



République Algérienne Démocratique et Populaire

Ministère De l'Enseignement Supérieur et de la Recherche Scientifique

Université Abbès Laghrour Khenchela

Faculté des Sciences et de la Technologies

Département de Génie Industriel

Mémoire Présenté en vue de l'obtention du diplôme de

Master Académique

Filière : Automatique

Spécialité : Automatique et informatique industrielle

Thème :

**Suivi de trajectoires d'un robot mobile :  
Approche basée sur la méthode de  
linéarisation du retour d'état (*State  
Feedback Linearization*)**

Présenté par :

- Moussaoui Nour Elaislam

Encadré par :

Mr. Allouani Fouad

2019/2020

# Dédicaces

*Je tiens à dédier ce mémoire :*

*A ma très chère Mère et à mon cher Père, en témoignage et en gratitude de leurs dévouements, de leur soutien permanent durant toutes mes années d'études, leurs sacrifices illimités, leurs réconforts moraux, eux qui ont consenti tant d'effort pour mon éducation, mon instruction et pour me voir atteindre ce but, pour tout cela et pour ce qui ne peut être dit, mes affectations sans limite.*

*A mes Chère Sœurs Leila et Zohra,*

*A ma chère fiancée Nadia,*

*A mon cher Frère Aymen,*

*A toute ma famille.*

*A mes chers Amis : Olid , Zinou , Yassin, Athmane, Khaled , Maninou , ZAKA ,Souli, Mouha ,Imad ,Babay,Fateh,Oussama,Lotfi , Bachir ,Amine ,Mostafa ,Alouani, Youssef ,Billel, Achour, Chihab ,Mouhamed, el3id, 3adel,Ridha.*

# Remerciements

Je remercie en premier **ALLAH** le tout puissant de ma avoir accordé la volonté et le courage pour réaliser ce mémoire.

Je tiens à remercier Mr. Allouani Fouad Maitre de conférences à l'université Abbas Laghrour Khenchela, et lui exprimer toute ma reconnaissance et ma profonde gratitude pour avoir encadré et dirigé ce travail. Je le remercie vivement, de ses conseils, ses orientations et le soutien que j'ai trouvé auprès de lui, et qu'il n'a jamais cessé de me témoigner durant la période de la réalisation de ce travail, ont permis d'achever ce travail.

J'exprime mes sincères remerciements aux membres de lecture de ce Mémoire.

J'associe à ces remerciements, tous les enseignants du département de génie industriel pour leur conseils et soutien tout au long de la période d'étude.

Je remercie ma sœur Laila de m'avoir aidé dans ce travail.

Je profite de cette occasion qui m'est offerte, pour remercier tous mes amis et collègues du département de génie industriel.

Enfin, les derniers mots sont pour remercier toute ma famille et mes amis pour leur aide et leurs encouragements constants.

### ملخص

يتعلق العمل المقدم في هذه المخطوطة بشكل أساسي بتشكيل وتطبيق قوانين التحكم ، استنادًا إلى النهج الذي يستند على البعد الخطي من خلال العودة للحالة (*state Feedback Linearization*) وبشكل أكثر دقة، طريقة التحكم المسماة بالإنكليزية '*Input-Output State Feedback linearization (I-O SFL)*' ، للتتبع مسارات الروبوتات المتنقلة غير الشاملة. في هذه الأطروحة، تم تخصيص اهتمام خاص لنوع الروبوتات المتنقلة ذات الدرجة الأحادية. بالإضافة إلى ذلك، تستند النماذج المدروسة بتشكيل قوانين التحكم على النموذج الحركي للروبوت المتحرك. يتم اختبار طريقة التحكم المقدمة من خلال محاكاة بواسطة برنامج MATLAB. تظهر جميع النتائج التي تم الحصول عليها فعالية طريقة التحكم المستعملة لمعالجة المشكلة المطروحة.

### Résumé :

Le travail présenté dans ce mémoire de Master concerne principalement, la synthèse et l'application de lois de commande basée sur l'aspect de la linéarisation de retour d'état (*state Feedback Linearization*) et plus précisément la méthode dite en anglais '*Input-Output State Feedback linearization (I-O SFL)*', pour le suivi de trajectoires des robots mobiles non holonomes. Dans ce mémoire, un intérêt particulier est réservé aux robots mobiles de type unicycle. En outre, les modèles considérés pour la synthèse des lois de commandes sont basés sur le modèle cinématique du robot mobile de type unicycle.

La méthode de commande présentée, est testée à travers une simulation par logiciel MATLAB. L'ensemble des résultats obtenus, montre l'efficacité de cette approche pour le problème traité.

### Abstract:

The work presented in this manuscript, is mainly concerns the synthesis and application of control laws, based on the aspect of *state Feedback Linearization* technique and more precisely the *Input-Output State Feedback linearization (IO-SFL)* control method, for tracking the trajectories of non-holonomic mobile robots. In this thesis, particular interest is reserved for unicycle mobile robots' type. In addition, the models considered for the synthesis of the control laws are based on the kinematic model of the mobile robot.

The presented control method, is tested through a simulation by MATLAB software. All obtained results show the effectiveness of this approach for the treated problem.

# Table des matières

Introduction générale P. 1-2

Chapitre I : Généralité sur la robotique mobile P. 3-22

- I.1. Introduction P. 3
- I.2. Concepts sur la robotique mobile P. 3
- I.3. Les nouvelles approches de la robotique mobile P. 10
- I.4. L'autonomie d'un robot P. 12
- I.5. Environnement dynamique et incertain P. 15
- I.6. Problématiques de la robotique mobile P. 15
- I.7. Architectures de contrôle P. 17
- I.8. Conclusion P. 22

Chapitre II : La cinématique du robot mobile P. 23-45

- II.1. Introduction P. 23
- II.2. Définitions P. 23
- II.3. Roulement sans glissement et contraintes non holonomes P. 24
- II.4. Les grandes classes de robots mobiles et leurs modèles P. 28
- II.5. Propriétés du modèle cinématique d'un robot mobile P. 44
- II.6. Conclusion P. 45

Chapitre III : Problème de suivi de trajectoire P. 46-57

- III.1. Introduction P. 46
- III.2. Problème de suivi de trajectoire P. 46
- III.3. Conclusion P. 57

Chapitre IV : Suivi de trajectoire pour un robot mobile en utilisant la technique I-O SFL (Synthèse et simulations) P. 58-71

- IV.1. Introduction P. 58
- IV.2. Technique de linéarisation de retour d'état (*State Feedback Linearization*) P. 58
- IV.3. Suivi de trajectoire pour un robot mobile en utilisant la technique I-O SFL P. 66
- IV.4. Simulation de la méthode I-O SFL P. 68
- IV.5. Conclusion P. 71

Conclusion générale P. 72

Annexe P. 73-77

Références Bibliographiques P. 78

# Introduction générale

La commande des robots mobiles a fait l'objet ces dernières années, de nombreuses recherches de qualités, en raison de leur utilisation de plus en plus fréquente en milieu dangereux ou inaccessible où l'être humain peut difficilement intervenir. A titre d'exemples, nous pouvons citer l'intervention sur des sites nucléaires dangereux, l'exploration spatiale, les sites des attentats ou comme récemment l'avertissement et le control des grand publiques devant la propagation de la pandémie de COVID-19.

Généralement, les modèles utilisés dans ce contexte sont non linéaires et la propriété de non holonomie qui caractérise un certain nombre de ces robots mobiles conduit à l'utilisation de techniques de commande non linéaire pour les piloter. En effet, la problématique générale se réduit dans la plupart des cas étudiés à faire se déplacer le robot dans un environnement connu ou inconnu, tout en évitant d'éventuels obstacles fixes ou mobiles, pour réaliser une tâche prescrite. Il en découle qu'il faut pouvoir définir une stratégie de mouvement (planification), puis exécuter le déplacement prescrit.

L'objectif de ce mémoire de Master trouve son intérêt dans la synthèse et l'application de lois de commande basée sur l'aspect de la linéarisation de retour d'état (*state Feedback Linearization*) [2] et plus précisément la méthode dite en anglais '*Input-Output State Feedback linearization (I-O SFL)*' [27], pour le suivi de trajectoire des robots mobiles non holonomes. Nous nous intéressons dans notre étude seulement aux robots mobiles de type unicycle. Il est à noter aussi que, les modèles considérés pour synthétiser les lois de commandes de ce mémoire sont basés sur le modèle cinématique du robot mobile de type unicycle.

Le premier chapitre de ce mémoire présente un aperçu général des concepts de base de la robotique mobile.

Le second chapitre parle ou profond sur la cinématique de quelques types de robots mobiles à savoir : les robots de type unicycle, les robots de type tricycle, et les robots de type voiture.

Le troisième chapitre, présente le problème de suivi de trajectoire sans exceptions environnementaux. Plus précisément, il parle sur ce problème aux deux sens généraux (i)

contrôle d'orientation et (ii) stabilisation de mouvement avec leurs inverses (contrôle/sans contrôle et stabilisation/sans stabilisation).

Dans le dernier chapitre de ce mémoire, nous parlerons au sens général sur l'approche de commande basée sur l'aspect de linéarisation de retour d'état (*State Feedback Linearization*) ainsi que celle dite en anglais '*Input-Output State Feedback linearization (I-O SFL)*'. Ce dernier (chapitre) porte aussi un exemple illustratif et une validation par simulation relative à la technique I-O SFL.

Enfin, ce mémoire est achevé par une conclusion générale.

# Chapitre I : Généralité sur la robotique mobile

## I.1. Introduction

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

## I.2. Concepts sur la robotique mobile

La mobilité autonome des robots est devenue un sujet de recherche développé par tous les pays industrialisés. Qu'il s'agisse de robots mobiles à pattes, à roues ou même sous-marins et aériens, les applications sont vastes et multiples : robots de services, surveillance, construction, nettoyage, manipulation de charges, automobile intelligente, robots d'intervention, robots d'exploration planétaire ou de fonds marins, satellites, robots militaires, etc. Le marché potentiel de la robotique est considérable, même s'il faut pour cela résoudre des problèmes plus importants et plus fondamentaux que prévus initialement dans la quête vers la machine intelligente [1,2].

### I.2.1. Définition d'un robot mobile

Un robot est une machine chargée d'effectuer une ou plusieurs tâches. Le mot vient d'une pièce de théâtre tchèque dans laquelle apparaissait un travailleur artificiel employé pour le « servage », désigné en tchèque par « *robota* ». L'appellation *Robot mobile* regroupe tous les types de robots qui ont la capacité de déplacement qui est la caractéristique commune entre eux, la différence réside dans la manière, qui dépend du domaine d'utilisation de robot, par laquelle le robot va atteindre cette faculté de mouvement. La mobilité par les roues est la structure mécanique la plus communément appliquée. Cette technique assure selon l'agencement et les dimensions des roues un déplacement dans toutes les directions avec une accélération et une vitesse importante.

Afin d'être autonome, un robot doit être équipé de capacités de perception, de décision et d'action qui lui permettent d'agir de manière autonome dans son environnement en fonction de la perception qu'il en a et de ses objectifs [1], mais également de savoir comment réagir en conséquence, suivant le niveau d'autonomie. C'est à lui de planifier son parcours et de déterminer avec quels mouvements il va atteindre son objectif. Les recherches dans ce domaine portent principalement d'une part sur la localisation du véhicule autonome et la cartographie de son environnement, d'autre part sur le contrôle et la navigation autonome de tels véhicules (structure de contrôle, stratégies de commande, planification et navigation).

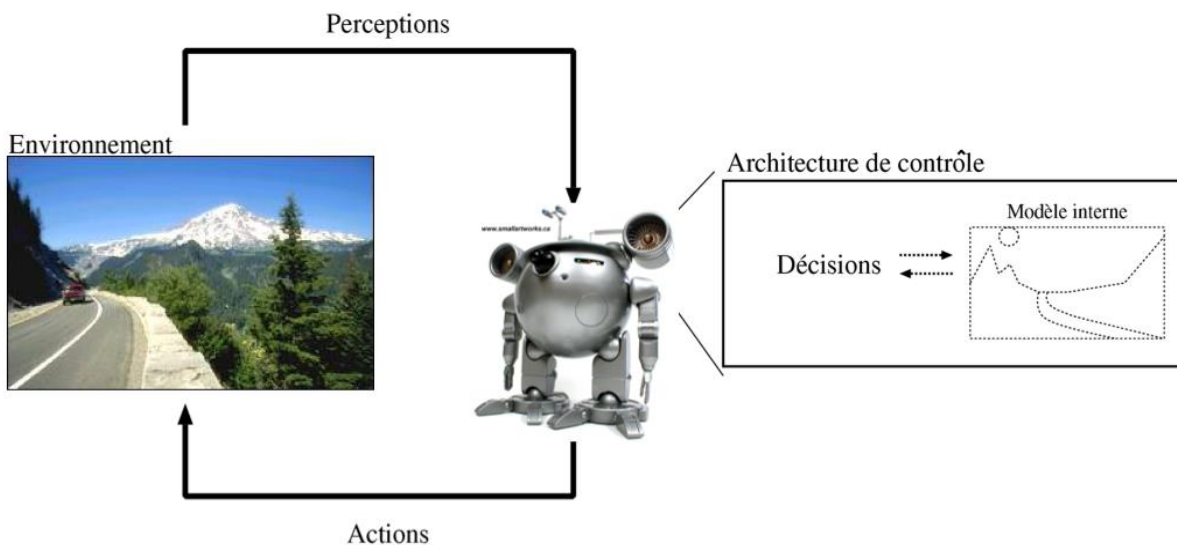
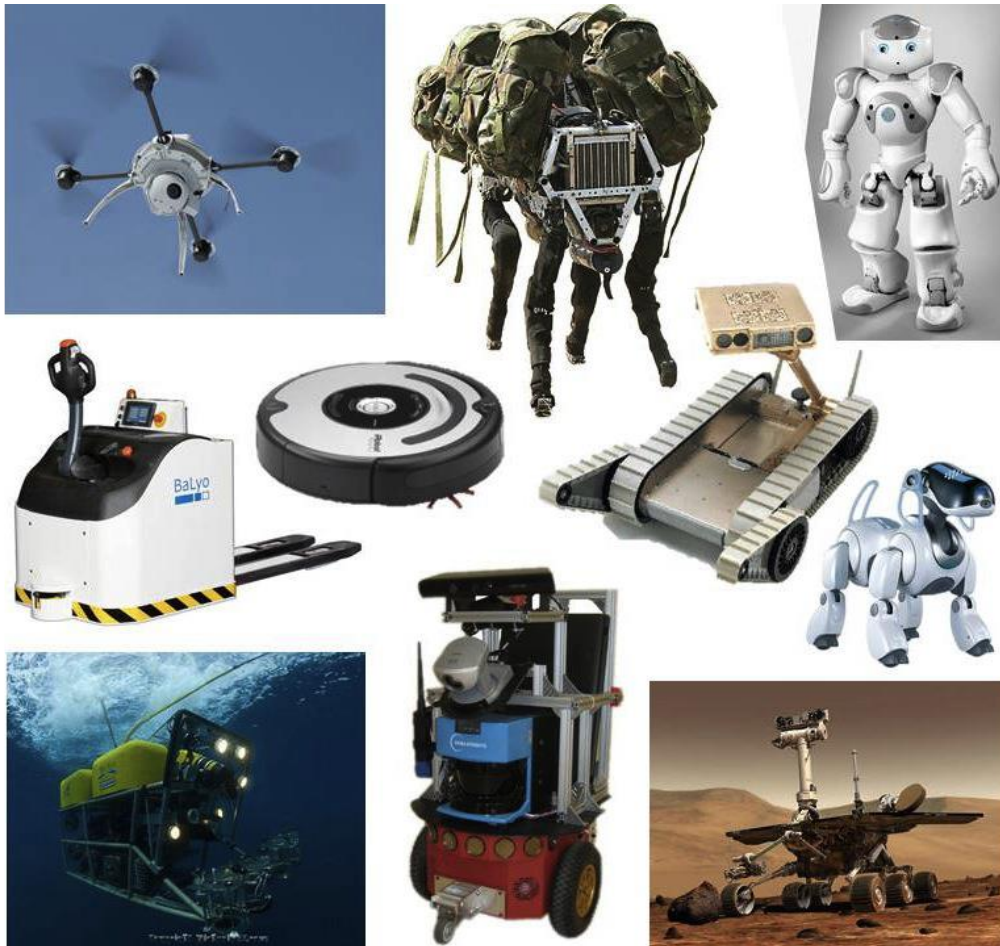


Figure I.1. Les interactions d'un robot avec son environnement.

**I.2.2. Pourquoi la robotique mobile ?**

D'après tout ce que nous voyons aujourd'hui, il est clair que dans un avenir proche, la robotique occupera une place importante dans notre quotidien. Elle possède de nombreux champs d'applications comme la robotique industrielle ou la robotique de service. Qu'il s'agisse de robot civil ou militaire, il existe désormais des robots capables d'étonnantes prouesses dans de nombreux secteurs : robots-compagnons assistant les personnes à domicile ou en charge de la surveillance et des soins, robots assurant la logistique dans les hôpitaux, robots assistant les industriels dans la réalisation de gestes pénibles et répétitifs, ou encore permettant le développement de prothèses ou d'orthèses intelligentes (Figure I.2).



**Figure I.2.** Différents types de robots mobiles selon le champ d'application.

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

### I.2.3. Architecture des robots mobiles

L'architecture d'un robot mobile s'articule autour de trois modules fondamentaux (Figure I.3) [1,2] :

- Module locomotion,
- Module perception,
- Module décision.

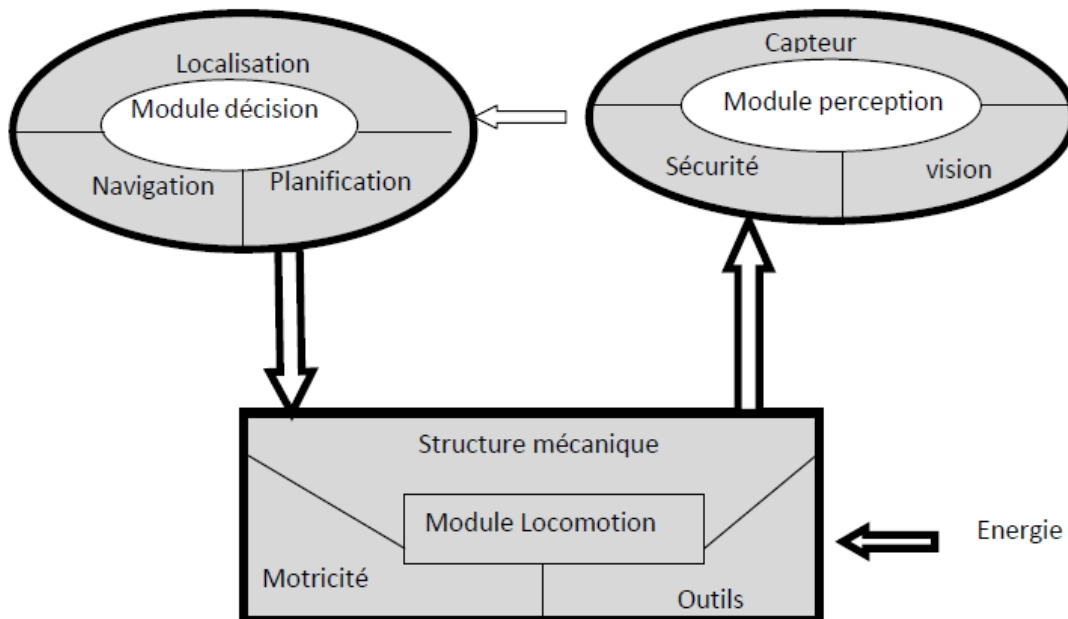


Figure I.3. Structure d'un robot mobile.

#### - Module locomotion

Le module *locomotion* comporte alors la structure mécanique, motricité et l'énergie utilisée dans le déplacement du robot mobile. Les robots mobiles sont en effet le plus souvent désignés

par leur type de locomotion, qu'ils soient à roues, marcheurs, sous-marins ou aériens. Actuellement plusieurs laboratoires de recherche contribuent jusqu'à présent à la construction des différents robots mobiles selon le type de locomotion.

