

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد حطاب البليدة  
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا  
Faculté de Technologie

قسم الإلكترونيك  
Département d'Électronique



## Mémoire de Projet de Fin d'Études

présenté par

Aouari Ibrahim

et

Benterkia falehedinne

pour l'obtention du diplôme de master en Électronique option système vision et robotique

---

Thème

---

**Ajustement visuel et automatique de la pince d'un  
manipulateur mobile sur la poignée d'une porte en  
exploitant la caméra embarquée : Application au robot  
*RobuTER/ULM.***

---

Proposé par : Hentout Abdelfetah et Naamane Abderrahmane

Année Universitaire 2012-2013



# Remerciement

---

Nous tenons à remercier notre dieu, le plus puissant de nous avoir donné le courage, la volonté et la patience pour réaliser ce travail.

Nous remercions vivement tous ceux qui m'ont assisté dans la préparation, le suivi et la réalisation du présent mémoire et plus particulièrement notre encadreur, Dr Hentout Abdelfetah et toutes l'équipe du robotique de CDTA, ainsi que notre enseignant, Dr Naaman Abderrahman .

Nous remercions également nos enseignants, Président et examinateurs, qui nous ont honoré en acceptant de faire partie du jury de soutenance de notre mémoire.

---

---

# Resume

---

**ملخص:** مشروعنا يتمثل في تجسيد مهمة فتح الباب بواسطة روبوت متحرك RobuTER/ULM الذي يتألف من قاعدة متحركة و ذراع مناور ULM. الذراع يحمل كاميرا على المنفذ. الهدف من هذا العمل هو حل مشكلتين أساسيتين, الأولى تتعلق في حساب موقع الحقيقي ثلاثي الأبعاد لمقبض الباب. الثانية متعلقة في التسوية الأوتوماتكية لقابض الروبوت على مقبض الباب باستغلال الصور المأخوذة من الكاميرا.

**كلمات المفاتيح:** مناور متحرك, كاميرا محمولة, موضع الحقيقي ثلاثي الأبعاد, التسوية الأوتوماتكية.

---

**Résumé :** Notre projet consiste à réaliser la mission d'ouverture de la porte par le robot manipulateur mobile *RobuTER/ULM*, qui se compose d'une plateforme mobile de type *RobuTER* surmontée d'un bras manipulateur ultraléger *ULM*. Le bras manipulateur est équipé d'une caméra embarquée sur son effecteur. L'objectif de ce travail est de résoudre deux problèmes principaux : Le premier concerne le calcul de la position réelle 3D de la poignée de la porte. Le second concerne l'ajustement automatique de la pince du robot sur la poignée de la porte, exploitant les images issues de la caméra embarquée.

**.Mots clés :** manipulateur mobile; caméra embarquée ; la position réelle 3D; ajustement automatique.

---

**Abstract:** our project consists to carry out the opening door task by the mobile manipulator robot *RobuTER/ULM*, composed of lightweight manipulator arm *ULM* surmounted on *RobuTER* type mobile platform. The manipulator arm features eye-in-hand camera on its effector. The goal of this work is solving two main encountered problems. The first concern to calculate 3D real position of the door's handle. The second having treat the automatic adjustment of the clip on the door handle, exploiting from images of onboard camera.

**Keywords:** manipulator mobile ; onboard camera ; 3D real position ; automatic adjustment

---

# Tables des matières

---

Resume

Remerciement

Table des matières

Introduction générale.....6

## Chapitre 1 : Manipulateurs mobile

1.1.	Introduction .....	8
1.2.	Différents types des manipulateurs .....	9
1.2.1.	Multipèdes.....	9
1.2.2.	Manipulateurs mobiles sous-marin.....	9
1.2.3.	Manipulateurs mobiles a roues.....	10
	Caractéristique des manipulateurs mobile.....	11
	Capteur utilisés dans la robotique mobile .....	12
1.4.1.	Capteurs proprioceptifs.....	12
1.4.2.	Capteurs extéroceptifs .....	14
1.5.	Domaines d'applications des manipulateurs mobiles.....	18
1.5.1.	Environnements hostiles.....	19
1.5.2.	Industrie.....	19
1.5.3.	Militaire.....	20
1.5.4.	Agriculture.....	20
1.5.5.	Service médicale .....	20
1.5.6.	Domaine de service.....	21
1.6.	Conclusion.....	22

## Chapitre 2 : tâche d'ouverture une porte

2.1.	Introduction.....	23
2.2.	Formulation du problème.....	23
2.3.	Tâches d'ouverture de la porte .....	25
2.3.1.	Détection de la poignée de la porte .....	25
2.3.2.	Déplacement de la base mobile.....	25
2.3.3.	Ouverture de la porte .....	26
2.4.	Principaux travaux effectués.....	26
2.5.	Conclusion.....	36

## Chapitre 3 : calibration de système de vision

3.1.	Introduction.....	37
3.2.	Configuration caméra/robot.....	37
3.3.	Modélisation d'une caméra .....	38
3.4.	Différentes approches de la calibration.....	39
3.5.	ARToolKit.....	42
	3.5.1. Définition.....	42
	3.5.2. Reconnaissance de marqueur.....	44
3.6.	Calibration pince/caméra.....	47
	3.6.1. Définition.....	47
	3.6.2. Méthodes de calibration pince/caméra.....	47
3.7.	Conclusion.....	52

## **Chapitre 4 : mise en œuvre**

4.1.	Introduction.....	53
4.2.	Architecture du système robotique.....	53
	4.2.1. Architecture globale du système robotique.....	54
4.3.	Modélisation du système robotique .....	63
	4.3.1. Modélisation géométrique de bras manipulateur ULM.....	56
	4.3.2. Modèle géométrique et cinématique de la base mobile.....	64
4.4.	Calibration de la caméra .....	67
	4.4.1. Modèle géométrique de la caméra .....	67
	4.4.2. Calibration de la caméra .....	67
4.5.	Calcul des coordonnées 3D.....	72
4.6.	Conclusion.....	74

## **Chapitre 5 : résultats expérimentaux**

5.1.	Introduction.....	75
5.2.	Choix des outils de développement.....	75
5.3.	Interfaces des applications.....	76
5.4.	Ouverture d'une porte .....	80
	5.4.1. Étape 1.....	81
	5.4.2. Étape 2.....	84
5.5.	Résultat obtenus et discussion des résultats.....	84
5.6.	Conclusion .....	86

<b>Conclusion générale.....</b>	<b>87</b>
---------------------------------	-----------

## **Annexe 1**

### **Références bibliographie**

## Table des figures

<b>Figure 1</b> : Robot humanoïde ASIMO-HANDA.....	9
<b>Figure 2</b> : (a) Robot Victor 6000, (b) Robot Achille 2000.....	10
<b>Figure3</b> : Robot care-o-bot.....	10
<b>Figure 4</b> : Robot Curiosity.....	11
<b>Figure 5</b> : odomètre.....	12
<b>Figure 6</b> : principe de fonctionnement d'encodeur optique.....	13
<b>Figure 7</b> : les différents types des encodeurs.....	13
<b>Figure 8</b> : (a) : Gyroscope de Foucault (b) : Gyroscope électronique GY520 FUTABA.....	14
<b>Figure 9</b> : Différents capteurs à ultrason actuel <i>MSU04 et MS-EZ1</i> .....	15
<b>Figure 10</b> : Principe de fonctionnement du télémètre infrarouge Sharp.....	15
<b>Figure 11</b> : Télémètres lasers Sick (LMS 100, LMS 291).....	17
<b>Figure 12</b> : capteur de fin de secoure dans un articulateur d'un robot bipède PID.....	17
<b>Figure 13</b> : La capture d'une caméra panoramique.....	18
<b>Figure 14</b> : les robots d'environnements hostile.....	19
<b>Figure15</b> : Chaînes de montage de l'industrie automobile.....	19
<b>Figure16</b> : robot Mini-Andross.....	20
<b>Figure17</b> : robot TALON.....	20
<b>Figure 18</b> : Robot agricole Naïo.....	20
<b>Figure 19</b> : robot PerMMA.....	21
<b>Figure 20</b> : Robot chirurgical Da Vinci.....	21
<b>Figure21</b> : robot de Fujitsu.....	21
<b>Figure 22</b> : PR2, développé par Willow Garage.....	22
<b>Figure 23</b> : Représentation de la problématique.....	25
<b>Figure 24</b> : Robot base mobile nomade XR4000 et d'un bras manipulateur PUMA560.....	27
<b>Figure 25</b> : Ouverture d'une porte par le manipulateur mobile LIAS.....	28
<b>Figure 26</b> : Manipulateur mobile Ellie.....	29
<b>Figure 27</b> : Configuration initiale de DLR-Hand-II après avoir localisé la poignée de porte.....	30
<b>Figure 28</b> : Ouverture de porte par DLR-Lightweight-Robot-II.....	31
<b>Figure 29</b> : Le robot PR2 en train d'ouvrir une porte.....	32
<b>Figure 30</b> : Perception à base de laser pour l'identification de la porte et la poignée.....	32

<b>Figure 31</b> :Tirer sur la porte pour l'ouvrir.....	34
<b>Figure 32</b> :Poussez la porte pour l'ouvrir.....	34
<b>Figure 33</b> :Le robot autonome PR2 branche une prise électrique et ouvre une porte.....	34
<b>Figure 34</b> :La machine d'état utilisé par la tâche d'ouverture.....	35
<b>Figure 35</b> :Détection de la porte avec l'une des caméras stéréo par ARToolKit.....	36
<b>Figure 36</b> : les différents Configurations caméra/robot.....	38
<b>Figure 37</b> : Modèle sténopé de la caméra.....	39
<b>Figure 38</b> : Schéma de principe d'ARToolKit.....	41
<b>Figure 39</b> : matrice de transformation de Artoolkit.....	44
<b>Figure 40</b> : Binarisation de l'image.....	44
<b>Figure 41</b> : Recherche de composants liés.....	45
<b>Figure 42</b> : Recensement des arrêtes et coins du marqueur.....	45
<b>Figure 43</b> : Principe de normalisation.....	45
<b>Figure 44</b> : Principe de reconnaissance des marqueurs.....	46
<b>Figure 45</b> : Format de marqueur au niveau de gris.....	46
<b>Figure 46</b> : déplacement d'un système pince caméra.....	47
<b>Figure 47</b> : Nouvelle formulation de calibrage pince/caméra.....	50
<b>Figure 48</b> : la procédure d'étalonnage pince/caméra dans le cas d'une scène inconnue..	51
<b>Figure 49</b> : Architecture du système robotique.....	55
<b>Figure 50</b> : PC embarqué.....	56
<b>Figure 51</b> : Caméra IP Axis 213.....	57
<b>Figure 52</b> : RobuTER/ULM.....	58
<b>Figure 53</b> : Carte MPC555.....	58
<b>Figure 54</b> : Transformations de repères nécessaires.....	60
<b>Figure 55</b> : Paramètres de Denavit-Hartenberg .....	61
<b>Figure 56</b> : Paramètres de Denavit-Hartenberg du bras ULM.....	62
<b>Figure 57</b> : Représentation du repère de la base mobile $R_B$ dans le repère absolu $R_A$ .....	64
<b>Figure 58</b> : Différents paramètres de mouvement de la base mobile.....	65
<b>Figure 60</b> :Modèlecalib_dist et modèlecalib_cpara.....	68
<b>Figure 61</b> : L'utilisateur marque lespointsd'étalonnage.....	69
<b>Figure 62</b> : Exemple de l'image calibrée.....	69
<b>Figure 63</b> : Placement des lignes horizontal et des lignes vertical.....	70

<b>Figure 64</b> : image calibrée.....	70
<b>Figure 65</b> : Exemples de Patterns.....	71
<b>Figure 66</b> : Fenêtre video du 'mk_patt'.....	72
<b>Figure 67</b> : Interface opérateur de contrôle du manipulateur mobile RobuTER/ULM.....	76
<b>Figure 68</b> : Partie concernant la base mobile.....	77
<b>Figure 69</b> : Partie concernant le bras manipulateur.....	78
<b>Figure 70</b> : Acquisition par système de vision.....	78
<b>Figure 71</b> : Affichage des données du LMS.....	79
<b>Figure 72</b> : Affichage de la matrice poignée/bras.....	79
<b>Figure 73</b> : Partie concernant la tâche d'ouverture d'une porte.....	80
<b>Figure 74</b> : Paramètres personnalisés.....	81
<b>Figure 75</b> : Situation initiale du bras manipulateur après calibration de ce dernier.....	81
<b>Figure 76</b> : Position de capture pour la caméra.....	77
<b>Figure 78</b> : La poignée de la porte appartient à l'espace de travail.....	82
<b>Figure 79</b> : Calcul de la position 3D de la poignée.....	82
<b>Figure 80</b> : Ajustement de la pince sur la poignée.....	83
<b>Figure 81</b> : La poignée n'appartient pas à l'espace de travail actuel du robot.....	83
<b>Figure 82</b> : Différents tests d'ajustement effectués.....	84

## **La liste des tableaux**

<b>Tableau 1</b> : Caractéristiques du télémètre laser Sick LMS200.....	16
<b>Tableau 2</b> : les paramètres de déplacements et orientations de chaque articulateur.....	62
<b>Tableau 3</b> : Positions de la poignée calculées par rapport au repère lié au bras manipulateur.....	85







# Chapitre 1 Manipulateurs mobiles

---

## 1.1. Introduction

Historiquement, les premiers robots produits en furent les bras manipulateurs, leurs utilisations très souvent dans l'industrie. Ainsi ils se trouvent à des espaces structurés, dédiés à des tâches précises, simples comme la saisie d'objets, l'assemblage de pièces, etc. et d'autre part la robotique mobile été développée, caractérisées par leur aptitude à évoluer dans des environnements de grande taille. Ce type de robot est utilisé pour des missions de locomotion, notamment dans les milieux hostiles (nettoyeurs des métros, robots de surveillance etc.). Mais le besoin de robots ayant à la fois des capacités de locomotion et de manipulation, une nouvelle génération de robots est apparue, c'est les manipulateurs mobiles. Ce type des robots augmente les possibilités de franchissement d'obstacles, déplacement des objets (ouverture de porte, etc.).

Les manipulateurs mobiles consistent en des bras manipulateurs embarqués sur des bases mobiles. Ils ont l'avantage d'exécuter des tâches dans un environnement ouvert. De plus, leur indépendance par rapport à un fauteuil facilite leur mobilité. Ils constituent la solution la plus flexible afin d'améliorer l'autonomie de la robotique, mais ils sont plus complexes et difficiles à commander et à contrôler [1].

Pour réaliser un système de robotique mobile intelligent, il est nécessaire d'utiliser des capteurs qui permettent au robot pour mesurer les paramètres intérieurs et extérieurs à partir de l'environnement dans lequel il évolue, dans le but de guider ou de lui permettre de se localiser, il existe deux types des capteurs les capteurs proprioceptifs et les capteurs extéroceptifs.

## 1.2. Différents types des manipulateurs mobiles

Parmi les manipulateurs mobiles existant actuellement, nous pouvons citer trois grandes familles [2] :

### 1.2.1. Multipèdes

Comme leur nom l'indique, les manipulateurs mobiles « multipèdes » peuvent posséder une à plusieurs jambes.



robot humanoïde ASIMO- HANDA

Figure 1 : Robot humanoïde ASIMO-HANDA

À la gravité qu'ils représentent, les humanoïdes en sont les plus populaires représentants tant chez le grand public que chez les chercheurs. Du point de vue de la locomotion, ils posent le problème de la marche. D'un point de vue de la manipulation, le souhait de bio-mimétisme conduit à les équiper de mains sophistiquées dont la conception et la commande restent des sujets encore très ouverts. Leurs utilisations possibles sont limitées d'un point de vue industriel et leur principal débouché se situe dans la robotique de service (assistance aux personnes âgées, compagnons domestiques) ou, tels les robots humanoïdes ASIMO- HONDA (fig.1), comme démonstrateurs technologiques.

## 2.1. Manipulateurs mobiles sous-marins

Les manipulateurs mobiles sous-marins sont aujourd'hui les manipulateurs mobiles les plus utilisés à des fins de travail. Souvent télé-opérés, il s'agit alors de ROV (Remotely Operated Vehicles), ils permettent l'accès à des zones maritimes non accessibles aux plongeurs et fournissent des capacités de prélèvement, de manipulation, de mesure et d'acquisition de données adaptables aux missions envisagées.

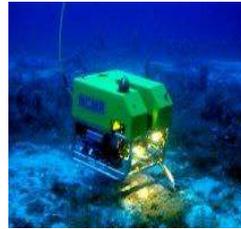


Figure 2 : (a) Robot Victor 6000, (b) Robot Achille 2000

Nous pouvons, à titre d'exemple, citer le robot Victor 6000 (Fig. 2 (a)) de l'IFREMER, équipé de deux bras manipulateurs et qui peut intervenir à des profondeurs de 6000 mètres ainsi que le robot Achille 2000 (Fig.2(b)) de la société Comex3, lui aussi équipé de deux bras manipulateurs et dont la profondeur principale d'intervention est de 2000 mètres [2].

### 2.3. Manipulateurs mobiles à roues

Moins médiatiques que les premiers, moins exploités que les seconds, les manipulateurs mobiles à roues sont cependant plus répandus que ceux précédemment cités. Cela tient particulièrement à deux faits [2] :

- leur conception mécanique relativement simple.
- l'adéquation naturelle de leur moyen de locomotion à un grand nombre de terrains [2].

Les différences majeures entre ces systèmes à roues sont principalement liées au choix du type de locomotion. Ainsi, nous distinguons :

- Les manipulateurs mobiles à roues, tels que le robot Care-o-bot du Fraunhofer Institut (Fig. 3), utilisant des mécanismes de suspension rudimentaires et plus adaptés à des environnements d'intérieur lisses et assez plats.



Figure3 : Robot care-o-bot

- Les manipulateurs mobiles à roues, tels que le robot Curiosity (fig. 4), dont la base mobile peut-être dite tout terrain.

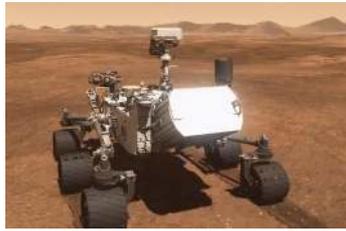


Figure 4 : Robot Curiosity

### 1.3. Caractéristiques des manipulateurs mobiles

On présente les propriétés principales des manipulateurs mobiles dans les paragraphes suivantes :

- **Les capteurs utilisés et leurs positions** : les types de capteurs utilisés et leurs positions représentent un critère important dans le choix d'un manipulateur mobile, souvent l'environnement de manipulation (la localisation et la navigation).
- **holonomie** : un système mobile holonome peut se déplacer dans toutes les directions depuis sa position courante. La contrainte non holonome est la limitation des vitesses admissibles d'un objet, ce qui veut dire que le robot non holonome ne peut effectuer de mouvement instantanément que dans certaines directions [3].
- **Nombre de manipulateur installés et leurs emplacements sur la base mobile** : en général, un manipulateur mobile est composé d'une base mobile surmontée d'un seul manipulateur. Toutefois, ils existent des manipulateurs mobiles qui disposent de deux manipulateurs (dual-arm) comme c'est le cas du robot ALIA [4].

Un autre critère très important est l'emplacement de manipulateur au-dessus de la base mobile qui doit être placé de sorte à permettre l'équilibre de l'ensemble. De plus et selon le type de la tâche à accomplir par le manipulateur mobile, il peut se trouver à l'avant de la base dans le cas de l'ouverture d'une porte par exemple [5] ou à côté dans le cas d'une tâche de saisie d'un objet avec la base mobile en mouvement.

- **Nombre et position des roues motrices et des roues faible de la base mobile** : pour faire en sorte que le robot se déplace correctement selon la consigne, la position et le type des roues doivent être connus. Il y a des manipulateurs l'emplacement des roues différentielles (latérales, situées au milieu, etc.) motrice et directrices. De plus, deux

roues libres peuvent être placées de telle sorte à équilibrer la base mobile comme pour le manipulateur mobile H2bis de LAAS [2].

- **Taille du robot** : la taille des robots représente un critère de sélection important, car dans des environnements fortement encombrés par exemple, les petits robots peuvent être d'un grand secours, par contre, s'il y a nécessité de transport d'objets lourds, alors les grands robots sont très intéressants [3].
- **le mode de manipulation** : les manipulateurs mobiles généralement divisés en deux missions, la locomotion qui assure par la base mobile et la tâche de manipulation qui se fait avec un ou plusieurs bras articulés. Il y a des cas particuliers où les roues qui assurent la tâche de déplacement et la manipulation au même temps.

## 1.4. Capteurs utilisés dans la robotique mobile :

Il existe deux catégories de capteurs couramment utilisées en robotique mobile, les *capteurs proprioceptifs* et les *capteurs extéroceptifs*.

### 1.4.1. Capteurs proprioceptifs

Les capteurs proprioceptifs sont installés à bord du robot et mesurent les états internes de celui-ci (la position et vitesse de roues, l'angle de rotation, etc.).

#### a. Odomètres

Les odomètres sont des capteurs qui permettent d'estimer le déplacement à partir de la mesure de la vitesse et la position angulaire de l'axe des roues, les encodeurs optiques sont les plus utilisés pour effectuer la mesure de la rotation. Cet encodeur dispose sur l'axe de la roue ou sur le moteur [4].



Figure 5 : odomètre

Le principe de fonctionnement de ses encodeurs optique, une lumière émise par une diode électroluminescente est réfléchié par les graduations d'un disque vers une photodiode ou phototransistor qui se sature et se bloque à la cadence du défilement des graduations (Fig. 6).

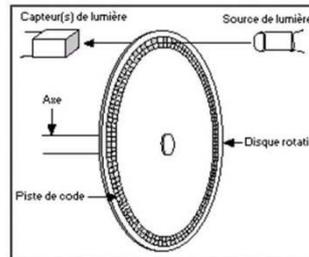


Figure 6 : principe de fonctionnement d'encodeur optique

Il existe néanmoins d'autres types des encodeurs tels que les encodeurs absolus «Fig. 7(a) ». Les encodeurs magnétiques «Fig. 7(b) » les potentiomètres « Fig. 7(c) ».



Figure 7 (a) Encodeur absolu (b) Encodeur magnétique (c) Capteur rotatif potentiomètre

### b. Les accéléromètres.

L'accéléromètre est un capteur qui mesure l'accélération linéaire en un point donné. En pratique, la mesure de l'accélération est effectuée à l'aide d'une masse d'épreuve  $M$ , de masse  $m$ , reliée à un boîtier du capteur. Le principe de ce capteur est de mesurer l'effort massique non gravitationnel qu'on doit appliquer à  $M$  pour le maintenir en place dans le boîtier lorsqu'une accélération est appliquée au boîtier. Le calcul du déplacement élémentaire du système mobile est obtenu par double intégration de ces informations. Cette double intégration conduit généralement à des accumulations importantes d'erreurs. Ce capteur est plus coûteux que des odomètres. [4]

### c. Gyroscopes

Les gyroscopes sont des capteurs permettant de mesurer la position angulaire par rapport à un référentiel inertiel, il existe plusieurs sortes de gyroscopes : mécanique et optique, sont les plus connus, ainsi que les structures vibratoire et capacitifs [5].

L'image classique d'un gyroscope mécanique est un rotor suspendu par un cardan sans friction isolant le centre du rotor des couples extérieurs, comme le montre « Fig. 8(a) »



Figure 8(a) : Gyroscope de Foucault (b) : Gyroscope électronique GY520 FUTABA

### 1.4.2. Capteurs extéroceptifs

Les capteurs extéroceptifs permettent au robot pour mesurer des paramètres extérieurs à partir de l'environnement dans lequel il évolue.

Ces capteurs sont nécessaires dans les domaines d'application tels que l'évitement d'obstacle, la localisation, la navigation et la modélisation d'environnements.

#### a. Télémètres

Les télémètres sont des capteurs qui utilisent pour connaître la distance qui sépare le robot par rapport d'un objet ou obstacle. Il existe plusieurs technologies pour réaliser un télémètre. Les télémètres les plus utilisés dans la robotique mobile nous allons présenter dans la suite :

- **Télémètres ultrason** : Les capteurs à ultrasons (ou télémètres ultrasonores) sont aujourd'hui les capteurs les plus communément employés dans les systèmes de robotique mobile autonome en raison de leurs bas prix, de leur faible consommation, de leur simplicité et de leur compacité (Figure 9). Ils ont été exploités dans des domaines tels que la navigation, la modélisation d'environnement, l'évitement d'obstacle ou l'estimation de la position [4]. Le principe de base d'un télémètre ultrason utilisant consiste à transmettre un paquet d'onde ultrasonique ensuite à

mesuré le temps qu'on met cette onde à revenir au capteur, après avoir été réfléchi par un objet ou obstacle. La distance de l'objet par rapport au capteur peut être calculé connaissant la vitesse de propagation de son et le temps de vol pour un aller-retour avec considération des constantes de la chaleur et de gaz.



Figure 9 : Différents capteurs à ultrason actuel *MSU04* et *MS-EZ1*

- **Télémètres infrarouges** : Les capteurs ultrason et infrarouge sont le même principe de fonctionnement (émetteur-récepteur), en utilise une source lumineuse par une diode électroluminescente infrarouge. Le récepteur est une photodiode ou un phototransistor [5].

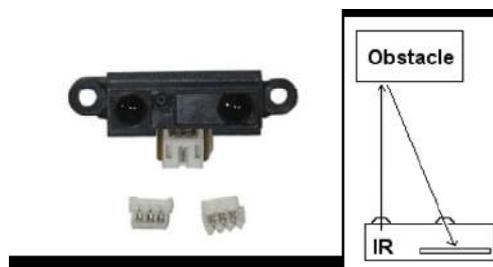


Figure 10 : Principe de fonctionnement du télémètre infrarouge Sharp

- **Télémètres laser** : Les télémètres laser sont à ce jour le moyen le plus répandu en robotique mobile pour obtenir des mesures précises de distance. Leur principe de fonctionnement est le suivant. A un instant donné, une impulsion lumineuse très courte est envoyée par l'intermédiaire d'une diode laser de faible puissance. La réflexion de cette onde donne un écho qui est détecté au bout d'un temps proportionnel à la distance capteur-obstacle. La direction des impulsions est modifiée par rotation d'un miroir, l'angle de balayage couvrant généralement entre 100 et 180 degrés sur des produits commerciaux. La portée du capteur dépend de la réflectivité

des milieux rencontrés, mais une valeur typique de 30 mètres est atteinte avec un télémètre de bonne qualité. Outre cette portée relativement satisfaisante pour une application de navigation à basse vitesse, les autres performances de ces capteurs en termes de précision de mesure, de résolution angulaire et de stabilité en température font d'eux les meilleurs télémètres pour la robotique mobile. La gamme de produits de la marque Sick (voir Fig. 11) fait référence pour la navigation des robots mobiles. Les performances d'un modèle tel que le LMS200 sont données dans le tableau 1 [6].

