

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
UNIVERSITE MOULOU MAMMERI DE TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire

Présenté pour l'obtention du **diplôme** de **Master Professionnel**

En : Génie Electrique

Spécialité : Electronique industrielle

Par : Mr. DEMICHE Amazigh & Mr. DJEBBAB Juba

Thème

Planification automatique de trajectoire d'un fauteuil roulant
intelligent

Soutenu publiquement, le 26/09/2018, devant le jury composé de :

Mr. Y. ATTAF	M.A.B	à l'UMMTO	Président
Mr. K. ABAINIA	M.A.B	à l'UMMTO	Promoteur
Mr. R. BOUCHEBBAT	M.A.B	à l'UMMTO	Examineur

Remerciements

En préambule à ce mémoire

Nous remercions le dieu qui nous a aidé et nous a donné la patience et le courage durant ces longues années d'étude.

Nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Nous tenant à remercier sincèrement Mr ABAINIA, en tant que promoteur,

Aux membres de jury

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours soutenue et encouragée au cours de la réalisation de ce mémoire.

Merci à tous

Dédicace

Ma Mère, Mon Père

Affable, honorable, aimable : vous représentez pour moi, le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

Soyez sûrs que je continuerai mon chemin.

Je vous dédie ce travail en témoignage de mon profond amour.

Puisse Dieu, le tout puissant, vous préserver et vous accorder santé, longue vie et bonheur.

A mes sœurs

A mon frère

En témoignage de l'attachement, de l'amour et de

L'affection que je porte pour vous.

A tous mes amis de proches.

DEMICHE Amazigh

Dédicace

*Je Tiens tout d'abord à remercier le bon Dieu de m'avoir aidé à
réaliser ce mémoire*

Que je dédie à :

*A la prunelle de mes yeux, celle qui m'a soutenu surtout avec son
Dou'a*

*jour et nuit pour qu'elle me voit au sommet et comme une étoile, à la
source de la Douceur.*

A toi ma très chère mère.

*A la personne qui a sacrifié sa vie pour moi, et qui a pris le défi pour
mes études*

Et mas éclairé le chemin de ma réussite.

A toi mon très cher Père.

A la mémoire de mes grands-parents.

*A mes chers frères et mes chères sœurs Qui sont toujours à mes
côtés et qui m'ont soutenu durant mon parcours universitaire*

A toute ma famille

A tous mes ami(e)s

A mon binôme et sa famille

DJEBBAB Juba

Résumé

La robotique regroupe trois disciplines telles que la mécanique, l'électronique et l'informatique, d'où elle est devenue un domaine plus vaste et très utile dans la vie humaine afin de faciliter les tâches difficiles de l'homme. Depuis quelques années, un intérêt croissant est porté, au sein de la communauté robotique et au développement des systèmes intelligents autonomes dans le cadre de la robotique mobile.

C'est dans ce cadre que repose notre étude qui consiste à réaliser un fauteuil roulant électrique en mode autonome (ou intelligent) pour les personnes ayant un handicap physique lourd. L'intelligence d'un fauteuil roulant réside dans sa capacité à percevoir l'environnement grâce à différents types de capteurs. Pour participer à une perspective de développement, nous nous sommes intéressés à l'optimisation de la planification de chemin et de mouvement du fauteuil roulant électrique en utilisant un algorithme bio-inspiré, i.e. optimisation par essaim de particules.

L'objectif du fauteuil roulant électrique intelligent est d'atteindre une destination qui lui a été assignée en évitant la collision avec les obstacles statique et les obstacles dynamiques dans l'environnement où il agisse physiquement.

Abstract

Robotics regroups together three disciplines such as mechanics, electronics and computer science, and has become widespread and more useful field in human life (facilitate the difficult tasks of man). In the last decades, there has been a great interest in the robotics community and the development of autonomous intelligent systems in the context of mobile robotics.

It is in this regards, we conduct a study to make an electric autonomous wheelchair (or intelligent) that is dedicated to people with a heavy physical disability. The wheelchair intelligence lies in the ability to perceive the environment based on different sensors. To contribute in developing a perspective, we focus on the path planning of electric wheelchair using a bio-inspired algorithm, i.e. Particle Swarm Optimization.

The goal of the autonomous electric wheelchair is to reach a target position by avoiding static and dynamic obstacles in the environment.

Mot clé: la robotique robot mobile, la navigation, les fauteuils roulants électriques intelligents, Algorithme PSO, Evitement des obstacles.

Table des Matières

Liste des figures	
Introduction générale.....	1
Chapitre 1 : Généralité sur la robotique mobile	
1-Introduction.....	3
2-Définition.....	3
3-Evolution de la robotique.....	4
4-Différents types de robots mobiles.....	8
4.1- Robots à roues.....	8
4.2- Robots à chenilles.....	9
4.3- Robots marcheurs.....	9
4.4- Robots rampants.....	10
5- Notion de navigation.....	11
5.1-Définition.....	11
5.2-Méthodes de navigation.....	11
5.2.1-Navigation réactive.....	11
5.2.2-Navigation par asservissement.....	12
5.2.3 Navigation par carte.....	12
6- Architectures de contrôle.....	13
6.1-Contrôleur hiérarchique.....	13
6.2-Contrôleur réactif.....	15
6.3-Contrôleur hybride.....	16
7- Collaboration en robotique.....	16
7-1- Collaboration homme-robot.....	17
7-2- Collaboration entre les robots.....	17
8- Conclusion.....	18

Chapitre 2 : Fauteuils roulants électriques

1-Introduction aux Fauteuils roulants électriques	20
2- Généralité	20
2.1- Origines du fauteuil roulant électrique.....	20
2.2- Descriptif du fauteuil roulant électrique	20
3- Fauteuils roulants intelligents	21
3.1- Projet de VAHM	21
3.2- Projet WAD « WHEELCHAIR ATTRACTORDYNAMICS».....	22
4- Capteurs.....	22
4.1- Capteurs proprioceptifs	23
4.2- Télémètres	23
a-Télémètres à ultrason.....	23
b-Télémètres à infrarouge.....	24
c-Télémètres laser	25
4.2- Capteurs photographiques.....	26
4.3- Caméras.....	26
a- Caméras stéréoscopiques.....	27
b- Caméras panoramiques	27
5-Conception et réalisation d'une carte de commande	27
a- Choix du fauteuil roulant.....	27
b-Choix des capteurs	29
6-Transmission des données.....	30
6.1-Le module radio nRF24L01	30
6.2-Émission et réception	30
7- Commande des moteurs du fauteuil roulant.....	31
7.1- Alimentation de l'Arduino	32
7.2-Récupération des informations moteurs.....	33

Chapitre 3 : Planification de chemin du fauteuil

1- Algorithme d'optimisation par essaim de particules (PSO).....	36
1.1- Principe	37
1.2- Paramètres de l'algorithme	38
1.2.1- La dimension du problème.....	38
1.2.2- Le nombre de particules	38
1.2.3- La notion de voisinage	38
1.2.4- La vitesse maximale et la vitesse minimale	39
1.2.5- Les Coefficients de confiance	39
1.2.6- Le facteur d'inertie.....	39
2- Pseudo code.....	40
3- Evitement d'obstacles	41
3.1- Evitement des obstacles statiques	43
3.2- Evitement des obstacles dynamiques	44
4- Notre approche	46
5- Simulation de l'algorithme.....	47
6- Conclusion.....	49
Conclusion générale.....	51

Table des figures

Figure 1.1. Le robot mobile Mars PathFinder développé par l'agence spatiale NASA	3
Figure 1.2. L'automate Jacquemart	4
Figure 1.3. Le robot musicien Jaquet-Droz	5
Figure 1.4. Le premier robot humanoïde de Léonard De Vinci	5
Figure 1.5. Shakey premier robot mobile contrôlé par ordinateur	6
Figure 1.6. Wabot-1 le premier robot humanoïde à marche bipède	6
Figure 1.7. Le robot humanoïde de Honda ASIMO	7
Figure 1.8. Le robot Sophia	8
Figure 1.9. Exemples de robots mobiles à roues	9
Figure 1.10. Exemples de robots mobiles à chenilles	9
Figure 1.11. Exemples de robots marcheurs	10
Figure 1.12. Exemples de robots rampants	10
Figure 1.13. Architecture réactive	11
Figure 1.14. Schéma du contrôleur hiérarchique.....	14
Figure 1.15. Exemple de collaboration homme-robot.....	17
Figure 1.16. Exemple de collaboration entre plusieurs robots pour transporter un boîtier	18
Figure 2.1. Les différents types de fauteuils (A: Fauteuil d'intérieur, B: Fauteuil d'extérieur, C: Fauteuil verticalisateur, D: Fauteuil à hauteur variable).....	20
Figure 2.2. Le projet de VAHM	21
Figure 2.3. PROJET WAD « WHEELCHAIR ATTRACTOR DYNAMICS »	22
Figure 2.4. Principe général de l'odométrie.....	23
Figure 2.5. Principe du télémètre à ultrasons et exemple de télémètre réel	24
Figure 2.6. Principe du télémètre infrarouge à triangulation et exemple de télémètre réel	25

Figure 2.7. Illustration du principe de fonctionnement d'un télémètre Laser et un exemple de Télémètre Laser à balayage, fournissant 720 mesure réparties sur 360 degrés, à 5Hz (marque LBEO).....	25
Figure 2.8. Les capteurs photographiques (A: capteur CCD, B: capteur CMOS)	26
Figure 2.9. Exemple de données fournies par des caméras stéréoscopiques.....	27
Figure 2.10. Principe des caméras panoramiques catadioptriques	28
Figure 2.11. Caractéristiques techniques du notre fauteuil roulant électrique	28
Figure 2.12. Synoptique du fauteuil roulant éclectique.....	29
Figure 2.13. Module nRF24L01+ chinois avec antenne "trace"	30
Figure2.14. Vue prototypage du montage de commande.....	32
Figure2.15. Vue schématique de l'abaisseur de tension 7812.....	32
Figure2.16.schéma d'un filtre RC	33
Figure2.17.Vue de signal PWM à la sortie du filtre RC	33
Figure 3.18. Volée d'Anser en formation en V.....	36
Figure 3.19. Schéma de principe du déplacement d'une particule	38
Figure 3.20. Organigramme de la méthode des essaims particulaires	40
Figure 3.21. Illustration de <i>collision</i> entre le <i>robot</i> et un obstacle.....	42
Figure 3.22. Division de l'obstacle statique en 4 parties, et calculer la nouvelle position du <i>robot</i>	43
Figure 3.23. Création de cercle limitant pour un obstacle dynamique.....	44
Figure 3.24. Création de la ligne droite entre les deux points, un point est dans la <i>BB</i> du <i>robot</i> dans la meilleure position globale, et le deuxième point est dans la <i>BB</i> du <i>robot</i> dans la position courante	45
Figure 3.25. Deux points d'intersection entre la droite et le cercle obtenus par la résolution de l'ensemble d'équation (9) et (10)	45
Figure 3.26. Trouver le point le plus proche à la meilleure position globale en utilisant la distance euclidienne.....	46

Figure 3.27. Création de plus court chemin virtuel, qui commence par le point de départ (carré vert) et se termine par le point d'arriver (carré rouge)	47
Figure 3.28. Capture d'écran de la simulation de notre approche dans un environnement avec des obstacles et en utilisant 20 particules	48
Figure 3.29. Le graph de fitness de la simulation de notre approche dans un environnement avec des obstacles et en utilisant 20 particules.....	48
Figure 3.30. Le chemin traversé par le robot dans la simulation de notre approche dans un environnement avec des obstacles et en utilisant 20 particules.....	49

INTRODUCTION GENERALE

Introduction générale

Depuis quelques années, un intérêt croissant est porté, au sein de la communauté robotique, au développement des systèmes intelligents autonomes dans le cadre de la robotique mobile. Un tel intérêt peut être perçu comme une conséquence logique à l'apparition des applications potentielles des machines intelligentes, i.e. industriels, services, manutention ou encore de l'aide à la mobilité des personnes âgées ou handicapées, etc.

Un nombre important de personnes handicapés ne peuvent pas utiliser un fauteuil électrique standard ou le conduisent difficilement. Pour permettre à ces personnes de se déplacer d'une manière autonome, des chercheurs se sont intéressés depuis plus de vingt ans à l'amélioration de l'interface homme-machine. Plus spécifiquement, soit en améliorant les actions de l'utilisateur sur le joystick classique du fauteuil, soit en implémentant de nouvelles interfaces de commande. Un autre axe de recherche concerne le développement de fauteuils roulants électriques (FRE) « intelligents ». L'intelligence d'un fauteuil roulant réside dans sa capacité à percevoir l'environnement grâce à différents types de capteurs. De nombreux prototypes de fauteuils intelligents ont été développés dans les laboratoires de recherche. La plupart des technologies et méthodes de la robotique mobile ont été implémentées et testées sans que malheureusement cela n'aboutisse à de réels progrès pour l'utilisateur : seules des fonctionnalités de bas niveau (détection d'obstacles, suivi de lignes) sont actuellement commercialisées et donc accessibles à tout le monde.

La problématique abordée dans ce travail consiste à proposer une fonctionnalité d'aide au pilotage des personnes malades ou âgées, et aux personnes atteint d'un handicap lourd. Ainsi, faire un pas vers le transfert technologique de fonctions de mobilité autonome plus complexe. Cette solution consiste à développer une commande pour la conduite des fauteuils électriques d'une façon autonome en introduisant un système de navigation intelligent à l'aide d'un algorithme bio-inspiré (ou algorithme évolutionnaire).

Pour participer à une perspective de développement de la robotique, nous nous sommes intéressés à l'optimisation de la planification du chemin et de mouvement du fauteuil roulant électrique en utilisant une méthode bio-inspirée. Afin d'effectuer ce travail, nous avons étudié l'algorithme d'optimisation par essaim de particules, qui est inspiré par le comportement des animaux qui marchent en groupe pour trouver la nourriture, cet algorithme a été introduit la première fois en 1995.

Motivation

Naviguer dans les hôpitaux est une tâche fastidieuse et exigeante pour les patients et les gardiens. Avec un nombre croissant de patients âgées ou handicapées, il y aura une plus grande demande de soins et de ressources santé dans un avenir proche. Pour aider les patients plus efficacement tout en réduisant la main-d'œuvre, dans cette recherche, nous avons proposé

un fauteuil roulant électrique FRE « intelligents » commandé par une carte Arduino pour la navigation dans un hôpital d'une façon complètement autonome. La proposition est capable de déterminer le chemin optimal pour trouver la destination avec précision et peuvent contrôler avec succès le fauteuil en mouvement pendant la navigation. Ainsi, il peut supprimer le besoin d'apprendre les tenants et les aboutissants des hôpitaux et d'améliorer la qualité de vie pour ses utilisateurs.

CHAPITRE 1

Généralité sur la robotique mobile

1. Introduction

Les limites de l'être humain dans la réalisation des différentes missions à cause de leurs difficultés le poussent à découvrir et développer des techniques qui facilitent la vie quotidienne. A cet égard, les robots sont conçus dans ce but pour contourner les limites humaines.

Dans ce chapitre, on aborde une des catégories des robots, i.e. les robots mobiles, où nous présentons quelques définitions, un historique sur l'évolution de la robotique et les différents types de robots mobiles. En outre, nous décrivons quelques caractéristiques des robots mobiles, notamment les différentes techniques de navigation et les architectures de contrôle.

2. Définition d'un robot mobile

Un robot mobile est un système regroupant trois disciplines, i.e. la mécanique, l'électronique et l'informatique, où il agisse physiquement sur un environnement afin d'atteindre un objectif ou d'accomplir une mission ou qui lui a été assignée.

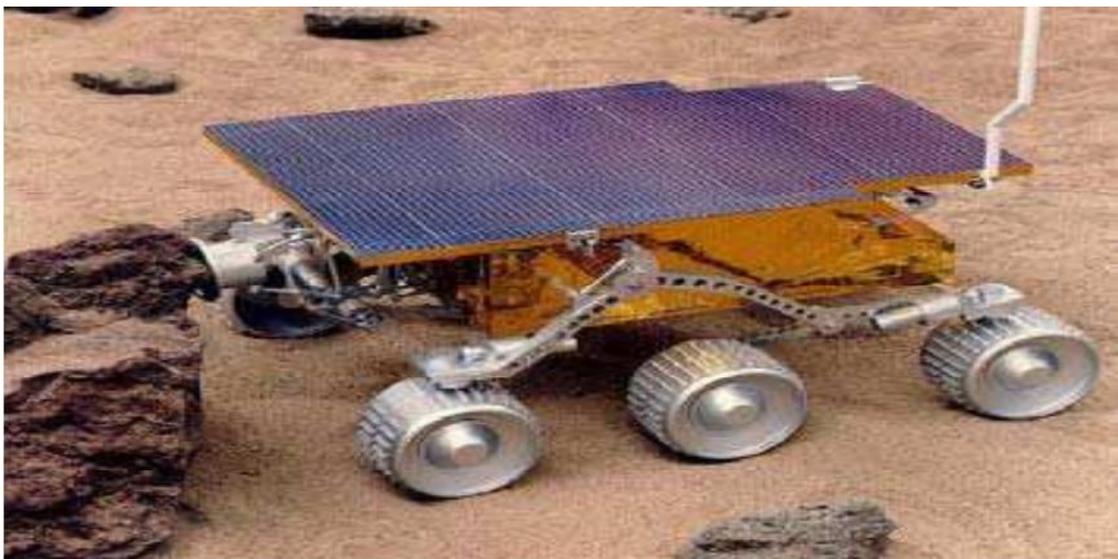


Figure 1.1. Le robot mobile Mars PathFinder développé par l'agence spatiale NASA.

Un robot mobile est une machine polyvalente capable de s'adapter à certains changements dans ses conditions de fonctionnement. Cette machine est dotée de fonctions de perception, de décision et d'action. Ainsi, le robot devrait être capable d'effectuer des tâches diverses de différentes manières [S1]. Plus spécifiquement, cette machine devrait être capable de s'adapter aux différentes situations rencontrées et d'accomplir correctement sa tâche.

3. Evolution de la robotique

Bien que le terme robot (*mot d'origine tchèque « robota »*) est apparu dans le 20^{ème} siècle dans la pièce de théâtre écrite par Karel Čapek, la notion de robot remonte à l'époque médiévale. Même si le terme n'existait pas pour décrire ce que nous appelons aujourd'hui, les gens de cette époque ont tout de même imaginé des mécanismes capables d'exécuter des tâches humaines [2].

Les automates, ce sont des machines aux formes humaines animées par un mécanisme intérieur, et ont été créés pour impressionner les paysans qui fréquentaient les églises et leur inculquer la notion d'« être » suprême.



