

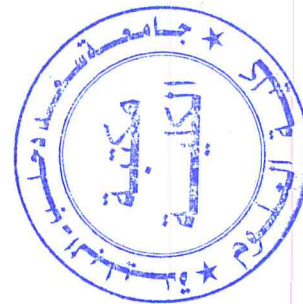
MA-004-189-1

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE**

Université SAAD DAHLEB de BLIDA

Faculté des Sciences

Département d'Informatique



Mémoire de fin d'études

Pour l'obtention du diplôme de master en informatique

Option génie logiciel

Thème :

***EXÉCUTION D'UNE TÂCHE DE SAISIE D'UN
OBJET PAR LE MANIPULATEUR MOBILE
ROBUTER/ULM.***

Présenté par :

Mr Melzi abdelwahab

Mr Guerroumi ismail

Promotrice :

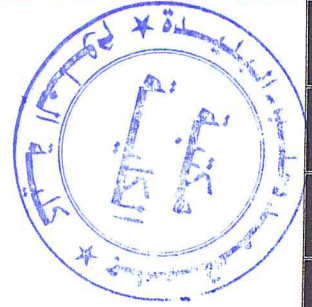
Meme Isma akli

Co-Promoteur :

Mr mohamed ridha sidoumo

Promotion: 2012/2013

MA-004-189-1



Remerciement

Nous remercions le DIEU tous puissant de nous donner le courage, la santé et la volanté de nous avoir permis de réaliser ce travail dans des bonnes condition.

Nous remercions nos parents qui nous ont aidés, conseillé et assisté tout au long de notre formation.

Nous remercions notre promoteurs **I.Akli** et **M.Sidomou** pour son aide précieuse, et pour le suivi qu'il nous a prodigué au cour de ce projet.

Et enfin, nous remercions tous les enseignants qui ont contribués ont notre formation et aussi à touts qui nous a aide de prés ou de loin.

Dédicace

Je dédie ce travail humble

Avant tout à la plus chère et plus doux que j'ai dans ce monde

Pour ma chère mère, et c'est pourquoi je suis ici aujourd'hui, et à tous mes

frères et sœurs; et les remercier pour leur soutien financier et moral.

Et je remercie tous ceux qui ont contribué à ce travail de près ou de loin.

Aussi dédie ce travail à quelqu'un qui sait ismail.

Et bien sûr à mon binôme abdelwahab.

Ismail

Dédicace

Je dédie ce modeste travail à :

- ❖ *Ma mère, mon père, sans eux, je n'aurais pas abouti à ce stade d'étude, que Dieu puisse m'aider à les honorer, les servir et les combler.*
- ❖ *A mes frères et mes sœur.*
- ❖ *A mon chère binôme ismail et sa famille.*
- ❖ *A Tous mes amis, en particulier (Elhadhje, Redouane, Mohamed, Sid Ahmed, Homi, salim, Elmir, Abdelkader, Zoubir, Nacer, Kadar, Hossin et son binôme didine, Ali, Hossem, Zino, Islem, Abdelwahab, Noureddine, Imed , Brahim ,...)*
- ❖ *A toute la section TRT8, en particulier (Boula7ya, Ilham, hamid, alal,)*

Abdelwahab

Sommaire

Introduction générale	1
Chapitre I : généralités sur la robotique de manipulation mobile	
1. Introduction.....	3
2. Le système robotique.....	3
2.1. Les capteurs.....	3
3. Bras manipulateur.....	4
3.1 Les fonctionnalités des bras manipulateur	4
4. Robots mobiles	5
4.1 Définition des robots mobiles.....	5
4.2 Définition des robots mobiles autonome.....	5
4.3 Les différents types des robots mobiles.....	5
4.3.1 Les robots holonomes.....	5
4.3.2 Les robots différentiels.....	5
4.3.3 Les robots omnidirectionnelles.....	6
4.3.4 Les robots non holonomes.....	7
4.3.5 Les robots à pattes.....	8
5. Manipulateur mobile.....	8
5.1 Les types des manipulateurs mobiles.....	9
5.1.1 Les multipèdes	9
5.1.2 Les manipulateurs mobiles sous-marins.....	10
5.1.3 Les manipulateurs mobiles à roues.....	10
5.2 Les propriétés des manipulateurs mobiles.....	10
5.3 Domaines d'applications	11
5.3.1 Domaine spatial.....	11
5.3.2 Domaine de service.....	12
5.3.3 Domaine militaire.....	12
5.3.4 Domaine médical.....	13

6. Conclusion	13
---------------------	----

Chapitre II : Description du problème

1. Introduction	14
2. Etat de l'art de travaux relatifs à l'exécution de tâches pour des robots mobiles manipulateurs	14
3. Les approches principales de l'étude du comportement d'un système robotique....	17
3.1. Approche Délibérative - Penser et puis agir.....	17
3.2 Approche Réactive - Ne pas penser, (ré) agir.....	17
3.3 Approche Hybride - Penser et agir concurremment	18
3.4 Approche Contrôle à base de comportement - Penser de façon vous réagissez..	18
4. Le Système multi-agents existant..	18
4.1 Architecture physique	18
4.2 Architecture logicielle	19
4.3 Structure des messages échangés dans le système	21
5. Le contrôle du robot	21
5.1 Mouvement du bras manipulateur...	21
5.2 Déplacement de la base mobile.....	21
6. Formulation du problème de saisie d'un objet	22
6.1 Description du problème lié à la saisie d'un objet par la pince du bras.....	22
6.2 Description du problème lié au déplacement de la plateforme mobile.....	22
8. Conclusion.....	23

Chapitre III : calibration de caméras avec utilisation de marqueurs

1. Introduction.....	24
2. Librairie ARTToolkit	24
2.1. Utilisation de la librairie ARTToolkit.....	25
2.1.1 Algorithme de détection du capteur.....	25
2.1.2 Principe de fonctionnement.....	26
3. Calibration de caméras.....	27
3.1 Les méthodes classiques de calibration.....	27
3.1.1 Calibrage photogrammétrique.....	27
3.1.2 Auto-calibrage.....	27

3.1.3	Calibrage d'une rotation pure.....	28
3.1.4	Technique de Zhang.....	28
4.	Les résultats obtenus	28
4.1	Calibration avec la librairie ARToolkit.....	28
4.1.1	Positionnement de la camera	28
4.1.1.2	Processus expérimental de calibrage	29
4.1.1.3	Reconnaissance de marqueur	30
5.	conclusion	30

Chapitre IV : Exploitation de l'architecture multi-agents existante

1.	Introduction	31
2.	Scénario d'exécution la tâche étudiée par un manipulateur mobile.....	31
3.	conception et Modélisation.....	31
3.1	Diagramme de séquences	31
3.2	Diagramme d'activité	34
3.2.1	Digramme d'activité pour la tâche de saisie.....	34
3.2.2	Digramme d'activité pour la localisation d'objet.....	35
3.2.3	Digramme d'activité pour le déplacement de la base.....	35
4.	Déroulement de la tâche de saisie par le manipulateur mobile.....	36
5.	L'exécution de la tâche de saisie par le bras manipulateur.....	37
5.1	Passage de repère bras au repère pince	38
5.2	Passage de repère pince au repère caméra.....	39
5.3	Passage de repère caméra au repère objet	40
5.3.1	Passage de repère caméra au repère marqueur	40
5.3.2	Passage de repère marqueur au repère objet	41
5.4	Calculer la position d'objet dans le repère bras	41
5.5	Déplacement de la base mobile	42
6.	Conclusion	43

Chapitre V : Résultats Expérimentaux

1.	Introduction.....	44
2.	Environnement matériel	44
3.	Environnement logiciel	44
3.1	L'interface graphique	45

4. Les résultats obtenus.....	47
4.1 Les résultats de mouvement de plateforme.....	47
4.2 Résultat de saisie.....	48
4. Conclusion.....	50
Conclusion générale.....	51

Bibliographie

Annexe A

Annexe B

Annexe C

Liste des figures

- Figure 1.1:** srx10 (4axes) Robot FANUC LRMate 200i, 6ddl.
- Figure 1.2 :** Divers structures des robots manipulateurs.
- Figure 1.3:** Exemple d'un robot holonomie.
- Figure 1.4 :** Exemple de robot différentiel. Pioneer 2 DX de la société MobileRobots. Urban Robot de la société iRobot.
- Figure 1.5 :** Exemple de robot omnidirectionnelle à roues orientables.
- Figure 1.6:** Exemple de robot omnidirectionnelle à roues suédoises.
- Figure 1.7 :** Exemple de robot non holonome.
- Figure 1.8 :** Exemples de robots à pattes. Hexapode de AAI Canada
- Figure.1.9 :** Exemple des robots multipèdes.
- Figure 1. 10 :** Robot ROV3.
- Figure 1.11:** lynxmotion-aluminum-a4wd1-aut.
- Figure 1.12 :** Exemple d'un manipulateur mobile de spatiale.
- Figure 1.13 :** asimo-honda.
- Figure 1.14 :** Robot camp.
- Figure 1.15 :** Robot DERMAROB.
- Figure 2.1 :** Détection de la porte avec l'une des caméras stéréo par ARToolKit.
- Figure 2.2 :** le système mobile pour la saisie.
- Figure 2.3 :** Saisie d'une balle pendant la marche.
- Figure 2.4 :** Les éléments de la base mobile du robot SAM.
- Figure 2.5:** Expérimentations sur HRP2.
- Figure 2.6:** Architecteur physique de système.
- Figure 2.7 :** Modèle de l'architecture de contrôle et de supervision proposée.
- Figure 2.8 :** Structure d'un message.
- Figure 2.9 :** description de problème de localisation d'objet.
- Figure 2.10 :** description de problème de déplacement de la base mobile.
- Figure 3.1 :** Exemples de marqueurs.
- Figure 3.2:** Algorithme de détection du capteur.
- Figure 3.3 :** la phase de binarisation.
- Figure 3.4:** la phase d'identification de marqueur.
- Figure 3.5 :** caractéristiques de la camera.
- Figure 4.1 :** Diagramme de séquences de la mission de saisie d'un objet.
- Figure 4.2 :** Digramme d'activité pour la tâche de saisie

Liste des figures

Figure 4.3 : Digramme d'activité pour la localisation d'objet.

Figure 4.4: Digramme d'activité pour le déplacement de la base.

Figure 4.5 : description de problème de localisation d'objet.

Tableau 4.1 : *les paramètres de DH modifiés.*

Tableau 4.2 : *les paramètres de DH concernant la caméra.*

Tableau 4.3 : *les paramètres de DH concernant l'objet.*

Figure 4.6 : description de problème de déplacement de la base mobile.

Figure 5.1 : architecteur physique du système.

Figure 5.2 : Interface opérateur pour le contrôle du système robotisé

Figure 5.3 : Interfaces dédiées au Contrôle du Robot

Figure 5.4 : Interface opérateur pour le contrôle du système robotisé.

Figure 5.5 : La partie concernant la tâche de déplacement.

Figure 5.6 : état initiale de robot.

Figure 5.7 : état intermédiaire de robot.

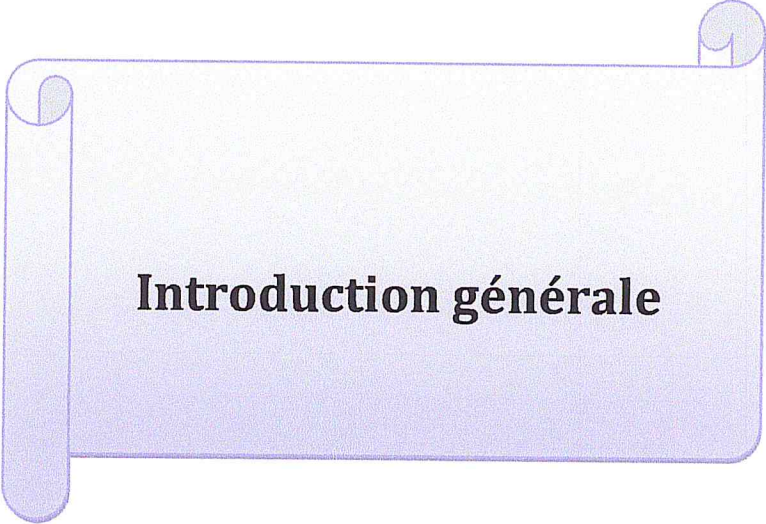
Figure 5.8 : état finale de robot.

Figure 5.9 : La partie concernant la tâche de saisie.

Figure 5.10 : la tâche de saisie.

Figure 5.11 : les étapes pour saisir un objet.

Figure 5.12 : la position finale de la pince



Introduction générale

Introduction générale

Le but principal des chercheurs en robotique est de concevoir un robot intelligent, sachant accomplir une mission complexe de manière autonome. Les premiers robots à avoir été conçus furent les bras manipulateurs, dont l'utilisation dans l'industrie s'est peu à peu banalisée.

Cependant, toute manipulation est limitée par les dimensions et par le nombre de degrés de liberté (ddl) de ces robots. Ainsi, ils se trouvent confinés à des espaces structurés, avec, le plus souvent, de faibles capacités de travail. C'est la raison pour laquelle sont apparues les plateformes mobiles, caractérisées par leur aptitude à évoluer dans des environnements de grande taille, en regard de leurs dimensions. En vue d'applications industrielles, ces plateformes mobiles ont dû être dotées d'outils divers (nettoyeurs de métro, exploration ou surveillance grâce à des caméras, etc.).

Pour les missions en milieux hostiles, spatiales, ou qui requièrent, tout simplement, des capacités combinées de la locomotion et de la manipulation, ces plateformes ont dû être dotées d'un bras manipulateur pour devenir des manipulateurs mobiles. Il faut noter que l'étude effective des systèmes mécaniques que constituent les manipulateurs mobiles est relativement récente.

La conjonction de ces deux fonctions (locomotion et manipulation) démultiplie les possibilités de ces fonctions séparées. L'association d'un bras manipulateur et d'une plateforme mobile bénéficie de la complémentarité des deux systèmes mécaniques qu'ils constituent :

Au niveau de l'espace de travail : tout d'abord, un bras manipulateur associé à une plateforme mobile augmente les possibilités de franchissement d'obstacles (saisir des objets, déplacement des obstacles, etc.). La plateforme mobile permet au bras manipulateur d'étendre son espace de travail, jusqu'alors restreint.

Au niveau de la nature des tâches pouvant être effectuées : un bras manipulateur permet de diversifier les travaux possibles, généralement limités au simple transport pour une plateforme mobile. Le manipulateur mobile, en procurant des ddl supplémentaires, augmente les possibilités de manipulation du bras manipulateur.

Le passage d'un bras manipulateur à base fixe à un manipulateur mobile pouvant explorer un espace de travail plus grand offre ainsi de nouvelles perspectives. On peut envisager des applications où le bras manipulateur est amené par sa plateforme mobile sur le site de travail, ou, plus intéressant encore, des missions nécessitant la coordination des deux systèmes qu'ils constituent. L'association plateforme mobile/bras manipulateur ouvre, enfin, de nouveaux horizons en matière de perception. Le système mécanique constitué permet, en effet, une meilleure exploration de l'espace de travail, et donc sa reconnaissance à l'aide de capteurs notamment visuels. Par exemple, des caméras peuvent être orientées aisément si elles sont montées sur le bras manipulateur.

Introduction générale


Les manipulateurs mobiles permettent d'accomplir la plupart des missions usuelles de la robotique qui requièrent, en même temps, des capacités de locomotion et de manipulation. Ces robots ont des applications dans plusieurs domaines tels que le domaine manufacturier pour accomplir des tâches de saisie et de transport d'objets, ou pour accomplir des missions de manutention, dans le domaine militaire pour le déminage, etc. Récemment, les environnements cibles d'applications de tels robots ont migrés des environnements industriels vers les environnements humains. 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 la saisie des objets qui fait l'objet de ce travail.

L'objectif de ce mémoire est la conception, d'une mission compliqué qui consiste à faire saisir un objet par un manipulateur mobile. L'architecture de contrôle de ce dernier est une architecture multi-agents.

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.

Outre cette introduction générale et une conclusion générale, ce mémoire est composé de cinq chapitres et une annexe donnés comme suit :

- Le chapitre 1 présente les manipulateurs mobiles. il donne les divers types des manipulateurs mobiles existant dans le monde, leurs caractéristiques et leurs principaux domaines d'applications.
- La formulation de problème de saisie 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 présente les différents outils et méthodes que nous doit suivre pour résoudre nos problèmes et les différentes bibliothèques permettant la localisation d'un capteur visuel (Camera) par rapport à un repère défini. Le chapitre présente de plus près à la bibliothèque ARToolkit, que nous avons utilisé dans notre application.
- La première partie du chapitre 4, présente le concept des agents, leurs caractéristiques, leurs architectures et leurs différents types. La seconde partie du chapitre donne une présentation détaillée des systèmes multi-agents (SMA), l'interaction (la coopération et la communication) dans les SMA et les différentes méthodes de coopération. Nous terminons ce chapitre par une conception et une implémentation de notre problème.
- Des tests expérimentaux sont présentés et discutés dans le dernier chapitre en vue de la validation de l'application développée.
- La modélisation géométrique directe et inverse du manipulateur ULM est donnée en Annexe. Cette annexe donne aussi le modèle cinématique de la base mobile différentielle et non-holonome RobuTER.



Chapitre I :
Généralités sur la robotique de
manipulation mobile

1. Introduction :

Afin de situer notre travail dans la littérature existante, nous proposons un rappel des études menées dans le cadre de la manipulation mobile selon un classement thématique. Nous commençons par un très court rappel afin de donner un descriptif des deux sous-systèmes constituant les robots étudiés. En effet, nous allons commencer par présenter les manipulateurs à base fixe, ensuite, les robots à base mobile, et enfin nous présenterons robots manipulateurs mobiles.

2. Le système robotique :

On définit généralement un système robotique comme étant un système artificiel ou une machine programmable disposant de moyens de perception, d'interprétation et d'action qui lui permettent d'agir sur un environnement physique. La robotique s'est développée depuis une vingtaine d'années principalement dans la production manufacturière sous forme de bras manipulateurs destinés à des tâches de soudure, de peinture, et de manutention. Cette robotique, dite industrielle, a permis d'accroître de manière importante la productivité et la flexibilité des ateliers de production en soulageant l'homme de travaux pénibles.

Les systèmes robotiques se composent de sous-systèmes mécaniques, capteurs, actionneurs, régulateur, et d'un sous-système de traitement de l'information. En plus, ces sous-systèmes peuvent communiquer entre eux via des interfaces dont la fonction consiste à décoder l'information transmise. Un système robotique, quel qu'il soit, se doit de déterminer et de réaliser les mouvements à accomplir pour réaliser une tâche donnée [4].

2.1 Les capteurs :

Dans un système autonome, il est nécessaire de connaître en temps réel toutes les caractéristiques liées à la fois à la nature de l'environnement (état des surfaces, inclinaison du sol, présence ou non d'obstacles, ...) et à celle du robot (coordonnées, position de tous les segments, inclinaison par rapport au sol, accélération, ...).

Les capteurs sont capables de fournir dans des temps assez courts des informations très importantes sur l'état des lieux, et l'état du robot.

Les capteurs sont des composants de la chaîne d'acquisition dans une chaîne fonctionnelle. Ils prélèvent une information sur le comportement de la partie opérative et la transforme en une information exploitable par la partie commande.

On peut caractériser les capteurs selon deux critères [38]:

Placés sur le robot ils seront de deux natures différentes, soit ils informent le robot sur la nature de l'environnement, alors ce seront des capteurs *extéroceptifs* (télémètre laser, les ultrasons, camera), soit sur son état interne, alors ce seront des capteurs *proprioceptifs* (vitesse des roues, l'angle des articulations, la charge des batteries...) [37].

Nous allons présenter dans ce qui suit les types de robots les plus couramment utilisés.

3. Bras manipulateur :

Un bras manipulateur est le bras d'un robot généralement programmable, avec des fonctions similaires à un bras humain. Les liens de ce manipulateur sont reliés par des axes permettant, soit du mouvement de rotation ou de translation (linéaire) de déplacement.

Il peut être autonome ou contrôlé manuellement et peut être utilisé pour effectuer une variété de tâches avec une grande précision [1].

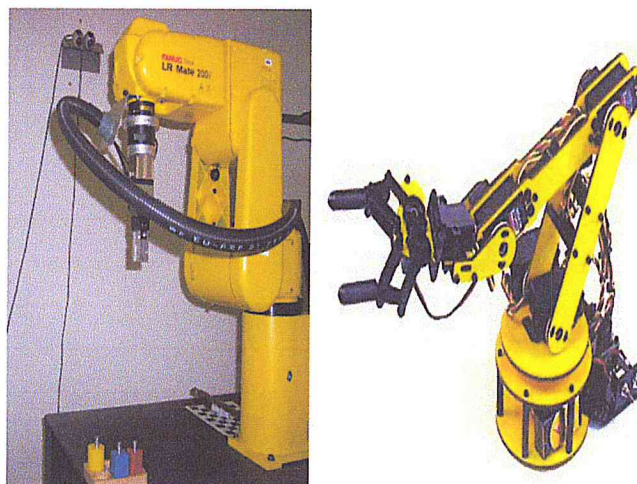
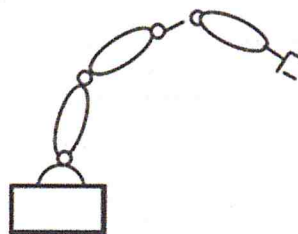


Figure 1.1: (a) srx10 (FANUC), (b) LRMate 200i

3.1 Les fonctionnalités des bras manipulateur :

Un robot manipulateur est constitué de deux sous-ensembles distincts [26]:

- *Un organe terminal* : c'est un dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique, il regroupe les procédés destinés à manipuler des objets, ou à les transformer, il s'agit donc d'une interface permettant au robot d'interagir avec son environnement [12].
- *Une structure mécanique articulée* : architecture composée de plusieurs chaînes de corps rigides assemblés par des liaisons appelées articulations. Les chaînes peuvent être soit ouvertes (ou en série), arborescentes ou fermées.



Structure série.

Figure 1.2 : Divers structures des robots manipulateurs.

4. Robots mobiles :

4.1 Définition des robots mobiles :

De manière générale, on regroupe sous l'appellation robots mobiles l'ensemble des robots à base mobile, par opposition notamment aux robots manipulateurs. L'usage veut néanmoins que l'on désigne le plus souvent par ce terme les robots mobiles à roues. Les autres robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheurs, sous-marins ou aériens [3].

4.2 Définition des robots mobiles autonome :

Les Robots Mobiles Autonomes (RMA) sont nés avec la robotique dite de troisième génération et sont donc des machines capables avant tout de réaliser le lien entre perception et action. Par définition le comportement d'un RMA n'est pas explicitement programmé à l'avance : il doit pouvoir se reconfigurer au gré des tâches à effectuer et s'adapter en temps réel à toute modification imprévue du monde qui l'entoure. Ce qui paraît caractériser le plus un RMA c'est son pouvoir d'interaction avec l'environnement très complexe dans lequel il évolue, ainsi que son haut degré d'autonomie opérationnelle et décisionnelle [6].

