



Université ABBES LAGHROUR Khenchela
Faculté des Sciences et de la Technologie
Département de Génie Industriel
جامعة عباس لغزور خنشلة
كلية العلوم والتكنولوجيا
قسم الهندسة الصناعية



N° Série :

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Automatique

Spécialité : Automatique et informatique industrielle

THEME

**Etude et Simulation d'un Robot
Mobile (Khepera 4)**

Réalisé par : - ZALAGH Abdelfateh

- AGGOUN Imed Eddine

Soutenu le 15 / 09 /2020 Devant le jury composé de:

Mr. A.SAIDI
Mr. A.BERKANI
Mr. A.MALLEM

Président
Encadreur
Examineur

Université Abbes Laghrour-Khenchela
Université Abbes Laghrour-Khenchela
Université Abbes Laghrour-Khenchela

Promotion 2019/2020

الملخص:

تم تصميم الروبوتات المتنقلة المستقلة خصيصًا لتصميم الأنظمة التي يمكنها التحرك بشكل مستقل. وتشمل التطبيقات المباشرة السيارات واستكشاف الكواكب وروبوتات الخدمة، على سبيل المثال.

والموضوع الذي تم تناوله في هذه الأطروحة هو محاكاة لطريقة عمل و تنقل الروبوت الصغير (khepera4) وتجاوزه للعقبات التي تصادفه عن طريق التنسيق بين مكوناته الكهربائية و بعض الانظمة التي تسيير هذه المكونات.

الكلمات المفتاحية:

simulation ، V-REP ، robot mobile ، robot ، kheperaVI ، khepera

Resumé :

Les robots mobiles autonomes sont spécifiquement conçus pour concevoir des systèmes capables de se déplacer indépendamment. Les applications en direct incluent les voitures, l'exploration planétaire et la robotique de service, par exemple.

Le thème abordé dans cette mémoire est une simulation de la façon dont le petit robot (khepera4) fonctionne et se déplace et surmonte les obstacles rencontrés en coordonnant ses composants électriques avec certains des systèmes qui exécutent ces composants.

Mots clés :

khepera, kheperaVI, robot, robot mobile , V-REP ,simulation



Remerciment

Je remercie d'abord **ALLAH** le tout puissant de me donner la patience, la puissance et la volonté pour réaliser mon travail.

Je voudrais remercier mon encadreur monsieur **Berkani AbdElhakim** d'avoir accepté de diriger mon travail et pour toutes les informations, les conseils qu'il m'a donné et tout ses efforts qu'il a consenti durant la réalisation de ce travail.

Je voudrais remercier tous les enseignants de l'université Abbes LAGHROR –Khenchela pour leur contribution à notre formation.

Mes plus remerciements vont aussi aux membres de jury pour l'intérêt qu'ils ont porté à mon travail.

MERCI

A tous ceux qui m'ont aidé de près ou de loin dans l'élaboration et la finalisation de ce travail.



Dédicace

Je dédie ce travail à, Ma mère et mon père, a habite le plus hauts paradis ceux à qui je doit tant , pour leur amour et leur support continu tout le long du cursus de mes études. Que ce travail soit le témoignage sincère et affectueux de ma profonde reconnaissance pour tout ce que vous avez fait pour moi. Ils m'ont soutenu au prix de sacrifices inoubliables.

Mes frères et mes sœurs qu'i m'encourager et soutenu des mes moments les plus difficiles.



SOMMAIRE

Introduction générale.....	I
Chapitre I. Généralités sur les robots	
I.1. Introduction.....	15
I.2. Notions de base	15
I.2.1. Domaine robotique.....	15
I.2.2. Automate	15
I.2.3. Robot.....	16
I.2.4. Intelligence artificielle	16
I.3. Historique.....	16
I.3.1. Automates (1ère génération)	16
I.3.2. Robots (2ème génération)	17
I.3.3. Robots dotés d'IA (3ème génération).....	17
I.4. Disciplines de la robotique.....	17
I.5. Classification des robots	18
I.5.1. Manipulateurs.....	18
I.5.2. Télémanipulateurs.....	19
I.6. Robots mobiles.....	20
I.6.1. Concepts sur la robotique mobile	20
I.6.2. Les nouvelles approches de la robotique mobile	26
I.6.3. L'autonomie d'un robot	28

I.6.4. Environnement dynamique et incertain.....	31
I.6.5. Problématiques de la robotique mobile.....	31
I.6.6. Architectures de contrôle.....	33
I.7. Conclusion.....	37

Chapitre II. Description technique du robot mobile Khepera IV

II.1. Introduction.....	39
II.2. Historique sur les robots Khepera	39
II.3. Généralité sur le robot Khepera IV	41
II.3.1. Définition.....	41
II.3.2. Description physique.....	41
II.4. Fonctionnalités principales du robot mobile Khepera IV.....	42
II.5. Caractéristiques techniques du robot Khepera IV	42
II.6. Les différents composants du robot mobile Khepera IV.....	43
II.6.1. Vue globale du robot	44
II.6.2. Vue détaillé des composantes du robot	46
II.7. Moyenne d'extension	54
II.7.1. La tourelle d'extension.....	55
II.7.2. Télémètre laser	55
II.7.3. Le préhenseur	56
II.7.4. Système de positionnement global intérieur (Stargazer).....	56
II.8. Conclusion	57

Chapitre III. Etude et Simulation d'un Robot Mobile (Khepera 4)

III.1. Introduction	59
III.2. Simulation Des Robots Mobiles	60
III.3. Comparaison des simulateurs	60
III.4. V-REP	61
III.5 modèle Khepera IV V-REP	63
III.5.1 Khepera IV	63
III.5.2. Conception graphique des éléments du modèle	63
III.5.3. Bibliothèque Khepera IV en V-REP	64
III.5.4 Exemple simple avec le modèle (commandes de vitesse)	65
III.5.5. Contrôle de position ou « stabilisation des points »	67
III.5.6. Expérience de contrôle de position	68
III.5.7. Expérience d'évitement des obstacles	71
III.5.8. Suivi du chemin d'accès	74
III.6- CONCLUSION	75
Conclusion Général	76
Webgraphie	77
Bibliographie	78

Liste des figures

- Figure 1 : Automate d'Hugo Cabret.....	15
- Figure 2 : Constitution d'un robot manipulateur.....	18
- Figure 3 : Parties constituant un RM.....	19
- Figure 4 Les interactions d'un robot avec son environnement	21
- Figure 5 Différents types de robots mobiles selon le champ d'application.	22
- Figure 6 : Structure d'un robot mobile.....	23
- Figure 7 . Exemple de plateforme unicycle.....	24
- Figure 8 : Exemple de plate-forme holonome.....	25
- Figure 9 : Exemple de plate forme non holonome	25
- Figure 10 : Exemple de plate forme omnidirectionnelle ROMNI	26
- Figure 11 : Le robot Shakey de Stanford en	26
- Figure 12 : Boucle de contrôle pour la commande d'un robot mobile	33
- Figure 13 : Architecture de contrôle cognitive.....	34
- Figure 14 : Architecture de contrôle réactive.....	34
- Figure 15 : Contrôle cognitif versus réactif	35
- Figure 16 : Architecture de contrôle hybride	36
- Figure 17 : Le premier robot Khepera publié en 1996.....	39
- Figure 18 : Khepera II	40
- Figure 19 : Khepera III.....	40
- Figure 20 : Khepera IV.....	40
- Figure 21 : Vue du bas, du haut, de l'avant et de la gauche du robot	42
- Figure 22 : Vue de gauche.....	44
- Figure 23 : Vue arrière	44

- Figure 24 : Vue de droite.....	44
- Figure 25 : Vue de dessous	44
- Figure 26 : Vue de dessus	45
- Figure 27 : Capteurs à infrarouges vus de fond	47
- Figure 28 : Valeur IR par rapport à la distance	48
- Figure 29 : Capteurs ultrasoniques vus de sommet.....	49
- Figure 30 : Positions de microphones	51
- Figure 31 : Les directions d'accélération détectables	52
- Figure 32 : Les directions de taux angulaires détectables	53
- Figure 33 : Bloc du moteur avec roue.....	54
- Figure 34 : La tourelle d'extension.....	55
- Figure 35 : Extension télémètre laser	55
- Figure 36 : Extension avec le préhenseur.....	56
- Figure 37 : Le Stargazer.....	56
- Figure 38 : Vue V-REP IDE	62
- Figure 39 : Robot à roues Khepera IV	63
- Figure 40 : Composants du modèle Khepera IV.....	64
- Figure 41 : Exemple d'utilisation de la bibliothèque.....	66
- Figure 42 : Schéma bloc d'un asservissement de contrôle de position.....	69
- Figure 43 : Expérience de contrôle de position.....	69
- Figure 44 : Trajectoire de mouvement.....	70
- Figure 45 : Schéma bloc de contrôle avec évitement d'obstacle	71
- Figure 46 : Expérience d'évitement des obstacles	73
- Figure 47 : Expérience de Suivi du chemin d'accès	74

Liste des abréviations

IA	Intelligence Artificielle
OT	Organe Terminal (outil)
SMA	Système Mécanique Articulé
RM	Robot Manipulateur
CAO	Conception Assistée par Ordinateur
I2C	Inter Integrated Circuit
IMU	Unité de Mesure Inertielle
CC	Courant Continue
IR	Infra-Rouge
LRF	Télémètre laser (Laser Range Finder)
FPS	Cadre Par Seconde (Framework Per Second)

Introduction

Générale

Introduction Générale

Un robot mobile autonome est une machine qui a besoin, pour sa mise en œuvre, de spécialistes dans de nombreux domaines dont la mécanique, l'automatique, l'électronique, l'électrotechnique, l'informatique, les mathématiques...etc, C'est un carrefour des technologies et des méthodologies. Des progrès doivent encore être réalisés dans chacune de ces disciplines afin de rendre la machine plus adaptative à l'environnement. Des progrès dans le domaine de la perception sont encore nécessaires pour une meilleure intégration dans l'espace d'évolution. Malgré ces réserves, il est inéluctable que les développements et les applications des robots mobiles autonomes, totalement ou partiellement, iront en s'amplifiant.

La robotique mobile autonome vise plus spécifiquement à concevoir des systèmes capables de se déplacer de façon autonome. Les applications directes se situent notamment dans les domaines de l'automobile, de l'exploration planétaire ou de la robotique de service par exemple. [I]

De nombreuses applications restent à découvrir, qui ne découlent pas directement des avancées de la robotique mais qui utilisent ses méthodes et ses développements.

Actuellement, l'intérêt pour les systèmes de robots mobiles augmente en raison de leur utilisation dans diverses missions telles que les missions spatiales, les opérations dans des environnements dangereux et des applications militaires.

Le thème de cette mémoire est une simulation du fonctionnement d'un robot mobile, programmé pour l'automobilité et effectuant certaines tâches en prenant des photos et des vidéos, en enregistrant des sons, en surmontant certains obstacles.

Le nom de ce robot simple (khepera 4).

Le robot Khepera IV est la 4ème version du robot mobile Khepera pour l'éducation et la recherche. Ce robot mobile compact a été conçu pour des surfaces planes (sol du laboratoire, table...). Il embarque ce qui se fait de mieux en terme de capteurs et moteurs de précision.

Cette nouvelle version du robot Khepera apporte de nombreuses nouveautés : OS Linux, caméra couleur, Wifi, Bluetooth, Host USB, accéléromètre, gyroscope et une odométrie améliorée et plus précise. [II]

Introduction Générale

Structure de la mémoire

L'organisation générale de cette mémoire est décomposée en cinq chapitres :

- **Chapitre I** : Ce chapitre est un aperçu du monde de la robotique, en particulier les robots mobiles, où nous avons discuté des types et des catégories de robots mobiles et de certaines de leurs utilisations.
- **Chapitre II** : Ce chapitre est une explication détaillée du robot mobile khepera des caractéristiques ainsi qu'un aperçu historique de ses générations
- **Chapitre III** : Ce chapitre est une simulation du robot khepera 4 par V-REP pour voir certaines des tâches et des activités que ce robot peut effectuer en le programmant dans les langages de programmation.

- En fin, nous terminerons notre travail par une **Conclusion Générale** .

Chapitre I

I.1. Introduction

L'utilisation des robots est aujourd'hui couramment envisagée pour l'automatisation de nombreuses tâches. Celles-ci sont particulièrement diversifiées : le nettoyage, le transport dans les ateliers automatisés, l'agriculture, l'exploitation des mines, l'assistance aux personnes handicapées et l'exploration de milieux hostiles en sont quelques exemples.

On va entamer ce chapitre par un aperçu historique sur les robots, leurs caractéristiques, et leurs domaines d'application, ainsi que les robots mobiles.

I.2. Notions de base

Dans cette partie nous allons introduire quelque notion concernons notre sujet, on peut citer les suivants :

I.2.1. Domaine robotique

Ensemble des études et des techniques de conception et de mise en œuvre des robots effectuant des tâches déterminées en s'adaptant à leur environnement [1]. La robotique vient de l'anglais robotiques, imaginé par le romancier « Isaac Asimov » et popularisé par un livre publié en 1942, (**Runaround**) [2].

I.2.2. Automate

Machine qui par le moyen de dispositifs mécaniques, pneumatiques, hydrauliques, électriques ou électroniques, est capable d'actes imitant ceux des corps animés [3]. Structure physique ou informationnelle qui fonctionne d'après des règles strictes, heuristiques ou probabilistes, sans l'intervention consciente de l'être humain [4]. La figure I.1 illustre un exemple automate.



Figure I.1 : Automate d'Hugo Cabret [5]

I.2.3. Robot

Un robot est un automate doté de capteurs et d'effecteurs lui donnant une capacité d'adaptation et d'emplacement proche de l'autonomie. Un robot est un agent physique réalisant des tâches dans l'environnement dans lequel il évolue [6].

I.2.4. Intelligence artificielle

Discipline scientifique relative au traitement des connaissances et au raisonnement, dans le but de permettre à une machine d'exécuter des fonctions normalement associées à l'imitation de la sortie humaine : Compréhension, raisonnement, dialogue, adaptation, apprentissage, etc. [7]

I.3. Historique [8]

On peut schématiquement distinguer trois principales ères en robotique :

les automates, les robots n'étant pas dotés d'intelligence artificielle (IA), et ceux disposant d'une IA.

I.3.1. Automates (1^{ère} génération)

Un automate contrairement à un robot (même s'il ne dispose pas d'IA), obéit uniquement à un programme préétabli, que ce soit de manière mécanique ou électronique. De ce fait, il n'y a aucune adaptation possible entre l'automate et son environnement.

On attribue la paternité du tout premier automate de l'humanité à Architos de Tarente (4^{ème} siècle avant J.C.). Il s'agissait d'une représentation d'un pigeon capable de voler et étant propulsé par de la vapeur. Malheureusement, aucun vestige ni aucun schéma ou représentation fidèle n'ont été retrouvés.

Les toutes premières traces d'automates remontent à l'antiquité par Héron d'Alexandrie au 1^{er} siècle après J.C., ses réalisations ornèrent les temples et les théâtres de la ville égyptienne.

En 1495 Léonard de Vinci présenta un chevalier humanoïde capable de s'asseoir, de relever sa visière et de bouger ses bras.

L'automate le plus célèbre est le canard mécanique de Jacques De Vaucanson, capable d'ingurgiter de la nourriture et de la digérer tout en se déplaçant. De Vaucanson aurait également présenté en 1738 un second automate représentant un homme jouant d'un instrument à vent.

I.3.2. Robots (2^{ème} génération)

Les robots de 2^{ème} génération disposent d'organe(s) sensoriel(s), autrement dit des capteurs, pouvant influencer sur leurs comportements. Ils sont donc relativement adaptables à leur environnement.

Le chien électrique de Hammond et Miessner (1915) est le premier robot de ce genre. Il se déplace selon la luminosité de l'endroit grâce à son capteur optique.

Le chien Phillidog de Henri Piraux en 1928, et le renard de Ducrocq (1953) fonctionnent selon le même principe.

Walter Grey équipe en 1950 sa tortue cybernétique de capteurs tactiles et lumineux.

I.3.3. Robots dotés d'IA (3^{ème} génération)

Pour être intégré dans cette famille, un robot doit parvenir à effectuer une tâche par lui-même, sans aucune aide extérieure.

En 1973, l'université de Waseda présente le tout premier humanoïde "intelligent" dénommé Wabot-1. Il est doué de la vision, peut manipuler des objets, effectue un semblant de marche et est même capable de débiter une conversation en japonais.

Hi-T-Hand d'Hitachi en 1974 manipule des aiguilles à travers des trous grâce à la détection de force.

Vers la fin des années 70 Hans Morava présente les premiers robots capables d'évoluer à l'extérieur, de façon autonome.

I.4. Disciplines de la robotique

- **Mécanique**

Conception, réalisation, modélisation des robots.

- **Electronique**

Mise en place de composants pour les robots, télécommunications, communications, commande, acquisition des données.

- **Informatique**

Gestion des fonctionnalités des robots et de transfert d'information.

- **Automatique**

Commande, calibrage des capteurs, des effecteurs, identification des paramètres.

- **Traitement du signal**

Analyse des informations enregistrées par les capteurs du robot.

- **Mathématiques**

Modèles mathématiques pour la prise de décision ou/et l'apprentissage, calcul de trajectoires, localisation, planification.

- **Sciences cognitives**

Interactions homme-machine, machine-machine, prise de décision.

I.5. Classification des robots

On peut les classés en trois catégories :

- **Manipulateurs [9]**

Ce sont des robots industriels destinés à effectuer des tâches bien déterminées et de manière répétitive. A noter que pour ce type de robot :

Les trajectoires sont bien déterminées dans l'espace ; Les positions sont discrètes avec 2 ou 3 valeurs par axe ; La commande est séquentielle.

La figure suivante illustre les différents éléments qui constituent un robot manipulateur :

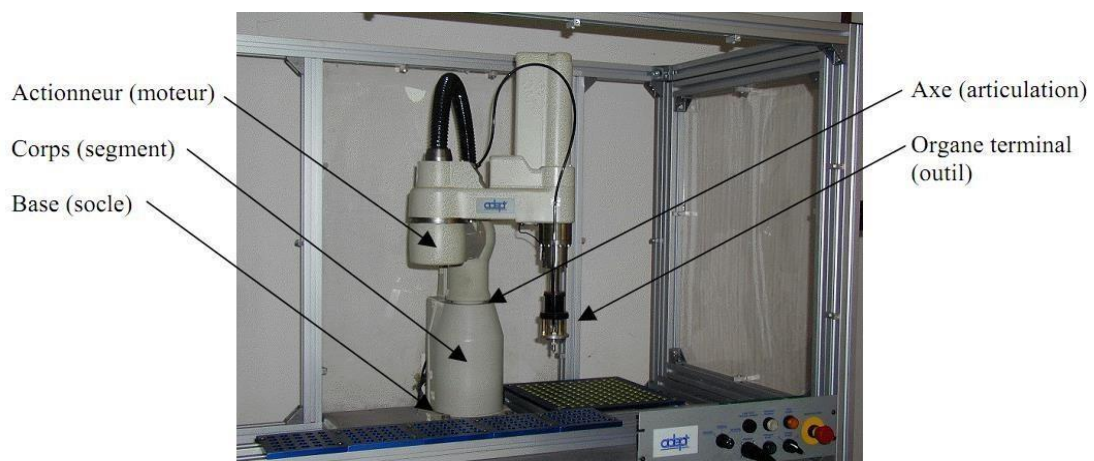


Figure I.2 : Constitution d'un robot manipulateur

Sous le terme organe terminal (OT), on regroupe tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression, etc.), ou à les transformer (outils, torche de soudage, pistolet de peinture, etc.). En d'autres termes, il s'agit

d'une interface permettant au robot d'interagir avec son environnement.