Figure 1.2. L'automate Jacquemart.

A titre d'exemple, l'automate Jacquemart (Figure 1.2) crée l'illusion du mouvement automatique (non assisté). Plus précisément, les jacquemarts sont des personnages mécaniques capables de marquer le temps en frappant sur une cloche avec leur hache. Or, cette technologie étant à peu près inconnue au XIII^e siècle [1].

Au XVIII^e siècle, les automates miniatures sont devenus des jouets populaires chez les gens très riches. Ces figurines étaient à l'image d'humains ou de petits animaux et imitaient leurs mouvements [1].

Le robot musicien « Jaquet-Droz » a été fabriquée en 1890, où il peut tourner la tête dans les deux sens, et de jouer avec son instrument tout en battant la mesure avec le pied.

Une multitude de récits littéraires témoignent la création d'une forme de vie artificielle. En 1818, Mary Shelly a écrit Frankenstein, un récit qui relate la fabrication d'une créature d'apparence humaine. Le robot imaginé par l'auteur ressemblait à un homme, mais

fonctionnait comme une machine et était composé d'éléments aux formes humaines maintenus ensemble à l'aide de boulons et d'écrous. Il est remarqué qu'il y a même des pinces pour retenir ensemble les parties de sa tête, et selon Shelly, un robot devait être plus costaud que la moyenne d'entre nous et être doté d'une force surhumaine [1].



Figure 1.3. Le robot musicien Jaquet-Droz.



Figure 1.4. Le premier robot humanoïde de Léonard De Vinci.

En 1921, Karel Čapek, un dramaturge tchèque, a imaginé un personnage créé artificiellement et doté d'intelligence, auquel il a donné le nom de « robot ». Ce mot, qui signifie « travail forcé » en tchèque, est graduellement entré tel quel dans la langue française. Même s'il a été créé cent ans après la publication du roman Frankenstein de Shelly, le robot de Čapek a lui aussi des formes humaines [1].

Même si le concept de robot trouve son origine dans une époque très lointaine, il a fallu attendre les années 1940 et l'arrivée des ordinateurs pour que les robots des temps modernes fassent leur apparition. En 1948 Grey Walter invente le premier robot mobile autonome, une tortue se dirigeant vers les sources de lumière qu'elle perçoit. Cependant, ce robot n'est pas programmable.

La mise en place de robots n'a été possible que dans les années 50 avec la création des transistors et circuits intégrés. En 1956, John Mc. Carthy définissait le terme intelligence

artificielle, et le premier laboratoire de recherche en Intelligence Artificielle au M.I.T a été lancé entre 1960-1964. Ensuite, en 1967 Shakey a été le premier robot mobile créé qui utilisait une énergie électrique. Une grande caméra a été placée sur la tête et peut tourner et s'incliner, ainsi son corps est constitué d'un immense ordinateur reposant sur une plate-forme à trois roulettes [S2].

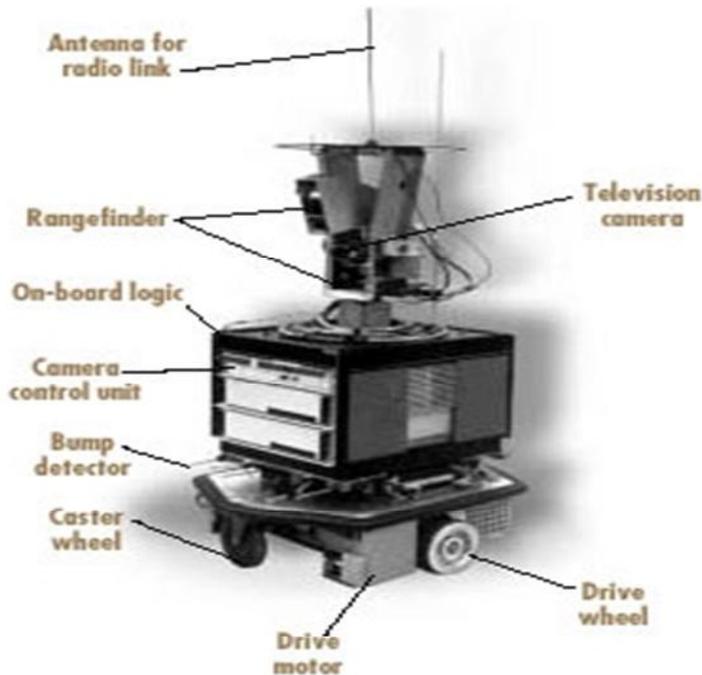


Photo courtesy of SRI International.

Figure 1.5. Shakey premier robot mobile contrôlé par ordinateur.

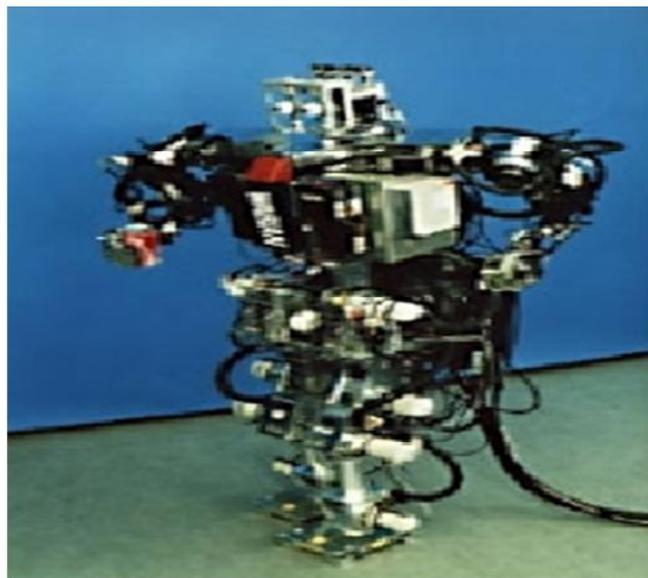


Figure 1.6. Wabot-1 le premier robot humanoïde à marche bipède.

En 1961, le premier robot industriel a été créé et intégré aux lignes d'assemblage de General Motors. Par la suite, en 1972, Nissan ouvre la première chaîne de production complètement robotisée.

La maîtrise de la bipédie est certainement l'accomplissement le plus technique de l'histoire des robots humanoïdes. En 1973 le robot WABOT-1 de l'université de Waseda était l'un des premiers robots humains capables de se déplacer de façon rudimentaire [1].



Figure 1.7. Le robot humanoïde de Honda ASIMO.

Le développement continu de l'intelligence artificielle et de la robotique entre les années 1980 à 1997 fait apparaître des nouveaux robots :

- En 1985, le robot WHL-11 d'Hitachi.
- Entre 1986-1993 les premiers robots de Honda E0 à E6.
- Entre 1992-1997 P1 à P3 capable de marcher comme un humain et de monter des escaliers.
- En 1997 le premier robot mobile extra planétaire envoyé sur Mars.

En 2000, Honda a développé le 11^{ème} robot, Asimo (Figure 1.7) qui est capable de courir. La dernière version de 2011 peut sauter avec un pied, courir à 9km/h, manipuler des objets délicatement (remplir une tasse), communiquer en langage de signes, etc.

A partir de 2005, la société française Aldebaran Robotics a développé le petit robot humanoïde, i.e. Nao. Ce dernier est autonome, programmable et forme une plateforme pédagogique idéale pour la recherche et l'enseignement. Par la suite, une version améliorée « Next Gen » a été lancée en fin 2011. En 2006, un petit robot humanoïde, i.e. iCub, a été développé conjointement par plusieurs universités européennes et mis en open-source. . En 2010, le robot Kobian de l'université Waseda au Japon est un robot capable de reproduire les expressions humaines.

Enfin, certains robots sont développés dans le maximum réalisme comme *Geminoid-F* du laboratoire ATR de Hiroshi Ishiguro (université d'Osaka). On trouve aussi le robot Sophia de 2017 qui a eu une nationalité pour la première fois dans l'histoire comme les êtres humains.



Figure 1.8. Le robot Sophia

Le terme « robotique » se rapporte à l'étude et à l'utilisation des robots et remonte à 1941, où il a été utilisé par le chercheur et écrivain Isaac Asimov dans un chapitre intitulé « Cycle fermé ». En grosso modo, il a proposé des lois de robotique comme suit [1]:

- Première loi: Un robot ne peut porter atteinte à un être humain ni restant passif, laisser cet être humain exposé au danger.
- Deuxième loi : Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres sont en contradiction avec la première loi.
- Troisième loi : Un robot doit protéger son existence dans la mesure où cette protection n'est pas en contradiction avec la première ou la deuxième loi.

4. Différents types de robots mobiles

4.1. Robots à roues

Compte tenu de la simplicité du mécanisme de locomotion, ce type de robot est le plus répandu actuellement. La plupart des robots mobiles à roues opèrent dans des sites aménagés, des sites industriels ou des environnements intérieurs. Cependant, il existe également des applications en environnement extérieur, comme l'exploration spatiale. La grande majorité des robots de ce type présentent des contraintes de non holonomie qui limitent le mouvement instantané que le robot peut réaliser. Ces contraintes ont pour conséquence d'augmenter la complexité du problème de planification de trajectoires et de son contrôle [3].

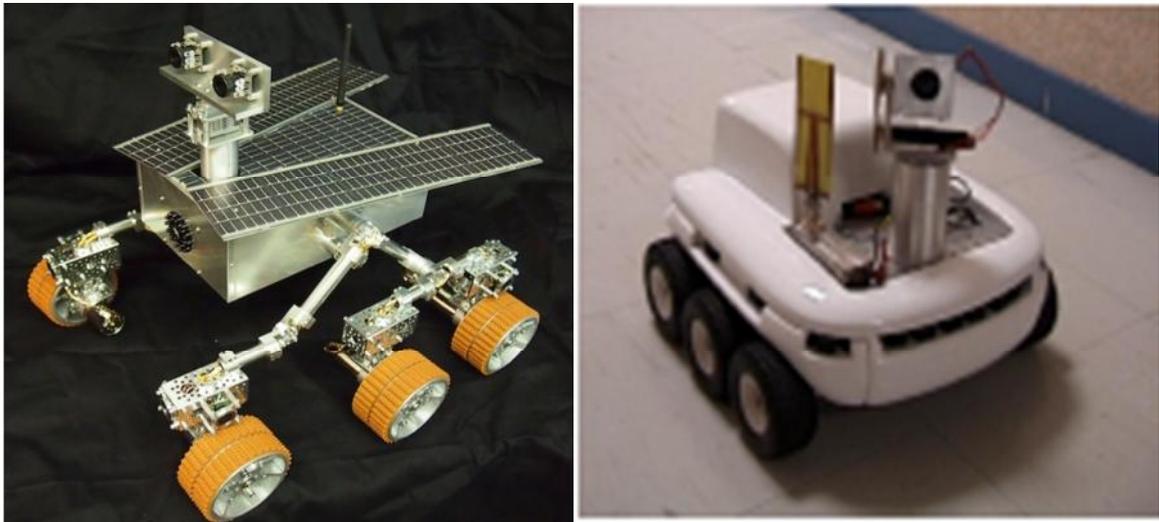


Figure 1.9. Exemples de robots mobiles à roues.

4.2. Robots à chenilles

Lorsque le terrain est accidenté (en relief), les roues perdent leur efficacité de locomotion. Ceci limite la capacité d'évolution du robot mobile équipé de ce type de système de locomotion. Dans ces conditions, les chenilles sont plus intéressantes, car elles permettent d'augmenter l'adhérence au sol et de franchir des obstacles plus importants. Ce type de robots présente également des contraintes de non holonomie [3].



Figure 1.10. Exemples de robots mobiles à chenilles.

4.3. Robots marcheurs

Dans la situation où le terrain est encore plus incertain, avec des grandes différences de hauteur comme par exemple un escalier ou un terrain très accidenté, les robots à roues ou à chenilles ne sont plus efficaces, et nous avons recours aux robots mobiles à pattes. Les robots mobiles à pattes ont des points d'appui discrets sur le terrain et sont donc la solution à ce problème de mouvement. Par contre, la conception et le contrôle d'un engin à pattes est très complexe, ainsi sa vitesse d'évolution est généralement très réduite [3].

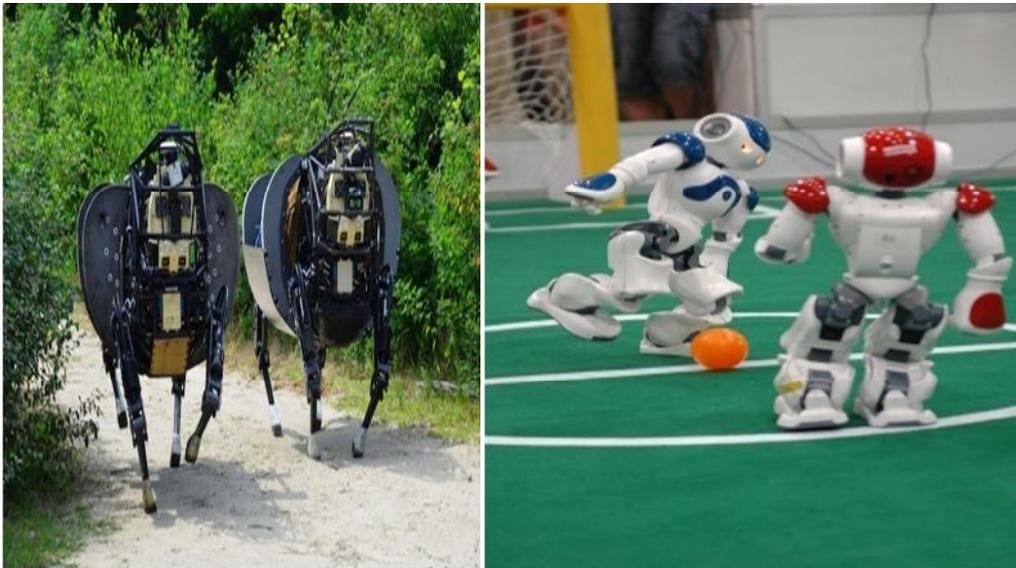


Figure 1.11. Exemples de robots marcheurs.

4.4. Robots rampants

La reptation est une solution de locomotion pour un environnement de type tunnel qui conduit à réaliser des structures filiformes. Le système est composé d'un ensemble de modules dont chacun plusieurs mobilités. Les techniques utilisées découlent des méthodes de locomotion des animaux [3].

- Le type scolopendre constitue une structure inextensible articulée selon deux axes orthogonaux.
- Le type lombric comprend trois articulations, deux rotations orthogonales et une translation dans le sens du mouvement principal.
- Le type péristaltique consiste à réaliser un déplacement relatif à un module par rapport aux voisins.

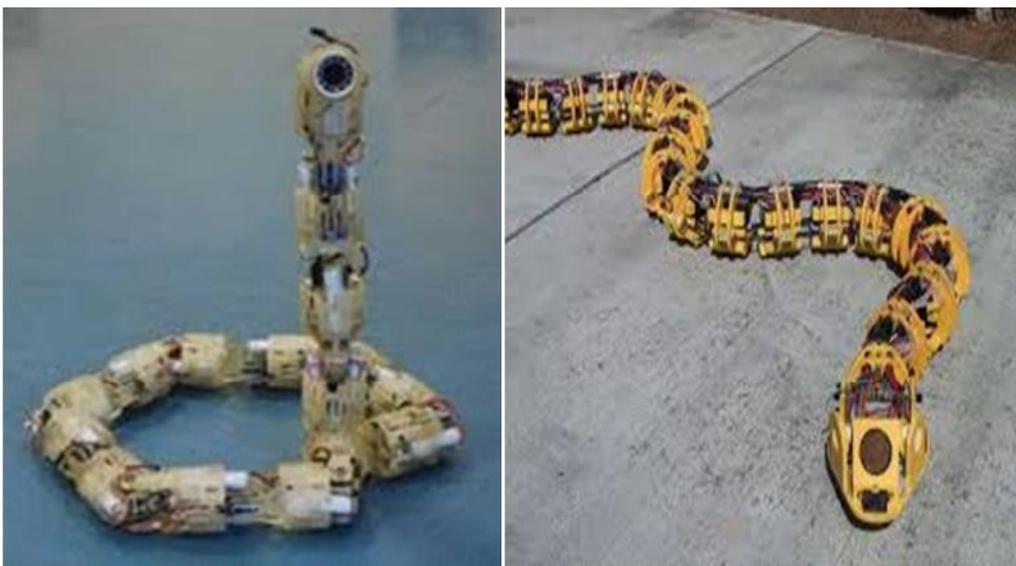


Figure 1.12. Exemples de robots rampants.

5. Notions de navigation

5.1. Définition

La navigation est une étape très importante en robotique mobile, d'où elle permet une large autonomie. Le système de navigation comporte plusieurs modules qui peuvent être traités différemment et parmi lesquels on distingue celui de la localisation et celui de l'évitement d'obstacles [12]. La détection et l'évitement d'obstacles est l'étape fondamentale de l'évolution d'un robot sur un territoire inconnu. On dispose à cet effet de plusieurs types de capteurs : caméras, capteurs laser, capteurs infrarouge et capteurs à ultrasons [13][14][15].

D'après Levitt et Lawton (1990), la navigation est définie comme le procédé permettant de répondre aux trois questions suivantes:

- Où suis-je ?
- Où sont les autres lieux par rapport à moi ?
- Comment puis-je atteindre ces autres lieux depuis l'endroit où je me trouve ?

Donc, la navigation est la science et l'ensemble des techniques qui permettent de connaître la position (coordonnées) du robot par rapport à un système de référence, ou par rapport à un point fixe déterminé. Ainsi, les techniques qui permettent de calculer ou mesurer le chemin à suivre pour rejoindre un autre point de coordonnées connues en respectant un certain nombre de contraintes [S5].

5.2. Méthodes de navigation

5.2.1. Navigation réactive

En 1986, Brooks a proposé une approche réactive qui se distingue par l'abandon des phases de modélisation et de planification (Figure 1.13). Autrement, le contrôle du robot se fait alors sans utiliser le modèle de l'environnement [11]. Les stratégies de navigation réactives n'utilisent que les valeurs courantes des capteurs et non des données provenant d'un modèle interne, pour décider de l'action à effectuer [10].

