

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



Université de Batna
Faculté de Technologie
Département de Génie Electrique



Mémoire

Présenté en vue de l'obtention du diplôme de Magister en Electronique

OPTION

Robotique

PAR

MALLEM ALI

Ingénieur d'état en électronique

THEME

Algorithmes d'aide à la navigation d'un robot mobile

Soutenu le :...../...../2011

Membres du jury

Président:	Mr. ARAR DJEMAI	<i>M.C</i>	<i>U. de Batna</i>
Rapporteur:	Mr. SLIMANE NOUREDDINE	<i>M.C</i>	<i>U. de Batna</i>
Examineur:	Mr. BOUTAMINA BRAHIM	<i>MC</i>	<i>U. de Constantine</i>
Examineur:	Mr. ABDESSEMED YASSINE	<i>M.C</i>	<i>U. de Batna</i>

Remerciements

J'exprime ma profonde gratitude à mon encadreur Pr. **SLIMANE NOUREDDINE**, professeur à l'université de *BATNA*, pour m'avoir accueilli au sein de son

équipe et pour avoir dirigé ce travail de thèse avec autant de dévouement. Je le remercie très sincèrement pour la qualité de ses conseils, sa disponibilité, sa patience et son aide tout au long de ma thèse.

Mes vifs remerciements vont à monsieur **Djemai ARAR**, maître de conférences à l'université de *BATNA* pour avoir accepté de présider mon jury, je le remercie pour son aide fraternel et ses encouragements incessants.

J'adresse mes plus sincères remerciements à monsieur **Yassine ABDESSEMED** maître de conférences à l'université de *BATNA* et monsieur **Brahim BOUTAMINA**, maître de conférences à l'université de *CONSTANTINE* pour l'intérêt qu'ils ont porté à ce travail en acceptant d'être membres de jury de ma thèse.

Je tiens à remercier particulièrement tout le personnel du département d'Electronique pour leur gentillesse et leur disponibilité.

Mes remerciements vont également à tous les enseignants ainsi qu'à tous les responsables de notre département qui ont veillé au bon déroulement de notre formation.

Enfin, nous exprimons notre remerciement à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

SOMMAIRE

Sommaire

Introduction générale.....	1
Chapitre 1 : Généralités sur la robotique mobile	
1.1 Introduction	4
1.2 Robot mobile autonome.....	5
1.3 Les différents types de terrain.....	6
1.4 Les systèmes de perception.....	8
1.4.1 Les capteurs proprioceptifs.....	9
1.4.1.1 Les capteurs de déplacement.....	9
1.4.1.2 Les capteurs d'attitude.....	10
1.4.2 Les capteurs extéroceptifs.....	11
1.4.2.1 Les capteurs télémétriques.....	11
1.4.2.2 Les systèmes de vision.....	14
1.5 Les méthodes de localisation.....	19
1.5.1 Localisation a l'estime.....	19
1.5.2 Localisation absolue.....	20
1.5.2.1 Repères artificiels.....	20
1.5.2.2 Repères naturels.....	22
1.5.3 Méthodes hybrides.....	23
1.6 Conclusion.....	23
Chapitre 2 : Modélisation des robots mobiles	
2.1 Définitions.....	24
2.2 Roulement sans glissement et contraintes non holonomes.....	25
2.2.1 Roulement sans glissement.....	25
2.2.2 Contraintes non holonômes.....	26
2.3 Les grandes classes de robots mobiles et leurs modèles.....	28
2.3.1 Disposition des roues et centre instantané de rotation.....	28
2.3.2 Robots mobiles de type unicycle.....	29
2.3.2.1 Description.....	29
2.3.2.1 Modélisation.....	30

2.3.3 Robots mobiles de type tricycle et de type voiture.....	34
2.3.3.1 Description.....	34
2.3.3.2 Modélisation.....	35
2.3.4 Robots mobiles omnidirectionnels.....	37
2.3.4.1 Description.....	37
2.3.4.2 Modélisation.....	38
2.4 Propriétés du modèle cinématique d'un robot.....	39
2.4.1 Représentation d'état.....	39
2.4.2 Commandabilité des robots mobiles à roues.....	39

Chapitre 3 : Navigation des robots mobiles

3.1 Introduction.....	41
3.2 Le problème de navigation.....	41
3.3 Méthodes sans trajectoire.....	42
3.3.1 Champs de potentiels artificiels.....	42
3.3.2 Réseaux de neurones.....	44
3.3.2 Logique floue.....	45
3.4 Méthodes de suivi de trajectoire.....	47
3.5 Discussion.....	48
3.5.1 Synthèse des méthodes de navigation.....	48
3.5.2 Nomenclature et lien avec une architecture de commande pour l'autonomie.....	49
3.6 Poursuite de trajectoire.....	53
3.6.1 Modélisation du système.....	54
3.6.2 Calcul des paramètres l et α	56
3.6.3 Simulation du suivi de trajectoire.....	58
3.6.4 Résultats de simulation.....	58
3.7 Conclusion.....	63

Chapitre 4 : Modèle et algorithme d'évitement d'obstacles

4.1 Introduction.....	64
-----------------------	----

4.2 Méthode des champs de potentiel.....	64
4.3 Méthode Vector Field Histogram.....	67
4.4 Méthode de la fenêtre dynamique.....	68
4.5 Développement de la méthode des champs de potentiel	70
4.5.1 Potentiel attractif.....	72
4.5.2 Potentiel répulsif	73
4.5.3 Planification de trajectoires guidées par champs potentiels.....	74
4.6 Résultats de simulation.....	75
4.7 Conclusion.....	79
 Chapitre 5 : Système flou pour la navigation d'un robot mobile	
5.1 Introduction.....	81
5.2 La logique floue en robotique.....	82
5.3 Architecture structurelle du système.....	83
5.4 Comportement de navigation basé la logique floue.....	84
5.4.1 Module de détection d'obstacles.....	84
5.4.2 Comportement de navigation libre.....	85
5.4.2.1 Fuzzification.....	86
5.4.2.2 Inférence	87
5.4.2.3 Defuzzification.....	87
5.4.2.4 Résultats expérimentaux	87
5.4.3 Comportement d'évitement d'obstacles.....	93
5.4.3.1 Fuzzification.....	93
5.4.3.2 Inférence.....	94
5.4.3.3 Defuzzification.....	95

5.4.3.4 Résultats expérimentaux.....	95
5.5 Conclusion.....	101
Conclusion générale.....	102
Bibliographie	

INTRODUCTION GENERALE

L'homme a toujours cherché à concevoir des entités à l'image de l'être humain sur lesquelles il pourrait exercer son autorité, exécutant ainsi ses moindres désirs. Les premières approches remontent à l'époque médiévale où des automates étaient placés dans les églises afin d'impressionner les paysans, fascinés par ces "êtres suprêmes". Les automates sont ensuite devenus des jouets populaires au 18^{ème} siècle sous forme miniature. La littérature a également fourni un bon nombre d'ouvrages concernant ces êtres artificiels. On peut citer Mary Shelley, qui en 1818 écrivit Frankenstein, créature à l'apparence humaine dont les parties de son corps tenaient ensemble fixés par des boulons et des écrous. En 1921, Karel Capek introduisit le terme "robot" dans sa pièce de théâtre "R.U.R." (Rossum's Universal Robot), du tchèque "robota" signifiant travail forcé, corvée. Il raconte l'histoire d'un savant appelé Rossum, ayant réussi à mettre au point des créatures semblables physiquement à des êtres humains, que son fils exploita au sein de son entreprise. Le terme "robotique" fut lui amené par l'écrivain Isaac Asimov, qui proposa les trois lois suivantes de la robotique :

- Loi 1 : Un robot ne peut blesser un être humain, par son inaction, ni permettre qu'un humain soit blessé.
- Loi 2 : Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres sont en contradiction avec la première loi.
- Loi 3 : Un robot doit protéger sa propre existence aussi longtemps qu'une telle protection n'est pas en contradiction avec la première et/ou la deuxième loi.

Un tournant fut franchi dans les années 60, où les robots industriels firent leur apparition essentiellement dans l'industrie automobile. Ces derniers n'ont plus l'apparence humaine comme les avaient imaginés leurs créateurs, mais ont l'allure de bras mécaniques, exécutant aveuglément les mêmes mouvements. Ce premier type fut appelé robots de première génération. La deuxième génération quant à elle introduisit les fonctions de perception et de régulation issues de l'automatisme et permettant les corrections de mouvement par rapport à l'évolution des paramètres de l'environnement. Finalement, la troisième génération concerne les robots capables d'interagir avec l'environnement, de percevoir et de modéliser l'ensemble de leur espace d'évolution et de prendre des décisions.

Les différentes architectures mécaniques existantes permettent d'envisager tous les types de déplacement dans l'espace. Le plus simple est bien évidemment l'utilisation de roues, qui à moindre coût peut assurer un déplacement du robot sur un sol plat. Toutefois, dans le domaine de "l'intelligence artificielle", les progrès sont moindres. Même si la problématique reste plus modeste, la capacité d'un robot à intégrer des informations sur l'environnement relève donc d'une certaine forme d'apprentissage.

La navigation d'un robot mobile en milieu conditionné, c'est à dire un milieu d'intérieur parfaitement connu, dans lequel on a disposé ou pas des balises permettant au robot de se repérer, reste quelque chose de réalisable. Néanmoins, la partie "intelligente" du système est loin d'être maîtrisé. Cela montre bien les difficultés que représente la réalisation d'un robot autonome capable d'évoluer en milieu inconnu. Pourtant bon nombre de chercheurs se sont intéressés à la possibilité de donner à un robot les moyens de percevoir un environnement complètement inconnu, motivés par de vastes domaines d'applications. On peut notamment penser aux applications d'aides aux personnes handicapés, comme les fauteuils roulants intelligents, le domaine automobile où on souhaiterait pouvoir réaliser des véhicules n'ayant plus besoin de chauffeurs. Mais comment reproduire cette analyse de l'environnement que notre cerveau effectue en temps réel et de façon naturelle ?

Les progrès en matière de capteurs ont été significatifs ces dernières années en terme de précision, même si cela est souvent au détriment de la quantité de données qu'il faut manipuler, qui reste toutefois loin de celle des "capteurs" humains. Par exemple la rétine contient environ 125 millions de capteurs, contre 25 millions pour un film photographique, et un nombre encore bien inférieur pour une caméra CCD. De plus, les informations fournies par ces divers capteurs peuvent être de type tout a fait différent : distance avec un objet, vitesse, déplacement relatif, ... La diversité des informations fournies par ces capteurs a nécessité un développement de méthodes de fusion de données permettant d'obtenir une information sensorielle globale. Finalement, on peut s'apercevoir que la conception d'un robot et la perception de l'environnement sont déjà quasiment maîtrisés : Le robot peut se déplacer et percevoir des informations sur le monde l'entourant pour se localiser. Toutefois, une dernière étape est encore nécessaire pour libérer le robot d'une quelconque connaissance préalable : un apprentissage de son milieu d'évolution en fonction des informations sensorielles. Aussi, dans ce travail on essaye de contribuer, par l'apport de quelques algorithmes, à une navigation autonome du robot mobile. Pour cela on propose le plan de travail suivant :

Organisation du mémoire

Dans un premier temps, nous proposons des généralités sur la robotique mobile afin de sensibiliser le lecteur sur les travaux entrepris pour permettre une bonne navigation à un robot mobile, c'est-à-dire pour aboutir à la réalisation d'algorithmes efficaces permettant une totale autonomie à un robot mobile. Après une présentation des différents types de robots mobiles et des contraintes du terrain sur lequel ils sont conçus pour évoluer, nous abordons les différents capteurs de perception utilisés en robotique mobile ainsi que les différentes techniques de localisation des robots mobiles.

Le deuxième chapitre est consacré à la modélisation des différents types de robots mobiles et à leur étude cinématique. De plus, nous proposons une étude détaillée de leur commandabilité.

Dans le troisième chapitre, une étude des différents types de navigation des robots mobiles est exposée. La détermination d'une trajectoire ainsi que la proposition d'une commande pour le déplacement du robot mobile sont aussi exposées dans ce chapitre avec quelques travaux de simulation.

Dans le quatrième chapitre, nous présentons quelques méthodes d'évitement d'obstacles, notamment la méthode des champs de potentiels qui est le point essentiel de notre travail dans ce chapitre. Des tests et des travaux de simulation y afférant y sont présentés.

Dans le dernier chapitre, nous présentons un système flou réactif capable de faciliter la tâche de navigation, de surmonter les incertitudes présentées dans le monde réel et d'aider le robot à suivre de meilleurs chemins. Le problème des minimas locaux qui peut se présenter dans la méthode des champs de potentiels est carrément résolu par l'algorithme proposé. Deux types de contrôleurs sont introduits pour permettre cette amélioration. Des travaux de simulation illustrent ce modèle de navigation proposé.

Nous concluons enfin ce mémoire par un récapitulatif de l'ensemble des solutions proposées tout au long des chapitres, et sur l'analyse qualitative de l'ensemble des résultats obtenus.

CHAPITRE 1

GENERALITES SUR LA ROBOTIQUE MOBILE

1.1 Introduction

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

On peut estimer que les robots mobiles à roues constituent le gros des robots mobiles. Historiquement, leur étude est venue assez tôt, suivant celle des robots manipulateurs, au milieu des années 70. Leur faible complexité en a fait de bons premiers sujets d'étude pour les roboticiens intéressés par les systèmes autonomes. Cependant, malgré leur simplicité apparente (mécanismes plans, actionneurs linéaires), ces systèmes ont soulevé un grand nombre de problèmes difficiles. Nombre de ceux-ci ne sont d'ailleurs toujours pas résolus. Ainsi, alors que les robots manipulateurs se sont aujourd'hui généralisés dans l'industrie, rares sont les applications industrielles qui utilisent des robots mobiles. Si l'on a vu depuis peu apparaître quelques produits manufacturiers (chariots guidés) ou grand public (aspirateurs), l'industrialisation de ces systèmes bute sur divers problèmes délicats. Ceux-ci viennent essentiellement du fait que, contrairement aux robots manipulateurs prévus pour travailler exclusivement dans des espaces connus et de manière répétitive, les robots mobiles sont destinés à évoluer de manière autonome dans des environnements peu ou pas structurés.

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

Pour un robot mobile, la propulsion se fera le plus souvent par un système de roues, chenilles, etc..., susceptibles de glissement; il n'y aura plus de lien rigide entre les indications d'éventuels capteurs proprioceptifs et la position du robot. Il est donc indispensable pour un robot mobile de pouvoir se localiser par rapport à l'environnement par

des capteurs appropriés. Le rôle de ces capteurs est double : ils doivent assurer la perception de l'environnement et la localisation du robot dans cet environnement. La perception permet de détecter l'environnement proche ou éloigné du robot. Elle est nécessaire pour la sécurité du robot, pour la modélisation de l'environnement et pour sa localisation relativement à cet environnement. La localisation permet de se positionner par rapport à une position initiale ou par rapport à un environnement connu d'avance.

Nous allons dans ce présent chapitre présenter les différents types de robots mobiles et les différents capteurs utilisés en perception.

1.2 Robot mobile autonome

En robotique, on distingue deux principaux types : les robots manipulateurs et les robots mobiles. Les robots manipulateurs ont une base fixe contrairement aux robots mobiles qui peuvent se déplacer. Ainsi pour étudier les déplacements de ces robots, nous pouvons soit utiliser un repère allocentrique (absolu), soit un repère égocentrique (fixé sur le robot).

Bien souvent, quand on parle de robotique mobile, on sous entend robots mobiles à roues. Ce sont en effet les systèmes les plus étudiés, parce qu'ils sont plus simples à réaliser que les autres types de robots mobiles, ce qui permet d'en venir plus rapidement à l'étude de leur navigation. Ce type de robots est notamment très souvent utilisé pour l'étude des systèmes autonomes. Vient ensuite la robotique mobile à pattes, avec notamment la robotique humanoïde, mais également des robots avec un nombre de pattes plus élevé qui offrent de bonnes propriétés pour la locomotion en milieu difficile (milieux forestiers et agricoles). La stabilité des mouvements de ce type de robots est en particulier un thème de recherche important. Enfin il existe également de nombreux autres types de robots mobiles (robots marins, sous marins, drones volants, micro et nano robots), dont l'étude se fait généralement dans des thématiques spécifiques avec des problèmes particuliers à l'application visée.

Il existe deux principaux modes de fonctionnement pour un robot mobile : télé-opéré et autonome. En mode télé-opéré, une personne pilote le robot à distance. Elle donne ses ordres via une interface de commande (joystick, clavier/souris...), et ceux-ci sont envoyés au robot via un lien de communication (internet, satellite ...). Dans ce domaine, les efforts de recherche sont beaucoup portés sur les problèmes liés au réseau de télécommunication (retards dans le réseau de communication, problèmes de commande, pertes de données) et sur l'amélioration de la perception de l'environnement par l'opérateur (interfaces haptiques, retours d'efforts).

A l'inverse, en mode autonome le robot doit prendre ses propres décisions. Cela signifie qu'il doit être capable à la fois de percevoir correctement son environnement, mais également de savoir comment réagir en conséquence, suivant le niveau d'autonomie. C'est à lui de planifier son parcours et de déterminer avec quels mouvements il va atteindre son objectif. Les recherches dans ce domaine portent principalement d'une part sur la localisation du véhicule autonome et la cartographie de son environnement, d'autre part sur le contrôle de tels véhicules (structure de contrôle, stratégies de commande, planification).

Cette notion d'autonomie prise en exemple ci-dessus, que nous pourrions qualifier de décisionnelle, ne doit pas être confondue avec celle d'autonomie énergétique (capacité du robot à gérer efficacement son énergie, à la préserver, voir à se ravitailler), même si ces deux notions sont étroitement liées : idéalement une des préoccupations principales d'un robot mobile totalement autonome (du point de vue décisionnel), serait en effet de pouvoir gérer de lui-même ses réserves d'énergie.

1.3 Les différents types de terrain

Nous rencontrons principalement 3 types d'espaces de navigation : les terrains plats, les terrains accidentés et les espaces 3D. Les terrains plats sont généralement utilisés pour modéliser les milieux urbains et les intérieurs de bâtiments. Le robot évolue sur un plan 2D considéré sans pentes, et tout objet qui sort de cet espace 2D est considéré comme un obstacle (figure 1.1).

Cette représentation est la plus simple à étudier et la plus répandue pour les robots mobiles à roues. En première approche, elle permet de se concentrer sur les problèmes de contrôle et de navigation autonome du robot.



Figure 1.1 : Robucab

Les terrains accidentés (ou $2D^{1/2}$) correspondent généralement aux milieux en extérieur, comme les forêts, les champs en robotique agricole, ou encore les terrains rocheux (figure 1.2). La différence avec les terrains plats est la présence de pentes, de bosses et de creux sur le terrain d'évolution du robot. Cela interdit d'utiliser une métrique standard 2D et cela complique pour beaucoup la détection d'obstacles et la modélisation des déplacements du robot. De plus il devient également important de vérifier que le robot ne bascule pas quand il escalade une pente ou enjambe un obstacle. Le système de locomotion du robot doit dans ce cas être adapté à la topologie du terrain.

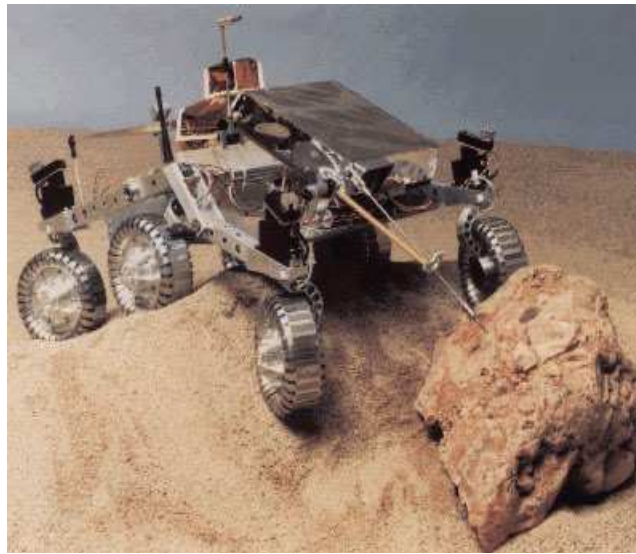


Figure 1.2 : Le robot martien Sojourner

Enfin les espaces d'évolution 3D sont par exemple utilisés pour modéliser la navigation des robots sous-marins (figure 1.3) et des drones volants (figure 1.4). Les problèmes rencontrés sont spécifiques à l'application visée.

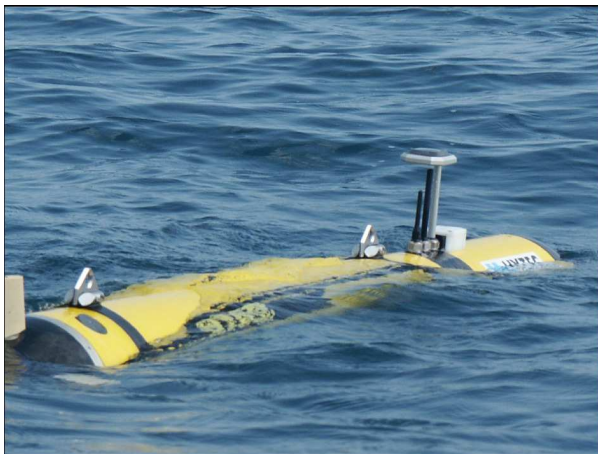


Figure 1.3: Robot sous marin TAIPAN



Figure 1.4: Drone de surveillance RESSAC

Chaque type de terrain correspond à des problématiques bien spécifiques. Le type de robot considéré dans notre travail est destiné à circuler en environnement urbain, la modélisation terrain plat sera utilisée. Cela signifie que l'on considère que tous les mouvements sont contenus dans un plan de navigation parallèle au sol.

1.4 Les systèmes de perception

La fonction perception consiste globalement à saisir un certain nombre d'informations sensorielles dans le but d'acquérir une connaissance et une compréhension du milieu d'évolution. Comme nous l'avons précisé précédemment, elle est le préalable indispensable aux étapes suivantes qui sont généralement pour un robot mobile les étapes de localisation et de mise à jour de carte de l'environnement (figure 1.5).

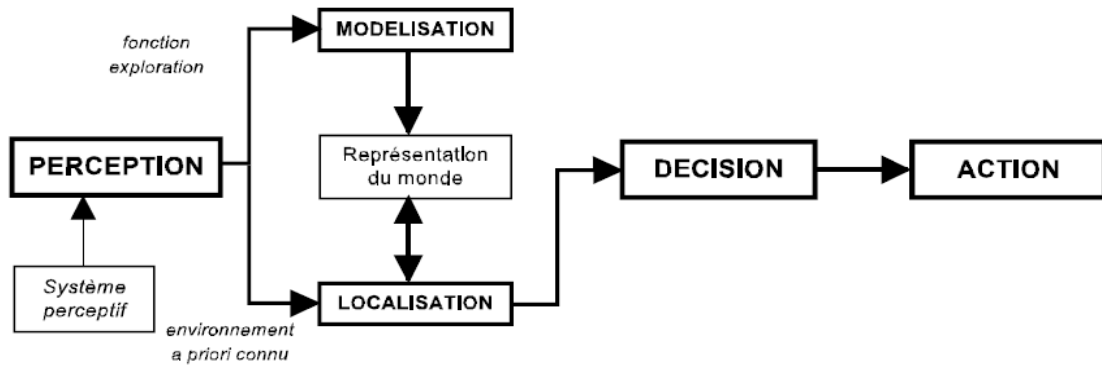


Figure 1.5 : Chaîne fonctionnelle d'un système de navigation.

Plusieurs constats peuvent être faits sur cet organe essentiel de la chaîne fonctionnelle de la navigation. Le premier consiste à noter que le choix d'un système de perception est souvent dépendant du milieu d'évolution du robot mobile ainsi que des fonctionnalités mises en œuvre sur le robot pour qu'il puisse remplir sa mission. Le second se révèle être générique et consiste à affirmer qu'un système de perception constitué d'un unique capteur sera rarement suffisant pour percevoir correctement l'environnement. Il en va de même pour l'être humain qui, pour appréhender son environnement, sollicite plusieurs organes perceptifs. Le système de perception d'un robot mobile intégrera le plus souvent plusieurs capteurs qui seront de types complémentaires pour un enrichissement des informations sensorielles, ou de types redondants pour répondre au problème de fonctionnement en mode dégradé. Dans ce cadre, des méthodes de fusion de données seront généralement employées pour conditionner ces informations sensorielles. Enfin le troisième constat qui pourra être dressé est celui du coût de l'intégration de capteurs sur le véhicule autonome. La précision désirée et une fréquence d'acquisition élevée seront autant de facteurs qui augmenteront le coût d'un

capteur. Il s'agit donc là d'une contrainte qui pèsera inévitablement sur le choix d'un système de perception. La classification des capteurs est généralement faite par rapport à deux familles : Les capteurs proprioceptifs qui fournissent des informations propres au comportement interne du robot, c'est-à-dire sur son état à un instant donné et les capteurs extéroceptifs qui fournissent des informations sur le monde extérieur au robot.

1.4.1 Les capteurs proprioceptifs

Les capteurs proprioceptifs fournissent, par intégration, des informations élémentaires sur les paramètres cinématiques du robot. Les informations sensorielles gérées dans ce cadre sont généralement des vitesses, des accélérations, des angles de giration, des angles d'attitude. On peut regrouper les capteurs proprioceptifs en deux familles [FRA90] :

- les capteurs de déplacement qui comprennent les odomètres, les accéléromètres, les radars Doppler, les mesureurs optiques. Cette catégorie permet de mesurer des déplacements élémentaires, des variations de vitesse ou d'accélération sur des trajectoires rectilignes ou curvilignes.
- les capteurs d'attitude, qui mesurent deux types de données : les angles de cap et les angles de roulis et de tangage. Ils sont principalement constitués par les gyroscopes et les gyromètres, les capteurs inertiels composites, les inclinomètres, les magnétomètres. Ces capteurs sont en majorité de type inertiels.

1.4.1.1 Les capteurs de déplacement

Les odomètres

Permettent de fournir une quantification des déplacements curvilignes du robot en mesurant la rotation de ses roues. Le calcul de la position relative du robot est réalisé par intégration des rotations élémentaires des roues. Les odomètres sont généralement composés de codeurs incrémentaux qui permettent de mesurer les angles de rotation avec une précision qui dépend de la résolution du codeur. L'information de déplacement nécessitera la connaissance du diamètre des roues, de l'entraxe des roues, de la structure mécanique et cinématique du véhicule. Ce capteur est fortement utilisé en robotique mobile puisqu'il présente l'avantage d'être simple à mettre en œuvre et surtout d'être peu coûteux.

Les accéléromètres

L'accéléromètre est un capteur qui mesure l'accélération linéaire en un point donné. En pratique, la mesure de l'accélération est effectuée à l'aide d'une masse d'épreuve M , de masse m , reliée à un boîtier du capteur. Le principe de ce capteur est de mesurer l'effort massique non gravitationnel qu'on doit appliquer à M pour le maintenir en place dans le

boîtier lorsqu'une accélération est appliquée au boîtier. Le calcul du déplacement élémentaire du robot est obtenu par double intégration de ces informations. Cette double intégration conduit généralement à des accumulations importantes d'erreurs. Ce capteur est plus coûteux que des odomètres.

Le radar Doppler

Le radar Doppler fournit une estimation instantanée de la vitesse linéaire d'une plateforme mobile par rapport à un objet de la scène en se basant sur l'effet Doppler-Fizeau. Le principe est de diriger un faisceau électromagnétique de fréquence F vers le sol. Le faisceau reçu après rediffusion sur le sol présente un décalage de fréquence ΔF proportionnel à la vitesse V du déplacement relatif du véhicule par rapport au sol. L'intégration de la mesure de vitesse fournira une estimation du déplacement du mobile. Ce capteur est le plus souvent utilisé en milieu extérieur. Il présente le gros avantage d'être insensible aux glissements des roues ou des chenilles du mobile. En revanche, sa précision se dégrade rapidement avec la diminution de la vitesse. De plus il est généralement lourd à mettre en œuvre, tant du point de vue électronique que du point de vue du traitement des informations reçues.

1.4.1.2 Les capteurs d'attitude

Les capteurs d'attitude permettent d'estimer les paramètres intrinsèques du robot qui sont les angles de cap, de roulis et de tangage. Ces capteurs sont principalement de type inertiel. Ces capteurs ont pour point commun d'être généralement coûteux et sensibles au bruit, d'où une intégration moins fréquente dans les systèmes embarqués que les odomètres.

Gyroscope, gyromètre et gyrocompas

Les gyroscopes permettent de mesurer une variation angulaire. Ils sont intéressants en robotique mobile parce qu'ils peuvent compenser les défauts des odomètres. Une erreur d'orientation odométrique peut entraîner une erreur de position cumulative qui peut être diminuée voire compensée par l'utilisation conjointe de gyroscopes [BOR96]. Les gyroscopes très précis sont trop onéreux pour être utilisés en robotique mobile. Cependant, les gyroscopes à fibre optique, connu pour leur grande précision, ont vu leur prix chuter et sont donc devenus une solution attractive pour la navigation en robotique mobile.

Le gyromètre est un capteur qui permet de mesurer une vitesse angulaire. Il existe plusieurs types de gyromètres : les premiers à avoir fait leur apparition furent mécaniques, aujourd'hui, on utilise surtout des gyromètres laser ou des gyromètres optiques.

Le gyrocompas est un capteur qui permet de mesurer le cap. Il est composé d'un gyroscope et d'un compas magnétique. Le gyrocompas conserve le nord magnétique durant tout le déplacement du véhicule, après l'avoir initialement déterminé de façon autonome.



Figure 1.6 : Gyroscope mécanique de Foucault et gyromètre Futaba à structure vibrante

Le magnétomètre ou compas magnétique

Le magnétomètre qui est aussi appelé compas magnétique mesure la direction du champ magnétique terrestre pour déduire l'orientation du robot. Toutefois, les mesures sont perturbées par l'environnement magnétique du robot (comme par exemple les lignes d'énergie ou les structures en acier). Ceci rend difficile l'utilisation de ce capteur en milieu intérieur.

1.4.2 Les capteurs extéroceptifs

Les capteurs extéroceptifs permettent de percevoir le milieu d'évolution du robot. Ils sont généralement le complément indispensable aux capteurs présentés précédemment. Des méthodes de fusion de données seront alors utilisées pour conditionner et traiter les informations sensorielles de natures différentes. Deux familles de capteurs extéroceptifs embarqués peuvent être identifiées : les capteurs télémétriques et les systèmes de vision.

1.4.2.1 Les capteurs télémétriques

Les capteurs à ultrasons

Les capteurs à ultrasons utilisent l'air comme milieu de propagation. La méthode de mesure consiste à exciter une céramique piézo-électrique à l'aide de quelques impulsions de fréquence égale à la fréquence de la pastille. Connaissant la vitesse de propagation de l'onde acoustique dans un milieu donné, la mesure du temps de vol de l'onde permet d'obtenir la distance d'un objet par rapport au capteur.

Les capteurs ultrasonores les plus couramment utilisés sont les capteurs Polaroid qui émettent une onde ultrasonore à 44 kHz et qui permettent de mesurer des distances comprises entre

trente centimètres et dix mètres. Les avantages majeurs de ces capteurs sont leur faible prix de revient et leur simplicité d'implantation. En revanche de nombreux inconvénients induisent leur utilisation de moins en moins fréquente en robotique mobile :

- Une très faible directivité qui est liée au cône d'émission de l'onde dont l'angle d'ouverture est important. Ceci implique une erreur généralement importante qui entache la mesure (figure 1.7).

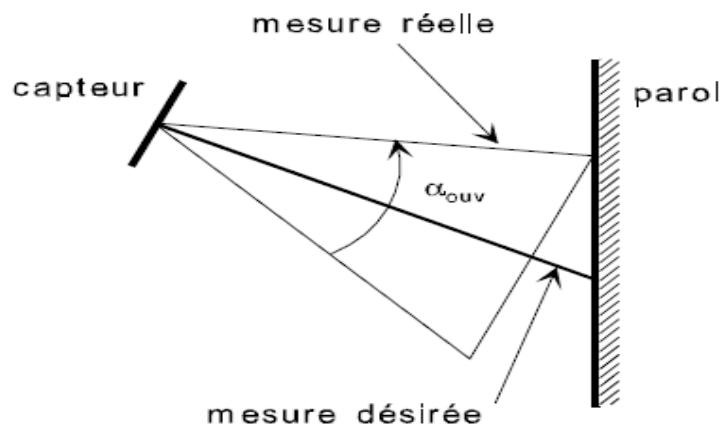


Figure 1.7 : Cône d'émission d'un faisceau ultrasonore

- Un angle d'incidence relativement faible qui n'excède généralement pas 30 à 40° suivant le matériau de la paroi.
- Une forte sensibilité aux conditions d'utilisation, telles que la température, l'humidité et les échos parasites. Ces paramètres influent directement sur la précision de mesure.
- Une forte influence aux problèmes de réflexions multiples. Ce phénomène se produit lorsque l'onde ultrasonore heurte plusieurs parois avant de revenir sur le capteur.



Figure 1.8 : Télémètres ultrasonores Polaroid USP 3 et Migatron RPS 409 IS



Figure 1.9 : Télémètre ultrasonore MSU08 (commercialisé par Lextronic)

Les capteurs laser

Le capteur laser est basé sur l'émission d'une onde électromagnétique structurée. Ce capteur permet donc d'obtenir un faisceau d'ondes très concentré, contrairement aux capteurs ultrasonores. Les systèmes laser possèdent de nombreux avantages qui en font des capteurs souvent utilisés dans les applications de robotique mobile :

- la résolution angulaire : des faisceaux angulairement très fins peuvent être obtenus avec des lentilles d'émission de petite taille, du fait de la courte longueur d'onde émise.
- la réponse optique des cibles : les longueurs d'ondes courtes permettent d'obtenir un écho pour des angles d'incidence du faisceau sur la cible allant jusqu'à 75° et améliore de ce fait la probabilité de détection.
- la précision sur la mesure de distance par rapport à la portée, pour les télémètres, est importante. Une précision de l'ordre du dixième de millimètre peut être obtenue sur des distances de l'ordre de 30 mètres.

L'inconvénient majeur du laser se situe au niveau de l'électronique de mesure qui doit être capable, étant donnée la vitesse de propagation des ondes électromagnétiques, de mesurer des intervalles de l'ordre de la pS pour des précisions de distance de l'ordre du millimètre. Ceci implique un matériel coûteux et complexe. En robotique les systèmes laser sont utilisés pour des mesures de distance, d'angles et pour la modélisation 3D de l'environnement. La source laser sera dans ce cas toujours associée à une caméra et fournira des informations sur la profondeur.



Figure 1.10 : Famille des télémètres lasers Sick [SICK04]

1.4.2.2 Les systèmes de vision

Les systèmes de vision en robotique sont basés sur l'utilisation d'une caméra CCD. L'arrivée des capteurs CCD (Charge Coupled Device) en 1975 a été déterminante dans l'évolution de la vision : la rapidité d'acquisition, la robustesse et la miniaturisation sont autant d'avantages qui ont facilité leur intégration. Les systèmes de vision sont très performants en termes de portée, précision et quantité d'informations exploitables. Ils sont de plus les seuls capables de restituer une image sensorielle de l'environnement la plus proche de celle perçue par l'être humain. En revanche, l'inconvénient majeur de tels systèmes de perception se situe au niveau de la gestion du flux important de données exploitables : traiter une image demeure une opération délicate et surtout coûteuse en temps de calcul.

Utilisé seul, une caméra CCD ne peut fournir qu'une information 2D. Les techniques qui vont permettre d'obtenir des informations 3D à partir d'un tel capteur sont généralement liées à l'adjonction d'un autre capteur. Dans ce cadre nous pouvons identifier les techniques suivantes : La stéréovision et Les techniques de lumières structurées

On peut ajouter aux deux catégories précédentes celle consistant à associer des systèmes de réflexion de type miroir au capteur CCD : il s'agit des techniques de vision omnidirectionnelle. Précisons que cette dernière sera bien sûr utilisée pour obtenir un modèle sensoriel panoramique de l'environnement.

La stéréovision

La stéréovision consiste à observer une même scène avec deux caméras qui sont éloignées l'une de l'autre et dont on connaît la distance qui les séparent. Connaissant la géométrie exacte du système stéréoscopique, la première étape de reconstruction 3D consiste à mettre en correspondance les deux images. Cette phase réside dans la détermination de couples de points observés dans les deux images, ou dans l'appariement de points d'intérêt. L'information 3D pourra alors être fournie par triangulation. La stéréovision est basée sur le même principe de reconstitution de la profondeur que la vision chez l'être humain.

La figure 1.11 décrit plus précisément le principe de la stéréovision. Tout point M visible depuis les deux caméras se projette en $P1$ sur le plan image de la première caméra et en $P2$ sur le plan image de la deuxième caméra. Inversement, connaissant deux points en correspondance $P1$ et $P2$, ainsi que la distance D séparant les centres optiques $O1$ et $O2$ des deux caméras, les coordonnées du point P sont données par l'intersection de $(O1p1)$ et $(O2p2)$.

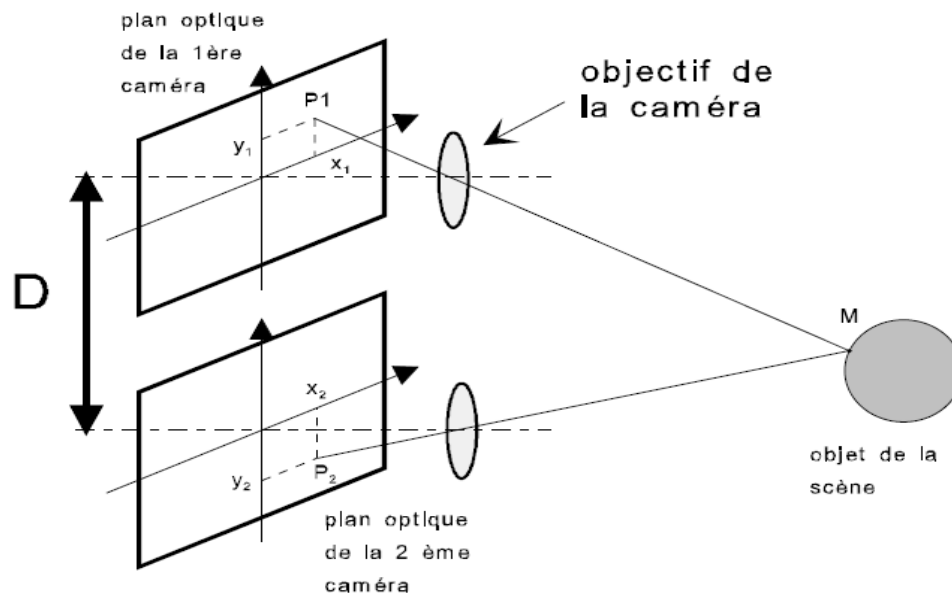


Figure 1.11 : Principe de la stéréovision

La stéréovision apparaît comme un des moyens de perception les plus performants en robotique. Toute la problématique de la stéréovision réside dans la robustesse de la phase de mise en correspondance des informations : elle est souvent liée à de nombreuses ambiguïtés mais aussi à des temps de calcul très importants (aspect fortement combinatoire de l'appariement).

Les capteurs à triangulation active

L'obtention de la 3D est gérée comme pour la stéréovision, mais on remplace une caméra par un laser qui sera chargé de projeter un motif de lumière structurée. Le principe est d'éclairer l'environnement avec une lumière cohérente et de l'observer avec une caméra sous une certaine parallaxe. Connaissant la position de la source lumineuse et celle de la caméra, l'information peut être obtenue par triangulation.

Ces capteurs sont utilisés pour obtenir des images de profondeur, mais également pour l'évitement d'obstacles. Plusieurs patrons de lumière structurée peuvent être utilisés :

- *Capteur à simple triangulation* : les capteurs à simple triangulation acquièrent la distance d'un seul point lumineux à la fois.

