

MA-004-391-1

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université SAAD DAHLAB Blida
Faculté des Sciences
Département d'Informatique



Mémoire de Fin d'Etudes

Mémoire Présenté par :

KHALAL Khadidja

CHARIF Meriem

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Informatique

Option : Ingénierie de logiciel

Thème :

***Exploitation d'un simulateur 3D pour la
téléopération: application au robot manipulateur mobile
RobuTER/ULM»***

Soutenu le :

Mme. MANCER. Y

M. OUELD AISA. A

M.KAMECHE. A

Mme. AKLI. I

Présidente

Examineur

Promoteur

Promotrice

Année universitaire : 2016/2017

MA-004-391-1

Remerciements



En premier lieu,

Nous tenons remercier notre Dieu pour la patience qui nous offre pour Finaliser ce travail dans les bonnes conditions.

Nous exprimons notre plus grande reconnaissance à notre promotrice **Mme AKLI Isma** pour le Temps, la patience et l'attention qu'elle a consacrés au bon déroulement de ce travail.

Nous remercions chaleureusement notre promoteur **Mrs KAMACHE Abdallah**

D'avoir accepté de diriger ce travail, pour leur disponibilité, leurs Critiques Constructives et leurs suggestions pertinentes.

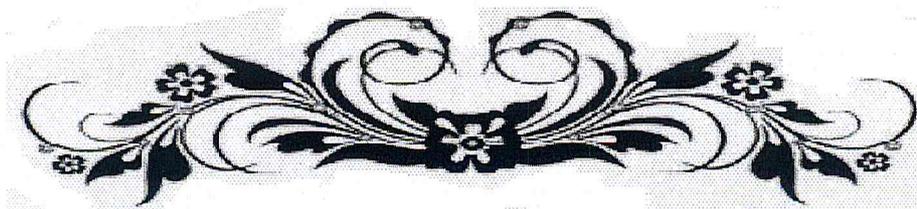
Nous remercions s'adressent également à **Monsieur le précédent de jury** et

Les membres du jury pour l'honneur qu'ils nous font d'avoir bien voulu étudier ce Travail et de le juger.

Nous tenons à remercier toutes **les familles** et **amis** qui nous ont aidé tant moralement que matériellement. A **nosparents** pour le courage et l'amour qui nous

Ont caractérisés durant tout ce temps de vie étudiante.

Nous n'oublions pas d'adresser un grand merci à **tous les enseignants, toutes les personnes qui ont contribué de près et de loin à l'enrichissement** et à notre épanouissement intellectuel durant tout ce parcours universitaire aux **Département Mathématique & Informatique** de l'Université de SAAD DAHLEB Bledia.



Dédicace

A la mémoire de mon Père...

Je dédie cet événement de ma vie à la mémoire de mon père disparu trop tôt. J'espère que, du monde qui est le sien maintenant, il apprécie cet humble geste comme preuve de reconnaissance de la part d'une fille qui a toujours prié pour le salut de son âme. Puisse Dieu, le tout puissant, l'avoir en sa sainte miséricorde !

A ma très chère mère...

Affable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études. Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte.

A mes très chères frères et sœurs : «Nadîr et Raouf et Farouk, Meroïne, Awatif, Samîra et Faïza»

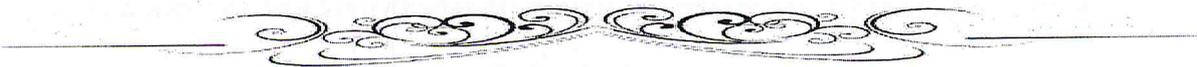
Mes chères frères et sœurs présentes dans tous les moments doux et difficile dans ma vie par votre soutien moral et vos belles surprises sucrées. Je vous souhaite un avenir plein de joie, de bonheur, de réussite et desérénité. Je vous exprime à travers ce travail mes sentiments de fraternité et d'amour. A tous les membres de ma famille, petits et grands. Veuillez trouver dans ce modeste travail l'expression de mon affection.

A ma chère binôme Khadîdja

En témoignage de l'amitié qui nous uni et des souvenirs de tous les moments que nous avons passé ensemble, je vous souhaite une vie pleine de santé et de bonheur.

A mes chères ami(e)s : «Aïda ; Ikram et Sara »

Je ne peux trouver les mots justes et sincères pour vous exprimer mon affection et mes pensées, vous êtes pour moi des frères, sœurs et des amis sur qui je peux compter. En témoignage de l'amitié qui nous uni et des souvenirs de tous les moments que nous avons passé ensemble, je vous dédie ce travail et je vous souhaite une vie pleine de santé et de bonheur.



Meriem

Résumé :

La simulation est aujourd'hui dans le contexte de la robotique mobile est un outil incontournable pour l'étude et la conception du système robot. Dans le cadre du stage au Centre de développement des Technologie Avancée (le laboratoire Robotique de la division productive et Robotique) .il est demandé la conception et la réalisation d'un système de processus communication bidirectionnelle entre un simulateur 3D et un robot manipulateur mobile réel (Robuter/ULM). Avec différents modes de contrôle à savoir (la simulation online, la manipulation online, Simulation l'offset la manipulation offline).

Pour atteindre cet objectif, nous avons tout d'abord fait une étude conceptuelle, enrichie par des diagrammes UML .Ensuite, nous Avons implémenté Notre système. En utilisant C++ et Python comme langage de programmation, et ROS comme un système d'exploitation pour robot.

Mots-clefs : Robot, Robuter/ULM, Simulateur 3D, Simulation, Manipulation, Simuler, ROS, Téléopération, Gazebo,

Abstract :

The simulation is today in the context of mobile robotics is a must-have tool for the study and design of robot systems. During the internship at the Advanced Technology Development Center (the Robotics Laboratory of the Productivity and Robotics Division), the design and realization of a bidirectional communication system between a 3D simulator and a real mobile robot (Robuter / ULM). With different modes of control, namely (online simulation, online manipulation, offline simulation and offline manipulation).

To achieve this goal, we first made a conceptual study, enriched by UML diagrams. Then we implemented our system. Using C ++ and Python as a programming language, and ROS as an operating system for robots.

Keywords: Robot, Robuter / ULM, 3D Simulator, Simulation, Manipulation, Simulate, ROS, Teleoperation, Gazebo,

ملخص:

المحاكاة الآن في سياق الروبوتات المتنقلة هو أداة يجب أن يكون لديك لدراسة وتصميم أنظمة الروبوت. خلال التدريب في مركز تطوير التكنولوجيا المتقدمة (مختبر الروبوتات من قسم الإنتاجية والروبوتات)، تصميم وتنفيذ نظام الاتصالات ثنائي الاتجاه بين الروبوت الروبوت (روبوتر / ULM) ومحاكاة 3D. مع وسائط مختلفة من السيطرة، وهي (محاكاة على الانترنت، والتلاعب على الانترنت، حاليا محاكاة والتلاعب حاليا).

ولتحقيق هذا الهدف، قمنا أولا بإجراء دراسة مفاهيمية ، أثرتها مخططات ULM. ثم قمنا بتنفيذ نظامنا. استخدام ++ C و بيثون كلغة برمجة، و روس كنظام تشغيل للروبوتات.

Table des matières

| | |
|------------------------------|---|
| Introduction General : | 2 |
|------------------------------|---|

Chapitre 1 : Généralité sur la robotique :

| | |
|--------------------------------------|----|
| 1. Introduction | 5 |
| 2. Robotique Générale : | 5 |
| 3. Les Composants d'un Robot : | 6 |
| 4. Les Types Des Robots : | 7 |
| 5. Conclusion | 22 |

Chapitre2: La simulation

| | |
|----------------------------------|----|
| 1. Introduction:..... | 24 |
| 2. La Simulation : | 24 |
| 3. Le Simulateur:..... | 27 |
| 4. L'environnement Gazebo :..... | 32 |
| 5 .La Téléopération :..... | 34 |
| 6. Conclusion | 34 |

Chapitre.3: Conception

| | |
|-------------------------------------|----|
| 1. Introduction | 37 |
| 2. Principe de travail : | 37 |
| 3. Architecture de Système : | 41 |
| 4.Conception de Notre Système | 43 |
| 5. Conclusion: | 47 |

Chapitre4 : Implémentation.

| | |
|--|----|
| 1. Introduction..... | 52 |
| 2. Outil de Développement..... | 52 |
| 3. Test et Validation de L'application : | 57 |
| 4. Conclusion | 62 |
| Conclusion General | 64 |
| Référence | 65 |

Table des figures :

I. Chapitre : 01.Généralité sur la robotique :

| | |
|--|----|
| Figure [1.1] : Robot Mobile | 7 |
| Figure [1.2] : Robot de type unicycle..... | 8 |
| Figure [1.3] : Robot de type tricycle | 9 |
| Figure [1.4] : Robot de type voiture..... | 10 |
| Figure [1.5] : Robot de type omnidirectionnel..... | 10 |
| Figure [1.6] : Un bras manipulateur | 11 |
| Figure [1.7] : L'espace de travail de robot cylindrique. | 12 |
| Figure [1.8] : L'espace de travail de robot rectiligne. | 12 |
| Figure [1.9] : L'espace de travail de robot sphérique..... | 13 |
| Figure [1.10] : L'espace de travail de robot articulé..... | 13 |
| Figure [1.11] : Robot SCARA 4 Axes | 14 |
| Figure [1.12] : Robot Manipulateur Mobile..... | 15 |
| Figure [1.13] : Robot Manipulateur mobile MM-400..... | 16 |
| Figure [1.14] : Robot Manipulateur mobile MM500..... | 16 |
| Figure [1.15] : Robot Manipulateur mobile MM-700..... | 17 |
| Figure [1.16] : Présentation du manipulateur mobile Robuter/ULM..... | 18 |
| Figure [1.17] : Robot au bloc opératoire..... | 20 |
| Figure [1.18] : RobotDomestique..... | 21 |
| Figure [1.19] : RobotOuverture de porte..... | 21 |
| Figure [1.20] : Robot Saisie d'Objet..... | 22 |

II. Chapitre : 02.La simulation:

| | |
|--|----|
| Figure [2.1] : Un utilisateur commande un robot virtuel pour déplacer l'objet manipulé en utilisant un dispositif haptie. Le dispositif permet à l'utilisateur d'interagir et de contrôler la tâche de manipulation. Alors que le robot humanoïde ajuste de façon autonom..... | 25 |
|--|----|

| | |
|--|----|
| Figure [2.2] : cette séquence d'images montre l'interaction dynamique complexe entre deux figures humanoïdes et un grand nombre d'objets dans la scène. Les nombreux contacts entre les objets dans cette scène et leurs mouvements résultants sont calculés interactive me | 26 |
| Figure [2.3] : Logo De GAZEBO | 30 |
| Figure [2.4] : Pioneer mobile robot equipped with a camera and Hokuyo laser range finder. | 32 |
| Figure [2.5] : Structure du système de Téléopération. | 33 |

III. Chapitre : 03.Conception:

| | |
|--|----|
| Figure [3.1] :Simulateur 3D Qui Existe | 38 |
| Figure [3.2] : Architecture de perception et de Génération de Mouvement. | 39 |
| Figure [3.3] : Protocole de communication..... | 39 |
| Figure [3.4] : Modèle de l'architecture proposée..... | 40 |
| Figure [3.5] : Schéma Globale De L'architecture (Cas Simulation)..... | 41 |
| Figure [3.6] : Schéma Globale De L'architecture (Cas Manipulation)..... | 42 |
| Figure [3.7] : Schéma Globale De L'architecture (Cas Simuler)..... | 43 |
| Figure [3.8] : Diagramme de cas d'utilisation Globale..... | 44 |
| Figure [3.9] : Diagramme de cas d'utilisation Controller Le Robot Réel..... | 45 |
| Figure [3.10] : Diagramme de cas d'utilisation Controller Le Simulateur. | 46 |
| Figure [3.11] : Diagramme de Séquence Cas De Simulation. | 47 |
| Figure[3.12] : Diagramme de Séquence Cas De Manipulation. | 48 |
| Figure[3.13] :Diagramme de Séquence Cas De Simuler..... | 49 |
| Figure [3.14] : Diagramme de déploiement général..... | 50 |

IV. Chapitre : 04.Implémentation :

| | |
|---|----|
| Figure [4.1] : interface et logo d'ubuntu..... | 52 |
| Figure [4.2] : logo de C++..... | 53 |
| Figure [4.3] : Logo de Python..... | 53 |
| Figure [4.4] : Logo de ROS. | 54 |
| Figure [4.5] : Logo d'outil ROS. | 55 |
| Figure [4.6] : Roscore..... | 57 |
| Figure [4.7] : le nœud du serveur..... | 57 |

| | |
|--|-----------|
| Figure [4.8] : le nœud du System_man..... | 58 |
| Figure [4.9] : l'environnement de gazebo..... | 58 |
| Figure [4.10] : le nœud du C_bras..... | 59 |
| Figure [4.11] : le nœud du subs_j_s..... | 59 |
| Figure [4.12] : Mode Simulation..... | 60 |
| Figure [4.13] : Mode Simuler (Pos1)..... | 61 |

Table des tableaux :

I. Chapitre.2. La simulation:

Tableau [2.1] : *Comparaison entre les Simulateurs Commerciaux*.....31

Tableau [2.2] : *Comparaison entre les Simulateurs Open-Source*.....32

II. Chapitre.3. Conception :

Tableau [3.1] : Description du Diagramme de cas d'utilisation Globale.....44

Tableau [3.2] : Description du Diagramme de cas d'utilisation Controller Le Robot Réel....45

Tableau [3.3] : Description du Diagramme de cas d'utilisation Controller Le Simulateur....46

III. Chapitre.4. Implémentation :

Tableau [4.1] : Temps d'exécution pour chaque Mode.....62

Introduction générale

Introduction générale

La robotique touche aujourd'hui de nombreux secteurs de la vie. Elle a extrêmement progressé durant le siècle dernier, notamment dans l'industrie.

La robotique comme ensemble d'études, de techniques de conception, de création et de mise en œuvre des robots.

Cependant, les robots manipulateurs existants souffrent encore de faiblesses qui les empêchent de mener à bien certaines tâches et limitent leurs champs d'action.

Ces robots sont en effet rigides, lourds et encombrants ce qui se traduit par une grande énergie consommée et une vitesse d'exécution lente. La consommation peut constituer un point crucial quand l'énergie est limitée, comme c'est le cas dans les applications spatiales.

D'autre part, La simulation permet entre autres mettre en offrant la possibilité de tester les programmes robots virtuellement avant qu'il ne soit trop tard. La simulation vient donc influencer la modélisation de la cellule. Le simulateur rend possible la modélisation et la planification de la cellule robotique complète. Il peut comporter un contrôleur virtuel, un boîtier de commande virtuelle (intéressant pour la formation du personnel), un éditeur de programme en ligne, un simulateur d'entrées/sorties et un simulateur de mécanismes. Il peut également offrir la possibilité d'automatiser certaines tâches et de développer de la programmation robot paramétrique en fonction de la géométrie des pièces.

La présente étude a pour objectif de contrôler un simulateur graphique 3D développé par gazebo. Cette application sera utilisée pour simuler, suivre le comportement du robot « RobuTer/ULM », qui est un manipulateur mobile du centre de développement des technologies avancées (CDTA).

Le centre de développement des technologies avancées (CDTA) est un établissement public à caractère scientifique et technologique créé en 1982 (au sein du commissariat aux énergies nouvelles).

Ce mémoire peut se décomposer en quatre chapitres qui peuvent être organisés de la façon suivante:

- ✓ **Le premier chapitre** porte sur les concepts de la robotique, les composants ainsi que les types des robots qui existent dans la littérature et les domaines d'utilisation des robots.

Introduction générale

- ✓ **Le deuxième chapitre** est consacré à une étude approfondie sur la simulation et les différents simulateurs en se basant sur gazebo nous terminons avec un aperçu sur la téléopération.
- ✓ **Le troisième chapitre** présente la conception de notre solution en mettant en évidence son architecture globale et les différentes communications existantes.
- ✓ **Le quatrième chapitre** porte sur l'implémentation et la réalisation de notre travail. Nous y présentons le simulateur et le protocole de communication qui existe, ainsi que les différents outils de développement qui nous avons utilisé.

1. Introduction :

La robotique est la science qui s'intéresse aux robots. En fait, il s'agit d'un domaine multidisciplinaire on y trouve des aspects concernant la mécanique, l'informatique, l'électronique. On va présenter dans ce chapitre une vue générale sur les robots tels que : la définition de la robotique, l'historique, les catégories de robot, les domaines d'utilisation des robots, ainsi que les composants d'un robot. Nous allons nous intéresser particulièrement à l'étude des composantes des robots manipulateurs mobiles.

2. Robotique Générale :

Dans cette partie, nous allons présenter la définition de robotique et ensuite la définition et l'historique du robot :

2.1. Définition de Robotique :

La robotique est l'ensemble des études et des techniques de conception et d'œuvres des robots, peuvent être définis comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes. [1]

2.2. Définition de Robot :

Un robot est une machine équipée de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception assurée par leurs capteurs. [1]

2.3. Historique :

L'origine du mot robot provient de la langue tchèque dans laquelle son ancêtre "**robota**" signifie **travail forcé**. Au cours de l'histoire on peut distinguer trois types de robots correspondant en quelques sortes à l'évolution de cette "espèce" créée par l'Homme.

Le **premier type** de machine que l'on peut appeler robot correspond aux "**Automates**".

Le **second type** de robot correspond à ceux qui sont équipés de capteurs (en fait les sens du robot). On trouve des capteurs de température, photo électronique, à ultrasons pour par exemple éviter les obstacles ou suivre une trajectoire.

Enfin le **dernier type** de robot existant correspond à ceux disposant d'une intelligence dite "artificielle" et reposant sur des modèles mathématiques complexes tels que les réseaux de neurones. En plus de capteurs physiques comme leurs prédécesseurs. Ces robots peuvent prendre des décisions beaucoup plus complexes et s'appuient également sur un apprentissage de leurs erreurs comme peut le faire l'être humain. [2]

3. Les Composants d'un Robot :

Un robot, en tant que système, se compose d'éléments, qui sont intégrés pour former un ensemble. La plupart des robots contiennent les éléments suivants:

3.1. Actionneurs: Ce sont des mécanismes qui permettent à /aux effecteur(s) d'exécuter une action, de convertir les commandes logicielles (Software) en mouvements physiques, leur but primaire est de produire assez de force pour provoquer le mouvement du robot, celle-ci représente la transformation d'une énergie source en énergie mécanique. La technologie des actionneurs est étroitement liée à l'énergie de base utilisée (pneumatique, hydraulique, électrique). [3]

3.2. Capteurs: Ce sont des outils de perception qui permettent de gérer les relations entre le robot et son environnement. Il existe deux types de capteurs tels que : les capteurs proprioceptifs qui mesurent l'état mécanique interne du robot (comme les capteurs de position, de vitesse ou d'accélération), et les capteurs extéroceptifs qui recueillent des informations sur l'environnement (comme la détection de présence, mesure de distance... etc.). Les capteurs ont comme fonction de lire les variables relatives au mouvement du robot pour permettre un contrôle convenable. [3]

3.3. Effecteurs : Ce sont tous les mécanismes à travers lesquels le robot peut effectuer des changements propres à lui, ou relatifs à l'environnement, ces changements se font grâce aux actionneurs. [3]

3.4. Contrôleur : Le contrôleur reçoit les données de l'ordinateur (le cerveau du système), commande les mouvements des actionneurs, et coordonne les mouvements avec les informations reçues par les capteurs. [4]

3.5. Processeur: le processeur est le cerveau du robot. Il calcule les mouvements des différentes parties du robot, la vitesse de chaque moteur et supervise les actions coordonnées du contrôleur et les capteurs. Dans certains systèmes, le contrôleur et le processeur sont intégrés ensemble en une seule unité, et dans d'autres cas, ce sont des unités séparées. [4]

3.6. Logiciel: trois groupes de logiciels sont utilisés dans un robot. L'un est le système d'exploitation qui exploite le processeur. Le second est le logiciel robotique qui calcule le mouvement nécessaire de chaque moteur du robot. Le troisième groupe est la collection d'application - orientée les routines et les programmes développés pour utiliser le robot ou ses périphériques pour des tâches spécifiques telles que l'assemblage, le chargement de machines, la manutention et les routines de vision. [4]

4. Les Types De Robots :

Il existe plusieurs types de robots. On va citer quelques types, nous allons nous intéresser particulièrement aux robots mobiles et manipulateurs aussi et aux manipulateurs mobiles avec leurs types.

4.1. Robots Mobiles :

Un robot mobile (**figure [1.1]**) est une machine automatique capable de se déplacer dans un environnement donné. Ils sont équipés ou non de manipulateurs suivant leur utilisation. Il existe différents types de ce robot tel que les robots explorateurs, les robots de service et les robots ludiques. [5]



Figure [1.1] : Robot Mobile. [6]

4.1.1. L'architecture des Robots Mobiles : L'architecture des robots mobiles se structure en quatre éléments :

- La structure mécanique et motrice.
- Les organes de sécurité.
- Le système de traitement des informations et gestion des tâches.
- Le système de localisation.

4.1.2. Les Robots à Roues :

Il existe plusieurs classes de robots à roues déterminées, principalement, par la position et le nombre de roues utilisées.

