

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Spécialité : **Commande des systèmes**
Filière : **Génie électrique**

Présenté par
OUMOUSSA Yacine
REBIAI Nazli

Mémoire dirigé par **Mr. MELLAH Rabah**

Thème

Commande à distance d'un robot mobile

Mémoire soutenu publiquement le 15/09/2016 devant le jury composé de :

ALI BEY Mohamed
MAB, UMMTO, Président

MELLAH Rabah
MCA, UMMTO, Encadreur

ZAABOT Zohra
MAB, UMMTO, Examineur

CHELLI Takfarinas
MAA, UMMTO, Examineur

Remerciements

Nous tenons à remercier notre DIEU, le tout puissant, de nous avoir donné la santé et la volonté pour compléter ce modeste travail.

Au terme de ce modeste travail nous tenons à remercier chaleureusement tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet de fin d'étude.

Nous tenons à remercier notre promoteur, monsieur **MELLAH.R.**, pour ses conseils précieux et pour toutes les commodités et aisances qu'il nous a apportées durant notre étude et réalisation de ce projet.

Tout notre respect et nos remerciements vont vers les membres de jury qui vont pleinement consacrer leurs temps et leurs attention afin d'évaluer notre travail, qui espérons le sera à la hauteur de leur attente.

Tous nos infinis remerciements vont à tous les enseignants qui ont contribué à notre formation durant notre cursus universitaire.

Et enfin, que nos chers parents et familles, et bien avant tout, trouvent ici l'expression de nos remerciement les plus sincères et les plus profonds en reconnaissance de leurs sacrifices, aides, soutien et encouragement afin de nous assurer cette formation dans les meilleurs conditions.

Dédicaces

A mes parents

Pour les sacrifices déployés à mon égard ; pour leur patience
Leur amour et confiance en moi.

Ils ont tout fait pour mon bonheur et ma réussite,
Qu'ils trouvent dans ce modeste travail, le témoignage de ma
Profonde affection et de mon attachement indéfectible.
Nulle dédicace ne puisse exprimer ce que je leur dois
Que dieu leur réserve la bonne santé et une longue vie.

A mon frère et ma sœur

A mes oncles

A ma grande famille

A la mémoire de mon grand père

A mes amis

En témoignage de mes sincères reconnaissances pour les efforts
Qu'ils ont consentis pour me soutenir au cours de mes études.

A tous mes camarades de la promotion « automatique »

2015\2016

NAZLI

Dédicaces

A mes parents

Pour les sacrifices déployés à mon égard ; pour leur patience

Leur amour et confiance en moi.

Ils ont tout fait pour mon bonheur et ma réussite,

Qu'ils trouvent dans ce modeste travail, le témoignage de ma

Profonde affection et de mon attachement indéfectible.

Nulle dédicace ne puisse exprimer ce que je leur dois

Que dieu leur réserve la bonne santé et une longue vie.

A mon frère et mes sœurs

A mes oncles

A ma grande famille

A la mémoire de mon grand père

A mes amis

En témoignage de mes sincères reconnaissances pour les efforts

Qu'ils ont consentis pour me soutenir au cours de mes études.

A tous mes camarades de la promotion « automatique »

2015\2016

Introduction générale.....	1
----------------------------	---

Chapitre I : Robotique générale

Introduction.....	3
I.1 Définition.....	4
I.1.1 Définition de la robotique.....	4
I.1.2 Etymologie du mot robot.....	4
I.3 Historique de la robotique.....	6
I.4 Types de robots.....	7
I.4.1 Robots industriels (manipulateurs).....	7
I.4.2 Robots humanoïdes	8
I.4.3 Robots mobiles (explorateurs).....	8
I.4.3.1 Robots mobiles à déplacement articulé	9
I.4.3.2 Robots mobiles à roues	9
a) Classification des types de roues	10
b) Principales structures cinématiques	11
c) Modélisation des robots mobiles selon leurs différentes catégories ...	11
d) Modèle cinématique du robot unicycle	12
e) Modèle cinématique du robot tricycle et voiture	14
f) Modèle cinématique du robot omnidirectionnel.....	16
Conclusion.....	17

Chapitre II : Localisation et navigation d'un robot mobile

Introduction.....	18
II.1 Perception.....	19
II.2 Localisation.....	19
II.2.1 Localisation relative.....	20
II.2.1.1 Localisation par odométrie.....	20

a) Modélisation des erreurs odométriques.....	21
II.2.1.2 Localisation inertiel.....	22
a) Systèmes inertiels.....	22
b) Capteurs inertiels.....	23
c) Erreurs des capteurs inertiels.....	24
II.2.2 Localisation absolue.....	24
II.2.2.1 Types de repères utilisés.....	25
a) Localisation par compas magnétique.....	25
b) Localisation par balises actives.....	25
c) Localisation grâce à des repères passifs.....	25
d) Localisation basée sur un modèle.....	26
II.2.2.2 Méthodes de calcul de position et d'orientation.....	27
a) Trilatération (multilatération).....	27
b) Triangulation.....	27
II.2.3 Localisation hybride.....	28
II.3 Modélisation de l'environnement d'intérieur pour la navigation.....	29
II.3.1 Cartes métriques.....	30
II.3.2 Cartes topologiques.....	31
II.3.3 Grilles d'occupations.....	32
II.3.4 Cartes hybrides.....	32
II.4 Navigation.....	33
II.4.1 Principales méthodes réactives de planification de trajectoires de véhicule autonomes.....	34
II.4.1.1 Méthode des champs de potentiels.....	34
II.4.1.2 Méthodes basées sur la connectivité de l'espace navigable.....	35
II.4.1.3 Méthodes des tentacules.....	36
II.4.1.4 Méthodes des champs de braquage constant.....	37
II.4.1.5 Fenêtre dynamique.....	38
II.4.1.6 Méthodes basées sur un modèle de prédiction.....	38
II.4.2 Notre approche de navigation.....	39
Conclusion.....	40

Chapitre III: Description du Dr. Robot I90

Introduction.....	41
III.1 Définition.....	41
III.2 Caractéristiques.....	42
III.3 Architecture de robot.....	43
III.4 Modules matérielles du robot mobile I90.....	43
III.4.1 Partie mécanique	43
III.4.1.1 Châssis mécanique I 90-ME.....	43
III.4.1.2 Moyens de locomotion.....	44
a) Roues motrices.....	44
b) Roue folle.....	44
III.4.1.3 Moteurs.....	45
a) Moteur à courant continu.....	45
b) Servomoteurs RC séparés.....	46
III.4.2 Partie électronique.....	46
III.4.2.1 Capteurs.....	46
a) Capteurs extéroceptifs.....	46
b) Capteurs proprioceptifs.....	49
c) Autres capteurs.....	50
III.4.2.2 LED.....	51
III.4.2.3 Gestion de l'alimentation et le système de recharge.....	51
III.4.2.4 Câble de connexion-CCR2150.....	52
III.4.2.5 Ecran couleur tactile LCD PDM1950.....	52
III.4.3 Partie informatique.....	52
III.4.3.1 Contrôleur du mouvement et des sensations du Robot-PMS5005.....	52
III.4.3.2 Contrôleur Multimédia PMB5010.....	53
III.4.3.3 Interface MCR3210.....	53
III.4.3.4 Joystick de contrôle.....	54
III.4.3.5 Communication wifi.....	54
III.4.3.6 Modem.....	55
III.5 Modules logiciels.....	55

III.5.1 Logiciel SDK.....	55
Conclusion.....	58

Chapitre IV : La commande du Dr. Robot I90

Introduction.....	59
IV.1 Langage de Programmation C#.....	59
IV.1.1 Pourquoi l'orienté objet.....	59
IV.1.2 Définition.....	59
IV.1.2.1 Les données.....	60
IV.1.2.2 les instructions qui les manipulent.....	60
IV.1.3 Classes, interfaces et événement.....	61
IV.1.3.1 Notion d'objet.....	61
IV.1.3.2 Définition d'une classe.....	61
IV.1.3.3 Interface.....	62
IV.1.3.4 Evénements.....	62
IV.2 Programmation élaboré.....	63
IV.2.1 Création d'une interface graphique	63
IV.2.2 Récupérations des données de positions des moteurs, et du capteur à ultrason (labels).....	65
IV.2.2.1 position des moteurs	65
IV.2.2.2 Capteur sonar	66
IV.2.3 Les différentes opérations utilisées pour le control de mouvement des moteurs (boutons).....	66
IV.2.3.1 Marche Avant	67
IV.2.3.2 A droite	67
IV.2.3.3 A gauche	67
IV.2.3.4 Arrière	68
IV.2.3.5 Arrêt de robot (bouton stop).....	68
IV.2.3.6 Turner 90°	68
IV.2.3.7 Turner -90°.....	68
IV.3 Enchaînement des mouvements	69

IV.4 Organigramme du fonctionnement du robot.....	69
IV.5 Fonctionnement de robot.....	71
Conclusion.....	72
Conclusion générale.....	73
Bibliographies	

Figure I.1	Architecture d'un Robot.....	5
Figure I.2	Robot manipulateur.....	7
Figure I.3	Robot humanoïde	8
Figure I.4	Robot mobile	9
Figure I.5	Roue centrée et roue décentrée	10
Figure I.6	Roue tronc sphériques et suédoise	10
Figure I.7	Repérage d'un robot mobile.....	12
Figure I.8	Robot mobile de type uni cycle	13
Figure I.9	Modèle cinématique en posture du robot de type uni cycle.....	13
Figure I.10	Robot mobile de type tricycle.....	14
Figure I.11	Modèle cinématique en posture du robot de type tricycle.....	15
Figure I.12	Un robot mobile de type voiture.....	16
Figure I.13	Représentation d'un robot mobile omnidirectionnel.....	16
Figure II.1	Schéma fonctionnelle d'un véhicule automatisé.....	18
Figure II.2	La Trilatération.....	27
Figure II.3	Principe de la triangulation.....	28
Figure II.4	Exemple de représentation d'un environnement avec carte métrique.....	30
Figure II.5	Exemple de représentation topologique d'un environnement constitué de couloir (c) et de pièces (P).....	31
Figure II.6	Principe de la méthode des champs des potentiels : (a) champs attractifs, (b) champs répulsifs, Illustration d'évitement d'obstacle.....	35
Figure II.7	Graphes de visibilité.....	36
Figure II.8	Diagramme de Voronoï.....	36
Figure II.9	Principe de la méthode des tentacules.....	37
Figure II.10	Principe de la méthode des champs de braquage constant : (a) milieu non structuré, (b) milieu structuré.....	37
Figure II.11	Prédiction de trajectoire : interpolation polynomiale.....	39
Figure II.12	Principe de la méthode utilisé dans notre application.....	39
Figure III.1	Dr.Robot I90.....	41
Figure III.2	Roue motrice.....	44
Figure III.3	Roue folle.....	44
Figure III.4	Module de control DM5253.....	45

Figure III.5	Module à ultrason DUR5200.....	47
Figure III.6	Module à infrarouge GP2Y0A21YK.....	48
Figure III.7	Caméra panoramique PTZ.....	48
Figure III.8	Module capteur d'accélération et tilt-DTA5102.....	49
Figure III.9	Module de détection de mouvement humain-DHM5150.....	51
Figure III.10	LED.....	51
Figure III.11	Câble de connexion.....	52
Figure III.12	Contrôleur du mouvement et des sensations-PMS5005.....	53
Figure III.13	Contrôleur Multimédia PMB5010.....	53
Figure III.14	Interface MCR3210.....	54
Figure III.15	Module WFS802 série (802.11g).....	55
Figure III.16	L'architecture du système Robot.....	57
Figure IV.1	Nouveau projet C#.....	64
Figure IV.2	Application Windows Form.....	64
Figure IV.3	Interface graphique.....	65
Figure IV.4	Enchaînement des mouvements.....	70
Figure IV.5	Interface de connexion.....	71
Figure IV.6	Les deux points d'accès au robot.....	71

Introduction générale

Une augmentation sensible de l'intérêt porté à la robotique mobile est constatée depuis quelques années. Plus précisément, plusieurs groupes de laboratoires et d'industriels se sont associés en vue de démontrer qu'il est possible de mettre au point des moyens de transport automatisés.

La robotique mobile autonome est un axe de recherche qui vise à donner à une machine la capacité de se mouvoir dans un environnement sans assistance ni intervention humaine. Bien souvent, quand on parle de robotique mobile, on sous entend robots mobiles à roues. Ce sont en effet les systèmes les plus étudié [1].

Parmi les capacités décisionnelles, on peut citer la planification de trajectoire et de mouvement, c'est-à-dire la capacité de déterminer le mouvement qui permet de passer d'une position du robot à une autre en respectant les contraintes de déplacement du robot (cinématiques et dynamiques) et en évitant les obstacles de l'environnement.

Cependant, toutes ces applications ne pouvaient être mises en pratique sans en avoir associé aux robots mobiles le concept de la commande à distance.

Le développement des moyens informatiques dont disposent les chercheurs en robotique rend de plus en plus facile l'utilisation d'algorithmes puissants pour donner aux robots une capacité d'autonomie.

Contrairement à la programmation linéaire (procédurale), qui est définie comme une programmation impérative (suite d'instruction les une à la suite des autres) à laquelle on ajoute la possibilité de créer des fonctions et des procédures effectuant un traitement générique. Une programmation orienté objet était créer afin de permettre une approche de la programmation un peu plus proche du raisonnement humain. C'est une technique d'organisation d'un programme en le groupant en objets. Dans notre cas sont des éléments individuels comportant des informations (valeurs de données) et des fonctionnalités [2].

L'utilisation de langage orienté objet C# est basés sur la clarté du code et la facilité d'ajout des éléments sans avoir à le modifier. Ce qui nous à facilité la programmation du robot mobile Dr. Robot I90 pour une navigation autonome.

Dans ce mémoire, on s'est intéressé à la commande du robot mobile « *Dr. Robot I90* ». L'idée est d'élaborer un programme permettant de piloter le robot d'un point de départ à un point d'arriver selon une planification de trajectoire et de mouvements bien définies tout en évitant un obstacle, ce programme est effectué à base de langage de programmation orienté objet C#.

Le mémoire est organisé comme suit :

Le premier chapitre, est consacré à des généralités sur la robotique, ainsi que les modèles cinématiques des robots mobiles selon leurs différentes catégories.

Le second chapitre est consacré aux différentes méthodes de localisation et de navigation des robots mobiles, ainsi notre méthode utilisé pour la navigation (évitement d'obstacle).

Le troisième chapitre est consacré la description matériels et logiciels. Ainsi, les caractéristiques du notre robot mobile « *Dr. Robot I90* ».

Au quatrième chapitre, on présente le langage de programmation utiliser (C#), ainsi que les différentes étapes de notre application.

Le mémoire se termine par une conclusion générale.

Chapitre I

Introduction :

La répétitivité, la difficulté, et la dangerosité de divers travaux depuis les années 70 sont les principales raisons pour lesquelles les chercheurs et les scientifiques se sont intéressés à fabriquer un nouveau type de machines appelé robot [1]. Cette machine est un dispositif mécanique qui est capable d'exécuter une variété de tâches sur commande ou selon des instructions programmées à l'avance [3].

Ces dernières années, la robotique a un impact considérable sur de nombreux aspects de la vie moderne [4]. Les deux premières orientations des robots étaient de répondre aux besoins de l'industrie manufacturière ainsi que l'industrie en milieux hostiles à l'homme.

La robotique industrielle a répondu dans un premier temps au besoin de manipuler de manière répétitive et automatique des objets entre les machines de fabrication (peinture, soudure,...), elle est complétée maintenant dans le domaine manufacturier par des robots mobiles de transport ou d'assistance.

La robotique en milieu hostile répond aux besoins de manipulation d'objets à distance (déménagement, lutte contre le terrorisme,...) sans l'intervention de l'homme du fait de leur dangerosité (chimie, explosif), de la nocivité du milieu (nucléaire) ou de la difficulté à s'y rendre (espace), en utilisant des robots à base des systèmes électromécaniques et télé-opérés [5].

Une troisième orientation est apparue après la robotique industrielle et la robotique d'intervention en raison des progrès en miniaturisation, en microélectronique et en micromécanique, combinés avec les nouvelles capacités des systèmes de traitement de l'information et de communication [6]. Ceci a créé les conditions technologiques favorables au développement de robots mobiles autonomes ou semi-autonomes grâce aux capacités d'apprentissage et d'intelligence d'entités artificielles, matérielles ainsi que leur introduction dans des environnements en forte interaction avec l'homme pour réaliser des applications de service professionnel (agriculture, médical, nettoyage...) et de service personnel (jeux, tâches domestiques...) [7].

Dans ce chapitre, nous allons voir l'histoire de la robotique et ses différents types ainsi ses domaines d'application.

I.1 Définition :

I.1.1 Définition de la robotique :

La robotique est la science qui étudie les systèmes électromécaniques actionnés et contrôlés par le biais d'un ensemble de logiciels leur conférant une intelligence dite artificielle; aussi l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots.

La robotique est pluridisciplinaire implique nombreuses thématiques telles que la mécanique, l'électronique, l'automatique, l'informatique ou l'intelligence artificielle [8].

I.1.2 Etymologie du mot robot :

Le terme « robot » découle de la langues slaves (Tchèque) et signifie textuellement « esclave » ou bien (Robota) qui signifie travail forcé. Un robot est un système mécanique commandé par un ordinateur qui est destiné à effectuer une grande variété de tâches.

Par ailleurs, la robotique actuelle trouve des applications dans différents domaines : la robotique industrielle, domestique, médicale, militaire, scientifique et transport.

La robotique fait appel à plusieurs spécialités à savoir :

- Mécanique : conception, réalisation, modélisation des robots.
- Electronique : La mise en place de composants pour les robots en vue de les commander à distance en utilisant les concepts de télécommunications.
- Informatique: création de programmes destinés à la gestion du transfert d'informations entre les différents composants du robot.
- Automatique : commande, calibrage des capteurs, des effecteurs, identification des paramètres.

- Traitement du signal : analyse des informations enregistrées par les capteurs du robot.
- Mathématiques: modèle mathématiques pour la prise de décisions ou/et l'apprentissage, calcul de trajectoires, localisation, planification.
- Sciences cognitives: interactions homme-machine, machine-machine, prise de décisions.

La figure suivante représente un schéma synoptique d'un robot en interaction avec son environnement.