- *Les capteurs par plan de lumière* : la source de lumière projetée n'est plus un point, mais un plan de lumière.
- *Capteurs à plusieurs plans de lumière* : il s'agit dans ce cas de projeter un motif tel que des lignes ou de petits disques sur la scène observée (figure 1.12)

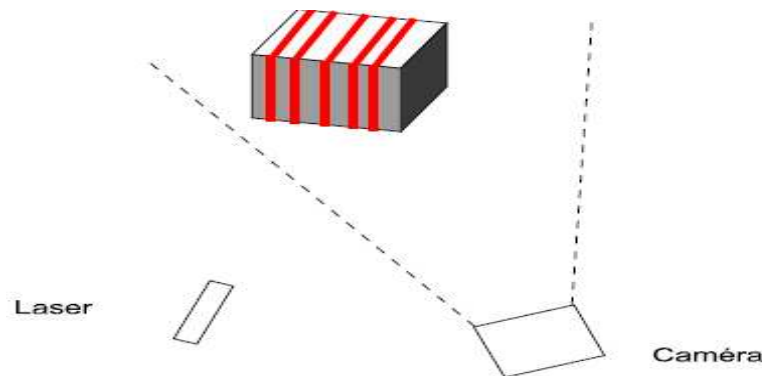


Figure 1.12 : Triangulation par plusieurs plans laser

La technique de triangulation active est plus robuste que la stéréovision car elle permet de faire l'économie de l'étape de mise en correspondance. En revanche la précision n'est pas uniforme sur l'ensemble de l'image : plus un objet est éloigné, plus l'erreur de mesure est importante [FOF97].

Les systèmes de vision omnidirectionnelle

Ces systèmes de perception associent une caméra CCD et un élément permettant d'obtenir une vue sur 360 degrés de l'environnement. A ce titre ils sont de plus en plus utilisés en robotique mobile au détriment de la vision monoculaire. Suivant l'élément ou les éléments ajoutés à une caméra CCD, on pourra distinguer quatre techniques pour obtenir une image omnidirectionnelle.

- ***Génération d'images multiples par utilisation de plusieurs caméras***

Cette première technique consiste à utiliser plusieurs caméras couvrant un champ de vision égal à 360 degrés : par exemple quatre caméras séparées par des angles de 90 degrés.

Dans ce cadre, nous pouvons citer le système développé au laboratoire de Bell. AT & T qui intègre quatre caméras conventionnelles et quatre miroirs triangulaires [NAL96]. On dispose donc de quatre images pour pouvoir calculer une image panoramique. Le résultat est une image panoramique à haute résolution (3800 par 480 pixels). Ce capteur, possédant un seul point de vue effectif, permet de calculer n'importe quelle projection planaire pour obtenir des images purement perspectives.

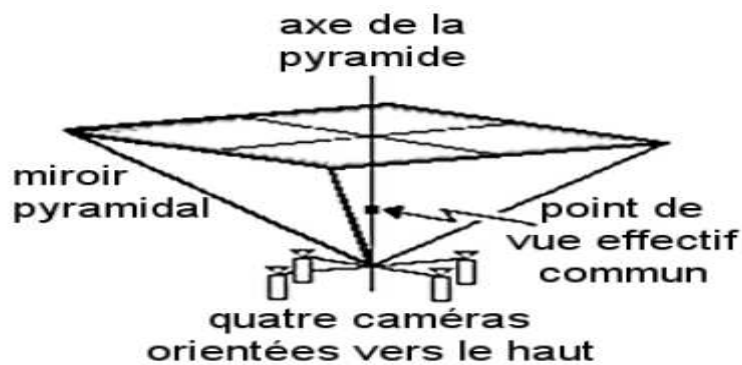


Figure 1.13 : Système Full view des laboratoires Bell

- **Génération d'images multiples par rotation d'une caméra**

Cette deuxième technique consiste à faire pivoter une caméra autour d'un axe [SAR89], [ISH92]. Cette méthode permet d'avoir une prise de l'environnement avec une seule caméra, mais il faut effectuer une rotation complète avant d'obtenir une vue sur 360 degrés.

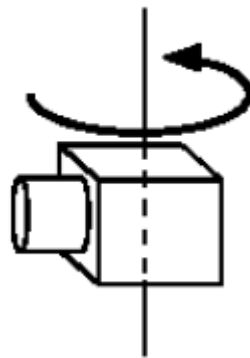


Figure 1.14 : Caméra rotative.

De tels systèmes génèrent un volume de données qui nécessite des temps de traitement très importants. En outre, les aspects mécaniques, la gestion du flou dû au mouvement de la caméra font que ces méthodes sont complexes à mettre en œuvre. Toutefois, quelle que soit la solution envisagée, l'acquisition d'une scène panoramique avec une caméra matricielle ou linéaire reste une méthode lourde à mettre en œuvre.

- **Utilisation de lentille spéciale**

Cette troisième méthode consiste à ajouter un ensemble optique sur la caméra CCD afin de dévier les rayons lumineux, comme par exemple l'oeil de poisson (fish-eye) [CAO86].

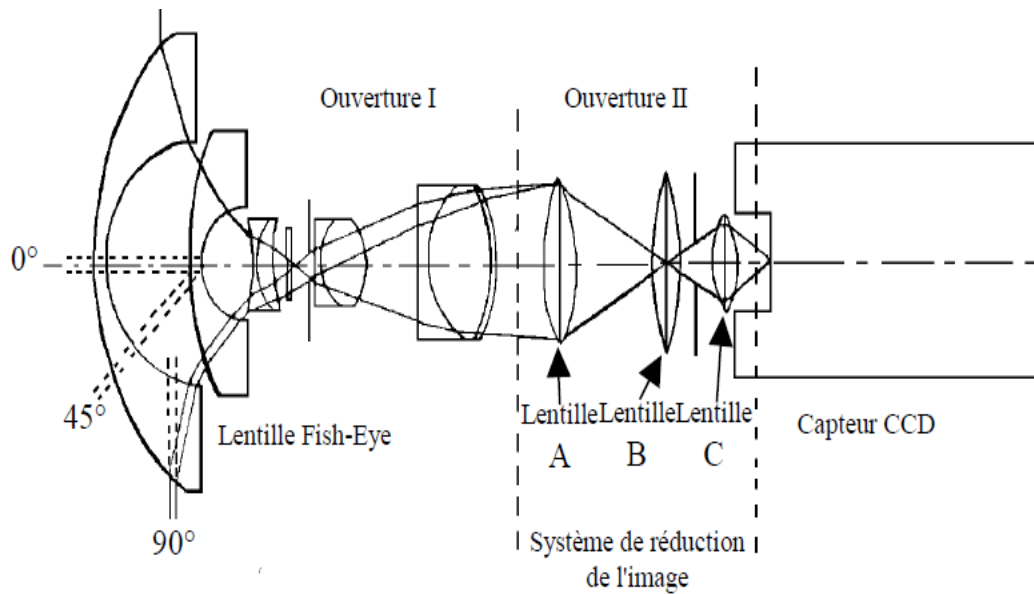


Figure 1.15 : Schéma du système optique du capteur omnidirectionnel utilisé par Cao

Une caméra Fish-eye, qui utilise un objectif Fish-eye au lieu d'un objectif conventionnel, peut acquérir une vue hémisphérique en temps réel. Morita a proposé une méthode stéréoscopique de l'analyse du mouvement basée sur cette technique [MOR89]. Les chercheurs de l'université de Cincinnati ont appliqué une caméra Fish-eye au contrôle de position d'un robot mobile en utilisant des cibles [CAO86], [RON86] et en suivant des lignes [ELK94], [MAT95b]. L'analyse de ces images est difficile car les éléments utiles se situent généralement sur la périphérie, là où la résolution est faible. De plus, il est difficile de générer une image complètement perspective car les objectifs Fish-eye n'ont pas un centre de projection unique. L'image obtenue comporte beaucoup de distorsion et des méthodes de calibration pour de tels systèmes optiques sont nécessaires.

- **Utilisation de miroirs convexes**

La dernière méthode consiste à placer un miroir réfléchissant face à la caméra pour obtenir une image qui est la projection omnidirectionnelle 2D de l'environnement. Ces capteurs omnidirectionnels sont appelés capteurs catadioptriques. Ils présentent l'avantage de fournir une vue globale de l'environnement en une seule acquisition. Autre avantage : ils n'ont aucune partie mobile et donc consomment peu d'énergie (pas de mécanisme de rotation et d'orientation). Par contre, le redressement d'image pose problème. Rappelons que la catadioptrie est la théorie des systèmes optiques comportant une succession d'éléments réfléchissants et réfractants (dioptries : lentilles...). Les miroirs utilisés pour la vision omnidirectionnelle sont de plusieurs types : conique, sphérique, hyperboloïdal et parabololoïde.

1.5 Les méthodes de localisation

Nous avons précédemment positionné le problème de localisation comme étant un préalable incontournable de celle de la génération de carte incrémentale de l'environnement. La suite logique de la synthèse faite sur les capteurs portera donc sur les différentes méthodes de localisation. Par analogie avec la classification des capteurs, nous pourrions distinguer deux types de méthodes de localisation :

- Les méthodes de localisation relatives, basées sur l'utilisation des capteurs proprioceptifs.
- Les méthodes de localisation absolues, basées sur l'utilisation de capteurs extéroceptifs.
- Les méthodes dites « hybrides » qui sont basées sur l'utilisation conjointe des deux types de capteur.

1.5.1 Localisation à l'estime

La navigation à l'estime, ou navigation relative, consiste à évaluer la position, l'orientation et éventuellement la vitesse du robot mobile par intégration des informations fournies par des capteurs dits proprioceptifs. L'intégration se fait par rapport au point de départ du robot. Ces données peuvent être des informations de déplacement (odomètre), de vitesse (vélocimétrie) ou d'accélération (accéléromètre). Ces systèmes permettent d'obtenir un flux relativement important au niveau de l'obtention des estimations de position. Cette caractéristique intéressante a favorisé l'utilisation de ces systèmes de localisation en robotique mobile, ainsi que dans des secteurs de pointe tels que les domaines spatial et militaire : fusées, missiles, avions, sous-marins...

L'odométrie permet de déterminer la position et le cap (x, y, θ) d'un véhicule par intégration de ses déplacements élémentaires, et ce, par rapport à un repère lié à sa configuration initiale. L'algorithme de localisation est basé sur le comptage des impulsions générées par des codeurs durant une période d'échantillonnage connue. Connaissant le rayon des roues et la distance entre chaque roue, il est possible d'exprimer de manière récursive la position et le cap du robot. Les avantages de l'odométrie résident dans sa simplicité de mise en œuvre et dans son faible coût : ces caractéristiques en font un système de localisation couramment utilisé en robotique mobile. En outre, l'odométrie permet d'obtenir une estimation de la position et du cap à une cadence relativement élevée. L'inconvénient est une précision très médiocre sur des distances importantes, à cause des erreurs cumulatives. Une étude intéressante d'intégration poussée de l'odométrie a été réalisée par Borenstein dans [BOR95]. Il y propose une méthode originale IPEC (Internal Position Error Correction),

permettant d'obtenir une estimation odométrique fiable sans avoir recours à un système de localisation absolue ou à un système de navigation inertielle.

La vélocimétrie est une méthode qui consiste à mesurer directement la vitesse du véhicule et à l'intégrer pour obtenir le déplacement. La vitesse de rotation instantanée, dans la majorité des cas, est obtenue avec des gyromètres. La vitesse linéaire est quant à elle généralement obtenue par l'utilisation d'un radar Doppler dirigé vers le sol. Dans ce cadre nous pouvons citer ici les travaux de H. Makela [MAK91] qui localise à l'estime un véhicule circulant sur des chemins forestiers à l'aide d'un radar Doppler.

Ce type de navigation utilise les accéléromètres, les gyroscopes et les compas magnétiques. La localisation relative d'un robot est déterminée à partir d'informations inertielles acquises au cours de son mouvement. Les méthodes de localisation inertielles utilisent les informations d'accélération suivant des axes prédéterminés ou alors l'effet gyroscopique par rapport à des axes de rotations prédéfinis. Le calcul de la position est effectué par double intégration de ces informations. Le robot VEGA du LIRMM utilise pour se localiser un odomètre et un système inertiel composé de six capteurs : un accéléromètre trois axes, quatre gyromètres piézo-électriques et un gyromètre à fibre optique [VAG93].

1.5.2 Localisation absolue

La localisation absolue est une technique qui permet à un robot de se repérer directement dans son milieu d'évolution, que ce soit en environnement extérieur (mer, espace, terre), ou en environnement intérieur (ateliers, immeubles, centrales nucléaires...). Ces méthodes de localisation sont basées sur l'utilisation de capteurs extéroceptifs.

Pour répondre à la problématique qu'est la localisation d'un robot dans son environnement, deux types de stratégies sont utilisables :

- La première consiste à utiliser des points de repère naturels.
- La deuxième à utiliser des points de repère artificiels.

1.5.2.1 Repères artificiels

Les repères artificiels sont des balises caractéristiques qui sont ajoutées au milieu d'évolution du robot et dont les positions sont connues. L'inconvénient de ce type de techniques réside essentiellement dans son manque de souplesse et dans sa lourdeur d'utilisation. En effet tout changement de configuration de l'environnement impliquera une remise en cause du réseau de balises. En revanche cette technique a le gros avantage d'être précise, robuste et surtout de satisfaire la contrainte temps réel.

Les balises artificielles peuvent être de deux types :

- *Actives* : elles émettent des signaux.
- *Passives* : elles ne peuvent pas émettre.

Balises actives et GPS

Les balises actives sont généralement interactives avec le système de perception de l'engin mobile puisque ce sont elles 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 (infrarouge par exemple) et les antennes émettrices hyperfréquences. Les systèmes à balises actives sont le moyen de navigation le plus communément utilisé pour les bateaux et les avions.

Un système de localisation absolue basé sur le repérage de balises actives infrarouges par un système de vision monoculaire a été développé et implanté. Les balises artificielles sont constituées par trois diodes électroluminescentes émettant des impulsions infrarouges codées (figure 1.16). La méthode de localisation consiste à extraire à partir de l'image d'une balise les distances des trois balises par rapport au robot et pour ensuite effectuer une triangulation.

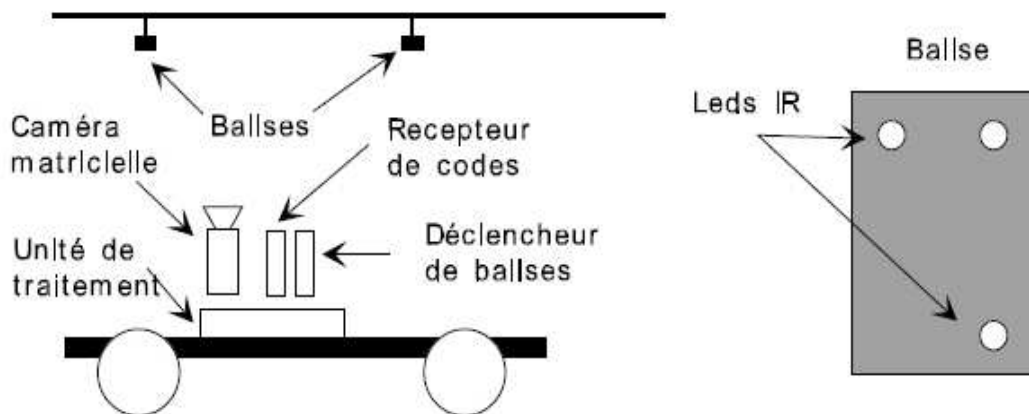


Figure 1.16 : Système de localisation par repérage de balises actives

Le système GPS est finalement un système de localisation en environnement extérieur basé sur le repérage de balises actives que sont les satellites. Ce système, développé par le département de la défense des Etats-Unis, est composé de vingt-quatre satellites. En utilisant des méthodes avancées de trilatération, le récepteur terrestre peut calculer une position en mesurant le temps de vol des ondes renvoyées par les satellites. En théorie, trois satellites peuvent permettre de calculer la latitude, la longitude et l'altitude du récepteur.

Balises passives

On appelle balises passives des repères artificiels n'échangeant pas de signaux avec le système de perception. Comme précédemment ce type de méthode est basé sur le repérage de balises artificielles placées à des positions connues dans le milieu d'évolution du robot. Ces systèmes ont l'avantage d'être moins lourds à installer que les systèmes précédents.

Les véhicules mobiles industriels AGV (Autonomous Guided Vehicle), développés par la société GEC-Caterpillar utilisent un système de navigation goniométrique constitué par un laser tournant et des balises passives rétro réfléchissantes [HOL91]. La position du chariot est ensuite déterminée par triangulation.

Armingol [ARM98] utilise comme balise artificielle des ronds noirs collés sur les murs d'un environnement intérieur. Ces ronds sont détectés à l'aide d'une caméra CCD orientable.

Kortenkamp utilise des cylindres avec différents motifs disposés dans l'environnement pour localiser le robot Carmel (figure 1.17).

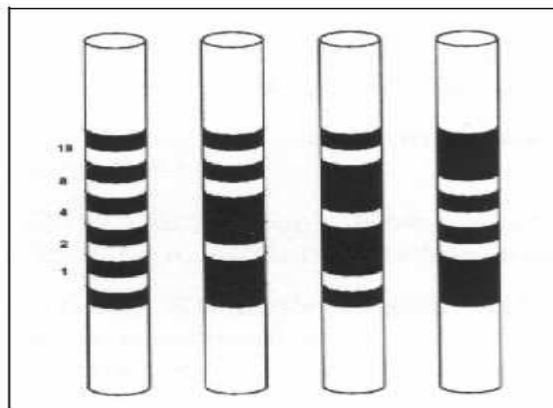


Figure 1.17 : Balises passives utilisées par Kortenkamp

Il est à noter que le point commun de toutes ces méthodes est le calcul de la configuration du robot basée systématiquement sur une des deux méthodes :

- La trilatération (triangulation par télémétrie)
- La triangulation (triangulation par goniométrie)

Enfin, la précision obtenue sur la localisation et l'adéquation avec la contrainte temps réel font que cette technique est souvent intégrée pour des systèmes de convoyage industriels.

1.5.2.2 Repères naturels

Cette technique consiste à utiliser les éléments caractéristiques de l'environnement pour estimer la position du robot. L'intérêt de ces méthodes est leur souplesse d'utilisation puisqu'elles ne nécessitent pas d'aménager le milieu d'évolution du robot. Pour la

problématique localisation, une connaissance de l'environnement sera nécessaire. Il s'agira d'une représentation cartographique qui intégrera la position des amers qui serviront à localiser le robot. La fonction localisation peut se décliner en deux étapes :

- La construction du modèle sensoriel : elle consiste à conditionner les mesures capteur.
- La mise en correspondance de ce modèle avec la carte de l'environnement : elle consiste à appairer les observations avec des primitives cartographiques. Il s'agit là de l'étape prépondérante permettant d'estimer la configuration du robot.

1.5.3 Méthodes hybrides

Les chercheurs se sont rendus compte de l'intérêt à utiliser conjointement la localisation à l'estime et la localisation absolue. On parle alors de recalage dynamique. Mais les méthodes hybrides intègrent aussi celles qui s'appuient sur la coopération de plusieurs capteurs proprioceptifs ou extéroceptifs.

1.6 Conclusion

Pour permettre à un robot mobile de percevoir l'environnement dans lequel il évolue, divers capteurs de perception existent. Le tour d'horizon des capteurs de perception que nous avons fait dans ce chapitre nous a permis d'avoir une idée sur la grande diversité de leur principe et de leurs performances. Par rapport aux systèmes de perception utilisés en robotique mobile, deux familles de capteurs peuvent être utilisées. Nous avons pu constater que celles-ci sont plus complémentaires que concurrentes. C'est pour cette raison qu'un système de navigation sera généralement basé sur l'exploitation des informations sensorielles émanant de ces deux catégories de capteurs. Nous pouvons étendre cette remarque à la problématique spécifique qu'est la perception du milieu d'évolution du robot : l'emploi d'un unique capteur extéroceptif pour cette mission sera généralement insuffisant. Ainsi, l'association de plusieurs capteurs extéroceptifs, qu'ils soient complémentaires ou redondants, permettra d'obtenir un modèle sensoriel robuste et hautement descriptif. Enfin parmi les capteurs extéroceptifs à privilégier, les systèmes de vision omnidirectionnelle semblent être particulièrement adaptés pour le problème de la navigation en robotique mobile. Un système de perception embarqué efficace et robuste sera généralement fondé sur la coopération de plusieurs capteurs. Des remarques similaires peuvent être amenées sur les méthodes de localisation. Les deux grandes familles de méthodes, relative et absolue, possèdent chacune des avantages et des inconvénients.

CHAPITRE 2

MODELISATION DES ROBOTS MOBILES

2.1 Définitions

On note $R = (O, \bar{x}, \bar{y}, \bar{z})$ un repère fixe quelconque, dont l'axe \bar{z} est vertical et $R' = (O', \bar{x}', \bar{y}', \bar{z}')$ un repère mobile lié au robot. On choisit généralement pour O' un point remarquable de la plate-forme, typiquement le centre de l'axe des roues motrices s'il existe, comme illustre la figure 2.1.

Par analogie avec la manipulation, on appelle situation [FOU99] ou souvent posture [CAM96] du robot le vecteur :

$$\xi = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (2.1)$$

Où x et y sont respectivement l'abscisse et l'ordonnée du point O' dans R et θ l'angle (\bar{x}, \bar{x}') .

La situation du robot est donc définie sur un espace M de dimension $m = 3$, comparable à l'espace opérationnel d'un manipulateur plan.

La configuration d'un système mécanique est connue quand la position de tous ses points dans un repère donné est connue. Alors que pour un bras manipulateur cette notion est définie sans ambiguïté par les positions angulaires des différentes articulations, on peut, dans le cas d'un robot mobile, donner une vision plus ou moins fine de la configuration, comme on le verra par la suite. Dans tous les cas, on définira la configuration du robot mobile par un vecteur :

$$q = \begin{pmatrix} q_1 \\ q_2 \\ \cdot \\ \cdot \\ q_n \end{pmatrix} \quad (2.2)$$

de n coordonnées appelées coordonnées généralisées. La configuration est ainsi définie sur un espace N de dimension n , appelée l'espace des configurations.

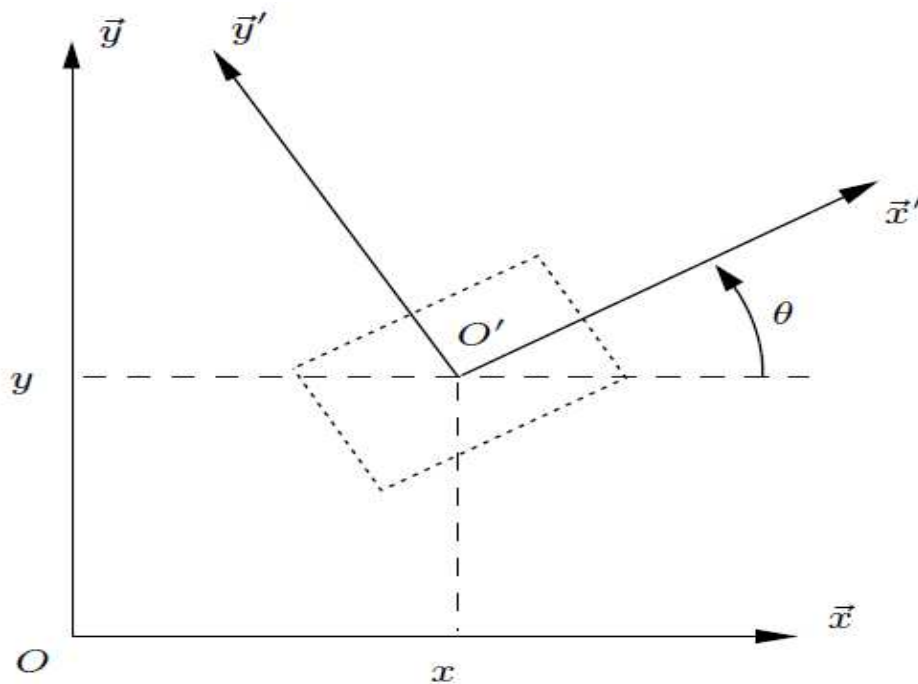


Figure 2.1 : Repérage d'un robot mobile

2.2 Roulement sans glissement et contraintes non holonomes

2.2.1 Roulement sans glissement

La locomotion à l'aide des roues exploite la friction au contact entre roue et sol. Pour cela, la nature du contact (régularité, matériaux en contact) a une forte influence sur les propriétés du mouvement relatif de la roue par rapport au sol. Dans de bonnes conditions, il y a roulement sans glissement (r.s.g.) de la roue sur le sol, c'est-à-dire que la vitesse relative de la roue par rapport au sol au point de contact est nulle. Théoriquement, pour vérifier cette condition, il faut réunir les hypothèses suivantes :

- le contact entre la roue et le sol est ponctuel.
- les roues, de rayon r , sont indéformables.

En pratique le contact se fait sur une surface, ce qui engendre bien évidemment de légers glissements. De même, alors qu'il est raisonnable de dire que des roues pleines sont indéformables, cette hypothèse est largement fautive avec des roues équipées de pneus.

Malgré cela, on supposera toujours qu'il y a roulement sans glissement et, par ailleurs, que le sol est parfaitement plan.

Mathématiquement, on peut traduire la condition de r.s.g. sur une roue. Soit P le centre de la roue, Q le point de contact de la roue avec le sol, φ l'angle de rotation propre de la roue et θ l'angle entre le plan de la roue et le plan (O, \vec{x}, \vec{y}) comme indiqué à la figure 2.2.

La nullité de la vitesse relative \vec{v}_Q roue/sol au point de contact permet d'obtenir une relation vectorielle entre la vitesse \vec{u}_P du centre P de la roue et le vecteur vitesse de rotation $\vec{\omega}$ de la roue :

$$\vec{v}_Q = \vec{u}_P + \vec{\omega} \wedge \overline{PQ} = \vec{0} \quad (2.3)$$

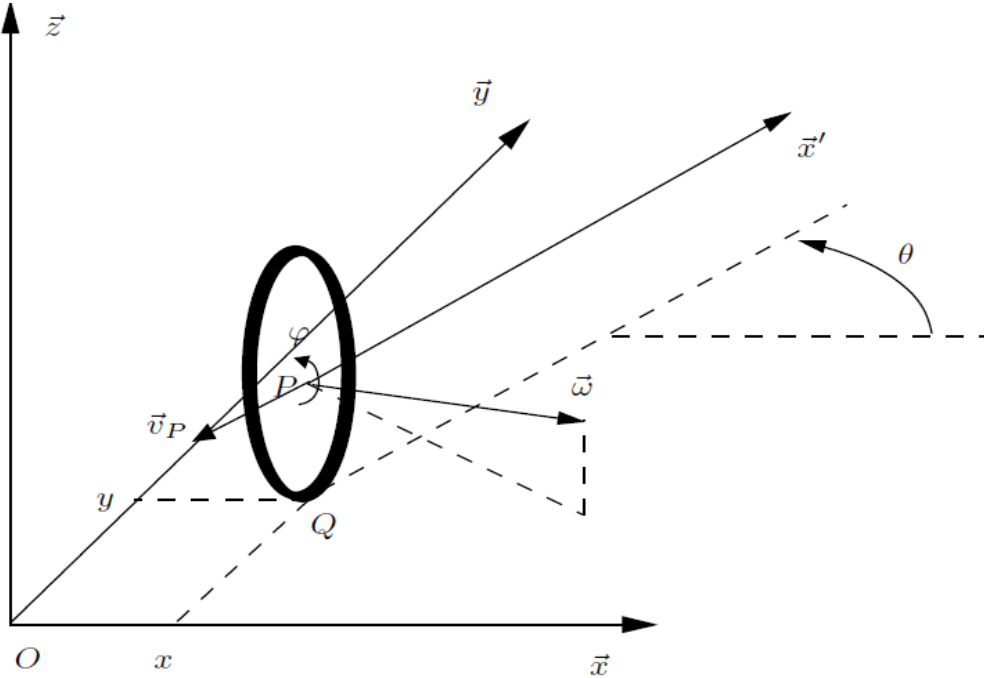


Figure 2.2 : Caractérisation du roulement sans glissement

Les points P et Q ont pour coordonnées respectives $(x, y, r)^T$ et $(x, y, 0)^T$. Il vient alors :

$$\begin{aligned} \dot{x}\vec{x} + \dot{y}\vec{y} + (\dot{\theta}\vec{z} + \dot{\phi}(\sin\theta\vec{x} - \cos\theta\vec{y})) \wedge (-r\vec{z}) &= \vec{0} \\ (\dot{x} + r\dot{\phi}\cos\theta)\vec{x} + (\dot{y} + r\dot{\phi}\sin\theta)\vec{y} &= \vec{0} \end{aligned} \quad (2.4)$$

Ceci nous donne le système de contraintes scalaires :

$$\dot{x} + r\dot{\phi}\cos\theta = 0 \quad (2.5)$$

$$\dot{y} + r\dot{\phi}\sin\theta = 0 \quad (2.6)$$

que l'on peut transformer pour faire apparaître les composantes de vitesse dans le plan de la roue d'une part et perpendiculairement à la roue d'autre part :

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \quad (2.7)$$

$$\dot{x}\cos\theta + \dot{y}\sin\theta = -r\dot{\phi} \quad (2.8)$$

2.2.2 Contraintes non holonomes

Les équations précédentes, caractérisant le roulement sans glissement d'une roue sur le sol, sont des contraintes non holonomes. Nous nous proposons dans ce paragraphe de préciser ce que recouvre ce terme et de caractériser les systèmes non holonomes.

Soit un système de configuration q soumis à des contraintes indépendantes sur les vitesses, regroupées sous la forme $A^T(q)\dot{q}=0$. S'il n'est pas possible d'intégrer l'une de ces contraintes, elle est dite non intégrable ou non holonome. De manière concrète l'existence de contraintes non holonomes implique que le système ne peut pas effectuer certains mouvements instantanément. Par exemple, dans le cas de la roue, il ne peut y avoir de translation instantanée parallèlement à l'axe de la roue. Un tel déplacement nécessitera des manœuvres. De même, comme on le sait bien, une voiture ne peut se garer facilement sans effectuer de créneaux

Il n'est pas évident de dire à priori si une contrainte est intégrable ou non. Pour cela, on a recours à l'application du théorème de Frobenius, dont une version complète pourra être trouvée dans un ouvrage de référence de géométrie différentielle [WAR83] ou de commande non linéaire. Seule la connaissance du crochet de Lie est nécessaire à notre étude. Pour deux vecteurs $b_i(q)$ et $b_j(q)$, cet opérateur est défini par :

$$[b_i(q), b_j(q)] = \frac{\partial b_j}{\partial q} b_i - \frac{\partial b_i}{\partial q} b_j \quad (2.9)$$

Théorème : Soit un système de configuration q , de dimension n , soumis à un ensemble de contraintes indépendantes s'écrivant sous la forme $A^T(q)\dot{q}=0$. Soit : $B(q)=(b_1(q) \ b_2(q) \ \dots \ b_m(q))$ une matrice de rang plein m , orthogonale à $A(q)$ sur tout l'espace des configurations. Soit enfin l'algèbre de **Lie** de dimension p , avec $m \leq p \leq n$ engendrée par l'ensemble des colonnes de $B(q)$, auxquelles s'ajoutent les crochets de **Lie** successifs formés à partir de ces colonnes, à condition qu'ils augmentent la dimension de l'algèbre. Alors, parmi les contraintes auxquelles est soumis le système, $n-p$ sont intégrables.

Prenons l'exemple de la roue dont on a écrit le modèle précédemment. Sa configuration est entièrement définie par sa configuration $q=(x, y, \theta, \varphi)^T$. D'après (2.7) et (2.8) on peut déterminer :

$$A(q) = \begin{pmatrix} \sin \theta & \cos \theta \\ \cos \theta & \sin \theta \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (2.10)$$

Et en déduire :

$$B(q) = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \\ -\frac{1}{r} & 0 \end{pmatrix} \quad (2.11)$$

matrice formée par les colonnes indépendantes $b_1(q) = \left(\cos \theta \ \sin \theta \ 0 \ -\frac{1}{r} \right)^T$ et $b_2(q) = (0 \ 0 \ 1 \ 0)^T$ et qui vérifie bien $A^T(q)B(q) = 0$ pour tout q . On calcule alors les crochets de **Lie** successifs à partir de $b_1(q)$ et $b_2(q)$:

$$[b_1(q), b_2(q)] = (\sin \theta \ -\cos \theta \ 0 \ 0)^T \quad (2.12)$$

$$[b_1(q), [b_1(q), b_2(q)]] = 0$$

$$[b_2(q), [b_1(q), b_2(q)]] = (\cos \theta \ \sin \theta \ 0 \ 0)^T \quad (2.13)$$

On constate que $b_1(q)$, $b_2(q)$, $[b_1(q), b_2(q)]$ et $[b_2(q), [b_1(q), b_2(q)]]$ engendrent une algèbre de **Lie** de dimension 4, égale à celle de q . On arrête donc le calcul des crochets et on peut conclure qu'il n'existe pas de contrainte intégrable. Les contraintes (2.7) et (2.8) sont donc des contraintes non holonomes.

2.3 Les grandes classes de robots mobiles et leurs modèles

2.3.1 Disposition des roues et centre instantané de rotation

C'est la combinaison du choix des roues et de leur disposition qui confère à un robot son mode de locomotion propre. Sur les robots mobiles, on rencontre principalement trois types de roues (figure 2.3) :

- les roues fixes dont l'axe de rotation, de direction constante, passe par le centre de la roue ;
- les roues centrées orientables, dont l'axe d'orientation passe par le centre de la roue ;
- les roues décentrées orientables, souvent appelées roues folles, pour lesquelles l'axe d'orientation ne passe pas par le centre de la roue.

De manière anecdotique on rencontrera aussi des systèmes particuliers, tels que les roues suédoises, les roues à plusieurs directions de roulement, etc.

Bien évidemment, pour un ensemble de roues donné, toute disposition ne conduit pas à une solution viable. Un mauvais choix peut limiter la mobilité du robot ou occasionner d'éventuels blocages. Par exemple, un robot équipé de deux roues fixes non parallèles ne pourrait pas aller en ligne droite!

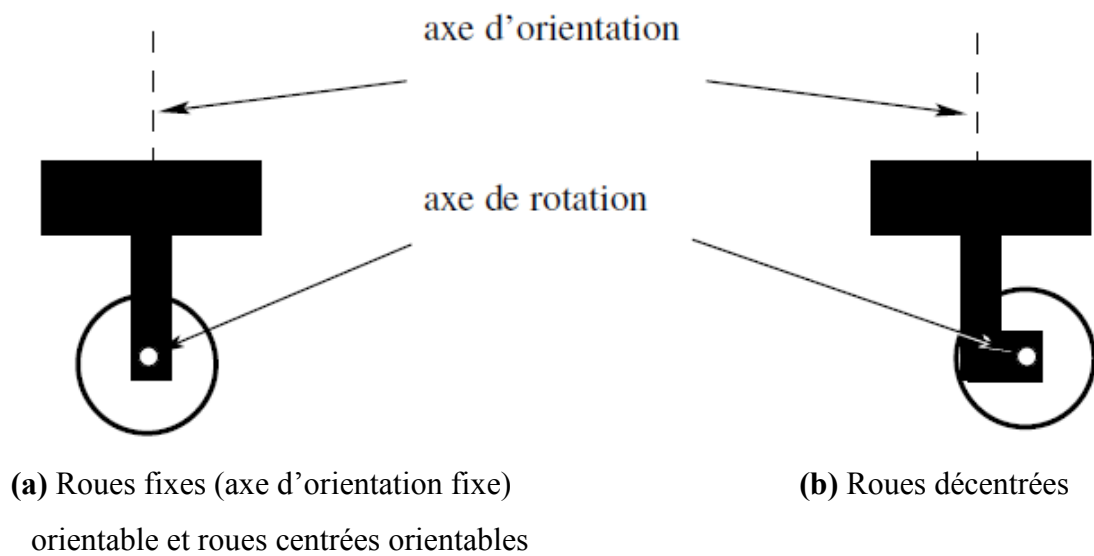


Figure 2.3 : Principaux types de roues

Pour qu'une disposition de roues soit viable et n'entraîne pas de glissement des roues sur le sol, il faut qu'il existe pour toutes ces roues un unique point de vitesse nulle autour duquel tourne le robot de façon instantanée. Ce point, lorsqu'il existe, est appelé centre instantané de rotation (CIR). Les points de vitesse nulle liés aux roues se trouvant sur leur axe de rotation, il est donc nécessaire que le point d'intersection des axes de rotation des différentes roues soit unique. Pour cette raison, il existe en pratique trois principales catégories de robots mobiles à roues, que l'on va présenter maintenant.

2.3.2 Robots mobiles de type unicycle

2.3.2.1 Description

On désigne par *unicycle* un robot actionné par deux roues indépendantes et possédant éventuellement un certain nombre de roues folles assurant sa stabilité. Le schéma des robots de type unicycle est donné à la figure 2.4. On y a omis les roues folles, qui n'interviennent pas dans la cinématique, dans la mesure où elles ont été judicieusement placées.

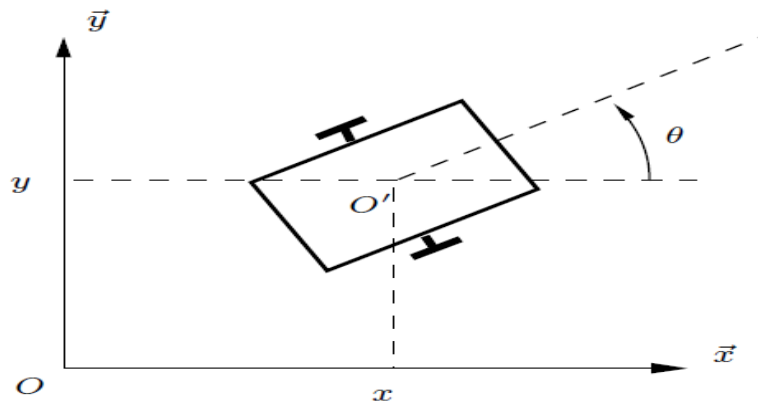


Figure 2.4 : Robot mobile de type unicycle

Ce type de robot est très répandu en raison de sa simplicité de construction et de propriétés cinématiques intéressantes. La figure 2.5 présente différents robots de type unicycle, depuis Hilare, en 1977, jusqu'aux modèles actuels, qui, à l'instar du robot Khepera, tendent parfois vers l'extrême miniaturisation.



Hilare, LAAS-CNRS, Toulouse, 1977 [GIR84]

Entraînement : batteries au plomb 24V, 2 moteurs DC avec codeurs incrémentaux

Calculateur : 4 processeurs Intel 80286, pas d'OS, multibus, modem série radio à 9600 bauds

Capteurs : odométrie, 16 capteurs US, un télémètre laser

Dimensions (L x l x h) : 80 cm _ 80 cm _ 60 cm

Poids : 400 kg



Khepera II, K-team, EPFL, Lausanne, 2002 [KTE02],

Entraînement : adaptateur secteur ou accus NiMH, 2 moteurs DC avec codeurs incrémentaux

Calculateur : processeur Motorola 68331 @25 MHz, 512 Ko RAM, 512 Ko mémoire Flash

Programmable par le port série, communication par port série jusqu'à 115 Kbauds (limitée à 9600 bauds en liaison radio), 3 entrées analogiques 0 à 4.3 V, 8 bits

Capteurs : 8 capteurs infrarouges de luminosité et de proximité (100 mm de portée) et un détecteur de batteries faibles

Dimensions (D x h) : 7 cm _ 3 cm

Poids : 80 g (charge admissible : 250 g)

Figure 2.5 : Evolution des robots mobiles de type unicycle

2.3.2.1 Modélisation

Centre instantané de rotation : Les roues motrices ayant même axe de rotation, le CIR du robot est un point de cet axe. Soit ρ le rayon de courbure de la trajectoire du robot, c'est-à-dire la distance du CIR au point O' (voir figure 2.6). Soit L l'entre axe et ω la vitesse de rotation du robot autour du CIR. Alors les vitesses des roues droite et gauche, respectivement notées v_d et v_g et définies à la figure 2.6, vérifient :

$$v_d = -r\dot{\varphi}_d = (\rho + L)\omega \quad (2.14)$$

$$v_g = r\dot{\varphi}_g = (\rho - L)\omega \quad (2.15)$$

Ce qui permet de déterminer ρ et ω à partir des vitesses des roues :

$$\rho = L \frac{\dot{\varphi}_d - \dot{\varphi}_g}{\dot{\varphi}_d + \dot{\varphi}_g} \quad (2.16)$$

$$\omega = -r \frac{\dot{\varphi}_d + \dot{\varphi}_g}{2L} \quad (2.17)$$

L'équation (2.16) permet de situer le CIR sur l'axe des roues. Par ailleurs ces équations expliquent deux propriétés particulières du mouvement des robots de type unicycle : si $\dot{\varphi}_d = -\dot{\varphi}_g$, le robot se déplace en ligne droite; si $\dot{\varphi}_d = \dot{\varphi}_g$, alors le robot effectue une rotation sur lui-même. L'utilisation de ces deux seuls modes de locomotion, bien que limitée, permet de découpler les mouvements et de fournir une solution simple pour amener le robot d'une posture à une autre. C'est sans doute là une des raisons du succès de ce type de robots. Pour élaborer une stratégie plus fine de déplacement, il est cependant intéressant de savoir comment la posture du robot est reliée à la commande de ses roues.