Nous citerons ici quatre classes principales de robots à roues :

a. RobotUnicycle :

Un robot mobile de type unicycle(**figure [1.2]**), Il est composé de deux roues motrices indépendantes au milieu du corps principal du robot. Il existe d'autres robots qui peuvent être classés dans la catégorie des robots mobiles de type unicycle, notamment le robot mobile avec deux roues motrices indépendantes à l'arrière du corps principal du robot et une roue non commandée à l'avant ou à l'arrière (roue folle) .Cette dernière est destinée uniquement à assurer la stabilité du robot. [7]

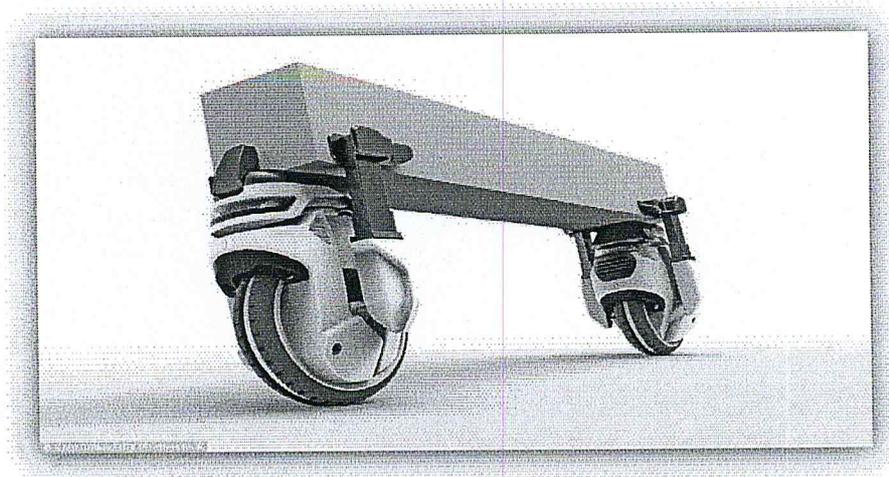


Figure [1.2] : Robot de type unicycle.[8]

b. Robot Tricycle :

Un robot mobile de type tricycle (**figure [1.3]**) est constitué de deux roues fixes placées sur un même axe et d'une roue centrée orientable placée sur l'axe longitudinal. Le mouvement du robot est donné par la vitesse des deux roues fixes et par l'orientation de la roue orientable. Son centre de rotation est situé à l'intersection de l'axe contenant les roues fixes et de l'axe de la roue orientable. C'est un robot non-holonyme. En effet, il est impossible de le déplacer dans une direction perpendiculaire aux roues fixes. Sa commande est plus compliquée. Il est en général impossible d'effectuer des rotations simples à cause d'un rayon de braquage limité de la roue orientable. [9]

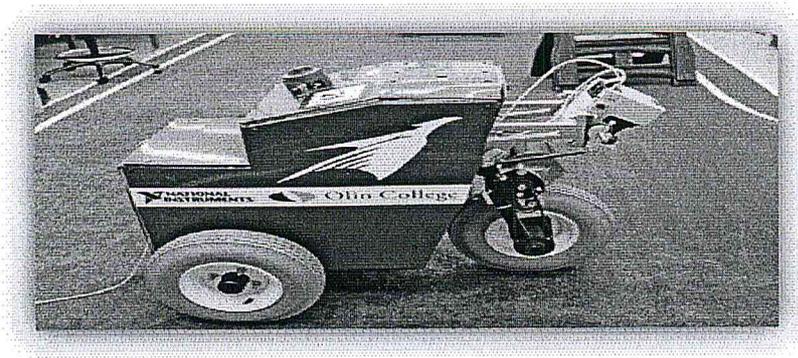


Figure [1.3] : Robot de type tricycle. [10]

c. Robot de Type Voiture :

Un robot de type voiture (**figure [1.4]**) est semblable au tricycle, il est constitué de deux roues fixes placées sur un même axe et de deux roues centrées orientables placées elles aussi sur un même axe.

Le robot de type voiture est cependant plus stable puisqu'il possède un point d'appui supplémentaire.

Toutes les autres propriétés du robot voiture sont identiques au robot tricycle, le deuxième pouvant être ramené au premier en remplaçant les deux roues avant par une seule placée au centre de l'axe, et ceci de manière à laisser le centre de rotation inchangé. [9]

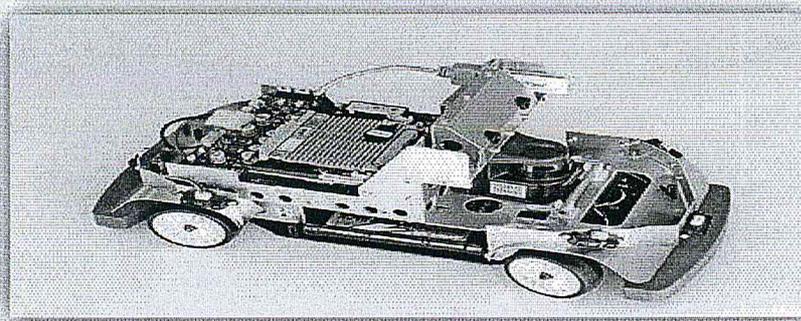


Figure [1.4] : Robot de type voiture. [11]

d. Robot Omnidirectionnel :

Un robot mobile est dit omnidirectionnel (**figure [1.5]**) si l'on peut agir indépendamment sur les vitesses : vitesse de translation selon les axes $\sim x$ et $\sim y$ et vitesse de rotation autour de $\sim z$. D'un point de vue cinématique on montre que cela n'est pas possible avec des roues fixes ou des roues centrées orientables. On peut en revanche réaliser un robot omnidirectionnel en ayant recours à un ensemble de trois roues décentrées orientables ou de trois roues suédoises disposées aux sommets d'un triangle équilatéral. Du point de vue de la transmission du mouvement, ceci ne va pas sans poser de problème. [12]

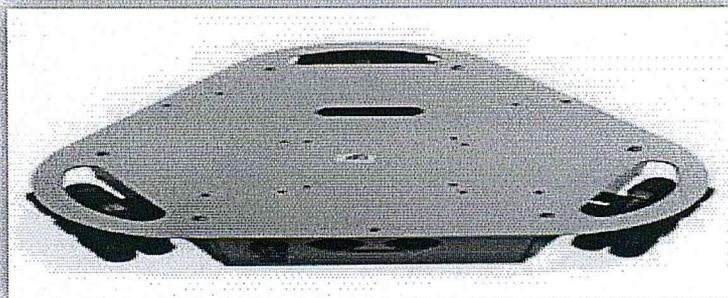


Figure [1.5] : Robot de type omnidirectionnel. [13]

4.2. Robots Manipulateurs :

Robots manipulateur (**figure [1.6]**) existe dans les applications industrielles .Ce robot est une série de segment relativement rigides dont les articulations sont commandées par des actionneurs et chaque segment équipé d'un effecteur (pince, main articulée, outil, ...) servant à

la réalisation de la tâche à accomplir, il existe différents types de ce robot tel que robots industriels et robots pour l'assistance médicale. [14]

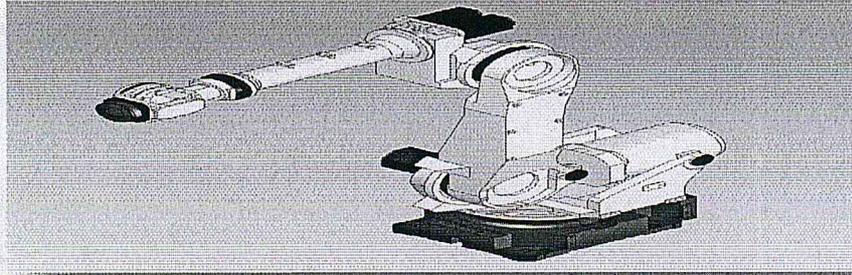


Figure [1.6] : Un bras manipulateur. [15]

4.2.1. Manipulateur: C'est le corps principal du robot qui comprend les jonctions, les articulations, et d'autres éléments de structure du robot. Il convient de noter ici que le manipulateur seul n'est pas un robot. [4]

4.2.2. Les Type De Robots Manipulateurs :

Les robots manipulateurs se présentent sous plusieurs formes. Qui se répartissent en en cinq grandes catégories :

► Robots Cylindriques :

- La géométrie cylindrique diffère de la géométrie cartésienne en ce que la 1^{ère} articulation prismatique est remplacée par une articulation rotoïde .
- La structure cylindrique offre une très bonne rigidité mécanique .
- L'articulation prismatique horizontale permet à l'organe terminal d'accéder à des cavités horizontales
- Utilisation typique: transport d'objets, même de grande taille (dans un tel cas, des moteurs hydrauliques sont préférés aux moteurs électriques). [16]

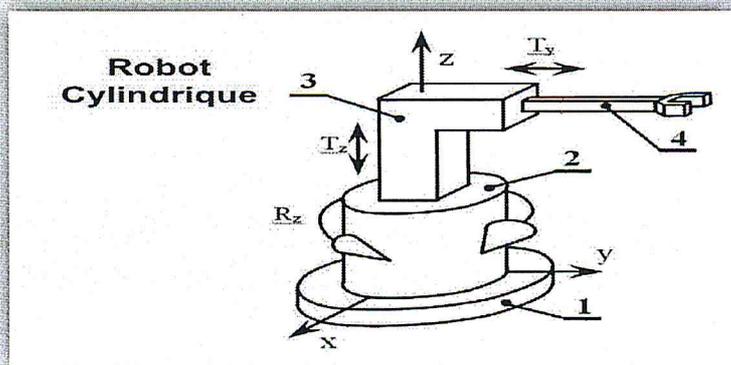


Figure [1.7] : L'espace de travail de robot cylindrique. [17]

► Robots Rectilignes :

Les robots rectilignes (figure [1.8]) a trois axes de mouvement (x, y, z). Pour cette raison, le robot rectiligne est parfois appelé Robot cartésien. Ces robots sont exploités par vérin pneumatique. [18]

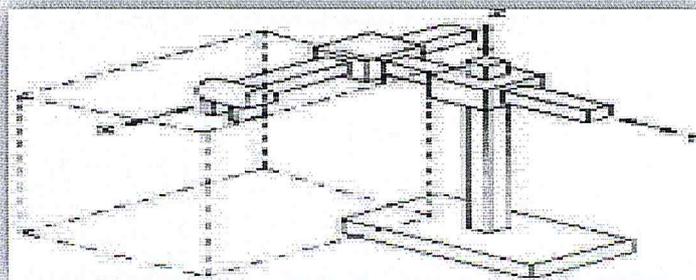


Figure [1.8] L'espace de travail de robot rectiligne. [19]

► Robots Sphériques :

- La géométrie sphérique diffère de la géométrie cylindrique en ce que la deuxième articulation prismatique est remplacée par une articulation rotoïde.
- La rigidité mécanique est inférieure à celle des deux robots précédents et la construction mécanique est plus complexe.

- Le volume de travail peut comprendre la base de support du robot, ce qui permet la manipulation d'objets sur le plancher.
- Utilisation typique: usinage. [16]

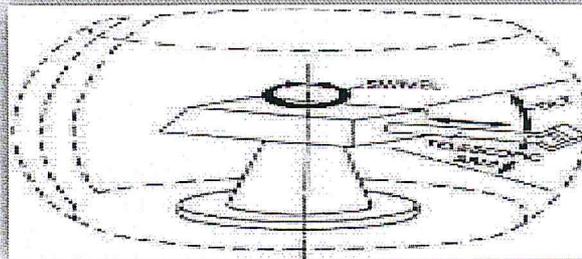


Figure [1.9] : L'espace de travail de robot sphérique. [20]

► Robots Articulés :

Le bras articulé (**figure [1.10]**) du robot ressemble à un bras humain. Il se compose de deux éléments, nommés l'avant-bras et le bras supérieur. Ce type de robot n'a généralement pas besoin d'un lieu séparé. [21]

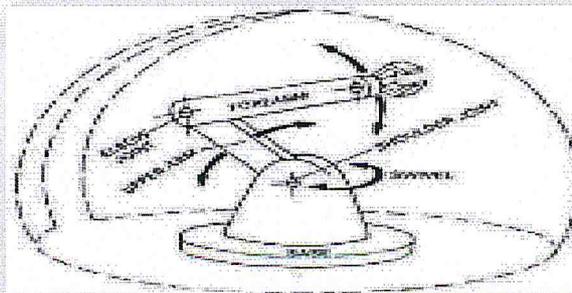


Figure [1.10] :L'espace de travail de robot articulé. [22]

► Robots SCARA :

- Selective Compliance Assembly Robot Arm
- Manipulateur sphérique à géométrie “spéciale”

- Deux articulations roto ides et une articulation prismatique, tous les axes sont parallèles.
- Rigidité élevée pour charges verticales et souplesse aux charges horizontales.
- Bien adapté à des tâches de montage vertical et à la manipulation de petits objets.
- Précis et très rapide (1er modèle: 1981). [23]

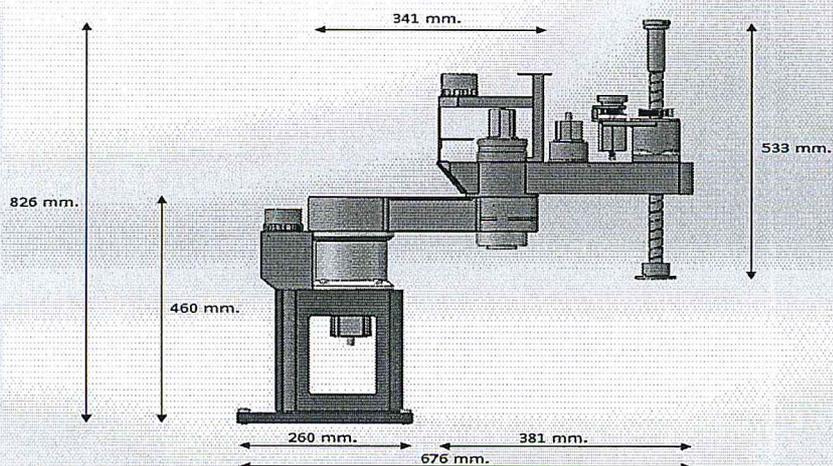


Figure [1.11] : Robot SCARA 4 Axes. [24]

4.3. Robots Manipulateurs Mobile :

La manipulation mobile diffère des domaines de recherche traditionnels en robotique, tels que la perception, la planification de mouvement, la manipulation, le contrôle, la cognition et l'intelligence artificielle. Pour comprendre ce que la manipulation mobile est exactement, nous devrions commencer par considérer l'histoire de ce domaine de recherche. Le terme manipulation mobile lui-même pourrait être trompeur. Il a été inventé dans les années quatre-vingt-dix, lorsque les laboratoires de recherche ont commencé à monter des robots manipulateurs sur des plates-formes mobiles. Une recherche menée sur une plate-forme expérimentale qui combine les capacités de mobilité et de manipulation. [25]

Un robot manipulateur mobile est constitué de composants matérielle tel que la plate-forme mobile et logicielle tel que le modules de vision, de localisation, de navigation. [26]

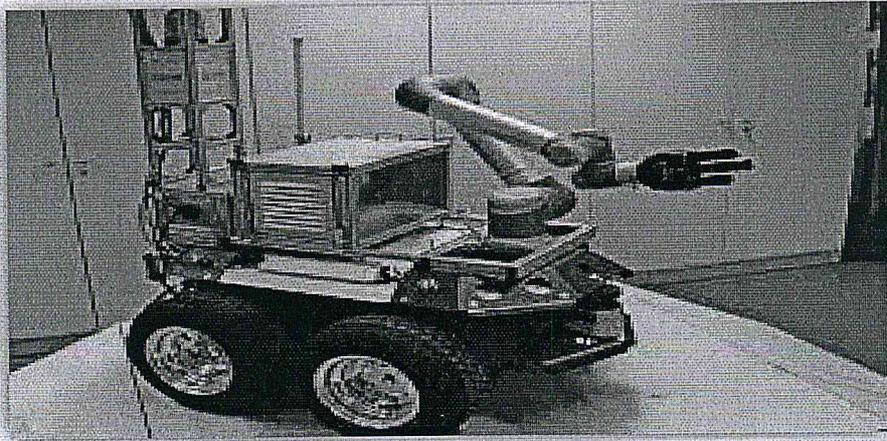


Figure [1.12] : Robot Manipulateur Mobile. [27]

4.3.1. Les Manipulateurs Mobiles Qui existent dans la Littérature :

Dans la littérature il existe différents modèles de robots manipulateurs mobiles, comme le Manipulateur Mobile MM-400, le MM-500, RobuTER/ULM ...etc. Nous présenterons dans cette partie quelques modèles :

➤ Manipulateur Mobile MM-400 :

Le manipulateur mobile MM-400 (**figure [1.13]**), combine le robot mobile MP-400 et un bras robot léger de Schunk pour créer un système de robot très compact et agile pour les applications de recherche.

Sa petite empreinte rend le MM-400 parfait pour être utilisé dans des environnements étroits où d'autres robots ne peuvent pas naviguer.

Le bras léger Schunk peut être personnalisé individuellement pour répondre parfaitement aux exigences de l'application.

Le logiciel de contrôle de la plate-forme comprend la localisation, la planification et l'interpolation de chemin et est basé sur le logiciel de test de NeobotixPlatformCtrl ou ROS.

Le bras Schunk peut également être contrôlé par ROS. [28]

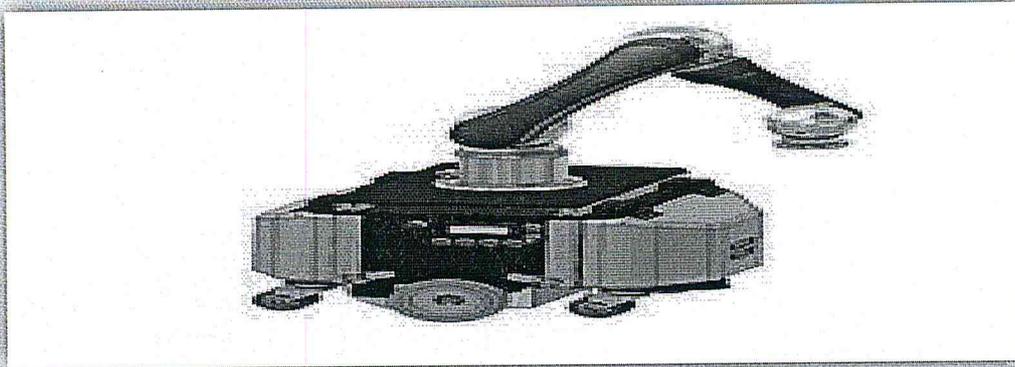


Figure [1.13] : Robot Manipulateur mobile MM-400. [28]

➤ **Manipulateur mobile MM-500 :**

Le manipulateur mobile MM-500 (**figure [1.14]**) est basé sur le MP-500 mais possède un bras robot léger de Schunk intégré à la plate-forme de base. Ce système de robot extrêmement polyvalent est prêt pour une utilisation industrielle complète.

Le MM-500 est particulièrement utile pour l'automatisation des applications de transport industriel et pour la manipulation de pièces plus petites. Les applications possibles sont l'alimentation entièrement autonome des machines, la cueillette des pièces du stockage et le transport général des pièces avec chargement autonome.

Ce robot est également une plate-forme très rentable pour la recherche en robotique de service et en manipulation mobile. Le bras léger Schunk peut être personnalisé individuellement pour répondre parfaitement aux exigences de l'application.

Le logiciel de contrôle de la plate-forme comprend la localisation, la planification et l'interpolation de chemin et est basé sur le logiciel de test de NeobotixPlatformCtrl ou ROS.

Le bras Schunk peut également être contrôlé par ROS. [28]



Figure [1.14] : Robot Manipulateur mobile MM-500. [28]

➤ Manipulateur mobile MM-700 :

Le manipulateur mobile MM-700 (figure [1.15]) intègre un bras robot léger d'Universal Robots ou Schunk au robot mobile MP-700.

Grâce à la très grande charge utile et à la robustesse du MP-700, d'autres bras robot, comme Baxter de RethinkRobotics ou KUKA LBR iiwa, peuvent également être facilement intégrés.

Le mouvement de la plate-forme peut être contrôlé soit par PlatformCtrl, soit par le système d'exploitation Robot gratuit. La plupart des bras de robot peuvent être contrôlés avec le logiciel de contrôle d'origine et leur unité de commande manuelle. Le contrôle par l'ordinateur de bord est également possible. Lorsqu'un bras Schunk est utilisé, il peut être contrôlé par ROS. [28]

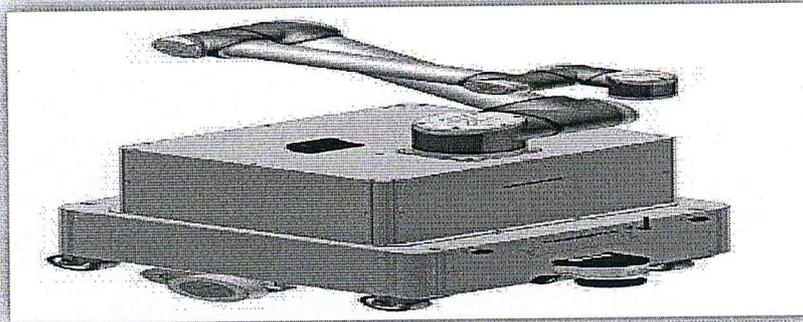


Figure [1.15] : Robot Manipulateur mobile MM-700. [28]

➤ Robot Robuter/ULM :

ROBUTER/ULM est un robot est constitué de deux sous-systèmes robotiques, un bras manipulateur ultraléger « ULM » à 6 liaisons rotoïdes porté sur une plateforme différentielle mobile « Robuter ».