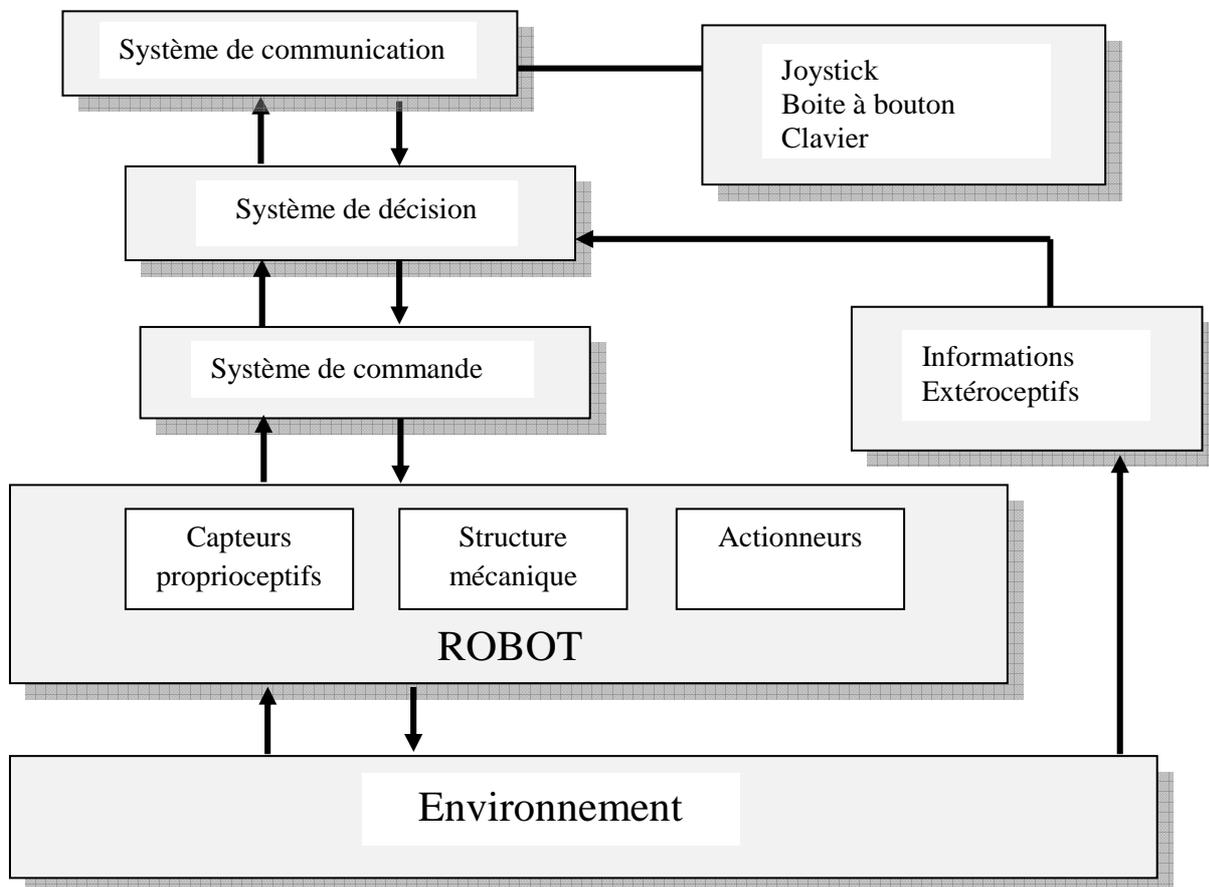


Figure I.1 : Architecture d'un Robot.

I.3 Historique de la robotique:

La notion de robot, ou d'automate, remonte à l'époque médiévale. Même s'il n'existait pas de terme pour décrire ce que nous appelons aujourd'hui des robots, les gens de cette époque ont tout de même imaginé des mécanismes capables d'exécuter des tâches humaines [9].

Le terme robot apparaît pour la première fois en 1921 par Karel Capek dans sa pièce théâtrale R.U.R (Rossums Universal Robots). Il provient du tchèque « robota » qui signifie corvée, travail obligatoire [10].

La Tortue cybernétique construite par Grey Walter dans les années 1950, est l'un des premiers robots mobiles autonomes, équipée de capteurs tactiles et lumineux lui permettant d'explorer son environnement.

Dans les années 60, avec l'apparition du transistor, les robots deviennent plus complexes mais qui vont réaliser des tâches similaires. Ainsi le robot « Beast » de l'université John Hopkins est capable de se déplacer au centre des couloirs en utilisant des capteurs ultrason.

Les premiers liens entre la recherche en intelligence artificielle et la robotique apparaissent à Stanford en 1969 avec Shakey. Ce robot utilise des télémètres à ultrason et une caméra. La Nasa développa aussi des rovers à partir de 1970 dans le cadre de programme Appolo.

Une étape importante est à signaler au début des années 1990 avec la mise en avant de la robotique réactive, représentée notamment par Rodney Brooks. Cette nouvelle approche qui a permis de passer de gros robots très lents à de petits robots, beaucoup plus réactifs et adaptés à leur environnement.

Enfin, pour les années 2000 les chercheurs ont pu développer des robots humanoïdes programmables et capable de courir, de communiquer en langage signe et de reproduire les expressions humaines.

Dates marquants la robotique [11]:

1947 : premier manipulateur électrique télé-opéré.

1954 : premier robot programmable.

1963 : utilisation de la vision pour commander un robot.

1973 : premier robot mobile à roues.

2011 : 5000 robots FANUC fabriqués chaque mois.

I.4 Types de robots :

On va aborder les trois catégories de robots les plus communes.

I.4.1 Robots industriels (manipulateurs) :

Les robots industriels sont les premiers à avoir été produits en grand nombre. Ils se trouvent plus particulièrement sur les chaînes de montage, et le plus souvent dans l'industrie automobile. Il existe des robots soudeurs, de démolition, de nettoyage, d'emballage ou de surveillance.



Figure I.2 : Robot manipulateur.

I.4.2 Robots humanoïdes :

C'est la catégorie la plus connue, en grande partie grâce à leur promotion faite par la Science fiction, elle regroupe tous les robots anthropomorphes, ceux dont la forme rappelle la Morphologie humaine. Ces robots ont généralement un torse, une tête, deux bras et deux jambes. Parfois, certains de ces robots ne représentent qu'une partie du corps. Lorsqu'un robot anthropomorphe imite non seulement l'apparence physique, mais aussi les comportements humains, on l'appelle un androïde.



Figure I.3 : Robot humanoïde.

I.4.3 Robots mobiles (explorateurs) :

De façon générale, cette catégorie englobe la sous-catégorie des robots mobiles à roues, les autres robots étant généralement appelés par leur nom de catégorie correspondant à leur fonctionnalité. Les robots mobiles à roue sont appelés en anglais UGV (UnmannedGroundVehicles), cette catégorie regroupe les robots à base actionnée par des roues ou par des chenilles. Ils sont généralement exploités pour faire de l'exploration, c'est la raison pour laquelle on les appelle des rovers (vagabonds en anglais).



Figure I.4 : Robot mobile.

I.4.3.1 Robots mobiles à déplacement articulé :

La moitié de la surface de la terre est inaccessible aux meilleurs véhicules à roues, Les recherches actuelles s'orientent vers l'étude de la marche à une ou plusieurs pattes. Le nombre de pattes détermine la complexité du système qui les commande. Les systèmes à une ou deux pattes comme l'homme ne fonctionnent qu'en mode dynamique, alors que quatre pattes ou plus autorisent une équilibre statique [12].

La reptation est le moyen utilisé pour progresser dans une galerie. La progression se fait au moyen de ventouses ou de griffes associées à un allongement des articulations [12]. On trouve différents types de déplacement : scolopendre, lombric et le péristaltique ou à soufflet.

I.4.3.2 Robots mobiles à roues :

Dans le cas des robots utilisant des roues pour se placer, le changement de direction est obtenu soit en faisant varier la vitesse des moteurs associés à chacune des roues latérales soit en faisant varier l'orientation des roues « véhicule automobile ».

a) Classification des types de roues :

La mobilité d'un robot mobile dépend du type de roues utilisé. Les différents types que l'on rencontre sont [13]:

- la roue fixe : cette roue n'autorise qu'un déplacement dans la direction de son plan médian, l'orientation est fixe.
- la roue centrée orientable : elle possède un axe d'orientation qui passe par le centre de la roue et un axe de rotation.
- la roue décentrée orientable ou roue folle : son axe d'orientation ne passe pas par le

Centre de la roue.

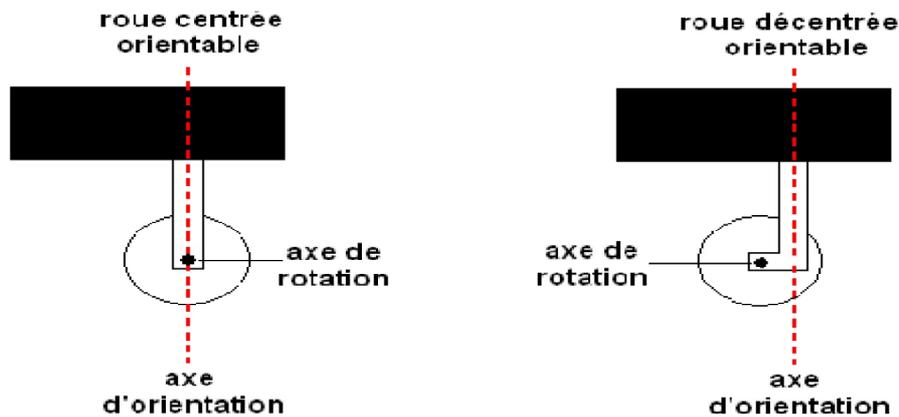


Figure I.5 : Roue centrée et roue décentrée.

En plus de ces roues classiques, d'autres roues ont été développées telle que les roues suédoises et tronco-sphériques [13] (Figure I.6) :

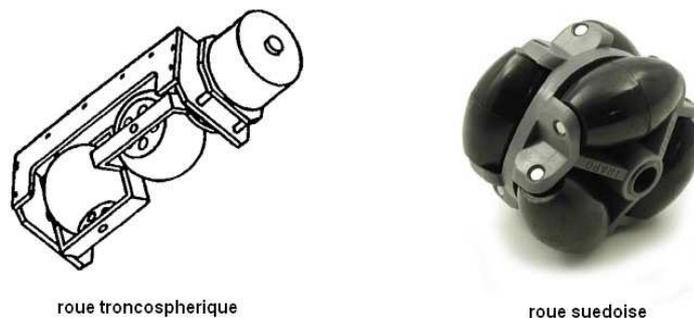


Figure I.6 : Roue troncosphérique et suédoise.

b) Principales structures cinématiques :

Tout système évoluant dans un plan 2D possède 3 degrés de liberté ; une translation selon l'axe \vec{x} , une translation selon l'axe \vec{y} et une rotation autour d'un axe \vec{z} .

Cependant une roue classique ne possède que 2 degrés de mobilité : elle ne peut faire qu'une translation (avancer ou reculer), ou une rotation sur elle-même. Elle ne peut pas séparer transversalement pour effectuer un mouvement de translation horizontal.

Cette contrainte empêche la plupart des véhicules « traditionnels » d'effectuer un déplacement instantané transversal (parallèlement à l'axe de rotation de la roue).

C'est une contrainte que l'on retrouve sur tous les robots mobiles de type voiture ou à roues différentielles. De tels véhicules, possédant un nombre de degrés de mobilité inférieur au nombre de degrés de liberté, sont dits non-holonomes, et cette contrainte touche principalement les robots mobiles à roues [15].

c) Modélisation cinématique des robots mobiles selon leurs différentes catégories :

Le modèle cinématique d'un robot mobile précise les possibilités d'évolution de toutes les coordonnées généralisées de ce système, il décrit complètement le comportement du robot mobile. Sa propriété évidente est sa linéarité par rapport aux vitesses.

Nous définissons $R = [O, \vec{x}, \vec{y}, \vec{z}]$ un repère fixe dans le plan d'évolution du robot, dont l'axe \vec{z} vertical et $R' = [O', \vec{x}', \vec{y}', \vec{z}']$ un repère mobile lié au robot. On choisit généralement pour O' le centre de l'axe des roues motrices (figure I.7) [16].

On appelle situation ou posture du robot le vecteur:

$$\varepsilon = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad (1.1)$$

Où x et y sont respectivement l'abscisse et l'ordonnée du point O' dans R et θ l'orientation du robot dans le plan (o, x, y) .

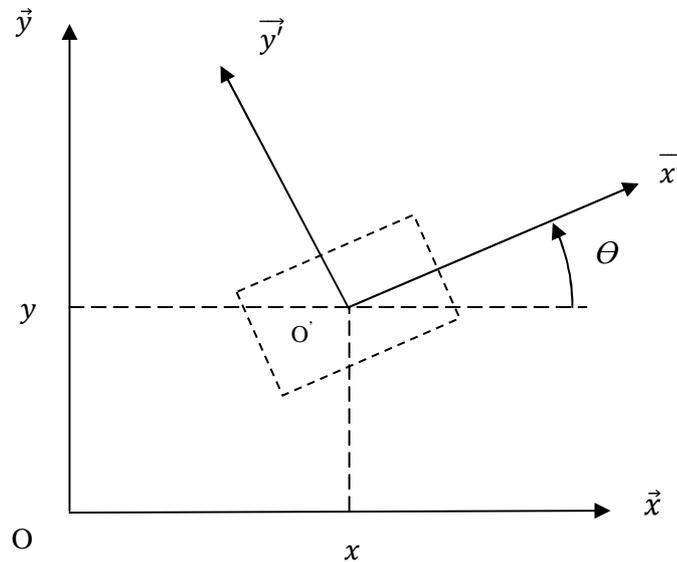


Figure I.7: Repérage d'un robot mobile.

Les modèles sont définies selon l'hypothèse de roulements sans glissement de la roue du véhicule et le sol toutes en réunissant les hypothèses suivantes :

- les roues de rayon r sont indéformables, et le contact entre la roue et le sol est ponctuel.
- la vitesse linéaire du point de contact d'une roue avec le sol est nulle.

d) Modèle cinématique du robot unicycle :

On désigne par unicycle un robot actionné par deux roues motrices indépendantes et une ou plusieurs roues folles afin d'assurer la stabilité. Son centre instantané de rotation (CIR) est situé sur l'axe reliant les deux roues motrices.

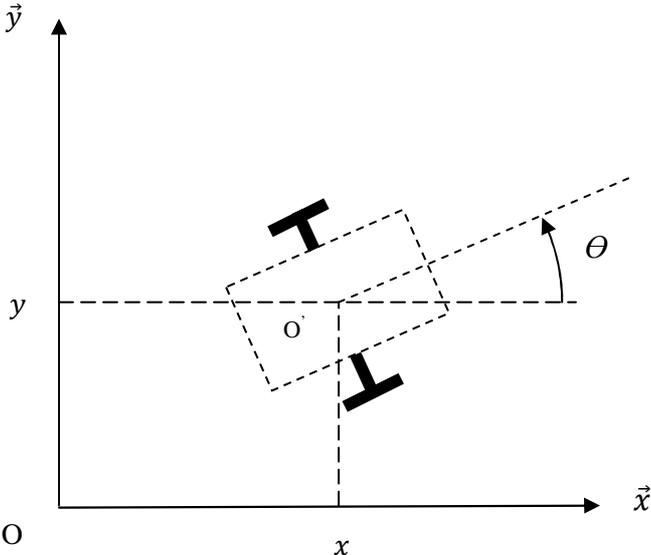


Figure I.8 : Robot mobile de type unicycle.

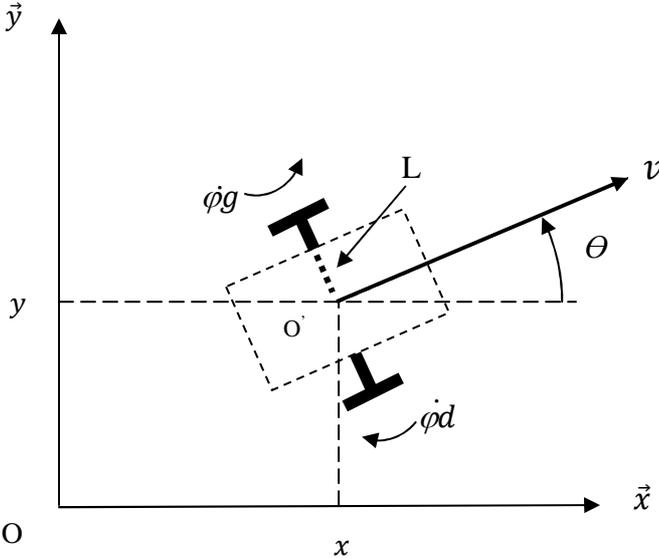


Figure I.9 :Modèle cinématique en posture du robot de type unicycle.

Le modèle cinématique est :

$$\begin{cases} \dot{x} = v \cos \theta. \\ \dot{y} = v \sin \theta. \\ \dot{\theta} = w. \end{cases} \quad (1.2)$$

Où :

$u=(v\omega)^T$: est le vecteur command cinématique.

v : est la vitesse longitudinale du robot.

w : La vitesse de rotation du robot autour du O' .

$\dot{\varphi}_d$: Vitesse angulaire roue droite.

$\dot{\varphi}_g$: Vitesse angulaire roue gauche.

L : Entre-axe des roues.

R : Rayon des roues.

e) Modèle cinématique du robot tricycle et voiture:

Ce robot est constitué de deux roues fixes de même axe et d'une roue centrée orientable placée sur l'axe longitudinal du robot (figure I.10). Le CIR du robot se situe à la rencontre des axes des roues fixes et de la roue orientable.

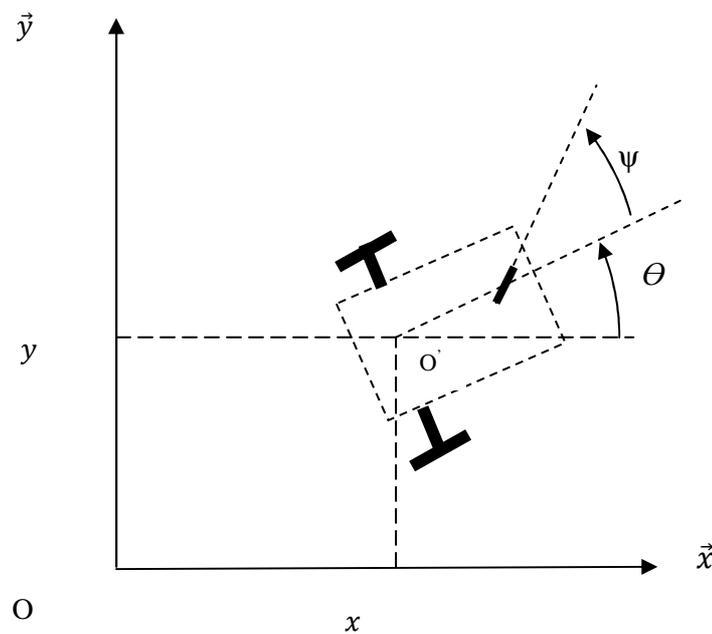


Figure I.10 : Robot mobile de type tricycle.

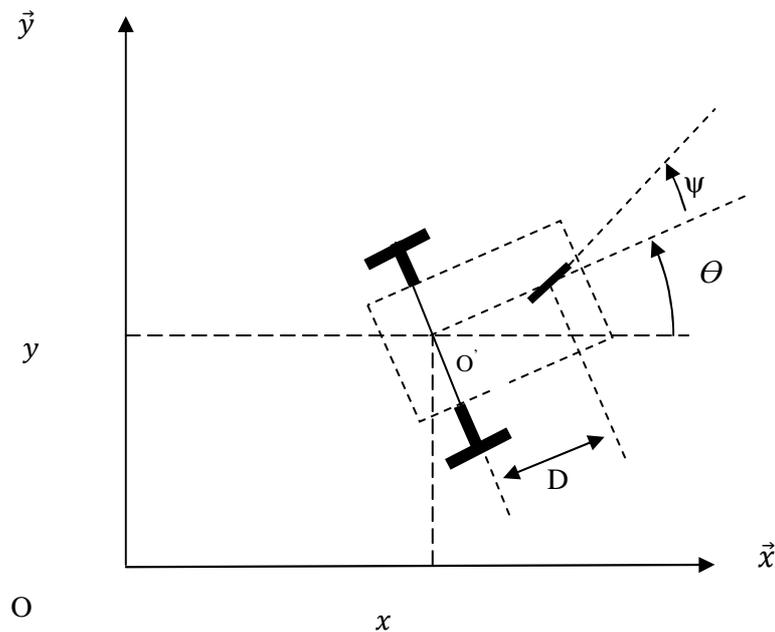


Figure I.11 : Modèle cinématique en posture du robot de type tricycle.

