

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

ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE



Département d'Automatique



Projet de Fin d'Etudes

Pour l'obtention du diplôme

D'ingénieure d'état en automatique

THEME

**Commande Non Linéaire et Hybride  
d'un Robot Mobile Unicycle**

Proposé et dirigé par :

Pr. M.TADJINE

Etudié par :

Hamza OUBADJI

Juin 2010

Laboratoire de commande des processus

Ecole Nationale Polytechnique, 10, AV.Hassen Badi, El-Harrach, Alger

**ملخص:** نهتم في هذا العمل بدراسة نظام غير هولونومي متمثل في جهاز آلي متحرك أحادي الدور. اقترحنا مجموعة من تقنيات التحكم التي يتم بواسطتها تثبيت الجهاز حول المركز، وهي كالاتي:

- التحكم المتقطع (نظرية المساحة غير المتغيرة)
- التحكم الإنزلاقي
- باك ستيننج
- التحكم الهجين

في القسم الأخير من العمل اهتمنا بدراسة عملية إتباع الجهاز الآلي المتحرك لمسار مرجعي باستعمال التحكم غير الخطي.

**مفاتيح:** نظام غير هولونومي، جهاز آلي أحادي الطور، التحكم المتقطع، التحكم الهجين، إتباع مسار مرجعي.

### Résumé :

Ce travail est consacré à l'étude du mouvement d'un robot mobile de type uni cycle (un système non homonome).

Au premier lieu, nous nous somme intéressés à la stabilisation du robot autour de l'origine. Dans ce cadre, différentes commandes cinématique sont appliqués au système :

- Commande par retour d'état discontinue
- Commande par mode glissant
- Technique du backstepping.
- Commande hybride.

En deuxième lieu, nous avons étudié la problématique de poursuit de trajectoire en appliquant une commande non linéaire

Mots clés : Système non holonome, robot mobile un cycle, commande par retour d'état discontinu, commande hybride, back stepping, poursuite de trajectoire.

### Abstract:

The subject of this work is the motion control of a wheeled mobile robot.

With reference to the unicycle kinematics, we review several control strategies for posture stabilization:

- Discontinues control (invariant manifold)
- Slipping mode control
- Back stepping
- Hybrid control

For the problem of trajectory tracking, a non linear control law is considered.

Key words: non homonymic system, unicycle mobile robot, discontinuous, control, hybrid control, back stepping, trajectory tracking.

## Remerciements :

Je tiens à exprimer mes sincères remerciements à mon promoteur Monsieur M.TADJINE, professeur à l'Ecole Nationale Supérieure Polytechnique d'Alger qui n'a cessé de m'apporter d'aide et soutien lors de ce travail. Qu'il trouve, ici le témoignage de ma profonde gratitude et reconnaissance.

Je remercie les membres de jury me faisant honneur par leur participation à l'examen de ce travail.

Je tiens également à remercier l'ensemble de tous mes enseignants pour le précieux savoir qu'ils m'ont prodigué

## *Dédicaces*

*A ma mère*

*A mon père*

*A mes chères grand-mères*

*A mon frère, A mes sœurs*

*A mes oncles, tantes et cousins*

*A mon ami Younes*

*A tous mes proches*

*Je dédie ce modeste travail.*

## Sommaire

<b>Introduction Générale</b> .....	01
<b>Chapitre 1 : Etat de l'art sur la robotique mobile</b> .....	04
1.1. Introduction : .....	04
1.2. Etude de la cinématique des véhicules à roues: .....	05
1.2.1. Hypothèses :.....	05
1.2.2. Roulement sans glissement :.....	05
1.2.3. Modélisation des robots à roues :.....	07
1.3. Les grandes classes de robots mobiles .....	07
1.3. 1 Disposition des roues et centre instantané de rotation:.....	07
1.3.2. Robot mobile différentiel ou unicycle :.....	08
1.3.2. a. Description :.....	08
1.3.2. b. Modélisation :.....	09
1.3.3. Robot mobile tricycle .....	11
1.3.3. a. Description.....	11
1.3.3. b. Modélisation :.....	12
1.3.4. Robot mobile de type voiture.....	13
1.3.4. a. Modélisation .....	14
1.3.5. Robot mobile omnidirectionnel :.....	15
1.3.5. a. Description :.....	15
1.3.5. b. Modélisation.....	15
1.3.6. Robot mobile à traction synchrone .....	17
1.3.6. a. Description.....	17
1.3.6. b. Modélisation cinématique.....	18
1.4. Holonomie et non-holonomie :.....	21
1.5. Conclusion .....	22

## **Chapitre2:Commandes Non Linéaires Stabilisantes**

<b>du Robot Mobile Unicycle.....</b>	<b>23</b>
2.1. Problématique :.....	23
2.1.1. Introduction :.....	23
2.1.2. Problème de commande des systèmes non-holonom.....	23
2.1.2. Stabilisation des systèmes non-holonomes :.....	24
2.2. Commande par retour d'état discontinu (invariant manifold approach)[3].....	25
2.2.1. Présentation de la forme chaînée :.....	26
2.2.2. Algorithme générale de stabilisation :.....	26
2.2.3. Application de l'algorithme de stabilisation au robot mobile :.....	29
2.2.4. Simulations.....	31
2.3. Commande stabilisante en coordonnées polaires [4] :.....	34
2.3.1. Transformation en coordonnées polaires :.....	34
2.3.2. Simulations :.....	35
2.3. Stabilisation du robot par technique du backstepping :.....	39
2.3.1. Cadre théorique :.....	39
2.3.2. Stabilisation du robot :.....	41
2.3.3. Simulations :.....	43
Conclusion:.....	46
<b>Chapitre 3 : Stabilisation du Robot Mobile par Commandes Hybride.....</b>	<b>47</b>
3.1. Théorie des systèmes hybrides [5].....	47
3.1.1. Définition des systèmes hybrides :.....	47
3.1.2. Définition des automates hybrides :.....	49
3.1.3. Exécution d'un système hybride :.....	51
	52

3.2. Commande hybride stabilisante de l'unicycle [6] :.....	
3.2.1. L'intégrateur non holonome (de Brockett1983) :.....	52
3.2.2. Commande hybride de l'intégrateur non-holonome.....	53
3.2.3. Conditions de transitions :.....	55
Simulations:.....	56
3.3. Stabilisation par une commande hybride dynamique [7] :.....	60
3.3.1. Le double intégrateur non holonome(DIN) :.....	60
3.3.2. Commande hybride du (DIN) :.....	63
Résultats de simulation :.....	65
Conclusion :.....	66
<b>Chapitre 4 : Poursuite de Trajectoire du Robot Mobile.....</b>	<b>67</b>
Introduction .....	66
4.1. Génération des commandes (feedforward) :.....	66
4.2. Lois de commandes stabilisant la trajectoire :.....	67
4.3. Contrôlabilité de la trajectoire :.....	68
4.4. Commande non linéaire de poursuite :.....	69
4.5. Simulations :.....	71
<b>Conclusion Générale.....</b>	<b>73</b>

### Introduction Générale :

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

On peut estimer que les robots mobiles à roues constituent le gros des robots mobiles. Historiquement, leur étude est venue assez tôt, suivant celle des robots manipulateurs, au milieu des années 70. Leur faible complexité en a fait de bons premiers sujets d'étude pour les roboticiens intéressés par les systèmes autonomes. Cependant, malgré leur simplicité apparente (mécanismes plans, actionneurs linéaires), ces systèmes ont soulevé un grand nombre de problèmes difficiles.

Ainsi, alors que les robots manipulateurs se sont aujourd'hui généralisés dans l'industrie, rares sont les applications industrielles qui utilisent des robots mobiles. Si l'on a vu depuis peu apparaître quelques produits manufacturiers (chariots guides) ou grand public (aspirateur), l'industrialisation de ces systèmes bute sur divers problèmes délicats.

Ceux-ci viennent essentiellement du fait que, contrairement aux robots manipulateurs prévus pour travailler exclusivement dans des espaces connus et de manière répétitive, les robots mobiles sont destinés à évoluer de manière autonome dans des environnements peu ou pas structurés.

Néanmoins, l'intérêt indéniable de la robotique mobile est d'avoir permis d'augmenter considérablement nos connaissances sur la localisation et la navigation de systèmes autonomes. Dans cette étude, on se limitera volontairement à une présentation succincte des robots mobiles à roues les plus classiques.

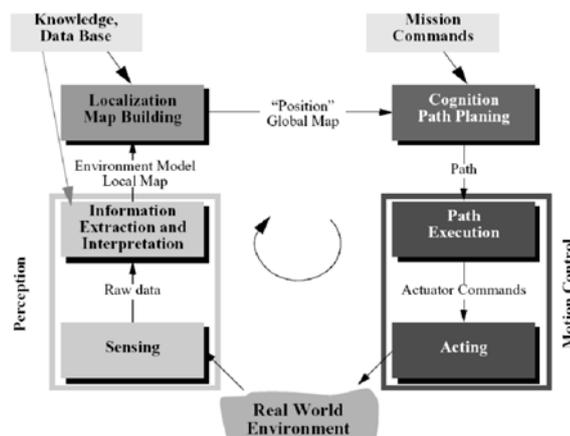
- **Problèmes en robotique mobile**

On distingue sans trop d'ambiguïté un certain nombre de problèmes en robotique mobile. 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.

## Introduction Générale

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 avec ou sans collecteur, 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. Bien évidemment, l'architecture du robot dépend fortement de l'offre et des choix technologiques du moment.



Un des problèmes majeurs en ce qui concerne la commande de ces systèmes, est non existence d'un retour d'état continu invariant dans le temps stabilisant. Pour remédier à ce problème, des méthodes de synthèse de lois de commandes ont été proposées. Parmi ces méthodes, on rencontre les retours d'état discontinus invariant dans le temps, les retours d'état continus mais variant dans le temps, la commande hybride et Les techniques de commande optimale.

## Introduction Générale

---

Le travail que nous présentons est décomposé en deux parties. la première est consacrée au problème de stabilisation des robots mobiles de type unicycle autour de l'origine. La deuxième partie, contient une étude succincte du problème de poursuite de trajectoire.

Plus précisément, la première partie comporte trois chapitres. Le chapitre 1 est consacré à la définition des différentes catégories de robots mobiles ainsi que leur modélisation cinématique.

Dans le deuxième chapitre, nous établissons notre problématique, il s'agit de parler de la stratégie de stabilisation du robot mobile de type unicycle par retour d'état discontinu et invariant dans le temps, commande en coordonnées polaires, ainsi que l'application de la technique de Backstepping.

Cette partie s'achève avec le chapitre 3 consacré, à l'approche hybride de stabilisation. Dans ce cadre, on développe deux lois de commande hybride stabilisantes, la première est appliquée au modèle cinématique du robot, or la deuxième s'étend au modèle dynamique simplifié.

Dans le dernier chapitre, on s'intéresse à la deuxième problématique de commande de notre robot qui est la poursuite de trajectoire.

Pour chacune de ces méthodes de commande nous présentons les différents résultats de simulations montrant les performances de commande.

Chapitre1 :  
Etat de l'Art sur la Robotique Mobile

# Chapitre 1 : Etat de l'art sur la robotique mobile

## 1.1. Introduction :

Un robot est une machine capable d'agir sur son environnement et de réaliser des tâches diverses. Il est doué d'une capacité de s'auto-adapter à différentes situations. Pour bien mener son fonctionnement, un robot est équipé de capteurs lui permettant de percevoir l'environnement dans lequel il évolue, et d'effecteurs (roue, bras, pince, jambe, etc.) pour agir sur cet environnement.

D'une façon générale, on distingue deux groupes de robots : les robots fixes désignant les robots manipulateurs et les robots mobiles ayant la tendance de se déplacer selon une trajectoire donnée.

On regroupe sous l'appellation «robots mobiles» l'ensemble des robots à base mobile, par opposition notamment aux robots manipulateurs.

Le terme « robot mobile » est attribué, dans la plupart des cas, aux robots mobiles à roues. Les autres robots mobiles sont, le plus souvent, désignés par leur type de locomotion, qu'ils soient marcheurs, sous-marins ou aériens.

Le domaine de la poursuite et de la planification de trajectoire constitue un sujet important dans l'étude du comportement des robots mobiles dans un environnement hostile. C'est pour cette raison qu'on s'attache à mettre l'accent sur les différentes techniques de suivi de trajectoire et leur implémentation dans des cas réels de robotique mobile.

Pour bien mener cette étude, nous consacrons ce chapitre à la présentation d'un état de l'art sur les robots mobiles. Pour ce faire, nous commençons par définir le principe de roulement sans glissement, puis présenter les différents types de robots mobiles à roues à savoir les plus utilisés en pratique, ainsi que leurs caractéristiques cinématiques.

## 1.2. Etude de la cinématique des véhicules à roues [1][2]:

### 1.2.1. Hypothèses :

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

- les véhicules sont considérés comme rigides (indéformables) et évoluant sur un plan.
- les véhicules sont dotés de roues conventionnelles : le point de contact entre la roue et le sol est réduit à un point  $I$  et la roue est soumise à la contrainte de roulement sans glissement.

### 1.2.2. Roulement sans glissement :

La locomotion à l'aide de roues exploite la friction au contact entre la roue et le sol. Pour cela, la nature du contact a une forte influence sur les propriétés du mouvement relatif de la roue par rapport au sol.

Dans de bonnes conditions, il y a roulement sans glissement (*r.s.g*) de la roue sur le sol, c'est-à-dire que la vitesse relative de la roue par rapport au sol au point de contact  $I$  est nulle.

En pratique, le contact se fait sur une surface, ce qui engendre de légers glissements. Nous considérons que ce glissement est faible c'est-à-dire que la vitesse au point de contact est supposée nulle.

Pour exprimer analytiquement la condition de r.s.g on considère une roue verticale qui roule sans glisser sur un sol plat (figure I.1), dans un repère  $R(O, X, Y, K)$ .

Le repère  $R_1(O_1, X_1, Y_1, K_1)$  est lié à la roue et le roulement sans glissement se traduit par une vitesse nulle au point  $I$  de la roue en contact avec le sol. Une vitesse qui sera exprimée en fonction de la vitesse de rotation  $\omega$  de la roue et de la vitesse de son centre  $O_1$ .

