

Titre: Conception d'un système intelligent d'ouverture de portes par
l'asservissement d'un bras robotique d'assistance sur fauteuil
roulant
Title:

Auteur: Gabriel Picard-Krashevski
Author:

Date: 2022

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Picard-Krashevski, G. (2022). Conception d'un système intelligent d'ouverture de
portes par l'asservissement d'un bras robotique d'assistance sur fauteuil roulant
[Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/10516/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/10516/>
PolyPublie URL:

**Directeurs de
recherche:** Sofiane Achiche
Advisors:

Programme: Génie mécanique
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Conception d'un système intelligent d'ouverture de portes par
l'asservissement d'un bras robotique d'assistance sur fauteuil roulant**

GABRIEL PICARD-KRASHEVSKI

Département de génie mécanique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie mécanique

Août 2022

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Conception d'un système intelligent d'ouverture de portes par
l'asservissement d'un bras robotique d'assistance sur fauteuil roulant**

présenté par **Gabriel PICARD-KRASHEVSKI**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

René MAYER, président

Sofiane ACHICHE, membre et directeur de recherche

Martin LEROUX, membre

DÉDICACE

Je dédie le travail de ce mémoire à ma mère, mon père et mon frère.

REMERCIEMENTS

Je remercie mon directeur de recherche Sofiane Achiche, professeur au Département de Génie mécanique à Polytechnique Montréal, pour le support et les conseils qu'il m'a apportés tout au long de ma recherche. Sans son soutien, cet ouvrage n'aurait pas vu le jour. Je le remercie également de m'avoir introduit aux plaisirs de l'enseignement par l'entremise de charges de cours, travaux dirigés et laboratoires.

Un grand merci également aux professeurs Maxime Raison, David Saussié et Abolfazl Mohebbi de Polytechnique Montréal pour m'avoir rencontré, au courant de ma recherche, et pour leurs conseils. Leur expertise m'a permis de mieux me diriger dans ma recherche et d'en apprendre davantage sur mon domaine.

Je tiens aussi à remercier tous mes collègues du laboratoire de Conception de Systèmes intelligents et Mécatroniques (CoSIM), du professeur Sofiane Achiche, pour leurs conseils et pour leur contribution à l'agréable atmosphère de travail dans laquelle j'ai évolué ces dernières années. Je tiens particulièrement à remercier mon collègue Cédric Leblond Ménard, doctorant en Génie mécanique, pour ses suggestions et son aide. Sa présence et tout le temps qu'il m'a octroyé m'ont permis de débattre sur mes idées et problèmes me permettant de mieux me diriger dans mon travail.

Je remercie aussi le Conseil de recherches en sciences naturelles et génie du Canada (CRSNG), le Front de Recherche du Québec – Nature et technologies (FRQNT) ainsi que mon directeur pour leur soutien financier.

Merci également à mes parents, mon frère et mes amis pour tout le support qu'ils m'ont apporté durant ces nombreuses années.

Finalement, je remercie le professeur René Mayer de Polytechnique Montréal ainsi que M. Martin Leroux, Développeur d'applications robotiques chez Kinova, pour avoir accepté d'être président et membre de mon jury respectivement.

RÉSUMÉ

Les manipulateurs sériels d'assistance sont des outils bénéfiques pour les personnes vivant avec des déficiences neuromotrices. Pour toujours être à la portée de l'utilisateur, certains des manipulateurs, dont le manipulateur Jaco², ont même été développés pour être montés sur le fauteuil roulant du patient. Effectivement, ceux-ci permettent aux utilisateurs d'accomplir plusieurs tâches de la vie quotidienne nécessitant généralement l'assistance d'un soignant ou d'un membre de la famille.

Cependant, certaines tâches quotidiennes restent tout de même difficilement réalisables à l'aide de ces manipulateurs ; l'ouverture de portes en est un exemple. En effet, face à cette tâche, les utilisateurs nécessitent habituellement de l'assistance humaine même lorsque le manipulateur est disponible. Pour pallier ce problème, l'objectif principal de ce travail est de développer un système intelligent d'assistance à l'ouverture des portes pour les utilisateurs en fauteuil roulant en asservissant uniquement le manipulateur sériel d'assistance. Le manipulateur utilisé dans ce travail de maîtrise est le Jaco² 6 DDL de Kinova Inc.

La solution présentée est composée de deux composantes principales. La première est un système de stéréovision supporté par l'analyse intelligente de scènes tridimensionnelles par vision artificielle pour déterminer la nature et la position des poignées. Celle-ci est munie d'un système de détection et de classification. La détection est réalisée en combinant deux méthodes : une détection à l'aide d'un algorithme de reconnaissance Faster R-CNN ainsi qu'une détection à l'aide de l'information tridimensionnelle de la scène. Ces deux méthodes sont combinées pour éliminer la détection de faux positifs. Malheureusement, dû à la taille limitée de la base de données utilisée pour l'entraînement, l'algorithme Faster R-CNN réalise régulièrement de mauvaises détections. Malgré cela, grâce à l'ajout de l'information tridimensionnelle, les résultats démontrent un taux de réussite de 0.9 pour la détection des poignées de porte rondes et de 1 pour les poignées à levier. En ce qui concerne la classification, celle-ci est réalisée à l'aide de descripteurs HOG et a été développée pour reconnaître les poignées rondes, à levier, et à tirer. Avec l'ensemble des expérimentations ne générant aucune erreur de classification, cette méthode s'est avérée 100% efficace pour déterminer la classe des poignées.

La deuxième est formée de trois systèmes de contrôle du manipulateur, permettant respectivement 1) d'atteindre la poignée, 2) de l'opérer, et 3) d'ouvrir la porte. Pour atteindre la poignée, la boucle

de contrôle mise en place a été munie d'un PID, d'une compensation gravitationnelle, et d'une action anticipative pour compenser les forces causées par les inerties internes. Pour les deux autres systèmes, un contrôle par impédance sans retour de force a été implémenté, muni d'un amortissement sur la vitesse angulaire des actuateurs pour rendre le système plus sécuritaire. De plus, pour opérer les poignées rondes, l'utilisation d'un intégrateur est aussi ajoutée pour faciliter la tâche. Suite aux simulations, les résultats démontrent que le système atteint en tout temps la poignée avec une erreur de position inférieure à 1mm et une erreur d'orientation inférieure à 0.05° . Le système est également en mesure d'opérer les poignées rondes et à levier avec un taux de réussite de 0.96 et 1 respectivement. Finalement, le système parvient à assister l'utilisateur lors de l'ouverture de la porte lui permettant d'accomplir cette tâche en moins de 30 secondes en tout temps. Toutefois, les résultats démontrent également que l'orientation de l'effecteur est importante pour que celui-ci parvienne à maintenir sa prise sur la poignée lors de situations réelles. Effectivement, dans les situations où le poignet de l'effecteur ne possédait pas un actuateur perpendiculaire au sol, il était facile pour l'utilisateur d'exercer des forces sur la poignée pouvant atteindre plusieurs centaines de Newtons.

ABSTRACT

Assistive serial manipulators are beneficial tools for people living with neuromotor impairments. In order to be within reach of the user, some of the manipulators, including the Jaco² manipulator, have been specifically developed to be mounted on the patient's wheelchair. Indeed, these allow users to accomplish several tasks pertaining to their daily routine, generally requiring the assistance of a caregiver or a family member.

However, some daily tasks remain difficult to perform using these manipulators. An example of this is opening doors. Indeed, faced with this task, users usually require human assistance even when the manipulator is available. To overcome this problem, the main objective of this work is to develop an intelligent door opening assistance system for wheelchair users by subjugating only the serial assistance manipulator. The manipulator used in this mastery work is the Jaco² 6 DOF manipulator of Kinova Inc.

The presented solution is composed of two main components. The first is a stereovision system supported by the intelligent analysis of three-dimensional scenes using artificial vision to determine the nature and position of the handles. It is equipped with a detection and classification system. Detection is performed by combining two methods: detection using a Faster R-CNN recognition algorithm and detection using three-dimensional information from the scene. These two methods are combined to eliminate the detection of false positives. Unfortunately, due to the limited size of the database used for training, the Faster R-CNN algorithm regularly makes erroneous detections. However, thanks to the addition of three-dimensional data, the results show a success rate of 0.9 for the detection of round and of 1 for the detection of lever door handles despite this difficulty. As far as classification is concerned, this is done using HOG descriptors and was developed to recognize round, lever and pull handles. With all the experiments generating no classification errors, this method proved to be 100% effective in determining the class of handles.

The second is made up of three manipulator control systems, respectively allowing 1) to reach the handle, 2) to operate it and 3) to open the door. To reach the handle, the control loop set up was provided with a PID, a gravitational compensation, and a feedforward action to compensate for the forces caused by the internal inertias. For the other two systems, impedance control without force feedback was implemented with damping on the angular speed of the actuators to make the system safer. In addition, to operate the round handles, the use of an integrator is also added to facilitate

the task. Following the simulations, the results show that the system reaches the handle at all times with a position error of less than 1mm and an orientation error of less than 0.05° . The system is also able to operate round and lever handles with a success rate of 0.96 and 1 respectively. Finally, the system manages to assist the user when opening the door allowing him to accomplish this task in less than 30 seconds at all times. However, the results also demonstrate that the orientation of the end-effector is important for the latter to be able to maintain its grip on the handle during real-life situations. Indeed, in situations where the wrist of the effector did not have an actuator perpendicular to the ground, it was easy for the user to exert forces on the handle that could reach several hundred Newtons.

TABLE DES MATIÈRES

DÉDICACE.....	III
REMERCIEMENTS	IV
RÉSUMÉ.....	V
ABSTRACT	VII
TABLE DES MATIÈRES	IX
LISTE DES TABLEAUX.....	XIII
LISTE DES FIGURES.....	XV
LISTE DES SIGLES ET ABRÉVIATIONS	XVIII
LISTE DES ANNEXES	XIX
CHAPITRE 1 INTRODUCTION.....	1
CHAPITRE 2 REVUE DE LITTÉRATURE ET OBJECTIFS	3
2.1 Manipulateur sériel d’assistance	3
2.2 Survol du processus d’ouverture de portes	4
2.3 Manipulateurs mobiles d’ouverture de porte	5
2.3.1 Détection de la poignée	5
2.3.2 Système de contrôle du manipulateur sériel.....	9
2.4 Reconnaissance d’objets par ordinateur	10
2.4.1 Reconnaissance par réseau neuronal convolutif.....	11
2.4.2 Classification par un descripteur	14
2.5 Contrôle et cinématique du manipulateur	14
2.5.1 Cinématique inverse.....	14
2.5.2 Contrôle du manipulateur.....	16
2.6 Objectif principal et objectifs spécifiques	23

CHAPITRE 3	MÉTHODOLOGIE	24
3.1	Cahier des charges	24
3.1.1	Choix du manipulateur sériel d'assistance	24
3.1.2	Étapes du processus d'ouverture d'une porte	24
3.1.3	Survol du manipulateur Jaco ²	25
3.1.4	Survol des dispositifs d'ouverture	27
3.1.5	Fonctions du système	27
3.2	Détection et classification de la poignée de porte	28
3.2.1	Survol de la solution utilisée	28
3.2.2	Position de la caméra	31
3.2.3	Reconnaissance de la poignée	32
3.2.4	Classification de la poignée	34
3.2.5	Position de la poignée	35
3.3	Simulation et modélisation	36
3.3.1	Simulation	37
3.3.2	Modélisation	37
3.4	Systèmes de contrôle	40
3.4.1	Atteindre le dispositif d'ouverture	40
3.4.2	Ouverture de la porte	42
3.4.3	Opérer le dispositif d'ouverture	44
3.5	Planification de la trajectoire	45
3.5.1	Atteindre le dispositif d'ouverture	45
3.5.2	Opérer le dispositif d'ouverture	48
3.5.3	Ouvrir la porte	49

3.6	Expérimentations.....	50
3.6.1	Détection et classification de poignées de porte	52
3.6.2	Systèmes de contrôle	54
CHAPITRE 4 RÉSULTATS ET ANALYSE.....		59
4.1	Détection et classification de la poignée de porte	59
4.1.1	Entrainement de l'algorithme Faster R-CNN et HOG	59
4.1.2	Détection des poignées de porte.....	60
4.1.3	Classification des poignées de portes.....	66
4.2	Systèmes de contrôle	66
4.2.1	Atteindre le dispositif d'ouverture	66
4.2.2	Opérer le dispositif d'ouverture	70
4.2.3	Ouvrir la porte	73
CHAPITRE 5 DISCUSSION		79
5.1	Détection et classification de la poignée de porte	79
5.1.1	Détection de la poignée	79
5.1.2	Classification de la poignée.....	81
5.2	Systèmes de contrôle.....	82
5.2.1	Atteindre le dispositif d'ouverture	82
5.2.2	Opérer le dispositif d'ouverture	82
5.2.3	Ouvrir la porte	83
5.2.4	Retour sur l'ensemble des systèmes de contrôle	83
5.3	Limitations et modifications.....	85
5.3.1	Détection et classification de la poignée de porte	85
5.3.2	Systèmes de contrôle.....	85

5.4	Retour sur les objectifs.....	86
CHAPITRE 6	CONCLUSION ET RECOMMANDATIONS	88
RÉFÉRENCES.....		90
ANNEXES		98

LISTE DES TABLEAUX

Tableau 2.1 Liste de méthodes permettant d'ajuster les gains d'un correcteur PID.	19
Tableau 3.1 Liste des étapes importantes à accomplir	25
Tableau 3.2 Les classes de dispositifs d'ouverture les plus communes	27
Tableau 3.3 Fonctions et sous-fonctions recherchées du système à développer.....	28
Tableau 3.4 Modèles Faster R-CNN dans Torchvision	33
Tableau 4.1 Comparaison des scores mAP	59
Tableau 4.2 Résultats de la détection des poignées rondes	60
Tableau 4.3 Résultats de la détection des poignées à levier.....	60
Tableau 4.4 Résultats de la détection des poignées à tirer	61
Tableau 4.5 Erreurs sur les 100 essais pour atteindre une poignée à levier *	68
Tableau 4.6 Erreurs sur les 100 essais pour atteindre une poignée ronde *	68
Tableau 4.7 Erreurs sur les 100 essais pour atteindre une poignée à tirer *	69
Tableau 4.8 Erreurs angulaires sur les 100 essais des poignées à levier.....	70
Tableau 4.9 Erreurs angulaires sur les 100 essais des poignées rondes	70
Tableau 4.10 Pourcentage de conclusion et d'ouverture de la porte réussi sur les 100 essais.....	70
Tableau 4.11 Taux de réussite ainsi que le temps nécessaire moyen et maximal obtenu pour ouvrir la porte lorsque le fauteuil se situe à diverses positions par rapport à la porte	74
Tableau 4.12 Taux de réussite ainsi que le temps nécessaire moyen et maximal obtenu pour ouvrir la porte lorsque le fauteuil se situe complètement devant la porte.....	74
Tableau 4.13 Forces et couples moyens ressentis par la poignée sur les 10 essais lorsque le fauteuil se situe à diverses positions par rapport à la porte *	75
Tableau 4.14 Forces et couples moyens ressentis par la poignée sur les 10 essais lorsque le fauteuil se situe complètement devant la porte *	76
Tableau B.1 Dimensions du Jaco ²	100

Tableau B.2 Limites angulaires.....	100
Tableau B.3 Paramètre pour DH.....	100
Tableau B.4 Relation entre DH et le réel	100
Tableau B.5 Paramètres pour la dynamique du Jaco ² (masse en kilogrammes, centre de masse en mètres et l'inertie est en kg*m ²)	101
Tableau C.1 Positions intermédiaires utilisé pour atteindre la poignée.....	102
Tableau D.1 Scores du Faster R-CNN fournis par l'évaluateur COCO.....	103
Tableau E.1 Gains utilisés pour atteindre la poignée de porte.....	104
Tableau E.2 Gains utilisés pour opérer la poignée de porte.....	104
Tableau E.3 Gains utilisés pour ouvrir la porte.....	104

LISTE DES FIGURES

Figure 2.1 Modèle d'un manipulateur Jaco de Kinova sur un fauteuil roulant	3
Figure 2.2 Analyse hiérarchique de tâches pour ouvrir puis traverser une porte	4
Figure 2.3 Capteur Kinect de Microsoft.....	6
Figure 2.4 Caméra Realsense D435	6
Figure 2.5 Reconnaissance d'objets par Faster R-CNN	12
Figure 2.6 Reconnaissance d'objets par YOLO	12
Figure 2.7 Exemple de reconnaissance par Mask R-CNN.....	13
Figure 2.8 Exemple de contrôle de position angulaire avec correcteur PID	17
Figure 2.9 Exemple de contrôle de position cartésienne avec correcteur PID.....	18
Figure 2.10 Force appliquée sur une masse retenue par un ressort et un amortisseur	21
Figure 2.11 Exemple de contrôle de position par impédance sans retour de force	22
Figure 3.1 Poignée ronde	27
Figure 3.2 Poignée à levier.....	27
Figure 3.3 Poignée à tirer	27
Figure 3.4 Fauteuil vu de dessus	31
Figure 3.5 Configuration complète	31
Figure 3.6 Schéma pour calculer l'angle et la hauteur de la caméra	32
Figure 3.7 Solution utilisée pour corriger la position d'une poignée à tirer sur le masque de la carte de profondeur. a) Illustre la photo utilisée pour la détection. b) Illustre le masque de la poignée avec le point central en rouge de celui-ci. c) Est la même illustration qu'en b), mais avec la droite horizontale passant par le point central et avec la position des pixels le plus à gauche et à droite sur cette droite. d) Illustre le nouveau centre de la poignée	36
Figure 3.8 Structure du système de contrôle pour atteindre la poignée	40
Figure 3.9 Structure du système de contrôle pour ouvrir la porte.....	43

Figure 3.10 Structure du système de contrôle pour opérer une poignée ronde	45
Figure 3.11 Décortication de la trajectoire. a) Illustre la position maison du manipulateur, b) la position intermédiaire, c) la position devant la poignée, et d) la position sur la poignée	46
Figure 3.12 Orientations de l'effecteur pour chaque classe de poignée. a), b) et c) illustrent respectivement l'orientation de l'effecteur pour les poignées à levier, ronde, et à tirer	46
Figure 3.13 Exemple d'accélération.....	48
Figure 3.14 Exemple de vitesses	48
Figure 3.15 Région de protection du fauteuil.....	50
Figure 3.16 Schéma décrivant l'ensemble du système	51
Figure 3.17 Installation utilisée pour détecter et classer les poignées de porte	52
Figure 3.18 Expérimentation utilisant l'installation	52
Figure 3.19 Calcul de l'IoU	54
Figure 3.20 Fauteuil partiellement à l'extérieur de la porte.....	58
Figure 3.21 Fauteuil complètement devant la porte	58
Figure 4.1 Détection de la poignée ronde numéro 7. a) illustre la détection réalisée par Faster R-CNN. b) et c) illustrent respectivement le masque de la poignée détecté et désiré	62
Figure 4.2 Détection de la poignée ronde 1 par Faster R-CNN	62
Figure 4.3 Masques de détection de la poignée ronde numéro 6. a) et b) illustre respectivement le masque de la poignée détecté et désiré.....	63
Figure 4.4 Détection de la poignée à levier numéro 1. a) illustre la détection réalisée par Faster R-CNN. b) et c) illustrent respectivement le masque de la poignée détecté et désiré	64
Figure 4.5 Détection de la poignée à tirer 6 par Faster R-CNN.....	64
Figure 4.6 Détection de la poignée à tirer 8 par Faster R-CNN.....	64
Figure 4.7 Masques de détection de la poignée à tirer numéro 2. a) et b) illustrent respectivement le masque de la poignée détecté et désiré.....	65

Figure 4.8 Masques de détection de la poignée à tirer numéro 4. a) et b) illustrent respectivement le masque de la poignée détecté et désiré.....	66
Figure 4.9 Trajectoire désirée et obtenue de l'essai causant la plus grande erreur angulaire	69
Figure 4.10 Angle exercé sur la poignée à levier en fonction de la résistance du ressort et de la distance entre celle-ci et la base du Jaco ² . a) et b) sont, respectivement, les cas étudiés lorsque la poignée se trouvait à gauche et à droite	71
Figure 4.11 Angle exercé sur la poignée ronde en fonction de la résistance du ressort et de la distance entre celle-ci et la base du Jaco ² . a) et b) sont, respectivement, les cas étudiés lorsque la poignée se trouvait à gauche et à droite	71
Figure 4.12 Erreur angulaire des poignées à levier situées à droite en fonction de la résistance du ressort et de la distance entre celle-ci et la base du Jaco ²	73
Figure 5.1 Exemple d'image d'armoire de la base de données utilisée	80
Figure 5.2 Poignée à levier 3.....	81
Figure 5.3 Poignée à levier 9.....	81
Figure A.1 Structure de la modélisation du Jaco ² sur Simscape.....	98
Figure A.2 Structure de la modélisation du fauteuil sur Simscape	98
Figure A.3 Structure de la modélisation de la porte et de la poignée sur Simscape	99
Figure B.1 Dimension du Jaco ²	100
Figure B.2 Structure des coordonnées pour DH.....	100

LISTE DES SIGLES ET ABRÉVIATIONS

AHT	Analyse hiérarchique de tâche
DDL	Degrés de liberté
DH	Denavit-Hartenberg
HOG	Histogramme de gradient orienté
IoU	Intersection over Union
mAP	Mean Average Precision
MSA	Manipulateur sériel d'assistance
RNC	Réseau neuronal convolutif

LISTE DES ANNEXES

Annexe A	Structure de la Modelisation dans Simscape	98
Annexe B	informations du manipulateur Jaco ²	100
Annexe C	Positions intermédiaires pour atteindre la poignée	102
Annexe D	Scores du Faster R-CNN fournis par l'évaluateur COCO	103
Annexe E	Gains utilisés pour les systèmes de contrôle	104

CHAPITRE 1 INTRODUCTION

Les personnes atteintes de maladies neuromusculaires graves, ou d'autres handicaps sévères, ont généralement besoin d'une supervision ou d'une assistance humaine pour accomplir les tâches de leur vie quotidienne, entraînant ainsi, entre autre, des dépenses importantes tant pour le système de santé que pour les familles des patients [1] en plus d'exacerber le sentiment de dépendance sur autrui. Cependant, dans le but d'améliorer leur autonomie, plusieurs progrès technologiques ont vu le jour, augmentant ainsi la qualité de vie de ces patients. Par exemple, les manipulateurs sériels d'assistance sont de nos jours des outils très bénéfiques pour les personnes vivant avec des déficiences neuromotrices [2]. En effet, des études ont démontré que ceux-ci permettaient de réduire de façon considérable le niveau d'assistance nécessaire provenant du soignant [3], [4]. Pour ces raisons, certaines entreprises ont trouvé des solutions pour permettre leurs manipulateurs de suivre l'utilisateur tout au cours de la journée. Ceci est, entre autres, le cas du manipulateur Jaco², développé par Kinova robotics inc [5], conçu pour être monté sur le fauteuil roulant motorisé du patient. Dans cette configuration, le manipulateur est disponible en tout temps pour aider l'utilisateur dans ses tâches quotidiennes.

Toutefois, certaines tâches sont difficilement réalisables à l'aide du Jaco², dont l'ouverture de porte. Effectivement, face à cette tâche, les utilisateurs nécessitent habituellement de l'assistance même lorsque le manipulateur est disponible [6]. Il faut noter que cette action qui semble simple, nécessite une coordination entre le manipulateur et le fauteuil roulant.

Pour pallier ce problème, certaines études ont mis au point un fauteuil muni d'un manipulateur capable d'accomplir la tâche d'ouvrir la porte à la place de l'utilisateur [7].

Cependant, cette solution n'est pas idéale pour les utilisateurs d'un point de vue économique. En effet, une telle solution obligerait le patient à remplacer son fauteuil par un autre, développé spécifiquement pour ouvrir des portes, ou à exécuter des modifications sur le fauteuil existant. De ce fait, l'objectif principal de ce travail est de développer un système intelligent d'assistance à l'ouverture des portes pour les utilisateurs en fauteuil roulant en asservissant uniquement le manipulateur sériel d'assistance Jaco² 6 DDL de Kinova inc. fixé sur le fauteuil. Ce mémoire présentera donc les sous-systèmes développés pour atteindre cet objectif. Ceux-ci sont conçus d'un système de stéréovision supporté par l'analyse intelligente de scènes tridimensionnelles par vision artificielle pour déterminer la nature et la position des poignées ainsi que de trois systèmes de

contrôle du manipulateur, permettant respectivement d'atteindre la poignée, de l'opérer et d'ouvrir la porte. Le Chapitre 2 présentera la revue de littérature et les objectifs du projet, le Chapitre 3 contiendra la méthodologie adoptée ainsi qu'une explication sur les expérimentations effectuées, le Chapitre 4 présente les résultats obtenus suivis d'une analyse de ceux-ci, et finalement le Chapitre 5 fera un retour sur les résultats et les analyses.

CHAPITRE 2 REVUE DE LITTÉRATURE ET OBJECTIFS

2.1 Manipulateur sériel d'assistance

Les manipulateurs sériels sont des appareils robotisés capables de manipuler des objets. Ceux-ci possèdent une seule chaîne cinématique en boucle ouverte débutant de la base jusqu'à l'effecteur, au long de laquelle sont positionnés les actionneurs. Ils font généralement partie d'une des deux catégories suivantes : les manipulateurs sériels industriels et les manipulateurs sériels d'assistance (MSA). Comme leur nom l'indique, les manipulateurs sériels d'assistance sont conçus dans le but d'assister les individus dans leurs tâches quotidiennes de façon sécuritaire. Contrairement aux modèles industriels qui nécessitent une distance d'utilisation par l'opérateur pour assurer un environnement sûr, ceux-ci sont conçus pour être en contact physique avec l'humain. Pour cette raison, ceux-ci sont d'excellents outils pour améliorer la qualité de vie des individus souffrant d'un handicap aux membres supérieurs [8]. Comme conclu dans l'étude de Laffont et coll. (2009) [2]: utiliser un MSA peut s'avérer bénéfique pour les personnes vivant avec des handicaps tels que la dystrophie musculaire, les lésions de la moelle épinière, l'amyotrophie spinale et d'autres déficiences neuromotrices. De plus, Romer et coll. (2005) [3] et Maheu et coll. (2011) [4] ont mesuré une réduction potentielle de l'assistance du soignant de 0,7 à 1,8 h par jour pour les patients possédant un MSA.



Figure 2.1 Modèle d'un manipulateur Jaco de Kinova sur un fauteuil roulant

L'un de ces MSAs est le Jaco², développé par Kinova robotics inc [5]. Ce MSA a été conçu pour être monté sur un fauteuil roulant motorisé, comme illustré à la Figure 2.1, et contrôlé par une

manette fixée sur l'appui-bras. Grâce à cette configuration, celui-ci reste accessible à l'utilisateur en tout temps. Il a été établi, qu'à court terme [9] et à long terme [6], les utilisateurs du Jaco² étaient généralement satisfaits de leur appareil et que ceux-ci faisaient l'expérience d'impacts psychosociaux positifs. De plus, ces études ont démontré une amélioration de l'autonomie des utilisateurs lorsque ceux-ci accomplissaient des tâches quotidiennes avec le manipulateur. Toutefois, certaines tâches quotidiennes sont difficilement réalisables à l'aide du Jaco² [6]. L'ouverture d'une porte en est un exemple. Effectivement, l'étude de Beaudoin et coll. (2019) [6] démontre que la majorité des utilisateurs ont besoin d'assistance pour ouvrir une porte malgré la disponibilité du Jaco². Une amélioration du système est donc nécessaire pour permettre aux utilisateurs d'accomplir, de façon autonome, cette tâche quotidienne.

2.2 Survol du processus d'ouverture de portes

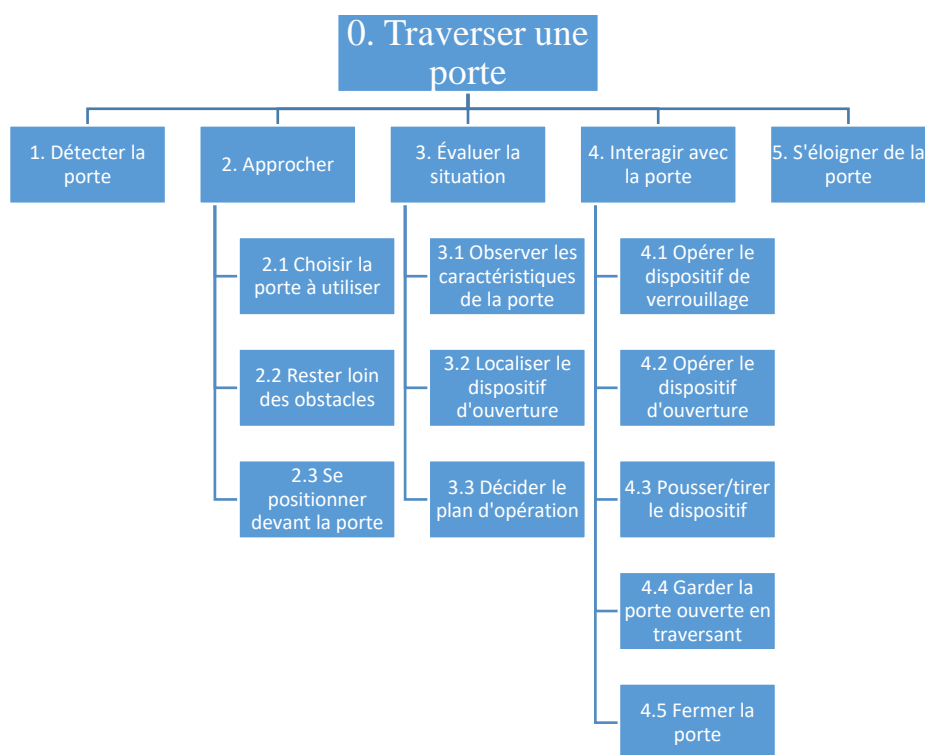


Figure 2.2 Analyse hiérarchique de tâches pour ouvrir puis traverser une porte

Ce travail vise à développer un système d'assistance, utilisant le bras Jaco², pour aider les utilisateurs en fauteuil roulant lorsque ceux-ci désirent ouvrir une porte. De ce fait, le fonctionnement normal du processus d'ouverture de portes a été étudié dans le but de détailler les

étapes importantes que le système complet, regroupant le fauteuil, le manipulateur et le système intelligent ajouté, devra accomplir.

Selon une étude de 2007 [10], la tâche de traverser une porte par un individu peut être déconstruite en cinq grandes étapes. Ces étapes sont visibles sur l'analyse hiérarchique de tâches (AHT) présentée à la Figure 2.2. En décortiquant l'AHT, il est possible de remarquer que ce sont les deux grandes étapes 3 et 4, en excluant 4.5, qui permettent d'accomplir la tâche spécifique d'ouvrir la porte. Le processus d'ouverture de porte est donc comblé par leurs sous-étapes.

2.3 Manipulateurs mobiles d'ouverture de porte

Dans la littérature, malgré qu'il existe très peu de MSAs sur fauteuil roulant adaptés pour l'ouverture de portes, plusieurs manipulateurs mobiles ont été développés pour accomplir cette tâche. Un manipulateur mobile est un système robotique construit d'un manipulateur sériel monté sur une plate-forme mobile robotisée comme un fauteuil roulant motorisé. Étant donné cette similitude avec le projet, ces travaux ont donc été étudiés et leur méthode de détection de la poignée ainsi que leur système de contrôle seront présentés dans cette section.

2.3.1 Détection de la poignée

Dans l'ensemble des travaux, la position cartésienne de la poignée est déterminée en réalisant deux étapes distinctes, soit : en repérant la poignée de porte, et en déterminant sa position cartésienne. Pour déterminer la position cartésienne, la littérature montre des méthodes relativement similaires. L'ensemble des travaux semble utiliser un dispositif leur permettant d'obtenir un nuage de points tridimensionnel de la scène, ou seulement de la poignée si la position de celle-ci a été obtenue avant.

2.3.1.1 Nuage de points par capteurs lasers

L'une des méthodes est l'utilisation de lasers. Ceci est le cas, entre autres, du Personal Robot 2 [11] et du manipulateur de déminage développé par Gaspers et coll. (2013) [12]. Pour générer le nuage de points, le Personal Robot 2 utilise un télémètre laser Hokuyo UTM-30 [13]. Ce dispositif fait constamment un balayage de la scène lors du déplacement du robot, lui permettant d'obtenir un nuage de points 3D complet de son environnement. Cependant, dû à la lumière du soleil qui affecte les capteurs laser, le Personal Robot 2 ne peut pas être utilisé à l'extérieur. Ceci peut

toutefois être réglé en positionnant le capteur et l'émetteur directement sur l'effecteur, comme utilisé pour le manipulateur de déminage. Effectivement, en les positionnant sur l'effecteur, ceux-ci peuvent être grandement rapprochés de la porte et ainsi ne pas être affectés par la lumière externe. Le manipulateur n'a plus qu'à balayer la porte de son capteur pour obtenir le nuage de points complet de celle-ci. [11], [12]

2.3.1.2 Nuage de points par capteurs infrarouges

Une autre solution est l'utilisation des capteurs infrarouges. Ceci est, entre autres, le cas pour Llopart et coll. (2017) [14] et Stuede et coll. (2019) [15] qui ont utilisé le capteur Kinect de Microsoft illustré à la Figure 2.3 [16]. Celui-ci est souvent utilisé en recherche et développement grâce à sa précision, sa robustesse, ainsi que son prix [17]. Toutefois, les capteurs infrarouges peuvent également être saturés par les rayons du soleil et donc ne peuvent pas fonctionner à l'extérieur.



Figure 2.3 Capteur Kinect de Microsoft



Figure 2.4 Caméra Realsense D435

2.3.1.3 Nuage de points par stéréovision

Finalement, la dernière méthode grandement utilisée est la stéréovision [18]–[21]. Celle-ci fonctionne en utilisant deux images provenant de deux capteurs d'images différents, physiquement décalés l'un de l'autre. Ensuite, le système associe les pixels équivalents des deux images. Lorsque les paires de pixels sont identifiées, il est par la suite possible de déterminer leur position cartésienne par rapport aux capteurs grâce au décalage des pixels retrouvés entre les deux images. Cette méthode est également très robuste et n'est pas affectée par la lumière extérieure. Elle peut être accomplie en utilisant diverses caméras distinctes, comme utilisé par un robot DLR mobile [18], ou en utilisant directement une caméra stéréovision comme le Bumblebee XB3 [22], utilisé par Klingbeil et coll. (2010) [19] et Andreopoulos et Tsotsos (2007) [20], et le Realsense D435 [23] utilisé par Ramôa et coll. (2020) [21] et illustré à la Figure 2.4.