Angle d'ouverture	180°
Résolution angulaire	0,25 – 0,5 – 1°
Temps de réponse (fonction de la résolution)	13 – 26 – 52 <i>ms</i>
Résolution	10 <i>mm</i>
Erreur systématique	-/+ 15 <i>mm</i>
Erreur statistique	5 <i>mm</i>
Classe d'équipement laser	Classe 1
Températures de fonctionnement	0° C ----- + 50° C
Distance maximale de mesure	80m
Interface	RS-442 et RS-232
Taux de transmission	9,6 – 19,2 – 38,4- 500 <i>k Baud</i>
Consommation	20 <i>W</i>
Poids	4,5 <i>kg</i>
Dimensions (L x l x h)	156 x 155 x 210 <i>mm</i>

Tableau 1 : Caractéristiques du télémètre laser Sick LMS200



Figure 11 : Télémètres lasers Sick (LMS 100, LMS 291)

### b. Les capteurs de fin de secoure :

Les capteurs de fin de secoure sont des détecteurs sensibles au contact. Il existe plusieurs variétés de capteurs de fin de secoure qui peuvent avoir de nombreuses formes et différents niveaux de sensibilité. Ces capteurs sont utilisés au-devant de la base mobile pour éviter la collision avec les obstacles ou aux extrémités de secoure dans chaque articulations de bras mobile pour éviter qu'une articulation atteigne sa butée mécaniques (Fig12) [7]



Figure 12 : capteur de fin de secoure dans un articulateur d'un robot bipède PID

### c. Caméras

Les caméras sont des capteurs largement utilisées dans la robotique mobile, car elle semble proche des méthodes utilisées par les humains en plus ces capteurs fournissant grande quantité d'information de l'environnement à partir d'une seule acquise. Le traitement des données volumineuses et complexes fournis par ces capteurs très souvent difficiles, pour cela en appliquant plusieurs méthodes et algorithmes pour le traitement d'image. Il existe plusieurs types des caméras nous citons trois plus connues :

- **Caméras simples** : Une caméra standard peut être utilisée de différentes manières pour la navigation d'un robot mobile. Elle peut être utilisée pour détecter des amers visuels (des points particuliers qui servent de repère, tels que des portes ou des

affiches) à partir desquels il sera possible de calculer la position du robot. Si ces amers sont simplement ponctuels, ou de petite taille, il sera en général simplement possible d'estimer leur direction. Dans le cas où les amers sont des objets connus en 2 ou 3 dimensions, il sera en général possible d'estimer complètement la position du robot par rapport à la leur. Elle peut également être utilisée pour détecter des "guides" de navigation pour le robot, tels que des routes ou des couloirs [8].

- **Caméras stéréoscopiques** : Lorsque l'on dispose de deux caméras observant la même partie de l'environnement à partir de deux points de vue différents, il est possible d'estimer la distance des objets et d'avoir ainsi une image de profondeur, qui peut être utilisée pour l'évitement d'obstacles ou la cartographie [8].
- **Caméras panoramiques** : La caméra panoramique (omnidirectionnelle) consiste à mesurer la réflexion de l'environnement sur un miroir parabolique. La mesure ainsi obtenue est très largement distordue, mais possède l'avantage d'être omnidirectionnelle et donc de permettre une reconstruction panoramique de la scène sur 360 degrés [9]. Fig13 donne un exemple de dispositif et d'images obtenus.



Figure 13 : La capture d'une caméra panoramique

## 1.5. Domaines d'applications des manipulateurs mobiles

La complexité des tâches réalisables par le véhicule dépend des performances de son système de localisation (précision, fiabilité, robustesse, temps de réponse, etc.) et de la mobilité que lui confère son système de locomotion. Un robot manipulateur est donc généralement conçu pour effectuer un certain nombre de tâches que l'on peut classer en fonction du type d'environnement dans lequel évolue le véhicule (milieu structuré ou semi-structuré, environnement naturel, etc.), du degré de connaissance de cet environnement, de

son caractère évolutif (obstacles fixes ou mobiles) ou encore du domaine d'application pour lequel le véhicule a été conçu.

Le développement des recherches en manipulateur mobile concerne un grand nombre d'applications, parmi lesquelles :

### 1.5.1. Environnements hostiles

L'utilisation des manipulateurs mobiles en milieu hostile se développe dans le domaine spatial qui s'intéresse aux problèmes d'exploration planétaire (Fig. 14(a)) et d'intervention sur les stations (Fig. 14(b)). Ces robots sont utilisés aussi dans le domaine nucléaire pour la surveillance et la maintenance des centrales (reconnaissance et cartographie des niveaux de radiation), l'aide au démantèlement/assainissement des sites radioactifs (Fig.14(c)).



Figure 14 : (a) Robot Curiosity (b) Manipulateur agile spécialisé de MDA (c) Robot de Mitsubishi (assistance nucléaire)

### 1.5.2. Industrie

L'utilisation des manipulateurs mobiles industriels comme des contrôles automatique reprogrammable ou remplacent l'humain dans les tâches pénibles et dangereuse (robots de soudages, de peinture et d'assemblage) (Fig.15).



Figure15 : Chaînes de montage de l'industrie automobile

### 1.5.3. Militaire :

Les manipulateurs mobiles militaires sont des robots, autonome ou contrôlé à distance, conçu pour la plupart dans le champ des robots d'interventions. Sont capables d'effectuer des tâches de déminage ou de destruction (Fig16). De plus, certains robots sont équipés d'un armement (Fig17) pour évoluer en milieu hostile, dans le but de remplacer les soldats pour limiter les pertes humaines.



Figure16 : robot Mini-Andross



Figure17 : robot TALON

### 1.5.3. Agriculture

Les manipulateurs mobiles ont aussi été créés pour faciliter la vie aux humains, et leur éviter des travaux lassants et épuisants. C'est le cas d'un robot **Naïo** dont la principale fonction est le **désherbage mécanique** des exploitations maraichères. Simultanément, il peut aussi collecter des données telles que la température, pression, humidité, etc. [10] (Fig.



18).

Figure 18 : Robot agricole Naïo

### 1.5.4. Service médicale

Ce domaine est basé pour le développement de véhicules pour l'aide aux handicapés (Fig19), ainsi dédiés pour des applications médicales et chirurgicales, exemple de robot Da

Vinci. Ce type de robot a été développé. Celui-ci est contrôlé à distance par les mains de l'habile chirurgien à l'aide de télémanipulateurs et de caméras (Fig. 20).



**Figure 19** : robot PerMMA



**Figure 20** : Robot chirurgical Da Vinci

### 1.5.5. Domaine de service :

Un manipulateur mobile domestique ou de service, est un robot utilisé pour des tâches ménagères, par exemple en vaisselle, en repassage, en nettoyage, etc. Ils peuvent également être utilisés dans le domaine de la restauration. Ce type des robots est très utilisé par les handicapés, pour faciliter leur vie. Le robot (Fig21) est un robot domestique commercialisé par Fujitsu



**Figure21** : robot de Fujitsu

Aussi le robot PR2 (Personal Robot 2) commercialisé Willow Garage[12], Ce robot possède deux bras à sept degrés de liberté montés sur une base mobile. Il est également équipé d'un large panel de capteurs (camera, télémètre laser, capteur tactile sur les pinces, centrale inertielle...)(Fig22).



**Figure 22** : PR2, développé par Willow Garage

## 1.6. Conclusion

Dans ce chapitre nous avons définis les différents types des manipulateurs mobiles qui combinent la tâche de locomotion et la tâche de manipulation et leurs principes caractéristiques, ainsi les capteurs nécessaires pour mesurer les états internes et externes de robot, à la fin nous présentons les domaines d'évolution de les manipulateurs mobiles.

L'objectif du chapitre suivant est de présenter la formulation mathématique du problème d'ouverture d'une porte par un manipulateur. Le chapitre décrit aussi l'état de l'art sur les divers travaux effectués traitant ce problème.

# Chapitre 2 Tâche d'ouverture d'une porte

---

## 2.1. Introduction

En robotique, les manipulateurs mobiles à roues sont actuellement des systèmes très privilégiés lorsqu'il s'agit de tâches qui impliquent, en même temps, la manipulation et la locomotion [13]. La conjonction de ces deux fonctions démultiplie leurs possibilités respectives [14]. Ainsi, l'espace de travail sera plus large et des évitements ou des franchissements d'obstacles sont rendus plus faciles. Dans notre cas, nous considérons un robot manipulateur mobile d'intérieur RobuTER/ULM pour effectuer une tâche correspondant à l'ouverture d'une porte. La détermination de l'emplacement de la poignée dans l'espace 3D en utilisant une caméra embarquée et fixée sur l'effecteur est un problème clé à résoudre au cours de la navigation du robot.

Le début de ce chapitre est dédié à la formulation du problème d'ajustement visuel automatique de la pince du manipulateur mobile sur la poignée de la porte. Enfin, on cite les principaux travaux qui traitent ce problème ainsi que les principaux résultats obtenus.

## 2.2. Formulation du problème

En robotique, nous sommes confrontés très souvent au problème de la détermination de la position et de l'orientation d'un objet dans son environnement qui sera utilisé pour la manipulation robotique. Pour la détermination de ses coordonnées, nous utilisons divers capteurs tels que les caméras, les scanners laser, des marqueurs spéciaux, etc.

Les capteurs les plus universels sont les caméras qui imitent la vision humaine. D'autre part, ils nécessitent des algorithmes plus sophistiqués pour la détermination et la reconnaissance d'objets.

Pour la détermination de l'emplacement de l'objet dans l'espace 3D, la solution la plus courante sont des caméras stéréo. Car il est bien connu que ce n'est pas possible d'estimer la position d'un point dans l'espace 3D en utilisant une seule caméra [15]. D'autre part, nous savons que les humains peuvent saisir et manipuler des objets en utilisant un seul œil. Cette capacité peut être imitée aussi avec la vision par ordinateur. Si nous connaissons l'apparence de l'objet et nous l'avons sous différents angles, nous pouvons estimer sa position à l'aide d'une seule caméra. Si nous avons complété la représentation géométrique de l'objet, puis, la position peut également être calculées analytiquement en utilisant une seule image, alors nous pouvons calculer exactement la position de l'objet dans l'espace en supposant que la caméra est calibrée. Cette fonctionnalité est offerte, également par la bibliothèque ARToolKit.

Dans ce travail, on essaie de réaliser la tâche d'ouverture d'une porte par un manipulateur mobile pour valider l'ajustement visuel et automatique de la pince sur la poignée de la porte. Le marqueur se trouve dans le champ de vision de la caméra alors, une stratégie pour la saisie de la poignée est définie par :

1. Déplacer la base mobile vers une position en face de la porte de sorte que le bras manipulateur peut atteindre la poignée. Il faut aussi que le marqueur reste toujours dans le champ de vision de la caméra.
2. Le système de vision détermine la position de la poignée.
3. Le système de planification de saisie choisit la position de saisie de la poignée et planifie le chemin à suivre.

Pour commander le manipulateur mobile, il est nécessaire, entre autres, de disposer de *modèles* du mécanisme. Plusieurs niveaux de modélisation sont possibles, il en découle des modèles *géométriques*, *cinématiques* et *dynamiques* à partir desquels peuvent être engendrés les mouvements du robot, ou bien des modèles *statiques* qui décrivent les interactions du mécanisme avec son environnement [16].

L'obtention de ces différents modèles n'est pas aisée. La difficulté varie selon la complexité de la cinématique du robot. Entrent en ligne de compte le nombre de degrés de

liberté et le type des articulations. La Fig. 1 donne une représentation simple de ce problème.

- **Problématique 1** : comment le robot détecte la poignée de la porte ?
- **Problématique 2** : à quel moment la base mobile va se déplacer ?
- **Problématique 3** : comment commander le bras manipulateur jusqu'à atteindre la poignée ?

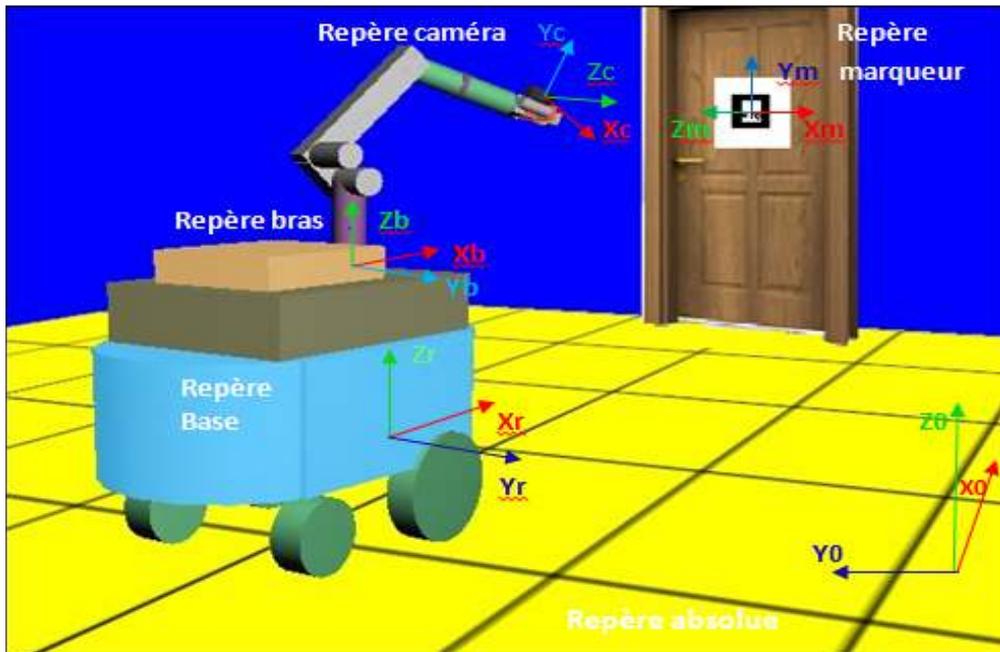


Figure 23 : Représentation de la problématique

## 2.3. Tâches d'ouverture de la porte

### 2.3.1. Détection de la poignée de la porte

L'algorithme de détection de la poignée est basé sur le traitement d'image en utilisant la caméra IP embarquée et fixée sur l'effecteur. Elle est aussi utilisée pour déduire les caractéristiques géométriques de la porte (si elle s'ouvre vers l'intérieure ou l'extérieure) et la position 3D de la poignée à partir d'un marqueur posté sur la porte.

### 2.3.2. Déplacement de la base mobile

Le marqueur doit être dans le champ de vision de la caméra. A cet effet, le système de vision détecte si les coordonnées de la poignée sont en dehors de l'espace de travail du

bras manipulateur. Si ce cas se présente, la base mobile va se déplacer de sa position initiale jusqu'à atteindre une position finale qui permettra au bras manipulateur d'atteindre la poignée. Aussi, des systèmes de navigation sont développés pour permettre au robot d'éviter les obstacles sur la trajectoire de la base mobile.

### **2.3.3. Ouverture de la porte**

La base mobile s'arrête à une certaine distance face à la porte. Le marqueur doit rester dans le champ de vision de la caméra afin que le système de vision puisse calculer les coordonnées 3D de la poignée.

Le robot doit ouvrir la pince, puis, suivre une trajectoire opérationnelle pour atteindre la poignée. Avec la tenue de l'effecteur sur la poignée de la porte, les degrés de liberté du mouvement de bras manipulateur sont limités à deux degrés, la rotation de la porte autour de ses charnières, et le déverrouillage de la poignée.

Bien que le bras manipulateur du robot soit le responsable de l'ouverture de la porte, la base mobile se déplace pour augmenter l'espace de travail du bras manipulateur. Elle se déplace en continu avec le bras lorsque la porte s'ouvre de plus en plus, donc la pince devrait rester toujours sur la poignée de porte.

Pour éviter les collisions entre la base mobile et son environnement, le dispositif de commande de la base mobile repose sur des mesures à partir des capteurs (LMS et US).

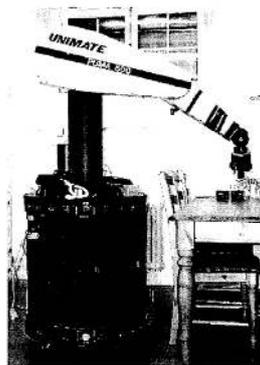
## **2.4. Principaux travaux effectués**

Peterson et ses collègues [17], réalisaient la tâche d'ouverture de porte en utilisant le système dynamique hybride. Ils décomposent ce problème en plusieurs étapes [17] :

1. Déplacer le robot à une certaine position devant la porte.
2. Localiser la poignée de porte.
3. Déplacé le bras manipulateur du robot jusqu'à atteindre la poignée de la porte.
4. Saisir la poignée.
5. Faire tourner la poignée.
6. Ouvrir la porte.

Les auteurs de [17] utilisent un robot équipé d'une plateforme nomade XR4000 et d'un bras manipulateur puma 560 pour la manipulation. La première étape de la mission consiste à

déplacer le robot devant la porte. Ce déplacement se fait à l'aide de la localisation et de module de navigation. L'étape suivante consiste à trouver la poignée de la porte dans l'image capturée de la porte. L'étape trois consiste à placer l'effecteur de façon perpendiculaire à la poignée de porte. Par la suite, un mouvement droit est alors lancé jusqu'à atteindre la poignée de porte. Puisque la vision ne permet pas de fournir des informations très précises de profondeur, le laser est utilisé pour mesurer la distance à la porte. Quand l'effecteur est à environ 10cm de la poignée, il est nécessaire de faire une prise sans visibilité de la poignée, car la poignée n'est plus visible dans l'image. Cette prise continue dans la direction de profondeur jusqu'à ce que le capteur de force indique que le contact est établi. La pince est alors fermée autour de la poignée qui est tournée. Puis, à chaque moment le système dynamique hybride doit estimer le rayon et le centre de rotation. Après, il exercera des forces et couples à l'effecteur pour changer sa direction de façon à suivre correctement l'arc. Un mécanisme est appliqué pour déplacer simultanément la base mobile et l'effecteur pour atteindre la configuration désiré.



**Figure 24** : Robot équipé d'une base mobile nomade XR4000 et d'un bras manipulateur PUMA560.

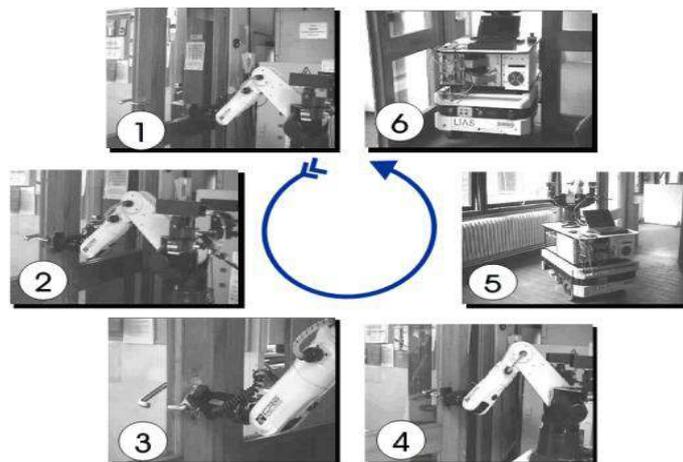
Waarsing [18] réalise la tâche d'ouverture d'une porte par le manipulateur mobile LIAS. Ce robot est équipé d'une plateforme commerciale « RobuTER » et d'un manipulateur industriel « SRI A465 ». La base mobile est équipée de nombreux capteurs (un module de balayage à laser, etc.). Le robot est équipé aussi d'un système de vision composé de plusieurs caméras pour permettre la vision stéréo. Le manipulateur à six ddl est équipé d'un capteur de force/couple et d'une pince. Waarsing a découpé la tâche en trois étapes [18] :

1. *Déplacer le robot devant la poignée de porte* : un mouvement direct de la base mobile est exécuté. Puis, elle s'arrête à une distance devant la porte. Un

comportement semblable et défini pour commander et suivre le bras manipulateur jusqu'à atteindre la poignée.

2. *Ouvrir la porte* : deux comportements du bras manipulateur sont utilisés pour faire tourner la poignée puis pousser. La combinaison de ces deux comportements laisse le manipulateur tirer une porte ouverte. Mais, puisque la zone de travail du manipulateur est limitée, ce comportement posera des problèmes quand le manipulateur atteint ses limites. Par conséquent, la base mobile devrait assurer le mouvement à un certain point.
3. *Franchir la porte* : quand la porte est entièrement ouverte, trois comportements sont utilisés pour assurer le franchissement de la porte :
  - Conduire en avant.
  - Eviter des obstacles à droit.
  - Eviter des obstacles à gauche.

Avec cette méthode, le robot n'a pas besoin de connaître la situation de la porte, c.-à-d. si la porte balance vers la droite ou vers la gauche, ni si elle s'ouvre vers l'intérieur ou à l'extérieur. Quand le robot note que la direction de son mouvement a changé assez, il suppose que la porte doit être ouverte et déclenche des commutateurs à la phase suivante.



**Figure 25** : Ouverture d'une porte par le manipulateur mobile LIAS.

Advait et Charles [19] présentent un ensemble de comportements qui permettent à un manipulateur mobile d'ouvrir sûrement une série de portes et de portes transversales en utilisant des doigts et un télémètre laser. Au début, l'opérateur humain illumine une poignée

de porte en utilisant un indicateur laser vert. Puis, le robot, à l'aide de caméra stéréo se relève à la poignée de porte, s'aligne avec la porte, localise la poignée, détecte le bout mobile, tord la poignée, pousse la porte, estime l'ampleur de la porte et traverse, alors, la porte tout en ouvrant la porte pour fournir le dégagement. Pour accomplir les deux buts, on a décomposé la tâche globale d'ouverture et de traverse d'une porte en plusieurs tâches secondaires. Les comportements développés qui peuvent accomplir chacune de ces tâches secondaires sont :

4. Localiser la poignée de porte
1. Décider si la porte est verrouillée
2. Tourner la poignée de porte
3. Décider si la porte peut être poussée
4. Pousser la porte et traverser la porte

Advait et Charles ont accompli ce travail avec le manipulateur mobile statiquement stable « Ellie ». Ce robot se compose d'un bras manipulateur KATANA 6M, une base mobile ERRATIQUE et un déclencheur linéaire « Zenither » qui peut soulever le bras manipulateur et les divers capteurs du niveau du sol à 90cm au-dessus du sol. « Ellie » utilise des doigts de détection de force. La rétroaction sensorielle d'utilisation de comportements, permet sans interruption de surveiller le changement des aspects de l'environnement et répondent à la variation de la tâche, comme si la porte est verrouillée ou débloquée, bloquée ou dégagée, s'ouvre à droite ou à gauche, ou a une poignée qui tourne vers le bas dans le sens des aiguilles d'une montre ou dans le sens contraire des aiguilles d'une montre. Mais, malheureusement ne traitent pas le cas où la porte s'ouvre à l'intérieure ou à l'extérieure. Ils ont examiné le robot dans 5 épreuves pour chacune des 6 portes avec 90% comme indices de réussite.



**Figure 26** : Manipulateur mobile Ellie.

Christian Ott et ses collègues [20] appliquent la tâche d'ouverture d'une porte sur un système mobile qui est une combinaison du « DLR-Lightweight-Robot-II » et de « DLR-Hand-II » avec une base mobile omnidirectionnelle. Le DLR-Lightweight-Robot-II est équipé de capteurs communs de couple en plus aux capteurs. Le bras DLR-Hand-II est équipé des capteurs communs de couple et une main. Cette main est composée de trois doigts, chacun à trois degré de liberté. Le bras DLR-Hand-II est à six degré de liberté. La base mobile directive a quatre roues. Ce système est équipé aussi d'un laser SICK et un système de vision qui se compose de trois caméras à bord (DIGICLOPS), et une collection de capteurs de distance incluses dans quelques parties de la bache pour mesurer les forces externes exercées sur la plateforme mobile. L'application entière de la mission d'ouverture d'une porte est divisée en trois tâches secondaires données comme suit [20] :

- *Localisation de la poignée de porte* : pour la localisation de la poignée de la porte, le robot se déplace devant la porte. Pour que ce déplacement soit fiable, la base mobile utilisera un laser pour le paléage et un système de vision pour mesurer la distance et l'orientation à la porte. Puis, la plateforme va se déplacer. Par la suite, un algorithme est appliqué pour chacune des trois images de la caméra pour extraire les coordonnées de la poignée. Après la localisation de la poignée de la porte, le bras du robot va se mouvoir jusqu'à atteindre la poignée de la porte (Fig. 27).



**Figure 27** : Configuration initiale de DLR-Hand-II après avoir localisé la poignée de porte.

- *Rotation et Ouverture de la poignée de porte* : la rotation et l'ouverture de la poignée de porte est fractionnée en plusieurs étapes. Les deux premières étapes consistent à déplacer le bras à gauche afin d'introduire la poignée dans la main du robot, puis se déplacer en arrière pour contacter la main avec la poignée. Quand le contact est établi, la main va se fermer.

La troisième étape consiste à faire tourner la main du robot. Dans cette étape, le mouvement de rotation réelle de la poignée se termine quand la force de contact dépasse une valeur prédéfini. Cette valeur indique que la poignée est tournée à son angle maximum. Après ceci, la porte est ouverte par un mouvement horizontal de 30mm dans la direction de la normale à la surface de porte et la poignée est tournée de nouveau à sa configuration libérée.



**Figure 28** : Ouverture de porte par DLR-Lightweight-Robot-II

- *Mouvement du robot au moment d'ouverture de porte* : dans cette étape, la porte sera déplacée jusqu'à ce qu'elle soit entièrement ouverte. L'idée principale pour ouvrir la porte est simplement de commander le bras tel que la porte est maintenue à une distance, alors que la base mobile se déplace sur une ligne droite. On note que cette méthode fonctionne sans aucune connaissance de la taille de la porte.

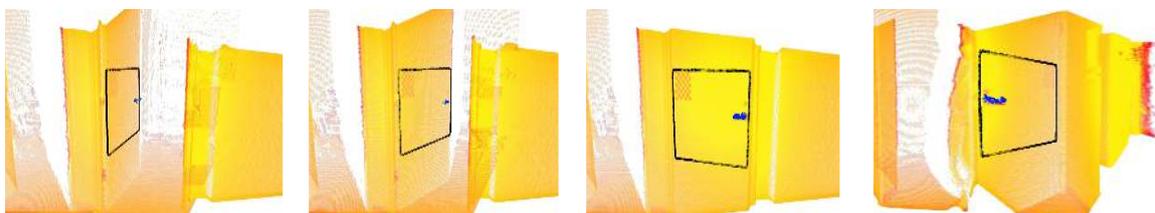
Sachin Chitta et ses collègues de Willow Garage [21] réalisent cette approche sur le robot PR2. Ce dernier est un manipulateur mobile avec une base omnidirectionnelle et deux bras à 7 degrés de liberté. Il dispose d'un vaste ensemble de capteurs utiles pour manipulation mobile comprenant un scanner laser basculant monté sur la tête, deux paires de caméras stéréo, un laser supplémentaire monté sur la base. Des Encodeurs sur chaque articulation fournissent, également, les angles d'articulation. Chaque bras possède, également, une caméra supplémentaire montée sur l'effecteur. Chaque pince comporte des capteurs de pression capacitifs qui peuvent être utilisés pour détecter la pression appliquée par la pince sur un objet. Les deux capteurs utilisés pour ce travail sont le laser de base et le balayage laser monté sur une plateforme basculante sur la tête.

Ces auteurs [21] montrent comment surmonter la haute dimensionnalité du problème en identifiant une représentation basée sur les graphes qui est assez petit pour une planification efficace encore assez riche pour contenir des mouvements réalistes qui ouvrent les portes. L'utilisation d'un graphe basé sur la recherche de planification de mouvements a permis de manipuler systématiquement la grande diversité des conditions dans lesquelles les portes doivent s'ouvrir.



**Figure 29** : Le robot PR2 en train d'ouvrir une porte.

Cette approche d'ouverture de la porte a deux composantes (a) la détection de la porte et la poignée et (b) la préparation et exécution de l'action d'ouverture de porte. L'approche décrite dans [22] de détection de la porte agit directement sur l'ensemble de points acquis de nuages de données provenant du capteur laser, et ne fait aucune utilisation des images de la caméra ou toute autre source d'informations pour identifier le plan de la porte et les coordonnées 3D de la poignée (Fig. 30).