La plateforme mobile se déplace avec deux roues motrices, les deux roues folles assurent la stabilité du système mécanique. Cette plateforme porte une ceinture de 24 capteurs ultra-sons. Elle contient aussi un capteur Laser de type LMS posé à l'avant du mobile. Le déplacement du mobile est mesuré par des capteurs odométriques emboîtés dans les roues motrices.

L'ULM est un manipulateur à 6 liaisons rotoïdes avec une pince dotée d'un capteur d'effort sur son extrémité. Chaque articulation contient un capteur de position. La pince contient un capteur d'effort intégré. Une caméra CCD est portée sur cette extrémité.

Le contrôle du robot se fait avec le système d'exploitation Linux 6.1 installé sur un PC industriel embarqué, qui contrôle quatre interfaces électroniques (cartes de commandes) à base de microcontrôleur MPC555. La communication est faite via un bus CAN¹. Le système d'exploitation est doté d'une couche software RTAI² 1.3. [29]

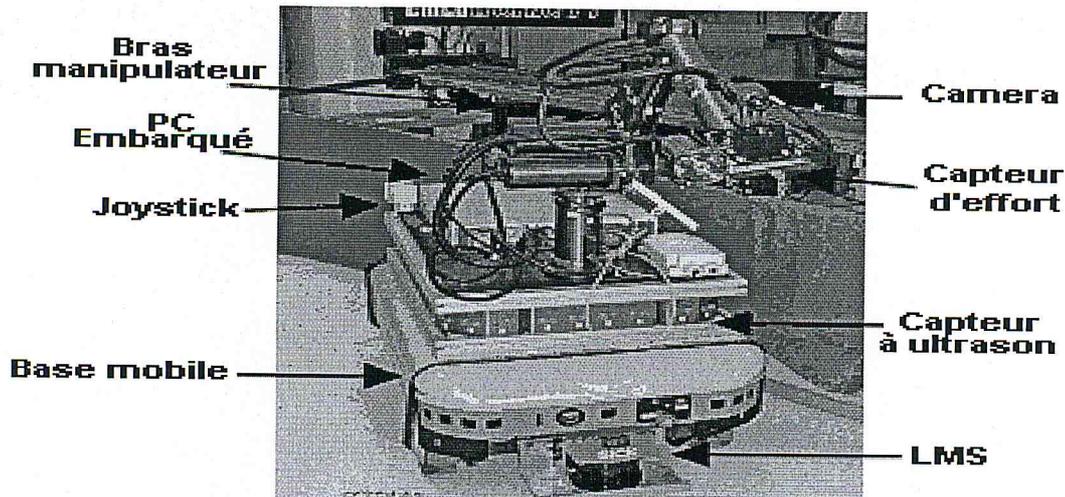


Figure [1.16] : Présentation du manipulateur mobile Robuter/ULM. [30]

4.3.2. Caractéristique Principales des Manipulateurs Mobiles :

Un certain nombre de caractéristique relatives aux manipulateurs mobiles ont peut être constatées lors de cette recherche bibliographique.

4.3.2.1. Redondance Mécanique :

Un robot est redondant lorsque le nombre de degrés de liberté de l'effecteur est supérieur au nombre de degrés de liberté de l'espace articulaire (nombre d'articulation motorisées). cette propriété permet d'augmenter le volume du domaine accessible du robot et de préserver les capacités de déplacement de l'effecteur en présence d'obstacle.

Les manipulateurs mobiles sont, la plupart du temps, des systèmes mécaniquement redondants dans le sens ou le nombre de degrés de liberté et total. [31]

4.3.2.2. Capteurs :

Les types de capteurs utilisés et leur position représentent un critère important dans le choix d'un manipulateur mobile. Cela dépendra du domaine d'application. [32]

¹ : Control Area Network.

² : Real Time Application Interface.

4.3.2.3. Mode de Manipulation :

Dans les définitions classiques des manipulateurs mobiles, la locomotion est assurée par un véhicule et la manipulation se fait grâce à un ou plusieurs manipulateurs. Cependant, il existe un cas particulier où ce sont les roues qui effectuent la tâche de manipulation cas du mobipulator. [33]

4.3.2.4. Type de Plateforme Mobile :

Pour faire en sorte que le robot se déplace correctement selon la consigne, la position et le type des roues doivent être connus. Cela définira également le type de plateforme mobile étudié, et permettra de définir correctement le modèle devant être utilisé. [32]

4.3.3. Domaine d'utilisation des Manipulateurs Mobiles :

On peut classer les différents manipulateurs mobiles selon les domaines d'application comme suit :

4.3.3.1. Domaine Spatial :

4.3.3.1.1. Mars Exploration Rover (MER) :

C'est une mission de la NASA composée de deux robots mobiles ayant pour objectif d'étudier la géologie de la planète Mars et en particulier, le rôle joué par l'eau dans l'histoire de la planète. Ces deux robots, MER-A renommé *Spirit* et MER-B renommé *Opportunity* ont été lancés au début de l'été 2003 et se sont posés en janvier 2004 sur deux sites martiens susceptibles d'avoir conservé des traces de l'action de l'eau dans leur sol. Chaque rover ou automobile, piloté par un opérateur depuis la Terre, a alors entamé une tournée en utilisant une batterie d'instruments embarqués pour analyser les roches les plus intéressantes.

Chaque rover se déplace sur 6 roues mues par l'énergie électrique fournie par des panneaux solaires.

Il est équipé de 3 paires de caméras utilisées pour la navigation et de plusieurs instruments scientifiques. [32]

4.3.3.1.2. SkyWorker :

Initié par NASA, consiste à créer une équipe de manipulateurs mobiles capables de se déplacer sur des panneaux solaires et d'y réaliser des tâches d'assemblage, d'inspection et de maintenance dépassant les capacités des astronautes. [34]

4.3.3.1.3. Domaine Manufacturier Les Manipulateurs Mobiles :

Le robot Kamro d'Institute for Real-time Computer Systems And Robotics de l'université de Karlsruhe en Allemagne, est destiné à des tâches autonomes en environnement industriel, il est constitué d'une plateforme mobile omnidirectionnelle lui permettant une grande facilité de mouvement, dotée de capteurs à ultrason, et de deux bras PUMA 200 équipés de capteurs d'effort six axes, ainsi que deux caméras se trouvant sur chacun des organes terminaux. [3]

4.3.3.1.4. Domaine Médical :

La robotique se développe et se perfectionne au grand bénéfice de la médecine. Un robot médicale (**figure [1.16]**) est un robot qui peut soit être piloter par un médecin et cela en vue d'une meilleure précision ou alors exécuter des opérations délicate, qui auront ainsi plus de chance de réussir.

De plus les robots ont la capacité d'intervenir sur des cellules cible ce qui a pour effet de limiter les effets secondaires après une opération ou après l'injection de médicament, en générale cet appareil aide à soigner des patients. [35]

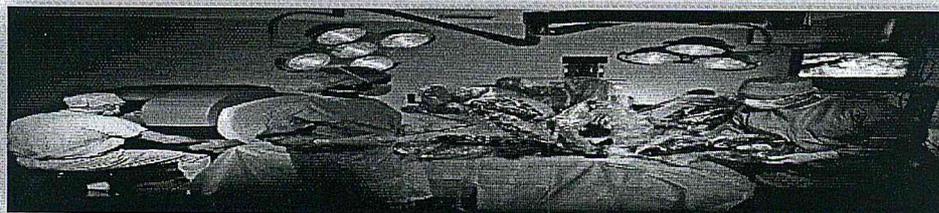


Figure [1.17] : Robot au bloc opératoire. [36]

4.3.3.1.5. Robotique de service :

Les robots de service (supports techniques) permettent aux personnes âgées et handicapées de vivre d'une façon autonome et soutenue dans leurs lieux de résidence privée (maisons, etc.) [37]

- **Robots domestiques :**

Robots utilisés à la maison (**figure [1.17]**). Ce type de robots comprend de nombreux appareils très différents, tels que les aspirateurs robotiques, robots nettoyeurs de piscines, balayeuses, nettoyeurs gouttières et autres robots qui peuvent faire différentes tâches. En

outre, certains robots de surveillance et de télé présence pouvaient être considérés comme des robots ménagers. [38]

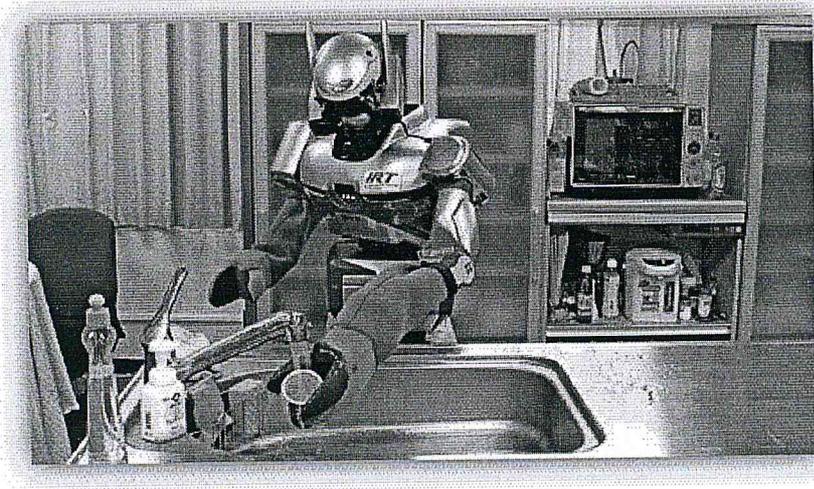


Figure [1.18] : RobotDomestique. [39]

- **Ouverture de Porte :**

Le robot lias (Leuven intelligent autonomes système, comporte une base mobile nom holonome de type RobuTER équipée de nombreux capteurs (un laser rotatif, etc.).La manipulation est accomplie grâce au manipulateur industriel CRSA465 à six liaisons roto ides, comprenant un capteur d'effort, un pince, une caméra assurant la stéréo vision .ce robot a servi de base expérimentale pour effectuer un mouvement spécifique d'ouverture d'une porte par approche réactive. [40]

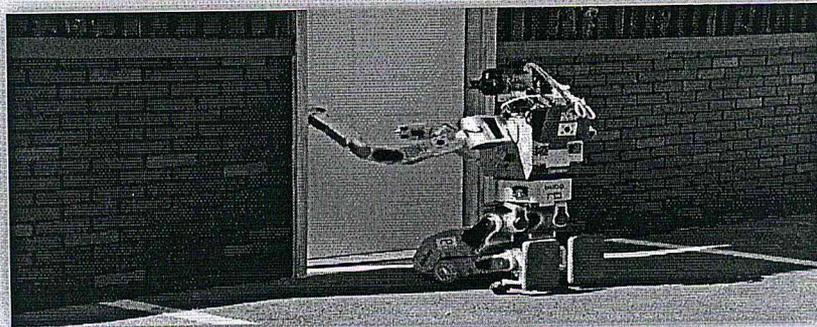


Figure [1.19] : RobotOuverture de Porte. [41]

- **Saisie d'Objet :**

Le robot El_E apporte les objets désigné à l'aide d'un pointeur laser pour aider les personnes handicapées ou immobilisées. Ses caméra lui permettent d'agir dans un environnement domestique, de repérer des personne et de saisir un objet se trouvent sur le sol, sur une table ou sur une étagère.[42]

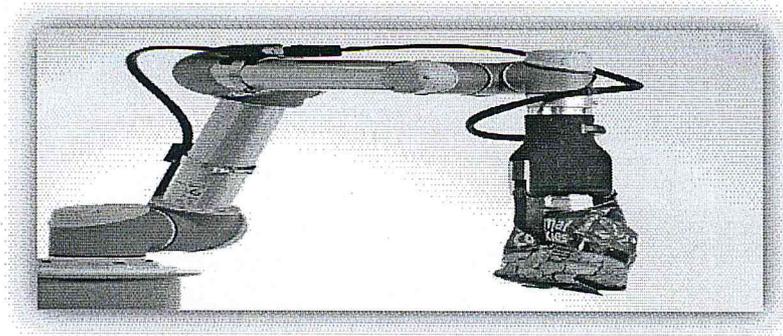


Figure [1.20] : Robot Saisie d'Objet. [43]

5. Conclusion :

La robotique est un domaine récent, qui comprend de nombreuses applications dans divers disciplines et domaines importants de l'informatique (militaire, médical, spatial). Dans ce chapitre, nous avons présenté la robotique en général et plus particulièrement les robots manipulateurs mobiles leurs caractéristiques principale et leur domaine d'utilisation ...) .le prochaine chapitre sera consacré à la simulation et Téléopération.

Chapitre 2 :

La simulation

1. Introduction :

L'environnement de la simulation de robotique est conçu pour faire face aux défis en matière de formation des astronautes et du personnel au sol. Il est en outre un outil essentiel pour le soutien des opérations spatiales. Le simulateur est utilisé actuellement pour entraîner les astronautes de la Station spatiale internationale à l'exécution des activités de robotique en orbite. La simulation de robotique avant son déploiement réel représente une alternative moins coûteuse et plus rapide que son expérimentation en environnement réel. On va présenter dans ce chapitre la définition de simulation et simulateurs, on parlera aussi du simulateur gazebo, ensuite on terminera par la téléopération.

2. La Simulation :

Ces dernières années, les chercheurs en robotique ont commencé à s'intéresser de plus en plus à la simulation en temps réel des robots car elle apporte beaucoup d'avantages :

- a. **Un gain en temps et en main d'œuvre :** La multiplication des ressources matérielles et logicielles liées aux robots, ainsi que les tâches qu'ils ont à réaliser, rend les tests sur les robots de plus en plus lents et coûteux, la simulation apporte une solution rapide et efficace à ce problème même si elle ne dispense pas des tests réels.
- b. **Situation réelle et répétée :** grâce à la simulation, les chercheurs peuvent s'approcher des situations réelles difficilement réalisables en laboratoire, tout en les répétant à l'identique autant de fois que cela s'avère nécessaire.
- c. **Test matériels :** la simulation permet aussi de tester du matériel non disponible dans le laboratoire (par exemple l'interaction entre plusieurs robots alors que le laboratoire n'en possède qu'un seul)

En robotique, on peut distinguer deux types de simulateurs, ceux restreints à la validation d'un composant particulier, et les autres qui comprennent la simulation de tout l'environnement pour tester les performances des robots. Il n'en n'existe pas beaucoup, nous allons nous intéresser à l'un d'entre cette deuxième catégorie, qui est eux le simulateur Gazebo. [44]

2.1. Simulation Haptique Interactive:

Le Dispositif haptique est un robot qui, au lieu de manipuler le monde physique conformément à une commande utilisateur, applique des forces à la main ou au doigt de l'utilisateur en fonction des forces résultant du mouvement de l'utilisateur dans un environnement virtuel. Comme si le virtuel est soumis à des forces.

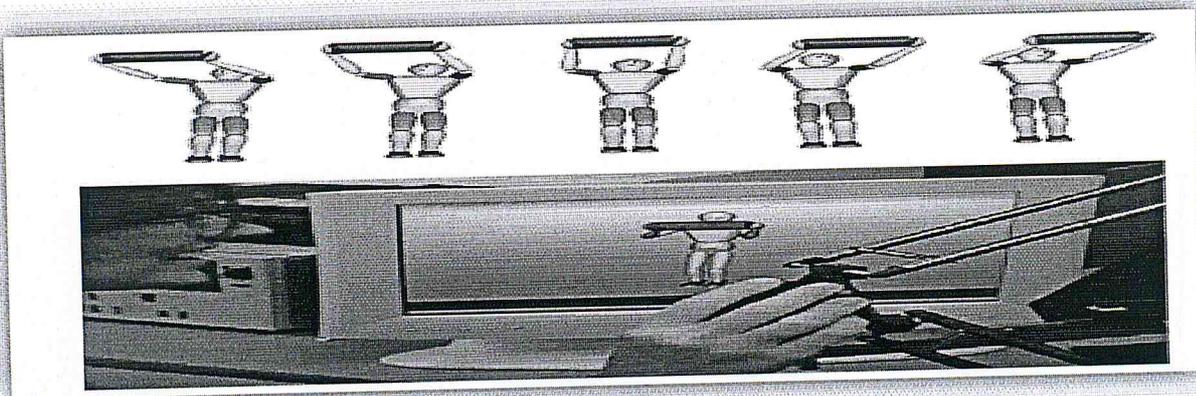


Figure [2.1] : Un utilisateur commande un robot virtuel pour déplacer l'objet manipulé en utilisant un dispositif haptique. Le dispositif permet à l'utilisateur d'interagir et de contrôler la tâche de manipulation. Alors que le robot humanoïde ajuste de façon autonome ses énergies de posture. [45]

Un dispositif haptique peut également servir de dispositif d'entrée, comme représenté sur la (figure [2.1]), l'utilisateur spécifie un mouvement pour l'objet détenu par les robots humanoïdes. La séquence d'images de la (figure [2.1]) montre le résultat sur tout le mouvement de la figure si l'objet est commandé de se déplacer d'un côté à l'autre. Alors que les objets sont manipulés de manière interactive, la figure ajuste de façon autonome sa posture en fonction d'énergies de posture simples. Le dispositif haptique est utilisé pour générer les forces du ressort virtuel qui apparaissent à l'utilisateur comme les forces de contraintes provoquées par le contact avec un environnement réel. Ce cadre permet également à l'ombrage haptique de lisser les bords dans des données polygonales et d'afficher des propriétés de surface, telles que le frottement et la texture. [45]

2.2. Simulation Dynamique :

La relation dynamique entre tous les points de contact existants peut être décrite. Cette relation est caractérisée par les masses perçues aux points de contact. Une force exercée au niveau d'un point de contact, qu'il s'agisse d'une collision avec un autre objet ou de l'interaction avec un utilisateur, peut être traduite en forces à tous les points de contact apparentés. Les calculs nécessaires peuvent être effectués avec un algorithme récursif efficace.

La représentation de l'espace de contact permet de décrire facilement l'interaction entre les groupes de systèmes dynamiques sans avoir à examiner les équations complexes du mouvement du système individuel de la Terre. En tant que tel, le modèle de collision peut être développé avec la même facilité que si on envisageait l'interaction seulement entre les corps simples. L'impact et les forces de contact entre les corps en interaction peuvent alors être efficacement résolus. Pour la simulation de la dynamique du corps articulé se déplaçant dans l'espace libre.

La (figure [2.2]) illustre un environnement virtuel. Deux personnages humanoïdes sont en collision et interagissent avec de nombreux objets dans l'environnement. [45]

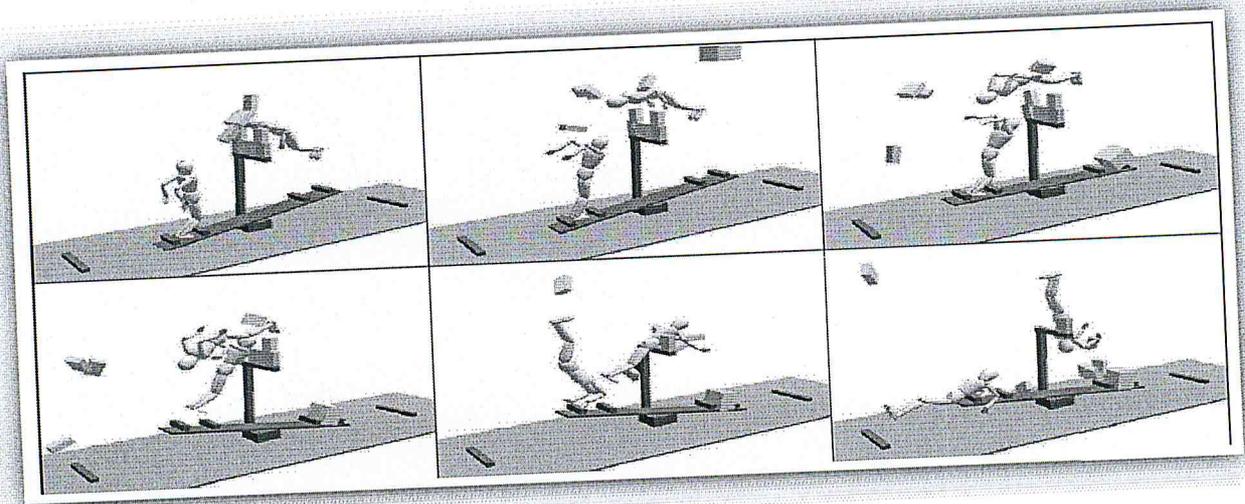


Figure [2.2] : cette séquence d'images montre l'interaction dynamique complexe entre deux figures humanoïdes et un grand nombre d'objets dans la scène. Les nombreux contacts entre les objets dans cette scène et leurs mouvements résultants sont calculés interactivement en fonction de leurs propriétés dynamiques. [45]

3. Le Simulateur :

Dans ce partie nous allons présent une vue générale sur les simulateurs, cette vue est définie comme suit :

3.1. Définition de Simulateur :

Un simulateur est un programme qui met en œuvre un modèle de simulation par événements discrets. La tâche première d'un simulateur est d'assurer que la chronologie des événements soit respectée. A chaque occurrence d'un événement, les actions qui sont associées à celui-ci sont exécutées. [46]

3.2. Les Simulateurs des Robots Existants:

Il existe deux types de simulateurs Les simulateurs comme MRSim, RoboticsLab, V-Rep, et le deuxième type Les simulateurs Open-source comme simbad, Stage et Gazebo Nous présenterons dans cette partie quelques simulateurs.