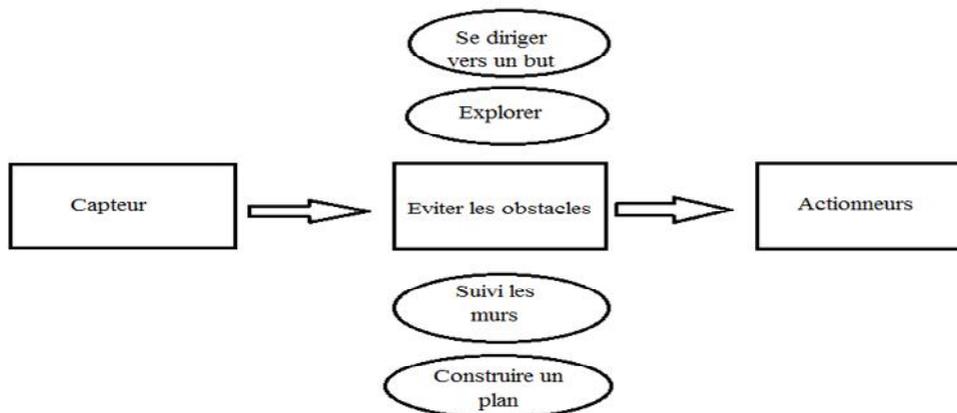


Figure 1.13. Architecture réactive

Le principe de l'approche réactive est basé sur la décomposition de la tâche de navigation en un ensemble de comportements actifs de base (explorer, aller au but, éviter les obstacles, suivi des murs, etc.). Pour guider le robot, il faut donc choisir à chaque instant lequel de ces comportements doit être activé ? Ce problème est connu dans la littérature scientifique sous le nom de sélection de l'action [10]. La solution proposée par Brooks consiste à diviser la tâche globale en un ensemble des sous tâches secondaires. Cette architecture est appelée "subsumption", utilise une hiérarchie de comportements qui se déclenchent donc selon un ordre de priorité en fonction des données de perception (fusion des comportements).

En effet, sans représentation interne de l'état de l'environnement, il est très difficile de planifier une suite d'actions en fonction d'un but à atteindre. Les robots utilisant cette architecture sont donc en général efficaces pour la tâche précise pour laquelle ils ont été conçus et dans l'environnement pour lequel ils ont été prévus, mais sont souvent difficiles à s'adapter à une tâche différente [16]. Puisque, le robot a une vue très réduite de son environnement, architectures réactives ne peuvent garantir le succès de la mission en toute circonstance. Les réussites de ces architectures sont liées au couplage direct entre la perception et l'action qui permet une prise en compte très rapide des phénomènes dynamiques de l'environnement, et donc une robustesse et fiabilité dans les situations complexes [16].

5.2.2. Navigation par asservissement visuel

L'asservissement visuel est une stratégie de navigation qui consiste à utiliser les informations fournies par une ou plusieurs caméras dans la boucle de commande d'un robot afin d'en contrôler le mouvement et de réaliser des tâches de navigation complexes dans des environnements encombrés d'obstacles. La complexité des problèmes à résoudre dépend principalement de la compréhension des informations issues des capteurs visuels ainsi que la transformation de ces informations en données parfaitement utilisables par les composants décisionnels du robot. Ces informations sont des segments de droites, courbes ou régions, selon les besoins des applications envisagées. Donc l'asservissement visuel est basé sur le traitement d'image. Les techniques d'asservissement visuel utilisent la perception afin d'améliorer l'action et augmentent la flexibilité et la précision des systèmes robotiques. La boucle perception-vision peut être complétée par la vision active où le mouvement du robot est contrôlé afin d'améliorer la perception [17].

5.2.3. Navigation par carte

Elle s'appuie sur un modèle interne du monde, i.e. une carte, qui supporte une planification. Ce modèle interne mémorise la structure spatiale de l'environnement, indépendamment d'un but précis. Chacune des positions mémorisées dans ce modèle interne peut être utilisée comme un but par le processus de planification, dont le rôle est de calculer un chemin vers ce but. Le principe de la navigation par carte est basé sur la cartographie, la localisation et la planification. On peut citer deux sous classe :

Navigation par carte topologique

Une carte topologique est un graphe, dans lequel chaque nœud correspond à un endroit caractéristique (carrefour entre couloirs, entrées dans les espaces ouverts, etc.), appelé *meeting point* dans plusieurs travaux, ou encore lieu [18]. Chaque lieu devra être décrit par un ensemble de caractéristiques propres, qui permettront au robot de le reconnaître. Une arête liant deux nœuds signifie qu'il existe une commande référencée capteur que le robot peut exécuter afin de se déplacer entre les deux lieux correspondants. Cette représentation est particulièrement adaptée pour décrire des réseaux de couloirs : les lieux correspondent aux carrefours, les arêtes aux couloirs qui les lient. Pour la génération d'une trajectoire entre deux lieux définis dans une carte topologique, il suffit de rechercher un chemin dans le graphe. Dans cette stratégie, le robot n'a pas besoin de connaître sa position précise par rapport à un repère global. Il doit seulement déterminer une localisation qualitative « *Je suis dans le lieu A* » et, au mieux, une estimation de sa position relative par rapport à un repère lié au lieu dans lequel il se trouve [19].

Navigation par carte métrique

La navigation métrique exploite une représentation géométrique du monde, qui peut être donné par un utilisateur, ou construite par le robot lui-même. Elle représente une extension de la précédente car elle permet au robot de planifier des chemins au sein de zones inexplorées de son environnement. Elle mémorise pour cela les positions métriques relatives aux différents lieux, en plus de la possibilité de passer de l'un à l'autre. Ces positions relatives permettent, par simple composition de vecteurs, de calculer une trajectoire allant d'un lieu à un autre, même si la possibilité de ce déplacement n'a pas été mémorisée sous forme d'un lien [20].

L'équivalent mathématique de la carte métrique, est un espace vectoriel muni d'une distance, où chaque vecteur AB exprime le chemin entre un lieu A et un lieu B en tenant compte de la distance et de l'orientation. Les détours et raccourcis métriques peuvent alors être manipulés via les opérations vectorielles telles que la norme (estimation des distances), l'addition ou la soustraction vectorielle. Il n'est plus requis d'avoir emprunté un chemin pour pouvoir l'utiliser, et c'est bien là la différence fondamentale avec la carte topologique [20].

6. Architectures de contrôle

Les architectures de contrôles peuvent être classées en trois grandes catégories, i.e. les contrôleurs hiérarchiques, les contrôleurs réactifs et les contrôleurs hybrides. Toutes ces architectures ne diffèrent pas forcément par les méthodes élémentaires employées mais plutôt par leur agencement et leurs relations.

6.1. Contrôleur hiérarchique

Ces approches s'appuient sur les techniques de l'intelligence artificielle classique. Elles visent à reproduire le mode de raisonnement humain ou tout au moins une certaine vision du

mode de raisonnement humain. Pour cela, ils sont appuyés sur un schéma d'intelligence artificielle conçu dans les années cinquante par Newell et Simon [21]. Le traitement est décomposé en une série d'opérations successives (Figure 1.14).

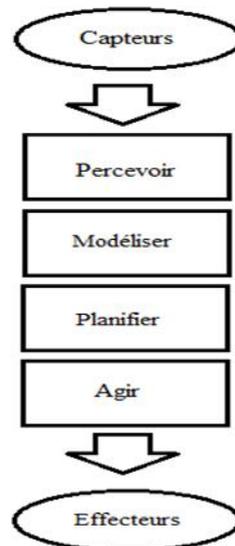


Figure 1.14. Schéma du contrôleur hiérarchique

La première opération consiste à traiter les données sensorielles qui fournissent au robot des informations sur son environnement. Ensuite, à partir des données des capteurs un modèle du monde dans lequel évolue le robot est construit. Ce dernier peut être par exemple une identification du type et de la position des obstacles que le robot doit éviter. Ce modèle est utilisé pour planifier une suite d'états conduisant le robot à effectuer la tâche recherchée. Enfin, la dernière opération consiste à calculer et d'exécuter les actions permettant au robot de suivre le plan généré par l'étape précédente.

La partie la plus importante et la plus coûteuse en termes de calculs, est celle qui concerne la modélisation et la planification, notamment dans un environnement peu structuré. Le temps disponible pour l'action est par conséquent beaucoup plus court, mais par contre, le temps entre les phases de perception et d'action peut être relativement long.

Dans une telle architecture, on essaie de construire un modèle de l'environnement le plus complet possible et ensuite de raisonner sur la représentation de celui-ci. On privilégie l'aspect cognitif dans le but de reproduire au mieux l'intelligence humaine. Ces architectures ont rapidement montré leurs limites et leur incapacité de fonctionner dans un environnement qui ne soit pas statique et simplifié à l'extrême.

L'avantage essentiel de ces architectures réside dans la possibilité d'intégrer des raisonnements de haut niveau (niveau mission, planification) qui s'appuient sur des modèles assez complets de l'environnement dans lequel évolue le robot. En revanche, leur inconvénient principal provient de leur faible vitesse de réaction. Elles sont en effet

totallement incapables de prendre en compte des obstacles dynamiques ou non détectés lors de la modélisation. Ceci est dû à la lenteur des phases de modélisation et de planification qui ne peuvent généralement pas être exécutées en temps réel, même lorsqu'elles sont implantées sur de gros calculateurs extérieurs au robot.

L'utilisation de ces approches semble donc limitée aux robots évoluant dans des environnements statiques et dont la structure est fortement contrainte et connue a priori. Les limitations de cette approche ont conduit certains chercheurs à développer des systèmes réactifs..

6.2. Contrôleur réactif

Cette approche, radicalement différente de la précédente et se distingue par l'abandon des phases de modélisation et de planification. Elle est née à partir de considérations issues de l'éthologie et plus particulièrement de l'entomologie. Nous savons aujourd'hui que de nombreux êtres vivants, dont en particulier les insectes, ne possèdent, en raison de leur taille que très peu de capacités de modélisation et de raisonnement. Ils n'en sont pas moins capables de mener à bien des tâches complexes. Leur comportement semble même du point de vue d'un observateur extérieur résulter d'une intelligence avancée, similaire par certains cotés à la notre.

Cette constatation induit l'idée suivante : pourquoi ne pas abandonner le concept centré sur la modélisation de l'environnement et le raisonnement, au profit d'une architecture totalement distribuée privilégiant un lien étroit entre perception et action. Arkin parle de modèle "action-oriented perception" précisant que les besoins perceptifs d'un agent doivent être basés sur ses nécessités en action. Cela s'oppose à une approche hiérarchique collectant des informations, puis travaillant à différents niveaux d'abstraction, sans vraiment prendre en considération les utilisations ultérieures qui en seront faites.

Brooks définit 8 niveaux de compétences dans le cas d'un robot d'exploration, et qui sont dans l'ordre [22] :

- Éviter les obstacles.
- Errer sans but (au hasard) sans rien toucher.
- Explorer le monde en observant à distance les emplacements qui semblent atteignables et s'y rendre.
- Construire une carte de l'environnement et des routes d'un lieu à l'autre.
- Remarquer les changements dans l'environnement statique.
- Reasonner sur le monde en termes d'objets identifiables et de réalisation de tâches relativement à certains objets.
- Formuler et exécuter un plan qui implique un changement voulu de l'état du monde.
- Reasonner sur les comportements des objets et modifier les plans en conséquence.

L'avantage de ce contrôleur est la rapidité de réponse qui lui donne la capacité d'évoluer dans des environnements fortement dynamiques. La simplicité des comportements permet d'accroître encore cette vitesse par une réalisation sur circuit imprimé. En outre, la robustesse qui découle de la parallélisations et de l'indépendance relative des comportements qui permet au robot de continuer à fonctionner en cas de panne de l'un d'entre eux, ce qui est très important lorsque l'on a à faire à un robot autonome.

6.3. Contrôleur hybride

Les approches hiérarchiques et réactives sont diamétralement opposées. Cependant, chacune présente des caractéristiques intéressantes. Pour cela, des 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, en s'appuyant sur des représentations abstraites des connaissances, avec des comportements réactifs garantissant la robustesse et la flexibilité [23].

La plupart des contrôleurs actuellement utilisés choisissent une solution intermédiaire entre ces deux approches sous la forme d'une architecture hybride. Cette notion est à distinguer de celle de système hybride en intelligence artificielle (ou système hybride neuro-symbolique) qui désigne un système alliant des composantes utilisant des techniques issues de l'intelligence artificielle symbolique classique et des réseaux de neurones artificiels.

L'architecture hybride se compose de deux niveaux. Le premier est chargé des tâches de navigation de haut niveau, telles que la localisation, la cartographie et la planification. Pour cela, il s'appuie sur un second niveau réactif qui est chargé d'exécuter les commandes avec le plus de précision possible et de gérer les éléments non modélisés de l'environnement tels que les obstacles inconnus ou dynamiques [24]. L'action conjointe de ces deux niveaux permet de réagir rapidement face aux variations imprévues de l'environnement, tout en permettant la réalisation d'actions planifiées au plus long terme.

Le bas niveau de ces architectures peut être réalisé sous forme de comportements, tels que ceux utilisés dans les architectures réactives. Ces comportements sont des boucles sensorimotrices qui relient les actions aux perceptions avec une phase de décision très courte, qui assurent la réactivité. Dans le même temps, les informations sensorielles sont utilisées par le haut niveau dans une boucle sensorimotrice à une échelle de temps beaucoup plus longue. C'est la mise en parallèles de ces deux échelles de temps qui fait la force de ces architectures.

7. Collaboration en robotique

La collaboration en robotique est une étape très importante qui consiste à répartir les tâches et les décomposer pour faciliter la mission. Or, on distingue deux types de collaboration :

7.1. Collaboration homme-robot

Les robots deviennent de plus en plus présents dans notre environnement et sont désormais capable d'interagir avec les humains. Cette discipline s'appelle la *cobotique*, et est le domaine de la collaboration homme-robot (l'interaction) qui peut être directe ou télé opérée pour atteindre un objectif commun [S4]. Lors de la collaboration homme-robot, le robot assiste l'homme. Cela signifie que la machine ne remplace pas l'homme, mais **complète ses aptitudes** et lui facilite les tâches lourdes répétitives et dangereuses, que ce soit dans un contexte industriel ou quotidien [4].



Figure 1.15. Exemple de collaboration homme-robot.

Dans les milieux industriels, des robots viennent notamment assister les opérateurs des chaînes d'assemblage pour des tâches fatigantes et dangereuses. Robots et opérateurs sont alors amenés à partager le même espace physique et à effectuer des tâches en commun, comme les robots manipulateurs à multi-applications [5]. Alors que la sécurité des humains à proximité des robots doit être garantie à tout instant. En effet, pour développer des robots capables de collaborer et de vivre en contact avec les humains, des chercheurs et industriels développent des algorithmes d'apprentissage [6].

7.2. Collaboration entre les robots

Les robots sont dotés d'un système informatique leurs permettant de commander leurs sorties, et d'acquérir les informations provenant des entrées. L'évolution des capteurs a permis aux robots de mieux appréhender l'environnement où ils se situent [7]. Toutefois, l'intelligence artificielle leur permette d'interagir entre eux, ainsi avec le monde extérieur ce qui présente une opportunité pour étendre leurs capacités de perception du monde extérieur [7]. Plusieurs robots coopérants peuvent effectuer de nombreuses tâches plus rapidement et de manière plus fiable qu'un seul robot. Un système multi robot est un système constitué d'un ensemble de robots, homogènes ou hétérogènes, mobile ou fixe, ces robots sont capables de communiquer entre eux et coopérer pour améliorer l'efficacité d'exécution des tâches ou encore pour permettre d'exécuter des tâches impossibles à exécuter de façon individuelle [9].

Ils doivent coopérer entre eux, ils s'auto-organisent pour soulever, porter pousser et déposer des objets qu'un seul robot ne pourrait réaliser tout seul. La collaboration entre les robots est basée sur quatre critères : la communication, la computation, la configuration, et la coordination [8]. La communication définit comment les robots communique entre eux. La computation définit la mise en œuvre du comportement des agents individuels, et elle détermine ainsi ce qui est communiqué. La configuration définit la structure d'interaction et indique quels sont les robots existants dans le système et quels sont les robots qui peuvent communiquer entre eux. La coordination définit les modes d'interaction, i.e. déterminé à quel moment certaines communications auront lieu [9].



Figure 1.16. Exemple de collaboration entre plusieurs robots pour transporter un boîtier.

9. Conclusion

Dans ce chapitre, nous avons présenté quelques définitions sur les robots mobiles, ainsi un petit historique concernant l'évolution de ce domaine. Nous avons détaillé également quelques caractéristiques techniques de cette catégorie de robots. En particulier, nous avons vu les différentes méthodes de navigation, i.e. réactive, par asservissement, par carte et hybride. En outre, les trois architectures de contrôle ont été abordées et détaillées dans ce chapitre, et la collaboration en robotique également.

CHAPITRE 2

Fauteuils roulants électriques

1. Introduction

Dans ce chapitre nous présentons tout d'abord les caractéristiques de notre fauteuil roulant électrique. En particulier, nous décrivons le choix du fauteuil roulant électrique, la partie commande et la partie puissance. Le point suivant va s'intéresser au deuxième volet du choix de matériels tels que le microcontrôleur, les capteurs et l'installation de ces derniers. Nous aborderons ensuite les différents logiciels qu'on va utiliser. Enfin nous concluons ce chapitre par la description de la carte de commande du fauteuil roulant électrique qu'on va réaliser.

2. Généralité sur les fauteuils roulants électriques

2.1. Les origines du fauteuil roulant électrique

A partir des années 1950, les premiers fauteuils roulants électriques sont apparus à l'usage des personnes souffrant de déficiences motrices sévères [25]. Cependant, le fauteuil roulant manuel n'a pas été abandonné et son utilisation est restée très largement répandue comme en attestent les résultats des enquêtes « H.I.D. » réalisées en France, où la prévalence des utilisateurs de fauteuils roulants électriques et manuels dans la population générale était respectivement de 5,4 et de 59 personnes pour 10000 habitants [26].

2.2. Descriptif du fauteuil roulant électrique

Les fauteuils roulants à propulsion par moteur électrique sont indiqués pour les personnes ne pouvant se déplacer seules en fauteuil roulant manuel. Ils permettent de retrouver une indépendance notamment au niveau des sorties, à l'extérieur du domicile. La conduite du fauteuil s'effectue par commande électronique mais parfois l'utilisation d'une commande spéciale est indispensable. Il existe différents types de fauteuils qui sont présentés dans la (figure 2.1).



Figure 2.1. Les différents types de fauteuils (A: Fauteuil d'intérieur, B: Fauteuil d'extérieur, C: Fauteuil verticalisateur, D: Fauteuil à hauteur variable).