Choix de la commande : En ce qui concerne la commande, si l'on se contente de traiter le cas cinématique, on peut considérer que celle-ci est donnée, au plus bas niveau, par les vitesses de rotation des roues. Ceci étant, on préfère généralement exprimer cette commande par la vitesse longitudinale du robot, notée v (en O') et sa vitesse de rotation $\dot{\theta}$ (autour de O'). Il y'a en effet équivalence entre les deux représentations. D'une part, on a :

$$v = \frac{v_d + v_g}{2} = r \frac{\dot{\varphi}_g - \dot{\varphi}_d}{2} \quad (2.18)$$

D'autre part, la vitesse de rotation du robot est égale à la vitesse de rotation autour du CIR [DUD00] :

$$\omega = \dot{\theta} = -r \frac{\dot{\varphi}_d + \dot{\varphi}_g}{2L} \quad (2.19)$$

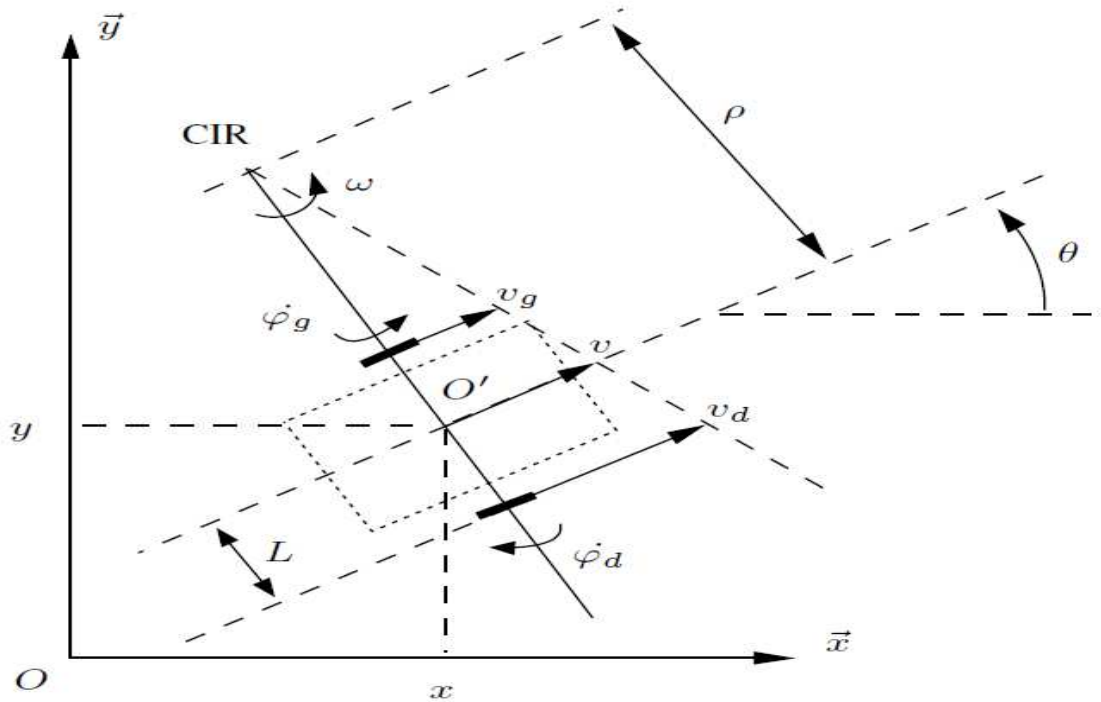


Figure 2.6 : Centre instantané de rotation d'un robot de type unicycle

Conformément à l'équation (2.17). On montre que ces relations sont parfaitement inversibles et qu'il y'a ainsi équivalence entre les couples $(\dot{\phi}_d, \dot{\phi}_g)$ et (v, ω) . Désormais, on utilise plutôt ce dernier couple de grandeurs, plus parlantes, quitte à calculer ensuite les angles ou vitesses de consigne des asservissements des roues.

Modèle cinématique en posture :

Relier la dérivée de la posture à la commande : $U = (v, \omega)^T$ est facile. Une simple considération géométrique donne :

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega\end{aligned}\tag{2.20}$$

Ce qui s'écrit, sous forme matricielle :

$$\dot{\varepsilon} = c(q)U\tag{2.21}$$

Avec :

$$c(q) = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix}\tag{2.22}$$

Ce modèle est appelé modèle cinématique en posture du robot [C96].

Modèle cinématique en configuration :

Considérons maintenant l'expression des conditions de r.s.g. des deux roues. On transpose les équations (2.5) et (2.6) aux deux roues de l'unicycle. Il vient alors :

$$\dot{x} + L\dot{\theta} \cos \theta + r\dot{\varphi}_d \cos \theta = 0$$

$$\dot{y} + L\dot{\theta} \sin \theta + r\dot{\varphi}_d \sin \theta = 0$$

$$\dot{x} - L\dot{\theta} \cos \theta - r\dot{\varphi}_g \cos \theta = 0$$

$$\dot{y} - L\dot{\theta} \sin \theta - r\dot{\varphi}_g \sin \theta = 0$$

En choisissant $q = (x, y, \theta, \varphi_d, \varphi_g)^T$ comme vecteur de configuration, ces contraintes se regroupent sous la forme matricielle $A^T(q)\dot{q} = 0$, avec :

$$A^T(q) = \begin{pmatrix} 1 & 0 & L \cos \theta & r \cos \theta & 0 \\ 0 & 1 & L \sin \theta & r \sin \theta & 0 \\ 1 & 0 & -L \cos \theta & 0 & -r \cos \theta \\ 0 & 1 & -L \sin \theta & 0 & -r \sin \theta \end{pmatrix} \quad (2.23)$$

On constate que $A(q)$ n'est pas de rang plein. On peut donc, sans perte d'information, supprimer la dernière ligne et réécrire les contraintes avec :

$$A^T(q) = \begin{pmatrix} 1 & 0 & L \cos \theta & r \cos \theta & 0 \\ 0 & 1 & L \sin \theta & r \sin \theta & 0 \\ 1 & 0 & -L \sin \theta & 0 & -r \sin \theta \end{pmatrix} \quad (2.24)$$

Par ailleurs il résulte des relations (2.17) et (2.19) la contrainte : $\dot{\theta} = -r \frac{\dot{\varphi}_d + \dot{\varphi}_g}{2L}$ qui s'intègre

$$\text{en :} \quad \theta = -r \frac{\varphi_d + \varphi_g}{2L} + \theta_0 \quad (2.25)$$

Où θ_0 est la valeur de θ à l'initialisation (les angles des roues, généralement mesurés par des codeurs incrémentaux, étant alors choisis nuls). Cette contrainte intégrable est donc en fait une contrainte holonome. On peut donc éliminer une des quatre variables constituant le vecteur q choisi initialement, qui n'était donc pas constitué de grandeurs indépendantes (et n'était donc pas un vecteur de configuration au sens strict).

En prenant maintenant $q = (x, y, \theta, \varphi_d)^T$ on trouve :

$$A^T(q) = \begin{pmatrix} 1 & 0 & L \cos \theta & r \cos \theta \\ 0 & 1 & L \sin \theta & r \sin \theta \end{pmatrix} \quad \text{Et} \quad B(q) = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \\ -\frac{1}{r} & -\frac{L}{r} \end{pmatrix}$$

En remarquant, d'après ce qui précède, que :

$$\dot{x} = v \cdot \cos \theta$$

$$\dot{y} = v \cdot \sin \theta$$

$$\dot{\theta} = \omega$$

$$\dot{\varphi}_d = -\frac{v}{r} - L \frac{\omega}{r}$$

On constate que, sous forme vectorielle, la dérivée du vecteur de configuration s'écrit :

$$\dot{q} = B(q) \cdot U \quad (2.26)$$

avec pour vecteur de commande cinématique $U = (v, \omega)^T$. La matrice $B(q)$ représente donc le modèle cinématique (en configuration) du robot mobile, ce résultat pouvant se généraliser [MOR03]. On notera que la connaissance de ce modèle n'est pas toujours utile du point de vue pratique. Dans le cas de l'unicycle, étant donné que l'on cherche essentiellement à contrôler la posture du robot et que la variable φ_d n'apparaît pas dans les équations régissant la dérivée de la posture, on se contentera généralement du modèle simplifié :

$$\begin{aligned} \dot{x} &= v \cdot \cos \theta \\ \dot{y} &= v \cdot \sin \theta \\ \dot{\theta} &= \omega \end{aligned} \quad (2.27)$$

On remarque que dans ce cas précis, le modèle simplifié est le modèle cinématique en posture. Ce n'est cependant pas le cas pour tous les robots mobiles.

2.3.3 Robots mobiles de type tricycle et de type voiture

Ces robots partagent des propriétés cinématiques proches, raison pour laquelle on les regroupe ici.

2.3.3.1 Description

Considérons tout d'abord le cas du tricycle, représenté à la figure 2.7. Ce robot est constitué de deux roues fixes de même axe et d'une roue centrée orientable placée sur l'axe longitudinal du robot. Le mouvement est conféré au robot par deux actions : la vitesse longitudinale et l'orientation de la roue orientable. De ce point de vue, il est donc très proche

d'une voiture. C'est d'ailleurs pour cela que l'on étudie le tricycle, l'intérêt pratique de ce type de robot (peu stable) restant limité.

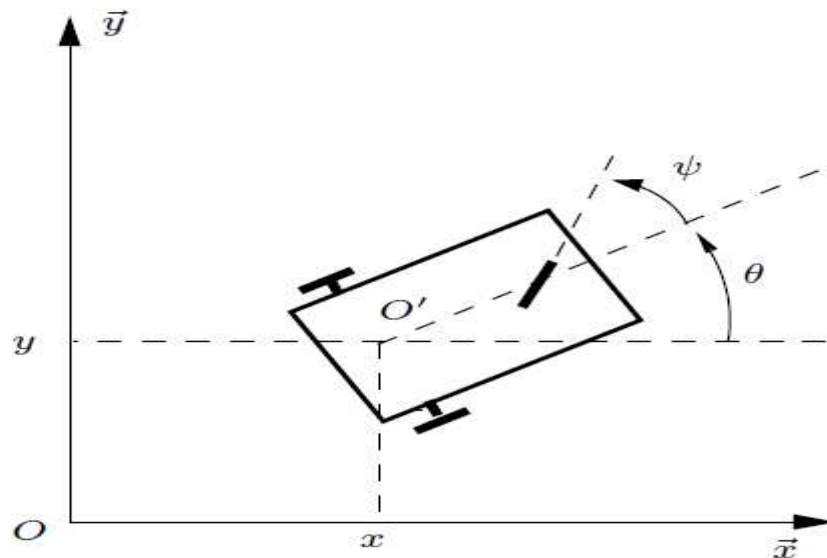


Figure 2.7 : Robot mobile de type tricycle

Le cas du robot de type voiture est très similaire à celui du tricycle. La différence se situe au niveau du train avant, qui comporte deux roues au lieu d'une. Cela va de soit, on rencontre beaucoup plus souvent ce type de systèmes. On parle de robot dès lors que la voiture considérée est autonome [KAN86], [URM04], donc sans chauffeur ni télépilotage. Il s'agit là d'un des grands défis issus de la robotique mobile. Deux réalisations sont montrées à la figure 2.8, basées sur des voitures de série instrumentées.



Figure 2.8 : Projets de voitures autonomes à l'université de Carnegie Mellon

2.3.3.2 Modélisation

Considérons tout d'abord le tricycle. Le CIR du robot se situe à la rencontre des axes des roues fixes et de la roue orientable, comme cela est représenté à la figure 2.9. On peut déterminer ρ de manière géométrique à partir de l'angle d'orientation de la roue avant et ω à partir de la vitesse linéaire v du véhicule (vitesse en O') et de ρ :

$$\rho = \frac{D}{\tan \varphi} \quad (2.28)$$

$$\omega = \frac{v}{D} \tan \varphi \quad (2.29)$$

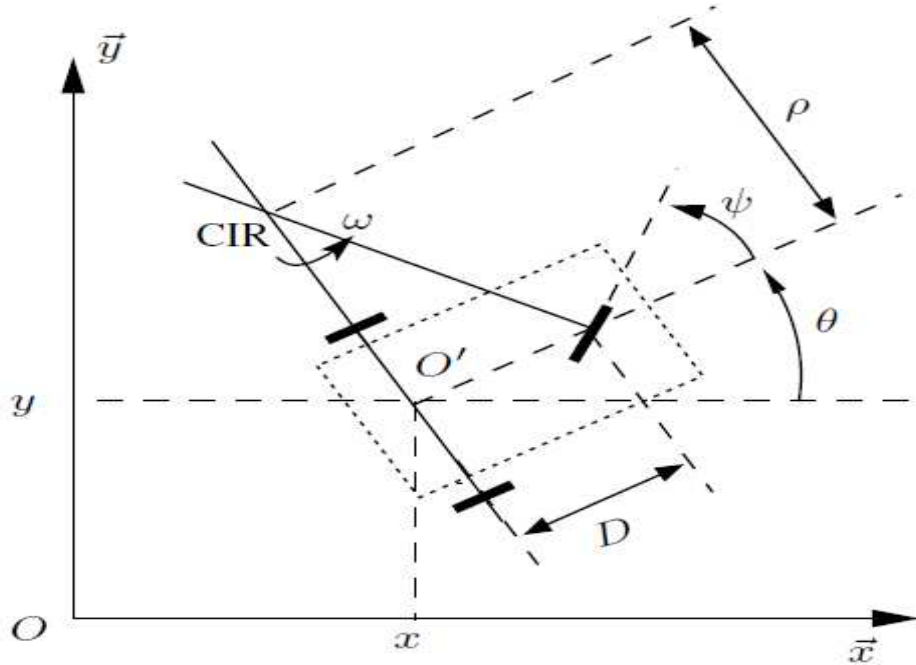


Figure 2.9 : Robot mobile de type tricycle et son CIR

Ce type de robot peut se diriger en ligne droite pour $\varphi = 0$ et théoriquement tourner autour du point O' (on pourrait dire sur place) pour $\varphi = \frac{\pi}{2}$. Néanmoins, le rayon de braquage de la roue orientable, généralement limité, impose le plus souvent des valeurs de φ telles que :

$$-\frac{\pi}{2} < \varphi < \frac{\pi}{2}, \text{ interdisant cette rotation du robot sur lui même.}$$

L'écriture des contraintes sur chacune des roues et un raisonnement similaire à celui suivi dans le cas de l'unicycle permettent de déterminer les modèles cinématiques des robots de type tricycle. Toutefois, par un simple raisonnement géométrique, on établit les équations :

$$\begin{aligned} \dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \frac{v}{D} \tan \varphi \\ \dot{\varphi} &= \eta \end{aligned} \quad (2.30)$$

où $U = (v, \eta)^T$ est le vecteur de commande cinématique, η représentant la vitesse d'orientation imposée à la roue orientable. Ces équations sont celles du modèle cinématique

en configuration simplifiée, la configuration associée $q = (x, y, \theta, \varphi)^T$ ne décrivant pas les rotations propres des différentes roues.

Comme on l'a vu précédemment, l'existence d'un CIR unique impose que les axes des roues du robot soient concourants. Dans le cas du robot de type voiture, cela impose aux roues du train avant de n'avoir pas la même orientation, comme illustré à la figure 2.10. Ainsi, les roues avant d'un robot de type voiture (et à fortiori d'une voiture) ne sont pas parallèles. Le roulement idéal, assurant que le CIR est bien unique, est réalisé sur une voiture par un système de braquage différentiel (dit d'Ackerman). Par ailleurs, les trajectoires des roues n'ayant pas même rayon de courbure, leurs vitesses sont également différentes (et liées évidemment).

L'équivalence entre tricycle et voiture est facile à montrer. Il suffit pour cela de figurer une roue virtuelle qui transformerait un robot de type voiture en tricycle en plaçant la roue orientable du tricycle au centre de l'axe des roues avant de la voiture, orientée de sorte que le CIR reste inchangé, conformément à la figure 2.10.

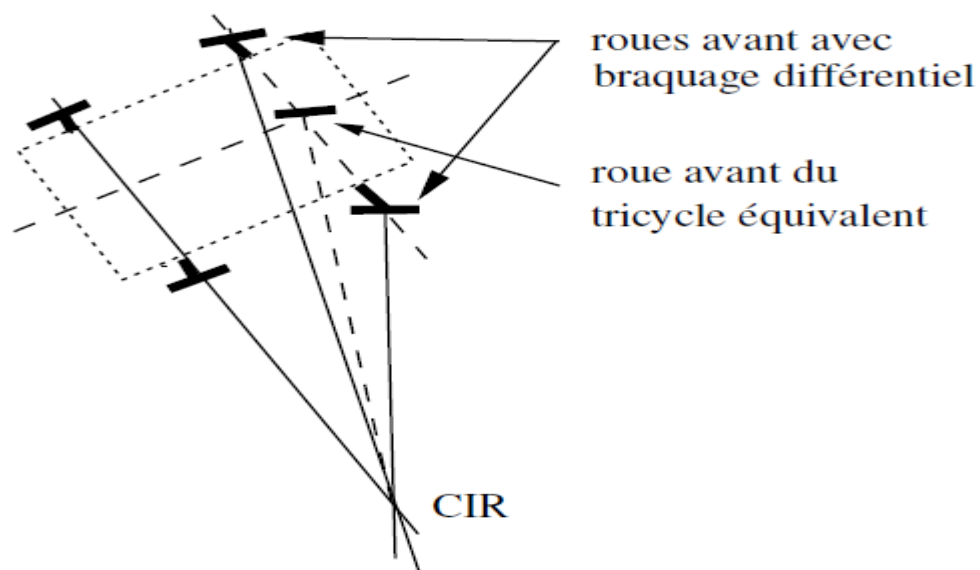


Figure 2.10 : Robot mobile de type voiture et son CIR

2.3.4 Robots mobiles omnidirectionnels

2.3.4.1 Description

Un robot mobile est dit omnidirectionnel si l'on peut agir indépendamment sur les vitesses : vitesse de translation selon les axes \bar{x} et \bar{y} et vitesse de rotation autour de \bar{z} .

D'un point de vue cinématique on montre que cela n'est pas possible avec des roues fixes ou des roues centrées orientables [CAM96]. On peut en revanche réaliser un robot

omnidirectionnel en ayant recours à un ensemble de trois roues décentrées orientables ou de trois roues suédoises disposées aux sommets d'un triangle équilatéral (voir figure 2.11). Du point de vue de la transmission du mouvement, ceci ne va pas sans poser de problème.

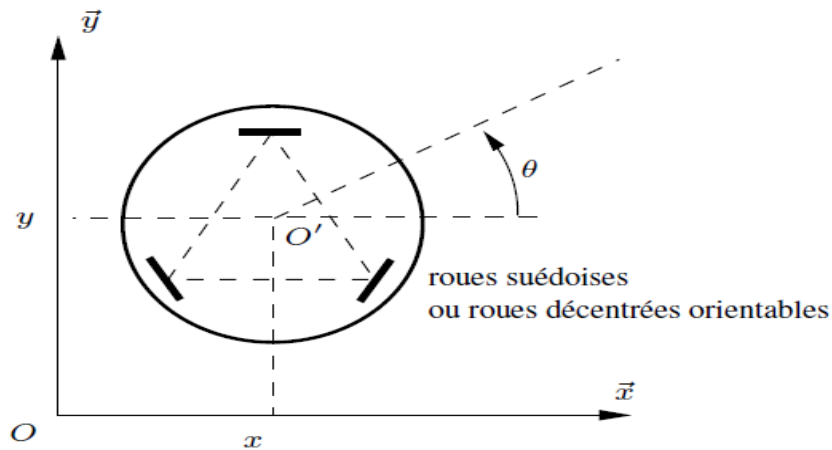


Figure 2.11 : Représentation d'un robot mobile omnidirectionnel

2.3.4.2 Modélisation

Dans ce cas on peut considérer qu'il est possible de commander directement la posture et le modèle cinématique en posture est donc :

$$\begin{aligned}\dot{x} &= v_1 \\ \dot{y} &= v_2 \\ \dot{\theta} &= v_3\end{aligned}\tag{2.31}$$

où $u = (v_1 \ v_2 \ v_3)^T$ représente le vecteur de commande. On choisit ainsi généralement ce type de robot pour se dispenser des problèmes de planification et de commande liés à la non-holonomie. L'avantage d'une cinématique extrêmement simple est cependant à mettre en balance avec les inconvénients liés à une localisation odométrique déficiente et à une plus grande complexité mécanique, généralement responsable d'un surcoût. On citera pour mémoire le cas du XR4000 Nomadic représenté à la figure 2.12, disparu en 2000.



Figure 2.12 : Robot mobile omnidirectionnel Nomadic XR4000

2.4 Propriétés du modèle cinématique d'un robot

2.4.1 Représentation d'état

De la modélisation que nous avons exposée tout au cours de ce chapitre il résulte, dans tous les cas, un modèle dynamique (au sens de l'Automatique) sous la forme (2.26). Si l'on considère que la configuration (réduite) q fait office de variable d'état du système on a :

$$\dot{x} = B(x)u \quad (2.32)$$

Avec $B(x)$ de dimension $n \times m$. Il s'agit d'une représentation non linéaire sans terme de dérivé, en comparaison de la représentation d'état classique d'un système linéaire invariant $\dot{x} = Ax + Bu$

C'est le cas des robots mobiles de type unicycle ou voiture, qui par ailleurs font partie d'une classe de système non linéaires particuliers dits systèmes chaînés, comme on le précisera au paragraphe consacré à la commande des robots.

2.4.2 Commandabilité des robots mobiles à roues

Pour qu'un robot mobile soit utile (ou utilisable) il faut en premier lieu s'assurer de sa commandabilité. Cette propriété signifie qu'il existe toujours une loi de commande $u(t)$ amenant le robot d'un état initial à un état final quelconque.

Pour caractériser la commandabilité d'un système non linéaire, on peut tout d'abord se demander si son linéarisé autour de tout point d'équilibre est commandable. En x_0 quelconque, le modèle linéarisé issu de (2.32) s'écrit : $\dot{x} = B(x_0)u$ avec $B(x_0) \in R^{n \times m}$. Pour les robots de type unicycle et voiture, la non holonomie va de pair avec une forme de sous-

actionnement qui se traduit par le fait que $m < n$. Appliquée à un tel système, la condition de rang (critère de commandabilité de Kalman) : $\text{Rang}(B, AB, \dots, A^{n-1}B) = n$ se résume à : $\text{Rang}B(x_0) = n$. Cette condition n'est jamais remplie puisque $m < n$. Le linéarisé du système n'est donc pas commandable autour d'un point d'équilibre quelconque.

Il faut alors, pour statuer sur la commandabilité du système, utiliser le théorème de **Chow**. On donne ici (sans preuve) son interprétation pour un système dont le modèle cinématique est donné par (2.32).

Théorème : *Commandabilité d'un robot mobile non holonome*

On note $B(x) = (b_1(x) b_2(x) \dots b_m(x))$ la matrice du modèle cinématique (2.32), de dimension $n \times m$.

Un robot mobile est commandable si les colonnes de $B(x)$ et leurs crochets de **Lie** successifs forment un ensemble de n colonnes indépendantes.

CHAPITRE 3

NAVIGATION DES ROBOTS MOBILES

3.1 Introduction

Dans le cadre de la navigation d'un robot mobile autonome, deux grandes familles de méthodes permettent à un robot d'atteindre une position souhaitée. Les méthodes sans trajectoire explicite (champs de potentiels, réseaux de neurones et logique floue) cherchent plutôt à contrôler le mouvement global du robot de manière à le guider vers son but. L'autre famille, celle des méthodes de suivi de trajectoire, telles que les fonctions transverses et les sorties plates, permet au robot de suivre « du mieux possible » une trajectoire de référence donnée, connaissant les contraintes cinématiques.

Le lien fort entre le type de méthode de navigation et l'architecture de contrôle global nous conduit à une réflexion qui nous amène à redéfinir la problématique de navigation dans le cadre d'une architecture de contrôle.

3.2 Navigation

Par abus de langage, un navigateur désigne généralement un module chargé de faire mouvoir le robot mobile dans son milieu d'évolution. Si on fait l'analogie avec le pilotage d'une voiture, se déplacer d'un endroit A à un endroit B nécessite au moins 3 tâches bien distinctes :

- Réfléchir à un itinéraire pour se rendre au point B : prendre telle autoroute en direction de telle ville, prendre la sortie numéro tant, tourner à droite au niveau du premier rond point etc.... C'est ce que nous appelons la planification de chemin. Il est préférable de déterminer son itinéraire à l'avance, et pour y parvenir quand on effectue le trajet pour la première fois, il est nécessaire d'utiliser des cartes, ou une représentation préétablie, pour trouver le meilleur itinéraire possible,

- Sur la route, il faut suivre du mieux possible l'itinéraire, en adaptant sa vitesse à la route, la circulation, aux limitations de vitesses, aux conditions météorologiques etc.... Il faut prendre correctement les ronds points, les virages, se frayer un passage dans la circulation. Il faut déterminer des trajectoires qui vont permettre de suivre du mieux possible l'itinéraire prévu, tout en s'adaptant aux conditions de circulation. Ces trajectoires seront dépendantes du

véhicule que l'on conduit (voiture, moto etc...), chaque véhicule ayant des capacités propres. Ce travail d'adaptation, c'est le travail du navigateur,

- Enfin, pour pouvoir suivre une trajectoire donnée, il faut appuyer sur l'accélérateur, tourner le volant, de manière générale il faut agir sur le véhicule. C'est le travail du pilote, qui doit agir sur les actionneurs du système pour appliquer la trajectoire souhaitée.

Dans l'architecture de contrôle, le navigateur reçoit un chemin de la part du planificateur, qui peut être sous différentes formes plus ou moins complètes : série de points de passage (x, y) pouvant contenir des informations supplémentaires comme l'angle d'orientation du robot, des vitesses indicatives etc... Concrètement, le rôle du navigateur consiste à générer des trajectoires qui soient réalisables par le robot (c'est-à-dire qui respectent le modèle cinématique du robot, les contraintes de non holonomie ainsi que les différentes contraintes liées aux actionneurs), qui n'intersectent pas avec les obstacles dans l'environnement local de navigation, et qui suivent au mieux le parcours proposé par le planificateur.

En fonction des méthodes utilisées, nous allons nous apercevoir que les délimitations entre pilotage/navigation/planification sont plus ou moins définies, il peut y avoir un recouvrement des tâches entre ces 3 grandes fonctions. Par exemple le rôle de navigateur peut-être tenu en partie ou complètement par le planificateur de chemin, si celui-ci intègre les possibilités de mouvement du robot dans son calcul de parcours. Mais le navigateur est un point clé pour un robot mobile autonome, puisque c'est lui qui fait le lien entre les capacités de mouvement du robot et l'environnement.

3.3 Méthodes sans trajectoire

Le premier type d'approche pour résoudre le problème de navigation d'un robot mobile autonome est de rendre la cible du robot attractive pour celui-ci, de manière à ce qu'il cherche à s'en rapprocher. Et inversement rendre les obstacles répulsifs, pour qu'il s'en écarte et reste à bonne distance. Dans cette approche, on ne cherche pas à anticiper la trajectoire du robot, on le laisse librement décider à chaque instant dans quelle direction il doit se rendre, à partir des informations recueillies concernant son environnement local. On fait l'hypothèse que contrôler son comportement général plutôt que de chercher à calculer une trajectoire, permettra tout de même au robot d'atteindre son but final.

Nous allons voir trois méthodes qui reposent sur cette philosophie : les champs de potentiels artificiels, les réseaux de neurones et la logique floue.

3.3.1 Champs de potentiels artificiels

Les méthodes de type APF (Artificial Potential Field) furent les premières employées dans les années 80 (Figure 3.1). Le principe des méthodes APF est de construire un champ de potentiels sur l'environnement de navigation du robot. La valeur de ce champ est minimale sur le point que le robot doit atteindre, et croît continûment au fur et à mesure que l'on s'écarte de ce point. Les obstacles génèrent un champ de potentiel répulsif pour le robot, de valeur supérieure à n'importe quel autre point ne correspondant pas à un obstacle du champ de potentiel. Et le champ de répulsion s'étend autour des obstacles avec une intensité inversement proportionnelle à la distance par rapport à l'obstacle.

L'idée est alors de demander au robot de se déplacer dans la direction du gradient de potentiel négatif le plus fort, sur le champ de potentiel global obtenu (figure 3.1).

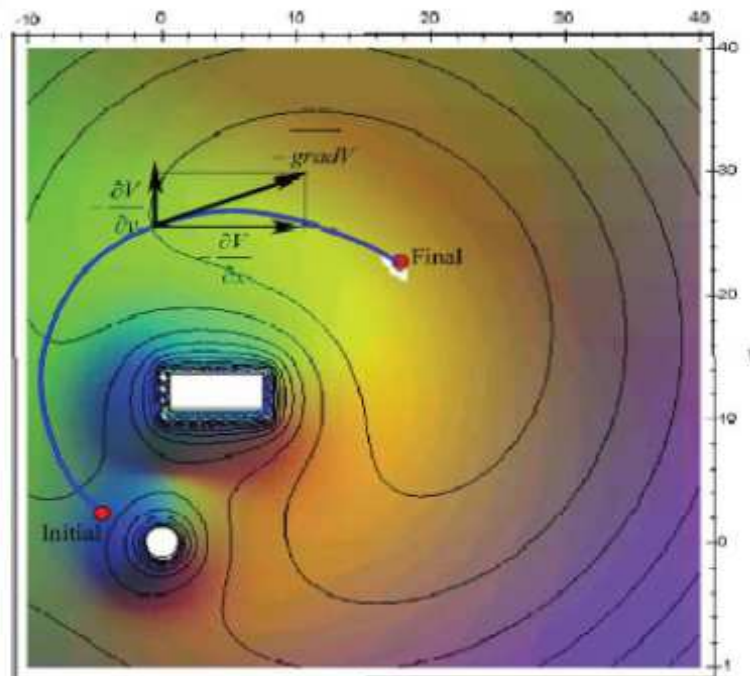


Figure 3.1 : Carte des champs de potentiels autour d'un obstacle

Khatib, ainsi que Andrews et Hogan dans [AND83], furent les premiers à imaginer l'idée de forces imaginaires agissant sur le robot. Ces méthodes présentent l'avantage d'être simples à mettre en œuvre, et elles furent les premières à être implantées physiquement sur de vrais robots en 1985 par Brooks [BRO86] et en 1989 par Atkin. Malgré le matériel informatique encore limité à cette époque, les calculs de trajectoires/commande étant très rapides avec ce type d'approche, cela permit de premières expérimentations sur des robots relativement lents.

Plus récemment dans [ADA03], Agirrebeitia propose une extension du principe des méthodes APF pour la navigation de robots dans un espace 3D.

L'expérimentation a révélé certains problèmes récurrents liés au principe même de ces méthodes :

- Minimas locaux entraînant des situations où le robot est piégé (classiquement le piège en forme de U).
- Pas de passage détecté entre des obstacles assez proches.
- Oscillations en présence d'obstacles dans les passages étroits.

Dans [KOR91], Koren et Borenstein ont démontré mathématiquement les problèmes d'instabilité de ces méthodes (entraînant des oscillations), problèmes qui apparaissent fortement dès lors que ces méthodes sont implémentées sur des systèmes « rapides ».

3.3.2 Réseaux de neurones

Le neurone est l'entité de base du système de réaction animal, et les neurologues McCulloch et Pitts furent les premiers à étudier leur fonctionnement chez la grenouille [LET59]. Les réseaux de neurones désignent à la fois l'étude de ces systèmes biologiques et leur modélisation informatique, plus ou moins simplifiée, à différentes applications, comme la reconnaissance de caractères, son premier terrain d'application historique, ou la navigation des robots mobiles autonomes. Dans ce cadre a été définie la notion de réseaux de neurones artificiels.

Dans un réseau de neurones artificiel (figure 3.2), chaque neurone possède plusieurs entrées et une sortie. Chacune de ses entrées se voit affecter un poids (dit poids synaptique) différent, et si la somme ainsi pondérée des signaux des différentes entrées dépasse un seuil, la sortie prend une valeur positive (le neurone se déclenche).

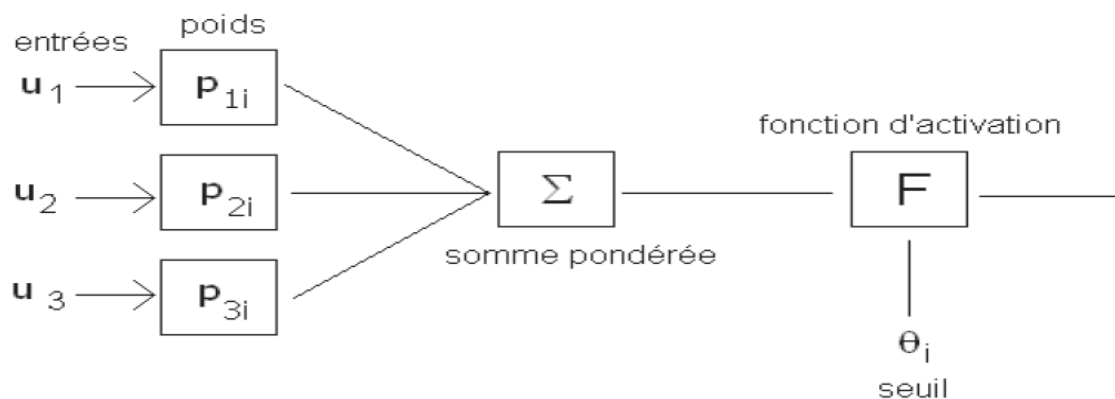


Figure 3.2 : Modélisation d'un réseau de neurones artificiel

Un réseau complet est constitué de différentes couches de neurones. Les premières couches correspondent à des couches de détection, en robotique les neurones correspondants

sont directement liés aux informations délivrées par les capteurs du robot. Les couches supérieures correspondent plutôt à des couches d'interprétation, et pour un robot elles seront plutôt liées aux effecteurs. De plus il y a généralement un certain nombre de couches intermédiaires entre les couches d'entrée et de sortie.

Un apprentissage du réseau est réalisable en adaptant les poids des entrées synaptiques. Pour cela, le principe est de donner au réseau un grand nombre d'exemples (set d'entrées donné), et de rétropropager l'erreur obtenue entre la sortie du réseau et la sortie attendue. L'algorithme d'apprentissage aura alors pour tâche de régler les différents poids synaptiques pour minimiser cette erreur.

De nombreux contrôleurs pour robots mobiles ont été construits sur ces principes [LEB05], [BEL02]. Les dernières générations utilisent un codage en fréquence pour la transition de l'information, comme le font les neurones biologiques. Sur une fenêtre temporelle donnée, le nombre de pics émis quantifie par exemple la proximité d'un obstacle. Wang dans [WAN08] a développé un contrôleur basé sur ce principe, pour un robot unicycle équipé de capteurs ultrasonores.

3.3.2 Logique floue

La logique binaire présente l'avantage de la simplicité, mais est assez éloignée de la façon humaine de raisonner. Si on prend l'exemple de la qualification de la proximité d'un obstacle, la logique floue permet de faire intervenir des notions telles que « assez près » ou « très loin », au lieu de se cantonner à une définition binaire « obstacle ou pas obstacle ». Celle-ci a été formalisée par Zadeh en 1965.

Le principe d'un contrôleur basé sur la logique floue se décline en trois phases : une phase de fuzzification, qui va transformer les variables d'entrée en variables floues ; une phase faisant appel à une table des règles de comportement, règles logiques du type « si (condition 1) et/ou (condition 2) alors (action sur les sorties) » ; et une dernière phase dite de défuzzification, permettant de traduire l'action déterminée par les règles de comportement en commande à envoyer aux actionneurs.

L'étape de fuzzification fait appel à des intervalles flous, qui vont délimiter l'espace des variables d'entrée en un certain nombre de sous ensembles flous (par exemple pour une proximité, on pourra avoir très proche (contact), assez proche, distance moyenne, assez loin et très loin) ; des fonctions d'appartenance sont alors utilisées pour définir le degré de vérité (probabilité d'appartenance) de la variable floue en fonction de la grandeur d'entrée. Ces fonctions peuvent être de type triangle, gaussienne etc.. Ainsi pour une mesure de distance

donnée, la règle d'appartenance nous dira « il y a 95% de chances que l'obstacle soit assez proche, 5% de chance que l'on soit en contact ». Cette notion est basée sur le fait qu'il y a toujours des incertitudes concernant les mesures des capteurs et les informations dont on dispose en général.

La seconde étape consiste en l'élaboration de règles de comportement pour le robot, suivant la combinaison des variables floues en entrée. La table des règles de comportement est construite manuellement, et est tributaire de l'expérience de la personne qui va régler le contrôleur. Pour un robot mobile, une règle pourra être : «s'il y a un obstacle assez proche sur la droite, il faut tourner à gauche et diminuer la vitesse du robot ».

La dernière étape, appelée défuzzification, consiste à transformer le comportement obtenu par la table des règles, en commande pour le robot. Une méthode qui peut être utilisée pour faire cela est celle des centres de gravité, qui va consister à faire la moyenne pondérée des commandes à appliquer.

Un contrôleur flou pour permettre la navigation d'un robot mobile de type voiture à double braquage, un Robucar, a été étudié dans [OUA05]. Ce contrôleur permet au robot d'atteindre sa position finale tout en respectant les contraintes cinématiques du robot, mais ne tient pour le moment pas compte des obstacles. Des travaux sont en cours pour intégrer la prise en compte de l'orientation finale du véhicule.

Dans [CHA96], R. Chatterjee et F. Matsuno utilisent la symétrie droite/gauche des règles de logique de comportement du robot pour simplifier celles-ci et ainsi diminuer le temps de calcul.

Le problème de ces méthodes par logique floue est généralement le même que celui des méthodes APF, c'est à dire le problème des minimas locaux qui se traduit par le fait que le robot peut rester piéger dans des culs de sac par exemple. Une technique pour sortir de ces pièges a été proposée par W.L. Xu [XUW99]. Cette technique utilise des cibles virtuelles pour sortir le robot du piège dans lequel il est tombé, à partir du moment où il reconnaît être tombé dans un piège (figure 3.3).

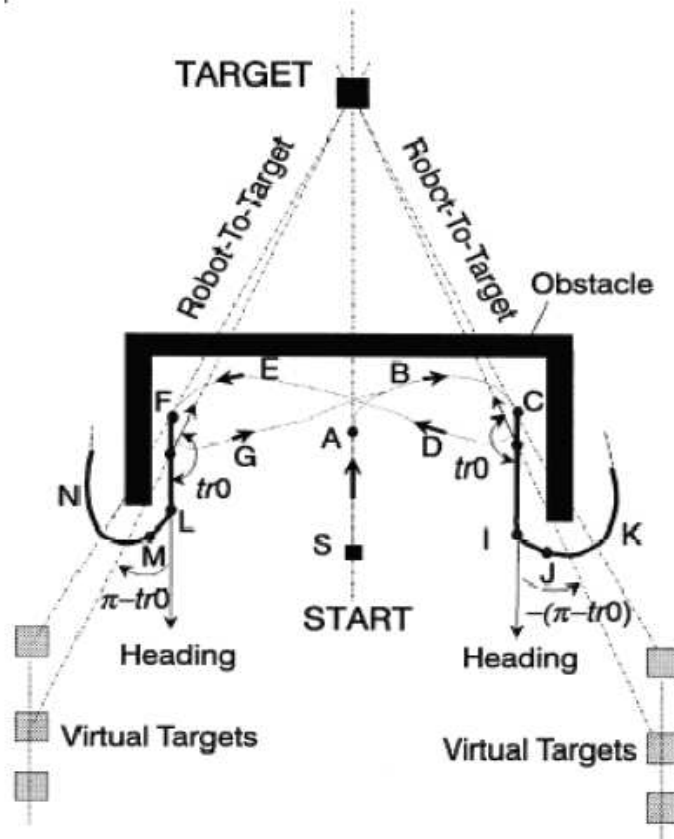


Figure 3.3 : Principe des virtual target pour sortir d'un minima local

Un autre problème inhérent aux méthodes par logique floue est qu'elles sont trop spécialisées pour un type d'environnement donné, et donc elles souffrent de problèmes d'adaptation à des environnements différents. Cependant il existe des méthodes dites d'apprentissage qui permettent au robot de modifier lui même ses règles de comportement au fur et à mesure qu'il explore un nouvel environnement. Le problème de ces méthodes est qu'elles demandent un temps d'apprentissage plus ou moins long avant que le robot ne puisse naviguer efficacement [ERM04].

