THÈSE POUR OBTENIR LE GRADE DE DOCTEUR
DE L'UNIVERSITE DE MONTPELLIER

En Systèmes Automatiques et Microélectroniques

École doctorale : Information, Structures, Systèmes

Unité de recherche UMR5506

Contribution à la stratégie de commande d'un manipulateur mobile bi-bras pour une manipulation interactive d'objets : application à l'assistance des personnes âgées.

Présentée par NOWAK Jordan
Le 7 juin 2022

Sous la direction de Philippe FRAISSE,
d'Andrea CHERUBINI et de Jean-Pierre DAURÈS

Devant le jury composé de

Christine AZEVEDO, Directrice de Recherche, INRIA-CNRS Montpellier	Présidente du Jury
Rachid ALAMI, Directeur de recherche, LAAS-CNRS Toulouse	Rapporteur
Charles FATTAL, Médecin chef HDR, CRF Bouffard-Vercelli USSAP, Perpignan	Rapporteur
Vincent BONNET, Maître de conférences, LAAS-CNRS Toulouse	Examinateur
Philippe FRAISSE, Professeur des universités, Université de Montpellier	Directeur de Thèse
Andrea CHERUBINI, Maître de conférences, Université de Montpellier	Co-Directeur de Thèse

UNIVERSITÉ DE MONTPELLIER

Remerciements

Je tiens tout d'abord à remercier les professionnels qui m'ont entouré et que j'ai rencontré tout au long de mon parcours. À commencer par la doctoresse Émilie GUETTARD qui m'a offert l'opportunité de réaliser un stage, puis ce doctorat sur un sujet si passionnant qu'est l'assistance robotique pour les personnes âgées. Je la remercie pour son écoute, son dynamisme, sa bonne humeur, sa pédagogie, et pour avoir toujours cru en moi. Je n'oublie pas les kinésithérapeutes, notre secrétaire Nelly et le docteur Serge LORENZO du service de Médecine Physique et Réadaptation pour leur accueil et les moments partagés au sein de la clinique. Un grand merci également aux ergothérapeutes du Pôle Autonomie Santé de Lattes pour leurs précieux conseils, leurs disponibilités, et le temps consacré à tester et donner leurs avis.

Ma reconnaissance va tout naturellement vers les professeurs Philippe FRAISSE et Andrea CHERUBINI de l'Université de Montpellier qui n'ont cessé de croire en moi et qui ont accepté de m'accompagner dans la réalisation de ce projet professionnel. Ils m'ont permis d'avoir un support et le cadre académique nécessaires pour le bon déroulement de la thèse. Je salue également le professeur Jean-Pierre DAURÈS, directeur de l'équipe de recherche de la Clinique Beau Soleil, pour sa bonne humeur, sa détermination, ses conseils pour mener à bien les projets et coordonner les équipes. Dans cette équipe, je remercie Sandy, Nicolas et Françoise pour leur accueil, leurs conseils, leur implication et leur réactivité.

Je remercie également le responsable système d'information de la Clinique, Kévin BIDEAULT, pour son aide quant à la mise en place des ressources informatiques nécessaires pour le bon développement des projets, ainsi qu'au temps qu'il a pu me consacrer pour la gestion administrative.

Je remercie l'ensemble du personnel médical de la cité des aînés de Montpellier et plus particulièrement son directeur, M. Sylvain LEBRIS, pour la mise à disposition des locaux qui m'ont permis de développer et d'expérimenter plus facilement. J'apprécie son implication dans les projets et l'aide qu'il a pu nous apporter en sélectionnant et proposant à certains des résidents de participer à nos expérimentations. Ma reconnaissance va donc aussi tout naturellement aux différents volontaires, dont les résidents de la cité des aînés qui m'ont accordé toute leur confiance durant ces manipulations. Je les remercie pour le temps qu'ils ont pu me consacrer et leurs retours qui nous ont permis d'avancer et d'ajuster nos applications.

Je salue également l'ensemble de mes collègues, chercheurs, doctorants et amis présents au LIRMM pour leur aide et tous leurs conseils (Sonny, Kamal, Firas, Benjamin, Robin, Anastasia, Guillaume...). Ils ont su m'écouter, m'orienter et m'apporter leur soutien quand j'en avais besoin. Je n'oublie pas Rémy et Guilain que j'ai eu l'occasion d'encadrer en projet et en stage durant leurs fins d'études de Master. Ils m'ont apporté une aide précieuse pour l'avancement du projet et m'ont permis de proposer dans les temps une application plus concrète au sein de la cité des aînés. Cette première expé-

rience en tant qu'encadrant m'a beaucoup appris et fut très enrichissante.

L'ensemble de mes proches et de mes amis sont vivement remerciés pour avoir toujours été présents, dans les bons comme les mauvais moments.

Pour finir, j'aimerais écrire quelques lignes aux deux personnes les plus importantes dans ma vie et qui ont été ma plus grande source de motivation.

Tout d'abord ma compagne, qui a toujours cru en moi et a toujours été là pour me réconforter dans les moments les plus difficiles. Elle m'a permis d'avoir constamment plus de force et d'ambition et me donne envie chaque jour de me surpasser. Je lui dois toutes ces années de réussite, elle a toujours soutenu toutes mes démarches, trouvé les mots pour me rassurer, me remotiver, m'écouter, me proposer des idées, me conseiller... Tout cela m'a permis de m'interroger continuellement et de trouver des solutions adaptées afin de répondre à mes nombreuses interrogations. Je la remercie pour sa patience et sa présence rassurante pour pallier mon stress et mes inquiétudes. Je la remercie grandement pour la femme qu'elle est et l'amour sans faille qu'elle me donne chaque jour.

Je souhaite finir ces remerciements par ces quelques lignes pour mon fils Noah. J'ai commencé ce doctorat en étant un jeune homme et aujourd'hui, je finis ces années de doctorat en étant père de ce petit trésor. Il n'y a pas vraiment de mots pour décrire le bonheur qu'il nous procure et la fierté que je ressens à ses côtés. Ces derniers mois ont été assez intenses afin de préparer sa venue et afin de finir ce doctorat. Il me motive à être la meilleure version de moi-même et à me dépasser à chaque instant pour le rendre fier.

Résumé

En raison d'une baisse de la mortalité des enfants de moins d'un an et des personnes âgées de plus de 70 ans, l'espérance de vie augmente. Le nombre de naissances est également beaucoup moins élevé que pendant les années du baby-boom. Ces raisons ont pour conséquence de rendre la population mondiale de plus en plus vieillissante. La majoration de cette population âgée nous met face à de nouvelles problématiques. En effet, leur âge avancé augmente le risque de perte d'autonomie et tend à rendre cette population de plus en plus dépendante. Il est donc essentiel de réfléchir dès à présent à ces enjeux et de trouver des solutions adaptées. Des sociétés proposent dès à présent des robots assistants, pouvant rendre des services et assister des personnes à leurs domiciles ou en institution. Les professionnels de santé s'intéressent également sur l'aide que peut proposer ces nouveaux outils afin d'alléger certaines de leurs tâches répétitives, ou directement pour apporter des services auprès de leurs patients et ainsi compléter leurs prises en charge existantes.

L'objectif de nos travaux de recherche consiste à mettre en place un service d'aide robotisé capable d'apporter un objet du quotidien à une personne âgée au sein d'un EHPAD (Établissement d'Hébergement pour Personnes Âgées Dépendantes) ou au sein même de son domicile. Ce service a pour objectif de réduire les déplacements de la personne afin de limiter ses risques de chute, d'optimiser son environnement de vie, améliorer son confort, et enfin de préserver son autonomie. Nous proposons donc un framework se basant sur le modèle bio-mécanique des membres supérieurs de l'humain et l'effet de la manipulabilité et de la gravité sur la position de transfert. La contribution de nos recherches porte sur la génération d'un espace de transfert adapté à la morphologie et aux préférences de l'utilisateur. Nous introduisons un nouveau paramètre pertinent pour le confort de l'individu, qui est l'indice de gravité terrestre. Il permet de minimiser les efforts exercés sur les articulations du bras de l'humain lors du transfert. Nous nous adaptons aux caractéristiques physiques de l'individu, c'est-à-dire que nous sommes capables de définir une zone de transfert personnelle qui tient compte de son poids, de sa taille, de ses amplitudes articulaires, ainsi que de ses préférences. Enfin, pour apporter des objets au plus près des personnes dépendantes, nous intégrons également la localisation et la navigation du robot au sein d'un EHPAD.

L'ensemble de nos développements ont pu être testés et validés au cours d'expérimentations impliquant des personnes âgées avec le robot humanoïde Pepper. Ces expérimentations ont eu pour objectif d'analyser la pertinence de la zone de transfert générée. Les résultats montrent que les déplacements du robot et la zone atteinte pour le transfert sont satisfaisants pour récupérer l'objet dans de bonnes conditions. Les réponses aux questionnaires proposés pour ces études tendent à faire ressortir une préférence générale envers les zones les plus proches d'elles. C'est-à-dire avec une forte disponibilité de l'objet et donc peu de mouvements à effectuer pour le récupérer. Les personnes présentant des limitations articulaires sont également satisfaites de la position de transfert proposée qui leur semble bien adaptée à leurs conditions.

Abstract

Due to a decline in the mortality of children under the age of one and people over the age of 70, life expectancy is increasing. The number of births is also much lower than during the baby boom years. These reasons have the consequence of making the world's population increasingly aging. The increase in this elderly population brings us new challenges. Indeed, their advanced age increases the risk of loss of autonomy and tends to make this population more and more dependents. It is therefore essential to think about these issues now and find appropriate solutions. Companies are now offering robot assistants which can provide services and help people at home or in institutions. Health professionals are also interested in the help that these new tools can offer in order to lighten some of their repetitive tasks, or directly to provide services to their patients and thus supplement their existing care.

The objective of our research is to set up a robotic assistance service capable of bringing an everyday object to an elderly person in a nursing home or in even from his home. The aim of this service is to reduce the movement of the person in order to limit their risk of falling, to optimise their living environment, to improve their comfort, and finally to preserve their autonomy. We propose a framework based on the bio-mechanical model of the human upper limbs and the effect of manipulability and gravity on the transfer position. The contribution of our research focuses on the generation of a transfer space adapted to the morphology and preferences of the user. We introduce a new relevant parameter for the comfort of the individual, which is the gravity index. It minimises the forces exerted on the joints of the human arm during the transfer. We adapt to the physical characteristics of the individual, i.e., we are able to define a personal transfer zone that takes into account their weight, height, joint amplitudes and their preferences. Finally, in order to bring objects as close as possible to dependent people, we also integrate the location and navigation of the robot within a nursing home.

All of our developments have been tested and validated during experiments involving elderly people with the Pepper robot. These experiments aimed to analyse the relevance of the generated transfer zone. The results show that the robot's movements and the area reached for the transfer are satisfactory for retrieving the object in good conditions. The answers to the questionnaires proposed for these studies tend to show a general preference for the areas closest to them. That is to say with a high availability of the object and therefore few movements required to retrieve it. People with joint limitations are also satisfied with the proposed transfer position which seems well suited to their conditions.

Table des matières

Table des matières 7

Liste des figures 11

1 Introduction 15
1.1 Contexte et motivation . . . 16
1.1.1 Vieillissement de la population . . . 16
1.1.2 Concept de fragilité . . . 18
1.1.3 Motivation . . . 18
1.1.4 Opportunités offertes par la robotique d'assistance . . . 19
1.1.5 La robotique au service des seniors . . . 20
1.1.6 Projet ROBOd'Home . . . 22
1.1.7 Projet à la cité des aînés . . . 24
1.2 Organisation de la thèse . . . 26
1.2.1 Énoncé du problème . . . 26
1.2.2 Contributions . . . 27

2 État de l'art 28
2.1 Introduction . . . 29
2.2 La transmission d'un objet . . . 30
2.3 Phase 1 : Le pré-transfert . . . 31
2.3.1 Stratégie de communication . . . 31
2.3.2 Analyse et saisie de l'objet . . . 32
2.3.3 Estimation du point de transfert . . . 33
2.3.4 Planification et contrôle du mouvement avant contact . . . 35
2.4 Phase 2 : Transfert physique de l'objet . . . 37
2.4.1 Transfert de force de préhension . . . 37
2.5 Minimiser l'effort . . . 38
2.6 Conclusion . . . 39

3 Plateforme expérimentale 40
3.1 Introduction . . . 41
3.2 Plateforme robotique . . . 42
3.2.1 Un robot assistant . . . 42
3.2.2 Capteurs et actionneurs utilisés . . . 42

3.2.3 Station de recharge . . . 44
3.3 Plateforme informatique . . . 45
3.3.1 Version du robot et de son système d'exploitation . . . 45
3.3.2 Kit de développement logiciel . . . 47
3.3.3 Logiciel Choregraphe . . . 47
3.3.4 Capacités informatiques nécessaires . . . 48
3.4 Développement logiciel . . . 49
3.4.1 Contexte et motivation . . . 49
3.4.2 Une Interface de Programmation d'Applications . . . 50
3.4.3 Utilisation du module DCM . . . 51
3.4.4 Développement avec la méthodologie PID . . . 51
3.5 Conclusion . . . 52

4 Modélisation et analyse de l'espace de transfert 53
4.1 Introduction . . . 54
4.2 Espace atteignable . . . 55
4.2.1 Modèle géométrique . . . 55
4.2.2 Génération de l'espace . . . 56
4.2.3 Discrétisation . . . 57
4.2.4 Zone interdite . . . 58
4.3 Espace ergonomique . . . 59
4.3.1 Indice de manipulabilité . . . 59
4.3.2 Indice de gravité terrestre . . . 60
4.3.3 Mise en commun des espaces . . . 62
4.3.4 Enlever les doublons . . . 63
4.3.5 Discussions . . . 64
4.4 Conclusion . . . 65

5 Stratégie de commande pour une interaction physique avec l'humain 66
5.1 Introduction . . . 67
5.2 Estimation de la posture de l'humain . . . 68
5.2.1 Système de caméra RGB-D . . . 68
5.2.2 Principes de stéréovision . . . 69
5.2.3 Mise en correspondance de point . . . 69
5.2.4 Mise en correspondance de points sans paramètre . . . 71
5.2.5 Estimation de la pose avec OpenPose . . . 80
5.2.6 Estimation des rotations du torse . . . 81
5.3 Estimation du point de transfert . . . 86
5.3.1 Quelques ajustements sur l'espace de transfert généré . . . 86
5.3.2 Zone de transfert ergonomique du robot . . . 86
5.3.3 Estimation du point de transfert . . . 87
5.4 Commande cinématique du robot Pepper . . . 89
5.4.1 Modèle du robot . . . 90
5.4.2 Choix des moteurs pour le transfert . . . 91
5.4.3 Saisie d'objet par le robot Pepper . . . 92

5.4.4 Interaction physique et sécurité . . . 93
5.4.5 Synthèse de nos développements . . . 93
5.5 Localisation et navigation du robot . . . 97
5.5.1 Localisation à l'aide de marqueurs . . . 97
5.5.2 Estimation du plus court chemin . . . 99
5.5.3 Commande du déplacement . . . 102
5.6 Conclusion . . . 104

6 Résultats et discussion **105**
6.1 Introduction . . . 106
6.2 Première expérimentation . . . 107
6.2.1 Déroulement de l'étude . . . 107
6.2.2 Analyse des résultats . . . 109
6.2.3 Problèmes rencontrés et solutions apportées . . . 110
6.3 Seconde expérimentation . . . 112
6.3.1 Déroulement de l'étude . . . 112
6.3.2 Analyse des résultats . . . 114
6.4 Conclusion . . . 118

7 Conclusion **119**

A Description du robot Pepper **122**
A.1 Description du robot Pepper . . . 123
A.1.1 Informations utiles fournies par le constructeur . . . 123
A.1.2 Modèle géométrique du robot Pepper . . . 125
A.1.3 Tableau qui définit le modèle géométrique du robot Pepper . . . 126
A.2 Modèle géométrique de l'humain . . . 127
A.2.1 Modèle géométrique des bras de l'humain . . . 127
A.2.2 Tableau qui définit le modèle géométrique des bras de l'humain . . . 128
A.2.3 Expression des transformations homogènes . . . 128

B Différents aperçus **129**
B.1 Application de l'indice de manipulabilité . . . 130
B.2 Application de l'indice de gravité . . . 131

C Expérimentations **132**
C.1 Première expérimentation . . . 133
C.1.1 Lettre d'information . . . 133
C.1.2 Formulaire de consentement . . . 134
C.1.3 Droit à l'image . . . 135
C.1.4 Questionnaire . . . 136
C.1.5 Résultats . . . 138
C.2 Deuxième expérimentation . . . 139
C.2.1 Lettre d'information . . . 139
C.2.2 Formulaire de consentement . . . 140
C.2.3 Droit à l'image . . . 141

C.2.4 Questionnaire 142
C.2.5 Résultats 146

Bibliographie **157**

D Contribution de l'auteur **158**

Liste des figures

1.1 Histogrammes présentant la pyramide des âges en France au 1er janvier 2020 à gauche puis son estimation pour 2070 à droite (source : INSEE [2]). 16

2.1 Schéma global qui répertorie l'ensemble des étapes mises en place pour un transfert d'objet entre un robot (donneur) et un humain (receveur). . . 30

3.1 Photo du robot Pepper de la société SoftBank Robotics, avec les dispositifs matériels utilisés. 43
3.2 Suite de photos prises permettant de visualiser les étapes que doit suivre Pepper pour se recharger de manière autonome sur sa station de recharge. 45
3.3 Explication concernant le problème pour utiliser des librairies externes. . 50
3.4 Explication concernant la solution apportée pour utiliser des librairies externes. 50

4.1 Modélisation des articulations prises en compte pour notre modèle biomécanique, ainsi que leurs amplitudes maximales retenues. Le point rouge sur chacun des modèles indique l'articulation considérées pour les amplitudes dessinées. 56
4.2 Ensemble de points contenus dans un voxel et discrétisés en son centre. . 57
4.3 Représentation graphique du parallélépipède rectangle définissant la zone interdite où les points de l'espace de transfert ne sont pas enregistrés. . . 58
4.4 Deux vues de l'ensemble des points ayant un indice de manipulabilité élevé (supérieur à 90% de la valeur maximale) par rapport aux membres supérieurs. Le repère de référence R_0 est situé entre les deux épaules de l'homme. 60
4.5 Deux vues de l'ensemble des points obtenus en minimisant l'indice de gravité terrestre (moins de 25% de la valeur maximale) par rapport aux membres supérieurs. Le repère de référence R_0 est situé entre les deux épaules de l'homme. 62
4.6 Deux vues de l'ensemble des points obtenus en augmentant l'indice de manipulabilité et en minimisant l'indice de gravité terrestre (respectivement supérieur à 90% et inférieur à 25% de la valeur maximale) par rapport aux membres supérieurs. Le repère de référence R_0 est situé entre les deux épaules de l'homme. 63

4.7 Deux vues de l'ensemble des points obtenus en augmentant l'indice de manipulabilité et en minimisant l'indice de gravité terrestre (respectivement supérieur à 90% et inférieur à 25% de la valeur maximale) par rapport au bras gauche en rouge et au bras droit en bleu. Le repère de référence R_0 est situé entre les deux épaules de l'humain. 63

5.1 Schéma de la tête de Pepper avec la représentation verticale et horizontale du champ de vision de sa caméra de profondeur à gauche des doubles traits, puis à droite, celle de la caméra frontale. 68
5.2 Les premières images, à gauche et au milieu, sont les images originales capturées respectivement par la caméra frontale RGB, puis la caméra de profondeur située dans l'œil de Pepper. La dernière image, à droite, est la représentation de l'image de profondeur en couleur après mise en correspondance. 71
5.3 Présentation des étapes suivies par nos algorithmes afin d'estimer la position des pixels dans l'image de profondeur correspondant aux pixels de l'image RGB sélectionnée (ici par détection du squelette avec OpenPose). 1 - Sélection de point dans l'image RGB; 2 - Estimation des droites épipolaires dans l'image de profondeur; 3 - Échantillonage de ces droites; 4 - Projection des points échantillonnés dans l'image RGB; 5 - Estimation des points les plus proches des points d'origine; 6 - Mise en évidence de ces points projetés les plus proche dans l'image de profondeur. 73
5.4 Trois modes de sélections pour simplifier la calibration et permettre de prendre plus de point dans une zone de l'image. 75
5.5 Deux caméras observent un même objet. La position de l'objet dans l'image de la première caméra induit une infinité de possibilités de position de cet objet dans l'image de la deuxième caméra. Cette infinité de possibilités est définie suivant une ligne traçable sur la deuxième image, ce qui forme la droite épipolaire. 76
5.6 Sortie générée par nos algorithmes à partir d'images obtenues par les caméras du robot Pepper. En haut à gauche : image RGB originale avec les points sélectionnés manuellement parmi les objets contenus dans la scène. En haut à droite : l'estimation de leur correspondance est mise en évidence avec les points colorés dans l'image de profondeur. En bas : deux points de vues différents du nuage de points colorés en 3D sont obtenus. 78
5.7 Résultats obtenus pour trois caméra RGB-D. On utilise ici la Realsense d'Intel, la caméra de Pepper (caméra frontale et l'Asus Xtion dans ses yeux), puis on teste avec deux autres caméras bien dissociées : la webcam d'un ordinateur et la carte de profondeur obtenue avec la Realsense d'Intel. 79
5.8 L'image de gauche donne le résultat obtenu après analyse d'OpenPose sur l'image de profondeur colorée. L'image de droite met en évidence, par des points rouges, la position des pixels retournés par OpenPose dans l'image de profondeur originale (en niveaux de gris). 81

5.9 Estimation des rotations de roulis ϕ (schéma de gauche) et de lacet ψ (schéma de droite) du torse de l'humain à partir de la position des épaules (épaule droite : R_{rsh}; épaule gauche : R_{lsh}) et du repère de référence (R_0). 82
5.10 Schéma expliquant l'estimation de la position du point R_t par rapport à la position des épaules de l'humain et de la rotation ϕ. 82
5.11 Représentation de l'estimation de la position et de la rotation du torse humain. L'image de gauche montre la sélection des points dans l'image de profondeur au niveau du torse. L'image de droite montre le nuage de points récupéré en trois dimensions avec l'estimation de la position et la rotation du repère de référence R_0 en fonction de la position de la caméra de profondeur (représenté par le repère R_{cam}). 83
5.12 Ensemble de points générés pour le bras droit de Pepper en minimisant l'indice de gravité (à 75% de sa valeur maximale) et en maximisant l'indice de manipulabilité (à 25% de sa valeur maximale). Pepper est représenté ici en filaire pour une meilleure visualisation (points et segments noirs). Dans les deux images, nous montrons le repère de la base mobile et le repère de référence du robot (Rp). 87
5.13 Barycentre (étoile rouge) de la zone de transfert générée (violet transparent) pour le robot Pepper. La zone est générée pour le bras droit de Pepper en minimisant l'indice de gravité (à 75% de sa valeur maximale) et en maximisant l'indice de manipulabilité (à 25% de sa valeur maximale). . . 88
5.14 Représentation des trois chaînes cinématiques du robot Pepper que nous utilisons dans nos développements. La chaîne en rouge part de la base mobile jusqu'à la main gauche du robot, la chaîne en bleu jusqu'à la main droite et enfin la chaîne violette jusqu'à la caméra de profondeur. Elles permettant d'exprimer l'ensemble des positions et rotations des repères dans le repère d'origine situé au niveau de la base mobile. 90
5.15 Noms des moteurs sélectionnés du bras droit afin de transférer un objet avec Pepper. Les deux noms de moteurs barrés en rouge ne sont pas sélectionnés au niveau du bras contrôlé (ici le droit). La base mobile est également mise en évidence dans cette image car elle est aussi utilisée pour atteindre le point de transfert. 92
5.16 Organigramme général de notre application pour un transfert confortable vers un humain. 94
5.17 Représentation de l'algorithme qui permet au robot Pepper de transmettre des objets de manière sûre et confortable. Ce schéma global présente l'ensemble des blocs de commande de notre application. 96
5.18 Plan d'une chambre de la cité des aînés de Montpellier avec positionnement des ArUco définissant le chemin à suivre pour aller jusqu'au bord du lit. 98
5.19 Représentation de l'algorithme qui permet au robot Pepper de se repérer à l'aide de marqueurs, puis d'estimer le plus court chemin pour atteindre la zone souhaitée. 98

5.20 Schéma présentant le déplacement désiré du robot Pepper dans le plan d'une chambre de la cité des aînés de Montpellier pour aller jusqu'au bord du lit. 101
5.21 Schéma présentant les marqueurs ainsi que leur zone de ralentissement ($50cm$ de rayon représenté en rouge) et leur zone pour valider l'atteinte du marqueur ($20cm$ de rayon représenté en vert). 102
5.22 Représentation de l'algorithme qui permet au robot Pepper de naviguer en suivant le chemin pré-défini, tout en vérifiant en permanence sa localisation. 103

6.1 Ensemble des points obtenus pour la première expérimentation par différents seuils sur l'indice de gravité terrestre ($s(q)$) et sur l'indice de manipulabilité ($m(q)$). Le pourcentage représente l'emplacement du seuil par rapport à la valeur maximale des indices. 108
6.2 Histogramme montrant les scores obtenus (sur 5) pour chaque zone générée de la première expérimentation. 109
6.3 Séquence d'images permettant de visualiser le transfert d'un objet par le robot Pepper à une personne étant debout. 110
6.4 Ensemble des points obtenus pour la deuxième expérimentation par différentes minimisations de l'indice de gravité terrestre ($s(q)$) et maximisations de l'indice de manipulabilité ($m(q)$). Le pourcentage représente l'emplacement du seuil par rapport à la valeur maximale. Le repère de référence R_0 représente le repère situé entre les deux épaules de l'humain. 113
6.5 Histogramme montrant les notes (sur 5) pour chaque zone générée dans la deuxième expérimentation En orange pour les 7 résidents de la maison de retraite et en bleu pour les 12 participants de cette seconde expérimentation (les 7 résidents + 5 autres personnes). 115
6.6 Histogramme proposant les distances et les hauteurs moyennes ainsi que leurs écarts-types, exprimées en mètre dans le cadre des transferts d'objet lors de la seconde expérimentation. 116
6.7 Séquence d'images permettant de visualiser trois transferts d'un objet par le robot Pepper à une personne âgée assise. Si l'on regarde de gauche à droite, la première colonne montre le robot en phase initiale avec l'objet tenu en main. Puis la deuxième colonne montre le robot en mouvement et la dernière montre que le point de transfert est atteint et que la personne peut récupérer l'objet. 117

Chapitre 1

Introduction

Sommaire

1.1 Contexte et motivation 16
1.1.1 Vieillissement de la population 16
1.1.2 Concept de fragilité 18
1.1.3 Motivation 18
1.1.4 Opportunités offertes par la robotique d'assistance 19
1.1.5 La robotique au service des seniors 20
1.1.6 Projet ROBOd'Home 22
1.1.7 Projet à la cité des aînés 24
1.2 Organisation de la thèse 26
1.2.1 Énoncé du problème 26
1.2.2 Contributions 27

1.1 Contexte et motivation

1.1.1 Vieillissement de la population

D'après une étude statistique réalisée par l'INSEE (Institut National de la Statistique et des Études Économiques) [1], une augmentation considérable du vieillissement de la population est à envisager dans les années à venir. En effet, la population en France pourrait atteindre les 74 millions d'habitants d'ici 2050, soit 8.2 millions de plus qu'en 2013. Les projections montrent également que la France compterait 20 millions de personnes âgées de plus de 65 ans, soit 8.6 millions en plus dans le pays par rapport à 2013. Dans ces 20 millions de personnes âgées, 12.1 millions seraient âgées de 75 ans ou plus. En parallèle, la population âgée entre 20 et 64 ans diminuerait entre 2020 et 2050 et se stabiliserait vers les 51.6% de la population française en 2070. La figure 1.1 illustre la répartition des âges de la population française en 2020 et une estimation de cette répartition pour les années 2070. D'après ces histogrammes, on observe une répartition plus homogène jusqu'à 70 ans pour 2070. On observe surtout une augmentation considérable des populations âgées de plus de 70 ans dans les années à venir d'après les estimations. La majoration de cette population vieillissante nous met face à de nouveaux enjeux d'ordre économiques, sociaux et environnementaux.

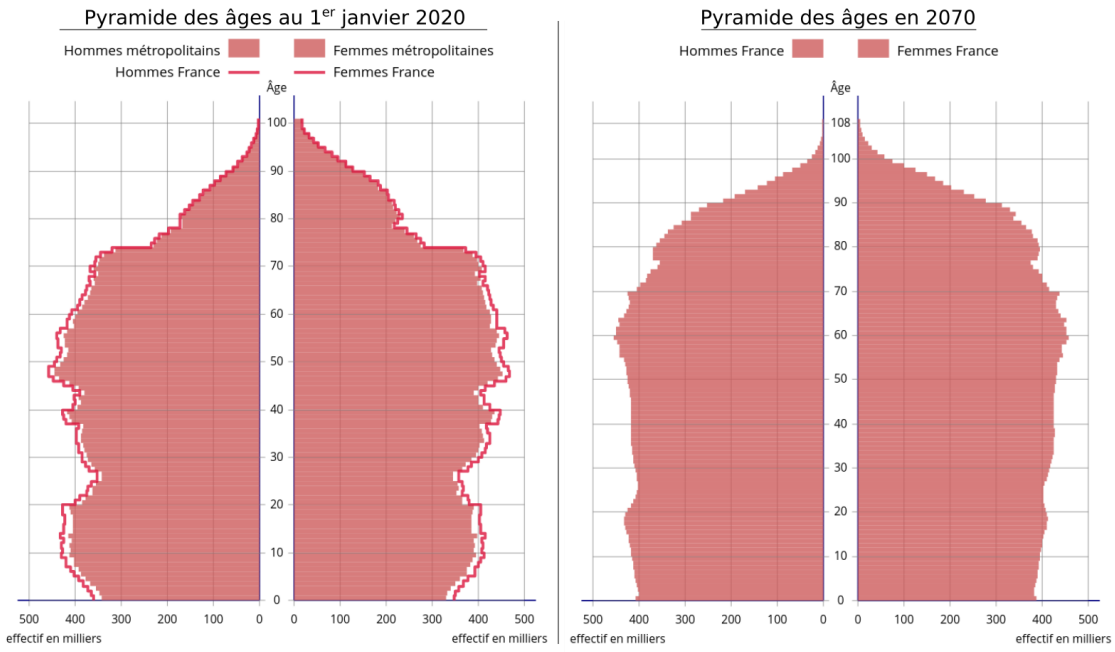


FIGURE 1.1 – Histogrammes présentant la pyramide des âges en France au 1er janvier 2020 à gauche puis son estimation pour 2070 à droite (source : INSEE [2]).

Le vieillissement de la population provient de deux facteurs : l'espérance de vie qui augmente et le taux de natalité actuel qui est moins important que pour les générations d'après la Seconde Guerre mondiale (baby-boom) [3], [4]. Les deux tiers de

l'augmentation de l'espérance de vie sont dus au fait que la mortalité infantile a fortement diminué ces dernières années [5]. En effet, depuis la fin de la Seconde Guerre mondiale, les décès d'enfants de moins d'un an ont fortement diminué [6], notamment grâce aux progrès médicaux contre les maladies infectieuses [7]. La diminution de la mortalité après 70 ans joue également un rôle dans le vieillissement de la population. Le nombre de naissances dans les pays diminue également au fil des années. La population âgée de plus de 65 ans, née entre les années 1945 et 1975, augmentera considérablement le nombre de départs à la retraite, les besoins sanitaires et le nombre de décès dans les années à venir.

Ces quelques chiffres rendent compte de l'instabilité démographique à laquelle notre pays devra faire face dans les années à venir. L'OMS ("Organisation Mondiale de la Santé") [8] définit le vieillissement comme étant une "accumulation d'un vaste éventail de dommages moléculaires et cellulaires au fil du temps" sur un individu entraînant "une dégradation progressive des capacités physiques et mentales, une majoration du risque de maladie et, enfin, le décès". Cambois et Robine [9] indiquent que, dans une population âgée de 55 ans ou plus, 95% des résidents en institution présentent des limitations fonctionnelles (concernant la vue, la locomotion, ...). La locomotion est une activité essentielle de la vie quotidienne. Il est donc important de maintenir cette activité et de mettre en place des aides et des activités pour maintenir la personne en bonne forme physique (à domicile, mais aussi en maison de retraite). Il y aura sans aucun doute un accroissement des besoins en personnel soignant et en institutions pour accompagner les personnes âgées les plus dépendantes. Il est donc nécessaire de travailler dès aujourd'hui sur le développement d'outils adéquats permettant d'aider les personnes dépendantes dans leurs vies quotidiennes.

Nos travaux portent sur l'apport d'objets au bord du lit des personnes âgées vivant en institution et limitent ainsi les risques de chute liés aux déplacements fréquents de la personne. En effet, un rapport de la HAS [1] ("Haute Autorité de Santé") [10] précise que ce risque est plus élevé pour les personnes âgées fragiles (voir Sect. 1.1.2), le plus souvent âgées de 75 ans et plus. Il est noté que les chutes répétées peuvent survenir lors de la marche ou de transferts (changements de position de la personne). Elles ont de lourdes conséquences traumatiques et psychologiques sur l'individu, ce qui augmente son risque de morbi-mortalité [11]. Elles peuvent également entraîner une perte d'autonomie et augmenter leur dépendance [12]–[14]. Enfin, leur prise en charge médico-sociale a un coût financier considérable. C'est pourquoi, il est important de trouver des solutions pour limiter le risque de chute. Les activités proposées par le personnel soignant permettent de maintenir la forme et la force physique. L'objectif est de trouver des solutions permettant de limiter les mouvements fréquents des résidents qui peuvent être évités et d'apporter une aide au personnel soignant. Le choix d'inclure un robot doit être complémentaire à la rééducation. Il ne doit être placé au domicile que lors de phases transitoires (par exemple après une hospitalisation), lorsque la personne est plus à risque d'être fragilisée.

1. https://www.has-sante.fr/

1.1.2 Concept de fragilité

L'article de H. Michel [15] permet de mieux comprendre cette notion de fragilité chez la personne âgée et les conséquences psychiques et physiques qu'elle représente. En effet, le terme de fragilité est un concept gériatrique qui constate que la personne vieillissante peut présenter un état d'instabilité avec risque de perte fonctionnelle, ou de majoration de la perte fonctionnelle existante [16]. Cet état de fragilité peut donc avoir de lourdes conséquences sur la personne, par exemple elle peut entraîner des chutes, des maladies, une perte de l'autonomie et voir même amener au décès selon le degré de fragilité engagé. Cette instabilité peut résulter de nombreuses causes comme un problème de santé (infection virale, chute avec fracture ou même après une hospitalisation programmée quelconque), un changement de lieu de vie ou d'organisation de son quotidien (un déménagement, la reprise professionnelle chez un aidant familial, le décès d'un proche ou d'un voisin...), une dépression, etc. En effet, comme le précise Arveux et al. [17] la fragilité résulte d'une "réduction multisystémique des réserves fonctionnelles, limitant les capacités de l'organisme à répondre à un stress, même mineur".

Pour mesurer cette fragilité, les études réalisées sur les patients se basaient essentiellement sur l'analyse de l'apparition des indicateurs physiques comme la perte de poids, la réduction de la mobilité, de l'activité, de la force, de l'endurance... Depuis quelques années maintenant, ces études s'efforcent de mesurer les dimensions psychologiques sur les patients examinés et surtout la mesure du lien social. Mesurer ces facteurs permet de mieux repérer la fragilité et permet donc d'adopter une démarche préventive cohérente afin de mettre en place une stratégie de prise en charge adaptée. L'idée est d'intervenir en amont sur l'avancement de cette fragilité qui est réversible. C'est cette réversibilité qui est tout l'enjeu de nos travaux de recherche qui visent à mettre en place des solutions robotisées permettant d'estimer et d'agir le plus précocement possible sur le développement de cette fragilité. Le fait de pouvoir intervenir rapidement permet d'éviter les conséquences irréversibles sur le patient pouvant entraîner une ré-hospitalisation, le rendre dépendant ou majorer son état de dépendance et entraîner donc une mise en institution telle qu'un EHPAD [2].

1.1.3 Motivation

Tout au long de sa vie, l'être humain tente au mieux d'acquérir de l'autonomie dans ses actions quotidiennes. Cette capacité est essentielle, car elle lui permet d'évoluer et d'avoir un sentiment de sécurité et de liberté, tout en répondant à ses besoins naturels. Cependant, en raison de l'âge ou du handicap, l'autonomie d'une personne peut être fortement impactée et la vie quotidienne peut devenir un véritable parcours du combattant. Pour les professionnels de santé, il est essentiel de réfléchir constamment à l'amélioration de leurs actions auprès des patients. Cela passe par la mise en place de mesures de prévention durant l'accompagnement, mais aussi par le biais d'outils adaptés à leurs handicaps. Ces derniers ont pour objectif de maintenir les personnes

2. Établissement d'Hébergement pour Personnes Âgées Dépendantes

âgées en perte d'autonomie à leur domicile, et leur permettre de retrouver un confort acceptable dans leur vie quotidienne. En effet, il existe de nombreux outils et aides adaptées à des besoins très spécifiques par exemple, ouvre-boîtes et décapsuleurs électriques, barres d'appui dans les toilettes ou la douche, etc. Avec l'arrivée de la robotique, les professionnels de santé voient un réel intérêt pour répondre au mieux aux besoins du patient, mais aussi afin d'alléger certaines de leurs tâches répétitives. On peut également imaginer l'aide que ce type d'outils peut apporter aux aidants.

La robotique d'assistance est une opportunité d'étudier cet enjeu de santé publique pour aider notre population vieillissante à limiter sa perte d'autonomie et améliorer sa prise en charge. Elle présente de nombreux avantages, comme le simple fait de tenir compagnie, de permettre des appels vidéo, de permettre la téléprésence, d'alerter les professionnels du soin ou l'aidant en cas de besoin ou encore d'apporter des objets au patient. Les applications peuvent être multiples et sont étudiées par les chercheurs et ingénieurs depuis des années.

1.1.4 Opportunités offertes par la robotique d'assistance

Le développement de la robotique d'assistance est actuellement en plein essor. En effet, d'après le communiqué de presse de l'IFR ("International Federation of Robotics") à ce sujet [18], ces dernières années une augmentation de la vente des robots de service a pu être observée (32%). Parmi eux, il y a les robots de service à usage personnel et domestique, et les robots médicaux. Elle renforce l'idée que le monde souhaite investir dans leurs services et leur accorder de plus en plus de confiance pour leur confier des tâches du quotidien. C'est donc une opportunité à étudier face à ce défi de santé publique pour aider notre population vieillissante à limiter sa perte en autonomie et améliorer sa prise en charge que ce soit en institution ou au domicile. L'équipe médicale de la Clinique Beau Soleil de Montpellier se propose de réfléchir à la mise en place d'études pilotes sur l'utilisation de nouveaux outils : les robots d'assistance. L'idée est de définir les besoins auxquels ils pourraient répondre, avec quelles personnes et comment procéder. L'intérêt est de développer des applications et de mettre en place ces robots assistants afin d'évaluer directement sur le plan médico-social leur impact. Il faut donc relever la satisfaction des utilisateurs afin de savoir si pour eux le robot comble bien un besoin et s'il leur a été utile.

La robotique est une science alliant différentes disciplines comme la mécanique, l'électronique, l'informatique et enfin l'automatique. L'alliance de ces différents domaines permet de réaliser des machines capables de comprendre et de réagir à des événements internes et externes à leurs propres systèmes. Les robots sont avant tout des outils industriels, scientifiques ou domestiques capables de réaliser des tâches bien spécifiques dans le seul but de répondre au mieux à un besoin prédéfini. Dans le cas de la robotique d'assistance, le robot doit être capable d'assister la personne humaine dans des tâches de vie quotidienne. Ces machines intelligentes doivent donc être capables de se déplacer dans des environnements conçus pour l'homme et d'interagir avec ce dernier. Dans le cadre de nos recherches, nous nous intéressons aux robots capables d'apporter une assistance aux personnes âgées, que ce soit pour diri-

ger des séances d'exercices physiques, informer sur les actualités, rappeler les prises de médicaments, etc. Le progrès de la robotique promet des robots de plus en plus autonomes et capables d'interactions de plus en plus élaborées. L'utilisation de robots assistants au sein même de nos domiciles nous permettrait de gagner en autonomie et combler certaines lacunes pour éviter une entrée en institution pour les personnes âgées dépendantes. Dans ces établissements de soin, leur utilisation permettrait de soulager le personnel soignant par exemple et pourrait avoir une fonction de compagnon ou remplir des services bien spécifiques et utiles aux résidents.

1.1.5 La robotique au service des seniors

Il existe de plus en plus de services utilisant les nouvelles technologies, développées à destination des personnes âgées. Ces dispositifs ont pour la plupart un même objectif qui est d'améliorer la qualité de vie des seniors en leur permettant de rester à leur domicile, d'avoir accès à des services de télé-assistance pour améliorer leur sécurité, leur communication avec leur famille et éviter au maximum la solitude : c'est le monde de la "Silver économie". En France, plusieurs prototypes de robots ont été développés ces dernières années pour remplir la fonction de compagnon idéal que ce soit au domicile ou en EHPAD.

La société Robosoft Engineering [19] implantée au Pays Basque est une entreprise de robotique qui a développé depuis 2009 des robots de santé et d'assistance, nommés Kompaï [20]. Ce robot est destiné au développement de programmes de recherche dédiés en particulier à l'assistance des personnes âgées. Il collecte des données qui sont essentielles pour le personnel soignant ; il peut rappeler les horaires de prise médicamenteuse, propose des activités et il permet aussi d'appeler facilement la famille ou les proches de la personne âgée isolée. Ces années de travail leur ont permis d'expérimenter ce robot et d'ajouter des fonctionnalités correspondant aux usages des personnes âgées et des professionnels de santé. Dans le cadre du projet Kepa, en partenariat avec d'autres entreprises, Kompaï a reçu le premier prix au concours mondial d'innovation pour la catégorie Silver Economy en 2016.

Le robot Buddy [21] a été développé par la société Blue Frog Robotics, jeune start-up. C'est un robot compagnon de petite taille destiné à toute la famille. Il est considéré comme étant un assistant personnel capable de proposer des recettes, des musiques, des vidéos ou de rappeler des événements importants. Il peut également surveiller la maison en cas d'absence et alerter s'il détecte un élément anormal. La société a aussi souhaité le développer à des fins thérapeutiques dans l'objectif de pouvoir "vieillir chez soi" [22]. En effet, Buddy peut assurer une interaction sociale et une assistance avec les personnes âgées vivants seules et isolées. Ce robot permet encore une fois de rappeler des événements importants, la prise de médicaments, les rendez-vous, les livraisons... Il peut détecter les chutes et les situations inhabituelles. D'après son constructeur, c'est un robot qui est capable de lutter contre la solitude de nos seniors, leur permettant de rester indépendants d'une aide humaine et de repousser au plus tard l'entrée en institution. Buddy a aussi été récompensé par un "Best of innovation Award", Blue Frog Robotics au CES ("Consumer Electronics Show") en janvier 2018 à Las Vegas.

Le robot Cutii [23] de la société française Yumii promet également de faciliter le maintien au domicile des personnes âgées en améliorant la communication entre la personne âgée (à son domicile) et ses soignants, sa famille, et même des associations. Ce robot peut en effet détecter la personne âgée et la rejoindre afin de répondre au téléphone par exemple, consulter des activités, etc... La jeune start-up mise également sur le concept de communauté. En effet, ils ont créé une application permettant à n'importe qui d'intégrer cette communauté et, par le biais de cette application, de partager des vidéos de cuisine, des visites à travers le monde, ou partager tout simplement sa passion. Il est possible également de devenir animateur et de proposer des animations aux utilisateurs par le biais de la plateforme. Pour cela, il est nécessaire de soumettre sa candidature à l'équipe en charge du robot [24] pour proposer ses propres activités aux seniors. La personne âgée aura donc accès à cette grande quantité de données, elle pourra partager, discuter avec cette communauté et participer à des animations. Cette idée innovante a pour objectif de lutter contre la solitude.

À ce jour, les robots d'assistance sont peu ou ne sont pas diffusés au service des personnes âgées. L'un des premiers freins à l'application de ces technologies est leur coût qui reste encore très élevé pour un particulier et ces aides techniques ne sont pas prises en charge par la Sécurité Sociale, et l'Allocation Personnalisée à l'Autonomie (APA). En effet, elles ne prennent en compte que certaines solutions comme la téléassistance et les aides humaines. Le frein principal reste celui de l'acceptation de ces robots, plus difficile en Europe qu'en Asie par exemple.

Pour avoir un aperçu de l'opinion que peuvent avoir les Européens sur la robotique, un sondage fut réalisé par "TNS Opinion and Social" [25] auprès de plus de 27 000 personnes, dans un total de 28 pays européens. Cette étude permet de se rendre compte que les Européens ont en grande partie une opinion positive envers les robots (environ 64%). Une personne sur sept (environ 14%) a répondu qu'elle avait déjà utilisé ou interagi avec un robot. Ce sondage renseigne aussi sur les attentes des usagers concernant l'utilisation de ces machines intelligentes. La grande majorité (85%) pense que les robots sont capables d'effectuer les tâches trop difficiles ou trop dangereuses pour l'humain. Cependant, les personnes interrogées montrent leurs réticences vis à vis des robots qui fournissent des services et de la compagnie aux personnes âgées ou infirmes (51%), et l'expansion de la robotique reste modeste en Europe. Si on se réfère au communiqué de presse de l'IFR [26] à ce sujet, il est précisé qu'entre 2010 et 2016, "le taux de croissance annuel moyen de densité de robots était de 9% en Asie, 7% dans les Amériques et seulement 5% en Europe".

De manière générale, les outils robotisés présentés proposent de lutter contre l'isolement social, d'assurer la sécurité et de stimuler cognitivement les personnes âgées isolées. Ces robots utilisent tous une tablette leur permettant d'interagir bien plus facilement avec leurs utilisateurs. L'objectif de nos projets de recherche est de mettre en place un robot assistant au sein du domicile ou d'un EHPAD pour lutter contre l'isolement social et détecter le plus précocement possible une fragilité chez l'utilisateur. Le robot proposé pour ces développements se nomme Pepper et il présente également une tablette sur son torse, ainsi qu'une base mobile pour se déplacer facilement. Contrairement aux robots présentés, il présente une particularité physique

intéressante à exploiter pour apporter de nouveaux services : des bras. La coordination entre professionnels de la robotique et professionnels du soins ont permis de mettre en place deux projets d'étude. Le premier consiste à placer le robot Pepper au sein du domicile d'une personne âgée isolée et sortant d'une hospitalisation. Le second propose de placer ce même robot à l'étage d'un EHPAD afin de rendre des services prédéfinis aux résidents directement dans leurs chambres.

1.1.6 Projet ROBOd'Home

Le projet ROBOd'Home est une étude pilote proposant de tester une solution robotisée au service des personnes âgées isolées à leurs domiciles. L'idée est de mettre en place un robot assistant directement chez la personne pendant un certain temps après son hospitalisation. En effet, le personnel médical identifie cette phase de retour à domicile comme pouvant être un modèle typique de situation de fragilité. Le robot accompagnera et assistera le patient dans son lieu de vie tout en garantissant sa sécurité et son autonomie pour éviter une ré-hospitalisation. Le projet a été imaginé par le personnel soignant de la Clinique Beau Soleil de Montpellier puis construit en partenariat avec des ergothérapeutes, ainsi que des ingénieurs et chercheurs du domaine de la robotique. Dans le cadre de ce projet, nous avons développé l'ensemble des applications et fonctionnalités défini par le cahier des charges. Ce projet s'inscrit dans le cadre juridique de la recherche biomédicale afin d'en garantir une méthodologie et une évaluation rigoureuses et scientifiques. Les besoins réels du patient sont effectivement souvent supérieurs aux aides proposées. La sécurité sociale souhaite d'ailleurs diminuer les temps d'hospitalisation au maximum. Si la sortie est trop précoce, elle entraîne une augmentation des besoins chez le patient alors que les aides et les coûts en personnel sont limités. Cet outil d'assistance devrait permettre d'améliorer la sécurité du patient et la qualité de sa prise en charge.

L'objectif de cette étude est d'ajouter un outil numérique et interactif auprès du patient afin d'optimiser la communication avec son aidant référent ou les soignants qui le prennent en charge. Ce projet vise donc à améliorer l'accompagnement en ajoutant un service en plus des aides et visites au domicile déjà proposées. Durant son séjour chez le patient, ce robot récupère des paramètres médicaux par le biais de questionnaires. Ces paramètres fournissent à l'équipe soignante un ensemble d'informations leur permettant de réagir précocement en cas de changements de comportement. Le patient, selon son état psychique et physique, pourra ne plus être en totale capacité d'assouvir l'ensemble de ses besoins.

Afin de réaliser convenablement cette étude, l'expérimentation devra être réalisée auprès de douze patients. Sa durée pourra être variable de deux à quatre semaines selon les choix et besoins de la personne. Durant cette période, le robot la sollicitera afin qu'elle exprime sa satisfaction (critère principal de l'étude) en lui attribuant une notation sur une échelle de Likert [27]. Pour être éligible à cette étude, le patient doit répondre aux critères d'inclusion et de non-inclusion : il doit être âgé de plus de 65 ans. Il doit conserver des fonctions cognitives correctes pour son âge. Celui-ci doit bien comprendre la nature, le but et la méthodologie de l'étude et doit être en accord pour co-

opérer lors des évaluations. Il doit avoir un aidant référent qui doit être en accord avec le projet et accepter de recevoir des alertes par mails et/ou par SMS à tous moments de la journée. Le robot mis à disposition a une base mobile, ne lui permettant pas de monter les marches, le patient doit donc habiter dans un logement de plain-pied.

Afin de vérifier que les personnes peuvent participer à cette étude, une première phase d'évaluation est réalisée à la Clinique durant l'hospitalisation du patient : c'est la visite de pré inclusion. À ce moment, médecins et ergothérapeutes réalisent bilans et évaluations et identifient l'aidant référent. À sa sortie, le patient sera alors raccompagné à son domicile avec le robot pour une mise en service. Après cette première phase, le patient bénéficiera de l'accompagnement du robot durant une période de deux à quatre semaines, en plus des aides humaines habituellement proposées. Lors de la mise en place du robot au domicile, nous facilitons au maximum l'appropriation du robot par la personne à travers un paramétrage sur mesure, et une démonstration de son utilisation. Après cette période d'expérimentation, si le personnel soignant juge que la personne n'est plus en situation de fragilité, le robot sera retiré du domicile. Enfin, une dernière évaluation est prévue à trois mois pour faire le point.

Un travail préalable a permis de recenser les besoins par les professionnels du soin et les possibilités techniques offertes par la robotique. Ce robot est capable de rappeler les traitements médicaux prescrits par le médecin et rappeler également la prise de repas ou des rendez-vous enregistrés. Il peut aussi surveiller des paramètres essentiels comme l'humeur, la douleur, l'alimentation, l'hydratation, le nombre de fois où le patient est allé à la selle, etc. Cette étude propose de recueillir les variables nécessaires et indispensables à la mise en place d'une évaluation médico-économique pour un déploiement de plus grande ampleur. Ce projet ROBOd'Home a été retenu, soutenu et totalement financé par la Fondation de l'Avenir [3]. La mise en application de cette étude et l'obtention des premiers résultats sont attendus très prochainement et ne sont pas inclus dans ce document.

