

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



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

*Ecole National Polytechnique*  
*Département de Génie Mécanique*

*Projet de fin d'études*  
*Pour l'obtention du diplôme*  
*D'ingénieur d'état en Génie Mécanique*

*Thème*

*Etude et simulation des trajectoires d'un robot  
mobile*

Proposé et dirigé par :

Prof. A.ZAATRI

Prof. M.BOUAZIZ

Présenté par :

MEZAACHE Ibrahim

Promotion: Juin 2012

## ملخص:

هذا العمل هو عبارة عن دراسة لتحقيق الروبوت المتحرك الاحادي مع ملاحه ذاتية التحكم أو تتبع مسار معين.

قمنا بدراسة الحركية لإيجاد العلاقة بين الموقع وسرعة كل من العجلات. قمنا باستعمال برنامج ماتلاب من اجل تخطيط المسارات و اخترنا نظام ملاحه جديد يعتمد على حقل قوى افتراضي لكي يتجنب الروبوت العقبات في مجال حركته و اخيرا قمنا بوصف مراحل انشاء الروبوت.

## Résumé :

Ce travail est une étude et réalisation d'un robot mobile unicycle à roues différentielles avec navigation autonome ou pour le suivi d'une trajectoire donnée.

L'étude cinématique a permis d'établir la relation entre la position et la vitesse des deux roues. La simulation est faite sur le logiciel MATLAB pour la planification des trajectoires. Pour éviter les obstacles dans l'environnement de navigation, on a utilisé un système basé sur un champ des forces virtuel. L'étude est accompagnée d'une réalisation.

## Abstract :

This work is a study for the realization of mobile robot with Differential-Drive Steering with an autonomous navigation or followed the trajectory given.

We did a kinematic study for obtain the relationship between the position and speed of both wheels. The simulation is done with the MATLAB software for planning trajectories. And we chose a new navigation system reactive based on a virtual field of forces for avoiding obstacles in its environment. And finally we have a description for our development stages of realization.

# DEDICACE

*Je dédie ce modeste travail :*

*À mon père (que Dieu ait son âme et l'accueille dans son vaste paradis) 'Un père vaut plus qu'une centaine de maîtres d'école'*

*À ma mère et À ma mère et À ma mère et encore À ma mère qui m'a éclairée mon chemin et qui m'a encouragé et soutenue toute au long de mes études.*

*À mes très chères sœurs : Yasmina, Sara et Nesrine*

*À mes grands parents et mes chères amies Amar, Mousoud, Oussama et tous mes amis de promotion.*

*Ibrahim*

# Remerciements

## Remerciements

Je remercie en premier lieu ALLAH le tout puissant de m'avoir donné la santé et le pouvoir d'accomplir ce modeste travail.

Je remercie mes parents pour leur patience, leurs encouragements et leur soutien.

Mes remerciements vont à Mr M.Bouaziz et Mr zaatrie d'avoir accepté de m'encadrer, de leurs précieux conseils et de leur disponibilité.

Je tien aussi, à remercier les membres du jury de me faire l'honneur de lire et d'évaluer ce travail.

Je remercie également tout le personnel de laboratoire des applications avancées LATA de l'université de Constantine.

Mes remerciements vont aussi à tous les enseignants du département Génie Mécanique à l'école nationale polytechnique ainsi qu'à tous les enseignants qui ont participé à notre formation.

## Table des matières

I. INTRODUCTION .....	Erreur ! Signet non défini.
II. ETAT DE L'ART .....	Erreur ! Signet non défini.
II.1 Robots mobiles a roues orientable .....	Erreur ! Signet non défini.
II.2 Robot rodem pour les déplacements de personnes handicapées.	Erreur ! Signet non défini.
II.3 Robots mobiles de type voiture .....	Erreur ! Signet non défini.
II.4 Robots mobiles unicycle (differentiel).....	Erreur ! Signet non défini.
II.5 Robots mobiles omnidirectionnels .....	Erreur ! Signet non défini.
III. L'ETAT DE L'ART EN ALGERIE.....	Erreur ! Signet non défini.
III.1. Robot mobile a roues orientables .....	Erreur ! Signet non défini.
III.2. Robots mobiles pour les personnes handicapées.....	Erreur ! Signet non défini.
IV. MODELISATION DES GRANDES CLASSES DES ROBOTS MOBILES ET LEUR MODELES CINEMATQUES .....	Erreur ! Signet non défini.
IV.1. Définitions.....	Erreur ! Signet non défini.
IV.2. Hypothèses .....	Erreur ! Signet non défini.
IV.3. Définitions d'un système non holonome .....	Erreur ! Signet non défini.
IV.4. Robots mobiles de type tricycle et de type voiture.....	Erreur ! Signet non défini.
IV.5. Robots mobiles omnidirectionnels .....	Erreur ! Signet non défini.
IV.6. Robots mobiles de type unicycle (differentielle) .....	Erreur ! Signet non défini.
V. CONCLUSION .....	Erreur ! Signet non défini.
Chapitre 2 .....	Erreur ! Signet non défini.
ETUDE D'UN PLANIFICATEURDE TRAJECTOIRE DE ROBOT .....	Erreur ! Signet non défini.
I. INTRODUCTION .....	Erreur ! Signet non défini.
II. PROBLEMATIQUE.....	Erreur ! Signet non défini.
III. MODELISATION.....	Erreur ! Signet non défini.
III.1. Récapitulation.....	Erreur ! Signet non défini.
III.2. Simulation a l'aide de logiciel MATLAB.....	Erreur ! Signet non défini.
III.2.1. Description du programme .....	Erreur ! Signet non défini.
III.3. Choix d'une courbe pour la construction de la trajectoire .....	Erreur ! Signet non défini.
IV. CONCLUSION .....	Erreur ! Signet non défini.
Chapitre 3 .....	Erreur ! Signet non défini.
SYSTEME DE NAVIGATION REACTIF .....	Erreur ! Signet non défini.
I. INTRODUCTION.....	Erreur ! Signet non défini.
II. SYSTEME DE NAVIGATION REACTIF.....	Erreur ! Signet non défini.

II.1. Description .....	Erreur ! Signet non défini.
II.2. Principe de comportement réactif .....	Erreur ! Signet non défini.
II.3. Détermination de la force attractive .....	Erreur ! Signet non défini.
II.4. Détermination de la force répulsive .....	Erreur ! Signet non défini.
II.5. Gestion de l'avance linéaire .....	Erreur ! Signet non défini.
II.6. Gestion des rotations .....	Erreur ! Signet non défini.
II.7. Gestion de la marche arrière .....	Erreur ! Signet non défini.
II.8. Minima locaux .....	Erreur ! Signet non défini.
II.9. Simulation .....	Erreur ! Signet non défini.
III. CONCLUSION.....	Erreur ! Signet non défini.
<i>Chapitre 4</i> .....	Erreur ! Signet non défini.
REALISATION ET IMPLEMENTATION DE LA COMMANDE EFFECTIVE D'UN ROBOT MOBILE .....	Erreur ! Signet non défini.
I. INTRODUCTION .....	Erreur ! Signet non défini.
II. CONCEPTION MECANIQUE .....	Erreur ! Signet non défini.
II.1. Dimensions du robot .....	Erreur ! Signet non défini.
II.2. Dimensions des moteurs .....	Erreur ! Signet non défini.
III. CONCEPTION ELECTRONIQUE ET INFORMATIQUE :.....	Erreur ! Signet non défini.
III.1. Interface de puissance .....	Erreur ! Signet non défini.
III.2. Carte de commande .....	Erreur ! Signet non défini.
III.3. Capteurs.....	Erreur ! Signet non défini.
IV. MONTAGE FINAL .....	Erreur ! Signet non défini.
V. PROGRAMMATION DE LA CARTE ARDUINO .....	Erreur ! Signet non défini.
Conclusion générale.....	Erreur ! Signet non défini.
Perspective : .....	Erreur ! Signet non défini.

# INTRODUCTION GENERALE

Le problème de la navigation autonome d'un robot mobile en environnements naturels a suscité un intérêt croissant ces dernières années, en particulier sous l'impulsion des missions à succès d'exploration planétaire de la NASA sur Mars [1]

Les applications de la robotique mobile d'extérieur se multiplient. Outre l'exploration planétaire, on peut citer :

- L'exploration ou l'intervention dans des milieux hostiles pour l'homme, en général, (zone radioactive, polaire, présence de feu, endroit toxique ...) ou simplement difficiles d'accès,
- Les opérations militaires délicates telles que le déminage ou l'accompagnement de troupes.
- Les missions de surveillance ou de protection.
- L'aide aux personnes handicapées pour la manipulation des objets quotidiens. une application consistant à fixer le bras manipulateur sur un robot mobile autonome qui peut se rendre dans une autre pièce pour aller chercher l'objet, comme le montre cette photo.



Source [4]

Les roboticiens cherchent à augmenter progressivement le degré d'autonomie de leurs robots, jusqu'à arriver à une autonomie complète et fiable pour des missions de très longue durée.

L'augmentation de l'autonomie et de l'intelligence des véhicules atteint jusqu'à l'automobile, équipée de plus en plus de systèmes prenant des décisions de manière plus ou moins autonome en cas d'urgence, telles que corriger la trajectoire ou accentuer un freinage de sécurité, en attendant les premiers véhicules entièrement automatisés.

Le point fondamental pour un robot mobile est sa capacité à réaliser de manière effectivement autonome un déplacement d'un point à un autre (ou à une série d'objectifs), donné, par ses coordonnées  $(x, y)$  dans un environnement non parfaitement connu à priori. Ce problème, bien que largement étudié par les chercheurs depuis quelques décennies, ne peut encore être considéré comme résolu.

Réaliser cette action de déplacement implique de disposer de diverses fonctionnalités (perception, décision sur le mouvement ou la trajectoire, réalisation de ce mouvement). De nombreuses méthodes ont déjà été étudiées et développées pour cela. Toutefois, vu la diversité des situations qui peuvent être rencontrées en environnements extérieurs, aucune de ces méthodes ne peut prétendre permettre d'obtenir de bons résultats dans tous les cas. De plus, une grande variété de types de plates-formes ayant également été étudiées et conçues, une méthode donnée ne peut généralement pas s'adapter à toutes ces plates-formes. Par conséquent, il n'existe pas de méthode de déplacement universelle et idéale et il semble illusoire d'en rechercher une.

Les diverses expériences du passé sur les robots mobiles montrent d'une façon convaincante qu'un robot mobile autonome devrait disposer de plusieurs modes de déplacement, à savoir plusieurs manières de réaliser les étapes de la navigation et la locomotion, afin de pouvoir négocier le plus de situations possibles. Dès lors, conserver le caractère autonome du robot se servant de ces fonctionnalités demande qu'il soit capable de choisir en ligne de manière autonome le meilleur mode de déplacement à utiliser. Il convient bien entendu d'effectuer ce choix à partir d'une reconnaissance de la situation dans laquelle le robot se trouve, sachant les situations auxquelles sont dédiées chacun des modes. Il semble également judicieux de prendre en compte pour cette décision le comportement du robot et du mode couramment utilisé. En effet, quelle que soit la situation, il vaut mieux pour le robot trouver une autre solution plutôt qu'insister avec un mode qui ne permet pas de réaliser un déplacement satisfaisant. C'est essentiellement sur ce second aspect que se sont focalisés nos travaux.

L'objectif principal de ce Projet de Fin d'Etudes est de chercher pour un robot mobile un mouvement dans l'espace sans collisions, à partir d'une configuration initiale  $\mathbf{q}_0 = \mathbf{q}(t_0)$  à une configuration finale  $\mathbf{q}_f = \mathbf{q}(t_f)$ . Dans ce cadre, ce travail porte essentiellement sur les points suivants :

- ❖ Etude et comparaison des différents modes de détermination des trajectoires

- ❖ Etude d'un planificateur de trajectoire de robot dans un environnement statique
- ❖ Etude d'un système de navigation permettant l'évitement d'obstacles
- ❖ Réalisation et implémentation de la commande effective d'un robot mobile

## *Chapitre I*

# ETUDE LES DIFFERENTS MODES DE DETERMINATION DES TRAJECTOIRES

## I. INTRODUCTION

L'intérêt indéniable de la robotique mobile à roue est d'avoir permis d'augmenter considérablement nos connaissances sur la localisation et la navigation de systèmes autonomes. La gamme des problèmes potentiellement soulevées 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.

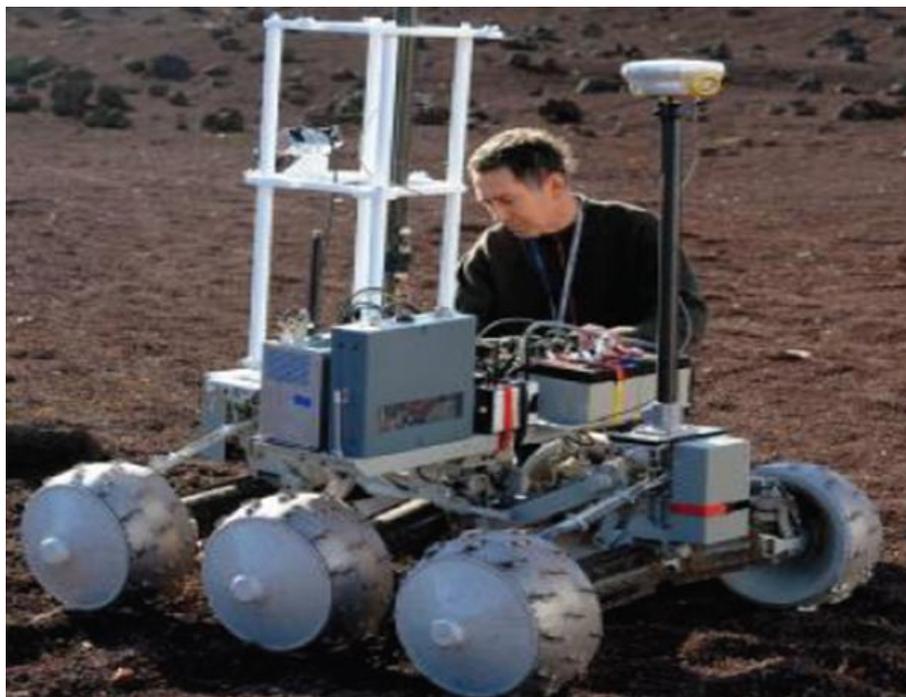
Dans ce chapitre on va voir l'état de l'art pour les robots mobiles à roues et nous allons choisir un type pour modéliser et établir ses équations cinématique.

## II. ETAT DE L'ART

### II.1 Robots mobiles a roues orientable

Ils sont des robots mobiles et ont des roues orientables pour la direction. Il en y a plusieurs. Citons quelques types :

- Robot toulousain destiné à explorer Mars.



Le robot Toulousain  
*Figure 1.1 Source [5]*

Le Centre National d'Etudes spatiales de Toulouse a développé un prototype totalement autonome. Le robot est à six roues et est destiné à étudier la composition chimique du sol martien. C'est un curieux engin, comme le châssis d'une voiturette qui aurait été montée sur six roues crantées, toutes motorisées et orientables.

- Robot IARES (c'est le nom du rover).

Il est doté d'un super-cerveau, des logiciels embarqués capables de guider le moindre de ses mouvements. La trajectoire du robot, totalement modélisée, ne laisse aucune place au hasard, alerte le véhicule de la moindre aspérité, du moindre caillou, du moindre coup de vent qui pourrait modifier le paysage de dunes martiennes. La caméra qui domine le robot offre une vision stéréo, construit une carte en trois dimensions du terrain et des obstacles. À partir de cette connaissance, le croisement d'autres algorithmes permet de définir une trajectoire et de piloter les mouvements [5].

## II.2 Robot rodem pour les déplacements de personnes handicapées

Robot Rodem est un nouveau type de véhicule mobile robotisé du style du Segway car il peut se contrôler via les mouvements du corps. C'est la société japonaise TMSUK qui en est le constructeur.

Il est destiné principalement aux personnes handicapées et les personnes âgées en perte d'autonomie. Mais il peut aussi servir de véhicule de déplacement en intérieur par toute personne.

Le Rodem se conduit soit avec une manette de type joystick soit avec un balancement du corps pour lui indiquer la direction à prendre [6]



**Figure 1.2 Source [6]**

### **II.3 Robots mobiles de type voiture**

On parle de robot dès lors que la voiture considérée est autonome, 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, basées sur des voitures de série instrumentées.



**(a) Robot Kanade**

Projets de voitures autonomes à l'université de Carnegie Mellon

*Figure 1.3 Source [2]*



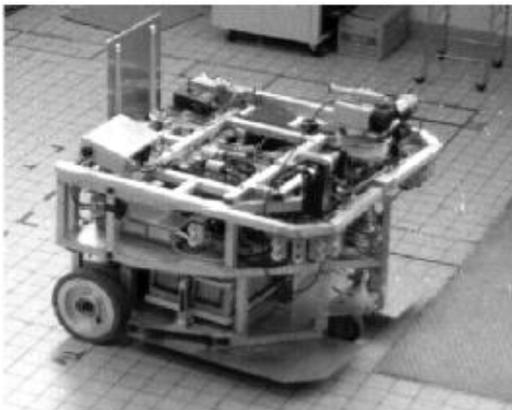
**Figure 1.4**

L'auto est développée par l'[IFSTTAR](#), l'Institut français des sciences et technologies des transports *Source [7]*

Le prototype de Renault Espace modifié est capable de répondre à certains tests, comme celui d'un piéton qui apparait devant lui. En bourrant le véhicule de capteurs, d'un GPS et de 4 caméras, les chercheurs peuvent contrôler la direction de l'auto, son comportement, sa trajectoire et sa vitesse.

## II.4 Robots mobiles unicycle (différentiel)

Ils sont des robots actionnés par deux roues indépendantes pour la direction et possédant éventuellement un certain nombre de roues folles assurant sa stabilité.



**Hilare, LAAS-CNRS, Toulouse, 1977 [Giralt 84]**

**Entraînement :** batteries au plomb 24V, 2

moteurs DC avec codeurs incrémentaux

**Calculateur :** 4 processeurs Intel 80286, pas d'OS, multibus, modem série radio a9600

bauds **Capteurs :** odométrie, 16 capteurs US, un télémètre laser

**Dimensions (L | h) :** 80cm 80cm 60cm

**Poids:** 400 kg

**Figure 1.5 Source [2]**



**Figure 1.6 source [2]**

**Pioneer P3-DX, Active Média Robotics, 2004 [ActivMedia04]**

**Entraînements :** batteries 252Wh, 2 moteurs DC avec codeurs incrémentaux

**Calculateur :** microcontrôleur Hitachi HS-8, I/O Bus, 2 ports série

**Capteurs :** odométrie, 8 capteurs US en façade + options (bombers, télémètre laser, gyroscope)

**Autonomie :** 24 - 30 h

**Vitesse :** maximum 1; 6 m/s

**Dimensions (L \_ l \_ h) :** 44 cm \_ 38 \_ 22 cm

**Poids :** 9 kg (charge admissible : 23 kg)



Figure 1.7 :robot mobile (Version Pro) WiFi X80SVP Dr. Robot source [4 int]

Le robot représenté à la figure 1.7 est équipé de :

- deux moteurs 12 V avec chacun un couple de plus de 550 oz -inch (40 kg.cm), vitesse max 0,75 m/sec
- d'une caméra CCD haute résolution, zoom 2,6x, panoramique, inclinaison, avec liaison audio bidirectionnelle
- d'une batterie : 12 V 3700 mAh avec plus de 3 heures de fonctionnement nominal

Il est destiné à des chercheurs développant des applications avancées de robotique Il peut porter une charge utile additionnelle de 15 kg (33 lb) Source [8]

## II.5 Robots mobiles omnidirectionnels

Ils sont des robots mobiles tels qu'on peut agir indépendamment sur les vitesses de translation selon les axes x, y et de rotation, autour de z, car ils ont des roues suédoise.



Figure 1.8. Exemples de robots mobiles omnidirectionnels :  
(a) Robot omnidirectionnel NOMAD muni de roues décentrées orientables.  
(b) Robot omnidirectionnel muni de roues suédoises.

Source [2]

### III. L'ETAT DE L'ART EN ALGERIE

L'Algérie est aussi entrée dans le domaine des robots mobiles à partir de réalisations de quelque type de robots mobiles à travers le laboratoire LATA de l'Université de Constantine.



#### III.1. Robot mobile a roues orientables

Figure 1.9

*Robot mobile a roue orientable*

*Ce robot est réalisé par l'étudiant Amara. C'est un robot à roues orientable muni d'une caméra pour la navigation. Source LATA (Laboratoire des Applications des Technologies Avancées de l'Université de Constantine)*



Figure 1.10

*Robot mobile à roues orientables réalisé par l'étudiant Touaba Oussama*



### Compétition Nationale de Robotique : Eurobot Algeria 2010

Il s'agit d'une compétition de robotique amateur destinée aux étudiants des grandes écoles et universités Algériennes. Elle est aussi ouverte à d'autres équipes organisées dans un cadre reconnu telles que les maisons de jeunes, associations scientifiques ou culturelles.

### III.2. Robots mobiles pour les personnes handicapées



*Figure 1.11*

#### *Chaise mobile pour les handicapés*

Chaise mobile pour les personnes handicapées réalisé par l'étudiant Touaba Oussama

Ce travail est un grand pas pour réaliser cette type de chaise ici en Algérie et évite leur l'importation et ses coûts très chers.

## IV. MODELISATION DES GRANDES CLASSES DES ROBOTS MOBILES ET LEUR MODELES CINEMATIQUES

Nous allons modéliser les grandes classes des robots mobiles que nous avons vu précédemment. Nous commençons par les robots mobiles tricycle, de type voiture, omnidirectionnel, et finalement robot mobile unicycle.