2.3.1.4 Détection avec assistance

Pour repérer la poignée de porte, diverses méthodes ont été implémentées. L'une d'entre elles consiste à faire appel à un opérateur comme utilisé pour le manipulateur de déminage. Dans cette solution, ce dernier doit indiquer sur le nuage de point où se trouvent les deux extrémités horizontales de la poignée à levier. Avec cette information, le système peut ensuite déterminer la ligne horizontale qui représente la poignée et en obtenir la position par coordonnées cartésiennes à l'aide du nuage de points. Ce système simplifie donc le travail de l'utilisateur, mais ne rend pas la tâche automatique. [12]

2.3.1.5 Détection avec une image de référence

Pour une solution plus programmée, le DLR mobile utilise une image de référence déjà connue de la poignée. Après avoir pris des images de la porte à l'aide de trois caméras, l'image de référence est balayée sur ces images pour trouver la poignée correspondante. Connaissant ensuite la position de la poignée sur les trois images, la méthode de stéréovision est appliquée pour obtenir le nuage de points de celle-ci. Bien que cette solution ne nécessite pas l'aide d'un opérateur, celle-ci demande d'avoir au préalable une image de la poignée à détecter. Cette méthode ne fonctionne donc pas sur des poignées inconnues. [18]

2.3.1.6 Détection avec le nuage de points

Pour une solution automatique fonctionnant sur un plus grand ensemble de poignées, le Personal Robot 2 utilise directement l'information tridimensionnelle du nuage de points. Lorsque celui-ci se positionne devant la porte, le système extrait de ce nuage de points la section de la porte pouvant contenir la poignée, en éliminant par exemple le haut et le bas de la porte. Par la suite, celui-ci réalise une analyse du nuage de points pour détecter le plan de la porte. En détectant le plan, il est ensuite possible d'extraire la position de la poignée étant donné que celle-ci se retrouvera en avant du plan.[11]

Toutefois, le problème avec cette dernière méthode est que tous les points ne faisant pas partie du plan seront conservés, et non seulement les points de la poignée. Ceci peut donc entraîner énormément de faux positifs. Pour contrer ce problème, Llopart et coll. (2017) [14] utilisent deux méthodes pour détecter la poignée. La première, tout comme le Personal Robot 2, consiste à extraire le masque de la poignée en déterminant le plan de la porte à l'aide du nuage de points. La

deuxième extrait le masque en réalisant une segmentation visuelle par couleur. Cette deuxième méthode tient pour acquis que l'image possédera deux gammes de couleurs distinctes, soit celle de la porte et celle de la poignée. Toutefois, ceci peut également créer plusieurs faux positifs. La porte elle-même pourrait contenir plusieurs couleurs ou un autre objet de couleur différente. De ce fait, les nuages de points obtenus par les deux méthodes sont ensuite comparés et seuls les points se trouvant dans les deux analyses sont conservés. Ceci réduit donc les risques d'obtenir de faux positifs étant donné que les deux nuages de points ne sont pas obtenus de la même façon. [14]

2.3.1.7 Détection avec un dictionnaire de traits de Haar

Au lieu d'utiliser l'information du nuage de points ou de la couleur, il est aussi possible d'utiliser d'autres caractéristiques de la poignée pour la détecter. Par exemple, Klingbeil et coll. (2010) [19] et Andreopoulos et Tsotsos (2007) [20] utilisent un algorithme d'apprentissage supervisé qui préentraîne un dictionnaire de traits de Haar [24] de poignées de porte. Ce dictionnaire est développé en passant divers filtres sur des images de poignées de porte, lui permettant d'apprendre, à l'aide d'un vecteur de score, les traits de l'image. Par la suite, avec l'image de la porte, il est possible d'y balayer les filtres et d'y comparer les traits permettant au système de trouver la poignée.

2.3.1.8 Détection avec réseau neuronal convolutif

Finalement, il est aussi possible d'utiliser un réseau neuronal convolutif (RNC) permettant de déterminer de meilleures caractéristiques pour distinguer les poignées de porte. Ceci est entre autres la méthode utilisée par Stuede et coll. (2019) [15]. Ceux-ci utilisent plus précisément la méthode Faster R-CNN à l'aide d'un extracteur de traits Resnet 101 [25]. Plus d'explication concernant cette méthode seront présentées à la section 2.4. Cette méthode a été entraînée à l'aide d'une base de données de 200 images développées manuellement et obtient un *Mean Average Precision* (mAP) de 0.95 mAP@0.5IoU. Dans les cas où le système détecte plusieurs poignées de porte, ceux-ci conservent la détection dont le système est le plus certain. Cette stratégie conduit à un taux de réussite de 100% pour la détection sur l'ensemble de l'évaluation et lors de l'application pratique. Grâce à ce résultat, cette solution est donc la plus efficace sur l'ensemble des solutions présentées. Toutefois, comme bien des travaux présentés plus haut, celle-ci a uniquement été testée avec des poignées à levier.

Un RNC a également été utilisée par Arduengo et coll. (2021) [26] en utilisant la méthode YOLO, expliquée à la section 2.4. Cette méthode a également été développée à l'aide d'une base de données personnalisée. Celle-ci contenait 1213 images et permettait l'entraînement pour la détection de portes, d'armoires et de poignées. Après l'entraînement, le YOLO obtenait un score de 0.45 mAP@0.5IoU.

2.3.2 Système de contrôle du manipulateur sériel

Maintenant que la position cartésienne de la poignée est connue, la prochaine étape est d'ouvrir la porte. Ceci est réalisé en trois étapes, soit : d'atteindre la poignée, tourner la poignée, et ouvrir la porte. Dans la littérature, l'étape d'atteindre la poignée est très peu discutée, mais fait souvent recours à un système de contrôle de position cartésienne. Cependant, il est clair que l'ensemble des étapes est accompli dans la littérature en commandant le manipulateur sériel ainsi que la base mobile. Toutefois, cette section ne priorisera pas le contrôle de la base mobile.

Dans la littérature, les manipulateurs qui ouvrent des portes semblent se trouver dans deux catégories: ceux qui utilisent un contrôle d'impédance [7], [15], [18], [27], [28] et ceux qui n'en utilisent pas [12], [19], [20], [29]. Le contrôle par impédance permet de contrôler l'interaction de l'effecteur final avec l'environnement. Plus spécifiquement, ça permet au manipulateur de réagir aux forces extérieures qu'il ressent. Plus d'explication concernant ce type de contrôle seront présentées à la section 2.5.

2.3.2.1 Avec contrôle par impédance

Pour tourner une poignée à levier, les méthodes utilisant un contrôle d'impédance appliquent simplement un mouvement vers le bas de l'effecteur sur la poignée [15], [18]. Idéalement, avec cette méthode, l'effecteur va naturellement suivre la rotation de la poignée due aux forces ressenties et ainsi conserver un bon contact avec celle-ci. Pour y parvenir, les méthodes imposent une faible rigidité sur l'orientation de l'effecteur pour que celle-ci ne résiste pas aux rotations [15], [18]. Pour la tâche d'ouvrir la porte, une méthode similaire est appliquée sur l'ensemble des travaux. En général, ceux-ci contrôlent simultanément les mouvements du manipulateur par impédance et les mouvements de la base par une stratégie de contrôle autre [7], [15], [18], [27] à l'exception du manipulateur développé par Chung et coll. (2009) [28] qui n'utilise pas la base lors de l'ouverture de la porte. Ceci est réalisé selon l'une des trois structures suivantes :

- Le manipulateur conserve la distance entre la porte et la base pendant que celle-ci traverse la porte [15], [18];
- La base se rapproche de la porte pendant que le manipulateur l'ouvre en la poussant [27];
- Le système réalise les deux tâches simultanément pour ouvrir la porte à une vitesse constante [7].

Dans tous les cas, pour le contrôle du manipulateur, les gains de rigidité et d'amortissement sont déterminés pour permettre au bras de s'adapter aux mouvements de la porte [7], [15], [18], [27], [28]. Un retour de force est aussi utilisé dans certains cas à l'aide d'un capteur de force au poignet [7], [27] ou aux doigts de l'effecteur [28].

2.3.2.2 Sans contrôle par impédance

Pour les systèmes n'utilisant pas de contrôle par impédance, la tâche devient plus complexe. Comme mentionné dans le travail d'Andreopoulos et Tsotsos (2007) [20], si leur manipulateur était capable de fonctionner à l'aide d'un contrôle par force ou couple, l'utilisation d'un contrôle par impédance aurait simplifié la situation. En effet, étant donné que le système ne peut pas s'adapter aux forces exercées par la porte sur l'effecteur, celui-ci doit donc suivre une trajectoire précise pour l'ouvrir. Ceci nécessite donc plus d'informations sur la porte : connaître la position de l'axe de rotation de la poignée et de la porte. Ceci a été réalisé dans la littérature à l'aide d'un opérateur [12], d'un traqueur laser [29] et d'une caméra de stéréovision [19]. Connaissant ces informations, le système peut ensuite calculer et exécuter une trajectoire circulaire pour tourner la poignée et ouvrir la porte [12], [19], [20], [29].

2.4 Reconnaissance d'objets par ordinateur

La section 2.3 a reporté diverses méthodes pour repérer les poignées de portes dont plusieurs utilisent un apprentissage de traits des poignées pour ensuite les retrouver sur l'image. La reconnaissance d'objet par traits a pris de l'ampleur dès 2001 lorsque Viola et Jones (2001) ont proposé le premier algorithme capable de détecter des visages en temps réel à l'aide d'un dictionnaire de traits de Haar [30]. Toutefois, plusieurs méthodes de reconnaissance par trait ont pris le dessus depuis. Par exemple, les algorithmes de reconnaissance utilisant des RNCs font partie des méthodes les plus performantes de nos jours en termes de reconnaissance d'objet. Cependant,

il existe d'autres méthodes beaucoup plus simples qui, selon les circonstances, peuvent être aussi efficaces que les RNCs pour faire de la classification. Cette section présentera donc quelques algorithmes de reconnaissance utilisant des RNCs ainsi qu'une alternative à la classification utilisant un descripteur au lieu d'un RNC.

2.4.1 Reconnaissance par réseau neuronal convolutif

Un RNC est un algorithme qui apprend par lui-même à retirer les traits importants d'une classe d'objets grâce à des images. Ceci est utile pour faire de la classification d'objet. Il est possible grâce au RNC d'analyser l'image d'un nouvel objet et d'y déterminer sa classe selon ses traits. C'est donc un outil utilisé pour la classification. Toutefois, cet algorithme ne peut pas retrouver à lui seul un objet qu'il connaît dans une image. De ce fait, il est nécessaire d'avoir une méthode pour envoyer des régions de l'image au RNC afin de faire de la reconnaissance d'objet. Cependant, analyser toutes les régions possibles d'une image prend beaucoup trop de temps et choisir des régions de façon aléatoire peut nuire à la détection. Des méthodes plus efficaces ont donc été développées. Dans cette section, trois algorithmes de reconnaissance utilisant des RNCs seront présentés : Faster R-CNN [25], YOLO [31] et Mask R-CNN [32]. Ceux-ci ont été choisis selon leurs performances de détection et leurs performances de temps.

2.4.1.1 Faster R-CNN

Faster R-CNN fait partie des algorithmes de reconnaissance les plus performants en termes de détection. Il est une amélioration de l'algorithme de reconnaissance Fast R-CNN qui est une amélioration de R-CNN. Pour déterminer les régions à envoyer au RNC, R-CNN réalise une segmentation sémantique simple à l'aide de Selective Search [33]. Avec cette méthode, 2000 régions sont ensuite transmises au RNC, qui extrait les traits les plus importants de chaque région. Ces traits sont ensuite envoyés à un classificateur pour déterminer si la région représente un objet recherché ou non. Si plusieurs régions s'entrecroisent et sont classées pour le même objet, alors celui avec le plus grand score est conservé. Bien que cette méthode soit très performante pour la détection, elle est très lente à cause du nombre nécessaire de régions à traiter avec le RNC. Elle prend environ 50 secondes par images. Pour surmonter ce problème, Fast R-CNN n'utilise le RNC qu'une seule fois sur l'image initiale, créant ainsi une carte des traits. Ensuite, tout comme R-CNN, celui-ci utilise une méthode sélective pour déterminer les régions importantes de l'image. Leurs

régions équivalentes sont ensuite trouvées dans la carte de traits, analysées et classifiées. Cette solution réduit le temps de calcul par image à 2 secondes environ. Toutefois, Faster R-CNN pousse la solution encore plus loin. Plutôt que d'utiliser une méthode sélective sur l'image de base, Faster R-CNN utilise un deuxième réseau de neurones qui apprend à ressortir les régions d'intérêt directement de la carte de traits, réduisant ainsi le temps de calcul à 0.2 seconde environ. La Figure 2.5 illustre les étapes de reconnaissance de la méthode. [25], [34], [35]

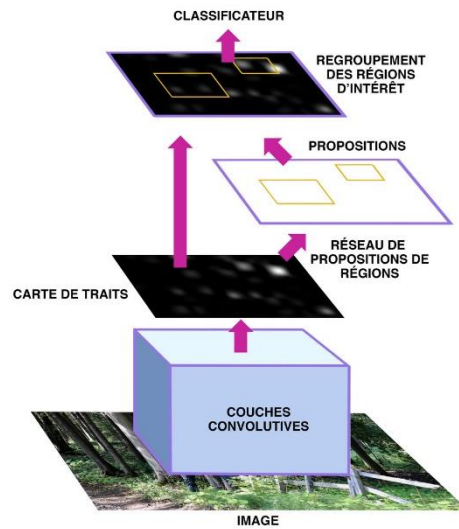


Figure 2.5 Reconnaissance d'objets par Faster R-CNN

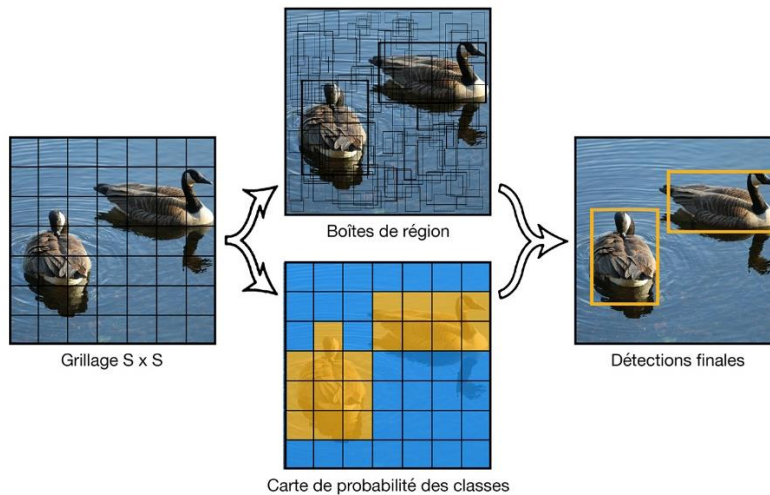


Figure 2.6 Reconnaissance d'objets par YOLO

2.4.1.2 *You Only Look Once (YOLO)*

Malheureusement, Faster R-CNN n'est pas assez rapide pour fonctionner en temps réel, contrairement à l'algorithme YOLO qui prend environ de 0.02 secondes par images [36]. Cette dernière méthode fonctionne en prenant l'image et en la divisant en un nombre préchoisi de cases de dimensions égales. Ensuite, pour chaque case, un nombre préchoisi de boîtes de régions sont générées aléatoirement. De plus, l'image au complet est analysée par le RNC pour obtenir les traits de chaque case et permettre ensuite de les classer. Finalement, les objets sont obtenus en associant les régions à la classe des cases et en éliminant les régions excédentaires à l'aide de la méthode d'*Intersection over Union* (IoU). La Figure 2.6 illustre les étapes de reconnaissance de la méthode. Cette stratégie permet à YOLO d'être environ 10 fois plus rapide que Faster R-CNN. Toutefois, il reste tout de même moins performant que Faster R-CNN en termes de détection [36]. [31]

2.4.1.3 Mask R-CNN

Mask R-CNN, quant à lui, est plus lent et performant que Faster R-CNN pour la détection. Toutefois, contrairement aux deux algorithmes présentés plus haut, celui-ci ne donne pas uniquement la boîte se trouvant autour des objets recherchés. En fait, comme le nom l'indique, celui-ci donne le masque de l'objet, comme montrer à la Figure 2.7. Pour y parvenir, il utilise pratiquement la même structure que Faster R-CNN. Toutefois, une segmentation sémantique par RNC est appliquée sur les régions finales des objets pour en ressortir le masque. [32]



Figure 2.7 Exemple de reconnaissance par Mask R-CNN

2.4.2 Classification par un descripteur

Comme mentionné précédemment, les RNCs sont des outils très efficaces pour classer une image ou une région d'une image. Ceci est dû au fait que le RNC apprend par lui-même les traits importants à regarder chez un objet. Toutefois, ceci nécessite bien entendu un entraînement du système et une grosse base de données. De plus, dans certaines circonstances, les traits permettant la classification peuvent être très évidents pour le développeur. Dans cette situation, il n'est pas nécessaire de demander au système d'apprendre ces caractéristiques par lui-même, mais simplement de lui en informer. Par exemple, il est possible d'utiliser simplement un descripteur. Un descripteur contient la description des caractéristiques visuelles retrouvée chez un objet.

L'un des descripteurs les plus connus et utilisés est l'histogramme de gradient orienté (HOG). Cette méthode fonctionne en observant, pour chaque pixel, la différence d'intensité, ou l'amplitude du gradient, entre les deux pixels adjacents sur l'axe x et sur l'axe y de l'image. Ensuite, pour chaque pixel, l'amplitude totale (M_g) du gradient et son orientation (θ_G) sont calculées avec l'amplitude du gradient en x (G_x) et en y (G_y) à l'aide des équations 1 et 2 ci-dessous.

$$M_g = \sqrt{G_x^2 + G_y^2} \quad (1)$$

$$\theta_G = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (2)$$

Finalement, les pixels sont regroupés en sous-groupes pour générer un histogramme de l'amplitude des gradients selon leurs angles. Ces histogrammes, conservés en vecteurs, forment ainsi les descripteurs de l'objet. Pour de la classification, le classificateur n'a plus qu'à être entraîné à reconnaître la combinaison de ces histogrammes. Cette méthode peut être très efficace pour classer, par exemple, des objets qui ont des formes très distinctes entre eux. [37]

2.5 Contrôle et cinématique du manipulateur

2.5.1 Cinématique inverse

En pratique, le contrôle du manipulateur peut se faire dans l'espace du manipulateur (l'espace des joints) ou dans l'espace de la tâche (l'espace cartésien). Par contre, pour passer d'un espace à l'autre, l'utilisation de la cinématique directe et inverse du manipulateur est nécessaire. La

cinématique directe permet de déterminer la position de l'effecteur en fonction des angles articulaires du manipulateur alors que la cinématique inverse permet de faire le contraire. Bien que pour un manipulateur sériel la cinématique directe puisse être obtenue analytiquement sans trop de difficulté, ceci n'est pas toujours le cas pour la cinématique inverse. En effet, lorsque le manipulateur possède plus de 5 joints, il devient rarement possible de calculer la cinématique inverse analytiquement, à l'exception des situations où celui-ci possède un poignet sphérique. Ceci est causé, entre autres, par la complexité des équations, qui sont formées de sinus, et par le nombre de combinaisons de joints possibles permettant d'atteindre une même position cartésienne désirée.

2.5.1.1 Cinématique inverse numérique

Pour contrer ce problème, diverses méthodes numériques ont été développées pour permettre de calculer rapidement la cinématique inverse d'un manipulateur. Les plus communes sont les approches numériques par optimisation. Ceux-ci fonctionnent en déplaçant l'effecteur, à l'aide de la cinématique directe, vers la position cartésienne désirée. L'objectif ici est donc de minimiser la distance entre la position actuelle et la position voulue. Pour y arriver, il est possible d'utiliser la Jacobienne du manipulateur. Celle-ci représente la relation entre la déviation angulaire des joints et la déviation cartésienne de l'effecteur. De ce fait, pour chaque petit déplacement angulaire, il est possible d'avoir avec la Jacobienne le petit déplacement équivalent cartésien. En utilisant l'inverse de la Jacobienne avec l'erreur cartésienne comme déviation, il serait donc, en théorie, possible d'obtenir la déviation angulaire nécessaire pour s'approcher de la position cartésienne désirée et ainsi obtenir une position angulaire permettant d'arriver au résultat voulu en itérant le tout. Toutefois, la Jacobienne n'est pas toujours inversible et, dans les situations où elle l'est, elle peut parfois être proche d'une singularité et déstabiliser l'algorithme. Une des approches pour contrer ceci est d'utiliser la pseudo-inverse de la Jacobienne et la méthode des moindres carrés. La pseudo-inverse est une méthode qui permet d'approximer l'inverse d'une matrice inversible ou non inversible tandis que la méthode des moindres carrés ajoute un amortissement (λ) à cette pseudo-inverse pour éviter les singularités. Cet amortissement est choisi d'avance par le développeur. Plus la valeur de l'amortissement est faible, plus le calcul sera rapide. Toutefois, si elle est trop faible, l'algorithme pourra devenir instable par singularité. L'équation 3 illustre ce qui est utilisé pour remplacer l'inverse de la Jacobienne (J) de la méthode initiale. [38]

$$J^{-1} \Rightarrow J^T(JJ^T + \lambda I)^{-1} \quad (3)$$

2.5.1.2 Cinématique inverse polynomiale

Toutefois, le problème avec les méthodes numériques est que celles-ci ne donnent qu'une seule configuration possible pour une position cartésienne recherchée. Une alternative à cela est de réaliser une cinématique inverse polynomiale. Étant donné que les équations de la cinématique inverse analytique peuvent s'avérer complexes, ces méthodes tentent de remplacer celles-ci par des équations polynomiales. L'une des façons est de représenter la position de l'effecteur par une combinaison de positions cartésiennes des joints au lieu de la représenter à l'aide des positions angulaires. Ceci peut facilement être fait en analysant, par exemple, la géométrie du manipulateur, ou en utilisant directement les matrices de Denavit Hartenberg. Toutefois, remplacer les équations angulaires par des polynômes ne permet pas d'obtenir la cinématique inverse analytiquement étant donné que ça crée des polynômes dont l'ordre est trop grand pour être résolu. Cependant, contrairement à la méthode utilisant les équations angulaires, il existe des méthodes numériques pour résoudre les polynômes et en ressortir toutes les solutions existantes. Ceci permet ainsi de connaître toutes les combinaisons de positions cartésiennes nécessaires de chaque articulation. Transformer ces positions cartésiennes en positions angulaires pour chaque joint redevient ensuite un problème de cinématique inverse analytique, mais beaucoup plus simple et réalisable. [39]

2.5.2 Contrôle du manipulateur

Comme montré à la section 2.3, l'utilisation d'un contrôle de position par impédance est très efficace pour l'ouverture de portes. Toutefois, les systèmes ne peuvent pas tous utiliser ce type de contrôle. Effectivement, les actionneurs du manipulateur choisis doivent subir un contrôle par couple en boucle fermée. Ceci est possible uniquement si les actionneurs en question possèdent des capteurs de couple. Si c'est le cas, une boucle de contrôle peut donc être réalisée à chaque actionneur pour que ceux-ci produisent le couple demandé. Cependant, le contrôle de position par impédance n'est pas la seule méthode de contrôle de position par couple utilisée dans la littérature. De plus, bien que le manipulateur fonctionne selon son espace angulaire, il est aussi possible de le contrôler dans l'espace cartésien. Cette section présentera donc diverses alternatives pour le contrôle de position.

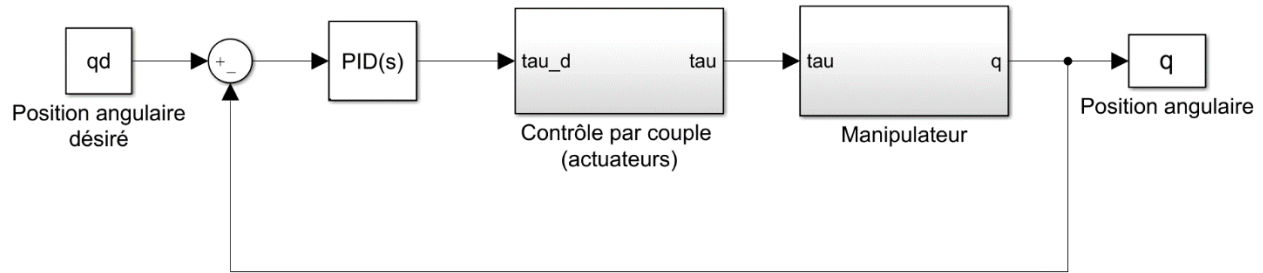


Figure 2.8 Exemple de contrôle de position angulaire avec correcteur PID

2.5.2.1 Contrôle de position angulaire avec correcteur PID

En pratique, les contrôleurs PID (proportionnel, intégrateur et dérivateur) sont fortement utilisés et ceci est aussi le cas pour le contrôle de manipulateurs sériels. Un exemple de contrôleur de position avec correcteur PID est illustré à la Figure 2.8.

Lorsque l'on désire contrôler le manipulateur en fonction de sa position angulaire, il faut comparer la position du manipulateur obtenue par les capteurs avec la position désirée. Cette erreur de position peut ensuite être utilisée pour déterminer le couple nécessaire à envoyer aux actuateurs pour la réduire. Toutefois, comme le montre la Figure 2.8, cette erreur doit avant tout être traitée par un correcteur pour obtenir un meilleur comportement du manipulateur. Le correcteur PID traite l'erreur de trois façons. Premièrement, il multiplie l'erreur par un gain pour modifier son amplitude. D'un côté, augmenter l'erreur peut accélérer la correction de la position ; d'un autre, une erreur trop grande peut déstabiliser le système ou bien saturer les moteurs. Deuxièmement, il dérive l'erreur initiale, lui permettant de voir la tendance de celle-ci et multiplie le résultat par un autre gain. Ceci lui permet de prédire le comportement de l'erreur et de la corriger d'avance. Finalement, il intègre l'erreur initiale et multiplie le résultat par un gain dans le but d'assurer que l'erreur à long terme deviendra nulle. Ces trois traitements sont ensuite additionnés ensemble et envoyés aux actuateurs comme consigne de couple. Le tout est ensuite reproduit jusqu'à la fin de la tâche. [40]

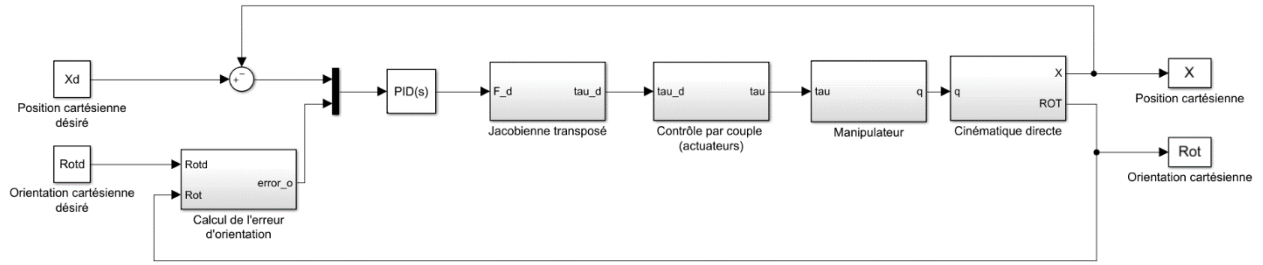


Figure 2.9 Exemple de contrôle de position cartésienne avec correcteur PID

2.5.2.2 Contrôle de position cartésienne avec correcteur PID

Bien que l'espace de fonctionnement du manipulateur soit angulaire, il est aussi possible de contrôler la position cartésienne de celui-ci. Ceci n'entraîne aucun changement par rapport au fonctionnement du PID, à l'exception que les gains devront être choisis en conséquence, mais certaines modifications doivent être apportées à la boucle de contrôle. Un exemple de contrôleur de position cartésienne avec correcteur PID est illustré à la Figure 2.9.

La première modification concerne l'orientation. Étant donné que celle-ci est sous forme matricielle, son erreur ne peut pas être calculée par une simple soustraction. Sous forme de matrice de rotation, celle-ci doit être calculée comme suit [41]:

$$R = Rot * Rot^{-1} * Rotd * Rot^{-1} \quad (4)$$

$$\begin{cases} \theta = \cos^{-1} \left(\frac{R_{11} + R_{22} + R_{33} - 1}{2} \right) \\ v = \frac{1}{2 \sin \theta} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix} \end{cases} \quad (5)$$

$$error_o = \theta v \quad (6)$$

De plus, lorsque les deux erreurs sont concaténées ensemble, le PID renvoie un signal de force et de couple (F) sous forme cartésienne. L'utilisation de la Jacobienne transposée (J^T) est donc utilisée pour retrouver les couples (τ) à l'aide de l'équation 7.

$$\tau = J^T F \quad (7)$$

Finalement, étant donné que les manipulateurs utilisent des capteurs de position angulaire, leurs valeurs doivent donc être changées en position cartésienne de l'effecteur à l'aide de la cinématique directe. [40]

2.5.2.3 Ajustement d'un correcteur PID

Pour un système non linéaire complexe, comme un manipulateur sériel, il n'est pas toujours possible d'ajuster les gains d'un PID pour que le système se comporte adéquatement sur l'ensemble des conditions d'utilisation. Pour cette raison, il est pratique d'utiliser un séquençement de gains. Ce principe consiste à séparer les conditions d'utilisation pour déterminer individuellement les meilleurs gains qui y correspondent. Ces gains peuvent ensuite s'alterner lorsque le système passe d'une condition à l'autre. Ceci implique donc le besoin d'ajuster plusieurs ensembles de gains PID. [42]

Tableau 2.1 Liste de méthodes permettant d'ajuster les gains d'un correcteur PID.

Ziegler-Nichols [43]	Lambda [43]	SIMC [43]
AMIGO [43]	Cohen-Coon [42]	IMC-PID [42]
Padula and Visioli [42]	Wang and Cluett [42]	Chien-Hrones-Reswick [44]

Heureusement, il existe plusieurs méthodes pour aider l'ajustement des gains d'un correcteur PID. Le Tableau 2.1 en liste quelques-unes. L'une des plus connues est la méthode Ziegler-Nichols [43]. Cette méthode consiste à amener le système à la limite de la stabilité en augmentant uniquement le gain proportionnel. La valeur de ce gain (k_c) ainsi que la période d'oscillation (T) sont ensuite utilisées pour définir les gains du correcteur (C) :

$$k_p = 0.6k_c \quad k_i = \frac{2k_p}{T} \quad k_d = k_p(0.125T)$$

Toutefois, les méthodes listées dans le Tableau 2.1 permettent d'obtenir une solution fonctionnelle, mais généralement non robuste au changement [43]. Pour cette raison, il est nécessaire d'ajuster les paramètres manuellement après l'utilisation de ces méthodes. Ceci peut être réalisé à l'aide de la méthode présentée dans *Control System Design Guide* quatrième édition [45]. Cette méthode consiste à traiter un PID comme deux ensembles de contrôleurs : le proportionnel-dérivateur (PD)

et l'intégrateur (I). Le contrôleur PD est avant tout ajusté pour améliorer la stabilité du système et l'intégrateur est au final modifié pour réduire l'erreur en régime permanent.

Cependant, ces méthodes ne sont pas développées pour les systèmes fonctionnant avec plusieurs entrées et plusieurs sorties. Pour les méthodes du Tableau 2.1, ce problème peut être contré, dans certains cas, en considérant la boucle de contrôle comme étant plusieurs boucles indépendantes entre elles possédant chacune une entrée et une sortie [46]. Cependant, cette simplification n'est pas possible pour les manipulateurs sériels étant donné que la dynamique de chaque actuateur est très affectée par celles des autres.

Pour ce qui concerne l'ajustement manuel, la méthode présentée dans *Control System Design Guide* quatrième édition sert de guide pour aider l'ajustement des gains et peut facilement être modifiée pour les systèmes fonctionnant avec plusieurs entrées et plusieurs sorties. Dans le cadre de ce projet, l'ajustement manuel des gains sera donc priorisé.

2.5.2.4 Contrôle de position avec information dynamique

Il est aussi possible d'utiliser la dynamique du système pour déterminer les couples nécessaires à envoyer aux actionneurs pour que le manipulateur se comporte comme désiré. En effet, en donnant une certaine commande de couples le système va se comporter en fonction de sa dynamique. Il serait donc théoriquement possible de calculer la dynamique inverse du système, pourvu que celle-ci soit connue, avec le comportement désiré, pour obtenir l'ensemble de couples permettant d'arriver à ce comportement. Toutefois, en pratique, il est impossible de reproduire parfaitement la dynamique d'un système complexe sous forme d'équations. Il est cependant possible de l'estimer et d'utiliser cette estimation pour assister la boucle de contrôle. Par exemple, il serait possible d'utiliser un correcteur PID, mais d'ajouter au couple sortant l'estimation du couple nécessaire pour compenser la gravité. De ce fait, la tâche du PID sera réduite, en quelque sorte, puisqu'il n'aura pas besoin de compenser la gravité à lui seul.

La dynamique d'un manipulateur peut être obtenue à l'aide de la méthode Euler-Lagrange ou Newton-Euler. Lorsque les équations obtenues par ces méthodes sont évaluées symboliquement, dans l'espace des joints, pour tout manipulateur, ils donnent une équation dynamique qui peut être écrite sous la forme :

$$\tau = M(q)\ddot{q} + C(q, \dot{q}) + G(q) \quad (8)$$

Où q , \dot{q} , \ddot{q} et τ sont respectivement le vecteur de positions, de vitesses, d'accélération et de couples angulaires des actuateurs et M , C et G représentent respectivement la matrice carrée d'inertie du manipulateur, le vecteur des termes de Coriolis et des forces centripètes et le vecteur de force gravitationnelle. Pour la dynamique calculée selon l'espace cartésien de l'effecteur, l'équation 8 devient :

$$F = M_x(q)\ddot{X} + C_x(q, \dot{q}) + G_x(q) \quad (9)$$

Où \ddot{X} et F sont respectivement l'accélération linéaire et angulaire de l'effecteur et les forces/couples de celui-ci. M_x , C_x et G_x quant à eux, représentent les mêmes informations qu'à l'équation 8, mais pour l'effecteur. [40]

2.5.2.5 Contrôle de position par impédance sans retour de force

Le contrôle par impédance, quant à lui, est habituellement développé dans l'espace cartésien étant donné que les forces ressenties par l'effecteur proviennent de l'espace de la tâche. Cette méthode reprend en quelque sorte le principe d'une force appliquée sur une masse retenue par un ressort et un amortisseur. La Figure 2.10 illustre ce principe.