Un OT peut être multifonctionnel, au sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable. Un robot, enfin, peut-être multi-bras, chacun des bras portant un OT différent.

Le système mécanique articulé (SMA) est un mécanisme ayant une structure plus ou moins proche de celle du bras humain. Il permet de remplacer, ou de prolonger, son action.

Son rôle est d'amener OT dans une situation (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données.

Précisons la notion d'articulation : Une articulation lie deux corps successifs en limitant le nombre de degré de liberté de l'un par rapport à l'autre.

Ces différents éléments du robot manipulateur (RM) peuvent être regroupés en quatre parties principales comme indiqué sur la (Figure I.3).

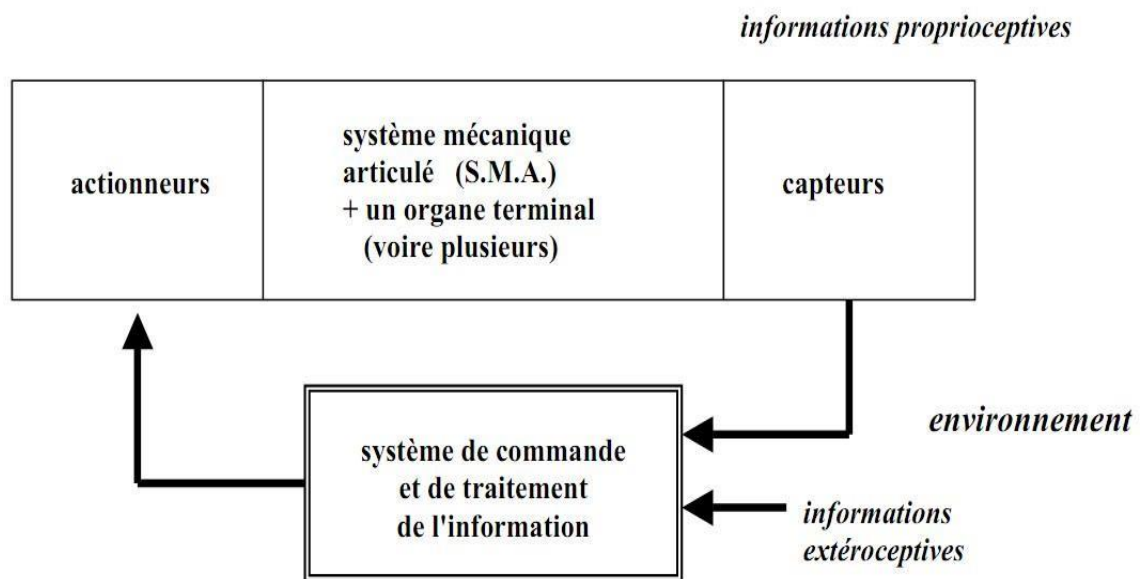


Figure I.3 : Parties constituant un RM

- **Télemanipulateurs [10]**

Appareils de manipulation à distance (pelle mécanique, pont roulant, etc.), apparus vers 1945 aux USA. Dans ce cas :

- Les trajectoires peuvent être quelconques dans l'espace ;
- Les trajectoires sont définies de manière instantanée par l'opérateur.

I.6. Robots mobiles

La robotique mobile est un domaine dans lequel l'expérience pratique est particulièrement illustratrice et importante pour la compréhension des problèmes. Elle est la branche de l'intelligence artificielle concernée par l'étude de systèmes automatiques capables d'une interaction directe avec le monde physique. Elle est l'ensemble des techniques permettant la conception, la réalisation de machines automatiques ou de robots mobiles. La robotique mobile est un domaine de recherche essentiellement pluridisciplinaire. Sa problématique porte sur la conception et l'étude de fonctions de *perception*, de *décision* et d'*action*, et sur l'*intégration* cohérente de ces fonctions en une machine physique. Cette machine devra mettre en œuvre ses fonctions sensori- motrices et décisionnelles pour réaliser de façon autonome une diversité de tâches dans un environnement dynamique imparfaitement modélisé, pour interagir avec d'autres machines et avec des humains, et pour améliorer ses propres performances par apprentissage.

Ce deuxième chapitre se veut un aperçu des concepts de base de la robotique mobile. Certains points seront mentionnés plus en détail que d'autres, car ils sont plus étroitement liés à notre travail de recherche. Nous présenterons également les problématiques de la robotique mobile en présentant les approches, les solutions apparues dans les dernières années.

I.6.1. Concepts sur la robotique mobile

La mobilité autonome des robots est devenue un sujet de recherche développé par tous les pays industrialisés. Qu'il s'agisse de robots mobiles à pattes, à roues ou même sous-marins et aériens, les applications sont vastes et multiples : robots de services, surveillance, construction, nettoyage, manipulation de charges, automobile intelligente, robots d'intervention, robots d'exploration planétaire ou de fonds marins, satellites, robots militaires, etc. Le marché potentiel de la robotique est considérable, même s'il faut pour cela résoudre des problèmes plus importants et plus fondamentaux que prévus initialement dans la quête vers la machine intelligente [11], [12].

- **Définition d'un robot mobile**

Un robot est une machine chargée d'effectuer une ou plusieurs tâches. Le mot vient d'une pièce de théâtre tchèque dans laquelle apparaissait un travailleur artificiel employé pour le « servage », désigné en tchèque par « *robota* ». L'appellation *Robot mobile* regroupe tous les types de robots qui ont la capacité de déplacement qui est la caractéristique commune entre eux, la différence réside dans la manière, qui dépend du domaine d'utilisation de robot, par laquelle le

robot va atteindre cette faculté de mouvement. La mobilité par les roues est la structure

mécanique la plus communément appliquée. Cette technique assure selon l'agencement et les dimensions des roues un déplacement dans toutes les directions avec une accélération et une vitesse importante.

Afin d'être autonome, un robot doit être équipé 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 qu'il en a et de ses objectifs [13], mais également de savoir comment réagir en conséquence, suivant le niveau d'autonomie. C'est à lui de planifier son parcours et de déterminer avec quels mouvements il va atteindre son objectif. Les recherches dans ce domaine portent principalement d'une part sur la localisation du véhicule autonome et la cartographie de son environnement, d'autre part sur le contrôle et la navigation autonome de tels véhicules (structure de contrôle, stratégies de commande, planification et navigation).

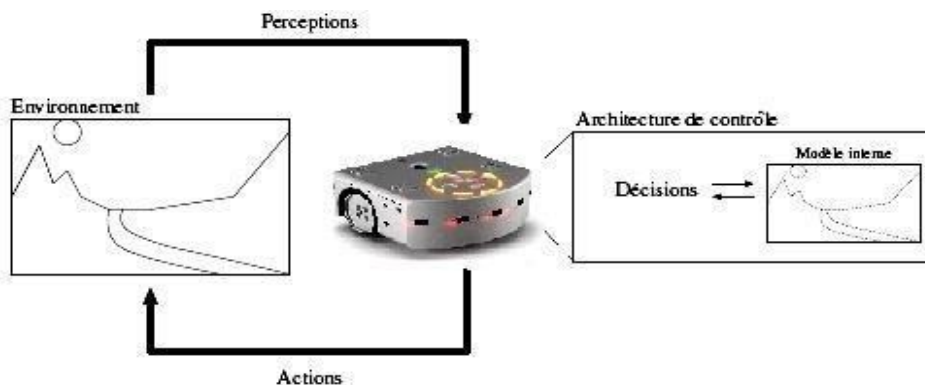


Figure I.4 Les interactions d'un robot avec son environnement.

- **Pourquoi la robotique mobile ?**

Dans un avenir proche, la robotique occupera une place majeure dans notre quotidien. Elle possède de nombreux champs d'applications comme la robotique industrielle ou la robotique de service. Qu'il s'agisse de robot civil ou militaire, il existe désormais des robots capables d'étonnantes prouesses dans de nombreux secteurs : robots-compagnons assistant les personnes à domicile ou en charge de la surveillance et des soins, robots assurant la logistique dans les hôpitaux, robots assistant les industriels dans la réalisation de gestes pénibles et répétitifs, ou encore permettant le développement de prothèses ou d'orthèses intelligentes .



Figure I.5 Différents types de robots mobiles selon le champ d'application

Néanmoins, l'intérêt indéniable de la robotique mobile est d'avoir permettre d'augmenter considérablement nos connaissances sur la localisation et la navigation de systèmes autonomes. La gamme des problèmes potentiellement soulevés par le plus simple des robots mobiles à roues en fait un sujet d'étude à part entière et forme une excellente base pour l'étude de systèmes mobiles plus complexes.

- **.Architecture des robots mobiles**

L'architecture d'un robot mobile s'articule autour de trois modules fondamentaux (Figure I.6) [14], [15] :

- Module locomotion,
- Module perception,
- Module décision

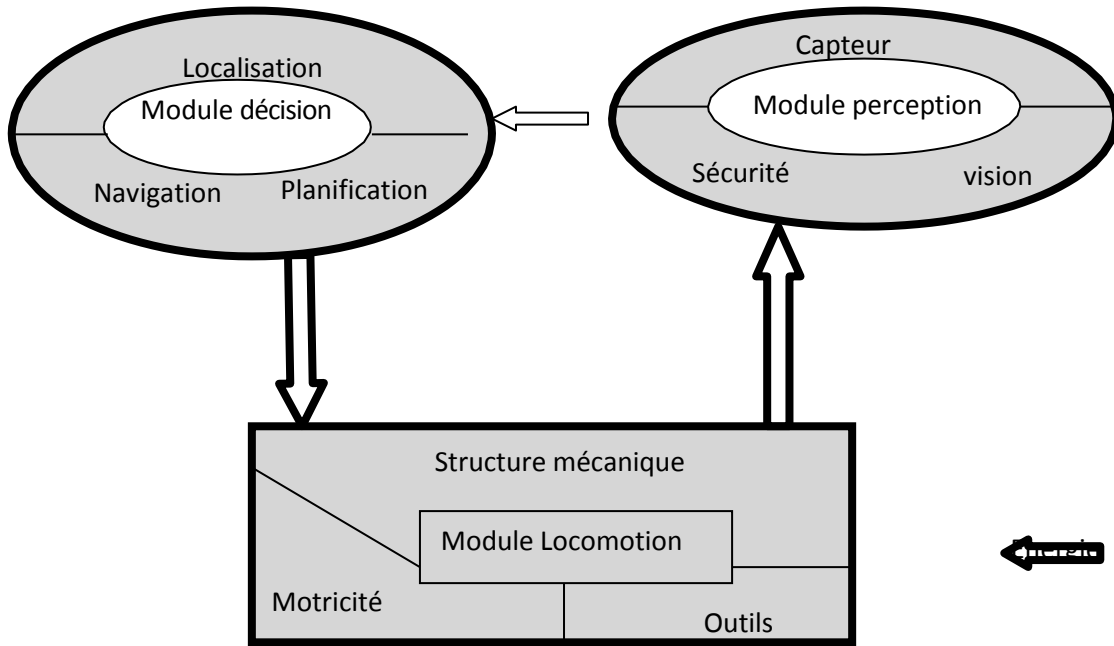


Figure I.6 : Structure d'un robot mobile

- **Module locomotion**

Le module locomotion comporte alors la structure mécanique, motricité et l'énergie utilisée dans le déplacement du robot mobile. Les robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient à roues, marcheurs, sous-marins ou aériens. Différents laboratoires de recherche contribuent jusqu'à présent à la construction des différents robots mobiles selon le type de locomotion.

- **Module perception**

La notion de perception en robotique mobile est relative à la capacité du système à recueillir, traiter et mettre en forme des informations utiles au robot pour agir et réagir dans le monde qui l'entoure. Alors que pour des tâches de manipulation on peut considérer que l'environnement du robot est relativement structuré, ce n'est plus le cas lorsqu'il s'agit de naviguer de manière autonome dans des lieux très partiellement connus. Aussi, pour extraire les informations utiles à l'accomplissement de sa tâche, il est nécessaire que le robot dispose de nombreux capteurs mesurant aussi bien son état interne que l'environnement dans lequel il évolue. Le choix des capteurs dépend bien évidemment de l'application envisagée. Pour se focaliser sur le problème de navigation, nous allons évoquer dans ce chapitre les capteurs utiles à cette tâche.

- **Module Décision**

Les informations en provenance des différents capteurs doivent être interprétées comme autant d'éléments utiles à la prise de décision sur l'action à faire, le but étant de délivrer les ordres corrects aux actionneurs, bras pinces ou moteurs des roues. C'est lors de cette phase de la conception d'un robot qu'il est nécessaire de lui donner une forme d'intelligence en lui laissant le choix sur l'action à entreprendre. Cette prise de décision est souvent arbitraire au début, mais elle permet de développer une forme d'apprentissage qui tient compte des résultats des décisions précédentes.

- **Types de plates formes mobiles**

On peut classer les robots mobiles à roues dans des sous-classes selon le type de la plateforme mobile utilisée :

- **Plates forme unicycle**

C'est l'une des configurations les plus utilisées pour les robots mobiles d'intérieur par ce qu'elle nécessite un sol très plan et non accidenté.

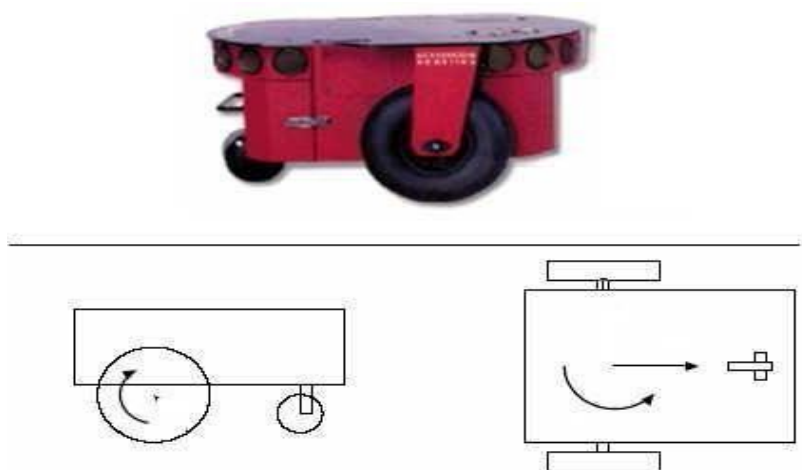


Figure I.7. Exemple de plateforme unicycle [16].

- **Plates formes holonomes**

Les systèmes holonomes sont beaucoup plus rares dans notre vie quotidienne. Ils ont une structure mécanique complexe qui leur permet de se déplacer dans toutes les directions sans manoeuvre.

La société *Nomadic* disparue en l'an 2000 ; a conçu un système mobile holonome : le XR4000 (Figure I.8). Il dispose de 4 roues motrices et directrices montées comme des roues de chariot. La synchronisation des 8 axes (2 par roue, rotation et orientation) est assurée par une carte dédiée basée sur le microcontrôleur Motorola 68332 et des circuits FPGAs et la structure mécanique est composée d'engrenages coniques [17].

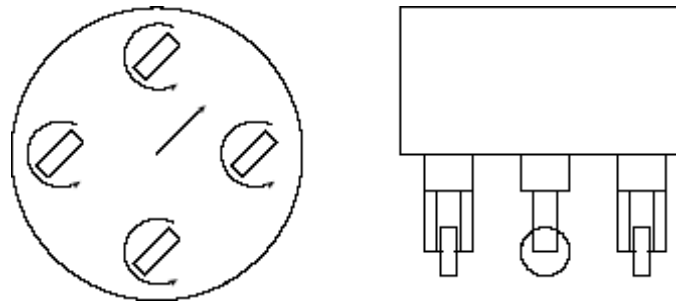


Figure I. 8: Exemple de plate-forme holonome [16].

- **Plates formes non-holonomes**

Les systèmes mobiles dit non-holonomes sont ceux que l'on rencontre le plus dans la vie courante (voiture particulière, bus, camion, ...etc.). Ces systèmes ont une structure mécanique relativement simple (des roues motrices, des roues directrices et des roues libres). Une roue peut avoir une, deux ou trois fonctions. Mais tous ces systèmes ont une caractéristique commune : la direction de la vitesse d'entraînement (vitesse linéaire) est imposée par la direction des roues directrices.

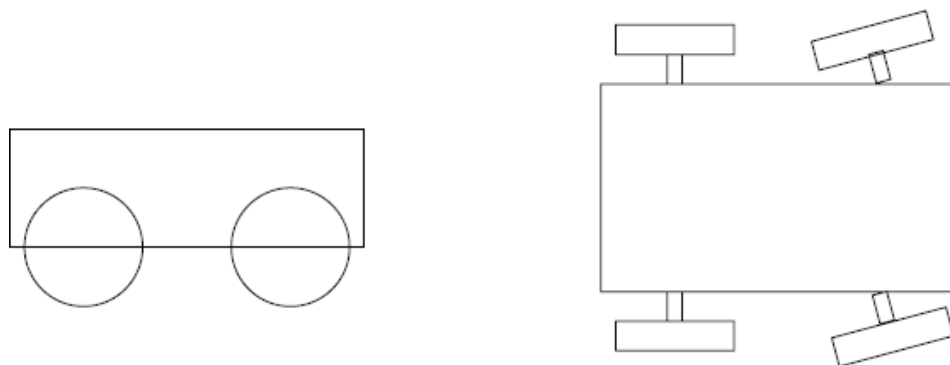


Figure I. 9: Exemple de plate forme non holonome [16].

- **Robot omnidirectionnel**

Un robot omnidirectionnel est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral.

L'énorme avantage du robot omnidirectionnel est qu'il est holonome puisqu'il peut se déplacer dans toutes les directions. Mais ceci se fait au dépend d'une complexité mécanique bien plus grande.



Figure I. 10: Exemple de plate forme omnidirectionnelle ROMNI [18]

I.6.2. Les nouvelles approches de la robotique mobile

En 1983, Hans Moravec [19] développa un robot mobile minimaliste, le *Stanford Cart*, équipé d'une tourelle de vision stéréoscopique. Ce robot, bien que très lent, était capable de mettre à jour son modèle du monde et d'y inclure de nouveaux obstacles, sans pour autant recourir à une représentation symbolique. Son système de vision servait à localiser des espaces libres d'obstacles et aucune reconnaissance d'objets en tant que telle n'était réalisée [20].

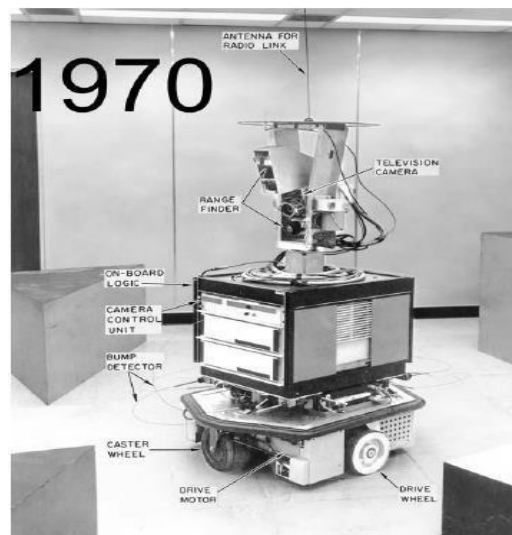


Figure I.11 : Le robot Shakey de Stanford en [20].

C'est à cette période que certains chercheurs commencèrent à se douter que l'approche classique, plutôt que de résoudre et de simplifier les problèmes, les compliquait inutilement en concentrant les efforts sur le développement de systèmes de traitement symbolique de l'information.