- *Fauteuil d'intérieur* : Pliant, il est moins volumineux qu'un fauteuil d'extérieur. Notons qu'il est cependant difficile et fatiguant pour la tierce personne de le plier. Son autonomie va de 10 à 16 km et sa vitesse est réglable de 0 à 6 km/h. On peut le mettre dans le coffre d'une voiture.
- *Fauteuil d'extérieur* : Non pliant, il est plus volumineux. Son autonomie va de 16 à 45 km et sa vitesse est réglable de 0 à 10 km/h. Ce fauteuil nécessite un moyen de

transport adapté car on ne peut pas le mettre dans un coffre de voiture. Le diamètre de ses roues permet de franchir des seuils ou des petits obstacles.

- *Fauteuil verticalisateur* : La verticalisation permet de quitter la position assise pour se placer en position verticale. Elle permet également de mieux appréhender l'environnement et peut être important pour l'intégration sociale. Le fauteuil doit être équipé de cales genoux et d'une sangle au niveau du tronc.
- *Fauteuil à hauteur variable* : Permet de lever l'utilisateur en position assise. Tout comme le verticalisateur il permet de mieux appréhender l'environnement et peut être important pour l'intégration sociale. Certains fauteuils électriques peuvent combiner verticalisation et hauteur variable en position assise.

3. Les Fauteuils roulants intelligents

3.1. Le projet de VAHM

Le projet VAHM (Véhicule autonome pour Handicapés Moteurs) [27] a pour but de proposer aux personnes ayant un handicap physique lourd de se déplacer facilement. L'interface est composée d'un bouton et d'un cadran indiquant successivement huit directions de déplacement, comme présente la (figure 2.2). Lorsque l'utilisateur veut suivre une direction, il doit attendre qu'elle s'affiche sur le cadran et presser le bouton à ce moment. Lorsqu'il relâche le bouton, le fauteuil s'arrête et il faut choisir une nouvelle direction [28].

Ce système présente l'avantage de ne nécessiter qu'un seul type d'action de la part de l'utilisateur, mais a l'inconvénient d'être lent et fastidieux. C'est pourquoi le fauteuil doit se déplacer de la manière la plus autonome possible. Pour cela, il est équipé de capteurs qui collectent des données sur son environnement et d'agents réactifs qui lui permettent des comportements simples (longer les murs, éviter les obstacles, etc.).

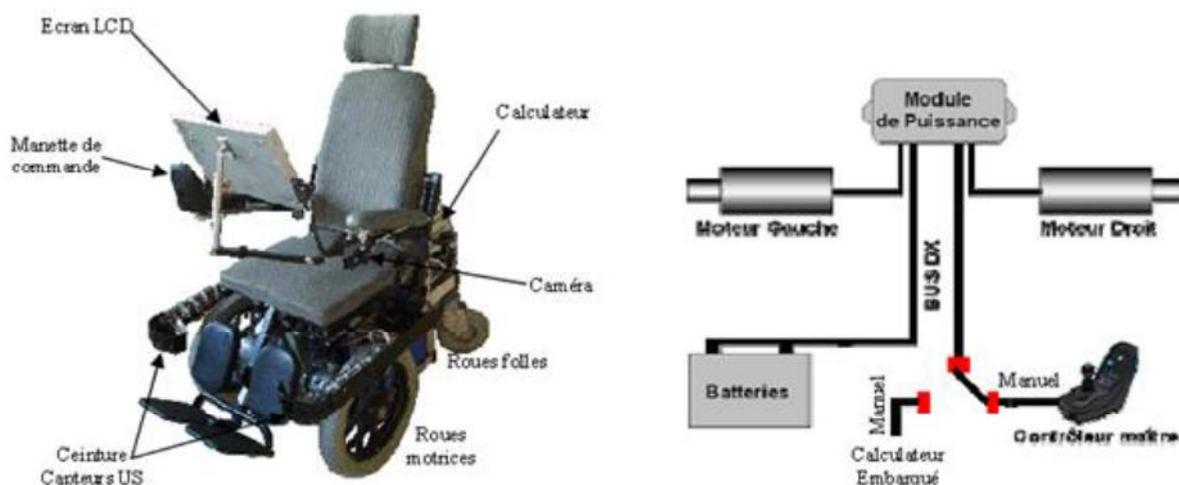


Figure 2.2. Le projet de VAHM

Lors de ce projet, un simulateur a été développé. Celui-ci permet de remplacer les expérimentations avec un véritable fauteuil roulant par des expérimentations informatiques, plus rapides, donnant des résultats sensiblement proches.

3.2. Projet WAD « WHEELCHAIR ATTRACTOR DYNAMICS »

L'architecture de commande du Projet WAD (Wheel chair Attractor Dynamics) [29] est installée actuellement sur un fauteuil roulant électrique modèle CRUISER de chez INVACARE qui est montré dans la (figure 2.3). Ce fauteuil est équipé d'un système DX de base [30] (joystick et module de puissance), auquel nous avons rajouté le module DX Key, qui permet un interfaçage aisé via le port parallèle d'un ordinateur portable.

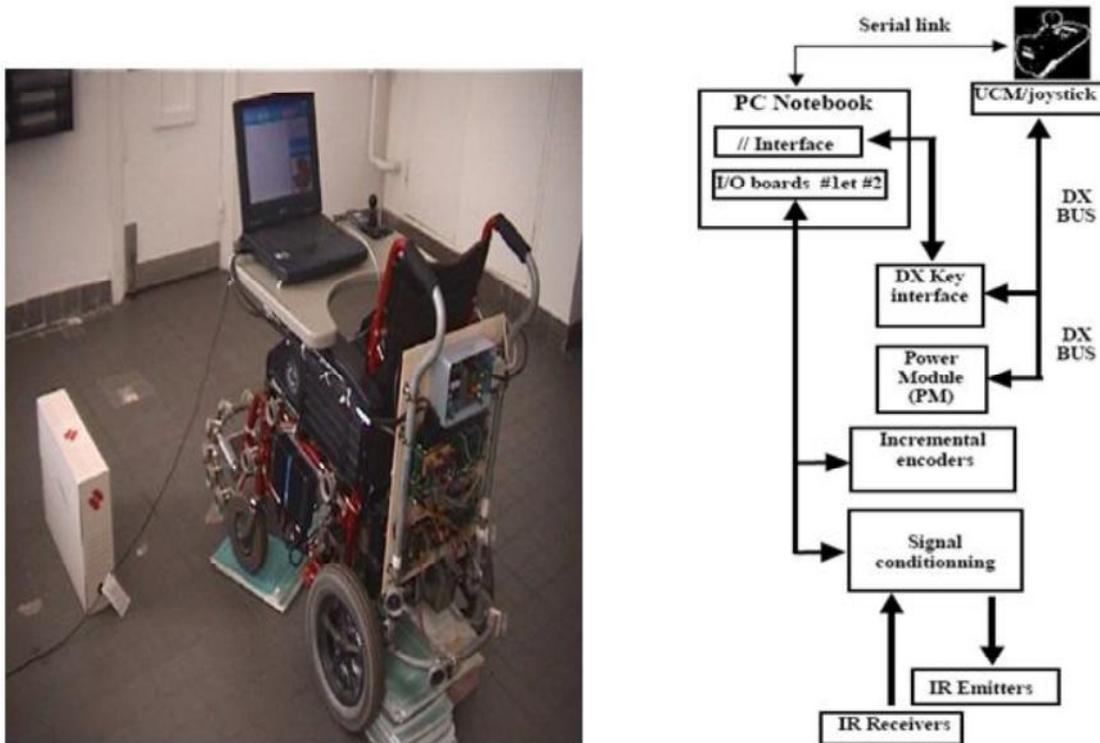


Figure 2.3. *Projet WAD « WHEELCHAIR ATTRACTOR DYNAMICS ».*

En mode manuel, le pilote impose au moyen du joystick des commandes de vitesse et de direction qui sont transmises au module de puissance et à la motorisation via le bus DX. En mode automatique, les capteurs de proximité infrarouge et les codeurs incrémentaux renseignent sur les contraintes de l'environnement [31], ainsi que sur l'orientation et la distance de la cible par rapport à la position courante du véhicule.

Ces informations proprioceptives (codeurs incrémentaux) et extéroceptives (capteurs de proximité) permettent de calculer des commandes de vitesse et de direction qui sont transmises au module de puissance via l'interface DX Key sans avoir à manipuler le joystick [32]. Le fauteuil atteint seul sa destination. A tout moment le pilote peut passer du mode automatique au mode manuel et inversement.

4. Les capteurs

Nous présentons dans cette section les capteurs les plus couramment utilisés en robotique mobile pour les besoins de la navigation.

4.1. Les capteurs proprioceptifs

Les capteurs proprioceptifs permettent une mesure du déplacement du robot [S6]. Ce sont les capteurs que l'on peut utiliser le plus directement pour la localisation, mais ils souffrent d'une dérive au cours du temps qui ne permet pas en général de les utiliser seuls. Dans cette classe des capteurs on trouve l'odométrie.

L'odométrie permet d'estimer le déplacement de la plateforme à partir de la mesure de rotation des roues (ou du déplacement des pattes). La mesure de rotation est en général effectuée par un codeur optique disposé sur l'axe de la roue, ou sur le système de transmission [33].

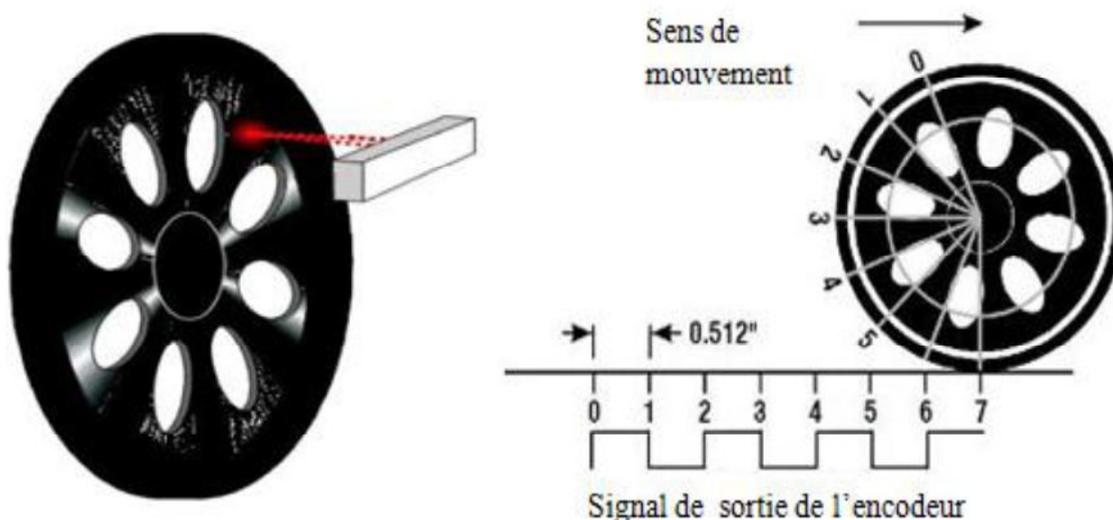


Figure 2.4. Principe général de l'odométrie.

Le problème majeur de cette mesure est que l'estimation du déplacement fournie dépend très fortement de la qualité du contact entre la roue (ou la patte) et le sol. Elle peut être relativement correcte pour une plate-forme à deux roues motrices sur un sol plan de qualité uniforme, mais est en général quasiment inutilisable seule pour un robot à chenille par exemple. Pour limiter ce problème, il peut être intéressant de positionner le codeur optique sur une roue non motrice qui glisse à moins. Notons cependant que l'erreur de ces méthodes se retrouve en général principalement sur l'estimation de la direction du robot, tandis que la mesure de la distance parcourue est souvent de meilleure qualité.

4.2. Les télémètres

Il existe différents types de télémètres, qui permettent de mesurer la distance aux éléments de l'environnement, utilisant divers principes physiques.

a. Télémètres à ultrasons

Les télémètres à ultrason sont historiquement les premiers à avoir été utilisés. Ils utilisent la mesure du temps de vol d'une onde sonore réfléchiée par les obstacles pour estimer la distance comme illustre-la (figure 2.5).

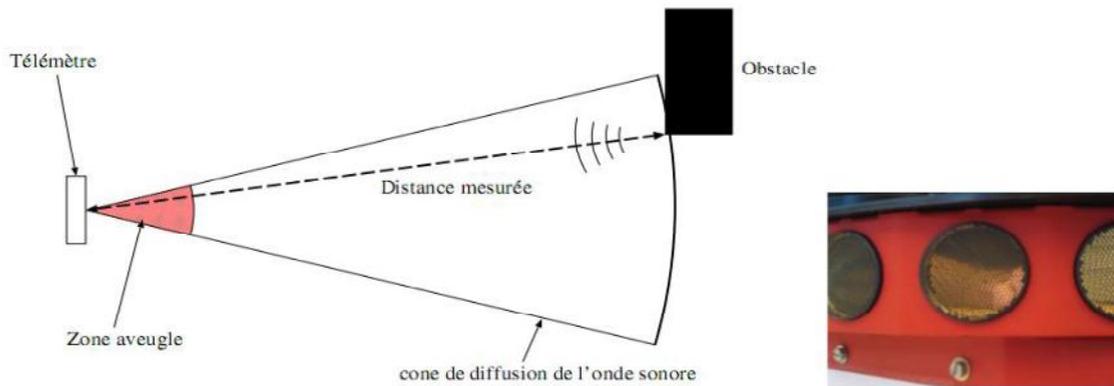


Figure 2.5. Principe du télémètre à ultrasons et exemple de télémètre réel.

Ces télémètres sont très simple et peu cher, et sont donc très répandus. Ils possèdent une “zone aveugle”, de quelques centimètres, en dessous de laquelle ils ne peuvent détecter les obstacles. Cette zone est due à une temporisation entre l’émission de l’onde sonore et le début de la détection de l’onde réfléchie qui est nécessaire pour ne pas perturber cette mesure.

$$\text{La distance } D \text{ entre le capteur et l'obstacle est : } D = T \cdot V / 2 \quad (1)$$

Avec (V) la vitesse et (T) le temps de propagation de l’onde.

Les télémètres ultrason détectent les obstacles se situant dans un cône relativement large (Angle au sommet d’environ 30 degrés). Cette caractéristique présente un avantage, car des éléments relativement fins (les pieds de table ou de chaise par exemple) sont détectés dans ce cône, alors qu’ils pourraient ne pas être détectés par des télémètres ayant un angle d’ouverture très fin.

b. Télémètres à infrarouges

Les télémètres infrarouges possèdent l’avantage d’avoir un cône de détection beaucoup plus restreint. Ils utilisent une lumière infrarouge au lieu d’une onde sonore pour la détection et peuvent être basés sur différentes techniques qui permettent de recueillir plus ou moins d’information.

Il est possible de mesurer simplement le retour ou le non-retour d’une impulsion codée, ce qui permet de détecter la présence ou l’absence d’un obstacle dans une certaine portion de l’espace. Il est également possible de réaliser une triangulation sur le faisceau de retour de l’onde lumineuse, ce qui permet d’avoir une mesure de la distance de l’obstacle, comme illustre la (figure 2.6).

Les inconvénients de ces télémètres sont liés à leur portée, en général relativement restreinte, et à leur sensibilité aux sources de lumières qui contiennent un fort rayonnement infrarouge. Un projecteur du type de ceux utilisés pour la télévision pointé sur le robot, par exemple, sature en général complètement le récepteur et empêche toute détection d’obstacle. Ils sont également très sensibles à la couleur et à la nature de la surface de l’obstacle (par exemple, ils détectent difficilement les vitres et les obstacles noir mats).



Figure 2.6. Principe du télémètre infrarouge à triangulation et exemple de télémètre réel.

c. Télémètres laser

Les télémètres les plus utilisés à l'heure actuelle pour des applications de cartographie et de localisation sont les télémètres laser à balayage. Ils utilisent un faisceau laser mis en rotation afin de balayer un plan, en général horizontal, et qui permet de mesurer la distance des objets qui coupent ce plan comme montre la (figure 2.7).

Cette mesure peut être réalisée selon différentes techniques soit en mesurant le temps de vol d'une impulsion laser, soit par triangulation. Les télémètres courants ont une bonne résolution angulaire car ils permettent d'obtenir une mesure de distance tout les demi degrés, sur une zone de 180 ou 360 degrés selon les modèles. La mesure est de plus relativement précise (avec un bruit de l'ordre de quelques centimètres) à une distance relativement grande (plusieurs dizaines de mètres). La fréquence d'acquisition est en général de l'ordre de la dizaine de Hertz, voire proche de la centaine pour certains modèles.

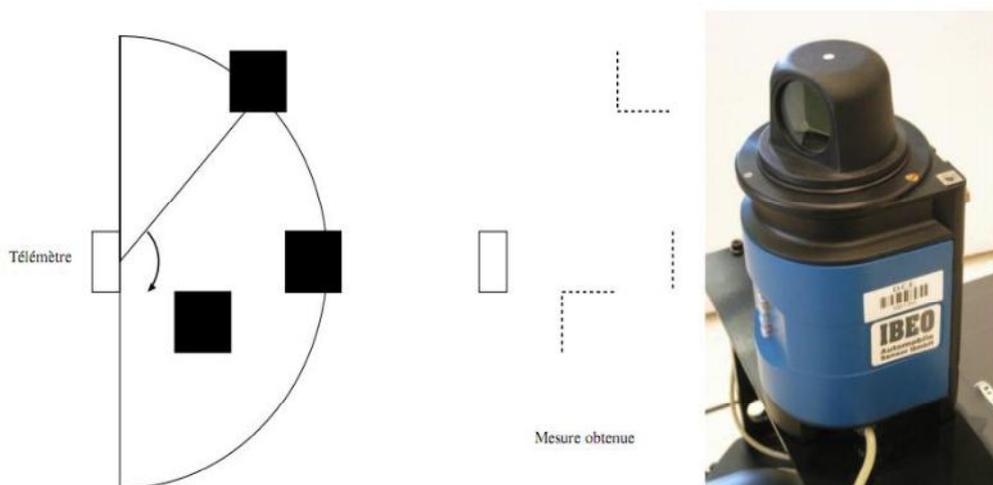


Figure 2.7. Illustration du principe de fonctionnement d'un télémètre Laser et un exemple de Télémètre Laser à balayage, fournissant 720 mesure réparties sur 360 degrés, à 5Hz (marque LBEO).

4.3. Les capteurs photographiques

Un capteur photographique [S7] est un composant électronique photosensible servant à convertir un rayonnement électromagnétique (UV, visible ou IR) en un signal électrique analogique. Ce signal est ensuite amplifié, puis numérisé par un convertisseur analogique-numérique et enfin traité pour obtenir une image numérique. Le capteur est donc le composant de base des appareils photo numériques, l'équivalent du film en photographie argentique.