3.2.1. MRSIM :

Multi-Robot Simulator (MRSIM) est un simulateur 2D pour MatLab, étant une boîte à outils pour Mac OSX, Linux et Windows créée en 2012 au Mobile Robotics Laboratory³ (LMR) d'ISR-UC. Bien que la boîte à outils MRSim soit open source, MatLab est un outil pay-to-use. Le code du contrôleur est écrit en langage MatLab et n'est pas portable sur de véritables plateformes robotiques. [47]

Il permet de simuler des capteurs de distance généralement les sonars et les capteurs laser. Et il comprend déjà un grand nombre de fonctionnalités du monde réel tel que la propagation RF, voix et feu. Néanmoins, il ne simule pas la dynamique des objets ou toute autre caractéristique 3D et a donc de faibles besoins matériels fonctionnant dans tout ce qui peut s'exécuter MatLab. En raison de toutes ses simplifications, il peut simultanément simuler un nombre élevé de robots. [47]

3.2.2. RoboticsLab :

Est un simulateur de robot 3D uniquement sous Windows (Windows 7, Vista et XP). La création du programme de contrôle utilise C, C ++, Python et Java, étant portable sur de vraies plateformes robotiques. Il simule les capteurs de gamme, de caméra et d'odométrie.

En outre, il permet également de simuler des environnements réalistes, de la physique mondiale et de la dynamique des objets, nécessitant au moins un Intel Core2duo avec 1 Go de RAM et une carte graphique standard avec OpenGL lib. [48]

3.2.3. V-Rep :

Est un simulateur robotique 3D pour Windows, Mac OSX et Linux. La programmation peut être réalisée en utilisant C, C ++, Python, Java, Lua, MatLab et Urbi. Le code est uniquement portable en plates-formes robotiques réelles s'il est utilisé avec ROS.

Cependant, ROS a pour inconvénient d'offrir des services limités à V-Rep. Il fournit des simulations pour les caméras et les capteurs de proximité / gamme pour éviter les collisions. Il nécessite une carte graphique standard. Il génère des environnements réalistes, la physique et la dynamique des objets, réduisant ainsi le nombre de robots simulés en temps réel. le V-Rep est gratuit pour les étudiants seulement. [49]

3.2.4. Stage :

Est un simulateur 2D pour Linux, Mac OSX et Windows le plus souvent utilisé comme module de plugin Player. Les contrôleurs peuvent être codés en C, C ++, Python et Java via Player ou ROS et le code écrit est portable sur de véritables plates-formes robotiques. Il simule les capteurs de gamme et d'odométrie.

Il a peu d'exigences matérielles en raison de l'existence de 2D fournissant un environnement de simulation de base qui peut être mis à l'échelle pour modéliser un à une centaines de robots à la fois. Bien qu'il puisse créer des simulations visuelles 3D, les données et la cartographie sont toujours en 2D. [50]

3.2.4. Simbad :

Est un simulateur robotique 3D Java uniquement pour Linux, Mac OSX et Windows, il est utilisé à des fins scientifiques et éducatives. La portabilité du code est limitée en raison de son exclusivité pour le langage Java. Il simule les capteurs de vision, de distance et de contact.

Bien qu'il puisse simuler des environnements 3D, il ne simule pas la physique ou la dynamique des objets. Par conséquent, son objectif principal est d'être un outil pour étudier l'Intelligence Artificielle, l'Apprentissage par Machine et plus généralement les algorithmes d'Intelligence Artificielle (AI), dans le contexte de la Robotique Autonome et des Agents Autonome, c'est-à-dire qu'il n'est pas destiné à la simulation du monde réel.

Cette simplicité lui confère de faibles besoins matériels à exécuter par rapport à d'autres simulateurs 3D, nécessitant uniquement une bibliothèque Java 3D, et lui confère un nombre plus élevé de robots qui peuvent être simulés en temps réel. [51]

3.2.5. Gazebo :

Est un simulateur 3D robotique utilisé uniquement sur Linux. Les programmes peuvent être écrits en C, C ++, Python et Java en utilisant Player ou ROS et le code écrit est portable sur de véritables plates-formes robotiques.

Il supporte des simulations pour l'odométrie, la gamme et les capteurs de caméra. Il est capable de simuler des environnements très réalistes, la physique et les commentaires des capteurs. De plus, il simule également une dynamique très rigide du corps, nécessitant pour cela une bonne carte graphique et processeur CPU (pas beaucoup mieux qu'un matériel standard aujourd'hui) et OpenGL Lib.

Par conséquent, le nombre de robots simulés en temps réel est limité aux performances matérielles. Sur une note parallèle, Gazebo et Stage sont compatibles avec ROS. Par conséquent, un client écrit pour l'un d'entre eux (soit Gazebo ou Stage) peut généralement fonctionner sur l'autre avec peu ou pas de modification.

Alors que Stage fournit une simulation en temps réel pour un nombre élevé de robots à faible fidélité, Gazebo fournit une simulation en temps réel pour un faible nombre de robots, très fiables sur le matériel du PC, mais avec une très grande fidélité. [52]

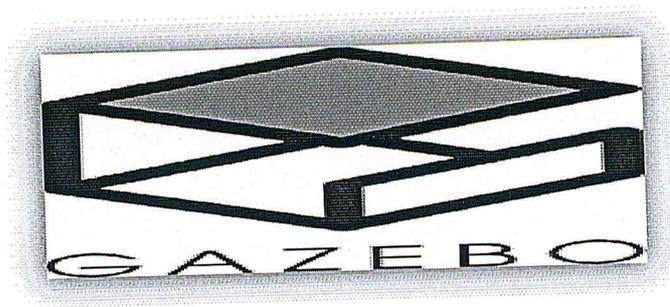


Figure [2.3] : Logo De GAZEBO. [52]

3.3.Comparaison Entre les Simulateurs :

-Cette tableau ce dessus qui représente une comparaison entre les simulateurs commerciaux :

| | MRSim | RoboticsLab | V-Rep |
|---------------------------------|-----------------|---------------------------|---|
| SE | Mac, Linux, Win | Win | Win, Mac, Linux |
| Type de simulateur | 2D | 3D | 3D |
| Langage de Programmation | Matlab | C, C++, Python, Java, | C, C++, Python, Java, Lua, Matlab, Urbi |
| Portabilité | Non | Oui | Avec ROS seulement |
| Capteurs | Gamme | Caméra , gamme , odomètre | Caméra, gamme |
| Exigences | Faible | Norme | Norme |
| Environnement réaliste | Oui | Oui | Oui |

La simulation

| | | | |
|-----------------------------|-----|-----|-----|
| Dynamique des objets | Non | Oui | Oui |
|-----------------------------|-----|-----|-----|

Tableau [2.1] : Comparaison entre les Simulateurs Commerciaux. [53]

-Cette tableau ce dessus qui représente une comparaison entre les simulateurs Open source :

| | Stage | Gazebo | Simbad |
|---------------------------------|-----------------------|------------------------|------------------------|
| SE | Mac, Linux, Win | Linux | Win, Mac, Linux |
| Type de Simulateur | 2D | 3D | 3D |
| Langage de Programmation | C, C++, Python, Java, | C, C++, Python, Java, | Java, |
| Portabilité | Oui | Oui | Limité |
| Capteurs | gamme,odomètre | Caméra, gamme,odomètre | Caméra, gamme, contact |
| Exigences | Faible | Norma | Faible |
| Environnement réaliste | Non | Oui | Non |

| | | | |
|----------------------|-----|-----|-----|
| Dynamique des objets | Non | Oui | Non |
|----------------------|-----|-----|-----|

Tableau [2.2] : Comparaison entre les Simulateurs Open-Source.[53]

4. L'environnement Gazebo :

▪ Description de Gazebo :

Gazebo est un simulateur 3D soutenu par Open Source Robotics Foundation¹ (OSRF). Dans la (figure [2.5]), nous voyons un robot mobile Pioneer avec un viseur laser et une caméra en haut dans un monde Gazebo.

Créé à l'origine pour fonctionner comme over *Player*, il a été adapté pour également travailler sur ROS, éventuellement independant de tout logiciel tiers et autosuffisant. Étant donné que ROS et Gazebo utilisent des sujets pour partager des données dans les processus, un pont de plugin peut être créé, ce qui permet l'utilisation de tous les outils ROS et les piles dans Gazebo. [53]

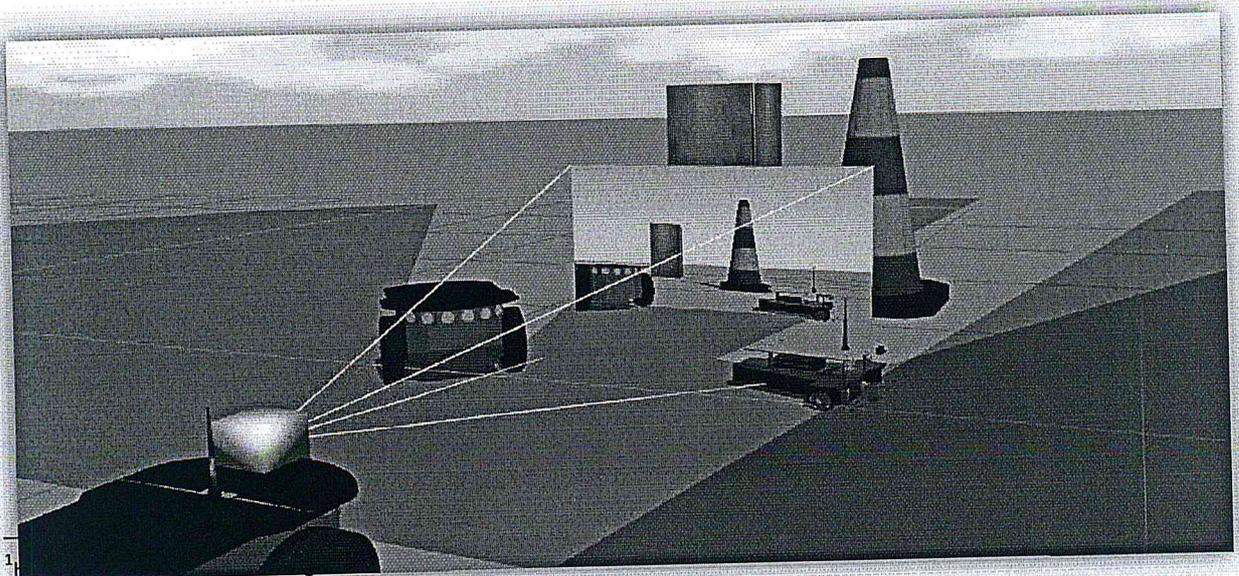


Figure [2.4]: Un robot mobile Pioneer équipé d'une caméra et d'un viseur laser. [53]

L'interface graphique de gazebo : L'éditeur est composé des 2 parties :

- **La palette** à gauche a deux onglets. L'onglet Insertion vous permet d'insérer des pièces (liens et autres modèles) dans la scène pour créer un modèle. L'onglet Modèle affiche une liste de toutes les pièces qui composent le modèle que vous construisez.
- **La vue 3D** sur la droite vous permet d'avoir un aperçu de votre modèle et interagir avec lui pour modifier ses propriétés et créer des liens entre les liens.

Les outils GUI de la barre d'outils supérieure peuvent être utilisés pour manipuler les articulations et les liens dans la vue 3D. [54]

5. La Téléopération :

5.1. Présentation du la téléopération :

Le terme téléopération est composé de deux mots, « télé » de racine grecque signifie à distance, et le mot latin « opération » qui désigne l'exécution d'une certaine tâche. La téléopération désigne donc un mode d'exécution d'une tâche, d'une action, d'un mouvement ou d'un travail à distance. [55]

En robotique, la téléopération, appelée aussi télérobotique, désigne les principes et les techniques qui permettent à un opérateur humain d'accomplir une tâche à distance, à l'aide d'un système robotique d'intervention, commandé à partir d'une station de contrôle, par l'intermédiaire d'un canal de transmission. [55]

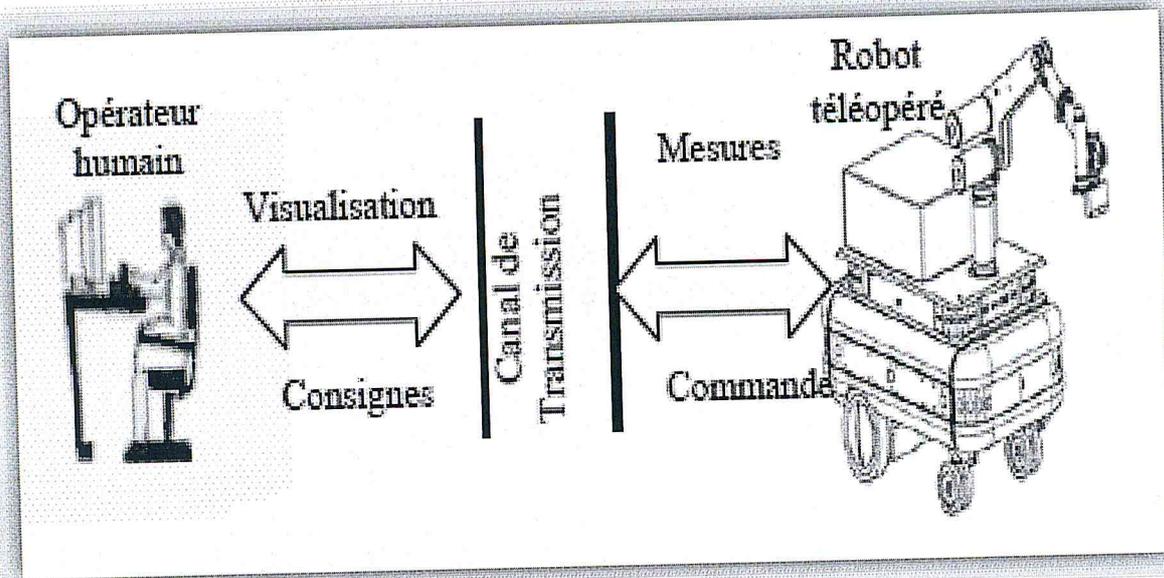


Figure [2.5]: Structure du système de Téléopération. [55]

La téléopération est un axe de recherche actuel, elle suscite beaucoup d'intérêts et trouve son application dans divers disciplines et activités de la vie moderne. Elle concerne la réalisation de travaux dépassant les capacités physiques de l'homme tels que la manipulation d'objets lourds, ou présentant un danger (cas de robots industriels) (figure[2.5]), ou bien des travaux qui sont réalisés sur des sites inconnues ou hostiles à l'homme (tels que: les robots d'exploration martiens , robots industriels, les robots sous-marins, les robots permettant la détection et la réparation de fissures dans des conduits radioactifs, les robots chirurgiens, les robots destinés à l'exploration de l'univers, etc...). On peut citer de grands domaines d'applications de la téléopération, ex le domaine nucléaire, militaire, domaine spatial, le domaine médical.[55]

5.2. Les fonctions de la téléopération

On distingue trois fonctions principales en téléopération, la perception la décision et l'action:

- **La perception** : assure le contrôle de la situation du site distant à partir d'informations issues de capteurs.
- **La décision** : permet de choisir les actions à réaliser en se basant sur l'information issue du module de perception.
- **L'action** : réalisée à l'aide d'un système de commande, elle permet d'agir sur l'environnement.

Ces atouts ont permis de téléopérer des robots esclaves qui se situent à distances considérables par rapport au site maître. [56]

5.3. Les différentes approches dans les systèmes de téléopération

On distingue trois approches pour les systèmes de téléopération. Représentants les modes d'interaction Homme-machine:

❖ La téléopération assistée par ordinateur (TAO)

C'est la commande à distance d'un système robotique par un opérateur humain à travers une interface informatique. Cette structure est née du besoin de réaliser des manipulations intelligentes dans des environnements opérationnels incertains ou éloignés (nucléaire, spatial, médical).

La TAO s'oriente de plus en plus vers le développement des interfaces homme-machine pour la programmation hors ligne, l'entraînement de l'opérateur à la désignation de missions téléopérées etc...

Depuis, la TAO bénéficie d'un atout nouveau qui est la réalité virtuelle. [57]

❖ La téléopération à désignation d'objectif (TDO)

Dans ce mode l'opérateur est vu comme un superviseur, l'intervention de l'opérateur se limite dans ce cas à la désignation d'objectif qui seront réalisés par le robot ; par exemple l'opérateur peut désigner sur l'écran un objet à atteindre par le robot. [58].

❖ La téléopération semi autonome (TAS)

La plupart des systèmes de téléopération actuels, s'orientent vers l'utilisation des deux concepts précédents (TAO et TDO). En effet ils tentent de moderniser la téléopération par une meilleure exploitation simultanée de l'autonomie du robot et des capacités de l'opérateur.

6. Conclusion :

Dans ce chapitre, nous avons commencé par la présentation de la Simulation et la définition de simulateur. Ensuite, nous avons présenté quelques simulateurs commerciaux et open-source et avons fait une comparaison entre eux. Enfin nous avons parlé du simulateur Gazebo, et la téléopération. Le chapitre suivant sera dédié à la conception de notre solution.

Chapitre 3 :

Conception

1. Introduction :

Dans les chapitres précédents, nous avons présenté la robotique en générale et la simulation. Dans ce chapitre, nous allons présenter la conception de notre travail. Dans cette phase pour réaliser la tâche de notre système nous avons choisis le langage de modélisation UML (Unified Modeling Language), en nous basant principalement sur les diagrammes, les cas d'utilisation et les séquences.

D'abord, nous commencerons par présenter notre principe de travail, ensuite nous aborderons l'architecture générale du système puis nous finirons par des diagrammes UML réalisés.

2. Principe de Travail :

Dans cette partie, nous allons présenter l'état de l'existant, ensuite la problématique et enfin l'objectif.

2.1. Etat de l'existant :

Il existe deux états est définie comme suit :

2.1.1. Simulateur :

C'est un simulateur 3D (**figure [3.1]**), permettant de simuler le comportement des robots Roboter/ULM dans un environnement intérieur et aussi construire l'environnement dans lequel doit évoluer les robots virtuels ainsi que simuler le comportement de capteurs et enfin générer les mouvements des robots. *[59]*

➤ Outils Utilisés :

Les outils utilisés pour modéliser ce simulateur le système d'exploitation Linux elles sont utilisées aussi le système d'exploitation robot ROS (Robot Operating System) et langage de programmation python, solidworks pour dessiner le modèle géométrique et Gazibo pour l'environnement. *[59]*

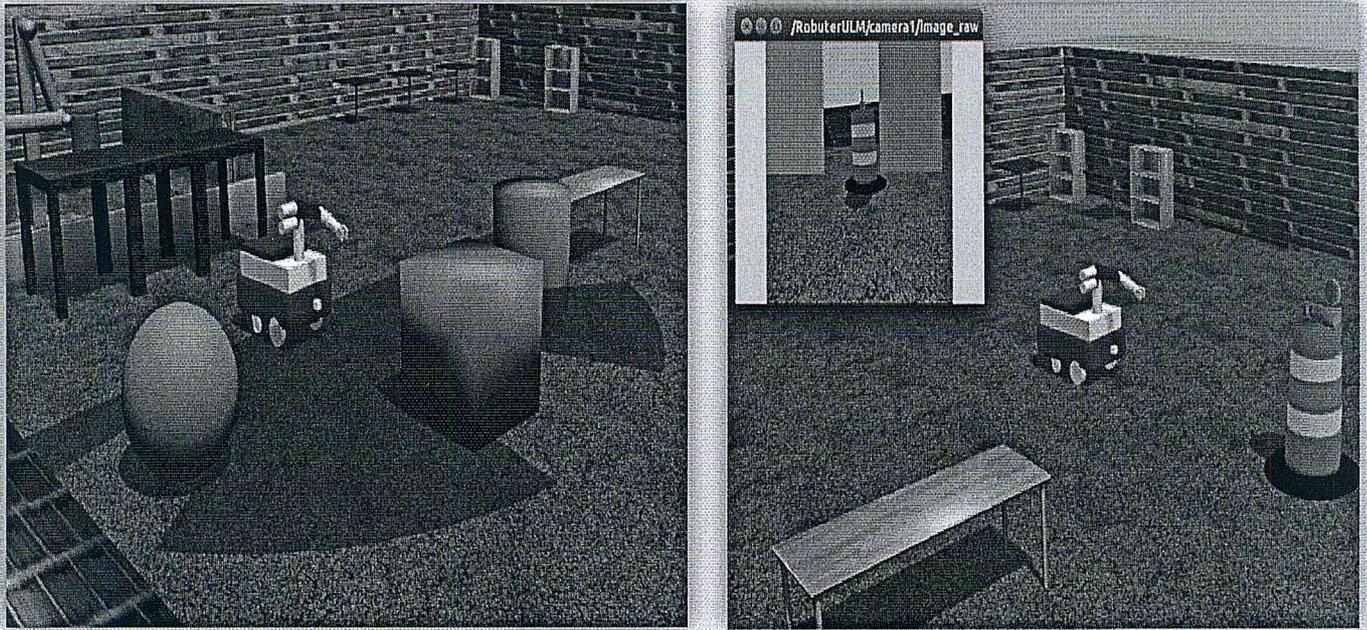


Figure [3.1] : Simulateur 3D Qui Existe. [56]

2.1.2. Architecture de Génération de Mouvement:

C'est une architecture qui comportant deux parties distinctes reliées par une connexion qui gère les flux de données entre le pc embarqué et le pc distant. (Figure [3.3]).