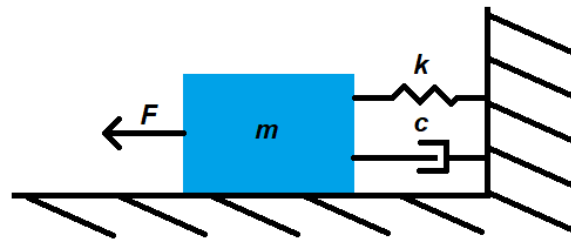


Figure 2.10 Force appliquée sur une masse retenue par un ressort et un amortisseur

Dans cette situation, la masse (m) va ressentir trois forces : soit la force externe (F) appliquée, la force du ressort résultante, dont la constante de rigidité est « k », et la force de l'amortisseur résultante, dont la constante est « c ». Dans cette situation, la masse va succomber à la force externe et va se déplacer en conséquence. Toutefois, grâce au ressort, plus elle va s'éloigner de sa position initiale, plus elle va travailler pour y revenir. De ce fait, lorsque la force est relâchée, le système pourra retourner à la position recherchée. Quant à l'amortisseur, celui-ci permet de contrôler, en quelque sorte, la vitesse de déplacement et de faire converger les oscillations causées par le ressort vers un point d'équilibre.

Pour un manipulateur, ce principe peut être intéressant. Habituellement, un système de contrôle est développé pour obtenir un comportement demandé, peu importe les perturbations externes. Cependant, pour un manipulateur, il peut être intéressant, dans certaines circonstances, d'utiliser les forces externes pour guider les déplacements de celui-ci. Ceci peut permettre de faciliter l'accomplissement de certaines tâches qui nécessitent de travailler avec l'environnement ou bien de rendre le manipulateur plus sécuritaire. Toutefois, aucun ressort et amortisseur n'est réellement intégré dans un manipulateur. Toutefois, il est possible d'imiter ce comportement pour chaque translation et orientation cartésienne à l'aide des actuateurs. La Figure 2.11 illustre la structure d'une boucle de contrôle permettant de reproduire ce comportement.

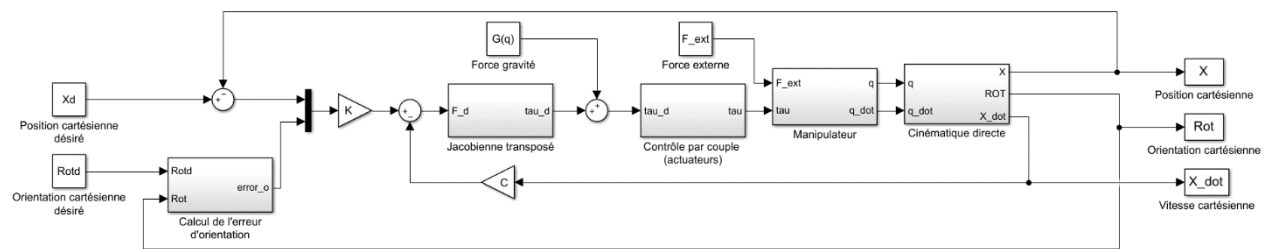


Figure 2.11 Exemple de contrôle de position par impédance sans retour de force

Comme on peut le remarquer en comparant la boucle avec celle de la Figure 2.9, le correcteur PID a été remplacé par un gain K . Ce gain est un vecteur de dimension 6×1 et représente numériquement les constantes de résistance des ressorts que l'on souhaite imiter. Tout comme un ressort, la force résultante sera produite par la différence entre la position linéaire ou rotationnelle désirée et obtenue, multipliée par la constante de résistance de celui-ci. Ensuite, pour imiter l'amortissement, la vitesse cartésienne sera multipliée par le vecteur de gain C , représentant numériquement les constantes d'amortissement, et la force résultante sera soustraite de la force de résistance. Finalement, le dernier changement est l'ajout de la force gravitationnelle. Cette compensation permet au manipulateur de réagir à toutes les forces externes, comme s'il avait un système de ressorts et d'amortisseurs, à l'exception de la force appliquée par la gravité. [40], [41]

2.5.2.6 Contrôle de position par impédance avec retour de force

Pour obtenir plus d'information concernant les forces externes appliquées sur l'effecteur, certains manipulateurs possèdent un capteur de forces et de couples à leur poignet. À l'aide de ce type de capteur, il est possible de tirer avantage de ces informations et de les appliquer dans la boucle de

contrôle lorsqu'un système d'impédance est utilisé. Par exemple, il est possible de retourner la force, ressentie par le capteur, aux actuateurs pour que ceux-ci produisent un déplacement dans la direction de la force. Ceci évite au manipulateur d'appliquer de la résistance lorsqu'il subit une force à l'effecteur. De plus, si ce retour de force est combiné à un système d'impédance, le manipulateur parviendra à revenir à sa position initiale lorsque la force externe devient nulle. [40], [41]

2.6 Objectif principal et objectifs spécifiques

La revue de littérature a démontré que les MSAs sont fort utiles pour les personnes souffrantes d'un handicap aux membres supérieurs. Toutefois, il reste encore du travail à faire pour rendre ces outils plus efficaces dans l'accomplissement de tâches complexes, dont l'ouverture de portes. De ce fait, l'objectif principal de ce mémoire est de **développer un système intelligent d'assistance à l'ouverture des portes pour les utilisateurs en fauteuil roulant en asservissant un manipulateur sériel d'assistance fixé sur le fauteuil**. Dans la littérature, les manipulateurs mobiles utilisés pour ouvrir des portes contrôlent à la fois le manipulateur sériel et la base sur laquelle il est fixé. Réaliser la même solution avec un fauteuil roulant non motorisé demanderait de modifier son fonctionnement, ce qui pourrait nécessiter plus de dépenses pour l'adapter ou pour l'achat d'un nouveau fauteuil. Pour cette raison, l'utilisateur gardera donc pleinement contrôle de son fauteuil et seul le manipulateur sera contrôlé.

Pour parvenir à cet objectif, les objectifs spécifiques suivants doivent être atteints :

1. Développer un système de stéréovision supporté par l'analyse intelligente de scènes tridimensionnelles par vision artificielle pour déterminer la nature et la position des poignées.
2. Développer un système de contrôle du manipulateur Jaco² permettant à celui-ci d'opérer de façon autonome les poignées
3. Valider les sous-systèmes en simulation à l'aide de conditions réelles.

CHAPITRE 3 MÉTHODOLOGIE

3.1 Cahier des charges

3.1.1 Choix du manipulateur sériel d'assistance

Cette recherche fait partie d'une série de travaux de recherche réalisés dans le laboratoire du Professeur Sofiane Achiche à l'École Polytechnique Montréal dont le but est de faciliter l'utilisation des manipulateurs robotique Jaco² et Mico de Kinova pour l'accomplissement de certaines tâches complexes [47]–[49]. De ce fait, le MSA choisi pour cette recherche est le manipulateur Jaco² de Kinova inc. Celui-ci a été priorisé au Mico puisque le deuxième et le troisième membre du Mico sont plus petits que celui du Jaco². Cette différence de portée avec le Mico aurait pu, dans certaines circonstances, rendre la tâche d'ouvrir une porte plus difficile, voire impossible à cause de l'interférence avec le fauteuil roulant. De plus, contrairement au Jaco², le Mico de Kinova est maintenant discontinué, le rendant ainsi moins utilisé par la communauté.

3.1.2 Étapes du processus d'ouverture d'une porte

Comme démontré au Chapitre 2, seules deux grandes étapes doivent être complétées pour permettre l'ouverture d'une porte, soit : l'étape 3 et 4 de la Figure 2.2 en excluant l'étape spécifique 4.5. Effectivement, les grandes étapes 1, 2 et 5 ne concernent pas directement la tâche d'ouvrir la porte et elles peuvent facilement être complétées par l'utilisateur.

Toutefois, dans le cadre de ce travail, on supposera que les portes sont déverrouillées et que la tâche d'ouvrir la porte se termine lorsque celle-ci est grande ouverte. De ce fait, l'étape 4.1 de la Figure 2.2 n'a pas été prise en compte et il ne sera pas nécessaire pour le système d'accomplir l'étape 4.4. Si la porte s'ouvre en s'éloignant de l'individu, il est fort probable que l'utilisateur traverse le cadre pour terminer la tâche d'ouverture ; si la porte s'ouvre dans le sens inverse, il est fort probable que la tâche d'ouverture se termine sans que l'individu traverse la porte. De plus, les portes possédant un système de fermeture automatique n'ont pas été prises en compte dans ce travail. De ce fait, dans cette situation, lorsque la tâche sera accomplie, l'utilisateur reprendra simplement possession des commandes du Jaco², le ramènera à sa position initiale, et traversera la porte.

Ceci étant dit, il est maintenant possible de déterminer la liste des étapes importantes que le système complet devra accomplir. Le Tableau 3.1 illustre les étapes principales provenant de l'AHT ainsi que les étapes spécifiques du système complet et de l'utilisateur.

Tableau 3.1 Liste des étapes importantes à accomplir

	Étapes spécifiques selon AHT	Étapes spécifiques du système
2.3	Se positionner devant la porte	Positionnement du fauteuil par l'utilisateur
3.1	Observer les caractéristiques de la porte	Détection et classification de la poignée de porte par la caméra et le système de vision par ordinateur
3.2	Localiser le dispositif d'ouverture	Détermination de la position de la poignée de porte par la caméra et le système de vision par ordinateur
3.3	Décider du plan d'opération	Planification de trajectoire du manipulateur
		Cinématique inverse
4.2	Opérer le dispositif d'ouverture	Déplacement du manipulateur à la poigné
		Agrippement de la poignée et rotation de celle-ci par le manipulateur si nécessaire
4.3	Pousser/tirer le dispositif	Déplacement du fauteuil par l'utilisateur
		Adaptation de la position du manipulateur selon les mouvements du fauteuil par contrôle d'impédance

3.1.3 Survol du manipulateur Jaco²

Maintenant que les étapes spécifiques pour le système complet ont été déterminées, la deuxième étape consiste à en ressortir plus précisément ce que le système intelligent devra ajouter au manipulateur Jaco². Effectivement, plusieurs systèmes ou commandes sont déjà en place dans le Jaco² et donc certaines étapes spécifiques peuvent déjà être accomplies par le manipulateur actuel. De ce fait, cette section fera un survol du manipulateur Jaco² et des étapes spécifiques qui ne seront pas prises en compte dans ce mémoire.

Dans le Jaco², 6 modes de contrôle sont déjà mis en place pour commander ses actuateurs [50] :

- Contrôle par position cartésienne;

- Contrôle par position angulaire;
- Contrôle par vitesse cartésienne;
- Contrôle par vitesse angulaire;
- Contrôle par force à l'effecteur;
- Contrôle par couple aux joints.



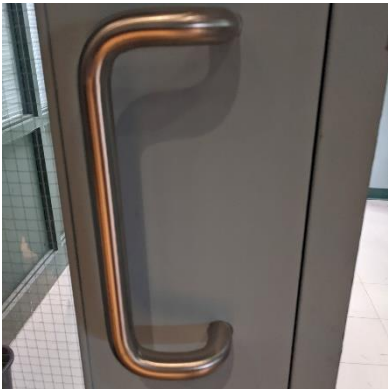
À première vue, on pourrait penser que le mode de contrôle par position cartésienne serait idéal pour le déplacement du manipulateur à la poignée. La trajectoire cartésienne pourrait simplement être envoyée au manipulateur sans avoir recours à la cinématique inverse. Toutefois, la littérature montre qu'il serait plus adéquat d'utiliser un contrôle par impédance pour l'ouverture de la porte. En d'autres mots, le manipulateur devra être en mode de contrôle par couple. Selon le guide d'utilisation du Jaco² [5], pour passer au mode de contrôle par couple, il est important que le manipulateur ne soit pas en contact avec un autre objet, à l'exception de sa base. De plus, il est préférable que celui-ci soit positionné à sa configuration initiale lors de ce changement. Pour ces deux raisons, le changement de mode devra se faire avant tout déplacement du manipulateur. Le déplacement du manipulateur à la poignée doit donc être réalisé à l'aide d'un système de contrôle de position par couple obligeant ainsi l'utilisation de la cinématique inverse.

Toutefois, il y a un problème avec le mode de contrôle par couple du manipulateur Jaco de deuxième génération. Habituellement, pour les manipulateurs sériels provenant de Kinova, l'action de refermer l'effecteur et de l'ouvrir peut se faire avec une simple commande, peu importe le mode de contrôle utilisé [51]. Cependant, pour leurs manipulateurs sériels de deuxième génération, cette commande n'est pas accessible en mode de contrôle par couple [5]. Lorsqu'ils sont dans ce mode, il n'est pas possible de commander l'effecteur. Néanmoins, étant donné que le mode de contrôle par couple est inévitable et que cette faille a été réglée par la suite par l'entreprise, on considérera que le contrôle de l'effecteur sera encore disponible. De ce fait, ce travail considère que l'étape d'agripper la poignée peut déjà être réglée par le système actuel et donc elle ne sera pas mise de l'avant par cette recherche.

3.1.4 Survol des dispositifs d'ouverture

Pour ouvrir la porte, il est crucial que le système soit en mesure de bien détecter et d'utiliser le dispositif d'ouverture. Toutefois, il en existe de toutes sortes. Heureusement, certains sont plus communs que d'autres [52]. Le Tableau 3.2 présente donc les trois classes de dispositifs les plus communes que le système devra être en mesure d'actionner. De plus, dans le cadre de ce projet, le manipulateur Jaco² sera considéré comme étant capable de maintenir une prise ferme sur ces trois types de poignées. Cette décision a été prise étant donné que l'effecteur du Jaco² n'a pas été développé spécifiquement pour cette tâche et donc nécessitera une modification éventuelle pour être en mesure de maintenir sa poigne sur ces trois types de poignées.

Tableau 3.2 Les classes de dispositifs d'ouverture les plus communes

Poignée ronde	Poignée à levier	Poignée à tirer (verticale sans loquet)
Généralement ronde ou ovale dont l'axe de rotation à tourner pour libérer le loquet est au milieu.	Tige horizontale dont l'axe de rotation à tourner pour libérer le loquet est positionné à une des extrémités.	Tige verticale fixée sur la porte donnant un support à la main pour tirer la porte et l'ouvrir.
		
Figure 3.1 Poignée ronde	Figure 3.2 Poignée à levier	Figure 3.3 Poignée à tirer

3.1.5 Fonctions du système

Il est maintenant possible de mieux définir les fonctions recherchées du système à développer. Le Tableau 3.3 présente ces fonctions ainsi que les sous-fonctions que le système devra accomplir. Il est important de noter que les fonctions listées sont accompagnées des choix technologiques déjà décidés. Alors que dans un exercice de design cela ne serait pas le cas, ici dans le cadre d'un projet

de maîtrise nous voulons contraindre l'espace de conception pour le développement d'une solution fonctionnelle.

Tableau 3.3 Fonctions et sous-fonctions recherchées du système à développer

Détecter la position de poignées de porte et de les classer parmi les trois classes.
1. Obtenir une carte de profondeur de la scène.
2. Déterminer la position de la poignée de porte par rapport au manipulateur.
3. Reconnaître la classe de la poignée de porte.
Planifier la trajectoire du manipulateur Jaco².
1. Planifier la trajectoire du manipulateur selon la position et la classe de la poignée.
2. Transformer la trajectoire en coordonnées angulaires par cinématique inverse.
Opérer le dispositif d'ouverture à l'aide du manipulateur Jaco².
1. Déplacer le manipulateur jusqu'à la poignée à l'aide d'un contrôle de position par couple.
2. Déplacer le manipulateur pour libérer le loquet à l'aide d'un contrôle par impédance.
Ouvrir la porte.
1. Déplacer le manipulateur à l'aide d'un contrôle par impédance pour qu'il s'adapte au déplacement du fauteuil et de la porte.

3.2 Détection et classification de la poignée de porte

3.2.1 Survol de la solution utilisée

Pour détecter la poignée de porte, plusieurs méthodes différentes ont été utilisées dans la littérature. Toutefois, les solutions adoptées possédaient toutes une structure similaire. Notamment, ceux-ci utilisaient un ou plusieurs capteurs pour générer un nuage de points de la scène ainsi qu'une méthode pour repérer la poignée et la positionner dans l'espace cartésienne. Cependant, dans certains cas les méthodes s'entrecroisaient. De ce fait, une décision sur le processus général de la détection doit être prise avant tout. Cette section présentera donc la structure générale utilisée dans ce travail pour détecter et classer la poignée de porte.

3.2.1.1 Méthode de repérage de la poignée

Sur toutes les méthodes utilisées par les manipulateurs mobiles pour détecter la poignée de porte, celle qui semble être la plus efficace est une méthode utilisant RNC, soit : Faster R-CNN. Effectivement, elle était capable de reconnaître des poignées jamais observées avant et, lors de l'évaluation de celle-ci, aucune erreur n'a été réalisée sur la base de données de test [15]. De ce fait, pour le repérage de la poignée, une méthode RNC a été utilisée pour cette recherche.

En ce qui concerne l'entraînement, malgré l'existence de plusieurs grandes bases de données gratuites disponibles pour la détection d'objets, dont COCO [53], ImageNet [54] et Open Images [55], aucune d'entre elles n'a été développée pour permettre la détection de poignées de porte. De plus, sur l'ensemble des petites bases de données publiques, seule celle développée par Arduengo et coll. (2021) [26], pour l'entraînement de la méthode YOLO, semble permettre la détection de poignées de porte. De ce fait, malgré le score de $0.45\text{mAP}@0.5\text{IoU}$ obtenu dans l'article, leur base de données a été choisie pour l'entraînement. De plus, cette base de données permet également de détecter divers types de portes. De ce fait, pour permettre une meilleure comparaison avec le modèle développé avec YOLO, l'entraînement a été fait avec l'ensemble de ces classes même si seulement les poignées sont utilisées par la suite.

En ce qui concerne le choix de l'algorithme de reconnaissance, étant donné que la base de données fonctionne avec des boîtes de régions, Mask R-CNN ne pourra pas être utilisé. De ce fait, a été choisi. Malgré qu'il soit moins rapide que YOLO, il est actuellement le plus performant des algorithmes de reconnaissance et est le plus rapide parmi les R-CNN. De plus, seule une image sera analysée par tâche d'ouverture de porte, donc il n'est pas nécessaire de détecter la poignée en temps réel. De ce fait, tant que l'algorithme détecte la poignée dans un temps convenable, comme Faster R-CNN, il est préférable de prioriser la performance de détection plutôt que la rapidité.

Toutefois, l'algorithme de reconnaissance Faster R-CNN n'est pas la seule méthode utilisée pour détecter la poignée. Inspirée de l'article de Llopart et coll. (2017) [14], la détection de la poignée de porte va se réaliser en fusionnant deux méthodes, soit : une détection à l'aide de l'algorithme de reconnaissance Faster R-CNN, ainsi qu'une détection à l'aide de l'information tridimensionnelle provenant du nuage de points de la scène. Cette solution a été choisie pour tenter de filtrer les faux positifs détectés par la méthode Faster R-CNN et ainsi augmenter les performances du système.

3.2.1.2 Classification de la poignée de porte

Étant donné que le système doit être en mesure d'ouvrir trois types de poignées de porte différents, il est important que celui-ci parvienne à les classer. Cependant, il n'existe aucune base de données d'images possédant diverses classes des poignées. De ce fait, réaliser la classification à l'aide de Faster R-CNN nécessiterait la création d'une grande base de données personnelle pour son entraînement. Toutefois, étant donné que la poignée sera détectée, il est possible de simplement utiliser cette détection et de la classer à l'aide d'un autre outil. Par exemple, la classification pourrait se faire à l'aide d'un RNC supplémentaire ou même avec un descripteur HOG. En effet, étant donné que les trois types de poignées possèdent des formes très distinctes entre elles, leur descripteur HOG devrait se différencier facilement. De plus, l'utilisation d'un descripteur HOG est très simple à implémenter et prend moins de place en mémoire qu'un RNC. Ceci nécessite tout de même une base de données pour permettre l'entraînement du classificateur, mais elle sera beaucoup plus petite et rapide à produire. Sa simplicité rend également le modèle plus adaptable. Par exemple, si un utilisateur possède chez lui des poignées d'allure complexe qui rendent la tâche de classification difficile, alors il pourrait être possible d'entraîner le classificateur à nouveau en quelques secondes en ajoutant une image de la poignée en question. À l'inverse, entraîner un classificateur utilisant un RNC peut prendre beaucoup de temps.

3.2.1.3 Capteur visuel

Pour positionner la poignée dans l'espace cartésien, un capteur permettant de générer un nuage de points doit être choisi. De plus, celui-ci doit également permettre l'obtention d'une image couleur de la scène pour permettre la reconnaissance par Faster R-CNN. Trois capteurs capables de fournir ces résultats sont disponibles au laboratoire de Sofiane Achiche et ont donc été considérés.

- La Kinect de Microsoft;
- La caméra Realsense D435 d'Intel;
- La caméra Realsense D415 d'Intel.

Après considération, la caméra de stéréovision D415 d'Intel fut sélectionnée pour plusieurs raisons. Entre les deux caméras d'Intel, la D415 est considérée deux fois plus précise que la D435 [56]. Pour ce qui est de la Kinect, elle possède une moins grande portée de vue que la D415. Effectivement, la distance minimum pour la Kinect est de 0.5 mètre contrairement à 0.16 mètre

pour la D415 et la distance maximum est de 4.5 mètres contre 10 mètres, respectivement [47]. La Kinect pourrait également s'avérer plus encombrante pour l'utilisateur dû à son volume qui est environ 24 fois plus grand que la D415 [47]. Enfin, celle-ci ne peut pas être utilisée à l'extérieur étant donné qu'elle fonctionne à l'aide d'un capteur infrarouge [47].

3.2.2 Position de la caméra

La caméra doit être positionnée sur le fauteuil pour avoir une vue sur la poignée de porte. Toutefois, lorsque l'utilisateur se positionnera devant la porte, la poignée pourrait se retrouver autant vers sa gauche que vers sa droite. De ce fait, pour respecter cette symétrie, il a été décidé de positionner la caméra sur le plan central du fauteuil, montré à la Figure 3.4. De plus, dans le but d'avoir une vue horizontale complète de la porte, la caméra doit être distancée de celle-ci. L'installer derrière l'utilisateur comme le montre la Figure 3.5 est donc la meilleure solution, surtout lorsque l'on considère que la distance d'utilisation idéale pour la caméra Intel D415 est au-dessus de 45cm [57]. Par ailleurs, la positionner devant l'utilisateur serait encombrant pour celui-ci.

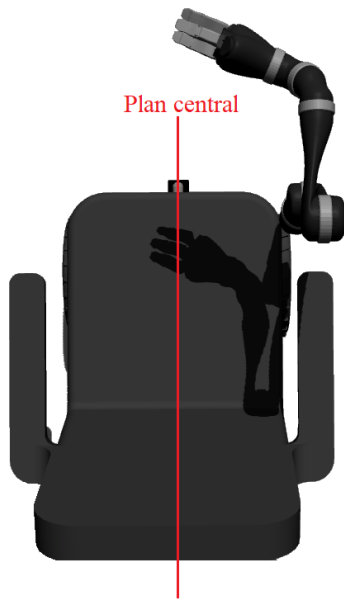


Figure 3.4 Fauteuil vu de dessus



Figure 3.5 Configuration complète

En ce qui concerne la position exacte de la caméra, celle-ci doit être le plus bas possible pour ne pas être trop encombrante pour l'utilisateur tout en couvrant le plus possible la scène pouvant contenir la poignée de porte. De ce fait, la hauteur en fonction du siège (h_c) ainsi que l'angle de la

caméra (θ) ont été calculés à l'aide des dimensions présentées sur la Figure 3.6. Ces dimensions sont des valeurs extrêmes qui ont été choisies pour respecter les objectifs de position sans que la caméra voie la tête de l'utilisateur. Les mesures concernant l'utilisateur ont donc été effectuées avec quelqu'un possédant une grande taille [58] et un grand tour de tête [59] selon la normale, soit : 1,88 m et 64 cm respectivement. Pour ce qui est des mesures du fauteuil, ceux-ci proviennent d'un fauteuil roulant Métro d'Everest & Jannings, qui était accessible. La distance horizontale entre la poignée et la caméra, de 90 cm, a été déterminée en mesurant la distance horizontale entre les porte-pieds et le dossier d'un fauteuil. La hauteur de la poignée choisie est la hauteur standard d'une poignée de porte [60]. L'angle de 20° est dû au fait que le champ de vision vertical de la caméra est de 40° [57]. Finalement, la distance de 5 cm a été ajoutée comme jeu. Après calcul, ceci donne une hauteur (h_c) de 116 cm et un angle (θ) de 16° .

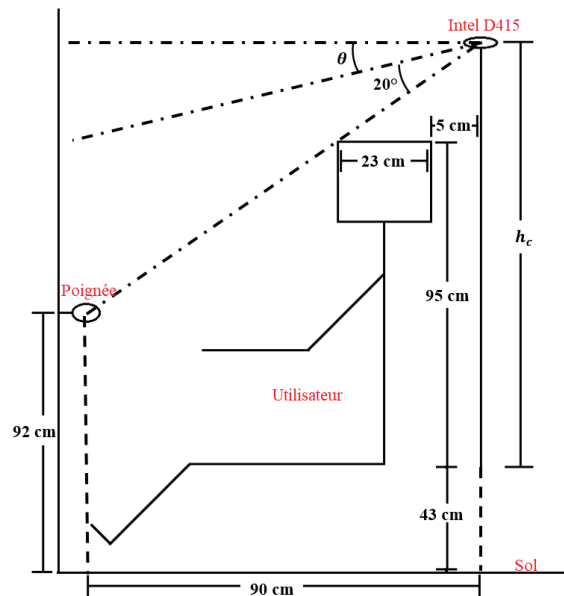


Figure 3.6 Schéma pour calculer l'angle et la hauteur de la caméra

3.2.3 Reconnaissance de la poignée

3.2.3.1 Détection par Faster R-CNN

Le développement de l'algorithme Faster R-CNN a été réalisé à l'aide d'une bibliothèque de Python à code ouvert nommé PyTorch Lightning [61]. Cette bibliothèque fournit une interface de haut niveau pour PyTorch et facilite le développement de projets impliquant l'apprentissage profond [61]. Pour ce qui est de l'entraînement, même si l'algorithme devra apprendre à identifier

de nouveaux objets, il est une bonne pratique d'utiliser un modèle dont les poids ont déjà été préentraînés à l'aide d'une autre base de données. Ceci permet de réduire le temps d'apprentissage. De ce fait, les modèles Faster R-CNN préentraînés disponibles dans la bibliothèque Torchvision de Pytorch ont été comparés. Le Tableau 3.4 montre les caractéristiques de ces modèles [62]. Ces trois modèles ont tous été préentraînés sur l'ensemble d'instances de COCO train2017 [62]. Comme on peut le constater, le modèle Faster R-CNN ResNet-50 FPN possède une précision moyenne plus élevée que les deux autres. De plus, bien qu'il soit le plus lent, son temps de test par objet est très faible et ne devrait pas être trop lent pour l'utilisateur. C'est donc celui-ci qui a été choisi. En ce qui concerne l'entraînement, diverses tailles de lot ainsi que divers taux d'apprentissage ont été testés. Finalement, il a été décidé d'utiliser une taille de lot de 8 ainsi qu'un taux d'apprentissage de 0.005.

Tableau 3.4 Modèles Faster R-CNN dans Torchvision

Modèles	Précision moyenne (%)	Temps d'entraînement (s/objet)	Temps de test (s/objet)
Faster R-CNN ResNet-50 FPN	37	0.229	0.059
Faster R-CNN MobileNetV3-Large FPN	33	0.102	0.042
Faster R-CNN MobileNetV3-Large 320 FPN	23	0.098	0.038

3.2.3.2 Correction de la détection par l'information provenant du nuage de points

La caméra Realsense D415 permet d'obtenir une carte de profondeur qui peut par la suite être transformée en nuage de points cartésiens à l'aide des informations sur les paramètres intrinsèques de la caméra. De ce nuage de points, il est possible de repérer le plan de la porte à l'aide de RANSAC [63]. RANSAC est un outil fortement utilisé en vision artificielle qui permet de traiter un nuage de points et d'en estimer le plan le plus approprié pour le plus grand ensemble de points. À l'exception des points cartésiens aux extrémités de l'image qui peuvent faussement représenter l'environnement, l'ensemble des points du nuage sont fournis à l'outil RANSAC pour déterminer le plan de la porte. Étant donné que l'utilisateur sera positionné juste en face de la porte, l'image

sera majoritairement formée par celle-ci. De ce fait, il n'est pas nécessaire d'utiliser spécifiquement les points de la porte.

Après avoir déterminé le plan de la porte, chaque boîte de région obtenue à l'aide de Faster R-CNN est évaluée en commençant par la boîte ayant le plus grand score de confiance. Cette évaluation regarde, pour chaque pixel dans la boîte, si le point cartésien équivalent se trouve au moins à un centimètre devant le plan de la porte. Si moins de deux pixels sont devant le plan, alors la boîte est éliminée et la suivante est analysée. Dans le cas contraire, cette boîte est conservée et toutes les autres régions ayant un plus petit score de confiance sont éliminées. Les points de la boîte devant le plan sont ensuite associés à la carte de profondeur, mais sous forme de masque permettant ainsi d'avoir le masque de la poignée. Cette limite de deux pixels a été choisie à cause du bruit de la caméra. Dans de rares cas, un seul pixel pouvait être considéré comme étant au moins à un centimètre devant le plan. Cette limite pourrait être légèrement augmentée, mais cette augmentation ne s'est pas avérée nécessaire.

3.2.4 Classification de la poignée

Lorsque la poignée est reconnue, il est ensuite nécessaire de la classer en envoyant la partie de l'image contenant cette poignée de porte au classificateur. Ceci est réalisé en utilisant le masque de la poignée obtenue sur la carte de profondeur. Ce masque est avant tout analysé en largeur et en hauteur pour déterminer le centre de celui-ci et pour déterminer si celui-ci est plus long à la verticale ou à l'horizontale. Ensuite, il est possible de définir une région en forme de carré autour de ce point central, dont les dimensions des côtés sont équivalentes à la longueur la plus grande du masque. Toutefois, étant donné que le masque ne contient pas toujours la totalité de la poignée, il a été décidé d'allonger cette région de 3 pixels dans chaque direction. Finalement, la région équivalente de l'image couleur est extraite et envoyée au classificateur.

La classification est ensuite réalisée à l'aide d'une machine à vecteurs de support, utilisant un noyau de fonction de base radial, entraîné pour classer les vecteurs de caractéristique HOG. De ce fait, pour l'entraînement, une base de données contenant 35 images de chaque classe a été créée à l'aide d'images provenant de Google. De cette base de données, 30 images de chaque classe ont été utilisées pour l'entraînement ainsi que 5 pour tester l'efficacité. De plus, dans le but de doubler la base de données pour l'entraînement, les images ont également été tournées dans le sens horizontal. Finalement, les données d'entraînement ont été converties en vecteur de caractéristique par HOG.

3.2.5 Position de la poignée

La position de l'effecteur est définie par un seul point. De ce fait, la position de la poignée sera définie par le point cartésien qui devra être atteint par l'effecteur. Toutefois, ce point est différent selon la classe de la poignée. Pour les poignées rondes, ce point doit être sur l'axe de rotation de la poignée. Pour les poignées à levier et à tirer, le point recherché sera au centre du manche. Ce point peut être déterminé à l'aide du centre du masque déterminé pour la classification de la poignée. Toutefois, pour chaque type de poignées, une correction de la position peut être nécessaire.

Par exemple, à cause de la position de la caméra par rapport à la poignée, le point central du masque ne se trouve pas exactement sur l'axe de rotation des poignées rondes. Effectivement, en réalisant des tests, il a été remarqué que le pixel central était environ toujours un pixel plus haut que l'axe de rotation sur la carte de profondeur. Ceci semble causer simplement un décalage cartésien d'un demi-centimètre, à peu près. Toutefois, étant donné que ce décalage était relativement constant, il a été décidé de descendre le point central du masque pour les poignées rondes d'un pixel à chaque détection. Ceci se produit également dans l'autre direction lorsque la poignée ronde se trouve plus aux extrémités de l'image. De ce fait, le même principe a été utilisé pour la position horizontale de la poignée en séparant l'image en trois régions verticales de largeurs égales. Si la poignée se trouve dans la région centrale, aucun changement n'est réalisé ; si la poignée se trouve dans la région de gauche, le point central du masque est décalé d'un pixel vers la gauche ; et si la poignée est dans la région de droite, alors il est décalé d'un pixel vers la droite.

Pour les poignées à levier et à tirer, ceci n'est pas vraiment un problème puisqu'atteindre exactement le centre de la poignée n'est pas aussi important. Toutefois, le point central de leur masque n'est pas toujours positionné sur eux, comme le montre la Figure 3.7 b). De ce fait, pour les poignées à tirer, une droite horizontale passant par le point central est évaluée pour définir le pixel le plus à droite et à gauche sur cette droite, comme le montre la Figure 3.7 c). Le nouveau centre sera ensuite déterminé par la moyenne de ces deux positions, tel qu'illustré sur la Figure 3.7 d). Toutefois, si le masque n'est pas complet à cause d'une erreur et que ceci entraîne une absence de pixel sur la droite, alors le centre de la poignée ne sera pas mis à jour étant donné que dans la plupart des cas, le centre n'aura pas besoin d'être déplacé. Le même principe est utilisé pour les poignées à levier, mais avec une droite verticale passant par le point central, au lieu d'une droite

horizontale, et en définissant le pixel le plus bas et le plus haut sur cette droite, au lieu du pixel le plus à droite et à gauche.

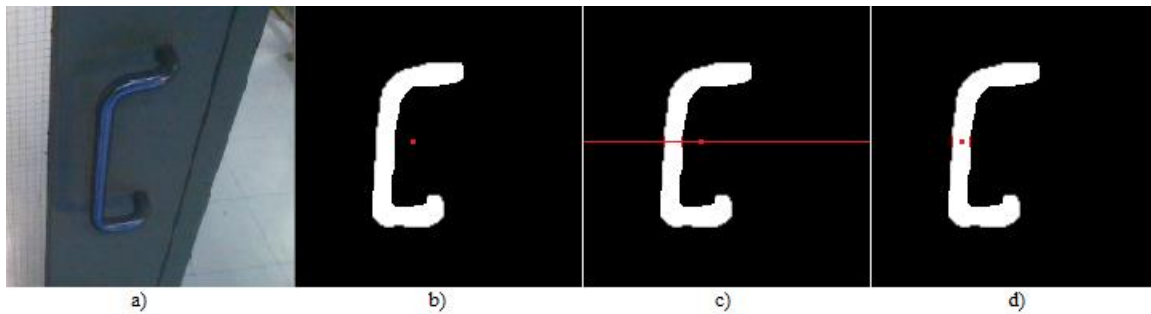


Figure 3.7 Solution utilisée pour corriger la position d'une poignée à tirer sur le masque de la carte de profondeur. a) Illustre la photo utilisée pour la détection. b) Illustre le masque de la poignée avec le point central en rouge de celui-ci. c) Est la même illustration qu'en b), mais avec la droite horizontale passant par le point central et avec la position des pixels le plus à gauche et à droite sur cette droite. d) Illustre le nouveau centre de la poignée

Finalement, après avoir déterminé la position de la poignée sur la carte de profondeur, la position cartésienne de celle-ci par rapport à la caméra peut être déterminée à l'aide des informations sur les paramètres intrinsèques de la caméra. Cette position est ensuite utilisée pour le Jaco² en réalisant un changement de repère.