Dans ce cas, on fait correspondre à tout point  $P$  un champ de vecteurs noté  $v_p$ . Ce dernier qui représente la vitesse absolue de l'origine par rapport au repère  $R$  est donné comme suit :

$$\vec{v}_p = \frac{d}{dt}(\vec{op}) \quad (1.1)$$

Soit  $\Omega$  la résultante du torseur représentant le vecteur de rotation du corps par rapport à  $R$ . La connaissance de  $v_p$  et de  $\Omega$  permet de calculer la vitesse de la roue en un point  $A$  par la relation fondamentale suivante :

$$\vec{v}_p = \vec{v}_A + \vec{\Omega} \wedge \vec{Ap} \quad (1.2)$$

A partir de la relation (1.2), on peut calculer aisément la vitesse du point  $I$  par rapport au repère  $R$ . D'où :

$$\vec{V}\left(I/R\right) = \vec{o} = \vec{V}(O_1/R) + \vec{\omega}_{(I/O)} \wedge \overrightarrow{O_1I}$$

$$\vec{V}\left(I/R\right) = \vec{o} = \dot{x}_{re}\vec{X} + \dot{y}_{re}\vec{Y} + (\dot{\varphi}\vec{k} + \dot{\theta}(-\sin\varphi\vec{X} + \cos\varphi\vec{Y})) \wedge (-r\vec{k})$$

$$\vec{V}\left(I/R\right) = \vec{o} = (\dot{x}_{re} - r\dot{\theta}\cos\varphi)\vec{X} + (\dot{y}_{re} - r\dot{\theta}\sin\varphi)\vec{Y}$$

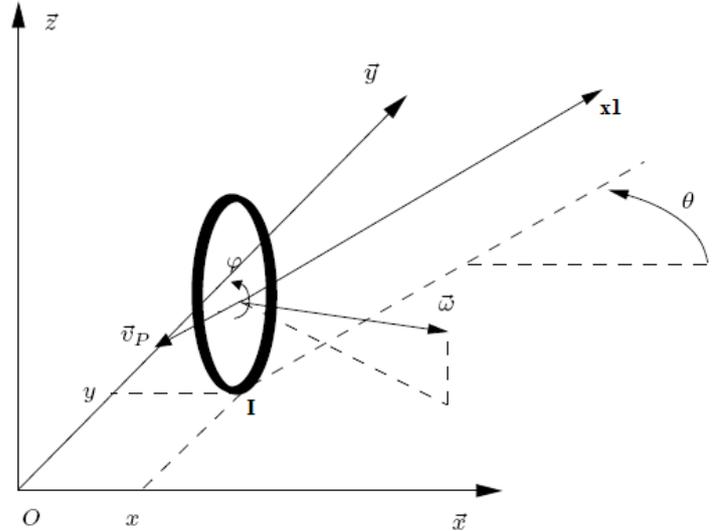


Figure 1.1. Caractérisation du roulement sans glissement.

Avec r le rayon de la roue et (x, y) les coordonnées du point O<sub>1</sub> dans R.

$$\vec{V}\left(I/R\right) = \vec{o}$$

Ceci implique que deux types de contraintes peuvent être déduits par l'intermédiaire des équations suivantes :

$$\dot{x}_{re} - r\dot{\theta}\cos\varphi = 0 \tag{1.3}$$

$$\dot{y}_{re} - r\dot{\theta}\sin\varphi = 0 \tag{1.4}$$

Ces deux équations peuvent être transformées pour faire apparaître les composantes de vitesse dans le plan de la roue d'une part et perpendiculairement à la roue d'autre part. Les nouvelles équations sont données comme suit :

$$-\dot{x}_{re}\sin\varphi + \dot{y}_{re}\cos\varphi = 0 \tag{1.5}$$

$$\dot{x}_{re}\cos\varphi + \dot{y}_{re}\sin\varphi = 0 \tag{1.6}$$

Ce système d'équation traduit les deux propriétés suivantes :

- 1) la vitesse du centre de la roue est parallèle au plan de la roue
- 2) la vitesse du centre de la roue est  $r\dot{\theta}$ .

### 1.2.3. Modélisation des robots à roues :

D'une façon générale, le comportement d'un robot à roues peut être modélisé sous la forme suivante :

$$\dot{q} = \sum_{i=1}^m g_i(q) u_i$$

Où  $q \in \mathbb{R}^n$ ,  $m < n$ ,  $u_i$  ( $i = 1, \dots, m$ ) sont des variables de commande et  $g_i(q)$  ( $i = 1, \dots, m$ ) sont des champs de vecteurs différentiables sur  $\mathbb{R}^n$ .

### 1.3. Les grandes classes de robots mobiles [1] [2]

#### I.3.1 Disposition des roues et centre instantané de rotation:

La configuration d'une roue et le nombre de paramètres nécessaires à sa description dépendent du type de roue considérée. Nous introduisons ici les quatre types de roues principalement utilisés en robotique mobile (figure 1.2) :

- les roues fixes dont l'axe de rotation passe par le centre de la roue, tandis que l'axe d'orientation est constant (figure I.2a).
- les roues centrées orientables dont l'axe d'orientation, perpendiculaire au sol, passe par le centre de la roue (figure I.2b).
- les roues décentrées orientables dont l'axe d'orientation, perpendiculaire au sol, ne passe pas par le centre de la roue. Ces roues sont appelées roues folles, (figure I.2c).
- les roues suédoises dont la bande de roulement a été remplacée par des galets inclinés par rapport à la normale au plan de la roue. C'est la combinaison de la rotation de la roue avec la rotation libre du galet en contact avec le sol qui permet un déplacement sans glissement sur le sol (figures I.2d).

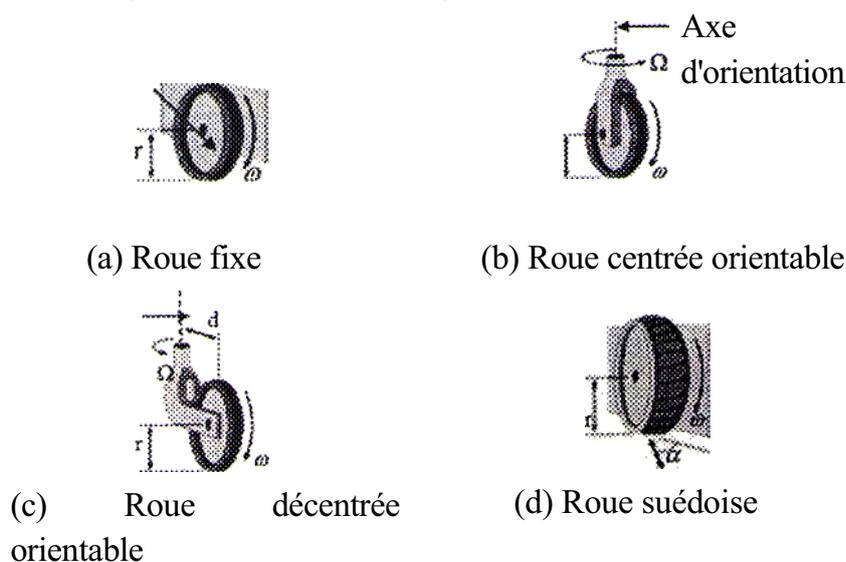


Figure 1.2. Les principaux types de roues dans la robotique mobile

Ces quatre types de roues sont les plus utilisés en robotique mobile. Cependant, ils existent d'autres types comme les roues sphériques qu'on peut trouver dans plusieurs prototypes vu qu'elle offre au robot une propriété très intéressante qui est la propriété omnidirectionnelle (voir figure 1.14).

Pour un ensemble de roues données, toute disposition ne conduit pas à une solution viable. Un mauvais choix de roues 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 glissements sur le sol, toutes les roues doivent contenir un point de vitesse nulle unique autour duquel tourne le robot en instantanée. Ce point, lorsqu'il existe, il unique et appelé Centre Instantané de Rotation (CIR). Le CIR s'obtient par l'intersection des axes de rotation des différentes roues. Ce point est belle bien unique.

Après avoir donné quelques hypothèses et les différentes configurations de roues existantes, on présente dans ce qui suit les principales catégories de robots mobiles.

### 1.3.2. Robot mobile différentiel ou unicycle :

#### 1.3.2. a. Description :

Une des configurations les plus utilisées pour les robots mobiles est la configuration différentielle (differential drive) qui comporte deux roues fixes non orientables commandées indépendamment. Une ou plusieurs roues folles sont ajoutées à l'avant ou à l'arrière du robot pour assurer sa stabilité. Dans certains cas, ils existent certains robots différentiels avec quatre roues commandées indépendamment sauf qu'ils sont modélisés en deux roues. Le schéma de principe du robot différentiel est présenté par la figure 1.3.

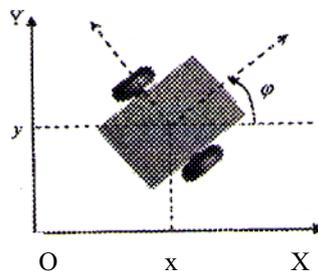
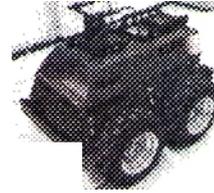


Figure 1.3. Schéma de principe d'un robot mobile différentiel

Ce type de robot est très répandu en raison de sa simplicité de développement et de ses propriétés cinématiques intéressantes, comme sa capacité de tourner sur lui même. La figure 1.4 présente deux exemples de robots mobiles différentiels.



(a) Robot mobile SuperMario



(b) robot ATRV2

Figure 1.4. Exemples de robots mobiles différentiels.

### 1.3.2.b. Modélisation :

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

Soit  $\rho$  le rayon de courbure de la trajectoire du robot, c'est à dire la distance du *CIR* au point *C* (figure 1.5).

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

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

$$v = v_g + \vec{\Omega} \wedge \overrightarrow{CA}$$

$$v = v_d + \vec{\Omega} \wedge \overrightarrow{CB}$$

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

$$2v = v_g + v_d + \vec{\Omega} \wedge (\overrightarrow{CA} + \overrightarrow{CB}) = v_g + v_d$$

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

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

$$v_d - v_g + \Omega \wedge (\overrightarrow{CB} - \overrightarrow{CA}) = 0 ; \text{ or } (\overrightarrow{CB} - \overrightarrow{CA}) = 2L \text{ d'où l'équation (1.10)}$$

$$v_d = (R + L)\Omega \quad (1.8)$$

$$v_g = (R - L)\Omega \quad (1.9)$$

A partir de ces deux équations, on peut déterminer  $R$  et  $\Omega$  (en fonction des vitesses des roues) de la façon suivante :

$$\Omega = \frac{v_d - v_g}{2L} \quad (1.10)$$

$$R = L \frac{v_d + v_g}{v_d - v_g} \quad (1.11)$$

La vitesse linéaire  $v$  du robot au point  $O'$  est :

$$v = \frac{v_d + v_g}{2} \quad (1.12)$$

La vitesse de rotation du robot est égale à la vitesse de rotation autour du *CIR* :

$$\Omega = \dot{\varphi} = \frac{v_d - v_g}{2L} \quad (1.13)$$

L'équation (1.11) permet de situer le *CIR* sur l'axe des roues. Par ailleurs, ces équations expliquent deux propriétés particulières du mouvement des robots différentiels:

- Si  $v_d = v_g$  la vitesse angulaire  $\Omega$  sera nulle et le rayon de courbure  $\rho$  est infini et le robot se déplace donc en ligne droite.
- Si  $v_d = -v_g$ ,  $\Omega \neq 0$  et  $\rho$  est nul, alors le robot effectue une rotation sur lui-même.

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

L'utilisation de ce mode de locomotion fournit une solution simple pour déplacer le robot d'une position à une autre. C'est, sans doute, une des raisons du succès de ce type de robot.

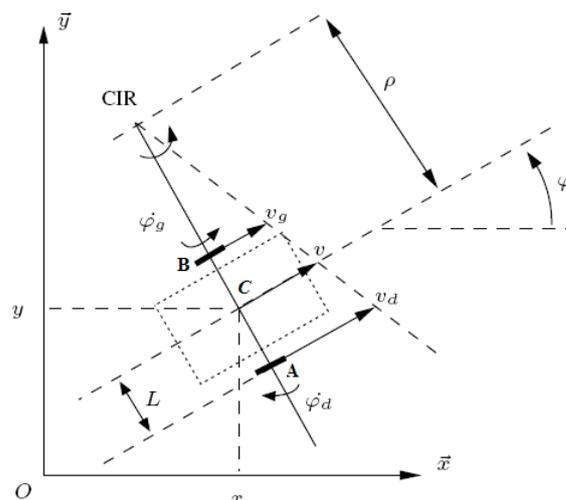


Figure 1.5. Le CIR d'un robot mobile différentiel

Il est clair que  $v = (\dot{x}^2 + \dot{y}^2)^{\frac{1}{2}}$ , par l'introduction de la contrainte de  $gsr(1.5)$ .

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

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \Omega \end{cases} \quad (1.14)$$

Ces équations relient la dérivée de la position  $(x, y, \theta)$  du robot à la commande  $u = (v, \Omega)^T$ , avec  $\theta$  la rotation instantanée du robot par rapport au repère  $(O, X, Y)$ .

### 1.3.3. Robot mobile tricycle

#### 1.3.3.a. Description

L'architecture d'un robot mobile tricycle est représentée dans la figure 1.6. Ce robot est constitué de deux roues fixes de même axe et d'une roue centrée orientable placée sur l'axe longitudinal du robot. Le mouvement du robot dépend de deux actions : la vitesse longitudinale et l'orientation de la roue orientable.

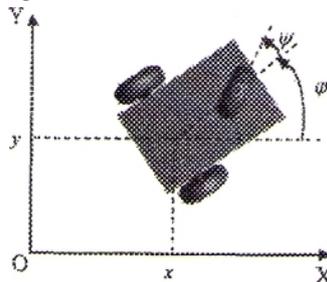


Figure 1.6. Schéma de principe d'un robot mobile tricycle

Deux prototypes de robots tricycles sont représentés dans la figure 1.7.

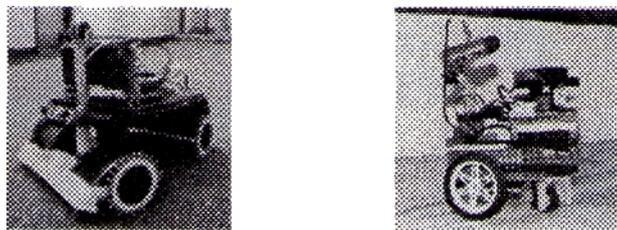


Figure 1.7. Exemples de robots mobiles de type tricycle

### 1.3.3.b. Modélisation :

Le *CIR* du robot se situe à l'intersection des axes des roues fixes et de la roue

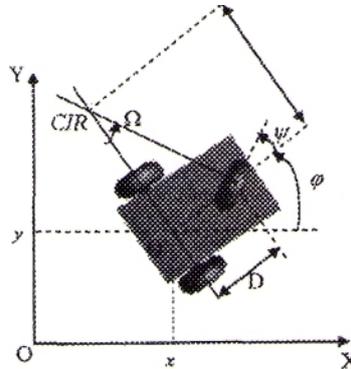


Figure 1.8. Le *CIR* du robot mobile de type tricycle

On peut déterminer  $R$  de manière géométrique à partir de l'angle d'orientation  $\psi$  de la roue avant et la vitesse de rotation autour du CIR ( $\Omega$ ) à partir de la vitesse linéaire  $v$  du robot (vitesse en  $O'$ ). Les expressions de  $R$  et  $\Omega$  sont données comme suit :

$$R = \frac{D}{\tan \psi} \quad (1.20)$$

$$\Omega = \frac{v}{D} \tan(\psi) \quad (1.21)$$

La vitesse linéaire  $v$  peut être exprimée en fonction de la vitesse linéaire de la roue orientable  $v_s$  :

$$v = v_s \cos \psi \quad (1.22)$$

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

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

- **Modélisation cinématique :**

L'écriture des contraintes correspondantes à chacune des roues est similaire à celle établie dans le cas du robot différentiel. Ceci permet de déterminer le modèle cinématique du robot tricycle. Toutefois, par un simple raisonnement géométrique, nous pouvons établir les équations représentant la dérivée de la position du robot de la façon suivante :

$$\dot{x} = v \cos \varphi \quad (1.23)$$

$$\dot{y} = v \sin \varphi \quad (1.24)$$

$$\dot{\varphi} = \Omega = \frac{v}{D} \tan \psi \quad (1.25)$$

$$\dot{\psi} = \Omega_s \quad (1.26)$$

Où  $u = (v, \Omega_s)^T$  est le vecteur de commande cinématique,  $\varphi$  la rotation instantanée du robot par rapport au repère (O, X, Y), et la vitesse d'orientation correspondante à la roue orientable.