### **- Module perception**

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

### **- Module Décision**

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

### **I.2.4. Types de plates formes mobiles**

Les robots mobiles à roues peuvent être classés selon le type de la plateforme mobile utilisée dans les sous-classes suivantes :

#### **I.2.3.1. Plates forme unicycle**

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

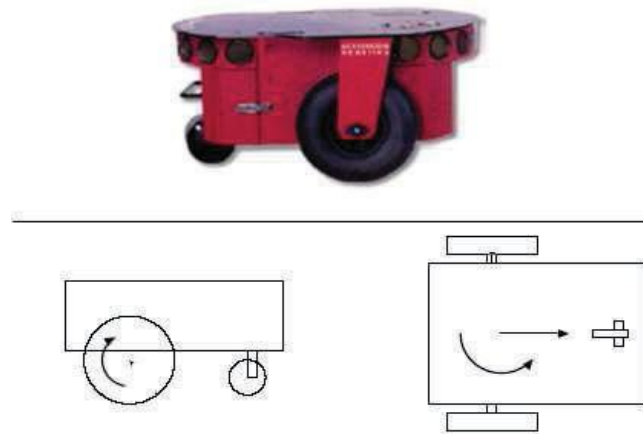


Figure I.4. Exemple de plateforme unicycle [4].

### I.2.3.2. Plates formes holonomes

Les systèmes dits *holonome* sont beaucoup plus rares dans notre vie quotidienne. Ils ont une structure mécanique complexe qui leur permet de se déplacer dans toutes les directions sans manœuvre. La société *Nomadic* disparue en l'an 2000 ; a conçu un système mobile holonome : le XR4000 (Figure I.5). Il dispose de 4 roues motrices et directrices montées comme des roues de chariot. La synchronisation des 8 axes (2 par roue, rotation et orientation) est assurée par une carte dédiée basée sur le microcontrôleur Motorola 68332 et des circuits FPGAs et la structure mécanique est composée d'engrenages coniques [1,2].

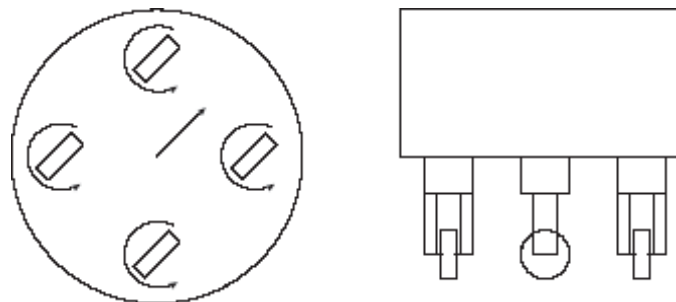


Figure I.5. Exemple de plate-forme holonome [4].

### I.2.3.3. Plates formes non-holonomes

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

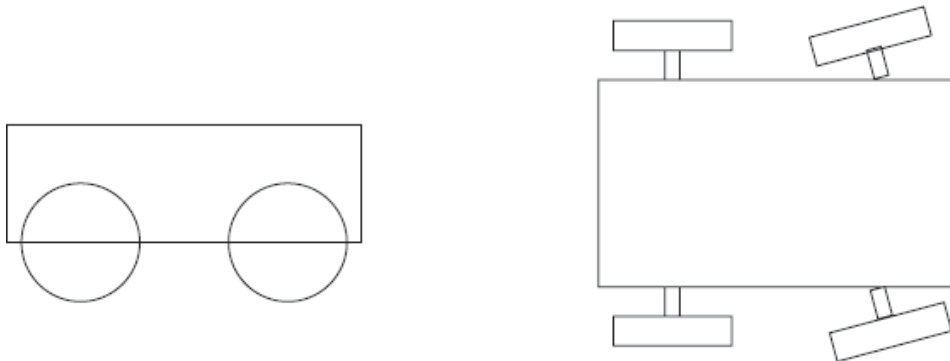


Figure I.6. Exemple de plateforme non holonome [4].

### I.2.3.4. Robot omnidirectionnel

Un robot *omnidirectionnel* est un robot qui peut se déplacer librement dans toutes les directions. Il est en général constitué de trois roues décentrées orientables placées en triangle équilatéral. L'énorme avantage du robot omnidirectionnel est qu'il est holonome puisqu'il peut se déplacer dans toutes les directions. Mais ceci se fait au dépend d'une complexité mécanique bien plus grande.

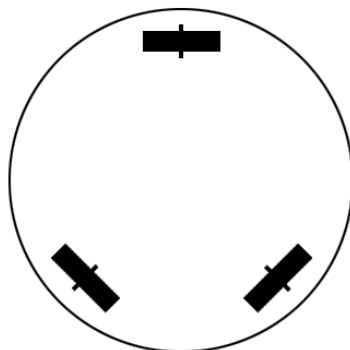


Figure I.7. Robot de type omnidirectionnel [4]

### I.3. Les nouvelles approches de la robotique mobile

En 1983, Hans Moravec [5] développe un robot mobile minimaliste, le *Stanford Cart*, équipé d'une tourelle de vision stéréoscopique. Ce robot, bien que très lent, était capable de mettre à jour son modèle du monde et d'y inclure de nouveaux obstacles, sans pour autant recourir à une représentation symbolique. Son système de vision servait à localiser des espaces libres d'obstacles et aucune reconnaissance d'objets en tant que telle n'était réalisée [6].

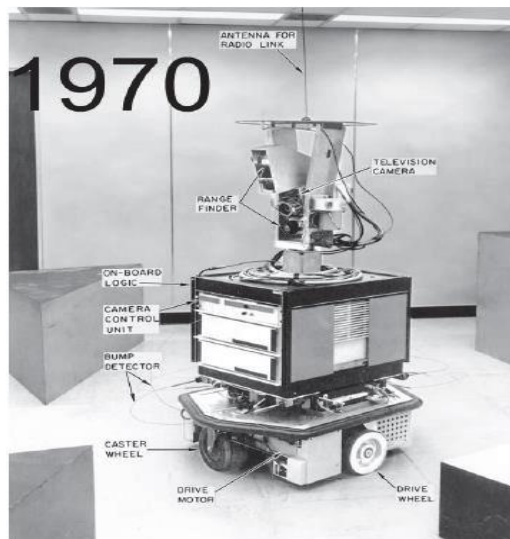


Figure I.8. Le robot Shakey de Stanford en [6].

C'est à cette période que certains chercheurs commencèrent à se douter que l'approche classique, plutôt que de résoudre et de simplifier les problèmes, les compliquait inutilement en concentrant les efforts sur le développement de systèmes de traitement symbolique de l'information. Le problème le plus critique était certainement celui du fondement des symboles, en anglais "*symbol grounding problem*", qui se réfère à la difficulté de mettre en relation la signification d'un symbole avec l'objet réel ou l'événement qu'il représente. La description d'un symbole au moyen d'autres symboles conduit très souvent à des structures circulaires ou récursives. L'introduction de nouveaux symboles pour définir un symbole existant conduit à la prolifération de ceux-ci, sans pour autant leur attribuer une signification réelle. Les humains sont capables de fonder les symboles sur leurs perceptions sensorielles (c'est le

fondement des symboles) : la signification des symboles émerge de l'interaction avec les objets de l'environnement, elle ne leur est pas inhérente.

Un autre problème lié aux approches classiques est dû à la faible robustesse des systèmes de contrôle à l'égard du bruit et des événements inattendus. L'interaction d'un robot réel avec son environnement comporte un risque élevé de voir celui-ci se perdre s'il est piloté par un système classique. En effet, il lui est très difficile de tenir compte des écarts qui apparaissent entre sa perception du monde et la représentation symbolique qu'il en a, et de les corriger. Ainsi, un système de navigation par amers reposant sur une représentation symbolique se perdra ou ne saura pas réagir de manière appropriée lorsque ses capteurs lui fourniront des mesures inattendues ou entachées de bruit : si un amer perçu ne peut pas être mis en relation avec un symbole connu, ou qu'il est mis en relation avec un symbole incorrect, le système réagira de manière erronée. Un tel système ne peut donc pas être considéré comme entièrement autonome. D'autres approches s'avèrent nécessaires pour faire face aux nombreux problèmes inhérents au monde réel, tels que l'incertitude des connaissances, le bruit entachant les grandeurs mesurées par les capteurs, les modifications dynamiques de l'environnement, la nécessité de répondre rapidement et correctement à des situations inattendues, etc [6].

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

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

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

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

### **I.4. L'autonomie d'un robot**

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

Selon Steels [7], un système est autonome s'il développe les lois et les stratégies qui lui permettent de contrôler son comportement. En fait, l'autonomie est une capacité relative et on peut considérer qu'il existe une progression insensible du niveau le plus bas au plus élevé. Elle représente la capacité à choisir une stratégie, en termes de sous-buts pour arriver à un but fixé. La limite basse de l'autonomie est constituée par un système automatique qui s'autorégule en fonction de lois préétablies i.e. qui ne génère pas les lois que les activités de régulation cherchent à satisfaire.

L'autonomie désigne littéralement la capacité d'une entité à « *se gouverner par ses propres lois* ». Cela signifie entre autres le choix de ses buts. Dans la littérature robotique, on peut distinguer trois points de vue :

- L'autonomie au sens fort qui nous ramène à des questions de volonté et de but propre.
- L'autonomie au sens faible désigne la capacité de maintenir sa structure au sein d'un milieu complexe à travers des mécanismes tels que l'auto-organisation, l'évolution, l'adaptation et l'apprentissage.

- L'automatisme ou absence de contrôle extérieur est le sens implicite que l'on attribue généralement au mot autonomie.

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

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

Un robot complètement autonome n'est pas forcément souhaitable. On peut vouloir en prendre le contrôle à certains moments ou intervenir indirectement à travers la modification de certains paramètres. Le terme d'autonomie ajustable désigne justement l'interruption temporaire de l'automatisme, du fait d'un superviseur ou de l'agent lui-même, afin d'accroître l'efficacité du système [8]. Le choix du degré d'automatisme dont doit être doté un agent autonome dépend de l'architecture de contrôleur.

### **1.4.1. Autonomie du mouvement et autonomie décisionnelle.**

L'autonomie du mouvement passe par la détermination des déplacements, de manière planifiée et réactive, et par la commande des déplacements. Si les méthodes mises en jeu sont fortement dépendantes des modèles cinématiques et dynamique des engins considérés, des méthodes génériques commencent à apparaître, surtout pour la planification de trajectoire et la commande référencée sur des éléments de l'environnement. Dans ce contexte, la taille de l'environnement considéré nous amènera plus particulièrement à nous intéresser à l'intégration de méthodes locales de gestion des mouvements avec des méthodes plus globales, qui considèrent l'ensemble des informations disponibles sur l'environnement et la mission à réaliser. Ces derniers aspects relèvent surtout de l'autonomie décisionnelle : il s'agit de déterminer les modalités de déplacement à adopter et les informations à acquérir.

Les systèmes automatiques sont entièrement prévisibles, du moment que leur état interne est connu, alors que les systèmes autonomes ne le sont pas, puisqu'ils sont capables de prendre eux-mêmes des décisions en fonction de critères qui peuvent échapper à l'observateur.

L'autonomie implique la liberté de contrôle. Un système autonome prend ses décisions lui-

même, il n'est pas contrôlé par un agent externe. La prise de décision implique une capacité à évaluer des alternatives en fonction d'un état courant et de l'expérience acquise par le passé [6].

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

### **I.4.2. Autonomie envisageable pour les robots actuels**

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

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

### **I.5. Environnement dynamique et incertain**

#### **I.5.1. Notion d'environnement dynamique**

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

#### **I.5.2. Notion d'incertitude**

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

### **I.6. Problématiques de la robotique mobile**

La robotique mobile cherche depuis des années à rendre une machine mobile autonome face à son environnement pour qu'elle puisse sans intervention humaine accomplir les missions qui lui sont confiées. Le spectre des missions que les roboticiens veulent voir accomplir par leurs machines est immense : exploration en terrain inconnu, manipulation d'objets, assistance aux personnes handicapées, transport automatisé, etc [9].

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

Enfin, une autre capacité robotique tout aussi importante que celles énumérées précédemment, est la capacité pour un robot de prendre lui-même ses propres décisions nécessaires pour réaliser et coordonner des missions complexes. Cette capacité est d'une grande importance puisque, pour beaucoup de tâches robotiques, il peut exister de nombreuses façons de les réaliser. Par un raisonnement, le robot doit alors sélectionner les meilleures actions à effectuer pour réussir adéquatement sa mission [10]. Bien évidemment, l'aspect matériel, qui consiste à choisir et dimensionner aussi bien la structure mécanique du système que sa motorisation, son alimentation et l'architecture informatique de son système de contrôle-commande apparaît comme le premier point à traiter. Le choix de la structure est souvent effectué parmi un panel de solutions connues et pour lesquelles on a déjà résolu les

problèmes de modélisation, planification et commande.

Le choix des actionneurs et de leur alimentation est généralement assez traditionnel. La plupart des robots mobiles sont ainsi actionnés par des moteurs électriques à courant continu, alimentés par des convertisseurs de puissance fonctionnant sur batterie. De la même façon, les architectures de contrôle- commande des robots mobiles ne sont pas différentes de celles des systèmes automatiques ou robotiques plus classiques. On y distingue cependant, dans le cas général, deux niveaux de spécialisation, propres aux systèmes autonomes : une couche décisionnelle, qui a en charge la planification et la gestion (séquentielle, temporelle) des événements et une couche fonctionnelle, chargée de la génération en temps réel des commandes des actionneurs [11].

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

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

Un robot mobile est commandé par une boucle de contrôle (ou sens général), cela est illustré à la figure I.4. De façon itérative, cette boucle fait une lecture des données reçues par les capteurs, les interprète, calcule les commandes motrices et les envoie aux actionneurs. Typiquement, cette boucle est exécutée environ dix (10) fois par seconde ; la fréquence peut varier selon les types de capteurs et d'actionneurs utilisés. La boucle de contrôle n'est pas unique ; selon l'architecture utilisée, elle peut être décomposée en plusieurs sous boucles de contrôle agencées de manières différentes [10].

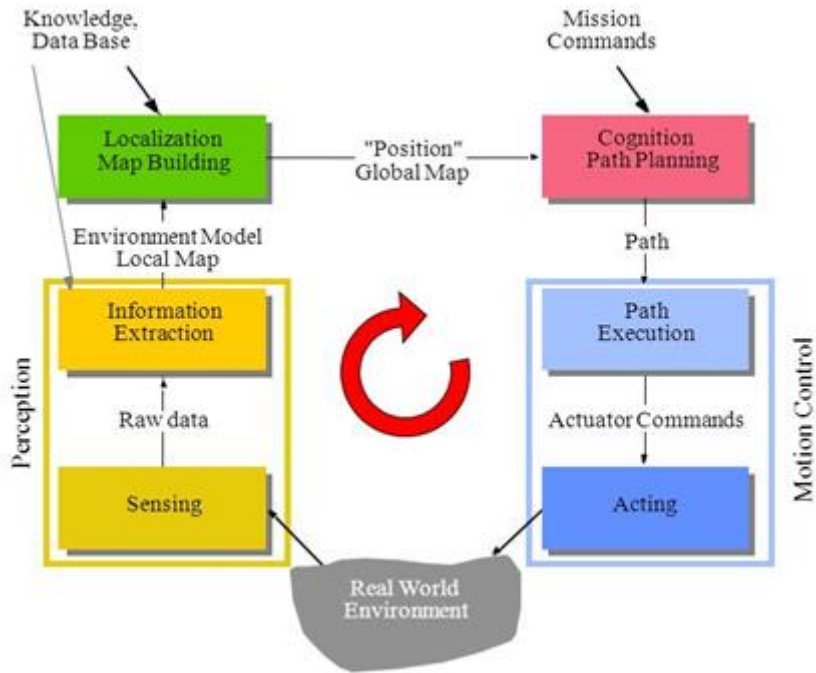


Figure I.9. Structure de commande générale d'un robot mobile.

### I.7. Architectures de contrôle

Un robot est donc composé d'un ensemble de modules, chacun étant responsable d'une ou plusieurs fonctions. Un des premiers défis à résoudre est de déterminer comment relier efficacement les différents modules. Pour ce faire, il faut élaborer une architecture de contrôle qui dictera les responsabilités de chacun des modules et comment les informations circuleront entre ces derniers. Depuis les débuts de la robotique, beaucoup d'architectures ont été proposées. Elles peuvent être généralement classées en trois grandes catégories : délibérative, comportementale et hybride.

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

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

### I.7.1. Architectures de contrôle cognitives

Dans une architecture de contrôle cognitive, le robot cherche tout d'abord à modéliser les connaissances et les méthodes d'inférence associées permettant d'élaborer les actions que le robot doit entreprendre. Le robot commence par traiter les données recueillies par ses capteurs. Ensuite, il identifie les objets qui sont dans son environnement proche. Puis il construit une représentation de la scène dans son ensemble, et l'utilise pour générer un plan. Après quoi, il calcule au mieux une séquence de commandes vers les effecteurs pour exécuter le plan prévu (voir figure II.10).

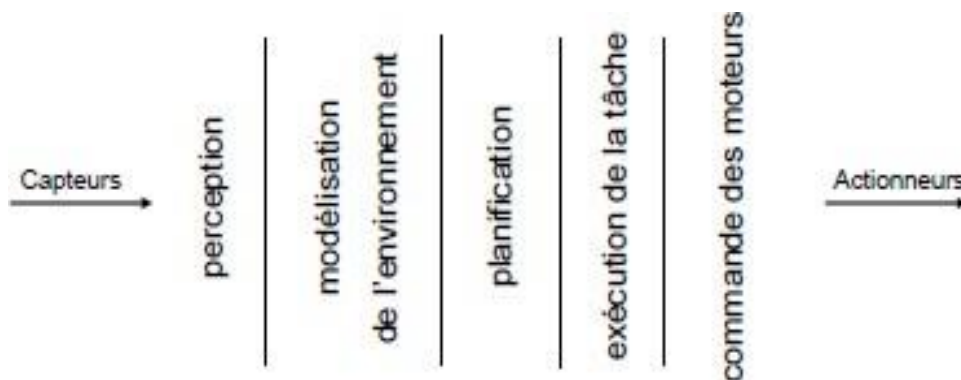


Figure I.10. Architecture de contrôle cognitive.

### I.7.2. Architectures de contrôle réactives

Les architectures réactives ont pour but de concevoir et de réaliser un robot, capable d'apporter une réponse immédiate à toute nouvelle modification de l'environnement le concernant. Ces approches sont initialement fondées sur l'étude de l'interaction de l'animal avec son environnement naturel (l'éthologie) de manière à essayer de comprendre les mécanismes mis en œuvre et de les reproduire. Elles sont fondées sur le fait que lorsqu'un être humain évolue dans un environnement, il ne passe pas son temps à modéliser son environnement, et à planifier ses actions, mais qu'au contraire il réagit à de simples stimuli qui lui permettent de se déplacer vers son but tout en faisant face aux obstacles. Ainsi, contrairement aux systèmes hiérarchiques, le comportement global du robot apparaît comme le résultant de plusieurs comportements actifs simultanément. Dans ce type d'architectures il n'y a pas de phase de planification ou de modélisation de l'environnement (voir figure II.11).

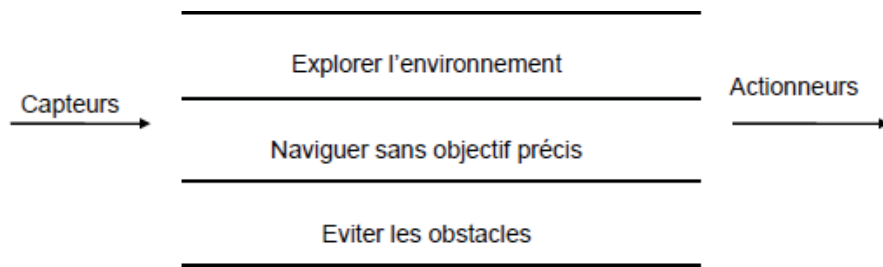


Figure I.11 Architecture de contrôle réactive.

