





UNIVERSITÉ DE SHERBROOKE  
Faculté de génie  
Département de génie électrique et de génie informatique

# TÉLÉOPÉRATION D'UN ROBOT COLLABORATIF PAR OUTIL HAPTIQUE

Mémoire de maîtrise  
Spécialité : génie électrique

Tristan CROSSMAN

Jury : Wael SULEIMAN (Directeur)  
Ruben GONZALEZ-RUBIO (Rapporteur)  
Gustavo ARECHAVALETA (Examineur)



# RÉSUMÉ

Ce projet de recherche, intitulé *Téléopération d'un robot collaboratif par outil haptique* traite un des problèmes contemporains de la robotique, à savoir la coopération entre l'humain et la machine.

La robotique est en pleine expansion depuis maintenant deux décennies : les robots investissent de plus en plus l'industrie, les services ou encore l'assistance à la personne et se diversifient considérablement. Ces nouvelles tendances font sortir les robots des cages dans lesquelles ils étaient placés et ouvrent grand la porte vers de nouvelles applications. Parmi elles, la coopération et les interactions avec l'humain représentent une réelle opportunité pour soulager l'homme dans des tâches complexes, fastidieuses et répétitives.

En parallèle de cela, la robotique moderne s'oriente vers un développement massif du domaine humanoïde. Effectivement, plusieurs expériences sociales ont montré que l'être humain, constamment en interaction avec les systèmes qui l'entourent, a plus de facilités à contribuer à la réalisation d'une tâche avec un robot d'apparence humaine plutôt qu'avec une machine.

Le travail présenté dans ce projet de recherche s'intègre dans un contexte d'interaction homme-robot (IHR) qui repose sur la robotique humanoïde. Le système qui en découle doit permettre à un utilisateur d'interagir efficacement et de façon intuitive avec la machine, tout en respectant certains critères, notamment de sécurité. Par une mise en commun des compétences respectives de l'homme et du robot humanoïde, les interactions sont améliorées. En effet, le robot peut réaliser une grande quantité d'actions avec précision et sans se fatiguer, mais n'est pas nécessairement doté d'une prise de décision adaptée à la situation, contrairement à l'homme qui est capable d'ajuster son comportement naturellement ou en fonction de son expérience. En d'autres termes, ce système cherche à intégrer le savoir-faire et la capacité de réflexion humaine avec la robustesse, l'efficacité et la précision du robot. Dans le domaine de la robotique, le terme d'interaction intègre également la notion de contrôle. La grande majorité des robots reçoit des commandes machines qui sont généralement des consignes de trajectoire, qu'ils sont capables d'interpréter. Or, plusieurs interfaces de contrôle sont envisageables, notamment celles utilisant des outils haptiques, qui permettent à un utilisateur d'avoir un ressenti et une perception tactile. Ces outils comme tous ceux qui augmentent le degré de contrôle auprès de l'utilisateur, en ajoutant un volet sensoriel, sont parfaitement adaptés pour ce genre d'applications. Dans ce projet, deux outils haptiques sont assemblés puis intégrés à une interface de contrôle haptique dans le but de commander le bras d'un robot humanoïde. Ainsi, l'homme est capable de diriger le robot tout en ajustant ses commandes en fonction des informations en provenance des différents capteurs du robot, qui lui sont retranscrites visuellement ou sensoriellement.

**Mots-clés :** Téléopération d'un robot, collaboration homme-robot, contrôleur haptique



# REMERCIEMENTS

La réalisation de ce mémoire ainsi que la réussite de cette expérience universitaire ont été possibles grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma reconnaissance.

En premier lieu, je voudrais adresser mes remerciements à Wael Suleiman, professeur au département de génie électrique et de génie informatique à l'Université de Sherbrooke. En tant que directeur de recherche, il a su me guider dans l'avancement de ma recherche et m'a aidé à trouver des solutions à chaque fois qu'un problème se présentait à moi. Sa disponibilité et surtout ses judicieux conseils ont grandement contribué à alimenter ma réflexion tout au long de mon travail.

Je souhaite remercier l'équipe de l'assistance technique de Rethink Robotics et tout particulièrement Ian McMahon qui a accepté de répondre à mes questions tout en me fournissant des axes de progression sur le robot Baxter. Mes remerciements s'adressent également à la communauté du kit de développement logiciel du robot collaboratif, dont l'activité a grandement facilité mon travail. Mes remerciements vont autant à Patrick Beeson et Barrett Ames, qui ont fourni des outils d'optimisation (TRAC-IK) pour l'algorithme de la cinématique inverse.

Je n'oublierai pas d'exprimer ma reconnaissance à l'entreprise émergente 3D TRIP qui a apporté sa contribution au projet de recherche en imprimant des pièces de qualité, essentielles à l'aboutissement des objectifs fixés.

Enfin, je tiens à témoigner toute ma gratitude à ma famille, à tous mes proches et amis, qui m'ont toujours encouragé en m'apportant leur support moral et intellectuel au cours de la réalisation de ce mémoire et plus généralement tout au long de mon cursus au sein de l'Institut Interdisciplinaire d'Innovation technologique (3IT).





# TABLE DES MATIÈRES

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>2</b>	<b>LES INTERACTIONS HOMME-MACHINE</b>	<b>5</b>
2.1	La robotique et les interactions homme-robot . . . . .	5
2.2	Les interactions homme-robot à distance . . . . .	8
2.3	Les interactions homme-robot locales . . . . .	10
2.4	Les outils des interactions homme-robot . . . . .	11
2.5	Les dispositifs haptiques et leurs applications . . . . .	14
2.6	Contexte et enjeux des interactions homme-robot . . . . .	18
<b>3</b>	<b>UNE INTERFACE UTILISATEUR NATURELLE</b>	<b>21</b>
3.1	Le joystick haptique Falcon . . . . .	21
3.1.1	Structure articulaire . . . . .	22
3.1.2	Cinématique inverse . . . . .	23
3.1.3	Cinématique directe . . . . .	25
3.1.4	Libnifalcon . . . . .	26
3.2	Le robot collaboratif Baxter . . . . .	28
3.3	Construction d'un système haptique à cinq DDL . . . . .	30
3.3.1	Modifications matérielles . . . . .	30
3.3.2	Compensation de gravité . . . . .	32
3.3.3	Contrôleur à cinq DDL . . . . .	34
<b>4</b>	<b>UTILISATION D'UN CONTRÔLEUR HAPTIQUE POUR LA MANI- PULATION D'UN BRAS ROBOTIQUE</b>	<b>37</b>
4.1	La plateforme robotique ROS . . . . .	37
4.2	Le contrôleur haptique à cinq degrés de liberté . . . . .	38
4.2.1	L'espace de travail . . . . .	38
4.2.2	Planification de trajectoire pour le contrôleur . . . . .	40
4.3	Téléopération du robot collaboratif Baxter . . . . .	46
4.3.1	Description de la stratégie . . . . .	46
4.3.2	Les procédures de commandes . . . . .	48
4.3.3	Filtrage des commandes . . . . .	54
4.3.4	La rétroaction haptique du contrôleur à cinq DDL . . . . .	55
4.3.5	Téléopération du robot . . . . .	58
<b>5</b>	<b>CONCLUSION</b>	<b>63</b>
	<b>LISTE DES RÉFÉRENCES</b>	<b>65</b>



# LISTE DES FIGURES

1.1	Photo de Hiroshi Ishiguro (à gauche) et son double humanoïde <i>Geminoid HI-2</i> (à droite) [Ishiguro, 2011] . . . . .	1
1.2	Exemples de robots humanoïdes . . . . .	2
2.1	Évolution de la robotique de service dans le secteur professionnel [IFR, 2015]	6
2.2	Deux exemples d’interaction directe avec un robot : en haut, une communication vocale avec l’humanoïde <i>Pepper</i> [Aldebaran Robotics, 2015], en bas, un contact gestuel entre un humain et un robot. . . . .	12
2.3	Schéma explicite d’une interaction indirecte avec un bras robotique. . . . .	13
2.4	Exemples d’outils haptiques . . . . .	14
2.5	Schéma de la communication entre le simulateur iCub, l’interface haptique et le joystick. . . . .	16
3.1	Le joystick haptique Falcon avec différents effecteurs : à gauche il s’agit de l’effecteur standard qui vient à l’achat de l’appareil, à droite, le poignet pistolet pour les jeux vidéo . . . . .	21
3.2	Structure mécanique du Novint Falcon [Michelotti, 2011] . . . . .	23
3.3	Schémas de la structure articulaire du Falcon pour le calcul de la cinématique	24
3.4	Description du processus de calibration du Falcon avant qu’un utilisateur puisse l’utiliser . . . . .	27
3.5	Photo du cobot Baxter de Rethink Robotics . . . . .	29
3.6	Vue de face de l’outil haptique qui combine les deux Falcons . . . . .	31
3.7	Vue d’ensemble de la disposition des Falcons pour la configuration verticale du système haptique (vue de dessus) . . . . .	31
3.8	Centre de masse du lien en courbe de l’outil haptique Novint Falcon . . . . .	32
3.9	Forces et couples appliqués sur la $i^{eme}$ chaine cinématique du Falcon . . . . .	32
3.10	Schématisation des repères pour le contrôleur construit à partir des deux Falcons. P correspond au point central du stylet, représenté par un rectangle rouge . . . . .	34
3.11	Orientation du stylet selon la norme <i>RPY</i> des angles d’Euler . . . . .	35
4.1	Comparaison des espaces de travail entre le contrôleur haptique à 5 DLL et l’effecteur standard du Falcon. À gauche, les différents plans correspondent aux espaces accessibles pour l’effecteur et à droite à ceux du nouveau joystick haptique. . . . .	39
4.2	Illustrations des problèmes rencontrés si l’on fait varier la fréquence . . . . .	42
4.3	Suivi de la trajectoire en position du stylet (fréquence de 800Hz) . . . . .	43
4.4	Suivi de la trajectoire en orientation du stylet . . . . .	44
4.5	Réponse du contrôleur face à une impulsion (à gauche avec la compensation de gravité et à droite sans). Dans cette expérience, la position désirée (courbe rouge) est envoyée sous forme d’impulsion pendant 1s puis arrêtée brutalement afin d’observer l’évolution de la position du stylet. . . . .	45

4.6	Description du système de téléopération du robot Baxter . . . . .	47
4.7	Schéma de l'algorithme de cinématique inverse . . . . .	49
4.8	Le graphique de gauche représente la trajectoire des joints ( $s_0$ , $s_1$ , $w_0$ , $w_1$ , $w_2$ , $e_0$ et $e_1$ ) lorsque les commandes ne sont pas filtrées. À droite un filtre numérique d'ordre 4 est appliqué sur les mêmes données, à partir de 90ms (trait vertical magenta) . . . . .	54
4.9	Seuil de détection de contact/collision . . . . .	56
4.10	Interpolation polynomiale pour le facteur force appliqué au contrôleur haptique . . . . .	57
4.11	Scénario de téléopération à distance . . . . .	59
4.12	Transformation entre le bras contrôlé et le bras suiveur. . . . .	60
4.13	À droite, retour visuel de la configuration temps réel du robot collaboratif. À gauche, les retours visuels des caméras disposées aux extrémités de chacun des bras de Baxter . . . . .	61
4.14	Aperçu de l'espace de travail accessible par Baxter [Rethink Robotics, 2012] . . . . .	62

# LISTE DES TABLEAUX

3.1	Tableau des différentes variables pour la cinématique du Novint Falcon [Machulis, 2007] . . . . .	33
-----	---	----



# LISTE DES ACRONYMES

Acronyme	Définition
IHR	Interaction homme-robot
3IT	Institut Interdisciplinaire d'Innovation technologique
EMG	Électromyographie
E/S	Entrées/Sorties
IFR	In- ternational Federation of Robotics
IHM	Interaction homme-machine
IEEE	Institute of Electrical and Electronics Engineers
RoMan	Robot and Human Interactive Communication
AM-FM	Minimum phase-All phase
TCP/IP	Transmission Control Protocol/Internet Protocol
pHRI	Interaction physique homme-robot
GUI	Interface Utilisateur Graphique
NUI	Interface Utilisateur Naturelle
FPGA	Field-Programmable Gate Array
VR	Virtual Reality
FPS	First-Person Shooter
DDL	Degrés De Liberté
KDL	Kit de Développement Logiciel
USB	Universal Serial Bus
FTDI	Future Technology Devices International
UART	Universal Asynchronous Receiver Transmitter
DSP	Digital Signal Processor
BSD	Berkeley Software Distribution
RPY	Roll-Pitch-Yaw
LCD	Liquid Crystal Display
API	Application Programming Interface
ROS	Robot Operating System
OS	Operating System
URDF	Unified Robot Description Format
RSDK	Research Software Development Kit
XML	Extensible Markup Language
PID	Proportional Integral Derivative
KDL	Kinematics and Dynamics Library
IK	Inverse Kinematic
MPC	Model Predictive Control
QP	Quadratic Problem
SQP	Sequential Quadratic Programming
Hz	Hertz
N	Newton





# CHAPITRE 1

## INTRODUCTION

De nos jours, on trouve les robots un peu partout, dans les usines et dans les champs, au fond des mers et dans l'espace, dans les jardins et dans les salons. En 2013, il a été recensé près d'un robot pour 5000 personnes selon les chiffres fournis par Marshall Brain, l'auteur de *Robotic Nation* et le fondateur de *How Stuff Works*. En outre, ils n'ont pas seulement pénétré le monde industriel, ils sont en train de pénétrer notre vie quotidienne et notre culture. Effectivement, les prouesses technologiques permettent aux robots d'aujourd'hui de prendre une apparence humaine. Pour ne citer qu'un exemple Hiroshi Ishiguro, un des pionniers de la robotique a créé un automate de sa propre personne (voir figure 1.1). Cette



Figure 1.1 Photo de Hiroshi Ishiguro (à gauche) et son double humanoïde *Geminoid HI-2* (à droite) [Ishiguro, 2011]

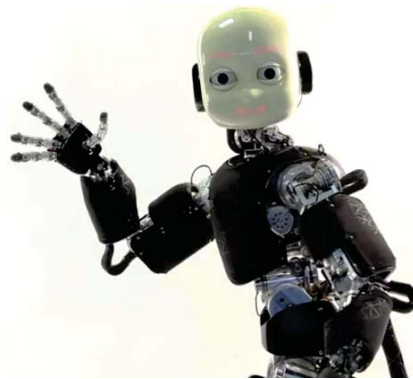
génération de robot relativement récente que l'on nomme androïde ou encore humanoïde, a été dotée de cette forme humaine dans le but de faciliter leur intégration dans notre quotidien [Pamela J. Hinds, 2004], mais leurs fonctions restent similaire aux anciennes générations, à savoir remplacer l'homme dans des tâches précises et répétitives. Les recherches dans ce secteur sont actuellement très actives et ne cessent d'évoluer depuis la fin du XX<sup>e</sup> siècle. Par exemple, l'entreprise Boston Dynamics, qui s'est fait connaître en premier lieu pour son robot *Big Dog* (dont le principal objectif

était de pouvoir se déplacer dans n'importe quel milieu sans jamais tomber) a récemment annoncé une version humanoïde de ce robot chien (*Atlas*).

Jusqu'à récemment les systèmes robotiques s'inscrivaient dans le domaine des applications industrielles et pour la recherche académique. Ces dernières années, la technologie robotique est devenue beaucoup plus fine et les progrès sont continus et persistants. Elle élabore maintenant des robots androïdes très réalistes comme *Nao* [Gouaillier *et al.*, 2009] et son grand frère *Roméo* de la compagnie Aldebaran Robotics, *Asimo* [Sakagami *et al.*, 2002a] de Honda ou encore *Robotis-OP2*. La robotique humanoïde a très vite été le moyen de



(a) Asimo de Honda  
[Sakagami *et al.*, 2002b]



(b) iCub, l'humanoïde open source  
[Tsagarakis *et al.*, 2007]

Figure 1.2 Exemples de robots humanoïdes

regrouper les différentes disciplines que sont la mécanique, l'informatique et l'automatique dans la réalisation d'un projet commun. Eiichi Yoshida, codirecteur du laboratoire JRL AIST-CNRS UMI3218/RL, donne une définition on ne peut plus simple de cette nouvelle spécialité : « *un robot humanoïde est une des réalisations ultimes de robot : il peut se déplacer dans des environnements conçus pour l'Homme, utiliser des outils ou appareils conçus pour l'humain, et communiquer avec nous de manière multimodale* ».

Les résultats de cette évolution en cours font état d'une expansion de la robotique vers des environnements domestiques, des bureaux ou encore d'autres lieux habités par l'homme. Alors que la robotique classique et industrielle s'articulait autour des trois *D*, « *dull, dirty and dumb* », caractérisant les travaux sans intérêt, ennuyeux et sales pour lesquels les robots devaient remplacer les hommes, cette nouvelle robotique se développe autour des trois *E*, « *education, entertainment, everyday* », dans laquelle les robots sont présents pour nous accompagner au quotidien, pour nous rendre service, nous stimuler ou nous divertir. Un des principaux enjeux de ce changement se trouve être l'interaction entre l'homme et la machine. L'intérêt d'une telle collaboration est la mise en commun des savoirs respectifs de l'Homme et de la machine dans le but de réaliser une certaine tâche ou action. En effet, l'Homme fait preuve d'une grande souplesse et d'une flexibilité, d'une capacité de réflexion ou encore d'une prise de décision face à des événements imprévisibles qui demandent plus d'expérience, mais malgré cela, sa force et son endurance ont des limites. Les robots, quant à eux, ne sont jamais fatigués, peuvent soulever de lourdes charges pendant une longue durée avec une précision millimétrique sans jamais se lasser de la répétitivité d'un travail.

Alors que beaucoup d'ouvriers dans les usines voient leur travail peu à peu remplacé par des machines, une telle coopération ne pourrait être que bénéfique puisqu'elle écarterait les ouvriers d'un éventuel licenciement tout en satisfaisant les PDG quant à l'amélioration de la productivité. Mais attention, un tel partenariat n'exclut pas certains risques, comme le cas de cet ouvrier d'une usine Volkswagen à Kassel qui a été tué par un robot en juillet 2015. Cette dernière remarque soulève un point très important de ces nouvelles applications : le critère de sécurité, primordial à la réussite d'une telle collaboration. La sécurité est d'autant plus importante que les robots sont de plus en plus amenés à partager un espace de travail avec des humains. Cela implique alors des systèmes capables d'adapter leur comportement envers leur homologue humain afin de garantir une sécurité absolue pour ce dernier. Il s'agit d'une simple évolution des principes de base instaurés par l'écrivain de science-fiction Isaac Asimov [Asimov, 1950], qui stipulent que :

- « un robot ne peut porter atteinte à un être humain ni, restant passif, permettre qu'un être humain soit exposé au danger » (loi 1)
- « un robot doit obéir aux ordres que lui donne un être humain, sauf si de tels ordres entrent en conflit avec la Première loi » (loi 2)
- « un robot doit protéger son existence tant que cette protection n'entre pas en conflit avec la Première ou la Deuxième loi » (loi 3)

En parallèle de l'évolution des applications robotiques vers les interactions homme-robot, se sont développés différents systèmes de contrôle. Hormis les machines totalement autonomes, des commandes doivent être communiquées au robot pour qu'une action soit réalisable. À l'origine, ces consignes étaient principalement transmises par ordinateur mais des techniques qui relevaient, il y a peu de temps, de la science-fiction sont actuellement employées : les robots sont contrôlables au moyen de la pensée (EMG), grâce à des téléphones intelligents ou par le biais de la réalité virtuelle comme les gants sensoriels et les joysticks haptiques. Cette dernière catégorie, l'haptique, a fait l'objet de beaucoup de recherches dans le domaine des interactions homme-robot, puisqu'elle fournit à un utilisateur la sensation du toucher. Typiquement, ces outils échangent des informations (périphériques d'entrée-sortie E/S) entre le processeur (ordinateur) et les différents périphériques qui lui sont associées et sont dotés d'éléments mécaniques permettant un ressenti physique comme une vibration ou un mouvement.

Ce projet de recherche aborde la problématique du contrôle robotique avec comme appui la technologie haptique. Les outils haptiques ont été brièvement introduits ci-haut et sont de plus en plus utilisés en robotique : en particulier, ils permettent à leurs utilisateurs de

ressentir physiquement des forces ou des couples. De plus, parmi les cinq sens humains, le toucher est le plus compétent de tous, le seul capable d'échanger simultanément des informations (périphérique E/S).

D'autre part, cette étude s'intéresse particulièrement aux nouvelles générations de robots, les robots collaboratifs spécialement conçus pour interagir avec les humains. Nous sommes partis du principe qu'un contrôle peut être amélioré de manière significative si l'homme et le robot travaillent en tant que partenaires. Effectivement en choisissant correctement un type de robot et un outil de contrôle adapté à une tâche spécifique, un système plus performant est développé. En somme, cette recherche vise à contrôler un robot collaboratif en utilisant des outils haptiques.

Nos motivations nous ont poussé à utiliser des logiciels à code source ouvert (*open source*) afin d'ajouter des nouvelles fonctionnalités ou d'en modifier d'autres, déjà existantes. À la différence des logiciels commerciaux où le code source n'est pas accessible et où des restrictions sont liées à la licence, l'*open source* présente l'avantage d'être gratuit et possède une communauté active de développeurs qui travaillent en collaboration, surtout en milieu académique. À travers ce projet, on s'efforce d'utiliser un maximum d'outils *open source*, pour les raisons citées ci-dessus.

La contribution de ce projet de recherche est la création d'une plateforme logicielle qui permet de contrôler un robot collaboratif grâce à la technologie haptique. Ce développement logiciel intègre une communication bilatérale entre un opérateur et un robot collaboratif, le robot Baxter, grâce au système d'exploitation robotique ROS. Un module de gestion des retours haptiques a été intégré à la plateforme et permet à l'utilisateur d'expérimenter un contrôle bidirectionnel robuste.

En premier lieu, une vue d'ensemble des travaux et des techniques réalisés jusqu'à présent dans le domaine des interactions homme-robot sera présentée. On fournira également des exemples d'outils et d'applications haptiques.

La section suivante détaillera la conception et le développement de l'outil haptique et de la plateforme robotique de contrôle haptique. Une description des caractéristiques du robot Baxter sera également fournie.

Ensuite, le fonctionnement de la plateforme haptique sera évalué : on estimera notamment la précision de l'outil de contrôle grâce à une série d'expérimentations. Finalement, le dernier chapitre fait un bilan du travail effectué, tire les conclusions et expose les travaux futurs.

# CHAPITRE 2

## LES INTERACTIONS HOMME-MACHINE

### 2.1 La robotique et les interactions homme-robot

Les disciplines contemporaines de la robotique sollicitent de plus en plus les robots, notamment les humanoïdes, en raison de leurs meilleures capacités d'intégration dans l'environnement de l'homme. Ces nouvelles tendances de recherches visent avant tout, à faciliter le quotidien des humains. Bien que la robotique reste en majorité concentrée dans le secteur industriel, les robots trouvent progressivement leur place dans des secteurs divers et variés : dans le domaine médical (robot Da Vinci [Sun et Yeung, 2007]), dans le domaine militaire (robot démineur Teodor [Schoor *et al.*, 2012]) et spatial (le robot Spirit [Tunstel *et al.*, 2005]) ou encore dans le domaine des services (robots d'aide aux personnes handicapées, robots aspirateurs, robots tondeuse à gazon, robots ménagers, etc.). Au Japon, pays pionnier et très impliqué dans la robotique, il n'est pas étrange de voir des robots donner des leçons dans les écoles, comme c'est le cas du petit androïde Saya [Hashimoto *et al.*, 2013].

D'après le bilan annuel *World Robotics* dressé suite aux études menées par l'IFR (International Federation of Robotics), l'association constate une nette progression de la robotique de service dans le secteur professionnel (cf. figure 2.1 ci-dessous). Cette même organisation définit un robot de service comme étant un mécanisme capable d'accomplir des tâches utiles pour les humains ou pour un autre mécanisme, en excluant les applications d'automatisations industrielles (norme ISO-Standard 8373). De cette définition ressort une classification possible des robots en fonction de l'application qui leur a été attribuée. D'autre part, en épluchant les récentes statistiques sur la robotique et les pronostiques envisagés par l'IFR, une augmentation graduelle du nombre de robots de service (14% d'augmentation entre 2013 et 2014) est prévue dans le secteur professionnel, avec une tendance à se généraliser chez les particuliers !