### IV.1. Définitions

On note  $R = (O, x, y, z)$  un repère fixe quelconque, dont l'axe  $z$  est vertical et  $R' = (O', x', y', z')$  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 à la figure 1.9 [2]

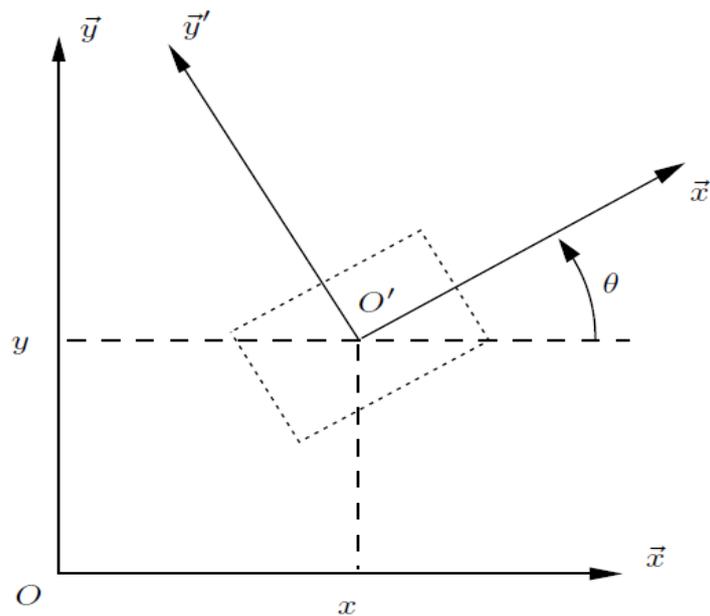


FIGURE 1.12

### IV.2. Hypothèses

La problématique de la commande des robots mobiles étant trop vaste pour pouvoir être présentée de façon exhaustive, nous introduisons dans cette partie un certain nombre d'hypothèses simplificatrices :

- Les véhicules sont considérés comme rigides (indéformables) et évoluant sur un plan.
- Les véhicules sont dotés de roues conventionnelles : le point de contact entre la roue et le sol est réduit à un point I et la roue est soumise à la contrainte du roulement sans glissement.
- On néglige l'accélération pour simplifier les équations.

#### IV.2.1 Roulement sans glissement

*La locomotion à l'aide de 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 sont indéformables, de rayon r .

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 r.s.g. 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, l'angle de rotation propre de la roue et l'angle entre le plan de la roue et le plan ( O , x , z ) comme indiqué a la figure 1.13. La nullité de la vitesse relative  $\vec{V}_Q$  roue/sol au point de contact permet d'obtenir une relation vectorielle entre la vitesse  $\vec{V}_P$  du centre P de la roue et le vecteur vitesse de rotation de la roue

$$\vec{V}_Q = \vec{V}_P + \vec{\omega} \wedge \overrightarrow{PQ}$$

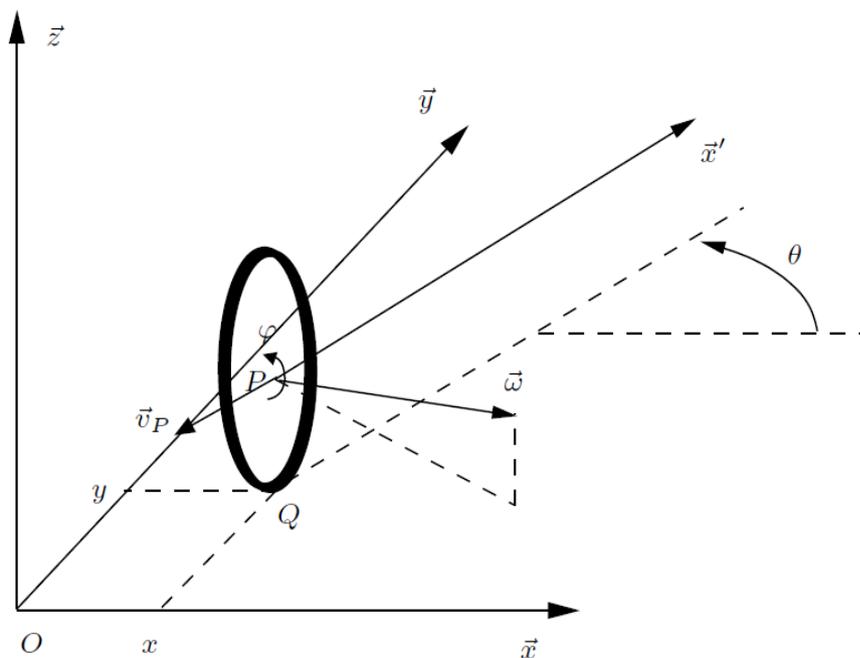


FIGURE 1.13

Tel que :  $\vec{V}_P = \begin{pmatrix} \bullet & \bullet \\ x, y, 0 \end{pmatrix}_{R_0}$

$$\overrightarrow{PQ} = -r \vec{z}$$

r : rayon de la roue

$$\vec{\omega} = \begin{pmatrix} \bullet & \bullet \\ 0, \varphi, \theta \end{pmatrix}_{R'}$$

$\omega$  : est la vitesse angulaire de la roue par rapport au repère absolu.

En écrivant ce vecteur dans le repère absolu, donc :

$$\vec{\omega}_{R_0} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \bullet \begin{pmatrix} 0 \\ \dot{\varphi} \\ \dot{\theta} \end{pmatrix}_{R'} = \begin{pmatrix} \dot{\varphi} \sin \theta \\ \dot{\varphi} \cos \theta \\ \dot{\theta} \end{pmatrix}_{R_0}$$

On aura :

$$\vec{V}_Q = \left( \dot{x} \vec{x} + \dot{y} \vec{y} \right) + \left( \dot{\theta} \vec{z} + \dot{\varphi} (\sin \theta \vec{x} + \cos \theta \vec{y}) \right) \wedge (-r \vec{z}) = 0$$

$$(\dot{x} + r \dot{\varphi} \cos \theta) \vec{x} + (\dot{y} + r \dot{\varphi} \sin \theta) \vec{y} = 0$$

Ceci nous donne le système de contraintes scalaires :

$$\dot{x} + r \dot{\varphi} \cos \theta = 0 \dots \dots (1.1)$$

$$\dot{y} + r \dot{\varphi} \sin \theta = 0 \dots \dots (1.2)$$

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 :

$$-\dot{x} \sin \theta + \dot{y} \cos \theta = 0 \dots \dots (1.3)$$

$$\dot{x} \cos \theta + \dot{y} \sin \theta = -r \dot{\varphi} \dots (1.4)$$

Ces contraintes traduisent le fait que le vecteur  $\vec{V}_p$  soit dans le plan de la roue et ait pour module  $r \dot{\varphi}$  [2]

### IV.3. Définitions d'un système non holonome

Soit un système de configuration  $\mathbf{q}$  soumis à des contraintes indépendantes sur les vitesses

$$A^t(\mathbf{q}) \dot{\mathbf{q}} = 0.$$

S'il n'est pas possible d'intégrer l'une de ces contraintes, elle est dite non intégrable ou non holonôme.

Contraintes non holonômes : certains mouvements instantanés interdits [2].

## IV.4. Robots mobiles de type tricycle et de type voiture

### IV.4.1. Description

L'architecture d'un robot mobile tricycle est représentée dans la figure 1.11. 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 du robot dépend de deux actions : la vitesse longitudinale et l'orientation de la roue orientable [2].

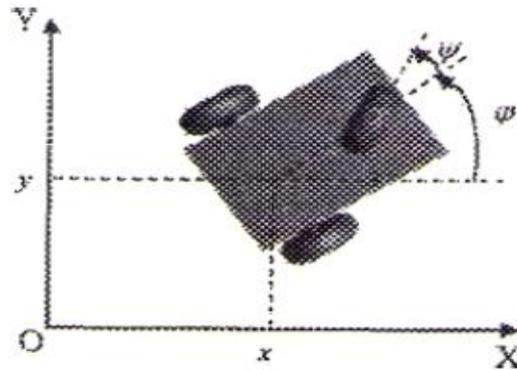


FIGURE 1.14 : Principe d'un robot mobile tricycle

### IV.4.2. Modélisation de robot tricycle

Le CIR du robot se situe à l'intersection des axes des roues fixes et de la roue orientable [2].

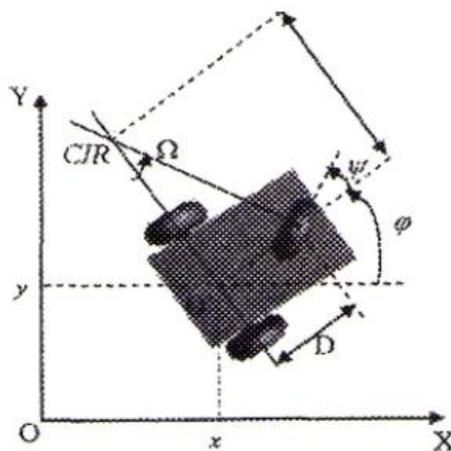


FIGURE 1.15 : CIR du robot mobile de type tricycle

$\rho$  : Rayon de braquage

$\Omega$  : la vitesse angulaire de robot par rapport de CIR

$D$  : distance entre l'axe des roues arrière et le centre de la roue orientable

$\psi$  : l'angle de braquage de la roue avant.

On peut déterminer  $\rho$  de manière géométrique à partir de l'angle d'orientation  $\psi$  de la roue avant et la vitesse de rotation autour du CIR à partir de la vitesse linéaire  $v$  du robot.

Les expressions de  $\rho$  sont données comme suit :

$$\rho = \frac{D}{\tan \psi} \dots\dots\dots (1.5)$$

$$\Omega = \frac{v}{D} \tan(\psi) \dots\dots\dots (1.6)$$

La vitesse linéaire  $V$  peut être exprimée en fonction de la vitesse linéaire de la roue orientable  $V_s$  :

$$V = V_s \cos \psi \dots\dots\dots (1.7)$$

Ce type de robot peut se diriger en ligne droite pour  $\psi=0$  et peut tourner (en théorie) autour du point  $O'$  (sur lui-même) pour  $\psi = \frac{\pi}{2}$ .

Néanmoins, le rayon de braquage de la roue orientable impose le plus souvent des valeurs de  $-\frac{\pi}{2} < \psi < \frac{\pi}{2}$ , interdisant la rotation du robot autour de lui-même.

Puisque les roues sont similaires à celle établie dans le cas du robot différentiel, ceci permet de déterminer le modèle cinématique du robot tricycle. Toutefois, par un simple raisonnement géométrique, nous pouvons établir les équations représentant la dérivée de la position du robot de la façon suivante :

$$\dot{x} = v \cos \varphi \dots\dots\dots (1.8)$$

$$\dot{y} = v \sin \varphi \dots\dots\dots (1.9)$$

$$\dot{\varphi} = \Omega = \frac{v}{D} \tan \psi \dots\dots\dots (1.10)$$

$$\dot{\psi} = \Omega_s \dots\dots\dots (1.11)$$

$$x(t) = \int_0^t v(t) \cos(\varphi(t)) dt \dots\dots\dots (1.12)$$

$$y(t) = \int_0^t v(t) \sin(\varphi(t)) dt \dots\dots\dots (1.13)$$

$$\varphi(t) = \int_0^t \Omega(t) dt \dots\dots\dots (1.14)$$

### IV.4.3. Robot mobile de type voiture

#### IV.4.3.a. Description

Le cas des robots de type voiture est similaire à celui du tricycle. La différence se situe au niveau du train avant, qui comporte deux roues au lieu d'une seule roue au milieu. En pratique, on rencontre beaucoup plus souvent ce type de système [2].

#### IV.4.3.b. Modélisation de robot mobile de type voiture

Comme nous l'avons 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 avants d'avoir une orientation différente, comme illustré dans la figure 1.16 le roulement idéal, assurant que le CIR est bien unique, est réalisé par un système de braquage différentiel (dit d'Ackerman). par ailleurs, les roues n'ayant pas le même rayon de courbure ont des vitesses différentes [2].

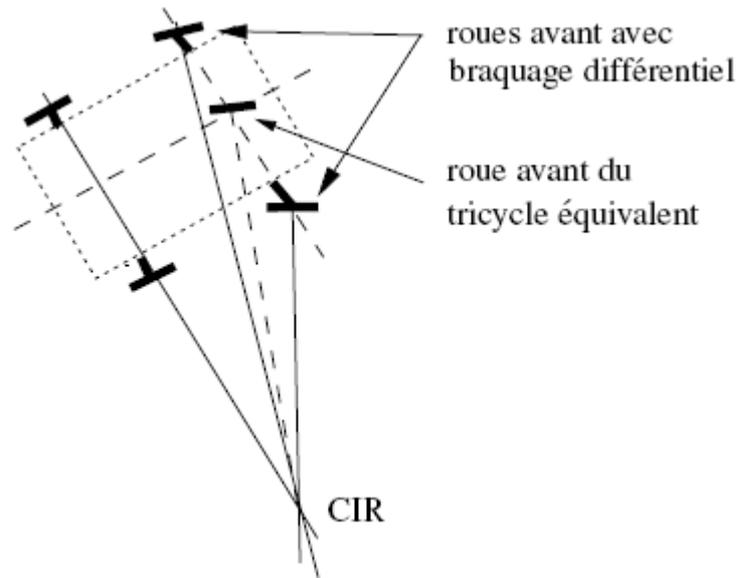


FIGURE 1.16 schéma de principe d'un robot mobile de type voiture  
Le modèle cinématique du robot de type voiture est donné par les équations suivantes :

$$\dot{x} = v \cos \varphi \dots\dots\dots (1.15)$$

$$\dot{y} = v \sin \varphi \dots\dots\dots (1.16)$$

$$\dot{\varphi} = \frac{v}{D} \tan \psi \dots\dots\dots (1.17)$$

$$\dot{\psi} = \Omega \dots\dots\dots (1.18)$$

$$x(t) = \int_0^t v(t) \cos(\varphi(t)) dt \dots\dots\dots (1.19)$$

$$y(t) = \int_0^t v(t) \sin(\varphi(t)) dt \dots\dots\dots (1.20)$$

$$\varphi(t) = \int_0^t \Omega(t) dt \dots\dots\dots (1.21)$$

## IV.5. Robots mobiles omnidirectionnels

### IV.5.1. Description

Un robot mobile est dit omnidirectionnel si l'on peut agir indépendamment sur les vitesses de translation selon les axes  $x$ , et  $y$ , et la vitesse de rotation autour de  $z$ . D'un point de vue cinématique, cela n'est pas possible avec des roues fixes ou des roues centrées orientables. On peut, en revanche, réaliser un robot omnidirectionnel ayant recours à un ensemble de trois roues décentrées orientables ou de trois roues décentrées orientables ou de trois roues suédoises aux sommets d'un triangle équilatéral

Trois roues sont donc suffisantes. Cependant, dans certains cas, une quatrième roue offre des possibilités d'optimisation et permet de rendre le système plus robuste en évitant le glissement dans le cas d'un sol qui n'est pas parfaitement plat par exemple [2].

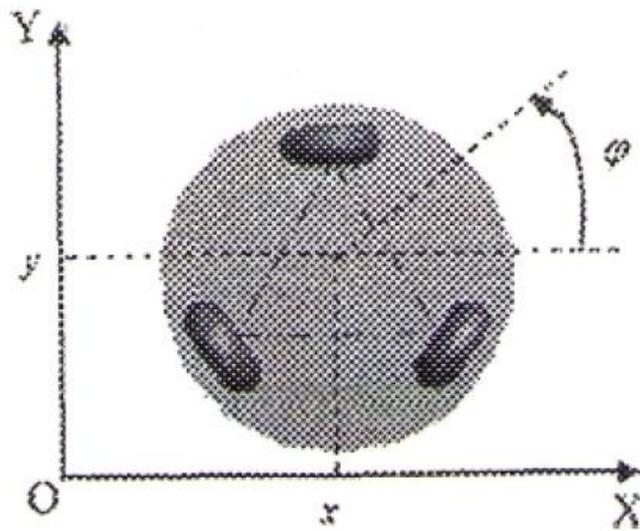


FIGURE 1.17 Schéma de principe d'un robot mobile omnidirectionnel

### IV.5.2. Modélisation de robot omnidirectionnels

Dans ce cas, on peut considérer qu'elle est d'appliquée directement la commande sur le modèle cinématique qui sera défini par les équations suivantes :

$$\begin{cases} \dot{x} = u_1 \\ \dot{y} = u_2 \\ \dot{\varphi} = u_3 \end{cases}$$

## IV.6. Robots mobiles de type unicycle (différentielle)

### IV.6.1. 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 1.18. On y a omis les roues folles, qui n'interviennent pas dans la cinématique, dans la mesure où elles ont été judicieusement placées.

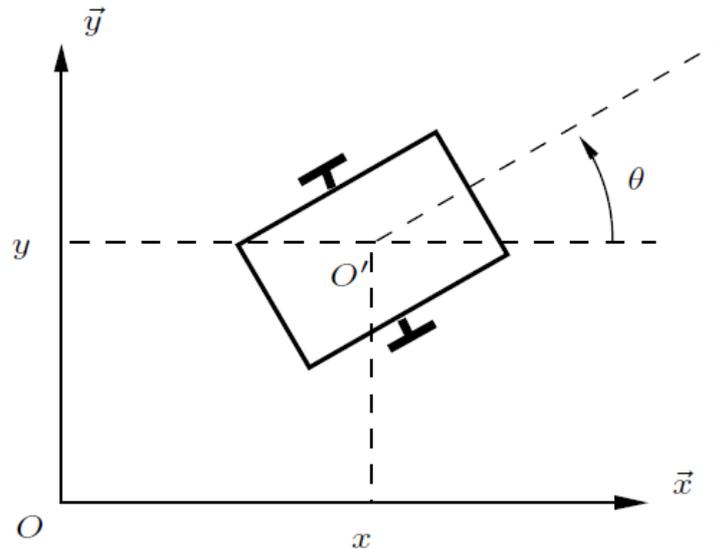


Figure 1.18

Ce type de robot est très répandu en raison de sa simplicité de construction et de propriétés cinématiques intéressantes.

Nous avons choisi le robot mobile unicycle pour modéliser car ses propriétés cinématique intéressantes.

### IV.6.2. Etude et modélisation du robot mobile unicycle

Dans ce cas de robots, les roues motrices ont le même axe de rotation. Le CIR dans ce cas est un point de cet axe.

Soit  $\rho$  le rayon de courbure de la trajectoire du robot, c'est-à-dire la distance du CIR au point C (figure 1.19).

Soit  $2L$  la distance qui sépare les deux roues A et B et  $\omega$  la vitesse angulaire du robot par rapport au CIR.

Les vitesses linéaires des roues droite et gauche, respectivement notées  $v_d$  et  $v_g$  vérifient :

$$\vec{v} = \vec{v}_g + \vec{\omega} \wedge \overline{CA} \dots\dots (1.22)$$

$$\vec{v} = \vec{v}_d + \vec{\omega} \wedge \overline{CB} \dots\dots (1.23)$$

Si nous effectuons la somme des deux équations précédentes, on trouve :

$$2\vec{v} = \vec{v}_g + \vec{v}_d + \vec{\omega} \wedge (\vec{CA} + \vec{CB}) = \vec{v}_g + \vec{v}_d$$

donc

$$v = \frac{v_d + v_g}{2} \dots\dots\dots (1.24)$$

Car les deux vecteurs  $\vec{CA}$ ,  $\vec{CB}$  sont égaux en amplitude, et opposés en sens.

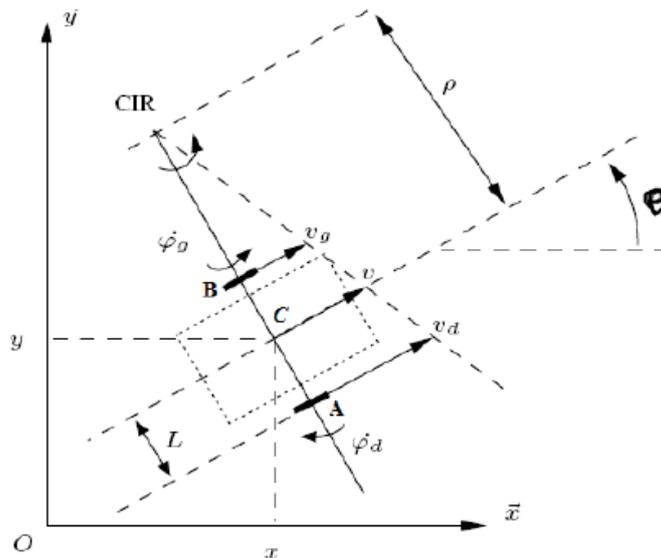


FIGURE1.19

De la même façon, si nous faisons la soustraction, nous obtenons :

$$0 = \vec{v}_g - \vec{v}_d + \vec{\omega} \wedge (\vec{CA} - \vec{CB}) \quad (\vec{CA} - \vec{CB}) = 2L$$

$$\vec{v}_d = (\rho + L)\vec{\omega}$$

$$\vec{v}_g = (\rho - L)\vec{\omega}$$

A partir de ces deux équations, on peut déterminer rho et omega en fonction de vitesse des roues :

$$\omega = \frac{v_d - v_g}{2L} = \dot{\theta} \dots\dots\dots (1.25)$$

La vitesse de rotation est égale à la vitesse de rotation autour du CIR

$$\rho = L \frac{v_d + v_g}{v_d - v_g} \dots\dots\dots (1.26)$$

Ces équations expliquent deux propriétés particulières du mouvement des robots différentiels :

Si  $v_d = v_g$  la vitesse angulaire omega sera nulle et le rayon de courbure rho est infini et le robot se déplace donc en ligne droite.

Si  $v_d = -v_g$  omega != 0 et rho est nul, alors le robot effectue une rotation sur lui-même.

Cependant, dans le cas où  $v_d \neq v_g$ , le déplacement du robot est un virage à gauche ou à droite et ceci en fonction du signe de  $V_d$  par rapport à  $V_g$  (le virage est dans une direction qui correspond à la vitesse inférieure).