4.3 Les différents types des robots mobiles:

4.3.1 Les robots holonomes:

La capacité d'un robot mobile à pouvoir se déplacer à partir d'une situation dans n'importe quelle direction est appelée holonomie. En effet, le mécanisme holonome permet au robot de manœuvrer dans n'importe quelle direction arbitraire à partir de n'importe quelle configuration arbitraire. Il n'y a pas de contrainte sur les commandes. Cela signifie une très grande simplification des problèmes de contrôle [6].

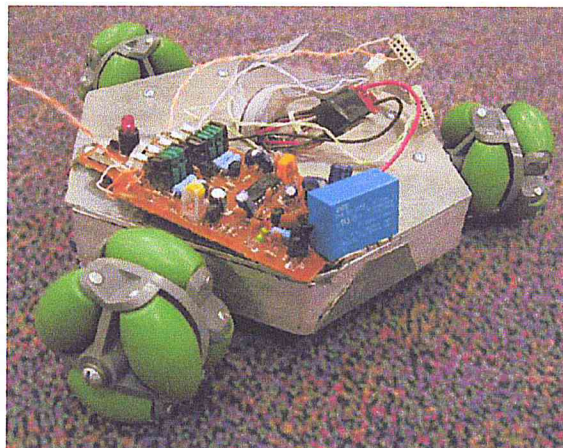


Figure 1.3: Exemple d'un robot holonomie.

4.3.2 Les robots différentiels :

Une des configurations les plus utilisées pour les robots mobiles d'intérieur est la configuration différentielle qui comporte deux roues commandées indépendamment. Une ou plusieurs roues folles sont ajoutées à l'avant ou à l'arrière du robot pour assurer sa stabilité (Figure 1.4). Cette plate-forme est très simple à commander, puisqu'il suffit de

spécifier les vitesses des deux roues, et permet de plus au robot de tourner sur place. Cette possibilité permet de traiter dans certains cas le robot comme un robot holonome, ce qui va simplifier la planification de déplacement et la commande du robot [5].

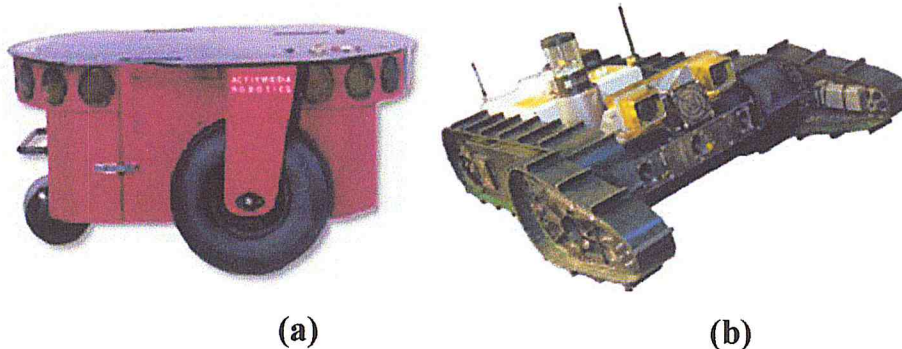


Figure 1.4 : Exemple de robot différentiel. (a) Pioneer 2 DX .
(b)Urban Robot.

4.3.3 Les robots omnidirectionnelles

Les plates-formes omnidirectionnelles permettent de découpler de manière plus nette le contrôle de la rotation et de la translation d'un robot et sont donc quasiment holonomes.

Il existe différents types de plateformes omnidirectionnelles. Le premier utilise trois ou quatre roues qui tournent à la même vitesse pour fournir une translation et un mécanisme qui permet d'orienter simultanément ces roues dans la direction du déplacement souhaitée (Figure 1.5).

Le corps du robot lui-même n'effectue pas de rotation mais uniquement des translations. Ce système permet un contrôle très simple et relativement rapide car les changements de direction ne concernent que les roues et peuvent donc se faire très vite. Par contre ces plates-formes sont relativement limitées en capacité de franchissement et requièrent un sol très plan.

Une deuxième catégorie de plateformes utilise des roues dites "suédoises", qui n'offrent pas de résistance au déplacement latéral (Figure 1.6). La plateforme comporte trois roues dont les axes sont fixes. Les déplacements dans toutes les directions et en rotation sont obtenus en faisant varier individuellement les vitesses des roues. La plateforme tourne sur place lorsque les trois roues tournent dans le même sens, à la même vitesse. Lorsqu'une roue est fixe, et que les deux autres tournent en sens opposé, la plateforme avance en direction de la roue fixe. Différentes combinaisons de vitesses permettent d'obtenir des déplacements quelconques. [5]

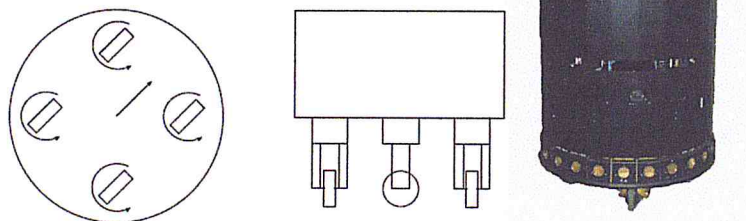


Figure 1.5 : Exemple de robot omnidirectionnelle à roues orientables [5].

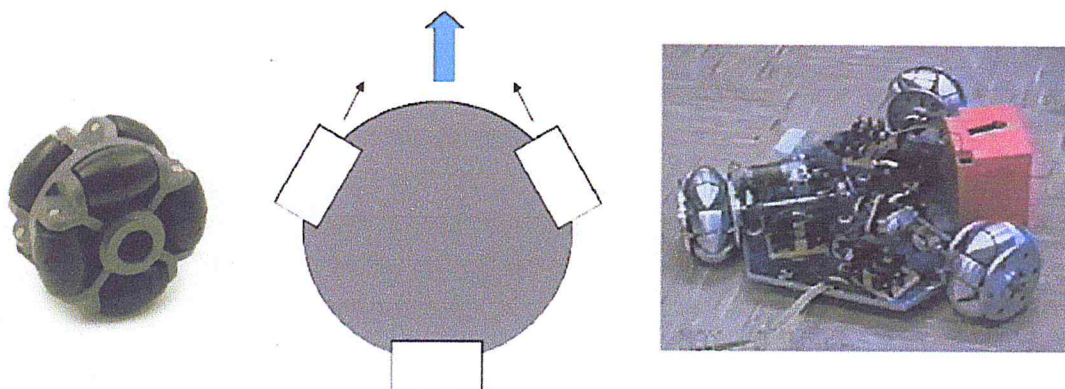


Figure 1.6: Exemple de robot omnidirectionnelle à roues suédoises [5].

4.3.4 Les robots non holonomes :

Des plates-formes non holonomes, telles que les voitures, sont également utilisées en robotique mobile (Figure 1.7). C'est plus particulièrement le cas dans le domaine des véhicules intelligents. Ces plates-formes sont toutefois plus difficiles à commander car elles ne peuvent pas tourner sur place et doivent manœuvrer, ce qui peut être difficile dans des environnements encombrés. La commande de ces plates-formes pour réaliser un déplacement particulier est un problème à part entière que nous n'aborderons pas dans ce cours. Par contre, il est possible de prendre en compte ces contraintes de manière relativement simple dans la planification. [5]

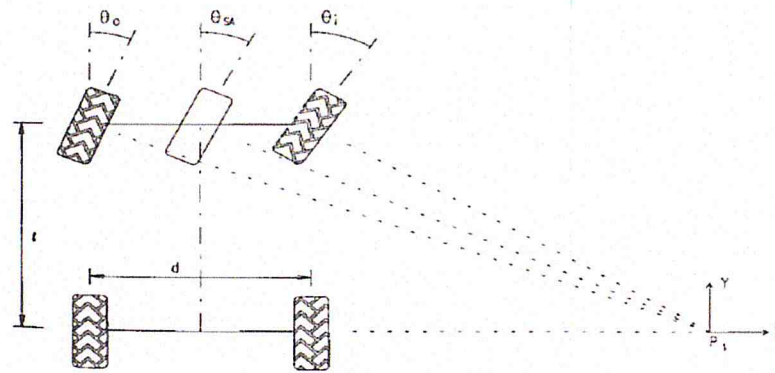


Figure 1.7 : Exemple de robot non holonome [5].

3.3.5 Les robots à pattes :

Des plates-formes à deux, quatre ou six pattes peuvent également être utilisées. Elles ont l'avantage théorique de pouvoir se déplacer sur des terrains assez complexes, même si en pratique la plupart de ces plates-formes ne fonctionnent que sur des sols plans.

Les plates-formes à six pattes sont relativement pratiques car le robot peut être en équilibre permanent sur au moins 3 pattes, ce qui facilite le contrôle. Les plates-formes à deux ou quatre pattes sont plus complexes à commander et le simple contrôle de la stabilité et d'une allure de marche correcte reste aujourd'hui difficile, ce qui les rend en général relativement lentes. L'odométrie de ce type de plates-formes est de plus généralement d'assez faible qualité. Ces différents facteurs font que ces plates-formes sont rarement utilisées quand l'application visée a un besoin précis de positionnement et de navigation. De telles plates-formes commencent cependant à apparaître à relativement grande échelle. [5].

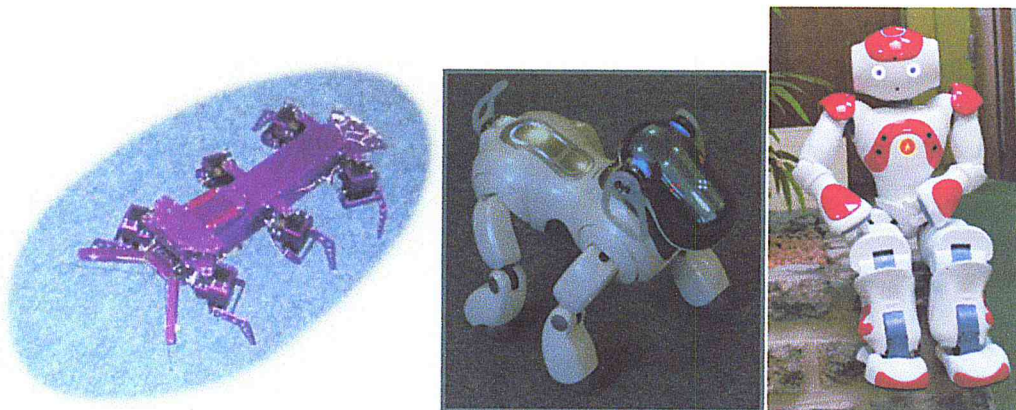


Figure 1.8 : Exemples de robots à pattes. Hexapode de AAI Canada [5].

5. Manipulateur mobile :

Le concept de manipulateur mobile est très simple. Il s'agit d'associer, dans un même système, un ou des moyens de locomotion à un ou des moyens de manipulations. Le(s) moyen(s) de locomotion assure(nt) au système un espace de travail limité principalement

par son autonomie énergétique. Le(s) moyen(s) de manipulation assure(nt) les capacités de préhension et de manipulation. Cette définition ouverte laisse la place à un grand nombre de combinaisons possibles, illustrées par la diversité des dispositifs expérimentaux et/ou commerciaux existants à ce jour. Avant de donner quelque exemple de ces systèmes, notons que leur succès est lié à l'espace de travail quasiment infini qu'ils présentent. Quand un bras manipulateur industriel voit son espace de travail limité à quelque mètres cubes, le manipulateur mobile peut lui évoluer sur plusieurs centaines de mètres carrés ou de mètres cubes pour ceux, comme les manipulateurs mobiles sous-marins, dont le moyen de locomotion ne nécessite pas de contact direct avec le sol. Ils ouvrent donc un grand nombre de possibilités qui pour la plupart restent à explorer [3].

5.1 Les types des manipulateurs mobiles

Parmi les manipulateurs mobiles existant actuellement, nous pouvons citer trois grandes familles [3]

- Les « multipèdes ».
- Les manipulateurs mobiles sous-marins.
- Les manipulateurs mobiles à roues.

5.1.1 Les multipèdes :

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

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) [3].

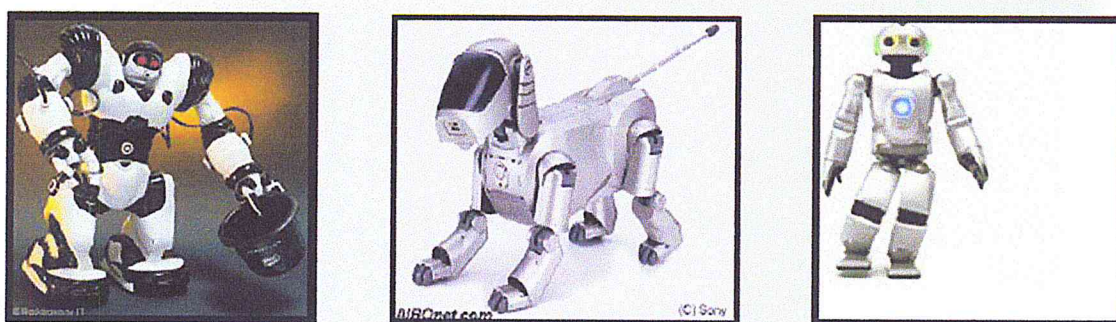


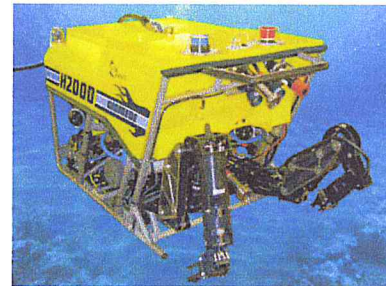
Figure.1.9 : Exemple des robots multipèdes [3].

5.1.2 Les 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 1.10 et figure1.11) [4].



(a)



(b)

Figure 1.10 : Robots Sous-Marins(a) Robot ROV3, (b) Robot H2000 [4].

5.1.3 Les manipulateurs mobiles à roues.

Moins médiatiques que les premiers, moins exploités que les seconds, les manipulateurs mobiles à roues sont (figure 1.12), cependant, plus répandus que ceux précédemment cités. Cela tient particulièrement à deux faits [3] :

- Leur conception mécanique relativement simple.
- L'adéquation naturelle de leur moyen de locomotion à un grand nombre de terrains.



Figure 1.11: lynxmotion-aluminum-a4wd1-robot [3].

5.2 Les propriétés des manipulateurs mobiles :

Le paragraphe suivant présente les propriétés principales relatives aux manipulateurs mobiles [3]:

- L'atout indéniable que présente ce type de robots tient dans la combinaison de deux sous-systèmes aux fonctions très différentes. Il est même tentant de résumer cet atout en disant que là où le manipulateur ne peut aller, la plateforme l'y emmène et que ce que la plateforme ne peut pas prendre, le manipulateur le prend pour elle. Cette vision est très réductrice. Elle limite fortement le type de mission envisageable et amène à regarder le système comme deux sous-systèmes agissant avec peu de coordination : « La plateforme va en (x, y) . Le manipulateur prend l'objet A. La plateforme va en (u, v) . Le manipulateur pose l'objet A sur l'objet B » ou encore « le manipulateur effectue une opération à l'endroit L, la plateforme va à l'endroit M. Le manipulateur effectue une opération à l'endroit M, etc. ». Pour un manipulateur mobile à roues il est toujours possible de décomposer le mouvement coordonné du système en plusieurs mouvements simples et indépendants pour chacun des sous-systèmes [3].
- Une seconde caractéristique importante pour ces systèmes est qu'ils sont souvent redondant vis à vis de la tâche assignée à (aussi appelée tâche principale). De manière non formelle, il est assez juste de dire que cette redondance est liée à la présence d'un plus grand nombre d'actionneurs que le minimum nécessaire pour que le système puisse exécuter sa tâche principale. Cette propriété assure l'existence d'une infinité de solutions au problème de commande opérationnelle, infinité parmi laquelle la meilleure solution, au sens d'un critère donné, est à choisir [3].
- Une des propriétés des manipulateurs mobiles à roues est aussi leur caractère non holonome. Disons qu'un système mécanique non holonome est un système soumis à des contraintes sur les vitesses des différents corps qui le composent. L'exemple typique de ce genre de système est la voiture. Les roues d'une voiture doivent respecter une contrainte de roulement sans glissement sur le sol ce qui contraint leurs vitesses respectives à ne pas pouvoir évoluer de manière indépendante les unes des autres. Ces contraintes sont très fortes puisqu'elles limitent à chaque instant les directions du mouvement de chacun des corps du système [3].
- Un manipulateur mobile à roues est un système non holonome et mis à part dans le cas particulier de plateformes omnidirectionnelles, les propriétés de ces systèmes pour la commande diffèrent de celles des bras manipulateurs.
- L'espace de travail « illimité » et la redondance du système offrent des possibilités en termes de missions qui n'existent pas pour des bras manipulateurs (du point de vue de l'espace de travail en particulier) ou pour des robots mobiles à roues seuls [3].

5.3 Domaines d'applications

5.3.1 Domaine spatial

Les manipulateurs mobiles spatiaux sont destinés à explorer des environnements où l'homme ne peut pas se rendre, c.-à-d. des environnements souvent mortels pour l'homme [7]. Un exemple concerne l'exploration d'autres planètes comme la planète Mars.

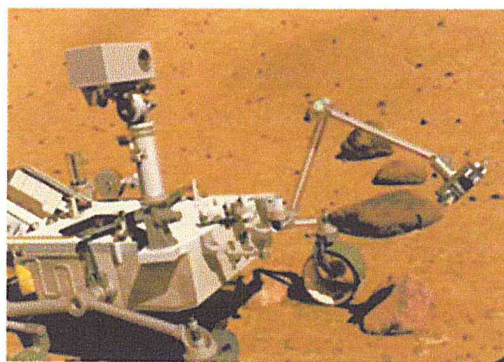


Figure 1.12 : Exemple d'un manipulateur mobile de spatiale

5.3.2 Domaine de service

Un robot domestique, ou robot du service, est un robot autonome qui est utilisé pour les tâches quotidiennes (figure 1.14). Beaucoup de ces robots sont utilisés pour les tâches domestiques de base. D'autres sont utilisés à des fins pédagogiques ou pour le divertissement.

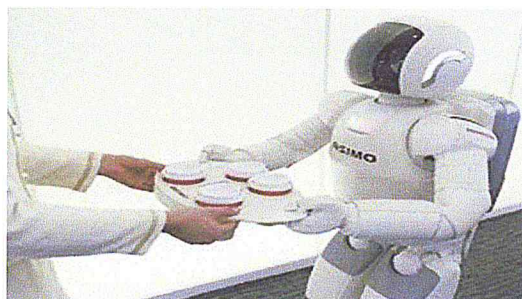


Figure 1.13 : asimo-honda [8].

5.3.3 Domaine militaire

Quelques experts et universitaires ont questionné l'usage de robots pour des combats militaires, surtout quand quelques degrés de fonctions autonomes sont donnés à de tels robots. La Marine Américaine a consolidé un rapport qui indique que, comme les robots militaires deviennent plus complexes, il devrait y avoir la plus grande attention à l'implication de leur capacité à prendre des décisions autonomes.



Figure 1.14 : Robot camp [8].

5.3.4 Domaine médical

Un manipulateur mobile médical est une machine utilisée pour soigner des patients. Il est piloté par un médecin, son comportement change peu en fonction de son environnement. On note que dans le domaine médical les manipulateurs mobiles sont rarement utilisés. Par contre, les bras de manipulation sont très utilisés (Figure 1.15). De nouvelles techniques sont exploitées, comme les actionneurs AMS (alliages à mémoire de forme), la micro robotique, etc. Des algorithmes d'analyse d'images sont développés dans la même voie [9].

L'objectif principal d'introduire les robots « médico-chirurgicaux » est de développer une coopération entre un homme (le chirurgien) et une machine (le robot) qui cherche à exploiter les capacités des deux pour réaliser une tâche mieux que ce que pourrait faire l'un des deux tout seul.

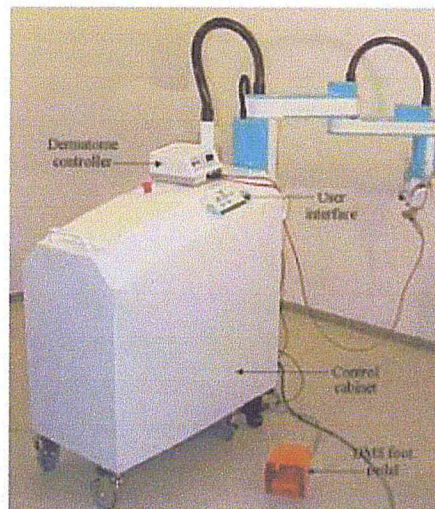
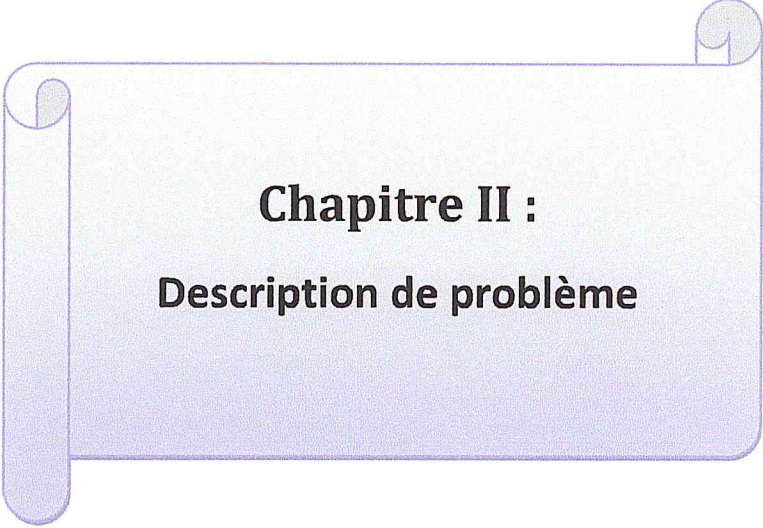


Figure 1.15 : Robot DERMAROB [9].

6. Conclusion :

Un Robot manipulateur mobile se compose de deux sous-systèmes mécanique hétérogènes conçus de manière distinctes (base mobile, manipulateur) et réagissant différemment aux influences extérieurs.

Un tour du monde sur les manipulateurs mobiles qui existent de nos jours a été effectué dans ce chapitre. Nous avons également cité les différents domaines d'applications de ces robots. Nous allons nous intéresser dans le prochain chapitre à la description du problème que nous avons traité.



Chapitre II :
Description de problème

1. Introduction :

Dans le contexte de la robotique de service, un robot doit effectuer des tâches dans des environnements équivalents à l'espace atteignable des Hommes. Pour augmenter l'espace de travail, les bras manipulateurs sont montés sur des bases mobiles. Les tâches à réaliser sont plus diverses que dans le contexte industriel (transporter un objet, débarrasser une table, ramasser un objet au sol, amener un objet à l'Homme...). Chaque tâche peut être décomposée en plusieurs sous-tâches élémentaires.

Nous avons entrepris dans ce mémoire d'exploiter une architecture Multi-agents existante pour le contrôle d'un robot manipulateur mobile. La fonction entreprise consiste en l'exécution d'une tâche de saisie d'objet.

Dans ce chapitre nous commençons par un Etat de l'art de travaux relatifs à l'exécution de tâches de saisie d'objets par des robots mobiles manipulateurs. Nous abordons ensuite la description du système Multi-agents existant pour le contrôle des systèmes robotiques. Nous allons enfin définir l'objectif de notre travail et la formulation du problème.