Le modèle cinématique est :

$$\begin{cases} \dot{x} = v \cos \theta. \\ \dot{y} = v \sin \theta. \\ \dot{\theta} = \frac{v}{D} \tan \psi. \\ \dot{\psi} = \eta. \end{cases} \quad (1.3)$$

Où

$u=(v\eta)^T$ est le vecteur de commande cinématique.

ψ : Représentant la rotation de la roue orientable.

η : Représentant la vitesse d'orientation imposée à la roue orientable.

D : Distance entre le centre de la roue orientable et le centre de l'axe à l'arrière. Le robot de type tricycle n'est que rarement utilisé, car il n'est pas très stable.

Le cas du robot de type voiture est semblable au tricycle. La différence se situe au niveau du train avant, qui comporte deux roues au lieu d'une alors il est plus stable puisqu'il possède un point d'appui supplémentaire.

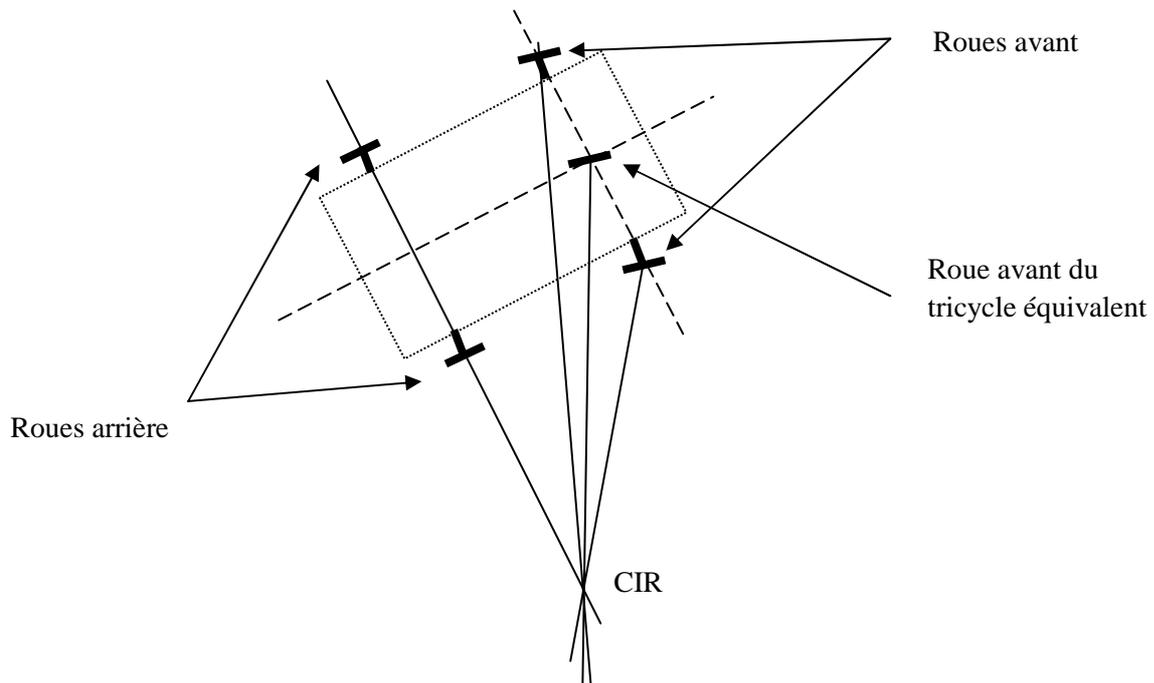


Figure I.12 : Un robot mobile de type voiture.

f) Modèle cinématique du robot omnidirectionnel :

Un robot mobile est dit omnidirectionnel s'il peut se déplacer librement dans toutes les directions. Il est en générale constitué de trois roues décentrées orientables placées en triangle équilatéral [17].

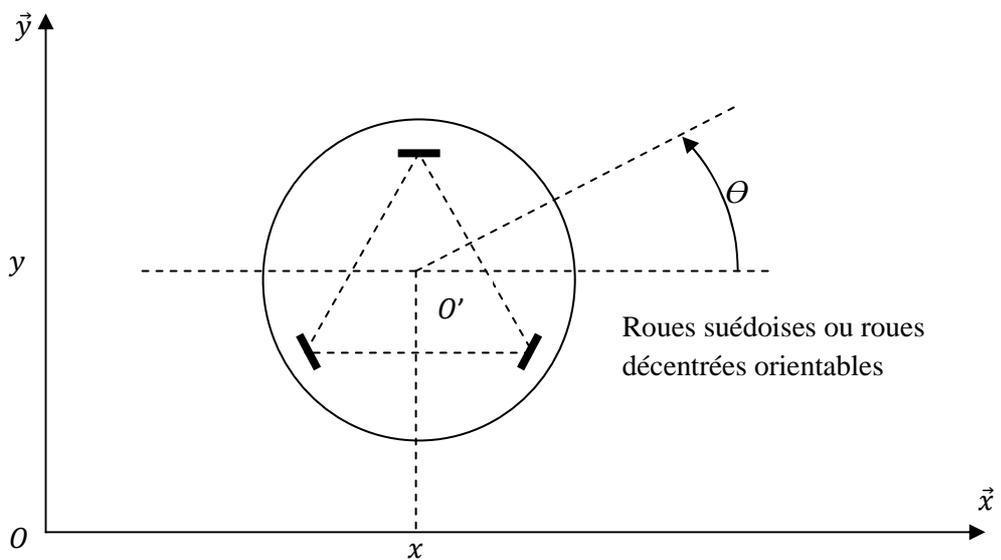


Figure I.13 : Représentation d'un robot mobile omnidirectionnel.

Le modèle cinématique est :

$$\begin{cases} \dot{x} = u_1. \\ \dot{y} = u_2. \\ \dot{\theta} = u_3. \end{cases} \quad (1.4)$$

Où $u = (u_1 u_2 u_3)^T$ représente le vecteur de commande.

Conclusion :

Dans ce chapitre, nous avons pu définir certaines notions fondamentales liées à la robotique en général.

Plus le temps avance plus la robotique évolue ; l'homme a tenté de concevoir des machines qui sont de plus en plus employés pour l'usage domestique. Ils peuvent se déplacer comme des hommes en marchant et même en courant. Mais rien de cela n'aurait été possible sans l'intelligence artificielle.

La robotique se développe actuellement en deux grandes branches telles que la robotique manufacturière utilisée comme un outil de la production. Ainsi la robotique non-manufacturière utilisée comme outil pour réaliser des tâches difficiles ou impossibles pour l'homme

Chapitre II

Introduction :

Un robot mobile est généralement conçu pour effectuer plusieurs tâches. Et pour qu'il soit utilisable, il doit comporter un système lui permettant un certain niveau d'autonomie dans la localisation et la navigation.

Pour schématiser ce processus, le robot doit être capable de répondre à triptyques de questions : « où suis-je ? », « où vais-je ? » et « comment y aller ? ». La première interrogation soulève le problème de la localisation, quant aux deux autres, elles sont liées à la planification de trajectoire et à la navigation proprement dite. La bonne exécution des deux dernières tâches est fortement liée à la première.

Cette étude se restreint aux robots mobiles navigants sur un plan (2D), ce qui englobe une très large partie des systèmes existants. Localiser le robot revient alors à déterminer trois paramètres : Deux coordonnées cartésiennes pour la position et un angle pour l'orientation. De façon plus formelle, la tâche de localisation consiste à calculer la transformation de passage d'un repère lié au robot à un repère lié à l'environnement [18].

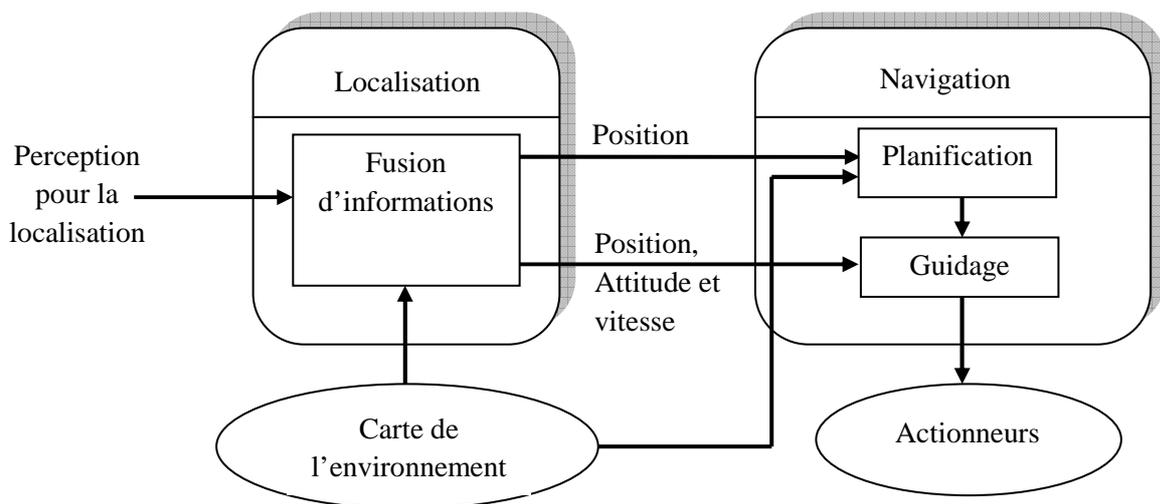


Figure II.1 : Schéma fonctionnelle d'un véhicule automatisé.

II.1 Perception :

La notion de perception en robotique mobile est relative à la capacité du robot à recueillir, traiter et mettre en forme des informations qu'elles lui sont utiles pour agir et réagir dans son environnement. Et pour tout cela il faut que le robot dispose de nombreux capteurs. Le choix de ces derniers dépend bien évidemment de l'application envisagée.

La perception est nécessaire pour la sécurité du robot, la modélisation de l'environnement et l'évitement et le contournement d'obstacles.

Les moyens utilisés pour la perception de l'environnement sont nombreux et variés, nous citons :

- Les systèmes de vision.
- Les télémètres lasers et ultrasonores.
- Les capteurs optiques et infrarouges.
- Les capteurs tactiles.

II.2 Localisation :

Un robot mobile doit connaître ses coordonnées de position soit en identifiant des repères artificiels, ou en mesurant les distances parcourues et les directions empruntées depuis sa position initiale pour être autonome vis-à-vis de l'espace.

Ces robots nécessitent un environnement structuré avec des systèmes de guidage. Le filoguidage ou guidage inductif, consiste à encastrer dans le sol un fil parcouru par un courant alternatif. Une force électromotrice est induite dans deux (02) bobines disposées sous le robot d'une part, et d'autre part, du fil inductif. Le guidage consiste à minimiser la différence entre les deux forces électromotrices induites afin de maintenir le robot sur la trajectoire matérialisée par le fil.

Dans le cas du guidage optique, la position relative du robot est obtenue par la réflexion d'une onde lumineuse sur une piste matérialisée par des bandes adhésives réfléchissantes disposées sur le sol, le signal détecté est maximal si le faisceau lumineux se réfléchit sur la bande, et il est minimal quand le faisceau se réfléchit en dehors de la bande. Des marquages particuliers sont disposés en certains endroits stratégiques pour informer le

robot mobile sur l'évolution de la trajectoire, sur sa position ou sur le point d'arrêt. Ce système ne fonctionne correctement que si le contraste entre le sol et le marquage au sol est suffisant [19].

La lourdeur de ces systèmes de guidage ont conduit les chercheurs à étudier d'autres systèmes plus souples et plus performants. C'est ainsi que les méthodes de localisation se regroupent en trois catégories suivantes [20]:

- La localisation relative ou a l'estime qui est en rapport avec des informations issues de capteurs proprioceptifs.
- La localisation absolue qui est en rapport avec des informations issues de capteurs extéroceptifs.
- La localisation dite hybride.

II.2.1 Localisation relative :

La localisation relative consiste à déterminer la position, l'orientation et la vitesse du robot mobile, elle peut être décomposé en deux catégories : la localisation par odométrie et la localisation inertielle.

II.2.1.1 Localisation par odométrie :

La localisation par odométrie permet de déterminer la position et l'orientation d'un système mobile par intégration de ses déplacements élémentaires, délivrés sous forme d'impulsions, toutes les fractions de tour de roue par des codeurs couplés aux roues.

Les avantages de cette technique résident dans sa simplicité de mise en œuvre et dans son faible cout. Ces caractéristiques font la technique de localisation la plus couramment utilisée en robotique mobile.

En outre, l'utilisation d'odomètres permet d'obtenir une estimation de la position et l'orientation à une cadence relativement élevée, avec une précision qui reste bonne sur de faibles distances. Cependant, dès que les distances à parcourir augmentent et que la trajectoire comporte de nombreux changements de direction, l'imprécision sur la position et l'orientation

du système mobile augmente considérablement ; imprécision due à l'erreur cumulative générée lors de l'intégration des déplacements élémentaires.

Les erreurs de cette technique peuvent être de différentes natures (systématiques ou non-systématiques), selon des paramètres liés au robot mobile (diamètres des roues), ou à l'environnement ainsi que l'état de terrain.

a) Modélisation des erreurs odométriques :

Le principal défaut de cette méthode est une dérive de la position proportionnelle à la distance parcourue par le robot mobile, principalement une grande erreur sur la position est causée par l'accumulation d'erreurs lors des intégrations successives, qui peuvent être classées en deux catégories [14] :

- **Erreurs systématiques :**

Ces erreurs s'ajoutent à chaque itération de la localisation du robot mobile et sont de deux natures différentes :

- Les erreurs sur les paramètres mécaniques du robot mobile (les roues de diamètre différents, erreurs sur la longueur entre axe des roues, non alignement des roues, etc....)
- Les erreurs de mesures (résolution des codeurs, fréquence d'échantillonnage inexacte, etc....)

Les erreurs systématiques du premier type sont en générale compensées en utilisant des méthodes de calibrage, alors que celles du second type ne peuvent pas être corrigées et doivent ainsi être prises en considération dans le modèle cinématique du robot mobile.

- **Erreurs non-systématiques :**

Ces erreurs sont directement dues aux contacts robot / terrain. Elles présentent un caractère aléatoire et dépendent en générale des caractéristiques du sol : Surface de roulement irrégulière (trous et bosses), présence d'objets sur la route ou encore glissement du robot mobile (virages, terrain glissant).

Du fait de l'impossibilité de les prédire, les erreurs non-systématiques sont quasiment impossibles à compenser sans recours à des méthodes de localisation absolu. Selon l'environnement dans lequel évolue le robot mobile, l'une ou l'autre de ces sources d'erreurs deviennent prépondérante devant l'autre. Par exemple, dans le cadre d'applications en environnement extérieur, les erreurs non-systématiques jouent un rôle primordial. En revanche dans un environnement d'intérieur, ce qui est dans notre cas, les erreurs non-systématiques étant quasi-inexistantes, ce sont les erreurs systématiques qui constituent la cause principale des dérives de la position.

II.2.1.2 Localisation inertiel :

Cette technique de localisation se base sur l'utilisation des systèmes inertiels de navigation (INS). D'après les lois de la mécanique formulées par *Isaac Newton*, la localisation du véhicule est déduite de la double intégration des composantes accélérométriques α du robot mobile [21]:

$$\text{Vitesse} \quad v = \int a \, dt \quad (1.5)$$

$$\text{Position} \quad p = \int v \, dt \quad (1.6)$$

Pour réaliser le positionnement dans un système, il est alors nécessaire de connaître l'orientation spatiale des composantes accélérométriques. Ainsi les systèmes INS sont composés de trois accéléromètres orientés à l'aide de gyroscopes.

$$\text{Orientation} \quad \theta = \int w \, dt \quad (1.7)$$

a) Systèmes inertiels :

Les différents systèmes inertiels peuvent être classés en deux catégories : Les systèmes à plates-formes stabilisées, et les systèmes à composants liés. Ces systèmes peuvent être composés de différents types de capteurs inertiels.

- **Systèmes inertiels à plates-formes stabilisées :**

Ce sont les premiers systèmes de navigation inertielle, le positionnement est réalisé par une double intégration des composantes d'accélération du robot mobile. Ce genre de système est très performant et offre l'avantage de protéger les capteurs de la plupart des vibrations encourues par le robot, prolongeant ainsi la durée de fonctionnement du système.

- **Systèmes inertiels à composants liés :**

Sont les fruits du progrès technologique et en particulier de l'informatique. Dans ces systèmes les accéléromètres et les gyromètres sont directement solidaires au corps du robot mobile. Les accélérations mesurées directement dans le repère lié au robot sont converties dans le repère référence en fonction de l'attitude mesurée par les gyromètres. Ces systèmes sont de structure très simple avec un faible encombrement spatial et résistant aux chocs.

Ils sont donc particulièrement appropriés pour le domaine de la robotique mobile. Néanmoins, cette configuration présente l'inconvénient de soumettre les capteurs aux vibrations du véhicule, leurs mesures sont alors bruitées et nécessitent d'importants prétraitements dégradant ainsi la précision par rapport aux systèmes à plates-formes stabilisées.

b) Capteurs inertiels :

Les capteurs inertiels touchent un large éventail des sciences contemporaines comme la mécanique, l'électronique ou encore la physique atomique, parmi ceux utilisés ces dernières décennies sont [22]:

- Le RLG : Créé dans les années 60 et apparut dans les systèmes *strapdown* dans les années 80, est un excellent capteur inertiel. Il a une très bonne stabilité, une faible sensibilité à l'accélération. Ce capteur aussi est très onéreux et n'est pas utilisé que dans les applications militaires et l'aviation.
- Le FOG : Inventé dans les années 80, il devait être un simple alternatif bas cout aux RLG. Actuellement, il obtient des performances équivalentes voire supérieures aux

RLG, avec en particulier une insensibilité envers les vibrations. Il est désormais très compétitif dans les applications militaires et les applications commerciales.

- Les gyromètres MEMS : Sont présentés comme les remplaçants des deux précédents capteurs, ils bénéficient d'une réduction de taille et de prix très important. Ces avantages impliquent une baisse des performances tels que l'augmentation de bruit de mesure, une faible sensibilité.

c) Erreurs des capteurs inertiels :

Nous énumérons deux sources d'erreurs qui peuvent influencer la mesure d'un capteur inertiel à savoir : un biais qui est le terme le plus pondérant dans la mesure de la vitesse, de la position et de l'orientation du robot mobile. En effet il cause une dérive qui augmente linéairement en fonction de la vitesse et l'orientation et quadratiquement pour la position. Aussi les bruits de mesure qui sont inévitables comme dans tous les capteurs, il a pour origine des composants électroniques.

II.2.2 Localisation absolue :

La localisation absolue est une technique qui permet à un système mobile de se repérer directement dans son milieu d'évolution, que ce soit en environnement extérieur (mer, terre), ou en environnement intérieur (immeuble, atelier) [22]. Elle est basée sur l'utilisation de capteurs extéroceptifs et nécessite toujours une représentation de l'environnement. Le système mobile possède donc une banque de données regroupant les éléments caractéristiques de son milieu d'évolution (balises par exemple), et à partir de ces éléments, le robot mobile va déduire sa position dans son environnement par des techniques qui se distinguent par la nature des repères utilisés, ou par la méthode de calcul.