Deux grandes familles de capteurs [S8] sont disponibles : les CCD et les CMOS (figure 2.8).

- Le capteur le plus répandu est le capteur CCD « Charge Couple Device ». A l'origine, il a été développé pour des applications d'imagerie. Il délivre des courants assez importants et requiert donc peu d'amplification. Malheureusement, il est très 'gourmand' en énergie et il est assez coûteux à fabriquer.
- La technologie CMOS est présente dans beaucoup de composants électroniques informatiques (mémoires d'ordinateur par exemple), les capacités de fabrication sont donc plus importantes et le prix moins cher. Ce type de capteur est également moins 'gourmand' en énergie mais demande plus d'amplification, il a tendance à générer plus de bruit.

Il faut noter également que le capteur CCD est plus rapide que le capteur CMOS.

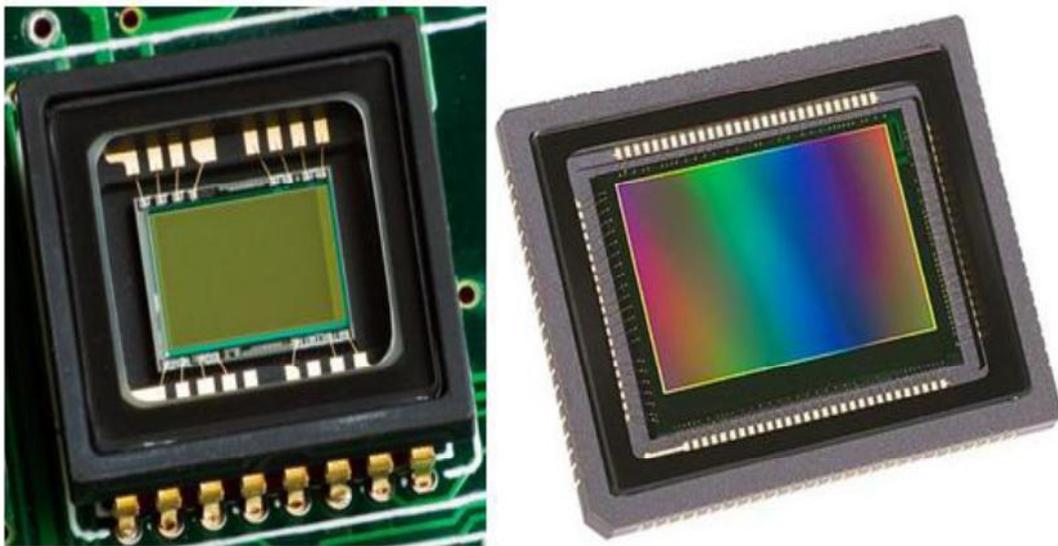


Figure 2.8. Les capteurs photographiques (A: capteur CCD, B: capteur CMOS).

4.4. Les caméras

L'utilisation d'une caméra pour percevoir l'environnement est une méthode attractive car elle semble proche des méthodes utilisées par les humains et fournit une grande quantité d'information sur l'environnement. Le traitement des données volumineuses et complexes fournies par ces capteurs est cependant souvent difficile, mais c'est une voie de recherche très explorée et prometteuse pour la robotique.

a. Les caméras stéréoscopiques



Figure 2.9. Exemple de données fournies par des caméras stéréoscopiques.

Lorsque l'on dispose de deux caméras observant la même partie de l'environnement à partir de deux points de vue différents, il est possible d'estimer la distance des objets et d'avoir ainsi une image de profondeur, comme illustre la (figure 2.9), qui peut être utilisée pour l'évitement d'obstacle ou la cartographie. Cette méthode [34] suppose toute fois un minimum d'éléments saillants dans l'environnement (ou un minimum de texture) et peut être limitée, par exemple dans un environnement dont les murs sont peints de couleurs uniformes.

b. Les caméras panoramiques

Les caméras panoramiques (catadioptriques) sont constituées d'une caméra standard pointant vers un miroir de révolution (par exemple un simple cône, ou un profil plus complexe qui peut s'adapter à la résolution exacte que l'on veut obtenir sur le panorama), illustré dans la (Figure 2.10). L'image recueillie permet d'avoir une vision de l'environnement sur 360 degrés autour de la caméra. Le secteur angulaire vertical observé dépend de la forme du miroir et peut être adapté aux besoins de chaque application, comme montre la (Figure 2.10).

Ce type de caméra est très pratique pour la navigation car une image prise par une caméra panoramique orientée verticalement permet de caractériser une position, indépendamment de la direction du robot.

Ces caméras sont donc très pratiques lorsqu'on caractérise une position de manière globale, mais peuvent aussi être utilisées pour détecter des amers ou pour estimer le flux optique.

5. Conception et réalisation d'une carte de commande

a. choix de fauteuil roulant

Il existe différents critères pour choisir un fauteuil et de nombreux accessoires en fonction du modèle retenu : La commande, L'assise, Le dossier, Les accoudoirs, Les repose-pieds, Les roues, Les batteries, les accessoires, repose jambe, équipement de route, commandes électriques, repose canne, protège vêtements, ceinture, monte trottoir, rallonge dossier [35].

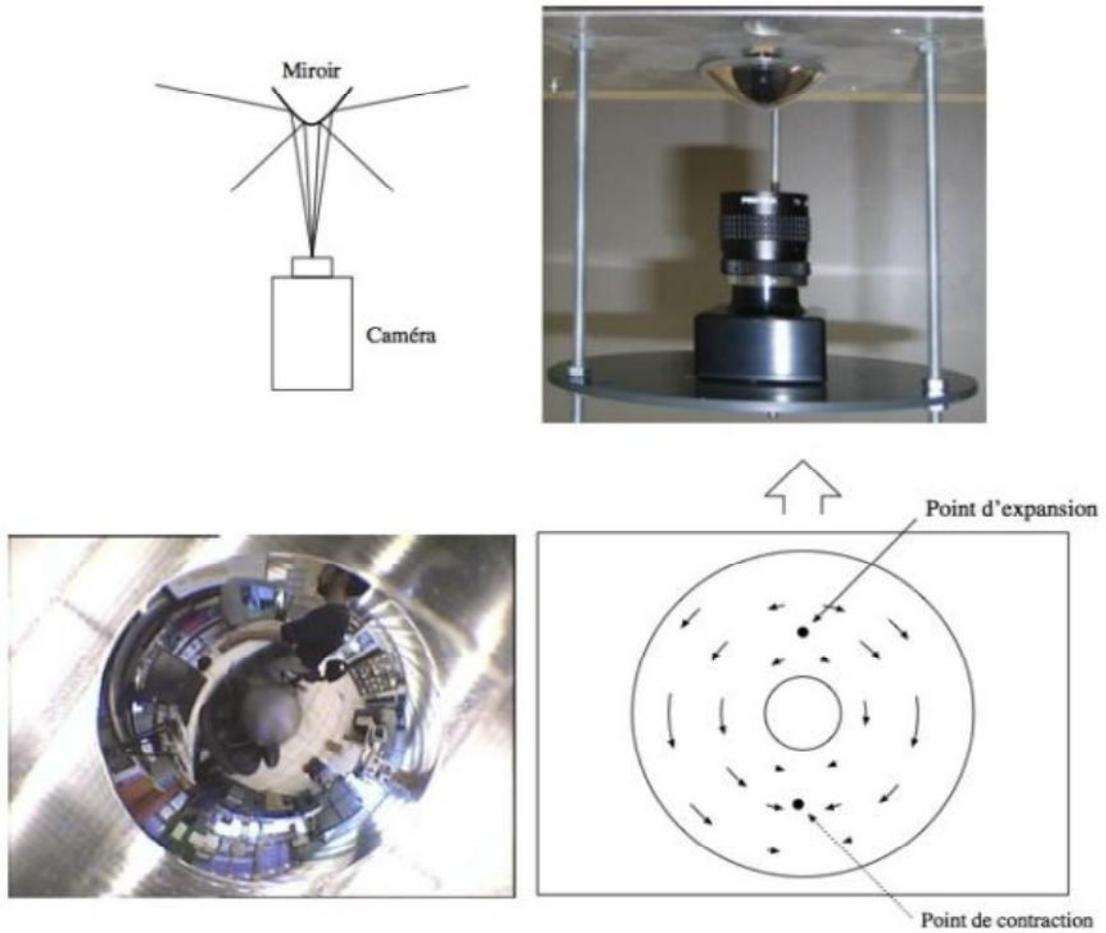


Figure 2.10. Principe des caméras panoramiques catadioptriques.



Figure 2.11. Caractéristiques techniques du notre fauteuil roulant électrique.

Ces nombreux critères doivent être pris en compte. Le fauteuil doit être adapté à la morphologie, et le positionnement étudié en fonction du handicap. Le fauteuil est défini pour

un usage à dominante intérieure ou au contraire extérieure. Le choix s'effectue alors au niveau de deux catégories bien distinctes. On a besoin d'un Fauteuil défini pour un usage à dominante intérieure, pour cela on a choisit le fauteuil illustré dans la (figure 2.11).

Chaque plateforme mobile est constituée d'une partie mécanique et une partie électrique. La partie mécanique représente la structure métallique et les actionneurs [36]. La partie électrique présente les capteurs, les moteurs électriques, la carte de puissance et celle de commande.

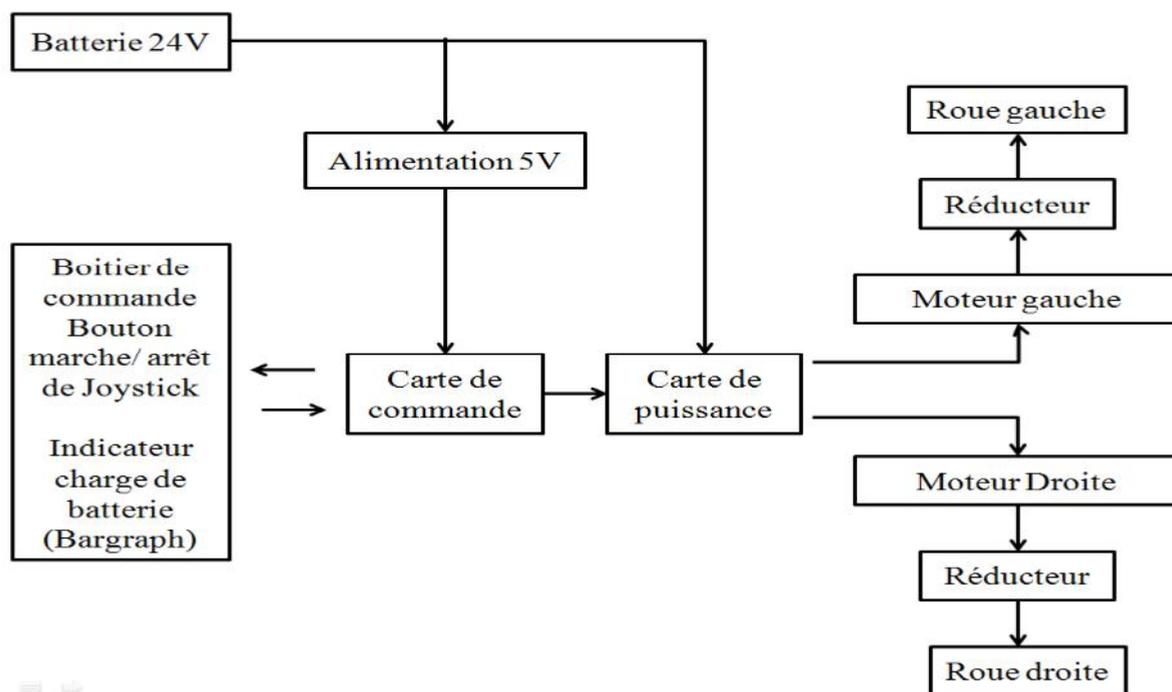


Figure 2.12. Synoptique du fauteuil roulant électrique.

b. choix des capteurs

Dans notre projet on a besoin des capteurs de distance qui assurent la détection d'obstacle et la variation de l'altitude pour cela on a étudié les différents types des capteurs.

Notre attention s'est essentiellement portée sur deux types de technologies : l'Infrarouge et l'Ultrason. Chacune de ces technologies ayant ses avantages et ses inconvénients. Coté Infrarouge, les avantages sont la rapidité de la mesure et le faible angle de détection (qui peut aussi devenir un inconvénient si l'on veut balayer un large espace avec des capteurs fixes.), en revanche ils sont sensibles à l'éclaircissement et à la couleur des objets, même si on arrive maintenant à avoir des capteurs qui permettent à peu près de ne pas en tenir compte. Ils ont aussi une caractéristique de réponse pas forcément linéaire, ce qui rend la conversion de la mesure plus difficile.

Le choix des capteurs dépend de ces caractéristiques :

- *L'étendue de la mesure* : c'est la différence entre le plus petit signal détecté et le plus grand perceptible sans risque de destruction pour le capteur.

- *La sensibilité* : c'est la plus petite variation d'une grandeur physique que peut détecter un capteur.
- *La rapidité* : c'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information prise en compte par la partie commande.
- *La précision* : c'est la capacité de répétabilité d'une information, position, d'une vitesse, etc.

6. Transmission des données

L'époque des communications câblées est révolue. Aujourd'hui, de plus en plus de communications se font sans fil, que ce soit pour le meilleur ou pour le pire. Dans cette partie, nous allons aborder ensemble: la communication entre cartes électroniques. Nous nous intéresserons à un duo de choc, aussi classique qu'efficace : le module radio nRF24L01 et cartes Arduino.

Avant de commencer cette partie, intéressons-nous d'abord au module radio lui-même :

6.1. Le module radio nRF24L01+

Le module radio nRF24L01+ est un module radio tout intégré du fabricant Nordic Semiconductor. Il s'agit d'un module radio intégrant tout le nécessaire pour émettre et recevoir des données sur la gamme de fréquences de 2.4GHz (comme le WiFi ou le Bluetooth) en utilisant le protocole de communication propriétaire de Nordic nommée "ShockBurst". Ce protocole de communication permet au nRF24L01+ d'être considéré comme un modem complet, avec adressage, gestion des erreurs de transmission et retransmission automatique en cas de non-réponse du destinataire.

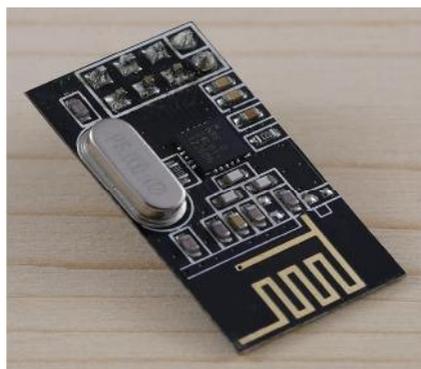


Figure 2.13. Module nRF24L01+ chinois avec antenne "trace".

6.2. Émission et réception

Le plus compliqué avec les modules nRF24L01 est la gestion des tailles de paquets en émission et en réception, car la taille des paquets est fixe. Pour que tout fonctionne correctement, il faut choisir une taille de paquet et s'y tenir pour tous les modules nRF24L01

susceptibles de communiquer entre eux. Si vous ne savez pas quelle taille va faire les données, utilisez la taille maximum de 32 octets comme valeur pour la variable `Mirf.payload`.

L'envoi d'un paquet de données se fait en trois étapes :

```
1 byte paquet[32];
2 strcpy(paquet, "Hello World!"); // Préparation du paquet
3
4 Mirf.send(paquet); // Envoi du paquet
5
6 while(Mirf.isSending()); // Attente de la fin de l'envoi
```

1. Mettre en forme les données pour que le tout donne un joli tableau de `Mirf.payload` octets.
2. Envoyer le paquet avec `Mirf.send()`.
3. Attendre la fin de l'envoi avec une boucle autour de `Mirf.isSending()`.

La réception est plus simple :

```
1 byte paquet[32];
2
3 while(!Mirf.dataReady()){ // On attend de recevoir quelque chose
4 // On attend ...
5}
6
7 Mirf.getData(paquet); // Réception du paquet
```

1. On prépare un tableau d'octets suffisamment grand pour recevoir un paquet de données.
2. On attend de recevoir un paquet avec une boucle autour de `Mirf.dataReady()`.
3. On copie les données du paquet reçu dans le tableau d'octets avec `Mirf.getData()`.

7. Commande des moteurs du fauteuil roulant

Afin de faire fonctionner notre module nous devons :

- Alimenter l'Arduino
- Récupérer les informations de commande des moteurs

Afin d'illustrer nos propos nous avons effectué le schéma suivant, par la suite nous allons en détailler chaque partie comme le montre la(Figure2.14).

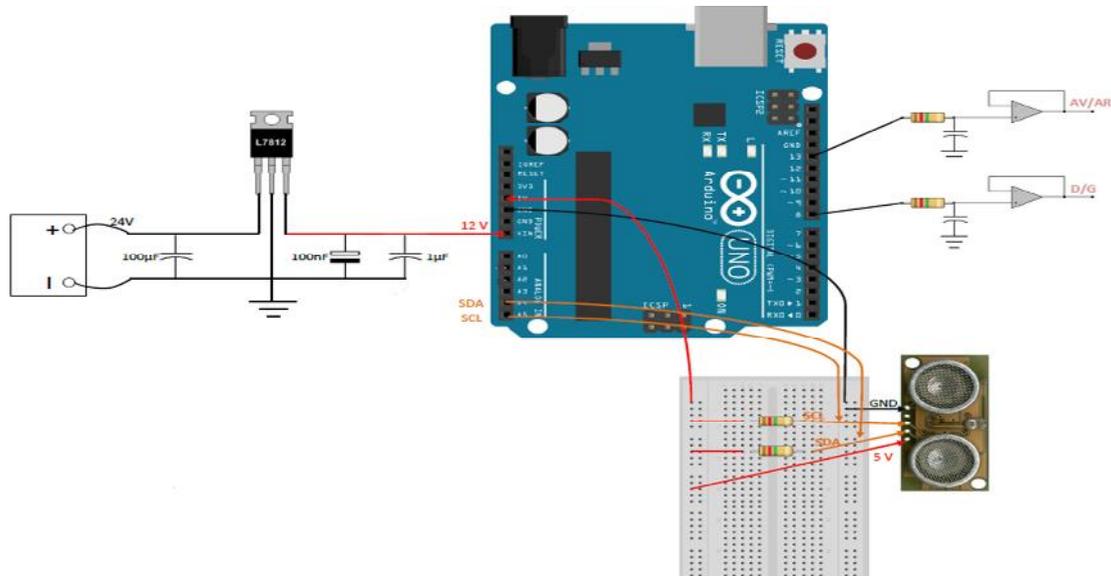


Figure2.14. Vue prototypage du montage de commande

7.1. Alimentation de l'Arduino

Afin d'alimenter l'Arduino nous avons étudié plusieurs possibilités :

- Alimentation à partir des batteries du Fauteuil Roulant Electrique
- Alimentation par batterie externe

Afin d'avoir une autonomie plus élevée de notre module nous avons décidé d'alimenter l'Arduino directement à partir des batteries du fauteuil. La commande du FRE devant elle aussi être alimentée nous avons récupéré la tension à ce niveau, cependant la tension prise à ce point est de 24 V et la tension d'alimentation de l'Arduino est comprise entre 7 V et 12 V. Nous avons donc opté pour un composant permettant d'abaisser la tension de 12 V à 24 V.