### I.7.3. Les architectures de contrôle comportemental réactif

Pour l'élaboration d'architecture de contrôle réactif de plus en plus complexe) propose de décomposer le comportement global du robot en un ensemble d'entrées élémentaires. Cette décomposition a comme principaux objectifs d'avoir une construction aisée de l'architecture de contrôle ainsi qu'une testabilité facile des comportements élémentaires. Ceci est rendu possible en isolant les comportements qui cohabitent dans une même structure de contrôle, impose l'adaptation de mécanismes approprié de coordination entre comportements.

Les architectures de contrôle réactives permettent d'avoir une réponse plus rapide grâce à un lien direct capteurs-actionneurs. Disposer des comportements élémentaires dans le cas des approches comportementales permet de les tester individuellement jusqu'à ce qu'ils soient adaptés à leurs tâches. La mise à jour se fait alors simplement en les ajoutant aux comportements déjà existe. [12], [13] ont comparé les caractéristiques des deux types architectures de contrôle. La Figure II.12 illustre la comparaison entre les deux architectures selon les degrés d'intelligences.



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

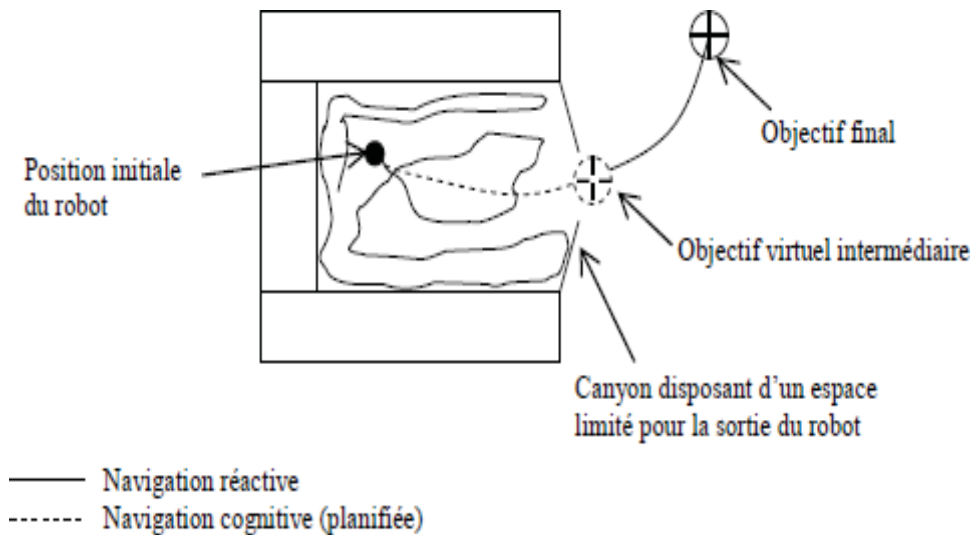


Figure I.13 Architecture de contrôle hybride.

### II.8 Conclusion

Le développement de l'intelligence artificielle classique a conduit à la réalisation des premiers robots mobiles, évoluant dans des environnements strictement contrôlés. Leurs actions dans le monde réel n'étaient que l'expression physique d'opérations symboliques réalisées dans un modèle abstrait. Ces robots étaient incapables de réagir de manière appropriée à des perturbations ou à des événements inattendus, ce qui a rendu impraticable leur utilisation dans le monde réel. L'objectif de ce chapitre est de fournir un aperçu général sur la robotique mobile, ses problématiques et les solutions apportées actuelles. Nous avons présenté également un panorama des architectures de contrôle dont la portée va au-delà de leur application en robotique mobile. La robotique mobile est un domaine dans lequel l'expérience pratique est primordiale. L'approche de la robotique présentée dans ce chapitre est essentiellement celle issue de l'intelligence artificielle. D'autres disciplines très importantes, telles que l'automatique sujet traité dans ce mémoire, sont toutefois impliquées et peuvent présenter un éclairage très significatif et différent, mais ils ne changent pas

## Chapitre I : Généralité sur la robotique mobile

fondamentalement les problèmes qui restent à résoudre.

Dans le chapitre suivant nous allons parler ou profond sur la cinématique de quelques types de robots mobiles, chose qui à nous permettre de comprendre le contenu des chapitres qui viennent par la suite.

## Chapitre II : La cinématique du robot mobile

### II.1. Introduction

La modélisation mathématique est une étape très importante pour la commande des robots. Deux types de modèles sont généralement utilisés lors de la commande, à savoir : le modèle cinématique et le modèle dynamique. D'après la littérature, on rencontre plusieurs types de robots à savoir : les robots de type unicycle, les robots de type tricycle, et les robots de type voiture (vus au chapitre I). Dans le cadre de notre travail, nous utiliserons un robot de type unicycle à cause de sa simplicité de construction et de ses propriétés cinématiques intéressantes. Plusieurs robots ont fait l'objet d'étude de certains chercheurs, mais un a retenu leur attention : le robot mobile de type unicycle.

### II.2. Définitions

On note  $R = (O, \vec{x}, \vec{y}, \vec{z})$  un repère fixe quelconque, dont l'axe  $\vec{z}$  est vertical et  $\hat{R} = (O, \vec{x}, \vec{y}, \vec{z})$  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 illustrée sur la Figure II.1.

Par analogie avec la manipulation, on appelle situation [14] ou souvent posture [15] du robot le vecteur suivant :

$$\xi = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$$

Où  $x$  et  $y$  sont respectivement l'abscisse et l'ordonnée du point  $O$  dans  $R$  et  $\theta$  l'angle  $(\vec{x}, \vec{x})$ . La situation du robot est donc définie sur un espace  $\mathcal{M}$  de dimension  $m = 3$ , comparable à l'espace opérationnel d'un manipulateur plan.

La configuration [16] 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 :

$$q = \begin{pmatrix} q_1 \\ q_2 \\ \dots \\ q_n \end{pmatrix}$$

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

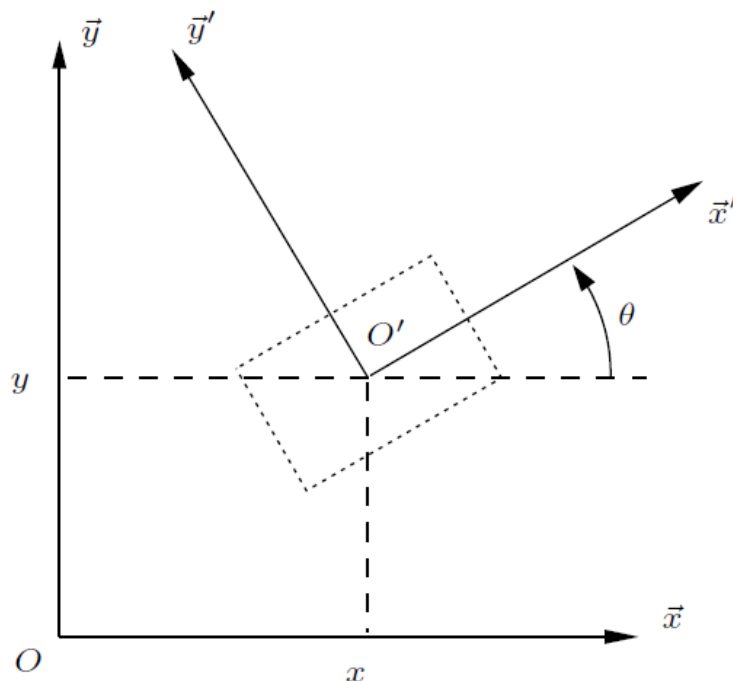


Figure II.1. Repérage d'un robot mobile.

### II.3. Roulement sans glissement et contraintes non holonômes

#### II.3.1. 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 (régulière, matériaux en contact) a une forte influence sur les priorités 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 équilibrées 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,  $\varphi$  l'angle de rotation propre de la roue et  $\theta$  l'angle entre le plan de la roue et le plan  $(O, \vec{x}, \vec{z})$  comme indiquées à la Figure II.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  $\vec{v}_P$  du centre  $P$  de la roue et le vecteur vitesse de rotation  $\vec{\omega}$  de la roue :

$$\vec{v}_Q = \vec{v}_P + \vec{\omega} \wedge \overrightarrow{PQ} = \vec{0}$$

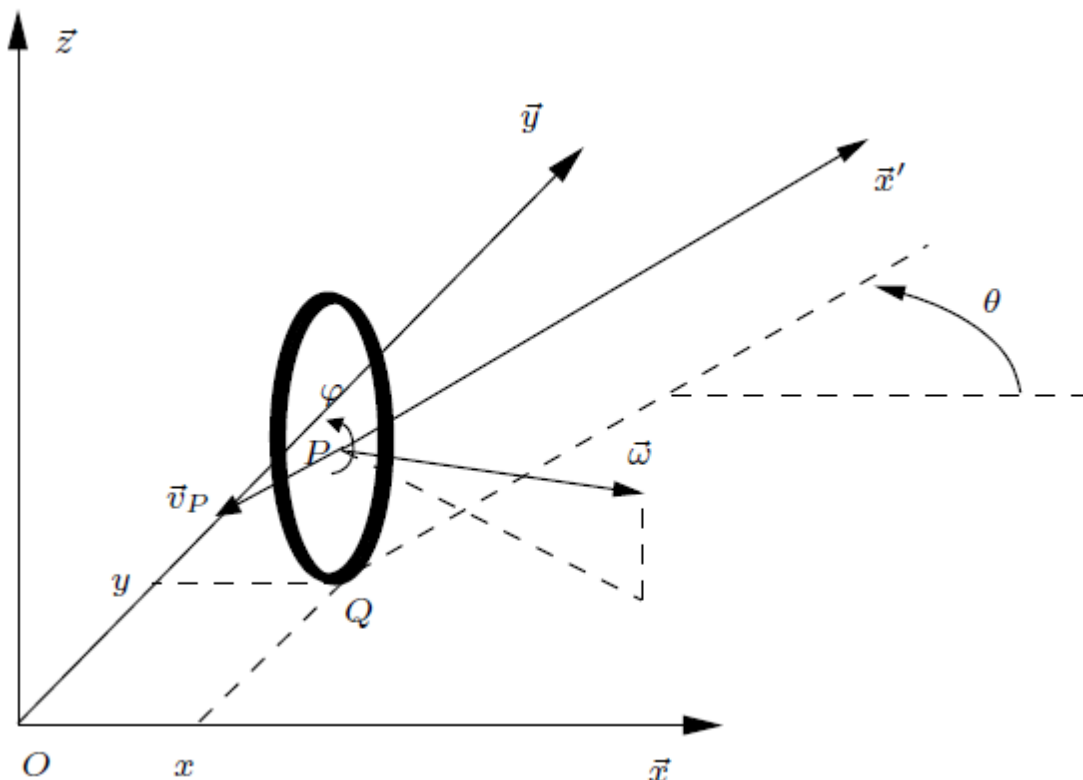


Figure II.2. Caractérisation du roulement sans glissement.

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

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

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

Ceci nous donne le système de contraintes scalaires :

$$\dot{x} + r\dot{\varphi}\cos\theta = 0, \tag{II.1}$$

$$\dot{y} + r\dot{\varphi}\sin\theta = 0, \tag{II.2}$$

Que l'on puisse 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 :

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0, \tag{II.3}$$

$$-\dot{x}\cos\theta + \dot{y}\sin\theta = -r\dot{\varphi}, \tag{II.4}$$

Ces contraintes traduisent le fait que le vecteur  $\vec{v}_P$  soit dans le plan de la roue et ait pour module  $r\dot{\varphi}$ .

### **II.3.2. Contraintes non holonomes**

Les équations précédentes, caractérisant le *r.s.g.* 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 (pfaffienne)  $A^T(q)\dot{q} = 0$ . 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 n'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 a 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 [17] ou de commande non-linéaire [18]. Seule la connaissance du crochet de Lie est nécessaire à notre étude.

Pour deux vecteurs  $b_i(q)$  et  $b_j(q)$ , cet operateur est défini par  $[b_i(q), b_j(q)] = \frac{\partial b_i}{\partial q} b_j - \frac{\partial b_j}{\partial q} b_i$ .

**Théorème II.1 :** 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 (II.3) et (II.4) on peut déterminer :

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

Et on déduire :

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

Matrice formée par les colonnes indépendantes  $b_1(q) = \left( \cos \theta \ \sin \theta \ 0 \ -\frac{1}{r} \right)^T$  et  $b_2(q) = \left( 0 \ 0 \ 1 \ r \right)^T$  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  $b_2(q)$ :

$$[b_1(q), b_2(q)] = (\sin \theta \quad -\cos \theta \quad 0 \quad 0)^T \quad (II.5)$$

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

$$[b_2(q), [b_1(q), b_2(q)]] = (\cos \theta \quad \sin \theta \quad 0 \quad 0)^T \quad (II.6)$$

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 (II.3) et (II.3) sont donc des contraintes non holonomes.

## II.4. Les grandes classes de robots mobiles et leurs modèles

### II.4.1. 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 (voir Figure II.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.

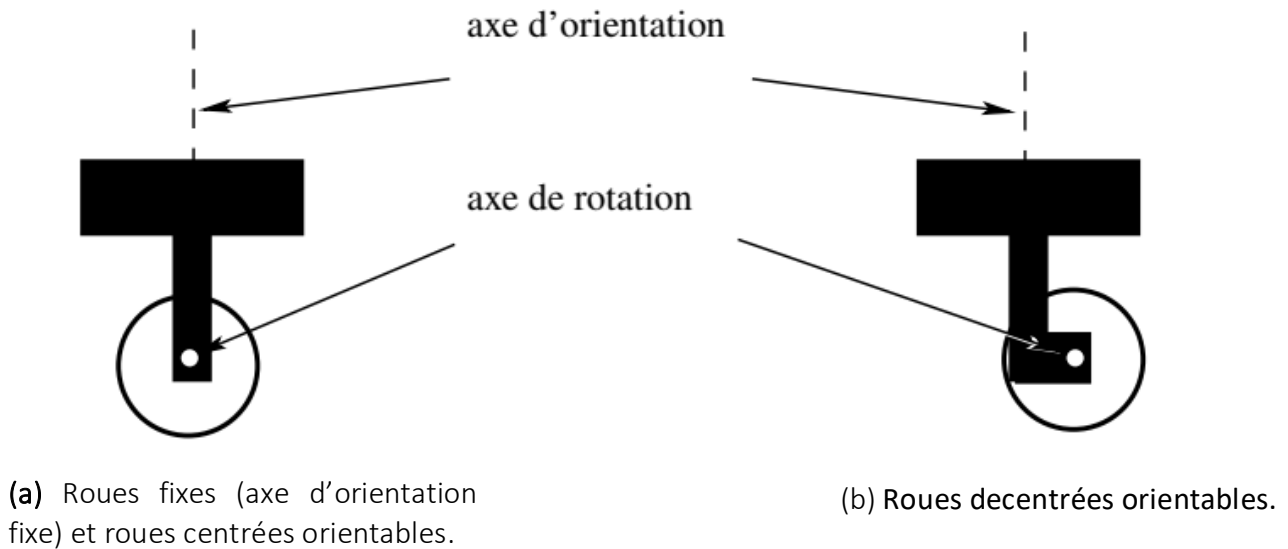


Figure II.3. Les principaux types de roues des robots mobiles.

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 ! 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.

II.4.2. Robots mobiles de type unicycle

II.4.2.1. 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 II.4. On y a omis les roues folles, qui n'interviennent pas dans la cinématique, dans la mesure où elles ont été judicieusement placées.

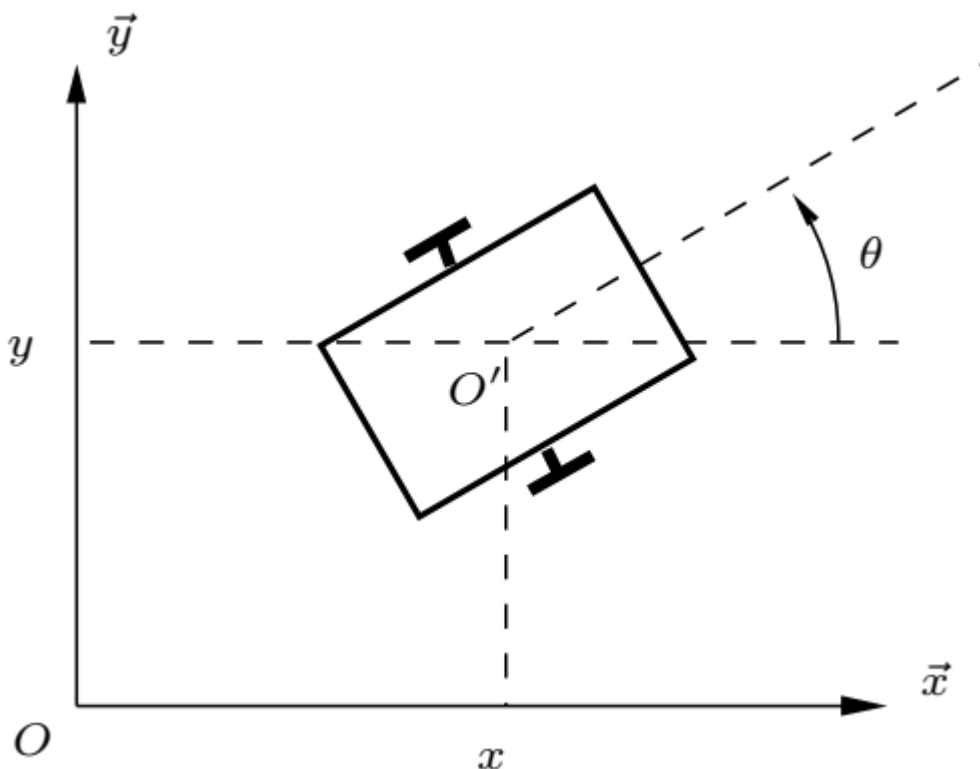
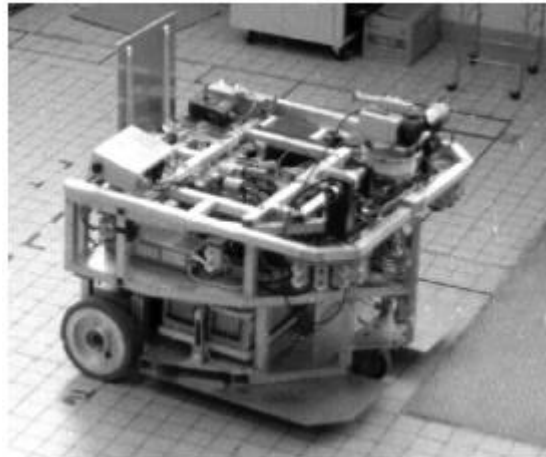


Figure II.4. Robot mobile de type unicycle.

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

Ci-dessous, sont présentés comme exemples trois types de robots (célèbres) mobiles de type unicycle :



**Figure II.5.** Robot mobile *Hilare*, LAAS-CNRS, Toulouse, 1977 [19].

**Ex.1.** Informations techniques (robot *Hilare*, Figure II.5) ;

- Entraînement : batteries au plomb 24V, 2 moteurs DC avec codeurs incrémentaux,
- Calculateur : 4 processeurs Intel 80286, pas d'OS, multi-bus, modem série radio à 9600 bauds,
- Capteurs : odométrie, 16 capteurs US, un télémètre laser,
- Dimensions (L × l × h) : 80 cm × 80 cm × 60 cm,
- Poids : 400 kg.



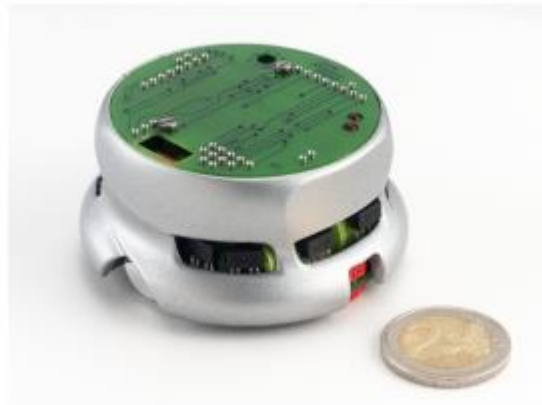
**Figure II.6.** Robot mobile *Pioneer P3-DX*, ActiveMedia Robotics, 2004 [20].