Nous allons voir comment s'effectue le mouvement en coordonnées  $x$  et  $y$  et  $\theta$  en fonction du temps. A tout instant, les coordonnées  $x$  et  $y$  du point central du robot évolue en fonction de sa vitesse et l'orientation, (nous traitons l'orientation comme angle mesuré en radians  $\theta$ ). On a :

Par intégration :

$$\frac{d\theta}{dt} = \frac{v_d - v_g}{2L}$$

$$\int_{\theta_0}^{\theta} d\theta = \int_0^t \frac{v_d - v_g}{2L} dt$$

$$\theta(t) = \frac{(v_d - v_g)}{2L} t + \theta_0 \dots\dots\dots (1.27)$$

Le modèle cinématique du robot différentiel est donné par les équations suivantes :

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

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

$$\dot{x}(t) = \frac{v_d + v_g}{2} \cos \left( \frac{v_d - v_g}{2L} t + \theta_0 \right) \dots\dots\dots (1.28)$$

$$\dot{y}(t) = \frac{v_d + v_g}{2} \sin \left( \frac{v_d - v_g}{2L} t + \theta_0 \right) \dots\dots\dots (1.29)$$

Par intégration avec les conditions initiales  $x(0) = x_0$  et  $y(0) = y_0$  et  $\theta(0) = \theta_0$

Si  $V_d \neq V_g$

$$x(t) = x_0 + \frac{L(v_d + v_g)}{v_d - v_g} \left[ \sin \left( \frac{v_d - v_g}{2L} t + \theta_0 \right) - \sin \theta_0 \right] \dots\dots\dots (1.30)$$

$$y(t) = y_0 + \frac{L(v_d + v_g)}{v_d - v_g} \left[ \cos \left( \frac{v_d - v_g}{2L} t + \theta_0 \right) - \cos \theta_0 \right] \dots\dots\dots (1.31)$$

Si  $V_d = V_g$

$$\dot{x}(t) = V_d \cos \theta_0 \dots\dots\dots (1.32)$$

$$\dot{y}(t) = V_d \sin \theta_0 \dots\dots (1.33)$$

Par intégration :

$$\int_0^t \dot{x}(t) dt = \int_0^t V_d \cos \theta_0 dt$$

$$\int_0^t \dot{y}(t) dt = \int_0^t V_d \sin \theta_0 dt$$

Donc :

$$x(t) = x_0 + V_d \cos \theta_0 t \dots\dots (1.34)$$

$$y(t) = y_0 + V_d \sin \theta_0 t \dots\dots (1.35)$$

#### IV.6.2.a Récapitulation

$$\left. \begin{array}{l} \text{Si } V_d = V_g \\ x(t) = x_0 + V_d \cos \theta_0 t \\ y(t) = y_0 + V_d \sin \theta_0 t \\ \text{Si } V_d \neq V_g \\ x(t) = x_0 + \frac{L(v_d + v_g)}{v_d - v_g} \left[ \sin \left( \frac{v_d - v_g}{2L} t + \theta_0 \right) - \sin \theta_0 \right] \\ y(t) = y_0 + \frac{L(v_d + v_g)}{v_d - v_g} \left[ \cos \left( \frac{v_d - v_g}{2L} t + \theta_0 \right) - \cos \theta_0 \right] \end{array} \right\}$$

Avec l'introduction des contraintes de roulement sans glissement

$$-\dot{x} \sin \theta + \dot{y} \cos \theta = 0 \dots\dots (1.3)$$

$$\dot{x} \cos \theta + \dot{y} \sin \theta = -r \dot{\varphi} \dots (1.4)$$

#### IV.6.2.b Programmation sur MATLAB

À partir de l'équation d'un cercle  $(x - x_0)^2 + (y - y_0)^2 = \rho^2$  et à partir de l'équation (1.26), on calcule le rayon de braquage :

$$\rho = L \frac{v_d + v_g}{v_d - v_g}$$

$$(x - x_0)^2 + (y - y_0)^2 = L^2 \left( \frac{v_d + v_g}{v_d - v_g} \right)^2$$

$$(y - y_0)^2 = L^2 \left( \frac{v_d + v_g}{v_d - v_g} \right)^2 - (x - x_0)^2$$

$$|y - y_0| = \left[ L^2 \left( \frac{v_d + v_g}{v_d - v_g} \right)^2 - (x - x_0)^2 \right]^{\frac{1}{2}}$$

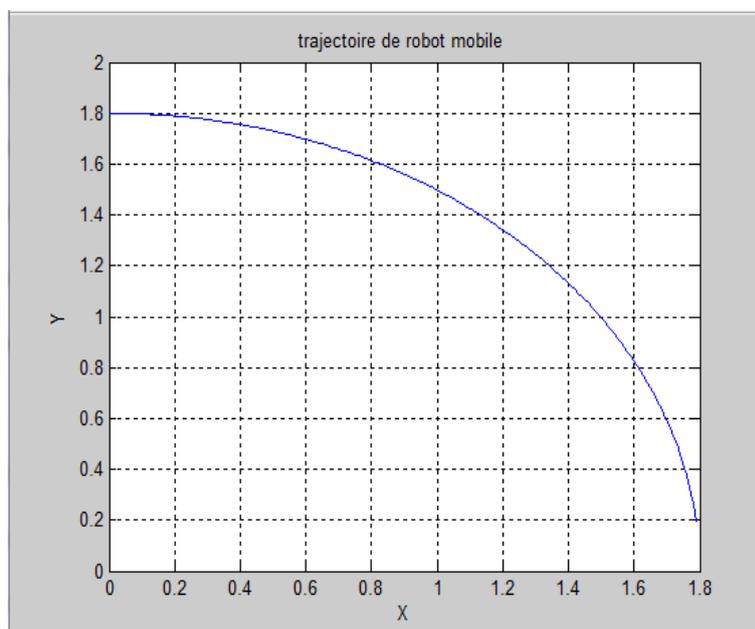
On pose  $y - y_0 \geq 0$

On programme cette équation en Matlab de type 'function' avec les entrées ( $V_d$ ,  $V_g$ ,  $L$ ,  $X_0$ ,  $Y_0$ )

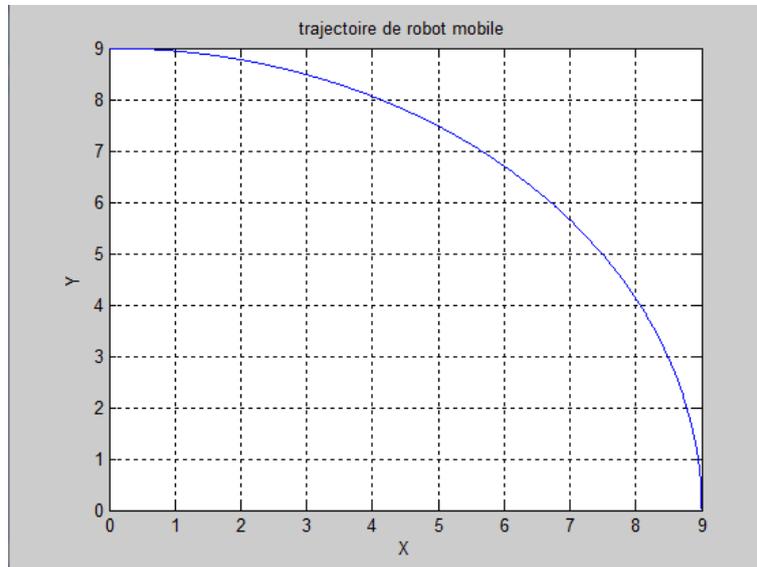
```
function [ y ] = trajectoire( vd,vg,L,yo,xo)
z=0;
i=1;
while z<=(L*(vd+vg)/(vd-vg))
    x(i)=z;
    y(i)=yo+(L^2*(vd+vg)^2/(vd-vg)^2-(x(i)-xo)^2)^0.5;
    z=z+0.01;
    i=i+1;
end
plot(x,y)
xlabel('X')
ylabel('Y')
title('trajectoire de robot mobile')
end
```

### Graphes obtenus

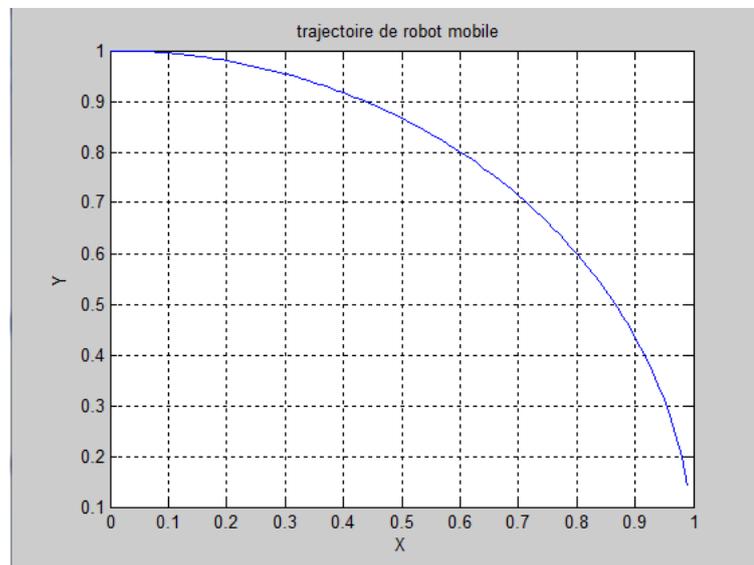
**Exemple1** : on introduit  $V_d=50\text{m/s}$ ,  $V_g=30\text{m/s}$ ,  $L=0.2\text{m}$ ,  $X_0=0$ ,  $Y_0=0$ .



**Exemple2** : on introduit  $V_d=50\text{m/s}$ ,  $V_g=30\text{m/s}$ ,  $L=1\text{m}$ ,  $X_0=0$ ,  $Y_0=0$ .

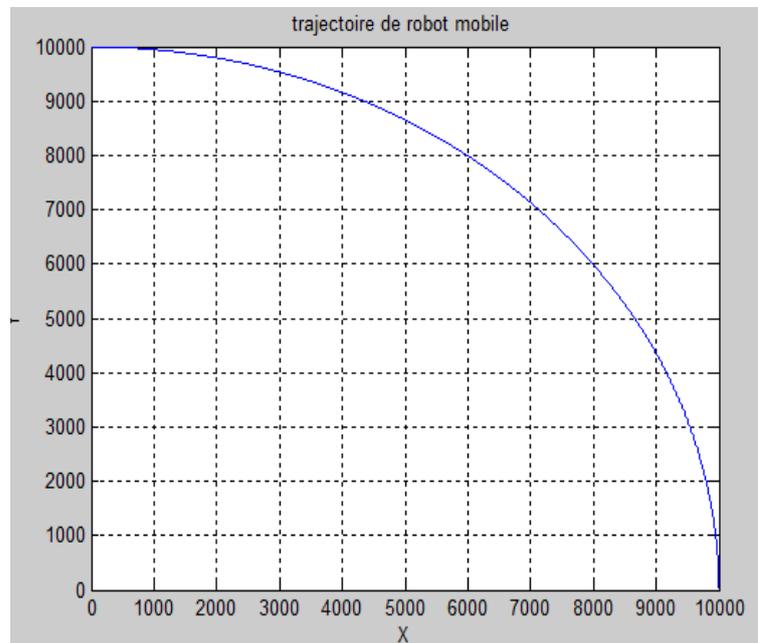


**Exemple3** : on introduit  $V_d=50\text{m/s}$ ,  $V_g=0\text{m/s}$ ,  $L=1\text{m}$ ,  $X_0=0$ ,  $Y_0=0$



On voit bien lorsque la vitesse de l'une des roues est nulle, le rayon de braquage est égal la distance entre les roues ( $L$ )

**Exemple4** : on introduit  $V_d=50\text{m/s}$ ,  $V_g=49.99\text{m/s}$ ,  $L=1\text{m}$ ,  $X_0=0$ ,  $Y_0=0$



On remarque que lorsque  $V_d \cong V_g$ , le rayon tend vers l'infini, c'est-à-dire la trajectoire devient rectiligne.

## V. CONCLUSION

Nous avons présenté dans ce chapitre les différents types de robots mobiles à roues les plus utilisés. Cette variété réside dans leur mode de locomotion qui dépend du type et de la disposition des roues utilisées. Une étude cinématique spécifique était donc nécessaire pour chaque type de robot avant toute étape de développement.

Nous avons choisi le robot mobile unicycle pour notre étude car il est très répandu, et pour ses propriétés cinématique intéressantes.

Après avoir modélisé les équations cinématiques pour le robot mobile unicycle, nous présentons notre première problématique qui traite la planification de son trajectoire. Ceci fera l'objet d'étude du chapitre suivant.

## CHAPITRE 2

# **ETUDE D'UN PLANIFICATEUR DE TRAJECTOIRE DE ROBOT**

# ETUDE D'UN PLANIFICATEUR DE TRAJECTOIRE DE ROBOT

## I. INTRODUCTION

Les robots mobiles à roues sont souvent destinés à l'exécution de tâches complexes dans de vastes espaces de travail. Afin d'améliorer l'efficacité d'un robot mobile lors de l'exécution d'une tâche donnée, il est nécessaire de déterminer la meilleure trajectoire permettant d'exécuter cette tâche.

Ce chapitre est consacré, à la formulation mathématique pour que le robot suive notre trajectoire.

## II. PROBLEMATIQUE

Dans ce travail, la planification de trajectoire est réalisée pour un robot mobile différentiel où les deux roues arrière sont commandées indépendamment. Leur différence de vitesse agit sur l'orientation du robot.

Donc nous allons chercher une loi de commande, fonction de  $V_d$ ,  $V_g$  pour amener le robot à suivre la trajectoire désirée.

## III. MODELISATION

La modélisation d'un système dynamique est une étape très importante pour la conception d'une commande telle que la commande prédictive. Nous allons proposer un modèle pour notre robot, basé sur une trajectoire rectiligne et qui considère le robot comme un point matériel [9].

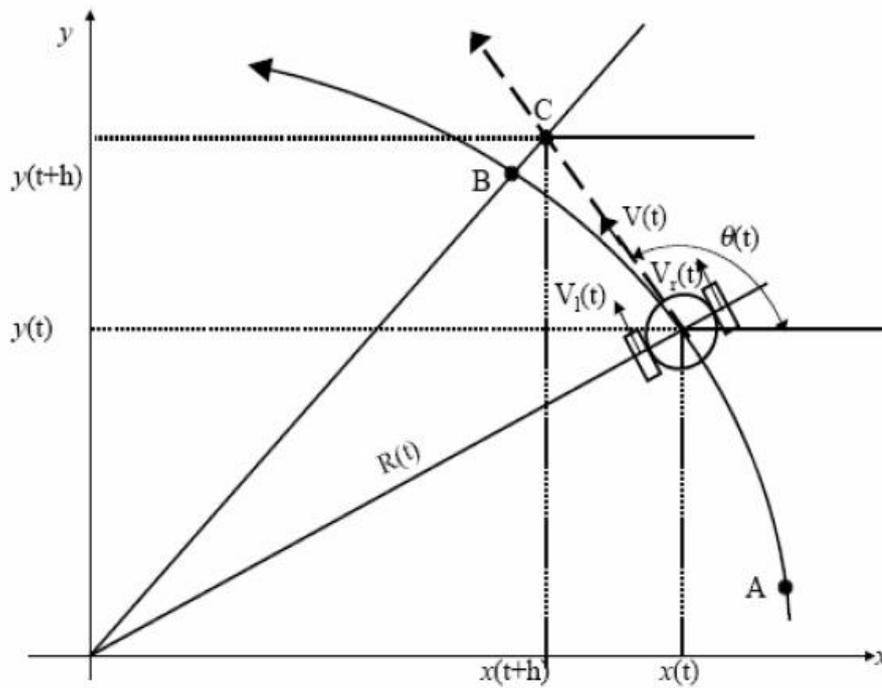


Figure 2.1

On a

$$x(i+1) = x(i) + \frac{h}{2} (V_d(i) + V_g(i)) \cos \theta(i) \dots \dots \dots (2.1)$$

$$y(i+1) = y(i) + \frac{h}{2} (V_d(i) + V_g(i)) \sin \theta(i) \dots \dots \dots (2.2)$$

$$\theta(i+1) = \theta(i) + \frac{h}{2L} (V_d(i) - V_g(i)) \dots \dots \dots (2.3)$$

Ces équations montrent qu'il s'agit d'un système multivariable qui possède deux entrées  $V_d(i)$  et  $V_g(i)$  qui sont respectivement les signaux et qui permettent d'alimenter les roues droite et gauche du robot. Il y a trois sorties qui sont  $x(i)$ ,  $y(i)$ ,  $\theta(i)$  qui sont respectivement l'abscisse, l'ordonnée et la position angulaire du robot dans le repère défini comme sur la figure précédente.

$h$  est la période d'échantillonnage  $h = t(i+1) - t(i)$  et  $2L$  est la distance entre les deux roues du robots. Le modèle d'état du robot est donné ci-dessous

$$\begin{pmatrix} x(i+1) \\ y(i+1) \\ \theta(i+1) \end{pmatrix} = \begin{pmatrix} x(i) \\ y(i) \\ \theta(i) \end{pmatrix} + \frac{h}{2} \begin{pmatrix} \cos \theta(i) & \cos \theta(i) \\ \sin \theta(i) & \sin \theta(i) \end{pmatrix} * \begin{pmatrix} V_d(i) \\ V_g(i) \end{pmatrix}$$

A partir de ces équations, on obtient :

$$V_d(i) + V_g(i) = 2 \frac{x_{i+1} - x_i}{h \cos \theta_i} \dots\dots\dots (2.4)$$

$$V_d(i) + V_g(i) = 2 \frac{y_{i+1} - y_i}{h \sin \theta_i} \dots\dots\dots (2.5)$$

$$V_d(i) - V_g(i) = 2L \frac{\theta_{i+1} - \theta_i}{h} \dots\dots\dots(2.6)$$

Tous calculs faits, on obtient :

$$V_d = \frac{(x_{i+1} - x_i)}{2h \cos \theta} + \frac{(y_{i+1} - y_i)}{2h \sin \theta} + \frac{L(\theta_{i+1} - \theta_i)}{h}$$

$$V_g = \frac{(x_{i+1} - x_i)}{2h \cos \theta} + \frac{(y_{i+1} - y_i)}{2h \sin \theta} - \frac{L(\theta_{i+1} - \theta_i)}{h}$$

### III.1. Récapitulation

$$\left\{ \begin{array}{l} V_d = \frac{(x_{i+1} - x_i)}{2h \cos \theta} + \frac{(y_{i+1} - y_i)}{2h \sin \theta} + \frac{L(\theta_{i+1} - \theta_i)}{h} \dots\dots\dots(2.7) \\ V_g = \frac{(x_{i+1} - x_i)}{2h \cos \theta} + \frac{(y_{i+1} - y_i)}{2h \sin \theta} - \frac{L(\theta_{i+1} - \theta_i)}{h} \dots\dots\dots(2.8) \end{array} \right\}$$

### III.2. Simulation a l'aide de logiciel MATLAB

Après avoir établi les équations qui serviront à la planification des trajectoires pour le robot, et avant de les implanter sur le robot, il est intéressant de les évaluer par la simulation. L'objectif de cette simulation est de vérifier si les signaux d'entrés ( $V_d, V_g$ ) aboutissent aux trajectoires désirées.

#### III.2.1. Description du programme

1<sup>ère</sup> étape : introduction des nombres des points choisis

2<sup>ème</sup> étape : Introduction des cordonnés des points choisis (ou lecture du fichier des coordonnées de ces points)

3<sup>ème</sup> étape : Introduction de la distance enter les 2 roues

4<sup>ème</sup> étape : Introduction de la vitesse maximale des roues  $V_{max}$  (contrainte sur les vitesses)

5<sup>ème</sup> étape : Calcul de la durée minimale pour parcourir la distance entre 2 points.

6<sup>ème</sup> étape : Introduction de la durée pour traverser la distance entre ces 2 points

7<sup>ème</sup> étape : Comparaison entre la durée introduite et la durée minimale. Si la première est supérieure à la seconde, on calcule les vitesses  $V_d$  et  $V_g$ , sinon on réintroduit une nouvelle valeur de la durée telle que  $t \geq t_{min}$  et on recalcule les vitesses à nouveau.

8<sup>ème</sup> étape : calcul des nouvelles coordonnées par l'injection de la vitesse calculée précédemment.

9<sup>ème</sup> étape : tracé de la trajectoire désirée

Le programme suivant établi sur Matlab reproduit ces étapes

```
n=input('donner le nombre des point')% le nombre des points choisi %
for i=1:n % boucle pour introduit les cordonnés et les tangente des points
choisis
    fprintf('x(%d)',(i-1))
    x(i)=input('=') %lire l'abscisse %
    fprintf('y(%d)',(i-1))
    y(i)=input('=') %lire l'ordonné %
    fprintf('teta(%d)',(i-1))
    teta(i)=input('=') % lire la tangente %
end
L=input('distance entre les roues=')
Vmax=input('Vitesse maximum des roues=')
for i=1:n-1
    T1=(((x(i+1)-x(i))^2+(y(i+1)-y(i))^2)^0.5)/Vmax
    t(i)=input('t=')% boucle pour introduit le temp%
    if (t(i)>=T1) %la contrainte de la vitesse maximal%
        clacule les vitesses Vd et Vg % Vd(i)=(0.5/t(i))*((x(i+1)-
x(i))/cos(teta(i))+(y(i+1)- y(i))/sin(teta(i))+L*(teta(i+1)-teta(i)))
Vg(i)=(0.5/t(i))*((x(i+1)-x(i))/cos(teta(i))+(y(i+1)-y(i))/sin(teta(i))-L*(teta(i+1)-teta(i)))
    else %si le temps est inferieur au Tmin donc donner la nouvelle t tel que t>= tmin%
        fprintf('Tmin= %f',T1)
        t(i)=input('donner la nouvel t=')
        %recalcule les vitesses pour%
```

```

    Vd(i)=(0.5/t(i))*((x(i+1)-x(i))/cos(teta(i))+(y(i+1)-y(i))/sin(teta(i))+L*(teta(i+1)-
teta(i)))
    Vg(i)=(0.5/t(i))*((x(i+1)-x(i))/cos(teta(i))+(y(i+1)-y(i))/sin(teta(i))-L*(teta(i+1)-
teta(i)))
end
end
% on va utiliser les vitesses calculées (les signaux de la commande) pour aboutisse
notre trajectoire%
for i=1:(n-1)
    x(i+1)=x(i)+(t(i)/2)*(Vd(i)+Vg(i))*cos(teta(i))
    y(i+1)=y(i)+(t(i)/2)*(Vd(i)+Vg(i))*sin(teta(i))
    teta(i+1)=teta(i)+(t(i)/L)*(Vd(i)-Vg(i))
end
plot(x,y,'-or')%tracé le graphe%
title('planification de trajectoire')
xlabel('X')
ylabel('Y')

```

### III.3. Choix d'une courbe pour la construction de la trajectoire

L'environnement dans lequel évolue notre robot ne comporte pas d'obstacles et ses limites sont rectangulaires, donc de forme simple. Il n'est pas nécessaire d'utiliser un algorithme de recherche de chemin entre les obstacles. Le robot doit juste suivre les points introduits, du point initial au point final, en respectant les contraintes cinématique et en essayant de minimiser l'erreur de position à l'arrivée.