**Figure 30** : Perception à base de laser pour l'identification de la porte et la poignée

L'approche actuelle de la tâche d'ouverture de porte repose sur un modèle connu de la porte. Le modèle n'a pas besoin d'être extrêmement précis, mais il doit spécifier la position de la poignée de la porte et sa charnière à un système de coordonnées de référence commune. Le système de détection de la porte et la poignée [22] accède à un modèle antérieure de la

porte y compris sa position, la direction de rotation et de l'emplacement de la charnière à partir d'une base de données topologique du bâtiment. Il utilise cette information comme une hypothèse initiale pour affiner son estimation de l'emplacement de la porte, la taille et l'emplacement de la poignée à l'aide de basculement du scanner laser monté sur la tête. La sortie de l'algorithme de détection de la porte donne l'emplacement de la poignée et l'estimation de l'endroit où la charnière est.

Le système de planification utilise le modèle de la porte estimée fourni par la perception pour exécuter une série de tâches secondaires :

1. Déplacer la base vers une position en face de la porte de sorte que le bras peut atteindre la poignée.
2. Tendre la main, saisir la poignée et déverrouillez la porte.
3. déterminer à partir du modèle si la porte doit être poussée ou tirée pour l'ouvrir. Poussez ou tirez légèrement la porte pour initier l'action d'ouverture.
4. Planifier un mouvement de bras et de la base pour ouvrir la porte. Le mouvement du bras et de la base mobile sont couplés à faire en sorte que la poignée de la porte reste à l'intérieur de l'espace de travail accessible du bras, et le mouvement généré par la base est libre de collisions avec tous les obstacles.
5. Si on tirant la porte ouverte, vérifier les collisions du bras avec la porte. Si les collisions sont détectées, relâcher la poignée sur un côté de la porte et déplacer le bras de manière à pouvoir saisir la poignée de l'autre côté de la porte. Ce mouvement fait déplacer le bras seulement et il maintient la base du robot stationnaire.
6. Si nécessaire, planifier à nouveau le mouvement coordonné pour achever le processus d'ouverture de la porte, mais cette fois en poussant la porte.

Le système de planification de la porte a été mis en œuvre sur le robot manipulateur mobile PR2. Il a été testé sur deux types de portes en tirant et en poussant. Le robot se dirige vers la porte et utilise son bras pour l'ouvrir légèrement. Le système de planification prévoit un chemin pour la base mobile pour ouvrir la porte. Le plan a, ensuite, été filtré et exécutée à l'aide des contrôleurs de trajectoires pour la base et le bras. Le robot a réussi l'ouverture d'une variété de portes à la fois en tirant (Fig. 31), ou en poussant (Fig. 32).

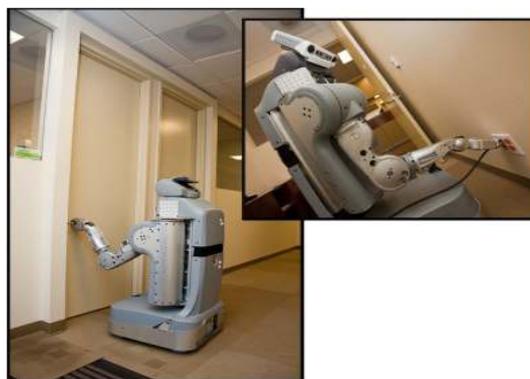


**Figure 31** :Tirer sur la porte pour l'ouvrir



**Figure 32** : Poussez la porte pour l'ouvrir

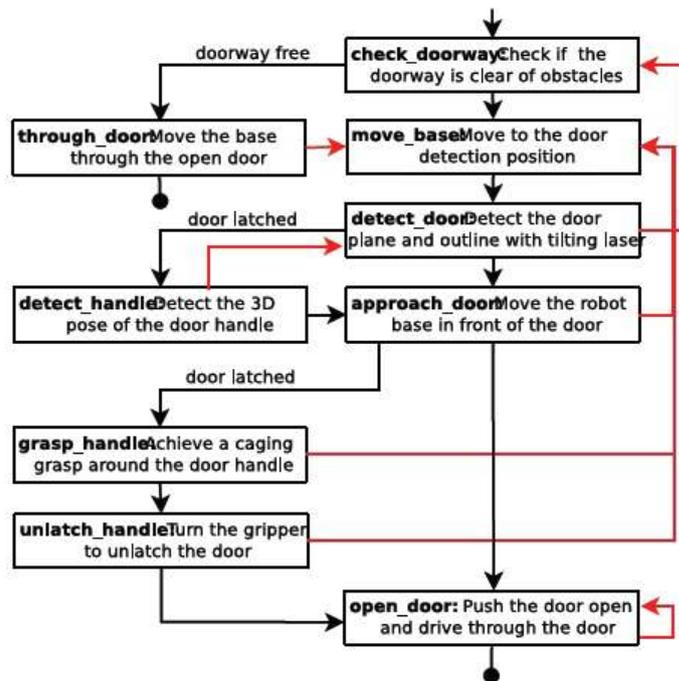
Wim Meeussen et ses collègues de Willow Garage [23] présentent un système robotique autonome capable de naviguer, on ouvrant les portes à tout état, y compris verrouillée, entre-ouverte, partiellement ouverte, et entièrement ouverte, et se branchant sur une prise électrique pour recharger si nécessaire, par l'intégration de contrôle conforme en mouvements de manipulation et des algorithmes de détection robustes pour les portes, les poignées de porte et les prises électriques, en combinant une approche basée sur la perception laser(présentée en détail dans [17]) et une approche basée sur l'image pour estimer rigoureusement l'emplacement de la poignée sans nécessité d'une haute précision du calibrage pince /caméra.



**Figure 33** :Le robot autonome PR2 branche une prise électrique et ouvre une porte

Wim Meeussen et ses collègues ont accompli la tâche d'ouverture de porte par action primitive qui combine des données des capteurs avec le contrôle conforme de manipulateur pour réaliser la détection réussite et la manipulation des portes. La figure 34 montre la machine d'état des différentes actions de système d'ouverture.

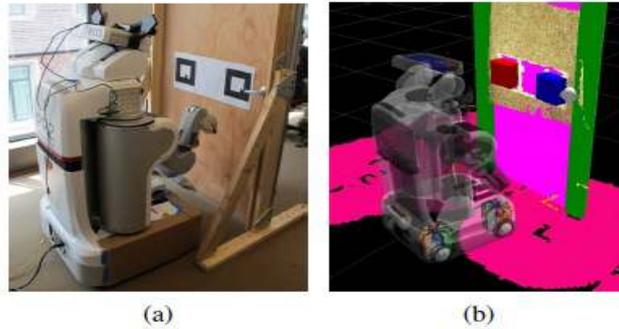
Les flèches noires indiquent les transitions d'état entre les actions, les flèches rouges montrent le recouvrement des erreurs. Ils ont effectuées leurs expériences sur le même robot PR2. Pour accélérer les tâches de vision par ordinateur, ils utilisent intensivement la bibliothèque de vision par ordinateur Open Source OpenCV4 [24].



**Figure 34 :** La machine d'état utilisé par la tâche d'ouverture

Steven Gray et ses collègues [25] développent récemment une approche d'ouverture de porte, en poussant et tirant ouverte, à la fois, les portes à ressort et non ressort, sur le même robot PR2. La tâche d'ouverture de porte se compose de deux étapes principales : la détection de la porte et la détermination de ses paramètres, suivie par la planification et l'exécution du mouvement d'ouverture de la porte. Pour la détermination de la porte et la taille des poignées et des positions, sont déterminés en utilisant une des caméras stéréo, qui fournit un cadre pour le suivi des marqueurs ARToolKit fiduciaires. Chaque porte est

mesurée à l'avance et de plusieurs propriétés de la porte sont enregistrées, y compris la distance entre chaque repère sur le bord de la porte, la distance par rapport à la poignée de porte, la profondeur de la poignée de la porte, du côté de la charnière, la direction d'oscillation et la force nécessaire à la poignée pour ouvrir la porte [25].



**Figure 35** :Détection de la porte avec l'une des caméras stéréo par ARToolKit  
(a) visualisation de la porte (b) la carte côtière 2D est gonflée en rose.

## 2.5. Conclusion

Dans ce chapitre, nous avons proposé ARToolKit pour la détection visuelle de poignée de la porte avec une seule caméra ainsi que les différentes étapes de la tâche d'ouverture d'une porte.

Nous avons aussi présenté quelques approches pour réaliser cette tâche qui reste toujours une idée difficile à réaliser, notamment pour la détection des portes et les poignées de portes. Certaines approches utilisent le balayage laser, d'autres utilisent la stéréovision, etc. Les meilleures approches sont celles qui combinent les deux (plus de précision).

Si les incertitudes sont faibles, la réussite de la réalisation de la tâche est très élevée. Sinon, avec des incertitudes plus importantes dans la mesure de la position de la poignée, on engendrerait un taux de réussite faible. La source de ces incertitudes est très variée. On peut citer, entre autres, un modèle imparfait, des erreurs de calibration du système de vision, etc.

Le chapitre suivant présente la calibration du système de vision sur lequel se base l'approche que nous avons proposée pour l'ajustement visuel et automatique de la pince du manipulateur mobile sur la poignée afin d'accomplir la tâche d'ouverture d'une porte.



# Chapitre 3 Calibration du système de vision

---

## 3.1. Introduction

Avant toute chose, pour qu'un système soit opérationnel, il faut passer par une phase de calibration. Ce chapitre décrit les deux points essentiels de la calibration. Tout d'abord, déterminer la matrice de passage homogène entre les coordonnées 2D images et celles 3D réelles d'un objet quelconque de l'environnement (dans ce cas, c'est la poignée d'une porte). Pour déterminer la relation 2D/3D, nous devons faire une modélisation mathématique de la caméra suivie par un calibrage.

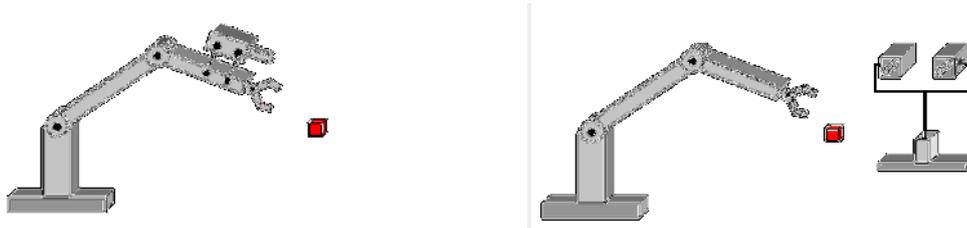
La modélisation mathématique des caméras consiste à obtenir les paramètres correspondantes à la transformation directe et inverse effectuée pour passer des points d'un objet 3D dans l'espace aux points de sa projection 2D dans le plan image et vice-versa. Pour cela, nous avons utilisé la bibliothèque ARToolKit (Augmented Reality Toolkit).

La bibliothèque ARToolKit est un software utile pour la création des applications de Réalité Augmentée (RA). ARToolKit utilise un système de marqueurs « Pattern ». Un marqueur est un motif simple sur fond blanc entouré d'un cadre noir. L'algorithme doit, tout d'abord, détecter, puis, identifier ce marqueur. Il en déduit, en temps-réel, sa position et son orientation par rapport à la caméra.

Pour pouvoir commander un robot à l'aide d'une caméra qui lui est rigidement attachée, il est nécessaire de connaître la transformation euclidienne fixe qui les sépare, c.-à-d. la transformation pince/caméra. L'étalonnage (ou calibrage) pince/caméra consiste à retrouver le lien de la pince du robot à la caméra. Il se fait, en général, en comparant les mouvements de la pince et ceux de la caméra.

## 3.2. Configurations caméra/robot

L'utilisation d'une caméra dans l'asservissement visuel de robot peut s'effectuer avec deux configurations possibles [26] :



**Figure 36** : Caméra fixée au robot « *eye-in-hand* » et caméra indépendante du robot « *eye-to-hand* »

- **Caméra fixée au robot « *eye-in-hand* »** : Souvent appelée « œil dans la main » (« *eye-in-hand* » en anglais). Ce système consiste en une caméra montée directement sur l'effecteur du robot (Fig. 36 à gauche). La relation est souvent constante entre la position de la caméra et la position de l'effecteur. La configuration d'œil dans la main à une vue partielle mais précise.
- **Caméra indépendante du robot « *eye-to-hand* »** : La deuxième configuration « l'œil à la main » consiste en une (ou plusieurs) caméra(s) fixe(s) et située(s) dans l'espace de travail (vue globale), sans liaison mécanique avec le robot (Fig. 36 à droite). Il existe une relation (transformation de coordonnées) entre la base de système de coordonnées de robot et celle des caméras. L'image de la cible sur la(les) caméra(s) se trouve alors indépendante de mouvement de robot.

### 3.3. Modélisation d'une caméra

La modélisation d'une caméra est basée sur la définition d'un ensemble de paramètres qui se rapproche le plus possible du comportement physique du capteur à l'aide d'un modèle géométrique, connu comme étant le modèle sténopé (en anglais « *pin-hole* »). Le modèle géométrique sténopé est constitué d'un plan. Il s'agit d'une modélisation simple et linéaire du processus de formation des images au sein d'une caméra (Fig. 37).

Il existe deux ensembles de paramètres. Tout d'abord, il y a l'ensemble des paramètres intrinsèques qui modélisent la géométrie interne et les caractéristiques optiques du capteur. Le second ensemble contient les paramètres extrinsèques qui lient le repère du

capteur au repère associé à la scène où se trouvent les objets de référence utiles pour la calibration [27]. L'ensemble des paramètres intrinsèques contient :

- la distance focale  $f$ , qui correspond à la distance (en mm) séparant le plan image et le point focal  $O_c$ .
- les paramètres de conversion  $k_u$  et  $k_v$ , qui représentent l'ajustement horizontal et vertical permettant de passer d'un repère du plan image (exprimé en mm) au repère image (exprimé en pixel).
- la position  $(u_0, v_0)$  du point principal (le centre de l'image) définie comme étant le projeté du centre optique  $O_c$  sur le plan image, exprimée en pixel.

Aussi, les paramètres extrinsèques contiennent la position et l'orientation du repère de la caméra par rapport au repère du monde. La position est donnée par un vecteur de translation 3x1 nommé  $T$  et une matrice rotation 3x3 nommée  $R$ .

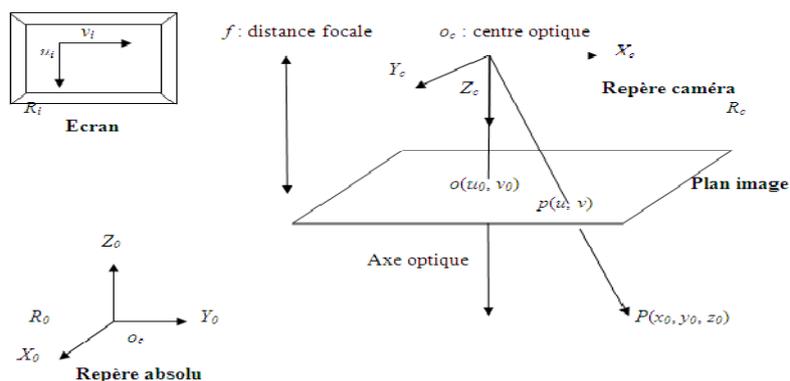


Figure 37 : Modèle sténopé de la caméra

### 3.4. Différentes approches de la calibration

Le calibrage de la caméra est le processus de détermination des caractéristiques géométriques et optiques internes des caméras (paramètres intrinsèques) et/ou la position 3D et l'orientation relative à un certain monde système du même rang (paramètres extrinsèques) pour :

- *Inférer l'information 3D des coordonnées image* : Il y a deux types d'information 3D à inférer. La première est l'information concernant la location de l'objet, la cible ou le dispositif. La deuxième est l'information concernant la position et l'orientation

de la caméra mobile (par exemple une caméra tenue par un robot) relativement au système de coordonnées de la cible.

- *Inférer les coordonnées 2D d'image de l'information 3D.*

Nous allons illustrer les différentes méthodes de calibrages existantes dans ce qui suit :

- **Calibrage photogrammétrique** : le calibrage photogrammétrique utilise un motif précis (grille, échiquier, etc.) dont la métrique est connue pour calibrer la caméra. Ce calibrage peut alors être réalisé avec une bonne précision. Seulement, le calibrage photogrammétrique ne peut pas détecter de changement d'information au cours du temps.
- **Transformation linéaire directe (DLT)** : cette méthode, conçue par Abdelaziz et Karar [28], est une méthode qui consiste à estimer la matrice de projection en résolvant un système surdéterminé d'équations linéaires, à partir d'un objet de calibrage dont les positions des points sont connus et non coplanaires. Dans cette méthode, on ne prend pas en considération la distorsion de la lentille.
- **Méthode de Tsai** : R. Tsai [29] présente une méthode basée, également, sur le principe du déplacement d'un plan mais avec la particularité de modéliser la calibration d'une caméra par une approche géométrique simplifiée. Cette technique de calibration présente la possibilité d'estimer le coefficient qui exprime la distorsion radiale du système optique.
- **Méthode des multi-plans** : c'est une méthode qui utilise plusieurs vues d'un même plan dans un volume donné [30] [31]. La distance entre la caméra et le plan de calibration est mesurée avec une extrême précision. Le principe consiste à modéliser le volume généré par les positions quelconques du plan dans l'espace. Initié par la méthode des deux plans où chaque point contenu dans les plans est une ligne de visée, cette méthode ne nécessite aucune formulation géométrique.
- **Méthode des points de fuites** : les points de fuite définis par les projections des arêtes d'un cuboïde dans l'image sont utilisés pour cela. Ces points correspondent à trois directions orthogonales dans l'espace tridimensionnel, ce qui fournit des contraintes sur la transformation entre la scène et sa projection dans l'image [32]. La méthode nécessite l'acquisition d'au moins deux images d'un cube de

dimensions connues vu sous deux angles différents pour y étudier chaque face. Sur chaque face du cube figure des lignes parallèles et l'orientation des lignes de chaque face est différente de sorte à obtenir trois points de fuite distincts. L'ensemble des points de fuite répartis sur les deux images permet de définir un orthocentre.

- **Méthodes d'auto-calibrage** : ces méthodes sont qualifiées par leurs auteurs comme un procédé permettant d'extraire les paramètres intrinsèques d'une caméra sans intervention manuelle et sans mire de calibration. L'intérêt majeur de cette technique réside dans l'utilisation de systèmes dits à vision active où sans cesse les paramètres optiques sont modifiés par le zoom ou la mise au point [33]. Thomas Skordas a réalisé un comparatif détaillé sur les trois catégories d'auto-calibration dans [34] :
  - La première méthode décrite est celle basée sur la géométrie épipolaire entre deux caméras. Elle repose sur le calcul de la matrice fondamentale qui intervient entre les deux caméras.
  - La seconde méthode est celle qui cherche à estimer un grand nombre de paramètres issus d'un système d'équations quadratiques.
  - La dernière méthode se base sur la connaissance a priori du mouvement de la caméra. Elle nécessite de faire une série de mouvements (de la caméra) ainsi que d'acquérir de nombreuses images. La connaissance de points caractéristiques permet d'estimer les paramètres intrinsèques de la caméra.
- **Calibrage d'une rotation pure** : dans cette méthode, au moins trois images sont prises du même point dans l'espace avec différentes orientations de la caméra. Le calibrage est calculé à partir d'une analyse de la relation des points entre les images [35].
- **Méthode de Zhang** : la méthode de Zhang estime les paramètres extrinsèques et intrinsèques de la caméra grâce à une grille de points coplanaires. Cette méthode se situe entre les méthodes photogrammétriques et les méthodes d'auto-calibrage car les équations sont calculées à partir des points 2D dans l'image et, donc, à part pour l'optimisation finale non-linéaire, les points 3D n'ont pas besoin d'être connus. Voici les étapes de calibrage définies dans [36] :

- imprimer un schéma et l'attacher à une surface plane.
- prendre quelques images du plan modèle sous différentes orientations en déplaçant soit le plan, soit la caméra.
- détecter les points caractéristiques dans les images.
- estimer les paramètres intrinsèques et extrinsèques.
- estimer les coefficients de la distorsion radiale.
- raffiner tous les paramètres.

## 3.5. ARToolKit

### 3.5.1. Définition

ARToolKit est une bibliothèque qui permet le développement d'applications de RA. Ce sont des applications qui impliquent la superposition des images virtuelles sur le monde réel [37]. Les fonctions de cette bibliothèque permettent de calculer la position réelle de la caméra et son orientation, par rapport à des marqueurs [38], en temps réel. ARToolKit reconnaît plusieurs marqueurs dans la scène, le plus souvent, ils sont constitués d'un motif dans un carré noir. Par exemple, la figure 38 montre un objet 3D permanent sur une carte réelle. Lorsque l'utilisateur bouge la carte, le personnage virtuel se déplace avec elle et apparaît attaché à l'objet réel [39]. ARToolKit inclut certaines caractéristiques [39] :

- suivi de la position et de l'orientation avec une seule caméra.
- des algorithmes de suivi qui utilisent des carrés noirs simples comme marqueurs.
- la capacité d'utiliser n'importe quel marqueur carré avec motif.
- le calibrage de la caméra est simple grâce aux fonctions données.
- coût d'utilisation très bas.
- une bibliothèque multiplateforme (Windows, Linux, Mac OS X, SGI).
- open source avec licence GPL pour un usage non commercial.

Les schéma de principe de la bibliothèque ARToolKit est présenté dans la figure 3. Le principal avantage de la bibliothèque ARToolKit pour les applications robotique est la capacité de reconnaître la position et l'orientation du marqueur en utilisant une seule caméra. La possibilité de superposer un objet sur le marqueur reconnu permet de visualiser l'orientation estimée de l'objet, qui sera manipulée par le robot [39].



**Figure 38** : Schéma de principe d'ARToolKit

L'algorithme de détection d'ARToolKit effectue une suite d'étapes sur l'image capturée du flux vidéo afin de repérer la présence d'un marqueur [40].

- la caméra capture la vidéo du monde réel et l'envoie à l'ordinateur via un périphérique classique tel qu'une webcam, une caméra, etc.
- ARToolKit recense toutes les formes carrées présentes dans l'image issue de la vidéo en cours de traitement.
- pour chaque forme carrée détectée, la position et l'orientation de la caméra sont calculées mathématiquement. La matrice de transformation est de la forme donnée par Fig. 39 [41].
- le motif du marqueur est identifié et comparé avec les motifs présents en mémoire afin de connaître l'augmentation qui lui est associée.
- l'augmentation associée est générée. Celle-ci peut prendre diverses formes, la plus classique étant la création d'un model en trois dimensions.
- cette augmentation est, ensuite, superposée à l'image capturée, positionnée, orientée et mise à l'échelle suivant les données fournies par le marqueur.

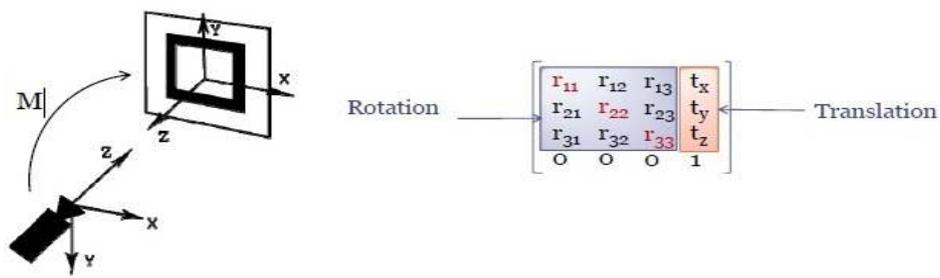


Figure 39 : matrice de transformation d'ArToolKit

### 3.5.2. Reconnaissance de marqueur

Il est intéressant de s'intéresser particulièrement à la méthode de reconnaissance des formes carrées dans l'image et du traitement du motif reconnu afin de chercher une correspondance avec les marqueurs en mémoires. Le fait de comprendre ce fonctionnement permet d'adapter au mieux les paramètres envoyés à ARToolKit et de mesurer l'influence des divers fichiers de configurations que l'utilisateur doit fournir à cette bibliothèque [40].

La première étape fondamentale de cette reconnaissance consiste à binariser l'image reçue afin de passer celle-ci en noir et blanc pures (pas de nuances de gris intermédiaires). Cette binarisation est évaluée selon un paramètre nommé threshold (seuil) correspondant au taux de luminosité nécessaire pour déterminer si un point de l'image sera noir ou blanc [40].

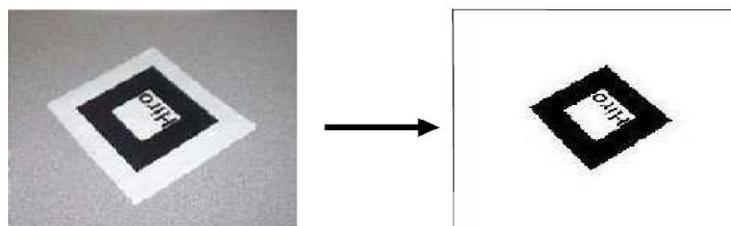
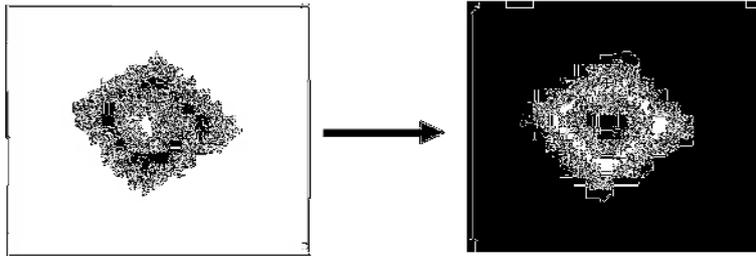


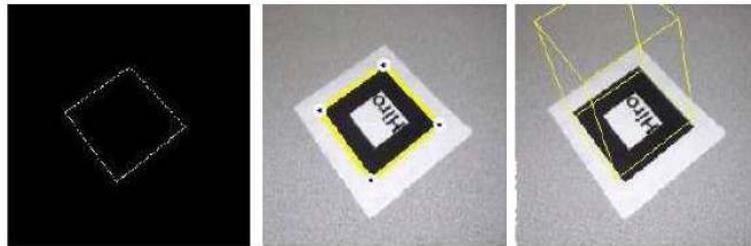
Figure 40 : Binarisation de l'image

L'étape suivante porte sur la recherche, dans l'image binarisée, de composants liés afin de dresser une liste de ces derniers. Étant donné la simplicité des motifs utilisés, ces composants seront des éléments du symbole central au marqueur et pourront, donc, être utile dans les étapes ultérieures (normalisation du marqueur, confrontation aux motifs stockés en mémoire, etc.) [40].



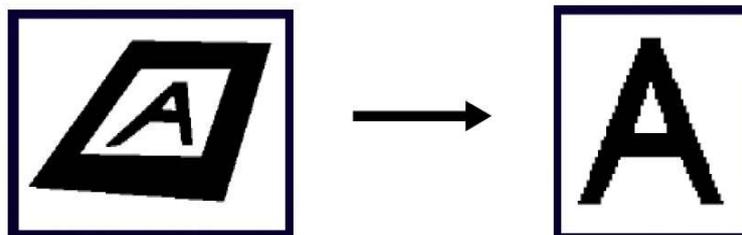
**Figure 41** : Recherche de composants liés

Le fait d'utiliser un motif à contour carré rend relativement aisé la détection des coins et arrêtes de celui-ci. L'algorithme dresse, donc, la liste des contours, lignes et coins du marqueur afin de pouvoir effectuer sa normalisation, nécessaire à la reconnaissance. Il ne faut, en effet, pas oublier que le marqueur n'est pas forcément présenté perpendiculairement à la caméra, il est très probable que celui-ci soit incliné suivant un ou plusieurs des degrés de liberté disponibles. L'image du marqueur inclinée résultante laissera, donc, apparaître un parallélogramme [40].