La (figure [3.2]) qui représente les topics les nœuds utilisé dans cette architecture. Et leur communication entre les deux parties (le pc_ros et le pc_embarque) en utilisant les threads avec le protocole de connexion TCP/IP.[2]

➤ Protocole de communication :

Le pc distant connecté dans le réseau local du laboratoire de la division robotique du CDTA par une liaison wifi, dont tous les robots du laboratoire sont connectés et éventuellement Robuter/ULM est connecté dans ce réseau.[2]

➤ Outils Utilisés

Cette architecture est développée par langage de programmation C et C++ et utilisée Linux comme système d'exploitation et le système d'exploitation robot ROS (Robot Operating System).[2]

Conception

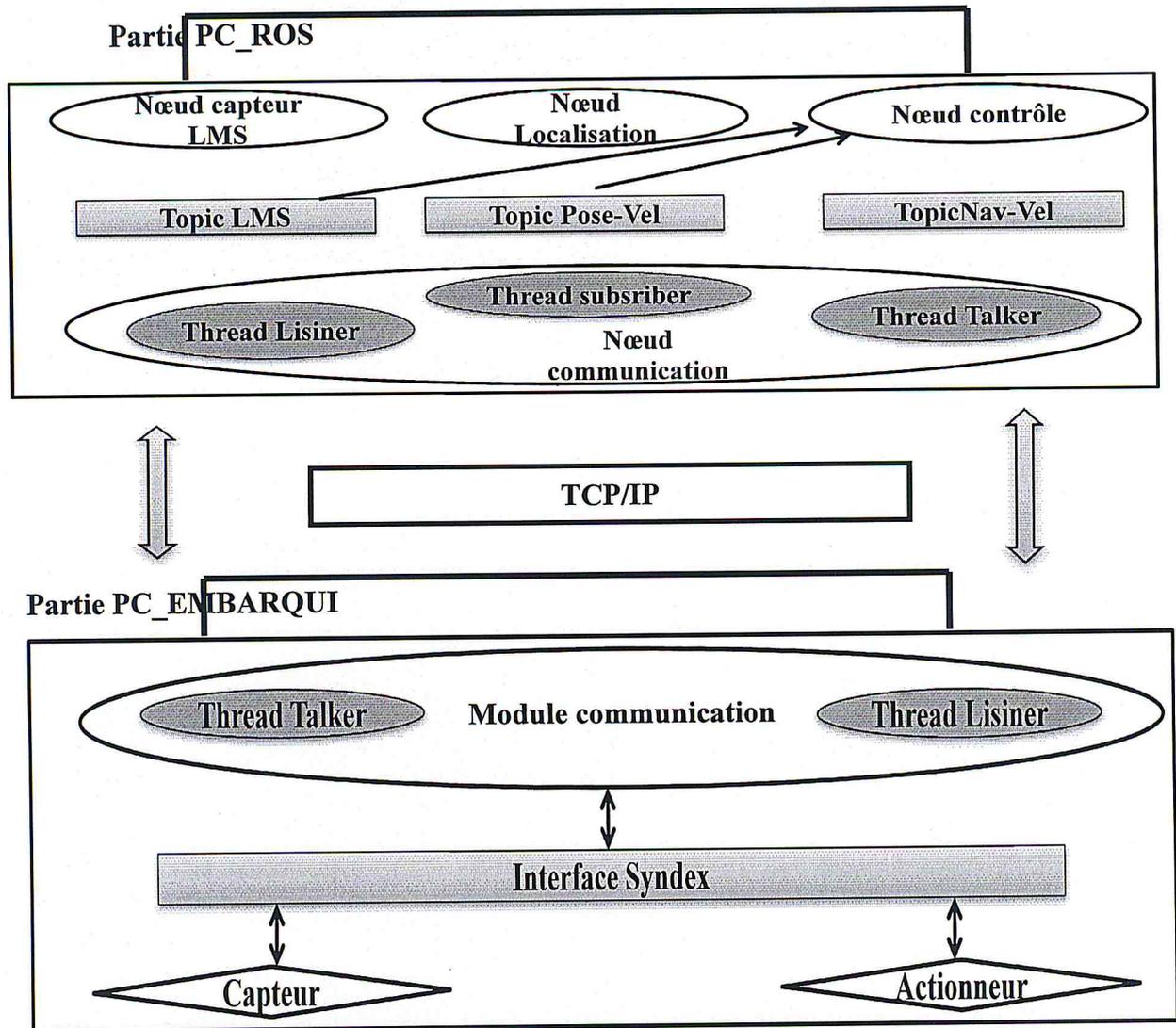


Figure [3.2] : Architecture de perception et de Génération de Mouvement

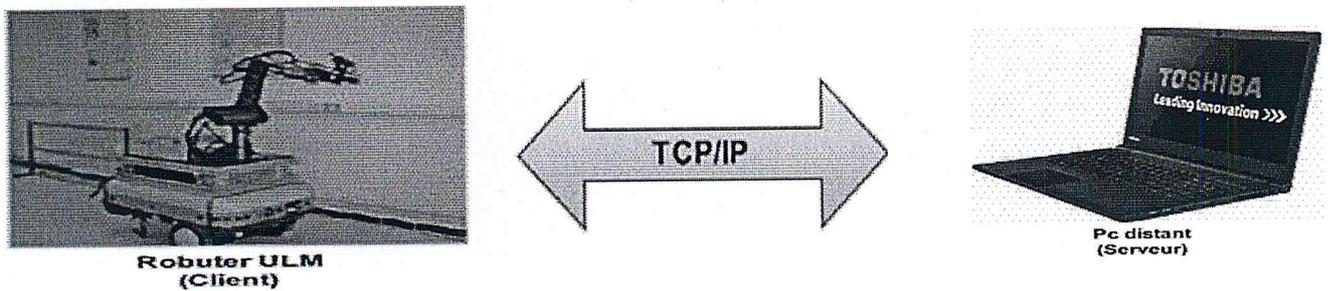


Figure [3.3] : Protocole de communication. [2]

2.2. Description de Problème :

L'équipe de robotique et productique (RP) du Centre de Développement des technologies avancées (CDTA) veut développer un système pour un processus de communication entre un simulateur 3D conçu et réalisé sous Gazebo et un robot manipulateur mobile rééquipé par un pc embarqué, cette communication entre eux se fait par le protocole TCP/IP. Le travail demandé dans ce mémoire est la conception et la réalisation de ce système.

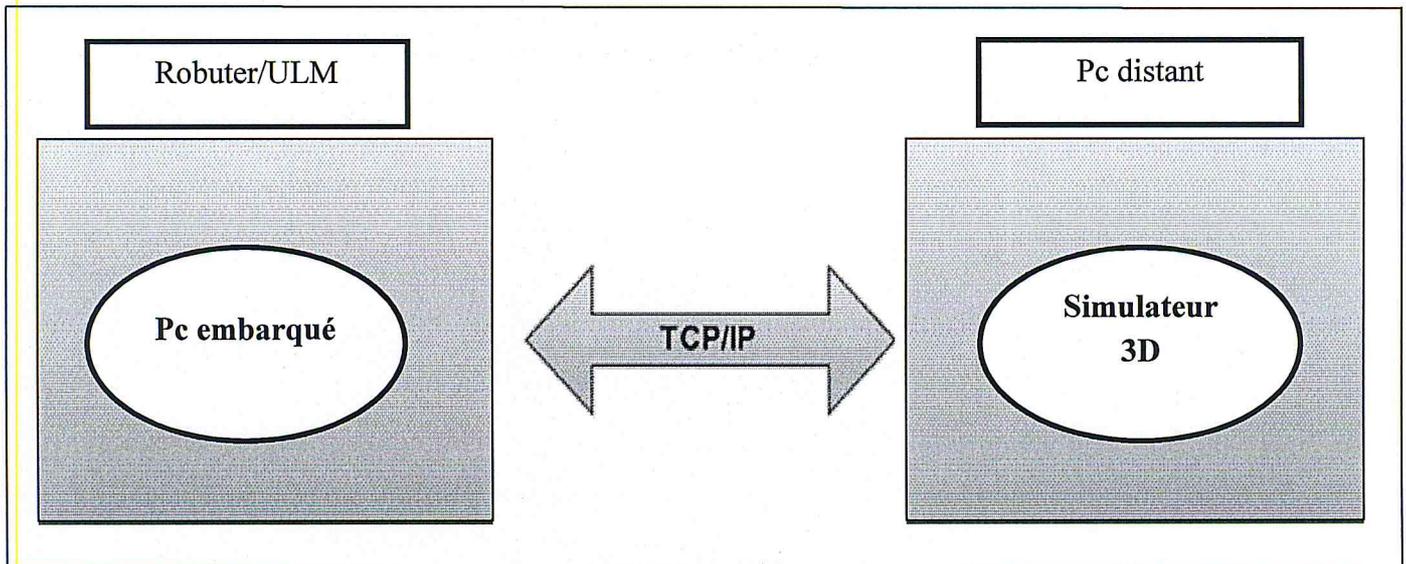


Figure [3.4] : Modèle de l'architecture proposée.

2.3. Objectif :

Le but étant le développement d'une interface avec l'utilisateur qui permette une communication bidirectionnelle entre le simulateur graphique et le robot RobuTER/ULM (Robot réel /simulateur).

Les différents modes de contrôle sont :

- Le contrôle du mouvement du robot réel donc faire appel au simulateur.
- Le contrôle du mouvement du robot simulé sans faire appel au robot réel.
- Le Contrôle du robot réel via le robot simulé.
- Le contrôle du robot simulé via le robot réel.

3. Architecture de Système :

Conception

Après étude nous avons proposé l'architecture suivante représentée ci-dessous :

A. Pour le cas Simulation :

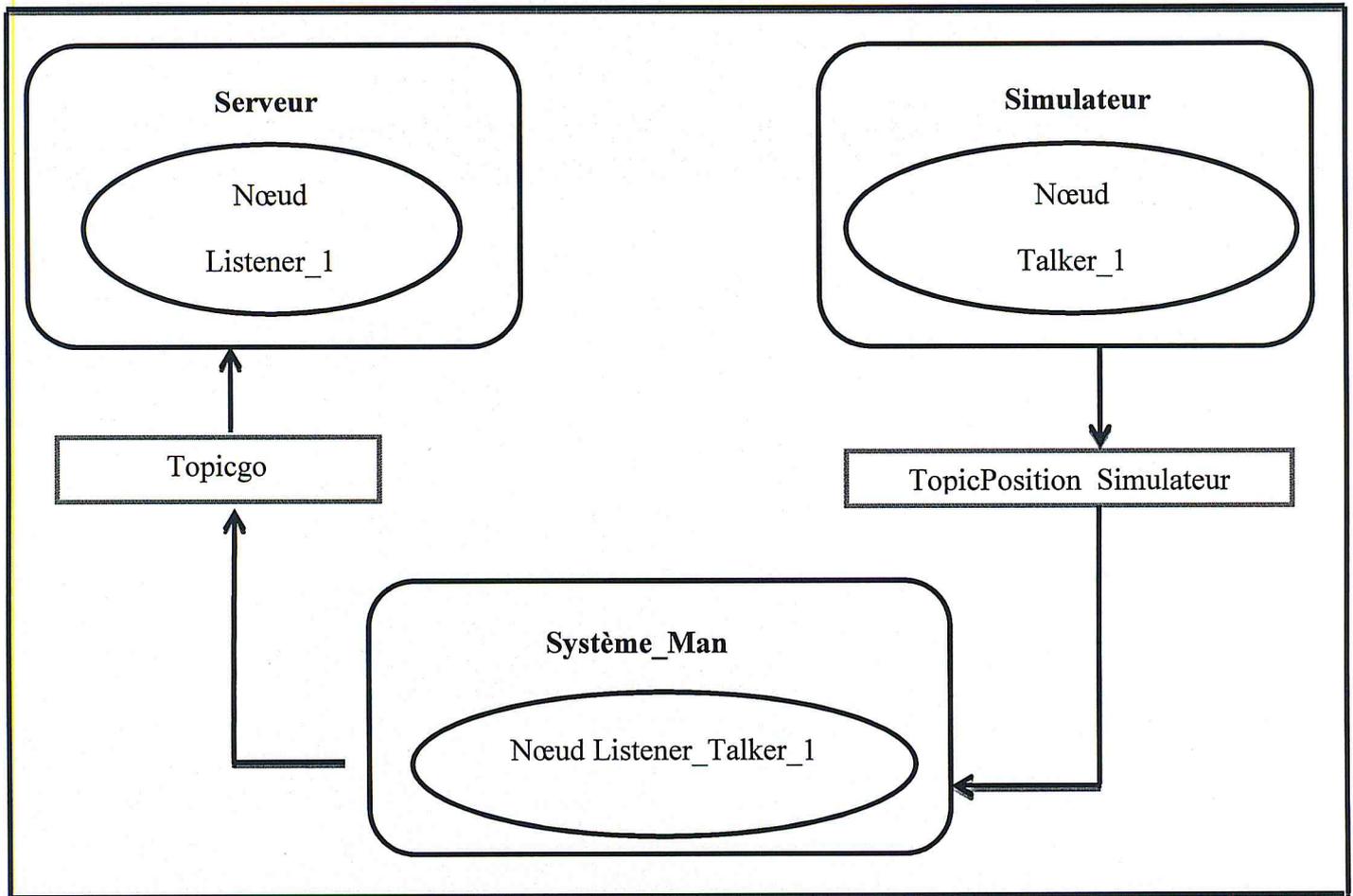


Figure [3.5] : Schéma Global De L'architecture (Cas Simulation).

⇒ Description :

Le nœud **Talker_1** qui se trouve dans le simulateur sert à récupérer les données vitesse et position des roues et des articulations de bras du robot virtuel à partir du Topic/**RobuterULM/joint_state** et les transmet au **TopicPosition_simulateur**.

Le nœud **listener_talker_1** qui se trouve dans le **Système_Man**, reçoit les données du **TopicPosition_simulateur**. Et les transmet en même temps au **TopicGo**.

Le nœud **Listener_1** qui se trouve dans le serveur peut alors déplacer le robot par le Protocol **TCP/IP** grâce aux données qu'il a récupérées du **TopicGo**.

B. Pour le cas Manipulation :

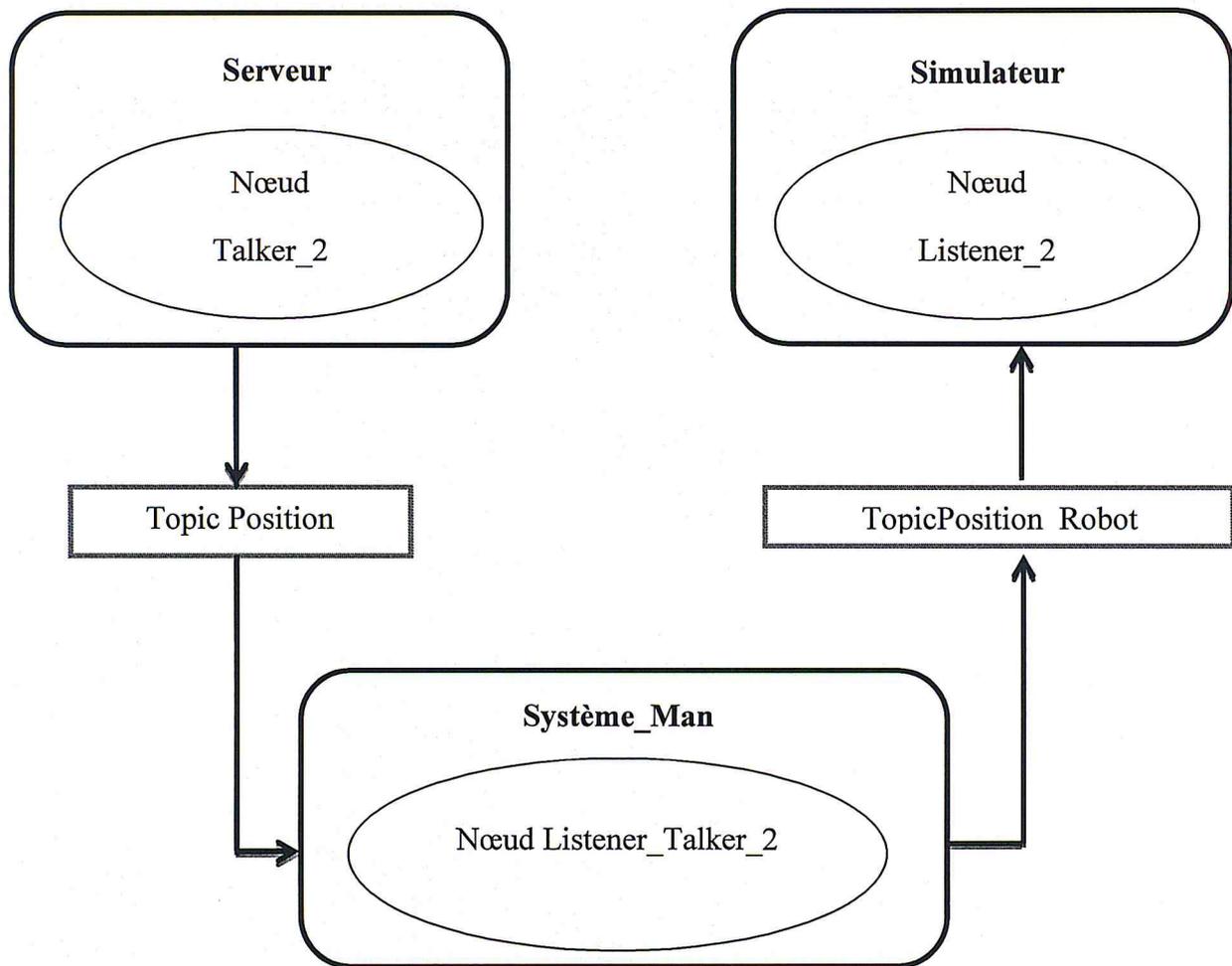


Figure [3.6] : Schéma Globale De L'architecture (Cas Manipulation).

⇒ Description :

Le nœud **Talker_2** qui se trouve dans le serveur publie les données de vitesse et articulation de robot réel dans le **topicPosition**.

Le système_Man reçoit les données se trouvant dans le **topicPosition** et les transmet en même temps au **topicposition Robot**, et tout cela par l'intermédiaire du nœud **listener_talker_1**.

Le nœud **Listener_2** qui se trouve dans le simulateur récupère ces données et les transmet au **Topic/RobuterULM/joint_state**. Afin de déplacer le robot virtuel.

C. Pour le cas Simulation offline :

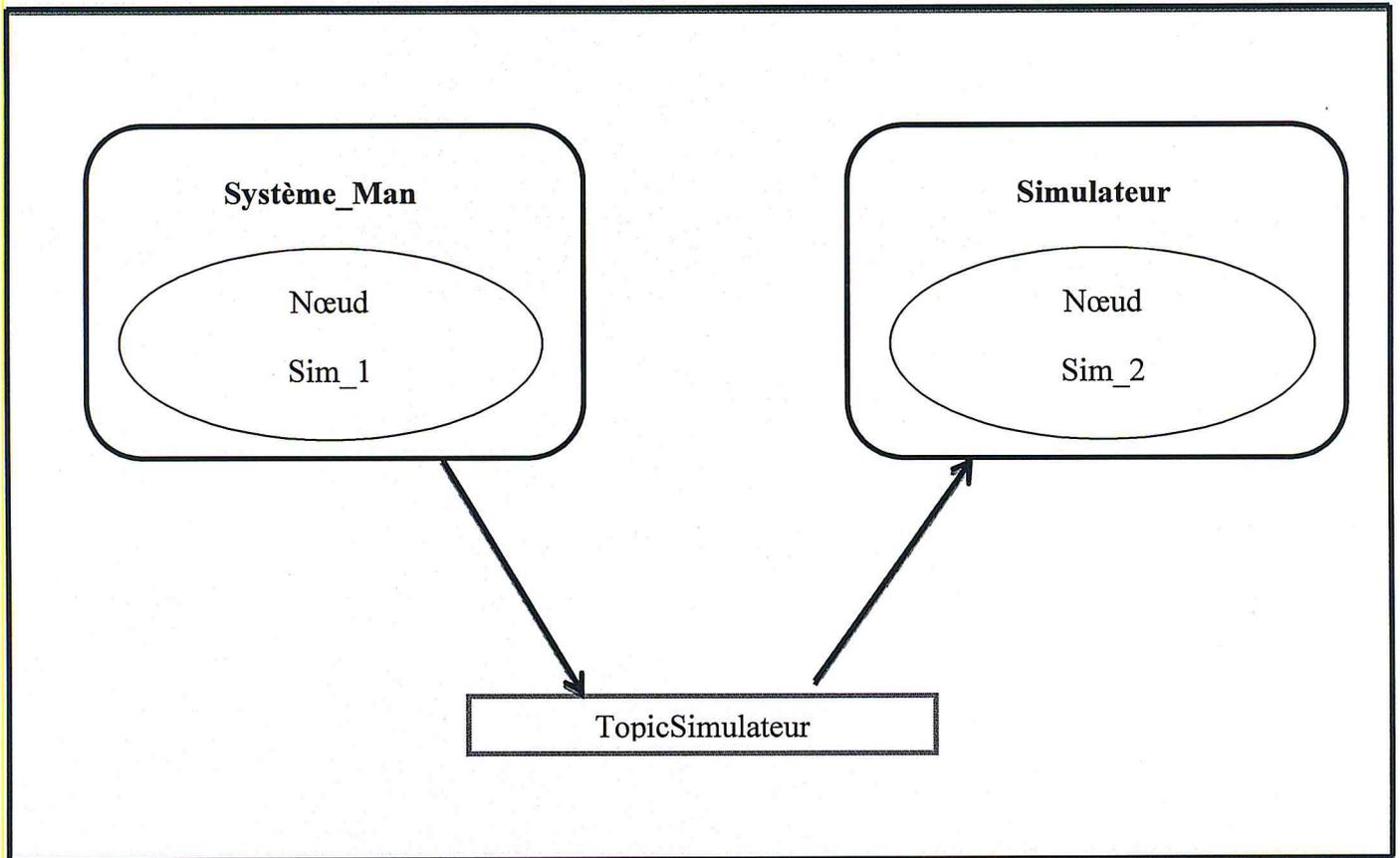


Figure [3.7] : Schéma Globale De L'architecture (Cas Simuler).