#### III.3.1. Trajectoire 1 :

Nous avons choisi la trajectoire indiquée à la figure 2.2. Nous désirons que le robot suive cette trajectoire.

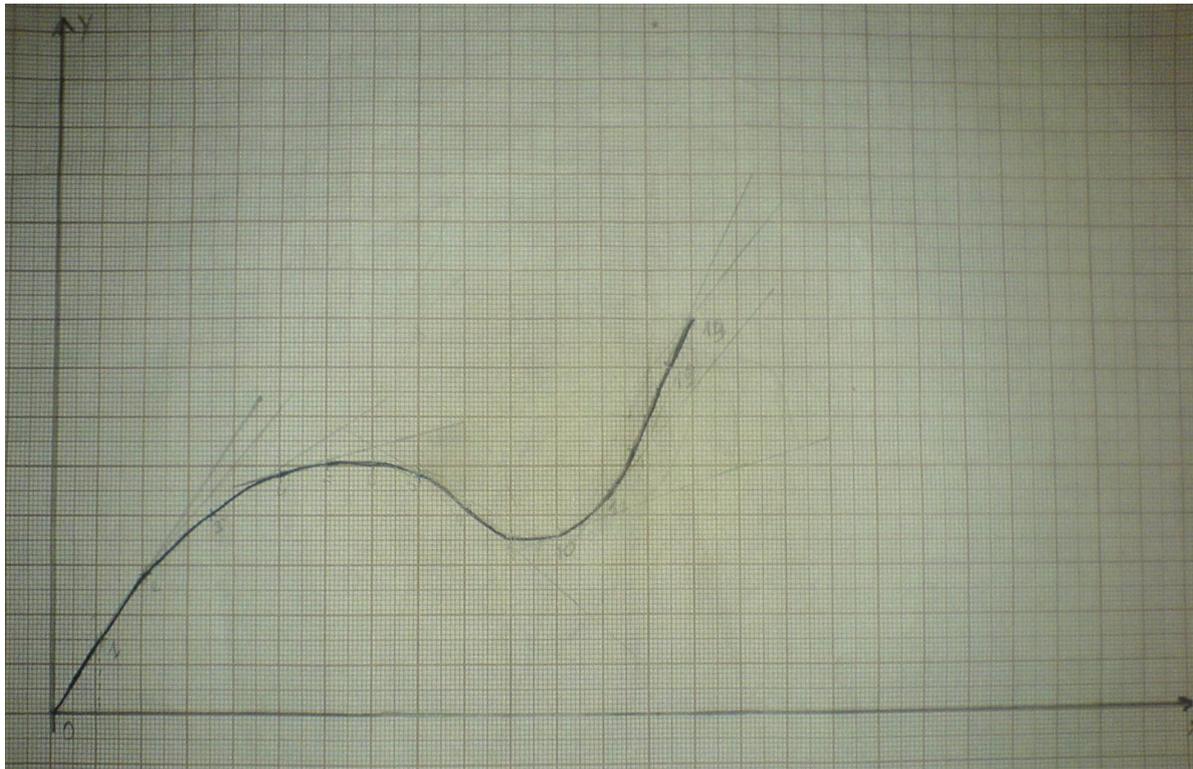


Figure 2.2

### III.3.1.a. Choix des points :

A partir de cette trajectoire, nous avons choisi les coordonnées et les tangentes des points suivants :

$$\left\{ \begin{array}{l} x_0 = 0 \\ y_0 = 0 \\ \theta_0 = 0.82 \end{array} \right\}, \left\{ \begin{array}{l} x_1 = 0.11 \\ y_1 = 0.15 \\ \theta_1 = 0.96 \end{array} \right\}, \left\{ \begin{array}{l} x_2 = 0.21 \\ y_2 = 0.28 \\ \theta_2 = 0.855 \end{array} \right\}$$

$$\left\{ \begin{array}{l} x_3 = 0.35 \\ y_3 = 0.40 \\ \theta_3 = 0.541 \end{array} \right\}, \left\{ \begin{array}{l} x_4 = 0.50 \\ y_4 = 0.48 \\ \theta_4 = 0.262 \end{array} \right\}, \left\{ \begin{array}{l} x_5 = 0.60 \\ y_5 = 0.50 \\ \theta_5 = 0.087 \end{array} \right\}$$

$$\left\{ \begin{array}{l} x_6 = 0.70 \\ y_6 = 0.50 \\ \theta_6 = 6.196 \end{array} \right\}, \left\{ \begin{array}{l} x_7 = 0.80 \\ y_7 = 0.48 \\ \theta_7 = 5.812 \end{array} \right\}, \left\{ \begin{array}{l} x_8 = 0.90 \\ y_8 = 0.42 \\ \theta_8 = 5.585 \end{array} \right\}$$

$$\left\{ \begin{array}{l} x_9 = 1.00 \\ y_9 = 0.36 \\ \theta_9 = 6.248 \end{array} \right\}, \left\{ \begin{array}{l} x_{10} = 1.10 \\ y_{10} = 0.36 \\ \theta_{10} = 0.349 \end{array} \right\}, \left\{ \begin{array}{l} x_{11} = 1.22 \\ y_{11} = 0.45 \\ \theta_{11} = 0.872 \end{array} \right\}$$

$$\left\{ \begin{array}{l} x_{12} = 1.34 \\ y_{12} = 0.70 \\ \theta_{12} = 0.96 \end{array} \right\}, \left\{ \begin{array}{l} x_{13} = 1.40 \\ y_{13} = 0.80 \\ \theta_{13} = 0.872 \end{array} \right\}$$

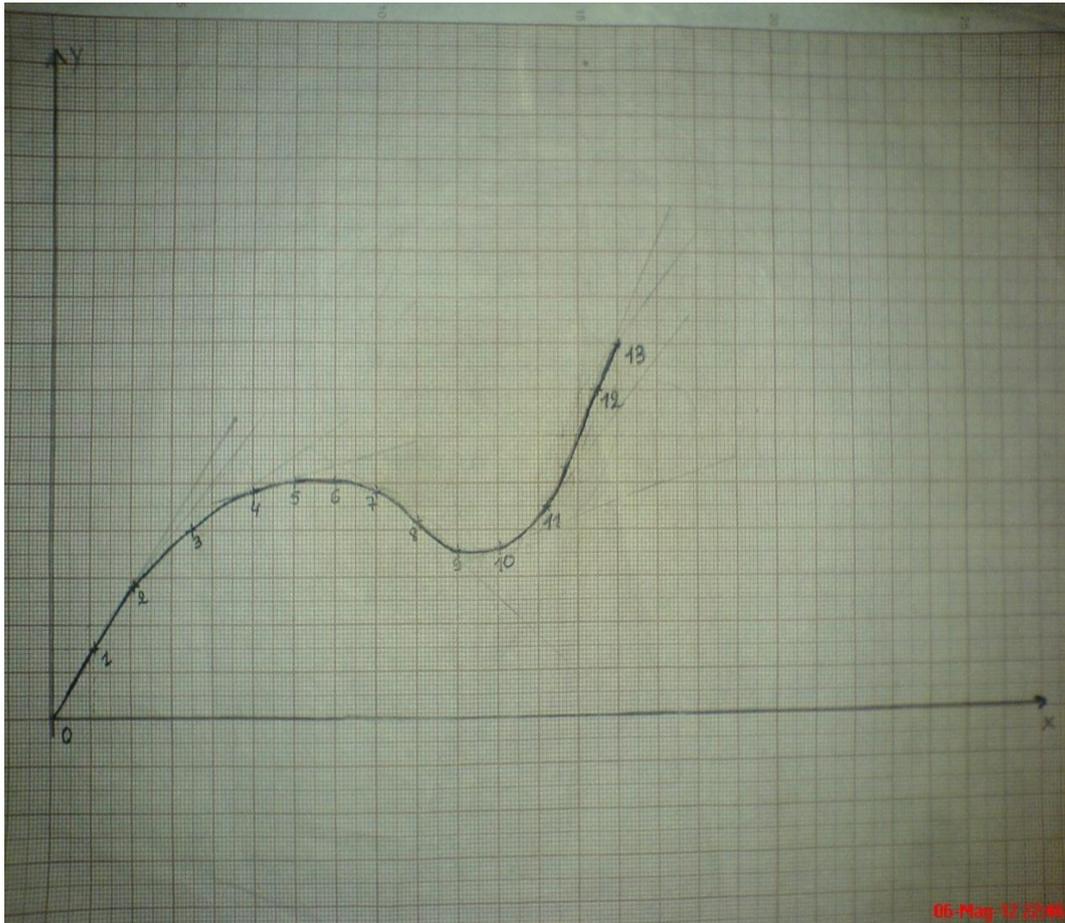


Figure 2.3

### III.3.1.b. Résultats de simulation :

Données choisies :

- distance entre les 2 roues :  $2L=0.2\text{m}$
- vitesse maximale :  $3 \text{ dm/s}$
- durée en deux points :  $2\text{s}$

Les résultats obtenus pour les vitesses (dm/s) et les coordonnées (dm) sont :

$$V_d = \begin{pmatrix} 0.9230 \\ 0.8274 \\ 0.9152 \\ 0.8119 \\ 0.4414 \\ 0.5575 \\ 0.8060 \\ 0.5997 \\ 0.5833 \\ -0.0288 \\ 1.0034 \\ 1.2871 \end{pmatrix}$$

$$V_g = \begin{pmatrix} 0.9090 \\ 0.8379 \\ 0.9466 \\ 0.8398 \\ 0.4624 \\ -0.0569 \\ 0.8444 \\ 0.6224 \\ 0.5362 \\ 0.5419 \\ 0.9511 \\ 1.2783 \end{pmatrix}$$

$$x*10 = \begin{pmatrix} 0 \\ 1.2498 \\ 2.2049 \\ 3.4267 \\ 4.8425 \\ 5.7154 \\ 6.2154 \\ 7.8594 \\ 8.9483 \\ 9.8058 \\ 10.3058 \\ 12.1424 \\ 13.7927 \\ 14.4428 \end{pmatrix}$$

$$y*10 = \begin{pmatrix} 0 \\ 1.3394 \\ 2.7036 \\ 4.1085 \\ 4.9591 \\ 5.1932 \\ 5.2192 \\ 5.0755 \\ 4.5208 \\ 3.8012 \\ 3.6856 \\ 4.3540 \\ 6.3180 \\ 7.2465 \end{pmatrix}$$

$$\theta = \begin{pmatrix} 0.8200 \\ 0.9600 \\ 0.8550 \\ 0.5410 \\ 0.2620 \\ 0.0520 \\ 6.1960 \\ 5.8120 \\ 5.5850 \\ 6.0560 \\ 0.3490 \\ 0.8720 \\ 0.9600 \\ 0.8720 \end{pmatrix}$$

## Trajectoire simulée

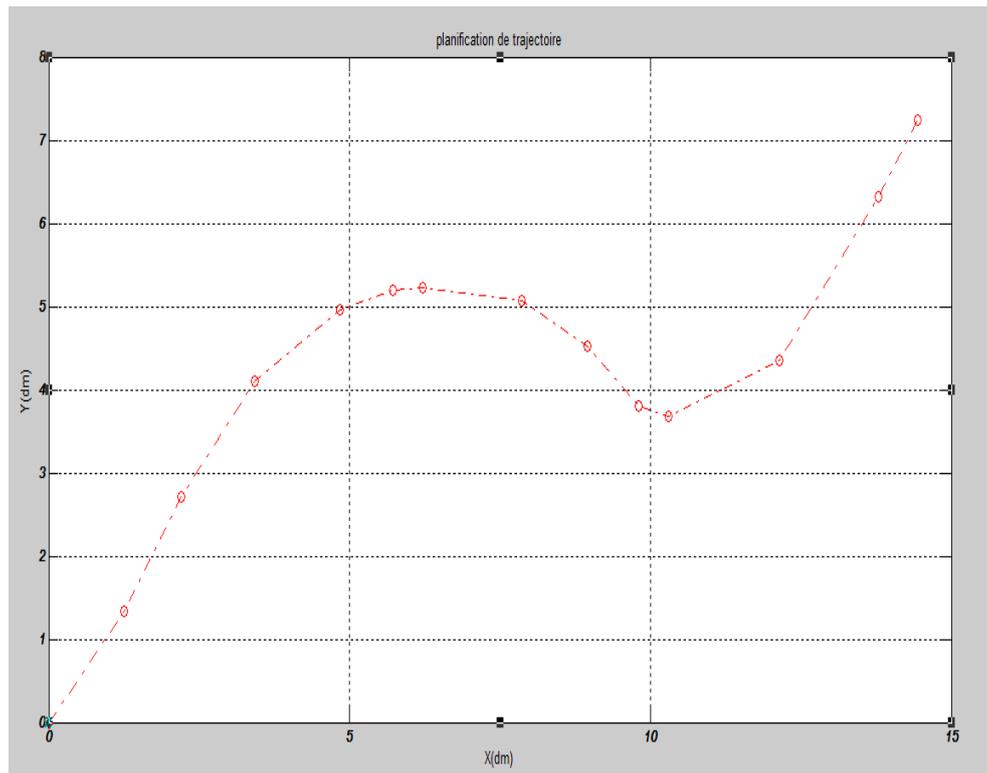


Figure 2.4

- Trajectoire choisie

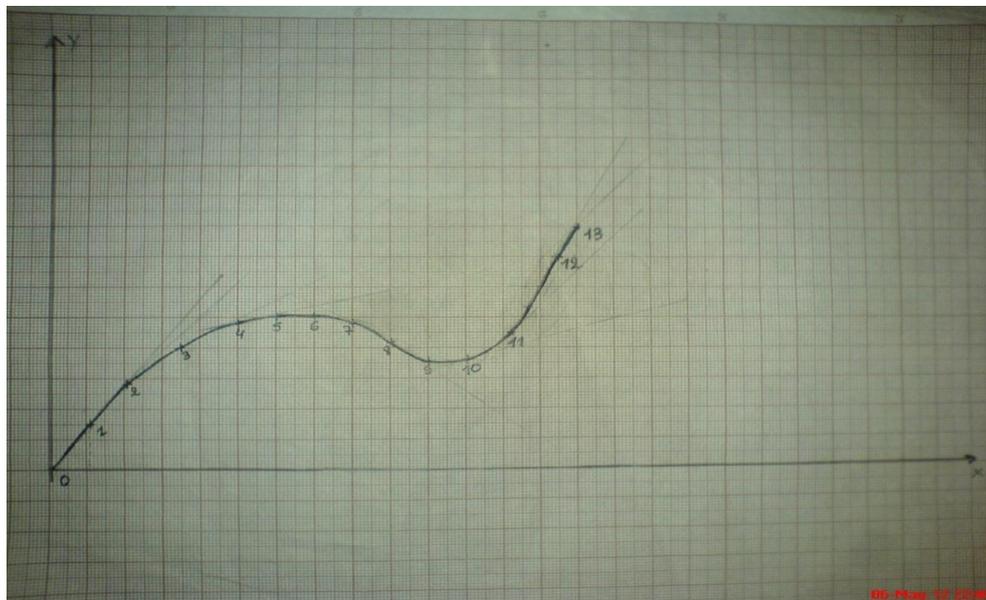


Figure 2.5

### III.3.1.c. Estimation de l'erreur

$$\varepsilon_x = |x_{\text{donné}} - x_{\text{calculé}}| = \begin{pmatrix} 0 \\ 1.1 \\ 2.1 \\ 3.5 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12.2 \\ 13.4 \\ 14 \end{pmatrix} - \begin{pmatrix} 0 \\ 1.2498 \\ 2.2049 \\ 3.4267 \\ 4.8425 \\ 5.7154 \\ 6.2154 \\ 7.8594 \\ 8.9483 \\ 9.8058 \\ 10.3058 \\ 12.1424 \\ 13.7927 \\ 14.4428 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.1498 \\ 0.1049 \\ 0.0733 \\ 0.1575 \\ 0.2846 \\ 0.7846 \\ 0.1406 \\ 0.0517 \\ 0.1942 \\ 0.6942 \\ 0.0576 \\ 0.3927 \\ 0.4428 \end{pmatrix}$$

$$\varepsilon_y = |y_{\text{donné}} - y_{\text{calculé}}| = \begin{pmatrix} 0 \\ 1.5 \\ 2.8 \\ 4 \\ 4.8 \\ 5 \\ 5 \\ 4.8 \\ 4.2 \\ 3.6 \\ 3.4 \\ 4.5 \\ 7 \\ 8 \end{pmatrix} - \begin{pmatrix} 0 \\ 1.3394 \\ 2.7036 \\ 4.1085 \\ 4.9591 \\ 5.1932 \\ 5.2192 \\ 5.0755 \\ 4.5208 \\ 3.8012 \\ 3.6856 \\ 4.3540 \\ 6.3180 \\ 7.2465 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.1606 \\ 0.0964 \\ 0.1085 \\ 0.1591 \\ 0.1932 \\ 0.2192 \\ 0.2755 \\ 0.3208 \\ 0.2012 \\ 0.2856 \\ 0.1460 \\ 0.6820 \\ 0.7535 \end{pmatrix}$$

On remarque que l'erreur maximale dans la position suivant x est de  $\varepsilon_x = 0.7846 \text{ dm} = 7.846 \text{ cm}$ , et suivant y, elle vaut  $\varepsilon_y = 0.7535 \text{ dm} = 7.535 \text{ cm}$

### III.3.2. Trajectoire 2 :

On a choisi la trajectoire représentée dans la figure suivante

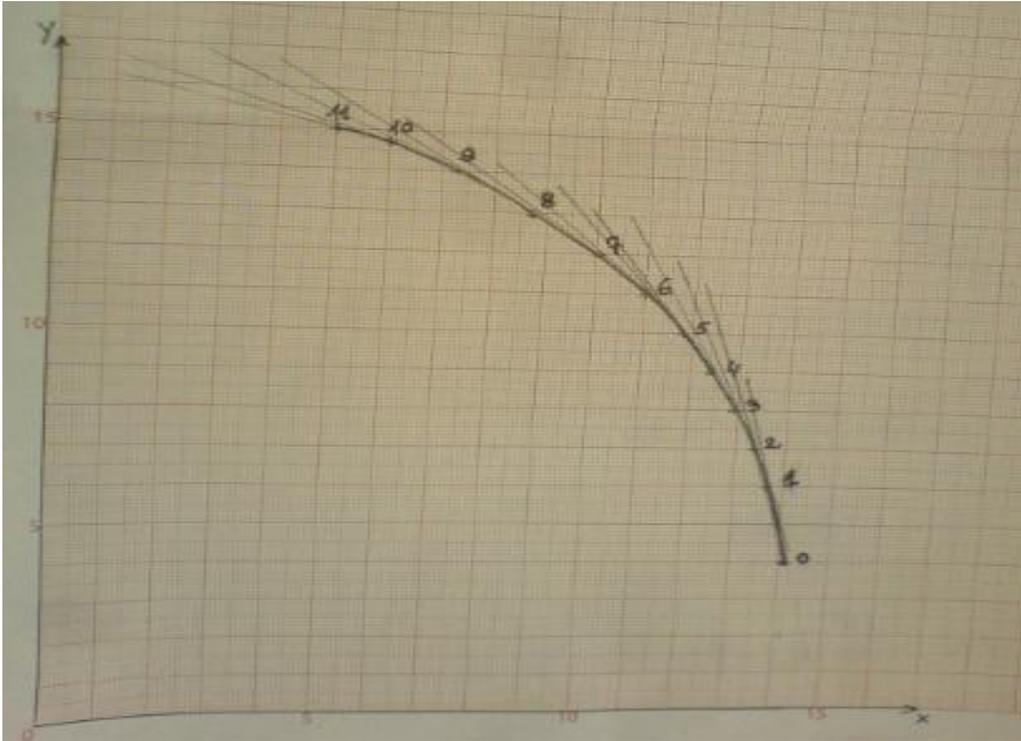


Figure 2.6

#### III.3.2.a. Choix des points

$$x_2 = \begin{pmatrix} 14.1 \\ 13.7 \\ 13.3 \\ 13 \\ 12.4 \\ 11.9 \\ 11 \\ 10 \\ 8.8 \\ 7.3 \\ 6 \\ 5 \end{pmatrix} \quad y_2 = \begin{pmatrix} 3 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 14.7 \\ 15 \end{pmatrix} \quad \theta_2 = \begin{pmatrix} 1.7802 \\ 1.8326 \\ 1.9198 \\ 2.0071 \\ 2.1293 \\ 2.1816 \\ 2.3213 \\ 2.4085 \\ 2.4958 \\ 2.6780 \\ 2.7576 \\ 2.8274 \end{pmatrix}$$

### III.3.2.b. RESULTATS DE SIMULATION

- Vitesses obtenues

$$V_{d2} = \begin{pmatrix} 1.2504 \\ 0.6504 \\ 0.4897 \\ 0.6369 \\ 0.5333 \\ 0.7045 \\ 0.7128 \\ 0.7817 \\ 0.8941 \\ 0.7587 \\ 0.4733 \end{pmatrix} \quad V_{g2} = \begin{pmatrix} 1.2452 \\ 0.6416 \\ 0.4810 \\ 0.6247 \\ 0.5281 \\ 0.6905 \\ 0.7040 \\ 0.7729 \\ 0.8758 \\ 0.7507 \\ 0.4663 \end{pmatrix}$$

$$x = \begin{pmatrix} 14.1000 \\ 13.5812 \\ 13.2476 \\ 12.9156 \\ 12.3825 \\ 11.8200 \\ 11.0200 \\ 10.0537 \\ 8.8985 \\ 7.4850 \\ 6.1349 \\ 5.2636 \end{pmatrix} \quad y = \begin{pmatrix} 3.0000 \\ 5.4411 \\ 6.6893 \\ 7.6015 \\ 8.7449 \\ 9.6450 \\ 10.7877 \\ 11.8239 \\ 12.8642 \\ 13.9294 \\ 14.6043 \\ 14.9564 \end{pmatrix} \quad \theta = \begin{pmatrix} 1.7802 \\ 1.8320 \\ 1.9198 \\ 2.0071 \\ 2.1293 \\ 2.1816 \\ 2.3213 \\ 2.4085 \\ 2.4958 \\ 2.6780 \\ 2.7576 \\ 2.8274 \end{pmatrix}$$