Pour ce faire nous avons utilisé le 7812 et réalisé le schéma suivant :

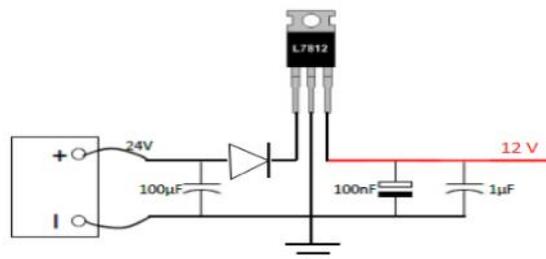


Figure2.15. Vue schématique de l'abaisseur de tension 7812

Nous allons alors chercher les 24 V de la batterie en passant par la commande du FRE. La diode nous permet de protéger le 7812 en cas de mauvais branchement, le condensateur d'entrée est un condensateur de filtrage. La présence des condensateurs en sortie du composant nous permet d'éviter les oscillations et donc d'avoir un résultat stable et fixe.

Cette tension est ensuite envoyée directement dans l'Arduino grâce à un connecteur Jack. L'Arduino possédant un régulateur interne cela nous permet de créer une tension de 5V nécessaire à l'alimentation des capteurs.

7.2. Récupération des informations des moteurs

N'ayant pas de sortie analogique sur l'Arduino il est nécessaire de créer un signal continu à partir d'un signal PWM. Dans un premier temps nous devons interpréter les signaux analogiques trouvés puis les convertir en signaux PWM et à la sortie de l'Arduino nous allons réaliser un filtre RC suivi d'un AOP qui permettra d'abaisser l'impédance si besoin. On réalise donc le schéma suivant (Figure2.16) :

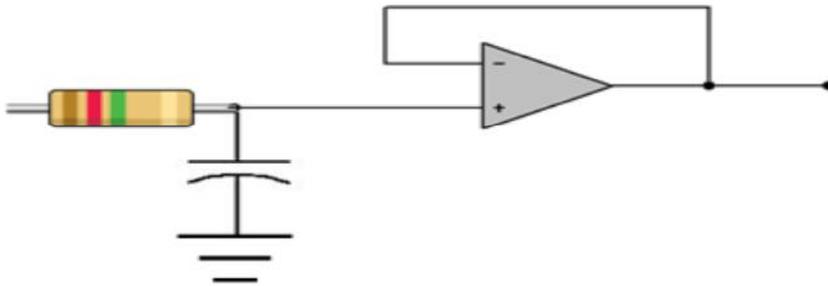


Figure2.16.schéma d'un filtre RC

Les résultats obtenus sont les suivants :

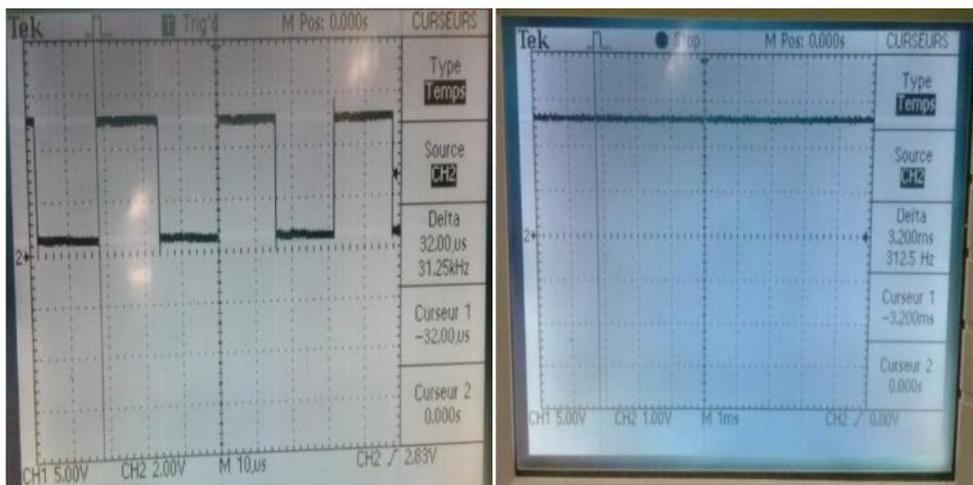


Figure2.17.Vue de signal PWM à la sortie du filtre RC

On observe alors que pour un signal PWM avec un rapport cyclique de 50% à la sortie du filtre RC on obtient une tension constante de 2,5 V. L'état bas du signal PWM est de 0v, et l'état haut est de 5v. Cependant la valeur de l'état haut peut varier en fonction de l'alimentation de l'Arduino, ce qui impactera forcément la valeur à la sortie du filtre RC.

Une fois traités, les signaux seront redirigés vers la carte électronique à partir de laquelle une nappe les reliera jusqu'à la carte électronique du boîtier de commande du FRE.

8. Conclusion

Dans ce chapitre, nous avons présenté les fauteuils roulants électriques et leurs caractéristiques, ainsi que les différents projets qu'ont été réalisés sur les fauteuils roulants intelligents en particulier comme par exemple le projet de VAM et le projet de WAD. Nous avons détaillées les différents types des capteurs et des cameras qu'on va utiliser dans notre projet et on a présenté quelque caractéristique de ces matériaux. Ensuite nous avons vu la méthode de transmission de données avec le module nRF24L01+. Enfin nous avons conclu ce chapitre par la description de la carte de commande des moteurs du fauteuil roulant électrique qu'on va réaliser.

CHAPITRE 3

Planification de chemin du fauteuil

1. Algorithme d'optimisation par essaim de particules (PSO)

Pour donner à notre fauteuil roulant électrique une forme d'intelligence artificielle et aussi pour la planification de son chemin dans l'environnement on a utilisé une approche bio-inspirée, c'est l'optimisation par essaim de particules (OEP) connu sous le nom anglophone de Particle Swarm Optimization (PSO), cette approche a été inspirée de la biologie et de la nature.

L'optimisation par essaims particulaires est une méta-heuristique d'optimisation, proposée par Russel Eberhart et James Kennedy en 1995 [Eberhart et Kennedy, 1995] s'inspirent du monde du vivant. Cette méta-heuristique permettant de simuler le déplacement d'un groupe d'oiseaux migrateurs qui doivent parcourir des longues distances et qui doivent donc optimiser leurs déplacements en termes d'énergie dépensée et du temps, comme par exemple la formation en V.



Figure 3.18. Volée d'Anser en formation en V.

Les techniques d'optimisation inspirées de l'intelligence en essaim sont devenues de plus en plus populaires au cours de la dernière décennie. Ils sont caractérisés par un mode de fonctionnement décentralisé qui imite le comportement des essaims d'insectes sociaux, les troupes d'oiseaux, les écoles de poissons.

Le déplacement de ses animaux en essaim est complexe, sa dynamique obéit à des règles et des facteurs bien spécifiques qu'il s'agit de cerner :

- Chaque individu dispose d'une certaine intelligence limitée (qui lui permet de prendre une décision).
- Chaque individu doit connaître sa position locale et disposer d'information locale de chaque individu se trouvant dans son voisinage.
- Obéir à ces trois règles simples, « rester proche des autres individus », « aller dans une même direction » ou « voler à la même vitesse ».

Tous ses facteurs et règles sont indispensables pour le maintien de la cohésion dans l'essaim, ceci par l'adoption d'un comportement collectif complexe et adaptatif.

La méta-heuristique d'optimisation se base sur la collaboration des individus entre eux : chaque particule se déplace et à chaque itération, la plus proche de l'optimum communique aux autres sa position pour qu'elles modifient leur trajectoire. Cette idée veut qu'un groupe d'individus peu intelligents puisse posséder une organisation globale complexe.

L'avantage de cette approche par rapport aux autres techniques traditionnelles est la robustesse et flexibilité. Ces propriétés en font le succès de l'intelligence en essaim, il aide à la conception des algorithmes de plus en plus pour résoudre des problèmes complexes.

1.1. Principe

L'optimisation par essaim de particules est constituée de deux phases, exploration et exploitation. Le premier est responsable de la détection des régions les plus prometteuses dans l'espace de recherche, tandis que l'autre permet de promouvoir la convergence des particules vers la meilleure solution détectée.

L'OEP repose sur un ensemble d'individus originellement disposés de façon aléatoire et homogène, que nous appellerons dès lors des particules, qui se déplacent dans l'espace de recherche et constituent chacune une solution potentielle. Chaque particule dispose d'une mémoire concernant sa meilleure solution visitée ainsi que la capacité de communiquer avec les particules constituant son entourage. A partir de ces informations, la particule va suivre une tendance faite, d'une part, de sa volonté à retourner vers sa solution optimale, et d'autre part, de son mimétisme par rapport aux solutions trouvées dans son voisinage. A partir des optimums locaux et empiriques, l'ensemble des particules va normalement converger vers la solution optimale globale du problème traité [38] [39].

Un essaim de particules, qui sont des solutions potentielles au problème d'optimisation, survole l'espace de recherche, à la recherche de l'optimum global. Le déplacement d'une particule est influencé par les trois composantes suivantes :

1. Une composante *d'inertie*: la particule tend à suivre sa direction courante de déplacement ;
2. Une composante *cognitive* : la particule tend à se diriger vers le meilleur site par lequel elle est déjà passée ;
3. Une composante *sociale* : la particule tend à se fier à l'expérience de ses congénères et, ainsi, à se diriger vers le meilleur site déjà atteint par ses voisins.

Donc le déplacement de la particule dépend de la meilleure performance des particules voisines (la meilleure position globale), sa meilleure performance (la meilleure position locale) et la position actuelle de la particule. Notons qu'à chaque particule on associe une vitesse, qui détermine la direction de déplacement. C'est celles-ci qui vont faire évoluer la population.

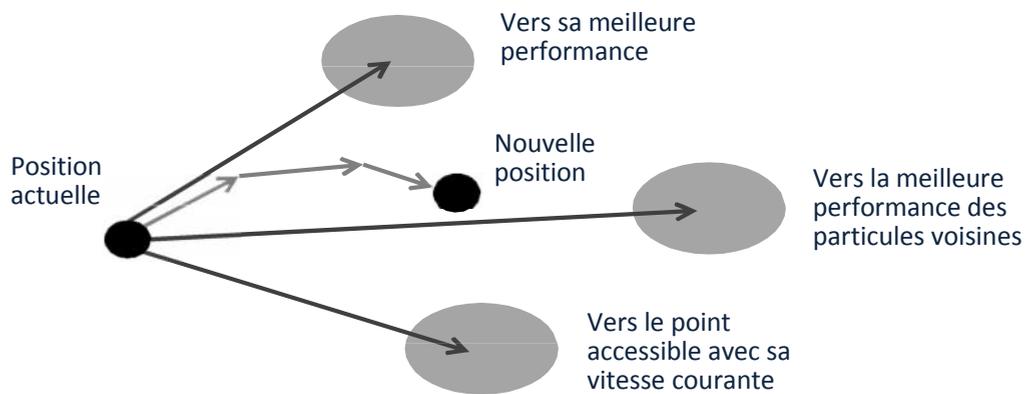


Figure 3.19. Schéma de principe du déplacement d'une particule.

PSO est une méthode d'optimisation stochastique qui est inspiré d'un comportement social des animaux évoluant en essaim. Ce comportement social est modélisé par une équation mathématique permettant de guider les particules durant le processus de déplacement.

1.2. Paramètres de l'algorithme

Six paramètres rentrent en compte :

1.2.1. La dimension du problème

Le problème type sur lequel les systèmes d'essaim de particule sont particulièrement efficaces est la recherche du minimum ou du maximum d'une fonction f non linéaire continue dans un espace à D dimensions [Clerc, 2005]. Cette fonction est la fonction objectif du problème qu'il faut optimiser.

Dans un espace de recherche de dimension D , la particule p de l'essaim est modélisée par son vecteur position X_p et par son vecteur vitesse V_p . La qualité de sa position est déterminée par la valeur de la fonction objectif en ce point.

1.2.2. Le nombre de particules

Il n'y a pas une règle générale pour déterminer le nombre de particules, pour notre cas on a choisie le nombre de particule ($N=20$).

1.2.3. La notion de voisinage

Le voisinage d'une particule est le sous-ensemble de particules de l'essaim avec lequel il a une communication directe. Ce réseau de rapports entre toutes les particules est connu comme la sociométrie, ou la topologie de l'essaim. Il existe deux principaux types de voisinage :

Les voisinages géographiques : les voisins sont considérés comme les particules les plus proches. Cependant, à chaque itération, les nouveaux voisins doivent être recalculés à partir d'une distance prédéfinie dans l'espace de recherche. C'est donc un voisinage dynamique.

Les voisinages sociaux : les voisins sont définis à l'initialisation et ne sont pas modifiés ensuite. C'est le voisinage le plus utilisé, pour plusieurs raisons :

Il est plus simple à programmer et moins coûteux en temps de calcul. En cas de convergence, un voisinage social tend à devenir un voisinage géographique. Pour ce faire, on dispose (virtuellement) les particules en cercle puis, pour la particule étudiée, on inclut progressivement dans ses informatrices, d'abord elle-même, puis les plus proches à sa droite et à sa gauche, jusqu'à atteindre la taille voulue. On peut aussi choisir les informatrices au hasard.

1.2.4. La vitesse maximale et la vitesse minimale

Pour faire déplacer la particule régulièrement on limite la vitesse de déplacement par une vitesse maximale notée (V_{\max}), et une vitesse minimale notée (V_{\min}), pour éviter que le déplacement des particules soit trop rapide, ce qui conduit à sortir de l'espace de recherche, et pour limiter la divergence des particules. Car la particule peut passer à côté d'un optimum sans le marquer.

Pour limiter la vitesse (V) de la particule (P) à l'instant (t) on utilise cette règle :

$$V_p(t) = \begin{cases} V_{\max} & \text{si } V_p(t) > V_{\max} \\ V_{\min} & \text{si } V_p(t) < V_{\min} \end{cases} \quad (1)$$

1.2.5. Les Coefficients de confiance

Les coefficients ρ_1 et ρ_2 de l'équation (2) sont appelés coefficients de confiance, ils permettent de pondérer les tendances des particules à suivre la meilleure particule de la population (meilleure position globale) ou de suivre la meilleure particule de son voisinage (meilleure position locale).

Les coefficients r_1 , r_2 sont des variables aléatoires évaluées à chaque itération suivant une loi uniforme sur le domaine $[0, 1]$ et c_1 , c_2 appelées coefficients d'accélération, ils sont des constantes positives définies par la relation $c_1 + c_2 \leq 4$.

$$\begin{cases} \rho_1 = c_1 r_1 \\ \rho_2 = c_2 r_2 \end{cases} \quad (2)$$

1.2.6. Le facteur d'inertie

Le coefficient d'inertie ω , introduit par Y. Shi et Eberhart [40], contrôle l'influence de la direction de la particule sur le déplacement futur. Le but de l'introduction de ce paramètre est

de réaliser un équilibre entre la recherche locale (exploitation) et la recherche globale (exploration). L'intensité de l'exploration de l'espace de recherche dépend de la valeur du poids d'inertie, une grande valeur de ω facilite l'exploration globale, alors qu'une petite valeur facilite l'exploration locale. Du fait de son influence sur les performances de l'algorithme PSO, le poids d'inertie a suscité un grand intérêt de la part de la communauté des chercheurs. Les chercheurs ont proposé un coefficient d'inertie dynamique qui varie au cours du temps. Il commence par une valeur proche de 0,9 et descend linéairement pour arriver à 0,4 [41]. Cette stratégie a beaucoup amélioré les performances du PSO dans plusieurs problèmes d'optimisation.

Donc on obtient l'équation de la vitesse sous la forme :

$$\vec{v}_{p(t+1)} = \omega \vec{v}_{p(t)} + \rho_1 (\vec{X}_{pbest} - \vec{X}_t) + \rho_2 (\vec{X}_{gbest} - \vec{X}_t) \quad (3)$$

Avec $\rho_1 = c_1 r_1$ et $\rho_2 = c_2 r_2$

On peut ensuite déterminer la position suivante de la particule grâce à la vitesse que l'on vient de calculer :

$$\vec{x}_{p(t+1)} = \vec{X}_p(t) + \vec{v}_{p(t+1)} \quad (4)$$

2. Pseudo code

PSO est une population basée sur un algorithme de recherche et est initialisé avec une population des solutions aléatoires, appelés particules. Le principe de l'algorithme peut être plus facilement visualisé grâce à la figure suivante :

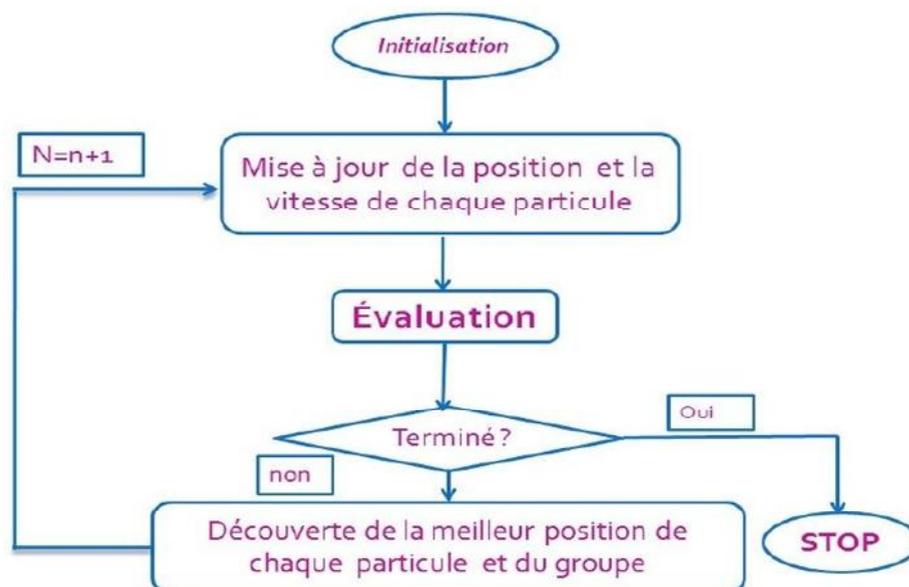


Figure 3.20. Organigramme de la méthode des essais particulaires [42].