Le problème le plus critique était certainement celui du fondement des symboles, en anglais "*symbol grounding problem*", qui se réfère à la difficulté de mettre en relation la

signification d'un symbole avec l'objet réel ou l'événement qu'il représente. La description d'un symbole au moyen d'autres symboles conduit très souvent à des structures circulaires ou récursives. L'introduction de nouveaux symboles pour définir un symbole existant conduit à la prolifération de ceux-ci, sans pour autant leur attribuer une signification réelle. Les humains sont capables de fonder les symboles sur leurs perceptions sensorielles (c'est le fondement des symboles): la signification des symboles émerge de l'interaction avec les objets de l'environnement, elle ne leur est pas inhérente.

Un autre problème lié aux approches classiques est dû à la faible robustesse des systèmes de contrôle à l'égard du bruit et des événements inattendus. L'interaction d'un robot réel avec son environnement comporte un risque élevé de voir celui-ci se perdre s'il est piloté par un système classique. En effet, il lui est très difficile de tenir compte des écarts qui apparaissent entre sa perception du monde et la représentation symbolique qu'il en a, et de les corriger. Ainsi, un système de navigation par amers reposant sur une représentation symbolique se perdra ou ne saura pas réagir de manière appropriée lorsque ses capteurs lui fourniront des mesures inattendues ou entachées de bruit : si un amer perçu ne peut pas être mis en relation avec un symbole connu, ou qu'il est mis en relation avec un symbole incorrect, le système réagira de manière erronée. Un tel système ne peut donc pas être considéré comme entièrement autonome. D'autres approches s'avèrent nécessaires pour faire face aux nombreux problèmes inhérents au monde réel, tels que l'incertitude des connaissances, le bruit entachant les grandeurs mesurées par les capteurs, les modifications dynamiques de l'environnement, la nécessité de répondre rapidement et correctement à des situations inattendues, etc [20].

Un certain nombre de nouvelles approches, en partie inspirées par les modèles cybernétiques et biologiques, ont donc été proposées :

- **Une approche sans représentations** : les représentations internes, de forme purement symbolique, permettent de décrire un phénomène de façon abstraite et n'ont qu'un lien ténu avec la réalité. Une approche sans représentations permet, en s'acquittant d'un niveau d'abstraction élevé, de garder un lien direct entre le robot et le monde physique dans lequel celui-ci évolue.

- **Une approche réactive** : pour de nombreuses tâches, la réponse directe et réactive à des événements extérieurs constitue une solution simple et efficace. Elle permet de s'acquitter des représentations symboliques complexes souvent nécessaires aux systèmes de planification classiques. Les comportements d'un robot basé sur une telle approche sont sélectionnés directement en fonction des stimuli perçus. Les comportements qui entraînent une modification du monde (ou une modification de sa perception) entraînent à leur tour l'activation de nouveaux comportements.

- **Une approche distribuée** : la majorité des systèmes d'intelligence artificielle classique reposaient sur le postulat que l'intelligence pouvait être assimilée à un processus séquentiel de traitement de l'information, basé principalement sur des inférences logiques. L'approche distribuée combine les caractéristiques des deux approches décrites ci-dessus et répartit la prise de décision dans des modules séparés. Le choix final de l'action à exécuter résulte de la coopération ou de la compétition des divers modules. L'architecture à subsomption de Brooks est un exemple d'une telle approche, distribuée, réactive et sans représentations [26], Ces nouvelles approches s'appuient souvent sur des modèles biologiques. De par leur lien étroit avec les recherches en vie artificielle, elles ouvrent de nouvelles perspectives pour l'intelligence artificielle, Elles ont entre autres donné le jour à la robotique comportementale, à l'approche animât et à la robotique animale.

I.6.3. L'autonomie d'un robot

L'autonomie est la capacité à résister à des perturbations externes en utilisant les ressources internes. L'autonomie d'un robot est une faculté relative et non absolue, elle est liée aux capacités du robot, aux caractéristiques de l'environnement dans lequel il est plongé, à leurs variations, et enfin aux tâches qu'il doit effectuer. Le concept d'autonomie est complexe lorsqu'on considère l'interaction d'un robot avec son environnement.

Selon Steels [28], un système est autonome s'il développe les lois et les stratégies qui lui permettent de contrôler son comportement. En fait, l'autonomie est une capacité relative et on peut considérer qu'il existe une progression insensible du niveau le plus bas au plus élevé. Elle représente la capacité à choisir une stratégie, en termes de sous-buts pour arriver à un but fixé. La limite basse de l'autonomie est constituée par un système automatique qui s'autorégule en fonction de lois préétablies i.e. qui ne génère pas les lois que les activités de régulation cherchent à satisfaire.

L'autonomie désigne littéralement la capacité d'une entité à « *se gouverner par ses propres*

lois ». Cela signifie entre autres le choix de ses buts. Dans la littérature robotique, on peut distinguer trois points de vue :

- L'autonomie au sens fort qui nous ramène à des questions de volonté et de but propre.
- L'autonomie au sens faible désigne la capacité de maintenir sa structure au sein d'un milieu complexe à travers des mécanismes tels que l'auto- organisation, l'évolution, l'adaptation et l'apprentissage.
- L'automatisme ou absence de contrôle extérieur est le sens implicite que l'on attribue généralement au mot autonomie.

S'adapter consiste pour un robot à modifier son comportement pour faire face à des changements internes ou externes afin de maintenir certaines propriétés. L'adaptation se rapporte toujours à quelque chose : adaptation d'un groupe à la panne d'un individu, adaptation à la présence de passants dans l'environnement...etc. Les bases de l'autonomie sont concrètement les propriétés qu'un robot doit exhiber pour être autonome, par exemple :

- Navigation et localisation pour accéder à tous les points de l'environnement.
- Surveillance, c'est-à-dire détection d'évènements anormaux.
- Efficience dans son travail, en adoptant une stratégie de patrouille.

Un robot complètement autonome n'est pas forcément souhaitable. On peut vouloir en prendre le contrôle à certains moments ou intervenir indirectement à travers la modification de certains paramètres. Le terme d'autonomie ajustable désigne justement l'interruption temporaire de l'automatisme, du fait d'un superviseur ou de l'agent lui-même, afin d'accroître l'efficacité du système. Le choix du degré d'automatisme dont doit être doté un agent autonome dépend de l'architecture de contrôleur.

▪ **Autonomie du mouvement et autonomie décisionnelle.**

L'autonomie du mouvement passe par la détermination des déplacements, de manière planifiée et réactive, et par la commande des déplacements. Si les méthodes mises en jeu sont fortement dépendantes des modèles cinématiques et dynamique des engins considérés, des méthodes génériques commencent à apparaître, surtout pour la planification de trajectoire et la commande référencée sur des éléments de l'environnement. Dans ce contexte, la taille de l'environnement considéré nous amènera plus particulièrement à nous intéresser à l'intégration de méthodes locales de gestion des mouvements avec des méthodes plus globales, qui

considèrent l'ensemble des informations disponibles sur l'environnement et la mission à réaliser. Ces derniers aspects relèvent surtout de l'autonomie décisionnelle : il s'agit de déterminer les modalités de déplacement à adopter et les informations à acquérir.

Les systèmes automatiques sont entièrement prévisibles, du moment que leur état interne est connu, alors que les systèmes autonomes ne le sont pas, puisqu'ils sont capables de prendre eux-mêmes des décisions en fonction de critères qui peuvent échapper à l'observateur. L'autonomie implique la liberté de contrôle. Un système autonome prend ses décisions lui-même, il n'est pas contrôlé par un agent externe. La prise de décision implique une capacité à évaluer des alternatives en fonction d'un état courant et de l'expérience acquise par le passé [20].

Posée dans le contexte d'environnements larges, éventuellement dynamiques, enrichie par la considération de systèmes multirobots pas nécessairement homogènes, et de contraintes de communication et de gestion des ressources, cette problématique est encore très ouverte. La possibilité d'interactions avec des opérateurs, distants ou non, doit aussi être considérée à différents niveaux au sein des robots. Ces problèmes d'autonomie ajustable passent notamment par le développement de concepts d'organisation des processus décisionnels (architecture de contrôle).

▪ **Autonomie envisageable pour les robots actuels**

Il ne paraît pas suffisant de viser le développement d'un ensemble de fonctionnalités indépendantes, sous forme de comportements ou de modules, mais un ensemble cohérent où toutes les fonctionnalités puissent être mise en œuvre sur des démonstrateurs dans une même unité de lieu et de temps. La phase d'étude des missions a permis d'extraire des tâches "*robotisables*" que l'on peut décomposer en comportements sensorimoteurs ou en fonctions de surveillance. Les premiers se distinguent des suivants par la présence d'actions motrices. Ces éléments sont listés ci-après [21]:

- faire un déplacement dans une direction et selon une distance données
- se diriger vers un amer visuel (désigné au robot par un opérateur humain)
- suivre un guide visuel au sol (bord de mur, de trottoir, d'accotement, couloir)
- s'orienter dans une direction donnée ;
- contourner un obstacle
- explorer une zone délimitée
- porter ou déposer une charge

- collecter un objet ;
- se servir d'un outil spécifique
- utiliser des armes non létales

I.6.4. Environnement dynamique et incertain

▪ **Notion d'environnement dynamique**

Un environnement est dit dynamique s'il comporte des obstacles susceptibles de changer au cours du temps. Soit le cas des obstacles qui se déplacent (un piéton, un véhicule, ...), ou ceux qui changent de forme ou de ceux qui apparaître/disparaître (une porte coulissante semble « disparaître » dans le mur quand elle s'ouvre).

▪ **Notion d'incertitude**

Une information est incertaine, si elle est bruitée (mauvaises conditions de mesures), incomplète (obstruction d'un capteur ou portée limitée, absence d'informations sur l'évolution d'un objet ou d'un phénomène) ou imprécise (les glissements des roues par rapport au sol sont observables mais rarement mesurables avec précision).

I.6.5. Problématiques de la robotique mobile

La robotique mobile cherche depuis des années à rendre une machine mobile autonome face à son environnement pour qu'elle puisse sans intervention humaine accomplir les missions qui lui sont confiées. Le spectre des missions que les roboticiens veulent voir accomplir par leurs machines est immense : exploration en terrain inconnu, manipulation d'objets, assistance aux personnes handicapées, transport automatisé, etc [22].

On distingue sans trop d'ambiguïté un certain nombre de problèmes en robotique mobile. En plus de pouvoir percevoir globalement son environnement, un robot doit souvent être capable d'identifier des objets, de reconnaître des personnes, de lire des indications, et même de repérer des symboles graphiques. Ces opérations sont effectuées en analysant les perceptions acquises par les capteurs.

Enfin, une autre capacité robotique tout aussi importante que celles énumérées précédemment, est la capacité pour un robot de prendre lui-même ses propres décisions nécessaires pour réaliser et coordonner des missions complexes. Cette capacité est d'une grande importance puisque, pour beaucoup de tâches robotiques, il peut exister de nombreuses façons de les réaliser. Par un raisonnement, le robot doit alors sélectionner les meilleures actions à effectuer pour réussir adéquatement sa mission [14].

Bien évidemment, l'aspect matériel, qui consiste à choisir et dimensionner aussi bien la

structure mécanique du système que sa motorisation, son alimentation et l'architecture informatique de son système de contrôle-commande apparaît comme le premier point à traiter. Le choix de la structure est souvent effectué parmi un panel de solutions connues et pour lesquelles on a déjà résolu les problèmes de modélisation, planification et commande.

Le choix des actionneurs et de leur alimentation est généralement assez traditionnel. La plupart des robots mobiles sont ainsi actionnés par des moteurs électriques à courant continu, alimentés par des convertisseurs de puissance fonctionnant sur batterie. De la même façon, les architectures de contrôle- commande des robots mobiles ne sont pas différentes de celles des systèmes automatiques ou robotiques plus classiques. On y distingue cependant, dans le cas général, deux niveaux de spécialisation, propres aux systèmes autonomes : une couche décisionnelle, qui a en charge la planification et la gestion (séquentielle, temporelle) des évènements et une couche fonctionnelle, chargée de la génération en temps réel des commandes des actionneurs [23].

Au fil des ans, plusieurs solutions ont été proposées afin de donner aux robots la capacité de coordonner leurs missions. L'une d'entre elles est l'utilisation d'outils de planification et navigation dans le domaine de l'intelligence artificielle [14].

Afin de faire fonctionner un robot mobile, plusieurs modules logiciels sont mis à contribution. Ces modules peuvent servir à interpréter les données perçues par les capteurs afin d'y extraire des informations ou à traiter des commandes de haut niveau et générer d'autres commandes à un niveau inférieur. Les modules les plus fréquemment utilisés sont les modules de localisation, de navigation, de vision, d'audio et de séquençement d'activités du robot.

Un robot mobile est commandé par une boucle de contrôle, cela est illustré à la)figure I.12(. De façon itérative, cette boucle fait une lecture des données reçues par les capteurs, les interprète, calcule les commandes motrices et les envoie aux actionneurs. Typiquement, cette boucle est exécutée environ dix (10) fois par seconde ; la fréquence peut varier selon les types de capteurs et d'actionneurs utilisés. La boucle de contrôle n'est pas unique ; selon l'architecture utilisée, elle peut être décomposée en plusieurs sous boucles de contrôle agencées de manières différentes [14].

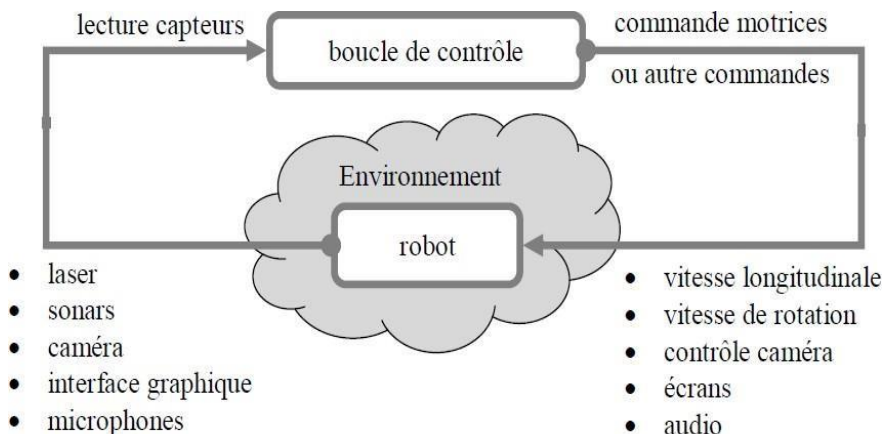


Figure I.12 : Boucle de contrôle pour la commande d'un robot mobile

I.6.6. Architectures de contrôle

Un robot est donc composé d'un ensemble de modules, chacun étant responsable d'une ou plusieurs capacités. Un des premiers défis à résoudre est de déterminer comment relier efficacement les différents modules. Pour ce faire, il faut élaborer une architecture de contrôle qui dictera les responsabilités de chacun des modules et comment les informations circuleront entre ces derniers.

Depuis les débuts de la robotique, beaucoup d'architectures ont été proposées. Elles peuvent être généralement classées en trois grandes catégories : délibérative, comportementale et hybride.

La définition des différents types d'architecture s'est faite au fil du temps, et s'est faite grâce à l'analyse des points faibles de chaque type d'architecture [24]. La classification des architectures de contrôle repose sur la stratégie utilisée pour contrôler le système et pour atteindre ses objectifs :

- L'application directe des techniques d'intelligence artificielle telles que la planification dans l'école cognitive ;
- L'utilisation des comportements réactifs dans l'école réactive ;
- La centralisation de contrôle du groupe dans un superviseur externe ou la distribution.

▪ Architectures de contrôle cognitives

Dans une architecture de contrôle cognitive, le robot cherche tout d'abord à modéliser les connaissances et les méthodes d'inférence associées permettant d'élaborer les actions que le robot doit entreprendre. Le robot commence par traiter les données recueillies par ses capteurs. Ensuite, il identifie les objets qui sont dans son environnement proche. Puis il construit une représentation de la scène dans son ensemble, et l'utilise pour générer un plan.

Après quoi, il calcule au mieux une séquence de commandes vers les effecteurs pour exécuter le plan prévu.

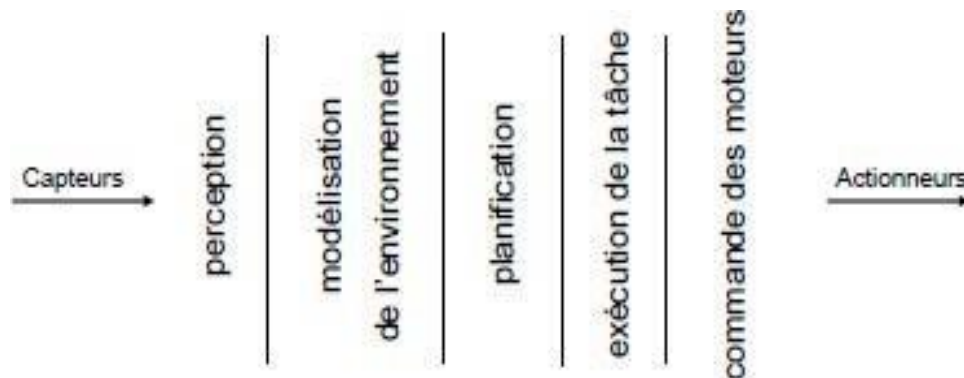


Figure I.13 : Architecture de contrôle cognitive [25]

▪ Architectures de contrôle réactives

Les architectures réactives ont pour but de concevoir et de réaliser un robot, capable d'apporter une réponse immédiate à toute nouvelle modification de l'environnement le concernant. Ces approches sont initialement fondées sur l'étude de l'interaction de l'animal avec son environnement naturel (l'éthologie) de manière à essayer de comprendre les mécanismes mis en œuvre et de les reproduire. Elles sont fondées sur le fait que lorsqu'un être humain évolue dans un environnement, il ne passe pas son temps à modéliser son environnement, et à planifier ses actions, mais qu'au contraire il réagit à de simples stimuli qui lui permettent de se déplacer vers son but tout en faisant face aux obstacles.

Ainsi, contrairement aux systèmes hiérarchiques, le comportement global du robot apparaît comme le résultant de plusieurs comportements actifs simultanément. Dans ce type d'architectures il n'y a pas de phase de planification ou de modélisation de l'environnement [24].

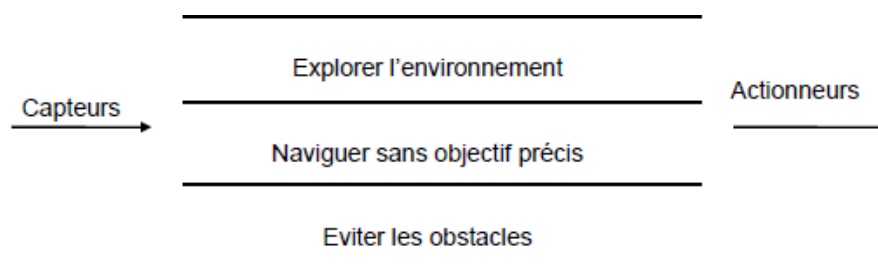


Figure I.14 : Architecture de contrôle réactive [25]

▪ Les architectures de contrôle comportemental réactif

Pour l'élaboration d'architecture de contrôle réactif de plus en plus complexe, [26]) propose de décomposer le comportement global du robot en un ensemble d'entrées élémentaires. Cette décomposition a comme principaux objectifs d'avoir une construction aisée de

l'architecture de contrôle ainsi qu'une testabilité facile des comportements élémentaires. Ceci est rendu possible en isolant les comportements qui cohabitent dans une même structure de contrôle, impose l'adaptation de mécanismes approprié de coordination entre comportements.