**Figure 42** : Recensement des arrêtes et coins du marqueur

La normalisation du marqueur fait appel au fichier de configuration de la caméra utilisée (fichier .dat), car il permet de prendre en compte les distorsions causées par la lentille de l'appareil. Ces paramètres sont, donc, propres à chaque matériel même si un fichier de configuration générique peut être utilisé [40].



**Figure 43** : Principe de normalisation

L'étape finale constitue la confrontation du motif détecté avec ceux stockés en mémoire afin de procéder au traitement adéquat par la suite. Toute détection de marqueurs suit rigoureusement le schéma donné par la figure 9 [40].

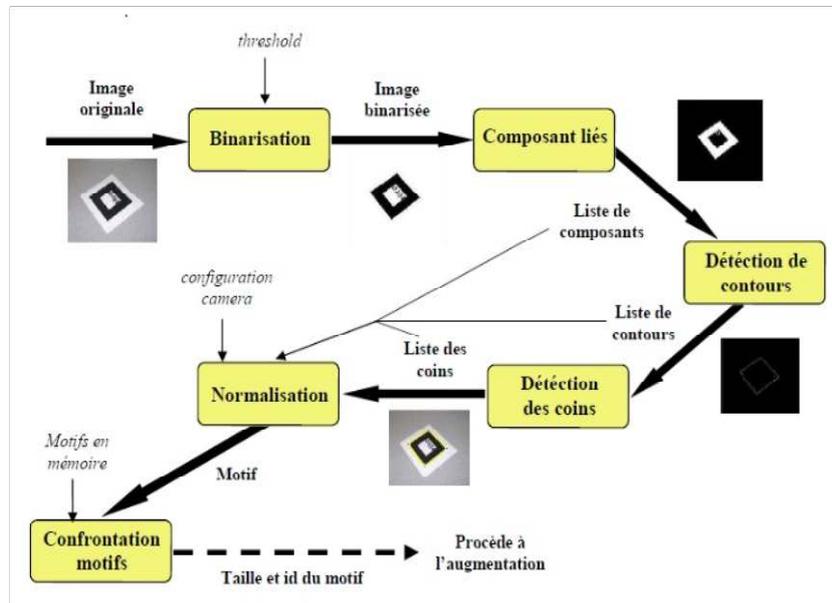


Figure 44 : Principe de reconnaissance des marqueurs

Le motif redressé est, par la suite, comparé au pattern chargé dans le programme. Ces derniers sont un fichier texte où chaque pixel est représenté par sa valeur en niveaux de gris (entre 0 et 255). Les degrés de similitude ou « confiance » sont, ensuite, calculés par rapport à chaque pattern dans quatre sens différents, et le plus haut est retenu. Cela permet de déterminer et générer l'augmentation associée.

```

255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 247
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 248
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 250
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 252
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255
249 244 251 254 255 251 253 254 252 254 254 255 249 254 251 246
172 237 181 165 206 181 204 178 245 248 152 231 194 216 169 215
182 233 215 202 225 193 181 178 241 206 205 206 175 217 193 181
251 244 254 255 251 248 250 255 243 247 255 247 230 250 254 247
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 247
255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 248

```

Figure 45 : Format de marqueur au niveau de gris

## 3.6. Calibration pince/caméra

### 3.6.1. Définition

Le problème du calibrage pince/caméra consiste à déterminer la transformation rigide (translation et rotation) entre le repère de la caméra et celui de la pince. Pour faire apparaître cette transformation, on peut considérer le déplacement d'un système constitué d'une pince et d'une caméra fixée l'une à l'autre (Fig. 46).

Lorsque la pince se déplace selon la transformation rigide  $B = (R_B, \vec{t}_B)$ , la caméra se déplace, elle, selon la transformation rigide  $A = (R_A, \vec{t}_A)$ . A et B sont liées par la relation :

$$AX=XB \quad (1)$$

C'est cette équation matricielle qu'il faut maintenant résoudre après avoir obtenu B grâce au modèle cinématique du robot et A par des calculs de pose ou par reconstruction tridimensionnelle [42].

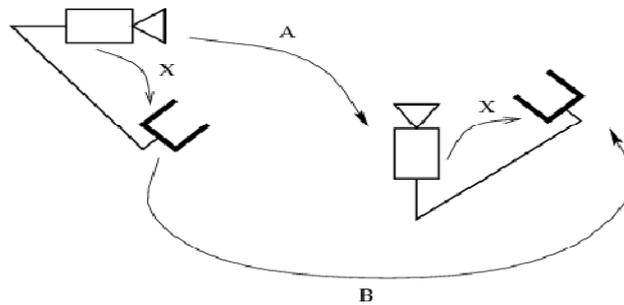


Figure 46 : déplacement d'un système pince caméra

### 3.6.2. Méthodes de calibration pince/caméra

#### a. Axe/Angle

Cette méthode résout le problème de calibrage en deux étapes [29]. Tout d'abord, on détermine l'axe  $n$  et l'angle  $\vartheta$  de la rotation pince/caméra à l'aide des équations (2) et (3) où  $\vec{n}_A$  et  $\vec{n}_B$  sont les axes de rotation du mouvement effectué par la caméra et le robot respectivement, et  $As(.)$  est la matrice antisymétrique associée au produit vectoriel.

$$As(\vec{n}_A + \vec{n}_B)\vec{n} = \vec{n}_A - \vec{n}_B \dots \quad (2)$$

$$\theta = 2 \tan^{-1} \|\vec{n}\| \quad (3)$$

Puis, on construit la matrice de rotation  $R$  associée à  $\vec{n}$  et  $\vartheta$  pour trouver la translation pince/caméra  $t$  par la méthode des moindres carrés appliquée au système :

$$(RB-I) t = RtA... \quad (4)$$

Cette méthode est simple à mettre en œuvre mais elle repose sur l'hypothèse que les mouvements de rotation sont suffisamment amples pour pouvoir en extraire les axes. De plus, les résultats de la résolution aux moindres carrés sont sensibles aux mesures aberrantes.

Des méthodes similaires sont proposées avec des variations sur le thème de l'estimation de la rotation pince/caméra. Dans [42], on trouve la rotation en calculant une solution générale pour chaque mouvement qui dépend alors d'un seul angle. La rotation pince/caméra est donc la rotation qui est solution pour tous les mouvements et obtenue en trouvant les angles permettant d'unifier toutes ces solutions particulières. Dans [43], la rotation est trouvée par la méthode des moindres carrés linéaires en se basant sur les groupes de Li. La solution proposée n'est pas seulement simple et efficace mais elle élimine aussi la dépendance au choix du facteur de longueur pour l'espace physique. La condition de l'existence et de l'unicité des solutions du système  $AX=XB$  peut être fixée de manière compacte. Ainsi, Li et Betsis [44] trouve la rotation pince/caméra en ramenant les rotations de la pince et celles de la caméra à des rotations canonique de la forme :

$$\begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}... \quad (5)$$

## **b. Quaternions duaux unitaires**

Cette méthode est basée sur la représentation des déplacements rigides par des quaternions duaux unitaires [45]. On obtient, alors, deux équations dont la solution détermine la transformation pince/caméra. Cette méthode permet de calculer simultanément la rotation et la translation pince/caméra. Cependant, elle repose, elle aussi, sur l'hypothèse que les rotations soient conséquentes. En outre, l'application de quelques contraintes nécessite de trouver une solution réelle à une équation polynomiale de second degré dont il n'est pas prouvé que le déterminant soit positif. Partant d'une formulation équivalente, [46] proposent une solution analytique pour les mesures parfaites et se

rabattent sur une solution itérative par descente de gradient lorsque les mesures sont erronées.

### **c. Minimisation non linéaire**

Cette méthode est présentée dans [47]. Elle réalise une minimisation non-linéaire selon l'algorithme de Levenberg-Marquardt d'une fonction de coût qui utilise la représentation des rotations par des quaternions unitaires. Elle nécessite une initialisation proche de la solution et des rotations de grande amplitude, mais elle permet une grande précision de résolution.

### **d. Méthode de Zhuang**

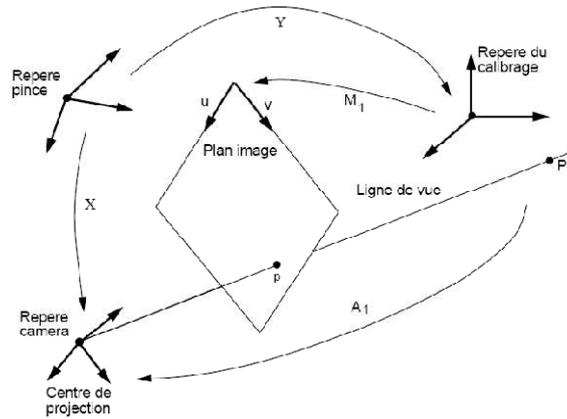
Dans [48], un « Identificateur Jacobian » est défini. C'est une matrice Jacobienne de dimension 12x6 lorsque deux calculs de pose sont utilisés. Elle joue un rôle important dans l'identification de la matrice de transformation  $X$ . Les conditions d'observabilité des paramètres d'erreur de calcul de pose sont dérivées de la matrice Jacobienne.

### **e. Méthode de Rémy**

Cette méthode proposée dans [49], permet, à partir de l'observation d'une mire particulière, d'obtenir simultanément la transformation pince/caméra et la position de la mire dans le repère de base du robot. Elle ne nécessite pas de calcul de déplacement de la caméra. En effet, elle minimise l'erreur de reprojection des points de la mire en fonction des deux transformations inconnues. Par contre, pour obtenir l'information nécessaire à la résolution du système, il faut tout de même déplacer la caméra.

### **f. Formulation de Dornaika**

L'intérêt de cette méthode, présentée dans [50], est de se libérer de la connaissance explicite des paramètres intrinsèques et de remplacer l'estimation de la transformation repère caméra/repère pince par l'estimation directe de la transformation entre le repère du calibrage et le repère de la pince (Fig. 47).



**Figure 47** : Nouvelle formulation de calibrage pince/caméra

Le système à résoudre devient alors (avec  $i=1\dots n$ ) :

$$\begin{cases} M2Y = M1YB12 \\ \cdot \\ MiY = M1YB1i \dots \\ \cdot \\ MnY = M1YB1n \end{cases} \quad (6)$$

L'avantage de cette nouvelle formulation est que les erreurs éventuelles, associées à la décomposition des matrices de projection en paramètres intrinsèques et extrinsèques, seront éliminées. Ainsi, les résultats de cette formulation sont plus précis que ceux de la formulation classique.

### g. Méthode linéaire avec calcul de pose

Dans cette méthode, à partir d'images d'une mire de calibration, les poses successives de la caméra sont calculées. Puis, à l'aide des mouvements préalablement enregistrés du robot, la transformation pince/caméra est calculée. Cette méthode est inspirée de la ressemblance du problème d'étalonnage pince/caméra avec l'équation matricielle dite de Sylvester. Elle permet une analyse algébrique très simple du problème, et elle donne de meilleurs résultats en rotation que les autres méthodes.

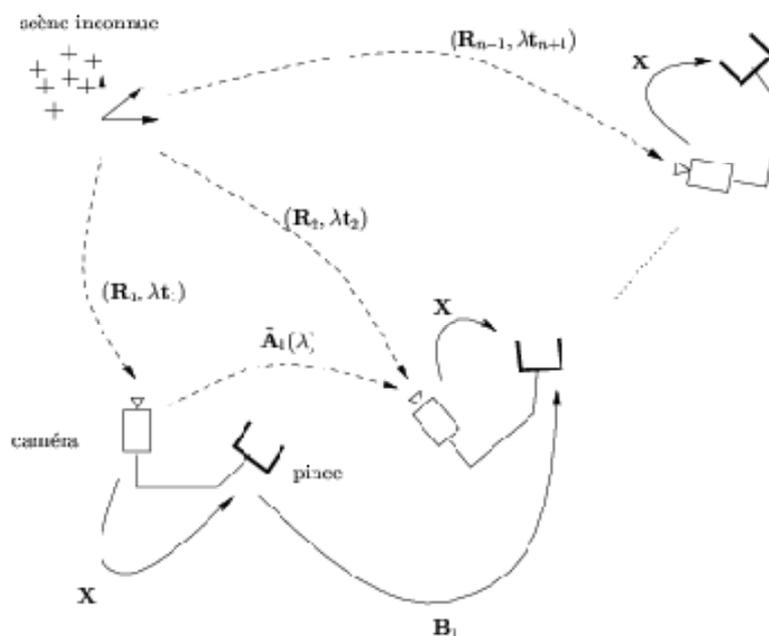
### h. Auto-étalonnage

L'auto-étalonnage pince/caméra est un effort d'augmenter l'autonomie du robot en relâchant les contraintes pratiques de l'étalonnage caméra. Dans un calibrage classique, les

informations sont données, non seulement, par les déplacements du robot, mais, également, par le modèle 3D de la mire observée.

Dans l'approche d'auto-calibrage, ces informations sont uniquement données par les déplacements du robot.

Dans [40], les auteurs ont proposé une méthode de calibrage en conjonction avec la reconstruction euclidienne. Dans [50], la méthode proposée est basée sur une reconstruction projective. Elle estime, simultanément, le passage d'une structure projective à une structure euclidienne, les paramètres intrinsèques de la caméra et la transformation pince/caméra. Les déplacements du robot sont moins précis que les mesures 3D a priori d'une mire de calibrage. Néanmoins, ce manque de précision peut être levé par l'utilisation d'un robot précis. [52] propose une méthode d'auto-étalonnage simultané d'un robot et du lien pince/caméra similaire à [48] à l'exception qu'il représente les rotations par des quaternions. Dans [53], deux types d'auto-étalonnage pince/caméra sont identifiés. Le premier consiste à déplacer la caméra devant une scène totalement inconnue (Fig. 48). Le deuxième permet la détermination de la transformation pince/caméra par l'observation d'un faisceau de droites.



**Figure 48** : la procédure d'étalonnage pince/caméra dans le cas d'une scène inconnue

## 1.7. Conclusion

Dans ce chapitre, nous avons donné les différentes configurations de la caméra avec la pince du robot. Les systèmes avec caméra embarquée sur le robot simplifient évidemment la tâche de positionnement. Avant le calibrage de la caméra, nous avons parlé de la modélisation de la caméra et, plus particulièrement, le modèle sténopé pour définir ses paramètres intrinsèques et extrinsèques.

Le calibrage est une étape nécessaire dans la vision tridimensionnelle. Pour cela, nous avons étudié différents travaux de la littérature utilisant le calibrage dans la vision robotique. Dans le deuxième point du chapitre, nous avons décrit la bibliothèque ARToolKit et son principe de fonctionnement. À cause de sa facilité d'utilisation, cette bibliothèque sera utilisée dans ce travail pour le calibrage automatique de la caméra. Le chapitre se termine par le problème de calibrage pince/caméra pour déterminer la relation entre le repère de la pince et celui de la caméra.

L'objectif du chapitre suivant est de décrire le système robotique expérimental. Il donne, aussi, les résultats de la calibration de la caméra et pince/caméra, les coordonnées 3D et le modèle géométrique.

# Chapitre 4 mis en œuvre

---

## 4.1. Introduction

La mise en place du robot manipulateur mobile RobuTER/ULM nécessite plusieurs ressources matérielles et logicielles. Il est évident que le robot et ses accessoires représentent la partie essentielle dans l'architecture du système robotique, c'est à travers ces accessoires qu'on va pouvoir agir sur l'environnement.

Ce système robotique est, en fait, un assemblage de pièces mécaniques qui nécessitent des logicielles de pilotage pour les mettre en service d'où le développement d'une application logicielle, pour la gestion du matériel permettant, ainsi, à l'opérateur d'accomplir les tâches demandées.

Ce chapitre est consacré à la description et à la modélisation du système robotique disponible au niveau de la *Division Productique et Robotique (DPR)* du *Centre de Développement des Technologies Avancées (CDTA)*.

## 4.2. Architecture du système robotique

Le système robotique expérimental, donné par Fig. 49, consiste en deux sites connectés par un système de communication sans fil :

- *Site local (opérateur)* : il consiste en un PC hôte (off-board) sous Windows XP et qui est dédié au contrôle distant du robot.
- *Site distant (téléopéré)* : il consiste en le manipulateur mobile RobuTER/ULM qui est contrôlé par un PC embarqué (on-board) sous Linux Redhat 6.2.

### 1.2.1. Architecture globale du système robotique

L'échange d'informations se fait par voie réseau entre un PC sous Linux et un autre sous Windows. Le traitement et l'analyse des images nécessitent un ordinateur performant pour que les calculs soient effectués en temps-réel. Pour cela, on utilise une caméra IP qui transmet le signal vidéo par réseaux WIFI vers un PC hôte très performant (client).

L'opérateur entre en communication indirecte avec le robot, par l'intermédiaire du PC hôte qui représente le Client (vis-à-vis de la communication client/serveur). Grâce à une interface de communication Homme/machine, il agit sur le robot et décide comment ce dernier doit se déplacer et quelle tâche doit-il effectuer, tout en ayant la possibilité de recevoir des informations sur son état afin de contrôler ses mouvements. Grâce à un système imageur qui a pour rôle l'acquisition des images concernant le robot et son environnement [54].

Le PC embarqué joue le rôle d'un serveur. Ce dernier reçoit les consignes qui sont envoyées par l'opérateur via le réseau de communication sans fil, il élabore les lois de commande convenables pour le robot, il assure leur exécution, et, enfin, il reçoit les informations de retour venant des capteurs.

Pour répondre à tous ces besoins, et par rapport aux problématiques posées et aux matériels disponibles, nous avons adopté l'architecture suivante :



Figure 49 : Architecture du système robotique

#### **a. PC hôte (client)**

La commande est assurée par un ordinateur portable qui représente le *Client* sous lequel est installé le système d'exploitation Windows. Ce PC fait la liaison entre l'opérateur et le robot distant :

- Il permet la gestion de communication depuis et vers le robot, par l'envoi des consignes de l'utilisateur et la réception des informations capteurs.
- La réception, l'affichage et le traitement des images vidéo issues du système imageur.
- La réception et l'affichage des différentes informations issues des capteurs équipant le robot.
- Le contrôle/commande du robot via une interface graphique.

#### **b. Interface Homme/Machine**

L'Interface Homme/Machine est un élément essentiel qui offre la possibilité de communiquer avec le robot distant. Cette interface assure les fonctionnalités suivantes :

- La traduction des besoins de l'opérateur en consignes pour le robot.
- L'affichage des images vidéo issues de la caméra IP du robot et son environnement qui offre la possibilité de voir l'exécution des commandes en temps réel.
- L'affichage des différentes caractéristiques dynamiques du robot (vitesse, orientation, position, etc.).
- La prise en charge des consignes provenant de l'opérateur.

### c. PC embarqué (serveur)

Le robot est doté d'un PC embarqué (Serveur) qui tourne sous Linux Redhat 6.2. Ce module accomplit les tâches suivantes :

- Recevoir les commandes envoyées par l'interface dans le PC Client et les interpréter pour que le robot puisse exécuter les opérations désirées par l'opérateur à l'aide d'un programme en c/c++.
- Recevoir les informations sur l'environnement du robot, les interpréter et les envoyer à l'opérateur.



**Figure 50** : PC embarqué.

### d. Système imageur

La vision artificielle par ordinateur est la science qui développe les bases algorithmiques et théoriques par lesquelles l'information utile relative à l'environnement peut être automatiquement extraite et analysée à partir d'une image, d'un ensemble d'images ou d'une séquence d'images [55]. En robotique, la vision artificielle par ordinateur consiste à créer un *œil* pour le robot afin qu'il puisse :

- se localiser dans son environnement.

- localiser un objet dans une scène par rapport à un référentiel donné (reconnaître la position et l'orientation d'un objet observé).
- identifier les objets de la scène.

La caméra IP (*AXIS 213*), est fixée sur la pince du bras manipulateur. Elle est suffisamment manœuvrable grâce au six degrés de liberté du manipulateur [03] afin d'explorer tout l'environnement du robot. La puissance du PC embarqué ne permet pas d'effectuer les traitements nécessaires des images acquises à son niveau à cause des capacités moyennes de ce PC et les performances que demande le traitement d'images. À cet effet, les images sont directement transmises, via le WIFI, vers le PC hôte (client) pour y subir les traitements nécessaires. Les images vidéo capturées sont des images en couleur d'une résolution de 704x576 pixels, jusqu'à 25 images par seconde (Fig. 51).



**Figure 51** : Caméra IP Axis 213

#### **e. Architecture détaillée du manipulateur mobile *RobuTER/ULM***

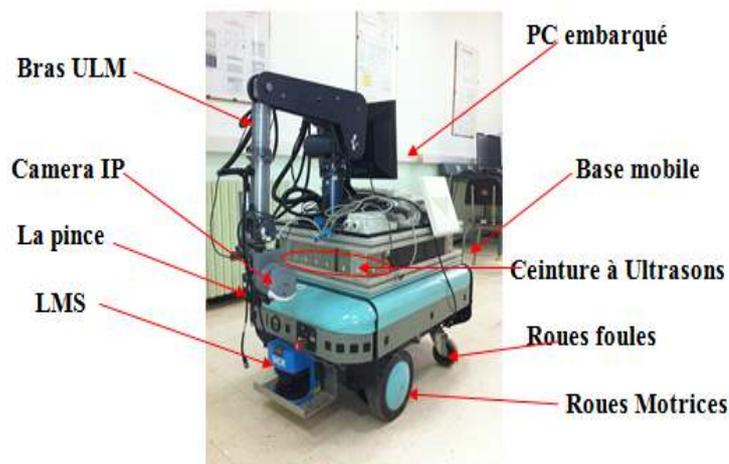
Le robot *RobuTER/ULM* est constitué, comme le montre la figure 52 ci-dessous, d'une base mobile rectangulaire à quatre roues, sur laquelle est installé, de façon rigide, un bras manipulateur ultra léger à six degrés de liberté avec une pince électrique à deux doigts. Toutes les liaisons du manipulateur sont des liaisons rotoïdes. Le tout est piloté par un PC MMX industriel (on-board) embarqué qui tourne sous *Linux* 6.2 et un système de commande à base de quatre cartes microcontrôleurs *MPC555* [56].

Les cartes contrôlent l'ensemble des actionneurs et des capteurs. La première carte *MPC555* contrôle la base mobile. La deuxième et la troisième carte contrôlent, respectivement, les trois premières et les trois dernières articulations du manipulateur. La dernière carte *MPC555* contrôle les capteurs d'effort des six axes.

La base mobile différentielle non-holonome *RobuTER* dispose de quatre roues. Deux roues motrices assurent sa mobilité et deux roues folles qui maintiennent la stabilité et l'équilibre de l'ensemble.

La base mobile est équipée d'un système sensoriel composé d'un ensemble de capteurs proprioceptifs et extéroceptifs. La base mobile contient un capteur de charge de batteries, un capteur odométrique (d'une résolution de 500 points/tour) sur chacune des roues motrices. Pour les besoins de la navigation et l'évitement d'obstacles, le robot dispose de deux types de capteurs de distance et de proximité qui sont une ceinture de 24 capteurs ultrasonores (d'une portée maximale de 3m) et d'un télémètre laser (LMS) à l'avant de la base mobile (d'une résolution de 0.25°, 0.5° et 1°) ayant une portée maximale de 8m.

Le bras manipulateur ULM dispose des capteurs proprioceptifs tels qu'un capteur de position des axes et un capteur de fin de course sur chaque articulation, ainsi qu'un capteur d'effort six axes.



**Figure52:** RobuTER/ULM.

Le robot est alimenté par quatre batteries embarquées à 48v (12v chacune). Ces batteries alimentent la base mobile, le manipulateur ainsi que tous les capteurs installés [55].



**Figure 53 :** Carte MPC555

### 3. Modélisation du système robotique

Le travail à effectuer dans ce mémoire consiste à localiser la poignée de la porte, se trouvant devant le robot. Cela implique qu'il faut positionner la base mobile devant la porte et le bras manipulateur sur la poignée.

Le robot se trouve dans une position bien définie dans le repère de référence, dont le bras manipulateur doit être dans une position connue pour que la caméra puisse capter le marqueur, il va manipuler la poignée qui doit être bien défini dans le repère du bras manipulateur.

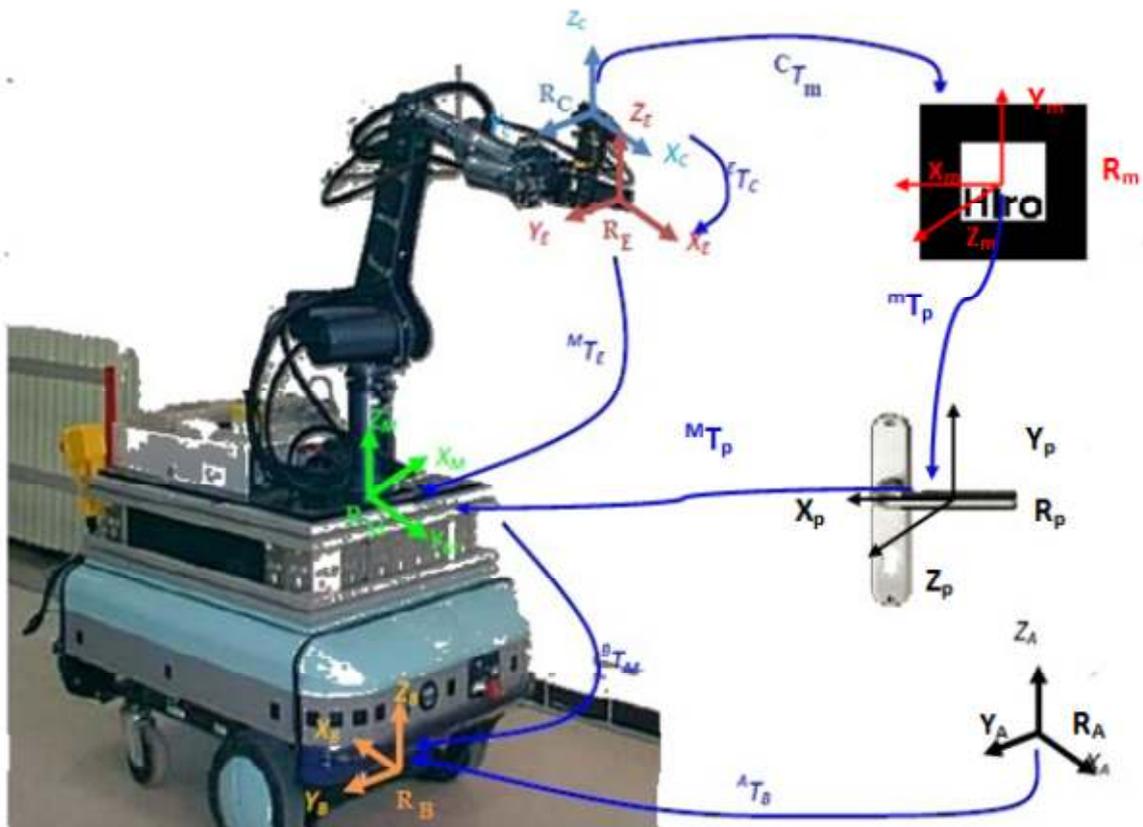
Pour pouvoir saisir la poignée, il faut d'abord la localiser dans le repère 3D, c.-à-d. connaître ses coordonnées cartésiennes dans un repère de référence fixe (repère lié à la scène), et connaître ses coordonnées dans le repère lié au robot. Cela ne peut se faire que si on connaît avec précision les différentes transformations des repères mis en jeu (Fig. 54) :

- $R_A=(O_A, X_A, Y_A, Z_A)$  : le repère absolu lié à l'environnement.
- $R_B=(O_B, X_B, Y_B, Z_B)$  : le repère attaché à la base mobile.
- $R_M=(O_M, X_M, Y_M, Z_M)$  : le repère placé à la base du manipulateur.
- $R_E=(O_E, X_E, Y_E, Z_E)$  : le repère fixé à l'effecteur.
- $R_K=(O_k, X_k, Y_k, Z_k)$  : ce repère est lié à l'articulation  $k$  ( $k=1...6$ ). Ces repères se succèdent jusqu'à atteindre celui de l'effecteur.
- $R_C=(O_C, X_C, Y_C, Z_C)$  : le repère attaché à la caméra.
- $R_m=(O_m, X_m, Y_m, Z_m)$  : le repère lié au marqueur qu'utilise ARToolKit.
- $R_p=(O_p, X_p, Y_p, Z_p)$  : le repère lié à la poignée de la porte.