2. Etat de l'art de travaux relatifs à l'exécution de tâches pour des robots mobiles manipulateurs :

Nous citons quelques travaux qui ont été effectués dans le domaine de la manipulation mobile et de la saisie d'objets.

Steven Gray [23] et ses collègues ont développé récemment une approche d'ouverture de porte (saisie le poignet de la porte). Les approches adoptées ont été expérimentées sur le robot PR2

La tâche d'ouverture d'une porte se compose de deux étapes principales: la détection de la porte et la détermination de ses paramètres. Cela est suivi par la planification et l'exécution du mouvement d'ouverture de la porte. La localisation de la porte ainsi que la taille et la position de la poignée sont déterminés en utilisant une des caméras stéréo. Celle-ci fournit un cadre pour le suivi des marqueurs ARToolKit. Les propriétés de portes quelconques sont définies à l'avance. Ces propriétés 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, du côté de la charnière, la direction d'oscillation, et la force nécessaire à appliquer pour ouvrir la porte.



Figure 2.1 : Détection de la porte avec l'une des caméras stéréo par ARToolKit.

L'objectif principal du travail présenté par *Trabelsi et al* [31] est de donner au robot des capacités de manipulation dans un environnement humain. Dans ce contexte, ils se sont intéressés à la saisie d'objets à l'aide d'un bras manipulateur doté d'une pince et de deux capteurs (caméra et ultrasons). L'information de distance fournie par le capteur ultrasonore est associée à l'image afin d'avoir une connaissance sur la profondeur (distance pince-objet). Ils ont développé une stratégie de saisie pour deux types différents d'objets (Sphère et Cylindre) en se basant sur le modèle géométrique de la caméra et la distance pince-objet.

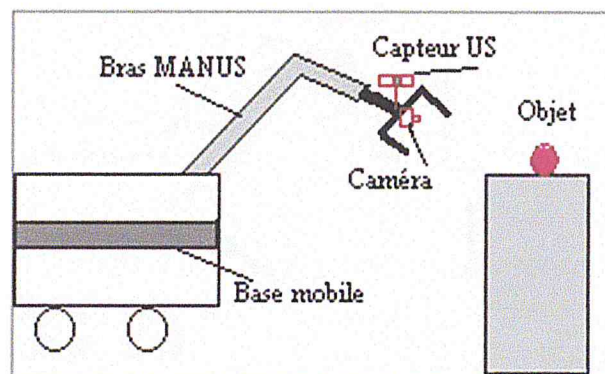


Figure 2.2 : le système mobile pour la saisie.

Mansard et al [32] Dans ce travail, ils ont proposé un cadre général pour réaliser une tâche complexe en utilisant des méthodes de contrôle référencées capteur. Leur approche consiste à diviser l'action à réaliser en plusieurs tâches élémentaires référencées capteur. Ces tâches sont exécutées simultanément par une structure de contrôle nommée pile de tâches.

Un contrôleur de haut niveau est alors utilisé pour assurer la convergence de l'ensemble des tâches et le respect de contraintes secondaires. Cette approche a été implémentée en utilisant un asservissement visuel. À titre d'exemple, ils ont réalisé une tâche de saisie pendant la marche avec un robot humanoïde de type HRP-2. Le robot marche le long d'une trajectoire imposée, tout en gardant l'objet à saisir dans le champ de

vue des caméras. Lorsqu'il se trouve suffisamment proche, une tâche de saisie guidée vision permet au robot de saisir la marche sans interrompre la marche.

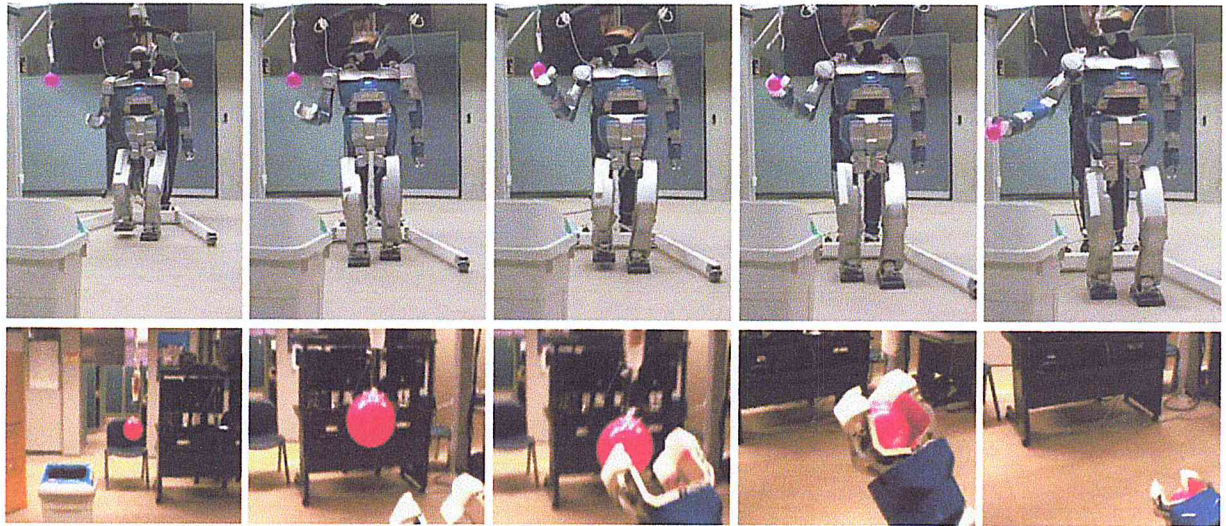


Figure 2.3 : Saisie d'une balle pendant la marche : exemple typique d'exécution avec le robot HRP-2. Haut : vue de la caméra externe.

Bas : vue de la caméra embarquée, utilisée pour le contrôle des mouvements du robot.

Nadeau et Dune [33] ont présenté un travail où ils se sont intéressés à la saisie d'objets inconnus par un bras robotique équipé d'un système de vision. Une nouvelle méthode permettant d'améliorer l'estimation de la pose et la forme de l'objet à saisir est proposée. Cette méthode s'appuie dans un premier temps sur une reconstruction voxellique de cet objet puis sur l'estimation de ses axes principaux et de ses dimensions. Des choix simplificateurs dans leur implémentation des techniques de reconstruction voxellique assurent un gain en temps de calcul tout en conservant la forme générale de l'objet reconstruit.

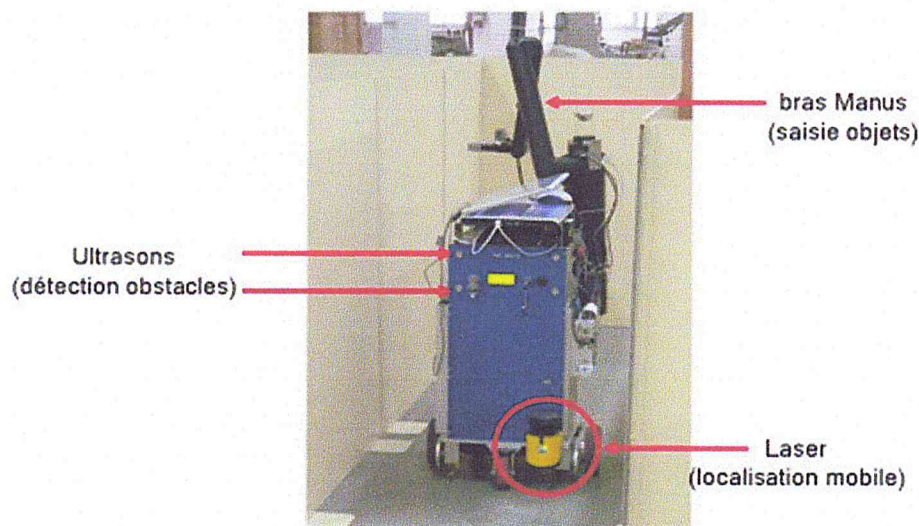


Figure 2.4 : Les éléments de la base mobile du robot SAM.

Moughlbay et al [39], ont étudié le cas de la manipulation à deux bras d'un objet pour lequel il est nécessaire de définir différentes tâches robotiques référencées capteurs : commande du bras gauche (manipulation/saisie), commande du bras droit (manipulation/saisie), commande de la tête (visibilité), commande du bassin (équilibre), commande des pieds (stabilité/marche). Pour cela ils ont définis un ensemble de points de contrôle sur le robot et associés à chacun des fonctions de tâches particulières. La stratégie de commande repose sur l'utilisation du formalisme de tâches redondantes hiérarchiques afin de s'assurer que chacune des tâches est compatible avec les autres tout en respectant une hiérarchie préétablie.

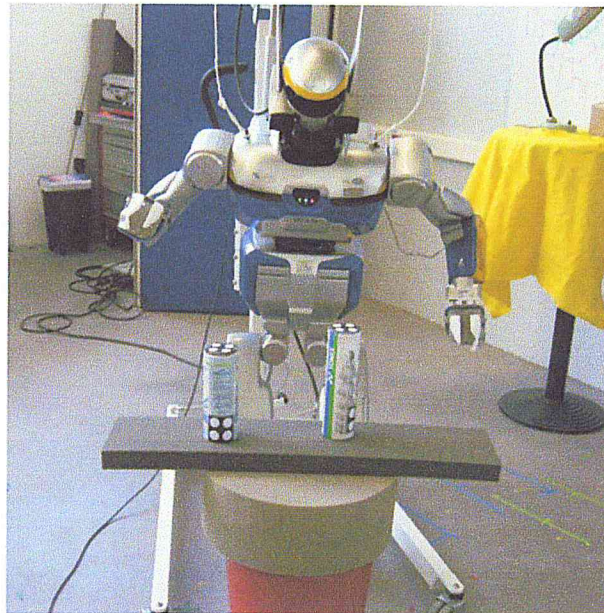


Figure 2.5 : Expérimentations sur HRP2.

3. Les approches principales de l'étude du comportement d'un système robotique :

De nos jours, les robots possèdent des rôles importants dans beaucoup de domaines. Pour construire un contrôleur de robot, et faire exécuter la tâche étudiée à un robot manipulateur mobile, nous avons quatre approches comportementales principales : [25]

3.1 Approche Délibérative - Penser et puis agir :

Dans ce contrôleur, le robot utilise toutes des informations qu'il reçoit (par le capteur) et par des règles, il conserve, il décide quelle action il va établir. Cette approche est bonne pour des robots qui travaillent dans un environnement qui ne change pas souvent.

3.2 Approche Réactive - Ne pas penser, (ré) agir :

C'est une approche dans laquelle le robot peut répondre rapidement pour des environnements changeants et déstructurés. Par contre, cette architecture contient des

limites : les comportements sont préétablis, ils n'ont pas de capacité de changer pour chaque environnement.

3.3 Approche Hybride - Penser et agir concurremment :

Pour limiter des faiblesses des deux approches au-dessus, il a été proposé une architecture hybride. Cette architecture possède des capacités de raisonnement propres d'architecture délibérative et les capacités réactives d'architecture réactive. En conséquence, le composant réactif s'occupe de demandes immédiates comme marche lorsqu'il y a évitement d'obstacles. Dans le même temps, le composant délibératif opère des tâches qui ne sont pas urgentes comme accomplir la planification d'un chemin global.

3.4 Approche Contrôle à base de comportement – Penser la façon dont vous réagissez :

D'une façon générale, la théorie basée comportement réclame que la meilleure manière de construire les robots intelligents est d'accumuler des contrôleurs de petites et simples tailles, chacun est un expert dans son propre domaine. Les comportements sont tous autonomes dans le sens où d'autres comportements ne peuvent pas les influencer directement. Les comportements produisent des commandes, qui sont combinées, par un certain mécanisme de coordination dans la commande finale [31].

L'idée de commande basée comportements est de découper le problème de la mission en petits sous problèmes, résoudre ces problèmes et synthétiser un contrôleur.

Le contrôle à base comportement se fait :

- **Par Action Primitive :** le problème est devisé en tâches par ordre de primitives prévues appelées « actions primitives ». Chaque action primitive est conçue avec un mécanisme d'ajustement des erreurs pour aider à traiter des erreurs de positionnement de la base mobile [35].
- **Par contrôleur multi-agents :** Un système multi-agents peut être étudié comme un système asynchrone, concurrent, partagé et aléatoire. En programmation objet, un agent est généralement vu comme un objet doté d'activités propres ce qui lui confère de facto une certaine autonomie. Un agent autonome évolue dans un environnement et est capable d'agir de son propre chef afin de remplir les objectifs pour lesquels il a été créé [36].

4. Le Système multi-agents existant:

4.1 Architecture physique :

Le robot Robuter/ULM est constitué d'une plateforme mobile rectangulaire surmontée d'un bras manipulateur à six articulations. La plateforme est à commande différentielle. Un capteur laser LMS est placé à l'avant du véhicule pour éviter des obstacles dans l'environnement. Une caméra IP est placée sur la pince du bras embarqué comme outil d'aide à la manipulation d'objets. Les caractéristiques techniques du robot ainsi que des capteurs sont présentés en annexe.

La figure suivante illustre l'architecture matérielle du système de contrôle du robot. Le système expérimental englobe un site local et un site distant.

- **Le site local :** constitué d'un PC Hôte (avec le système d'exploitation Windows XP), ainsi qu'un système de communication Wifi.
- **Le site Distant :** constitué du Robot manipulateur mobile Robuter/ULM contrôlé par le PC embarqué (avec le système d'exploitation Linux), ainsi qu'un module de communication Wifi. La caméra embarquée sur la pince du bras est connectée également à ce module.

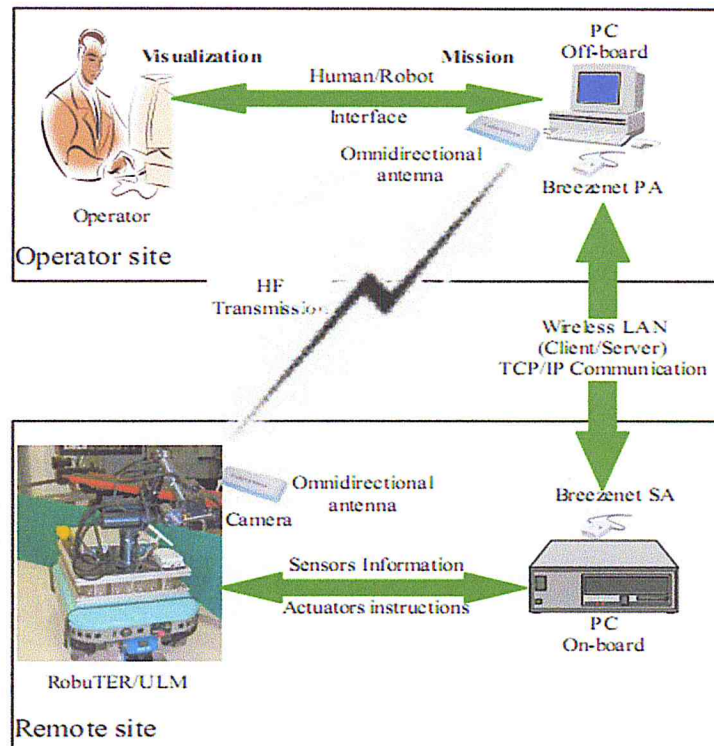


Figure 2.6: Architecteur physique de système robotique [35].

4.2 Architecture logicielle :

L'architecture multi-agents de contrôle et de supervision proposée (Figure 2.7) est constituée d'un ensemble de sept agents hybrides (délibératifs et réactifs) qui pilotent un ensemble de ressources hétérogènes du robot (base mobile, bras manipulateurs, système de capteurs et système de vision) : *Agent Superviseur*, *Agent Robot Manipulateur Local*, *Agent Robot Mobile Local*, *Agent Robot Manipulateur Distant*, *Agent Robot Mobile Distant*, *Agent Simulateur* et *Agent Vision* [29].

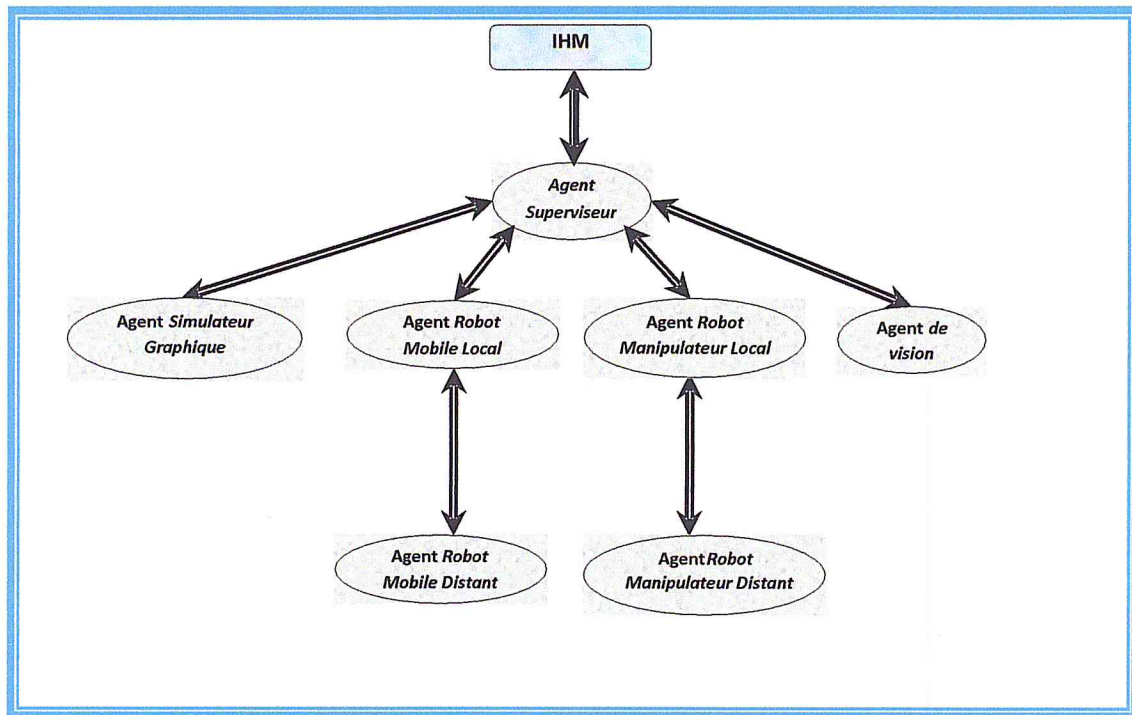


Figure 2.7 : Modèle de l'architecture de contrôle et de supervision proposée

Les *Agent Robot Mobile Local* et *Agent Robot Manipulateur Local*, contrôlent respectivement, la base mobile et le bras manipulateur du robot à partir d'un site local (off-board). L'*Agent Superviseur* a pour rôle la gestion des missions et la répartition des plans d'opérations sur les différents agents ainsi que la communication et la négociation avec ces derniers.

Les agents *Agent Superviseur*, *Agent Robot Mobile Local* et *Agent Robot Manipulateur Local*, possèdent un niveau substantiel d'autonomie à travers des capacités de perception de l'environnement qui est réalisée à travers la communication avec les *Agents Distant* et l'échange d'informations entre les agents. Ils possèdent aussi des connaissances suffisantes pour une *prise de décision* autonome.

Deux autres agents ont été introduits afin d'assurer la communication entre les agents locaux et les ressources matérielles, ainsi que d'autres fonctionnalités. Ces agents sont l'*Agent Robot Mobile Distant* pour le contrôle de la base mobile, et l'*Agent Robot Manipulateur Distant* pour le contrôle du bras manipulateur.

Les caractéristiques que doit posséder cette architecture multi-agents sont :

- **Modularité** : La modularité de l'architecture des agents doit s'articuler autour de la décomposition en modules qui peuvent être développés et mis en œuvre. Cette caractéristique permet à tout système de contrôle, d'évoluer en lui rajoutant de nouvelles fonctionnalités.
- **Réactivité à l'environnement** : Les agents doivent être capables de gérer en temps réel des événements extérieurs asynchrones afin de respecter la dynamique de l'environnement et celle de l'architecture. Un événement extérieur peut avoir plusieurs

origines : présence d'un obstacle imprévu, panne inopinée, requête en provenance d'un autre agent, etc. La réactivité implique généralement un traitement en temps réel de ces événements. Ainsi, le système de contrôle doit intégrer la notion de priorité et d'urgence de traitement des événements.

- **Comportements intelligents** : L'intelligence des agents se traduit par des capacités de perception, de communication, de raisonnement et d'action [29].

4.3 Structure des messages échangés dans le système :

Les messages échangés entre les différents agents du système sont composés de trois champs : L'émetteur, le destinataire et l'information utile. Cette même information utile est composée de deux champs : Le type et les données utiles. La taille de l'information utile varie d'un message à un autre. Elle est composée de plusieurs données. Ces données sont séparées par des dièses (#). La figure 4.4 schématise la structure d'un message.

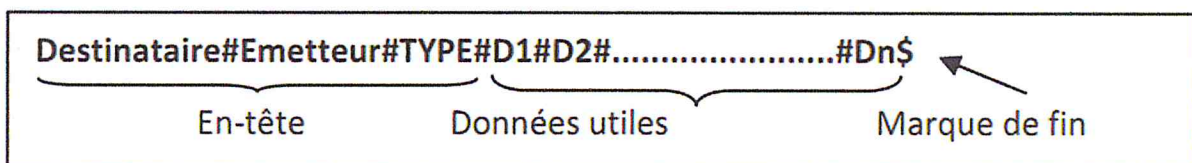


Figure 2.8 : Structure d'un message

Chaque agent dispose de sa propre méthode d'interprétation des messages. Cette interprétation dépend, non seulement du type du message et de la taille de l'information utile, mais aussi de l'émetteur [29].

5. Le contrôle du robot :

5.1 Mouvement du bras manipulateur:

Le mouvement du bras manipulateur s'accomplit selon deux modes de control. Le premier mode consiste en un control en position. Cela signifie que dans le message envoyé, les positions à faire atteindre aux différentes articulations sont spécifiées. Par contre, quand le robot manipulateur est contrôlé en vitesse, il ne suffit d'imposer que les vitesses de chaque articulation. Le message envoyé doit inclure les vitesses à imposer à chaque moteur. Dans notre application, nous n'utilisons que le mode de control en position.

5.2 Déplacement de la base mobile:

Le déplacement de la base nécessite une génération de trajectoire et cela se fait par deux modes hors ligne et en ligne. Dans notre travail, on déplace la base mobile par le mode hors ligne (génération de trajectoire). Cependant, si la base rencontre un obstacle (détecté par capteur LMS) dans sa trajectoire, le control du robot va basculer et passer au mode en-ligne pour l'éviter.

En mode hors-ligne, le message envoyé au robot consistera en une succession de vitesses envoyées aux roues de la plateforme mobile. Quant au mode en-ligne, un module de navigation sera sollicité, et le message envoyé sera constitué de la position et de

l'orientation finale à faire atteindre au robot mobile. Des détails concernant les deux modes de contrôle du robot mobile sont présentés en annexe.

6. Formulation du problème de saisie d'un objet :

Dans ce travail, on essaie de réaliser la tâche de saisie d'un objet par un manipulateur mobile. La planification d'exécution d'une tâche se divise en deux sous-problèmes (correspondant à deux niveaux de considération du problème) :

- la planification de mouvements,
- la planification des tâches.