Selon la nature des repères utilisés, les approches les plus connues sont la localisation grâce aux éléments suivants: Compas magnétique; repères actifs ou passifs; un modèle de présentation.

Selon les techniques de calcul utilisées, on distingue entre autres, les méthodes basées sur la trilatération (ou multilatération), les méthodes de triangulation.

II.2.2.1 Types de repères utilisés :

a) Localisation par compas magnétique :

Un compas magnétique permet de déterminer une orientation absolue en mesurant la composante horizontale du champ magnétique terrestre, engendré par les mouvements du noyau métallique liquide des couches profondes de la Terre. Le repère par rapport auquel on se positionne est dans ce cas lié à la terre [18].

L'inconvénient principal ; est le fait que ce champ soit altéré au voisinage de lignes à haute tension.

b) Localisation par balises actives :

Les balises actives sont ajoutées au milieu d'évolution du système mobile, dont les positions sont connues. Elles sont généralement interactives avec le système de perception du système mobile, puisque c'est les balises actives qui émettent l'onde captée par l'équipement de mesure. Les deux principaux types d'organes de transmission utilisés sont les émetteurs de sources lumineuses (ex : infrarouge) et les antennes émettrices hyperfréquences [18].

L'utilisation de ces balises a le gros avantage d'être précise, robuste et surtout de satisfaire la contrainte temps réel. En revanche, Leur inconvénient réside essentiellement dans leur manque de souplesse ainsi que leur lourdeur d'utilisation.

Un des systèmes les plus connus en utilisant ce type de balise, est le système GPS. C'est un système de balises placées sur des satellites en orbite terrestre, utilisant les ondes radio émises partout à la surface du globe. Le calcul de la position et l'orientation est basée sur la trilatération ou la triangulation.

c) Localisation grâce à des repères passifs :

On distingue deux types de repères passifs, les *balises passives* et les *amers*. Contrairement aux balises actives, les balises passives n'échangent pas de signaux avec le système de perception, elles se contentent de réfléchir un signal provenant de l'équipement de mesure. Il peut agir par exemple de miroirs [18].

Ces balises doivent avoir une position connue dans l'environnement et peuvent également comporter des informations plus évoluées (codes-barres) qui facilitent le problème de la mise en correspondance entre elles.

Deuxièmement, On appelle amers des éléments distincts de l'environnement que le robot peut reconnaître grâce à ses capteurs extéroceptifs. Ces amers peuvent être des formes géométrique (rectangles, lignes, cercles) [18]. Cependant leur détection et leur identification peuvent être plus difficiles ce qui engendre les risques d'ambiguïté ou de fausses détections plus élevées.

L'extraction d'une seule image utilisons un capteur de vision, peut représenter plusieurs repères, il est alors possible de trouver la position et l'orientation du robot mobile grâce à un contraste suffisant des balises par rapport au fond. Aussi, ils peuvent servir à corriger la position du robot mobile lors de la navigation.

En environnement intérieur, ces repères peuvent être des réflecteurs placés sur le mur. La localisation du robot mobile est alors réalisée par une triangulation pour laquelle les angles séparant le robot des balises sont mesurés à partir d'un capteur à ultrasons tournant et des réflecteurs acoustiques.

d) Localisation basée sur un modèle :

Dans cette technique, un plan ou modèle de l'espace de travail du robot, est stocké en mémoire. A un instant donné, le robot perçoit et construit grâce à ces capteurs un plan local de son environnement, le système effectue alors une mise en correspondance entre le plan local et la partie correspondante dans le modèle global. Si une telle mise en correspondance est trouvée alors la position et l'orientation du robot peuvent être calculées.

Il existe plusieurs façons de représenter l'environnement, citons par exemple : les cartes métriques, les cartes topologiques, les grilles d'occupations, les cartes hybrides.

II.2.2.2 Méthodes de calcul de position et d'orientation :

a) Trilatération (multilatération) :

La Multilatération est une méthode relativement simple et intuitive, en utilisant plus que trois points de référence. La position d'un nœud est calculée en connaissant les positions d'un certaines références et les distances estimées de ce nœud aux différentes références.

La localisation se fait grâce au principe de différence de temps d'arrivée de ces signaux. La figure II.2 montre un exemple de calcul de la position du robot selon la connaissance des distances respectives aux balises B1, B2, B3.

Pour chaque paire de balises sont établies des surfaces hyperboles, dont la différence en distance à ces balises est constante, la position du robot est à l'intersection de ces surfaces.

La précision d'un système de multilatération dépend de la géométrie du système formé par le robot et les balises, ainsi que de la précision de la mesure du temps d'arrivée des signaux [18].

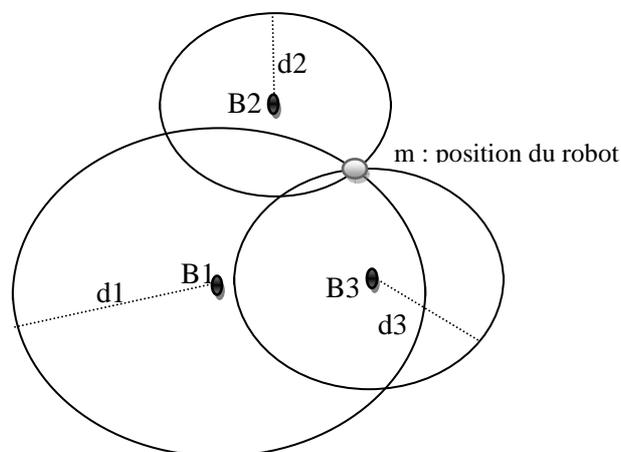


Figure II.2 : La trilatération.

b) Triangulation :

Dans ce cas les mesures utilisées sont goniométriques. Les mesures des angles $\lambda_1, \lambda_2, \lambda_3, \dots$ avec lesquels sont vus les repères peuvent être effectuées grâce à un capteur (récepteur ou émetteurs) rotatif embarqué sur le robot, ou encore une caméra panoramique [18].

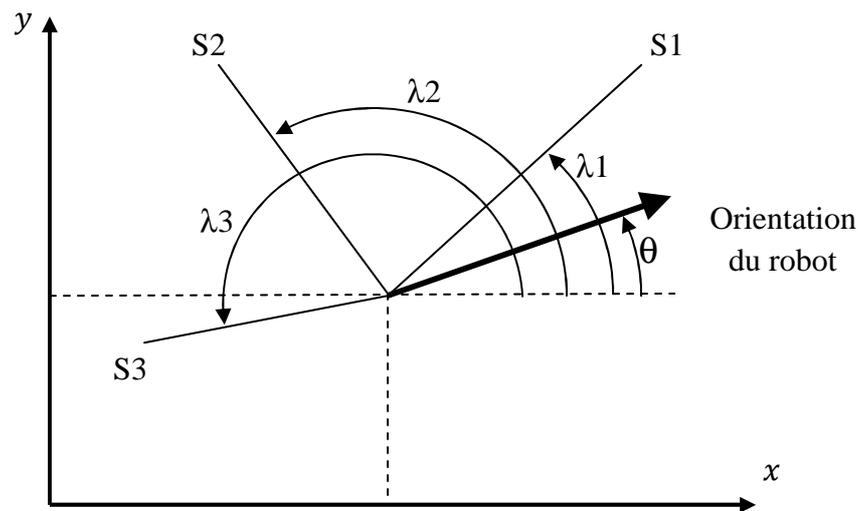


Figure II.3 : Principe de la triangulation

II.2.3 Localisation hybride :

Les différentes méthodes de localisation présentent divers inconvénients et avantages. Les avantages principaux de la localisation relative, sont l'utilisation des capteurs proprioceptifs propres au système mobile, une cadence d'acquisition des informations élevée qui permet de calculer précisément le déplacement du robot à court terme.

Par contre à long terme, cette méthode diverge en raison de l'accumulation des erreurs lors des intégrations successives des déplacements élémentaires. En revanche la localisation absolue s'appuie sur des données extéroceptives, et nécessitent une connaissance à priori de l'environnement. Ce type de localisation n'est pas affecté par ce problème, mais le résultat de la localisation est assujéti à la visibilité des sources d'information.

D'après ce constat, une seule méthode de localisation ne peut pas permettre une localisation fiable d'un robot. Alors la plupart des systèmes de localisations sont réalisés par des systèmes hybrides, combinant plusieurs sources d'informations, où on peut distinguer deux cas [22]:

Soit ces informations sont de même nature et sont :

- Redondantes : Auxquelles la combinaison des informations tente d'accroître la précision et le degré de confiance du système de localisation, en fonction de plusieurs observations distinctes.

- Complémentaires : Dans ce cas de fusion des informations, il permet en général de fournir une information plus complète sur le comportement du robot mobile dans son environnement, qui ne peut pas être obtenues en utilisant ces capteurs séparément.

Soit ces informations ne sont pas de mêmes natures (informations proprioceptives et extéroceptives), ou le système de localisation absolu est en général chargé de corriger régulièrement l'estimation du système de localisation relatif, afin de pallier les problèmes de dérives rencontrés dans la localisation relative [22].

Pour mettre en œuvre de tels systèmes de localisation, de nombreuses techniques de fusion de données sont disponibles. Ces techniques peuvent être séparées en quatre grandes classes : l'approche analytique telles que le filtre de Kalman et ses extensions qui sont les plus utilisées dans les approches hybrides de localisation en robotique mobiles, les approches numériques telles que le filtre particulaire et ses extensions, les approches ensemblistes que ce soit les approches à erreur bornée ou les approches par analyse d'intervalles, et enfin les approches basées sur la théorie des croyances ou la théorie des possibilités.

II.3 Modélisation de l'environnement d'intérieur pour la navigation :

Dans la robotique mobile, une multitude de représentations de l'environnement a été employée par les chercheurs, d'ailleurs plusieurs cartes ont été développées.

Il existe un consensus général sur le fait que ces différents types ont des avantages et des limitations et qu'ils sont plus ou moins adaptés selon la mission à accomplir.

Par exemple les cartes métriques, sont difficiles à élaborer et à mettre à jour, à cause de l'incohérence entre le mouvement du robot mobile et la perception. En revanche les cartes topologiques sont mieux adaptées pour représenter les grands environnements, ce qui permet de faire une localisation globale.

Lors de la construction des cartes topologiques, la distinction des différentes composantes est difficile sans l'utilisation d'informations métriques, ce qui rend les représentations complémentaires favorables, pour les besoins de navigation du robot mobile. C'est pourquoi les approches hybrides, qui combinent différents types de modèles élémentaires, se généralisent:

III.3.1 Cartes métriques :

L'approche métrique permet une génération d'une carte géométrique plus ou moins détaillée de l'environnement à partir des données perçues. Les informations de longueur, distance, position sont précises dans les cartes métriques, qui sont souvent plus faciles à être interpréter car elles offrent une relation bien définie avec le monde réel.

L'élaboration des cartes métriques cohérentes, nécessite de gérer des contraintes géométriques particulières, c'est une représentation de base utilisant l'ensemble des primitives ; qui sont construites par fusions successives de données élaborées à partir d'autres données métriques non traitées, soit 2D acquises par un balayage horizontal d'un télémètre laser, ou des primitives 3D comme les segments regroupés en façades. Ce qui permet de décrire des amers dédiés à la localisation et de représenter aussi l'espace navigable [8].

Plusieurs types d'amers géométriques peuvent être distingués, comme les points, les coins, les segments, les polygones,...etc.

- Points : dans les cartes 2D, ils représentent notamment les objets verticaux tels que des poteaux ou des troncs d'arbres en environnement extérieur, et les coins de murs en environnement intérieur.
- Coins : par rapport au point, ils incluent une information d'orientation, et éventuellement un angle relatif entre les segments adjacents à ce coin.
- Segments : ils correspondent généralement aux frontières d'obstacles.
- Polygones : ils sont de forme et de taille quelconques, ils correspondent aux frontières d'obstacles dans le plan horizontal.

La représentation métrique a pour avantage d'être robuste et relativement efficace. En revanche, elle possède le gros désavantage d'être lourde en temps de calcul et limité en précision [23].



Figure II.4 : Exemple de représentation d'un environnement avec carte métrique.

III.3.2 Cartes topologiques :

Elles représentent l'espace navigable du robot sous la forme d'un graphe, qui contient toutes les informations sur la connexité. Autrement dit, c'est une représentation qui décrit les relations entre les éléments de l'environnement sans utiliser une référence métrique.

Les environnements structurés sont les plus facilement modélisés d'une façon topologique, car ils induisent une structure topologique avec moins d'ambiguïtés, aussi l'exploitation de données multi sensorielles est très importante. Mais les systèmes de vision sont généralement les plus privilégiés pour construire des graphes de zones significatives au sens de la navigation. Pour se localiser, le robot a seulement besoin de savoir sur quel connecteur il se déplace en quittant une place. Cependant il doit être capable de distinguer facilement les places.

La carte topologique présente quelques avantages, où l'avantage majeur est d'éviter les approximations géométriques par primitives, tout en rendant compte de la structure de l'environnement. Ceci peut permettre une localisation qualitative dans un lieu effectué à partir d'informations odométriques, et une connaissance de la structure et de la nature des lieux qui la constituent [23].

C'est de plus une représentation très économe de l'espace mémoire. La concision et la facilité de construction d'une carte, puisque elle ne nécessite pas une connaissance très pointue de l'environnement. Par contre, l'absence d'informations géométriques dans une carte topologique pure peut être inadéquate dans certain cas où une localisation précise est nécessaire.

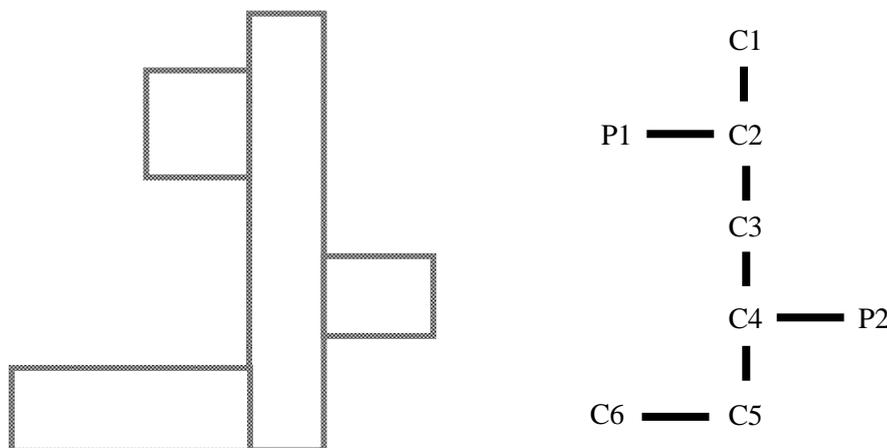


Figure II.5 : Exemple de représentation topologique d'un environnement constitué de couloir (c) et de pièces (P).

III.3.3 Grilles d'occupations :

Les grilles d'occupations constituent la représentation surfacique la plus simple et la plus populaire. Ce sont des représentations de l'environnement qui décomposent l'espace dans lequel un robot mobile évolue, en un ensemble de cellules distincts soit carrée ou rectangulaire de même taille. La probabilité d'occupation de chaque cellule est une valeur estimée à partir des mesures fournies par les capteurs, indiquant si l'espace correspondant est plutôt libre ou occupé [23].

La grille d'occupation, représente un modèle capable de faire la mise à jour de l'environnement à une fréquence élevée, et permettant de réviser facilement les probabilités d'occupation, donc de suivre l'évolution de l'environnement autour du robot, ce qui est indispensable pour une meilleure réactivité.

Elle est capable de modéliser des environnements de forme quelconque, et ne cherche pas à approximer les données par des primitives qui peuvent être inadéquates. L'avantage principal des grilles d'occupation est leur capacité à représenter l'espace de manière très dense. Elle est en général préférée pour les applications qui reposent sur la détection de l'espace navigable ou pour l'évitement d'obstacles. Elle est facile à interpréter par l'homme.

Enfin, les grilles d'occupations permettent une intégration aisée de divers types de capteurs. En revanche, l'inconvénient majeur de cette représentation est d'être relativement lourd du point de vue mémoire, car elle exige un espace de stockage important pour les grandes résolutions [16].

III.3.4 Cartes hybrides :

L'idée de la carte hybride est d'utiliser deux ou plusieurs types de représentations, ce qui permet de profiter des avantages de chaque représentation, et pourrait aider à surmonter les limites.

Par exemple, plusieurs chercheurs ont proposé des méthodes hybrides métriques-topologiques, dans l'intention de combiner la précision des cartes métriques avec l'extensibilité des cartes topologiques, car la topologie permet de subdiviser l'environnement de manière naturelle, elle induit un espace de stockage et des temps de calcul réduits et assure une connexité ainsi qu'une cohérence à grande échelle. En contrepartie, les cartes métriques induisent une précision locale élevée [8].

II.4 Navigation :

Un robot mobile autonome doit pouvoir changer de voie, suivre la route, éviter les obstacles fixes et mobiles.

La navigation est divisée en deux modules :

- Un module de planification de trajectoire qui calcule un itinéraire local sous la forme de points de passage.
- Un module de guidage qui calcule les consignes de vitesse et de direction et qui permet au robot mobile de suivre une trajectoire, en interpolant les points calculés par le module de planification.

De ce fait, le problème de la navigation d'un robot mobile consiste à trouver un mouvement dans l'espace des configurations sans collision, ce mouvement mène le robot d'une configuration initiale à une configuration finale utilisant des techniques pour l'évitement d'obstacles.

La planification de trajectoire se fait généralement en deux étapes : Une planification globale et une planification locale. Au stade de la planification globale, l'itinéraire global est déterminé à partir des informations d'une carte numérique et du système de localisation. La planification locale est faite à partir de l'itinéraire global, et des informations de l'environnement proche du robot mobile.

En effet, à partir des données de la perception de l'environnement proche, et de la localisation du robot mobile ; une carte de l'environnement est construite avec les informations relatives aux obstacles. Ensuite plusieurs trajectoires locales possibles sont calculées dans l'espace navigable.

La trajectoire de référence est choisie en se basant sur plusieurs critères, de manière à assurer une navigation tout en évitant les obstacles.

II.4.1 Principales méthodes réactives de planification de trajectoires des robots mobiles :

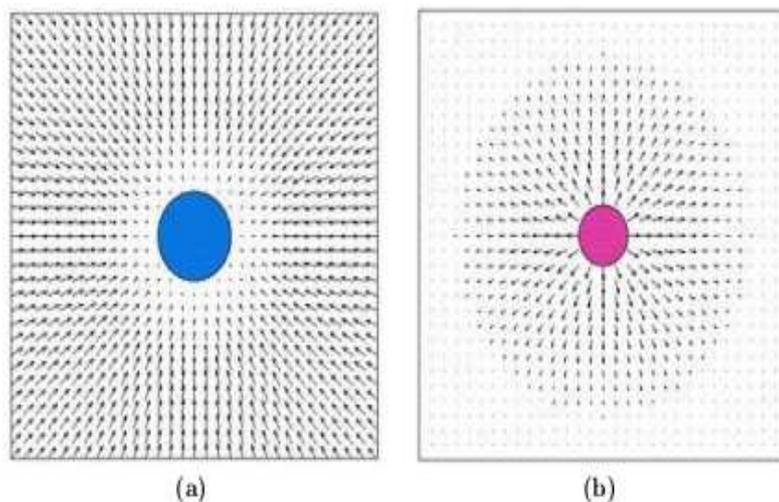
Le problème de planification de trajectoire consiste à trouver la meilleure trajectoire pour aller d'un point de départ à un point cible. Différentes approches de planification de trajectoire ont été développées pour la navigation des robots mobiles autonomes.