3.4 Méthodes de suivi de trajectoire

Le second type d'approche consiste à contrôler le robot pour lui faire suivre une trajectoire précise. Ces méthodes (figure 3.4) nécessitent généralement de déterminer un modèle inverse du robot, c'est-à-dire un modèle qui permet de calculer les commandes à envoyer au robot (espace articulaire) connaissant la trajectoire que l'on souhaite lui faire suivre (dans l'espace cartésien).

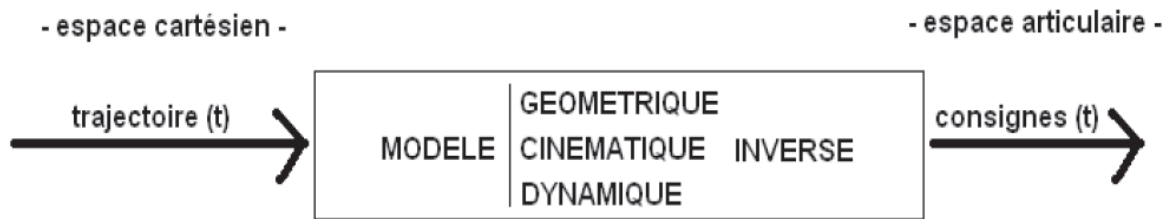


Figure 3.4 : Approche par modèle inverse pour effectuer un suivi de trajectoire

La détermination d'un tel modèle est souvent le point central de ce type de méthode, et le but est d'obtenir la meilleure convergence possible du robot vers la trajectoire de référence.

Autres recherches sur le suivi de trajectoire

D'autres méthodes pour faire du suivi de trajectoires sont basées sur la reconnaissance d'images le long du parcours. Ces images peuvent être prises avec une caméra, ou avec des télémètres laser. Le principe est de créer une base de données d'images à reconnaître, et pour cela le plus simple est de piloter le robot en manuel dans une phase dite d'enregistrement, en lui faisant suivre le parcours de référence. Une fois la base de données créée, on peut procéder à la phase de suivi de trajectoire en autonome. A chaque instant d'échantillonnage, l'image vue par le robot est comparée à l'image qui correspond dans la base de données, et la corrélation entre les deux images permet de calculer les commandes à envoyer au robot, pour le recalculer sur la trajectoire de référence.

Ce type de méthode a été appliqué à la navigation de véhicule en milieu urbain en utilisant des camera vidéo [COU08] [COU09], et en intérieur de bâtiments en utilisant des télémètres lasers.

La reconnaissance d'amers est un principe proche de la reconnaissance d'images, sauf que cette fois-ci on repère des éléments remarquables, fixes dans l'environnement de navigation, pour guider le robot le long du parcours. Cette méthode a été appliquée sur un véhicule CyCab, évoluant en milieu extérieur grâce à des balises installées dans l'environnement de navigation [PRA05].

3.5 Discussion

3.5.1 Synthèse des méthodes de navigation

Le problème de navigation en robotique mobile n'est pas un problème parfaitement arrêté. La frontière entre planification de chemin, navigation, et pilotage est plus ou moins définie, il existe souvent des recouvrements entre ces différents domaines connexes. Ces différences d'appréhension du problème se ressentent notamment au niveau des structures de contrôle des robots, qui sont en quelque sorte le reflet de la manière dont chaque équipe

perçoit et découpe les différentes tâches nécessaires à la réalisation d'un système robotique autonome. Il existe ainsi beaucoup de définitions différentes du navigateur, d'où certaines différences dans la manière de répondre à la problématique.

Par exemple, beaucoup de travaux ne font pas la distinction entre chemin, trajectoire et même pilotage, et tous ces problèmes sont ainsi résolus par un unique module. C'est le cas d'un grand nombre d'approches à base de réseaux de neurones et de logique floue, qui cherchent à résoudre tous ces problèmes en même temps en contrôlant un comportement global du robot. Ce genre d'approche peut donner de bons résultats dans une situation donnée. Mais il n'y a pas de planification à long terme de ce que le robot doit faire, le robot n'est donc pas capable de s'adapter à une situation inconnue. Il s'agit généralement d'approcher de la meilleure manière possible un comportement souhaité dans un type de situation donné, ce qui ne correspond pas à la définition d'un système autonome qui serait capable de s'adapter à différentes situations non prévues.

Une autre approche pour la navigation porte sur la détermination de méthodes pour effectuer un suivi de trajectoire ou de chemin. La trajectoire servant de référence peut être fournie par un module de planification, ou encore provenir d'un robot leader qu'il faut suivre à une certaine distance. En considérant un robot classique de type voiture, le problème à traiter provient notamment des contraintes de non holonomie des robots, qui réduisent sa mobilité et l'empêchent donc d'effectuer certains mouvements pour rattraper la trajectoire de référence. Dans ce domaine, les approches par fonctions transverses ou celles par les sorties plates ont montré d'excellents résultats. La difficulté de ces méthodes est qu'elles nécessitent de déterminer un modèle bien spécifique du robot. Les méthodes pour déterminer ces modèles ne sont pas systématiques, et de plus ces modèles n'existent pas pour tous les robots.

3.5.2 Nomenclature et lien avec une architecture de commande pour l'autonomie

L'étude des méthodes de navigation pour un robot autonome a mis en évidence les recouvrements entre les différentes fonctions nécessaires au commandement d'un robot mobile autonome. Il en découle une nécessité de définir plus précisément les différentes notions que nous développerons dans la suite de ce manuscrit, de manière à bien se mettre d'accord sur ce dont on parle. Cette nomenclature nous amènera alors à présenter l'architecture de commande sur laquelle elle s'appuie.

Ainsi, pour la suite de ce document, nous définissons les notions de chemin et trajectoire de la manière suivante :

- Un chemin est une succession continue ou discrète de positions et orientations du robot ; un chemin s'inscrit dans une planification globale du parcours que doit effectuer le robot. Typiquement un chemin sera constitué de différents points de passages successifs que le robot doit atteindre, et en outre le chemin est décorrélé de la notion de temps. Au mieux le chemin prendra en compte les paramètres géométriques du robot (circonférence du robot), mais pas ses capacités cinématiques. Il en résulte que le chemin détermine par où le robot devra passer, mais pas au bout de combien de temps il y passera.

- Une trajectoire contient les paramètres cinématiques du robot le long de son parcours, elle est donc directement reliée à la notion de temps contrairement au chemin. Il en découle qu'une trajectoire prendra en compte les capacités cinématiques et dynamiques du robot. Une trajectoire sera donc définie sur un horizon temporel donné, et prendra en compte des contraintes de nature plus locale que le chemin.

Cette manière de décomposer la notion de chemin et de trajectoire s'inscrit dans l'architecture de commande multi-niveaux présentée par Gilles Mourioux dans [MOU06]. Il s'agit d'une structure à plusieurs niveaux, qui contient plusieurs boucles action/perception imbriquées les unes dans les autres. Les couches supérieures sont les plus lentes et les plus « intelligentes », au sens où elles intègrent des informations provenant des couches inférieures, et elles travaillent à plus long terme et sur des espaces plus grands. Plus on descend dans les niveaux, et plus on a des boucles rapides et réactives.

Chaque bloc de la partie décision (figure 3.5) a pour entrée les ordres d'un bloc analogue de niveau supérieur, et les informations fournies par un bloc de la partie perception. Ces informations lui permettent de prendre ses décisions, qui sont les sorties du bloc. Ces ordres joueront alors le rôle de consigne pour le bloc décisionnel inférieur.

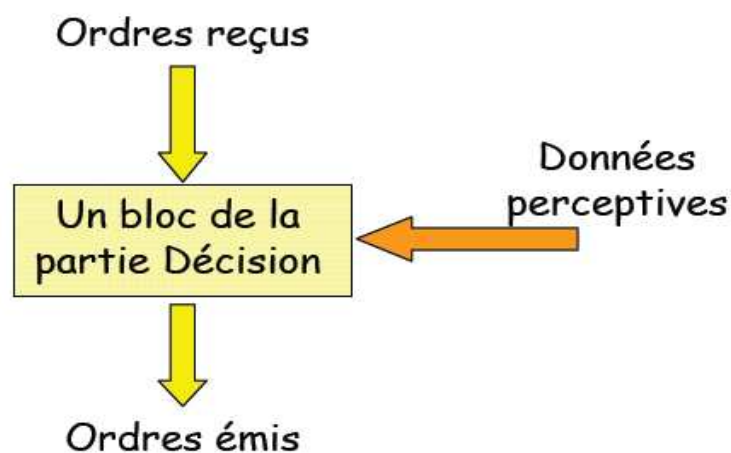


Figure 3.5 : Bloc de la partie décision

Les blocs de la partie perception (figure 3.6) reçoivent des informations émanant des capteurs. L'ensemble ou une partie des données reçues sont transmises vers les blocs de la partie décision, et également vers les blocs des niveaux supérieurs de la partie perception. Ces données peuvent avoir subi au préalable un traitement dans les blocs de la partie décision. Ainsi, niveau après niveau, les données brutes sont enrichies par les traitements successifs et la confrontation avec d'anciennes et nouvelles informations gardées en mémoire. Cela permet notamment d'améliorer la robustesse des informations et de réaliser des cartes.

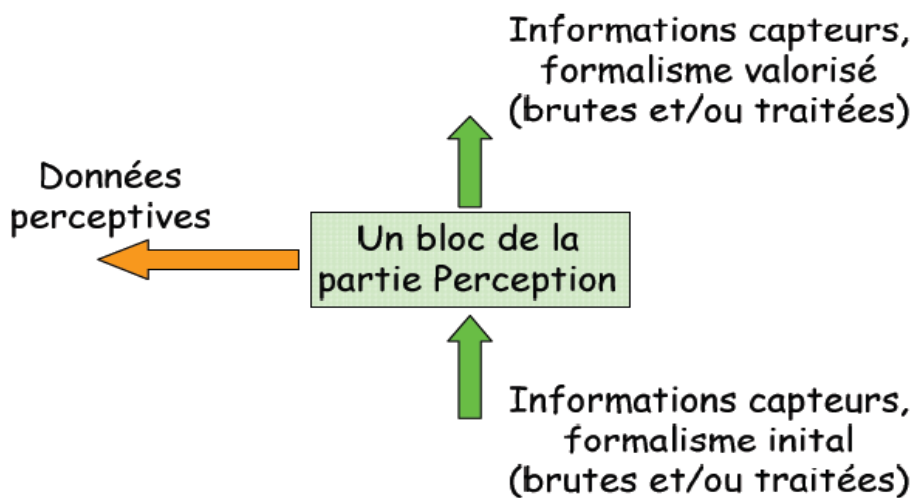


Figure 3.6 : Bloc de la partie perception

L'architecture générale (figure 3.7) est ainsi composée de différentes boucles action/perception sur différents niveaux, placés au dessus du système mécatronique articulé. Les boucles de bas niveau sont rapides et travaillent à court terme, elles correspondent à la partie réactive de l'architecture. Les boucles de haut niveau travaillent sur des horizons temporels et spatiaux plus importants, sont plus lentes et forment la partie délibérative de cette architecture.

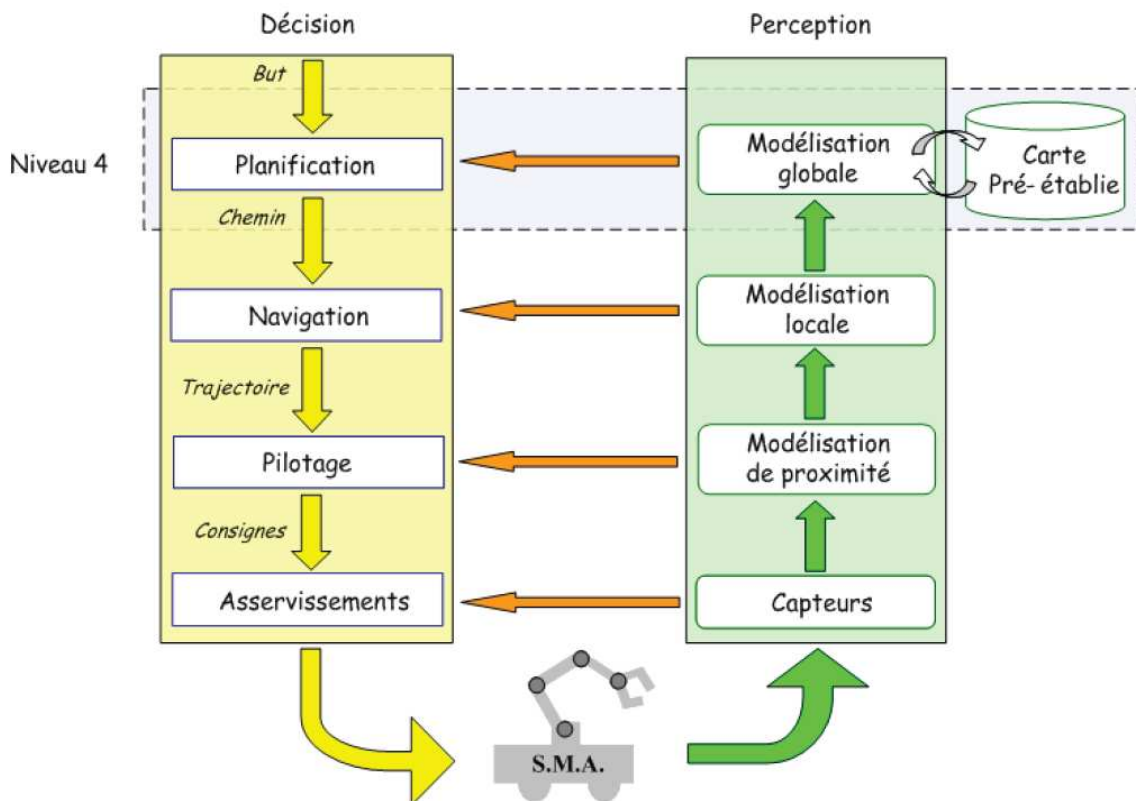


Figure 3.7 : Architecture de contrôle

Le niveau 0 représente la partie physique du robot, à savoir le système mécanique articulé et les actionneurs pour faire bouger le robot. Cet élément constitue la base sur laquelle toute l'architecture est construite.

Le niveau 1 est constitué des boucles d'asservissement bas niveau du robot. La partie perception contient les capteurs, et les informations qu'ils délivrent sont utilisées par le bloc de la partie décision en vue de l'asservissement de chaque moteur.

Le second niveau est celui du pilote. Il reçoit une trajectoire à suivre de la part du navigateur, et utilise les informations fournies par le bloc perception pour déterminer les consignes à envoyer aux asservissements de chaque moteur. La partie perception effectue un traitement des données brutes, par exemple sous forme de segmentation ou de détermination de l'espace libre, qu'elle fournit au pilote. En plus de sa mission principale consistant à transformer une trajectoire fournie par le navigateur en consignes pour les moteurs, le pilote peut également avoir la capacité de prendre certaines décisions comme des manœuvres d'évitement d'obstacle en urgence.

Le niveau 3 est celui du navigateur. Cette couche utilise un modèle de l'environnement local du robot, généralement sous forme de carte, fabriqué par la partie perception de ce niveau. Cette carte est réactualisée en permanence par les nouvelles

informations des capteurs externes qui sont remontées et traitées par les blocs perception inférieurs. Le navigateur se sert de cette carte locale pour générer une trajectoire pour le robot, lui permettant de se déplacer entre les obstacles tout en cherchant à suivre du mieux possible un parcours. Ce parcours lui est fourni par le niveau supérieur, le planificateur. La trajectoire générée tient ainsi compte des obstacles, mais elle tient également compte des contraintes cinématiques voire dynamiques du robot. Le navigateur est donc le bloc charnière entre ce que l'on souhaite que le robot fasse (le chemin), les contraintes liées aux capacités de mouvement du robot, et les contraintes locales que sont les obstacles. Une fois la trajectoire déterminée, le navigateur l'envoie au pilote, qui se chargera de la faire appliquer par les actionneurs.

Le quatrième et dernier niveau représenté est celui du planificateur. La partie décision de ce bloc se voit fournir un but à atteindre, qui peut être une destination finale ou une série de lieux à visiter. Sa mission consiste alors à déterminer le chemin optimal pour atteindre ce but, chemin qui sera envoyé comme consigne au bloc navigateur du 3^e niveau. Pour cela, il utilise un modèle global de son environnement fourni par la partie perception du niveau 4. Pour fabriquer cette carte globale, la partie perception utilisera généralement des cartes préétablies présentes dans sa base de données, qu'elle pourra fusionner avec des informations courantes comme la carte locale de manière à actualiser la modélisation de l'environnement.

Dans l'architecture de contrôle proposée par Gilles Mourioux, il existe un niveau supérieur nommé générateur de mission, qui permet de donner un niveau d'autonomie supplémentaire au système. Ce niveau détermine un but ou une mission utilisable comme entrée pour le planificateur de chemin. Enfin, cette architecture permet également d'intégrer la téléopération par l'intermédiaire d'un troisième bloc, en surplus des blocs décision et perception. Ce bloc téléopération est subdivisé en couches de la même manière que les deux autres blocs, ce qui permet de faire intervenir un mode téléopéré à n'importe quel niveau.

3.6 Poursuite de trajectoire

Pour une trajectoire donnée, il faut choisir un point virtuel, lié au robot, qui va suivre cette trajectoire. Pour un véhicule de type RMI (Roues Motrice Indépendantes), il est usuel de choisir le point milieu de l'axe des roues motrices. Ainsi la direction du véhicule est toujours tangente à la trajectoire de ce point.

Parfois, il est préférable de commander sur la trajectoire un point virtuel n lié au robot et situé en avant des roues motrices, (généralement c'est le point où est fixé un appareil de mesure de type goniomètre ou télémètre). Le système en boucle fermée doit annuler la

distance l et l'angle α entre le point n et la tangente (T) à la trajectoire au point B et telle que (nB) soit perpendiculaire à (T) (figure 3.8)

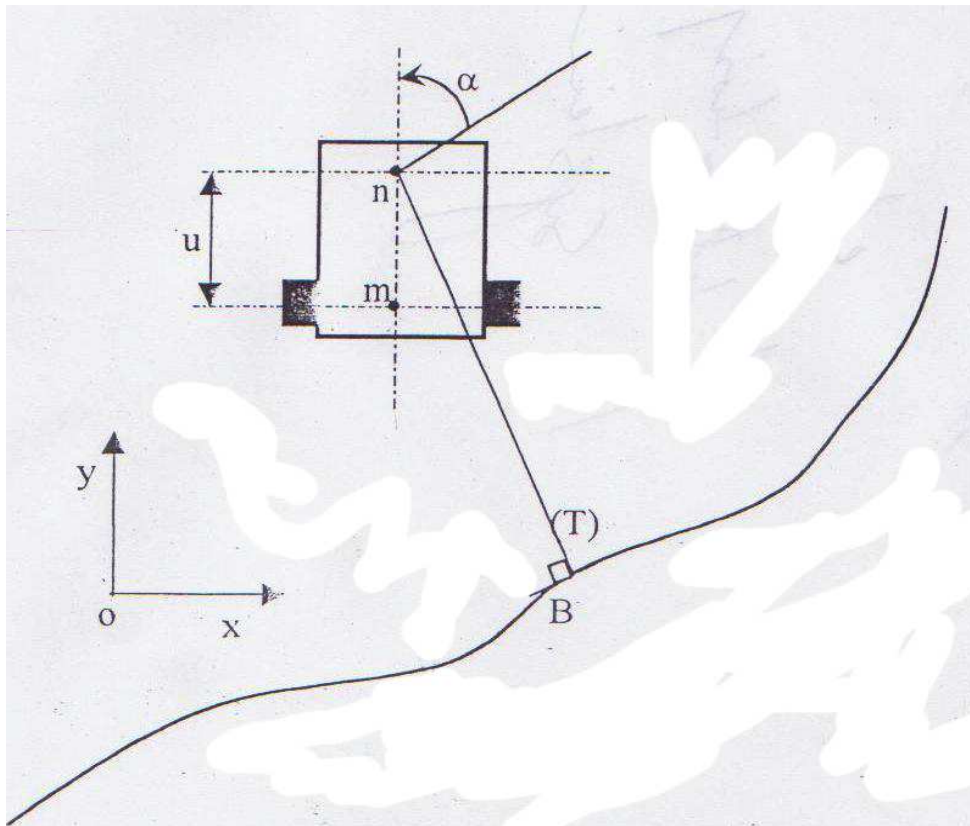


Figure 3.8 : Définition du suivi de trajectoire

3.6.1 Modélisation du système

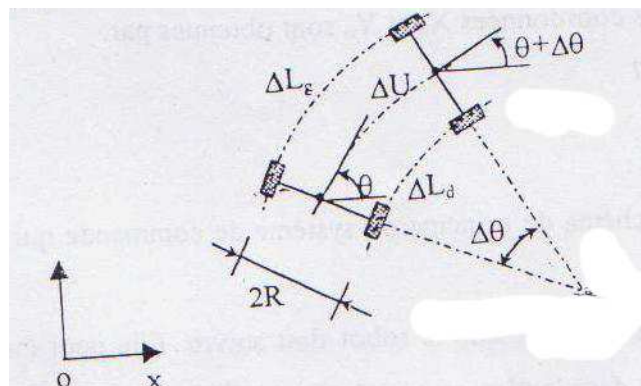


Figure 3.9 : Evolution du robot

D'après la figure 3.9, on peut tirer les équations suivantes :

$$\Delta\theta = \frac{(\Delta L_g - \Delta L_d)}{2R} \quad (3.1)$$

$$\Delta U = \frac{(\Delta L_g + \Delta L_d)}{2}$$

En dérivant, on aboutit aux équations donnant les vitesses de déplacement et de rotation du robot comme suit :

$$\frac{d\theta}{dt} = \frac{(v_g - v_d)}{2R} \quad (3.2)$$

$$v = \frac{(v_g + v_d)}{2}$$

v est la vitesse linéaire du point milieu des deux roues arrières.

Nous choisissons de commander symétriquement en vitesse les roues droite et gauche du robot ainsi nous aurons :

Pour la roue droite : $v_d = v + \Delta v$

Pour la roue gauche : $v_g = v - \Delta v$

Les composantes scalaires de la vitesse du point m sont alors données par :

$$\frac{dx_m}{dt} = v \cos \theta$$

$$\frac{dy_m}{dt} = v \sin \theta \quad (3.3)$$

$$\frac{d\theta}{dt} = \frac{\Delta v}{R}$$

D'après la figure 3.8, les coordonnées X_n et Y_n sont obtenues par :

$$X_n = X_m + u \cos \theta$$

$$Y_n = Y_m + u \sin \theta \quad (3.4)$$

La figure 3.10 décrit le schéma de principe du système de commande qui réalise un suivi de trajectoire. L'entrée du système est la trajectoire que le robot doit suivre. Elle peut être donnée sous la forme d'une équation dans le plan ou sous la forme d'une suite de points.

A partir de la trajectoire désirée nous calculons la position et l'orientation du robot par rapport à celui-ci et nous en déduisons les deux paramètres X_m et Y_m , en coordonnées du point N à asservir. Le système est ainsi rebouclé.

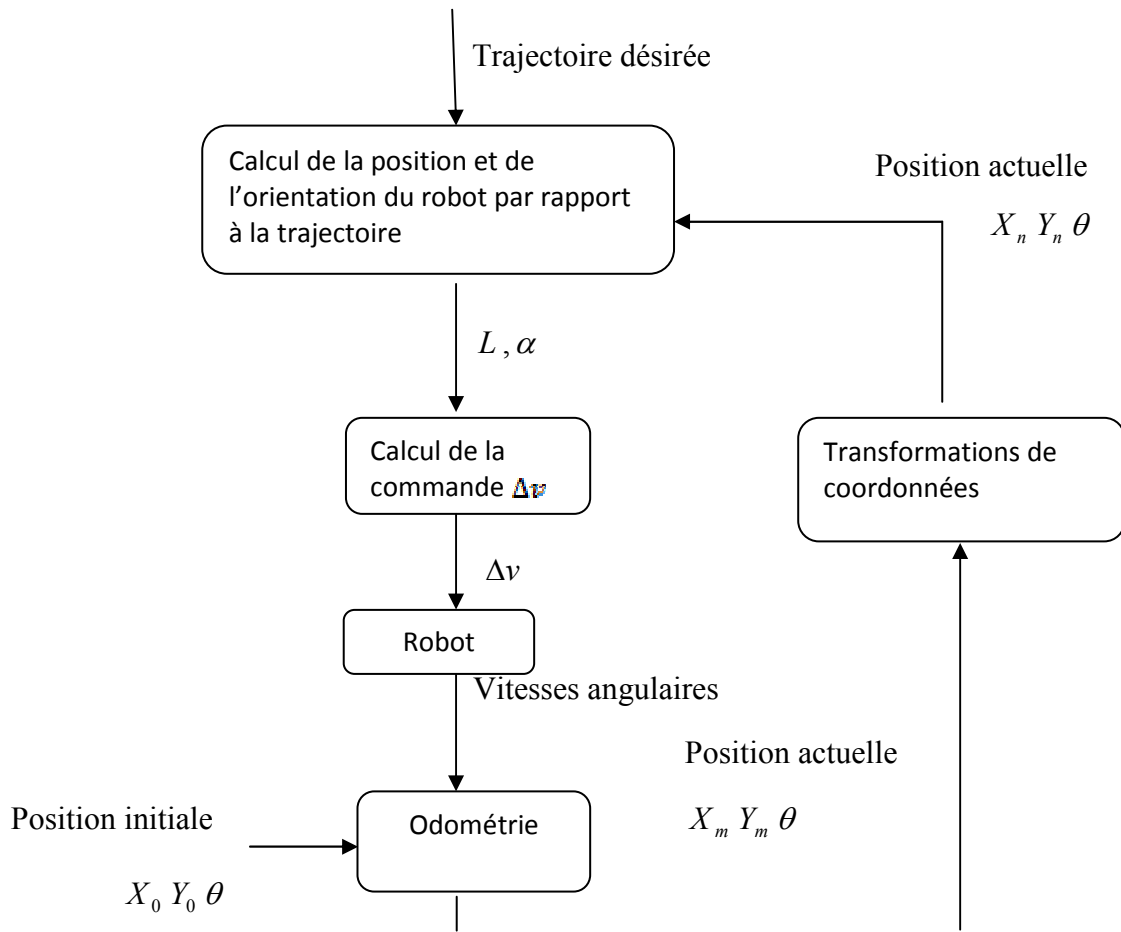


Figure 3.10 : Schéma de principe du suivi de trajectoire

Nous allons appliquer ce principe dans deux cas de suivi :

- Le suivi de trajectoire linéaire
- Le suivi de trajectoire circulaire

Ces deux cas s'avèrent suffisants pour beaucoup d'applications.

Les deux trajectoires sont définies par leurs équations dans le plan. Il est donc nécessaire maintenant de calculer deux paramètres L et α en fonction des coordonnées (X_n, Y_n) .

3.6.2 Calcul des paramètres l et α

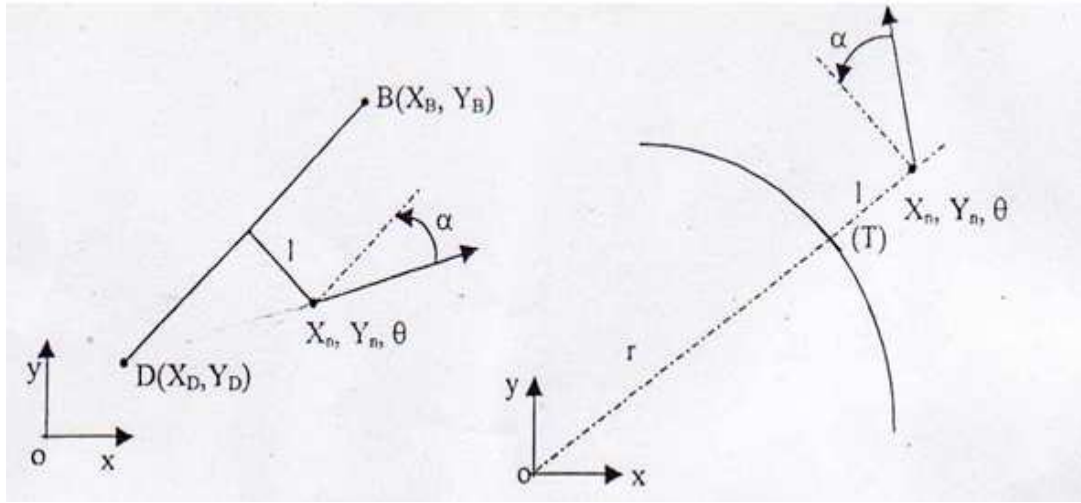


Figure 3.11 : Définition des trajectoires à suivre

L'équation de la droite (DB) (figure 3.11) est définie par :

$$\alpha(X - X_D) + b(Y - Y_D) = 0 \tag{3.5}$$

Avec :

$$\begin{aligned} \alpha &= Y_D - Y_B \\ b &= X_B - X_D \end{aligned} \tag{3.6}$$

La distance l est donnée par :

$$l = (X_n - X_D) \sin \varphi - (Y_n - Y_D) \cos \varphi \tag{3.7}$$

Et l'angle α est donné par :

$$\alpha = \arctg \left[\frac{Y_B - Y_D}{X_B - X_D} \right] - \theta \tag{3.8}$$

Pour le cercle centré sur l'origine, nous avons comme équation :

$$X^2 + Y^2 = r^2 \tag{3.9}$$

Le calcul de l donne :

$$l = \sqrt{X_n^2 + Y_n^2} - r \tag{3.10}$$

Et pour l'angle α :

$$\alpha = \arctg \left[\frac{Y_n}{X_n} \right] + \frac{\pi}{2} - \theta \tag{3.11}$$

X_m et Y_m étant obtenus par odométrie

3.6.3 Simulation du suivi de trajectoire

Le système d'équation régissant le mouvement du point n est tiré à partir des relations (3.3) et (3.4) :

$$\begin{aligned}\frac{dX_n}{dt} &= v \cos \theta - u \sin \theta \frac{d\theta}{dt} \\ \frac{dY_n}{dt} &= v \sin \theta + u \cos \theta \frac{d\theta}{dt} \\ \frac{d\theta}{dt} &= \frac{\Delta v}{R}\end{aligned}\quad (3.12)$$

Ce système doit assurer la convergence du point n vers la trajectoire décrite par L .

Nous choisissons une loi de commande introduisant une boucle d'orientation et une boucle de position. Elle s'écrit :

$$\frac{d\theta}{dt} = \frac{1}{R} (k_1 l + k_2 \alpha) \quad (3.13)$$

Nous avons alors la commande Δv :

$$\Delta v = (k_1 l + k_2 \alpha) \quad (3.14)$$

3.6.4 Résultats de simulation

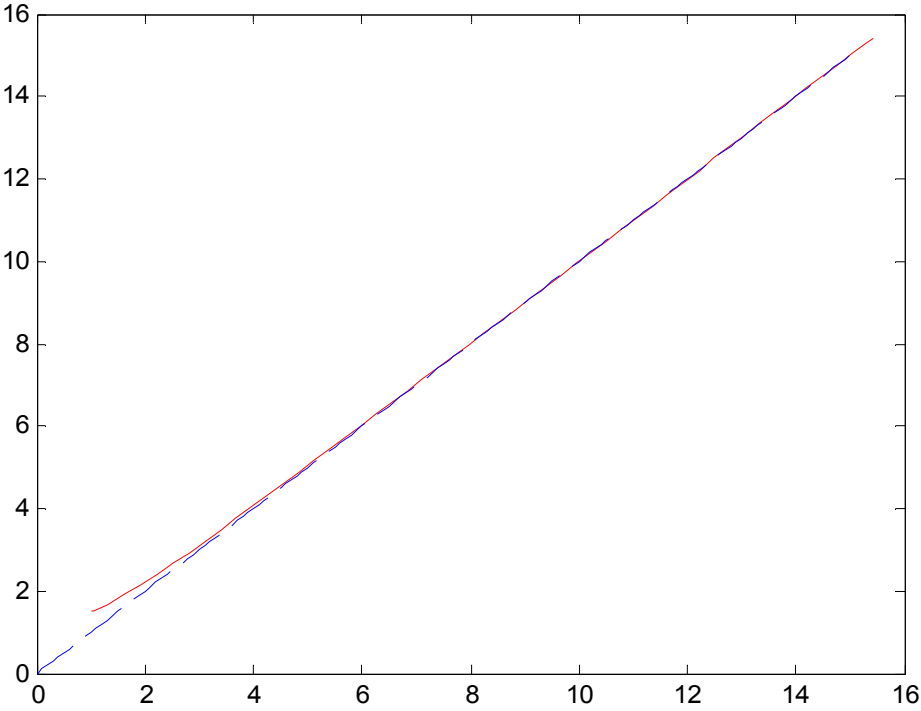
Nous implantons la loi de commande en Δv , pour un suivi de trajectoire linéaire et circulaire en utilisant les expressions de l et α calculées précédemment.

Pour réaliser ces trajectoires, nous déterminons les valeurs des gains k_1 et k_2 qui nous permettent d'obtenir les meilleures performances, pour cela nous avons choisi les valeurs suivantes : $k_1 = 0.2$ et $k_2 = 0.5$.

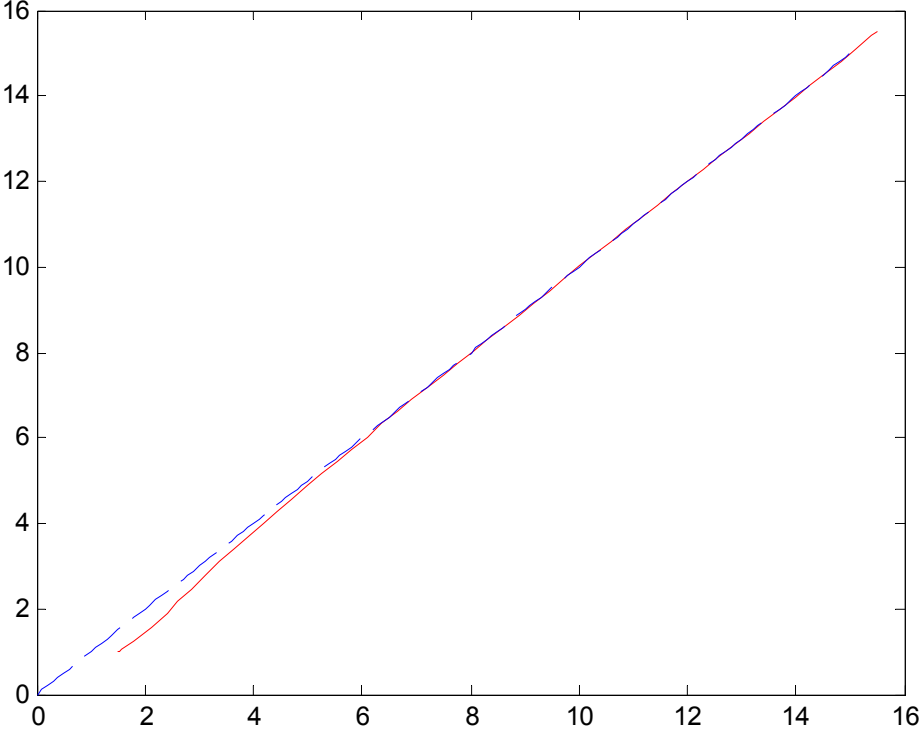
Nous choisissons plusieurs conditions initiales et nous testons la convergence du robot vers la trajectoire de référence.

Pour le suivi de trajectoire linéaire les résultats obtenus en simulation sont présentés sur les figures (3.12). On constate d'après ces figures que le robot rejoint la trajectoire de référence plus au moins rapidement suivant les conditions initiales, et qu'il n'y a pas d'erreur de position.