Nous avons donc formulé les deux problèmes en un certain nombre de questions comme cela est présenté ci-après.

6.1 Description du problème lié à la saisie d'un objet par la pince du bras:

Pour saisir un objet par un manipulateur, il faut connaître la position et l'orientation de l'objet par rapport à un repère fixe. Généralement, la détection des objets se fait à l'aide des caméras embarquées sur l'effecteur des manipulateurs mobiles. La question qui reste posée: comment extraire les coordonnées d'objet par rapport au repère fixe R_B par une caméra?

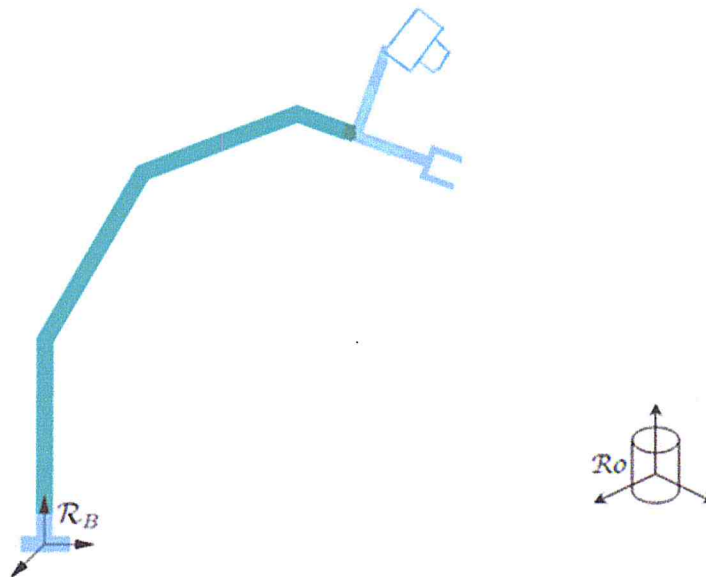


Figure 2.9 : description de problème de localisation d'objet.

Sur la figure présentée ci-dessus, le repère R_B est le repère lié à la base du bras manipulateur. Le repère R_o est le repère lié à la position de l'objet. Pour une tâche de saisie, il est défini par l'algorithme de traitement d'image utilisé.

6.2 Description du problème lié au déplacement de la plateforme mobile :

Le bras manipulateur peut atteindre des distances limitées, ces distances sont appelées espace de travail. Quand le bras est embarqué sur une base mobile, cela augmentera cet espace. Si les coordonnées d'objet sont en dehors de la zone du travail du bras, la plateforme mobile va se déplacer d'une position initiale vers une position finale plus proche de l'objet,

ce qui permettra au bras manipulateur d'atteindre l'objet. Mais à quel moment le système robotique va se déplacer comment?

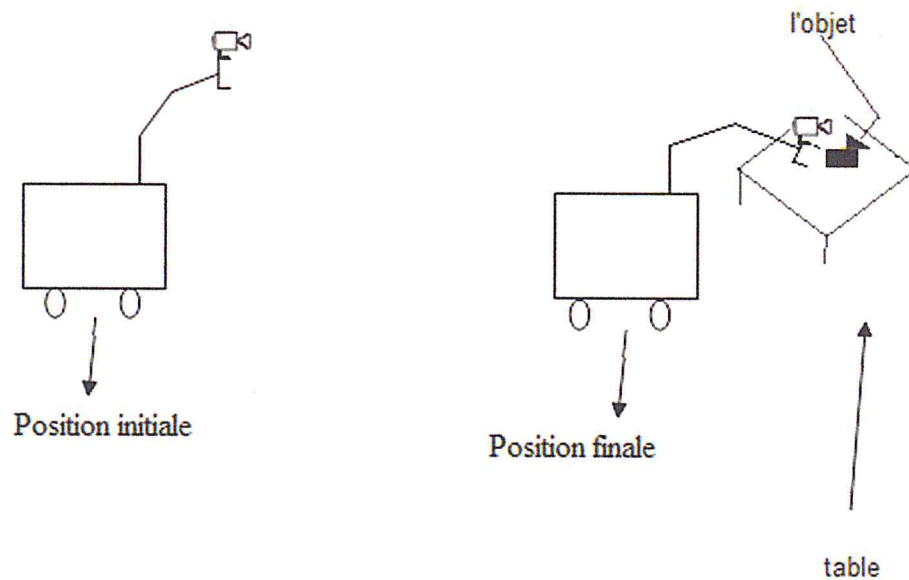
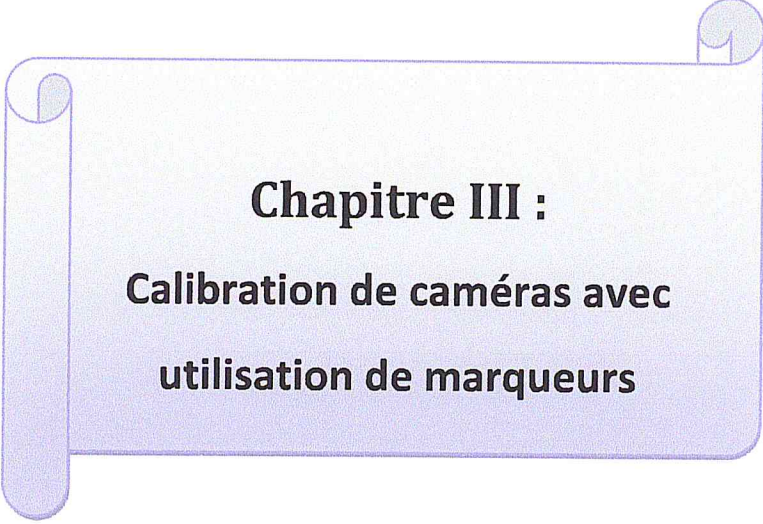


Figure 2.10 : description de problème de déplacement de la base mobile.

7. Conclusion :

Nous avons présenté les différents travaux effectués et les plus grandes approches concernant la saisie d'objets. Nous avons passé en revue le problème général de la saisie d'objet par un manipulateur mobile, évoluant dans un environnement semi-connu et nous avons montré les méthodes de résolution proposées dans la littérature. Nous allons présenter dans les prochains chapitres la méthode proposée pour la résolution du problème posé. En effet nous avons exploité l'architecture multi-agents existante pour mouvoir le robot, et nous avons ajouté un module de vision pour la localisation de l'objet à saisir. La solution est basée sur l'exploitation de la librairie Artoolkit.



Chapitre III :
Calibration de caméras avec
utilisation de marqueurs

1. Introduction:

Dans le chapitre précédant nous avons cité les deux grands problèmes de la saisie qui concernent le déplacement de la base mobile et la détection de l'objet à saisir.

Nous allons dans ce chapitre évoquer une phase important de notre travail qui concerne l'exploitation et l'utilisation de la librairie ARToolKit. En effet, nous allons commencer par présenter cette librairie, et la manière avec laquelle nous allons m'exploiter. Nous allons ensuite Nous allons exploiter les caractéristiques de cette bibliothèque pour calibrer la caméra embarquée sur le robot.

2. Librairie ARToolkit :

ARToolkit est une bibliothèque (il y a des autre librairies sont détaillé dans l'annexe) qui permet le développement d'applications de Réalité Augmentée RA (La réalité augmentée désigne les systèmes informatiques qui rendent possible la superposition d'un modèle virtuel 3D ou 2D à la perception que nous avons naturellement de la réalité et ceci en temps réel). Elle a été conçue pour connaître le point de vue de l'utilisateur. Une application de RA nécessite des informations sur la façon dont l'utilisateur regarde le monde réel. Les fonctions de cette bibliothèque permettent de calculer la position réelle de la caméra et son orientation, par rapport à des marqueurs, en temps réel. ARToolKit inclut certaines caractéristiques [11]:

- Suivi de la position et de l'orientation du marqueur avec une seule caméra.
- La capacité d'utiliser n'importe quel marqueur carré avec motif.
- Le calibrage de la caméra est simple grâce aux fonctions données.
- Assez rapide pour des applications RA en temps réel.
- Coût d'utilisation très bas.
- Une bibliothèque multiplateforme (Windows, Linux, Mac OS X, SGI),
- Open Source avec licence GPL pour un usage non commercial [17].

Le principal avantage d'ARToolKit, est qu'elle permet de récupérer des informations 3D à partir d'une seule caméra. On peut obtenir non seulement les positions des marqueurs par rapport à la caméra, mais aussi leur orientation. Les autres méthodes proposent un suivi 2D par image, ce qui implique l'utilisation d'au minimum deux caméras pour pouvoir obtenir des informations 3D. Le seul inconvénient d'ARToolkit par rapport aux autres méthodes est d'être la seule à utiliser des marqueurs. L'utilisation des marqueurs implique la perte du suivi quand le marqueur n'est pas (ou très peu) visible, même si l'objet associé est toujours visible par la caméra. Elle reste cependant la méthode la moins coûteuse du fait du nombre des caméras utilisée [17].

Plusieurs méthodes peuvent être utilisées afin de repérer la position et l'orientation de la caméra dans l'environnement. ARToolKit est basé sur l'utilisation de marqueur (Un marqueur est un motif simple noir dessiné sur un fond blanc, lui-même entouré d'un cadre noir [19]).

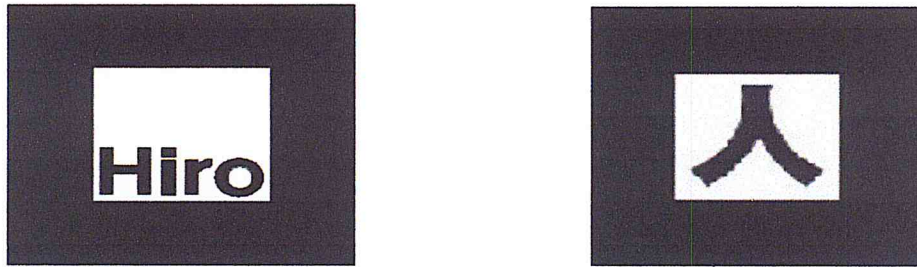


Figure 3.1 : Exemples de marqueurs

ARToolKit utilise un algorithme afin de détecter la présence de ce marqueur dans chaque trame du flux vidéo en temps réel et en déduit la position et l'orientation de la caméra. Nous verrons que les conditions d'utilisation de capture vidéo joue un rôle important (luminosité, résolution, distance, mise au point...) et que le marqueur peut évidemment être déplacé dans toutes les directions, soumis néanmoins à un degré de liberté limité (ce dernier devant rester visible par la caméra) [11].

2.1 Utilisation de la librairie ARToolkit :

2.1.1 Algorithme de détection du capteur:

L'algorithme de détection d'ARToolkit consiste en une suite d'opérations sur chaque image du flux vidéo afin d'une part de repérer la présence d'un marqueur, puis de l'identifier parmi les différents marqueurs chargés dans l'application. Nous pouvons décrire le fonctionnement des différentes fonctions d'ARToolKit de la manière suivante [18]:

Initialisation :

- Initialisation du flux vidéo en lisant les paramètres de la caméra et les données des patterns.

Boucle infinie :

- Sélection d'une image du flux vidéo.
- Détection et reconnaissance des patterns sur l'image.
- Calcul des transformations de la caméra par rapport aux patterns détectés.

Fermeture :

- Fermeture du flux vidéo.

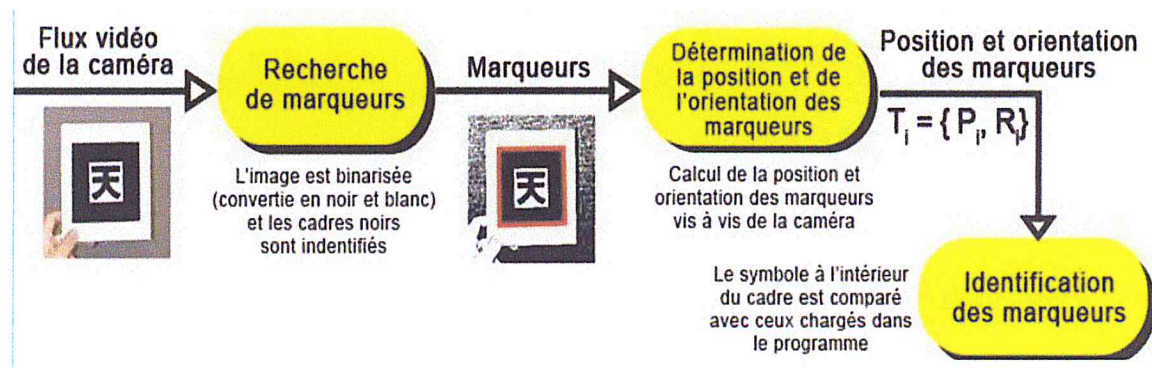


Figure 3.2 : Algorithme de détection du capteur.

3. Calibration de caméras :

La calibration d'une caméra est un processus de modélisation de la correspondance entre les objets 3D et les images 2D. Ce processus est nécessaire lors de la récupération des informations 3D à partir d'images 2D. Nous devons connaître plus précisément les paramètres intrinsèques et extrinsèques de la caméra.

Les paramètres intrinsèques en imagerie décrivent les caractéristiques géométriques d'une caméra. Ils dépendent du modèle de projection utilisé. Parmi les paramètres intrinsèques que la plupart des approches estiment, nous trouvons:

- La distance focale, f .
- Les facteurs d'agrandissement de l'image k_u et k_v .
- Le facteur d'inclinaison, s .
- Le point principal (u_0, v_0) .
- Les paramètres de distorsion tangentielle et radiale 1
- Le paramètre de forme de miroir (s).

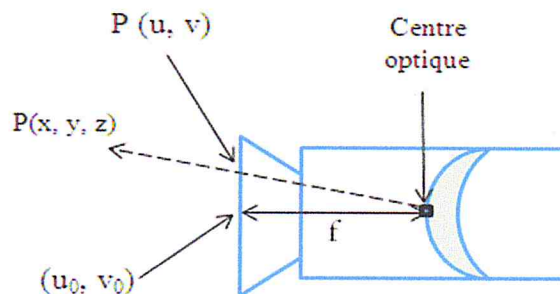


Figure 3.8 : caractéristiques de la camera.

3.1 Les méthodes classiques de calibration:

3.1.1 Calibrage photogrammétrique:

Il est réalisé en observant un objet dont la géométrie dans un espace 3D est connue avec une grande précision. Le calibrage est généralement composé de deux à trois plans orthogonaux l'un à l'autre. Parfois, on utilise aussi un plan qui est subi d'une certaine translation [15].

3.1.2 Auto-calibrage:

Le principe de l'auto-calibrage est d'utiliser des cibles virtuelles, contraintes géométriquement par des informations a priori éventuelles sur les paramètres internes de la caméra. Ces cibles sont des entités imaginaires (sans réalité physique), localisées à l'infini, qui sont invariantes lors des mouvements de la caméra. Les méthodes d'auto-calibrage nécessitent généralement un traitement mathématique plus complexe que les méthodes de calibrage.

3.1.3 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, et le calibrage est calculé à partir d'une analyse de la relation des points entre les images [50].

3.1.4 Technique de Zhang:

La méthode de Zhang facile à utiliser et précise pour obtenir des paramètres de l'appareil visuel à la fois intrinsèques et extrinsèques. Cet algorithme a été mis en œuvre dans Matlab Caméra Calibration Toolbox par Jean-Yves Bouguet et en langage C dans la bibliothèque Intel OpenCV. Ces deux bibliothèques sont probablement des outils de calibrage de la caméra les plus largement utilisés de nos jours.

La méthode de calibration de Zhang nécessite une grille plane en damier à être placée à différentes orientations (plus de deux) en face de la caméra. L'algorithme développé utilise les points de coins extraits du motif en damier pour calculer une homographie entre les points d'image des images différentes n , jusqu'à un facteur d'échelle. Ensuite, à l'intérieur de l'appareil et des paramètres extérieurs sont récupérées à l'aide d'une solution de forme fermée, tandis que les termes de distorsion radiale du quatrième ordre sont récupérés dans une solution des moindres carrés linéaire. Une minimisation non linéaire finale de l'erreur de projection, résolu en utilisant une méthode de Levenberg-Marquardt, affine tous les paramètres récupérés [16].

4. Les résultats obtenus :

Dans cette partie, nous donnons l'ensemble des résultats obtenus lors des essais expérimentaux en vue de la vérification des performances de la commande implémentée sur le manipulateur mobile Robuter/ULM, de son adéquation avec les objectifs tracés.

4.1. Calibration avec la librairie ARToolKit:

Nous avons utilisé la méthode en une étape et les observations sur les mesures d'erreur de positionnement ont permis de montrer l'amélioration due à la calibration. La calibration est donc une opération nécessaire, qui aura une influence sur l'enregistrement des marqueurs.

4.1.1. positionnement de la camera :

On place devant la caméra un mur de calibrage (feuille imprimer) qui contient 24 points (matrice 6 sur 4) la distance entre un point et l'autre 10 cm. (figure 3.6)

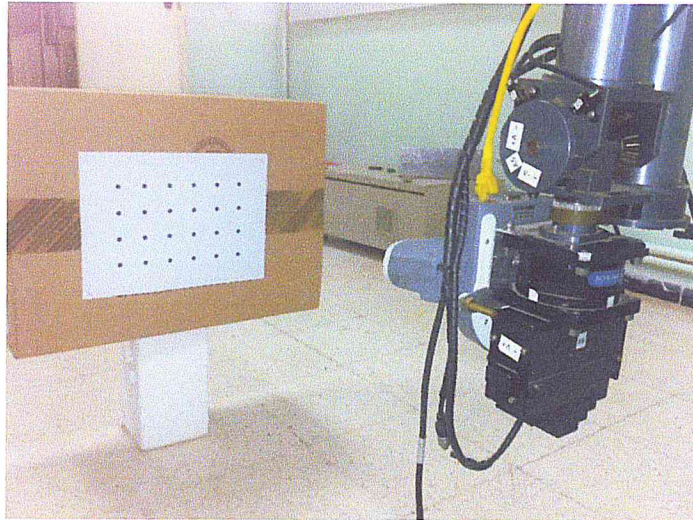


Figure 3.6 : Positionnement de la camera avant de calibrer.

4.1.2. Processus expérimental de calibrage :

Nous commençons par prendre les coordonnées des différents points de la mire par rapport à la caméra, comme illustré sur la figure 3.7. Ensuite, une acquisition est faite pour cette position de la mire, ses coordonnées vont être utilisées pour définir les caractéristiques de la caméra.

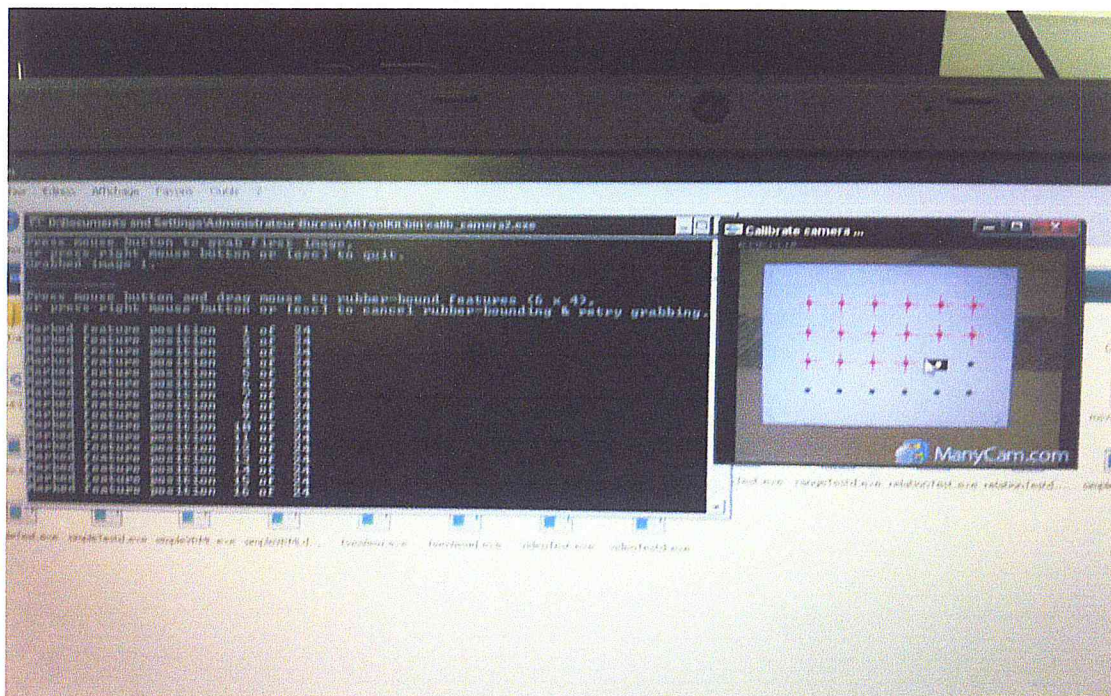


Figure 3.7 : Processus expérimental au moment du calibrage.

Nous avons obtenu les résultats de calibrage correspondant à la caméra utilisé, d'après la sélection du dernier point de la mire (figure 3.8).

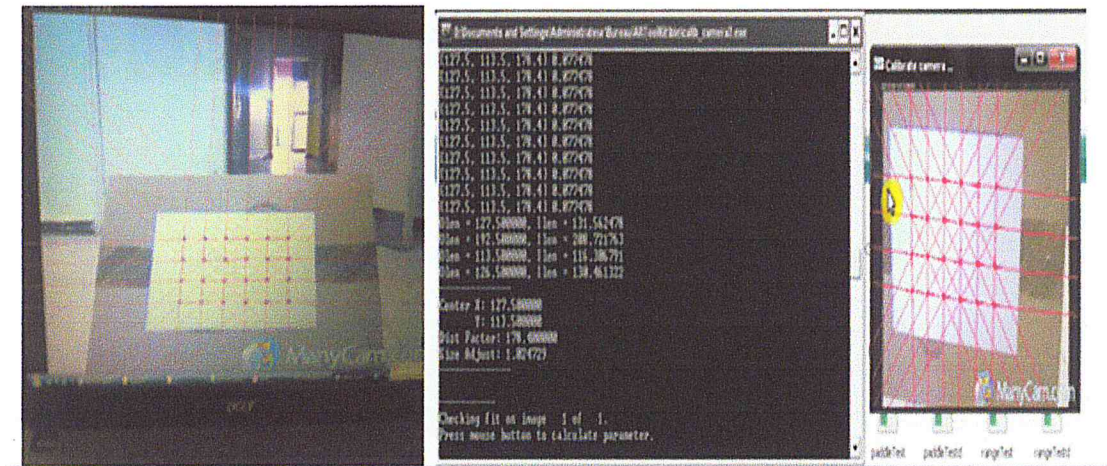


Figure 3.8 : Processus expérimental après la calibration.

4.1.3. Reconnaissance de marqueur :

L'algorithme de détection d'ARToolkit consiste en une suite d'opérations sur chaque image du flux vidéo afin d'une part de repérer la présence d'un marqueur, puis de l'identifier parmi les différents marqueurs chargés dans l'application.