En effet, ces transformations élémentaires sont au nombre de sept :

- ${}^C T_m$  : une transformation du repère marqueur au repère caméra qui permet d'exprimer les coordonnées du marqueur dans le repère caméra en utilisant la bibliothèque ARToolKit.
- ${}^m T_p$  : une transformation du repère poignée au repère marqueur par une simple translation.
- ${}^E T_C$  : une transformation du repère caméra au repère pince.
- ${}^M T_E$  : une transformation du repère pince au repère bras par une modélisation géométrique du bras manipulateur.

- ${}^B T_M$  : une transformation du repère bras manipulateur au repère base mobile.
- ${}^M T_p$  : une transformation du repère poignée au repère bras manipulateur.
- ${}^A T_B$  : une transformation du repère base mobile au repère absolu



**Figure 54** : Transformations de repères nécessaires

Une transformation homogène entre deux repères, un repère fixe et un repère relatif s'exprime, en général, en un déplacement et une rotation. Si on adopte la notation matricielle, cette transformation est composée d'une matrice de rotation notée  $R$  formée par les rotations autour des axes, et d'une matrice de translation notée  $T$  entre les deux repères. Elle peut s'écrire sous la forme suivant :

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ 0 & 1 \end{bmatrix} \quad (1)$$

### 4.3.1. Modélisation géométrique du bras manipulateur ULM

#### a. Modèle géométrique direct (MGD) du manipulateur ULM

La technique la plus répandue pour décrire la géométrie d'un bras manipulateur consiste à utiliser les *paramètres de Denavit-Hartenberg*. Il faut définir au préalable les repères liés aux corps qui composent le bras manipulateur ainsi que les quatre paramètres de Denavit-Hartenberg Modifiés (DHM) présenté dans la figure 7 [57] :

- $\theta_{i-1}$  : l'angle de rotation autour de l'axe  $X_i$  entre les axes  $Z_i$  et  $Z_{i+1}$ .
- $a_{i-1}$  : la distance entre l'axe  $Z_i$  et l'axe  $Z_{i+1}$ , le long de  $X_i$ .
- $d_i$  : la distance entre l'axe  $X_{i-1}$  et l'axe  $X_i$ , le long de  $Z_i$ .
- $\alpha_i$  : l'angle de rotation entre les axes  $X_{i-1}$  et  $X_i$  autour de  $Z_i$ .

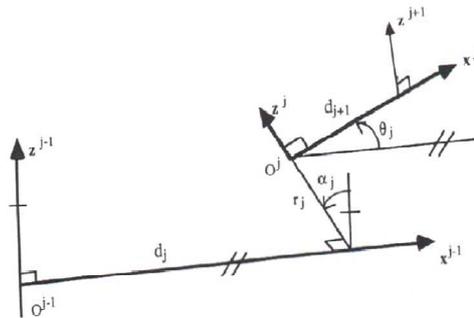


Figure 55 : Paramètres de Denavit-Hartenberg [58]

La matrice de passage homogène d'un repère à un autre des paramètres Denavit-Hartenberg modifiée à la forme suivante [59] :

$$\begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_i \\ \cos(\alpha_i) \sin(\theta_i) & \cos(\alpha_i) \cos(\theta_i) & -\sin(\alpha_i) & -d_i \sin(\alpha_i) \\ \sin(\alpha_i) \sin(\theta_i) & \sin(\alpha_i) \cos(\theta_i) & \cos(\alpha_i) & d_i \cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

L'application de cette approche sur notre bras manipulateur, nous a permis de définir les repères suivants (Fig.56) :

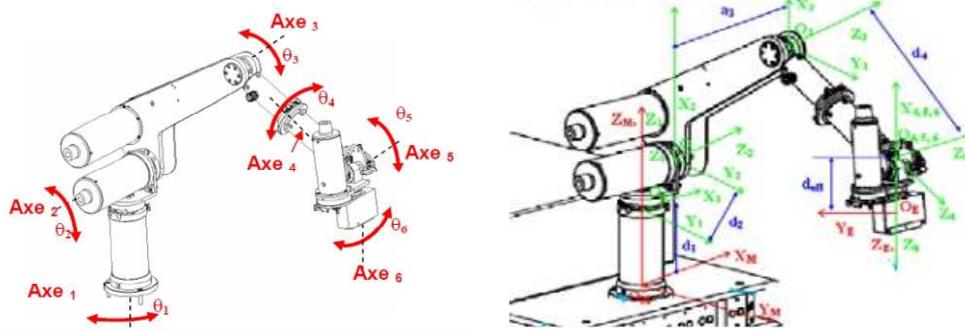


Figure 56 : Paramètres de Denavit-Hartenberg du bras ULM

Après avoir placé les repères et désigner les différents paramètres du système, on peut écrire le tableau des paramètres de Denavit-Hartenberg ci-après :

Articulation	$\alpha_i$ (rad)	$d_i$ (mm)	$\theta_i$	$a_i$ (mm)	$Q_{Min}$ (°)	$Q_{Max}$ (°)
1	0	$d_1=290$	$\theta_1$	0	-95	96
2	90	$d_2=108.49$	$\theta_2$	0	-24	87
3	-90	$d_3=113$	0	$a_3=402$	–	–
4	90	0	$\theta_3$	0	-2	159
5	90	$d_4=389$	$\theta_4$	0	-50	107
6	-90	0	$\theta_5$	0	-73	40
7	90	$d_{eff}=220$	$\theta_6$	0	-91	91

Tableau 2 : les paramètres de déplacements et orientations de chaque articulateur

Cette notation nous permet de représenter les déplacements de l'articulation  $i$  relativement à l'articulation  $i-1$ , grâce à la matrice de transformation d'espace  ${}^{i-1}T_i$  donnée par (2).

$${}^M T_E = {}^1 T_7 = {}^1 T_2 * {}^2 T_3 * {}^3 T_4 * {}^4 T_5 * {}^5 T_6 * {}^6 T_7 \quad (3)$$

La matrice  ${}^M T_E$  de transformation d'espace illustre mathématiquement la situation de l'effecteur relativement au repère lié à la base du bras manipulateur, elle est de la forme :

$${}^M T_E = \begin{bmatrix} r11 & r12 & r13 & tx \\ r21 & r22 & r23 & ty \\ r31 & r32 & r33 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$${}^M T_E = \begin{bmatrix} \cos(\theta) \cos(\psi) - \sin(\rho) \cos(\theta) \sin(\psi) & -\cos(\rho) \sin(\psi) - \sin(\rho) \cos(\theta) \cos(\psi) & \sin(\rho) \sin(\theta) & XE \\ \sin(\theta) \cos(\psi) - \cos(\rho) \cos(\theta) \sin(\psi) & -\sin(\rho) \sin(\psi) + \cos(\rho) \cos(\theta) \cos(\psi) & -\cos(\rho) \sin(\theta) & YE \\ \sin(\theta) \sin(\psi) & \sin(\theta) \cos(\psi) & \cos(\theta) & ZE \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Les positions de l'effecteur seront déduites de la matrice  ${}^M T_E$  telles que :

$$\begin{cases} x_E = tx \\ y_E = ty \\ z_E = tz \end{cases} \quad (6)$$

Pour les rotations, la représentation non-redondante utilisée est celles des angles d'Euler classiques  $(\vartheta_E, \psi_E, \phi_E)$  selon les trois axes précédents [08]. Le calcul de ces orientations est donné par (7) :

$$\begin{aligned} \Theta &= \begin{cases} \text{Atan2}(r_{13}, r_{23}) \\ \text{Atan2}(r_{13}, -r_{23}) = \Theta + 180^\circ \end{cases} \\ \phi &= \text{Atan2}(\sin(\Theta) \cdot r_{13} - \cos(\Theta) \cdot r_{23}, r_{33}) \\ \Psi &= \text{Atan2}(-\cos(\Theta) \cdot r_{12} - \sin(\Theta) \cdot r_{22}, \cos(\Theta) \cdot r_{11} + \sin(\Theta) \cdot r_{21}) \end{aligned} \quad (7)$$

### b. Modèle géométrique inverse (MGI) du manipulateur ULM

Le MGI du bras manipulateur est calculé en utilisant la méthode analytique exposée dans [57]. Les valeurs des variables articulaires sont comme suit tels que  $Eps_0$ ,  $Eps_1$  et  $Eps_3$  sont égaux à  $\pm 1$  :

$$\begin{aligned} S1 &= (X \cdot d_2 - Eps_0 \cdot Y \cdot (X^2 + Y^2 - d_2^2)^{1/2}) / (X_2 + Y_2) \\ C1 &= (-Y \cdot d_2 - Eps_0 \cdot X \cdot (X_2 + Y_2 - d_2^2)^{1/2}) / (X_2 + Y_2) \\ \vartheta_1 &= \text{Atan2}(S1, C1) \end{aligned} \quad (8)$$

En considérant  $\text{Tr}_1 = Eps_0 \sqrt{X^2 + Y^2 + d_2^2}$  et  $\text{Tr}_2 = [Z - d_1]$

$$\begin{aligned} S2 &= (\text{Tr}_4 \cdot \text{Tr}_5 + Eps_1 \cdot \text{Tr}_3 \cdot (\text{Tr}_3^2 + \text{Tr}_4^2 - \text{Tr}_2^5)^{1/2}) / (\text{Tr}_2^3 + \text{Tr}_2^4) \\ C2 &= (\text{Tr}_3 \cdot \text{Tr}_5 - Eps_1 \cdot \text{Tr}_4 \cdot (\text{Tr}_3^2 + \text{Tr}_4^2 - \text{Tr}_2^5)^{1/2}) / (\text{Tr}_2^3 + \text{Tr}_2^4) \\ \vartheta_2 &= \text{Atan2}(S2, C2) \end{aligned} \quad (9)$$

En considérant  $\text{Tr}_7 = \frac{\text{Tr}_7[S_2 - C_2] + \text{Tr}_2[S_2 + C_2] - a_3 - d_3}{d_4}$  on en déduit  $C_3$  et  $S_3$  tels que :

$$\begin{aligned} S3 &= \frac{\text{Tr}_7 + Eps_3 \sqrt{2 - (\text{Tr}_7)^2}}{2} \\ C3 &= \frac{-\text{Tr}_7 - Eps_3 \sqrt{2 + (\text{Tr}_7)^2}}{2} \end{aligned}$$

$$\theta_3 = \text{Atan2}(S3, C3) \quad (10)$$

$$C4 = \frac{\text{Eps3}.m13}{\sqrt{m33^2 + m13^2}}$$

$$S4 = \frac{\text{Eps3}.m33}{\sqrt{m33^2 + m13^2}}$$

$$\vartheta_4 = \text{Atan2}(S4, C4) \quad (11)$$

$$S5 = C4m_{13} + S4m_{33}$$

$$C5 = -m_{23}$$

$$\vartheta_5 = \text{Atan2}(S5, C5) \quad (12)$$

$$S6 = -S4.m_{11} + C4.m_{31}$$

$$C6 = -S4.m_{12} + C4.m_{32}$$

$$\vartheta_6 = \text{Atan2}(S6, C6) \quad (13)$$

#### 4.3.2 Modèle géométrique et cinématique de la base mobile

Le repère  $R_B$  est attaché à la base mobile différentielle et non-holonyme *RobuTER*. En supposant que le robot se déplace sur un plan, ce repère sera représenté, dans le repère  $R_A$ , par les coordonnées cartésiennes de position  $(x_B, y_B, z_B)$  de son origine  $O_B$ , ainsi que par l'angle d'orientation de ses axes  $\vartheta_B$  tel que montré par Fig. 57 [55].



**Figure 57** : Représentation du repère de la base mobile  $R_B$  dans le repère absolu  $R_A$

Durant son mouvement, la base mobile calcule en temps-réel, par odométrie, sa situation telle que montrée par Fig. 58 [55] :

- $P_k, P_{k+1}$  : la situation courante et la situation suivante de la base mobile.
- $\omega_{Lk}, \omega_{Rk}$  : la vitesse angulaire des roues gauche et droite à la situation  $P_k$ .

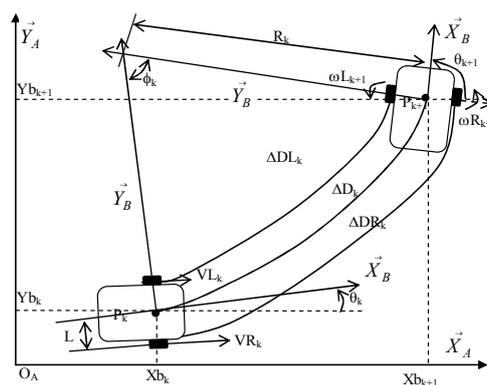
- $\Delta\vartheta_k$ : la rotation élémentaire de la base mobile.
- $V_{Lk}, V_{Rk}$ : la vitesse linéaire de la roue gauche et droite à la situation  $P_k$ .
- $V_k, \omega_k$ : la vitesse linéaire/angularaire de la base mobile à la situation  $P_k$ .
- $\Delta D_{Lk}, \Delta D_{Rk}$ : le déplacement élémentaire de la roue gauche et droite de la base mobile.
- $\Delta D_k$ : le déplacement élémentaire de la base mobile.
- $\phi_k$ : l'angle d'orientation à la situation  $P_k$ .
- $R_k$ : le rayon de rotation de la base mobile à la situation  $P_k$ .
- $2*L$  : la distance entre les deux roues motrices de la base mobile (entre-axes).
- $r$  : le diamètre des roues motrices.

Les déplacements élémentaires droit et gauche du robot  $\Delta D_{Lk}$  et  $\Delta D_{Rk}$  sont donnés comme suit :

$$\begin{pmatrix} \Delta D_{Lk} \\ \Delta D_{Rk} \end{pmatrix} = \begin{pmatrix} R_{k+L} \\ R_{k-L} \end{pmatrix} * \phi_k \quad (14)$$

La rotation élémentaire de la base mobile  $\Delta\vartheta_k$  est donnée par (15) :

$$\Delta\vartheta_k = \frac{\Delta D_{Rk} - \Delta D_{Lk}}{2*L} \quad (15)$$



**Figure 58** : Différents paramètres de mouvement de la base mobile

Le déplacement élémentaire de la base mobile  $\Delta\vartheta_k$  est donné par (16) :

$$\Delta D_k = \frac{\Delta D_{Rk} + \Delta D_{Lk}}{2} \quad (16)$$

Finalement, la situation suivante (position et angle d'orientation) de la base mobile est donnée par (17) :

$$\theta_{k+1} = \theta_k + \Delta\theta_k$$

$$\begin{bmatrix} xb_{K+1} \\ yb_{K+1} \\ zb_{K+1} \end{bmatrix} = \begin{bmatrix} xb_K \\ yb_K \\ zb_K \end{bmatrix} + \Delta D_k * \begin{bmatrix} \cos \theta_{k+1} \\ \sin \theta_{k+1} \\ 0 \end{bmatrix} \quad (17)$$

Concernant le modèle cinématique de la base mobile, les vitesses linéaires de la roue droite et gauche  $V_{RK}$  et  $V_{LK}$  sont données comme suit :

$$V_{RK} = -r * \omega_{RK} = (R_R - L) * \omega_k$$

$$V_{LK} = -r * \omega_{LK} = (R_R - L) * \omega_k \quad (18)$$

De (18),  $\omega_k$  et  $V_k$  sont données, alors comme suit:

$$\omega_k = r * \frac{V_{LK} - V_{RK}}{2 * L}$$

$$V_k = \frac{V_{RK} + V_{LK}}{2} = r * \frac{V_{LK} - V_{RK}}{2} \quad (19)$$

Finalement, le modèle cinématique de la base mobile est donné par :

$$\begin{bmatrix} x_I \\ y_I \\ \theta_I \end{bmatrix} = \begin{bmatrix} \cos \theta_k & 0 \\ \sin \theta_k & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} V_k \\ W_k \end{bmatrix} \quad (20)$$

À partir de ces dernières équations, on peut tirer les conclusions suivantes sur le modèle cinématique de la base mobile [55] :

- Si  $V_{RK} = V_{LK}$ : le rayon de rotation  $R_k$  est égal à l'infini. La trajectoire de la base mobile consiste en une droite avec une vitesse  $V_k = V_{RK} = V_{LK}$  vers l'avant ou vers l'arrière selon le sens de rotation des roues.
- Si  $V_{RK} = -V_{LK}$ : la base mobile fait une rotation autour du point  $P_k$  avec une vitesse de rotation égale à  $\omega_k = \frac{V_{RK}}{L}$ .
- Si  $V_{RK} > V_{LK}$ : la base mobile fait un braquage vers la gauche.
- Si  $V_{RK} < V_{LK}$ : la base mobile fait un braquage vers la droite.
- Si  $\theta_k = \pi/2 + n.\pi$  ( $n \in \mathbb{R}$ ) :  $\tan \theta_k = \infty$ , c.-à-d.  $\frac{y_1}{x_1} = \infty$ , ce qui veut dire que  $x_1 = 0$ .

- Il n'y a pas de mouvement suivant l'axe  $X_A$  mais suivant l'axe  $Y_A$  uniquement.
- -Si  $\vartheta_k = n.\pi (n \in R) : \cotg\vartheta_k = \infty$ , c.-à-d.  $\frac{Y_1}{X_1} = \infty$ , autrement dit  $Y_1=0$ . Il y a donc un mouvement suivant l'axe des  $X_A$  seulement.

## 4.4. Calibration de la caméra

La localisation de points dans un espace tridimensionnel au moyen d'un système de vision implique la connaissance des relations géométriques entre les dispositifs impliqués, c.-à-d. la caméra et l'objet. Ces relations géométriques se trouvent dans les paramètres de la caméra, à savoir la taille des points dans l'image, la distance focale de la caméra, le centre de projection et la position de la caméra dans la scène. Le processus de calcul de ces paramètres est appelé la calibration de caméra [61].

La calibration de caméra permet de déterminer le passage des coordonnées 2D images d'un point aux coordonnées 3D réelles de même point et inversement. La détermination de la relation 3D/2D passe par le choix préalable d'un modèle de caméra et par une phase dite de calibrage destinée à identifier les paramètres de ce modèle.

### 4.4.1. Modèle géométrique de la caméra

La modélisation des caméras consiste à obtenir les paramètres correspondant à la transformation directe et inverse effectuée pour passer des points d'un objet 3D dans l'espace aux points de sa projection 2D dans le plan image, c.-à-d. trouver la matrice de transformation entre le repère caméra et le repère objet.

Pour calculer le modèle géométrique de la caméra, on a utilisé la bibliothèque ARToolKit qui permet de donner, directement, cette matrice en évitant les calculs ennuyeux. Mais, il faut d'abord faire un calibrage de la caméra pour trouver les paramètres intrinsèque de cette dernière. Il s'agit d'une matrice 4x4.

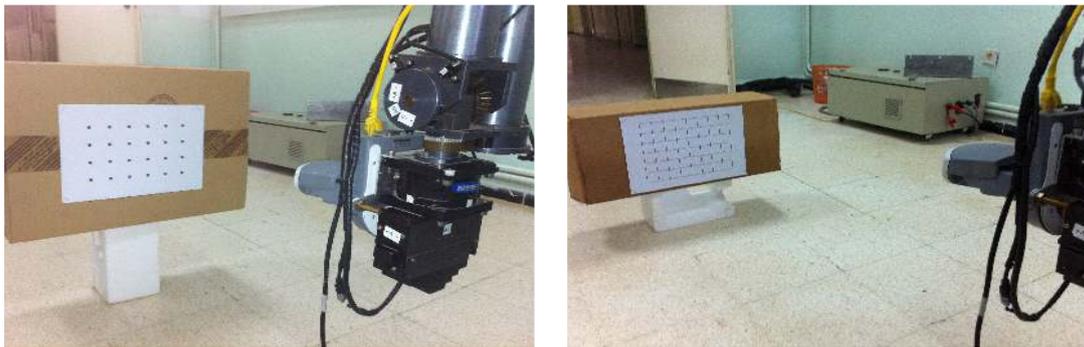
### 4.4.2 Calibration de la caméra

Dans la version actuelle de ARToolKit, les propriétés de la caméra par défaut sont contenues dans le fichier des paramètres de la caméra, 'caméra\_para.dat', qui est lu à chaque fois qu'une application est lancée. Les paramètres doivent être suffisants pour un large éventail de différentes caméras. Cependant, l'utilisation d'une technique d'étalonnage

de l'appareil est relativement simple .Il est possible de générer un fichier séparé de paramètres pour chaque appareil spécifique utilisé. Dans cette section, nous allons expliquer, de manière générale, comment effectuer le calibrage de la caméra IP par l'application d'ARToolKit.

On utilise, pour cela, la grille de lignes présentée dans la figure 60 qui devrait être mis à l'échelle de telle sorte que les lignes sont exactement 40mm et la grille de 6x4 de points qui devraient être modulée de sorte que les points sont de 40mm.

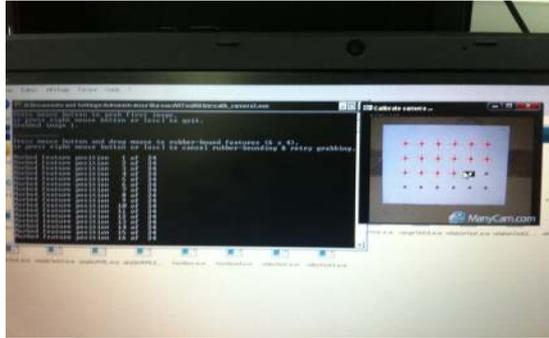
Le programme `calib_dist` est utilisé pour mesurer le point central de l'image et la distorsion de la lentille, tandis que `calib_param` produit les autres propriétés de la caméra, Le programme `calib_dist` doit être exécuté en premier, ensuite, `calib_cparam`, car `calib_cparam` utilise la sortie de `calib_dist` [62].



**Figure60** :Modèle `calib_dist` et modèle `ecalib_cpara`

#### **a. Exécuter `calib_dist`**

On exécute le logiciel `calib_dist` qu'utilise l'image de la figure 61 et on branche la caméra IP. Une fenêtre apparaîtra, affichant la vidéo en direct. Nous Pointons l'appareil photo vers le modèle de calibrage de sorte que tous les points soient en vue et on clique sur le bouton gauche de la souris .Cela fige l'image vidéo, comme le montre la figure 12, ensuite, on clique avec le bouton gauche de la souris sur l'image(en maintenant le bouton de la souris enfoncé) pour sélectionner un rectangle noir sur chaque point de l'image à son tour, en commençant par le point le plus proche du coin supérieur gauche de l'image et continue jusqu'à ce que tous les points soient trouvés [62]. Après chaque rectangle dessiné, un logiciel de traitement d'image trouve le point délimitée par le rectangle et met une croix rouge au centre. On répète ce processus pour les 5 à 10 images sous différents angles et positions.



**Figure 61** : L'utilisateur marque les points d'étalonnage.

Une fois qu'on a pris 5 à 10 images, on quitte l'application pour arrêter la capture d'image et on commence à calculer les valeurs de distorsion de la caméra. Le centre de valeurs x et y et du facteur de distorsion sont les valeurs de clés finales générées par le code `calib_dist` [62]. Chaque image calibrée est donné par la figure 62.



**Figure 62** : Exemple de l'image calibrée

### **b. Exécuter `calib_cparam`**

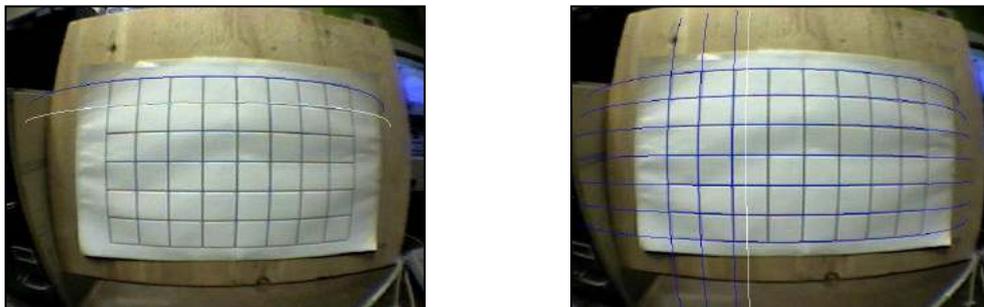
On exécute le programme `calib_cparam` qui est utilisé pour trouver la distance focale et d'autres paramètres .Il utilise le modèle de la figure 62, une fenêtre de vidéo apparaît.

- On met la grille d'étalonnage en face de la caméra de sorte que la carte est perpendiculaire à la caméra et que toutes les lignes de la grille sont visibles.
- On clique sur le bouton gauche de la souris pour saisir l'image. Une ligne horizontale blanche apparaît superposée sur l'image de la figure 14.
- On déplace la ligne blanche à superposer avec la ligne de la grille, la ligne est déplacée vers le haut et vers le bas (à l'aide des touches fléchées vers le haut et vers le bas), alors qu'il est tourné vers la droite et vers la gauche (à l'aide des touches fléchées gauche et droite).Une fois la ligne blanche est aligné avec la ligne de la grilles

supérieure on appuie sur la touche Entrée .Cette ligne va maintenant virer au bleu et une autre ligne blanche apparaît(voir la figure64).Ce processus doit être répété pour toutes les lignes horizontales.

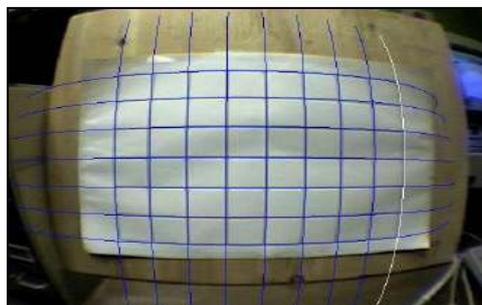
Une fois la dernière ligne horizontale a été placée, une ligne blanche verticale apparaît et le processus doit être répété pour toutes les lignes verticales. La première ligne blanche verticale doit être placée sur la plupart des lignes de la grille de gauche et les autres lignes placées de gauche à droite (voir figure 63).

L'ordre des lignes est très important. Ils doivent être placés de haut en bas, puis, de gauche à droite jusqu'à ce que l'ensemble des 16lignes soient tirées sur l'écran [62].



**Figure 63** : Placement des lignes horizontal et des lignes vertical

- Une fois ce processus terminé, la grille doit être déplacée 100mm loin de la caméra (en gardant la caméra perpendiculaire à la structure) et le processus est répété à nouveau. La figure 15 montre le dernier placement de ligne pour l'une de ces images capturées [10].



**Figure 64** : image calibrée

- On répète le processus cinq fois en déplaçant la mire (un total de 500mm loin de la caméra). Après la cinquième étape de calibrage, le programme calculera

automatiquement les paramètres de la caméra et stocke les paramètres dans un fichier 'caméra\_para.dat' [62].

- On transfère ce dernier fichier de paramètres dans le répertoire 'data' de l'application pour donner une matrice exacte [62].

### c. Création d'un pattern

Divers outils permettent de créer un pattern. Nous avons, pour notre cas, utilisé 'mk\_patt' qui se trouve dans le répertoire 'bin' permettant de générer un 'fichier.patt' ayant la dimension souhaitée.