Parmi les approches classiques, nous pouvons citer la méthode des champs de potentiels, les méthodes basées sur la connectivité de l'espace navigable, les méthodes des tentacules, les méthodes des champs de braquage constant, les méthodes basées sur un modèle de prédiction, et fenêtre dynamique.

II.4.1.1 Méthode des champs de potentiels :

Initialement, cette méthode a été proposée par *O. Khatib* en 1986. Elle consiste à définir une fonction qui affecte un grand potentiel aux obstacles (champs répulsifs) et un faible potentiel aux points objectifs (champs attractifs) de la trajectoire à naviguer. Le problème de navigation est ainsi réduit à un problème d'optimisation trouvant la commande qui amène le robot vers un minimum global de la fonction objective [8].

Ainsi, la trajectoire calculée mène vers l'objectif, tout en évitant les obstacles. Cette fonction à minimiser peut être définie par rapport à l'objectif de la navigation et aux obstacles.



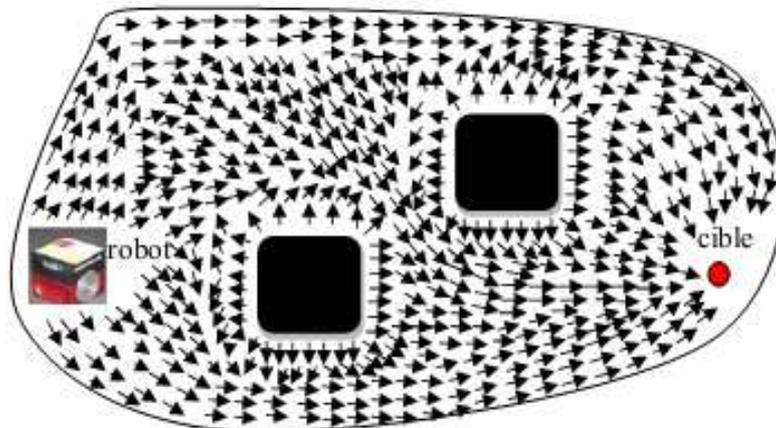


Figure II.6 : Principe de la méthode des champs des potentiels : (a) champs attractifs, (b) champs répulsifs, Illustration d'évitement d'obstacle.

Cette méthode représente l'avantage de ne pas nécessiter la connaissance précise des contours des obstacles, elle se base sur la probabilité de l'existence d'un obstacle.

Le principale inconvénient de cette méthode est la charge importante de calcul, ce qui peut être contraignant lors de son application en temps réel pour la navigation, aussi elle ne peut pas donner de bons résultats dans les environnements très contraints (grande vitesse, changements soudains) ce qui provoque des possibilités de collision.

II.4.1.2 Méthodes basées sur la connectivité de l'espace navigable:

Les méthodes de planification nommées "roadmaps" sont basées sur l'utilisation de la théorie des graphes. Le principe est de construire un graphe reliant la configuration initiale à la configuration finale tout en évitant les obstacles. Ces techniques sont parmi les plus utilisées et les plus développées ces dernières années. Nous citons [8]:

- **Diagramme de Voronoï :** Sont des structures de données fondamentales largement étudiés en géométrie, dans le cadre de la planification de trajectoire, le diagramme de Voronoï est construit en traçant des lignes d'égale distance aux obstacles. Une fois que le graphe est construit, l'objectif est de déterminer le chemin

reliant la configuration initiale du robot à la configuration finale dans ce graphe en minimisant le critère de distance.

- **Graphe de visibilité :** Est un graphe où pour chaque sommet d'objet, on trace une ligne droite entre celui-ci et les sommets des autres objets que l'on peut voir. Le graphe de visibilité est utilisé dans plusieurs domaines notamment le calcul de plus court chemin, et l'évitement d'obstacles. Depuis la configuration initiale, on considère les droites reliant le point de départ aux sommets visibles des obstacles, puis l'opération est réitérée entre les sommets des obstacles et les sommets visibles des suivants jusqu'au point d'arrivée. Une fois que le graphe de visibilité est construit, le plus court chemin reliant la position de départ et celle d'arriver dans l'espace morcelé est calculé.

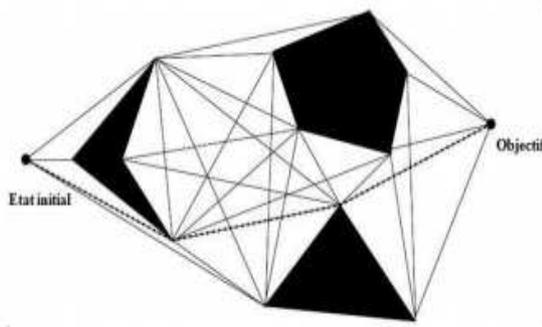


Figure II.7 : Graphes de visibilité.

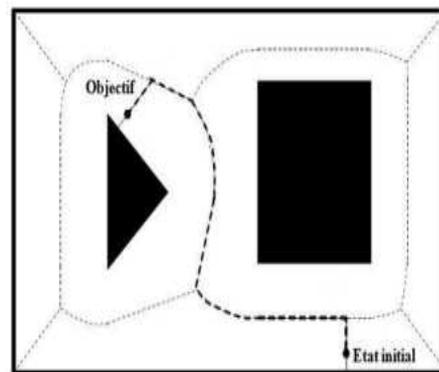


Figure II.8 : Diagramme de Voronoï.

II.4.1.3 Méthodes des tentacules :

Le principe de cette méthode consiste à utiliser un ensemble d'antennes virtuelles appelées « tentacules » dans une grille d'occupation égo-centrée liée au robot mobile, et qui représente l'environnement proche de ce dernier avec la position des obstacles.

Une fois les trajectoires virtuelles sont créés, une évaluation de ces dernières est effectuée afin de choisir la meilleure. Le meilleur tentacule ou trajectoire de références, sera exécuté par le robot à chaque pas de calcul.

Cette méthode présente l'avantage d'être très réactive, permettant d'éviter des collisions et de circuler dans un environnement incertain sans besoin d'une grande quantité

d'informations. En outre, elle n'accumule pas les données et utilise une carte locale égo-centrée.

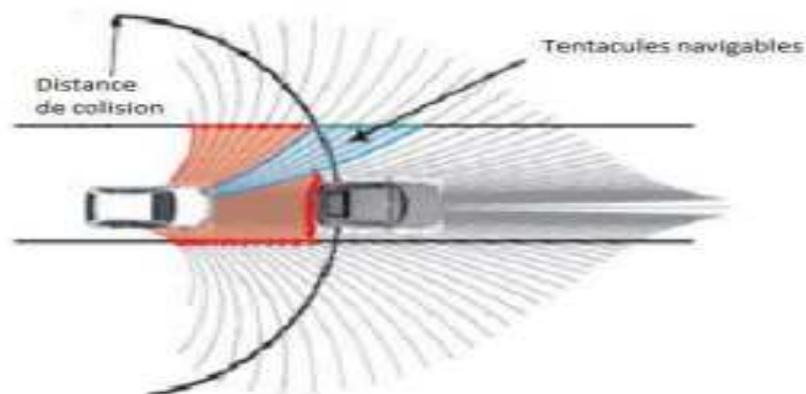


Figure II.9 : Principe de la méthode des tentacules.

II.4.1.4 Méthodes des champs de braquage constant :

La méthode des champs de braquage utilise les obstacles pour limiter l'angle de braquage dans un domaine continu. Cette méthode illustrée par la figure II.10 est très proche de la méthode de base des tentacules.

L'inconvénient majeur de cette approche est la difficulté à suivre une trajectoire de référence en utilisant un braquage constant.

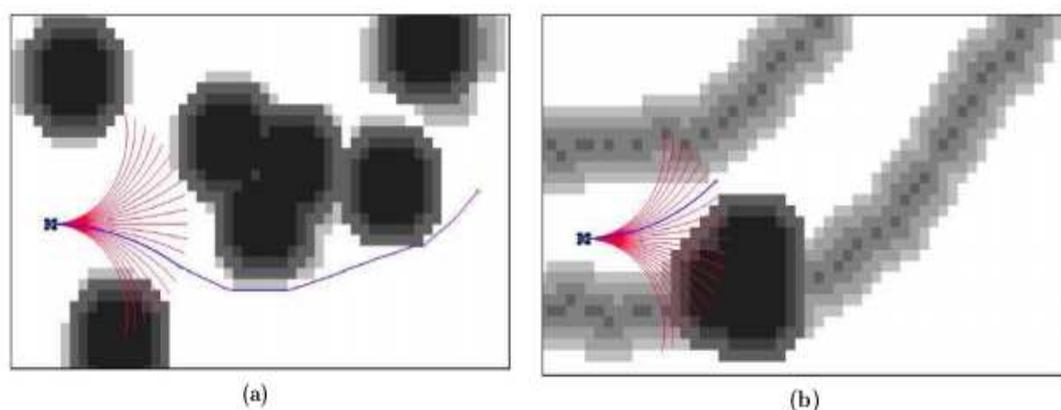


Figure II.10 : Principe de la méthode des champs de braquage constant : (a) milieu non structuré, (b) milieu structuré.

II.4.1.5 Fenêtre dynamique :

Le principe de la méthode est de travailler dans l'espace des vitesses plutôt que dans l'espace des positions. La recherche de commandes à envoyer au robot pour éviter les obstacles mobiles est alors effectuée directement dans cet espace.

La dynamique du robot est introduite dans la méthode par la réduction de l'espace de recherche aux seules vitesses admissibles par le robot, en prenant en compte les contraintes dynamiques telles que les valeurs minimales et maximales des vitesses linéaire et angulaire .

L'avantage de cette méthode est qu'elle tient compte de la dynamique du robot mobile dans le calcul d'une solution. Avec les fenêtres dynamiques, un évitement d'obstacle robuste à vitesse élevée est possible. L'inconvénient de cette méthode est qu'elle est sensible aux minimas locaux, car elle ne considère que la pose de l'objectif ; il n'y a aucune information sur la connectivité de l'espace libre.

II.4.1.6 Méthodes basées sur un modèle de prédiction :

Ayant la configuration initiale et la configuration finale (souhaitée), les solutions les plus proches sont trouvées. Ensuite, une interpolation entre les solutions proches, puis une optimisation est faite pour trouver la meilleure solution sans nécessiter beaucoup de calcul.

Pour décrire la trajectoire entre la configuration initiale et la configuration finale, une interpolation polynomiale est faite. La figure II.11 illustre un changement de voies à l'aide d'une interpolation polynomiale (5^{ème} ordre).

Généralement, cette méthode est couplée avec deux contrôleurs, longitudinaux et latéraux, pour suivre approximativement la trajectoire de référence, représentée par une séquence d'états désirés sur un horizon de temps t [8]. L'avantage de cette méthode est qu'elle prend en compte la dynamique du robot mobile. En outre, elle couple généralement les étapes de la planification et du contrôle, ce qui permet une bonne optimisation. Son inconvénient se situe au niveau du temps de calcul relativement grand pour une optimisation non linéaire.

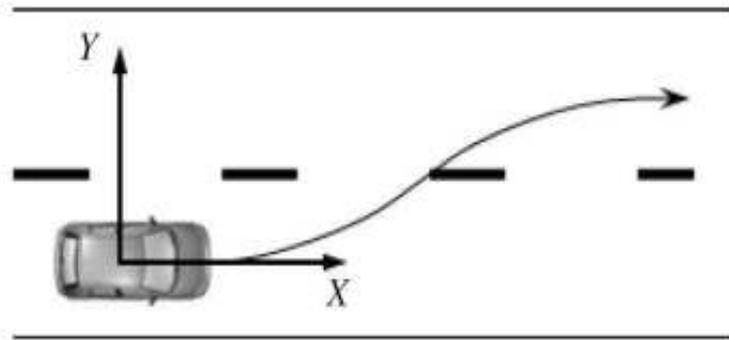


Figure II.11 : Prédiction de trajectoire : interpolation polynomiale

II.4.2 Notre approche de navigation :

Notre approche de navigation est caractérisé par l'utilisation d'un capteur à ultrasons afin d'assurer la perception de l'environnement; tout en mesurant la distance qui sépare notre robot avec l'obstacle.

Cette méthode se base sur l'évitement d'obstacles par contournement effectuant principalement trois types de déplacements soit : avancement, rotation de 90° ou bien -90° .

La réactivité par rapport aux obstacles est parmi les principaux avantages que cette méthode présente, ainsi la navigation dans des environnements peut connus. En outre, elle ne permet pas la navigation dans des environnements encombrés (plus d'un seul obstacle).

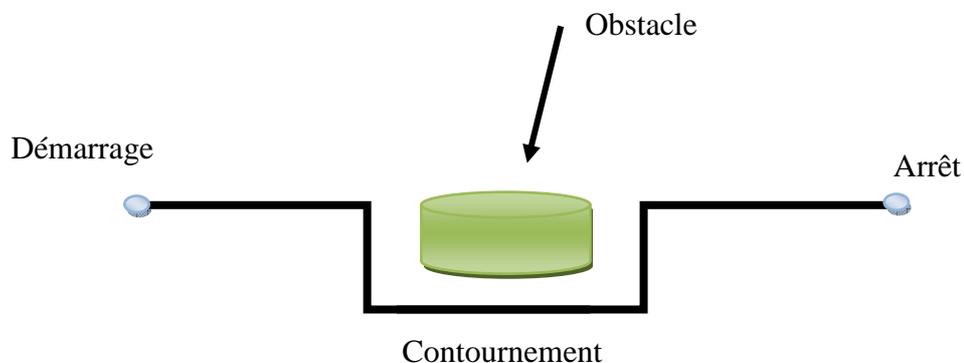


Figure II.12 : Principe de la méthode utilisé dans notre application.

Conclusion :

Nous avons présenté brièvement la problématique de la localisation et de la navigation en robotique mobile, ainsi que l'éventail d'approches et de techniques utilisées pour les effectuer. De part, nous avons mis l'accent sur l'état de l'art des différentes méthodes de représentation des environnements d'intérieurs.

Nous avons présenté les principales approches de planification de trajectoires appliquées aux robots mobiles. De cette étude, il ressort que les approches de planification qui contiennent une moindre marge d'erreur quant à la fiabilité, est la méthode des tentacules qui est représenté comme l'un des axes majeurs dans ce domaine de recherche.

Le succès de ces approches dépend du fonctionnement rentable du robot et au composants internes et externes proprement dit. Dans le chapitre suivant on feras une description exhaustive du robot mobile « Dr. Robot I90 ».

Chapitre III

Introduction :

Nous voulons dans ce présent chapitre aborder les principaux composants du robot mobile « *Dr. Robot I90* », nous décrivons sa structure formelle, représentée par sa hauteur, sa largeur, sa longueur et son poids. Puis nous entamons ses différents modules matériels et logiciels et leurs caractéristiques, ainsi que les relations qu'entretiennent entre eux.

Pour que le robot puisse fonctionner, une liaison sans fil relie les contrôleurs de mouvements et de sensations à une unité de commande.

III.1 Définition :

Principalement *Dr. Robot I90* montré à la figure III.1 est un outil sophistiqué à l'usage des chercheurs, développant des applications avancées en robotique; telles que la surveillance à distance dans différents environnements, localisation pour la navigation et patrouille autonome, et diverses fonctionnalités.

Il est Conçu pour répondre à des exigences qui permettront de résoudre des difficultés et des imprécisions que connut ce domaine de recherche pointue. Il est le résultat d'efforts de collaborateurs afin de façonner une plate forme qui pourra être plus légère, économique du point de vu de la consommation énergétique, durablement fonctionnel à cout d'achat relativement abordable.



Figure III.1: Dr. Robot I90.

III.2 Caractéristiques:

Nous avons souligné plus haut, que le souci du bon usage et d'adéquation du robot pour toutes circonstances est pris en compte [24]:

- Il pèse 5 kg de poids, donc portable.
- Quant à ses dimensions et format :

D'une largeur de 43,0 cm, et de 38,0 cm de Longueur, d'une hauteur de 30.0 cm.

- Doté d'une caméra à haute résolution de 704*480 pixels, avec deux audio.
- Une connexion complètement sans fil 802.11g, entièrement intégré avec deux canaux de communication série, supportant les protocoles UDP et TCP / IP.
- Sa vitesse maximale atteint 75 cm/sec.
- Il dispose des capteurs de navigation tels que trois ultrasoniques, sept infrarouges détecteurs de distance, et deux détecteurs de mouvements humains.
- Deux moteurs à courant continu, avec un encodeur optique intégré à 800 mesures par cycle.
- Une Protection du circuit compréhensive.
- Il peut porter une charge utile additionnelle de 15 kg.
- Faculté de commande à distance.
- Grande plate-forme de montage supérieur pour les appareils supplémentaires tels qu'un ordinateur portable.
- Les options mise à niveau :
 - Localisation et navigation autonome sans collisions.
 - Base de vision, douée d'un capteur de localisation d'intérieure GPS fournissant la position précise du robot ainsi que les informations de direction.
 - Six heures de validité pour la batterie et l'énergie du système.

III.3 Architecture de robot:

La technologie assignée au *Dr. Robot I90*, répond à l'architecture de calcul de distribution, initialement développé comme étant humanoïde. En utilisant cette approche, le contrôle de haut niveau du robot est maintenue par un PC / serveur distant ou local communiquant par une liaison sans fil sécurisée.

La Fonctionnalité de bas niveau est gérée par un processeur embarqué de signaux numériques (DSP). Avec la bande passante élevée du module wifi 802.11g, le système peut télécharger toutes les données du capteur à un PC ou d'un serveur à des fréquences supérieures à 10Hz. De même, la répartition audio et vidéo pour la surveillance directe [24].

Les Commandes et les instructions envoyées au robot via la même liaison sans fil, passent également à des fréquences supérieures à 10Hz et offrant un accès et un contrôle en temps réel.

Le *Dr. Robot I90* comprend tous les composants logiciels de développement du robot, ce qui permet un accès facile à toutes les données et informations dans un environnement de programmation standard Microsoft Windows (par exemple, C#). Selon l'approche de l'aide d'un PC séparé pour le contrôle de haut niveau, il n'y a plus de restrictions à bord sur la puissance de traitement, la mémoire et le stockage d'un système mobile [24].

III.4 Modules matériels du robot mobile i90 :

Nous représentons ci-dessous les principaux éléments qui se trouvent sur la base mécanique, électronique et informatique du robot :

III.4.1 Partie mécanique :

Constituée de trois principaux éléments distincts :

III.4.1.1 Châssis mécanique I 90-ME :

L'équilibre de la base de notre robot est maintenu grâce au rattachement de l'ensemble des éléments le constituant sur une plaque.

III.4.1.2 Moyens de locomotion :

La locomotion est très importante pour une base mobile telle que notre robot. Les principes constitutifs qui assurent le déplacement et la stabilité du robot pour les différents mouvements possibles sont nombreux et variés. Nous distinguons comme grandes catégories les roues.

a) Roues motrices:

La mobilité de notre structure est assurée par deux roues motrices, d'un diamètre de sept (07) pouces équivalent à 18 cm; parallèles fixées à l'avant de châssis par un axe. Ces roues sont plus faciles à contrôler, elles permettent au robot mobile de se déplacer plus rapidement.