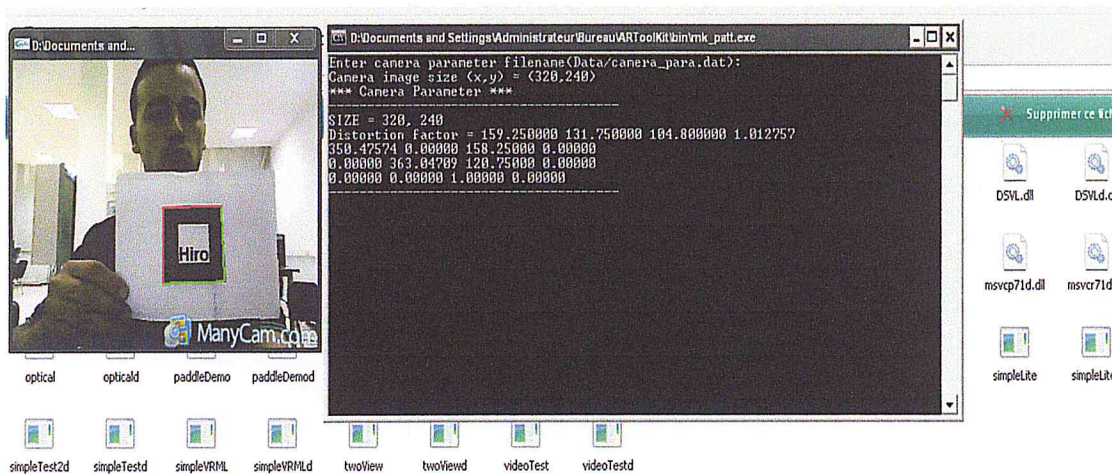



Figure 3.9 : La reconnaissance du marqueur.

5. conclusion :

Nous avons présenté dans cette partie les différentes méthodes de localisation de capteurs visuels en utilisant des marqueurs visuels. Notre choix a été porté sur la librairie Artoolkit pour sa simplicité d'utilisation, et le grand nombre d'applications utilisant cette bibliothèque. Nous nous sommes ensuite intéressés à la calibration de la caméra sans le but d'effectuer des tâches de saisie par un bras manipulateur robotique.



Chapitre IV :
Exploitation de l'architecture
multi-agents existante

1. Introduction :

Ce chapitre est consacré à la présentation de la conception et de l'implémentation de la mission de saisie d'objet par un manipulateur mobile. À cet effet, le langage de modélisation UML a été utilisé. Le chapitre présente aussi l'intégration de cette mission dans l'architecture multi-agents de contrôle de ces robots décrite dans le chapitre précédent.

2. Scénario d'exécution la tâche étudiée par un manipulateur mobile :

1. *Déplacer le robot à une position proche de l'objet à saisir :*

Si l'objet n'appartient pas à l'espace de travail du bras, le robot se rapproche de l'objet à une distance suffisante pour que le bras saisisse l'objet d'une manière autonome. Pour éviter les collisions entre la base et son environnement le dispositif de commande de la base repose sur des mesures à partir du capteur (LMS).

2. *Détection et localisation d'objet*

L'algorithme de détection et localisation d'objet est basé sur le traitement d'image en utilisant la camera IP embarquée et fixée sur l'effecteur (et la librairie ARToolkit) pour déduire les caractéristiques géométriques de l'objet. Cet algorithme permet d'extraire plusieurs informations de l'image.

3. *Déplacé le bras manipulateur de robot jusqu'à atteindre l'objet :*

Cette étape se fait de manière autonome si le marqueur est détecté et si l'objet est dans l'espace de travail du bras.

4. *Saisir l'objet :* déplacer l'objet vers une position finale.

3. conception et Modélisation :

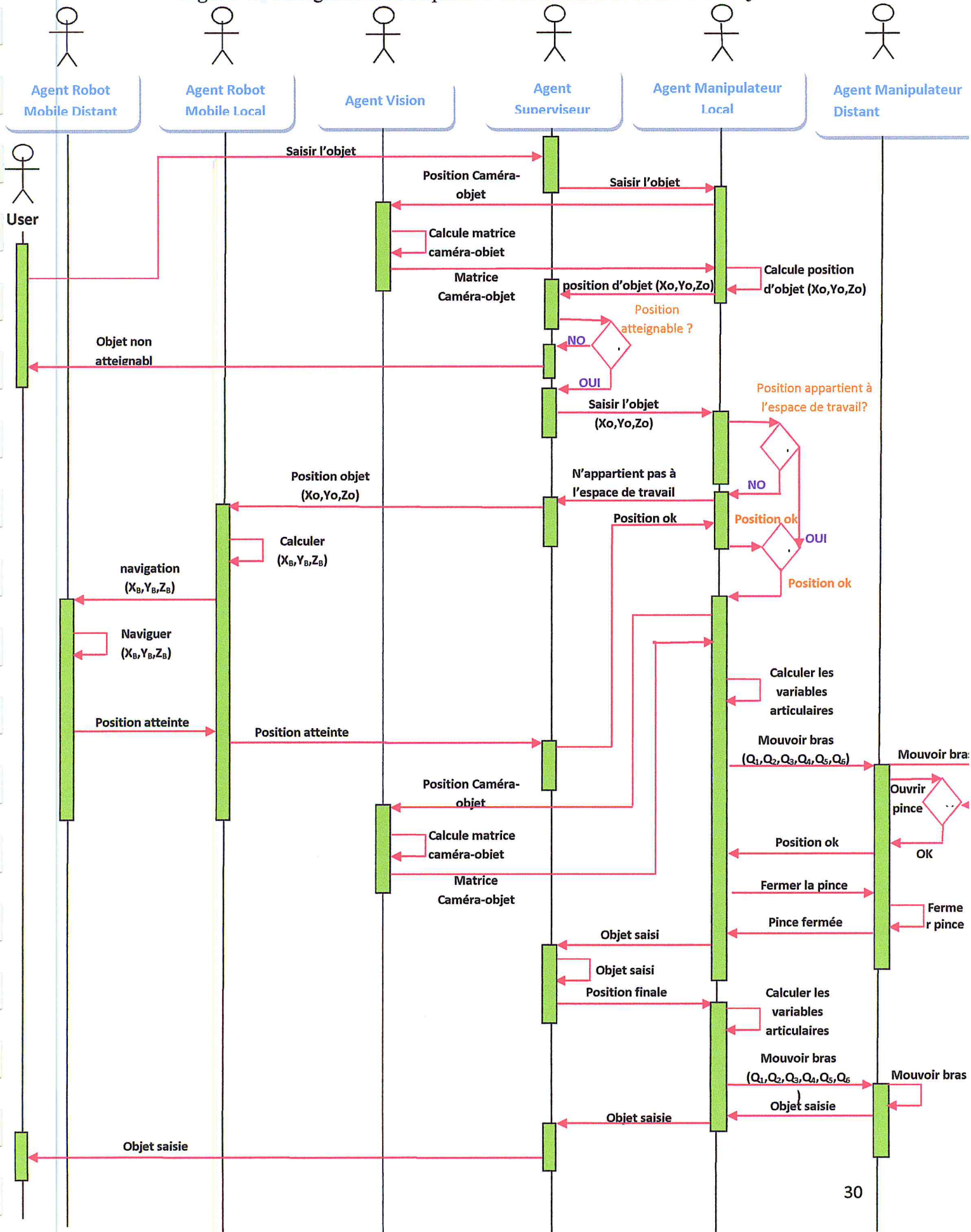
3.1. Diagramme de séquences :

Le diagramme de séquence est un moyen d'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, ARMD, ARML, AV, AS, AML et ARMD.

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 4.1 : Diagramme de séquences de la mission de saisie d'un objet.



Le diagramme de la figure 4.1 montre le déroulement de la mission de saisie d'un objet par un manipulateur mobile, contrôlé par un architecteur multi-agents. Quand l'opérateur humain introduit la mission et lorsque l'agent AS la valide, il envoie à l'agent AV une demande pour localiser l'objet et extraire ses coordonnées 3D. A la fin de cette opération, AV envoie alors les coordonnées d'objet à AS. Si la position de l'objet est atteignable par le robot, l'AS envoie à l'AML une demande de saisie l'objet, sinon l'AS informe l'opérateur que l'objet n'est pas atteignable. Dès que la demande arrive à AML, il teste si l'objet est dans l'espace de travail de bras ou non, dans le premier cas l'AML envoie une demande à ARMD pour atteindre l'objet. Dans le cas contraire, AML envoie la demande à ARML pour déplacer la base mobile et ce dernier calcule la position finale et envoie à l'ARMD de naviguer jusqu'à atteindre cette position, que la base est dans la position finale l'ARMD informe l'ARML que la position est atteinte et l'ARML envoie à l'AV de calculer la nouvelle position de l'objet. L'AV envoie un message à AML qui contient les nouvelles coordonnées d'objet. L'AMD fait la saisie d'objet qu'il reçoit une demande d'AML. Enfin, l'objet est saisi l'AMD informe l'AML et de même l'AML informe l'AS que l'objet est saisi et lui-même informe l'utilisateur.

3.2 Diagramme d'activité :

3.2.1 Diagramme d'activité pour la tâche de saisie :

Ce diagramme présente le déroulement de la mission de saisie d'objet. Cette mission est divisée en deux étapes :

- La localisation d'objet : pour extraire les coordonnées de l'objet à saisir, puis tester si ce que l'objet est atteignable ou non, l'objet appartient à l'espace de travail ou non.
- Déplacer la base : pour mettre l'objet dans l'espace de travail de bras manipulateur.

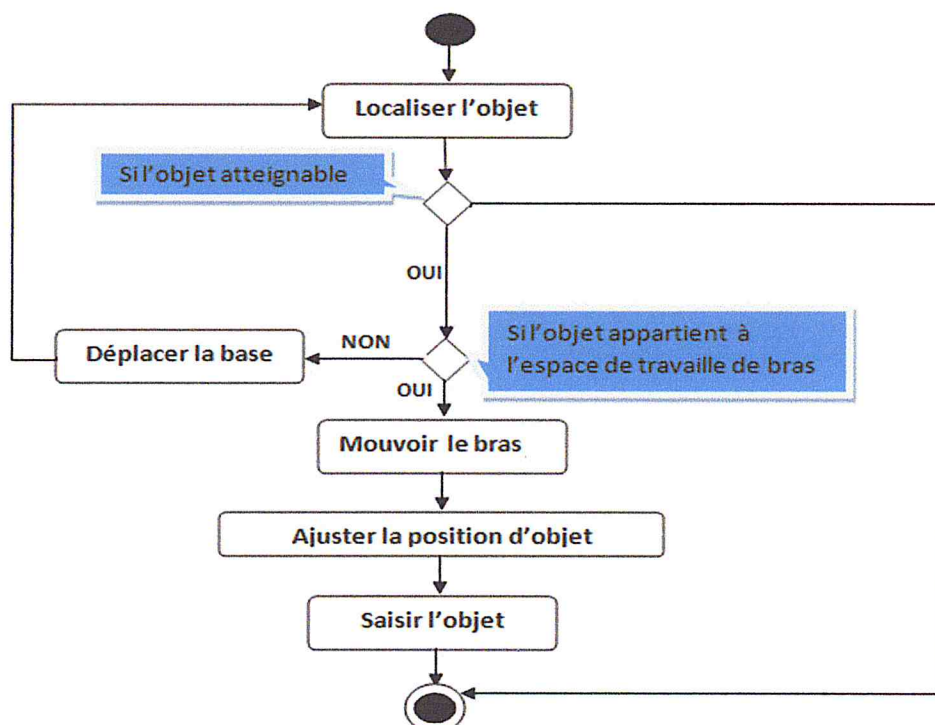


Figure 4.2 : Diagramme d'activité pour la tâche de saisie

3.2.2 Digramme d'activité pour la localisation d'objet :

Ce diagramme présente la procédure de la localisation d'objet au niveau de l'Agent Vision.

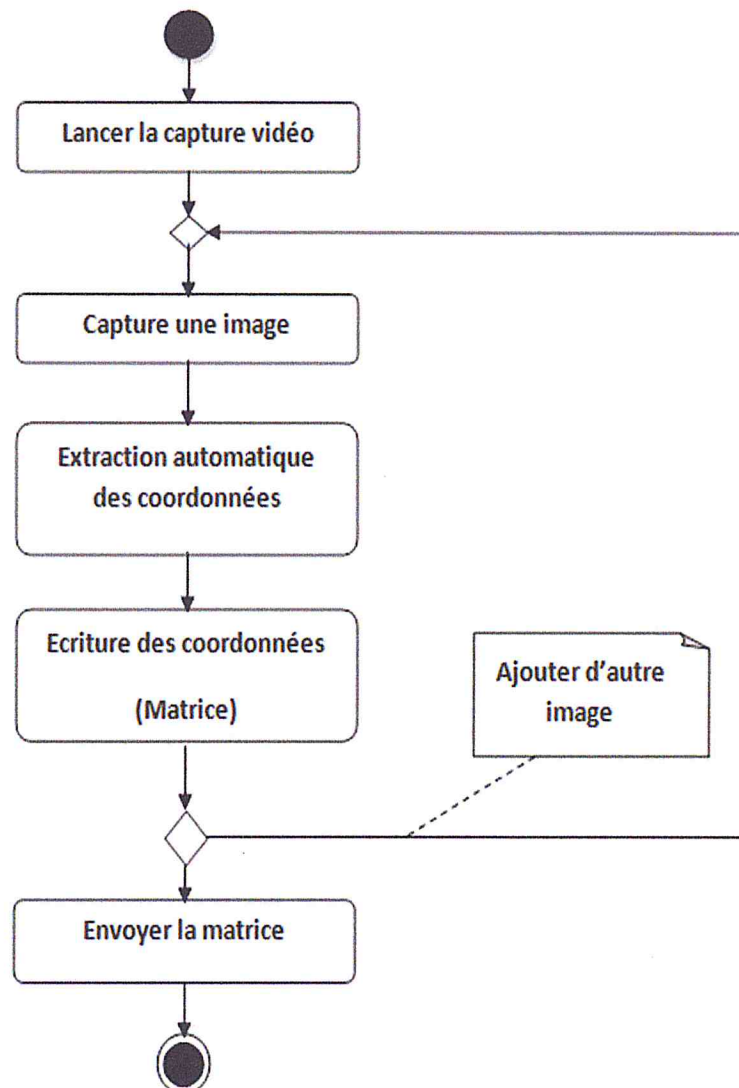


Figure 4.3 : Digramme d'activité pour la localisation d'objet

3.2.3 Digramme d'activité pour le déplacement de la base :

Ce diagramme présente la procédure de déplacement de la base mobile pour maitre l'objet dans l'espace de travail du bras manipulateur. On traite seulement le cas où l'objet n'est pas dans l'espace de manipulation.

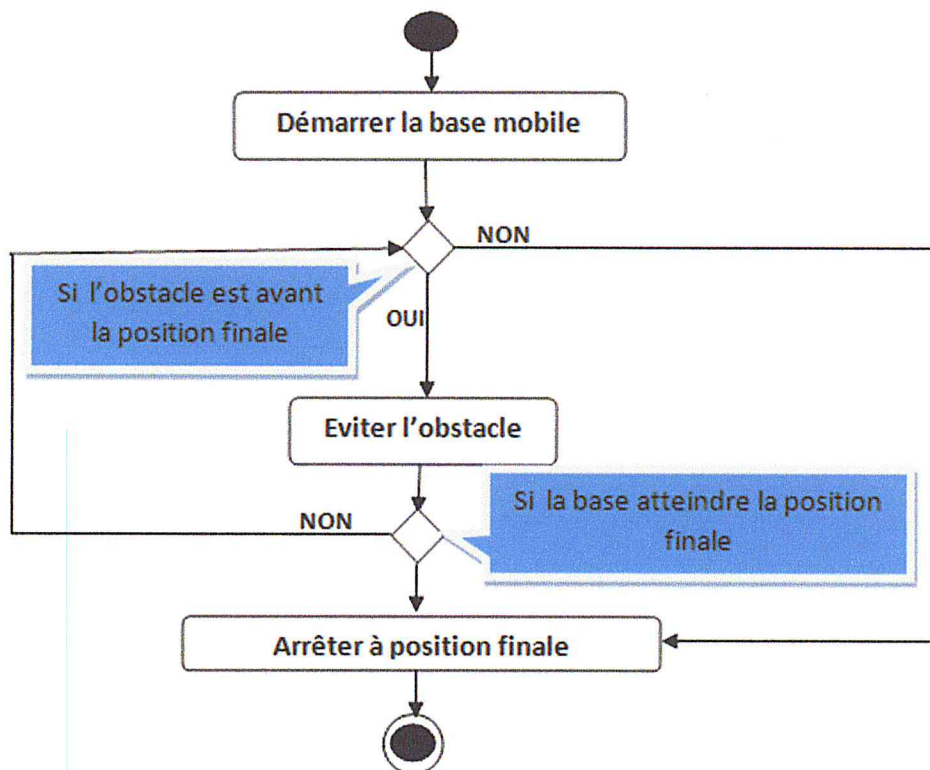


Figure 4.4 : Digramme d'activité pour le déplacement de la base.

Le déplacement se fait par Agent Robot Mobile Distant, il récupère les vitesses consécutives des roues. Quand la base mobile navigue, le capteur LMS retourne les informations (la distance) sur les obstacles qui sont devant la base mobile, grâce à ces informations Agent Robot Mobile teste est ce que l'obstacle est avant la position finale de la base mobile ou non. Si il y a un obstacle la base mobile va basculer et l'éviter après il continue vers la position finale.

4. Déroulement de la tâche de saisie par le manipulateur mobile:

Les agents existe dans le pc distant sont : l'agent robot manipulateur locale (ARMaL), l'agent robot mobile locale (ARML), l'agent de vision (AV), l'agent superviseur (AS).

L'agent robot mobile distant (ARMD) et l'agent robot manipulateur distant (ARMD), sont résident dans le PC embarqué le rôle de ces agents est de mouvoir soit le bras manipulateur, soit la base mobile. Lorsque l'opérateur demande à nce qu'un objet soit saisi, le robot commence la mission :

- L'agent superviseur demande la saisie.
- Le Pc distant reçoit une séquence d'image (30 image par seconde ce qui forme un vidéo) par la caméra sur l'URL « <http://10.1.4.200/axis-cgi/mjpg/video.cgi>.
- L'AV traite chacune des images pour extraire la position et l'orientation du marqueur (x, y, z, phi, teta, psi) par rapport à un repère fixe, à l'aide de la librairie ARToolkit, et envoi ces coordonnées à l'ARMaL.

- L'ARMaL utilise le MGD (voir l'annexe) pour calculer la position de l'objet, et le MGI pour calculer les différentes Q_i ($i=1..6$) correspondant à cette position, d'après un test, l'ARMaL déduit que soit la position de l'objet fait un partie de son espace de travail actuel ou non. Si oui alors envoie un message (POS#Q1#Q2#Q3#Q4#Q5#Q6 \$) à l'ARMaD, si non demander à ARML de mouvoir la base pour atteindre l'objet.
- Si l'agent ARMaD reçoit un message, alors il envoie un signal aux cartes microcontrôleurs qui permet d'envoyer des consignes de vitesses (V_1, \dots, V_6) aux actionneurs des articulations du manipulateur afin de se mouvoir à la configuration désirée.
- Si l'ARML qui reçoit la demande de mouvoir la base, il calcule la position d'objet mais cette fois par rapport à la base mobile, en suite il envoie un message qui contient (X_B, Y_B, θ_B) à ARMD pour exécuter le déplacement.
- L'agent ARMD reçoit les coordonnées (X_B, Y_B, θ_B) et déplace la base mobile vers la position désirée.

5. L'exécution de la tâche de saisie par le bras manipulateur:

Pour saisir un objet il faut connaître sa position par rapport au repère absolu, donc le problème de la saisie c'est un problème de position. Quand on parle sur la position d'objet on parle ici sur les coordonnées d'objet.

Pour calculer les coordonnées d'objet désir à saisie il faut passer du repère absolu (repère bras R_B) jusqu'à le repère d'objet par une suite de changement de repère.

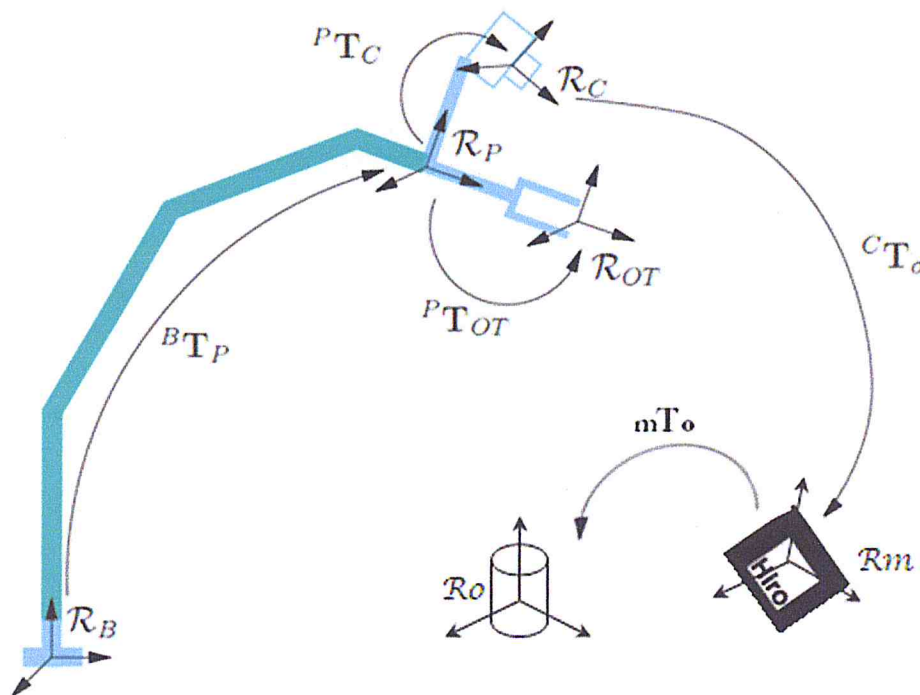


Figure 4.5 : description de problème de localisation d'objet.

RB: repère lié à la base du bras manipulateur.

Ro: repère lié à la position de l'objet. Pour une tâche de saisie, il est défini par l'algorithme de planification de saisie.

RP: repère lié au poignet du bras manipulateur.

RC: repère lié à la caméra.

ROT : repère lié à l'organe terminal (Effecteur).

Rm: repère lié à l'origine de marqueur.

Les différentes transformations qui interviennent :

BTP: transformation qui lie la base du robot à son poignet, elle est donnée par le modèle géométrique direct du robot.

PTOT: transformation qui définit la position de l'organe terminal.

PTC: transformation correspondant aux paramètres extrinsèques de la caméra, elle lie la caméra au repère poignet. Lorsque on parle d'une configuration caméra embarquée, la détermination de cette transformation est appelée calibration pince-caméra dans la littérature.

CTo: transformation calculée par la librairie ARToolkit (voire chapitre 3), elle fournit la position de marqueur par rapport à la caméra.

mTo : transformation qui lie le repère de marqueur (voire chapitre 3) et la position de prise.

5.1. Passage de repère bras au repère pince :

Ce passage se fait au niveau de l'agent AML. La position de l'effecteur (pince) par rapporte à R_B est calculé comme suit :

- Fixer des repères à chaque corps du bras.
- Calculer les matrices homogènes entre chaque corps.
- Calculer la matrice homogène entre la base du bras manipulateur et l'effecteur.