- **Trajectoire simulée**

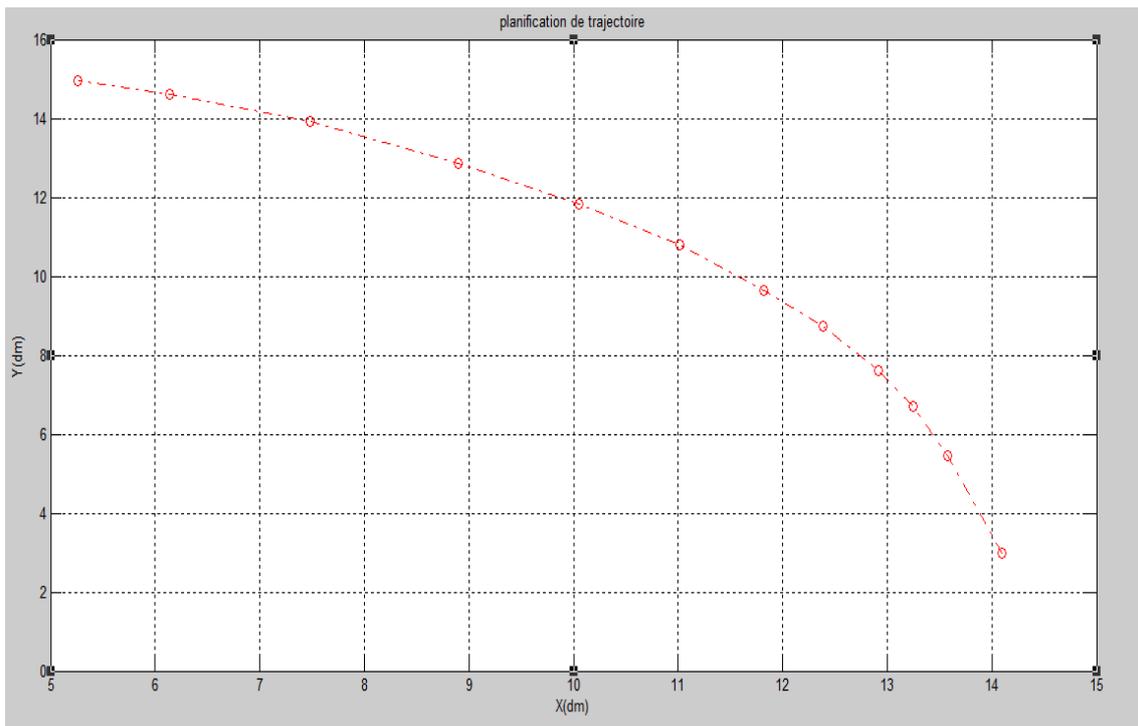


Figure 2.7

- **Trajectoire choisie**

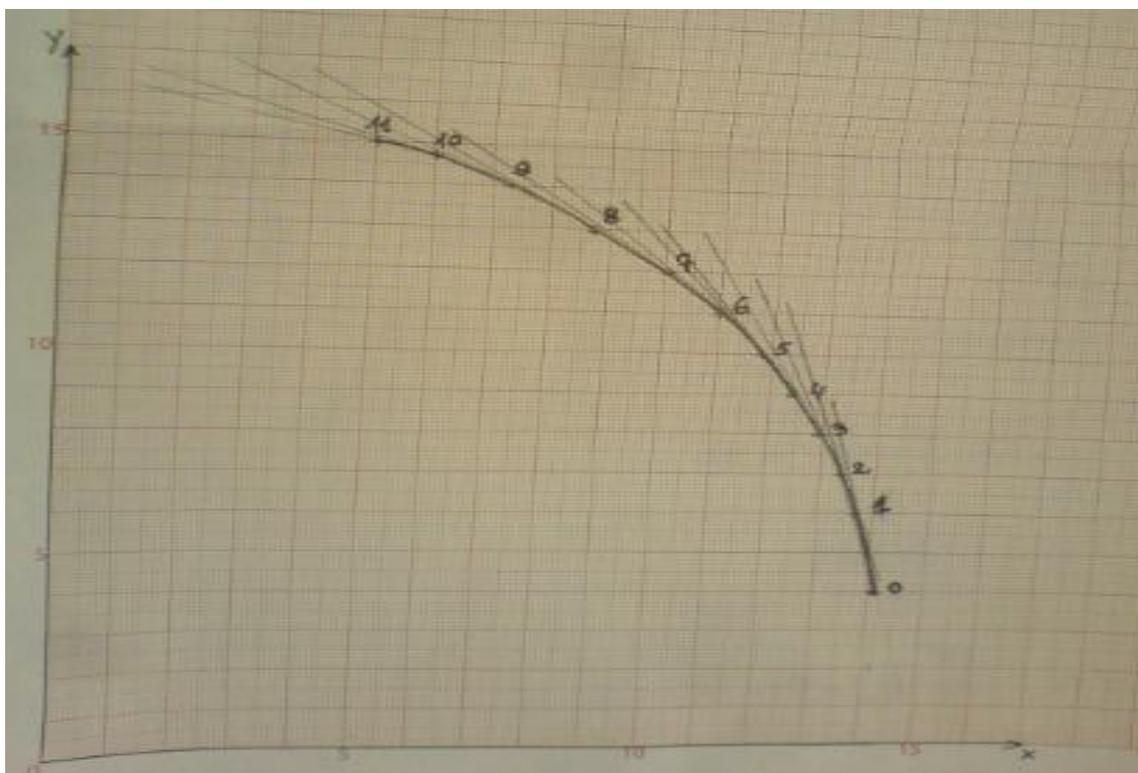


Figure 2.8

### III.3.2.c. Estimation de l'erreur

$$\varepsilon_{2x} = |x_{2\text{donné}} - x_{2\text{calculé}}| = \begin{pmatrix} 14.1 \\ 13.7 \\ 13.3 \\ 13 \\ 12.4 \\ 11.9 \\ 11 \\ 10 \\ 8.8 \\ 7.3 \\ 6 \\ 5 \end{pmatrix} - \begin{pmatrix} 14.1000 \\ 13.5812 \\ 13.2476 \\ 12.9156 \\ 12.3825 \\ 11.8200 \\ 11.0200 \\ 10.0537 \\ 8.8985 \\ 7.4850 \\ 6.1349 \\ 5.2636 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.1188 \\ 0.0524 \\ 0.0844 \\ 0.0175 \\ 0.0800 \\ 0.0200 \\ 0.0537 \\ 0.0985 \\ 0.1850 \\ 0.1349 \\ 0.2636 \end{pmatrix}$$

$$\varepsilon_{2y} = |y_{2\text{donné}} - y_{2\text{calculé}}| = \begin{pmatrix} 3 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 14.7 \\ 15 \end{pmatrix} - \begin{pmatrix} 3.0000 \\ 5.4411 \\ 6.6893 \\ 7.6015 \\ 8.7449 \\ 9.6450 \\ 10.7877 \\ 11.8239 \\ 12.8642 \\ 13.9294 \\ 14.6043 \\ 14.9564 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.5589 \\ 0.3107 \\ 0.3985 \\ 0.2551 \\ 0.3550 \\ 0.2123 \\ 0.1761 \\ 0.1358 \\ 0.0706 \\ 0.0957 \\ 0.2636 \end{pmatrix}$$

On remarque que l'erreur maximale dans la position suivant x est de  $\varepsilon_x = 0.2636 \text{ dm} = 2.636 \text{ cm}$ , et suivant y, elle est de  $\varepsilon_y = 0.5589 \text{ dm} = 5.589 \text{ cm}$

### III.3.3. Trajectoire 3

Pour cette trajectoire, nous avons choisi un arc de cercle de rayon 1.12m

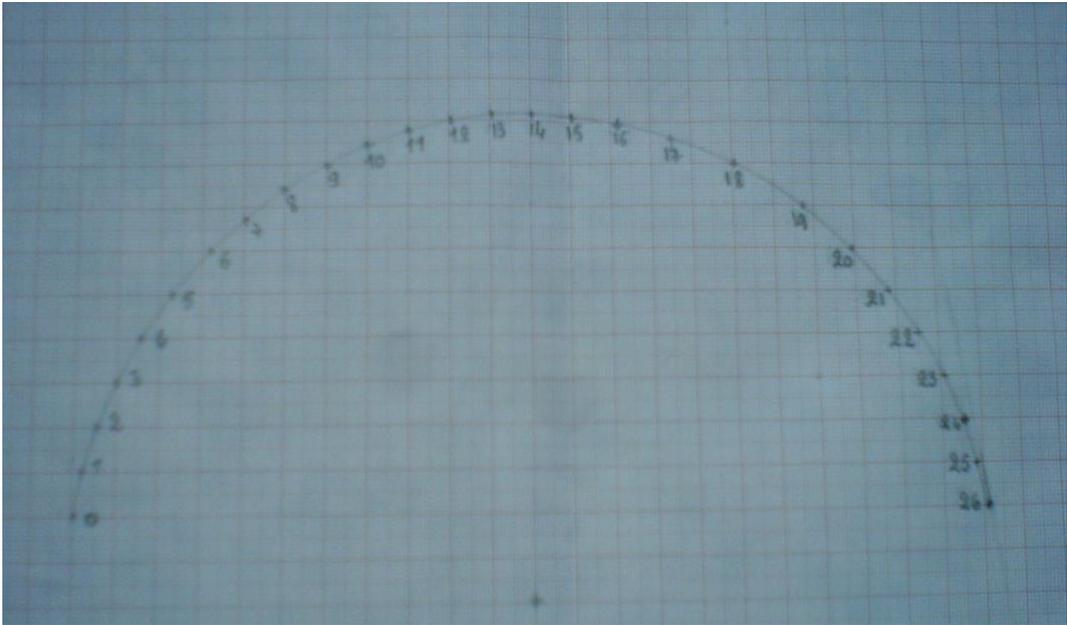


Figure 2.9



## VITESSE DES ROUES

$$V_g = \begin{pmatrix} 1.0381 \\ 1.0770 \\ 1.0930 \\ 1.1732 \\ 1.2328 \\ 1.3573 \\ 1.0738 \\ 1.2276 \\ 1.1815 \\ 1.0574 \\ 1.0048 \\ 0.9940 \\ 1.0424 \\ 1.0021 \\ 1.0004 \\ 1.1317 \\ 1.4036 \\ 1.6359 \\ 1.9004 \\ 1.6030 \\ 1.5671 \\ 1.2293 \\ 1.1913 \\ 1.0819 \\ 1.0462 \\ 1.0313 \end{pmatrix} \quad V_d = \begin{pmatrix} 1.0015 \\ 1.0770 \\ 1.0610 \\ 1.1591 \\ 1.2084 \\ 1.3334 \\ 1.0522 \\ 1.2136 \\ 1.1501 \\ 1.0888 \\ 0.9385 \\ 0.9905 \\ 1.0249 \\ 0.9602 \\ 0.9620 \\ 1.1492 \\ 1.3687 \\ 1.5835 \\ 1.8899 \\ 1.5192 \\ 1.5985 \\ 1.2119 \\ 1.1634 \\ 1.0644 \\ 1.0254 \\ 1.0313 \end{pmatrix}$$

Les coordonnées des points de simulation

$$\begin{array}{r}
 x_{3calclue} = \left( \begin{array}{l}
 3.0000 \\
 3.2000 \\
 3.6000 \\
 4.0000 \\
 4.6000 \\
 5.3001 \\
 6.2002 \\
 7.0002 \\
 8.0001 \\
 8.9995 \\
 9.9944 \\
 10.8273 \\
 11.7979 \\
 12.8126 \\
 13.7884 \\
 14.7642 \\
 15.8548 \\
 17.2107 \\
 18.7031 \\
 20.2168 \\
 21.4127 \\
 22.1065 \\
 22.8066 \\
 23.3953 \\
 23.7973 \\
 24.1002 \\
 24.2978
 \end{array} \right)
 \end{array}$$

$$\begin{array}{r}
 y_{3calclue} = \left( \begin{array}{l}
 6.0000 \\
 7.0000 \\
 8.0000 \\
 9.0000 \\
 9.9999 \\
 10.9999 \\
 11.9998 \\
 12.6998 \\
 13.3998 \\
 14.0002 \\
 14.4022 \\
 14.9026 \\
 15.1089 \\
 15.3060 \\
 15.4086 \\
 15.3060 \\
 14.9726 \\
 14.6845 \\
 14.0814 \\
 12.9410 \\
 11.9376 \\
 10.5150 \\
 9.5150 \\
 8.4954 \\
 7.5004 \\
 6.5100 \\
 5.4977
 \end{array} \right)
 \end{array}$$

$$\begin{array}{r}
 \theta_{3calclue} = \left( \begin{array}{l}
 1.3734 \\
 1.1903 \\
 1.1903 \\
 1.0303 \\
 0.9600 \\
 0.8379 \\
 0.7188 \\
 0.6108 \\
 0.5410 \\
 0.3840 \\
 0.5410 \\
 0.2094 \\
 0.1919 \\
 0.1047 \\
 -0.1047 \\
 -0.2967 \\
 -0.2094 \\
 -0.3840 \\
 -0.6457 \\
 -0.6981 \\
 -1.1170 \\
 -0.9600 \\
 -1.0472 \\
 -1.1868 \\
 -1.2740 \\
 -1.3780 \\
 -1.3780
 \end{array} \right)
 \end{array}$$

### III.3.3.c Estimation de l'erreur

$$\mathcal{E}_{3x} = |x_{\text{donné}} - x_{\text{calculé}}| = \begin{pmatrix} 3 \\ 3.2 \\ 3.6 \\ 4 \\ 4.6 \\ 5.3 \\ 6.2 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16.2 \\ 17.5 \\ 19 \\ 20.7 \\ 21.9 \\ 22.8 \\ 23.5 \\ 24.1 \\ 24.5 \\ 24.8 \\ 25 \end{pmatrix} - \begin{pmatrix} 3.0000 \\ 3.2000 \\ 3.6000 \\ 4.0000 \\ 4.6000 \\ 5.3001 \\ 6.2002 \\ 7.0002 \\ 8.0001 \\ 8.9995 \\ 9.9944 \\ 10.8273 \\ 11.7979 \\ 12.8126 \\ 13.7884 \\ 14.7642 \\ 15.8548 \\ 17.2107 \\ 18.7031 \\ 20.2168 \\ 21.4127 \\ 22.1065 \\ 22.8066 \\ 23.3953 \\ 23.7973 \\ 24.1002 \\ 24.2978 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.0001 \\ 0.0002 \\ 0.0002 \\ 0.0001 \\ 0.0005 \\ 0.0056 \\ 0.1727 \\ 0.2021 \\ 0.1874 \\ 0.2116 \\ 0.2358 \\ 0.3452 \\ 0.2893 \\ 0.2969 \\ 0.4832 \\ 0.4873 \\ 0.6935 \\ 0.6934 \\ 0.7047 \\ 0.7027 \\ 0.6998 \\ 0.7022 \end{pmatrix}$$

$$\varepsilon_{3y} = |y_{\text{donné}} - y_{\text{calculé}}| = \begin{pmatrix} 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 12.7 \\ 13.4 \\ 14 \\ 14.4 \\ 14.8 \\ 15 \\ 15.2 \\ 15.3 \\ 15.2 \\ 14.9 \\ 14.6 \\ 14 \\ 13 \\ 12 \\ 11 \\ 10 \\ 9 \\ 8 \\ 7 \\ 6 \end{pmatrix} - \begin{pmatrix} 6.0000 \\ 7.0000 \\ 8.0000 \\ 9.0000 \\ 9.9999 \\ 10.9999 \\ 11.9998 \\ 12.6998 \\ 13.3998 \\ 14.0002 \\ 14.4022 \\ 14.9026 \\ 15.1089 \\ 15.3060 \\ 15.4086 \\ 15.3060 \\ 14.9726 \\ 14.6845 \\ 14.0814 \\ 12.9410 \\ 11.9376 \\ 10.5150 \\ 9.5150 \\ 8.4954 \\ 7.5004 \\ 6.5100 \\ 5.4977 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.0001 \\ 0.0001 \\ 0.0002 \\ 0.0002 \\ 0.0002 \\ 0.0002 \\ 0.0022 \\ 0.1026 \\ 0.1089 \\ 0.1060 \\ 0.1086 \\ 0.1060 \\ 0.0726 \\ 0.0845 \\ 0.0814 \\ 0.0590 \\ 0.0624 \\ 0.4850 \\ 0.4850 \\ 0.5046 \\ 0.4996 \\ 0.4900 \\ 0.5023 \end{pmatrix}$$

On voit que l'erreur maximale suivant la position x est de  $\varepsilon_x = 0.7047 \text{ dm} = 7.047\text{cm}$ , et suivant y  $\varepsilon_y = 0.5046\text{dm} = 5.046 \text{ cm}$

#### IV. CONCLUSION

Dans le présent chapitre, nous avons présenté une méthode pour la planification de trajectoire qui consiste à faire des discrétisations linéaires de la trajectoire pour le robot mobile unicycle étudié et nous avons extrait la loi de commande des vitesses des roues.

Le prochain chapitre sera consacré à la navigation autonome du robot dans un milieu comportant des obstacles.

## Chapitre 3

# **SYSTEME DE NAVIGATION REACTIF**

## Chapitre 3

# SYSTEME DE NAVIGATION REACTIF

### I. INTRODUCTION

Le contrôle d'un robot mobile navigant dans un environnement encombré est un problème fondamental et fait l'objet d'une grande attention de la communauté robotique.

Il s'agit principalement de garantir au robot mobile un contrôle fiable lui permettant une navigation en toute sécurité loin de tout risque de collision, respect des contraintes structurelles du robot.

Nous allons étudier la navigation d'un robot mobile dans un environnement en présence d'obstacles, le robot doit atteindre la cible en évitant ces obstacles.

### II. SYSTEME DE NAVIGATION REACTIF

#### II.1. Description

- Chaque objet de l'environnement est associé à un champ de forces
- Les forces peuvent varier avec la distance, peuvent agir sur une étendue limitée
- Le robot suit le champ de forces [6]<sup>i</sup>

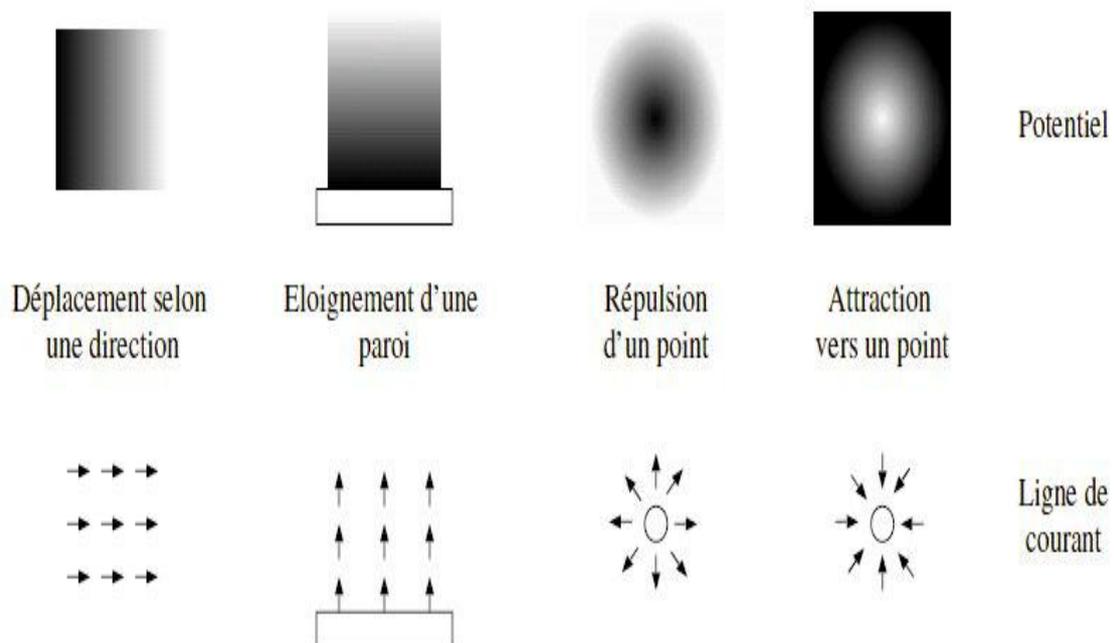


Figure 3.1 Source [10]<sup>ii</sup>

La combinaison donne la direction à prendre en chaque point de l'espace

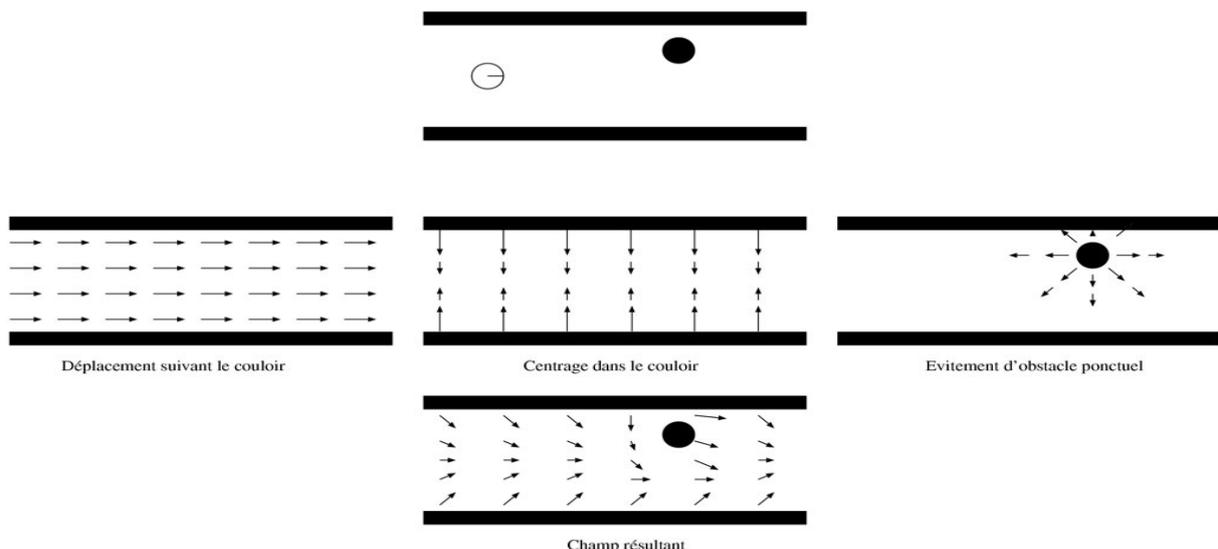


Figure 3.2 Source [6]<sup>iii</sup>

## II.2. Principe de comportement réactif

Cette méthode consiste à assimiler le robot à une particule contrainte à se déplacer dans un champ des forces, champ attractif au but et l'autre ensemble de champs répulsif modélisant la présence d'obstacles dans l'espace du robot, La direction de notre robot est déterminé par la direction du vecteur de la force résultante

$$\vec{F}_{rés} = \vec{F}_{att} + \vec{F}_{rép} \dots\dots\dots(3.1)$$

$F_{rés}$  : la force résultante

$F_{att}$  : la force attractive au but

$F_{rép}$  : la force répulsive par les obstacles

## II.3. Détermination de la force attractive

Le rôle de ce vecteur étant de permettre au robot de rejoindre le but désigné, il est tout naturellement orienté du centre du véhicule vers le point à atteindre. Plusieurs formules existent pour déterminer la force attractive mais tous ces formule ont le même principe elle peut être constant sur la totalité de l'espace ou augmente au fur et mesure que le robot s'approche.