☞ Description :

Après que l'utilisateur ait saisi les valeurs des axes de bras affiché par le **Système_Man**, le nœud **Sim_1** envoie ces données au **TopicSimulateur**, ainsi le bras manipulateur bouge suivant les données récupérées par le nœud **Sim_2**.

4. Conception de Notre Système :

4.1. Modélisation en UML :

4.1.1. Diagramme des cas d'utilisation :

UML prend en compte de manière complètement intégrée l'ingénierie des besoins (cas d'utilisation).

4.1.1.1. Diagramme de cas d'utilisation Globale :

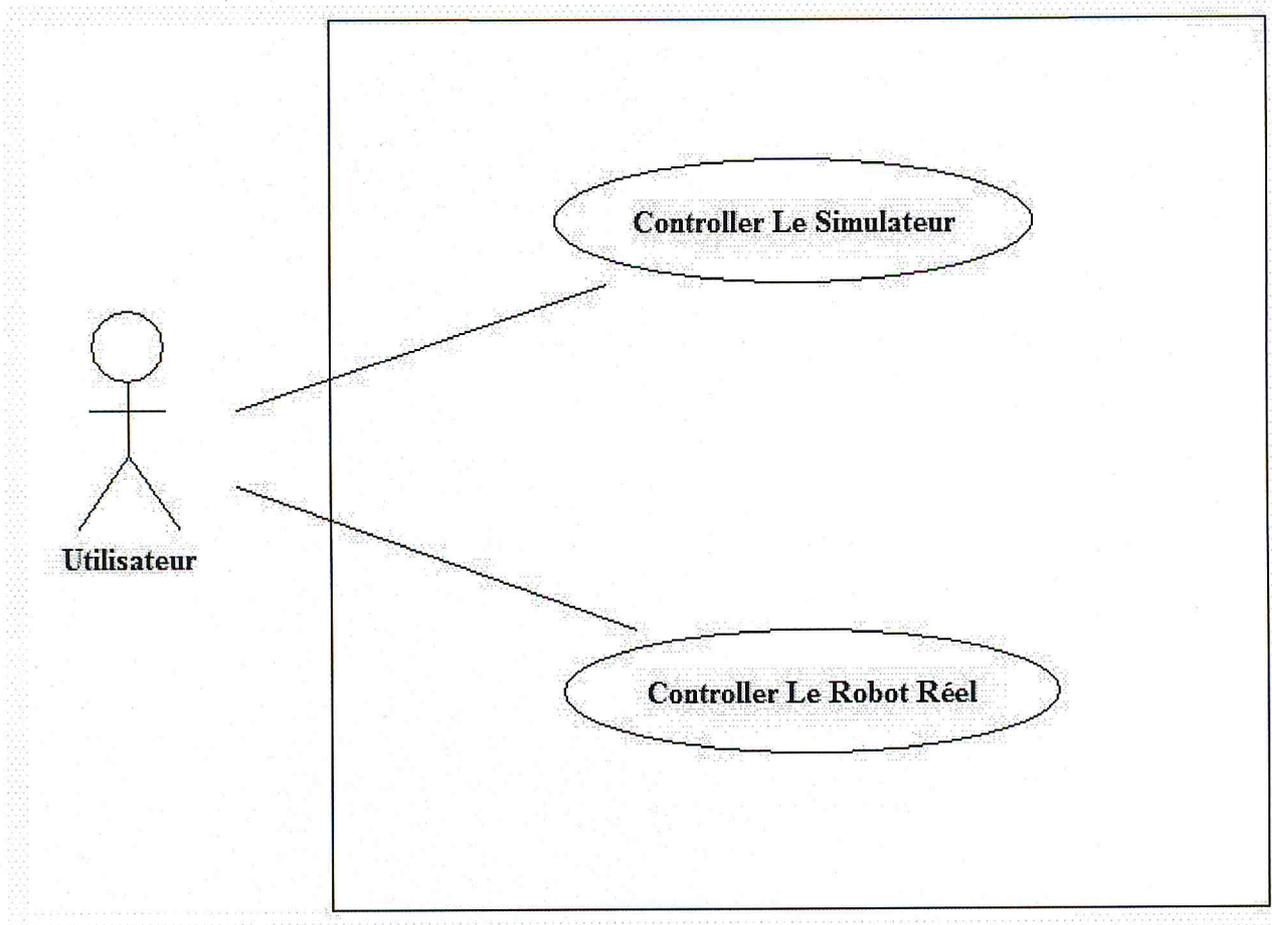


Figure [3.8] : Diagramme de cas d'utilisation Globale.

| Cas d'utilisation | Acteur | Description |
|--------------------------|-------------|--|
| Controller Le Simulateur | Utilisateur | -Simulation ON_LIGNE : contrôler le Robot Réel -Simulation OFF_LIGNE : Contrôler le Robot Virtuelle |
| Controller Le Robot Réel | Utilisateur | -control ON_LIGNE -control OFF_LIGNE |

Tableau [3.1] : Description du Diagramme de cas d'utilisation Globale.

4.1.1.2. Diagramme de cas d'utilisation Controller Le Robot Réel :

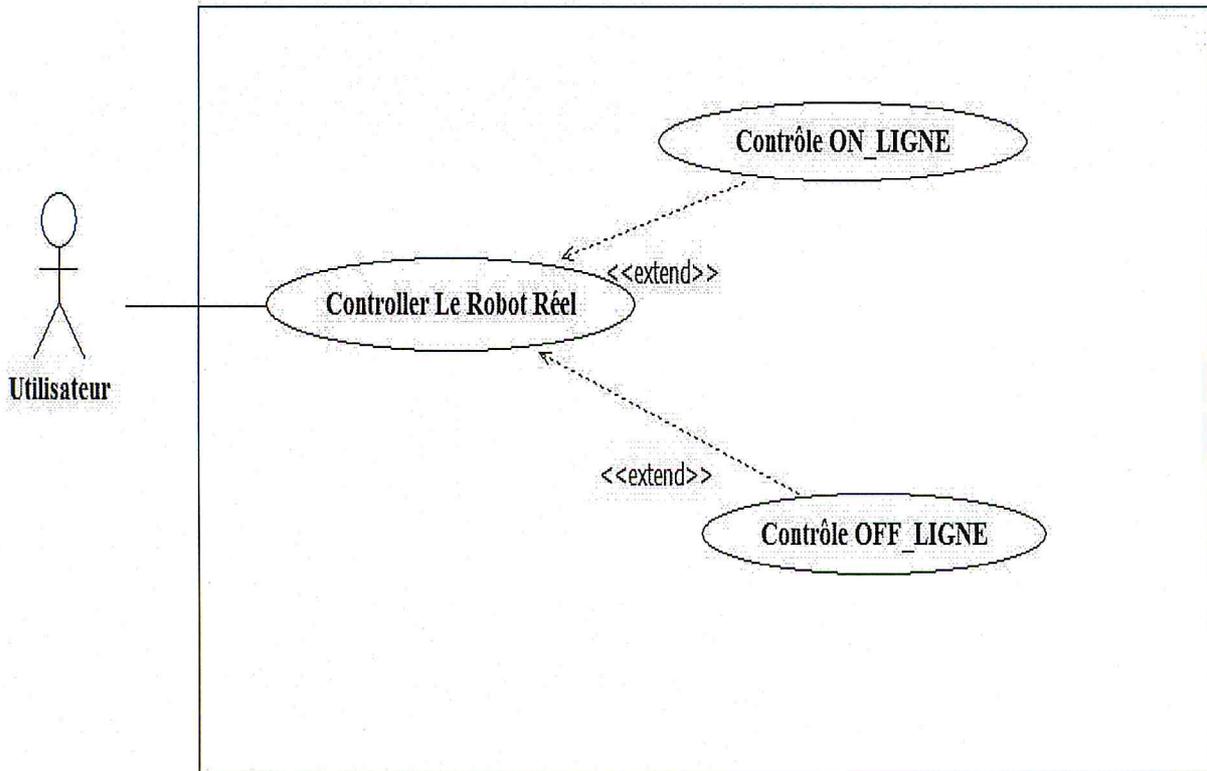


Figure [3.9] : Diagramme de cas d'utilisation Controller Le Robot Réel.

| Cas d'utilisation | Acteur | Description |
|--------------------|-------------|---|
| Contrôle ON_LIGNE | Utilisateur | -contrôle le Simulateur : robot réel envoi les données vers le simulateur pour déplacement. |
| Contrôle OFF_LIGNE | Utilisateur | -contrôle manuel : pilote le robot réel de manière manuel. |

Tableau [3.2] : Description du Diagramme de cas d'utilisation Controller Le Robot Réel.

4.1.1.3. Diagramme de cas d'utilisation Controller Le Simulateur :

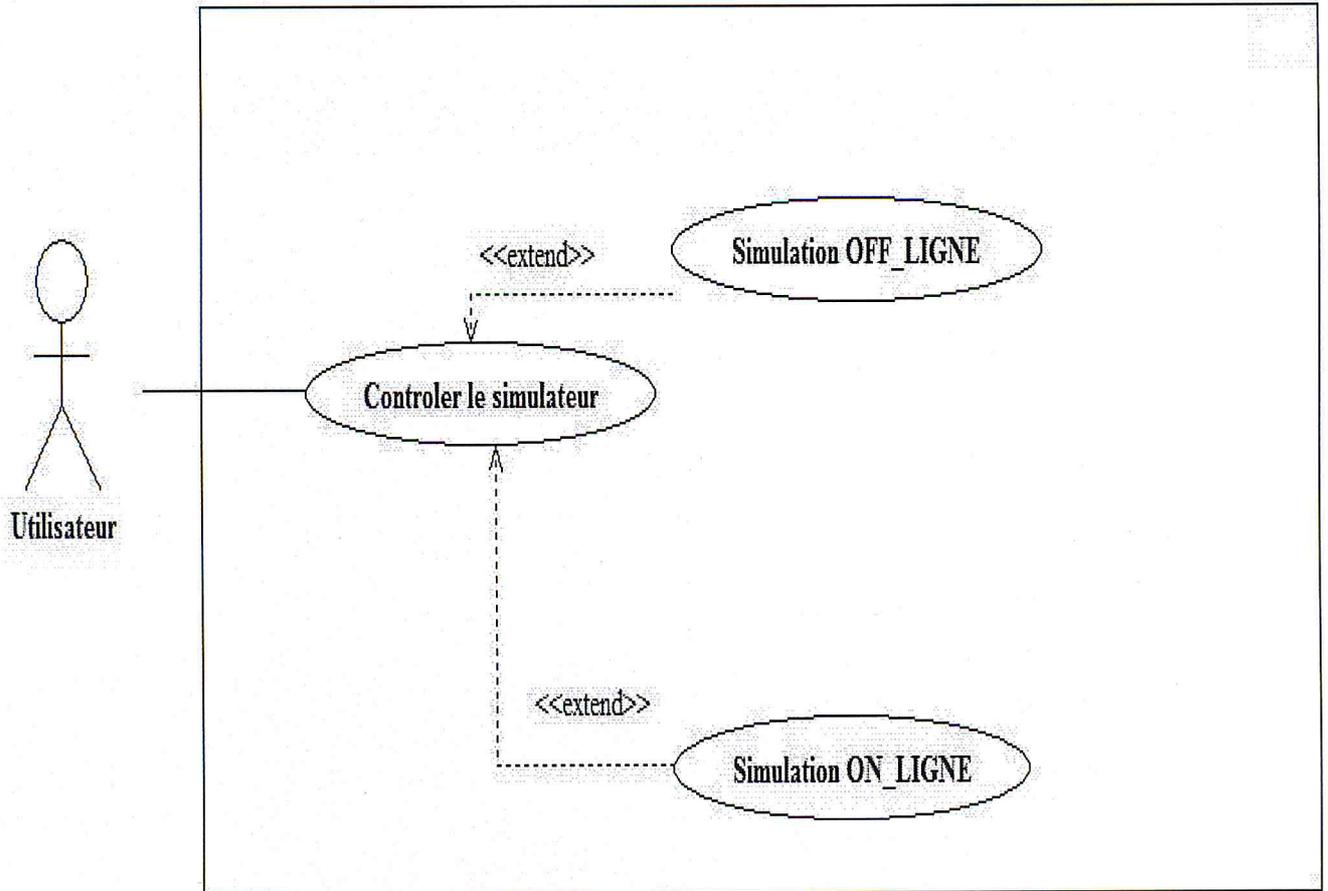


Figure [3.10] : Diagramme de cas d'utilisation Controller Le Simulateur.

| Cas d'utilisation | Acteur | Description |
|----------------------|-------------|---|
| Simulation ON_LIGNE | Utilisateur | Controler le Robot Réel : le simulateur envoi les données vers robot réel pour déplacement. |
| Simulation OFF_LIGNE | Utilisateur | Controler le Robot Virtuelle : Déplacer naviguer le R.V |

Tableau [3.3] : Description du Diagramme de cas d'utilisation Controller Le Simulateur.

4.1.2. Diagramme De Séquence :

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel et peuvent servir à illustrer des cas d'utilisation. Donc, on y met l'accent sur la chronologie des envois de messages.

4.1.2.1. Diagramme De Séquence Cas De Simulation :

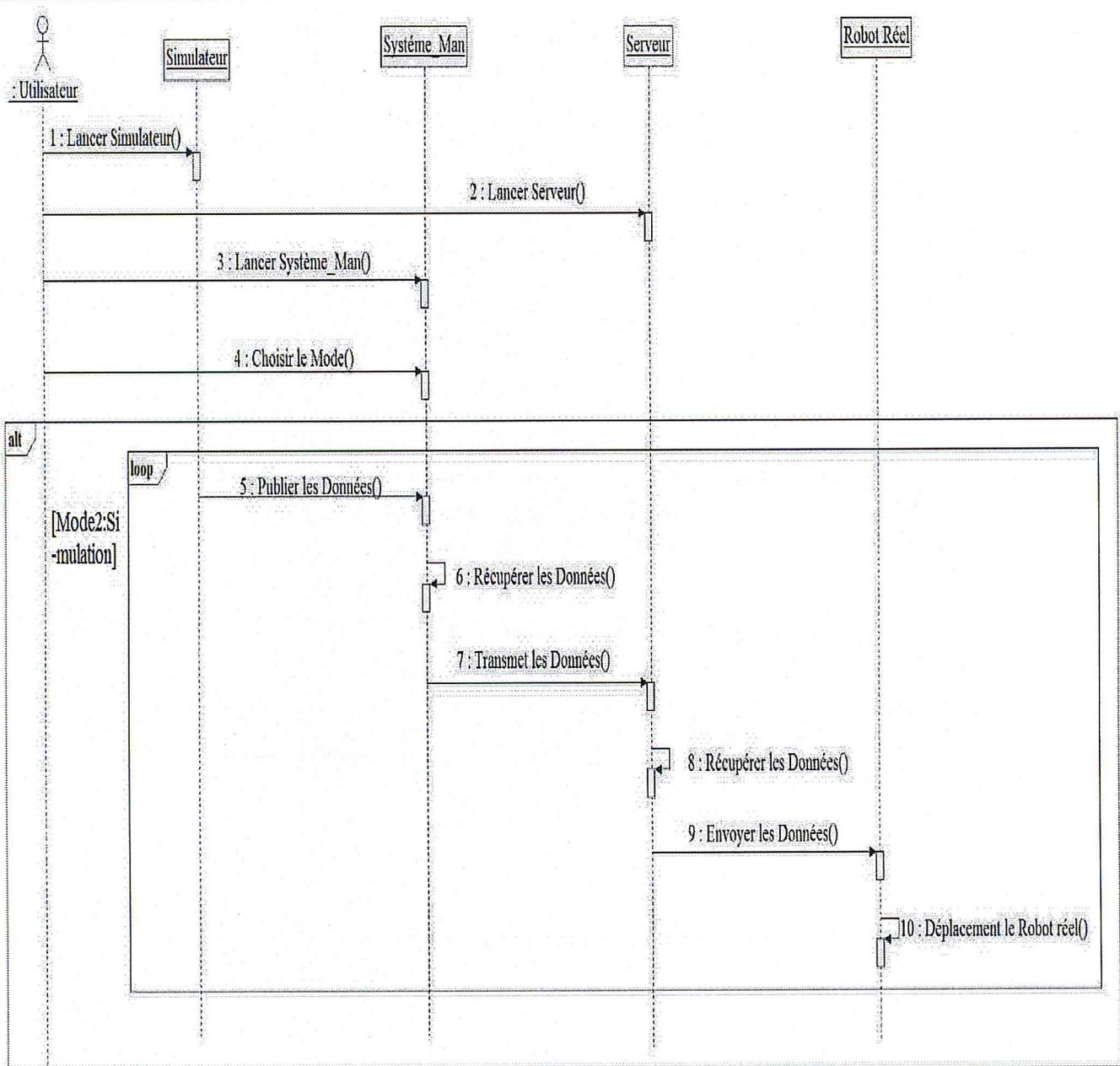


Figure [3.11] : Diagramme de Séquence Cas De Simulation.

4.1.2.2. Diagramme De Séquence Cas De Manipulation :

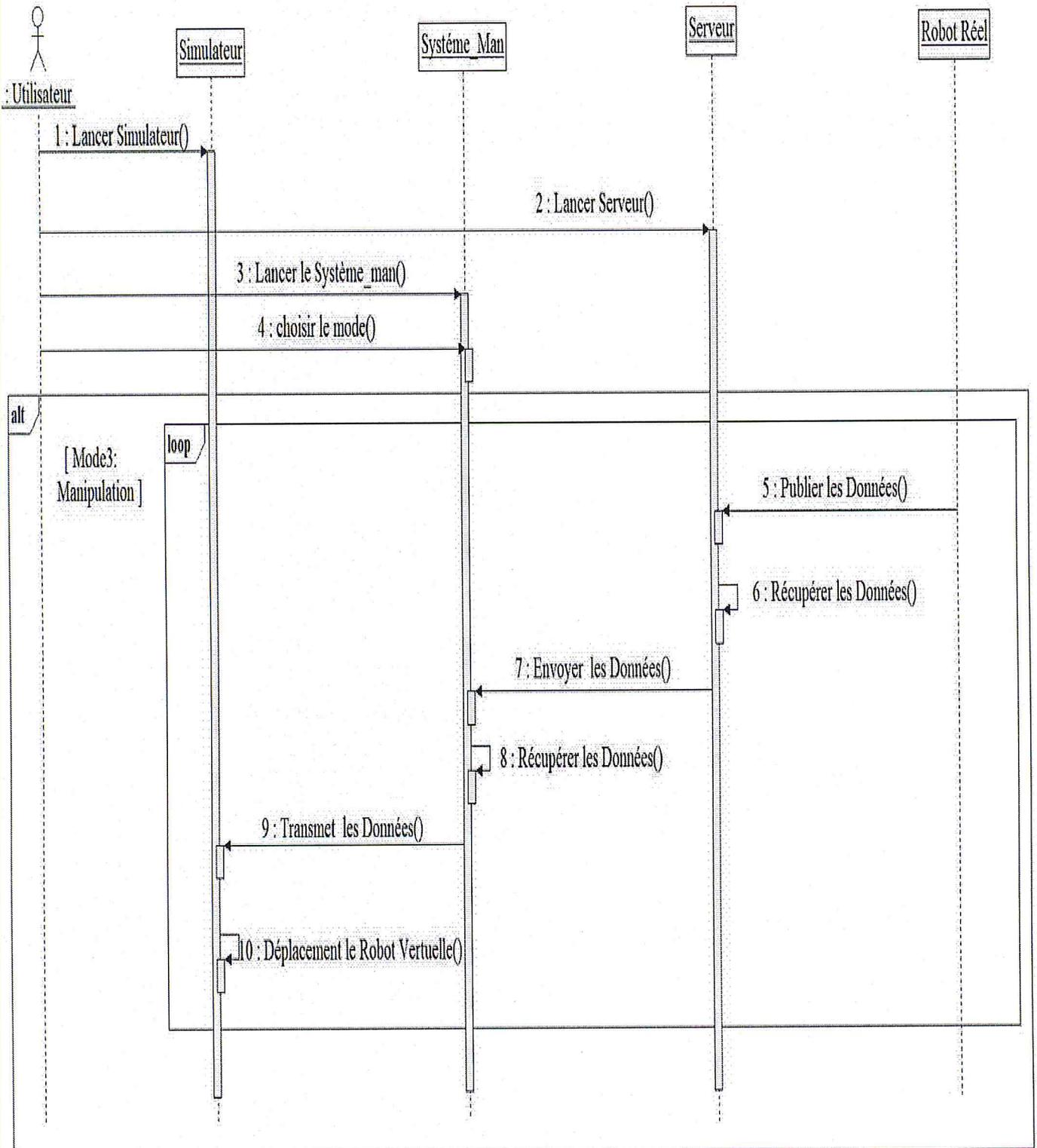


Figure [3.12] : Diagramme de Séquence Cas De Manipulation.