**Ex.2.** Informations techniques (*Pioneer P3-DX*, ActiveMedia Robotics, 2004, Figure II.6);

- Entraînement : batteries 252 W h, 2 moteurs DC avec codeurs incrémentaux,
- Calculateur : microcontrôleur Hitachi HS-8, I/O Bus, 2 ports série,

## Chapitre II : la cinématique du robot mobile

- Capteurs : odométrie, 8 capteurs US en façade + options (dumpers, 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 II.7.** Robot mobile *Khepera II*, K-team, EPFL, Lausanne, 2002 [21].

**Ex.3.** Informations techniques (*Khepera II*, K-team, EPFL, Lausanne, 2002, Figure II.7) ;

- 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,
- Autonomie : 1 h en fonctionnement continu, l'ajout d'extensions diminuant l'autonomie,
- Vitesse: maximum 1 m/s, minimum 0; 02 m/s,
- Dimensions (D × h) : 7 cm × 3 cm,
- Poids : 80 g (charge admissible : 250 g).

**II.4.2.2. Modélisation**

**Centre instantané de rotation** : Les roues motrices ayant même axe de rotation, le CIR du robot 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  $O'$  (voir Figure II.8). Soit  $L$  l'entre-axe et  $\omega$  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 II.8, vérifient :

$$v_d = -r\dot{\varphi}_d = (\rho + L)\omega \quad (II.6)$$

$$v_g = r\dot{\varphi}_g = (\rho - L)\omega \quad (II.7)$$

Ce qui permet de déterminer  $\rho$  et  $\omega$  à partir des vitesses des roues :

$$\rho = L \frac{\dot{\varphi}_d - \dot{\varphi}_g}{\dot{\varphi}_d + \dot{\varphi}_g} \quad (II.8)$$

$$\omega = -\frac{r(\dot{\varphi}_d + \dot{\varphi}_g)}{2L} \quad (II.9)$$

L'équation (2.8) permet de situer le CIR sur l'axe des roues. Par ailleurs ces équations expliquent deux propriétés remarquables du mouvement des robots de type unicycle : si  $\dot{\varphi}_d = -\dot{\varphi}_g$ , le robot se déplace en ligne droite ; et 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.

**II.4.2.3. 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 :

$$v = \frac{v_d + v_g}{2} = \frac{r(\dot{\varphi}_g - \dot{\varphi}_d)}{2} \quad (II.10)$$

D'autre part, la vitesse de rotation du robot est égale à la vitesse de rotation autour du CIR [25]:

$$\omega = \dot{\theta} = -\frac{r(\dot{\varphi}_g + \dot{\varphi}_d)}{2L} \quad (II.11)$$

Conformément à l'équation (2.9). On montre que ces relations sont parfaitement inversibles et qu'il y a ainsi équivalence entre les couples  $(\dot{\varphi}_d, \dot{\varphi}_g)$  et  $(v, \omega)$ . Desormais, on utilise plutôt ce dernier couple de grandeurs, plus parlantes, quitte à calculer ensuite les angles ou vitesses de consigne des asservissements des roues.

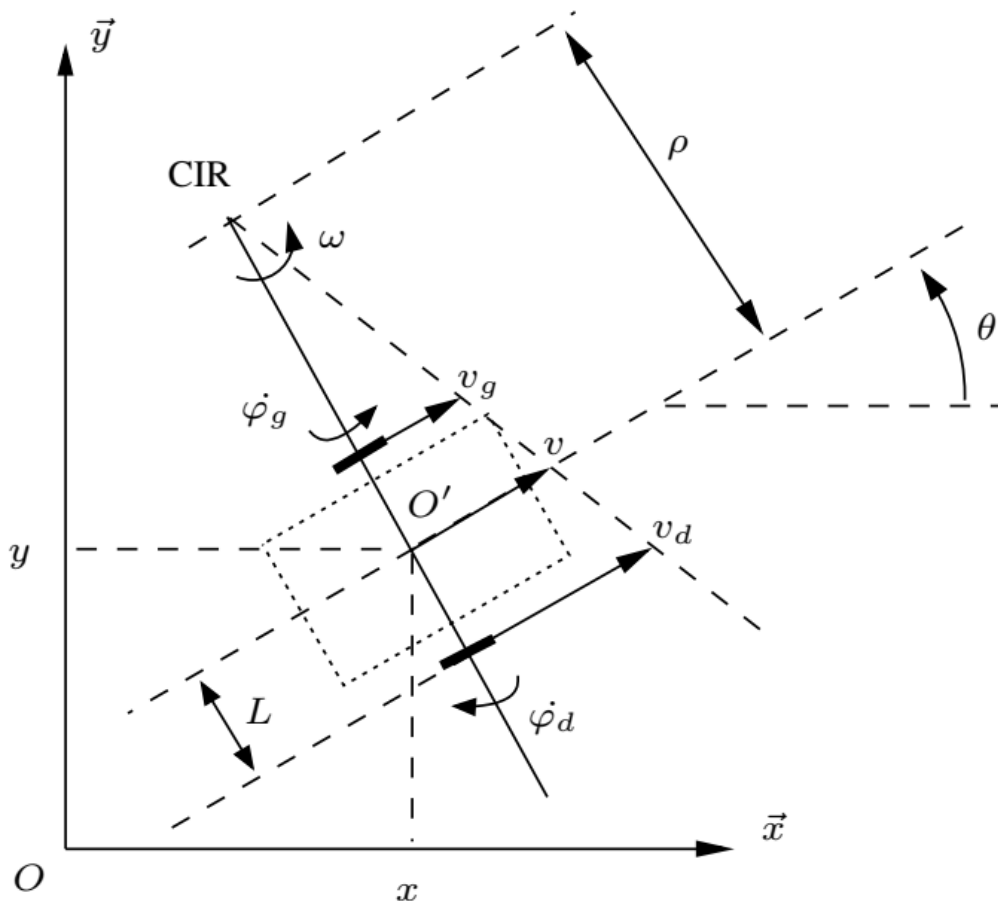


Figure II.8. Centre instantané de rotation d'un robot de type unicycle.

#### II.4.2.4. Modèle cinématique en posture

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

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

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

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

Ce qui s'écrit, sous forme matricielle :

$$\dot{\xi} = C(q)u \quad (II.12)$$

Avec ;

$$C(q) = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \quad (II.13)$$

Ce modèle est appelé *modèle cinématique en posture* du robot [15].

#### II.4.2.5. Modèle cinématique en configuration

Considérons maintenant l'expression des conditions de *r.s.g.* des deux roues. On transpose les équations (II.1) et (II.2) aux deux roues de l'unicycle. Il vient alors :

$$\begin{cases} \dot{x} + L\dot{\theta} \cos \theta + r\dot{\varphi}_d \cos \theta = 0 \\ \dot{y} + L\dot{\theta} \sin \theta + r\dot{\varphi}_d \sin \theta = 0 \\ \dot{x} - L\dot{\theta} \cos \theta - r\dot{\varphi}_g \cos \theta = 0 \\ \dot{y} - L\dot{\theta} \sin \theta - r\dot{\varphi}_g \sin \theta = 0 \end{cases}$$

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 :

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

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 :

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

Par ailleurs il résulte des relations (2.9) et (2.11) la contrainte :

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

Qui s'intègre en :

$$\theta = -\frac{r(\varphi_d + \varphi_g)}{2L} + \theta_0$$

Où  $\theta_0$  est la valeur de  $\theta$  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 distinguant, d'après ce qui précède, que :

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega \\ \dot{\varphi}_d &= -\frac{v}{r} - \frac{L\omega}{r}\end{aligned}$$

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

$$\dot{q} = B(q)u \quad (\text{II.14})$$

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 [22]. 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  $\varphi_d$  n'apparaît pas dans les équations régissant la dérivée de la posture, on se contentera généralement du modèle simplifié :

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega\end{aligned} \quad (\text{II.15})$$

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

**II.4.3. Robots mobiles de type tricycle et de type voiture**

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

**II.4.3.1. Description**

Considérons tout d'abord le cas du *tricycle*, représenté à la Figure II.9. 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.

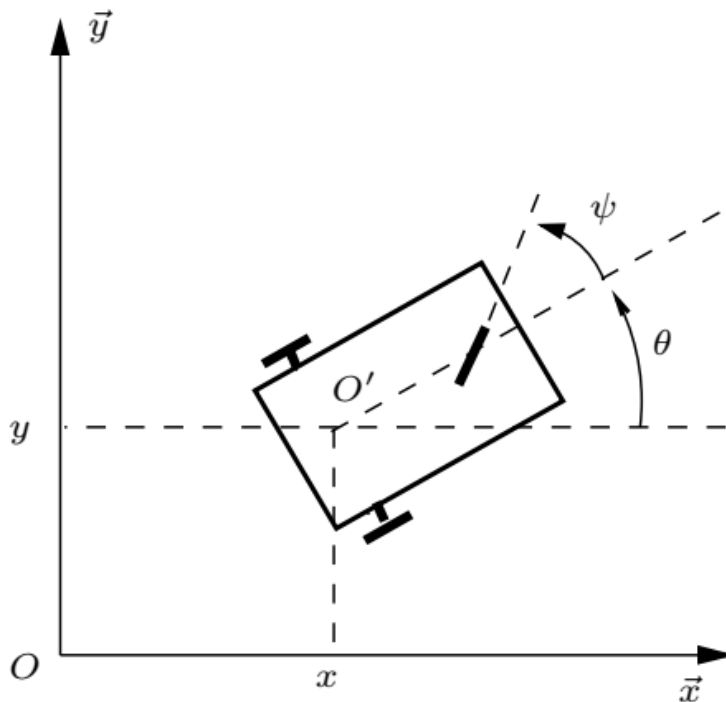


Figure II.9. Robot mobile de type tricycle.

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 soi, on rencontre beaucoup plus souvent ce type de systèmes. On parle de robot dès lors que la voiture considérée est autonome [23,24], donc sans chauffeur ni télé-pilotage. Il s'agit là d'un des grands défis issus de la robotique mobile. Deux réalisations sont montrées à la Figure II.10, basées sur des voitures de série instrumentées.



Figure II.10. Projets de voitures autonomes à l'université de Carnegie Mellon [23,24].

#### II.4.3.2. Modélisation

Considérons tout d'abord le cas 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 II.11. On peut déterminer  $\rho$  de manière géométrique à partir de l'angle d'orientation de la roue avant et  $\omega$  à partir de la vitesse linéaire  $v$  du véhicule (vitesse en  $O'$ ) et de  $\rho$  :

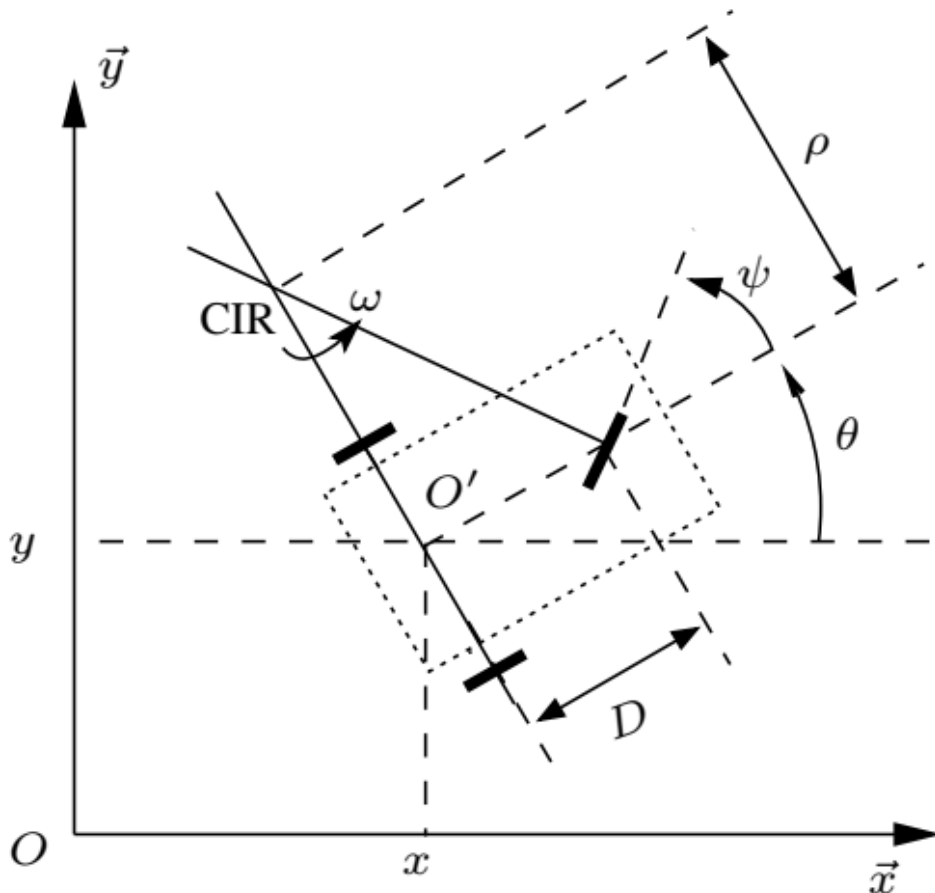


Figure II.11. Un robot mobile de type tricycle et son CIR.

$$\rho = \frac{D}{\tan(\varphi)} \quad (\text{II.16})$$

$$\omega = \frac{v}{D} \tan(\varphi) \quad (\text{II.17})$$

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  $\varphi$  de telles que  $-\frac{\pi}{2} < \varphi < \frac{\pi}{2}$ , 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 :

$$\begin{aligned} \dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \frac{v}{D} \tan(\varphi) \\ \dot{\varphi} &= \eta \end{aligned} \quad (\text{II.18})$$

Où  $u = (v \ \eta)^T$  est le vecteur de commande cinématique,  $\eta$  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 II.12.

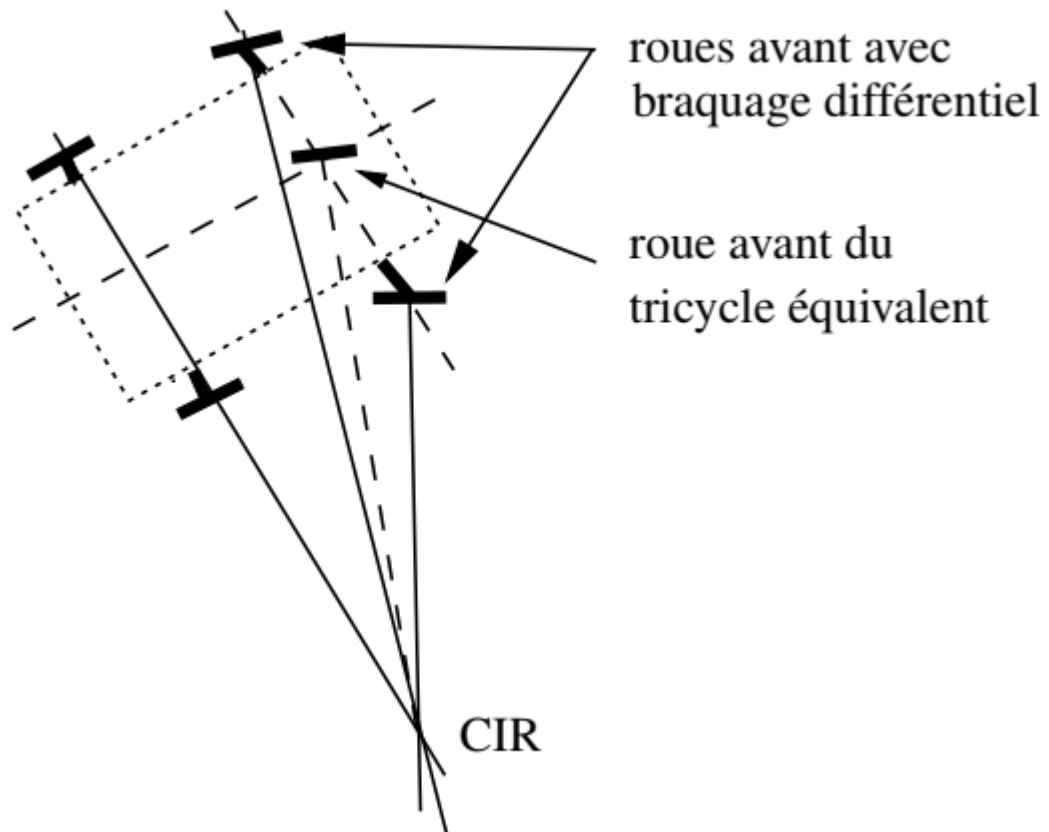


Figure II.12. Un robot mobile de type tricycle et son CIR.

Ainsi, les roues avant d'un robot de type voiture (et a 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 II.12.

**II.4.4. Robots mobiles omnidirectionnels**

**II.4.4.1. Description**

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

D'un point de vue cinématique on montre que cela n'est pas possible avec des roues fixes où des roues centrées orientables [15]. On peut en revanche réaliser un robot omnidirectionnel en ayant recours à un ensemble de trois roues décentrées orientables où de trois roues suédoises disposées aux sommets d'un triangle équilatéral (voir Figure II.13). Du point de vue de la transmission du mouvement, ceci ne va pas sans poser de problème.

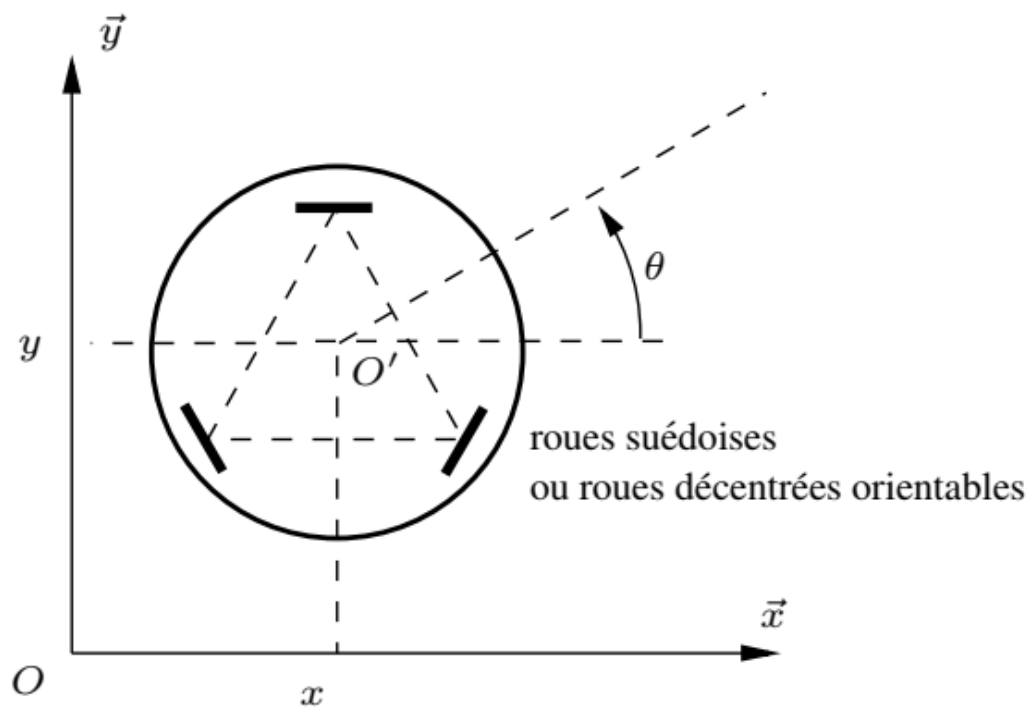


Figure II.13. Représentation d'un robot mobile omnidirectionnel.

### II.4.4.2. Modélisation

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

$$\begin{aligned}\dot{x} &= u_1 \\ \dot{y} &= u_2 \\ \dot{\theta} &= u_3\end{aligned}\tag{II.19}$$

Où  $u = (u_1 \ u_2 \ u_3)^T$  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 surcout. On citera pour mémoire le cas du XR4000 Nomadic représenté à la Figure II.14, disparu en 2000...



Figure II.14. Robot mobile omnidirectionnel Nomadic XR4000.

## II.5. Propriétés du modèle cinématique d'un robot mobile

### II.5.1. Représentation d'état

A partir de la modélisation que nous avons présentée tout au cours de ce chapitre il résulte, dans tous les cas, un modèle dynamique (au sens de l'Automatique) sous la forme (II.14). Si l'on considère que la configuration (réduite)  $q$  fait office de variable d'état du système on a :

$$\dot{x} = B(x)u \quad (II.20)$$

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 chaines*, comme on le précisera au paragraphe consacré à la commande des robots.

