



ÉCOLE UNIVERSITAIRE
DE PHYSIQUE ET D'INGÉNIERIE
Université Clermont Auvergne



UNIVERSITE CLERMONT AUVERGNE

Clermont-Ferrand

Ecole Universitaire de Physique et d'Ingénierie

MEMOIRE DE STAGE

2^{ème} année de Master

Spécialité : Perception artificielle et Robotique (PAR)

HANI DANIEL ZAKARIA Mélodie

CONTROLE MULTIMODAL DE L'INTERACTION HUMAIN-HUMANOIDE

Encadrants : Youcef MEZOUAR, Sébastien LENGAGNE et Juan CORRALES

Date de soutenance : 13/09/2019

Laboratoire de recherche : Institut Pascal



Résumé

Le travail effectué à l'institut Pascal au sein du projet ATTRIHUM consiste à préparer une interaction homme-robot dans le but de connaître les répercussions cognitives sur l'humain. Nous avons, tout d'abord, fait un récapitulatif des principales branches de l'interaction homme-robot en incluant toutes les disciplines nécessaires pour en réaliser une.

Nous avons par la suite choisi de préparer une manipulation consistant à rendre Nao, un robot humanoïde, capable de jouer à "Camelot Jr", jeu où il faut construire un chemin pour que deux figurines puissent se rejoindre. Pour cela, il faut s'occuper de deux aspects à savoir la perception et le contrôle. Nous nous sommes servi de QR codes pour assurer la détection et la localisation d'objets à savoir les cubes qui serviront pour construire le chemin et la base sur laquelle le puzzle est construit. Nous avons utilisé un contrôleur basé sur une pile de tâches et qui commande tout le corps du robot pour faire atteindre à la main la position désirée. Nous avons démontré que cette plateforme expérimentale est bonne à travers des tests que nous avons effectués.

Finalement, nous allons présenter un cadre méthodologique permettant de se servir de cette plateforme expérimentale pour mettre au point une interaction entre un humain et Nao. Ce cadre sera basé sur la théorie des jeux qui permet dans ce contexte au robot de "penser" comme un humain et de pouvoir s'adapter aux autres joueurs. Cette dernière partie sera mise en place lors d'une thèse.

Remerciements

En premier lieu, je tiens à exprimer ma gratitude envers mes encadrants Monsieur Youcef MEZOUAR, Monsieur Sébastien LENGAGNE et Monsieur Juan CORRALES qui ont eu confiance en moi et ont accepté de diriger ce travail. Leurs conseils judicieux et leur disponibilité sans faille m'ont apporté secours aux moments les plus difficiles de l'accomplissement de ce travail.

De même, ma reconnaissance à l'égard de mon tuteur universitaire Monsieur Romuald AUFRÈRE n'a pas de limite car il était toujours là pour résoudre tout problème administratif dans le plus bref délai.

Ma considération et respect s'adressent aux membres du jury qui m'ont accordé un grand honneur en acceptant de lire ce travail malgré leurs occupations.

Je remercie également le personnel administratif de l'Institut Pascal de son aide et disponibilité.

Enfin, je tiens à exprimer ma gratitude au LABEX IMOBS3 qui a financé ce stage dans le cadre du projet CAP 20-25.

Table des matières

Introduction générale	4
1 Contexte	6
1.1 Introduction	6
1.2 Projet ATTRIHUM	6
1.3 Institut Pascal	7
1.4 Thématique et cahier de charges	9
1.5 Plan du stage	11
2 État de l’art	12
2.1 Introduction	12
2.2 Types d’interaction	12
2.2.1 Cognitive	13
2.2.2 Physique	15
2.3 Disciplines de l’interaction Homme-Robot	18
2.3.1 Perception (reconnaissance) pour l’interaction	18
2.3.2 Réalisation (contrôle/commande)	21
2.3.3 Évaluation	23
2.4 Conclusion	25
3 Plateforme expérimentale	26
3.1 Introduction	26
3.2 Nao	26
3.2.1 Présentation du robot	26
3.2.2 Spécifications de Nao 6 [1, 2]	27
3.2.3 Chorégraphe	28
3.2.4 Limitations et avantages	29
3.3 ROS	30
3.3.1 Définition	30
3.3.2 Historique de ROS	30
3.3.3 Le système de fichiers de ROS	30
3.3.4 Les notions de base de ROS	30
3.3.5 Fonctions de base de ROS	31
3.3.6 URDF	32
3.3.7 Langages de programmation	32
3.3.8 Robots	32
3.3.9 Conclusion	32
3.4 Application	33
3.4.1 La nuit des fantômes	33

3.4.2	Code couleur	34
3.4.3	IQ	34
3.4.4	Camelot Jr	34
3.5	Conclusion	35
4	Techniques et méthodes	36
4.1	Introduction	36
4.2	Perception	36
4.2.1	Calibration de la caméra	36
4.2.2	Localisation des marqueurs à l'aide d'Aruco	37
4.3	Contrôleur	38
4.4	Programmes	39
4.5	Conclusion	40
5	Résultats	41
5.1	Introduction	41
5.2	Configuration de départ	41
5.3	Puzzle 1	41
5.4	Analyse des résultats du puzzle 1	42
5.4.1	Construction biaisée	42
5.4.2	occultations	42
5.4.3	Non détections	42
5.4.4	Répétabilité	43
5.5	Puzzle 2	43
5.6	Analyse des résultats du puzzle 2	43
5.6.1	Défi relevé	43
5.6.2	Limites articulaires	43
5.6.3	Jeu articulaire	44
5.7	Puzzle 3	44
5.8	Analyse des résultats du puzzle 3	44
5.9	Modification de la plateforme expérimentale	44
5.10	Conclusion	45
6	Perspectives	49
6.1	Introduction	49
6.2	Théorie des jeux	49
6.2.1	Jeux coopératifs	49
6.2.2	Jeux séquentiels et jeux simultanés	50
6.2.3	Jeux à somme nulle et jeux à somme non nulle	50
6.2.4	Jeux répétés	51
6.2.5	Jeux à information complète ou non	51
6.2.6	Jeux à mémoire parfaite et imparfaite	51
6.2.7	Jeux en cas de joueurs "irrationnels"	51
6.3	Cadre Méthodologique	52
6.4	Conclusion	53
	Conclusion générale	54
	Références	55

A Informations brutes (mesures capteurs)	64
B Contrôleur de Nao	67
B.1 Mouvement d'un bras	67
B.1.1 Bras droit	67
B.1.2 Bras gauche	67
B.2 Mouvement de la tête	67
C Nœuds	68
C.1 Relation entre les différents nœuds	68
C.2 Présentation des 3 nœuds principaux	69
C.2.1 Nœud "/recherche_cube"	69
C.2.2 Nœud "/saisie_cube"	71
C.2.3 Nœud "/solve_puzzle"	74

Introduction générale

L'interaction homme-robot (HRI) peut être définie comme un domaine dédié à la compréhension, à la conception et à l'évaluation des méthodes mises en place autour de systèmes robotiques destinés à être utilisés par ou avec des humains [3]. L'interaction homme-robot est le sujet d'un champ de recherches interdisciplinaires, à la frontière entre la robotique, l'ergonomie et la psychologie, qui cherche à comprendre et à développer des technologies pour soutenir la coexistence entre robots (humanoïdes) et humains. L'interaction homme-robot est la plupart du temps bidirectionnelle mais peut être, dans quelques cas spéciaux (e.g. exosquelette), monodirectionnelle. On dit qu'une interaction est bidirectionnelle lorsque l'humain et le robot sont tous deux capables de raisonnement logique c'est-à-dire que le robot est également capable de prendre des décisions tout seul. Contrairement, le cas monodirectionnel c'est où l'humain décide et le robot suit machinalement sans volonté propre et c'est ce qu'on appelle une relation maître-esclave. L'interaction est composée d'un ou de plusieurs robots, d'un ou de plusieurs humains et d'un environnement.

- Les robots comportent tout dispositif mécatronique (alliant mécanique, électronique et informatique) conçu pour accomplir des tâches imitant ou reproduisant des actions humaines dans le cadre d'un domaine précis. On peut considérer qu'ils sont composés de deux sous-parties indépendantes : la première permet la coordination sensori-motrice pour simuler un comportement animal ou social ; la seconde, essentiellement logicielle, permet l'interaction "intelligente" avec l'humain. Dans le cas de robots collaboratifs industriels, nous allons plutôt utiliser le terme "cobot".
- L'humain désigné ici est l'opérateur qui va être en interaction avec le robot.
- L'environnement définit le lieu où se trouve le robot et auquel il devra s'adapter comme une maison, une usine, un bureau, un musée, ...etc. Mais il désigne aussi tout ce qui entoure le robot c'est-à-dire tout ce qu'il y a dans la pièce dans laquelle se trouve le robot.

Pour que les robots puissent participer avec succès à des activités bidirectionnelles d'interactions sociales avec les gens, ils doivent être capables de reconnaître, d'interpréter et de réagir efficacement aux repères sociaux affichés par un humain pendant l'interaction. En particulier, lors des interactions sociales, les expressions affectives d'une personne peuvent communiquer ses pensées et son intention sociale [4]. L'affect d'une personne est une combinaison complexe d'émotions, d'humeurs, de positions interpersonnelles, d'attitudes et de traits de personnalité qui influencent son propre comportement [5]. C'est à partir de l'affect qu'on peut mettre en place le cadre pour évaluer l'interaction. Un robot capable d'interpréter l'affect aura une capacité accrue de prendre des décisions et d'assister les humains grâce à sa capacité à réagir à leur affect [6]. Les environnements de tous les jours, comme les maisons, ont des règles sociales beaucoup plus complexes que celles appliquées dans les usines ou même dans les environnements militaires. Ainsi, le robot a-t-il besoin de capacités de perception et de compréhension

pour construire des modèles dynamiques de son environnement afin de pouvoir s'y adapter. Il doit donc être en mesure de classifier les objets, reconnaître et localiser les humains et faire constat de leurs émotions en vue de pouvoir suivre, comprendre et prédire les mouvements humains en temps réel dans un environnement dynamique faiblement structuré.

Parmi les applications actuelles de tous les robots humanoïdes, mentionnons les guides-interprètes robotisés dans les musées [7], les aides thérapeutiques dans les foyers de soins [8] et les garderies [9, 10], les aides pédagogiques dans les salles de classe [11, 12, 13], et les applications futures qui devraient être beaucoup plus vastes. Les applications futures possibles des robots mis au point pour une interaction physique avec les humains sont illustrées par la Figure 1. Les robots peuvent servir de collègues de travail dans le cadre industriel, de serveurs mobiles, d'appareils ou accessoires fonctionnels pour les personnes à mobilité réduite et de service d'entretien dans le cadre des activités ménagères en général. Toutes ces applications partagent l'exigence commune d'une interaction physique étroite, sûre et fiable entre l'homme et le robot dans un espace de travail partagé [14].



FIGURE 1 – Exemples d'applications des pHRI (physical Human Robot Interaction), allant de la logistique et de la manipulation en atelier (a,b), en passant par les robots professionnels de service et les appareils d'assistance pour handicapés (c,d), aux robots de service pour applications domestiques (e,f) [14]

Ce stage se poursuit par une thèse sur la thématique de l'interaction homme-robot, ainsi durant lequel nous avons mis en place la plateforme expérimentale où le robot effectue tout seul la tâche qui servira de point de départ pour mettre au point l'interaction avec l'humain dans le cadre de la thèse. Le stage a servi à bien connaître l'état de l'art (un article de revue est en cours de rédaction), nous avons aussi pris en main le matériel ainsi que la mise au point de la plateforme expérimentale et proposé le cadre méthodologique à adopter durant la thèse.

Dans le premier chapitre nous présentons le cadre du projet ATTRIHUM dans lequel ce travail a été effectué ainsi que l'Institut Pascal, laboratoire au sein duquel le stage a été réalisé ; puis nous décrivons le cahier des charges et montrons le diagramme de Gantt du stage. Le chapitre 2 est composé d'un bref résumé de l'état de l'art en matière d'interaction homme-robot. Celle-ci est présentée dans le chapitre 3 laquelle se compose de Nao qui a été programmé sous ROS avec du Python et du C++, et du jeu Camelot Jr. Le robot est censé pouvoir réaliser la solution du puzzle tout seul, solution qu'il connaît d'avance. Ensuite, nous explicitons, dans le chapitre 4, les méthodes que nous avons utilisées pour la perception et pour la commande de Nao (le robot humanoïde utilisé) et ce pour mettre en œuvre la plateforme expérimentale. Nous montrons dans le chapitre 5 les résultats expérimentaux que nous avons obtenus pour terminer par le chapitre 6 qui contient les perspectives futures à réaliser durant la thèse.

Chapitre 1

Contexte

1.1 Introduction

Dans ce chapitre, nous allons tout d'abord présenter le projet ATTRIHUM, cadre dans lequel ce travail a été réalisé. Puis nous allons parler de l'Institut Pascal, laboratoire au sein duquel ce stage a été effectué (plus précisément dans l'équipe MACCS de l'axe ISPR). Ensuite nous énoncerons le cahier de charges du stage et enfin, nous allons expliciter le plan final que nous avons suivi pour faire ce travail.

1.2 Projet ATTRIHUM

Ce stage est effectué dans le cadre du projet ATTRIHUM qui porte sur la Robotique Humanoïde et la Cognition (RHC). L'objectif de ce projet est de comprendre l'influence de l'apparence humanoïde de certains robots sur les processus cognitifs, cérébraux et neurophysiologiques des humains en interaction avec les robots socio-humanoïdes. Ces effets seront évalués pour tous les âges de la vie. Ce projet regroupe 4 laboratoires :

- Le LAPSCO (labo de psychologie sociale et cognitive) qui est le laboratoire principal du projet et qui traite la partie du projet concernant les sciences cognitives et l'influence de l'environnement social sur la cognition dans le cadre d'une interaction homme-robot.
- Le LIMOS est un laboratoire d'informatique, de modélisation et d'optimisation des systèmes qui gèrera la partie liée à l'interprétation de l'interaction et plus précisément de l'état émotionnel de l'humain en interaction avec le robot pour adapter le comportement du robot en fonction de l'humain.
- Le PHIER est un laboratoire de philosophies et Rationalités qui s'occupera du questionnement philosophique suscité par le développement de la robotique sociale humanoïde dont le point culminant est la distinction homme-machine. Le problème est généré par les capacités cognitives mais surtout émotionnelles des robots.
- L'Institut Pascal est le laboratoire qui traite tout ce qui a attiré à la robotique comme la perception des robots et surtout leur contrôle.

1.3 Institut Pascal

Ce stage s'est déroulé au sein de l'Institut Pascal¹ (UMR 6602), un laboratoire de recherche interdisciplinaire né de la fusion successive - en 2012, puis en 2017 - à vocation structurante de six laboratoires couvrant les domaines stratégiques des sciences de l'ingénierie et des systèmes dont le génie des procédés, la mécanique, la robotique, la physique des sciences de l'information et la santé. L'institut Pascal est placé sous la tutelle de l'Université Clermont Auvergne (UCA), du CNRS (institut principal INSIS, instituts secondaires INS2I, INP) et de l'école d'ingénieur SIGMA Clermont. Le CHU de Clermont-Ferrand est un des partenaires du laboratoire ainsi que les partenaires industriels : MICHELIN, RENAULT, PSA, LIGIER, CONSTELIUM, LIMAGRAIN, SNCF, EDF, SAFRAN, BABYMOOV, ANSALDO, ALEDIA, QUANTEL MEDICAL, SUPERSONIC IMAGINE, THALES, TERUMO EUROPE, ... etc.

L'Institut Pascal développe des systèmes innovants et intelligents par une approche systémique et multi-échelle qui s'appuient sur les champs scientifiques suivants : électromagnétisme, fiabilité, génie des procédés et des bioprocédés, génie civil, imagerie, matériaux, mécanique, micro-systèmes, nanophysique, perception, photonique, robotique, séparation et fonctionnalisation de biomolécules, technologies interventionnelles et diagnostiques médicales.

L'institut Pascal est en Collaboration avec plus de 50 laboratoires en France dont Limos, Lapsco, Ecole Polytechnique, Institut Néel, C2N, CRHEA, GEPEA, Rapsodee, Instituts Fresnel et Laplace, INRIA, Ceream, Irstea, INRA, CEA, ... etc. et plus de 50 Universités et organismes de recherche internationaux dont University of London King's College, Oxford, Cambridge, Singapour, ESA Agence Spatiale Européenne, UAS Université Autonome de Barcelone, Bri Corée (Université de Sungkyunkwan/Corée du Sud), Oklahoma University, University of Canberra, Institut IOFFE, ... etc. Le laboratoire a pour directrice Evelyne Gil et son personnel compte 389 personnes et est composé de cinq axes de recherche (cf. Figure 1.1) :

- Génie des Procédés, Énergétique et Biosystèmes (GePEB) dont le responsable est Claude-Gilles DUSSAP. L'axe a pour domaines d'étude : Réacteurs et bioréacteurs, Photoprocédés, Bioénergies, Ecosystèmes clos artificiels, Bioraffineries.
- Image, Systèmes de Perception, Robotique (ISPR) dont le responsable est Lounis Adouane. L'axe a pour domaines d'étude : Image, Perception artificielle, Commande pour la robotique.
- Mécanique, Génie Mécanique, Génie Civil, Génie Industriel (M3G) dont le responsable est Nicolas Gayton. L'axe a pour domaines d'étude : Mécanique, Matériaux, Structure, Machines.
- Photonique, Ondes, Nanomatériaux (PHOTON) dont le responsable est Guillaume Malpuech. L'axe a pour domaines d'étude : MicroSystèmes, Nanosciences, Nanostructures, Photonique, Compatibilité Electromagnétique.
- Thérapies Guidées par l'Image (TGI) dont le responsable est Jean-Jacques Lemaire. L'axe a pour domaines d'étude : Guidage par l'Image (pelvis, abdomen, cerveau), Techniques endovasculaires, Endoscopie et Vision par ordinateur, Analyse d'impact.

L'axe dont dépend notre thématique est l'ISPR (Image, Systèmes de Perception, Robotique). Cet axe apporte des contributions scientifiques dans le domaine de la perception et de la vision

1. Toutes les informations et organigrammes de cette section sont issus du site internet de l'Institut Pascal : <http://www.institutpascal.uca.fr/index.php/fr/>

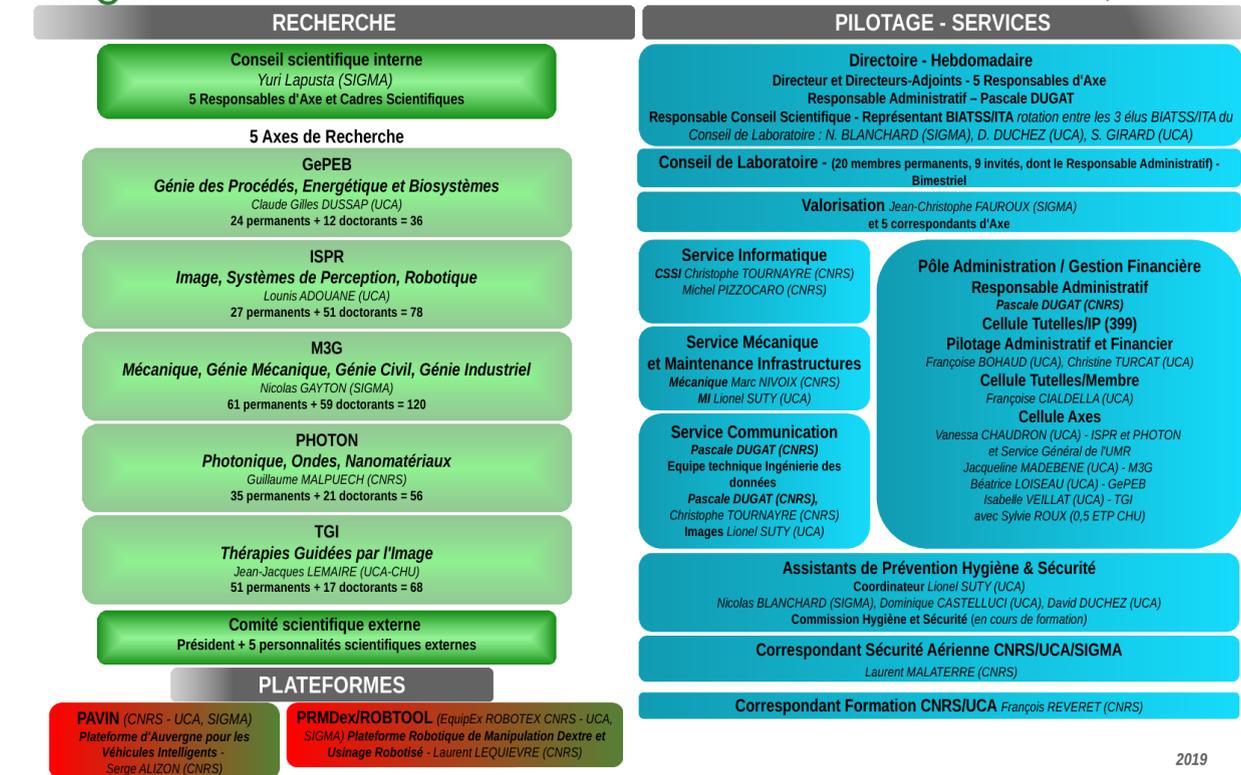


FIGURE 1.1 – Organigramme de l'Institut Pascal (IP)

artificielle pour la commande des systèmes robotiques. Son objectif est donc le développement des concepts théoriques, méthodologiques et architecturaux pour la perception et le contrôle des systèmes. Ses points forts sont :

- Les architectures matérielles et logicielles pour la perception artificielle.
- L'étalonnage de capteurs de précision et reconstruction 3D sans contact d'objets rigides ou déformables.
- La reconnaissance automatique est suivie en temps réel de motifs visuels par des techniques d'apprentissage et plus particulièrement de réseaux de neurones profonds.
- La localisation absolue de véhicules mobiles (tout type d'environnement) et cartographie automatique.
- La modélisation et commande de systèmes robotiques.
- La navigation autonome de systèmes robotiques.

L'axe ISPR se compose de 4 équipes (dont l'organigramme est présenté par la Figure 1.2) :

- Vision artificielle (ComSee) : Les travaux de l'équipe ComSee - contraction de COMPUTERS that SEE - portent sur la recherche dans le domaine de la vision par ordinateur et surtout sur deux grands thèmes : la vision géométrique (la modélisation et l'étalonnage de capteurs de vision, la localisation et la reconstruction 3D par vision) et l'interprétation de séquences d'images (détecter, reconnaître, suivre les différents éléments d'une

scène dans des séquences dynamiques complexes et parfois dans des conditions de visibilité dégradées via des techniques d'apprentissage et de transfert d'apprentissage). Les domaines d'application de ces travaux vont des transports intelligents, à la réalité augmentée, en passant par la vidéosurveillance et la vision robotique.

- Modélisation, Autonomie et Contrôle dans les Systèmes Complexes (MACCS) : Les recherches de cette équipe portent sur la modélisation et le contrôle de robots mobiles et manipulateurs, la vision des robots, la vision active, la maîtrise visuelle et les comportements d'anticipation. Les applications de ces recherches sont liées à la fabrication robotisée et aux systèmes de transport intelligents.
- Systèmes de perception multi sensorielle (PerSyst) : Les travaux de recherche de l'équipe PerSyst (Système de Perception) portent sur :
 - Multicapteurs de localisation et fusion : qui consiste à développer de nouvelles approches de localisation prenant en compte à la fois des capteurs de modélisation, du SLAM multi-modalités (lidar, vision, radar, ...), du filtrage optimal, ... etc.
 - Collaboration multi-robots : il s'agit de mettre au point de nouvelles approches pour la localisation non centralisée et multi-robots / multi-capteurs SLAM pour la robotique mobile et en pelotons.
 - Cartographie des scènes, modélisation et compréhension : ces travaux portent sur la construction automatique de nouvelles cartes 3D pour une meilleure compréhension des scènes dynamiques. Les nouvelles approches proposées combinent la cartographie, la modélisation pour les scènes extérieures et l'interprétation.
- Architecture matérielle et logicielle pour la perception (DREAM) : L'équipe DREAM se concentre sur la recherche de pointe dans la modélisation et l'analyse architecturale, langage- approches pour les systèmes embarqués, des caméras intelligentes, middleware et la technologie multi-core . L'expertise du groupe couvre la mise au point de concepts à diverses technologies de mise en œuvre pour la construction de prototypes de travail réalistes dans des domaines d'application comme le traitement d'image et la vision par ordinateur.

Le stage a été effectué au sein de l'équipe MACCS de l'axe ISPR de l'institut Pascal. Travail portant sur la perception et la commande d'un robot humanoïde.