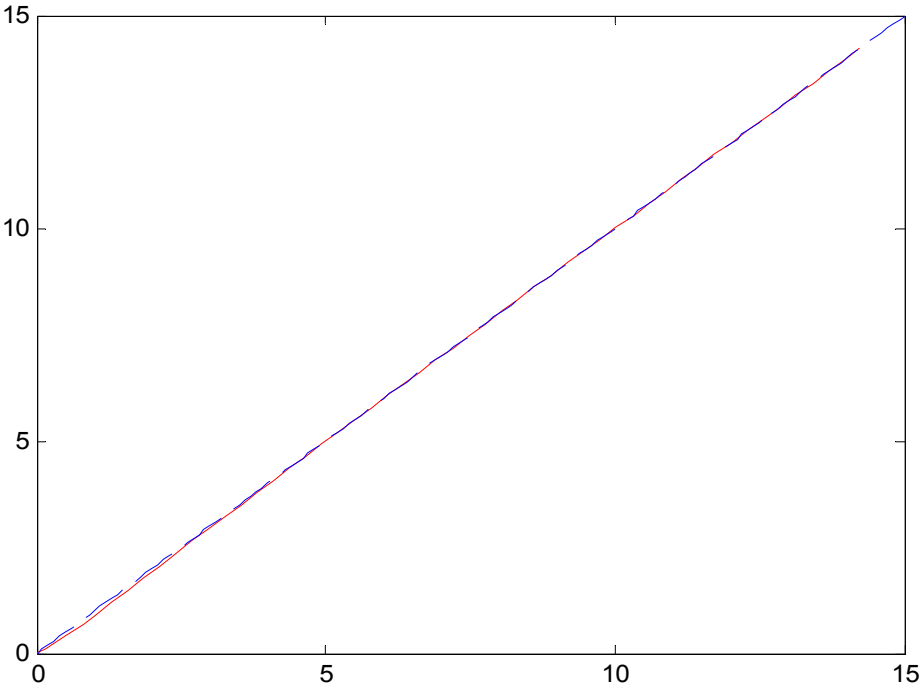
Pour le suivi de trajectoire circulaire les résultats obtenus sont présentés sur les figures (3.13). On remarque l'existence d'une erreur statique



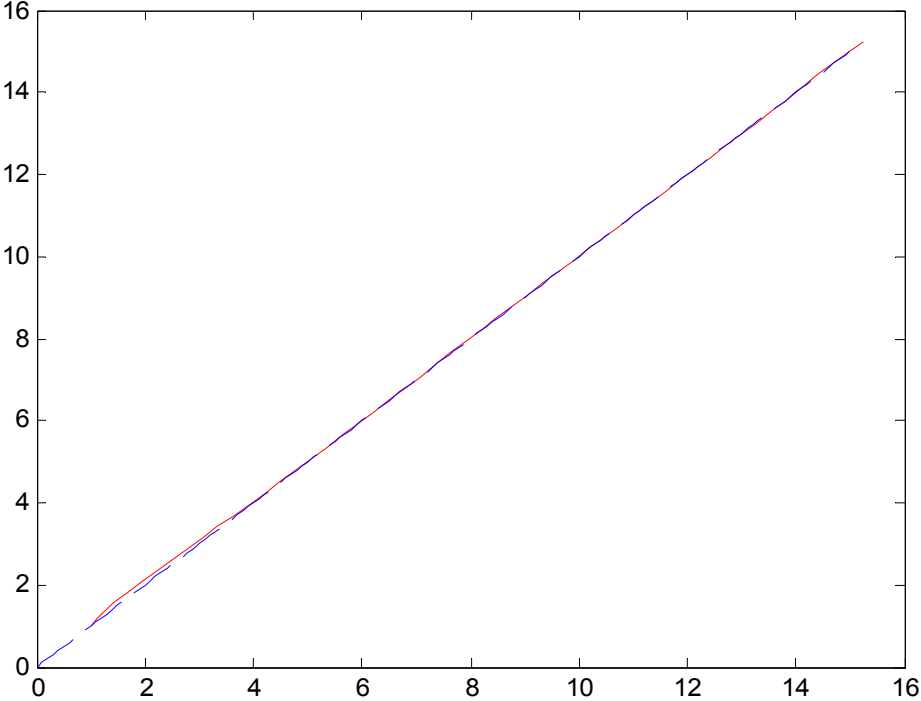
Condition initiale (1, 1.5, 0)



Condition initiale (1.5, 1,0)

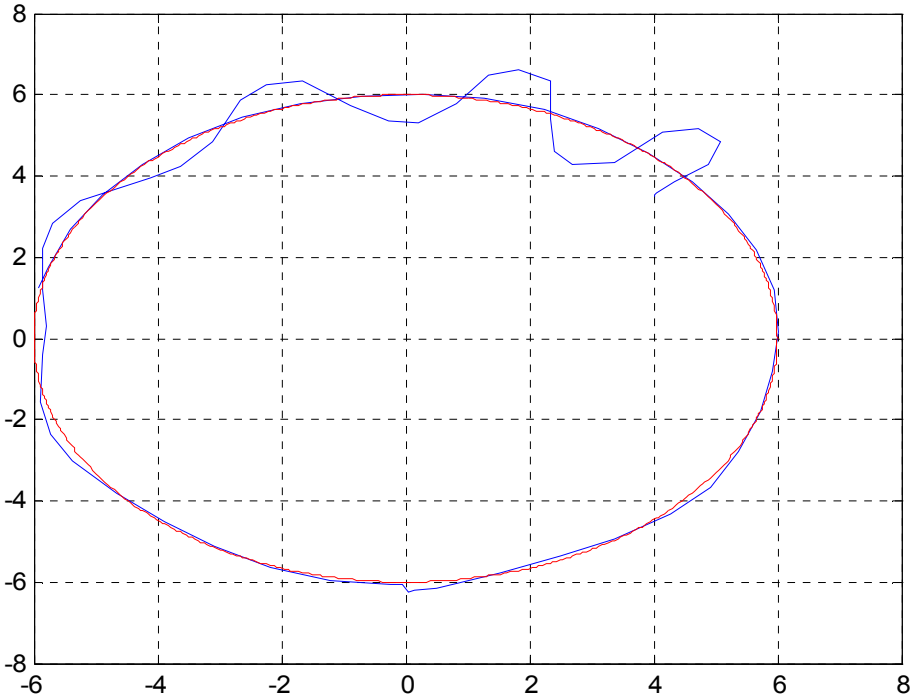


Condition initiale (0, 0,0)

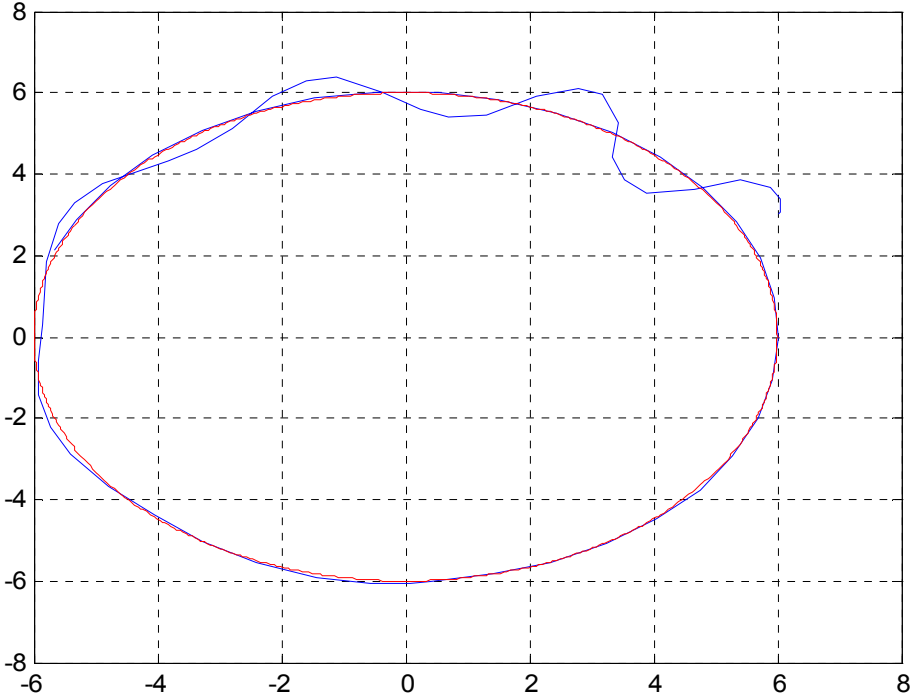


Condition initiale (1, 1, pi/2)

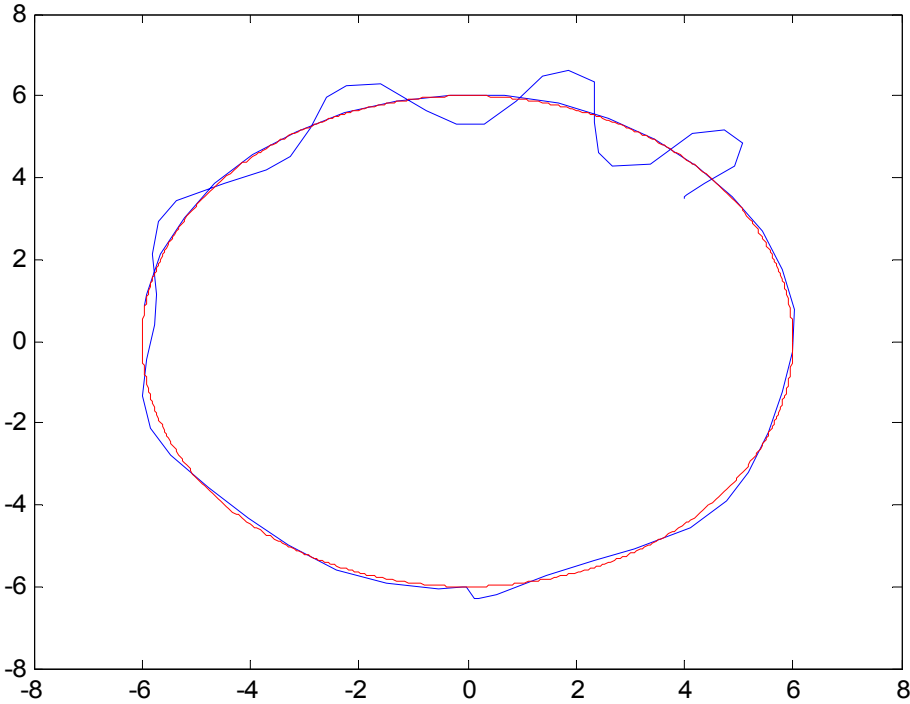
Figure 3.12 : Suivi de trajectoire linéaire



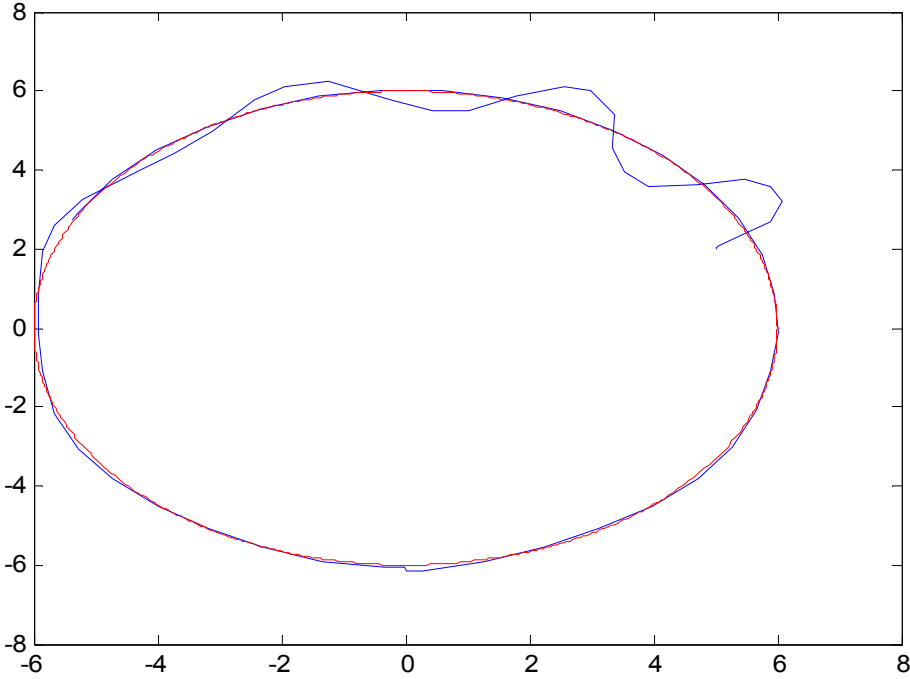
Conditions initiales :(4, 3.5, pi/2)



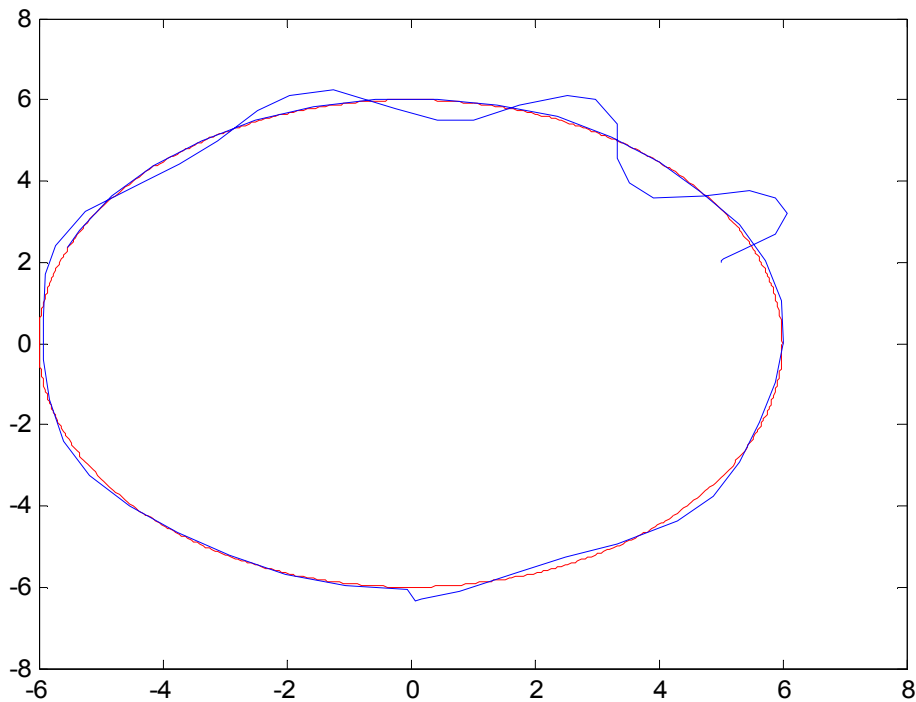
Conditions initiales : (6, 3, pi/2)



Conditions initiales :(4, 3.5, pi/4)



Conditions initiales :(5, 2, pi/4)



Conditions initiales : (5, 2, $\pi/3$)

Figure 3.13 : Suivi de trajectoire d'un cercle

3.7 Conclusion

Avec le système de commande bas niveau que nous avons présenté, nous pouvons définir une trajectoire à l'aide des segments de droite sans avoir à la laisser. Pour des milieux très contraints, une description plus fine de la trajectoire (par exemple à l'aide de cercles) peut s'avérer nécessaire. Nous avons montré que pour une trajectoire circulaire le système était encore performant.

CHAPITRE 4

MODELE ET ALGORITHME D'EVITEMENT D'OBSTACLES

4.1 Introduction

L'évitement d'obstacles est un comportement de base présent dans quasiment tous les robots mobiles. Il est indispensable pour permettre au robot de fonctionner dans un environnement dynamique et pour gérer les écarts entre le modèle interne et le monde réel. Les méthodes que nous présentons sont efficaces à condition d'avoir une perception correcte de l'environnement. Elles seront par exemple très efficaces avec un télémètre laser, mais donneront des résultats plus bruités avec des sonars. Pour limiter ce problème, il est possible d'appliquer ces méthodes sur une représentation locale (c'est-à-dire de l'environnement proche du robot et centrée sur le robot) de l'environnement qui sera construit en fonction des données de quelques instants précédents. Cette représentation intermédiaire permettra de filtrer une grande partie du bruit des données individuelles (en particulier pour les sonars).

4.2 Méthode des champs de potentiel

Dans la méthode d'évitement d'obstacles par champs de potentiels, on assimile le robot à une particule se déplaçant suivant les lignes de courant d'un potentiel créé en fonction de l'environnement perçu par le robot [KHA86]. Ce potentiel traduit différents objectifs tels que l'évitement d'obstacles ou une direction de déplacement préférée. Il est calculé par sommation de différentes primitives de potentiels traduisant chacun de ces objectifs (figure 4.1). Ces différents potentiels peuvent avoir une étendue spatiale limitée ou non (par exemple, n'avoir une influence que près des obstacles) et leur intensité peut dépendre ou non de la distance.

Le gradient de ce potentiel donne, en chaque point de l'espace, la direction de déplacement du robot (figure 4.1). Comme c'est ce gradient, et non la valeur absolue du potentiel, qui nous intéresse, il est possible de calculer directement en chaque point sa valeur par une simple somme vectorielle en ajoutant les valeurs issues des différents potentiels primitifs. Ainsi, pour un robot se déplaçant en ligne droite en espace ouvert et évitant les obstacles qu'il peut rencontrer, nous obtenons par exemple les lignes de courant illustrées sur la figure 4.2.

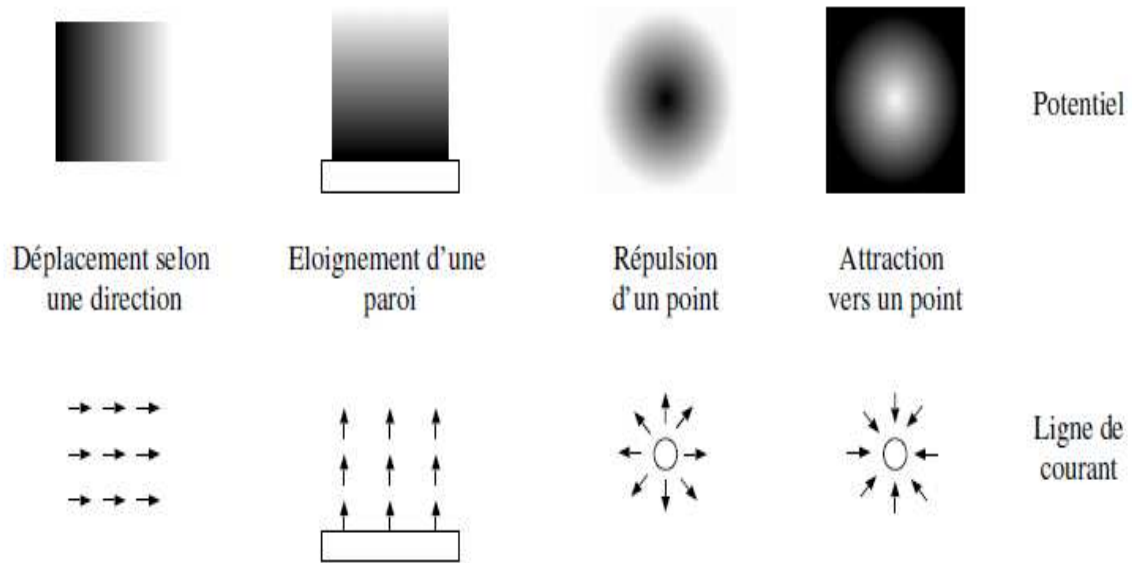


Figure 4.1: Illustration de potentiels primitifs dont la combinaison guide les déplacements du robot. Le robot se déplacera selon les lignes de courant

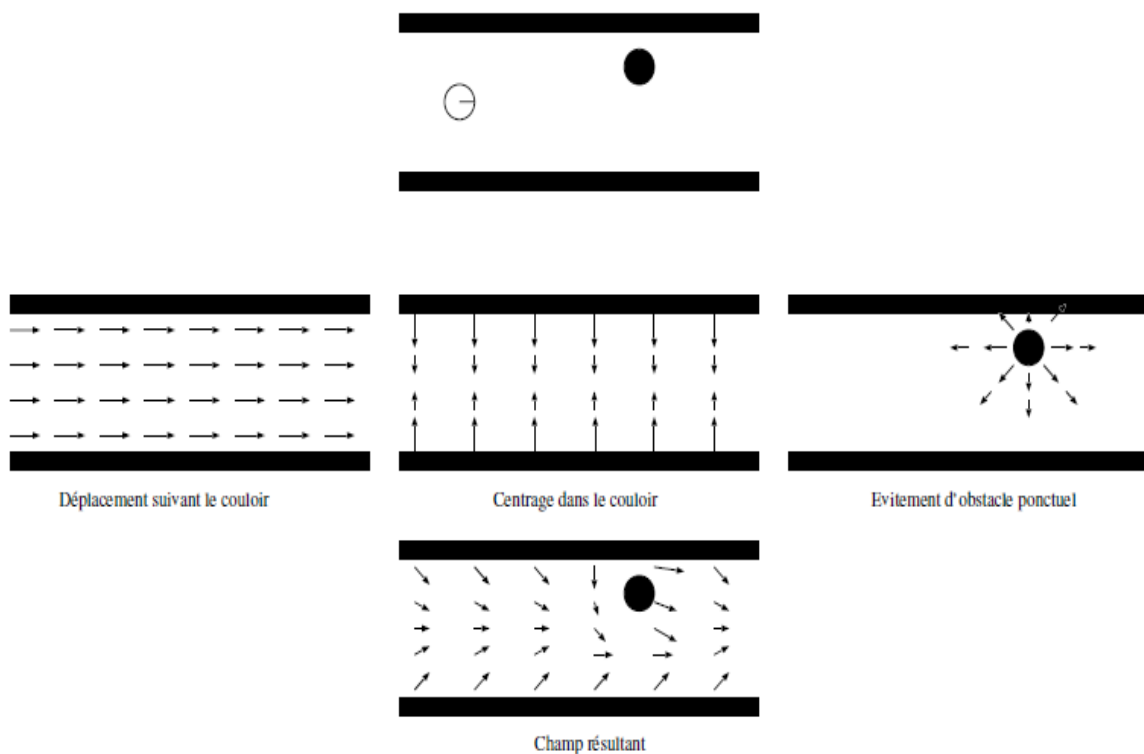


Figure 4.2 : Illustration de la combinaison de différents potentiels primitifs

De plus, dans la pratique, pour l'évitement d'obstacles, le potentiel est en général calculé dans l'espace relatif au robot et ne sert qu'à décider de la vitesse et de la direction courante. Il n'est donc nécessaire de l'estimer que pour la position courante du robot, en sommant simplement la contribution des différents éléments perçus (figure 4.3).



Figure 4.3 : Illustration de la combinaison de différents potentiels primitifs dans l'espace relatif au robot

Le principal inconvénient de cette méthode d'évitement d'obstacles est l'existence, pour certaines configurations d'obstacles (relativement courantes) de minimum locaux du potentiel qui ne permettent pas de décider de la direction à prendre (figure 4.4). Ce problème peut être traité de différentes façons. Il est par exemple possible de déclencher un comportement particulier lorsque l'on rencontre un tel minimum (déplacement aléatoire, suivi de murs ...). Il est aussi possible d'imposer que le potentiel calculé soit une fonction harmonique, ce qui garanti qu'il n'ait pas de minima, mais complexifie beaucoup son calcul.

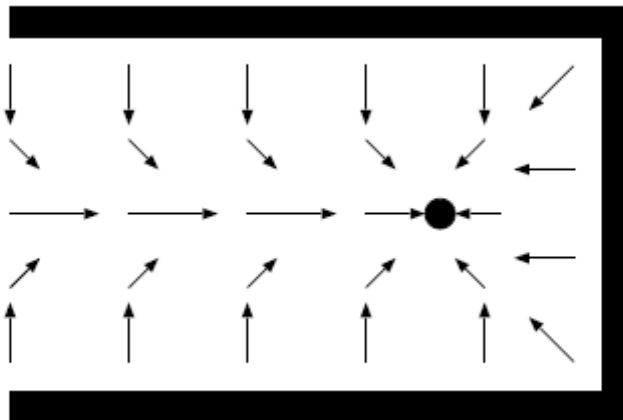


Figure 4.4 : Exemple de minimum local dans un champ de potentiel

Le principe de ces champs de potentiels est formalisé sous le nom de *schéma moteur* par R. Arkin [ARK89]. Un schéma moteur est une action définie sous forme de potentiel en fonction des perceptions du robot. Ils sont utilisés comme bas niveau dans une architecture hybride.

4.3 La méthode Vector Field Histogram

Cette méthode a été conçue spécifiquement pour utiliser une grille d'occupation locale construite à partir de capteurs à ultrasons. Cette méthode est introduite par Borenstein et Korem [BOR91]. Cette grille est construite de manière très rapide par la méthode "Histogrammic in motion mapping" qui produit une grille dont chaque cellule contient un nombre d'autant plus élevé qu'elle a souvent été perçue comme contenant un obstacle (figure 4.5).

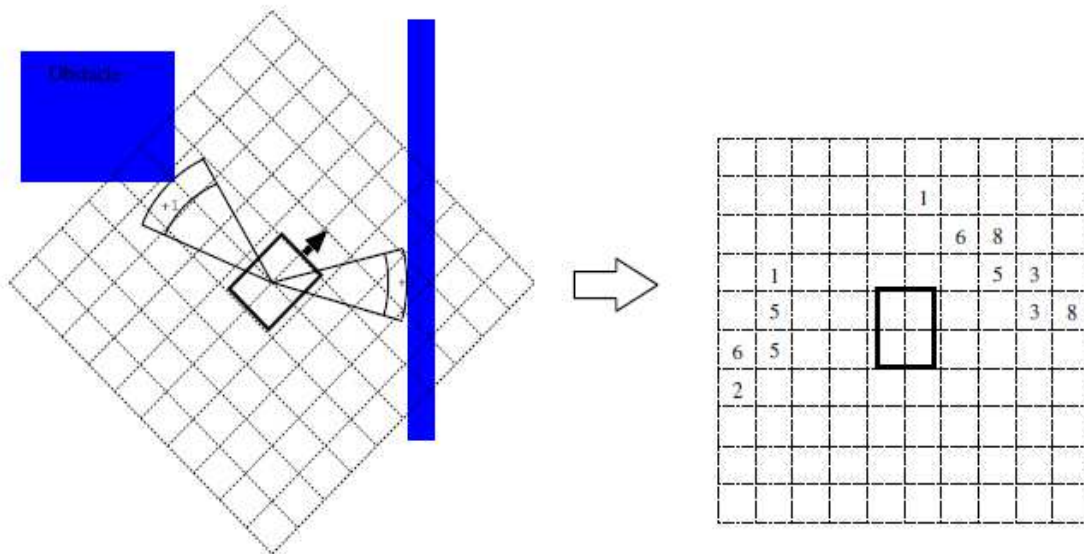


Figure 4.5 : Grille d'occupation locale construite par la méthode "Histogrammic in motion mapping"

La grille est construite dans le référentiel du robot : un compteur est incrémenté pour chaque cellule appartenant au secteur angulaire dans lequel un obstacle a été détecté et les valeurs sont déplacées d'une cellule à l'autre en fonction des déplacements du robot

Un histogramme représentant l'occupation de l'environnement autour du robot est ensuite construit à partir de cette grille d'occupation locale. Pour cela, l'environnement est discrétisé en secteurs angulaires pour lesquels la somme des valeurs des cellules est calculée (figure 4.6). Un seuil permettant de tolérer un certain bruit est ensuite utilisé pour déterminer les directions possibles pour le robot : toutes les directions dont la valeur est inférieure au seuil sont considérées.

Le choix de la direction est finalement réalisé parmi les directions possibles en fonction de contraintes externes (par exemple la direction la plus proche de la direction du but).

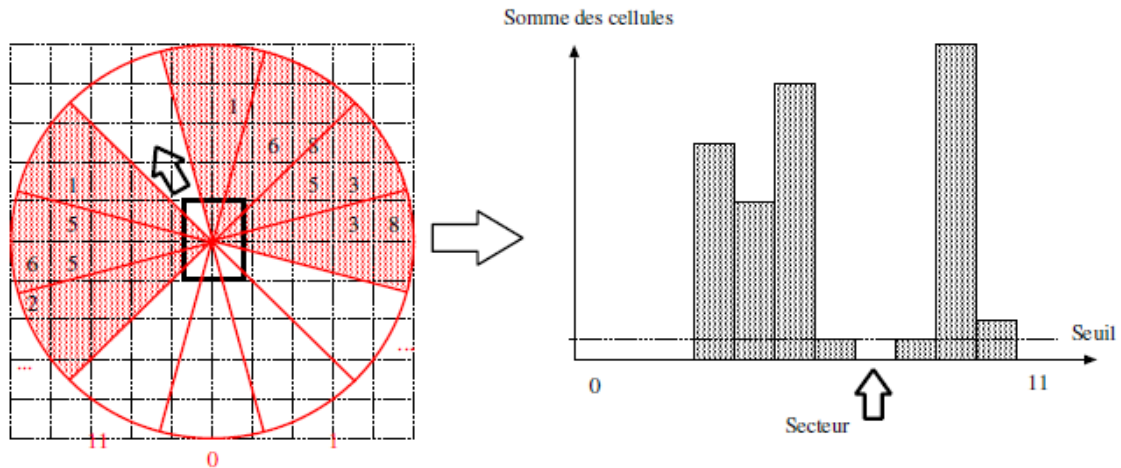


Figure 4.6 : Utilisation de l'histogramme des obstacles pour déterminer la direction de déplacement du robot

Cette méthode est extrêmement rapide et a permis historiquement un déplacement réactif à des vitesses assez élevées (environ 1 m/s). Diverses améliorations pour permettre le réglage de la vitesse du robot en fonction de la densité des obstacles sont possibles.

4.4 Méthode de la fenêtre dynamique

La méthode de la fenêtre dynamique proposée dans [BIC97] et [BIC20] permet, à partir de la perception locale de l'environnement, de sélectionner un couple (v, ω) de vitesses de translation et de rotation du robot qui répond à différentes contraintes, dont celle d'éviter les obstacles. Un tel couple de vitesses, lorsqu'il est appliqué au robot, produit une trajectoire circulaire, pour laquelle la satisfaction des différentes contraintes peut être évaluée. A l'issue de l'évaluation de toutes les contraintes pour tous les couples de vitesses possibles, la méthode de la fenêtre dynamique permet de sélectionner le couple le plus pertinent (qui répond le mieux aux contraintes).

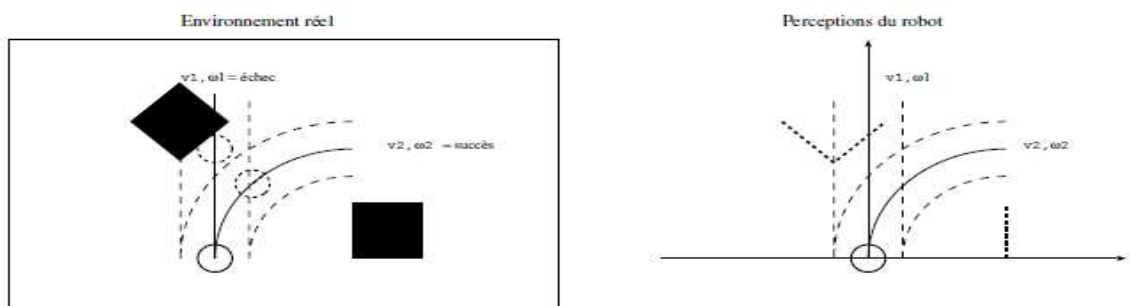


Figure 4.7 : Contrainte d'évitement d'obstacles pour la méthode de la fenêtre dynamique

La première contrainte est la contrainte d'évitement d'obstacles. C'est une contrainte dure au sens où elle est binaire (succès / échec) et doit obligatoirement être satisfaite. Elle est évaluée

pour chacune des trajectoires possibles à partir de la perception locale de l'environnement à un instant donné et de la position estimée du robot à un pas de temps fixé dans le futur pour la trajectoire courante. Si le robot n'a pas rencontré d'obstacles à cet horizon, la contrainte est respectée ; dans le cas contraire, elle ne l'est pas (figure 4.7).

Le respect ou le non respect de cette contrainte est reporté dans un graphe des vitesses qui indique, pour chaque couple de vitesses possible (donc chaque trajectoire), si le robot va ou ne va pas rencontrer un obstacle (figure 4.8). Dans ce graphe, il est alors possible de tracer la fenêtre des vitesses accessibles au prochain pas de temps à partir des vitesses courantes du robot et des valeurs d'accélération et décélération maximales. C'est cette fenêtre qui donne son nom à la méthode car elle permet de prendre en compte la dynamique du robot (à travers la capacité de freinage et d'accélération). Il reste alors à choisir, au sein de cette fenêtre, un couple de vitesses qui ne conduisent pas à percuter un obstacle pour garantir un déplacement sûr du robot.

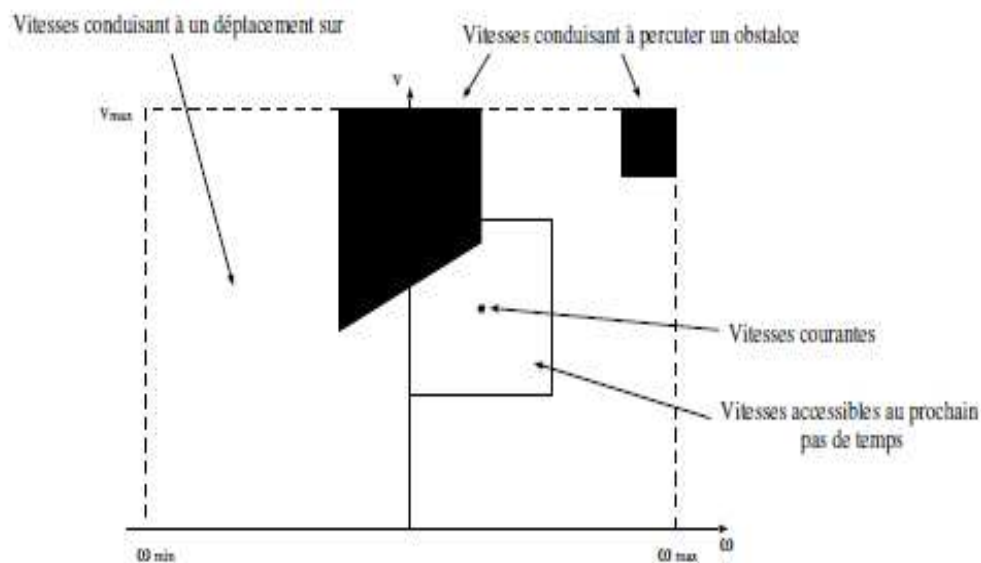


Figure 4.8 : Fenêtre de sélection des vitesses tenant compte de la dynamique du robot.

Pour faire le choix parmi toutes les vitesses possibles au sein de cette fenêtre, il est possible d'utiliser des contraintes "souples" supplémentaires pour exprimer des préférences au sein de cet espace des vitesses accessibles. Ces contraintes s'expriment par une fonction de coût $G(v, \omega)$ qui est en général la somme de plusieurs termes. Ces termes peuvent exprimer une préférence à priori sur les vitesses, une préférence pour les trajectoires s'éloignant le plus des obstacles, ou une préférence de direction si l'on dispose par exemple d'une estimation de la direction d'un but à long terme (figure 3.9). Le couple de vitesses minimisant ce coût au

sein de la fenêtre est alors sélectionné. Il garantit un déplacement sans rencontrer d'obstacles et le meilleur respect possible des contraintes souples dans ce cadre.

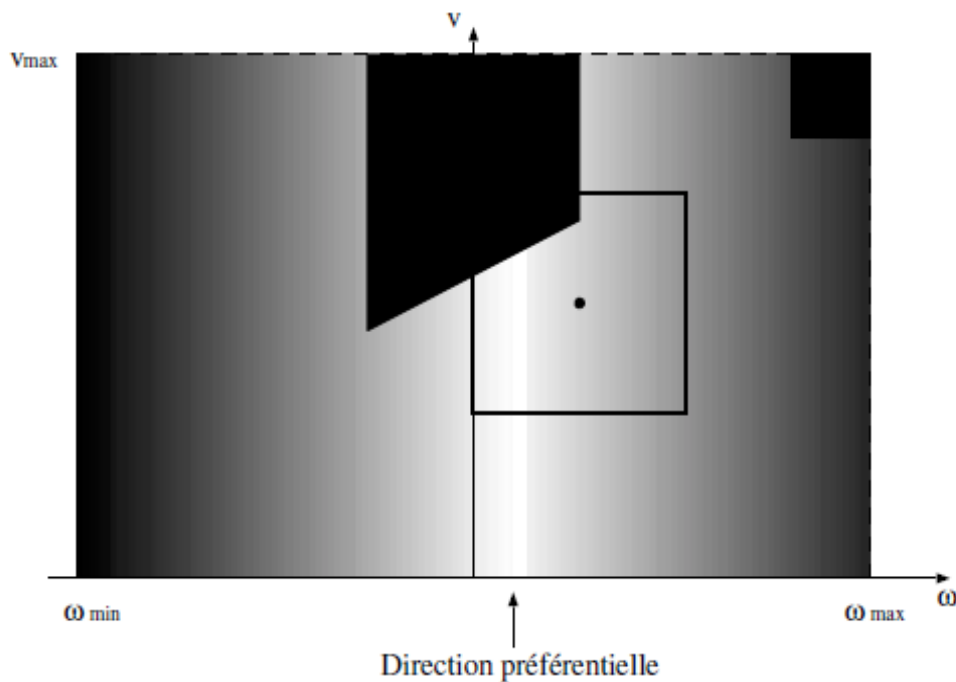


Figure 4.9 : Contrainte "souple" exprimant une préférence sur la direction à prendre

Dans la pratique, les valeurs des différentes contraintes sont évaluées en différents points du graphe des vitesses, le nombre de points dépendant notamment de la puissance de calcul disponible et de la complexité de l'évaluation de chaque contrainte. L'utilisation de la fenêtre dynamique est très intéressante pour un robot se déplaçant rapidement, ou pour un robot ayant des capacités d'accélération et de ralentissement limitées. Elle permet alors de produire un déplacement du robot sûr et régulier. Pour des robots qui ont une forte capacité d'accélération et de décélération (par exemple un robot léger avec de bons moteurs électriques), on peut considérer que toutes les vitesses sont accessibles presque instantanément. Il peut alors être suffisant de ne considérer que la cinématique, et non la dynamique, ce qui se traduit par la prise en compte d'un seul point du graphe, et non d'une fenêtre. La recherche du couple de vitesse est ainsi simplifiée.

4.5 Développement de la méthode des champs de potentiel

La planification de chemin à l'aide des champs de potentiel artificiel est basée sur un simple et puissant principe, d'abord proposée par O. Khatib dans [KHA95]. Le robot est considéré comme une particule qui se déplace prolongé dans un champ potentiel généré par le but et par les obstacles présents dans l'environnement. L'objectif génère un potentiel attractif tandis que

chaque obstacle génère un potentiel répulsif. Un champ de potentiel peut être considéré comme un domaine de l'énergie et donc son gradient, à chaque point, est une force. Le robot immergé dans le potentiel déposé est soumis à l'action d'une force qui l'anime à l'objectif (il est l'action de la force d'attraction qui résulte de la pente du potentiel attractif généré par l'objectif) tout en maintenant l'écart des obstacles (c'est l'action d'une force répulsive qui est le gradient du potentiel de répulsion généré par les obstacles).

Le mouvement du robot dans le champ potentiel peut être interprété comme le mouvement d'une particule dans un champ de vecteur gradient généré par des particules électriques positives et négatives. Dans cette analogie, le robot est une charge positive, le but est une charge négative et les obstacles sont des ensembles de charges positives. Les gradients dans ce contexte peuvent être interprétés comme des forces qui attirent le robot de particules chargées positivement à une particule négative qui agit comme le but. La combinaison de la force attractive vers l'objectif et les forces de répulsion des obstacles entraînent le robot dans une voie sûre vers le but.

Les fonctions potentielles peuvent également être considérées comme un paysage avec des montagnes (générées par les obstacles) et les vallées, avec le point le plus bas de la vallée qui représente l'objectif du robot. Le robot suit le chemin le long de la pente négative de la fonction potentielle qui désigne le mouvement de descente vers le point le plus bas dans la vallée. Avec cette analogie, il est clair que le robot peut être piégé dans des minima locaux loin de l'objectif. Soit q représentant la position du robot, considérée comme une particule se déplaçant dans une dimension d'espace R^n . Par souci de simplicité de présentation examinons le problème appliqué à un robot point mobile dans un plan, c'est à dire, $n = 2$ et que le robot est défini par $q = (x, y)$.

Le potentiel de champ artificiel où le robot se déplace est une fonction scalaire $U(q) : R^2 \rightarrow R$ générée par la superposition de potentiels attractifs et répulsifs

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (4.1)$$

Le potentiel de répulsion est le résultat de la superposition des potentiels répulsifs générés par les obstacles, et ainsi (3-1) peut être écrite comme :

$$U(q) = U_{att}(q) + \sum_i U_{rep_i}(q) \quad (4.2)$$

$U_{rep_i}(q)$ représente le potentiel répulsif généré par l'obstacle i .

Considérons maintenant que $U(q)$ est dérivable. A chaque q , la pente du champ de potentiel, notée $\nabla U(q)$, vecteur qui pointe dans la direction, augmente au maximum $U(q)$.

Concernant les champs de potentiel en fonction des méthodes de navigation du robot, le potentiel attractif est choisi pour être nul au but et à augmenter à mesure que le robot est loin du but et le potentiel répulsif, associé à chaque obstacle est très élevé (infini) dans le proche voisinage des obstacles et diminue lorsque la distance à l'obstacle diminue. Selon ces principes, différents potentiels attractifs peuvent être choisis. En outre, la force qui pousse le robot est le gradient négatif du potentiel artificiel, c-à-d :

$$F(q) = F_{att}(q) + F_{rep}(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) = \left(\frac{\partial U(x,y)}{\partial x} \quad \frac{\partial U(x,y)}{\partial y} \right)^T \quad (4.3)$$

Avec $q = (x, y)$

La force $F(q)$ dans (4.3) est un vecteur qui pointe dans la direction et à chaque q diminue localement au maximum U . Cette force peut être considérée comme le vecteur de la vitesse qui entraîne le robot.

4.5.1 Potentiel attractif

Le champ attractif peut être simplement défini sous la forme d'une *parabolique* :

$$U_{att}(q) = \frac{1}{2} \xi p_{goal}^2(q) \quad (4.4)$$

Avec ξ un scalaire positive et p_{goal} la distance euclidienne : $\|q - q_{goal}\|$. La fonction $U_{att}(q)$ est donc positive ou nulle et atteint son minimum à q_{goal} où $U_{att}(q_{goal}) = 0$.

La force F_{att} est différentiable partout dans \mathbf{C} et on a :

$$F_{att}(q) = -\nabla U_{att}(q) = -\xi p_{goal}(q) \nabla p_{goal}(q) = -\xi (q - q_{goal}) \quad (4.5)$$

Une autre forme de U_{att} peut être de type *conique* :

$$U_{att}(q) = \xi p_{goal}(q) \quad (4.6)$$

On aura alors une force de type :

$$F_{att}(q) = -\xi \nabla p_{goal}(q) = -\frac{(q - q_{goal})}{\|q - q_{goal}\|} \quad (4.7)$$

L'avantage de la forme conique sur la parabolique, c'est que la force est constante sur l'espace : elle ne tend pas vers l'infini lorsqu'on s'éloigne de q_{goal} comme la parabolique. Cependant, elle n'est pas nulle à q_{goal} .

La force F_{att} entraîne le robot vers l'objectif avec une vitesse qui diminue lorsque le robot approche du but. La force F_{att} représente une dépendance linéaire vers l'objectif, ce qui

signifie qu'elle se développe tant que q s'éloigne de l'objectif et peut engendrer une vitesse rapide du robot chaque fois qu'il est loin de l'objectif. Dans ce cas, cette force impose qu'il s'approche rapidement de l'objectif avec une grande vitesse

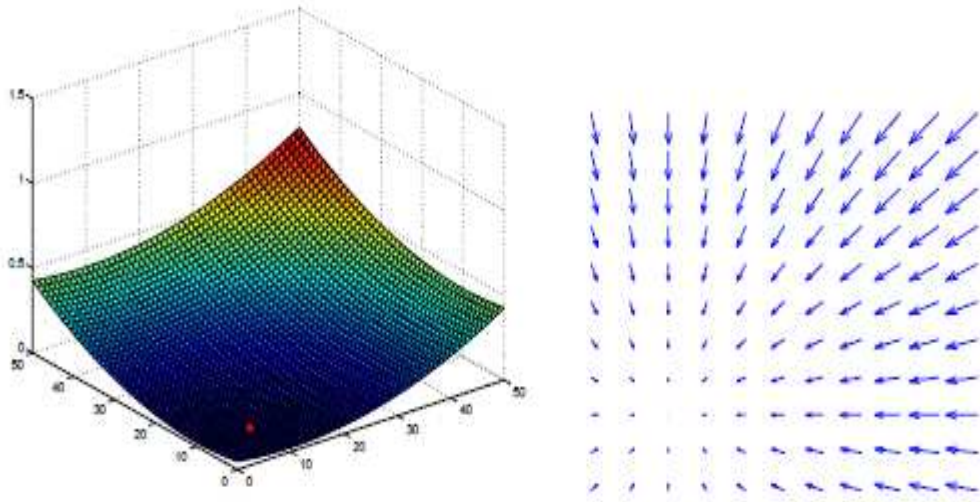


Figure 4.10 : a) potentiel attractif b) force attractive vers le but

La figure (4.10) représente le potentiel attractif et la force du gradient de champ négatif pour une situation où l'objectif (but) à (10,10) est représenté par une marque.