Par le biais de ce robot, nous proposons également des jeux cognitifs et des exercices physiques adaptés qui ont été définis en amont par son kinésithérapeute et son ergothérapeute. Ces activités lui permettront un maintien de sa santé physique, cognitive et morale à son domicile. On souhaite en effet rendre le robot plus ludique avec la possibilité de jouer, de lancer facilement des animations ou des courts dialogues avec le robot (blagues, réponses à des questions, explications, ...). On tente également de rendre cet outil le plus pratique possible pour le patient et les personnes qui gravitent autour d'elle (personnel soignant, aidant ou toute autre personne proche). En effet, un humain peut demander au robot des informations concernant la météo, l'éphéméride, l'heure, le niveau de sa batterie, etc. Il est possible également d'ajouter des notes sur un cahier de liaison numérique permettant de faire le lien et de passer des informations rapidement entre les différents intervenants. L'application que nous avons faite sur la tablette du robot, propose d'ajouter également des rendez-vous à un agenda numérique. Cela permet au robot de faire des rappels au patient. On souhaite également que ce robot assistant ne soit pas contraignant pour le patient ou les proches venant au domicile. Pour cela, il est possible par exemple de le mettre en veille à tout moment

3. https://www.fondationdelavenir.org/

en posant sa main sur le dessus de sa tête pendant quelques secondes. Pour éviter au patient de se soucier de l'état de sa batterie, le robot est capable aussi d'analyser son niveau de charge et d'aller de manière autonome sur sa station de recharge. L'application a été conçue et réalisée avec l'accord d'ergothérapeutes. Ces professionnels ont l'habitude de proposer et d'adapter des outils pour les personnes âgées. L'objectif est d'avoir une bonne ergonomie de l'application permettant donc à la personne d'être totalement autonome avec celle-ci.

1.1.7 Projet à la cité des aînés

Le projet à la cité des aînés consiste à mettre en place un robot assistant au sein d'un EHPAD afin de fournir des services directement auprès des résidents dans leurs chambres. Un des objectifs premier est de développer un outil robotisé qui permet de soulager le personnel médical de tâches répétitives et bien souvent chronophages. Le but n'est pas de les remplacer, mais de leur permettre de gagner du temps et de valoriser leurs fonctions de soignant.

Parmi ces taches, nous avons choisi un service particulier : permettre un échange vidéo entre le résident et ses proches en utilisant un outil robotisé comprenant une tablette. En effet, l'épidémie de COVID-19 nous a permis de constater l'importance de la mise en place de moyens pour communiquer avec l'extérieur. Lors du premier confinement, ces personnes se sont vues du jour au lendemain enfermées dans leurs établissements de soin et n'avaient plus le droit de recevoir de visite. Il leur est parfois difficile pour certains d'utiliser des technologies plus récentes pour appeler leurs proches. Malgré la levée des restrictions concernant les visites, cette pandémie a souligné l'intérêt de mettre en place certains gestes barrières entre ces personnes, ayant une santé fragile, et les personnes extérieures pouvant véhiculer des maladies. Limiter les contacts physiques et les visites permet de limiter les chances pour eux d'être contaminés. Cependant, l'isolement a des répercussions psychologiques importantes sur la santé de l'humain. L'isolement physique doit absolument être accompagné d'un moyen permettant de ne pas être totalement exclu socialement et donc avoir une compagnie virtuelle. L'idée est donc de mettre en place un robot assistant, capable de se déplacer de chambre en chambre et d'interagir avec les résidents dans leur lieu de vie pour leur proposer d'effectuer un échange vidéo avec les membres de leurs familles. Il peut également fournir d'autres informations et leur faire des rappels sur leurs activités journalières. Pour remplir ces tâches, il faut permettre au robot de se localiser au sein de l'établissement et de se déplacer convenablement jusqu'aux chambres des patients préalablement enregistrés et concernés par l'échange.

Dans le cadre de ce projet, nous avons imaginé un scénario type pour le développement de nos applications. Dans un premier temps, le robot se trouve dans la salle commune au même étage que les résidents où il doit se présenter. Un appel avec un communicant (personne à contacter et proche du résident) qui a été enregistré et permet au robot de savoir quand il doit aller dans sa chambre. Au moment venu, le robot descend de sa station de recharge et cherche à se localiser. Pour cela, nous utilisons des marqueurs spécifiques que nous plaçons au sol (voir chapitre 5). S'il aperçoit un de ces

marqueurs, le robot l'identifie et peut estimer sa position. Il génère ensuite un chemin entre ce marqueur et le marqueur "cible" qui est celui proche du lit de la personne à aller voir. S'il ne trouve pas de marqueur, le robot regarde autour de lui, de gauche à droite, et augmente ainsi ses chances d'en apercevoir au moins un. Pour le déplacement, le robot navigue entre les différents marqueurs jusqu'à ce qu'ils soient visibles les uns après les autres et suffisamment proche pour passer au suivant. Le robot peut ainsi aller au marqueur se trouvant le plus proche du lit du résident concerné.

Lorsque le robot atteint le dernier marqueur, il cherche la présence d'un humain à l'aide de l'une de ses caméras RGB. Lorsqu'il l'aperçoit, le robot lui annonce qu'il peut lancer un appel vidéo avec l'un de ses proches s'il le souhaite. Le robot affiche sur sa tablette les personnes enregistrées qu'il peut appeler (photos accompagnées de leurs prénoms ou surnoms). Si la personne ne souhaite pas communiquer alors le robot enregistre cette information et retourne à sa station de recharge dans la pièce commune. S'il souhaite communiquer, le robot envoie le lien de connexion au communicant et lance l'appel. Après avoir fini cette communication, le robot lance un questionnaire de satisfaction sur sa tablette que devra remplir le résident. Il envoie également un questionnaire au communicant via internet. Ces questionnaires ont été établis avec les acteurs gravitant autour de ce projet et nous permettant de faire une analyse et une évaluation rigoureuses et scientifiques. Ce projet a été retenu et partiellement financé par le conseil scientifique d'AÉSIO Santé. Dès que l'expérimentation de ROBOd'Home sera terminée, le robot Pepper sera implanté, d'abord à la cité des aînés de Montpellier puis à la cité des aînés de Saint-Étienne.

L'ensemble des développements seront réalisés pour suivre le plus fidèlement possible ce scénario d'étude. Cependant, nous avons rencontré quelques difficultés concernant la mise en place de l'appel vidéo avec cette version du robot (voir 3). Les services existants pour réaliser des appels vidéo ne fonctionnent pas sur la tablette du robot. Afin d'avancer sur ce projet, nous avons mis entre parenthèses cet appel vidéo pour se concentrer sur un autre service. L'objectif est de valider l'ensemble des développements réalisés autour de ce projet en EHPAD. Le service proposé consiste à apporter et transférer un objet au bord du lit de la personne.

1.2 Organisation de la thèse

1.2.1 Énoncé du problème

La population vieillissante est de plus en plus importante en nombre et en âge plus avancé. Ces personnes souhaitent le plus possible rester à leur domicile si cela leur est possible en fonction des besoins qu'ils leur sont nécessaires. On a donc en EHPAD des personnes plus âgées et encore plus dépendantes qu'avant. Nous avons aussi de plus en plus de personnes âgées restant à leur domicile et nécessitant tout de même un accompagnement. Les générations plus jeunes (moins de 65 ans) sont moins nombreuses et doivent permettre de remplacer les départs en retraite tout en aidant cette population plus dépendante à vieillir convenablement. De plus, ces personnes peuvent présenter un état de fragilité avec risque de perte fonctionnelle, ou de majoration de la perte fonctionnelle existante. L'instabilité causée par cette fragilité peut avoir de lourdes conséquences sur leurs états psychiques et physiques, pouvant entraîner maladies, chutes et hospitalisations. C'est un état réversible qui peut être modifié s'il est détecté suffisamment tôt et si l'on propose un accompagnement adapté au patient. Dans le cas contraire, la personne entre dans un cercle vicieux qui ne fera que la fragiliser d'autant plus, l'entraînant vers une mise en institution (EHPAD) ou vers le décès selon le degré de fragilité engagé. C'est pourquoi, il faut réfléchir dès aujourd'hui au développement d'outils et de services permettant d'aider les personnes dépendantes dans leurs activités de vie quotidienne. Il est primordial pour les soignants de détecter cette fragilité le plus précocement possible et de proposer des outils adaptés pour sortir de cet état d'instabilité.

L'autonomie de la personne est primordiale pour lui permettre de rester le plus possible à son domicile et de limiter les aides humaines. Mettre à disposition un robot d'assistance au domicile permet d'apporter un outil interactif et suffisamment autonome pour remplir certains services. Des entreprises proposent des robots permettant d'agir sur cette fragilité. En effet, leurs services permettent de détecter des chutes, d'alerter les proches ou les services spécialisés, de rappeler la prise de médicaments, des rendez-vous, et permettre aussi d'appeler ses proches pour lutter contre l'isolement social. L'idée est de fournir un service permettant de bien "vieillir chez soi" ou de lutter contre la solitude dans les EHPAD pour des personnes fragiles qui ne se servent que très peu de la technologie existante.

Ces outils ne présentent pas de bras et ne proposent pas d'aide physique, mis à part Kompaï pour l'aide à la marche tel un déambulateur. Les robots low-cost actuels ne permettent pas de fournir des services d'aide physique comme aider à se relever ou apporter des objets car souvent trop lourds pour les moteurs et leurs structures. Nous proposons par le biais de nos travaux de recherche d'établir un service d'aide robotisé capable d'apporter des objets du quotidien (légers) à la personne âgée dans son lieu de vie. Proposer d'apporter des objets à la personne, c'est : limiter ses déplacements afin de limiter son risque de chute ; optimiser son environnement de vie et son confort ; et enfin éviter de majorer ses troubles existants pour préserver son autonomie.

1.2.2 Contributions

Au cours des dernières années, nous avons constaté un intérêt croissant pour la transmission d'objet dans la recherche en robotique. Dans les études précédentes, les chercheurs se sont concentrés plutôt sur la performance, l'optimisation et l'amélioration du temps de transfert ainsi que sur la sécurité et le confort physique que l'individu peut ressentir lors de l'échange (voir chapitre 2). Par le bais de nos recherches, nous incluons un nouveau paramètre permettant de vérifier pour une position de bras donnée l'impact de la force de gravité sur les articulations du bras (voir chapitre 4). Ainsi, en minimisant cette métrique, nous garantissons moins d'effort de la part de l'humain pour contrer l'effet de la pesanteur sur les articulations de ses bras. Il y a donc une meilleure stabilité en ces points et il est possible de changer de position plus facilement. L'originalité vient aussi de la combinaison d'indices utilisée pour générer cet espace de transfert. En effet, on souhaite définir une zone de transfert avec une bonne "manipulabilité" où la personne peut bouger ses membres supérieurs facilement afin de garantir une saisie correcte de l'objet. Pour finir, l'énergie potentielle du bras est calculée afin d'assurer une zone de transfert basse. L'ensemble de nos développements nous permettent d'adapter le transfert aux caractéristiques physiques de l'individu, c'est-à-dire que nous sommes capables de définir une zone de transfert personnelle qui tient compte de son poids, de sa taille, de ses amplitudes articulaires, ainsi que de ses préférences. L'ensemble des développements ont été réalisés sur le robot Pepper de la société SoftBank Robotics. La création d'une interface de programmation d'application intermédiaire permettant la communication entre nos développements et les modules du robot fut nécessaire (voir chapitre 3). En effet, elle nous permet de développer plus librement nos applications sur ce robot.

Pour définir et atteindre la position du lieu de transfert, nous proposons une stratégie de commande dans son ensemble (voir chapitre 5). Dans un premier temps, il faut vérifier la position de l'humain à l'aide des caméras du robot. Puis reconsidérer les espaces de transfert selon la situation des deux parties. Ensuite, on définit le point qui semble le plus approprié aux conditions actuelles. Pour finir, la planification et le contrôle du robot sont réalisés afin d'atteindre le point désiré. Nous intégrons également la localisation et la navigation du robot, pour qu'il puisse apporter l'objet à la personne là où elle se trouve. L'objectif est de limiter les déplacements de la personne âgée et ainsi de réduire les risques de chute.

Enfin, nous avons eu l'opportunité de réaliser une expérimentation originale impliquant des personnes âgées résidant au sein d'un EHPAD (voir chapitre 6). Certains de ces résidents présentaient des troubles articulaires au niveau des épaules. Cette mise en pratique, en condition réelle, nous permet d'avoir de précieux retours. En effet, leurs avis sur ces expérimentations sont essentiels afin de mieux comprendre leurs besoins et la manière dont ce type de robot doit être inclus.

Chapitre 2

État de l'art

Sommaire

2.1 Introduction 29
2.2 La transmission d'un objet 30
2.3 Phase 1 : Le pré-transfert 31
2.3.1 Stratégie de communication 31
2.3.2 Analyse et saisie de l'objet 32
2.3.3 Estimation du point de transfert 33
2.3.4 Planification et contrôle du mouvement avant contact 35
2.4 Phase 2 : Transfert physique de l'objet 37
2.4.1 Transfert de force de préhension 37
2.5 Minimiser l'effort 38
2.6 Conclusion 39

2.1 Introduction

L'objectif de ce chapitre est de proposer un aperçu des études existantes sur le sujet de la transmission d'objets entre un humain et un robot. Ces études permettent d'identifier les besoins et les optimisations à réaliser pour être le plus efficace possible lors de ces échanges. En effet, échanger un objet entre deux personnes semble si anodin et commun qu'on en oublierait la complexité de la tâche. Tout d'abord, cette faculté de préhension nous est presque exclusive en tant qu'humain, car elle est seulement réalisable par les primates. La mobilité de nos cinq doigts permettent une bonne manipulation des objets et la préhension n'est possible que grâce au pouce opposable. Il est intéressant d'observer également que l'humain a une forte capacité d'adaptation pour interagir avec son environnement et les personnes qui l'entourent. C'est cette adaptabilité que les chercheurs souhaitent intégrer aux robots pour qu'ils puissent entrer dans nos lieux de vie et nous permettre de nous assister au quotidien tout en interagissant le plus naturellement possible. On souhaite donc développer une stratégie de commande au robot Pepper permettant d'échanger un objet du mieux possible à un humain. Pour cela, il nous faut préalablement étudier sa mise en place, décortiquer chaque étape de sa réalisation et comprendre pour chacune leurs intérêts. Le but est de pouvoir mettre en place une stratégie de commande cohérente pour garantir sécurité, confort et satisfaire l'humain de son expérience utilisateur.

Les différentes parties présentées dans ce chapitre donnent, pour commencer, une définition de cette action de coordination entre deux individus. Puis, pour mieux comprendre les capacités cognitives et physiques nécessaires pour réaliser cet échange, nous décrivons l'ensemble des processus mis en place durant les phases de pré-transfert et de transfert physique de l'objet. Nous détaillons les études existantes dans ce domaine qui nous ont aidés à comprendre l'importance d'optimiser certains de ces processus dans les transferts entre robots et humains. Enfin, nous présentons les indices utilisés dans de précédents travaux afin de minimiser l'effort de l'humain lors du transfert d'un objet. Notre réflexion vis à vis de ces études est primordiale afin d'expliquer au mieux nos motivations de recherches.

2.2 La transmission d'un objet

Le transfert d'objets consiste à passer un objet tenu par un donneur, à un receveur. Cet échange comprend un ensemble de processus cognitifs et physiques entre les deux individus. Avant même de réaliser le transfert physique de l'objet, le donneur doit l'analyser, le saisir correctement en fonction de sa fonctionnalité par exemple, percevoir l'intention du receveur à recevoir, estimer une zone de transfert et planifier des mouvements adéquats. Puis, durant l'échange physique, les deux individus doivent adapter entre eux leurs forces de préhension afin de transférer entièrement la charge de l'objet dans la main du receveur. La complexité de ce type d'interaction est qu'elle n'est pas à sens unique, c'est une action collaborative qui demande un effort de coordination entre les deux individus. En effet, il faut percevoir, prédire les objectifs et ajuster en permanence les mouvements et forces impliqués pour garantir fluidité, efficacité, et sécurité. Malgré les avancées majeures dans ce type d'interaction, le transfert d'objet est encore une problématique étudiée par les chercheurs du monde entier. C'est ce que montre l'étude récente de Ortenzi *et al.* [28] permettant de répertorier la littérature disponible sur les transferts d'objets entre humains et robots.

Mason *et al.* [29] et MacKenzie *et al.* [30] définissent un transfert d'objet comme étant une succession de phases et plus particulièrement deux phases. L'une concerne le pré-transfert (voir section 2.3) et l'autre le transfert physique de l'objet (voir section 2.4). Durant ces deux phases, les deux individus ont des objectifs bien distincts [29]. Dans la suite de ce chapitre, nous proposons un ensemble de travaux et d'études qui ont permis d'identifier, de comprendre et d'optimiser les processus mis en jeu dans ces deux phases. La figure 2.1 propose un schéma global contenant l'ensemble des étapes et phases pour la réalisation d'un transfert d'objet.

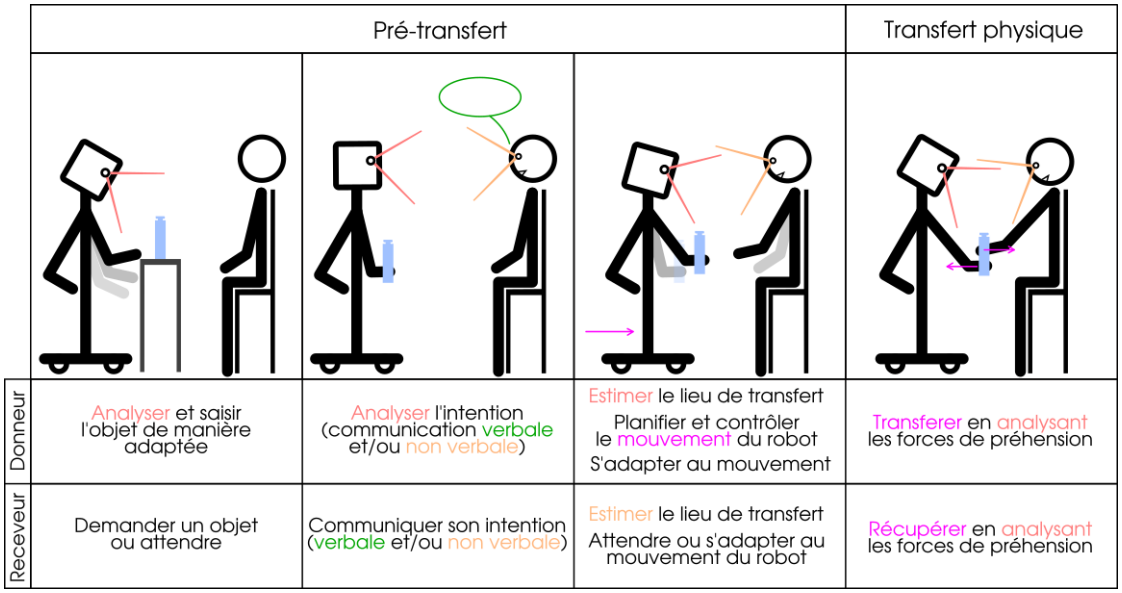


FIGURE 2.1 – Schéma global qui répertorie l'ensemble des étapes mises en place pour un transfert d'objet entre un robot (donneur) et un humain (receveur).

2.3 Phase 1 : Le pré-transfert

Cette première phase, comme son nom l'indique, concerne toutes les interactions et planifications mise en place avant même de transmettre l'objet, ainsi que les mouvements effectués jusqu'au lieu de transfert. C'est donc tout ce qui concerne les étapes avant l'échange physique de l'objet entre les deux individus. Le donneur doit s'assurer de la sécurité concernant l'objet et doit saisir et présenter l'objet au receveur de la manière la plus appropriée possible. Le receveur doit quant à lui communiquer son intention et son implication durant le transfert.

2.3.1 Stratégie de communication

Que ce soit le donneur ou le receveur, la communication tout au long du processus est importante. Elle coordonne l'échange et améliore les prédictions des deux partenaires [31], [32] en fournissant différentes informations. Ces informations permettent aux deux individus de prédire l'intention d'échanger, et peuvent renseigner sur le possible lieu de transfert et le moment [33].

Il est possible pour cela d'utiliser la communication verbale (par la parole) et/ou la communication non-verbale (par le regard, les gestes, ...). La parole peut être un bon moyen de communiquer entre les deux individus pour discuter des conditions de l'échange. Cependant, elle peut aussi altérer la coordination en découpant l'action du transfert par différents temps d'attente et d'écoute. Le plus souvent, l'information fournie par une communication non-verbale suffit à comprendre l'intention du receveur. Par exemple, Castiello *et al.* [34] indiquent que l'homme avait la faculté de comprendre les intentions d'autrui par le biais du mouvement des yeux.

Des études ont mis en place cette interaction visuelle en permettant au robot de bouger les yeux et en dirigeant la tête. Ils ont montré qu'il était possible d'améliorer la vitesse du transfert, car il diminue le temps de réponse de l'humain [35]. Boucher *et al.* [35] ont également rapporté, de par leurs expérimentations, que l'humain est capable d'apprendre et de s'adapter à un robot qui fait des mouvements de la tête. Cette interaction visuelle permet d'avoir un échange plus naturel pour les receveurs humains [36]–[38]. C'est aussi ce que montrent les travaux de Moon *et al.* [37] qui mettent en évidence l'importance du regard du robot lors de transferts d'objets avec des humains. En effet, la direction du regard donne de nombreuses informations sur l'intention, la localisation et le temps pour passer l'objet. Leurs résultats indiquent l'amélioration du temps de transfert d'un objet entre deux individus lorsque ce signal est correctement employé. De plus, ils précisent que l'humain a besoin de contact visuel pour se sentir pleinement impliqué dans la communication.

Les travaux de Cakmak *et al.* [39] proposent d'analyser les effets des contrastes spatiaux et temporels du donneur afin d'améliorer la fluidité des remises. Ces derniers permettent la communication non-verbale par les gestes. En effet, ils ont montré qu'en les marquant davantage, l'intention est plus prévisible et cela permet de réduire le délai du transfert. Le contraste spatial fait référence à la distinction de la pose avec laquelle l'objet est présenté à la personne. Le contraste temporel quant à lui représente la rapi-

dité d'exécution à laquelle l'objet est présenté à la personne (vitesse du mouvement). L'objet doit donc être tendu dans la direction de la personne et suffisamment rapidement pour comprendre que le robot souhaite lui donner. Cependant, cette rapidité ne doit pas être trop conséquente sous peine de surprendre la personne ce qui ne ferait que ralentir la transmission et rendrait le transfert inconfortable.

2.3.2 Analyse et saisie de l'objet

Dans cette partie du transfert, le donneur doit saisir l'objet tout en analysant la manière dont le receveur devra le récupérer. L'idée est de garantir la sécurité du receveur selon la dangerosité de l'objet transmis. Il doit s'assurer que l'objet puisse être attrapé facilement, confortablement [40], de manière à ce qu'il ne tombe pas, puis permettre au receveur, après transfert, de réaliser plus rapidement la tâche à exécuter sans manipulation supplémentaire [41], [42]. D'autres études montrent l'importance concernant cette préhension ([43] et [44]) et l'orientation ([43] et [45]). En effet, il est important de saisir l'objet suivant sa fonctionnalité (en considérant son affordance [1]) et en analysant convenablement ses propriétés physiques pour améliorer le transfert de l'objet. Des études ont été menées pour analyser l'affordance des objets [44], [46], [47] afin de trouver la meilleure façon de saisir un objet [47], tout en anticipant la façon dont l'humain le récupérerait ensuite – en laissant libres les parties saisissables [42]. Puis, à l'inverse, [48] et [49] ont également montré qu'il était possible d'estimer la capacité du robot à saisir l'objet dans la main de l'humain en attrapant les parties restées libres. Les travaux de Chan *et al.* [45] proposent de collecter un ensemble d'orientations de transfert utilisées par les humains pour un ensemble d'objets du quotidien. Ces informations permettent alors de caractériser les objets suivant la manière dont ils sont transférés. Ils introduisent ainsi la notion d'axes d'affordance afin de réaliser des transferts plus efficaces et socialement acceptables.

L'adaptation de cette préhension est donc primordiale pour garantir que l'objet puisse être récupéré dans les meilleures conditions [50], [51]. La prise se veut différente suivant la forme, la taille et la tâche à réaliser avec l'objet [52], [53]. Les travaux de Cini *et al.* [42] montrent les changements de type de préhension, de force ou de précision, et le placement de la main selon l'objet et l'action que l'on souhaite réaliser. Dans le cadre d'un transfert, ils ont montré que le donneur préfère les préhensions de précision pour laisser suffisamment d'espace afin de permettre au receveur de saisir confortablement l'objet et d'effectuer plus facilement des préhensions de force. Ils répertorient donc cinq facteurs principaux qui influencent le choix de la saisie : les contraintes imposées par l'objet, la tâche, le receveur, selon les habitudes du receveur liées à l'objet et enfin toutes autres contraintes liées au hasard (position initiale, environnement, ...).

Des méthodes d'apprentissage automatique pilotées par les données sont également étudiées pour améliorer l'interaction homme-robot lors du transfert d'objets. Par exemple, Quispe *et al.* ont permis d'améliorer le confort du transfert en apprenant les préférences de l'utilisateur en fonction de son activité et de sa position [54]. Ces méthodes montrent des résultats plutôt convaincants et intéressants. Cependant, elles

1. L'affordance est la faculté d'un objet à communiquer la manière dont il doit être utilisé.

ne sont mises en œuvre que pour des tâches très spécifiques et leurs algorithmes ont besoin de nombreuses données pour fonctionner correctement. De plus, ce sont ces données qui façonnent l'algorithme et son interprétation. En effet, deux algorithmes totalement identiques peuvent agir de manière radicalement différente en raison des données sur lesquelles ils ont été entraînés. Cela peut être une source de biais dans la prédiction des algorithmes. Par conséquent, tant la quantité que la qualité des données peuvent engendrer des problèmes d'ajustement plus ou moins importants. Avec ces méthodes, il est plus difficile de s'adapter directement à une personne en particulier (par exemple, avec une taille et un poids différents) car cela nécessiterait un entraînement spécifique et personnel. Dans notre travail, la prise en compte du modèle biomécanique de l'humain permet une adaptation plus rapide du robot aux propriétés articulaires de la personne, ainsi qu'à ses préférences.

2.3.3 Estimation du point de transfert

Lorsque deux humains réalisent entre eux un transfert d'objet, une estimation de la zone où doit être transmis l'objet est effectuée implicitement. Hansen *et al.* [55] ont démontré qu'entre humains, les transferts sont effectués à peu près à mi-chemin entre le donneur et le receveur. Cette zone de transfert doit être un compromis et doit être facilement accessible par les deux individus. C'est ce qu'ont montré les travaux de Vahrenkamp *et al.* [56] concernant l'intersection des espaces atteignables entre les deux individus. Certaines études s'intéressent directement à la génération de l'espace atteignable. Par exemple, Zacharias *et al.* [57] ont utilisé un robot mobile pour ouvrir des tiroirs. Pour réaliser cette tâche, ils ont dû générer un espace atteignable par le bras du robot et mettre en évidence l'accessibilité de cet espace. Ceci est possible en calculant l'indice d'accessibilité : pour un point donné (atteignable par le robot), plus l'indice est élevé, plus il est facile pour le robot d'atteindre ce point depuis différentes positions. Dans notre cas, nous ne calculons pas cet indice, mais nous notons l'importance d'avoir plusieurs possibilités pour atteindre un point donné. Il peut facilement être comparé à l'indice de manipulabilité, qui met en évidence les zones où le robot ou l'humain aura plus de facilité à changer de position.

Sisbot *et al.* [58] [59], [60], ont travaillé sur un planificateur de mouvements spécifiquement conçu pour le transfert d'objets entre humains et robots. Pour ce faire, ils prennent en compte la visibilité, la sécurité et le confort physique des mouvements du bras humain. L'objet doit donc être donné dans le champ de vision de la personne, suffisamment loin pour des raisons de sécurité mais à un endroit où l'humain ne doit pas faire trop d'efforts pour atteindre l'objet. Pour définir ces critères, les auteurs se réfèrent à des études d'utilisateurs ([61], [62] et [63]), à la théorie de la proximité [64] ou à l'ergonomie humaine ([65] [66]). Koay *et al.* [62] et Dautenhahn *et al.* [63] ont travaillé sur l'approche que doit avoir le robot en termes de direction (préférence d'approche) et de distance. Ces études informent l'importance de prendre en considération les préférences humaines, concernant les distances, la direction et les mouvements. Afin d'évaluer leur planificateur, ils ont mis en place une étude expérimentale pour valider leurs développements et évaluer l'effet de cette interaction. Les études menées

par Koay *et al.* [61], [62] proposent d'examiner les préférences des volontaires. Ces préférences concernent la direction et la distance d'approche du robot, la proximité de ses mains et les mouvements de ses bras durant le transfert d'un objet. La majorité des participants (58.3%) montrent une préférence quand le robot s'approche de face et dans leur zone personnelle. Pour définir le point de transfert, il est donc intéressant de prendre en compte la position du receveur, sa morphologie, ses amplitudes articulaires et la distance qui les sépare. Nemlekar *et al.* [67] ont montré que la distance entre les individus a un impact sur la position et sur la hauteur où doit être réalisé le transfert. La prise en compte du modèle biomécanique dans notre cas a pour objectif d'accentuer la manipulabilité et réduire les efforts du receveur. Comme eux, nous cherchons à générer un espace de transfert ergonomique en maximisant la sécurité et le confort pour le bras humain. Contrairement à eux, nous n'utilisons pas les critères de distance, de direction (concernant l'approche du robot et de l'objet) et d'ergonomie humaine pour obtenir des zones de transfert adéquates. Nous définissons notre espace de transfert en fonction de la manipulabilité (possibilité de mouvement en un point donné) et l'énergie nécessaire à l'humain pour maintenir la pose ou de générer un mouvement (effet de la gravité sur les articulations du bras).

Ces études montrent l'intérêt d'optimiser les mouvements et les positions du robot afin de maximiser l'ergonomie du receveur [68] en permettant donc de réduire les troubles articulaires et l'inconfort. De plus, pour prendre en compte les difficultés des personnes à se déplacer, Mainprice *et al.* [69] et Waldhart *et al.* [70] ont étudié la planification du mouvement et du déplacement du robot lors de ces transferts tout en tenant compte de la mobilité humaine. Ils introduisent ainsi l'indice de mobilité, permettant d'ajuster le comportement du robot vis à vis des contraintes de mouvement de l'humain dans un environnement pouvant être encombré. Ces contraintes dépendent de la difficulté à se déplacer, et à tenir la position debout.

Dans notre travail, nous nous inspirons principalement des travaux de Pyo *et al.* [71], qui ont conçu un système – reposant sur un chariot et un robot – pour la remise d'objets aux patients dans les hôpitaux. La particularité de leur article est qu'il calcule, pour cette application, l'indice de maniabilité afin de donner l'objet dans le meilleur état possible (en mettant en commun la carte de manipulabilité du robot et du patient). Cette mise en commun leur permet de mettre en évidence une hauteur optimale pour le transfert. Leur travail permet d'assurer une manipulabilité maximale pour l'humain et le robot lors du transfert. Nous souhaitons utiliser cet indice de manipulabilité et intégrer l'indice de gravité terrestre pour définir des zones plus acceptables pour l'humain surtout. Cet indice de gravité terrestre permet de minimiser les efforts nécessaires sur le bras de l'humain afin de contrer l'effet de la gravité sur chacune de ses articulations. Cette minimisation permet d'engendrer un mouvement plus facilement et ainsi de garantir le confort de la récupération de l'objet par l'humain. Nous proposons de générer un nuage de points avec ces indices qui définit un espace de transfert plus confortable et plus sûr.

2.3.4 Planification et contrôle du mouvement avant contact

Durant le transfert d'un objet entre deux humains, leurs mouvements se coordonnent en temps réel pour avoir une adaptation continue des deux individus. En effet, le receveur et le donneur engagent tous deux leurs bras en même temps [72], [73]. Ils assurent ainsi la fluidité du mouvement afin d'éviter d'avoir des étapes séparées et successives.

Contrôle du mouvement

En ce qui concerne le contrôle des robots, Cakmak *et al.* [74] ont permis de mettre en évidence que les humains préfèrent les mouvements soient naturels et similaires à ceux de l'homme. Ils permettent ainsi d'avoir une bonne lisibilité. Pour que les bras du robot aient des mouvements dits lisibles pour le receveur humain, il est possible d'utiliser des algorithmes de contrôle en cinématique inverse qui ont la particularité d'être naturels [75], [76]. Ces algorithmes permettent de minimiser l'erreur de position du membre terminal avec une pose désirée en calculant et ajustant les valeurs articulaires des moteurs du robot. Les recherches menées par Huber *et al.* [77] se sont intéressées à l'amélioration des temps de transfert en analysant les temps de réaction chez l'humain. D'après leurs résultats, le temps de réaction est plus court lorsqu'ils utilisent des profils à secousses minimales pour contrôler le robot par rapport aux profils de vitesse trapézoïdale. Notre stratégie de commande se base sur ce type de contrôleur en cinématique inverse, basé sur l'architecture de programmation quadratique (QP). Nous utilisons la librairie logicielle proposée par Tarbouriech *et al.* [78] qui permet d'imposer des contraintes de vitesse sur les déplacements du membre terminal. Elle prend en compte aussi les limites articulaires du robot et les contraintes spatiales pour éviter les collisions (avec lui-même par exemple).

Approches réactives

Pour avoir des mouvements plus flexibles et robustes, il faut que le robot soit capable d'avoir des retours sensoriels afin de modifier en conséquence ses mouvements pour s'adapter au receveur [79]. Il faut donc planifier dans un premier temps un mouvement, puis le changer si besoin en fonction des réactions et des déplacements de l'humain. Les primitives de mouvement probabilistes [80] permettent d'estimer l'intention humaine et de contrôler le robot selon les mouvements humains observés. Comme a pu le montrer Huang *et al.* [81] pour avoir un impact positif sur l'expérience utilisateur, le robot doit être contrôlé de manière à avoir une coordination réactive, même si cela implique que les performances soient moins bonnes. Leur étude montre en effet que pour améliorer les performances (la rapidité de transfert), il faut une coordination pro-active du robot, mais cela sera négatif pour l'expérience utilisateur. Fujita *et al.* [82] montrent qu'un mouvement à faible vitesse diminue le possible stress du receveur humain et améliore ainsi son expérience. Cependant, même si les mouvements ont une vitesse réduite, il est important d'avoir une bonne réactivité du robot (capable de réagir rapidement) pour que l'interaction soit jugée plus satisfaisante [83], [84].

Des études récentes s'intéressent à récolter des ensembles de données et d'images

permettant l'entraînement de réseaux neuronaux dans le but d'estimer la pose de l'objet ou de la main [85]–[87]. Ainsi, ils permettent d'envisager des approches plus réactives qui tentent de suivre l'objet ou la main pour planifier la saisie. C'est évidemment une difficulté, car ils peuvent être plus ou moins obstrués, ce qui rend leur suivi plus difficile avec les capteurs de vision. Les travaux de Rosenberger *et al.* [49] ont permis la reconnaissance et la segmentation d'objets et de la main durant un transfert pour déterminer comment le saisir correctement. Le but est de ne pas entrer en collision avec la main du donneur et donc d'améliorer la sécurité. Costanzo *et al.* [88] proposent également une approche réactive pour récupérer l'objet dans la main d'un donneur (humain ou robot) ou sur une table. Pour cela, ils ont mis en place un asservissement visuel leur permettant, par le biais d'une caméra RGB-D, de se positionner correctement devant l'objet à récupérer. Cela est possible à l'aide d'une image cible permettant de détecter l'erreur de position et de rotation par rapport à l'objet et ainsi se placer correctement devant même si ce dernier est en mouvement. En effet, cela permet d'assurer que les mouvements du receveur sont en accord avec les mouvements du donneur, ce qui permet un transfert fluide. Cette fluidité est une caractéristique essentielle pour considérer que le transfert d'objet est efficace [73]. Le robot peut ainsi suivre l'objet sans utiliser aucun marqueur spécifique tout comme les travaux de Nemlekar et al., [67], Yang et al., [48] et Rosenberger et al., [49]). Ces études diffèrent de celles menées par Medina *et al.* [73], Pan *et al.* [89] et Yamane *et al.* [72] qui présentent l'utilisation de marqueurs ou suivent le poignet de l'humain.

Dans leur étude, Zhao *et al.* [90], proposent un cadre utilisant un modèle LSTM-RNN ("Long Short-Term Memory Recurrent Neural Network") pour générer un mouvement adapté du robot par rapport au mouvement de l'humain lors d'un transfert d'objet. Il permet au robot de s'adapter en temps réel aux mouvements de l'humain. Cependant, ils ne peuvent pas garantir la sécurité et le confort du transfert, puisqu'ils n'utilisent pas de modèle mathématique permettant de le représenter (comme dit précédemment, les performances sont ainsi moins bonnes [81]). Dans notre travail, nous exploitons le modèle biomécanique de la personne, pour intégrer ces notions de sécurité et de confort au sein même de la loi de commande. L'objectif est de transférer l'objet dans les meilleures conditions selon la personne, et qu'elle n'ait plus qu'à le saisir. Nous nous concentrons alors sur la pose du squelette de l'humain pour estimer le point de transfert. Comprendre la position et l'orientation du torse de l'humain, nous permet d'estimer l'espace de transfert disponible par les bras de l'humain en assurant confort et sécurité.

2.4 Phase 2 : Transfert physique de l'objet

La seconde phase commence au moment du contact physique entre l'objet et le receveur. Ici, le donneur doit analyser les échanges de forces de préhension et peut également percevoir visuellement les mouvements du receveur pour confirmer la bonne saisie de l'objet. Ce dernier quant à lui doit attraper convenablement l'objet et le retirer progressivement de la main du donneur pour communiquer son intention de le prendre tout en garantissant sa bonne tenue. Durant cette phase, les deux individus sont impliqués d'un point de vue cognitif et physique.

2.4.1 Transfert de force de préhension

Durant l'échange d'un objet, il y a un partage de la charge qui informe sur l'avancement de ce transfert. En effet, Mason et MacKenzie [29] ont mis en évidence un transfert de force de préhension entre les deux individus lors d'un échange. Quand l'un diminue sa force de préhension, l'autre l'augmente jusqu'à ce que la charge soit totalement transférée. Le fait de ressentir cette force de préhension provenant de l'autre individu nous renseigne sur l'intention qu'il a de lâcher ou prendre l'objet et également s'il le tient convenablement. On parle alors de retour haptique. Les travaux de Costanzo *et al.* [88] soulignent l'importance de cette perception haptique lors d'échanges d'objets entre humain et robot. En effet, leur étude se base sur la perception de la force dans la pince du robot qui permet d'avoir une visualisation en 6 dimensions des forces exercées sur l'objet. Cette visualisation permet au contrôleur d'être réactif durant cette phase du transfert. Pour cela, ce contrôleur analyse les tractions qui s'appliquent sur l'objet de manière à identifier l'intention du receveur à le réceptionner. Afin de s'assurer de la bonne répartition de sa charge et pour éviter qu'il chute, il vérifie les torsions. Cette mise en évidence des forces de préhension permet d'assurer le bon déroulement des transferts. Dans ce même objectif, Chan *et al.* [91] ont présenté une stratégie similaire de relâchement de l'objet qui s'adapte et a montré son efficacité. La manière dont sont réparties ces forces renseignent sur les responsabilités de chacun durant la transmission de l'objet. En effet, le donneur semble garant de la sécurité de l'objet en tenant compte des forces qu'applique le receveur pour le retirer. Le receveur quant à lui va s'assurer de l'efficacité de l'échange [29], [92] en informant et en récupérant l'objet convenablement.

Dans le cadre de notre étude, le robot Pepper a des capacités haptiques très limitées. En effet, il ne présente aucun capteur de force au niveau de ses mains. Il n'est donc pas en mesure de détecter la saisie de l'objet par l'humain. Nous pourrions donc envisager de mener des travaux complémentaires en ajoutant des capteurs au robot utilisé. Cela permettrait d'améliorer considérablement l'interaction physique durant le transfert par la prise en compte des forces de préhensions dans le cadre de nos expérimentations. Pour le moment, une interaction tactile (sur la tablette) ou une communication vocale permet de signaler la bonne réception par l'humain afin que le robot lâche définitivement l'objet.

2.5 Minimiser l'effort

Pour minimiser les efforts sur les bras humains, les travaux de Vahrenkamp et al. [56] et de Suay et al. [60] se basent sur l'étude de Katayama et al. [65]. Ce dernier propose en effet cinq indices permettant de quantifier l'effort que peut ressentir l'humain sur ses articulations. De leurs expérimentations, il en ressort que minimiser le stress musculaire permet de maximiser le confort ressenti de l'humain. Cet indice E_{mmsi} ("Minimum muscle stress index") se calcul de la manière suivante :

$$\mathrm{E}_{mmsi} = \sum(\mathrm{T} * \mathrm{PCSA})^2 \tag{2.1}$$

Avec T la tension musculaire et PCSA ("Physiological cross Sectional Areas") qui est une constante propre à chacun des muscles et qui est mesurée selon le volume du muscle divisé par la longueur des fibres musculaires. Suay et al. [60] soulignent le fait que ces contraintes et le travail du bras sont liés au mouvement, à la charge soulevée et qu'il faut également avoir une connaissance détaillée de la physiologie de l'humain. C'est pourquoi, dans leur cas ils utilisent l'indice E_{mjai} ("Medium joint angle index") :

$$\mathrm{E}_{mjai} = \sum(\mathrm{Q}_{moyen} * \mathrm{Q}_{courant})^2 \tag{2.2}$$

Il permet ainsi de minimiser les efforts en étant le plus proche possible des valeurs articulaires au repos et prennent les points les plus bas en gardant les plus faibles valeurs d'énergie potentielle. Vahrenkamp et al. quant à eux utilisent l'indice E_{mjti} ("Minimum joint torque index"). Ils minimisent ainsi la valeur des couples sur les moteurs du bras.

$$\mathrm{E}_{mjti} = \sum(\mathrm{J}^{\mathrm{T}} * \mathrm{F})^2 \tag{2.3}$$

Dans notre cas, nous introduisons un nouvel indice qui permet de calculer les variations d'énergie potentielle dues aux forces de gravité pour une pose donnée. Il représente le travail nécessaire par les articulations pour contrer ces forces. S'il est minimisé, on obtient des points d'équilibre où l'humain a plus de facilité à engendrer un mouvement sur son bras, car il est moins perturbé par l'effet de la gravité dans ces positions. Cet indice, que l'on nomme indice de gravité terrestre $s(q)$, est présenté en Sect. 4.3.2, tel que :

$$s(q) = \sqrt{\sum |\mathbf{g}(\mathbf{q_i})|^2}. \tag{2.4}$$

Avec $\mathbf{g}$ le vecteur de force de gravité qui est fonction des variables articulaires $\mathbf{q}$.

Comme pour Suay et al. nous calculons également l'énergie potentielle afin de ne garder que les poses dont le centre de gravité du bras se trouve en dessous du niveau des épaules (dans le cas de la position assise ou debout).

2.6 Conclusion

Ce chapitre nous aura permis de spécifier l'ensemble des enjeux cognitifs et physiques qui se mettent en place durant le transfert d'objets. Ce type d'échange implique que le donneur et le receveur soient totalement en coordination et attentifs l'un à l'autre tout en réalisant l'action. L'ensemble de ces études mettent en évidence le lieu, l'instant et la manière dont doit être transmis l'objet. En effet, la communication est essentielle pour informer à tout instant de ses intentions, de son implication et du possible lieu d'échange. L'analyse de l'objet et de l'environnement permet de le prendre convenablement pour optimiser la saisie du receveur. L'estimation du lieu de transfert permet de considérer la position de l'humain, ses propres capacités et permettre le confort de l'échange pour les deux acteurs. Puis, la planification et le contrôle du mouvement avant le transfert physique, également importants pour être réactif, s'adapter aux mouvements de l'humain et avoir un comportement fluide et prédictible. Enfin, l'échange physique implique un échange interactif de forces de préhension et de transfert de charge entre les deux individus. L'ensemble de ces travaux mettent en évidence la complexité de cette tâche. Malgré la technologie actuelle, il est encore très difficile pour les robots de détecter, comprendre et utiliser toutes ces informations pour avoir des transferts aussi robustes et fluides que ceux réalisés entre humains.

Les limitations fonctionnelles du robot Pepper ne nous permettent pas de prendre en compte toutes ces notions lors de l'échange. Cependant, il présente d'autres atouts qui nous permettrons de l'inclure plus facilement auprès de personnes âgées au sein même d'un EHPAD. Nos recherches proposent de générer des espaces de transfert jugés confortables et sûrs en utilisant l'indice de manipulabilité et en incluant un nouveau paramètre : l'indice de gravité terrestre (en considérant également l'énergie potentielle). Ces zones sont personnelles et adaptables aux caractéristiques physiques de l'individu. En effet, elles tiennent compte de son poids, de sa taille, de ses amplitudes articulaires, ainsi que de ses préférences. L'ensemble de nos développements ont pu être validés dans une étude utilisateur impliquant des personnes âgées, afin d'analyser la pertinence de la zone de confort générée.

Chapitre 3

Plateforme expérimentale

Sommaire

3.1 Introduction 41
3.2 Plateforme robotique 42
3.2.1 Un robot assistant 42
3.2.2 Capteurs et actionneurs utilisés 42
3.2.3 Station de recharge 44
3.3 Plateforme informatique 45
3.3.1 Version du robot et de son système d'exploitation 45
3.3.2 Kit de développement logiciel 47
3.3.3 Logiciel Choregraphe 47
3.3.4 Capacités informatiques nécessaires 48
3.4 Développement logiciel 49
3.4.1 Contexte et motivation 49
3.4.2 Une Interface de Programmation d'Applications 50
3.4.3 Utilisation du module DCM 51
3.4.4 Développement avec la méthodologie PID 51
3.5 Conclusion 52

3.1 Introduction

Ce chapitre a pour objectif de définir la plateforme de développement utilisée dans nos travaux de recherche. Dans un premier temps, nous présenterons la plateforme robotique que nous utilisons, ce qui revient à présenter le robot Pepper, ainsi que les capteurs et actionneurs. Dans un second temps, il est nécessaire de présenter la plateforme informatique. Cette seconde partie permet de spécifier le système d'exploitation (OS) du robot choisi pour nos développements, ainsi que les interfaces de programmation et de communication disponibles. Dans le cadre de nos recherches, nous nous sommes confrontés à certains problèmes qui nous ont amenés à créer une Interface de Programmation d'Applications (API) intermédiaire entre nos applications et celle fournie par le fabricant. Cette API nous a donc permis de séparer nos codes de la librairie constructeur et de contourner tous les problèmes d'incompatibilité. Nous présentons alors dans un dernier point les développements logiciels liés à cette interface de programmation afin d'expliquer au mieux nos motivations ainsi que sa mise en place. C'est dans cette partie que nous verrons comment nos algorithmes peuvent communiquer avec les dispositifs matériels du robot pour le contrôler ou vérifier son état.

3.2 Plateforme robotique

3.2.1 Un robot assistant

Pepper (voir figure 3.1) est un robot assistant développé par la société Softbank Robotics. Ce robot d'une taille de 1,20 mètre est équipé de 20 moteurs et de divers capteurs lui permettant d'analyser et d'interagir avec son environnement. Il a la particularité d'avoir une tablette fixée sur sa poitrine, facilitant les échanges avec l'humain, en plus de l'interaction vocale existante. En ce qui concerne ses déplacements, le fabricant a choisi d'utiliser une base mobile, contenant trois moteurs omnidirectionnels.

Le constructeur a imaginé et construit ce robot dans le but d'interagir directement avec les humains et de répondre à leurs besoins. Par exemple, dans le cas de la réception d'un hôtel, le robot peut répondre aux questions d'un client sur l'établissement.

Dans le contexte de notre étude, l'apparence humaine du robot favorise la compréhension générale de son fonctionnement et permet ainsi à l'humain de mieux appréhender l'interaction qu'il doit avoir avec lui. De plus, le fait qu'il soit sur une base mobile rend l'échange plus fluide et plus facile et limite le risque de chute du robot en avançant vers la personne. Pandey et Gelin [93] mettent en évidence les particularités auxquelles doit répondre un robot compagnon. En effet, l'aspect humanoïde et les modalités d'interaction sont importants pour ce type de robot. Afin d'être acceptés et plus attachants, ces robots doivent avoir un langage corporel et simuler des comportements sociaux similaires à ceux des humains [94]. Pepper a été conçu selon ces critères. Il est adapté au contact et à l'interaction avec le grand public. D'après Pandey et Gelin [93], les fabricants se sont concentrés sur son physique, son prix abordable, son interactivité, son autonomie et sur la sécurité. En ce qui concerne l'apparence, ils se sont focalisés sur certaines caractéristiques : la taille, la forme, le look et la voix. Tout d'abord, ils soulignent le fait que ce robot est androgyne. Cela signifie qu'il n'est pas possible d'associer le robot à un genre, homme ou femme, en raison de sa voix ou de son physique. Son corps est conçu pour avoir l'apparence d'un humain, tout en évitant une ressemblance trop exacte pour ne pas tomber dans la "vallée de l'étrange" [95]. Utiliser ce robot avec des personnes âgées nous a semblé évident afin que les interactions puissent se dérouler dans les meilleures conditions, sans préjugés ni craintes de leur part. C'est donc un compromis qui a dû être fait pour avoir un robot assistant capable d'interagir avec des personnes âgées en EHPAD malgré les nombreuses limites rencontrées.

3.2.2 Capteurs et actionneurs utilisés

La figure 3.1 montre les dispositifs matériels que nous utilisons du robot Pepper pour réaliser nos applications et interagir avec lui en toute sécurité. Pour plus de détails sur ce robot, nous proposons les descriptions présentées en annexe A.1.1 reprenant la documentation fournie par le constructeur.

Afin de percevoir son environnement et de permettre au robot d'estimer la présence et la posture d'un humain, nous utilisons ses capacités visuelles. Pour cela, nous récupérons le flux d'image provenant de sa caméra RGB frontale, ainsi que le flux de

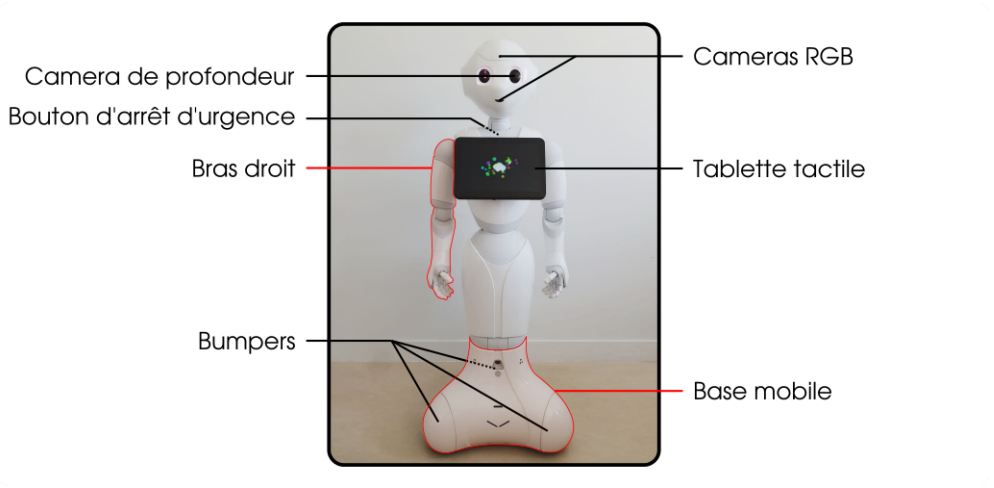


FIGURE 3.1 – Photo du robot Pepper de la société SoftBank Robotics, avec les dispositifs matériels utilisés.

données provenant de sa caméra de profondeur située dans son œil. La caméra RGB située dans sa bouche permet d'observer vers le bas, afin de détecter et d'estimer la position de marqueurs au sol. Ces marqueurs permettront la localisation du robot (voir 5.5.1).