### II.5.2. Commandabilité des robots mobiles

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 linéarise autour de tout point d'équilibre est commandable. En  $x_0$  quelconque, le modèle linéarisé issu de (2.20) s'écrit :

$$\dot{x} = B(x_0)u$$

Avec  $B(x_0) \in \mathbb{R}^{n \times m}$ . 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 traduit comme suit :

$$\text{Rang}(x_0) = n$$

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 (II.20).

***Théorème II.1 : 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 (II.20), 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.*

On considère le cas de l'unicycle. A partir du modèle cinématique (II.14) et des crochets (II.5) et (II.6) on obtient 4 vecteurs indépendants pour un système de dimension 4. Le modèle cinématique de l'unicycle est donc commandable. On montrerait de même que celui de la voiture l'est aussi. La conséquence de cette propriété est l'existence de retours d'états pour commander le système, comme on le verra au chapitre 3.

### II.6. Conclusion

Dans ce chapitre, nous avons présenté des notions très importantes relatives aux robots mobiles au sens général. Nous avons parlé sur l'état général lorsque nous avons le cas des roulements sans glissement ainsi la signification de ce que on appelle contraints non holonomes.

En outre, trois grandes classes de robots mobiles et leurs modèles ont été présenté ; à savoir ; *le type unicycle, le type tricycle et le type omnidirectionnel*. Ce chapitre est terminé par la présentation de quelques propriétés du modèle cinématique d'un robot mobile.

Pareillement, il est à noter que, dans ce mémoire le type de robot sur lequel articulé l'objectif envisagé et le type *unicycle*.

## Chapitre III : Problème de suivi de trajectoire

### III.1. Introduction

Le problème de suivi d'une trajectoire de référence pour un robot mobile non-holonome est apparu comme un problème de premier ordre pour la communauté roboticienne dans ces dernières années. En effet, la forte utilisation des robots mobiles dans les domaines où l'être humain ne peut pas être présent, notamment dans les sites nucléaires à haut risque ou dans le cas de l'exploration spatiale, nécessite la mise en œuvre de lois de commande autonomes et performantes pour assurer les tâches assignées aux robots. Plusieurs travaux concernant la poursuite de trajectoire ont été développés dans ce contexte [1,2].

Dans ce chapitre, le problème de suivi de trajectoire sans exception (glissements, contraintes externes,...) est abordé. Plus précisément, nous parlerons sur ce problème aux deux sens généraux (i) *contrôle d'orientation* et (ii) *stabilisation de mouvement* avec leurs inverses (*contrôle/sans contrôle et stabilisation/sans stabilisation*).

### III.2. Problème de suivi de trajectoire

Le suivi de trajectoire est une tâche primordiale pour les robots autonomes. Simple et facile à implémenter, les modèles cinématiques sont les plus utilisés dans la littérature. Cette solution est satisfaisante pour les robots ayant des caractéristiques mécaniques simples et des vitesses déviation relativement réduites. On peut citer ici l'exemple des robots d'exploration où le modèle cinématique garde ses performances et sa robustesse.

Dans ce qui suit, nous parlons sur ce que nous appelons exécution du mouvement en termes de suivi d'une trajectoire.

#### *III.2.1. Les différents problèmes*

##### *III.2.1.1. Définitions et hypothèses*

On se réfère souvent au terme de commande pour décrire ce problème. Il s'agit bien évidemment d'un problème de commande dans la mesure où l'on cherche à déterminer les

lois qui permettent au robot de suivre une trajectoire donnée. Ceci étant il est prudent d'être plus précis, bien que le vocabulaire en la matière soit relativement peu unifié.

Nous adopterons la terminologie suivante pour désigner les deux problèmes que l'on va évoquer.

On distinguera :

- le *suivi de chemin* (suivi d'une trajectoire), problème qui consiste à trouver la commande pour asservir la distance d'un point du robot a une trajectoire de référence, la vitesse longitudinale de déplacement du robot étant donnée ;

- la *stabilisation de mouvement*, qui consiste à déterminer la commande du système permettant de stabiliser asymptotiquement à zéro l'erreur de suivi du robot par rapport à un robot fictif de référence.

Dans les deux cas, on recherche une commande par retour d'état, puisque la commande en boucle ouverte n'offrant que des possibilités limitées dans un environnement aussi incertain que celui envisagé. Les solutions envisageables sont différentes selon que l'on souhaite ou non contrôler l'orientation du robot.

### III.2.1.2. Paramétrage

Il est à noter que, le paramétrage le plus général des problèmes de commande traités ici est donné à la Figure III.1. Le point  $O'$ , origine du repère  $R'$  est le milieu de l'axe des roues et  $P$  est un point de coordonnées  $(a, b, 0)^T$  dans  $R'$ .

On considère un repère de référence  $R_r = (O_r, \vec{x}_r, \vec{y}_r, \vec{z}_r)^T$  tel que  $\vec{z}_r = z_r$ . On note aussi,  $\theta_r$  l'angle  $(\vec{x}, \vec{x}_r)$  définissant l'orientation de  $R_r$  dans  $R$  et  $\theta_e = \theta - \theta_r$ .

Selon les cas  $R_r$  est sera un *repère de Frénet* sur la trajectoire (suivi de chemin) ou le repère associé à un robot de référence (stabilisation de mouvement).

### III.2.2. Suivi de chemin (suivi d'une trajectoire)

#### III.2.2.1. Problématique

Ce problème consiste à asservir la distance d'un point  $P$  du robot a un chemin (trajectoire) de référence  $C$  que l'on souhaite suivre. Pour cela, on cherche la commande par retour d'état  $\omega$ , la vitesse longitudinale  $v$  étant donnée (cas particulier).

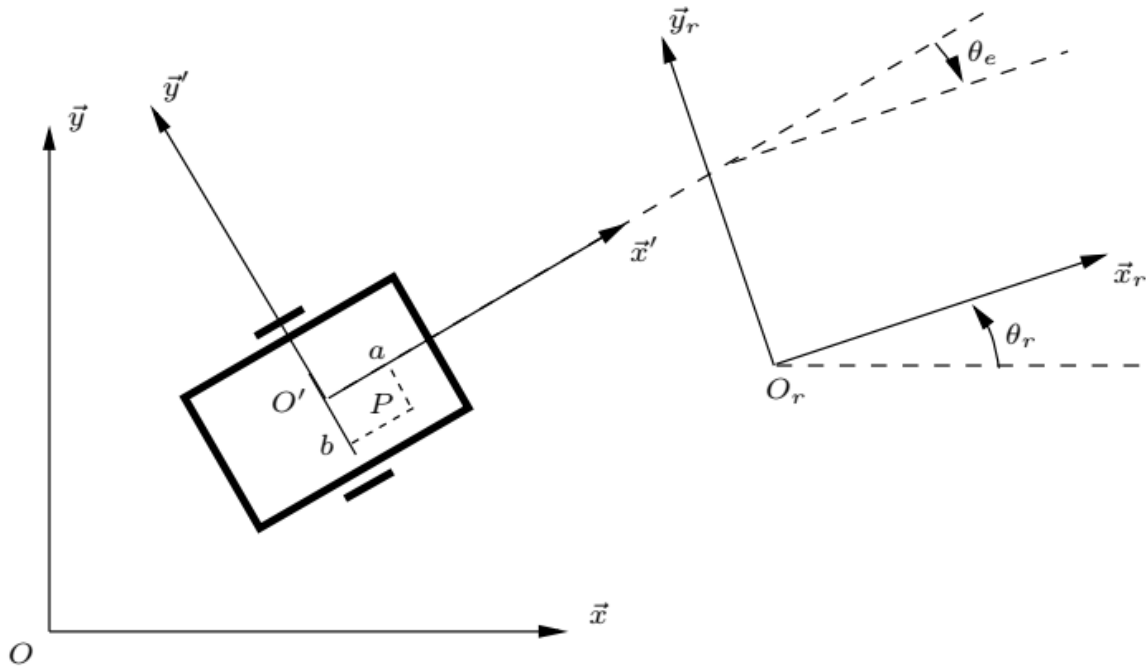


Figure III.1. Paramétrage pour les problèmes de commande.

### III.2.2.2. Paramétrage

Le paramétrage du problème est donné à la Figure III.2. Le repère de référence est choisi de façon à ce que  $O_r$  soit le point de  $C$  le plus proche de  $P$ . Alors  $R_r$  est tel que  $\vec{x}_r$  soit tangent à  $C$  en  $O_r$  et  $\vec{y}_r$  normal à  $C$  en  $O_r$ . On note aussi,  $d$  la distance algébrique de  $O_r$  à  $P$ . Aussi, Le repère  $R_r$  mobile sur la trajectoire de référence  $C$  est un repère de Frènet, le point  $O_r$  étant repéré sur la trajectoire par son abscisse curviligne  $s$ , que l'on suppose normalisée.

Pour que  $O_r$  existe toujours et soit unique, il faut que  $P$  soit toujours à une distance de  $C$  inférieure au rayon de courbure minimum de  $C$ , ce qui s'écrit  $|d| < |p(s)|, \forall s \in [0,1]$ .

D'une manière équivalente, on a :

$$|dc(s)| < 1, \forall s \in [0,1], \quad (\text{III.1})$$

Où  $c(s)$  est la courbure de  $C$  en  $O_r$ .

### III.2.2.3. Modélisation dans le repère de Frènet

Pour le problème de suivi de chemin, la modélisation consiste à déterminer les mouvements de  $P$  et  $R'$  dans  $R_r$  et celui de  $R_r$  sur  $C$ , qui sont complètement caractérisés par la donnée de  $\dot{s}$ ,  $\dot{d}$  et  $\dot{\theta}_e$ .

La courbure  $c(s)$  de  $C$  en  $O_r$  est définie par  $c(s) = \frac{d\theta_e}{ds}$  donc :

$$\begin{aligned}\dot{\theta}_e &= \dot{\theta} - \dot{\theta}_r \\ &= \dot{\theta} - \frac{d\theta_r}{ds} \frac{ds}{dt}\end{aligned}$$

Soit :

$$\dot{\theta}_e = \omega - \dot{s}c(s) \quad , \quad (III.2)$$

La vitesse du point  $P$  vaut d'une part :

$$\begin{aligned}\vec{v}_p &= \vec{v}_{O'} + \vec{\omega} \wedge O'P \\ &= \vec{v}_{x'} + \omega \vec{z} \wedge (a\vec{x} + b\vec{y}) \\ &= (v - b\omega)(\cos\theta_e \vec{x}_r + \sin\theta_e \vec{y}_r) + a\omega(-\sin\theta_e \vec{x}_r + \cos\theta_e \vec{y}_r)\end{aligned}$$

Soit :

$$\vec{v}_p = ((v - b\omega)\cos\theta_e - a\omega\sin\theta_e)\vec{x}_r + ((v - b\omega)\sin\theta_e + a\omega\cos\theta_e)\vec{y}_r \quad (III.3)$$

D'autre part, on peut exprimer la vitesse du point  $P$  par :

$$\begin{aligned}\vec{v}_p &= \frac{d}{dt}(\vec{OP}) \\ &= \frac{d}{dt}(\vec{OO}_r + \vec{O}_rP)\end{aligned}$$

$$\begin{aligned}
 &= \dot{s}\vec{x}_r + \frac{d}{dt}(d\vec{y}_r) \\
 &= \dot{s}\vec{x}_r + \dot{d}\vec{y}_r - d\dot{\theta}_r\vec{x}_r
 \end{aligned}$$

Soit :

$$\vec{v}_p = \dot{s}(1 - dc(s))\vec{x}_r + \dot{d}\vec{y}_r \quad (\text{III.4})$$

On déduit de (III.2), (III.3) et (III.4) que :

$$\dot{s} = \frac{(v-b\omega)\cos\theta_e - a\omega\sin\theta_e}{1-dc(s)} \quad (\text{III.5})$$

$$\dot{d} = (v - b\omega)\sin\theta_e + a\omega\cos\theta_e \quad (\text{III.6})$$

$$\dot{\theta}_e = \omega - \dot{s}c(s) \quad (\text{III.7})$$

#### III.2.2.4. Suivi de chemin sans contrôle d'orientation

On recherche une loi  $\omega$  permettant de stabiliser la distance à la courbe de référence sans contrôle particulier de l'orientation du robot.

On choisit  $P$  située à l'avant du robot sur l'axe longitudinal et donc  $b = 0$ . Alors (III.6) s'écrit :

$$\dot{d} = v\sin\theta_e + a\omega\cos\theta_e \quad (\text{III.8})$$

La loi :

$$\omega = -\frac{v\sin\theta_e}{a\cos\theta_e} - \frac{v}{\cos\theta_e}k(d, \theta_e)d \quad (\text{III.9})$$

Avec  $k(d, \theta_e) \geq 0$  continu tel que  $k\left(d, \mp\frac{\pi}{2}\right) = 0$  permet de linéariser (III.8) de sorte que la distance  $d$  du point  $P$  à la trajectoire vérifie :

$$\dot{d} = vak(d, \theta_e)d \quad (\text{III.10})$$

Si  $a$ ,  $v$  et  $k(d, \theta_e)$  sont strictement positifs,  $|d|$  est décroissante le long de toute trajectoire. Par définition de  $k(d, \theta_e)$ , ceci implique que l'erreur angulaire d'orientation doit rester strictement dans l'intervalle  $\left] -\frac{\pi}{2}, \frac{\pi}{2} \right[$ .

On montre que si  $v$  est de signe constant et suffisamment régulière,  $d$  tend asymptotiquement vers 0. Il reste donc à vérifier que si  $\theta_e$  est initialement dans l'intervalle  $\left] -\frac{\pi}{2}, \frac{\pi}{2} \right[$  il y reste. Dans le cas où l'erreur d'orientation initiale n'est pas dans l'intervalle en question le robot aura à faire une manœuvre lui permettant de s'y replacer. Avec la loi (III.9), (III.7) s'écrit :

$$\dot{\theta}_e = v \left( -\frac{c(s)\cos\theta_e}{1-dc(s)} - \left( 1 + \frac{ac(s)\sin\theta_e}{1-dc(s)} \right) \left( \frac{\tan\theta_e}{a} + \frac{\text{signe}(v)k(d, \theta_e)d}{\cos\theta_e} \right) \right)$$

Donc lorsque  $\theta_e$  tend vers  $-\frac{\pi}{2}$  par valeurs supérieures,  $\dot{\theta}_e$  est du signe de :

$$\left( 1 + \frac{ac(s)}{1-dc(s)} \right) \frac{v}{a}$$

Et donc,  $\dot{\theta}_e > 0$  si :

$$\frac{ac(s)}{1-dc(s)} < 1$$

De la même manière on montre que lorsque  $\theta_e$  tend vers  $\frac{\pi}{2}$  par valeurs inférieures  $\dot{\theta}_e < 0$  si :

$$\frac{ac(s)}{1-dc(s)} > -1$$

Si bien que finalement  $\theta_e$  reste dans l'intervalle  $\left] -\frac{\pi}{2}, \frac{\pi}{2} \right[$  si :

$$\left| \frac{ac(s)}{1-dc(s)} \right| < 1$$

Le long de la trajectoire.

**III.2.2.5. Suivi de chemin avec contrôle d'orientation**

L'objectif cette fois est de déterminer une loi  $\omega$  qui permette maintenant de stabiliser à la fois la distance à la trajectoire de référence et l'erreur en orientation durant le suivi.

On considère pour simplifier (et parce que cela suffit par la suite) que  $a = b = 0$ . On pose  $x_1 = s$ , et  $u_1 = \dot{s}$ , si bien que.  $\dot{x}_1 = u_1$ . Alors (III.5) s'écrit :

$$u_1 = \frac{v \cos \theta_e}{1 - dc(s)}$$

Et donc (III.6) devient :

$$\begin{aligned} \dot{d} &= v \sin \theta_e \\ &= u_1 (1 - dc(s)) \tan \theta_e \end{aligned}$$

En posant  $x_2 = d$ ,  $x_3 = (1 - dc(s)) \tan \theta_e$  et  $u_2 = \dot{x}_3$ , il vient finalement la forme chaînée :

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_1 x_3 \\ \dot{x}_3 &= u_2 \end{aligned}$$

**Note explicative**

Soit un système représenté par son état  $x = (x_1 \ x_2 \ x_3 \ \dots \ x_n)^T$  et ayant pour commande  $u = (u_1 \ u_2)^T$ . La représentation d'état du système est dite sous forme chaînée si elle s'écrit :

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_1 x_3 \\ &\dots \\ \dot{x}_{n-1} &= u_1 x_n \\ \dot{x}_n &= u_2 \end{aligned}$$

L'intérêt de cette représentation est qu'il existe des techniques de commande non-linéaires adaptées à cette classe de système.

Le problème de commande initial qui consistait à rechercher un retour d'état  $\omega$  à  $v$  donné est transformé par ce changement de variables. On recherche maintenant un retour d'état  $u_2$ , avec  $u_1$  donné déduit de  $v$ . L'obtention de  $u_2$  permet ensuite de recalculer  $\omega$ .

Si l'on choisit le retour d'état proportionnel :

$$u_2 = -u_1 k_2 x_2 - |u_1| k_3 x_3$$

Avec  $k_2$  et  $k_3$  deux constantes strictement positives, alors :

$$\dot{x}_3 = -u_1 k_2 x_2 - |u_1| k_3 x_3$$

Et donc,

$$\ddot{x}_3 + |u_1| k_3 \dot{x}_3 + u_1^2 k_2 x_3 = 0$$

Ce qui signifie que pour  $u_1$  constant quelconque  $x_3$  et donc  $x_2$  sont stables, le point de stabilité étant l'origine  $x_2 = x_3 = 0$ . Le système de départ est donc stable en  $d = 0$  et  $\theta_e = 0$ , ce qui résout notre problème.

On peut montrer selon [26] que le système est stable pour  $u_1$  quelconque. En effet, pour des conditions initiales sur  $d$  et  $\theta_e$  vérifiant :

$$x_2^2(0) + \frac{x_3^2}{k_2} < \frac{1}{c_{max}^2}$$

Où  $c_{max}$  est la courbure maximale le long de  $C$ , la condition sur la distance  $|dc(s)| < 1$  est vérifiée et la fonction :

$$V(x) = \frac{1}{2} \left( x_2^2 + \frac{x_3^2}{k_2} \right)$$

tend vers zéro si  $u_1$  est suffisamment régulière (bornée, dérivable et de dérivée bornée et ne tendant pas vers zéro lorsque  $t$  tend vers l'infini).

III.2.3. Stabilisation de mouvement

III.2.3.1. Problématique

Ce problème consiste à asservir la posture d'un robot mobile par rapport à un robot virtuel de référence.

III.2.3.2. Paramétrage

Le paramétrage du problème est donné à la Figure III.2.

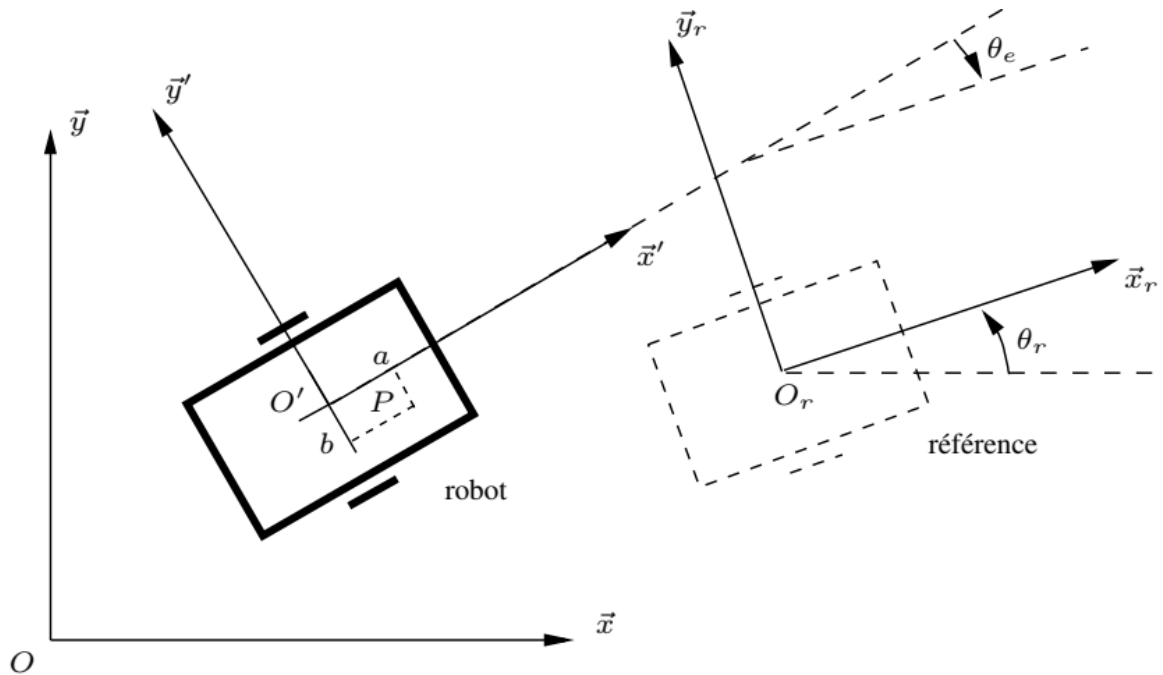


Figure III.2. Paramétrage pour la stabilisation de mouvement.