Nous avons choisi la formule la plus stable [7]<sup>iv</sup>

$$\vec{F}_{but} = F_{cst} \left( \frac{x_{but} - x_0}{d_{but}} \vec{i} + \frac{y_{but} - y_0}{d_{but}} \vec{j} \right) \dots\dots\dots (3.2)$$

$F_{cst}$  : représente une force d'attraction constante

$(x_{but}, y_{but})$  : les coordonnées de but

$d_{but}$  : la distance séparant le robot au but

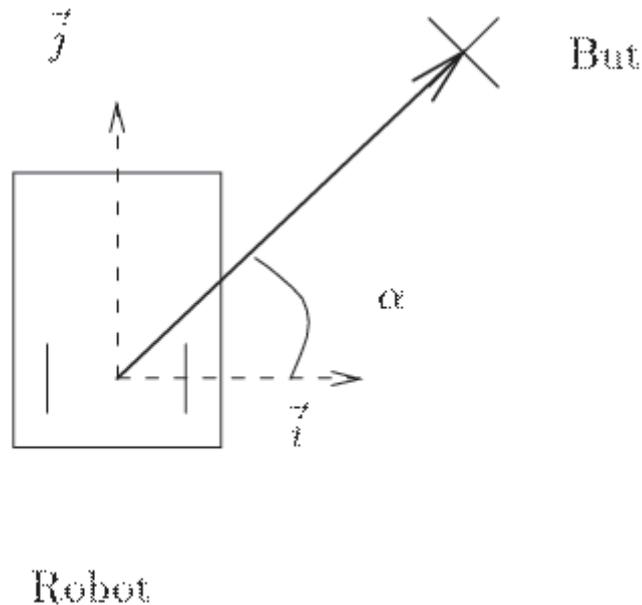


Figure 3.3

#### II.4. Détermination de la force répulsive

Le rôle de ce vecteur étant permettre le robot d'éviter les obstacles existants dans l'espace de navigation pour atteindre le but, la force totale  $F_{obstacle}$  est la somme de toutes les forces répulsives élémentaires.

D'une manière classique, le module du vecteur de force doit augmenter si on s'approche de l'obstacle et diminuer dans le cas contraire, nous avons choisi la fonction suivante : [7]<sup>v</sup>

$$F_{repi} = \frac{F_{cst}}{d_i / d_{min}} \dots\dots\dots (3.3)$$

$F_{cst}$  : force constant de répulsion

$d_i$  : la distance mesurée par le capteur i

$d_{min}$  : la distance minimale qui doit conserver entre le robot et l'obstacle

Le sens de cette force est normal et opposé par rapport à capteur qui détecte l'obstacle

$$\vec{F}^{totale\ répulsi\ f} = \sum_i \vec{F}^{rép\ i} \dots\dots\dots (3.4)$$

**II.5. Gestion de l'avance lineaire**

Si les capteurs ne détectent aucun obstacle et le but fait un angle  $\theta=0$  avec le centre de robot, ce dernier devra marcher linéairement.

**II.6. Gestion des rotations**

Les capteurs sont répartis sur la ceinture du robot de part et d'autre du centre de rotation. Supposons qu'un obstacle soit détecté sur le coté droit du véhicule. Si cet obstacle est détecté par un capteur situé en avant du centre de rotation, le véhicule devra tourner vers la gauche pour s'en écarter. Si en revanche, la détection est réalisée par un capteur situé en arrière, il devra tourner vers la droite. La direction du vecteur de force dépend donc de la position du capteur par rapport à ce centre.

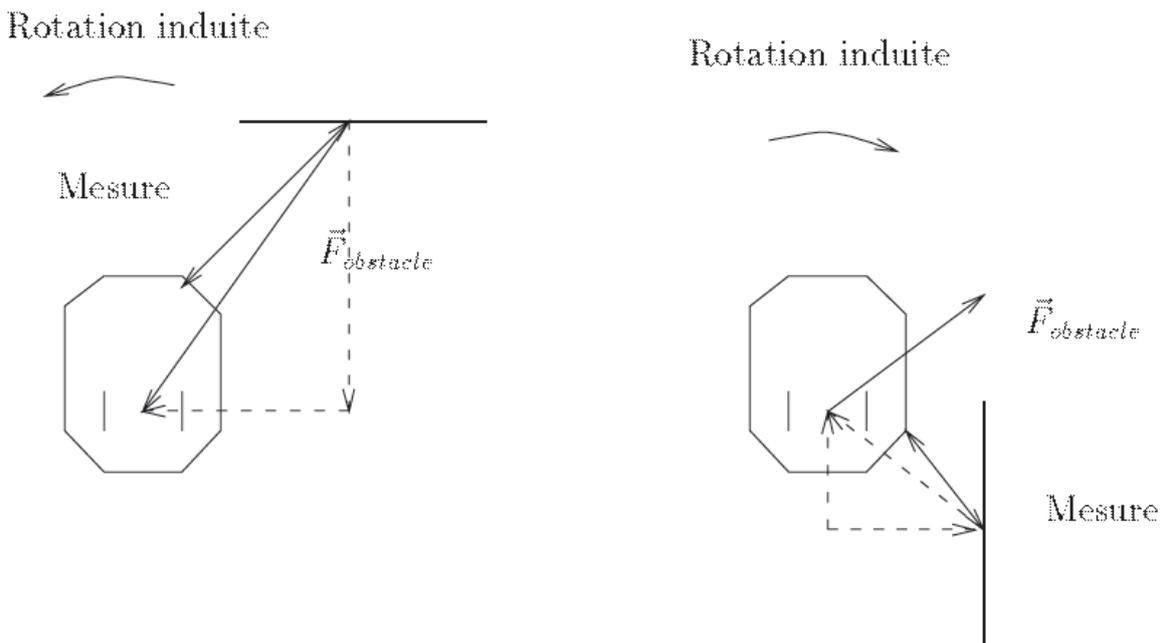


Figure 3.4 Direction des vecteurs forces associés aux obstacles en fonction du capteur responsable de la mesure

**II.7. Gestion de la marche arrière**

Si le robot se trouve dans une région étroite et ne peut pas faire un demi-tour pour ressortir, il exécute alors une marche arrière. Ce mouvement est imposé par la position du but et non par la direction du vecteur de la commande. Il sera poursuivi jusqu'à ce que les capteurs détectent suffisamment de place sur la droite ou sur la gauche du robot afin d'exécuter un demi-tour et permettre un repositionnement en marche avant.

## II.8. Minima locaux

Un des problèmes principaux des méthodes à base de champs de vecteurs est la présence de minima locaux.

Les minima locaux sont des phénomènes de blocage lorsque la somme des forces répulsives générées par l'obstacle est égale à la force attractive générée par le but. Le robot ne peut donc pas bouger (blocage).

### II.8.1. Gestion des minima locaux

Comme nous l'avons signalé, les minima locaux sont un des problèmes majeurs, le champ des forces que nous avons choisi ne réalise en aucun cas ce que l'on pourrait appeler un vecteur de navigation.

Il s'agit donc de les détecter et de mettre en place quelques heuristiques pour essayer de permettre au robot d'en ressortir.

Un minimum local correspond à un point d'équilibre stable pour le véhicule, c'est-à-dire que toute tentative de sortie de ce point provoque la génération de commandes L'y ramenant

Nous nous sommes intéressés à deux causes principales :

Le robot doit momentanément s'éloigner du but pour pouvoir l'atteindre.

Le robot doit rentrer dans un passage étroit (couloir par exemple)

#### II.8.1.a. 1ère cas

La situation de blocage peut être provoquée par exemple lorsque le robot se trouve parallèle à un mur, le but à atteindre étant situé de l'autre côté (voir figure 3.5). La direction du champ de force va osciller entre  $90^\circ$  et  $-90^\circ$ , imposant au robot de ne pas exécuter de translation

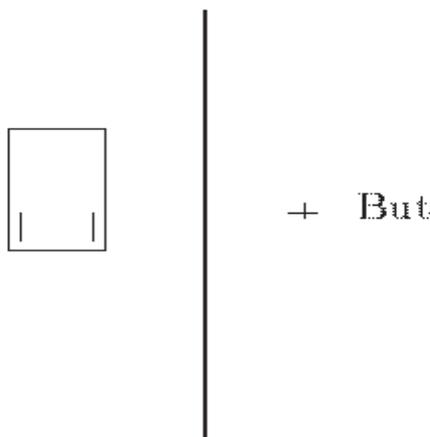


Figure 3.5 situation provoquant un minimum local

En cas de détection de présence d'un minimum local, le robot passe en mode recherche. Cela signifie que quel que soit l'angle commandé, le contrôleur impose une vitesse linéaire constante. Le robot se mettra donc en mouvement et sortira éventuellement de la zone de blocage. Ce mode recherche est supprimé :

- Si le but est atteint
- Si le robot fait de nouveau face au but,
- Si le nombre de cycles maximum autorisé en recherche est dépassé

Le dernier cas signifie que l'on suppose que le robot a quitté la zone d'influence du minimum local et on reprend le mode de fonctionnement normal.

### **II.8.1.b 2ème cas**

Dans le cas où nous avons un passage étroit (couloir) et le but se trouve dans le couloir, les forces provenant des murs sont trop fortes et créent une barrière infranchissable à l'entrée du passage (figure 3.6)

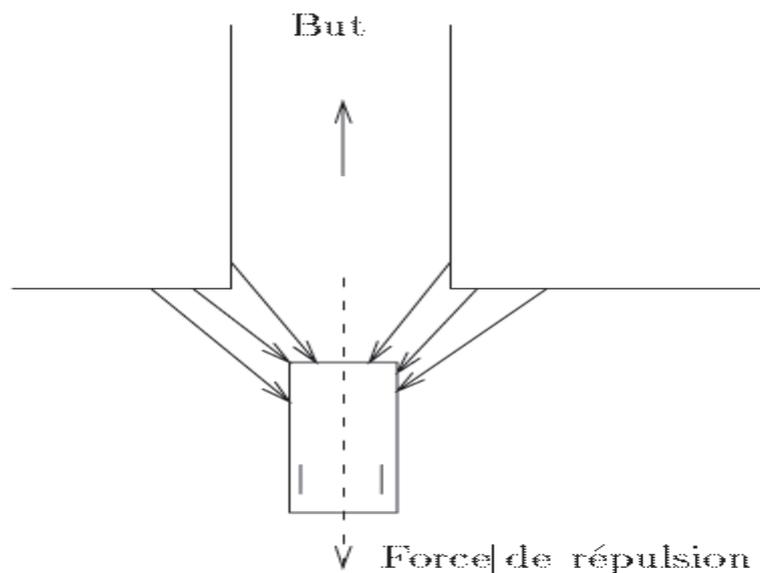


Figure 3.6

Le robot est repoussé par les deux murs

Le champ de force est alors adapté en diminuant à chaque cycle la valeur de  $F_{cst}$  jusqu'à ce que la force répulsive revienne en dessous de la force attractive  $F_{att}$

## **II.9. Simulation**

### **II.9.1. Force attractive**

D'après l'équation (3.2) la variation de la force  $F_{att}$  suit une équation de type  $1/x$

### II.9.1.a. Le programme en matlab

Nous avons fait un programme de type 'fonction' ses entrés sont les coordonnées initial de notre robot( $x_0, y_0$ ), les coordonnées du but ( $x_{but}, y_{but}$ ) et la valeur de la force attractive.

Le programme calcule la variation de la force attractive en fonction de la distance parcourue et tracer un graphe pour cette variation.

```
function [ F ] = force_att( xo,yo,xbut,ybut,Fcst )%introduire les entrés
les coordonnées initial de notre robot et les coordonnées du but+la valeur
de force%
if xo <= xbut % partager la distance parcourue%
X=linspace(xo,xbut,500)
else
X=linspace(xbut,xo,500)
end
if yo <= ybut
Y=linspace(yo,ybut,500);
else
Y=linspace(ybut,yo,500)
end

for i=1:500 %boucle pour calculer la force att%
    dis(i)=sqrt((X(i)-xbut)^2+(Y(i)-ybut)^2)
    Fx(i)=Fcst*(xbut-xo)/dis(i);
    Fy(i)=Fcst*(ybut-yo)/dis(i);
    F(i)=sqrt(Fx(i)^2+Fy(i)^2);
end

plot(dis,F,'*') %tracer les graphes%
xlabel('la distance entre le robot et le but')
ylabel('la force attractive')
title('la variation de la force attractive en fonction de la distance
parcourue')
end
```

Nous avons obtient le graphe de variation de la force attractive en fonction de la distance parcourue pour les valeurs suivantes ( $x_0=0, y_0=0$ ), ( $x_{but}=10, y_{but}=20$ ), et la valeur de la force constante  $F_{cst}=1000$  N

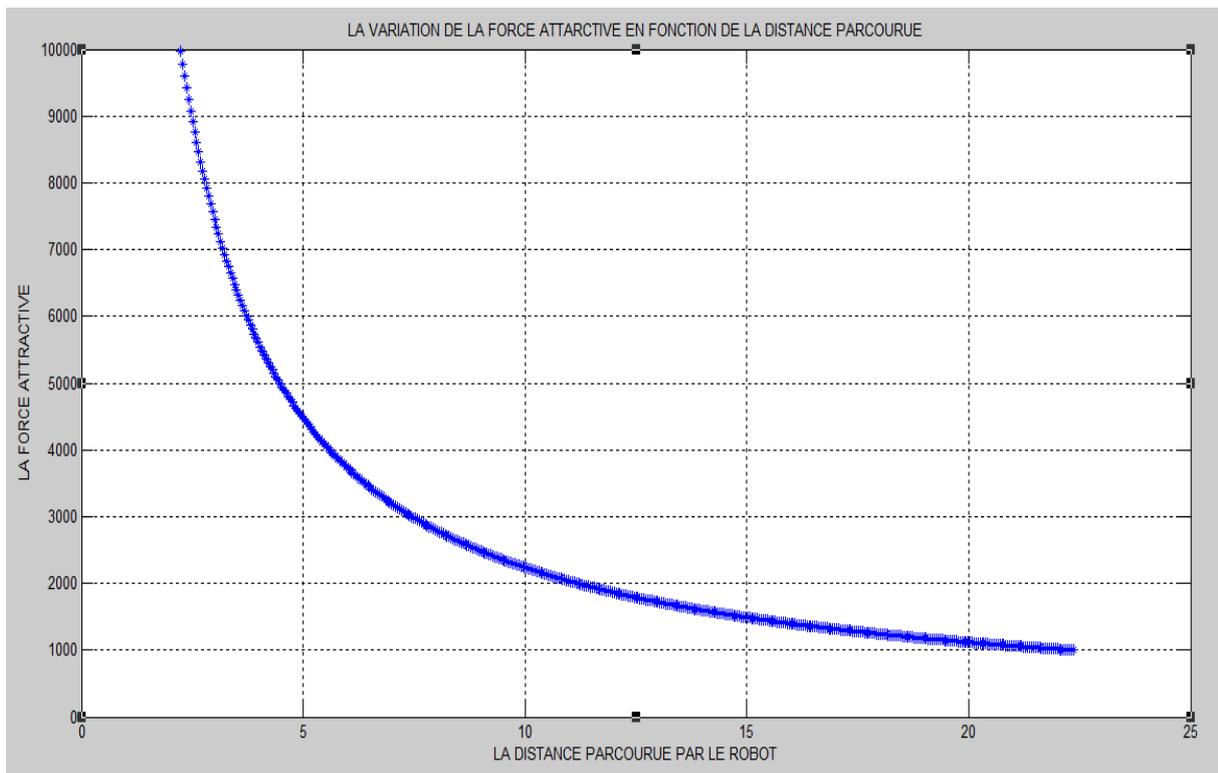


Figure 3.7

## II.9.2. Force répulsive

D'après l'équation (3.3)  $F_{repi} = \frac{F_{cst}}{d_i / d_{min}}$  la variation de la force répulsive suit

d'une fonction de type 1/x.

### II.9.2.a. Programme en MATLAB

```

n=input('donner le nombre des obstacles détecters=')% on donne le nombre
des obstacle détecter par les capteur
dmin=input('donner la distance minimal=')% on donne la valeur de la
distance minimal qui il faut respecter entre le robot et l'obstacle
F_cst=input('donner la valeur de force=')% on donne le valeur de la force
constante
for i=1 :n % boucle pour introduire les valeurs des distances mesuré par
les capteurs
    fprintf('d(%d)',i)
    d(i)=input('\n')
end
F_tot(1)=0 ;
for i=1 :n ; % boucle pour etudier la variation de la force répulsive en
fonction de la distance
    ds=linspace(dmin,d(i),100)
    for j=1 :100
        F_rep(j)=F_cst/(ds(j)/dmin) ;
        if i==1
            F_tot(j)=F_rep(j)% calcule la force répulsive total
        end
    end
end

```

```

else
    F_tot(j)=(F_rep(j)+F_tot(j))
end
if j==100
    plot(ds,F_rep,'*')% trace les graphes de variation des forces
    répulsives élémentaire en fct de la distance
    figure
end
end
end
end
dg=linspace(dmin,200,101)
plot(dg,F_tot,'*r') %trace le graphe de la force répulsive totale en
fonction de la distance
figure

```

### EXEMPLE

Nous avons obtenu les graphes suivantes pour les valeurs suivantes :  
 Le nombre des obstacles détecté par les capteurs sont 3 obstacles.  
 La distance minimale qu'il faut rester entre notre robot et l'obstacle est 50 mm.  
 La valeur de la force constante est 1000 N  
 Les trois distances détectés sont  $d_1=120$  mm  $d_2=150$  mm  $d_3=90$  mm