4.5.2 Potentiel répulsif

Le potentiel répulsif doit servir à créer une barrière de potentiel autour des obstacles (autour de C -obstacles) qui ne peut être traversé par le robot [PLU92]. De plus, on ne veut pas que ce potentiel affecte le mouvement du robot lorsque celui-ci est suffisamment loin des obstacles.

Une des formes correspondant à ces objectifs est :

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta(1/p(q) - 1/p_0) & \text{si } p(q) < p_0 \\ 0 & \text{si } p(q) > p_0 \end{cases} \quad (4.8)$$

p_0 est appelée la distance d'influence de C -obstacles. La fonction U_{rep} est positive ou nulle et tend vers l'infini lorsqu'on se rapproche de la frontière de C -obstacles. Ce potentiel de répulsion est plus fort lorsque le robot est plus proche de l'obstacle et a une basse influence lorsque le robot est loin. On a alors :

$$F_{rep}(q) = -\nabla U_{rep}(q) \quad (4.9)$$

$$F_{rep}(q) = \begin{cases} \eta(1/p(q) - 1/p_0)1/p^2(q)\nabla p(q) & \text{si } p(q) < p_0 \\ 0 & \text{si } p(q) > p_0 \end{cases} \quad (4.10)$$

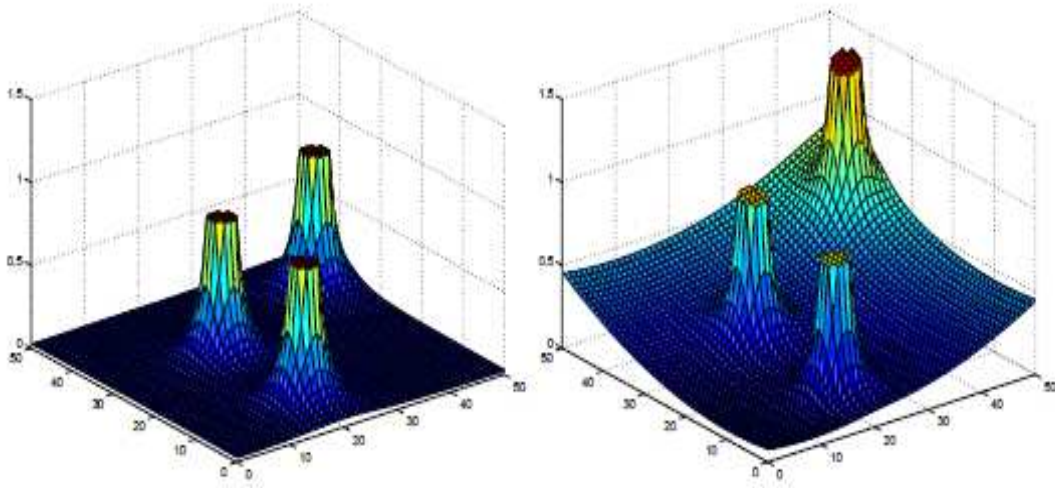


Figure 4.11 : a) potentiel répulsif b) potentiel répulsif + attractif

Le potentiel répulsif pour trois obstacles est représenté sur la figure 4.11-a tandis que la somme des potentiels attractif et répulsif est représentée à la figure 4.11-b.

4.5.3 Planification de trajectoires guidées par champs potentiels :

Une fois définis les potentiels, plusieurs techniques sont possibles. Les deux plus courantes sont, pour des espaces de configuration à faibles dimensions (inférieur à cinq) :

- Une descente de gradients,
- Une recherche du chemin de moindre coût.

La descente de gradient consiste simplement à suivre la direction indiquée par la force F , et à avancer dans cette direction d'un pas de longueur δ_i . Par exemple, avec $q = (x, y)$, on aura :

$$\begin{cases} x(q_{i+1}) = x(q_i) + \delta_i \partial U(x, y) / \partial x \\ y(q_{i+1}) = y(q_i) + \delta_i \partial U(x, y) / \partial y \end{cases} \quad (4.11)$$

L'algorithme s'arrêtera lorsqu'on sera à une distance de q_{goal} inférieure à δ_i , dans le cas où un chemin est déterminé. L'inconvénient de cette méthode est que l'on peut tomber sur un minimum local. Le critère d'arrêt sera :

- Soit que la distance entre la position q_{goal} et q_{robot} soit inférieure au pas delta
- Soit un nombre d'itérations maximal très grand

La recherche du chemin de moindre coût consiste à échantillonner l'espace en une grille fine, puis à appliquer un algorithme de propagation, appelé Best-First Planning (BFP)

4.6 Résultats de simulation

Dans nos travaux de simulation, on considère des cas simples où l'espace de configuration est limité à deux dimensions et où les obstacles sont réduits à des points. On utilise un espace de configuration de taille 30x30 où l'on va disposer des obstacles ponctuels.

A- Présence d'un seul obstacle :

On va commencer notre simulation en présence d'un seul obstacle, avec la variation du paramètre η on verra les performances de cette technique :

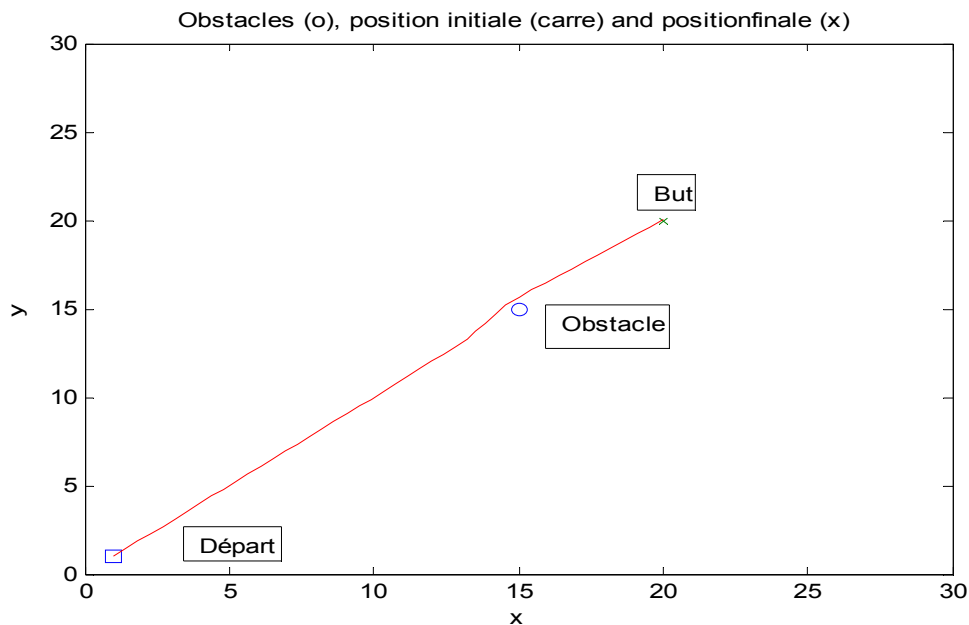


Figure 4.12 : Planification de chemin en présence d'un seul obstacle avec $\eta=1.3$

D'après la figure ci-dessus on remarque que le robot se dirige vers le but avec une trajectoire de forme rectiligne qui a pratiquement l'allure d'une droite et cela suite à la commande imposée. Cependant quand il arrive à l'obstacle, il dévie avec un certain angle de sa trajectoire pour éviter la collision avec l'obstacle puis il suit sa trajectoire vers le but sous forme d'une droite. Ce comportement répond bien à l'influence de la technique proposée.

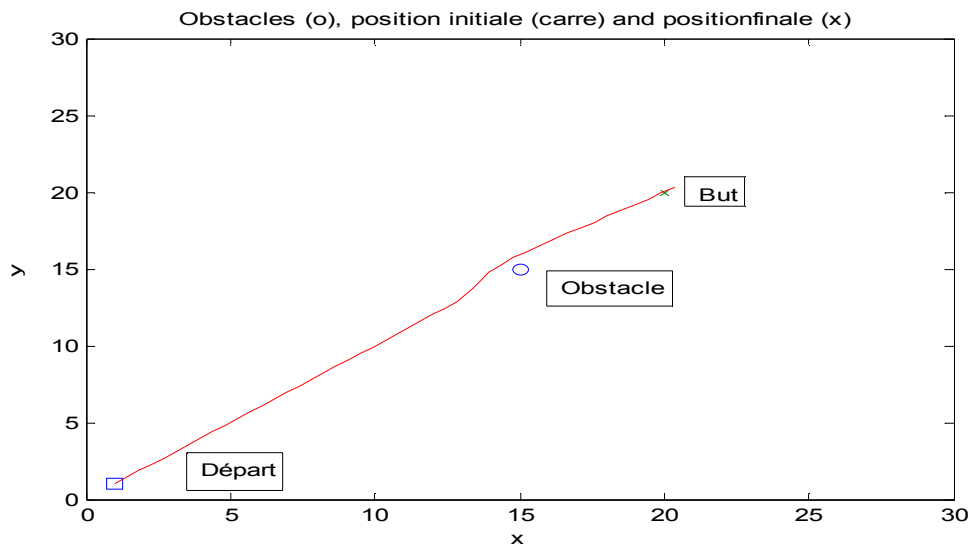


Figure 4.13 : Planification de chemin en présence d'un seul obstacle avec $\eta=3$

D'après les deux figures ci-dessus on constate que l'angle de déviation dépend de la constante de répulsion η (il augmente avec l'augmentation de η).

Visualisation du potentiel attractif :

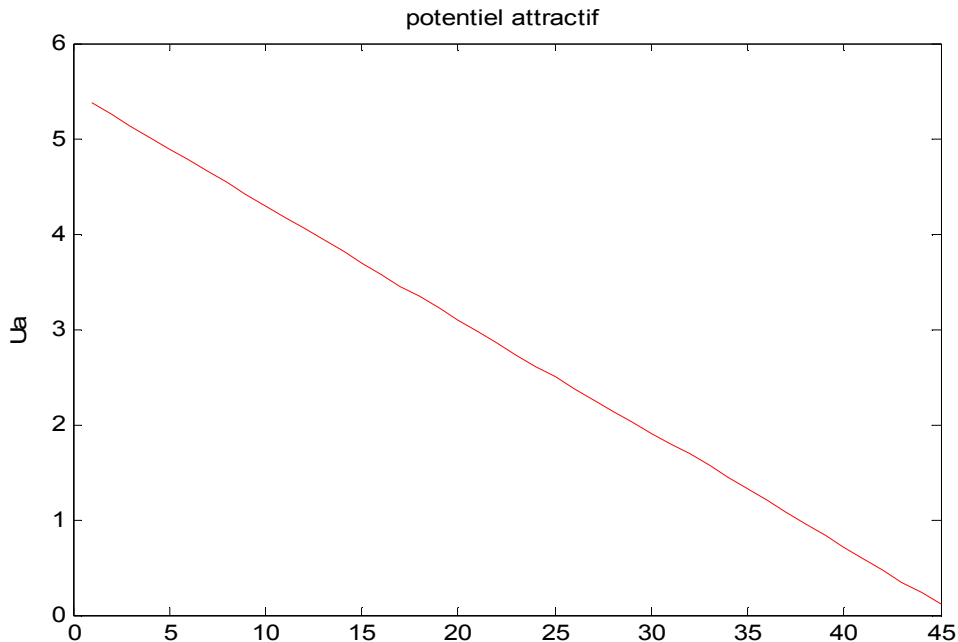


Figure 4.14 : Potentiel attractif

D'après la figure ci-dessus on remarque que le potentiel attractif diminue quand le robot s'approche du but et il est nul quand le robot arrive au but.

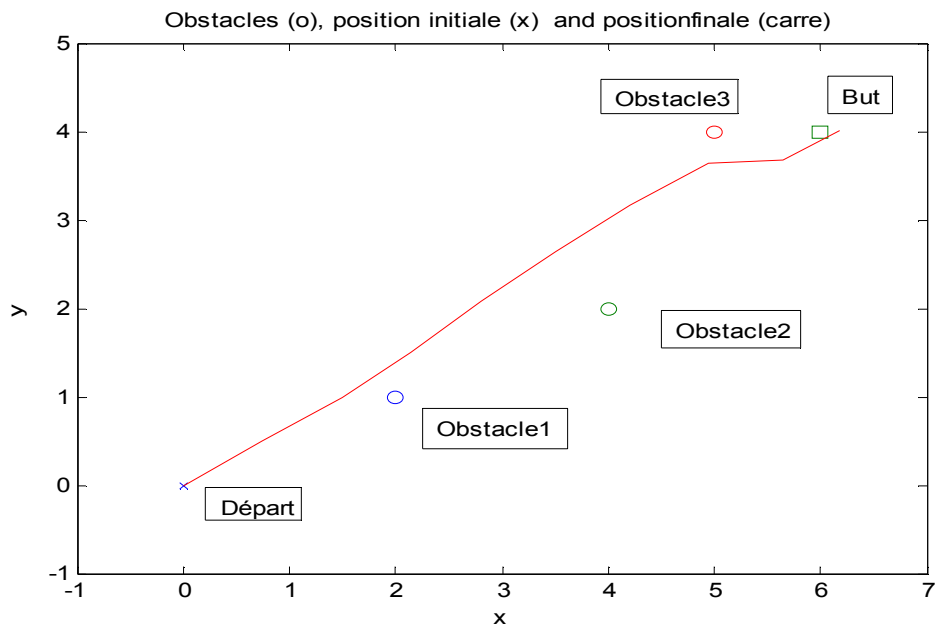
B- Présence de trois obstacles :

Figure 4.15 : Planification de chemin en présence de trois obstacles

A partir de la figure ci-dessus on constate que le robot suit une trajectoire optimale pour arriver au but sans collision avec les trois obstacles, ce qui interprète l'efficacité de cette technique.

Visualisation du potentiel attractif :

D'après la figure ci-dessous on voit que le potentiel attractif tend vers zéro au voisinage du but.

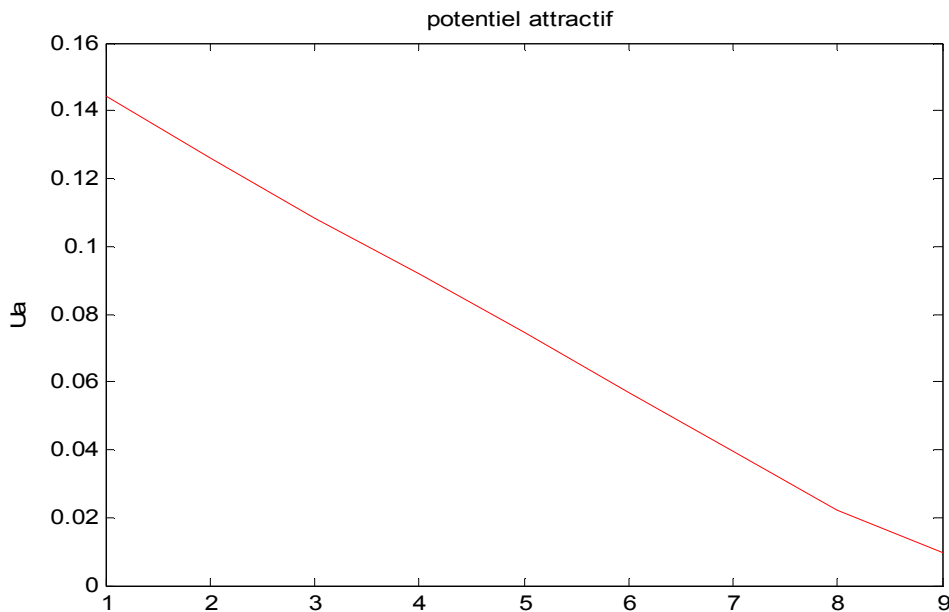


Figure 4.16 : Potentiel attractif

C- Présence de plusieurs obstacles :

On suppose que notre espace de travail contient six obstacles, ces derniers gênent le robot mobile dans son évolution du point de départ à l'arrivée au but. On fait varier la constante de répulsion η et on voit l'influence de la technique des champs potentiels artificiels.

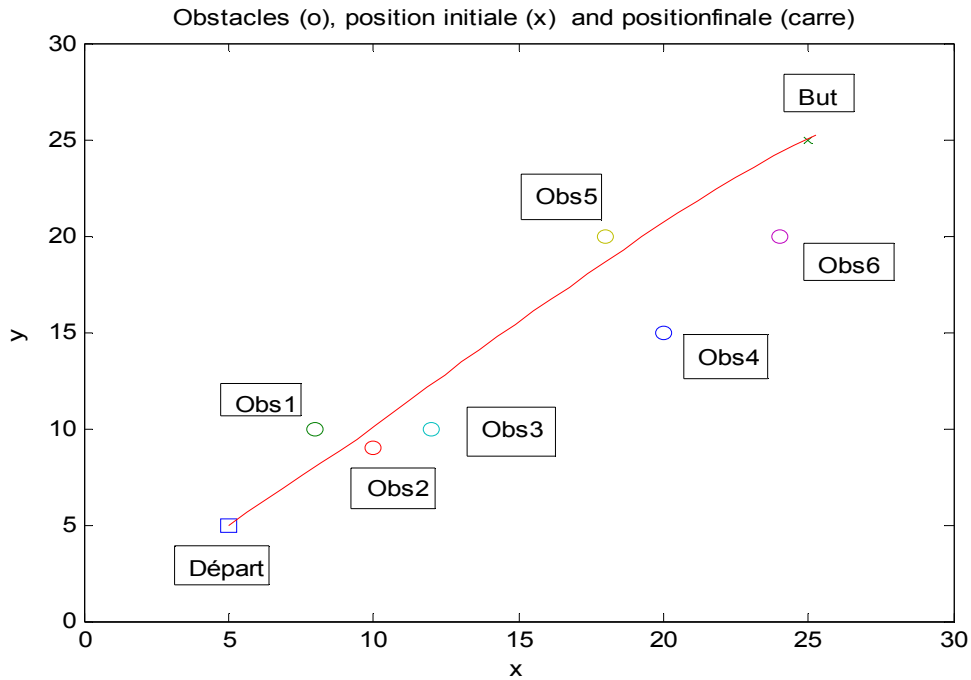


Figure 4.17 : Planification de chemin en présence de 6 obstacles avec $\eta=0.07$

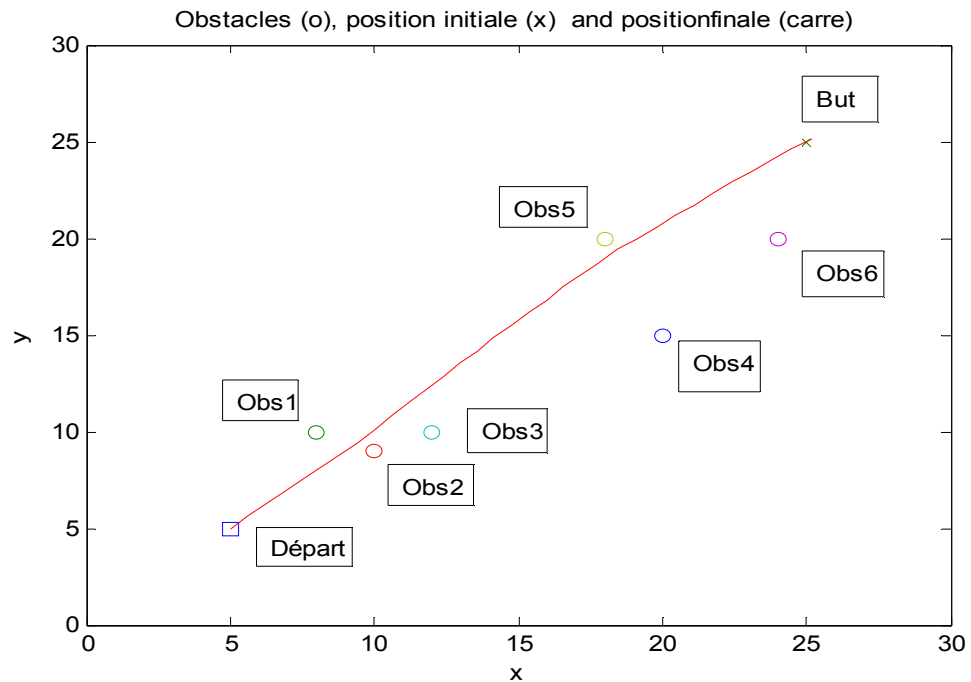


Figure 4.18 : Planification de chemin en présence de 6 obstacles avec $\eta=0.1$

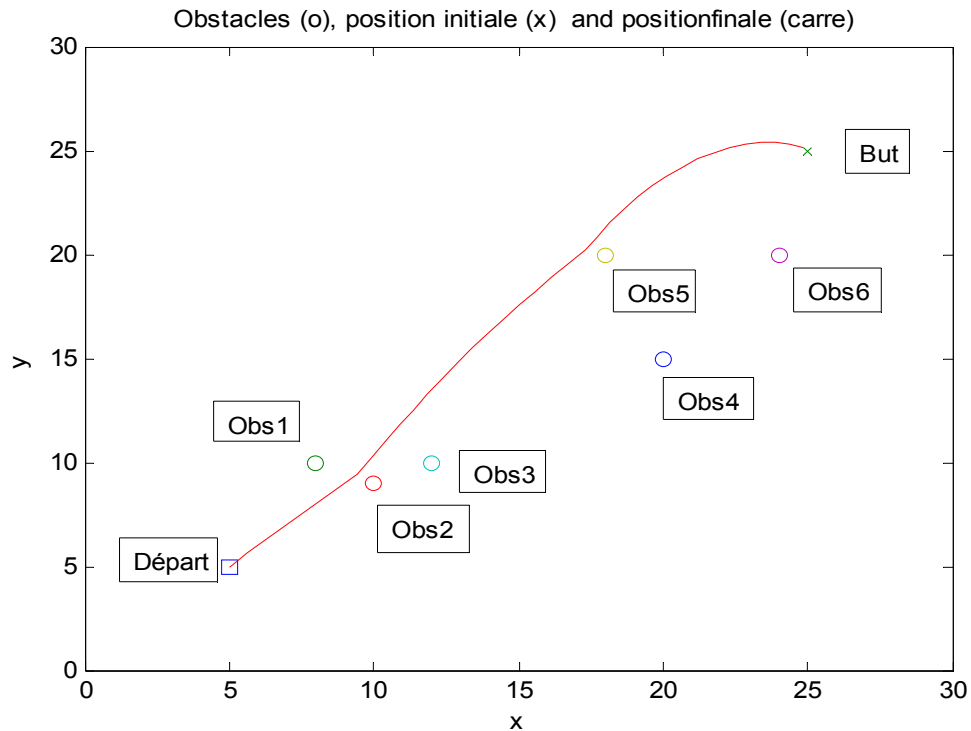


Figure 4.19 : Planification de chemin en présence de 6 obstacles avec $\eta=0.3$

Les figures ci-dessus montrent quelques trajectoires qu'on a obtenues avec différentes valeurs du paramètre η . Une trajectoire optimale du robot mobile est obtenue pour $\eta = 0.07$.

Pour $\eta = 0.1$, on obtient une trajectoire pareille à celle de la figure 4.17 avec une légère différence au niveau de l'obstacle numéro cinq.

Pour $\eta = 0.3$, une grande différence est constatée. Le robot mobile évite carrément tous les obstacles et tend vers le but mais suivant une trajectoire quelconque.

4.7 Conclusion

Nous avons proposé ici une méthode de navigation d'un robot mobile basée sur la technique des champs de potentiels. La planification de trajectoires avec évitement d'obstacles d'un robot guidé par des champs de potentiels, qui est présentée ici, est une voie simple de technique de planification qui a un principe de fonctionnement intuitif basé sur les champs de type d'énergie. Pour un environnement statique et complètement connu, le potentiel peut être évalué hors ligne, cela nous permet d'obtenir le profil de vitesse à appliquer à un robot mobile pour se déplacer d'un point de départ à un point d'arrivée par l'utilisation de l'énergie sur le terrain. En outre, cette technique peut être appliquée en temps réel sur une version qui englobe un élément d'évitement d'obstacles. Le potentiel répulsif peut

être le résultat de l'environnement des obstacles connus qui sont présentés dans un plan à priori ou des obstacles qui sont détectés par les capteurs du robot lors de sa motion.

Tous les travaux de simulation entrepris ont montré l'efficacité de cette méthode des champs de potentiel quant à un éventuel évitement d'obstacles. Les résultats obtenus montrent bien ces évitements. De plus pour un environnement connu et structuré, cette méthode peut être utilisée en temps réel par une algorithmique appropriée.

Dans sa version la plus simple, la méthode du champ de potentiel est très efficace et rapide et propose des résultats avec un minimum de calculs, cependant elle présente quelques inconvénients, en particulier la sensibilité aux minimas locaux qui surviennent généralement suite à la symétrie de l'environnement, aux obstacles concaves et aux oscillations du robot lors de la traversée d'espaces réduits.

CHAPITRE 5

SYSTEME FLOU POUR LA NAVIGATION D'UN ROBOT MOBILE

5.1 Introduction

A l'heure actuelle le développement de techniques pour la navigation autonome d'un robot mobile constitue l'un des problèmes majeurs en robotique mobile. Un problème important dans la navigation autonome est l'incertitude des informations. La logique floue est un outil adéquat pour traiter ce genre de problème [HUH02]. Les méthodes actuelles basées sur le champ de potentiel souffrent de plusieurs limitations dont la difficulté à trouver des chemins : cela peut conduire le robot de tomber dans des minima locaux et suivre la mauvaise voie [BOR89]. Une méthode de décomposition graphique donne toujours un chemin mais elle nécessite une parfaite connaissance de l'environnement et offre une trajectoire qui n'est pas facile à suivre [YAN05]. Un autre problème avec ces algorithmes classiques est de ne pas utiliser une technique d'inférence rapide et dans la plupart des cas l'environnement est incertain ou incomplet en raison des bruits.

Afin de répondre à ces problèmes, l'idée principale du développement d'un système flou qui prend un chemin sur la base des informations; qui peut être ou ne pas être complet et/ou précis et de l'adapter à un environnement donné ; en utilisant des appareils de détection limités. Généralement ces systèmes utilisent des capteurs à ultrasons. Ces capteurs n'ont pas une capacité de détection précise.

La logique floue est le meilleur moyen de convertir les stratégies de contrôle du langage naturel qui sont utilisées par l'homme à une forme utilisable par les machines. Des expériences ont montré qu'un contrôleur classique donne parfois même de meilleurs résultats que l'opérateur humain [FAT05]. Le principal avantage de la logique floue est qu'elle permet aux experts humains d'exprimer des concepts. Ce qui entraîne un gain de temps et d'espace pour une recherche dans les règles pour l'exécution d'une situation donnée.

Un autre avantage de la logique floue est le fait que le calcul de la charge des systèmes d'inférence floue est considérablement plus léger que ceux de détection de bord ; grilles de certitude et des champs de potentiel. Le succès de la gestion de l'incertitude et/ou l'incomplétude des données est un attribut de la logique floue fondée sur des moteurs d'inférence.

Le principal problème de chemin de routage des robots mobiles est de mettre le robot à partir du point A vers B sans collision avec des obstacles, en respectant les contraintes temporelles. Dans cette étude nous décomposons ce problème en deux sous problèmes :

- ❖ Trouver un chemin vers la cible : c'est un problème global mais la plus simple solution est de trouver un chemin à partir d'une position de départ à une donnée.
- ❖ Eviter les obstacles : cela peut souvent être résolu en utilisant uniquement l'information locale.

Le problème à résoudre dans ce travail nécessite une architecture réactive pour que le robot réagisse en temps réel. Nous avons subdivisé le système de navigation en trois modules faciles à gérer afin d'assurer au mieux la navigation autonome du robot et mettre l'accent sur l'exécution de tâches spécifiques. Nous proposons également un module de détection d'obstacles, un contrôleur de navigation libre et un contrôleur d'évitement d'obstacles capables de surmonter les incertitudes présentées par le monde réel et d'aider le robot à suivre de meilleurs et rapides chemins. Cette subdivision est une approche efficace, car elle rend le système modulaire, ce qui simplifie la solution de navigation; ainsi qu'elle offre la possibilité d'ajouter de nouveaux comportements au système sans provoquer une augmentation de la complexité. Dans ce travail, notre objectif est de démontrer la faisabilité de la logique floue pour l'évitement d'obstacles d'un robot mobile en utilisant un minimum de matériel.

5.2 La logique floue en robotique

La commande floue a montré depuis plusieurs années sa fiabilité et sa capacité d'adaptation aux problèmes industriels concrets. Les exemples d'utilisation concluant de la logique floue sont très nombreux. La navigation avec évitement d'obstacles par la logique floue pour un robot mobile omnidirectionnel est présentée dans [ZAV00]. Ce travail décrit un système de navigation intelligent pour un robot mobile omnidirectionnel basé sur la logique floue. Ce système réactif est particulièrement adapté pour les applications en temps réel, il permet l'évitement d'obstacles.

La navigation autonome basée sur la logique floue pour un véhicule est proposée dans le travail suivant. Une double couche de contrôle flou a été conçue pour la navigation autonome d'un véhicule dans un environnement 2D. La première couche du contrôleur floue permet de détecter la possibilité de collision avec les obstacles. La deuxième couche se compose du contrôleur principal qui effectue l'évitement de collision en temps réel.

La navigation d'un robot mobile basée sur le flou est proposée dans le travail suivant. Dans cette étude, la logique floue est utilisée pour la conception des comportements individuels et l'action de coordination des comportements.

5.3 Architecture structurelle du système

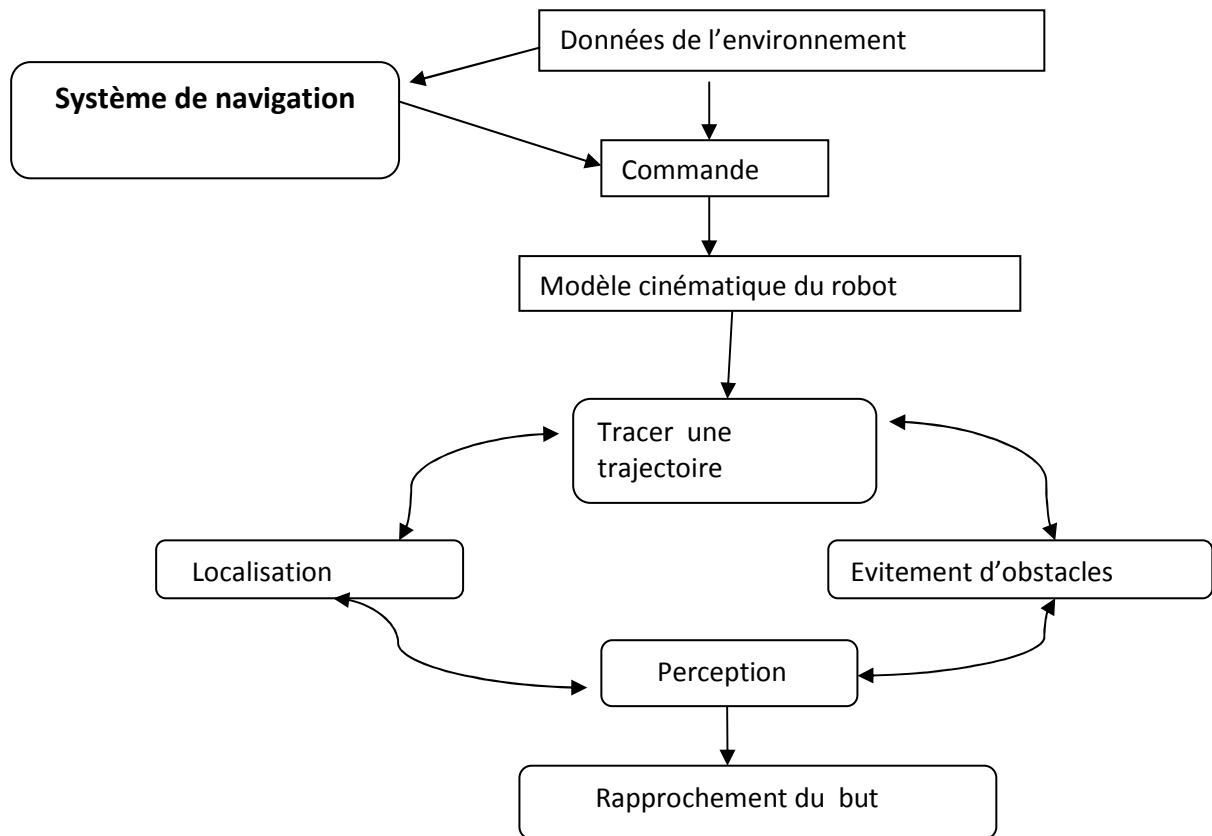


Figure 5.1 : Architecture structurelle du système de navigation proposé

Le problème à résoudre dans ce travail nécessite un architecteur réactif pour que le robot réagisse en temps réel afin d'éviter la collision avec les obstacles détectés. Le système de navigation autonome reçoit les données de l'environnement et génère la commande adéquate à une situation donnée. Le modèle cinématique exploite la commande pour tracer la trajectoire nécessaire pour le rapprochement du but. La figure 5.1 illustre l'architecture structurelle de notre système ainsi que les fonctionnalités pour que le robot exécute sa tâche de navigation autonome qui sont la perception, l'évitement d'obstacles, le rapprochement du but et la localisation.

5.4 Comportement de navigation basé sur la logique floue

La stratégie de navigation de base est réactive, combinant trois comportements de base afin d'assurer la navigation autonome du robot : la navigation libre, l'évitement d'obstacles fixes et le rapprochement du but. Cette subdivision a pour objectif de faciliter et d'assurer au mieux la navigation autonome du robot.

La figure 5.2 présente la structure de base du système de navigation composée d'un module de détection d'obstacles, d'un contrôleur flou de navigation libre et d'un contrôleur flou d'évitement d'obstacles. Cette structure permet de déterminer en temps réel l'angle de braquage à assigner au robot pour lui permettre de naviguer dans un environnement structuré et parsemé d'obstacles.

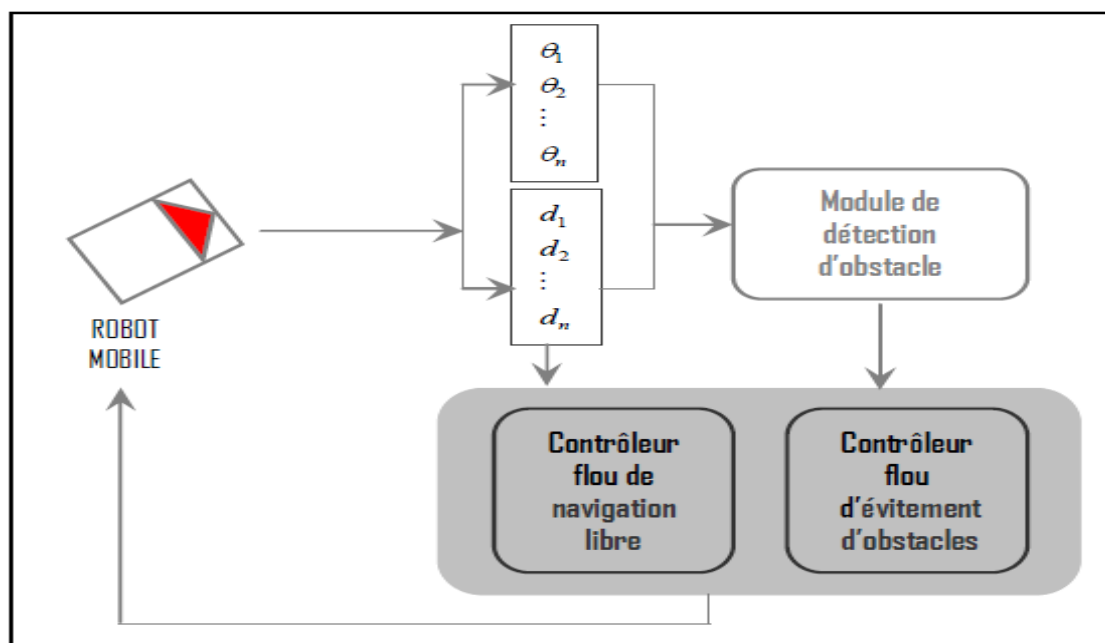


Figure 5.2 : Structure de base du système de navigation proposé

5.4.1 Module de détection d'obstacles

Le robot utilise des capteurs pour observer autour de lui. Nous avons limité le champ global de perception du robot à $\pi/2$. Le champ global de perception est décomposé en trois secteurs de détection. Chaque secteur comporte un capteur ultrason pour obtenir une plus grande précision relative à la position des obstacles par rapport à notre robot et pour approcher le comportement et les capacités de notre robot à ceux de cas réel. Les secteurs de détection sont :

Latéral _ Négatif : champs latéral négatif du robot ;

Dans _ Avant : champs avant du robot ;

Latéral _ positif : champs latéral positif.

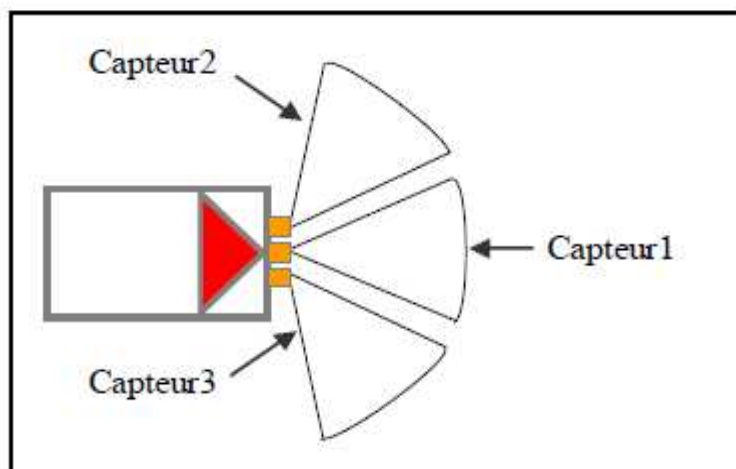


Figure 5.3 : Secteurs de détection du robot

Le robot récupère les informations relatives à l'état actuel de l'environnement (θ l'angle entre l'axe portant le point obstacle et le centre de repère et l'axe du robot), puis il réalise un balayage de façon continue afin déterminer en temps réel les obstacles détectés dans les différents champs de perception qui sont latéral _ Négatif , Dans _ Avant et Latéral _ Positif.

5.4.2 Comportement de navigation libre

Si le robot ne reçoit aucune information de ses capteurs au cours de son déplacement alors il considère que le chemin est libre, c'est-à-dire aucune présence d'obstacle; donc dans cette situation il se dirige vers la cible. Nous avons choisi un contrôleur de type Mamdani dont la conclusion est un ensemble flou. Ce type de contrôleur est généralement le plus utilisé dans ce type de problème. Ce modèle présente l'avantage d'être facilement interprétable et bien adapté à notre application. Il reçoit comme entrées θ' l'angle entre la cible et le robot et renvoie en sortie la commande φ qui correspond au changement de direction du robot (l'angle de braquage). La figure 5.4 représente la structure du contrôleur de navigation libre.

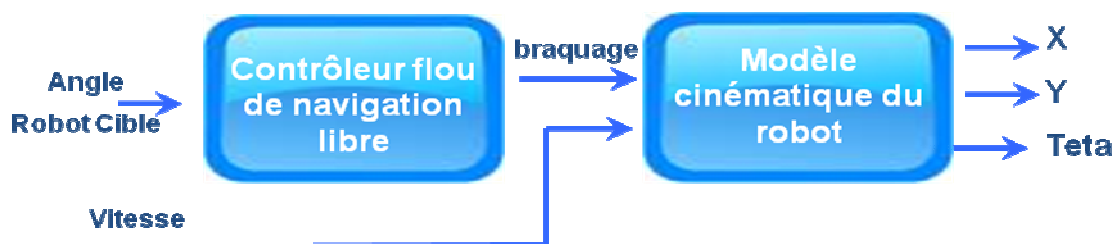


Figure 5.4 : Structure du contrôleur de navigation libre

Dans ce qui suit nous allons décrire brièvement la structure de base de notre contrôleur flou qui comporte trois parties : la fuzzification, l'inférence et la défuzzification.

5.4.2.1 Fuzzification

Les valeurs délivrées par les capteurs du robot sont traduites en labels de sous ensemble flous caractérisant les variables linguistiques associées.