Comme énoncé précédemment, le déplacement du robot sur le sol se fait à l'aide de sa base mobile. Celle-ci est composée de trois roues omnidirectionnelles lui permettant d'aller dans n'importe quel sens et direction. Dans le cadre de nos travaux, cette faculté permettra au robot de se déplacer aisément entre différents points d'intérêts. Elle permettra aussi de faciliter l'atteinte du point de transfert, et donnera une approche plus fluide et naturelle. Enfin, pour récupérer ou transmettre l'objet, nous aurons besoin de contrôler les moteurs qui composent les bras du robot. Pour cela, il est primordial de connaître leur état, et donc leurs positions articulaires à chaque instant. On utilise alors les données retournées par les encodeurs.

En ce qui concerne l'interaction avec Pepper, nous proposons dans un premier temps de se servir des fonctionnalités fournies par le constructeur. Cela permet d'utiliser ses haut-parleurs afin de communiquer verbalement grâce à la synthèse vocale existante. Ces fonctions permettent également au robot d'écouter, d'analyser et d'estimer les phrases ou mots prononcés par un humain. Selon l'application établie sur le robot, cette reconnaissance vocale permet de déclencher certaines actions ou réponses. Les applications développées utilisent donc les microphones positionnés sur le dessus de la tête du robot, ainsi que ses deux hauts-parleurs situés de part et d'autre de la tête (au niveau des oreilles). De plus, la tablette fixée au torse du robot est une interface graphique qui s'avère être très utile entre l'utilisateur et le robot. En effet, elle assure que l'information est bien transmise à la personne en appuyant graphiquement les annonces vocales par le biais d'images, vidéos, ou directement de textes. La tablette est en lien direct avec les modules du robot par le biais d'une API dédiée en JavaScript. Une application créée à partir du langage HTML ("HyperText Markup Language") et

JavaScript, permet de récupérer différentes informations que peut fournir la personne avec cette interface. Elle sera utile pour diffuser des questionnaires ou toute autre application pour faire le lien entre le robot et l'humain.

Lors de nos expérimentations, certains capteurs peuvent déclencher ou stopper des actions du robot. Pour cela, on vérifie si la surface tactile de la tablette a été touchée et on récupère également les informations provenant des trois capteurs tactiles de la tête. Ces informations permettent de stopper les mouvements du robot si nécessaire (sécurité) ou lancer l'affichage d'application. Il est essentiel que Pepper garde sa tête en direction de l'humain afin d'améliorer l'interaction visuelle et de permettre ainsi des échanges plus naturels avec l'humain [36]–[38]. Pour cela, nous analysons le flux d'images provenant de la caméra RGB frontale du robot pour détecter la présence d'un humain. Puis nous utilisons la motorisation de son cou pour garder la tête de l'humain dans le centre de l'image lorsque le robot se déplace dans sa direction. Pour des questions de sécurité, les informations provenant des trois bumpers situés autour de sa base mobile sont aussi récupérées afin de savoir à tout moment si le robot est en contact avec un obstacle. Il est aussi possible pour l'humain de stopper l'avancée du robot en les actionnant avec son pied. Enfin, en dernier recours l'utilisateur peut activer le bouton d'arrêt d'urgence situé dans la nuque de Pepper afin de couper toute son alimentation.

3.2.3 Station de recharge

Dans certaines de nos applications, nous utilisons la station de recharge du robot lui permettant de la reconnaître et d'aller se positionner dessus pour se recharger. La figure 3.2 permet d'avoir un aperçu de cette station de recharge, ainsi que des étapes à suivre pour que Pepper se recharge de manière autonome. Au départ, le robot aperçoit cette base par le biais de ses caméras. Il se déplace alors de manière à être juste devant. Arrivé en face, il se retourne, s'abaisse et recule afin de se placer correctement dessus. Enfin, il se redresse et est maintenant en charge en attendant l'instruction qui lui demandera de sortir de cette base. Concernant cette étape de sortie, le robot se baisse légèrement pour garder une meilleure stabilité, il va s'avancer légèrement pour descendre de la base, et se redresser. Il n'a plus qu'à attendre les prochaines instructions.



FIGURE 3.2 – Suite de photos prises permettant de visualiser les étapes que doit suivre Pepper pour se recharger de manière autonome sur sa station de recharge.

3.3 Plateforme informatique

3.3.1 Version du robot et de son système d'exploitation

Les deux robots Pepper que nous avons en notre possession sont des versions 1.8a avec pour système d'exploitation NAOqi 2.5. La version 1.8 du robot est la plus récente. La principale différence vient des caméras permettant d'avoir la carte de profondeur. Notre version de Pepper a une caméra de profondeur utilisant le principe de projection de lumière structurée (une Asus Xtion) dans ses yeux. Dans un œil, il y a le projecteur qui envoie les motifs lumineux et dans l'autre, il y a une caméra infrarouge qui permet de les visualiser. Enfin, l'ensemble du dispositif est capable de retourner directement la carte de profondeur estimée. La version 1.8, quant à elle, comprend deux caméras RGB, dans chacun de ses yeux, et peut donc utiliser les principes de la stéréovision avec ces deux caméras pour estimer la carte de profondeur.

La plus grande différence peut provenir des systèmes d'exploitation disponibles pour ce robot. En effet, que ce soit pour la version 1.8 ou pour la version 1.8a, il est possible de choisir entre deux systèmes d'exploitation : NAOqi 2.5 et QiSDK (NAOqi 2.9). D'après la comparaison réalisée sur ces deux versions par Chevallier et al. [96], ces deux OS ont le même objectif : développer ou installer des applications sur ce robot pour lui donner vie et lui permettre d'interagir avec les personnes et son environnement. Cependant, ils n'intègrent pas ces fonctionnalités de la même manière et il est important de comprendre la différence entre ces deux versions pour faire un choix adapté à la manière dont on utilise ces robots.

Description de Naoqi 2.5

NAOqi 2.5 est un système d'exploitation qui exploite Pepper comme une plateforme de développement basée sur le langage de programmation Python. Son environnement est assez libre et permet une programmation de bas niveau avec un accès direct aux capteurs et aux actionneurs. L'idée derrière cet OS est de permettre aux développeurs de programmer leurs services en utilisant les fonctionnalités existantes du

robot (fournies par le constructeur) ou en développant entièrement leurs propres applications. Il n'y a donc pas de service préétabli sur ce robot et cela leur permet d'imaginer, de concevoir et de réaliser leurs propres applications. Pour cela, le développeur a le choix entre le logiciel Choregraphe utilisant le langage Python et l'utilisation d'un SDK ("Sotfware Development Kit"). Il existe en effet un SDK Python et un SDK C++ qui permettent d'exécuter des scripts sur Pepper ou via un ordinateur externe. En ce qui concerne les applications réalisées sur la tablette, il faut utiliser le langage HTML et le JavaScript. Une communication interne est faite entre la tablette et le robot pour partager les données et charger des pages HTML, des images ou même des vidéos. Pour plus d'informations, il est possible de se référer à la documentation fournie par SoftBank Robotics concernant NAOqi 2.5 [1].

Description de QiSDK

QiSDK, ou NAOqi 2.9, propose une solution plus stable, mais bien moins flexible que ce qui est proposé avec NAOqi 2.5. Celui-ci est sous l'OS Android et est beaucoup plus adapté pour le marché. Il donne la possibilité de développer des applications avec une programmation de haut niveau. Il est possible par exemple de le faire bouger, parler et interagir avec les gens de manière simple à partir d'applications sur la tablette ou des fonctionnalités vocales du robot. Pour développer son application, le développeur doit utiliser Android Studio qui fournit un environnement de programmation assez complet en installant le plug-in associé à QiSDK. Ce plug-in permet de programmer directement sur Android Studio, en langage de programmation Kotlin ou Java, et d'utiliser un ensemble d'outils graphiques et de bibliothèques Java disponibles sur ce logiciel. Grâce à son éditeur graphique, il est possible de travailler directement sur l'écran de la tablette et donc de créer des applications plus complètes et plus interactives. Pour plus d'informations, il est possible de se référer à la documentation fournie par SoftBank Robotics concernant QiSDK [2] (NAOqi 2.9).

Naoqi 2.5 ou QiSDK

Il faut tout d'abord savoir que ces deux systèmes d'exploitation peuvent être installés sur le robot Pepper quelle que soit sa version (1.8 ou 1.8a). Il est spécifié toutefois que l'intégration de NAOqi 2.5 semble la plus idéale pour Pepper 1.8a et QiSDK pour Pepper 1.8. Pour mettre à jour Pepper (de NAOqi 2.5 vers NAOqi 2.9), il suffit de contacter le support de SoftBank Robotics. Cependant, il est précisé qu'il ne sera pas possible de revenir sur la version antérieure (NAOqi 2.5). Il faut donc bien réfléchir à notre utilisation de Pepper avant de se lancer dans toute démarche permettant de le mettre à jour. Nous souhaitons avoir un total accès aux données provenant des capteurs et un contrôle sur chacun des moteurs du robot. De plus, nous serons amenés à travailler avec des librairies externes, dont le langage de programmation n'est disponible qu'en

1. https://developer.softbankrobotics.com/pepper-naoqi-25/naoqi-developer-guide/getting-started
2. https://developer.softbankrobotics.com/pepper-qisdk/qisdk

Python ou C++. C'est donc tout naturellement que nous avons choisi d'utiliser la version NAOqi 2.5 pour garder cette flexibilité et d'utiliser ce robot comme "plateforme de développement robotique" pour nos applications de recherche.

3.3.2 Kit de développement logiciel

Comme dit précédemment, le constructeur fournit un SDK Python [3] et un SDK C++ [4] pour pouvoir échanger avec le système d'exploitation NAOqi 2.5 du robot. Grâce à l'API qu'il renferme, il sera possible d'obtenir l'ensemble des données brutes des capteurs, de contrôler chacune des articulations du corps, ainsi que la base mobile. Dans ce SDK, nous utilisons les fonctions fournies qui permettent de faire appel aux différents modules du robot. Chaque module représente un service auquel on peut accéder. Ces services peuvent permettre d'accéder à la mémoire du robot, de récupérer le flux des caméras (RGB et de profondeur), d'allumer des LED ("Light-Emitting Diode"), etc.

Le langage C++ est un outil reconnu pour sa puissance, car il permet aux développeurs de gérer et d'utiliser au mieux les ressources informatiques du système. Cette capacité permet donc d'optimiser au maximum nos algorithmes et d'avoir une vitesse d'exécution bien plus élevée que si nous utilisions le langage Python. De plus, les librairies externes que nous utilisons sont codées principalement en langage C++. Dans le cadre de nos développements, nous avons besoin d'un ordinateur externe ayant une puissance de calcul plus importante que le robot et comprenant une carte graphique spécifique de la marque NVIDIA. Ce type de carte graphique nous permet d'optimiser certains calculs nécessaires par exemple à l'utilisation de réseaux de neurones pour la reconnaissance du squelette humain dans une image RGB. De plus, le robot Pepper a une multitude de services activés sur son ordinateur interne et il met à jour en permanence les variables de sa mémoire. Cela peut rendre nos applications plus lentes. Il est donc nécessaire d'utiliser un ordinateur externe pour les applications robotiques recherchées. Nous utilisons alors le SDK C++ de NAOqi 2.5 pour communiquer et contrôler le robot via un ordinateur externe.

3.3.3 Logiciel Choregraphe

Choregraphe est le logiciel graphique du constructeur basé sur Python permettant de programmer et de déployer rapidement des applications sur le robot Pepper ou Nao. En effet, c'est une plateforme de programmation robotique permettant d'éditer et de générer des animations, des comportements, des dialogues, etc. Sur ce logiciel, il y a aussi un simulateur qui permet de contrôler directement les actionneurs et un éditeur permettant de générer des poses spécifiques du robot.

En bref, c'est un logiciel assez complet pour développer rapidement de petites applications sur le robot Pepper. Cependant, on ne peut pas se satisfaire de cette interface de développement et du langage de programmation Python imposé par le logiciel.

3. https://developer.softbankrobotics.com/pepper-naoqi-25/naoqi-developer-guide/sdks/python-sdk
4. https://developer.softbankrobotics.com/pepper-naoqi-25/naoqi-developer-guide/sdks/c-sdk

Ce langage de programmation est directement “exécutable” en l'état, et n'est pas compilé par l'ordinateur avant. Cela veut donc dire que toutes les instructions du code sont lues les unes après les autres et cela rend globalement les applications plus lentes. Pour des applications plus conséquentes, demandant plus de mémoire et de puissance de calcul, la rapidité d'exécution dépend des capacités informatiques internes du robot. C'est pourquoi, nous utilisons un ordinateur externe avec le SDK C++ pour développer nos algorithmes. Cependant, dans certains cas, il peut être préférable d'utiliser des fonctionnalités proposées par Choregraphe. En effet, il ne nous est pas toujours possible de contrôler le robot avec toutes les fonctionnalités possibles par ce logiciel. C'est le cas par exemple des fonctions dédiées à l'utilisation de la tablette qui ne sont pas disponibles dans le SDK C++. Choregraphe nous permet donc de développer une application qui se lancera au démarrage du robot et qui permettra à nos algorithmes de lancer certains services. Pour cela, on utilisera des variables définies dans la mémoire du robot et accessibles en appelant le module “ALMemory” [5].

Sur ce principe, nous avons choisi de faire différentes “boîtes” qui remplirons chacune des services différents. Elles vont permettre de lancer facilement la reconnaissance vocale, le retour ou la sortie de la station de chargement, générer l'affichage sur la tablette (application HTML, questionnaires, fond d'écran, pages internet...) ou encore envoyer un E-mail. Il est possible d'en ajouter, mais l'idée est d'en avoir le moins possible sur le robot afin d'en déporter au maximum sur l'ordinateur externe.

3.3.4 Capacités informatiques nécessaires

Dans le cadre de nos travaux, des applications de reconnaissance d'humain et d'estimation de pose ont dû être utilisées. Pour leurs bons fonctionnements et surtout afin d'avoir une vitesse de calcul plus importante, il faut un ordinateur qui contient une carte graphique NVIDIA compatible avec la librairie CUDA (“Compute Unified Device Architecture”). CUDA est une API qui permet d'envoyer des calculs à la carte graphique de l'ordinateur, le GPU (“Graphics Processing Unit”). Il évite donc de passer par le processeur, le CPU (“Central Processing Unit”) qui est beaucoup plus lent pour des applications identiques. Dans notre cas, l'ordinateur utilisé tourne sous le système d'exploitation de Linux Ubuntu 18.04 et a une “NVIDIA Quadro T2000”. Concernant la version des pilotes installés, pour NVIDIA nous avons la version 465.19.01 et pour CUDA la version 10.1.243. Tout dépend de la compatibilité nécessaire avec les librairies externes utilisées, mais dans notre cas, cette combinaison de versions fonctionne.

5. http://doc.aldebaran.com/2-5/naoqi/core/almemory.html?highlight=almemory

3.4 Développement logiciel

Afin de communiquer avec le système d'exploitation du robot, nous utilisons le SDK C++ fourni par le constructeur. Il permet de déporter des algorithmes plus ou moins coûteux en capacité de calcul et de stockage de l'information directement sur un ordinateur externe. Il est possible alors d'intégrer des librairies externes qui sont utiles pour nos développements. Cependant, ce SDK ne permet pas d'intégrer toutes les librairies que l'on souhaite utiliser. C'est pourquoi, nous avons fait le choix de développer notre propre API permettant une communication interne entre nos applications et les fonctions fournies par le SDK du constructeur.

3.4.1 Contexte et motivation

Malgré les fonctionnalités offertes par le système d'exploitation du robot, on se trouve rapidement limité et contraint par les fichiers internes au SDK et l'utilisation de "qiBuild" [6]. QiBuild est un outil qui permet de gérer plusieurs projets, ainsi que leurs dépendances. Il fait le lien direct entre ces projets et le SDK ou toute autre librairie installée sur la machine. En fait, il utilise CMake [7] (abréviation de "Cross platform Make") qui est un outil permettant de gérer la compilation de projet, en vérifiant que toute dépendance est bien disponible par l'application. Il est donc possible de construire et de compiler notre projet en utilisant qiBuild afin d'avoir accès aux fonctionnalités du SDK. Cependant, le code source du SDK n'est pas fourni, il est déjà compilé et donné en langage binaire. Quelle que soit la plateforme, utilisée (Linux, Windows, macOS), il est conseillé par le constructeur d'avoir des versions d'OS plus anciennes afin d'éviter toute incompatibilité.

Dans notre cas, nous utilisons le système d'exploitation Linux Ubuntu 18.04 et d'après le guide d'installation [8] il nous est conseillé de prendre la version Ubuntu 14.04. Cette version peut nous poser des problèmes de compilation sur des librairies externes plus récentes. En effet, l'intégration de librairies externes était souvent difficile, voire impossible avec ce SDK. Des erreurs apparaissaient au moment de l'exécution de notre code compilé, visiblement à cause de différences d'ABI ("Application Binary Interface"). L'ABI est une convention qui permet au programme, après compilation du code, d'accéder aux données binaires et de les lire convenablement. Il faut spécifier à notre compilateur quel est l'ABI utilisé. Or, cet ancien ABI pose problème pour lire les fichiers binaires des librairies plus récentes. Notre objectif est donc de contourner ce problème et de permettre à nos applications d'être totalement indépendantes de NAOqi et de son SDK.

La figure 3.3 schématise le problème observé lorsque nous cherchons à utiliser des librairies externes dans nos applications et à faire communiquer directement les algorithmes aux différents modules de NAOqi.

6. http://doc.aldebaran.com/qibuild/index.html
7. https://cmake.org/
8. http://doc.aldebaran.com/2-5/dev/cpp/install_guide.html

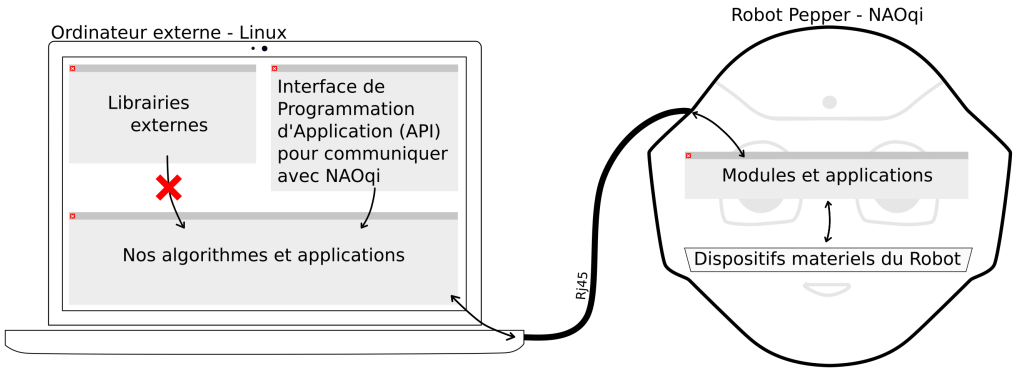


FIGURE 3.3 – Explication concernant le problème pour utiliser des librairies externes.

3.4.2 Une Interface de Programmation d'Applications

L'idée est donc de créer une API, permettant de dissocier nos algorithmes des fonctions du SDK de Pepper. On met alors en place un serveur de communication UDP/IP (pour "User Datagram Protocol" et "Internet Protocol") entre deux algorithmes bien distincts. Ce protocole de communication a la particularité de ne pas attendre de retour du serveur lorsqu'il envoie un paquet de données. Les cadences d'envoi et de réception sont ainsi plus élevées. D'un côté, on aura un algorithme qui utilisera le SDK de Pepper pour récupérer les données des capteurs et contrôler les actionneurs. Cet algorithme a le rôle de serveur, il devra donc être toujours actif et attendra les instructions données par nos applications qui eux, ont le rôle de clients. Effectivement, d'un autre côté, on a un second programme qui inclus les applications et un driver permettant directement d'échanger avec le serveur. La connexion entre ces deux algorithmes est établie en interne sur l'ordinateur.

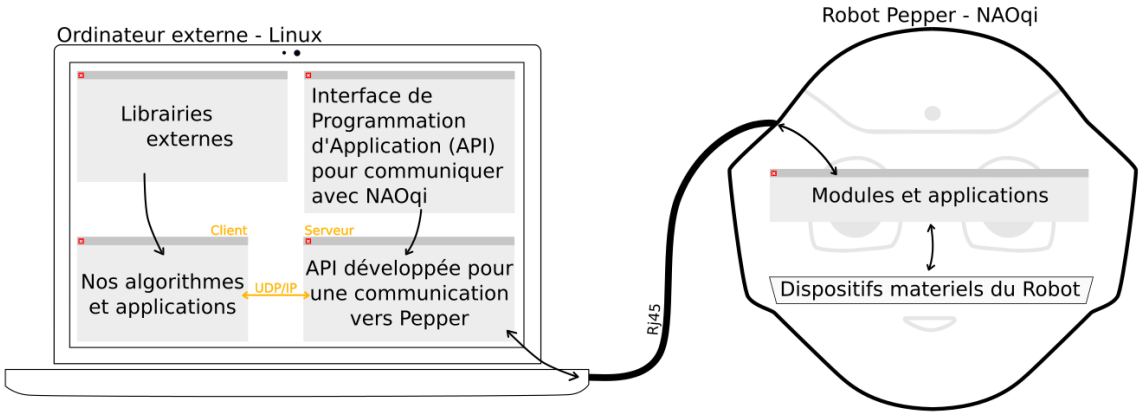


FIGURE 3.4 – Explication concernant la solution apportée pour utiliser des librairies externes.

Ce serveur est donc une sorte de passerelle entre l'ordinateur externe et le robot Pepper. Le schéma de la figure 3.4 permet de visualiser l'architecture générale de com-

munication mise en place pour transmettre nos commandes jusqu'au robot Pepper. Cette solution nous permet de nous affranchir le plus possible des fonctionnalités de NAOqi dans nos algorithmes et de pouvoir installer et utiliser des librairies externes sans avoir de soucis de compatibilité.

3.4.3 Utilisation du module DCM

DCM [9] ("Device Communication Manager") est un module qui est intégré au système d'exploitation de Pepper et il communique avec la grande majorité des appareils électroniques du robot (cartes, capteurs, actionneurs...). En effet, il peut envoyer des commandes directement aux actionneurs et il récupère des données provenant de certains capteurs. Ils sont ainsi mis à jour dans la mémoire du robot dans le module "ALMemory". En interrogeant ce dernier, puis en récupérant les données les unes après les autres, la cadence de lecture est trop faible pour permettre de contrôler convenablement le robot (environ 20ms à 30ms pour lire les données provenant des 17 encodeurs). On va donc lire ces données directement à partir du module DCM qui est beaucoup plus proche des dispositifs électroniques et qui nous permettra de gagner de précieuses millisecondes. Pour cela, nous utilisons le module "mc_naoqi_dcm" [10] développé par Bolotnikova et al. [97]. En interrogeant ce module, on pourra récupérer l'ensemble des données provenant des encodeurs en seulement 1ms. Cependant, il faut noter que les valeurs des capteurs eux-mêmes ne sont mises à jour que toutes les 12ms (fréquence de la boucle du module DCM). On pourra donc s'assurer d'avoir de nouvelles valeurs toutes les 12ms.

3.4.4 Développement avec la méthodologie PID

Nos applications utilisent la méthodologie PID [11] (Packages Integral Development) développée par R. Passama et B. Navarro. L'objectif est de créer un environnement où l'utilisateur peut développer l'ensemble de ses projets. Cela nous est utile pour permettre une installation automatisée des dépendances qui sont nécessaires au bon fonctionnement de nos applications. En effet, cet environnement est basé sur les API et les commandes CMake et cela permet d'automatiser le processus de développement de projets en langage C/C++. Si l'utilisateur le souhaite, il lui est possible de télécharger indépendamment les applications sans avoir à créer d'environnement. L'application est ainsi plus flexible selon les besoins du développeur.

9. https://developer.softbankrobotics.com/pepper-naoqi-25/naoqi-developer-guide/naoqi-apis/dcm
10. https://github.com/jrl-umi3218/mc_naoqi_dcm
11. https://pid.lirmm.net/pid-framework/

3.5 Conclusion

Dans le cadre de notre étude, nous avons deux robots Pepper de la société SoftBank Robotics. Leur système d'exploitation nous permet de contrôler l'ensemble des dispositifs matériels du robot par le biais de modules de programmations. Pour communiquer avec ces modules, nous utilisons le SDK C++ fourni par le constructeur. L'avantage avec ce dernier est que nous pouvons solliciter les capacités informatiques d'un ordinateur externe plus adaptées pour nos développements. Nous avons développé notre propre API, pour contourner les problèmes d'incompatibilité entre les fichiers binaires disponibles dans ce SDK et les fichiers provenant des librairies externes dont nous avons besoin. Cette API nous permet donc de mettre en place un serveur entre nos algorithmes et les fonctions fournies par le constructeur qui permet les échanges avec le robot. Afin d'optimiser nos applications et la gestion des dépendances, nous avons utilisé la méthodologie PID qui nous permet d'automatiser le processus de développement de projets C/C++. Pour certaines applications (reconnaissance vocale, affichage sur la tablette, ...) nous avons préféré utiliser l'interface de Choregraphe car il était plus simple de passer par ce biais. Nous avons également fait le choix de rester sous l'OS NAOqi 2.5 comme plateforme de développement afin d'avoir accès à tous les capteurs et actionneurs du robot. Cependant, cela implique certaines limitations pour le développement d'applications Android sur la tablette (plus avancé avec NAOqi 2.9).

Chapitre 4

Modélisation et analyse de l'espace de transfert

Sommaire

4.1 Introduction 54
4.2 Espace atteignable 55
4.2.1 Modèle géométrique 55
4.2.2 Génération de l'espace 56
4.2.3 Discrétisation 57
4.2.4 Zone interdite 58
4.3 Espace ergonomique 59
4.3.1 Indice de manipulabilité 59
4.3.2 Indice de gravité terrestre 60
4.3.3 Mise en commun des espaces 62
4.3.4 Enlever les doublons 63
4.3.5 Discussions 64
4.4 Conclusion 65

4.1 Introduction

Ce chapitre explique la manière dont nous avons procédé pour générer un espace de transfert ergonomique pour l'humain. L'objectif est de définir un espace adapté pour garantir des transferts plus sûrs et plus confortables. Pour cela, nous prenons en compte les paramètres concernant ses préférences, ses amplitudes articulaires, sa taille et son poids.

Pour commencer, ce chapitre présente le modèle biomécanique pris en compte pour la modélisation des membres supérieurs de l'humain. Il nous permettra alors de générer l'espace atteignable par ses mains en testant une multitude de configurations articulaires sur ses bras. On obtient alors un nuage de point adapté selon sa morphologie et ses troubles articulaires. Avant de passer au tri de cette zone pour obtenir une zone confortable et sûre, nous discrétisons ce nuage de point ce qui permettra par la suite de supprimer les doublons. La prochaine étape consiste à analyser cet espace et à garder seulement les points qui sont les plus confortables pour recevoir ou donner un objet. En effet, la configuration du bras obtenue pour chacune des positions de cet espace nous permet de mesurer, par le biais d'indices, la manipulabilité et l'effet de la gravité sur ses membres supérieurs pour ainsi minimiser l'effort et garantir un transfert plus confortable et sûr.

4.2 Espace atteignable

Afin de modéliser et d'analyser l'espace atteignable par les mains de l'humain, nous proposons de décomposer ce travail en deux parties. La première consiste à faire la modélisation géométrique des membres supérieurs de la personne qui seront en interaction avec le robot. La seconde partie utilise ce modèle afin de générer les positions atteignables par les mains qui seront ensuite discrétisées. Les étapes définissant cet espace atteignable sont représentées dans le schéma global de la figure 5.17.

4.2.1 Modèle géométrique

Pour générer cet espace de travail, nous nous appuyons sur un modèle géométrique basé sur la biomécanique du corps humain. Ce modèle détermine la position de la main de l'humain dans l'espace selon les valeurs articulaires attribuées aux articulations de son bras. Cet ensemble de valeurs donne une "configuration" des articulations sur les membres supérieurs. Cette configuration a toute son importance pour estimer les indices de manipulabilité et de gravité calculés dans la partie suivante. Toutes les valeurs articulaires seront donc retenues avec les positions de la main qui en résultent.

Le corps humain possède plus de 600 muscles, et ils lui permettent d'effectuer une multitude de mouvements complexes. Dans le cadre de nos recherches, nous nous intéresserons uniquement à la motricité des membres supérieurs : les bras et les mains. Pour déterminer les articulations à prendre en compte dans la génération du nuage de points et pour disposer des valeurs limites articulaires, nous considérons le modèle biomécanique de Kapandji [98]. Ce modèle nous donne les mouvements possibles des articulations du bras de l'humain. Dans notre cas, nous simplifions ce modèle en ne considérant que trois rotations pour l'épaule et une pour le coude. En effet, nous ne prenons pas en compte les translations possibles de l'épaule et les articulations responsables de la motricité du poignet et des doigts de la personne. L'intérêt de cette simplification est que nous ciblons les articulations responsables de la projection dans l'espace de la main. On garde ainsi le poignet dans sa position de repos, et on ne prend pas en compte les translations possibles de l'épaule. Ces translations sont utilisées seulement dans des cas particuliers ou dans des conditions aux limites. Leur usage est donc considéré comme inconfortable pour récupérer un objet dans l'espace généré. Les articulations retenues, ainsi que leurs amplitudes sont schématisées dans la Fig. 4.1.

Notre objectif est d'établir un modèle personnalisé afin que l'espace de travail généré soit adapté aux capacités et aux caractéristiques physiques de la personne. Nous avons d'abord besoin de connaître la taille et le poids de la personne. En effet, le premier paramètre nous permet, selon les tables anthropométriques [99] [100], de donner une approximation de la taille des segments des membres supérieurs de l'humain. Le paramètre du poids est utilisé pour calculer l'énergie potentielle et l'indice de gravité terrestre en fonction de la configuration articulaire des bras. En effet, comme pour la longueur des membres, nous utilisons le modèle de Winter [101] qui donne une approximation de la masse des membres du corps humain en fonction du poids de la

personne. Enfin, afin de s'adapter au mieux à la personne en cas de troubles articulaires, il est nécessaire de connaître les amplitudes maximales qu'elle peut atteindre. Nous proposons tout d'abord d'initialiser ces valeurs avec le modèle biomécanique de Kapandji [98]. Celui-ci donne une référence sur les amplitudes articulaires de ces quatre degrés de liberté sélectionnés pour une personne ne présentant aucun trouble (voir figure 4.1).

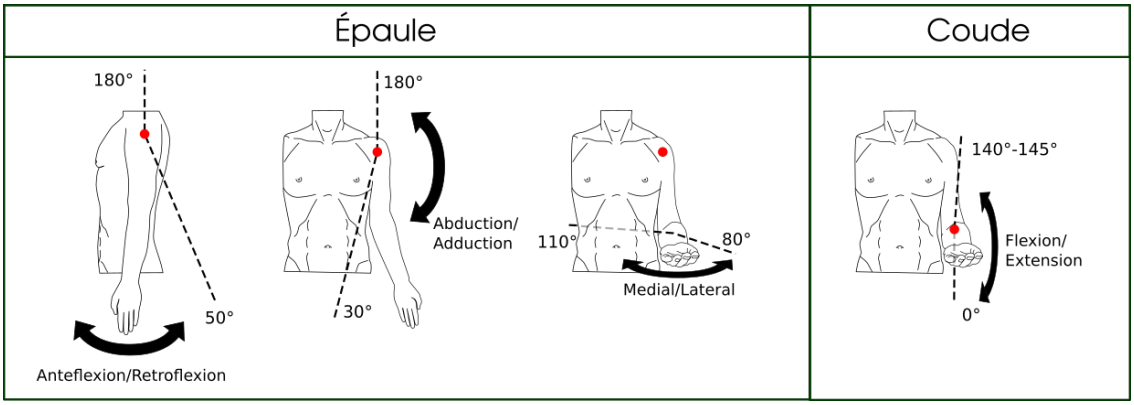


FIGURE 4.1 – Modélisation des articulations prises en compte pour notre modèle biomécanique, ainsi que leurs amplitudes maximales retenues. Le point rouge sur chacun des modèles indique l'articulation considérées pour les amplitudes dessinées.

Après avoir défini le modèle, nous pouvons calculer la pose du repère de la main exprimé par rapport au repère de référence situé entre les deux épaules et noté R_0. Dans cette thèse, tous les nuages de points associés à l'humain seront générés à partir de ce repère.

Pour plus de détails concernant la description géométrique des membres supérieurs de l'humain, nous proposons en annexe A.2.1 une représentation de l'humain, ainsi que la disposition des différents repères définit par notre modèle. Le tableau de l'annexe A.2.2 décrit la position des repères les uns des autres.

4.2.2 Génération de l'espace

Ayant obtenu le modèle biomécanique des membres supérieurs de l'humain, il est possible de générer tous les points accessibles par ses mains. Pour cela, nous faisons tourner les articulations autour des axes sélectionnés précédemment. Pour générer l'espace atteignable, on enregistre dans un premier temps les limites articulaires propres à la personne. On choisit ensuite le pas d'échantillonnage permettant de définir toutes les valeurs articulaires à tester, de la valeur minimale à la valeur maximale. Par exemple, la valeur de l'articulation du coude peut être modifiée entre 0° et 140°. Avec un pas d'échantillonnage de 5°, nous prendrons toutes les valeurs comprises entre 0° et 140° tous les 5°. À partir du modèle défini dans la section précédente, nous pouvons déterminer – pour chaque valeur articulaire, sur les quatre degrés de liberté – la

position dans l'espace cartésien de la main. Pour chaque point généré, les quatre valeurs articulaires sont enregistrées.

4.2.3 Discrétisation

Après cette première étape, nous devons prendre toutes les coordonnées enregistrées et les discrétiser sous la forme de voxels. Un voxel est le plus petit élément d'une "image" tridimensionnelle, souvent représenté sous la forme d'un cube. En règle générale, il est caractérisé par sa position en coordonnées cartésiennes (x, y et z qui définissent la position du point en son centre), sa couleur et sa densité. L'idée est donc de définir la taille de ce plus petit élément et d'approximer l'ensemble des points contenus dans ces voxels par leur position. Pour chacun des points sauvegardés dans les voxels, on garde les informations concernant la configuration articulaire du bras pour atteindre cette position. Cette discrétisation permettra, lorsque l'analyse de ce nuage de points sera effectuée, de supprimer plus facilement les doublons. La figure 4.2 montre cette étape de discrétisation de points contenus dans un voxel. Dans notre cas, le volume du voxel est de $27cm^3$ (arête de $3cm$).

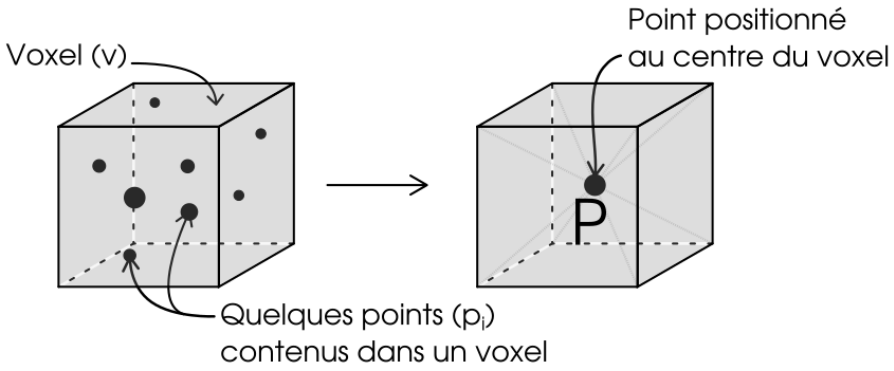


FIGURE 4.2 – Ensemble de points contenus dans un voxel et discrétisés en son centre.

Si l'on considère un ensemble de points $\mathbf{p_i}$, dont i est l'indice allant de 1 à n (le nombre total de points), contenu dans un seul et même voxel noté v. Chacun de ces points est défini selon les quatre variables articulaires du modèle géométrique du bras (Q1, Q2, Q3 et Q4). On note alors $\mathbf{p_i}(Q1_i, Q2_i, Q3_i, Q4_i)$. Après cette discrétisation, l'ensemble des points enregistrés au centre de ce voxel complète les informations de la matrice **P**. Cette matrice contient l'ensemble des configurations articulaires possibles pour atteindre ce voxel (point positionné au centre du voxel) tel que :

$$\mathbf{P} = \begin{bmatrix} Q1_1, Q2_1, Q3_1, Q4_1 \\ Q1_2, Q2_2, Q3_2, Q4_2 \\ \cdots \\ Q1_i, Q2_i, Q3_i, Q4_i \\ \cdots \\ Q1_n, Q2_n, Q3_n, Q4_n \end{bmatrix}$$

4.2.4 Zone interdite

Pour se rapprocher le plus possible de la zone réellement atteignable par le bras de l'humain, nous avons établi quelques contraintes spatiales. Ces contraintes sont définies par un parallélépipède rectangle. Il définit la zone où se trouve le corps de la personne et permet donc d'éviter de prendre en compte les positions de bras qui fusionnent ou sont beaucoup trop proche du corps. On va focaliser notre attention sur la position du coude et celle de la main pour une configuration donnée, et vérifier que ces deux éléments se retrouvent en dehors de cette zone interdite. Si l'un des deux se trouve dans ce parallélépipède rectangle, le point d'atteinte n'est pas considéré comme atteignable. Il ne sera pas comptabilisé dans le nuage de points final.

Pour les limites de ce parallélépipède rectangle, nous avons choisi de prendre 90% de la longueur séparant R_0 d'une épaule sur les axes x et y (90% de l_{os}). Puis sur l'axe z, les limites ont été définies de façon à supprimer les points de la tête jusqu'au pied. Pour cela, on utilise encore une fois les tables anthropométriques [99] [100]. La figure 4.3 représente la zone interdite décrite dans cette section.

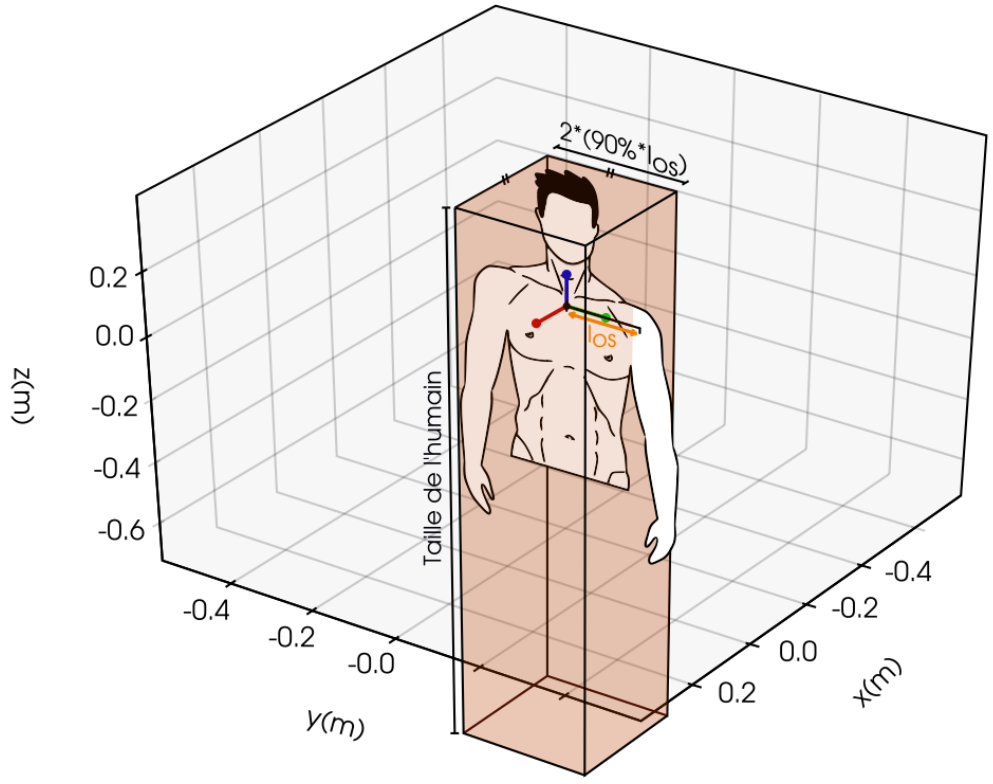


FIGURE 4.3 – Représentation graphique du parallélépipède rectangle définissant la zone interdite où les points de l'espace de transfert ne sont pas enregistrés.

4.3 Espace ergonomique

La partie précédente explique comment obtenir la zone atteignable par la personne en prenant en compte les propriétés géométriques du haut du corps du patient. On prend en considération sa morphologie, ainsi que son poids et sa taille. On prend également en compte les limites articulaires de cette personne, afin d'obtenir le nuage de points le plus fidèle possible de sa réalité. Une fois cet espace atteignable défini, il est important d'extraire les points formant le sous-espace assurant un transfert sûr et confortable. Les étapes suivies dans notre framework pour définir cet espace ergonomique sont représentées dans le schéma global de la figure 5.17.

4.3.1 Indice de manipulabilité

Pour transmettre un objet à un humain, on doit s'assurer que la zone de transfert soit sûre. Pour cela, on propose de mesurer l'indice de manipulabilité proposé dans de précédents travaux [71], [102], [103]. Cet indice est une métrique qui informe sur la facilité à changer de pose à partir d'une pose donnée.

En effet, dans certaines configurations, lorsque l'on se rapproche des valeurs limites d'une des articulations, cet indice diminue, car la possibilité de changer de position est plus limitée. Le fait de se retrouver dans une configuration où l'on peut changer rapidement et simplement la position de notre main permet de définir cette notion de manipulabilité. L'objet sera, en effet, plus manipulable et l'humain aura une plus grande facilité à s'adapter et à être réactif dans ces configurations-là. Garantir une plus grande manipulabilité, c'est augmenter la sûreté du transfert réalisé et rendre le transfert plus confortable lors de la saisie de l'objet.

Pour calculer cet indice, nous utilisons la matrice jacobienne $\mathbf{J}$ qui est calculée à partir du modèle géométrique. Cette matrice relie le vecteur vitesse de déplacement du membre terminal, noté $\dot{\mathbf{x}}$, au vecteur des vitesses angulaires des moteurs, noté $\dot{\mathbf{q}}$:

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{4.1}$$

L'équation (4.2) présente le calcul effectué pour déterminer l'indice de manipulabilité $m(q)$, en fonction des variables articulaires q :

$$m(q) = \sqrt{det(\mathbf{J}(\mathbf{q})\mathbf{J}^{\mathbf{T}}(\mathbf{q}))} \in \Re^{+} \tag{4.2}$$

Pour maximiser la manipulabilité, il nous faut dans un premier temps, calculer cet indice pour chacun des points de l'espace atteignable. Puis choisir les points qui ont les plus grandes valeurs de manipulabilité. Pour cela, on définit un seuil, exprimé en pourcentage de la valeur maximale des indices calculés, pour ne garder que les points ayant pour indice une valeur supérieure à ce seuil. Les autres points ne sont pas conservés dans le nuage de points final. Par exemple, si l'on défini un seuil à 90% de la valeur maximale, on obtient le nuage de points de la figure 4.4.

Pour mieux visualiser les zones que l'on peut obtenir, il est proposé en annexe B.1 plusieurs rendus avec différentes valeurs de seuils.

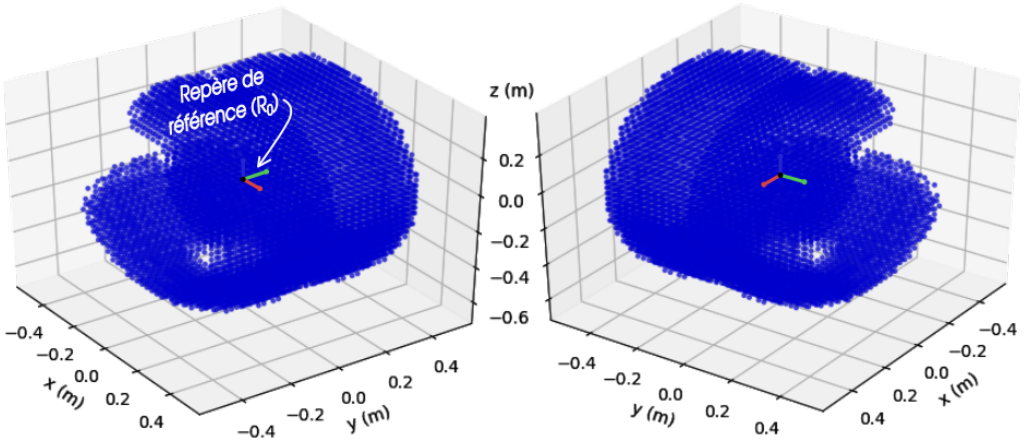


FIGURE 4.4 – Deux vues de l'ensemble des points ayant un indice de manipulabilité élevé (supérieur à 90% de la valeur maximale) par rapport aux membres supérieurs. Le repère de référence R_0 est situé entre les deux épaules de l'homme.

4.3.2 Indice de gravité terrestre

Nous présentons un nouvel indice qui permet d'affiner l'espace de transfert. Il s'agit de l'indice de gravité terrestre, qui est une métrique informant de l'impact qu'exercent les forces de gravité sur le bras dans une configuration donnée. Pour calculer cet indice, on s'intéresse à l'équation de mouvement du système défini à partir des équations de Lagrange (ou équations d'Euler-Lagrange). Ces équations décrivent la dynamique d'un système en fonction des forces extérieures qui agissent sur lui.

$$\frac{d}{dt}\frac{\delta \mathrm{L}}{\delta \dot{\mathbf{q}}} - \frac{\delta \mathrm{L}}{\delta \mathbf{q}} = \mathbf{F_q} \tag{4.3}$$

Pour que le bras reste dans une même position, il faut compenser les efforts extérieurs qui agissent sur le bras. Dans ce cas, les forces généralisées $\mathbf{F_q}$ sont l'ensemble des forces qui s'exercent sur le système. En position statique, les seules forces appliquées sur le bras concernent celles liées à la gravité. En effet, le bras de l'humain n'est soumis qu'à la force $\overrightarrow{\mathrm{P}}$ induite par la masse m de son bras et l'accélération de la pesanteur $\overrightarrow{g}$.

$$\sum \overrightarrow{\mathrm{F}_{ext}} = \overrightarrow{\mathrm{P}} = m * \overrightarrow{g} \tag{4.4}$$

Cette force $\overrightarrow{\mathrm{P}}$ est dite conservative, car le travail W qui en résulte ne dépend pas du chemin suivi, mais seulement du point de départ et du point d'arrivée. Dans ce cas, il est noté que la variation du travail δW est l'opposé de la variation d'énergie potentielle $\nabla \mathrm{E}_p$ entre deux positions distinctes.

$$\delta \mathrm{W} = -\nabla \mathrm{E}_p \tag{4.5}$$

De plus, l'expression du Lagrangien L dépend de l'énergie cinétique E_c et de l'éner-

gie potentielle E_p qui agissent sur le bras :

$$L = E_c - E_p \tag{4.6}$$

L'énergie cinétique étant nulle dans notre cas et d'après les équations 4.6 et 4.3, $\mathbf{F_q}$ est directement liée aux variations de l'énergie potentielle E_p. On notera par la suite que $\mathbf{F_q}$ correspond au vecteur des forces de gravité, noté $\mathbf{g(q)}$. On obtient alors :

$$\mathbf{g(q)} = \frac{\delta E_p}{\delta \mathbf{q}} = \nabla_q E_p \tag{4.7}$$

Le vecteur des forces de gravité $\mathbf{g(q)}$ nous renseigne sur les forces qu'exerce la gravité sur chacune des articulations du bras. D'après l'équation 4.4 il nous faut compenser ces forces en exerçant un travail opposé à ces variations d'énergie qui tirent le bras vers le bas. Afin de pouvoir minimiser ce vecteur, on propose de calculer sa norme. On définit ainsi l'indice de gravité terrestre $s(q)$.

$$s(q) = \|\mathbf{g(q)}\|_2 = \sqrt{\sum_{i=1}^{4} |\mathbf{g(q_i)}|^2}. \tag{4.8}$$

Minimiser l'indice de gravité terrestre $s(q)$ permet de réduire l'effet de la gravité sur les articulations du bras. Il est donc plus simple de maintenir la position de la main, car les efforts nécessaires pour compenser les forces liées à la gravité sont minimisés. Cependant, le repère de référence étant situé au niveau de R_0, on obtient donc une symétrie par rapport au plan horizontal en ce point. On souhaite garder les points les plus proches du sol et donc en dessous de ce plan. On propose donc de calculer l'énergie potentielle sur le bras et de ne garder que les valeurs des indices pour lesquels l'énergie potentielle est négative. Ces valeurs correspondent aux positions pour lesquelles le centre de masse du bras se trouve en dessous des épaules. L'énergie potentielle du bras est donc calculée selon l'équation suivante :

$$E_p = g * (m_1 * Z_1(\mathbf{q}) + m_2 * Z_2(\mathbf{q})) \tag{4.9}$$

avec g l'accélération de la pesanteur à la surface de la Terre, m_1 la masse du bras (segment 1) et m_2 prenant en compte la masse de l'avant-bras, ainsi que celle de la main (segment 2). Enfin, Z_1 et Z_2 sont les hauteurs respectives des centres de masse des segments 1 et 2.

Les masses ainsi que les centres de masse sont estimés à partir du modèle de Winter [101] qui donne une approximation de leur valeur en fonction du poids général de la personne. Nous utilisons le modèle géométrique pour connaître les coordonnées des centres de masse par rapport au repère de référence R_0 et pour les exprimer en fonction des quatre valeurs articulaires considérées.

À partir des équations 4.8 et 4.9, pour chacun des points qui composent l'espace atteignable, nous enregistrons l'indice de gravité terrestre $s(q)$ et le signe de l'énergie potentielle E_p. Nous cherchons cette fois-ci à minimiser cet indice afin de réduire l'énergie nécessaire à l'humain pour contrer l'effet de la gravité lorsqu'il reçoit l'ob-

jet. Comme pour l'indice de manipulabilité, il nous faut définir un seuil. Ce seuil est un pourcentage de la valeur maximale de l'indice de gravité terrestre. Il permettra à l'algorithme de garder que les points dont les valeurs lui sont inférieures.

Pour exemple, nous définissons ce seuil à 25% de la valeur maximale des indices calculés. Les points récupérés sont représentés sur la Fig. 4.5.

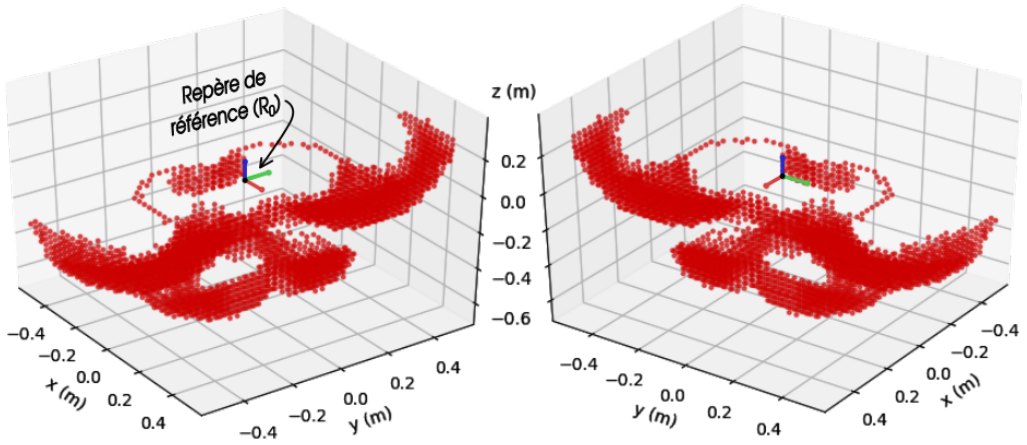


FIGURE 4.5 – Deux vues de l'ensemble des points obtenus en minimisant l'indice de gravité terrestre (moins de 25% de la valeur maximale) par rapport aux membres supérieurs. Le repère de référence R_0 est situé entre les deux épaules de l'homme.