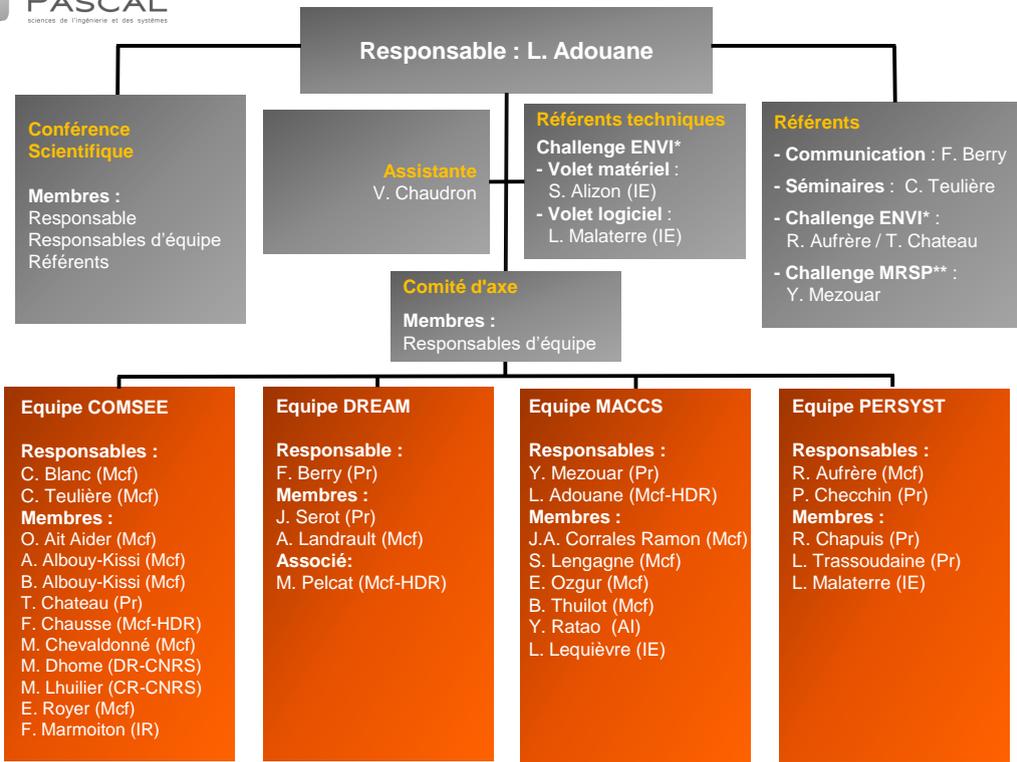
1.4 Thématique et cahier de charges

Le stage sert de point de départ pour la thèse qui sera effectuée au sein du projet ATTRIHUM. Les objectifs du stage devront servir de base et prépareront le terrain pour le travail qui sera effectué pendant la thèse et sont donc : l'étude bibliographique de l'interaction homme-robot (cf. chapitre 2), la prise en main et mise en place de l'environnement expérimental (cf. chapitre 3) et la proposition d'un cadre méthodologique qui sera réalisé pendant la thèse (cf. chapitre 6). L'objectif de la thèse est de mettre au point l'interaction et le contrôle d'un système homme-robot dans le cadre de celle-ci pour des tâches de manipulation (rétroaction interactive).

La contrainte imposée par le LAPSCO exige que l'interaction entre l'humain et le robot soit une interaction à but social c'est-à-dire qu'il y ait un effet cognitif sur l'humain par exemple : une conversation entre les deux et que le robot soit un robot humanoïde social comme Nao et Pepper. Les interactions mises en place dans ce genre de contexte sont : les robots infirmiers [15] qui soutiennent les malades et leur apportent les médicaments, les robots d'accueil [16], ceux



Axe ISPR
Image, Systèmes de Perception, Robotique



* Exploration et Navigation pour Véhicules Intelligents (ENVI) : véhicules routiers et urbains
** Manipulation Robotisée et Système de Perception dans l'usine du futur (MRSP)

FIGURE 1.2 – Organigramme de l'axe ISPR

qui distribuent le courrier dans les bureaux [17] ou encore qui servent de guides touristiques dans les musées [7], ... etc. Ils servent surtout pour les thérapies et rééducations pour les enfants [18], essentiellement ceux atteints de troubles du spectre de l'autisme (TSA) [19, 20].

Comme nous voulons plutôt travailler sur des applications où l'interaction est plus physique, c'est-à-dire que le robot effectue des tâches qui nécessitent un certain effort physique comme déplacer ou porter un objet, nous avons choisi un jeu qui pourra donc servir pour une interaction avec des enfants et qui ressemble à une tâche industrialisable comme l'assemblage de pièces (cf. Figure 1.3). L'environnement expérimental mis au point est détaillé dans le chapitre 4.



FIGURE 1.3 – Tâches d'assemblage industriel [21]

1.5 Plan du stage

Le Gantt présenté par la Figure 1.4 est le plan final, après les adaptations dues aux contraintes de matériel, que nous avons rencontrées. J'ai lu environ 240 articles, ce qui m'a permis de commencer à écrire un article de revue sur l'interaction homme-robot et de pouvoir choisir l'application que nous allons réaliser dans le cadre de l'interaction homme-robot. Nous avons en parallèle pris en main Nao le robot humanoïde que nous avons choisi pour faire l'interaction et ROS le logiciel que nous avons utilisé pour faire la perception des parties de l'environnement nécessaire à la réalisation de l'application et le contrôle des différentes parties du corps du robot : les doigts, les bras et le pied. Il faut ensuite penser à la maniabilité des bras du robot pour que le robot soit capable de changer l'orientation d'un objet ou prendre un objet d'une main et le poser dans l'autre, ... etc. Enfin, il faut mettre en place un cadre méthodologique pour la thèse. Nous avons opté pour faire une étude bibliographique sur la théorie des jeux. Tout ces points seront explicités dans les chapitres suivants.

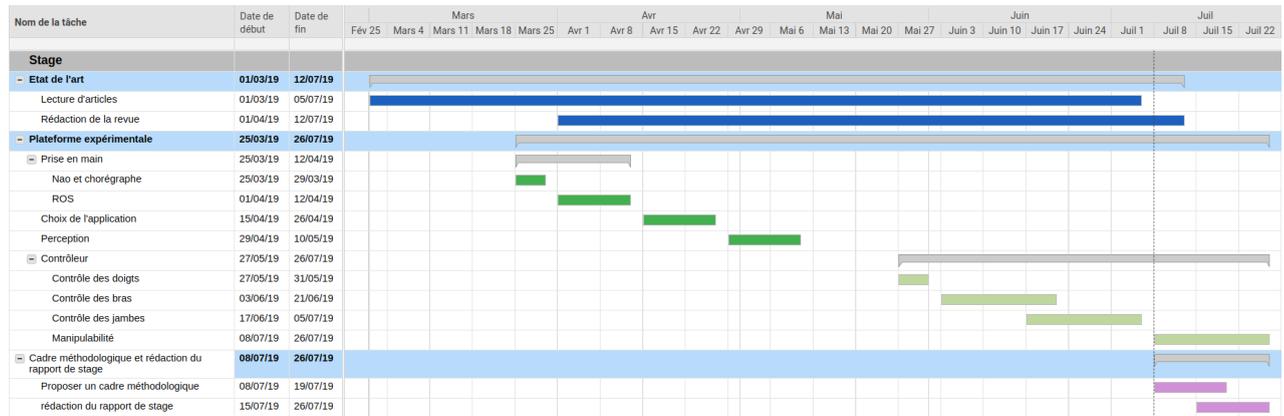


FIGURE 1.4 – Le diagramme de Gantt du stage

Chapitre 2

État de l'art

2.1 Introduction

Ce chapitre, bref récapitulatif de l'interaction homme-robot, présente les différents types d'interaction (cognitive ou physique) à savoir que l'interaction cognitive a une influence sur la réflexion de l'humain (apprendre des choses aux enfants) alors que l'interaction physique repose sur le fait que la tâche accomplie par le robot et l'humain a un impact physique sur ce dernier (déplacer des objets). Il présente aussi les différents capteurs positionnés sur l'humain ou sur le robot et les 3 parties représentant la décomposition des principales branches de l'interaction homme-robot : la perception (la reconnaissance), la réalisation et l'évaluation.

2.2 Types d'interaction

Il existe plusieurs façons de classifier le type d'interaction entre l'humain et le robot. La méthode présentée par [16] classe l'interaction en fonction de l'application :

- Contrôle supervisé : l'humain commande, surveille l'action du robot mais lui laisse une certaine autonomie dans le cadre de tâches telles que chercher ou emballer des objets...etc
- Téléopération : lorsque l'humain commande totalement un robot à distance avec un retour de force, comme dans le cas des drones, que ce soit dans un milieu aérien, terrestre ou sous-marin.
- Véhicule autonome : l'humain est un passager qui détermine la destination, comme pour un trajet autoroutier, ferroviaire ou en mode pilote automatique dans un avion.
- Interaction sociale : c'est le domaine de la robotique de service qui traite des robots apportant de l'assistance aux gens ou faisant office d'hôtesse d'accueil.

Il existe des classifications basées sur un modèle d'échange d'informations entre le robot et l'humain. Il est possible de complexifier ce modèle avec une modélisation de la tâche et/ou de l'environnement : dans [22], ils intègrent la modélisation de 5 éléments (la tâche, l'utilisateur, le robot, l'environnement, et un ensemble d'interactions) et ce afin d'assurer la sécurité de l'humain et prendre en compte dans l'interaction les contraintes qu'apporte celui-ci. D'autres méthodes sont proposées par [23], la classification est basée sur :

1. la nature des informations captées par le robot ou échangées entre lui et l'humain. Les échanges peuvent concerner les données de l'environnement, celles du problème à résoudre, l'état courant de la tâche, ou des commandes directes. Les informations sont brutes et non traitées (position, image, son, commande), ou abstraites (résultat d'un traitement complexe, mouvement ou geste à interpréter, ordre vocal à reconnaître).
2. le type d'action appliquée par les effecteurs, soit sur l'environnement, soit de l'humain vers le robot, soit le contraire. L'échange d'informations et l'action ont parfois une frontière floue. L'action est définie comme l'application d'une force d'intensité variable et dont le résultat ne se résume pas au changement binaire d'un état.
3. le cheminement de l'information, de la perception de l'environnement jusqu'à l'action. Certaines informations vont systématiquement de l'environnement au robot, puis à l'humain ; d'autres de l'humain au robot, selon un ordre et un rythme propres à l'application.

Nous avons décidé de choisir une application (le jeu) qui sera classée comme étant une interaction physique (qui sollicite le robot et l'humain du point de vue effort physique). De même, selon la classification standard des interactions homme-robot [3], l'application peut être considérée comme un cas d'interaction cognitive (qui sollicite les capacités cognitives de l'humain) puisqu'elle sera entre le robot et un humain autiste (exemple : un enfant) pour l'aider à améliorer ses capacités cognitives et motrices. Ce qui démontre qu'utiliser le type d'applications pour classer l'interaction n'est pas un bon délimiteur puisque dans le cas de la tâche choisie, nous n'arrivons pas à trancher si elle est physique ou cognitive. Voilà pourquoi nous avons décidé de présenter une nouvelle méthode de classification basée sur l'impact qu'aura l'interaction sur l'humain. Avec cette nouvelle méthode notre application sera considérée comme une interaction cognitive.

2.2.1 Cognitive

Dans le cadre de l'interaction homme-robot, cognitive ou sociale, les robots doivent être socialement intelligents, en mesure de réagir de façon appropriée aux signaux affectifs et sociaux des humains afin de s'engager efficacement dans des communications bidirectionnelles. L'intelligence sociale permet au robot d'entrer en relation, de comprendre, d'interagir et de partager des informations avec des personnes dans des environnements réels centrés sur l'homme [24]. L'interaction cognitive (sociale) englobe toutes les applications nécessitant un effet cognitif du robot sur l'humain. Dans le cas de l'interaction cognitive, les mouvements effectués par le robot n'auront d'impact que sur le milieu environnant. Ceci concerne des applications telles que les thérapies dédiées aux adultes ou aux enfants [18], les interventions auprès des enfants autistes [19] ou encore les robots infirmiers socialement engagés livrant des médicaments à des résidents âgés en soins de longue durée [15].

L'interaction sociale homme-robot est donc essentiellement à but informationnel. L'action du robot est beaucoup plus dirigée sur l'environnement que sur l'humain. Ce type d'interaction ne nécessite pas de grandes capacités motrices (manipulation, préhension, ... etc.) de la part du robot. Toutefois, puisqu'un traitement complexe de l'information existe, cela sous-entend une certaine aptitude à effectuer une tâche complexe et donc une certaine indépendance, ce qui nous laisse penser que les tâches effectuées par le robot peuvent facilement dériver sur le concept de jouet intelligent, ou sur des applications industrielles [23]. Les interactions sociales (affectives) entre un humain et un robot peuvent avoir plusieurs impacts sur l'humain : motiver, soigner les

enfants atteints de TSA (Troubles du Spectre de l'Autisme), apporter une distraction (réduire la sensation d'isolement) et les aider à apprendre plus rapidement tout en s'amusant.

Motivation

Dans le cas d'une interaction collaborative à but cognitif, il ne sera pas question d'aide physique (celle-ci sera explicitée dans la section "cobotique" de l'interaction physique) mais d'une collaboration pour encourager l'humain à accomplir une tâche nécessitant un aspect physique. Un robot doit être capable d'identifier l'état affectif d'une personne afin d'éviter de mal interpréter les indices sociaux pendant la collaboration et d'améliorer la performance de l'équipe [25]. Certains roboticiens ont fait valoir que, dans ces cas, les gens collaborent plus naturellement et plus facilement avec des robots humanoïdes qu'avec des robots de type machine [26]. Ces robots sont aussi utiles pour la rééducation motrice des enfants. Les scientifiques et thérapeutes ont constaté que les enfants, en besoin de rééducation, ont souvent tendance à suivre au début les exercices qu'ils doivent effectuer et lorsque les premiers résultats apparaissent, ils les négligent complètement surtout s'ils doivent les effectuer seuls ; voilà pourquoi les auteurs de [27] proposent un cadre dans lequel le robot interagit avec un enfant pour le superviser quand il fait ses exercices de rééducation et pour l'encourager à continuer. Les auteurs de [28] développent un robot coach capable de montrer aux patients âgés des exercices de rééducation, de les regarder effectuer leurs exercices et de leur donner un retour d'information afin d'améliorer leurs performances et de les encourager. Dans [29], ils évaluent le taux de défaillance de la mémoire des personnes atteintes de déficience cognitive légère (troubles de la mémoire : Alzheimer). Le robot les suit et essaye de les aider à réduire le problème tout en les soutenant moralement.

Effet sur l'attention

Dans cette partie, nous présentons les travaux concernant l'impact cognitif et émotionnel qu'aura le robot sur l'humain notamment sur son attention. Les auteurs de [30] présentent des tests permettant de savoir si les robots humanoïdes, socialement interactifs, affectaient le contrôle de l'attention (c'est-à-dire la capacité cognitive primordiale) et ce en utilisant le "gold standard" des mesures d'attention [31] : la tâche de Stroop obligeant les individus à identifier la couleur avec laquelle un mot est imprimé au détriment du mot lui-même. Cependant, la question est de savoir si et quand la présence de robots humanoïdes sociaux renforce également le contrôle attentionnel chez l'humain. La présence de robots - quel que soit leur impact émotionnel positif (robot amical) ou négatif (robot inamical) - aurait dû entraîner un changement de performance chez l'humain, mais, la tâche de Stroop n'a montré qu'une amélioration de performances chez l'humain lorsqu'il est en présence d'un robot inamical (négatif). La présence de personnes potentiellement menaçantes améliore la sélectivité de l'attention portée à l'information pertinente au détriment des indices concurrents (dans la tâche Stroop, la couleur du mot imprimé au détriment du mot lui-même).

Effet curatif

L'imitation de l'affect peut être utilisée pour établir une coordination sociale entre les partenaires en interaction et encourager la coopération [32]. Par exemple : en utilisant un jeu d'imitation, les enfants apprennent à mieux contrôler leurs gestes-moteurs et leur attention conjointe en reproduisant les gestes que le robot effectue puis, celui-ci corrige le mouvement si nécessaire [33]. Une interaction enfant-robot basée sur un apprentissage où l'enfant, qui a des difficultés d'écriture, apprend au robot à écrire et lui dit comment améliorer sa calligraphie, ce qui l'aide

à améliorer la sienne par imitation, est présentée par [34]. On peut savoir si un enfant est atteint ou non de TSA grâce à un ensemble de métriques, basées sur l'analyse des mouvements du corps et de la tête, de la magnitude et des directions du regard (gauche, droite et devant) et des énergies cinétiques capables de décrire le comportement des enfants pendant la tâche avec le robot et ce en les comparant aux résultats d'enfants normaux [35]. L'étude de [36] s'est concentrée sur une intervention à domicile d'une durée d'un mois visant à améliorer les habiletés de communication sociale d'enfants atteints de TSA âgés de 6 à 12 ans et ce à l'aide d'un robot social autonome qui soutient la croissance sociale et cognitive en utilisant des moyens sociaux plutôt que physiques [37], ce qui a permis de savoir l'effet cognitif du robot sur les enfants [38].

Distraction et apprentissage

Les enseignants, avec l'aide des scientifiques, ont testé de nouvelles méthodes d'apprentissage pour enfants [39], en particulier faire intervenir un robot humanoïde comme enseignant : il prend connaissance des transparents et de la leçon que l'enseignant a à enseigner puis se charge de l'explication en classe en faisant des gestes de "Body Language" et veille à garder les enfants concentrés, sinon il commence à leur raconter des histoires ou à danser pour attirer leur attention. Les résultats démontrent que les enfants considèrent le robot humanoïde comme ami et non comme professeur, ce qui leur a permis d'acquérir plus aisément les connaissances enseignées [40]. Cet apprentissage social est présent dans les interactions en classe avec les pairs et enseignants, mais fait souvent défaut à la maison. Si les robots sociaux peuvent transformer l'activité d'apprentissage à domicile en une activité sociale dynamique et imprégner leurs interlocuteurs d'un sentiment de camaraderie, appelé "connectivité sociale" [41], l'apprentissage social humain-robot pourra être efficace pour les enfants qui développent leurs compétences en lecture et en alphabétisation. Le travail avec un robot social peut de même améliorer l'alphabétisation préscolaire [42] et le vocabulaire [43] et l'amélioration de l'expérience sociale (fournir une voix plus expressive et naturelle) peut donner de meilleurs résultats d'apprentissage [44].

Le robot peut accomplir des tâches aidant les gens peu importe leur âge ou leur condition physique. De même, trouver un objet que l'utilisateur a perdu chez lui ou au travail et ce en faisant du SLAM avec des robots pour vérifier s'ils sont capables de se localiser et de faire une carte d'intérieur avec des pièces, chambres et meubles en utilisant des codes barres 2D collés sur les murs ("Landmarks") et des images (caméra) pour visualiser l'environnement [45]. Il peut aussi distraire les enfants en jouant avec eux à cache-cache tout en usant du même moyen. Enfin, il peut distraire et aussi secourir les personnes âgées en leur parlant ou en prévenant le médecin pour le tenir informé de leur état ou leur donner des conseils en attendant des secours [46].

2.2.2 Physique

L'interaction physique regroupe toutes les applications qui nécessitent que le robot ait un contact physique avec la personne ou les tâches exigeant un certain effort physique de la part de l'humain qui sera réduit grâce à sa collaboration avec le robot. Dans ce cas, les mouvements effectués par le robot auront un impact sur l'humain et/ou le milieu environnant. L'interaction physique entre l'homme et le robot (pHRI) a été largement utilisée dans de nombreux domaines tels que les thérapies de réadaptation (e.g. le robot guide un patient le long d'un couloir ou chemin [47]) et les tâches de manipulation coopérative. Comme les dispositifs robotiques sont appliqués à des problèmes qui dépassent le cadre traditionnel de la fabrication et de l'industrie, nous constatons que l'interaction entre les robots et les humains, en particulier l'interaction phy-

sique, est devenue un domaine en plein essor. Considérons l'application de la robotique dans le domaine de la santé où l'on trouve des dispositifs télérobotiques dans la salle d'opération facilitant les interventions chirurgicales d'habileté, des exosquelettes dans le domaine de la réadaptation comme aide à la marche et des appareils d'aide au mouvement des membres supérieurs, et même des membres robotisés qui sont physiquement intégrés aux amputés qui cherchent à retrouver leur indépendance et leur mobilité. Dans chacun de ces scénarios, le couplage physique entre l'homme et le robot, souvent appelé interaction physique entre l'homme et le robot (pHRI), facilite de nouvelles capacités de performance humaine et crée une occasion d'explorer le partage de l'exécution et du contrôle des tâches entre l'homme et le robot [48]. Au-delà de la proximité, la nature de l'interaction physique peut être comprise dans le contexte des tâches et des rôles assumés par le robot et les acteurs humains dans un scénario pHRI. La plupart des travaux sur pHRI peuvent généralement être classés en trois grandes catégories d'interaction : de soutien, de collaboration et de coopération. Classées de cette façon, nous soulignons que ces interactions sont marquées par une fréquence croissante ainsi que par la nécessité d'un contact physique avec le robot et par le niveau de proximité avec l'utilisateur (cf. Figure 2.1).

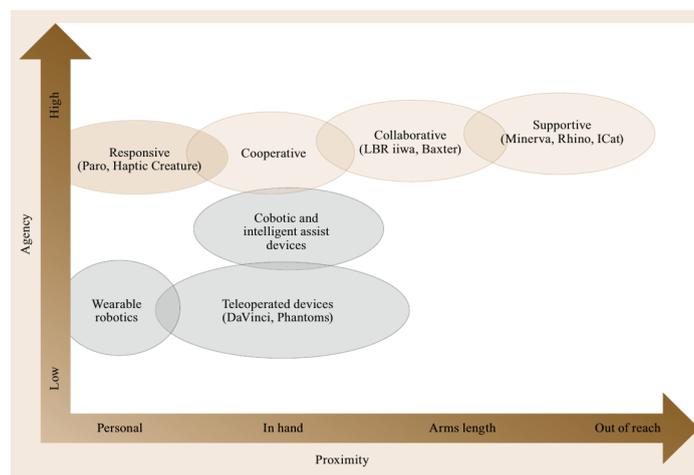


FIGURE 2.1 – Schéma de classification du pHRI, par proximité de l'interaction et de l'agence (autonomie disponible) du robot [14]

Soins et soutien physique

Les mouvements humains peuvent être affectés par différents déficits moteurs (contrôle cervical) comme les troubles musculo-squelettiques (TMS) dans un contexte professionnel, ou dans les principales maladies chroniques cérébro-vasculaires qui affectent gravement des personnes ou celles causées par un accident vasculaire cérébral (AVC) lesquelles touchent un patient sur cinq [49]. Pour diminuer les TMS, les exosquelettes des membres supérieurs peuvent être utilisés [50]. Dans l'optique de soigner les TMS, il existe une demande importante de robots de rééducation pouvant être utilisés pour améliorer la motricité des membres supérieurs et simplifier le processus de rééducation. Beaucoup d'applications de robots d'assistance ou de rééducation existent de nos jours [51]. Les recherches montrent qu'un traitement thérapeutique consistant en des mouvements répétitifs intensifs des membres affectés est l'une des approches les plus efficaces pour restaurer en partie la capacité motrice [52].

Dans le cas d'interaction de l'exosquelette, le couplage entre l'humain et le robot est direct. La partie motrice joue le rôle principal tandis que l'interface (dispositif permettant la communi-

cation entre deux éléments d'un système informatique, le plus souvent un PC) tient un rôle secondaire. L'objectif de cette interaction est de réduire au maximum l'effort fourni par l'humain au cas où l'exosquelette l'aide à se mouvoir ou l'annuler totalement au cas où l'exosquelette sert de prothèse ; pour cela l'action du robot doit être simultanée et proportionnée à l'humain. Dans cette interaction, peu d'informations sur l'environnement sont relevées par le système. Le retour de force est la seule information (valeurs articulaires) nécessitant des capteurs pour la mesurer et informer l'exosquelette de son appui au sol.

Collaboration

Collaboration dans le cadre non industriel

Dans le cas de tâches nécessitant un contact physique entre l'humain et le robot, on évoque la collaboration entre les deux aussi bien que l'interaction. Des solutions robotiques collaboratives faciles à utiliser, où les travailleurs et les robots partagent leurs compétences, font leur entrée sur le marché, devenant ainsi la nouvelle frontière de la robotique industrielle. Ils permettent de combiner les avantages des robots qui jouissent de niveaux élevés de précision, de vitesse et de répétabilité, avec la flexibilité et les compétences cognitives de l'homme. Cependant, pour parvenir à une collaboration efficace entre les deux, plusieurs défis doivent être relevés [53] :

- L'interaction sûre doit être garantie pour ne pas nuire aux humains en contact avec le robot en mouvement.
- Les interfaces utilisateurs intuitives doivent être correctement conçues afin que les opérateurs humains puissent programmer et interagir avec le robot pour tirer pleinement partie des compétences humaines.
- Trois thèmes clés se dégagent de ces scénarios de contrôle partagés, à savoir la détection d'intention, l'arbitrage et le retour d'informations.

Collaboration dans le cadre industriel

La cobotique est le domaine scientifique de l'étude des systèmes collaboratifs homme-robot (systèmes cobotiques) [23]. Cobot est un néologisme formé des termes "collaboratif" et "robot" désignant des robots collaboratifs ayant le même espace de travail que l'humain [54]. Les cobots n'ont la plupart du temps qu'un seul degré de liberté mécanique quelle que soit la dimension de leur espace de travail [55]. La cobotique désigne la conception, la production, l'étude et l'utilisation de systèmes de travail dans lesquels un ou plusieurs opérateurs interagissent avec un ou plusieurs robots dans un espace partagé ou à distance [56]. Elle est à la frontière du domaine de la robotique, la cognitive et l'ergonomie (cf. Figure 2.2) ; ses problématiques sont : la perception, les types d'interaction, l'efficacité, l'intensité, la prise de décisions, le choix de l'action à effectuer. Une revue approfondie de la collaboration homme-robot en milieu industriel est présentée dans [53], avec un accent sur les questions liées à l'interaction physique et cognitive. Ce type de système (les cobots) est très présent dans l'industrie. La Cobotique vise à assurer la flexibilité de la chaîne de production, à améliorer les performances, à réduire le nombre de maladies et d'accidents du travail et à garantir l'employabilité des opérateurs experts. Pour y parvenir, il faut tirer partie de la complémentarité des qualités humaines (flexibilité, prise de décision, expertise) et robotiques (force, endurance, vitesse, précision, répétabilité) [57].