### 1.3.4. Robot mobile de type voiture

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

A titre d'exemple, voici deux prototypes de robots mobiles de type voiture (figure 1.9).



(a) Robot Kanade



(b) Robot Cycab

Figure 1.9. Exemples de robots mobiles de type voiture.

### 1.3.4.a. Modélisation

Comme nous l'avons vu précédemment, l'existence d'un *CIR* unique impose que les axes des roues du robot soient concourants. Dans le cas du robot de type voiture, cela impose aux roues avants d'avoir une orientation différente, comme illustré dans la figure 1.10. Le roulement idéal, assurant que le *CIR* est bien unique, est réalisé par un système de braquage différentiel (dit d'Ackerman). Par ailleurs, les roues n'ayant pas le même rayon de courbure ont des vitesses différentes.

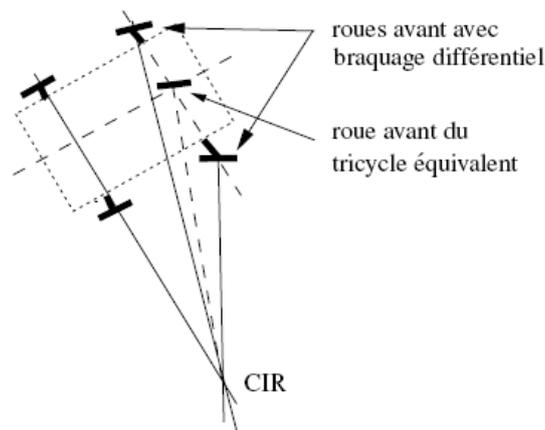


Figure 1.10. Schéma de principe d'un robot mobile de type voiture

- **Modélisation cinématique**

Le modèle cinématique du robot de type voiture est donné par les équations suivantes :

$$\dot{x} = v \cos \varphi \quad (1.30)$$

$$\dot{y} = v \sin \varphi \quad (1.31)$$

$$\dot{\varphi} = \frac{v}{D} \tan \psi$$

$$\dot{\psi} = \Omega \quad (1.32)$$

Ces équations relient la dérivée de la position  $(x,y,\varphi)$  du robot à la commande  $u = (v, \psi)^T$ , avec  $\varphi$  la rotation instantanée du robot par rapport au repère  $(O, X,Y)$ .

### 1.3.5. Robot mobile omnidirectionnel :

#### 1.3.5.a. Description :

Un robot mobile est dit omnidirectionnel si l'on peut agir indépendamment sur les vitesses de translation selon les axes  $x$  et  $y$ , et la vitesse de rotation autour de  $z$ . D'un point de vue cinématique, cela n'est pas possible avec des roues fixes ou des roues centrées orientables. On peut, en revanche, réaliser un robot omnidirectionnel ayant recours à un ensemble de trois roues décentrées orientables ou de trois roues suédoises disposées aux sommets d'un triangle équilatéral (figure 1.11).

Trois roues sont donc suffisantes. Cependant, dans certains cas, une quatrième roue offre des possibilités d'optimisation et permet de rendre le système plus robuste en évitant le glissement dans le cas d'un sol qui n'est pas parfaitement plat par exemple. En ce qui concerne les roues suédoises (figure I.2d), la rotation des galets inclinés permet aux roues du robot de rouler dans une direction perpendiculaire à celle autour duquel elles roulent normalement. C'est ce qui permet au robot de se déplacer dans diverses directions sans avoir besoin de tourner. Dans ce cas, le robot n'effectue pas de rotation mais uniquement des translations.

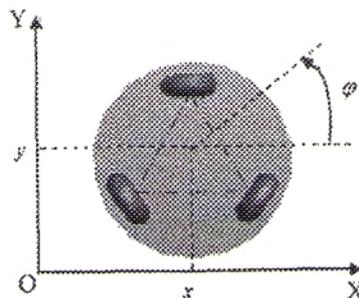


Figure 1.11. Schéma de principe d'un robot mobile omnidirectionnel.

#### 1.3.5.b. Modélisation

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

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

Où  $u = (u_1, u_2, u_3)^T$  représente le vecteur de commande. Nous choisissons d'une façon générale ce type de robot pour se dispenser des problèmes de planification et de

commande liés à la non-holonomie. La figure 1.12 représente des exemples de ce type de robots.



Figure 1.12. Exemples de robots mobiles omnidirectionnels :

- (a) Robot omnidirectionnel NOMAD muni de roues décentrées orientables.
- (b) Robot omnidirectionnel muni de roues suédoises.

Une autre technique consiste à utiliser des roues sphériques ou des associations de roues sphériques tronquées, montées orthogonalement les unes par rapport aux autres. Dans ce dernier cas, lors du pivotement de l'axe motorisé, le contact sur le sol se fait en alternance avec l'une ou l'autre des deux roues (sphères) formant un essieu (figure 1.13). Les axes de rotation des deux sphères sont perpendiculaires entre eux et également perpendiculaires et concourants avec l'axe longitudinal de l'essieu.

La mobilité interne que constitue la rotation libre des sphères permet un déplacement en translation de l'essieu (et de la structure qui le porte) dans sa direction longitudinale, sans glissement des roues sur le sol (figure 1.14).



Figure 1. 13. Représentation 3D d'un essieu

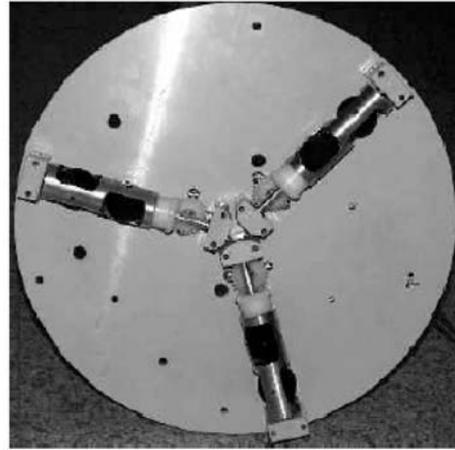


Figure 1.14. Disposition de trois essieux

### 1.3.6. Robot mobile à traction synchrone

#### 1.3.6.a. Description

La traction synchrone (synchronous drive) est une technique utilisée pour minimiser l'effet de glissement et augmenter la force de traction.

On rencontre ce type de robot dans l'industrie automobile et dans les robots tout terrain. La configuration du robot à traction synchrone est similaire à un robot à trois ou quatre roues couplées de façon qu'elles soient actionnées en même temps, en ayant la même vitesse et la même orientation.

Ce système est réalisé grâce à deux moteurs, un pour la traction et l'autre pour l'orientation. L'ensemble est relié par une chaîne (ou une ceinture) pour s'assurer que toutes les roues tournent de façon synchrone. La figure 1.15 montre un robot à quatre roues couplées.

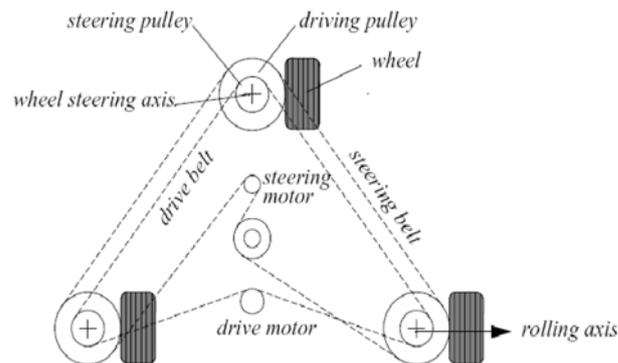


Figure 1.15. Architecture d'une plate-forme de robot mobile à traction synchrone.

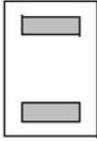
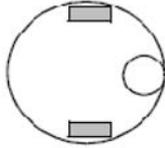
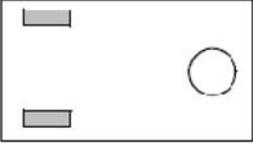
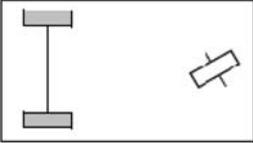
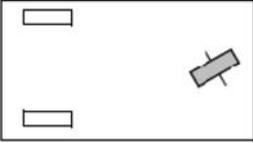
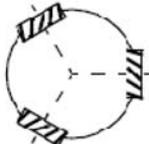
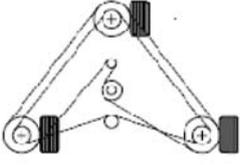
### 1.3.6.b. Modélisation cinématique

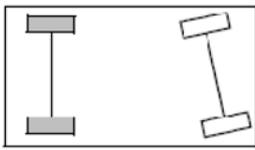
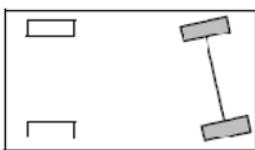
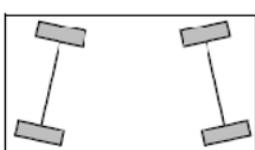
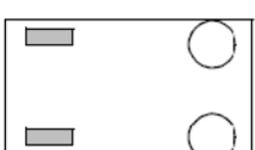
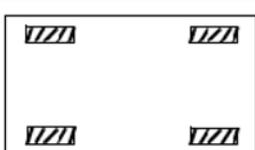
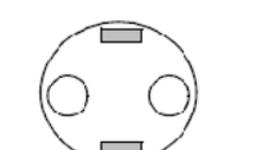
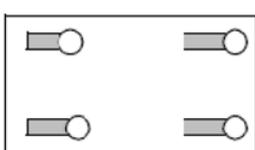
Afin de trouver les équations de déplacement d'un robot à traction synchrone, on utilise les contraintes déduites, du principe de roulement sans glissement, présentées dans les équations (1.3) et (1.4).

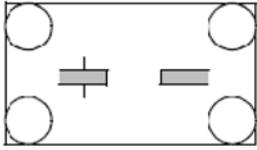
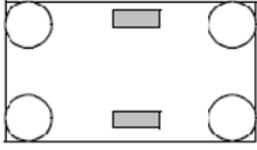
En introduisant la vitesse de roulement de la roue  $v = r \dot{\theta}$  et sa vitesse de rotation autour de l'axe  $k$  (figure 1.18), on forme le modèle cinématique suivant :

$$\begin{aligned} \dot{x}_{re} &= v \cos \varphi \\ \dot{y}_{re} &= v \sin \varphi \\ \dot{\varphi} &= \Omega \end{aligned}$$

Nous présentons dans le tableau récapitulatif (ci-dessous) les différentes architectures des robots mobiles à roues :

# of wheels	Arrangement	Description	Typical examples
2		One steering wheel in the front, one traction wheel in the rear	Bicycle, motorcycle
		Two-wheel differential drive with the center of mass (COM) below the axle	Cye personal robot
3		Two-wheel centered differential drive with a third point of contact	Nomad Scout, smartRob EPFL
		Two independently driven wheels in the rear/front, 1 unpowered omnidirectional wheel in the front/rear	Many indoor robots, including the EPFL robots Pygmalion and Alice
		Two connected traction wheels (differential) in rear, 1 steered free wheel in front	Piaggio minitrucks
		Two free wheels in rear, 1 steered traction wheel in front	Neptune (Carnegie Mellon University), Hero-1
		Three motorized Swedish or spherical wheels arranged in a triangle; omnidirectional movement is possible	Stanford wheel Tribolo EPFL, Palm Pilot Robot Kit (CMU)
		Three synchronously motorized and steered wheels; the orientation is not controllable	"Synchro drive" Denning MRV-2, Georgia Institute of Technology, I-Robot B24, Nomad 200

# of wheels	Arrangement	Description	Typical examples
4		Two motorized wheels in the rear, 2 steered wheels in the front; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with rear-wheel drive
		Two motorized and steered wheels in the front, 2 free wheels in the rear; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with front-wheel drive
		Four steered and motorized wheels	Four-wheel drive, four-wheel steering Hyperion (CMU)
		Two traction wheels (differential) in rear/front, 2 omnidirectional wheels in the front/rear	Charlie (DMT-EPFL)
		Four omnidirectional wheels	Carnegie Mellon Uranus
		Two-wheel differential drive with 2 additional points of contact	EPFL Khepera, Hyperbot Chip
		Four motorized and steered castor wheels	Nomad XR4000

# of wheels	Arrangement	Description	Typical examples
6		Two motorized and steered wheels aligned in center, 1 omnidirectional wheel at each corner	First
		Two traction wheels (differential) in center, 1 omnidirectional wheel at each corner	Terregator (Carnegie Mellon University)
Icons for the each wheel type are as follows:			
	unpowered omnidirectional wheel (spherical, castor, Swedish);		
	motorized Swedish wheel (Stanford wheel);		
	unpowered standard wheel;		
	motorized standard wheel;		
	motorized and steered castor wheel;		
	steered standard wheel;		
	connected wheels.		

#### 1.4. Holonomie et non-holonomie :

La capacité d'un robot mobile à pouvoir se déplacer, à partir d'une situation donnée, dans n'importe quelle direction est appelée « holonomie ». En effet, le mécanisme holonomique permet au robot de manœuvrer dans n'importe quelle direction arbitraire à partir de n'importe quelle configuration arbitraire.

Un système holonomique est un système qui comporte le même nombre de déplacements virtuels que le nombre de coordonnées généralisées pour le décrire. Comme nous l'avons vu, nous avons besoin de trois coordonnées généralisées pour décrire d'une façon unique la configuration d'un robot mobile sur le plan. A titre d'exemple, un robot holonomique omnidirectionnel est un robot pour lequel les trois mouvements planaires indépendants (deux de translation et un de rotation) sont admissibles à partir d'une

configuration quelconque.

Par contre, de nombreux robots mobiles sont des systèmes non-holonomes tels que les robots différentiels, les tricycles, les robots voitures, et les robots à traction synchrone. Pour ce type de robots, bien que l'espace de configuration sur le plan soit de trois dimensions, le mouvement est produit par deux déplacements indépendants seulement. Le robot, dans ce cas, ne possède que deux degrés de liberté.

D'un point de vue mathématique, nous pouvons dire que la non-holonomie d'un robot est due au fait que ses équations de contraintes ne sont pas intégrables. Pour cette raison, l'analyse cinématique, dynamique et le contrôle sont plus complexes étant donné que les coordonnées de leur représentation ne peuvent être éliminées en utilisant les équations de contraintes. A cet effet, les systèmes qui possèdent des contraintes non-holonamiques nécessitent toujours un nombre plus grand de coordonnées pour leur description que le nombre de degrés de liberté.

Un robot est dit *non-holonome* lorsque son *modèle cinématique* est régi par un système non linéaire de la forme :

$$\dot{x} = u_1 X_1(x) + \dots + u_m X_m(x), \quad x \in CS$$

CS est *l'espace de configuration* qui représente l'ensemble de toutes les configurations (ou positions) possibles du robot.