On considère dans la suite que le mouvement que décrit le robot de référence est admissible. Pour un robot dont le mouvement est décrit par la représentation d'état  $\dot{\mathbf{x}} = B(\mathbf{x})\mathbf{u}$ , un mouvement admissible est une loi  $\mathbf{x}_r$  sur un intervalle de temps  $[0, T]$  telle que  $\dot{\mathbf{x}}_r = B(\mathbf{x}_r)\mathbf{u}_r$ .

Le problème de stabilisation consiste alors à trouver la commande par retour d'état  $\mathbf{u}(\mathbf{x}, \mathbf{x}_r, t)$  telle que l'origine du système d'erreur :

$$\dot{\mathbf{x}}_e = B(\mathbf{x})\mathbf{u} - B(\mathbf{x}_r)\mathbf{u}_r$$

soit asymptotiquement stable.

### III.2.3.3. Stabilisation sans contrôle d'orientation

L'objectif est de déterminer une loi  $(\omega, v)$  qui permette de stabiliser l'erreur entre la position courante et la position de référence sans contrainte particulière sur l'erreur en orientation.

On choisit pour cela un point  $P$  situé sur l'axe longitudinal du robot, c'est-à-dire tel que  $b = 0$ . On peut calculer la dérivée de l'écart entre les positions de  $P$  et  $O_r$  dans  $R$ .

Soit  $\mathbf{p}_e$  l'écart de position et  $\mathbf{p}_r$  la position de  $O_r$  dans  $R$  :

$$\begin{aligned}\dot{\mathbf{p}}_e &= \overrightarrow{v_{P/R}} - \dot{\mathbf{p}}_r \\ &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} v - b\omega \\ a\omega \end{pmatrix} - \dot{\mathbf{p}}_r\end{aligned}$$

Par analogie avec (III.3). En posant le changement de variables :

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & -b \\ 0 & 0 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

Inversible car par hypothèse  $a \neq 0$ , on obtient la représentation :

$$\dot{\mathbf{p}}_e = \mathbf{u} - \dot{\mathbf{p}}_r$$

Il est alors facile de stabiliser ce système par un simple retour d'état :

$$\mathbf{u} = \dot{\mathbf{p}}_r - \mathbf{K}\dot{\mathbf{p}}_e$$

Avec  $\mathbf{K}$  définie positive.

Dans ce qui précède c'est le point  $P$  qui est asservi à la position de  $O_r$  et il persiste une erreur constante de  $a \neq 0$ . On va voir par la suite comment asservir l'origine du repère associée au robot à celle du robot virtuel de référence.

**III.2.3.4. Stabilisation de mouvements admissibles avec contrôle d'orientation**

L'objectif est de déterminer une loi  $(\omega, v)$  qui permette de stabiliser l'erreur entre la posture courante et la posture de référence.

On exprime l'erreur de posture  $\xi_e$  dans le repère  $R_r$  du robot de référence :

$$\xi_e = \begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} = \begin{pmatrix} \cos\theta_r & \sin\theta_r & 0 \\ -\sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x - x_r \\ x - y_r \\ \theta - \theta_r \end{pmatrix} \quad (III.11)$$

On montre en dérivant la relation (III.11) que :

$$\begin{pmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{pmatrix} = \begin{pmatrix} \cos\theta_r \\ \sin\theta_r \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega - \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} v_r - \begin{pmatrix} -y_e \\ x_e \\ 1 \end{pmatrix} \omega_r$$

Si l'on pose le changement de variables :

$$\begin{aligned} x_1 &= x_e \\ x_2 &= y_e \\ x_3 &= \tan\theta_e \\ u_1 &= v\cos\theta_e - v_r \\ u_2 &= \frac{\omega - \omega_r}{\cos^2\theta_e} \end{aligned}$$

Défini partout sauf en  $\theta_e = \pm \frac{\pi}{2}$ , la nouvelle représentation d'état du système s'écrit :

$$\begin{aligned} \dot{x}_1 &= u_1 + \omega_r x_2 \\ \dot{x}_2 &= u_1 x_3 - \omega_r x_1 + v_r x_3 \\ \dot{x}_3 &= u_2 \end{aligned}$$

On reconnaît une forme chaînée si  $v_r = \omega_r$ . Alors, la loi de commande par retour d'état :

$$u_1 = -k_1|v_r|(x_1 + x_2x_3)$$

$$u_2 = -k_2v_rx_2 - k_3|v_r|x_3$$

Avec  $k_1$ ,  $k_2$  et  $k_3$  trois constantes strictement positives rend le système asymptotiquement stable si  $v_r$  est suffisamment régulière (bornée, dérivable et de dérivée bornée et ne tendant pas vers zéro lorsque  $t$  tend vers l'infini) [26].

Une version linéarisée autour de zéro (le terme  $x_2x_3$  disparaît alors) de cette commande stabilise également l'origine de l'état.

### III.3. Conclusion

Le problème de suivi d'une trajectoire de référence pour un robot mobile non-holonome au sens général a été présenté dans chapitre. En effet, le sujet suivi d'une trajectoire pour un robot mobile à travers une loi de commande synthétisée est considéré très important vu l'importance de l'utilisation des robots mobiles dans des différentes tâches relatives aux activités spécifiques dans notre vie.

Dans ce chapitre, nous avons présenté le problème en question, en concentrant sur la présence de deux scénarios (i) *suivi d'une trajectoire avec contrôle/sans contrôle d'orientation* et (ii) *suivi d'une trajectoire stabilisation/sans stabilisation*.

Dans le chapitre suivant, nous serons plus pointés dans notre présentation. Nous nous parlerons sur le problème de suivi d'une trajectoire de référence pour un robot mobile non-holonome en utilisant la technique de commande dite en anglais '*Input-Output State Feedback linearization* (I-O SFL)' [27].

Le chapitre suivant présente les résultats de simulations lors de l'application de la technique I-O SFL [27].

## Chapitre IV :

# Suivi de trajectoire pour un robot mobile en utilisant la technique I-O SFL (Synthèse et simulations)

### IV.1. Introduction

Comme nous l'avons expliqué au chapitre précédent, un chemin (pour un robot mobile) est une « ligne » qui doit être parcouru par ce dernier dans un espace à des coordonnées généralisées. Plus précisément, si un chemin est défini en fonction du temps, c'est-à-dire, le mouvement tout le long du chemin est synchronisé avec le temps, dans ce cas ce chemin est dit trajectoire. Chaque fois que le planning de mouvement d'un robot mobile est connu préalablement, la trajectoire (de référence) de ce dernier peut être écrite comme une fonction temporelle  $\xi_{ref}(t) = [x_{ref}(t), y_{ref}(t), \theta_{ref}(t)]^T$ .

Pour des raisons pratiques, la trajectoire est toujours définie sur un intervalle de temps fini  $t \in [0, T]$ , c'est-à-dire que la trajectoire de référence à un point de départ et un autre de fin. Une commande de suivi d'une trajectoire donnée est un mécanisme qui assure une proche (que possible) de  $\xi(t)$  au  $\xi_{ref}(t)$  quelques soit les contraintes rencontrées.

Dans ce chapitre, nous parlons au sens général sur l'approche de commande basée sur l'aspect de linéarisation de retour d'état (*state Feedback Linearization*) suivi par un exemple de simulation illustratif qui permet de comprendre cette dernière (approche) au profond. Par la suite, nous synthétisons une autre loi de commande dite en anglais '*Input-Output State Feedback linearization* (I-O SFL)' [27]. Une validation par simulation relative à cette technique est aussi présentée. Les résultats de simulation obtenus, sont aussi illustrés.

### IV.2. Technique de linéarisation de retour d'état (*State Feedback Linearization*)

La linéarisation de retour d'état (*State Feedback Linearization*) est une technique de commande des systèmes non linéaires très puissante. Ainsi, l'idée principale de cette dernière est de faire transformer algébriquement la dynamique d'un système non linéaire donné en un système entièrement ou partiellement linéarisé afin que l'ensemble de méthode de commande

## Chapitre IV : Suivi de trajectoire pour un robot mobile en utilisant la technique I-O SFL (Synthèse et simulations)

par retour d'état puissent être appliqué [2]. Il est à noter que, cette technique, qui est une transformation exacte d'un état (*State*) et un retour (*Feedback*), est entièrement différente de la linéarisation conventionnelle basée sur des approximations de la série de Taylor.

Selon [2], les étapes qui doivent être suivies lors de la conception d'une linéarisation de retour d'état pour intérêt de commande sont les suivantes :

- Les sorties appropriées (dites ; flat outputs) doivent être choisies. Leur nombre est le même que le celui des d'entrées de système,
- Les sorties (dites ; flat outputs) doivent être dérivées ainsi leurs dérivés doivent vérifiées la présence fonctionnelle des entrées. Cette étape est répétée jusqu'à toutes les entrées (ou leurs dérivées) apparaissent dans les dérivées des sorties (dites ; flat outputs). De plus, si toutes les entrées (plus précisément leurs dérivées les plus élevées) peuvent être dérivées à partir de ce système d'équations, nous pouvons passer à l'étape suivante.
- Le système d'équations est résolu pour les dérivées les plus élevées des entrées individuelles. Pour obtenir les entrées réelles du système, une chaîne d'intégrateurs (*integrators*) doit être utilisé sur les dérivés d'entrée. D'autre part, les dérivés de sortie, servir comme de nouvelles entrées d'système.

On considère le cas d'un robot mobile à roues a le modèle exprimé dans le section II.4.2.4, les sorties (dites ; flat outputs) sont respectivement  $x(t)$  et  $y(t)$ . Ainsi, les dérivées premières de ces dernières sont données comme suit :

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

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

Dans les premières dérivées, seule la vitesse de translation  $v$  est apparaît ainsi la différenciation peut être contenue :

$$\ddot{x} = \dot{v} \cos \theta - v \dot{\theta} \sin \theta$$

$$\ddot{y} = \dot{v} \sin \theta + v \dot{\theta} \cos \theta$$

## Chapitre IV : Suivi de trajectoire pour un robot mobile en utilisant la technique I-O SFL (Synthèse et simulations)

Il est clair que, dans les expressions des dérivées secondes de  $x$  et  $y$ , les deux vitesses ( $v$  et  $\omega = \dot{\theta}$ ) sont apparues. Maintenant, le système d'équations est réécrit pour que les dérivées secondes des sorties sont décrites comme des fonctions des dérivées les plus élevées des entrées individuelles ( $\dot{v}$  et  $\omega$  dans ce cas) :

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -v\sin\theta \\ \sin\theta & v\cos\theta \end{bmatrix} \begin{bmatrix} \dot{v} \\ \omega \end{bmatrix} = F \begin{bmatrix} \dot{v} \\ \omega \end{bmatrix} \quad (\text{IV.1})$$

La matrice  $F$  dans (IV.1), qui est non singulière si  $v \neq 0$ . Ainsi, le système d'équations précédent peut donc être résolu pour en cherchant  $\dot{v}$  et  $\omega$ :

On obtient ainsi :

$$\begin{bmatrix} \dot{v} \\ \omega \end{bmatrix} = F^{-1} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\frac{\sin\theta}{v} & \frac{\cos\theta}{v} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \quad (\text{IV.2})$$

En effet, la grandeur  $\omega$  est la solution de l'équation (IV.2) qui est en réalité l'entrée réelle du robot, tandis que la solution  $\dot{v}$  doit être intégrée avant d'être pouvoir utilisée comme entrée.

De plus, le système linéaire obtenu a les entrées suivantes  $[u_1, u_2]^T = [\ddot{x}, \ddot{y}]^T$ , et comme états  $z = [x, y, \dot{x}, \dot{y}]^T$  (notant que le modèle cinématique exprimé dans le section II.4.2.4 a trois états ; le quatrième est dû à une intégration supplémentaire).

De ce fait, La dynamique du nouveau système peut être décrit aisément par la représentation dans l'espace d'état suivante :

$$\begin{bmatrix} \dot{x} \\ \dot{x} \\ \dot{y} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (\text{IV.3})$$

Ou sous la forme compacte suivante :

$$\dot{z} = Az + Bu \quad (\text{IV.4})$$

## Chapitre IV : Suivi de trajectoire pour un robot mobile en utilisant la technique I-O SFL (Synthèse et simulations)

Le système (IV.4) est contrôlable puisque sa matrice de contrôlabilité donnée l'expression (IV.5) a un rang plein, chose qui implique que et donc le contrôleur d'état existe pour un polynôme caractéristique choisi arbitrairement dans la boucle fermée.

$$Q_C = [B, AB] \quad (IV.5)$$

Aussi, une exigence supplémentaire est de concevoir une loi de commande pour que le robot suive une trajectoire de référence. Dans le cadre des systèmes plats (flat systems), une trajectoire de référence est donnée pour les sorties (dites ; flat outputs), dans ce cas, ils sont respectivement  $x_{\text{ref}}(t)$  et  $y_{\text{ref}}(t)$ .

La référence peut alors être facilement obtenue pour l'état du système  $z_{\text{ref}}(t) = [x_{\text{ref}}, \dot{x}_{\text{ref}}, y_{\text{ref}}, \dot{y}_{\text{ref}}]^T$  ainsi l'entrée du système  $u_{\text{ref}}(t) = [\ddot{x}_{\text{ref}}, \ddot{y}_{\text{ref}}]$ .

L'équation (IV.4) peut également être écrite pour les signaux de référence :

$$\dot{z}_{\text{ref}} = Az_{\text{ref}} + Bu_{\text{ref}} \quad (IV.6)$$

L'erreur entre les états réels et les états de référence est définie comme par  $\tilde{z} = z - z_{\text{ref}}$ . Soustraire l'équation (IV.6) de (IV.4), nous donne :

$$\dot{\tilde{z}} = A\tilde{z} - B(u - u_{\text{ref}}) \quad (IV.7)$$

Il est à noter que cette dernière équation (IV.7) décrit la dynamique de l'erreur d'état. Cette dynamique doit être stable et suffisamment rapide. Une façon de prescrire la dynamique en boucle fermée est d'imposer des pôles spécifiques en boucle fermée.

Nous avons déjà montré que la paire  $(A, B)$  est contrôlable, chose qui nous permet de faire la sélection correcte d'une matrice de gain de contrôle  $K$  (de dimension  $2 \times 4$ ), des emplacements arbitraires des pôles en boucle fermée dans le demi-plan gauche du plan complexe peut être réalisé.

Par conséquent, (IV.7) devient :

$$\ddot{\tilde{z}} = (A - BK)\tilde{z} - BK\dot{\tilde{z}} + B(u - u_{\text{ref}})$$

$$\ddot{\tilde{z}} = (A - BK)\tilde{z} + B(K\tilde{z} + \dot{u} - \dot{u}_{\text{ref}}) \quad (\text{IV.8})$$

Si le dernier terme de l'équation (IV.8) est nul, les erreurs d'état convergent vers 0 avec la dynamique prescrite, donnée par la matrice du système en boucle fermée  $(A - BK)$ .

En effet, forcer ce terme à 0 définit la loi de commande relative à cette approche :

$$u(t) = K(z_{\text{ref}}(t) - z(t)) + \dot{u}_{\text{ref}}(t) \quad (\text{IV.9})$$

Une représentation sous forme d'un diagramme fonctionnel du système de commande conçu est illustrée sur la Figure IV.1.

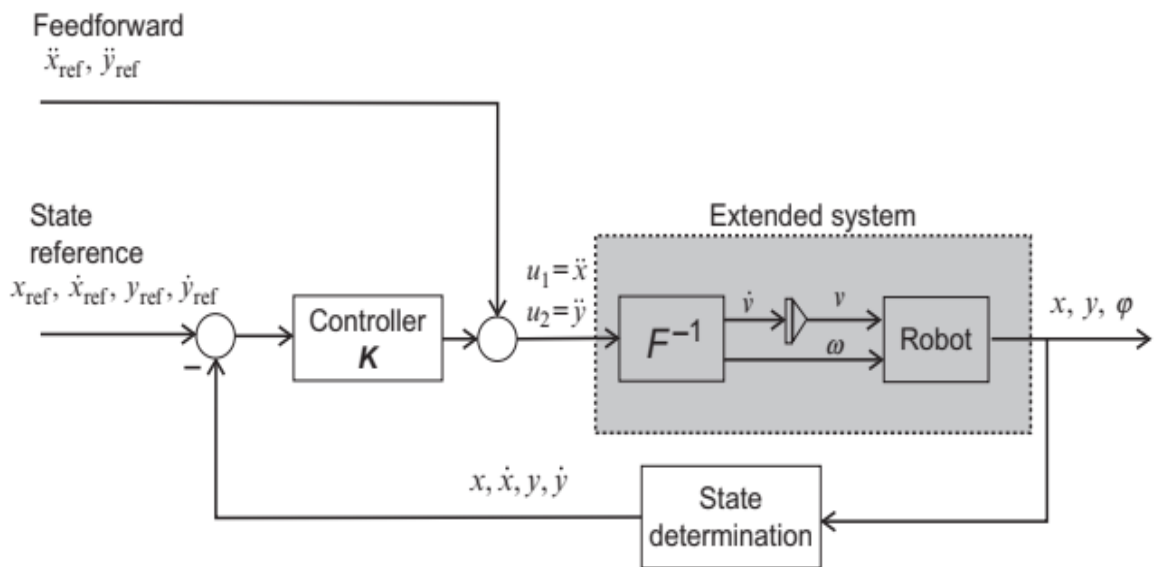


Figure IV.1. Commande pour suivi d'une trajectoire à base d'une linéarisation de retour d'état [2].

## Chapitre IV : Suivi de trajectoire pour un robot mobile en utilisant la technique I-O SFL (Synthèse et simulations)

En raison d'une forme spécifique des deux matrices  $A$  et  $B$  dans (IV.3), d'où  $u_1$  a uniquement une influence sur les deux états  $z_1$  et  $z_2$  tandis que  $u_1$  n'influence seulement que sur les deux états  $z_3$  et  $z_4$ , de ce fait la matrice de gain de contrôle prend la forme spéciale suivante :

$$K = \begin{bmatrix} k_1 & k_2 & 0 & 0 \\ 0 & 0 & k_3 & k_4 \end{bmatrix} \quad (\text{IV.10})$$

La loi de commande (IV.9) peut donc être complètement décomposée comme suit :

$$\begin{aligned} u_1(t) = \ddot{x}(t) &= k_1(x_{\text{ref}}(t) - x(t)) + k_2(\dot{x}_{\text{ref}}(t) - \dot{x}(t)) + \ddot{x}_{\text{ref}}(t) \\ u_2(t) = \ddot{y}(t) &= k_3(y_{\text{ref}}(t) - y(t)) + k_4(\dot{y}_{\text{ref}}(t) - \dot{y}(t)) + \ddot{y}_{\text{ref}}(t) \end{aligned} \quad (\text{IV.11})$$

Il est à noter que, l'approche de commande proposée nécessite la connaissance de tous les états. En effet, malgré que  $x$  et  $y$  sont généralement mesurés, leurs dérivés ne le sont pas.