Cette croissance s'explique par l'apparition de nouveaux axes de recherches en robotique, visant à développer des applications pour les nouvelles générations de robots, qui évoluent dans un environnement d'intervention humaine. Autrement dit, les études portent sur les moyens d'interactions avec ces robots. Effectivement, n'importe quelle application robotique possède une forme d'interaction, même les systèmes qualifiés de totalement

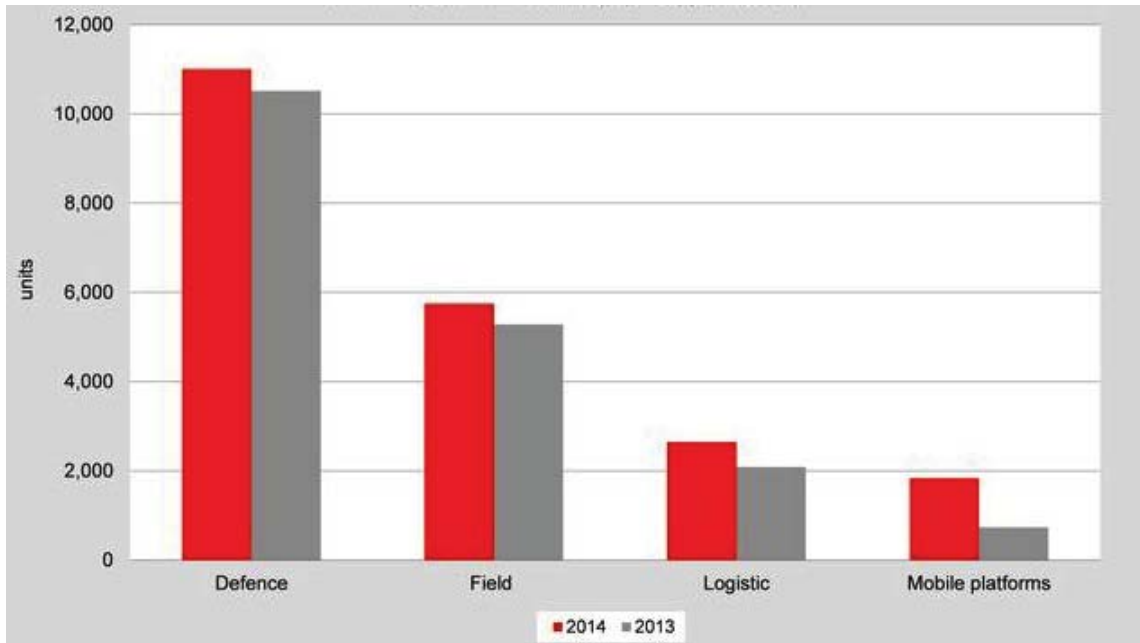


Figure 2.1 Évolution de la robotique de service dans le secteur professionnel [IFR, 2015]

autonomes telle que la voiture automatique de Google. En particulier, la robotique de service est étroitement liée avec un autre secteur de recherche qui s'intitule les interactions homme-robot (IHR). Ce domaine interdisciplinaire à la frontière entre la robotique et la psychologie est dédié à la compréhension, à la conception et à l'évaluation de systèmes robotiques, conçus pour être utilisés par, ou avec des humains. Par définition, les IHR explorent toutes les formes de communication entre l'humain et la machine et cherchent à créer des dispositifs matériels et logiciels permettant à un utilisateur d'interagir avec une machine.

La communication entre un robot et un opérateur humain peut prendre plusieurs formes qui sont très largement influencées par la proximité des deux acteurs. En conséquence, les interactions sont séparées en deux catégories : les interactions de proximité, dans le cas où l'homme et le robot sont rassemblés dans un même espace, alors que les interactions distantes mettent en avant des acteurs séparés dans l'espace, voire dans le temps. L'exploration de la planète Mars par les robots *Mars Rover* de la NASA est catégorisée d'interaction de contrôle à distance alors que les robots de service, comme les aspirateurs automatiques qui peuvent se trouver à proximité d'humains, sont classés dans la première catégorie. Par conséquent, il n'est pas difficile d'imaginer que les enjeux sont bien différents qu'il s'agisse d'une catégorie ou d'une autre : la sécurité sera primordiale pour les communications de proximité alors que la qualité de la communication est privilégiée dans l'autre catégorie.

Ces nouveaux défis technologiques ont permis la naissance d'une nouvelle génération de machines, les robots collaboratifs spécialement conçus pour répondre aux besoins des IHR. La compagnie française RB3D, pionnière dans le milieu, décrit la robotique collaborative ou cobotique, comme une branche multidisciplinaire « *émergente de la technologie à l'interface de la cognitive et du facteur humain, de la biomécanique et de la robotique* ». En effet, pour développer des robots capables de collaborer et de cohabiter avec des hommes, chercheurs et industriels travaillent à développer des algorithmes, étudient l'aspect mécanique, mènent des recherches sur les matériaux et étudient les comportements humains pour les intégrer dans les machines. Ces robots présentent l'avantage d'être facilement programmables et d'être dotés de systèmes de sécurité, ce qui leur permet d'être installés à proximité d'opérateurs humains et non plus dans des cages de sécurité.

Ce domaine est très vaste et fait appel à des compétences en mécanique, électrique et électronique, informatique, mais aussi en sciences sociales (psychologie, cognitive, communication). Parmi les principaux champs d'expertise, on retrouve :

- l'apprentissage automatique (*machine learning*),
- l'ergonomie,
- la téléopération robotique et l'haptique,
- les interfaces homme-machine (IHM),
- la recherche opérationnelle.

Pour ne citer que quelques exemples, l'exosquelette *Rewalk* [ReWalk Robotics, 2014] qui assiste les personnes à mobilité réduite ou l'humanoïde *Robonaut* [Diftler *et al.*, 2003] conçu par la NASA pour aider les astronautes dans l'exploration spatiale sont des projets représentatifs de cette coopération pluridisciplinaire. D'autre part, une interaction sera considérée comme réussite si le robot est capable d'adapter son comportement en fonction de celui de son homologue humain, sans toutefois que la machine puisse prendre le dessus sur l'homme.

Même si les progrès technologiques ont été fulgurants ces dernières années, la coopération homme-machine reste à ce jour, un des défis les plus importants avec des robots toujours plus immergés dans l'environnement humain. En effet, la perception de l'environnement pour la machine est un enjeu clé de cette collaboration. Plus le robot est équipé de capteurs précis ou de caméras performantes, meilleur sera sa perception de son entourage, ce qui lui permettra d'augmenter sa capacité décisionnelle. De cette constatation émerge un premier

enjeu important des IHR, à savoir le coût de conception, très fortement lié à la précision des interfaces des IHR.

Un problème d'IHR consiste dans un premier temps à comprendre la forme des interactions entre un ou plusieurs humains et un ou plusieurs robots. Il convient ensuite de concevoir une technologie adaptée, c'est-à-dire en adéquation avec un besoin, tout en procurant à l'utilisateur une facilité d'utilisation et d'apprentissage, une fiabilité et une efficacité. Le facteur budgétaire est aussi un défi de taille qu'il est indispensable de considérer et qu'il ne faut pas négliger.

En définitive, l'instauration de congrès comme la conférence internationale IEEE RoMan (IEEE International Symposium on Robot & Human Interactive Communication), une approche de la robotique multidisciplinaire ainsi que la compétition entre les différents laboratoires de recherche constituent les principaux facteurs de contribution à l'émergence des interactions homme-robot comme un domaine à part entière de la robotique.

## 2.2 Les interactions homme-robot à distance

À l'origine, les premiers robots faisaient partie des interactions qualifiées de distantes puisqu'ils étaient encore dans des pièces ou dans des cages, sans aucune présence humaine. Au fur et à mesure que la technologie a évolué, les interactions homme-machine se sont adaptées et font aujourd'hui référence à la téléopération, à la télémanipulation, au contrôle de donnée ou encore à la supervision.

La téléopération est un domaine émergent des développements modernes de la robotique et des techniques de communication. En télérobotique, la combinaison d'un opérateur local avec un robot autonome commandé à distance permet d'exploiter de façon optimale l'intelligence humaine ainsi que l'automatisation de la machine. Ces systèmes sont d'autant plus intéressants que complets lorsque des informations transitent entre les deux acteurs à travers divers moyens de communication, ce qui a pour effet d'augmenter les capacités de l'opérateur à mener des travaux de plus en plus complexes.

Lorsqu'on parle de ces nouvelles formes d'interaction, il est souvent fait référence au contrôle à distance de robots standards, mobiles ou humanoïdes afin de remplacer l'homme dans des zones inaccessibles par ce dernier. Par exemple, le département d'aéronautique de l'Université Embry-Riddle [Fletcher *et al.*, 2012] utilise un robot mobile pour des opérations de recherche et de sauvetage.



Les techniques de contrôle évoluent en parallèle des nouveautés et des avancées technologiques, qui ont dépassé depuis longtemps le duo classique formé par la souris et le clavier d'ordinateur, même si celui-ci reste un outil toujours utilisé [Li *et al.*, 2015a]. Les téléphones intelligents qui ont très largement envahi le marché, commencent depuis quelques années à être utilisés pour les interactions homme-robot et se révèlent être de bonnes solutions de contrôle à distance. Avec leurs communications vidéos, leurs reconnaissances et synthèses vocales ainsi que leurs capteurs proprioceptifs (accéléromètre, gyroscope, boussole, système de géolocalisation) ces petits bijoux technologiques ont fait leurs preuves dans le contrôle de robots : dans [Walker et Miller, 2012], on a recours aux aspects proprioceptifs afin de téléopérer un petit robot portatif, en particulier grâce aux données issues des accéléromètres du téléphone. Certaines IHR peuvent faire intervenir plusieurs humains ou plusieurs robots dans une même tâche de coopération ou de contrôle. La mise en place de tels scénarios est plus complexe et nécessite généralement des systèmes décentralisés qui fonctionnent en parallèle. Par exemple, une méthode de reconnaissance basée sur des techniques visuelles, capable de suivre et d'identifier plusieurs humains en temps réel [Nair *et al.*, 2011] est implantée pour faciliter l'interaction d'un robot avec ces homologues humains.

Souvent en robotique, les retours visuels sont utilisés pour prévenir des collisions et assurer une sécurité optimale pour les hommes en présence de machines, ce qui est d'ailleurs effectif dans cette application où le système de vision permet une supervision de l'interaction. La robotique a tout autant recours à des méthodes de communication, sélectionnées en fonction de la portabilité, des caractéristiques des composants et des besoins auxquels doit répondre le système. Dans cette IHR distante, les sous-systèmes communiquent entre eux selon un protocole TCP/IP et le système complet est contrôlable par n'importe quel dispositif équipé d'un Wi-Fi.

Depuis plusieurs décennies, le contrôle par impédance a démontré ses capacités et ses avantages en terme de sécurité et de flexibilité dans la robotique de contrôle. Il s'agit d'une stratégie efficace pour ajuster à la fois la position et la force d'un robot en fonction des besoins d'une application [Modares *et al.*, 2016]. Dans les travaux de [Wang *et al.*, 2014], les données issues de signaux électromyographiques (EMG) sont capturées pendant une interaction entre un robot et son opérateur humain dans le but de mesurer l'activité du muscle squelettique. Une fois cette information acquise, l'amplitude du signal EMG est extraite grâce à des méthodes de décomposition AM-FM. Enfin après linéarisation, une estimation de l'impédance est déterminée et utilisée pour contrôler à distance un bras robotique.

## 2.3 Les interactions homme-robot locales

Les interactions locales sont souvent associées à des interactions physiques homme-robot (pHRI). Elles sont généralement définies par un contact physique qui peut être spontané ou durable, direct ou indirect. Les solutions généralement adoptées pour ces interactions physiques utilisent des interfaces graphiques (GUI) qui permettent à l'utilisateur d'envoyer et de voir des commandes à travers une multitude d'outils comme des écrans tactiles, des joysticks ou encore la souris et le clavier. Ces outils sont très souvent locaux et connectés directement sur le robot, d'où le nom d'interactions locales. Cependant, la robotique moderne recherche de plus en plus des interfaces adaptées, naturelles, faciles à utiliser et, dans la mesure de possible, peu onéreuses. Ainsi, des applications utilisent, par exemple, des contrôles vocaux [Liu et Li, 2010] ou des commandes gestuelles [Koceski et Koceska, 2010] que ça soit dans le biomédicale [Guerrero *et al.*, 2012], l'assistance ou la réhabilitation robotique [Vitiello *et al.*, 2013].

Dans ce domaine de recherche multidisciplinaire, la collaboration entre l'homme et la machine est omniprésente. En d'autres termes, cette complicité est retrouvée à chaque fois que les robots accomplissent des tâches qui nécessitent une interaction directe ou une coordination avec leur homologue humain. De plus, la coexistence est une capacité tout aussi présente dans ce genre d'application où les robots sont amenés à partager leur espace de travail avec d'autres individus, tout en s'assurant d'évoluer en permanence de façon sécuritaire dans un espace partagé.

Le critère de sécurité ne doit pas être négligé bien au contraire, il s'agit d'un élément crucial : c'est la caractéristique la plus importante à implanter sur la génération de robots impliqués dans des actions et dans un environnement à proximité d'êtres humains. Or, les techniques classiques de sécurité concernent les robots industriels et ne sont pas adaptées aux pHRI. En effet, des forces de contact, des mouvements trop brusques ou de mauvaises détections de la part des robots peuvent très rapidement venir perturber une telle coopération. C'est ainsi qu'il a fallu revoir les techniques de sécurité déjà existantes pour les rendre conformes avec ces nouvelles approches de la robotique.

À la différence des interactions à distance, les applications de pHRI sont conçues pour créer des réponses comportementales chez la machine au lieu de chercher à optimiser la commande ou le contrôle. En effet, on souhaite développer une attitude réactive chez le robot pour qu'il puisse s'adapter en temps réel, aux changements de situations et aux imprévus causés par l'intervention humaine. Ce type d'approche tente de doter les robots d'intelligences complexes afin de ne plus considérer les humains comme de simples obstacles à éviter. On peut citer les travaux réalisés par [Pachierotti *et al.*, 2005], qui utilisent des

algorithmes de planification de déplacements robotiques en prenant compte les contraintes de comportement spatial des humains. Qu'il s'agisse de la robotique d'assistance, de la robotique médicale ou de la robotique sociale, l'attitude d'un robot doit représenter au mieux le comportement attendu par le sujet qui l'utilise.

D'autres volets sont tout autant exploités et s'orientent vers des systèmes pratiques, plus appropriés aux interactions entre l'homme et la machine. Parmi eux, on retrouve la reconnaissance d'intentions humaines, particulièrement adaptés aux interactions de contacts. Derrière le travail effectué par [Kim *et al.*, 2010], le toucher qui est important pour l'échange d'informations, est exploité dans un algorithme de reconnaissance de formes de toucher, afin de garantir une réponse réactive du robot, une fois l'algorithme implanté sur celui-ci. D'autres travaux se sont penchés sur la reconnaissance des ambitions humaines, comme les gestes ou les mouvements [Tan et Kawamura, 2015], les images du visage [Yim *et al.*, 2015] ou la position du regard [Wang *et al.*, 2012]. Le but de ces approches est d'essayer de traduire l'expression humaine en langage machine, compréhensible par les robots. En particulier, la reconnaissance vidéo a très largement été étudiée à travers un grand nombre d'applications comme l'amélioration de la surveillance des chutes chez des patients, l'analyse des mouvements et des comportements corporels, notamment pour la pratique sportive.

## 2.4 Les outils des interactions homme-robot

Une grande problématique des IHR réside dans le choix des outils de contrôle. Faut-il privilégier des dispositifs précis généralement coûteux à des appareils plus abordables, mais moins précis ? Est-il judicieux d'opter pour des systèmes intuitifs qui sont faciles à commander plutôt que d'autres plus complexes, mais mieux adaptés à un type d'application ?

Il existe une multitude d'approches bien que la plupart soient calquées sur la façon dont les humains interagissent les uns avec les autres. Tous les jours nous communiquons de façon directe ou indirecte avec les machines, que ce soit graphiquement par le trio classique clavier, souris et écran d'ordinateur, tactilement et vocalement avec nos téléphones intelligents ou encore avec des joysticks sur nos consoles vidéo. L'homme est imprégné depuis plus d'une décennie dans ce contexte technologique qui ne cesse de se développer. Alors qu'il y a une quarantaine d'années se commercialisaient à grande échelle les premières interfaces physiques entre l'homme et la machine avec l'arrivée sur le marché des ordinateurs, de nos jours, on fait face à des systèmes à la pointe de la technologie qui répondent aux besoins grandissants de dispositifs plus ergonomiques, plus efficaces, plus adaptés aux

contraintes des applications. Ces interactions informatiques émergentes sont appelées interface utilisateur naturelle (NUI) et mettent l'accent sur les capacités humaines telles que le toucher, la vision, la voix et le mouvement. Cet ensemble d'outils et de logiciels qui permettent à un utilisateur de communiquer et d'interagir avec un système interactif se subdivise en plusieurs disciplines telles que la communication homme-machine, le dialogue homme-machine ou encore l'interaction personne-machine.

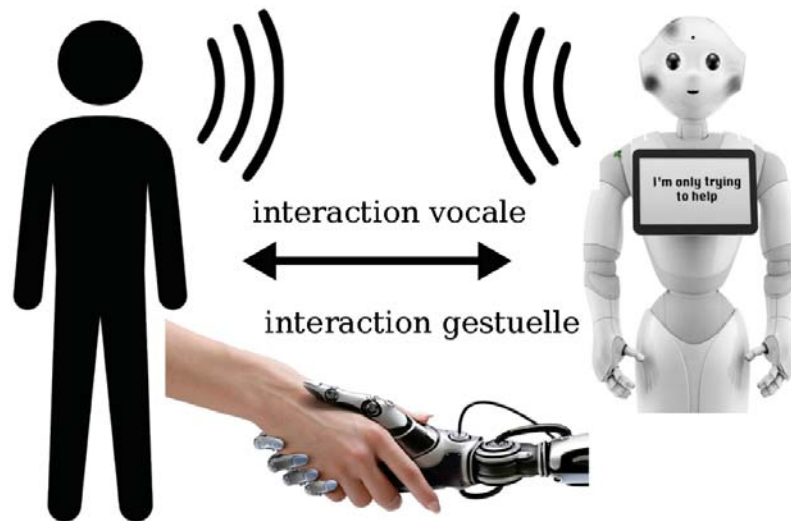


Figure 2.2 Deux exemples d'interaction directe avec un robot : en haut, une communication vocale avec l'humanoïde *Pepper* [Aldebaran Robotics, 2015], en bas, un contact gestuel entre un humain et un robot.

On distingue plusieurs systèmes interactifs qu'il est possible de classer en deux grandes catégories : les outils directs (cf. figure 2.2) sont en grande majorité utilisés pour l'apprentissage dans des activités sociales alors que les outils indirects sont plus souvent utilisés pour le contrôle. Les outils ne sont donc pas les mêmes qu'il s'agisse de la première ou de la deuxième classe. Par exemple, il est possible de dialoguer comme deux humains le feraient, avec le robot humanoïde *iCub*, doté de trois sens que sont la vue, l'ouïe et le toucher [Vernon *et al.*, 2007], [Metta *et al.*, 2008]. Il s'agit là d'une interaction directe puisque l'utilisateur n'est pas obligé de passer par une interface externe. D'autres applications visent davantage un contact physique avec la machine, comme le bras articulé construit dans [Jindai *et al.*, 2006] pour une action de serrer la main. Il y a aussi le dernier robot collaboratif de Rethink Robotics *Sawyer*, qui est capable de reproduire un mouvement qu'un humain lui enseigne [Rethink Robotics, 2015]. Certains systèmes, équipés de caméras ou d'autres capteurs sensoriels, permettent à un robot la détection de visage [Pan *et al.*, 2009] ce qui joue un rôle important dans le développement de robots qualifiés de sociaux, de par leurs interactions avec des humains voire avec d'autres systèmes. Un autre exemple

est celui du petit humanoïde *Nao* [Gouaillier *et al.*, 2009] qui, lorsque équipé d'un système similaire est capable de suivre la trajectoire d'une tête humaine grâce à la détection de visage [Ismail *et al.*, 2011].

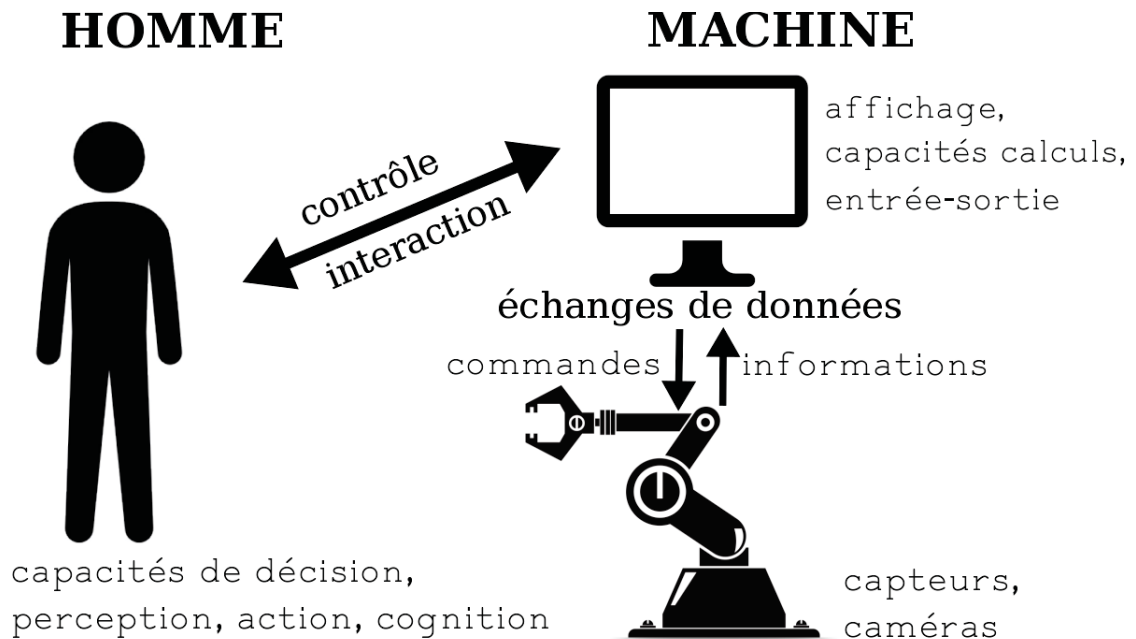


Figure 2.3 Schéma explicite d'une interaction indirecte avec un bras robotique.

En revanche pour les applications de téléopération ou de contrôle à distance d'un robot, il est plus fréquent de trouver une forme de communication indirecte (cf. figure 2.3), c'est-à-dire qui s'effectue par le biais d'un autre système pouvant être un ordinateur, un contrôleur, un FPGA, etc.

L'intensification, l'immersion et la complexité des dernières générations de robots dans notre société ont donné naissance à de nouveaux outils qui sont venus à la rescousse des anciennes interfaces de communication, les GUI, insuffisantes face aux prouesses des nouvelles technologies de la robotique. Effectivement, ce sont les joysticks, les écrans tactiles, les entrées visuelles, les outils haptiques et de réalité virtuelle (VR) qui sont de plus en plus utilisés dans la robotique. Dans [Cheng *et al.*, 2012] ou plus récemment dans [Lin et Chou, 2015], des systèmes permettent de contrôler un robot humanoïde en utilisant le langage corporel grâce à l'outil Kinect de chez Microsoft. En réalité augmentée, l'*Oculus Rift* est utilisé pour la téléopération d'une caméra embarquée sur un robot mobile [Krückel *et al.*, 2015]. Par un simple mouvement de la tête, l'utilisateur qui porte les lunettes virtuelles est capable de déplacer la caméra et donc d'adapter son champ de vision en temps réel. Les études en téléopération à distance cherchent en permanence à améliorer l'agilité et la maniabilité du contrôle avec notamment des interfaces équipées de rétroactions tactiles

[Sarakoglou *et al.*, 2012] ou visuelles [Willaert *et al.*, 2012], qui fournissent à l'opérateur un contrôle beaucoup plus intuitif qu'une interface graphique.

## 2.5 Les dispositifs haptiques et leurs applications

L'haptique ou logique de la sensation, fait référence à la science du toucher et du tactile. Elle est plus généralement associée aux phénomènes tactilo-kinesthésiques, c'est-à-dire la perception consciente de la position ou des mouvements des différentes parties du corps dans l'environnement. L'haptique diffère du tactile du fait qu'il est rattaché aux applications d'interaction et de contrôle assistées par ordinateur. Cette technologie est notamment utilisée massivement en réalité virtuelle puisqu'elle offre une dimension supplémentaire aux environnements virtuels, plongeant ainsi l'utilisateur dans un milieu encore plus réaliste. Dans la téléphonie mobile, elle est omniprésente et peut accompagner l'appui d'un son ou faire vibrer le téléphone, donnant l'impression d'appuyer réellement sur une touche physique : il s'agit du retour haptique qui concerne à la fois le toucher et les forces qui s'y développent.

Les interfaces haptiques et les outils associés ont été conçus pour venir compléter les informations sur les sens humains (la vision et l'audition), qu'on pouvait préalablement obtenir avec les caméras et les appareils auditifs. Il n'est pas difficile d'imaginer que dans un contexte d'interaction homme-robot, ces outils ont rapidement trouvé leur place. Ce



(a) Le gant à retour de force *Gloveone VR* [Neurodigital Technologies, 2015]

(b) Le joystick haptique *Geomagic Touch X* [3DSYSTEMS, 2013]

Figure 2.4 Exemples d'outils haptiques

sont des périphériques E/S qui permettent d'envoyer des commandes et de recevoir des informations, qui se traduisent par une sensation physique réelle sur l'outil. Par exemple,

certaines interfaces permettent à un utilisateur d'expérimenter un contact physique dans un espace virtuel grâce à des outils tels que des gants à retour de force comme le *data glove* (cf. figure 2.4a) ou des joysticks (cf. figure 2.4b). Il est possible d'imaginer un scénario dans lequel un utilisateur porte un *data glove* qu'il peut observer visuellement dans un environnement virtuel où se trouve également une sphère : s'il déplace sa main jusqu'à observer un contact virtuellement avec l'objet, une force ou une vibration se fera sentir sur le *data glove*, donnant ainsi une réelle sensation de contact. Le gant sensoriel est également utilisé dans des applications de suivi de mouvement [Kim *et al.*, 2009], ou bien de téléopération de bras robotiques équipés de mains [Fang *et al.*, 2015], dans le but de rendre les interactions les plus naturelles possible.