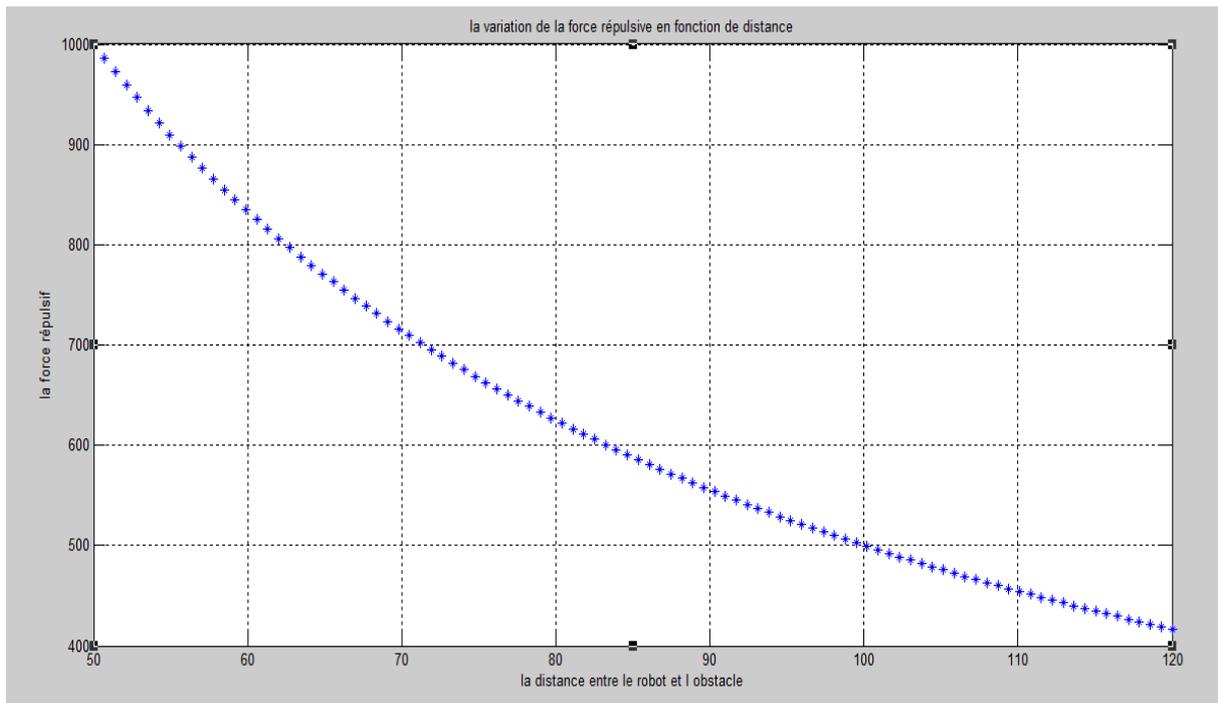


Figure 3.8

La variation de la force répulsive pour l'obstacle 1

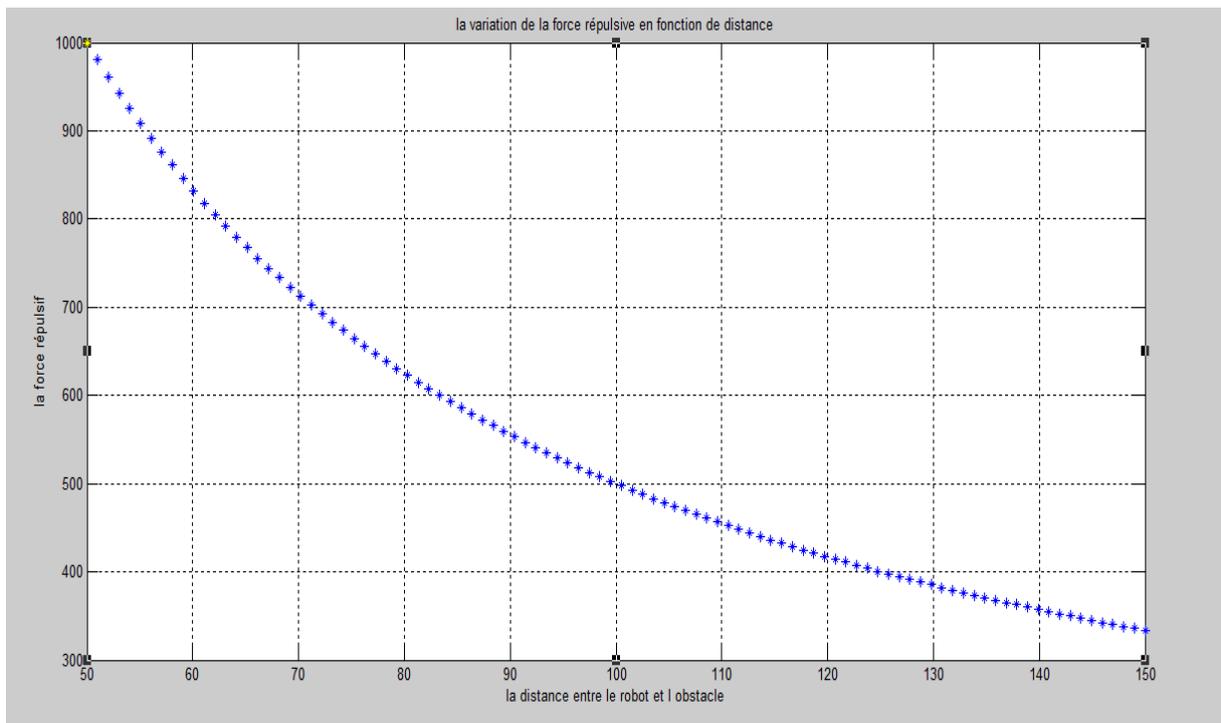


Figure 3.9 La variation de la force répulsive pour l'obstacle 2

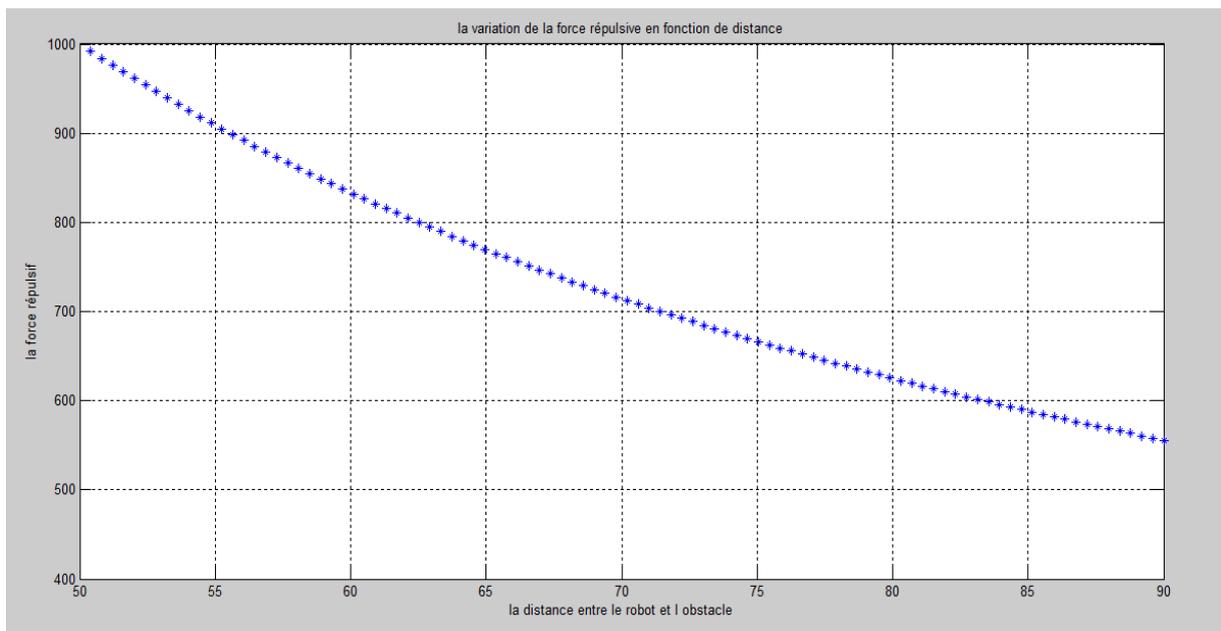


Figure 3.10

La variation de la force répulsive pour l'obstacle 3

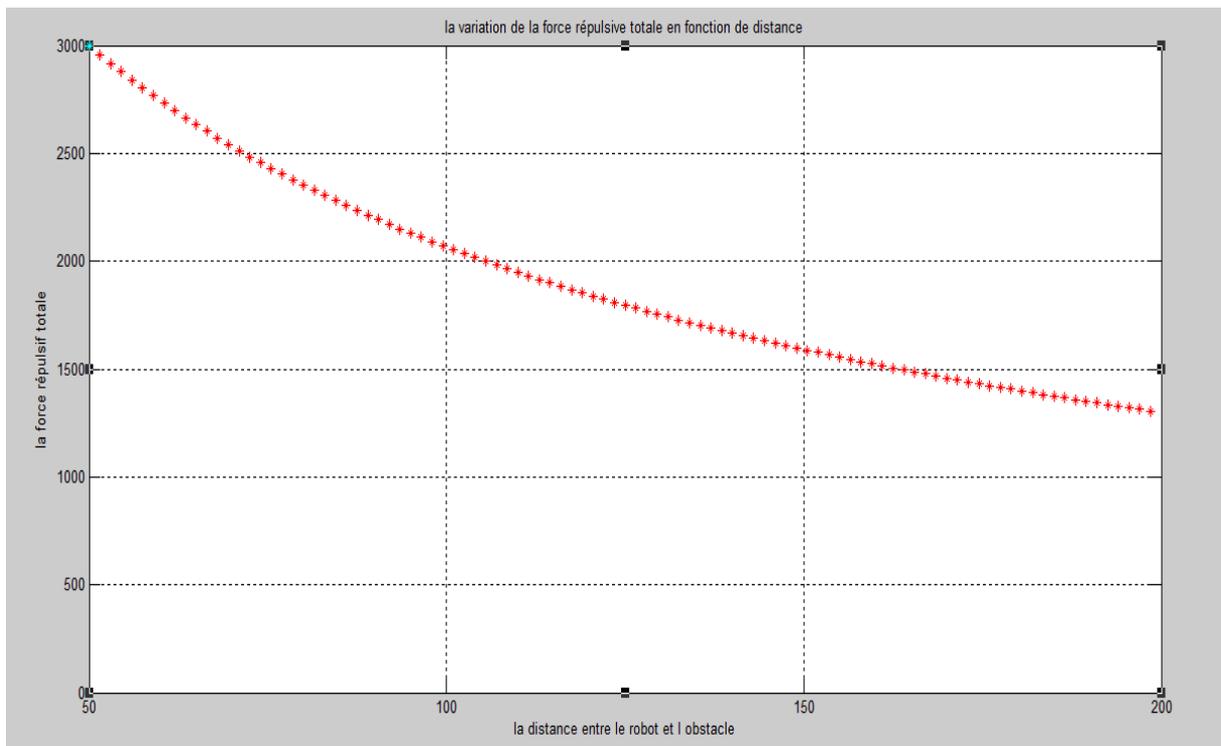


Figure 3.11

Variation de la force répulsive totale

Nous avons remarqué que la force répulsive augmente considérablement lorsque notre robot approche de l'obstacle.

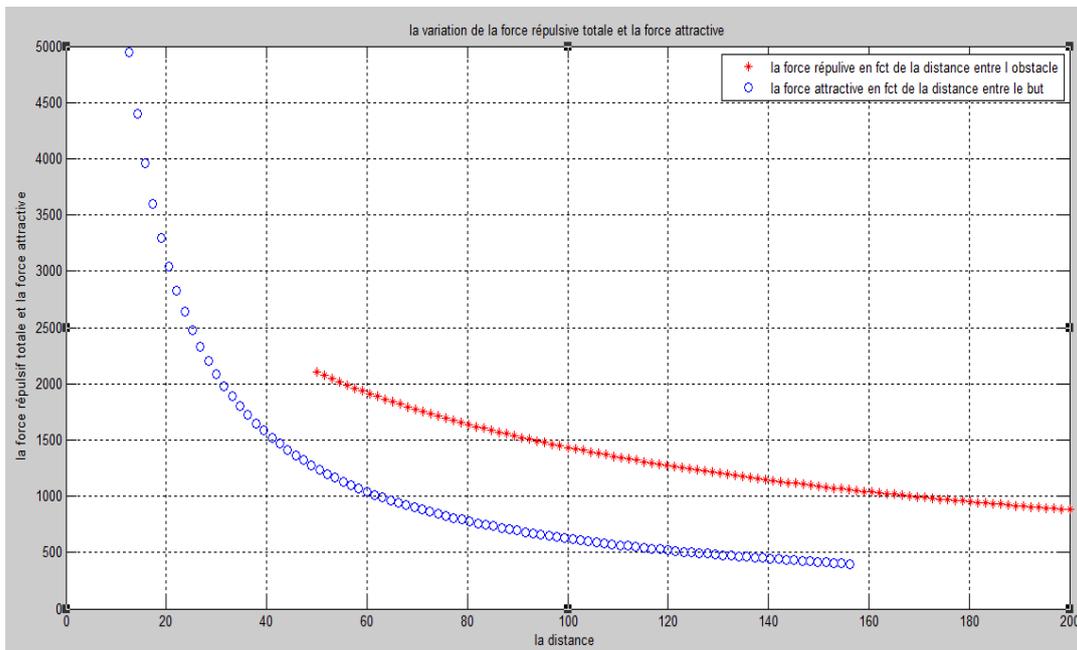


Figure 3.12

Variation de la force attractive et la force répulsive

Nous avons tracé le graphe de variation de la force attractive en fonction de la distance entre notre robot mobile et notre but, et la variation de la force répulsive en fonction de la distance entre le robot et l'obstacle.

### **III. CONCLUSION**

Nous avons étudié la navigation autonome d'un robot mobile basé sur le comportement réactif, cette méthode qui contrôle le robot par un champ des forces virtuelles attractives vers le but et répulsives quand il y'a des obstacles.

La méthode que nous avons proposée a dans certains cas des problèmes des minima locaux, quelques solutions nous avons cité pour sortir notre robot de blocage.

Dans le chapitre suivant nous essayons de réaliser notre robot muni d'une navigation autonome.

## **Chapitre 4**

# **REALISATION ET IMPLEMENTATION DE LA COMMANDE EFFECTIVE D'UN ROBOT MOBILE**

# REALISATION ET IMPLEMENTATION DE LA COMMANDE EFFECTIVE D'UN ROBOT MOBILE

## I. INTRODUCTION

Comme pour toute étude théorique, il est nécessaire de faire une réalisation ou une application en vue de valider les résultats théoriques, sinon, l'étude reste incomplète et peut-être inutile.

Pour compléter notre travail théorique, nous concevons un robot mobile unicycle simple, ce qui permettra d'enrichir nos connaissances et appliquer notre étude.

## II. CONCEPTION MECANIQUE

Les figures ci-dessous représentent la base mécanique (châssis et roues) du robot étudié, qui constitue la partie principale. Les deux roues motrices sont disposées soigneusement de façon à avoir un bon parallélisme. Une roue folle est montée à l'arrière pour stabiliser la base. Deux par-chocs sont fixés l'un à l'avant et l'autre à l'arrière pour protéger les éléments intérieurs.

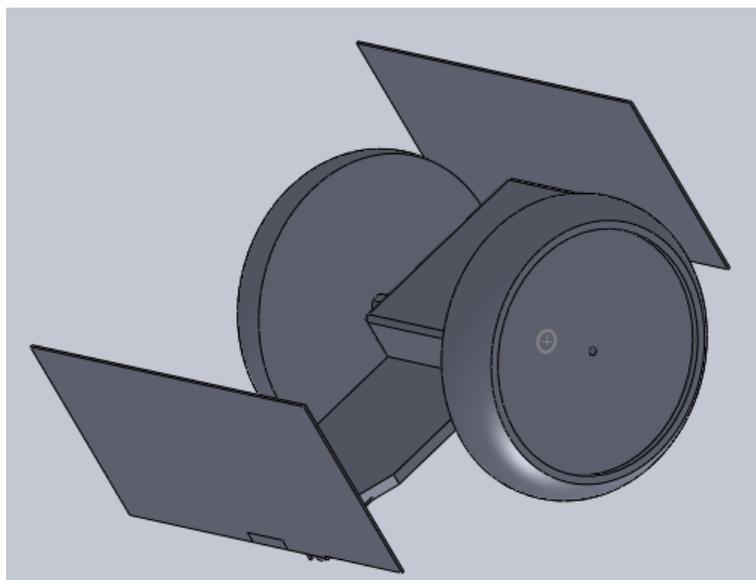


Figure 4.1 : vue isométrique de la base mécanique

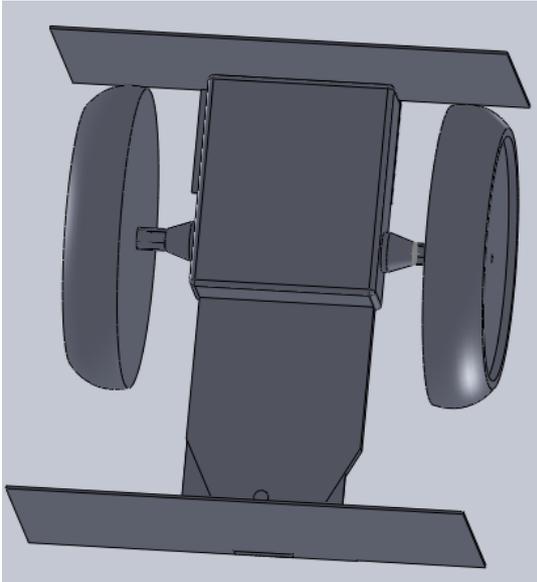


Figure 4.2 : Vue de dessus de la base mécanique

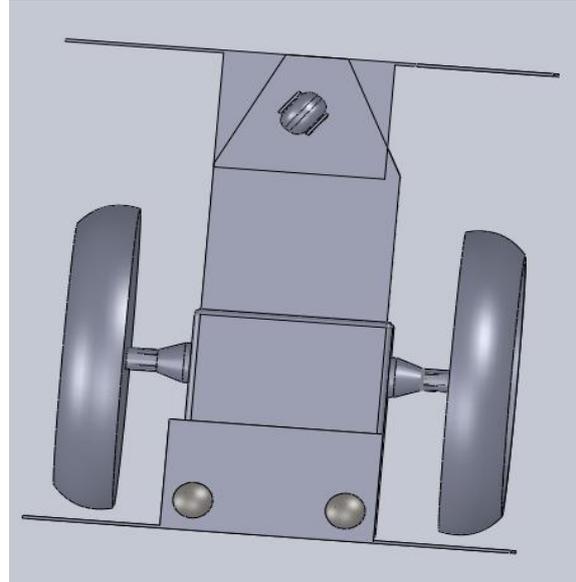


Figure 4.3 : Vue de dessous de la base mécanique

## II.1. Dimensions du robot

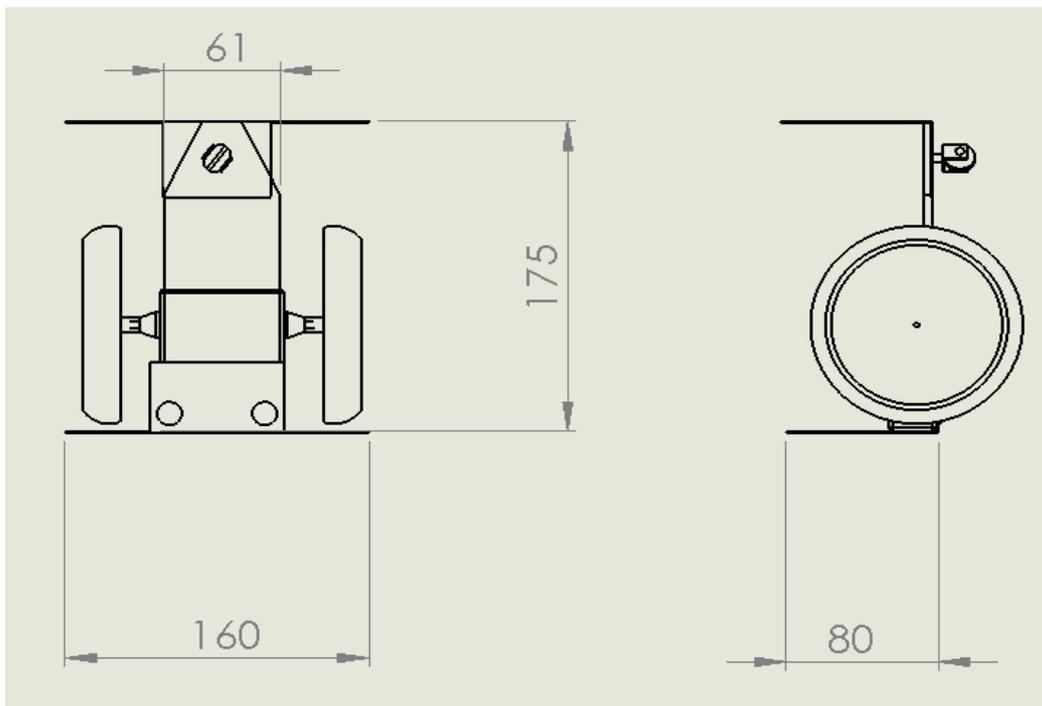


Figure 4.4 : Dimensions de robot

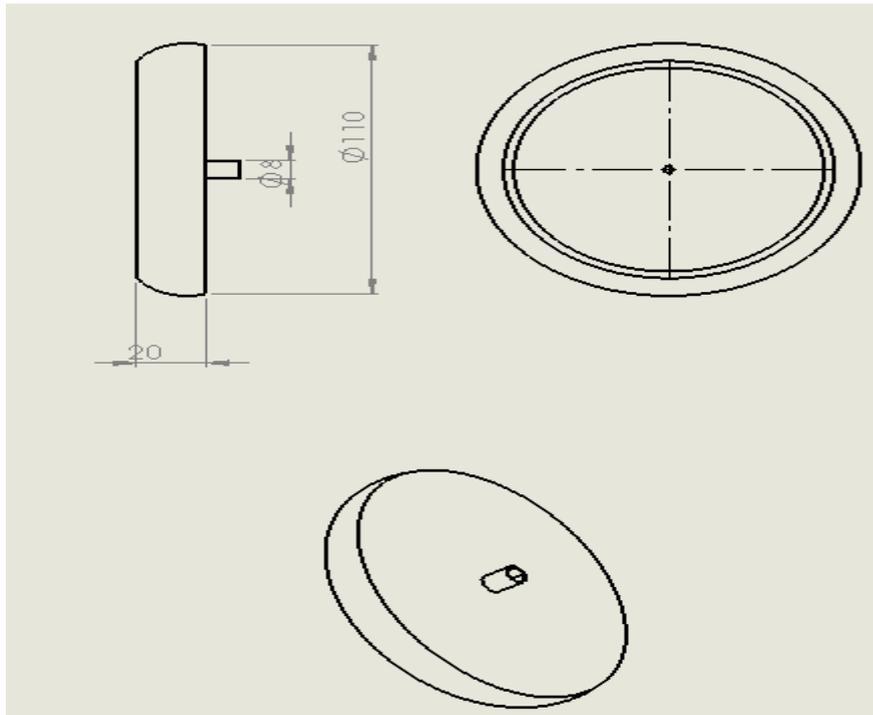


Figure 4.5 : Dimensions des roues motrices

## II.2. Dimensions des moteurs

Le choix des moteurs d'un robot est généralement une tâche fastidieuse. Il faut dans un premier temps déterminer avec suffisamment de précision le travail qui sera accompli par chacun des moteurs. Évidemment, pour pouvoir valider un choix, il faut connaître les performances à atteindre.

Dans ce qui suit, on présente la méthodologie pour dimensionner les moteurs à utiliser.

Il est essentiel de connaître les paramètres suivants ou de les fixer comme objectifs :

- 1) la masse du robot  $M_r$  [Kg]
- 2) le rayon des roues  $R_r$  [m]
- 3) la vitesse de déplacement maximale du robot  $v_{\max}$  [m/s].

Dans notre présente étude, le problème se pose inversement, c'est-à-dire que les performances à atteindre sont déterminées à partir des caractéristiques des deux moteurs.

Nous avons choisi deux moteurs à courant continue de puissance 15W chacun.

### II.2.1. Vitesse maximale

La vitesse de rotation des roues est déterminée comme suit :

$$\omega_{roue} = \frac{v_{roue}}{R_{roue}} \dots\dots\dots (4.1)$$

La vitesse maximale de translation du robot est déterminée expérimentalement en le faisant déplacer à la puissance maximale. On obtient :

$$V_{maxroue} = 0.2\text{m/s}$$

La roue étant de diamètre de 0.11 m, sa vitesse angulaire est de :

$$\omega_{maxroue} = 2 \times 0.2 / 0.11 = 3.6364 \text{ rd/s}$$

Les deux roues sont entraînées par des systèmes à engrenages identiques, on a donc  $\omega_{roue1} = \omega_{roue2} = \omega_{roue}$ , par conséquent,  $\omega_{moteur1} = \omega_{moteur2} = \omega_{moteur}$ .