Pour les mettre en mouvement, Chaque roue est actionnée séparément par un moteur à CC 12V avec un encodeur optique.



Figure III.2 : Roue motrice.

b) Roue folle:

Notre robot se dispose d'une roue libre à l'arrière, qui assure sa stabilité et son équilibre pour les différents mouvements possibles.



Figure III.3 : Roue folle.

III.4.1.3 Moteurs :

Pour circuler dans son environnement, le robot est muni de plusieurs moteurs pour convertir l'énergie électrique en énergie mécanique.

Ces moteurs peuvent faire tourner les roues du robot afin d'effectuer des déplacements. La motorisation du robot est constituée essentiellement de moteurs à courant continu 12V, et de servomoteurs.

a) Moteur à courant continu :

Le moteur à courant continu est composé de 2 parties principales : L'inducteur ou dite stator (partie fixe), dont le rôle est de créer un champ magnétique \vec{B} , qui peut être à aimants permanents ou constitué d'électro-aimants. L'induit ou dite rotor (la partie mobile) porte des conducteurs alimentés par un courant continu et soumis à une force qui va faire tourner le rotor en créant le couple moteur [25]. Les roues seront fixées sur l'axe au milieu des aimants constituant le stator-rotor.

Le module de control utilisé est le *MDM5253*, qui est une commutation à trois canaux avec une carte amplificateur de puissance. Il peut être commandé directement par le signal PWM de conduite à une fréquence de 20 KHz et une tension de fonctionnement de 5.0V à 28.0V. Caractérisé par une sortie analogique courant de rétroaction et une sortie de protection de court-circuit [26].

Pour les trois canaux indépendants, le *MDM5253* contient également le retour d'informations et des connecteurs de courant intégré, pour des capteurs de position tels que des potentiomètres précis.



Figure III.4 : Module de control DM5253.

b) Servomoteurs RC séparés :

C'est un type de moteur électrique spécifique pour un déplacement angulaire, capable d'atteindre des positions prédéterminées, puis de les maintenir ou à l'inverse, de tourner de façon continue dans un sens donné à une vitesse comprise de 60 et 120 TPM (tours par minute) [24]. Ils sont commandés par un signal PWM de fréquence 50Hz, et possèdent une connectique composée de 3 câbles :

- rouge : VCC a +5v.
- noir : masse.
- orange ou jaune : signal de commande PWM.

Dans le domaine de la robotique, le servomoteur a pour but de permettre la rotation des capteurs en connaissant leurs angles de rotation.

III.4.2 Partie électronique :**III.4.2.1 Capteurs :**

Les capteurs ont une place prépondérante dans le système de traitement d'un robot. Ils ont pour fonction d'acquérir des données provenant de l'environnement et de l'informer sur ses propres actions, en vérifiant l'état de ses actionneurs. La capacité de perception et les besoins de la localisation et de la navigation sont des fonctions essentielles en robotique, c'est la raison pour laquelle nous dotons notre plateforme mobile est dotée de deux types de Capteurs : extéroceptifs et proprioceptifs.

a) Capteurs extéroceptifs :

Les capteurs extéroceptifs fournissent des informations sur le monde extérieur au robot (température, contact, etc.). Deux familles de capteurs extéroceptifs embarqués peuvent être identifiées : les capteurs télémétriques et les systèmes de vision.

- **Capteurs télémétriques :**

Il existe différents types de capteurs télémétriques qui permettent de mesurer la distance aux éléments de l'environnement qui sont :

- **Capteurs à ultrasons :**

Ils utilisent la mesure du temps de vol d'une onde sonore réfléchiée par les obstacles pour estimer la distance. Ces obstacles se situant dans un cône relativement large (d'angle au sommet d'environ 30 degrés) [27].

Le capteur de distance par rapport à un obstacle est constitué par un émetteur et un récepteur à ultrason. L'émetteur envoie une onde ultrason qui est renvoyée par un obstacle rencontré sur son chemin. Le récepteur à ultrasons recevra cette onde.

Le module utilisé est le *DUR5200* qui peut détecter l'information d'une plage de 10 cm à 340 cm avec une fréquence de travail de 40 KHz. Les données de distance sont précisément présentées par l'intervalle de temps entre l'instant où la mesure est activée et l'instant où le signal d'écho est reçu [26].



Figure III.5 : Module à ultrason DUR5200.

- **Capteurs infrarouges :**

Ce type de capteurs est constitués d'un ensemble émetteur/récepteur fonctionnant avec des radiations non visibles, dont la longueur d'onde est juste inférieure à celle du rouge visible, il délivre alors une tension en fonction de la distance de l'obstacle détecté. Ils sont souvent utilisés comme détecteurs de proximité.

Entre les différents types de capteurs infrarouges, notre robot est doté du module *GP2Y0A21YK*, qui est un capteur de proximité infrarouge fabriqué par Sharp. Il dispose d'une

alimentation de +5V, une sortie analogique de 3,1V à 4V, un signal de sortie de 0 à 3 V et une mesure de distance de 10cm à 80cm [26].



Figure III.6 : Module à infrarouge GP2Y0A21YK.

➤ **systèmes de vision :**

En robotique ces systèmes sont basés sur l'utilisation d'une caméra pour percevoir l'environnement. Nous citons plusieurs types de caméras utilisées dans ce domaine, tels que caméras simples, stéréoscopiques et panoramiques.

Celle utilisée dans notre cas, est une caméra panoramique AV-PTZ-VH avec une résolution de 704x480 pixels, une capacité d'inclinaison, un zoom optique et numérique 10x et une liaison audio bidirectionnelle [26]. Ainsi l'image recueillie permet d'avoir une vision de l'environnement sur 360 degrés autour de la caméra.

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.

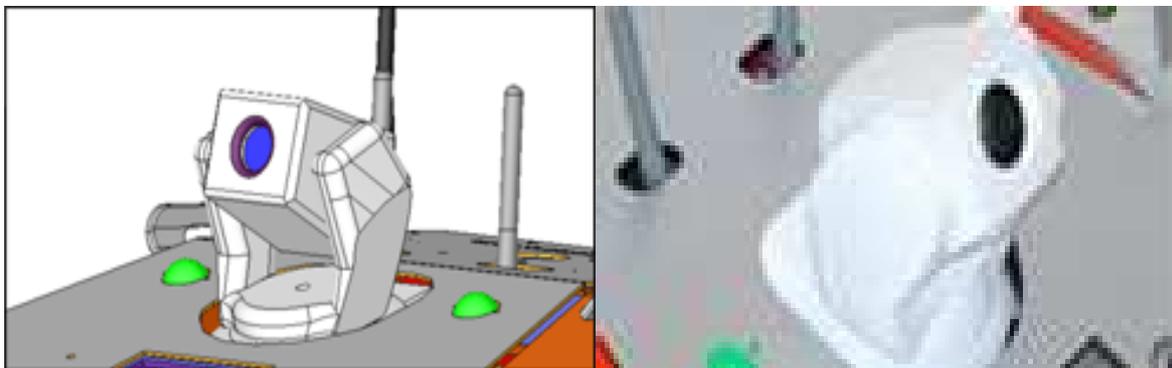


Figure III.7 : Caméra panoramique PTZ.

b) Capteurs proprioceptifs :

Ils fournissent des informations propres au comportement interne du robot, ainsi qu'ils renseignent sur son état à un instant donné (vitesses des roues, accélérations etc.)[28].

On peut les regrouper en deux familles:

- **Les capteurs de déplacements :**

- **odomètres :**

L'odométrie permet d'estimer le déplacement du robot à partir du calcul de la mesure de rotation des roues. Cette mesure est en général effectuée par un codeur optique disposé sur l'axe de la roue, ou sur le système de transmission. L'information de déplacement nécessitera la connaissance de l'entraxe, diamètre des roues et la structure mécanique et cinématique du véhicule. La précision de mesure dépend du glissement, plus important dans les virages.

- **Accéléromètre :**

L'accéléromètre est un capteur capable de mesurer les trois accélérations linéaires en un point donné selon trois axes orthogonaux, Ils peuvent donc mesurer les déplacements d'un robot.

Le *DTA5102* est le module de capteur utilisé, capable de mesurer à la fois l'accélération statique (inclinaison ou orientation) causée par la gravité terrestre et le choc ou la vibration causée par un impact Sur une plage de ± 1 g. Il utilise un accéléromètre micro-usiné IC combiné avec un filtre passe-bas et un amplificateur de signal [26].



Figure III.8 : Module capteur d'accélération-DTA5102.

➤ **Capteur de position :**

Pour avoir la position de la plateforme, on utilise deux capteurs de type potentiomètre. Le principe de fonctionnement consiste à l'exploiter de la relation entre la position du curseur et la tension à la borne du potentiomètre puis à faire l'étalonnage avec la position du robot [13].

• **Capteurs d'attitude :**

Les plus utilisées sont des gyromètres et gyroscopes, ils permettent respectivement d'effectuer une mesure de la vitesse et la variation angulaire.

c) **Autres capteurs :**

• **GPS :**

Il est conçu principalement pour avoir un système de localisation universel possible, vu les besoins de localisation et de précision que ces calculs offrent [27].

Le GPS (Global Positioning System) fonctionne grâce au calcul de distance qui sépare un récepteur GPS de plusieurs satellites, la somme des informations nécessaires au calcul de la position des satellites étant transmise régulièrement au récepteur. Ce dernier mesurant la distance qui le sépare des satellites reconnaîtra ses coordonnées [29].

• **Capteur humain :**

Un robot mobile est équipé de détecteurs de mouvements et de présence, pour détecter des sources de chaleur en mouvement tels que le corps humain, qui émet un rayonnement thermique dont l'intensité en fonction de sa température de surface. Un capteur pyroélectrique adapté au domaine infrarouge reçoit ce rayonnement, et le convertit en tension électrique.

Le *Dr. Robot I90* utilise le détecteur *DHM5150*, capable de détecter la présence humaine dans la gamme jusqu'à 500 cm, ainsi que la direction du mouvement de cette chaleur dans la plage allant jusqu'à 150cm avec l'utilisation de deux modules [26].



Figure III.9 : Module détecteur de mouvement humain-DHM5150.

III.4.2.2 LED :

Une LED (diode électroluminescente) est un semi-conducteur qui émet de la lumière quand un courant le traverse. Les matériaux semi-conducteurs que les LED utilisent transforment l'énergie électrique en un rayonnement électromagnétique visible, c'est-à-dire en lumière.

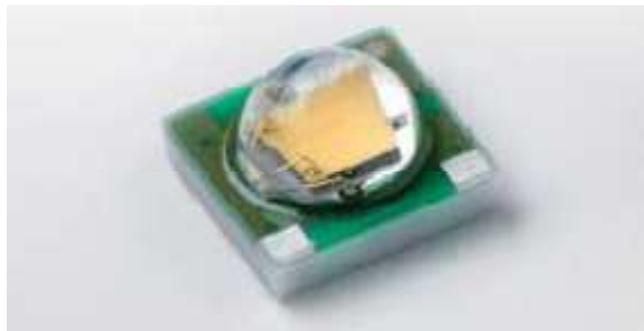


Figure III.10 : LED.

III.4.2.3 Gestion de l'alimentation et le système de recharge :

Les différents circuits et actionneurs du robot nécessitent des tensions et des courants différents. Pour cela le *Dr. Robot I90* est alimenté avec deux (02) batteries 12V de type Pack [24].

III.4.2.4 Câble de connexion-CCR2150 :

C'est un Câble avec connecteurs femelles DSUB 9 broches aux deux extrémités, avec 1.5m et une connexion croisée à 9 fils RS232 DTE a DTE.



Figure III.11 : Câble de connexion.

III.4.2.5 Ecran couleur tactile LCD PDM1950 :

Ce type d'écran est programmable, à une résolution de 320*240 pixels [26]. Son principe de fonctionnement se résume à une dalle composée de plusieurs couches qui sont transpercées par une lumière polarisée; venant de néons afin de créer des images constituées de millions de cellules minuscules.

III.4.3 Partie informatique :

III.4.3.1 Contrôleur du mouvement et des sensations du robot-PMS5005- :

Le *PMS5005* est utilisé pour la détection, le contrôle et l'exécution du mouvement. Il rend les modules de commande du moteur et de communication sans fil transparente pour les utilisateurs. Et Il élimine les risques de conception, de rationaliser le développement du matériel et du logiciel. Comme il peut conduire 6 moteurs standards, surveiller la tension du système et du référence [26].



Figure III.12 : Contrôleur du mouvement et des sensations du robot-PMS5005- .

III.4.3.2 Contrôleur Multimédia PMB5010 :

Ce contrôleur multifonctionnel permet de rendre les modules tels que l'audio et le module d'image transparente pour les utilisateurs. Il est caractérisé par une horloge en temps réel, des entrées numériques pour les signaux offs et un port d'extension parallèle pour la détection et de contrôle [26].

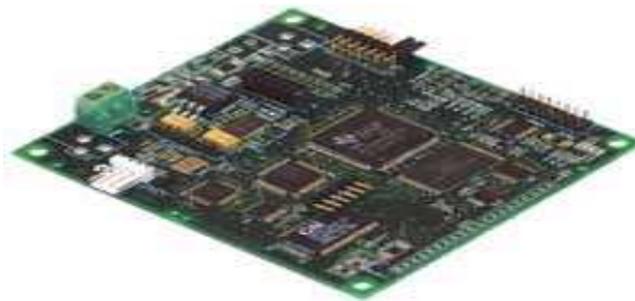


Figure III.13 : Contrôleur Multimédia-PMB5010.

III.4.3.3 Interface MCR3210 :

C'est une interface d'adaptation de matériel compatible avec les spécifications électriques RS232 standard. Il est nécessaire pour la connexion filaire de PMS5005 à un PC ou à des appareils, avec RS232 le port de communication série. Propulsé par une simple alimentation de 3.3V, caractérisé par 9 broches connecteur D-SUB [26].



Figure III.14 : Interface MCR3210.

III.4.3.4 Joystick de contrôle:

La fonction d'un joystick est de convertir une action manuelle en une tension électrique. C'est l'interface idéale du prototypage ou pour s'intégrer dans une commande à distance. Il intègre deux potentiomètres; un pour l'axe avant/arrière, l'autre pour l'axe gauche/droite.

III.4.3.5 Communication wifi:

La communication avec l'opérateur est réalisée par une liaison sans fils radio fréquence pour commander notre robot. Cette liaison se résume en une transmission de données à distance qui pourra se réaliser à l'aide d'une télécommande bidirectionnelle, capable d'envoyer et de recevoir des données.

Le module wifi utilisé est le *WFS802G- série (802.11g)* intégré pour ajouter un réseau de logiciel sans fil au robot.

Le WFS802G intègre un TCP/IP de réseau complètement développés, établit des contacts pour permettre l'accès à un réseau local ou internet. Il inclut aussi un serveur web encadré qui peut être utilisé pour configurer de façon distante l'appareil attaché [26].

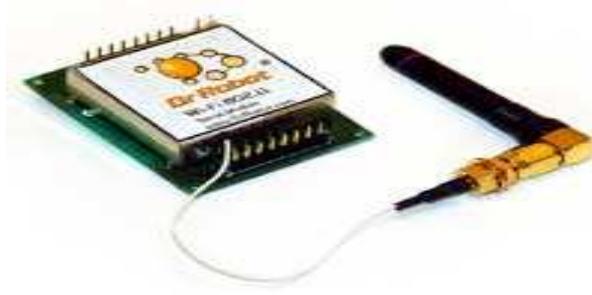


Figure III.15 : Module *WFS802G série (802.11g)*

III.4.3.6 Modem :

Cet appareil électronique permet d'envoyer et de recevoir de l'information par une ligne de télécommunication.

III.5 Modules logiciels :

Ces modules servent à interpréter les données perçues par les capteurs afin d'y extraire des informations et d'envoyer les ordres relatifs aux moteurs pour faire fonctionner le robot. Tous ces modules sont assemblés dans une architecture robotique.

III.5.1 logiciel SDK :

Le SDK (Soft Development Kit) contient les installations de gestion de la mémoire, la communication du système et de l'interface utilisateur, ainsi que les services publics pour entrée/sortie d'audio et vidéo, acquisition de données de capteur et de commande de mouvement [30].

Dans le cadre du système architecture du Robot, tous les contrôleurs sont connectés en une chaîne.

Tous les systèmes de contrôle embarqués ont au moins deux ports de SCI pour les communications du système : port orifice-supérieure et port orifice-inférieure, avec la direction relativement au contrôleur central. L'architecture de connexion contrôleur est représentée dans la figure III.16 :

Les APIs (Application programmable interface) sont l'interface entre le logiciel de niveau d'application et le système matériel du robot. Les Programmes développés en utilisant les SDK (Soft Development Kit) fonctionnent sur le PC pour envoyer et recevoir des données vers et à partir du *Dr. Robot I90* par fil ou sans fil.

Le microprogramme sur les contrôleurs embarqués prend soin de tous les faibles opérations au niveau des modules fonctionnels du système, telles que l'acquisition des données, rapidité de boucle de commande de mouvement de bas niveau, l'image et la capture audio et de compression, la lecture audio et communication sans fil, ils sont transparents au système de logiciel de haut niveau en cours d'exécution sur le contrôleur de PC central.

Tout le développement de logiciel du système peut être exercé uniquement dans le système de PC à accès facile. Les SDK sont disponibles pour Visual studio.

Le Programme utilisateur utilise ce composant dans le programme Visuel studio pour communiquer avec les contrôleurs du *Dr. Robot I90* PMS5005 ou/et PMB5010 [30].