De plus, La dérivée numérique conduit pratiquement à une amplification de bruit chose qui doit être évitée. A cet effet, Deux solutions possibles sont proposées :

- Les états non mesurés peuvent être estimés par des observateurs d'État.
- Si l'orientation du robot  $\theta$  est mesurée, les dérivées peuvent être calculées comme à travers les deux équations  $\dot{x} = v \cos \theta$  et  $\dot{y} = v \sin \theta$ .

L'exemple de simulation illustratif donné dans ce qui suit, nous permettre de prendre une idée très profonde sur cette approche.

### **Exemple de simulation illustratif [2] :**

Un robot mobil à conduite différentielle doit être commandé d'une façon qui permet de suivre la trajectoire de référence définie par  $x_{\text{ref}} = 1,1 + 0,7 \sin(\frac{2\pi t}{30})$  et  $y_{\text{ref}} = 0,9 + 0,7 \sin(\frac{4\pi t}{30})$ . Le pas de simulation est égal à  $T_s = 0,033$  s. La position initiale est  $[x(0), y(0), \theta(0)] = [1,1, 0,8, 0]$ .

Le détail de cet exemple est illustré dans script MATLAB (voir le code MATLAB **M-1** dans l'annexe).

## Chapitre IV : Suivi de trajectoire pour un robot mobile en utilisant la technique I-O SFL (Synthèse et simulations)

On Implémente l'algorithme de commande présenté en utilisant le Matlab, ainsi tous résultats obtenus sont illustrés dans les figures Figure IV.2 , Figure IV.3 et Figure IV.4.

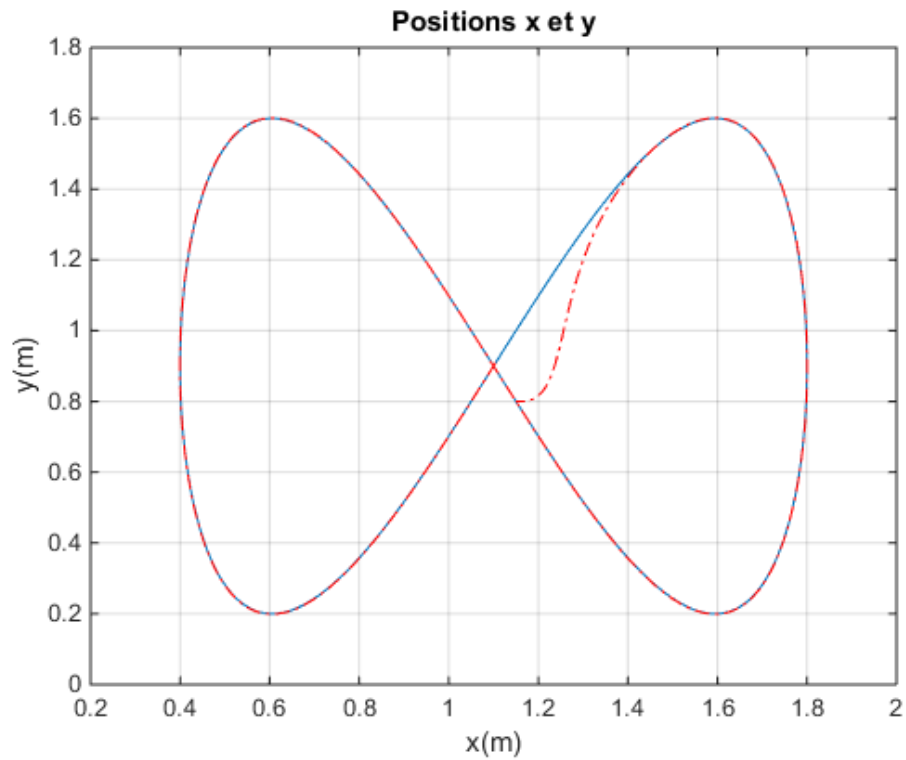


Figure IV.2. Positions  $x(t)$  et  $y(t)$  ( $x_{\text{ref}}(t)$  et  $y_{\text{ref}}(t)$  trajectoire de référence en bleu,  $x_{\text{réelle}}(t)$  et  $y_{\text{réelle}}(t)$  en rouge discontinu).

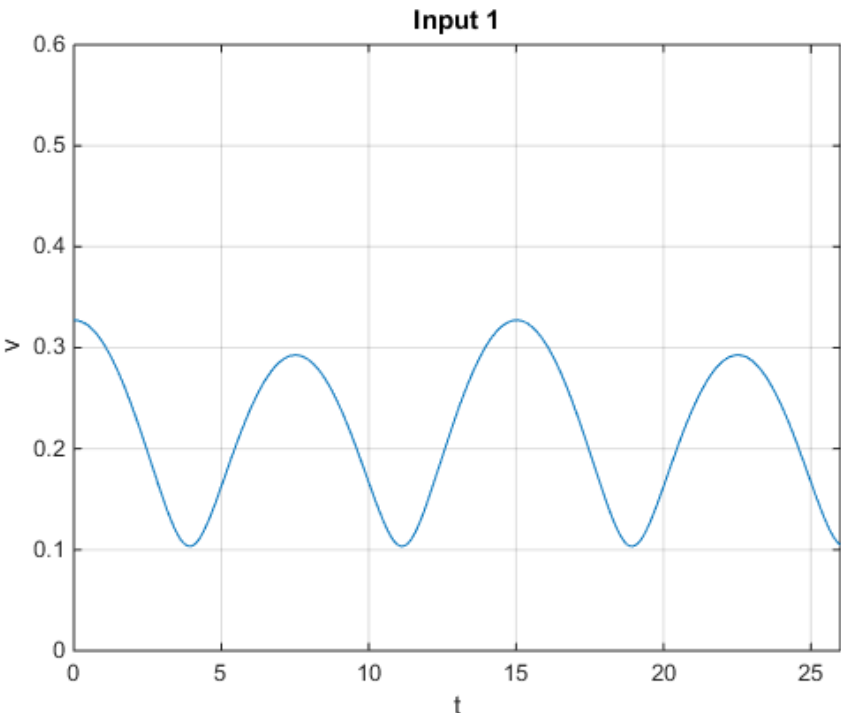


Figure IV.3. Sortie 1 du régulateur ( $v$ ).

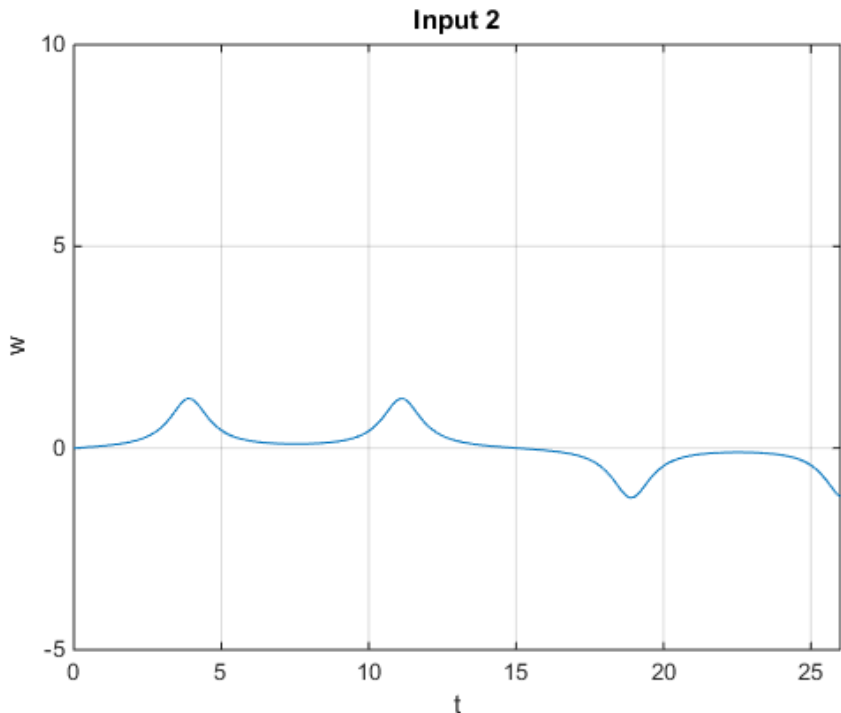


Figure IV.4. Sortie 2 du régulateur ( $\omega$ ).

### IV.3. Suivi de trajectoire pour un robot mobile en utilisant la technique I-O SFL

Dans ce qui suit, nous présentons le problème de suivi d'une trajectoire de référence pour un robot mobile non-holonyme en utilisant la technique de commande dite en anglais '*Input-Output State Feedback linearization (I-O SFL)*'. En effet, on travaille à développer une loi de commande basée sur cette approche.

On considère le cas du robot mobile de type unicycle représenté sur la figure Figure IV.5. En effet, Il est possible de contrôler le mouvement de ce robot avec une vitesse constante et linéaire quelle que soit la courbure de la trajectoire imposée [27].

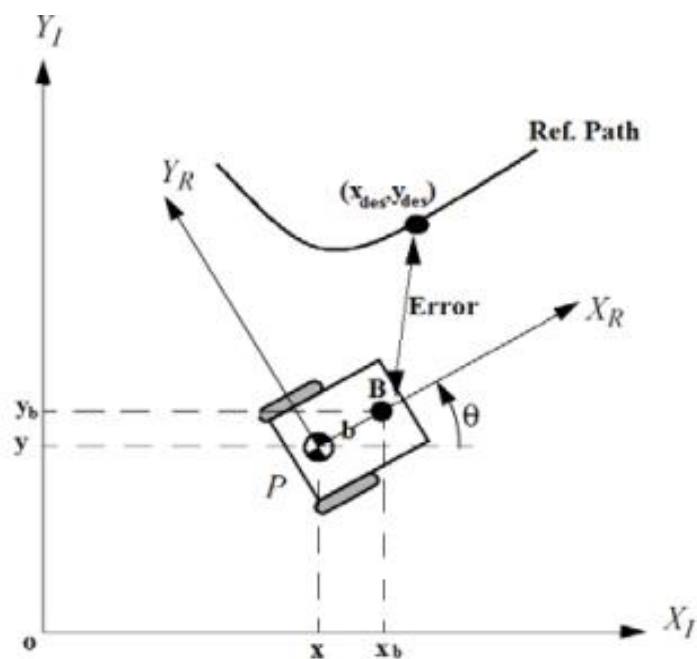


Figure IV.5. Description du problème de commande.

A partir de cette figure, nous avons :

$$\begin{aligned} x_B &= x + b \cos \theta \\ y_B &= y + b \sin \theta \end{aligned} \quad (IV.12)$$

Après dérivation, on obtient :

$$\begin{aligned} \dot{x}_B &= v \cos \theta + \omega b \sin \theta \\ \dot{y}_B &= v \sin \theta + \omega b \cos \theta \end{aligned} \quad (IV.13)$$

## Chapitre IV : Suivi de trajectoire pour un robot mobile en utilisant la technique I-O SFL (Synthèse et simulations)

Avec :  $\dot{\theta} = \omega$ .

On peut récrire les équations précédentes sous la forme suivante :

$$\begin{bmatrix} \dot{x}_B \\ \dot{y}_B \end{bmatrix} = \begin{bmatrix} \cos\theta & -b\sin\theta \\ -\sin\theta & b\cos\theta \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (\text{IV.14})$$

Ce qui implique que :

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & -b\sin\theta \\ -\sin\theta & b\cos\theta \end{bmatrix}^{-1} \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \end{bmatrix} \quad (\text{IV.15})$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\frac{1}{b}\sin\theta & \frac{1}{b}\cos\theta \end{bmatrix} \begin{bmatrix} \dot{x}_B \\ \dot{y}_B \end{bmatrix} \quad (\text{IV.15})$$

Étant donné une trajectoire désirée à suivre  $(x_{\text{dés}}(t), y_{\text{dés}}(t))$  comme le montre la Figure IV.5, il est possible de trouver les entrées  $\dot{x}_B$  et  $\dot{y}_B$  qui garantissent un suivi asymptotique, cela peut être vu comme suit :

$$\begin{aligned} \dot{x}_B &= \dot{x}_{\text{dés}}(t) + k e_x(t) \\ \dot{y}_B &= \dot{y}_{\text{dés}}(t) + k e_y(t) \end{aligned} \quad (\text{IV.16})$$

D'où :

$$\begin{aligned} e_x(t) &= (\dot{x}_{\text{dés}}(t) - \dot{x}_B) \\ e_y(t) &= (\dot{y}_{\text{dés}}(t) - \dot{y}_B) \end{aligned} \quad (\text{IV.17})$$

$$\begin{aligned} \dot{e}_x + k_1 e_x &= 0 \\ \dot{e}_y + k_2 e_y &= 0 \end{aligned} \quad (\text{IV.18})$$

La solution du système d'équation (IV.18), nous donne :

$$\begin{aligned} e_x(t) &= c_1 e^{-k_1 t} \\ e_y(t) &= c_2 e^{-k_2 t} \end{aligned} \quad (\text{IV.19})$$

Pour des  $k_1$  et  $k_2$  positifs,  $\lim_{t \rightarrow \infty} e_x(t) = 0$  et  $\lim_{t \rightarrow \infty} e_y(t) = 0$

Le système est alors asymptotiquement stable et l'erreur est toujours converge vers zéro.

#### IV.4. Simulation de la méthode I-O SFL

Nous simulons cette méthode, en utilisant le MATLAB. Ainsi, les paramètres relatifs à la loi de commande I-O SFL et le robot mobile sont illustrés dans le code MATLAB M-2 dans l'annexe.

Dans ce qui suit, nous illustrons les résultats de simulations obtenus lors de l'application de cette approche. En effet, la Figure IV.6 montre le comportement du robot (comparaison entre positions réelles  $(x_{réelle}(t), y_{réelle}(t))$  et positions désirées  $(x_{dés}(t), y_{dés}(t))$ ).

On observe clairement, que cette loi de commande à permettre au robot mobile de suivre la trajectoire imposée avec une précision remarquable comme les montrent les deux figures Figure IV.7 et Figure IV.8 qui représentent l'évolutions des deux erreurs de poursuite  $e_x(t)$  et  $e_y(t)$ .

Les signaux de commandes générés  $v(t) = u_1(t)$  et  $\omega(t) = u_2(t)$ , sont dans les normes souhaitées, ils sont illustrés dans les Figure IV.9 et Figure IV.10.

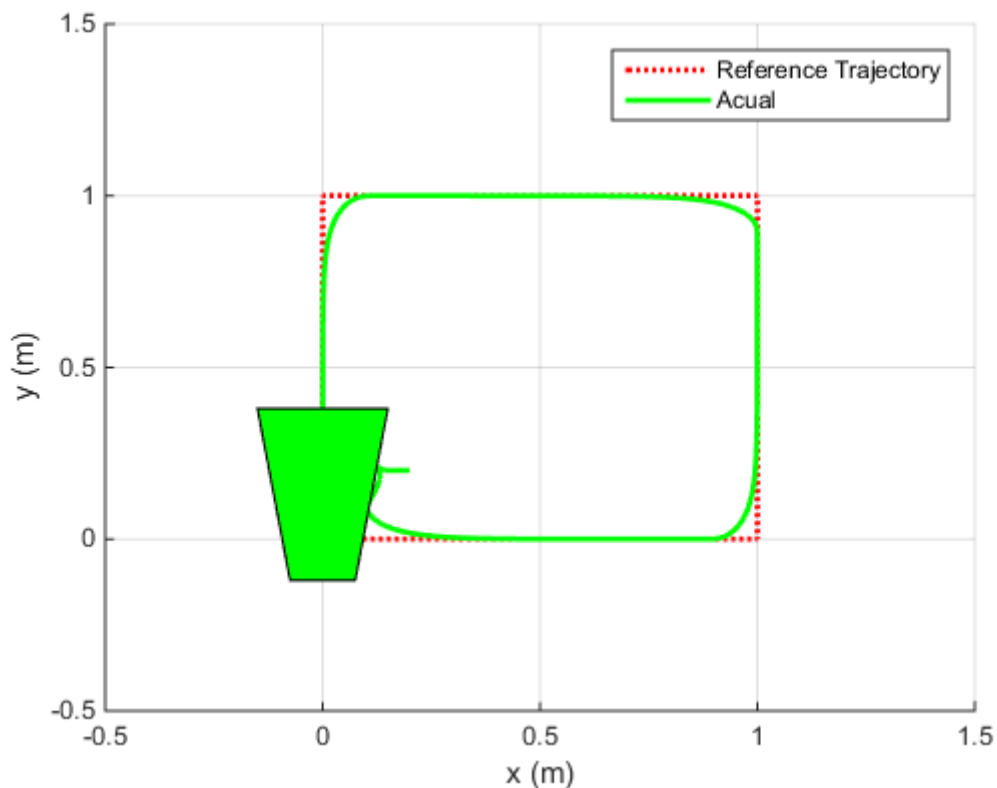


Figure IV.6. Positions  $x(t)$  et  $y(t)$  ( $x_{dés}(t)$  et  $y_{dés}(t)$  trajectoire de référence en rouge discontinu,  $x_{réelle}(t)$  et  $y_{réelle}(t)$  en vert).

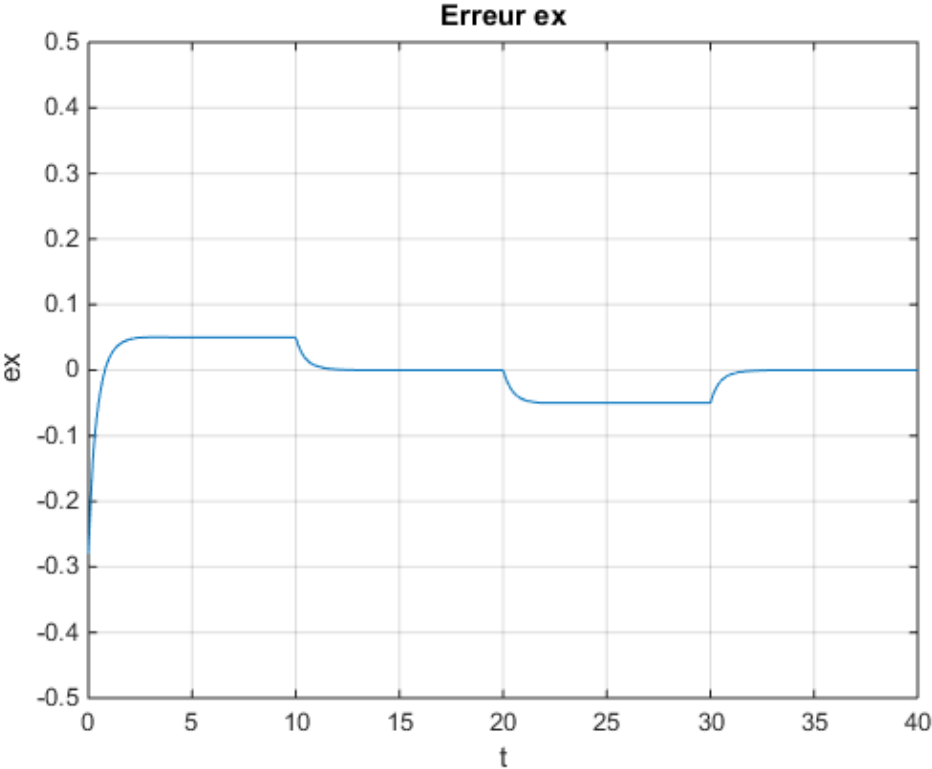


Figure IV.7. Erreur de poursuit  $e_x(t)$ .