Pour mieux visualiser les zones que l'on peut obtenir, il est proposé en annexe B.2 plusieurs rendus avec différentes valeurs de seuils.

4.3.3 Mise en commun des espaces

Notre objectif est de définir une zone optimale de transfert d'objets pour un individu afin de rendre la tâche plus sûre et confortable.

En résumé, nous générons dans un premier temps un nuage de points qui nous donne l'espace atteignable pour la main de l'humain. Nous calculons par la suite, pour chacun de ces points, l'indice de manipulabilité. En maximisant cet indice, nous définissons la zone la plus manipulable. Nous calculons ensuite l'indice de gravité terrestre. En minimisant ce nouvel indice, nous définissons une zone qui donne les points où l'énergie potentielle est la plus faible, et donc où il est plus facile de maintenir la pose. Si on reprend les zones générées avec les critères de maximisation et de minimisation de ces deux indices, nous pouvons faire ressortir les points communs entre ces deux zones. Ils nous permettent alors de définir un espace de transfert plus acceptable.

En effet, en ne conservant que les points communs précédemment représentés sur la figure 4.4 et la figure 4.5, nous obtenons les points qui définissent la zone souhaitée et qui correspond à nos attentes. Ces points sont représentés sur la figure 4.6.

Le nuage de points de la figure 4.6 représente la zone atteignable en prenant en compte les deux bras de l'utilisateur. Avec nos algorithmes, nous donnons la possibilité de générer la zone de transfert en fonction des préférences de l'humain. En effet, il

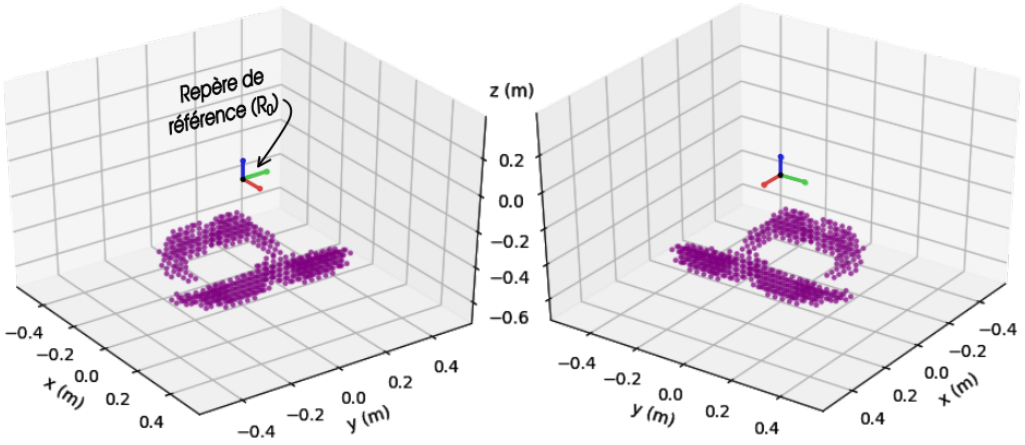


FIGURE 4.6 – Deux vues de l'ensemble des points obtenus en augmentant l'indice de manipulabilité et en minimisant l'indice de gravité terrestre (respectivement supérieur à 90% et inférieur à 25% de la valeur maximale) par rapport aux membres supérieurs. Le repère de référence R_0 est situé entre les deux épaules de l'homme.

est possible de définir la main avec laquelle on souhaite recevoir l'objet. La figure 4.7 permet de visualiser les points générés pour le bras droit en bleu et pour le bras gauche en rouge.

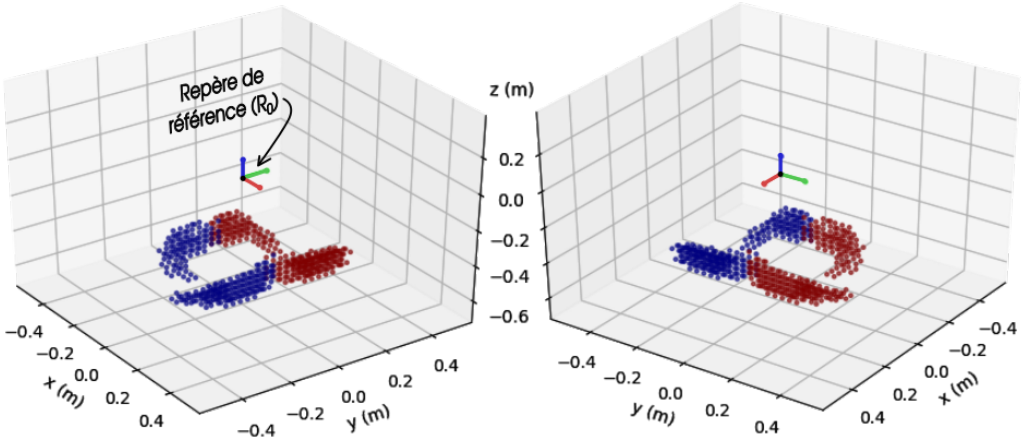


FIGURE 4.7 – Deux vues de l'ensemble des points obtenus en augmentant l'indice de manipulabilité et en minimisant l'indice de gravité terrestre (respectivement supérieur à 90% et inférieur à 25% de la valeur maximale) par rapport au bras gauche en rouge et au bras droit en bleu. Le repère de référence R_0 est situé entre les deux épaules de l'humain.

4.3.4 Enlever les doublons

Après avoir discrétisé le nuage de points et calculé la valeur des indices de manipulabilité et de gravité, on observe un grand nombre de clones. En effet, des points

sensiblement proches peuvent avoir la même valeur d'indice. L'idée est de reprendre tous ces points, un à un, et de regarder s'il n'existe pas des doublons afin de les supprimer en conséquence. Cette suppression a pour but de limiter le nombre total de points. En effet, dans un premier temps, cela nous évite de stocker en mémoire des doublons qui n'ont pas de réelles utilités pour notre application. Puis, dans un second temps, il y aura moins de points à exprimer dans le repère humain et donc moins de calculs à effectuer au moment venu. On évite alors la redondance d'informations qui peut être coûteuse en capacité de stockage et de calcul.

4.3.5 Discussions

La section 2.5 présente les calculs réalisés dans plusieurs travaux antérieurs ([56] et [60]) afin de minimiser les efforts ressentis par l'humain pour un échange d'objet. *L'indice d'angle d'articulation moyen* (voir l'équation 2.2) propose de mesurer la distance angulaire des articulations du bras par rapport à un angle moyen. Effectivement, cet angle est considéré comme confortable car le bras n'est ni en flexion, ni en extension. Minimiser ce paramètre revient à s'éloigner le moins possible de cet angle moyen. Cependant, cette métrique ne mesure pas directement l'impact des forces exercées sur le bras pour une pose donnée. C'est pourquoi, *l'indice de couple minimal de l'articulation* (voir l'équation 2.3) propose quant à lui de mesurer les couples nécessaires aux muscles pour maintenir une pose. Il se concentre ainsi sur l'articulation en elle même, la considérant comme un moteur. Cet indice semble donc plus approprié pour un bras robotisé que pour un individu car il ne prend pas en compte les propriétés musculaires de l'humain.

D'après Katayama *et al.* [65] l'indice qui semble être le plus pertinent est *l'indice de stress musculaire minimum* (voir l'équation 2.1). Il permet de mesurer la contrainte musculaire appliquée sur les muscles. Pour cela, il prend en compte la physiologie musculaire (le volume du muscle, ainsi que la longueur de ses fibres) et la tension qui lui est appliquée. Il semble très difficile à mettre en place car il faudrait connaître plus en détail la physiologie musculaire de la personne.

L'indice que nous proposons est capable de représenter mathématiquement le travail des forces de gravité exercées sur le bras selon sa pose. Minimiser ce paramètre revient à trouver les positions où l'effet de la gravité a le moins d'impact sur la motricité du bras. Si ces forces ont moins d'effet en ces positions, alors il est possible d'induire plus facilement un mouvement. L'effort ressenti est ainsi minimisé lorsqu'il prendra l'objet. C'est donc un indice qui relève des poses de bras considéré plus stable car moins impacté par les forces de pesanteur. Le fait de coupler ce nouveau paramètre à l'indice de manipulabilité rend la position de transfert plus manipulable et plus appréciable encore. La personne peut en effet plus facilement changer de position, rendant ainsi la manipulation et la saisie plus simples et plus sûres.

4.4 Conclusion

Cette partie décrit une méthode de génération d'un espace de transfert ergonomique, garantissant sécurité et confort pour la récupération d'objet. Chacun de ces points représente une position finale de transfert que peut atteindre le bras humain en fonction de son repère de référence. Dans cette position, l'indice de manipulabilité assure à la personne une certaine facilité à bouger ses membres supérieurs pour garantir une saisie correcte de l'objet. Puis, l'indice de gravité terrestre nous permet d'assurer que les efforts mis en jeu sur le bras humain sont minimisés au moment du transfert.

Pour une meilleure efficacité et adaptabilité, il est possible de prendre en compte les limitations articulaires de la personne et du robot. Cela permet de modifier les amplitudes enregistrées pour une personne ne présentant aucun trouble et ainsi avoir une zone de transfert toujours plus confortable. Nous pouvons aussi ajouter certaines préférences pour l'humain. Il peut alors choisir de recevoir l'objet dans une main en particulier ou les deux. Le nuage de points est alors généré en conséquence pour permettre au robot de cibler au mieux les attentes du receveur.

Enfin, les seuils définis sur les indices permettent de les minimiser ou de les maximiser. Cela a pour conséquence d'obtenir différentes formes, plus ou moins élargies selon la valeur de ces seuils choisis. En effet, plus ils seront proches de la limite maximale, moins on aura de points dans ce nuage et plus les points seront proches de la personne. À l'inverse, si l'on prend des seuils plus éloignés de la limite maximale, on aura des zones plus élargies et le robot pourra en conséquence donner l'objet d'un peu plus loin et également plus haut. Chaque personne peut avoir des attentes différentes vis à vis du point de transfert, il sera donc possible de tester directement avec eux différentes zones afin d'établir celle qui leur convient le mieux.

Cet espace de transfert étant maintenant généré, il nous faut mettre en place une stratégie de commande permettant l'interaction entre le robot Pepper et l'humain dans le cadre d'un transfert d'objet.

Chapitre 5

Stratégie de commande pour une interaction physique avec l'humain

Sommaire

5.1 Introduction 67
5.2 Estimation de la posture de l'humain 68
5.2.1 Système de caméra RGB-D 68
5.2.2 Principes de stéréovision 69
5.2.3 Mise en correspondance de point 69
5.2.4 Mise en correspondance de points sans paramètre 71
5.2.5 Estimation de la pose avec OpenPose 80
5.2.6 Estimation des rotations du torse 81
5.3 Estimation du point de transfert 86
5.3.1 Quelques ajustements sur l'espace de transfert généré 86
5.3.2 Zone de transfert ergonomique du robot 86
5.3.3 Estimation du point de transfert 87
5.4 Commande cinématique du robot Pepper 89
5.4.1 Modèle du robot 90
5.4.2 Choix des moteurs pour le transfert 91
5.4.3 Saisie d'objet par le robot Pepper 92
5.4.4 Interaction physique et sécurité 93
5.4.5 Synthèse de nos développements 93
5.5 Localisation et navigation du robot 97
5.5.1 Localisation à l'aide de marqueurs 97
5.5.2 Estimation du plus court chemin 99
5.5.3 Commande du déplacement 102
5.6 Conclusion 104

5.1 Introduction

Le chapitre précédent propose une méthode d'élaboration d'un nuage de points spécifiant une zone de transfert confortable et adaptée au receveur. Cependant, pour rendre l'action possible, il nous faut mettre en place une stratégie de commande permettant au robot d'analyser son environnement et adapter convenablement cette zone à la situation.

La première étape consiste donc à analyser et à estimer la posture de la personne qui se trouve face à lui. Pour le robot Pepper, nous utiliserons sa caméra RGB frontale, ainsi que sa caméra de profondeur qui donneront aux algorithmes les informations nécessaires. L'idée est d'estimer au mieux la position et la rotation du torse à partir de ces informations afin de projeter l'ensemble du nuage de points dans le référentiel humain.

La seconde étape consiste à revoir la zone précédemment générée, et réaliser de petits ajustements en fonction de la situation observée. Par exemple, il est possible de prendre en compte la position assise ou debout du receveur, ou encore d'enlever les points situés dans son dos. Après ces quelques ajustements, il nous faut choisir le point de cette zone finale qui sera défini comme étant le point de transfert idéal.

Pour mettre en place cette application de transfert, il nous faut expliquer nos choix concernant la saisie de l'objet par le robot. Cette saisie est un point essentiel à ne pas négliger, car elle a toute son importance dans cette application. Elle permet de garantir le bon maintien de l'objet par exemple, et de permettre au receveur d'accéder facilement à l'objet pour le récupérer. Nous abordons également dans cette partie tous les développements concernant la géolocalisation et la navigation du robot à l'aide de marqueurs au sol. Cela nous permet de naviguer et d'aller apporter les objets directement au plus près des personnes à qui on doit transmettre l'objet. Pour finir, afin de réaliser la transmission de l'objet, il nous faut établir le modèle du robot et choisir les moteurs à utiliser afin d'atteindre ce point de transfert.

5.2 Estimation de la posture de l'humain

Le nuage de points de l'espace de transfert est déterminé par rapport à un repère fixé au niveau du torse de l'humain, entre ses deux épaules (R_0). L'objectif est de connaître la pose de l'humain afin de reporter convenablement tous ces points. Dans le but de déterminer la position de ce repère par rapport au robot, nous utilisons la caméra frontale RGB et la caméra de profondeur de Pepper.

5.2.1 Système de caméra RGB-D

Notre configuration est la suivante : une caméra RGB et une caméra de profondeur observent une scène commune depuis deux points de vue différents. Cette disposition nous permet d'utiliser les principes de la stéréovision [104]. En effet, la stéréoscopie exploite les informations contenues dans deux images RGB afin d'estimer la carte de profondeur de la scène observée. Cela est possible si l'on connaît parfaitement les propriétés internes de chaque caméra, ainsi que leur disposition l'une par rapport à l'autre. La carte de profondeur peut alors être établie à partir de la disparité observée entre les deux images. Pepper possède une caméra de profondeur située dans ses yeux. C'est un dispositif comprenant un projecteur à lumière structurée d'un côté et une caméra infrarouge de l'autre. L'estimation de la carte de profondeur est alors possible en observant les déformations de ces lumières projetées sur la scène. Pepper possède également une simple caméra RGB au niveau de son front que nous utiliserons dans le cadre de cette estimation.

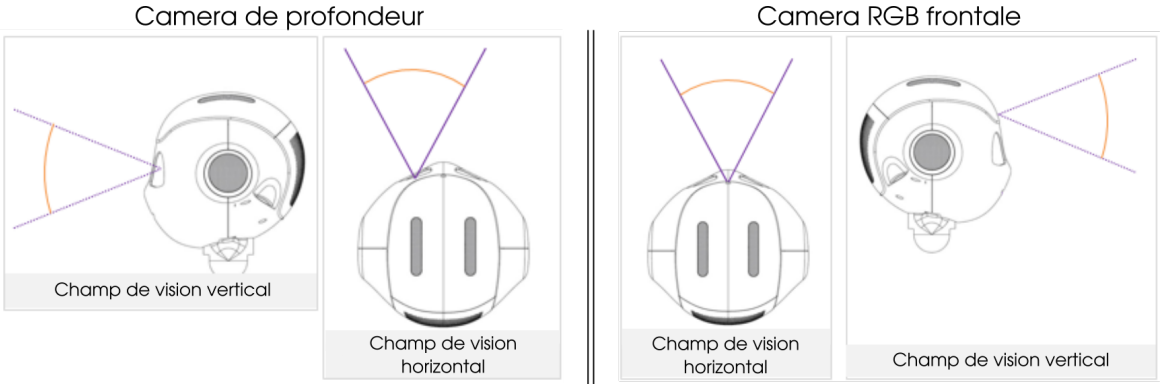


FIGURE 5.1 – Schéma de la tête de Pepper avec la représentation verticale et horizontale du champ de vision de sa caméra de profondeur à gauche des doubles traits, puis à droite, celle de la caméra frontale.

Ces deux caméras observent une même scène mais avec des points de vue légèrement différents. Ce type de dispositif peut s'apparenter à une caméra RGB-D. En effet, ce genre de caméra donne simultanément deux images : une image de couleur et une image de profondeur. Ces informations permettent à tout moment de reconnaître un objet ou un individu, puis d'avoir directement sa position. Dans notre étude, nous cherchons à reconnaître un humain et à estimer sa position afin de lui transmettre

l'objet au bon endroit. Nous proposons donc de coupler ces deux caméras comme un dispositif RGB-D. Afin de faire correspondre les informations contenues dans ces deux images, il nous faut définir quelques notions de stéréovision.

5.2.2 Principes de stéréovision

Il est important dans un premier temps de comprendre quels sont les paramètres qui caractérisent une paire de caméras. On parle ici de deux types de paramètres : les paramètres intrinsèques et les paramètres extrinsèques.

Les paramètres intrinsèques fournissent les caractéristiques internes et propres à chacune des caméras. Ils renseignent sur les distances focales f_x et f_y le long des axes x et y, c_u et c_v les coordonnées en pixel du centre optique dans le plan de l'image, et sur γ qui est l'inclinaison entre les axes. Ces paramètres intrinsèques constituent alors ce qu'on appelle la matrice caméra $\mathbf{K}$:

$$\mathbf{K} = \begin{bmatrix} f_x & \gamma & c_u \\ 0 & f_y & c_v \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.1}$$

Concernant les paramètres extrinsèques, ils désignent la différence de position entre le repère de la caméra et un repère monde ou de référence. On peut alors déduire la matrice de rotation $\mathbf{R}$ et le vecteur de translation $\mathbf{t}$ entre ces deux repères.

Les paramètres intrinsèques et extrinsèques permettent finalement d'obtenir la matrice de projection $\mathbf{P}$ du dispositif RGB-D comme :

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}], \tag{5.2}$$

La matrice de projection $\mathbf{P}$ permet d'estimer la projection de tout point 3D $\mathbf{M} = [x, y, z]^{\mathrm{T}}$ observé par une caméra dans le plan de l'image, en valeur de pixel $\mathbf{m} = [u, v]^{\mathrm{T}}$, à un facteur d'échelle λ :

$$\lambda \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \tag{5.3}$$

Dans la section suivante, nous expliquons comment nous utilisons ces paramètres intrinsèques et extrinsèques afin d'obtenir une image de profondeur colorée à partir d'une paire de caméras RGB et de profondeur.

5.2.3 Mise en correspondance de point

Nous décrivons ici la méthode utilisée dans notre programme pour aligner les deux images. Cet alignement consiste à faire correspondre les informations de couleur de l'image RGB avec les informations de profondeur provenant de la seconde caméra. La différence de position des deux caméras, et la présence de différents objets, structures

ou individus sur différentes profondeurs ne nous permettent pas de superposer directement les deux images. Pour réaliser cette superposition, pixel par pixel, il est nécessaire d'utiliser la matrice de projection (équation 5.2) qui permet de projeter un point 3D dans le plan de l'image RGB.

La documentation technique du robot Pepper fournie par le fabriquant [1, 2] nous donne les valeurs de ces paramètres et permettent de calculer la matrice de projection. Cependant, les calculs sont théoriques et peuvent être calculés plus ou moins éloignés des valeurs réelles suivant le robot. Il est donc nécessaire d'effectuer un étalonnage à l'aide d'un damier dans le but d'avoir des matrices de caméra **K** et de projection **P** plus proches de la réalité. Nous effectuons alors cette calibration à l'aide d'un paquet ROS *pepper/Tutorials/Calibration* [3] utilisant les paquets *camera_calibration* [4] et *ros-naoqi/extrinsic_calibration* [5]. Le paquet *camera_calibration* permet de calibrer facilement les paramètres intrinsèques des deux caméras à l'aide d'un damier de calibration dont les formes sont facilement détectables dans les images RGB et infrarouge. Puis, le paquet *ros-naoqi/extrinsic_calibration* est utilisé pour déterminer les paramètres extrinsèques des deux caméras. Ces paramètres sont définis par rapport au repère de la caméra RGB. Les matrices sont par la suite enregistrées dans un fichier de configuration écrit sous la forme du langage de sérialisation YAML [6].

Après avoir effectué cet étalonnage, nous projetons tous les points 3D fournis par l'image de profondeur dans le plan de l'image RGB. Nous utilisons pour cela la matrice caméra $\mathbf{K_{2D}}$ de la caméra RGB, et la matrice $[\mathbf{R_{3D}}|\mathbf{t_{3D}}]$ pour exprimer tous les points de la carte de profondeur dans le repère de la caméra RGB, comme :

$$\lambda \begin{bmatrix} u_{2D} \\ v_{2D} \\ 1 \end{bmatrix} = \mathbf{K_{2D}} * [\mathbf{R_{3D}}|\mathbf{t_{3D}}] \begin{bmatrix} x_{3D} \\ y_{3D} \\ z_{3D} \\ 1 \end{bmatrix} \tag{5.4}$$

avec $[u_{2D}, v_{2D}]^T$ les coordonnées des pixels dans le plan de l'image RGB et avec $[x_{3D}, y_{3D}, z_{3D}]^T$ les points de l'image de profondeur en coordonnées métriques.

Cette projection nous permet de récupérer toute l'information de couleur contenue dans l'image RGB pour chaque point projeté. On sauvegarde ces informations afin d'attribuer les couleurs aux pixels correspondant de l'image de profondeur. On obtient ainsi une image de profondeur colorée dont la couleur est l'estimation par projection des points de la carte de profondeur dans l'image RGB.

La figure 5.2 représente le résultat de la correspondance des points obtenus. Puis, afin de mieux comprendre l'utilisation de cette mise en correspondance dans cette application, il est proposé de se reporter à la figure 5.17.

1. http://doc.aldebaran.com/2-5/family/pepper_technical/video_2D_pep_v18a.html
2. http://doc.aldebaran.com/2-5/family/pepper_technical/video_3D_pep_v18a.html#video-3d-pepper-v18a
3. http://wiki.ros.org/pepper/Tutorials/Calibration
4. http://wiki.ros.org/camera_calibration
5. https://github.com/ros-naoqi/extrinsic_calibration
6. https://yaml.org/



FIGURE 5.2 – Les premières images, à gauche et au milieu, sont les images originales capturées respectivement par la caméra frontale RGB, puis la caméra de profondeur située dans l'œil de Pepper. La dernière image, à droite, est la représentation de l'image de profondeur en couleur après mise en correspondance.

5.2.4 Mise en correspondance de points sans paramètre

Motivation

Les caméras de profondeur utilisent des méthodes actives ou passives. Dans le premier cas, une mesure directe de la profondeur est obtenue par une lumière structurée ou par des faisceaux laser (mesure TOF : "Time Of Flight"). La seconde méthode repose sur le principe de la stéréovision. Elle est considérée comme passive car elle n'utilise pas de mesure directe. En effet, elle permet d'estimer la profondeur à partir de deux images (RGB) prises simultanément à partir de points de vue différents.

Les technologies de lumière structurée comprennent un projecteur et une caméra infrarouge. Pour obtenir l'image de profondeur, le logiciel embarqué estime la disparité entre le motif projeté et le motif observé par la caméra infrarouge. De manière générale, ce type de caméra est souvent couplé à une autre caméra fournissant une image en couleur. Ce type de couplage a un nom, on parle de caméra RGB-D. Dans ce cas, l'image couleur est directement associée à l'image de profondeur correspondante, par une bibliothèque embarquée conçue et fournie par le fabricant de la caméra. La caméra de ce type la plus connue est la Kinect. Elle a été créée en 2010 par Microsoft pour permettre aux utilisateurs d'interagir sans manette avec la console de jeux vidéo Xbox 360. Parmi les autres caméras RGB-D, nous pouvons citer la RealSense [7].

Les fabricants proposent des bibliothèques pour permettre à leurs clients d'utiliser les caméras et/ou de programmer leurs propres applications. Pour les caméras Asus, les fabricants proposent d'utiliser le SDK OpenNI 2 [105]. Pour la Kinect, Microsoft propose un SDK spécifique [106]. Intel a également développé une bibliothèque pour leur RealSense [107]. Ces bibliothèques utilisent les propriétés physiques, les paramètres intrinsèques et extrinsèques des caméras, pour aligner les images RGB avec les images de profondeur. Les paramètres intrinsèques définissent les propriétés internes de chaque caméra (comme la longueur focale, le centre optique, la distorsion

7. https://www.intelrealsense.com/

de l'image...). Les paramètres extrinsèques définissent la position et l'orientation d'une caméra par rapport à l'autre. Pour obtenir ces paramètres, l'utilisateur peut se référer à la fiche technique du fabricant. Cependant, dans certains cas, cette fiche technique peut être inexistante, difficile d'accès, imprécise par rapport à la réalité du terrain, voire incomplète.

Dans notre cas, Pepper a une caméra de la série Xtion d'Asus utilisant le principe de projection de lumières structurées. La caméra RGB associée est obstruée par la structure de la tête. En effet, cette structure permet seulement de faire passer la lumière projetée par un des deux yeux, puis d'observer cette lumière avec la caméra infrarouge présente dans l'autre œil. La carte de profondeur peut être ainsi estimée et elle nous est directement fournie par le biais de la librairie constructeur de Pepper. En ce qui concerne l'information de couleur, il est possible d'utiliser l'image RGB obtenue par la caméra frontale du robot. En effet, de par sa position, son orientation et son angle de vue, cette dernière observe à peu de chose près la même scène que la caméra infrarouge. On cherche donc à coupler ces deux images afin d'en extraire les informations de couleur et de position de chaque pixel.

Les méthodes classiques de calibration de ce type de caméra permettent de faire correspondre un damier identifiable par les deux caméras. En effet, les carreaux du damier sont visibles par la caméra RGB et la caméra infrarouge. Cependant, avec la librairie de NAOqi, nous pouvons recueillir l'image RGB de la caméra frontale et directement la carte de profondeur. Les carreaux d'un damier classique ne peuvent être identifiés directement dans l'image de profondeur, il faut donc trouver une solution pour identifier des points similaires entre les deux images dans le but de calibrer le dispositif.

État de l'art

Certains chercheurs ont utilisé la calibration conventionnelle pour trouver les paramètres de la caméra et estimer la distorsion de l'image de profondeur. Par exemple, Khoshelham, Elberink [108], et Smisek *et al.* [109] proposent de calibrer le capteur de profondeur de la Kinect. Leur calibration réduit la distorsion de l'objectif et de la profondeur. Pour trouver les paramètres de la caméra de profondeur avec une calibration conventionnelle de la caméra, les auteurs obstruent d'abord le projecteur de la caméra. Puis, à l'aide d'une source lumineuse halogène, ils éclairent le damier. Cela simplifie l'extraction du damier dans l'image de la caméra infrarouge. Il est possible d'utiliser cette méthode avec ROS : Robot Operating System [8] avec le tutoriel [110]. Cette idée fonctionne seulement si on a une caméra de profondeur de ce type et que l'image infrarouge est disponible. Dans notre cas, nous souhaitons rendre possible la calibration directement à partir de la carte de profondeur et non de l'image infrarouge.

Dans ce même objectif, Wu *et al.* [111] proposent une approche différente utilisant un damier semi-transparent pour calibrer la Kinect. Celui-ci est, en effet, facilement visible dans les deux images sans utiliser l'image infrarouge de la Kinect. La calibration est réalisée à partir d'un outil numérique où l'utilisateur montre où se trouve le damier

8. https://www.ros.org/

dans l'image de profondeur. C'est une sorte de quadrillage virtuel que l'utilisateur doit placer manuellement pour chaque image. Il peut ensuite obtenir les paramètres des caméras. En ce qui concerne le travail de Liu *et al.* [112], ils utilisent une sphère dont les propriétés géométriques sont connues afin de calibrer leur dispositif RGB-D. Leur framework est capable de détecter la sphère dans les deux images. Ils estiment les paramètres intrinsèques et extrinsèques simultanément et corrigent également l'erreur de mesure de la profondeur. Cependant, contrairement à nos développements, leur code n'est pas accessible. De plus, nous proposons une méthode qui cherche à obtenir cette mise en correspondance sans utiliser les paramètres intrinsèques et extrinsèques du dispositif. Nous cherchons en effet à utiliser les principes généraux de stéréovision et plus particulièrement des droites épipolaires.

Descriptif de l'approche

Afin de bien comprendre notre méthode, nous proposons en figure 5.3 les six grandes étapes de cet algorithme. La sélection de points dans la première étape est obtenue à partir de la librairie open-source OpenPose [113], qui permet d'avoir une estimation du squelette de la personne visible dans l'image RGB.

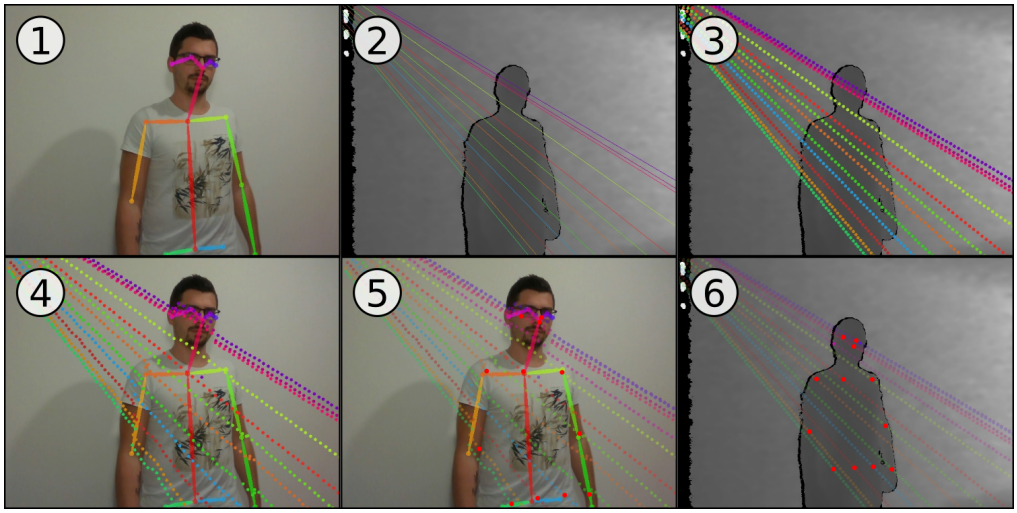


FIGURE 5.3 – Présentation des étapes suivies par nos algorithmes afin d'estimer la position des pixels dans l'image de profondeur correspondant aux pixels de l'image RGB sélectionnée (ici par détection du squelette avec OpenPose). 1 - Sélection de point dans l'image RGB; 2 - Estimation des droites épipolaires dans l'image de profondeur; 3 - Échantillonage de ces droites; 4 - Projection des points échantillonnés dans l'image RGB; 5 - Estimation des points les plus proches des points d'origine; 6 - Mise en évidence de ces points projetés les plus proche dans l'image de profondeur.

Les étapes proposées par notre librairie sont les suivantes :

1. L'utilisateur ou un algorithme extérieur peut être utilisé afin de sélectionner dans un premier temps les pixels dans l'image RGB;

2. La seconde étape a pour objectif d'estimer les droites épipolaires dans la carte de profondeur correspondant aux pixels précédemment sélectionnés. Les équations de ces droites sont obtenues avec la matrice fondamentale **F**;
3. Ces droites donnent une infinité de solutions, la troisième étape consiste donc à les discrétiser afin d'avoir un nombre fini de possibilités;
4. La quatrième étape consiste à projeter ces points échantillonnés dans l'image RGB à l'aide de la matrice de projection **P**;
5. L'avant dernière étape permet de trouver les points les plus proches des points sélectionnés à l'origine dans la première étape;
6. Finalement, la dernière étape de notre méthode met en évidence ces points projetés les plus proches dans l'image de profondeur. Ils sont ensuite colorés selon la couleur des pixels sélectionnés dans l'image RGB.

Calibration du couple RGB-D

Notre objectif est de faire une sélection de pixels dans l'image RGB, manuellement ou automatiquement en estimant la position d'un objet ou d'un individu dans l'image. Puis de trouver les pixels correspondants dans l'image de profondeur à l'aide des droites épipolaires et de la matrice de projection. En effet, dans un premier temps, il nous faut calibrer notre dispositif afin de définir la *Matrice fondamentale* **F** et la *Matrice de projection* **P**. Toutes deux peuvent être définies sans l'utilisation des paramètres nominaux, mais si ceux-ci ne sont pas connus à l'avance, une étape de calibration (également implémentée dans nos développements) est nécessaire. Notre calibration consiste dans un premier temps à sélectionner une multitude de points similaires dans l'image RGB et la carte de profondeur obtenues dans un même instant. On prend une forme simple d'objet, comme une feuille de papier ou une boite, et on la présente devant les deux caméras. Afin d'avoir la meilleure répartition possible de points, il faut tenir cet objet sur différentes distances et sur différents emplacements dans les images prisent. Cela permettra une meilleure estimation de ces matrices. Il est conseillé aussi d'utiliser un arrière-plan non encombré, tel qu'un mur, afin de mieux apercevoir les contours de l'objet présenté devant la caméra de profondeur.

Nous avons implémenté dans notre librairie des outils numériques permettant de faire plusieurs types de sélection. Le premier est le plus simple et consiste à sélectionner point par point les pixels correspondant dans les deux images (mode "point"). Le second permet à partir de deux points de sélectionner une ligne (mode "ligne" permettant d'avoir une interpolation de points entre les deux pixels sélectionnés). Enfin, le dernier, permet de définir un quadrilatère en sélectionnant quatre pixels (définissant les sommets) et une multitude de points sont ainsi enregistrés à l'intérieur de ce quadrilatère (mode "quadrilatère").

La résolution peut être différente entre l'image couleur et l'image de profondeur. Cependant, cette résolution doit être la même lors de la calibration et de la mise en application. Cela est nécessaire pour que **F** et **P** fonctionnent correctement. Pour notre méthode, l'image RGB et l'image de profondeur doivent avoir la même résolution au moment de la mise en correspondance. Nous gardons donc la résolution de la caméra

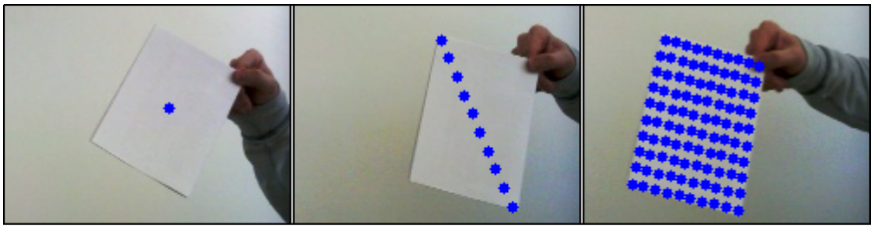


FIGURE 5.4 – Trois modes de sélections pour simplifier la calibration et permettre de prendre plus de point dans une zone de l'image.

de profondeur comme référence et nous redimensionnons l'image RGB en conséquence. Pour éviter de perdre trop d'informations dans l'image RGB, il est conseillé d'utiliser directement la même taille d'image ou une taille aussi proche que possible.

La bibliothèque a besoin d'un fichier *Parameter*. Il contient les paramètres nécessaires à la calibration comme w, h, H_{FOV}, V_{FOV}, c_u et c_v. Les paramètres intrinsèques H_{FOV}, V_{FOV}, c_u et c_v permettent d'estimer les coordonnées x et y si l'on a que la composante z dans la carte de profondeur. Dans notre cas, NAOqi fournit directement la carte de profondeur avec toutes les composantes. Il est nécessaire d'informer l'algorithme concernant la résolution de la carte de profondeur (la hauteur h et la largeur w) afin de pouvoir la lire correctement.

Pour obtenir la *Matrice fondamentale* **F**, il suffit d'appliquer la fonction OpenCV : *cv::findFundamentalMat()*[9] à l'ensemble des points de calibration. Cette matrice **F** permet d'estimer l'équation de la droite épipolaire dans l'image de profondeur selon le pixel sélectionné dans l'image RGB. En effet, pour ce pixel, il existe une infinité de solutions pour la mise en correspondance dans l'image de profondeur. Cette infinité de solutions définit une droite. La solution dépend de la position réelle de l'objet par rapport à la caméra de profondeur. Nous proposons d'échantillonner cette droite et de projeter ces points dans l'image RGB afin de trouver le point le plus adéquat. Pour cela, nous utiliserons la matrice de projection.

La figure 5.5 est proposée, afin de mieux comprendre ce qu'est une droite épipolaire et ce qu'elle représente.

La matrice de projection **P** permet de projeter et de faire correspondre un point 3D ($[x_i, y_i, z_i]$) au pixel correspondant ($[u_i, v_i]$) dans une image (voir l'équation 5.5). Généralement, cette matrice est déterminée par les paramètres intrinsèques et extrinsèques de la caméra (voir section 5.2.2). Mais elle peut aussi être définie expérimentalement en appliquant les moindres carré aux points sélectionnés lors de la calibration, c'est-à-dire les pixels de l'image RGB associés à leurs coordonnées 3D dans le repère de la caméra de profondeur. Cette méthode est déjà bien connue et permet de relier un grand nombre de pixels à leur correspondance dans l'espace tridimensionnel.

9. https://docs.opencv.org/3.4.12/d9/d0c/group__calib3d.html

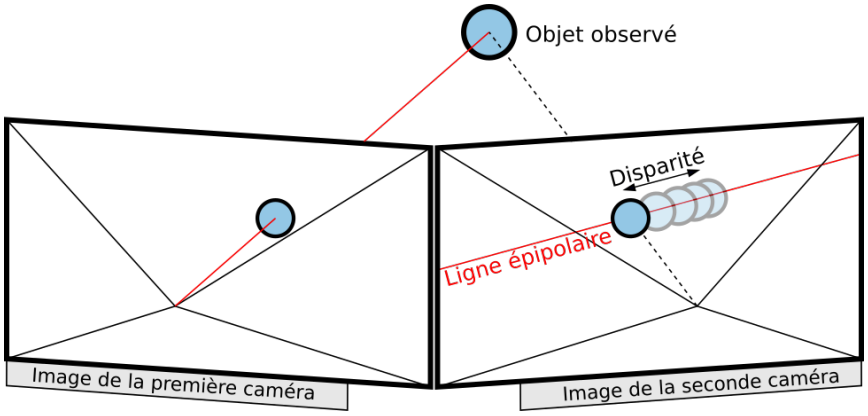


FIGURE 5.5 – Deux caméras observent un même objet. La position de l'objet dans l'image de la première caméra induit une infinité de possibilités de position de cet objet dans l'image de la deuxième caméra. Cette infinité de possibilités est définie suivant une ligne traçable sur la deuxième image, ce qui forme la droite épipolaire.

$$\lambda \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{P} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{P}_{11} & \mathrm{P}_{12} & \mathrm{P}_{13} & \mathrm{P}_{14} \\ \mathrm{P}_{21} & \mathrm{P}_{22} & \mathrm{P}_{23} & \mathrm{P}_{24} \\ \mathrm{P}_{31} & \mathrm{P}_{32} & \mathrm{P}_{33} & \mathrm{P}_{34} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \tag{5.5}$$

En développant (5.5), pour n points nous obtenons 2n équations linéaires indépendantes avec douze inconnues (les éléments de $\mathbf{P}$). Puisque l'équation (5.5) est définie à un facteur d'échelle près, nous pouvons normaliser tous les coefficients par P_{34}. Nous obtenons alors une nouvelle équation matricielle à 11 inconnues :

$$\mathbf{Ax} = \mathbf{b}, \tag{5.6}$$

avec :

$$\mathbf{A} = \begin{bmatrix} 0,0,0,0,x_1,y_1,z_1,1,-v_1x_1,-v_1y_1,-v_1z_1,-v_1 \\ x_1,y_1,z_1,1,0,0,0,0,-u_1x_1,-u_1y_1,-u_1z_1,-u_1 \\ 0,0,0,0,x_2,y_2,z_2,2,-v_2x_2,-v_2y_2,-v_2z_2,-v_2 \\ x_2,y_2,z_2,2,0,0,0,0,-u_2x_2,-u_2y_2,-u_2z_2,-u_1 \\ \dots \\ 0,0,0,0,x_n,y_n,z_n,1,-v_nx_n,-v_ny_n,-v_nz_n,-v_n \\ x_n,y_n,z_n,1,0,0,0,0,-u_nx_n,-u_ny_n,-u_nz_n,-u_n \end{bmatrix}$$

$$\mathbf{x} = [\mathrm{X}_{11},\mathrm{X}_{12},\mathrm{X}_{13},\mathrm{X}_{14},\mathrm{X}_{21}\dots\mathrm{X}_{33}]^\top = \left[\frac{\mathrm{P}_{11}}{\mathrm{P}_{34}},\frac{\mathrm{P}_{12}}{\mathrm{P}_{34}},\frac{\mathrm{P}_{13}}{\mathrm{P}_{34}},\frac{\mathrm{P}_{14}}{\mathrm{P}_{34}},\frac{\mathrm{P}_{21}}{\mathrm{P}_{34}}\dots\frac{\mathrm{P}_{33}}{\mathrm{P}_{34}}\right]^\top$$

$$\mathbf{b} = \begin{bmatrix} u_1 & v_1 & \dots & u_n & v_n \end{bmatrix}^\mathrm{T}$$

La solution à l'équation (5.6) peut être obtenue par pseudo-inverse :

$$\mathbf{x} = \mathbf{A}^\dagger\mathbf{b} \tag{5.7}$$

Pour estimer tous les éléments de $\mathbf{x}$, nous avons besoin d'un minimum de 6 points.

De **x**, on déduit **P** via :

$$\mathbf{P} = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & 1 \end{bmatrix}. \tag{5.8}$$

Comme expliqué précédemment, les matrices **F** et **P** sont estimées et enregistrées dans un fichier de configuration de la librairie. Leur utilisation permettra de définir à tout moment les droites épipolaires et de projeter n'importe quel point 3D de la carte de profondeur dans l'image RGB.

Boîte à outils open-source

Nous avons développé une boîte à outils open-source qui s'intitule : OpenHSML [114], [115] (*"Open-source Hybrid Stereovision Matching Library"*). Cette librairie permet à n'importe quel utilisateur de coupler deux caméras observant une même scène, l'une fournissant une image RGB et l'autre fournissant une carte de profondeur. L'utilisateur sélectionne des pixels dans l'image RGB et les algorithmes fournis dans cette librairie permettent de trouver leurs correspondances dans la carte de profondeur. Cette librairie est exploitée sous la licence GNU Lesser General Public License version 3 (LGPLv3) [116]. Cette licence permet l'intégration de logiciels à code source ouvert ou fermé, à condition que toute modification soit partagée avec la communauté. OpenHSML est écrit en C++ et développé sur le système Unix Ubuntu 18.04. Il est empaqueté à l'aide de PID [10], un système de construction et de déploiement basé sur CMake. Il permet de déployer automatiquement toutes les dépendances dont le projet a besoin pour fonctionner. L'utilisateur peut utiliser cet environnement pour ses développements ou installer le projet de manière autonome, c'est-à-dire sans avoir à créer manuellement un espace de travail PID. Un fichier ReadMe explique les étapes nécessaires pour l'installer avec ses dépendances (OpenCV, Eigen et yaml-cpp) et lancer une démonstration.

Expérimentations et analyse qualitative

L'objectif de cette librairie est simple, sélectionner des pixels dans une image RGB et trouver leurs correspondances dans la carte obtenue par une caméra de profondeur observant la même scène. En entrée des algorithmes, nous donnons l'image RGB, les coordonnées de pixels sélectionnés et la carte de profondeur. La sortie contient alors, pour chaque pixel sélectionné, les coordonnées du pixel correspondant dans l'image de profondeur, ainsi que les coordonnées cartésiennes récupérées en ce point. La figure 5.6 montre la sortie obtenue lorsqu'ils estiment la position 3D (dans le repère de la caméra de profondeur) de pixels sélectionnés dans l'image RGB.

Dans cette partie, nous présentons quelques résultats obtenus avec différentes paires de caméras. Nous avons eu l'opportunité de tester ces algorithmes sur trois paires de caméras différentes. La première est composée de la caméra Intel RealSense D435, un

10. http://pid.lirmm.net

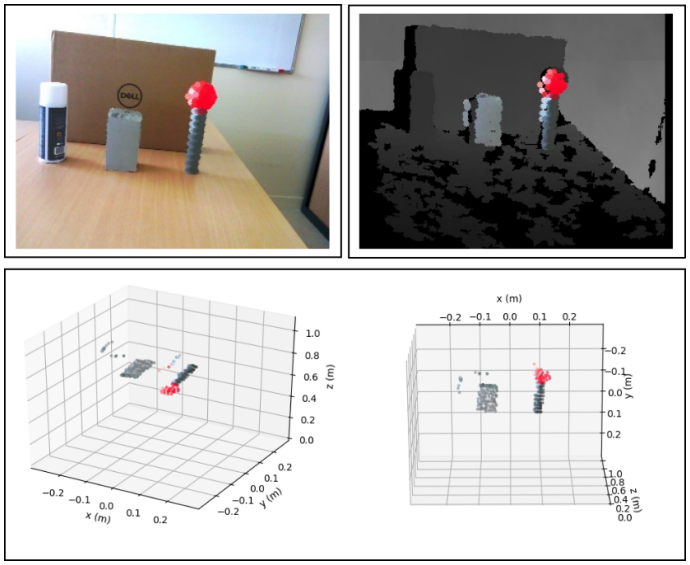


FIGURE 5.6 – Sortie générée par nos algorithmes à partir d'images obtenues par les caméras du robot Pepper. En haut à gauche : image RGB originale avec les points sélectionnés manuellement parmi les objets contenus dans la scène. En haut à droite : l'estimation de leur correspondance est mise en évidence avec les points colorés dans l'image de profondeur. En bas : deux points de vues différents du nuage de points colorés en 3D sont obtenus.

système RGB-D compact avec une caméra de profondeur à lumière structurée. La seconde est composée de la caméra de profondeur Asus Xtion du robot Pepper (lumière structurée également), couplée à sa caméra frontale. Le dernier dispositif testé est le couplage d'une webcam d'ordinateur avec la caméra de profondeur RealSense D435 afin de montrer que le système peut fonctionner sur n'importe quel type de couples de caméras stéréo hybride (caméra RGB et profondeur). Après avoir calibré les trois couples de caméras, nous capturons des images de test où un objet est présent dans la scène. La figure 5.7 présente les résultats obtenus avec les trois paires de caméras.

Malgré de bons résultats, nous avons observé deux types d'erreurs fréquentes. La première est causée par des "trous" dans la carte de profondeur. Cela signifie que l'information de profondeur pour ces pixels est manquante. Elles peuvent être dues à l'erreur d'estimation du système de caméra de profondeur, par exemple si le motif de la lumière structurée n'est pas visible. La deuxième raison concerne le cas où les deux caméras sont trop éloignées l'une de l'autre, un point 3D peut être situé plus loin derrière l'objet observé (par exemple, sur un mur en arrière-plan). Il peut donc être caché dans l'image RGB, ce qui est problématique pour la projection. En effet, ce point peut être projeté sur l'objet en question dans l'image RGB et l'algorithme peut le sélectionner comme point correspondant dans l'image de profondeur.

Nous avons supposé que les images de profondeur et RGB ne sont pas déformées. Dans certains cas, l'utilisateur peut avoir besoin de calibrer ces caméras au préalable pour améliorer leurs rendus. Cependant, dans nos expériences avec la RealSense D435 et avec la caméra de Pepper, nous n'avons pas effectué de calibration standard des pa-

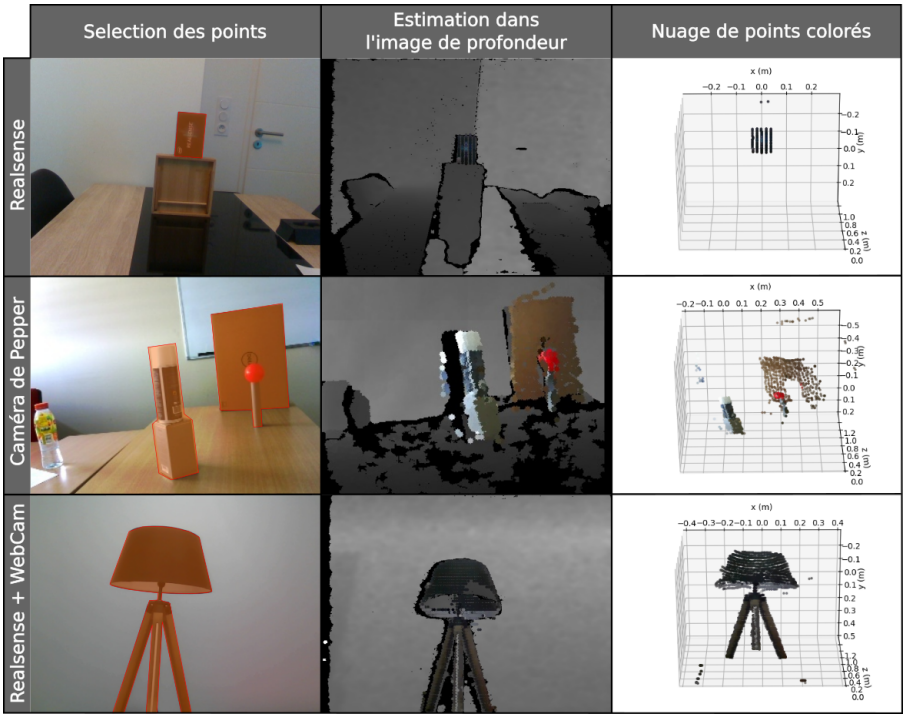


FIGURE 5.7 – Résultats obtenus pour trois caméra RGB-D. On utilise ici la Realsense d'Intel, la caméra de Pepper (caméra frontale et l'Asus Xtion dans ses yeux), puis on teste avec deux autres caméras bien dissociées : la webcam d'un ordinateur et la carte de profondeur obtenue avec la Realsense d'Intel.

ramètres intrinsèques et extrinsèques. Pourtant, nous avons obtenu des résultats cohérents à des distances raisonnables (moins de 2 mètres) et avec des objets centrés dans l'image. En effet, l'estimation de la profondeur est meilleure au centre de l'image et à des profondeurs plus faibles. Cela peut également dépendre de l'étape de calibration et si les points de calibration couvrent plusieurs profondeurs de la zone à analyser. De plus, comme la calibration est effectuée à une certaine distance de la caméra, cela peut entraîner des erreurs d'estimation pour les zones plus éloignées. Dans nos exemples, nous ne dépassons pas deux mètres lors de la calibration par exemple. Nous avons fait ce choix car ces caméras sont connues pour être plus précises en intérieur et à courte distance. Les yeux du robot Pepper sont formés à partir d'un revêtement en verre incurvé et cela induit des erreurs de distordions plus importantes sur les bords de l'image.