Les recherches sur l'haptique ont avant tout émergé avec l'industrie du divertissement, tout particulièrement avec les jeux vidéo fortement liés à la VR. En outre, la technologie des retours de forces a très fortement amélioré l'aspect physique des jeux en créant une immersion beaucoup plus profonde tout en perfectionnant les compétences du joueur. Cette technologie est très implantée dans les jeux FPS, même si on la retrouve un peu partout comme dans [Paolis *et al.*, 2007] où une interface haptique permet de jouer au billard. Les jeux vidéos ont aussi permis la naissance de nouveaux contrôleurs haptiques : c'est le cas d'un joystick qui utilise un mécanisme à cardan, deux moteurs continus pour les retours de force et des câbles pour les transmissions [Bae *et al.*, 2001]. Hormis le côté purement récréatif que l'on retrouve dans une grande majorité des jeux vidéos, une nouvelle tendance vise à combiner divertissement et apprentissage. Ces interfaces qu'on peut qualifier d'éducatives donnent accès à de l'information à travers un environnement de réalité virtuelle, comme c'est le cas du jeu éducatif *Haptibasic* [Albraikan *et al.*, 2013]. Ce jeu montre qu'en intégrant les retours haptiques avec des jeux de mémorisation, l'habilité du joueur à trouver des objets similaires est accrue. Une autre application du même genre simule un lance-pierre pour permettre aux utilisateurs d'interagir avec les lois de la physique [Kretz *et al.*, 2005]. L'étude a montré que l'utilisation d'un retour de force aide l'utilisateur à se créer un modèle mental afin de comprendre les lois de la physique.

En chirurgie, les systèmes de télérobotique chirurgicale sont très utilisés puisqu'ils améliorent les capacités des chirurgiens grâce à leur grande précision, leur visualisation 3D ainsi qu'une panoplie d'autres instruments que l'on peut venir intégrer sur ces plateformes multifonctionnelles. L'implantation d'outils haptiques dans de tels systèmes renforce la perception tactilo-kinesthésique et vient améliorer les performances d'une opération chirurgicale de différentes manières : par exemple, un chirurgien peut prévenir des traumatismes non désirés ou des incisions non souhaitées sur les tissus humains [Greenwald *et al.*,

2012] et peut également estimer les propriétés mécaniques de ces mêmes tissus (texture ou raideur) [Okamura *et al.*, 2011]. En outre, ces dernières années, la médecine a été révolutionnée par l'utilisation de ces instruments comme un moyen d'entraînement sous simulation [Esteban *et al.*, 2014] et des outils haptiques spécifiques à cet apprentissage ont vu le jour [Mortimer *et al.*, 2014]. On trouve également des systèmes de réhabilitation ou de rééducation faisant appel à l'haptique pour renforcer des organes et muscles blessés ou accidentés [Estes *et al.*, 2015], [Morito *et al.*, 2013]. Les possibilités semblent infinies lorsqu'on combine l'haptique et la réalité virtuelle : une interface haptique à la frontière entre la robotique et la VR facilite une réhabilitation des mains pour des patients souffrant de déficits neurologiques moteurs [Sivak *et al.*, 2010].

En robotique, Pierre Renon et son équipe ont implémenté une interface haptique entre un joystick 3D et un simulateur virtuel de robot [Renon *et al.*, 2013]. L'équipe a fait le choix de s'orienter vers le simulateur open source du petit humanoïde iCub avec son architecture qui fournit une application temps réel tout en simplifiant l'interfaçage avec d'autres dispositifs. Le joystick haptique est le Falcon de Novint Technologies, une facilité d'intégration et un bon rapport qualité-prix. Deux approches ont été comparées : l'une fait appel à une librairie haptique open source CHAI3D et l'autre utilise la *toolbox Haptik* de MATLAB. Des bibliothèques temps réels sont créées afin d'analyser les données issues du joystick (position cartésienne et valeur des boutons) avant de les envoyer vers le simulateur où elles sont analysées et traitées afin de bouger le bras du simulateur iCub. Une boucle de rétroaction faisant le chemin inverse, transmet les informations recueillies par le simulateur jusqu'au Falcon, sous forme de couples appliqués aux moteurs ; ainsi l'utilisateur obtient un retour sur ces commandes. De manière similaire, la *toolbox Haptik* fournit un accès à l'outil haptique, puis utilise l'architecture de iCub pour interagir avec le simulateur. Le schéma de communication entre les différents blocs du projet est représenté sur la figure 2.5 ci-dessous.

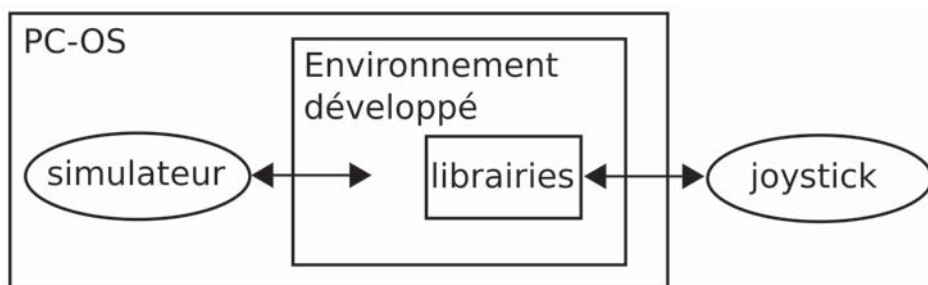


Figure 2.5 Schéma de la communication entre le simulateur iCub, l'interface haptique et le joystick.



L'haptique présente l'avantage d'être applicable à un grand nombre de domaines, qui ne sont pas forcément rattachés à la robotique, mais est trop souvent freiné par les restrictions budgétaires. Ainsi, des recherches récentes portent sur des outils haptiques bon marché, peu nombreux au jour d'aujourd'hui. Le coût d'un outil est directement lié à sa précision et donc à son nombre de degrés de liberté. Cette notion, qui porte l'acronyme DDL, désigne la capacité d'un système à se mouvoir selon un axe de translation ou de rotation. En comptant chacun des axes, on obtient le nombre de degrés de liberté total du système i.e le type de mouvement réalisable, ce qui renseigne à la fois sur son potentiel et sa complexité : plus il y a de mouvements possibles dans l'espace, plus le prix est élevé. Cette notion requiert six paramètres indépendants pour décrire un mouvement dans l'espace trois dimensions : trois paramètres qui définissent la position d'un point du solide et trois autres déterminent l'orientation du solide par rapport à ce même point.

Un contrôleur haptique comme le PHANTOM Omni, considéré comme un des plus rentables sur le marché, affiche un prix supérieur à \$10 000, auquel il faut ajouter un coût supplémentaire lié au développement d'une interface capable de s'intégrer à différents systèmes. Ce dispositif permet de réaliser l'ensemble des mouvements dans l'espace trois dimensions, c'est-à-dire trois translations selon les axes x, y et z ainsi que trois rotations autour de ces axes. Néanmoins, le développement de robots peu onéreux comme Ono [Vandeveld *et al.*, 2013], reproductible pour 300€, pousse les chercheurs à concevoir des plateformes robotiques à coût réduit. Par exemple, un projet de rame haptique [Rose *et al.*, 2014] a été conçu, mais ne possède qu'un seul axe de rétroaction d'où un coût de fabrication réduit. Cependant il est resté au stade d'outil d'apprentissage dans les universités. De même, OSHap [Lavatelli *et al.*, 2014] est une interface haptique créée et licenciée librement qui utilise un outil à deux degrés de liberté. L'outil est construit avec seulement deux articulations afin de réduire le coût lié à sa conception tout en étant suffisamment modulable pour laisser la possibilité à l'utilisateur d'ajuster les dimensions de l'espace de travail à sa guise. Un dernier exemple serait celui exposé dans [D'Auria *et al.*, 2015] qui propose un système bon marché, basé sur une interface haptique et un gant sensoriel pour la réadaptation du poignet chez les patients. Le gant haptique permet de détecter l'orientation du poignet ce qui donne la possibilité au patient d'être guidé et formé pour réaliser certains mouvements utiles à sa rééducation.

## 2.6 Contexte et enjeux des interactions homme-robot

L'évolution des technologies de la science électronique, mécanique et informatique a permis à la robotique d'évoluer encore plus rapidement ces dernières années. Dans les précédentes sections, les nouvelles tendances de ce domaine ont été exposées et il ne fait aucun doute que les interactions homme-robot vont prendre de plus en plus d'importance avec des robots de plus en plus présents autour de nous. Une conséquence directe de cette évolution réside dans l'adaptation conceptuelle de la fabrication des robots. Leurs aspects et leurs formes se transforment afin d'y intégrer les dernières technologies (capteurs, caméras, etc...) tout en essayant de faciliter leur acceptation parmi les humains. Dans l'industrie par exemple, une nouvelle vision se dessine depuis quelques années : une industrie plus flexible, plus abordable, plus performante fondée sur la robotique collaborative ou cobotique. Ce secteur de recherche robotique entreprend de concevoir des outils pratiques pour augmenter la productivité des robots et travaille notamment à développer des robots qui assistent ou guide l'homme pour une tâche spécifique. Alors qu'un robot standard est conçu pour être programmé et pour fonctionner plus ou moins de façon autonome, les cobots sont présentés comme des assistants qui nécessitent une collaboration et non plus comme de simples exécutants. Contrairement aux applications traditionnelles où les robots industriels étaient généralement séparés des humains et n'étaient contrôlables que par des opérateurs aguerris, les récentes plateformes d'interactions peuvent être équipées de capacités spécifiques qui permettent aux robots d'adopter des comportements sociaux, leur permettant de travailler avec des opérateurs non experts dans des environnements non industriels. Les défis scientifiques et technologiques sont de taille et la route est longue avant d'obtenir des interfaces sécuritaires, robustes et fiables qui puissent permettre une communication mutuelle et une adaptation entre l'homme et le robot de façon intelligente et naturelle. La conception et l'évaluation des interfaces sont un autre défi que les ingénieurs doivent se poser afin de déterminer quel concept est le plus adapté pour assister un opérateur humain dans une tâche spécifique.

Depuis la conception des robots collaboratifs (cobots, exosquelettes et humanoïdes) l'interaction homme-robot s'articule autour de plusieurs axes technologiques. Les chercheurs sont de plus en plus confrontés à la problématique de la collaboration et de la cohabitation homme-robot qui nécessite une adaptation des techniques d'interaction et de contrôle : on cherche à développer des applications, des interfaces et des outils qui permettent de communiquer, d'interagir et de coopérer le plus naturellement possible et de façon sécuritaire avec ces nouveaux "acteurs" de notre société. Alors qu'avant, les robots n'avaient pas ou très peu de moyens de perception de l'environnement, aujourd'hui ils sont dotés de cap-

teurs de force et d'effort de couple pour détecter ce qui se passe autour d'eux. Ceci met en évidence un des défis de la robotique moderne, un défi avant tout financier : la section précédente faisait état des coûts liés à la précision des capteurs et bien souvent les capteurs utilisés dans les robots industriels sont trop chers et viennent s'ajouter aux autres coûts relatifs à la mécanique et l'électronique du robot. Pour ne citer que quelques exemples, Nao [Gouaillier *et al.*, 2009] qui fait partie des robots les moins chers affiche un prix de 12000€, l'HOAP-3 de Fujitsu, coûte quant à lui pas loin de \$70 000US et la 3<sup>eme</sup> version d'Asimo [Sakagami *et al.*, 2002a] n'est disponible qu'en location pour un coût avoisinant les \$170 000US par année. À l'heure actuelle, un robot performant et compact vaut encore trop cher, mais de nouvelles études se penchent sur cette problématique qui soulève les interrogations suivantes : comment est-il possible d'assembler les éléments constitutifs d'un robot, de manière robuste, précise et rentable tout en conservant les mêmes caractéristiques ? Une autre question découle directement de celle-ci : quels sont les outils de contrôle à privilégier pour diminuer davantage les prix des interfaces liées au robot ? La réduction du coût lié à la robotique est difficile à appréhender pour la simple raison que les machines commencent tout juste à être commercialisées à grande échelle. Il faut donc espérer que la croissance se poursuive selon les expertises et les statistiques actuelles, afin que le prix d'achat et la maintenance liée aux robots diminuent. Or, la concurrence des nouvelles entreprises qui se lancent dans le développement et la fabrication de robots, va jouer en faveur d'une robotique meilleure marché.



# CHAPITRE 3

## UNE INTERFACE UTILISATEUR NATURELLE

### 3.1 Le joystick haptique Falcon

Le choix de cet appareil ne fait pas l'objet de cette recherche, mais s'intègre parfaitement à ces objectifs. Comme nous en avons déjà discuté, cette technologie permet un ressenti tactile auprès de l'utilisateur, recréant la sensation de toucher, en appliquant forces, vibrations ou même en actionnant certains moteurs afin de reconstituer un mouvement à partir de l'outil. Ces outils furent initialement créés pour l'industrie du jeu vidéo en exploitant particulièrement les performances de leurs perceptions tactilo-kinesthésiques, mais restent très utilisés dans le domaine de la robotique. C'est en 2007, lors de la *Consumer Electronics Show* que la sortie du Falcon (c.f figure 3.1) de la société Novint Technologies est annoncée. Cet appareil est présenté comme une alternative à la conventionnelle souris dans l'industrie du jeu vidéo, fournissant ainsi au joueur une expérience toujours plus immersive et une communication plus réaliste. Il est présenté comme le premier outil tactile à trois dimensions (3 DDL) accessible au grand public qui permet à l'utilisateur de ressentir le sens du toucher en informatique.



Figure 3.1 Le joystick haptique Falcon avec différents effecteurs : à gauche il s'agit de l'effecteur standard qui vient à l'achat de l'appareil, à droite, le poignet pistolet pour les jeux vidéo

Dans le domaine de la recherche, cet appareil a très rapidement trouvé sa place grâce à un kit de développement logiciel (KDL) *Libnifalcon* (c.f 3.1.4) fourni lors de l'achat de

l'outil. Il s'agit d'une solide plateforme de programmation qui donne la possibilité aux programmeurs de développer des applications haptiques spécifiques de différentes complexités. D'un point de vue tactile, cet appareil offre un mouvement fluide et continu en comparaison à d'autres dispositifs tactiles comme le *Phantom Omni* de la société *Sensable Technologies*, au moins cinquante fois plus cher que le Falcon : on parle de \$200 pour le premier et de plus de \$10 000 pour le second. Effectivement dans [Vanacken *et al.*, 2010] les deux outils sont comparés dans une tâche de suivi de trajectoire et les conclusions montrent que les différences, en terme de performance, sont faibles et non significatives.

Comme beaucoup d'outils haptiques, il présente un espace de travail relativement réduit ( $10\text{cm}^3$ ) dû à son nombre de degrés de liberté limité aux trois axes de translation. En revanche, sa conception lui procure une assez grande polyvalence, notamment grâce à un poignet détachable et remplaçable : la société a par ailleurs construit un poignet semblable à un pistolet (cf. figure 3.1) qu'on connecte à la place du poignet d'origine et qui permet principalement de jouer à des jeux FPS. Bien que très largement utilisé dans le domaine des jeux vidéo, une autre division beaucoup plus professionnelle a su utiliser le Falcon pour répondre aux besoins des différents axes de recherches : on le retrouve dans des applications robotiques où il sert, par exemple, de contrôleur pour la téléopération de robots chirurgicaux [Gambaro *et al.*, 2014], [Li *et al.*, 2015b].

### 3.1.1 Structure articulaire

La modélisation d'un robot, considéré comme étant un système mécanique articulé, actionné et commandé, consiste à en établir un modèle mathématique. Or pour développer ce modèle, il faut en premier lieu étudier sa structure mécanique. Celle du Falcon a été introduite par Tsai et Stamper dans un rapport technique sur le manipulateur *Maryland* [Stamper, 1997]. Il s'agit d'un robot de type parallèle ayant une structure delta qui contient une plateforme mobile connectée à une base fixe par trois chaînes cinématiques parallèles. Chacune de ces chaînes est composée de deux liens et trois joints rotatifs. Cette structure est assez particulière, car elle contient un seul joint actif, commandé en couple par un moteur attaché au niveau de la base fixe de la structure. Les autres joints de la chaîne restent passifs et sont influencés uniquement par le mouvement du joint actif. Ainsi, lorsqu'une commande est envoyée par un utilisateur, les trois moteurs enclenchent les câbles qui permettent de mettre en mouvement les trois bras (i.e chaînes cinématiques) et par la même occasion la plateforme mobile, tout en assurant une friction minimale. Cette structure articulaire est représentée figure 3.2. Par conséquent, un seul capteur de position angulaire est nécessaire par chaîne cinématique et mesure donc le joint actif qui fait la liaison entre

la base fixe et le 1<sup>er</sup> lien du robot. Cette architecture a l'avantage d'avoir des solutions de formes fermées, une position et une orientation découplées et une construction qui n'utilise que des joints rotatifs ce qui a tendance à diminuer le coût de fabrication. L'organe terminal ou effecteur est attaché sur la plateforme mobile qui peut être déplacée selon les trois axes de translation  $(x, y, z)$ , définis par un repère de référence lui-même placé au centre de la base fixe. Autrement pas de rotation possible pour cette structure. La caractérisation du Falcon [Martin et Hillier, 2009] ainsi que les informations fournies par le constructeur ont permis d'identifier les principaux éléments constitutifs de ce joystick. Trois moteurs Mabuchi à courant continu (modèle RS-555PH-15280) sont utilisés pour le contrôle des bras. La communication USB se fait via une puce FTDI à interface USB-UART (modèle FT232R) et le traitement numérique du signal ainsi que le contrôle du Falcon se fait par le biais d'un microcontrôleur DSP de Texas Instrument (modèle TMS320RMS2811).

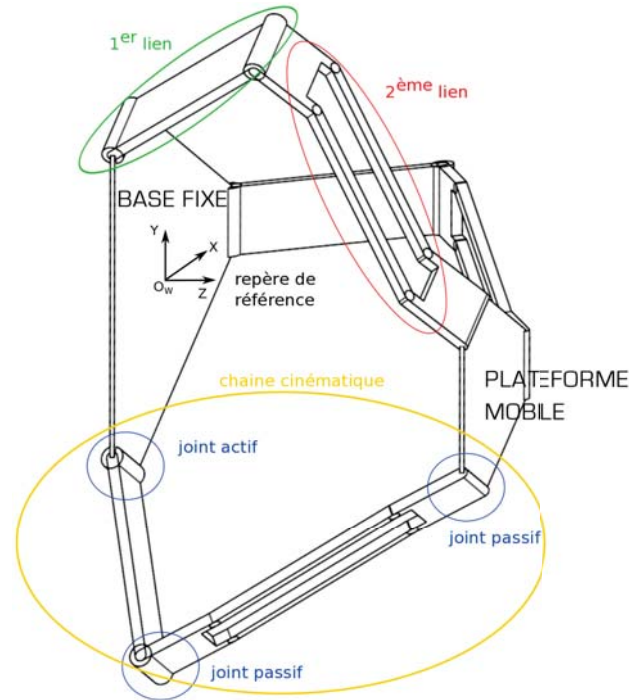


Figure 3.2 Structure mécanique du Novint Falcon [Michelotti, 2011]

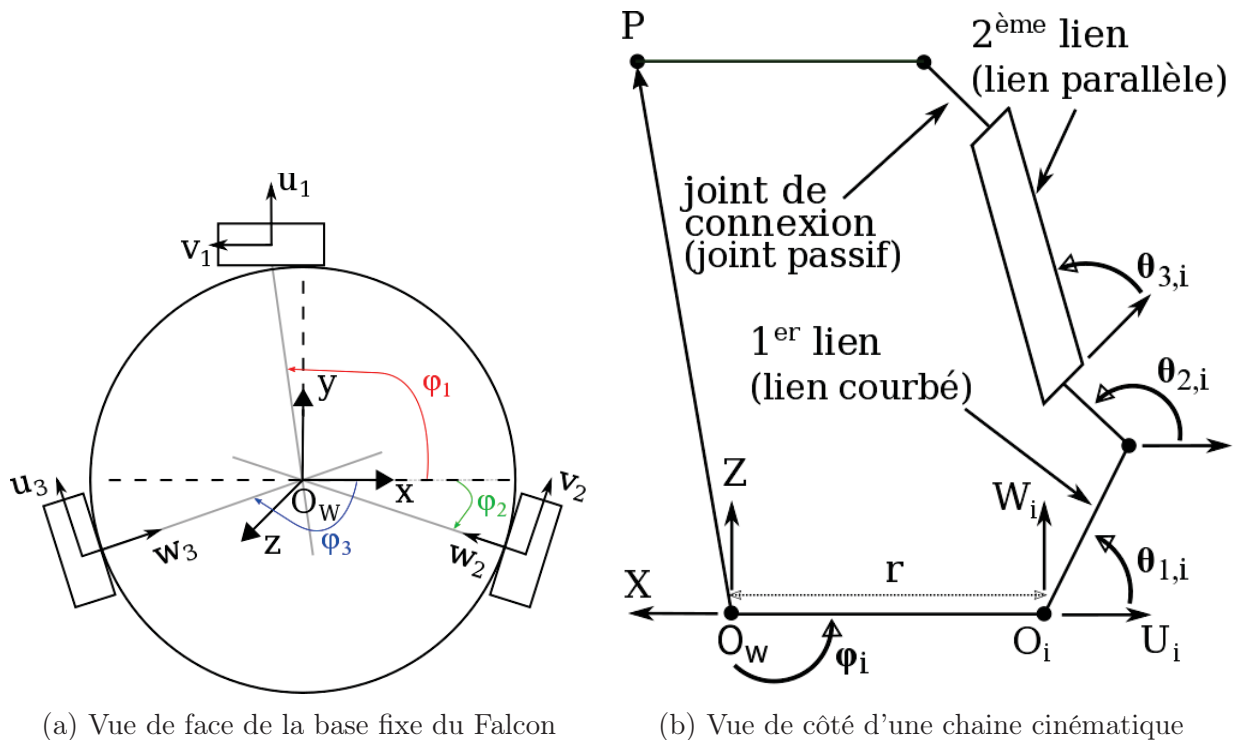
### 3.1.2 Cinématique inverse

De manière générale, la cinématique tente de décrire le mouvement d'un corps sans faire référence aux notions de forces ou d'énergies. Les mouvements sont étudiés à l'aide des notions d'espace et de temps, en faisant abstraction de leurs causes. La cinématique utilise les concepts de position, de déplacement, de vitesse et d'accélération. La cinématique inverse, quant à elle, cherche à calculer les positions et rotations d'un modèle articulaire afin d'obtenir une pose désirée. Dans notre cas, il faut caractériser uniquement la position (on rappelle qu'il n'y a pas de rotation possible pour le Falcon) de la plateforme mobile, définie dans l'espace cartésien et l'exprimer dans l'espace des joints. Or, Tsai a montré, dans son travail [Stamper, 1997], qu'il était possible de considérer une seule des trois chaînes cinématiques pour la résolution de cette problématique. Il suffit ensuite de répéter la même action pour les autres chaînes.

La première étape consiste à exprimer la position de l'effecteur du Falcon (i.e le centre de la plateforme mobile) qu'on notera  $P(x, y, z)$ , dans le système de coordonnées  $UVW_i$  correspondant à un des trois bras du robot. Pour que les équations soient générales, on note  $i$  le  $i^{eme}$  bras du robot. Cette transformation peut être représentée par une matrice homogène qui tient compte de la rotation et de la translation entre les deux repères :

$$H = \begin{pmatrix} \cos(\varphi_i) & \sin(\varphi_i) & 0 & -r \\ -\sin(\varphi_i) & \cos(\varphi_i) & 0 & s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.1)$$

Cette matrice exprime un simple changement de repère : on exprime  $P(x, y, z)$  non plus dans le repère de référence  $O_W$  mais dans le repère  $UVW_i$ . L'angle  $\varphi_i$  représente la différence angulaire (selon l'axe  $z$ ) entre les deux systèmes de coordonnées et la dernière colonne de  $H$  correspond au vecteur  $(-r, s, 0)^T$  qui montre l'offset entre ces deux mêmes systèmes, selon  $(x, y, z)^T$ . Ces données sont visibles sur la figure 3.3a où l'axe central correspond au repère de référence  $O_{world}$  du Falcon, tandis que les repères  $UVW_i$  sont ceux à la base des trois bras de la structure. Comme la structure du Falcon est composée uniquement de joints rotatifs, pour la contrôler la cinématique inverse cherche à obtenir des relations



(a) Vue de face de la base fixe du Falcon

(b) Vue de côté d'une chaîne cinématique

Figure 3.3 Schémas de la structure articulée du Falcon pour le calcul de la cinématique



angulaires. Il faut alors déterminer les expressions des trois angles  $\theta_{1i}$ ,  $\theta_{2i}$  et  $\theta_{3i}$  du  $i^{eme}$  bras. Pour une meilleure compréhension, ces angles sont représentés sur la figure 3.3b.  $\theta_{1i}$  correspond à l'angle actif de la chaîne alors que  $\theta_{2i}$  et  $\theta_{3i}$  sont ceux passifs. Grâce à la transformation H et aux démonstrations dans [Stamper, 1997] on obtient les relations suivantes :

$$\theta_{1i} = 2 \tan^{-1}(t_{1i}) \quad (3.2)$$

$$\theta_{2i} = \cos^{-1}\left(\frac{p_{ui} - a \cos(\theta_{1i}) + c}{d + e + b \sin(\theta_{3i})}\right) \quad (3.3)$$

$$\theta_{3i} = \cos^{-1}\left(\frac{p_{vi} - s}{b}\right) \quad (3.4)$$

où  $p_{ui}$  et  $p_{vi}$  correspondent aux positions du point P dans le repère  $UVW_i$ .  $a$ ,  $b$ ,  $c$ ,  $d$  et  $e$  sont les dimensions physiques des liens, joints et des connecteurs de la structure du Falcon. La variable  $s$  désigne l'offset selon l'axe  $y$  entre le repère  $O_W$  et  $UVW_i$ . Finalement  $t_{1i}$  représente les relations trigonométriques de la chaîne cinématique. Si le lecteur désire des détails supplémentaires, il peut se référer au rapport technique de Tsai et Stamper [Stamper, 1997].