La dimension  $n$  de l'espace CS doit être plus grande que la dimension  $m$  des contrôles.

## 1.5. Conclusion

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

Nous avons montré aussi la différence entre un système holonome et un système non-holonome. Nous avons observé que la majorité des robots mobiles sont des systèmes non-holonomes sauf que dans le cas des robots omnidirectionnels.

Après avoir effectué un tour d'horizon sur les robots mobiles, nous présentons notre première problématique qui traite la commande des systèmes non-holonomes pour réaliser une stabilisation en un point d'équilibre. Ceci fera l'objet d'étude dans le chapitre suivant.

Chapitre2:  
Commandes Non Linéaires Stabilisantes du  
Robot Mobile Unicycle

## Chapitre2: Commandes Non Linéaires Stabilisantes du Robot Mobile Unicycle

### 2.1. Problématique:

#### 2.1.1. Introduction :

La recherche d'algorithmes de planification et de stratégies de commande pour des robots mobiles non holonomes constitue aujourd'hui l'un des principaux axes de recherche de la robotique moderne. Plusieurs raisons contribuent à cet engouement. La première est que les véhicules à roues constituent de nos jours le moyen de transport individuel principal. Leur automatisation, précédemment limitée aux expérimentations en laboratoire, est maintenant envisagée pour des applications grand public (systèmes de transport urbain intelligent, etc.). Une autre raison plus technique tient au fait que les équations régissant le déplacement des systèmes non-holonomes revêtent un intérêt théorique particulier dans le domaine de l'automatique non linéaire.

Dans ce cadre, plusieurs travaux ont été engagés pour résoudre les problèmes de commande des robots mobiles. Ces travaux considèrent différentes difficultés que rencontre le concepteur de lois de commande automatique, pour essayer d'automatiser sa démarche en fonction des contraintes industrielles.

#### 2.1.2. Problème de commande des systèmes non-holonomes :

Les obstacles majeurs concernant la commande de systèmes non-holonomes sont :

1. le caractère **non linéaire** du système,
2. la **non commandabilité** de leur approximation linéaire, malgré que le système est commandable.
3. la non adéquation avec la condition nécessaire de Brockett [4] pour l'existence d'un retour d'état continu et indépendant du temps, stabilisant le système.

La difficulté relative du problème de la commande ne dépend pas seulement de la nature non-holonyme du système, mais aussi de l'objectif du contrôle tels que :

- la stabilisation autour d'une position,
- poursuite de trajectoires (suivi de chemins).

Dans ce qui suit, nous étudions le problème de stabilisation des systèmes non-holonomes (robot mobile unicycle dans notre cas) vers un état d'équilibre ainsi que les solutions existantes pour le résoudre.

### 2.1.2. Stabilisation des systèmes non-holonomes :

Nous nous intéressons dans ce travail au problème de la stabilisation en un point fixe, c'est-à-dire trouver une loi de commande telle que le point d'équilibre qui représente généralement l'origine, soit stable.

Les principales applications des techniques de stabilisation sont, à titre d'exemple, les manœuvres de parking.

- **Présentation du problème de stabilisation :**

Un aspect primordial du problème de la stabilisation des systèmes non-holonomes est lié à la non existence d'un retour d'état continu et indépendant du temps assurant une stabilisation. Ceci découle d'un résultat établi par Brockett qui est décrit ci-dessous.

**Théorème [4]** : Soit le système de commande ( $\dot{x} = f(x,u)$ ,  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$ ) avec  $f$  différentiable et  $(x,y)=(0,0)$  un point d'équilibre de ce système. Une condition nécessaire pour qu'il existe un retour d'état  $u$  continu tel que l'origine du système bouclé

$$\dot{x} = f(x, u(x))$$

Soit localement asymptotiquement stable, est la surjectivité locale de l'application  $(x,u) \rightarrow f(x,u)$  au voisinage de  $(x,u)=(0,0)$ . Plus précisément, pour tout voisinage  $\Omega$  de  $(0,0)$  dans  $\mathbb{R}^{n+m}$ , l'image par  $f$  de  $\Omega$  doit être un voisinage de 0 dans  $\mathbb{R}^n$ .

Ce résultat implique que de nombreux systèmes non linéaires ne sont pas asymptotiquement stabilisables par retour d'état continu malgré qu'ils soient commandables. C'est le cas des systèmes non holonomes et en particulier les robots mobiles à roues.

A titre d'exemple, nous démontrons ce résultat dans le cas d'un robot mobile de type unicycle donné par le modèle cinématique suivant :

$$\dot{x} = f(x, u)$$

$$\text{Avec : } x = (x_1, x_2, x_3), \quad u = (u_1, u_2), \quad f(x, u) = (u_1 \cos x_3, u_1 \sin x_3, u_2)^T.$$

Considérons un vecteur  $f$  dans  $\mathbb{R}^3$  donné par la forme :  $f(x, u) = (0, \delta, 0)$  avec  $\delta \neq 0$ . On constate immédiatement que cette équation n'admet pas la solution au voisinage de  $(x, u) = (0, 0)$ , puisque la première équation à satisfaire, c'est-à-dire  $u_1 \cos x_3 = 0$  implique  $u_1 = 0$ . Ainsi, il n'existe pas de solution à la deuxième équation si  $\delta$  est différent de zéro. Ceci montre bien que  $f$  n'est pas localement surjective au voisinage de  $(x, u) = (0, 0)$ .

Il est bien connu que l'approximation linéaire des systèmes non holonomes n'est pas commandable. En effet, en prenant le même type de robot mobile, c'est-à-dire celui de l'unicycle, nous remarquons que le linéarisé de ces équations n'est pas commandable. Ainsi les méthodes de commande linéaire ne peuvent pas être utilisées pour stabiliser asymptotiquement (localement) le point d'équilibre.

En conséquence, le robot unicycle ne peut être stabilisé asymptotiquement par un retour d'état continu invariant dans le temps.

Afin de résoudre ce problème de stabilisation, plusieurs approches ont été proposées, à savoir celles basées sur les techniques de commande par retour d'état continu non stationnaire ou discontinu invariant.

Ces approches sont classifiées en trois catégories :

1. Stabilisation par retour d'état discontinu (eg, invariant manifold approach)
2. Stabilisation par retour d'état continu variant dans le temps (eg, Backstepping)
3. Stabilisation par retour d'état hybride

Résoudre le problème de stabilisation consiste à trouver un contrôleur feedback adéquat permettant au robot mobile d'atteindre une configuration statique à partir de n'importe quelle configuration initiale.

Dans ce qui suit, nous élaborerons, pour chaque classe de commande, une technique stabilisante.

Avant d'introduire les différentes approches de commande stabilisante, nous allons commencer par donner une représentation ou un changement de coordonnées du modèle général représentatif du robot mobile. Une forme qui facilitera l'étude des différentes commandes et leur application au robot.

Pour chacune de ces méthodes, nous présentons des résultats de simulation corroborant nos études théoriques.

## **2.2. Commande par retour d'état discontinu (invariant manifold approach)[3] :**

Dans la plupart des cas, l'application d'une loi de commande non linéaire est devenue complexe et onéreuse lorsqu'il s'agit de contrôler un système non-holonome. Pour ce faire, les automaticiens ont réfléchi à simplifier le modèle du système en lui donnant une forme beaucoup plus simple à contrôler appelée « forme chaînée ». L'avantage de cette nouvelle représentation est de faciliter le calcul d'une loi de commande applicable à n'importe quel ordre du système (jusqu'à n-dimension).

Dans ce cas, pour stabiliser un robot mobile, la première étape consiste à transformer son modèle cinématique en forme chaînée d'ordre 3. Cette dernière est exploitée pour élaborer une loi de commande stabilisante.

La loi de commande établie est composée de deux principaux éléments : un retour d'état linéaire et un retour d'état discontinu.

### 2.2.1. Présentation de la forme chaînée :

C'est l'une des manières qui permet de mettre un système sous une certaine forme canonique qui facilite l'analyse et le calcul des contrôles. L'allure générale de cette forme est donné par:

$$\begin{cases} \dot{z}_1 = u_1 \\ \dot{z}_2 = u_2 \\ \dot{z}_3 = z_2 u_1 \\ \vdots \\ \dot{z}_n = z_{n-1} u_1 \end{cases} \quad (2.1)$$

Où  $z = (z_1, z_2, \dots, z_n)^T$  représente le vecteur d'état, et  $u = (u_1, u_2)^T$  indique le vecteur d'entrée du système.

Cette forme est appelée **forme chaînée simple** (2, n) où 2 représente le nombre de contrôle et  $n$  le nombre d'états.

### 2.2.2. Algorithme générale de stabilisation :

Soit un système non-holonyme dont la forme chaînée correspondante est donnée par le système (2.1). Stabiliser le système non-holonyme revient à stabiliser sa forme chaînée correspondante.

La stabilisation de notre système, consiste alors à appliquer un contrôleur par retour d'état discontinu invariant  $(u_1, u_2)^T$  qui stabilise le système modélisé sous la forme (2.1). Dans ce cas, les deux commandes  $u_1$  et  $u_2$  peuvent s'écrire :

$$\begin{aligned} u_1(z) &= v_1(z) \\ u_2(z) &= v_2(z) + w_2(z) \end{aligned} \quad (2.2)$$

Pour calculer les différents termes de  $u_1(z)$  et  $u_2(z)$  nous procédons de la manière suivante :

Posons au départ  $w_2(z)=0$ , et considérant le retour d'état linéaire  $(v_1(z), v_2(z))^T$  :

$$\begin{aligned} u_1(z) &= v_1(z) = -k_1 z_1 \\ u_2(z) &= v_2(z) = -k_2 z_1 - k_3 z_2 \end{aligned} \quad (2.3)$$

Où  $k_2 \in \mathbb{R}$ ,  $k_1, k_3$  sont strictement positifs et  $k_1 \neq k_3$ .

Le système en boucle fermée (2.1)-(2.3) peut être intégré, étape par étape, pour obtenir :

$$\begin{aligned}
 z_1(t) &= z_{10} e^{(-k_1 t)}. \\
 z_2(t) &= \left( z_{20} - \frac{k_2 z_{10}}{k_a} \right) e^{(-k_3 t)} + \frac{k_2 z_{10}}{k_a} e^{(-k_1 t)}. \\
 z_3(t) &= \frac{k_1}{k_b} z_{10} \left( z_{20} - \frac{k_2 z_{10}}{k_a} \right) e^{(-k_b t)} + \frac{k_2}{2k_a} z_{10}^2 e^{(-2k_1 t)} + S_3(z_0). \\
 z_n(t) &= \frac{k^{n-2} z^{n-2}}{k_b(k_b+k_1)\dots(k_b+(n-3)k_1)} \left( z_{10} - \frac{k_2 z_{10}}{k_b} \right) e^{-(k_b+(n-3)k_1)t} + \\
 &\frac{k_2 z_{10}^{n-1}}{(n-1)!k_b} e^{-(n-1)k_1 t} + \sum_{i=1}^{n-3} \frac{z_{10}^i}{i!} S_{n-1}(z_0) e^{(-ik_1 t)} + S_n(z_0). \tag{2.4}
 \end{aligned}$$

Avec:  $K_a = k_1 - k_3$ ,  $K_b = k_1 + k_3$ . et  $S_3(z_0), \dots, S_n(z_0)$  sont des constantes d'intégration qui peuvent être déterminées, à  $t=0$ , comme une fonction des conditions initiales  $z_{10}, z_{20} \dots z_{n0}$ .

Prenons  $S_j(z_0) = S_j$  ( $3 \leq j \leq n$ ). Remplacer  $t = 0$  dans (2.4) et déterminer  $S_j(z_0)$  et ensuite substituer  $z_0$  par  $z$ .

Les nouvelles expressions de  $S_j$  sont données comme suit :

$$\begin{aligned}
 S_3(z) &= z_3 - \frac{k_1}{k_b} z_1 z_2 + \frac{k_2}{2k_b} z_1^2. \\
 S_4(z) &= z_4 - z_1 z_3 + \frac{k_1}{(k_1+k_b)} z_1^2 z_2 - \frac{k_2}{3(k_1+k_b)} z_1^3 \\
 S_n(z) &= z_n + \sum_{i=1}^{n-3} \frac{(-1)^i z_1^i z_{n-i}}{i!} + \frac{(-1)k_1 z_1^{n-2}}{(k_b+(n-3)k_1)(n-3)!} + \frac{(-1)^{n-1} k_2 z_1^{n-1}}{(k_b+(n-3)k_1)(n-3)!(n-1)} \tag{2.5}
 \end{aligned}$$

• **Définition :**

La surface  $M = \{x \in \mathbb{R}^n : \varphi(x) = 0\}$  est dite invariante par la commande  $u$  du système  $\dot{x} = f(x, u)$  si pour n'importe quel  $x_0 \in M$  tous les trajectoires  $x(t), t \geq 0$  du système sous la commande  $u$  resteront dans cette surface.

**Préposition [3] :**

La surface  $M_s = \{z \in \mathbb{R}^n : S_j(z) = 0\}$  est une surface invariante pour le système (2.1)-(2.3).

A partir de (2.4), nous constatons que  $(z_1, z_2, z_3, z_4 \dots z_n)$  tendent vers  $(0, 0, S_3(z_0), S_4(z_0), \dots, S_n(z_0))$  quand  $t$  tend vers l'infini.

Dans ce cas, si nous prenons des conditions initiales tel que  $S_j(z_0) = 0$  ( $3 \leq j \leq n$ ), c-à-d appartiennent à  $M_s$  tous les états tendent, alors, vers l'origine.

En résumé, la stabilisation de la forme chaînée (2.1) est équivalent à la stabilisation de  $(0, 0, S_3(z_0), S_4(z_0), \dots, S_n(z_0))$  en prenant en compte le paramètre  $w_2(z)$ .



### 2.2.3. Application de l'algorithme de stabilisation au robot mobile :

Le robot mobile considéré est un robot de type unicycle avec deux roues avant motrices et une roue arrière stabilisante comme le montre la figure ci-dessous :

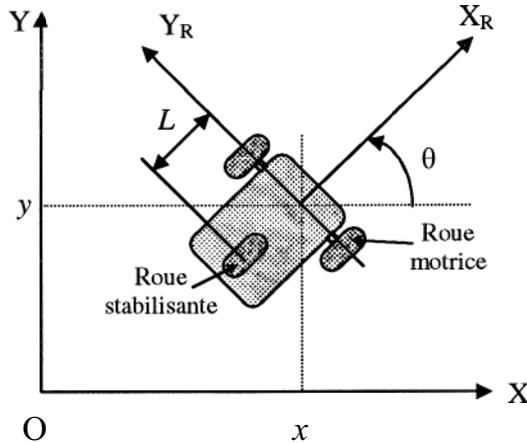


Figure 2.1. Robot mobile de type unicycle

La commande de ce type de véhicule est établie par l'intermédiaire d'une vitesse linéaire d'un point M notée «  $v$  » et d'une vitesse angulaire dans la direction des roues avant notée «  $w$  ».

Le robot roule sans glisser sur un sol horizontal.

La configuration de ce robot est décrite par le vecteur  $(x, y)$ , où  $(x, y)$  sont les coordonnées du point M localisées à la mi-distance des deux roues avants.