Fuzzification de la variable angle robot _ cible

La variable d'entrée, l'angle entre le robot et la cible θ' , possède cinq variables floues (NG : Négatif Grand ; NP : Négatif Petit ; Z : Zéro ; PP : Positif Petit ; PG : Positif Grand) définies comme suit :

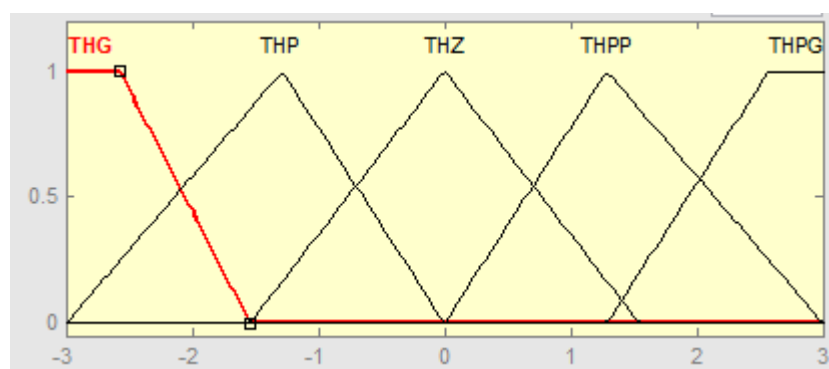


Figure 5.5 : Fonction d'appartenance de l'angle robot _ cible

Fuzzification de la variable angle de braquage

La variable de sortie qui est l'angle de braquage φ possède cinq variables floues (NG : Négatif Grand ; NP : Négatif Petit ; Z : Zéro ; PP : Positif Petit ; PG : Positif Grand) définies comme suit :

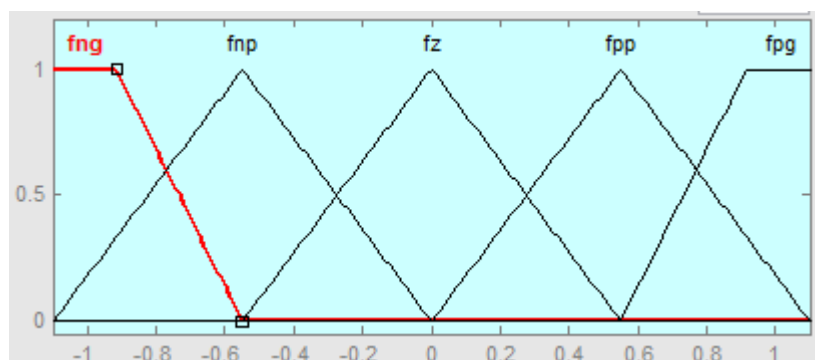


Figure 5.6 : Fonction d'appartenance de l'angle de braquage

5.4.2.2 Inférence

Un contrôleur flou prend généralement la forme d'une série de règles « Si- alors ». L'inférence est basée sur des opérations min et max afin d'effectuer l'inférence des règles représentées dans les tableaux 5.1 et l'opérateur max pour l'agrégation des règles. La réalisation de la navigation libre est réalisée par un ensemble de règles floues de la forme :

Si θ' est NG alors φ est NG

Si θ' est PG alors φ est PG

Si θ' est zéro alors φ est zéro

θ'	NG	NP	Zéro	<i>PP</i>	<i>PG</i>
φ	NG	NP	zéro	<i>PP</i>	<i>PG</i>

Tableau 5.1 : Les règles d'inférence de la navigation libre

5.4.2.3 Défuzzification

La méthode de defuzzification la plus utilisée est celle de la détermination du centre de gravité de la fonction d'appartenance résultante. Cette méthode est la plus utilisée car elle est caractérisée par un simple calcul très rapide.

5.4.2.4 Résultats expérimentaux

Nous présentons ici quelques exemples d'expérimentations réalisées. L'environnement est une salle carrée. Le robot est un rectangle qui se déplace selon le modèle cinématique mentionné dans le chapitre deux. La cible est représentée comme un drapeau dont les coordonnées sont prédéfinies. Plusieurs formes d'obstacles peuvent être simulées à partir d'une forme cylindrique de base.

Nous avons utilisé l'environnement Matlab-Simulink qui est un environnement de travail complet, puissant et performant.

Après la spécification du point but, dans le cas où le robot est initialement orienté vers sa destination, le robot effectue une pure translation à partir de sa position initiale. Les figures 5.7 et 5.8 illustrent ces déplacements quel que soit la position du robot et de la cible.

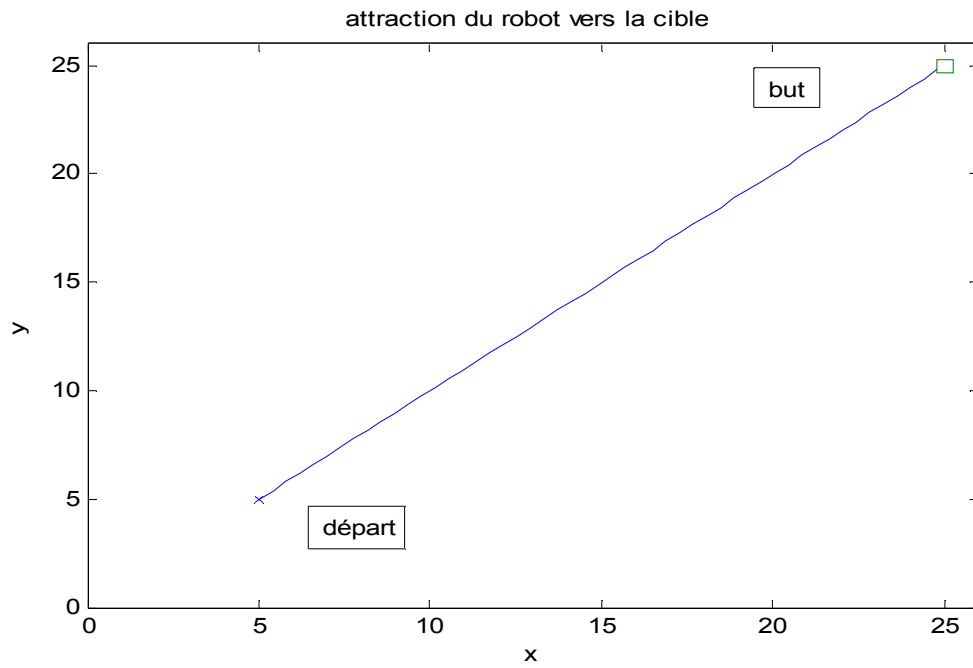


Figure 5.7 : a) Navigation libre avec la cible devant le robot avec $\theta_0 = \pi / 4$

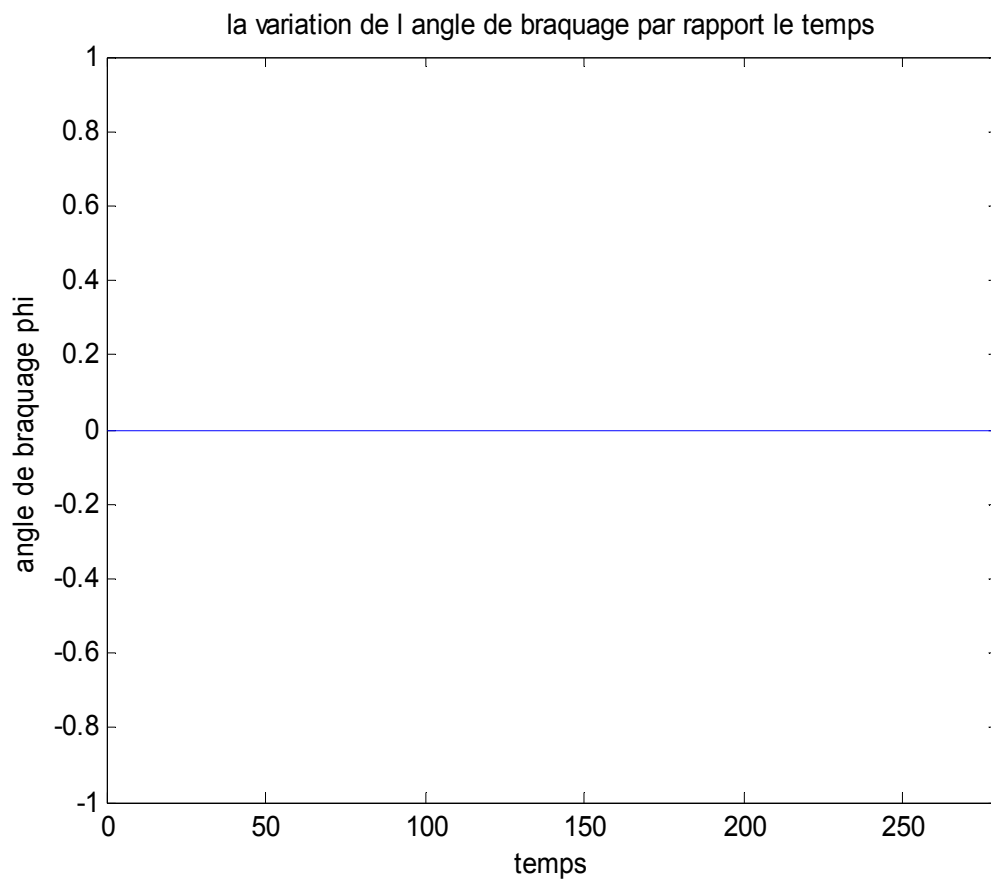


Figure 5.7 : b) Variation de l'angle de braquage du robot par rapport au temps

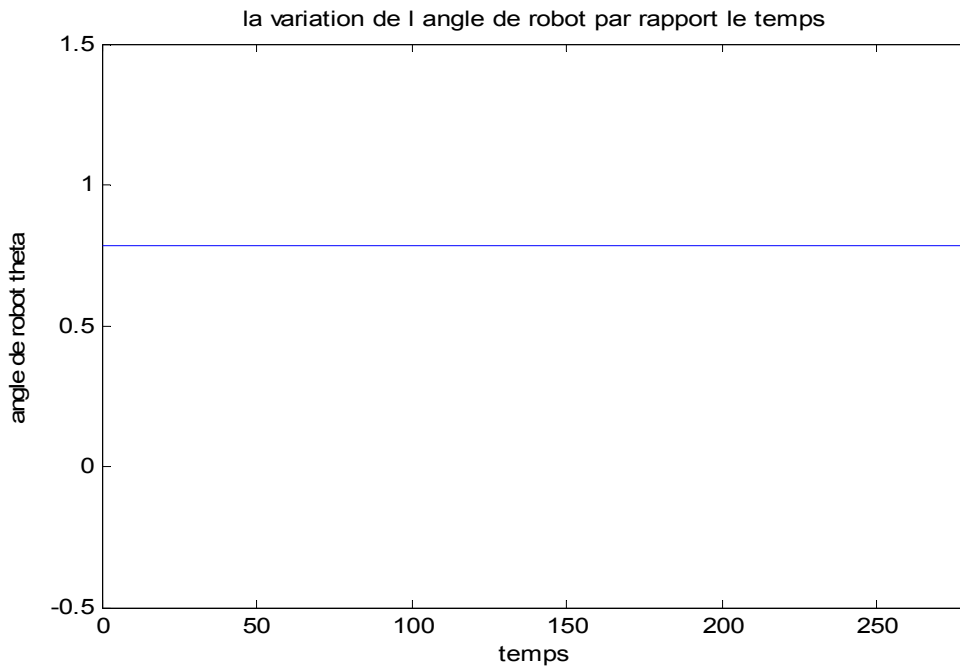


Figure 5.7 : c) Variation de l'angle du robot par rapport au temps

En cas d'absence d'obstacles et si la cible est en face du robot, la trajectoire décrite lors du déplacement de notre robot mobile peut être classée dans la catégorie de trajectoire rectiligne comme le montrent les figures 5.7-a et 5.8-a.

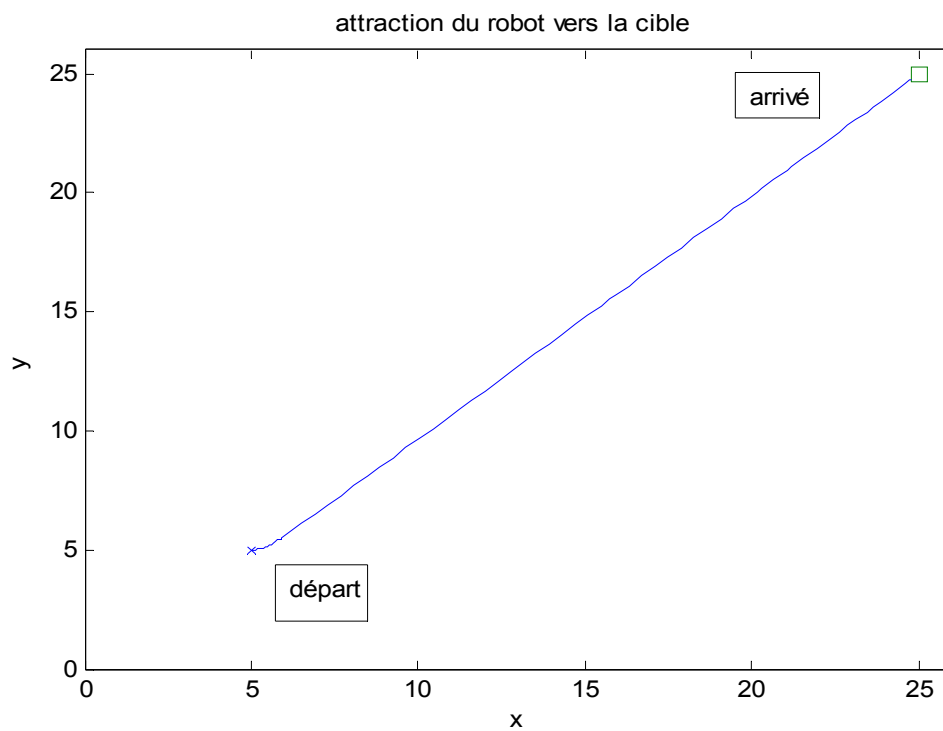


Figure 5.8 : a) Navigation libre avec la cible avant le robot avec $\theta_0 = 0$

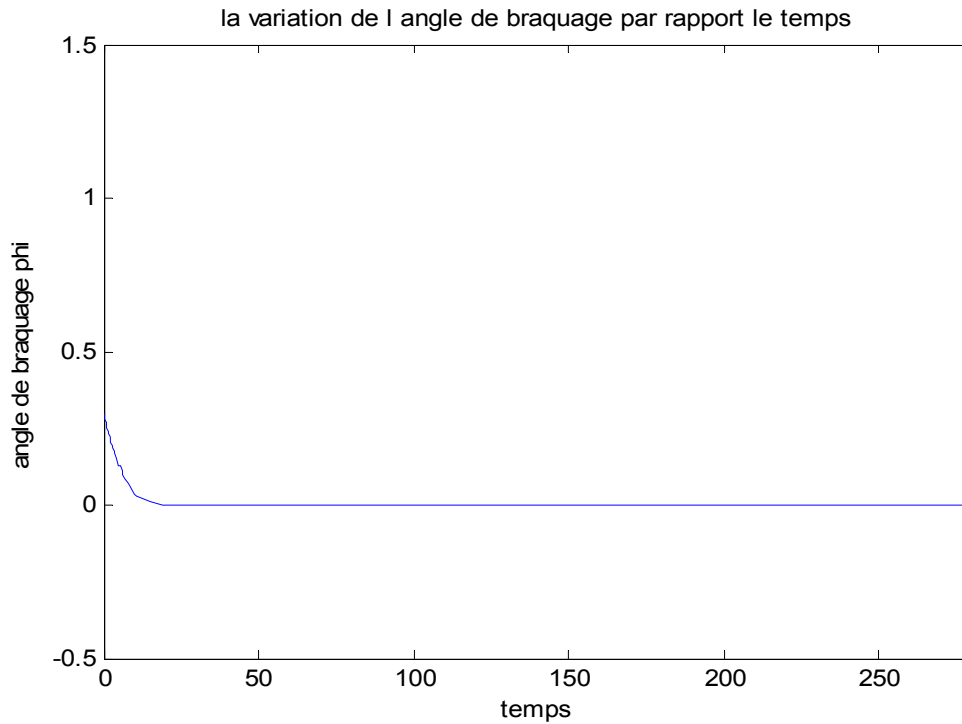


Figure 5.8 : b) Variation de l'angle de braquage du robot par rapport au temps

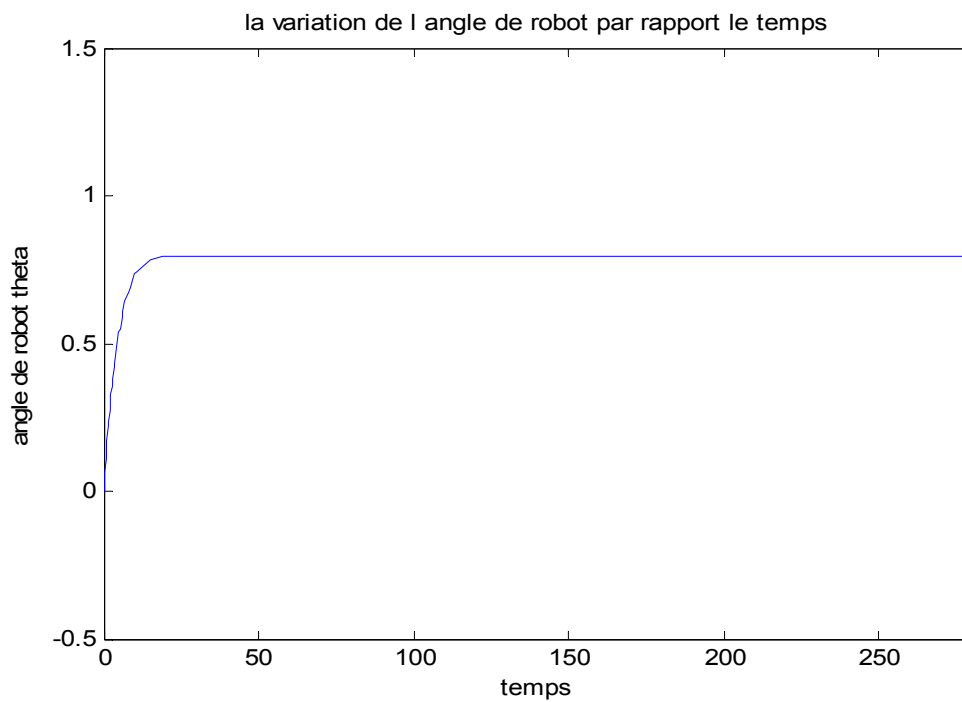


Figure 5.8 : c) Variation de l'angle du robot par rapport au temps

Les figures 5.8-b et 5.8-c présentent respectivement la variation de l'angle du robot par rapport au temps et la variation de l'angle de braquage du robot par rapport au temps.

Dans le cas où le robot est orienté dans une direction autre que celle du point d'arrivée, le robot commence à se déplacer en ligne droite avec un braquage permettant de corriger son orientation. Une fois le robot mobile orienté correctement, il reprendra l'étape précédente. Les figures 5.9 et 5.10 représentent le cas d'un robot mobile dont l'orientation initiale est différente de celle de sa destination.

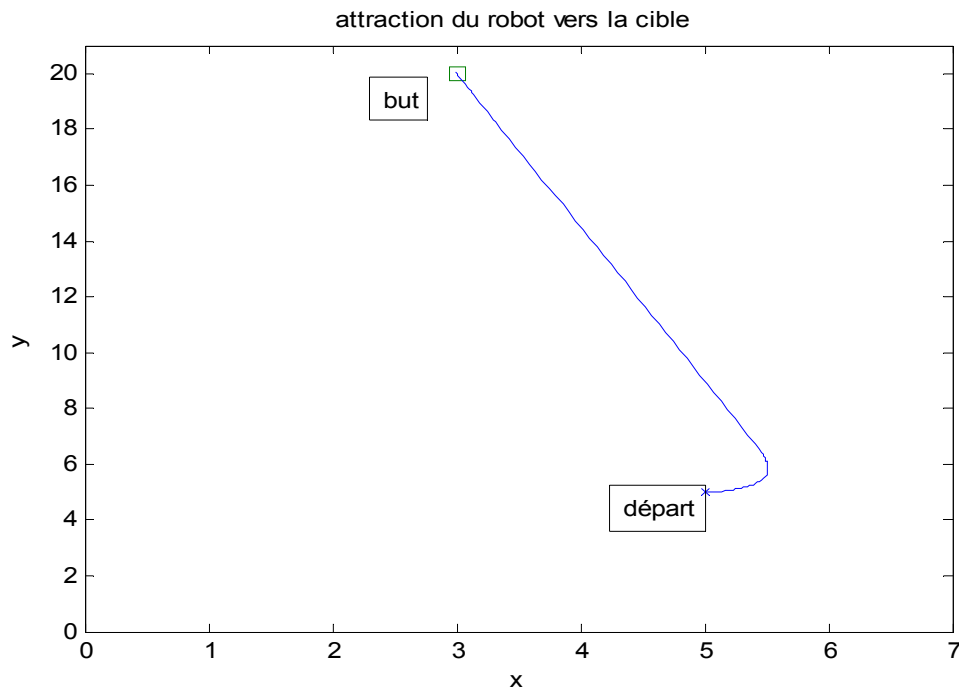


Figure 5.9 : a) Navigation libre avec la cible derrière le robot avec $\theta_0 = 0$

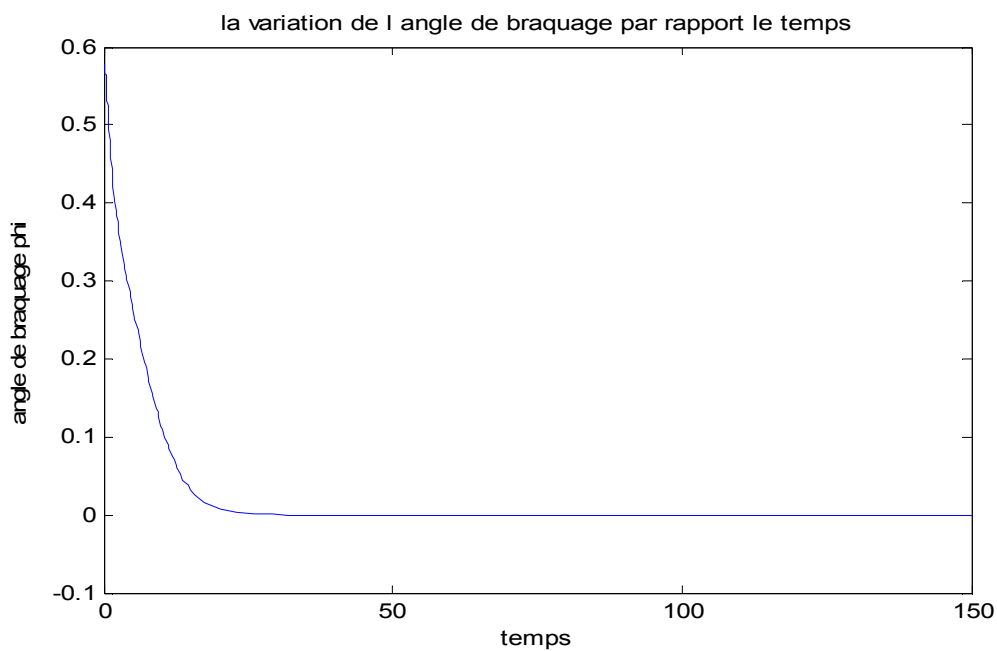


Figure 5.9 : b) Variation de l'angle de braquage du robot par rapport au temps

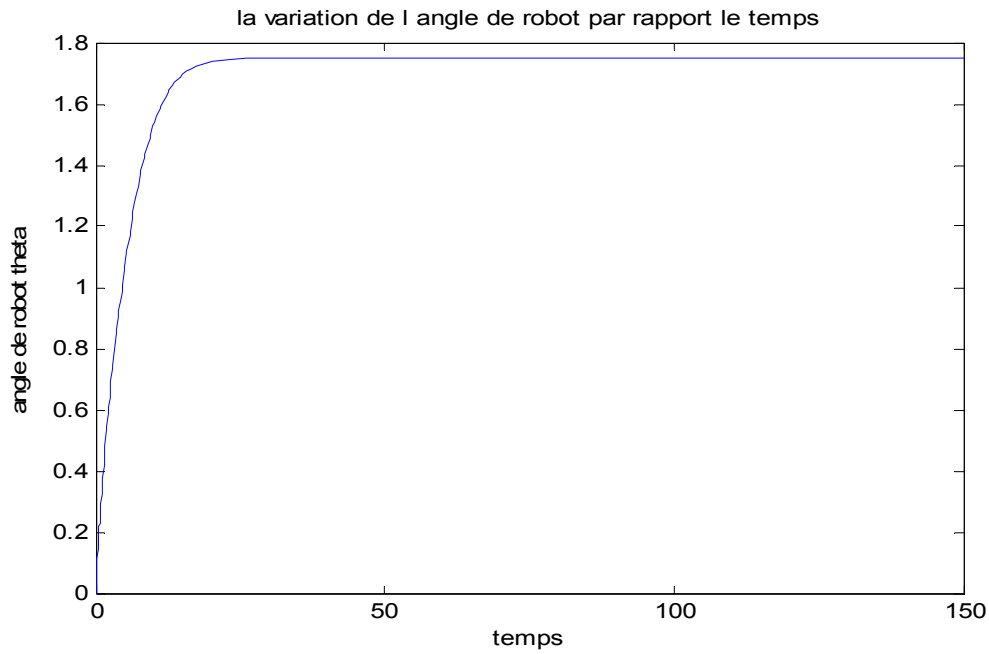


Figure 5.9 : c) Variation de l'angle du robot par rapport au temps

Un cas de navigation libre avec la cible derrière le robot est présenté sur la figure 5.10.

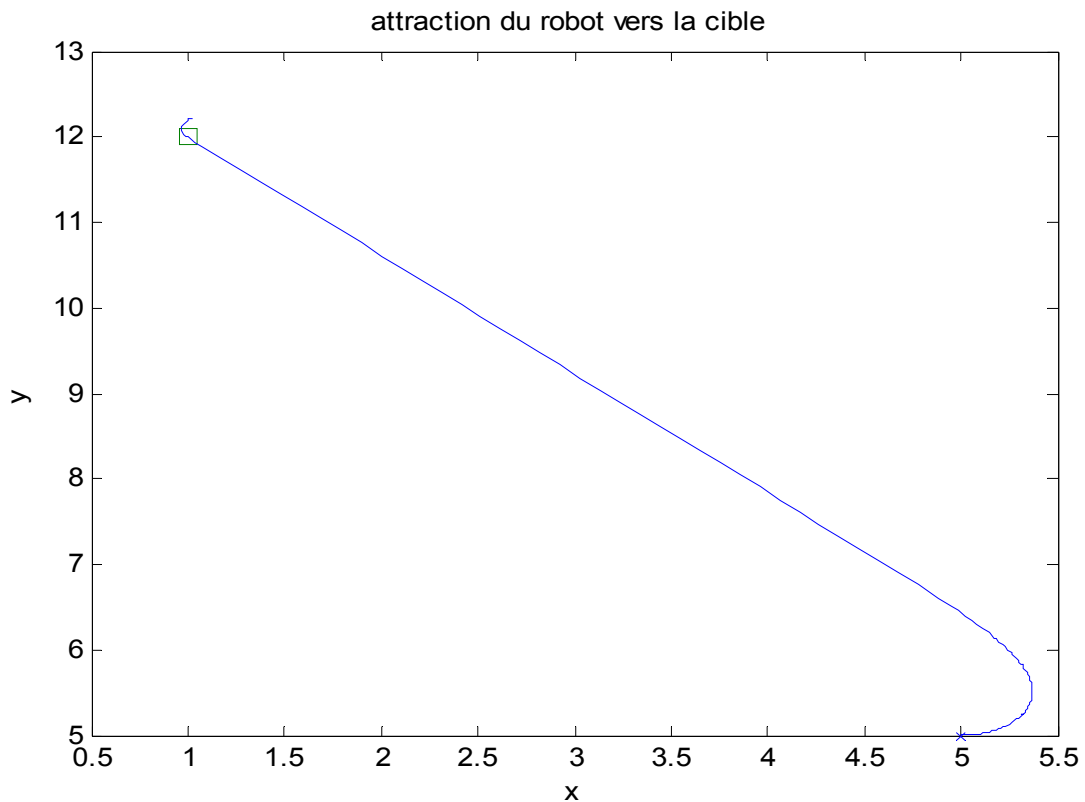


Figure 5.10 : Navigation libre avec la cible derrière le robot

5.4.3 Comportement d'évitement d'obstacles

Pour éviter les obstacles, nous avons développé un système de contrôle réactif en utilisant la logique floue. Notre système d'évitement d'obstacles met en œuvre une stratégie de contrôle intelligente réactive en ce sens que l'environnement immédiat réagit (avec l'aide du raisonnement approximatif) au cours de données sensorielles. Par exemple, si le robot est à proximité immédiate d'un obstacle devant lui, le contrôleur flou permet de déterminer une nouvelle direction à suivre en temps réel afin d'éviter la collision avec l'obstacle. Le robot sera alors immédiatement régi en fonction de la directive émise par le contrôleur flou. Ces comportements réactifs permettent au robot mobile d'éviter les obstacles statiques de l'environnement, sans exiger de modèles prédéfinis. L'existence d'un but nous permet de nous concentrer pleinement sur deux comportements fondamentaux : l'évitement des collisions et le rapprochement du but.

Ce contrôleur travaille en adaptant en étroite collaboration avec les capteurs afin de détecter et d'éviter les obstacles, en adaptant la trajectoire à la situation actuelle. Ce comportement est plus complexe par rapport à la navigation libre (dans la zone critique). Il reçoit comme entrées la distance entre le robot et l'obstacle dans les différentes perceptions et renvoie en sortie la commande φ , qui correspond au changement de direction du robot (l'angle de braquage). La figure 5.11 montre la structure du contrôleur d'évitement d'obstacle.

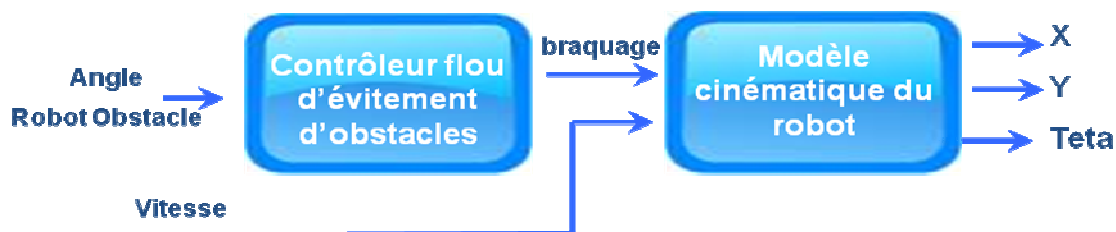


Figure 5.11 : Structure du contrôleur d'évitement d'obstacles

5.4.3.1 Fuzzification

Fuzzification des variables d'entrées

La fonction d'appartenance des variables d'entrées (Latéral _ Négatif, Dans _ Avant et Latéral _ Positif) sont exprimées par trois ensembles flous qui sont : pas d'obstacle, prêt et loin. Ces variables sont représentées sur la figure 5.12.

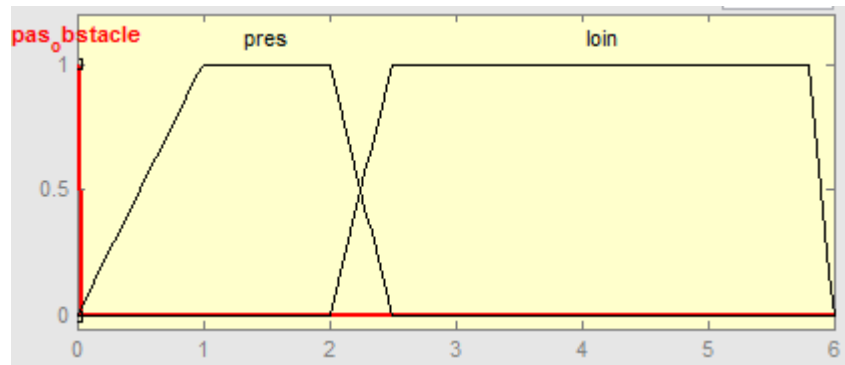


Figure 5.12 : Fonction d'appartenance des variables d'entrées

Fuzzification de la variable de sortie

La variable de sortie qui est l'angle de braquage φ possède cinq variables floues (NG : Négatif Grand, NP : Négatif Petit, Z : Zéro, PP : Positif Petit, PG : Positif Grand) définies comme suit :

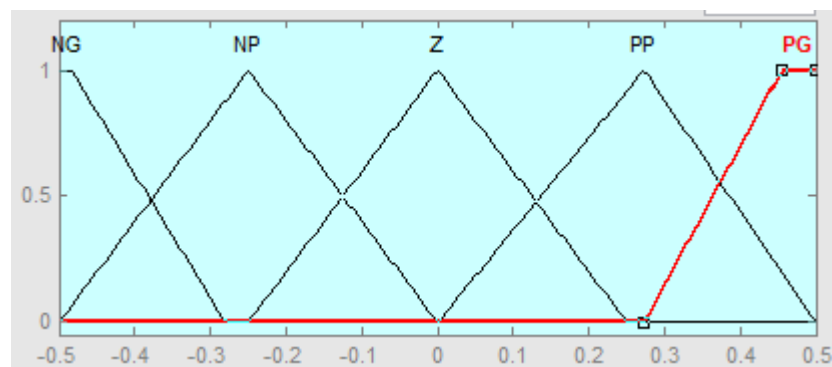


Figure 5.13 : Fonction d'appartenance de l'angle de braquage

5.4.3.2 Inférence

Le tableau 5.2 regroupe la série des règles d'inférence floues.

Latéral _ Négatif	Dans _ Avant	Latéral _ Positif	φ
Pas _ Obstacle	Pas _ Obstacle	Prés	NG
		Loin	Z
	Prés	Pas _ Obstacle	NG/PG
		Prés	NG
		Loin	NG
	Loin	Pas _ Obstacle	NP/PP
		Prés	NP
		Loin	NP
	Prés	Pas _ Obstacle	Pas _ Obstacle
Prés			Z
Loin			Z
Prés		Pas _ Obstacle	PG
Loin		Pas _ Obstacle	PP
Loin	Pas _ Obstacle	Pas _ Obstacle	Z
		Prés	Z
		Loin	Z
	Prés	Pas _ Obstacle	PG
	Loin	Pas _ Obstacle	PP

Tableau 5. 2 : Les règles d'inférence floue d'évitement d'obstacles

5.4.3.3 Defuzzification

La méthode de défuzzification la plus utilisée est celle de la détermination du centre de gravité de la fonction d'appartenance résultante. Elle est caractérisée par un simple calcul et un temps de calcul court donc, elle est la plus utilisée.

5.4.3.4 Résultats expérimentaux

Si le robot rencontre un obstacle imprévu, celui-ci est détecté par l'intermédiaire d'un module de détection d'obstacles ; le robot doit réagir en temps réel c'est-à-dire exécuter une

action de contournement autour de l'obstacle détecté. Les figures ci-dessous montrent quelques cas de notre système de navigation avec évitement d'obstacles.

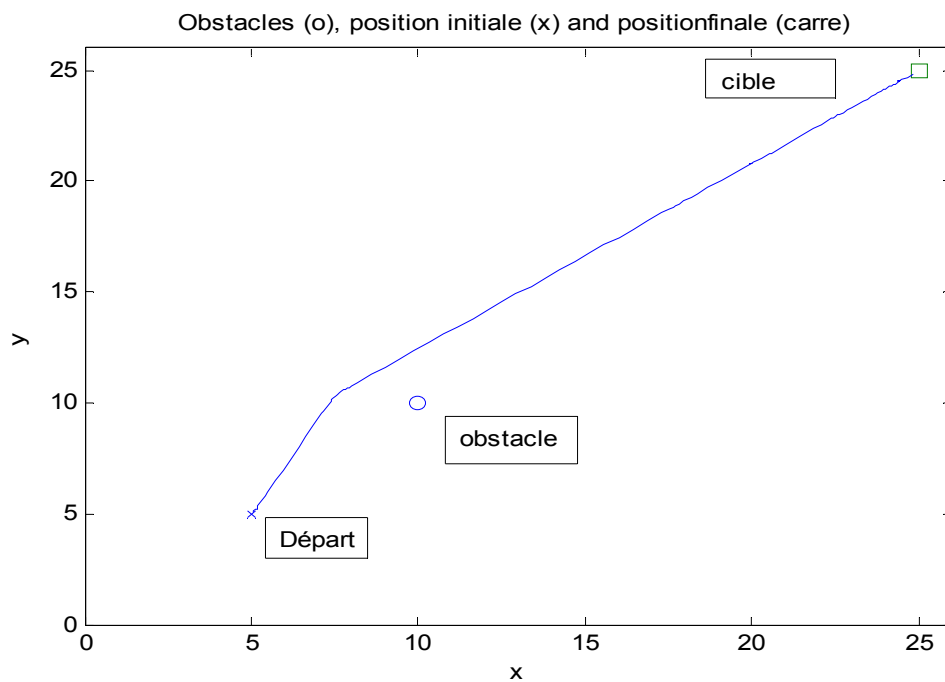


Figure 5.14 : a) Navigation avec évitement d'un seul obstacle

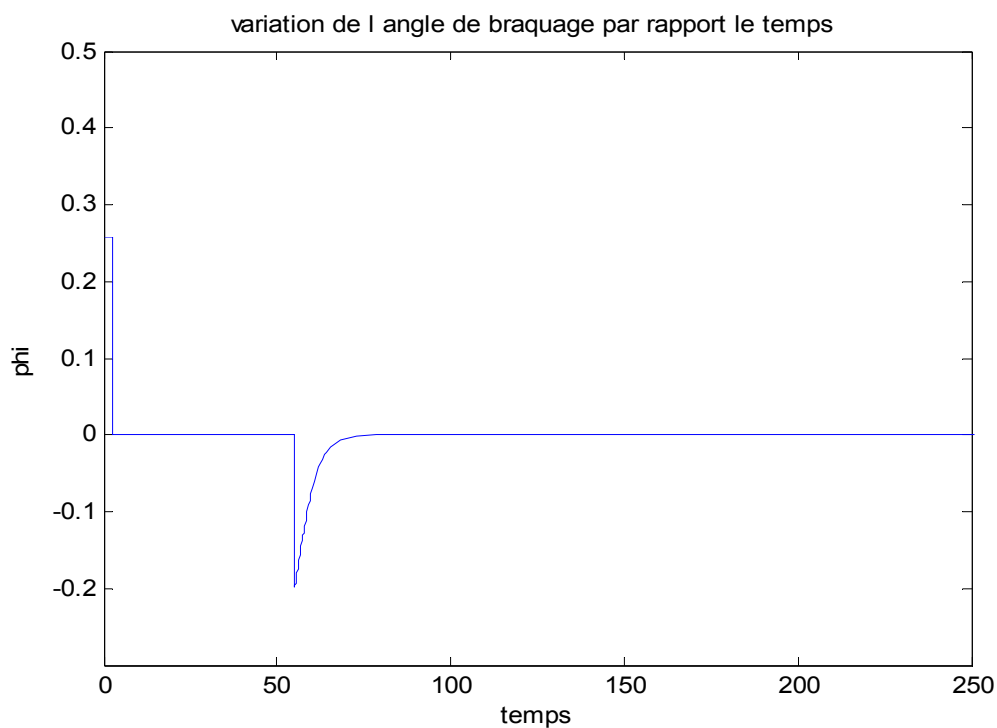


Figure 5.14 : b) Variation de l'angle de braquage du robot

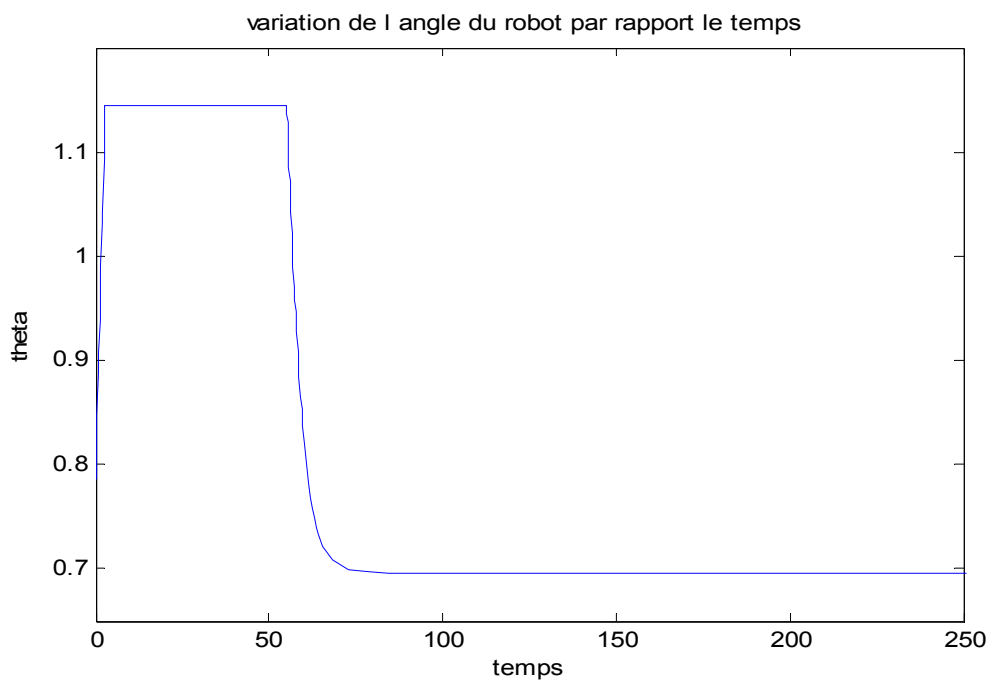


Figure 5.14 : c) Variation de l'angle du robot

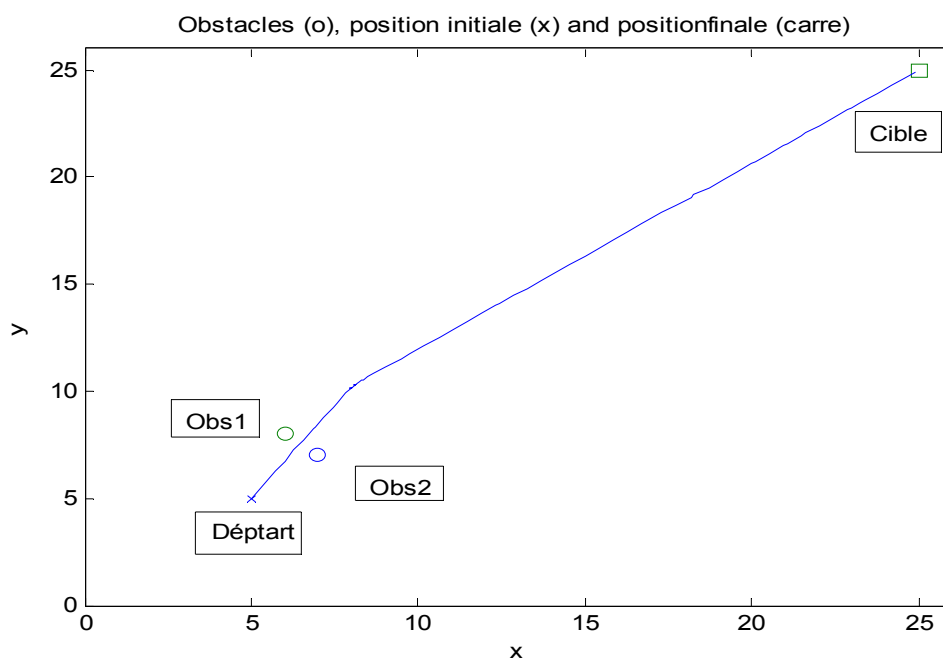


Figure 5.15 : a) Navigation avec évitement de deux obstacles

Dans le premier cas de test présenté à la figure 5.14-a et 5.15-a, un simple environnement avec un nombre réduit de petits obstacles a été testé. Le robot utilise l'angle de braquage afin d'éviter l'obstacle ou les deux obstacles, ensuite il se déplace selon des trajectoires plus ou moins rectilignes vers le but. A l'approche de la cible, le robot ralentit et s'immobilise.