### 3.1.3 Cinématique directe

Le calcul de la cinématique directe est essentiellement le problème inverse de la cinématique inverse. Il consiste à déterminer la position de l'organe terminal connaissant les angles de chacun des joints. Contrairement aux manipulateurs classiques, les robots parallèles possèdent une cinématique directe beaucoup plus complexe à résoudre que celui de la cinématique inverse. Cependant, il s'agit souvent d'une forme très utile surtout si l'outil est utilisé comme périphérique d'entrée. Il ne faut donc pas la négliger. Ici, le calcul de la cinématique directe combine la méthode standard de Newton-Raphson pour les systèmes linéaires avec la matrice Jacobienne.

La matrice Jacobienne est très utilisée en robotique et constitue un outil très pratique pour la caractérisation d'un robot puisqu'elle permet de déterminer les configurations singulières (configurations à éviter), d'analyser la redondance et d'exprimer la relation entre les forces et les moments appliqués au niveau de l'organe terminal. Elle exprime l'état actuel d'une structure robotique par la transformation entre la vitesse de l'effecteur dans l'espace cartésien, et la vitesse angulaire des joints. Elle est exprimée par le terme  $J(q)$  dans l'équation générale ci-dessous :

$$v_e = [\dot{p}_e \ \omega_e]^T = \dot{p}_e = J(q)\dot{q} \quad (3.5)$$

telles que  $v_e(q) = \dot{p}_e = [\dot{x} \ \dot{y} \ \dot{z}]^T$  représente la vitesse linéaire de l'organe terminal (pas de vitesse angulaire  $\omega_e = 0$ , car il n'y a pas de rotation possible) et  $\dot{q} = [\dot{\theta}_{1i} \ \dot{\theta}_{2i} \ \dot{\theta}_{3i}]^T$ , la vitesse angulaire des joints.

L'approche de la cinématique directe utilise une méthode optimisée par rapport à celle décrite par Tsai [Stamper, 1997]. Le détail de cette méthode n'est pas indispensable pour le travail réalisé dans cette recherche. Si toutefois le lecteur désire plus d'informations, il peut se référer à [Ouellet, 2008] et [Zsombor-Murray, 2009].

### 3.1.4 Libnifalcon

Un kit de développement logiciel est fourni avec l'outil haptique de Novint Technologies. Une fois connecté à un ordinateur via un port USB, le Falcon est pris en charge par un ensemble de fonctions et de driver qui permettent la communication avec le micrologiciel installé sur l'appareil. Toutefois ce KDL ne fonctionne que sur les systèmes d'exploitation Windows, mais grâce à une distribution massive du contrôleur, notamment dans l'industrie des jeux vidéo, une alternative a été conçue. Il s'agit d'une multi-plateforme distribuée avec un code source ouvert, nommé *Libnifalcon* [Machulis, 2007] qui fournit un ensemble de fonctionnalités pour accéder, communiquer et contrôler l'appareil. On retrouve toutefois la majorité des possibilités qu'offre le KDL fourni avec le Falcon, et parmi les principales fonctions figure la possibilité de connecter l'appareil en USB, de charger le micrologiciel sur le microcontrôleur interne de l'outil ou encore de récupérer la position de l'effecteur grâce aux capteurs.

Cette librairie C++ sous la licence BSD, se veut aussi flexible que possible en intégrant la cinématique de l'outil afin de permettre à l'utilisateur de s'en servir à la fois comme un périphérique d'entrée, mais aussi comme un périphérique actif, c'est-à-dire qui reçoit un retour de force/couple sur ces moteurs. L'ensemble des fonctions incorpore une gestion des différents poignets qu'il est possible d'installer sur l'effecteur, mais aussi un ensemble d'applications afin de tester le bon fonctionnement du joystick. L'utilisateur peut alors bénéficier pleinement de tous les avantages qu'apporte cet appareil, mais l'on notera tout de même une légère différence entre les performances de cette librairie et le KDL. En effet, dans [Martin et Hillier, 2009] l'auteur remarque qu'en théorie l'interface devrait fonctionner à une fréquence de 1kHz alors qu'en pratique elle ne reçoit pas toutes les commandes envoyées lorsque la fréquence est comprise entre 800Hz et 1kHz. De plus, un retard relativement notable entre les commandes en force issues de l'ordinateur et les changements enregistrés par les capteurs sur les encodeurs est aussi constaté sur l'interface.

Il faudra donc en tenir compte et fixer une fréquence inférieure ou égale à 800Hz pour éviter de faire face à ce phénomène.

Une fois que le Falcon est connecté à un ordinateur, il faut lancer manuellement le driver qui s'occupe en premier lieu d'ouvrir la communication avec le microcontrôleur, puis de charger le micrologiciel. L'utilisateur doit ensuite déplacer le poignet d'une extrémité à l'autre afin de calibrer l'outil et permettre au micrologiciel de passer en mode *home* (c.f figure 3.4). Ce mode lance les encodeurs, ce qui donne la possibilité à l'utilisateur de procéder à l'envoi des commandes. On notera que lorsqu'on initialise pour la première fois l'outil i.e après chaque démarrage de l'ordinateur, il faut relancer le programme une seconde fois pour que le micrologiciel soit chargé correctement et que la communication soit alors possible.

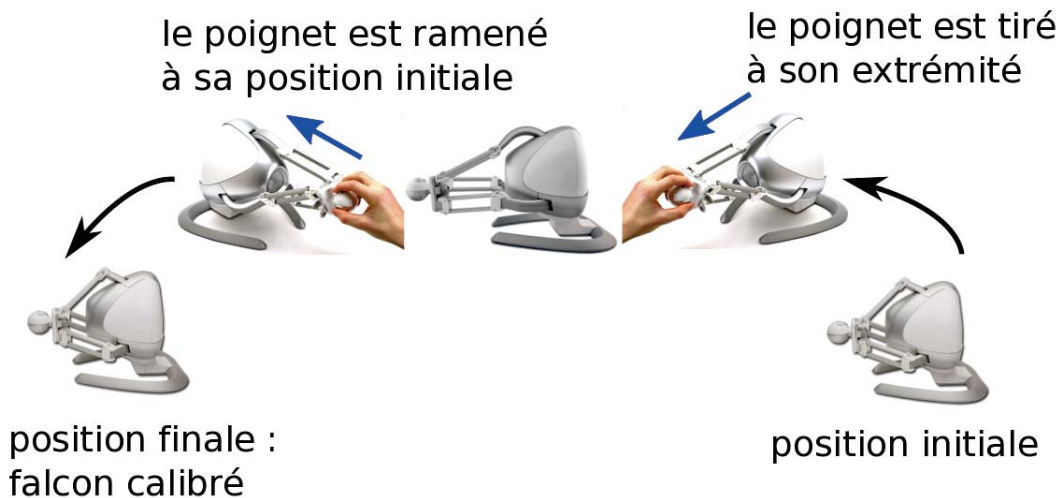


Figure 3.4 Description du processus de calibration du Falcon avant qu'un utilisateur puisse l'utiliser

Cette librairie open source servira de base au développement de la plateforme robotique pour le contrôle haptique. Certaines fonctions ont été étudiées, puis modifiées pour être adaptées aux besoins du projet. Une fonctionnalité additionnelle a été développée et implantée dans ce travail de recherche, elle permet notamment de détecter plusieurs dispositifs de type Falcon connectés à un ordinateur et cela toujours dans le même ordre.

## 3.2 Le robot collaboratif Baxter

Le robot mis à disposition pour effectuer les manipulations avec l'outil haptique est un type de robot qu'on appelle *cobot* conçu par la société Rethink Robotics<sup>1</sup> et disponible pour la somme de 22 000\$. Son créateur, Rodney Brooks, a conçu ce robot collaboratif dans l'optique de redéfinir la façon dont les robots sont utilisés dans l'industrie manufacturière en mettant l'accent sur deux points majeurs : lui procurer une apparence humaine tout en maintenant un prix très compétitif pour que toutes les usines s'en procurent un.

Il est constitué d'un tronc avec deux bras fixés sur un piédestal mobile et d'un écran LCD en guise de tête. Chaque bras forme une chaîne cinématique à sept degrés de liberté avec pour chaque joint un actionneur série élastique, différent des actionneurs classiques. Effectivement un ressort est ajouté entre le moteur (le système d'engrenage) et la sortie de l'actionneur. Cette technique développée par Matthew Williamson [Williamson, 1995] tend à diminuer le bruit lié à la force de contrôle, à mieux protéger les contacts avec d'autres éléments et rend possible une mesure du couple appliqué sur chaque actionneur, ce qui introduit la possibilité d'un contrôle en couple et en impédance.

Chacun des bras de Baxter possède un nombre de joints supérieur au nombre de translations et rotations possible dans l'espace à trois dimensions, afin de lui procurer une redondance cinématique qui vient renforcer sa maniabilité et sa sécurité. Ce choix est souvent adopté par les producteurs de robots, car il permet d'éviter plus facilement les obstacles.

Toujours dans l'idée de perfectionner la sécurité dans l'espace de travail partagé avec Baxter, un modèle de collision en trois différents niveaux est intégré sur le robot :

- chaque lien à son propre bloc de collision, légèrement plus volumineux que leur taille réelle,
- les impacts et les écrasements sont identifiés par le second modèle de détection (changement soudain du couple sur un des joints),
- l'évitement de collision entre les deux bras est détecté par le dernier niveau du modèle.

De plus, Baxter est équipé de deux caméras situées aux extrémités des bras, plus une au niveau de la tête ainsi que des poignets interchangeables.

---

<sup>1</sup><http://www.rethinkrobotics.com/baxter/>



Figure 3.5 Photo du cobot Baxter de Rethink Robotics

Côté informatique, un kit de développement logiciel est mis à disposition du programmeur et fournit une série de fonctionnalités permettant de contrôler le robot ainsi qu'un certain nombre d'exemples pour se familiariser avec l'architecture de l'interface. Grâce à cette API open source qui fonctionne avec ROS (*Robot Operating System* cf. section 4.1) l'utilisateur peut développer des programmes puis les tester sur deux niveaux : dans un premier temps, sur le simulateur Gazebo fourni avec Baxter, puis directement sur le robot en envoyant à son processeur les mêmes commandes via un ordinateur. En effet, le robot n'est pas équipé d'un ordinateur intégré, mais uniquement d'un CPU troisième génération Intel Core i7-3770. Cela lui suffit pour réaliser une grande variété d'applications, qu'il s'agisse de tâches industrielles, d'apprentissage ou de production dans un environnement collaboratif, tout en assurant une sécurité optimale envers lui-même, mais surtout envers son ou ses homologue(s) humain(s). Cette sécurité est rendue possible grâce à son modèle de collision, décrit ci-haut.

### 3.3 Construction d'un système haptique à cinq DDL

Le Falcon est un outil haptique qui se contrôle à l'aide de la main. On rappelle qu'il possède uniquement trois degrés de liberté, autrement dit, trois translations qui décrivent uniquement une position dans l'espace. Or, l'organe terminal (poignet) du robot qu'on cherche à contrôler possède également des rotations. Au total il faut six DDL pour décrire le mouvement du poignet de Baxter : trois translations pour la position et trois rotations pour l'orientation. Il n'est donc pas possible de contrôler de façon robuste et efficace le robot si notre outil haptique n'est pas capable de représenter l'orientation dans l'espace. Ceci constitue le plus gros inconvénient du Falcon, même si on pourrait également citer son espace de travail restreint à un volume d'environ  $10\text{cm}^3$ .

Heureusement, le travail réalisé par un laboratoire de l'université de l'Utah [Shah *et al.*, 2010] permet de pallier ce problème. L'idée derrière ce concept est de combiner deux Novint Falcon afin d'obtenir un système possédant deux rotations supplémentaires qui viennent s'ajouter aux trois translations déjà existantes, tout en augmentant la taille de l'espace de travail. Cette technique tend à améliorer la maniabilité de l'outil et augmente le niveau de contrôle auprès de l'utilisateur, grâce à une meilleure description des commandes. Certes, cela n'est pas suffisant pour couvrir l'ensemble des mouvements dans l'espace trois dimensions (6 DDL), mais ce projet a trouvé une alternative en proposant deux configurations : une verticale et une horizontale.

On notera que l'outil a été conçu à des fins éducatives : il permettait à des étudiants d'expérimenter une grande variété de rétroactions de forces et de couples. En ce qui nous concerne, nous avons voulu l'utiliser pour le contrôle du bras robotique de Baxter. Les modifications matérielles seront décrites dans un premier temps. Ensuite, une méthode mathématique de compensation de gravité sera exposée pour finir avec la conception d'un contrôleur haptique. Il s'agira du premier élément constitutif de la plateforme de contrôle.

#### 3.3.1 Modifications matérielles

Si le lecteur sent le besoin d'obtenir plus de détails sur la procédure, toutes les modifications apportées sont disponibles sur le site du laboratoire *Telerobotics*<sup>2</sup>. Dans ce projet universitaire, deux Falcons sont raccordés l'un avec l'autre par un stylet, formant ainsi un système dont l'architecture est en boucle fermée et qu'il est possible d'observer sur la figure 3.6. Comme il est précisé dans [Shah *et al.*, 2010], toutes les transformations sont non-destructibles, ce qui constitue un réel avantage.

---

<sup>2</sup><http://www.telerobotics.utah.edu/index.php/Research/5-DOFFalcons>

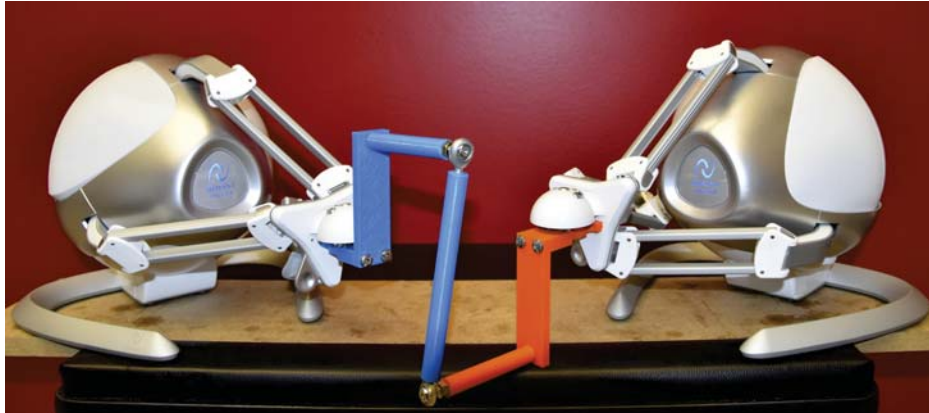


Figure 3.6 Vue de face de l'outil haptique qui combine les deux Falcons

Deux types de configurations sont présentées dans leur travail : une verticale et une horizontale facilement interchangeable en fonction des besoins des applications. Dans notre cas, seule la disposition verticale nous intéresse compte tenu de la structure de Baxter. En effet, sa grande taille, l'amplitude de ses bras, mais surtout la force de ses moteurs lui rende difficilement réalisables des tâches horizontales. Il est donc plus judicieux d'opter pour une structure permettant un contrôle vertical.

Une partie de l'effecteur original est retirée pour être remplacée par une pièce (de couleur bleu et orange sur la figure 3.6) qu'on fixe grâce à un joint sphérique au stylet, permettant ainsi la connexion entre les deux Falcon. Les pièces ont toutes été imprimées par une imprimante 3D<sup>3</sup>, ce qui leur confère une masse relativement faible comparée aux autres composants du dispositif :  $m_e = 91.5g$ . On fixe ensuite les deux appareils sur une planche avec un espacement  $d$  et une orientation de  $45^\circ$  pour maximiser l'espace de travail (position et orientation). Le schéma ci-dessous indique les dimensions de l'installation :

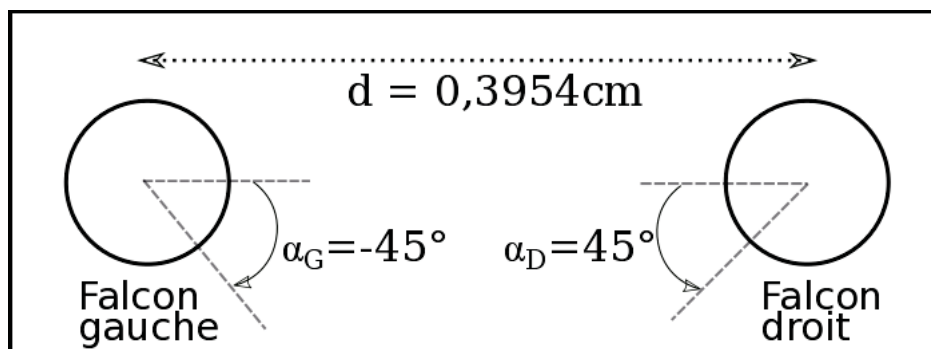


Figure 3.7 Vue d'ensemble de la disposition des Falcons pour la configuration verticale du système haptique (vue de dessus)

<sup>3</sup><http://3d-trip.ca/>

### 3.3.2 Compensation de gravité

L'ajout d'une masse supplémentaire au système nécessite une compensation de manière à ce que le mouvement ne soit pas influencé par la gravité, autrement dit que l'effecteur ne tombe pas, peu importe sa position.

On commence par faire l'hypothèse que les deux joysticks se partagent la masse du nouveau stylet de façon équitable. Le travail réalisé dans [Stamper, 1997] est adapté à l'architecture du Falcon, qui présente toutefois quelques différences. Premièrement, la structure et la disposition du manipulateur décrit par Stamper le contraignent à des manipulations verticales, à la différence du Falcon qui, orienté sur le côté, est contraint à des manipulations horizontales. Ceci engendre alors une répartition inégale des charges sur chacun des bras du robot parallèle.

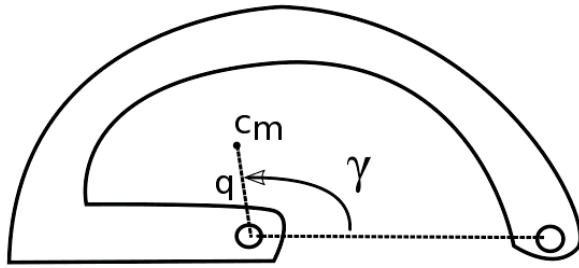


Figure 3.8 Centre de masse du lien en courbe de l'outil haptique Novint Falcon

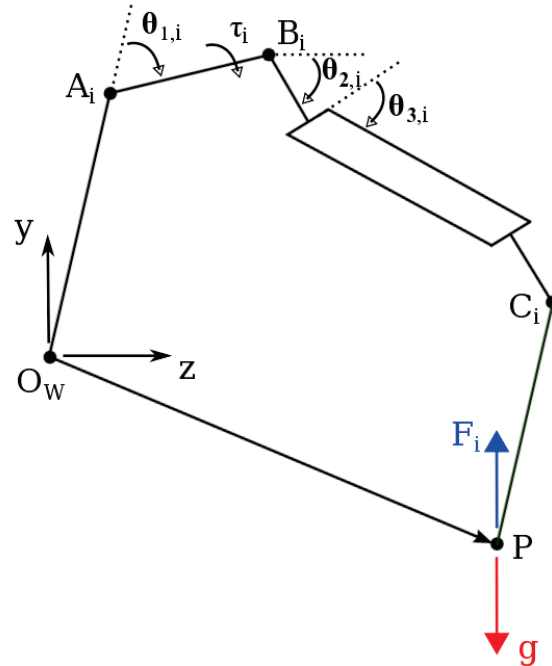


Figure 3.9 Forces et couples appliqués sur la  $i^{eme}$  chaîne cinématique du Falcon

Deuxièmement, la courbure du premier lien du Falcon (lien courbé) lui confère un centre de masse différent de s'il s'agissait d'un lien rectiligne. La figure 3.8 montre comment de façon expérimentale, le centre de masse  $C_m$  du premier lien est trouvé dans [Shah *et al.*, 2010] à une distance  $q$  et un angle  $\gamma$ . Pour la suite de la procédure, seul un bras est considéré sachant qu'elle est applicable aux trois chaînes cinématiques. Ces deux ajustements exposés, l'équation du mouvement de Newton-Euler qui consiste à faire la somme des



moments autour du joint actif ( $A_i$  sur la figure 3.9) est appliquée. On obtient l'équation suivante :

$$\sum M_{A_i} = I_A \ddot{\theta}_{1i} + c_d \dot{\theta}_{1i} + \tau_{mp,i}^* \quad (3.6)$$

où  $M_{A_i}$  est le moment autour du joint  $A_i$ ,  $I_A$  correspond au moment d'inertie,  $c_d$  au coefficient d'amortissement visceux et  $\tau_{mp,i}^*$  est un tableau des charges appliquées au joint actif de chaque bras.  $\ddot{\theta}_{1i}$  et  $\dot{\theta}_{1i}$  représentent respectivement l'accélération et la vitesse angulaire de l'angle actif. La figure 3.9 fait état des forces qui s'appliquent au niveau de l'organe terminal du Falcon (point P) : on cherche à compenser la gravité  $g$  qui s'applique au point P par une force  $F_i$ . En remplaçant dans l'équation 3.6 les termes par les couples moteurs et les forces de gravitation, on obtient :

$$\begin{aligned} \tau_i = & gm_a q (\sin(\gamma - \theta_{1i}) \sin(\phi_i)) - g(m_b + m_d) a (\sin(\theta_{1i}) \sin(\phi_i)) \\ & + g(3(m_b + m_d) + m_c + m_e) J^{-T} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \end{aligned} \quad (3.7)$$

où  $\tau_i$  est le couple associé à la force  $F_i$ ,  $\theta_{1i}$  l'angle actif de la chaîne cinématique considérée et  $J^{-T}$  l'inverse de la matrice Jacobienne. Les autres symboles et leur signification sont répertoriés dans le tableau 3.1. Toutes les valeurs numériques ont été trouvées dans la librairie *Libnifalcon* [Machulis, 2007]. Pour plus de détails, le lecteur peut se référer aux travaux de [Shah *et al.*, 2010] et [Stamper, 1997].

Nom de la variable	Valeur	Description
$m_a$	0.08935kg	masse du 1 <sup>er</sup> lien (courbé)
$m_b$	0.00841kg	masse du lien parallèle
$m_c$	0.03278kg	masse de la plateforme mobile
$m_d$	0.01037kg	masse des joints
$m_e$	0.0915kg	masse du nouveau stylet
$g$	$9,81m^3kg^{-1}s^{-2}$	constante de gravitation
$\gamma$	100°	c.f figure 3.8
$q$	22cm	c.f figure 3.8
$\phi_1$	105°	angle du 1 <sup>er</sup> bras c.f figure 3.3a
$\phi_2$	-15	angle du 2 <sup>eme</sup> bras c.f figure 3.3a
$\phi_3$	-135°	angle du 3 <sup>eme</sup> bras c.f figure 3.3a

Tableau 3.1 Tableau des différentes variables pour la cinématique du Novint Falcon [Machulis, 2007]

### 3.3.3 Contrôleur à cinq DDL

Maintenant que les modifications physiques et la compensation de gravité ont été traitées, il est indispensable de coordonner les deux joysticks de façon à ce que l'utilisateur contrôle un système à cinq degrés de liberté. En d'autres termes, l'utilisateur doit se servir du stylet comme s'il s'agissait d'un seul outil : il faut donc que les deux Falcons fonctionnent simultanément.