$\theta$  représente l'orientation du véhicule, entre l'axe du robot et l'axe des abscisses  $x$ .

Le modèle cinématique du robot est donné donc par l'expression suivante :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega \quad (2.7)$$

Ici, l'objectif est de commander le robot afin de le ramener à une position d'équilibre en suivant une trajectoire donnée.

Pour appliquer la technique de la commande stabilisante au robot concerné, nous transformons le système (2.7) en forme chaînée d'ordre 3.

La forme chaînée, représentant le modèle du robot unicycle, est donnée par l'expression suivante :

$$\begin{cases} \dot{z}_1 = u_1 \\ \dot{z}_2 = u_2 \\ \dot{z}_3 = z_2 u_1 \end{cases} \quad (2.8)$$

L'expression (2.8) est obtenue en effectuant le changement de variable suivant :

$$\begin{cases} z_1 = x \\ z_2 = \tan \theta \\ z_3 = y \end{cases} \quad (2.9)$$

La relation donnant les entrées de contrôle «  $v$ ,  $w$  » en fonction des nouveaux contrôles «  $u_1, u_2$  » est la suivante :

$$v = \frac{u_1}{\cos\theta}$$

$$w = u_2 \cos^2(\theta) \quad (2.10)$$

Pour ramener le robot à une position d'équilibre, nous allons concevoir un contrôleur avec un retour d'état discontinu invariant. Pour ce faire, nous utilisons l'algorithme de stabilisation présenté dans la section 2.2.2

- **Calcul des termes de  $u_1, u_2$**  : on a vu que les commandes s'expriment comme suit :

$$\begin{aligned} u_1(z) &= v_1(z) \\ u_2(z) &= v_2(z) + w_2(z) \end{aligned} \quad (2.11)$$

avec :

$$v_1(z) = -k_1 z_1$$

$$v_2(z) = -k_2 z_1 - k_3 z_2$$

$$w_2(z) = C^T Q(z_1) S$$

l'ordre du système est de 3, alors :

$$C = c_3;$$

$$Q(z_1) = \frac{1}{z_1};$$

$$S = S_3;$$

$$D'où w_2(z) = c_3 \frac{1}{z_1} S_3.$$

La valeur du  $C$  est choisi de tel sorte que la valeur de la matrice  $(A + B.C)^T$  possède des valeurs propres à parties réelles négatives.

Dans notre cas,

$$A = k_1$$

$$B = b_3 = \frac{-k_1}{k_b},$$

$$\text{Alors : } (A + BC)^T = k_1 - \frac{k_1}{K_b} c$$

Cette matrice possède des valeurs propres à parties réelles négatives est équivalent à dire que :

$$k_1 \left(1 - \frac{C}{K_b}\right) < 0 \Rightarrow \left(1 - \frac{C}{K_b}\right) < 0 \Rightarrow c_3 > K_b$$

(Rappelons que  $K_b = k_1 + k_3$ )

### 2.2.4. Simulations :

A partir de l'étude réalisée précédemment, nous illustrons le comportement du robot (observer les états  $x$ ,  $y$ ,  $\theta$ ) en appliquant la commande stabilisante donnée par les expressions (2.11). Aussi, nous nous intéressons à observer le comportement des commandes en rotation et en translation pendant le mouvement du robot.

Nous testons les performances du contrôleur à différentes conditions initiales( $x_0, y_0, \theta_0$ ).

Nous fixons d'abord la valeur des différents termes de chaque commande.

Les valeurs des  $k_i$  ( $1 < i < 3$ ) sont données comme suit:  $k_1 = 1$ ,  $k_2 = 0$ ,  $k_3 = 3$ .

Nous pouvons, de ce fait, déduire les valeurs de  $K_a$ ,  $K_b$  et  $C$  qui valent :

$c_3 > K_b \Rightarrow c_3 > 4$  (on prend  $c_3 = 10$ ).

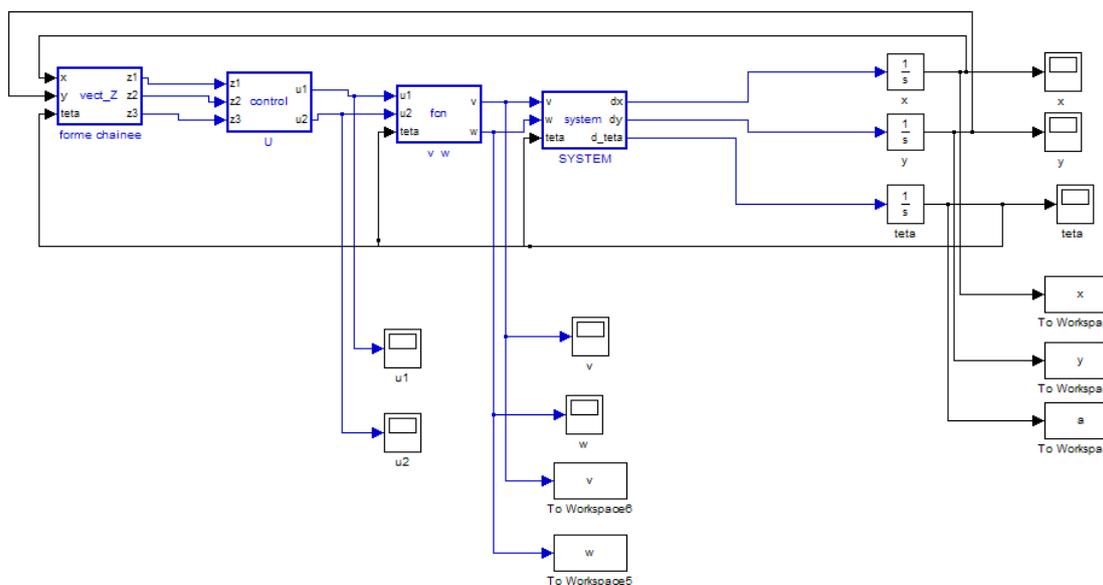
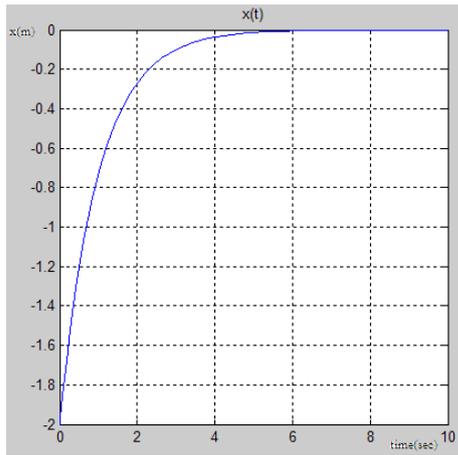


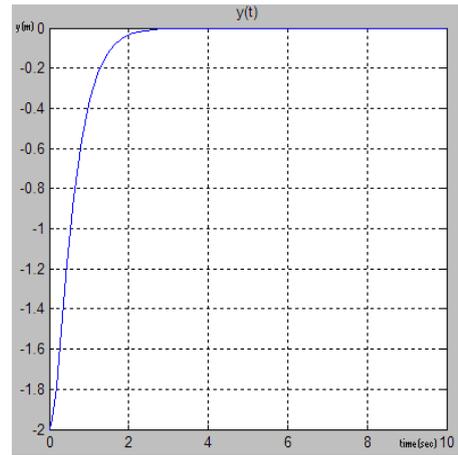
Figure 2.2: schéma bloc Simulink de la commande

Dans le cadres des simulations les indices (a), (b), (c) font référence aux allures des coordonnées :  $x(t)$ ,  $y(t)$ ,  $\theta(t)$  respectivement.

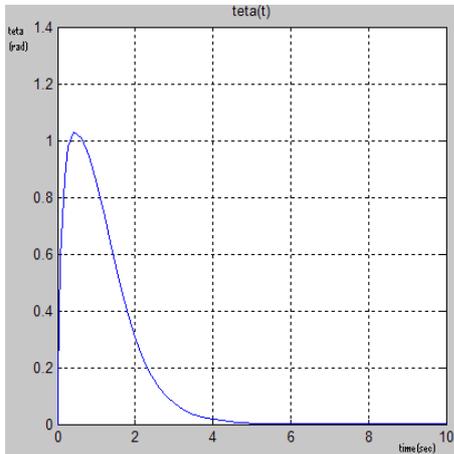
Egalement l'indice (d) correspond aux signaux de commandes.



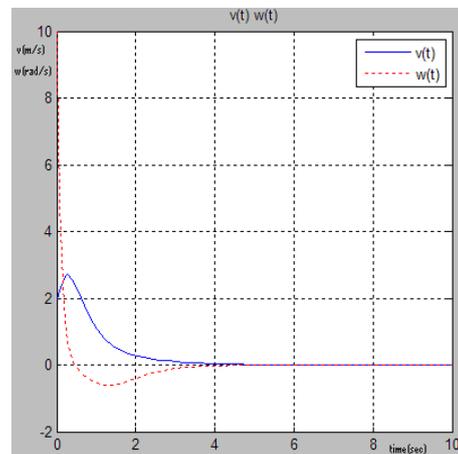
(a)



(b)



(c)



(d)

Figure 2.3: Résultats de simulation pour l'état initial  $(x_0, y_0, \theta_0) = (-2, -2, 0)$

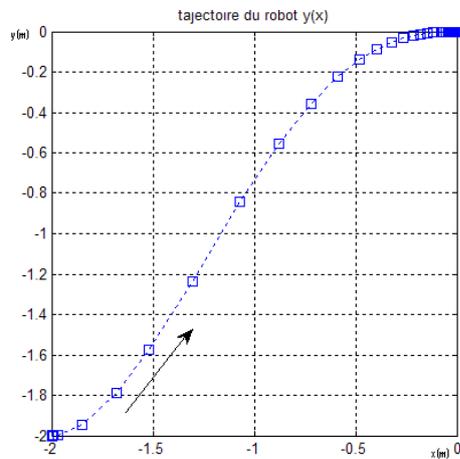


Figure 2.4 : trajectoire du robot  $y(x)$

Une autre condition initiale est utilisée  $(x_0, y_0, \theta_0) = (2, -2, 0)$ . les allures des coordonnées du robot ainsi les signaux de commande sont les suivants :

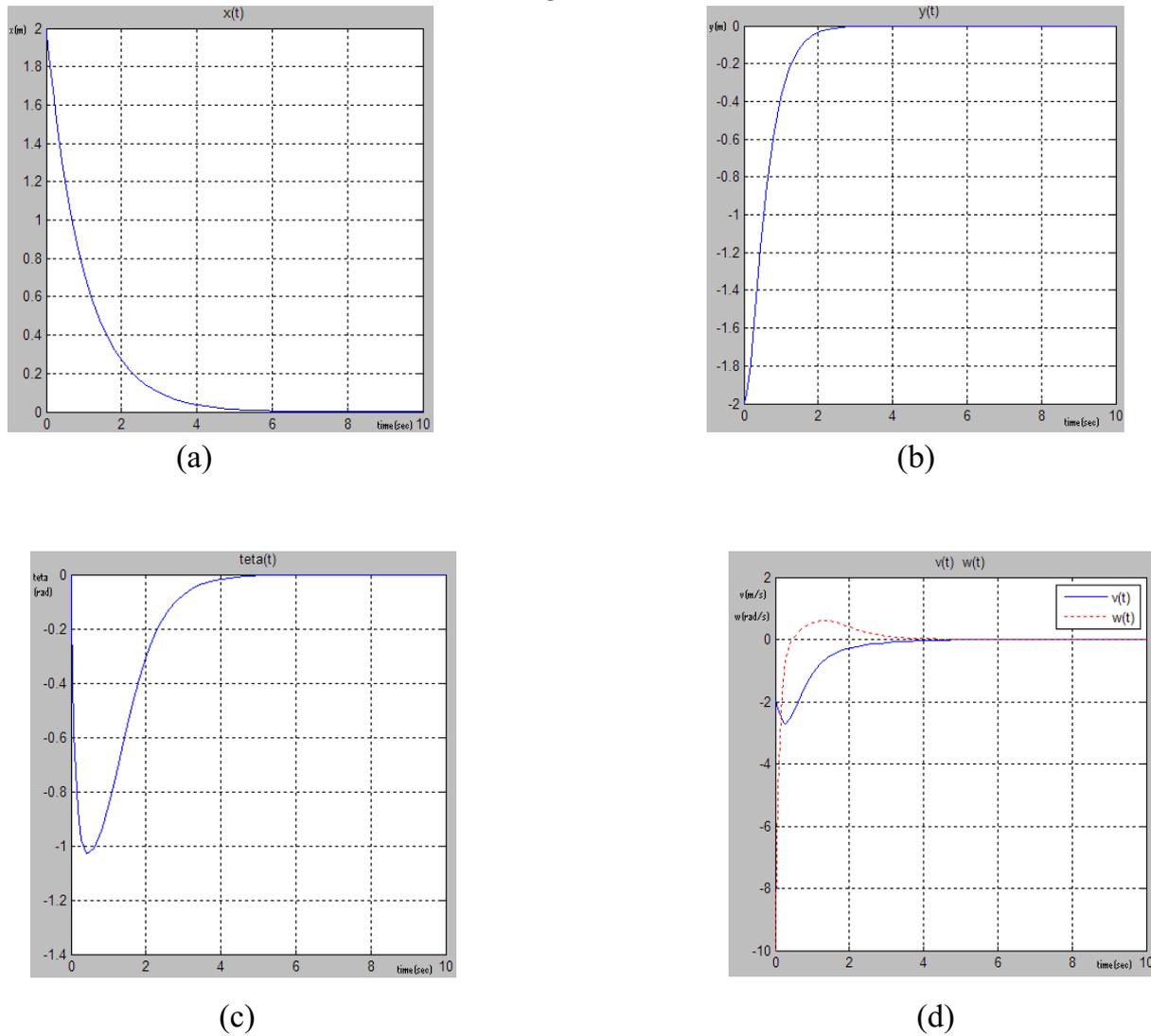


Figure 2.5 : Résultats de simulation pour l'état initial  $(x_0, y_0, \theta_0) = (2, -2, 0)$

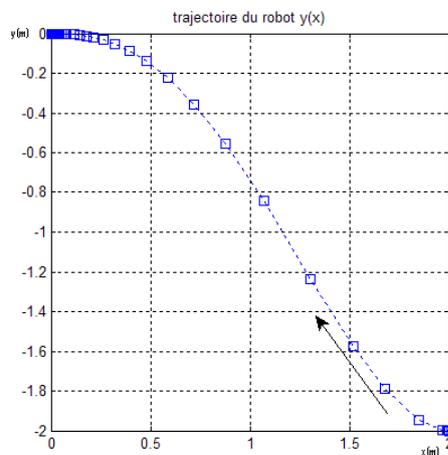


Figure 2.6: trajectoire du robot  $y(x)$

### 2.3. Commande stabilisante en coordonnées polaires [4] :

Parmi les techniques de commande appliquées sur le robot mobile unicycle on trouve la stabilisation en coordonnées polaires.

Le contrôleur, obtenu par la méthode de Lyapounov, appartient à la catégorie des contrôleurs discontinus.

