, 0],
                               [0, 0, 1, 0],
                               [0, 0, 0, 1]])

    25   rotation = np.dot(rotation_z, np.dot(rotation_y,
                                             rotation_x))

    transformation_matrix = np.dot(transformation, rotation)

    return transformation_matrix

```

```

# Trouver la matrice de transformation entre deux bases en
# connaisant l'expression d'une trajectoire dans les deux
offset = np.load("offset_marker.npy")
# Read data from the file
data_file = 'camera_log/trajectory.txt'
5 points = []
rotations = []
with open(data_file, 'r') as file:
    for line in file:
        values = line.strip().split()
        # Extract point coordinates (x, y, z)
        10 point = [float(values[0])/1000, float(values[1])/
                    1000, float(values[2])/1000]
        points.append(point)
points = np.array(points)
data = "Trajectory.txt"
15 X,Y,Z,CRIx,CRIy,CRIz = effector_coord(data,offset)

virtuose_points = np.array([X,Y,Z]).T

def objectif(X):
20    transformation_matrix = construct_transformation_matrix(
        X)

    points_homogeneous = np.hstack((points, np.ones((points.
        shape[0], 1))))

```

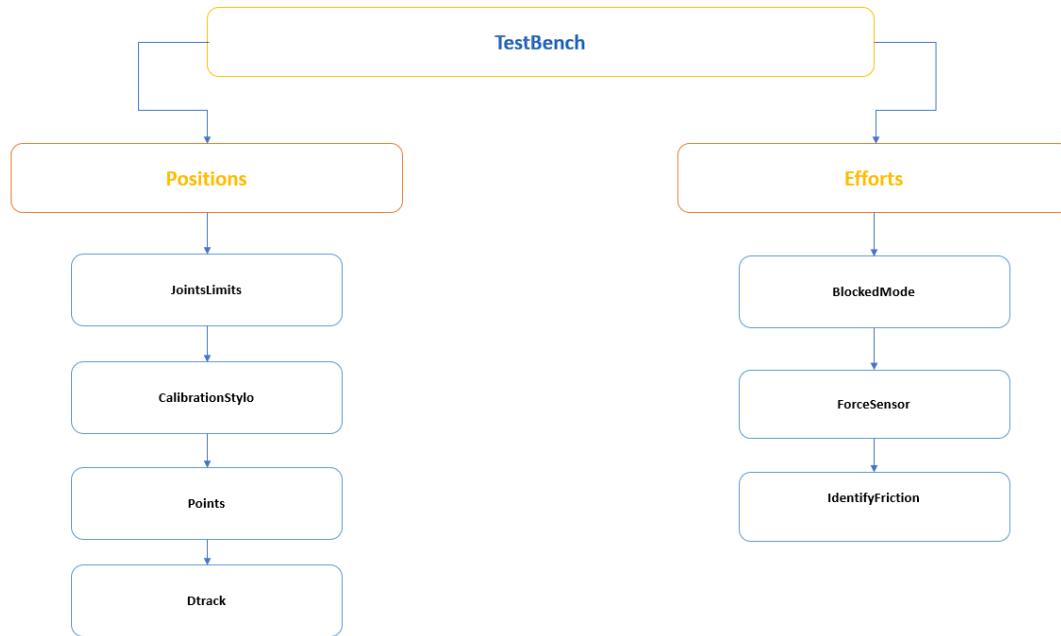
25 # Appliquer la transformation aux points de la
 # trajectoire

```

    transformed_points_homogeneous = np.dot(
        transformation_matrix, points_homogeneous.T).T
    # Enlever la colonne des coordonnées homogènes
    transformed_points = transformed_points_homogeneous[:, :
        3]
    M = transformed_points - virtuose_points
30    result = np.linalg.norm(M)
    return result

```

Plan du banc de test



Mots clés :

API : Application Programming Interface
ART : Advanced Real-time Tracking
CRI : Centre de Rotation Instantané
TCP : Tool Center Point
MGD : Modèle Géométrique Direct
DOF : Degree Of Freedom