Les architectures de contrôle réactives permettent d'avoir une réponse plus rapide grâce à un lien direct capteurs-actionneurs. Disposer des comportements élémentaires dans le cas des approches comportementales permet de les tester individuellement jusqu'à ce qu'ils soient adaptés à leurs tâches. La mise à jour se fait alors simplement en les ajoutant aux comportements déjà existe. [27], [25] ont comparé les caractéristiques des deux types architectures de contrôle. La Figure II.11 détermine la comparaison entre les deux architectures selon les degrés d'intelligences.

Cognitive	Réactive
<p>Vitesse de la réponse</p> <p>→</p>	
<p>Capacités prédictives</p> <p>←</p>	
<p>Dépendance à la précision, modèle du monde complet</p> <p>←</p>	
<ul style="list-style-type: none"> • Modélisation explicite de l'environnement • Vitesse de réponse relativement lente dans certains environnements • Haut niveau d'intelligence • Planification sophistiquée et préalable de la tâche. • Peut tenir compte de son passé. 	<ul style="list-style-type: none"> • Absence d'une modélisation explicite. • Fonctionne en temps réel. • Peu ou pas d'intelligence • Fonctionne en stimulus-réponse • Pas de mémoire de son histoire.

Figure I.15 : Contrôle cognitif versus réactif [25]

▪ Architectures Hybrides

Les approches cognitives et réactives sont diamétralement opposées. Cependant, chacune présente des caractéristiques intéressantes. Pour cela, les chercheurs ont essayé de les combiner en mettant au point des architectures hybrides permettant notamment d'allier des capacités de raisonnement et de décision de haut niveau, s'appuyant sur des représentations abstraites des connaissances, avec des comportements réactifs garantissant robustesse et flexibilité. En effet, l'exécution de plans est basée sur un contrôle réactif (ce que détectent les capteurs permet de réagir immédiatement en cas de nécessité), alors que la formulation de plan(s) est assurée par un système hiérarchique pouvant reposer sur un modèle de l'environnement et des obstacles connus [24].

Les développements les plus récents convergent vers l'utilisation d'architectures à deux ou trois niveaux. Dans ce type d'architecture, les niveaux bas sont consacrés à un ensemble de comportements sensorimoteurs et à leur mécanisme séquenceur, tandis que le niveau le plus élevé est chargé de la planification des enchaînements de comportements et le niveau intermédiaire, responsable de l'activation des comportements, est basé sur une approche appelée séquencement conditionnel. Ce niveau utilise les plans établis par le niveau supérieur.

Ces plans sont considérés comme des ressources (ou des conseils) et non comme des ordres par le séquenceur. La remise en question d'un plan fait suite à l'échec d'une action détectée par des routines de supervision implantées au-dessus des comportements.

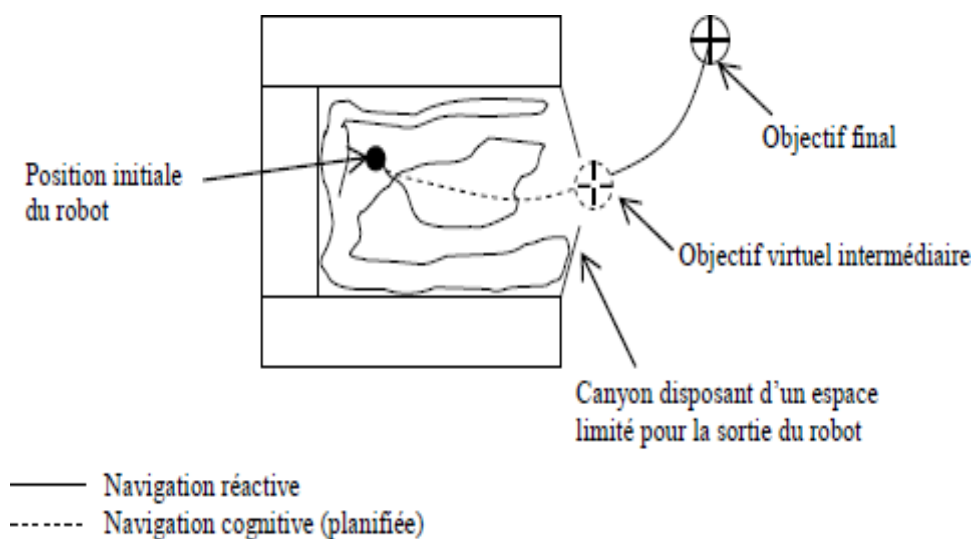


Figure I.16 : Architecture de contrôle hybride

I.7. Conclusion

Le développement de l'intelligence artificielle classique a conduit à la réalisation des premiers robots mobiles, évoluant dans des environnements strictement contrôlés. Leurs actions dans le monde réel n'étaient que l'expression physique d'opérations symboliques réalisées dans un modèle abstrait. Ces robots étaient incapables de réagir de manière appropriée à des perturbations ou à des événements inattendus, ce qui a rendu impraticable leur utilisation dans le monde réel.

L'objectif de ce chapitre est de fournir un aperçu général sur la robotique mobile, ses problématiques et les solutions apportées actuelles. Nous avons présenté également un panorama des architectures de contrôle dont la portée va au-delà de leur application en robotique mobile. La robotique mobile est un domaine dans lequel l'expérience pratique est primordiale. L'approche de la robotique présentée dans ce chapitre est essentiellement celle issue de l'intelligence artificielle. D'autres disciplines, telles que l'automatique, sont toutefois impliquées et peuvent présenter un éclairage différent, mais ne changent pas fondamentalement les problèmes qui restent à résoudre.

Dans le chapitre suivant nous expliquerons une détaille de robot mobile khepera des caractéristiques ainsi qu'un aperçu historique de ses générations.

Chapitre II

II.1. Introduction

Le Khepera est un petit robot mobile à roues différentielles (5,5 cm) qui a été développé dans le laboratoire LAMI du professeur Jean Daniel Nicoud à l'EPFL (Lausanne, Suisse) au milieu des années 1990. Il a été développé par Edo. Franzi, Francesco Mondada, Andre Jenniard et coll.

Il est petit, rapide et conçu autour de Motorola 68331, a servi des chercheurs pendant 10 ans et est largement utilisé par plusieurs universités à travers le monde.

On va commencer ce chapitre par un petit aperçu historique sur les différentes versions de robot Khepera, et mettre en évidence la quatrième génération du robot Khepera et les composants existant dans cette génération et leur caractéristique.

II.2. Historique sur les robots Khepera

Khepera est un petit robot mobile pour recherche et enseignement. Des modules additionnels peuvent être très simplement ajoutés (caméras, gripper, entrée sortie (IO) interface). (Plus de 700 unités vendues à la fin 1998).

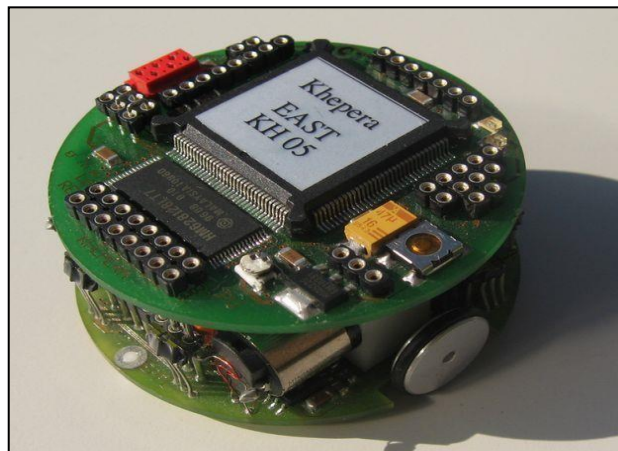


Figure II.1 : Le premier robot Khepera publié en 1996

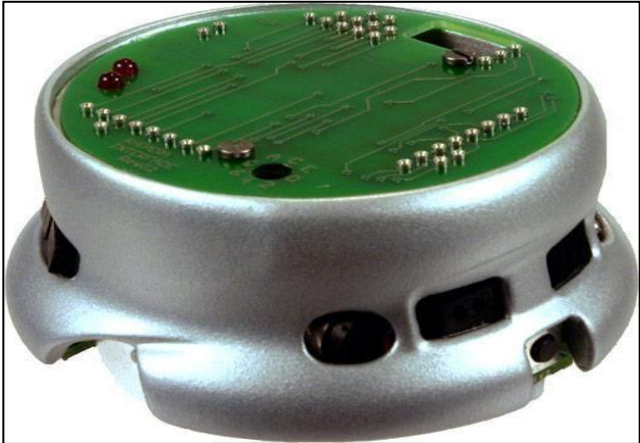


Figure II.2 : Khepera II



Figure II.3 : Khepera III



Figure II.4 : Khepera IV

II.3. Généralité sur le robot Khepera IV

- **Définition**

Le robot Khepera IV, conçu et produit par K-Team, est l'une des dernières plateformes à rejoindre notre boîte à outils. C'est un robot compact pour une utilisation en intérieur, avec un certain nombre de capteurs intégrés et une extensibilité facile, ce qui le rend particulièrement bien adapté aux expériences multi-robots. [29]

Le robot Khepera IV est la 4^{ème} version du robot mobile Khepera pour l'éducation et la recherche. Ce robot mobile compact a été conçu pour des surfaces planes (sol du laboratoire, table etc.). Il embarque ce qui se fait de mieux en termes de capteurs et moteurs de précision. [30]

Cette nouvelle version du robot Khepera apporte de nombreuses nouveautés : OS Linux, caméra couleur, Wi-Fi, Bluetooth, Host USB, accéléromètre, gyroscope et une odométrie améliorée et plus précise.

Le robot mobile Khepera IV peut être utilisé pour la recherche et l'éducation en navigation, intelligence artificielle, les systèmes multi-agents, le contrôle, les comportements collectifs, la programmation temps - réelle etc.

- **Description physique**

Le robot mobile Khepera IV est un petit robot différentiel à roues conçu pour l'utilisation intérieure. Il a la forme d'un cylindre, avec un diamètre de 14,08 cm et une hauteur au sol de 5,77 cm (y compris les roues). Sa coquille extérieure est composée de deux pièces en plastique avec fentes pour les capteurs et actionneurs. La figure II.5 montre les dessins techniques pour le robot. Le robot complet pèse 566 g.

Les deux roues actionnées ont un diamètre de 42 mm (comprenant les joints toriques qui servent de pneu) et sont centrés sur chaque côté du robot espacés de 10,54 cm, transfert de deux boules unités, à l'avant et à l'arrière, fournir le reste points de contact, cette solution aboutit à 0.5-1 mm de dégagement moulu, rendant le robot très stable mais empêchant son utilisation sur une surface qui n'est pas à plat et lisse.

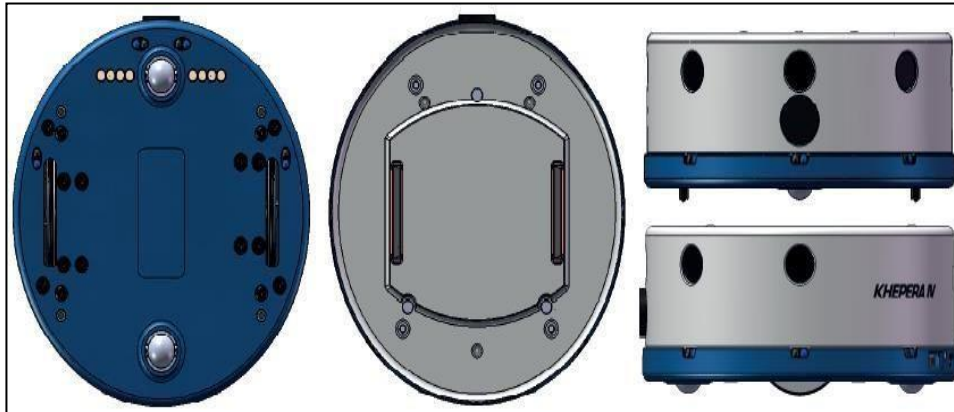


Figure II.5 : Vue du bas, du haut, de l'avant et de la gauche du robot

II.4. Fonctionnalités principales du robot mobile Khepera IV [31]

Le robot khepera IV est :

- Compacte ;
- Hautement modulaire ;
- Forte capacité de calcul embarqué ;
- Nombreux capteurs embarqués ;
- Moteur courant continue (CC) précis et de haute qualité ;
- Nombreux points de fixation pour adapter vos extensions.

II.5. Caractéristiques techniques du robot Khepera IV [17]

Les caractéristiques techniques du robot Khepera IV sont les suivantes :

- Processeur : Linux embarqué fonctionnant sous ARM Cortex-A8 @ 800 MHz + DSP Core (C64x) ;
- RAM : 512 Mo ;
- Mémoire : 512 Mo NAND flash + 4 Go on micro SD card ;
- Motorisation : 2 moteurs CC à balais avec encodeurs incrémentaux (environ 147 pulses par mm de déplacement du robot) ;
- Vitesse maximale : 1 m/s directement et 0,9 m/s en utilisant le PID par défaut ;

- Contrôleur de vitesse : Min : 0.003 m/s ; □ Capteurs :
 - ❖ 8 capteurs de proximité infrarouge pour la navigation ;
 - ❖ 4 capteurs infrarouges pour le suivi de ligne et la détection de rebord ;
 - ❖ 5 sonars à ultrasons pour la détection d'obstacle à longue distance (de 25 cm à 250 cm) ;
 - ❖ Accéléromètre 3 axes et gyroscope 3 axes.
- Audio : 2 microphones embarqués ;
- 1 haut-parleur : 0.7 W 400 à 20000 Hz ;
- Vidéo : Caméra couleur intégrée (752x480 pixels, 30 FPS) ;
- Pack batterie à 7.4 V / 3400 mAh ;
- Autonomie : entre 4 et 7 heures ;
- Communication : 1x port USB 2.0 500 mA (maître) et 1x port USB 2.0 500 mA (esclave) ;
- Connectivité Bluetooth ;
- Wifi 802.11 b/g ;
- Taille : Diamètre de 140 mm, hauteur de 58 mm ;
- Garde au sol : 4 mm. Le robot Khepera IV s'utilise sur une surface plane et solide ;
- Rayon de virage : 0 cm ;
- Indication : 3 LED RVB sur le dessus ;
- Poids : 540 g ;
- Charge utile : environ 2 kg ;
- Plage de température opérationnelle : entre 0°C et 40°C ;
- Environnement de développement pour comportement autonome GNU C/C++ compilé.

II.6. Les différents composants du robot mobile Khepera IV

- **Vue globale du robot**

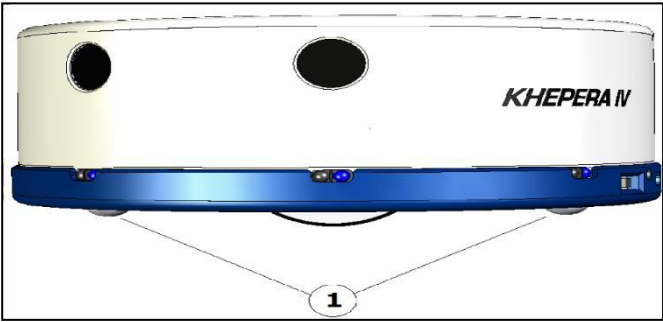


Figure II.6 : Vue de gauche

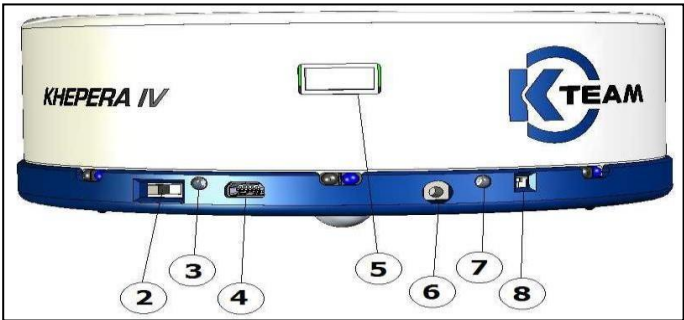


Figure II.7 : Vue arrière

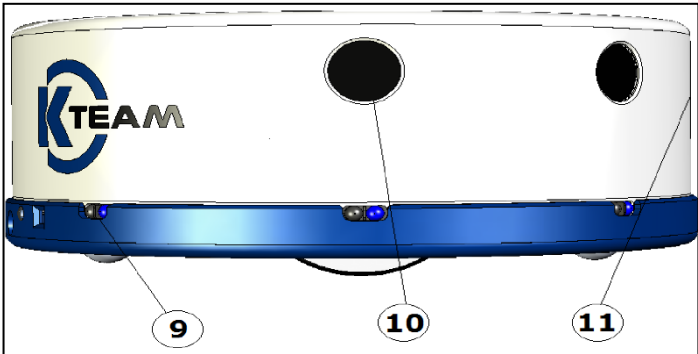


Figure II.8 : Vue de droite

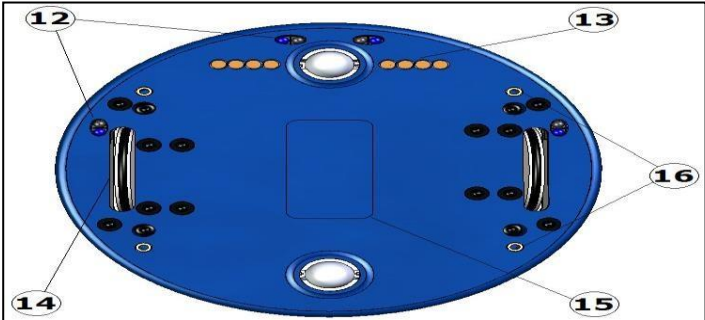


Figure II.9 : Vue de dessous

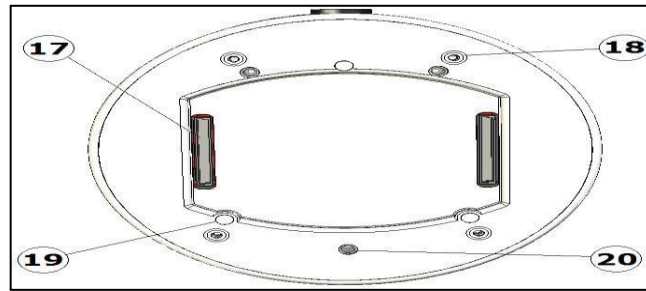


Figure II.10 : Vue de dessus

- 1 Les roues pivotantes
- 2 Bouton marche-arrêt
- 3 LED d'état
- 4 Connecteur Mini-USB B (mode appareil, sans charge)
- 5 Connecteur USB A (mode hôte, 500 mA)
- 6 Prise d'alimentation (9 V, 1.5 A, 0.65 mm centre positif)
- 7 Voyant d'état de charge
- 8 Bouton de réinitialisation
- 9 Capteurs infrarouges (8x)
- 10 Capteurs à ultrasons (5x)
- 11 Caméra
- 12 Capteurs à infrarouges de fond (bas) (4x)
- 13 Contacts pour station d'accueil
- 14 Roues
- 15 Autocollant
- 16 Bas écrous M3
- 17 Connecteurs d'extension 250 Ko
- 18 Haut écrous M3
- 19 Aimants
- 20 LED RVB

- **Vue détaillé des composantes du robot :**

- **Les roues pivotantes**

Il y a deux roues pivotantes en dessous du robot. Ces deux roues permettent au robot d'être très stable, même à forte charge utile ou avec les modules d'extension d'en porte-à-faux long. En retour, le robot n'est pas en mesure de passer par les seuils de port.