Dans l'étape d'initialisation toute particule est initialisée avec une position aléatoire dans l'environnement ainsi que son vitesse de déplacement et les autres paramètres de l'algorithme. Après cette étape, une étape de mise à jour des particules, dans cette étape chaque particule met à jour sa vitesse en utilisant l'équation (3), puis elle met à jour sa position à l'aide de l'équation (4).

Puis une étape d'évaluation consiste à évaluer les particules de la population pour localiser la meilleure particule.

La dernière étape c'est le test d'arrêt, si le critère d'arrêt est satisfait alors l'algorithme se termine sinon on retourne à l'étape 2.

Le critère d'arrêt peut être le nombre maximal d'itérations, ou l'objectif est atteint, ça veut dire qu'il n'y a pas autre position est meilleure que la position actuelle.

Nous avons utilisé 20 particules qui cherchent à optimiser l'objectif. Les positions des particules sont initialisées avec le point de départ, et chaque particule dispose 3 variables à chercher, 2 variables représentent les abscisses dans les axes et sont recherchées par l'algorithme, et la 3ème variable c'est l'angle de rotation calculé par l'équation (5) où x est la différence entre le x de la meilleure position globale et le x de la position actuelle de la particule, et la même chose pour le z .

Comme la dimension du problème ne concerne que deux variables, alors le problème est de dimension $d = 2$.

La valeur de l'angle est comprise entre $[0, 360]$, l'abscisse x entre $[-1/(u/4), -1+(width*u)-(u/4)]$, l'abscisse z entre $[-1/(u/4), -1+(height*u)-(u/4)]$, la vitesse v est comprise entre $[-val, +val]$ où $val = ((width+height/3)*u*0.04)$ et $width$, $height$ sont nombre de case de la largeur et la longueur respectivement et u c'est l'unité utilisée dans la simulation (qui vaut 0.2 dans ce travail).

Le facteur d'inertie est égal à 0.5, le facteur $c1$ est égal à 1, et $c2$ est égal à 2 et enfin le nombre maximal d'itération est égale à 1000.

3. Evitement d'obstacles

Le robot est dans un environnement continu, dans cet environnement il y a des murs (obstacles statiques) et des obstacles dynamiques qui peuvent être des robots ou autres mobiles.

Le robot doit partir à partir du point de départ et arriver à sa destination en évitant tout les obstacles statiques et dynamiques, donc une position dans l'instant t n'est pas valable dans l'instant $t+1$, et pour réaliser ce travail nous avons utilisé notre propre approche, qui est basé sur l'algorithme PSO modifié ainsi que notre propre algorithme pour éviter les obstacles.

Pour l'évitement d'obstacles, nous avons utilisés 2 méthodes différentes, chaque méthode destinée à un type d'obstacle, cependant la méthode de détection de collision avec les obstacles est la même dans les deux cas.

Notre robot est dans un environnement 3D, mais on nous n'intéresse que les deux axes \vec{x} et \vec{z} donc la collision est sur deux axes, et pour détecter cette collision nous avons englobé les obstacles et le robot par des boites enveloppes 2D ce qu'on appel Bounding Box 2D en anglais, donc la collision est faite entre deux BB.

Pour tester la collision, on teste chaque point de la BB du robot et si le point touche la BB de l'obstacle alors il y en a une collision entre le robot et l'obstacle dynamique ou statique (figure 3.21), s'il n'y a aucun point qui touche la BB de l'obstacle alors il n'y a pas une collision.

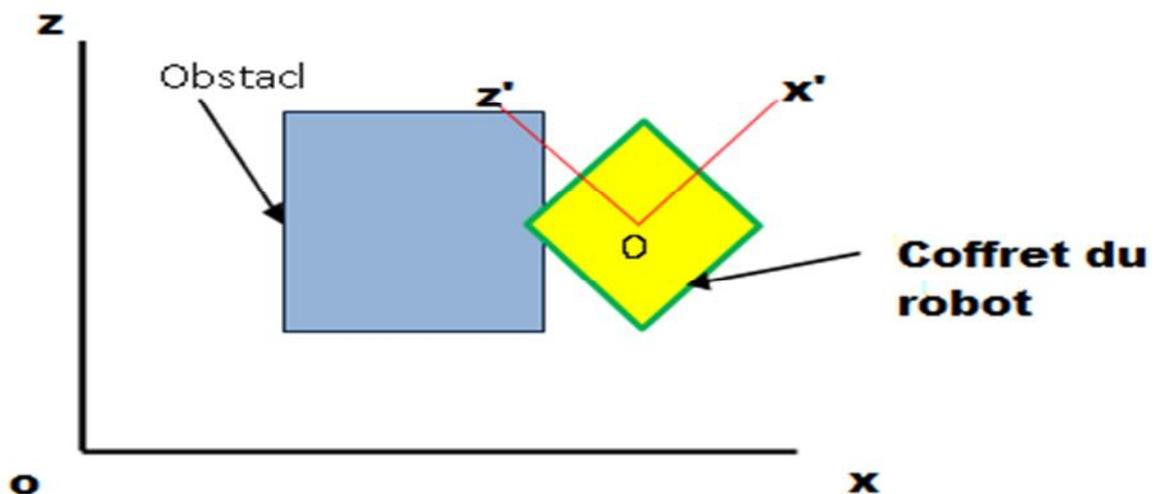


Figure 3.21. Illustration de *collision* entre le *robot* et un obstacle.

Dans notre travail, le *robot* dispose un angle de *rotation*, donc avant de tester les points de la *BB* on fait une rotation des coordonnées de la *BB* dans le nouveau plan en utilisant la *matrice de rotation* 2D (5), et pour calculer les nouvelles coordonnées on utilise les équations (6) et (7) :

$$R(\Theta) = \begin{pmatrix} \cos \Theta & -\sin \Theta \\ \sin \Theta & \cos \Theta \end{pmatrix} \quad (5)$$

$$x' = (Px - Cx) \cdot \cos(\Theta) - (Pz - Cz) \cdot \sin(\Theta) + Cx \quad (6)$$

$$z' = (Px - Cx) \cdot \sin(\Theta) + (Pz - Cz) \cdot \cos(\Theta) + Cz \quad (7)$$

$$\text{Avec } \Theta = 2 \arctan \left(\frac{z}{x + \sqrt{x^2 + z^2}} \right) \quad (8)$$

Où x' et z' sont les nouvelles coordonnées dans le nouveau plan, Cx et Cz sont les coordonnées du centre de la BB du *robot* dans le plan global (plan de l'environnement), et enfin Px et Pz sont les coordonnées du point P .

3.1. Evitement des obstacles statiques

Pour éviter les *obstacles statiques*, nous décomposons l'obstacle en 4 parties équivalentes (4 triangles équivalents), où ils partagent le même point du centre de l'obstacle (figure 3.22). Après cette étape on teste dans quelle partie triangle le *robot* est en *collision*.

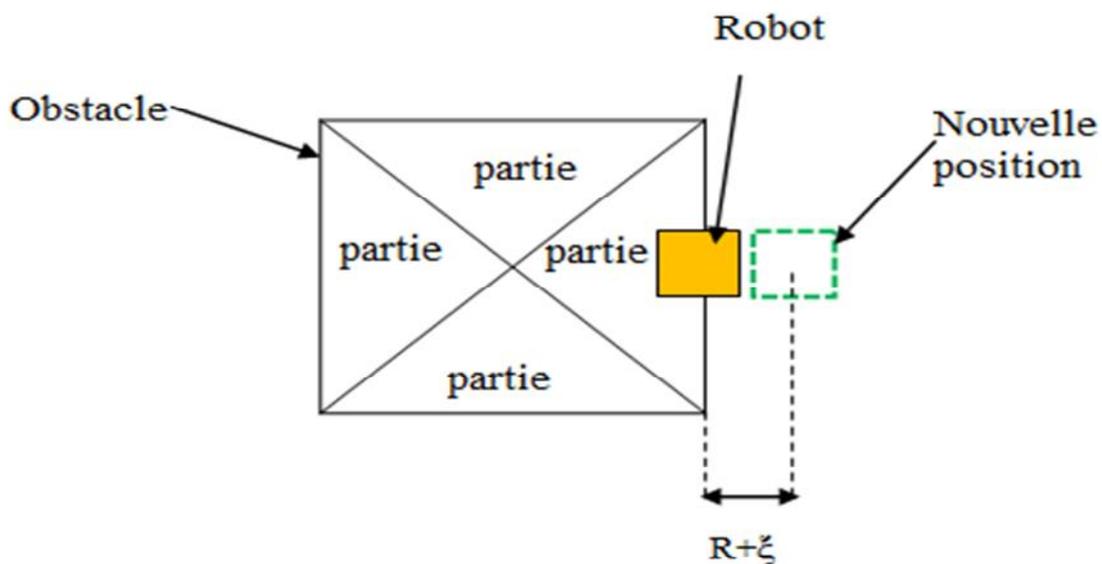


Figure 3.22. Division de l'obstacle statique en 4 parties, et calculer la nouvelle position du *robot*.

Pour faire ce test, on prend un triangle qui est défini par trois points (A , B et C), puis on teste chaque point (P_i) de la BB du *robot* s'il touche ou non le triangle, on répète ce processus pour chaque triangle. Pour tester l'intersection du triangle avec le point, on doit suivre l'algorithme suivant:

- Calculer le vecteur \overrightarrow{PA}

- Calculer le vecteur \overrightarrow{PB}
- Calculer le vecteur \overrightarrow{PC}
- Calculer la sommation des vecteurs $\overrightarrow{somVec} = \overrightarrow{PA} + \overrightarrow{PB} + \overrightarrow{PC}$
- Calculer la longueur du vecteur (normalisation) norme = $\|\overrightarrow{somVec}\|$
- Si norme ≤ 1 alors le point est à l'intérieur du triangle (intersection), sinon il n'y a pas collision.

Après avoir déterminé quelle est la partie en *collision* avec le *robot*, on déplace le *robot* vers une nouvelle position qui est à l'extérieur du triangle en *collision*, la valeur de déplacement est calculée par $R+\xi$ (figure 3.22), où R c'est la valeur de rayon de la *BB* du *robot*, et ξ c'est un petit nombre et est égal à 0.01 (ou en 5 centimètre).

3.2. Evitement des obstacles dynamiques

Les *obstacles dynamiques* ont une *velocity* (vitesse) de déplacement constante, ils peuvent déplacer dans tout l'environnement en évitant la *collision* entre eux et avec les *obstacles statiques* et le *robot* aussi.

Cette fois *l'évitement de l'obstacle* par le *robot* est différent, si le *robot* est en *collision* avec l'obstacle (dynamique) alors nous créons un cercle enveloppe ou *Bounding Circle* en anglais (figure 3.23), ce cercle ou *BC* englobe l'obstacle et est représenté par l'équation (9), cette équation est utilisée pour représenter le *BC* mathématiquement.

$$(x-x_0)(z-z_0) = r^2 \quad (9)$$

Où x_0 et z_0 sont les coordonnées du centre du cercle, et r c'est le rayon du cercle.

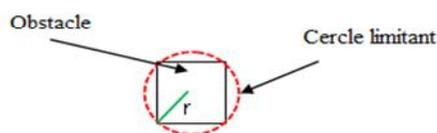


Figure 3.23. Création de cercle limitant pour un obstacle dynamique.

Après cette étape nous créons une ligne droite qui passe par deux points et est représentée par l'équation (10), le premier point est dans la *BB* du *robot* dans la meilleure position globale, et le deuxième point est dans la *BB* du *robot* dans la position courante (position courante de la *particule*), et l'indice de ces points est le même indice du point en *collision* avec l'obstacle (figure 3.24).

$$F(x) = aX + c = Z \quad (10)$$

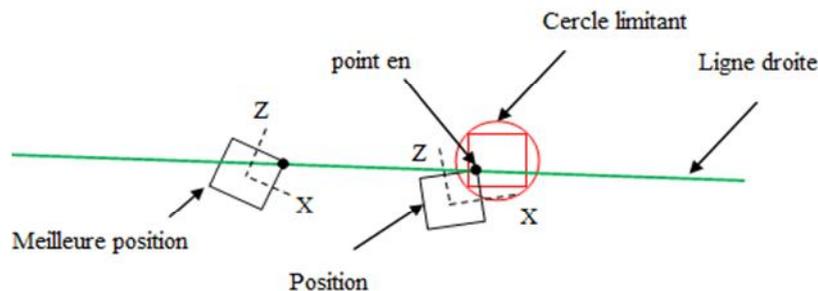


Figure 3.24. Création de la ligne droite entre les deux points, un point est dans la *BB* du *robot* dans la meilleure position globale, et le deuxième point est dans la *BB* du *robot* dans la position courante.

Puis on fait l'intersection entre la droite et le cercle par la résolution de l'ensemble d'équations (9) et (10). En résultat on obtient un point d'intersection ou deux points, dans le cas d'un point d'intersection on garde ce point, dans l'autre cas (2 point d'intersections) on garde le point le plus proche à la meilleure solution globale (figure 3.25).

Pour connaître le point le plus proche à la meilleure solution globale, on calcule la *distance euclidienne* entre chaque point et la meilleure position globale, et à la fin on garde le point qui a une distance minimale.

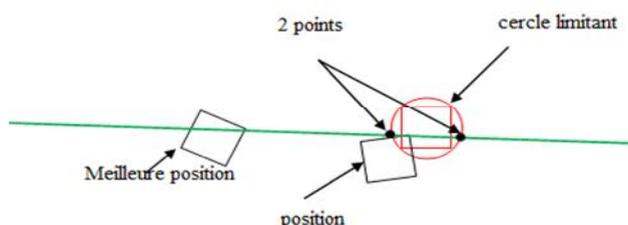


Figure 3.25. Deux points d'intersection entre la droite et le cercle obtenus par la résolution de l'ensemble d'équation (9) et (10).

La dernière étape consiste à déplacer le *robot* vers le point gardé précédemment, et on ajoute R (où R est le rayon du *bounding box* du *robot*) à l'abscisse x et z en utilisant l'équation (11) et (12) (figure 3.26), cette méthode est un cas générale pour n'importe quel angle de *rotation* du *robot* et n'importe quelle direction de déplacement.

$$\text{NewPosX} = x + R \quad (11)$$

$$\text{NewPosZ} = z + R \quad (12)$$

Où a, b et c sont des constantes de l'équation de la ligne (dans notre cas b est égale à 1), P_x et P_z sont les coordonnées du point P .

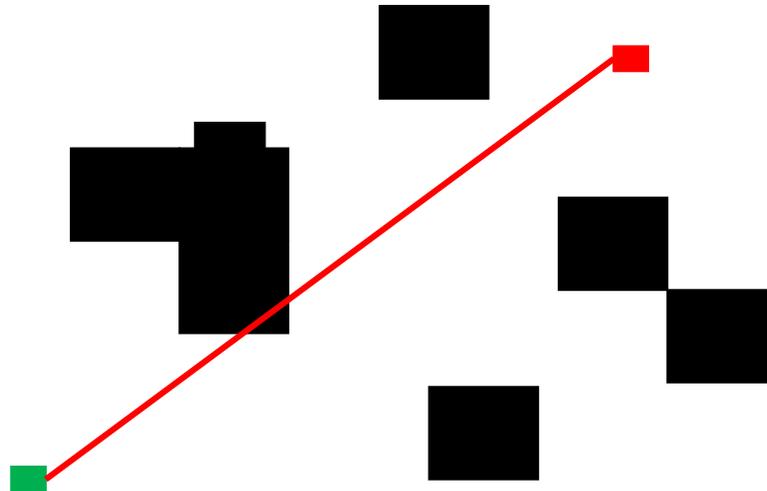


Figure 3.27. Création de plus court chemin virtuel, qui commence par le point de départ (carré vert) et se termine par le point d'arriver (carré rouge).

Encore cette modification nous a montré quelques problèmes dans cas spéciaux. Pour résoudre ces problèmes nous avons multiplié les deux *fitness* par des facteurs $fct1$ et $fct2$ successivement, où la somme de ces deux facteurs est égale à 1, nous avons implémenté ces facteurs de manière dynamique, et dans le démarrage $fct1$ est plus grand que $fct2$.

$$\text{Fitness} = fct1 * \text{distance_euclidienne} + fct2 * \text{distance_point_chemin} \quad (16)$$

Si après N itérations le robot reste bloqué sur une position, on réinitialise la *vitesse* à la valeur initiale, et si après M itérations et réinitialisation de *vitesse* le *robot* reste bloqué alors on décrémente $fct1$ par fct_inc (facteur d'incrémentatation et est fixé à 0.1) et on incrémente $fct2$ par fct_inc . Dans le cas où $fct1$ est arrivé à la valeur minimale et $fct2$ le contraire, dans ce cas on réinitialise $fct1$ et $fct2$ à leurs valeurs initiales.

$$N = \text{nb_iter_max} / (fct_inc * 100) / 4 \quad (17)$$

$$M = N * 2 \quad (18)$$

Où nb_iter_max est le nombre maximal d'itérations, et fct_inc est le facteur d'incrémentatation.

5. Simulation de l'algorithme

La fenêtre du simulateur est divisée en trois sous fenêtres, la première en noire représente le schéma pour le déplacement des particules (lignes avec différentes couleurs), et la ligne rouge c'est notre chemin virtuel, les carrés orangés sont des obstacles dynamiques et enfin les carrés blancs sont des obstacles statiques. La deuxième fenêtre à droite c'est la vue 3D de la simulation où le cube roux est notre robot mobile, les cubes en marron clair sont des obstacles dynamiques, et les cubes texturés en pierre sont des obstacles statiques. La dernière sous

fenêtre est en bas à droite, elle contient quelques informations utilisées dans la simulation comme le compteur N (Vel) et le compteur M (counter), angle de rotation, la taille de *swarm*, etc.