La figure 4.6 permet de calculer le rapport de transmission entre une roue et son moteur d'entraînement. On a pour  $Z_1=10$  ;  $Z_2=48$  ;  $Z_3=Z_4=16$  ;  $Z_5=48$  ;  $Z_6=34$  :

$$u = \frac{\omega_{moteur}}{\omega_{roue}} = \frac{Z_2 Z_4 Z_6}{Z_1 Z_3 Z_5} = \frac{48 \times 16 \times 34}{10 \times 16 \times 48} = 3.4$$

La vitesse angulaire maximale de chacun des moteurs est donc :

$$\omega_{moteur} = u \times \omega_{roue} = 3.4 \times 3.6364 = 12.3638 \text{ rd/s}$$

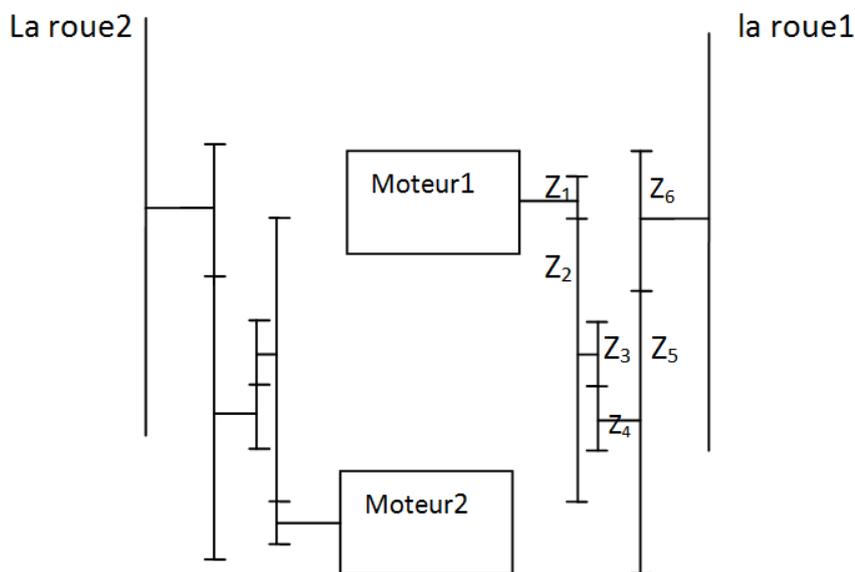


Figure 4.6 : Chaîne d'engrenage du robot

### II.2.2. Couple maximum

Puisque nous avons le cas des faibles puissances (15 W), on suppose que la puissance mécanique du moteur est égale à sa puissance électrique, donc son rendement vaut 1, ce qui permet d'écrire :

$$P_{max} = C_{max} \times \omega_{max}$$

Ce qui donne :

$$C_{max} = \frac{P_{max}}{\omega_{max}} = \frac{15}{12.3638} = 1.213 \text{ Nm}$$



Figure 4.7 : moteur électrique utilisé

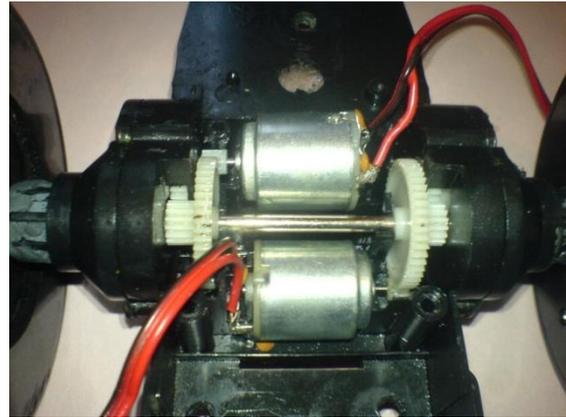


Figure 4.8 : Système de transmission

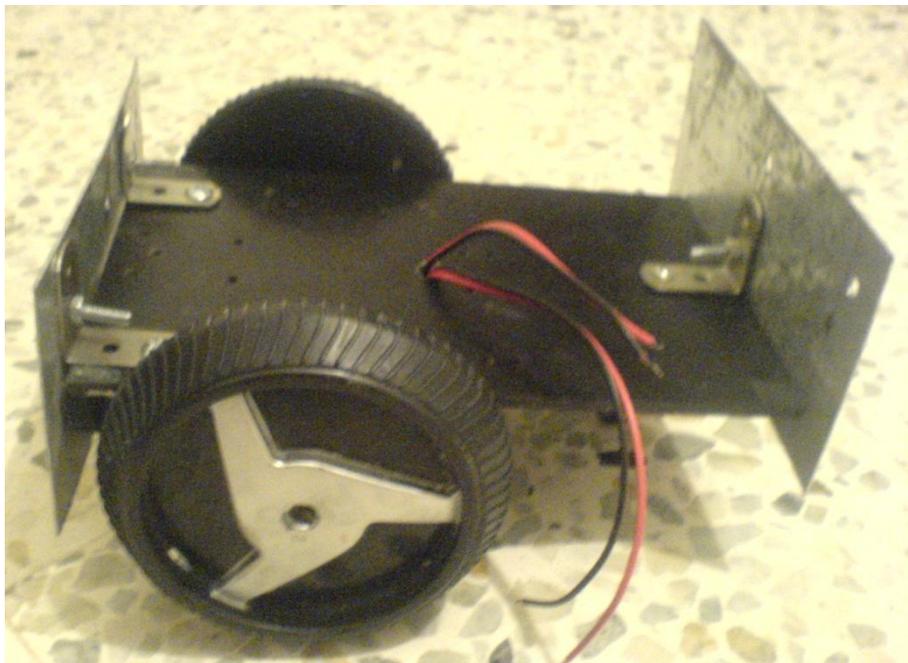


Figure 4.9 : Robot mobile unicycle

### III. CONCEPTION ELECTRONIQUE ET INFORMATIQUE :

Après avoir fait la conception mécanique et déterminé les performances du Robot mobile à partir de celles des deux moteurs, on passe maintenant à un autre niveau, c'est la conception électronique et informatique qui représente l'étape finale de la partie réalisation.

#### III.1. Interface de puissance

C'est une carte électronique servant à la connexion entre les moteurs et la carte de commande puisque celle-ci est alimentée sous une tension de 5.5V; les

deux moteurs fonctionnant sous 12 V. La figure 4.10 illustre le schéma électrique de cette interface.

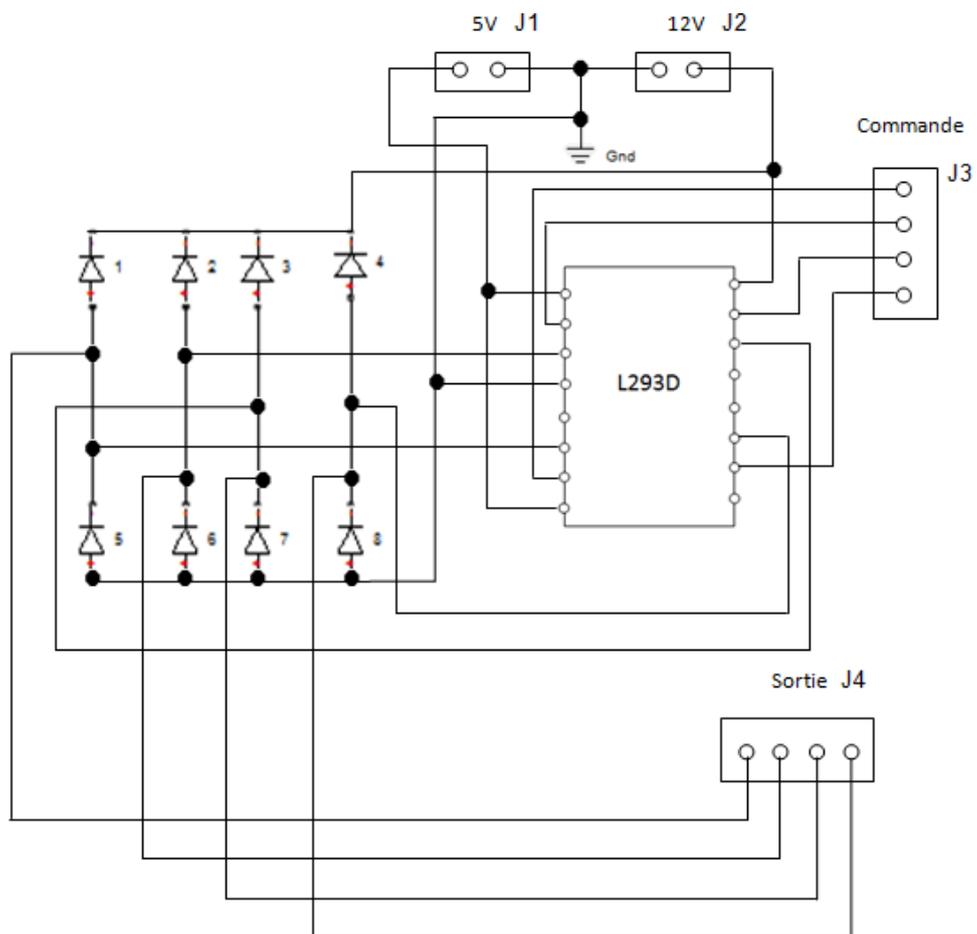


Figure 4.10 : Schéma électrique de l'interface

Légende :

L298D : pilote du moteur le plus utilisé pour les petits robots.

J1 et J2 : pour l'alimentation 5V et 12 V

J3 : pour relier avec la carte de command

J4 : pour relier avec les deux moteurs

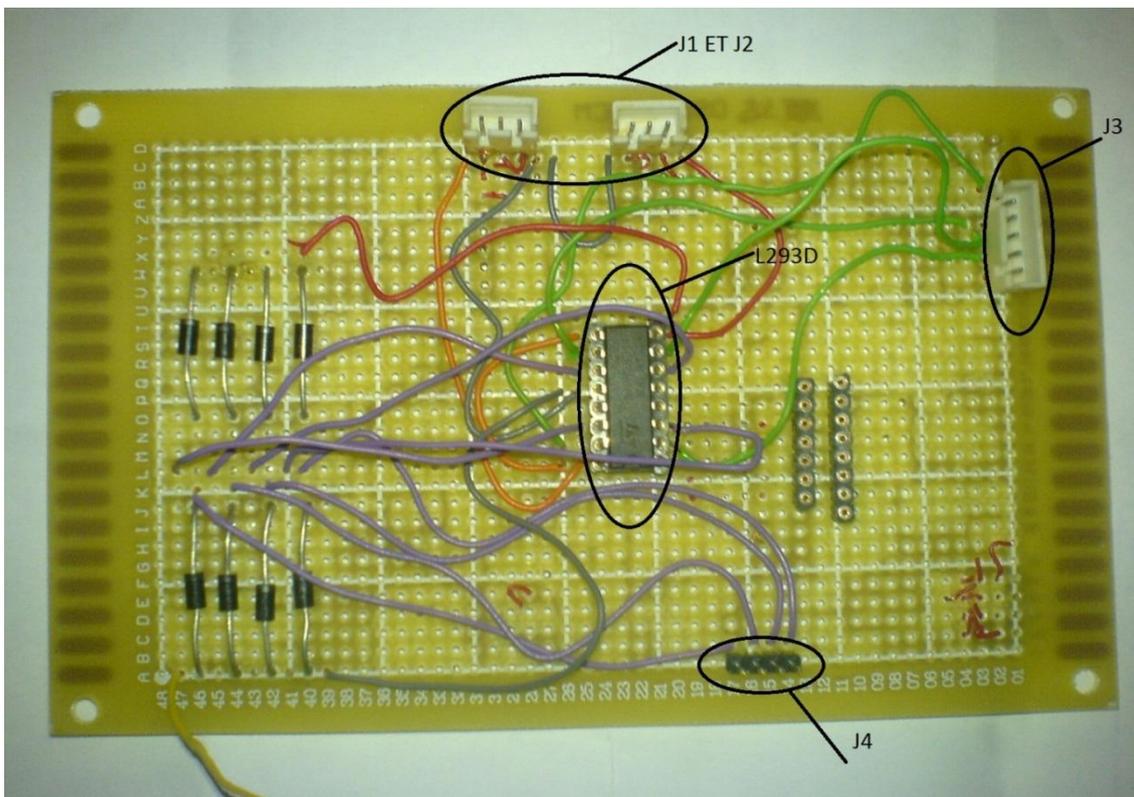


Figure 4.11 : Interface de puissance

### III.2. Carte de commande

Nous avons choisi une carte de type ARDUINO UNO pour implémenter la commande. C'est une carte qui utilise le langage ARDUINO (langage C) pour implémenter la commande.



Figure 4.12 : Carte ARDUINO UNO

### III.3. Capteurs

Nous avons choisi des capteurs infrarouges SHARP GP2 D15 pour déterminer la distance et détecter la présence d'obstacles.



Figure 4.13 : Capteur infrarouge SHARP GP2 D15

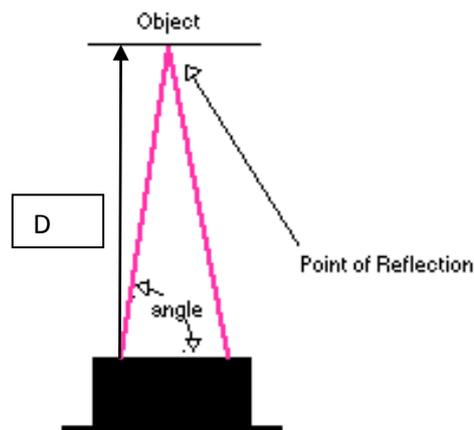


Figure 4.14 : Principe de mesure à distance

Ils fonctionnent en émettant une courte pulsation de lumière infrarouge qui est réfléchi en cas de présence d'obstacle. Si l'onde envoyée par l'émetteur est réfléchi vers le récepteur dans un laps de temps donné, le capteur la perçoit et mesure l'angle entre l'émetteur, l'obstacle et le récepteur, sinon il considère qu'il n'y a rien devant lui.

L'angle varie suivant la distance à l'obstacle. La lentille de réception focalise l'onde retour, qui illumine ainsi une cellule de la rangée, ce qui détermine la distance de l'objet par trigonométrie.

#### IV. MONTAGE FINAL

Les figures suivantes (4 .15 et 4 .16) présentent le montage final du robot mobile.

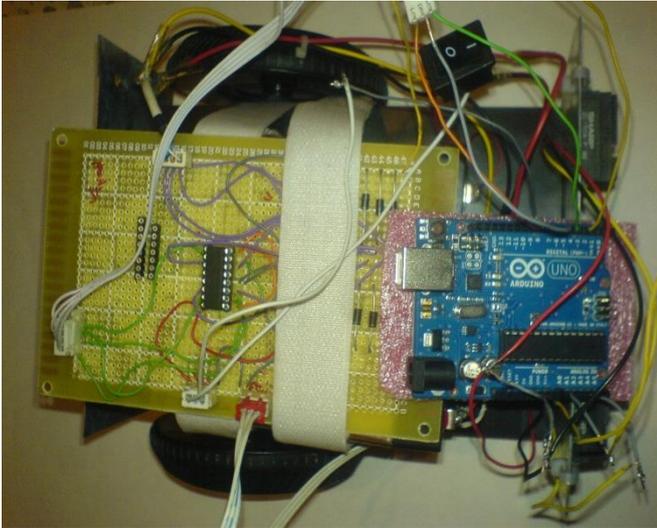


Figure 4.15

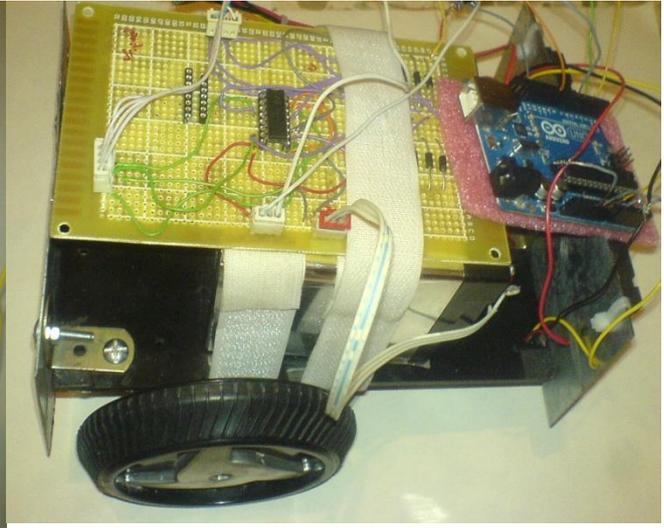


Figure 4.16

## V. PROGRAMMATION DE LA CARTE ARDUINO

La programmation est faite dans le langage C

Ce programme est fait pour la navigation autonome

```

int Left,Right;
void setup(){
pinMode(3,OUTPUT);    // Motor A1  relier a la pin numéro 3
pinMode(5,OUTPUT);    // Motor A2  relier a la pin numéro 5
pinMode(6,OUTPUT);    // Motor B2  relier a la pin numéro 6
pinMode(9,OUTPUT);    // Motor B1  relier a la pin numéro 9
}
void Forward(){       // Robo aller tout droit
analogWrite(3,70);
analogWrite(5,0);
analogWrite(6,70);
analogWrite(9,0);
}
void Backward(){     // Robot faire un marche arriere
analogWrite(3,0);
analogWrite(5,70);
analogWrite(6,0);
analogWrite(9,70);
}
void Spin(){         // robot éviter l'obstacle
analogWrite(3,0);
analogWrite(5,70);

```

```
analogWrite(6,70);
analogWrite(9,0);
}
void stope(){           // robot faire un stop
digitalWrite(3,0);
digitalWrite(5,0);
digitalWrite(6,0);
digitalWrite(9,0);
}
void loop(){
Left = analogRead(0); // lire la valeur du capteur gauche
Right = analogRead(3); // lire la valeur du capteur droite
if (Left==LOW | Right==LOW){
Forward();
delay(1600);
}
else{
Backward();
delay(500);
Spin_Left();
delay(500);
}
}
```

# **CONCLUSION ET PERSPECTIVE**

## Conclusion générale

Nous avons eu, dans notre travail, à faire une étude complète et une réalisation d'un robot mobile unicycle. Comme toute étude d'un projet en robotique, nous avons effectué une modélisation de point de vue géométrique, cinématique et dynamique d'une structure unicycle, puis, on a synthétisé des lois de commande, premièrement en utilisant la technique de discrétisation de trajectoire pour faire une poursuite de trajectoire, deuxièmement pour la navigation autonome nous avons choisi un système de navigation réactive basé sur un champ de force virtuel l'un est attractive et l'autre est répulsive.

Notre étude théorique est concrétisée par une conception d'un robot mobile sur trois plans :

- ✓ **Mécanique** : dimensionnement de la base mécanique, d'un système de fixation des moteurs et des deux roues motrices du robot. Ceci est effectué à l'aide du logiciel de conception mécanique ' *Solide Works*'. La base mécanique est conçue de façon à avoir une symétrie autant que possible autour de l'axe des deux moteurs dans l'objectif d'avoir une répartition homogène de la charge.
- ✓ **Electronique** : conception d'une carte électronique qui répond aux exigences de l'application. Une interface de puissance est réalisée pour assuré la connexion entre la carte de commande et les deux moteurs, une carte ARDUINO UNO pour l'implémentation du command et traduire les impulsions des deux capteurs en command effective pour la navigation autonome.
- ✓ **Informatique** : élaboration d'un programme en C qui assure l'asservissement des vitesses des deux moteurs et essentiellement la navigation autonome de robot mobile.

## Perspective :

**'...la pluie commence par une goutte d'eau...'**

Ce Projet de Fin d'Etudes est un travail de base en robotique mobile qui ouvre d'autres horizons de recherches dont on peut citer :

- Concevoir une autre architecture mécanique qui permettra au robot d'évoluer dans des terrains accidentés ou pour des opérations militaires délicates telles que le déminage ou l'accompagnement de troupes. (ajout d'une suspension, développer une mobilité hybride, etc).
- Remplacer les moteurs utilisés par d'autres qui permettront l'amélioration des performances en vitesse du robot.
- Ajouter un bras manipulateur sur la base mobile pour l'exploration ou l'intervention dans des milieux hostiles pour l'homme, en général, (zone radioactive, présence de feu, endroit toxique ...)
- Utiliser la communication sans fil émetteur/récepteur infrarouge ou bien radio.
- Expérimenter d'autres techniques de localisation comme, le système GPS (Global Positioning System) ou bien en utilisant des retours visuels.

## **BIBLIOGRAPHIE**

**[1]** Robotique mobile, David FILLIAT, Ecole Nationale Supérieure de Technique Avancées Paris Tech, 2011

**[2]** Robotique mobile, Bernard BAYLE, Ecole Nationale Supérieure de Physique de Strasbourg Université de Strasbourg, 2007

**[3]** Locomotion d'un robot mobile, Stéphane Lens, 2008

**[4]** Recherche en Robotique pour le handicap, Fondation caisses d'épargne pour la solidarité (le projet ANSO), 2007

**[5]** Trajectory tracking through predictive control: application to the Khepera robot, Juan Herrera, Diploma thesis, 1999

**[6]** Localisation et Navigation en Robotique Mobile ENSTA, David FILLIAT, 2010

**[7]** pilotage réactif d'un robot mobile étude d'un lien entre la perception et l'action, Patrick REIGNIER, thèse soutenue le 13 décembre 1994 l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

### **Sites internet**

**[1\_int]** <http://www.ladepeche.fr/article/2010/12/09/966476-le-robot-toulousain-va-explorer-mars.html>

**[2\_int]** <http://www.robotshop.com/blog-fr/medical>

**[3\_int]** <http://www.vincentabry.com/une-voiture-robot-developpee-en-france-14962>

**[4\_int]** <http://www.robotshop.com/ca/plateforme-mobile-version-pro-wifi-x80svp-dr-robot.html>

---

<sup>i</sup> Localisation et Navigation en Robotique Mobile ENSTA, David FILLIAT, 2010

<sup>ii</sup> Localisation et Navigation en Robotique Mobile ENSTA, David FILLIAT, 2010

<sup>iii</sup> Localisation et Navigation en Robotique Mobile ENSTA, David FILLIAT, 2010

<sup>iv</sup> pilotage réactif d'un robot mobile étude d'un lien entre la perception et l'action, Patrick REIGNIER, thèse soutenue le 13 décembre 1994 l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

<sup>v</sup> pilotage réactif d'un robot mobile étude d'un lien entre la perception et l'action, Patrick REIGNIER, thèse soutenue le 13 décembre 1994 l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE