



HAL
open science

Etude et développement de tâches de manipulation mobile dans le contexte de l'industrie 4.0

Wafae Sebbata

► **To cite this version:**

Wafae Sebbata. Etude et développement de tâches de manipulation mobile dans le contexte de l'industrie 4.0. Automatique / Robotique. Normandie Université, 2021. Français. NNT : 2021NORMLH20 . tel-03655797

HAL Id: tel-03655797

<https://theses.hal.science/tel-03655797v1>

Submitted on 30 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité Automatique, Productique and Robotique

Préparée au sein de l'université Le Havre Normandie

Étude et développement de tâches de manipulation mobile dans le contexte de l'industrie 4.0

Présentée et soutenue par **Wafae SEBBATA**

Thèse soutenue publiquement le 29-10-2021 devant le jury composé de

Patrick BONNIN	Professeur des Universités, Université de Versailles Saint-Quentin	Rapporteur et Examineur
Jean-Pierre GAZEAU	Ingénieur de recherche (HDR), Université de Poitiers	Rapporteur et Examineur
Pierre BLAZEVIC	Professeur des Universités, Université de Versailles Saint-Quentin	Examineur
Viviane CADENAT	Maîtresse de conférences (HDR), Université de Toulouse III	Examinatrice
Luc ROLLAND	Enseignant-chercheur à University of West Scotland	Examineur
M'hammed SAHNOUN	Enseignant-chercheur, CESI Rouen	Examineur
Sophie CROS	Professeure des Universités, Université Le Havre Normandie	Examinatrice
Jean-François BRETHÉ	Professeur des Universités, Université Le Havre Normandie	Directeur de thèse

Remerciements

Je tiens à remercier Messieurs Patrick BONNIN et Jean-Pierre GAZEAU pour leur consciencieux travail de rapporteurs, et les membre du jury d'avoir examiné ma thèse.

Je remercie profondément le professeur Jean-François BRETHÉ, mon directeur de thèse, pour son encadrement, son suivi scientifique et ses précieux conseils durant toute la période de préparation de ma thèse ainsi que ses qualités humaines envers les doctorants.

J'exprime mes remerciements à l'ensemble du personnel de l'ISEL-PIL, aux collègues partenaires du projet CoRoT et aux collègues du laboratoire GREAH. Un merci spécial à l'équipe robotique du laboratoire pour la richesse de nos échanges et la convivialité des partages d'informations. Je remercie en particulier le docteur Mourad KENK pour son aide durant son séjour et sa collaboration avec l'équipe robotique.

Je n' imagine pas cette aventure de thèse sans le soutien indéfectible de ma famille. Certes, cela a été difficile loin d'eux, mais leurs encouragements quotidiens m'ont permis de surmonter tout les obstacles. Je remercie chaleureusement mes très chers parents pour tout ce qu'ils ont fait pour moi, mes frères pour leurs encouragements, mes grands-parents pour leurs invocations. Une très grande pensée à ma grand-mère Hasna qui nous a quitté avant de voir l'aboutissement de mon travail mais je sais qu'elle serait très fière de moi.

Mes remerciements vont aussi à mes ami(e)s pour leur soutien et les moments familiaux que nous avons vécu ensemble. Je remercie toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

À mes très chers parents

Table des matières

Remerciements	iii
Introduction Générale	13
0.1 Organisation du manuscrit	15
0.2 Publications associées à cette thèse	16
1 Les manipulateurs mobiles : État de l’art	17
1.1 Les manipulateurs mobiles et leurs milieux d’intervention	18
1.1.1 Les manipulateurs mobiles sous-marins	19
1.1.2 Les manipulateurs mobiles aériens	19
1.1.3 Les manipulateurs mobiles spatiaux	20
1.1.4 Les manipulateurs mobiles terrestres	21
1.2 La mécatronique des manipulateurs mobiles à roues	23
1.2.1 La plate-forme mobile « à roues »	24
1.2.2 Les bras manipulateurs	26
1.2.3 Les préhenseurs	30
1.2.4 Les applications des manipulateurs mobiles à roues	31
1.3 Instrumentation des manipulateurs mobiles	31
1.3.1 Capteurs usuels en robotique de manipulation	32
1.3.2 Capteurs usuels en robotique mobile	34
1.3.3 Le traitement d’image	35
1.4 Commande des manipulateurs mobiles	36
1.4.1 Commande des bras manipulateurs	37
1.4.2 Commande des robots mobiles	37
1.4.3 Commande des manipulateurs mobiles	37
2 Alimentation robotisée autonome d’une machine-outil	39
2.1 Le projet CoRoT	40
2.1.1 Le contexte du projet	40
2.1.2 Les finalités du projet	41

2.1.3	Les livrables du projet	41
2.2	Le cas d'étude GREAH	42
2.2.1	Le scénario	42
2.2.2	La plate-forme robotique utilisée	43
2.2.3	Principes de navigation d'un manipulateur mobile	45
2.2.4	Principes de manipulation robotique	54
2.3	Mise en oeuvre du cas d'étude	58
2.3.1	Navigation du manipulateur mobile	58
2.3.2	La saisie et la dépose des pièces par le manipulateur mobile	58
2.3.3	La communication avec la machine d'usinage	60
3	La saisie des pièces basée sur l'apprentissage profond	63
3.1	L'identification de l'objet par apprentissage profond	64
3.1.1	YOLO	64
3.1.2	L'apprentissage par transfert	70
3.2	Le cas d'étude GREAH : Saisie de pièces par préhension adaptative	71
3.2.1	Les objectifs	71
3.2.2	Description de la plate-forme robotique	72
3.2.3	Le processus de préhension d'objet	73
3.2.4	Les étapes de pick-and-place avec préhension adaptative	75
3.3	Les résultats expérimentaux	78
4	Modélisation et évaluation des incertitudes de pose d'un manipulateur mobile	91
4.1	État de l'art sur l'alimentation robotisée des machines	92
4.2	Modélisation de la précision de positionnement d'un manipulateur mobile	94
4.2.1	Modélisation de la précision du bras manipulateur	95
4.2.2	Modélisation de la précision du robot mobile	99
4.2.3	Modélisation de la précision du manipulateur mobile	101
4.3	Stratégie optimale pour un assemblage contraint	105
	Bibliographie	119

Table des figures

1.1	Les manipulateurs mobiles sous-marins	20
1.2	Les manipulateurs mobiles aériens	20
1.3	Les manipulateurs mobiles spatiaux	21
1.4	Des exemples des robots multipèdes	22
1.5	Les manipulateurs mobiles hybrides	23
1.6	Des manipulateurs mobiles avec différents types de base mobile.	26
1.7	Des manipulateurs sériels.	28
1.8	Des manipulateurs parallèles.	29
1.9	Quelques manipulateurs hybrides.	30
1.10	Exemple des types de capteurs caméras.	33
1.11	Exemple des lidars.	34
2.1	La carte de la halle robotique du PIL	44
2.2	Le manipulateur mobile (GREAH)	44
2.3	Représentation du robot dans l’entrepôt simulé PIL du laboratoire	47
2.4	L’architecture de navigation Stack sous ROS	48
2.5	Repère de localisation de la base mobile	50
2.6	LiDAR	50
2.7	Visualisation sur RVIZ des particules distribuées lors de la localisation du robot dans la carte	52
2.8	Le robot localisé dans la carte de l’entrepôt réel PIL sur RVIZ	53
2.9	L’espace des configurations.	55
2.10	les techniques de génération des trajectoires.	56
2.11	Le déplacement du robot dans l’entrepôt PIL	59
2.12	Visualisation sur RVIZ du déplacement du robot dans l’entrepôt PIL	59

TABLE DES FIGURES

2.13	La saisie basée sur les marqueurs code	60
2.14	La machine d'usinage utilisée dans le développement (a) en vue de face (b) le robot se localise par rapport au marqueur code de la machine	61
3.1	La détection par YOLO basée sur l'approche de régression	65
3.2	L'architecture du réseau de neurones YOLO-V3 dans notre cas d'études . . .	68
3.3	L'architecture du réseau de neurones YOLO-V3 dans notre cas d'études . . .	69
3.4	Les principales stratégie de transfert learning.	71
3.5	Quelques produits industriels pouvant être saisis.	72
3.6	L'objet conçu.	73
3.7	La plate-forme robotique utilisée.	74
3.8	Les trois stratégies de préhension.	75
3.9	Architecture de notre système de pick-and-place.	76
3.10	Le cadre 2D de l'objet détecté (à gauche), le cadre 3D de l'objet détecté dans le nuage des points (à droite).	77
3.11	Illustration des trajectoires suivies par le bras robotique dans RVIZ	78
3.12	Le système de préhension adaptative proposé	79
3.13	Des exemples des images (a) synthétiques (b) réelles	81
3.14	Résultats de détection d'objet.	85
3.15	Résultats de classification des face géométriques (YOLO-V3)	85
3.16	Résultats de classification des face géométriques (YOLO-V4)	86
3.17	Les quatre étapes de picking de l'objet multifaces.	87
4.1	Les étapes de prise et dépose de la pièce multifaces pour son usinage.	93
4.2	La répétabilité de positionnement et l'exactitude	95
4.3	La répétabilité d'orientation (ISO 9283)	95
4.4	Le dispositif de mesure pour les tests de répétabilité du robot du bras robotique	96
4.5	Schéma de principe des trièdres en position nominale	98
4.6	Schéma des dispositifs pour les tests sur la précision de la base mobile	100
4.7	Les points du LiDAR avant.	101
4.8	Illustration du point P et des cotations nécessaires.	102
4.9	Vue globale du dispositif métrologique	106

TABLE DES FIGURES

4.10	Position nominale de dépose de la pièce dans l'étai de la machine.	107
4.11	Ellipse de saut à 90 %.	110
4.12	Ellipses de saut au risque de 1% et 1‰	111

Liste des tableaux

3.1	Les scores d'évaluation de la détection d'objet basée sur l'entraînement du réseau par des images synthétiques (YOLO-V3).	83
3.2	Les scores d'évaluation de la détection d'objet basée sur l'entraînement du réseau par des images réelles (YOLO-V3).	83
3.3	Les scores d'évaluation de la détection d'objet basée sur l'entraînement du réseau par des images réelles (YOLO-V4).	84
3.4	Les scores d'évaluation de la classification des faces géométriques basée sur l'entraînement du réseau par des images réelles (YOLO-V3).	86
3.5	Les scores d'évaluation de la classification des faces géométriques basée sur l'entraînement du réseau par des images réelles (YOLO-V4).	88
3.6	Taux de réussite du scénario de pick and place (YOLO-V3).	88
3.7	Taux de réussite du scénario de pick and place (YOLO-V4).	88

Liste des acronymes

GREAH : *Groupe de Recherche en Electronique et Automatique du Havre*

PME : *Petites et Moyennes Entreprises*

UR : *Universal Robot*

DDL : *Degrés De Liberté*

ROS : *Robot Operating System*

URDF : *Unified Robot Description Format*

KDL : *Kinematics and Dynamics Library*

YOLO : *You Only Look Once*

LiDAR : *Light Detection and Ranging*

GPU : *Graphics Processing Unit*

Rviz : *ROS visualization*

Gmapping : *Grid Mapping*

DWA : *Dynamic Window Approach*

AMCL : *Adaptive Monte Carlo Localization*

IoU : *Intersection-over-Union*

CNN : *Convolutional Neural Network*

Introduction Générale

L'industrie 4.0 vise à mettre en place une nouvelle génération d'usines connectées, numérisées et robotisées en intégrant de nouvelles technologies telles que l'Internet des objets, l'intelligence artificielle (IA), la robotique collaborative, etc. Cette révolution contribue à rendre plus flexibles les activités de production pour permettre de franchir de nouvelles étapes : évolution de la production de masse vers la production personnalisée (customisée en français), meilleure accessibilité aux données via leur numérisation (digitalisation), développement d'outils de simulation pour la conception intelligente, pilotage supervisé et à distance des installations industrielles grâce au concept de jumeau numérique. L'ensemble de ces changements structurels et organisationnels vise à mieux répondre aux attentes du consommateur et s'inscrivent aussi dans une démarche de développement durable dont on connaît les enjeux en termes de sauvegarde de l'environnement et de survie de l'espèce humaine.

Les travaux de thèse s'inscrivent dans le cadre du projet CoRoT [1]. Au plein coeur des enjeux de l'industrie 4.0, ce projet vise à améliorer l'outil de production des entreprises manufacturières en particulier les petites et moyennes entreprises, par l'introduction de solutions robotisées conjuguées avec des outils numériques. CoRoT regroupe deux types de partenaires dans la zone France-Manche-Angleterre :

- Des partenaires académiques représentés par ; l'Université Le Havre Normandie, l'ESIGELEC, le CESI, l'Université de Greenwich et l'Université d'Exeter.
- Des partenaires industriels qui sont BA Systèmes, Autofina et CERI.

Dans ce projet, le premier objectif poursuivi par l'Université du Havre Normandie, en particulier par le laboratoire GREAH de par les travaux de recherches menés dans cette

thèse, est de mettre en oeuvre un scénario de manipulation d'objets dans un atelier flexible, comprenant la saisie des pièces brutes dans un magasin, la navigation autonome du robot au sein de l'atelier et l'alimentation de la machine-outil par les pièces saisies précédemment, le tout réalisé en autonomie, i.e. sans intervention humaine. Le robot utilisé dans ces travaux est un manipulateur mobile qui a été développé au sein du GREAH par l'équipe robotique. Il est constitué d'un bras manipulateur UR5 embarqué sur une plate-forme mobile Summit XL et est équipé de différents capteurs (2 LiDARs, 2 caméras PTZ, 2 caméra RGB-D, 1 capteur d'efforts 6 axes, 1 caméra embarquée en bout de bras,...) afin d'améliorer les critères de performances tels que la précision, la répétabilité et l'autonomie. Le deuxième objectif du GREAH est de participer au développement de solutions robotiques au sein du projet CoRoT en collaboration avec l'ensemble des partenaires. Le GREAH est impliqué particulièrement dans la partie manipulation robotique où il possède une expertise reconnue au niveau régional.

Les principaux objectifs des travaux menés dans cette thèse se déclinent selon les points suivants :

- La navigation autonome du robot dans les locaux de la halle robotique du PIL (Pôle Ingénieur Logistique) de l'ISEL (Institut Supérieur d'Etudes Logistiques) entre la zone du stockage des pièces brutes et la zone où se situe la machine d'usinage.
- La saisie des pièces à usiner via le bras robotique avec un mode de préhension approprié. Ce système de pick-and-place est réalisé grâce à un traitement d'images pour la détection et l'estimation de pose de la pièce. Pour la saisie, nous avons une contribution en intelligence artificielle publiée dans la conférence ETFA 2020. Il s'agit de l'intégration de l'apprentissage automatique au traitement visuel pour avoir un système robotisé capable de détecter l'objet, d'estimer sa pose et de choisir le mode de préhension en fonction des résultats du traitement.
- La modélisation des incertitudes de position et d'orientation du manipulateur mobile et l'estimation des performances en précision du manipulateur mobile en tant que système. Ce travail représente la deuxième contribution scientifique de cette thèse et permet de choisir les capteurs et le mode de commande optimal permettant d'assurer la dépose précise de la pièce dans la machine d'usinage avec une grande fiabilité.

0.1 Organisation du manuscrit

Ce manuscrit est composé de quatre chapitres rendant compte d'une démarche scientifique démarrant de l'état de l'art et aboutissant à des contributions scientifiques, en passant par l'utilisation d'outils d'ingénierie et de recherche dans un environnement complexe, en particulier :

Le premier chapitre présente une étude bibliographique sur les différents manipulateurs mobiles existants en mettant l'accent sur les manipulateurs mobiles à roues. À cette étude nous associons des présentations des éléments robotiques qui constituent les manipulateurs mobiles.

Le second chapitre présente le premier développement du scénario envisagé consistant à déplacer de manière autonome le manipulateur mobile, à saisir de manière autonome les pièces brutes dans le magasin et à déplacer le robot vers la machine d'usinage pour déposer les pièces à usiner sur l'étau.

Le troisième chapitre est consacré à la partie manipulation basée sur l'apprentissage profond. Cette partie contient le traitement d'image pour la perception de l'objet par apprentissage profond et la préhension associée à chaque face afin d'obtenir un picking adaptatif.

Le quatrième chapitre est consacré à la dernière partie de notre scénario, qui est la dépose de la pièce à usiner dans la machine d'usinage. Ce chapitre présente l'étude de l'incertitude de positionnement de la plate-forme robotique afin d'estimer si le manipulateur mobile peut déposer la pièce dans l'étau en respectant les contraintes de positionnement définies à partir du jeu fonctionnel.

Le manuscrit se clôture par des conclusions sur les travaux réalisés et des perspectives sur la suite du travail.

0.2 Publications associées à cette thèse

"An adaptive robotic grasping with a 2-finger gripper based on deep learning network."
Wafae SEBBATA, Mourad A. KENK and Jean-François BRETHÉ. 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). Vol. 1. IEEE, 2020.

Chapitre 1

Les manipulateurs mobiles : État de l'art

Table des matières

1.1	Les manipulateurs mobiles et leurs milieux d'intervention	18
1.1.1	Les manipulateurs mobiles sous-marins	19
1.1.2	Les manipulateurs mobiles aériens	19
1.1.3	Les manipulateurs mobiles spatiaux	20
1.1.4	Les manipulateurs mobiles terrestres	21
1.2	La mécatronique des manipulateurs mobiles à roues	23
1.2.1	La plate-forme mobile « à roues »	24
1.2.2	Les bras manipulateurs	26
1.2.3	Les préhenseurs	30
1.2.4	Les applications des manipulateurs mobiles à roues	31
1.3	Instrumentation des manipulateurs mobiles	31
1.3.1	Capteurs usuels en robotique de manipulation	32
1.3.2	Capteurs usuels en robotique mobile	34
1.3.3	Le traitement d'image	35
1.4	Commande des manipulateurs mobiles	36
1.4.1	Commande des bras manipulateurs	37
1.4.2	Commande des robots mobiles	37

1.4.3 Commande des manipulateurs mobiles 37

Dans ce premier chapitre, nous présentons le contexte général mécatronique et robotique relatif aux travaux de thèse, en particulier une présentation des manipulateurs mobiles en mettant l'accent sur les manipulateurs mobiles à roues, sur les éléments qui les constituent, sur les différents domaines d'utilisation ainsi que sur la commande des parties opératives robotiques.

1.1 Les manipulateurs mobiles et leurs milieux d'intervention

Un manipulateur mobile résulte d'une combinaison de deux fonctionnalités : la manipulation et la locomotion. La manipulation est réalisée via un ou plusieurs bras manipulateurs équipés ou non de préhenseurs et la locomotion est assurée par une plate-forme mobile. Généralement les bras manipulateurs à base fixe sont placés dans des espaces structurés avec un volume de travail limité et réalisent des tâches spécifiques, qui dépendent des degrés de liberté du bras manipulateur, de ses dimensions et des positions atteignables par le préhenseur. L'association de bras robotiques à des bases mobiles élargit le champ d'action de chacun, augmente son espace de travail et ouvre la possibilité à d'autres applications robotiques.

En 2005 La Fondation Nationale pour la Science (NFS) et l'administration nationale de l'aéronautique et de l'espace des États-Unis (NASA) ont défini dans un workshop de la manipulation mobile autonome [2] que la manipulation mobile combine une grande variété de domaines de recherche, allant du contrôle des forces à la conception de mécanismes en passant par la vision par ordinateur. Un an plus tard dans un autre workshop de manipulation mobile autonome [3], cette dernière a été considérée comme une discipline relativement jeune dans le domaine de la robotique.

Les manipulateurs mobiles de par leur diversité de formes ont fait l'objet de recherches approfondies afin de s'adapter à des missions dédiées et à des tâches précises. Ces missions sont caractérisées par des objectifs de nature différente : exploration, surveillance, intervention et manipulation. Les milieux d'exécution de ces travaux diffèrent ; ils peuvent

être terrestres, aquatiques, aériens ou spatiaux. La manipulation peut s'effectuer avec un ou plusieurs bras. Dans ce manuscrit l'étude est focalisée sur les manipulateurs mobiles avec un seul bras robotique. Nous pouvons distinguer quatre grandes familles de manipulateurs mobiles suivant le milieu dans lequel ils évoluent : sous-marin, aérien, spatial et terrestre.

1.1.1 Les manipulateurs mobiles sous-marins

Les manipulateurs mobiles sous-marins sont des engins sous-marins équipés de bras manipulateurs robotisés, capable d'accéder aux zones maritimes non accessibles aux plongeurs pour différentes applications, telles que le prélèvement des sédiments ou des espèces vivantes, la récupération des artefacts archéologiques, la pose des câbles de communication etc. Il existe trois types de manipulateurs mobiles sous-marins : les engins habités, les engins télé-opérés et les engins autonomes [4]. Les habités sont conçus pour plonger dans l'eau en embarquant des passagers et ils sont munis de bras robotiques pour interagir avec l'environnement comme par exemple le sous-marin habité présenté sur la figure 1.1(a). Les robots télé-opérés sont généralement reliés par un câble à un navire pour leur pilotage et leur alimentation. Ils sont destinés à des zones maritimes non accessibles aux plongeurs pour des tâches de prélèvement, de manipulation, de mesure et d'acquisition de données, comme par exemple l'engin télé-opéré de la figure 1.1 (b). Les engins sous-marins autonomes accomplissent la même diversité de tâches que les engins habités ou télé-opérés à l'exception qu'il n'y a pas d'intervention humaine. Un sous-marin développé par l'université de Gérone sur la figure 1.1(c) est utilisé pour des expérimentations sur la manipulation sous-marine.

1.1.2 Les manipulateurs mobiles aériens

Les manipulateurs mobiles aériens sont des bras manipulateurs montés sur des drones ou UAV (Unmanned Aerial Vehicles) [5]. Ce type de manipulateur a ouvert la voie à d'autres applications grâce à leur capacité d'atteindre des zones hors de portée des robots terrestres. Le domaine de la manipulation aérienne innove pour développer des manipulateurs mobiles capables d'effectuer un ensemble de tâches telles que l'assemblage et la construction de structures, la préhension et le transport, le pick-and-place ou l'inspection et la maintenance des installations. Différentes architectures de manipulateurs mobiles aériens ont été déve-

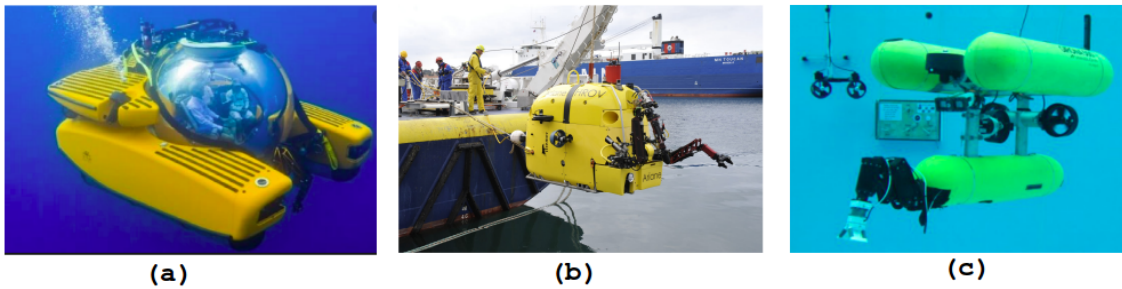


FIGURE 1.1: Les manipulateurs mobiles sous-marins : (a) Le sous-marin TRITON 3300/3 MKII commercialisé par l'entreprise TRITON, (b) le sous-marin H-ROV Ariane de l'Ifremer, (c) le sous-marin Girona 500 de l'Université de Gérone.

loppées, la figure 1.2 en présente quelques-unes comme le manipulateur aérien à un seul bras articulé (a), à bras multiples, typiquement deux (b) et un manipulateur parallèle (c).

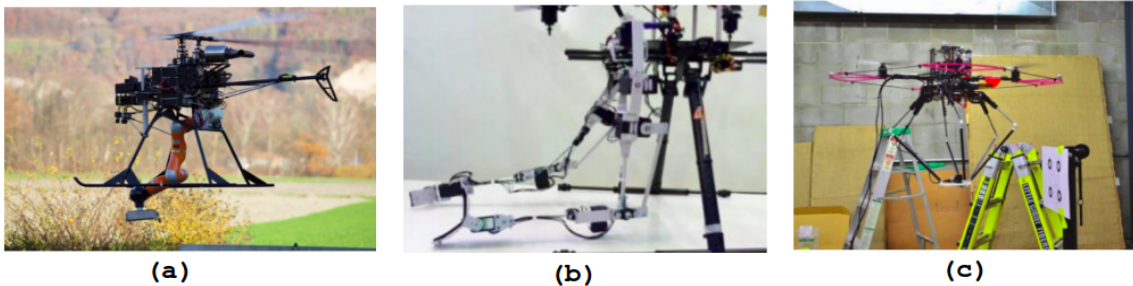


FIGURE 1.2: Des exemples de manipulateurs mobiles aériens : (a) un véhicule aérien sans pilote équipé par un bras industriel de 7 DDL [6], (b) deux bras de 5 DDL [7], (c) un manipulateur parallèle [8].

1.1.3 Les manipulateurs mobiles spatiaux

Les manipulateurs mobiles spatiaux sont des robots conçus pour l'exploration spatiale. La navette spatiale américaine conçue par la NASA présentée sur la figure 1.3.a) est le premier robot dédié à cette fonction. Elle est composée d'un orbiteur équipé d'un bras, d'un réservoir et de propulseurs. Elle est capable de transporter de gros satellites depuis

et vers l'orbite terrestre [9]. D'autres manipulateurs mobiles spatiaux, les rovers [10][11] sont dédiés aux missions sur Mars comme les robots Spirit, Opportunity et Curiosity (figure 1.3.b). Ils possèdent une plate-forme mobile à 6 roues afin d'atterrir et se déplacer ainsi qu'un ou plusieurs bras manipulateurs pour saisir des échantillons du sol de Mars et les analyser avec leurs instruments embarqués.

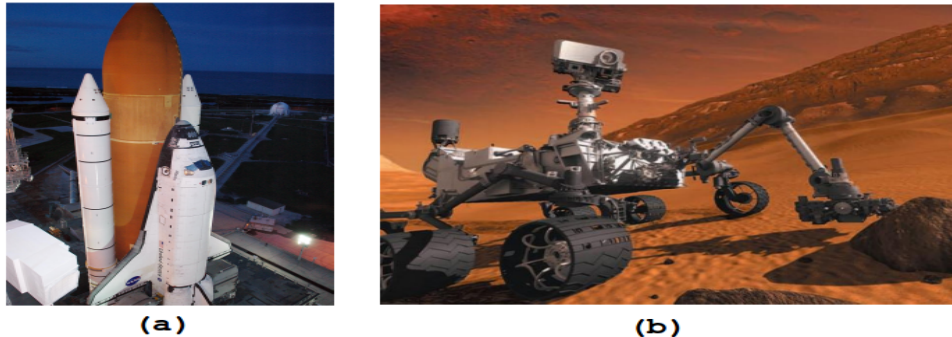


FIGURE 1.3: Les manipulateurs mobiles spatiaux : (a) Navette spatiale américaine, (b) robot Curiosity.

1.1.4 Les manipulateurs mobiles terrestres

Les manipulateurs mobiles multipèdes

Les manipulateurs mobiles multipèdes possèdent une ou plusieurs pattes et ont une capacité de locomotion basée sur la marche. Ils disposent aussi de capacités de manipulation et d'interaction avec l'homme à l'aide de dispositifs de perception de l'environnement. Les humanoïdes sont les plus populaires. Ces derniers sont des robots anthropomorphes qui présentent des fonctionnalités inspirées de celles des humains. Généralement un robot multipède peut se mouvoir dans des environnements moins structurés tel qu'un terrain rocheux s'il arrive à déterminer les zones accessibles et qu'il est équipé des pattes adaptées. Ces manipulateurs mobiles sont largement étudiés dans la littérature ainsi que la recherche qui est toujours active à ce jour avec ce type de robot. Ce livre [12] est une revue récente des robots humanoïdes. De nombreux manipulateurs mobiles multipèdes ont été conçus pour différentes applications ; comme des applications de service en éducation, en assistance aux

personnes âgées, en milieu médical, pour la recherche scientifique et les compétitions robotiques. Bien que de nombreux succès aient été obtenus par des sociétés de développement des manipulateurs multipèdes, en particulier par l'entreprise Boston Dynamics : l'humanoïde ATLAS illustré dans la figure 1.4.(a); le robot SPOT dans la figure 1.4.(b) explore avec quatre pattes des environnements difficiles d'accès ou accidentés et peut même monter des escaliers.



(a)



(b)

FIGURE 1.4: Des exemples de robots multipèdes : l'humanoïde ATLAS, le robot à quatre pattes SPOT.

Les manipulateurs mobiles à roues

Les manipulateurs mobiles à roues sont des véhicules à guidage automatique équipés d'un ou plusieurs bras manipulateurs. Ils permettent un déplacement par roulement (avec friction au contact entre les roues et le sol) et une manipulation robotique. Les manipulateurs mobiles de ce type sont les plus utilisés au vue de leurs avantages par rapport aux autres manipulateurs mobiles basés sur les autres types de locomotion. Ils possèdent l'avantage d'avoir une bonne adéquation à différents types de terrains, d'une facilité de déplacement sur les sols relativement plats même à grande vitesse tout en restant économique pour la consommation énergétique. Ils présentent une simplicité de la structure mécanique et de la commande, et une bonne adaptation au monde industriel. Ce type de manipulateur mobile sur lequel s'appuie notre travail est étudié en détail dans la deuxième partie de ce chapitre.

Les manipulateurs mobiles à plate-forme hybride «roues-pattes»

Les plates-formes mobiles hybrides résultent d'une combinaison de deux types de locomotion, multipède et roues. Cette combinaison consiste à utiliser alternativement les deux modes de locomotion (les pattes et les roues) et dans cette optique, chaque patte du robot mobile contient aussi une roue. Cette base mobile permet une locomotion avec un point de contact irrégulier et discontinu là où le robot traditionnel à roues peut avoir des difficultés à circuler. Des manipulateurs mobiles avec ce mécanisme ont été inventés et d'autres de même concept sont en cours de développement [13] [14] [15]. HANDLE est un exemple de manipulateur mobile avec une plate-forme hybride (voir figure 1.5) de l'entreprise Boston Dynamics. La deuxième version de ce robot a été lancée en 2019 et est dédiée à la palettisation ou/et la dépalettisation et le transport des cartons [16].



FIGURE 1.5: Un exemple de manipulateur mobile hybride, le robot HANDLE [16].

1.2 La mécatronique des manipulateurs mobiles à roues

L'invention du premier manipulateur mobile à roues remonte à 1984 avec le robot MORO (MObiler ROboter) [17]. Le robot MORO était capable de manipuler et de déplacer des pièces de travail en naviguant librement dans un atelier. Les manipulateurs sur une plate-forme mobile à roues résolvent certaines problématiques rencontrées en manipulation robotique classique sur des plates-formes fixes telles que les singularités, l'espace de travail limité, le contournement d'obstacles etc. Plusieurs manipulateurs mobiles industriels sont

apparus sur le marché comme H2BIS-GT6A [18], YAMABICO [19], DENSO [20] et Neobotix MM- KR16 [21], etc. De nombreuses recherches ont été menées sur cette famille de robots : la modélisation et la commande font partie des premières thématiques étudiées et sont toujours présentes au coeur de la recherche, en tenant compte du type de base mobile (holonome, non holonome) ainsi que des degrés de liberté du bras robotique [22] [23]. Dans la littérature, la commande des manipulateurs mobiles par découplage des tâches de manipulation et de locomotion est largement étudiée, ainsi que les performances de stabilité et de précision des manipulateurs mobiles [24][25][26].

Une deuxième vague de manipulateurs mobiles a été mise sur le marché avec l'apparition de la notion de collaboration homme-robot. Le principe des robots collaboratifs repose sur de nouvelles normes sécuritaires dans le domaine industriel, en particulier les normes NF ISO 10218-1 2 (2011) qui expriment les exigences de sécurité pour les robots industriels (Robots et dispositifs robotiques), les spécificités techniques liées à la norme NF ISO/TS 15066 (robots et systèmes robotiques / robots collaboratifs), la directive européenne "Machine" 2006/42/CE et la nouvelle norme ISO/TC 199/WG12 qui définit les interactions homme-machine. Parmi les manipulateurs mobiles collaboratifs, nous pouvons citer : KUKA OmniRob [27], XL-GEN [28], RB-1 [29]. D'autres plates-formes ont été assemblés dans des laboratoires de recherche comme celle du robot utilisé dans notre travail. La notion de manipulateurs mobiles collaboratifs [30] a entraîné de nouvelles recherches comme la collaboration homme-robot, la collaboration et la coopération robot-robot et homme [31][32].

1.2.1 La plate-forme mobile « à roues »

La plate-forme mobile représente l'élément assurant le déplacement du manipulateur mobile d'un emplacement vers un autre. Dans cette partie nous étudions la base mobile terrestre à roues sur laquelle le travail de cette thèse s'est focalisé. Il existe une variété de formes de base mobile avec différentes dispositions des roues. Elles peuvent être classées selon deux grandes classes : non-holonomes et holonomes :

Les bases mobiles non-holonomes

Les plates-formes mobiles non-holonomes disposent de roues ordinaires, dont la locomotion possède deux degrés de mobilité; une translation suivant l'axe X et une rotation suivant le Z. Ce type de plate-forme conduit à des contraintes cinématiques où les chemins admissibles doivent vérifier une contrainte liée au déplacement.