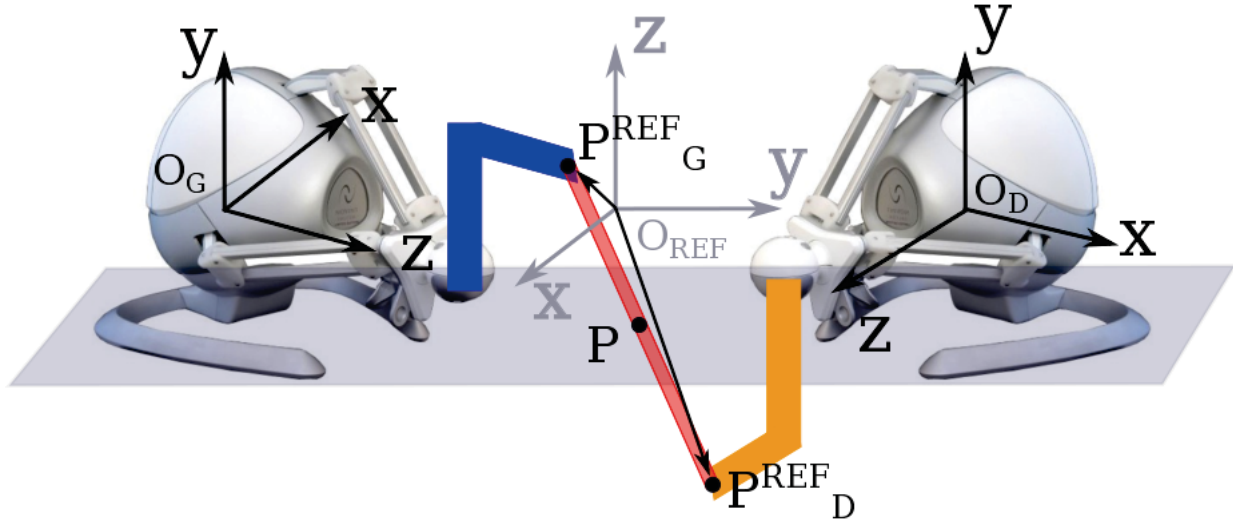


Figure 3.10 Schématisation des repères pour le contrôleur construit à partir des deux Falcons. P correspond au point central du stylet, représenté par un rectangle rouge

On commence par définir un repère de référence situé exactement au milieu et à la même hauteur que les repères de référence de chaque Falcon (repère  $O_{REF}$  sur la figure 3.10). Ce repère servira de base pour les calculs qui suivent. Il s'agit ensuite d'exprimer dans ce repère les positions respectives des effecteurs des deux dispositifs haptiques qu'on notera  $P_G^{REF}$  (pour le Falcon de gauche) et  $P_D^{REF}$  (pour celui de droite). Pour ce faire, on utilise la matrice homogène de transformation ci-dessous :

$$P_i^{REF} = \begin{bmatrix} \sin(\alpha_i) & 0 & \cos(\alpha_i) & 0 \\ \cos(\alpha_i) & 0 & -\sin(\alpha_i) & \pm \frac{d}{2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P_i \quad \text{tel que} \quad P_i = \begin{bmatrix} P_{xi} \\ P_{yi} \\ P_{zi} \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ y_{offset} \\ z_{offset} \\ 1 \end{bmatrix} \quad (3.8)$$

avec  $i = \{G, D\}$ ,  $d$  la distance entre les deux Falcons (cf. figure 3.7) et  $P_i$  représente la position de l'organe terminal (après modification) dans les repères respectifs des Falcons. Grâce à ces données,  $P_i^{REF}$  est calculée en récupérant la position de chaque effecteur ( $P_i$ ),

en y ajoutant un offset  $\begin{bmatrix} 0 \\ y_{offset} \\ z_{offset} \end{bmatrix}$  lié à la modification des outils et enfin en appliquant la matrice de rotation qui effectue un changement de repère. Pour une meilleure compréhension, les repères sont dessinés sur la figure 3.10 ci-dessus.

On souhaite dès lors définir une position en un point P du stylet ainsi qu'une orientation. Ces informations pourront ensuite servir de commandes pour le robot. P est défini comme étant le point central du stylet tel que :

$$P = \frac{P_G^{REF} + P_D^{REF}}{2} \quad (3.9)$$

Pour ce qui est de l'orientation du stylet, on se sert des angles d'Euler, plus précisément de la représentation *roll-pitch-yaw* (RPY). Celle-ci permet d'exprimer l'ensemble des rotations du stylet dans un espace à trois dimensions. La méthode RPY est très utilisée

dans le domaine de l'aéronautique, car elle indique les changements d'altitudes d'un avion dans les airs. Euler a défini dans cette représentation, trois angles qui suffisent à décrire l'orientation d'un corps rigide dans l'espace 3D. Le choix de ce modèle plutôt qu'une autre est justifié, car cette série

de trois angles  $\begin{bmatrix} \varphi \\ \vartheta \\ \psi \end{bmatrix}$  représente une rota-

tion autour d'un axe fixe, attaché au centre de masse du solide considéré. Ainsi, notre choix concernant la position du point P est validé du fait qu'il correspond exactement au centre de masse du stylet (centre de masse d'un cylindre).

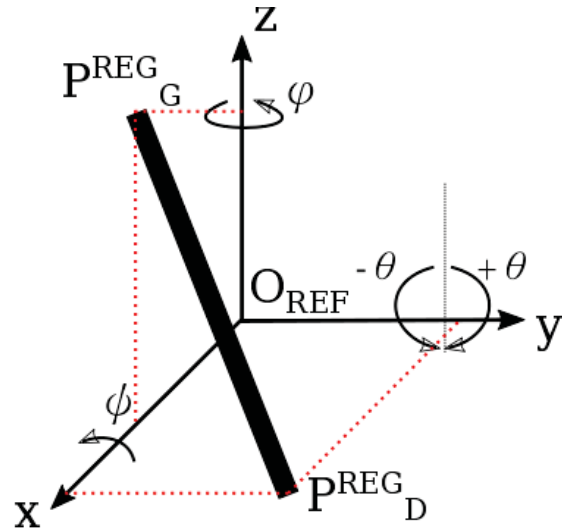


Figure 3.11 Orientation du stylet selon la norme RPY des angles d'Euler

Les angles d'Euler apparaissent en fonction de leur axe de rotation sur la figure 3.11. La méthode est décrite par la procédure suivante :

- rotation du repère de référence d'un angle  $\psi$  autour de l'axe  $x$  (yaw),
- rotation du repère de référence d'un angle  $\vartheta$  autour de l'axe  $y$  (pitch),

- rotation du repère de référence d'un angle  $\varphi$  autour de l'axe  $z$  (roll),

Par une projection axiale et grâce aux relations trigonométriques, on obtient les trois équations 3.10, 3.11 et 3.12 suivantes :

$$\psi = \text{atan2}(\sqrt{(P_G^w(x) - P_D^w(x))^2 + (P_G^w(z) - P_D^w(z))^2}, P_G^w(y) - P_D^w(y)) \quad (3.10)$$

$$\vartheta = \text{atan2}(P_G^w(x) - P_D^w(x), P_G^w(z) - P_D^w(z)) \quad (3.11)$$

$$\varphi = 0 \quad (3.12)$$

De plus, l'angle  $\vartheta$  est défini de façon à ce que le stylet soit vertical lorsque  $\vartheta = 0$  ;  $\vartheta$  est négatif si on tourne dans les sens inverse des aiguilles d'une montre et positif autrement. Il n'y pas de rotation possible autour de l'axe  $z$ , c'est pour cela qu'on se retrouve avec un système à cinq degrés de liberté et que plusieurs configurations du stylet sont prévues, comme cela est expliqué dans [Shah *et al.*, 2010].

# CHAPITRE 4

## UTILISATION D'UN CONTRÔLEUR HAPTIQUE POUR LA MANIPULATION D'UN BRAS ROBOTIQUE

### 4.1 La plateforme robotique ROS

ROS (Robot Operating System) est un méta-système d'exploitation complet (entre l'OS et le *middleware*), conçu pour la robotique de service et qui remplit les mêmes fonctions que n'importe quel autre système d'exploitation, tout en fournissant des fonctionnalités de haut niveau supplémentaires. La création d'un tel ensemble de programmes vient proposer aux utilisateurs des fonctionnalités standardisées faisant abstraction du matériel, afin d'éviter au concepteur de devoir à chaque fois recréer un logiciel embarqué pour chaque nouveau robot. Cet outil de développement robotique va permettre de faciliter l'intégration du contrôleur haptique dans une application interactive avec le robot baxter. ROS s'est développé autour de cinq principes, listés ci-dessous :

- une architecture *peer-to-peer*, permet à chaque bloc d'une application de dialoguer en direct avec tous les autres blocs du système,
- un OS multi-langage (python, C++, java, Lisp...),
- un design basé sur un *microkernel* qui utilise un grand nombre de petits outils pour compiler et lancer les blocs d'une application,
- une facilité d'utilisation (la complexité se trouve cachée dans les librairies),
- le paquet est complètement open source.

Cette collection d'outils et d'algorithmes fournit à l'utilisateur un grand nombre de possibilités : par exemple, il peut modéliser un robot avec le format standard utilisé par ROS (URDF), le visualiser et le simuler en 2D (Stage) ou en 3D (Gazebo, Rviz), traiter des images en provenance du robot (Opencv) ou encore calculer plus facilement la cinématique du robot grâce à des outils de transformations (tf).

Or, la compagnie Rethink Robotics qui à conçu le robot Baxter, fournit en plus d'un SDK, un kit de développement logiciel de recherche (RSDK) qui fonctionne sous ROS. On retrouve dans ce RSDK toutes les fonctionnalités qui permettent de communiquer et de commander le robot collaboratif, d'accéder aux capteurs et aux retours visuels des caméras. De plus, une description XML du robot permet à ROS de générer une simulation physique et mécanique du robot grâce à ses simulateurs. Cette interface avec ROS est très utile puisque l'utilisateur peut développer ses programmes et applications dans n'importe quel langage accepté par ROS, le tester en simulation sur ordinateur puis directement sur Baxter. En effet, une des grandes forces du RSDK est qu'il n'y a aucune différence entre les commandes sous simulation et celles sur le robot réel.

## 4.2 Le contrôleur haptique à cinq degrés de liberté

L'outil haptique conçu pour la maîtrise présente l'avantage d'avoir une librairie de fonctions open source en comparaison avec d'autres outils commercialisés. Certaines fonctions déjà développées sous *Libnifalcon* vont pouvoir être réutilisées, modifiées et adaptées au contrôleur. Comme nous venons de le voir, ROS est un OS très puissant en robotique : il sert donc de base au développement de la plateforme de contrôle haptique. De plus, tous les blocs ROS sont codés en langage C/C++ et Python.

Dans la section 3.3.3 nous avons parlé de la position et de l'orientation du stylet de l'outil haptique. L'ensemble des équations a été intégré dans un bloc ROS qui correspond aux commandes de notre système. Il convient dans un premier temps de vérifier si ces commandes sont respectées.

### 4.2.1 L'espace de travail

L'inconvénient de l'outil haptique Falcon est qu'il est limité par son espace de travail. En effet, les trois chaînes cinématiques qui fonctionnent en parallèle dans le but de faire bouger la plateforme mobile sont limitées par la longueur de leurs liens et par les angles de leurs joints. L'espace de travail accessible par l'effecteur est délimité par une forme tri-hémisphérique légèrement tordue, qui chevauche l'axe longitudinal  $z$  [Martin et Hillier, 2009].

Pour cette expérience, il faut commencer par brancher les Falcons à l'ordinateur, puis les basculer en mode *home* (c.f figure 3.4). Le bloc qui calcule la position et l'orientation est alors lancé et l'utilisateur commence à déplacer le stylet, alors que les positions et

orientations sont récupérées. Lorsqu'on génère les espaces de travail (en utilisant le logiciel

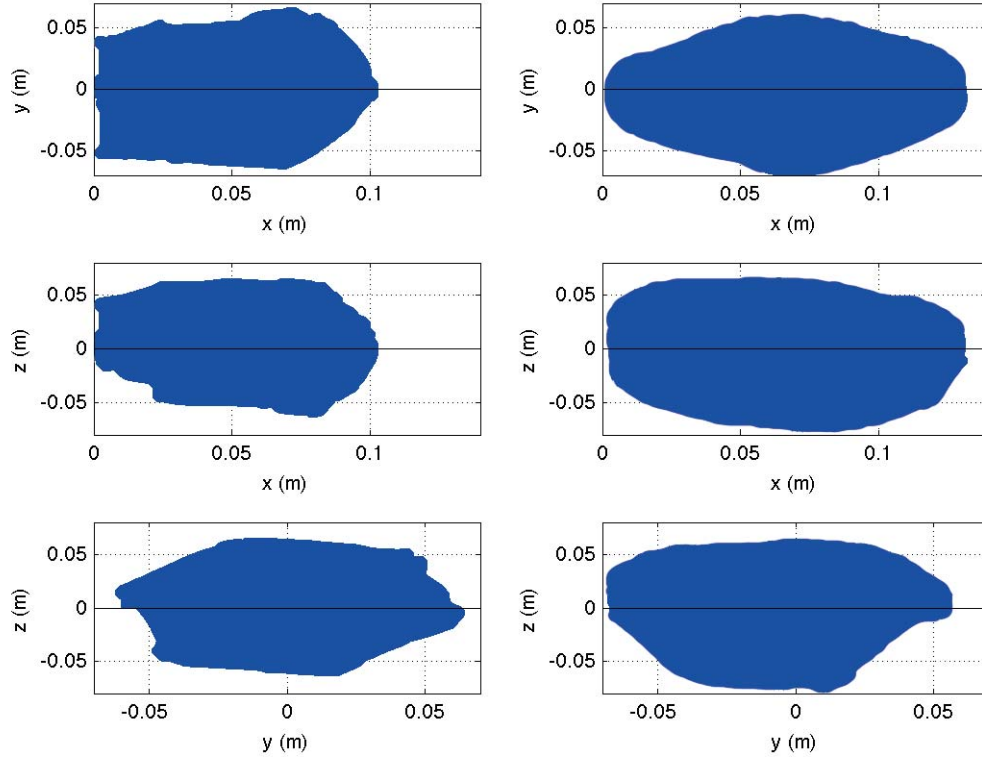


Figure 4.1 Comparaison des espaces de travail entre le contrôleur haptique à 5 DLL et l'effecteur standard du Falcon. À gauche, les différents plans correspondent aux espaces accessibles pour l'effecteur et à droite à ceux du nouveau joystick haptique.

Matlab) relatifs au Falcon seul et au nouveau contrôleur haptique qui combine deux Falcons (c.f figure 4.1), on constate que ce dernier possède une zone accessible par le stylet agrandi de 25%, par rapport à l'organe terminal du Falcon seul. Celui-ci passe de  $10\text{cm}^3$  à pas loin de  $12,5\text{cm}^3$ , tout en ajoutant la possibilité de deux rotations autour des axes  $x$  et  $y$ . Effectivement, l'objectif d'associer deux Falcons visait à augmenter le nombre de DDL à cinq et permettre ainsi un contrôle plus réaliste. Cependant, la structure physique du nouveau dispositif, combiné avec celle du Falcon, limite les rotations du stylet qui sont contraintes à respecter les plages angulaires suivantes :

$$\frac{\pi}{4} < \psi(x) < \frac{3\pi}{4} \quad (4.1)$$

$$-\frac{\pi}{4} < \vartheta(y) < \frac{\pi}{4} \quad (4.2)$$

On rappelle que  $\psi$  correspond à l'angle entre l'axe  $x$  et le stylet alors que  $\vartheta$  représente la rotation autour de l'axe  $y$  (c.f figure 3.11). Même si l'augmentation de l'espace de travail n'est pas significative, les deux nouvelles rotations possibles ont leur importance. Or, la main humaine, qui déplacera le stylet, possède un maximum de degrés de liberté avant de franchir le cap de la redondance, autrement dit six : trois translations et trois rotations. Il est vrai qu'avec seulement 5 DDL tous les mouvements ne peuvent pas être reproduits, mais on considère qu'il s'agit déjà d'une bonne amélioration compte tenu des dépenses allouées à cette installation. Le coût total de cette installation avoisine les \$500 : deux Falcons au prix de \$200 auquel on ajoute \$100 pour les pièces qui ont permis la construction du contrôleur.

Comparé au prix du Phantom Omni à six DDL qui est supérieur à \$10 000, notre contrôleur est quant à lui dix cinquante fois moins cher. Certes notre dispositif possède un DDL de moins et il reste à évaluer sa précision, mais en ce qui nous concerne, il s'agit d'une alternative bon marché.

## 4.2.2 Planification de trajectoire pour le contrôleur

Le fonctionnement du contrôleur haptique est vérifié à travers plusieurs expérimentations :

- un suivi de trajectoire en position : on fixe une position désirée ainsi qu'un profil de trajectoire pour y parvenir, puis on transmet ces informations au contrôleur. On relève alors les valeurs réelles obtenues sous ces conditions, qu'on compare avec les valeurs attendues (tous les graphiques sont réalisés en utilisant le logiciel Matlab),
- un suivi de l'orientation du stylet : on procède de façon similaire au suivi de trajectoire en position, mais avec l'orientation du stylet,
- comparaison avec et sans compensation de gravité : une commande est envoyée au contrôleur puis brusquement arrêtée afin d'observer l'influence de la compensation de gravité.

### 4.2.2.1 Suivi de la trajectoire en position

On commence par définir un profil de trajectoire en considérant uniquement la position. C'est une polynomiale cubique qui sera utilisée pour définir cette trajectoire en position : elle est décrite par un mouvement point à point. L'équation qui régit cette génération de



trajectoire est donnée par :

$$q(t) = a_3t^3 + a_2t^2 + a_1t + a_0 \quad (4.3)$$

où  $a_3$ ,  $a_2$ ,  $a_1$  et  $a_0$  sont les variables à déterminer. Cette équation a pour caractéristiques, une vitesse parabolique  $\dot{q}(t)$  et une accélération linéaire  $\ddot{q}(t)$  par rapport aux angles des joints  $q(t)$  :

$$\dot{q}(t) = 3a_3t^2 + 2a_2t + a_1 \quad (4.4)$$

$$\ddot{q}(t) = 6a_3t + 2a_2 \quad (4.5)$$

Une trajectoire entre deux points dans l'espace cartésien  $P_A$  (point initial) et  $P_B$  (point final) est établie par le système d'équations suivant :

$$a_0 = P_A \quad (4.6)$$

$$a_1 = \dot{P}_A \quad (4.7)$$

$$a_3t_f^3 + a_2t_f^2 + a_1t_f + a_0 = P_B \quad (4.8)$$

$$3a_3t_f^2 + 2a_2t_f + a_1 = \dot{P}_B \quad (4.9)$$

Pour cette méthode, les vitesses des joints initiales et finales sont généralement imposées et fixées à zéro. En effet, à un instant  $t = 0$ , un joint possède une accélération ainsi qu'une vitesse nulle.

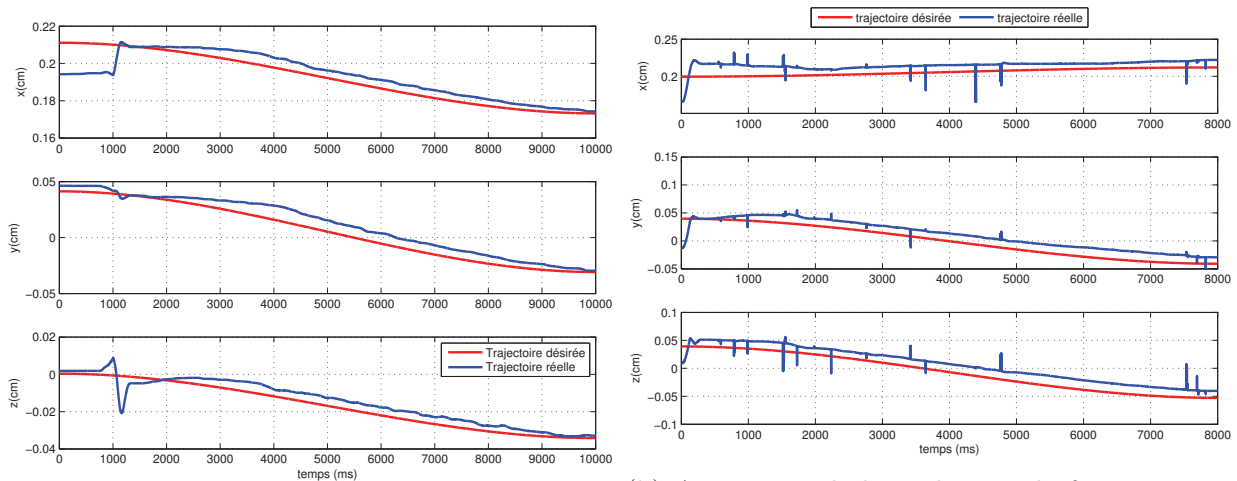
Dans le contrôleur haptique développé, une orientation et une position définissent l'emplacement du stylet à chaque instant. En revanche, à l'intérieur du contrôleur, chaque Falcon est contrôlé séparément. On est obligé de fonctionner ainsi, car il n'est pas possible de contrôler avec une seule commande les deux Falcons. En conséquence, une trajectoire est générée pour le stylet puis chaque point est envoyé au contrôleur, qui s'occupe de reconstruire les positions des deux appareils couplés, dans leurs repères respectifs. Ce procédé est l'inverse de celui présenté dans la section 3.3.3.

Les positions propres à chaque outil sont ensuite transmises à un régulateur PID qui veille au bon fonctionnement du système. L'équation 4.10 fixe la force à envoyer au micrologiciel des Falcons qui s'occupe de la convertir en couple et de la transmettre aux moteurs :

$$F = K_p\Delta P + K_i \int_0^t \Delta P + K_d \frac{d\Delta P}{dt} \quad (4.10)$$

où  $K_i$ ,  $K_d$  et  $K_p$  sont les gains respectifs de l'intégrateur, du dérivateur et du proportionnel.  $\Delta P$  correspond à l'erreur en position et  $F$  est la force de contrôle du PID. Afin de déterminer les gains, une impulsion est envoyée au système et par une technique dites d'essais-erreurs, les caractéristiques du PID sont déterminées en veillant toutefois à respecter les critères suivants :

- $K_i$  et  $K_d$  sont à zéro ; le coefficient  $K_p$  est réglé de manière à améliorer le temps de réponse du système, c'est-à-dire il doit permettre au système de se rapprocher très vite de la consigne tout en faisant attention de garder la stabilité du système : il faut veiller à ce que le système cesse d'osciller,
- une fois ce coefficient réglé, vient le tour de  $K_i$  qui annule l'erreur finale du système pour que celui-ci respecte exactement la consigne.  $K_i$  est donc fixé de manière à avoir une réponse exacte en peu de temps tout en essayant de minimiser les oscillations apportées par l'intégrateur,
- enfin, on peut passer au dernier coefficient dérivateur  $K_d$ , qui tend à stabiliser le système. Son réglage permet donc de diminuer les oscillations.



(a) Suivi de la trajectoire en position du stylet (fréquence de 1KHz)

(b) Apparition de bruit lorsque la fréquence est trop basse : sur la courbe bleue, on observe des pics qui reflètent les oscillations

Figure 4.2 Illustrations des problèmes rencontrés si l'on fait varier la fréquence

Grâce au contrôleur il est possible de relever les positions réelles du stylet, qui peuvent alors être comparées aux données transmises. Les commandes sont envoyées au contrôleur à une fréquence de 800Hz pour pallier le problème mentionné dans la section 3.1.4. Pour se replacer dans le contexte, si la fréquence est trop élevée, certaines commandes ne seront pas reçues. Comme l'avait déjà constaté [Martin et Hillier, 2009], lorsque les données transitent

à une fréquence de 1kHz on observe un décalage temporel voire une perte des informations transmises par rapport à celles recueillies (c.f figure 4.2a).

De même, on constate que la baisse de fréquence engendre une légère diminution de la fluidité du mouvement, autrement dit, la trajectoire est un peu plus saccadée et le stylet se déplace davantage par "à coup". Ceci se traduit par un léger bruit qui influence le suivi de trajectoire et qu'on relève pour une fréquence inférieure à 800Hz (c.f figure 4.2b). Une solution à ce problème sera proposée dans la section 4.3.3. Ces phénomènes montrent l'importance de bien choisir la fréquence d'envoi des données.

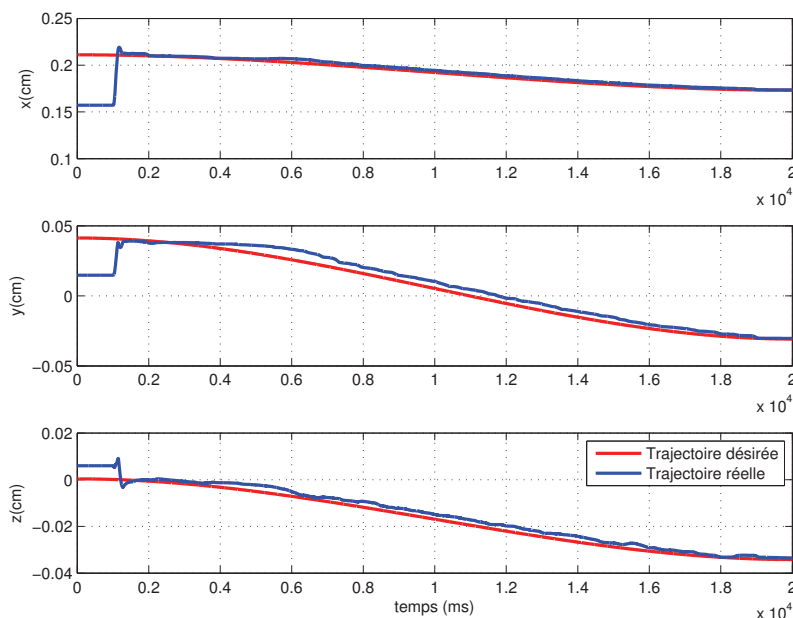


Figure 4.3 Suivi de la trajectoire en position du stylet (fréquence de 800Hz)

En revanche, pour une fréquence de 800Hz les phénomènes décrits précédemment, ne sont plus présents et le contrôleur suit la trajectoire sans trop d'erreurs comme le montre la figure 4.3. Cette trajectoire est respectée à travers le temps avec des erreurs maximales de 0,2% pour un suivi selon l'axe  $x$ , de 1,4% pour l'axe  $y$  et de 1,9% pour l'axe  $z$ . En comparant avec les trajectoires à 1kHz, on obtient un pourcentage d'erreur légèrement plus élevé : 0,6% pour l'axe  $x$ , 3,7% pour l'axe  $y$  et 3,8%. On préférera donc une fréquence de 800Hz, qui selon les conclusions des expériences, présente les meilleurs résultats en terme de suivi de trajectoire. Avec un pourcentage d'erreur maximal inférieur à 2%, nous pouvons conclure quant à la précision du dispositif, qui est plus qu'acceptable pour un suivi de trajectoire en position. Par ailleurs, on remarque également que le mouvement du stylet est souple, fluide et sans oscillations ce qui est tout à fait conforme à nos attentes. En effet, on désire éviter au maximum des bruits sur le stylet, même s'ils sont négligeables,

car ils peuvent avoir des répercussions plus importantes sur une interaction homme-robot qui demande une précision certaine.