On suppose que le bras est constitué d'un chaînage de $n+1$ corps liés entre eux par 'n' articulations. A chaque corps, on associe un repère R_i .

La matrice homogène entre la base et l'effecteur est donnée comme suit :

$${}^B T_E = {}^M T_2 * {}^2 T_3 * {}^3 T_4 * \dots * {}^{i-1} T_i * {}^i T_E \quad (1)$$

${}^M T_E$ est la transformation matricielle qui définit R_E dans R_M , ${}^M T_2$ définit R_2 dans R_M , ${}^{i-1} T_i$ définit R_i dans R_{i-1} , ${}^i T_E$ définit R_E dans R_i et ${}^{i-1} T_i$ définit le repère R_i dans R_{i-1} .

$${}^{i-1} T_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & d_i \\ \cos(\alpha_i)\sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i) & -r_i\sin(\theta_i) \\ \sin(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\cos(\theta_i) & \cos(\alpha_i) & r_i\cos(\theta_i) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Toutes ces matrices sont données par la figure.. où les termes θ_i , α_i , r_i et d_i , sont appelés les paramètres de Denavit-Hartenberg :

Articulation	α_i (rad)	d_i (mm)	θ_i	a_i
1	0	$d_1=290$	θ_1	0
2	$\pi/2$	$d_2=108,49$	θ_2	0
3	$-\pi/2$	$d_3=113$	0	$a_3=402$
4	$\pi/2$	0	θ_3	0
5	$\pi/2$	$d_4=389$	θ_4	0
6	$-\pi/2$	0	θ_5	0
7	$\pi/2$	$D_{eff}=220$	θ_6	0

Tableau 4.1 : les paramètres de DH modifiés.

Modèle géométrique direct (MGD) :

Le repère se trouve à la base du bras, alors que les repères des différentes articulations se succèdent jusqu'à atteindre celui de l'effecteur. 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 (1).

La matrice de transformation d'espace illustre mathématiquement la situation de l'effecteur relativement à un repère de référence [29].

$${}^B T_E = \begin{pmatrix} r_{11} & r_{12} & r_{13} & tx \\ r_{21} & r_{22} & r_{23} & ty \\ r_{31} & r_{32} & r_{33} & tz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La matrice ${}^B T_E$ c'est la matrice de passage bras-pince.

5.2. Passage de repère pince au repère caméra:

Ce passage se fait au niveau de l'agent AML.

Le problème de changement de repère pince-caméra consiste à déterminer la transformation rigide (translation et rotation) entre le repère de la pince et le repère de la caméra. On utilise le principe de Denavit-Hartenberg pour calculer la matrice de passage pince-caméra.

paramètre	α_i (rad)	d_i (mm)	θ_i	a_i
	$-\pi/2$	130	0	-50

Tableau 4.2 : les paramètres de DH concernant la caméra.

Matrice de passage pince-caméra ${}^P T_c = \begin{pmatrix} 0 & 1 & 0 & 130 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -50 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

5.3. Passage de repère caméra au repère objet :

La caméra est un capteur de vision, donc ce traitement se fait au niveau d'agent AV,

La position d'objet n'est pas fixée par rapport à la caméra donc il faut utiliser un mécanisme qui donne le passage caméra-objet. Pour résoudre ce problème, nous avons utilisé la librairie ARToolkit. Le principe de cette librairie se base sur l'utilisation d'un marqueur, donc pour passer de repère caméra au repère objet il faut passer de repère marqueur. Ce passage se divise en deux étapes :

5.3.1. Passage de repère caméra au repère marqueur :

ARToolkit permet d'extraire plusieurs informations grâce à l'image de marqueur par exemple la position de ce marqueur par rapport à la caméra, mais pour utiliser cette librairie il faut créer des marqueurs bien précis et calibrer la caméra selon ces besoins.

- **Création du marqueur:**

Les marqueurs manipulés par la librairie ne sont pas directement contenus dans des fichiers d'images aux formats standards, mais dans des fichiers de calibre dont le format est spécifique à la librairie. Ces fichiers sont stockés par défaut dans le répertoire *ARToolKit/bin/Data/* et leur nom suit par convention la forme *patt.monPattern*.

Un exécutable (*ARToolKit/bin/mk_patt*) est fourni avec la librairie et permet de générer des fichiers de calibre. Cet utilitaire nécessite de filmer une réalisation du marqueur pour générer le calibre. Cette méthode permet de prendre en compte et donc de compenser les distorsions de la caméra. Il est donc nécessaire d'effectuer une calibration avant d'enregistrer nos propres patterns.

Ce mode de génération de calibre est contraignant mais offre de bons résultats, notamment dans des applications demandant un nombre important de patterns dans des conditions particulières (éclairage, environnement).

Le groupe GDC a cependant développé le programme *patternMaker* spécialement pour la librairie ARToolkit. Cet outil permet de générer automatiquement des milliers de patterns et leurs calibres. Le contenu des patterns générés est un damier 4x4 aléatoire. Le calibre est cette fois générique et n'est donc pas adapté aux conditions d'utilisation.

Un pattern doit nécessairement être composé d'un carré noir délimitant le motif, de plus les motifs ne doivent pas comporter des détails trop précis car en interne la librairie utilise un damier de 16x16 pour reconnaître les motifs. Enfin si l'application nécessite de connaître l'orientation de la caméra, le motif ne doit pas présenter de symétries.

- **Calibration avec ARToolkit:**

Afin d'obtenir les performances optimales de détection des patterns, la librairie permet de compenser les distorsions de la caméra. Les paramètres principaux sont le point central de l'image, le facteur de distorsion et un facteur d'ajustement de la taille de l'image à la

réalité. Les paramètres sont stockés par défaut dans le fichier *ARToolKit/bin/Data/camera_para.dat*. Ces paramètres sont lus au démarrage de chaque application et chaque application peut définir son propre fichier de paramètre.

Que la caméra est bien calibré, et grâce à la librairie ARToolkit on obtient la matrice de passage caméra-marqueur suivante :

$$\begin{pmatrix} r11 & r12 & r13 & tx \\ r21 & r22 & r23 & ty \\ r31 & r32 & r33 & tz \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

r11 : rotation de x (de marqueur) selon x (de camera).

r12 : rotation de y (de marqueur) selon x (de camera).

r13 : rotation de z (de marqueur) selon x (de camera).

r21 : rotation de x (de marqueur) selon y (de camera).

r21 : rotation de y (de marqueur) selon y (de camera).

r21 : rotation de z (de marqueur) selon y (de camera).

r31 : rotation de x (de marqueur) selon z (de camera).

r32 : rotation de y (de marqueur) selon z (de camera).

r33 : rotation de z (de marqueur) selon z (de camera).

(tx, ty, tz) coordonnées de marqueur.

5.3.2. Passage de repère marqueur au repère objet :

Ce passage se calculer avec un simple translation (principe de Denavit-Hartenberg), on fixe l'objet dans une position par rapport au marqueur.

paramètre	α_i (rad)	d_i (mm)	i	a_i
	0	80	0	50

Tableau 4.3 : les paramètres de DH concernant l'objet.

Matrice de passage pince-caméra $M_{T_O} = \begin{pmatrix} 0 & 1 & 0 & 80 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 50 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

A la fin, pour connaître le passage caméra-objet il faut multiplier la matrice de passage caméra-marqueur et la matrice de passage marqueur-objet.

5.4. Calculer la position d'objet dans le repère bras :

L'agent AV est le responsable de calculer la position d'objet par rapport à la caméra (caméra-objet) et il envoie le résultat à l'agent AML (cet agent calcule les passages bras-pince, pince-caméra) qui doit calculer la position d'objet dans le repère bras avec un produit matriciel :

(Bras-objet) = (bras-pince)*(pince-caméra)*(caméra-objet).

Grace à la matrice finale (bras-objet) l'agent AML calcule la position (x,y,z, θ , Φ , Ψ) d'objet dans le repère bras.

$$(\text{Bras-objet}) = \begin{pmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(x, y, z) : sont les coordonnées d'objet dans le repère bras.

(θ , Φ , Ψ) : sont les orientation d'objet dans le repère bras, les angle d'Euler sont utilisés telque :

$$\theta = \text{ATAN 2}(-r_{13}, r_{23})$$

$$\Phi = \text{ATAN 2}(\text{Sin}(\theta) r_{13} - \text{Cos}(\theta) r_{23}, r_{33})$$

$$\Psi = \text{ATAN 2}(-\text{Cos}(\theta) r_{12} - \text{Sin}(\theta) r_{22}, \text{Cos}(\theta) r_{11} + \text{Sin}(\theta) r_{21})$$

L'agent AML utilise cette position (x,y,z, θ , Φ , Ψ) pour calculer les variables articulaires (Q₁... Q₆) puis envoie ces variable à l'agent AMD pour fait la saisie.

5.5. Déplacement de la base mobile :

Si les coordonnée d'objet (XO, YO, ZO) sont en dehors de la zone du travail de bras, la plateforme mobile va se déplacer d'une position initiale Base Init (XBInit, YBInit, θ BInit) jusqu'à atteindre une position finale Base Fin (XBFin, YBFin, θ BFin) qui permet au bras manipulateur d'atteindre l'objet. C'est grâce au système de locomotion du robot mobile (roues) que le bras manipulateur peut être amené dans une zone où l'objet à l'intérêt peut être atteint et saisie par la pince.

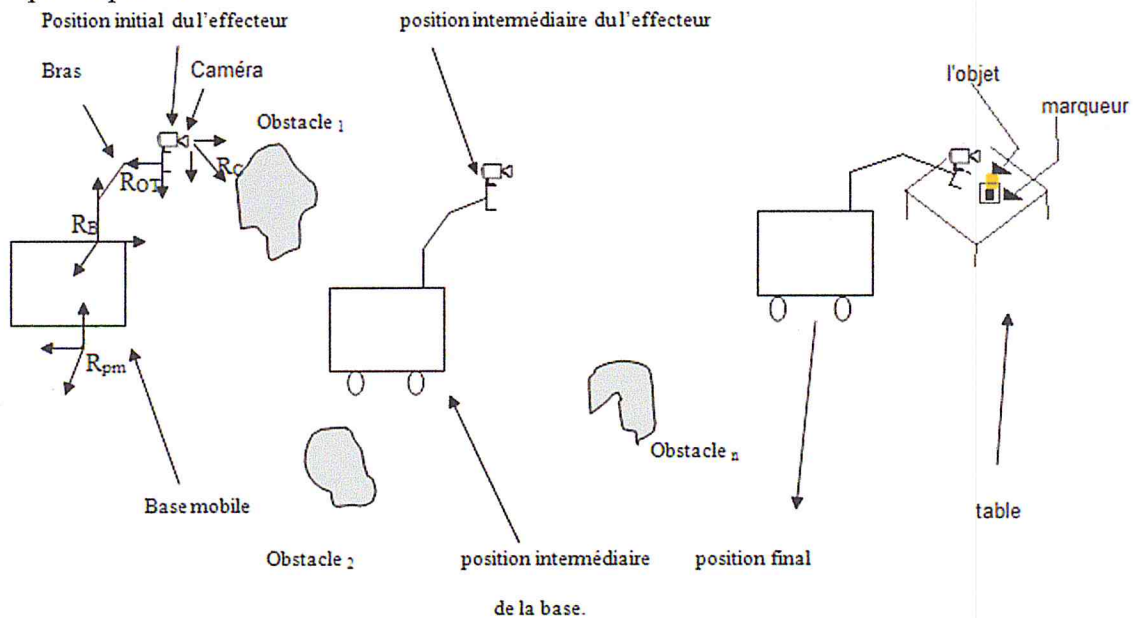


Figure 4.6 : description de problème de déplacement de la base mobile.

- R_{pm} : le repère mobile lié à la plateforme mobile.
- R_b : le repère lié à la base du bras manipulateur.
- R_o : le repère lié à l'effecteur de bras.

- $R_{pm}(0, 0, 0)$: la position initiale de la base mobile par rapporte.
- Position finale : la position finale de la base mobile donnée dans R_{pm} .

Si l'objet n'appartient pas à l'espace de travail de bras manipulateur l'AMD envoie un demande (position d'objet (x_o, y_o, z_o)) à l'agent ARML de déplacer la base mobile, ce dernier va calculer la position qui doit la base atteindre grâce à la position d'objet.

Quand il y a une trajectoire générée hors ligne, ces trajectoires sont soumises au facteur temps. C'est-à-dire que les consignes sont des vitesses envoyées aux moteurs du robot directement. Ces tâches sont exécutées pour un environnement connu du manipulateur mobile.

Lorsque le robot rencontre un obstacle inconnu, la planification qui a été exécuté précédemment n'est plus valable, ce qui fait que le contrôle du robot doit être modifié pour passer à un mode totalement réactif (en ligne). La plateforme va se faire à l'information fournie par le capteur (LMS), et le robot va déplacer grâce au module de navigation.

On va maintenant s'intéresser à la génération de trajectoires, c'est à dire comment faire pour aller d'un point A à un point B, On souhaite donc faire passer le robot d'une position de départ $(X_{int}, Y_{int}, Z_{int})$ à une position d'arrivée. Pour l'arrivée $(X_{fin}, Y_{fin}, Z_{fin})$, on ne spécifie pas d'orientation puisqu'elle ne présentait pas d'intérêt dans notre cas.

La position où se trouve le robot est la position initial $(0, 0, 0)$, l'ARMD calcule la position $P(X, Y, \theta)$ à atteindre par la base mobile et le envoie à l'agent ARMD pour déplacer la base vers cette position, l'ARMD lance la navigation en évitant les obstacles éventuels (s'il existe) jusqu'à atteindre la position P' , informe l'ABML que atteindre la position P' . l'AML calcule la nouvelle position d'objet et l'AMD doit le saisir.

6. Conclusion :

Ce chapitre présente les différentes étapes d'exécution d'une tâche de saisie d'objet par un robot manipulateur mobile contrôlé par un système multi-agent.

Le problème de la saisie est modélisé en UML en utilisant quelques diagrammes, à savoir : le diagramme de séquence et le diagramme d'activité.

Cette mission a été exécutée par le robot dans deux cas distincts selon la distance entre le robot et l'objet :

- Dans le premier cas, le robot se trouve devant l'objet. Ici, le robot ne fait appel qu'à son bras manipulateur.
- Dans le deuxième cas, le robot se trouve à une certaine distance de l'objet. À cet effet, le robot a besoin de sa base mobile afin de se déplacer à une autre position et se rapprocher de l'objet désiré.



Chapitre V :
Résultats Expérimentaux

1. Introduction :

Dans ce chapitre, nous avons présenté les résultats des tests que nous avons effectués pour l'exécution de la tâche de saisie par le robot manipulateur mobile ROBUTER/ULM.

Nous commençons ce chapitre par la présentation de l'environnement matériel et logiciel que nous avons utilisé pour atteindre notre objectif. Nous allons ensuite présenter les résultats obtenus pour que les différentes étapes de travail. En effet nous allons évoquer la localisation d'objet, le mouvement de la base mobile, et enfin la saisie de l'objet.

2. Environnement Matériel :

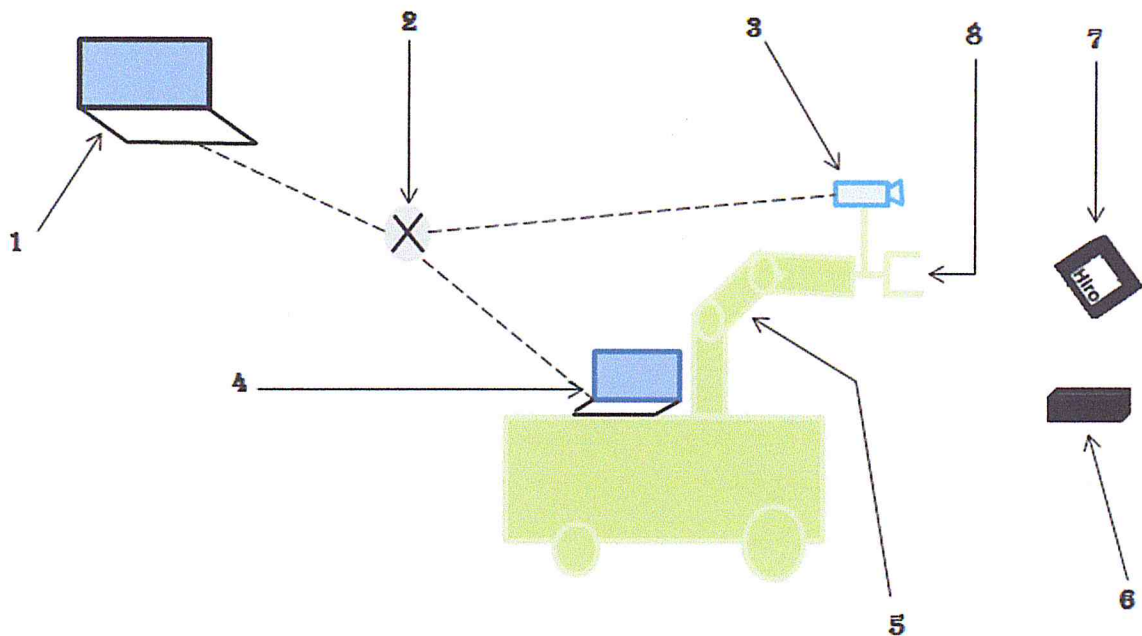


Figure 5.1 : architecteur physique du système.

- 1 : PC distant (l'adresse de ce PC est 10.1.4.158) ;
- 2 : Retour (10.1.4.168) ;
- 3 : Caméra IP axis 213 (10.1.4.200) ;
- 4 : PC embarqué (10.1.4.157) ;
- 5 : Bras manipulateur ;
- 6 : L'objet à saisir ;
- 7 : Le marqueur ;
- 8 : La pince ;

3. Environnement logiciel :

La partie embarquée au niveau du robot a été développée avec le langage C/C++ car l'ordinateur embarqué ne dispose que du compilateur C/C++. deux agents ont été développés à ce niveau, l'agent robot mobile distant, et l'agent robot manipulateur distant.

Les autres agents sont développés sous visual studio c# 2010 au niveau d'un PC distant (off-board).

Dans ce projet, les logiciels ayant été exploités sont : linux redhat 6.2 pour la partie ayant été implémentée sur le système embarqué, et windows 7 pour la partie se trouvant sur le PC hôte. Nous avons également utilisé Matlab et Maple pour les calculs matriciels qui sont relativement complexes.

3.1 L'interface graphique :

L'interface Homme/Robot présentée ci-dessous est implémentée sur le PC off-board. Laquelle permet de contrôler le robot à distance. Cette interface est composée de sept parties principales qui sont réparties comme suit :

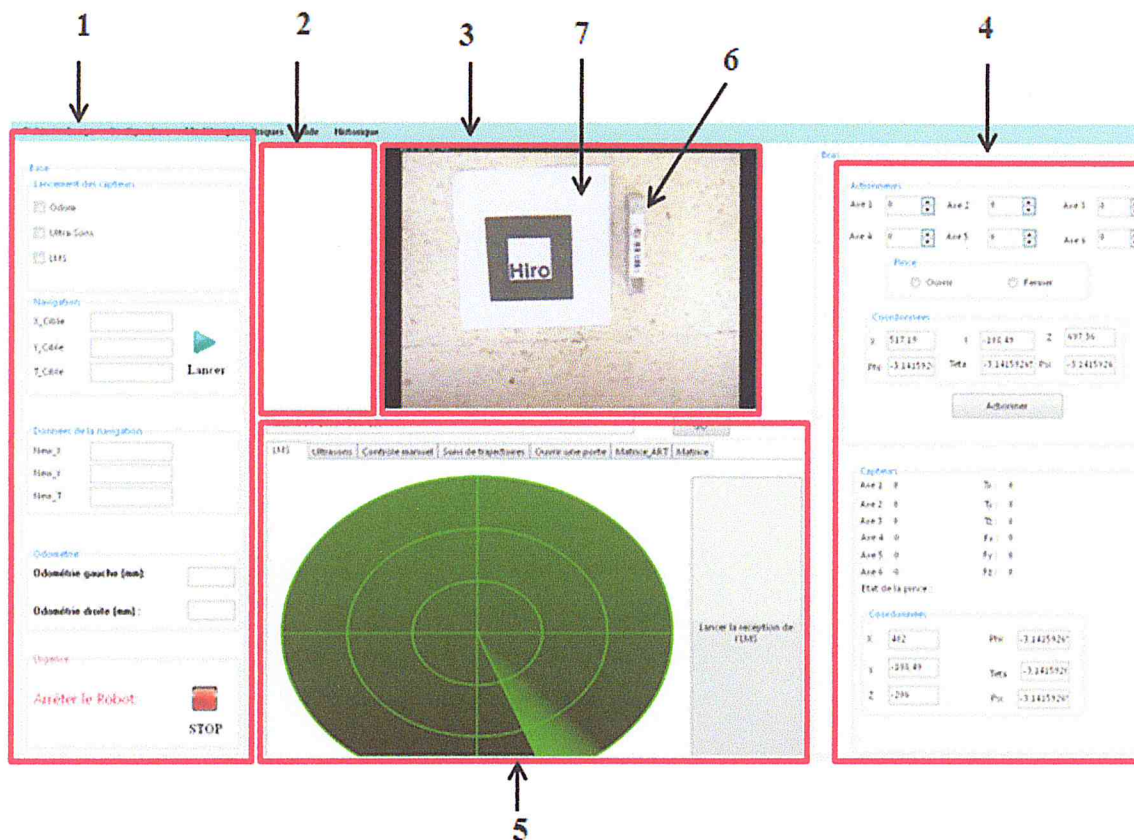


Figure 5.2 : Interface opérateur pour le contrôle de robot.

Partie 1 : cette partie donnée par la figure 5.2 concerne la base mobile. Elle interagit avec l'agent *robot mobile local*. On y retrouve les coordonnées de la position courante de la base mobile (New_X , New_Y , $New_θ$), comme on peut lancer la lecture des capteurs ou ordonner à la base mobile de se déplacer vers une position cible (X_Cible , Y_Cible , $θ_Cible$).