3.3 Simulation et modélisation

Bien que la détection et la classification des poignées de porte doivent être réalisées dans un environnement réel, étant donné que ceci est produit avec une caméra physique, le reste du système a été développé et expérimenté en simulation. Toutefois, le système doit être en mesure de fonctionner de façon séquentielle dans laquelle chaque étape utilise l'information de l'étape précédente. De ce fait, même si une partie du système a été développé avec des expérimentations réelles et l'autre avec des expérimentations simulées, le système de planification et de contrôle ainsi que la simulation ont été développés pour être en mesure de prendre l'information fournie par le système de vision. En effet, le système de vision enregistre l'information concernant la position et la classe de la poignée sur un fichier texte, et le système de planification et de contrôle ainsi que la simulation utilisent ce fichier texte pour continuer et simuler la tâche. Cependant, avant de considérer les autres sous-systèmes, il faut calculer la dynamique du manipulateur Jaco² et

déterminer les paramètres importants des autres systèmes impliqués dans la problématique, comme la porte, la poignée, et le fauteuil. Le tout pourra par la suite être simulé pour le développement des systèmes de contrôle et pour les expérimentations.

3.3.1 Simulation

Il a été décidé de développer les systèmes de contrôle dans l'environnement Matlab/Simulink. De ce fait, il était important de trouver un outil compatible avec Simulink permettant la modélisation de systèmes dynamique. Simscape en est un et est déjà implémenté dans Simulink. Pour cette raison, l'ensemble des systèmes dynamiques présents dans ce projet ont été modélisés à l'aide de Simscape. Leur structure est disponible à l'Annexe A. Pour le visuel du manipulateur, les fichiers STL ont été repris du dépôt Github créé par Kinova pour implémenter le bras dans ROS/Gazebo [64]. Pour ce qui concerne le visuel du fauteuil, les fichiers STL ont été obtenus à l'aide des informations retrouvées dans le Github nommé *Nuric wheelchair model 2* [65].

Toutefois, étant donné que chaque sous-tâche pour ouvrir la porte possède quelques conditions de modélisation différente entre elles, discutées dans la section 3.3.2, alors il n'est pas possible d'utiliser une seule modélisation Simscape pour l'ensemble de la simulation. Pour cette raison, une modélisation Simscape a été développée pour chaque système de contrôle. De ce fait, les systèmes de contrôle ont dû être développés dans Simulink séparément et sont reliés ensemble à l'aide d'un programme Matlab. Pour simuler le système de contrôle au complet dans l'environnement, Matlab roule chaque Simulink et Simscape un après l'autre en utilisant comme valeur initiale les valeurs finales du Simulink et Simscape précédent.

3.3.2 Modélisation

3.3.2.1 Détail sur la modélisation du manipulateur Jaco²

La dynamique du manipulateur Jaco² a été déterminée à l'aide des dimensions et paramètres présentés dans la thèse de Mendes (2017) [66]. Les informations du Jaco² sont présentées dans l'Annexe B. À l'aide de ces informations, il était possible de simuler la dynamique du Jaco² à l'aide de Simscape ainsi que de déterminer les matrices de l'équation 8 en utilisant Denavit-Hartenberg (DH).

En pratique, les actionneurs d'un manipulateur vont toutefois subir de la friction, ce qui n'est pas pris en compte dans l'équation ni dans les paramètres. Cependant, il est important de mentionner que par défaut, lorsque le manipulateur Jaco² est opéré en contrôle par couple, la gravité ainsi que la friction des actionneurs sont compensées automatiquement [50]. À cause de cette compensation automatique, la friction n'est pas considérée dans ce travail. Par contre, ceci est uniquement vrai si la méthode de communication utilisée pour l'interface de programmation d'application est celle de haut niveau. Ceci entraîne quelques complications. Tel que présenté dans la thèse de Mendes (2017) [66], la méthode de communication peut être de haut niveau ou de bas niveau. Celle de bas niveau permet de communiquer directement avec les actionneurs avec un débit de communication à 500 Hz. De son côté, la communication de haut niveau passe par un processeur de signal numérique et sa fréquence de débit est de 100 Hz. Ceci entraîne beaucoup de délais au système et va complexifier le développement des systèmes de contrôle. Dans la simulation, le signal du couple envoyé au manipulateur devrait ainsi subir un délai de 0.01 seconde. Toutefois, la simulation a ici été faite avec un délai de 0.011 pour assurer une certaine robustesse des systèmes de contrôle face à la fréquence de débit de 100Hz.

De plus, étant donné qu'un système de contrôle des actionneurs par couple est déjà mis en place, la simulation fait également passer le signal du couple envoyé au manipulateur par un filtre passe-bas dans le but de représenter la dynamique de ce sous-système. Toutefois, dans la littérature, il n'existe aucune information concernant le temps de réponse du système de contrôle des actionneurs. Cependant, il est normal de supposer que la dynamique du système de contrôle des actionneurs soit aussi rapide ou plus rapide que le débit de communication de l'interface de programmation d'application. De ce fait, pour représenter le pire cas possible, la dynamique de ce sous-système a été simulée pour être aussi rapide que l'interface, qui a été développée pour fonctionner avec un débit de communication à 500 Hz.

Finalement, les limites de rotation ont également été implémentées à l'aide des informations du guide d'utilisation des manipulateurs de deuxième génération de Kinova [5] ainsi que les limites de couple à l'aide du guide d'utilisation des actionneurs de Kinova [67].

3.3.2.2 Détail sur la modélisation du fauteuil roulant

Toute information et paramètres concernant le fauteuil ont été tirés du Github *Nuric wheelchair model 2* [65]. L'idée ici est de reproduire les déplacements que le manipulateur va subir à cause du

fauteuil, donc la dynamique de celui-ci n'a pas besoin d'être exacte. En ce qui concerne les déplacements, un système de contrôle de vitesse par dynamique inverse avec une PID a été implémenté rapidement pour que celui-ci réagisse convenablement aux commandes qui lui sont envoyées. Les équations de la dynamique inverse ont été basées sur la dynamique inverse d'un robot mobile présentée dans l'article de Ivanjko et coll. (2010) [68]. Dans le but de représenter le plus possible la situation qui aura lieu en pratique, ces commandes de vitesses sont envoyées au code à l'aide d'une manette externe contrôlée par un humain. Finalement, il est important de mentionner que dans le cadre de cette recherche, lorsque le fauteuil est placé devant la porte, celui-ci est considéré parfaitement droit par rapport à celle-ci, que l'erreur est négligeable. De ce fait, au début de chaque simulation, l'utilisateur fait parfaitement face à la porte.

3.3.2.3 Détail sur la modélisation de la porte et la poignée

Pour la modélisation de la porte, le frottement des charnières est considéré comme négligeable et n'est donc pas pris en compte dans la simulation. De ce fait, seules les dimensions de la porte et la densité, utilisé pour calculer l'inertie, sont nécessaires. Les dimensions ont été prises d'une porte de domicile régulière, soit : une hauteur de 2.07m, une largeur de 0.91m et une épaisseur de 0.025m. En ce qui concerne la densité, une valeur moyenne de la densité du bois a été utilisée, soit : 500 kg/m^3 [69]. Finalement, un bloqueur est implémenté dans la simulation pour imposer la direction dans laquelle la porte doit s'ouvrir.

Pour la modélisation de la poignée, la hauteur de celle-ci est déterminée automatiquement par l'information produite par la caméra, et la distance de celle-ci par rapport aux charnières a été mesurée à 0.84m. En ce qui concerne sa dynamique, la masse et l'inertie sont considérées comme négligeables et ne sont donc pas prises en compte dans la simulation. Finalement, pour les poignées tournantes, une résistance de ressort de 5.04 Nm/rad est utilisée. Cette valeur correspond à la résistance de ressort d'une poignée mesurée dans l'article de Gao et coll. (2019) [29].

En ce qui concerne l'interaction entre la poignée et l'effecteur, lorsque le manipulateur agrippe la poignée, la simulation considérera celui-ci comme étant parfaitement fixe sur la poignée. De ce fait, le joint de contact entre les deux sera un joint fixe.

3.4 Systèmes de contrôle

3.4.1 Atteindre le dispositif d'ouverture

Pour atteindre la poignée, il a été décidé de contrôler le Jaco² dans l'espace des actuateurs. Ceci permet d'avoir un certain contrôle sur l'ensemble du manipulateur, dont la position des coudes, et non uniquement sur l'effecteur. Un système de position par couple est donc utilisé. Plus précisément, la boucle de contrôle mise en place est munie d'un PID, d'une compensation gravitationnelle, et d'une action anticipative pour compenser les forces causées par les inerties internes. La Figure 3.8 illustre la structure du système de contrôle.

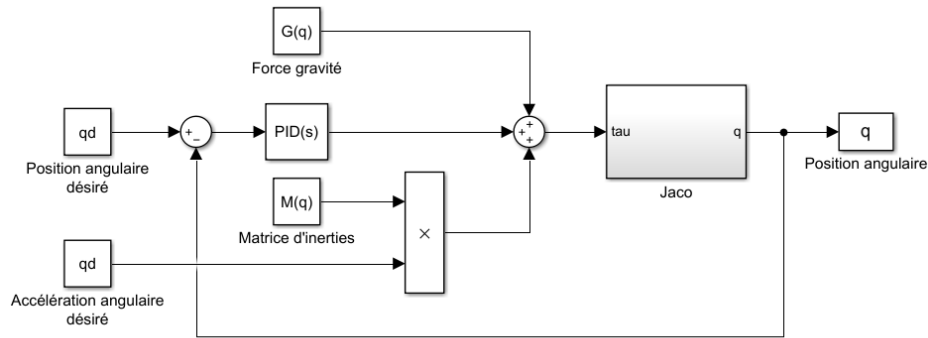


Figure 3.8 Structure du système de contrôle pour atteindre la poignée

3.4.1.1 Compensation gravitationnelle et action anticipative

La compensation gravitationnelle et l'action anticipative sont toutes les deux créées à l'aide des informations concernant la dynamique du manipulateur. Elles sont respectivement obtenues à l'aide de la matrice gravitationnelle et de la matrice d'inertie du manipulateur Jaco². Elles sont également calculées à l'aide de la position angulaire des actuateurs. Toutefois, pour l'action anticipative, la matrice d'inertie est ensuite multipliée avec l'accélération angulaire prévue par la trajectoire, au lieu d'être multipliée avec l'accélération angulaire du manipulateur. Par expérience, ceci entraînait de meilleurs résultats. En pratique, étant donné que le manipulateur Jaco² compense automatiquement la force gravitationnelle lorsqu'il est opéré en contrôle par couple, cette compensation gravitationnelle n'aura pas besoin d'être ajoutée dans le système final.

3.4.1.2 Correcteur PID

Pour le correcteur PID, plusieurs groupes de gains ont été créés. Effectivement, à cause de la complexité du manipulateur et de la fréquence de débit du processeur de signal numérique, il n'est pas possible d'utiliser un seul ensemble de gains pour toutes les positions possibles du manipulateur lorsque la méthode de communication est haut niveau. De ce fait, 6 groupes de gains ont été créés, soit : 2 par classe de poignées, un si la poignée est plus vers la gauche, et un autre si elle est plus vers la droite. Il est important de mentionner que les gains proportionnels, intégrateurs et dérivateurs sont chacun représentés par une matrice diagonale de dimension 6x6, à cause des 6 actionneurs.

Pour le premier groupe de gains, les gains ont été déterminés à l'aide d'une méthode présentée dans *Control System Design Guide* quatrième édition [45].

L'idée derrière cette méthode est de traiter un PID comme deux ensembles de contrôleurs : le proportionnel-dérivateur (PD) et l'intégrateur (I). Toutefois, quelques adaptations ont été apportées étant donné que 6 actionneurs doivent être contrôlés en même temps et qu'un actionneur instable peut entraîner une instabilité sur l'ensemble du système. Les gains proportionnels et dérivateurs ont donc été déterminés comme suit :

1. Définition d'une trajectoire qui implique des positions moyennes que peut atteindre l'effecteur pour la tâche d'ouverture d'une porte, pour un groupe de gains spécifique. Plus d'information sur la trajectoire est présentée à la section 3.5.
2. Initiation de tous les gains à zéro pour ne pas affecter le comportement du bras Jaco².
3. Augmentation lente du gain proportionnel de l'actionneur situé le plus près de l'effecteur en commençant par 0.001 et en s'arrêtant lorsque la trajectoire réalisée par un actionneur ne peut plus s'approcher de la trajectoire désirée.
4. Retour sur l'étape 3 avec l'actionneur suivant jusqu'à ce que tous les gains proportionnels des actionneurs soient trouvés.
5. Retour sur l'étape 3 et 4 en débutant cette fois avec les gains proportionnels déjà trouvés et jusqu'à ce qu'il ne soit plus possible d'améliorer la trajectoire.
6. Augmentation des gains dérivateurs pour chaque actionneur de la même façon que pour les gains proportionnels, comme aux étapes 3, 4 et 5.

7. Retour sur les étapes 5 et 6 jusqu'à ce qu'il ne soit plus possible d'améliorer la trajectoire.
8. Expérimentation avec plusieurs trajectoires et modification des gains en conséquence.

Ensuite, les gains intégrateurs ont été déterminés comme suit :

1. Augmentation des paramètres de dynamique de l'actuateur de 10% pour créer une différence entre celui-ci et la dynamique calculée pour la matrice gravitationnelle et la matrice d'inertie.
2. Augmentation des gains intégrateurs, en suivant les étapes 3 et 4 pour les gains proportionnels, avec la trajectoire moyenne définie initialement, mais en essayant cette fois de réduire le plus possible l'erreur totale entre la trajectoire réalisée et la trajectoire désirée.
3. Expérimentation avec plusieurs trajectoires et divers paramètres, en augmentant ou diminuant les paramètres de façon aléatoire jusqu'à 10%, et modification des gains en conséquence.

Pour les autres groupes de gains, les mêmes étapes ont été réalisées, mais les gains obtenus pour le premier groupe ont été utilisés comme gain de départ pour sauver du temps. Il est aussi nécessaire de préciser que dans ce cas, certains gains ont dû être réduits dès le début.

3.4.2 Ouverture de la porte

Le manipulateur Jaco² ne possède pas un capteur de force à son poignet, de ce fait, le contrôle d'impédance utilisé pour l'ouverture de la porte ne peut se baser sur un retour de force. La structure de celui-ci est donc très similaire à la Figure 2.11 présentée dans la revue de littérature. Toutefois, un gain d'amortissement selon la vitesse angulaire des actuateurs est aussi ajouté. Ceci permet de rendre le système plus sécuritaire dans cette situation. Effectivement, bien que dans le cadre de ce projet la prise de l'effecteur sur la poignée soit considérée comme parfaite, le manipulateur doit rester sécuritaire pour l'utilisateur advenant que l'effecteur lâche la poignée lors d'une situation réelle. Or, avoir un amortissement uniquement sur la position de l'effecteur ne réduit pas nécessairement les oscillations que peuvent subir respectivement les actuateurs. Ces oscillations ne sont pas prévisibles et peuvent entrer en contact avec l'utilisateur. De ce fait, ajouter un amortissement directement aux actuateurs permet d'éviter ce problème. La Figure 3.9 illustre la structure du système de contrôle.

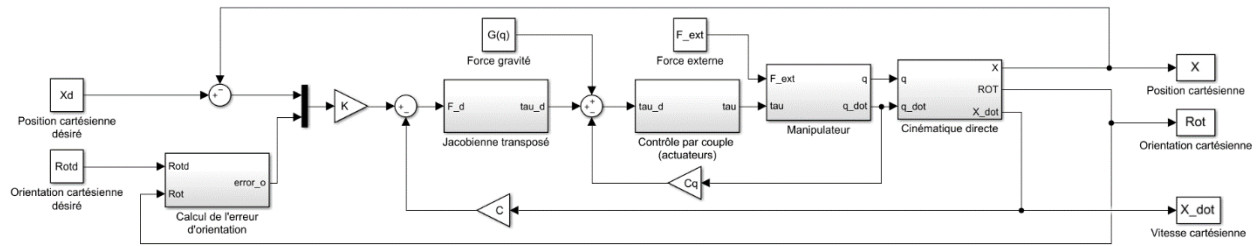


Figure 3.9 Structure du système de contrôle pour ouvrir la porte

3.4.2.1 Sélection des gains du contrôleur

Comme illustré à la Figure 3.9, trois groupes de gains, chacun représenté à l'aide d'une matrice diagonale de dimension 6×6 , ont été déterminés : les gains de résistance sur l'effecteur (K), les gains d'amortissement sur l'effecteur (C) et les gains d'amortissement sur les actuateurs (C_q). Toutefois, plusieurs combinaisons de gains sont possibles pour la tâche. Cependant, lors de la décision des gains, certains critères devaient être respectés.

La stabilité du système, en tout temps, en est un exemple. De plus, étant donné que les déplacements du fauteuil sont inconnus, il est important que le manipulateur se laisse guider le plus possible par ces déplacements. Effectivement, si celui-ci résiste trop aux déplacements, il y a une forte chance que l'effecteur ne parvienne pas à garder sa prise sur la poignée en situation réelle. De ce fait, les directions et rotations cartésiennes de l'effecteur affectées par les mouvements du fauteuil et de la porte doivent posséder moins de résistance. À l'inverse, il est préférable que les directions et rotations qui sont naturellement non-affectées par ces mouvements possèdent une plus grande résistance pour que les perturbations les affectent moins. Par exemple, selon ce critère, la direction perpendiculaire au sol peut posséder un gain de résistance de 100 N/m tandis que les deux autres peuvent être nuls. Pour l'orientation, le gain de résistance pour la rotation par rapport à l'axe perpendiculaire au sol peut être à zéro contrairement aux deux autres. Toutefois, il est aussi important que le manipulateur parvienne à déplacer la porte. De ce fait, aucun de ces gains de résistance touchant à la position ne doit être nul. Les directions parallèles au sol doivent posséder une résistance assez élevée pour déplacer la porte, mais assez faible pour se laisser guider par le fauteuil.

En ce qui concerne les gains d'amortissement pour l'effecteur, ceux-ci doivent rester le plus près de zéro que possibles pour ne pas donner trop de résistance aux déplacements rapides du fauteuil.

Toutefois, ils doivent également être en mesure d'éliminer toute oscillation que pourrait subir l'effecteur. Ceci doit être vrai lorsque l'effecteur est fixe sur la poignée et lorsque celui-ci ne l'est pas pour garder le système sécuritaire si jamais l'effecteur perd sa prise.

Finalement, lorsque les autres gains sont identifiés, il est possible de déterminer les gains d'amortissement pour les actionneurs avec ce dernier critère : lorsque l'effecteur n'est pas en contact avec la poignée, les gains d'amortissement doivent être augmentés tout en étant le plus bas possible pour réduire autant que possible les oscillations subies par les coudes du manipulateur.

3.4.3 Opérer le dispositif d'ouverture

L'opération de la poignée de porte doit uniquement se faire pour la poignée ronde et la poignée à levier. Étant donné que le système sera en contact avec l'environnement, il a été décidé d'utiliser un contrôle par impédance pour accomplir cette tâche. Pour opérer la poignée à levier, la structure du système de contrôle est exactement la même que pour l'ouverture de la porte, présenté à la Figure 3.9). Toutefois, pour opérer la poignée ronde, un intégrateur sur la rotation de l'axe perpendiculaire à la porte a été ajouté, comme illustré à la Figure 3.10. Ceci a été fait pour aider la rotation de l'actionneur au poignet du Jaco², étant donné que c'est majoritairement lui qui doit opérer la poignée ronde. Puisque l'effecteur possède une inertie très faible, augmenter le gain de résistance pour influencer l'actionneur au poignet peut facilement entraîner une instabilité au système. Or, avec la structure initiale, il est impossible de tourner la poignée sans augmenter ce gain. Toutefois, un intégrateur permet de contourner ce problème. L'intégrateur va permettre au gain de s'adapter selon l'erreur pour que celui-ci devienne nul. Dans ce cas-ci, il va spécifiquement toucher l'actionneur du poignet et lui demander de forcer en conséquence pour corriger l'erreur. Toutefois, un limiteur doit être donné à l'intégrateur pour éviter que la valeur résultante augmente trop et cause une instabilité.

3.4.3.1 Méthode pour sélectionner les gains

Pour les gains permettant d'opérer la poignée à levier, ceux-ci ont été déterminés comme suit :

1. Définition d'une position moyenne initiale que peut avoir le manipulateur lorsqu'il maintient la poignée à levier.
2. Initialisation des gains aux valeurs obtenues pour ouvrir la porte.

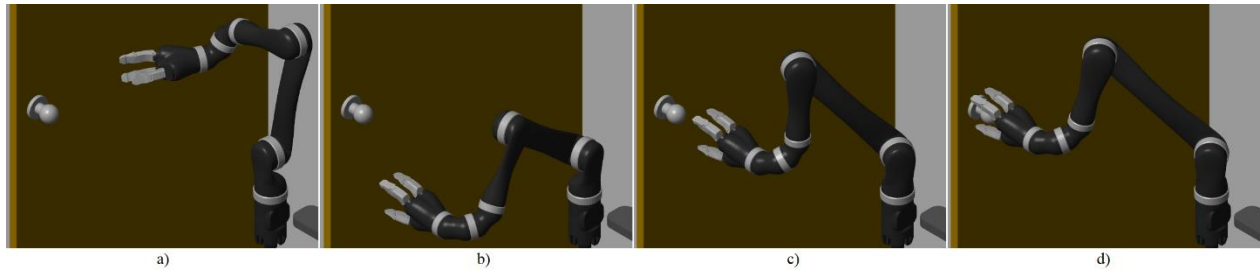


Figure 3.11 Décortication de la trajectoire. a) Illustre la position maison du manipulateur, b) la position intermédiaire, c) la position devant la poignée, et d) la position sur la poignée

3.5.1.1 Orientation de l'effecteur

Les orientations qui ont été choisies pour chaque classe sont illustrées à la Figure 3.12. L'orientation pour les poignées à levier et à tirer a été choisie pour assurer une bonne prise de l'effecteur et pour assurer que celle-ci soit réalisable selon différentes positions cartésiennes. Pour l'orientation concernant les poignées rondes, elle fut choisie pour que le manipulateur soit en mesure de tourner la poignée à l'aide de son 6^e actuateur. En supposant que le fauteuil roulant est parfaitement droit devant la porte, ces orientations peuvent facilement être trouvées et représentées par une matrice de rotation. Il est important de mentionner que l'orientation pour la poignée à tirer a été choisie puisque le Jaco² utilisé dans le cadre de ce projet sera situé à la droite du fauteuil. Si le manipulateur avait été à la gauche, l'effecteur aurait fait dos à nous sur la Figure 3.12 c).

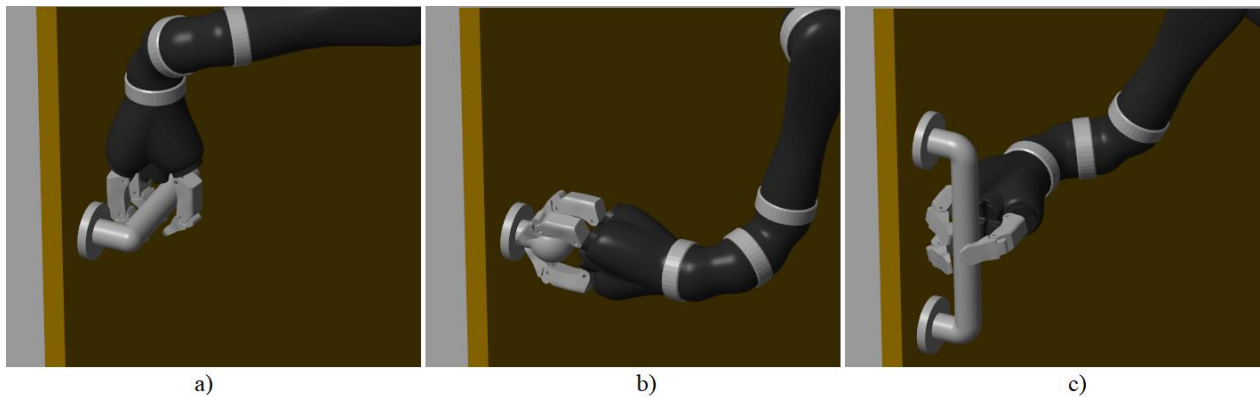


Figure 3.12 Orientations de l'effecteur pour chaque classe de poignée. a), b) et c) illustrent respectivement l'orientation de l'effecteur pour les poignées à levier, ronde, et à tirer

3.5.1.2 Cinématique inverse

Afin de passer de l'espace cartésien à l'espace angulaire du bras articulaire, l'utilisation de la cinématique inverse est nécessaire. La méthode des moindres carrés avec la pseudo-inverse de la Jacobienne a été choisie pour sa simplicité de calcul. En ce qui concerne l'amortissement, celui-ci est fixé à 0.01. Par essai-erreur, il a été déterminé qu'avoir un amortissement plus faible pouvait, dans certains cas, entraîner des instabilités.

3.5.1.3 Position intermédiaire

Une position intermédiaire du manipulateur a également été ajoutée à la trajectoire pour mieux prédire ces déplacements et ainsi faciliter le développement du contrôleur. Cette position est également utilisée comme valeur initiale lors du calcul de la cinématique inverse numérique pour éviter les minimums locaux et ainsi converger vers la solution désirée. De ce fait, 6 positions intermédiaires ont été précalculées à l'aide d'une cinématique inverse polynomiale, soit : 2 par classe de poignées : une si la poignée est plus vers la gauche, et une autre si elle est plus vers la droite. Cette cinématique inverse révèle, pour une configuration cartésienne désirée, toutes les configurations des actionneurs possibles permettant ainsi de choisir celle qui convient le mieux à la tâche. Par exemple, il est préférable d'avoir :

- Un coude vers le haut pour ne pas entrer en contact avec l'utilisateur;
- Une configuration angulaire similaires à la configuration de départ pour réduire les déplacements;
- Un coude vers l'intérieur de la porte pour éviter tout contact avec le cadre de porte lors de son ouverture.

En ce qui concerne les 6 configurations cartésiennes désirées, leur orientation correspond à l'orientation finale selon la classe de la poignée, et la position a été déterminée pour :

- Que l'effecteur soit le plus près possible du fauteuil, mais tout de même devant celui-ci;
- Que l'effecteur soit environ à la hauteur moyenne des poignées tout en respectant le premier point;
- Que l'effecteur soit aligné pour atteindre une poignée directement en face de sa base si celle-ci est à droite ou directement en face de la roue gauche si celle-ci est à gauche.

Les positions intermédiaires du manipulateur utilisées sont présentées à l'Annexe C.

3.5.1.4 Vitesse et accélération le long de la trajectoire

Chaque déplacement d'une position à une autre se fait à des accélérations et vitesses différentes. De plus, la trajectoire fait un arrêt complet aux 4 positions. Pour chaque déplacement entre deux positions, l'accélération subie par chaque joint suit une trajectoire similaire à ce qui est montré sur la Figure 3.13. De ce fait, la vitesse augmente de façon exponentielle pendant une seconde, décélère de la même façon pendant une autre seconde pour atteindre sa vitesse maximale, comme le montre la Figure 3.14, et réalise l'inverse pour s'arrêter à la position désirée. En ce qui concerne la vitesse maximale, elle dépend du joint qui devra réaliser le plus grand déplacement. Le temps de déplacement pour l'ensemble des joints est calculé pour que ce joint se déplace à une vitesse maximale de 16 degrés par seconde. La vitesse maximale des autres joints est ensuite calculée selon ce temps de déplacement et la distance qu'ils doivent réaliser respectivement.

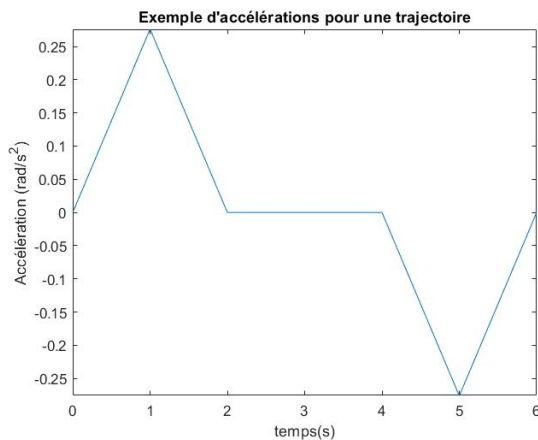


Figure 3.13 Exemple d'accéléérations

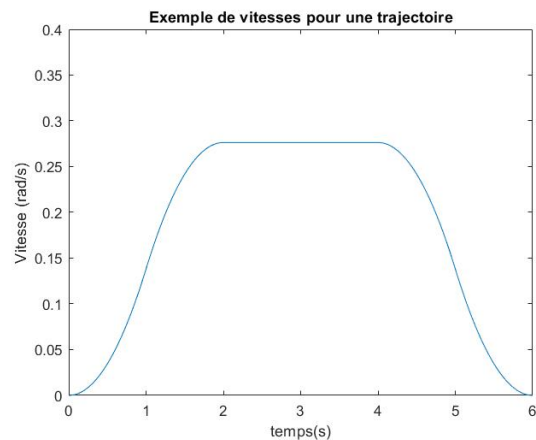


Figure 3.14 Exemple de vitesses

3.5.2 Opérer le dispositif d'ouverture

La trajectoire pour opérer la poignée est réalisée que pour les poignées rondes et les poignées à levier. Elle est réalisée dans l'espace cartésien étant donné que le contrôle se fait dans cet espace. De plus, l'opération se fait en trois étapes : tourner la poignée, ouvrir légèrement la porte, et ramener la poignée à sa position initiale. Le temps alloué pour chacune de ces étapes est respectivement 4, 3, et 4 secondes. Toutefois, à cause de la différence d'orientation du manipulateur, les poignées rondes et les poignées à levier ne s'ouvrent pas de la même façon.

3.5.2.1 Opérer une poignée ronde

Pour opérer une poignée ronde, la trajectoire implique simplement de faire une rotation de 90 degrés sur l'axe des « y » de la base du manipulateur Jaco² et de -90 degrés pour la ramener à sa position initiale. Cet axe est l'axe perpendiculaire à la porte si l'on considère que le fauteuil est parfaitement droit devant celui-ci. Selon la configuration du manipulateur, ceci devrait uniquement faire tourner le dernier actuateur du Jaco².

3.5.2.2 Opérer une poignée à levier

L'opération d'une poignée à levier est un peu plus complexe. En premier lieu, il est demandé à l'effecteur de faire un mouvement de 10 cm vers le bas. Lors de son déplacement, celui-ci va subir une rotation angulaire, selon l'axe des « y » de la base, à cause de la poignée. En fonction de cette nouvelle orientation, le déplacement de 10 cm sera corrigé, en tout temps, pour permettre l'effecteur de suivre ce changement d'angle. De plus, pour aider la rotation de l'effecteur, il est aussi demandé à celui-ci, en tout temps, de tourner d'un angle de 16 degrés de plus que l'angle qu'il a subi jusqu'à maintenant. Pour ramener l'effecteur à sa position initiale, l'information de cette position est simplement renvoyée au manipulateur.

3.5.2.3 Ouvrir légèrement la porte

Finalement, pour ouvrir légèrement la porte, la trajectoire demande au manipulateur d'éloigner l'effecteur du fauteuil de 5 cm. Si après une seconde le manipulateur n'atteint pas au moins un déplacement de 4 cm, alors la trajectoire demande au manipulateur de rapprocher l'effecteur du fauteuil de 5 cm, tout en conservant l'information qu'il faut tirer la porte. Dans le cas contraire, il conserve l'information qu'il faut la pousser.

3.5.3 Ouvrir la porte

Jusqu'à maintenant, le fauteuil roulant restait immobile. Il sera nécessaire que l'utilisateur déplace son fauteuil pour parvenir à ouvrir la porte. Toutefois, le manipulateur Jaco² va tout de même assister l'utilisateur. Celui-ci va pousser/tirer la porte en fonction de l'information obtenue lorsque celui-ci opérait la poignée.

3.5.3.1 Pousser la porte

Pour pousser la porte, il est constamment demandé au manipulateur de déplacer son effecteur en avançant de 5 cm de façon perpendiculaire à la porte. Tout au long de l'opération, cette direction sera corrigée en fonction de l'orientation de l'effecteur. En ce qui concerne la rotation en « z », en fonction de la base, celle-ci est en tout temps remise à jour en fonction de l'orientation causée par la porte et le fauteuil.

3.5.3.2 Tirer la porte

Pour tirer la porte, la même méthode est utilisée, mais dans l'autre direction. Toutefois, il est important de mettre ici une limite pour que la porte ne frappe pas le fauteuil ou l'utilisateur. De ce fait, la trajectoire de l'effecteur s'arrête aux extrémités de la région de protection illustrée à la Figure 3.15. Cette région peut être facilement redéfinie en fonction de la position du manipulateur et des dimensions du fauteuil.

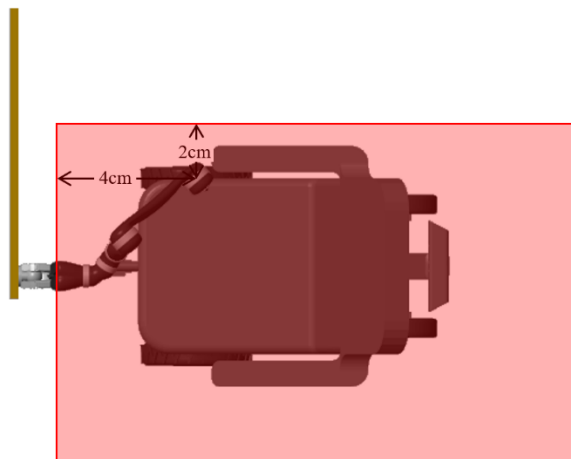


Figure 3.15 Région de protection du fauteuil

3.6 Expérimentations

Le schéma illustré à la Figure 3.16 montre comment les sous-systèmes s'assemblent pour permettre le fonctionnement du système global. Toutefois, bien que ces sous-systèmes aient été développés pour fonctionner ensemble, l'évaluation de ceux-ci a été réalisée à l'aide d'expérimentations séparées. Effectivement, puisque le système complet fonctionne de façon séquentielle, alors il est préférable d'évaluer chaque système de façon distincte pour vérifier le fonctionnement et les limites de chacun.