### 4.2.2.2 Suivi de la trajectoire en orientation

En second lieu, on cherche à vérifier si l'orientation du stylet est respectée lorsque soumise à une commande. Une trajectoire linéaire basée sur les angles d'Euler est alors générée :

$$\phi_e = \phi_i + \frac{s}{\|\phi_f - \phi_i\|}(\phi_f - \phi_i) \quad (4.11)$$

où  $s$  est une variable qui évolue en fonction du temps. Les vecteurs  $\phi_i$ ,  $\phi_f$  et  $\phi_e$  représentent respectivement les trios d'angles d'Euler  $(\varphi, \vartheta, \psi)$  initial, final et échantillonné.

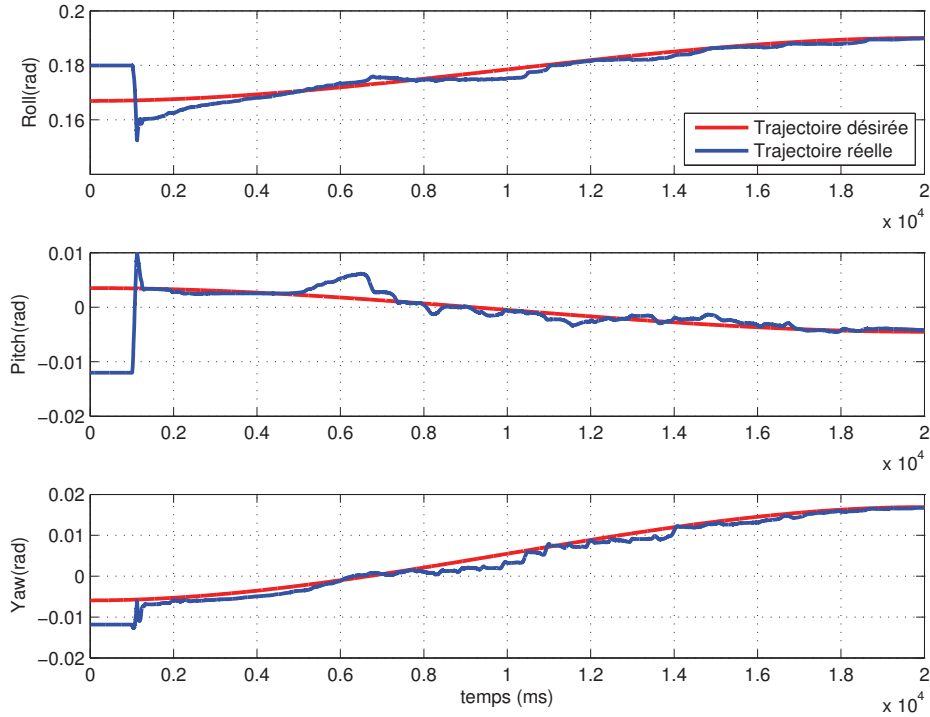


Figure 4.4 Suivi de la trajectoire en orientation du stylet

De la même manière que précédemment, les commandes sont envoyées à chacun des Falcons et les valeurs des angles sont relevées grâce au contrôleur, puis comparées avec celles transmises.

On constate que les trois trajectoires *roll*, *pitch* et *yaw* ont été respectées avec une erreur maximale de 7% pour l'angle  $\vartheta$ , tandis que pour les autres rotations, les erreurs sont respectivement de 0.07% et 1.23%. Cette différence maximale proche des 10% peut être

expliquée par le phénomène dit "d'à coup" qui vient perturber à un instant donné la trajectoire du stylet, comme mentionné dans le précédent paragraphe. Ce problème sera résolu et détaillé dans une section suivante. Hormis ce phénomène, les résultats obtenus sont encourageants et l'orientation reste bien suivie pour chacun des angles.

### 4.2.2.3 Compensation de gravité du stylet haptique

Afin de vérifier la compensation de la gravité pour le stylet modifié, une position sous forme d'impulsion est envoyée au contrôleur. On désire évaluer l'effet et l'intérêt qu'apporte une force qui vient contre-balancer l'influence de la gravité.

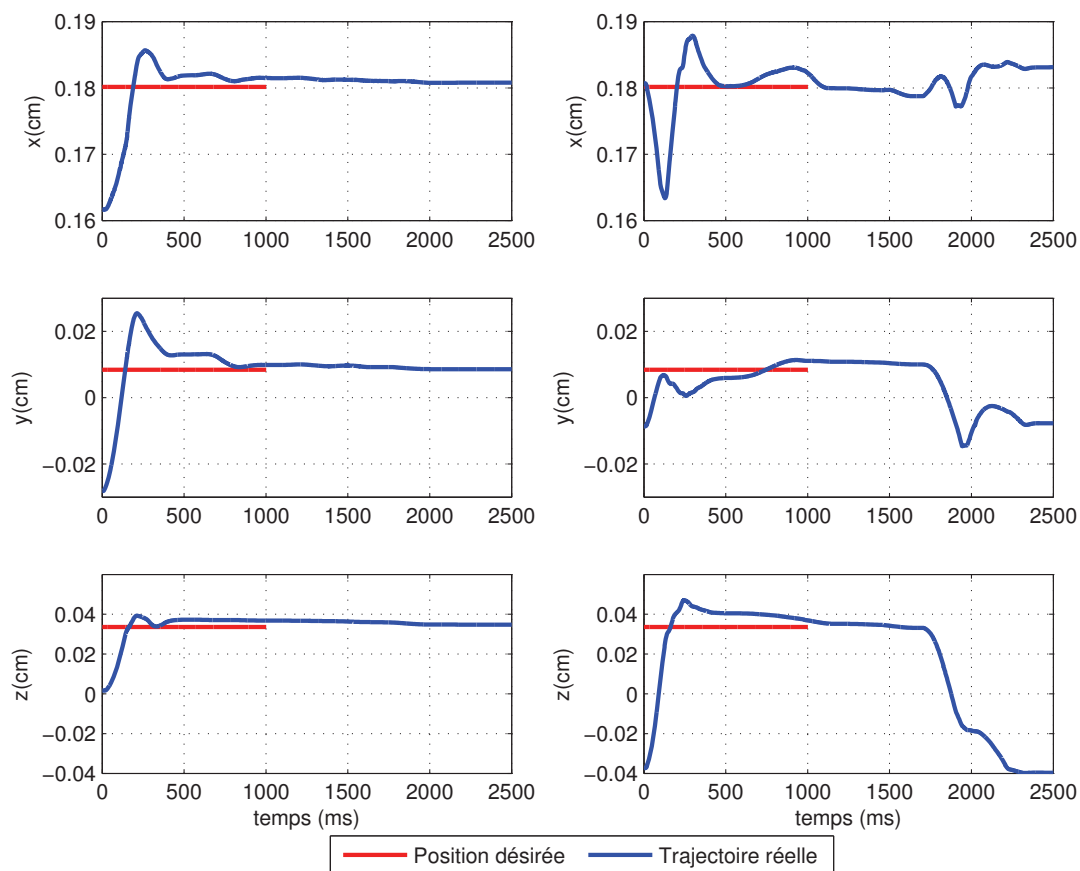


Figure 4.5 Réponse du contrôleur face à une impulsion (à gauche avec la compensation de gravité et à droite sans). Dans cette expérience, la position désirée (courbe rouge) est envoyée sous forme d'impulsion pendant 1s puis arrêtée brutalement afin d'observer l'évolution de la position du stylet.

Les résultats de cette expérimentation sont relevés sur les graphiques de la figure 4.5. La position est envoyée pendant 1s puis on observe l'effet du stylet une fois qu'on arrête d'émettre le signal : sur les graphiques de gauche (compensation de gravité activée), on voit

que la position est maintenue même après la fin d'émission de la position tandis qu'à droite (compensation de gravité désactivée), la position dérive sous l'influence de la pesanteur notamment selon l'axe  $z$ , directement relié à la gravité. De plus, lorsque la compensation de gravité est activée, une force s'applique sur chaque Falcon, ce qui a pour effet d'améliorer le contrôleur *PID*. Effectivement, cela évite d'avoir trop d'oscillations dues à l'application d'une force soudaine.

D'un point de vue du ressenti, le mouvement du stylet est amélioré et procure à l'utilisateur une sensation de fluidité et de légèreté lorsque la compensation de gravité est active. À contrario, la position est moins bien suivie et l'utilisateur a la sensation de déplacer le stylet avec de la friction et sans fluidité. D'autre part, le stylet ne garde pas la position lorsque la main de l'utilisateur est retirée puisque la compensation n'étant pas active, il tombe sous le poids de la gravité.

En conclusion, la compensation de gravité améliore grandement le ressenti auprès de l'utilisateur qui se sent plus à l'aise dans ces mouvements et voit la précision de ses mouvements augmentée.

## 4.3 Téléopération du robot collaboratif Baxter

Cette expérimentation vient répondre à l'objectif principal de cette maîtrise de recherche, à savoir contrôler un robot grâce à un outil haptique. L'outil haptique à cinq degrés de liberté a été évalué et présente une bonne suivie de position et d'orientation au cours du temps. De ce fait, nous avons cherché à déployer ce dispositif dans une application concrète d'interaction homme-robot. Il est question de téléopérer, c'est-à-dire d'opérer à distance, le robot collaboratif Baxter afin de lui faire attraper un objet. Dans l'ordre seront présentés la description du système et des techniques de contrôle pour l'interface haptique, suivie d'une explication sur le fonctionnement du retour de force par l'outil haptique et enfin, l'expérience de téléopération finale sera détaillée.

### 4.3.1 Description de la stratégie

Le système est composé des deux Falcons combinés qui servent d'outil de contrôle pour le robot collaboratif Baxter. Un ordinateur qui supporte l'OS robotique ROS permet de superviser le système et de gérer l'échange d'informations entre les différents composants par le biais d'une plateforme de contrôle (c.f figure 4.6). Celle-ci intègre notamment un bloc

de conversion des commandes issues de l'outil haptique, en commandes compréhensibles par Baxter.

L'amélioration du contrôleur avec l'ajout d'une orientation au stylet permet dès lors d'envoyer des commandes (position et orientation) à l'organe terminal de Baxter, qui seront ensuite converties en valeurs angulaires pour chacune des sept articulations du robot : c'est le rôle de la cinématique inverse. On rappelle que le mouvement du poignet, connecté à l'organe terminal du bras commandé, est descriptible avec 6 DDL : trois paramètres pour la position et trois autres pour l'orientation. D'autre part, la configuration verticale du stylet ne permet pas de rotations selon l'axe  $z$  : il faudra donc s'assurer que l'objet ne soit pas dans une orientation inaccessible par la pince de Baxter.

Le scénario imaginé met en oeuvre un objet placé sur une table que l'utilisateur doit attrapé en commandant un bras robotique. L'objet considéré est un cube de taille suffisamment petite pour que le poignet du robot puisse l'attraper. Grâce à un mapping direct, le bras de Baxter se déplace en même temps que l'utilisateur déplace le stylet : il s'agit d'une commande en temps réel. Enfin, une fonction permet à l'utilisateur de presser sur des touches du clavier pour ouvrir et fermer la pince, connectée au poignet de Baxter, qui permet d'attraper ou de relâcher l'objet.

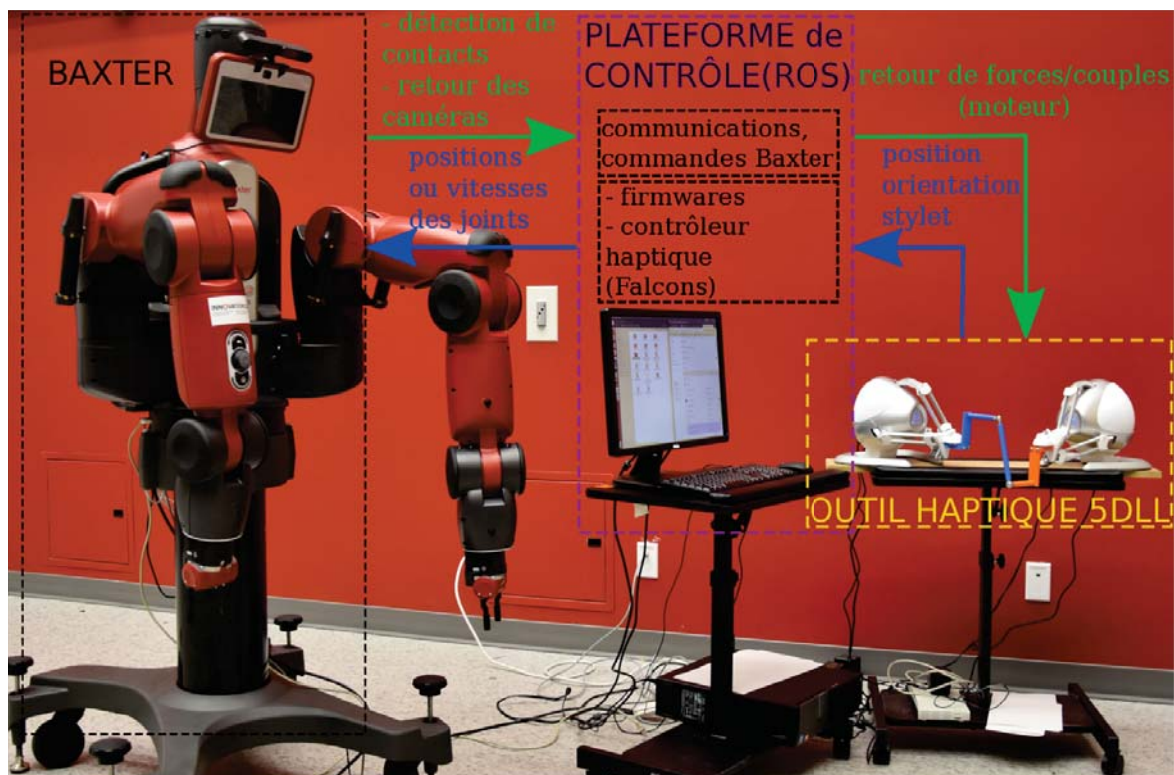


Figure 4.6 Description du système de téléopération du robot Baxter

Ensuite, les informations issues des capteurs de couples présents sur le robot serviront à émettre une force de rétroaction. Cette procédure cherche à améliorer le contrôle robotique en créant une interaction entre l'homme et le robot. Lorsque l'opérateur déplace le stylet du dispositif haptique, le robot se déplace de manière simultanée (mapping direct) et renvoie des informations en temps réel (issues des capteurs) sur son mouvement à l'utilisateur, qui pourra alors apprécier ces données sous forme de forces appliquées au niveau du stylet. Des paramètres relatifs à ce retour haptique seront ensuite définis dans la section 4.3.4.

Finalement, l'utilisation des caméras de Baxter viendra compléter les informations haptiques avec des retours visuels : l'utilisateur verra son champ de contrôle encore augmenté. La figure 4.6 ci-dessus décrit les différents éléments constitutifs du système collaboratif.

### 4.3.2 Les procédures de commandes

La première étape consiste à convertir les informations du contrôleur (position et orientation du stylet) en langage compréhensible par la machine, c'est-à-dire en commandes robotiques. Or, les concepts robotiques proposent plusieurs techniques de commande pour les joints du robot. Il convient de comparer ces possibilités et de déterminer laquelle est la plus appropriée et efficace pour notre application IHR. À première vue, il est possible de contrôler en position, en vitesse ou en couple un robot. D'autre part, l'ensemble des approches passent par le calcul de la cinématique inverse qui convertit position et orientation de l'organe terminal à partir de l'espace cartésien vers l'espace articulaire des joints du robot. Quelques techniques d'optimisation seront également présentées.

#### 4.3.2.1 La cinématique inverse standard

Le problème de la cinématique inverse est utilisé dans la robotique. De ce fait, il n'est pas difficile à imaginer que beaucoup de pistes de solution ont été proposées.

Rethink Robotics fournit dans son kit de développement logiciel, un paquet destiné à la gestion de la cinématique de leur robot Baxter. Celui-ci fonctionne avec deux autres paquets *ROS*, *KDL* et *tf* [Foote, 2013]. Le premier, intitulé *The Kinematics and Dynamics Library* est un outil qui permet de modéliser et de résoudre des calculs sur les chaînes cinématiques, par exemple de robots ou de modèles biomécaniques humains. Cet ensemble de fonctions regroupé dans une librairie procure à l'utilisateur une classe pour la gestion d'objets géométriques (repère, lignes, points) et des chaînes cinématiques en tout genre. Il s'intègre dans un projet, *The Orocos Project*, qui vise à améliorer intelligemment le contrôle et l'automatisation des machines dans le domaine de la robotique. D'ailleurs, il



existe une version de la librairie KDL, développée pour Baxter dans le but d'accélérer la vitesse d'exécution des commandes. C'est grâce à une description URDF (format des modèles robotiques sous ROS) de Baxter que deux chaînes cinématiques sont construites et vont servir de base à toutes les fonctions présentes dans ce paquet. On retrouve des fonctions pour le calcul des cinématiques directe et inverse, d'autres pour déterminer et faire des calculs sur la Jacobienne du robot ou encore pour l'estimation des inerties des différents joints.

Le paquet *tf* quant à lui, permet à l'utilisateur de garder une trace temporelle d'une multitude de repères de coordonnées. Effectivement, chaque système robotique possède plusieurs repères (au moins un par articulation, si ce n'est plus) qui évoluent au fil du temps, en fonction du mouvement du robot. C'est là qu'intervient ce paquet, qui donne la possibilité à l'utilisateur de travailler sur les différents repères d'une structure robotique : il peut par exemple utiliser les fonctionnalités de *tf* pour obtenir la position relative de l'effecteur du robot par rapport à n'importe quel repère de référence, et cela à l'instant actuel, ou 5, 10, 20 secondes auparavant. En plus de cela, l'outil ROS fonctionne de manière distribuée, c'est-à-dire que toutes les informations sur les repères d'un robot sont accessibles par tous les autres composants ROS ainsi que tous les autres ordinateurs du système.

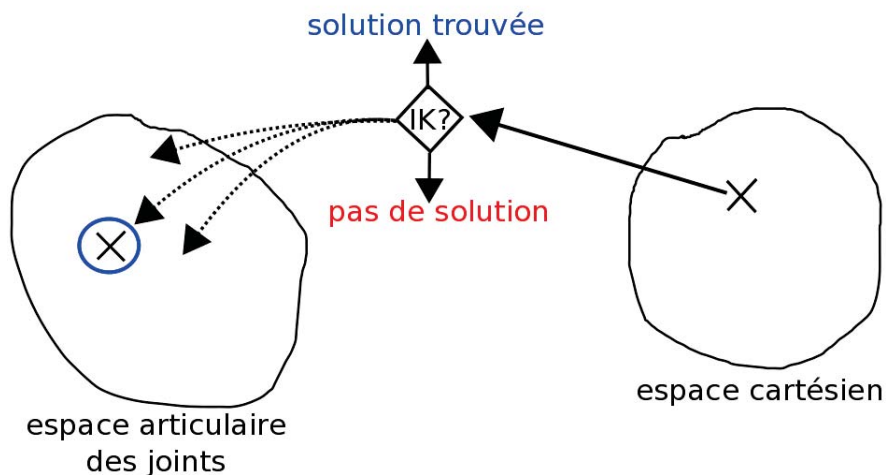


Figure 4.7 Schéma de l'algorithme de cinématique inverse

L'utilisation de ce paquet fourni dans le RSDK fut la première solution envisagée pour le contrôle du robot. Il s'agit de la solution la plus directe et la plus facile à prendre en main, mais ne se trouve pas être la plus efficace.

En utilisant la fonction de la cinématique inverse décrite sous différentes fonctions de KDL, les positions et orientations du stylet sont traduites en valeurs angulaires puis transmises aux joints de Baxter. Le schéma de la figure 4.7 représente le processus derrière l'algorithme de la cinématique inverse (IK) : une position de l'espace cartésien entre de l'algorithme IK qui se met alors à chercher une solution dans l'espace articulaire des joints. Deux issues sont alors possibles : soit l'algorithme trouve une solution parmi l'ensemble des solutions soit, à l'inverse, ne trouve aucune solution.

### 4.3.2.2 La cinématique inverse optimisée

Plus récemment, une alternative aux techniques de calcul de la cinématique inverse proposée par la bibliothèque KDL, a été présentée sous le nom de TRAC-IK [Beeson et Ames, 2015]. Cette librairie open source propose de solutionner un problème souvent rencontré avec l'algorithme de la cinématique inverse standard, à savoir, aboutir à des solutions proches des limites articulaires. Effectivement, dans la méthode décrite plus haut lorsqu'une solution est proche d'une limite articulaire d'un joint, aucune solution n'est trouvée.

Contrairement aux méthodes standard, TRAC-IK ne s'appuie pas uniquement sur les mathématiques de Newton, mais combine simultanément deux algorithmes de cinématique inverse :

- le premier est une extension de l'algorithme KDL qui détecte et réduit les minimaux locaux causés par la présence de limites des joints,
- le second est un algorithme séquentiel quadratique d'optimisation non-linéaire (SQP) qui utilise la méthode quasi-Newton pour améliorer la gestion des contraintes angulaires des joints.

En effet, les techniques d'optimisation utilisent les commandes prédictives ou *model predictive control* (MPC), qui sont des stratégies de contrôle avancées permettant de déterminer les entrées d'un système afin d'optimiser et d'anticiper le futur comportement de celui-ci. Ces techniques utilisent un modèle dynamique du processus pour calculer les actions de contrôle en temps réel ainsi qu'un modèle mathématique pour les prédictions. Qu'il s'agisse d'un MPC linéaire ou non linéaire, la détermination d'une solution rapide, fiable et en temps réel est l'enjeu majeur de ces algorithmes. Lorsqu'il est question de MPC linéaire, un seul problème quadratique (QP) doit être résolu à chaque instant d'échantillonnage. En revanche, s'il s'agit de processus non linéaires, on opte plus pour les méthodes de programmation séquentielle quadratique (SQP) qui sectionnent, à chaque itération, le problème

en plusieurs QP. En résumé, la grande majorité des commandes prédictives s'appuie sur la résolution de problèmes quadratiques convexes en temps réel.

La méthode d'optimisation TRAC-IK utilise des techniques d'algorithmes combinatoires afin d'augmenter les performances de l'algorithme IK tout en diminuant le temps de calcul. Deux algorithmes fonctionnent en parallèle et un résultat est retourné immédiatement lorsque l'un des deux converge vers une solution. De plus, des contraintes secondaires de distance et de maniabilité sont également implantées afin de déterminer le meilleur des résultats de cinématique inverse. Ainsi, il est assuré de trouver une solution lorsque les joints sont à proximité de leurs limites angulaires. Dans [Beeson et Ames, 2015], des résultats comparatifs entre la méthode classique d'Orocos KDL et TRAC-IK sont fournis pour quelques robots. Pour les résumer, on constate que :

- TRACK-IK trouve plus de solutions et affiche un temps moyen de calcul plus rapide, pour tous les robots soumis au test,
- dans le cas de Baxter, la méthode KDL trouve une solution avec un pourcentage de 61.07% en un temps moyen de 2.21ms, alors que la démarche TRAC-IK révèle un pourcentage de solutions trouvées de 99.17% et une moyenne de 0.60ms par solution.

Enfin, une dernière option serait de faire appel à un ensemble de stratégies dynamiques (*online active set strategy*) qui a récemment permis d'améliorer les commandes prédictives [Ferreau *et al.*, 2008]. En émettant l'hypothèse que l'ensemble des contraintes ne change pas énormément d'un problème quadratique à un autre et en tenant compte des solutions des précédents QP, la rapidité des calculs dans les MPC est nettement augmentée. Cette stratégie a par ailleurs été implémentée sous une librairie open source du nom de qpOASES [Ferreau *et al.*, 2014]. L'ensemble des fonctions résout des problèmes quadratiques ayant la forme suivante :

$$\min \quad \frac{1}{2}x^T Hx + x^T g(w_0) \quad (4.12)$$

$$s.t. \quad lbA(w_0) \leq Ax \leq ubA(w_0), \quad (4.13)$$

$$lb(w_0) \leq x \leq ub(w_0), \quad (4.14)$$

où  $x$  est une variable qui représente la solution à déterminer,  $H$  est une matrice Hessienne,  $A$  une matrice de contraintes,  $g$  un gradient,  $lbA(w_0)$ ,  $lb(w_0)$ ,  $ubA(w_0)$  et  $ub(w_0)$  sont des vecteurs de contraintes minimales et maximales.