Nous proposons une approche simple pour trouver la correspondance ponctuelle entre une image RGB et une image de profondeur. Cette approche utilise les principes de la stéréovision pour trouver les points d'intérêts dans l'image de profondeur sans passer par l'alignement des images. Notre méthode nécessite une calibration rapide et simple, sans avoir recours à un damier, pour déterminer le modèle stéréo. Ce modèle permet de résoudre le problème sans avoir à déterminer les paramètres de la caméra. L'avantage de cette librairie OpenHSML est qu'elle est très facile à utiliser. Elle ne dé-

pend pas d'une caméra particulière. Il est tout à fait possible de coupler une caméra RGB avec une caméra de profondeur de n'importe quel type. Cependant, la boîte à outils que nous avons développée, ne permet pas actuellement de corriger les erreurs dues à la distorsion de l'image de profondeur. Elle ne permet pas non plus d'estimer l'erreur systématique dans la carte de profondeur du dispositif utilisé.

5.2.5 Estimation de la pose avec OpenPose

À partir de cette image de profondeur colorée, nous pouvons plus facilement reconnaître les objets ou les individus dans cette image. Dans ce cas, on s'intéresse à voir si un humain est présent devant le robot et on cherche à connaître sa posture.

Pour cela, on utilise la bibliothèque open-source OpenPose [113] estimant en temps réel la pose des principales articulations du corps humain (épaules, coudes, poignets, hanches, ...) dans une image RGB. Ces algorithmes peuvent aussi estimer la position des mains, des pieds et du visage, et obtenir pour ces parties du corps une multitude de points clés.

Ce qui est intéressant avec cette librairie, c'est qu'elle fournit la position (dans l'image RGB) du point de référence (position du repère R_0). Pour rappel, le nuage de points généré pour le transfert est exprimé dans ce repère situé entre les deux épaules de l'humain. Avec la position des épaules et la position de ce point dans l'image de profondeur colorée, on peut estimer la position et la rotation du repère de référence par rapport au robot. En effet, OpenPose retourne ces points clés du corps humain sous la forme de coordonnées, u et v, de pixels dans l'image de profondeur colorée. On peut alors récupérer les informations de coordonnées métriques à l'emplacement de ces pixels afin d'obtenir la position du repère de référence et la position des épaules par rapport à la caméra du robot.

La projection des points de l'image de profondeur dans l'image RGB n'est pas toujours parfaite. Il peut y avoir des petits décalages ou des erreurs provenant de la distorsion dans l'image de profondeur (déformation de l'image). OpenPose utilise une intelligence artificielle pour estimer la pose du squelette de l'humain dans l'image RGB. On décide alors d'utiliser cet algorithme seulement après projection et donc directement sur l'image de profondeur colorée. Cela nous permet d'assurer une meilleure estimation de la pose du squelette dans l'image de profondeur.

La figure 5.8 montre l'estimation du squelette par OpenPose d'une personne visible par les caméras du robot. L'organigramme de la figure 5.17 donne un aperçu général de l'application sous la forme de schéma fonctionnel. Ce schéma permet de mieux comprendre la place de cette estimation au sein de l'application générale (voir le bloc "Estimation de la posture de l'humain"). Avec la détermination de la position du squelette dans l'image de profondeur colorée, il est possible d'estimer les rotations du torse de l'humain.

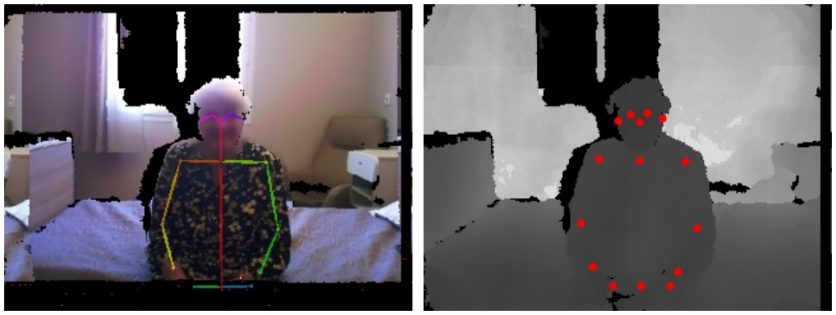


FIGURE 5.8 – L'image de gauche donne le résultat obtenu après analyse d'OpenPose sur l'image de profondeur colorée. L'image de droite met en évidence, par des points rouges, la position des pixels retournés par OpenPose dans l'image de profondeur originale (en niveaux de gris).

5.2.6 Estimation des rotations du torse

Afin de définir la posture de l'humain, nous utilisons les informations de couleur et de profondeur obtenues à l'aide de la caméra frontale et de la caméra de profondeur du robot. L'utilisation des algorithmes d'OpenPose permet d'obtenir la position dans l'image (en coordonnées de pixels) des épaules et du repère de référence de l'humain positionné devant le robot. La carte de profondeur délivre directement une estimation de la position tridimensionnelle (coordonnée x, y et z) avec pour référentiel le repère de la caméra de profondeur. On a donc la position de l'épaule gauche (que l'on notera R_{lsh}), la position de l'épaule droite (R_{rsh}) et la position du repère de référence (R_0) entre les deux épaules. On peut donc à partir de la position de ces points, estimer les rotations du torse de l'humain par rapport à la caméra de profondeur. Trois rotations sont à exprimer et définissent les angles de Cardan : lacet, tangage et roulis. La première à prendre en compte concerne la rotation autour de l'axe z qui est ψ (lacet). La seconde est la rotation autour de l'axe y notée θ (tangage) et la dernière, autour de l'axe x, se note ϕ (roulis). Dans un premier temps, on estime rapidement les rotations de roulis ϕ et de lacet ψ à partir de la position des épaules. La figure 5.9 permet de visualiser la manière dont est déterminée ces deux rotations.

Le roulis, associé à la valeur ϕ, est calculé de la manière suivante :

$$\phi = \arccos\left(\frac{y_{lsh} - y_{rsh}}{\sqrt{(z_{lsh} - z_{rsh})^2 + (y_{lsh} - y_{rsh})^2}}\right) \tag{5.9}$$

Le lacet, associé à la valeur ψ, est calculé de la manière suivante :

$$\psi = \arccos\left(\frac{y_{lsh} - y_{rsh}}{\sqrt{(x_{lsh} - x_{rsh})^2 + (y_{lsh} - y_{rsh})^2}}\right) \tag{5.10}$$

Pour la rotation liée au tangage, c'est un peu plus complexe. En effet, nous ne pouvons pas obtenir directement cette rotation à partir de ces trois points. Si l'on avait la position de la hanche par exemple, on pourrait estimer cette rotation aussi facilement.

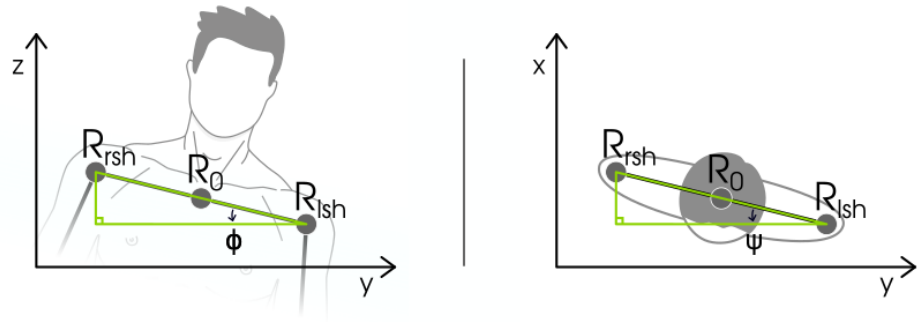


FIGURE 5.9 – Estimation des rotations de roulis ϕ (schéma de gauche) et de lacet ψ (schéma de droite) du torse de l'humain à partir de la position des épaules (épaule droite : R_{rsh}; épaule gauche : R_{lsh}) et du repère de référence (R_0).

Le robot est souvent assez proche de l'humain et ses hanches ne sont pas visibles sans mouvement de la tête. Pour estimer la pose de l'humain nous cherchons à le faire à un seul et même instant. On va donc chercher à estimer un nouveau point sur le torse de l'humain (R_t) dans la carte de profondeur. On projette ce nouveau point en fonction de la position des deux épaules et de la rotation ϕ calculée précédemment. Une représentation de la projection souhaitée est représentée en figure 5.10.

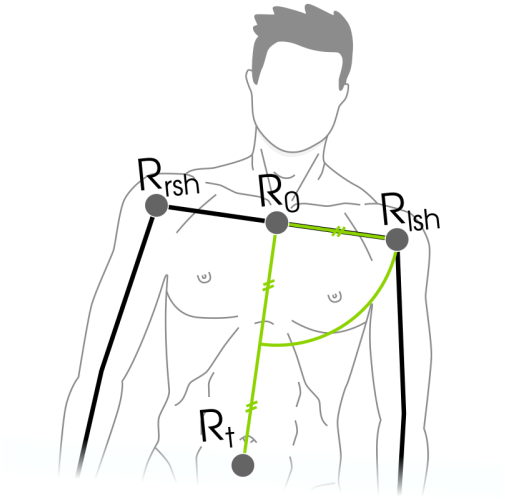


FIGURE 5.10 – Schéma expliquant l'estimation de la position du point R_t par rapport à la position des épaules de l'humain et de la rotation ϕ.

On cherche alors un maximum de points entre ces trois points : R_{rsh}, R_{lsh} et R_t. Le nuage de points ainsi récupéré dans la carte de profondeur entre ces trois points nous permet d'avoir une estimation de l'équation de plan à la surface du torse de l'humain. En effet, avec la méthode des moindres carrés, il est possible de définir l'équation de plan passant par un maximum de ces points. Après avoir obtenu cette équation du plan, on calcule la rotation possible entre ce plan et le plan de coordonnées (yOz).

On obtient alors une estimation du tangage (θ). La figure 5.11 permet de visualiser un résultat obtenu avec cette méthode d'estimation.

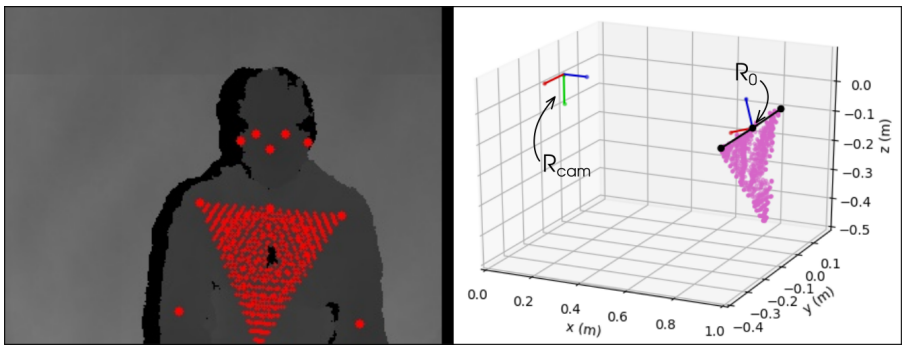


FIGURE 5.11 – Représentation de l'estimation de la position et de la rotation du torse humain. L'image de gauche montre la sélection des points dans l'image de profondeur au niveau du torse. L'image de droite montre le nuage de points récupéré en trois dimensions avec l'estimation de la position et la rotation du repère de référence R_0 en fonction de la position de la caméra de profondeur (représenté par le repère R_{cam}).

L'estimation de la position et de l'orientation du torse humain donne la matrice de transformation homogène qui permet d'exprimer R_0 dans le repère de la caméra de profondeur du robot. La transformation homogène entre ces deux repères permet d'exprimer l'ensemble du nuage de points de la zone de transfert (chapitre 4) dans le repère monde. Le schéma fonctionnel de la figure 5.17 précise le moment dans l'application où ces calculs sont effectués (voir les blocs "Estimation de la posture de l'humain" et "Adaptation de l'espace de transfert et estimation du point").

Lors de l'estimation de la posture de l'humain, il nous faut être capable de gérer certaines situations qui peuvent amener à des erreurs. La première consiste à estimer le nombre de personnes visibles par la caméra frontale du robot. S'il y a personne, il faut que le robot effectue une recherche de l'humain autour de lui. Pour cela, nous avons mis en place un mode "pantin" qui va permettre au robot de regarder légèrement à gauche puis légèrement à droite pour agrandir son champ de vision. Pour estimer s'il y a la présence d'un humain, nous utilisons la détection de squelette dans l'image toujours avec OpenPose. Puis, s'il ne trouve pas une personne, il va tourner avec sa base mobile de 90°sur sa gauche puis refaire la même chose pour vérifier la présence d'un humain, et ainsi de suite jusqu'à ce qu'il trouve quelqu'un autour de lui.

La deuxième possibilité est que nous avons plusieurs personnes dans le champ de vision du robot. Il faut alors faire un choix pour donner l'objet à la "bonne" personne. On considère simplement que l'humain qui doit recevoir cet objet est celui qui est le plus proche du robot. Il faut donc regarder dans l'image de profondeur où se trouve chaque humain par rapport à la caméra de profondeur. Pour cela, on relève les coordonnées du point de référence (R_0) estimé à l'aide d'OpenPose dans l'image de profondeur coloré. Puis on calcule chaque distance euclidienne qui sépare la caméra de ces points. C'est la personne la plus proche du robot qui sera considérée comme le receveur.

Pour faire cette estimation, il nous faut centrer le plus possible le torse de l'humain dans l'image de profondeur. Pour aller plus vite, on va chercher à centrer le point situé entre les deux épaules (R_0) estimé dans l'image RGB de la caméra frontale à l'aide d'OpenPose. Nous faisons alors un contrôle pan-tilt de la tête et de la base mobile afin d'être bien positionné dans la direction de la personne et d'avoir son torse le plus visible possible dans l'image de profondeur. Dans l'image RGB, la différence de position en pixel (noté e pour erreur) entre le point R_0 et le centre de l'image nous permet de générer une légère rotation sur les moteurs de la base mobile et de la tête. L'objectif est de rendre cette différence de pixel la plus proche possible de 0. Le schéma fonctionnel de la figure 5.17 précise le moment où ce contrôle est effectué (voir le bloc "Pan-tilt : base mobile + tête").

$$\begin{cases} e_u &= \frac{u_{max}}{2} - R_0(u) \\ e_v &= \frac{v_{max}}{2} - R_0(v) \end{cases} \tag{5.11}$$

Si l'erreur e_u est inférieure à $\frac{u_{max}}{25}$, on considère qu'il est suffisamment centré et on ne génère pas de rotation sur la base mobile. Sinon on génère une faible vitesse de rotation (de $0.1°.s^{-1}$) proportionnelle à cette erreur :

$$\dot{\theta} = \frac{0.1 * \pi}{180} * e_u \; rad.s^{-1} \tag{5.12}$$

Même chose pour l'erreur e_v où il faut cette fois-ci faire une rotation sur la tête (moteur nommé "HeadPitch") pour annuler cette erreur. Donc si l'erreur e_v est inférieure à $\frac{v_{max}}{25}$, on considère qu'il est suffisamment centré et on ne génère pas de rotation sur la tête. Sinon on génère une faible rotation (de $0.1°$) proportionnelle à cette erreur :

$$\psi = \frac{0.1 * \pi}{180} * e_v \; rad \tag{5.13}$$

Pour éviter de trop grandes vitesses articulaires, on sature la commande à $1.5°.s^{-1}$ pour la base mobile et $1.5°$ pour la tête. Ces valeurs permettent un mouvement fluide et surtout de limiter l'effet de flou dû aux mouvements des caméras entre deux prises.

Le contrôle des moteurs de la tête du robot ne peut se faire qu'en position articulaire et non en vitesse articulaire. En effet, seule la base est contrôlable ainsi par le SDK qui communique avec les modules du robot (voir chapitre 3). Cependant, la boucle de contrôle du pan-tilt est suffisamment courte (50ms) pour que le mouvement ne soit pas fini avant l'envoi de la prochaine commande (cela évite les saccades). Cette valeur a été choisie ainsi car notre configuration matérielle donne à OpenPose la possibilité de retourner son estimation de la pose du squelette toutes les 50ms (20fps). Il est donc inutile d'envoyer de nouvelles commandes plus tôt. L'API contrôle les moteurs de manière réactive [11] avec la fonction "ALMotionProxy::setAngles" [12]. Cette fonction est non bloquante et il est possible à chaque temps de cycle de ALMotion (toutes les 20ms) de changer la valeur de position que l'on souhaite atteindre. L'algorithme qui contrôle ce pan-tilt va donc actualiser la valeur de position articulaire de la tête selon l'erreur de

11. http://doc.aldebaran.com/2-5/naoqi/motion/control-joint.html
12. http://doc.aldebaran.com/2-5/naoqi/motion/control-joint-api.html

position observée dans l'image entre la position du pixel où se trouve R_0 et le centre de l'image. Dans cette fonction "ALMotionProxy::setAngles", il faut définir une proportion de la vitesse maximale que peut atteindre le contrôleur du robot. Pour ne pas avoir des vitesses trop importantes, nous contrôlons ces moteurs à 10% de leurs vitesses articulaires maximales. Les résultats obtenus avec ce contrôle sont satisfaisants pour l'application.

Notre méthode d'estimation est robuste seulement si la personne est "face" à la caméra. Dans le cas contraire, la précision de la méthode peut fortement diminuer. Pour ce type de tâche, certains utilisent le deep-learning permettant d'estimer la position d'un humain à partir d'image RGB [117]. Notre méthode est suffisante pour les applications visées, mais il serait intéressant de coupler cette estimation à une estimation utilisant le deep-learning pour améliorer la robustesse en fonction de l'angle de vue de la caméra.

5.3 Estimation du point de transfert

Le nuage de points généré dans le chapitre 4 définit une zone qui est confortable et sûre pour un transfert d'objet à un humain. Cependant, lorsque l'on effectue un transfert, nous devons procéder à quelques ajustements en fonction de la situation pour définir un point de transfert cohérent. L'ensemble des développements détaillés dans cette section seront représentés dans le schéma fonctionnel général de la figure 5.17 du bloc "Adaptation de l'espace de transfert et estimation du point".

5.3.1 Quelques ajustements sur l'espace de transfert généré

Le premier ajustement à réaliser concerne les points toujours présents dans le dos de la personne. En effet, nos algorithmes sortent un nuage de points où on maximise la manipulabilité et minimise l'effort nécessaire pour changer de pose. Avec les amplitudes articulaires maximales, des points apparaissent derrière le plan frontal de la personne. Ces points de transferts ne sont évidemment pas naturels pour cette tâche. Nous décidons de les supprimer du nuage de points avant de les exprimer par rapport à la position réelle de l'individu.

Le second ajustement dépend de la position de la personne. En effet, si la personne se trouve être assise ou debout, la position de ses genoux peut être dans la zone définie par le nuage de points. Le robot peut estimer le point à cet endroit et "percuter" le genoux. Il faut alors enlever ces points pour éviter tout contact non-volontaire avec le corps du receveur. On regarde pour cela les tables anthropométriques du corps humain pour trouver la hauteur que doit avoir le repère de référence par rapport au sol (en position debout). Cette information de hauteur nous permettra facilement de savoir la posture de la personne. Si l'individu est assis, on estime la hauteur de ses hanches par rapport au repère de référence, puis on supprime tous les points situés en dessous de cette hauteur.

5.3.2 Zone de transfert ergonomique du robot

Pour que le robot puisse donner l'objet à la personne, il faut assurer l'accessibilité du point de transfert ciblé. Or, le robot Pepper est capable de se déplacer sur le sol dans n'importe quelle direction. Par conséquent, ses seules limites concernent la hauteur que peut atteindre son bras. Pour des raisons de sécurité, nous avons bloqué les moteurs de la jambe de Pepper qui lui permettrait d'atteindre des zones plus basses. En effet, nous souhaitons éviter les mouvements involontaires sur ces moteurs qui pourraient provoquer sa chute.

On propose alors de définir le modèle géométrique des bras du robot à partir des données constructeur. Puis comme pour l'humain, dans le chapitre 4, nous générons et discrétisons le nuage de points résultant. Le nuage de points est généré pour le bras qui donnera l'objet. En général, nous prenons le bras droit du robot pour donner l'objet, mais il est tout à fait possible de faire la même chose avec le bras gauche.

Nous souhaitons également ajuster cette zone de portée du robot en maximisant la manipulabilité et en minimisant l'indice de gravité. Il faut alors, comme pour l'humain

dans le chapitre 4, calculer les indices associés. Afin de disposer d'une zone suffisamment large, nous choisissons de conserver les points dont l'indice de manipulabilité est supérieur à 25% de sa valeur maximale et les points dont l'indice de gravité est inférieur à 75% de sa valeur maximale. Le nuage de points résultant est représenté en violet sur la figure 5.12.

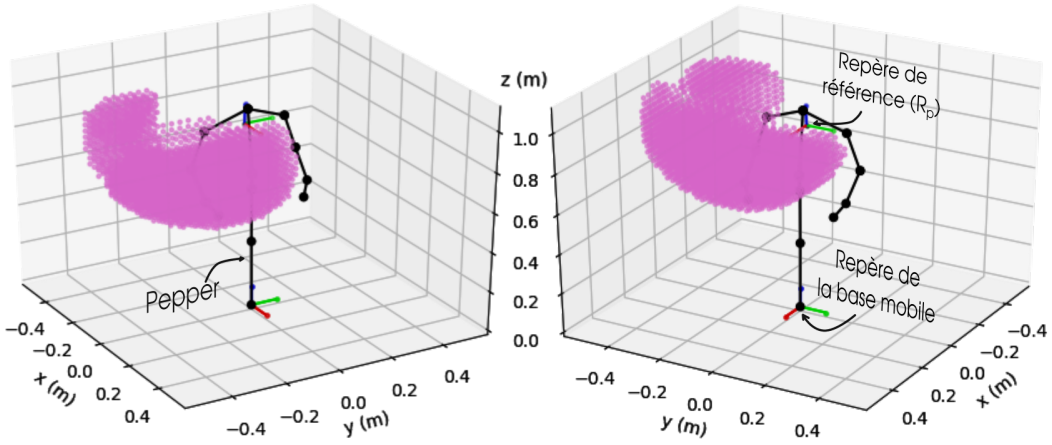


FIGURE 5.12 – Ensemble de points générés pour le bras droit de Pepper en minimisant l'indice de gravité (à 75% de sa valeur maximale) et en maximisant l'indice de manipulabilité (à 25% de sa valeur maximale). Pepper est représenté ici en filaire pour une meilleure visualisation (points et segments noirs). Dans les deux images, nous montrons le repère de la base mobile et le repère de référence du robot (Rp).

Ce nouvel espace est plus approprié pour le robot et définit donc deux nouvelles limites à ne pas dépasser. En effet, nous conservons les points de la zone dite confortable de l'humain entre ces deux limites de hauteur.

5.3.3 Estimation du point de transfert

Notre algorithme nous donne un nuage de points qui définit une zone de transfert pour l'humain. Le robot doit atteindre cette zone pour transférer l'objet dans les meilleures conditions. Nous devons donc faire une sélection dans cet espace pour savoir quel point devra être atteint pour le transfert de l'objet. Notre première idée était de prendre le point le plus proche entre ce nuage de points et la position de la main du robot afin de limiter les déplacements de l'un vers l'autre. Lors de nos différents tests, nous avons observé que cette proposition fonctionnait. Cependant, le point à atteindre était généralement trop bas. Pour régler ce problème et prendre en compte l'espace de transfert de Pepper généré précédemment, nous avons choisi de prendre le barycentre de cette zone à la place de la position de la main.

Le calcul des coordonnées du barycentre $\mathbf{b}$ d'un nuage de points est possible en faisant la somme pondérée des coordonnées de chacun de ces points, divisée par la somme des pondérations $\mathbf{m}$. Le pondération est attribuée au point d'indice i ($\mathbf{p_i}$ =

$[x, y, z]$) par un poids relatif noté m_i. L'équation 5.14 donne l'expression du barycentre pour n points et une pondération équivalente sur tous ces points ($m_i = 1$) :

$$\mathbf{b} = \frac{\sum_{i=1}^{n} (\mathbf{p_i} * m_i)}{\sum_{i=1}^{n} m_i} \tag{5.14}$$

La zone de transfert du robot étant fixe, nous proposons de visualiser ce barycentre dans la figure 5.13.

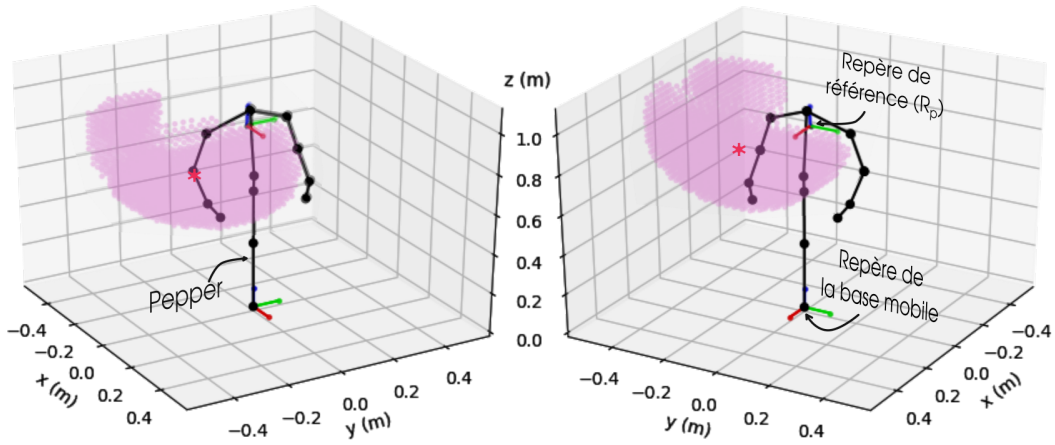


FIGURE 5.13 – Barycentre (étoile rouge) de la zone de transfert générée (violet transparent) pour le robot Pepper. La zone est générée pour le bras droit de Pepper en minimisant l'indice de gravité (à 75% de sa valeur maximale) et en maximisant l'indice de manipulabilité (à 25% de sa valeur maximale).

Nous recherchons alors, dans le nuage de points de l'humain, le point le plus proche de ce barycentre. Pour cela, on calcule simplement la distance euclidienne d_i qui sépare chacun des points (i points) de la zone de transfert de l'humain $\mathbf{p_i} = [x_{pi}, y_{pi}, z_{pi}]$ à ce barycentre $\mathbf{b} = [x_b, y_b, z_b]$:

$$d_i = \sqrt{(x_b - x_{pi})^2 + (y_b - y_{pi})^2 + (z_b - z_{pi})^2} \tag{5.15}$$

Puis on garde le point $\mathbf{p}$ d'indice i dont la distance euclidienne d_i est la plus courte. Le point de transfert à atteindre est ainsi déterminé.

5.4 Commande cinématique du robot Pepper

Pour atteindre le point de transfert sélectionné, nous devons contrôler le bras du robot tout en gérant la navigation de la base. Pour cela, nous utilisons le contrôleur RKCL ("Robot Kinematic Control Library") développé par Tarbouriech *et al.* [78]. Cette bibliothèque C++ permet de mettre en place rapidement un contrôleur du robot Pepper et offre les outils nécessaires pour réaliser des tâches dans l'espace cartésien. Pour ce faire, le modèle du robot est d'abord fourni à la bibliothèque pour définir les propriétés cinématiques que doit respecter le contrôleur. En particulier, cela assure la mise en correspondance de la configuration des articulations avec la pose de la main du robot. Ce modèle fournit également d'autres paramètres tels que les limites des articulations et les vitesses articulaires maximales. Dans RKCL, d'autres paramètres spécifiques à l'application peuvent être fournis dans un fichier de configuration séparé écrit en YAML. Parmi eux, certains sont nécessaires pour définir comment le contrôleur cartésien doit fonctionner : quelles articulations sont activées, et quel membre terminal du robot elles visent à contrôler (dans notre cas, ce sera la main droite du robot). D'autres paramètres affectent la loi de contrôle elle-même, comme les paramètres de gain, ou le pas de temps de la boucle de contrôle. Nous avons fixé ce dernier à $50ms$ pour donner au contrôleur suffisamment de temps pour traiter les données de vision entre les étapes tout en étant suffisamment réactif.

En ce qui concerne la loi de commande pour le contrôle du robot dans l'espace de la tâche, RKCL propose une commande en position du type :

$$\dot{x} = \dot{x}_{des} + \mathrm{K} * (x - x_{des}) \tag{5.16}$$

avec $\dot{x}$ la commande en vitesse cartésienne, $\dot{x}_{des}$ la vitesse cartésienne désirée issue du générateur de trajectoire et $(x - x_{des})$ l'erreur sur la pose cartésienne entre ce que fournit le générateur de trajectoire x_{des} et la pose actuelle x [118].

La résolution au niveau articulaire se fait via un processus de cinématique inverse (IK), basé sur l'architecture de programmation quadratique (QP) qui permet de considérer un ensemble de contraintes. Ce processus génère donc les vitesses articulaires $\dot{q}_{cmd}$ permettant de suivre la commande en vitesse cartésienne $\dot{x}$:

$$\begin{aligned} \dot{q}_{cmd} \in \underset{\dot{q}}{min} &= |\mathrm{J} * \dot{q} - \dot{x}|_2 \\ s.t.\ &\mathrm{A} * \dot{q} <= b \end{aligned} \tag{5.17}$$

avec $\dot{q}$ la vitesse articulaire actuelle des moteurs et J la jacobienne de la chaîne cinématique à contrôler. A et b permettent la prise en compte des contraintes articulaires.

Pour ce qui est des contraintes, elles concernent la position, la vitesse, et l'accélération articulaire des moteurs, mais aussi du membre terminal dans l'espace cartésien. Les contraintes cartésiennes sont prises en compte directement par le générateur de trajectoire et les contraintes articulaires par le QP.

Grâce à cette bibliothèque, nous pouvons rapidement déployer le robot dans le cadre d'une tâche de transfert avec la garantie que la commande conjointe ne dépasse pas les limites autorisées en termes de position et de vitesse.

5.4.1 Modèle du robot

Afin de contrôler convenablement le robot Pepper avec la bibliothèque RKCL, il faut fournir dans un premier temps le modèle géométrique et les propriétés cinématiques des moteurs. Ces informations peuvent être récupérées dans le fichier URDF de Pepper V18A [13] fourni par le constructeur. Pour la bibliothèque, on choisit de fournir ce modèle en format YAML. Un fichier YAML est donc écrit pour définir la position et la rotation de chaque moteur du robot. Cette description est définie selon trois chaînes cinématiques. Elles sont représentées en figure 5.14.

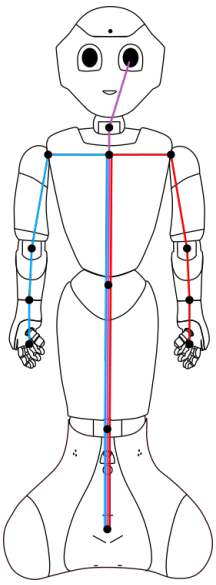


FIGURE 5.14 – Représentation des trois chaînes cinématiques du robot Pepper que nous utilisons dans nos développements. La chaîne en rouge part de la base mobile jusqu'à la main gauche du robot, la chaîne en bleu jusqu'à la main droite et enfin la chaîne violette jusqu'à la caméra de profondeur. Elles permettant d'exprimer l'ensemble des positions et rotations des repères dans le repère d'origine situé au niveau de la base mobile.

Cela permet de bien reprendre le fichier URDF fourni, d'enlever les informations dont on ne se sert pas (comme la masse, l'inertie, ...) et de compléter ce modèle avec d'autres repères. En effet, pour contrôler la base mobile du robot, il y a trois roues omnidirectionnelles. Le fichier URDF donne la position de chacune des roues par rapport à la base mobile. Avec NAOqi, nous pouvons contrôler directement la translation de cette base selon l'axe X et l'axe Y, puis contrôler sa rotation θ. RKCL permet de contrôler directement ces deux translations et cette rotation selon les besoins. On définit alors des moteurs virtuels : "joint1_base" pour la translation suivant l'axe X, "joint2_base" pour la translation suivant l'axe Y, "joint3_base" pour la rotation θ autour de l'axe Z.

RKCL retourne la matrice de rotation et de translation de n'importe quel repère indiqué dans le modèle du robot. On définit alors le repère "ReferenceBetweenShoulders" qui permet de connaître la matrice de transformation homogène du repère de

13. doc.aldebaran.com/2-5/_downloads/JULIETTEY20V171.urdf

référence R_p du robot. On récupère également à tout moment la position de la caméra de profondeur de Pepper ("DepthCamera"). Cela nous sera utile pour reporter tout ce que peut voir et estimer le robot dans son repère d'origine ("joint1_base").

Pour plus de détails concernant la description géométrique du robot Pepper, nous proposons en annexe A.1.1 l'ensemble des figures fournies par le constructeur qui définissent les distances entre chaque moteur et leurs amplitudes articulaires. Puis en annexe A.1.2, il y a une représentation de Pepper et la disposition des différents repères définissants le modèle géométrique du robot. Enfin, le tableau de l'annexe A.1.3 précise quelques données du modèle et décrit la position des repères les uns des autres.

Cependant, pour tendre l'objet à une personne, tous les moteurs du robot ne sont pas utiles. C'est pourquoi, il nous faut faire un choix dans l'ensemble de ces moteurs pour avoir le mouvement adéquat.

5.4.2 Choix des moteurs pour le transfert

Le robot doit atteindre un point dans l'espace tridimensionnel avec la main comme membre terminal. Nous avons fait le choix de ne pas imposer de rotation afin que le robot puisse atteindre plus rapidement l'objectif et être sûr de trouver une solution pour atteindre le point de transfert. En effet, puisque nous bloquons les moteurs de la jambe, le robot ne peut pas descendre plus bas et certaines positions ne pourraient pas être atteintes si on imposait une rotation de la main. Dans notre cas, le robot donne à la personne un paquet de mouchoirs ou du gel hydroalcoolique, et leurs rotations importe peu. Si nous souhaitons donner un verre d'eau ce serait un problème et nous devrions prendre en compte cette rotation pour ne pas renverser l'eau durant le transfert.

Afin d'avoir un geste simple de Pepper pour donner l'objet, nous avons supprimé les rotations de roulis (les moteurs ayant pour dénomination "Yaw") sur le bras au niveau du poignet et du coude. Les moteurs gardés permettent alors de lever simplement le bras pour atteindre le point de transfert désiré. Comme expliqué précédemment, nous contrôlons aussi la base mobile afin que le robot avance, recule, aille sur les côtés ou tourne. Le moteur de la main ("RHand") permet juste de maintenir l'objet pendant le transfert et de le lâcher lorsque la personne signale qu'elle l'ait saisi. Lors de notre première expérimentation (voir chapitre 6), nous avons eu des positions de bras non-naturelles. En effet, la rotation de roulis du poignet et celle du coude pouvait être modifiées par RKCL et pouvait donc mettre le bras dans des positions peu naturelles pour transmettre un objet. Malgré le fait que le point de transfert était atteint, cela pouvait surprendre les receveurs et ils ne savaient pas toujours s'ils pouvaient prendre ou non l'objet. Nous avons donc choisi de ne plus contrôler ces deux articulations "RElbowYaw" et "RWristYaw" afin d'avoir un mouvement plus simple et un transfert plus naturel. Les noms des moteurs sélectionnés pour cette commande sont représentés en figure 5.15.

Pour faire cette sélection, RKCL lit un fichier de configuration écrit en format YAML lui permettant de configurer la façon dont il doit contrôler le robot pour atteindre l'objectif. On définit alors ici un groupe sous la forme d'un vecteur avec les noms de tous

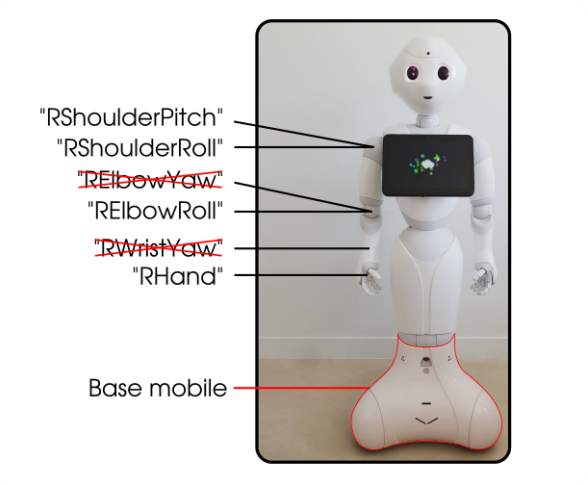


FIGURE 5.15 – Noms des moteurs sélectionnés du bras droit afin de transférer un objet avec Pepper. Les deux noms de moteurs barrés en rouge ne sont pas sélectionnés au niveau du bras contrôlé (ici le droit). La base mobile est également mise en évidence dans cette image car elle est aussi utilisée pour atteindre le point de transfert.

les moteurs du robot. Une matrice de sélection permet ensuite de contrôler que certains de ces moteurs. Dans ce fichier, nous devons aussi renseigner sur les vitesses maximales de commande à ne pas dépasser avec ce contrôleur. En testant de nombreuses limitations, nous avons estimé que $0.2 rad.s^{-1}$ permettait d'avoir un mouvement fluide, suffisamment rapide pour être considéré comme naturel, mais aussi suffisamment lent pour ne pas surprendre le receveur et lui permettre d'anticiper plus facilement les mouvements du robot.

Avant le transfert, le robot est dans la direction de la personne, cela lui permet d'aller à l'essentiel lors du transfert en projetant le bras vers l'avant et en avançant devant lui (suivant l'axe x). Cependant, pour obtenir un déplacement plus naturel du robot, nous contrôlons également la rotation de sa base autour de l'axe z et son déplacement sur les côtés (suivant l'axe y). Le mouvement est ainsi fluide, pas trop rapide et surtout naturel pour le receveur. Les volontaires sont satisfaits de ces mouvements lors des expérimentations réalisées (chapitre 6).

5.4.3 Saisie d'objet par le robot Pepper

Le robot Pepper a des mains plutôt basiques qui n'ont pas été conçues pour la saisie d'objets. En effet, leur fonction primaire consiste à compléter l'interaction vocale que peut avoir le robot en ajoutant une interaction non-verbale. Comme pour un humain, cela lui permet d'ouvrir, fermer les mains et bouger les bras tout en parlant. Pour ses mains, le robot dispose d'un seul moteur qui actionne tous les doigts en même temps, ce qui ne lui permet pas d'avoir des gestes de préhension précis. La force dans les doigts du robot pour la saisie n'est pas suffisante et les moteurs des bras sont également limitatifs. En effet, ils ne permettent pas de soulever des objets pesant plus de

200g (environ). C'est pourquoi, nous avons choisi des objets simples pour nos transferts, comme un paquet de mouchoirs ou une petite bouteille de gel hydroalcoolique. Cela permet de s'assurer que l'objet est bien tenu durant le transfert, car le plastique adhère bien mieux sur la partie antidérapante de ses mains.

Nous avons donc choisi de simplifier la saisie initiale de l'objet par le robot en lui donnant directement et correctement dans sa main. Pour cela, le robot se place dans une configuration spécifique (extension du bras et ouverture de la main). Ensuite, un opérateur place correctement l'objet désiré dans la main de Pepper, en prenant soin de laisser une partie facilement accessible pour que l'humain puisse le saisir. Après deux secondes d'attente, le robot ferme sa main pour maintenir l'objet le plus fermement possible et rapproche son bras le long de son corps. Cette saisie d'objet est représentée dans le schéma fonctionnel de la figure 5.17 (voir bloc "Récupérer l'objet"). Cette phase de préhension est importante et l'amélioration de cette partie sera considérée comme un travail futur.

5.4.4 Interaction physique et sécurité

Les expérimentations réalisées et présentées dans le chapitre 6 ont révélé des problèmes d'interactions et de sécurité dans notre application d'origine. Il est donc important d'ajouter des fonctionnalités permettant à l'humain de déclencher une sécurité et de stopper le mouvement, ou tout simplement pour signaler la prise de l'objet.

La sécurité intégrée à Pepper est trop limitante et le bloque lorsqu'un objet ou un humain se trouve trop proche de lui (30cm). Pour garder un mouvement fluide et ne pas déclencher cette sécurité, nous avons mis en place d'autres moyens pour estimer si le robot est en contact avec l'utilisateur ou si l'humain souhaite arrêter son avancée. Dans les cas extrêmes, il est possible de couper l'alimentation du robot en utilisant un bouton d'arrêt d'urgence fixé au niveau de sa nuque. Pour éviter de stopper le robot brutalement de cette façon, l'algorithme regarde à tout moment si un bumper est activé ou si la tablette est touchée. Si cela est fait, le robot stoppe son mouvement et peut passer à l'étape suivante : donner l'objet. Cette sécurité est schématisée par la figure 5.17 dans l'avant-dernier bloc : "Atteindre le point de transfert".

Puisque qu'il n'y a pas de capteurs de pression aux mains du robot, il n'est pas possible de détecter l'interaction physique lorsque l'humain prend l'objet. Pour signaler le maintien de l'objet et lui demander de le lâcher, la personne peut toucher la tablette ou simplement dire "Merci" par exemple. En effet, nous utilisons ici la fonctionnalité offerte par NAOqi concernant sa reconnaissance vocale. Cette interaction est également schématisée par la figure 5.17 dans le dernier bloc : "Fin du transfert". Le robot peut ouvrir la main pour lâcher l'objet et pour finir : reculer de 30cm.

5.4.5 Synthèse de nos développements

L'objectif de cette partie est d'expliquer le plus précisément possible les étapes qui permettent au robot Pepper d'échanger un objet à l'humain. L'organigramme présenté dans la figure 5.16 permet de visualiser les grandes étapes de notre application. Les

explications plus détaillées concernant cet organigramme sont réalisées par la suite étape par étape.

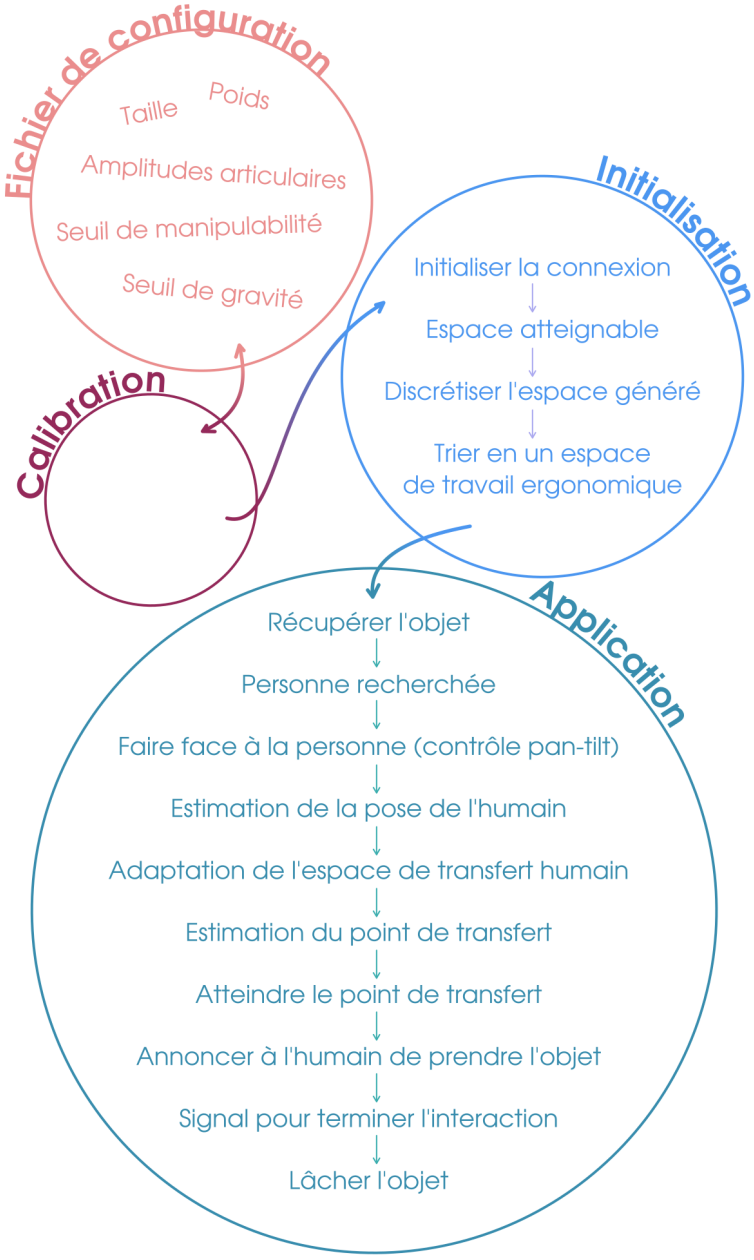


FIGURE 5.16 – Organigramme général de notre application pour un transfert confortable vers un humain.

La première étape consiste à demander quelques informations à la personne qui va recevoir l'objet. Nous lui demandons sa taille, son poids et si elle présente des problèmes articulaires au niveau de ses épaules et de ses coudes. Elle peut renseigner également sa préférence concernant la main qui recevra l'objet. Enfin, il faut indiquer les

seuils des indices de manipulabilité et de gravité pour estimer une zone de transfert confortable. Ces informations sont alors enregistrées dans la partie *"calibration"*.

La partie suivante est *"l'initialisation"*. Elle consiste à initialiser la connexion avec le robot et à générer la zone de transfert (voir le chapitre 4) en fonction des paramètres donnés précédemment. Le nuage de points obtenu pour la zone d'atteinte est ensuite discrétisé (Sect. 4.2.3), puis nous calculons pour chaque position l'indice de manipulabilité (Sect. 4.3.1), l'indice de gravité et l'énergie potentielle (Sect. 4.3.2) associés à la pose du bras. Cet indice de manipulabilité est maximisé en supprimant les points dont la valeur est inférieure au seuil proposé. Pour l'indice de gravité, il est minimisé en supprimant les points dont la valeur est supérieure au seuil. Ici, nous générons également la zone de transfert du bras du robot qui tient l'objet (Sect. 5.3.2). Ce sera utile pour vérifier l'accessibilité de la zone de l'humain et pour estimer le point de transfert le plus adéquat.

L'application est ensuite lancée. Le robot lève sa main, de manière automatique, pour récupérer l'objet (Sect. 5.4.3). Il recherche ensuite la présence d'un humain autour de lui à l'aide de sa caméra frontale. Dès qu'il l'a trouvé, il effectue un mouvement sur un des moteurs de sa tête (mouvement vertical) et sur sa base mobile (mouvement horizontal), afin d'être bien dans l'axe de la personne. C'est un contrôle pan-tilt permettant de positionner le corps du robot dans la direction de la personne avant le transfert. Maintenant, le robot peut estimer la position de l'humain à partir de sa caméra frontale et de sa caméra de profondeur (Sect. 5.2). Il exprime l'ensemble du nuage de points de la zone de transfert générée, dans la pose de l'humain. Puis un dernier ajustement du nuage de points est effectué avant l'estimation du point de transfert (Sect. 5.3.1). En effet, les points non accessibles par le robot sont supprimés, ainsi que les points situés dans le dos de l'humain. Pour estimer le point de transfert, nous calculons le barycentre de la zone de transfert du robot (Sect. 5.3.3). Ensuite, nous récupérons le point de la zone de transfert de l'humain qui est le plus proche de ce barycentre. Ce point sera notre point cible pour le transfert.

Pour finir, le robot peut effectuer la tâche. Les algorithmes contrôlent les différents moteurs nécessaires au robot pour atteindre le point de transfert (Sect. 5.4). Le contrôleur utilisé permet au robot de se déplacer avec sa base mobile et d'étendre son bras en même temps. Lorsque la position est atteinte, Pepper annonce verbalement à la personne qu'elle peut récupérer l'objet. Elle peut répondre d'un simple "Merci", ou toucher la tablette pour signaler qu'elle a bien reçu l'objet. Le robot relâche alors la main, puis recule d'une trentaine de centimètres.

Dans le cadre de nos expérimentations, nous demandons aux volontaires d'attribuer une note à chacun des transferts effectués (voir le chapitre 6). L'objectif est de mettre en évidence leur satisfaction vis à vis de la distance entre le robot et l'humain, de l'accessibilité de l'objet au moment de l'échange et du déplacement du robot.

Il est proposé en figure 5.17, un schéma global qui reprend l'ensemble des algorithmes mis en place pour que ce transfert d'objet soit réalisable. Ces blocs donnent de manière simplifiée les explications réalisées dans le chapitre 4 et dans ce chapitre 5. Il est donc plus détaillé que l'organigramme présenté en figure 5.16 qui ne précise pas directement comment chacune des fonctionnalités sont développées.

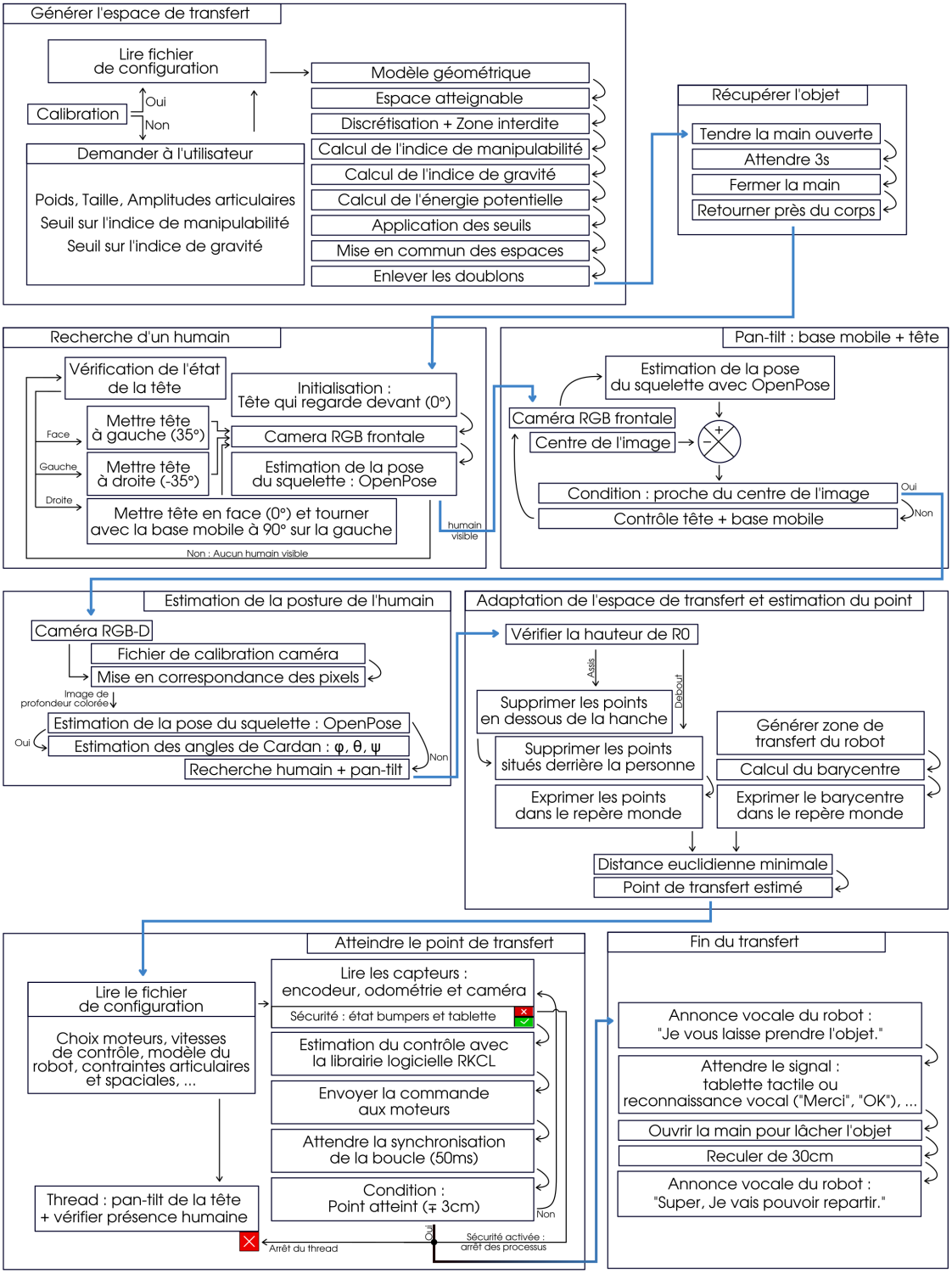


FIGURE 5.17 – Représentation de l'algorithme qui permet au robot Pepper de transmettre des objets de manière sûre et confortable. Ce schéma global présente l'ensemble des blocs de commande de notre application.