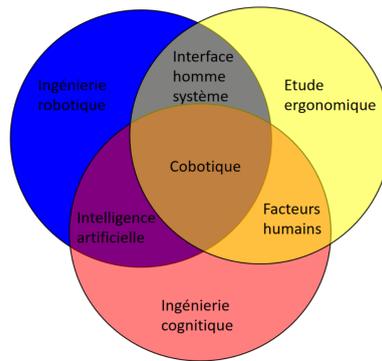


FIGURE 2.2 – Domaines utilisés pour résoudre les problématiques de la cobotique [23]

Aide

Dans les interactions de type robotique de service, où le robot effectue les tâches pour ménager l'humain sans grand chose à faire, nous regroupons les interactions dans lesquelles le robot ne fait pas partie intégrante de la performance centrale d'une tâche, mais fournit à l'homme les outils, le matériel et les informations nécessaires pour optimiser la performance ou les objectifs de la tâche humaine : ex. les robots guides de visite dans les musées, ceux d'aide aux achats pour aider les personnes âgées [11], ceux de soins à domicile [14] et ceux de livraison de courrier dans les bureaux d'une agence [17]. Dans ce contexte, l'pHRI concerne la sécurité, c'est-à-dire la prévention et l'atténuation des effets de contacts ou de collisions inattendus.

2.3 Disciplines de l'interaction Homme-Robot

Dans cette partie, nous présentons les étapes nécessaires pour créer et évaluer une interaction homme-robot. Nous montrons des pistes permettant de faire la reconnaissance (détection de personnes, de lieux, de l'activité de l'humain, d'intentions et d'objets), le contrôle et l'évaluation de l'interaction selon différents critères (ergonomie, effort, confort, naturel, sociabilité et confiance). Les notions de reconnaissance, de contrôle et d'évaluation dont nous parlons n'ont pas, pour la plupart des cas, été développées dans le cadre d'une interaction homme-robot.

2.3.1 Perception (reconnaissance) pour l'interaction

Le robot doit être capable de percevoir, de distinguer ce qui l'entoure et de faire le lien entre les données que lui renvoient les capteurs dont il est équipé et son environnement. Il doit surtout être capable de trouver l'objet autour duquel il y a une interaction avec l'homme sans se tromper d'opérateur quand plusieurs personnes se trouvent dans son environnement.

Détection d'intentions

La détection d'intentions permet au robot de connaître des aspects de l'action planifiée de l'humain et de pouvoir participer, de manière appropriée, à sa réalisation [3]. Nous allons présenter la détection d'intentions selon la méthode utilisée par le robot pour y parvenir : capteurs ou apprentissage. Trois étapes sont nécessaires pour transmettre de façon unidirectionnelle l'intention de l'humain au robot :

- L'intention de l'utilisateur doit être définie.
- La modalité selon laquelle les informations d'intention sont mesurées par le robot doit être déterminée, ce qui est nommé "méthode de mesure d'intention".
- Savoir comment cette mesure doit être comprise par le robot en tant que représentation de l'intention, et comment elle doit être intégrée à la structure de contrôle du robot [48]. Cette étape est nommée "interprétation de l'intention".

Capteurs

Des chercheurs s'intéressent à la détection de l'intention à partir de données haptiques issues de capteurs tactiles comme dans [58]. Dans [59], ils proposent un modèle où l'information haptique, fournie par le toucher et la proprioception, permet aux individus en interaction d'estimer l'objectif du mouvement du partenaire et de l'utiliser afin d'améliorer leurs propres performances motrices. D'autres scientifiques ont essayé d'imiter les gestes humains pour connaître l'intention du mouvement et surtout de pouvoir prédire le suivant. Une méthode imitant les gestes humains à l'aide du FMRI (cf. Annexe A) est exposée par [60]. Quant à [61], il présente un moyen d'imiter les gestes du haut du corps humain en utilisant un capteur Kinect pour savoir la position des différentes articulations. Enfin, des chercheurs ont choisi de savoir l'intention de la personne en interaction avec le robot à l'aide du BCI (cf. Annexe A) [62]. Dans [63], l'intention de mouvement est détectée conformément à la corrélation entre les potentiels corticaux liés au mouvement (MRCP = movement related cortical potentials) mesurés par une EEG (cf. Annexe A). Dans [48], la détection d'intention est réalisée par le traitement des mesures, obtenues à l'aide de capteurs de force, avec un modèle mathématique de la tâche de manipulation coopérative quand un humain et un robot collaborent pour déplacer une table volumineuse [64].

Apprentissage

Les chercheurs détectent l'intention de l'humain via des réseaux de neurones à partir de capteurs EEG. Dans [65], ils utilisent la méthode de décodage (ou classification) des entrées appelée analyse discriminante linéaire (LDA) et des machines à vecteurs de support (SVMs), technique courante de déduction de données neuronales. Une extension intelligente de l'algorithme LDA pour obtenir plus qu'une simple sortie binaire move/rest est développée par [66]. Les mêmes 2 classes (déplacement et arrêt) sont utilisées ; et pour chaque nouveau point de données les probabilités a posteriori de chaque classe sont calculées fournissant ainsi une sortie continue comprise entre 0 et 1.

Détection de personnes

Le robot doit détecter les personnes l'entourant, distinguer celle avec qui il entre en interaction, puis la suivre. Les auteurs de [67] proposent une approche permettant au robot de ré-identifier une personne, par une approche d'apprentissage, parmi les différentes images prises par les caméras dont il est équipé. Il existe aussi une méthode basée sur l'apprentissage de la détection de l'âge, point discriminant capable de distinguer les gens [68] ou procédant en passant par la détection de visages à l'aide d'une méthode basée sur la transformée de Gabor [69].

Détection de lieux

La reconnaissance visuelle de lieux a été récemment l'objet d'une attention considérable [70], en vision par ordinateur [70, 71] et dans les communautés robotiques [72, 73], motivées par des applications en conduite autonome [73], en réalité augmentée [74] ou en géo-localisation d'imagerie archivistique [75]. Son importance pour l'interaction homme-robot se révèle quand le robot reconnaît l'endroit où il se trouve et s'adapte aux règles sociales. Mais comment reconnaître un même coin de rue en ville ou à l'échelle d'un pays alors qu'il peut être capté sous des éclairages différents ou changer d'aspect dans le temps ? La question fondamentale est de savoir la représentation appropriée d'un lieu suffisamment riche pour distinguer des lieux d'apparence similaire tout en étant compact pour représenter des villes entières ou des pays [70].

Reconnaissance de l'activité humaine

La reconnaissance de l'activité humaine (HAR = Human activity recognition) joue un rôle important dans la vie quotidienne des gens. En robotique, on peut acquérir des connaissances approfondies de haut niveau sur l'activité humaine à partir des données caméras. Elle est utile dans le cas de l'interaction pour que le robot détecte le type d'activité qu'effectue l'humain et puisse s'y adapter. Les applications de HAR comprennent l'analyse du comportement à domicile [76], la surveillance vidéo [77], l'analyse de la démarche [78] et la reconnaissance gestuelle [79]. Il existe deux types de HAR : par vidéo et par capteurs [80]. Celle basée sur la vidéo analyse les vidéos ou les images contenant des mouvements humains provenant de la caméra ; tandis que celle basée sur le capteur se concentre sur les données de mouvements provenant de capteurs intelligents tels qu'un accéléromètre, un gyroscope, Bluetooth, des capteurs sonores ...etc.

Détection d'objets

La détection d'objets permet au robot de reconnaître des parties de l'environnement c'est-à-dire interagir avec l'humain. Elle repose sur des méthodes d'apprentissage (machine learning) notamment du deep learning [81]. Le réseau de neurones extrait des caractéristiques clés à partir d'images ou données haptiques issues de différents capteurs pour détecter l'objet recherché, trouve un modèle localisant l'objet dans une image ou isolant les données lui correspondant (des données haptiques), applique des filtres aux images pour traquer le mouvement de l'objet d'une image à l'autre et reconnaît les objets à partir de leur définition relevée d'un dictionnaire.

Images

Les méthodes de détection peuvent être divisées en 3 catégories [82] : celles basées sur l'extraction de caractéristiques [83, 84], celles basées sur la recherche de modèles [85, 86] et celles basées sur la détection de mouvements [87]. Les méthodes d'apprentissage profond font partie du premier groupe. La façon la plus directe est d'appliquer un classificateur profond aux régions d'intérêt générées par l'approche extensive des fenêtres coulissantes [88, 89]. Néanmoins, la détermination de la position, de la taille et de l'échelle de l'objet peut être intégrée au réseau neuronal [88, 90] en ajoutant des couches de type spécial. Une méthode de détection de cibles (ECCSR-SVM) balaie le problème du facteur d'échelle résultant de la distance entre l'objet et la caméra dont est équipé le robot quand on préfère se servir de la caméra au lieu des capteurs tactiles ou encore si l'on veut faire une fusion de données, ce qui est présenté par [91].

Données Haptiques

La reconnaissance visuelle d'un objet exige sa présence dans le champ de vision de la caméra, quoique limitée par des contraintes d'éclairage et des occultations visuelles. Pour vaincre ces contraintes, on utilise la reconnaissance visuelle avec des sources d'informations supplémentaires comme les données haptiques (le toucher) qui aident à surmonter ces contraintes et fournissent des informations supplémentaires quant à la forme, la taille, la texture et la conformité d'un objet, ce qui améliore la reconnaissance [92]. Les méthodes d'apprentissage à partir d'un dictionnaire offrent une description sémantique de l'objet. Un réseau de neurones "zero-shot learning" apprend à distinguer les objets, sans être au préalable entraîné avec des données labellisées et les classe [93], en comparant leur définition, obtenue à l'aide du dictionnaire, aux données haptiques que le robot reçoit de ses capteurs tactiles [92].

2.3.2 Réalisation (contrôle/commande)

Les dernières années, on a assisté à l'apparition de "robots d'assistance", des systèmes robotiques pour la téléopération, des systèmes de conduite assistée dans les voitures, des chaises roulantes robotiques pour augmenter la mobilité des personnes ayant des déficits physiques ou cognitifs, des postes de travail avec rétroaction haptique qui peuvent être utilisés pour former les chirurgiens, des exosquelettes robotiques pour augmenter les capacités de force de l'utilisateur et des robots de réadaptation pour augmenter la quantité et l'intensité de la thérapie physique après un AVC [94]. Dans cette section, nous allons présenter différentes méthodes qui peuvent contrôler le robot afin qu'il effectue correctement la tâche qui lui est attribuée.

Comprendre le contrôle sensori-moteur chez l'humain

Savoir comment les humains contrôlent l'interaction motrice avec un partenaire peut aider à concevoir des stratégies efficaces d'interaction homme-robot (HRI). De plus, un robot qui adapte son rôle et son action de la même manière qu'un partenaire humain peut être plus intuitif pour un opérateur humain, ce qui nécessite moins d'effort pendant l'utilisation. Pour cela, nous allons voir comment le cerveau humain ressent les différents membres du corps et gère la partie contrôle pour les bouger. Les scientifiques ont pu démontrer, à l'aide d'expériences et de simulations, que le cerveau humain arrive à doser l'effort nécessaire au mouvement lors d'une tâche de partage de force avec non seulement les commandes motrices mais aussi les signaux haptiques renvoyés [95]. Le fait d'avoir un outil en main aura une influence sur la façon dont le cerveau devra gérer les efforts, ce qui est démontré par deux expériences dans [96].

Contrôle des robots

Dans la revue [97], les solutions de la robotique traditionnelle sont comparées, impliquant des articulations rigides, à celles de la littérature récente. À préciser que le contrôle conforme (tout mouvement du robot au cours duquel la trajectoire de l'effecteur final est modifiée, voire générée, sur la base des informations fournies par les capteurs en ligne) est fondamental lorsqu'il s'agit d'environnements non structurés, comme dans le cas de l'interaction homme-robot. De nos jours, l'intérêt pour le contrôle conforme est lié aux approches où le contrôleur façonne l'impédance mécanique du système robotique, c'est-à-dire la relation dynamique entre la position ou la vitesse du robot et les forces externes. Façonner l'impédance mécanique permet d'interagir en toute sécurité avec l'environnement inconnu car, au lieu de contrôler une position

ou une force, nous contrôlons la puissance transférée à l'environnement, grâce à la paire énergétique force-vitesse. Une description des technologies de contrôle de conformité appliquées aux articulations rigides et souples à base de moteurs électriques est essentiellement proposée.

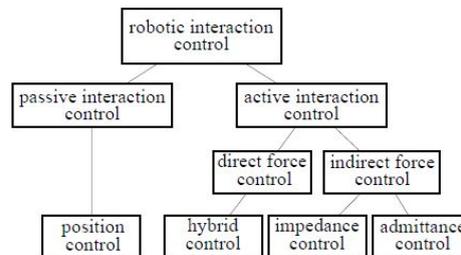


FIGURE 2.3 – Classification du contrôle d'interaction. Ici, les termes direct et indirect font référence à la présence ou non d'un contrôleur de force pure. [97]

Le contrôle d'une tâche dans un environnement ayant des contraintes cinématiques et/ou dynamiques est généralement appelé contrôle d'interaction. Dans la littérature sur la robotique traditionnelle, il existe plusieurs approches pour résoudre ce problème lesquelles sont schématiquement organisées dans la Figure 2.3. La principale distinction est entre le contrôle d'interaction active et passive. Dans l'interaction active la conformité est entièrement due au contrôle alors qu'en interaction passive, elle l'est à la mécanique. Le contrôle d'interaction passive est en pratique réalisé grâce à la commande en position d'un manipulateur. Le contrôle actif est divisé en sous-catégories directes et indirectes. La catégorie de contrôle direct de la force comprend des algorithmes qui contrôlent séparément la force et la position des sous-espaces, appelés contrôle hybride position-force [98]. D'autre part, le contrôle indirect de la force conduit à des techniques de mise en forme de l'impédance ou de l'admittance du robot. Ce concept a du succès lorsque le robot est en espace libre mais aussi lorsqu'il est en contact avec l'environnement, ce qui lui permet même d'être utilisé dans des environnements inconnus. Dans la littérature récente sur les articulations souples, une approche mixte active / passive est très courante et ne peut être colocalisée dans la taxonomie de la Figure 2.3. Pour cette raison, ils ont proposé la classification illustrée à la Figure 2.4, où le nœud parent est nommé contrôle conforme ; cela a le même sens que le contrôle indirect de la force, mais ce terme est plus largement utilisé.

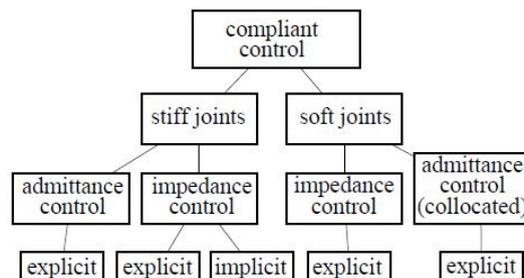


FIGURE 2.4 – La nouvelle classification du contrôle d'interaction. Le nœud explicite/implicite indique la présence ou l'absence d'un retour de force. Notons que le contrôle d'interaction passive et le contrôle hybride force/position de la Figure 2.3 ne sont pas inclus car ils visent à contrôler la position ou la force indépendamment et non dans leur relation. [97]

Distribution de rôles

Les applications, citées en introduction de cette section, exigent une interaction physique continue, ou du moins prolongée, entre un robot et son utilisateur humain vers un but commun, au cours de laquelle ils utilisent des rôles similaires ou complémentaires. Dans certains cas, ces rôles sont établis a priori en conséquence directe de la nature de la tâche. Dans d'autres, la même tâche permet une variété d'affectations de rôles : par exemple, deux sujets portant une table lourde pourraient collaborer en tant que partenaires égaux, coopérer en tant que dyade maître-esclave (c'est-à-dire qu'un sujet fournirait la majeure partie de la manœuvre sous la supervision de l'autre), ou même travailler comme concurrents si chacun d'eux veut amener la table à un autre endroit. Dans de telles tâches, les robots auraient la capacité de négocier et d'adapter leur propre rôle à l'objectif global et au comportement de leur partenaire humain [94].

2.3.3 Évaluation

Cette partie sert à évaluer si l'interaction est non seulement réussie dans le sens où le robot a effectué ce qu'il devait faire mais aussi si, en effectuant la tâche, il a pu mettre à l'aise l'opérateur. Il y a plusieurs critères qui permettent d'assurer ce point. Nous allons expliciter les critères d'évaluation les plus importants. Nous nous concentrons sur les critères en lien avec l'interaction, nous ne présenterons donc pas les critères en lien avec la tâche elle-même comme par exemple la durée que met le robot à effectuer une tâche, la robustesse, ... etc.

Ergonomie

L'ergonomie est la science qui étudie la relation entre l'Homme et ses moyens, méthodes et milieux de travail¹ et qui applique ces connaissances à la conception du système de travail de manière à ce qu'il soit le plus confortable, le plus sécurisant et le plus efficace possible² en termes d'accomplissement de tâches. Les scientifiques ont voulu inclure les positions de travail ergonomique à l'interaction collaborative homme-robot pour que le travail soit le moins pénible possible pour l'humain et qui lui permette ainsi d'éviter les TMS. Pour cela, ils ont, d'un côté, voulu appliquer les méthodes bien connues d'évaluation de postures de l'ergonomie comme : "REBA" (Rapid Entire Body Assessment) [99], "RULA" (Rapid Upper Limb Assessment) [100], "dREBA" (Differentiable REBA) [101] ou celle proposée dans [102] qui s'adapte à n'importe quel élément évalué dans des feuilles de calcul ergonomiques standard à l'aide de leur compatibilité avec une taxonomie des activités composée de quatre niveaux.

Effort

Les scientifiques n'ont pas voulu suivre des positions prédéfinies mais plutôt s'adapter à la situation où se trouve l'humain. Ils ont pensé apporter des changements dans le fonctionnement de l'algorithme ou du mécanisme utilisé de telle sorte que l'effort physique que l'humain produit pour effectuer une tâche soit réduit [103]. Ils diminuent la surcharge du moment statique du couple articulaire d'un partenaire humain lors de l'exécution de tâches partagées avec un robot grâce au calcul du centre de pression alors que dans [104], ils mesurent à chaque instant l'EMG (cf. Annexe A) qui indique la mesure de l'activité du muscle afin de minimiser les effets d'une charge externe sur les articulations du corps humain. Certains travaux se focalisent sur les trajectoires nécessitant le moins d'effort de la part de l'humain durant l'interaction [105].

1. Extrait de la définition adoptée par le VIème Congrès international d'ergonomie (1969)

2. Extrait de la définition de l'ergonomie retenue par la Société d'Ergonomie de Langue Française

Confort

La notion de confort dénote l'absence de gêne et de stress pour les humains qui interagissent avec un ou plusieurs robots [106]. Il faut d'abord identifier les causes d'inconfort chez l'humain en fonction du comportement du robot pour y remédier : la principale source est que le robot doit garder une certaine distance entre lui et l'humain, l'autre est la vitesse à laquelle il avance. Pour régler le premier problème, on crée un espace personnel virtuel [107] autour de chaque individu lequel le robot ne doit pas pénétrer; ce concept est nommé "proximité" [108]. La distance choisie par les humains en fonction de leur relation avec l'autre est explicitée dans la Table 2.1. Elle diminue au fil du temps d'après [109]. Il faut aussi observer la direction du robot s'approchant de l'humain [110]. De même, dans [111], une étude démontre comment les personnes interagissent avec le regard d'un robot fixe mais pouvant bouger la tête pour établir ou éviter le regard d'une personne. Les auteurs en ont déduit qu'il doit faire une bonne première impression, et si la réaction du sujet est difficile, l'interaction ne doit pas alors être trop intense. Quant à la solution au deuxième problème, le robot ne doit avancer ni trop rapidement ni trop lentement. Les scientifiques, dans [112], ont déterminé que 0,5 m/s semble une vitesse acceptable pour les personnes en interaction avec le robot.

Distance/Relation	Valeur	Destiné à
Distance intime	0 - 45 cm	Faire la bise, toucher, murmurer, ...etc.
Distance personnelle	45 - 120 cm	Amis
Distance sociale	1,2 - 3,6 m	Connaissances et étrangers
Distance publique	> 3,6 m	Parler en public

TABLE 2.1 – Les distances que gardent les individus entre eux en fonction de leur relation [108]

Naturel

La similitude entre les robots et les humains (ou les animaux) dans le comportement est la notion du "naturel" [106]. Pour cela, il faut attribuer aux robots des traits humains, c'est ce qu'on appelle "l'anthropomorphisme" (l'attribution de traits humains à des entités non humaines) [113]. La probabilité qu'un observateur humain interprète le mouvement du robot [114] est plus élevée si le robot se déplace de manière "naturelle". Il faut que le robot puisse se comporter comme l'humain pour être plus facilement accepté par celui-ci [115]. Un comportement naturel dépend de la trajectoire du mouvement parcouru et la vitesse avec laquelle le robot évolue. Il faut bien en tenir compte surtout si le robot est dans un environnement où il y a beaucoup d'humains ou/et d'autres robots [116] puisqu'il cherchera à ne pas être retardé ni bloqué et ce en évitant les trajectoires humaines prévues [117, 118]. La similitude du comportement du robot n'a aucun impact négatif sur l'humain, bien au contraire, toutefois, il peut y avoir un phénomène d'inconfort chez l'humain dans le cas où le robot devient trop humain du point de vue apparence. Ce phénomène se nomme "La Vallée Dérangante (Uncanny Valley)" [119].

Sociabilité

C'est l'adhésion du robot à des conventions culturelles explicites [106]. L'inconfort est généré par la violation des règles sociales plutôt que par le non-respect de la zone de confort dans le cadre d'une interaction sociale [111, 112, 116]. Pour l'éviter, il faudrait ajouter au comportement du robot des caractéristiques tenant compte du protocole social : serrer la main, saluer une fois par jour, frapper à une porte avant d'entrer, etc. Si l'on n'ajoute pas de contraintes pour

appliquer les règles sociales, le robot peut les violer tout en respectant les règles de confort, d'ergonomie et de comportement naturel : s'il y a une file d'attente, il peut, pour respecter la distance minimale entre lui et les personnes autour, se mettre en tête de file, ce qui peut les exaspérer. Dans [120], ils ont incorporé aux règles sociales de base d'autres normes sociales : éviter de passer à travers un groupe de personnes, devancer une personne par la droite, ...etc.

2.4 Conclusion

Nous avons présenté les principales méthodes de classification citées dans l'état de l'art ainsi que celle que nous avons proposée et qui est résumée par la Figure 2.5. Nous avons aussi exposé les techniques (cf. Figure 2.6) les plus répandues pour faire la perception, le contrôle et l'évaluation afin de mettre au point une interaction homme-robot, la plus optimale possible.

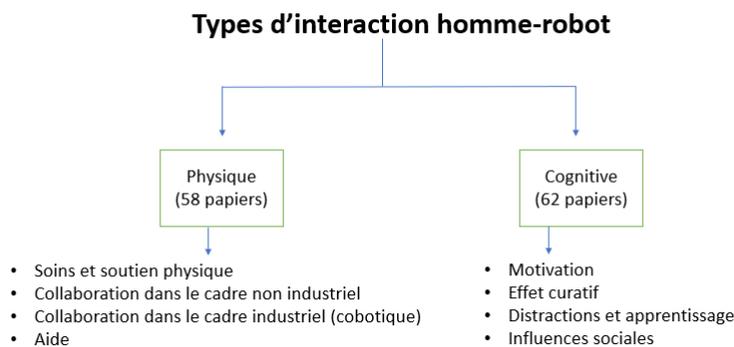


FIGURE 2.5 – Résumé de la méthode que nous avons proposée pour la classification de l'interaction homme-robot avec le nombre d'articles que nous avons lus pour faire la revue

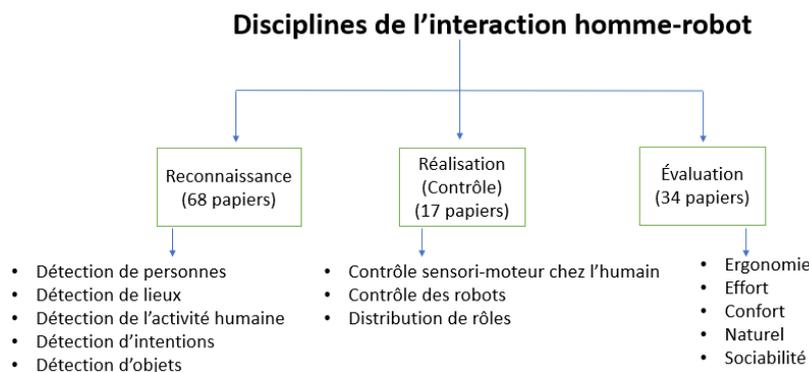


FIGURE 2.6 – Résumé des disciplines nécessaires pour mettre au point une interaction optimale homme-robot avec le nombre d'articles que nous avons lus pour faire la revue

Chapitre 3

Plateforme expérimentale

3.1 Introduction

Dans ce chapitre, nous allons exposer le matériel mis à notre disposition pour faire la manipulation et réaliser l'application (le cadre mis en place pour faire l'interaction homme-robot). Nous allons commencer par faire une présentation rapide des articulations, des capteurs, et des dimensions de Nao ce qui permettra dans un second temps de savoir ses limitations. Ensuite nous allons présenter ROS et expliquer pourquoi nous avons décidé d'utiliser ce logiciel. Enfin, nous allons énoncer le jeu sélectionné pour faire l'application ainsi que les raisons de ce choix.

3.2 Nao

3.2.1 Présentation du robot

Nous avons choisi Nao, le robot humanoïde créé par Aldebaran Robotics (devenu SoftBank Robotics) pour effectuer les tests de notre interaction homme-robot. Nao est particulièrement bien adapté aux applications HRI en raison de sa conception, de ses spécifications matérielles, de ses capacités de programmation et de son coût faible. En effet, Nao peut se lever, marcher, se promener, danser, jouer au football, s'asseoir, reconnaître et saisir des objets simples, détecter et identifier des personnes, localiser des sons, comprendre des mots prononcés, s'engager dans des dialogues simples et ciblés et faire la synthèse de la parole. Ceci est rendu possible grâce à la structure articulée à 24 degrés de liberté du robot (corps, jambes, pieds, bras, mains, tête, ... etc.) et aux capteurs dont il est équipé ainsi qu'à son matériel informatique embarqué et ses logiciels également embarqués.