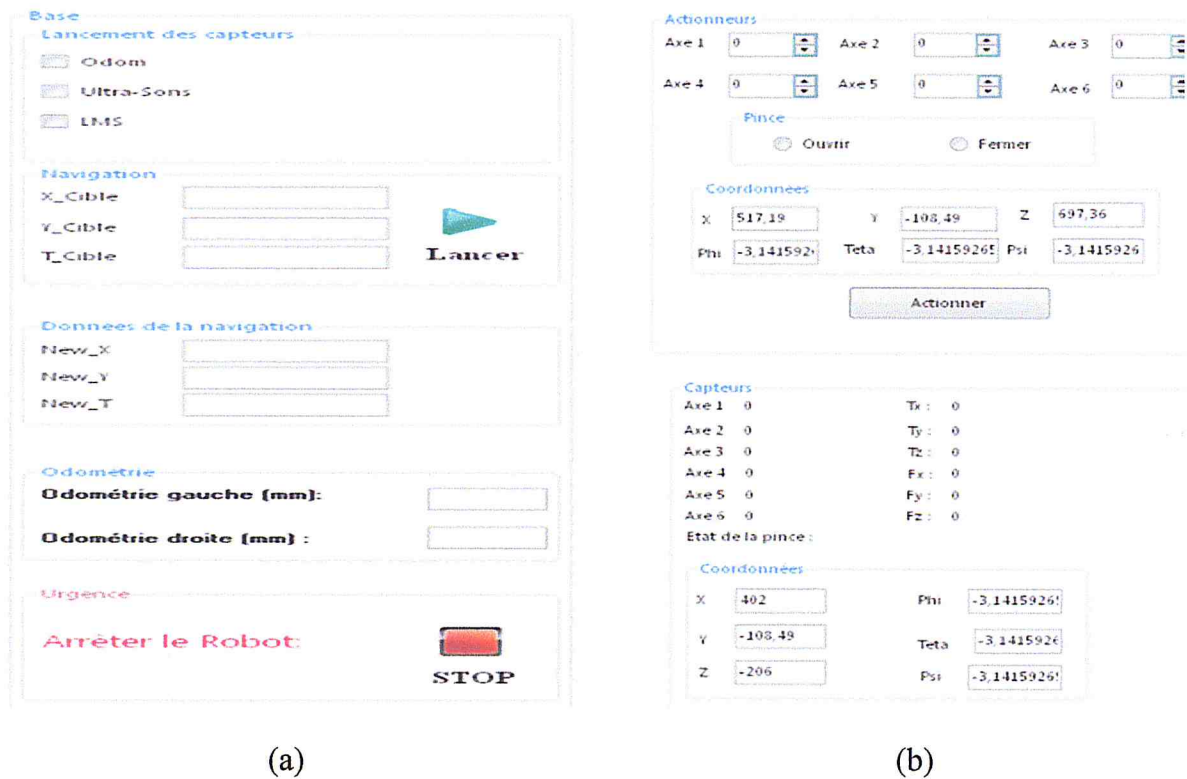


Figure 5.3 : Interfaces dédiées au Contrôle du Robot : (a) Partie concernant la base mobile, (b) Partie Concernant le bras manipulateur.

Partie 2 : cette espace permet l'affichage des positions de l'organe terminale (l'effecteur) après l'exécution de la mission de saisie.

Partie 3 : Partie supplémentaire pour l'affichage de la vidéo acquise par la caméra IP en temps réelle.

Partie 4 : cette partie concerne le bras manipulateur, elle interagit avec l'agent *robot manipulateur mobile local*.

Partie 5 : Cette partie permet l'affichage des données issues du capteur LMS sur un radar et des valeurs des capteurs ultrasons.

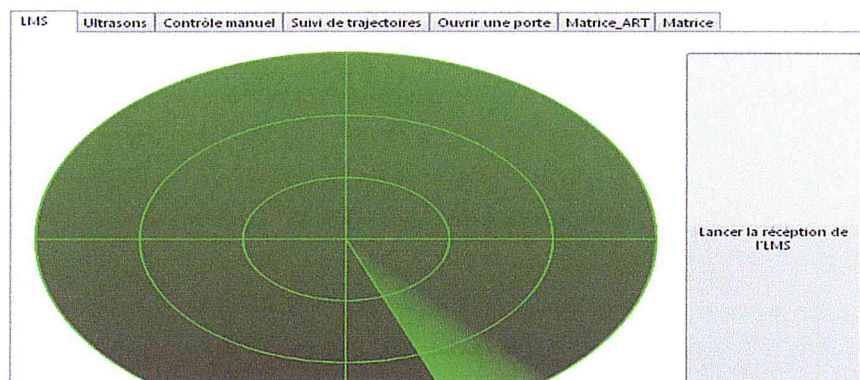


Figure 5.4 : Interface opérateur pour le contrôle du système robotisé.

Partie 6 : le marqueur qui nous permet d'identifier indirectement la position de l'objet à saisir

Partie 7 : l'objet à saisir.

4. Les résultats obtenus :

4.1 Les résultats de mouvement de la plateforme :

Au lancement de la mission de saisie, l'agent de vision déduit que la position de l'objet ne fait pas partie de son espace de travail actuel. Il faut donc mouvoir la base mobile à une nouvelle position plus proche de l'objet.

Pour la phase de déplacement de la base mobile dans l'espace libre, d'un point initial à un point final a été déjà faite, mais elle n'a pas été validée auparavant. Nous avons tenté dans notre travail de faire le basculement entre le control hors ligne et en ligne.

La figure suivante montre l'interface de contrôle du mouvement du robot mobile en mode hors ligne, dans lequel le robot collecte la consigne (les différentes vitesses des roues droite et gauche) d'un fichier texte. Ces consignes sont envoyées à l'agent *robot mobile distant*.



Figure 5.5 : La partie concernant la tâche de déplacement.

Nous avons utilisé un objet C# de type « Timer » pour fixer la cadence d'envoi des vitesses successives aux roues. La position finale du robot est fixée au cas où il y aura basculement vers le mode en-ligne, et que le module de navigation est sollicité.

Les figures suivantes montrent les différentes phases de mouvement de la plateforme mobile.

- La position initiale est la position où se trouve le robot (la position (0, 0, 0)).



Figure 5.6 : état initiale de robot.

- La position finale est déduite par rapport à la position objet (localisation objet).



Figure 5.7 : état intermédiaire de robot.

- Le robot arrête à la position finale.

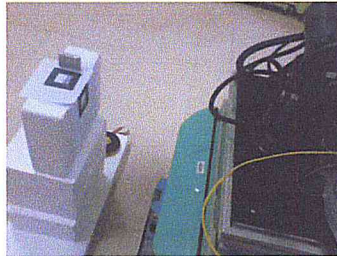


Figure 5.8 : état finale de robot.

4.2 Résultat de saisie :

Nous avons utilisé un la librairie ARToolkit afin de détecter la présence du marqueur dans chaque trame du flux vidéo en temps réel. Nous détectons au départ la position et l'orientation du marqueur par rapport à la camera (Matrice ARToolki). Et après le changement de repère que nous avons présenté précédemment, nous déduisons la position et l'orientation du marqueur par rapport à un repère fixe (base du bras). La position et l'orientation de l'objet à saisir se réfère à la position du marqueur. En effet, il suffit d'ajouter une matrice de transformation.

Nous avons constaté que les conditions extérieurs relatives au marqueur jouent un rôle important (luminosité, résolution, distance, mise au point...) et qu'il peut évidemment être déplacé dans toutes les directions, soumis néanmoins à un degré de liberté limité (ce dernier devant rester visible par la caméra).



Figure 5.9 : La partie concernant la tâche de saisie.

Nous avons affiché sur l'interface présentée dans la figure 5.9 différentes matrices de passage telles que :

- ❖ *Matrice ARToolkit* : permet l'affichage de la matrice pour signifier la position et l'orientation du marqueur par rapport à la camera.
- ❖ *Matrice pince* : permet de définir la position de l'effecteur par rapport au repère du bras.
- ❖ *Position du marqueur* : c'est la matrice permettant la définition de la position de marqueur par rapport à repère de bras.
- ❖ *Saisie* : c'est un objet de type «Button » qui permet donner l'ordre au robot d'exécuter la tâche de saisie l'objet.

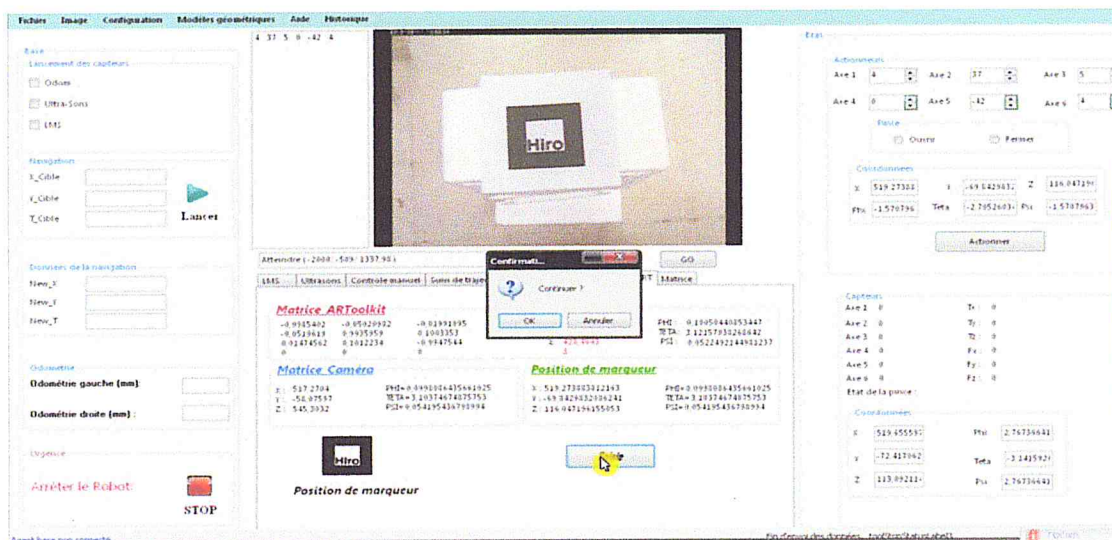


Figure 5.10 : la tâche de saisie



a. Mouvement du bras vers l'objet :

Les images suivantes (Figure 5.11) montrent les résultats obtenus pour la tâche de saisie d'un objet appartenant à l'espace de travail du bras manipulateur

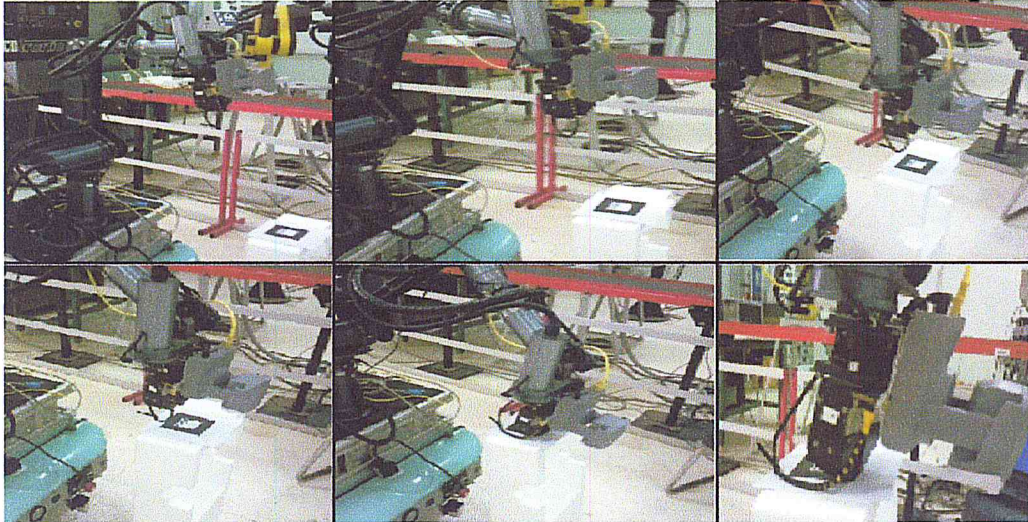


Figure 5.11 : les étapes pour saisir un objet.

b. La saisie l'objet :

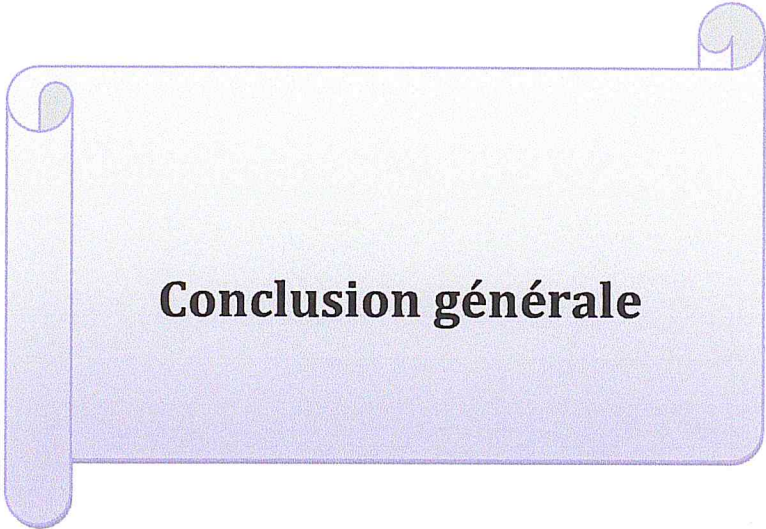


Figure 5.12 : la position finale de la pince.

5. Conclusion :

Ce chapitre a été consacré à la validation expérimentale de notre projet pour le robot manipulateur mobile Robuter/ULM. Nous avons réservé la première partie à la description technique du système robotique sur lequel nous avons travaillé. Nous avons également décrit ses caractéristiques ainsi que son environnement logiciel afin de comprendre son fonctionnement.

Nous sommes passés, ensuite, à la présentation des résultats. En effet, nous avons présenté les interfaces permettant le control de la plateforme mobile pour le déplacement vers une position proche de l'objet à atteindre. Nous avons ensuite présenté également la partie logicielle qui concerne le control du bras manipulateur pour l'exécution de la tâche de saisie.



Conclusion générale

Conclusion générale

De nombreux laboratoires de robotique travaillant sur les *plateformes mobiles* ou les *bras manipulateurs* se dotent actuellement des manipulateurs mobiles. Si, du point de vue des perspectives d'utilisation, ces systèmes découplent les possibilités, ils offrent également de nouveaux enjeux scientifiques en raison de leur redondance et de la difficulté de modéliser les différentes tâches ou missions qui peuvent leur être assignées. En effet, enchaîner des tâches de navigation et de manipulation, décomposer en sous-tâches, décider de la coordination du bras manipulateur et de la plateforme mobile, sont autant de problèmes à résoudre.

L'objectif de ce travail était de réaliser la mission de saisie d'un objet par un manipulateur mobile contrôlé par architecture logicielle multi-agents. Cette architecture de contrôle est composée de six agents hybrides répartis sur trois niveaux hiérarchiques. Aussi, à chaque agent est affecté un ensemble de compétences.

- Le niveau supérieur est composé d'un seul agent (*Agent Superviseur*). Cet agent fait office d'interface entre l'architecture de contrôle du manipulateur mobile (le système multi-agents) et l'opérateur.
- Le deuxième niveau hiérarchique est composé de trois agents. Un agent est dédié au contrôle de chaque sous-système robotique. L'agent *Robot Mobile Local* pour le contrôle de la base mobile, l'agent *Robot Manipulateur Local* pour le contrôle du bras manipulateur et l'agent *Vision* pour le contrôle du système de vision. Ces agents coopèrent et coordonnent leurs actions afin de réaliser les différentes opérations correspondant à la mission allouée au robot par l'opérateur.
- Le troisième et dernier niveau hiérarchique est composé de deux agents. Un agent pour contrôler chaque sous-système robotique. L'agent *Robot Mobile Distant* pour le contrôle des mouvements de la base mobile et l'agent *Robot Manipulateur Distant* pour le contrôle des mouvements du bras manipulateur.

Les deux premiers niveaux hiérarchiques sont installés sur le PC off-board du *RobuTER/ULM*, alors que le dernier niveau est installé sur le PC on-board du robot.

La conception de la mission de saisie d'un objet a été intégrée à cette architecture de contrôle. Cette conception a été effectuée en utilisant le langage UML et en exploitant deux types de diagrammes :

- *Diagramme d'activité* : la mission de saisie d'un objet est décomposée en un ensemble d'opérations. Chacune de ces opérations nécessite la coopération entre agents.
- *Diagramme de séquence* : ce diagramme montre le déroulement de la mission de saisie d'un objet par un manipulateur mobile contrôlé par une architecture multi-agents.

L'application développée est valable avec les petits objets, deux cas possibles ont été distingués pour la mission de saisie d'un objet selon que l'objet en question se trouve dans l'espace atteignable du manipulateur ou en dehors de cet espace :

- *Dans le premier cas* : la mission consiste à saisir l'objet désiré 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 saisir l'objet.
- *Dans le deuxième cas* : la mission consiste à saisir d'un objet qui fait de l'espace de travail du bras manipulateur. Ici, la mission ne nécessite que le mouvement du bras manipulateur afin de pouvoir saisir l'objet.

Conclusion générale

Cette mission implique tous les agents de l'architecture de contrôle. Comme continuité à ce travail, la première chose est de finaliser la partie concernant la localisation de l'objet en utilisant le système de vision. La deuxième perspective consiste à utiliser ce dernier système pour l'ajustement de la pince sur l'objet de. La dernière perspective consiste à développer la commande du bras manipulateur permettant de suivre une trajectoire imposée à l'effecteur du robot pour déplacer l'objet vers une position finale.

bibliographie

[1] : V.Padois, “*Enchainements Dynamique De Tâche Pour Des Manipulateur Mobile A Roues*”, thèse de doctorat, institut national polytechnique de Toulouse, France, 2005.

[2] : BENSARKHI, H et DERBOUA, Détection des bords de la chaussée par vision monoculaire pour un robot mobile de type voiture. Université Saad Dahleb, Blida, PFE, 2009.

[3] : DERRADJI, M, Capteurs de proximité d'un robot Bipède, rapport de stage eINRIA,Grenoble, 2006.

[4] : http://fr.wikipedia.org/wiki/Bras_manipulateur

[5] : <http://projetar.renouv.fr/presentation-artoolkit/>

[6] : I.Akli, *Elaboration d'une stratégie de coordination de mouvements pour un manipulateur mobile redondant*, mémoire du magistère, USTHB, 2007.

[7]: Bernard Bayle, “*Robotique mobile*”, article, Université de Strasbourg.

[8] : J. Pichon, « *Guidage visuel d'un robot mobile autonome d'inspiration bionique* », thèse de doctorat de l'institut nationale poly technique de Grenoble, 1991.

[9] : David Filliat, Robotique mobile, École Nationale Supérieure de Techniques Avancées ParisTech, 2011.

[10] : Arnaud Canu, “*Introduction à la robotique*”, Article, Greyc-Cnrs, Université de Caen, France, 2008/2009.

[11] : http://en.wikipedia.org/wiki/Domestic_robot

[12] : http://fr.wikipedia.org/wiki/Robot_m%C3%A9dical

[13]S.Gray, S.Chitta, V.Kumar, M.Likhachev, “A Single Planner for a Composite Task of Approaching, Opening and Navigating through Non-spring and Spring-loaded Doors”, 2012.

[14] : M.Trabelsi, N.Aitoufroukh, S.Lelandais, « Saisie d'objets à l'aide d'une caméra et des ultrasons pour la robotique de service », Université d'Evry, 2005.

[15] : N.Mansard, O.Stasse, F.Chaumette1, K.Yokoi, « Saisie guidée par la vision pendant la marche », INRIA Rennes, France, 2006.

[16] : C. Dune, C. Nadeau, « Vers une saisie en un seul clic : caractérisation de la

forme 3D d'un objet à partir d'informations visuelles », IRISA, 2009.

[17] : Abou Moughlbay A, Martinet P, Mansard N, « Commande par vision d'un robot redondant multi bras : Manipulation à deux bras avec le HRP2 », LAAS, Université de Toulouse, 2010.

[18] : V.Quang Huy, « Implémentation d'une architecture à base de comportements pour robot mobile », institut de la francophonie pour l'informatique, Rapport final du TPE, 2009.

[19] : A.KIMOUICHE, A.AITER, Architecture Multi-agents de Contrôle des Manipulateurs Mobiles : Ouverture d'une porte. USTHB, 2011.

[20] : Goulven GUILLOU, Architecture multi-agents pour le pilotage automatique des voiliers de compétition Et Extensions algébriques des réseaux de Petri, UNIVERSITÉ DE BRETAGNE OCCIDENTAL, 2010.

[21] M.BENBOALI, « supervision et contrôle d'un robot manipulateur mobile à base d'un système multi-agents », USTHB, 2009.

[22] : David COCHARD, *ARToolKit Plugin For Virtools 4*, Rapport du projet, l'université de technologie compiégne, 2007.

[23]: MA Portela Sotelo, J-M Moreau, « Développement d'un module de suivi de mouvements et localisation d'un patient par rapport à un ensemble d'équipements dans une salle de traitement par radiothérapie », Rapport Interne, université Claude Bernard Lyon, 2009.

[24]: J.Gobert,N.Renous, *réalité augmentée à l'Ensicaen*, rapport de mini projet, École nationale supérieure d'ingénieurs de Caen & centre de recherche,2010.

[25]: <http://tpe-la-realite-augmentee.e-monsite.com/pages/iii-application-avec-un-logiciel/a-presentation.html>

[26] : http://en.wikipedia.org/wiki/Domestic_robot

[27]: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.1987.

[28] : M.ABouskia, Localisation spatiale d'objets en vue du contrôle/commande d'un manipulateur mobile. Cas du RobuTER/ULM, USTHB, 2009.

[29]: Gabriel Falcao, Natalia Hurtos, Joan Massich, *Plane-based calibration of a projector-camera system, VIBOT Master 2008*.

[30] : J.Ferber, « Les systèmes multi-agents: Vers une intelligence collective. InterEditions », Paris, 1995.

[31] : Hichem RAHAB, Une approche à base d'agents adaptatifs pour la résolution des systèmes complexes, UNIVERSITE HADJ LAKHDAR BATNA, 2011.

[32] : Encarta 2009.

[33] : http://www.memoireOnline.com/m_strategie-de-rendez-vous-dans-les-systemes-multi-agents1.html

[34] : Laëtitia Matignon, Introduction à la robotique, Université de Caen, 2012.

[35] : http://www.memoireOnline.com/m_strategie-de-rendez-vous-dans-les-systemes-multi-agents1.html

[36] :E.Bernard, « Communication et langages de communication dans les SMA » Aix-Marseille Université (AMU), 2012.

[37] : TRINH Thanh Hai, Utilisation et optimisation de la communication entre agents dans les cas de tâches collaboratives ou concurrentes, IF'I, 2007.

[38] : Pierre AVANZINI , Modélisation et Commande d'un Convoi de Véhicules, thèse de doctorat, UNIVERSITÉ BLAISE PASCAL - CLERMONT II, 2010 .

[39]: 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, Italy, 2003.

[40]: I.Akli, A.Hantout, B. Bouzouia , Control modes validation for Generalized Application: Robuter/UML Mobile ManipulatorvPoint to Point Task Executions CDTA, 2012.

1. Agent :

1.1 Définition d'un Agent :

De nos jours, le mot « agent » est utilisé dans plusieurs domaines et, de ce fait, plusieurs sens lui sont attachés. D'ailleurs, même à l'intérieur du domaine de l'informatique, plusieurs chercheurs ont défini le concept d'agent de manières différentes. L'une des définitions d'un agent qui fait consensus dans la communauté des SMA est donnée par [30]. Un agent est une entité informatique qui possède un comportement individuel, caractérisé principalement par quatre propriétés :

- Autonomie ou pro activité : capacité à agir sans intervention extérieure, prise d'initiative.
- Sensibilité : capacité à percevoir l'environnement ou les autres agents.
- Localité : limitation de la perception et des actions.
- Flexibilité : réaction aux changements perçus.