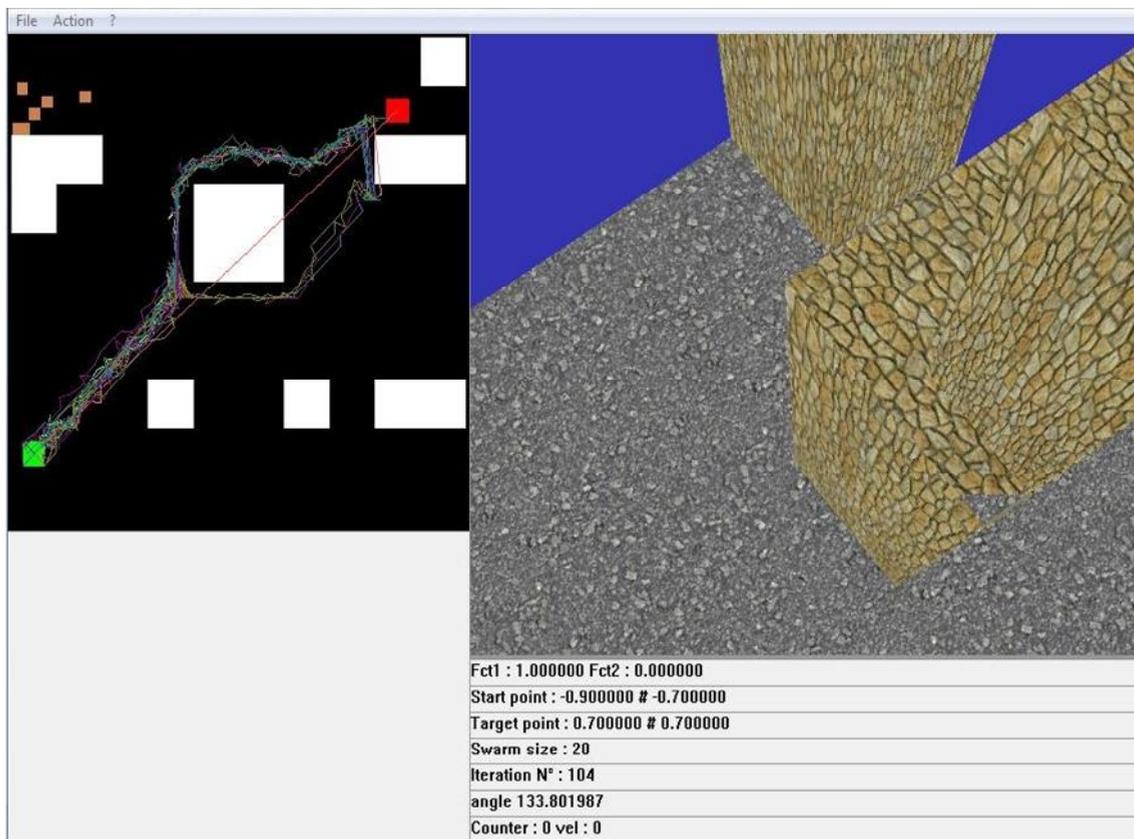


Figure 3.28. Capture d'écran de la simulation de notre approche dans un environnement avec des obstacles et en utilisant 20 particules.

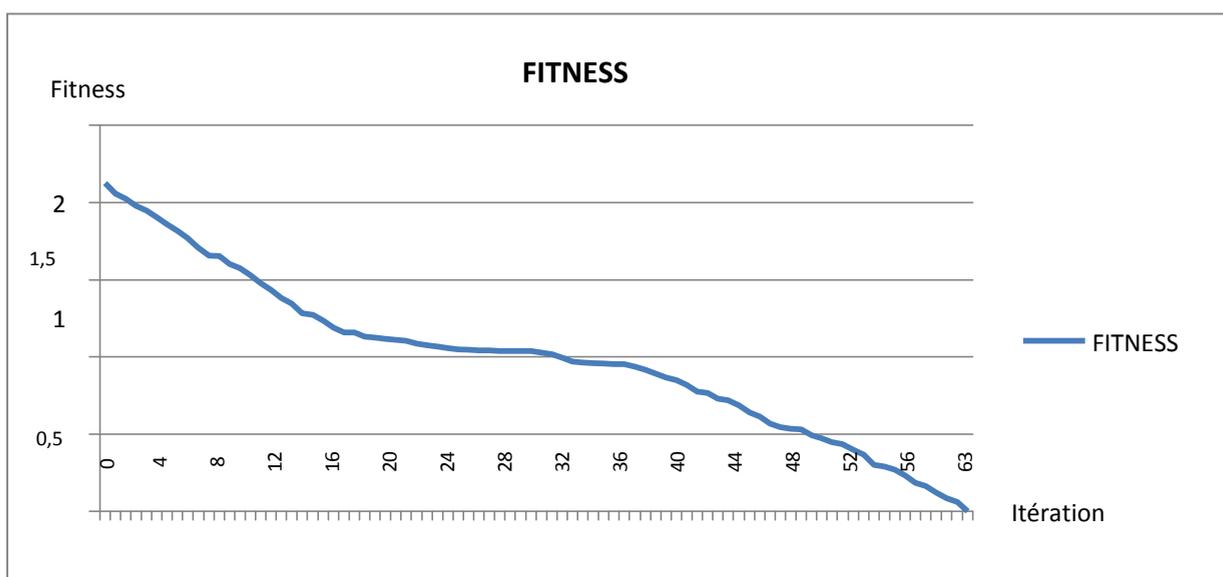


Figure 3.29. Le graph de fitness de la simulation de notre approche dans un environnement avec des obstacles et en utilisant 20 particules.

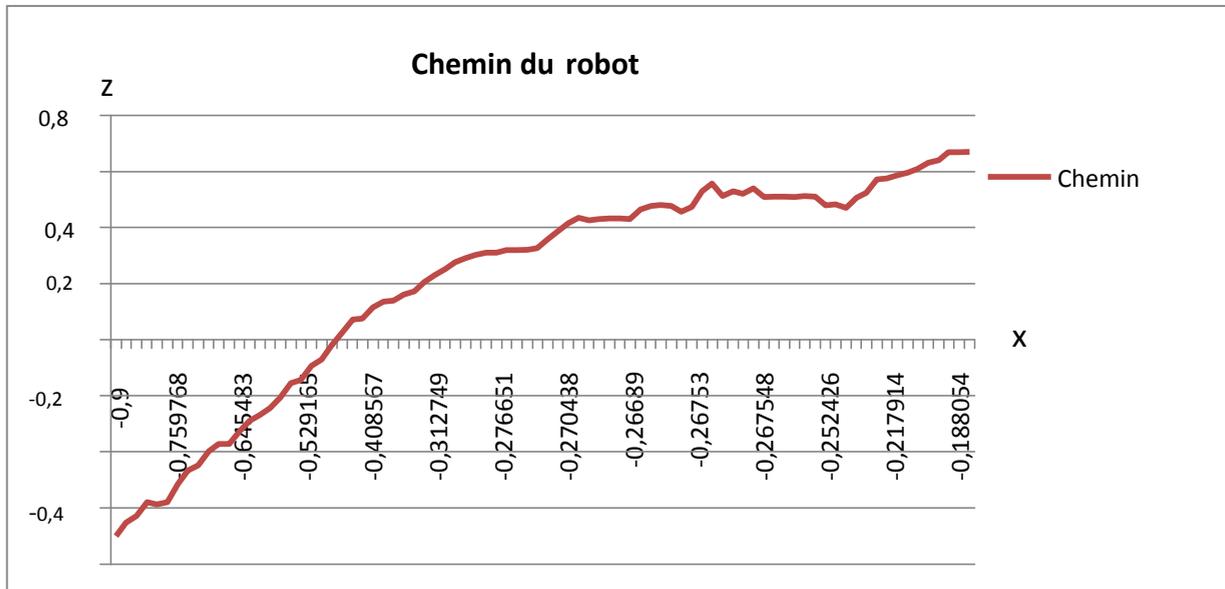


Figure 3.30. Le chemin traversé par le robot dans la simulation de notre approche dans un environnement avec des obstacles et en utilisant 20 particules.

6. Conclusion

Dans ce chapitre, nous avons présenté notre algorithme de planification de chemin du fauteuil intelligent, ainsi notre méthode d'évitement d'obstacles. En particulier, nous avons utilisé l'algorithme PSO pour la planification automatique du chemin, et des méthodes géométriques pour l'évitement d'obstacles. Enfin, nous avons présenté les résultats de simulation de l'algorithme.

CONCLUSION GENERALE

Conclusion générale

Notre travail visait à démontrer l'intérêt de la commande des fauteuils roulants électriques en mode autonome aux personnes ayant un handicap physique lourd. Cette étude nous a permis de mettre en valeur notre sens de l'analyse et de réflexion. Il a fallu combiner nos compétences et faire valoir notre sens de l'équipe pour arriver à nos fins. Ce type d'environnement de travail apporte beaucoup, aide à la compréhension du système et rapproche du monde industriel.

Ce mémoire est scindé en trois chapitres principaux. Au début, nous avons établi un état de l'art sur les robots mobiles en passant par un petit historique de l'évolution de la robotique, ainsi les différents types de robots mobiles. Ensuite nous avons vu les différents types de navigation en robotique, et la fin quelques exemples sur la collaboration en robotique.

Dans les trois derniers chapitres nous avons présenté les fauteuils roulants électriques en présentant les fameux projets réalisés sur ces fauteuils, ainsi que les divers types de capteurs et de caméras les plus utilisés en robotique. Après cette étude, nous avons effectué dans la deuxième partie le choix du fauteuil roulant et les capteurs qu'on doit utiliser.

Pour l'autonomie de notre fauteuil, nous avons utilisé un algorithme bio-inspiré basé sur l'optimisation par essaim de particules (ou PSO). Or, nous avons modifié l'algorithme afin de l'adapter à notre problème.

Les travaux effectués dans ce projet montrent bien la faisabilité de concevoir un algorithme pour la navigation automatique du fauteuil roulant électrique. D'autre part, quelques perspectives à court terme et à long terme ressortent de cette étude. Ainsi, bien que l'optimisation du temps de réponse de commande. Nous pouvons aussi ouvrir un autre axe de recherche consacré à la commande vocale et de commande par détection de mouvement facial du fauteuil roulant électrique.

Bibliographie

- [1] : L'histoire des robots, par Asimov, Isaac.
- [2] : Introduction à la robotique, par Laetitia Matignon Université de Caen.
- [3] : Généralités sur la robotique mobile, par ISTIA, Université Angers.
- [4] : Reconnaissance de gestes et actions pour la collaboration homme-robot sur chaîne de montage par Eva Coupeté, Interface homme-machine [cs.HC]. PSL Research University Français 2016.
- [5]: Conception de systèmes cobotiques industriels Approche cognatique Application à la production pyrotechnique au sein d'Ariane Group, Par Théo Moulières-Seban.
- [6] : Etude des conditions d'acceptation de la collaboration homme-robot en utilisant la réalité virtuelle par Vincent Weistroffer.
- [7] : Collaboration entre les robots intelligents et les objets connectés, opportunités et défis, par M. RABBAH.
- [8]: Matthis Radestock , S Eisenbach. Cordination in Evolving Systems. Departments of computing Imperial College of Science, Technology and Medicine.
- [9] : Coordination de robots pour le transport d'objets par Imene BOUYOUCHEF Université Paris Est Créteil 2013.
- [10] : Planification de trajectoire pour un robot mobile, par Bouali Abdelmalek Ingénieur d'état en électronique Mémoire Présenté pour obtenir le diplôme de magister Electronique, Université El-Hadj Lakhdar – BATNA.
- [11]: J. Borenstein, H. R. Everett, and L. Feng, "Where am I, Sensors and Methods for Mobile Robot Positioning", University of Michigan, 1996.
- [CUE05]: F. Cuesta, A. Ollero, "Intelligent Mobile Robot Navigation", Springer-Verlag, Berlin Heidelberg, 2005.
- [12]: Rajiv Ellepola and Peter Kevesi, 'Mobile robot navigation in a semi-structured environment'. The Fourth International Conference on Control, Automation, Robotics and Vision (ICARCV'96), pp. 914-918, Singapore, 3-6 December, 1996.
- [13]: Contrôle d'un robot mobile par ABDELLAOUI Zakarya et BENKHELIFA Badreddine Université Abou Bekr Belkaïd de Tlemcen 2016.
- [14]: Akihisa Ohya, Akio Kosaka and Avi Kak, 'Vision-based navigation of mobile robot with obstacle avoidance by single camera vision and ultrasonic sensing'. IEEE/RSJ, International.

- [15]: Tobias Einsele and Georg Färber, 'Real-time self-localization in unknown indoor environments using a panorama laser range finder'. IEEE/RSJ, Conference on Intelligent Robots and Systems, IROS'97, pp. 704-711, Grenoble, France, 7-11 September 1997.
- [16]: Approche de Navigation Coopérative et Autonome des Robots Mobiles (Application sur un chantier de construction) Par Mr Abdelmadjid BENMACHICHE BADJI MOKHTAR UNIVERSITY –ANNABA 2016.
- [17] N. OUADAH, M. HAMERLAIN et M.S. Djouadi 2003 'Positionnement d'un bras manipulateur par une commande référence vision', CISTEMA'2003 Conférence Internationale sur STEMA, 27-29 Septembre 2003, Tlemcen.
- [18] E. Fabrizi, et A. Saffiotti, "Augmenting topology-based maps with geometric information." *Robotics and Autonomous Systems*, 40(23) :9197, 2002.
- [19] D. Dufourd, "Des Cartes Combinatoires Pour La Construction Automatique De Modèles D'Environnement Par Un Robot Mobile". Thèse de doctorat, Institut National Polytechnique de Toulouse. 2.4.2. (2005).
- [20] S. Thrun, "Robotic mapping : A survey," In Lakemeyer G. et Nebel, B., éditeurs : *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
- [21] Newell, A., & Simon, H. A. *Computers in psychology* 1972.
- [22] R. Brooks, "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation*, vol. 2, no.1, pp.14-23, 1986.
- [23] T. Duckett, et A. Saffiotti, "Building globally consistent gridmaps from topologies." In *Proceedings of the International Proc. IFAC Symposium on Robot Control*, pp. 357-361, Wien, Austria, 2000.
- [24] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, et S. Teller, "An atlas framework for scalable mapping." In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1899-1906, 2003
- [25] Watson N., Woods B. (2008) A social and technological history of the wheelchair. In : *Le Fauteuil Roulant, Actes des 21ème Entretiens de la Fondation Garches.*, pp. 9-20. Watson N., Woods B. (2008) A social and technological history of the wheelchair. In : *Le Fauteuil Roulant, Actes des 21ème Entretiens de la Fondation Garches.*, pp. 9-20.
- [26] Vignier N., Ravaud J.F. (2008) Les utilisateurs de fauteuil roulant en France : aspects socioépidémiologiques. In : *Le Fauteuil roulant. Actes des 21^{ème} entretiens de la Fondation Garches.* Pp. 21-33.
- [27] Yanco, H. A. (1998). *Wheelesley: A robotic wheelchair system: Indoor navigation and user interface*. *Lecture Notes in Computer Science*, 1458, 256.
- [28] J-W Min, K Lee, S-C Lim, D-S Kwon, —Human-Friendly Interfaces of Wheelchair Robotic System for Handicapped Persons“, *Proceedings of the 2002 IEEE/RSJ Intl.*

Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland, October 2002, pp 1505-1510.

[29] P Mallet, G Schöner —WAD Project where Attractor Dynamics aids wheelchair navigation“. Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems. EPFL, Lausanne, Switzerland October 2002.

[30] E Bicho, P Mallet and G. Schöner, Using Attractor Dynamics to control autonomous vehicle motion, “In Proceedings of IECON’98, IEEE Industrial Electronics Society, pp. 1176-1181, 1998.

[31] Levine, S. P., Bell, D. A., Jaros, L. A., Simpson, R. C., Koren, Y., Member, S., & Borenstein, J.(1999). The NavChair assistive wheelchair navigation system. IEEE Transactions on Rehabilitation Engineering, 7, 443–451, 1999.

[32] ES. Boy, CL. Teo, E. Burdet. “Collaborative wheelchair assistant”. Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2002 Sep 30 Oct 5; Lausanne, Switzerland. Piscataway (NJ): IEEE; 2002. p. 1511–16.

[33] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In IEEE International Conference on Robotics and Automation (ICRA99), May 2005.

[34] R. I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN : 0521540518, second edition, 2004. Cité page 49.

[35] U. Borgolte, H. Hoyer, C. Buehler, H. Heck, R. Hoelper. “Architectural concepts of a semi-autonomous wheelchair”. J Intell Robotic Syst.1998;22(3/4):233–53.

[36] Wang, H., Ishimatsu, T.:”Vision-based Navigation for an Electric Wheelchair Using Ceiling Light Landmark”, Journal of Intelligent and Robotic Systems, 41, 4, 283-314, 2005.

[Eberhart et Kennedy, 1995] Kennedy, J., and R. Eberhart. "Particle Swarm Optimization." Proceedings of the IEEE International Conference on Neural Networks. Perth, Australia, 1995, pp. 1942–1945.

[38] Utilisation des méthodes d'optimisations métaheuristiques pour la résolution du problème de répartition optimale de la puissance dans les réseaux électriques, par Abdelmalek Gacem thèse magister Centre Universitaire d'El-oued.

[39] Jean-Jacques E. Slotine, Weiping Li, "Applied Nonlinear Control"

[Clerc, 2005]

[40] Y. Shi and R. C. Eberhart, “A modified particle swarm optimizer,” in Proc. IEEE Congress on Evolutionary Computation (CEC’98), Anchorage, AK, 1998, pp. 69-73.

[41] Particle Swarm Optimization with Adaptive Inertia Weight Sameh Kessentini and Dominique Barchiesi

[42] Optimisation par la méthode des essais particuliers d'une fonction trigonométrique, par BOUZID Allal El Moubarek, <http://www.scribd.com/doc/33669521/essai-particulaire-PSO>, 2008-2009.

Wébographie

[S1]: <http://www.robot24.fr/definition-historique-de-la-robotique/> (mai 2018).

[S2]: <https://sites.google.com/site/3agaelleamelie/1967-shakey-premier-robot-mobile-contrôle-par-ordinateurstanford-research-institute> (mai 2018).

[S3]: <https://fr.wikipedia.org/wiki/Robotique#Historique> (mai 2018).

[S4]: <https://www.generation-nt.com/robotique-collaborative-cobots-cobotique-actualite-1954068.html> (mai 2018).

[S5] l'encyclopédie libre Wikipédia, [<http://fr.wikipedia.org/wiki/Navigation>] (mai 2018).

[S6] <http://fr.wikipedia.org/wiki/Capteur> (Mars 2012)

[S7] <http://www.robot-electronics.co.uk/htm/srf08tech.shtml> (Février 2012)

[S8] http://fr.wikipedia.org/wiki/Capteur_photographique (Mars 2012)

[S9] <http://www.absolut-photo.com/cours/capteur/ccd-cmos.php> (Mars 2012)