#### 2.3.1. Transformation en coordonnées polaires :

Considérons le changement de variable aux nouvelles coordonnées  $(\rho, \gamma, \delta)$  défini comme suit (selon la figure ci-dessous)

$$\rho = \sqrt{x^2 + y^2}$$

$$\gamma = \text{ATAN2}(y, x) - \theta + \pi$$

$$\delta = \gamma + \theta \tag{2.12}$$

Où : ATAN2 est la fonction tangente inverse du quatrième quadrant.

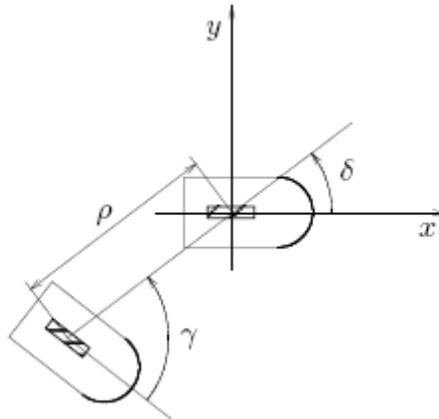


Figure 2.7 : définition des coordonnées polaires pour l'unicycle.

Dans ces coordonnées, les équations de l'unicycle deviennent :

$$\begin{aligned}\dot{\rho} &= \cos \gamma * v \\ \dot{\gamma} &= \frac{\sin \gamma}{\rho} v - w \\ \dot{\delta} &= \frac{\sin \gamma}{\rho} v\end{aligned}\tag{2.13}$$

- **Théorème[4] :**

Considérons les lois de commande suivantes :

$$v = k_1 \rho \cos \gamma\tag{2.14}$$

$$w = k_2 \gamma + k_1 \frac{\sin \gamma \cos \gamma}{\gamma} (\gamma + k_3 \delta)\tag{2.15}$$

Avec  $k_1, k_2$  deux constantes positives.

Le système en boucle fermée est ramené globalement asymptotiquement à la position  $(p, \gamma, \delta) = (0,0,0)$

La preuve de ce théorème est basée sur l'application du théorème de Lasalle en utilisant la fonction de Lyapunov suivante :

$$V = 1/2 (\rho^2 + \gamma^2 + k_3 \delta^2)$$

Et sa dérivée temporelle en boucle fermée :

$$\dot{V} = -k_1 \cos^2 \gamma \rho^2 - k_2 \gamma^2 \leq 0$$

### 2.3.2. Simulations :

Nous allons montrer le comportement du robot en appliquant la commande stabilisante  $(v, w)$  donnée précédemment .nous allons tester les performances du contrôleur à différentes conditions initiales .

Les valeurs numériques des gains sont choisies comme suit :  $k_1 = 1, k_2 = 3, k_3 = 2$

Aussi, nous nous intéressons à observer le comportement des commandes en translation et en rotation pendant le mouvement du robot.

Les résultats obtenus sont présentés ci-dessous :

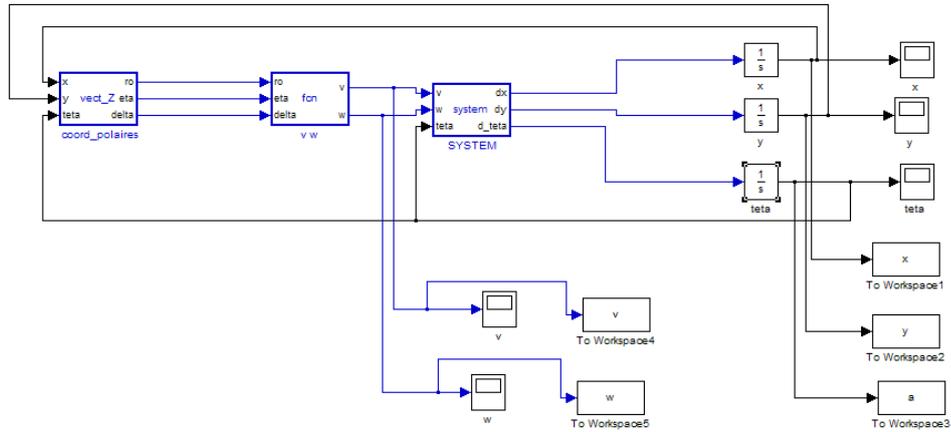


Figure 2.8:schéma bloc Simulink de la commande en coordonnées polaires.

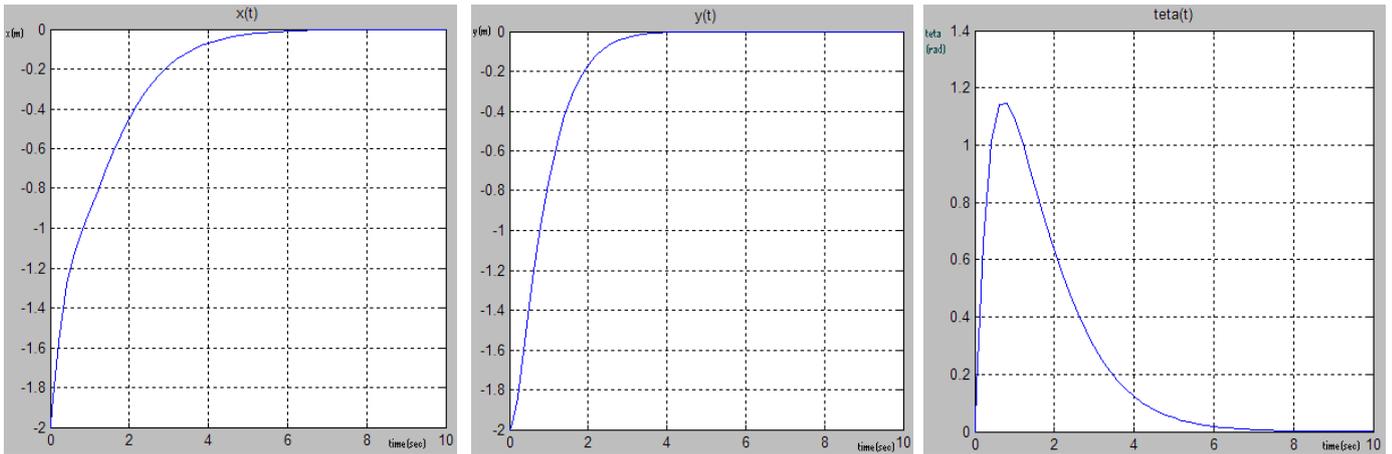


Figure 2.9 : Evolution des variables d'état (positon initiale (-2,-2,0))

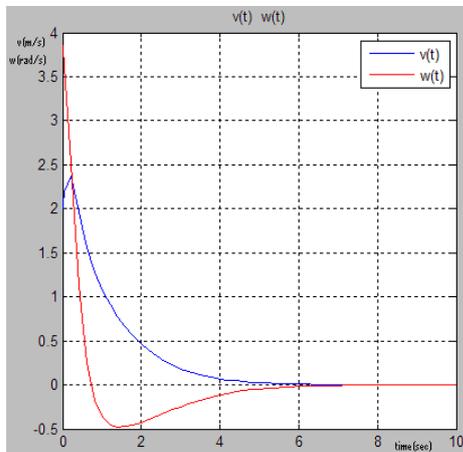


Figure 2.10:Evolution des commandes v,w

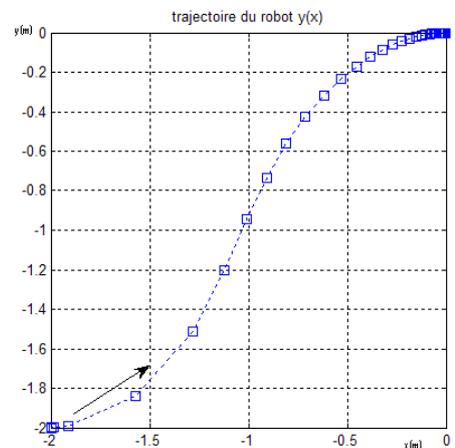


Figure 2.11:Trajectoire du robot

Pour la position initiale  $(x_0, y_0, \theta_0) = (0, -2, 0)$  les résultats de simulation sont donnés par :

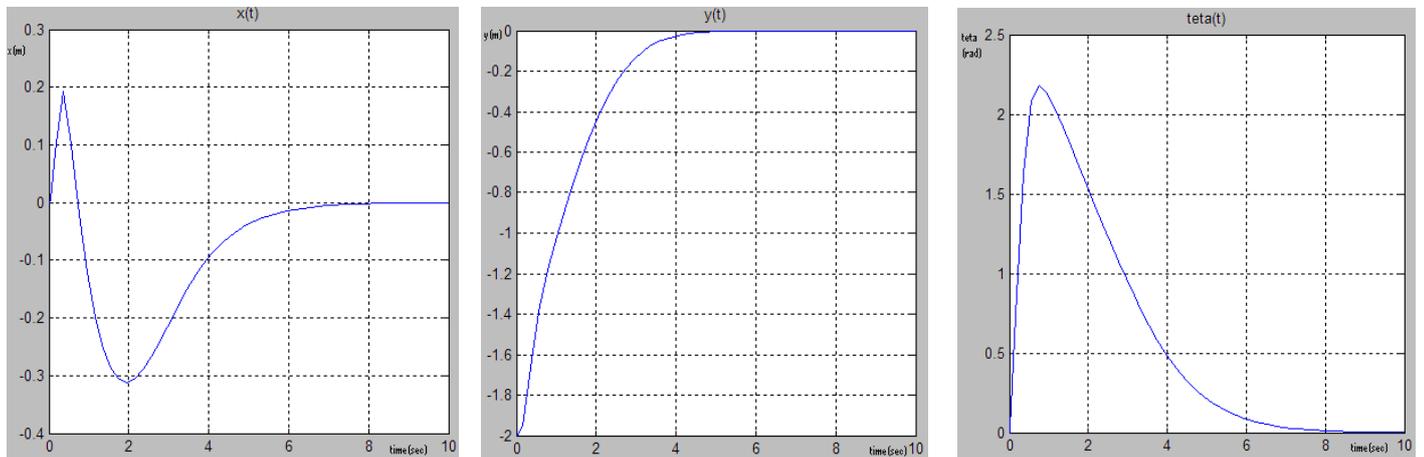


Figure 2.12: Evolution des variables d'état (position initiale  $(0, -2, 0)$ )

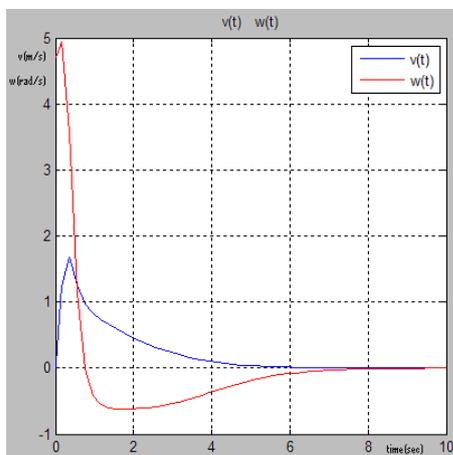


Figure 2.13 : évolution des commandes  $v, w$

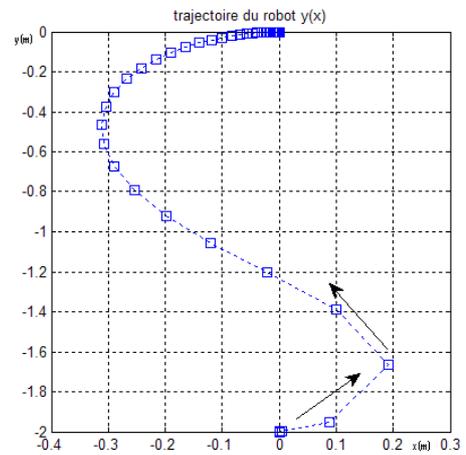


Figure 2.14 : Trajectoire du robot

Une autre condition initiale est utilisée  $(x_0, y_0, \theta_0) = (2, -2, 0)$ . les allures des coordonnées du robot ainsi les signaux de commande sont les suivants :

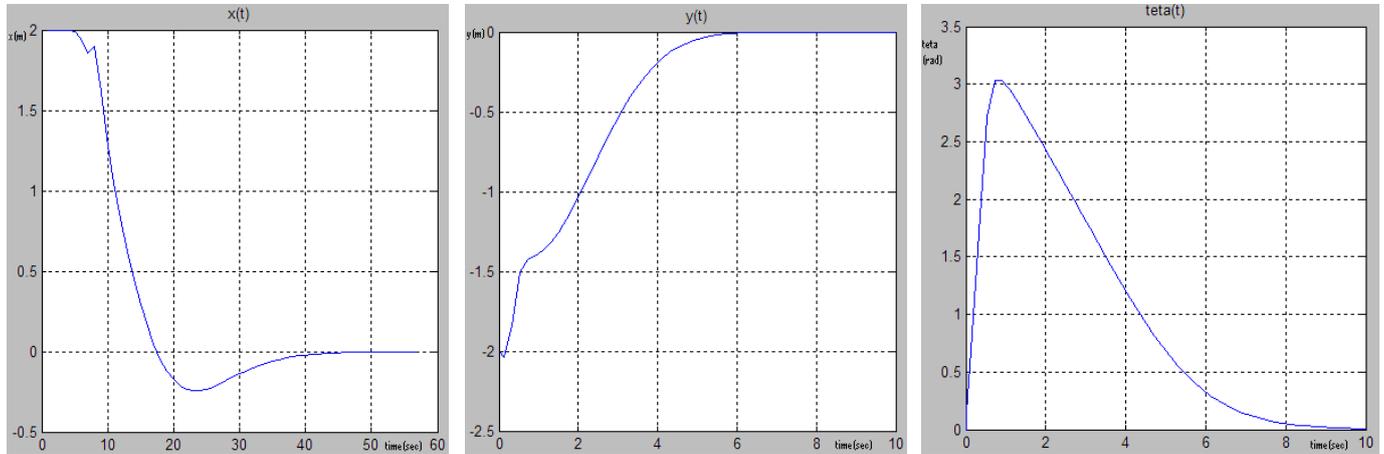


Figure 2.15: Evolution des variables d'état (position initiale (2,-2,0))

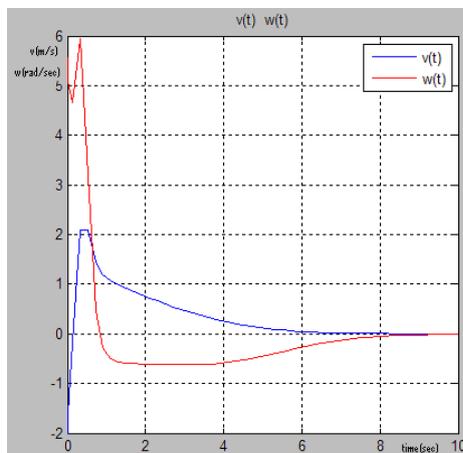


Figure 2.16: évolution des commandes v, w

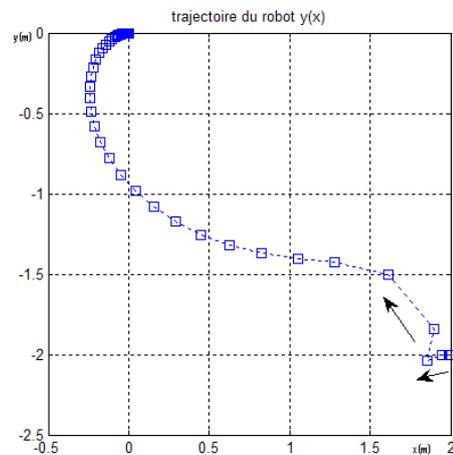


Figure 2.17: Trajectoire du robot

## 2.4. Stabilisation du robot par technique du backstepping :

- **Introduction :**

Dans les années 90, une approche très intéressante a été développée pour la détermination systématique d'une fonction de Lyapunov candidate dans le cadre stabilisation des systèmes, en particulier, systèmes en cascade. Cette approche est connue sous le nom « Méthodologie de backstepping ».