Ce robot a deux inconvénients majeurs. Premièrement, les ressources informatiques embarquées sont intrinsèquement limitées, ce qui rend difficile de mettre en œuvre des algorithmes sophistiqués de vision par ordinateur et d'analyse du signal audio requis par des tâches interactives avancées. Deuxièmement, la programmation de nouvelles fonctionnalités de robots implique actuellement le développement de logiciels embarqués, ce qui est une tâche difficile en soi, nécessitant des connaissances spécialisées. Raisons pour lesquelles dans [121], ils ont développé une architecture logicielle distribuée qui tente de surmonter ces deux limites.

Nous avons utilisé la 6ème génération de Nao (cf. Figure 3.1). Cette nouvelle version a été améliorée à plusieurs niveaux : la puissance, le mouvement, la robustesse, l'audition (audio),

la vision ainsi que la sécurité. Sa carte mère couplée avec la version de Naoqi 2,8 lui donne de meilleures performances du point de vue du temps d'initialisation et de mise à jour mais également l'Ethernet et le WIFI.



FIGURE 3.1 – Nao 6, 6ème génération du robot Nao créé par Aldebaran Robotics devenu Soft-Bank Robotics [1]

3.2.2 Spécifications de Nao 6 [1, 2]

- Dimensions : 574x 311x 275 mm (cf. Figures 3.2 et 3.3)
- Poids : 5.48 Kg
- Autonomie : 60 minutes en mode "Active" et 90 minutes en mode "Normal"
- 25 moteurs à courant continu (25 Degrés de liberté) pour le mouvement des organes par l'intermédiaire de hacheurs
- Processeur : Intel Atom E3845
- RAM : 4GB DDR3
- Système : Linux embarqué (Distribution Gentoo)
- Compatible OS : Windows, Mac OS, Linux
- Langages de programmation :
 - Embarqué : C++ et Python
 - A distance : Java et Matlab
- Capteurs (cf. Figure 3.4) :
 - Vision : 2 caméras OV5640 2592x1944 (sur le front et la bouche)
 - 4 microphones (pour écouter au niveau de la tête : 1 devant, 1 derrière et 1 de chaque côté)
 - 2 émetteurs récepteurs infrarouges pour détecter (au niveau des yeux)
 - 2 sonars à ultrasons (2 émetteurs et 2 récepteurs au niveau du torse)
 - 1 centrale inertielle (accéléromètre 3 axes et gyromètre 2 axes au niveau du torse)
 - 9 capteurs tactiles (3 sur la tête et 3 au dos de chaque main)
 - 8 capteurs de pression FSR pour faciliter son équilibre (4 sous chaque pied)
 - 2 détecteurs mécaniques pour détecter les chocs (1 au bout de chaque pied)
 - 32 codeurs rotatifs magnétiques sans contact pour connaître la position relative de chaque organe (tête, bras, avant-bras, cuisses, jambes, pieds).
 - 2 haut-parleurs pour s'exprimer (au niveau des oreilles)
 - 2 mécanismes de préhension (au niveau des doigts) par l'intermédiaire de hacheurs
- Connectivité : Ethernet et WIFI

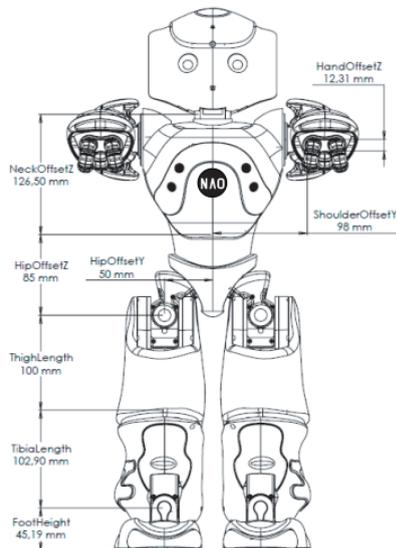


FIGURE 3.2 – Dimensions de Nao 6 (vue de face) [1]

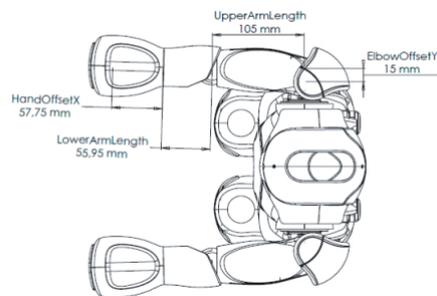


FIGURE 3.3 – Dimensions de Nao 6 (vue d'au- dessus) [1]

3.2.3 Chorégraphie

Chorégraphie est une interface de programmation graphique qui utilise NAOqi, logiciel principal qui s'exécute sur le robot et le contrôle, lui permettant ainsi d'utiliser rapidement les fonctionnalités basiques de Nao simplement en glissant et en déposant une des 80 boîtes de comportements pour robots humanoïdes.

Caractéristiques de Chorégraphie [122] :

- Boîtes et comportements préprogrammés :
 - Mouvement : Animations, marcher, s'asseoir, se lever...
 - Audio : Localisation, synthèse et reconnaissance vocale...
 - Objets : détection et reconnaissance faciale, reconnaissance d'objets...
 - Communication : email, infrarouge, DELs
 - Commandes : for, if, multiplier, attente...
- Contrôle à distance du robot :
 - Gestionnaire de comportements : Charger et lancer des comportements à distance
 - Moniteur vidéo : Accéder à la caméra de votre robot en réseau local
 - Vue du robot : restitution 3D et contrôle du robot
- Accéder à tous les capteurs et moteurs :
 - Avec Chorégraphie, en programmant votre Nao Next Gen
 - Avec Monitor, fourni avec Chorégraphie
- Interface de programmation graphique :
 - Interface explicite : glisser et déposer des boîtes, suivi en temps réel de Nao Next Gen...
 - Outils intégrés : Vue du robot, fenêtre de debug, éditeur de script...
- Éditeur chronologique pour les animations :
 - Mode feuille de travail : Editeur de mouvements de la position des articulations représentés par des rectangles

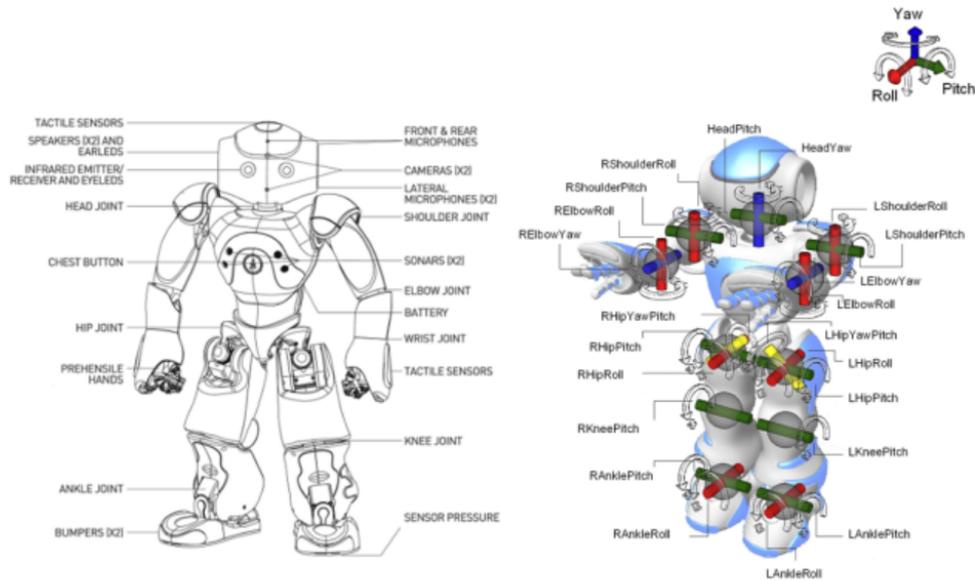


FIGURE 3.4 – Capteurs dont est équipé Nao 6 ainsi que ses degrés de liberté [2]

- Mode de courbe : Editeur de mouvements de la position des articulations représentés par des rectangles
- Outils de programmation :
 - Représentation graphique chronologique : programmation simultanée ou chronologique avec des couches de mouvements et de comportements
 - Diagramme de flux : programmation séquentielle, parallèle ou orientée, événement avec des boîtes reliées
 - Boîte de Script : Boîte où le codage Python est possible
 - Extraction automatique en C++ ou en Python à partir de mouvements enregistrés

Chorégraphe est un outil efficace et peu complexe pour effectuer des tâches simples avec Nao telles que le faire marcher, parler, ouvrir et fermer les doigts, ...etc. Cependant, on ne peut pas intégrer de programmes externes pour la détection d'objet comme Aruco que nous avons utilisé et qui sera présenté dans le chapitre 4. Chorégraphe est un outil destiné uniquement aux robots de SoftBank (e.g. Nao et Pepper).

3.2.4 Limitations et avantages

À partir des dimensions et du poids de Nao, on peut déduire qu'il n'est pas adapté pour porter ni pour déplacer des objets volumineux et/ou lourds. Il n'est pas grand de taille ce qui ne lui permet pas de faire une application où il faudra atteindre une position élevée. Par contre, il est plus adapté à faire des interactions de type social qui nécessitent la plupart du temps que Nao parle avec l'humain ou montre quelque chose (e.g. un mouvement à faire dans le cas de thérapies). Voilà pourquoi nous avons pensé utiliser une interaction de type physique où Nao déplace un objet pas lourd comme des pièces d'un jeu, ce qui nous permet d'utiliser une interaction physique pour une application à but social (jouer avec un enfant) puis nous laisse l'occasion par la suite de faire de véritables interactions physiques avec des robots de plus grande taille ayant des capacités de mouvement plus développées dans le cadre d'une tâche d'assemblage industriel.

3.3 ROS

3.3.1 Définition

ROS (Robot Operating System) est un méta système d'exploitation (ce qui signifie qu'il se situe entre le système d'exploitation et le middleware), sous licence libre (open source), pour robots. Il fournit des bibliothèques et des outils pour aider les développeurs de logiciels à créer des applications robotiques et doit s'interfacer avec le système d'exploitation LINUX. Il fournit une abstraction matérielle, des pilotes de périphériques, les bibliothèques, les outils de visualisation, le passage de messages, la gestion des paquets, et plus encore. Cet outil offre plusieurs fonctionnalités à différents niveaux, il assure notamment l'abstraction et la communication entre les différents modules robotiques à travers un mécanisme spécial très riche [123]. ROS est multi-langages : il permet la programmation d'applications en différents langages. La spécification de ROS intervient au niveau message.

3.3.2 Historique de ROS

ROS a été développé par l'université de Standford dans le cadre du projet "STAIR" en 2007. Tous les développements effectués entre 2008 et 2013 ont été issus des contributions apportées par Willow Garage. À partir de 2013, ROS dépend de l'Open Source Foundation Robotics [124]. ROS est actuellement plutôt utilisé dans le cadre de la recherche scientifique. Avant la création de ROS, chaque concepteur de robot et/ou chaque chercheur en robotique passait un certain temps à concevoir le robot aussi bien du point de vue matériel que logiciel. ROS a permis d'éviter de réinventer la roue pour chaque robot en proposant des fonctionnalités standardisées faisant donc abstraction du matériel [123].

3.3.3 Le système de fichiers de ROS

ROS est organisé dans une structure hiérarchique, et il contient plusieurs concepts importants [123] :

- Le package : C'est l'unité principale pour organiser les logiciels de ROS. Un package est un répertoire (dossier) qui contient les nœuds (cela sera explicité par la suite), les bibliothèques externes, des données, des fichiers de configuration et un fichier de configuration xml nommé manifest.xml.
- La stack : C'est une collection de packages. Elle propose une agrégation de fonctionnalités telles que la navigation, la localisation, ... etc. C'est surtout un répertoire qui contient les répertoires des packages ainsi qu'un fichier de configuration nommé stack.xml.

3.3.4 Les notions de base de ROS

Nous allons présenter les notions de bases de ROS qui permettent de savoir ce qu'il faut intégrer au programme codé en un ou plusieurs langages de programmation pour le passer sous ROS [123] :

- Un nœud : est une instance d'un exécutable. Le nœud peut représenter un capteur, un moteur, un algorithme de traitement, ... etc. Pour faire simple, les nœuds sont les bouts de programmes que nous créons et qui, une fois lancés ensemble, réalisent une tâche complexe.

- Un master : est un service de déclaration et d'enregistrement des nœuds qui permet ainsi aux nœuds de se connaître et d'échanger l'information essentiellement via des topics ou des services.
- Un topic : est un système de transport de l'information basé sur le système abonnement/publication (subscribe/publish) ; un ou plusieurs nœuds pourront publier l'information sur un topic (on appelle ces nœuds des "publishers"), tandis que d'autres pourront la lire (on appelle ces nœuds des "subscribers"). C'est donc un système de communication asynchrone Many-to-Many. Les topics ont la particularité d'être fortement type ; l'information publiée (le message) est toujours structurée de la même manière.
- Un message : est une structure de données composites, composée de données de types primitifs (chaînes de caractère, booléens, entiers, flottants, ... etc.).
- Un service : permet une communication synchrone entre les nœuds. Ainsi, l'échange de l'information s'effectue soit de manière asynchrone via un topic ou de manière synchrone via un service. Pour voir la relation entre les nœuds, les services et les topics, il faut se référer à la Figure 3.5.
- Un bag : est un fichier qui stocke les messages échangés. Ce système permet de rassembler les données mesurées par les capteurs, par exemple, et les rejouer ensuite autant de fois qu'on veut afin de faire de la simulation sur des données réelles ou pour déboguer un système a posteriori.

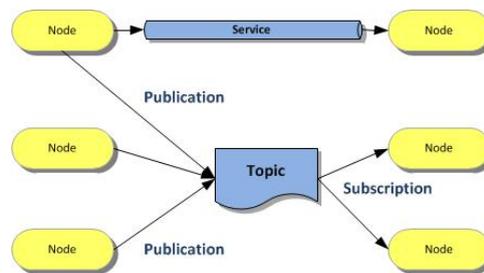


FIGURE 3.5 – Relation entre les nœuds, les services et les topics [123]

3.3.5 Fonctions de base de ROS

Il existe un grand nombre de fonctions fréquemment associées à ROS, telles que les bibliothèques de navigation, des outils et des algorithmes. Ces fonctions forment un ensemble puissant qui permet de travailler avec ROS facilement. Nous rappelons ci-dessous quelques outils ou algorithmes que le programmeur de ROS retrouvera souvent [123] :

- Gazebo : est un simulateur 3D capable de créer un plan 3D sur l'ordinateur avec des robots, des obstacles et de nombreux autres objets.
- ROS Graph (rqt graph) : une interface pour visualiser comment les choses (les nœuds, les topics, ... etc.) sont connectées.
- roslaunch : est un outil permettant de lancer facilement plusieurs nœuds ROS localement et à distance via SSH, ainsi que de définir des paramètres sur le serveur de paramètres.
- Rviz : est un système de visualisation 3D en temps réel (il n'inclut pas de moteur physique, comme Gazebo).
- Le paquet tf : permet de manipuler les coordonnées et les transformations de repères entre les différentes articulations (pour faire le modèle cinématique inverse).

- Point Cloud Library : permet la reconstruction d'environnement 3D à partir de mesures d'un laser.
- rxbag : permet de visualiser graphiquement les données enregistrées par un rosbag dans (fichiers .bag).
- Opencv : est un outil de traitement d'images.

3.3.6 URDF

ROS propose plusieurs autres notions que nous ne présenterons pas. Nous citerons toutefois une contribution intéressante de ROS à la robotique, qui est l'URDF (Unified Robot Description Format). C'est un fichier au format XML permettant de décrire sous une forme standardisée un robot complet (tout ce qui touche aux articulations : leurs types, les connections entre les différentes articulations, les limites articulaires, ... etc.). Le robot peut ainsi être décrit sous forme statique ou dynamique et des propriétés physiques et de collisions peuvent y être ajoutées. L'URDF est utilisé par exemple par le simulateur Gazebo pour représenter le robot.

3.3.7 Langages de programmation

Le cadre ROS est facile à mettre en œuvre dans n'importe quel langage de programmation moderne. Nous allons présenter les bibliothèques de ROS les plus utilisées et qui sont dédiées aux langages de programmation Python et C++ [125] :

- rospy : est une bibliothèque client, sous ROS, uniquement dédiée à Python. L'API client rospy permet aux programmeurs Python de s'interfacer rapidement avec les thèmes, services et paramètres ROS. La conception de rospy favorise la vitesse d'implémentation (c'est-à-dire le temps de développement) par rapport à la performance d'exécution afin que les algorithmes puissent être rapidement prototypés et testés dans ROS. Beaucoup d'outils ROS sont écrits en rospy pour profiter des capacités d'introspection de python, tels que rostopic et rosservice, sont construits sur rospy.
- roscpp : est une implémentation C++ sous ROS. Il fournit une bibliothèque client qui permet aux programmeurs C++ de s'interfacer rapidement avec les thèmes, services et paramètres ROS. roscpp est la bibliothèque client ROS la plus largement utilisée et est conçue pour être une bibliothèque de haute performance pour ROS.

3.3.8 Robots

L'écriture de logiciels pour les robots est difficile, d'autant que la richesse et la portée de la robotique ne cessent de croître. Les robots peuvent être très variables de point de vue matériel, ce qui rend la réutilisation de code non triviale, ROS favorise la réutilisation du code. Donc l'idée principale de l'outil ROS est de proposer des fonctionnalités standardisées qui sont variables et utilisables dans tous les types de robots. Les robots interfacés ROS sont [126] : ABB manipulators, ADAS development vehicle kit, Baxter, Motoman, Nao, Pepper, Qtrobot, Reem, ROSbot, Romeo, Shadow Hand, Turtlebot, ... etc.

3.3.9 Conclusion

Comme nous l'avons mentionné précédemment, ROS est fait de manière à s'adapter à tous les robots interfacés avec ROS. Il suffit pour passer d'un robot à un autre de changer de topic

sur le quel les commandes sont publiées. Ceci est la raison principale pour laquelle nous avons préféré utiliser ROS à la place de Chorégraphe qui ne marche qu'avec les robots de SoftBank (Aldebaran) sachant que l'objectif pendant la thèse serait de tester la plateforme expérimentale avec d'autres robots. Enfin, on ne peut pas intégrer des blocs externes sous chorégraphe comme par exemple un bloc pour faire de la détection d'objets.

3.4 Application

Comme nous l'avons mentionné précédemment, quant au choix de l'application, nous avons décidé d'allier une application à but social pour répondre aux attentes du LAPSCO à une application mettant en lumière le côté interaction physique. Quoi de mieux qu'un jeu pour être à la fois l'un et l'autre ? Nous avons donc décidé de choisir un jeu comme application pour faire une interaction homme-robot. Ce jeu devra remplir les deux critères suivants : ses différentes pièces pourront être manipulées et détectées par Nao et il pourra éventuellement servir de base pour tester une application industrielle par exemple une tâche d'assemblage. Nous avons alors pensé à une catégorie de jeux basés sur la tactique stimulant et développant les compétences cognitives. Pour cela il y a les jeux classiques comme les dominos, les puzzles, les jeux de construction type LEGO, ... etc. Puis nous avons trouvé les jeux de "SmartGames" assez intéressants puisqu'ils permettent de renforcer chez les enfants la logique, la concentration, les connaissances cognitives et sociales, les capacités motrices, la perception, la planification et la résolution de problèmes. Ce qu'il y a de plus attractif avec ces jeux c'est qu'il y a non seulement différents niveaux de difficultés mais également différents niveaux d'interaction : le robot peut jouer tout seul (c'est ce que nous allons faire dans le cadre du stage), l'humain peut jouer tout seul et le robot (Nao) apporte une aide orale de la stratégie à suivre ou les deux jouent ensemble. Dans le cas où les deux vont jouer en même temps, il y a également différents niveaux d'interaction puisqu'ils peuvent jouer l'un contre l'autre (confrontation), ils peuvent s'entraider pour trouver la solution plus rapidement que si la tâche était faite individuellement ou encore chacun joue à tour de rôle. Nous allons présenter quelques-uns des jeux de "SmartGames", ceux qui nous ont semblé les plus pertinents :

3.4.1 La nuit des fantômes

L'objectif avec ce jeu est de positionner les pièces avec des lampes torches dessinées dessus de telle sorte que chaque fantôme soit illuminé par une lampe torche en utilisant le moins de mouvement possible celui-ci étant limité. L'avantage de ce jeu est qu'on peut aisément faire un algorithme qui permet au robot sans connaître d'avance la solution de la trouver par lui-même.



FIGURE 3.6 – La nuit des fantômes : jeu développé par "Smart Games" [127]



FIGURE 3.7 – Code Couleur : jeu développé par "Smart Games" [128]

3.4.2 Code couleur

Ce jeu repose sur le sens d'observation du joueur. Il s'agit de reconstruire un motif en empilant les pièces avec des formes géométriques dessinées dessus les unes sur les autres (cf. Figure 3.7). Pour ce jeu, il n'est pas nécessaire que le robot connaisse d'avance la solution puisqu'il pourra savoir quelles pièces utiliser et dans quel ordre les empiler à partir de la couleur et la forme géométrique des pièces.

3.4.3 IQ

C'est un jeu destiné aux enfants comme aux adultes, les 2 variantes qui nous ont le plus intéressés de la catégorie de jeu IQ de "Smart Games" sont "IQ" Fit et "IQ Twist". Pour la première, les pièces sont déposées selon le schéma correspondant au niveau ensuite les pièces restantes doivent être remises dans la boîte de façon à ce que toutes les pièces soient dans la boîte (cf. Figure 3.8). Pour la deuxième, on positionne les embouts selon le schéma correspondant au niveau et on place les pièces de telle sorte que toutes soient placées dans la boîte tout en s'assurant que les tours des pièces seront comblés par des embouts de la même couleur que les pièces (cf. Figure 3.9).



FIGURE 3.8 – IQ Fit : jeu développé par "Smart Games" [129]



FIGURE 3.9 – IQ Twist : jeu développé par "Smart Games" [130]

3.4.4 Camelot Jr

Ce jeu a pour but de construire un chemin, à l'aide de pièces qui diffèrent d'un niveau à un autre, entre les deux tours pour que le prince et la princesse puissent se rejoindre (cf. Figure 3.10). Pour chaque niveau, il y a un niveau de difficulté marqué au bas de la page du livret qui explique comment est la configuration de départ et quelles pièces il faut placer pour construire le chemin (cf. Figure 3.11). Les pièces peuvent être mises dans n'importe quel sens du moment que le chemin tienne en place, que les pièces s'emboîtent correctement et qu'il n'y ait pas de chutes de pièces (cf. Figure 3.13). La solution est fournie en verso de la page du niveau dans le livret (cf. Figure 3.12), ce qu'on exploitera pour éviter d'avoir à trouver la solution. Nous avons décidé de choisir ce jeu parce que ses pièces sont facilement manipulables pour Nao, puisque le côté le plus petit d'un cube fait 3 cm raison pour laquelle les QR codes collés sur les cubes font 2.5 cm de côté, contrairement aux autres jeux présentés précédemment où il faudra souvent augmenter l'épaisseur des pièces pour que Nao puisse les saisir. On peut facilement l'étendre pour avoir une tâche d'assemblage industrielle.



FIGURE 3.10 – Camelot Jr : jeu développé par "Smart Games" [131]



FIGURE 3.11 – Niveau de "Camelot Jr" choisi



FIGURE 3.12 – Solution du niveau 10 du jeu "Camelot Jr"

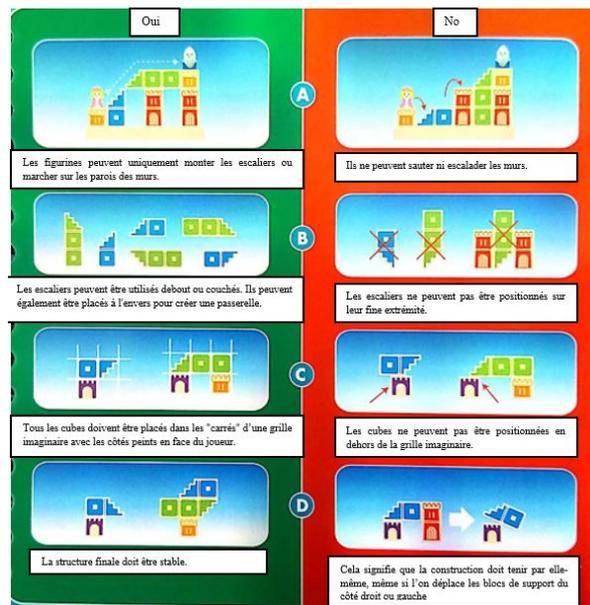


FIGURE 3.13 – Règles du jeu Camelot JR

3.5 Conclusion

Notre plateforme expérimentale sera donc composée de Nao le robot humanoïde, du jeu "Camelot Jr" de "SmartGames" et de ROS Kinetic (la version 2016 de ROS) interfacé avec Ubuntu 16.04 (la version de Linux qui est adaptée à la version choisie de ROS) en ce qui concerne le matériel informatique. Nous allons coder les programmes, pour contrôler Nao afin qu'il perçoive son environnement, en C++ et en Python. Dans le cadre du stage nous n'allons utiliser que le cas, pour les niveaux d'interaction, où le robot effectue la tâche tout seul c'est-à-dire qu'il arrive à placer les pièces pour construire le chemin tout seul sachant que Nao connaît d'avance la résolution du problème.

Chapitre 4

Techniques et méthodes

4.1 Introduction

Dans ce chapitre, nous allons présenter les différentes techniques utilisées, et les relier à l'état de l'art, pour la perception du robot de façon à ce qu'il soit capable de détecter et de localiser les cubes afin de pouvoir construire la solution d'un puzzle du jeu et pour contrôler le corps du robot afin qu'il puisse bouger la tête pour chercher ou suivre les pièces du jeu, et qu'il puisse aussi bouger le reste de son corps dans le but de saisir le cube et le déplacer d'un endroit à un autre. Puis nous allons finir par exposer les programmes que nous avons mis au point pour relier toutes ces méthodes pour réaliser le puzzle.