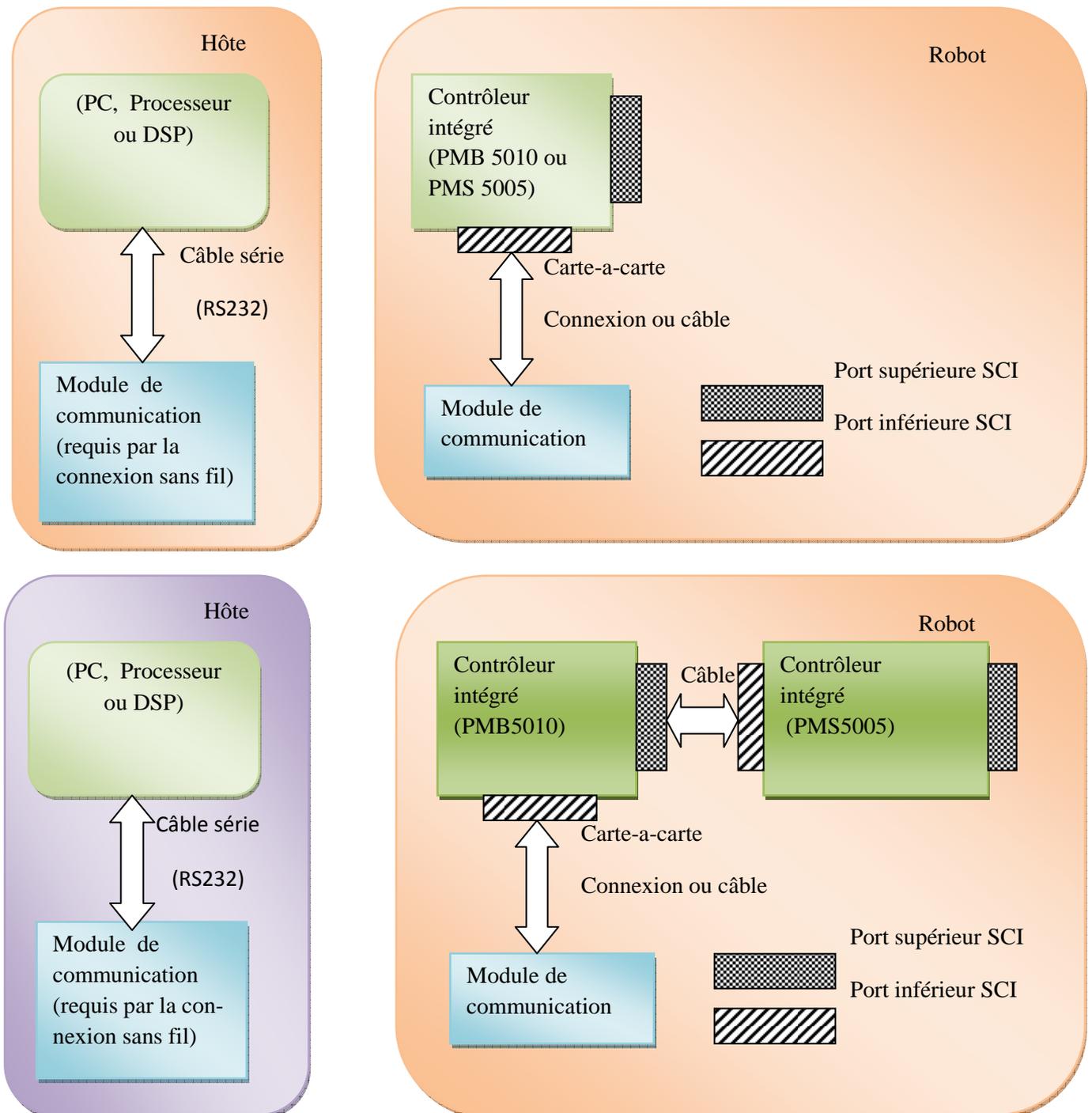


Figure III.16 : L'architecture du système du Robot

Conclusion :

La présentation des constituants du *Dr. Robot I90* basée sur trois parties fondamentales ; mécanique, électronique et informatique, ainsi que ses fonctions relèvent d'une haute technologie. Disposé d'un logiciel SDK de reconnaissance entre le robot et le logiciel de programmation *studio visuel*. Ils assurent la liaison qui épargne et réduit le degré d'imprécisions et d'erreurs.

A un stade élevé de la recherche scientifique, il assure une fiabilité et une interprétation d'informations précises utiles pour promouvoir des acquis théoriques et pratiques dans différents domaines.

Il permet a la fois d'atteindre des buts d'une extrême précision grâce a sa perfection en prototype humain, aussi il permet d'accomplir des tâches délicates.

En aval, nous avons choisis une trajectoire pour notre robot mobile, tout en évitant des obstacles prémédités ; puis rejoindre sa trajectoire initiale en utilisant ses différents composants afin d'accomplir et d'exécuter un programme dans le but d'expérimentation de sa validité scientifique, ce que nous allons voir dans le chapitre suivant.

Chapitre IV

Introduction :

La description matérielle et logicielle appropriée au *Dr. Robot I90*, nous mène à l'élaboration d'un programme permettant de piloter le robot selon une planification de trajectoire et de mouvements, bien définie.

Ainsi, le *Dr. Robot I90* est programmé avec le logiciel orienté objet studio visuel en raison de ses propriétés de stabilité, de facilité de construction itérative et de réutilisabilité. Chaque objet se décrit par un ensemble d'attributs et de méthodes appropriés à ces derniers.

IV.1 Langage de programmation C# :**IV.1.1 Pourquoi l'orienté objet :**

L'orienté objet est un ensemble de règles, de concepts, d'outils du langage de programmation pour rendre le développement plus aisé grâce au groupement d'ensembles des fonctions et des variables, toute en facilitant leurs gestions par groupe et en ouvrant de multiples possibilités. Un langage Orienté objet est décrit par une syntaxe qu'on peut créer et manipuler en utilisant des mots clefs.

Cet ensemble de concepts et d'outils permet par exemple de grouper dans un coin les objets dont leurs données peuvent servir soit à faire l'interface avec l'utilisateur, ou bien aux objets qui servent à appliquer les calculs et procédures logiques de fond [31]. De ce fait nous pourrons disposer d'un code simple et lisible.

IV.1.2 Définition :

Visual Studio 2008 est un puissant langage orienté objets, créé par Microsoft qui est doté d'un environnement de programmation riche en outils comportant toutes les fonctionnalités nécessaires pour créer des projets C# de toute taille, Sa syntaxe est très inspirée du C, C++ et le java qui sont des langages de programmation classique.

Son but est d'être plus lisible et plus pratique que l'assembleur pour simplifier la vie des développeurs.

Un programme est divisé en deux principales parties [31]:

IV.1.2.1 Données :

Un programme manipule des données caractérisées par un nom et un type. Les différentes données utilisées en C# sont des variables définies selon leurs types et des opérateurs numériques. Ainsi, les variables sont identifiées par un nom que l'on définit impérativement au début du programme, et servent à stocker des valeurs. On distingue deux types de variables qui sont : variables de type structure et variables de type classe ou variables d'instance.

Une variable de type structure stocke des données, elle se manipule via sa valeur et elle ne peut pas être créée sans être initialisée, On cite :

- les nombres entiers, réels et décimaux (Int, Float, décimal...etc.).
- les caractères et chaînes de caractères (char).

Quant au variable de type classe, elle stocke les références aux données qui sont considérées comme des objets automatiquement initialisés lors de la création de cette variable qui est manipulée via son adresse. Les données peuvent être : les booléens (ces des valeurs clefs true, false ou nul), les objets (Object) ou bien Chaîne de caractères (string) [32].

IV.1.2.2 Instructions qui manipulent les données: On distingue:

- Les instructions élémentaires de C# sont les suivantes :
 - Ecriture sur l'écran.
 - Le changement de type de variable.
 - Les opérateurs.
 - Affectation de la valeur d'une expression à une variable...etc.
- Les instructions appropriées au contrôle du déroulement d'un programme sont :
 - Arrêt (utilisant la méthode exit).
 - Structure de choix simple (Condition if).
 - Structure de cas (swich).
 - Structure de répétition (boucles for, foreach, while et do-while).

De nombreuses fonctions C# sont susceptibles de générer des exceptions, c'est-à-dire des erreurs. Lorsqu'une fonction est susceptible de générer une exception, le programmeur devrait la gérer dans le but d'obtenir des programmes plus résistants aux erreurs [31].

IV.1.3 Classes, interfaces et événements:

IV.1.3.1 Notion d'objet:

La complexité du monde réel nous conduit à définir, classifier, organiser et hiérarchiser les différents objets qui le composent. Ces objets possèdent des propriétés et des comportements propres. Une même famille d'objets, c'est-à-dire avec des caractéristiques communes, peut être regroupée dans un ensemble appelé classe [33].

Un objet est une entité qui contient des données qui définissent son état (on les appelle des champs, attributs, ...) et des fonctions (on les appelle des méthodes).

La programmation orientée objet reprend à son compte ces notions d'objet et de classe. Le développeur pourra à sa guise créer une classe, définissant les caractéristiques et les propriétés d'une même famille d'objet. Du point de vue externe, l'objet est une boîte noire accessible à travers d'une certaine interface (les méthodes) [31].

IV.1.3.2 Définition d'une classe:

Nous avons ici la définition d'une classe, donc d'un type de données. Lorsqu'on va créer des variables de ce type, on les appellera des objets ou des instances de classe.

Une classe est donc un moule à partir duquel sont construits des objets. Les membres ou champs d'une classe peuvent être des données, des méthodes et des propriétés. Les propriétés sont des méthodes particulières servant à connaître ou fixer la valeur d'attributs de l'objet [33]. Ces champs peuvent être accompagnés de l'un des trois mots clés suivants [32]:

- **Private** : Un champ privé qui limite l'accès à la classe même.
- **public** : Un champ public qui donne tout l'accès à toutes les méthodes de toutes les classes.
- **protected** : Un champ protégé n'est accessible que par les seules méthodes internes de la classe ou d'un objet dérivé.

En général, les données d'une classe sont déclarées privées alors que ses méthodes et propriétés sont déclarées publiques. Cela signifie que l'utilisateur d'un objet (le programmeur)

- n'aura pas accès directement aux données privées de l'objet.
- pourra faire appel aux méthodes publiques de l'objet et notamment à celles qui donneront accès à ses données privées.

Remarque :

Une application C # peut contenir plusieurs classes qu'il faut spécifier lors de la compilation de l'application.

Les classes seront rangées un peu comme dans des dossiers en utilisant des espaces de noms, le plus important étant système contenant des classes que la plupart des applications utilisent pour interagir avec le système d'exploitation. Une directive *using* est utilisé afin d'accéder à un objet de cet espace de noms.

IV.1.3.3 Interface:

Une interface est un modèle de classe qui contient des ensembles de prototypes de méthodes ou de propriétés qui forment un contrat. Une classe qui décide d'implémenter une interface s'engage à fournir une implémentation de toutes les méthodes définies dans l'interface. C'est le compilateur qui vérifie cette implémentation [32].

Les propriétés et méthodes de l'interface ne sont définies que par leurs signatures. Elles ne sont pas implémentées (n'ont pas de code). Ce sont les classes qui implémentent l'interface qui donnent du code aux méthodes et propriétés de l'interface.

IV.1.3.4 Evénements:

Les évènements font parti d'un type de programmation à part entière, on parle de programmation événementielle qui s'oppose partiellement à la programmation procédurale. La programmation événementielle consiste à réagir en fonction des évènements. Les applications à interfaces graphiques sont souvent associées aux concepts de programmation événementielle [31]. Par exemple, un utilisateur qui clique sur un des boutons de l'interface lance une procédure. Le fait que l'utilisateur puisse à tout moment agir sur n'importe quel bouton entend que ce n'est pas une procédure à variable qui se déroule de manière uniforme. C'est bien agir en fonction des interactions. Les évènements se déclarent avec le mot clé *event*.

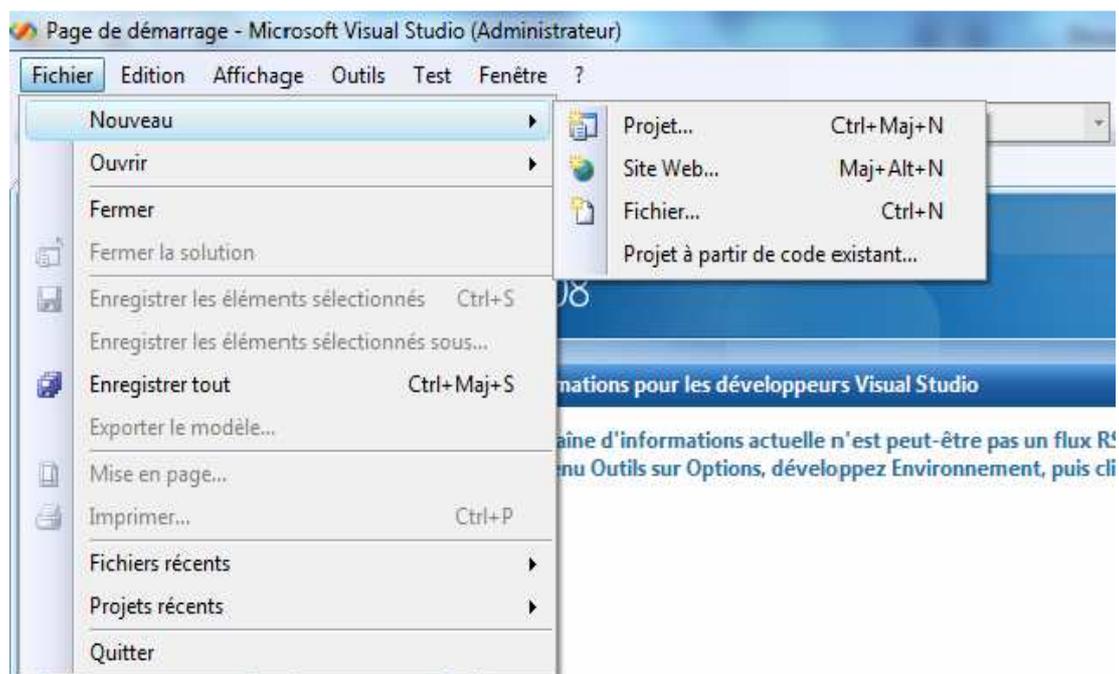
IV.2 Programmation élaboré:

IV.2.1 Création d'une interface graphique:

Le projet créé est une interface graphique contenant des différents objets qui seront programmés séparément pour effectuer les différentes procédures.

Les étapes à suivre pour créer un nouveau projet C# dans Visual studio sont multiples. Pour commencer, on lance Visual studio 2008, puis on clique sur fichier_ nouveau_ projet.

On choisie *Windows* comme type de projet, ainsi *Application Windows Form* comme modèle. Ensuite on donne un nom pour le projet « *Dr. RobotI90* » et on spécifie un chemin d'accès. La fenêtre IV.1 résume ces étapes.



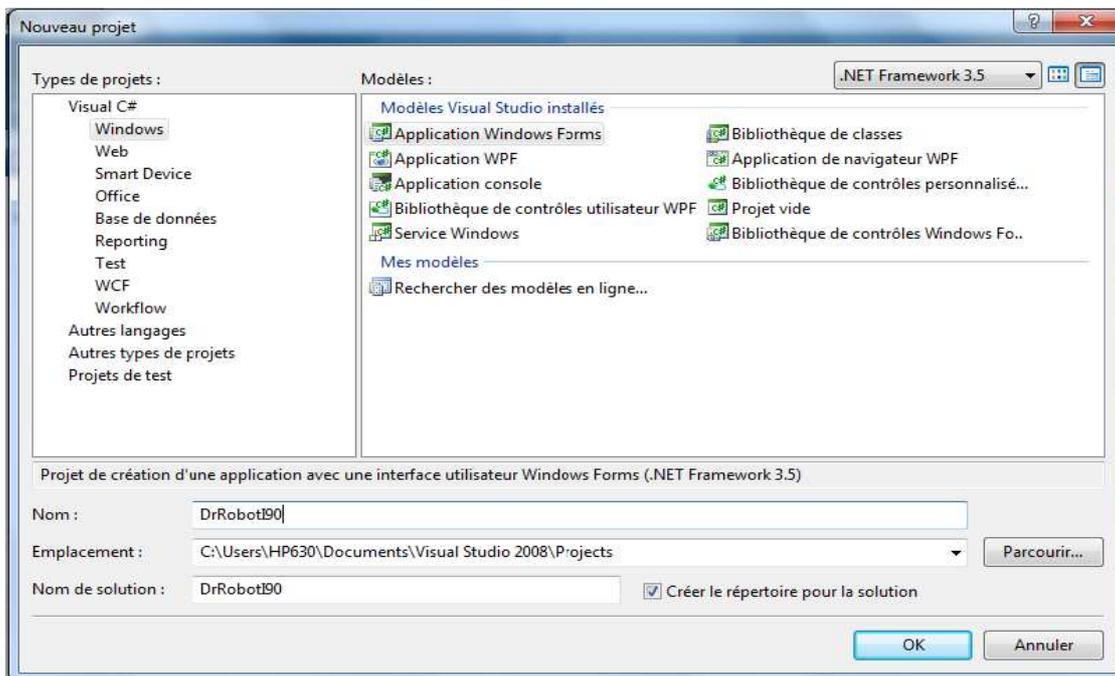


Figure IV.1 : Nouveau Projet C#

La partie droite de l'écran correspond à l'explorateur de solution qui permet de voir les bibliothèques du projet quant à la partie gauche, un écran graphique dénommé *Form1* est créé par défaut, comme le montre la figure IV.2 :

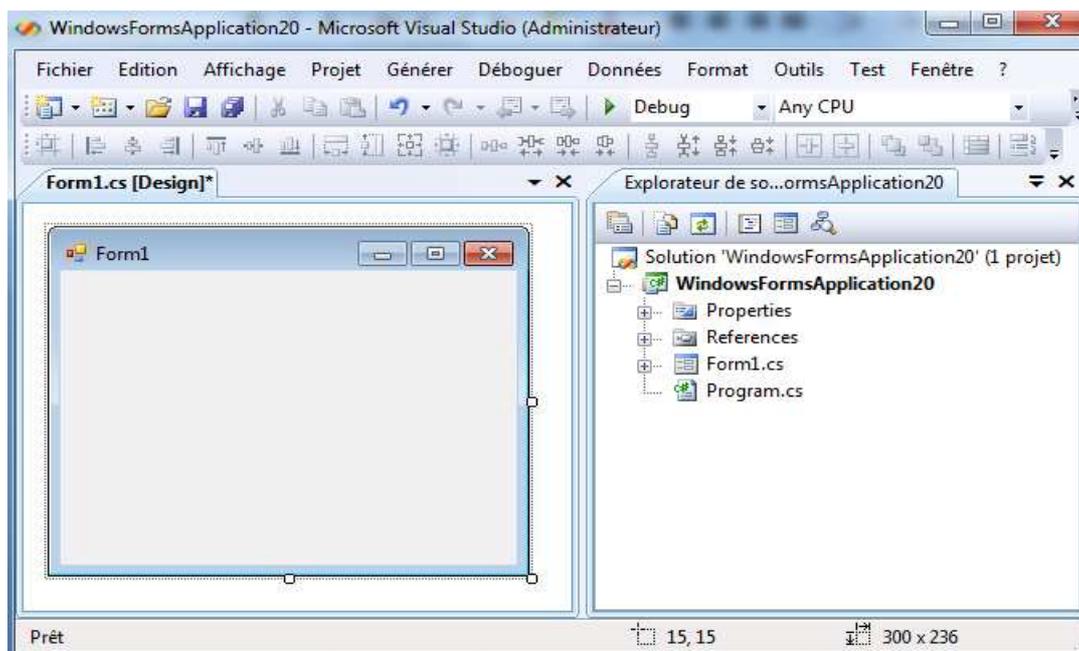


Figure IV.2: Application Windows Form.

En suite on a ajouté à *Form1* sept objets de type Botton programmé pour l'arrêt, rotation et le déplacement du robot dans les différentes directions, cinq labels pour les données dont deux moteurs (gauche, droite) et trois capteurs à ultrasons et un track bar pour le contrôle de la vitesse.

L'interface créée est montrée à la figure IV.3

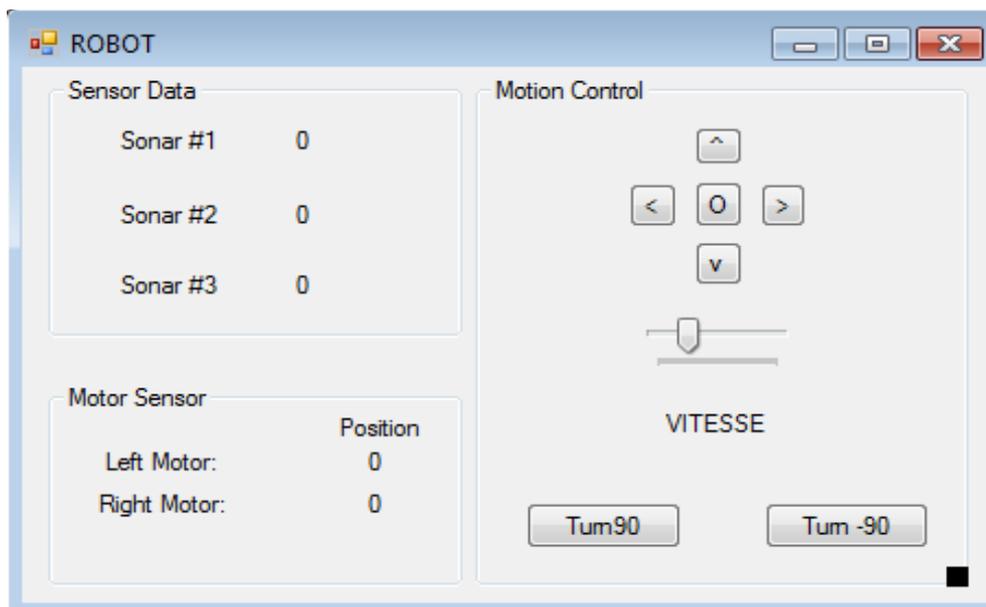


Figure IV.3: Interface graphique.