Ainsi, comme les articulations de Baxter sont soumises à des contraintes physiques, un modèle de contrôle prédictif peut très bien être adapté pour une telle application de

contrôle avancé du robot. Le système d'équations adapté à notre système devient alors :

$$\min \quad \frac{1}{2} \dot{q}^T I_d \dot{q} \quad (4.15)$$

$$s.t. \quad v_e \leq J(q)\dot{q} \leq v_e, \quad (4.16)$$

$$\dot{q}^- \leq \dot{q} \leq \dot{q}^+, \quad (4.17)$$

avec  $J(q)$  la matrice Jacobienne,  $\dot{q}$  la vitesse angulaire des joints,  $\dot{q}^+$  et  $\dot{q}^-$  sont les limites maximales et minimales des vitesses des joints et  $v_e$  représente la vitesse en position et en orientation de l'organe terminal de Baxter (c.f équation 3.5). Ainsi posée, l'inégalité 4.16 devient une égalité correspondant à l'équation différentielle de la cinématique.

Cette solution est semblable à TRAC-IK, mais compte tenu des résultats de performances (sur Baxter) présentés ci-haut, la méthode TRACK-IK sera utilisée dans le cadre de cette recherche.

### 4.3.2.3 La matrice jacobienne pour le contrôle robotique

Parmi les autres options largement explorées pour le contrôle robotique, figure la matrice Jacobienne qui est décrite à travers une équation différentielle. À la différence des méthodes décrites précédemment, celle-ci contrôle le robot en vitesse et non pas en position. La Jacobienne représente l'état actuel d'une structure robotique, c'est-à-dire les dérivées partielles dans l'espace cartésien concernant les valeurs actuelles des joints. Elle est donnée par la relation :

$$v_e = \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} = J(q)\dot{q} \quad (4.18)$$

où  $\dot{p}_e$  et  $\omega_e$  correspondent respectivement à la vitesse linéaire et angulaire de l'effecteur,  $J(q)$  est la matrice Jacobienne et  $\dot{q}$  représente la vitesse des joints. Cette équation différentielle cinématique représente une correspondance linéaire entre l'espace des vitesses des joints et l'espace des vitesses opérationnelles, qui varie en fonction de la configuration du robot. Ainsi, il est possible de résoudre le problème de la cinématique inverse en utilisant cette équation différentielle. La nouvelle configuration des joints est calculée à partir de l'inverse de la Jacobienne  $J^{-1}$  qui décrit les dérivées partielles dans l'espace de joints par rapport à l'espace cartésien. Une solution au problème de la cinématique inverse est alors obtenue par itération de la fonction :

$$q_t = q_{t-1} + J^{-1} \Delta p \quad (4.19)$$

dans laquelle la cinématique directe permet de calculer la nouvelle erreur en position, représentée par  $\Delta p$  ( $q_{t-1}$  étant la configuration précédente). Pour chaque itération une nouvelle valeur articulaire ( $q_t$ ) est calculée puis injectée dans la cinématique directe afin de déterminer une nouvelle position. Cette position est alors soustraite à la position désirée (on obtient alors  $\Delta p$ ) et le processus continue jusqu'à  $\Delta p$  soit en dessous d'un certain seuil. Lorsque ce seuil ou ce critère d'arrêt est obtenu, la boucle itérative s'arrête et la dernière configuration articulaire  $q_t$  résout le problème.

Une amélioration à cette approche consiste à considérer deux cas, en fonction du nombre de degrés de liberté de la structure robotique considérée : si elle n'est pas redondante, on utilise  $J^{-1}$  tandis que la pseudo-inverse de la Jacobienne  $J^\dagger$  [Siciliano *et al.*, 2009] est considérée dans le cas contraire. Pour ce qui est de Baxter, ces deux bras possèdent chacun sept degrés de liberté, ce qui lui confère un profil de robot redondant et donc le calcul de la vitesse des joints est donné à travers les relations suivantes :

$$\dot{q} = J^\dagger(q)v_e \quad (4.20)$$

$$J^\dagger = J^T(JJ^T)^{-1} \quad (4.21)$$

où  $J$  et  $J^\dagger$  sont respectivement la matrice Jacobienne et sa pseudo-inverse,  $q$  est la valeur articulaire des joints,  $\dot{q}$  est la vitesse des joints et  $v_e$  est défini dans l'équation 4.18.

Cependant, cette relation ne tient pas compte des singularités du robot, ce qui peut devenir problématique dans certaines circonstances. Pour passer outre ce problème, le calcul de la  $J^\dagger$  est perfectionné en utilisant la méthode Levenberg-Marq, plus communément connue sous le nom de *damped least-squares method*. Pour plus de détails sur l'utilisation d'une telle méthode pour le contrôle de robot, le lecteur peut se référer à [Wampler II et Leifer, 1988]. Le calcul de la pseudo-inverse devient alors :

$$J^\dagger = J^T(JJ^T + k^2 Id)^{-1} \quad (4.22)$$

où  $k$  est le facteur d'amortissement pour le calcul d'inversion de la Jacobienne et  $Id$  est la matrice identité.

### 4.3.3 Filtrage des commandes

Lors des premières expériences menées sur le robot, des oscillations ont été observées sur les joints de Baxter. En effet, lorsque le contrôleur haptique est utilisé pour envoyer une commande en position et en orientation constante, on observe de légères vibrations sur les moteurs des bras des Falcons, qui se traduisent par de plus grandes oscillations sur le bras du robot contrôlé. En outre, le transfert des coordonnées dans l'espace accessible par le contrôleur vers l'espace de travail du robot se fait par association linéaire (mapping direct) :

$$P_{Baxter} = \frac{P_{Falcon} - min_{Baxter}}{max_{Baxter} - min_{Baxter}}(max_{Falcon} - min_{Falcon}) + min_{Falcon} \quad (4.23)$$

Or, comme la zone accessible par le contrôleur est nettement plus petite que celle du robot, une petite oscillation de l'outil haptique aura une plus grande répercussion sur le bras de Baxter. D'où la nécessité de filtrer ce signal pour éviter de faire face à un tel phénomène. De plus, si on se place dans le contexte d'une application d'interaction homme-robot qui nécessite un minimum de précision, ces secousses peuvent vite devenir des inconvénients non négligeables.

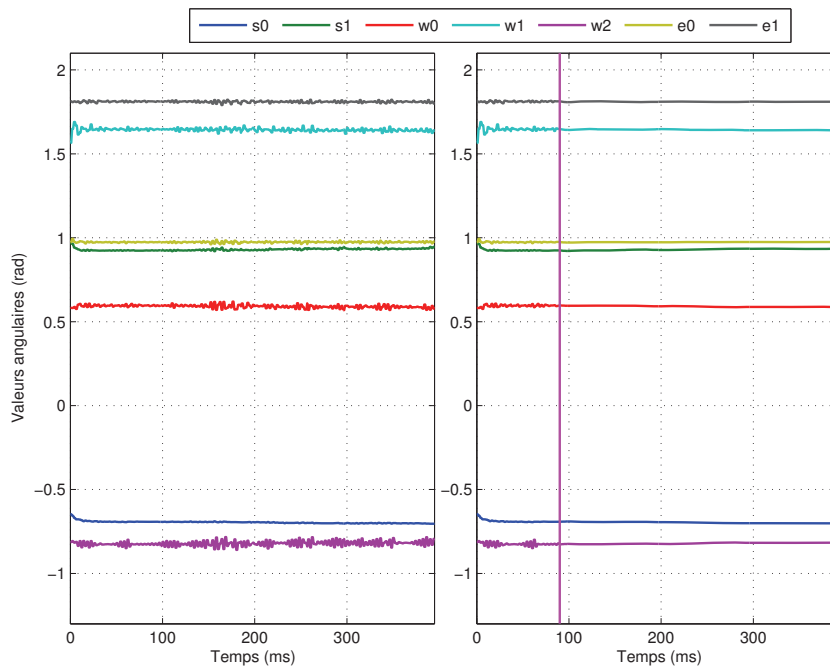


Figure 4.8 Le graphique de gauche représente la trajectoire des joints (s0, s1, w0, w1, w2, e0 et e1) lorsque les commandes ne sont pas filtrées. À droite un filtre numérique d'ordre 4 est appliqué sur les mêmes données, à partir de 90ms (trait vertical magenta)

Pour remédier à cela, un dispositif de filtre passe-bas est installé en sortie du contrôleur et sert à réduire l'amplitude des composantes de fréquences supérieures à la celle de la fréquence de coupure. Notre choix s'est orienté vers un filtre linéaire de Butterworth, qui présente l'avantage d'être facile à calculer, possède un gain constant dans sa bande passante et est très bien adapté pour l'analyse des mouvements. L'ordre du filtre est déterminé par essai-erreur grâce aux fonctions de *Matlab*. De même, on fixe la fréquence de coupure à 5Hz et l'ordre minimum du filtre numérique permettant d'avoir de bons résultats est quatre.

Les résultats sont présentés sur la figure 4.8 où l'on peut s'apercevoir que les oscillations pour chacune des sept articulations ( $s_0$ ,  $s_1$ ,  $w_0$ ,  $w_1$ ,  $w_2$ ,  $e_0$  et  $e_1$ ) du robot disparaissent après le filtrage. Une légère modification a été apportée au filtre afin de le rendre plus en accord avec son utilisation : comme il s'agit d'un filtre d'ordre quatre, les quatre valeurs précédant un instant  $t$  sont utilisées pour filtrer le signal à ce moment précis ; ainsi pour éviter d'avoir un filtre dont les premières valeurs partiraient de zéro, on considère que le passe-bas devient actif après la quatrième itération.

#### 4.3.4 La rétroaction haptique du contrôleur à cinq DDL

Les applications qui utilisent des outils haptiques cherchent à doter l'utilisateur d'informations supplémentaires, souvent dans le but d'être combinées avec d'autres données comme les retours visuels, trop limités pour permettre un contrôle de bonne qualité. Ces données additionnelles peuvent servir d'indicateur soit pour anticiper soit pour réajuster un mouvement. D'autre part, les appareils haptiques sont conçus de manière à être contrôlés de façon la plus intuitive possible, afin de faciliter la prise en main par l'utilisateur.

L'application d'IHR imaginée dans cette étude aspire à commander le bras du robot Baxter grâce au contrôleur haptique afin d'attraper un objet cubique, déposé sur une surface dure (table). Pour éviter de trop forcer voire d'abimer les moteurs associés à chaque articulation du bras robotique, il est nécessaire de savoir quand la pince, attachée à l'organe terminal, va rentrer en contact avec la surface. C'est là qu'entre en jeu le bloc de rétroaction haptique, qui va recevoir une force de rétroaction lorsque les couples mesurés sur la pince dépassent un certain seuil, qui sera déterminée par essai-erreur. En procédant ainsi, on s'assure de la détection des collisions dans l'environnement du robot.

Nous avons fait le choix de configurer le dispositif haptique de manière à favoriser un contrôle vertical. Ce choix, déjà justifié, implique que la pince du bras contrôlé pointe vers le bas. Il est vrai que le bras pourrait venir attraper le cube horizontalement, mais cela impliquerait l'autre configuration du stylet (c.f section 3.3.1). De plus, il a été évoqué dans

la section 4.2, que les rotations du stylet autour des trois axes  $x$ ,  $y$  et  $z$  sont limitées, ce qui engendre les mêmes contraintes pour les mouvements de la pince du robot. On cherche donc à rendre conforme l'orientation de l'effecteur du robot avec la structure mécanique du contrôleur haptique. Une transformation permet d'associer à une position verticale du stylet, une position de l'effecteur de Baxter pointant vers le bas, ce qui se traduit par la matrice de rotation suivante :

$$R = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0.0008 & -1 \\ 0 & -1 & -0.0008 \end{pmatrix} \quad (4.24)$$

En ce qui nous concerne, seul l'effort vertical du bras sera considéré. Comme le but est de détecter quand un contact entre la table et la pince (afin d'indiquer à l'utilisateur qu'il peut fermer la pince pour attraper le cube) se produira et comme l'ensemble des mouvements se fera verticalement, nous avons choisi de nous intéresser uniquement à l'effort vertical. La détection des efforts selon les autres directions pourra s'intégrer dans un prochain travail, mais l'effort vertical reste suffisant pour notre application IHR.

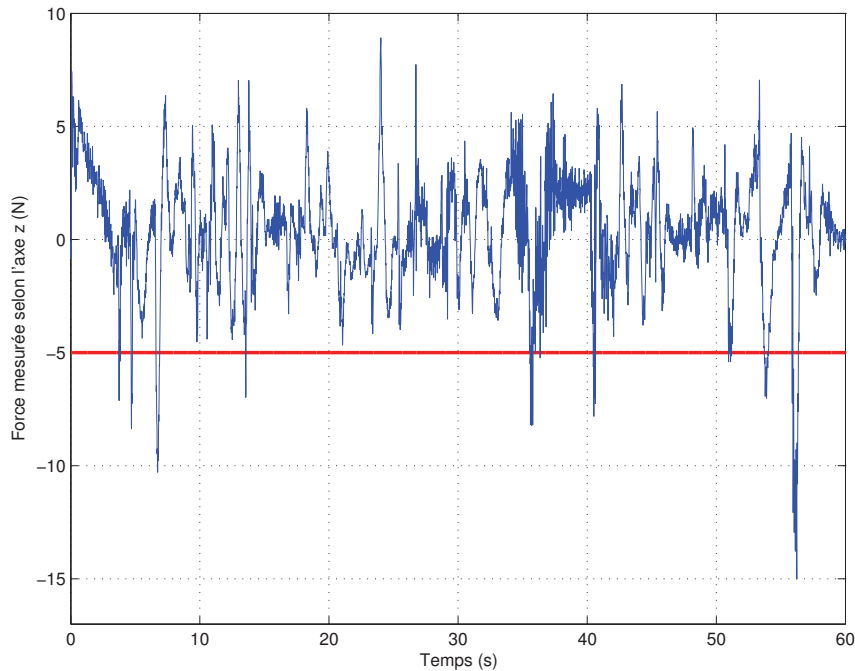


Figure 4.9 Seuil de détection de contact/collision

Le seuil de détection de collisions doit maintenant être caractérisé. Après plusieurs mesures réalisées sur le robot, le seuil est fixé et détermine le moment à partir duquel le couple mesuré verticalement correspond à un contact possible avec un obstacle présent dans



l'environnement du robot (i.e la table). Le graphique 4.9 montre les résultats obtenus : lorsque les forces appliquées sur l'effecteur dépassent  $-5N$ , un facteur force commence progressivement à être appliqué aux moteurs des Falcons. Ce facteur force suit un modèle d'interpolation cubique (c.f. figure 4.10), construit à partir des données issues des mesures réalisées sur le cobot.

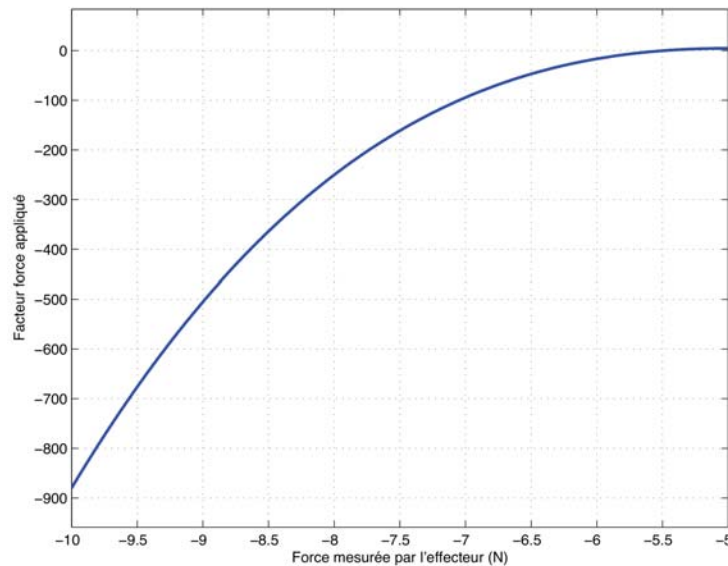


Figure 4.10 Interpolation polynomiale pour le facteur force appliqué au contrôleur haptique

La transmission des forces aux Falcons n'a demandé aucun développement particulier puisqu'une fonction déjà implantée sous la librairie libnifalcon permet de traduire les forces en couples moteurs. Le modèle d'interpolation prévoit une augmentation de la force appliquée sur le stylet haptique proportionnelle à la force de contact mesurée aux pinces de Baxter. Néanmoins, l'outil haptique Falcon affiche des limitations de 8,9N pour les forces applicables sur les moteurs. Ainsi, par souci de sécurité et pour éviter de briser les moteurs, on limite le facteur force afin de garder une marge de sécurité avant d'atteindre les valeurs extrêmes. D'autre part, le modèle a été défini de manière à ce que le stylet se bloque lorsque la force mesurée par l'organe terminal dépasse 8N : en y associant un facteur force déterminé par essai-erreur, lorsqu'une force supérieure à 8N est détectée, le stylet empêche tout mouvement vers le bas, mais autorise les déplacements latéraux et ceux vers le haut.

La phase expérimentale consiste à mettre en situation le bras de Baxter afin que l'utilisateur ressente les forces appliquées sur le contrôleur haptique. Pour ce faire, une table est

placée devant le robot et l'utilisateur doit amener le bras à entrer en collision avec la table (c.f figure 4.11). Les différents points ci-dessous sont constatés :

- lorsque l'utilisateur bouge le stylet trop rapidement vers le bas, la force se fait ressentir correctement,
- lorsqu'un contact est détecté (i.e force inférieure ou égale à 8N), le stylet se bloque et l'utilisateur ne peut plus continuer son mouvement vers le bas ; il peut cependant le bouger dans toutes les autres directions,
- de légères interférences sont perceptibles à cause de la forme saccadée des forces mesurées ; comme on peut le constater sur la figure 4.9, le signal reçu n'est pas lisse ce qui vient causer des forces trop brusques sur le stylet et ainsi dévier légèrement la trajectoire du bras. Une solution aurait été de filtrer ce signal, mais le risque de ne pas détecter une collision est trop important pour adopter cette démarche.

Les résultats auraient probablement été meilleurs avec un outil plus précis et donc plus couteux, mais compte tenu des caractéristiques du contrôleur utilisé, l'efficacité du système est relativement bonne. Par exemple, le ressenti des forces de rétroaction aurait pu être plus doux et plus fluide avec un système plus sensible et performant, alors que notre système réagit un peu trop brusquement dans certaines situations.

### 4.3.5 Téléopération du robot

Finalement pour valider notre contrôleur haptique, une application de téléopération sur le robot Baxter est mise en oeuvre. La revue de littérature a inspiré ce scénario dans lequel on combine les informations du contrôleur avec celles des retours visuels du robot, de sorte que l'utilisateur puisse contrôler le bras robotique uniquement avec ces données. Dans cette expérience l'opérateur et le robot sont placés dans la même pièce (figure 4.11). Pour le bon fonctionnement de l'expérience, l'opérateur joue le jeu de ne pas regarder les déplacements du robot, mais uniquement son écran d'ordinateur. De même, une autre expérience où le robot serait séparé physiquement de l'utilisateur est tout à fait envisageable.

L'opérateur utilise les deux Falcons pour contrôler le bras droit du robot collaboratif Baxter, tandis que le bras gauche suit le mouvement du droit en gardant une position relative à celui-ci. Cette approche permet d'obtenir deux angles de vue différents, chacun provenant des caméras fixées aux extrémités des bras du robot :

- le premier retour visuel (celui du bras contrôlé) renseigne l'utilisateur sur la hauteur à laquelle se situe le bras ; les limitations d'orientation du stylet (c.f section 4.2)

imposent que l'effecteur du robot pointe toujours vers le bas. Grâce aux images issues de la caméra, l'opérateur pourra se placer au-dessus de l'objet à ramasser tout en évaluant la distance qui sépare le poignet de ce même objet,

- on utilise une matrice de transformation  $T$  (4.25) pour la gestion de suivi de trajectoire du bras gauche. Pour avoir un champ de vision suffisamment large, le bras suiveur respecte une distance de 50 cm avec l'autre bras, et conserve une orientation définie sur la figure 4.12 par rapport au repère de référence (celui du poignet droit). Les fonctionnalités de ROS, plus précisément du paquet *TF Transform*, ont grandement facilité le travail, même s'il est nécessaire de recalculer à chaque itération la cinématique inverse pour le bras gauche. Ainsi, l'opérateur obtient un deuxième rendu visuel sur l'objet, ce qui lui octroie un meilleur champ de contrôle.

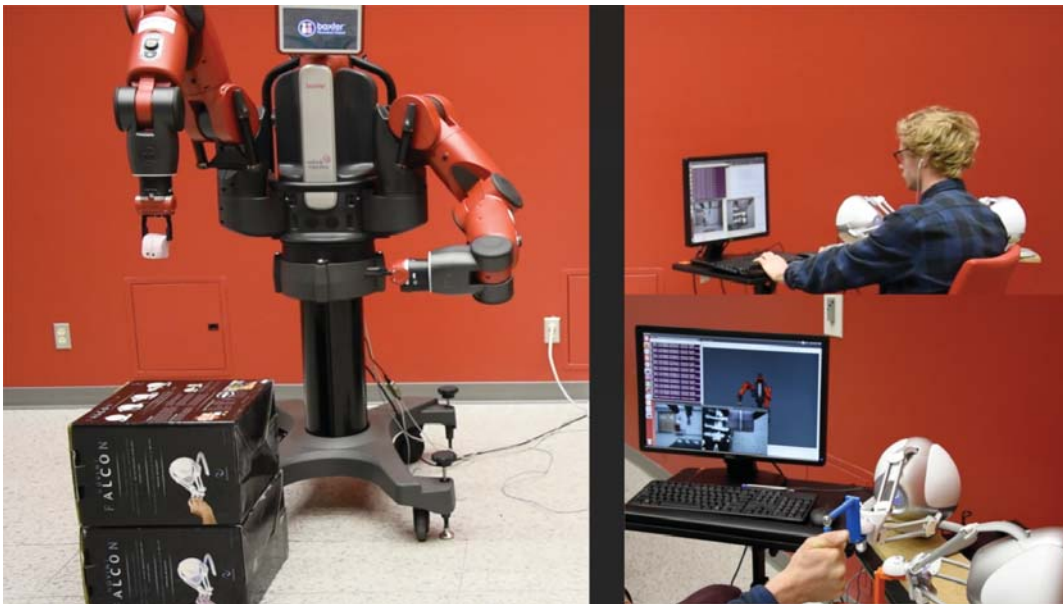


Figure 4.11 Scénario de téléopération à distance

Les résultats s'avèrent être concluants puisqu'il est possible pour un utilisateur novice d'attraper l'objet assez rapidement : en moyenne un novice mettra une minute et trente secondes.

D'autre part, si les données de contrôle composées du retour haptique (qui détecte lorsque la pince touche la table sur laquelle est déposé le cube) et des deux retours visuels issus des caméras ne suffisent pas, l'utilisateur peut s'aider du visualiseur RVIZ (c.f figure 4.13) afin d'observer l'évolution du robot. En effet, il s'agit d'un outil de visualisation 3D, fonctionnant avec ROS, qui donne la possibilité d'observer la configuration, en temps réel, du robot Baxter et de ces données sensorielles (données des caméras, mesures des distances

infrarouges, données des sonars, etc.). Une fois que l'opérateur a réalisé plusieurs essais, celui-ci devient plus expérimenté avec l'outil haptique et la prise de l'objet se fait plus efficacement : le temps est réduit à quarante-cinq secondes.

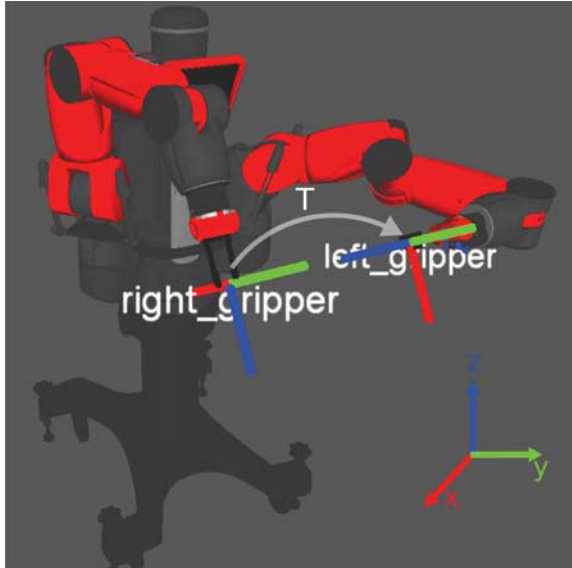


Figure 4.12 Transformation entre le bras contrôlé et le bras suiveur.

$$T = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0.5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.25)$$

Alors que l'outil haptique permet à un utilisateur d'attraper le cube placé sur une table, les forces de rétroactions se font ressentir lorsqu'il y a collision entre les pinces et la table. De plus, la prise en main de cet appareil de contrôle est facile, selon les retours des différents utilisateurs. Le stylet est donc une bonne solution pour le contrôle robotique et ses propriétés sont adaptées aux interactions homme-robot. On notera toutefois que l'utilisateur doit faire preuve d'un minimum de dextérité afin d'être capable d'attraper un objet placé à un endroit spécifique de l'espace. Effectivement, l'espace de travail de Baxter (c.f figure 4.14) est beaucoup plus grand que celui du contrôleur haptique : on parle d'environ  $2m^3$  pour le premier, alors que celui du joystick n'est que de  $12,5cm^3$ . Ainsi un petit déplacement du stylet aura un impact beaucoup plus important sur le mouvement du bras de Baxter, c'est pourquoi le contrôle doit se faire tout en douceur avec des mouvements fluides et lents.