4.2 Perception

4.2.1 Calibration de la caméra

La calibration d'un système en général sert à déterminer la position et l'orientation de chaque capteur de ce système par rapport à un repère absolu ainsi que ses caractéristiques internes. Calibrer une caméra, c'est déterminer la transformation permettant de passer d'un objet défini dans un repère absolu à son image ce qui implique de savoir passer du repère absolu au repère caméra. Autrement dit la calibration d'une caméra, c'est connaître les paramètres intrinsèques de la caméra. Ces paramètres diffèrent d'une caméra à l'autre puisque la lentille et la rétine photo-sensible changent d'une caméra à l'autre. On peut calibrer une caméra à partir d'une ou plusieurs images d'un objet de calibrage dont la structure est connue (e.g. connaissance des coordonnées des points d'intérêt dans un repère quelconque). Le calibrage permet d'estimer la géométrie interne et celle externe de la caméra, quoique souvent, seule la géométrie interne est recherchée. La géométrie externe est la position et l'orientation de la caméra par rapport à l'objet de calibrage.

Nous avons fait la calibration en passant par le paquet ROS "camera_calibration"¹ qui fait la calibration grâce à un damier (cf. Figure 4.1). Le paquet permet de faire plusieurs photos du damier (pour réduire le bruit au maximum) dans différentes positions (il faut bouger le damier dans toutes les directions), ce qui permet d'avoir à la fin un fichier avec calibration de la caméra (la valeur de ses paramètres intrinsèques).

1. Paquet et damier disponibles sur le site : http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration

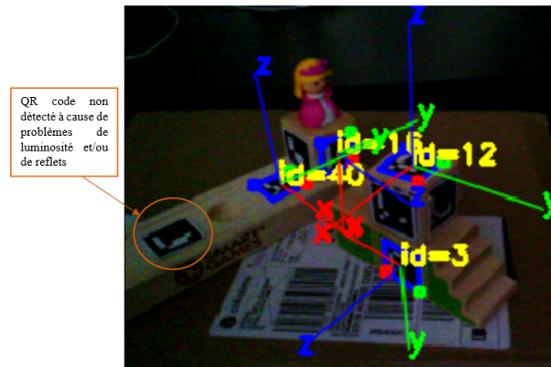


FIGURE 4.4 – Image prise durant les tests pour exposer le phénomène de non détection que nous avons rencontré

Il existe plusieurs papiers qui traitent de la détection de QR codes mais nous avons décidé d'utiliser la technique la plus connue, celle soumise par les auteurs de [132]. Ils ont proposé un système de marqueur standard spécialement adapté à la localisation à partir de caméras dans des applications telles que la réalité augmentée ou la robotique. Au lieu d'utiliser un ensemble prédéfini de marqueurs, une méthode générale pour générer des dictionnaires configurables en taille et en nombre de bits a été proposée. L'algorithme repose sur une recherche probabiliste maximisant deux critères : les distances entre les marqueurs et le nombre de transitions de bits. En outre, la distance maximale théorique entre marqueurs qu'un dictionnaire, avec des fabricants de carrés, peut avoir entre eux a été dérivée. L'article a également proposé une méthode automatique pour détecter les marqueurs et corriger les erreurs éventuelles.

4.3 Contrôleur

Nous avons fait en sorte de contrôler en même temps toutes les articulations de Nao, que ce soit pour bouger la caméra située au niveau de la bouche de Nao pour permettre au robot de chercher un cube ou le suivre quand il le déplace. Le contrôleur asservit toutes les articulations de Nao pour que la main atteigne la position 3D donnée par le QR code, pour saisir le cube et le déplacer puis le reposer et ce en déplaçant les articulations du bras pour le bouger et les jambes pour baisser le robot ou le faire se pencher. Lorsqu'on décide de bouger une seule articulation toutes les autres en sont affectées généralement sauf l'ouverture et la fermeture des doigts pour faire la saisie. Le contrôleur que nous avons utilisé classe les contraintes par ordre de priorité, ce qui est communément appelé une "pile de tâches" [133] et qui implique qu'il privilégie les tâches les plus importantes (en haut de la liste), c'est la hiérarchisation des tâches et gère ainsi la redondance des robots. Notre liste de tâches se décline par :

1. Respecter les limites articulaires de toutes les articulations de Nao pour ne pas le casser.
2. Faire en sorte que la projection au sol du centre de masse reste dans le polygone de sustentation pour garantir l'équilibre du robot.
3. Le changement de repères entre les 2 pieds du robot reste constant pour assurer l'équilibre.
4. Contrôle de la position de la main droite dans un repère monde (cf. Annexe B).
5. Contrôle de la position de la main gauche dans un repère monde (cf. Annexe B).

6. Bouger la tête de Nao pour changer l'orientation de la caméra (cf. Annexe B).
7. S'assurer que les doigts de la main droite sont à la même hauteur (sur l'axe Z) pour faire en sorte que la main garde une position stable (selon l'axe Z).
8. S'assurer que les doigts de la main gauche sont à la même hauteur (sur l'axe Z) pour faire en sorte que la main garde une position stable (selon l'axe Z).
9. Minimiser le facteur Q de façon à ce que la norme de la valeur articulaire tende vers zéro, ce qui assure que les articulations restent le plus proche les unes des autres. Cela permet également de remettre les articulations à la même position au cas où la pose finale donnée est redondante et ce en triant toutes les solutions possibles et en ne gardant toujours que la même à la fin.

L'asservissement en boucle fermée, lorsqu'il faut asservir selon une seule tâche dont le schéma est présenté par la Figure 4.5, est constitué du calcul de l'erreur (ε), entre la position désirée (X_d) et la position actuelle (X_a), dont la jacobienne (jacobienne = $\frac{\partial \text{erreur}}{\partial q}$; avec q la valeur articulaire) est J et à partir de sa pseudo-inverse (J^+), on extrait la valeur des vitesses articulaires du robot (\dot{q}) qu'on intègre pour avoir la valeur des positions articulaires (q) qu'on envoie au robot et qui lui sont nécessaires pour corriger sa position. Par contre, lorsqu'on a affaire à une pile de tâches avec une hiérarchisation, comme c'est le cas avec le contrôleur que nous avons utilisé, il faut modifier l'approche pour une tâche unique en empilant les tâches les unes après les autres selon l'ordre de priorité suivant l'équation [133] :

$$\dot{q}_i = \dot{q}_{i-1} + (J_i P_{i-1})^+ (\dot{e}_i^* - J_i \dot{q}_{i-1}) \quad (4.1)$$

avec $J_i = \frac{\partial \text{erreur}_i}{\partial q_i}$ avec i allant 1 jusqu'à N (le nombre de tâches), $P_i (= I_n - J_i^+ J_i$ avec n la dimension de J_i) est la projection de J_i dans l'espace nul (L'espace nul est un espace dans lequel l'erreur de la tâche est nulle c'est-à-dire que la tâche (contrainte) est satisfaite.) et

$$\dot{e}_i^* = \begin{cases} -\lambda e_i, & \text{si } e_i > 1 \\ 0, & \text{sinon} \end{cases} \quad (4.2)$$

avec λ le gain proportionnel

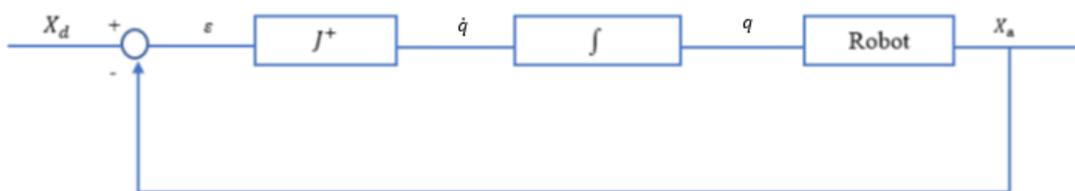


FIGURE 4.5 – Schéma blocs de l'asservissement en boucle fermée selon une tâche

4.4 Programmes

Nous allons présenter les programmes (machines d'états) que nous avons fait pour que Nao puisse résoudre le puzzle et qui, sous ROS, s'appellent des nœuds. Le premier "solve_puzzle"

est le nœud global ou principal qui gère les 2 autres afin de créer la séquence nécessaire à la résolution d'un puzzle, celle-ci étant supposée connue par le robot. Au début, on utilise le nœud "recherche_cube" qui permet de rechercher la position du QR code dans son espace de travail, qui est assez restreint à cause de la petite taille de Nao, grâce au contrôleur de la tête. Puis, quand on a localisé la position du marqueur collé sur le cube et celui posé sur la base et qui correspond à l'endroit où l'on veut que le cube soit déposé pour résoudre le puzzle, on lance "saisie_cube" le nœud qui permet de bouger le bras et les jambes pour atteindre la position du cube, le saisir, le relever et le poser sur la base puis le Bras de Nao retourne à sa position de départ. On recommence ce processus autant de fois qu'il y a de cubes à placer ; cette partie est agencée par "solve_puzzle". Pour connaître l'architecture détaillée des différents nœuds, il faut se référer à l'Annexe C

4.5 Conclusion

Nous avons présenté, dans ce chapitre, Aruco la méthode que nous avons utilisée pour la perception du robot et qui est basée sur la détection par vision de QR codes, ainsi que le contrôleur dont nous nous sommes servis pour bouger le robot afin qu'il puisse chercher, localiser, atteindre, saisir et poser le cube. Nous avons enfin présenté le rôle des nœuds que nous avons programmés pour mettre au point toutes ces étapes.

Chapitre 5

Résultats

5.1 Introduction

Dans ce chapitre, nous allons présenter les tests que nous avons effectués pour nous assurer de la validité de la plateforme expérimentale mise au point. Nous avons testé la construction de 3 puzzles du jeu "Camelot Jr" que nous allons exposer ainsi que l'analyse de chaque résultat et les limites que nous avons rencontrées.

5.2 Configuration de départ

La configuration initiale doit être correctement mise en place en tenant compte de la hauteur et de la longueur des bras du robot. Il faut surélever la base et les cubes du jeu, correspondant au puzzle (niveau) choisi, pour s'assurer que le robot peut les atteindre sans perdre son équilibre. Il faut aussi vérifier qu'ils ne sont pas placés trop loin par rapport à Nao (cf. Figure 5.1).

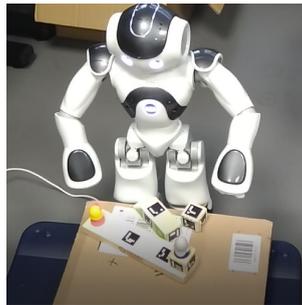


FIGURE 5.1 – Exemple de la configuration initiale du puzzle 1

5.3 Puzzle 1

Nous avons commencé par tester le premier niveau (le plus simple) : il s'agit de déposer les 2 cubes (le violet et le vert) l'un à côté de l'autre (cf. Figure 5.2 (a)). Il faut commencer par bien positionner les cubes et la base (configuration initiale : cf. Figure 5.2 (b)). Ensuite, les étapes suivantes sont effectuées pour déplacer et positionner les cubes :

- Puis, le robot commence à chercher le cube à l'aide du QR code collé dessus et le marqueur de l'emplacement où il faudra le mettre (cf. Figure 5.2 (c)).

- La main gauche de Nao se dirige vers le cube violet, le premier à saisir (cf. Figure 5.2 (d)), et s'arrête au-dessus de lui.
- Nao baisse la main et se saisit du cube (cf. Figure 5.2 (e)).
- Nao relève la main puis positionne le cube au-dessus de la position finale (cf. Figure 5.2 (f)).
- Nao pose le cube (cf. Figure 5.2 (g)).
- La main gauche de Nao revient à sa position initiale (cf. Figure 5.2 (h)).

Les étapes décrites précédemment se répètent pour déplacer le cube vert et les Figures 5.2 (c à h) sont remplacées par les Figures 5.2 (i à n).

5.4 Analyse des résultats du puzzle 1

5.4.1 Construction biaisée

Comme nous pouvons le remarquer en comparant la Figure 5.2 (n), qui représente le puzzle construit par le robot, et la Figure 5.2 (o), qui montre la solution du puzzle, les cubes posés par le robot sont légèrement positionnés de biais. Nous pensons que ce phénomène découle du fait que l'asservissement fait en sorte que le centre de la main atteint le centre du marqueur selon la position 3D, ce qui génère le problème de l'orientation du cube : les centres seront superposés mais le cube sera légèrement de biais puisque, en superposant 2 points, ils peuvent avoir n'importe quelle orientation ; les résultats seront démontrés dans la partie consacrée aux tests. Pour résoudre ce problème, il faut asservir en position et en orientation ; ce que nous n'avons pas pu faire par manque de temps.

5.4.2 occultations

Nous avons pensé qu'il serait important d'avoir des QR codes sur la main du robot pour pallier le problème d'occultations généré par Nao lorsqu'il saisit le cube et cache ainsi le marqueur posé sur lui. Mais nous n'avons pas pu détecter les marqueurs parce que le corps de Nao a des parties bombées surtout au niveau des bras et ils doivent être droits pour être détectés par Aruco (cf. chapitre 4). Vu que les QR codes ne pouvaient pas être détectés à cause de la forme spéciale du bras de Nao, nous avons décidé de ne pas mesurer la position réelle de la main mais de détecter et de sauvegarder les poses des QR codes au début du programme en ajoutant les mouvements des articulations générés par le contrôleur pour atteindre le cube.

5.4.3 Non détections

Nous avons rencontré des cas de non détections des QR codes (cf. Figure 4.4), apparents sur la caméra, par Aruco. Nous avons déjà cité cette limite classique reliée à la vision par ordinateur qu'utilise Aruco. Cela se produit généralement à cause des reflets dus à une forte luminosité. Pour pallier ce problème, nous avons réalisé les tests dans un environnement assez sombre. D'autres fois le problème est généré parce que les marqueurs ne sont pas plats. Pour cela nous n'avons mis des QR codes que sur les cubes et la base pour s'assurer qu'ils le soient. L'éloignement du marqueur par rapport à la caméra représente la dernière cause de la génération de ce phénomène. C'est pour cela qu'il ne faut pas que les QR codes, qui apparaissent dans l'image prise par la caméra, soient trop petits. Pour y remédier, nous avons rapproché le robot des cubes et de la base (la caméra que nous utilisons se situe au niveau de la bouche du robot).

5.4.4 Répétabilité

Nous avons essayé de refaire la construction du puzzle 1 plusieurs fois pour vérifier la répétabilité et la robustesse de la manipulation. Lorsque les cubes et la base sont placés exactement aux mêmes endroits que là où les configurations initiales ont permis la construction du puzzle, le robot est capable de reconstruire, ce qui nous permet de dire que la manipulation est répétable. Mais vu que le nombre de configurations initiales permettant de construire le puzzle ne dépasse pas les trois, la manipulation n'est par conséquent pas robuste. Le nombre de configurations de départ qui aboutissent à la construction du puzzle est limité à cause de la petite taille de Nao et le jeu de ses articulations, ce qui fait que le mouvement de la main de Nao n'est pas contrôlé au millimètre près, ce qui implique que les cubes et la base doivent être séparés par un espace égal à l'épaisseur des doigts (environ 3 cm) plus 1 cm pour le jeu. Pour réunir toutes ces conditions avec la zone atteignable par le robot et la zone visible par la caméra, qui est limitée par le champ de rotation du cou du robot (zone dans laquelle la caméra tourne), il ne reste pas beaucoup de choix quant à la position initiale des cubes et de la base.

5.5 Puzzle 2

Nous avons choisi le niveau 12 comme deuxième puzzle à construire. La solution de ce puzzle est d'empiler un cube sur l'autre afin de construire le passage entre les 2 figurines (cf. Figure 5.3 (o)), ce qui nous permet de vérifier que le robot est capable de poser un petit cube (le bleu) sur un plus grand cube (le vert) présents dans la Figure 5.3 (a). Comme pour le précédent, nous débutons le test en positionnant la base et les cubes selon la configuration initiale exposée par la Figure 5.3 (b). Nous avons suivi la même procédure pour la construction de ce puzzle que celle explicitée pour le puzzle 1 et que l'on peut suivre via les Figures 5.3 (c) à (n).

5.6 Analyse des résultats du puzzle 2

5.6.1 Défi relevé

Comme nous pouvons le remarquer en comparant la Figure 5.3 (n), qui représente le puzzle construit par le robot, et la Figure 5.3 (o), qui montre la solution du puzzle, les cubes posés par le robot sont légèrement positionnés de biais comme pour le puzzle précédent. Nous avons explicité les raisons du biais et les possibilités de s'en défaire. Nous allons plutôt mettre l'accent sur le fait que le défi de déposer un cube sur l'autre a été relevé haut la main puisque le résultat obtenu est semblable à celui du puzzle 1, ce qui confirme que déposer un cube plus petit sur un cube plus grand ne rajoute de contrainte ni pour le contrôleur ni pour le robot.

5.6.2 Limites articulaires

Nous avons rencontré le même phénomène que pour le puzzle 1, sauf que pour ce puzzle une seule configuration initiale aboutit à une construction réussie à cause de la taille du cube bleu qui est plus volumineux que le cube mauve utilisé dans le puzzle 1, ce qui rend les possibilités de placer les cubes et la base pour une position initiale encore plus limitées. Pour ce puzzle, nous avons dû poser le cube bleu hors du champ de vision du robot pour respecter la limite de l'espace atteignable par le bras gauche de Nao et de l'y replacer après que le robot a pris et a correctement déposé le cube vert sur la base pour libérer de la place.

5.6.3 Jeu articulaire

Il existe un jeu dans les articulations pour assurer une certaine tolérance concernant les positions atteignables par chaque articulation pour éviter qu'elle se casse, ce qui fait que le contrôle n'est pas assez précis. Nous avons surtout remarqué ce phénomène quand le robot saisit le cube et au lieu de le saisir avec les 3 doigts il n'en utilise que 2 (cf. Figure 5.3 (k et l)) car sa main s'est légèrement décalée vers la droite dans certains tests à cause du jeu des articulations présent dans le corps de Nao. Cela ne pose pas vraiment de problèmes quant à la construction du puzzle puisque Nao serre les doigts assez fort pour être capable de saisir un cube avec uniquement 2 doigts.

5.7 Puzzle 3

Nous avons voulu construire le niveau 18 et nous avons considéré un puzzle 3 pour tester si, au lieu de déposer le petit cube sur le grand comme dans le puzzle 2, le robot arrivait à faire le contraire, à savoir résoudre le puzzle 3 (cf. Figure 5.4 (n)) en déposant le grand cube (le vert) sur le petit cube (le mauve), puzzle présenté par la Figure 5.4 (a). Pour cela, nous avons commencé avec la configuration initiale présente dans la Figure 5.4 (b) puis nous avons effectué les mêmes étapes pour construire le puzzle 1 et 2 lesquelles sont explicitées par les Figures 5.4 (c) à (n).

5.8 Analyse des résultats du puzzle 3

Les mêmes remarques faites à partir des tests du puzzle 1 et 2 sont présentes pour ce puzzle 3. La nouveauté c'est qu'à partir de la Figure 5.4 (n) le grand cube (vert) est incliné sur le petit cube (mauve) c.à.d que le robot n'arrive pas à le placer horizontalement sur l'autre. Ceci est dû au jeu qui existe dans les articulations de Nao, cité précédemment, et qui fait que le robot ne pose pas le cube exactement à la position exigée ce qui fait qu'il tombe sur le côté étant donné que le cube du dessus est plus grand et plus lourd. La solution dont les étapes sont démontrées par la Figure 5.4 est la meilleure que nous avons pu obtenir. Nous pouvons dire que déposer un grand cube sur un très petit ne peut actuellement être effectué par le robot de manière satisfaisante.

5.9 Modification de la plateforme expérimentale

Nous allons apporter dans le cadre de la thèse quelques améliorations à la plateforme expérimentale mise au point dans le cadre de ce stage pour la rendre moins contraignante et l'adapter à l'interaction homme-robot. Tout d'abord, il faudra probablement changer le mode de perception des cubes en utilisant de la vision par ordinateur basée sur la couleur de chaque cube et peut être aussi en utilisant un filtre pour détecter les lignes droites permettant ainsi de détecter la forme de l'objet au lieu des QR codes. Il faudra aussi songer à un moyen pour gérer les problèmes d'occultations et de luminosité (intensité et reflets). Cette partie nécessitera 4 à 6 semaines pour être mise au point.

Ensuite, il faudra impérativement intégrer d'autres tâches à la pile de tâches du contrôleur et en modifier d'autres. Le temps nécessaire à ces modifications est réparti selon la Table 5.1. Il faudra entre autres :

Tâche	Durée
1	2 semaines
2	3 semaines
3	2 semaines
4	2 semaines

TABLE 5.1 – Durée de travail pour modifier le contrôleur

1. Intégrer l'orientation de l'objet à la position désirée, qu'on envoie au contrôleur, la transformant ainsi en pose désirée pour s'assurer que le cube est déposé à la bonne orientation.
2. Faire en sorte que Nao utilise ses deux mains en même temps pour qu'il saisisse le cube avec la main la plus proche de lui et ainsi étendre sa zone de travail.
3. Se servir de la capacité de changer de main pour changer l'orientation des cubes.
4. Se déplacer.

5.10 Conclusion

De tous les tests que nous avons effectués, nous pouvons affirmer que le robot est capable de construire des puzzles en posant soit des cubes les uns à côté des autres soit en posant un petit sur un autre de plus grande taille ; dans les deux cas il y aura un léger biais dû à une erreur de contrôle de Nao laquelle résultant de la prise en considération limitée à la position 3D. Cette erreur peut être évitée en intégrant l'orientation de la position à atteindre, ce qui n'est pas le cas quand il s'agit de placer un grand cube sur un petit. Ceci est majoritairement dû au jeu qui existe dans les articulations. La répétabilité de la manipulation est garantie du moment que la configuration initiale ne change pas ou très peu. Ce qui rend la manipulation peu robuste. Ceci est dû aux limites articulaires assez contraignantes de Nao qui font que l'espace atteignable par le robot est assez limité. Nous avons de même mis en lumière les principales modifications qu'il faudra apporter à la plateforme expérimentale pour la rendre moins contraignante.



FIGURE 5.2 – Description de toutes les étapes pour la résolution du puzzle 1 : (a) puzzle à construire, (b) configuration initiale, (c) le robot tourne la tête pour visualiser le marqueur du 1er cube et celui de son emplacement final, (d) la main de Nao se dirige vers le cube, (e) Nao saisit le 1er cube, (f) Nao relève le bras et se positionne au-dessus de la position cible, (g) Nao pose le cube, (h) la main gauche de Nao retourne à sa position initiale, (i) le robot tourne la tête pour visualiser le marqueur du 2ème cube et celui de son emplacement final, (j) la main de Nao se dirige vers le cube, (k) Nao saisit le 2ème cube, (l) Nao relève le bras et se positionne au-dessus de la position cible, (m) Nao pose le cube, (n) la main gauche de Nao retourne à sa position initiale et le puzzle est fini et (o) la solution du puzzle 1.

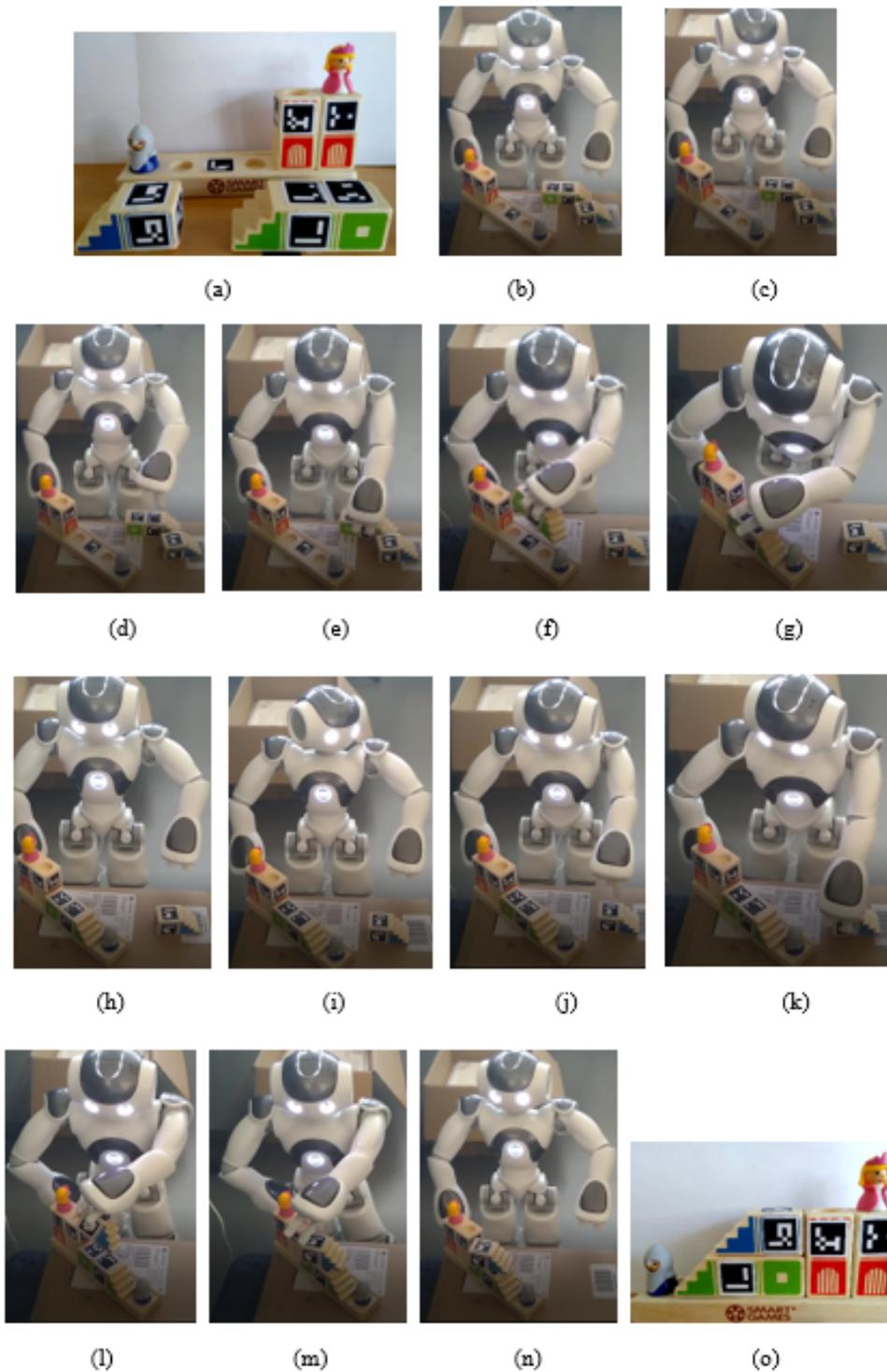


FIGURE 5.3 – Description de toutes les étapes pour la résolution du puzzle 2 : (a) puzzle à construire, (b) configuration initiale, (c) le robot tourne la tête pour visualiser le marqueur du 1er cube et celui de son emplacement final, (d) la main de Nao se dirige vers le cube, (e) Nao saisit le 1er cube, (f) Nao relève le bras et se positionne au-dessus de la position cible, (g) Nao pose le cube, (h) la main gauche de Nao retourne à sa position initiale, (i) le robot tourne la tête pour visualiser le marqueur du 2ème cube et celui de son emplacement final, (j) la main de Nao se dirige vers le cube, (k) Nao saisit le 2ème cube, (l) Nao relève le bras et se positionne au-dessus de la position cible, (m) Nao pose le cube, (n) la main gauche de Nao retourne à sa position initiale et le puzzle est fini et (o) la solution du puzzle 2.