- **Bouton marche-arrêt**

Ce commutateur aura une action sur la réglementation interne du robot et non pas sur la batterie, ce qui signifie que vous pouvez le charger même s'il est éteint. Pour activer le robot, mettre le commutateur sur l'intérieur. Pour le désactiver, mettre le commutateur sur l'extérieur.

- **LED d'état**

Par défaut, ce voyant bicolore vous permet de connaître l'état du robot. Lorsque vous le mettez sous tension, le voyant vert reste allumé et le voyant rouge clignote jusqu'à ce que le système soit prêt. Cette LED est contrôlable par l'utilisateur.

- **Connecteur Mini-USB B (mode appareil, sans charge)**

Ce connecteur vous permet d'ouvrir un lien de communication entre le robot et un ordinateur. Il n'est pas possible de charger le robot de cette manière.

- **Connecteur USB A (mode hôte, 500 mA)**

C'est un hôte USB connecteur compatible. Vous pouvez brancher sur ce périphérique USB que vous voulez, aussi longtemps qu'il ne pioche pas plus que 500 mA de courant. Vous pouvez par exemple connecter un module GPS ou une clé de mémoire.

NB : Ce connecteur n'a pas ses signaux partagés avec le KO-250. Les connecteurs d'extension, ce qui signifie que vous ne pouvez pas utiliser un port USB sur les postes en même temps que sur ce connecteur.

- **Prise d'alimentation (9 V, 1.5 A, 0.65 mm centre positif)**

Il s'agit de la prise centrale positive de 0,65 mm utilisée pour charger la batterie interne du robot. Utilisez uniquement l'adaptateur fourni. La tension d'entrée est de 9 V. Courant consommé par le robot est 1A, de sorte qu'un 1,5 + un adaptateur est nécessaire.

- **Voyant d'état de charge**

Il s'agit d'une LED d'indication bicolore indiquant l'état de la charge. Là sont deux modes, selon si le robot est sous tension ou non.

Si le robot est arrêté et que l'adaptateur secteur est branché, le voyant rouge est allumé

aussi longtemps que la batterie est en charge. Une fois la charge terminée, le voyant s'éteint. Si le robot est allumé et que l'adaptateur secteur est branché, la LED rouge s'allume pendant la charge de la batterie. Une fois la charge terminée, la LED rouge s'éteint et la LED verte s'allume.

- **Bouton de réinitialisation**

Ce bouton permet de réinitialiser l'ensemble de robot, y compris les modules d'extension.

- **Capteurs infrarouges**

Khepera IV a 8 capteurs infrarouges placés tout autour du robot et 4 placés sur le fond. Ces derniers permettent aux expériences comme ligne suivant ou l'évitement de l'obstacle.

Ils sont placés et numérotés comme indiqué dans la figure ci-dessous :

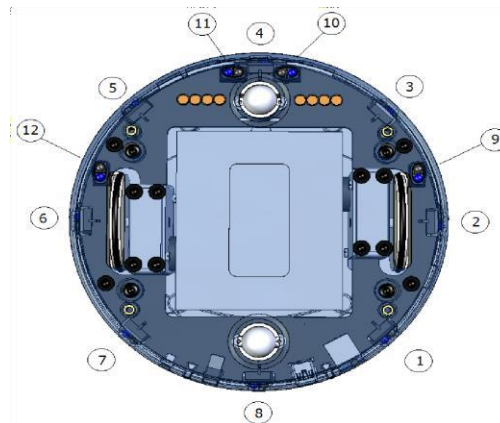


Figure II.11 : Capteurs à infrarouges vus de fond

Ces capteurs intègrent une lumière infrarouge émettrice et le récepteur. Pour une description détaillée, veuillez-vous reporter à la fiche technique du fabricant.

Les douze capteurs sont TCRT5000 capteurs optiques réfléchissants de Vishay Telefunken. Plage de mesure est de 2 à 250 mm et une longueur d'onde de 950 nm. Chaque capteur est séparé de son voisin d'un angle de 45°. La fréquence d'échantillonnage des capteurs infrarouges est de 200 Hz, quel que soit le mode de fonctionnement.

Ce type de détecteurs permet deux mesures :

- La lumière ambiante normale : Cette mesure est réalisée en utilisant uniquement la partie de l'appareil récepteur, sans émettre de la lumière avec l'émetteur.

Une nouvelle mesure est effectuée tous les 5 ms. La valeur renvoyée à un moment donné est le résultat de la dernière mesure effectuée ;

• La lumière réfléchié par les obstacles : Cette mesure est réalisée en émettant de la lumière à l'aide de l'émetteur de l'appareil. La valeur retournée est la différence entre la mesure effectuée en émettant de la lumière, et la lumière mesurée sans émission de lumière (lumière ambiante). Une nouvelle mesure est effectuée tous les 5 ms. La valeur renvoyée à un moment donné est le résultat de la dernière mesure effectuée.

- **Mesure de la lumière ambiante**

Mesure de la lumière ambiante est fortement influencé par l'environnement du robot. Selon le type de source de lumière, la couleur, et de la distance de mesure de la lumière ambiante, le profil peut varier. Il n'est pas recommandé d'utiliser avec une grande source de lumière dans l'émission infrarouge, car cela pourrait perturber les capteurs IR. Valeurs 0 à 1023, 0 signifie aucune lumière et 1023 pour la pleine lumière.

- **Mesure de la lumière réfléchié (proximité)**

Les capteurs sont principalement destinés à détecter des obstacles autour du Khepera. Les mesures pour la lumière réfléchié dépendent de la réflectivité des objets et de la lumière ambiante. Les couleurs de l'objet, les matériaux et les surfaces n'ont une influence sur la réponse du capteur. En outre, comme tout capteur, capteurs IR sont soumis au bruit dans l'environnement. Pour toutes ces raisons, les graphiques ci-dessous sont donnés à titre indicatif et ne doivent pas être considérés comme des références. Valeurs 0 à 1023 bit, 0 signifie Pas d'obstacle, 1023 bit pour de très près de l'obstacle :

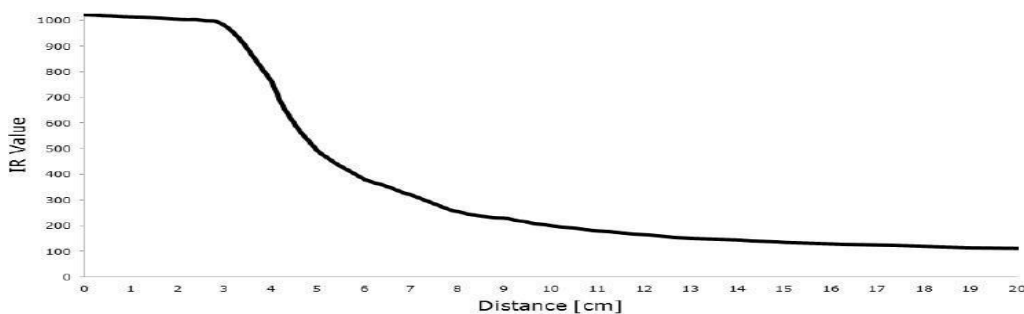


Figure II.12 : Valeur IR par rapport à la distance

La valeur IR ne tombe jamais à 0, puisque, même sans obstacle, l'IR se reflète sur le sol et ajoute une valeur statique. Comme tous les capteurs ne sont pas exactement les mêmes, la solution est d'effectuer un étalonnage de l'IR sans obstacle devant. Avec cet étalonnage, l'utilisateur sera en mesure d'améliorer la détection d'obstacle à une distance supérieure à 20 cm.

- **Capteurs à ultrasons**

Cinq capteurs sont placés autour du robot et sont positionnés et numérotés comme indiqué dans la figure ci-dessous. Ces capteurs sont des émetteurs-récepteurs, ce qui signifie qu'ils peuvent émettre et recevoir les impulsions.

Les capteurs à ultrasons sont entraînés par un Vpp 85 source. La fréquence nominale de ces capteurs est de 40 kHz +/- 1 kHz. La valeur retournée est la distance à l'objet en centimètres, avec une tolérance de +/-2 cm. Plage de mesure est de 25 à 200 cm. Chaque transducteur est séparé de son voisin d'un angle de 45°.

Chaque capteur peut être désactivée afin d'obtenir des taux de rafraîchissement pour un particulier (ou un groupe). Un capteur mesure prend 20 ms. Tous les cinq capteurs besoin de 100 ms à être lu.

Le taux d'échantillonnage effectif dépend du nombre des capteurs activés, allant de 50 Hz pour un seul capteur à 10 Hz si l'ensemble est en utilisation.

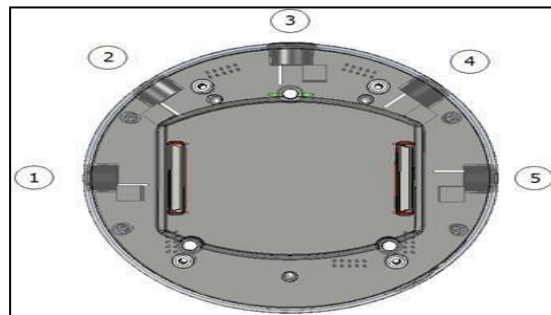


Figure II.13 : Capteurs ultrasoniques vus de sommet

- **Caméra**

Dans l'avant du robot, il y a une caméra couleur avec objectif interchangeable par l'utilisateur. Il peut être utilisé pour prendre des photos ou des films qui peuvent être traitées à bord. Le Khepera IV est équipé d'un appareil photo couleur, éliminés sous l'avant du capteur à ultrasons. Le capteur est un MT9V034C12ST de Aptina. C'est un capteur CMOS 1/3" WVGA.

- Taille d'imageur active : 4.51 x 2.88 mm
- Pixels actifs : 752 x 480

Le réglage par défaut de l'objectif a une focale de 2,1 mm, avec filtre anti-IR et mise au point fixe. La fixation filetage M12x0.5. C'est champ de vision diagonal de 150°, horizontal est 131° et vertical est 101°.

• Capteurs à infrarouges de fond (bas)

Quatre capteurs infrarouges sont disposés sur la partie inférieure du robot. Ils sont utilisés pour éviter le robot de tomber mais peut également être utilisé pour suivre une ligne.

• Contacts pour station d'accueil

Le Khepera IV a montré quelque contacts ci-dessous son corps qui signifie que vous pourriez utiliser une station d'accueil pour charger sa batterie ou de communiquer avec elle. Il est également possible d'imaginer de développer une extension qui serait en dessous du robot, par exemple une base mécanique avec les voies. Les signaux qui sont offerts sont :

Batterie (de contrôlé par un relais de roseau), 9V dans l'I2C.

• Roues

Le robot est conduit du différentiel, avec 2 roues équipé d'un joint torique. Les roues sont entraînées par des moteurs à courant continu (CC) avec encodeur et boîte de vitesses.

• Autocollant

Ici vous trouverez le numéro de série de votre robot.

• Bas écrous M3

Il y a la possibilité de fixer une extension vers le robot à partir de ci-dessous.

• Connecteurs d'extension 250 Ko

Ces deux connecteurs sont utilisés pour connecter les modules d'extension pour le robot. La broche de sortie est en conséquence à Ko-250, avec quelques améliorations mineures. Vous pouvez réutiliser Khepera III ou les modules d'extension KoreBot (mais pas le Grappin Kh3).

NB : Ces connecteurs ont leurs signaux d'hôte USB partagé avec l'hôte USB Connecteurs, ce qui signifie que vous ne pouvez pas utiliser un port USB sur les postes en même temps que sur le connecteur. Périphérique USB est utilisable sur les deux en même temps que le robot est équipé d'un moyeu.

• Haut écrous M3

Il y a la possibilité de fixer une extension vers le robot à partir de ci-dessus.

• Aimants

Il y a trois aimants utilisés principalement pour fixer le grappin, mais ils peuvent être utilisés pour n'importe quel autre module.

• LED RVB

Trois LED RVB (19-337/GHBHC6R-A01/2T d'Everlight) sont montés sur le haut de la carte principale. Chacune de ces LED est dotée d'un guide de lumière sur elle.

Les LED sont entraînées par un pilote de LED dédiés (LTC3219 de linéaire) qui offre une résolution de 6 bits (0-63) pour chaque couleur.

Ces LED peuvent être utilisées pour localiser le Khepera IV avec un appareil photo et de différencier chaque robot (en application de l'essaim). Comme les LED sont placées sur un triangle isocèle, la direction du robot peut également être détectée.

• Microphones

Le Khepera IV est équipé de deux microphones amplifiés PU0414HR5H-SB de Knowles. Ils sont directement connectés à l'Overo l'entrée micro. Le droit microphone est connecté à l'entrée principale et le micro à gauche sur l'entrée MIC SOUS.

- Gain 20 dB
- Sensibilité (type) -22 dbv/Pa
- Directivité Omnidirectionnelle
- Tension d'alimentation 2.5 V

Leurs positions sur l'ont Khepera indiqué sur la photo ci-dessous :

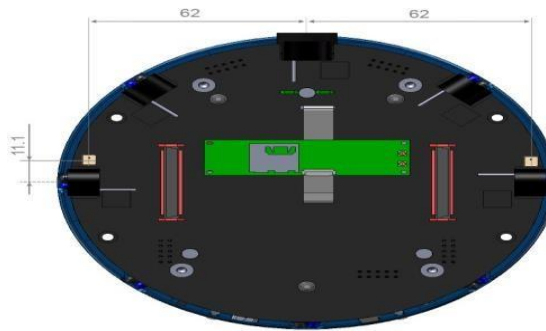


Figure II.14 : Positions de microphones

• Haut-parleur

Un SMS-1308MS-2-R haut-parleur de PUI audio est monté sur le Khepera IV. Ce haut-parleur est entraîné par un 1W basse distorsion amplificateur de puissance. Le haut-parleur est connecté sur la sortie audio de l'HSOLF OVERO. L'OVERO peut aussi couper l'amplificateur avec GPIO64 (0 = muet, 1= ampli).

- Le président Power 0.7 W (max 1W)
- Impédance 8 Ohms

- SPL de sortie 88 dB
- Distorsion (max) 5 %
- Fréquence de résonance 850 Hz \pm 20%
- Gamme de fréquence 400 ~ 20000 Hz

• Accéléromètre

L'accéléromètre monté sur le Khepera IV est une LSM330DLC de ST. Ce dispositif inclut dans un paquet un accéléromètre 3D et un gyroscope 3D.

L'appareil est exactement au centre du robot (placé sur le centre de rotation).

L'appareil est situé sur le haut de la carte principale.

L'accéléromètre est orienté vers l'axe 1 à l'avant droit, et cela renvoie une valeur positive pour l'axe X pour aller de l'avant. L'axe Y est positif sur la gauche, et enfin de l'axe Z est négatif avec la gravité.

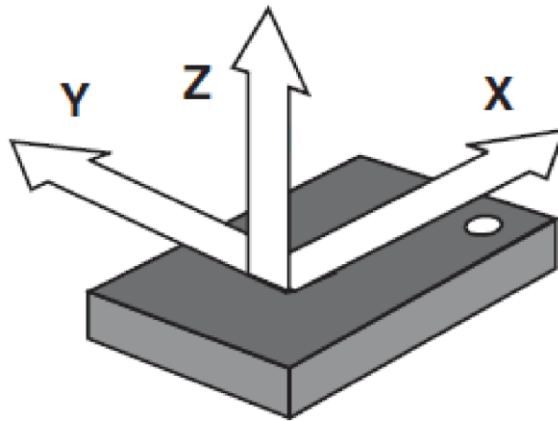


Figure II.15 : Les directions d'accélération détectables

L'accéléromètre renvoie 12 bits de données (complément à 2) avec une gamme de ± 2 g. Cela signifie qu'une valeur de 1g sera de retour d'une valeur de 16384. Le débit de données est configuré à 100 Hz, comme le dsPIC des actualisations Khepera 10 valeurs à la fois, l'utilisateur doit lire toutes les 100 ms (10 Hz) pour obtenir de nouvelles données.

• Gyroscope

Le gyroscope de Khepera IV est inclus dans le même package que l'accéléromètre.

Les directions des taux angulaire détectables sont définies autour de l'axe de l'accéléromètre.

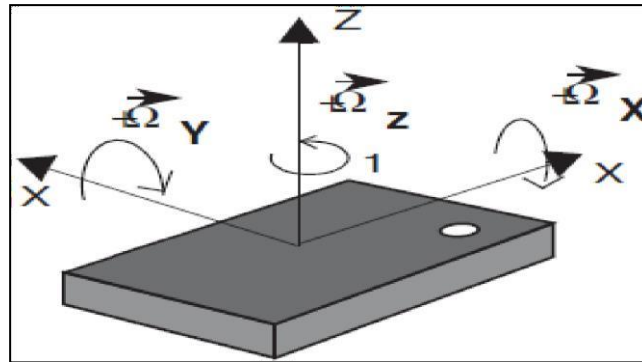


Figure II.16 : Les directions de taux angulaires détectables

Le format de données est sur 12 bits trop, la gamme complète est configuré à +/- 2000 dps (360 dps = 5898) et le débit de données est configurée à 95 Hz.

Le gyroscope, les données sont lues par des paquets de 10 valeurs à la fois, ce qui signifie que l'utilisateur peut lire les nouvelles données chaque 105 ms pour obtenir de nouvelles valeurs.

La sortie doit être multipliée par 0,066 d'avoir [deg/s] unités.

• Périphérique USB (mini-USB B connecteur)

Un mini-USB B vous donne accès à un port USB/série adaptateur (FT234XD de FTDI) pour accéder directement à le ttyS2 de la Gumstix. En utilisant un terminal fournit l'accès à l'amorçage du système.

Lorsque vous connectez un ordinateur à ce connecteur pour la première fois, le système vous demande pour pilote. FT234XD pilote peut être trouvé à <http://www.ftdichip.com/Products/ICs/FT234XD.html>.

• Carte MicroSD

Une carte MicroSD de 4 Go est fournie à l'intérieur du Khepera IV. Le robot démarre sur elle et utiliser cette carte. Il contient déjà l'OS, ainsi que les fichiers d'amorçage du noyau.

• Moteurs

Le Khepera IV a eu 2 moteurs à courant continu afin d'entraîner ses deux roues. Les moteurs ont 1.96 W puissance nominale. La boîte de vitesses intégrée a un rapport de réduction de 19:1 et d'une efficacité de 78 %. Il y a une autre boîte de vitesses dans le carter du bloc moteur, avec un ratio de 2:1 et d'une efficacité de 85 %. Ratio total est alors de 38:1 et l'efficacité est de 66,3 %, ce qui signifie qu'il est utilisable de 15 puissances mécaniques par roue.

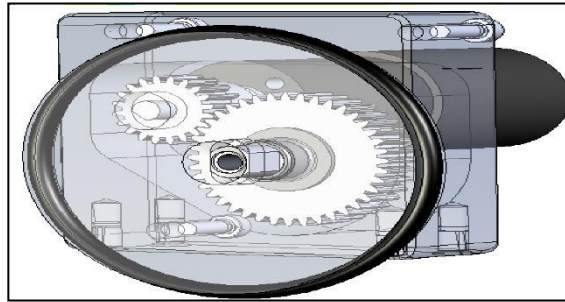


Figure II.17 : Bloc du moteur avec roue

• Batterie

Le Khepera IV est équipé d'un processus interne non-amovible batterie lithium polymère. Il est construit dans une configuration 2S1P 7.4 V, 8.4 V, nominale Tension de charge et une capacité de 3400 mAh.