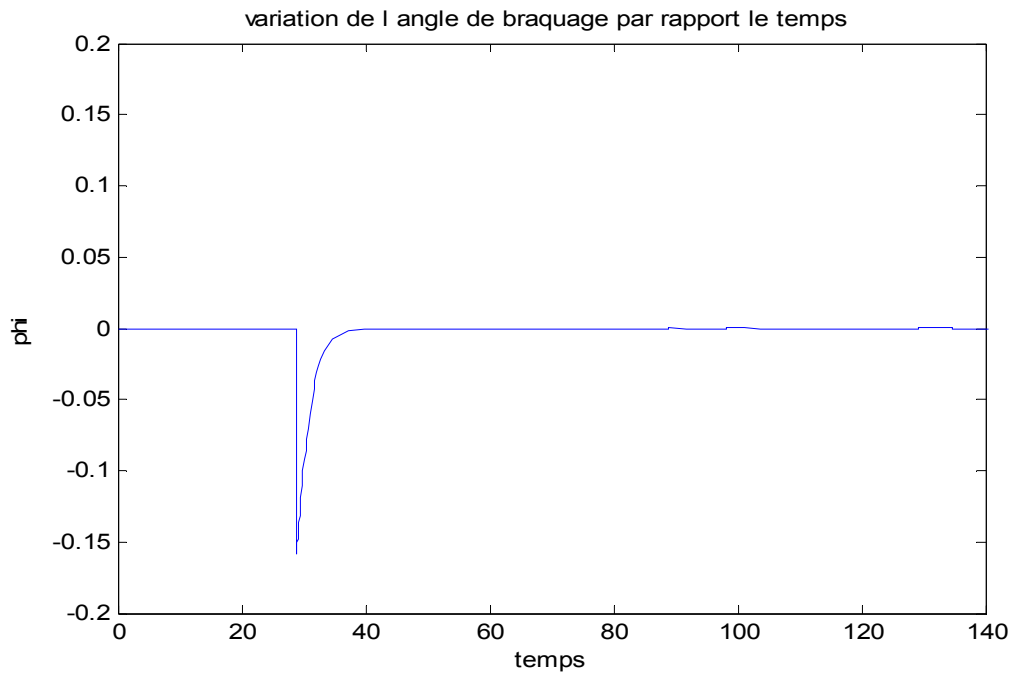


Figure 5.15 : b) La variation de l'angle de braquage du robot

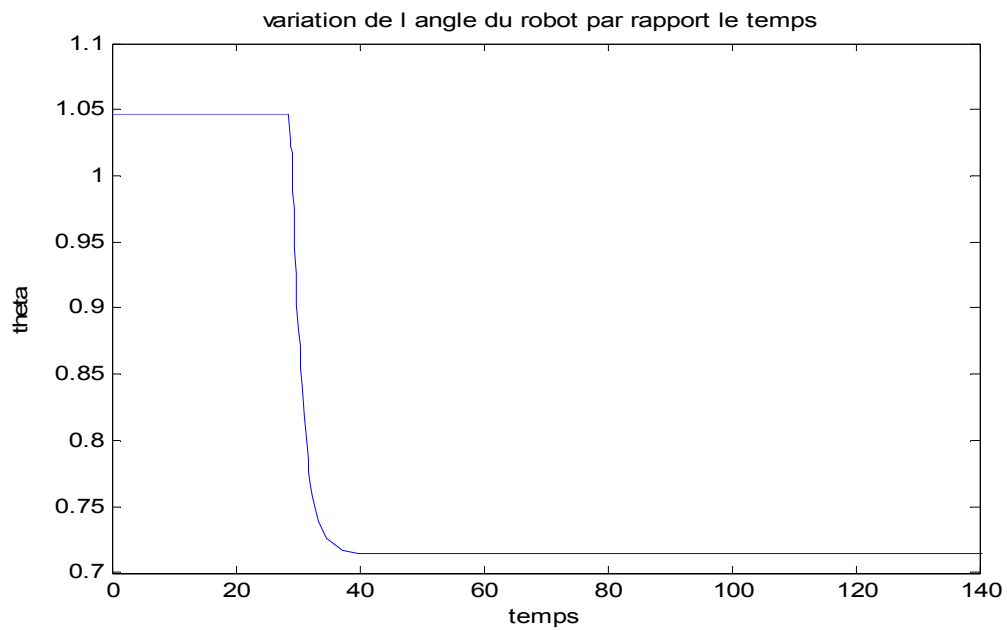


Figure 5.15 : c) La variation de l'angle du robot

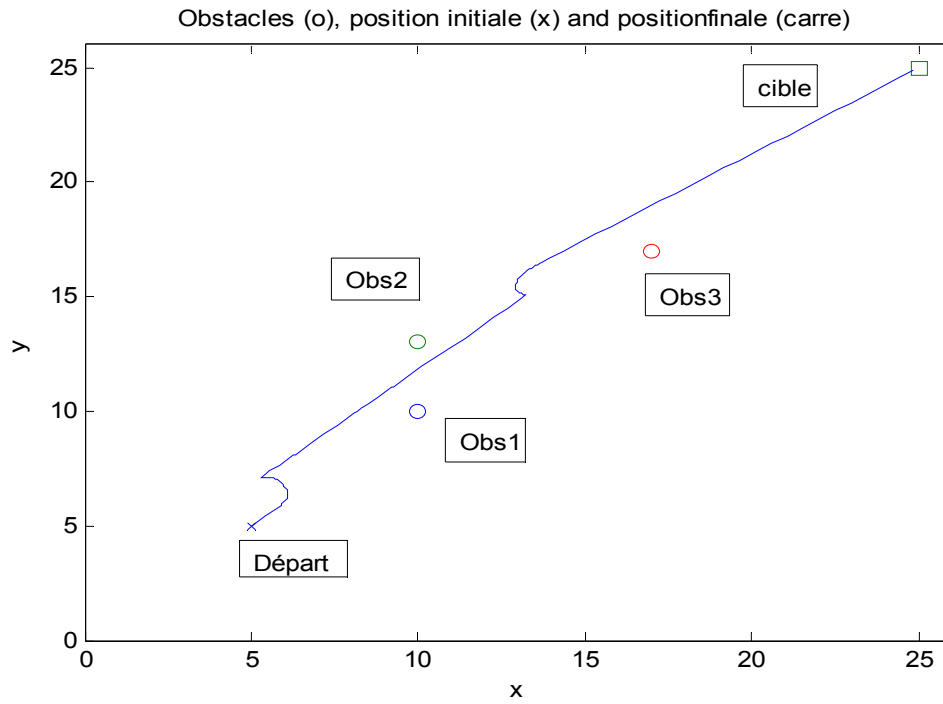


Figure 5.16 : a) Navigation avec évitement de trois obstacles

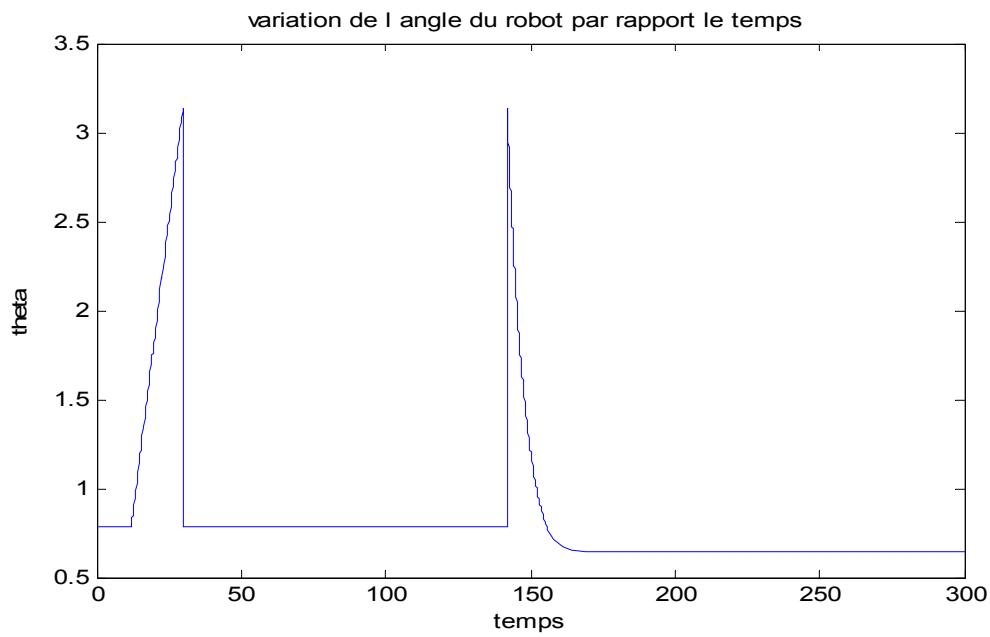


Figure 5.16 : b) Variation de l'angle du robot

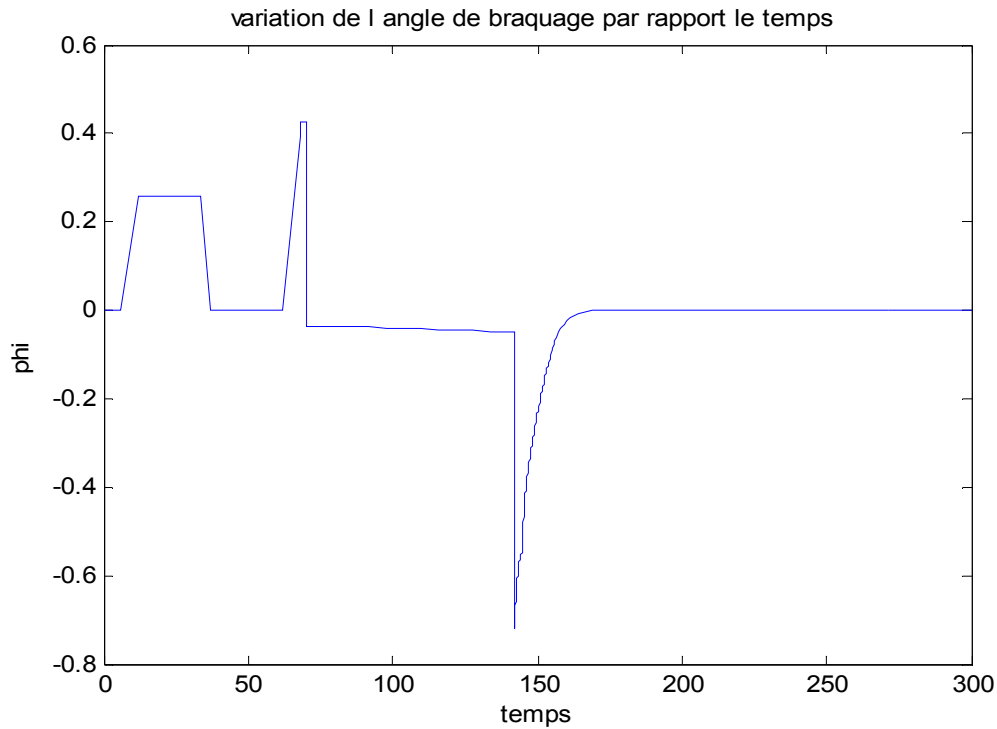


Figure 5.16 : c) Variation de l'angle de braquage du robot

Dans le deuxième cas de test présenté sur les figures 5.16-a, 5.17 et 5.18-a, de simples environnements avec présence de trois obstacles ont été testés. Le robot utilise toujours l'angle de braquage afin d'éviter les obstacles, ensuite il se déplace selon des courbes vers le but assigné. Il ralentit et s'immobilise à l'approche de la cible.

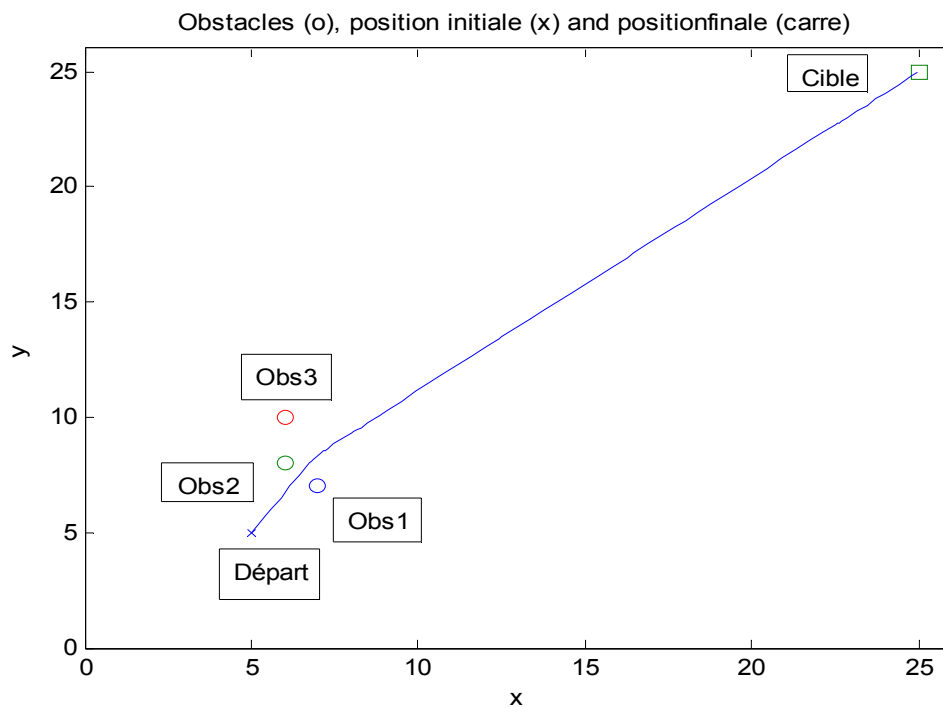


Figure 5.17 : Navigation avec évitement de trois obstacles

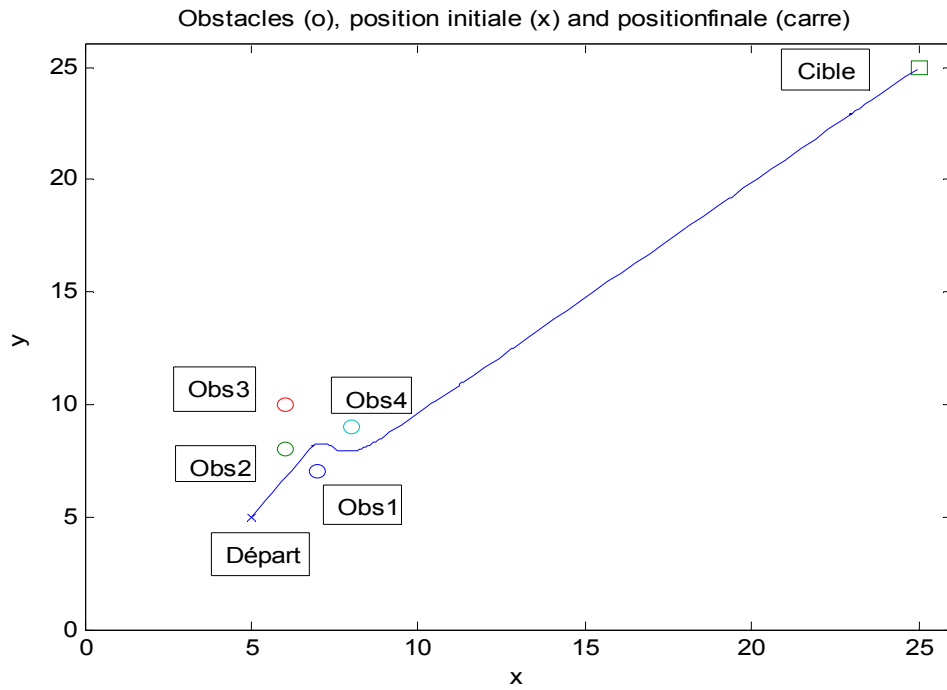


Figure 5.18 : a) Navigation avec évitement de quatre obstacles

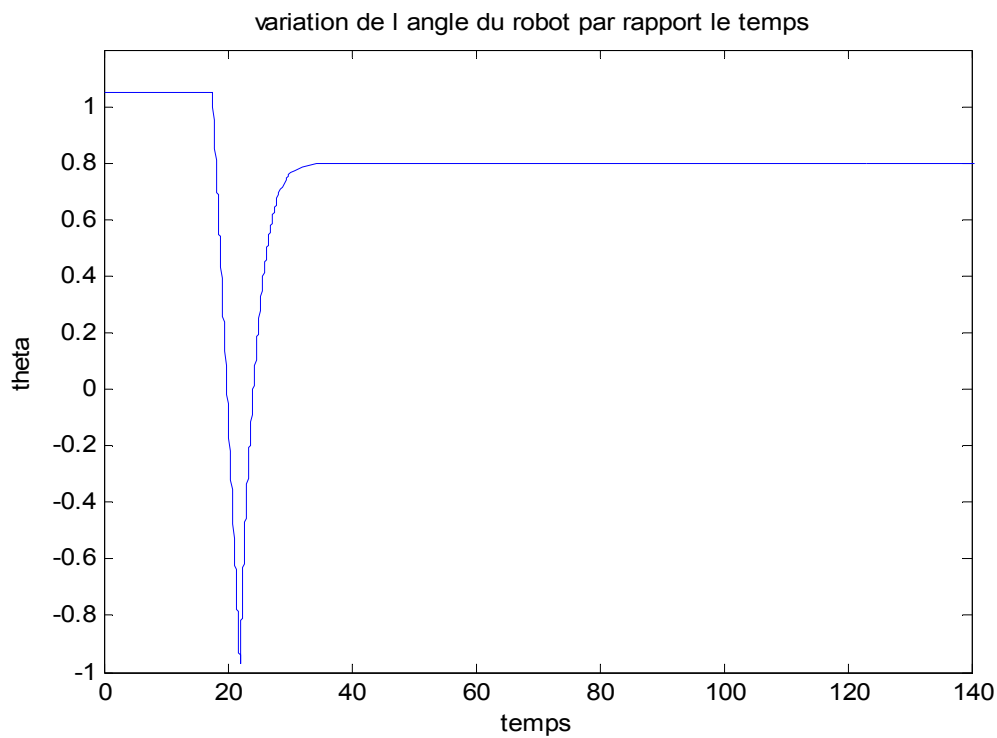


Figure 5.18 : b) Variation de l'angle du robot

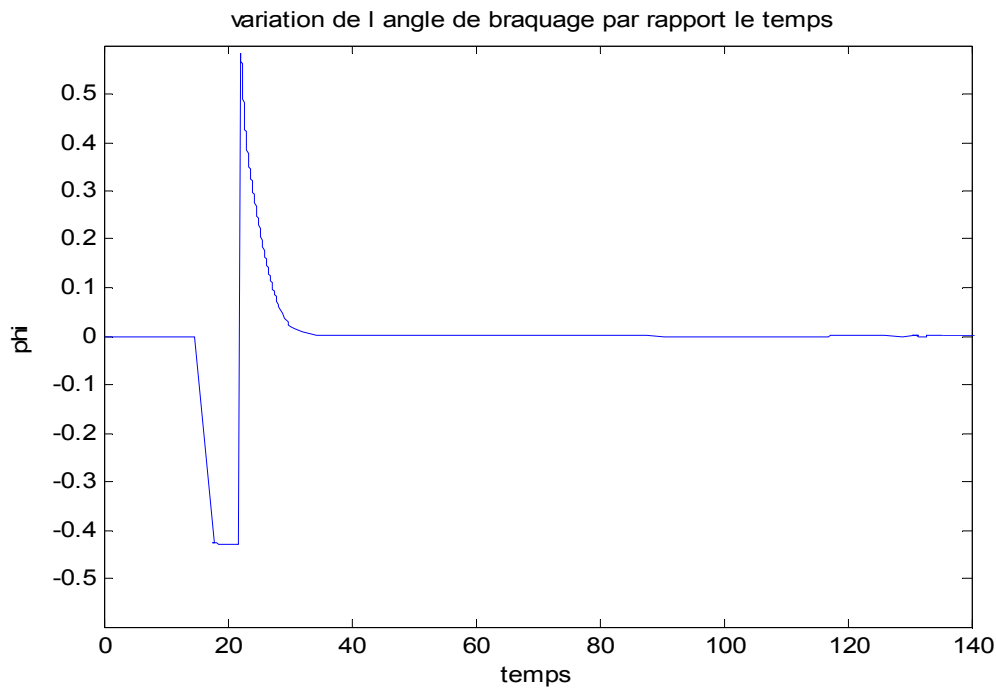


Figure 5.18 : c) Variation de l'angle de braquage du robot

5.5 Conclusion

La faisabilité du contrôleur flou intelligent pour mettre en œuvre le calcul approximatif de raisonnement nécessaire pour la manipulation de l'incertitude, inhérente au problème d'évitement d'obstacles, avec le minimum de capteurs (trois capteurs semblent largement suffisants pour permettre une navigation du robot), a été constatée ici. Ce chapitre se focalise sur la conception détaillée de notre système de navigation floue.

Tout d'abord, nous avons décrit la structure globale et l'architecture détaillée de notre système ainsi que les contrôleurs d'inférences floues pratiquement utilisés. Ensuite nous avons présenté les résultats de simulation de notre système en utilisant l'environnement Matlab-Simulink. Puis nous avons présenté le fonctionnement du système et les contextes applicatifs de la navigation. Enfin nous avons décrit la manière dont nous avons implémenté chaque module de notre système, ainsi que les structures de données utilisées et les algorithmes de fonctionnement.

La logique floue est un outil puissant, simple à implémenter. Ces contrôleurs présentent néanmoins quelques inconvénients, entre autre la nécessité d'évaluer et d'affiner les contrôleurs (les fonctions d'appartenance d'entrée /sortie et les tables d'inférences). Cette évaluation est difficile à cause des règles d'inférence et nécessite plus de temps, c'est-à-dire les règles doivent être construites par un expert, qui peut définir pour chaque règle la conclusion correspondante.

CONCLUSION GENERALE

Dans le domaine de la robotique, le contrôleur flou peut concurrencer les contrôleurs classiques. Il offre de bonnes caractéristiques en termes de temps de réponse, de dépassement et surtout de robustesse dans certaines applications. En plus, il n'a besoin que d'une connaissance sommaire sur le système à contrôler exprimée sous forme de relations approximatives où l'on enregistre le comportement du système.

Un système de contrôle intelligent a été développé ici pour mettre en œuvre le calcul approximatif de raisonnements nécessaires pour le problème de la navigation autonome d'un robot mobile avec évitement d'obstacles en temps réel dans un environnement structuré.

Nous avons développé un ensemble de contrôleurs pour aider le robot à générer des chemins qui permettent de rejoindre un objectif avec la possibilité de contourner les obstacles encombrant son environnement.

L'objectif de ce mémoire est l'élaboration d'algorithmes d'aide à la navigation pour un robot mobile d'intérieur. Notre travail s'inscrit dans le contexte de la robotique et nous nous sommes focalisés sur le principe de la démarche humaine dans le sens de capturer la vision approximative de la pensée humaine et de l'exprimer avec des outils appropriés.

Pour l'optimisation de la navigation dans notre cas, une architecture coopérative à plusieurs niveaux a été nécessaire. Une solution satisfaisante pour la navigation réactive présentant de grande capacité de généralisation a été obtenue à l'aide de la méthode des champs de potentiels artificiels et d'un système expert d'inférence floue.

La complexité des méthodes de conception est dans l'interaction robot-environnement, un moyen efficace de structurer la tâche de navigation en vue de régler le problème de la navigation du robot mobile est démontré. Les questions de conception des comportements spécifiques et l'action de conception et de coordination des comportements ont été traitées.

L'approche présentée dans ce mémoire porte essentiellement sur la résolution du problème d'évitement d'obstacles lié à la navigation d'un robot mobile. Notre robot est testé dans un environnement de simulation. De ce fait, on peut agir sur le temps de calcul, on peut introduire beaucoup de données et on peut faire varier les paramètres de l'environnement selon plusieurs configurations. On peut faire exécuter au robot plusieurs types de trajectoires et on peut aussi modifier les données de la position initiale. Nous avons mis en œuvre une

stratégie de contrôle intelligente réactive, qui est une technique d'inférence simple et rapide permettant de prendre des décisions en temps réel. La méthode des champs de potentiels artificiels est présentée comme une méthode classique de navigation d'un robot mobile cependant un problème de minimas locaux est parfois constaté.

La logique floue comme outil adéquat est utilisée pour traiter le problème de la navigation autonome d'un robot mobile. L'emploi de cette technique, en plus de sa simplicité, a un intérêt supplémentaire qui réside dans sa capacité d'impliquer l'intelligence humaine dans sa stratégie de contrôle. De plus, en raison de son temps de réponse court, le navigateur flou est particulièrement adapté aux applications en ligne avec de fortes exigences par rapport au temps réel. De plus le problème des minimas locaux est complètement résolu par cette méthode.

Nous remarquons que les résultats de la simulation démontrent bien l'efficacité des approches utilisées pour la conception des contrôleurs accomplissant une tâche de navigation rapide, précise et robuste. Le système élabore en permanence le bon chemin autour des obstacles avec des braquages lisses. En plus le chemin de déplacement obtenu est proche de l'optimal et plus réaliste. Le robot a la capacité de se déplacer dans son environnement sans aucune collision avec les obstacles et avec l'utilisation des méthodes d'inférence rapides le temps de parcours est minimisé. Un avantage supplémentaire de la stratégie utilisée est la modélisation et la conception du système en temps réduit.

Perspectives

Des perspectives ont émergé pendant la réalisation de ce travail, on citera entre autres :

- Extension de notre approche à des environnements dynamiques.
- Utilisation d'autres méthodes d'optimisation tel que la commande robuste.
- Amélioration du système de perception et test des différents paramètres du procédé pour optimiser la trajectoire globale.

Nous espérons que ce travail motivera le lecteur à créer plusieurs applications avec l'exploitation des propriétés intrinsèques de la méthode des champs de potentiels et la logique floue. La suite de ce travail peut prendre de nombreuses directions et rejoindre d'autres thèmes scientifiques autour de la robotique mobile.

BIBLIOGRAPHIE

- [ADA03] Adachi, Y.; Saito, H.; Matsumoto, Y.; Ogasawara, T., "*Memory-based navigation using data sequence of laser range finder*", Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International symposium on Volume 1, Issue , 16-20 July 2003 Page(s): 479 - 484 vol.1
- [AND83] Andrews, J. R. and Hogan, N., "*Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator.*" Control of Manufacturing Processes and Robotic Systems, Eds. Hardt, D. E. and Book, W., ASME, Boston, 1983, pp. 243-251.
- [ARK89] Arkin R. C., « *Motor Schema-Based Mobile Robot Navigation* », The International Journal of Robotics Research, vol. 8, n° 4, p. 92-112, 1989.
- [ARM98] J.M. Armingol, L. Moreno, A. de la Escalera et M.A. Salichs – "*Landmark perception planning for mobile robot localization*" – IEEE International Conference on Robotics and Automation ICRA'98, Leuven, Belgique, mai 1998.
- [BEL02] Belker, T., Schulz, D. « *Local Action Planning for Mobile Robot Collision Avoidance*», Intelligent Robots and System, 2002. IEEE/RSJ International Conference on Volume 1, 30 Sept.-5 Oct. 2002 Page(s):601 - 606 vol.1
- [BIC20] E. Bicho, *Dynamic Approach to Behavior-Based Robotics: design, specification, analysis, simulation and implementation*. Shaker Verlag, 2000.
- [BIC97] E. Bicho and G. Schoner, "*The dynamic approach to autonomous robotics demonstrated on a low-level vehicle platform,*" Robotics and Autonomous Systems, vol. 21, pp. 23 – 35, 1997.
- [BOR91] J. Borenstein and Y. Koren, "*The vector field histogram - fast obstacle avoidance for mobile robots,*" IEEE Transaction on Robotics and Automation, vol. 7, no. 3, pp. 278 – 288, 1991.
- [BOR89] J. Borenstein and Y. Koren, « *Real-time Obstacle Avoidance for Fast Mobile Robots*» IEEE Transactions on Systems, Man, and Cybernetics,

Vol. 19, No. 5, pp. 1179-1187 Sept. /Oct. 1989

- [BOR95] J. Borenstein – "*Internal correction of dead-reckoning errors with a dual-drive compliant linkage mobile robot*" – Journal of Robotic systems , Vol. 12, n°4, 1995, p. 257-273.
- [BOR96] J. Boreinstein, L. Feng – "*Gyrodometry : a new method for combining data from gyros and odometry in mobile robots*" – Proc. of the IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota - April 1996.
- [BRO86] Brooks, R. A., "*A Robust Layered Control System for a Mobile Robot.*" IEEE Journal of Robotics and Automation, Vol. RA-2, No. 1, 1986, pp. 14-23
- [CAM96] 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.
- [CAO86] Z.L. Cao, S.J. Oh and E.L. Hall – "*Dynamic Omnidirectionnal Vision for Mobile Robots*" – Journal of Robotic System, 3(1), pages 5-17, 1986.
- [CHA96] Chatterjee, R., Matsuno, F., « *Use of single side reflex for autonomous navigation of mobile robots in unknown environments* » ; Robotics and Autonomous Systems, Volume 35, Issue 2, 31 May 2001, Pages 77-96
- [COU08] Courbon, J., Mezouar, Y., Lequievre, L., Eck, L., "*Navigation of urban vehicle: An efficient visual memory management for large scale environments*", IROS 2008: 1817-1822
- [COU09] J. Courbon, Y. Mezouar, L. Eck, P. Martinet, "*A generic framework for topological navigation of urban vehicle*", ICRA09 Workshop on Safe navigation in open and dynamic environments Application to autonomous vehicles, ICRA09, Kobe, Japan, May 12th, 2009
- [DUD00] G. Dudek et M. Jenkin. *Computational principles of mobile robotics.* Cambridge University Press, 2000.
- [ELK94] R.T. Elkins and E.L. Hall – "*Three dimensional line following using omnidirectional vision*" – Proc. SPIE Intelligent Robots and Computer Vision XIII, vol. 2354, pp 130-144, 1994
- [ERM04] Er, M.J., Tien Peng Tan, Sin Yee Loh, « *Control of a mobile robot using generalized dynamic fuzzy neural networks* »; Microprocessors and Microsystems, Volume 28, Issue 9, 2 November 2004, Pages 491-498

- [FAT05] A. Fatmi, A. Al Yahmadi, L. Khriji, and N. Masmoudi, «*A Fuzzy Logic Based Navigation of a Mobile Robot*» International Journal of Applied Mathematics and Computer Sciences 1;2 , Spring 2005
- [FOF97] D. Fofi – "*Détection d'obstacles par perception visuelle pour un robot mobile*" –Rapport de DEA, Ecole Nationale Supérieure de l'Electronique et de ses Applications, Université de Cergy- Pontoise, Août 1997.
- [FOU99] J.-Y. Fourquet et M. Renaud. Coordinated Control of a Non-Holonomic Mobile Manipulator. In ISER'1999, pages 115–125, Sydney, Australie, mars 1999.
- [FRA90] G. Frappier – "*Système inertiels de navigation pour robots mobiles*" – Séminaire "Les robots mobiles", EC2, Paris, 1990
- [GIR84] G. Giralt, R. Chatila et M. Vaisset. *An Integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots*. In First International Symposium on Robotics Research, pages 191–214, Cambridge, 1984.
- [HOL91] J. Hollingum – "*Caterpillar make the earth move : automatically*" – Industrial Robot, Vol. 18, N° 2, pp 15-18, 1991.
- [HUH02] D. Huh, J.Park , U. Huh and H. Kim, «*Path planning and navigation for autonomous mobile robot*» Departement of Electronic., Inha University., Inchon, South Korea, 2002
- [ISH92] H. Ishiguro, M. Yamamoto and S. Tsuji – "*Analysis of omnidirectional views at different location*" – In Proceedings of IEEE International Conference on Intelligent Robotics and Systems, pp 659- 664, 1990.
- [KAN86] T. Kanade, C. Thorpe et W. Whittaker. *Autonomous Land Vehicle Project at CMU*. In ACM Annual Computer Science Conference, pages 71–80. ACM Press, 1986.
- [KHA95] O. Khatib, "*Real-time obstacle avoidance for manipulators and mobile robots,*" International Journal of Robotics Research, vol. 5, no. 1, pp. 90–98, 1995.
- [KHA86] O. Khatib, '*Real-time Obstacle Avoidance for Manipulators and Mobile Robots,*' International Journal of Robotics Research, 1986, 5(1), pp 90-98.
- [KOR91] Koren, Y.; Borenstein, J. , "*Potential field methods and their inherent limitations for mobile robot navigation*"; *Robotics and Automation*, Sacramento, California, April 7-12, 1991, pp. 1398-1404 vol.2

- [KTE02] K-team. Khepera II user manual, 2002.
- [LEB05] Lebedev, D.V., Steil, J.J. and Ritter, H.J. « *The dynamic wave expansion neural network model for robot motion planning in time-varying environments* » ; Neural Networks, Volume 18, Issue 3, April 2005, Pages 267-285
- [LET59] Lettvin, J.Y., Maturana, H.R., McCulloch, W.S., & Pitts, W.H., *What the Frog's Eye Tells the Frog's Brain*”, Proceedings of the IRE, Vol. 47, No. 11, pp. 1940-51, 1959
- [MAK91] H. Makela, K. Koskinen – "*Navigation of outdoor mobile robots using dead reckoning and visually detected landmarks*" – 5th Int. Conf. on Advanced Robotics ICAR'91, Pisa, Italy, June 1991, pp 1051-1056.
- [MAT95a] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe et T. Litwin – "*Mars microrover navigation : performance evaluation and enhancement*" – Autonomous Robot Journal, special issue on autonomous vehicle for planetary exploration, vol. 2, n. 4, 1995.
- [MAT95b] B.O. Matthews, D. Perdue and E. L. Hall – "*Omnidirectional vision applications for line following*" – In Proc. SPIE Intelligent Robots and Computer Vision XIV, vol. 2588, pp 438-449, 1995.
- [MOR03] P. Morin et C. Samson. *Introduction à la commande par retour d'état des systèmes non-holonomes*. Ecole des Mines de Paris, 2003.
- [MOR89] T. Morita and Y. Inamoto and T. Uchiyama and Y. Yasukawa and S. Kawakami – "*Measurement in three dimensions by motion stereo and spherical mapping*" – In Proc. of Computer Vision and Pattern Recognition, pp 422-428, 1989.
- [NAL96] V. Nalwa – "*Bell labs solid-state video invention, CCD, comes full circle*"
- [OHY94] A. Ohya, E. Shoji et S. Yuta – "*3-D range sensor using fiber grating for recognition of autonomous mobile robots passage space*" – International Conference on Intelligent Robots and Systems, 1994.
- [OLS98] C.F. Olson et L.H. Matthies – "*Maximum likelihood rover localization by matching range maps*" – Proc. of the IEEE Int. Conf. On Robotics and Automation ICRA'98, p. 272-277, May 1998.

- [OUA05] Ouadah, N., Azouaoui, O., Hamerlain, M., "Implémentation d'un contrôleur flou pour la navigation d'un robot mobile de type voiture », Troisième Congrès francophone, Majecstic 2005, 16-18 Novembre 2005, Rennes (France).
- [PRA05] Pradalier, C., Hermosillo, J., Koike, C., Braillon, D., Bessière, P., Laugier, C., "The CyCab: a car-like robot navigating autonomously and safely among ". Robotics and Autonomous Systems (RAS) 50(1):51-67 (2005)
- [RON86] J.J. Roning, Z.L. Cao, and E.L. Hall – "Color target recognition using omnidirectional vision" – SPIE Vol 728 Optics, Illumination, and Image Sensing for Machine Vision, 1986.
- [SAR89] K. Sarachik – "Characterising an indoor environment with a mobile robot and uncalibrated stereo" – In Proc. IEEE Conference on Robotics and Automation, pages 984--989, 1989.
- [URM04] C. Urmson, J. Anhalt, M. Clark, T. Galatali, J.P. Gonzalez, J. Gowdy, A. Gutierrez, S. Harbaugh, M. Johnson-Roberson, H. Kato, P.L. Koon, K. Peterson, B.K. Smith, S. Spiker, E. Tryzelaar et W. L. Whittaker. *High Speed Navigation of Unrehearsed Terrain : Red Team Technology for Grand Challenge 2004*.
- [VAG93] J. Vaganay – "Conception d'un système multisensoriel de localisation dynamique 3D pour robot mobile" – Thèse de doctorat, LIRMM, Montpellier, juillet 1993.
- [WAN08] Wang, X.; Hou, Z.; Zou, A.; Tan, M.; Cheng, L. , "A behavior controller based on spiking neural networks for mobile robots" Neurocomputing Volume 71 , Issue 4-6 (January 2008) Pages 655-666
- [WAR83] F. W. Warner. Foundations of differentiable manifolds and lie groups. Graduate Texts in Mathematics. Springer-Verlag, 1983.
- [XUW99] Xu, W.L., « A virtual target approach for resolving the limit cycle problem in navigation of a fuzzy behaviour-based mobile robot » ; *Institute of Technology and Engineering, College of Sciences, Massey University, Palmerston North, New Zealand* ; Received 22 June 1998 ; accepted 20 August 1999
- [YAN05] X. Yang, M. Moallem, and Rajni V. Patel, «A Layered Goal-Oriented Fuzzy Motion Planning Strategy for Mobile Robot Navigation », IEEE Transactions on Systems, Man, and Cybernetics, Part B, Volume 35. Volume 35, NO,2005