Lorsque les mouvements de l'utilisateur sont trop rapides, l'algorithme quadratique de cinématique inverse TRAC-IK, exploité pour cette expérience, ne trouve pas systématiquement des solutions dans un temps acceptable. En effet, comme nous l'avons vu, cette méthode optimisée se base sur les solutions précédemment trouvées pour accélérer le processus de calcul d'une nouvelle position. Or, on constate dans certaines situations que

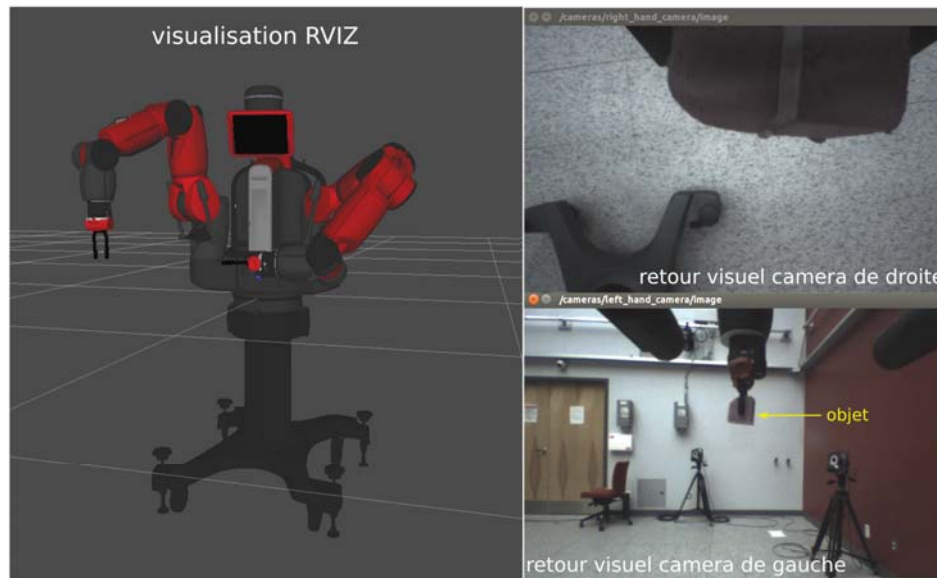


Figure 4.13 À droite, retour visuel de la configuration temps réel du robot collaboratif. À gauche, les retours visuels des caméras disposées aux extrémités de chacun des bras de Baxter

le bras de Baxter reste quelques secondes dans la même position, avant de continuer sa trajectoire. Ce phénomène est observable lorsque la nouvelle position à trouver est trop éloignée par rapport à la dernière configuration déterminée, ou bien lorsque l'algorithme n'a pas réussi à trouver des solutions depuis quelques itérations. Cependant, les résultats des expériences ont montré que si l'utilisateur maintient un mouvement fluide, continu et sans faire des déplacements trop rapides, ce léger désagrément n'est pas observé. Une solution à ce problème serait d'implanter un autre algorithme pour le calcul de la cinématique inverse.

Toujours dans ce même scénario, on peut observer le programme d'évitement d'obstacle, intégré sur le robot Baxter. Cette situation peut être rencontrée lorsque le bras commandé se rapproche trop du bras suiveur qui, compte tenu de ces contraintes en position et en orientation, ne peut plus se déplacer. Dans ce contexte, deux solutions sont envisageables :

- si le contact entre les deux bras est inévitable, un des deux bras (aléatoire) recule pour éviter la collision,
- si la collision est détectée suffisamment tôt, une trajectoire sans collision est générée automatiquement et prend le dessus sur les commandes préalablement envoyées, permettant ainsi d'éviter un contact qui pourrait causer des dégâts au robot.

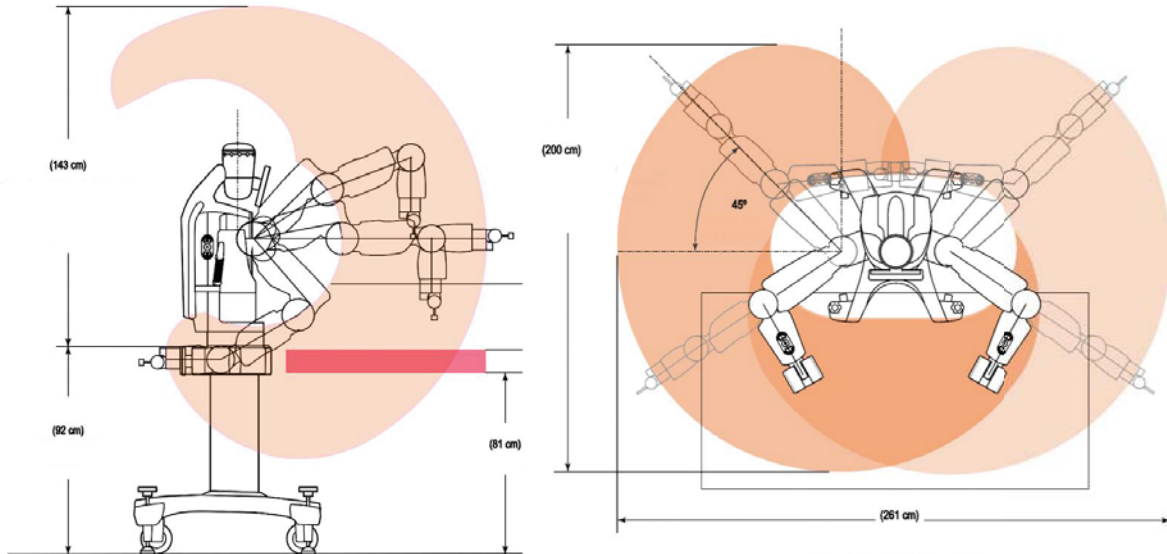


Figure 4.14 Aperçu de l'espace de travail accessible par Baxter [Rethink Robotics, 2012]

Avec leurs sept degrés de liberté, les deux bras pourraient également entrer en collision avec la tête voire même le torse du robot. Rethink Robotics a opté pour une solution de force rétroactive, possible grâce aux actionneurs série élastique : lors d'un contact avec un élément externe, plus le bras se rapproche d'une collision qui pourrait endommager certaines parties du cobot, plus les moteurs du robot forcent graduellement pour éviter le contact. Effectivement, c'est dans cette situation que les moteurs contrôlés par des actionneurs série élastique, interviennent. Les actionneurs équipés de ressorts sont capables d'amortir les chocs et donc d'éviter d'abimer des parties du robot, ainsi que de mesurer la force appliquée sur chacun des moteurs : si cette force dépasse un certain seuil, le bras de Baxter interrompt sa trajectoire jusqu'à ce que la force, qui a enclenché la collision, s'estompe. Le cobot peut ensuite poursuivre son mouvement normalement. Ce robot a été pensé et étudié pour collaborer avec des humains, il ne représente donc aucun danger pour ses coéquipiers.

# CHAPITRE 5

## CONCLUSION

Ces dernières années, les technologies de la robotique se sont diversifiées pour répondre aux besoins d'un contexte socio-économique en perpétuelle évolution. La place des robots dans notre quotidien n'a jamais été aussi omniprésente que récemment, et cette tendance tend à s'accroître avec des machines et des humains de plus en plus connectés les uns avec les autres. Au cœur de cette branche de la robotique, repose les interactions homme-robot qui nécessitent l'exploration et le développement de nouvelles techniques de contrôle, de communication et de sécurité. Parmi les outils considérés, la technologie haptique qui désigne la science du toucher, répond parfaitement à l'émergence de cette approche multidisciplinaire aux frontières entre la cognitive, la biomécanique et la robotique.

Dans ce projet de recherche, une plateforme logicielle de contrôle basée sur la technologie haptique a été conçue et développée pour répondre aux besoins des interactions homme-robot. Elle comprend un appareil haptique, construit à partir de deux joysticks Falcons connectés entre eux par un stylet reconfigurable. Cette technique permet d'augmenter le nombre de degrés de liberté du dispositif tout en limitant les dépenses liées à cette modification. L'objectif étant d'utiliser ce dispositif pour contrôler un robot collaboratif du nom de Baxter, un contrôleur composé des plusieurs blocs ROS a été développé à cet effet. Il récupère la position et l'orientation du stylet pour les transformer en configurations angulaires, grâce à un algorithme optimisé de cinématique inverse, TRAC-IK. Ces configurations sont ensuite envoyées comme commandes au cobot, qui peut alors exécuter le mouvement. Finalement, un bloc de rétroaction haptique a été intégré au système. Celui-ci assiste l'utilisateur dans son contrôle : il se traduit par une force ressentie sur le stylet, lorsque les pinces du robot entrent en contact avec un élément présent dans son environnement.

Les objectifs ont été atteints puisque le bras du robot Baxter a été contrôlé avec succès grâce au dispositif haptique. Les expériences montrent qu'un dispositif haptique open source peut très bien être utilisé pour un contrôle robotique. Sa prise en main est très intuitive et l'utilisateur acquiert rapidement une aisance et une facilité d'utilisation. Cette interface a fait ses preuves dans un scénario de téléopération : en peu de temps, un utilisateur est capable de prendre le contrôle d'un bras robotique, tout en ressentant les forces

de la rétroaction haptique. Cependant, dans certaines situations, ces forces se sont avérées saccadées et altèrent la qualité du contrôle.

Une solution envisageable consisterait à remplacer la force de contact par une force qui estimerait la distance entre l'organe terminal du robot et une collision. Baxter est effectivement muni de capteurs infrarouges qui pourraient être configurés afin d'évaluer cette distance.

D'autre part, la projection adaptative est une méthode qui réduit ou augmente l'espace de travail en fonction du degré de précision souhaité. Son but est d'adapter la projection entre l'espace accessible par le contrôleur et l'espace de travail du robot. Cette alternative pourrait faire l'objet de travaux futurs. En particulier, elle permettrait à l'utilisateur d'adapter l'accélération du robot en fonction des zones plus restreintes où davantage de précision de contrôle serait requise. Elle pourrait également contrecarrer la limitation de l'espace de travail du contrôleur et améliorer le contrôle, notamment en augmentant la précision du système.



# LISTE DES RÉFÉRENCES

- 3DSYSTEMS (2013). <http://www.geomagic.com/en/products/phantom-desktop/overview>.
- Albraikan, A., Badawi, H., Hamam, A. et Saddik, A. E. (2013). Haptibasic : Learning basic concepts of a haptic technology through edutainment games. Dans *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. p. 1–4.
- Aldebaran Robotics (2015). <https://www.aldebaran.com/fr/cool-robots/pepper>.
- Asimov, I. (1950). *I, Robot*. Gnome Press.
- Bae, B., Koo, T., Park, K. et Kim, Y. (2001). Design and control of a two degree of freedom haptic device for the application of pc video games. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*. volume 3. p. 1738–1743 vol.3.
- Beeson, P. et Ames, B. (2015). TRAC-IK : An open-source library for improved solving of generic inverse kinematics. Dans *Proceedings of the IEEE RAS Humanoids Conference*.
- Cheng, L., Sun, Q., Su, H., Cong, Y. et Zhao, S. (2012). Design and implementation of human-robot interactive demonstration system based on kinect. Dans *24th Chinese Control and Decision Conference (CCDC)*. p. 971–975.
- D’Auria, D., Persia, F. et Siciliano, B. (2015). A low-cost haptic system for wrist rehabilitation. Dans *IEEE International Conference on Information Reuse and Integration (IRI)*. p. 491–495.
- Diftler, M. A., Culbert, C. J., Ambrose, R. O., Platt, R. et Bluethmann, W. J. (2003). Evolution of the nasa/darpa robonaut control system. Dans *IEEE International Conference on Robotics and Automation*. volume 2. p. 2543–2548 vol.2.
- Esteban, G., Fernández, C., Conde, M. A. et Lera, F. J. R. (2014). Implementation of a haptic simulation environment for surgical learning systems. Dans *International Symposium on Computers in Education (SIIE)*. p. 81–86.
- Estes, L. T., Backus, D. et Starner, T. (2015). A wearable vibration glove for improving hand sensation in persons with spinal cord injury using passive haptic rehabilitation. Dans *9th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. p. 37–44.
- Fang, B., Guo, D., Sun, F., Liu, H. et Wu, Y. (2015). A robotic hand-arm teleoperation system using human arm/hand with a novel data glove. Dans *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. p. 2483–2488.
- Ferreau, H., Bock, H. et Diehl, M. (2008). An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, volume 18, numéro 8, p. 816–830.

- Ferreau, H., Kirches, C., Potschka, A., Bock, H. et Diehl, M. (2014). qpOASES : A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, volume 6, numéro 4, p. 327–363.
- Fletcher, J., Fountain, C., Kerley, C., Torres, J., Barott, W. C., Salamah, S. et Stansbury, R. (2012). Immersive audio-visual system for an unmanned robotic vehicle in a senior design project. Dans *Proceedings of IEEE Southeastcon*. p. 1–5.
- Foote, T. (2013). tf : The transform library. Dans *IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*,. Open-Source Software workshop, p. 1–6.
- Gambaro, E. F., Zollo, L. et Guglielmelli, E. (2014). Development and experimental validation of a master interface with vibrotactile feedback for robotic telesurgery. Dans *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechanics*. p. 726–731.
- Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J. et Maisonnier, B. (2009). Mechatronic design of nao humanoid. Dans *IEEE International Conference on Robotics and Automation (ICRA)*. p. 769–774.
- Greenwald, D., Cao, C. G. L. et Bushnell, E. W. (2012). Haptic detection of artificial tumors by hand and with a tool in a mis environment. *IEEE Transactions on Haptics*, volume 5, numéro 2, p. 131–138.
- Guerrero, C. R., Marinero, J. F., Turiel, J. P. et Farina, P. R. (2012). Using psychophysiological feedback to enhance physical human robot interaction in a cooperative scenario. Dans *4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob)*. p. 1176–1181.
- Hashimoto, T., Kobayashi, H., Polishuk, A. et Verner, I. (2013). Elementary science lesson delivered by robot. Dans *8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. p. 133–134.
- IFR (2015). World robotics service robot.
- Ishiguro, H. (2011). <http://www.geminoid.jp/en/robots.html>.
- Ismail, L., Shamsuddin, S., Yussof, H., Hashim, H., Bahari, S., Jaafar, A. et Zahari, I. (2011). Face detection technique of humanoid robot nao for application in robotic assistive therapy. Dans *IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. p. 517–521.
- Jindai, M., Watanabe, T., Shibata, S. et Yamamoto, T. (2006). Development of a handshake robot system for embodied interaction with humans. Dans *RO-MAN - The 15th IEEE International Symposium on Robot and Human Interactive Communication*. p. 710–715.
- Kim, J. H., Thang, N. D. et Kim, T. S. (2009). 3-d hand motion tracking and gesture recognition using a data glove. Dans *IEEE International Symposium on Industrial Electronics*. p. 1013–1018.

- Kim, Y. M., Koo, S. Y., Lim, J. G. et Kwon, D. S. (2010). A robust online touch pattern recognition for dynamic human-robot interaction. *IEEE Transactions on Consumer Electronics*, volume 56, numéro 3, p. 1979–1987.
- Koceski, S. et Koceska, N. (2010). Vision-based gesture recognition for human-computer interaction and mobile robot’s freight ramp control. Dans *32nd International Conference on Information Technology Interfaces (ITI)*. p. 289–294.
- Kretz, A., Huber, R. et Fjeld, M. (2005). Force feedback slider (ffs) : interactive device for learning system dynamics. Dans *5th IEEE International Conference on Advanced Learning Technologies (ICALT’05)*. p. 457–458.
- Krückel, K., Nolden, F., Ferrein, A. et Scholl, I. (2015). Intuitive visual teleoperation for ugvs using free-look augmented reality displays. Dans *IEEE International Conference on Robotics and Automation (ICRA)*. p. 4412–4417.
- Lavatelli, A., Ferrise, F. et Bordegoni, M. (2014). Design of an open-source low cost 2dof haptic device. Dans *IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. p. 1–6.
- Li, H., Li, B. et Xu, W. (2015a). Development of a remote-controlled mobile robot with binocular vision for environment monitoring. Dans *IEEE International Conference on Information and Automation*. p. 737–742.
- Li, Z., Feiling, J., Ren, H. et Yu, H. (2015b). A novel tele-operated flexible surgical arm with optimal trajectory tracking aiming for minimally invasive neurosurgery. Dans *IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. p. 239–244.
- Lin, H. I. et Chou, C. C. (2015). Humanoid robot motion imitation using kinect. Dans *International Conference on Advanced Robotics and Intelligent Systems (ARIS)*. p. 1–4.
- Liu, H. et Li, X. (2010). A selection method of speech vocabulary for human-robot speech interaction. Dans *IEEE International Conference on Systems Man and Cybernetics (SMC)*. p. 2243–2248.
- Machulis, K. (2007). Libnifalcon a cross platform driver for the novint falcon. <http://qdot.github.io/libnifalcon/>.
- Martin, S. et Hillier, N. (2009). Characterisation of the novint falcon haptic device for application as a robot manipulator. Dans *Australasian Conference on Robotics and Automation (ACRA)*.
- Metta, G., Sandini, G., Vernon, D., Natale, L. et Nori, F. (2008). The icub humanoid robot : An open platform for research in embodied cognition. Dans *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*. PerMIS ’08. ACM, New York, NY, USA, p. 50–56.
- Michelotti, M. B. (2011). *Application of the novint falcon haptic device as an actuator in real-time control*. Thèse de doctorat, University of Illinois.

- Modares, H., Ranatunga, I., Lewis, F. L. et Popa, D. O. (2016). Optimized assistive human-robot interaction using reinforcement learning. *IEEE Transactions on Cybernetics*, volume 46, numéro 3, p. 655–667.
- Morito, C., Shimono, T., Motoi, N., Fujimoto, Y., Tsuji, T., Hasegawa, Y., Abe, K., Sakurai, Y. et Ishii, S. (2013). Development of a haptic bilateral interface for arm self-rehabilitation. Dans *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. p. 804–809.
- Mortimer, M., Horan, B. et Stojcevski, A. (2014). 4 degree-of-freedom haptic device for surgical simulation. Dans *World Automation Congress (WAC)*. p. 735–740.
- Nair, S., Dean-Leon, E. et Knoll, A. (2011). 3d position based multiple human servoing by low-level-control of 6 dof industrial robot. Dans *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. p. 2816–2823.
- Neurodigital Technologies (2015). <https://www.gloveonevr.com/>.
- Okamura, A. M., Verner, L. N., Yamamoto, T., Gwilliam, J. C. et Griffiths, P. G. (2011). *Surgical Robotics : Systems Applications and Visions*, chapitre Force Feedback and Sensory Substitution for Robot-Assisted Surgery. Springer US, Boston, MA, p. 419–448.
- Ouellet, K. (2008). *Libnifalcon kinematic analysis* (Rapport technique).
- Pacchierotti, E., Christensen, H. I. et Jensfelt, P. (2005). Human-robot embodied interaction in hallway settings : a pilot user study. Dans *14th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*. p. 164–171.
- Pamela J. Hinds, Teresa L. Roberts, H. J. (2004). Whose job is it anyway? a study of human–robot interaction in a collaborative task. Dans *Human-computer Interaction*. volume 19. p. 151–181.
- Pan, Y., Ge, S. S., He, H. et Chen, L. (2009). Real-time face detection for human robot interaction. Dans *RO-MAN - The 18th IEEE International Symposium on Robot and Human Interactive Communication*. p. 1016–1021.
- Paolis, L. T. D., Pulimeno, M. et Aloisio, G. (2007). The simulation of a billiard game using a haptic interface. Dans *11th IEEE International Symposium Distributed Simulation and Real-Time Applications (DS-RT)*. p. 64–67.
- Renon, P., Yang, C., Ma, H. et Cui, R. (2013). Haptic interaction between human and virtual icub robot using novint falcon with chai3d and matlab. Dans *32nd Chinese Control Conference (CCC)*. p. 6045–6050.
- Rethink Robotics (2012). <http://www.rethinkrobotics.com/baxter/tech-specs/>.
- Rethink Robotics (2015). <http://www.rethinkrobotics.com/sawyer-intera-3/>.
- ReWalk Robotics (2014). <http://rewalk.com/rewalk-personal-3/>.

- Rose, C. G., French, J. A. et O'Malley, M. K. (2014). Design and characterization of a haptic paddle for dynamics education. Dans *IEEE Haptics Symposium (HAPTICS)*. p. 265–270.
- Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N. et Fujimura, K. (2002a). The intelligent asimo : system overview and integration. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*. volume 3. p. 2478–2483 vol.3.
- Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N. et Fujimura, K. (2002b). The intelligent asimo : system overview and integration. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*. volume 3. p. 2478–2483 vol.3.
- Sarakoglou, I., Garcia-Hernandez, N., Tsagarakis, N. G. et Caldwell, D. G. (2012). Integration of a tactile display in teleoperation of a soft robotic finger using model based tactile feedback. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 46–51.
- Schoor, W., Förster, M. et Radetzky, A. (2012). Realistic training simulations of explosive ordnance disposal and improvised explosive device disposal robots. Dans *IEEE 10th International Conference on Industrial Informatics*. p. 875–880.
- Shah, A. V., Teuscher, S., McClain, E. W. et Abbott, J. J. (2010). How to build an inexpensive 5-dof haptic device using two novint falcons. Dans *Proceedings of the International Conference on Haptics : Generating and Perceiving Tangible Sensations, Part I*. p. 136–143.
- Siciliano, B., Sciavicco, L., Villania, L. et Oriolo, G. (2009). *Robotics : Modeling, Planning, and Control*. Springer.
- Sivak, M., Unluhisarcikli, O., Weinberg, B., Mirelman-Harari, A., Bonato, P. et Mavroidis, C. (2010). Haptic system for hand rehabilitation integrating an interactive game with an advanced robotic device. Dans *IEEE Haptics Symposium*. p. 475–481.
- Stamper, R. E. (1997). *A Three Degree Of Freedom Parallel Manipulator With Only Translational Degrees Of Freedom*. Thèse de doctorat, University of Maryland, Department of Mechanical Engineering and Institute for Systems Research.
- Sun, L. W. et Yeung, C. K. (2007). Port placement and pose selection of the da vinci surgical system for collision-free intervention based on performance optimization. Dans *IEEE/RSJ International Conference on Intelligent Robots and Systems*. p. 1951–1956.
- Tan, H. et Kawamura, K. (2015). Generation of acceptable actions using imitation learning, intention recognition, and cognitive control. Dans *24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. p. 389–393.
- Tsagarakis, N., Metta, G., Sandini, G., Vernon, D., Beira, R., Becchi, F., Righetti, L., Santos-Victor, J., Ijspeert, A., Carrozza, M. et Caldwell, D. (2007). icub - the design and realization of an open humanoid platform for cognitive and neuroscience research.

- Journal of Advanced Robotics, Special Issue on Robotic platforms for Research in Neuroscience*, volume 21, numéro 10, p. 1151–1175.
- Tunstel, E., Maimone, M., Trebi-Ollennu, A., Yen, J., Petras, R. et Willson, R. (2005). Mars exploration rover mobility and robotic arm operational performance. Dans *IEEE International Conference on Systems, Man and Cybernetics*. volume 2. p. 1807–1814 Vol. 2.
- Vanacken, L., De Boeck, J. et Coninx, K. (2010). The phantom versus the falcon : Force feedback magnitude effects on user’s performance during target acquisition. Dans *Proceedings of the 5th International Conference on Haptic and Audio Interaction Design. HAID’10*. Springer-Verlag, Berlin, Heidelberg, p. 179–188.
- Vandevelde, C., Saldien, J., Ciocci, M.-C. et Vanderborght, B. (2013). *Systems Overview of Ono*. Springer International Publishing, Cham, p. 311–320.
- Vernon, D., Metta, G. et Sandini, G. (2007). The icub cognitive architecture : Interactive development in a humanoid robot. Dans *IEEE 6th International Conference on Development and Learning*. p. 122–127.
- Vitiello, N., Lenzi, T., Roccella, S., Rossi, S. M. M. D., Cattin, E., Giovacchini, F., Vecchi, F. et Carrozza, M. C. (2013). Neuroexos : A powered elbow exoskeleton for physical rehabilitation. *IEEE Transactions on Robotics*, volume 29, numéro 1, p. 220–235.
- Walker, A. M. et Miller, D. P. (2012). Tele-operated robot control using attitude aware smartphones. Dans *7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. p. 269–270.
- Wampler II et Leifer, L. J. (1988). Applications of Damped Least-Squares Methods to Resolved-Rate and Resolved-Acceleration Control of Manipulators. *Journal of Dynamic Systems Measurement and Control-transactions of The Asme*, volume 110.
- Wang, M., Maeda, Y. et Takahashi, Y. (2012). Human intention recognition via eye tracking based on fuzzy inference. Dans *Joint 6th International Conference on Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS)*. p. 846–851.
- Wang, N., Yang, C., Lyu, M. R. et Li, Z. (2014). An emg enhanced impedance and force control framework for telerobot operation in space. Dans *IEEE Aerospace Conference*. p. 1–10.
- Willaert, B., Bohg, J., Brussel, H. V. et Niemeyer, G. (2012). Towards multi-dof model mediated teleoperation : Using vision to augment feedback. Dans *IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE)*. p. 25–31.
- Williamson, M. M. (1995). *Serie Elastic Actuators*. Thèse de doctorat, University of Oxford, Department of Electrical Engineering and Computer Science.
- Yim, J., Jung, H., Yoo, B., Choi, C., Park, D. et Kim, J. (2015). Rotating your face using multi-task deep neural network. Dans *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. p. 676–684.

Zsombor-Murray, P. (2009). *An Improved Approach to the Kinematics of Clavel's Delta Robot*, Center for Intelligent Machines, McGill University, Montreal, Canada.