4.1.2.3. Diagramme De Séquence Cas De Simuler (simulation offline) :

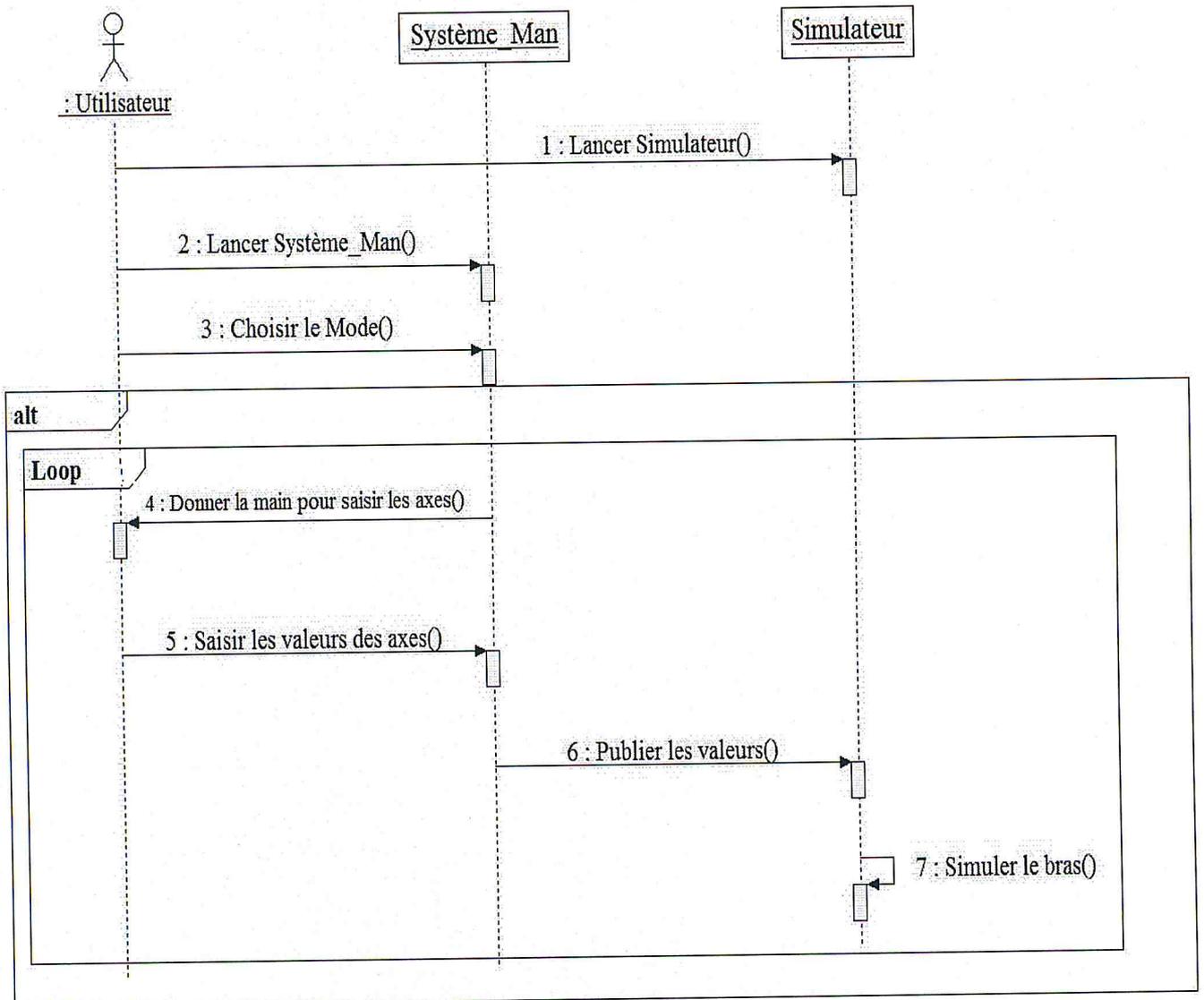


Figure [3.13] : Diagramme de Séquence Cas De Simuler.

4.1.3. Diagramme De déploiement:

Un diagramme de déploiement décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nœud, le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds. Les diagrammes de déploiement existent sous deux formes : spécification et instance.

4.1.3.1. Diagramme De déploiement général :

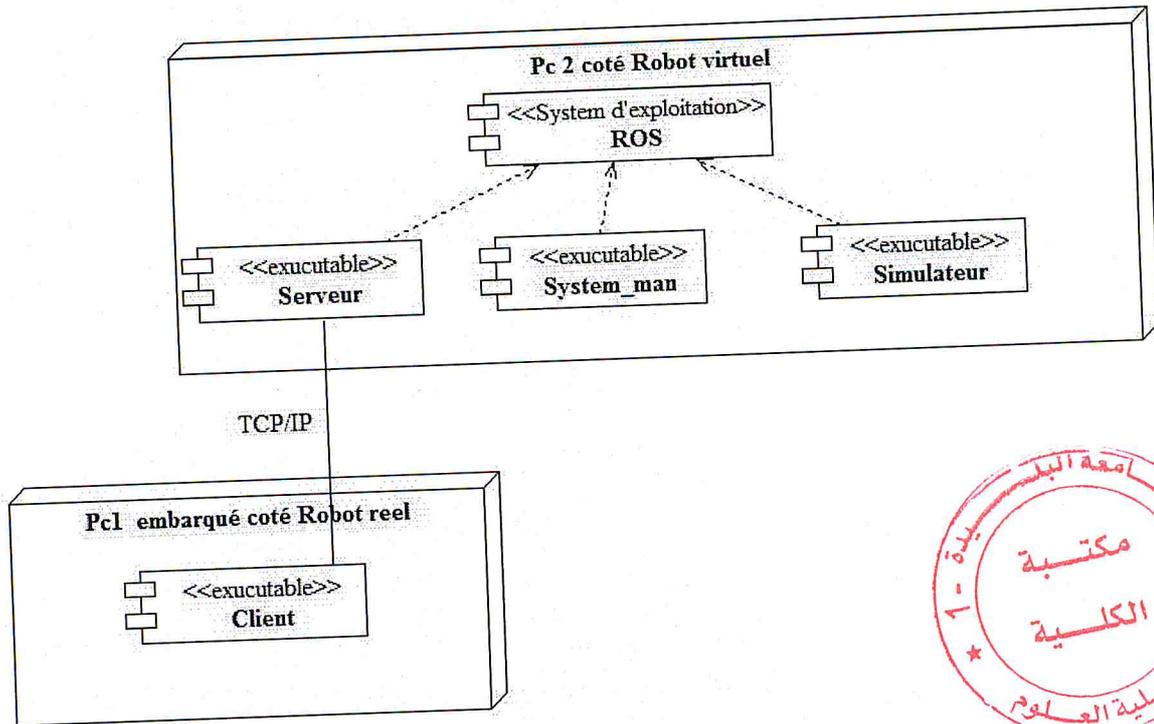


Figure [3.14] : Diagramme de déploiement général.

5. Conclusion :

Dans ce chapitre on a présenté le plan de notre travail en commençant par une description des problèmes et les objectifs, ensuite nous avons proposé une solution avec les différents diagrammes du langage de modélisation UML, Dans le chapitre suivant nous allons passer à l'étape de l'implémentation et le test de l'architecture.

Chapitre 4 :

Implémentation

Implémentation

1. Introduction :

Dans ce chapitre, nous allons présenter la mise en œuvre de notre application, En commençant tout d'abord par une présentation des outils de développement utilisés par une vue détaillée sur les systèmes d'exploitations et les langages de programmation utilisés.

Ensuite nous définirons l'état de l'existant. Et enfin nous présenterons les captures d'écran de l'exécution de notre application.

2. Outil de Développement :

2.1. Linux :

Linux est un système d'exploitation qui a d'abord été créé à l'Université d'Helsinki en Finlande par un jeune étudiant nommé Linus Torvalds. À cette époque, l'étudiant travaillait sur un système UNIX qui fonctionnait sur une plate-forme coûteuse. En raison de son faible budget et de son besoin de travailler à la maison, Il a décidé de créer une copie du système UNIX afin de l'exécuter sur une plate-forme moins coûteuse, Comme un PC IBM. Il a commencé son travail en 1991 lorsqu'il a publié la version 0.02 et a travaillé en permanence jusqu'en 1994, lorsque la version 1.0 du Kernel Linux a été diffusée.

L'actualité complète la version à ce moment est 2.2.X (publié le 25 janvier 1999), et le développement se poursuit.[60]



Figure [4.1] : interface et logo d'ubuntu.[61]

2.2. Langage de programmation C++ :

Apparu au début des années 90, le langage C++ est actuellement l'un des plus utilisés dans le monde, aussi bien pour les applications scientifiques que pour le développement des logiciels. En tant qu'héritier du langage C, le C++ est d'une grande efficacité. Mais il a en plus des fonctionnalités puissantes, comme par exemple la notion de classe, qui permet d'appliquer les

Implémentation

techniques de la programmation-objet. Le but de ce cours est de présenter la syntaxe de base du langage C++. Certains traits propres au C, dont l'usage s'avère périlleux, sont passés sous silence. La programmation-objet, quant à elle, sera abordée dans un autre cours. [62]

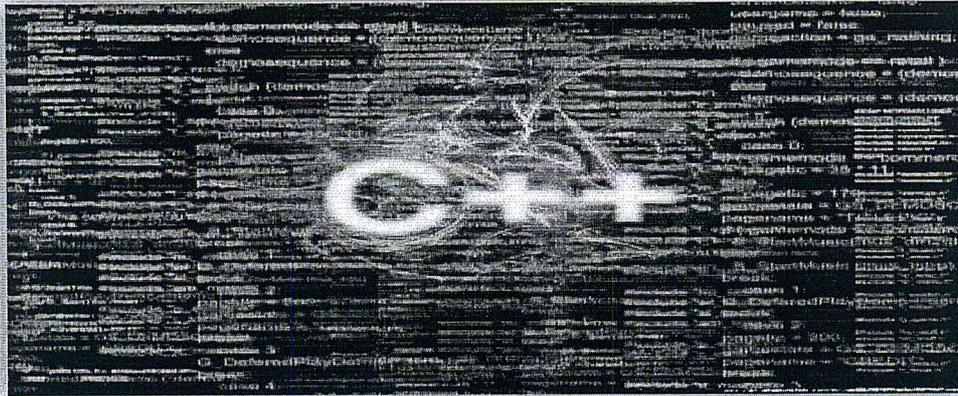


Figure [4.2] :logo de C++. [63]

2.3. Langage de Programmation Python :

2.3.2. Définition du logiciel Python :

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et d'une approche de la programmation orientée objet simple mais efficace. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur de nombreuses plateformes.



Figure [4.3] : Logo de Python. [64]

2.4. ROS :

2.4.1. Définition :

Comme son nom l'indique, ROS (Robot Operating System) est un système d'exploitation pour robots. De même que les systèmes d'exploitation pour PC, serveurs ou appareils autonomes, ROS est un système d'exploitation complet pour la robotique de service.

ROS est un méta système d'exploitation, quelque chose entre le système d'exploitation et le middleware. Il fournit des services proches d'un système d'exploitation (abstraction du matériel, gestion de la concurrence, des processus...) mais aussi des fonctionnalités de haut niveau (appels asynchrones, appels synchrones, base de données centralisée de données, système de paramétrage du robot...). [2]

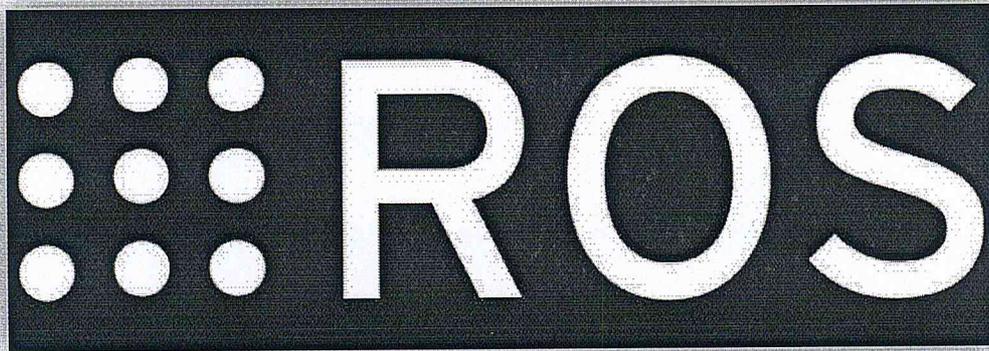


Figure [4.4] : Logo de ROS. [65]

2.4.2. L'apport de l'outil ROS :

- ❖ ROS apporte un environnement de développement non tributaire du hardware pour le domaine de la robotique.
- ❖ Il permet la portabilité des programmes développés sur une plateforme donnée vers une autre plateforme.
- ❖ Il offre un gain de temps énorme en donnant la possibilité d'implémenter les mêmes algorithmes parfois tributaires du hardware, il suffit de les adapter à la plateforme désirée. [66]

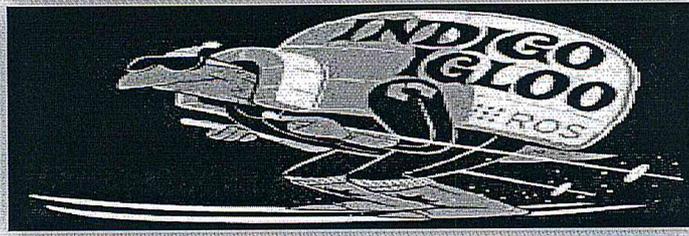


Figure [4.5] : Logo d'outil ROS. [67]

2.4.3. Le système de fichiers de ROS :

Les nœuds :

Un nœud est une instance d'un exécutable. Un nœud peut correspondre à un capteur, un moteur, un algorithme de traitement, de surveillance. Chaque nœud qui se lance se déclare au Master. On retrouve ici l'architecture microkernel où chaque ressource est un nœud indépendant.

Le Master :

Le **Master** est un service de déclaration et d'enregistrement des nœuds qui permet ainsi à des nœuds de se connaître et d'échanger de l'information. Le Master est implémenté via XMLRPC.

Le Master comprend une sous-partie très utilisée qui est le **Parameter Server**. Celui-ci, également implémenté sous forme de XMLRPC, comme son nom l'indique est une sorte de base de données centralisée dans laquelle les nœuds peuvent stocker de l'information et ainsi partager des paramètres globaux.

Les Topics :

Un **topic** est un système de transport de l'information basé sur le système de l'abonnement / publication (subscribe / publish). Un ou plusieurs nœuds pourront publier de l'information sur un **topic** et un ou plusieurs nœuds pourront lire l'information sur ce **topic**

Le **topic** est typé, c'est-à-dire que le type d'information qui est publiée (le **message**) est toujours structuré de la même manière. Les nœuds envoient ou reçoivent des messages sur des topics.

Les Message :

Un **message** est une structure de donnée composite. Un message est composé d'une combinaison de types primitifs (chaines de caractères, booléens, entiers, flottants...) et de message (le message est une structure récursive). Par exemple un nœud représentant un servomoteur du robot, publiera certainement son état sur un topic (selon ce que vous aurez programmé) avec un message contenant par exemple un entier représentant la position du moteur, un flottant représentant sa température, un autre flottant représentant sa vitesse.

La description des messages est stockée dans *nom_package/msg/monMessageType.msg*. Ce fichier décrit la structure des messages.

Les Services :

Le topic est un mode de communication asynchrone permettant une communication many-to-many. Le **service** en revanche répond à une autre nécessité, celle d'une communication synchrone entre deux nœuds. Cette notion se rapproche de la notion d'appel de procédure distante (remoteprocedure call).

La description des services est stockée dans *nom_package/srv/monServiceType.srv*. Ce fichier décrit les structures de données des requêtes et des réponses.

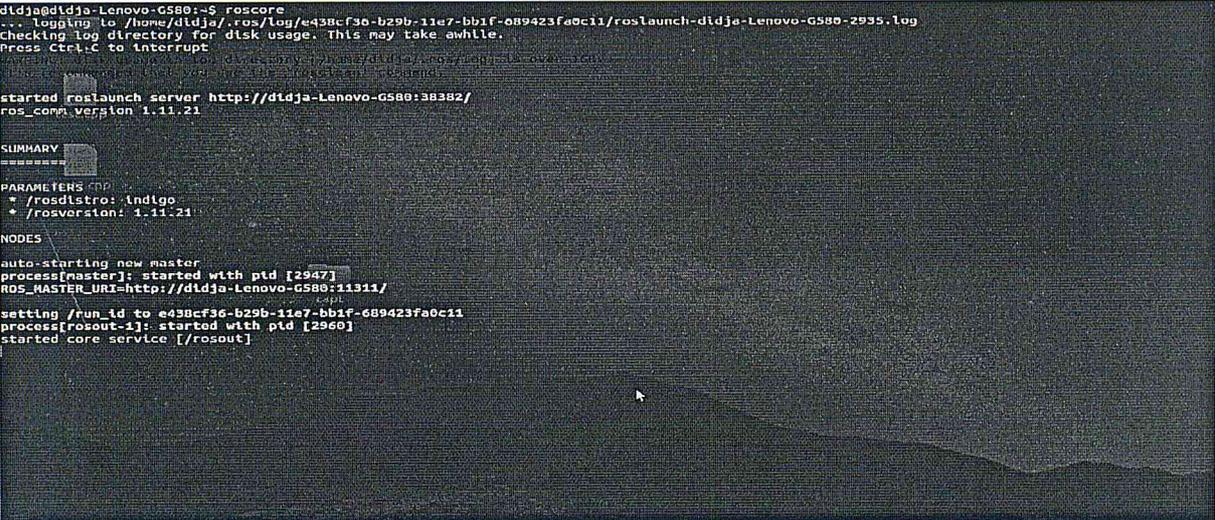
Les Bags :

Les « bags » sont des formats pour stocker et rejouer les messages échangés. Ce système permet de collecter par exemple les données mesurées par les capteurs et les rejouer ensuite autant de fois qu'on le souhaite afin de faire de la simulation sur des données réelles. Ce système est également très utile pour déboguer un système a posteriori.

L'outil `rxbag` permet de visualiser graphiquement les données enregistrées dans les fichiers bag.

4. Test et Validation de L'application :

La commande `roscore`, permet à l'utilisateur de lancer `ros` (figure [4.6]), pour que les nœuds puis communiquent entre eux.



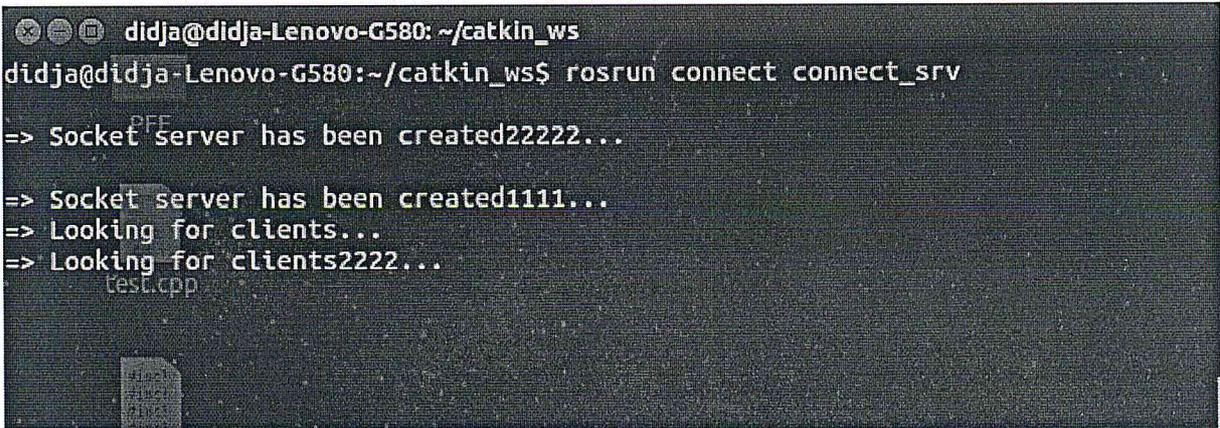
```
didja@didja-Lenovo-G580:~$ roscore
.. Logging to /home/didja/.ros/log/e438cf36-b29b-11e7-bb1f-689423fa0c11/roslaunch-didja-Lenovo-G580-2935.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl.C to interrupt
..
started roslaunch server http://didja-Lenovo-G580:38382/
ros_core version 1.11.21

SUMMARY
=====
PARAMETERS [DP]
 * /roslstro: indigo
 * /rosversion: 1.11.21

NODES
auto-starting new master
process[master]: started with pid [2947]
ROS_MASTER_URI=http://didja-Lenovo-G580:11311/
setting /run_id to e438cf36-b29b-11e7-bb1f-689423fa0c11
process[rosout-1]: started with pid [2960]
started core service [/rosout]
```

Figure [4.6] : Roscore.

Lancement du l'exécutable de serveur (figure [4.7]), qui permet la communication entre le robot real et robot virtuel.



```
didja@didja-Lenovo-G580:~/catkin_ws
didja@didja-Lenovo-G580:~/catkin_ws$ roslaunch connect connect_srv
=> Socket server has been created22222...
=> Socket server has been created1111...
=> Looking for clients...
=> Looking for clients2222...
test.cpp
```

Figure [4.7] : le nœud du serveur.

Lancement du l'exécutable de `System_man` (figure [4.8]), l'interface qui permet à l'utilisateur de choisir un mode de communication.