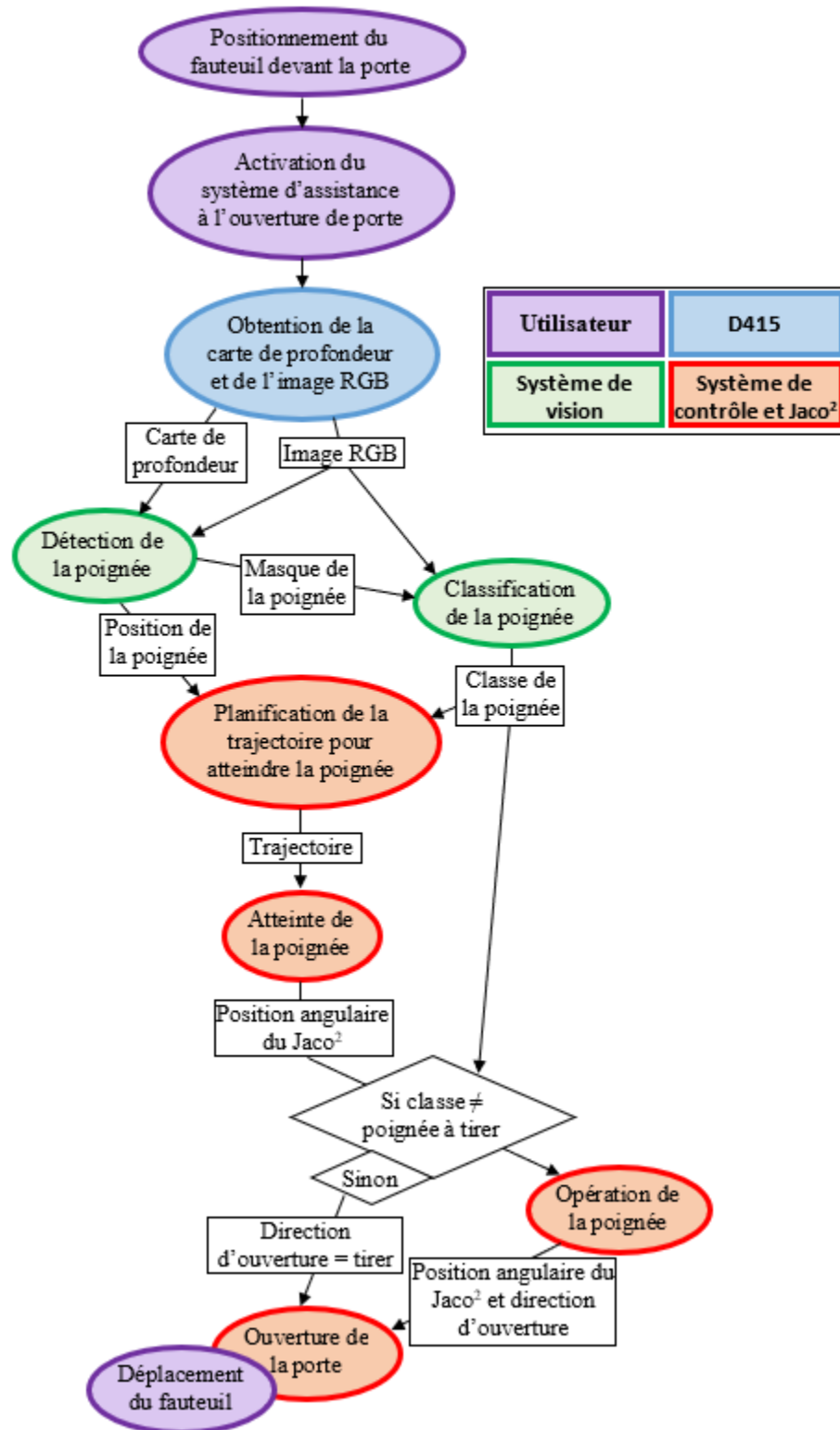


Figure 3.16 Schéma décrivant l'ensemble du système

3.6.1 Détection et classification de poignées de porte

3.6.1.1 Installation utilisée

Contrairement au contrôle du manipulateur Jaco², il n'est pas vraiment possible de simuler la détection et la classification de poignées de porte étant donné que cette section du projet nécessite l'utilisation d'une caméra réelle. De ce fait, l'installation illustrée à la Figure 3.17 a été mise en place dans le but d'évaluer la détection et la classification des poignées de porte. Cette installation est constituée d'un fauteuil roulant Métro d'Everest & Jannings, de la caméra Realsense D415, fixée selon la position et l'orientation présentées dans la section 3.2.2, et d'un ordinateur portable possédant le code Python permettant l'utilisation de la caméra et le traitement des images RGB et des cartes de profondeur.



Figure 3.17 Installation utilisée pour détecter et classer les poignées de porte



Figure 3.18 Expérimentation utilisant l'installation

3.6.1.2 Expérimentation utilisant l'installation

Comme illustrée à la Figure 3.18, l'installation a été positionnée devant diverses portes à l'aide d'un utilisateur pour détecter et classer les poignées. L'utilisateur devait, autant que possible, positionner le fauteuil en parallèle de la porte et au centre de celle-ci. Le fauteuil devait également

être positionné devant la porte à une distance inférieure à 80 cm par rapport à la position qu'aurait la base du manipulateur, environ à 50 cm par rapport aux portes-pieds. Ceci correspond à la région moyenne accessible par le manipulateur, dépendamment où la poignée se situe.

En ce qui concerne les portes, les expérimentations ont été réalisées avec 10 poignées de chaque classe pour un total de 30 expérimentations. De plus, deux portes identiques ne pouvaient pas compter comme deux expérimentations différentes. De ce fait, celles-ci devaient posséder au moins une différence entre elles soit :

- Une différence de poignée;
- Une différence de couleur de la porte;
- Et une différence de structure de la porte.

Pour chaque essai, le système devait enregistrer dans un fichier texte le type de la poignée ainsi que sa position cartésienne par rapport à la caméra. Ceci permet au système de contrôle d'utiliser par la suite ces fichiers textes pour en retirer l'information. De plus, les détections par Faster R-CNN ainsi que le masque de la poignée devaient apparaître sur l'écran pour analyser la qualité de la détection. Finalement, pour l'évaluation des résultats, les images couleur ainsi que les cartes de profondeur ont été enregistrées également.

3.6.1.3 Évaluer la détection avec et sans l'information 3D

Il est facile d'observer si le système est parvenu à détecter la poignée ou non. Toutefois, il serait intéressant aussi d'analyser la qualité de la détection, lorsque celle-ci est réussie, et d'observer si l'ajout de l'information 3D entraînerait une amélioration considérable ou non. De ce fait, pour chaque détection, l'*Intersection over Union* (IoU) a été calculée avec la boîte de région uniquement priorisée par Faster R-CNN, avec la boîte de région de la détection finale, et avec le masque de la poignée de porte.

L'IoU est calculé en comparant la région détectée avec la région désirée. Cette comparaison est réalisée mathématiquement en calculant l'aire commune des deux régions, aussi considérée comme étant la région vraie positive, et en la divisant avec l'aire totale recouverte par les deux régions, contenant la région vraie positive, fausse positive et vraie négative. De ce fait, le score résultant donne un ratio sur la qualité de la détection allant de zéro à un. Zéro indique qu'aucune région ne

s'entrecroise et un indique une détection parfaite. La Figure 3.19 illustre comment calculer l'IoU avec des boîtes de région. La même méthode peut également s'appliquer à des masques. [70]

Toutefois, pour calculer les IoU de chaque détection, il est nécessaire de connaître les régions recherchées de la poignée de porte. De ce fait, les 30 images enregistrées lors de l'expérimentation utilisant l'installation ont été par la suite traitées à la main pour représenter la boîte de région et le masque réels de la poignée.

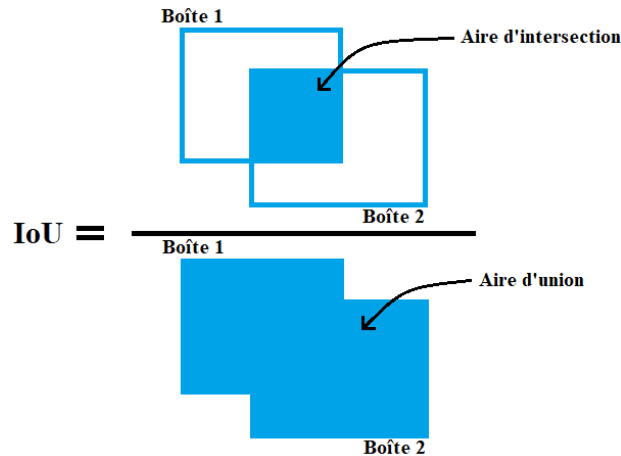


Figure 3.19 Calcul de l'IoU

3.6.2 Systèmes de contrôle

Les expérimentations avec les systèmes de contrôle ont été réalisés en simulation. Cette section présente donc les simulations réalisées pour permettre d'évaluer les 3 systèmes de contrôle développés.

3.6.2.1 Atteindre le dispositif d'ouverture

Pour atteindre le dispositif d'ouverture, six groupes de trajectoires ont été développés, possédant chacun leur ensemble de gains pour la boucle de contrôle. Ces trajectoires dépendent de la position et de la classe de la poignée. De ce fait, le système de contrôle implémenté a été évalué pour chaque groupe de trajectoire indépendamment. Pour chaque groupe de trajectoire, le manipulateur Jaco² devait réaliser la tâche d'atteindre le dispositif d'ouverture 100 fois. À chaque essai, la poignée a été positionnée à une position aléatoire, tout en représentant une situation réelle, selon les critères suivants :

- La poignée doit se situer à une hauteur entre 0.85 et 1.05 mètre du sol, ce qui est environ un décimètre en dessous et au-dessus de la hauteur habituelle des poignées de porte.
- La poignée doit être atteignable par le manipulateur sans nécessiter un déplacement du fauteuil. Une distance trop éloignée pour le manipulateur ne causera pas de problème d'instabilité au système, mais fausserait les résultats.
- La position de la poignée doit être en lien avec la trajectoire expérimentée. Par exemple, si l'expérimentation concerne les poignées positionnées à droite alors la poignée ne doit pas se trouver vers la gauche. Ceci ne causerait pas de problème au système étant donné que celui-ci s'adapte automatiquement, mais fausserait les résultats.
- Le plan de la porte doit se situer devant le fauteuil et la position résultante de la poignée doit respecter une distance perpendiculaire minimale de 0.4 mètre de la base du Jaco² ainsi qu'une distance minimale perpendiculaire de 0.1 mètre de l'effecteur lorsque le manipulateur est à sa position intermédiaire. Ces limites ont été choisies pour représenter la distance minimale entre le fauteuil et la porte et pour éviter que la position intermédiaire de l'effecteur entre en contact avec la porte. Il est important de mentionner que cette deuxième limite est déjà prise en compte dans le code et, si elle n'est pas respectée, le code envoie un message d'erreur à l'utilisateur pour lui indiquer qu'il est trop près de la porte.

L'objectif ici était d'observer si le système de contrôle est en mesure de suivre la trajectoire voulue, peu importe la position de la poignée.

De plus, à chaque essai, les paramètres de la dynamique utilisés pour la modélisation du manipulateur Jaco², présenté à l'Annexe B ont été également modifiés de façon aléatoire. Ceux-ci ont été individuellement multipliés par un modificateur allant de 0.9 à 1.1, pour générer une déviation de 10%. L'objectif de cette modification est de pallier la dynamique exacte du manipulateur qui est inconnue.

Finalement, chaque expérimentation a été réalisée sur une période de 30 secondes pour laisser assez de temps au manipulateur pour atteindre la poignée. Pour chaque expérimentation, l'erreur moyenne et l'erreur maximale de chaque actuateur, pour l'ensemble de la trajectoire, ont été calculées ainsi que l'erreur de position et d'orientation cartésiennes finales. Ensuite, sur l'ensemble

des expérimentations, la moyenne, avec l'écart type, et la valeur maximale de ces résultats ont été conservées.

3.6.2.2 Opérer le dispositif d'ouverture

Étant donné que la capacité du Jaco² à opérer la poignée peut différer selon le type de poignée et si celle-ci se trouve vers la gauche ou la droite de la porte, le système de contrôle implémenté a été évalué pour chacune de ces conditions. Pour chaque ensemble de conditions, le manipulateur Jaco² devait réaliser la tâche d'opérer le dispositif d'ouverture 100 fois. Chaque expérimentation a été réalisée :

- Sur une période de 15 secondes;
- En positionnant la poignée de façon aléatoire, selon les mêmes critères que pour atteindre le dispositif d'ouverture, tout en permettant au manipulateur de la pousser d'au moins 5cm pour lui permettre de déloger la porte du cadre;
- En modifiant aléatoirement les paramètres de la dynamique du manipulateur, selon les mêmes critères que pour atteindre le dispositif d'ouverture;
- En multipliant aléatoirement la résistance du ressort de la poignée par un modificateur allant de 0.8 à 1.2, pour générer une déviation de 20%.

Les résultats ont ensuite été recueillis pour but d'évaluer la capacité du système à :

- Tourner la poignée selon la résistance du ressort, en évaluant l'angle de rotation atteint par la poignée;
- Maintenir la poignée immobile lors du déplacement de la porte, en calculant les erreurs moyennes et les erreurs maximales angulaires de la poignée;
- Maintenir la porte ouverte, en vérifiant si la poignée maintient au moins une distance de 2 cm par rapport à sa position initiale lorsque la porte est ouverte;
- Conclure si la porte s'ouvre en poussant ou en tirant.

Pour les erreurs angulaires de la poignée, seules la moyenne, avec l'écart type, et la valeur maximale de ces résultats, sur l'ensemble des 100 expérimentations, ont été conservées pour l'analyse.

3.6.2.3 Ouvrir la porte

Pour ouvrir la porte, le manipulateur et l'utilisateur doivent travailler simultanément. De ce fait, lors des expérimentations, les mouvements du fauteuil dans la simulation ont été contrôlés en temps réel par un utilisateur à l'aide d'une manette. Celui-ci avait comme objectif d'assister le manipulateur dans le but d'ouvrir la porte d'au moins 90°. Il était en mesure de voir en temps réel la simulation de la tâche à l'aide d'une animation.

La capacité du Jaco² à ouvrir la porte peut différer selon quatre conditions :

- La classe de la poignée;
- La position (gauche/droite) de la poignée sur la porte;
- La direction d'ouverture de la porte (tirer/pousser), à l'exception des poignées à tirer qui sont rarement utilisées pour pousser une porte;
- La position du fauteuil par rapport à la porte.

De ce fait, pour chaque ensemble de conditions, le manipulateur Jaco² devait réaliser la tâche d'opérer le dispositif d'ouverture 10 fois pour un total de 200 expérimentations. Pour chaque essai, la valeur des paramètres de la dynamique du manipulateur a été modifiée de façon aléatoire en suivant les mêmes critères que pour atteindre le dispositif d'ouverture.

Pour la position du fauteuil par rapport à la porte, 100 expérimentations ont été réalisées en positionnant la poignée de façon aléatoire en suivant les mêmes critères que pour atteindre le dispositif d'ouverture. Dans ces conditions, le fauteuil pouvait se retrouver partiellement à l'extérieur des extrémités de la porte comme le montre la Figure 3.20. Toutefois, en pratique, il est préférable pour l'utilisateur d'être complètement devant la porte comme à la Figure 3.21 étant donné que cette position permet au fauteuil de traverser plus facilement la porte. De ce fait, pour les 100 autres expérimentations, la poignée a été positionnée de façon aléatoire par rapport à la base du manipulateur tout en assurant que le fauteuil soit complètement devant la porte.



Figure 3.20 Fauteuil partiellement à l'extérieur de la porte

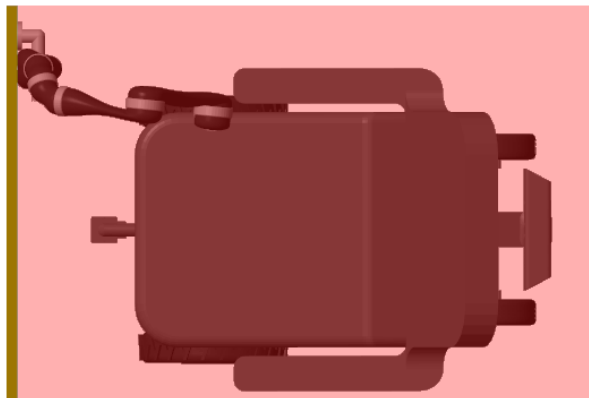


Figure 3.21 Fauteuil complètement devant la porte

Pour évaluer le système, la tâche d'ouvrir la porte a été considérée comme réussie si l'individu était en mesure d'ouvrir la porte d'au moins 90° en moins de 30 secondes. De plus, les forces et les couples exercés sur la poignée par l'effecteur ont également été mesurés à chaque essai pour évaluer si l'effecteur risque de perdre sa prise sur la poignée en pratique.

CHAPITRE 4 RÉSULTATS ET ANALYSE

4.1 Détection et classification de la poignée de porte

4.1.1 Entraînement de l'algorithme Faster R-CNN et HOG

Avant de passer aux expérimentations réalisées avec la caméra, il est important d'évaluer l'efficacité des deux algorithmes entraînés, soit : la détection par Faster R-CNN et la classification par HOG.

4.1.1.1 Détection par Faster R-CNN

En utilisant l'évaluateur COCO avec les modèles préentraînés disponibles dans la bibliothèque Torchvision de Pytorch, diverses métriques d'évaluations, présentes dans l'Annexe D, sont offertes automatiquement. Toutefois, dans le but de comparer le modèle développé avec celui de l'article de Arduengo et coll. (2021) [26], utilisant YOLO, seul le score de mAP selon les critères de la compétition PASCAL VOC 2012 [71] a été pris en compte, étant donné que c'est la méthode d'évaluation utilisée dans l'article. Celui-ci calcule le mAP avec les résultats de détection possédant un score IoU d'au moins 0.5. Le score du modèle Faster R-CNN entraîné pour le projet est présenté dans le Tableau 4.1, ainsi que le score de l'article de Arduengo et coll. (2021) [26], utilisant YOLO. Comme démontré, le score obtenu est très similaire à celui de l'algorithme YOLO de la littérature utilisant la même base de données. Toutefois, en comparant ce résultat avec celui de l'algorithme Faster R-CNN utilisé dans l'article de Stuede et coll. (2019) [15], qui a obtenu un score de 0.95mAP@0.5IoU, on peut conclure que ce résultat reste tout de même faible. Cependant, étant donné que dans cet article le système détecte uniquement les poignées à levier, qu'il n'a pas été entraîné et évalué avec la même base de données, et que l'information sur celle-ci n'est pas disponible, cette comparaison doit être prise à la légère.

Tableau 4.1 Comparaison des scores mAP

Méthode	mAP@0.5IoU
Faster R-CNN du projet	0.46
YOLO de la littérature [26]	0.45

4.1.1.2 Classification avec HOG

En ce qui concerne la classification avec HOG, le résultat obtenu après l'entraînement du modèle montrait un score sans faute avec les 15 images tests. De plus, 15 autres images de poignée de porte ont également été prises à Polytechnique Montréal avec la caméra Realsense. Le modèle a par la suite été testé sur leur vecteur de caractéristique HOG et est parvenu à classer toutes les poignées sans faire d'erreur.

4.1.2 Détection des poignées de porte

4.1.2.1 Résultats

Tableau 4.2 Résultats de la détection des poignées rondes

Numéro	Détection	Classification	IoU sans information 3D	IoU avec information 3D	IoU du masque
1	Oui	Oui	0.00	0.74	0.68
2	Oui	Oui	0.00	0.65	0.56
3	Oui	Oui	0.39	0.84	0.74
4	Oui	Oui	0.00	0.70	0.70
5	Oui	Oui	0.36	0.61	0.57
6	Oui	Oui	0.34	0.71	0.70
7	Non	Oui	0.44	0.47	0.37
8	Oui	Oui	0.00	0.73	0.66
9	Oui	Oui	0.00	0.66	0.65
10	Oui	Oui	0.41	0.85	0.72

Tableau 4.3 Résultats de la détection des poignées à levier

Numéro	Détection	Classification	IoU sans information 3D	IoU avec information 3D	IoU du masque
1	Oui	Oui	0.61	0.60	0.54
2	Oui	Oui	0.00	0.77	0.64
3	Oui	Oui	0.42	0.53	0.55
4	Oui	Oui	0.37	0.74	0.70
5	Oui	Oui	0.58	0.69	0.66
6	Oui	Oui	0.67	0.74	0.53
7	Oui	Oui	0.00	0.68	0.50
8	Oui	Oui	0.63	0.72	0.61
9	Oui	Oui	0.57	0.58	0.51
10	Oui	Oui	0.54	0.52	0.53

Tableau 4.4 Résultats de la détection des poignées à tirer

Numéro	Détection	Classification	IoU sans information 3D	IoU avec information 3D	IoU du masque
1	Non	-	0.00	-	-
2	Oui	Oui	0.00	0.84	0.55
3	Non	-	0.00	-	-
4	Oui	Oui	0.00	0.60	0.45
5	Non	-	0.00	-	-
6	Non	Non	0.10	0.08	0.14
7	Non	-	0.00	-	-
8	Non	Non	0.07	0.06	0.02
9	Oui	Oui	0.00	0.58	0.48
10	Oui	Oui	0.00	0.88	0.57

Le Tableau 4.2, Tableau 4.3 et Tableau 4.4 illustrent respectivement les résultats obtenus lors des expérimentations avec les poignées rondes, les poignées à levier et les poignées à tirer. En ce qui concerne la détection, celle-ci a été considérée comme réussie si la boîte de région finale possédait un score de IoU avec information 3D d'au moins 0.5. Ceci étant dit, les tableaux démontrent une réussite de 90% pour les poignées rondes, de 100% pour les poignées à levier et de 40% pour les poignées à tirer. L'analyse concernant la cause de ces résultats sera discutée davantage selon chaque type de poignées.

4.1.2.2 Poignée ronde

Concernant la mauvaise détection de la poignée 7, le Tableau 4.2 permet de remarquer que le IoU n'est pas très loin sous la valeur de 0.5 et que la classification a été tout de même bien réussie. Effectivement, comme le montre la Figure 4.1, la méthode Faster R-CNN est parvenue à détecter la poignée sur l'image et l'information tridimensionnelle a permis de conserver cette détection. Toutefois, si l'on compare le masque de la poignée obtenu, résultant de l'information tridimensionnelle, avec celui désiré, on remarque que ceux-ci sont grandement différents. En effet, la poignée est surmontée par un mécanisme de sécurité qui est, lui aussi, devant la porte. De ce fait, presque tous les pixels dans la détection Faster R-CNN ont été considérés comme faisant partie de la poignée. De plus, étant donné que la boîte de région, résultant du Faster R-CNN, est trop grande, un grand nombre de pixels ont été pris en considération. Ceci peut entraîner une mauvaise position du centre de la poignée et donc empêcher le manipulateur d'atteindre convenablement celle-ci.

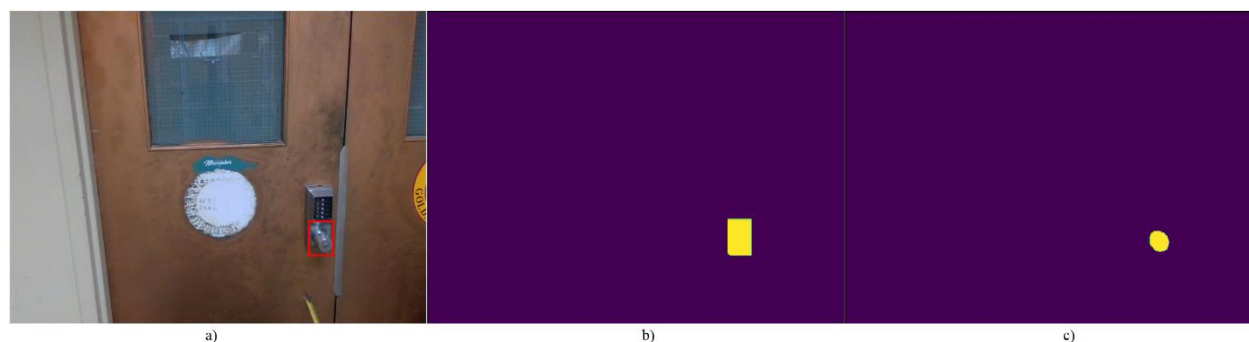


Figure 4.1 Détection de la poignée ronde numéro 7. a) illustre la détection réalisée par Faster R-CNN. b) et c) illustrent respectivement le masque de la poignée détecté et désiré

À l'exception de ce cas particulier, la détection a été un succès à chaque fois pour les poignées rondes. Toutefois, le Tableau 4.2 montre que ceci ne serait pas vraiment possible sans l'information tridimensionnelle. Effectivement, le IoU de la détection sans l'information tridimensionnel est souvent nul, malgré le succès de la détection. Ceci signifie que la détection Faster R-CNN a bel et bien détecté la poignée désirée, mais qu'elle a priorisé une autre détection. Par exemple, dans la Figure 4.2, Faster R-CNN est parvenu à détecter la poignée, mais a priorisé la charnière, pensant qu'il y avait plus de chances que celle-ci soit la poignée en question. C'est grâce à l'information tridimensionnelle que la charnière a pu être éliminée des choix.

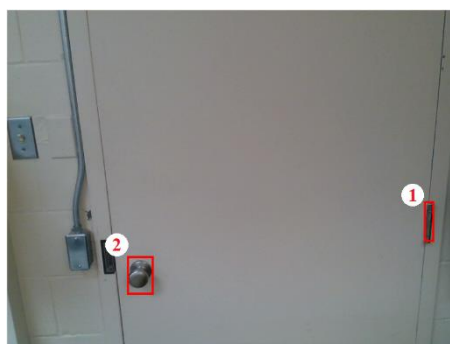


Figure 4.2 Détection de la poignée ronde 1 par Faster R-CNN

En ce qui concerne les IoU non-nuls sans l'information tridimensionnelle, ceux-ci restent tout de même plus faibles que les IoU avec l'information tridimensionnelle. Ceci est dû au fait que dans tous les cas, la boîte de région de la poignée, résultante du Faster R-CNN, est beaucoup trop grande, comme on peut le voir à la Figure 4.1 a) et à la Figure 4.2. Ce problème est ensuite corrigé avec l'information tridimensionnelle, sauf pour la poignée ronde numéro 7, puisque celle-ci permet de ressortir, en quelque sorte, la forme de la poignée. Toutefois, les masques générés ne sont pas

parfaits. Effectivement, même si les IoU des masques sont au-dessus de 0.5 pour les détections réussies, le Tableau 4.2 montre qu'aucun d'entre eux ne parvient à atteindre un score de 0.75. En analysant les détections, comme à la Figure 4.3, on observe que, pour les poignées rondes, le masque est toujours un peu plus grand que celui désiré. Il est fort probable que ceci soit causé par le bruit présent dans la carte de profondeur.

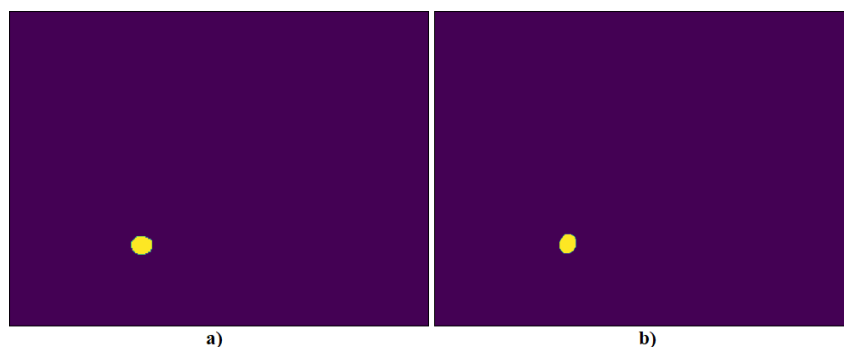


Figure 4.3 Masques de détection de la poignée ronde numéro 6. a) et b) illustre respectivement le masque de la poignée détecté et désiré

4.1.2.3 Poignée à levier

Pour les poignées à levier, le Tableau 4.3 permet d'observer que ce type de poignée est beaucoup plus facile à détecter par le Faster R-CNN que les poignées rondes. Effectivement, le IoU de la détection sans l'information tridimensionnelle est très rarement nul, indiquant que la détection Faster R-CNN a plus de facilité à prioriser et à détecter les poignées à levier. De plus, celui-ci est aussi généralement plus élevé que le IoU sans information tridimensionnelle des poignées rondes. En analysant les détections par Faster R-CNN, comme à la Figure 4.4 a), on observe que les boîtes de régions encadrent beaucoup mieux les poignées à levier.

Toutefois, contrairement aux poignées rondes, l'information tridimensionnelle n'entraîne pas une aussi grande amélioration du IoU. Dans certains cas, l'IoU est même un peu réduite, tout en restant très similaire. En effet, comme le montre la Figure 4.4 b) par rapport à la c), le système semble avoir plus de difficulté à obtenir le masque des poignées à levier. Plus spécifiquement, il semble avoir de la difficulté à détecter le centre des poignées. Dû à la faible épaisseur de celles-ci, il est possible que la caméra stéréovision ait de la difficulté à repérer leurs centres. Toutefois, ceci ne devrait pas affecter grandement la détection de la position de la poignée. En effet, si les deux

extrémités sont détectées, alors le point central reste facile à déterminer. La profondeur sera toutefois peut-être légèrement affectée.

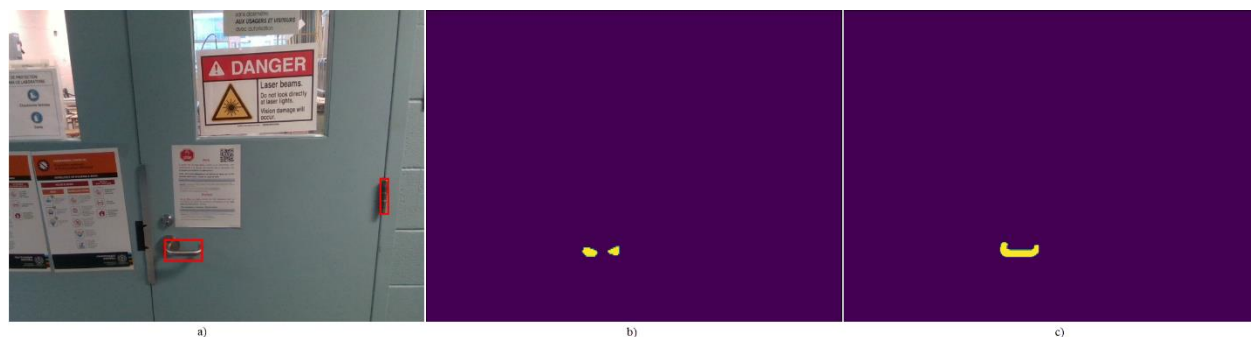


Figure 4.4 Détection de la poignée à levier numéro 1. a) illustre la détection réalisée par Faster R-CNN. b) et c) illustrent respectivement le masque de la poignée détecté et désiré

4.1.2.4 Poignée à tirer

Pour les poignées à tirer, la détection était très difficile. Effectivement, à l'intérieur du Tableau 4.4, plusieurs cases sont simplement vides. Ceci est le cas lorsqu'aucune détection finale n'a été conservée. En effet, dans toutes les rangées du tableau dont il manque de l'information, Faster R-CNN n'a tout simplement pas été en mesure de repérer la poignée et l'information tridimensionnelle a éliminé, de son côté, toutes les mauvaises détections ne laissant ainsi aucune détection.

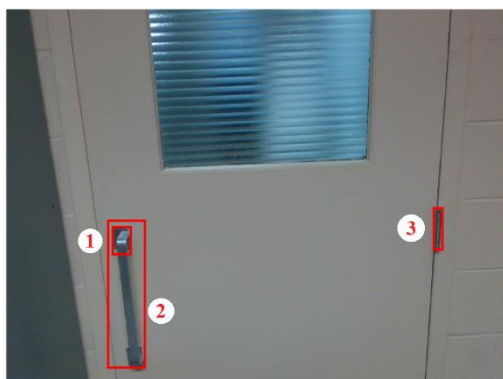


Figure 4.5 Détection de la poignée à tirer 6 par Faster R-CNN

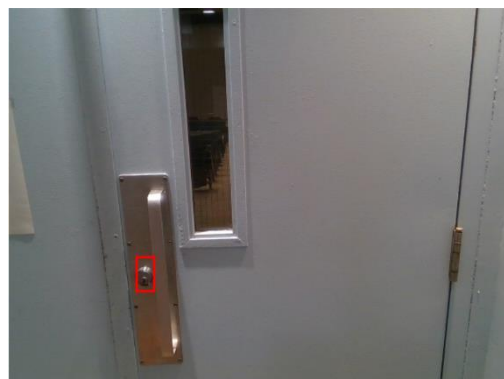


Figure 4.6 Détection de la poignée à tirer 8 par Faster R-CNN

En ce qui concerne la poignée 6 et 8, leurs détections par Faster R-CNN sont respectivement présentées à la Figure 4.5 et à la Figure 4.6. Comme on peut le voir pour la poignée numéro 8, le Faster R-CNN n'a pas non plus été en mesure de détecter la poignée. Toutefois, la boîte de région de la serrure était un peu en contact avec la poignée, ce qui a généré une fausse détection. Pour ce qui est de la poignée 6, on remarque que le système a été capable de repérer la poignée, mais celui-ci a simplement priorisé la région supérieure de cette dernière au lieu de prioriser l'ensemble. Ceci a donc également généré une fausse détection.

En ce qui concerne les détections réussies, le Tableau 4.4 montre que dans aucune situation, la détection par Faster R-CNN n'a priorisé la poignée à tirer. Effectivement, c'est grâce à l'information tridimensionnelle que la poignée a pu être trouvée. Sur les quatre réussites, le IoU de la détection avec l'information tridimensionnelle a même démontré à chaque coup un score très acceptable. Toutefois, ces scores sont drastiquement réduits lorsqu'on remarque le IoU des masques. La Figure 4.7 et la Figure 4.8 illustrent deux causes différentes expliquant cette réduction. D'un côté, comme à la Figure 4.7, lorsque la poignée à tirer est plus éloignée de la surface de la porte, la carte de profondeur semble élargir la région de la poignée, comme pour les poignées rondes. De l'autre côté, comme la Figure 4.8, lorsque la poignée à tirer est très approchée de la porte, le système semble avoir de la difficulté à différencier celle-ci du plan de la porte, générant un masque non complet. Même si, pour les deux poignées présentées à la Figure 4.7 et à la Figure 4.8, ce n'est pas vraiment problématique pour obtenir la position centrale, ceci pourrait s'avérer problématique si le problème présenté à la Figure 4.8 se réalisait avec une poignée comme celle de la Figure 4.7.