5.5 Localisation et navigation du robot

Dans le cadre du projet à la cité des aînés de Montpellier et de Saint-Étienne (voir chapitre 1), il nous est nécessaire d'ajouter des fonctionnalités de localisation et de navigation au sein de l'établissement. En effet, pour mener à bien nos applications et permettre de rendre le robot le plus autonome possible, il faut donner la possibilité au robot de se localiser et se déplacer jusqu'aux résidents pour fournir le service demandé. Le stagiaire G. PETEMOYA a participé au développement de ce projet en travaillant sur ces applications de localisation et de navigation du robot Pepper.

5.5.1 Localisation à l'aide de marqueurs

Les capteurs du robot sont rapidement saturés ou pas assez précis. Des études ont été menées sur l'utilisation de ces capteurs et de la caméra de profondeur [119], [120] pour améliorer la localisation du robot. Nous souhaitons mettre en place un moyen simple, rapide et efficace afin de déployer au plus vite nos applications. Nous avons donc opté pour l'utilisation de marqueurs : des ArUco [121] [122]. Nous nous sommes inspirés des travaux présentés directement sur le Github de la société SoftBank Robotics [14], mais qui ne sont disponibles que pour les versions supérieures du robot (en langage de programmation Java ou Kotlin, donc pour Pepper ayant NAOqi 2.9). Il n'est pas possible dans notre cas d'utiliser ces développements car la version du robot ne peut être mise à jour vers NAOqi 2.9 (voir la section 3.3.1).

Dans un premier temps, il nous faut choisir l'emplacement désiré des marqueurs. Dans le but de déployer ce service au sein d'un étage de l'EHPAD, nous avons fait le choix, pour commencer, de travailler et de valider nos développements dans une seule et unique chambre de l'établissement. La figure 5.18 permet de visualiser un plan de cette chambre ainsi que le positionnement des ArUco.

Chaque ArUco est ensuite enregistré dans un fichier json ("JavaScript Objet Notation") afin de pouvoir à tout moment récupérer toutes les informations le concernant (positions, dépendances, ...). L'odométrie du robot nous donne une estimation de sa position, par rapport à son emplacement de départ (au démarrage de Pepper) selon ses déplacements effectués. Le fait de pouvoir identifier ces marqueurs et de connaître leur disposition, permet à nos algorithmes d'ajuster l'odométrie du robot et de connaître plus précisément sa position dans l'établissement. En effet, des fonctions nous permettent d'avoir la position de l'ArUco par rapport à la caméra du robot et donc on peut savoir où le robot se situe par rapport à ce marqueur.

L'organigramme présenté en figure 5.19 donne une représentation de l'algorithme mis en place pour que le robot se géolocalise à partir d'ArUco (marqueur). Deux parties sont visibles dans cette figure : chercher un marqueur et estimer la position du robot ainsi que le plus court chemin pour naviguer jusqu'à la zone souhaitée. En effet, le robot va regarder autour de lui en tournant simplement la tête de gauche à droite puis tourner sur lui-même si besoin jusqu'à apercevoir un ArUco au sol. L'algorithme sort alors de cette boucle pour lancer la partie du programme qui concerne l'estimation

14. https://github.com/softbankrobotics-labs/pepper-aruco

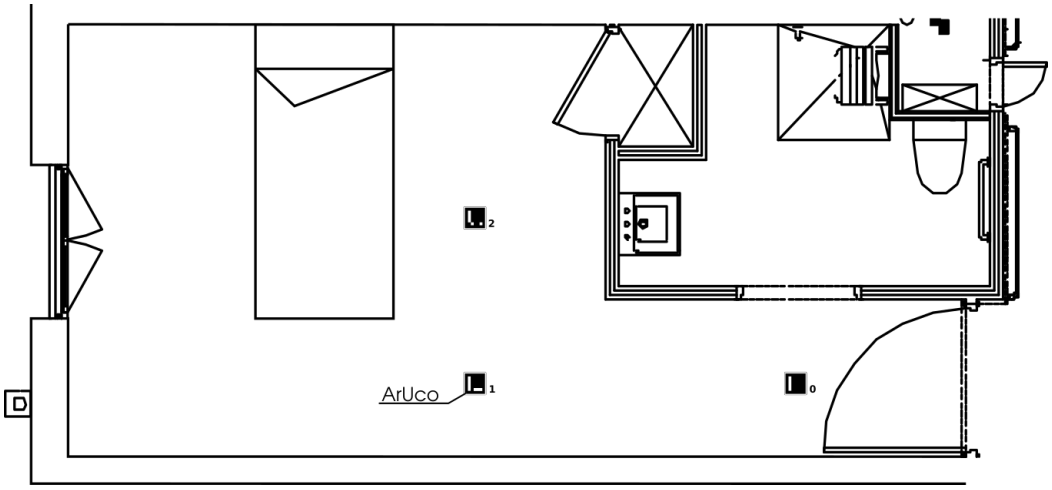


FIGURE 5.18 – Plan d'une chambre de la cité des aînés de Montpellier avec positionnement des ArUco définissant le chemin à suivre pour aller jusqu'au bord du lit.

de la position du robot dans la carte (définie par ces ArUco) et du plus court chemin pour atteindre le point d'intérêt désiré. Ce plus court trajet est obtenu à l'aide de l'algorithme de Dijkstra [123] qui est présenté dans la section 5.5.2 (voir l'algorithme 1).

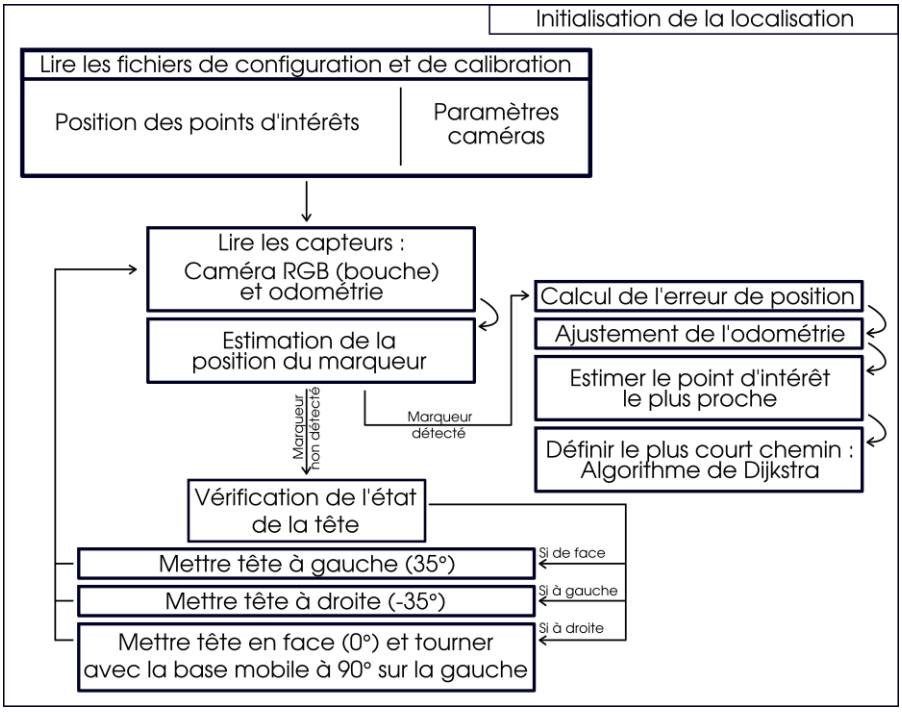


FIGURE 5.19 – Représentation de l'algorithme qui permet au robot Pepper de se repérer à l'aide de marqueurs, puis d'estimer le plus court chemin pour atteindre la zone souhaitée.

5.5.2 Estimation du plus court chemin

Chaque ArUco est enregistré selon sa position dans un graphe par rapport à un marqueur de référence situé vers la base de rechargement par exemple. Il est possible également d'ajouter des points d'intérêts virtuels dans ce graphe. Ce qui permet de définir des positions à atteindre dans la carte sans spécifiquement en ajouter au sol. Dans ce graphe, il y a des marqueurs de passage qui permettent de définir les chemins et des marqueurs spécifiques qui indiquent des points d'intérêts dans la carte (bord de lit, vers la station de recharge, ...). On enregistre alors ces points d'intérêts dans nos algorithmes pour qu'à tout moment, lorsqu'un ArUco du graphe est identifié, le robot puisse estimer le plus court chemin pour atteindre le lieu désiré.

L'algorithme de Dijkstra [123] nous permet de trouver le plus court chemin dans un graphe. Celui-ci est un réseau représenté par des sommets qui sont les points d'intérêt de la carte que l'on souhaite parcourir. On définit alors les dépendances entre chacun des points (définis comme les arêtes du graphe). On donne à l'algorithme le sommet de départ puis le sommet final que l'on souhaite atteindre. Ensuite, l'algorithme parcourt l'ensemble du réseau en gardant l'information de distance entre chaque point. Il délivre alors les chemins les plus courts pour atteindre l'objectif selon l'endroit où l'on souhaite aller. En donnant le nœud de fin (ArUco à atteindre), cet algorithme rend en sortie un vecteur avec l'ensemble des nœuds (identifiants d'ArUco) à parcourir pour aller du point de départ jusqu'au point d'arriver le plus vite possible.

Le pseudo-code 1 montre la méthodologie de l'algorithme de Dijkstra utilisée pour estimer le plus court chemin dans notre étude. Cet algorithme est découpé en trois parties, l'initialisation, l'estimation et enfin la reconstitution du chemin à parcourir. Nous expliquons les grandes étapes de cet algorithme en s'appuyant sur le pseudo-code 1.

Partie "initialisation"

L'algorithme va récupérer les informations qui définissent le graphe dans un fichier de configuration $Fichier Conf$ (ligne 1). Les données recueillis par ce fichier remplissent les vecteurs **sommets**, **aretes** et **poids**. Le vecteur **sommets** contient alors tous les identifiants d'ArUco du graphe. Le vecteur **aretes** a toutes les dépendances existantes entre chaque sommet du graphe et **poids** définit l'ensemble des valeurs attribuée aux arêtes. Puis un vecteur f est initialisé pour contenir l'ensemble des sommets (ligne 2). Il sera actualisé dans la suite de l'algorithme pour savoir si le sommet courant étudié a déjà trouvé son chemin le plus court ou s'il doit encore être traité. Le tableau de Dijkstra est initialisé à la ligne 3 et 4. Cela veut dire que le vecteur **d**, qui contiendra les distances parcourues au cours de l'estimation, est initialisé avec pour chaque composante une valeur "infinie". Ce vecteur a la même taille que **sommets** et dans notre cas, cette valeur "infinie" est de 10^5 pour être suffisamment grande. Le vecteur **p** est aussi initialisé avec la même taille que **sommets**. Il sera rempli au cours de l'estimation pour définir à chaque sommet, l'arête à prendre pour atteindre le nœud d'arrivées. La ligne 4 initialise la distance du sommet de départ $NoeudDepart$ à la position adéquate dans le vecteur **d** à 0.

Algorithm 1: Pseudocode de l'algorithme de Dijkstra utilisé.

```
   Input: NoeudDepart, NoeudFin, FichierConf
   Output: chemin = { NoeudFin, ..., NoeudDepart }
1  poids, sommets, aretes ← LireFichierDeConfiguration(FichierConf);
2  f = sommets;
3  d, p ← InitialisationDuTableauDeDijkstra();
4  d = InitialisationDuSommetDeDepart(sommets, NoeudDepart);
5  n = f.size();
6  while n != 0 do
7      u = TrouverLaDistanceLaPlusCourte(d, f);
8      for k = 1, 2, ..., aretes.size() do
9          v = IdentifierLeSommetDependant(u, sommets, aretes[k]);
10         if (v!=-1) and (d[v]>(d[u]+poids[k])) then
11             d[v] = d[u] + poids[k];
12             p[v] = sommets[u];
           end
       end
13     f[u] = "NaN";
14     n = NombreDeSommetNonTraverse(f);
   end
15 chemin.push_back(NoeudFin);
16 i = IdentifierLaPositionDuNoeudDArriver(NoeudFin, sommets);
17 while (p[i] != NoeudDepart) do
18     chemin.push_back(p[i]);
19     for k = 1, 2, ..., sommets.size() do
20         if sommets[k] == p[i] then
21             i=k;
           end
       end
   end
22 chemin.push_back(NoeudDepart);
```

Partie "estimation"

La valeur de n (initialisée à la ligne 5) contient l'information permettant d'arrêter l'estimation du plus court chemin. En effet, il définit la condition de la boucle *while* (ligne 6) et une vérification est faite en fin de cette boucle (ligne 14) afin de déterminer si des sommets n'ont pas encore été traités. L'algorithme identifie la distance la plus petite pour les sommets non traités (c'est pourquoi le vecteur **f** est également vérifié ici) à la ligne 7. La position dans le vecteur **d** est notée dans la variable u. La boucle *for* à la ligne 8 parcours l'ensemble des arêtes du graphe. À la ligne 9 l'algorithme identifie si un sommet est dépendant et il donne sa position v dans le vecteur **sommets**. C'est dans cette boucle qu'il est possible de vérifier (ligne 10) si l'arête observée permet de

parcourir moins de distance (distance du sommet étudié en plus du poids de l'arête : $\mathbf{d}[u] + \mathbf{poids}[k]$) que les valeurs déjà enregistrées dans **d** pour ce sommet dépendant ($\mathbf{d}[v]$). Ici, si à la ligne 9 l'arête vérifiée ne permet pas d'identifier le sommet traité, alors $v = -1$ et on continue de parcourir le vecteur **aretes**. Lorsque la condition est vérifiée en ligne 10, on enregistre cette nouvelle distance au sommet dépendant : $\mathbf{d}[v]$ (ligne 11). Puis on enregistre aussi le nom du sommet traité comme étant la référence du sommet dépendant à la ligne 12. Cela permettra d'identifier à la fin les sommets à parcourir pour définir le plus court chemin. La valeur $\mathbf{f}[u]$ est alors inscrite comme traitée à partir de ce moment.

Partie "reconstitution du chemin"

Le vecteur **p** est maintenant rempli, il faut donc le parcourir pour avoir la liste d'ArUco à suivre. Le premier à enregistrer est le sommet final : N$oeud$Fin (ligne 15). On identifie sa position dans le vecteur **sommets** et on enregistre cette position dans la variable i (ligne 16). Cette dernière sera mise à jour à la ligne 21 au fur et à mesure que le vecteur **p** est parcouru. En effet, la boucle *while* tourne jusqu'à ce que le sommet final soit atteint (ligne 17). À l'intérieur de cette boucle, l'algorithme ajoute l'identifiant du sommet enregistré à l'emplacement **p**[**i**] dans le vecteur **chemin** (ligne 18). Puis de la ligne 19 à 21, la position i est actualisée en cherchant cet identifiant dans le vecteur **sommets**. Pour finir, on enregistre l'identifiant de départ N$oeud$D$epart$ (ligne 22). La sortie de l'algorithme nous donne le vecteur **chemin** qui contient l'ensemble des sommets à parcourir afin d'aller du sommet N$oeud$D$epart$ jusqu'au sommet N$oeud$Fin.

L'algorithme 1 donne alors le chemin à parcourir pour atteindre l'ArUco situé au bord du lit. La figure 5.20 présente le déplacement du robot que l'on souhaite faire entre l'entrée de la chambre et le bord du lit du résident.

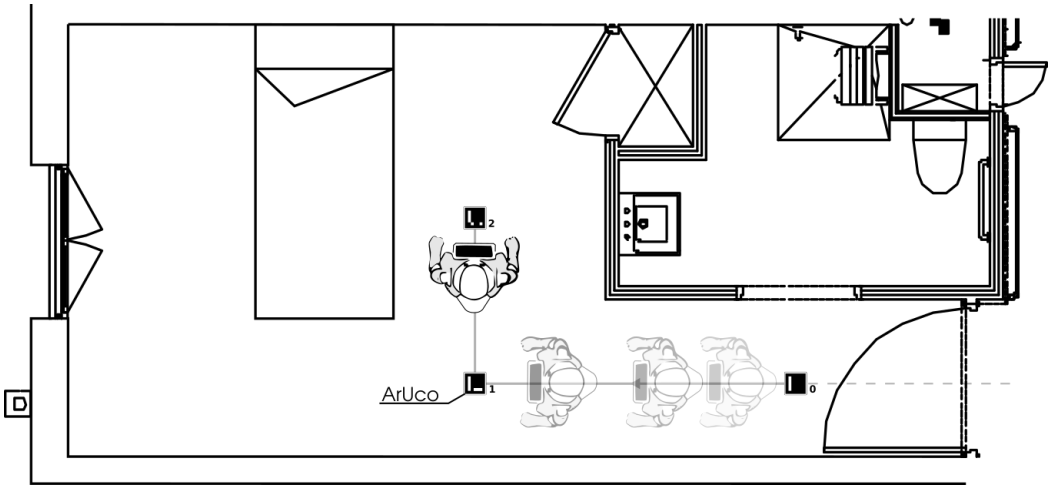


FIGURE 5.20 – Schéma présentant le déplacement désiré du robot Pepper dans le plan d'une chambre de la cité des aînés de Montpellier pour aller jusqu'au bord du lit.

5.5.3 Commande du déplacement

Après avoir identifié la position d'un ArUco, le robot cherche le point le plus proche de lui. Cela lui permet de se replacer sur le chemin s'il n'y est pas. On définit ensuite l'angle que doit avoir le robot pour se diriger vers le prochain marqueur. Puis, on attribue une vitesse au robot sur l'axe x ($0,27 m.s^{-1}$), permettant au robot d'avancer simplement vers l'avant.

Lorsqu'on demande au robot de passer d'une vitesse de $0,27 m.s^{-1}$ à $0 m.s^{-1}$, NAOqi fait une diminution progressive de la vitesse pour atteindre la vitesse nulle. Cela évite de projeter le robot en avant en freinant d'un coup sec. On observe donc un léger glissement du robot à l'arrivée de chaque marqueur. Pour limiter ce glissement, on définit une zone de ralentissement autour de l'ArUco pour que le robot passe à une vitesse plus faible ($0,10 m.s^{-1}$). Cette zone est définie autour de chaque ArUco ou point virtuel du graphe par un cercle virtuel de rayon $50 cm$. Pour aller plus vite, on considère également que le robot a atteint le marqueur lorsqu'il entre dans un rayon de $20 cm$ autour du marqueur. La figure 5.21 permet de visualiser plus concrètement ces zones de ralentissement, ainsi que le rayon de $20 cm$ pour valider l'atteinte du marqueur et passer au suivant.

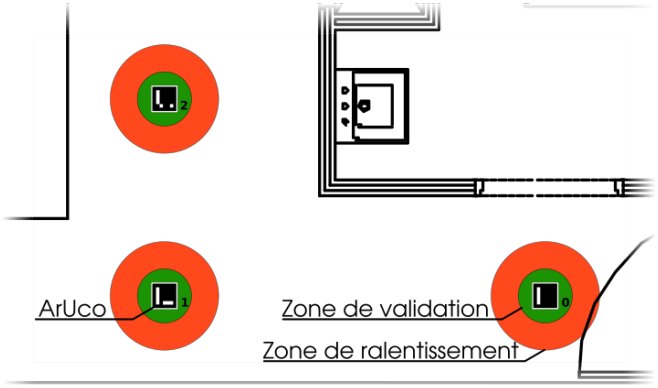


FIGURE 5.21 – Schéma présentant les marqueurs ainsi que leur zone de ralentissement ($50 cm$ de rayon représenté en rouge) et leur zone pour valider l'atteinte du marqueur ($20 cm$ de rayon représenté en vert).

L'organigramme présenté en figure 5.22 donne une représentation de l'algorithme mis en place pour que le robot puisse naviguer dans l'établissement tout en se localisant en permanence à l'aide des marqueurs. Deux parties sont visibles dans cette figure. La première concerne le thread qui permet de se localiser en fonction du flux d'images récupéré et analysé par les librairies dédiées à l'identification et l'estimation de la pose de ces ArUco [15]. Cette partie permet de mettre à jour, de manière continue, l'erreur de position observée sur l'odométrie du robot et donc d'ajuster la position de Pepper. Puis, la seconde partie, concerne la navigation du robot avec les multiples vérifications effectuées afin de savoir si le robot est dans la bonne direction, s'il a atteint la

15. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

zone de freinage ou la zone de validation. Enfin, le robot passe ainsi de point en point, en respectant le chemin qu'il doit suivre et jusqu'à atteindre la zone souhaitée.

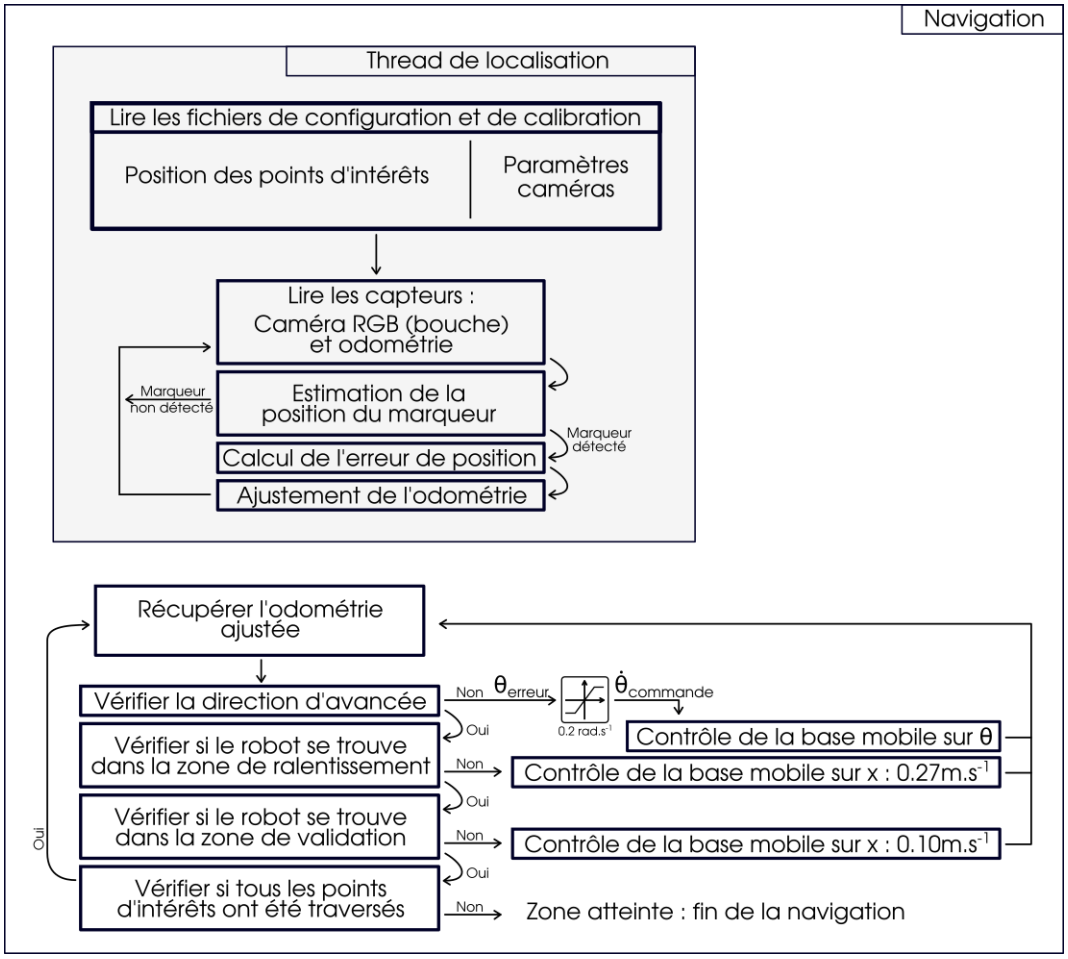


FIGURE 5.22 – Représentation de l'algorithme qui permet au robot Pepper de naviguer en suivant le chemin pré-défini, tout en vérifiant en permanence sa localisation.

5.6 Conclusion

Dans cette partie, nous expliquons l'ensemble de la stratégie mise en place pour commander le robot et mettre en application la zone de transfert ergonomique générée (voir le chapitre 4). Ces développements nous permettent d'analyser l'environnement, de déterminer la position du receveur, d'estimer le point de transfert idéal et enfin d'effectuer la transmission de l'objet. Nous utilisons le module RKCL pour la cinématique inverse permettant de générer des trajectoires et de suivre des points générés par notre stratégie de commande. Nous contrôlons ainsi la base mobile et les moteurs du bras qui porte l'objet afin d'atteindre rapidement et simplement le point de transfert. On a donc un transfert d'objet avec un mouvement fluide, suffisamment rapide pour ne pas surprendre le receveur et avec un rendu naturel.

L'estimation de la posture de l'humain est réalisée à partir des images obtenues de la caméra RGB frontale et de la caméra de profondeur du robot. Après alignements des informations de couleurs et de profondeurs, nous obtenons une image de profondeur colorée. Nous utilisons donc une librairie externe permettant la reconnaissance et l'estimation de la position du squelette dans cette image. Selon la disposition des épaules et de l'ensemble des points récupérés sur le torse de l'humain observé, il est possible de donner une approximation de la position et la rotation du repère de référence. Connaître la position de ce repère est essentiel pour exprimer notre zone de transfert dans le référentiel humain.

Avant de réaliser cette transmission, nous effectuons aussi quelques ajustements de la zone de transfert de l'humain. En effet, cela nous permet de supprimer des points inutiles, des points pouvant être dangereux pour l'individu, et de prendre en compte également l'accessibilité du robot. L'estimation de la posture et ces derniers ajustements nous permettent de trouver le point de transfert idéal à atteindre.

Nous présentons également dans cette partie nos travaux concernant la localisation et la navigation du robot au sein d'un EHPAD. L'objectif étant de se rapprocher au plus près de l'environnement de l'humain pour lui éviter de se lever, limiter certains déplacements et donc par ce biais les risques de chute.

Chapitre 6

Résultats et discussion

Sommaire

6.1 Introduction 106
6.2 Première expérimentation 107
6.2.1 Déroulement de l'étude 107
6.2.2 Analyse des résultats 109
6.2.3 Problèmes rencontrés et solutions apportées 110
6.3 Seconde expérimentation 112
6.3.1 Déroulement de l'étude 112
6.3.2 Analyse des résultats 114
6.4 Conclusion 118

6.1 Introduction

Afin d'évaluer les performances de nos développements, nous avons mis en place des expérimentations auprès de différentes personnes, dont des personnes âgées résidant en maison de retraite. Elles consistent à générer différentes zones et à réaliser différents transferts du robot vers l'humain afin d'évaluer la satisfaction du receveur. Cela nous permet dans un premier temps de valider l'interaction entre le robot et l'humain dans le cadre de cette application. Puis de mettre en évidence les seuils sur les indices qui sont généralement plus appréciés. En effet, afin de vérifier la satisfaction des receveurs, nous leur demandons de répondre à de simples questionnaires. Ils peuvent par ce biais juger leurs préférences en attribuant une note et nous spécifier tout autre avis par commentaire s'il le souhaite.

Pour cette partie expérimentale, nous avons choisi de présenter deux expérimentations. Elles diffèrent légèrement dans leurs mises en place et leurs questionnaires utilisateurs. L'objectif final est le même : tester nos développements avec des volontaires et trouver une combinaison de seuils acceptables pour le transfert d'objets. Nous expliquons dans cette partie en quoi ces deux démonstrations sont différentes.

6.2 Première expérimentation

Au départ, nous voulions simplement tester nos algorithmes, relever les éventuels dysfonctionnements et voir si les zones générées étaient cohérentes avec nos hypothèses. Nous avons donc réalisé des expérimentations dans un premier temps pour valider le principe d'interaction avec des personnes plutôt jeunes (huit personnes dont l'âge moyen est de 38,5 ans).

6.2.1 Déroulement de l'étude

Dans cette première étude, quatre transferts sont proposés et la personne doit juger, à chaque fois, si le transfert était acceptable en le notant sur une échelle de Likert (sur 5 niveaux). Après avoir réalisé ces quatre transferts, la personne doit aussi les classer par ordre de préférence afin de nous permettre de confirmer les notes précédemment attribuées de manière individuelle.

Avant chaque expérimentation, un ensemble de documents est remis aux volontaires concernant cette étude. Une première feuille explique l'objectif et ce qui va se passer concrètement durant cette expérimentation (voir annexe C.1.1). Pour chaque expérimentation réalisée, le volontaire doit signer la feuille de consentement (voir annexe C.1.2) et s'il nous autorise, le droit à l'image (voir annexe C.1.3). On leur demande ensuite de remplir un premier questionnaire afin de connaître leur taille, leur poids, et leurs limitations articulaires concernant leurs épaules et leurs coudes (première page présentée en annexe C.1.4).

Nous demandons à la personne de se placer debout devant le robot à une distance d'environ deux mètres, puis nous lançons l'expérimentation. Notre algorithme commence par générer le nuage de points de la zone de transfert en fonction des informations fournies dans le premier questionnaire. Une fois cette première étape d'initialisation terminée, le robot peut faire une estimation de la position de l'humain à l'aide de la librairie OpenPose sur l'image provenant de sa caméra RGB-D. Le référentiel humain peut alors être exprimé à tout moment en fonction de la position du robot.

La commande du bras droit et de la base mobile du robot est réalisée afin d'atteindre le point de transfert estimé. Lorsque la position est atteinte, Pepper annonce à la personne qu'elle peut prendre l'objet. Après 2 secondes, le robot ouvre légèrement la main afin de faciliter la prise de l'objet par le volontaire. La personne peut ensuite répondre au questionnaire de satisfaction concernant le transfert effectué (voir la première page de l'annexe C.1.4).

Enfin, après avoir réalisé ces quatre transferts, on demande à la personne de répondre à trois dernières questions (voir la deuxième page de l'annexe C.1.4). Ces questions sont plus générales sur l'ensemble des applications et les personnes peuvent ajouter des commentaires pour expliquer leurs ressentis si elles le souhaitent. Les questions posées concernent l'approche du robot (mouvement), la distanciation et la pertinence de l'étude.

Pour des raisons de sécurité, la personne peut à tout moment activer l'un des bumpers du robot pour stopper son avancée et arrêter le transfert. De plus, nous contrôlons

le robot à faible vitesse ($0,1 m.s^{-1}$ maximum) afin d'anticiper tout problème et éviter de blesser une personne. Un bouton d'arrêt d'urgence est également présent sur le robot (dans sa nuque) si les bumpers ne suffisent pas à arrêter le robot. C'est également pour garantir la sécurité que nous préférons effectuer ces transferts avec des volontaires debout afin de leur permettre de se déplacer si nécessaire.

Les quatre transferts proposés aux volontaires nous permettent de diversifier les seuils pour obtenir des formes plus ou moins différentes. Pour mieux visualiser ces formes, nous montrons sur la figure 6.1 les formes générées pour une personne n'ayant aucun trouble articulaire sur ses membres supérieurs.

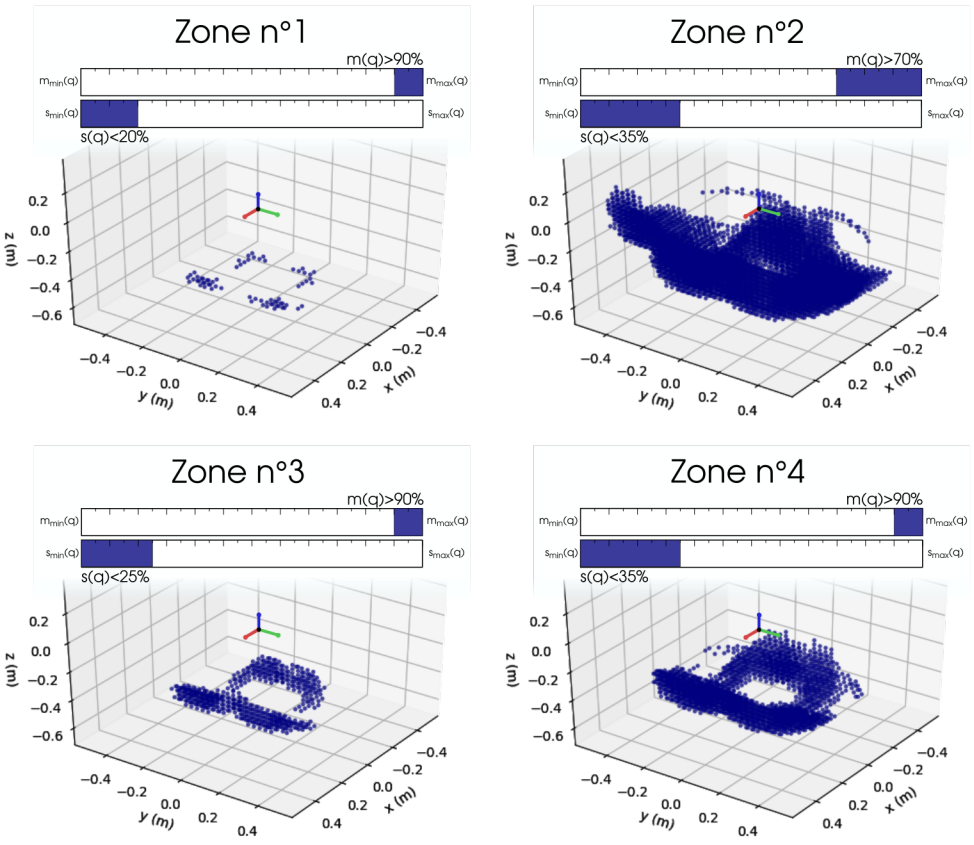


FIGURE 6.1 – Ensemble des points obtenus pour la première expérimentation par différents seuils sur l'indice de gravité terrestre ($s(q)$) et sur l'indice de manipulabilité ($m(q)$). Le pourcentage représente l'emplacement du seuil par rapport à la valeur maximale des indices.

Les quatre combinaisons présentées sur la figure 6.1 sont les suivantes :

— Zone n°1: nous conservons les points dont l'indice de manipulabilité est supérieur à 90% de sa valeur maximale et les points dont l'indice de gravité terrestre est inférieur à 20% de sa valeur maximale.

— Zone n°2: nous conservons les points dont l'indice de manipulabilité est supérieur à 70% de sa valeur maximale et les points dont l'indice de gravité terrestre

est inférieur à 35% de sa valeur maximale.

— Zone n°3: nous conservons les points dont l'indice de manipulabilité est supérieur à 90% de sa valeur maximale et les points dont l'indice de gravité terrestre est inférieur à 25% de sa valeur maximale.

— Zone n°4: nous conservons les points dont l'indice de manipulabilité est supérieur à 90% de sa valeur maximale et les points dont l'indice de gravité terrestre est inférieur à 35% de sa valeur maximale.

6.2.2 Analyse des résultats

Ces transferts sont l'occasion pour nous de confronter nos développements avec des personnes extérieures au projet. Les réponses aux questionnaires sont présentées en annexe C.1.5. Les positions de transfert atteintes ont globalement été appréciées par les volontaires. L'échantillon de participants est composé de huit personnes dont l'âge est compris entre 23 et 71 ans (moyenne 38,5 ans ± 18,35 ans). Selon leurs préférences, les classements obtenus pour ces zones de transfert (figure 6.1) sont présentés sous la forme d'un histogramme en figure 6.2.

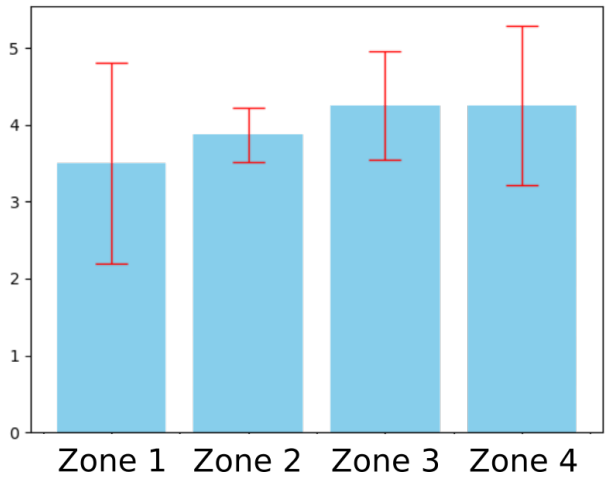


FIGURE 6.2 – Histogramme montrant les scores obtenus (sur 5) pour chaque zone générée de la première expérimentation.

Certains volontaires ont également ajouté des commentaires en supplément des notes attribuées sur chaque transfert. Cela nous permet de justifier les scores obtenus en histogramme 6.2. En effet, de manière générale la zone n°1 n'a pas été appréciée, car la zone atteinte par le bras du robot était trop basse pour donner l'objet à la personne. Pour la zone n°2, c'est l'inverse, le robot était cette fois-ci trop loin. Les zones n°3 et n°4 sont plus appréciées, parce que le point de transfert est globalement plus haut et plus proche de la personne. Les volontaires semblent préférer que le robot donne l'objet dans une zone suffisamment proche, leur permettant d'effectuer le moins de mouvements possible. Les résultats montrent donc qu'il faut un certain dosage entre les seuils pour éviter un transfert trop loin ou trop bas par rapport à la personne.

Dans notre questionnaire, concernant les appréhensions vis à vis de l'expérimentation et du robot, ils ont tous répondu "non" et ont même manifesté leur curiosité. À la fin de l'expérimentation, nous posons trois autres questions. La première donne les informations suivantes : le mouvement du robot n'a pas surpris, mais ne semble pas toujours adapté et naturel. La seconde nous renseigne sur la distanciation : bonne en général, mais aussi parfois peu naturelle. Enfin, la dernière question est plus générale et leur demande de juger de la pertinence de l'étude. Ces volontaires constatent que ce type d'aide peut être très utile pour les personnes dont la mobilité est réduite, à domicile ou en institution. Dans notre cas, cette fonction d'apporter des objets du quotidien leur semble globalement être une bonne aide.

Nous proposons en figure 6.3 une série d'images permettant de visualiser l'un des transferts effectués lors de cette première expérimentation.



FIGURE 6.3 – Séquence d'images permettant de visualiser le transfert d'un objet par le robot Pepper à une personne étant debout.

6.2.3 Problèmes rencontrés et solutions apportées

Pour cette première expérimentation, nous avons rencontré quelques problèmes principalement liés à l'interaction avec le robot. Les avis des participants nous ont permis d'aller plus loin et d'améliorer notre dispositif. En effet, le robot tourne parfois trop sur lui-même et atteint le point de transfert dans une position non-naturelle. Dans ces positions, le robot ne regarde plus dans la direction du receveur. Cette légère rotation au niveau de la base fait perdre le contact visuel car le robot ne regarde plus dans la direction où se trouve la personne. Le fait de ne plus regarder dans la direction de la personne rend l'interaction bien moins naturelle et peut perturber le receveur. Certaines études ont permis de mettre en évidence l'importance du regard durant un transfert d'objet avec un humain (voir section 2.3.1).

Pour palier à ce problème, nous avons ajusté le correcteur utilisé par le contrôleur RKCL sur la rotation de la base. Cela permet au robot de trouver une solution plus rapide avec le déplacement de son bras et ses mouvements linéaires par rapport au sol. Puis afin de garder ce contact avec le receveur, nous avons ajouté une tâche en parallèle qui permet de contrôler les moteurs de la tête du robot. En effet, en utilisant l'analyse du squelette par OpenPose, nous avons la position de la tête dans l'image. Cela permet à nos algorithmes de garder cette position au centre de l'image par un

simple contrôle des moteurs de la tête du robot. On peut alors parler de dispositif pan-tilt capable par sa motorisation de regarder dans différentes directions ou de suivre un objet mobile (par rapport au référentiel de la caméra) ou un individu dans notre cas.

Le robot pouvait également faire des erreurs d'estimation si plusieurs personnes étaient face à sa caméra frontale. Nous avons donc cherché à estimer la personne la plus proche (en utilisant la caméra de profondeur) afin d'éliminer les autres personnes qui pouvaient se trouver dans le champ de vision. Ces erreurs nous ont également permis de constater que la sécurité n'était pas suffisante pour arrêter le robot. En effet, les bumpers ne sont pas si faciles à utiliser pour demander de stopper le mouvement du robot. Nous avons donc ajouté la possibilité d'arrêter le robot en touchant la surface de sa tablette.

Pour donner l'objet à la personne, on utilise au départ tous les moteurs du bras du robot et le déplacement possible par la base mobile. Nous avons choisi de transmettre l'objet en un point donné, sans imposer de rotation sur le membre terminal du robot. Cependant, la rotation du coude "RElbowYaw" et celle du poignet "RWristYaw" induit parfois une légère rotation qui rend la position finale de la main peu naturelle. Puisque ces moteurs ne sont pas essentiels dans notre cas, nous choisissons de ne plus les contrôler (voir section 5.4.2). C'est donc avec toutes ces améliorations que nous décidons de faire une deuxième expérimentation cette fois-ci auprès de personnes plus âgées.

6.3 Seconde expérimentation

Suite à notre première expérimentation, nous avons pu réaliser de nombreux progrès. Cela concerne majoritairement l'interaction humain-robot, les ajustements sur le contrôle permettant une approche plus naturelle et enfin l'ajout de sécurité lors du transfert. Ces progrès nous permettent donc de partir plus sereinement sur une nouvelle expérimentation, cette fois-ci en incluant des personnes plus âgées résidant en EHPAD (sept résidents dont l'âge moyen est de 91,86 ans). C'est en effet le public visé par nos développements et c'est une opportunité pour nos recherches de tester et d'avoir de précieux retours de leurs parts. Puis nous avons inclus des personnes plus jeunes permettant d'avoir un total de douze volontaires (âge moyen à 68,83 ans).

C'est également l'occasion d'aller un peu plus loin dans ce projet en permettant au robot de naviguer en EHPAD (en chambre) jusqu'au bord du lit de la personne âgée afin de lui rendre ce service. On a donc un aperçu bien plus concret de l'application finale recherchée.

6.3.1 Déroulement de l'étude

Comme pour la première expérimentation, un ensemble de documents est remis aux participants de l'étude. Il y a toujours la lettre d'information qui explique l'objectif et ce qu'il va se passer durant cette expérimentation (annexe C.2.1). Puis les papiers que doit signer le participant : la feuille de consentement (annexe C.2.2) et le droit à l'image (annexe C.2.3). Enfin, on leur propose le questionnaire (annexe C.2.4).

Pour une meilleure estimation des préférences, nous avons effectué cette fois-ci quatre transferts par zone générée. Cela porte le nombre de transferts à douze car nous nous concentrons dans cette seconde expérimentation sur seulement trois zones différentes. Les questionnaires permettront de déterminer la zone la plus appréciée par le volontaire. Puis avec cette zone, nous réaliserons quatre nouveaux transferts en faisant naviguer le robot jusqu'au bord du lit. En effet, cela permet de donner une application plus complète de notre étude et de valider également le service dans sa globalité.

Pour mieux visualiser ces trois zones obtenues, comme précédemment, nous proposons la figure 6.4 qui représente les formes générées pour une personne n'ayant aucun trouble articulaire sur ses membres supérieurs.

Les trois combinaisons permettant de générer ces zones sont les suivantes :

— Zone n°1: nous gardons les points dont l'indice de manipulabilité est supérieur à 90% de sa valeur maximale et les points dont l'indice de gravité terrestre est inférieur à 25% de sa valeur maximale. C'est également celle qui a été préférée par les utilisateurs de la première expérimentation.

— Zone n°2: nous gardons les points dont l'indice de manipulabilité est supérieur à 70% de sa valeur maximale et les points dont l'indice de gravité terrestre est inférieur à 40% de sa valeur maximale.

— Zone n°3: nous gardons les points dont l'indice de manipulabilité est supérieur à 50% de sa valeur maximale et les points dont l'indice de gravité terrestre est inférieur à 50% de sa valeur maximale.

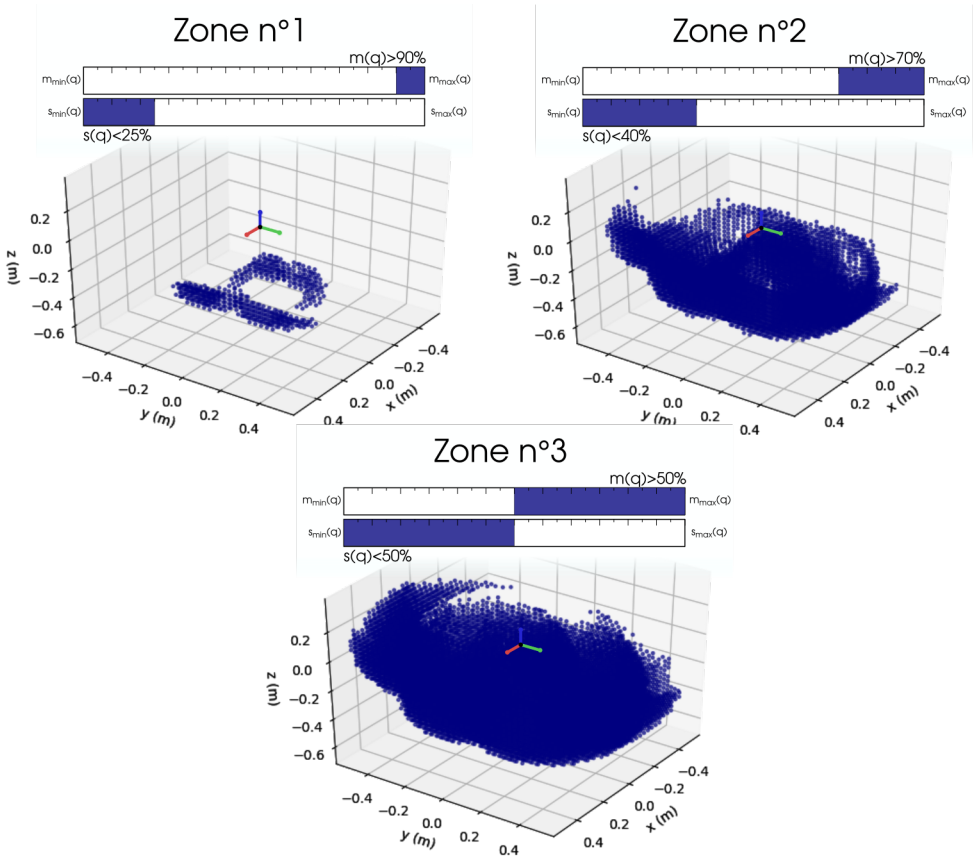


FIGURE 6.4 – Ensemble des points obtenus pour la deuxième expérimentation par différentes minimisations de l'indice de gravité terrestre ($s(q)$) et maximisations de l'indice de manipulabilité ($m(q)$). Le pourcentage représente l'emplacement du seuil par rapport à la valeur maximale. Le repère de référence R_0 représente le repère situé entre les deux épaules de l'humain.

Ces trois zones sont réparties aléatoirement sur les 12 transferts pour éviter tout biais inconscient. La personne est cette fois-ci assise sur le lit pour ces démonstrations et n'est plus debout.

Comme précédemment, nous demandons quelques informations à la personne : poids, taille et limites des amplitudes articulaires. Notre algorithme génère la zone de transfert ergonomique. Le robot va dans un premier temps se tourner face à la personne. Pour cela, il va pivoter sur sa base mobile et baisser ou lever sa tête pour centrer la personne dans l'image. À partir de là, tout se passe comme lors de la première expérimentation : le robot estime la pose de l'humain, cartographie le nuage de points autour de l'humain et estime le point de transfert à atteindre. Le transfert peut alors commencer et le robot continue à observer la personne pendant le transfert. Il annonce ensuite à la personne qu'elle peut prendre l'objet lorsqu'il a atteint le point de transfert estimé.

Nous avons ajouté la reconnaissance vocale à ce moment-là avec les fonctions de base de NAOqi afin d'avertir le robot que l'humain tient l'objet. En effet, un simple "merci" suffit pour que le robot ne cesse d'attendre et ouvre doucement la main. Le fait de toucher la tablette permet également de laisser tomber l'objet. Enfin, le robot remercie la personne d'avoir pris l'objet et recule de trente centimètres.

6.3.2 Analyse des résultats

Cette nouvelle expérimentation est une réelle opportunité pour nous de tester nos développements auprès de personnes âgées en maison de retraite. En effet, il nous a semblé tout à fait pertinent de présenter des résultats avec le public visé : des personnes plus fragiles, à risque de dépendance et/ou partiellement dépendantes, pouvant présenter des troubles articulaires aux membres supérieurs. Les réponses aux questionnaires sont présentées en annexe C.2.5.

Dans un premier temps, nous avons réalisé cette étude auprès de sept personnes dont l'âge est compris entre 86 et 99 ans (moyenne 91,86 ans ± 4,78 ans). Nous avons ensuite inclus des personnes plus jeunes pour obtenir plus de données. Ces nouvelles inclusions nous donnent un total de douze volontaires âgés de 21 à 99 ans (moyenne 68,83 ans ± 30,28 ans).

Il est important pour nous de mettre en avant ces sept premiers volontaires et d'analyser dans un premier temps séparément leurs données. Puis dans un second temps d'évaluer l'ensemble des données obtenues sur ces 12 participants.

Dans cet échantillon de sept résidents, nous avions trois personnes présentant des limitations sur leurs articulations (majoritaires aux épaules). Leur prise en compte nous a permis de générer un nuage de points encore plus centré devant eux et donc d'être beaucoup plus adapté à leurs contraintes articulaires. Si la personne dit qu'elle n'a pas de problème au niveau des articulations du coude et de l'épaule, nous gardons les valeurs maximales définies dans le chapitre 4.

En fonction des notes attribuées par les volontaires aux questionnaires, nous avons obtenu le classement présenté dans l'histogramme de la figure 6.5 pour les zones de transfert générées (Fig. 6.4). Cet histogramme montre les résultats obtenus pour les sept résidents de la maison de retraite (en orange), et les résultats obtenus pour l'ensemble des participants (en bleu).

Durant cette deuxième expérimentation, nous avons enregistré la position du robot, de l'humain et du point de transfert. Or, 26 des 144 transferts réalisés n'ont pu être sauvegardés correctement. Les calculs effectués sont donc établis à partir de 118 transferts. L'ensemble de ces données nous permet de relever la distance qui sépare les torses respectifs du robot et de l'humain (entre les deux repères de référence). Il permet aussi de mesurer la distance entre l'humain et la main du robot au moment du transfert d'objet, puis de récupérer la hauteur du point de transfert. Pour ce dernier, on regarde seulement la composante z qui est exprimée par rapport au sol. Le calcul pour les distances est effectué à partir des axes x et y. L'histogramme de la figure 6.6 montre les moyennes et écart-types obtenus pour chaque zone.