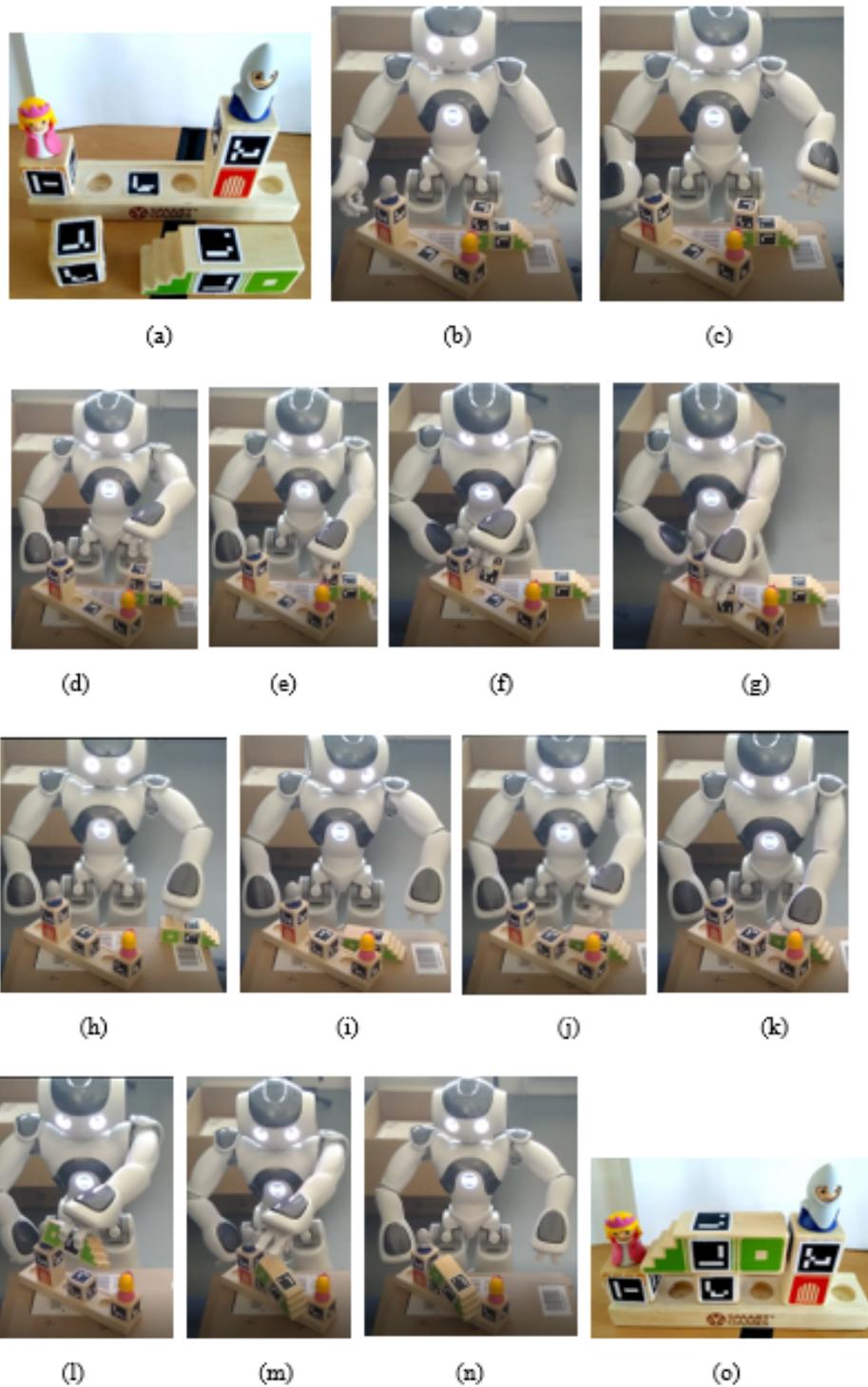


FIGURE 5.4 – Description de toutes les étapes pour la résolution du puzzle 3 : (a) puzzle à construire, (b) configuration initiale, (c) le robot tourne la tête pour visualiser le marqueur du 1er cube et celui de son emplacement final, (d) la main de Nao se dirige vers le cube, (e) Nao saisit le 1er cube, (f) Nao relève le bras et se positionne au-dessus de la position cible, (g) Nao pose le cube, (h) la main gauche de Nao retourne à sa position initiale, (i) le robot tourne la tête pour visualiser le marqueur du 2ème cube et celui de son emplacement final, (j) la main de Nao se dirige vers le cube, (k) Nao saisit le 2ème cube, (l) Nao relève le bras et se positionne au-dessus de la position cible, (m) Nao pose le cube, (n) la main gauche de Nao retourne à sa position initiale et le puzzle est fini et (o) la solution du puzzle 3.

Chapitre 6

Perspectives

6.1 Introduction

Nous allons présenter dans ce chapitre la théorie des jeux qui servira de cadre méthodologique que nous allons mettre en place durant la thèse de même que l'interaction homme-robot. Nous allons également mettre l'accent sur certaines modifications que nous allons apporter à la plateforme expérimentale exposée dans les chapitres précédents.

6.2 Théorie des jeux

Les mathématiciens, à partir du XIX^{ème} siècle, ont mis en place des modèles pour simplifier des situations à priori compliquées, telles que les conflits politiques, diplomatiques ou les relations humaines complexes, en les assimilant à des sortes de jeux plus intuitifs. D'où le nom "théorie des jeux". Ce domaine des mathématiques s'intéresse aux choix des individus appelés "joueurs" et permet de simplifier la variable humaine, à savoir les réactions émotionnelles des protagonistes, qui sont dures à prévoir, en réfléchissant de manière plus mathématique. Grâce à la théorie des jeux, on arrive à étudier non seulement des phénomènes politiques, mais également économiques et psychologiques. La théorie des jeux classe les jeux (ce n'est pas forcément des jeux puisqu'ils peuvent être des situations réelles représentées sous forme d'un jeu) d'après leurs approches de résolution. Nous allons exposer les catégories les plus courantes par la suite [134].

6.2.1 Jeux coopératifs

Pour les jeux coopératifs, il faut étudier la formation de coalitions entre les joueurs afin d'obtenir les meilleurs résultats. Un jeu coopératif est un jeu où les joueurs ont la possibilité de se concerter et de s'engager à coopérer avant de définir la stratégie à adopter. Le modèle donné par la théorie des jeux pour ce type de jeu se présente sous la forme de deux joueurs, deux stratégies et une matrice de gains (cf. Table 6.1). Des joueurs rationnels vont coopérer sur l'une ou l'autre des stratégies et recevoir les gains élevés. Pour ce faire, ils doivent pouvoir se coordonner sur l'une ou l'autre des stratégies, sous peine de se retrouver dans une situation défavorable.

	Joueur 1, Stratégie 1	Joueur 1, Stratégie 2
Joueur 2, Stratégie 1	A_1, A_2	B_1, B_2
Joueur 2, Stratégie 2	C_1, C_2	D_1, D_2

TABLE 6.1 – Matrice des gains pour deux joueurs et deux stratégies où $A_1 > C_1$, $D_1 > B_1$, $A_2 > B_2$, $D_2 > C_2$

6.2.2 Jeux séquentiels et jeux simultanés

Dans un jeu séquentiel, on peut spécifier l'ordre des décisions de telle sorte qu'un joueur peut décider de sa stratégie en fonction de ce que les autres joueurs ont joué précédemment comme les échecs ou le jeu de go. Par contre, dans un jeu simultané, les joueurs décident de leur stratégie au même moment comme par exemple le dilemme du prisonnier et le jeu pierre-feuille-ciseaux. À titre d'exemple, nous allons modéliser la situation du "dilemme du prisonnier" grâce à la théorie des jeux. La version la plus courante du jeu se présente comme suit : Deux personnes (vous et un complice) ont monté une escroquerie mais, la police se doute de quelque chose et arrête les 2 protagonistes, puis après les avoir placés dans deux cellules différentes, les policiers les obligent à faire un choix entre avouer ou non, ce qui les met face au dilemme suivant (cf. Figure 6.1) :

- les deux personnes avouent, elles écotent toutes les deux une peine de cinq ans de prison.
- l'un des deux avoue et est libéré alors que l'autre dément et passera dix ans derrière les barreaux.
- aucun des deux n'avoue, et par manque de preuve, la police les relâche au bout de six mois.

La meilleure solution pour les deux serait de ne pas avouer, mais malheureusement, ne pouvant se parler, impossible de savoir si l'autre va coopérer. Le problème réside dans le fait de savoir quel choix faire ? Pour répondre à cette question, étudions les différents cas :

- l'un avoue, l'autre a intérêt à avouer et dans ce cas les deux passeront 5 ans en prison, dans le cas contraire, celui qui n'a pas avoué passera dix ans en prison alors que celui qui a avoué sera libéré.
- l'un dément, l'autre a intérêt à avouer pour sortir tout de suite.

Dans tous les cas, chacun a intérêt à avouer. Si les deux prisonniers suivent ce raisonnement, ils passeront tous deux cinq ans en prison et n'auront pas ainsi réussi à atteindre la solution optimale (ils sont libérés tous les deux au bout de six mois). Tout le paradoxe est là : la solution trouvée par un raisonnement cartésien mène à une situation qui n'est pas la meilleure pour aucun des deux protagonistes. Dans cet exemple, utiliser la théorie des jeux est utile pour passer le moins de temps possible sous les barreaux.

6.2.3 Jeux à somme nulle et jeux à somme non nulle

Les jeux à somme nulle (jeux strictement compétitifs) sont des jeux à deux joueurs où l'intérêt d'un joueur est strictement opposé à celui de l'autre. Lorsqu'on représente l'intérêt de chaque joueur par une fonction de gain, la somme des deux fonctions est toujours nulle. Les échecs, le tarot, le jeu pierre-feuille-ciseaux ou le poker sont des jeux à somme nulle puisque les gains

Complice	Avouer	Démentir
Vous	5 ans	10 ans
Avouer	5 ans	Sortie
Démentir	Sortie	6 mois
	10 ans	6 mois

FIGURE 6.1 – Matrice de gains représentant le raisonnement issu de la théorie des jeux dans le cas du "dilemme du prisonnier" [135]

d'un joueur sont égaux aux pertes de l'autre et ce contrairement au dilemme du prisonnier qui ne l'est pas puisque dans certains cas, les deux joueurs peuvent perdre.

6.2.4 Jeux répétés

La répétition d'un jeu change souvent son déroulement. Par exemple, il peut être utile de prendre parfois le risque de perdre pour tester les autres joueurs et mettre en place des stratégies en conséquence. Il se développe des phénomènes de réputation qui vont avoir un impact sur la stratégie suivie par les autres joueurs. Dans le dilemme du prisonnier, le fait de savoir qu'on va jouer plusieurs fois avec un joueur qui n'avoue jamais, ou avec un autre joueur qui avoue toujours, change diamétralement la solution optimale.

6.2.5 Jeux à information complète ou non

Un jeu est à information complète si chaque joueur connaît lors de la prise de décision ses possibilités d'action, celles des autres joueurs, les gains résultant de ces actions et les motivations des autres joueurs. Les jeux en information incomplète (jeux bayésiens) sont des situations où l'une des conditions précédentes n'est pas vérifiée. Ce qui peut être dû au fait qu'une des motivations d'un acteur est cachée (domaine important pour l'application de la théorie des jeux à l'économie).

6.2.6 Jeux à mémoire parfaite et imparfaite

Les jeux à mémoire parfaite représente des situations où chaque joueur peut se souvenir des coups précédemment joués, si besoin, en prenant note de tous les coups joués. Les jeux à mémoire imparfaite supposent qu'une partie ou la totalité des coups qui ont été joués sont oubliés par les joueurs comme dans le cas des jeux de guerre durant lesquels si les commandements de zones opérationnelles ne parviennent pas à communiquer entre eux ou avec l'État-Major, ils n'auront pas de trace des mouvements déjà effectués par les troupes alliées.

6.2.7 Jeux en cas de joueurs "irrationnels"

La théorie des jeux prédit le comportement de joueurs "rationnels". Comme les gens ne sont pas des machines, ils n'adoptent pas toujours la stratégie la plus raisonnable. Lorsqu'il y a une différence entre ce qui est prévu par la théorie et ce qui se passe réellement, un phénomène psychologique influence cette décision :

L'aversion pour l'injustice

On va illustrer ce phénomène à travers le jeu de l'ultimatum qui se joue à deux : Le joueur 1 dispose de 100 euros, et doit faire deux tas avec : un tas de X euros pour lui, un tas de $100 - X$ euros pour le joueur 2. Donc, plus le joueur 1 prend de l'argent, moins le joueur 2 en aura. Ensuite le joueur 2 décide d'accepter ou non le partage :

- s'il accepte, le joueur 1 gagne ses X euros et le joueur 2 gagne $100 - X$ euros.
- s'il refuse, aucun des deux joueurs ne gagne d'argent.

Ce jeu représente les situations économiques où il y a un droit de véto, ou encore un boycott qui pénalise tout le monde. En théorie, le joueur 2 a tout le temps intérêt à accepter le partage, car il gagnera toujours plus d'argent en acceptant qu'en refusant. Pourtant, lorsque le partage est trop inéquitable, il aura tendance à refuser. La nature humaine n'aime pas l'injustice et les gens refusent parfois les situations injustes même s'ils en tirent profit en cas d'acceptation. Il s'agit ici d'aversion pour l'injustice.

L'aversion de la perte

Nous allons nous servir d'un autre jeu : le joueur 1 reçoit 150 euros, et on lui laisse alors deux choix : il les empoche directement ou il les mise dans un jeu de pile ou face, sachant qu'il va remporter 400 euros s'il gagne. La théorie des jeux prédit qu'un joueur devra toujours miser son argent, puisqu'en misant son argent, on gagne "en moyenne" 200 euros, alors qu'en ne le misant pas, on ne gagne que 150 euros. Cependant, beaucoup de joueurs décident de se contenter des 150 euros, par peur de les perdre ; on parle dans ce cas d'aversion pour la perte. Il faut se méfier de cette peur qui dissuade d'investir dans des placements pourtant prometteurs.

6.3 Cadre Méthodologique

Dans la thèse, nous allons mettre au point une interaction homme-robot. Pour cela, il faut que le robot soit capable de jouer au jeu et de réagir en fonction de l'humain et de la progression du jeu. Pour le premier point, nous allons nous servir de la plateforme expérimentale que nous avons mise au point et qui est présentée précédemment et ce en y apportant quelques modifications que nous avons exposés dans le chapitre précédent. Pour le second, nous allons nous servir de la théorie des jeux qui non seulement permet d'avoir la meilleure solution à un jeu ou à un dilemme réel qu'on aura représenté sous forme d'un jeu mais elle permet aussi de faire entrer en jeu le facteur relié à l'incertitude quant à la réaction de l'humain puisque celle-ci n'est pas toujours prévisible étant donné que le facteur psychologique peut rendre le raisonnement non "raisonnable" (prévisible) pour le robot.

Il n'est pas exclu que pendant la thèse nous nous servions d'autres robots (raison pour laquelle nous avons utilisé ROS), qu'ils soient humanoïdes ou non ; dans ce cas, il s'agira de tâches d'assemblages industriels et non plus de jeu, ainsi que d'autres jeux pour prouver que le cadre méthodologique est indépendant du robot et de la tâche d'assemblage (qu'il s'agisse d'un jeu ou non).

6.4 Conclusion

Nous avons dans ce chapitre montré toutes les possibilités de modèles donnés par la théorie des jeux pour obtenir le bon raisonnement à faire, chacun s'adaptant à une situation différente, prouvant ainsi un bon potentiel pour doter un robot de bonnes capacités de raisonnement humain.

Conclusion générale

Nous avons, au travers de ce rapport de stage, présenté le cadre dans lequel ce travail a été effectué (chapitre 1) ainsi que les raisons qui nous ont menés à choisir la plateforme expérimentale (chapitre 3) comme base pour mettre au point, durant la thèse, l'interaction homme-robot en se servant de la théorie des jeux comme d'un cadre méthodologique (cf. chapitre 6). Nous avons aussi résumé dans le chapitre 2 toutes les disciplines nécessaires pour mettre au point une interaction homme-robot réussie.

La plateforme expérimentale se compose de Nao, un robot humanoïde, du système d'exploitation Ubuntu 16.04, de la version Kinetic de ROS et du jeu "Camelot Jr". Nous avons décidé de choisir un petit robot humanoïde et d'utiliser un jeu comme application pour que l'humain, dans l'interaction humain-robot, puisse être un enfant et ce afin de pouvoir connaître les impacts cognitifs qu'aura le robot sur l'enfant. Ceci est l'objectif principal du projet ATTRIHUM dans le cadre duquel ce stage a été réalisé. Nous avons choisi de programmer la plateforme en se servant de ROS en raison de son adaptabilité au cas où l'on change le robot utilisé.

Pour que le robot soit capable de construire le chemin, entre les deux figurines, qui représente la solution du jeu, connue d'avance par Nao, il faut s'occuper de deux points à savoir la perception et le contrôle du robot (cf. chapitre 4). Pour la perception, nous avons opté pour une localisation ainsi que pour une détection des cubes et de la base à l'aide de QR codes collés dessus. Quant au contrôle, nous avons utilisé un contrôle basé sur une pile de tâches lequel commande toutes les articulations du robot afin d'atteindre une position désirée.

Les tests effectués, dans le chapitre 5, ont permis de constater que la répétabilité de la manipulation est assurée, du moment que les conditions initiales de chaque niveau du jeu, dont notamment la position initiale des cubes et de la base, restent les mêmes. Ce qui fait que la plateforme expérimentale n'est pas robuste quant aux changements de la position initiale des cubes. Les tests ont mis en lumière les limites du contrôleur et des modifications nécessaires (cf. chapitre 6) pour rendre la plateforme expérimentale moins contraignante comme par exemple le fait de devoir asservir Nao non seulement en position mais aussi en orientation.

Références

- [1] “Fiche technique Nao : www.generationrobots.com/fr/403100-robot-humanoide-programmable-nao-v6.html.”
- [2] “Capteurs du robot Nao : stephane.genouel.free.fr/FT/Dossier_Multimedia/RobotNao/co/Robot_Nao_11.html.”
- [3] M. A. Goodrich and A. C. Schultz, “Human-robot interaction : A survey,” *Foundations and Trends® in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.
- [4] D. Keltner and J. Haidt, “Social functions of emotions at four levels of analysis,” *Cognition & Emotion*, vol. 13, pp. 505–521, Sept. 1999.
- [5] K. R. Scherer, “Psychological models of emotion,” *The neuropsychology of emotion*, vol. 137, no. 3, pp. 137–162, 2000.
- [6] R. W. Picard, *Affective computing*. MIT press, 2000.
- [7] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, “Experiences with an interactive museum tour-guide robot,” *Elsevier : Artificial Intelligence*, pp. 1–53, May 1999.
- [8] K. Wada and T. Shibata, “Living with seal robots-its sociopsychological and physiological influences on the elderly at a care house,” *IEEE Transactions on Robotics*, vol. 23, pp. 972–980, Oct. 2007.
- [9] P. H. J. Kahn, B. Friedman, D. R. Pérez-Granados, and N. G. Freier, “Robotic pets in the lives of preschool children,” *Interaction Studies*, vol. 7, no. 3, pp. 405–436, 2006.
- [10] F. Tanaka, A. Cicourel, and J. R. Movellan, “Socialization between toddlers and robots at an early childhood education center.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, pp. 17954–17958, 2007.
- [11] T. Kanda and H. Ishiguro, *Human-Robot Interaction in Social Robotics*. CRC Press, 1 ed., Mar. 2017.
- [12] C. Breazeal, P. L. Harris, D. DeSteno, J. M. K. Westlund, L. Dickens, and S. Jeong, “Young children treat robots as informants,” *Topics in Cognitive Science*, vol. 8, pp. 481–491, Apr. 2016.
- [13] J. Kennedy, P. Baxter, E. Senft, and T. Belpaeme, “Social robot tutoring for child second language learning,” in *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 231–238, Mar. 2016.
- [14] S. Haddadin and E. Croft, “Physical human-robot interaction,” in *Springer Handbook of Robotics*, pp. 1835–1874, Cham : Springer International Publishing, 2016.
- [15] M. Swangnetr and D. B. Kaber, “Emotional state classification in patient-robot interaction using wavelet analysis and statistics-based feature selection,” *IEEE Transactions on Human-Machine Systems*, vol. 43, pp. 63–75, Jan. 2013.

- [16] T. B. Sheridan, “Human-robot interaction : Status and challenges,” in *Human Factors and Ergonomics Society*, vol. 58, pp. 525–532, Apr. 2016.
- [17] S. Rosenthal, M. Veloso, and A. K. Dey, “Is someone in this office available to help me?,” *Journal of Intelligent & Robotic Systems*, vol. 66, pp. 205–221, July 2011.
- [18] H. L. Cao, G. V. de Perre, J. Kennedy, E. Senft, P. G. Esteban, A. D. Bei, R. Simut, T. Belpaeme, D. Lefeber, and B. Vanderborght, “A personalized and platform-independent behavior control system for social robots in therapy : Development and applications,” *IEEE Transactions on Cognitive and Developmental Systems*, Jan. 2017.
- [19] K. C. Welch, U. Lahiri, C. Liu, R. Weller, N. Sarkar, and Z. Warren, “An affect-sensitive social interaction paradigm utilizing virtual reality environments for autism intervention,” in *International Conference on Human-Computer Interaction*, pp. 703–712, Springer, 2009.
- [20] A. Gardecki and A. Kawala-Janik, “The application extending the capabilities of the human-robot interaction of humanoid robot Nao,” *Proceedings of the Electrotechnical Institute*, pp. 45–53, 2016.
- [21] “Tâches d’assemblages industriels : www.tirard-burgaud.com/categories/accessoires-communs-pour-portail/cadres/systemes-d-assemblage-mecanique-de-profiles/systeme-d-assemblage-mecanique-de-profiles.”
- [22] M. Rahimi and W. Karwowski, “A research paradigm in human-robot interaction,” *International Journal of Industrial Ergonomics*, vol. 5, pp. 59–71, Jan. 1990.
- [23] T. Moulières-Seban, D. Bitonneau, J.-M. Salotti, J.-F. Thibault, and B. Claverie, “Human factors issues for the design of a cobotic system,” *Advances in Intelligent Systems and Computing*, vol. 499, pp. 375–385, 2017.
- [24] D. McColl, A. Hong, N. Hatakeyama, G. Nejat, and B. Benhabib, “A survey of autonomous human affect detection methods for social robots engaged in natural HRI,” *Journal of Intelligent & Robotic Systems*, vol. 82, pp. 101–133, Aug. 2015.
- [25] M. Scheutz, P. Schermerhorn, and J. Kramer, “The utility of affect expression in natural language interactions in joint human-robot tasks,” in *Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pp. 226–233, Mar. 2006.
- [26] P. Hinds, T. Roberts, and H. Jones, “Whose job is it anyway? a study of human-robot interaction in a collaborative task,” *Human-Computer Interaction*, vol. 19, pp. 151–181, June 2004.
- [27] F. J. Rodriguez-Lera, L. Gomes, P. Ziafati, A. Nazarikhorrām, A. Stefanetti, and A.-M. Schuller, “Emotional robots for coaching : Motivating physical rehabilitation using emotional robots,” *Personal Robots for Exercising and Coaching (PREC)*, Mar. 2018.
- [28] M. Devanne, S. M. Nguyen, O. Remy-Neris, B. L. Gals-Garnett, G. Kermarrec, and A. Thepaut, “A co-design approach for a rehabilitation robot coach for physical rehabilitation based on the error classification of motion errors,” in *IEEE International Conference on Robotic Computing (IRC)*, pp. 352–357, Feb. 2018.
- [29] O. Pino, G. Palestra, R. Trevino, and B. D. Carolis, “The humanoid robot Nao as trainer in a memory program for elderly people with mild cognitive impairment,” *International Journal of Social Robotics*, Feb. 2019.
- [30] C. B. Nicolas Spatola, A. Normand, P. Chausse, S. Monceau, M. Augustinova, V. Barra, P. Huguet, and L. Ferrand, “Not as bad as it seems : When the presence of a threatening humanoid robot improves human performance,” *Science Robotics : Human-Robot Interaction*, vol. 3, Aug. 2018.

- [31] C. M. MacLeod, "The stroop task : The "gold standard" of attentional measures," *Journal of Experimental Psychology General*, vol. 121, no. 1, pp. 12–14, 1992.
- [32] P. Bourgeois and U. Hess, "The impact of social context on mimicry," *Biological Psychology*, vol. 77, pp. 343–352, Mar. 2008.
- [33] A. Taheri, M. Alemi, A. Meghdari, H. PourEtemad, and S. Holderread, "Clinical application of humanoid robots in playing imitation games for autistic children in Iran," *Elsevier : Procedia-Social and Behavioral Sciences*, vol. 176, pp. 898–906, Feb. 2015.
- [34] D. Hood, S. Lemaignan, and P. Dillenbourg, "When children teach a robot to write : An autonomous teachable humanoid which uses simulated handwriting," *Association for Computing Machinery (ACM)*, pp. 83–90, Mar. 2015.
- [35] S. M. Anzalone, J. Xavier, S. Boucenna, L. Billeci, A. Narzisi, F. Muratori, D. Cohen, and M. Chetouani, "Quantifying patterns of joint attention during human-robot interactions : an application for autism spectrum disorder assessment," *Elsevier : Pattern Recognition Letters*, Mar. 2018.
- [36] B. Scassellati, L. Boccanfuso, C.-M. Huang, M. Mademtzi, M. Qin, N. Salomons, P. Ventola, and F. Shic, "Improving social skills in children with ASD using a long-term, in-home social robot," *Science Robotics : Human-Robot Interaction*, vol. 3, Aug. 2018.
- [37] M. J. Matarić, "Socially assistive robotics : Human augmentation versus automation," *Science Robotics : Human-Robot Interaction*, vol. 2, Mar. 2017.
- [38] E. S. Kim, L. D. Berkovits, E. P. Bernier, D. Leyzberg, F. Shic, R. Paul, and B. Scassellati, "Social robots as embedded reinforcers of social behavior in children with autism," *Journal of Autism and Developmental Disorders*, vol. 43, pp. 1038–1049, May 2013.
- [39] T. Belpaeme, J. Kennedy, A. Ramachandran, B. Scassellati, and F. Tanaka, "Social robots for education : A review," *Science Robotics : Human-Robot Interaction*, vol. 3, Aug. 2018.
- [40] I. A. Hameed, G. Strazdins, H. A. M. Hatlemark, I. S. Jakobsen, and J. O. Damdam, "Robots that can mix serious with fun," in *International Conference on Advanced Machine Learning Technologies and Applications*, pp. 595–604, Jan. 2018.
- [41] Lee, R. M., Robbins, and S. B., "Measuring belongingness : The social connectedness and the social assurance scales," *Journal of Counseling Psychology*, vol. 42, no. 2, pp. 232–241, 1995.
- [42] J. Kory and C. Breazeal, "Storytelling with robots : Learning companions for preschool children's language development," in *IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*, pp. 643–648, Aug. 2014.
- [43] S. Spaulding, G. Gorden, and C. Breazeal, "Affect-aware student models for robot tutors," in *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 643–648, 2016.
- [44] J. M. K. Westlund, S. Jeong, H. W. Park, S. Ronfard, A. Adhikari, P. L. Harris, D. DeSteno, and C. L. Breazeal, "Flat vs. expressive storytelling : Young children's learning and retention of a social robot's narrative," *Frontiers in Human Neuroscience*, vol. 11, June 2017.
- [45] L. George and A. Mazel, "Humanoid robot indoor navigation based on 2D bar codes : Application to the Nao robot," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Oct. 2013.

- [46] K. Jokinen, S. Nishimura, K. Watanabe, and T. Nishimura, "Human-robot dialogues for explaining activities," *9th International Workshop on Spoken Dialogue Systems Technology*, vol. 400, May 2018.
- [47] H. Modares, I. Ranatunga, F. L. Lewis, and D. O. Popa, "Optimized assistive human-robot interaction using reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 46, pp. 655–667, Mar. 2016.
- [48] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human-robot interaction," *Applied Mechanics Reviews*, vol. 70, Feb. 2018.
- [49] "The world health report : www.who.int/whr/2008/en," 2008.
- [50] N. Sylla, V. Bonnet, F. Colledani, and P. Fraise, "Ergonomic contribution of ABLE exoskeleton in automotive industry," *International Journal of Industrial Ergonomics*, vol. 44, no. 4, pp. 475–481, 2014.
- [51] J. Huang, W. Huo, W. Xu, S. Mohammed, and Y. Amirat, "Control of upper-limb power-assist exoskeleton using a human-robot interface based on motion intention recognition," *IEEE Transactions on Automation Science and Engineering*, vol. 12, pp. 1257–1270, Oct. 2015.
- [52] S. Barreca, S. L. Wolf, S. Fasoli, and R. Bohannon, "Treatment interventions for the paretic upper limb of stroke survivors : A critical review," *Neurorehabilitation and Neural Repair*, vol. 17, no. 4, pp. 220–226, 2003.
- [53] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human-robot collaboration in industrial settings : Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, June 2018.
- [54] M. Peshkin and J. E. Colgate, "Feature cobots," *MCB university press*, vol. 26, no. 5, pp. 335–341, 1999.
- [55] M. Peshkin, J. Colgate, W. Wannasuphprasit, C. Moore, R. Gillespie, and P. Akella, "Cobot architecture," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 377–390, 2001.
- [56] B. Claverie, B. L. Blanc, and P. Fouillat, "La cobotique," *Presses université de Bordeaux*, vol. 2, no. 44, pp. 203–214, 2013.
- [57] T. Moulières-seban, D. Bitonneau, and J. François Thibault, "Simulation for designing industrial cobotic systems," *30th International Seminar on Ergonomics*, May 2017.
- [58] G. Ganesh, A. Takagi, R. Osu, T. Yoshioka, M. Kawato, and E. Burdet, "Two is better than one : Physical interactions improve motor performance in humans," *Robotics for computational neuroscience*, Jan. 2014.
- [59] S. J. Lederman and R. L. Klatzky, "Haptic perception : A tutorial," *Attention, Perception & Psychophysics*, vol. 71, pp. 1439–1459, Sept. 2009.
- [60] O. Cohen, S. Druon, S. Lengagne, A. Mendelsohn, R. Malach, A. Kheddar, and D. Friedman, "fMRI robotic embodiment : A pilot study," in *IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, June 2012.
- [61] S. Mukherjee, D. Paramkusam, and S. K. Dwivedy, "Inverse kinematics of a Nao humanoid robot using Kinect to track and imitate human motion," in *International Conference on Robotics, Automation, Control and Embedded Systems (RACE)*, Feb. 2015.

- [62] G. Muller-Putz, A. Schwarz, J. Pereira, and P. Ofner, “From classic motor imagery to complex movement intention decoding : The noninvasive graz-BCI approach,” *Elsevier : Progress in Brain Research*, vol. 228, pp. 39–70, 2016.
- [63] S. Aliakbaryhosseinabadi, N. Jiang, A. Vuckovic, R. Lontis, K. Dremstrup, D. Farina, and N. Mrachacz-Kersting, “Detection of movement intention from movement-related cortical potentials with different paradigms,” *Jensen W., Andersen O., Akay M. (eds) Replace, Repair, Restore, Relie - Bridging Clinical and Engineering Solutions in Neurorehabilitation. Biosystems & Biorobotics*, vol. 7, pp. 237–244, Jan. 2014.
- [64] A. Mörtl, M. Lawitzky, A. Kucukyilmaz, M. Sezgin, C. Basdogan, and S. Hirche, “The role of roles : Physical cooperation between humans and robots,” *The International Journal of Robotics Research*, vol. 31, pp. 1656–1674, Aug. 2012.
- [65] D. P. McMullen, G. Hotson, K. D. Katyal, B. A. Wester, M. S. Fifer, T. G. McGee, A. Harris, M. S. Johannes, R. J. Vogelstein, A. D. Ravitz, W. S. Anderson, N. V. Thakor, and N. E. Crone, “Demonstration of a semi-autonomous hybrid brain-machine interface using human intracranial EEG, eye tracking, and computer vision to control a robotic upper limb prosthetic,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, pp. 784–796, July 2014.
- [66] M. Sarac, E. Koyas, A. Erdogan, M. Cetin, and V. Patoglu, “Brain computer interface based robotic rehabilitation with online modification of task speed,” in *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, pp. 1–7, June 2013.
- [67] Z. YanWei, S. Hao, Z. BeiChen, Z. Jun, and X. Zhang, “Weight-based sparse coding for multi-shot person re-identification,” *Science China : Information Sciences*, vol. 58, Oct. 2015.
- [68] S. Wang, D. Tao, and J. Yang, “Relative attribute SVM + learning for age estimation,” *IEEE Transactions on Cybernetics*, vol. 46, pp. 827–839, Mar. 2015.
- [69] L. Fang and S. Li, “Face recognition by exploiting local Gabor features with multitask adaptive sparse representation,” *IEEE Transactions on Instrumentation and Measurement*, vol. 64, pp. 2605–2615, Oct. 2015.
- [70] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “Netvlad : CNN architecture for weakly supervised place recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5297–5307, 2016.
- [71] S. Cao and N. Snavely, “Graph-based discriminative learning for location recognition,” *International Journal of Computer Vision*, vol. 112, pp. 239–254, Nov. 2014.
- [72] M. Cummins and P. Newman, “FAB-MAP : Probabilistic localization and mapping in the space of appearance,” *The International Journal of Robotics Research*, vol. 27, pp. 647–665, June 2008.
- [73] C. McManus, W. Churchill, W. Maddern, A. D. Stewart, and P. Newman, “Shady dealings : Robust, long-term visual localisation using illumination invariance,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014.
- [74] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt, “Scalable 6-DOF localization on mobile devices,” in *Computer Vision – ECCV 2014*, pp. 268–283, Springer International Publishing, 2014.
- [75] M. Aubry, B. C. Russell, and J. Sivic, “Painting-to-3d model alignment via discriminative visual elements,” *ACM Transactions on Graphics*, vol. 33, pp. 1–14, Apr. 2014.

- [76] P. Vepakomma, D. De, S. K. Das, and S. Bhansali, “A-wristocracy : Deep learning on wrist-worn sensing for recognition of user complex activities,” in *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 1–6, June 2015.
- [77] J. Qin, L. Liu, Z. Zhang, Y. Wang, and L. Shao, “Compressive sequential learning for action similarity labeling,” *IEEE Transactions on Image Processing*, vol. 25, pp. 756–769, Feb. 2016.
- [78] N. Y. Hammerla, S. Halloran, and T. Plötz, “Deep, convolutional, and recurrent models for human activity recognition using wearables,” *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.
- [79] Y. Kim and B. Toomajian, “Hand gesture recognition using micro-doppler signatures with convolutional neural network,” *IEEE Access*, vol. 4, pp. 7125–7130, 2016.
- [80] D. Cook, K. D. Feuz, and N. C. Krishnan, “Transfer learning for activity recognition : a survey,” *Knowledge and Information Systems*, vol. 36, pp. 537–556, June 2013.
- [81] L. Shao, S. Member, F. Zhu, and X. Li, “Transfer learning for visual categorization : A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 1019–1034, May 2014.
- [82] V. D. Kustikova, I. B. Meyerov, and N. Y. Zolotykh, “Video-based vehicle detection method,” *Pattern Recognition and Image Analysis*, vol. 24, pp. 588–592, Dec. 2014.
- [83] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1627–1645, Sept. 2010.
- [84] J. Shotton, A. Blake, and R. Cipolla, “Contour-based learning for object detection,” in *Tenth IEEE International Conference on Computer Vision (ICCV)*, vol. 1, pp. 503–510, 2005.
- [85] C. Hilario, J. Collado, J. Armingol, and A. de la Escalera, “Pyramidal image analysis for vehicle detection,” in *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pp. 88–93, 2005.
- [86] Y. Amit, *2D Object Detection & Recognition - Models, Algorithms & Networks*. MIT Press, 2002.
- [87] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*. Springer US, 1993.
- [88] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, “Scalable object detection using deep neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2155–2162, June 2014.
- [89] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Weakly supervised object recognition with convolutional neural networks,” in *Proc. of NIPS*, 2014.
- [90] P. Luo, Y. Tian, X. Wang, and X. Tang, “Switchable deep network for pedestrian detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2014.
- [91] H. Liu and S. Li, “Target detection using sparse representation with element and construction combination feature,” *IEEE Transactions on Instrumentation and Measurement*, vol. 64, pp. 290–298, Feb. 2015.
- [92] Z. Abderrahmane, G. Ganesh, A. Crosnier, and A. Cherubini, “Haptic zero-shot learning : Recognition of objects never touched before,” *Robotics and Autonomous Systems*, pp. 11–25, Mar. 2018.

- [93] Y. Fu, T. Xiang, Y.-G. Jiang, X. Xue, L. Sigal, and S. Gong, “Recent advances in zero-shot recognition,” *IEEE Signal Processing Magazine*, vol. 35, pp. 112–125, Oct. 2017.
- [94] N. Jarrassé, V. Sanguineti, and E. Burdet, “Slaves no longer : Review on role assignment for human-robot joint motor action,” *Adaptive Behavior*, vol. 22, pp. 70–82, Feb. 2014.
- [95] L. Aymerich-Franch and G. Ganesh, “The role of functionality in the body model for self-attribution,” *Elsevier : Neuroscience Research*, vol. 104, pp. 31–37, Mar. 2016.
- [96] G. Ganesh, T. Yoshioka, R. Osu, and T. Ikegami, “Immediate tool incorporation processes determine human motor planning with tools,” *Nature Communications*, July 2014.
- [97] A. Calanca, R. Muradore, and P. Fiorini, “A review of algorithms for compliant control of stiff and fixed-compliance robots,” *IEEE/ASME Transactions on Mechatronics*, vol. 21, pp. 613–624, Apr. 2016.
- [98] M. H. Raibert and J. J. Craig, “Hybrid position/force control of manipulators,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 103, no. 2, p. 126, 1981.
- [99] B. Busch, G. Maeda, Y. Mollard, M. Demangeat, and M. Lopes, “Postural optimization for an ergonomic human-robot interaction,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–8, Dec. 2017.
- [100] A. Shafti, A. Ataka, B. U. Lazpita, A. Shiva, H. Wurdemann, and K. Althoefer, “Real-time robot-assisted ergonomics,” in *IEEE ICRA*, vol. 3, Mar. 2019.
- [101] B. Busch, M. Toussaint, and M. Lopes, “Planning ergonomic sequences of actions in human-robot interaction,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.
- [102] A. Malaisé, P. Maurice, F. Colas, and S. Ivaldi, “Activity recognition for ergonomics assessment of industrial tasks with automatic feature selection,” *IEEE Robotics and Automation Letters*, vol. 4, pp. 1132–1139, Apr. 2019.
- [103] W. Kim, J. Lee, L. Peternel, N. Tsagarakis, and A. Ajoudani, “Anticipatory robot assistance for the prevention of human static joint overloading in human-robot collaboration,” *IEEE Robotics and Automation Letters*, vol. 3, pp. 68–75, July 2017.
- [104] L. Peternel, W. Kim, J. Babic, and A. Ajoudani, “Towards ergonomic control of human-robot co-manipulation and handover,” in *IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pp. 55–60, Nov. 2017.
- [105] A. Sobhee, S. Druon, and A. Crosnier, “Human-performed assembly task evaluation through a virtual operator,” in *IEEE International Conference on Advanced Robotics (ICAR)*, Sept. 2009.
- [106] T. Kruse, A. Pandey, R. Alami, and A. Kirsch, “Human-aware robot navigation : A survey,” *Elsevier : Robotics and Autonomous Systems*, vol. 61, pp. 1726–1743, June 2013.
- [107] K.-C. Huang, J.-Y. Li, and L.-C. Fu, “Human-oriented navigation for service providing in home environment,” in *Proceedings of SICE Annual Conference 2010*, pp. 1892–1897, Aug. 2010.
- [108] E. T. Hall, *The hidden dimension*. 1966.
- [109] M. L. Walters, M. A. Oskoei, D. S. Syrdal, and K. Dautenhahn, “A long-term human-robot proxemic study,” in *IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*, pp. 137–142, Aug. 2011.
- [110] E. Pacchierotti, H. Christensen, and P. Jensfelt, “Evaluation of passing distance for social robots,” in *IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*, pp. 315–320, Sept. 2006.

- [111] J. Mumm and B. Mutlu, “Human-robot proxemics : Physical and psychological distancing in human-robot interaction,” in *6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 331–338, Mar. 2011.
- [112] J. T. Butler and A. Agah, “Psychological effects of behavior patterns of a mobile personal robot,” *Autonomous Robots*, vol. 10, pp. 185–202, Mar. 2001.
- [113] N. Epley, A. Waytz, and J. T. Cacioppo, “On seeing human : A three-factor theory of anthropomorphism,” *Psychological Review*, vol. 114, no. 4, pp. 864–886, 2007.
- [114] F. Stulp, J. Grizou, B. Busch, and M. Lopes, “Facilitating intention prediction for humans by optimizing robot motions,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 2015.
- [115] L. P. Roberts, “Personality in the human robot interaction literature : A review and brief critique,” in *24th Americas Conference on Information Systems*, May 2018.
- [116] K. L. Koay, E. A. Sisbot, D. S. Syrdal, M. L. Walters, K. Dautenhahn, and R. Alami, “Exploratory study of a robot approaching a person in the context of handing over an object,” in *Multidisciplinary Collaboration for Socially Assistive Robotics, Papers from the 2007 AAAI Spring Symposium*, Mar. 2007.
- [117] A. F. Foka and P. E. Trahanias, “Predictive autonomous robot navigation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 490–495, Oct. 2002.
- [118] T. Ohki, K. Nagatani, and K. Yoshida, “Collision avoidance method for mobile robot considering motion and personal spaces of evacuees,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1819–1824, Oct. 2010.
- [119] M. Mori, “Bukimi no tani [the uncanny valley],” *Energy*, vol. 7, pp. 33–35, Jan. 1970.
- [120] A. K. Pandey and R. Alami, “A framework towards a socially aware mobile robot motion in human-centered dynamic environment,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5855–5860, Oct. 2010.
- [121] F. Badeig, I. Gebru, Q. Pelorson, X. Li, S. Arias, G. Evangelidis, V. Drouard, and R. Horaud, “A distributed architecture for interacting with Nao,” in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (ICMI)*, pp. 385–386, Nov. 2015.
- [122] “Caractéristiques Chorégraphe : www.generationrobots.com/fr/401618-suite-logicielle-pour-robot-humanoïde-programmable-nao-evolution.html.”
- [123] “Principales fonctions de ROS : www.generationrobots.com/fr/content/55-ros-robot-operating-system”
- [124] “Histoire de ROS : en.wikipedia.org/wiki/Robot_Operating_System.”
- [125] “Documentation ROS : wiki.ros.org/.”
- [126] “Liste des robots interfacés ROS : robots.ros.org/all/.”
- [127] “La nuit des fantômes (jeu de "Smart Games") : www.smartgames.eu/fr/jeux-pour-1-joueur/la-nuit-des-fant”
- [128] “Code Couleur (jeu de "Smart Games") : www.smartgames.eu/fr/jeux-pour-1-joueur/code-couleur.”
- [129] “IQ Fit (jeu de "Smart Games") : www.smartgames.eu/fr/jeux-pour-1-joueur/iq-fit.”

- [130] "IQ Twist (jeu de "Smart Games") : www.smartgames.eu/fr/jeux-pour-1-joueur/iq-twist."
- [131] "Camelot Jr (jeu de "Smart Games") : www.smartgames.eu/fr/jeux-pour-1-joueur/camelot-jr."
- [132] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, pp. 2280–2292, June 2014.
- [133] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transactions on Robotics*, vol. 25, pp. 670–685, jun 2009.
- [134] "Théorie des jeux : fr.wikipedia.org/wiki/th
- [135] "Matrice représentant le raisonnement issu de la théorie des jeux dans le cas du "dilemme du prisonnier" : <https://www.institut-pandore.com/economie/comprendre-theorie-des-jeux/>."
- [136] U. Chaudhary, N. Birbaumer, and A. Ramos-Murguialday, "Brain-computer interfaces for communication and rehabilitation," *Corrigendum : Brain-computer interfaces for communication and rehabilitation*, vol. 2, pp. 513–525, Sept. 2016.
- [137] D. J. McFarland and J. R. Wolpaw, "Brain-computer interfaces for communication and control," *Commun ACM*, pp. 60–66, Oct. 2011.
- [138] M. Witkowski, M. Cortese, M. Cempini, J. Mellinger, N. Vitiello, and S. R. Soekadar, "Enhancing brain-machine interface (BMI) control of a hand exoskeleton using electrooculography (EOG)," *Journal of NeuroEngineering and Rehabilitation*, vol. 11, no. 1, p. 165, 2014.
- [139] W. Hassani, S. Mohammed, and Y. Amirat, "Real-time EMG driven lower limb actuated orthosis for assistance as needed movement strategy," in *Robotics : Science and Systems IX*, June 2013.
- [140] "Processus accomplis par le BCI : www.mechaterrain.com/screenless-display/page/0/1."
- [141] G. Ganesh, K. Nakamura, S. Saetia, A. M. Tobar, E. Yoshida, H. Ando, N. Yoshimura, and Y. Koike, "Utilizing sensory prediction errors for movement intention decoding : A new methodology," *American Association for the Advancement of Science*, May 2018.
- [142] "Appareil de mesure pour FMRI : www.vumc.org/ccm/fmri-functional-magnetic-resonance-imaging."
- [143] "Différences entre EEG, FMRI et MRI : imotions.com/blog/eeg-vs-mri-vs-fmri-differences/."
- [144] "Appareil de mesure pour EOG : metrovision.fr/mv-eo-notice-fr.html."
- [145] "Appareil de mesure pour EMG : edu.mnhn.fr/pluginfile.php/12524/mod_resource/content/1/TP_electromyographie.pdf."

Annexe A

Informations brutes (mesures capteurs)

Nous allons commencer par présenter l'interface la plus utilisée pour assurer la communication entre l'humain et le robot dans le cadre d'une interaction entre les deux : Le Brain-Computer Interface (BCI) ou Brain-Machine Interface (BMI) (en français : Interface neuronale directe) est une interface de communication directe entre un cerveau et un dispositif externe (un ordinateur, robot, un système électronique, ...etc.) qui permet à ce dernier de savoir à quoi la personne pense. Ces systèmes peuvent être conçus dans le but d'étudier le cerveau, d'assister, d'améliorer ou de réparer des fonctions humaines de cognition ou d'actions défaillantes. Les méthodes de fonctionnement sont détaillées ainsi que les améliorations pouvant lui être apportées dans [136, 137]. La Figure A.1 montre tout le processus accompli par le BCI lequel consiste à extraire et traiter des données issues du cerveau puis d'en extraire les caractéristiques et les répartir en fonction de 4 des 5 sens de l'humain (l'odorat, la vue, l'ouïe et le toucher) et ce en les transmettant au dispositif externe de façon à lui apprendre ce à quoi pense l'humain.

Nous allons ensuite exposer les capteurs qui mesurent les signaux bioélectriques les plus utilisés, dans le cadre de l'interaction homme-robot, pour détecter ce que pense l'humain, permettant ainsi de réaliser la détection d'intention ou le contrôle d'exosquelette :

- FMRI : L'imagerie par résonance magnétique fonctionnelle (FMRI) est une application de l'imagerie par résonance magnétique permettant de visualiser, de manière indirecte, l'activité cérébrale. Il s'agit d'une technique d'imagerie utilisée pour l'étude du fonctionnement du cerveau. Elle consiste à enregistrer des variations hémodynamiques (variations des propriétés du flux sanguin) cérébrales locales minimes, lorsque ces zones sont stimulées. La Figure A.3 montre l'appareil utilisé pour faire le FMRI. La différence entre l'FMRI, l'MRI et l'EEG est montrée par la Figure A.4.
- EEG : L'électroencéphalographie (EEG) mesure l'activité électrique du cerveau, par un casque d'électrodes placé sur le cuir chevelu du sujet (cf. Figure A.2), produite lors de l'accomplissement d'une action.
- EOG : L'électrooculographie (EOG) est une technique de mesure du potentiel cornéorétinien debout qui existe entre l'avant et l'arrière de l'œil humain. Le signal qui en résulte est appelé électrooculogramme et mesure le potentiel de repos de l'œil. Ce type de signaux est utilisé pour le contrôle des exosquelettes [138]. Contrairement à l'électro-rétinogramme (ERG) qui est un examen électrophysiologique réalisé dans des services spécialisés de neurophysiologie clinique ou en ophtalmologie et qui permet de diagnostiquer certaines anomalies de la rétine, l'EOG ne mesure pas la réponse aux stimuli visuels individuels. La Figure A.5 montre l'appareil qui mesure le signal électrooculogramme.

- EMG : L'électromyographie (EMG) est un signal qui permet d'analyser l'activité électrique des nerfs et des muscles dans le but de contrôler, par exemple, un exosquelette de l'articulation du genou de l'utilisateur [139]. La Figure A.6 montre l'appareil qui mesure le signal EMG des muscles du bras.

Nous achevons cette partie en mettant en lumière un dispositif utilisé pour faire ressentir à l'humain qu'il effectue des actions qu'il n'est pas réellement en train de faire : La simulation vestibulaire galvanique (GVS) est un dispositif (cf. Figure A.2) qui se positionne sur le cerveau et envoie des impulsions électriques spécifiques au système vestibulaire qui est un organe sensoriel barosensible, situé dans l'oreille interne, qui contribue à la sensation de mouvement et à l'équilibre. Le GVS permet à un humain de ressentir qu'il est en train d'accomplir une action alors qu'en réalité il ne l'a jamais faite ; par exemple il permet de faire sentir à l'opérateur qu'il est assis sur un fauteuil tournant à gauche ou à droite.

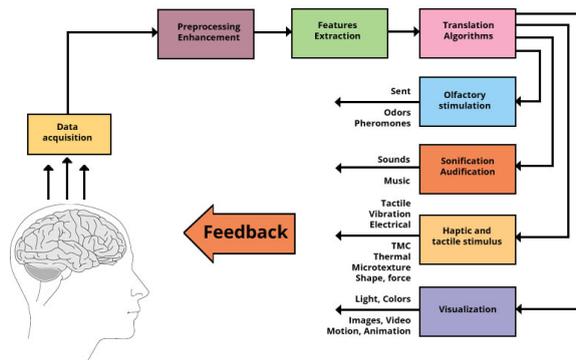


FIGURE A.1 – Processus accompli par le BCI [140]

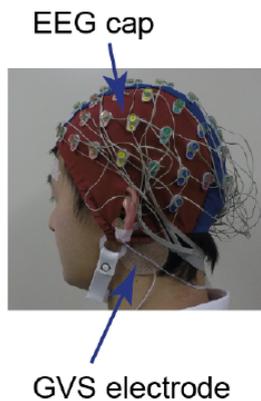


FIGURE A.2 – Disposition d'un casque d'électrodes pour la mesure des signaux EEG et d'une électrode GVS sur un humain [141]

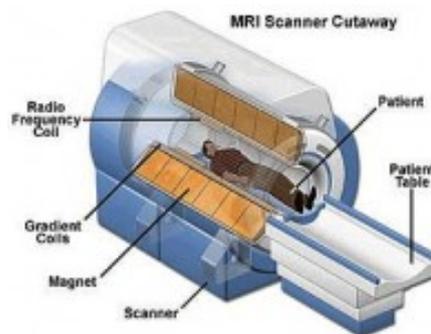


FIGURE A.3 – Appareil de mesure pour FMRI [142]

EEG vs. MRI vs. fMRI



The main differences

Each imaging methodology has distinct advantages and disadvantages, some of which are summarized below.

	EEG	MRI	fMRI
Temporal resolution	High	Low	Low
Spatial resolution	Low	High	High
Measures brain activity?	Directly	Only structure	Indirectly (BOLD response)
Level of expertise needed	Some training	Extensive training	Extensive training
Cost	Accessible to many researchers	Requires extensive funding	Requires extensive funding
Portability	Both fully portable and semi-portable devices available	Not portable	Not portable

FIGURE A.4 – Différences entre EEG, FMRI et MRI [143]



FIGURE A.5 – Appareil de mesure pour EOG [144]

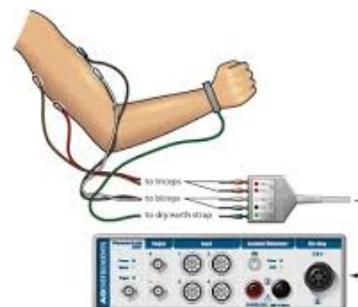


FIGURE A.6 – Appareil de mesure pour EMG [145]

Annexe B

Contrôleur de Nao

B.1 Mouvement d'un bras

B.1.1 Bras droit

On définit un point dans le repère monde situé au niveau du poignet droit du robot (c'est-à-dire qu'on fait un changement de repères entre le repère du poignet droit et le repère monde). Puis on calcule l'erreur entre le point qu'on veut atteindre et le point du poignet (on fait une transformation de repère entre le repère de référence, (la cheville droite) repère dans lequel est exprimé les points donnés par Aruco, et le repère monde) et on fait en sorte qu'elle tende vers zéro et ce en faisant tendre la jacobienne de la distance vers zéro pour que les 2 points soient superposés. Cette partie souscrit au topic `/goal_position_right` qui permet d'envoyer la position finale désirée de la main droite; de même elle publie sur le topic `/goal_position_right/distance` la distance entre la position supposée de la main, calculée par le contrôleur sachant la position initiale de la main et le mouvement des articulations effectuées et la position finale désirée.

B.1.2 Bras gauche

On suit la même procédure que pour le bras droit en remplaçant le poignet droit par le gauche. Cette partie souscrit au topic `/goal_position_left` qui permet d'envoyer la position finale désirée de la main gauche; de même elle publie sur le topic `/goal_position_left/distance` la distance entre la position supposée de la main, calculée par le contrôleur sachant la position initiale de la main et le mouvement des articulations effectuées et la position finale désirée.

B.2 Mouvement de la tête

Il faut d'abord définir 2 points pour former une droite dans la direction de la caméra (l'axe perpendiculaire à l'image) du robot dans le repère monde, qui est situé au niveau du bassin du robot mesuré, pour la position de départ du robot (c'est-à-dire qu'on fait un changement de repère entre le repère caméra et le repère monde). Puis on calcule l'erreur entre le point qu'on veut regarder et sa projection orthogonale sur la droite (on fait une transformation de repère entre le repère de référence (la cheville droite, repère dans lequel sont exprimés les points donnés par Aruco, et le repère monde) et on fait en sorte qu'elle tende vers zéro (la distance tend vers zéro pour que les 3 points soient alignés) et ce en faisant tendre la jacobienne de la projection orthogonale vers zéro. Cette partie souscrit au topic `/goal_looking_at` qui permet de suivre une pose initiale donnée quand celle-ci bouge et publie sur le topic `/goal_looking_at/distance` l'erreur.

Annexe C

Nœuds

C.1 Relation entre les différents nœuds

Nous allons expliciter la relation entre les différents nœuds et topics utilisés pour faire la manipulation (cf. Figure C.1) :

- Le nœud `"/solve_puzzle"` : Ce nœud joue le rôle de chef d'orchestre puisque c'est lui qui décide de quel cube chercher puis saisir et où le placer afin de résoudre le puzzle. Tout d'abord, il envoie l'identifiant du marqueur du cube qu'il faut saisir en premier au nœud `"/recherche_cube"` sur le topic `"/ID"` pour que la caméra se déplace et le retrouve ; une fois ceci fait, la position du marqueur est publiée sur `"/cube_pose"`, topic auquel souscrit `"/solve_puzzle"`. Ensuite il recommence l'étape précédente mais pour le marqueur de la position où il faut déposer le cube. Puis le nœud `"/solve_puzzle"` publie sur le topic `"/found_cube_position"`, les 2 positions 3D, auquel souscrit `"/saisie_cube"` permettant ainsi de déplacer le 1er cube. Lorsque la saisie et la pose du cube sont effectuées le nœud `"/saisie_cube"` publie 1 sur le topic `"/test_saisie"` pour que le nœud `"/solve_puzzle"` le sache et recommence les mêmes étapes afin de déplacer le cube suivant et ainsi de suite jusqu'à ce que tous les cubes soient à la bonne position.
- Le nœud `"/recherche_cube"` : Ce nœud sert à chercher un marqueur qui n'est pas détecté par la caméra comme dans la Figure 4.4. Le nœud d'Aruco `"/aruco_marker_publisher"` publie (envoie) les poses et les identifiants des marqueurs visibles par la caméra sur le topic `"/aruco_marker_publisher/markers"` et auquel le nœud `"/recherche_cube"` souscrit afin de faire bouger la tête de Nao (la caméra que nous utilisons est au niveau de la bouche du robot), en publiant sur le topic `"/goal_looking_at"` auquel souscrit le contrôleur dont le nœud est `"/controller"` et qui renvoie l'erreur entre la position du marqueur et l'axe de la caméra (perpendiculaire au plan de l'image et est au centre de cette dernière) sur le topic `"/goal_looking_at/distance"`, jusqu'à ce que le QR code, correspondant au cube que l'on veut déplacer ou celui de la position finale où l'on veut poser le cube, soit publié par le nœud d'Aruco. Une fois que cela est fait, le nœud `"/recherche_cube"` publie la pose du marqueur sur le topic `"/cube_pose"` auquel souscrit le nœud principal, `"/solve_puzzle"`.
- Le nœud `"/saisie_cube"` : Ce nœud se charge de déplacer le bras gauche, saisir l'objet et le déposer à la position désirée, ces positions sont déterminées par celles 3D des marqueurs, l'objet correspondant au cube et celui de la position finale. La partie du mouvement du bras gauche est effectuée par le contrôleur dont le nœud est `"/controller"` auquel il faut envoyer la position 3D du cube à saisir ou la position finale via le topic `"/goal-`

position_left" puis celui-ci renvoie les valeurs articulaires nécessaires pour atteindre la position désirée à "/naoqi_driver" via le topic "/joint_angles". Le contrôleur publie sur le topic "/goalposition_left/distance" la distance entre la position actuelle du bras (cette position est estimée par le contrôleur) du robot et sa position désirée, ce qui permet au nœud "/saisie_cube" de savoir si la position est atteinte ou non puisqu'il souscrit à ce topic. La saisie est effectuée par le contrôleur de Nao, dont le nœud est "/naoqi_driver", développé par SoftBank et qui ne permet de contrôler qu'une articulation à la fois. Pour cela, il suffit que "/saisie_cube" publie la valeur articulaire désirée ainsi que la vitesse du mouvement des doigts, qui est gérée par un même moteur et qui est donc considérée comme étant une seule articulation, sur le topic "/joint_angles". "/naoqi_driver" permet également de savoir la valeur articulaire de chacune des articulations du robot et les publie sur le topic "/joint_states" auquel souscrit le contrôleur "/controller".

Il reste encore à expliquer le rôle d'un nœud et de deux topics. Le Nœud "/viewer" souscrit au topic "/aruco_marker_publisher/result" sur lequel le nœud d'Aruco publie les marqueurs qui sont détectés afin qu'on puisse les visualiser et ainsi pouvoir se rendre compte de ceux qui ne le sont pas. Enfin le nœud "/naoqi_driver" publie sur le topic "/tf" les changements de repères nécessaires pour avoir la valeur de la pose fournie par Aruco dans un repère de référence. Nous avons pris comme repère de référence celui de la cheville droite de Nao.

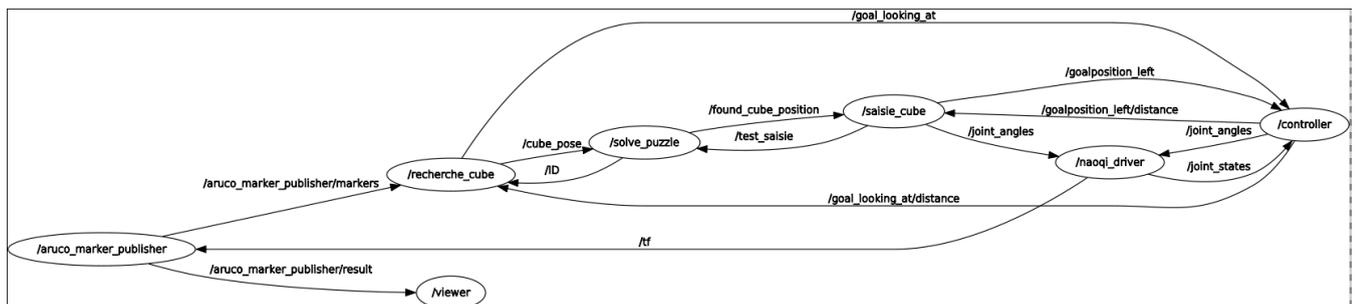


FIGURE C.1 – Schéma présentant la relation entre les différents nœuds (les cercles) et topics (les flèches)

C.2 Présentation des 3 nœuds principaux

C.2.1 Nœud "/recherche_cube"

Initialisation des variables

- Etat = 0 (c'est la variable qui en s'incrémentant changera l'état de la machine d'état)
- Limite_droite = position de la limite articulaire de la position de la tête de Nao du côté droit
- Limite_gauche = position de la limite articulaire de la position de la tête de Nao du côté gauche
- Test = -1 (permet de lancer et réinitialiser la machine d'état)

Publishers

- pub_tete : publie sur /goal Looking at (déplace la tête)

- pub_cube : publie sur /cube_pose (envoie la pose et l'ID du marqueur)

Subscribers

Pour chaque subscriber, il faut une fonction qui détermine comment les informations données par le subscriber seront attribuées aux variables propres au nœud.

Souscription au topic "/ID" a pour fonction "Identifiant"

- Erreur_tete = 100 (initialisation de l'erreur entre la position actuelle de la tête et la position désirée)
- si Test != ID (lorsqu'on récupère l'identifiant d'un marqueur, on lance la machine d'état)
 - on récupère l'ID du marqueur à trouver
 - Etat = 1
 - Trouve = 0 (initialisation d'un booléen qui sert de test pour savoir si le marqueur a été trouvé ou non)
 - pub_tete = Limite_gauche

Souscription au topic "/goal_looking_at/distance" a pour fonction "ErreurTete"

Récupération de la valeur de l'erreur de la position de la tête calculée par le contrôleur

Souscription au topic "/aruco_marker_publisher/markers" a pour fonction "Marqueur"

Si le marqueur se trouve dans l'image renvoyée par la caméra de Nao

- Récupération de la pose du marqueur à trouver
- Trouve = 1

Le cœur du programme

Tant que le nœud n'est pas tué (c'est-à-dire qu'on décide de fermer le programme), on fait les étapes suivantes :

- if Etat = 1 & erreur_tete < 5 cm
 - Etat = 2
 - pub_tete = Limite_droite
- elseif Etat = 2 & erreur_tete < 5 cm
 - Etat = 1
 - pub_tete = Limite_gauche
- elseif Etat = 3 & erreur_tete < 1 cm
 - pub_cube = la position et l'identifiant du marqueur recherché
 - Test = -1 (réinitialisation)
- if Trouve = 1
 - Etat = 3
 - pub_tete = la position du marqueur

Machine d'état

La Figure C.2 montre la machine d'état du nœud "/recherche_cube".

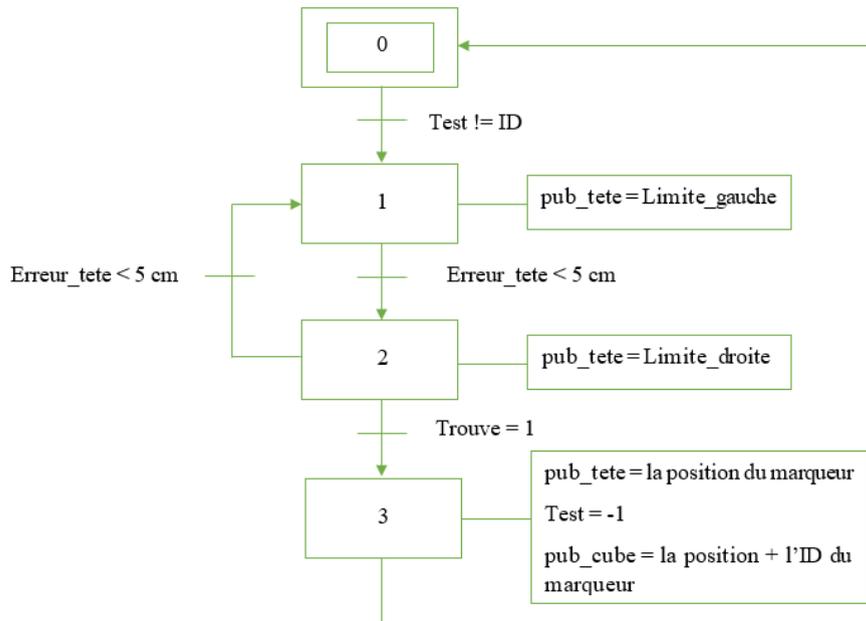


FIGURE C.2 – Machine d'état du nœud "/recherche_cube"

C.2.2 Nœud "/saisie_cube"

Initialisation des variables

- Etat = 0 (c'est la variable qui en s'incrémentant changera l'état de la machine d'état)
- offset = 7 cm (pour avoir assez d'espace pour ouvrir les doigts de Nao)
- epaisseur_cube = 4 cm (l'épaisseur d'un cube est de 3 cm à laquelle on ajoute 1 cm par mesure de sécurité)

Publishers

- pub_main : publie sur /joint_angles (ouvrir + fermer les doigts = l'articulation des doigts)
- pub_gauche : publie sur /goalposition_left (position finale de la main gauche)
- pub_voix : publie sur /speech (Nao parle)
- pub_saisie : publie sur /test_saisie (permet de savoir quand tout est fini)

Subscribers

Pour chaque subscriber, il faut une fonction qui détermine comment les informations données par le subscriber seront attribuées aux variables propres au nœud.

Souscription au topic "/found_cube_position" a pour fonction "RecupPostionCube"

- if Etat = 0
 - Pub_voix = Je vais essayer d'attraper l'objet
 - Etat = 1
 - On stocke la position initiale et finale du cube

Souscription au topic "/goalposition_left/distance" a pour fonction "Distance"

Stocker la valeur de la distance entre la position actuelle de la main et sa position finale

Le cœur du programme

Tant que le nœud n'est pas tué (c'est-à-dire qu'on décide de fermer le programme), on fait les étapes suivantes :

- if Etat = 1 (positionne la main au-dessus de l'objet avec un offset en z)
 - pub_gauche = la position du cube + l'offset en z
 - if distance > 1.5 cm
 - Etat = 2
- elseif Etat = 2 & distance < 1cm (ouvre la main)
 - Etat = 3
 - pub_voix = J'ai la main au-dessus de l'objet
 - pub_main = la valeur angulaire pour l'ouverture de la main (0.9)
- elseif Etat = 3 (baisse la main au niveau du cube)
 - pub_gauche = la position du cube
 - if distance > 1.5 cm
 - Etat = 4
- elseif Etat = 4 & distance < 1cm (attrape l'objet = ferme ses doigts)
 - Etat = 5
 - pub_main = la valeur angulaire pour la fermeture de la main (0.3)
 - pub_voix = Je pense avoir attrapé l'objet
- elseif Etat = 5 (relève la main de l'offset en z)
 - pub_gauche = la position du cube + l'offset en z
 - if distance > 1.5 cm
 - Etat = 6
- elseif Etat = 6 & distance < 1cm (dit qui a soulevé la main)
 - Etat = 7
 - pub_voix = J'ai soulevé l'objet
- elseif Etat = 7 (va au-dessus de la position finale avec un offset + l'épaisseur du cube en z)
 - pub_gauche = la position du cube + l'offset en z + l'épaisseur du cube en z
 - if distance > 1.5 cm
 - Etat = 8
- elseif Etat = 8 & distance < 1cm (dit que la main est au-dessus de la position finale)
 - Etat = 9
 - pub_voix = Je suis au-dessus de l'objectif
- elseif Etat = 9 (va au-dessus de la position finale avec l'épaisseur du cube en z)
 - pub_gauche = la position finale du cube + l'épaisseur du cube en z
 - if distance > 1.5 cm
 - Etat = 10
- elseif Etat = 10 & distance < 1cm (il ouvre ses doigts => il lâche le cube)
 - Etat = 11
 - pub_main = la valeur angulaire pour l'ouverture de la main (0.9)
 - pub_voix = Je suis sur l'objectif, je lâche l'objet
- elseif Etat = 11 (relève la main au-dessus de la position finale avec un offset + l'épaisseur du cube en z)
 - pub_gauche = la position finale du cube + l'offset en z + l'épaisseur du cube en z
 - if distance > 1.5 cm

- Etat = 12
- elseif Etat = 12 (retour à la position initiale du bras gauche)
 - pub_gauche = la position initiale du bras gauche
 - if distance > 1.5 cm
 - Etat = 13
- elseif Etat = 13 & distance < 1cm (il a fini, le programme se réinitialise dans l'attente de recommencer)
 - Etat = 0
 - pub_saisie = 1 (= l'opération est finie)
 - pub_voix = J'ai fini, j'ai posé l'objet

Machine d'état

La Figure C.3 montre la machine d'état du nœud "/saisie_cube".

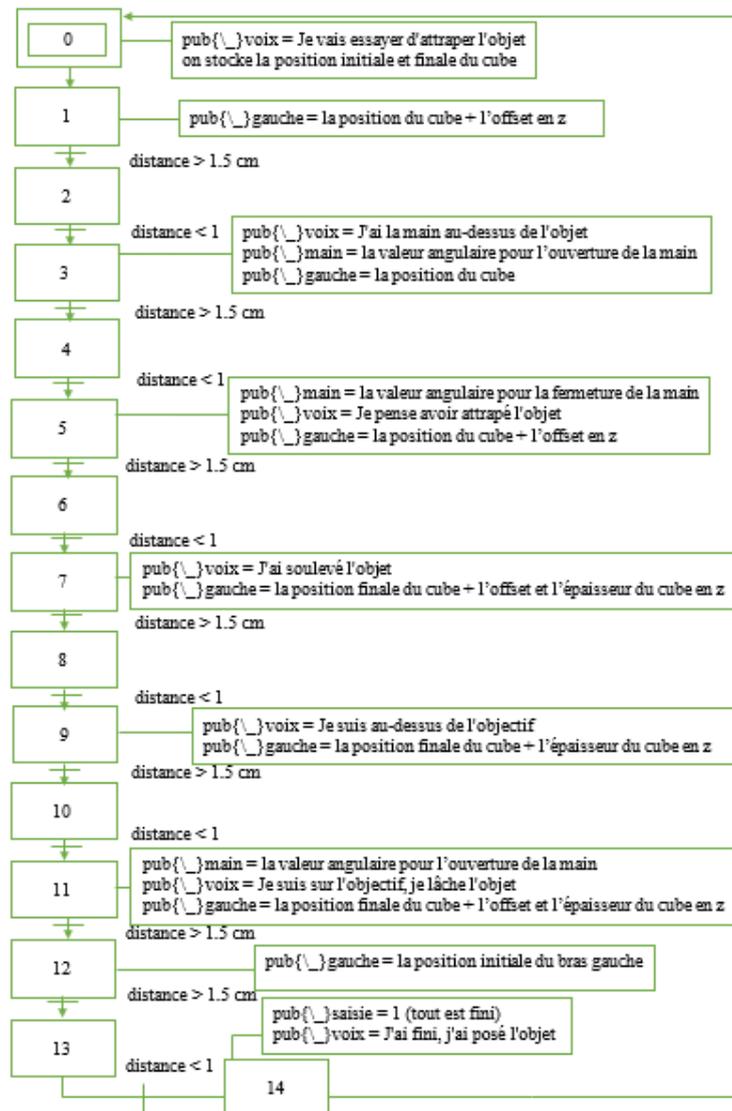


FIGURE C.3 – Machine d'état du nœud "/saisie_cube"

C.2.3 Nœud `"/solve_puzzle"`

Initialisation des variables

- Etat = 0 (c'est la variable qui en s'incrémentant changera l'état de la machine d'état)
- marqueur_detecte = 0 (initialisation de la variable indiquant l'ID détecté)

Publishers

- pub_ID : publie sur /ID (spécifie le marqueur à chercher)
- pub_cube_position = publie sur /found_cube_position (publie la position initiale et finale du cube)

Subscribers

Pour chaque subscriber, il faut une fonction qui détermine comment les informations données par le subscriber seront attribuées aux variables propres au nœud.

Souscription au topic `"/cube_pose"` a pour fonction `"DetectFinRecherche"`

- marqueur_detecte = ID du marqueur détecté
- Stocke la position du marqueur

Souscription au topic `"/test_saisie "` a pour fonction `"Distance"`

- si le cube a été déplacé
- fin_placement = 1

Le cœur du programme

Tant que le nœud n'est pas tué, (c'est-à-dire qu'on décide de fermer le programme), on fait les étapes suivantes :

- if Etat = 0 (recherche l'ID du cube à l'aide de recherche_cube)
 - pub_ID = l'ID du marqueur du 1er cube à trouver
 - fin_placement = 0 (initialisation d'un booléen pour savoir si le cube a été déplacé)
 - si marqueur_detecte = ID du marqueur du 1er cube à trouver (on a trouvé l'ID du cube)
 - Etat = 1
- elseif Etat = 1 (recherche l'ID de la position finale à l'aide de recherche_cube)
 - On stocke la position du cube
 - pub_ID = l'ID du marqueur de la position finale du 1er cube
 - si marqueur_detecte = ID du marqueur de la position finale du 1er cube
 - Etat = 2
- elseif Etat = 2 (une fois l'ID trouvé on envoie la position initiale et finale du cube à saisie_cube pour le déplacer)
 - On stocke la position finale du cube
 - pub_cube_position = la position initiale et finale du cube
 - si fin_placement = 1 (le 1er cube a été déplacé)
 - Etat = 3
 - fin_placement = 0 (reinitialisation)

- elseif Etat = 3 (recherche l'ID du cube à l'aide de recherche_cube)
 - pub_ID = l'ID du marqueur du 2ème cube à trouver
 - si marqueur_detecte = ID du marqueur du 2ème cube à trouver
 - Etat = 4
- elseif Etat = 4 (on a trouvé l'ID du cube)
 - On stocke la position du cube
 - pub_ID = l'ID du marqueur de la position finale du 2ème cube
 - si marqueur_detecte = ID du marqueur de la position finale du 2ème cube
 - Etat = 5
- elseif Etat = 5 (une fois l'ID trouvé on envoie la position initiale et finale du cube à saisie_cube pour le déplacer)
 - On stocke la position finale du cube
 - pub_cube_position = la position initiale et finale du cube
 - si fin_placement = 1 (le 2ème cube a été déplacé)
 - Etat = 0

Machine d'état

La Figure C.4 montre la machine d'état du nœud "/solve_puzzle".

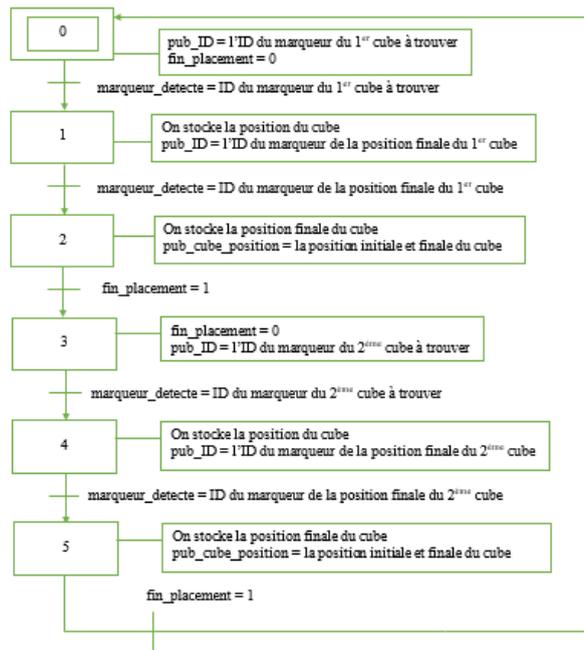


FIGURE C.4 – Machine d'état du nœud "/solve_puzzle"