• Tension nominale	7.4 V
• La tension de coupure	6.0 V
• La tension de charge	8.4 V
• Capacité nominale	3400 mAh
• Courant de décharge max	3400 mA (1C)
• Courant de charge	1100 mA
• Temps pour une charge complète	Environ 4 à 5 heures

Il n'y a pas de système de gestion de l'alimentation spécifique sur le Khepera. Lorsque la tension de la batterie tombe sous 6V, la batterie s'ouvre lui-même le circuit pour éviter une décharge profonde des cellules. Les utilisateurs peuvent mettre en œuvre leur propre système de gestion de l'alimentation du logiciel pour gérer les extensions à arrêter correctement avant de ce cas se produit.

La batterie peut être chargée à partir de 3 endroits :

- À partir de la prise ;
- Des contacts situés sous le robot ;
- Des connecteurs d'extensions 250 Ko.

II.7. Moyenne d'extension

La fonctionnalité native du robot peut être étendue Par l'utilisation de périphériques USB ou Bluetooth génériques, ou en concevant des cartes personnalisées enfichables dans le bus Ko-250. Cette connexion à 100 broches fournit une alimentation, I2C, SPI, Bus USB,

ainsi que des lignes plus spécifiques pour, par exemple, Interface dsPIC. K-Team commercialise plusieurs planches, y compris un préhenseur, un télémètre laser etc.

- **La tourelle d'extension**

La tourelle d'extension peut être branchée sur le haut du Khepera IV afin d'améliorer les capacités du Khepera IV. Cette tourelle offrira de nombreuses options différentes pour le robot :

- Télémètre laser (LRF) capacité de connexion (Hokuyo URG-04LX-UG01) ;
- StarGazer (global position intérieure et le roulement de la fonction de connexion) ;
- Batterie supplémentaire pour une plus grande autonomie avec la K4 pack chargeur externe ;
- L'accessibilité de l'IO.



Figure II.18 : La tourelle d'extension

- **Télémètre laser**

La tourelle d'extension peut recevoir en option un module télémètre laser (LRF). Il est utilisé pour faire de la navigation, spécialement pour la cartographie et la localisation.

Un laser à l'intérieur du capteur est balayage de l'environnement du robot, de mesurer chaque 0,36 degrés la distance à partir de n'importe quel objet à sa hauteur.



Figure II.19 : Extension télémètre laser

- **Le préhenseur**

Le préhenseur est une tourelle qui peut être branché sur le robot. Il donne à Khepera IV la capacité de saisir et déplacer des objets.

Deux degrés de liberté pour la manipulation et le transport précis et robuste d'objets de 5 cm de largeur et 50 g de charge.

Fait à partir de pièces mécaniques de haute précision.

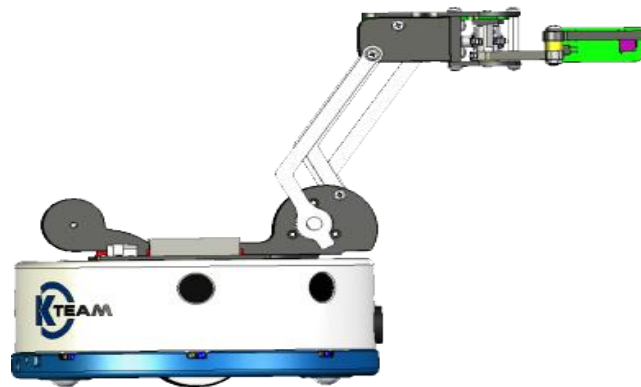


Figure II.20 : Extension avec le préhenseur

- **Système de positionnement global intérieur (Stargazer)**

Le Stargazer est un module permettant au robot Khepera IV d'avoir une position globale à l'intérieur et porter sur des repères passifs, Il est utilisé pour faire de la navigation, spécialement pour la cartographie et la localisation.

Le module a un IR LED's pour l'éclairage des points de repère et une caméra pour capturer l'image, qui est traité par l'électronique embarquée. Il calcule la position et le roulement liés aux repères collés au plafond, Il utilise le module Stargazer.

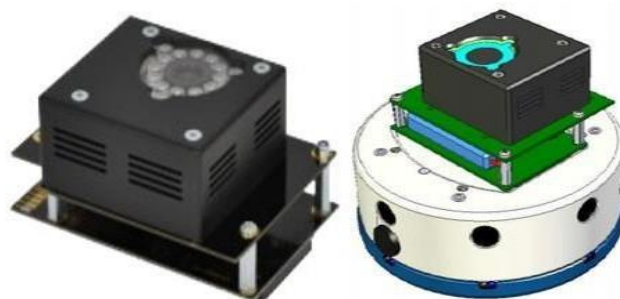


Figure II.21 : Le Stargazer

II.8. Conclusion

Dans ce chapitre on découvre les différents composants du robot mobile Khepera IV embarqué de nombreux capteurs et une puissance de calcul appréciable pour un robot mobile autonome, Et découvre aussi les fonctionnalités du robot (caméra couleur, Wi-Fi, Bluetooth, hôte USB, accéléromètre, gyroscope, microphone, haut-parleur, 3 LED RVB supérieures, odométrie améliorée et précision).

Ainsi que à la fin de ce chapitre, nous avons vu les différents moyens d'extension pour le robot mobile Khepera IV.

Chapitre III

III.1. Introduction

La robotique est un domaine multidisciplinaire qui a été introduit dans l'éducation à tous les niveaux au cours des dernières années. Le contrôle, l'informatique, l'électronique, la mécanique, la physique et les mathématiques sont des exemples de domaines de connaissances qui se limitent à la robotique. Cela le rend très polyvalent du point de vue pédagogique. Il peut être introduit à partir de faibles niveaux d'éducation (expériences simples de mouvement, virages, capteurs d'obstacles, capteurs de lumière, etc.) à des niveaux élevés (avec des expériences beaucoup plus difficiles telles que la localisation simultanée et la cartographie (SLAM), la navigation avec la vision par ordinateur, le contrôle de la formation, etc.).

Il y a quelques années, les robots étaient de grosses et coûteuses machines, qui dans de nombreux cas devaient être connectés à des ordinateurs avec des câbles. Cela a fait des robots restent loin des salles de classe. Aujourd'hui, il est facile de construire de petits robots mobiles avec actionneurs, capteurs, moteurs et systèmes embarqués à faible coût. Actuellement, les robots mobiles sont utilisés dans plusieurs cours et diplômes d'études supérieures en informatique, en génie du contrôle de l'automatisation, en génie mécanique et en mécatronique (Fabregas et al. (2016)).

Dans la plupart des domaines de recherche, les simulateurs sont très importants pour tester des théories, des idées, des conceptions avant de mettre en œuvre de véritables expériences. Mais dans le domaine de la robotique, ces simulateurs sont d'autant plus importants que les robots avancés sont encore chers et qu'aucun étudiant ne peut effectuer des expériences avec ce dispositif en même temps. Actuellement, il existe de nombreux simulateurs pour différents domaines de la robotique avec de très bons résultats. Par exemple : ARGoS (Pinciroli et al. (2011)), Webots (Michel (2004)), Gazebo (Koenig et Howard (2004)), RFCSIM (Fabregas et al. (2014)) et V-REP (Coppelia Robotics GmbH (2015)), pour ne citer que les plus utilisés.

Ces simulateurs ont plusieurs caractéristiques spécifiques, mais en général tous sont bien faits et ont une bonne performance. Certaines de ces plates-formes ont des licences qui peuvent être utilisées gratuitement à des fins éducatives. La plupart de ces simulateurs comprennent des modèles des robots les plus connus et commercialisés. Dans certains cas, les développements peuvent être exportés vers le robot réel pour les tests après l'utilisation du simulateur.

Le simulateur V-REP mérite une mention spéciale pour être l'un des plus largement utilisés à des fins pédagogiques aujourd'hui. Ce simulateur est un cadre polyvalent et évolutif pour créer des simulations 3D dans un laps de temps relativement court [33].

V-REP dispose d'un environnement de développement intégré (IDE) basé sur une architecture distribuée et pilotée par script : chaque objet de scène peut avoir un script intégré attaché, tous fonctionnant en même temps, d'une manière fileté ou non fileté. V-REP est livré avec un grand nombre d'exemples, de modèles de robots, de capteurs et d'actionneurs pour créer un monde virtuel et interagir avec lui en temps de fonctionnement. De nouveaux modèles peuvent être conçus et ajoutés à V-REP pour mettre en œuvre des expériences de simulation personnalisées.

III.2. Simulation Des Robots Mobiles :

La robotique mobile est une branche de la robotique qui traite avec des robots qui ne sont pas physiquement statiques pendant leurs opérations. Dans la littérature, un robot mobile est défini comme : « une machine automatique capable de locomotion ». Les élèves peuvent voir ce genre de robots comme des jouets attrayants, mais avec eux, ils peuvent analyser, tester et comprendre des concepts qui sont difficiles à expliquer d'un point de vue théorique.

Actuellement, il existe plusieurs simulateurs commerciaux et open source sur le marché, parmi lesquels on peut mentionner ARGoS, Gazebo, Webots, RFCSIM et V-REP. Dans la sous-section suivante, une comparaison entre les simulateurs les plus utilisés est présentée. Cette comparaison est basée sur Staranowicz et Mariottini (2011).

III.3. Comparaison des simulateurs :

ARGoS est un simulateur multirobots qui a été conçu pour simuler des essaims hétérogènes de robots en temps réel. Son approche est modulaire et permet à l'utilisateur d'ajouter des fonctionnalités personnalisées dans un environnement virtuel. Sa conception est assez différente des autres simulateurs. Sa caractéristique la plus distinctive est que le monde simulé en 3D peut être divisé en régions, et chaque région peut être affectée à un moteur physique différent. Tous les composants d'ARGoS sont des plugins (modèles de robots, capteurs, actionneurs, moteurs physiques, visualisations, etc.). Les utilisateurs peuvent étendre ou remplacer les plug-ins existants pour construire des robots et implémenter les simulations. ARGoS est disponible pour Linux et MacOS, mais pas pour Windows en raison des limitations du format DLL de Windows.

Webots est un environnement de développement utilisé pour modéliser et simuler des robots mobiles. Une liste complète de capteurs et d'actionneurs simulés est disponible pour construire les robots dans un monde virtuel. Leurs contrôleurs peuvent être programmés à l'intérieur de l'IDE ou avec des environnements de développement tiers. Les programmes de contrôleur peuvent éventuellement être transférés à des robots réels disponibles dans le commerce. Webots est utilisé dans de nombreuses universités et centres de recherche à travers le monde et ont été maintenus en permanence pendant plus de 19 ans [34] Webots est disponible pour Linux, MacOS et Windows.

RFCSIM est un simulateur développé dans EJS (Easy Java Simulations). Son but principal est de concevoir des expériences pour des systèmes multirobots pour l'éducation en ingénierie. Cette approche est axée sur le contrôle de position des robots à roues différentielles et des algorithmes d'évitement des obstacles. L'utilisateur peut modifier la loi de contrôle de position et l'algorithme d'évitement des obstacles. RFCSIM est disponible pour tous les formulaires plats qui supports et Java Virtual Machine [35]

Le tableau 1 montre une brève comparaison de certains des simulateurs ci-dessus. La prochaine sous-section présente le V-REP, le simulateur utilisé dans ce travail.

		<i>ARGoS</i>	<i>Webots</i>	<i>RFCSIM</i>	<i>VREP</i>
Usability		X	✓	✓	✓
Scalability		✓	✓	✓	✓
Visual	2D	✓	✓	✓	✓
	3D	✓	✓	X	✓
Program	C	X	✓	-	✓
	C++	✓	✓	-	✓
	Java	X	✓	✓	✓
Free License		X	✓	✓	✓

Tableau III.1 : Comparaison de simulateurs

III.4. V-REP

La plate-forme d'expérimentation de robots virtuels (V-REP) est un cadre polyvalent et évolutif pour développer des simulations 3D dans un laps de temps relativement court. Ce simulateur a été créé en 2010 et ont augmenté au cours des dernières années. Aujourd'hui, c'est

l'un des simulateurs les plus utilisés pour l'éducation dans le domaine de la robotique (avec une licence gratuite à des fins éducatives). Comme le disent ses auteurs : « V-REP est le couteau de l'armée suisse parmi les simulateurs de robots. »

V-REP dispose d'un environnement de développement intégré (IDE) basé sur une architecture de contrôle distribuée : chaque objet/modèle peut être contrôlé individuellement via un script intégré, un plug-in, un nœud robot operating system (ROS), un client API distant ou une solution personnalisée. Les contrôleurs peuvent être écrits dans différentes langues : C/C++, Python, Java, Lua, Matlab, Octave ou Urbi [36]

V-REP est livré avec un grand nombre d'exemples, de modèles de robots, de capteurs et d'actionneurs pour créer un monde virtuel et interagir avec lui en temps de fonctionnement. La figure 1 montre l'un de ces exemples : la ligne suivant et l'évitement des obstacles avec le célèbre robot Pioneer 3D-X.

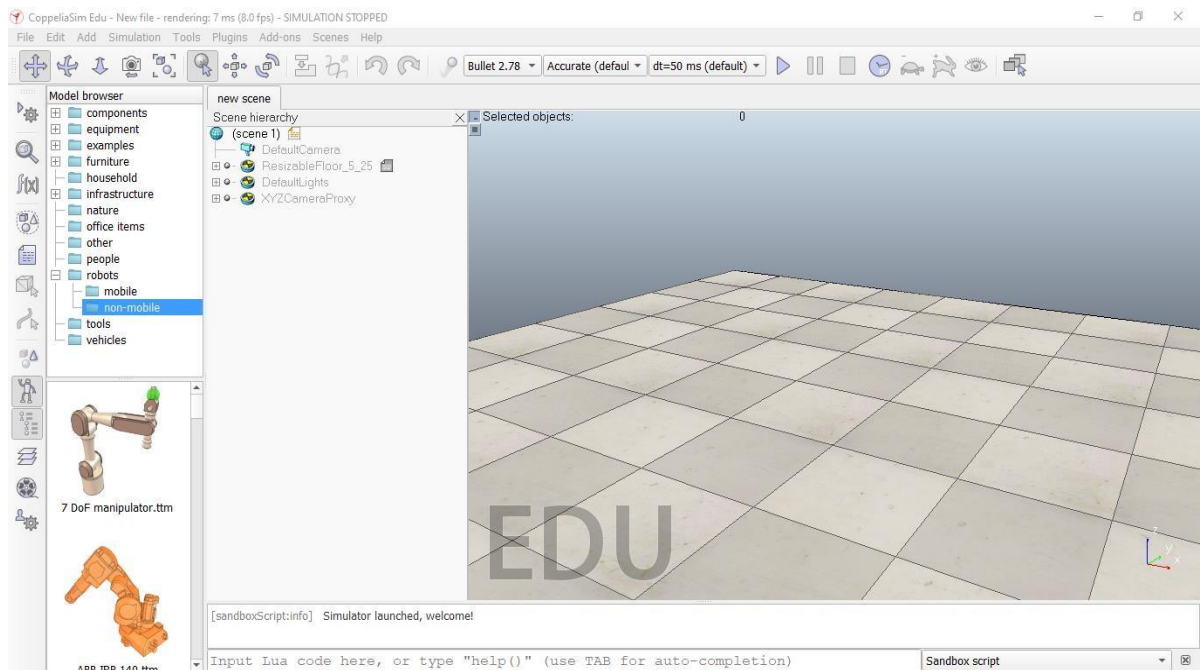


Figure III. 1 : Vue V-REP IDE

Le cadre permet également d'ajouter de nouveaux objets et de leur fournir des propriétés dynamiques pour concevoir et construire de nouveaux robots. En utilisant cet avantage, un modèle pour le robot Khepera IV ont été conçus et développés pour le simulateur V-REP. La section suivante explique les détails de ce modèle.

III.5 modèle Khepera VI V-REP

III.5.1 Khepera IV

Le robot Khepera IV (Figure III. 2) est le développement le plus récent de la société suisse K-Team (2015). Ce robot a été conçu à des fins de recherche et d'éducation. Il offre des communications wifi et bluetooth, un grand nombre de capteurs et d'actionneurs avancés tels que: infrarouges, ultrasons, accéléromètre, gyroscope, caméra, et une batterie de grande durée.



Figure III. 2: Robot à roues Khepera IV

Le robot peut être programmé à partir d'un système d'exploitation basé sur Linux et d'un langage C. Khepera IV a une configuration modulaire, avec une forme cylindrique pour minimiser les dommages et les effets négatifs dans ses pièces dans des conditions de collision. Khepera est un robot à roues différentielle dont le mouvement est basé sur deux roues conduites séparément placées de chaque côté de son corps. L'appareil peut être utilisé pour différentes expériences telles que : suivi de ligne, détection et évitement d'obstacles dans un environnement dynamique, traitement avancé du signal, contrôle de mouvement, communication sans fil, traitement d'image, contrôle de formation, et beaucoup d'autres sujets avancés

III.5.2. Conception graphique des éléments du modèle

Toutes les parties visuelles du robot (cas, base, roues, etc...) ont été modélisées à l'aide d'Autodesk Inventor [37] et assemblées en V-REP. Le prototype obtenu a été importé au format V-REP sous forme de format .stl tel qu'il est indiqué dans la (Figure III. 3). V-REP permet d'ajouter des propriétés physiques et de la dynamique à ce genre de prototypes.

La plupart des capteurs du robot ont été conçus sur la base du modèle V-REP Khepera III. Dans le cas de la caméra (qui n'est pas inclus dans la version précédente Khepera) a été configuré sur la base de la caméra existante de la bibliothèque V-REP.

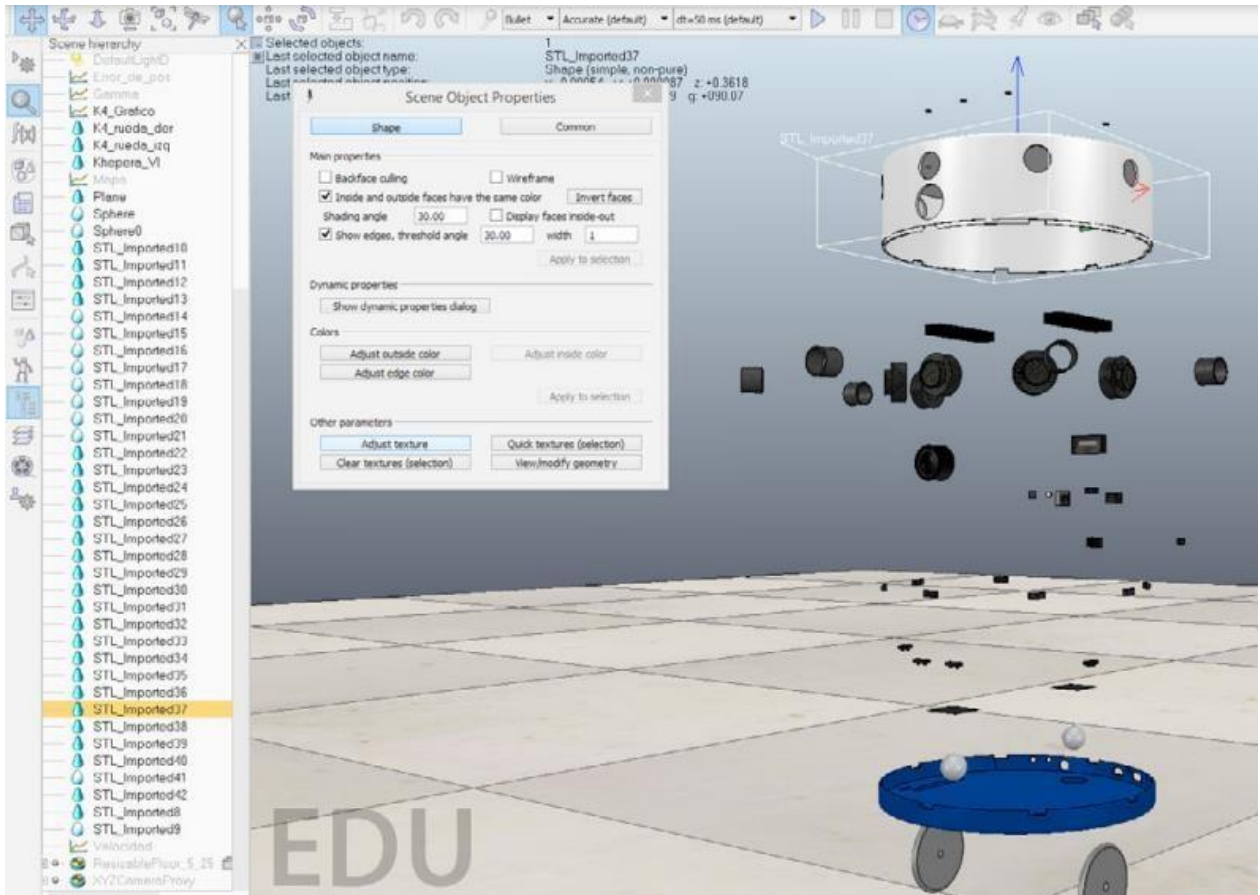


Figure III. 3 : Composants du modèle Khepera IV

III.5.3. Bibliothèque Khepera IV en V-REP

Le modèle Khepera IV comprend tous les capteurs et actionneurs du vrai robot. Ainsi, le modèle a cinq ultrasons capteurs, douze capteurs infrarouges (huit autour du boîtier et quatre dans le fond du robot pour effectuer la ligne suivante), une caméra couleur, deux moteurs à roues, et deux roues en roulettes.