Globalement, il était plus difficile pour les volontaires de voir les différences entre

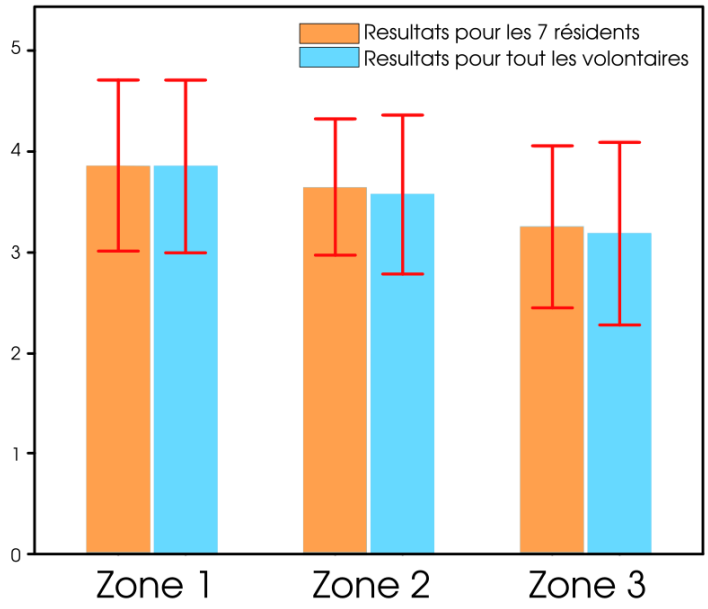


FIGURE 6.5 – Histogramme montrant les notes (sur 5) pour chaque zone générée dans la deuxième expérimentation En orange pour les 7 résidents de la maison de retraite et en bleu pour les 12 participants de cette seconde expérimentation (les 7 résidents + 5 autres personnes).

les différentes zones. En effet, la première expérimentation présentait quatre zones de transfert différentes et la personne était donc sûre que chaque transfert ne serait pas identique. Cette fois-ci, nous avons trois zones générées pour douze transferts et elles sont distribuées aléatoirement pour le questionnaire. Cela ne leur permet pas de savoir si l'une est vraiment différente de la précédente ou de la suivante. Cependant, les résultats obtenus sont considérés comme plus fiables avec cette méthode car ils ne peuvent se référer qu'à leur ressenti à l'instant 't'.

Nous pouvons constater qu'ils préfèrent que l'objet leur soit donné assez près. Dans le contexte de notre expérimentation, cela ne les surprend pas et ils semblent confiants quant à la proximité de l'échange avec le robot. Cependant, malgré la difficulté à noter les transferts, nous pouvons tout de même ressentir une certaine préférence dans leurs notes. En effet, les zones 1 et 2 ont été globalement appréciées car elles étaient toujours facilement accessibles et plus proches. Pour les classements des deux expérimentations, nous nous retrouvons avec la même zone de préférence. Celle où l'indice de manipulabilité est supérieur à 90% de sa valeur maximale et où l'indice de gravité terrestre est inférieur à 25% de sa valeur maximale.

Comme pour la première expérimentation, nous posons la question de l'appréhension, et cette fois-ci une seule personne a répondu "Oui" sans spécialement ajouter de commentaire. Cette personne fait partie des 7 résidents de l'EHPAD. Pour le mouvement du robot, les 12 participants n'ont pas été surpris et semble convenir pour ce genre d'application. Pour la distance, il n'y a pas eu de surprise non plus, seulement

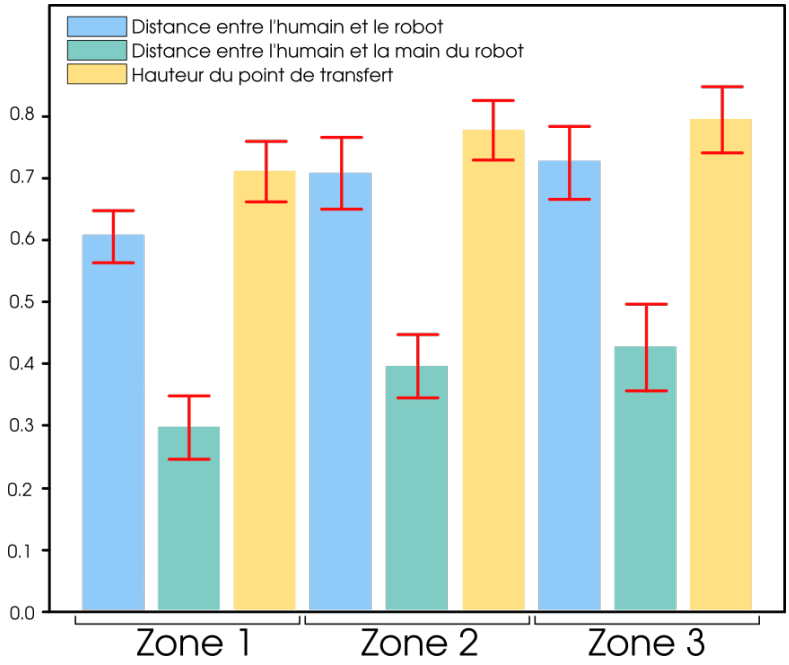


FIGURE 6.6 – Histogramme proposant les distances et les hauteurs moyennes ainsi que leurs écarts-types, exprimées en mètre dans le cadre des transferts d'objet lors de la seconde expérimentation.

quelques transmissions trop lointaines d'après les commentaires.

Enfin, pour la question de la pertinence de l'étude, les personnes les plus concernées (les résidents) restent beaucoup plus neutres face à cette question et donnent parfois leurs craintes. Par exemple, la crainte que le robot remplace l'humain, et appauvrisse les interactions humaines déjà considérées comme trop faibles. Ils craignent également que l'interaction et la compréhension de l'outil robotique ne soient pas si simples pour des personnes aux fonctions cognitives dégradées.

Il nous semble important de noter ici que deux autres résidents ont refusé de participer à cette expérimentation. L'un pour la difficulté à répondre au questionnaire posé et l'autre personne craignait de s'engager en signant des papiers (consentement et droit à l'image). Il est, en effet, important d'expliquer notre travail, nos objectifs, mais aussi nos difficultés rencontrées. Ces deux refus peuvent s'expliquer par un manque de compréhension vis à vis de l'étude ou de l'intérêt général de nos équipes de recherche. Il est donc important d'avoir des personnes ayant de bonnes fonctions cognitives pour être attentives au robot, aux objectifs et répondre correctement aux questionnaires.

La figure 6.7 montre une série d'images permettant de visualiser trois des transferts effectués lors de cette première expérimentation. Une vidéo [1] explicative et démonstrative est également disponible.

1. http://bit.do/Handover-Experiment-With-The-Pepper-Robot

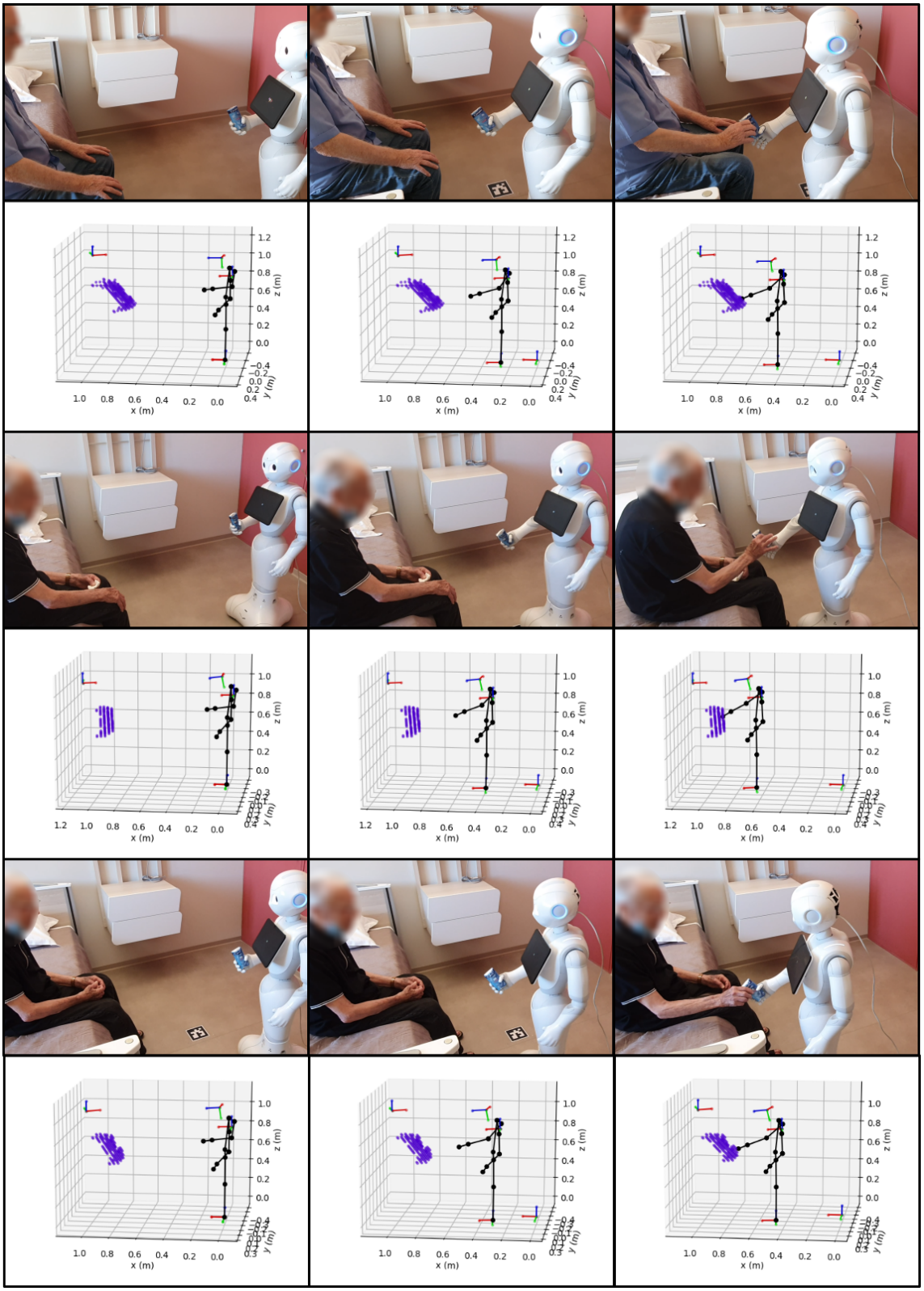


FIGURE 6.7 – Séquence d'images permettant de visualiser trois transferts d'un objet par le robot Pepper à une personne âgée assise. Si l'on regarde de gauche à droite, la première colonne montre le robot en phase initiale avec l'objet tenu en main. Puis la deuxième colonne montre le robot en mouvement et la dernière montre que le point de transfert est atteint et que la personne peut récupérer l'objet.

6.4 Conclusion

Ces expérimentations ont permis de valider l'ensemble de nos algorithmes afin de réaliser des transmissions d'objets confortables directement auprès de volontaires, dont des personnes âgées. En effet, nous avons eu l'opportunité de confronter nos travaux auprès de 20 volontaires (8 sur la première expérimentation et 12 dans la seconde) dont 7 résidents d'un EHPAD, et avoir ainsi leur avis sur l'aide que nous souhaitons mettre en place par le biais de ce robot Pepper. Le but recherché derrière ces expérimentations fut de générer différentes zones de transfert adaptées et d'évaluer la satisfaction du receveur vis à vis des transferts réalisés. Elles nous permettent également de vérifier l'ensemble de nos développements autour de ce transfert d'objet comme la localisation et la navigation du robot.

La première expérimentation, auprès d'un public plus jeune, nous aura permis de noter certains dysfonctionnements et donc d'apporter de nettes améliorations. En effet, nous avons ajouté quelques fonctionnalités permettant une meilleure interaction avec l'humain et une meilleure sécurité en permettant à tout moment de stopper plus facilement le robot si nécessaire. C'est pourquoi, cette deuxième expérimentation a été possible auprès de personnes âgées au sein même d'un EHPAD.

Malgré un premier contact entre ces personnes âgées et le monde de la robotique, ils ne semblaient pas spécialement surpris mais plus amusés par les expérimentations à réaliser. De manière générale, ils restent tout de même craintifs vis à vis de la place d'un robot au sein de ce type d'établissement. Que ce soit vis à vis de certains résidents selon leurs niveaux de dépendance, ou vis à vis du remplacement du personnel soignant pour certaines de leurs tâches. En effet, même pour de simples actions, le robot enlèverait le contact humain que les patients peuvent rechercher. Cependant, ils sont globalement satisfaits par nos applications, ils ne sont pas surpris par la distanciation ou par la rapidité du mouvement qu'a le robot lors de l'échange. Une vidéo a pu être réalisée à la suite de ces expérimentations et est disponible au lien suivant : `http://bit.do/Handover-Experiment-With-The-Pepper-Robot`.

Nous avons pu constater que l'intégration d'un robot de ce type peut être difficile selon le profil de l'utilisateur. En effet, il serait intéressant d'avoir une évaluation préalable du profil cognitif, par exemple avec le MMSE ("Mini-Mental State Examination") [124]. Ce bilan nous permettrait d'évaluer ou de nous faire une idée de la capacité à juger, à comprendre les consignes, à réaliser une action, mais aussi à mémoriser et à être attentif. Grâce à cela, nous pourrions affiner nos critères d'inclusion dans l'étude. En revanche, cela imposerait de solliciter plus de professionnels qualifiés pour réaliser ce type d'évaluation (ergothérapeutes, neuropsychologues, etc.). Nous savons que ces expérimentations sont réalisées sur un trop petit échantillon de personnes. Malgré les résultats encourageants, nous devrons réaliser un plus grand nombre de tests dans un futur proche.

Chapitre 7

Conclusion

L'objectif principal de ce travail est de réaliser le transfert d'un objet du robot à l'homme, à un point considéré comme optimal pour l'homme. Nous partons de l'hypothèse qu'avec deux métriques, nous pouvons établir un nuage de points qui nous permettrait de cartographier les points les plus confortables et les plus manipulables autour de l'individu. En effet, nous nous appuyons sur deux indices. Le premier (indice de manipulabilité) nous permet de savoir si la personne peut facilement changer de position dans une configuration donnée de son bras. Le second (indice de gravité terrestre) nous permet de connaître les positions pour lesquelles l'individu utilise le moins d'énergie pour maintenir sa position et contrer l'effet de la gravité. Ces indices nous permettent d'établir un nuage de points lié à la taille, au poids, aux préférences de la personne et à ses limitations articulaires. Le robot fait correspondre ce nuage de points à la position réelle de l'homme, grâce à la bibliothèque logicielle OpenPose, ainsi que sa caméra de profondeur. Nous proposons d'effectuer les transferts adaptés et personnalisés en fonction de la personne avec laquelle le robot échange des objets.

Les expérimentations réalisées nous ont permis de voir l'efficacité de notre framework et de nous rendre compte de la pertinence de ce type de tâche. Une première démonstration sur un jeune public nous a permis de fixer différents problèmes de notre application et d'avoir un premier avis sur les zones qui pourraient être plus satisfaisantes. Pour la seconde expérimentation, nous avons eu l'occasion d'expérimenter directement avec des personnes âgées et d'évaluer leur satisfaction quant aux zones de transfert générées.

De manière générale, l'objet était toujours facilement accessible et les volontaires étaient bien satisfaits. Leurs réponses aux questionnaires nous permettent néanmoins de voir une préférence dans les transferts effectués sur les zones les plus proches d'elles. Leur satisfaction concerne la forte disponibilité de l'objet, le peu de mouvement à effectuer pour le récupérer, une zone qui leur semble bien adaptée et enfin des mouvements et une distance avec le robot qui ne les surprennent pas.

Réflexions

Durant ce travail de recherche, de nombreuses réunions et échanges ont pu être effectués entre différents professionnels du soins et de la robotique. Leurs multiples conseils, propositions, ainsi que leurs avis ont permis de définir des objectifs communs afin d'améliorer et de mettre en place une stratégie de prise en charge adaptée pour les personnes âgées fragiles. La mise en application de nos développements auprès de ce public vivant au sein d'un EHPAD nous aura également permis d'améliorer notre analyse et notre réflexion vis à vis des robots assistants de demain.

En robotique industrielle, la performance est souvent l'intérêt premier pour réaliser une tâche. L'utilisation de robots assistants en contact direct avec des personnes âgées définit une autre priorité qui est l'acceptabilité. Il est primordial que l'utilisateur se sente en confiance avec ces nouveaux outils. Pour cela, il faut réfléchir au design du robot en termes de look et de fonctionnalité. En effet, comme pour n'importe quel objet, il est important que la personne identifie simplement et rapidement la structure de l'outil proposé. Dans le cas du robot Pepper, l'utilisateur peut facilement reconnaître l'utilité de chacun de ses membres, car ils s'apparentent à ceux d'un humain. Le robot doit également avoir une apparence androgyne. Il ne doit donc pas être associé à un genre, homme ou femme, en raison de sa voix ou de son physique. Pour se sentir en confiance, la structure du robot ne doit pas être imposante. De plus, dans la conception de ces robots, l'utilisation de tablette est très souvent privilégiée, car elle permet d'appuyer et d'améliorer l'interaction entre l'utilisateur et la machine. En effet, cette interface numérique donne au robot la possibilité d'afficher et de récupérer plus facilement des informations de l'utilisateur.

Les discussions avec les personnes âgées de l'EHPAD nous ont permis de comprendre qu'il faut définir une utilisation et des limites à ce type de système. Le robot doit donc remplir une fonction particulière, et donc être limité en action pour comprendre plus facilement ces intentions à chaque instant. Le besoin et le service doivent donc être bien définis au préalable pour éviter toute confusion, peur, refus, etc. Par exemple, la fonction d'un robot aspirateur est de se déplacer pour aspirer et ainsi nettoyer les sols de notre maison. Le robot assistant doit lui-même avoir des fonctions bien établies pour que l'utilisateur soit en totale confiance. Les fabricants de ce type de robot se concentrent le plus souvent sur la télé-assistance et la surveillance des personnes à leur domicile. Nos échanges avec les professionnels du soin et les personnes âgées ont permis d'identifier d'autres fonctionnalités intéressantes comme ramasser une canne tombée au sol, apporter des objets du quotidien (lunettes, médicaments, télécommande, ...), laver les zones non-accessibles par un robot aspirateur (toilettes, plan de travail, table, ...), vider ou remplir le lave-vaisselle, etc.

Les mouvements du robot doivent être fluides, naturels et sécurisés. La robustesse est également essentielle pour garantir son adaptation à n'importe quelle situation. Pour se déplacer dans des environnements humains, le robot humanoïde semble être le plus adapté. Cependant, ils ne sont pas suffisamment stables. C'est pourquoi, les robots assistants commercialisés à ce jour ont une base mobile et un corps plus ou moins apparenté à celui d'un humain (comme Pepper). Cette base leur permet une meilleure stabilité et garantit un déplacement plus sécurisé au sol.

Mettre en place un robot assistant peut avoir de nombreux avantages que ce soit au domicile des personnes âgées isolées ou en institution (EHPAD). En effet, il peut avoir des bénéfices pour le personnel soignant, pour le patient, mais aussi pour ses proches. Au domicile, les professionnels du soin peuvent obtenir des données leur permettant de rester informés et d'identifier précocement une fragilité ou une perte d'autonomie, évitant ainsi une hospitalisation ou une mise en institution. Pour le patient, le robot peut rendre des services, donner des informations, divertir, tenir compagnie, mais aussi le rassurer en lui proposant une assistance 24h sur 24h. Puis, pour ses proches, et plus particulièrement ses aidants, l'idée est de pouvoir améliorer la communication et de leur donner plus de temps pour eux tout en restant informés si nécessaire. En institution, le personnel soignant peut être soulagé de tâches répétitives, chronophages, leur permettant ainsi de mieux se focaliser sur leurs objectifs et leur rôle propre. Pour le résident, ce type d'outil peut aussi leur rendre des services et leur permettre d'appeler des proches. Ainsi, il n'est pas obligé d'attendre les professionnels souvent débordés pour obtenir des services simples.

Perspectives

Nous envisageons de poursuivre nos expérimentations afin d'obtenir plus de données sur ces transferts et de voir si le sentiment général ne tend pas à faire ressortir une autre zone. Nous souhaitons également améliorer notre framework en prenant en compte le champ de vision de la personne et ajouter des capteurs au niveau des mains du robot pour améliorer l'interaction physique lors de la prise de l'objet par l'humain. Il nous faut également améliorer l'estimation de la posture de la personne en trois dimensions et détecter quand une personne est allongée par exemple sur un lit pour donner l'objet correctement dans ce genre de situation. Nous souhaitons également permettre au robot de saisir des objets connus. Pour s'assurer que l'objet se positionne correctement dans la main et qu'il soit bien saisi par la main du robot, nous pourrions envisager d'utiliser des aimants. Cela nous permettrait d'inclure ce service au sein des projets déployés au domicile, ROBOd'Home, et en EHPAD.

Annexe A

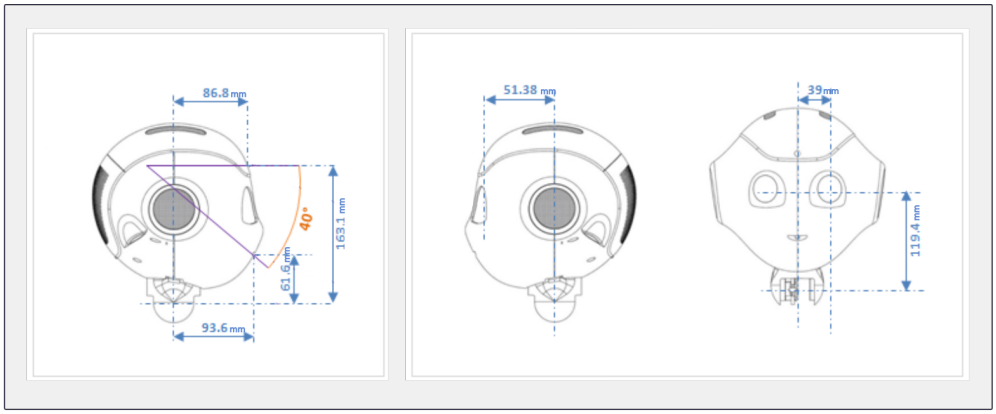
Description du robot Pepper

A.1 Description du robot Pepper

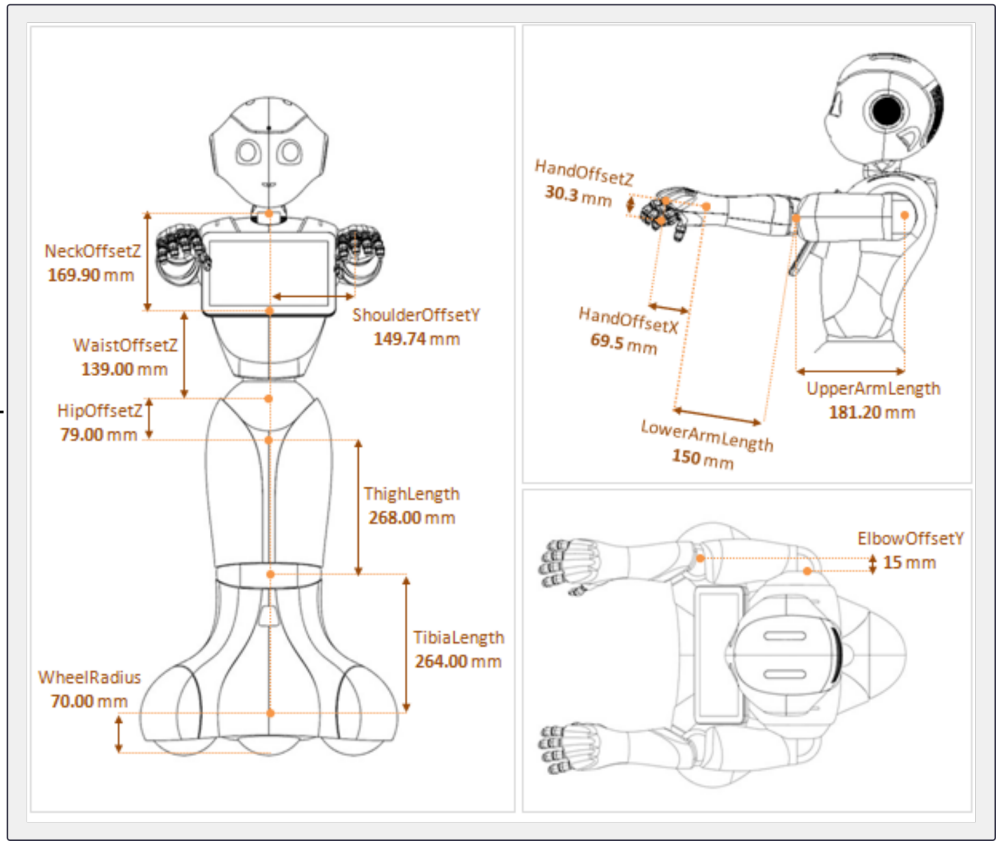
A.1.1 Informations utiles fournies par le constructeur

Pour les liaisons [1] [2] [3]

Tête

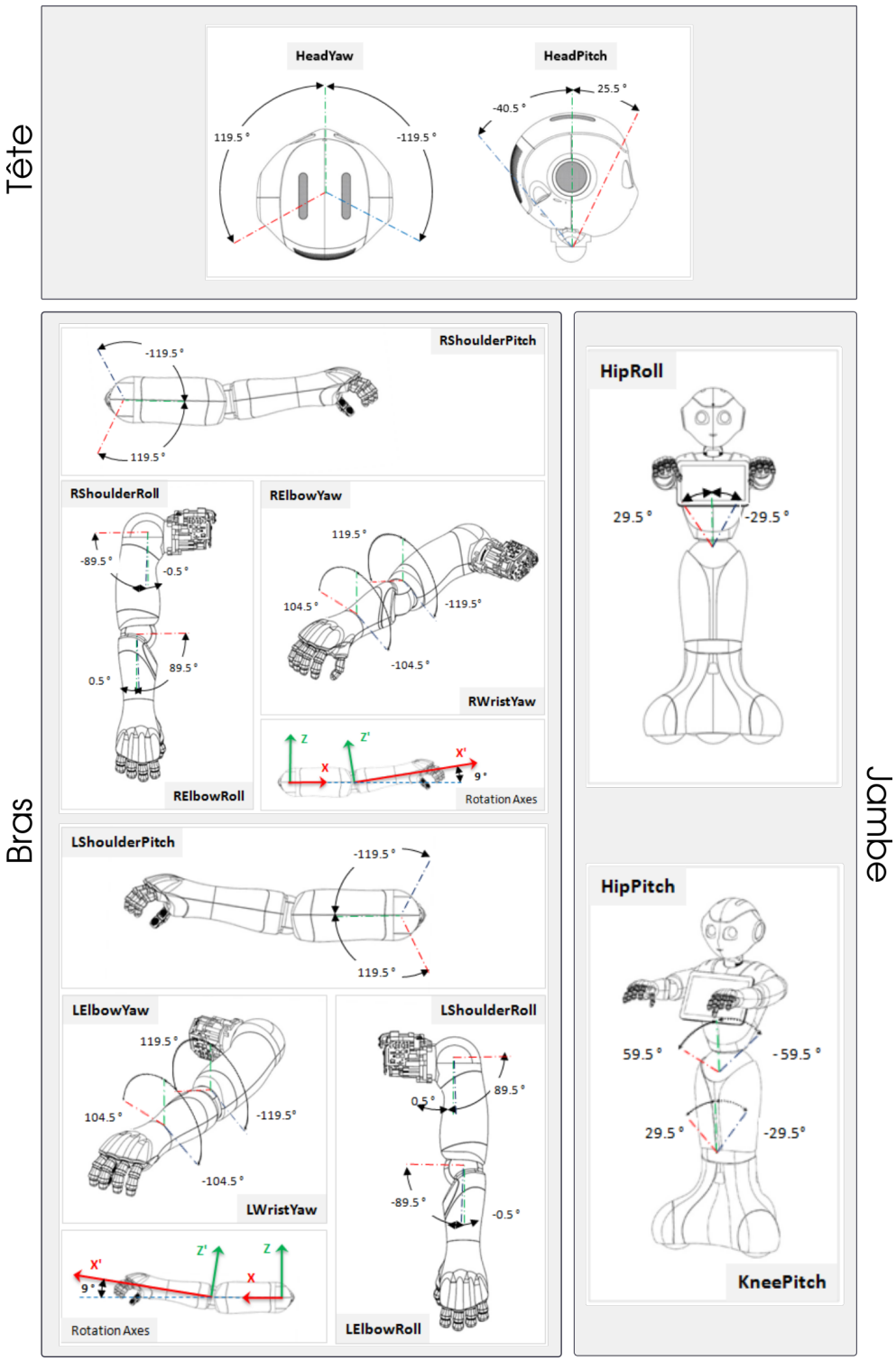


Corps



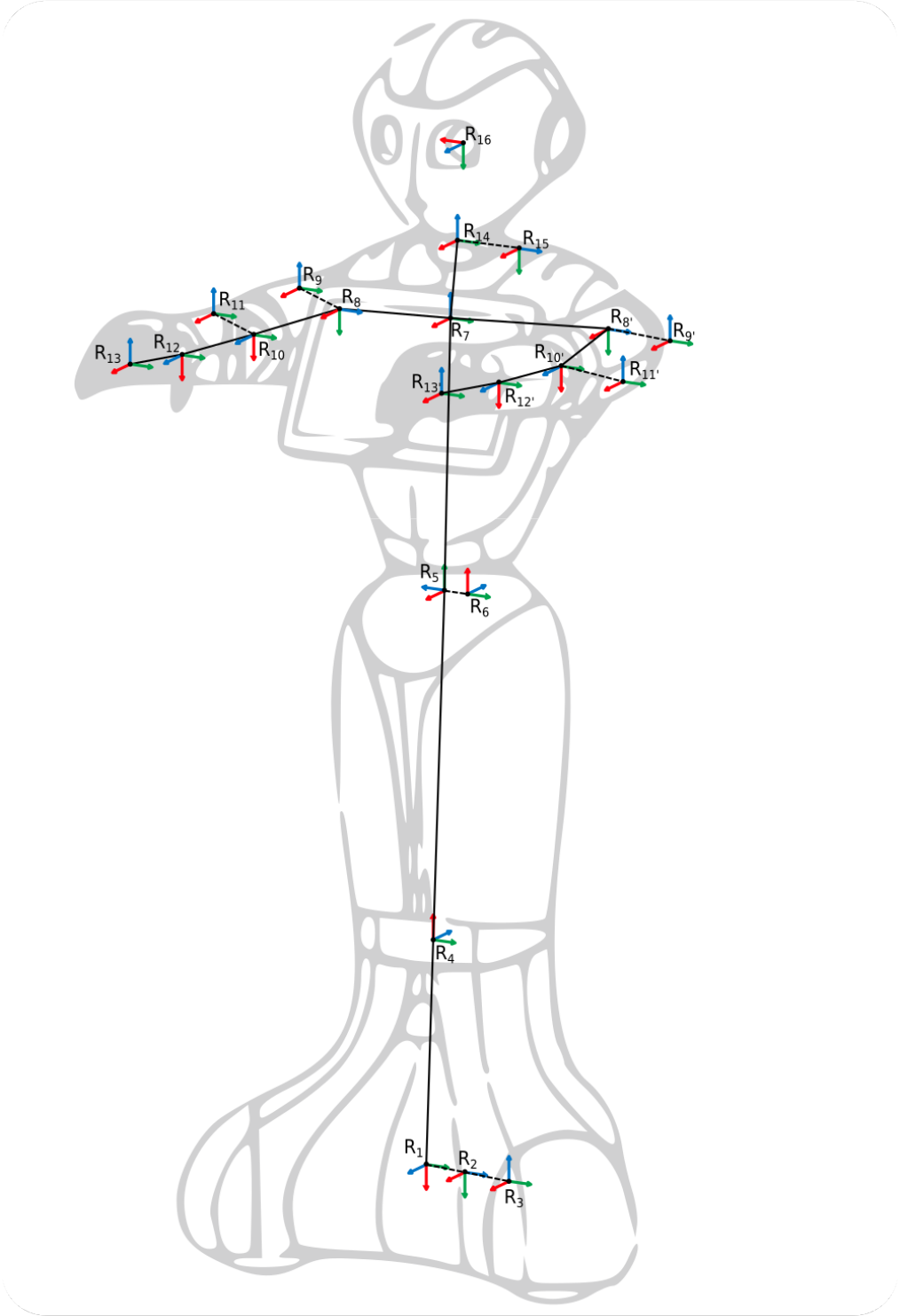
1. http://doc.aldebaran.com/2-5/family/pepper_technical/links_pep.html
2. http://doc.aldebaran.com/2-4/family/pepper_technical/video_pep.html#d-camera-pepper
3. http://doc.aldebaran.com/2-4/family/pepper_technical/video_3D_pep.html#d-camera-pepper

Pour les limites articulaires [4]



4. http://doc.aldebaran.com/2-5/family/pepper_technical/joints_pep.html

A.1.2 Modèle géométrique du robot Pepper

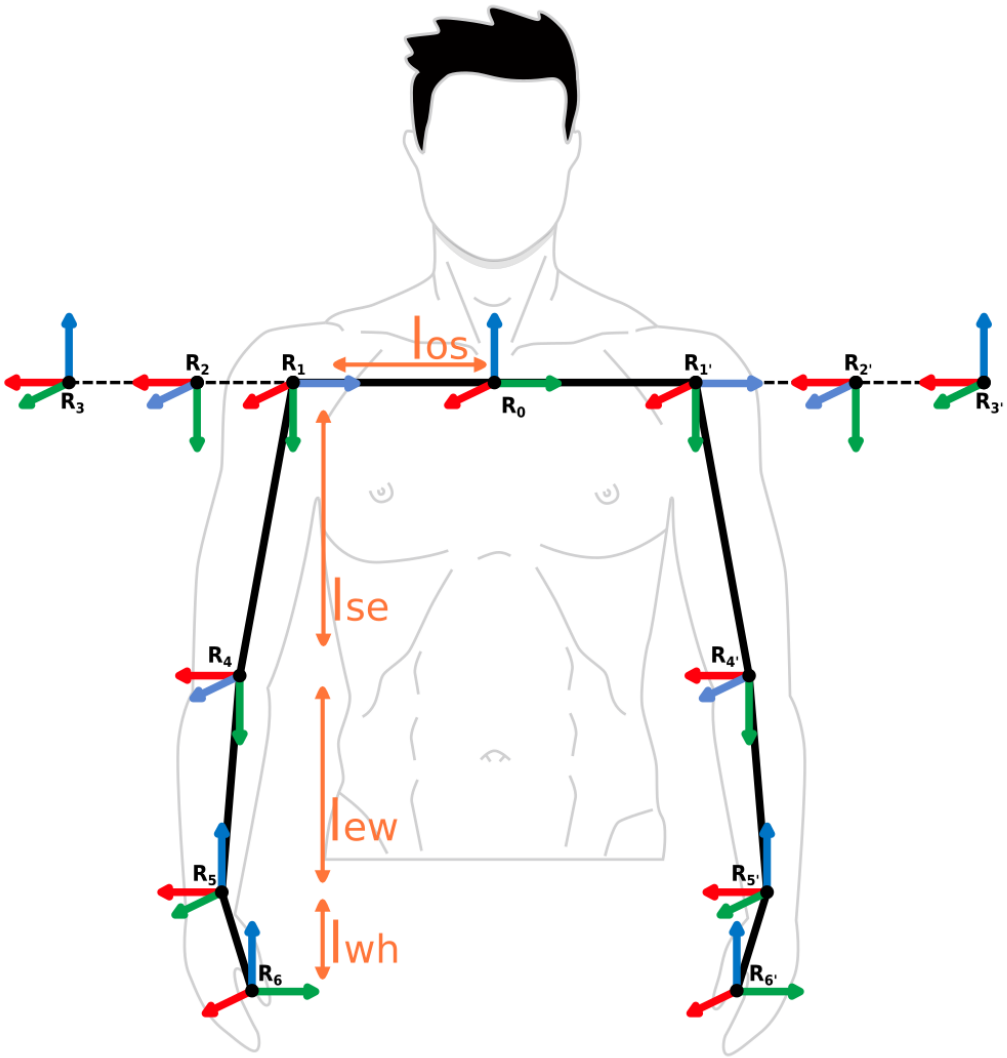


A.1.3 Tableau qui définit le modèle géométrique du robot Pepper

Nom du repère	x, y, z (m)	$\theta_x, \theta_y, \theta_z$ (rad)	θ_q (rad)	$[\theta_{min}, \theta_{max}]$ (rad)	$\dot{\theta}_{max}$ $(rad.s^{-1})$
$R_1 \rightarrow R_o$	0, 0, 0	$0, \frac{\pi}{2}, 0$	Q_1		
$R_2 \rightarrow R_1$	0, 0, 0	$0, -\frac{\pi}{2}, -\frac{\pi}{2}$	Q_2		
$R_3 \rightarrow R_2$	0, 0, 0	$\frac{\pi}{2}, 0, 0$	Q_3		
$R_4 \rightarrow R_3$	0, 0, 0.334	$0, -\frac{\pi}{2}, 0$	Q_4	[-0.515, 0.515]	640
$R_5 \rightarrow R_4$	0.268, 0, 0	$0, \frac{\pi}{2}, -\frac{\pi}{2}$	Q_5	[-1.039, 1.039]	640
$R_6 \rightarrow R_5$	0, 0.079, 0	$-\frac{\pi}{2}, 0, \frac{\pi}{2}$	Q_6	[-0.515, 0.515]	495
$R_7 \rightarrow R_6$	0.226, 0, 0.057	$0, \frac{\pi}{2}, 0$	Q_7		885
$R_8 \rightarrow R_7$	0, -0.150, 0	$-\frac{\pi}{2}, 0, 0$	Q_8	[-2.086, 2.086]	885
$R_9 \rightarrow R_8$	0, 0, 0	$\frac{\pi}{2}, 0, 0$	Q_9	$[-\frac{\pi}{2}$, -0.0087]	598
$R_{10} \rightarrow R_9$	0.181, -0.015, 0	$0, \frac{\pi}{2}, 0$	Q_{10}	[-2.086, 2.086]	
$R_{11} \rightarrow R_{10}$	0, 0, 0	$0, -\frac{\pi}{2}, 0$	Q_{11}	[0.0087, $\frac{\pi}{2}$]	885
$R_{12} \rightarrow R_{11}$	0.15, 0, 0	$0, \frac{\pi}{2}, 0$	Q_{12}	[-1.824, 1.824]	880
$R_{13} \rightarrow R_{12}$	-0.030, 0, 0.070	$0, -\frac{\pi}{2}, 0$	0		
$R_{8'} \rightarrow R_7$	0, 0.150, 0	$-\frac{\pi}{2}, 0, 0$	$Q_{8'}$	[-2.086, 2.086]	598
$R_{9'} \rightarrow R_{8'}$	0, 0, 0	$\frac{\pi}{2}, 0, 0$	$Q_{9'}$	[0.0087, $\frac{\pi}{2}$]	
$R_{10'} \rightarrow R_{9'}$	0.181, 0.015, 0	$0, \frac{\pi}{2}, 0$	$Q_{10'}$	[-2.086, 2.086]	
$R_{11'} \rightarrow R_{10'}$	0, 0, 0	$0, -\frac{\pi}{2}, 0$	$Q_{11'}$	$[-\frac{\pi}{2}$, -0.0087]	885
$R_{12'} \rightarrow R_{11'}$	0.15, 0, 0	$0, \frac{\pi}{2}, 0$	$Q_{12'}$	[-1.824, 1.824]	880
$R_{13'} \rightarrow R_{12'}$	0.030, 0, 0.070	$0, -\frac{\pi}{2}, 0$	0		598
$R_{14} \rightarrow R_7$	0.019, 0, 0.083	0, 0, 0	Q_{14}	[-2.086, 2.086]	885
$R_{15} \rightarrow R_{14}$	0, 0, 0	$-\frac{\pi}{2}, 0, 0$	Q_{15}	[-0.707, 0.637]	598
$R_{16} \rightarrow R_{15}$	0.051, -0.119, 0.039	$0, \frac{\pi}{2}, 0$	0		

A.2 Modèle géométrique de l'humain

A.2.1 Modèle géométrique des bras de l'humain



A.2.2 Tableau qui définit le modèle géométrique des bras de l'humain

$R_j \rightarrow R_i$	x (m)	y (m)	z (m)	θ_x (rad)	θ_y (rad)	θ_z (rad)	θ_q (rad)
$R_1 \rightarrow R_0$	0	-los	0	$-\frac{\pi}{2}$	0	0	Q_1
$R_2 \rightarrow R_1$	0	0	0	0	$\frac{\pi}{2}$	0	Q_2
$R_3 \rightarrow R_2$	0	0	0	$\frac{\pi}{2}$	0	0	Q_3
$R_4 \rightarrow R_3$	0	0	-lse	$-\frac{\pi}{2}$	0	0	Q_4
$R_5 \rightarrow R_4$	0	lew	0	$\frac{\pi}{2}$	0	0	0
$R_6 \rightarrow R_5$	0	0	-lwh	0	0	$\frac{\pi}{2}$	0
$R_{1'} \rightarrow R_0$	0	los	0	$-\frac{\pi}{2}$	0	0	$Q_{1'}$
$R_{2'} \rightarrow R_{1'}$	0	0	0	0	$\frac{\pi}{2}$	0	$Q_{2'}$
$R_{3'} \rightarrow R_{2'}$	0	0	0	$\frac{\pi}{2}$	0	0	$Q_{3'}$
$R_{4'} \rightarrow R_{3'}$	0	0	-lse	$-\frac{\pi}{2}$	0	0	$Q_{4'}$
$R_{5'} \rightarrow R_{4'}$	0	lew	0	$\frac{\pi}{2}$	0	0	0
$R_{6'} \rightarrow R_{5'}$	0	0	-lwh	0	0	$\frac{\pi}{2}$	0

A.2.3 Expression des transformations homogènes

Dans les équations suivantes, *c* représente le cosinus de l'angle précisé en indice et *s* le sinus de l'angle.

$$ {}^{\mathbf{i}}\mathbf{T}_{\mathbf{j}} = \overbrace{\begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}}^{\mathbf{T_{xyz}}} * \overbrace{\begin{pmatrix} c_{\theta_z} & -s_{\theta_z} & 0 & 0 \\ s_{\theta_z} & c_{\theta_z} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}^{\mathbf{T_{\theta_z}}} * \overbrace{\begin{pmatrix} c_{\theta_y} & 0 & s_{\theta_y} & 0 \\ 0 & 1 & 0 & 0 \\ -s_{\theta_y} & 0 & c_{\theta_y} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}^{\mathbf{T_{\theta_y}}} * \overbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\theta_x} & -s_{\theta_x} & 0 \\ 0 & s_{\theta_x} & c_{\theta_x} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}^{\mathbf{T_{\theta_x}}} $$

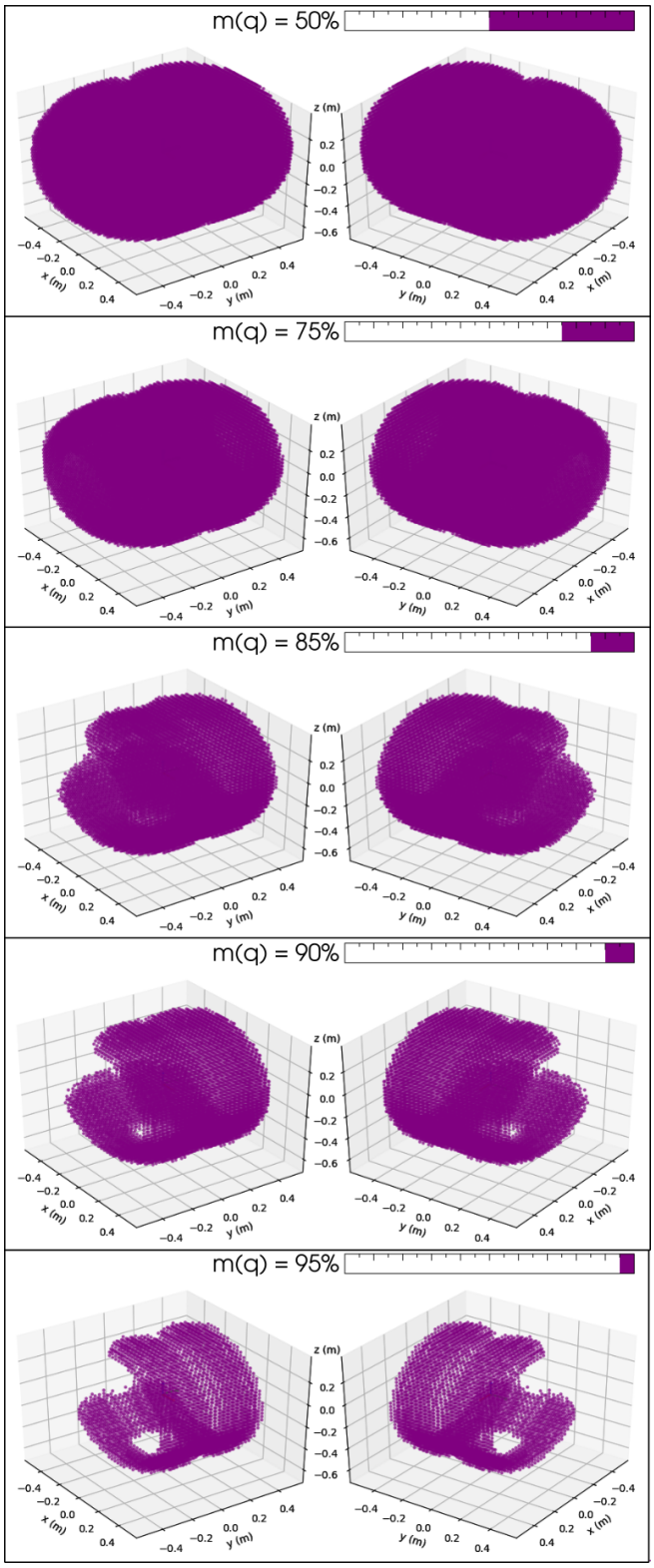
$$ {}^{\mathbf{i}}\mathbf{T}_{\mathbf{j}} = \begin{pmatrix} c_{\theta_z} * c_{\theta_y} & -s_{\theta_z} * c_{\theta_x} + c_{\theta_z} * s_{\theta_y} * s_{\theta_x} & s_{\theta_z} * s_{\theta_x} + c_{\theta_z} * s_{\theta_y} * c_{\theta_x} & x \\ s_{\theta_z} * c_{\theta_y} & c_{\theta_z} * c_{\theta_x} + s_{\theta_z} * s_{\theta_y} * s_{\theta_x} & -c_{\theta_z} * s_{\theta_x} + s_{\theta_z} * s_{\theta_y} * c_{\theta_x} & y \\ -s_{\theta_y} & c_{\theta_y} * s_{\theta_x} & c_{\theta_y} * c_{\theta_x} & z \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

$$ {}^{\mathbf{i}}\mathbf{T}_{\mathbf{j}}(\theta_{\mathbf{q}}) = \mathbf{T_{xyz}} * \mathbf{T}_{\theta_{\mathbf{z}}} * \mathbf{T}_{\theta_{\mathbf{y}}} * \mathbf{T}_{\theta_{\mathbf{x}}} * \begin{pmatrix} c_{\theta_q} & -s_{\theta_q} & 0 & 0 \\ s_{\theta_q} & c_{\theta_q} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

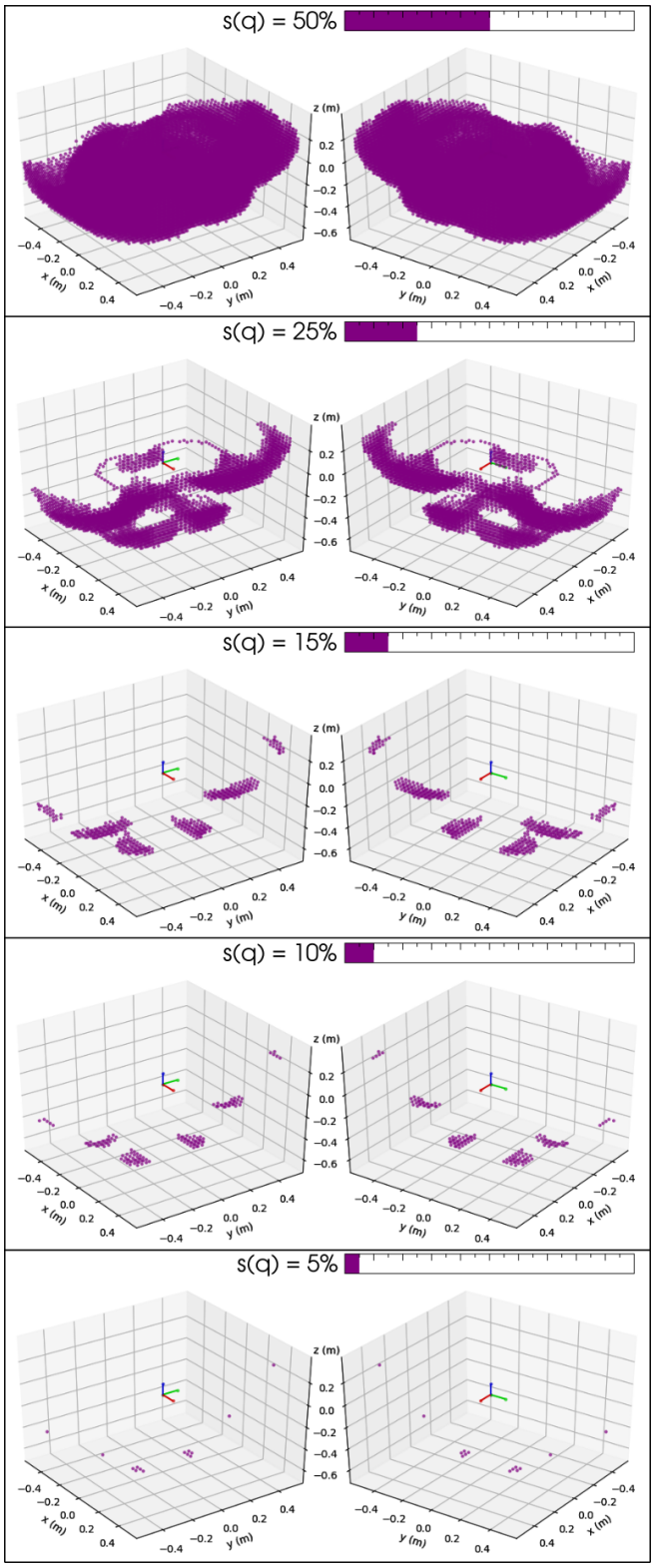
Annexe B

Différents aperçus

B.1 Application de l'indice de manipulabilité



B.2 Application de l'indice de gravité



Annexe C

Expérimentations

C.1 Première expérimentation

C.1.1 Lettre d'information

Lettre d'information

Présentation de l'étude :

Dans le cadre de nos recherches, nous travaillons sur l'assistance que peuvent apporter les robots humanoïdes tel que Pepper aux humains et plus particulièrement aux personnes âgées. Nous nous sommes intéressés aux interactions physiques comme le transfert d'un objet entre un robot et un humain.

Présentation de l'expérimentation :

L'objectif de cette expérimentation est d'estimer une zone de confort, autour de la personne se trouvant face au robot. Puis de lui transmettre un objet dans cette zone préalablement estimée.

Afin de personnaliser et estimer convenablement cette zone, vous devrez dans un premier temps remplir quelques informations comme votre taille, votre poids et dire si vous présentez des limitations articulaires anormales.