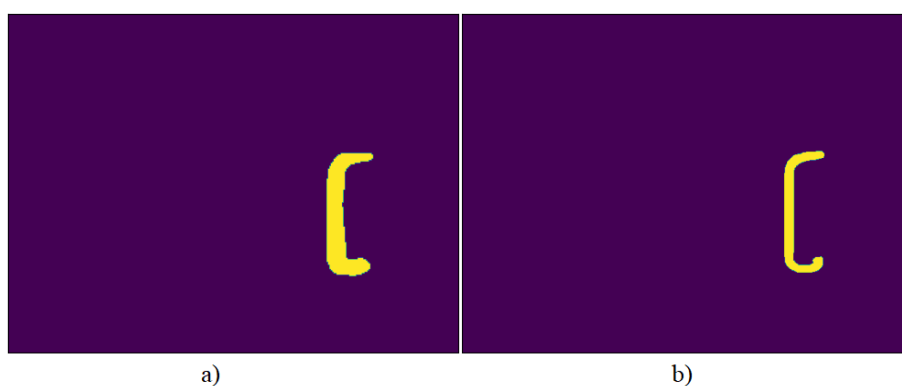


Figure 4.7 Masques de détection de la poignée à tirer numéro 2. a) et b) illustrent respectivement le masque de la poignée détecté et désiré

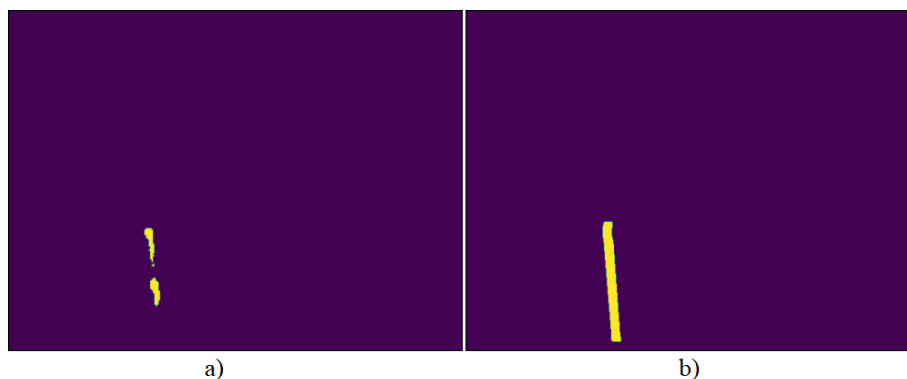


Figure 4.8 Masques de détection de la poignée à tirer numéro 4. a) et b) illustrent respectivement le masque de la poignée détecté et désiré

4.1.3 Classification des poignées de portes

En ce qui concerne la classification des poignées, comme le montrent le Tableau 4.2, le Tableau 4.3 et le Tableau 4.4, lorsque le système parvenait à repérer la poignée de porte, la classification se réalisait avec succès à chaque essai. Ceci inclut même la poignée ronde numéro 7. Même si, dans ce cas, la détection a été considérée comme un échec, le système avait tout de même repéré la poignée et la classification a été réalisée avec succès.

Toutefois, comme le montre la situation de la poignée à tirer numéro 6, illustré à la Figure 4.5, un bon encadrement de la poignée est nécessaire pour assurer une bonne classification. Effectivement, dans cette situation, seule la partie supérieure de la poignée a été considérée. De ce fait, lors de la classification, le logiciel a manqué d'information pour réaliser son analyse et a considéré celle-ci comme une poignée à levier.

4.2 Systèmes de contrôle

L'ensemble des gains utilisé pour le projet est disponible à l'Annexe E.

4.2.1 Atteindre le dispositif d'ouverture

4.2.1.1 Résultats

Les Tableaux 4.5 à 4.7 montrent la moyenne et le maximum sur 100 essais des erreurs calculés pour les 6 groupes de trajectoires permettant d'atteindre la poignée de porte. Les erreurs angulaires sont

par actionneurs, de la base jusqu'à l'effecteur, et en fonction de la trajectoire réalisée au complet, tandis que les erreurs cartésiennes concernent l'effecteur à la fin du trajet.

4.2.1.2 Erreurs de l'effecteur

En ce qui concerne les positions cartésiennes finales de l'effecteur, on constate que l'erreur de celles-ci est très faible pour l'ensemble des trajectoires. Effectivement, en évaluant la pire situation, soit : l'erreur maximale obtenue dans tous les essais, on constate que le système parvient à maintenir une position finale dont l'erreur est toujours inférieure à 1 millimètre. Pour l'orientation finale de l'effecteur, la pire situation est en dessous de 0.06 degré, ce qui est également très faible.

4.2.1.3 Erreurs des actionneurs

En ce qui concerne les erreurs angulaires de la trajectoire, il est clair que les trajectoires sont généralement très bien respectées, peu importe la position et la classe de la poignée. En effet, sur l'ensemble des essais, la moyenne de l'erreur des actionneurs reste pratiquement tout le temps en dessous de 0.3 degré, en prenant en compte les écarts types. Même la plus grande moyenne de l'erreur atteinte reste en dessous de 0.5 degré. Toutefois, si on regarde l'erreur angulaire maximale atteinte, on remarque que les actionneurs peuvent atteindre une erreur se rapprochant de 4 degrés. Par chance, ceci ne semble pas arriver souvent dans une même trajectoire, comme le montre les moyennes par essais des Tableaux 4.2 à 4.4. Toutefois, la moyenne des erreurs maximales montre tout de même que l'ensemble des trajectoires réalisent généralement au moins une erreur angulaire pouvant atteindre presque 1 degré. Ces valeurs ne sont pas catastrophiques, mais pourraient poser problème si cette déviation est causée lorsque l'effecteur atteint la poignée. Cependant, après une analyse plus approfondie des trajectoires, il a été remarqué que ces erreurs ne se produisent pas lorsque l'effecteur est près de la poignée. Comme le montre le graphique à la Figure 4.9, ces erreurs sont produites au tout début des trajectoires. Ceci est causé par la fréquence de communication de 110 Hz. En effet, lorsque les premières commandes sont envoyées au manipulateur, celui-ci ne réagit pas tout de suite au signal. Ceci entraîne une erreur perçue par le correcteur qui tente immédiatement de corriger la situation. Ceci entraîne donc le système à trop compenser l'erreur et génère une oscillation de faible amplitude au commencement de la trajectoire. Heureusement, le système parvient rapidement à amortir cette oscillation, malgré la continuité du délai, ce qui n'entraîne pas de grande erreur angulaire sur le reste de la trajectoire.

Tableau 4.5 Erreurs sur les 100 essais pour atteindre une poignée à levier *

Erreurs de l'essai	Gauche		Droite	
	Moyenne des 100	Maximum des 100	Moyenne des 100	Maximum des 100
Moyenne angulaire (deg)	$\begin{bmatrix} 0.005 \\ 0.005 \\ 0.02 \\ 0.06 \\ 0.012 \\ 0.013 \end{bmatrix} \pm \begin{bmatrix} 0.001 \\ 0.002 \\ 0.01 \\ 0.04 \\ 0.006 \\ 0.006 \end{bmatrix}$	$\begin{bmatrix} 0.009 \\ 0.010 \\ 0.05 \\ 0.18 \\ 0.027 \\ 0.031 \end{bmatrix}$	$\begin{bmatrix} 0.003 \\ 0.005 \\ 0.02 \\ 0.07 \\ 0.012 \\ 0.013 \end{bmatrix} \pm \begin{bmatrix} 0.001 \\ 0.002 \\ 0.01 \\ 0.04 \\ 0.006 \\ 0.007 \end{bmatrix}$	$\begin{bmatrix} 0.007 \\ 0.012 \\ 0.06 \\ 0.18 \\ 0.032 \\ 0.033 \end{bmatrix}$
Maximum angulaire (deg)	$\begin{bmatrix} 0.04 \\ 0.04 \\ 0.5 \\ 1.1 \\ 0.2 \\ 0.4 \end{bmatrix} \pm \begin{bmatrix} 0.02 \\ 0.03 \\ 0.3 \\ 0.7 \\ 0.1 \\ 0.2 \end{bmatrix}$	$\begin{bmatrix} 0.09 \\ 0.14 \\ 1.5 \\ 3.6 \\ 0.5 \\ 1.1 \end{bmatrix}$	$\begin{bmatrix} 0.04 \\ 0.05 \\ 0.5 \\ 1.2 \\ 0.2 \\ 0.4 \end{bmatrix} \pm \begin{bmatrix} 0.02 \\ 0.02 \\ 0.3 \\ 0.7 \\ 0.1 \\ 0.2 \end{bmatrix}$	$\begin{bmatrix} 0.10 \\ 0.12 \\ 1.3 \\ 3.1 \\ 0.6 \\ 1.0 \end{bmatrix}$
Position cartésienne (10^{-4} m)	3 ± 1	9	2.3 ± 0.7	5.7
Orientation cartésienne (deg)	0.005 ± 0.001	0.012	0.004 ± 0.001	0.007

*La résolution des variables est en fonction de l'écart type du joint.

Tableau 4.6 Erreurs sur les 100 essais pour atteindre une poignée ronde *

Erreurs de l'essai	Gauche		Droite	
	Moyenne des 100	Maximum des 100	Moyenne des 100	Maximum des 100
Moyenne angulaire (deg)	$\begin{bmatrix} 0.008 \\ 0.007 \\ 0.03 \\ 0.018 \\ 0.04 \\ 0.03 \end{bmatrix} \pm \begin{bmatrix} 0.002 \\ 0.003 \\ 0.01 \\ 0.009 \\ 0.02 \\ 0.01 \end{bmatrix}$	$\begin{bmatrix} 0.015 \\ 0.019 \\ 0.08 \\ 0.043 \\ 0.09 \\ 0.07 \end{bmatrix}$	$\begin{bmatrix} 0.006 \\ 0.006 \\ 0.02 \\ 0.03 \\ 0.09 \\ 0.05 \end{bmatrix} \pm \begin{bmatrix} 0.005 \\ 0.003 \\ 0.01 \\ 0.02 \\ 0.07 \\ 0.03 \end{bmatrix}$	$\begin{bmatrix} 0.026 \\ 0.015 \\ 0.06 \\ 0.11 \\ 0.34 \\ 0.15 \end{bmatrix}$
Maximum angulaire (deg)	$\begin{bmatrix} 0.08 \\ 0.06 \\ 0.4 \\ 0.3 \\ 0.4 \\ 0.7 \end{bmatrix} \pm \begin{bmatrix} 0.02 \\ 0.03 \\ 0.3 \\ 0.1 \\ 0.2 \\ 0.2 \end{bmatrix}$	$\begin{bmatrix} 0.19 \\ 0.12 \\ 1.2 \\ 0.6 \\ 0.9 \\ 1.1 \end{bmatrix}$	$\begin{bmatrix} 0.06 \\ 0.05 \\ 0.3 \\ 0.6 \\ 0.5 \\ 0.9 \end{bmatrix} \pm \begin{bmatrix} 0.03 \\ 0.03 \\ 0.2 \\ 0.4 \\ 0.3 \\ 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.18 \\ 0.15 \\ 0.7 \\ 1.6 \\ 1.5 \\ 2.2 \end{bmatrix}$
Position cartésienne (10^{-4} m)	3 ± 1	5	3 ± 1	5
Orientation cartésienne (deg)	0.005 ± 0.005	0.03	0.004 ± 0.005	0.05

*La résolution des variables est en fonction de l'écart type du joint.

Tableau 4.7 Erreurs sur les 100 essais pour atteindre une poignée à tirer *

Erreurs de l'essai	Gauche		Droite	
	Moyenne des 100	Maximum des 100	Moyenne des 100	Maximum des 100
Moyenne angulaire (deg)	$\begin{bmatrix} 0.004 \\ 0.006 \\ 0.02 \\ 0.05 \\ 0.14 \\ 0.019 \end{bmatrix} \pm \begin{bmatrix} 0.001 \\ 0.003 \\ 0.01 \\ 0.03 \\ 0.09 \\ 0.008 \end{bmatrix}$	$\begin{bmatrix} 0.008 \\ 0.014 \\ 0.05 \\ 0.14 \\ 0.43 \\ 0.04 \end{bmatrix}$	$\begin{bmatrix} 0.005 \\ 0.005 \\ 0.03 \\ 0.08 \\ 0.11 \\ 0.02 \end{bmatrix} \pm \begin{bmatrix} 0.003 \\ 0.003 \\ 0.02 \\ 0.06 \\ 0.08 \\ 0.01 \end{bmatrix}$	$\begin{bmatrix} 0.015 \\ 0.013 \\ 0.10 \\ 0.23 \\ 0.34 \\ 0.06 \end{bmatrix}$
Maximum angulaire (deg)	$\begin{bmatrix} 0.05 \\ 0.05 \\ 0.3 \\ 0.7 \\ 0.7 \\ 0.6 \end{bmatrix} \pm \begin{bmatrix} 0.02 \\ 0.03 \\ 0.2 \\ 0.4 \\ 0.3 \\ 0.2 \end{bmatrix}$	$\begin{bmatrix} 0.12 \\ 0.15 \\ 0.9 \\ 1.8 \\ 1.8 \\ 1.0 \end{bmatrix}$	$\begin{bmatrix} 0.06 \\ 0.05 \\ 0.5 \\ 1.0 \\ 0.9 \\ 0.5 \end{bmatrix} \pm \begin{bmatrix} 0.03 \\ 0.03 \\ 0.3 \\ 0.7 \\ 0.6 \\ 0.2 \end{bmatrix}$	$\begin{bmatrix} 0.15 \\ 0.12 \\ 1.4 \\ 2.7 \\ 2.6 \\ 1.1 \end{bmatrix}$
Position cartésienne (10^{-4} m)	3 ± 1	6	2 ± 1	4
Orientation cartésienne (deg)	0.005 ± 0.006	0.049	0.004 ± 0.002	0.016

*La résolution des variables est en fonction de l'écart type du joint.

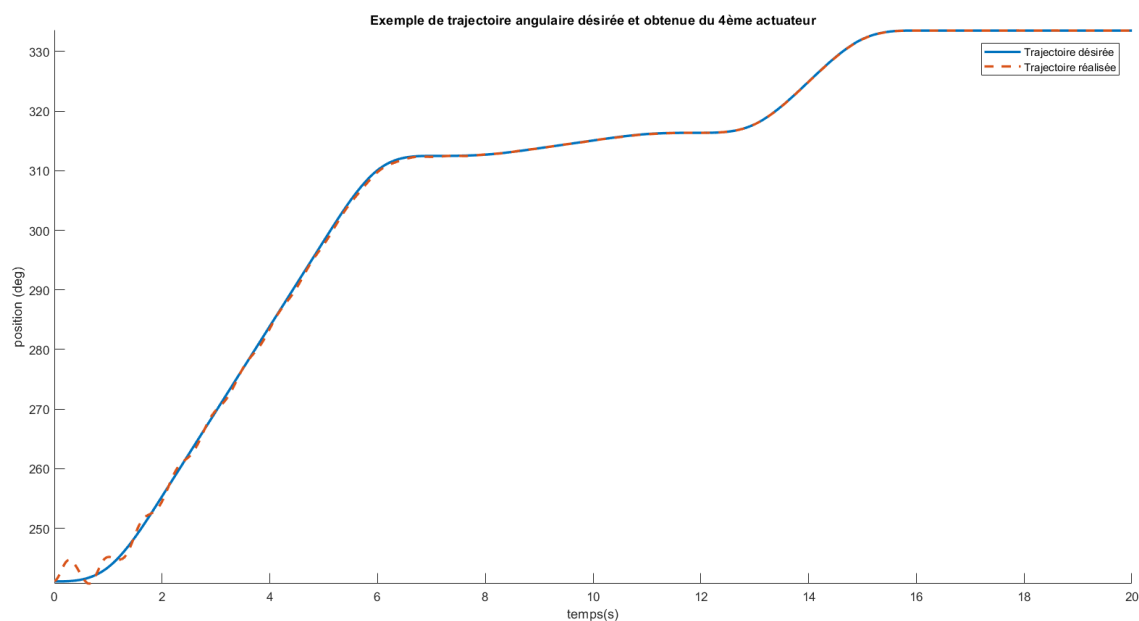


Figure 4.9 Trajectoire désirée et obtenue de l'essai causant la plus grande erreur angulaire

Finalement, il est important de mentionner que, peu importe sa position et sa classe, il faut moins de 30 secondes à l'effecteur pour atteindre la poignée. Toutefois, si la poignée est ronde ou à levier,

alors ce temps semble réduire. En effet, après les expérimentations, il a été déterminé que l'effecteur était capable à chaque essai d'atteindre les poignées rondes et à levier en moins de 25 secondes et 20 secondes, respectivement.

4.2.2 Opérer le dispositif d'ouverture

4.2.2.1 Résultats

Le Tableau 4.8 et le Tableau 4.9 montrent la moyenne et le maximum sur 100 essais des erreurs angulaires obtenues par la poignée lorsque celle-ci devait rester immobile. De ce fait, l'erreur de la poignée a été vérifiée lorsque la porte est en mouvement. De plus, le Tableau 4.10 montre le pourcentage de conclusions réussies par le système ainsi que le pourcentage de réussite du manipulateur à parvenir à maintenir la porte ouverte. Finalement, la Figure 4.10 et la Figure 4.11 illustrent la capacité du manipulateur à tourner la poignée en fonction de la résistance de son ressort ainsi que de la distance entre celle-ci et la base du Jaco².

Tableau 4.8 Erreurs angulaires sur les 100 essais des poignées à levier

Erreurs de l'essai	Gauche		Droite	
	Moyenne des 100	Maximum des 100	Moyenne des 100	Maximum des 100
Moyenne (deg)	1.3 ± 0.5	2.8	1.0 ± 0.8	6.6
Maximal (deg)	3 ± 1	6	2 ± 1	11

Tableau 4.9 Erreurs angulaires sur les 100 essais des poignées rondes

Erreurs de l'essai	Gauche		Droite	
	Moyenne des 100	Maximum des 100	Moyenne des 100	Maximum des 100
Moyenne (deg)	0.0 ± 0.1	0.5	0.1 ± 0.1	0.5
Maximal (deg)	0.1 ± 0.2	0.9	0.1 ± 0.3	0.9

Tableau 4.10 Pourcentage de conclusion et d'ouverture de la porte réussi sur les 100 essais

	Poignée à levier		Poignée ronde	
	Gauche	Droite	Gauche	Droite
Conclusion (%)	100	100	100	100
Ouverture (%)	100	100	94	98

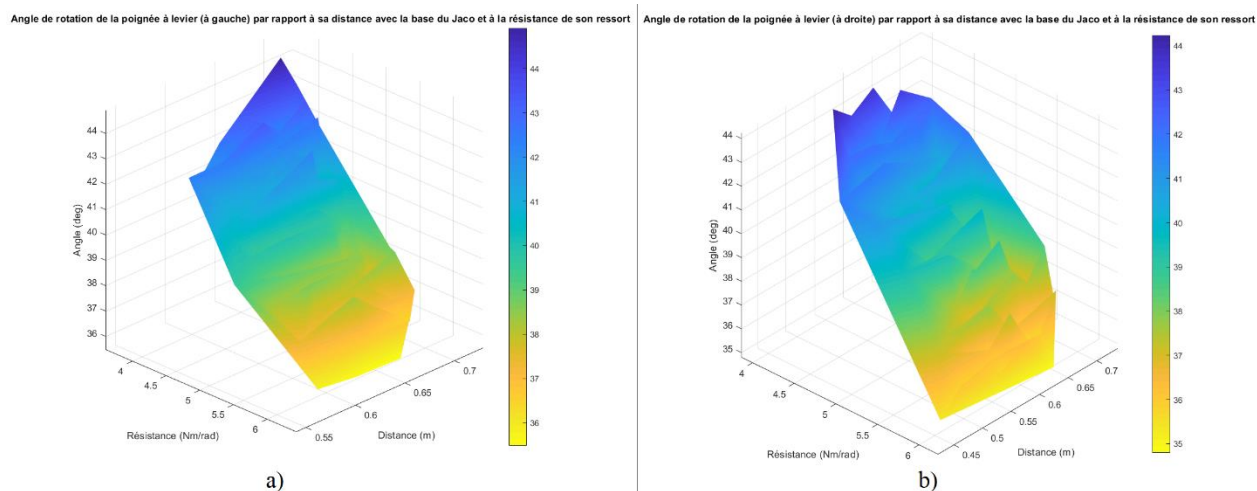


Figure 4.10 Angle exercé sur la poignée à levier en fonction de la résistance du ressort et de la distance entre celle-ci et la base du Jaco². a) et b) sont, respectivement, les cas étudiés lorsque la poignée se trouvait à gauche et à droite

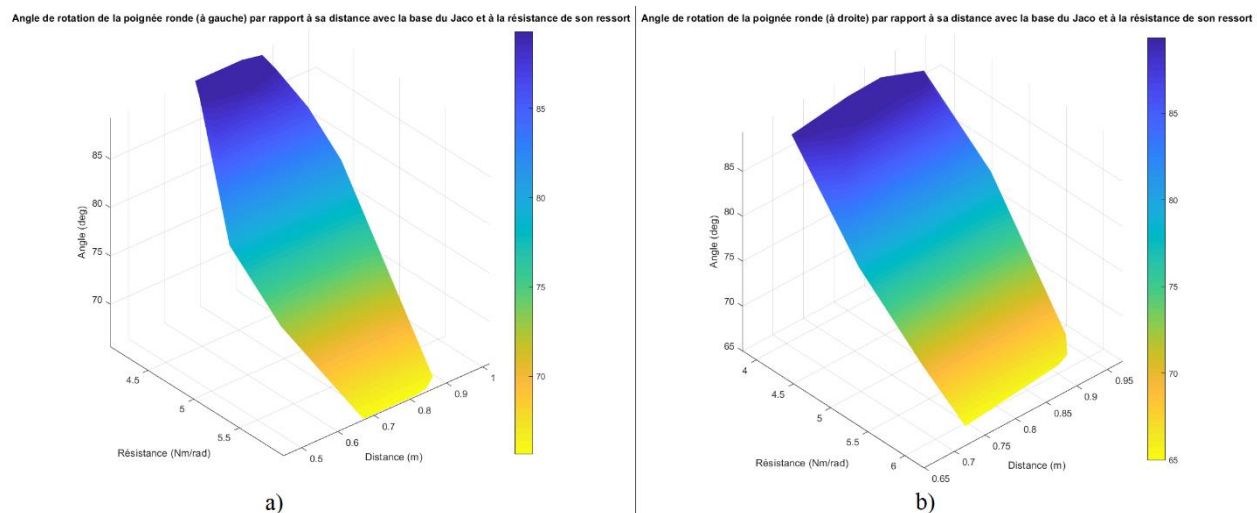


Figure 4.11 Angle exercé sur la poignée ronde en fonction de la résistance du ressort et de la distance entre celle-ci et la base du Jaco². a) et b) sont, respectivement, les cas étudiés lorsque la poignée se trouvait à gauche et à droite

4.2.2.2 Capacité du système d'opérer le dispositif d'ouverture

D'emblée, la première étape de cette tâche consiste à tourner la poignée pour permettre au loquet se déloger du cadre de porte. Toutefois, pour que le loquet soit en mesure de se rétracter du cadre, l'angle de rotation de la poignée doit être assez grand. Or, l'angle nécessaire varie de d'une poignée à l'autre, surtout selon la classe. En évaluant plusieurs poignées, il a été déterminé que l'angle

minimal d'ouverture pour une grande partie des poignées rondes et à levier tourne, respectivement, autour de 70° et de 35° . En comparant ces nombres avec la Figure 4.10 et la Figure 4.11, on remarque que le système est toujours en mesure d'opérer le dispositif d'ouverture, pour des résistances de ressort allant de 4.04 à 6.04 Nm/rad, sauf pour les poignées rondes possédant une résistance de ressort allant au-dessus de 5.6 Nm/rad.

La seconde étape consiste à ouvrir légèrement la porte et à ramener la poignée à son orientation de départ. Or, pendant que la poignée retourne à son orientation, il est important que l'effecteur ne referme pas la porte. En regardant le Tableau 4.10, on constate que pour les poignées à levier, cet objectif a été atteint à chaque essai. Toutefois, ceci n'est pas le cas pour les poignées rondes. Effectivement, sur les 200 essais, l'effecteur a refermé la porte 8 fois. Ce ratio est très bon, mais montre tout de même que, dans certaines conditions, le système a de la difficulté à accomplir la tâche. En analysant plus attentivement ces cas particuliers, on remarque que ceci se produisait uniquement lorsque la résistance du ressort était près du 6 Nm/rad, que la poignée se retrouvait loin de la base du manipulateur et que la porte devait être poussée. Effectivement, en réduisant légèrement la distance ou la résistance du ressort, le dispositif parvenait à nouveau à maintenir la porte ouverte. Ceci était également le cas en changeant la direction d'ouverture. Toutefois, avec cette résistance, le système ne devrait, de toute façon, pas être en mesure de tourner assez la poignée pour libérer la porte, tel que conclu précédemment.

Finalement, après avoir accompli la tâche, le système doit conclure si la porte doit être ouverte en poussant ou en tirant. Comme le montre le Tableau 4.10, celui-ci aboutit toujours à la bonne conclusion, et ce, peu importe la classe de poignées et sa position.

4.2.2.3 Erreurs angulaires de la poignée

Un autre problème pouvant empêcher la tâche d'être accomplie survient si l'effecteur ne parvient pas à maintenir l'angle de la poignée pendant qu'il ouvre la porte. Si l'angle réduit trop tôt, alors le loquet pourrait bloquer la porte avant que le manipulateur n'ait le temps de l'ouvrir. Il faudrait que ce problème survienne très tôt dans le mouvement pour bloquer la porte, mais toutes variations flagrantes présentent le risque que ça se produise. De plus, de grands changements d'angles lors du déplacement de la porte montreraient que le système manque de contrôle envers sa position.

Toutefois, ceci ne semble pas du tout être un problème pour les poignées rondes. Effectivement, comme le montre le Tableau 4.9, le pire changement d'angle de la poignée pendant l'ouverture de

la porte est de 0.9° . Si l'on considère que la partie sortante du loquet mesure 1.5 cm et que celui-ci est complètement rétracté dans la porte après une rotation de 70° , alors une déviation de 1° entraîne un déplacement du loquet de 0.2mm. Par contre, pour les poignées à levier, le risque est plus élevé. Dans le Tableau 4.8 on constate que l'erreur maximale d'un essai se rend en moyenne jusqu'à 4° , en prenant en compte l'écart type. Ceci n'est pas vraiment problématique étant donné que le déplacement du loquet qui s'ensuit est de seulement 2mm, si l'on considère que celui-ci est complètement à l'intérieur de la porte après une rotation de 35° . Toutefois, le déplacement maximal atteint lors des essais est de 11° , ce qui entraîne une translation du loquet de presque 5mm. Ce déplacement, si causé dès le début de l'ouverture de la porte, pourrait empêcher celle-ci de s'ouvrir. Par chance, comme le montre la Figure 4.12, cette situation est un cas isolé. De plus, ceci semble se produire lorsque la poignée est loin du manipulateur. Étant donné qu'un seul cas a été obtenu, il n'est pas possible de déduire de façon conclusive la cause de ce problème.

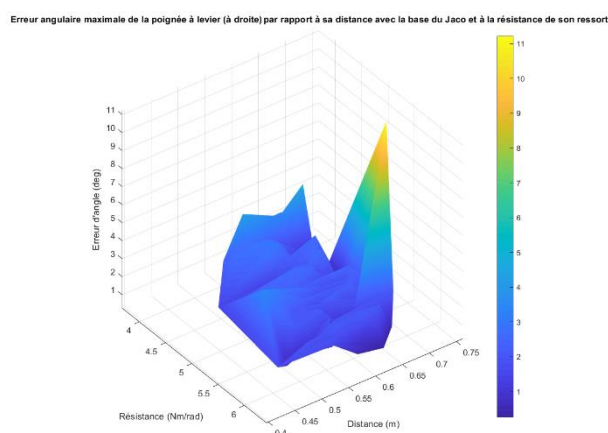


Figure 4.12 Erreur angulaire des poignées à levier situées à droite en fonction de la résistance du ressort et de la distance entre celle-ci et la base du Jaco2

4.2.3 Ouvrir la porte

4.2.3.1 Résultats

Le Tableau 4.11 et le Tableau 4.12 montrent le taux de réussite ainsi que le temps moyen et maximal pour chaque combinaison de conditions lorsque le fauteuil est respectivement situé à diverses positions devant la porte et complètement devant la porte. De plus, le Tableau 4.13 et le Tableau 4.14 montrent la moyenne des forces et couples moyens et maximaux ressentis par la poignée lorsque le fauteuil est situé à diverses positions devant la porte, et complètement devant la

porte, respectivement. Les vecteurs présentés dans ces tableaux représentent les forces et couples ressentis sur les axes x, y, z, où x est l'axe perpendiculaire au sol, y est l'axe sur le plan de la porte parallèle au sol, et z est l'axe normal à la porte.

Tableau 4.11 Taux de réussite ainsi que le temps nécessaire moyen et maximal obtenu pour ouvrir la porte lorsque le fauteuil se situe à diverses positions par rapport à la porte

Classe poignée	Direction porte	Position poignée	Réussi (/10)	Temps moyen (s)	Temps maximal (s)
À Levier	Pousser	Droite	10	10 ± 2	14
		Gauche	10	11 ± 3	16
	Tirer	Droite	10	14 ± 4	18
		Gauche	10	15 ± 2	19
Ronde	Pousser	Droite	10	16 ± 5	27
		Gauche	10	11 ± 2	14
	Tirer	Droite	10	12 ± 3	18
		Gauche	10	13 ± 2	17
À Tirer	Tirer	Droite	10	13 ± 2	17
		Gauche	10	13 ± 2	18

Tableau 4.12 Taux de réussite ainsi que le temps nécessaire moyen et maximal obtenu pour ouvrir la porte lorsque le fauteuil se situe complètement devant la porte

Classe poignée	Direction porte	Position poignée	Réussi (/10)	Temps moyen (s)	Temps maximal (s)
À Levier	Pousser	Droite	10	10 ± 2	16
		Gauche	10	12 ± 4	23
	Tirer	Droite	10	17 ± 4	23
		Gauche	10	12 ± 1	14
Ronde	Pousser	Droite	10	12 ± 2	16
		Gauche	10	15 ± 6	25
	Tirer	Droite	10	15 ± 5	24
		Gauche	10	15 ± 2	17
À Tirer	Tirer	Droite	10	19 ± 6	28
		Gauche	10	13 ± 6	24

Tableau 4.13 Forces et couples moyens ressentis par la poignée sur les 10 essais lorsque le fauteuil se situe à diverses positions par rapport à la porte *

Classe poignée	Direction porte	Position poignée	Moyenne des 10 essais			
			Force moyenne (N)	Force maximale (N)	Couple moyen (Nm)	Couple maximal (Nm)
À Levier	Pousser	Droite	$\begin{bmatrix} 1 \pm 2 \\ 1 \pm 1 \\ 0 \pm 1 \end{bmatrix} 10$	$\begin{bmatrix} 5 \pm 2 \\ 5 \pm 1 \\ 4 \pm 21 \end{bmatrix} 10^2$	$\begin{bmatrix} (17 \pm 6)10^{-4} \\ 1 \pm 2 \\ 2 \pm 5 \end{bmatrix}$	$\begin{bmatrix} (2 \pm 6)10^{-2} \\ (1 \pm 2)10^2 \\ (1 \pm 4)10^2 \end{bmatrix}$
		Gauche	$\begin{bmatrix} 0.6 \pm 0.4 \\ 2 \pm 2 \\ 1.0 \pm 0.9 \end{bmatrix}$	$\begin{bmatrix} 2 \pm 2 \\ 9 \pm 9 \\ 5 \pm 3 \end{bmatrix}$	$\begin{bmatrix} (18 \pm 6)10^{-4} \\ (2 \pm 2)10^{-1} \\ (3 \pm 3)10^{-1} \end{bmatrix}$	$\begin{bmatrix} (5 \pm 3)10^{-3} \\ (11 \pm 4)10^{-1} \\ (2 \pm 2) \end{bmatrix}$
	Tirer	Droite	$\begin{bmatrix} 2 \pm 3 \\ 2 \pm 2 \\ 4 \pm 8 \end{bmatrix}$	$\begin{bmatrix} 1 \pm 2 \\ 2 \pm 3 \\ 3 \pm 5 \end{bmatrix} 10$	$\begin{bmatrix} (17 \pm 5)10^{-4} \\ (5 \pm 9)10^{-1} \\ (3 \pm 6)10^{-1} \end{bmatrix}$	$\begin{bmatrix} (5 \pm 2)10^{-3} \\ 4 \pm 7 \\ 4 \pm 7 \end{bmatrix}$
		Gauche	$\begin{bmatrix} 0.6 \pm 0.6 \\ 0.7 \pm 0.3 \\ 0.9 \pm 0.5 \end{bmatrix}$	$\begin{bmatrix} 2 \pm 1 \\ 3 \pm 2 \\ 10 \pm 9 \end{bmatrix}$	$\begin{bmatrix} (17 \pm 4)10^{-4} \\ (2 \pm 2)10^{-1} \\ (8 \pm 7)10^{-2} \end{bmatrix}$	$\begin{bmatrix} (5 \pm 2)10^{-3} \\ (13 \pm 8)10^{-1} \\ (6 \pm 4)10^{-1} \end{bmatrix}$
Ronde	Pousser	Droite	$\begin{bmatrix} 1 \pm 3 \\ 0 \pm 1 \\ 1 \pm 1 \end{bmatrix} 10$	$\begin{bmatrix} 0 \pm 10 \\ 3 \pm 6 \\ 2 \pm 5 \end{bmatrix} 10^2$	$\begin{bmatrix} 1 \pm 2 \\ 2 \pm 6 \\ (1 \pm 1)10^{-3} \end{bmatrix}$	$\begin{bmatrix} (1 \pm 1)10^2 \\ (1 \pm 2)10^2 \\ (2 \pm 3)10^{-2} \end{bmatrix}$
		Gauche	$\begin{bmatrix} (1 \pm 3)10 \\ (1 \pm 3)10 \\ (3 \pm 6) \end{bmatrix}$	$\begin{bmatrix} (4 \pm 1)10^3 \\ (6 \pm 1)10^3 \\ (10 \pm 2)10^2 \end{bmatrix}$	$\begin{bmatrix} 3 \pm 6 \\ 2 \pm 6 \\ (9 \pm 2)10^{-4} \end{bmatrix}$	$\begin{bmatrix} (1 \pm 3)10^2 \\ (1 \pm 2)10^2 \\ (1 \pm 1)10^{-2} \end{bmatrix}$
	Tirer	Droite	$\begin{bmatrix} 0.8 \pm 0.4 \\ 0.8 \pm 0.2 \\ 0.5 \pm 0.2 \end{bmatrix}$	$\begin{bmatrix} 2 \pm 2 \\ 2 \pm 2 \\ 10 \pm 10 \end{bmatrix}$	$\begin{bmatrix} (8 \pm 6)10^{-2} \\ (2 \pm 1)10^{-1} \\ (8 \pm 5)10^{-4} \end{bmatrix}$	$\begin{bmatrix} (6 \pm 6)10^{-1} \\ (6 \pm 3)10^{-1} \\ (4 \pm 3)10^{-3} \end{bmatrix}$
		Gauche	$\begin{bmatrix} 5 \pm 4 \\ 6 \pm 5 \\ 8 \pm 5 \end{bmatrix}$	$\begin{bmatrix} 3 \pm 3 \\ 3 \pm 3 \\ 5 \pm 2 \end{bmatrix} 10^2$	$\begin{bmatrix} 1 \pm 1 \\ (10 \pm 9)10^{-1} \\ (10 \pm 4)10^{-4} \end{bmatrix}$	$\begin{bmatrix} (1 \pm 6)10^2 \\ (1 \pm 6)10^2 \\ (3 \pm 1)10^{-2} \end{bmatrix}$
À Tirer	Tirer	Droite	$\begin{bmatrix} (1 \pm 2)10 \\ 2 \pm 3 \\ 2 \pm 5 \end{bmatrix}$	$\begin{bmatrix} (0 \pm 1)10^3 \\ (1 \pm 2)10^2 \\ (1 \pm 3)10^2 \end{bmatrix}$	$\begin{bmatrix} 0 \pm 1 \\ (3 \pm 3)10^{-3} \\ 1 \pm 4 \end{bmatrix}$	$\begin{bmatrix} (2 \pm 6)10 \\ (6 \pm 3)10^{-3} \\ (7 \pm 2)10^2 \end{bmatrix}$
		Gauche	$\begin{bmatrix} (9 \pm 2)10 \\ 2 \pm 2 \\ (7 \pm 2)10 \end{bmatrix}$	$\begin{bmatrix} (0 \pm 1)10^3 \\ (8 \pm 9)10 \\ (3 \pm 7)10^2 \end{bmatrix}$	$\begin{bmatrix} 2 \pm 3 \\ (3 \pm 2)10^{-3} \\ 2 \pm 5 \end{bmatrix}$	$\begin{bmatrix} (1 \pm 1)10^2 \\ (6 \pm 5)10^{-3} \\ (1 \pm 2)10^2 \end{bmatrix}$

*La résolution des variables est en fonction de l'écart type de l'axe.