Elle est basée sur la méthode de Lyapunov qui utilise la notion des fonctions définies positives et négatives.

### 2.3.1. Cadre théorique :

Pour introduire la notion de stabilité par la méthode de Lyapunov les définitions suivantes sont nécessaires :

#### Définition1 :

Soit  $V(x)$  une fonction scalaire d'argument  $X \in \mathbb{R}^n$  (qui est généralement le vecteur d'état)  $V(x)$  est une fonction définie positive FDP (respectivement, définie négative FDN), sur un espace  $\mathbb{E} \subseteq \mathbb{R}^n$  contenant l'origine, si seulement si :

$$\forall x \in \mathbb{E} - \{0\}: V(x) > 0 \text{ (resp, } V(x) < 0)$$

et  $V(x) = 0 \Leftrightarrow X = 0$

#### Définition2 :

Soit le système non linéaire  $\dot{x} = f(x, u)$

la solution  $(x^*, u^*)$  de l'équation  $f(x, u) = 0$  est un point d'équilibre.

Pour ce type de systèmes, il est appelé point de fonctionnement.

La stabilité des points d'équilibre se base sur le théorème suivant :

#### Théorème :

Soit le système non linéaire  $\dot{x} = f(x, u)$  ayant  $(x^*, u^*)$  comme point d'équilibre.

Considérons une fonction scalaire  $V(x)$  FDP sur un espace  $E$  contenant le point d'équilibre ,

Si  $\dot{V}(x)$  est FDN sur  $E$  , alors ce point d'équilibre est asymptotiquement stable.

De plus si  $E = \mathbb{R}^n$  , il est globalement asymptotiquement stable.

**Définition3 :**

Un point d'équilibre est asymptotiquement stable si seulement si :

- il est L- stable :

$$\exists \delta > 0: \|x(0)\| < \delta \Rightarrow \|x(t)\| < R \quad \forall t > 0$$

Où R est une constante définie la limite des signaux  $x(t)$  tolérés par le système

- $\lim_{t \rightarrow \infty} \|x(t)\| = 0$  lorsque  $t \rightarrow \infty$

**❖ Technique du backstepping :**

Soit un système non linéaire  $\dot{x} = f(x, u)$  le problème de stabilisation du vecteur d'état  $x$  revient à chercher le couple  $(V(x), u^*(x))$

Où  $V(x)$  est une fonction candidate de Lyapunov  $u^*(x)$  est un retour d'état non linéaire déterminé de telle sorte que  $\dot{V}(x)$  soit une FDN sur  $\mathbb{R}^n$ .

La méthode du Backstepping est utilisée pour la recherche de  $V(x)$  adéquate.

L'exemple ci-dessous illustre les démarches suivies dans la technique du Backstepping :

Soit le système :

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

$$\dot{x}_3 = u$$

On peut observer que le signal de commande intervient uniquement sur la dernière équation.

La forme de ce système montre qu'on peut piloter  $x_1$  par  $x_2$  et  $x_2$  par  $x_3$  ( $x_3$  comme une commande fictive).

Première étape : stabiliser  $x_1$  par  $x_2$  par la méthode de Lyapunov, i.e, chercher  $V_1(x)$  FDP sur  $\mathbb{R}^3$  et une loi  $x_2^* = \phi_2(x)$  tel que  $\dot{V}_1(x)$  est FDN le choix le plus simple :

$$V_1(x) = \frac{1}{2} x_1^2$$

$$\dot{V}_1(x) = \dot{x}_1 x_1 = x_2 x_1$$

Si  $x_2^* = \phi_2(x) = -x_1$  alors  $\dot{V}_1(x) = -x_1^2$  FDN sur  $\mathbb{R}^3$

-La deuxième étape est similaire à la précédente mais cette fois le but est de stabiliser  $x_2$  autour  $\phi_2$  par  $x_3$  :

Soit  $V_1(x) = V_1(x) + \frac{1}{2}(x_2 - \phi)^2$

Après le calcul de  $\dot{V}_2$ , on cherche  $x_3^* = \phi_3(x)$  de telle sorte que  $\dot{V}_2$  soit FDN sur  $\mathbb{R}^3$

-La dernière étape consiste à assurer la stabilité de  $x_3$  autour de  $\phi_3$  par une loi  $u^*(x)$

La forme finale de la fonction candidate de Lyapunov qu'on cherche est :

$$V(x) = V_1(x) + V_2(x) + \frac{1}{2}(x_3 - \phi_3)^2$$

La loi  $u^*(x)$  est établie pour rendre  $\dot{V}(x)$  une fonction définie négative FDN sur  $\mathbb{R}^3$

### 2.3.2. Stabilisation du robot :

Considérons le modèle cinématique de l'unicycle en forme chaînée (voire section 2.2):

$$\begin{aligned}\dot{z}_1 &= u_1 \\ \dot{z}_2 &= u_2 \\ \dot{z}_3 &= z_2 u_1\end{aligned}\tag{2.16}$$

D'une simple analyse ; il est clair qu'on peut stabiliser d'abord l'état  $z_1$  par  $u_1$  ensuite, chercher une loi  $z_2 = \phi_2$  afin de stabiliser l'état  $z_3$  .

Enfin, la commande  $u_2$  est établie d'une façon à assurer la stabilité de l'état  $z_2$  autour de  $\phi_2$ :

- Première étape : soit la fonction de Lyapunov suivante :

$$V_1(z_1) = \frac{1}{2}z_1^2 \text{ c'est une fonction définie positive.}$$

Sa dérivée par rapport au temps est :

$$\dot{V}_1(z_1) = z_1 \dot{z}_1 = z_1 u_1$$

$$\text{Si : } u_1 = -k_1 z_1; k_1 > 0 \text{ alors : } \dot{V}_1(z_1) = -k_1 z_1^2 \leq 0$$

Elle devient une fonction définie négative, donc selon la théorie de Lyapunov : la commande  $u_1$  qui garantit la stabilité de  $z_1$  est donnée par :

$$u_1 = -k_1 z_1, \quad k_1 > 0\tag{2.17}$$

- Deuxième étape : considérons une nouvelle fonction de Lyapunov :

$V_2(z_3) = \frac{1}{2}z_3^2$  Fonction définie positive ;sa dérivée est:

$$\dot{V}_2(z_3) = z_3\dot{z}_3 = z_3(z_2u_1)$$

Or on a imposé  $u_1 = -k_1z_1$ , donc  $\dot{V}_2(z_3) = z_3(-k_1z_1)z_2$

Posons  $z_2 = \phi_2 = \frac{k_2z_3}{k_1z_1}$ ,  $k_2 > 0$

Alors  $\dot{V}_2(z_3) = -k_2z_3^2 \leq 0$  (définie négative)

On dit que  $\phi_2$  assure la stabilité de  $z_3$

- Troisième étape :

$$\text{Soit : } V_3(z) = \frac{1}{2}z_3^2 + \frac{1}{2}(z_2 - \phi_2)^2$$

C'est une fonction de Lyapunov candidate définie positive.

Calculons sa dérivée :

$$\dot{V}_3(z) = z_3\dot{z}_3 + (z_2 - \phi_2)(z_2 - \dot{\phi}_2)$$

Or :  $\dot{z}_3 = u_1z_2 = u_1[(z_2 - \phi_2) + \phi_2]$

$$u_1 = -k_1z_1, \quad \text{et} \quad \phi_2 = \frac{k_2z_3}{k_1z_1}$$

Ce qui nous donne :  $z_3\dot{z}_3 = -k_2z_3^2 + (z_2 - \phi_2)(u_1z_3)$

Donc :  $\dot{V}_3(z) = -k_2z_3^2 + (z_2 - \phi_2)[u_2 - \dot{\phi}_2 + u_1z_3]$

Si :  $u_2 = \dot{\phi}_2 - u_1z_3 - k_3(z_2 - \phi_2)$  (2.18)

Alors  $\dot{V}_3(z) = -k_2z_3^2 - k_3(z_2 - \phi_2)^2 < 0$  Une fonction définie négative

A la fin de cette étape, on a abouti, par la technique du Backstepping, à deux lois de commande (2.19) assurant la stabilité du système (2.16).

après simplifications on a :

$$u_1 = -k_1z_1$$

$$u_2 = -k_2z_2 + k_2\frac{z_3}{z_1} + k_1z_1z_3 - k_3\left(z_2 - \frac{k_2z_3}{k_1z_1}\right). \text{ Avec } k_i > 0, i = 1,2,3. \quad (2.19)$$

### 2.3.3. Simulations :

A partir de l'étude réalisée précédemment, nous illustrons le comportement du robot (observer les états  $x$ ,  $y$ ,  $\theta$ ) en appliquant la commande stabilisante donnée par les expressions (2.19) . Aussi, nous nous intéressons à observer le comportement des commandes en rotation et en translation pendant le mouvement du robot.

différentes conditions initiales  $(x_0, y_0, \theta_0)$  sont considérées.

nous fixons d'abord la valeur des différents termes de chaque commande.

Les valeurs des  $k_i$  ( $1 < i < 3$ )  $k_i = 1$ .

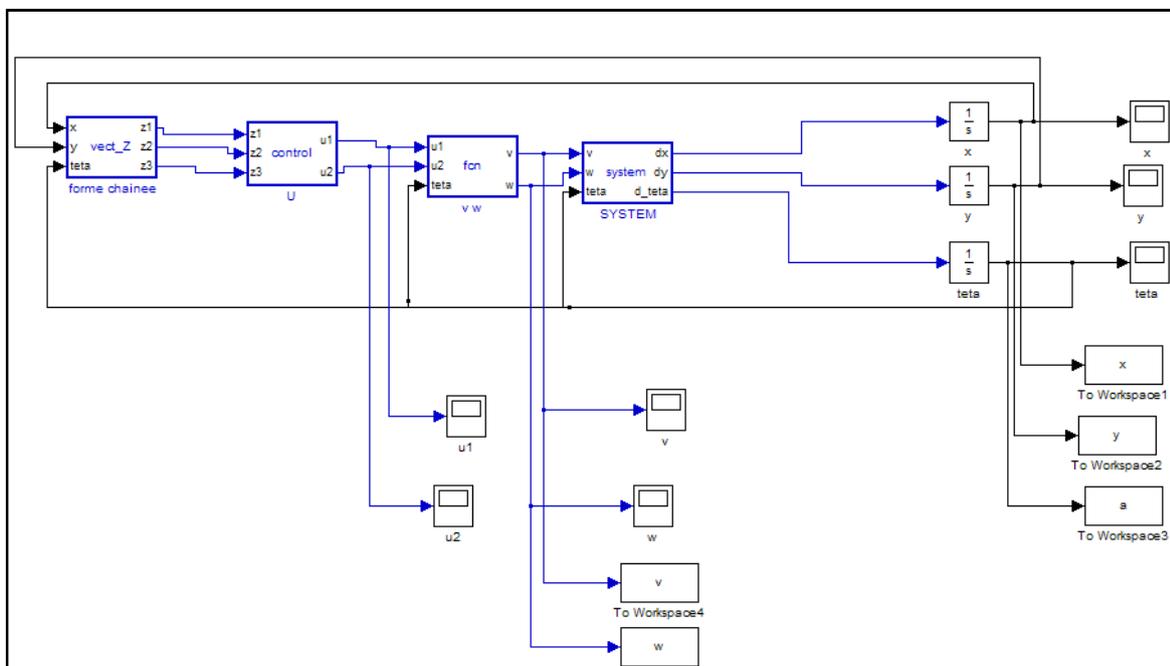


Figure 2.18: Schéma bloc Simulink utilisé.

Dans le cadres des simulations les indices (a),(b),(c) font référence aux allures des coordonnées :  $x(t)$ ,  $y(t)$ ,  $\theta(t)$  respectivement.

Egalement l'indice (d) correspond aux signaux de commandes.

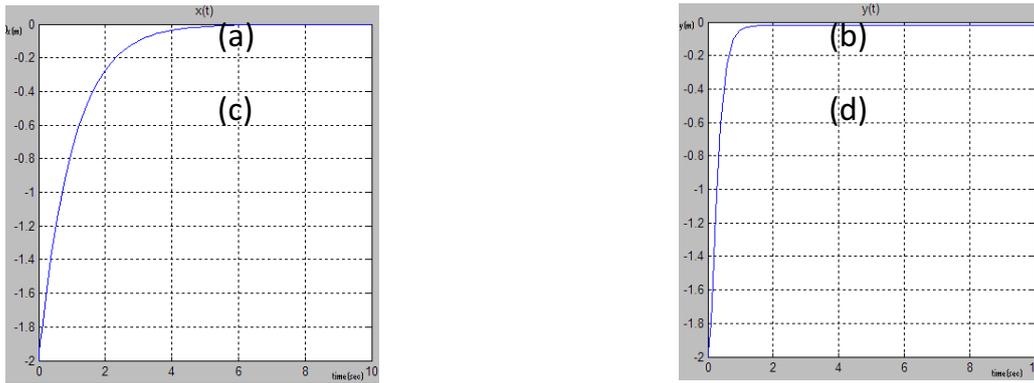


Figure 2.19: Résultats de simulation pour l'état initial  $(x_0, y_0, \theta_0) = (-2, -2, 0)$

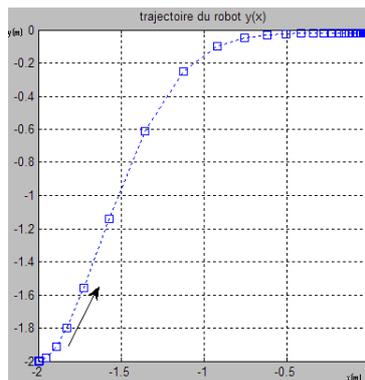
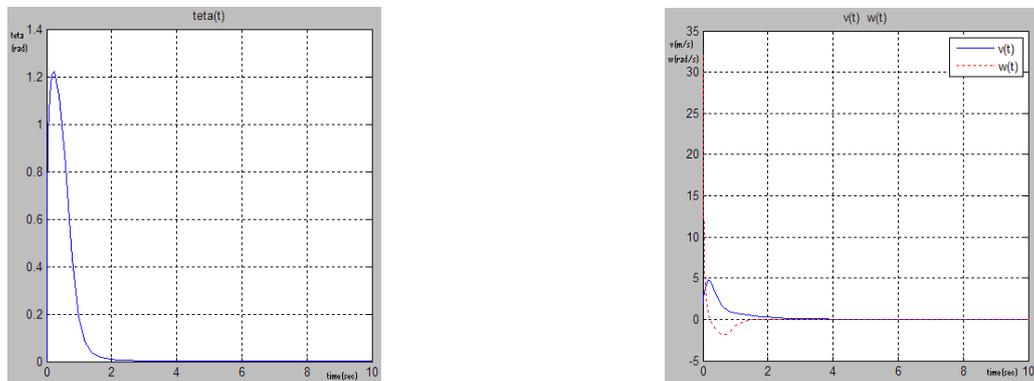