Les capteurs ultrasoniques ont été configurés comme le vrai robot. Ces capteurs sont situés autour du boîtier et se séparent à 45 degrés. Ils permettent de détecter des objets de 250 à 2000 mm.

Des capteurs infrarouges sont également situés autour du boîtier à 45 degrés. Ils peuvent détecter des objets de 0 à 250 mm. La configuration des quatre capteurs infrarouges au bas du robot a été prise à partir du modèle de la Khepera III. Notez que ces capteurs infrarouges inférieurs sont spécialement conçus pour détecter les couleurs et effectuer la ligne suivante.

À l'avant du robot, il ya une caméra couleur qui peut être utilisé pour prendre des photos ou des films qui peuvent être traités à bord. La résolution des images capturées est de 752x480 pixels.

Le Khepera IV est un robot différentiel, équipé de deux roues qui sont entraînées par des moteurs DC avec encodeur et boîte de vitesses. Seules les deux roues sont incluses dans le modèle. La vitesse minimale contrôlée du robot est de 3 mm/s et la vitesse maximale est d'environ 813 mm/s. Dans le cas des roues en roulettes, les deux sont considérées comme des sphères fixes dans le modèle.

Notez que les capteurs de gyroscope et d'accéléromètre n'ont pas été inclus dans le modèle puisque cette information est donnée directement par le simulateur V-REP pour chaque modèle inclus.

La bibliothèque comprend également deux exemples simples d'utilisation du robot. Le premier exemple montre comment manipuler la vitesse du robot en utilisant les moteurs. Le deuxième exemple montre un moyen simple d'effectuer le contrôle de position ou la stabilisation des points avec le modèle Khepera IV. Les deux exemples sont illustrés dans les sections suivantes.

III.5.4 Exemple simple avec le modèle (commandes de vitesse)

Un exemple simple a été mis en œuvre pour tester le modèle. L'idée principale est de suivre une trajectoire prédéfinie indiquée à la (figure III.4). L'expérience consiste à envoyer des commandes (valeurs calculées empiriquement) aux moteurs et observer le mouvement du robot. Le code suivant permet au robot de suivre toute la trajectoire.

```

1 rightMotor = simGetObjectHandle('K4_rightWheel')
2 leftMotor = simGetObjectHandle('K4_leftWheel')
3 while(true) do
4     v=0
5     while(v<200) do —Line (positive acceleration)
6         speed=(0.05*v)
7         simSetJointTargetVelocity(leftMotor , speed)
8         simSetJointTargetVelocity(rightMotor , speed)
9         simSwitchThread(5)
10        v=v+1
11    end
12    v=0
13    while(v<106) do —Circle
14        simSetJointTargetVelocity(leftMotor ,10)
15        simSetJointTargetVelocity(rightMotor ,12)
16        v=v+1
17        simSwitchThread(5)
18    end
19    v=0
20    while(v<200) do —Line (negative acceleration)
21        speed=(0.05*(200-v))
22        simSetJointTargetVelocity(leftMotor , speed)
23        simSetJointTargetVelocity(rightMotor , speed)
24        v=v+1
25        simSwitchThread(5)
26    end
27 end

```

Les deux premières lignes obtiennent les poignées des moteurs. Ensuite, le code entre dans une boucle infinie qui est divisée en trois blocs de code. Le premier bloc envoie la même valeur de vitesse aux deux moteurs (lignes 7 et 8) augmentant cette valeur (ligne 6 et 10) à chaque étape de la boucle. Le résultat devrait être que le robot se déplace en ligne droite augmentant sa vitesse (segments A-B et D-E). Le deuxième bloc envoie deux valeurs de vitesse différentes aux moteurs (lignes 14 et 15). Le résultat devrait être que le robot tourne vers la gauche décrivant une trajectoire semi-circulaire (segments B-C et E-F). Le troisième bloc envoie la même valeur de vitesse aux moteurs (lignes 22, 23 et 24) mais la diminuant (segments C-D et F-A) à chaque étape de la boucle (ligne 21). (Figure III.4) montre les résultats de cette expérience.

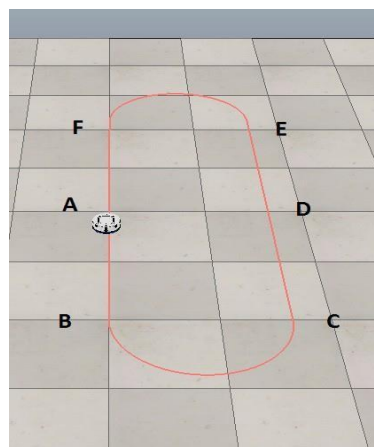


Figure III. 4 : Exemple d'utilisation de la bibliothèque

III.5.5. Contrôle de position ou « stabilisation des points »

Cette expérience consiste à conduire le robot mobile à roues à partir de tout point initial vers un point cible. Ce problème a été largement étudié au cours des dernières années, en raison de la cinématique modèle de ces robots peut sembler trompeusement simple, mais les contraintes non holonomiques posent des problèmes difficiles dans la conception de la loi sur le contrôle [38]

Cependant, ce genre de systèmes ne peut pas être asymptotiquement stabilisé par une rétroaction linéaire à l'état statique invariant depuis la linéarisation du système non holonomique sur n'importe quel point d'équilibre n'est pas asymptotiquement stable avec des commentaires d'état lisses [39] À surmonter ces limitations les techniques traditionnelles utilisent la loi sans contrôle utilise comme entrées la distance (d) et l'angle (α) entre le centre de masse du robot ($C(x_c; y_c)$) et le point cible ($P(x_p; y_p)$) qui sont calculés comme il est indiqué dans eq. (III.1) et (III.2)

$$d = \sqrt{(Y_p - Y_r)^2 + (X_p - X_r)^2} \quad (\text{III.1})$$

$$\alpha = \tan^{-1} \frac{(Y_p - Y_r)}{(X_p - X_r)} \quad (\text{III.2})$$

Avec ces deux entrées, la loi de contrôle calcule les vitesse (v) et la vitesse angulaire (ω) des robots à atteindre le point de destination tel qu'il est indiqué dans eq. (III.3) et (III.4)

$$v = \begin{cases} V_{max}, & \text{if } d > K_r \\ V_{max}/K_r, & \text{if } d \leq 0 \end{cases} \quad (\text{III.3})$$

$$\omega = \omega_{max} \sin(\alpha - \theta) \quad (\text{III.4})$$

Lorsque V_{max} est la vitesse linéaire maximale, k_r est le rayon d'une zone d'amarrage (autour du point cible) et ω_{max} est la vitesse angulaire maximale du robot. la segment de code suivant représente l'implémentation de cette loi de contrôle dans V-REP.

```

1 pos = simGetObjectPosition (bodyElements, -1)
2 target = simGetObjectPosition (Target, -1)
3 robotOr = simGetObjectOrientation (bodyElements, -1)
4 while (d < 0.01) do
5   gamma = robotOr [3]
6   Xr = pos [1]      --robot position
7   Yr = pos [2]
8   Xp = target [1]  --target point
9   Yp = target [2]
10  d = math.sqrt (((Xp-Xr)^2) + ((Yp-Yr)^2)) -- distance
11  alpha = math.atan2 (Yp-Yr, Xp-Xr)      --angle
12  W = (Wmax * math.sin (alpha - theta))  --angular V
13  if (d > Kr) then                        --linear V
14    V = Vmax
15  else
16    V = d * (Vmax / Kr)
17  end
18  Vr = (2 * V + W * L) / 2      --right Velocity
19  Vl = (2 * V - W * L) / 2      --left Velocity
20  simSetJointTargetVelocity (leftMotor, Vl)
21  simSetJointTargetVelocity (rightMotor, Vr)
22 end

```

La position du robot (X_r ; Y_r), son orientation (θ) et les coordonnées du point cible $P(X_p$; $Y_p)$ sont obtenus des lignes 1 à 8. Alors que les lignes 9 et 10 calculent d et l'angle (α) du robot au point cible représenté dans les points 1 et 2. Le linéaire (v) et les vitesses angulaires (ω) (représentées dans les eq.(III.3) et (III.4) sont obtenus à partir des lignes 11 à 16. Les lignes 17 et 18 calculent vitesses différentielles pour chaque roue (droite et gauche) et lignes 19 et 20 envoient ces vitesses aux moteurs de la Robot.

III.5.6. Expérience de contrôle de position

Cette expérience est de conduire le robot mobile à roues de sa position actuelle à un point cible prédéfini.

Ce problème a été largement étudié au cours des années, en raison du modèle cinématique de ces robots peuvent sembler trompeusement simples, mais des contraintes non holonomiques mettre en place des problèmes difficiles dans la conception de la contrôle (Fabregas et coll. (2016)). (Figure III. 5) montre schéma bloc le de cette expérience.

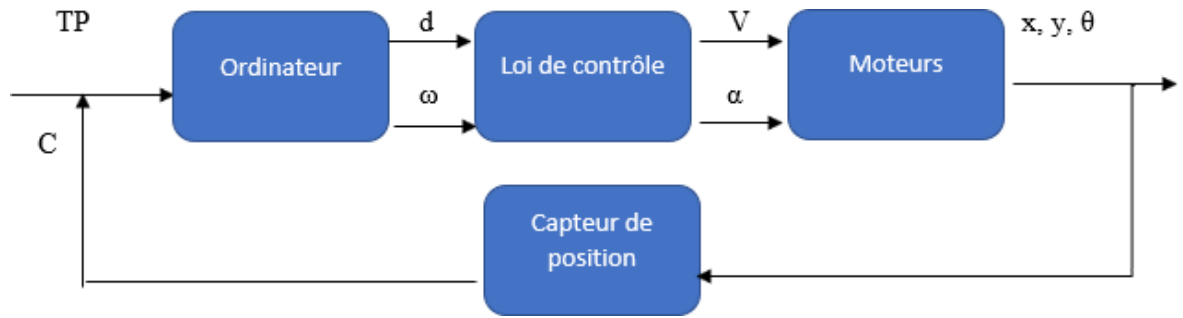


Figure III. 5 : schéma bloc d'un asservissement de contrôle de position

Le robot tente de minimiser son erreur d'orientation, $\theta_e = \alpha - \theta$, où α est l'angle actuel au point cible et θ est l'orientation actuelle du robot. En même temps, le robot tente de réduire la distance au point cible ($d \approx 0$).

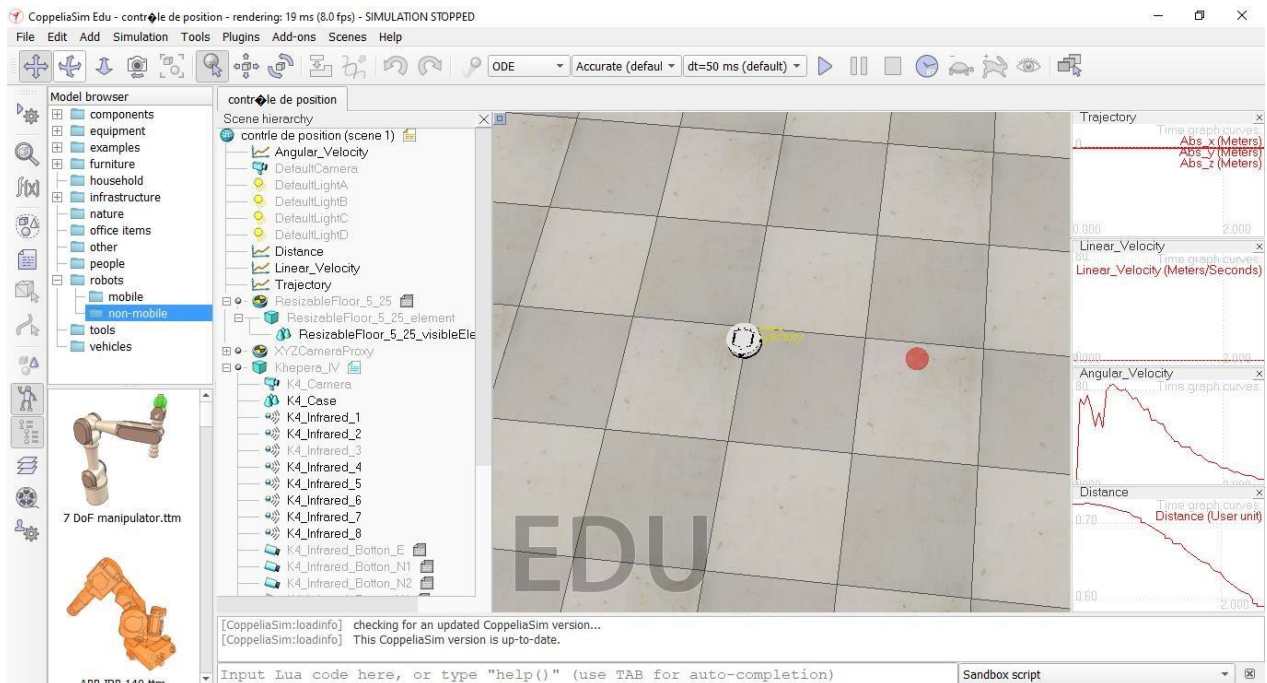


Figure III. 6 : Expérience de contrôle de position

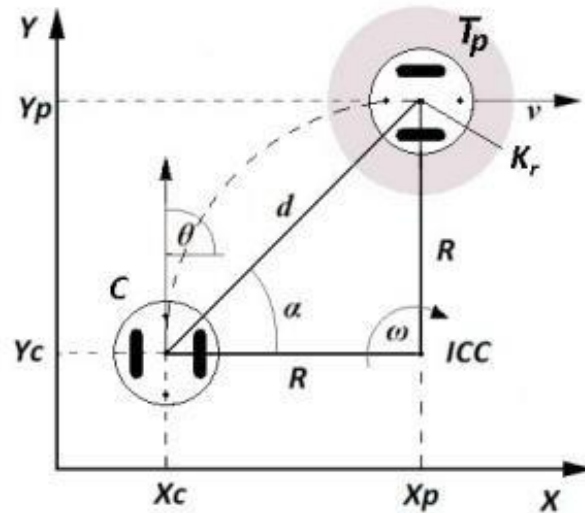


Figure III. 7 : Trajectoire de mouvement

Les équations (III.1) et (III.2) sont implémentées dans le bloc Ordinateur; en utilisant comme référence le point cible (T_p) et la position actuelle du robot (C).

$$d = \sqrt{(Y_p - Y_r)^2 + (X_p - X_r)^2} \quad (\text{III.1})$$

$$\alpha = \tan^{-1} \frac{(Y_p - Y_r)}{(X_p - X_r)} \quad (\text{III.2})$$

Avec les sorties du bloc Calculer le contrôle de **Loi de Bloc** calcule la vitesse linéaire (v) et l'angulaire vitesse (ω) du robot tel qu'il est montré dans les équations (III.3) et (III.4)

(Villela et coll. (2004)).

$$v = \begin{cases} V_{max}, & \text{if } d > K_r \\ V_{max} / K_r, & \text{if } d \leq 0 \end{cases} \quad (\text{III.3})$$

$$\omega = \omega_{max} \sin(\alpha - \theta) \quad (\text{III.4})$$

où V_{max} est la vitesse linéaire maximale, k_r est le rayon d'une zone d'amarrage (autour du point cible) et ω_{max} est la vitesse angulaire maximale du robot.

Le segment de code suivant représente l'implémentation de la position de fonction Control dans la bibliothèque KH4VREP.

Notez que cette fonction est appelée dans la ligne 9 de la fonction dans l'exemple précédent.

```

1 positionControl=function(xp,yp,xc,yc,theta)
2   d = math.sqrt(((xp-xc)^2)+((yp-yc)^2))
3   alpha = math.atan2(yp-yc,xp-xc) ---Target angle
4   Oc = alpha-theta ---Error angle
5   if(d>=0.05) then
6     w=Wmax*math.sin(Oc) ---Angular velocity
7     v=(vmax/kr)*d ---Linear velocity
8     if(v>vmax) then ---Linear velocity saturation
9       v=vmax
10    end
11  else v=0, w=0
12  end
13  return v,w ---Values returned
14 end

```

La ligne 1 est la définition de la fonction position Control qui reçoit comme paramètres les coordonnées du courant point cible (X_p ; Y_p) et la position actuelle (X_c ; Y_c) et orientation (θ) du robot. Les lignes 2 et 3 calculent équations (III.1) et (III.2). La ligne 4 obtient l'angle d'orientation Erreur. Les lignes 6 calculent l'équation (III.4) et la ligne 7 calcule équation (III.3). Les lignes 8 à 10 sont la saturation de la vitesse linéaire calculée (v). Les lignes 12 et 13 vitesses à zéro si le robot est très proche de la destination point (moins de 0,05 m). La fonction renvoie les valeurs des vitesses de la ligne 15.

III.5.7. Expérience d'évitement des obstacles

Le problème de l'évitement des obstacles a été largement étudié en utilisant différentes techniques. Notez que ce problème est une prochaine étape du contrôle de position. Le robot doit être en mesure d'éviter les obstacles qui apparaissent sur son chemin vers le Destination. Pour ce faire, le bloc d'évitement est ajouté après le bloc de la loi de contrôle. Ce bloc calcule les nouvelles vitesses (v' et ω') si les capteurs détectent les obstacles. En l'absence des obstacles $v'=v$ et $\omega'=\omega$. (Figure III. 8) montre le Schéma bloc de cette expérience.