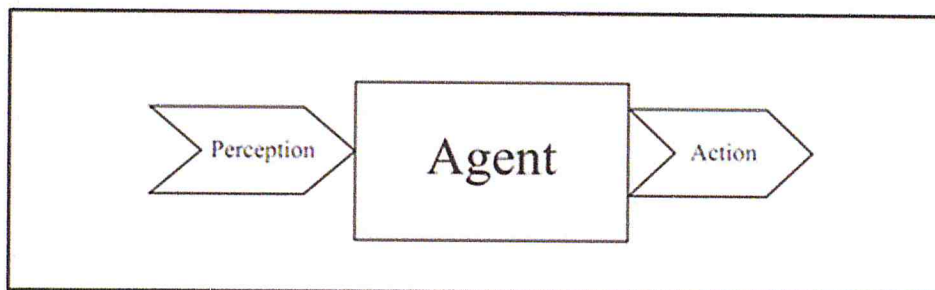


Figure A.1 : Représentation d'un agent [31].

1.2 Principales caractéristiques des agents :

Il existe différentes sortes d'agents, selon qu'il possèdent une ou plusieurs des caractéristiques suivantes : [32]

- capacité à communiquer, directement ou non, avec d'autres agents ;
- capacité de se reproduire ou d'être reproduit ;
- vue locale de l'environnement ;
- représentation partielle de cet environnement ;
- comportement téléonomique : comportement tendant à satisfaire un but, qui est défini par ses tendances ;
- comportement réflexe : comportement ne faisant pas intervenir de facteurs internes (contrairement au comportement téléonomique).

L'utilisation simultanée de plusieurs agents dans un même environnement permet de définir un système multi-agent.

1.3 Les type d'agents:

Les experts des systèmes multi-agents ont classifié ces derniers (agents) en deux grandes catégories selon un critère essentiel qui est la représentation de son environnement, et sont donc les agents réactifs et les agents cognitifs, et les systèmes dits hybrides [33].

1.3.1. Agent réactif :

Agents sans intelligence (sans anticipation, sans planification) qui réagissent par stimulus-réponse à l'état courant de l'environnement. Des comportements intelligents peuvent émerger de leur association.

- Pas de représentation explicite.
- Organisation implicite/induite
- Communication via l'environnement.

Fonctionnement basé sur une simple correspondance entre les situations et les actions (interagissent avec leur environnement mais sans raisonner dessus).

Le robot exécute une action selon l'état actuel de l'environnement dans lequel il évolue (le mapping état/action étant fixé initialement) [34].

1.3.2 Agent cognitif :

Les systèmes d'agents cognitifs sont fondés sur la coopération d'agents capables à eux seuls d'effectuer des opérations complexes. Un système cognitif comprend un petit nombre d'agents qui disposent d'une capacité de raisonnement sur une base de connaissances, d'une aptitude à traiter des informations diverses liées au domaine d'application, et d'informations relatives à la gestion des interactions avec les autres agents et l'environnement. Chaque agent est assimilable, suivant le niveau de ses capacités, à un système expert plus ou moins sophistiqué.

1.3.3 Agents hybrides

Ce type d'architecture combine les agents réactifs et cognitifs, qui sont généralement distribués sur plusieurs niveaux ou couches. La couche de haut niveau, délibérative, rassemble des agents purement cognitifs, s'occupe du raisonnement et de la prise de décision du système. La couche de bas niveau ne rassemble que des agents réactifs qui exécutent généralement des tâches élémentaires sous les ordres de la couche supérieure ou par leurs propres initiatives. La (les) couches intermédiaires, peuvent regrouper les deux types d'agent (réactifs et cognitifs), le nombre de couches intermédiaires dépend du modèle du système à concevoir [35].

2. Système multi-agents :

2.1 Définition :

La plupart du temps, un agent n'est pas seul dans son environnement, il y a d'autres agents présents autour de lui. Les agents doivent, par conséquent, être capables d'interagir entre eux. Ils peuvent soit coexister, coopérer ou être en compétition. S'ils ne font que

coexister, alors chaque agent ne fait que considérer les autres agents comme des composantes de l'environnement. S'ils coopèrent, alors les agents doivent pouvoir communiquer et se coordonner pour agir efficacement ensemble. S'ils sont en compétition, alors les agents doivent être en mesure de négocier.

Un système multi-agent, est un modèle informatique composé d'entités artificielles capables de communiquer, directement ou non, entre elles et situées dans un environnement avec lequel elles interagissent [35].

2.2 Interaction et coopération entre agent :

Une des principales propriétés de l'agent dans un SMA est celle d'interagir avec les autres agents. Ces interactions sont généralement définies comme toute forme d'action exécutée au sein du système d'agents et qui a pour effet de modifier le comportement d'un autre agent.

Elles permettent aux agents de participer à la satisfaction d'un but global. Cette participation permet au système d'évoluer vers un de ses objectifs et d'avoir un comportement intelligent indépendamment du degré de complexité des agents qui le composent.

2.3 La communication dans les SMA

La communication est la base de la résolution coopérative des problèmes. Elle permet de synchroniser les actions des agents et de résoudre les conflits de ressources et de buts par la négociation [30]. La communication est à la base des interactions et de l'organisation sociale d'un SMA. Les agents peuvent interagir soit en accomplissant des actions linguistiques (en communiquant entre eux), soit en accomplissant des actions non-linguistiques qui modifient leur environnement. En communiquant, les agents peuvent échanger des informations et coordonner leurs activités. Deux stratégies principales ont été utilisées pour supporter la communication entre agents :

- Les agents peuvent échanger des messages directement.
- Ils peuvent accéder à une base de données partagées (appelée tableau noir ou "Blackboard") dans laquelle les informations sont postées [35].

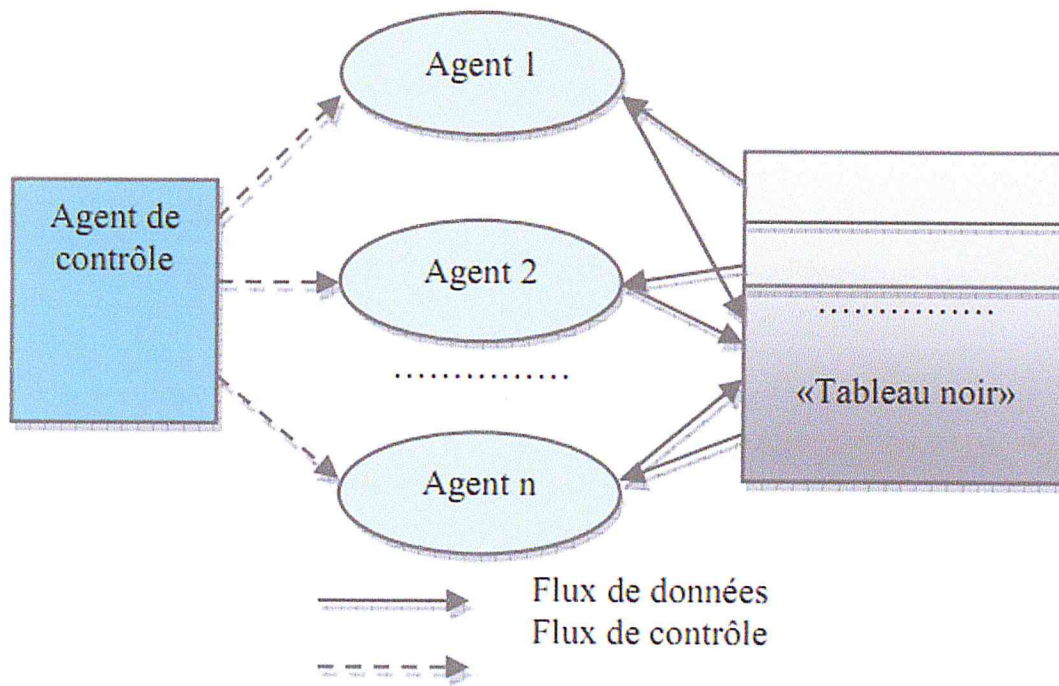


Figure A.2 : les stratégies de communication entre les agents [35].

2.4 Langage de communication :

2.4.1 KQML :

KQML est un langage et protocole de communication de haut niveau, orienté message pour l'échange d'information entre agents. Développé par le DARPA (Knowledge Sharing Effort) pour supporter l'interopérabilité entre les agents intelligents dans une application distribuée [36].

Exemple de message KQML [37]:

```
(tell
  : receiver A
  : sender B
  : in-reply-to id0
  : reply-with id1
  : language Prolog
  : ontology meteo
  : content "temps (nuageux)"
)
```

2.4.2 ACL-FIPA :

Le langage ACL de FIPA [51], est fondé également sur la théorie des actes de langage et a bénéficié grandement des résultats de recherche de KQML. Si l'approche du langage ACL est globalement semblable à celle de KQML, un grand soin a été apporté aussi bien à

la description formelle de la sémantique des actes de communication qu'à l'introduction de protocoles régissant les règles d'échanges de messages [37] .

Exemple : [37]

```
(inform
  :sender agent1
  :receiver hpl-auction-server
  :content
    (price(bid good02) 150)
  :in-reply-to round-4
  :reply-with bid04
  :language sl
  :ontology hpl-auction
)
```

2.5 Protocole de communication :

Un protocole est une stratégie de niveau élevé suivie par des agents logiciels qui interagissent avec d'autres agents. Nombreuses missions exécutées par le robot doivent se dérouler en temps réel. Il est donc nécessaire de mettre en place un protocole de communication simplifié afin de respecter les contraintes d'exécution de certaines missions [21].

1. Description de MM robuter/ULM :

Le robot manipulateur mobile est un système articulé à six articulations monté sur un système mobile. Les caractéristiques techniques sont les suivants :

- Base mobile (2 roues motrices, 2 roues folles);
- Direction par différentiel de vitesse
- Ceinture 24 capteurs US;
- Bras ultra-léger 6 axes;
- Pince électrique à deux doigts ;
- Capteur d'efforts 6 axes;
- Caméra IP Axis 213;
- Système de transmission vidéo sans fil ;
- PC embarqué MMX 233;
- 3 boutons d'arrêt d'urgence;
- 4 batteries de 12V;
- Entrée Joystick;
- Liaison Ethernet sans fil;
- Capteur LMS;
- Contrôle du robot via des cartes MPC555 de Robosoft ;
- OS : linux redhat 6.2 avec RTAI V-1.3;
- Environnement de programmation Syndex V 5.1.

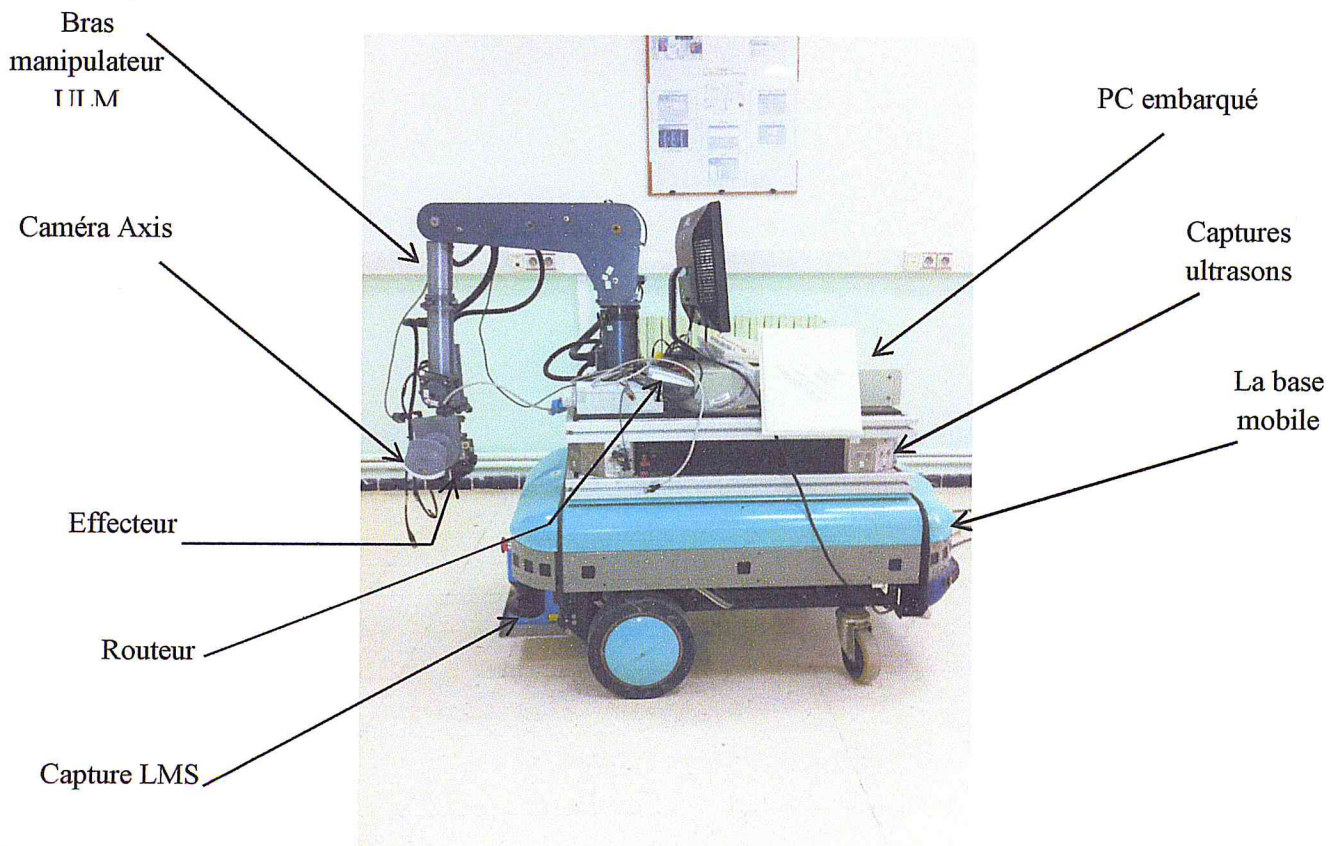


Figure B.1 : Description de robuter/ULM.

2. Les capteurs du robot :

- Capteurs Ultrasons (US)

RobuTER dispose d'une ceinture de 24 capteurs ultrasons. Chaque capteur est formé par un émetteur et un récepteur ultrason (Fig. B.2).



Figure B.2 : Capteur ultrasonore (Émetteur/Récepteur).

- Capteur LMS SICK200 :

Le LMS SICK200, donné par Figure.5 .3, est installé sur la base mobile RobuTER. C'est un scanner de mesure dans le plan horizontal avec un angle de 0° à 180° [Victorino et al, 2001].



Figure B.3 : Capteur LMS SICK200.

3. Description de la caméra utilisée :

La caméra réseau AXIS 213 PTZ constitue une solution avancée de gestion à distance, dotée de fonctions panoramique/inclinaison/zoom et pouvant être pilotée à partir de n'importe quel PC connecté au réseau local ou à Internet. Elle est idéale pour la surveillance de personnes et de propriétés (à l'intérieur et à l'extérieur) et pour l'apprentissage et le dépannage à distance.

Cette caméra réseau polyvalente dispose d'un zoom optique 26x, d'un objectif de mise au point automatique et d'un zoom numérique 12x. La caméra offre une couverture particulièrement large grâce aux fonctions panoramiques à 340 degrés,

aux possibilités d'inclinaison à 100 degrés et au zoom permettant d'agrandir le moindre détail.

La camera AXIS 213 PTZ délivre simultanément des flux vidéo MPEG-4 et Motion JPEG, ce qui permet l'optimisation de la qualité de l'image et de la bande passante.

Le mode de compression MPEG-4 est idéal dans le cas où la bande passante disponible est limitée et que des taux d'image plus élevés sont nécessaires.

La caméra peut être contrôlée manuellement ou programmée pour alterner entre une image couleur le jour et une image en noir et blanc dans des conditions de basse luminosité ou de nuit, en utilisant l'éclairage infrarouge intégré ou une lampe infrarouge externe pour les longues distances.

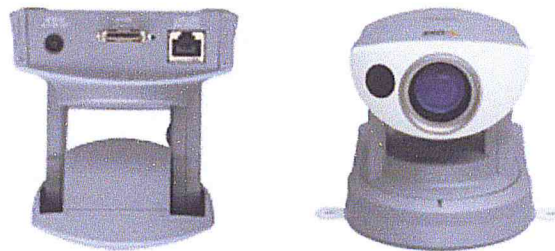


Figure B.4 : Caméra axis 213.

1. Génération de trajectoire:

On considère un robot mobile évoluant dans son espace de travail. Le problème général de planification est de déterminer le mouvement le plus pertinent selon un critère d'optimalité pour joindre une configuration initiale A donnée à une configuration finale B. Il s'agit au premier abord d'un problème combinatoire classique dans la discipline Informatique : il faut tout d'abord définir l'ensemble des séquences de configurations connexes (sans collision) qui mènent de A à B, puis choisir la solution optimale parmi les combinaisons dont le nombre croît de pair avec la complexité du système. De nombreuses contraintes sont à considérer comme les caractéristiques du robot (géométrie, cinématique), l'environnement et le type de tâche à effectuer [38].

1.1 type de trajectoire:

Les trajectoires peuvent être classées en plusieurs catégories (figure 3.1). La catégorie monodimensionnelle correspond à la trajectoire d'un degré de liberté du robot. La trajectoire multidimensionnelle est associée soit à un ensemble de degrés de liberté du robot (planification articulaire) soit à l'organe effecteur (planification cartésienne). Par opposition à la planification monodimensionnelle, la planification multidimensionnelle peut nécessiter de synchroniser temporellement l'ensemble des axes du robot. Les trajectoires qui sont définies par deux points (les conditions initiales et finales) sont des trajectoires point à point. Les trajectoires multipoints, quant-à-elles, passent par des points intermédiaires ou approximent un ensemble de points comme dans le cas des chemins. On distingue la génération de la trajectoire calculée dans l'espace articulaire du Robot de celle dans l'espace cartésien.

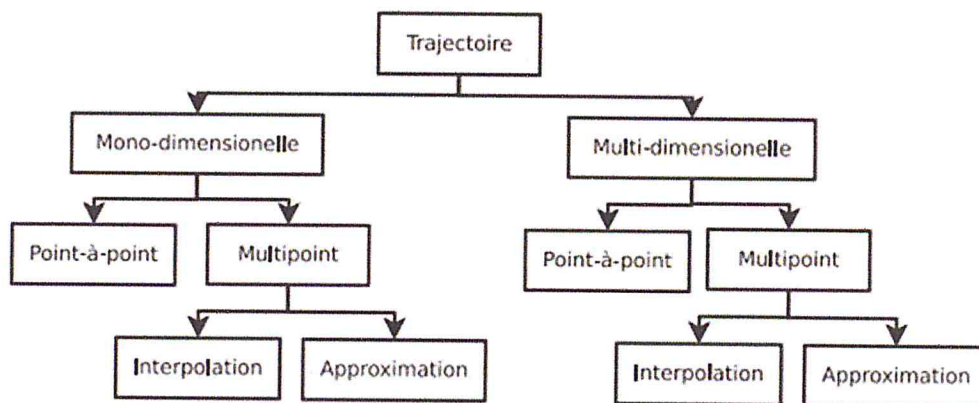


Figure C.1 : les catégories de trajectoire.

1.2 Génération de trajectoire des manipulateurs mobiles pour l'exécution de tâches point à point :

Cette catégorie de problèmes est classique en robotique. En matière de robots mobiles, on la retrouve parfois sous l'appellation de navigation. S'agissant de manipulateurs mobiles, le problème consiste à trouver une stratégie de déplacement d'une configuration initiale vers une configuration finale (le robot est au repos au point p_0 et doit terminer sa trajectoire avec une vitesse nulle en p_1 . Au point p_0 correspond des valeurs de position aux variables articulaires de q_0 , de même pour p_1 , on a q_1).

On distingue ici deux techniques [39]:

Les méthodes à base de planification, qui consistent à prédéterminer une solution au problème point à point, étant donné la structure de l'espace de travail et les contraintes agissant sur le système ;

- les méthodes réactives, qui permettent de générer les commandes du système sur la base de données perceptuelles.

Les manipulateurs mobiles posent nombre de difficultés pour aborder les problèmes point à point. Etant des systèmes d'ordre généralement élevé, ils posent le problème de la vitesse des algorithmes envisagés. Comment faire évoluer un manipulateur mobile? De manière découplée, en utilisant les connaissances sur les différents sous-systèmes, ou de manière coordonnée, quitte à revoir les principes de résolution?

2. contrôleur de la base mobile :

Le mode réactive est utilisé comme une technique fiable pour le comportement fusion. Dans ce travail, les suppléments de robots mobiles parmi cible atteindre, et des comportements d'évitement d'obstacles, les règles sont conçus pour mettre en œuvre ces comportements.

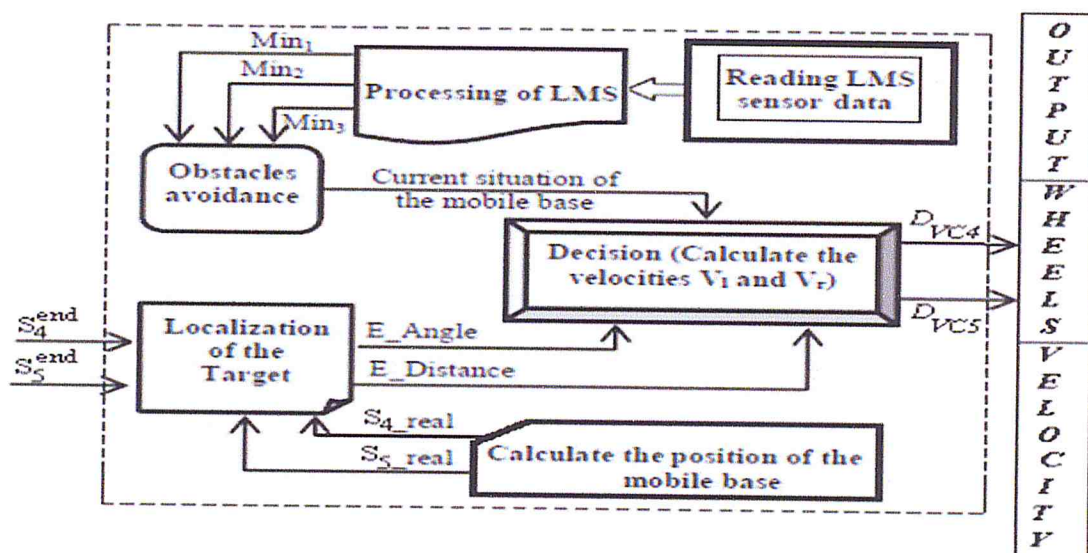


Figure C.2 : contrôle réactive de système mobile [40].

Les instructions DVC4 et DVC5 sont envoyés aux roues motrices (C.2). Les données fournies par les LMS de capteurs de mesure laser (indiquant la présence / absence d'obstacles) et la situation finale de la base mobile sont entrés. Les coordonnées cartésiennes du robot à roues sont convertis en E_Angle en coordonnées polaires (angle entre l'axe de déplacement du robot mobile et l'OP reliant l'axe et envoyer) et $E_Distance$ (distance entre OP et envoyer). L'espace de travail de la robuter est également subdivisé en trois zones $Zone_i$ ($i = 3$), afin de calculer les distances mini entre le robot et l'obstacle.

La situation réelle ($S4_Real$, $S5_Real$) est calculé selon aux données odométriques afin de calculer l'erreur de position E_Angle_Err et $E_Distance_Err$ d'erreur angulaire.