Pour créer un nouveau schéma de modèle, on imprime d'abord le 'blankPatt.gif' du fichier se trouvant dans le répertoire des modèles. Ceci est juste un carré noir avec un carré blanc vide au milieu. Ensuite, on crée une image (en noir et blanc ou en couleur) du motif désiré qui correspond au milieu de cette place et on l'imprime. Les meilleurs modèles sont ceux qui sont asymétriques. La figure 16 montre quelques modèles d'échantillons possibles [62].

Une fois que le nouveau modèle est finalisé, on accède au répertoire 'bin' et on exécute le programme 'mk\_patt' (en mode console uniquement). Le programme va alors ouvrir une fenêtre vidéo comme le montre la figure 66.

On place le modèle pour être formés sur une surface plane dans des conditions d'éclairage similaires comme il exister lorsque la demande de reconnaissance sera exécuté. Ensuite, on maintient la caméra vidéo au-dessus du modèle, un carré rouge et vert apparaît autour du motif. Ceci indique que le logiciel 'mk\_patt' a trouvé la place autour de la mire.



Figure 65 : Exemples de Patterns.

Une fois que la place a été trouvée et correctement orientée on clique sur le bouton gauche de la souris, et on entre un nom de fichier modèle par exemple 'patt.hiro' [62].



Figure 66 : Fenêtre vidéo du 'mk\_patt'

#### 4.5. Calcul des coordonnées 3D

La méthode de calcul utilisé fait appel à différentes matrices relatives à chaque repère. Cela signifie que chaque repère se déplace relativement à un autre. De ce fait,  $R_B$  et  $R_M$  sont considérés en mouvement par rapport à  $R_A$ , et  $R_E$  se déplace relativement à  $R_M$ . Cette approche est particulièrement structurée car elle est très implicite en termes de représentation et de vision géométrique dans l'espace, dans le sens où la notion de relativité apparaît grâce à la disposition des matrices. La situation de la base mobile ( $X_B, Y_B, \theta_B$ ) est donnée dans  $R_A$  par la matrice de transformation  ${}^A T_B$ , et au cours de déplacement, elle est calculée par odométrie.

La matrice de transformation Manipulateur/Base mobile est donnée par une simple translation. On obtient alors la matrice homogène suivante [03] :

$${}^B T_M = \begin{bmatrix} 1 & 0 & 0 & 30 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 520 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

La transformation repère caméra/repère pince est obtenue directement en mesurant les déplacements suivant x, y et z entre les repères, ainsi qu'une rotation autour de l'axe Z de 180°. On aura alors (les mesures sont en mm) :

$${}^E T_C = \begin{bmatrix} \cos(\pi) & -\sin(\pi) & 0 & 130 \\ \sin(\pi) & \cos(\pi) & 0 & 0 \\ 0 & 0 & 1 & -50 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 130 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -50 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (22)$$

Le marqueur se trouve dans le champ de vision de la caméra mais il ne se trouve pas dans l'espace de travail du manipulateur.

Le système de vision calcule la position de la poignée par rapport au repère bras cela est donné par :

$${}^M T_p = {}^m T_p * {}^C T_m * {}^E T_C * {}^M T_E \quad (23)$$

La matrice résultante est de forme (1), donc on peut déduire la position 3D de la poignée par les équations (6) et (7).

La base mobile se déplace vers une position face à la porte de telle sorte que la poignée se trouve dans l'espace de travail du bras manipulateur. Le marqueur reste toujours dans le champ de vision de la caméra. À ce point-là, le système de vision donne la position exacte de la poignée ( $X_{PM}, Y_{PM}, Z_{PM}, \vartheta_{PM}, \psi_{PM}, \phi_{PM}$ ) par (23)

## 1.6. Conclusion

Nous avons présenté dans ce chapitre l'architecture matérielle et logicielle du système robotique expérimental ainsi que la description détaillée du robot manipulateur mobile *RobuTER/ULM*.

Le modèle géométrique du manipulateur ainsi que le modèle géométrique et cinématique de la base mobile différentielle à roues ont été analysés. La modélisation géométrique du manipulateur a donné les équations nécessaires pour pouvoir positionner l'effecteur dans l'espace atteignable du manipulateur. La modélisation cinématique de la base mobile a permis d'obtenir les équations nécessaires pour la faire mouvoir dans son environnement. Nous avons également présenté le MGI du manipulateur.

Le chapitre a exposé, également, les méthodes de calcul des matrices de calibration de la caméra et pince/caméra en utilisant la bibliothèque ARToolKit.

Le chapitre suivant est dédié à la présentation de l'implémentation de notre système sur le robot manipulateur mobile *RobuTER/ULM*.

# Chapitre 5 Résultats expérimentaux

---

## 5.1. Introduction

Le système doit posséder une Interface Homme/Machine conviviale, qui peut être utilisée facilement par des utilisateurs pas forcément spécialistes du domaine de la robotique ou de l'informatique. Cette interface doit fournir à l'opérateur toutes les informations nécessaires concernant l'état du robot et la scène ainsi que tous les moyens nécessaires pour agir sur le robot.

Ce dernier chapitre présente, tout d'abord, l'interface Homme/Machine et ses différentes parties. Par la suite, il présente une série d'expérimentations de l'exécution de la tâche d'ajustement de l'effecteur sur la poignée de la porte et d'ouverture de cette dernière par le robot. Le chapitre va mettre l'accent sur la partie calcul visuel de la position de poignée de la porte et ajustement visuel.

L'opérateur commence par introduire la tâche à exécuter via l'interface homme/robot. Il donne, manuellement, des coordonnées articulaires au bras manipulateur ( $Q_1, \dots, Q_6$ ) pour se mouvoir vers une position permettant à la caméra de détecter le marqueur. Dès que le marqueur rentre dans le champ de vision de la caméra, l'opérateur lance la tâche d'ouverture de la porte.

Les principaux résultats obtenus ainsi que leurs interprétations et discussions sont présentés à la fin de ce chapitre.

## 5.2. Choix des outils de développement

Pour l'implémentation de la tâche d'ouverture de la porte, nous avons utilisé les moyens logiciels suivants :

- Visual C#.NET 2008.
- c/c++ sous Linux Redhat 6.2.

- Webcam Virtuelle ManyCam 3.1.43.
- Librairie ARToolKit.
- Librairie AForge.NET.

La partie de l'application implémentée sur le PC embarquée du manipulateur mobile RobuTER/ULM est développée en c/c++ sous Linux Redhat 6.2 (compilateurgcc). L'application sur le PC hôte est programmée en « Visual C#.NET 2008 ». Ce dernier est un langage de programmation récent orienté objet à typage fort, créé par Microsoft, et conçu pour la plateforme .NET. Nous avons utilisé aussi la librairie « ARToolKit » pour la détection de marqueurs et l'extraction des coordonnées 3D de la poignée de la porte par rapport à la caméra. « Webcam Virtuelle ManyCam 3.1.43 » pour la capture du flux vidéo à partir de la caméra IP embarquée sur l'effecteur du robot. Enfin, nous avons utilisé la librairie « AForge.NET » pour les effectuer différents traitements d'images.

### 5.3. Interfaces des applications

Cette interface est composée principalement de sept parties. Ces parties sont réparties comme montrée par la fig.67 ci-après :

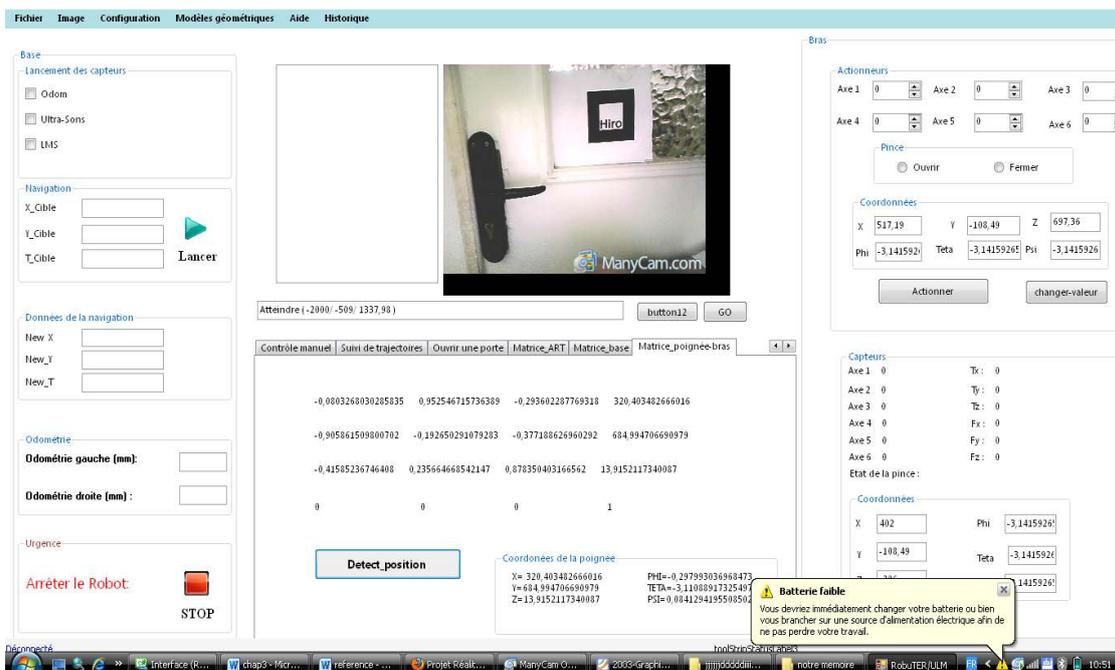


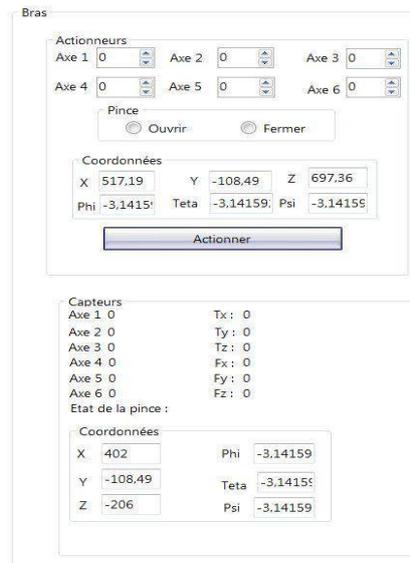
Figure 67 : Interface opérateur de contrôle du manipulateur mobile RobuTER/ULM

- **Partie1** : dans cette partie, donnée par fig.68, on trouve les coordonnées de la position courante de la base mobile ( $New\_X$ ,  $New\_Y$ ,  $New\_θ$ ) et les valeurs des encodeurs droite et gauche (Odométrie gauche, Odométrie droite). On peut aussi lancer la lecture des capteurs (Odométrie, Ultrasons, LMS) ou ordonner à la base mobile de se déplacer vers une position cible ( $X\_cible$ ,  $Y\_cible$ ,  $θ\_cible$ ).



**Figure 68** : Partie concernant la base mobile

- **Partie 2** : cette partie est donnée par fig.69. Elle est, elle-même, constituée de deux sous-parties. La première (actionneurs) sert à actionner le bras manipulateur en utilisant soit la commande axe par axe ( $Axe_1$ , ...,  $Axe_6$ ), soit en spécifiant la position cartésienne imposée ( $X$ ,  $Y$ ,  $Z$ ,  $Phi$ ,  $Theta$ ,  $Psi$ ). Elle permet aussi d'ouvrir ou de fermer la pince. La deuxième sous-partie concerne les capteurs. Elle sert à afficher toutes les données relatives aux différents capteurs ( $Axe_1$ , ...,  $Axe_6$ ,  $T_x$ ,  $T_y$ ,  $T_z$ ,  $F_x$ ,  $F_y$ ,  $F_z$ , État de la pince,  $X$ ,  $Y$ ,  $Z$ ,  $Phi$ ,  $Theta$ ,  $Psi$ ).



**Figure 69** : Partie concernant le bras manipulateur

- **Partie 3** : cette partie concerne l'acquisition des images vidéo (délivrée par la caméra IP installée sur l'effecteur du robot manipulateur mobile) par le système de vision telle que montrée par la figure 70.



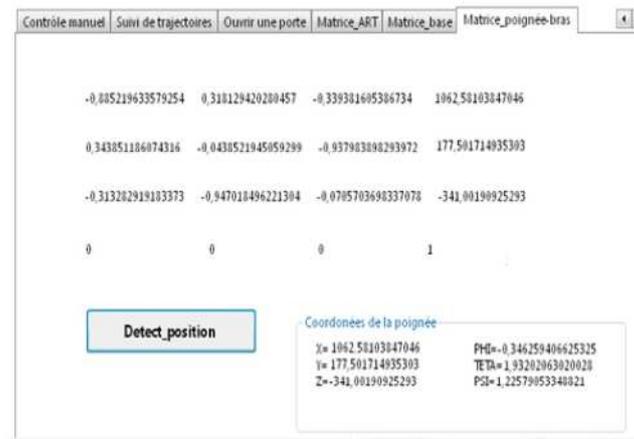
**Figure 70**: Acquisition par système de vision

- **Partie 4** : barre du menu.
- **Partie 5** : cette partie est donnée par la fig.71. Elle permet l'affichage des données issues du capteur LMS sur un radar et des valeurs des capteurs ultrasons.



**Figure 71** : Affichage des données du LMS.

- **Partie 6** : cette partie est donnée par la fig.73. Elle sert à calculer et extraire les coordonnées réelles 3D de la poignée de la porte par rapport au repère lié au bras manipulateur, et les introduire dans la partie ouvrir la porte (partie 7). Elle permet aussi d'afficher la matrice de transformation poignée de la porte/bras manipulateur.



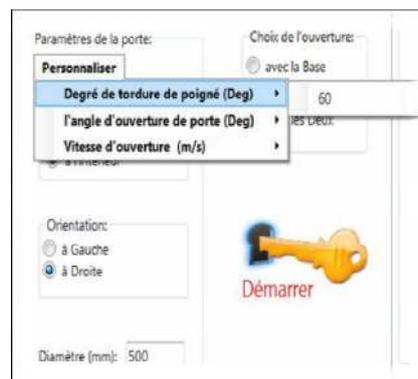
**Figure 72** : Affichage de la matrice poignée/bras

- **Partie7 (Ouvrir Porte)** : cette partie est donnée par la fig.73. Elle concerne principalement la tâche d'ouverture d'une porte. Elle est composée de trois sous-parties réparties comme suit :
  - la première sous-partie présente les divers paramètres à fournir au robot pour pouvoir accomplir la tâche d'ouverture d'une porte.
  - la deuxième sous-partie permet de sauvegarder, dans une base de données, les différents paramètres de la porte.
  - la troisième sous-partie permet de lancer l'exécution de la tâche.



**Figure 73** : Partie concernant la tâche d'ouverture d'une porte.

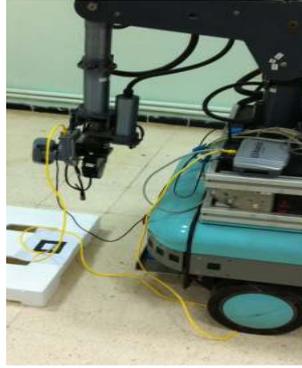
- **Partie supplémentaire:** cette partie est donnée par la fig.74. Elle présente les paramètres à personnaliser concernant la tâche d'ouverture et qui sont donnés par le degré de torsion de la poignée ( $^{\circ}$ ), l'angle d'ouverture de la porte ( $^{\circ}$ ) et, enfin, la vitesse d'ouverture (m/s).



**Figure 74** : Paramètres personnalisés

## 5.4. Ouverture d'une porte

L'opérateur lance l'interface homme/robot et se connecte au robot à distance via une connexion wifi. Dès que la liaison soit établie, une calibration du bras manipulateur est lancée automatiquement pour le positionner à (402mm, -108.49mm, -206mm,  $-\pi$ ,  $-\pi$ ,  $-\pi$ ). Cette situation correspond aux coordonnées articulaires ( $Q_1=0$ ,  $Q_2=0$ ,  $Q_3=0$ ,  $Q_4=0$ ,  $Q_5=0$ ,  $Q_6=0$ ) comme montrée par la fig.75.



**Figure 75** : Situation initiale du bras manipulateur après calibration de ce dernier

Ensuite, l'opérateur introduit les autres paramètres du bras manipulateur ( $Q_1=0$ ,  $Q_2=80$ ,  $Q_3=0$ ,  $Q_4=0$ ,  $Q_5=0$ ,  $Q_6=0$ ) qui correspondent à la position (558mm, 27mm, -108.49mm, 599.76rad, -1.57rad, -1.57rad) telle que montrée par la fig.76. Cette configuration correspond à la position de capture de la porte pour la caméra.



**Figure 76** : Position de capture pour la caméra

#### **5.4.1. Étape 1 : Positionnement du robot face à la porte**

Deux cas peuvent être distingués :

##### **a. La poignée appartient à l'espace de travail du bras manipulateur**

Le déroulement de la tâche d'ouverture est donné comme suit : le système de vision calcule la position de la poignée par rapport à celle du bras manipulateur. Elle est donnée par le produit matriciel de toutes les matrices mises en jeu (voir fig.78).



**Figure 78** : La poignée de la porte appartient à l'espace de travail

Ce calcul de produit matriciel se fait en temps-réel. L'opérateur appuie sur le bouton « Detect\_position » afin d'extraire les coordonnées 3D de la poignée et vérifier si cette dernière appartient ou non à l'espace de travail courant du manipulateur mobile (fig.79). Ensuite, l'opérateur les introduit dans la partie « Ouvrir porte ». Cette partie va déterminer les différentes  $Q_i$  ( $i=1, \dots, 6$ ) correspondant à la position de la poignée en utilisant le MGI du bras manipulateur.



**Figure 79** : Calcul de la position 3D de la poignée

Maintenant, l'opérateur clique sur « démarrer » pour lancer la tâche d'ouverture de la porte et ouvrir la pince du robot. À la fin du mouvement, la pince se trouve automatiquement ajustée sur la poignée (fig.80). Par la suite, la pince se ferme sur la poignée. Si la pince n'arrive pas à saisir la poignée, le programme refait l'ajustement une autre fois jusqu'à pouvoir la saisir.



**Figure 80** : Ajustement de la pince sur la poignée

### **b. La poignée n'appartient pas à l'espace du travail du bras manipulateur**

La base mobile se trouve à une position initiale éloignée de la porte telle que montrée par la fig.81. Le bras manipulateur, quant à lui, se trouve à la position initiale comme montrée précédemment par la fig.78.



**Figure 81** : La poignée n'appartient pas à l'espace de travail actuel du robot

Le système de vision calcule la position de la poignée par rapport au bras manipulateur. L'opérateur appuie sur le bouton « Detect\_position » pour extraire les coordonnées 3D de la poignée et vérifier l'appartenance de cette dernière à l'espace de travail actuel du manipulateur mobile. Dans ce second cas, la poignée ne se trouve pas dans cette espace. Alors, il demande à la base mobile de se déplacer vers la porte et de s'arrêter à une distance de :

- 500mm : si la porte s'ouvre vers l'extérieur.
- 650mm : si la porte s'ouvre vers l'intérieur.

Durant le mouvement de la base mobile, le marqueur doit rester toujours dans le champ de vision de la caméra. Une procédure de calcul et d'ajustement se déroule comme

prévu pour le premier cas. Malheureusement dans ce cas et à cause de la panne du capteur laser LMS, nous étions obligés de faire mouvoir la base mobile manuellement à l'aide du « joystick » du robot jusqu'à sa position finale.

#### 5.4.2. Étape 2 : Ouverture de la porte

Selon le choix du sens d'ouverture de la porte, trois cas se présentent :

- Mouvement de la base mobile toute seule : il est nécessaire que la base mobile suive un arc lors de cette étape. Le calcul de la position finale à atteindre par la base mobile se fait en fonction du degré d'ouverture de la porte.
- Mouvement du bras manipulateur tout seul : ici, la base mobile reste immobile lors du mouvement du manipulateur.
- Mouvement des deux : pour ce dernier cas, les deux sous-systèmes (base mobile et bras manipulateur) doivent effectuer un mouvement coordonné afin de pouvoir ouvrir la porte.

Malheureusement, le processus d'ouverture de la porte n'a pas été accompli. Cela est dû principalement à une panne matérielle au niveau du capteur laser LMS et de l'axe 6 du bras manipulateur.

#### 5.5. Résultat obtenus et discussion des résultats

Plusieurs tests ont été effectués pour l'ajustement automatique et visuel de la pince du manipulateur mobile sur la poignée d'une porte qui s'ouvre vers l'intérieur. Les résultats obtenus sont présentés dans ce qui suit :



Test 1

Test 2

Test 3

Test 4

**Figure 82** : Different tests d'ajustement effectués

Les positions de la poignée calculées par rapport au repère lié au bras manipulateur lors des différents tests sont présentées dans le tableau suivant :

	$X_{pM}$	$Y_{pM}$	$Z_{pM}$	Phi	Theta	Psi
Test1	850.60	-301.14	441.23	-1.72	-1.83	-0.24
Test2	826.23	-243.14	423.72	-1.71	-1.83	-0.24
Test3	865.05	-240.89	453.55	-1.72	-1.83	-0.24
Test4	814.31	-280.26	397.49	-1.72	-1.83	-0.24

**Tableau 3** : Positions de la poignée calculées par rapport au repère lié au bras manipulateur

L'erreur de positionnement de l'effecteur à la fin du mouvement du manipulateur est calculée comme suit :

$$Erreur = \sqrt{(X_{PM} - X_{EMFin})^2 + (Y_{PM} - Y_{EMFin})^2 + (Z_{PM} - Z_{EMFin})^2}$$

$$40 \leq Erreur \leq 15$$

tels que :

- $(X_{PM}, Y_{PM}, Z_{PM})$  sont les coordonnées de la poignée dans le repère lié au bras manipulateur.
- $(X_{EMfin}, Y_{EMfin}, Z_{EMfin})$  représentent les coordonnées de l'effecteur dans le repère lié au bras manipulateur calculé par le MGD du bras manipulateur.

L'erreur maximale était de l'ordre de 40mm. L'erreur minimale, quant à elle, est de l'ordre de 15mm. L'erreur moyenne pour les différents tests effectués était de l'ordre de 27mm.

Cette erreur est causé par la matrice que donne ArtoolKIT, cette matrice de passage entre le repère caméra et repère marqueur est trop sensible a la lumières et de la taille du marqueur, cette matrice se change en temps réel et les éléments de la matrice ne donne pas toujours la même valeur. Cette erreur s'accumule avec l'erreur du produit matriciel de passage d'un repère à un autre.

## 5.6. Conclusion

Dans ce chapitre, nous avons présenté les résultats d'ajustement de la pince sur la poignée de la porte obtenus lors de l'exécution de cette phase (ajustement de la pince sur la poignée d'une porte) par un manipulateur mobile dans un environnement d'intérieur. Cette phase rentre dans le cadre de la tâche d'ouverture d'une porte.

La tâche d'ouverture de la porte a été exécutée par le robot dans deux cas distincts. Le premier correspond à la saisie de la poignée de la porte, lorsque cette dernière appartient à son espace de travail actuel. Le deuxième cas, quand la poignée de la porte est en dehors de son espace de travail courant.

Pour l'ajustement visuel, le robot doit se trouver devant la porte. Le robot ne fait appel qu'au système visuel pour déterminer la position de la poignée et effectuer, ainsi, l'insertion de la pince dans la poignée de la porte.

En général, le taux de réussite était faible et dans 30% des cas traités, le robot était capable d'accomplir, avec succès, sa tâche.

# Introduction générale

---

Le domaine de la robotique est un domaine de recherche très vaste, et en pleine expansion, il a connu ces dernières années des développements énormes grâce aux progrès dans le domaine informatique et aux avancées technologiques dans tous les secteurs scientifiques. Ces développements technologiques ont permis de passer de la génération des robots manipulateurs fixes à celles des robots mobiles, puis à celle des robots manipulateurs mobiles, qui combinent la tâche de locomotion et la manipulation à la fois.

Un Robot Mobile Manipulateur est une plate forme mobile surmonté d'un bras manipulateur. L'association de ces deux systèmes augmente la capacité de manipulation et le champ de travail de ce robot. Grâce à cette structure un bras manipulateur peut atteindre n'importe quel point se trouvant dans son environnement.

La vision en robotique contribue de permettre de traiter les informations visuelles utiles à leurs tâches perceptives : orienter dans un espace tridimensionnel, éviter des obstacles, détecter un mouvement, se guider sur une trajectoire, ainsi l'ajustement visuelle d'objet en 3D.

L'utilisation de la vision en robotique va, non seulement augmenter l'autonomie et l'efficacité des robots, mais également les doter de capacités de reconnaissance et de raisonnement basées sur les informations visuelles.

L'utilisation des manipulateurs mobiles ne se limitent pas seulement dans les domaines d'industrie ou pour les milieux hostiles, actuellement grande évolution des robots d'environnement humain (robots de service), Parce qu'ils sont, particulièrement, mieux adaptés aux missions humaines, ces robots sont présents actuellement dans les bureaux, les hôpitaux, les maisons pour l'assistance aux personnes handicapées et/ou personnes âgées, etc. Parmi ces missions, on trouve l'ouverture d'une porte qui fait l'objet de ce travail.

L'objectif de notre projet consiste à ouvrir la porte par l'ajustement visuelle et automatique de la pince d'un manipulateur mobile sur la poignée de la porte en exploitant la caméra embarquée.

L'implémentation de cette mission a été effectuée sur le manipulateur mobile non-holonome *RobuTER/ULM* disponible au niveau de la *Division Productique et Robotique (DPR)* du *Centre de Développement des Technologies Avancées (CDTA)*. Ce manipulateur mobile est constitué d'un bras manipulateurs *ULM* monté sur une plateforme mobile non-holonome de type *RobuTER*.

Les chapitres de ce mémoire sont rédigés selon ce principe : La théorie avant la méthode, la méthode avant les applications

*Le chapitre 1* présente les manipulateurs mobiles. Il donne les divers types des manipulateurs mobiles existant dans le monde, leurs caractéristiques et leurs différents capteurs utilisés ainsi leurs principaux domaines d'applications.

La formulation de problème d'ouverture d'une porte est décrite dans *le chapitre 2*. Le chapitre contient aussi l'état de l'art sur les divers travaux effectués et traitant ce problème.

*Le chapitre 3* est consacré à l'état de l'art. Nous présentons la modélisation géométrique de la caméra, la bibliothèque Artoolkit, ainsi que les différentes méthodes de calibrage de la caméra, ainsi les méthodes de calibrage pince caméra.

*Le chapitre 4* présente l'architecture de système robotique, la description matériel et logiciel détaillée de *RobuTER/ULM*, et fournit les différentes étapes de l'implémentation de notre application.

Dans *le dernier chapitre* présente les résultats expérimentaux et la validation de notre travail sur le robot manipulateur mobile *RobuTER/ULM*.

Une conclusion résumant le travail fait et discutant les résultats obtenus ainsi que des perspectives d'amélioration de ce travail, est donnée à la fin de ce document.

## Conclusion générale

---

De nombreux laboratoires de robotique dans le monde travaillant sur les bases mobiles ou les bras manipulateurs se dotent actuellement des manipulateurs mobiles. Si du point de vue des perspectives d'utilisation, ces systèmes découpent les possibilités, ils offrent également de nouveaux enjeux scientifiques en raison de leur redondance et de la difficulté de modéliser les différentes tâches pouvant leur être assignées. En effet, enchaîner des tâches de navigation et de manipulation, décomposer en sous-tâches, décider de la coordination entre le bras manipulateur et la base mobile, restent autant de problèmes à résoudre.

Un manipulateur mobile peut accomplir plusieurs types de tâches précises sollicitant les deux sous-systèmes (manipulateur et base mobile). On cite, entre autres, le suivi de trajectoires opérationnelles imposées à l'effecteur en évitant les obstacles, la saisie et le transport d'objets, l'ouverture d'une porte, etc.