Tableau 4.14 Forces et couples moyens ressentis par la poignée sur les 10 essais lorsque le fauteuil se situe complètement devant la porte *

Classe poignée	Direction porte	Position poignée	Moyenne des 10 essais			
			Force moyenne (N)	Force maximale (N)	Couple moyen (Nm)	Couple maximal (Nm)
À Levier	Pousser	Droite	$\begin{bmatrix} 0.9 \pm 0.5 \\ 0.9 \pm 0.5 \\ 1.0 \pm 0.7 \end{bmatrix}$	$\begin{bmatrix} 4 \pm 3 \\ 4 \pm 3 \\ 5 \pm 3 \end{bmatrix}$	$\begin{bmatrix} (16 \pm 4)10^{-4} \\ (2 \pm 2)10^{-1} \\ (2 \pm 1)10^{-1} \end{bmatrix}$	$\begin{bmatrix} (6 \pm 3)10^{-3} \\ (14 \pm 9)10^{-1} \\ 1 \pm 1 \end{bmatrix}$
		Gauche	$\begin{bmatrix} 0.8 \pm 0.4 \\ 2 \pm 3 \\ 1 \pm 1 \end{bmatrix}$	$\begin{bmatrix} 2 \pm 2 \\ 8 \pm 8 \\ 5 \pm 2 \end{bmatrix}$	$\begin{bmatrix} (16 \pm 7)10^{-4} \\ (1 \pm 2)10^{-1} \\ (3 \pm 5)10^{-1} \end{bmatrix}$	$\begin{bmatrix} (4 \pm 2)10^{-3} \\ (11 \pm 4)10^{-1} \\ (1 \pm 2) \end{bmatrix}$
	Tirer	Droite	$\begin{bmatrix} 0.5 \pm 0.4 \\ 0.5 \pm 0.1 \\ 0.5 \pm 0.2 \end{bmatrix}$	$\begin{bmatrix} 2 \pm 1 \\ 2 \pm 2 \\ 3 \pm 2 \end{bmatrix}$	$\begin{bmatrix} (12 \pm 2)10^{-4} \\ (1 \pm 1)10^{-1} \\ (6 \pm 3)10^{-2} \end{bmatrix}$	$\begin{bmatrix} (39 \pm 9)10^{-4} \\ (6 \pm 7)10^{-1} \\ (4 \pm 4)10^{-1} \end{bmatrix}$
		Gauche	$\begin{bmatrix} 0.9 \pm 0.5 \\ 0.7 \pm 0.3 \\ 0.5 \pm 0.1 \end{bmatrix}$	$\begin{bmatrix} 2 \pm 2 \\ 5 \pm 4 \\ (1 \pm 1)10 \end{bmatrix}$	$\begin{bmatrix} (18 \pm 4)10^{-4} \\ (10 \pm 8)10^{-2} \\ (7 \pm 5)10^{-2} \end{bmatrix}$	$\begin{bmatrix} (5 \pm 2)10^{-3} \\ (2 \pm 3) \\ (10 \pm 8)10^{-1} \end{bmatrix}$
Ronde	Pousser	Droite	$\begin{bmatrix} 1 \pm 2 \\ 0 \pm 1 \\ 0 \pm 1 \end{bmatrix} 10$	$\begin{bmatrix} (0 \pm 1)10^3 \\ (3 \pm 8)10^2 \\ (3 \pm 9)10^2 \end{bmatrix}$	$\begin{bmatrix} 1 \pm 2 \\ 1 \pm 3 \\ (16 \pm 8)10^{-4} \end{bmatrix}$	$\begin{bmatrix} (1 \pm 2)10^2 \\ (1 \pm 3)10^2 \\ (2 \pm 5)10^{-2} \end{bmatrix}$
		Gauche	$\begin{bmatrix} (3 \pm 5)10 \\ (3 \pm 4)10 \\ (5 \pm 8) \end{bmatrix}$	$\begin{bmatrix} (1 \pm 1)10^3 \\ (1 \pm 2)10^3 \\ (2 \pm 4)10^2 \end{bmatrix}$	$\begin{bmatrix} 5 \pm 9 \\ (1 \pm 1)10 \\ (13 \pm 9)10^{-4} \end{bmatrix}$	$\begin{bmatrix} (2 \pm 4)10^2 \\ (1 \pm 3)10^2 \\ (1 \pm 2)10^{-2} \end{bmatrix}$
	Tirer	Droite	$\begin{bmatrix} 0.7 \pm 0.5 \\ 0.7 \pm 0.1 \\ 0.4 \pm 0.1 \end{bmatrix}$	$\begin{bmatrix} 2 \pm 2 \\ 2.1 \pm 0.4 \\ 2 \pm 3 \end{bmatrix}$	$\begin{bmatrix} (6 \pm 4)10^{-2} \\ (15 \pm 7)10^{-2} \\ (6 \pm 3)10^{-4} \end{bmatrix}$	$\begin{bmatrix} (4 \pm 8)10^{-1} \\ 1 \pm 1 \\ (3 \pm 4)10^{-3} \end{bmatrix}$
		Gauche	$\begin{bmatrix} 4 \pm 5 \\ 8 \pm 9 \\ 10 \pm 9 \end{bmatrix}$	$\begin{bmatrix} 3 \pm 3 \\ 6 \pm 8 \\ 7 \pm 6 \end{bmatrix} 10^2$	$\begin{bmatrix} 2 \pm 2 \\ 1 \pm 2 \\ (11 \pm 4)10^{-4} \end{bmatrix}$	$\begin{bmatrix} (2 \pm 2)10^2 \\ (1 \pm 1)10^2 \\ (3 \pm 3)10^{-2} \end{bmatrix}$
À Tirer	Tirer	Droite	$\begin{bmatrix} (1 \pm 3)10 \\ 1 \pm 2 \\ 3 \pm 9 \end{bmatrix}$	$\begin{bmatrix} (1 \pm 2)10^3 \\ (0 \pm 1)10^2 \\ (2 \pm 7)10^2 \end{bmatrix}$	$\begin{bmatrix} 1 \pm 2 \\ (3 \pm 2)10^{-3} \\ 2 \pm 5 \end{bmatrix}$	$\begin{bmatrix} (1 \pm 2)10^2 \\ (6 \pm 3)10^{-3} \\ (1 \pm 4)10^2 \end{bmatrix}$
		Gauche	$\begin{bmatrix} 2 \pm 4 \\ (1 \pm 3)10 \\ (1 \pm 2)10 \end{bmatrix}$	$\begin{bmatrix} (1 \pm 3)10^2 \\ (1 \pm 2)10^3 \\ (1 \pm 1)10^3 \end{bmatrix}$	$\begin{bmatrix} (1 \pm 2)10 \\ (5 \pm 3)10^{-3} \\ 2 \pm 6 \end{bmatrix}$	$\begin{bmatrix} (1 \pm 1)10^3 \\ (15 \pm 5)10^{-3} \\ (2 \pm 7)10^2 \end{bmatrix}$

*La résolution des variables est en fonction de l'écart type de l'axe.

4.2.3.2 Taux de réussite et temps

Comme le montrent le Tableau 4.11 et le Tableau 4.12, l'utilisateur est parvenu à ouvrir la porte à l'intérieur des 30 secondes lors des 200 essais, peu importe la classe et la position de la poignée et peu importe si la porte devait être tirée ou poussée. Toutefois, bien qu'en moyenne l'utilisateur parvenait à accomplir la tâche à l'intérieur des 20 secondes, les temps maximaux montrent que certains essais sont passés près d'atteindre les 30 secondes. Certaines expérimentations sont donc passées près d'être considérées comme non-accomplies.

4.2.3.3 Forces et couples ressentis

Lorsque le fauteuil était positionné à divers endroits en face de la porte, il est possible de constater, à l'aide du Tableau 4.13, que les forces et les couples ressentis par la poignée sont généralement beaucoup trop grands pour la tâche. En effet, en moyenne, la majorité des essais génèrent au moins une force dans les centaines de newtons. Ceci est nettement trop grand pour permettre à un effecteur de maintenir la poignée, même après avoir modifié celui-ci. En ce qui concerne les couples, ceux-ci sont également trop grands dans l'ensemble. De ce fait, même si en simulation, la tâche d'ouvrir la porte se fait facilement, le Tableau 4.13 montre qu'en pratique, l'effecteur aurait généralement de la difficulté à maintenir sa prise sur la poignée. Effectivement, dans l'ensemble des essais, seulement 3 combinaisons de conditions semblent permettre au manipulateur de ne pas générer des forces et des couples trop élevés pour un effecteur :

- Lorsque la poignée est à levier, positionnée à gauche et doit être poussée;
- Lorsque la poignée est à levier, positionnée à gauche et doit être tirée;
- Lorsque la poignée est ronde, positionnée à droite et doit être tirée.

Toutefois, si l'utilisateur positionne son fauteuil complètement devant la porte, alors on remarque que le nombre de conditions fonctionnelles augmente. Effectivement, comme le montre le Tableau 4.14, la majorité des situations présentent des résultats similaires, toutefois, toutes les situations impliquant une poignée à levier génèrent maintenant des forces et des couples relativement faibles. De ce fait, avec un bon effecteur, le système devrait être en mesure d'ouvrir toutes les portes possédant une poignée à levier ou bien les portes possédant une poignée ronde nécessitant d'être tirée devant lesquelles le manipulateur est directement positionné.

En réalisant plusieurs expérimentations, il a été conclu que cette problématique n'était pas directement causée par le système de contrôle. Ceci semble plutôt être causé par la structure du manipulateur et par le fait que le fauteuil soit contrôlé par un utilisateur. En effet, l'utilisateur ne peut pas suivre une trajectoire spécifique et, dû à la structure du Jaco², il semble être facile lors d'une trajectoire de lui imposer un mouvement, à l'aide du fauteuil, qu'il n'est pas en mesure d'accomplir, générant ainsi des forces et des couples sur la poignée. Ceci est aussi remarquable dans le Tableau 4.13 et le Tableau 4.14. En regardant les vecteurs de couples, on constate qu'il y en a toujours un vraiment plus faible que les autres. Toutefois, celui-ci semble changer d'axe selon la classe de la poignée. En fait, il correspond à l'axe du dernier actuateur et donc ne change pas selon la classe de la poignée, mais plutôt selon l'orientation de l'effecteur. Grâce à cet actuateur, le fauteuil n'est pas en mesure de générer de grands couples sur cet axe étant donné que l'actuateur se laisse guider par ces mouvements. Malheureusement, sur les autres axes, aucun actuateur n'est positionné pour empêcher cette génération de force.

Cet actuateur est également la raison pour laquelle la tâche d'ouverture de la porte semble être plus facile avec les poignées à levier. Effectivement, lorsque l'effecteur est orienté comme à la Figure 3.12 a), l'axe de l'actuateur correspond au vecteur normal du sol. C'est une grosse différence par rapport aux autres orientations puisque tous les mouvements du fauteuil et de la porte sont réalisés de façon parallèle au sol. De ce fait, tous ces mouvements peuvent entraîner des couples perpendiculaires du sol. Ces couples peuvent générer, quant à eux, plusieurs forces. Lorsque l'effecteur possède cette orientation, les couples générés par ces mouvements entraînent une rotation du poignet au lieu de générer des efforts sur la poignée. Dans cette situation, il est donc nettement plus difficile pour l'utilisateur de produire de grandes forces/couples sur la poignée. Lorsque le fauteuil n'est pas bien positionné devant la porte, l'utilisateur doit réaliser plus de manœuvres pour ouvrir la porte et se repositionner. Ceci entraîne donc des situations où le manipulateur tire ou pousse la poignée dans des directions non-idéales générant ainsi de grandes forces, comme le montre le Tableau 4.13.

CHAPITRE 5 DISCUSSION

5.1 Détection et classification de la poignée de porte

5.1.1 Détection de la poignée

5.1.1.1 Détection par Faster R-CNN

Les résultats ont démontré que le système est tout de même efficace pour détecter les poignées rondes et les poignées à levier. Toutefois, ce n'est pas le cas pour les poignées à tirer. Ceci semble être principalement la cause de l'algorithme Faster R-CNN. Effectivement, celui-ci a eu énormément de difficulté à détecter des poignées à tirer. De plus, même avec les autres types de poignées, celui-ci ne priorisait pas toujours la bonne détection. Ceci peut être problématique pour le système de détection. En effet, même si l'objectif était de filtrer les faux positifs à l'aide de l'information tridimensionnelle, avoir autant de mauvaises détections prioritaires augmente les chances du système de réaliser une fausse détection finale. Par exemple, si la fausse détection, résultante de l'algorithme Faster R-CNN, est également devant le plan de la porte, alors l'information tridimensionnelle ne permettra pas au système d'éliminer ce choix.

L'algorithme Faster R-CNN entraîné pour le système n'est donc pas idéal. Ceci peut s'expliquer avec la base de données utilisée lors de l'entraînement. Il est important de mentionner que sur les 1213 images contenues dans celle-ci, 410 d'entre elles ne possèdent pas de poignée. De plus, sur les 803 images restantes, seulement 280 d'entre elles comprennent des poignées de porte. Effectivement, la base de données a été développée pour permettre également la reconnaissance de poignées d'armoire et de poignées de réfrigérateur. De ce fait, la quantité d'images de poignée de porte disponibles pour l'entraînement est relativement faible. De plus, sur les 280 images, la majorité des poignées sont rondes ou à levier, expliquant pourquoi les poignées à tirer sont plus difficiles à détecter. Étant donné que les poignées d'armoire et de réfrigérateur peuvent être similaires à certaines poignées de portes, celles-ci ont tout de même été conservées pour améliorer la qualité de la détection. Toutefois, ils entraînent également un autre problème. Lorsque l'on compare certaines poignées d'armoires de la base de données avec les fausses détections générées par l'algorithme Faster R-CNN, il est plus facile de comprendre pourquoi celui-ci prend parfois de mauvaises décisions. En effet, plusieurs fausses détections concernent les charnières de porte,

comme à la Figure 4.2, à la Figure 4.4 a) et à la Figure 4.5. Si l'on compare ces charnières à plusieurs poignées d'armoires de la base de données, comme à la Figure 5.1, on remarque que celles-ci sont visuellement similaires.



Figure 5.1 Exemple d'image d'armoire de la base de données utilisée

Une solution à ce problème serait donc d'enrichir la base de données utilisée pour l'entraînement. En effet, en augmentant le nombre d'images de poignées de porte à tirer, l'algorithme devrait avoir plus de facilité à les reconnaître. De plus, en lui ajoutant également des images de charnières et d'autres objets générant actuellement de faux positifs, celui-ci devrait éventuellement avoir plus de facilité à les différencier.

5.1.1.2 Détection avec l'information tridimensionnelle

Malgré ces erreurs de détection, les résultats démontrent que l'information tridimensionnelle est un outil fort efficace pour filtrer ces mauvaises détections et ainsi permettre au système d'avoir de meilleurs résultats. Effectivement, sur les 12 expérimentations où l'algorithme Faster R-CNN était parvenu à détecter la poignée, mais avait priorisé une autre détection, 11 d'entre elles ont été réussies grâce à l'information tridimensionnelle. De plus, grâce à l'information tridimensionnelle, sur les 5 expérimentations où l'algorithme Faster R-CNN n'a simplement pas reconnu la poignée, 4 ont conclu qu'il n'y en avait aucune, au lieu de générer de fausses détections. Finalement, en aucun cas l'information tridimensionnelle n'a entraîné, à elle seule, une mauvaise détection. Ceci a même été expérimenté avec des portes possédant une grande superficie de vitre pour voir si le plan de la porte serait faussement déterminé. Toutefois, le plan a tout de même été obtenu avec

succès à l'aide du tour de la porte, et la détection a été réussie sans problème. De ce fait, il est évident que de combiner les deux méthodes de détection est très avantageux pour le système.



Figure 5.2 Poignée à levier 3

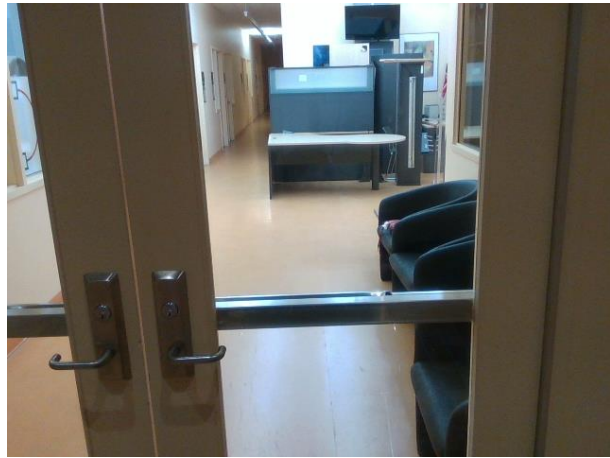


Figure 5.3 Poignée à levier 9

Par contre, les résultats ont également montré que le masque généré par l'information tridimensionnelle n'est pas idéal. C'est surtout le cas pour les poignées à levier et les poignées à tirer. Effectivement, leurs masques est souvent incomplet, ce qui peut entraîner, dans certains cas, des problèmes pour déterminer la position cartésienne de celles-ci. De ce fait, il serait préférable d'obtenir un masque mieux défini de la poignée pour assurer l'obtention d'une bonne position cartésienne. Une solution serait peut-être d'utiliser Mask R-CNN au lieu de Faster R-CNN, ou bien de travailler avec la caméra pour voir si modifier ses paramètres ou la changer entraînerait des résultats plus prometteurs.

5.1.2 Classification de la poignée

En ce qui concerne la classification des poignées, lorsque la détection était réussie, les résultats n'ont démontré aucune erreur de classification. De ce fait, il est clair que d'utiliser un descripteur HOG pour déterminer la classe des poignées est très efficace.

5.2 Systèmes de contrôle

5.2.1 Atteindre le dispositif d'ouverture

Les résultats concernant le système de contrôle mis en place pour atteindre la poignée de porte ont démontré que celui-ci, ainsi que la trajectoire développée pour cette tâche, sont très efficaces, peu importe la position et la classe de la poignée. Effectivement, les erreurs de position et d'orientation finales de l'effecteur sont assez faibles pour ne pas risquer que celui-ci manque la poignée. De plus, les actionneurs suivent relativement bien la trajectoire tout le long du trajet. Le seul moment où les erreurs angulaires des actionneurs sont un peu plus grandes est au tout début de la trajectoire. Toutefois, ces erreurs ne sont pas en mesure de causer des problèmes à la tâche et sont beaucoup trop faibles pour causer un risque de sécurité. Ces résultats démontrent également que le système s'adapte bien aux changements de paramètres de la dynamique. De ce fait, si la dynamique du Jaco² varie légèrement, alors le système de contrôle devrait tout de même être en mesure d'accomplir la tâche.

Bien entendu, la précision du manipulateur réelle n'a pas été prise en compte dans ce travail, il est normal de s'attendre à ce que la précision réelle soit un peu moindres. Malgré cela, ces résultats démontrent tout de même que le système est en mesure de s'adapter à de faibles changements de la dynamique, qu'il est en mesure d'atteindre la poignée peu importe sa position, si elle est accessible, et qu'il est en mesure de fonctionner avec le système de communication de haut niveau.

5.2.2 Opérer le dispositif d'ouverture

En ce qui concerne le système de contrôle mis en place pour opérer les poignées de porte, les résultats démontrent qu'il est en mesure d'opérer les poignées à levier possédant une constante de ressort entre 4,04 et 6,04 Nm/rad, est en mesure de s'adapter à la position de la poignée et aux différences de paramètres de la dynamique, et est en mesure de fonctionner avec le système de communication de haut niveau. Pour les poignées rondes, les résultats sont similaires, mais la constante de ressort ne doit toutefois pas dépasser 5.6 Nm/rad. Ce dernier critère est causé par une limitation physique du manipulateur et non du système de contrôle lui-même. De ce fait, le système mis en place a démontré qu'il est efficace pour permettre à l'effecteur d'opérer les poignées de porte.

Toutefois, lors d'une expérimentation avec les poignées à levier, le système a eu de la difficulté à maintenir l'angle de la poignée pendant l'ouverture de la porte. Cette situation ne semblait pas nuire à la tâche, bien qu'il serait préférable que l'effecteur soit en mesure de maintenir l'angle de la poignée. Étant donné que ceci s'est produit lorsque la poignée était très loin de la base du Jaco², il serait donc probablement préférable que l'utilisateur se positionne de manière à éviter une pleine extension du manipulateur.

5.2.3 Ouvrir la porte

En ce qui concerne l'ouverture de la porte, les résultats sont un peu plus problématiques. Malgré le fait qu'en simulation, la tâche d'ouvrir la porte ait été réalisée à chaque essai sans problème en moins de 30 secondes, les forces et les couples ressentis par la poignée démontrent que dans bien des cas, l'utilisateur pourrait entraîner l'effecteur à lâcher la poignée. Effectivement, seule l'orientation de l'effecteur utilisée pour les poignées à levier semble ne pas générer de grandes forces et couples. Toutefois, il est important que l'utilisateur positionne son fauteuil complètement en face de la porte pour lui éviter de faire des manœuvres l'obligeant à tirer ou à pousser la poignée dans la mauvaise direction. En ce qui concerne les autres poignées, seules les poignées rondes, situées en face du manipulateur, et qui nécessitaient d'être tirées ne génèrent pas de grandes forces. Toutefois, à part cette situation, il est fort probable que l'effecteur actuelle glisse de la poignée lors de la tâche.

5.2.4 Retour sur l'ensemble des systèmes de contrôle

En regardant l'ensemble des sous-tâches accomplies par le Jaco², on remarque que si l'on suit chaque étape prévue dans le cadre de ce travail, le système complet est uniquement en mesure d'ouvrir des portes possédant des poignées à levier. Effectivement, malgré qu'il soit possible d'atteindre et d'opérer toutes les poignées, la méthode d'ouverture développée fonctionne uniquement pour les poignées à levier. Toutefois, assister les utilisateurs lors de l'ouverture des portes pour l'ensemble des poignées est tout de même possible avec les sous-systèmes développés. Les résultats démontrent simplement que les consignes assignées à l'utilisateur seront un peu plus grandes. Actuellement, les seules consignes pour l'utilisateur sont de positionner son fauteuil complètement devant la porte et le plus perpendiculaire possible à la porte. Or ces consignes ne permettent pas d'ouvrir les portes possédant des poignées rondes ou des poignées à tirer.

Premièrement, les résultats ont démontré qu'il était possible pour les utilisateurs d'ouvrir des portes à poignée ronde nécessitant d'être tirées lorsque la poignée était en face du manipulateur. Si le fauteuil se positionne devant la porte et que la poignée est à gauche, et que le manipulateur est à droite, alors il est fort probable que l'effecteur glisse de la poignée durant la tâche. Dans cette condition, il est préférable pour l'utilisateur de toujours positionner le manipulateur Jaco² en face de la poignée au lieu d'essayer de positionner son fauteuil complètement devant la porte. Toutefois, ceci est uniquement possible si l'espace à côté de la porte est disponible.

Deuxièmement, toujours devant une porte à poignée ronde, mais qu'elle doit être poussée, alors l'utilisateur pourra positionner son fauteuil complètement devant la porte, mais devra reprendre possession des commandes de l'effecteur après le deuxième système de contrôle. Étant donné qu'avec les deux premiers systèmes de contrôle développés, la porte se fait ouvrir légèrement sans problème, l'utilisateur peut ensuite simplement pousser la porte avec le manipulateur sans nécessiter que l'effecteur maintienne la poignée. Effectivement, la grande complexité de la tâche est d'opérer la poignée et d'ouvrir une porte qui nécessite d'être tirée. Il n'est pas compliqué pour l'utilisateur de pousser la porte si celle-ci est légèrement ouverte. Cette solution pourrait également être utilisée pour toutes les portes à pousser, mais le troisième système de contrôle permet tout de même une assistance plus complète.

Finalement, pour les poignées à tirer, l'utilisateur devra se positionner pour que le manipulateur se trouve devant la poignée, comme pour les poignées rondes nécessitant d'être tirées. Toutefois, le code développé pour ouvrir les poignées à tirer ne pourra pas être utilisé. Effectivement, les résultats ont démontré que l'orientation utilisée pour les poignées à tirer n'est pas fonctionnelle pour ouvrir la porte. Toutefois, il est possible de maintenir les poignées à tirer avec la même orientation de l'effecteur que pour les poignées rondes, en tournant simplement le poignet de 90°. De ce fait, tout ce qui a été développé pour les poignées rondes, à l'exception du deuxième système de contrôle, peut être utilisé pour les poignées à tirer. Lorsque la caméra détectera donc une poignée à tirer, le premier et troisième système de contrôle utilisés seront ceux de la poignée ronde, avec une rotation supplémentaire du poignet de 90°.

5.3 Limitations et modifications

Sous la forme actuelle, le système possède plusieurs limitations concernant autant le système de vision que le système de contrôle. Cette section présentera donc les limites du système et les modifications qui pourraient être apportées pour les modifier.

5.3.1 Détection et classification de la poignée de porte

Le système a seulement été développé pour être en mesure de détecter trois classes de poignées : les poignées rondes, à levier, et à tirer. De ce fait, toutes poignées ne faisant pas partie de ces classes sont inutilisables avec le système. Toutefois, comme présenté à la section 5.1, le système rencontre tout de même certaines difficultés à détecter les poignées à tirer. Pour cette raison, il est préférable d'utiliser le système actuel uniquement pour les poignées rondes et à levier. Une solution serait d'augmenter la base de données permettant d'entraîner le système de détection. Celle-ci pourrait être augmentée pour améliorer la détection des trois classes de poignées et pour y ajouter d'autres classes. La base de données pourrait également être augmentée avec des images représentant mieux les situations qui seront observées par la caméra D415.

Une autre limite concerne les portes complètement en vitre. Effectivement, les portes complètement en vitre ne peuvent pas être repérées par la caméra stéréovision et il ne sera donc pas possible de déterminer le plan de la porte pour en retirer la poignée. De ce fait, dans cette situation, la détection de la poignée devrait se réaliser tout de même, mais le masque généré ne sera pas adéquat. Pour cette raison, l'information tridimensionnelle ne devrait pas être le principal outil pour déterminer le masque de la poignée. L'utilisation de Mask R-CNN pourrait être une bonne alternative, par exemple.

5.3.2 Systèmes de contrôle

Pour les systèmes de contrôle, plusieurs limites concernent la position du fauteuil. Par exemple, l'utilisateur doit absolument positionner son fauteuil de façon perpendiculaire à la porte. Le système ne peut donc pas fonctionner de côté. De plus, comme expliqué dans la section 5.2, l'utilisateur doit positionner son fauteuil par rapport à la poignée ou la porte selon le type de poignée à ouvrir et selon la direction d'ouverture de la porte. Cette restriction est causée par le troisième système de contrôle. Afin de réduire le nombres de consignes à respecter par l'utilisateur,

une solution serait de changer de manipulateur. Effectivement, les résultats ont démontré que d'avoir un actuateur au poignet dont l'axe correspond au vecteur normal du sol est idéal pour accomplir la tâche d'ouverture de porte. De ce fait, le Jaco² pourrait être remplacé par un autre manipulateur à 6 DDL possédant un poignet sphérique. À l'aide d'un poignet sphérique, les orientations initiales utilisées pour agripper les trois classes de poignées posséderaient toutes un actuateur au poignet dont l'axe correspondrait au vecteur normal du sol. L'ouverture de la porte générerait donc des forces et des couples moins grands sur la poignée. Ceci faciliterait donc l'utilisation du troisième système de contrôle, permettant d'être moins restrictif sur la position du fauteuil. Ceci pourrait même permettre un fonctionnement complet du troisième système de contrôle pour les trois classes de poignées, mais ceci reste à vérifier.

Finalement, le système est uniquement en mesure d'ouvrir des poignées à levier et rondes dont la constante de ressort est maximum 6.04 Nm/rad et 5.6 Nm/rad, respectivement. De plus, le système n'a pas été expérimenté avec des portes possédant des systèmes de fermeture automatique. De ce fait, il ne peut pas être conseillé pour ce genre de situation.

5.4 Retour sur les objectifs

L'objectif principal était de développer un système intelligent d'assistance à l'ouverture des portes pour les utilisateurs en fauteuil roulant en asservissant un manipulateur Jaco² 6 DDL fixé sur le fauteuil. Pour y parvenir, plusieurs sous-objectifs ont été déterminés.

Le premier est le développement d'un système stéréovision capable de déterminer la nature et la position des poignées. Cet objectif a été réalisé, mais pas pour les trois types de poignée recherchés. Effectivement, le système de détection possède beaucoup de difficulté avec les poignées à tirer. Toutefois, le système est efficace pour les poignées rondes et les poignées à levier. Ensuite, un système de contrôle de position devait être développé pour atteindre la poignée, et un autre pour l'opérer. Ces deux objectifs ont été atteints pour les trois classes de poignées. Finalement, le dernier système de contrôle à développer devait permettre au manipulateur Jaco² de maintenir son contact envers la poignée tout en s'adaptant au déplacement du fauteuil roulant lorsque l'utilisateur se déplace pour ouvrir la porte. Ceci a été atteint avec les poignées à levier, mais pas pour les poignées rondes et les poignées à tirer. Pour les poignées rondes, bien que le système fonctionne avec les portes à tirer si l'utilisateur positionne bien son fauteuil, ce système de contrôle n'est pas efficace

avec les portes à pousser. Pour les poignées à tirer, le système fonctionne, mais nécessite d'utiliser à la place celui développé pour les poignées rondes.

De ce fait, on pourrait conclure que l'objectif principal a été atteint, mais pas pour l'ensemble des poignées. Effectivement, avec les mauvaises détections, le système n'est pas en mesure d'assister l'utilisateur pour ouvrir les poignées à tirer. De plus, pour les poignées rondes, le système peut apporter une assistance complète pour les portes à tirer, mais ne parvient pas à assister l'utilisateur pour ouvrir complètement les portes à pousser.

CHAPITRE 6 CONCLUSION ET RECOMMANDATIONS

Ce travail de recherche avait pour but de simplifier la tâche ouverture de portes pour les utilisateurs en fauteuil roulant utilisant un MSA. Pour y parvenir, ce projet consistait à développer un système intelligent d'assistance à l'ouverture de portes en asservissant un manipulateur Jaco² 6 DDL fixé sur le fauteuil. De façon plus précise, le système devait être en mesure de détecter et classer les poignées de porte et de contrôler le manipulateur pour qu'il atteigne la poignée, l'opère, et ouvre la porte en s'adaptant aux mouvements du fauteuil contrôlé par l'utilisateur. Ce type de système semble ne jamais avoir été développé avant. Effectivement, les études précédentes concernant les MSA pour ouvrir des portes présentent uniquement des systèmes prenant contrôle du MSA et du fauteuil pour accomplir la tâche, contrairement à ce travail qui cherche à ne pas modifier le fonctionnement du fauteuil roulant de l'utilisateur. Malgré cette différence avec les études passées, certaines solutions utilisées dans ce travail proviennent de manipulateurs mobiles développés pour ouvrir des portes.

En ce qui concerne la détection des poignées, le système utilise un algorithme de reconnaissance d'objet Faster R-CNN entraîné par la base de donnée provenant de l'article de Arduengo et coll. (2021) [26]. De plus, et inspiré de l'article de Llopart et coll. (2017) [14], l'information tridimensionnelle de la scène, obtenue à l'aide d'une caméra Intel D415, est également utilisée lors de la détection pour permettre de filtrer les faux positifs générés par l'algorithme Faster R-CNN. L'ajout de cette deuxième méthode a démontré une grande amélioration du système de détection. Effectivement, malgré les mauvaises détections régulièrement produites par Faster R-CNN, le système a démontré un grand taux de réussite pour la détection des poignées de porte rondes et à levier. Toutefois, pour les poignées à tirer, l'algorithme Faster R-CNN nécessite certaines améliorations pour obtenir un taux de détection plus satisfaisant. En ce qui concerne la classification des poignées, un algorithme de classification utilisant des descripteurs HOG a été développé. Avec l'ensemble des expérimentations ne générant aucune erreur de classification, cette méthode s'est avérée fort efficace pour déterminer la classe des poignées.

Ensuite, pour les systèmes de contrôles, la solution contenait une boucle de contrôle par position utilisant un correcteur PID et certaines compensations provenant de la dynamique du manipulateur pour atteindre la poignée ainsi que deux boucles de contrôle par impédance sans retour de force pour respectivement opérer les poignées, et ensuite ouvrir la porte. Les deux premiers systèmes

présentés dans ce travail se sont avérés fort efficaces pour atteindre et opérer la poignée, à l'exception des poignées rondes qui possèdent une constante de résistance du ressort supérieur à 5.6 Nm/rad. Toutefois, en ce qui concerne l'ouverture de la porte, les résultats ont démontré que cette tâche peut facilement être échouée lorsque les déplacements du fauteuil sont contrôlés par l'utilisateur. Effectivement, il est possible pour le manipulateur d'assister l'utilisateur lors de l'ouverture de la porte, mais l'orientation de l'effecteur et la position initiale du fauteuil doivent être bien choisies pour réduire les chances que l'effecteur glisse de la poignée lors de la tâche. De ce fait, certaines décisions initiales présentées dans ce travail ont dû être repensées et modifiées.