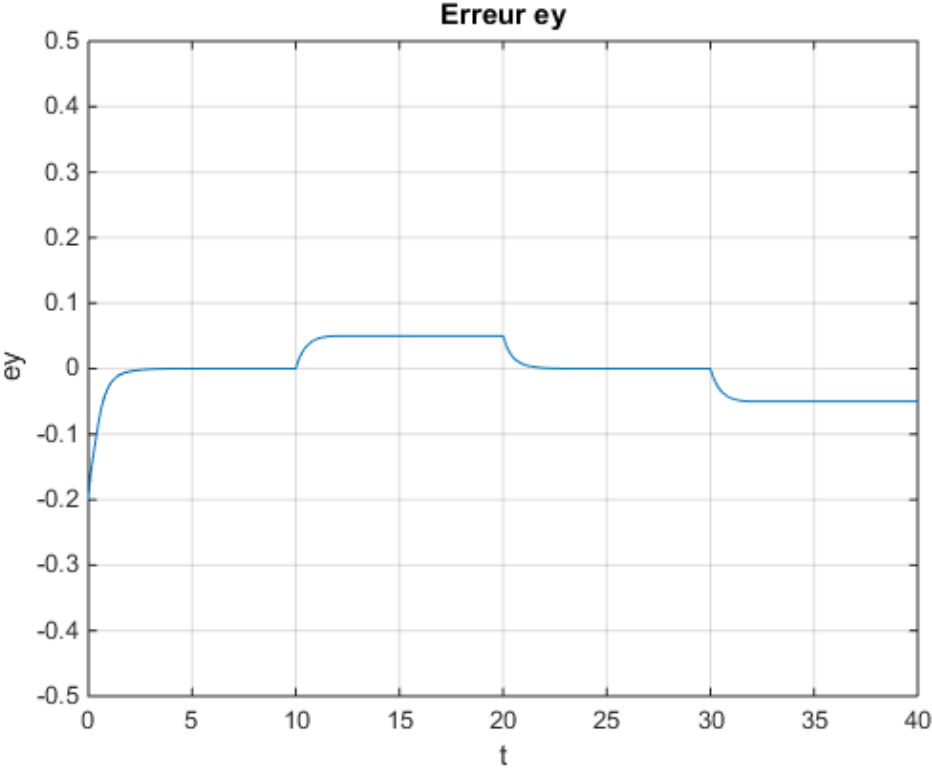


Figure IV.8. Erreur de poursuit  $e_y(t)$ .

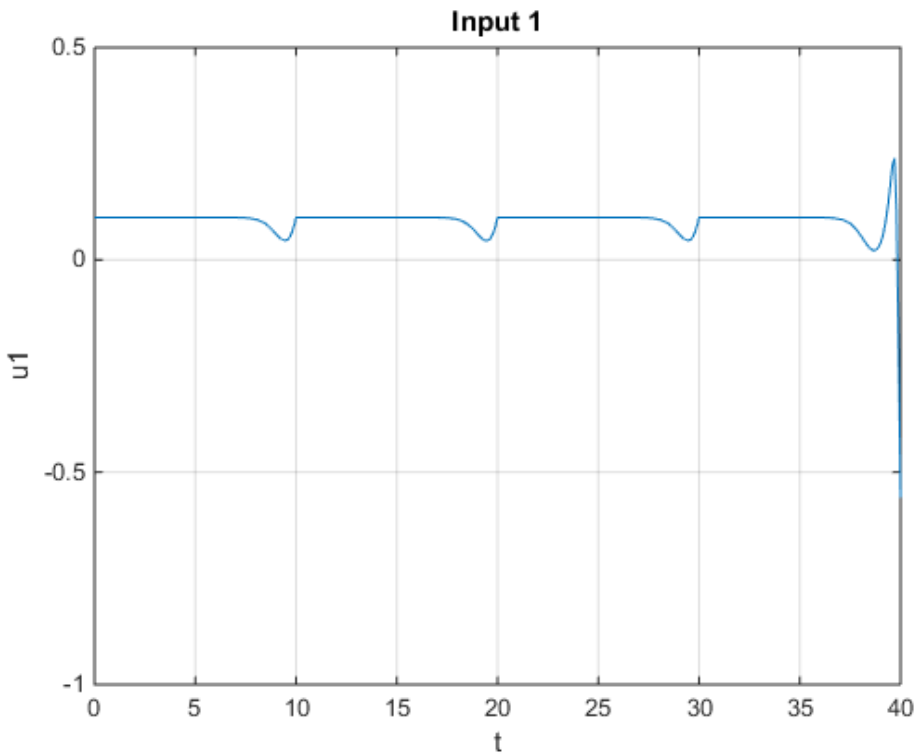


Figure IV.9. Sortie 1 du régulateur ( $v$ ) m/s.

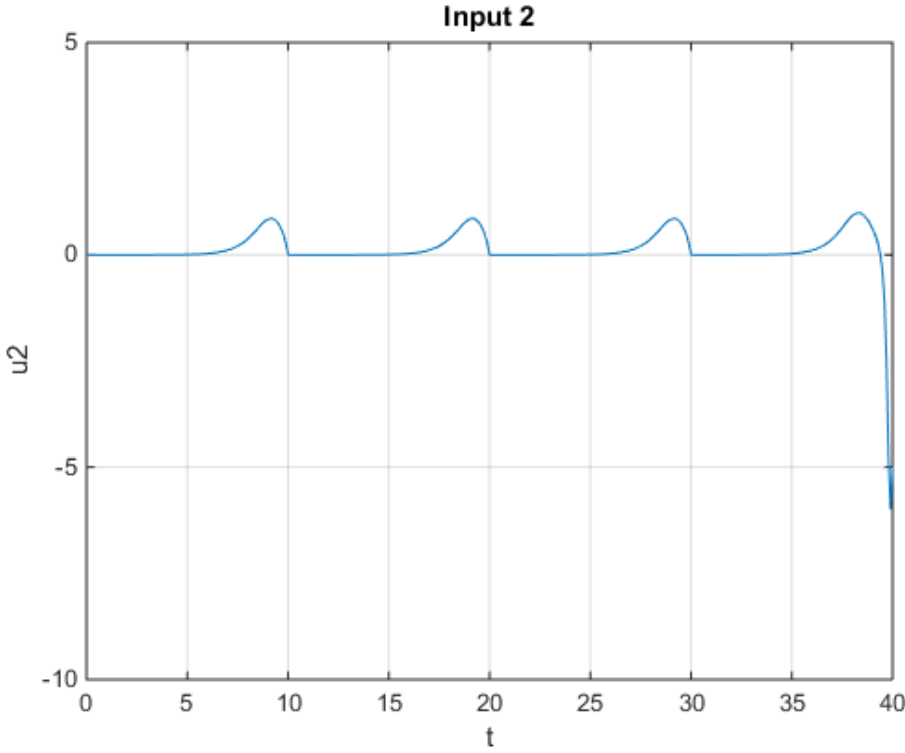


Figure IV.10. Sortie 2 du régulateur ( $\omega$ ) rad/s.

#### IV.5. Conclusion

Dans ce chapitre, nous avons parlé sur le problème de suivi d'une trajectoire de référence pour un robot mobile non-holonome en utilisant dans un premier temps (au sens général) une commande basée sur l'aspect de linéarisation de retour d'état (*state Feedback Linearization*) suivie par un exemple de simulation illustratif qui a nous permettre de prendre une idée claire sur cette dernière (approche).

Par la suite, nous avons synthétisé une autre loi de commande dite en anglais '*Input-Output State Feedback linearization (I-O SFL)*'. Une validation par simulation relative à cette dernière a été présentée. Les résultats de simulation obtenus, sont aussi illustrés.

Il est clair que, ce genre de techniques de commande permettent d'atteindre quelques objectives envisagées relatives au problème de poursuite d'une trajectoire.

Une autre remarque très importante doit être signalé, il s'agit de la simplicité très remarquable de ce type de commande. Néanmoins, l'existence du problème du choix des gains  $K$  accosés à la loi de commande.

## Conclusion générale

Le travail présenté dans ce mémoire est porté sur le suivi de trajectoire d'un robot mobile non holonome de type unicycle. Après la présentation d'un simple état de l'art sur les robots mobiles non holonomes, il nous est apparu puisque nous sommes débutants dans ce domaine d'étudier une approche de commande simple basée au sens général sur l'aspect de la linéarisation de retour d'état (*State Feedback Linearization*). Nous avons également étendu notre intérêt à une approche dite en anglais '*Input-Output State Feedback linearization (I-O SFL)*'.

Un des avantages principaux de cette méthode par rapport à la plupart des approches est qu'il est systématique et relativement simple lors de sa synthèse.

Afin de pouvoir examiner l'adaptation de cette approche au problème de suivi de trajectoire d'un robot mobile, une simulation sur MATLAB est réalisée. En effet, nous avons obtenu de bons résultats dans le suivi d'une trajectoire carrée. Même, l'exemple illustratif à montrer l'efficacité de cette technique (*State Feedback Linearization*) surtout au sens simplicité, synthèse et précision.

Le sujet traité dans ce mémoire, même concernant la technique de commande présentée est resté très ouvert. De plus, la matière scientifique présentée dans ce manuscrit, malgré sa simplicité mais elle permet aux lecteurs (étudiants) de prendre une idée acceptable sur la commande de ce type de robots.

# Annexe

Le code MATLAB **M-1** (Simulation de l'exemple illustratif)

```

clc
clear all
close all

Ts = 0.033; % Sampling time
t = 0:Ts:30; % Simulation time
L_velocity=[];
A_velocity=[];

% Reference
freq = 2*pi/30;
xRef = 1.1 + 0.7*sin(freq*t); yRef = 0.9 + 0.7*sin(2*freq*t);
dxRef = freq*0.7*cos(freq*t); dyRef = 2*freq*0.7*cos(2*freq*t);
ddxRef = -freq^2*0.7*sin(freq*t); ddyRef = -4*freq^2*0.7*sin(2*freq*t);
qRef = [xRef; yRef; atan2(dyRef, dxRef)];
uRef = [ddxRef; ddyRef];

q = [xRef(1)+.05; yRef(1)-0.1; 0]; % Initial robot pose
z1 = [q(1); dxRef(1)]; % Initial state [x, x']
z2 = [q(2); dyRef(1)]; % Initial state [y, y']
v = sqrt(z1(2)^2+z2(2)^2); % Initial state of velocity integrator

x_q=[q(1)];
y_q=[q(2)];

% Matrices of linearized system
A = [0, 1; 0, 0]; B = [0; 1]; C = [1, 0];
% State feedback controller
desPoles = [-2-1i; -2+1i]; % Desired poles (of the controller)
K = acker(A, B, desPoles); % Control gains obtained by pole placement

for k = 1:length(t)
% Reference states
zRef1 = [xRef(k); dxRef(k)];
zRef2 = [yRef(k); dyRef(k)];

% Error and control
ez1 = zRef1 - z1;
ez2 = zRef2 - z2;
uu = [ddxRef(k); ddyRef(k)] + [K*ez1; K*ez2];

% Compute inputs to the robot
F = [cos(q(3)), -v*sin(q(3)); ...
sin(q(3)), v*cos(q(3))];
vv = F\uu; % Translational acceleration and angular velocity
v = v + Ts*vv(1); % Integrate translational acceleration
u = [v; vv(2)]; % Robot input

L_velocity=[v L_velocity];

```

```
A_velocity=[vv(2) A_velocity];

% Robot motion simulation
dq = [u(1)*cos(q(3)); u(1)*sin(q(3)); u(2)];
noise = 0.00; % Set to experiment with noise (e.g. 0.001)
q = q + Ts*dq + randn(3,1)*noise; % Euler integration
q(3) = wrapToPi(q(3)); % Map orientation angle to [-pi, pi]

x_q=[q(1) x_q];
y_q=[q(2) y_q];

% Take known (measured) orientation and velocity to compute states
z1 = [q(1); u(1)*cos(q(3))];
z2 = [q(2); u(1)*sin(q(3))];
end

figure (1)
plot(t,L_velocity)
axis([0 26 0 0.6]);
grid

title('Input 1')
xlabel('t')
ylabel('v')

figure (2)
plot(t,A_velocity)
axis([0 26 -5 10]);
grid

title('Input 2')
xlabel('t')
ylabel('w')

figure (3)
plot(xRef,yRef)
hold on
plot(x_q,y_q,'-. r')
grid

title('Positions x et y')
xlabel('x(m)')
ylabel('y(m)')
```

## Le code MATLAB M-1 (Simulation de la méthode I-O SFL)

```

clear all
clc
close all
%%
global b

b = 0.18;
his.dt=[];

% IO_SFL
dt=0.1;
T=360;
k=2;
bref=0.08;
x=0.2;

%%
y=0.2;
theta=0/2;
X=[x;y;theta];

Vref=0.0;      % max 0.5 m/s
Omegaref=0.0; % max 3 rps

u_1_Vref=[];
u_2_Omegaref=[];

% Reference Trajectory
tperiod = 2*[0 5 10 15 20 ];

points =[0 0 ;1 0 ; 1 1 ; 0 1; 0 0];
x = points(:,1);
y = points(:,2);

t = 0:dt:tperiod(end);
x_des = interp1(tperiod,x,t);
y_des = interp1(tperiod,y,t);
% plot(x_des,y_des, '.',x_des(1),y_des(1),'ro',x_des(end),y_des(end),'go')

his.X=[];
his.ex=[];
his.ey=[];

%% Graphics
f3=figure;
ax3=axes('parent',f3);
PLOT.RefTrajectory=line('parent',ax3,'XData',[],'YData',[],'LineStyle',':',
'color','r','LineWidth',2);
PLOT.Robot=line('parent',ax3,'XData',[],'YData',[],'LineStyle','-
','color','g','LineWidth',2);
xlabel('x (m)')
ylabel('y (m)')
legend('Reference Trajectory','Acual ')

```

```

grid on

% robot dimensions
A.R_w = 0.3/2; % robot width/2
A.R_l=0.5/2; % robot length/2
A.a1 = [-A.R_l -A.R_w]';
A.b1 = [A.R_l -A.R_w/2]';
A.b2 = [A.R_l A.R_w/2]';
A.c = [-A.R_l A.R_w]';
A.P = [A.a1 A.b1 A.b2 A.c];
pl=[];

A.Rot = [ cos(X(3)) -sin(X(3)); sin(X(3)) cos(X(3))] * A.P; %rotated car
A.Prot_trasl = A.Rot + [ ones(1,4)*X(1); ones(1,4)*X(2)]; % add offset of
car's center

A.P_robot=patch(A.P(1,:),A.P(2,:), 'g');
A.P_robot.XData=A.Prot_trasl(1,:)' ;
A.P_robot.YData=A.Prot_trasl(2,:)' ;
axis([-0.5 1.5 -0.5 1.5])

%% Main loop

for i=1:length(x_des)

    % Simulate WMR
    X=Move_Robot(X, [Vref, Omegaref], dt);

    % IOSFL Controller
    xb=X(1)+bref*cos(X(3));
    yb=X(2)+bref*sin(X(3));

    ex= x_des(i)-xb;
    ey= y_des(i)-yb;

    Vd_x=k*ex;
    Vd_y=k*ey;

    Vref = cos(X(3))*Vd_x + sin(X(3))*Vd_y ;
    Omegaref = (1/bref)*(cos(X(3))*Vd_y-sin(X(3))*Vd_x);

    u_1_Vref=[Vref u_1_Vref];
    u_2_Omegaref=[Omegaref u_2_Omegaref];

    % Store the data

    his.X=[his.X X];
    his.ex=[his.ex ex];
    his.ey=[his.ey ey];

    % update the figures

    set(PLOT.RefTrajectory, 'XData', x_des(1:i), 'YData', y_des(1:i));
    set(PLOT.Robot, 'XData', his.X(1,:), 'YData', his.X(2,:));

```

```
A.Rot = [ cos(X(3)) -sin(X(3)); sin(X(3)) cos(X(3))] * A.P; %rotated car
A.Prot_trasl = A.Rot + [ ones(1,4)*X(1); ones(1,4)*X(2)]; % add offset
of car's center
A.P_robot.XData=A.Prot_trasl(1,:);
A.P_robot.YData=A.Prot_trasl(2,:);
drawnow
```

```
end
```

```
figure (2)
plot(t,his.ex)
axis([0 40 -0.5 0.5]);
grid
title('Erreur ex')
xlabel('t')
ylabel('ex')
```

```
figure (3)
plot(t,his.ey)
axis([0 40 -0.5 0.5]);
grid
title('Erreur ey')
xlabel('t')
ylabel('ey')
```

```
figure (4)
plot(t,u_1_Vref)
axis([0 40 -1 .5]);
grid
title('Input 1')
xlabel('t')
ylabel('u1')
```

```
figure (5)
plot(t,u_2_Omegaref)
axis([0 40 -10 5]);
grid
title('Input 2')
xlabel('t')
ylabel('u2')
```

# Références Bibliographiques

- [1] Siegwart, R., Nourbakhsh, I.: Introduction to Autonomous Mobile Robots. MIT Press (2004).
- [2] G. Klancar, A. Zdešar, S. Blažič, and I. Škrjanc, *Wheeled mobile robotics - From fundamentals towards autonomous systems*. Elsevier, Butterworth-Heinemann, 2017.
- [3] B. Bayle « Robotique Mobile », chapitre lessons, école nationale supérieure de physique de strasbourg 2010-2011.
- [4] S. Lens. *Locomotion d'un robot mobile*. Thèse de doctorat, Faculté des Sciences Appliquées, Université de Liège, 2008.
- [5] H. Moravec, *Stanford card and the CMU rover*, proceeding of IEEE vol 71, N° 7, pp 872-884, juillet 1983.
- [6] Pierre Arnaud, *Des moutons et des robots : architecture de contrôle réactive et déplacements collectifs de robots*, PUR presses polytechnique, 2000.
- [7] L. Steels. *When are robots intelligent autonomous agents?* Journal of robotics, Volume 15 pp 3-9, 1995.
- [8] F. Sempe, *Auto-organisation d'une collectivité de robots : Application à l'activité de patrouille en présence de perturbations*. Thèse de Doctorat de l'Université Pierre et Marie Curie, Paris 6, 2004.
- [9] D. Bonnafous, *Exécution réactive de trajectoires pour robots mobiles non-holonomes*. Thèse de doctorat. Institut National Polytechnique de Toulouse, Systèmes Automatiques. France, 2003.
- [10] Éric Beaudry, *Planification de tâches pour un robot mobile autonome*, Thèse de Maître de sciences, Université de Sherbrooke, Québec, Canada. Août 2006.
- [11] Bernard BAYLE, *Robotique mobile*, Thèse de doctorat, Ecole Nationale Supérieure de Physique de Strasbourg, Université de Strasbourg 2008.
- [12] R.C.Arkin, *Formation Behaviors, behavior-Base robotics*, The MIT Press. Cambridge, MA, USA, 1998.
- [13] L. Adouane, *Architecture de contrôle comportementale réactive pour la coopération d'un groupe de robots mobiles*. Thèse de doctorat, Laboratoire d'automatique de Besançon UMR (CNRS 6589) France, Avril 2005.
- [14] J.-Y. Fourquet et M. Renaud. *Coordinated Control of a Non-Holonomic Mobile Manipulator*. In ISER'1999, pages 115–125, Sydney, Australie, mars 1999.
- [15] G. Campion, G. Bastin et B. D'Andrea-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.
- [16] J. Neimark et N. Fufaev. *Dynamics of nonholonomic systems*, volume 33. Translations of Mathematical Monographs, 1972.
- [17] F. W. Warner. *Foundations of differentiable manifolds and lie groups*. Graduate Texts in Mathematics. Springer-Verlag, 1983.
- [18] H. Nijmeijer et A. J. Van der Shaft. *Nonlinear dynamical control systems*. Springer Verlag, New York, 1990.
- [19] 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.
- [20] ActivMedia. *P3-DX: World's Most Popular Intelligent Wheeled Robot*, 2004. <http://www.activrobots.com>.
- [21] K-team. *Khepera II user manual*, 2002. <http://www.k-team.com>.
- [22] P. Morin et C. Samson. *Introduction à la commande par retour d'état des systèmes non-holonomes*. Ecole des Mines de Paris, 2003.
- [23] 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.
- [24] C. Urmon, 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*. Rapport technique, Université de Carnegie Mellon, Pittsburgh, États-Unis, 2004.
- [25] G. Dudek et M. Jenkin. *Computational principles of mobile robotics*. Cambridge University Press, 2000.
- [26] J.-P. Laumond, éditeur. *La robotique mobile*. Hermes Sciences, 2001.

- [27] Khatib, E.I.A., Al-Masri, W.M.F., Mukhopadhyay, S., Jaradat, M.A., Abdel-Hafez, M.: A comparison of adaptive trajectory tracking controllers for wheeled mobile robots. In: 2015 10th International Symposium on Mechatronics and its Applications (ISMA), 8-10 Dec. 2015, pp. 1–6 (2015)