Afin de pouvoir étudier et proposer des solutions aux tâches citées précédemment, l'équipe *Systèmes Robotisés de Production (SRP)* de la *Division Productique et Robotique (DPR)* du *Centre de Développement des Technologies Avancées (CDTA)* s'est dotée, depuis quelques années, d'un manipulateur mobile *RobuTER/ULM* nu et ouvert. Le robot consiste en une base mobile rectangulaire non-holonome (*RobuTER*) sur laquelle est monté un manipulateur ultraléger (*ULM*) avec une pince. Le robot est contrôlé par un PC embarqué (on-board) et par un système de commande à base de cartes microcontrôleurs MPC555. *RobuTER/ULM* peut aussi être contrôlé par un PC distant (off-board) qui communique avec le PC on-board via des systèmes de communication sans fil.

Un des objectifs des travaux réalisés sur ce robot manipulateur mobile consiste à ouvrir une porte en utilisant l'effecteur d'un robot. Malheureusement, deux problèmes ont été rencontrés. Le premier est relatif au calcul des coordonnées réelles 3D de la poignée de

la porte. Le second concerne l'ajustement visuel et automatique de la pince du robot sur la poignée de la porte (en utilisant la caméra IP embraquée eye-in-hand).

L'objectif de ce travail est de résoudre les deux problèmes cités précédemment exploitant les images issues de la caméra embraquée et en exploitant la bibliothèque de développement d'applications de réalité augmentée *ARToolKit*. Ses fonctions permettent de calculer, en temps réel, la position et l'orientation de la caméra par rapport à des marqueurs placés dans la scène.

L'application a été réalisée en des différentes étapes :

- La première étape concerne le placement des différents marqueurs dans la scène.
- La deuxième étape est relative à la calibration (avec un logiciel de calibrage donné par *ARToolKit*) de la caméra IP.
- La troisième étape permet de calculer les coordonnées réelles 3D de la poignée de la porte par rapport la caméra. Pour cela, la bibliothèque *ARToolKit* a été utilisée pour donner la position exacte de l'objet par rapport à la caméra.
- La quatrième étape et après avoir calculé cette position, nous avons utilisé les transformations homogènes entre les différents repères avec le repère fixe lié à la scène et le repère lié au robot. Ce calcul permet de localiser le robot dans son environnement. Cette transformation est représentée par une matrice de translation et rotation.
- À la fin (cinquième étape), nous avons reliés les matrices de chaque repère et calculés la position de la poignée par rapport au repère lié au bras manipulateur pour, enfin, déduire le mouvement de chaque articulation et ajuster automatiquement la pince du robot sur la poignée de la porte.

Deux cas possibles ont été distingués pour la tâche d'ouverture d'une porte selon que cette dernière se trouve dans l'espace atteignable du manipulateur ou en dehors de cet espace :

- *Premier cas* : la tâche consiste à ouvrir une porte qui ne fait pas partie de l'espace de travail du bras manipulateur. Un mouvement coordonné entre les deux sous-systèmes robotiques (bras manipulateur et base mobile) est réalisé afin de pouvoir ouvrir cette porte.

- *Deuxième cas* : la tâche consiste à ouvrir une porte qui appartient à l'espace de travail du bras manipulateur. Ici, la tâche ne nécessite que le mouvement du bras manipulateur seulement afin de pouvoir ouvrir cette porte.

Pour l'ajustement de la pince sur la poignée de la porte, le robot doit se trouver devant cette dernière. Pour cela, il ne fait appel qu'au système visuel pour déterminer la position de la poignée et insérer, ainsi, sa pince dans la poignée de la porte. Le taux de succès de l'ajustement était faible. En général, dans 30% des cas traités, le robot accomplissait, avec succès, l'insertion et l'ajustement.

Le processus d'ouverture de la porte n'a pas été accompli. Cela est dû principalement à deux pannes matérielles :

- La première au niveau du capteur laser LMS de la base mobile qui est indispensable pour la navigation et l'évitement d'obstacles.
- La deuxième concerne la panne de l'axe 6 du bras manipulateur qui est indispensable pour tordre la poignée et ouvrir, ainsi, la porte.

Les plus qui peuvent être rajoutés pour améliorer les résultats obtenus dans notre tâche sont :

- d'utiliser une caméra spécifique pour la robotique qui permet un calibrage puissant et plus précis.
- ajouter une autre caméra dans la scène ainsi que d'autres marqueurs postés sur le robot pour améliorer la précision de l'ajustement visuelle au moment de l'insertion.

## 1. Introduction

Cette annexe est consacrée à l'intégration de la mission d'ouverture d'une porte dans l'architecture multi-agents de contrôle de ces robots décrite dans le chapitre précédent.

Trois types de diagrammes AUML ont été choisis :

- *Diagramme d'agents*: Ce diagramme est composé de six agents décrivant le plan d'opération nécessaire afin de réaliser la mission désirée.
- *Diagramme d'activité* : Il décrit les différents modules nécessaires pour la réalisation de la mission. Chaque module est un ensemble d'opérations planifiées par un ou plusieurs agents.
- *Diagramme de séquence* : Ce diagramme présente les différentes compétences de chaque agent. Ce diagramme présente aussi le schéma de communication entre les différents agents.

## 2. Diagramme d'agents

Le diagramme d'agents constitue un élément très important de modélisation. En effet, il permet de définir quelles seront les composantes du système final. En revanche, il ne permet pas de définir le nombre et l'état des instances individuelles. Néanmoins, on constate souvent qu'un diagramme de classes, proprement réalisé, permet de structurer le travail de développement de manière très efficace. Enfin, il permet de construire le système de manière correcte [38].

Le diagramme de la figure 1 présente les différents agents de l'architecture de contrôle présentée dans le chapitre trois. Chacun de ces agents représente une classe de diagramme. Chaque classe comporte un ensemble d'opérations (compétences) pour réaliser les missions confiées au robot. Chaque agent est composé d'un ensemble de modules.

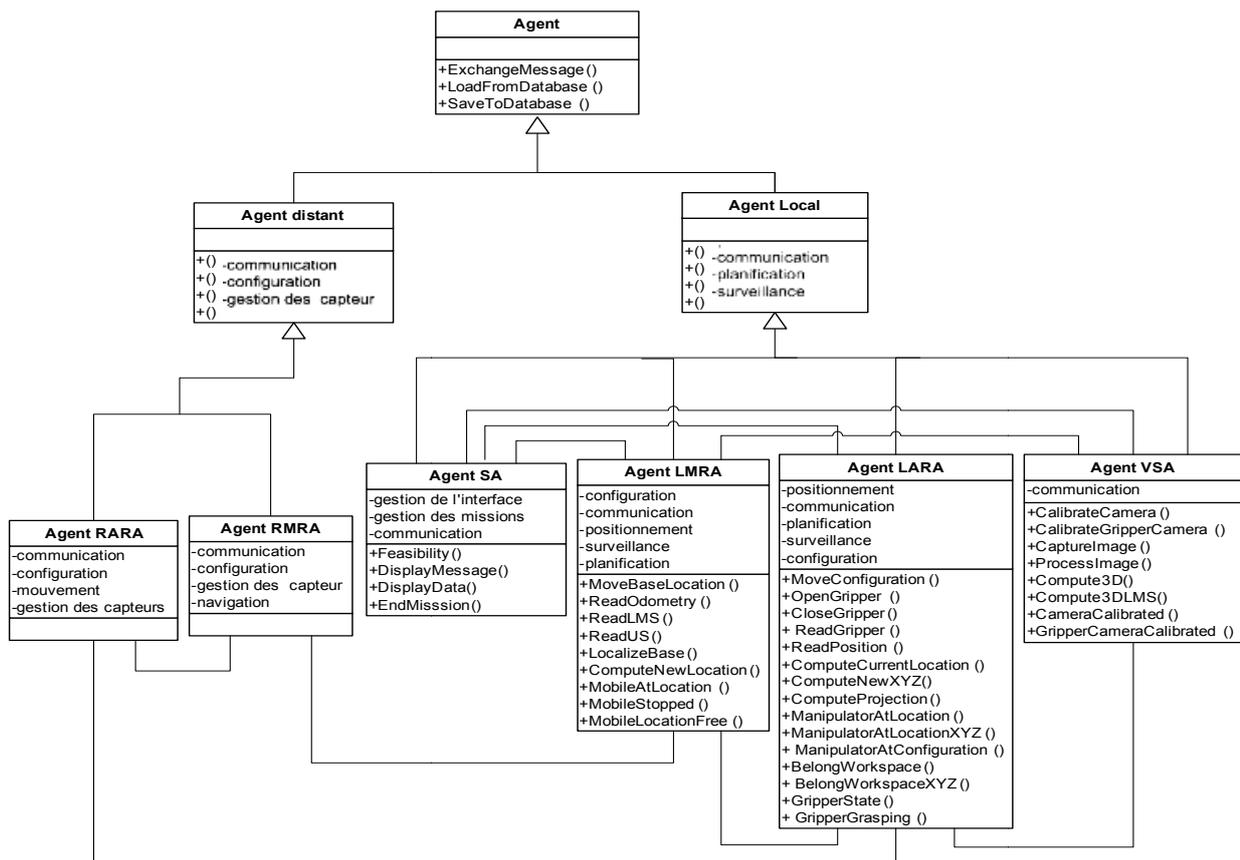


Figure.1. : Diagramme d'agents.

## 2.1. Compétences d'agents

Chaque agent dispose d'un ensemble d'opérations (compétences). De plus, chacun a les trois compétences suivantes en plus de celles données dans sa sous-section respective [38]:

- *ExchangeMessage(Sender, Receiver, Message)* : Il permet à *Sender* d'échanger *Message* avec *Receiver*.
- *LoadFromDatabase(Data)* : Cette opération charge *Data* de la base de données de la connaissance de l'agent.
- *SaveToDatabase(Data)* : Cette opération permet à l'agent de sauvegarder des données *Data* dans sa base de connaissance.

Les différentes compétences nécessaires pour la réalisation de la mission d'ouverture de porte sont données comme suit :

### 2.1.1. Supervisory Agent (SA)

Cet agent comporte quatre opérations [38] :

- *Feasibility ()* : Cette opération permet de vérifier la possibilité de l'exécution de la mission.
- *DisplayMessage (Message)* : Cette opération permet l'affichage de *Message* sur l'écran de l'opérateur.
- *DisplayData (Data)* : elle visualise des données *Data* sur l'interface homme/robot de l'opérateur.
- *EndMission (Success/Error)* : il termine la mission en cours soit en cas d'erreur (*Error*) soit après exécution de la mission (*Success*).

### 2.1.2. Local Manipulator Robot Agent (LARA)

Neuf opérations sont mises en application pour cet agent [38] :

- *MoveManipulator*( $x, y, z, \psi, \theta, \varphi$ ) : cette opération envoie une demande à *RARA* pour déplacer l'organe terminal du robot à l'endroit ( $x, y, z, \psi, \theta, \varphi$ ).
- *MoveManipulatorXYZ*( $x, y, z$ ) : elle envoie une demande à *RARA* afin de déplacer l'organe terminal du robot à l'endroit ( $x, y, z$ ).
- *MoveConfiguration*( $Q1 \dots Q6$ ) : cette compétence envoie une demande à *RARA* pour déplacer le manipulateur à la configuration désirée ( $Q1 \dots Q6$ ).
- *OpenGripper*() : il envoie une demande à *RARA* afin d'ouvrir la pince du manipulateur.
- *CloseGripper*() : cette opération envoie une demande à *RARA* afin de fermer la pince du manipulateur.
- *ReadGripper*( $Fx, Fy, Fz, Tx, Ty, Tz, Gripper\_State$ ) : il envoie une demande de lecture, à *RARA*, du capteur d'effort ( $Fx, \dots, Tx$ ) et de l'effecteur (*Gripper\_State*).
- *ReadPosition*( $P1 \dots P6$ ) : cette opération envoie une demande, à *RARA*, pour lire les capteurs incrémentaux de position ( $P1, \dots, P6$ ).
- *ComputeCurrentLocation*( $x_B, y_B, \theta_B, Q1 \dots Q6, x_E, y_E, z_E, \psi_E, \theta_E, \varphi_E$ ) : il calcule la position actuelle et l'orientation ( $x_E, y_E, z_E, \psi_E, \theta_E, \varphi_E$ ) de l'organe terminal, en  $R_A$ , en fonction de la position actuelle et de l'orientation de la base mobile ( $x_B, y_B, \theta_B$ ) et, les valeurs des coordonnées d'articulations du manipulateur ( $Q1 \dots Q6$ ).
- *ComputeNewXYZ*( $x_A, y_A, z_A, x_B, y_B, \theta_B, x_m, y_m, z_m$ ) : il renvoie les nouvelles coordonnées de la cible ( $x_m, y_m, z_m$ ), dans  $R_M$ , correspondant à la position initiale de la cible ( $x_A, y_A, z_A$ ), donnée en  $R_A$ , en fonction de la position actuelle et de l'orientation de la base mobile ( $x_B, y_B, \theta_B$ ).

### 2.1.3. Local Mobile Robot Agent (LMRA)

Sept opérations sont disponibles pour *LMRA* [38] :

- *MouveBase*( $V_R, V_L$ ) : elle consiste à faire mouvoir la base mobile avec les vitesses  $V_R$  et  $V_L$  de la roue gauche et droite respectivement.
- *MoveBaseLocation*( $x_B, y_B, \theta_B$ ) : cette opération envoie une demande à *RMRA* pour déplacer la base mobile à l'endroit ( $x_B, y_B, \theta_B$ ) tout en évitant des obstacles possibles.
- *ReadOdometry*( $Odo\_X, Odo\_Y, Odo\_theta$ ) : il envoie une demande, à *RMRA*, afin de lire les données odométrique de la base mobile ( $Odo\_X, Odo\_Y$ ) et calculer  $Odo\_theta$ .
- *ReadLMS*(*ValuesLMS*) : cette compétence envoie une demande, à *RMRA*, pour lire le capteur LMS.
- *ReadUS*(*ValuesUS*) : cette opération envoie une demande, à *RMRA*, pour lire les capteurs US.
- *LocalizeBase*( $Odo\_X, Odo\_Y, Odo\_theta, x_B, y_B, \theta_B$ ) : il calcule la position et l'orientation actuelle de la base mobile ( $x_B, y_B, \theta_B$ ), dans  $R_A$ , en fonction des données odométrique ( $Odo\_X, Odo\_Y, Odo\_theta$ ).
- *ComputeNewLocation*( $x, y, z, x_B, y_B, \theta_B$ ) : il envoie la nouvelle position et orientation de la base mobile ( $x_B, y_B, \theta_B$ ) correspondant à l'endroit ( $x, y, z$ ). *NewLocation* est calculée de sorte que l'endroit appartienne à la nouvelle zone de travail accessible du manipulateur.

### 2.1.4. Vision System Agent (VSA)

La quatrième catégorie inclut cinq opérations pour *VSA* :

- *CalibrateCamera*(*Camera\_Matrix*) : il fournit les paramètres intrinsèques et extrinsèques (*Camera\_Matrix*) de la caméra et les sauvegarde dans la base de connaissances.
- *CalibrateGripperCamera*(*Gripper\_Matrix*) : cette deuxième opération calcule les paramètres de calibrage caméra/pince (*Gripper\_Matrix*). Ces derniers sont enregistrés dans la base de connaissances du *VSA*.
- *CaptureImage*(*image*) : elle capture l'image du flux continu de la caméra CCD.

- *ProcessImage(image, UV, n)* : cette compétence traite l'image capturée, identifie et classe les différentes formes. Cette opération extrait également les coordonnées 2D (u, v) de chaque forme identifiée. Ces coordonnées sont sauvegardées dans tableau UV de n éléments.
- *Compute3D(u<sub>1</sub>, v<sub>1</sub>, u<sub>2</sub>, v<sub>2</sub>, x<sub>R</sub>, y<sub>R</sub>, z<sub>R</sub>)* : il calcule les coordonnées réelles (x<sub>R</sub>, y<sub>R</sub>, z<sub>R</sub>) d'un objet.

## 2.2. Pré-conditions

Différentes catégories des pré-conditions sont distinguées. Ces contraintes de précédence sont générées automatiquement. Seulement les opérations sans pré-conditions sont lancées immédiatement. Les autres opérations du plan seront lancées quand leurs contraintes respectives sont satisfaisantes.

### 2.2.1. Supervisory Agent (SA)

Seulement une pré-condition est distinguée pour cet agent [38] :

- *Feasibility(Mission)*: elle indique si le robot peut exécuter la mission selon la disponibilité et le statut de toutes les ressources exigées et, selon d'autres paramètres spécifiques à chaque mission.

### 2.2.2. Local Manipulator Robot Agent (LARA)

Sept pré-conditions pour LARA [38] :

- *ManipulatorAtLocation(x, y, z, ψ, θ, φ)* : elle indique si l'effecteur est à l'endroit (x, y, z, ψ, θ, φ) ou non.
- *ManipulatorAtLocationXYZ(x, y, z)* : elle vérifie si l'effecteur est à l'endroit (x, y, z) ou non.
- *ManipulatorAtConfiguration(Q1... Q6)*: elle examine si le manipulateur est à la configuration (Q1... Q6).
- *BelongWorkspace(x, y, z, ψ, θ, φ)* : elle détermine si l'endroit (x, y, z, ψ, θ, φ) appartient à l'espace de travail courant du manipulateur.
- *BelongWorkspaceXYZ(x, y, z)* : cette condition indique si (x, y, z) appartient de l'espace de travail courant du manipulateur.
- *GripperState(Opened/Closed)* : cette pré condition examine si la pince du robot est ouverte/fermé.
- *GripperGrasping(Holding/Free)* : elle indique si la pince tient un objet ou elle est libre.

### 2.2.3. Local Mobile Robot Agent (LMRA)

Trois pré conditions sont distinguées pour LMRA [38] :

- *MobileAtLocation(x, y, θ)* : elle indique si la base mobile est à l'endroit (x, y, θ).
- *MobileStopped()* : cette condition indique si la base mobile est arrêtée ou elle est en mouvement.
- *MobileLocationFree(x, y, θ)* : elle vérifie si l'endroit (x, y, θ) est libre ou occupé par un obstacle.

### 2.2.4. Vision System Agent (VSA)

Pour le VSA, deux pré conditions sont distinguées [38] :

- *CameraCalibrated()* : il indique si les paramètres intrinsèques et extrinsèques de la caméra ont été calculés.
- *GripperCameraCalibrated()* : elle détermine si les paramètres de calibrage pince/caméra ont été trouvés.

## 3. Diagramme d'activité

Ce diagramme présente le séquençement des différents modules nécessaires pour effectuer la mission d'ouverture de porte. Ces modules consistent en le : Module de faisabilité, Module de

localisation, Module de déplacement, Module de mouvement du bras, Module d'ajustement, Module de saisie de la poignée, Tordre la poignée, (Tirer ou pousser) la porte. Cette mission implique tous les agents de l'architecture de contrôle.

Au début, quand l'agent *Supervisory (SA)* reçoit la mission, il va tester si le robot est capable l'exécuter ?

Si le robot ne peut pas effectuer cette mission, *SA* va afficher un message d'échec sur l'interface de l'opérateur. Dans le cas contraire (la mission est acceptée), *SA* va envoyer la mission à l'agent *VSA* pour localiser la cible (la poignée). Par la suite, cet agent utilise la caméra pour capturer une image de la porte, la localiser dans l'image et extraire la position réelle de la poignée. Après, il envoie ces informations à l'agent *LARA*. Deux cas possibles :

- *La poignée appartient à la zone de travail actuelle du bras manipulateur* : *LARA* effectue les calculs nécessaires et envoie l'information à *RARA* pour effectuer le mouvement jusqu'à atteindre la position désirée.
- *La poignée n'appartient pas à la zone de travail courante du bras manipulateur* : ici, *LARA* envoie les coordonnées de la poignée à l'agent *LMRA*. Ce dernier calcule la nouvelle position de la base, qui permet au bras d'atteindre la poignée et envoie ces informations à *RMRA* pour faire mouvoir la base mobile.

Par la suite, un travail coopératif entre *VSA*, *LARA* et *RARA* afin d'ajuster l'effecteur sur la position exacte de la poignée et, après, la saisir.

L'étape suivante consiste à faire tourner la poignée. Ici, *LARA* et *RARA* sont concernés par cette opération. Si le robot n'arrive pas à faire tourner la poignée dans le premier sens, il va changer de position de prise de poignée à l'autre extrémité et essayer de faire tourner la poignée dans l'autre sens. Si ça ne marche pas, donc échec de la mission car la porte est considérée comme étant fermée.

La dernière étape de cette mission consiste à déplacer le robot tout en maintenant la poignée de la porte jusqu'à ce qu'elle soit entièrement ouverte. Cette étape nécessite la coopération entre la base mobile et le bras manipulateur.

Le diagramme de d'activité associé à la mission d'ouverture de la porte est donnée par la figure 2.

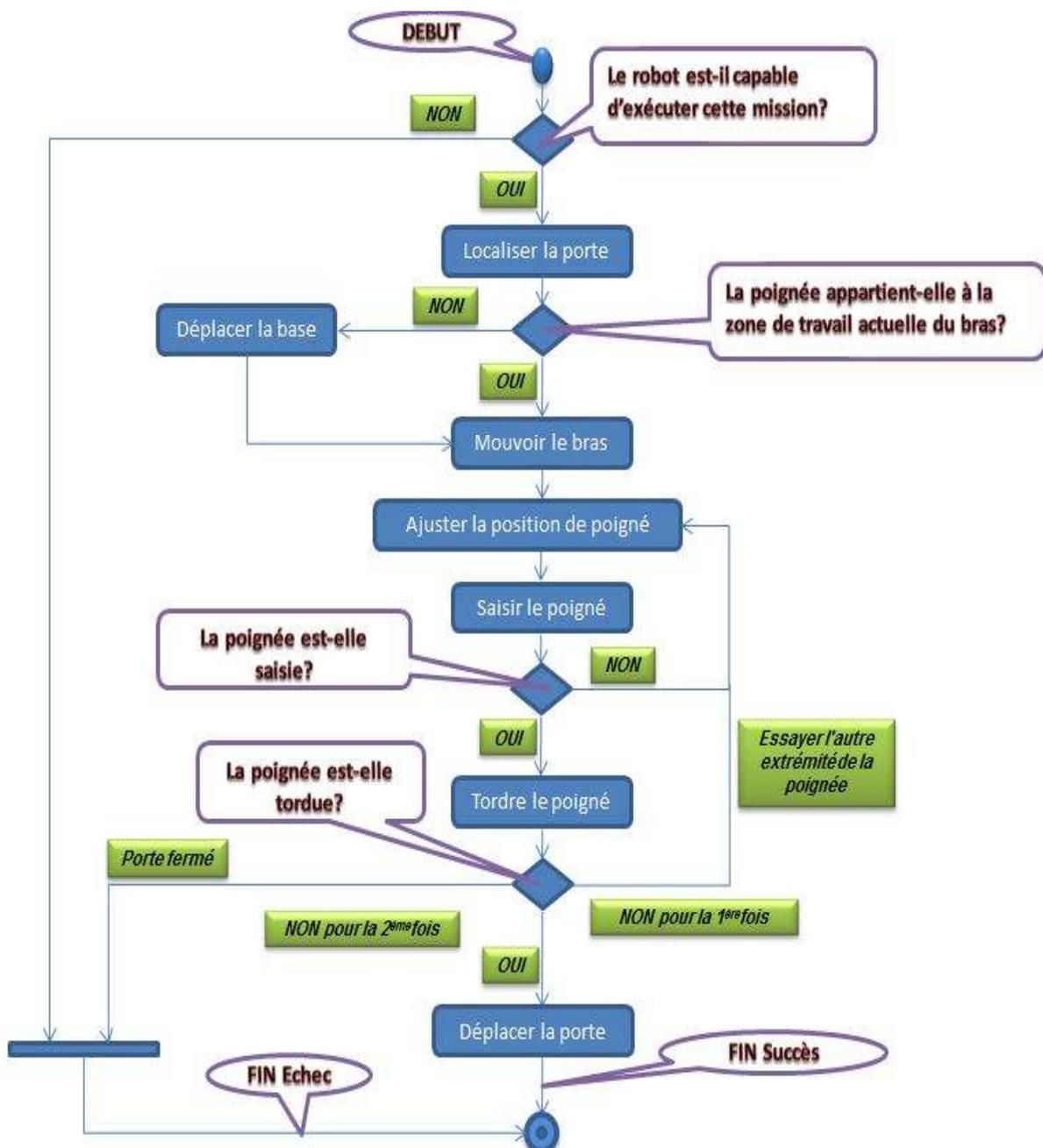


Figure.2. : Diagramme d'activité associé à la mission d'ouverture d'une porte.