Nous distinguons trois sous-classes de plates-formes mobiles non-holonomes : les plates-formes unicycles, tricycles et les voitures robot. Les unicycles sont actionnées par deux roues indépendantes. Ils possèdent aussi des roues folles pour assurer la stabilité de la plate-forme. Leurs centres de rotation se situent sur l'axe reliant les deux roues motrices. HERB [33] dans la figure 1.6(a) est un exemple d'un manipulateur mobile où la base mobile est unicycle. Les tricycles sont constitués de deux roues fixes placées sur le même axe dépendantes en vitesse et d'une roue centrée orientable placée sur l'axe longitudinal qui produit une vitesse angulaire dans le plan. Les plates-formes type voiture sont semblables aux tricycles du point de vue cinématique, la différence est au niveau du train avant. Elles possèdent deux roues orientables dépendantes au lieu d'une seule ce qui garantit une meilleure stabilité. Comme exemple de ce type, nous pouvons citer le robot XL-GEN dans la figure 1.6(b). Nous pouvons inclure dans cette catégorie, les plates-formes mobiles à chenille. Ce type de base mobile assure des multiples contacts avec le sol ce qui permet une répartition large des contraintes normales et tangentiels. Les bases mobiles à chenille sont utilisées lorsque le terrain présente des dénivelés importants. Ils existent surtout dans le domaine de la défense militaire comme le robot LAND Robot TALON IV [34] présenté sur la figure 1.6(c).

Les bases mobiles holonomes

Les plates-formes mobiles holonomes contiennent des roues omnidirectionnelles. Ces roues sont capables de se déplacer dans toutes les directions indépendamment et de tourner simultanément c'est-à-dire elles permettent une locomotion avec trois degrés de mobilité : deux translations suivant l'axe X et Y ainsi qu'une rotation suivant le Z. Les plates-formes mobiles holonomes peuvent se composer de trois roues décentrées orientables, placées en triangle équilatéral tel que le robot Pepper [35] ou de quatre roues type voiture comme le robot OmniRob [27] de la figure 1.6(d). La plate-forme robotique utilisée durant ces travaux

de thèse est un manipulateur mobile holonome Summit XL de la société Robotnik.

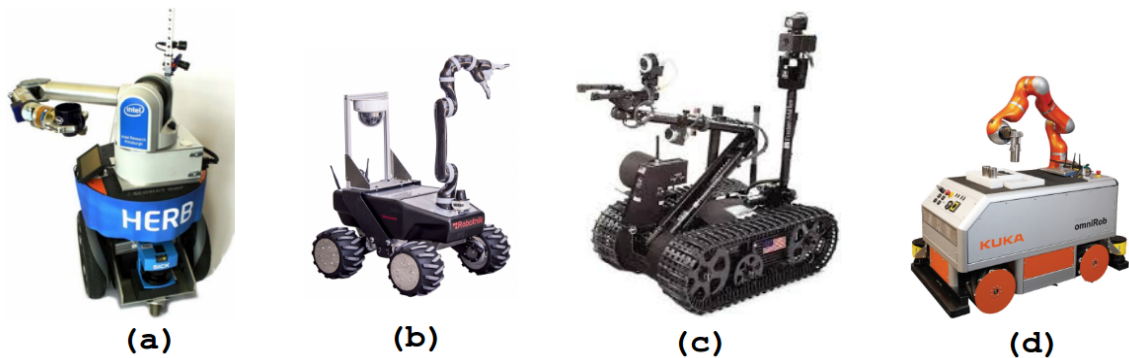


FIGURE 1.6: .

Des manipulateurs mobiles avec différents types de base mobile :(a) HERB [33], (b) XL-GEN, (c) LAND Robot TALON IV [34], (d) OmniRob [27]

1.2.2 Les bras manipulateurs

Le bras robotique est l'élément assurant la manipulation d'un ou plusieurs objets. Selon la définition de l'organisme RIA « Robot Institute of America », un bras robotique est un manipulateur multifonctions reprogrammable conçu pour déplacer des matériaux, des pièces, des outils ou des dispositifs spécialisés à travers des mouvements programmables variables pour l'exécution de diverses tâches. Pour assurer ces fonctions dans différentes applications, il existe plusieurs types de bras robotiques qui peuvent être classés selon différents critères tels que leurs architectures, leurs capacités de charge, leurs types d'utilisation ou autres. Nous présentons les manipulateurs suivant leur architecture cinématique : sériele, parallèle et hybride.

Les bras manipulateurs séries

Les bras manipulateurs à architecture série sont des robots avec une chaîne cinématique ouverte, composés d'une succession de corps rigides connectés par des articulations formant des liaisons (pivot ou glissière). Une chaîne cinématique est dite ouverte lorsqu'il n'y a qu'une seule séquence de liaisons reliant les deux extrémités de la chaîne. Dans une

chaîne cinématique ouverte, chaque articulation fournit à la structure un degré de mobilité. Une articulation avec une liaison glissière crée un mouvement de translation relatif entre les deux corps rigides liés à l'articulation, tandis qu'avec une liaison pivot, est créé un mouvement de rotation relatif entre les deux corps [36]. Selon les liaisons et leurs séquences partant de l'articulation de base du bras robotique jusqu'à l'organe terminal, nous avons les cinq classes usuelles des manipulateurs : cartésien, cylindrique, sphérique, SCARA et anthropomorphique [37].

- **Manipulateur cartésien** : Les manipulateurs cartésiens sont composés uniquement d'articulations à liaison glissière avec des axes typiquement orthogonaux entre eux dans l'espace. Cette structure permet au bras robotique cartésien d'avoir un espace de travail en forme de parallélépipède rectangle (figure 1.7(a)).
- **Manipulateur cylindrique** : Les manipulateurs cylindriques diffèrent des cartésiens en ce que la première articulation prismatique est remplacée par une liaison pivot, ce qui donne un espace de travail en forme de cylindre creux. La deuxième articulation avec une liaison glissière horizontale permet à l'organe terminal d'accéder à des cavités horizontales (figure 1.7(b)).
- **Manipulateur sphérique** : Les manipulateurs sphériques diffèrent des cylindriques en ce que leur seconde articulation prismatique est remplacée par une articulation pivot, ce qui donne un espace de travail en forme de portion de sphère creuse (figure 1.7(c)).
- **Manipulateur SCARA** : L'acronyme SCARA (Selective Compliance Assembly Robot Arm) signifie bras robotisé d'assemblage à compliance sélective. Il s'agit d'un manipulateur sphérique, obtenu en combinant trois liaisons pivot parallèles et une articulation glissière typiquement d'axe vertical de telle sorte que tous les axes de mouvement soient parallèles. Le volume de travail devient cylindrique (figure 1.7(d)).
- **Manipulateur anthropomorphique** : Les manipulateurs anthropomorphes n'utilisent que des articulations à liaison pivot. L'espace de travail est approximativement une partie d'une sphère. Plusieurs manipulateurs anthropomorphiques ont été inventés, certains destinés à l'industrie et d'autres qui appartiennent à la nouvelle généra-

tion des bras manipulateurs collaboratifs tel que le bras robotique Kuka de la figure 1.7(e).

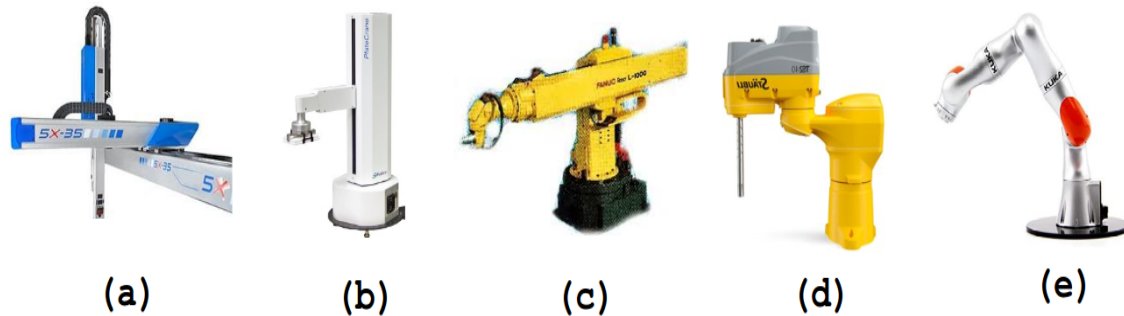


FIGURE 1.7: .

Quelques bras manipulateurs sériels : (a) Robot 5X Line, (b) PlateCrane EX, (c) Fanuc I-1000, (d) Staubli TS2-40, (e) Kuka

Les bras manipulateurs parallèles

Un manipulateur parallèle est un mécanisme à chaîne cinématique fermée où l'effecteur est relié à une base fixe par au moins deux chaînes cinématiques indépendantes. Ils sont constitués généralement d'une base fixe connectée à une base mobile par des pattes qui sont motorisées chacune par un actionneur [38]. Les robots parallèles ont une rigidité structurale élevée par rapport aux robots sériels, puisque la plate-forme mobile est supportée par plusieurs chaînes cinématiques en même temps ce qui permet une répartition des efforts appliqués. L'espace de travail de ce type de robot est limité. Les manipulateurs parallèles peuvent être classés selon deux grandes familles : les manipulateurs entièrement parallèles et les manipulateurs parallèles à câble.

- **Manipulateur entièrement parallèle** : La chaîne cinématique est strictement égale au nombre de degrés de liberté de la plate-forme mobile. Chaque chaîne cinématique comporte un seul actionneur. La plate-forme de Gough-Stewart illustrée sur la figure 1.8(a) est l'une des architectures connues de manipulateurs dites entièrement parallèles, composée de six pattes. Chacune de ces pattes est actionnée par un actionneur actif, connectée aux plates-formes à l'aide de joints sphériques passifs. Ce robot a

été utilisé principalement comme simulateur de vol [39]. Le robot Delta inventé par le Pr Clavel de l'EPFL est bien connu pour ses performances élevées dans les tâches de pick and place, et son architecture se base sur le principe de parallélogrammes spatiaux. Il est composé de trois bras reliés à des joints universels à la base, comme par exemple le Delta IRB 360 FlexPicker de ABB (figure 1.8(b)).

- **Manipulateur parallèle à câbles :** Ces manipulateurs utilisent des câbles pour la transmission du mouvement de la base fixe vers l'organe terminal. Ce type de manipulateur fait l'objet de nombreuses recherches telles que le contrôle des câbles reliés à la plate-forme par des robots mobiles (figure 1.8(c)). Nous citons quelques références des travaux récents à ce propos [40], [41].

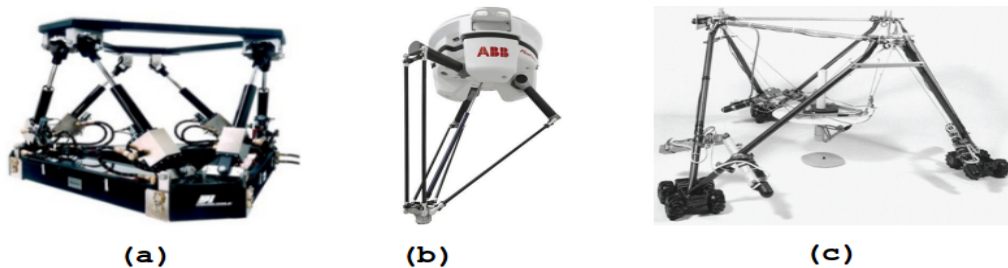


FIGURE 1.8: .

Quelques exemples de bras manipulateurs parallèles : (a) La plate-forme de Gough-Stewart, (b) Delta IRB 360 FlexPicker, (c) NIST RoboCrane

Les bras manipulateurs hybrides

Les manipulateurs séries-parallèles ou dits hybrides sont une combinaison entre les deux types de bras manipulateurs (série et parallèle). Cette combinaison permet aux bras hybrides de bénéficier des avantages des deux architectures : la rigidité des manipulateurs parallèles et le large espace de travail des manipulateurs séries. Plusieurs travaux ont été menés et d'autres sont en cours sur les manipulateurs hybrides. Le Tricept, présenté dans la figure 1.9(a) est l'un des manipulateurs hybrides les plus réussis. Il est utilisé dans différentes applications comme le perçage, l'assemblage robotique, le fraisage de métaux, etc. La figure 1.9(b) montre le robot Logabex-LX4, conçu par une mise en série de plusieurs plate-formes parallèles [42].

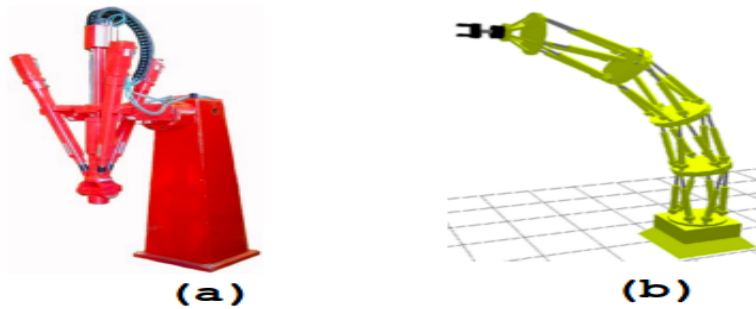


FIGURE 1.9: .

Des manipulateurs hybrides : (a) Tricept, (b) la CAO de Logabex-LX4 [43]

Parmi les trois architectures de manipulateurs, l'architecture sérielle est la plus utilisée pour la composition des manipulateurs mobiles ainsi que pour la construction de notre plateforme robotique car elle offre une dextérité importante par rapport aux autres.

1.2.3 Les préhenseurs

La manipulation des objets avec un seul bras robotique oblige à intégrer un préhenseur, i.e. un outil qui relie le bras robotique et l'objet à manipuler. Il existe une grande variété de technologie de préhenseurs qui répondent aux besoins de saisie d'objets variés avec différents critères (géométriques, textures, matériaux, poids et autres). Les préhenseurs peuvent être classés selon différents aspects [44].

Selon le type d'actionnement [44] : les préhenseurs peuvent être classés suivant les phénomènes physiques en jeu (préhenseurs à vide, magnétiques,...) et les sources d'énergie primaire utilisées (hydraulique, pneumatique et électrique). Les préhenseurs à vide fonctionnent selon le principe de Bernoulli pour produire une dépression à partir d'un flux d'air à grande vitesse entre le préhenseur et la surface de l'objet. Les préhenseurs pneumatiques sont basés sur la pression de l'air actionnant une pince pour la saisie des objets. Les préhenseurs hydrauliques exercent des efforts importants grâce aux hautes pressions obtenues par compression du fluide. Les préhenseurs électriques sont actionnés électriquement et peuvent être facilement et précisément contrôlés.

Dans notre étude nous nous intéressons uniquement aux préhenseurs à deux doigts ali-

mentés électriquement. La pince Robotiq utilisée peut saisir un objet par fermeture ou ouverture des doigts.

1.2.4 Les applications des manipulateurs mobiles à roues

Les manipulateurs mobiles à roues peuvent être employés dans différents domaines d'applications. Ils se répartissent en trois catégories [45] : les applications de logistique, d'assistance et de service.

- **Les manipulateurs mobiles à roues en logistique ou infralogistique** : Dans cette catégorie, les applications les plus appropriées sont : le pick-and-place des objets, la palettisation/ dépalettisation, la charge/décharge des pièces dans des machines d'usinages ou convoyeurs et le transport. Un exemple de scénario type a été développé par des chercheurs de l'Université d'Aalborg [46], avec le manipulateur mobile LH6. Leur travail consistait à introduire un module dans un plateau contenant du PCB. Ce travail a nécessité une saisie du module, un déplacement pour se rendre au plateau et le dépôt de la carte sur le convoyeur.
- **Les manipulateurs mobiles à roues d'assistance** : Cette catégorie englobe les applications de chargement dans les machines d'usinage, d'assemblage dans la production manufacturière à grande échelle et d'inspection. A titre d'exemple un assemblage multi-échelle de structures complexes réalisé par des multi-manipulateurs à roues est présenté dans cet article[47].
- **Les manipulateurs mobiles à roues de service** : Cette dernière catégorie est liée à la maintenance, la réparation et la révision.

Le choix du manipulateur mobile dépend de la fonction attribuée au bras manipulateur, du terrain de navigation pour la base mobile et du milieu d'exécution de la tâche pour le robot complet.

1.3 Instrumentation des manipulateurs mobiles

L'exécution autonome des différentes fonctions robotiques comme la localisation, la navigation, la manipulation d'objets, l'interaction sociale avec les humains, etc nécessitent

de connaître l'état interne du robot et de percevoir son environnement. Pour cela le robot doit être équipé par un ensemble de capteurs proprioceptifs et extéroceptifs. Nous allons maintenant présenter les capteurs rencontrés usuellement en robotique de manipulation et en robotique mobile.

1.3.1 Capteurs usuels en robotique de manipulation

La perception de l'environnement lors de la manipulation robotique consiste à extraire les informations nécessaires sur l'objet et son environnement. En particulier, il s'agit de détecter la présence de l'objet dans la scène et d'estimer sa pose par rapport au robot. À partir de ces informations extraites et des données sur les positions des articulations du bras robotique, ce dernier peut réagir pour saisir l'objet convenablement.

La perception proprioceptive en manipulation robotique peut être réalisée à travers différents capteurs comme les potentiomètres, les capteurs de contacts etc. Ce type de capteurs mesure des paramètres internes du bras robotique tels que les positions et les vitesses relatives aux articulations. A partir de ces paramètres mesurés, l'unité de commande peut suivre ou modifier la trajectoire en cours, afin qu'elle soit conforme à celle exigée.

La solution principale en perception extéroceptive pour la manipulation robotique est la vision. Depuis les années 70 jusqu'à ce jour le retour visuel est considéré comme une source importante d'informations sur les objets à manipuler. Une première étude sur l'importance de l'intégration de l'intelligence artificielle dans les applications robotique, notamment la vision dans la manipulation robotique, remonte à 1973 [48]. Depuis cette époque, différents algorithmes de traitement d'images pour l'acquisition des données 3D, avec différents types de caméras, ont été développés.

Nous distinguons les caméras RGB, les caméra RGB-D et les caméras stéréos :
Les caméras RGB ou dites monoculaires sont des dispositifs qui fournissent des données sur une scène en estimant l'intensité de trois couleurs de base rouge, vert et bleu, sur une échelle de zéro à une valeur maximale de 255. Les images peuvent être traitées à l'aide de bibliothèques informatiques. Les paramètres de choix des caméras RGB sont : la largeur du champ de vision, c'est-à-dire la longueur que l'objectif couvre à une certaine distance qui peut être mesurée horizontalement, verticalement ou en diagonale ; la distance focale, qui

est la distance, entre le point où les rayons lumineux convergent dans un objectif et le capteur numérique ; l'angle de vue, qui est une mesure du point de vue de l'objectif. Parmi les marques des caméras RGB qui existent nous citons la caméra Basler (figure 1.10(a) utilisée dans les travaux de manipulation robotique [49].

Les caméras RGB-D sont des dispositifs qui fournissent simultanément les données couleurs et les données de profondeur caractérisant les distances des objets présents dans une scène vue dans l'image [50]. Ces caméras se composent d'un émetteur laser infrarouge qui projette un motif sur son environnement, et d'un récepteur infrarouge qui capture le motif projeté. Les données obtenues par l'infrarouge permettent l'estimation de la distance de chaque pixel de l'image par rapport au capteur. En 2010, Microsoft a lancé la première caméra RGB-D Kinect [50], puis d'autres capteurs sont apparus sur le marché, comme la caméra Astra utilisée dans notre développement 1.10.b qui est caractérisée par un champ de vision horizontal de 60 degrés et vertical de 49.5 degrés et capable de détecter de 0.6 m jusqu'à 8 m. L'émergence de ce type de caméra a entraîné de nouveaux travaux sur des problématiques comme la reconnaissance d'objets dans des conditions lumineuses complexes, la reconstruction 3D, etc.

Les caméras stéréo sont composées de deux lentilles avec un capteur d'images distinct pour chacune, ce qui leur permet de simuler la vision binoculaire humaine et de capturer des images tridimensionnelles. Le fonctionnement des capteurs stéréo consiste à identifier les pixels de l'image qui correspondent à un même point dans une scène observée par plusieurs caméras en utilisant la méthode de triangulation.

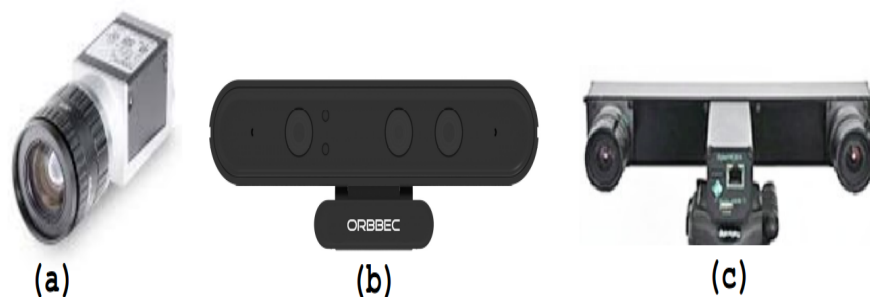


FIGURE 1.10: Exemple des types de capteurs caméras : (a) Basler (RGB), (b) Astra Orbbec (RGB-D), (c) Stereo

1.3.2 Capteurs usuels en robotique mobile

Dans la robotique mobile, le robot doit déterminer son déplacement relatif et percevoir l'environnement qui l'entoure afin de se localiser, de détecter les obstacles à éviter et assurer la navigation autonome.

Dans la perception proprioceptive en robotique mobile, le codeur incrémental est le plus communément utilisé, qui permet de mesurer la position et la rotation de la roue. D'autres capteurs inertiels utilisés dans le même contexte comme l'accéléromètre et le gyroscope.

La perception extéroceptive en robotique mobile est réalisée soit par des caméras comme dans le cas de la robotique de manipulation soit par les capteurs de distance. Généralement, les LIDARs sont les capteurs de distance les plus présents sur les robots mobiles. Leur principe de fonctionnement repose sur l'émission d'un nuage de rayons laser dans l'environnement et lors de la présence d'une cible le signal sera réfléchi vers le récepteur. La distance entre le LIDAR et la cible est déterminée en mesurant le temps pris par une impulsion laser pour parcourir l'aller-retour. Ce capteur est utilisé, avec une haute résolution temporelle et spatiale et une grande efficacité de mesure en raison des longueurs d'onde, pour réaliser des cartographies, de l'auto-localisation ainsi que la détection des obstacles lors de la navigation autonome. Il existe de nombreux modèles LiDARs, comme Hokuyo utilisé dans notre développement, Velodyne et RoboSense, tous trois illustrés dans la figure 1.11.

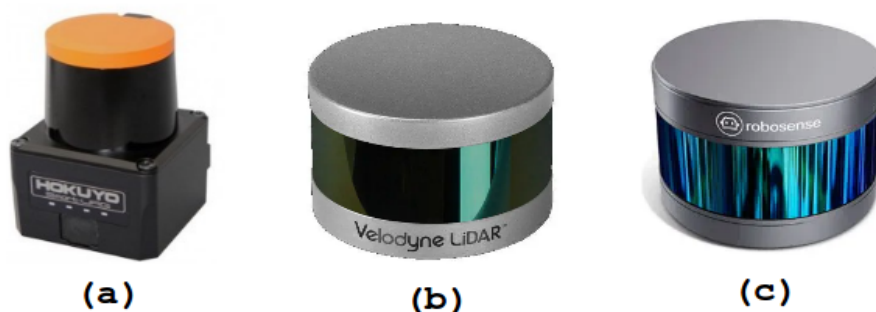


FIGURE 1.11: Exemple des lidars : (a) Hokuyo, (b) Velodyne, (c) RoboSense

1.3.3 Le traitement d'image

S'il existe de nombreux algorithmes de vision développés pour différentes applications robotiques, nous choisissons de parler dans ce manuscrit de trois algorithmes de traitement visuel qui ont été intégrés dans la manipulation robotique :

- Extraction de points d'intérêts : Ces algorithmes ont pour but de réduire les informations des éléments dans l'image en des sous-ensembles de points. Généralement ces algorithmes sont basés sur deux outils ; les détecteurs et les vecteurs descripteurs. Les détecteurs permettent une répartition en zones de points d'intérêt dans l'image et les vecteurs descripteurs analysent et décrivent les zones obtenues par les détecteurs. Une zone de points d'intérêt est une zone de pixels désignant une homogénéité soit en couleurs, en forme ou autres propriétés. Les deux techniques classiques à base d'extraction de points d'intérêt sont SIFT et SURF. SIFT « Scale-Invariant Feature Transform » apparus en 1999 [51] [52]. Il s'agit de détecter des points significatifs qui amènent à identifier un objet, à partir d'une représentation d'image multi-échelle. La détection de ces points donne lieu à la mise en place des vecteurs descripteurs associés à chacun d'eux. Ces vecteurs sont ensuite comparés pour déterminer s'il y a une correspondance entre le modèle et l'image où la recherche de l'objet doit être effectuée. L'approche SURF « Speeded-Up Robust Features » [53] est considérée comme une version accélérée de l'algorithme SIFT. Parmi les applications récentes des algorithmes d'extraction de points d'intérêt pour la détection, la reconnaissance et la localisation des objets dans la manipulation robotique, nous pouvons citer ces deux références [54] [55].
- Template matching : Cette approche consiste à trouver les objets dans l'image à partir de leurs modèles de référence. Le fonctionnement des algorithmes basés sur le template matching implique de déplacer l'image modèle vers toutes les positions possibles dans une image d'entrée et de calculer un index numérique qui indique dans quelle mesure le modèle correspond à l'image dans cette position. Deux références incontournables pour ces méthodes de template matching sont [56] [57]. Dans l'article [58] nous trouvons un exemple d'application de l'un des algorithmes de template matching pour détecter et reconnaître une canette de boisson à saisir par

le robot parmi les 100 types présents en stock.

Dans notre cas d'études, la pièce manufacturée est de couleur unie et ne contient pas de texture, ce qui rend les méthodes basées sur l'extraction de points d'intérêt ou de template matching inefficace pour détecter et distinguer les différentes faces de l'objet.

- Marqueur code : Les algorithmes basés sur les marqueurs code se caractérisent par un identifiant unique encodé dans leur structure suivant un système de modèle pré-défini composé de formes en noir et blanc. Ils fournissent un moyen de localisation des objets, des robots et autres. Ils sont utilisés dans de nombreuses applications en robotique, notamment la manipulation. Plusieurs systèmes de marqueurs ont été proposés. Si MAtrix [59] et ARToolKit [60] furent les premiers utilisés, d'autres systèmes de marqueurs de forme quadratique tels que Artag [61], AprilTag [62] et Aruco [63] ou circulaire comme le WhyCon [64] ont été inventés. Les algorithmes de détection de ces marqueurs se basent sur la théorie du codage numérique pour trouver leurs emplacements.

Les dernières études de détection d'objet et d'estimation de pose sont plus orientées vers l'apprentissage automatique avec des réseaux de neurones convolutifs (CNNs). Cela est rendu possible grâce à la présence de processeurs graphiques pour supporter la puissance de calcul et la disponibilité de grandes bases d'images annotées comme ImageNet et COCO. Ces avancées, au niveau matériels et logiciels, ont contribué au développement rapide de réseaux de neurones de plus en plus complexes qui établissent une nouvelle littérature appelée apprentissage profond ou deep learning.

1.4 Commande des manipulateurs mobiles

Après l'étape de la perception de l'environnement par les capteurs embarqués sur le robot et à partir de la modélisation des robots qui peut être représentée par un modèle cinématique, dynamique ou une combinaison des deux, la plate-forme robotique sera commandée soit cinématiquement (vitesses ou accélérations) soit dynamiquement (en couples ou forces) [65] afin de suivre la trajectoire générée et atteindre la position désirée.

1.4.1 Commande des bras manipulateurs

L'objectif de la commande des bras manipulateurs est de contrôler les articulations qui les constituent afin qu'elles suivent es chemins prédéfinis pour atteindre la position finale tout en respectant les contraintes cinématiques liées au mouvement. Les différentes structures mécaniques des bras robotiques qui existent comme montré dans la section 1.2 et les différentes applications sur lesquelles elles sont employées expliquent la variété des techniques de commande développées pour les manipulateurs, telle que la commande jacobienne, robuste, par apprentissage, etc. Cela a fait l'objet de plusieurs recherches scientifiques dans différents contextes. Parmi les travaux récents, nous citons ces trois références : [66] [67] [68].

1.4.2 Commande des robots mobiles

Il existe plusieurs techniques de commande des plate-formes mobiles comme la commande par mode glissant, par synthèse de Lyapunov, basée sur le Backstepping, etc. Un ensemble de paramètres est mis en jeu dans le choix de la technique de commande en fonction du type de base mobile, du milieu de navigation, des fonctionnalités attendues etc. Cela a attiré l'attention de beaucoup de recherches parmi lesquelles nous citons quelques travaux récents [69] [70] [71].

1.4.3 Commande des manipulateurs mobiles

La commande d'un manipulateur mobile nécessite une gestion des deux systèmes mécaniques conçus de manière distincte et réagissant différemment aux influences extérieures. Dans ce type de plate-forme la notion de redondance apparaît, ce qui amène à une infinité de configurations du système pour une situation donnée, amenant à concevoir des modes de commande innovants [72] [73] [74]. Dans notre cas d'étude, le travail ne couvre pas cette problématique scientifique et l'exécution des fonctions de manipulation ou de navigation s'effectue indépendamment.

Chapitre 2

Alimentation robotisée autonome d'une machine-outil

Table des matières

2.1	Le projet CoRoT	40
2.1.1	Le contexte du projet	40
2.1.2	Les finalités du projet	41
2.1.3	Les livrables du projet	41
2.2	Le cas d'étude GREAH	42
2.2.1	Le scénario	42
2.2.2	La plate-forme robotique utilisée	43
2.2.3	Principes de navigation d'un manipulateur mobile	45
2.2.4	Principes de manipulation robotique	54
2.3	Mise en oeuvre du cas d'étude	58
2.3.1	Navigation du manipulateur mobile	58
2.3.2	La saisie et la dépose des pièces par le manipulateur mobile	58
2.3.3	La communication avec la machine d'usinage	60

Dans ce chapitre nous présentons le projet CoRoT dans lequel s'inscrivent les travaux

de cette thèse, le cas d'étude développé et les travaux applicatifs réalisés.

2.1 Le projet CoRoT

2.1.1 Le contexte du projet

Le projet COROT a pour objectif de remédier à la baisse de productivité moyenne des PME du secteur logistique en France et au Royaume Uni, observée depuis 2008. Selon les rapports récents sur les tendances manufacturières (programme H2020-FoF, UK FFM 2013, NFI 2014, stratégie de spécialisation intelligente - RIS3), la mise en place de plates-formes robotiques innovantes, flexibles et avec des fonctionnalités différentes, participe à l'accroissement de la productivité.

La manipulation mobile robotique visant à robotiser les tâches de manipulation et de navigation dans les entrepôts logistiques est l'un des défis à relever dans le contexte de l'Industrie 4.0. À ce propos, divers projets de recherche européens ont été menés par des consortiums regroupant des acteurs académiques et industriels. Parmi ces projets de recherche et développement en milieu industriel, nous pouvons en citer quelques-uns étudiés et développés les cinq dernières années.

Le projet européen STAMINA [75] « Sustainable and Reliable Robotics for Part Handling in Manufacturing Automation » a été financé par le programme Horizon H2020. Le manipulateur mobile développé dans le cadre de ce projet est dédié à l'exécution des opérations de préparation et de distribution de kits de pièces dans l'industrie automobile.

Le projet ColRobot [76] a été lui aussi financé par le programme Horizon H2020. Il a pour objectif le développement d'une plate-forme robotique de manipulation mobile collaborative capable d'effectuer des tâches en industrie aérospatiale et automobile.

Le projet CoMManDIA [77] « Collaborative robotic Mobile Manipulation of Deformable Objects in Industrial Applications » a pour objet de recherche la Manipulation Mobile robotique Collaborative des objets Déformables dans les Applications Industrielles. CoMManDIA est soutenu par le programme Interreg Sudoe, financé par le Fonds européen de développement régional (FEDER). Ce projet propose de développer un ensemble de nou-

velles techniques et technologies de perception robotique, de commande et de planification pour la manipulation dynamique d'objets déformables, avec des manipulateurs mobiles collaboratifs.

Le projet CoRoT [1], dans le cadre duquel s'inscrivent les travaux de recherches de cette thèse, est financé par le programme Interreg France-Manche-Angleterre de la Commission européenne pour un budget de 3.902.330 €. Le projet CoRoT, qui a débuté officiellement en septembre 2017 et s'achève en novembre 2021, vise notamment à développer une solution autonome de manipulation mobile des pièces industrielles pour les déposer dans des machines-outils.

2.1.2 Les finalités du projet

La projet CoRoT a pour objectif de fournir aux entreprises manufacturières, en particulier les PME dans le secteur de la logistique, région transmanche, une solution technologique innovante et abordable de manipulateur mobile collaboratif accompagnée d'outils numériques pour les systèmes de production flexible tels que la supervision, l'intelligence artificielle, etc ainsi que les connaissances associées pour améliorer leur compétitivité.

CoRoT renforcera également le transfert entre les universités et les industries de la Manche. Pour cela, des institutions de recherche (Université Le Havre Normandie, CESI, ESIGELEC, Université de Greenwich et Université d'Exeter) innoveront avec des PME (BA Systèmes, AUTOFINA et CERI). Les partenaires industriels sont les fournisseurs des équipements robotiques utilisés dans développement de la solution CoRoT. Cette solution répond au cahier des charges de CERI, PME utilisatrice.

Des workshops industriels, des webinaires et des sessions interactives sont organisés chaque trimestre dans le cadre du projet pour participer à la diffusion des travaux et fournir un support technique à plus de 50 PME.

2.1.3 Les livrables du projet

Les travaux ont été répartis en sept modules contenant des livrables pour répondre à l'ensemble des objectifs attendus. Les deux premiers modules de travail sont la gestion de projet et la communication, qui consiste à mettre en place les outils de gestion de temps, de

budget et de risque, de répartir les tâches et d'assurer la communication entre les membres du projet par l'organisation de réunions mensuelles et trimestrielles ainsi que la diffusion du projet à d'autres PME et au grand public à travers des événements industriels, un site internet et des MOOCs.

Dans le troisième et le quatrième module de travail, les partenaires participent au développement du manipulateur mobile CoRoT par l'adaptation des plates-formes robotiques mises à disposition par Autofina et BASystème, au développement des bibliothèques nécessaires (navigation, manipulation et perception d'environnement) et à la réalisation du scénario envisagé. Dans ces deux modules, chacun des partenaires développe un cas d'étude avec leurs propres manipulateurs mobiles. En particulier, l'université du Havre Normandie contribue dans le cadre de cette thèse par le cas d'étude GREAH.

Le cinquième module est intitulé « Contrôle et simulation de système de production flexible ». Il a pour but la supervision des systèmes manufacturiers intégrés.

Au sein du sixième module, les partenaires s'intéressent à la formation de 100 membres du personnel de PME et de 500 jeunes à travers des MOOCs ainsi qu'à la création d'un espace d'échange avec d'autres PME via des workshops pour communiquer les résultats, recueillir les besoins, et affiner les axes techniques du projet.

Dans le dernier module de travail, les partenaires mettent en avant les résultats scientifiques et les produits innovants développés dans le cadre du projet vers les fournisseurs de technologie et les utilisateurs.

2.2 Le cas d'étude GREAH

2.2.1 Le scénario

Les opérations développées dans nos travaux concernent la saisie des pièces brutes du stock dans un magasin, le transport de celles-ci pour arriver à la machine d'usinage ainsi que la charge et le décharge de l'objet dans la machine. Ce scénario nécessite de maîtriser la mise en oeuvre des différentes fonctions du manipulateur mobile : la manipulation de l'objet à travers le bras manipulateur et la navigation autonome dans l'entrepôt logistique via le robot mobile. Ces fonctions sont réalisées en se basant sur la perception de l'environnement du

robot à travers les capteurs installés sur celui-ci. Le scénario développé se divise en trois étapes :

- **Étape 1** : Manipulation d'objet.

Les pièces à usiner sont stockées dans la zone A sur la figure 2.1. Le robot se positionne proche de la zone de stockage des pièces pour les saisir par le biais du bras manipulateur équipé d'un préhenseur à deux doigts. Le déplacement du bras robotique dépend de l'estimation de la pose de la pièce définie par le traitement visuel. La pièce est alors déposée sur la plate-forme mobile et le bras replié en position de sécurité avant de passer à l'étape suivante. La manipulation d'objet sera détaillée dans le chapitre 3 pour la saisie.

- **Étape 2** : Transport de l'objet.

Le robot mobile se déplace dans l'atelier pour aller de la zone A vers la zone C et arriver précisément à une position proche de la machine d'usinage. Le chapitre 2 explique l'exécution de la partie déplacement du robot mobile.

- **Étape 3** : La dépose de la pièce dans la machine.

La troisième partie du scénario consiste à placer précisément la pièce dans la machine d'usinage après correction des erreurs de position/orientation du robot par rapport à la machine. Cette étape sera traitée dans le chapitre 4.

2.2.2 La plate-forme robotique utilisée

L'exécution du scénario est réalisée par un manipulateur mobile. Ce dernier a été assemblé au sein du laboratoire par l'équipe robotique. Celui-ci est constitué d'un bras manipulateur à 6 ddl avec un préhenseur à deux doigts placé au poignet du bras robotique. L'ensemble du bras robotique et du préhenseur est embarqué sur le robot mobile holonome Summit XL.

La figure 2.2 illustre la version finale du manipulateur mobile développé au laboratoire GREAH contenant les éléments robotiques assemblés. Le robot est aussi équipé de capteurs pour la perception de son environnement lors de l'exécution des tâches de manipulation et de transport. Ces capteurs sont des deux caméras RGB-D placées sur deux mâts, deux caméras PTZ permettant de scruter le demi-espace situé au-delà de 30 cm du sol et des LiDARs dont

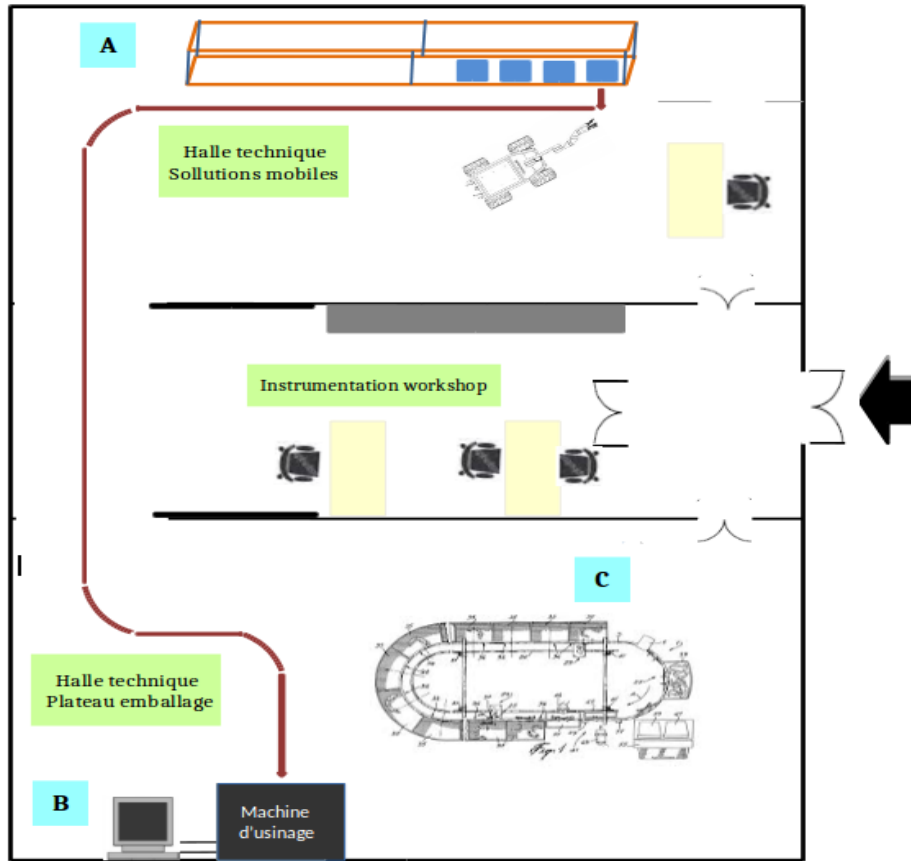


FIGURE 2.1: La carte de la halle robotique du PIL

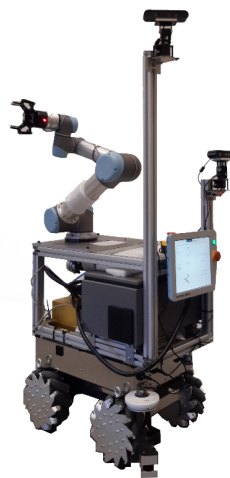


FIGURE 2.2: Le manipulateur mobile « GREAH »

la nappe est à 7 cm du sol, leurs emplacements et leurs caractéristiques seront étudiés dans les parties suivantes.

2.2.3 Principes de navigation d'un manipulateur mobile

La navigation autonome du manipulateur mobile consiste à déplacer toute la plate-forme robotique suivant un chemin généré pour atteindre une position finale. Le processus de navigation autonome s'appuie sur trois principes qui sont les suivants :

- Perception : acquérir les données sur l'environnement proche du robot pour assurer les étapes de localisation et planification.
- Localisation : pouvoir définir l'emplacement du robot dans son environnement.
- Planification : planifier la trajectoire à accomplir par le robot pour atteindre la position souhaitée.

Les approches de la navigation des robots mobiles à roues varient suivant que le robot évolue dans un espace non fermé (usuellement dénommé « outdoor navigation » dans la littérature) ou à l'intérieur de bâtiments, dans des entrepôts par exemple (« indoor navigation»). Cette variabilité d'environnement de navigation induit une différence de capteurs utilisés pour la perception de l'environnement du robot et une variation des techniques de localisation. [78].

Nous nous intéressons dans notre travail à la navigation indoor. Généralement la localisation dans ce type d'environnement se base sur une cartographie prédéfinie de l'espace, définissant l'espace libre et les obstacles statiques. A partir de cette carte, le robot peut se localiser et générer les trajectoires possibles. Les approches de génération des chemins [79] peuvent être réparties en deux groupes : des approches dites classiques et d'autres réactives. Les approches classiques de planification de trajectoire pour les robots mobiles sont identiques à celles développées pour les bras robotiques (voir la section 2.2.4). L'algorithme génétique, la logique floue, le réseau neuronal, l'algorithme de luciole, l'optimisation de l'essaim de particules sont des exemples des approches de planification de trajectoire réactives.

Le déplacement du manipulateur mobile de la zone de stockage à la machine-outil a été mis en oeuvre en utilisant des outils développés par la communauté ROS, en particulier le

module de navigation Stack. Notre contribution a consisté à intégrer ces outils logiciels sur nos matériels.

Navigation stack

La navigation stack sous ROS fournit un processus de navigation complet, flexible et adaptable à divers robots mobiles avec différents capteurs. La navigation stack englobe un ensemble de bibliothèques pour la localisation, la planification de trajectoires dans des environnements non structurés, dynamiques et à échelle variable ainsi que l'évitement d'obstacles.

Le concept de la navigation stack repose sur le principe du SLAM [80] « Simultaneous Localization And Mapping ». Le SLAM est un processus de localisation et de cartographie simultanées. Il consiste à créer une cartographie de l'environnement dans lequel le robot exécute son déplacement afin d'assurer son repérage sur cette carte et de créer un plan de mouvement en fonction de la carte. La carte obtenue fournit un ensemble d'informations statiques sur l'espace de navigation du robot. Les algorithmes de SLAM diffèrent selon les types des capteurs utilisés pour se localiser et construire la carte de l'environnement. Il existe une grande variété d'approches SLAM. Dans ce manuscrit nous citons trois catégories :

- Une catégorie d'approches SLAM qui se base sur l'extraction des images RGB par des caméras monoculaires pour la construction de la cartographie de l'environnement comme ORB-SLAM [81] et Kimera [82].
- Une autre catégorie utilise les données RGB-D tel que RTAB-Map [83] et Kintinuous [84].
- D'autres approches SLAM dites Laser-SLAM se basent sur des capteurs lasers comme le LiDAR pour la localisation et la cartographie. Les algorithmes les plus connus de ce type de SLAM sont le Gmapping [85] et le Hector-SLAM [86].

Le Gmapping « Grid Mapping » est un filtre à particules qui permet de créer des cartes en grille à partir de données laser [85]. Le processus de cartographie dans notre développement est basé sur le package Gmapping sous ROS. Initialement, le robot est déplacé dans les locaux de navigation à travers des commandes de vitesse par le joystick, lors de ce déplacement le nuage des points récupéré par les LiDARS embarqués sur la base mobile du

robot est enregistré afin de générer la carte 2D de l'environnement. La figure 2.3 représente une vue en simulation de l'entrepôt PIL du laboratoire à partir de la carte créée de cet environnement.

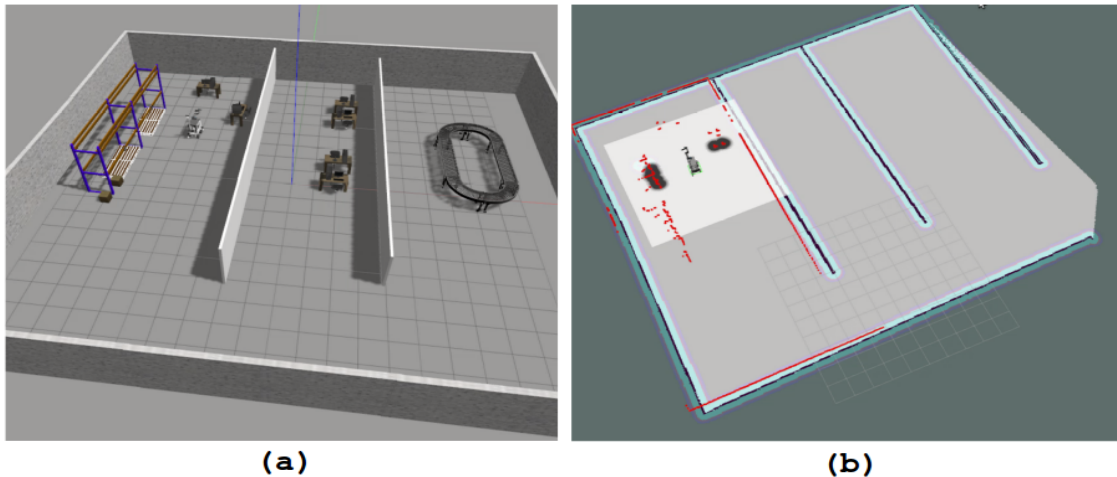


FIGURE 2.3: Représentation du robot dans dans l'entrepôt simulé PIL : (a) le robot dans l'environnement simulé sur GAZEBO (b) le robot sur RVIZ dans la carte créée, reliée à l'environnement GAZEBO avec les représentations des points LiDARs (en rouge).

La figure 2.4 illustre l'architecture de navigation stack contenant les trois rubriques de navigation autonome (perception, localisation et planification). A partir des données de l'environnement perçu par le capteur LiDAR et les informations géographiques issues du serveur Map, le robot se localise dans l'espace de navigation en utilisant l'approche AMCL « Adaptive Monte Carlo Localization » [87]. La trajectoire se génère par Move_base pour atteindre la position désirée en tenant compte des données de localisation du robot. Dans les parties suivantes de ce chapitre nous détaillons chaque rubrique.

Partie I : Perception de l'environnement du robot

Dans cette étape de la navigation autonome, le robot collecte les informations sur l'environnement qui l'entoure à travers les capteurs extéroceptifs ainsi que les informations sur son état interne à travers les capteurs proprioceptifs.

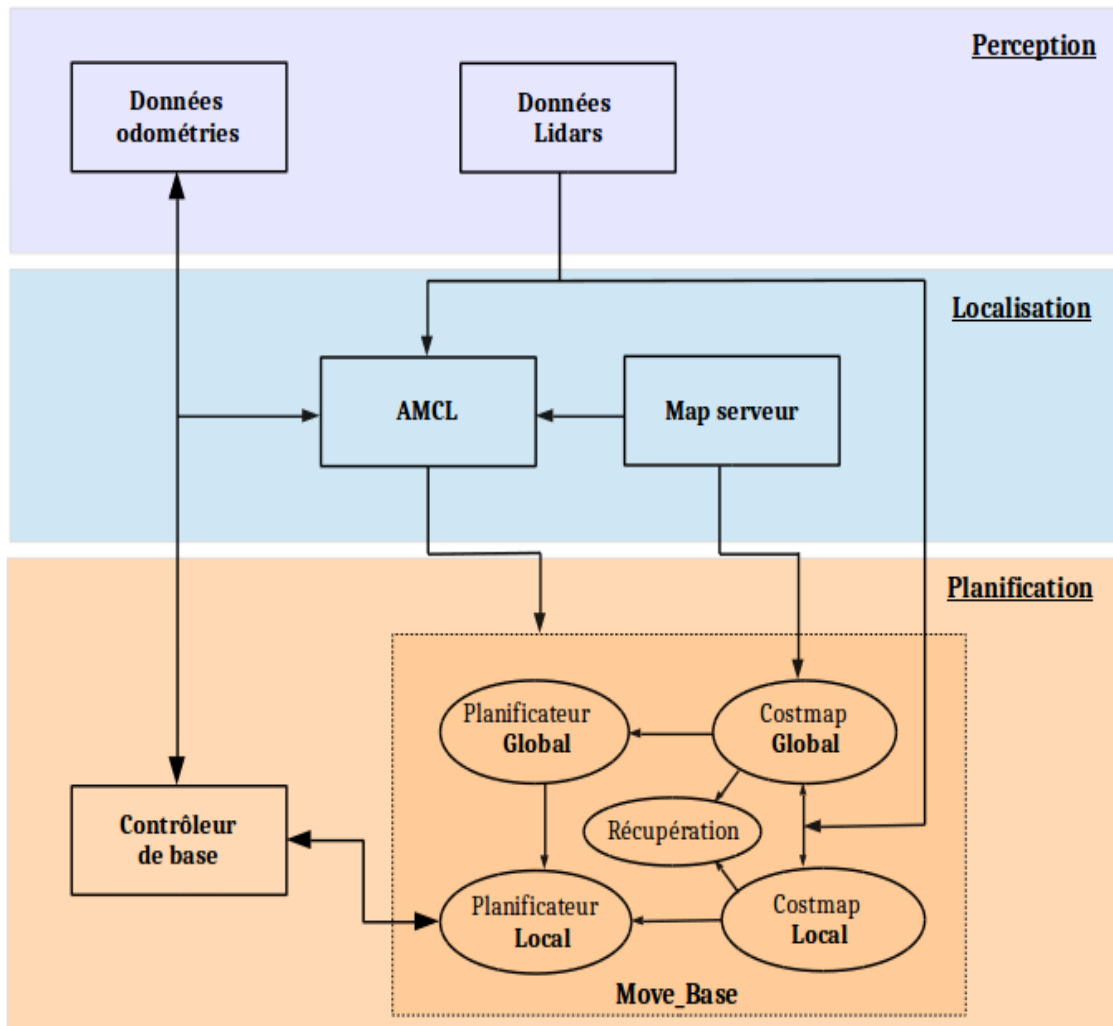


FIGURE 2.4: L'architecture de navigation stack sous ROS.

Perception proprioceptive

Le principe de l'odométrie repose sur le calcul du nombre de tours des roues. Sur le robot mobile holonome Summit XL, chaque roue est actionnée par un motoréducteur indépendant et disposant de codeurs permettant de compter le nombre de tours de la roue à partir des impulsions délivrées par le capteur. A partir de ces données et en tenant compte des paramètres du robot, les déplacements du robot sont estimés par odométrie via les équations de cinématique.

La figure 2.5 illustre la cinématique de la base mobile. Chaque roue du robot tourne autour du CIR (Centre Instantané de Rotation) le long d'un cercle avec un rayon de roue noté r . Le point CIR, qui représente l'intersection des axes des roues, est le point central de rotation lorsque le robot se déplace. La vitesse linéaire d'une roue est $v = 2\pi r / T$ en m/s, où T est le temps nécessaire pour effectuer un tour complet autour de CIR [88]. La vitesse angulaire est $\omega = \pi/T$ en rad/s. Les vitesses des roues (droite et gauche) s'écrivent comme suit :

$$v_d = r\omega_d = (R + l/2)\omega \quad (2.1)$$

$$v_g = r\omega_g = (R - l/2)\omega \quad (2.2)$$

Avec R la distance entre le CIR et le centre du robot, l la longueur de l'axe de la roue droite à l'axe de la roue gauche. En faisant (2.1) + (2.2) et (2.1) - (2.2), on obtient :

$$\begin{cases} R = \frac{l}{2} \frac{v_d + v_g}{v_d - v_g} \\ \omega = \frac{v_d - v_g}{l} \end{cases} \quad (2.3)$$

Supposons que le robot se déplace d'une position $(X_{robot}, Y_{robot}, \theta_{robot})$ vers une autre position $(X'_{robot}, Y'_{robot}, \theta'_{robot})$ avec une vitesse angulaire ω pendant δ_t secondes. L'estimation de la nouvelle pose du robot peut s'écrire :

$$\begin{cases} X'_{robot} = X_{robot} + R \cos \omega \delta_t \\ Y'_{robot} = Y_{robot} + R \sin \omega \delta_t \\ \theta'_{robot} = \theta_{robot} + \omega \delta_t \end{cases} \quad (2.4)$$

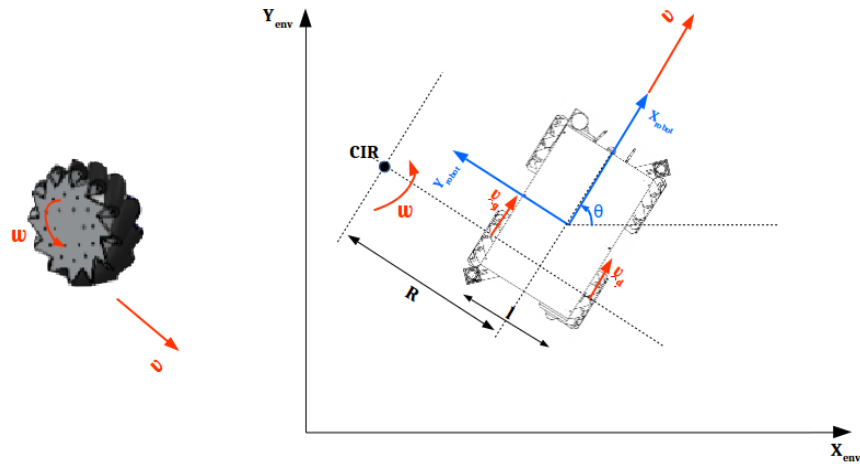


FIGURE 2.5: Repère de localisation de la base mobile

Perception extéroceptive

La plate-forme robotique utilisée dans les travaux de cette thèse est équipée de deux LiDARs Hokuyo UST-10LX, comme illustré dans la figure 2.6. Le premier est placé devant à droite, le deuxième est derrière à gauche. Les nappes sont à 0.12 m du sol. Leur portée de détection est de 10 m et le secteur angulaire couvert est de 270° avec une résolution angulaire de 0.25° et une durée d'échantillonnage de 25 msec.

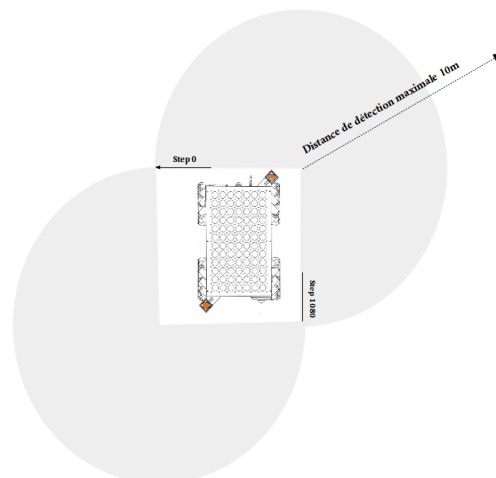


FIGURE 2.6: Illustration de l'emplacement des LiDARs et leurs angles de détection

Partie II : Localisation du robot dans l'environnement

L'opération dite de localisation du robot consiste à estimer sa pose courante dans l'environnement à partir des données sensorielles. La capacité d'un robot mobile à déterminer sa localisation dans l'espace et à localiser d'autres lieux d'intérêt dans son environnement sont des compétences fondamentales et essentielles pour la navigation autonome. À partir de cette localisation le robot assure sa planification de trajectoire stratégique pour des tâches telles que l'atteinte d'une position cible, l'exploration et l'évitement d'obstacles, et l'exécution en temps réel du processus de localisation permet de commander le robot pour suivre les trajectoires planifiées.

Généralement la pose d'un robot mobile est définie dans un espace à six dimensions, comprenant la position (x, y, z) et l'orientation (roulis, tangage, lacet), mais trois variables suffisent pour décrire son évolution (x, y, θ) où (x, y) sont les coordonnées cartésiennes du centre du robot dans le repère plan atelier et θ l'angle de lacet du robot. La base mobile que nous utilisons possède quatre roues holonomes, permettant de commander indépendamment les coordonnées x, y et θ .

Dans cette étape de navigation autonome le processus de localisation est réalisé par l'approche AMCL « Adaptive Monte Carlo Localization » sur laquelle la plate-forme de navigation Stack se base. AMCL [87] fournit une localisation du robot dans une carte 2D à l'aide d'un filtre à particules. L'algorithme commence par une distribution de probabilité de la pose du robot sous forme de particules représentant une pose probable du robot dans la carte à un instant t . Ces particules sont représentées dans la figure 2.7 par des flèches rouges et formulées en $X_t = x_t^1 + x_t^2 + .. + x_t^m$. Ces particules se propagent suivant les équations odométriques du robot à chaque changement de sa pose courante. Lors de la réception de nouvelles lectures des deux capteurs LiDARs (figure 2.6), le filtre calcule l'adéquation des mesures LiDAR à la sous-carte locale pour chaque particule, le résultat correspondant à la probabilité que la particule corresponde à la vraie pose du robot. Les particules à probabilités supérieures à un seuil donné sont gardées et les autres sont rejetées. L'algorithme continue à itérer les étapes de distribution des particules et de calcul des probabilités où chaque distribution X_t dépend de la précédente X_{t-1} . À chaque itération le nombre de particules restantes devient plus faible et converge vers un seul cluster qui correspond à la vraie

pose du robot jusqu'à l'obtention d'une seule particule qui représente la localisation réelle du robot (X_{robot} , Y_{robot} , θ_{robot}).

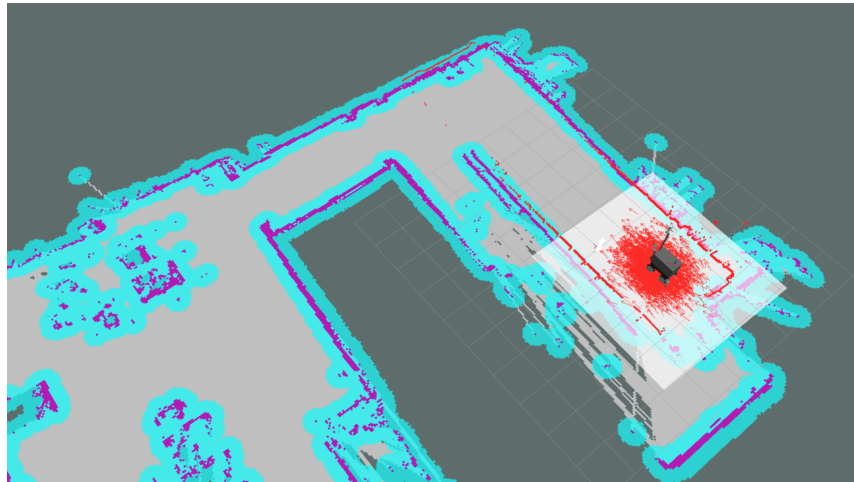


FIGURE 2.7: Visualisation sur RVIZ des particules distribuées lors de la localisation de notre robot dans la carte PIL, par AMCL

Partie III : La planification de la trajectoire

La partie planification de trajectoire d'un robot mobile consiste à déterminer un chemin optimal depuis l'emplacement actuel du robot jusqu'à la position souhaitée. La navigation Stack s'appuie sur une sous plate-forme Move_base [89]. Cette dernière englobe plusieurs approches de planification différentes, notamment l'algorithme de Dijkstra et l'algorithme best-first search A*. Move_base prend en considération l'ensemble des entrées : les données de localisation, les informations de la carte de l'environnement et les signaux des LiDARs. La planification par Move_base distingue une trajectoire globale et une trajectoire locale, sur deux cartes cartographiques connues sous les termes de « global and local costmaps ».

La trajectoire globale représente le chemin global du point de départ du robot qui est sa position courante jusqu'à la position finale. Cette trajectoire est définie sur la costmap globale. Cette costmap est une carte statique contenant toutes les informations sur l'environnement de navigation : les zones vides d'obstacles pour le déplacement, la projection des obstacles statiques dans le plan 2D, la zone de sécurité autour de ces obstacles pour éviter

la collision du robot avec ces obstacles.

La trajectoire locale représente le chemin suivi par le robot qui est variable en fonction des obstacles dynamiques autour du robot dans la costmap locale. Cette trajectoire est définie sur la costmap locale. Cette costmap est une carte de forme rectangulaire entourant le robot et qui est représentée dans la figure 2.7 par un rectangle blanc. La costmap locale suit en temps réel la position du robot afin d'éviter les collisions avec les obstacles dynamiques détectés à l'aide des signaux LiDARs.

La trajectoire globale est créée par le planificateur global en se basant sur les données de localisation issues d'AMCL, les informations de la costmap globale et la position désirée. Le planificateur local crée un déploiement de trajectoire sur la trajectoire globale pour revenir avec un moindre coût dans la trajectoire d'origine en utilisant l'approche de fenêtre dynamique dite DWA (Dynamic Window Approach) [90]. Le principe consiste à traduire une cible cartésienne (x, y) en commandes de vitesse $(dx, dy, d\theta)$ pour le robot, avec un choix de la vitesse optimale.

Lors de la navigation autonome du robot par Move_base trois trajectoires sont générées et visualisées sur RVIZ comme montré dans la figure 2.8 : la trajectoire globale, la trajectoire locale et une trajectoire qui représente le mouvement en temps réel du robot.

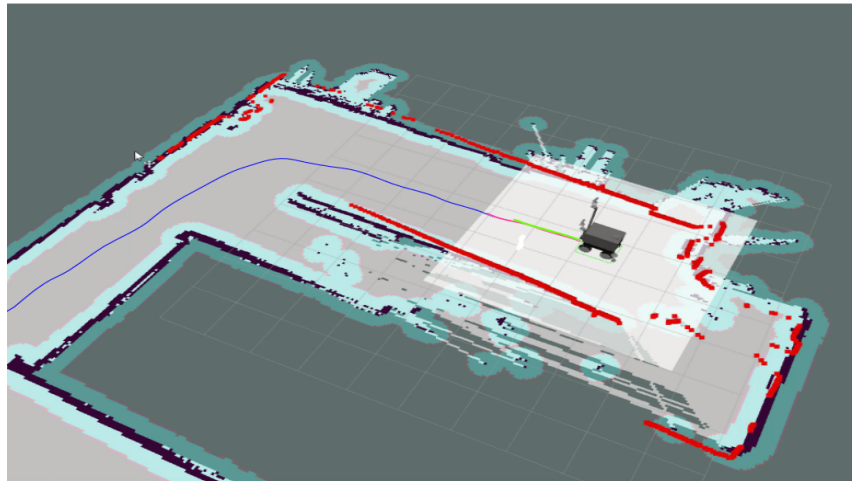


FIGURE 2.8: Le robot localisé dans la carte de l'entrepôt réel PIL du laboratoire sur RVIZ avec les trajectoires, globale en bleu, locale en rose et la trajectoire suivie en temps réel en vert.

2.2.4 Principes de manipulation robotique

La manipulation robotique d'un objet est la saisie ou/et la dépose en autonomie d'une pièce par le biais d'un ou plusieurs bras robotiques pour la réalisation des tâches de pick and place ou d'assemblage des pièces ou d'autres tâches d'usinage ou de manutention (ébauchage, perçage...). Dans notre cas d'étude nous nous intéressons à la fonction de pick and place pour la saisie des pièces à usiner du stock et les mettre dans le robot mobile et à la fin du scénario pour la dépose de la pièce dans la machine d'usinage.

Le pick-and-place est considéré comme une méta-discipline car l'accomplissement de cette tâche fait appel à un ensemble de sous-disciplines [91]. Les étapes de l'exécution de cette tâche peuvent être définis comme suit :

- L'analyse de la scène pour identifier l'objet à saisir parmi les autres éléments existants dans la scène.
- La localisation d'objet pour déterminer les coordonnées réelles de celui-ci ; sa position et son orientation par rapport au robot.
- Le chemin à suivre par le bras robotique pour déterminer comment l'objet doit être saisi de la position actuelle du robot vers la position désirée.
- La stratégie de préhension de l'objet dans le but de déterminer comment le préhenseur doit saisir correctement l'objet.

Planification de trajectoire du bras robotique

La manipulation robotique nécessite le déplacement du bras robotique d'une position de départ vers une position d'arrivée afin d'exécuter la tâche robotique souhaitée. Le calcul de la cinématique des variables articulaires est appelée la planification de trajectoire [92].

La planification de trajectoire a été un domaine de recherche actif et essentiel au cours des années 90 en robotique. Les concepts fondamentaux sont apparus dans les années 80, notamment la modélisation via l'espace de configuration C [93]. Cet espace est constitué de l'ensemble des configurations possibles du robot prenant en compte chaque articulation du bras robotique. Cet espace est décomposé en deux sous-espaces : le premier noté C_{free} désigne le sous-espace libre pour l'ensemble des configurations libres d'obstacles et sans auto-collision possible entre les solides de la chaîne cinématique, et sans collision entre

le robot et son environnement. Le second noté C_{obs} correspond au sous-espace constitué de l'ensemble des configurations où le robot est en collision. La plupart des techniques de planification de trajectoire sont basées sur ce concept pour assurer la faisabilité de la configuration de chaque articulation du bras robotique dans l'espace libre C_{free} pour aller d'une configuration initiale $q_I \in C_{free}$ à une configuration finale $q_F \in C_{free}$ en évitant tous les obstacles C_{obs} . La figure 2.9 représente un exemple d'une trajectoire entre une position initiale q_I et la position finale q_F d'un bras robotique de 2 DDL dans C_{free} , avec $\theta_1 \in [0; \pi]$ et $\theta_2 \in [0; \pi]$.

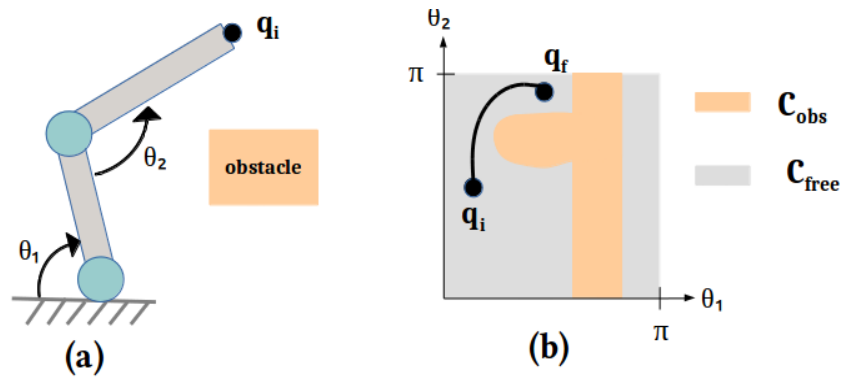


FIGURE 2.9: Représentation de l'espace des configurations (b) du bras robotique (a).

Selon la littérature nous pouvons distinguer six catégories d'approche de génération de trajectoire, que nous avons réparties en deux sous-ensembles dans notre rapport, le premier sous-ensemble contient la méthode des champs de potentiel, la décomposition cellulaire et l'approche "roadmap".

La théorie des champs de potentiel consiste à générer un champ de potentiel attractif autour de la position désirée et un champ de potentiel répulsif autour de la zone C_{obs} comme illustré dans la figure 2.10(a). L'algorithme calcule donc un vecteur résultant qui conduira le robot dans un mouvement pour arriver à la position à atteindre sans collision [94].

La méthode de planification de trajectoire par décomposition cellulaire tente de décomposer C_{free} en un certain nombre de régions, appelées cellules compte tenu des obstacles dans l'espace comme dans l'exemple de la figure 2.10 (b). Différentes techniques de décomposition sont développées. Un graphique de noeuds correspondant aux cellules est construit.

A partir de ce graphe un chemin reliant le point de départ et la position finale est calculée [95].

L'approche roadmaps est une représentation de connectivité de tout l'espace C_{free} [96]. La carte de connectivité contient les nœuds qui sont des configurations libres et les arêtes relient toute combinaison possible de deux nœuds comme dans l'exemple de la figure 2.10 (c). Le chemin de la solution, en allant de la configuration initiale à la configuration finale peut être obtenu en implémentant une des méthodes de recherche de graphe telle que Dijkstra [97].

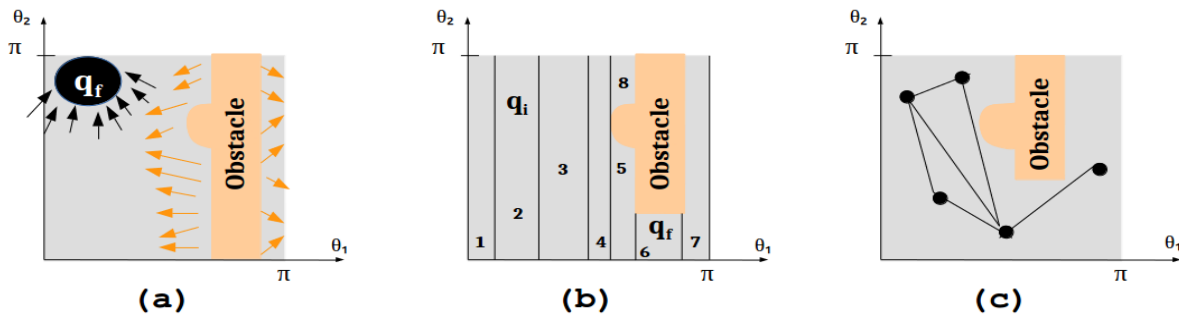


FIGURE 2.10: les techniques de génération des trajectoires : (a) par la méthode du potentiel, (b) par décomposition cellulaire, (c) méthode roadmaps.

Le deuxième sous-ensemble d'approche de génération de trajectoire de la plate-forme de planification de trajectoire de ROS (MoveIt!) est constitué d'algorithmes basés sur l'échantillonnage [98]. Les méthodes citées dans le premier sous-ensemble reposent sur une représentation explicite de la géométrie de C_{free} . En revanche, les méthodes basées sur l'échantillonnage remplacent l'espace de configuration continu par une représentation discrète sous forme d'échantillons en suivant deux étapes : l'échantillonnage aléatoire permet de sélectionner des itinéraires aléatoires sans collisions, puis une recherche de chemin par des techniques de graphes. La caractéristique principale de ces algorithmes est l'approche probabiliste qui ne garantit pas que l'algorithme trouve une solution, ni la solution optimale, mais qui, si une solution existe et que les moyens alloués au calcul sont suffisamment importants, l'algorithme peut trouver cette solution, alors qu'une approche classique par la force brute de calcul ne pourrait pas de conclure avec des délais compatibles avec les exigences temps

réel du processus. Nous pouvons classer les méthodes probabilistes en deux groupes : les planificateurs multi-requêtes et les planificateurs à requête unique.

- les planificateurs multi-requêtes fonctionnent de manière similaire à la technique road-maps car ils sont capables de déterminer plusieurs requêtes pour un même système, cela nécessitant cependant un prétraitement pour définir la PRM (probability RoadMap).

- les planificateurs à requête unique n'intègrent dans leurs algorithmes uniquement que la position initiale et finale. Une seule requête est générée qui relie la position actuelle et la position désirée. La méthode RRT «Rapidly-exploring Random Trees» [99] est l'une des méthodes à requête unique les plus largement utilisées. Cette méthode est représentée par un arbre dont la configuration initiale de la trajectoire est le premier noeud de l'arbre, arbre qui se construit dynamiquement en progressant vers les noeuds le plus proches jusqu'à atteindre la configuration finale.

MoveIt![100] est une plate-forme de planification de trajectoire compatible avec ROS, utilisée dans notre travail de manipulation sur le bras manipulateur à 6 ddl. MoveIt! fournit un système assez complet d'outils de planification et d'exécution de mouvement qui tient également compte de l'environnement perçu pour éviter les collisions aussi bien avec l'environnement que les auto-collisions. MoveIt! intègre un ensemble d'algorithmes de planification de mouvement et d'algorithmes d'évitement d'obstacles et de collisions sous différentes bibliothèques : OMPL [101], STOMP [102], SBPL [103], CHOMP [104].

OMPL « Open Motion Planning Library » se compose de nombreux algorithmes de planification de mouvement basés sur l'échantillonnage à la pointe. STOMP « Stochastic Trajectory Optimization for Motion Planning » est un planificateur de mouvement basé sur l'optimisation qui permet de planifier des trajectoires fluides en évitant les obstacles et en optimisant les contraintes. SBPL « Search-based Planning Library » est un ensemble de planificateurs capables de générer un chemin depuis n'importe quel point courant jusqu'à une cible en combinant plusieurs mouvements primitifs. CHOMP « Covariant Hamiltonian Optimization for Motion Planning » est une bibliothèque utilisée pour l'optimisation de la trajectoire invariante au reparamétrage, elle se base sur des techniques de gradient fonctionnel pour améliorer de manière itérative la qualité d'une trajectoire initiale. MoveIt! a été intégré avec succès à de nombreux robots, notamment le PR2 et les robots UR. MoveIt! est

entièrement écrit en C ++ mais inclut également des liaisons Python.

Les principaux avantages d'utilisation de MoveIt ! dans notre travail sont :

- Une génération adéquate d'une trajectoire sans collision pour atteindre la position de l'objet à saisir. Cela est réalisé par combinaison de deux bibliothèques sous ROS.
- Un calcul automatique de la cinématique inverse par le solveur jacobien numérique KDL « Kinematics and Dynamics Library » basé sur l'URDF « Unified Robot Description Format » du robot.
- Un contrôle bas niveau de la position, de la vitesse des articulations du bras manipulateur et de la synchronisation du trajet planifié.

2.3 Mise en oeuvre du cas d'étude

2.3.1 Navigation du manipulateur mobile

La carte 2D de l'environnement a été créée en utilisant la technique du Gmapping [85] en téléopérant au préalable le robot dans tout l'atelier où le scénario s'exécute pour collecter les signaux des LiDARs.

Après l'obtention de la carte cartographique du PIL, la navigation Stack a été implémentée avec les paramètres convenables du manipulateur mobile utilisé dans notre développement. Le robot se localise en premier dans la carte du PIL en se basant sur la bibliothèque AMCL [87] et l'odométrie du robot puis ce dernier assure une génération de la trajectoire par Move_base [89] pour aller du magasin de stockage des pièces à usiner dans « la salle technique des robot mobiles » figure 2.11 vers l'emplacement de la machine d'usinage dans « la salle de production » en passant par « la salle d'instrumentation ». Le robot exécute le mouvement en suivant la trajectoire générée et en vérifiant au fur et à mesure la présence d'un obstacle non initialement défini dans la carte pour modifier la trajectoire si nécessaire.

2.3.2 La saisie et la dépose des pièces par le manipulateur mobile

Dans un premier temps, le travail a été basé sur le traitement visuel des marqueurs code pour saisir la pièce du stock et la déposer dans la machine-outil. Ce travail de manipulation

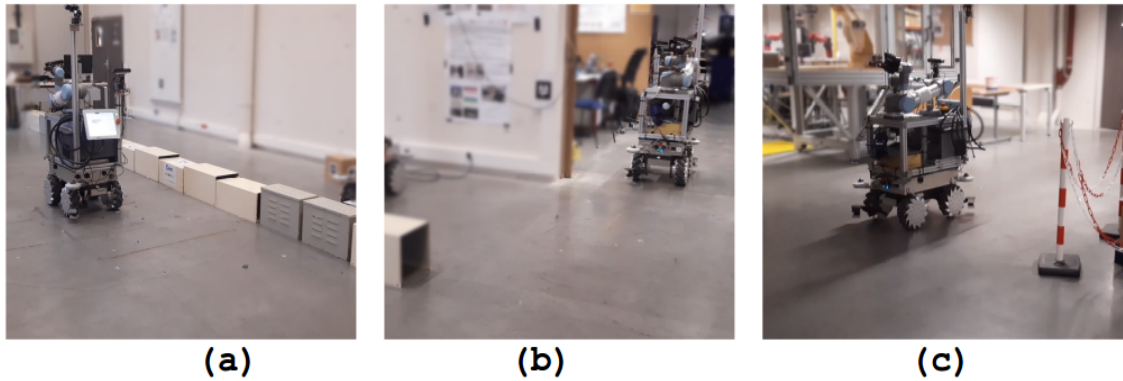


FIGURE 2.11: Le robot se déplace dans l'entrepôt PIL du magasin (a) vers l'emplacement de la machine d'usinage dans la salle de production (c) en passant par la salle d'instrumentation (b).

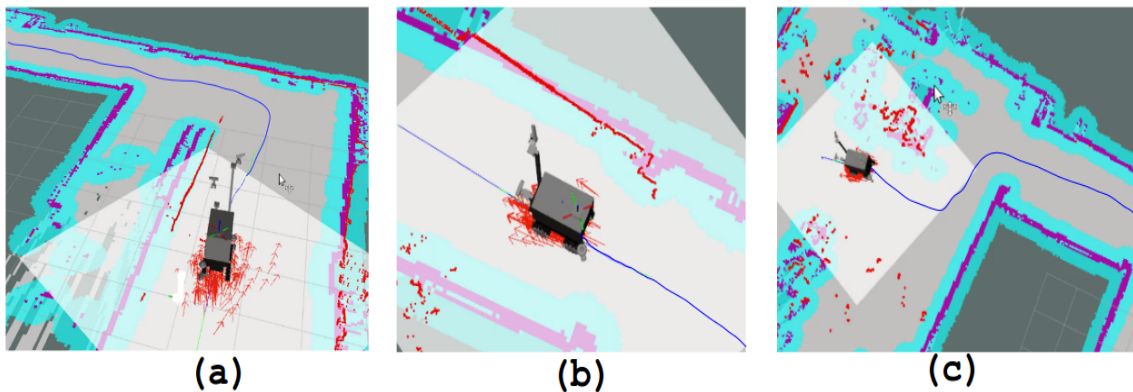


FIGURE 2.12: Visualisation sur RVIZ du déplacement du robot dans l'entrepôt PIL : Le robot se déplace dans l'entrepôt PIL de la salle technique des robot mobiles (a) vers l'emplacement de la machine d'usinage dans la salle de production (c) en passant par la salle d'instrumentation (b).

basé sur la perception visuelle des marqueurs codes de types Artag consiste à détecter la pièce et estimer sa pose par rapport au robot. Un Artag est collé sur la surface de la pièce que nous voulons saisir et la pièce est détectée à partir de la caméra embarquée sur le robot mobile. La pose de la pièce relativement à la base mobile supportant le bras est alors estimée comme illustré sur la figure 2.13.

Le déplacement du bras robotique est par la suite réalisé par le générateur de trajectoire sous ROS (MoveIt!). A partir des données visuelles extraites, le bras robotique se déplace vers la pose calculée de la pièce et assure sa saisie par fermeture des doigts du préhenseur.

Ce processus impose la contrainte physique de coller sur chaque pièce un Artag, ce qui n'est pas toujours possible. Afin de dépasser cette limitation, nous développons dans le chapitre III une méthode de détection d'objet par apprentissage profond.

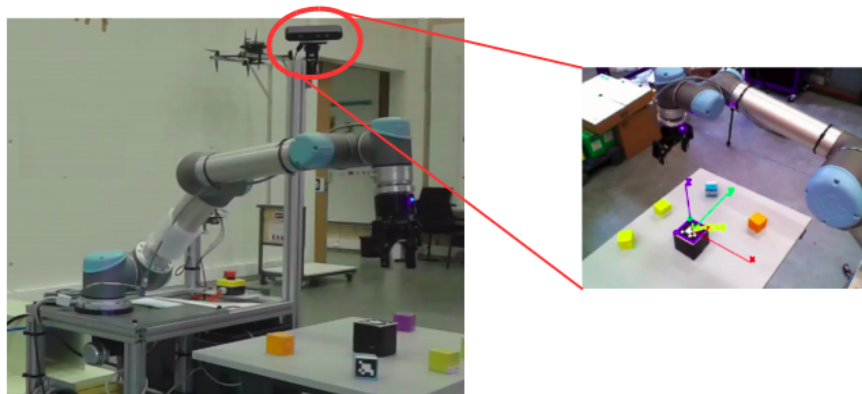


FIGURE 2.13: La saisie basée sur les marqueurs code

2.3.3 La communication avec la machine d'usinage

À l'issue du processus de navigation, le robot mobile est arrivé approximativement devant la machine-outil, mais le processus ignore tout des performances en précision de positionnement du robot. Il est donc illusoire et risqué de démarrer un déploiement du bras en boucle ouverte pour déposer la pièce. Il est impératif de vérifier la position et l'orientation du robot par rapport à la machine. Pour cela, nous utilisons un ARtag placé sur le capot de la machine et un algorithme de détection de marqueurs code, comme illustré sur la figure

2.14 (a).

La caméra RGB-D associée à un mécanisme motorisé de type pan-tilt permet alors d'orienter la caméra afin que l'ARtag puisse entrer dans son champ de vision comme illustré dans la figure 2.14 (b). La pose du robot peut alors être corrigée par asservissement visuel sur l'ARtag en utilisant des fonctionnalités de ROS et au final, le manipulateur vient se positionner toujours sur la même pose résultant du processus asservissement visuel. Cependant, les performances en précision de positionnement de ce processus ne sont pas non plus connues et leur étude sera l'objet des recherches que nous détaillerons dans le chapitre 4.

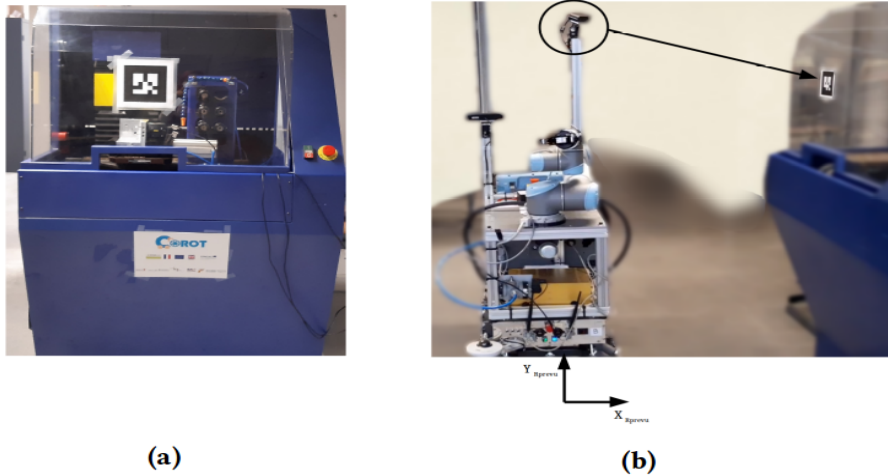


FIGURE 2.14: La machine d'usinage utilisée dans le développement (a) en vue de face (b) le robot se localise par rapport au marqueur code de la machine

Chapitre 3

La saisie des pièces basée sur l'apprentissage profond

Table des matières

3.1	L'identification de l'objet par apprentissage profond	64
3.1.1	YOLO	64
3.1.2	L'apprentissage par transfert	70
3.2	Le cas d'étude GREAH : Saisie de pièces par préhension adaptative	71
3.2.1	Les objectifs	71
3.2.2	Description de la plate-forme robotique	72
3.2.3	Le processus de préhension d'objet	73
3.2.4	Les étapes de pick-and-place avec préhension adaptative	75
3.3	Les résultats expérimentaux	78

Ce troisième chapitre est consacré à la partie manipulation d'objet basé sur l'apprentissage profond, qui a fait l'objet d'une publication scientifique [105]. Le chapitre explicite les étapes de ce développement : la détection d'objet, la classification des faces de l'objet pour un mode de préhension adaptatif et l'estimation de pose. Ce chapitre se clôture par la présentation des résultats expérimentaux obtenus.

3.1 L'identification de l'objet par apprentissage profond

La détection d'objets par apprentissage profond peut être répartie en deux méthodes : des algorithmes en une étape de traitement visuel et des algorithmes en deux étapes. Les algorithmes de détection d'objets en deux étapes impliquent dans un premier temps de découper l'image originale en régions puis dans un second temps d'utiliser les réseaux de classification et de régression tels que la série R-CNN [106], sa version améliorée Faster R-CNN [107] et Mask R-CNN [108]. Les algorithmes de détection d'objets en une seule étape peuvent accomplir les tâches de classification et de régression directement sur des cadres correspondant aux objets détectés sans avoir à effectuer de régionnement de l'image. Il s'agit par exemple de SSD [109], RetinaNet [110] et YOLO [111] l'algorithme sur lequel se base notre travail.

3.1.1 YOLO

YOLO est l'abréviation de « You Only Look Once ». Il s'agit d'un algorithme d'apprentissage profond basé sur un réseau neuronal convolutionnel pour la détection d'objets en temps réel. Il permet de prédire simultanément les contours carrés ou rectangulaires des objets dans l'image et leurs probabilités de classe [111].

Les algorithmes de détection d'objets par apprentissage profond existants utilisent différentes approches. Parmi ces approches celle de la fenêtre coulissante basée sur la réutilisation des classificateurs, qui veut dire la détection d'un objet dans l'image se fait par l'évaluation de son classificateur dans divers emplacements régulièrement espacés sur toute l'image [112]. La proposition de région est une autre approche qui vise à générer des cadres potentiels dans l'image puis à exécuter un classificateur sur ces cadres proposés. Un post-traitement est utilisé ensuite pour affiner ces cadres de délimitation et éliminer les détections en double. Nous trouvons cette approche par exemple dans le réseau de neurones R-CNN [113]. YOLO se base sur l'approche de régression qui permet de prédire les classes et les cadres des objets dans l'image en une seule exécution de l'algorithme. Cela augmente sa rapidité et sa précision de détection par rapport aux autres algorithmes. YOLO prend en compte l'intégralité de l'image lors de l'apprentissage. C'est ainsi qu'il apprend des repré-

sentations généralisables. Avec une fréquence de 45 images par seconde, YOLO permet une détection en vidéo et en temps réel en tenant en compte des images complètes. Lors de la détection par YOLO, l'image entière est utilisée comme entrée du réseau, elle est répartie entièrement en une grille $S \times S$ pour effectuer la détection dans chaque cellule de la grille. Chaque cellule prédit des cadres avec un score de confiance (SC) pour chaque cadre. Ce score correspond à la présence de l'objet dans le cadre prédit ou non. SC est représenté par l'équation suivante :

$$SC = Pr(Object) * IoU(réel, Prédiction)$$

avec :

$$Pr(Object) \in [0, 1]$$

Si aucun objet n'existe, le score de confiance est égal à zéro, sinon, le score de confiance est égal à l'intersection sur l'union (IoU) entre le cadre prédit et le réel. Un seul SC sera estimé au final pour chaque cellule avec une probabilité de classe conditionnelle qui lui correspond. La figure 3.1 illustre les étapes de détection par YOLO basé sur l'approche de régression.

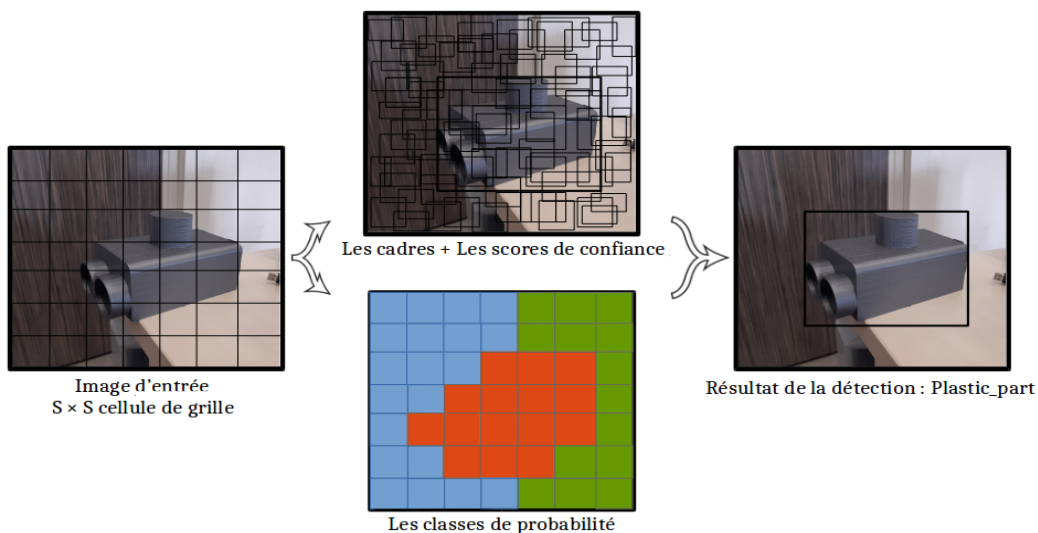


FIGURE 3.1: La détection par YOLO basée sur l'approche de régression

La première version de YOLO-V1 [111] a été développée en Mai 2016, quelques mois plus tard une deuxième version est introduite par les mêmes auteurs [114]. 2018 voit l'ap-

parition de la version 3 de YOLO (YOLO-V3). Dernièrement en 2020, d'autres chercheurs ont développé la quatrième version de l'algorithme, YOLO-V4 [115].

Dans notre développement nous avons utilisé YOLO-V3 puis YOLO-V4. Les explications de ces deux versions sont présentées dans la partie suivante et pour les comprendre dans le contexte, une présentation succincte des versions précédentes de YOLO est maintenant effectuée.

YOLO-V1 : YOLO-V1 [111] utilise le framework Darknet, réseau basé sur une architecture composée de 24 couches convolutives suivies de 2 couches entièrement connectées. Les 20 premières couches convolutives du réseau ont été utilisées pour un pré-entraînement avec 1000 classes de ImageNet. Quatre couches convolutives et deux couches entièrement connectées sont ajoutées au modèle pour le transformer en un réseau de détection. Les couches convolutives initiales du réseau permet l'extraction de caractéristiques de l'image, appelées features dans la littérature, tandis que les couches entièrement connectées prédisent les probabilités et les coordonnées de sortie. Cette architecture de réseau est inspirée du modèle GoogLeNet pour la classification des images [116], avec une utilisation des couches de réduction 1×1 suivies de couches convolutives 3×3 .

Quelques limitations sont apparues lors de la détection par YOLO-V1, comme la non détection des petits objets à proximité qui est due aux fortes contraintes spatiales imposées lors de la génération des cadres dans les cellules. On note aussi des difficultés de détection des objets ayant de nouvelles configurations ou des configurations inhabituelles par rapport à celles utilisées lors de l'apprentissage.

YOLO-V2 : [114] Cette version dénommée YOLO9000 peut détecter plus de 9000 catégories d'objets. Les auteurs de cette deuxième version de YOLO l'ont surnommée «YOLO9000 Better, Faster and stronger». Les améliorations apportées par YOLO-v2 visent à avoir une meilleure détection, au sens de la rapidité de réponse et du nombre de classes. Elle permet aussi d'obtenir une précision élevée sur des tailles différentes des images d'entrées tout en limitant les erreurs de localisation constatées avec YOLO-V1. L'une des innovations introduite dans YOLO-V2 pour atteindre ces objectifs est l'introduction de cadres d'ancrage. La première version de YOLO génère 98 cadres par image tandis que YOLO-V2 génère plus d'un millier de cadres avec les boîtes d'ancrage utilisées.

YOLO9000 utilise le framework Darknet-19 s'appuyant sur une architecture composée de 19 couches convolutives et 5 couches de pooling max. YOLO-V2 a associé la classification et la détection d'objets pour l'entraînement du réseau. Pour cela deux ensembles de données ont été fusionnés : des classes d'images ImageNet de la classification d'objets et des classes d'images de COCO de détection d'objet, afin que le réseau neuronal entraîné puisse faire la détection d'objets d'un grand nombre de classes d'objets.

YOLO-V3 : Cette troisième version [117] apporte d'autres modifications supplémentaires permettant d'améliorer la précision de détection et la précision du positionnement à vitesse rapide. L'algorithme YOLO-V3 introduit le principe FPN (Feature Pyramid Networks) pour réaliser une prédiction multi-échelles et utilise des idées de réseau résiduel profond (ResNet) pour extraire des caractéristiques d'image afin d'atteindre un certain équilibre entre la vitesse de détection et la précision.

L'architecture de YOLO-V3, représentée dans la figure 3.4, est un réseau profond sans couches entièrement connectées. Le réseau utilise le framework Darknet-53 plus puissant que Darknet-19, disposant de 53 couches entraînés sur ImageNet pour la tâche de détection, aboutissant au final à une architecture entièrement convolutive de 106 couches. Comme la figure 3.4 le montre, l'architecture YOLO-V3 contient des couches de convolution, des blocs résiduels et des couches sur-échantillonnées.

YOLO-V3 décompose les images d'entrée en unités de grille $S \times S$ suivant le principe d'origine des versions précédentes de YOLO. Des couches convolutives successives 3×3 et 1×1 sont utilisées comme extracteur de caractéristiques. Sachant que la structure du réseau est plus profonde, l'extraction des caractéristiques passent d'un niveau bas à un niveau haut, c'est-à-dire d'informations sur les couleurs et les bords à des informations sémantiques sur la catégorie d'objet. La dernière couche d'entités est alimentée en informations de prédiction d'images d'entrée coordonnées de la boîte englobante, score d'objectivité et confiance de classe. L'image d'entrée est sous-échantillonnée de 32, 16 et 8 respectivement. Au niveau de la première couche de détection (échelle 1) qui correspond à la couche 82 du réseau, la carte des caractéristiques est sous-échantillonnée d'un pas de 32 X pour détecter les gros objets. Au niveau de la deuxième couche de détection (échelle 2) correspondant à la couche 94, la carte des caractéristiques est sous-échantillonnée d'un pas de 16 X pour détecter les objets

de taille moyenne. La troisième couche de détection (échelle 3) correspond à la dernière couche 106, la carte des caractéristiques est sous-échantillonnée d'un pas 8 X pour détecter les objets de petite taille. Si un centre d'objet tombe sur une cellule de la grille, cette cellule représente la prédiction de l'objet et son score d'objectivité est égal à 1, le reste des cellules étant à 0. Ce même concept est utilisé lors de l'entraînement du réseau YOLO-V3 pour identifier la cellule qui représente le centre de l'objet. Chaque cellule de la grille génère trois cadres de tailles différentes appelés « Anchors ». La taille du cadre approprié à l'objet est défini en se basant sur la méthode de regroupement K-mean ainsi que sur l'ensemble de données d'apprentissage. Enfin, la fusion d'entités est utilisée pour prédire l'objet au niveau des trois dernières couches sous-échantillonnées avec trois échelles en tant que cartes d'entités à différentes profondeurs. YOLO-V3 modifie la taille des cartes d'entités de la couche profonde par sur-échantillonnage. Ensuite, la taille des cartes d'entités à différentes échelles sera la même. Les entités de la couche précédente sont fusionnées avec les entités de la couche de sur-échantillonnage par concaténation pour obtenir des prédictions précises.

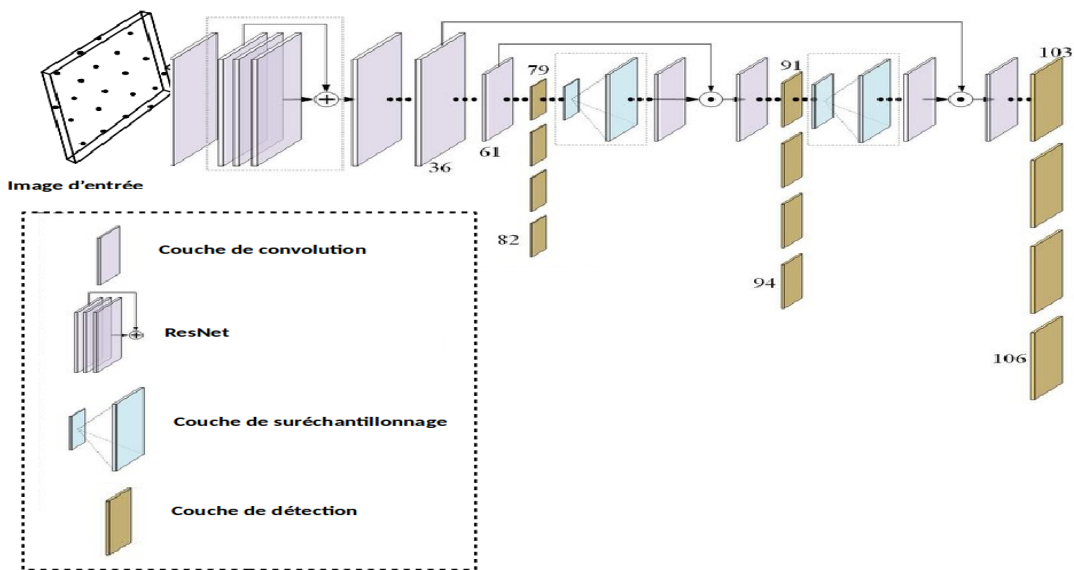


FIGURE 3.2: L'architecture du réseau de neurones YOLO-V3 dans notre cas d'études

YOLO-V4 : La 4e itération de YOLO [115] améliore l'architecture du réseau neuronal afin de rendre le système plus précis. L'architecture de YOLO-V4 est composée de trois

parties comme illustré dans la figure 3.3 : backbone, neck et head. La partie backbone est le coeur du réseau ; elle consiste à extraire les caractéristiques de l' image d' entrée. Le backbone est déjà entraîné séparément sur les cadres de classification d' image avec une base de milliers d' images de différents contenus. Cette variété des images permet au backbone d' apprendre à partir de divers features, de taille, de couleur, de forme ou d' autres. La deuxième partie de l' architecture de YOLO-V4 est le neck. Son objectif est la combinaison des features extraites par les différentes couches du backbone pour une détection de divers objets. La dernière partie dite head se base sur les informations fournies par les autres parties de l' architecture pour définir l' emplacement, le nom et le cadre de chaque objet. Deux concepts d' améliorations ont été apportés sur YOLO-V4 « bag of freebies » et « bag of specials ». Le bag of freebies regroupe l' ensemble des méthodes pour des modifications entrainant à une meilleure précision sans augmenter le coût d' inférence. Ces modifications reliées à la stratégie d' apprentissage comme l' augmentation des données, la cross mini-batch normalization et la fonction objective de la régression. Il n' y a pas un changement de l' architecture de base du réseau mais des modifications spécifiques de l' architecture du réseau au niveau du bag of specials, notamment la fonction d' activation Mish, le SPP (spatial pyramid pooling) et PAN (Path Aggregation Network for Instance Segmentation).

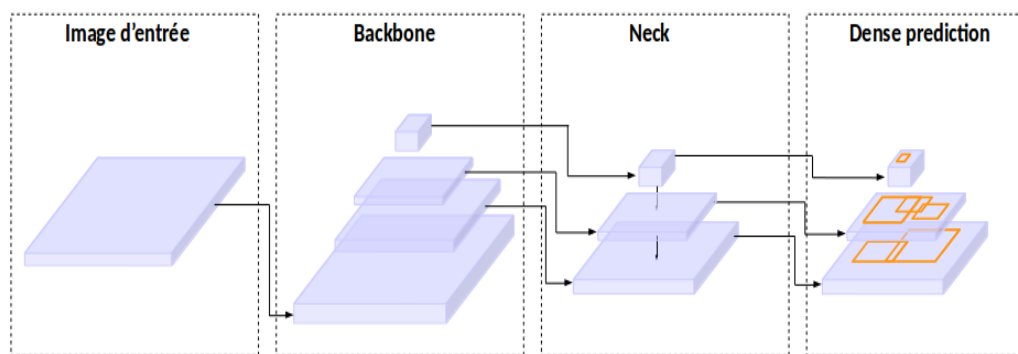


FIGURE 3.3: L'architecture du réseau de neurones YOLO-V4 dans notre cas d' études

3.1.2 L'apprentissage par transfert

La formation d'un réseau de neurones convolutionnel nécessite de créer le modèle de réseaux de neurones convolutifs, d'avoir une grande quantité de données de l'ordre de milliers d'images, d'avoir un ordinateur avec une mémoire capable de stocker plusieurs gigaoctets de données et avec un processeur graphique performant de caractéristiques efficaces pour assurer les calculs parallèlement (par exemple une dizaine de TFLOPS et multiple Cœurs). La durée d'entraînement de ce modèle peut prendre des jours. Pour les petits projets qui contiennent un nombre limité de données, l'apprentissage par transfert représente une solution efficace [118]. L'approche d'apprentissage par transfert utilise les couches convolutives dans le réseau pré-formé des connaissances acquises avec des fonctionnalités utiles de haut et bas niveau comme point de départ et ajoute de nouvelles couches pour une tâche spécifique. En apprentissage par transfert il existe généralement deux stratégies courantes : Feature extractor and fine-tuning.

Feature extractor

L'extraction des features appelé parfois dans la littérature «off-the-shelf CNN» [119]. Cette stratégie exploite les features apprises par le réseau pré-entraîné pour extraire des features des nouvelles images et alimenter le nouveau classifieur. Le modèle est alors constitué d'un réseau convolutif et d'un nouveau classifieur. Le réseau convolutif fournit un vecteur de features pour chaque image d'entrée qui est utilisé pour entraîner un nouveau classifieur. Le temps d'apprentissage est considérablement réduit en raison des features déjà extraites. Cette stratégie est utilisée lorsque le nombre d'images de la nouvelle collection est limité et similaire aux images de pré-entraînement.

Fine-tuning

Fine-tuning est l'une des techniques populaires en apprentissage automatique qui permet de se servir du réseau pré-formé comme initialisation pour un nouveau entraînement sur une nouvelle collection d'images. Cette stratégie consiste à profiter des capacités d'extraction des features apprises lors du premier entraînement. Il s'agit de fixer les poids des couches convolutives pré-entraînées et de remplacer les dernières couches du modèle qui

contrôlent la sortie par des nouveaux classifieurs et réentraîner le réseau. Notre travail s'est basé sur l'apprentissage par transfert précisément par fine-tuning afin d'exploiter la technique d'extraction des features de YOLO pour entraîner le réseau avec notre propre base de donnée.

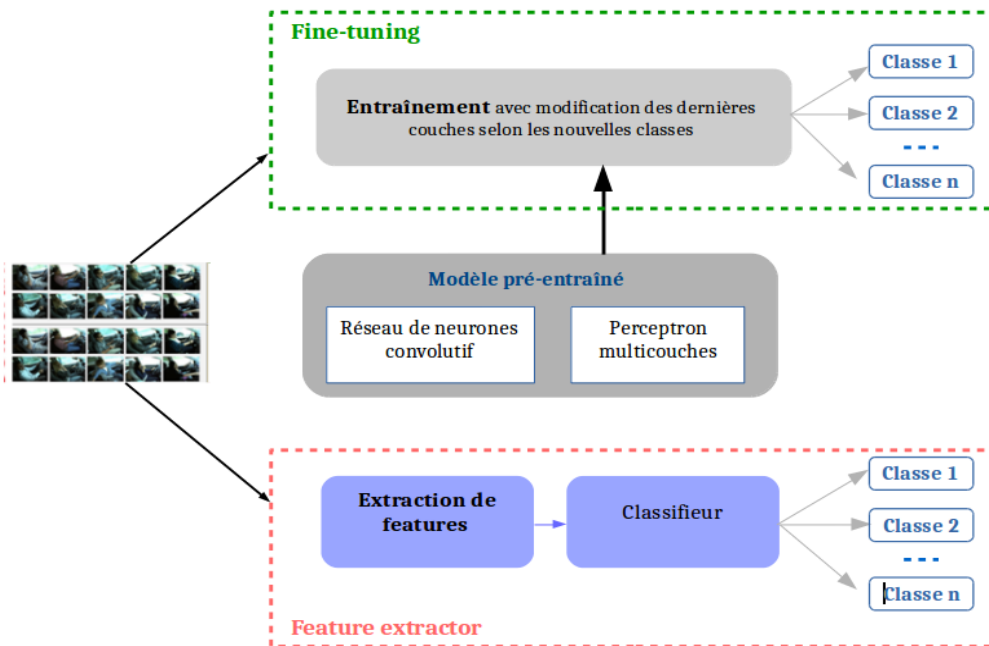


FIGURE 3.4: Les principales stratégie de transfert learning (a) Feature extractor, (b) Fine-tuning.

3.2 Le cas d'étude GREAH : Saisie de pièces par préhension adaptative

3.2.1 Les objectifs

Comme expliqué précédemment, la saisie de la pièce à usiner dans le stock représente la première partie du scénario global. Cette pièce industrielle est en plastique, de couleur unie, sans texture et présente des faces de caractéristiques différentes (Figure 3.6). Cette pièce conçue à façon avec des géométries différentes sur chaque face est utilisée pour illustrer la

variété des pièces que le robot sera amené à manipuler. Les différentes faces représentent cinq classes principales de composants industriels illustrées dans la figure 3.5.



FIGURE 3.5: Quelques produits industriels pouvant être saisis.

Notre travail de recherche a consisté à mettre en oeuvre un système de pick-and-place robotisé avec une préhension adaptative en se basant sur une approche d'apprentissage automatique. La préhension adaptative est réalisée par un préhenseur multifonction à 2 doigts parallèles permettant de saisir l'objet quelle que soit la face présentée, soit par l'ouverture, soit par la fermeture des doigts du préhenseur. L'objectif consiste à choisir le mode de préhension adéquat pour chaque face pour pouvoir saisir le produit quelle que soit la face observée. Le travail est basé sur la perception visuelle de l'objet précisément l'apprentissage automatique pour la détection, l'estimation de pose et la classification de la face de l'objet observée. Les états des doigts du préhenseur sont adaptés selon chaque face de l'objet défini dans l'étape de classification des faces observées [105].

3.2.2 Description de la plate-forme robotique

Les éléments constituant la plate-forme robotique utilisée pour la réalisation de la partie manipulation sont :

- Un bras cobotique « UR5 » de 6 DDL, de charge utile 5Kg, d'une portée 850mm et d'une répétabilité de pose de ± 0.10 mm.
- Un préhenseur à deux doigts parallèles « Robotiq », sous-actionné électriquement, et qui est fixé au poignet du bras robotique.

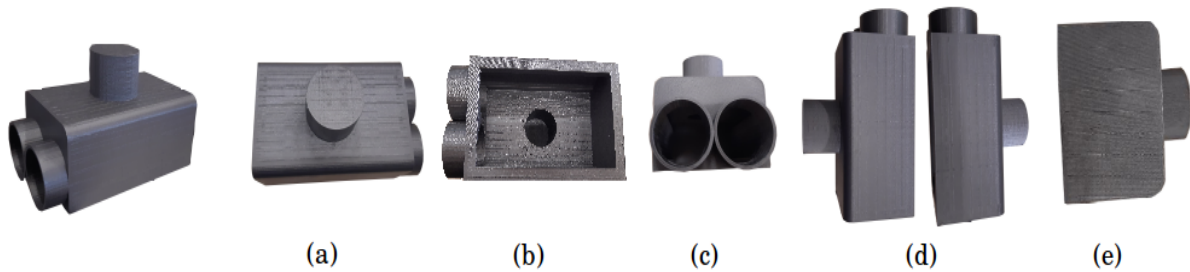


FIGURE 3.6: L'objet conçu ; (a) face supérieure, (b) face inférieure, (c) face droite, (d) face avant / arrière (e) face gauche.

- Deux caméras RGB-D « Astra Orbbec » : une caméra fixée au dessus de la scène avec un champ de vision qui couvre la table contenant la pièce à saisir ; une seconde caméra est placée sur le poignet de l'UR5.

- L'ensemble du bras robotique est placé sur le robot mobile. La figure 3.7 montre l'emplacement des différents éléments. Les expérimentations sont menées sur un ordinateur, NVIDIA Quadro M2000 avec 4 Go de mémoire, CUDA 10.0 et cuDNN v7.

3.2.3 Le processus de préhension d'objet

La préhension est une étape indispensable dans la manipulation robotique. Le processus de préhension fait appel à différentes stratégies qui peuvent être classées en trois catégories [120] : le processus de préhension par changement d'outils, le multi-préhenseur et préhension utilisant des doigts multifonctionnels.

1. La préhension par changement d'outils permet de saisir l'objet avec l'outil le plus adapté possible. Les outils de préhension sont généralement placés dans une station d'outillage et le robot se déplace vers la station pour changer d'outil. Cette solution permet de saisir une large variété d'objets mais avec un temps de cycle long, ce qui explique que cette technique de préhension avec changement d'outil est difficile à mettre en œuvre dans certaines applications.
2. Le multi-préhenseur consiste à installer plus d'un préhenseur sur le poignet du bras robotique. Cette technique permet de saisir différents objets comme par exemple

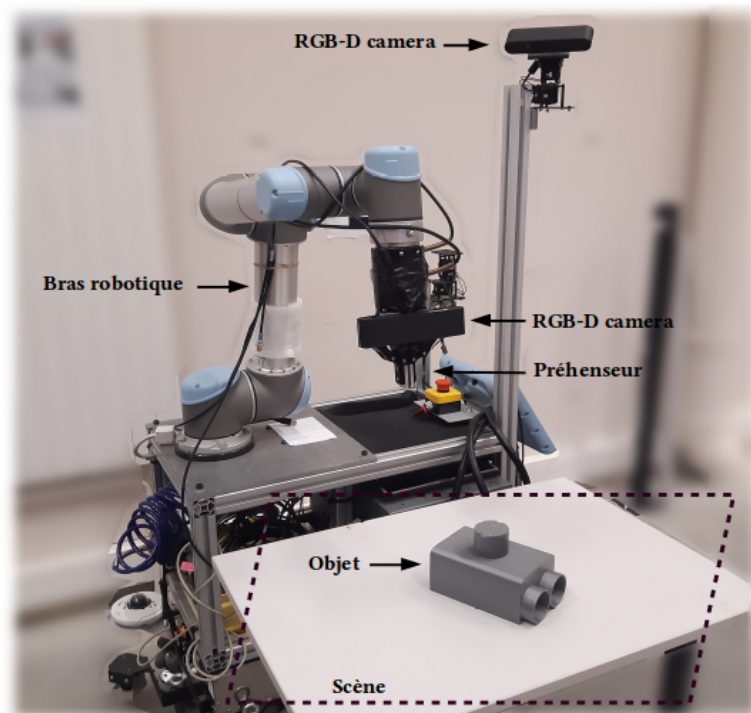


FIGURE 3.7: La plate-forme robotique utilisée.

dans la compétition « Amazon Robotics Challenge » de 2017, où l'un des systèmes robotiques gagnants contenait deux types de préhenseur (une ventouse et un préhenseur à deux doigts)[121]. Cette solution permet de saisir différents objets avec un temps de cycle plus court que la méthode avec changement d'outils.

3. La dernière catégorie est la main robotique multidigitale et multifonction. Le nombre de doigts peut varier de deux à huit. Notre système de préhension avec un préhenseur à deux doigts s'intègre dans cette catégorie. Il est alimenté électriquement et la saisie s'effectue par ouverture ou fermeture des doigts.

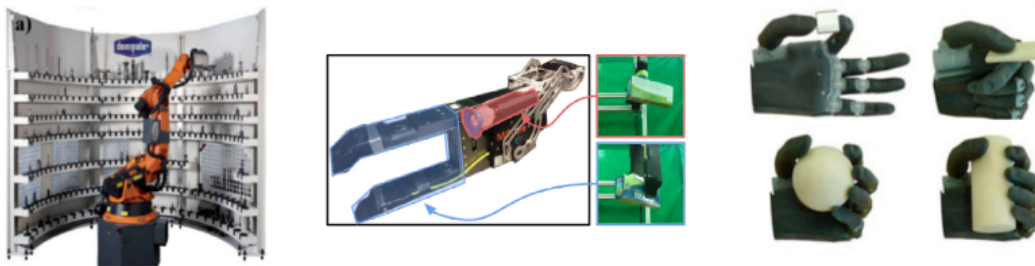


FIGURE 3.8: Illustration des trois stratégies de préhension : (a) par changement d'outils [120], (b) à multi-préhenseur [121], par main robotique multi-doigts [122].

3.2.4 Les étapes de pick-and-place avec préhension adaptative

Le scénario de pick-and-place démarre respectivement lorsque la navigation a déjà eu lieu et que le robot est positionné à proximité de la table. Tout d'abord, les données RGB de la caméra 1 sont utilisées pour la détection d'objet. La caméra 1 est placée au-dessus de la table où la pièce à saisir est stockée. Si la caméra 1 détecte la présence de notre objet fabriqué, l'estimation de la pose sera calculée en utilisant les nuages 3D de points de la même caméra. Ensuite, l'algorithme planifie le chemin optimal du manipulateur pour atteindre l'objet. Avant d'exécuter la préhension et lorsque la caméra du poignet du bras robotique (caméra 2) est au-dessus de l'objet une classification de la face de l'objet observé se fait. À partir de cette classification, l'algorithme prend la décision de préhension de l'objet adaptée à la face présentée.

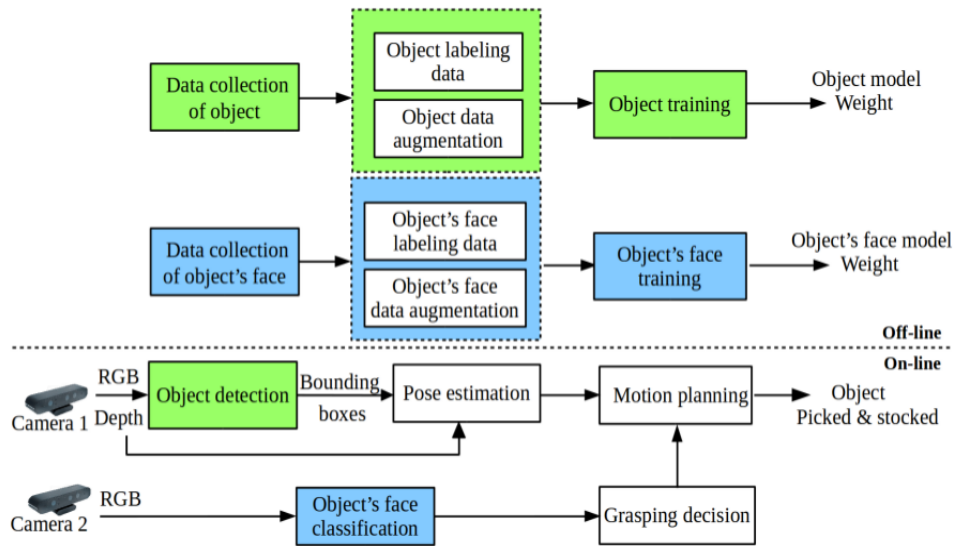


FIGURE 3.9: Architecture de notre système de pick-and-place.

a- La détection d'objet

Dans cette première étape nous vérifions la présence de l'objet dans la scène observée par la caméra 1 au dessus de la scène. Cela est réalisé en se basant sur l'approche YOLO par la technique de transfert learning d'une seule classe. Cette classe correspond à «plastic part». Ce travail a été réalisé dans un premier temps avec la troisième version de YOLO. Après le lancement de la quatrième version de YOLO un autre apprentissage a été effectué avec la nouvelle architecture de YOLO afin d'enrichir le travail par une comparaison des deux résultats obtenus (par YOLO-V3 et par YOLO-V4). Notre apprentissage par transfert learning s'est effectué par une base de données de l'objet dans différentes conditions, une annotation de toutes ces images et un entraînement du réseau neuronal. Ce développement est expliqué en détail dans la section des expérimentations.

b- Estimation de pose

La seconde étape est le calcul de la position de l'objet par rapport au robot. L'approche proposée consiste à utiliser le nuage des points de la scène obtenu par la caméra de profondeur 1 à l'aide de la bibliothèque PCL [123] et du rectangle 2D englobant l'objet obtenu

dans l'étape précédente pour générer des hypothèses de localisation d'objets en 3D à partir de [124]. Dans la scène réelle, le rectangle 2D dans l'image couleur contient non seulement l'objet de premier plan, mais également les parties de la scène d'arrière-plan ce qui nécessite de le rétrécir au plus près de l'objet de sorte que la position de l'objet est définie en ajustant un rectangle d'objet autour des points 3D projetés. En outre, la méthode proposée réduit la taille du nuage de points acquis à partir de la caméra RGB-D à la taille de l'objet, en obtenant un pavé 3D. Les informations dans les images couleur et de profondeur sont complémentaires les unes des autres : les données RGB encodent les caractéristiques d'apparence visuelle distinctives déduites par le rectangle 2D, les points 3D transmettent les structures géométriques de la surface de l'objet. Finalement l'estimation de la pose de l'objet est obtenue en calculant le centroïde de l'objet.

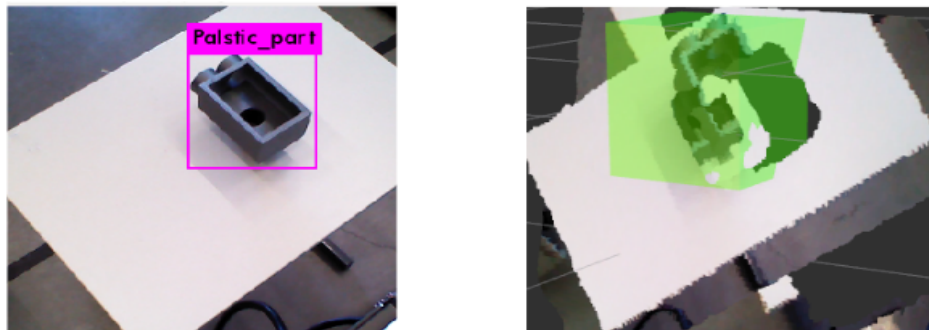


FIGURE 3.10: Le cadre 2D de l'objet détecté (à gauche), le cadre 3D de l'objet détecté dans le nuage des points (à droite).

c- Planification de trajectoire du bras robotique

Le déplacement du bras robotique est réalisé par Moveit!. Après la détection et l'estimation de la pose de la pièce dans la scène, le bras robotique se déplace au-dessus de la table pour observer la face de l'objet vue par la caméra 2 et la classer afin de déterminer le mode de préhension adapté. Le générateur de trajectoire est alors utilisé pour amener le robot au niveau de la pose identifiée pour saisir l'objet. Enfin, le bras robotique se déplace pour déposer la pièce sur la plate-forme du robot mobile. La figure 3.11 illustre les trajectoires générées par Moveit! pour déplacer le robot au-dessus de la table, puis saisir et déposer

la pièce. La simulation des trajectoires est réalisée dans la plate-forme de simulation RVIZ sous ROS.

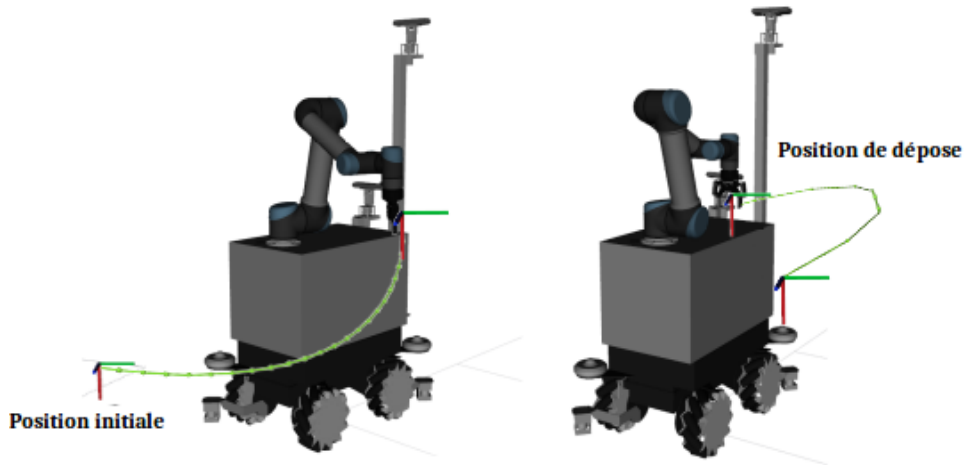


FIGURE 3.11: Illustration des trajectoires suivies par le bras robotique dans RVIZ

d- Décision du mode de préhension adaptative

La préhension adaptative de la pièce conçue est une étape indispensable pour assurer une prise et une dépose correcte. Cette fonction est réalisée par une pince adaptative à 2 doigts, qui peut prendre deux états ouverts ou fermés. La décision d'ouverture ou de fermeture des doigts du préhenseur est basée sur l'apprentissage automatique de reconnaissance de la face présentée par la pièce, puis l'association d'un état des doigts du préhenseur permettant la saisie de chaque face de l'objet par ouverture ou fermeture de la pince. La figure 3.12 présente quelques exemples de saisie de la pièce suivant la face qui se présente.

3.3 Les résultats expérimentaux

La détection de la pièce et la classification de sa face observée sont effectuées par YOLO-V3, précisément par la technique d'apprentissage par transfert. L'entraînement du réseau de neurones est réalisé deux fois : Une fois pour la détection d'objet avec une seule classe «Plastic Part» et le deuxième entraînement pour la classification des faces de l'objet

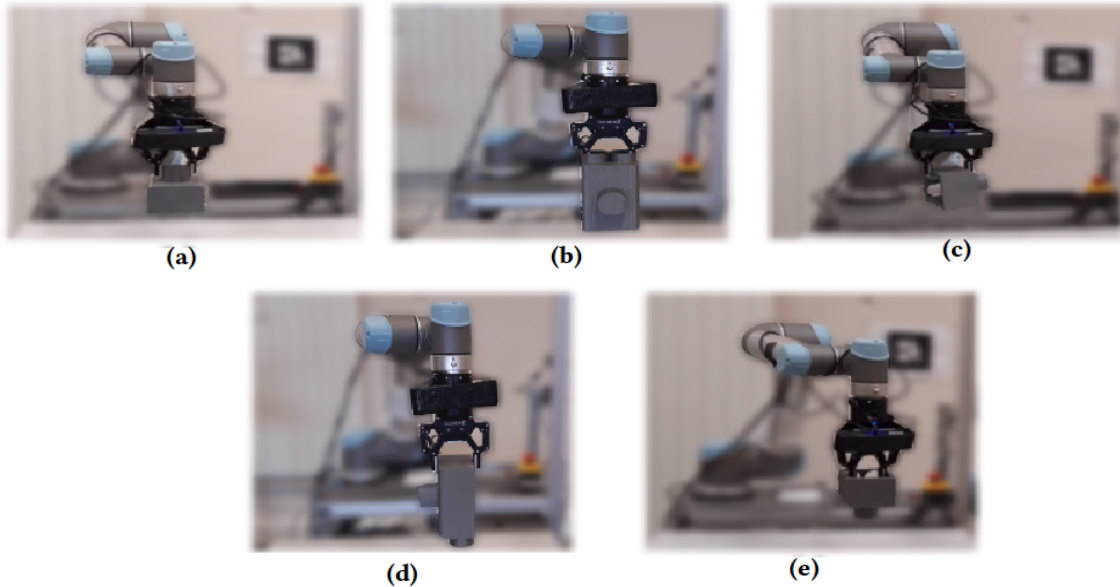


FIGURE 3.12: Le système de préhension adaptative proposé : le bras du robot saisit l'objet fabriqué à partir de la face (a) supérieure, (b) droite, (c) avant / arrière, (d) gauche, (e) inférieure

avec cinq classes comme montré dans la figure 3.5. L'apprentissage par transfert nécessite en premier de créer de nouveaux ensembles d'images pour des catégories d'objets spécifiques puis d'entraîner le réseau.

Collection de la base de données et son augmentation

Dans cette étape nous avons collecté trois sortes de photos : des images synthétiques de l'objet, des images réelles de l'objet et des images réelles qui correspondent aux faces de l'objet. Les deux premiers sous-ensembles d'images ont été utilisés pour l'entraînement du réseau neuronal pour la détection d'objets. L'entraînement est réalisé deux fois séparément, une fois avec l'ensemble d'images synthétique et la deuxième fois avec un ensemble d'images réelles de l'objet. Un troisième entraînement du modèle a été fait avec des images des différentes faces de l'objet pour l'étape de la classification des faces de la pièce à saisir.

Les images synthétiques de l'objet : Plus de 900 images de l'objet ont été générées à partir de son modèle 3D en utilisant Blender interfacé par Python. Le modèle 3D de l'ob-

jet a été conçu avec les différentes caractéristiques géométriques comme montré dans la figure 3.6 : cylindre, trou rectangulaire, trous circulaires et rectangle plein. Ces caractéristiques géométriques sont combinées dans le modèle 3D et représentées dans les différentes faces de l'objet. L'objet final a été imprimé par une imprimante 3D afin de pouvoir tester le système sur des images réelles ainsi que pour la réalisation du scénario réel. La figure 3.13 montre les images de la pièce prises dans Blender avec un positionnement multi-échelle et multi-vue et un arrière-plan encombré qui change dynamiquement. La caméra utilisée dans Blender a été commandée pour se déplacer sur une sphère autour de la scène, pointant vers le centre de la scène afin d'avoir des images du modèle 3D selon différents angles. Le facteur de distance entre chaque position de caméra est réglé de manière aléatoire tandis que l'angle de rotation est compris entre $[- 60^\circ : 60^\circ]$ avec un pas de 3 degrés. Le modèle 3D était éclairé par la lumière ambiante et une source de lumière ponctuelle qui pouvait être positionnée sur une sphère autour de la scène pour fournir une précision raisonnable.

Les images réelles de l'objet : L'ensemble d'images réelles de l'objet a été collecté via la caméra RGB-D Astra Orbbec et des caméras de téléphone portable avec différentes résolutions et selon différents angles et positions. Les conditions d'éclairage ont été variables. Les images ont été prises avec différents emplacements de l'objet et dans des environnements encombrés. 180 images ont été capturées pour l'objet qui représente le modèle imprimé en 3D.

Les images des faces de l'objet : les images réelles des différentes faces géométriques du modèle imprimé ont été collectées de la même façon que celle des images réelles de l'objet. Chaque face géométrique de l'objet a été considérée comme une classe à part c'est-à-dire les images de chaque face sont prises en tenant en compte les conditions d'éclairage, d'arrière-plan et avec différentes caméras (Astra Orbbec et des caméras de téléphone). Un total de 600 images est constitué avec 120 images de chaque face.

Une augmentation des données a été effectuée sur des images réelles de l'objet et de ses côtés géométriques en appliquant une mise à l'échelle aléatoire, une rotation, etc. Le nombre d'images a été augmenté au total de 996 images pour la détection d'objets et de 886 pour chaque côté géométrique de l'objet pour la classification des faces. L'ensemble des images synthétiques et réelles avec augmentation sont annotées manuellement à l'aide

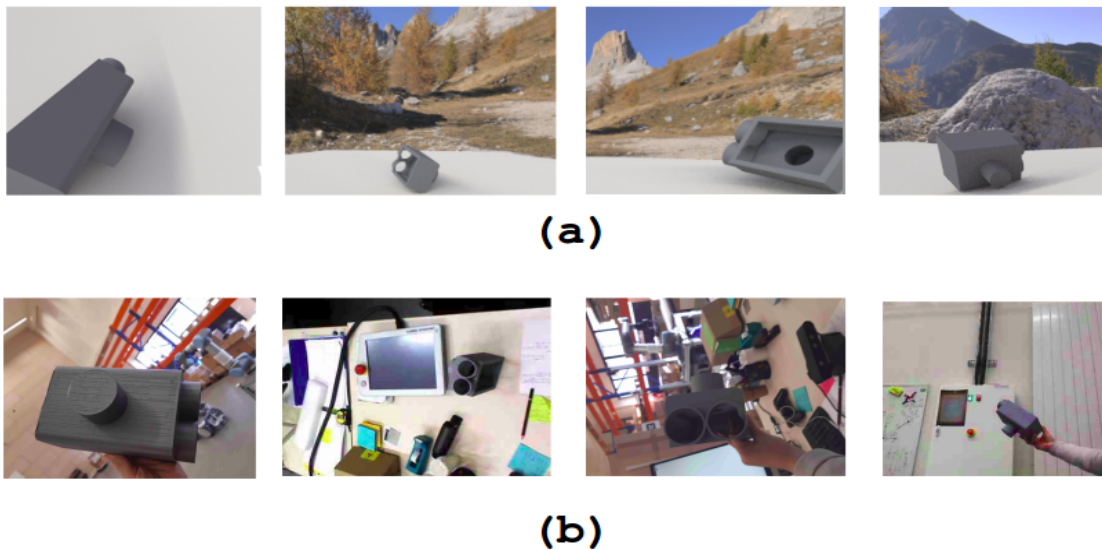


FIGURE 3.13: Des exemples des images (a) synthétiques (b) réelles

de LabelMe [125] pour six classes. Une classe qui correspond à l'objet lors de la détection sous le nom de « plastic part», les cinq autres classes correspondent aux faces de l'objet sous les noms : «plastic part :upface», «plastic part :downface», « plastic part :front/backface», «plastic part :rightface».

Entraînement du réseau neurones

Les réseaux de neurones ont été entraînés en utilisant 60% des images comme des données d'apprentissage et les 20% restants des images pour le test et la validation. YOLO-V3 open source basé sur le réseau neuronal Darknet a été utilisé dans Colab avec GPU pour notre entraînement. Ce processus a été exécuté deux fois pour la détection d'objets ; une fois basée sur des ensembles de données synthétiques et une seconde fois sur du réel, les deux pour une classe qui est « Plastic part ». Les entraînements à la détection d'objets ont duré environ 8 à 10 heures. Le troisième entraînement du réseau portait sur la classification des côtés de l'objet en se basant sur les données réelles de cinq classes qui ont nécessité environ 40 heures d'entraînement du réseau.

Dans la même plate-forme Colab deux autres entraînements ont été réalisés cette fois-ci par YOLO-V4. Un entraînement pour la détection de la pièce qui a duré environ 5 heures.

Le deuxième entraînement pour la classification de ses faces avec une base d'images plus importante que l'entraînement de YOLO-V3.

Evaluation de la détection d'objet et de la classification de ses faces

L'évaluation des résultats obtenus pour la détection et la classification a été basé sur la matrice de confusion pour identifier les 4 métriques d'évaluation suivantes : précision, rappel, F1-Mesure et FP-rate [126] :

La matrice de confusion est un outil de mesure des performances d'un modèle d'apprentissage automatique, qui sert à vérifier la fréquence de prédictions exactes et fausses par rapport à la réalité. La matrice est composée de quatre éléments : TP, FP, FN et TN. TP (True Positive) est le nombre de fois où l'objet est présent et correctement détecté dans la scène, FP (False Positive) est le nombre de fois où l'objet est détecté dans la scène sans être présent réellement, FN (False Negative) est le nombre de fois où l'objet est présent sans être détecté et TN (True Negative) est le nombre de fois où l'algorithme détecte l'absence de l'objet qui n'est effectivement pas présent, étant donné que dans tous les cas, d'autres objets similaires peuvent être présents dans la scène observée.

La précision est le taux des prédictions positives correctes parmi les résultats positifs comprise entre 0 et 1, et dont la performance optimale s'approche de 1 :

$$Précision = \frac{TP}{TP + FP}$$

Le rappel est le taux des vrais positifs sur l'ensemble des cas où l'objet est présent, compris entre 0 et 1, et dont la performance optimale s'approche de 1 :

$$Rappel = \frac{TP}{TP + FN}$$

F1_Mesure est la moyenne harmonique de la précision et du rappel pour combiner les deux paramètres en une seule mesure, comprise entre 0 et 1, et dont la performance optimale s'approche de 1 :

$$F1_Mesure = 2 \times \frac{Précision \times Rappel}{Précision + Rappel}$$

FP_rate est le taux de faux positifs sur l'ensemble des cas où l'objet est présent, compris entre 0 et 1, et dont la performance optimale est proche de 0 :

$$FP_rate = \frac{FP}{TP + FN}$$

Pour le calcul de ces métriques, un ensemble de tests de détection d'objet et de classification de ses faces géométriques ont été effectués dans des conditions différentes : un espace de travail dégagé autour de l'objet ou avec occlusion partielle par d'autres éléments presque de même couleur et texture, et avec différentes conditions d'éclairage. Ces tests ont été mis en œuvre pour évaluer l'entraînement à la détection d'objets des ensembles de données synthétiques et réels ainsi que l'entraînement à la classification des faces de l'objet.

Les tableaux suivants montrent les résultats de calcul des métriques d'évaluation. Ces scores ont été calculés en capturant une image par seconde (fps) pendant 20 minutes tandis que la vitesse de détection est de 37.4 fps, avec une prise en compte de la variation de position de l'objet fabriqué dans un espace de travail encombré avec d'autres objets de mêmes couleurs et textures.

TABLE 3.1: Les scores d'évaluation de la détection d'objet basée sur l'entraînement du réseau par des images synthétiques (YOLO-V3).

	Précision	Rappel	F1_Mesure	FP_rate
Condition Normale	0.648	0.786	0.711	0.426
Variation d'éclairage	0.589	0.716	0.646	0.500
Occlusion	0.581	0.671	0.623	0.484

TABLE 3.2: Les scores d'évaluation de la détection d'objet basée sur l'entraînement du réseau par des images réelles (YOLO-V3).

	Précision	Rappel	F1_Mesure	FP_Rate
Condition Normale	0.982	0.859	0.916	0.015
Variation d'éclairage	0.961	0.781	0.862	0.031
Occlusion	0.959	0.723	0.824	0.030

TABLE 3.3: Les scores d'évaluation de la détection d'objet basée sur l'entraînement du réseau par des images réelles (YOLO-V4).

	Précision	Rappel	F1_Mesure	FP_Rate
Condition Normale	0.896	0.987	0.905	0.195
Variation d'éclairage	0.846	0.905	0.875	0.164
Occlusion	0.894	0.883	0.888	0.104

D'après l'ensemble des expérimentations de détection d'objet dans différentes conditions, nous remarquons que les fausses détections sont apparues plus souvent lors de la détection d'objets basée sur un entraînement des données synthétiques ce qui a mené à des faibles résultats en précision (voir tableau 3.1). Le reste des entraînement des réseaux YOLO-V3 et YOLO-V4 pour la détection d'objet a été réalisé avec des images réelles. Les résultats ont montré une amélioration sur la précision, une diminution du nombre de manque de détection (FN) sur la détection d'objet YOLO-V4 par rapport à la détection d'objet YOLO-V3 (voir tableau 3.2 et tableau 3.3). La figure 3.14 montre des exemples des résultats de détection d'objet des différents entraînements du réseau.

Pour la classification des faces les résultats d'entraînement du réseau YOLO-V3 et YOLO-V4 sont proches avec une diminution du nombre de manque de détection des faces (FN) sur la classification avec YOLO-V4 ce qui a entraîné une amélioration des résultats rappel et F1_Mesure. Nous pouvons aussi conclure d'après nos expériences que la précision du réseau YOLO-V3 est meilleure que celle obtenue avec YOLO-V4 en raison de l'apparition de cas de fausses détections plus nombreuses avec YOLO-V4 (voir tableau 3.4 et tableau 3.5). La figure 3.15 et 3.16 montrent des exemples des résultats de classification des côtés de l'objet.

Évaluation du pick-and-place

L'évaluation du système de pick-and-place proposé a été mesurée par le taux de réussite de l'ensemble du scénario. Notre système de picking commence par la détection de l'objet dans l'espace de travail, puis l'estimation de la pose de l'objet, la planification du mouvement du bras du robot pour atteindre une position au-dessus de l'objet pour classifier la face

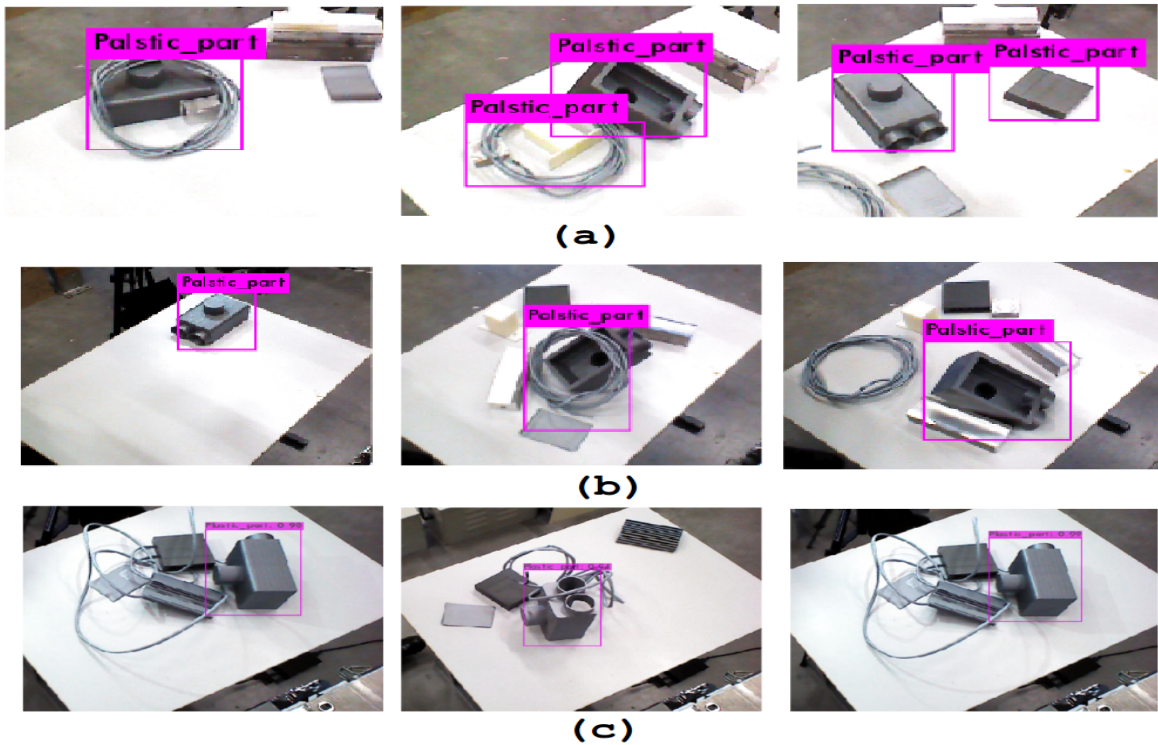


FIGURE 3.14: Résultats de détection d'objet avec un entraînement du réseau neurones YOLO-V3 (a) et (b) et YOLO-V4 (c); (a) sur une dataset synthétique et (b) et (c) sur une dataset réelle.

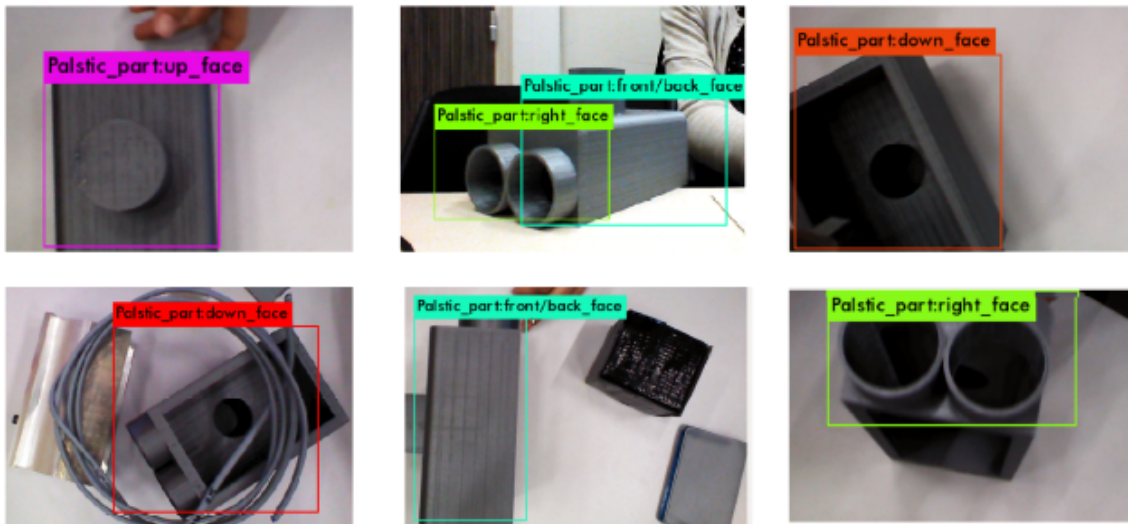


FIGURE 3.15: Résultats de classification des faces géométriques (YOLO-V3)

TABLE 3.4: Les scores d'évaluation de la classification des faces géométriques basée sur l'entraînement du réseau par des images réelles (YOLO-V3).

	Précision	Rappel	F1_Mesure	FP_Rate
Condition Normale	0.981	0.830	0.900	0.015
Variation d'éclairage	0.961	0.781	0.862	0.031
Occlusion	0.979	0.723	0.831	0.015



FIGURE 3.16: Résultats de classification des face géométriques (YOLO-V4)

géométrique observée par la caméra du poignet, la décision du mode de préhension et enfin la prise de l'objet et le stockage sur la plate-forme du robot (voir figure 3.17). Pour évaluer les performances du taux de réussite (G_s), le pourcentage du nombre d'essais fructueux a été calculé : $G_s = Gr/Nr$, où Gr est le nombre de succès et Nr est le nombre de répétitions ($Nr = 6$ essais par face).

Le calcul du taux de réussite de notre pick-and-place en se basant sur le réseau de neurones YOLO-V3 est réalisé sur 6 essais par face de la pièce dont nous changeons à chaque essai l'emplacement de la pièce dans la scène pour évaluer la robustesse de notre système. Si nous analysons les raisons pouvant expliquer que la détection l'objet a échoué, il s'agit de cas où l'objet ne rentrait pas complètement dans le champ de vision de la caméra. Dans

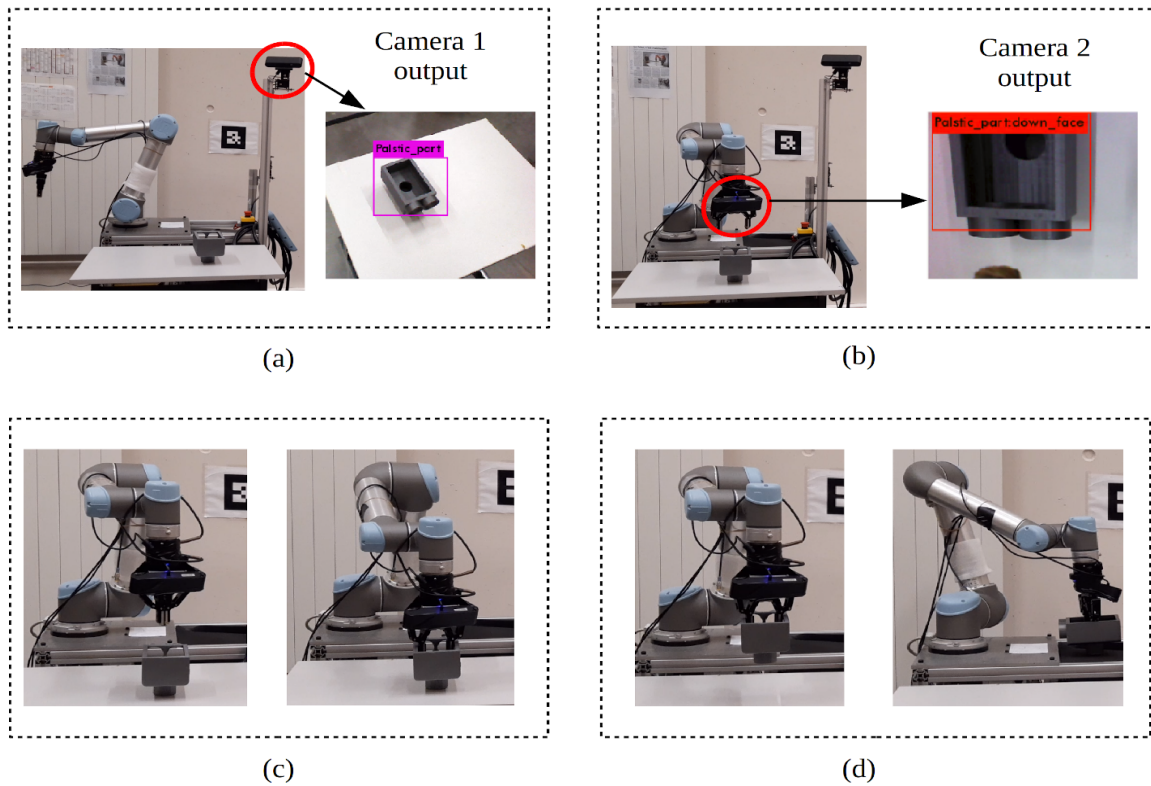


FIGURE 3.17: Les quatre étapes de picking de l'objet multifaces : (a) détection de l'objet et estimation de la pose, (b) classification de la face de l'objet et décision de mode de préhension, (c) saisie de l'objet, (d) planification et exécution du mouvement.

TABLE 3.5: Les scores d'évaluation de la classification des faces géométriques basée sur l'entraînement du réseau par des images réelles (YOLO-V4).

	Précision	Rappel	F1_Mesure	FP_Rate
Condition Normale	0.943	0.930	0.937	0.055
Variation d'éclairage	0.901	0.986	0.941	0.108
Occlusion	0.887	0.875	0.881	0.111

TABLE 3.6: Taux de réussite du scénario de pick and place (YOLO-V3).

La face de l'objet observée	Catégorie de l'objet	Taux de réussite	Taux de réussite Total
Top face	cylindrique	0.833	0.799
Bottom face	trou rectangulaire	0.833	
Front/Back face	rectangulaire	0.833	
Right face	trous circulaires	0.833	
Left face	carré	0.666	

la pratique, ce cas reste peu probable même s'il doit être considéré dans l'analyse de risque d'échec. Il convient alors de pondérer la fiabilité du système par la probabilité d'occurrence ce qui automatiquement fait remonter le taux de fiabilité à des niveaux plus acceptables. Pour

TABLE 3.7: Taux de réussite du scénario de pick and place (YOLO-V4).

La face d'objet observée	Catégorie d'objet	Taux de réussite	Taux de réussite Total
Top face	cylindrique	0.833	0.833
Bottom face	trou rectangulaire	0.833	
Front/Back face	rectangulaire	0.833	
Right face	trous circulaires	0.833	
Left face	carré	0.833	

comparer les performances du pick-and-place basé sur le réseau de neurones YOLO-V3 et YOLO-V4, nous avons réalisé 6 essais pour chaque face de pièce et obtenu des résultats très proches. Le système assure la détection de la pièce, le classement de la face observée et la

préhension adaptative dans tous les cas où la pièce est bien positionnée dans la scène, et les cas d'échecs sont liés au fait que la pièce n'est pas entièrement dans le champ de vision de la caméra placée sur le mât du robot mobile. Nous pouvons donc conclure que notre système est suffisamment fiable pour être applicable dans les lignes de production.

Chapitre 4

Modélisation et évaluation des incertitudes de pose d'un manipulateur mobile

Table des matières

4.1	État de l'art sur l'alimentation robotisée des machines	92
4.2	Modélisation de la précision de positionnement d'un manipulateur mobile	94
4.2.1	Modélisation de la précision du bras manipulateur	95
4.2.2	Modélisation de la précision du robot mobile	99
4.2.3	Modélisation de la précision du manipulateur mobile	101
4.3	Stratégie optimale pour un assemblage contraint	105

Ce chapitre s'intéresse à la dernière partie du scénario qui consiste à déposer la pièce transportée dans la machine-outil. Nous avons développé dans cette partie une modélisation des incertitudes de pose de la pièce dans la machine permettant de choisir la stratégie d'asservissement optimale pour la réalisation de la tâche de dépose en fonction des contraintes fonctionnelles de l'assemblage. À la fin du chapitre, quatre stratégies sont présentées suivant que les contraintes d'insertions sont plus ou moins fortes.

4.1 État de l'art sur l'alimentation robotisée des machines

L'alimentation automatique des machines-outils peut se réaliser au sein même des centres d'usinage les plus innovants qui intègrent toutes les étapes nécessaires de fabrication d'une pièce depuis sa conception jusqu'à l'usinage final. C'est ce paradigme qui est proposé par les centres d'usinage japonais KITAMURA [127] disposant de magasins de grande taille et de bras robotiques. Mais cette alimentation automatique peut se réaliser aussi par des robots externes au centre d'usinage ou à la machine de test et permet de conserver des machines plus anciennes en les intégrant dans un atelier Industrie 4.0. Cette introduction de manipulateurs mobiles pour effectuer ces tâches de manipulation est encore peu répandue et fait l'objet d'études scientifiques à forte valeur ajoutée.

Les études dans ce contexte présentent des solutions très spécifiques au secteur applicatif concerné comme dans l'article [128], où le bras robotique est placé sur des rails de guidage et les machines d'usinage sont placées autour de ces rails. Le bras robotique se déplace entre les machines en suivant les rails et s'arrête devant chaque machine. Une autre solution proposée consiste à faire coopérer des bras robotiques à base fixe et des robots mobiles, formant ainsi un système multirobots. Les bras robotiques assurent le picking, la charge et la décharge des pièces à partir de la base mobile tandis que les robots mobiles transportent les pièces d'une zone de l'atelier à une autre [129]. Dans ces deux solutions, les bras robotiques sont installés sur des supports fixes ce qui assure une très bonne répétabilité de pose, bien supérieure à celle obtenue lorsque le bras est fixé sur un support mobile.

La précision du système robotique complet base mobile et bras manipulateur n'a pas été encore étudiée de manière approfondie et constitue encore un verrou scientifique freinant l'utilisation plus large des manipulateurs mobiles dans l'industrie 4.0. Aussi le but de notre travail dans cette partie est de réaliser la dépose de la pièce dans la machine-outil en utilisant le manipulateur mobile et ceci en évitant toute collision avec l'environnement de dépose (capot, paroi, mandrin... de la machine-outil) et surtout en déposant la pièce avec une précision suffisante pour respecter le jeu fonctionnel entre la pièce et l'étau permettant son insertion avant serrage. La première étape du processus de dépose consiste à saisir la pièce sur le plateau du manipulateur mobile en utilisant les mêmes techniques que celles

présentées précédemment pour permettre la préhension adaptative (voir figure 4.1)

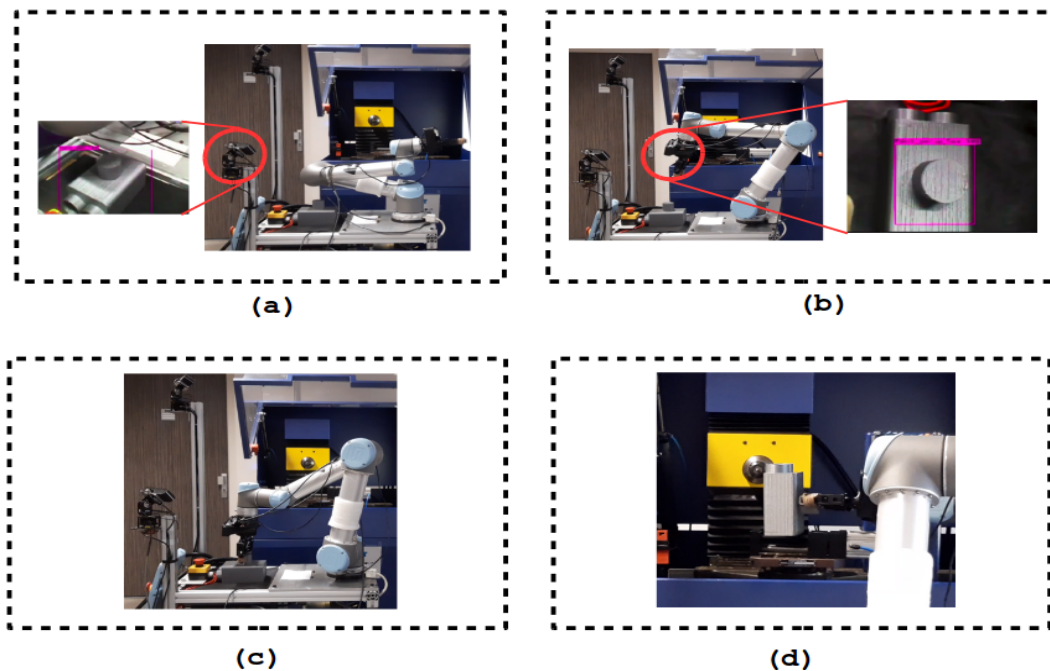


FIGURE 4.1: Les étapes de prise et dépose de l'objet multifaces pour son usinage : (a) détection de l'objet et estimation de la pose, (b) classification de la face de l'objet et décision de mode de préhension, (c) saisie de l'objet, (d) le bras robotique en position de dépose.

La distance maximale d'ouverture de l'étau est de 110 mm entre les mors et la largeur de la pièce à insérer pour l'usinage est de 80 mm. Nous disposons donc de 30 mm de jeu, ce qui à priori peut sembler suffisant. Cependant nous n'avons aucune certitude de réussir à réaliser cette tâche de dépose avec une grande fiabilité. Il est absolument nécessaire de modéliser la précision globale du manipulateur mobile afin de pouvoir estimer ses performances en termes de précision. Nous rappelons que lors de la navigation le robot mobile utilise les signaux LiDAR pour la localisation, puis une fois la position cible finale atteinte, la régulation de la base mobile est effectuée par un asservissement visuel sur un ARtag placé sur le capot de la machine-outil. Cette étude de la précision est indispensable pour choisir entre une stratégie de dépose en boucle ouverte ou par intégration d'une caméra avec champ de vision sur l'étau pour permettre un asservissement visuel lors de la dépose.

4.2 Modélisation de la précision de positionnement d'un manipulateur mobile

Généralement deux indicateurs de performance définissent la précision d'un robot : la répétabilité et l'exactitude de pose. La norme internationale ISO9283 [130] définit ces indicateurs à partir d'une série de 30 essais où le robot est commandé pour atteindre la même cible. Pour chacun de ces essais la pose est mesurée par un dispositif externe et au final, l'ensemble des poses mesurées constitue un nuage de points (voir figure 4.2). Ce nuage des points mesurés est souvent modélisé par une sphère même si en réalité il s'agit plutôt d'un ellipsoïde. La mesure de la dispersion de ce nuage de points autour d'une position moyenne représente la répétabilité du système robotique. L'écart entre la pose commandée et la moyenne du nuage des points définit l'exactitude du robot.

L'étude de la répétabilité d'un robot permet de déterminer l'étroitesse entre les poses atteintes durant les N essais et la pose demandée. La formule de la répétabilité de positionnement s'écrit sous la forme :

$$RP_L^{[N]} = \bar{L}^{[N]} + 3\sigma_L^{[N]} \quad (4.1)$$

où N est le nombre d'essais (N=30), $\bar{L}^{[N]}$ est la moyenne des distances L_j entre chaque point mesuré et le barycentre :

$$\bar{L}^{[N]} = \frac{1}{N} \sum_{j=1}^N L_j \quad (4.2)$$

avec :

$$L_j = \sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2 + (z_j - \bar{z})^2} \quad (4.3)$$

x_j, y_j, z_j sont les coordonnées de la pose mesurée dans chaque essai et $\bar{x}, \bar{y}, \bar{z}$ sont les coordonnées du barycentre de la même pose répétée N fois.

$\sigma_L^{[N]}$ est l'écart-type des distances qui s'écrit sous forme :

$$\sigma_L^{[N]} = \sqrt{\frac{1}{N-1} \sum_{j=1}^N (L_j - \bar{L})^2} \quad (4.4)$$

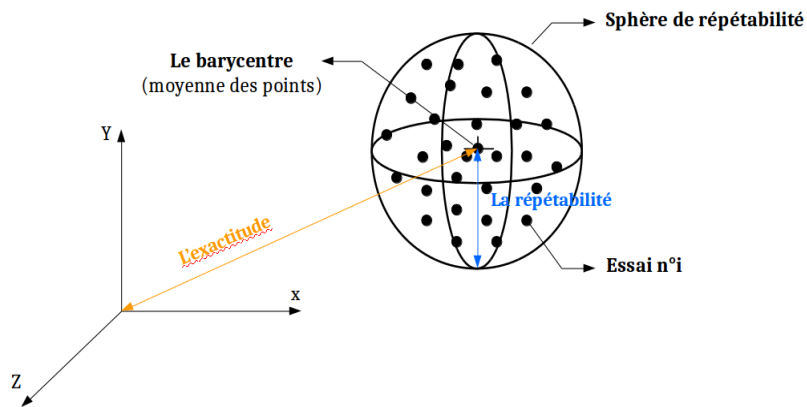


FIGURE 4.2: La répétabilité de positionnement et l'exactitude suivant la norme ISO 9283

L'erreur d'exactitude en orientation est la différence entre l'orientation visée pour la plate-forme robotique et la moyenne des orientations obtenues comme illustré dans la figure 4.3. La répétabilité d'orientation est caractérisée par l'étendue des variations angulaires en roulis, tangage et lacet $\pm 3S_a$, $\pm 3S_b$, $\pm 3S_c$ autour des coordonnées angulaires moyennes \bar{a} , \bar{b} , \bar{c} .

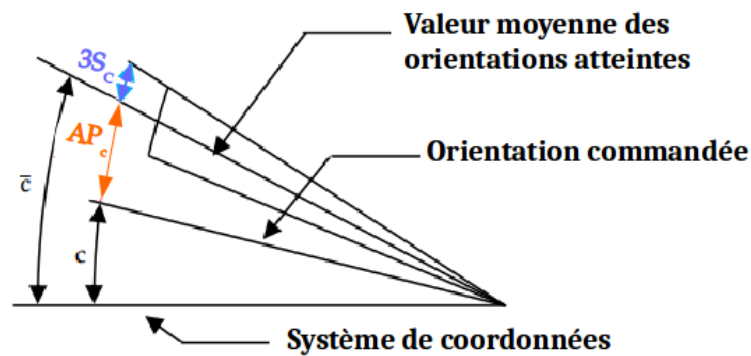


FIGURE 4.3: La répétabilité d'orientation (ISO 9283)

4.2.1 Modélisation de la précision du bras manipulateur

Le laboratoire GREAH a mené un ensemble de recherche sur la précision des manipulateurs séries et parallèles lorsque la base de ces robots est fixe [131][132][133]. Dans notre

cas, le bras robotique est fixé sur la base mobile et l'étude de la précision est faite lorsque le bras est proche de la position de dépose. L'étude nécessite d'utiliser un dispositif de mesure comportant deux trièdres, comme illustré dans la figure 4.4. Un trièdre dit fixe est tenu par les mors de l'étau de la machine-outil et est équipé par trois micromètres numériques Mitutoyo d'une résolution de $1 \mu\text{m}$. Un autre trièdre dit mobile, est fixé au poignet du bras robotique.

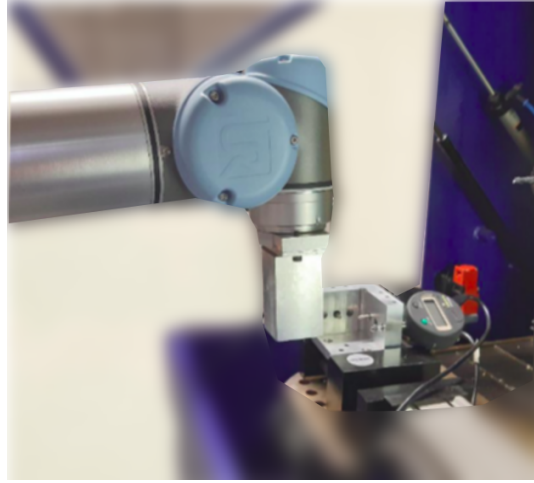


FIGURE 4.4: Le dispositif de mesure pour les tests de précision du bras robotique

La figure 4.5 nous montre les poses relatives des deux trièdres placés en position nominale après un ajustement via le teachpendant. Soit $(O_N, \vec{i}_N, \vec{j}_N, \vec{k}_N)$ le repère lié au trièdre mobile dont les axes sont parallèles aux directions principales du trièdre fixe défini par le repère $(O_M, \vec{i}_M, \vec{j}_M, \vec{k}_M)$. La position et l'orientation du trièdre mobile changent à chaque essai.

On considère le plan horizontal dit d'accostage où la partie basse de la pièce (ou du trièdre mobile) vient en contact potentiel avec la partie supérieure des mors de l'étau et le point G qui est le centre du trièdre mobile dans le plan horizontal.

Les trois micromètres (Sx_1, Sx_2, Sy) nous permettent d'obtenir les variations de position du point G dans le plan xy correspondant à la partie supérieure de l'étau et la variation d'orientation en lacet du trièdre mobile. Soient \vec{dG} et $\vec{d\Omega}_N$ respectivement les écarts de variation de position de G et d'orientation du trièdre mobile dans le plan horizontal par rapport à la position nominale prédéfinie. On note $\vec{dG} = (dGx, dGy, dGz)$ et $\vec{d\Omega}_N = [d\Phi, d\Theta, d\Psi]^t$.

Les points A, B et C sont les points de contact des micromètres avec le trièdre mobile dans un plan horizontal. Du fait que les variations de position \overrightarrow{dG} et d'orientation $\overrightarrow{d\Omega_N}$ restent faibles, on peut considérer au premier ordre un modèle différentiel linéaire issu du torseur des petits déplacements. Ces points subissent alors des variations de position entre la position courante et la position nominale données par :

$$\begin{aligned}\overrightarrow{dSx_1} &= \overrightarrow{dG} + \overrightarrow{Sx_1G} \wedge \overrightarrow{d\Omega_N} \\ \overrightarrow{dSx_2} &= \overrightarrow{dG} + \overrightarrow{Sx_2G} \wedge \overrightarrow{d\Omega_N} \\ \overrightarrow{dSy} &= \overrightarrow{dG} + \overrightarrow{SyG} \wedge \overrightarrow{d\Omega_N}\end{aligned}\tag{4.5}$$

Les variations algébriques des déplacements mesurées par les micromètres sont :

$$\begin{aligned}dSx_1 &= \overrightarrow{dSx_1} \cdot \vec{i} \\ dSx_2 &= \overrightarrow{dSx_2} \cdot \vec{i} \\ dSy &= \overrightarrow{dSy} \cdot \vec{j}\end{aligned}\tag{4.6}$$

Soient a et b les cotes définissant la géométrie du trièdre fixe. Nous remplaçons les équations 4.5 dans 4.6 et nous obtenons après développement :

$$\begin{aligned}dSx_1 &= dGx + a.d\Psi \\ dSx_2 &= dGx - a.d\Psi \\ dSy &= dGy + a.d\Psi\end{aligned}\tag{4.7}$$

D'après 4.7 les variations de position s'écrivent alors :

$$\begin{cases} dGx = \frac{dSx_1 + dSx_2}{2} \\ dGy = dSy + \frac{dSx_2 - dSx_1}{2} \\ d\Psi = \frac{dSx_1 - dSx_2}{2a} \end{cases}\tag{4.8}$$

Nous remarquons que les variations d'orientation en tangage et roulis n'ont pas d'incidence au premier ordre sur le positionnement dans le plan horizontal correspondant aux mors de l'étau.

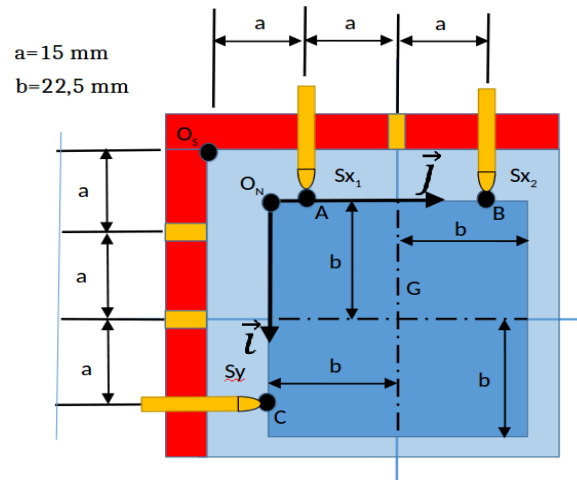


FIGURE 4.5: Schéma de principe des trièdres en position nominale

Nous estimons la répétabilité du bras lorsque la base mobile est arrêtée à l'issue de l'asservissement visuel sur l'ARtag fixé sur la machine-outil sur une campagne de 30 essais. La répétabilité de position est alors estimée à $\pm 7.8\text{mm}$ et la répétabilité d'orientation à $\pm 2.1^\circ$.

La matrice de covariance C_A de la variable aléatoire $(dX, dY, d\Psi)$ au point P dans le repère micromètre est estimée à [unités : $\text{mm}^2; \text{mm.rd}; \text{rd}^2$] :

$$C_A = \begin{bmatrix} 7.12 & -4.37 & -0.0179 \\ -4.37 & 8.95 & 0.0203 \\ -0.0179 & 0.0203 & 1.48 \times 10^{-4} \end{bmatrix} \quad (4.9)$$

La répétabilité du bras robotique UR5 annoncée par le constructeur est $\pm 0.1\text{ mm}$. La répétabilité de l'UR5 estimée par nos soins lorsqu'il est fixé sur la base mobile est 7.84 mm . Cette différence de performance s'explique logiquement par la compliance de la structure mécanique de la base mobile supportant le bras, qui entraîne une variabilité de la pose de la base lors du mouvement du bras robotique et détériore la répétabilité du robot. Plus le support du bras robotique est compliant, plus le bras perd ses performances en précision.

4.2.2 Modélisation de la précision du robot mobile

La modélisation de la précision de positionnement de la base mobile est réalisée par l'exploitation des données LiDAR et un dispositif métrologique constitué de deux équerres de référence qui sont ajoutées dans la scène comme l'illustre la figure 4.6.

Soit $(O_W, \vec{i}_W, \vec{j}_W, \vec{k}_W)$, $(O_R, \vec{i}_R, \vec{j}_R, \vec{k}_R)$, $(O_1, \vec{i}_1, \vec{j}_1, \vec{k}_1)$, $(O_{FL}, \vec{i}_{FL}, \vec{j}_{FL}, \vec{k}_{FL})$ respectivement les repères liés à l'atelier (Workshop), au Robot mobile, à l'équerre 1 et au LiDAR avant (Front LiDAR). Les positions des équerres et de la machine-outil sont connues dans le repère atelier.

Nous considérons deux poses différentes issues de la navigation du robot mobile, une pose de référence issue du premier essai, notée en exposant (1) et une pose issue d'un essai ultérieur notée en exposant (2). Entre ces deux essais, la position et l'orientation relatives des repères robot mobile et équerre 1 changent. Notons : $(O_R^{(1)}, \vec{i}_R^{(1)}, \vec{j}_R^{(1)}, \vec{k}_R^{(1)})$, $(O_R^{(2)}, \vec{i}_R^{(2)}, \vec{j}_R^{(2)}, \vec{k}_R^{(2)})$ le repère Robot mobile lors des essais (1) et (2), $(O_{FL}^{(1)}, \vec{i}_{FL}^{(1)}, \vec{j}_{FL}^{(1)}, \vec{k}_{FL}^{(1)})$, $(O_{FL}^{(2)}, \vec{i}_{FL}^{(2)}, \vec{j}_{FL}^{(2)}, \vec{k}_{FL}^{(2)})$ les repères FRONT LiDAR pour les essais (1) et (2), tels que décrits sur la figure 4.6. Pour le premier essai, la position finale du robot par rapport au repère atelier peut se décomposer de la manière suivante :

$$\overrightarrow{O_W O_R^{(1)}} = \overrightarrow{O_W O_1} + \overrightarrow{O_1 O_{FL}^{(1)}} + \overrightarrow{O_{FL}^{(1)} O_R^{(1)}} \quad (4.10)$$

et pour le second essai :

$$\overrightarrow{O_W O_R^{(2)}} = \overrightarrow{O_W O_1} + \overrightarrow{O_1 O_{FL}^{(2)}} + \overrightarrow{O_{FL}^{(2)} O_R^{(2)}} \quad (4.11)$$

D'après 4.10 et 4.11 nous obtenons les variations de position et il reste maintenant à exprimer les vecteurs dans le repère atelier grâce aux matrices de rotation $M(\alpha) = [\cos(\alpha) - \sin(\alpha); \sin(\alpha)\cos(\alpha)]$:

$$\begin{aligned} \overrightarrow{O_R^{(1)} O_R^{(2)}} &= \overrightarrow{O_1 O_{FL}^{(2)}} - \overrightarrow{O_1 O_{FL}^{(1)}} + \overrightarrow{O_{FL}^{(2)} O_R^{(2)}} - \overrightarrow{O_{FL}^{(1)} O_R^{(1)}} \\ &= -\overrightarrow{O_{FL}^{(2)} O_1} + \overrightarrow{O_{FL}^{(1)} O_1} - \overrightarrow{O_R^{(2)} O_{FL}^{(2)}} + \overrightarrow{O_R^{(1)} O_{FL}^{(1)}} \\ &= -[O_1^{(2)}]_{FL^{(2)}} + [O_1^{(1)}]_{FL^{(1)}} - [O_{FL}^{(2)}]_{R^{(2)}} + [O_{FL}^{(1)}]_{R^{(1)}} \\ &= -M(\Omega_R^{(2)}) \times [O_1^{(2)}]_{FL^{(2)}} + M(\Omega_R^{(1)}) \times [O_1^{(1)}]_{FL^{(1)}} \\ &\quad - M(\Omega_R^{(2)}) \times [\overrightarrow{O_R O_{FL}}]_R + M(\Omega_R^{(1)}) \times [\overrightarrow{O_R O_{FL}}]_R \end{aligned} \quad (4.12)$$

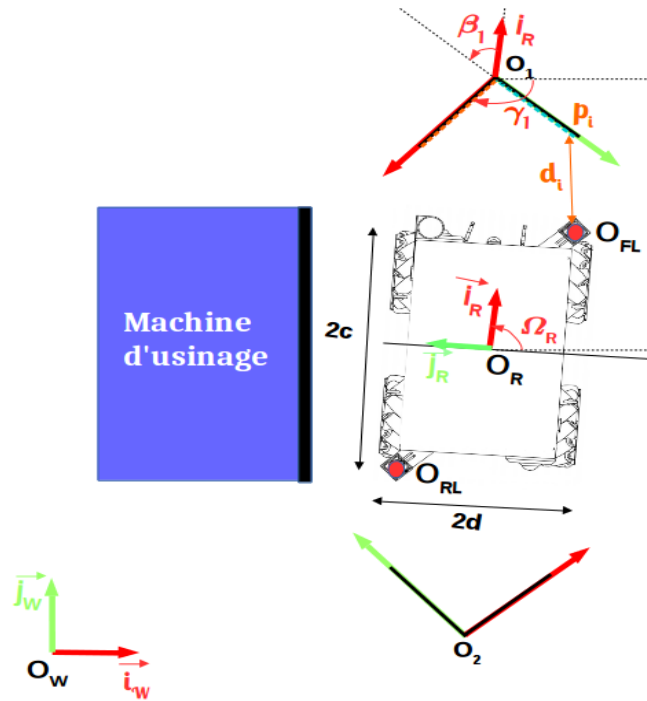


FIGURE 4.6: Schéma des dispositifs pour les tests sur la précision de la base mobile

Avec :

$$\overrightarrow{O_R O_{FL}} = \begin{bmatrix} c \\ -d \end{bmatrix} \quad (4.13)$$

Les relations angulaires autour de l'équerre 1 présentées dans la figure 4.6 :

$$\begin{cases} \Omega_R^{(1)} + \beta_1^{(1)} + \pi/2 - \gamma_1 = 2\pi \\ \Omega_R^{(2)} + \beta_1^{(2)} + \pi/2 - \gamma_1 = 2\pi \end{cases} \quad (4.14)$$

D'où par soustraction :

$$\Omega_R^{(2)} - \Omega_R^{(1)} = \beta_1^{(1)} - \beta_1^{(2)} \quad (4.15)$$

Nous utilisons la norme ISO9283 [130] pour calculer la répétabilité de la base mobile. Pour cela, nous récupérons les signaux des LiDARs avant et arrière pour des distances inférieures à 2 m ainsi qu'illustré sur la figure 4.6.a. Puis grâce à une interface graphique

spécifiquement développée, nous filtrons les signaux pour ne conserver que les points correspondants à l'intérieur des faces des équerres métrologiques, ce qu'illustre la figure 4.6.b. Les points restants sont utilisés pour estimer les équations des droites des faces latérales de l'équerre au sens des moindres carrés et déterminer ainsi précisément la position de O_1 point d'intersection de ces droites (voir figure 4.6.c).

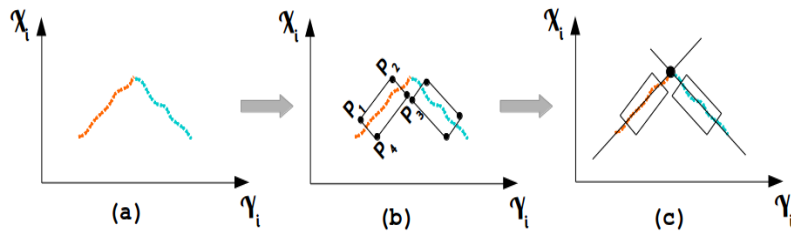


FIGURE 4.7: Les signaux du Front LiDAR (a) coupés à 2 mètres (b) fenêtres polygonales de filtrage par interface graphique (c) point d'intersection des droites des moindres carrés

Après les calculs des points d'intersection des faces latérales de l'équerre, les équations 4.12 et 4.15 permettent d'estimer la pose du robot mobile dans le repère atelier.

La répétabilité de pose du robot mobile est estimée à $\pm 4.15\text{cm}$ en position et à $\pm 3.49^\circ$ en orientation. Ceci laisse présager que le manipulateur mobile ne sera pas en mesure d'effectuer la tâche de dépôt de la pièce dans l'étoupe avec la fiabilité exigée d'un environnement de production industrielle.

Nous avons aussi estimé la matrice de covariance C_R de la variable aléatoire traduisant la variabilité de la pose du robot mobile ($dOx_R, dOy_R, d\Omega_R$) [unités : $\text{mm}^2; \text{mm.rd}; \text{rd}^2$]:

$$C_R = \begin{bmatrix} 123 & -59.7 & 8.2 \times 10^{-3} \\ -59.7 & 213 & 0.248 \\ -8.2 \times 10^{-3} & 0.248 & 3.98 \times 10^{-4} \end{bmatrix} \quad (4.16)$$

4.2.3 Modélisation de la précision du manipulateur mobile

Après avoir estimé séparément la précision du manipulateur et de la base mobile, dans cette partie nous modélisons la précision de l'ensemble bras robotique et base mobile. Dans

les expérimentations que nous avons réalisées, le manipulateur mobile réitère 30 fois le scénario suivant : départ d'un point situé à 3 mètres de la machine-outil, navigation référencée signaux LiDAR jusqu'à un emplacement proche de la machine-outil, ajustement référencé ARtag sur la position finale souhaitée proche de la machine-outil et déploiement du bras robotique à l'intérieur de la machine-outil afin que le trièdre mobile entre en contact avec les micromètres placés sur le trièdre fixe ; les valeurs des micromètres sont alors lues et stockées dans la base de données.

Soit $d\vec{\tau} = (dPx, dPy, d\Psi)$ les coordonnées du vecteur aléatoire caractérisant les variations de la pose du trièdre mobile au point P comme montré dans la figure 4.8.

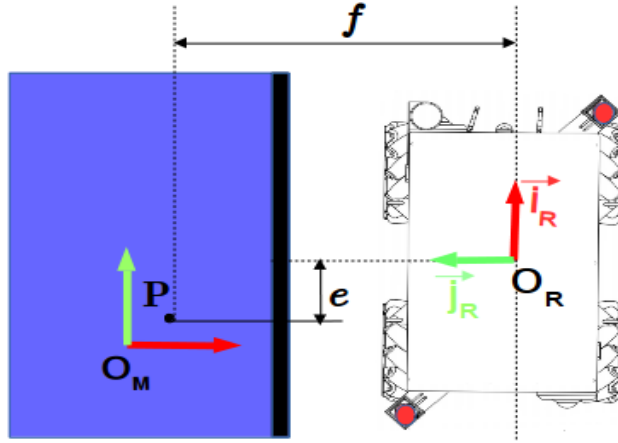


FIGURE 4.8: Illustration du point P et des cotations nécessaires.

Notons E_1 l'événement «le bras est stationnaire et le robot mobile se déplace» et E_2 l'événement «le bras se déplace et le robot mobile est stationnaire».

Nous pouvons faire l'hypothèse que les deux vecteurs aléatoires $d\vec{\tau}|E_1$ et $d\vec{\tau}|E_2$ sont indépendants dans la mesure où les commandes de la base mobile et du bras sont effectuées par deux systèmes totalement indépendants du point de vue des capteurs utilisés pour l'asservissement et du déphasage temporel des opérations. Nous avons alors :

$$d\vec{\tau} = d\vec{\tau}|E_1 + d\vec{\tau}|E_2 \quad (4.17)$$

Nous allons d'abord nous attacher à caractériser $d\vec{\tau}|E_1$ la variabilité de la pose du

trièdre mobile expliquée par la variabilité de la pose du robot mobile. Ceci correspond à la situation où le bras est étendu et figé et les écarts de pose du robot mobile se transmettent à l'extrémité du bras robotique. Au premier ordre, les variations de P sont modélisées par le modèle des petits déplacements et correspondent à l'équation :

$$\begin{aligned}
 \overrightarrow{dP} &= \overrightarrow{dO_R} + \overrightarrow{PO_R} \wedge \overrightarrow{d\Omega_R} \\
 &= \begin{bmatrix} dOx_R \\ dOy_R \\ dOz_R \end{bmatrix} + \begin{bmatrix} f \\ e \\ 0 \end{bmatrix} \wedge \begin{bmatrix} 0 \\ 0 \\ d\Omega_R \end{bmatrix} \\
 &= K \times \begin{bmatrix} dOx_R \\ dOy_R \\ d\Omega_R \end{bmatrix}
 \end{aligned} \tag{4.18}$$

avec $dOz_R = 0$, $e = 55$ mm, $f = 795$ mm et K est la matrice :

$$K = \begin{bmatrix} 1 & 0 & e \\ 0 & 1 & -f \\ 0 & 0 & 0 \end{bmatrix} \tag{4.19}$$

Quant à la variation d'orientation $d\Psi$, elle est égale à $d\Omega_R$. D'où la relation :

$$d\vec{\tau} = K_1 \times \begin{bmatrix} dOx_R \\ dOy_R \\ d\Omega_R \end{bmatrix} \tag{4.20}$$

où

$$K_1 = \begin{bmatrix} 1 & 0 & e \\ 0 & 1 & -f \\ 0 & 0 & 1 \end{bmatrix} \tag{4.21}$$

Pour rappel, l'image d'un vecteur gaussien Y de matrice de covariance C_Y par une application linéaire $X=AY$ donne un vecteur gaussien X de matrice de covariance $C_X = A \times C_Y \times A^t$.

Comme $(dOx_R, dOy_R, d\Omega_R)$ est un vecteur gaussien de matrice de covariance C_R , nous aurons la matrice de covariance du vecteur aléatoire $d\vec{\tau}|E_1$ qui vaut :

$$C_A|E_1 = K_1 \times C_R \times K_1^t \quad (4.22)$$

Nous allons maintenant caractériser la variabilité de la pose $d\vec{\tau}|E_2$ du trièdre mobile expliquée par la variabilité de la pose du bras. La matrice de covariance $C_A|E_2$ associée est tout simplement celle déterminée expérimentalement dans le paragraphe 4.2.1 Modélisation de la précision du bras manipulateur, soit :

$$C_A|E_2 = C_A \quad (4.23)$$

La matrice de covariance globale qui traduit la variabilité de la pose du trièdre mobile au point P dans le repère M s'obtient alors en sommant les deux matrices de covariance précédentes puisque les deux vecteurs aléatoires sont indépendants :

$$C_P = C_A|E_1 \cap E_2 = C_A|E_1 + C_A|E_2 \quad (4.24)$$

On obtient donc la matrice de covariance suivante :

$$C_P = \begin{bmatrix} 132 & -101 & -0.0480 \\ -101 & 867 & 0.585 \\ -0.048 & 0.585 & 5.46 \times 10^{-4} \end{bmatrix} \quad (4.25)$$

Cela donne les écarts-types suivants : $\sigma_x = 11.5$ mm ; $\sigma_y = 29.5$ mm ; $\sigma_\psi = 1.33^\circ$.

L'importance de l'écart-type dans la direction Y s'explique facilement par l'effet bras de levier qui transforme l'incertitude angulaire de l'ensemble de la plate-forme par la longueur du bras, qui atteint presque 800 mm lorsqu'il est déplié pour atteindre l'étau.

L'analyse de ces résultats montre qu'il sera impossible d'exécuter cette tâche avec une fiabilité compatible avec un contexte industrie 4.0 (probabilité d'échec inférieure à 1 pour 1000) en mettant en oeuvre uniquement cette stratégie de boucle ouverte pour la commande finale du bras. En effet en comparant la largeur du jeu existant dans la direction y qui est de 30 mm environ à six fois l'écart-type dans cette même direction, on déduit que le taux d'échec sera très fort.

Il est possible de calculer la répétabilité au sens ISO à partir de l'expression de la matrice de covariance ; on trouve ± 9.96 cm et pour l'orientation $\pm 3.99^\circ$

Il est intéressant de noter par ailleurs que nous n'avons pas été en mesure d'estimer la répétabilité du manipulateur mobile par notre dispositif expérimental du fait de cette grande dispersion. En effet, après avoir réalisé 30 essais, l'analyse des données montre que 8 essais sur 30 n'ont pas abouti à un contact entre le trièdre mobile et les trois micromètres placés sur le trièdre fixe, condition sine qua non pour l'acquisition de mesures fiables. Soit l'indication des micromètres est à zéro ce qui indique que le trièdre mobile est bloqué dans son mouvement par le trièdre fixe et donc que la commande le destinait à aller plus loin, soit les indications des micromètres dépassent la course totale des micromètres (14 mm). Dans ces conditions, il n'est pas possible de déterminer une bonne estimation de la répétabilité du manipulateur mobile. On pourrait penser qu'une valeur plancher pourrait être obtenue à partir des 30 valeurs mais c'est sans compter que les variations d'orientation ne pouvant pas être déterminées si l'un des micromètres $Sx1$ ou $Sx2$ n'est plus en contact ou se trouve en butée, les calculs des variations dGx et dGy sont alors erronées. D'ailleurs les valeurs obtenues par calcul sur l'ensemble des valeurs montrent l'ineptie d'une telle approche puisque les résultats ont alors une répétabilité de ± 18 mm en position et $\pm 17^\circ$ en orientation, ce qui est loin des réalités de l'expérience.

4.3 Stratégie optimale pour un assemblage contraint

Nous allons procéder en deux temps : tout d'abord modéliser la réussite d'une tâche d'assemblage dont on connaît les contraintes fonctionnelles de jeu et la variabilité de pose grâce à la matrice de covariance relative aux coordonnées $(dPx, dPy, d\Psi)$, puis ensuite

tester différentes procédures de dépose de la pièce et évaluer leur fiabilité.

Modélisation de la réussite d'une tâche d'assemblage

Il existe plusieurs méthodes pour assembler une pièce dans une autre : en particulier, l'insertion peut se faire sans contact ou avec contact. Pour donner la plus grande probabilité de réussite à une tâche d'assemblage donnée à effectuer sans intégrer de capteurs supplémentaires (caméra, capteurs d'effort, capteurs de contact...), il faut viser à placer la pièce à insérer en cherchant une position de référence $P^{(ref)}$ qui permet d'équilibrer les zones de jeu dont on dispose. Ceci est illustré dans la figure 4.10 où nous devons insérer une pièce de couleur verte de dimensions 120 x 80 mm dans un étai dont la surface libre de couleur blanche est de dimensions 100 x 110 mm.

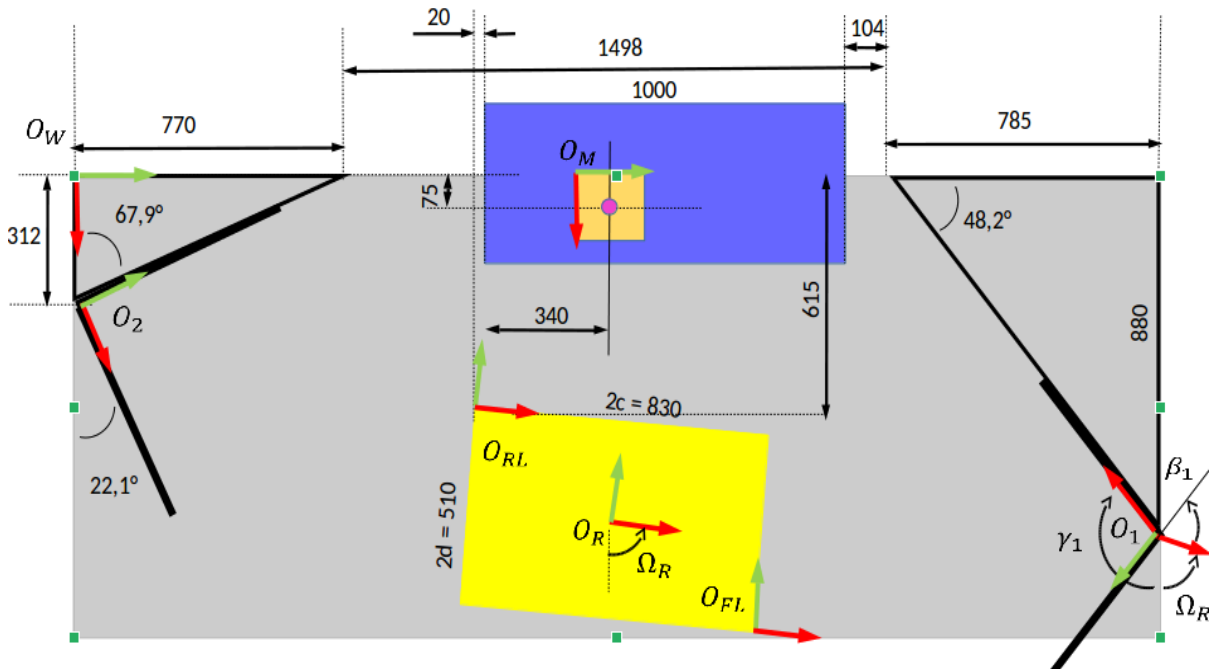


FIGURE 4.9: Vue globale du dispositif métrologique.

La procédure qui semble la plus adaptée est la suivante : la première étape consiste à amener le robot mobile sur sa référence $O_R^{(ref)}$, la seconde étape consiste à piloter le bras à l'aide du teachpendant pour placer la pièce au centre de l'étai et parallèlement aux mors. La position du bras est alors notée comme une position de référence pour le vecteur des

coordonnées articulaires $\Theta^{(ref)}$. Lorsque le mouvement complet du manipulateur mobile est répété, la pose de la pièce emportée par le manipulateur mobile, définie dans un plan horizontal parallèle aux bords supérieurs des mors de l'étau par les coordonnées (Px, Py, Ψ) subit une variabilité que l'on peut définir par rapport à la position nominale idéale d'assemblage par $(dPx, dPy, d\Psi) = (Px, Py, \Psi)^{(fin)} - (Px, Py, \Psi)^{(ref)}$.

$(dPx, dPy, d\Psi)$ est un vecteur aléatoire gaussien dont la matrice de covariance a été estimée précédemment. A partir du processus qualifié de saut, traduisant la différence entre la position de référence et la position suivante, il est donc possible de calculer des zones de confiance pour un risque donné, c'est-à-dire des régions de l'espace de type ellipses où le point P se trouvera avec une probabilité $1 - \alpha$ et donc sera en dehors de cette zone avec une probabilité de α .

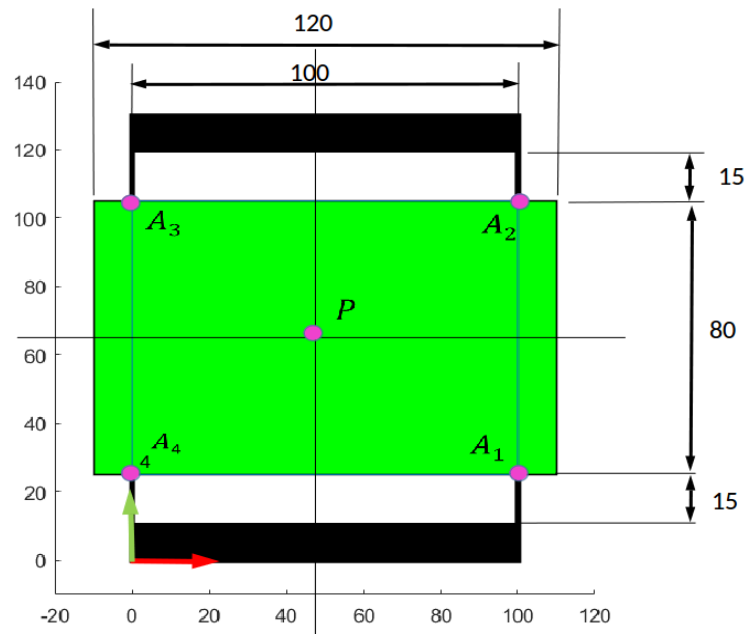


FIGURE 4.10: Position nominale de dépôt de la pièce dans l'étau de la machine.

La tâche d'assemblage est réussie si le rectangle $A_1A_2A_3A_4$ n'entre pas en collision avec les mors. Du fait que le rectangle et la zone libre de l'étau sont des surfaces convexes, il suffit de s'assurer que les déplacements des 4 sommets du rectangle restent dans la zone de jeu admissible. Les surfaces elliptiques de confiance liées au risque α ne doivent donc

pas empiéter sur les mors de l'étau et ceci pour les 4 points considérés.

Les variations de position du point A_i ($i = 1, \dots, 4$) sont caractérisées par une matrice de covariance que l'on établit toujours en se référant au torseur des petits déplacements :

$$\begin{aligned}
 \overrightarrow{dA_i} &= \overrightarrow{dP} + \overrightarrow{A_i P} \wedge \overrightarrow{d\Omega_P} \\
 &= \begin{bmatrix} dPx \\ dPy \\ dPz \end{bmatrix} - \begin{bmatrix} PAx_i \\ PAy_i \\ 0 \end{bmatrix} \wedge \begin{bmatrix} 0 \\ 0 \\ d\Psi \end{bmatrix} \\
 &= H \times \begin{bmatrix} dPx \\ dPy \\ d\Psi \end{bmatrix}
 \end{aligned} \tag{4.26}$$

où

$$H = \begin{bmatrix} 1 & 0 & -PAy_i \\ 0 & 1 & PAx_i \\ 0 & 0 & 0 \end{bmatrix}$$

La matrice de covariance de saut du point A_i est alors :

$$C_s(A_i) = 2 \times H \times C_P \times H^t \tag{4.27}$$

Stratégies d'assemblage et fiabilité

Nous allons maintenant lister quelques stratégies d'assemblage possibles dont certaines ont été mises en oeuvre pour insérer la pièce dans l'étau.

Stratégie n°1 :

1. Le manipulateur mobile navigue en utilisant l'algorithme de navigation Stack sous ROS par asservissement sur les signaux LiDAR jusqu'à une position proche de la machine.
2. Le robot mobile ajuste sa position par un asservissement visuel utilisant la caméra RGB-D du sommet du mât comme capteur et un ARtag placé sur le capot de la machine.

3. Le bras se déploie vers la cible P en suivant la trajectoire de référence pour la cible $\Theta^{(ref)}$.

Stratégie n°2 :

1. Le manipulateur mobile navigue en utilisant l'algorithme de navigation Stack sous ROS par asservissement sur les signaux LiDAR jusqu'à une position finale proche de la machine.
2. Le bras se déploie vers la cible P en suivant la trajectoire de référence pour la cible $\Theta^{(ref)}$.

Stratégie n°3 :

1. Le manipulateur mobile navigue en se basant sur la navigation Stack sous ROS par asservissement sur les signaux LiDAR jusqu'à une position proche de la machine.
2. Une estimation de la position réelle du robot mobile par traitement des signaux LiDAR grâce à une équerre placée dans l'environnement permet d'obtenir l'écart de la pose du robot par rapport à la pose de référence $\overrightarrow{\Delta O_R} = O_R^{(fin)} - O_R^{(ref)}$.
3. On calcule l'écart théorique entre la position de référence de la cible et la position que le bras atteindra du fait que la pose de la base mobile est différente de la pose de référence $\overrightarrow{\Delta P} = \overrightarrow{\Delta O_R} + \overrightarrow{PO_R} \wedge \overrightarrow{\Delta \Omega_R}$
4. On calcule la correction à apporter aux coordonnées articulaires du bras pour compenser l'écart dû à la base mobile en utilisant la matrice jacobienne inverse du bras $\overrightarrow{\Delta \Theta} = J^{-1} \times -\overrightarrow{\Delta P}$
5. Le bras se déploie vers la cible P en suivant la trajectoire de référence pour la cible incrémentée de la variation nécessaire pour compenser l'erreur de la base mobile. $\Theta^{(ref)} + \overrightarrow{\Delta \Theta}$.

Stratégie n°4 :

1. Le manipulateur mobile navigue en utilisant l'algorithme de navigation Stack sous ROS par asservissement sur les signaux LiDAR jusqu'à une position finale proche de la machine.

2. Le bras se déploie vers la cible P en suivant la trajectoire de référence pour la cible $\Theta^{(ref)}$ et se place en position d'attente d'insertion au-dessus de l'étau. Un asservissement visuel est alors mis en oeuvre en utilisant une caméra et trois options peuvent être proposées :

- la caméra utilisée est montée sur l'un des mâts du robot mobile.
- la caméra est embarquée sur le poignet (eye in hand).
- la caméra est installée à l'intérieur de la machine.

Le choix entre ces différentes options dépend de nombreux paramètres : présence de lubrifiant, éclairage, communication caméra à la base mobile, etc

À ce jour, nous avons mis en oeuvre la stratégie n°1 et la stratégie n°3 et voici les principaux résultats obtenus :

La stratégie n°1 ne peut pas fonctionner avec une fiabilité suffisante dans un contexte industrie 4.0. En effet, si on trace sur la figure 4.11 l'ellipse de saut au risque de 0.10, elle possède une intersection importante avec le mors de l'étau et donc l'assemblage sera un échec (partie hachurée de l'ellipse).

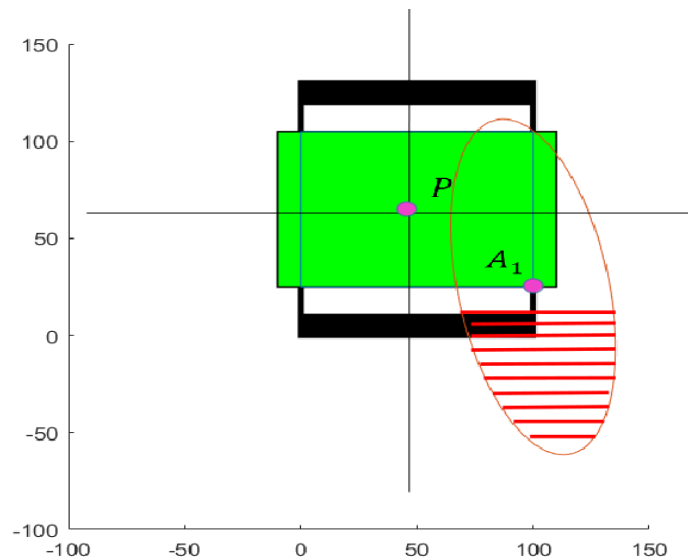


FIGURE 4.11: Ellipse de saut à 90 %.

La stratégie n°3 fonctionne avec un taux d'échec de 1 % et ne sera pas acceptable avec un taux d'échec de 1 ‰. La figure 4.12 nous montre en effet les ellipses de

saut pour les deux niveaux de risque et les 4 sommets du rectangle et il apparaît clairement que les ellipses de risque 1‰ ont une intersection avec la surface des mors.

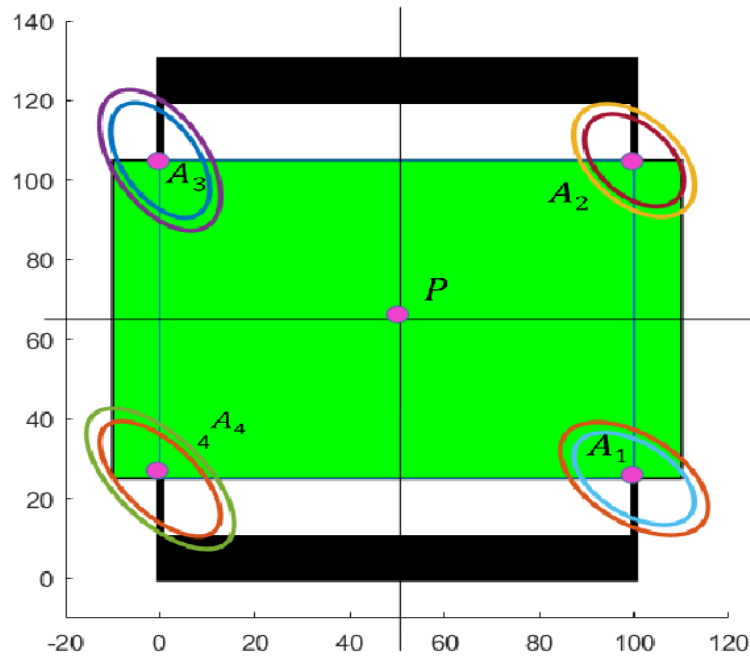


FIGURE 4.12: Ellipses de saut au risque de 1‰ et 1‰.

Dans notre prochain travail, nous allons tester les stratégies n°2 et n°4. Il est probable que les performances des stratégies n°2 et n°3 seront proches, la stratégie n°4 devrait donner les meilleurs résultats si tant est que le process caméra soit bien étudié.

Conclusion générale et perspectives

Les travaux de cette thèse sont menés dans le cadre du projet CoRoT, visant à développer des systèmes robotiques de manipulation mobile collaborative et innovante dans le contexte de l'Industrie 4.0. La manipulation mobile dans les ateliers flexibles ou les entrepôts logistiques implique de maîtriser les fondamentaux de la robotique mobile et de la robotique de manipulation, à savoir, du côté de la robotique mobile, la localisation du robot dans l'environnement, la planification et la navigation en évitant les obstacles statiques et dynamiques ; du côté de la robotique de manipulation, la perception de l'environnement à l'aide de capteurs de nature différente qui doit permettre l'identification de l'objet, puis sa localisation, ensuite la définition de la méthode de préhension et la planification de trajectoire, enfin le suivi de trajectoire en détectant et évitant les obstacles qui l'entourent. Ces divers concepts ont été étudiés, analysés et mis en oeuvre dans le cadre de ces travaux de recherche.

Dans le premier chapitre du manuscrit, nous avons dressé un état de l'art portant sur les différents robots manipulateurs mobiles, qui paradoxalement restent encore assez peu présents dans les lignes de production comparativement aux AGV et robots mobiles ou bras robotique et cobots manipulateurs placés sur des bases fixes. Une des hypothèses qui expliquerait ce retard réside dans la relative complexité de déploiement dû à la variété des constituants et logiciels à mettre en oeuvre ainsi qu'il ressort de l'état de l'art mené dans le premier chapitre.

Dans le second chapitre, nous avons détaillé un premier cas d'usage visant à alimenter en pièces brutes une machine d'usinage. Cela a nécessité la mise en oeuvre de briques technologiques et logicielles visant à :

- Saisir la pièce du stock en se basant sur la perception de l'objet à partir des marqueurs codes associés à chaque pièce.