Implémentation

```
didja@didja-Lenovo-G580: ~/catkin_ws
didja@didja-Lenovo-G580:~/catkin_ws$ source ~/catkin_ws/devel/setup.bash
didja@didja-Lenovo-G580:~/catkin_ws$ rosrunc connect connect_man
selectionner un mode
1:mode manuel
2:mode Simulation
3:mode manipulation
4:Mode simuler
```

Figure [4.8] : le nœud du System_man.

Lancement de l'environnement de gazebo (figure [4.9]), avec la commande `roslaunch RobuterULMgazebo.launch`



Figure [4.9] : l'environnement de gazebo.

Lancement de l'exécutable de C_bras (partie simulateur) (figure [4.10]), Contrôle de bras du robot virtuel.

Implémentation

```
didja@didja-Lenovo-G580: ~/catkin_ws
didja@didja-Lenovo-G580:~/catkin_ws$ rosrunc RobuterULM C_bras.py
[INFO] [WallTime: 1508177268.576899] [59.204000] layout:
  dim: []
  data_offset: 0
data: [10.0, 12.0, 14.0, 14.0, 10.0, 5.0, 8.0, 8.0, 0.0, 0.0]
[INFO] [WallTime: 1508177271.787614] [62.208000] 10.0
```

Figure [4.10] : le nœud du C_bras.

Lancement de l'exécutable de subs_j_s (partie simulateur) (figure [4.11]), Contrôle de robot real à partir du robot virtuel.

```
didja@didja-Lenovo-G580: ~/catkin_ws
didja@didja-Lenovo-G580:~/catkin_ws$ rosrunc RobuterULM subs_j_s.py
[INFO] [WallTime: 1508177395.893910] [180.200000] pos_table
[INFO] [WallTime: 1508177395.894163] [180.200000] (-3.3101453666972702e-06, -1.6
413272039272897e-06, -2.188652140411307e-06, -4.165066748385016e-06, -1.34936517
42635393e-06, -8.51498733016598e-05, -0.05158950620906744, 0.11747980754209664,
-0.002014650004991344, 3.116587547785871e-05, -0.0001592838000178242, -0.0018646
623946902523, 4.3828769525447366e-05, 0.0013392043738404756, 7.256316375634647e-
08, 5.984631559156212e-08)
^Cdidja@didja-Lenovo-G580:~/catkin_ws$
```

Figure [4.11] : le nœud du subs_j_s.

4.1. Mode Simulation :

L'utilisateur lance le simulateur, le serveur, et le système_man par des interfaces du terminal .après lancement, le simulateur affiche le robot virtuel et son environnement .Le système_man affiche une liste de mode pour que l'utilisateur en choisit un, dans cette liste il existe quatre modes : Le mode manuel, le mode simulation, le mode manipulation et le mode simuler. L'utilisateur choisit le deuxième mode c.à.d. la simulation, dans ce cas le simulateur transmet les données (vitesse et position de robot virtuel) au système_man ce dernier récupère ces données et les transmet en même temps au serveur qui les envoie à son tour au robot réel .afin que le déplacement du robot virtuel et le robot réel se fait en même temps.

La figure [4.12] qui représente l'environnement du simulateur, et les deux interfaces de la terminale la première en haut est celle du système_man (pour choisir le mode), et la deuxième du bas concerne le nœud du simulateur qui publie les données.

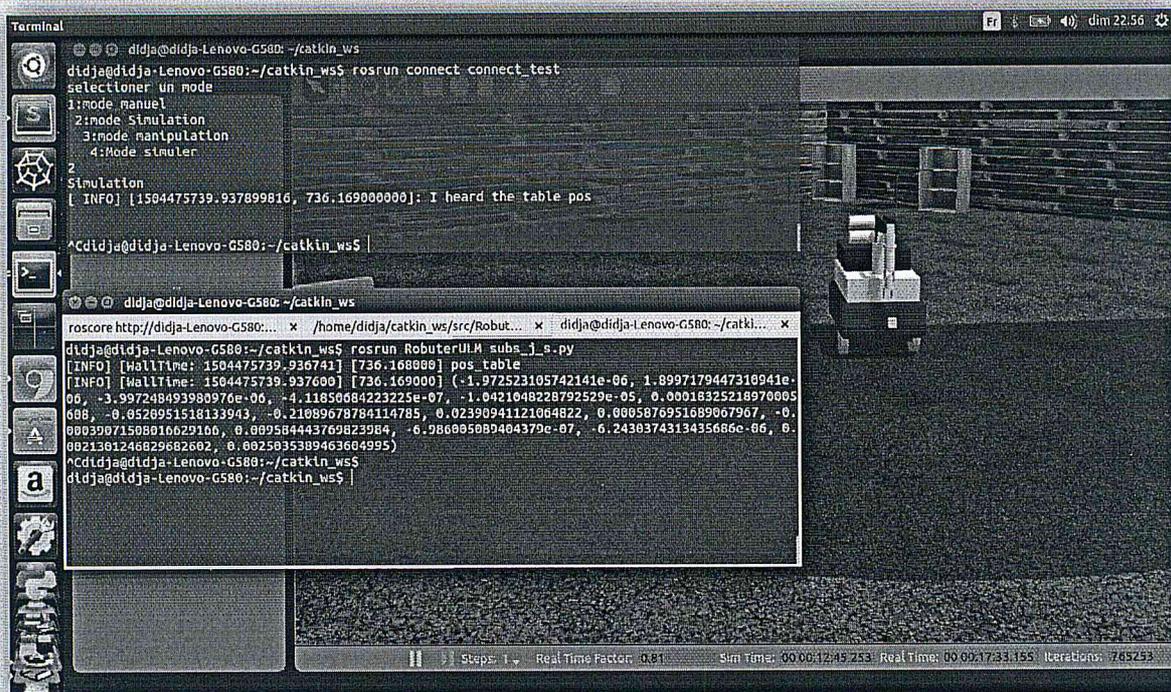


Figure [4.12] : Mode Simulation.

4.2. Mode Simuler (Pos1) :

Après le lancement du simulateur et du système Man par interface sur le terminal L'utilisateur choisi le quatrième mode, mode Simuler c.à.d. la simulation offline, le système_Mandonne la mainà'utilisateur pour qu'il saisisseles valeurs des axes. Elles seront transmises au simulateur.Si les valeurs saisies sont différentes de zéro le bras s'articule (articulation), sinon il garde la position initiale.

La figure [4.13] qui représente l'environnement du simulateur, et les deux interfaces du terminal, lapremière dubas (système_Man),pour choisir le mode et saisir les valeurs des axes. Ladeuxième duhaut articule le bras du Robot.



Figure [4.13] : Mode Simuler (Pos1).

4.3. Temps d'exécution :

Le tableau ci-dessus qui représente le temps d'exécution pour chaque mode.

| Mode | Description | Temps d'exécution |
|-----------------|--|-------------------|
| Mode manuel | gère les flux de données entre le pc embarqué et le pc distant. | 2300ms |
| Mode simulation | Partager les données de simulateur (vitesses, et positions, articulation) vers le pc embarqué. | 3000ms |
| Mode simulé | Partager les valeurs des axes de bras après le saisi par l'utilisateur vers le simulateur | Sim<1s |

Implémentation

Tableau [4.1] : Temps d'exécution pour chaque Mode.

5. Conclusion :

Après avoir achevé notre conception nous avons donné les outils nécessaires pour la réalisation de notre travail. Nous avons présenté aussi l'environnement de développement. A la fin nous avons présenté l'état de l'existant, On a donné quelques captures

Conclusion générale

Conclusion générale

Le projet que nous avons présenté dans ce mémoire consiste à faire une connexion bilatérale entre un simulateur et un robot manipulateur mobile.

L'objectif de notre travail était la réalisation d'une communication bidirectionnelle entre le simulateur 3D développé par gazebo et robot manipulateur mobile RobuTER/ULM en temps réelle. Si le robot réel déplace le simulateur déplace aussi et vice versa. Ensuite, nous avons proposé trois modes de contrôle :

- Mode simulation Off_ligne, l'utilisateur ne contrôle que le simulateur.
- Mode simulation On_ligne, le simulateur qui commande le robot réel (Roboter/ULM).
- Mode Manipulation, le robot réel RobuTER/ULM qui commande le simulateur.

Durant l'implémentation de notre travail, nous avons rencontré plusieurs problèmes des difficultés à apprendre ROS, contrôle le bras de simulateur.

Les principaux buts visés ont été atteints, et la majorité de besoins ont satisfaits.

Références Bibliographiques et Webographie

- [1] LAOUICI, Z. (2013). La Robotique & les différentes méthodes de navigation pour un robot mobile, mémoire de master, université d'Oran.
- [2] BRAHIM BOUNEB, A. SIDI MAMAR, N. (2016) contribution au développement d'une stratégie de génération de mouvement pour un robot manipulateur mobile, Mémoire master, université Blida 1, Blida, Algérie.
- [3] Akli, I. (2007). Elaboration d'une stratégie de coordination de mouvements pour un manipulateur mobile redondant, Thèse de doctorat.
- [4] Niku, S. B. (2001). Introduction to robotics: analysis, systems, applications (Vol. 7). New Jersey: Prentice Hall.
- [5] LENS, S. (2008) Locomotion d'un robot mobile. Thèse de doctorat. Université de Liège, Liège, Belgique.
- [6] Robot mobile. Consulté le 20 mars 2017 de http://www.icbanq.com/data/ICBShop/shop_img_m/djsf_m.jpg
- [7] GUECHI, E. (2010) Suivi de trajectoires d'un robot mobile non holonome : approche par modèle flou de Takagi-Sugeno et prise en compte des retards, mémoire doctorat, université Annaba, Annaba.
- [8] Dailymail. (2015). Robot unicycle Consulté le 5 mars 2017 de http://i.dailymail.co.uk/i/pix/2015/08/21/11/2B8BA29400000578-0-image-m-35_1440152706175.jpg
- [9] Lens, S. (2008). Locomotion d'un robot mobile (Doctoral dissertation, Université de Liège, Liège, Belgique).
- [10] Robot tricycle, Consulté le 9 mars 2017 de http://cse17-iiith.virtual-labs.ac.in/forwardkinematics/FK_images/img10.jpg
- [11] Robot voiture, Consulté le 9 mars 2017 de <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSQOb9ytIYPafGYtgfWXgtduyGf0ockNoaf2z1HwXjXXXioqpw>
- [12] Bayle, B. (2008). Robotique mobile. Ecole Nationale Supérieure de Physique de Strasbourg, University Louis Pasteur, France, 2007-2008.

Références Bibliographiques et Webographie

- [13] Robot omnidirectionnel Consulté le 10 mars 2017 de <http://www.robotshop.com/media/catalog/product/cache/14/image/900x900/9df78eab33525d08d6e5fb8d27136e95/k/i/kit-robot-triangulaire-omnidirectionnelles-100mm.jpg>
- [14] Gourdeau, R. (2014). cours modélisation des robots manipulateurs.
- [15] Bras manipulateur. Consulté le 10 mars 2017 de https://encryptedtbn0.gstatic.com/images?q=tbn:ANd9GcSHFA_1jtzw9zjymesNIbYG SJ1fXirZRB1Y4GW6PXWo12g_AleP
- [16] Belloufi, A. Cour robotique master académique .Génie mécanique, Université Ouargla.
- [17] Robot cylindrique. Consulté le 20 avril 2017 de <http://slideplayer.fr/1694810/6/images/10/Robot+Cylindrique.jpg>
- [18] Takhi, H., & Attachi, R. C. (2015). Conception et réalisation d'un robot mobile à base d'Arduino. Mémoire de master, Université Amar Telidji- Laghouat.
- [19] Robot rectiligne, Consulté le 20 avril 2017 de <http://www.memoireonline.com/01/16/9368/Conception-et-realisation-d-un-robot-mobile--base-d-arduino18.png>
- [20] Robot sphérique, Consulté le 20 avril 2017 de <http://slideplayer.fr/1694810/6/images/18/Robot+Sph%C3%A9rique.jpg>
- [21] Computer Aided Manufacturing , C. Elanchezhian, G. ShanmugaSundar, First Edition 2005, Second Edition 2007.
- [22] Robot articulé, Consulté le 20 avril 2017 de <http://www.zesmallfactory.com/wp-content/uploads/2014/11/robot-articule-impression3d-zortrax-m200.jpg>
- [23] Mathia, K. (2010). Robotics for electronics manufacturing: principles and applications in cleanroom automation. Cambridge University Press.
- [24] Robot SCARA quatre axes. Consulté de 23 avril 2017 de <http://www.servovision.com/Used%20&%20New%20Robot%20-%20Robo%20Work%20In%20Robot/Scara%20Robot%204%20Axes/%E0%B9%81%>

Références Bibliographiques et Webographie

[E0%B8%95%E0%B9%88%E0%B8%87%E0%B9%81%E0%B8%A5%E0%B9%89%E0%B8%A71.jpg](#)

- [25] Calomiris, C. W. (1990). Is deposit insurance necessary? A historical perspective. *The Journal of Economic History*, 50(2), 283-295. De <http://www.mobilemanipulation.org/index.php/about>
- [26] El metennani, M. Oueld taleb, M. (2017). Planification automatique et hiérarchique de tâches pour les robots manipulateurs mobiles, mémoire master, Université Saad dahlebblida 1.
- [27] Manipulateur mobile. Consulté le 12 juin 2017 de <http://www-sop.inria.fr/icare/robea/images/h2bis2.gif>
- [28] Neobotix. (2017). Robotic arm. Consulté le 04 septembre 2017 de <http://www.neobotix-robots.com/mobile-manipulators-overview.html>
- [29] Khiter, B., Hentout, A., Boutellaa, E., Benbouali, M. R., & Bouzouia, B. (2012, October). Internet-based telerobotics of mobile manipulators: application on RobuTER/ULM. In *International Conference on Intelligent Robotics and Applications* (pp. 635-644). Springer, Berlin, Heidelberg.
- [30] RobuTER/ULM. Consulté le 28 août 2017 de https://www.researchgate.net/publication/262238308_InternetBased_Telerobotics_of_Mobile_Manipulators_Application_on_RobuTERULM
- [31] BENMISRA, A. (2007). Programmation des robots industriels et application sur le robot manipulateur Algérie machines outil 1 mémoire Magistère en Génie Mécanique Université de Saad Dahleb de Blida, Algérie.
- [32] Hantout, A. (2011), supervision et control d'un robot manipulateur mobile, thèse de doctorat, université des sciences et de la technologie houari boumediene, Alger, ALGERIE.
- [33] Padois, V. (2005). Enchaînements dynamiques de tâches pour des manipulateurs mobiles à roues, Doctoral dissertation, Institut National Polytechnique de Toulouse-INPT, France.
- [34] Staritz, P. J., Skaff, S., Urmson, C., & Whittaker, W. (2001). Skyworker: a robot for assembly, inspection and maintenance of large scale orbital facilities. In *Robotics*

Références Bibliographiques et Webographie

and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on (Vol. 4, pp. 4180-4185). IEEE.

[35] AMMARI, N. MADOU, R. (2016) développement Simulateur 3D pour système Multi_Robot hétérogènes, Mémoire Master informatique, université Blida 1, Blida, Algérie.

[36] Consulté le 25 avril 2017 de <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTTYmYT5oQ7QzuQ73tnLrjwW7kQ-YzhNVGwIzXE9da8qzDqs1y>

[37] Herrera-Aguilar, I. (2007). Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service, Doctoral dissertation, Université Toulouse III-Paul Sabatier.

[38] Takhi, H., & Attachi, R. C. (2015). Conception et réalisation d'un robot mobile à base d'Arduino. Mémoire de master, Université Amar Telidji- Laghouat.

[39] Robot Domestique. Consulté le 25 avril 2017 de <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS0yvXbdoO6tN41NMXZhfsc5-H8vqTho9nEIBtusRJdHL3j7XP->

[40] Waarsing, B. J. W. Nuttin, M., & Van Brussel, H. (2003). Behaviour-based mobile manipulation: the opening of a door. In Proc. of the Int. Workshop on Advances in Service Robotics (pp. 168-175), Bardolin, Italy.

[41] Robot ouverture de porte. Consulté le 10 juin 2017 de https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQkc2oCXW1zYRcmI0dG2PLC6K7c9mPBc1QGDOvnW-0y7_y4mVVa

[42] Zaffagni, M. (2016). Sophia, le robot qui veut détruire l'humanité. Consulté le 10 mai 2017 de <http://www.futura-sciences.com/tech/actualites/robotique-sophia-robot-veut-detruire-humanite-62142/>

[43] Robot saisie d'objet. Consulté le 10 juin 2017 de https://lejournal.cnrs.fr/sites/default/files/styles/visuel_principal/public/assets/images/2014n00024h_72dpi.jpg

Références Bibliographiques et Webographie

- [44]: Gharbi, M. (2011). Bibliographie pour la réalisation de tâches interactives homme-robot. Mémoire master informatique, Ecole Nationale d'Ingénieurs de Brest, Plouzané.
- [45]Khatib, O., Brock, O., Chang, K. S., Conti, F., Ruspini, D., & Sentis, L. (2002). Robotics and interactive simulation. *Communications of the ACM*, 45(3), 46-51.
- [46] : Erard, P. J. (1999). Simulation par évènement discrets. Presses Polytechniques et Universitaires Romandes (PPUR).
- [47] M. S. Couceiro, D. Portugal and R. P. Rocha, "A Collective Robotic Architecture in Search and Rescue Scenarios," In Proc. of 28th ACM Symposium on Applied Computing (ASC 2013), Coimbra, Portugal, pp. 64-69, Mar. 18-22, 2013.
- [48] T. Choi, H. Do, C. Park, D. Park, S. Lee and J. Kyung, "Software Platform for the Industrial Dual-Arm Robot," *Robot Intelligence Technol. & Appl.*, pp. 911-920, 2012.
- [49] M. Freese, S. Singh, F. Ozaki and N. Matsushira, "Virtual Robot Experimentation Platform V-REP: A versatile 3D Robot Simulator," *SIMPAR*, pp. 51-62, 2010.
- [50]B. P. Gerkey, R. T. Vaughan and A. Howard, "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems," *Proceedings of the International Conference on Advanced Robotics (ICAR 2003)*, pp. 317-323, 2003.
- [51] L. Hugues and N. Bredeche, "Simbad: An Autonomous Robot Simulation Package for Education and Research," *SAB*, pp. 831-842, 2006.
- [52] N. Koenig and A. Howard, "Design and Use Paradigms for Gazebo, An Open-source Multi-Robot Simulator," in *International Conference on Intelligent Robots and Systems, Sendai, Japan, 2004*.
- [53]: Gonçalo, N. S. A. (2013). RobotTeamSim – 3D Visualization of Cooperative Mobile Robot Missions in Gazebo Virtual Environment. Type de document inédit, Université de Coimbra, Portugal.
- [54] : Chen, I. (2015). Model Editor. Consulté le 23 juillet 2017 de <http://gazebosim.org/tutorials?tut=modeleditor#Editjoints>

Références Bibliographiques et Webographie

[55] Rahiche, A., Bouzouia, B., & CDTA, B. H. Fusion Multi-sensorielles et Téléopération: Application à un Robot Mobile Manipulateur.

[56] <http://www.blogxml.net/.../02/21/11-introduction-a-php>

[57] N. HADJ KADDOUR, 'Etude et réalisation d'une station expérimentale de téléopération d'un bras manipulateur par réseau 'Mémoire de Magister' Université des sciences et de la technologie d'Oran" Mohammed Boudiaf" ', 'Oran, 2004'.

[58] K. BENAHMED, 'Etude et réalisation d'un système de calibration pour la téléopération', 'Mémoire de Magister 'Université des sciences et de la technologie d'Oran" Mohammed Boudiaf" ', 'Oran, 2007'.

[59] Ammari, N. Madaoui, R. (2016). Développement d'un simulateur 3D pour système Multi-Robots hétérogènes. Mémoire de master, Université Saad dahleb blida 1.

[60] Mourani, G. (2000). Securing and optimizing Linux: RedHat edition. Open Network Architecture and OpenDocs Publishing.

[61] Gayan. (2014). Consulté le 2 septembre 2017 de <https://www.hecticgeek.com/2014/04/ubuntu-14-04-review-stable-responsive-polished/>

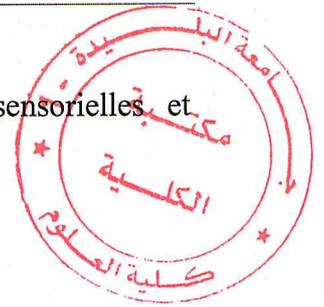
[62] Marguin, O. (2004). Cours d'informatique, C++ : LES BASES.

[63] C++. Consulté le 5 septembre 2017 de <http://www.lemauricien.com/sites/default/files/imagecache/400xY/article/2013/11/18/c.jpg>

[64] Python. Consulté le 4 septembre 2017 de <https://i.ytimg.com/vi/Z7soD7Yj9uw/hqdefault.jpg>

[65] Ros. Consulté le 5 juin 2017 de <https://113kf73dl0pg27adzc3ob3r9-wpengine.netdna-ssl.com/wp-content/uploads/2015/03/Introducing-the-New-ROS-Integrated-Segway-RMP-Line1-731x265.jpg>

[66] GénérationRobots. ROS - Robot Operating System. Consulté le 5 juin 2017 de https://www.generationrobots.com/fr/content/55-ros-robot-operating-system?code_lg=lg_fr



Références Bibliographiques et Webographie

[67]Outil de ros. Consulté le 5 juin 2017 de
http://www.ros.org/news/2014/04/22/indigoigloo_600.png