### 3.1. Module de faisabilité

Ce module permet de tester si le PC on-board du robot est connecté avec le PC off-board. Si le robot n'est pas connecté, ce module lance un message d'indication « Echec, vous n'êtes pas connecté ». Dans le cas contraire, le module va tester si la poignée est atteignable. Si la poignée se trouve plus bas qu'une certaine distance  $x$  du sol ou plus haut qu'une distance  $x$  au sol, ce module déclare que cette mission est impossible à exécuter (Echec, Mission impossible). Ce module est donné comme suit :

```

bool feasibility(){
    if(!Connecter ( )){
        DisplayMessage ("Echec, Vous n'êtes pas connecté ") ;
        return false ;
    }
    if(! Atteignable (Location)){

```

```

        Message ("Echec, Mission impossible ");
        return false ;
    }
    return true ;
}

```

### 3.2. Module de Localisation

La localisation de la poignée de la porte est effectuée en utilisant le système de vision seulement. Ce système calcule les coordonnées réelles 3D de la poignée par rapport au repère absolu  $R_A$ . La procédure d'extraction de ces coordonnées est donnée comme suit :

À l'aide de caméra embarquée sur l'effecteur du manipulateur mobile, on capture deux images de la porte à partir de deux positions différentes (position droite et position gauche). Par la suite et grâce à l'architecture multi-agents, l'agent *VSA*, traite ces images et extrait les coordonnées  $(X_P, Y_P, Z_P)$  de la poignée par rapport au repère absolu  $R_A$ .

```

PositionLocalizeDoor (Poignée, Location){
    LoadFromDatabase (Camera_Matrix);
    LoadFromDatabase (Gripper_Matrix);
    LoadFromDatabase (Left_Capture);
    LoadFromDatabase (Right_Capture);
    ExchangeMessage (VSA, LARA, MOVEARMC (Left_Capture));
    MoveConfiguration (Left_Capture);
    ManipulatorAtConfiguration (Left_Capture){
        CaptureImage (Image);
        ParallelOperations ( ){
            ProcessImage (Image, u1, v1);
            ExchangeMessage (VSA, LARA, MOVEARMC(Right_Capture));
            MoveConfiguration (Right_Capture);
            ManipulatorAtConfiguration (Right_Capture){
                CaptureImage (Image);
                ProcessImage (Image, u2, v2);
            }
        }
    }
    Compute3D (u1, v1, u2, v2, XP, YP, ZP);
    Position = (XP, YP, ZP);
}

```

### 3.3. Module de Navigation

Deux cas différents peuvent être distingués pour la mission d'ouverture de la porte :

- *La porte appartient à l'espace de travail courant du robot* : la mission exige seulement le mouvement de manipulateur.
- *La porte n'appartient pas à l'espace de travail courant du robot* : la base mobile se déplace jusqu'à ce que la porte rentre dans la nouvelle zone de travail du manipulateur. Puis, le robot manœuvre la poignée avec son effecteur.

```

void Navigation (Position(xp, yp, zp})) {
    if(!BelongWorkspaceXYZ (Position)) {
        ExchangeMessage (LARA, LMRA, MOVEBASE);
        ComputeNewLocation (xp, yp, zp, xB, yB, θB);
        MoveBase (xB, yB, θB);
        MobileAtLocation (xB, yB, θB) {

```

```

        ComptNewXYZ(xp, yp, zp, xB, yB, θB, xNp, yNp, zNp);
    }
}

```

### 3.4. Module de Mouvement du bras

Ce module consiste à déplacer le bras manipulateur jusqu'à atteindre la poignée. Ici, l'agent *LARA* calcule la nouvelle position de poignée par rapport au repère  $R_M$  et envoie une requête à *RARA* pour effectuer le mouvement.

```

Void Mouvement (Target){
    MoveManipulatorXYZ (Target);
    ManipulatorAtLocationXYZ (Target); {
        ExchangeMessage (RARA, LARA, OK);
    }
}

```

### 3.5. Module d'Ajustement

L'ajustement consiste à placer l'effecteur du robot dans une position la plus proche et perpendiculaire à la poignée de la porte. Deux cas différents peuvent être distingués aussi pour ce module :

- *L'Ajustement manuel* : l'ajustement se fait par l'opérateur du robot. L'ajustement manuel permet de réaliser la mission d'ouverture en absence de la caméra.
- *Ajustement automatique* : l'ajustement est effectué en exploitant la caméra du robot. *VSA* extrait les coordonnées précise de la poignée, *LARA* et *RARA*, effectuent un travail conjointement afin d'ajuster la pince à cette position.

```

Void AjustPosition( ){
    LoadFromDatabase (Camera_Matrix);
    LoadFromDatabase (Gripper_Matrix);
    LoadFromDatabase (Left_Capture);
    LoadFromDatabase (Right_Capture);
    ExchangeMessage (VSA, LARA, MOVEARMC (Left_Capture));
    MoveConfiguration (Left_Capture);
    ManipulatorAtConfiguration (Left_Capture){
        CaptureImage (Image);
        ParallelOperations ( ){
            ProcessImage (Image, u1, v1);
            ExchangeMessage (VSA, LARA, MOVEARMC(Right_Capture));
            MoveConfiguration (Right_Capture);
            ManipulatorAtConfiguration (Right_Capture){
                CaptureImage (Image);
                ProcessImage (Image, u2, v2);
            }
        }
    }
    Compute3D (u1, v1, u2, v2, Xp, Yp, Zp);
    Position = (Xp, Yp, Zp);
    MoveManipulatorXYZ (Position);
}

```

### 3.6. Module de Saisie de la poignée

Après l'ajustement de la position de la pince, cette dernière est donc perpendiculaire à la poignée de porte. L'étape suivante consiste à ouvrir la pince du manipulateur, puis, à saisir la poignée. Dès que la pince saisisse la poignée, le robot ferme sa pince.

```

Void SaisiePoignée (){
    OpenGripper ();
    GripperState (Opened){
        MoveManipulatorXYZ (Target);
        ManipulatorAtLocationXYZ (Target);{
            CloseGripper ();
            GripperState (Closed){
                ExchangeMessage (LARA, SA, OK);
            }
        }
    }
}

```

### 3.7. Module Tourner la poignée

Cette opération est simple. Il suffit de faire tourner l'axe quatre ou cinq du bras. L'agent *LARA* calcule la nouvelle position désirée de l'axe à faire tourner et l'envoie à *RARA* pour exécution.

```

Void TournerPoignée () {
    MoveConfiguration (Q1... Q6) ;
    ExchangeMessage (RARA, LARA, OK);
    ManipulatorAtConfiguration (Q1... Q6) {
        ExchangeMessage (LARA, SA, OK);
    }
}

```

### 3.8. (Tirer / pousser) la porte

Deux cas différents peuvent être distingués aussi pour ce module selon l'état de la porte :

- *La porte s'ouvre à l'intérieur* : la mission exige seulement de pousser la porte jusqu'à ce qu'elle soit entièrement ouverte.
- *La porte s'ouvre à l'extérieur* : ici, le robot fait l'opération inverse et qui consiste à tirer la porte jusqu'à ce qu'elle soit entièrement ouverte.

Cette étape peut être réalisée par le manipulateur mobile de trois manières différentes :

- *En utilisant seulement la base mobile* : la base mobile va se déplacer et le bras reste à l'arrêt.
- *En utilisant le bras manipulateur* : dans ce cas, le bras doit effectuer le mouvement d'ouverture et la base reste immobile.
- *En utilisant les deux sous-systèmes en même temps* : ici, on combine entre le mouvement du bras et le mouvement de la base mobile pour réaliser l'ouverture de porte.

```

Void Open(){
    Case1:MoveBase (VR, VL);
    Case2: MoveManipulatorXYZ(x, y, z);
    Case3: ParallelOperations (){
        MoveBase (VR, VL);
        MoveConfiguration (Q1... Q6);
    }
}

```

## 4. Diagramme de séquence

Le diagramme de séquence est un moyen pour exprimer le comportement du système multi-agents. Son rôle principal est de montrer l'interaction entre les agents du système. Les interactions seront effectuées dans un ordre chronologique. Ce diagramme présente:

- Les agents nécessaires qui doivent travailler conjointement afin de réaliser la mission, à savoir, *SA*, *VSA*, *LMRA*, *LARA*, *RARA* et *RMRA*.
- Trois types de formes :
  - Losange : cette forme signifie que dans cette étape d'exécution de la mission, deux cas peuvent se présenter. Ici, le robot doit faire un test pour choisir le cas valide tout dépend de la situation où se trouve le robot.
  - Rectangle : il représente le plan d'opérations planifiées par chaque agent.
  - Les flèches : les flèches entre les agents représentent les messages de communication.

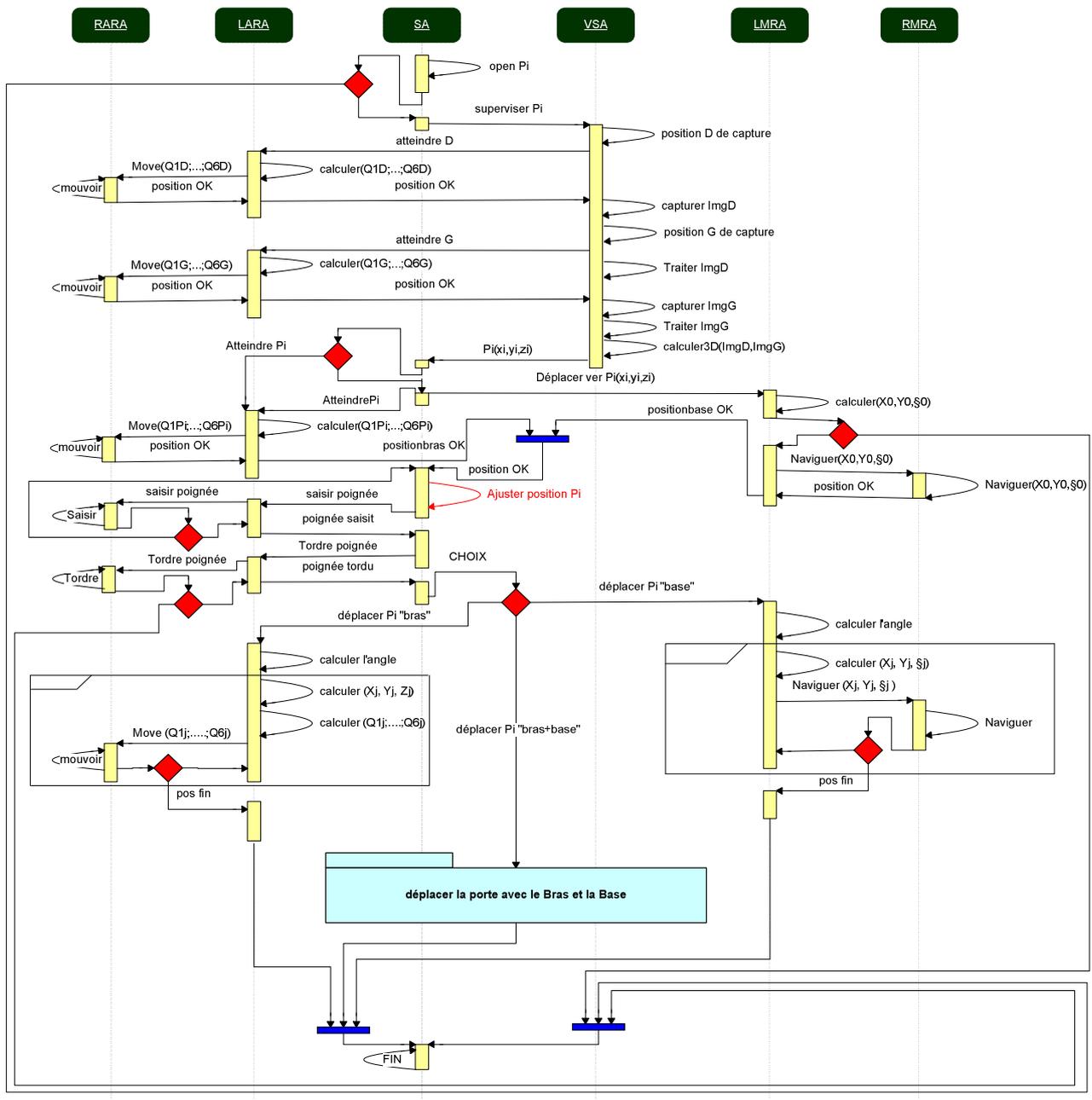


Figure.3. : Diagramme de séquence.

Le diagramme de la figure 3 montre le déroulement de la mission d'ouverture d'une porte par un manipulateur mobile contrôlé par une architecteur multi-agents. Quand l'opérateur du robot introduit la mission et lorsque l'agent *SA* la valide, il envoie à l'agent *VSA* une demande pour localiser la poignée de la porte et extraire ses coordonnées 3D. A la fin de cette opération, *VSA* envoie alors les coordonnées de la poignée à *SA*. Si la poignée est dans l'espace de travail actuel du bras, *SA* envoie une demande à *LARA* pour atteindre la poignée. Dans le cas contraire, *SA* envoie la demande à *LMRA* pour déplacer la base mobile. Quand le robot atteint la poignée, *SA* ajuste la position de la pince par rapport à la poignée et envoie la requête à *LARA* pour saisir la poignée. Si *LARA* réussit son opération, il envoie OK à *SA*. Sinon, il envoie une demande d'ajustement de la position à *SA*. Si *SA* reçoit OK, il envoie une autre demande à *LARA* pour torde la poignée. Ici, deux cas sont possibles :

- Si *LARA* n'arrive pas à faire tourne la poignée, il envoie à *SA* l'échec de la mission.
- Sinon, il envoie OK à *SA*.
- Pour la façon d'ouverture de la porte, si on choisit le bras, *SA* envoie une demande à *LARA* pour réaliser l'opération d'ouverture. Si on choisit la base, *SA* envoie la demande à *LMRA* pour déplacer la base mobile. Si on choisit les deux, le diagramme ci-dessous définit cette troisième façon d'ouverture (l'ouverture avec le bras et la base au même temps).

Ce diagramme présente le déroulement de la tâche d'ouverture avec la base et le bras. Ici, *SA* envoie une demande à *LMRA* pour suivre une trajectoire prédéfinie (sous forme d'un arc) de  $n$  points  $(p_1, p_2, \dots, p_n)$ . A chaque fois, *LMRA* calcule les nouvelles coordonnées de la base mobile qui se déplace pour atteindre la position désirée et envoie un requête *LARA* pour faire un mouvement selon la position actuelle de la base mobile. Si la position atteinte par *LMRA* n'est pas la position finale, on envoie une autre requête à *LMRA* pour se déplacer au point suivant. Cette procédure est répétée jusqu'à ce que la porte soit tirée.

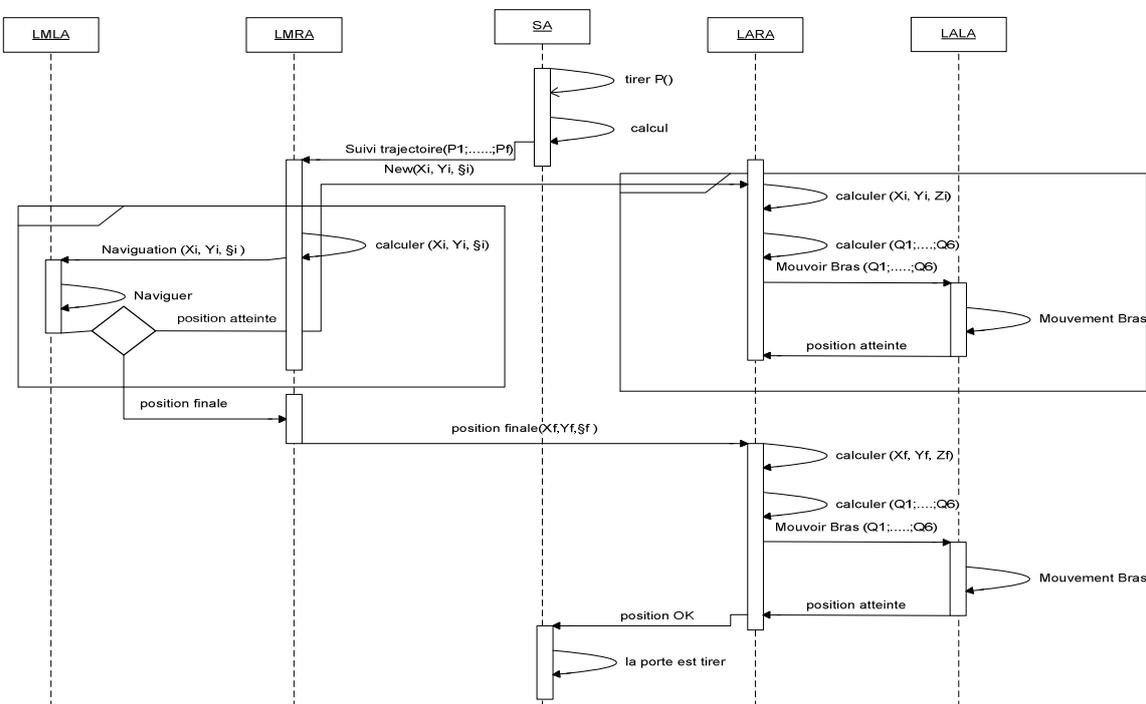


Figure.4. : déplacer la porte avec le Bras et la Base

## 5. Plan global d'exécution de la mission d'ouverture d'une porte

Cette mission consiste à ouvrir une porte qui se trouve à l'endroit  $(x, y, z)$  tout en évitant des obstacles possibles sur la trajectoire nominale du manipulateur mobile lors de la phase d'approche par la base mobile. L'étape nécessaire de cette mission consiste en la localisation de la poignée. Ici,

le robot doit calculer les coordonnées 3D réelle de la poignée de la porte en exploitant des données extraites à partir de deux images gauche et droite capturées par la caméra.

```
Opendoor(Location);{
    !Feasibility(Location);{
        DisplayMessage("Mission impossible");
        EndMission(Error);
    }
    LocalizeDoor (Poignée, Location) ;
    Navigation (Position, New_Position);
    Mouvement (Target);
    do{
        AjustPosition (Target) ;
        SaisiePoignée ( ) ;
    }While(GripperGrasping(Free));
    TournerPoignée ( ) ;
    if ( !ManipulatorAtConfiguration(Q1... Q6)){
        do{
            AjustPosition (Target) ;
            SaisiePoignée ( ) ;
        }While(GripperGrasping(Free));
        TournerPoignée ( ) ;
    }
    if ( !ManipulatorAtConfiguration(Q1... Q6)){
        DisplayMessage("closed door ");
        EndMission(Error);
    }
    ExchangeMessage(RARA, LARA, OK);
    Open ();
    ExchangeMessage(LARA, SA, OK);
    DisplayMessage("Mission carried out successfully");
    EndMission(Success);
}
```

## 6. Conclusion

Les techniques basées agents sont le modèle informatique idéal pour développer des logiciels ouverts et des systèmes distribués et connectés en réseau (comme l'Internet) ; les agents sont actifs et autonomes, donc ils sont des entités qui agissent pour atteindre des objectifs spécifiés.

Cette annexe présente l'architecture multi agents de contrôle de robot manipulateur mobile proposé pour la mission d'ouverture de porte.

## Bibliographie

---

- [1] K. Nait Chaabane, « Exploitation de la Redondance pour la Commande Coordinée d'un Manipulateur Mobile d'Assistance aux Personnes Handicapées », Thèse de doctorat, Université d'Évry Val d'Essonne, Novembre 2006.
- [2] Vincent. P., « Enchainements dynamiques de taches pour des manipulateurs mobiles à roues », Thèse de doctorat, Institut National Polytechnique de Toulouse, Novembre 2005.
- [3] Akli I., « Élaboration d'une stratégie de coordination de mouvements pour un manipulateur mobile redondant », Mémoire de magister, Université des Sciences et de la Technologie Houari Boumediene, Alger, juillet 2007.
- [4] Djekoune W., « Localisation et Guidage du Robot Mobile ATRV2 dans un Environnement Naturel », Thèse de doctorat, Université des Sciences et de la Technologie Houari Boumediene, Alger, Décembre 2010.
- [5] Kazed M., « Intelligence robotique », cours de master robotique
- [6] Johannes L, Gea Fernandez F. Design of an autonomous mobile dual-arm robot. 2011 IEEE International Conference on Robotics and Automation Shanghai International Conference Center. Mai 2011.
- [7] Bayle B., « modélisation et commande cinématiques des manipulateurs mobiles a roues » thèse de doctorat, LAAS-CNRS Université de Paul Sabatier, Décembre 2001
- [8] Colombaro R. rapport de stage « spécification, conception et réalisation d'une carte de gestion de sécurité pour robot bipède PID. juillet 2004.
- [9] Pajdla. T., « Omnidirectionnel Vision Course », Rapport technique, ICCV, 2003.
- [10] <http://www.humanoides.fr/2012/12/13/la-robotique-au-service-de-lagriculture/>

- [11]Filliat D., «Robotique Mobile-Cours», École Nationale Supérieure de Techniques Avancées (ENSTA), France, 2005.
- [12]<http://www.willowgarage.com/pages/pr2/overview>
- [13] : B.Bayle "Modélisation et Commande Cinématiques des Manipulateurs Mobiles à Roues" Thèse de Doctorat, France, 2001.
- [14]: Muir, P. F. and Neuman, C. P., Kinematic Modeling of Wheeled Mobile Robots, Journal of Robotic Systems, Vol. 4, 1987, pp. 281-340.
- [15] BojanNemec, "Robot Grasping using ArToolkit library" ,InstitutJožef Stefan, Department for Automatics, biocybernetics and robotics, Jamovacesta 39, 1000 Ljubljana, Slovenija
- [16] Jean-Louis Boimond "*cours robotique*" Université Angers.
- [17] L. Petersson, D. Austin. Kragic, "*High-Level Control Of A Mobile Manipulator For Door Opening* ", pp.s 2333-2338, 2000.
- [18] B.J.W. Waarsing ,M.Nuttin ,H. Van Brussel , "*Behavior-Based Mobile Manipulation : The Opening Of A Door*". Proceedings of the ASER 03 1st International Workshop On Advances In Service Robotics, Bardolino, Italy, Match 13-15, 2003.
- [19] AdvaitJain , Charles C. Kemp, "*Behaviors for Robust Door Opening and DoorwayTraversal with a Force-Sensing Mobile Manipulator*".
- [20] Christian Ott ,B.Baumel ,Ch.Borst .G.Hirzinger , "*Employing Cartesian Impedance Control For The Opening Of A Door :A Case Study In Mobile Manipulation*".
- [21] S. Chitta, B. Cohen, and M. Likhachev, "Planning for autonomous door opening with a mobile manipulator," in IEEE International Conference on Robotics and Automation (ICRA), 2010.
- [22] R. B. Rusu, W. Meeussen, S. Chitta, and M. Beetz, "Laser-based perception for door and handle identification," in International Conference on Advanced Robotics, 2009.
- [23] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, J. Hsu, R. B. Rusu, B. Marthi, G. Bradski, K. Konolige, B. Gerkey, and E. Berger, "Autonomous Door Opening and Plugging In using a Personal Robot," in International Conference on Robotics and Automation, 2010.
- [24] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with theOpenCV Library. O'Reilly Media, Inc., 2008, pp. 415–453.

- [25] Steven Gray, Sachin Chitta, Vijay Kumar, Maxim Likhachev, "A Single Planner for a Composite Task of Approaching, Opening and Navigating through Non-spring and Spring-loaded Doors", 2013
- [dh] W. Khalil, J. F. Kleininger. "A new Geometric Notation for Open and Closed-Loop Robots". Proceedings of the IEEE international conference on robotics and automation, USA, pp, 1175-1180, 1986.
- [26] Jacquet. W." Paradigme connexionniste pour l'apprentissage de fonctions robotique: application aux problèmes d'asservissement visuel de bras manipulateurs», thèse de doctorat, Université de haute-alsace U.F.R des sciences et techniques, Décembre 2001
- [27] CAUCHOIS.C. "Calibration du capteur de vision omnidirectionnelle SYCLOP", Université de Technologie de Compiègne, Août 1998.
- [28] Y.I.Abdel-Aziz et H.M.Karara, « Direct linear transformation into object space coordinates in close-range photogrammetry», dans Proc. Symp. Close Range Univ. De Illinois à Urbana-Champaign, Urbana, 1971, pp. 1-18
- [29] R.Y. Tsai, « A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shell TV cameras and lenses ». IEEE Journal of Robotics And Automation, 3(4): 323-344, Aug. 1987
- [30] Batista J. Iterative Multi-Step Explicit Camera Calibration – Computer Vision (ICCV98), Bombay, India, Janvier 1998.
- [31] B Paul, D Murray A Zisserman "Camera Calibration Using Multiple Images," Confirance européen. Comp. Vision 1992, pp. 312--320.
- [32] M Wilczkowiak, E Boyer P Sturm. "Calibrage de caméra et reconstruction 3D à l'aide de parallélépipèdes", rapport de stage. France 2002.
- [33] F Girard. "Autocalibrage de séquences d'images thoracopiques". ÉCOLE POLYTECHNIQUE DE MONTRÉAL .Fevrier 2009.
- [34] T Skordas. "Une revue des derniers progress en autocalibration de camera CCd – Traitement du signal – Vol 12 n°1 - 1995
- [35] R.I.Hartley, « Self-calibration from multiple views with a rotating camera », dans J.-O Eklundh, editor, Proceeding of the 3rd European Conference on Computer Vision, volume 800-801 de Lecture Notes in Computer Sciences, pages 471-478, Stockholm, Suède, Mai 1994, Springer-Verlag

- [36] Z. Zhang, "A flexible new technique for camera calibration", IEEE Trans Pattern Anal. Mach. Intell, vol. 11, 2000, pp.1330-1334.
- [37] Billingham, M., Kato, H., Poupyrev, I. (2001) The MagicBook: A Transitional AR Interface. Computers and Graphics, November 2001, pp. 745-753.
- [38] Philip Lamb. Artoolkit featurelist.Web :  
<http://www.hitl.washington.edu/artoolkit/documentation/features.htm>.
- [39] Bojan Nemeč, "Robot Grasping using ArToolKit library", Institut Jožef Stefan, Department for Automatics, biocybernetics and robotics, Jamovcesta 39, 1000 Ljubljana, Slovenija
- [40] David COCHARD, "ARToolKit Plugin for Virtools 4", utc 2007.
- [41] Gustavo Rovelo "WGM # 18 Artoolkit II"  
<https://jira.ai2.upv.es/confluence/display/LOTO/WGM>
- [42] Y.C. Shiu et S. Ahmad, « Calibration of wrist-mounted robotics sensors by solving homogeneous transformation equations of the form  $AX=XB$  », IEEE Transactions on robotics and automation, 5(1):16-29, (pp 21, 22), février 1989.
- [43] F.C. Park et B.J. Martin, « Robot sensor calibration: Solving  $AX=XB$  on the euclidean group », IEEE Trans. On robotics and automation, 10(5):717-721, octobre 1994
- [44] M. Li et D. Betsis, « Hand-eye calibration », dans International conference on computer vision, pages 40-45, 1995.
- [45] K. Daniilidis et E. Bayro-Corrochano, « The dual quaternion approach to hand-eye calibration », dans Proc. IAPR International conference on pattern recognition, pages 318-322, 1996.
- [46] D. Kim, « Calibration problems in robotics », thèse de PhD, université de Michigan, 1996.
- [47] R. Horaud et F. Dornaika, « Hand-eye calibration », International journal of robotics research, 14(3):195-210, juin 1995
- [48] H. Zhuang et Z. Qu, « A new identification jacobian for robotic hand/eye calibration », IEEE Trans. On systems, man and cybernetics, 24(8):1284-1287, août 1994.
- [49] S. Rémy, « Etalonnage d'un système de vision embarqué », Thèse de doctorat, université Blaise-Pascal, Clermont II, juillet 1998.
- [50] F. Dornaika, « Contributions à l'intégration vision/robotique : Calibrage, localisation et asservissement », thèse de doctorat de l'institut national polytechnique de Grenoble, septembre 1995

- [51] R. Horaud, F. Dornaika, B. Boufama et R. Mohr, « Self calibration of a stereoheadmounted onto a robot arm », dans Proc. ThirdEuropeanconference on computer vision, pages 455-462, Stockholm, Suède, mai 1994
- [52] S. Lee et S. Ro, « A self-calibration model for hand-eye systems with motion estimation », *Mathl. Comput. Modelling*, 24(5/6):49-77, 1996
- [53] N. Andreff, « Asservissement visuel à partir de droites et auto-étalonnage pince-caméra », thèse de doctorat préparée au laboratoire INRIA Rhône-Alpes, 29 novembre 1999.
- [54] RAHICHE Abderrahmane, « Fusion Multi-sensorielle Et Télé-opération : Application A Un Robot Mobile Manipulateur », thèse de magister, Blida, Octobre 2007.
- [55] Breton.S, « Une approche neuronale du control robotique utilisant la vision binoculaire par reconstruction tridimensionnelle », thèse de doctorat, université de Haute Alsace, U.F.R. des sciences et techniques, Spécialité Electronique Electrotechnique & Automatique, France, juin 1999, page 24-27.
- [56] A.Hentout, B. Bouzouia, A. Toumi, Z.Toukal, « Agent-Based Coordinated Control of Mobile Manipulator, The international Conference on Systems and Processing information (ICSIP'09), Algeria, 2009.
- [57]. Guerineau. N, Moignard. C et Pomiers. P, « RobuTER et Bras Ultra Leger: Manuel d'utilisation et de maintenance », Version.1, Robosoft, France, 1999.
- [58] W. Khalil, J. F. Kleinfinger. "A new Geometric Notation for Open and Closed-Loop Robots". *Proceedings of the IEEE international conference on robotics and automation*, USA, pp, 1175-1180, 1986.
- [59] S.Borsali, « cours Modelisation Des Robots », Master Automatique S2 2012, 45 pages.
- [60]. Renaud. M & Gorla. B, « Modeles des robots manipulateurs en vue de leur commande », IBID, S.E.E. (Societe des electriciens et des radioelectriciens), 1984.
- [61] B.Bayle, J.-Y. Fourquet, M. Renaud, « Génération des mouvements des Manipulateurs Mobiles : Etat de l'art et perspectives ». *Journal Européen des Systèmes Automatisés*, vol 35, 809-845, 2001.
- [62]. Ruben. S, Garcia. R, « Programmation bayesienne des bras manipulateurs », these doctorat, institut national polytechnique de Grenoble, France, p.165, 2003.
- [63] <http://www.hitl.washington.edu/artoolkit/documentation/usercalibration.htm>  
25.05.2013.