IV.2.2 Récupérations des données de positions des moteurs, et du capteur à ultrason (labels) [30]:

IV.2.2.1 position des moteurs :

- *Motor Sensor Event.*

Cet événement se déclenche après la réception des différentes commandes de rotation, d'avancement et de reculement par les deux moteurs. Ensuite, deux encodeurs et potentiomètres récupèrent les données de positionnement des moteurs grâce à cette instruction : « *Short Get Encoder Pulse X () ;* ».

IV.2.2.2 Capteur sonar :

- *Standard Sensor Event.*

Le déclenchement de cet événement est relativement relié à l'arrivée des données provenant des capteurs sonars, infrarouges et humains.

Comme dans notre cas on a utilisé que les capteurs sonars, donc la récupération des données sera avec l'instruction « Short Get Sensor Sonar X () ; », qui permet de convertir la distance qui sépare le module ultrasonique DUR5200 et l'objet devant, en une valeur numérique, puis l'afficher. L'unité est exprimée en « m ».

IV.2.3 Les différentes opérations utilisées pour le control de mouvement des moteurs (boutons):

- *Void Dc Motor Velocity Non Time Ctr All (Short cmd1, Short cmd2, Short cmd3, Short cmd4, Short cmd5, Short cmd6);*

Avec :

- *short cmd 3, 4, 5, 6* est la vitesse de la cible pour les différents canaux fixée à -32767.

Cette opération Envoi la commande de contrôle de vitesse à tous les six canaux de control du moteur sur le contrôleur de mouvement (PMS5005) au même temps. Ce contrôleur va entrainer le moteur pour atteindre la vitesse désirée.

Quand certains moteurs ne sont pas contrôlés, leurs valeurs de commande doivent être définies comme -32768. Cela signifie aucun-contrôle.

IV.2.3.1 Marche Avant :

```
private void btnForward_Click(object sender, EventArgs e)
{
    short speed = (short)trackBar1.Value;

    myRobot.DcMotorVelocityNonTimeCtrAll((short)speed, (short)-speed, -32767, -32767,
-32767, -32767);
}
```

IV.2.3.2 A droite :

```
private void btnRight_Click(object sender, EventArgs e)
{
    short speed = (short)trackBar1.Value;

    myRobot.DcMotorVelocityNonTimeCtrAll((short)-speed, (short)-speed, -32767, -32767,
-32767, -32767);
}
```

IV.2.3.3 A gauche:

```
private void btnLeft_Click(object sender, EventArgs e)
{
    short speed = (short)trackBar1.Value;

    myRobot.DcMotorVelocityNonTimeCtrAll((short)-speed, (short)speed, -32767, -32767,
-32767, -32767);
}
```

IV.2.3.4 Arrière:

```
private void btnBack_Click(object sender, EventArgs e)

{

    short speed = (short)trackBar1.Value;

    myRobot.DcMotorVelocityNonTimeCtrAll((short)-speed, (short)speed, -32767, -32767,
-32767, -32767); }
```

IV.2.3.5 Arrêt de robot (bouton stop):

- ```
private void btnStop_Click(object sender, EventArgs e)
{
 tmrMotorDemo.Enabled = false;
 myRobot.DisableDcMotor(0);
 myRobot.DisableDcMotor(1);
}
```

**IV.3.3.6 Turner 90°:**

```
private void btnTurn90_Click(object sender, EventArgs e)

{

 Turn(90);

}
```

**IV.3.3.7 Turner -90°:**

```
private void btnTurn-90_Click(object sender, EventArgs e)

{

 Turn(-90)

}
```

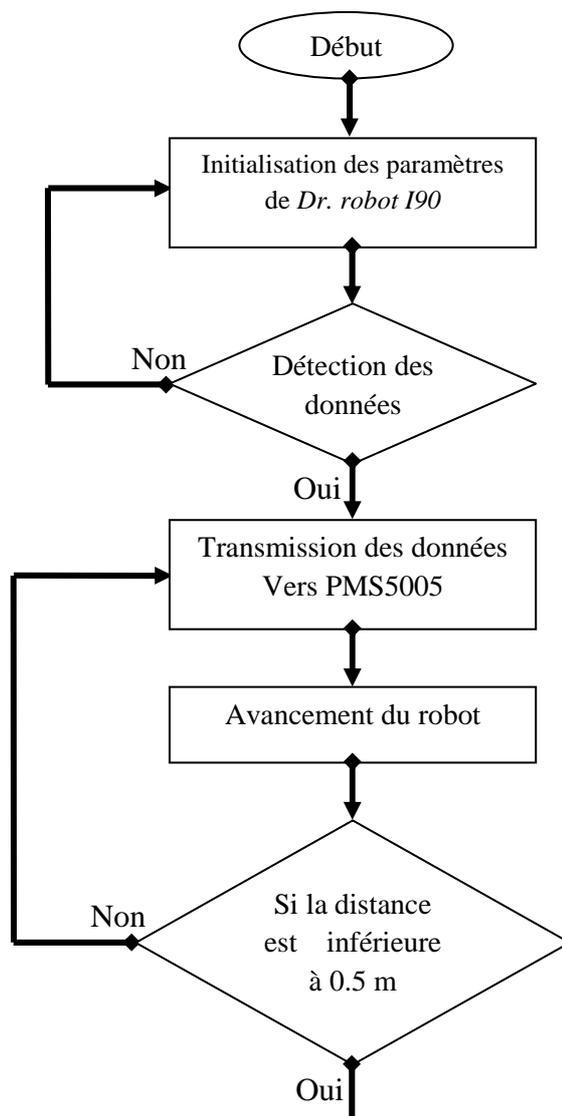
### IV.3 Enchaînement des mouvements :

Un système robotique peut être défini comme un système complexe composé de sous systèmes. Par exemple, pour un robot mobile, deux sous systèmes peuvent être facilement identifiés : un système pour effectuer des tâches de manipulation et un système de locomotion (base mobile à roue). L'actionneur généralement utilisé pour mouvoir ces types de robot est le moteur électrique.

L'exécution de la tâche de planification de trajectoire avec évitement d'obstacles sera accomplie suite à la succession des évènements à chaque bouton figurant dans l'interface graphique comme le montre la figure IV.4 tels que l'avancement, reculement, rotation à droite et à gauche et l'arrêt.

### IV.4 Organigramme du fonctionnement du robot:

Les différentes taches qu'exécute le robot sont décrites comme suit :



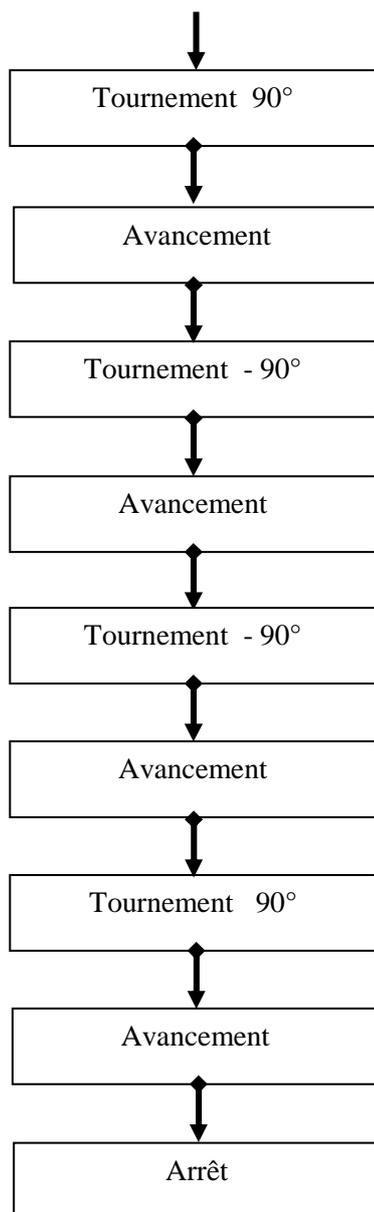


Figure IV.4 : Enchaînement des mouvements.

Le calcul de la distance qui sépare le robot avec l'obstacle se fait dès la connexion du robot. Après le clique sur le bouton avant un avancement tout droit du robot est remarqué, jusqu'à la détection de l'obstacle d'une distance de moins de 50 cm, le robot fait un contournement puis rejoint sa trajectoire ensuite il s'arrête.

IV.5 Fonctionnement de robot:

La connexion du robot à un réseau son fil est une tâche primordiale pour son fonctionnement ; et c'est a partir des différentes informations tels que le nom du robot, les adresses IP du wifi et de la caméra comme le montre la figure suivante :



Figure IV.5: Interface de connexion.

Deux points d'accès seront appelés pour établir des connexions de communication avec le système électronique disponible sur *Dr. RobotI90* comme la montre la figure ci-dessous :

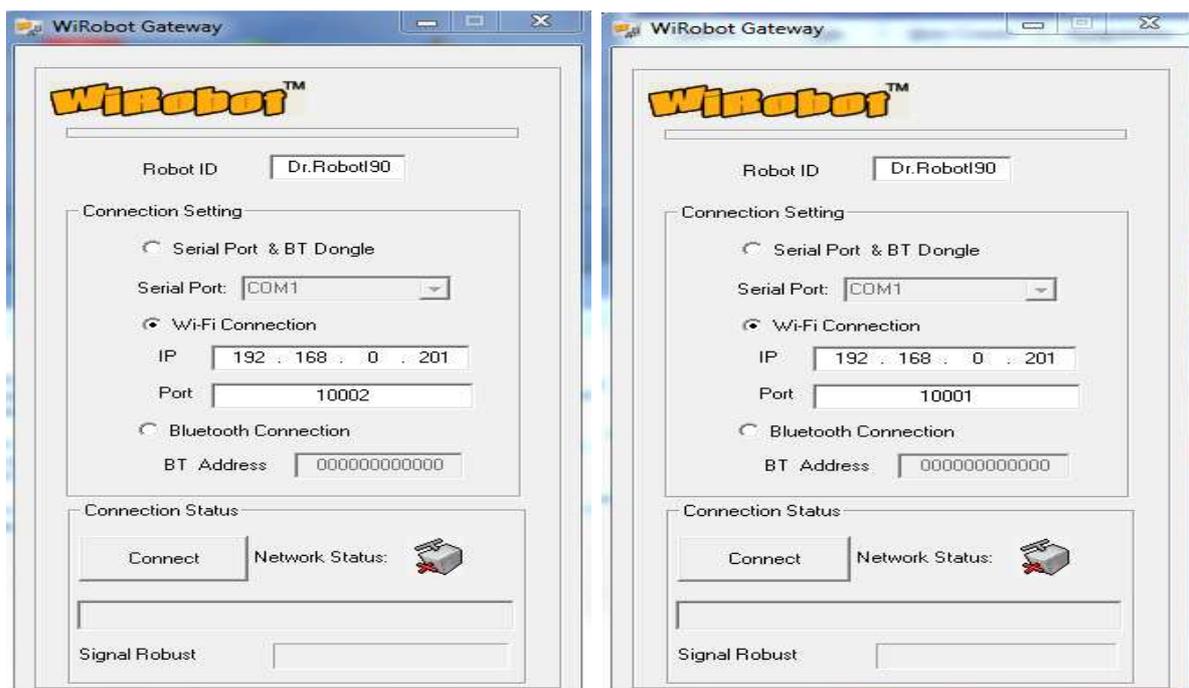


Figure IV.6: les deux points d'accès au robot

Où le port 10001 nous permet de contrôler la puissance du robot, tandis que le contrôle du mouvement est effectué par le port 10002.

### **Conclusion :**

Après les reconnaissances acquises en programmation orienté objet et l'utilisation des manuels propres au robot qui nous ont aidé à comprendre le fonctionnement des différents composants et leurs SDK, nous sommes arrivés à élaborer un programme qui fait marcher le robot dans les différentes directions et détecter des obstacles.

Les détecteurs à ultrasons que nous avons utilisés possèdent une faculté exceptionnelle de détecter les obstacles d'une manière sûre et fiable. Pour cette raison, les détecteurs à ultrasons sont de plus en plus utilisés.

# Conclusion générale

Le travail présenté, dans ce mémoire, s'inscrit dans le cadre de la commande à distance d'un robot mobile, à partir d'une planification de trajectoire qui lui permet de gérer ses mouvements avec l'évitement des obstacles prémédités, en exécutant un programme écrit en langage de programmation orienté objet C Sharp (C#).

Nous avons commencé par des généralités sur la robotique tout en citons les différents robots mobiles à roues et leurs modèles cinématique.

La suite du travail a été consacrée aux différentes méthodes appropriées à la localisation et la navigation qu'utilisent les robots mobiles lors de l'exécution des tâches ainsi que notre approche.

Par la suite nous sommes concentré sur la description des différents composants logiciels et matériels permettant le bon fonctionnement de notre robot mobile *Dr. Robot I90*, ainsi que leurs caractéristiques.

A la fin de mémoire, nous avons pu commander le robot mobile *Dr. Robot I90* grâce à une interface graphique créée utilisant le langage de programmation orienté objet C# qui est basés sur la modalité la clarté du code.

Pour les perspectives, on se propose de combinés les techniques de l'asservissement Visual avec celle de planification de trajectoire d'un robot mobile dans le but d'obtenir de meilleures performances de poursuite ainsi qu'une simplicité d'implémentation en pratique via le langage de programmation orienté objet C#.

# Références

# Bibliographique

- [1] Captain Nick Ham, Oregon Wing, Director of Aerospace Education, Introduction to robotic, 2011.
- [2] [www.elephorm.com](http://www.elephorm.com) > Code > Java > Apprendre la Programmation Orientée Objet.
- [3] Salhi Farid, Rezki Dahmane, « Commande à distance d'un chariot mobile avec évitement d'obstacles », Université Mouloud Mammeri de Tizi-Ouzou, 2010.
- [4] Christopher Parlitz-Schunk GmbH & Co.KG, «Service robotics is a key market of the future», 2011.
- [5] Direction Régionale de l'industrie, de la recherche et de l'environnement (DRIRE). Enjeux et défis de l'industrie de la robotique en Ile-de-France, 2010.
- [6] Brandt, Mark Van Den, US technological innovation systems for service robotics, 2010.
- [7] Nicole Merle-Lamoot, Gilles Pannetier, « Le développement industriel future de la robotique personnelle et de service », en France, 2012.
- [8] David Filliat, «Robotique mobile », École nationale supérieure de technique avancées paris Tech, octobre 2011.
- [9] Mathieu Hobon, « Modélisation et optimisation de la marche d'un robot bipède avec genoux anthropomorphiques », Par Spécialité Automatique.
- [10] Laetitia Matignon, « Introduction à la robotique », Université de Caen, France.
- [11] Boimond Jean-Lois, Cours-robotique.
- [12] Eric Beaudry, « Planification en intelligence artificielle pour la robotique mobile », Mémoire de maîtrise, Faculté des sciences, Université de Sherbrooke, Août 2006.
- [13] G. Campion, G. Bastin et B. D'Andréa-Novel. Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots. IEEE Transactions on Robotics and Automation, vol. 12, no. 1, pages 47–62, 1996.

- [14] Pin, F.G., Killough, M.S., « A New Family of Omnidirectional and Holonomic Wheeled Platforms for Mobile Robots», IEEE transactions on robotics and automation, Vol 10, N°4, August 1994
- [15] Bouali Abdelmalek, « Planification de trajectoire pour un robot mobile », Thèse de magister à l'université d'El-Hadj lakhdar-Batna, juin 2012.
- [16] Bernard Bayle, « Cours robotique mobile », À l'université de Strasbourg.
- [17] Frédéric Giamarchi, « Petits robots mobiles »,ed. Dunod, 2006.
- [18] Omar Ait-Aider, « Localisation référence modèle d'un robot mobile d'intérieur », Thèse en vue de l'obtention du doctorat à l'université d'Evry, 19 décembre 2002.
- [19] Benmakhlouf Abdeslam, «Contrôleur flou pour la navigation d'un robot mobile d'intérieure », Thèse en vue de l'obtention du diplôme de magister en robotique à l'université de Batna, 11 décembre 2006.
- [20] C.Durien, « Algorithmes de localisation d'un robot mobile dans un milieu balisé par mesure de distance ou d'angle de gisement en tenant compte des mesures aberrantes, Algorithmes de calibration et de recalage du champ de balise », Thèse de doctorat à l'université de paris sud, 1989.
- [21] Jean Laneurit, «Perception multisensorielle pour la localisation d'un robot mobile en environnement extérieur, application aux véhicules routières », Thèse pour obtenir le grade de docteur à l'université balaise pascal-clemontII, juillet 2006.
- [22] Oualid Djekoune, « Localisation et guidage du robot mobile atrv2 dans un environnement naturel », Thèse pour l'obtention du grade de doctorat à l'université Houari Boumediene, 15 décembre 2010.

- [23] Abdelatif Baba, «Cartographie de l'environnement et suivi simultané de cibles dynamiques par un robot mobile », Thèse en vue de l'obtention du doctorat a l'université
- [24] Dr Robot I90, «Wireless Networked Autonomus Mobile Robot with High Resolution Pan-Tilt-Zoom Camera», Quick Start Guide, 2010-2013.
- [25] Cours «moteur courant continu.odt», Au lycée polyvalent louis couffignal a Strasbourg, juillet 2006.
- [26] [www.Dr Robot.com/products\\_item.asp?itemNumber=i90](http://www.DrRobot.com/products_item.asp?itemNumber=i90).
- [27] Cyril Drocourt, «Localisation et modélisation de l'environnement d'un robot mobile par coopération de deux capteurs omnidirectionnels », Thèse pour l'obtention du grade de docteur de l'université de technologie de compiegne, fevrier 2002.
- [28] Barnard Bayle, «Robotique mobile », Ecole National Supérieure de Physique de Strasbourg Université Louis Pasteur, 2008-2009.
- [29] Philippe Béguyot, Bruno Chevalier et Hana Rothova, Le GPS en agriculture : Principes, applications et essais comparatifs, Educagri, 2004.
- [30] Dr Robot I90, «Wi Robot SDK Application Programming interface (API) Reference Manual-(for MS Windows) », January 2010
- [31] Serge Tahé. « Apprentissage du langage C# 2008 et du Framework .NET 3.5», L'université d'Angers, Mai 2008.
- [32] « C# l'essentiel st consacré ». Dotnet-France Association, 1<sup>er</sup> Avril 2008.
- [33] « Le langage C# introduction au framework.NET ».
- <http://www.siteduzero.com/tutoriel-3-10601-programmation> en-java.html.