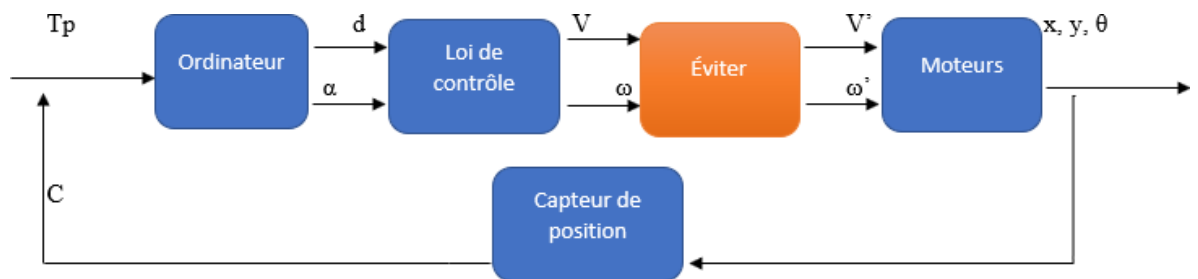


Figure III. 8 : Schéma bloc de contrôle avec évitement d'obstacle

Le bloc d'évitement peut être implémenté avec différents algorithmes en fonction des capteurs d'obstacles et de ses emplacements sur le robot. Dans ce cas, pour tester la

bibliothèque deux algorithmes ont été mis en œuvre : Braitenberg [40] et (VFH) Vector Field Histogramme[41]

Algorithme de Braitenberg. Cet algorithme crée une matrice qui convertit les entrées du capteur en vitesses des moteurs. Cette matrice est un tableau bidimensionnel avec le nombre de colonnes correspondant aux huit capteurs et aux lignes correspondant aux deux moteurs. Équation (III.5) représente matrice, où W_{ls1} est le poids du capteur S_1 dans la vitesse du moteur gauche. Équation (III.6) représente les valeurs des capteurs à chaque fois.

$$W = \begin{pmatrix} 1 & W_{ls2} & \dots & \dots & W_{lss} \\ W_{rs1} & W_{rs2} & \dots & \dots & W_{rss} \end{pmatrix} \quad (III.5)$$

$$S = (S_1 \ S_2 \ \dots \ S_8)^2 \quad (III.6)$$

Avec ces matrices, les vitesses de chaque moteur sont calculé comme indiqué dans l'équation (III.7), où (S_{max}) est le valeur maximale des capteurs de proximité si aucun obstacle n'est Détecté.

$$V_{L,R} = W * (1 - (S/S_{max})) \quad (III.7)$$

Les vitesses v' et ω' sont calculées à partir de V_R et V_L à l'aide les équations de modèle différentiel. Le segment de code suivant montre l'implémentation de cet algorithme.

```

1 braitenbergControl=function()
2 if (S[1]+S[2]+S[3]+S[4]+S[5] .. --Obstacle detected
3   ..+S[6]+S[7]+S[8]<=Smax*8) then
4   for i=1,8,1 do
5     Vl=Vl+WLS[i]*(1-(S[i]/Smax))
6     Vr=Vr+WRS[i]*(1-(S[i]/Smax))
7   end
8   v=(Vr+Vl)/2
9   w=(Vr-Vl)/L
10  end
11  return v,w
12 end

```

La fonction Braitenberg est appelée après la position de fonction Control dans le thread principal Fonction Ligne 1 est la définition de cette fonction. La ligne 2 demande la détection des obstacles. Si l'un des 8 capteurs n'a pas le maximum valeur, c'est parce qu'un obstacle a été détecté. Lignes 5 et 6 appliquer des équations (III.5), (III.6) et (III.7). Lignes 8 et 9 calculer v et ω à partir du modèle différentiel. Ligne 11 renvoie les valeurs des vitesses calculées

algorithme VFH. Cette méthode permet une mouvement du robot évitant les obstacles dans un Environnement. La vitesse est la plus maximale en l'absence de Obstacles. Le robot

tente de maintenir cette vitesse pendant la trajectoire à moins d'être forcé de ralentir par le VFH algorithm à la vitesse instantanée V et à change

$$V = V'(1 - \omega / \omega_{max}) \quad (III.8)$$

$$\omega = \omega_{max} \sin(\alpha - \theta_e) \quad (III.9)$$

Où ω_{max} est la vitesse angulaire maximale admissible pour le robot qui sera atteint lorsque l'erreur d'orientation, $\theta_e = \pm 90^\circ$, et v est défini dans les équations (III.10) et (III.11).

$$V' = V_{max} (1 - h_c' / h_m) \quad (III.10)$$

$$h_c' = \min(h_c, h_m) \quad (III.11)$$

Où v_{max} est la vitesse linéaire maximale, h_m est une constante empiriquement déterminée à provoquer une réduction de la vitesse en présence d'obstacles, et h_c' est la valeur de L'histogramme de la distance entre le robot et le Obstacles. Si $h_c = 0$ puis un obstacle se trouve devant le robot.

Grandes valeurs de h_c signifie qu'un obstacle plus important nous attend ou obstacle est très proche du robot. Dans les deux cas, un changement de la direction est nécessaire. Cela signifie qu'une réduction de la vitesse linéaire est nécessaire pour tourner vers la nouvelle direction.

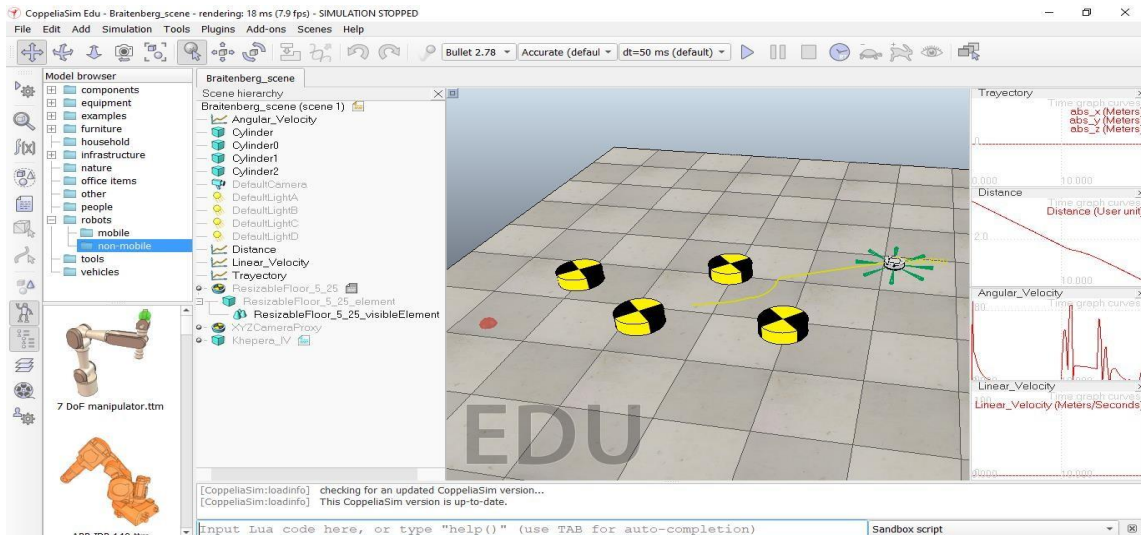


Figure III. 9 : Expérience d'évitement des obstacles

III.5.8. Suivi du chemin d'accès

Le problème suivant le chemin est similaire à la trajectoire problème de suivi. Mais dans ce cas, le robot doit suivre la trajectoire géométrique en tenant compte du temps arriver à chaque point. Une approche de la piste suivante problème est d'introduire une variable de chemin comme un « ntr » contrôler l'entrée et essayer de faire zéro l'erreur entre la trajectoire du robot et du chemin [42]

Dans V-REP, le suivi de trajectoire peuvent être sélectionnés comme propriétés d'un objet mobile qui doit atteindre une cible. Mais dans ces cas, la mise en œuvre a été faite par code source dans le modèle pour le tester avec ces expériences «complexes».

(Figure III. 10) montre les résultats de l'expérience suivant le chemin d'accès pour une trajectoire circulaire. Le panneau principal montre l'arène avec le chemin et la trace du robot. Comme dans le précédent par exemple, au bas de la figure, les graphiques montrent les valeurs de la position du robot, de la trajectoire et d'autres informations d'intérêt pour cette expérience. Comme on peut le voir, le modèle fonctionne très bien pour cette expérience comme prévu.

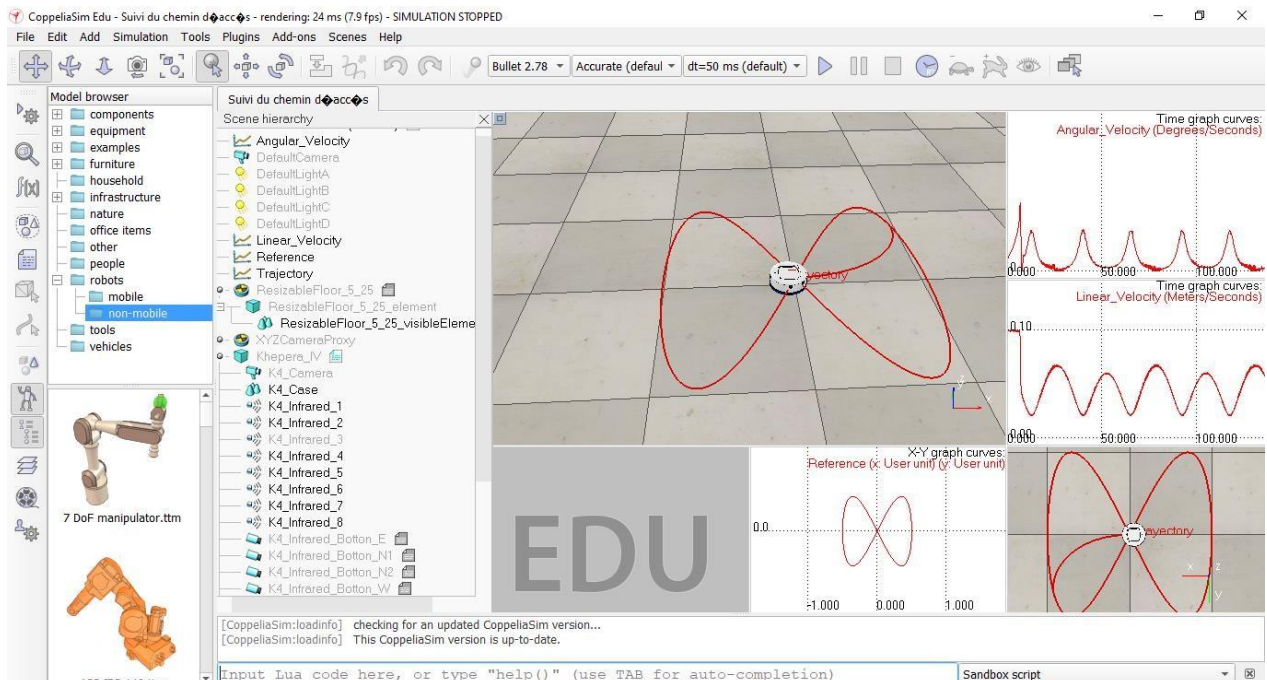


Figure III. 10 : Expérience de Suivi du chemin d'accès

4- CONCLUSION

Dans ce chapitre, la bibliothèque KH4VREP pour simulateur V-REP a été présentée. La bibliothèque est une encapsulation du modèle de la Khepera IV et comprend des méthodes intégrées pour manipuler le robot, et aussi un ensemble de simulations pour montrer de nombreux problèmes de contrôle avec les robots mobiles. Le résultat de ce développement est un outil prêt à l'emploi qui permet la mise en œuvre d'expériences avec ces robots dans un environnement interactif et attrayant. La bibliothèque permet tester et comprendre les concepts théoriques dans une dynamique afin d'enseigner le contrôle de l'ingénierie.

Dans un proche avenir, la bibliothèque fera partie d'un projet pilote plate-forme qui a été développée dans nos laboratoires.

L'objectif principal des travaux futurs sera de concevoir algorithmes de contrôle avancés à l'aide de la bibliothèque et l'environnement de simulation. Entre autres, des expériences multi agents avec problème de contrôle du consensus et communications entre les robots pourraient être incluses dans la bibliothèque.

*C*onclusion

*G*énérale

Conclusion Général

Le problème de la navigation des robots mobiles dans un environnement incertain, est l'un des problèmes les plus importants et les plus nouveaux dans les robots mobiles. De nombreuses recherches ont été faites pour résoudre ces problèmes.

dans ce travail, nous avons fourni une simulation de la façon dont le robot se déplace et surmonte les obstacles et les simulations se sont concentrées principalement sur

- Comment le robot khepera 4 se déplace d'un endroit à l'autre
- Comment le robot khepera 4 suit un chemin prédéfini spécifique
- Comment le robot khepera 4 surmonter certains des obstacles auxquels il fait face dans ses mouvements

En fin de compte, nous espérons et nous avons pensé que notre travail serait amélioré à l'avenir par d'autres étudiants, et nous suggérons en tant que visionnaire:

- Simulation d'une partie du travail effectué par le robot.
- Simulation à travers d'autres programmes de simulation.

Webgraphie

- [I] <https://my.editions-ue.com/catalog/details/store/gb/book/978-620-2-26626-0/des-outils-de-1%E2%80%99intelligence-artificielle-dans-la-robotique-mobile?fbclid=IwAR1iGqN2oqVSwbYDgeTmopA13WyqgfVanJQ5yaMTMcZDewaQWdwg3ImA7qM>
- [II] <https://www.generationrobots.com/fr/402241-robot-mobile-khepera-iv.html>
- [2] <http://www.futura-sciences.com/definitions/robotique>
- [4] <http://www.futura-sciences.com/definitions/automate>
- [5] <http://www.cinematheque.fr/objet/347.html>
- [6] <http://eduscol.education.fr/Robots>, page 4.
- [7] <http://www.futura-sciences.com/definitions/informatique/intelligence-artificielle>
- [8] <https://perso.univ-lyon2.fr/~mollon/EO-web/12-13/Virgo/definition%20et%20historique%20de%20la%20robotique/definition%20et%20historique%20de%20la%20robotique.html>
- [9] https://www.ensta-bretagne.fr/jaulin/mastersds_cours_robot_boimond.pdf, pages 3-4-5-6.
- [10] https://www.ensta-bretagne.fr/jaulin/mastersds_cours_robot_boimond.pdf, page
- [36] Coppelia Robotics GmbH (2015). V-REP simulator. URL <http://www.coppeliarobotics.com>.
- [37] Autodesk Inc. (2015). Autodesk Inventor. URL <http://www.autodesk.com/products/inventor/>.

Bibliographie

- [1] Trucs & Astuces sous Windows, gualbert adohoueto, page 428.
- [3] Le français dans le monde, numéros 3 à 4 ; numéro 322 à 324, cle international 2002, l'université du Michigan, 24 Mars 2008, page 1.
- [11] Y. Tigli, Vers une Architecture de contrôle pour Robot Mobile orientée Comportement, SMACH. PhD thesis, Université de Nice - Sophia Antipolis, 1996.
- [12] B. Haibe-Kains, Apprentissage d'une Tâche de Contrôle pour un Robot Mobile en LEGO Mindstorms. Thèse de doctorat, université libre de Bruxelles, 2003.
- [13] D. Filliat, Cartographie et estimation globale de la position pour un robot mobile autonome, Thèse de doctorat de
- [14] Éric Beaudry, Planification de tâches pour un robot mobile autonome, Thèse de Maître de sciences, Université de Sherbrooke, Québec, Canada. Août 2006.
- [15] F. Giamarchi. Petits robots mobiles. Edition Dunod, 2006.
- [16] S. Lens. Locomotion d'un robot mobile. Thèse de doctorat, Faculté des Sciences Appliquées, Université de Liege, 2008.
- [17] Robert Holmberg, Oussama Khatib, development and control of a holonomic mobile robot for mobile manipulation tasks, International Journal of Robotics Research, vol. 19, N°11, november 2000.
- [18] L. Amouri, C. Novales, M. Njah, M. Jallouli et G. Poisson, DVZ-based Obstacle Avoidance Control of a Wheelchair Mobile Robot. IEEE International Conference on Mechatronics (ICM), Istanbul, Turkey, 2011.
- [19] H. Moravec, Stanford card and the CMU rover, proceeding of IEEE vol 71, N° 7, pp 872-884, juillet 1983.
- [20] Pierre Arnaud, Des moutons et des robots : architecture de contrôle réactive et déplacements collectifs de robots, PPUR presses polytechnique, 2000.
- [21] A. Dalgalarondo. Intégration de la fonction perception dans une architecture de contrôle de robot mobile autonome. Thèse doctorat, université de paris sud centre d'ORSAY. France, 2001.
- [22] D. Bonnafous, Exécution réactive de trajectoires pour robots mobiles non-holonomes. Thèse de doctorat. Institut National Polytechnique de Toulouse, Systèmes Automatiques. France, 2003.
- [23] Bernard BAYLE, Robotique mobile, Thèse de doctorat, Ecole Nationale

Supérieure de Physique de Strasbourg, Université de Strasbourg 2008.

- [24] O. Aycard, Architecture de contrôle pour robot mobile en environnement intérieur structuré. Thèse de doctorat, université Henri Poincaré-Nancy 1, LORIA, Nancy, Juin 1998.
- [25] L. Adouane, Architecture de contrôle comportementale et réactive pour la coopération d'un groupe de robots mobiles. Thèse de doctorat, Laboratoire d'automatique de Besançon UMR (CNRS 6589) France, Avril 2005.
- [26] R.A Brooks, A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, Vol. 2, No.1, pp 14– 23, 1986.
- [27] R.C.Arkin, Formation Behaviors, behavior-Base robotics, The MIT Press. Cambridge, MA, USA, 1998.
- [28] L. Steels. When are robots intelligent autonomous agents?, *journal of robotics*, Volume 15 pp 3-9, 1995.
- [29] Arab Ali Cherif. Collective behavior for a micro-colony of robots. In *Proceedings of the Second European Conference on Artificial Life (ECAL'93)*, pages 1–10, Brussels, Belgium, May 1993. 1
- [30] Intergiciels pour systèmes multi-robots : état de l'art Stefan-Gabriel Chitic, Julien Ponge, Olivier Simonin, Université de Lyon.
- [31] David Popenoe. *Sociology* (11th Edition). Prentice Hall, 1999. 7
- [32] Fabregas, E., Farias, G., Dormido-Canto, S., Guinaldo, M., Sánchez, J., and
- [33] Rohmer, E., Singh, S., and Freese, M. (2013). V-REP: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1321–1326.
- [34] Guyot, L., Heiniger, N., Michel, O., and Rohrer, F. (2011). Teaching robotics with an open curriculum based on the e-puck robot, simulations and competitions. In *Proceedings of the 2nd International Conference on Robotics in Education*. Vienna, Austria.
- [35] Fabregas, E., Farias, G., Dormido-Canto, S., and Dormido, S. (2014). RFCSIM simulador interactivo de robótica móvil para control de formación con evitación de obstáculos. In *XVI Congreso Latinoamericano de Control Automático*, Cancún, Quintana Roo, México
- [38] Kühne, F., Lages, W.F., and da Silva Jr, J.M.G. (2005). Point stabilization of mobile robots with nonlinear model predictive control. In *Mechatronics and Automation, 2005 IEEE In*

- [39] KyuCheol, P., Hakyoung, C., and Jang Gyu, L. (2000). Point stabilization of mobile robots via state space exact feedback linearization. *Robotics and ComputerIntegrated Manufacturing*, 16(5), 353–363.
- [40] Yang, X., Patel, R.V., and Moallem, M. (2006). A fuzzy braitenberg navigation strategy for differential drive mobile robots. *Journal of Intelligent and Robotic Systems*, 47(2), 101–124.
- [41] Ulrich, I. and Borenstein, J. (2000). VFH*: Local obstacle avoidance with look-ahead verification. In *ICRA*, 2505– 2511.
- [42] Encarnacao, P. and Pascoal, A. (2000). 3D path following for autonomous underwater vehicle. In *Proc. 39th IEEE Conference on Decision and Control*.