L'objectif principal de développer un système intelligent d'assistance à l'ouverture des portes pour les utilisateurs en fauteuil roulant en asservissant un manipulateur Jaco² 6 DDL fixé sur le fauteuil a donc été atteint. Toutefois, celui-ci n'est pas parfait et nécessite plusieurs améliorations. Entre autres, le système de reconnaissance par Faster R-CNN n'est pas encore idéal. Pour cette raison, il est recommandé de l'entraîner à nouveau à l'aide d'une base de données plus complète et représentant mieux l'environnement de la tâche. De plus, il serait intéressant d'évaluer le système avec des portes possédant un système de fermeture automatique et observer quelles seraient les modifications à apporter. Expérimenter le tout à l'aide d'un système réel serait également important, mais nécessiterait avant tout une amélioration de l'effecteur pour assurer que celui-ci conserve mieux sa prise sur les poignées. Finalement, ce travail a également démontré la possibilité qu'utiliser un manipulateur possédant un poignet sphérique soit plus avantageux pour ouvrir les portes. Une étude intéressante serait donc d'évaluer si ce changement est plus adéquat pour assister les utilisateurs en fauteuil roulant pour ouvrir les portes.

RÉFÉRENCES

- [1] S. Dawson et L. J. Kristjanson, « Mapping the journey: family carers' perceptions of issues related to end-stage care of individuals with muscular dystrophy or motor neurone disease », *J. Palliat. Care*, vol. 19, n° 1, p. 36-42, 2003.
- [2] I. Laffont *et al.*, « Evaluation of a Graphic Interface to Control A Robotic Grasping Arm: A Multicenter Study », *Arch. Phys. Med. Rehabil.*, vol. 90, n° 10, p. 1740-1748, oct. 2009, doi: 10.1016/j.apmr.2009.05.009.
- [3] G. R. B. E. Romer, H. J. A. Stuyt, et A. Peters, « Cost-savings and economic benefits due to the assistive robotic manipulator (ARM) », dans *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, juin 2005, p. 201-204. doi: 10.1109/ICORR.2005.1501084.
- [4] V. Maheu, P. S. Archambault, J. Frappier, et F. Routhier, « Evaluation of the JACO robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities », dans *2011 IEEE International Conference on Rehabilitation Robotics*, juin 2011, p. 1-5. doi: 10.1109/ICORR.2011.5975397.
- [5] Kinova, « KINOVA_Gen2_Ultra_lightweight_robot_User_guide », *Kinova Robotics*. https://drive.google.com/file/d/1xQbkx1-v3SfAentKR9f3p3c2SVdViyQl/view?usp=sharing&usp=embed_facebook (consulté le 6 avril 2022).
- [6] M. Beaudoin, J. Lettre, F. Routhier, P. S. Archambault, M. Lemay, et I. Gélinas, « Long-term use of the JACO robotic arm: a case series », *Disabil. Rehabil. Assist. Technol.*, vol. 14, n° 3, p. 267-275, avr. 2019, doi: 10.1080/17483107.2018.1428692.
- [7] W. Guo, J. Wang, et W. Chen, « A manipulability improving scheme for opening unknown doors with mobile manipulator », dans *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, déc. 2014, p. 1362-1367. doi: 10.1109/ROBIO.2014.7090523.
- [8] S. W. Brose *et al.*, « The Role of Assistive Robotics in the Lives of Persons with Disability », *Am. J. Phys. Med. Rehabil.*, vol. 89, n° 6, p. 509-521, juin 2010, doi: 10.1097/PHM.0b013e3181cf569b.

- [9] F. Routhier, P. Archambault, M.-C. Cyr, V. Maheu, M. Lemay, et I. Gélinas, « BENEFITS OF JACO ROBOTIC ARM ON INDEPENDENT LIVING AND SOCIAL PARTICIPATION: AN EXPLORATORY STUDY », p. 4.
- [10] S.-K. Chang et C. Drury, « Task demands and human capabilities in door use », *Appl. Ergon.*, vol. 38, p. 325-35, juin 2007, doi: 10.1016/j.apergo.2006.04.023.
- [11] R. B. Rusu, W. Meeussen, S. Chitta, et M. Beetz, *Laser-based perception for door and handle identification*. 2009, p. 8.
- [12] B. Gaspers, J. Welle, et D. Schulz, « Opening doors with a mobile manipulator without force-torque feedback », dans *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, oct. 2013, p. 1-6. doi: 10.1109/SSRR.2013.6719354.
- [13] « UTM-30LN », *Hokuyo*. <https://hokuyo-usa.com/products/lidar-obstacle-detection/utm-30ln> (consulté le 3 mai 2022).
- [14] A. Llopart, O. Ravn, et Nils. A. Andersen, « Door and cabinet recognition using Convolutional Neural Nets and real-time method fusion for handle detection and grasping », dans *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, avr. 2017, p. 144-149. doi: 10.1109/ICCAR.2017.7942676.
- [15] M. Stuede, K. Nuelle, S. Tappe, et T. Ortmaier, « Door opening and traversal with an industrial cartesian impedance controlled mobile robot », dans *2019 International Conference on Robotics and Automation (ICRA)*, mai 2019, p. 966-972. doi: 10.1109/ICRA.2019.8793866.
- [16] « Kinect - Windows app development ». <https://developer.microsoft.com/en-us/windows/kinect/> (consulté le 4 mai 2022).
- [17] C. Bousquet-Jette, S. Achiche, D. Beaini, Y. S. Law-Kam Cio, C. Leblond-Ménard, et M. Raison, « Fast scene analysis using vision and artificial intelligence for object prehension by an assistive robot », *Eng. Appl. Artif. Intell.*, vol. 63, p. 33-44, août 2017, doi: 10.1016/j.engappai.2017.04.015.
- [18] C. Ott, B. Bäuml, C. Borst, et G. Hirzinger, « AUTONOMOUS OPENING OF A DOOR WITH A MOBILE MANIPULATOR: A CASE STUDY », *IFAC Proc. Vol.*, vol. 40, n° 15, p. 349-354, janv. 2007, doi: 10.3182/20070903-3-FR-2921.00060.

- [19] E. Klingbeil, A. Saxena, et A. Y. Ng, « Learning to open new doors », dans *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, oct. 2010, p. 2751-2757. doi: 10.1109/IROS.2010.5649847.
- [20] A. Andreopoulos et J. K. Tsotsos, « A Framework for Door Localization and Door Opening Using a Computer Controlled Wheelchair for People Living with Mobility Impairments », 2007. <https://www.semanticscholar.org/paper/A-Framework-for-Door-Localization-and-Door-Opening-Andreopoulos-Tsotsos/9ae8c209083bdcec13a0a230665d042d1557eb6a> (consulté le 3 mai 2022).
- [21] J. G. Ramôa, L. A. Alexandre, et S. Mogo, « Real-Time 3D Door Detection and Classification on a Low-Power Device », dans *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, avr. 2020, p. 96-101. doi: 10.1109/ICARSC49921.2020.9096155.
- [22] « Bumblebee® XB3 FireWire | Teledyne FLIR ». <https://www.flir.quebec/products/bumblebee-xb3-firewire?vertical=machine+vision&segment=iis> (consulté le 4 mai 2022).
- [23] « Depth Camera D435 », *Intel® RealSense™ Depth and Tracking Cameras*. <https://www.intelrealsense.com/depth-camera-d435/> (consulté le 4 mai 2022).
- [24] C. Messom et A. Barczak, « Stream processing for fast and efficient rotated Haar-like features using rotated integral images », *IJISTA*, vol. 7, p. 40-57, janv. 2009, doi: 10.1504/IJISTA.2009.025105.
- [25] S. Ren, K. He, R. Girshick, et J. Sun, « Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks », *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, n° 6, p. 1137-1149, juin 2017, doi: 10.1109/TPAMI.2016.2577031.
- [26] M. Arduengo, C. Torras, et L. Sentis, « Robust and adaptive door operation with a mobile robot », *Intell. Serv. Robot.*, vol. 14, n° 3, p. 409-425, juill. 2021, doi: 10.1007/s11370-021-00366-7.
- [27] W. Meeussen *et al.*, « Autonomous door opening and plugging in with a personal robot », dans *2010 IEEE International Conference on Robotics and Automation*, mai 2010, p. 729-736. doi: 10.1109/ROBOT.2010.5509556.

- [28] W. Chung, C. Rhee, Y. Shim, H. Lee, et S. Park, « Door-Opening Control of a Service Robot Using the Multifingered Robot Hand », *IEEE Trans. Ind. Electron.*, vol. 56, n° 10, p. 3975-3984, oct. 2009, doi: 10.1109/TIE.2009.2025296.
- [29] H. Gao *et al.*, « Dynamic Modeling and Experimental Validation of Door-Opening Process by a Mobile Manipulator », *IEEE Access*, vol. 7, p. 80916-80927, 2019, doi: 10.1109/ACCESS.2019.2919964.
- [30] P. Viola et M. Jones, « Robust real-time face detection », dans *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, juill. 2001, vol. 2, p. 747-747. doi: 10.1109/ICCV.2001.937709.
- [31] J. Redmon, S. Divvala, R. Girshick, et A. Farhadi, « You Only Look Once: Unified, Real-Time Object Detection », dans *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, juin 2016, p. 779-788. doi: 10.1109/CVPR.2016.91.
- [32] K. He, G. Gkioxari, P. Dollar, et R. Girshick, « Mask R-CNN », 2017, p. 2961-2969. Consulté le: 5 mai 2022. [En ligne]. Disponible à: https://openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html
- [33] J. Uijlings, K. Sande, T. Gevers, et A. W. M. Smeulders, « Selective Search for Object Recognition », *Int. J. Comput. Vis.*, vol. 104, p. 154-171, sept. 2013, doi: 10.1007/s11263-013-0620-5.
- [34] R. Girshick, J. Donahue, T. Darrell, et J. Malik, « Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation », dans *2014 IEEE Conference on Computer Vision and Pattern Recognition*, juin 2014, p. 580-587. doi: 10.1109/CVPR.2014.81.
- [35] R. Girshick, « Fast R-CNN », dans *2015 IEEE International Conference on Computer Vision (ICCV)*, déc. 2015, p. 1440-1448. doi: 10.1109/ICCV.2015.169.
- [36] L. Tan, T. Huangfu, L. Wu, et W. Chen, « Comparison of YOLO v3, Faster R-CNN, and SSD for Real-Time Pill Identification », In Review, preprint, juill. 2021. doi: 10.21203/rs.3.rs-668895/v1.

- [37] N. Dalal et B. Triggs, « Histograms of oriented gradients for human detection », dans *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, juin 2005, vol. 1, p. 886-893 vol. 1. doi: 10.1109/CVPR.2005.177.
- [38] A. Aristidou et J. Lasenby, *Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver*. 2009.
- [39] C. Gosselin et H. Liu, « Polynomial Inverse Kinematic Solution of the Jaco Robot », présenté à ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, janv. 2015. doi: 10.1115/DETC2014-34152.
- [40] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Third edition. Pearson/Prentice Hall, 2005.
- [41] H. Ochoa, « Compliant Control of the Kinova Robot for Surface Polishing », *undefined*, 2019, Consulté le: 6 mai 2022. [En ligne]. Disponible à: <https://www.semanticscholar.org/paper/Compliant-Control-of-the-Kinova-Robot-for-Surface-Ochoa/f727718b0b71546e4938ab7b7a0336425dd39a62>
- [42] L. Wang, *PID Control System Design and Automatic Tuning using MATLAB/Simulink*, 1^{re} éd. Wiley, 2020. doi: 10.1002/9781119469414.
- [43] J. L. Guzmán, T. Hägglund, K. J. Åström, S. Dormido, M. Berenguel, et Y. Piguet, « Understanding PID design through interactive tools », *IFAC Proc. Vol.*, vol. 47, n° 3, p. 12243-12248, janv. 2014, doi: 10.3182/20140824-6-ZA-1003.01328.
- [44] M. V. Subramanyam, K. Satyaprasad, et G. krishna rao P V, *Study on PID Controller Design and Performance Based on Tuning Techniques*. 2014. doi: 10.1109/ICCICCT.2014.6993183.
- [45] G. Ellis, *Control System Design Guide*. 2012. doi: 10.1016/B978-0-12-237470-8.50008-9.
- [46] T. N. L. Vu et M. Lee, « Multi-loop PI controller design based on the direct synthesis for interacting multi-time delay processes », *ISA Trans.*, vol. 49, n° 1, p. 79-86, janv. 2010, doi: 10.1016/j.isatra.2009.09.002.

- [47] G. Descôteaux, « Système autonome d'assistance au repas pour les personnes avec un handicap du haut du corps », masters, Polytechnique Montréal, 2020. Consulté le: 7 avril 2022. [En ligne]. Disponible à: <https://publications.polymtl.ca/5380/>
- [48] C. Bousquet-Jetté, « Analyse de scène rapide utilisant la vision et l'intelligence artificielle pour la préhension d'objets par un robot d'assistance », masters, École Polytechnique de Montréal, 2016. Consulté le: 9 mai 2022. [En ligne]. Disponible à: <https://publications.polymtl.ca/2225/>
- [49] Y.-S. Law-Kam Cio, « Asservissement d'un bras robotique d'assistance à l'aide d'un système de stéréo vision artificielle et d'un suiveur de regard », masters, École Polytechnique de Montréal, 2017. Consulté le: 9 mai 2022. [En ligne]. Disponible à: <https://publications.polymtl.ca/2704/>
- [50] A. Campeau-Lecours *et al.*, « Kinova Modular Robot Arms for Service Robotics Applications », *Int. J. Robot. Appl. Technol.*, vol. 5, p. 49-71, juill. 2017, doi: 10.4018/IJRAT.2017070104.
- [51] Kinova, « KINOVA Gen3 Ultra lightweight robot User Guide », *Kinova Robotics*. <https://www.kinovarobotics.com/product/gen2-robots> (consulté le 6 avril 2022).
- [52] Vishal et A. Ranjan Bairagi, « Design of Door Handle », mai 2014, Consulté le: 9 mai 2022. [En ligne]. Disponible à: <https://core.ac.uk/reader/53190392>
- [53] T.-Y. Lin *et al.*, « Microsoft COCO: Common Objects in Context », dans *Computer Vision – ECCV 2014*, Cham, 2014, p. 740-755. doi: 10.1007/978-3-319-10602-1_48.
- [54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, et L. Fei-Fei, « ImageNet: A large-scale hierarchical image database », dans *2009 IEEE Conference on Computer Vision and Pattern Recognition*, juin 2009, p. 248-255. doi: 10.1109/CVPR.2009.5206848.
- [55] A. Kuznetsova *et al.*, « The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale », *Int. J. Comput. Vis.*, vol. 128, n° 7, p. 1956-1981, juill. 2020, doi: 10.1007/s11263-020-01316-z.
- [56] « Choosing an Intel® RealSense™ Depth Camera », *Intel® RealSense™ Depth and Tracking Cameras*, 25 mai 2018. <https://www.intelrealsense.com/compare/> (consulté le 10 mai 2022).

- [57] Intel Realsense, « Compare Intel RealSense Depth Cameras (Tech specs and Review) », *Intel® RealSense™ Depth and Tracking Cameras*. <https://www.intelrealsense.com/compare-depth-cameras/> (consulté le 7 avril 2022).
- [58] M. Roser, C. Appel, et H. Ritchie, « Human Height », *Our World Data*, oct. 2013, Consulté le: 7 avril 2022. [En ligne]. Disponible à: <https://ourworldindata.org/human-height>
- [59] K. M. Bushby, T. Cole, J. N. Matthews, et J. A. Goodship, « Centiles for adult head circumference. », *Arch. Dis. Child.*, vol. 67, n° 10, p. 1286-1287, oct. 1992, doi: 10.1136/adc.67.10.1286.
- [60] RONA, « Door handles and locks: the key to choosing wisely | RONA ». <https://www.rona.ca/en/workshop/guides/choosing-door-handles-and-locks> (consulté le 7 avril 2022).
- [61] A. (Ari) Bornstein, « Why Should I Use PyTorch Lightning? », *Medium*, 7 juin 2021. <https://devblog.pytorchlightning.ai/why-should-i-use-pytorch-lightning-488760847b8b> (consulté le 11 mai 2022).
- [62] « Models and pre-trained weights — Torchvision 0.12 documentation ». <https://pytorch.org/vision/stable/models.html#id63> (consulté le 11 mai 2022).
- [63] S. Choi, T. Kim, et W. Yu, *Performance evaluation of RANSAC family*, vol. 24. 2009. doi: 10.5244/C.23.81.
- [64] « ROS packages for Jaco2 and Mico robotic arms ». Kinova Robotics, 5 mai 2022. Consulté le: 15 mai 2022. [En ligne]. Disponible à: <https://github.com/Kinovarobotics/kinova-ros>
- [65] A. Patil, « Nuric wheelchair model 2 ». 22 novembre 2021. Consulté le: 15 mai 2022. [En ligne]. Disponible à: https://github.com/patilnabhi/nuric_wheelchair_model_02
- [66] M. P. Mendes, « Computed-torque control of the Kinova JACO² Arm », présenté à Computed-torque control of the Kinova JACO² Arm, sept. 2017. Consulté le: 15 mai 2022. [En ligne]. Disponible à: <https://estudogeral.sib.uc.pt/handle/10316/82882>
- [67] « KINOVA Actuator Series user guide », *Kinova Robotics*. <https://drive.google.com/file/d/1zGBHD->

AuB9Hli0ppUz9s4Ivs5qCP3QUd/view?usp=sharing&usp=embed_facebook (consulté le 15 mai 2022).

[68] E. Ivanjko, T. Petrini, et I. Petrovic, « Modelling of Mobile Robot Dynamics », janv. 2010.

[69] « Properties and grades - Hardwood Lumber - Quebec Wood Export Bureau (QWEB) », *Quebec Wood Export Bureau*. <https://quebecwoodexport.com/en/products/hardwood-lumber/properties-and-grades/> (consulté le 15 mai 2022).

[70] M. A. Rahman et Y. Wang, « Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation », dans *Advances in Visual Computing*, Cham, 2016, p. 234-244. doi: 10.1007/978-3-319-50835-1_22.

[71] « The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Development Kit ». http://host.robots.ox.ac.uk/pascal/VOC/voc2012/html/doc/devkit_doc.html (consulté le 11 mai 2022).

ANNEXE A STRUCTURE DE LA MODELISATION DANS SIMSCAPE

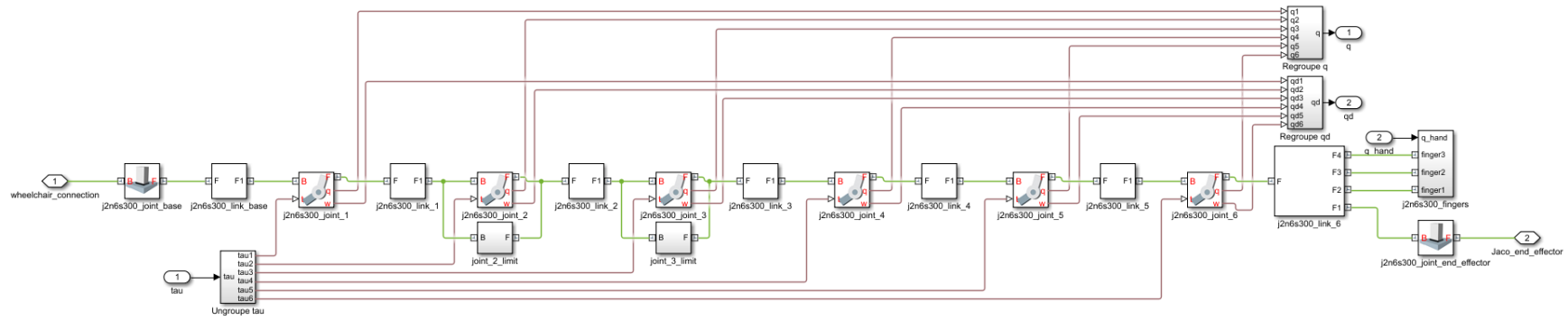


Figure A.1 Structure de la modélisation du Jaco² sur Simscape

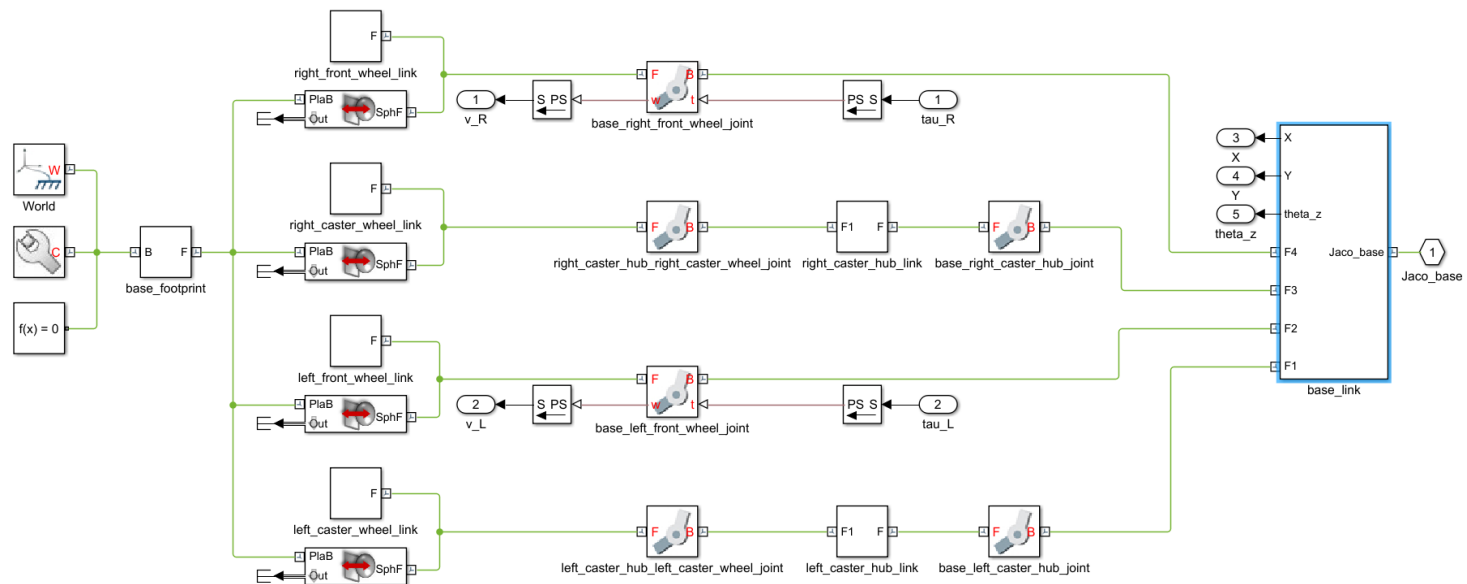


Figure A.2 Structure de la modélisation du fauteuil sur Simscape

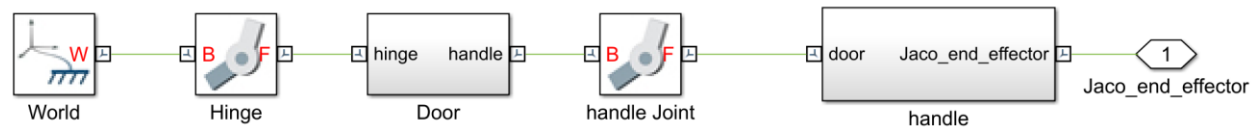


Figure A.3 Structure de la modélisation de la porte et de la poignée sur Simscape

ANNEXE B INFORMATIONS DU MANIPULATEUR JACO²

Tableau B.1 Dimensions du Jaco²

Jaco ² dimensions (en mètres)	
D1	0.2755
D2	0.4100
e2	0.0098
D3	0.2073
D4	0.0741
D5	0.0741
D6	0.1600

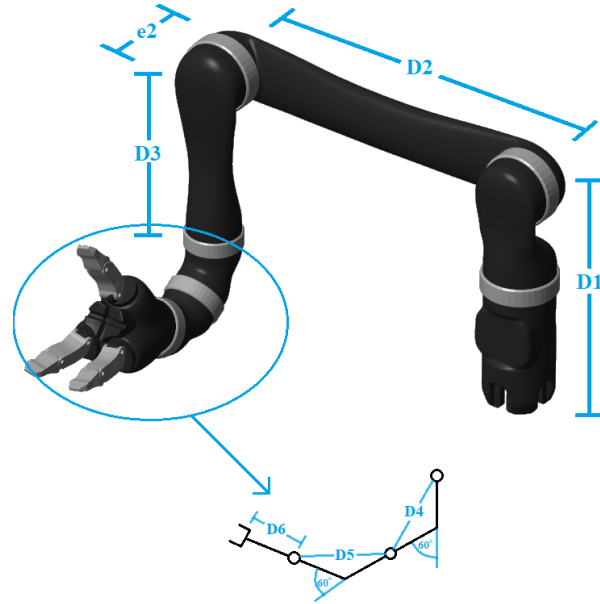


Figure B.1 Dimensions du Jaco²

Tableau B.2 Limites angulaires

Actuateurs	Min (°)	Max (°)
1	-10000	10000
2	50	310
3	19	341
4	-10000	10000
5	-10000	10000
6	-10000	10000

Tableau B.3 Paramètre pour DH

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	90°	0	D1	q1
2	180°	D2	0	q2
3	90°	0	-e2	q3
4	60°	0	-d4b	q4
5	60°	0	-d5b	q5
6	180	0	-d6b	q6

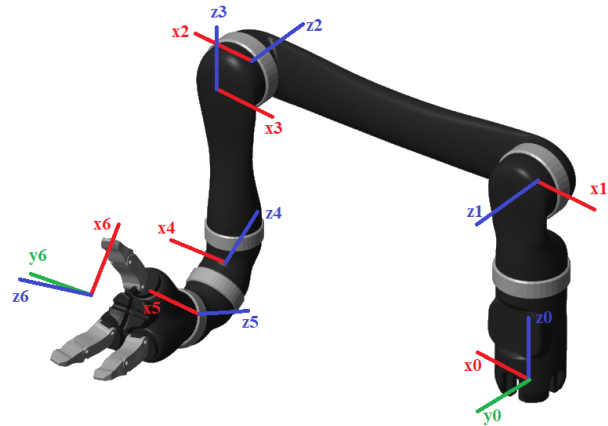


Figure B.2 Structure des coordonnées pour DH

$$\text{Où: } \begin{cases} d4b = D3 + \frac{D5}{2\cos(30^\circ)} \\ d5b = \frac{D4 + D5}{2\cos(30^\circ)} \\ d6b = D6 + \frac{D5}{2\cos(30^\circ)} \end{cases}$$

Tableau B.4 Relation entre DH et le réel

Angle DH	Angle Réel
q1	-q1
q2	q2 + 90°
q3	q3 - 90°
q4	q4
q5	q5 + 180°
q6	q6 - 90°

Tableau B.5 Paramètres pour la dynamique du Jaco² (masse en kilogrammes, centre de masse en mètres et l'inertie est en kg*m²)

	1	2	3	4	5	6
m	0.767664	0.99571300239	0.79668391394	0.41737	0.41737	1.07541
r_x	-0.0000638421371	-0.3225949597019	0.0000260855872	-0.0356524456753	-0.0225304783397	-0.0046337046336
r_y	-0.0958711563399	0.0000226798333	-0.0016756951312	-0.0000166954980	0.0000166954980	0.0000138010619
r_z	0.0024244913869	0.0141269620294	-0.0170639369991	0.0199130976902	0.0235353266641	-0.1062512535684
I_{txx}	0.0022310838229	0.0041625249431	0.0028612542225	0.0007084767282	0.0008274776043	0.0048337559453
I_{txy}	0.0000068156372	-0.0000005527369	0.0000004027733	0.0000082716433	0.0000082810781	0.0000043311794
I_{txz}	-0.0000197876009	-0.0014874685854	0.0000005664070	0.0001128339615	-0.0001016901654	0.0003615961394
I_{tyy}	0.0006207338407	0.0254954292810	0.0027386563864	0.0007404962916	0.0008520817703	0.0048415034931
I_{tyz}	-0.0003062887104	-0.0000002770111	-0.0003446591689	0.0000004942735	-0.0000000542227	-0.0000024551314
I_{tzz}	0.0023980074412	0.0217369354502	0.0003512427525	0.0001781932952	0.0001707778978	0.0001998215195

Où « m » est la masse augmentée, incluant l'actuateur, du membre i , « r_x, r_y, r_z » représente le centre de masse augmenté du membre i selon les coordonnées DH i , et les « I_t » forme la matrice d'inertie du membre i selon les coordonnées DH i .

ANNEXE C POSITIONS INTERMÉDIAIRES POUR ATTEINDRE LA POIGNÉE

Tableau C.1 Positions intermédiaires utilisé pour atteindre la poignée

Trajectoires	Position cartésienne (m)	Position angulaire (rad)
Poignée à levier (droit)	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -0.3 \\ 0 & 0 & -1 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 4.5779 \\ 3.1413 \\ 1.8130 \\ 5.4537 \\ 2.2831 \\ 0.6068 \end{bmatrix}$
Poignée à levier (gauche)	$\begin{bmatrix} 0 & 1 & 0 & 0.5 \\ 1 & 0 & 0 & -0.2 \\ 0 & 0 & -1 & 0.45 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 3.4651 \\ 3.7725 \\ 2.5656 \\ 5.2902 \\ 2.5113 \\ 5.6137 \end{bmatrix}$
Poignée ronde (droit)	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -0.5 \\ -1 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 4.5801 \\ 4.1521 \\ 0.6452 \\ 0.8483 \\ 3.7760 \\ 2.5102 \end{bmatrix}$
Poignée ronde (gauche)	$\begin{bmatrix} 0 & 1 & 0 & 0.5 \\ 0 & 0 & -1 & -0.17 \\ -1 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 2.9891 \\ 4.0264 \\ 1.3693 \\ 0.7672 \\ 2.0127 \\ 4.5237 \end{bmatrix}$
Poignée à tirer (droit)	$\begin{bmatrix} 0 & 0 & 1 & -0.1 \\ 1 & 0 & 0 & -0.3 \\ 0 & 1 & 0 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 5.6766 \\ 3.9277 \\ 1.3671 \\ 3.6674 \\ 3.5051 \\ 0.8604 \end{bmatrix}$
Poignée à tirer (gauche)	$\begin{bmatrix} 0 & 0 & 1 & 0.2 \\ 1 & 0 & 0 & -0.3 \\ 0 & 1 & 0 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 4.9390 \\ 3.8193 \\ 0.6305 \\ 2.4052 \\ 4.3875 \\ 0.6541 \end{bmatrix}$

Note : il est important de mentionner que la position du manipulateur Jaco² sur un fauteuil roulant peut varier grandement. Dans le cadre de ce travail, celui-ci a été positionné dans l'avant droit du fauteuil à une hauteur du sol de 0.65 mètre.

ANNEXE D SCORES DU FASTER R-CNN FOURNIS PAR L'ÉVALUATEUR COCO

Tableau D.1 Scores du Faster R-CNN fournis par l'évaluateur COCO

Métriques	Régions	Détections maximales	Scores
mAP@[0.5:0.95]IoU	Toutes	100	0.233
mAP@0.5IoU	Toutes	100	0.460
mAP@0.75IoU	Toutes	100	0.209
mAP@[0.5:0.95]IoU	Petit	100	0.048
mAP@[0.5:0.95]IoU	Moyen	100	0.133
mAP@[0.5:0.95]IoU	Grand	100	0.274
mAR@[0.5:0.95]IoU	Toutes	1	0.100
mAR@[0.5:0.95]IoU	Toutes	10	0.320
mAR@[0.5:0.95]IoU	Toutes	100	0.358
mAR@[0.5:0.95]IoU	Petit	100	0.091
mAR@[0.5:0.95]IoU	Moyen	100	0.236
mAR@[0.5:0.95]IoU	Grand	100	0.412

ANNEXE E GAINS UTILISÉS POUR LES SYSTÈMES DE CONTRÔLE

Tableau E.1 Gains utilisés pour atteindre la poignée de porte

Trajectoires	Gains utilisé (matrice diagonale)
Poignée à levier (droit)	$K_{proportionnel} = diag[65, 250, 30, 5, 8, 0.3]$
	$K_{derivatif} = diag[2, 5, 1.2, 0.2, 0.064, 0.015]$
	$K_{intégral} = diag[0.128, 200, 40, 10, 20, 10]$
Poignée à levier (gauche)	$K_{proportionnel} = diag[65, 250, 30, 5, 8, 0.3]$
	$K_{derivatif} = diag[2, 5, 1.2, 0.2, 0.064, 0.015]$
	$K_{intégral} = diag[0.128, 200, 40, 10, 20, 10]$
Poignée ronde (droit)	$K_{proportionnel} = diag[65, 250, 60, 12, 4, 0.15]$
	$K_{derivatif} = diag[2, 5, 2, 0.2, 0.05, 0.02]$
	$K_{intégral} = diag[0.128, 200, 60, 20, 2, 2]$
Poignée ronde (gauche)	$K_{proportionnel} = diag[65, 250, 40, 25, 6, 0.3]$
	$K_{derivatif} = diag[2, 5, 1.2, 0.5, 0.064, 0.015]$
	$K_{intégral} = diag[0.128, 200, 80, 64, 5, 5]$
Poignée à tirer (droit)	$K_{proportionnel} = diag[65, 250, 30, 7, 2, 0.3]$
	$K_{derivatif} = diag[2, 5, 1, 0.1, 0.05, 0.015]$
	$K_{intégral} = diag[0.128, 200, 25, 4, 1.5, 5]$
Poignée à tirer (gauche)	$K_{proportionnel} = diag[65, 250, 40, 10, 2, 0.3]$
	$K_{derivatif} = diag[2, 5, 2, 0.2, 0.05, 0.015]$
	$K_{intégral} = diag[0.128, 200, 80, 10, 1, 5]$

Tableau E.2 Gains utilisés pour opérer la poignée de porte

Gains utilisé (matrice diagonale)
$K = diag[1000, 1000, 1000, 0.05, 0.10, 0.00]$
$C = diag[800, 800, 800, 0.01, 0.01, 0.01]$
$K_i = diag[0, 0, 0, 0, 5, 0]$
$C_q = diag[0.05, 0.05, 0.05, 0.05, 0.05, 0.005]$

Tableau E.3 Gains utilisés pour ouvrir la porte

Gains utilisé (matrice diagonale)
$K = diag[10, 10, 100, 0.50, 0.50, 0.00]$
$C = diag[8, 8, 20, 0.01, 0.01, 0.01]$
$C_q = diag[0.05, 0.05, 0.05, 0.05, 0.05, 0.005]$