- Planifier la trajectoire du bras robotique pour atteindre la pose de la pièce détectée.
- Percevoir, localiser et planifier les trajectoires à suivre par le robot pour assurer son déplacement autonome dans l'entrepôt logistique afin de transporter la pièce à usiner de la station de travail vers la machine. Ce processus de navigation a été réalisé en s'appuyant sur la bibliothèque de navigation Stack.
- Ajuster la position et l'orientation du robot par rapport à la machine à partir de l'asservissement visuel et déposer la pièce par le même processus de planification de trajectoire utilisé.

Ces travaux exigent une mise en oeuvre de différentes techniques et technologies nécessitant une connaissance fine des nombreux savoir-faire. En ce sens, nous pouvons qualifier l'ensemble des réalisations de ce chapitre d'ingénierie des systèmes complexes. C'est notamment lorsque l'ingénieur est confronté au déploiement de tels systèmes que l'on comprend la nécessité d'un travail en équipe pluridisciplinaire et l'importance de mettre en oeuvre des techniques de travail collaboratif. La maîtrise de ces outils de travail collaboratif, tant au niveau des interfaçages entre logiciels et fournisseurs de logiciels, qu'au niveau de l'interopérabilité des composants aussi bien mécaniques que logiciels sont au coeur des gains de productivité de l'industrie 4.0. Le middleware ROS est un parfait exemple de ces efforts pour intégrer en terme de drivers, de commande et de supervision de nombreux composants robotiques. La collaboration open-source d'un grand nombre de développeurs et roboticiens a permis à ce jour de disposer de fonctionnalités beaucoup plus importantes qu'il y a 20 ans et ouvre le champ à des applications innovantes.

Le troisième chapitre s'est focalisé sur la première partie du scénario qui est le pick-and-place. Nous avons proposé d'utiliser des méthodes innovantes basées sur l'intelligence artificielle, plus précisément l'apprentissage profond, pour assurer une saisie adaptative de l'objet par ouverture ou fermeture de la pince employée. Un traitement visuel pour la détection, le calcul de pose et la classification des faces de l'objet a été développé en utilisant deux méthodes toutes les deux compatibles temps réel : la première est une méthode très simple basée sur les marqueurs code et la seconde est une méthode plus complexe à mettre en oeuvre reposant sur de l'apprentissage automatique. La méthode basée sur les marqueurs code a donné de bons résultats mais elle nécessite de coller un marqueur sur chaque objet,

ce qui reste contraignant. Afin de dépasser cette limitation les travaux se sont dirigés sur le développement d'un scénario de saisie basé sur l'apprentissage automatique profond. À travers l'approche d'apprentissage par transfert nous avons entraîné le réseau de neurone précisément YOLO-V3 et YOLO-V4 une fois avec une seule classe qui représente l'objet et un autre entraînement avec les cinq faces de l'objet. La saisie avec préhension adaptative par apprentissage automatique profond a été étudiée et implémentée sur un robot manipulateur mobile, et la qualité des résultats expérimentaux obtenus montre l'efficacité de l'algorithme proposé. La localisation des objets à saisir a été estimée en se basant sur le vecteur de sortie de YOLO et la profondeur de la caméra pour déterminer le cadre 3D de l'objet puis calculer son centroïde qui représente sa pose.

Ce travail a fait l'objet d'une publication dans la conférence internationale IEEE ETFA 2020 «Emerging Technologies and Factory Automation».

Concernant la planification de mouvement du bras robotique pour atteindre les différentes positions, nous avons pris le parti de la réaliser grâce à la plate-forme MoveIt sous ROS qui englobe l'ensemble des bibliothèques pour générer les différentes trajectoires tout en évitant les obstacles.

Les contributions de ce chapitre sont d'une part une adaptation des travaux de recherche menés au plus haut niveau par des spécialistes des réseaux neuronaux en vue de les transférer dans le domaine de la préhension robotique, d'autre part, un cadre méthodologique global pour la saisie d'objet avec l'identification, la localisation puis la saisie adaptative. Nous avons étudié ces concepts et nous les avons mis en oeuvre avec succès dans ce cas d'étude. Si la phase d'apprentissage reste assez longue et fastidieuse, il faut reconnaître que la rapidité d'exécution de l'algorithme permet par la suite un fonctionnement temps réel très satisfaisant avec une vitesse de détection de 37 images par seconde. Par la suite, nous pourrions nous intéresser à explorer plus en profondeur l'influence de la structuration des couches neuronales pour la reconnaissance des pièces. La méthodologie d'estimation de la pose de l'objet a exploité la richesse des signaux issus d'une caméra RGB-D ; d'une part, un filtrage volumique basé sur la partie Depth du signal, d'autre part, un algorithme d'estimation de pose portant sur les composantes RGB du signal. Il y a encore certainement de nombreuses potentialités à creuser ici dans l'exploitation simultanée de des deux signaux.

Le quatrième chapitre a l'ambition de proposer un paradigme pour la modélisation et la qualification de la précision du manipulateur mobile en vue de le commander de manière optimale pour déposer une pièce dans une machine-outil. Cette opération de dépose est souvent très délicate car les contraintes fonctionnelles de précision, quantifiées par rapport au jeu fonctionnel à respecter entre la pièce et les mors de l'étau, sont souvent importantes. La littérature scientifique à ce jour ne présentait pas d'études dans le contexte d'une modélisation des performances des manipulateurs mobiles contrairement à ce que l'on pouvait trouver sur la précision des bras manipulateurs lorsque la base est fixe.

Concernant la modélisation de la précision de positionnement d'une base mobile, nous avons proposé une innovation en insérant dans l'espace de travail, et plus particulièrement dans le plan des LiDARs, des obstacles ou amers, qui sont en fait des repères métrologiques. Dans un premier temps, la base mobile vient se positionner proche de la machine-outil grâce à un asservissement visuel. Nous mesurons les performances en précision de position (X, Y, Ω) de cet asservissement visuel grâce aux signaux des LiDARs embarqués qui viennent se réfléchir sur les amers métrologiques et déterminons une matrice de covariance de répétabilité de pose en (X, Y, Ω) .

Puis nous nous sommes intéressés à la précision de positionnement du système global en effectuant de nombreuses mesures utilisant deux trièdres : un cube fixé à l'extrémité du bras et un trièdre fixé sur l'étau muni de trois comparateurs micrométriques dans le plan xy . L'analyse des résultats de ces signaux permet d'obtenir les matrices de covariance caractérisant les performances de répétabilité du manipulateur mobile dans deux cas de figure suivants : le premier lorsque la base reste immobile entre deux déplacements du bras, le second lorsque la base est redéplacée entre deux déplacements du bras.

La répétabilité finale de positionnement de l'extrémité du bras manipulateur dépend globalement de deux facteurs : la répétabilité de pose du robot mobile et la répétabilité de pose du bras lui-même. Les deux causes étant indépendantes, nous pouvons sommer les matrices de covariance pour obtenir la répétabilité résultante du système manipulateur mobile. Le calcul théorique nous indique alors que les performances de répétabilité résultant de la procédure d'asservissement visuel et de commande en boucle ouverte du bras ne sont pas suffisantes pour garantir l'insertion de la pièce dans la machine-outil. Ceci est bien corroboré

par le fait que nous n'arrivons pas à mesurer expérimentalement la variabilité en raison des plages de mesure limitées à 14 mm des micromètres utilisés. Nous proposons alors plusieurs méthodes pour améliorer le processus dont deux nous semblent particulièrement pertinentes pour notre cas d'étude : 1. Utiliser les signaux des LiDARs pour obtenir les écarts de la base mobile par rapport à la position nominale et les corriger par une adaptation de la position du bras (via la jacobienne inverse). 2. Ajouter une caméra (eye-in-hand) sur le poignet du bras robotisé. Les travaux d'intégration de ces commandes sont en cours de réalisation et devraient nous permettre de divulguer de nouveaux résultats prochainement.

Nous avons par la suite détaillé le procédé permettant de commander un tel système pour avoir une probabilité de succès le plus important pour la tâche d'insertion à réaliser. Cette méthodologie consiste à régler la position nominale de dépose en laissant un jeu égal d'un côté et de l'autre de la pièce par rapport à l'étai. Ensuite, connaissant la matrice de covariance finale, nous sommes en mesure d'estimer la fiabilité de réussite de la tâche d'assemblage par l'introduction de la variable aléatoire saut, caractérisant la différence entre la position à venir et la position nominale. Les variations de position de l'objet à déposer doivent être analysées sur des points particuliers du solide. Dans notre cas, la pièce était un pavé droit, donc un objet convexe, ce qui simplifie l'analyse. L'étude de ces contraintes dans le cas d'objets non convexes est un sujet intéressant pour la suite des travaux.

D'un point de vue conceptuel, nous souhaitons proposer le néologisme d'amerisation de l'environnement pour signifier que nous introduisons de manière opportune, des amers métrologiques. L'avantage de ces amers est de permettre de réduire de manière très importante le volume de l'information traitée puisque nous savons où chercher l'information pertinente dans cette masse de donnée et la réduire par la suite à trois indicateurs ΔX , ΔY , $\Delta \Omega$. Nous passons ainsi de la notion de Big Data à celle de Smart Data. Nous pouvons affirmer aussi que la méthode proposée est très pragmatique et consiste in fine à rendre l'environnement intelligent par rapport au capteur le plus précis dont nous disposons sur la base mobile en mouvement, à savoir les LiDARs. La question qui vient alors naturellement à l'esprit est la suivante : faudrait-il alors s'affranchir de l'asservissement visuel sur les ARtags placés sur la machine-outil Pour reprendre notre néologisme, cet ARtag peut être caractérisé comme étant un autre type d'amerisation de l'environnement qui code deux informations, l'une géo-

graphique de position, l'autre sémantique d'identification de la machine-outil permettant de lever les ambiguïtés du perceptual aliases et renforçant la robustesse systémique codée dans l'ARtag. L'ARtag peut donc être utilisé dans la procédure à bon escient.

Le néologisme proposé d'amerisation de l'environnement regroupe différentes catégories d'amers, parmi lesquels figurent les amers géographiques, les amers sémantiques et les amers métrologiques. Il serait intéressant par la suite de s'intéresser aux amers par leur fonctionnalité et le type de capteur sensoriel utilisé pour décoder l'information. L'utilisation d'amers sonores est par exemple fréquente en ville pour pallier aux difficultés de vision des personnes malvoyantes.

Au final, nos travaux ouvrent des perspectives fécondes pour développer des stratégies de compensation d'incertitudes des manipulateurs mobiles pour les tâches d'assemblages afin d'améliorer grandement leur précision. Ces travaux feront l'objet d'une publication en revue qui est en cours d'études et développement.

Plus généralement les solutions étudiées et développées dans le cadre du projet CoRoT ainsi que dans d'autres projets de recherche du même concept dont quelques uns sont cités dans le manuscrit, ont pour vocation de répondre aux enjeux de l'industrie 4.0 en particulier dans l'intralogistique et la logistique au sens large.

Une première perspective des travaux sera reliée à l'étape de la saisie de la pièce à usiner et consistera à réaliser d'autres apprentissages profonds avec différents algorithmes tels que le SSD, R-CNN et ses versions évoluées, etc, ainsi que de réaliser la tâche de pick-and-place adaptatif sur de nouvelles pièces avec les différents algorithmes employés afin de comparer leurs performances en terme de fiabilité et flexibilité.

Une deuxième perspective suite aux travaux du dernier chapitre va consister à élargir l'étude sur la modélisation de la précision du manipulateur mobile suivant les différentes stratégies proposées dans le chapitre IV (introduction d'un asservissement visuel ou sur les signaux LiDARs pour ajuster la pose de la base mobile dans l'environnement prédéfini ou par l'insertion d'un ou plusieurs autres capteurs pour contrôler le mouvement du bras robotique). La mise en oeuvre de ces stratégies est en cours sur deux cas d'usage (PIL et entreprise CERI).

Bibliographie

- [1] CoRoT, <http://www.corot-project.org/info/>,.
- [2] Oliver Brock and Roderic Grupen. Nsf/nasa workshop on autonomous mobile manipulation. In *Workshop on manipulation in human environments at robotics : science and systems*. University of Massachusetts Amherst, Houston, 2005.
- [3] Yuandong Yang Brendan Burns Thomas Buckley Anna Grishkan Volodymyr Zhylkovskyy Oliver Brock Erik Learned-Miller Thomas Buckley Dov Katz, Emily Horrell. The UMass mobile manipulator UMan : An experimental platform for autonomous mobile manipulation. In *Workshop on manipulation in human environments at robotics : science and systems*. Citeseer, 2006.
- [4] François Leborne. *Contributions à la commande de bras manipulateurs de robot sous-marin pour la manipulation à grande profondeur d'échantillons biologiques déformables*. PhD thesis, Université de Montpellier, 2018.
- [5] Anibal Ollero and Bruno Siciliano. *Aerial Robotic Manipulation*. Springer, 2019.
- [6] K. Kondak, F. Huber, M. Schwarzbach, M. Laiacker, D. Sommer, M. Bejar, and A. Ollero. Aerial manipulation robot composed of an autonomous helicopter and a 7 degrees of freedom industrial manipulator. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2107–2112, 2014.
- [7] A. Suarez, A. E. Jimenez-Cano, V. M. Vega, G. Heredia, A. Rodriguez-Castaño, and A. Ollero. Lightweight and human-size dual arm aerial manipulator. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1778–1784, 2017.

- [8] Todd W Danko, Kenneth P Chaney, and Paul Y Oh. A parallel manipulator for mobile manipulating . In *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pages 1–6. IEEE, 2015.
- [9] Bruce A Aikenhead, Robert G Daniell, and Frederick M Davis. Canadarm and the space shuttle. *Journal of Vacuum Science & Technology A : Vacuum, Surfaces, and Films*, 1(2) :126–132, 1983.
- [10] Joy A Crisp, Mark Adler, Jacob R Matijevic, Steven W Squyres, Raymond E Arvidson, and David M Kass. Mars exploration rover mission. *Journal of Geophysical Research : Planets*, 108(E12), 2003.
- [11] JN Maki, JF Bell, Kenneth E Herkenhoff, SW Squyres, A Kiely, M Klimesh, M Schwochert, T Litwin, R Willson, A Johnson, et al. Mars exploration rover engineering cameras. *Journal of Geophysical Research : Planets*, 108(E12), 2003.
- [12] Ambarish Goswami and Prahlad Vadakkepat. *Humanoid robotics : A reference*. Springer, 2019.
- [13] Tobias Klamt and Sven Behnke. Anytime hybrid driving-stepping locomotion planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4444–4451. IEEE, 2017.
- [14] Tobias Klamt and Sven Behnke. Towards learning abstract representations for locomotion planning in high-dimensional state spaces. *2019 International Conference on Robotics and Automation (ICRA)*, pages 922–928, 2019.
- [15] Tobias Klamt, Max Schwarz, Christian Lenz, Lorenzo Baccelliere, Domenico Buongiorno, Torben Cichon, Antonio DiGuardo, David Droeschel, Massimiliano Gabardi, Malgorzata Kamedula, et al. Remote mobile manipulation with the centauro robot : Full-body telepresence and autonomous operator assistance. *Journal of Field Robotics*, 37(5) :889–919, 2020.
- [16] Boston Dynamics. Handle robot reimaged for logistics. Published : 2019.
- [17] Joachim Schuler. *Integration von Förder- und Handhabungseinrichtungen*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1987.

- [18] Gilles Foulon. *Generation de mouvements coordonnés pour un ensemble constitué d'une plateforme mobile à roues et d'un bras manipulateur*. PhD thesis, Toulouse, INSA, 1998.
- [19] S. Iida and S. Yuta. Vehicle command system and trajectory control for autonomous mobile robots. In *Proceedings IROS '91 :IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, pages 212–217 vol.1, 1991.
- [20] H Hibi. Development of mobile robot system adaptive to sharp fluctuation of production volume. In *Proceedings of the 2001 IEEE International Symposium on Assembly and Task Planning (ISATP2001). Assembly and Disassembly in the Twenty-first Century.(Cat. No. 01TH8560)*, pages 61–66. IEEE, 2001.
- [21] S. Angerer, C. Strassmair, M. Staehr, M. Roettenbacher, and N. M. Robertson. Give me a hand — the potential of mobile assistive robots in automotive logistics and assembly applications. In *2012 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pages 111–116, 2012.
- [22] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, and A. Casal. Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96*, volume 2, pages 546–553 vol.2, 1996.
- [23] Mustafa Mashali. *Kinematic Control of Redundant Mobile Manipulators*. PhD thesis, 2015.
- [24] Q. Huang, S. Sugano, and I. Kato. Stability control for a mobile manipulator using a potential method. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, volume 2, pages 839–846 vol.2, 1994.
- [25] Fumiaki Inoue, T Muralami, and K Ihnishi. A motion control of mobile manipulator with external force. *IEEE/ASME transactions on mechatronics*, 6(2) :137–142, 2001.
- [26] Shuai Guo, Tao Song, Fengfeng Xi, and Richard Phillip Mohamed. Tip-over stability analysis for a wheeled mobile manipulator. *Journal of Dynamic Systems, Measurement, and Control*, 139(5) :054501, 2017.
- [27] AG KUKA. Kuka ag-colleague omnirob is on the road, 2011.

- [28] XL-GEN MOBILE MANIPULATOR, <https://robotnik.eu/products/mobile-manipulators/xl-gen/>, Accessed : 2021-01-10.
- [29] RB-1 Robotnik MOBILE MANIPULATOR, <https://www.robotnik.eu/manipulators/rb-one/>, Accessed : 2021-01-10.
- [30] Manman Yang, Erfu Yang, Remi Christophe Zante, Mark Post, and Xuefeng Liu. Collaborative mobile industrial manipulator : A review of system architecture and applications. In *2019 25th International Conference on Automation and Computing (ICAC)*, pages 1–6. IEEE, 2019.
- [31] Zine Elabidine Chebab. *Conception et commande collaborative de manipulateurs mobiles modulaires (C3M3)*. PhD thesis, Université Clermont Auvergne, 2018.
- [32] Kim Wansoo, Marta Lorenzini, Balatti Pietro, Wu Yuqiang, and Arash Ajoudani. Towards ergonomic control of collaborative effort in multi-human mobile-robot teams. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages N–A, 2019.
- [33] Siddhartha S Srinivasa, Dave Ferguson, Casey J Helfrich, Dmitry Berenson, Alvaro Collet, Rosen Diankov, Garratt Gallagher, Geoffrey Hollinger, James Kuffner, and Michael Vande Weghe. Herb : a home exploring robotic butler. *Autonomous Robots*, 28(1) :5, 2010.
- [34] Gary E Marchant, Braden Allenby, Ronald Arkin, and Edward T Barrett. International governance of autonomous military robots. *Colum. Sci. & Tech. L. Rev.*, 12 :272, 2011.
- [35] A. K. Pandey and R. Gelin. A mass-produced sociable humanoid robot : Pepper : The first machine of its kind. *IEEE Robotics Automation Magazine*, 25(3) :40–48, 2018.
- [36] Lorenzo Sciavicco and Bruno Siciliano. *Modelling and control of robot manipulators*. Springer Science & Business Media, 2012.
- [37] R Shah and AB Pandey. Concept for automated sorting robotic arm. *Procedia Manufacturing*, 20 :400–405, 2018.
- [38] Hamid D Taghirad. *Parallel robots : mechanics and control*. CRC press, 2013.
- [39] Ilian Bonev. The true origins of parallel robots. *ParalleMIC*, 2003.

- [40] Tahir Rasheed. *Collaborative Mobile Cable-Driven Parallel Robots*. PhD thesis, École centrale de Nantes, 2019.
- [41] Nicolò Pedemonte, Tahir Rasheed, David Marquez-Gamez, Philip Long, Étienne Hocquard, Francois Babin, Charlotte Fouché, Guy Caverot, Alexis Girin, and Stéphane Caro. Fastkit : A mobile cable-driven parallel robot for logistics. In *Advances in Robotics Research : From Lab to Market*, pages 141–163. Springer, 2020.
- [42] J P Merlet. Structural synthesis and architectures. *Parallel Robots*, pages 19–94, 2006.
- [43] Juan Cortés and Thierry Siméon. Sampling-based motion planning under kinematic loop-closure constraints. In *Algorithmic Foundations of Robotics VI*, pages 75–90. Springer, 2004.
- [44] Baohua Zhang, Yuanxin Xie, Jun Zhou, Kai Wang, and Zhen Zhang. State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots : A review. *Computers and Electronics in Agriculture*, 177 :105694, 2020.
- [45] Mads Hvilshøj, Simon Bøgh, Oluf Skov Nielsen, and Ole Madsen. Multiple part feeding-real-world application for mobile manipulators. *Assembly Automation*, 32(1) :62–71, 2012.
- [46] Rasmus Eckholdt Andersen, Emil Blixt Hansen, David Cerny, Steffen Madsen, Biranavan Pulendralingam, Simon Bøgh, and Dimitrios Chrysostomou. Integration of a skill-based collaborative mobile robot in a smart cyber-physical environment. *Procedia Manufacturing*, 11 :114–123, 2017.
- [47] Mehmet Dogar, Ross A Knepper, Andrew Spielberg, Changhyun Choi, Henrik I Christensen, and Daniela Rus. Multi-scale assembly with robot teams. *The International Journal of Robotics Research*, 34(13) :1645–1659, 2015.
- [48] Nds J Nilsson et al. Study of equipment needs for artificial intelligence research. *SRI, NSF Grant GK*, 36487, 1973.

- [49] Sachin Kansal and Sudipto Mukherjee. Automatic single-view monocular camera calibration-based object manipulation using novel dexterous multi-fingered delta robot. *Neural Computing and Applications*, 31(7) :2661–2678, 2019.
- [50] Michael Firman. Rgb-d datasets : Past, present and future. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 19–31, 2016.
- [51] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. IEEE, 1999.
- [52] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.
- [53] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF : Speeded up robust features. In *European Conference on Computer Vision*, pages 404–417. Springer, 2006.
- [54] P. Willaume, P. Parrend, E. Gancel, and A. Deruyver. The graph matching optimization methodology for thin object recognition in pick and place tasks. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8, 2016.
- [55] A Beyhan and NG Adar. Real-time vision-based grasping randomly placed object by low-cost robotic arm using surf algorithm. In *IOP Conference Series : Materials Science and Engineering*, volume 938, page 012008. IOP Publishing, 2020.
- [56] Roberto Brunelli. *Template matching techniques in computer vision : theory and practice*. John Wiley & Sons, 2009.
- [57] P. Swaroop and Neelam Sharma. An overview of various template matching methodologies in image processing. *International Journal of Computer Applications*, 153 :8–14, 2016.
- [58] Aung Kaung Sat and Thuzar Tint. Object detection and recognition system for pick and place robot. In *International Conference on Big Data Analysis and Deep Learning Applications*, pages 315–323. Springer, 2018.

- [59] Jun Rekimoto. Matrix : A realtime object identification and registration method for augmented reality. In *Proceedings. 3rd Asia Pacific Computer Human Interaction (Cat. No. 98EX110)*, pages 63–68. IEEE, 1998.
- [60] H. Kato and M. Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pages 85–94, 1999.
- [61] M. Fiala. Artag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CV-PR'05)*, volume 2, pages 590–596 vol. 2, 2005.
- [62] Edwin Olson. Apriltag : A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407. IEEE, 2011.
- [63] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6) :2280–2292, 2014.
- [64] Matias Nitsche, Tomas Krajnik, Petr Cizek, Marta Mejail, Tom Duckett, et al. Whycon : an efficient, marker-based localization system. 2015.
- [65] Alain Liégeois. Modélisation et commande des robots manipulateurs. *Techniques de l'Ingénieur, S, 7* :730, 2000.
- [66] Cristina Floriana Pană, Daniela Maria Pătrașcu-Pană, Ionel Cristinel Vladu, Liviu Florin Manta, Florina-Luminița Besnea Petcu, Ștefan Irinel Cismaru, and Andrei Costin Trășculescu. Fuzzy control of the robotic arm for a smart electric wheelchair to assist people with movement disabilities. In *2021 22nd International Carpathian Control Conference (ICCC)*, pages 1–6. IEEE, 2021.
- [67] Yi Ge and Jing Zhang. Robotic arm dynamics modelling and robust control based on model recognition method. *Journal of Intelligent & Fuzzy Systems*, (Preprint) :1–11.
- [68] Ali Ebrahimi, Muller Urias, Niravkumar Patel, Russell H Taylor, Peter L Gehlbach, and Iulian Iordachita. Adaptive control improves sclera force safety in robot-assisted eye surgery : A clinical study. *IEEE Transactions on Biomedical Engineering*, pages 1–1, 2021.

- [69] Nguyen Thanh Binh, Nguyen Anh Tung, Dao Phuong Nam, and Nguyen Hong Quang. An adaptive backstepping trajectory tracking control of a tractor trailer wheeled mobile robot. *International Journal of Control, Automation and Systems*, 17(2) :465–473, 2019.
- [70] Yuanlong Xie, Xiaolong Zhang, Wei Meng, Shane Xie, Liquan Jiang, Jie Meng, and Shuting Wang. Coupled sliding mode control of an omnidirectional mobile robot with variable modes. In *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1792–1797. IEEE, 2020.
- [71] Ahmad M Alshorman, Omar Alshorman, Muhammad Irfan, Adam Glowacz, Fazal Muhammad, and Wahyu Caesarendra. Fuzzy-based fault-tolerant control for omnidirectional mobile robot. *Machines*, 8(3) :55, 2020.
- [72] Jae H Chung, Steven A Velinsky, and Ronald A Hess. Interaction control of a redundant mobile manipulator. *The International Journal of Robotics Research*, 17(12) :1302–1309, 1998.
- [73] Víctor Andaluz, Flavio Roberti, and Ricardo Carelli. Robust control with redundancy resolution and dynamic compensation for mobile manipulators. In *2010 IEEE International Conference on Industrial Technology*, pages 1469–1474. IEEE, 2010.
- [74] Ameer Hamza Khan, Shuai Li, Dechao Chen, and Liefu Liao. Tracking control of redundant mobile manipulator : An RNN based metaheuristic approach. *Neurocomputing*, 400 :272–284, 2020.
- [75] STAMINA, <https://www.basystemes.com/fr/projets-rd/stamina/>, 2021-01.
- [76] ColRobot, <https://colrobot.eu/about-colrobot/>, 2021-01.
- [77] commandia, <http://commandia.unizar.es/>, 2021-01.
- [78] Wilson Sakpere, Michael Adeyeye-Oshin, and Nhlanhla BW Mlitwa. A state-of-the-art survey of indoor positioning and navigation systems and technologies. *South African Computer Journal*, 29(3) :145–197, 2017.
- [79] BK Patle, Anish Pandey, DRK Parhi, A Jagadeesh, et al. A review : On path planning strategies for navigation of mobile robot. *Defence Technology*, 15(4) :582–606, 2019.

- [80] Stefan Kohlbrecher, Oskar Von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable SLAM system with full 3d motion estimation. In *2011 IEEE international symposium on safety, security, and rescue robotics*, pages 155–160. IEEE, 2011.
- [81] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. Orb-slam : A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5) :1147–1163, 2015.
- [82] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera : an open-source library for real-time metric-semantic localization and mapping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696. IEEE, 2020.
- [83] Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2) :416–446, 2019.
- [84] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, and John McDonald. Kintinuous : Spatially extended kinectfusion. 2012.
- [85] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1) :34–46, 2007.
- [86] Stefan Kohlbrecher, Johannes Meyer, Thorsten Graber, Karen Petersen, Uwe Klingauf, and Oskar von Stryk. Hector open source modules for autonomous mapping and navigation with rescue robots. In *Robot Soccer World Cup*, pages 624–631. Springer, 2013.
- [87] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1-2) :99–141, 2001.
- [88] James C Alexander and John H Maddocks. On the kinematics of wheeled mobile robots. *The International Journal of Robotics Research*, 8(5) :15–27, 1989.
- [89] David C Conner and Justin Willis. Flexible navigation : Finite state machine-based integrated navigation and control for ROS enabled robots. In *SoutheastCon 2017*, pages 1–8. IEEE, 2017.

- [90] Oliver Brock and Oussama Khatib. High-speed navigation using the global dynamic window approach. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 1, pages 341–346. IEEE, 1999.
- [91] Dirk Buchholz. *Bin-Picking : New Approaches for a Classical Problem*, volume 44. Springer, 2015.
- [92] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2016.
- [93] Tomas Perez-Lozano. Spatial planning : A configuration space approach. *IEEE Trans. Computers*, 32, 1983.
- [94] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505. IEEE, 1985.
- [95] J.C. Latombe. *Robot Motion Planning : Edition en anglais*. The Springer International Series in Engineering and Computer Science. Springer, 1991.
- [96] Nancy M Amato and Yan Wu. A randomized roadmap method for path and manipulation planning. In *Proceedings of IEEE international conference on robotics and automation*, volume 1, pages 113–120. IEEE, 1996.
- [97] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1) :269–271, 1959.
- [98] Howie M Choset, Seth Hutchinson, Kevin M Lynch, George Kantor, Wolfram Burgard, Lydia E Kavraki, Sebastian Thrun, and Ronald C Arkin. *Principles of robot motion : theory, algorithms, and implementation*. MIT press, 2005.
- [99] Steven M LaValle. Rapidly-exploring random trees : A new tool for path planning. 1998.
- [100] Sachin Chitta, Ioan Sucan, and Steve Cousins. Moveit ![ros topics]. *IEEE Robotics & Automation Magazine*, 19(1) :18–19, 2012.
- [101] date = 2021-09 url = <http://ompl.kavrakilab.org/> Kavraki Lab, title = The Open Motion Planning Library (OMPL).
- [102] title = Stochastic Trajectory Optimization for Motion Planning (STOMP) date = 2021-09 url = https://wiki.ros.org/stomp_motion_planner Kalakrishnan, M.

- [103] title = Search-based Planning Library (SBPL) date = 2021-09 url = <https://wiki.ros.org/sbpl> Likhachev, M.
- [104] Nathan Ratliff, Matt Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. Chomp : Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 489–494. IEEE, 2009.
- [105] Wafae SEBBATA, Jean-François BRETHÉ, et al. An adaptive robotic grasping with a 2-finger gripper based on deep learning network. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 620–627. IEEE, 2020.
- [106] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [107] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN : Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6) :1137–1149, 2016.
- [108] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [109] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD : Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
- [110] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [111] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once : Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

- [112] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9) :1627–1645, 2009.
- [113] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
- [114] Joseph Redmon and Ali Farhadi. Yolo9000 : better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.
- [115] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4 : Optimal speed and accuracy of object detection. *arXiv preprint arXiv :2004.10934*, 2020.
- [116] Christian Szegedy, W. Liu, Y. Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, D. Erhan, V. Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.
- [117] Joseph Redmon and Ali Farhadi. Yolov3 : An incremental improvement. *arXiv preprint arXiv :1804.02767*, 2018.
- [118] Zhi Yan, Li Sun, Tom Duckct, and Nicola Bellotto. Multisensor online transfer learning for 3d lidar-based human detection with a mobile robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7635–7640. IEEE, 2018.
- [119] H. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers. Deep convolutional neural networks for computer-aided detection : Cnn architectures, dataset characteristics and transfer learning. *IEEE Transactions on Medical Imaging*, 35(5) :1285–1298, 2016.
- [120] Mohammadali Honarpardaz, Johan Ölvander, and Mehdi Tarkian. Fast finger design automation for industrial robots. *Robotics and Autonomous Systems*, 113 :120–131, 2019.

- [121] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1–8. IEEE, 2018.
- [122] Cihan Uyanik, Syed F Hussaini, Erdem Erdemir, Erkan Kaplanoglu, and Ali Sekmen. A deep learning approach for final grasping state determination from motion trajectory of a prosthetic hand. *Procedia Computer Science*, 158 :19–26, 2019.
- [123] Radu Bogdan Rusu and Steve Cousins. 3d is here : Point cloud library PCL. In *2011 IEEE international conference on robotics and automation*, pages 1–4. IEEE, 2011.
- [124] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3D object detection in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016.
- [125] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme : a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3) :157–173, 2008.
- [126] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European conference on information retrieval*, pages 345–359. Springer, 2005.
- [127] date = 2021-08 url = <https://www.kitamura-machinery.com/> KITAMURA membres, title = KITAMURA.
- [128] Mazyar Ghadiri Nejad, Seyed Mahdi Shavarani, Hüseyin Güden, and Reza Vatankehah Barenji. Process sequencing for a pick-and-place robot in a real-life flexible robotic cell. *The International Journal of Advanced Manufacturing Technology*, 103(9) :3613–3627, 2019.
- [129] Alaa Khamis, Ahmed Hussein, and Ahmed Elmogy. Multi-robot task allocation : A review of the state-of-the-art. *Cooperative Robots and Sensor Networks 2015*, pages 31–51, 2015.
- [130] Norme ISO9283 AFNOR. Robots manipulateurs industriels, 1998.

- [131] Jean-François Brethé, Eric Vasselin, Dimitri Lefebvre, and Brayima Dakyo. Modeling of repeatability phenomena using the stochastic ellipsoid approach. *Robotica*, 24(4) :477, 2006.
- [132] Rolland-Michel Assoumou Nzue, Jean François Brethé, Eric Vasselin, and Dimitri Lefebvre. Comparative analysis of the repeatability performance of a serial and parallel robot. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 63–68. IEEE, 2010.
- [133] Jean-François BRETHÉ. Intrinsic repeatability : a new index for repeatability characterisation. In *2010 IEEE International Conference on Robotics and Automation*, pages 3849–3854. IEEE, 2010.