Pendant cette expérimentation, plusieurs formes de zones générées pourront être testées en faisant varier simplement deux paramètres. Ces deux paramètres sont les valeurs des pourcentages prises en compte afin de maximiser l'indice de manipulabilité et minimiser l'indice de gravité. Ils seront modifiés aléatoirement afin d'éviter un biais inconscient lors de l'expérimentation. Lors de cette expérimentation, nous vous demanderons de réaliser quatre transferts d'objet avec le robot et de juger votre satisfaction sur chacun d'entre eux. À la fin de ces quatre transferts, il vous sera demandé également de les classer par ordre de préférence (ou ex aequo).

Déroulement d'un transfert d'objet :

- Vous devez vous tenir debout à 1 ou 2 mètres environ du robot ;

- Vous ne devez pas bouger en attendant que le robot estime votre position, s'approche de vous et enfin vous tende l'objet ;

- Vous pourrez récupérer l'objet dans la main de Pepper lorsque ce dernier vous aura dit oralement de le récupérer ;

- Vous pourrez ensuite compléter le questionnaire concerné.

Notes importantes :

- Il vous sera demandé de ne pas prendre brusquement l'objet des mains de Pepper pour éviter toute détérioration des doigts (fragiles) du robot.

- L'interaction physique ici reste assez faible. Cependant il est possible qu'il y ai une erreur d'estimation par exemple sur votre position. Cela entrainerait une erreur sur la position finale à atteindre et donc la possibilité que le robot vous heurte. Néanmoins les vitesses du robot sont faibles et il sera possible pour le doctorant de stopper à tout moment le mouvement du robot. Il y a aussi un bouton d'arrêt d'urgence situé dans la nuque du robot.

C.1.2 Formulaire de consentement

Formulaire de consentement

Titre de l'expérimentation : "Analyse de la zone de confort générée pour un transfert d'objet d'un robot à un humain."

Promoteur : Clinique Beau Soleil - 119 avenue de Lodève, 34070 Montpellier (Tel : 04 67 75 97 84).

Je soussigné M./Mme. _____ (NOM Prenom)

né(e) le _____/_____/_____ à _____

Atteste que M. NOWAK Jordan m'a proposé de participer à l'expérimentation dans le cadre de ses recherches sur : "Analyse de la zone de confort générée pour un transfert d'objet d'un robot à un humain."

J'ai lu et compris la lettre d'information version 01 du 11/03/2021 que je conserve. L'investigateur, NOWAK Jordan, a répondu à toutes mes questions concernant l'étude. Je pourrai à tout moment lui demander des informations complémentaires sur cette étude réalisée (mail : nowak.jordan@outlook.fr).

J'ai eu le temps nécessaire pour réfléchir à mon implication dans cette étude et j'accepte librement et volontairement de participer. Cela n'engendrera aucun surcoût à ma charge. Je peux également à tout moment décider de quitter l'étude sans motiver ma décision.

J'accepte de répondre aux questionnaires qui me sont proposés durant cette expérimentation. J'ai compris que les données collectées à l'occasion de cette recherche seront protégées dans le respect de la confidentialité. Elles pourront uniquement être consultées par les professionnels appartenant à l'équipe de recherches. Je comprends également que ces données permettront une analyse statistique et pourront faire l'objet d'une publication scientifique.

Je certifie que je suis majeur(e) et que je ne bénéficie d'aucun régime de protection légale (sauvegarde de justice, tutelle, curatelle).

Mon consentement ne décharge pas le promoteur et l'investigateur de la recherche de leurs responsabilités. Je conserve tous les droits garantis par la loi.

Fait en deux exemplaires à _____ le _____/_____/_____.

NOM, Prénom de l'investigateur :	NOM, Prénom du participant :
Signature :	Signature :

Formulaire de consentement - 1er exemplaire
version 01 - 11/03/2021

C.1.3 Droit à l'image

Droit à l'image

Je soussigné _____ (NOM Prenom)

né(e) le _____/_____/_____ à _____ agissant en mon nom personnel autorise M. NOWAK Jordan à reproduire et exploiter mon image fixée dans le cadre de photographies et films pour l'expérimentation dans le cadre de ses recherches.

Cette autorisation emporte la possibilitée pour M. NOWAK Jordan d'apporter à la fixation initiale de mon image toutes modifications, adaptations ou suppressions qu'il jugera utile. Le Photographe pourra notamment l'utiliser, la publier, la reproduire, l'adapter ou la modifier, seule ou en combinaison avec d'autres matériels, par tous les moyens, méthodes ou techniques actuellement connues ou à venir.

Je garantis n'être lié(e) par aucun accord avec un tiers, de quelque nature que ce soit, ayant pour objet ou pour effet de limiter ou empêcher la mise en œuvre de la présente autorisation.

La présente autorisation d'exploitation de mon droit à l'image est consentie à titre gratuit.

Fait à _____, le _____/_____/_____,

Signature :

Droit à l'image - 1er exemplaire
version 01 - 11/03/2021

C.1.4 Questionnaire

Questionnaire

Informations :

Taille : Poids : Âge :



Anteflexion/Retroflexion

Gauche :

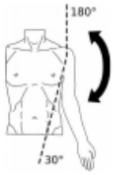
Droite :



Medial/Lateral

Gauche :

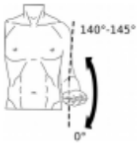
Droite :



Abduction/Adduction

Gauche :

Droite :



Flexion/Extension

Gauche :

Droite :

Questions avant l'expérimentation :

"Avez-vous des appréhensions vis à vis de l'expérimentation à réaliser ?"

Oui/Non :

Questions sur l'expérimentation :

Transfert 1. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 2. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 3. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 4. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Classement :

1 : 2 : 3 : 4 :

Questionnaire - version 01 - 11/03/2021 - page 1/2

Questions après l'expérimentation :

"Avez-vous été surpris par l'approche du robot concernant son mouvement ?"

☐ Très surpris ☐ Surpris ☐ Neutre ☐ Pas surpris ☐ Pas du tout surpris

"Globalement, comment jugeriez-vous la distanciation entre vous et le robot sur les positions de transfert ?"

☐ Très inacceptable ☐ Inacceptable ☐ Neutre

☐ Acceptable ☐ Très acceptable

"Que pensez-vous de la pertinence de l'étude ?"

☐ Pas du tout pertinente ☐ Pas pertinente ☐ Neutre

☐ Pertinente ☐ Tout à fait pertinente

C.1.5 Résultats

Réponses aux questionnaires sur les transferts

Personne n°	Âge	Taille	Poids	n°1	n°2	n°3	n°4	1er	2ème	3ème	4ème
1	23	146	43	4	4	5	5	3	4	2	1
2	62	173	76	4	4	4	5	4	3	2	1
3	71	170	80	4	4	3	5	4	1,2	3	-
4	30	173	64	2	3	4	4	3	4	2	1
5	28	162	57	4	4	4	4	4	3	2,1	-
6	42	182	85	4	4	5	5	3	4	2	1
7	25	155	60	5	4	5	4	3	1	4	2
8	27	180	90	1	4	4	2	3	2	4	1

Moyennes et écart types sur les zones

	Moyenne	Écart type
Âge	38.5	18.35
Zone n°1	3.5	1.31
Zone n°2	3.88	0.35
Zone n°3	4.25	0.71
Zone n°4	4.29	1.11

Avez-vous des appréhensions vis à vis de l'expérimentation à réaliser ?

Personne n°	1	2	3	4	5	6	7	8
	Non	Non	Non	Non	Non	Non	Non	Non

Avez-vous été surpris par l'approche du robot concernant son mouvement ?
("Très surpris"=1 à "Pas du tout surpris"=5)

Personne n°	1	2	3	4	5	6	7	8	Moyenne	Écart type
Note (/5)	4	4	5	2	5	2	4	4	3.75	1.16

Globalement, comment jugeriez-vous la distanciation entre vous et le robot ?
("Très inacceptable"=1 à "Très acceptable"=5)

Personne n°	1	2	3	4	5	6	7	8	Moyenne	Écart type
Note (/5)	5	4	4	4	4	4	4	4	4.13	0.35

Que pensez-vous de la pertinence de l'étude ?
("Pas du tout pertinente"=1 à "Tout à fait pertinente"=5)

Personne n°	1	2	3	4	5	6	7	8	Moyenne	Écart type
Note (/5)	5	5	5	5	5	4	5	4	4.75	0.46

C.2 Deuxième expérimentation

C.2.1 Lettre d'information

Lettre d'information

Présentation de l'étude :

Dans le cadre de nos recherches, nous travaillons sur l'assistance que peuvent apporter les robots humanoïdes tel que Pepper aux humains et plus particulièrement aux personnes âgées. Nous nous sommes intéressés aux interactions physiques comme le transfert d'un objet entre robot et humain.

Présentation de l'expérimentation :

L'objectif de cette expérimentation est d'estimer une zone de confort, autour de la personne se trouvant face au robot. Puis de lui transmettre un objet dans cette zone préalablement estimée.

Afin de personnaliser et estimer convenablement cette zone, vous devrez dans un premier temps remplir quelques informations comme votre taille, votre poids et dire si vous présentez des limitations articulaires anormales.

Pendant cette expérimentation, plusieurs formes de zones pourront être testées en faisant varier simplement deux paramètres. Ces deux paramètres permettent de maximiser l'indice de manipulabilité et minimiser l'indice de gravité. Ils seront modifiés aléatoirement afin d'éviter un biais inconscient lors de l'expérimentation. Nous vous demanderons de réaliser un total de douze transferts avec le robot et de juger votre satisfaction sur chacun d'entre eux. En effet, nous générons trois zones de confort différentes et pour la répétabilité de l'expérience, nous vous faisons faire quatre transfert par zone.

À la fin de ces douze transferts, on prendra celle qui vous a paru la plus satisfaisante selon vos réponses. Enfin, on fera une commande plus générale du robot avec déplacement de la porte d'entrée au lit de la chambre , avec transfert d'un objet avant de repartir en direction de la sortie de la chambre. Cette démonstration sera faite quatre fois afin de valider sa robustesse.

Déroulement d'un transfert d'objet :

Vous devez vous tenir assis sur le lit à 1 ou 2 mètres environ du robot. Vous ne devez pas bouger en attendant que le robot estime votre position, s'approche de vous et enfin vous tende l'objet. Vous pourrez prendre l'objet dans la main du robot lorsqu'il vous aura dit oralement de le récupérer. Enfin, vous pourrez compléter le questionnaire concerné.

Notes importantes :

- Il vous sera demandé de ne pas prendre brusquement l'objet des mains de Pepper pour éviter toute détérioration des doigts (fragiles) du robot.

- Il est possible qu'il y ait une erreur d'estimation sur votre position. Cela entrainerait une erreur sur la position finale à atteindre et donc la possibilité que le robot vous touche. Les vitesses du robot sont faibles et il sera possible de stopper à tout moment son mouvement en touchant simplement sa tablette, ses bumpers, ou en activant le bouton d'arrêt d'urgence situé dans sa nuque.

Lettre d'information - version 02 - 18/08/2021

C.2.2 Formulaire de consentement

Formulaire de consentement

Titre de l'expérimentation : "Analyse de la zone de confort générée pour un transfert d'objet d'un robot à un humain."

Promoteur : Clinique Beau Soleil - 119 avenue de Lodève, 34070 Montpellier (Tel : 04 67 75 97 84).

Je soussigné M./Mme. _____ (NOM Prenom)

né(e) le _____/_____/_____ à _____

Atteste que M. NOWAK Jordan m'a proposé de participer à l'expérimentation dans le cadre de ses recherches sur : "Analyse de la zone de confort générée pour un transfert d'objet d'un robot à un humain."

J'ai lu et compris la lettre d'information version 02 du 18/08/2021 que je conserve. L'investigateur, NOWAK Jordan, a répondu à toutes mes questions concernant l'étude. Je pourrai à tout moment lui demander des informations complémentaires sur cette étude réalisée (mail : nowak.jordan@outlook.fr).

J'ai eu le temps nécessaire pour réfléchir à mon implication dans cette étude et j'accepte librement et volontairement de participer. Cela n'engendrera aucun surcoût à ma charge. Je peux également à tout moment décider de quitter l'étude sans motiver ma décision.

J'accepte de répondre aux questionnaires qui me sont proposés durant cette expérimentation. J'ai compris que les données collectées à l'occasion de cette recherche seront protégées dans le respect de la confidentialité. Elles pourront uniquement être consultées par les professionnels appartenant à l'équipe de recherches. Je comprends également que ces données permettront une analyse statistique et pourront faire l'objet d'une publication scientifique.

Je certifie que je suis majeur(e) et que je ne bénéficie d'aucun régime de protection légale (sauvegarde de justice, tutelle, curatelle).

Mon consentement ne décharge pas le promoteur et l'investigateur de la recherche de leurs responsabilités. Je conserve tous les droits garantis par la loi.

Fait en deux exemplaires à _____ le _____/_____/_____.

NOM, Prénom de l'investigateur :	NOM, Prénom du participant :
Signature :	Signature :

Formulaire de consentement - 1er exemplaire
version 02 - 18/08/2021

C.2.3 Droit à l'image

Droit à l'image

Je soussigné _____ (NOM Prenom)

né(e) le _____/_____/_____ à _____ agissant en mon nom personnel autorise M. NOWAK Jordan à reproduire et exploiter mon image fixée dans le cadre de photographies et films pour l'expérimentation dans le cadre de ses recherches.

Cette autorisation emporte la possibilitée pour M. NOWAK Jordan d'apporter à la fixation initiale de mon image toutes modifications, adaptations ou suppressions qu'il jugera utile. Le Photographe pourra notamment l'utiliser, la publier, la reproduire, l'adapter ou la modifier, seule ou en combinaison avec d'autres matériels, par tous les moyens, méthodes ou techniques actuellement connues ou à venir.

Je garantis n'être lié(e) par aucun accord avec un tiers, de quelque nature que ce soit, ayant pour objet ou pour effet de limiter ou empêcher la mise en œuvre de la présente autorisation.

La présente autorisation d'exploitation de mon droit à l'image est consentie à titre gratuit.

Fait à _____, le _____/_____/_____,

Signature :

Droit à l'image - 1er exemplaire
version 02 - 18/08/2021

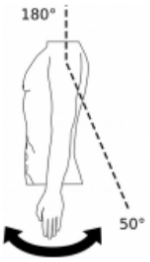
C.2.4 Questionnaire

Questionnaire

Informations :

Taille : Poids : Âge :

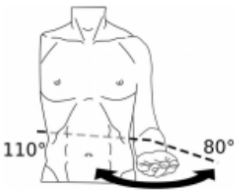
Anteflexion/Retroflexion



Droite :

Gauche :

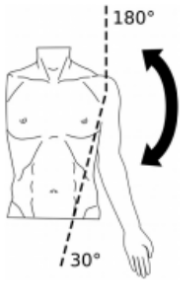
Medial/Lateral



Droite :

Gauche :

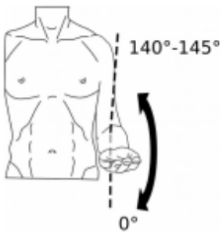
Abduction/Adduction



Droite :

Gauche :

Flexion/Extension



Droite :

Gauche :

Questions avant l'expérimentation :

"Avez-vous des appréhensions vis à vis de l'expérimentation à réaliser ?"

Oui/Non :

Questions sur l'expérimentation – transfert d'objet :

Transfert 1. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 2. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 3. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 4. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 5. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 6. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 7. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 8. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 9. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 10. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 11. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Transfert 12. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐Très insatisfait ☐Insatisfait ☐Neutre ☐Satisfait ☐Très satisfait

Classement : 1 : 2 : 3 :

Questions sur l'expérimentation – navigation et transfert d'objet :

Transfert 13. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐ Très insatisfait ☐ Insatisfait ☐ Neutre ☐ Satisfait ☐ Très satisfait

Transfert 14. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐ Très insatisfait ☐ Insatisfait ☐ Neutre ☐ Satisfait ☐ Très satisfait

Transfert 15. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐ Très insatisfait ☐ Insatisfait ☐ Neutre ☐ Satisfait ☐ Très satisfait

Transfert 16. "Globalement, êtes-vous satisfait du transfert effectué ?"

☐ Très insatisfait ☐ Insatisfait ☐ Neutre ☐ Satisfait ☐ Très satisfait

Questions après l'expérimentation :

"Avez-vous été surpris par l'approche du robot concernant son mouvement ?"

☐ Très surpris ☐ Surpris ☐ Neutre ☐ Pas surpris ☐ Pas du tout surpris

"Globalement, comment jugeriez-vous la distanciation entre vous et le robot ?"

☐ Très inacceptable ☐ Inacceptable ☐ Neutre

☐ Acceptable ☐ Très acceptable

"Que pensez-vous de la pertinence de l'étude ?"

☐ Pas du tout pertinente ☐ Pas pertinente ☐ Neutre

☐ Pertinente ☐ Tout à fait pertinente

C.2.5 Résultats

Réponses aux questionnaires sur les transferts

Pers.	1	2	3	4	5	6	7	8	9	10	11	12
Âge	95	96	89	89	86	99	89	21	42	31	27	62
Taille	170	162	154	163	184	164	165	169	182	175	180	173
Poids	68	54	64	53	86	52	44	78	80	65	90	79
tf.1	4	4	4	4	4	3	4	5	4	4	3	5
tf.2	4	4	4	3	4	4	4	4	5	4	4	4
tf.3	4	4	2	4	3	4	4	4	5	4	3	4
tf.4	3	4	4	2	3	4	3	4	5	2	2	4
tf.5	3	4	4	4	4	5	4	3	3	4	2	5
tf.6	3	4	4	4	3	2	4	3	4	3	2	4
tf.7	3	4	4	3	2	3	3	4	4	3	3	4
tf.8	4	4	4	2	4	5	5	4	4	4	4	5
tf.9	4	4	3	1	4	4	3	4	4	4	1	3
tf.10	3	4	4	3	3	4	2	3	3	5	1	3
tf.11	3	4	3	4	4	3	5	3	5	4	1	4
tf.12	2	4	3	2	5	5	4	5	5	5	3	5
Zone 1	3.25 ± 0.96	4.00 ± 0.00	3.75 ± 0.50	3.00 ± 1.15	4.25 ± 0.50	4.50 ± 1.00	4.25 ± 0.50	4.25 ± 0.96	4.00 ± 0.82	4.25 ± 0.50	3.00 ± 0.82	5.00 ± 0.00
Zone 2	3.50 ± 0.58	4.00 ± 0.00	3.25 ± 0.96	3.75 ± 0.50	3.50 ± 0.58	3.25 ± 0.96	4.25 ± 0.50	3.75 ± 0.50	4.00 ± 0.82	3.50 ± 1.29	1.75 ± 0.96	3.50 ± 0.58
Zone 3	3.25 ± 0.50	4.00 ± 0.00	3.75 ± 0.50	2.25 ± 0.96	3.00 ± 0.82	3.75 ± 0.50	2.75 ± 0.50	3.50 ± 0.58	4.75 ± 0.50	3.75 ± 0.50	2.50 ± 1.29	4.00 ± 0.00

Moyennes et écart types sur les zones, pour les 7 résidents

	Moyenne	Écart type
Âge	91.86	4.78
Zone n°1	3.857	0.848
Zone n°2	3.643	0.678
Zone n°3	3.25	0.799

Moyennes et écart types sur les zones, pour tous les volontaires

	Moyenne	Écart type
Âge	68.83	30.28
Zone n°1	3.96	0.87
Zone n°2	3.67	0.81
Zone n°3	3.27	0.94

Avez-vous des appréhensions vis à vis de l'expérimentation à réaliser ?

Pers.	1	2	3	4	5	6	7	8	9	10	11	12
	Non	Oui	Non	Non	Non	Non	Non	Non	Non	Non	Non	Non

Avez-vous été surpris par l'approche du robot concernant son mouvement ?
("Très surpris"=1 à "Pas du tout surpris"=5)

Pers.	1	2	3	4	5	6	7	8	9	10	11	12	Moyenne	Ecart type
Note (/5)	5	5	4	5	5	5	5	5	4	4	4	4	4.58	0.51

Globalement, comment jugeriez-vous la distanciation entre vous et le robot ?
("Très inacceptable"=1 à "Très acceptable"=5)

Pers.	1	2	3	4	5	6	7	8	9	10	11	12	Moyenne	Ecart type
Note (/5)	5	5	4	5	4	3	4	5	5	4	5	5	4.50	0.67

Que pensez-vous de la pertinence de l'étude ?
("Pas du tout pertinente"=1 à "Tout à fait pertinente"=5)

Pers.	1	2	3	4	5	6	7	8	9	10	11	12	Moyenne	Ecart type
Note (/5)	3	4	3	3	4	3	3	5	4	5	4	5	3.83	0.83

Bibliographie

[1] D. DESRIVIERRE, « D'ici 2050, la population augmenterait dans toutes les régions de métropole », 2017 (cf. p. 16).

[2] M. MARTIN et E. RIGNOLS, « Tableaux de l'économie française », *Collection INSEE Références Édition*, p. 24–25, 2020 (cf. p. 16).

[3] A. CARRÈRE, « Prise en charge de la perte d'autonomie des personnes âgées: Une analyse des déterminants de l'institutionnalisation ou du maintien à domicile », thèse de doct., Université Paris sciences et lettres, 2020, p. 1–3 (cf. p. 16).

[4] D. BLANCHET et F. LE GALLO, « Baby-boom et allongement de la durée de vie: Quelles contributions au vieillissement? », 2013 (cf. p. 16).

[5] S. PAPON, « La mortalité infantile est stable depuis dix ans après des décennies de baisse », 2018 (cf. p. 17).

[6] ——, « Les gains d'espérance de vie se concentrent désormais aux âges élevés », 2019 (cf. p. 17).

[7] F. MESLÉ, « Progrès récents de l'espérance de vie en france », *Population*, t. 61, n° 4, p. 437–462, 2006 (cf. p. 17).

[8] ORGANISATION MONDIALE DE LA SANTÉ (OMS), *Vieillissement et santé. [En ligne]*, Disponible : https://www.who.int/fr/news-room/fact-sheets/detail/ageing-and-health, (Oct. 2021) (cf. p. 17).

[9] E. CAMBOIS et J.-M. ROBINE, « Vieillissement et restrictions d'activité: L'enjeu de la compensation des problèmes fonctionnels », 2003 (cf. p. 17).

[10] H. A. de SANTÉ, « Evaluation et prise en charge des personnes âgées faisant des chutes répétées », *Recommandations. Avril*, p. 2009–06, 2009 (cf. p. 17).

[11] M. TINETTI, J. DOUCETTE, E. CLAUS et R. MAROTTOLI, « Risk factors for serious injury during falls by older persons in the community », *Journal of the American geriatrics society*, t. 43, n° 11, p. 1214–1221, 1995 (cf. p. 17).

[12] P. STALENHOEF, J. DIEDERIKS, L. DE WITTE, K. SCHIRICKE et H. FJM CREBOLDER, « Impact of gait problems and falls on functioning in independent living persons of 55 years and over: A community survey », *Patient education and counseling*, t. 36, n° 1, p. 23–31, 1999 (cf. p. 17).

[13] E. GREENE, C. CUNNINGHAM, A. EUSTACE, N. KIDD, A. CLARE et B. LAWLOR, « Recurrent falls are associated with increased length of stay in elderly psychiatric inpatients », *Int. journal of geriatric psychiatry*, t. 16, n° 10, p. 965–968, 2001 (cf. p. 17).

[14] J. DAVISON, J. BOND, P. DAWSON, I. STEEN et R. KENNY, « Patients with recurrent falls attending accident & Emergency benefit from multifactorial intervention—a randomised controlled trial », *Age and ageing*, t. 34, n° 2, p. 162–168, 2005 (cf. p. 17).

[15] H. MICHEL, « La notion de fragilité des personnes âgées: Apports, limites et enjeux d'une démarche préventive », *Retraite et société*, n° 1, p. 174–181, 2012 (cf. p. 18).

[16] P. LEBEL, N. LEDUC, M. KERGOAT, J. LATOUR et C. LECLERC, « Un modèle dynamique de la fragilité », *L'Année gérontologique (Ed. française)*, t. 13, p. 84–94, 1999 (cf. p. 18).

[17] I. ARVEUX, G. FAIVRE, L. LENFANT, P. MANCKOUNDIA et F. MOUREY, « Le sujet âgé fragile », *La revue de gériatrie*, t. 27, n° 7, p. 569–581, 2002 (cf. p. 18).

[18] INTERNATIONAL FEDERATION OF ROBOTICS (IFR), *Service Robots Record: Sales Worldwide Up 32%. [En ligne]*, Disponible : https://ifr.org/ifr-press-releases/news/service-robots-record-sales-worldwide-up-32, (Communiqué de presse (28 Oct. 2020) (cf. p. 19).

[19] ROBOSOFT ENGINEERING, *We develop the robot you need. [En ligne]*, Disponible : https://robosoft.com/, (2020) (cf. p. 20).

[20] KOMPAÏ ROBOTICS, *Kompaï, to help frail people and their caregivers. [En ligne]*, Disponible : https://kompai.com, (2020) (cf. p. 20).

[21] BLUE FROG ROBOTICS & BUDDY, *Buddy le premier robot compagnon émotionnel. [En ligne]*, https://buddytherobot.com/fr/buddy-le-robot-emotionnel-famille/, (2021) (cf. p. 20).

[22] BLUE FROG ROBOTICS & BUDDY, *Les robots compagnons, comme notre BUDDY, sont une solution idéale pour les seniors qui souhaitent rester plus longtemps chez eux ! [En ligne]*, Disponible : https://buddytherobot.com/fr/buddy-robot-seniors-isoles-personnes-ages-seules/, (2021) (cf. p. 20).

[23] *Cutii, une présence qui vous fait du bien. [En ligne]*, Disponible : https://www.cutii.io/ (cf. p. 21).

[24] CUTII, *Rejoignez la communauté des animateurs. [En ligne]*, Disponible : https://www.cutii.io/devenir-animateur/ (cf. p. 21).

[25] EUROPEAN UNION, « Special Eurobarometer 427 "Autonomous Systems". », 2015, Conducted by TNS Opinion and Social at the request of Directorate-General for Communications Networks, Content and Technology (DG CONNECT). United Nations Brussels, Belgium (cf. p. 21).

[26] INTERNATIONAL FEDERATION OF ROBOTICS (IFR), *La robotisation s'intensifie au niveau mondial. [En ligne]*, Disponible : https://ifr.org/img/uploads/2018-FEB-07_IFR_press_release_-_Robot_density-French.pdf, Communiqué de presse (2018) (cf. p. 21).

[27] R. LIKERT, « A technique for the measurement of attitudes », *Archives of psychology*, 1932 (cf. p. 22).

[28] V. ORTENZI, A. COSGUN, T. PARDI, W. CHAN, E. CROFT et D. KULIĆ, « Object handovers: A review for robotics », *IEEE Transactions on Robotics*, 2021 (cf. p. 30).

[29] A. MASON et C. MACKENZIE, « Grip forces when passing an object to a partner », *Experimental brain research*, t. 163, n° 2, p. 173–187, 2005 (cf. p. 30, 37).

[30] C. MACKENZIE et T. IBERALL, *The grasping hand*. Elsevier, 1994 (cf. p. 30).

[31] C. VESPER, S. BUTTERFILL, G. KNOBLICH et N. SEBANZ, « A minimal architecture for joint action », *Neural Networks*, t. 23, n° 8-9, p. 998–1003, 2010 (cf. p. 31).

[32] K. SAKI, Y. NATSUKI, V. GENTIANE, Y. EIICHI et G. GOWRISHANKAR, « The where of handovers by humans: Effect of partner characteristics, distance and visual feedback », *PloS one*, t. 14, n° 6, 2019 (cf. p. 31).

[33] K. STRABALA, M. LEE, A. DRAGAN, J. FORLIZZI, S. SRINIVASA, M. CAKMAK et V. MICELLI, « Toward seamless human-robot handovers », *Journal of Human-Robot Interaction*, t. 2, n° 1, p. 112–132, 2013 (cf. p. 31).

[34] U. CASTIELLO, « Understanding other people's actions: Intention and attention », *Journal of Experimental Psychology: Human Perception and Performance*, t. 29, n° 2, p. 416, 2003 (cf. p. 31).

[35] J.-D. BOUCHER, U. PATTACINI, A. LELONG, G. BAILLY, F. ELISEI, S. FAGEL, P. DOMINEY et J. VENTRE-DOMINEY, « I reach faster when I see you look: Gaze effects in human–human and human–robot face-to-face cooperation », *Frontiers in neurorobotics*, t. 6, p. 3, 2012 (cf. p. 31).

[36] M. STAUDTE et M. CROCKER, « The utility of gaze in spoken human-robot interaction », in *Proceedings of Workshop on Metrics for Human-Robot Interaction 2008, March 12th*, 2008, p. 53–59 (cf. p. 31, 44).

[37] A. MOON, D. TRONIAK, B. GLEESON, M. PAN, M. ZHENG, B. BLUMER, K. MACLEAN et E. CROFT, « Meet me where i'm gazing: How shared attention gaze affects human-robot handover timing », in *Proceedings of the 2014 ACM/IEEE Int. Conf. on Human-robot interaction*, 2014, p. 334–341 (cf. p. 31, 44).

[38] M. GHARBI, P.-V. PAUBEL, A. CLODIC, O. CARRERAS, R. ALAMI et J.-M. CELLIER, « Toward a better understanding of the communication cues involved in a human-robot object transfer », in *2015 24th IEEE Int. symposium on robot and human interactive communication (RO-MAN)*, IEEE, 2015, p. 319–324 (cf. p. 31, 44).

[39] M. CAKMAK, S. SRINIVASA, M. LEE, S. KIESLER et J. FORLIZZI, « Using spatial and temporal contrast for fluent robot-human hand-overs », in *Proceedings of the 6th Int. Conf. on Human-robot interaction*, 2011, p. 489–496 (cf. p. 31).

[40] D. GONZALEZ, B. STUDENKA, C. GLAZEBROOK et J. LYONS, « Extending end-state comfort effect: Do we consider the beginning state comfort of another? », *Acta psychologica*, t. 136, n° 3, p. 347–353, 2011 (cf. p. 32).

[41] R. MEYER M. Van Der Wel et S. HUNNIUS, « Higher-order action planning for individual and joint object manipulations », *Experimental brain research*, t. 225, n° 4, p. 579–588, 2013 (cf. p. 32).

[42] F. CINI, V. ORTENZI, P. CORKE et M. CONTROZZI, « On the choice of grasp type and location when handing over an object », *Science Robotics*, t. 4, n° 27, 2019 (cf. p. 32).

[43] W. CHAN, K. NAGAHAMA, H. YAGUCHI, Y. KAKIUCHI, K. OKADA et M. INABA, « Implementation of a framework for learning handover grasp configurations through observation during human-robot object handovers », in *2015 IEEE-RAS 15th Int. Conf. on Humanoid Robots (Humanoids)*, IEEE, 2015, p. 1115–1120 (cf. p. 32).

[44] M. IIZUKA et M. HASHIMOTO, « Detection of Semantic Grasping-Parameter using Part-Affordance Recognition », in *2018 19th Int. Conf. on Research and Education in Mechatronics (REM)*, IEEE, 2018, p. 136–140 (cf. p. 32).

[45] W. CHAN, M. PAN, E. CROFT et M. INABA, « Characterization of handover orientations used by humans for efficient robot to human handovers », in *2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015, p. 1–6 (cf. p. 32).

[46] L. SARTORI, E. STRAULINO et U. CASTIELLO, « How objects are grasped: The interplay between affordances and end-goals », *PloS one*, t. 6, n° 9, e25203, 2011 (cf. p. 32).

[47] I. LENZ, H. LEE et A. SAXENA, « Deep learning for detecting robotic grasps », *The Int. Journal of Robotics Research*, t. 34, n° 4-5, p. 705–724, 2015 (cf. p. 32).

[48] W. YANG, C. PAXTON, M. CAKMAK et D. FOX, « Human grasp classification for reactive human-to-robot handovers », in *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2020, p. 11 123–11 130 (cf. p. 32, 36).

[49] P. ROSENBERGER, A. COSGUN, R. NEWBURY, J. KWAN, V. ORTENZI, P. CORKE et M. GRAFINGER, « Object-independent human-to-robot handovers using real time robotic vision », *IEEE Robotics and Automation Letters*, t. 6, n° 1, p. 17–23, 2020 (cf. p. 32, 36).

[50] V. ORTENZI, F. CINI, T. PARDI, N. MARTURI, R. STOLKIN, P. CORKE et M. CONTROZZI, « The grasp strategy of a robot passer influences performance and quality of the robot-human object handover », *Frontiers in Robotics and AI*, t. 7, 2020 (cf. p. 32).

[51] E. LOPEZ-DAMIAN, D. SIDOBRE, S. DELATOUR et R. ALAMI, « Grasp planning for interactive object manipulation », in *Proc. of the Intl. Symp. on Robotics and Automation*, Citeseer, 2006, p. 1–6 (cf. p. 32).

[52] J. NAPIER, « The prehensile movements of the human hand », *The Journal of bone and joint surgery. British volume*, t. 38, n° 4, p. 902–913, 1956 (cf. p. 32).

[53] T. FEIX, I. BULLOCK et A. DOLLAR, « Analysis of human grasping behavior: Correlating tasks, objects and grasps », *IEEE transactions on haptics*, t. 7, n° 4, p. 430–441, 2014 (cf. p. 32).

[54] A. QUISPE, E. MARTINSON et K. OGUCHI, « Learning user preferences for robot-human handovers », in *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2017, p. 834–839 (cf. p. 32).

[55] C. HANSEN, P. ARAMBEL, K. BEN MANSOUR, V. PERDEREAU et F. MARIN, « Human–human handover tasks and how distance and object mass matter », *Perceptual and motor skills*, t. 124, n° 1, p. 182–199, 2017 (cf. p. 33).

[56] N. VAHRENKAMP, H. ARNST, M. WÄCHTER, D. SCHIEBENER, P. SOTIROPOULOS, M. KOWALIK et T. ASFOUR, « Workspace analysis for planning human-robot interaction tasks », in *2016 IEEE-RAS 16th Int. Conf. on Humanoid Robots (Humanoids)*, IEEE, 2016, p. 1298–1303 (cf. p. 33, 38, 64).

[57] F. ZACHARIAS, C. BORST, M. BEETZ et G. HIRZINGER, « Positioning mobile manipulators to perform constrained linear trajectories », in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, IEEE, 2008, p. 2578–2584 (cf. p. 33).

[58] E. SISBOT, L. MARIN-URIAS, X. BROQUERE, D. SIDOBRE et R. ALAMI, « Synthesizing robot motions adapted to human presence », *Int. Journal of Social Robotics*, t. 2, n° 3, p. 329–343, 2010 (cf. p. 33).

[59] E. SISBOT et R. ALAMI, « A human-aware manipulation planner », *IEEE Transactions on Robotics*, t. 28, n° 5, p. 1045–1057, 2012 (cf. p. 33).

[60] H. SUAY et E. SISBOT, « A position generation algorithm utilizing a biomechanical model for robot-human object handover », in *2015 IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 2015, p. 3776–3781 (cf. p. 33, 38, 64).

[61] K. KOAY, K. DAUTENHAHN, S. WOODS et M. WALTERS, « Empirical results from using a comfort level device in human-robot interaction studies », in *Proceedings of the 1st ACM SIGCHI/SIGART Conf. on Human-robot interaction*, 2006, p. 194–201 (cf. p. 33, 34).

[62] K. KOAY, E. SISBOT, D. SYRDAL, M. WALTERS, K. DAUTENHAHN et R. ALAMI, « Exploratory study of a robot approaching a person in the context of handing over an object. », in *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*, 2007, p. 18–24 (cf. p. 33, 34).

[63] K. DAUTENHAHN, M. WALTERS, S. WOODS, K. KOAY, C. NEHANIV, A. SISBOT, R. ALAMI et T. SIMÉON, « How may I serve you? a robot companion approaching a seated person in a helping context », in *Proceedings of the 1st ACM SIGCHI/SIGART Conf. on Human-robot interaction*, 2006, p. 172–179 (cf. p. 33).

[64] E. HALL, *The hidden dimension*. Garden City, NY: Doubleday, 1966, t. 609 (cf. p. 33).

[65] M. KATAYAMA et H. HASUURA, « Optimization principle determines human arm postures and "comfort" », in *SICE Annual Conf. Program and Abstracts SICE Annual Conf. 2003*, The Society of Instrument et Control Engineers, 2003, p. 47–47 (cf. p. 33, 38, 64).

[66] M. KÖLSCH, A. BEALL et M. TURK, « The postural comfort zone for reaching gestures », in *Proceedings of the human factors and ergonomics society annual meeting*, SAGE Publications Sage CA: Los Angeles, CA, t. 47, 2003, p. 787–791 (cf. p. 33).

[67] H. NEMLEKAR, D. DUTIA et Z. LI, « Object transfer point estimation for fluent human-robot handovers », in *2019 Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 2019, p. 2627–2633 (cf. p. 34, 36).

[68] B. BUSCH, G. MAEDA, Y. MOLLARD, M. DEMANGEAT et M. LOPES, « Postural optimization for an ergonomic human-robot interaction », in *2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2017, p. 2778–2785 (cf. p. 34).

[69] J. MAINPRICE, M. GHARBI, T. SIMÉON et R. ALAMI, « Sharing effort in planning human-robot handover tasks », in *2012 IEEE RO-MAN: The 21st IEEE Int. Symposium on Robot and Human Interactive Communication*, IEEE, 2012, p. 764–770 (cf. p. 34).

[70] J. WALDHART, M. GHARBI et R. ALAMI, « Planning handovers involving humans and robots in constrained environment », in *2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2015, p. 6473–6478 (cf. p. 34).

[71] Y. PYO, K. NAKASHIMA, T. TSUJI, R. KURAZUME et K. MOROOKA, « Motion planning for fetch-and-give task using wagon and service robot », in *2015 IEEE Int. Conf. on Advanced Intelligent Mechatronics (AIM)*, 2015, p. 925–932 (cf. p. 34, 59).

[72] K. YAMANE, M. REVFI et T. ASFOUR, « Synthesizing object receiving motions of humanoid robots with human motion database », in *2013 IEEE Int. Conf. on Robotics and Automation*, IEEE, 2013, p. 1629–1636 (cf. p. 35, 36).

[73] J. MEDINA, F. DUVALLET, M. KARNAM et A. BILLARD, « A human-inspired controller for fluid human-robot handovers », in *2016 IEEE-RAS 16th Int. Conf. on Humanoid Robots (Humanoids)*, IEEE, 2016, p. 324–331 (cf. p. 35, 36).

[74] M. CAKMAK, S. SRINIVASA, M. LEE, J. FORLIZZI et S. KIESLER, « Human preferences for robot-human hand-over configurations », in *2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, IEEE, 2011, p. 1986–1993 (cf. p. 35).

[75] A. ARISTIDOU et J. LASENBY, « FABRIK: A fast, iterative solver for the Inverse Kinematics problem », *Graphical Models*, t. 73, n° 5, p. 243–260, 2011 (cf. p. 35).

[76] V. ORTENZI, N. MARTURI, V. RAJASEKARAN, M. ADJIGBLE et R. STOLKIN, « Singularity-robust inverse kinematics solver for tele-manipulation », in *2019 IEEE 15th Int. Conf. on Automation Science and Engineering (CASE)*, IEEE, 2019, p. 1821–1828 (cf. p. 35).

[77] M. HUBER, M. RICKERT, A. KNOLL, T. BRANDT et S. GLASAUER, « Human-robot interaction in handing-over tasks », in *RO-MAN 2008-The 17th IEEE Int. Symposium on Robot and Human Interactive Communication*, IEEE, 2008, p. 107–112 (cf. p. 35).

[78] S. TARBOURIECH, B. NAVARRO, P. FRAISSE, A. CROSNIER, A. CHERUBINI et S. DAMIEN, « Admittance control for collaborative dual-arm manipulation », in *19th Int. Conf. on Advanced Robotics (ICAR)*, IEEE, 2019, p. 198–204 (cf. p. 35, 89).

[79] M. GIULIANI, C. LENZ, T. MÜLLER, M. RICKERT et A. KNOLL, « Design principles for safety in human-robot interaction », *Int. Journal of Social Robotics*, t. 2, n° 3, p. 253–274, 2010 (cf. p. 35).

[80] G. MAEDA, G. NEUMANN, M. EWERTON, R. LIOUTIKOV, O. KROEMER et J. PETERS, « Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks », *Autonomous Robots*, t. 41, n° 3, p. 593–612, 2017 (cf. p. 35).

[81] C.-M. HUANG, M. CAKMAK et B. MUTLU, « Adaptive Coordination Strategies for Human-Robot Handovers. », in *Robotics: Science and systems*, Rome, Italy, t. 11, 2015 (cf. p. 35, 36).

[82] M. FUJITA, R. KATO et A. TAMIO, « Assessment of operators' mental strain induced by hand-over motion of industrial robot manipulator », in *19th Int. Symposium in Robot and Human Interactive Communication*, IEEE, 2010, p. 361–366 (cf. p. 35).

[83] A. KOENE, S. ENDO, A. REMAZEILLES, M. PRADA et A. WING, « Experimental testing of the Coglaboration prototype system for fluent Human-Robot object handover interactions », in *The 23rd IEEE Int. Symposium on Robot and Human Interactive Communication*, IEEE, 2014, p. 249–254 (cf. p. 35).

[84] M. PRADA, A. REMAZEILLES, A. KOENE et S. ENDO, « Implementation and experimental validation of dynamic movement primitives for object handover », in *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, IEEE, 2014, p. 2146–2153 (cf. p. 35).

[85] Y. HASSON, G. VAROL, D. TZIONAS, I. KALEVATYKH, M. BLACK, I. LAPTEV et C. SCHMID, « Learning joint reconstruction of hands and manipulated objects », in *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2019, p. 11 807–11 816 (cf. p. 36).

[86] C. ZIMMERMANN, D. CEYLAN, J. YANG, B. RUSSELL, M. ARGUS et T. BROX, « Freihand: A dataset for markerless capture of hand pose and shape from single RGB images », in *Proceedings of the IEEE/CVF Int. Conf. on Computer Vision*, 2019, p. 813–822 (cf. p. 36).

[87] S. SHIVAKUMAR, M. OBERWEGER, M. RAD et V. LEPETIT, « HO-3D: A Multi-User, Multi-Object Dataset for Joint 3D Hand-Object Pose Estimation », *ArXiv. org e-Print archive*, 2019 (cf. p. 36).

[88] M. COSTANZO, G. DE MARIA et C. NATALE, « Handover Control for Human-Robot and Robot-Robot Collaboration », *Frontiers in Robotics and AI*, t. 8, p. 132, 2021 (cf. p. 36, 37).

[89] M. PAN, E. CROFT et G. NIEMEYER, « Exploration of geometry and forces occurring within human-to-robot handovers », in *2018 IEEE Haptics Symposium (HAPTICS)*, IEEE, 2018, p. 327–333 (cf. p. 36).

[90] X. ZHAO, S. CHUMKAMON, S. DUAN, J. ROJAS et J. PAN, « Collaborative human-robot motion generation using LSTM-RNN », in *2018 IEEE-RAS 18th Int. Conf. on Humanoid Robots (Humanoids)*, IEEE, 2018, p. 1–9 (cf. p. 36).

[91] W. CHAN, C. PARKER, H. VAN DER LOOS et E. CROFT, « A human-inspired object handover controller », *The Int. Journal of Robotics Research*, t. 32, n° 8, p. 971–983, 2013 (cf. p. 37).

[92] ——, « Grip forces and load forces in handovers: Implications for designing human-robot handover controllers », in *Proceedings of the seventh annual ACM/IEEE Int. Conf. on Human-Robot Interaction*, 2012, p. 9–16 (cf. p. 37).

[93] A. PANDEY et R. GELIN, « A mass-produced sociable humanoid robot: Pepper: The first machine of its kind », *IEEE Robotics & Automation Magazine*, t. 25, n° 3, p. 40–48, 2018 (cf. p. 42).

[94] C.-W. CHANG, J.-H. LEE, P.-Y. CHAO, C.-Y. WANG et G.-D. CHEN, « Exploring the possibility of using humanoid robots as instructional tools for teaching a second language in primary school », *Journal of Educational Technology & Society*, t. 13, n° 2, p. 13–24, 2010 (cf. p. 42).

[95] M. MORI, K. MACDORMAN et N. KAGEKI, « The uncanny valley [from the field] », *IEEE Robotics & Automation Magazine*, t. 19, n° 2, p. 98–100, 2012 (cf. p. 42).

[96] A. CHEVALLIER, C. BAILLEHACHE et A. SINGH, *Comparison of Pepper's OS versions [En ligne]*, Disponible : https://developer.softbankrobotics.com/blog/comparison-peppers-os-versions, (31 déc. 2020) (cf. p. 45).

[97] A. BOLOTNIKOVA, P. GERGONDET, A. TANGUY, S. COURTOIS et A. KHEDDAR, « Task-Space Control Interface for Softbank Humanoid Robots and its Human-Robot Interaction Applications », in *2021 IEEE/SICE Int. Symposium on System Integration (SII)*, IEEE, 2021, p. 560–565 (cf. p. 51).

[98] A. KAPANDJI, *Anatomie fonctionnelle: Physiologie de l'appareil locomoteur, t. 1: Membres supérieurs*. 2005, p. 4–13, 78, 98–99 (cf. p. 55, 56).

[99] L. HOSEY, « Hidden Lines: Gender, Race, and the Body in Graphic Standards », *Journal of Architectural Education*, t. 55, n° 2, p. 101–112, 2001 (cf. p. 55, 58).

[100] N. DIFFRIENT, A. TILLEY et D. HARMAN, *Humanscale*. Henry Dreyfuss Associates, 1974 (cf. p. 55, 58).

[101] D. WINTER, *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009 (cf. p. 55, 61).

[102] O. KHATIB, E. DEMIRCAN, V. DE SAPIO, L. SENTIS, T. BESIER et S. DELP, « Robotics-based synthesis of human motion », *Journal of Physiology-Paris*, t. 103, n° 3-5, p. 211–219, 2009 (cf. p. 59).

[103] M. ENGARDT, A. HEIMBURGER et P. SYDHOFF, *Manipulability index optimization for a planar robotic arm*, 2012 (cf. p. 59).

[104] R. HARTLEY et A. ZISSERMAN, *Multiple view geometry in computer vision*. Cambridge University Press, 2003 (cf. p. 68).

[105] STRUCTURE BY OCCIPITAL, *OpenNI 2 SDK Binaries and Docs [En ligne]*, Disponible : https://structure.io/openni (cf. p. 71).

[106] MICROSOFT, *Azure Kinect Developer Kit [En ligne]*, Disponible : https://azure.microsoft.com/en-gb/services/kinect-dk/ (cf. p. 71).

[107] INTEL, *Intel RealSense SDK 2.0: Start building your own depth applications [En ligne]*, Disponible : https://www.intelrealsense.com/developers/ (cf. p. 71).

[108] K. KHOSHELHAM et S. ELBERINK, « Accuracy and resolution of kinect depth data for indoor mapping applications », *Sensors*, t. 12, n° 2, p. 1437–1454, 2012 (cf. p. 72).

[109] J. SMISEK, M. JANCOSEK et T. PAJDLA, « 3D with Kinect », in *Consumer depth cameras for computer vision*, Springer, 2013, p. 3–25 (cf. p. 72).

[110] A. REIMANN, *Intrinsic calibration of the Kinect cameras [En ligne]*, Disponible : http://wiki.ros.org/openni_launch/Tutorials/IntrinsicCalibration, Dernière modification le 6 févr. 2015 (cf. p. 72).

[111] Z. WU, W. ZHU et Q. ZHU, « Semi-Transparent Checkerboard Calibration Method for KINECT's Color and Depth Camera », in *2018 Int. Conf. on Network, Communication, Computer Engineering (NCCE 2018)*, Atlantis Press, 2018 (cf. p. 72).

[112] H. LIU, D. QU, F. XU, F. ZOU, J. SONG et K. JIA, « Approach for accurate calibration of RGB-D cameras using spheres », *Optics Express*, t. 28, n° 13, p. 19 058–19 073, 2020 (cf. p. 73).

[113] Z. CAO, G. HIDALGO, T. SIMON, S. WEI et Y. SHEIKH, « Openpose: Realtime multi-person 2D pose estimation using Part Affinity Fields », *IEEE transactions on pattern analysis and machine intelligence*, t. 43, n° 1, p. 172–186, 2019 (cf. p. 73, 80).

[114] J. NOWAK, P. FRAISSE, A. CHERUBINI et J. DAURES, « Point Clouds With Color: A Simple Open Library for Matching RGB and Depth Pixels from an Uncalibrated Stereo Pair », in *2021 IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, IEEE, 2021, p. 1–7 (cf. p. 77).

[115] J. NOWAK, *OpenHSML - Open-source Hybrid Stereovision Matching Library. [En ligne]*, Disponible : https://github.com/jordan-nowak/OpenHSML, (Oct. 2020) (cf. p. 77).

[116] FREE SOFTWARE FOUNDATION, *GNU lesser general public license. [En ligne]*, Disponible : https://www.gnu.org/licenses/lgpl-3.0.en.html, (18 Nov. 2016) (cf. p. 77).

[117] D. MEHTA, S. SRIDHAR, O. SOTNYCHENKO, H. RHODIN, M. SHAFIEI, H. SEIDEL, W. XU, D. CASAS et C. THEOBALT, « Vnect: Real-time 3D human pose estimation with a single RGB camera », *ACM Transactions on Graphics (TOG)*, t. 36, n° 4, p. 1–14, 2017 (cf. p. 85).

[118] S. TARBOURIECH, « Dual-arm control strategy in industrial environments », thèse de doct., Université de Montpellier, 2019, p. 41–43 (cf. p. 89).

[119] G. SUDDREY, A. JACOBSON et B. WARD, « Enabling a pepper robot to provide automated and interactive tours of a robotics laboratory », *ArXiv preprint arXiv:1804.03288*, 2018 (cf. p. 97).

[120] V. PERERA, T. PEREIRA, J. CONNELL et M. VELOSO, « Setting up pepper for autonomous navigation and personalized interaction with users », *ArXiv preprint arXiv:1704.04797*, 2017 (cf. p. 97).

[121] S. GARRIDO-JURADO, R. MUÑOZ-SALINAS, F. MADRID-CUEVAS et R. MEDINA-CARNICER, « Generation of fiducial marker dictionaries using mixed integer linear programming », *Pattern Recognition*, t. 51, p. 481–491, 2016 (cf. p. 97).

[122] F. ROMERO-RAMIREZ, R. MUÑOZ-SALINAS et R. MEDINA-CARNICER, « Speeded up detection of squared fiducial markers », *Image and vision Computing*, t. 76, p. 38–47, 2018 (cf. p. 97).

[123] E. DIJKSTRA, *A short introduction to the art of programming*. Technische Hogeschool Eindhoven, 1971, t. 4 (cf. p. 98, 99).

[124] C. DEROUESNE, J. POITRENEAU, L. HUGONOT, M. KALAFAT, B. DUBOIS et B. LAURENT, « Mini-Mental State Examination: A useful method for the evaluation of the cognitive status of patients by the clinician. Consensual French version », *Presse medicale (Paris, France: 1983)*, t. 28, n° 21, p. 1141–1148, 1999 (cf. p. 118).

Annexe D

Contribution de l'auteur

Articles de conférence internationale

J. NOWAK, P. FRAISSE, A. CHERUBINI et J.-P. DAURES (Septembre 2021). « Point clouds with color: A simple open library for matching rgb and depth pixels from an uncalibrated stereo pair » *In 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 1–7. IEEE.

J. NOWAK, P. FRAISSE, A. CHERUBINI et J.-P. DAURES (Mai 2022). « Assistance to Older Adults with Comfortable Robot-to-Human Handovers » *In 2022 IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO)*, pp. 1–6. IEEE.