Figure 2.20 : Trajectoire du robot

Une autre condition initiale est utilisée  $(x_0, y_0, \theta_0) = (2, -2, 0)$ . les allures des coordonnées du robot ainsi les signaux de commande sont les suivantes :

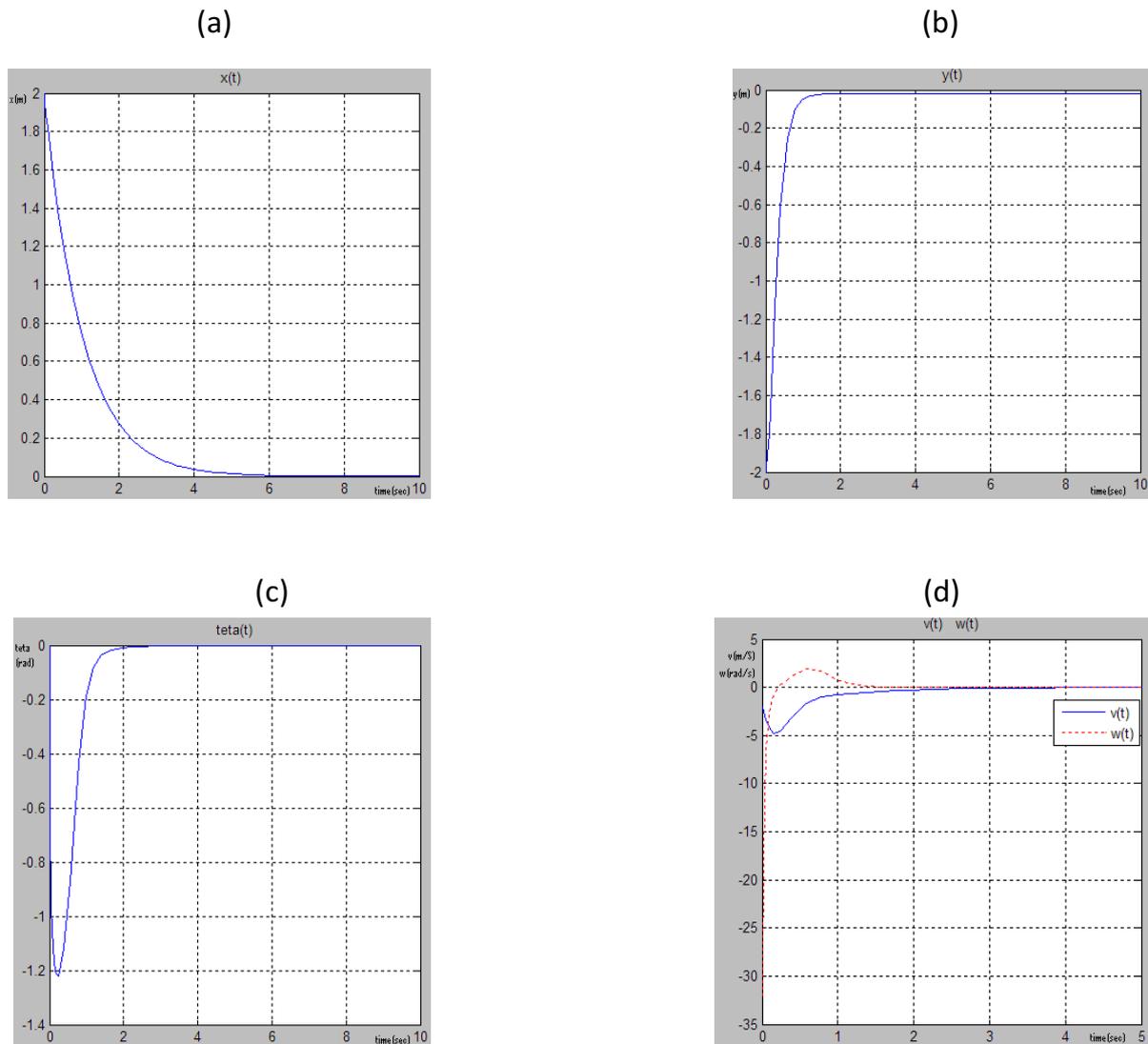


Figure 2.21 : Résultats de simulation pour l'état initial  $(x_0, y_0, \theta_0) = (2, -2, 0)$

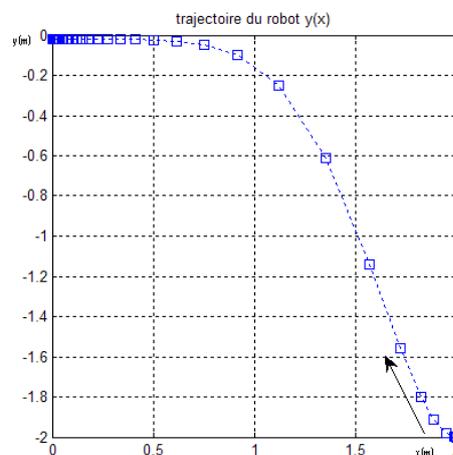


Figure 2.22 : Trajectoire du robot correspondante

- **Conclusion:**

Dans ce chapitre, trois méthodes de commande ont été synthétisées et testées afin de garantir la stabilisation d'un robot mobile de type unicycle.

La première méthode consiste à ajouter un terme discontinu par l'approche de la surface invariante à une commande par retour d'état linéaire. La loi de commande obtenue est applicable sur la forme chaînée du robot mobile.

En ce qui concerne la deuxième méthode, on s'intéresse à la transformation en coordonnées polaires, une loi de commande appliquée sur le nouveau système garantie la stabilisation du robot autour de l'origine.

La troisième méthode, consiste à appliquer la technique du Backstepping sur la forme chaînée. Pour chacune de ces commandes, nous avons effectué des tests de simulation montrant les performances des différents contrôleurs.

Après avoir étudié la stabilisation par la commande non linéaire classique, nous abordons dans le chapitre suivant une application de l'approche hybride sur la stabilisation du robot unicycle.

Chapitre3:  
Stabilisation du Robot Mobile par Commandes  
Hybrides

## Chapitre 3 : Stabilisation du Robot Mobile par Commandes Hybrides

### 3.1. Théorie des systèmes hybrides [5] :

#### 3.1.1. Définition des systèmes hybrides :

Tout système impliquant des processus continus et des phénomènes discrets peut être vu comme un système hybride. Par extension, lorsque dans un même système physique certaines grandeurs varient très rapidement (quasi-instantanément) par rapport aux autres, alors, une modélisation hybride de ce système est envisageable et donne souvent de bien meilleurs résultats qu'une modélisation continue.

Ainsi, les domaines d'application des systèmes hybrides sont extrêmement nombreux et variés; on peut citer entre autres l'informatique, l'industrie automobile, la robotique, le contrôle du trafic aérien.

En automatique, les systèmes physiques sont souvent représentés par un modèle dynamique continu ou par un modèle à événements discrets. La nature de chaque modèle est définie selon les variables utilisées pour décrire l'état du système et la variable caractérisant le temps. Il est important dans de nombreux cas d'utiliser l'une de ces deux catégories de modèles.

Cependant la majorité des systèmes complexes réalistes mélangeant le continu et le discret ne peuvent pas être classés ni dans la catégorie "système continu" ni dans la catégorie "système discret". Il est nécessaire alors d'utiliser des modèles hybrides permettant la prise en compte à la fois des variables continues et des variables discrètes ainsi que l'interaction entre elles. Plusieurs modélisations hybrides ont été développées dans la littérature. Elles peuvent être classées en trois catégories principales.

- L'approche discrète ou événementielle qui consiste à supprimer la dynamique continue du système, en interprétant le SDH comme un système à événement discret. Celle-ci est l'approche la plus utilisée mais elle n'est pas la meilleure.

En effet, parmi ses inconvénients, on a d'une part le risque d'obtenir un automate non déterministe et d'autre part le risque que la partition ne soit pas adéquate pour l'espace d'état. De plus les états continus peuvent alors avoir des transitions non admissibles.

- L'approche continue qui consiste à supprimer la dynamique discrète du système de telle façon que le système ne puisse être modélisé que par des équations différentielles

et cela peut poser un problème lors de la modélisation des systèmes complexes tels les convertisseurs par exemple. Cette approche ne prend pas en compte les changements éventuels pour le passage d'une dynamique à une autre.

- L'approche mixte qui utilise une même structure pour traiter les dynamiques continues et discrètes. Les SDH peuvent être modélisés par des automates hybrides, L'approche mixte interprète le SDH comme une interaction de systèmes dynamiques de types différents. Cette approche est plus complexe et regroupe les avantages des deux approches précédentes. Dans ce qui suit, nous rappelons pour la formulation que nous pensons la plus générale et générique introduite par Lygeros[] .

La figure (3.1) illustre un système dynamique hybride (SDH) chaque partie du SDH est caractérisé par des entrées ( $\sigma$  discrètes,  $u$  continues). Et des sorties ( $\psi$  discrète,  $y$  continues)

L'état «  $q$  » détermine la dynamique continue spécifique à la partie continue du système par l'interface (D/C : discret / continue).

Aussi, les signaux provenant de la partie continue définissent des événements qui permettent la transition dans l'espace discret d'un état vers l'autre.

Ainsi, l'interface (C/D, D/C) permet les interactions entre les variables continues et discrètes.

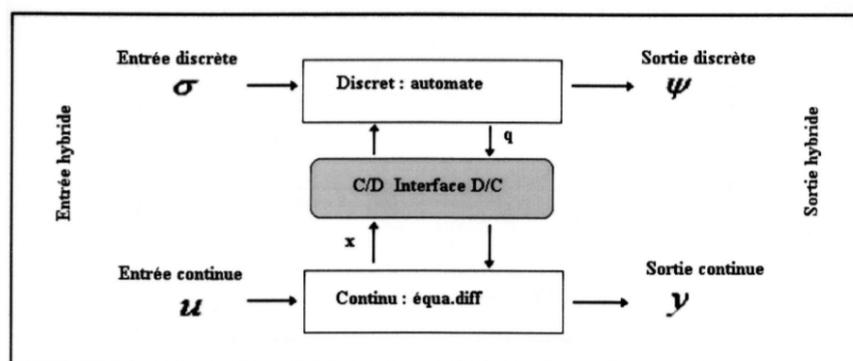


Figure 3.1 : l'Interaction entre les grandeurs continues et discrètes

### . Exemples de systèmes hybrides :

On peut prendre exemple pour les phénomènes hybrides sur la boîte à vitesse automatique. On considère qu'elle possède comme variables discrètes trois vitesses ( $V_1$ ,  $V_2$  et  $V_3$ ) et deux modes de conduites ville (avec les vitesses  $V_1$  et  $V_2$ ) et routes (avec toutes les vitesses).

La variable continue du système est le nombre de tours par minutes du moteur  $N(t)$ , sa dynamique est ainsi représentée par une équation différentielle.

Le passage d'une vitesse à une autre, lorsque la variable  $N(t)$  atteint certaines valeurs correspond à un phénomène hybride déclenché automatiquement. Or le passage d'un mode de conduite à un autre est décidé par le conducteur, Il correspond donc a un phénomène hybride contrôlé.

Lorsqu'on change de vitesse, la variable d'état  $N(t)$  est réinitialisée, sa dynamique change car on est passé d'un état à un autre. Aussi à chaque état correspond une dynamique différente. On a donc aussi une commutation de modèle.

On peut citer d'autres exemples de systèmes hybrides comme : la fontaine a deux bassins

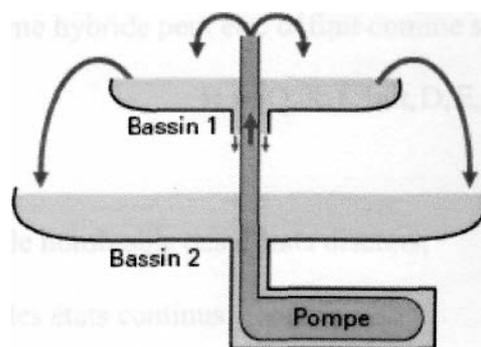


Figure 3.2 : *Système hybride* : fontaine à deux bassins

#### 3.1.2. Définition des automates hybrides :

Les systèmes hybrides sont communément modélisés avec l'utilisation des automates hybrides. L'idée est simple : Un " mode machine " fini est construit en combinant les changements de l'état continu, décrits par les équations différentielles, qu'on appelle les modes et les changements de l'état discret qui sont modélisés en utilisant les transitions, dépendant des conditions, et des performances des actions possibles, entre les modes.

**Définition** : un système hybride peut être défini comme suit

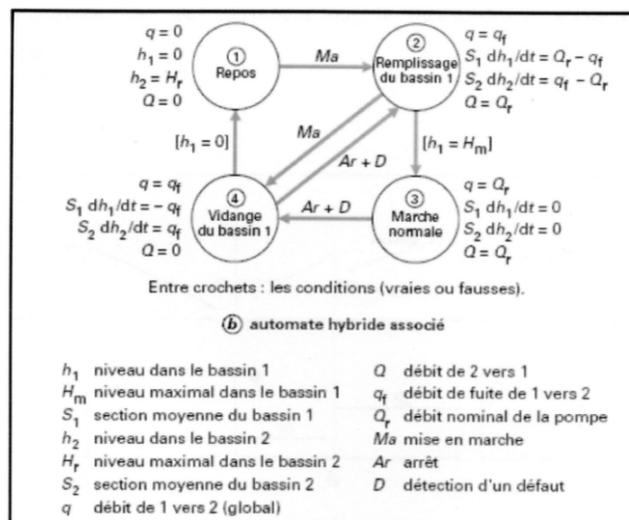
$$H = \{Q, X, f, \text{Init}, D, E, G, R, U\}$$

- Q est l'ensemble dénombrable des d'états discrets.
- X est l'ensemble des états continus.
- $I_{\text{init}}$  est l'ensemble des conditions initiales.
- $E \subset Q \times Q$  est l'ensemble des transitions.
- $D = \{D_q; q \in Q\}$  est l'ensemble des domaines continus,  $\forall q \in Q, D_q$  est sous ensemble de  $\mathbb{R}^n$  d'intérieur non vide.
- $U = \{U_q; q \in Q\}$  est l'ensemble des domaines continus autorisés,  $\forall q \in Q, U_q$  est un sous-ensemble de  $\mathbb{R}^n$ .
- $f = \{f_q; q \in Q\}$ , est l'ensemble des champs de vecteurs décrivant les dynamiques du système,  $\forall q \in Q, f_q : D_q \times U_q \rightarrow \mathbb{R}^n$ .
- $G = \{G_e; e \in E\}$  est l'état de garde,  $\forall e = (q; q') \in E, G_e \subseteq D_q$ .
- $R = \{R_e; e \in e\}$  est l'ensemble des fonctions resets.

$\forall e = (q; q') \in E, R_e : G_q \rightarrow 2D_{q'}$  où  $2D_{q'}$  dénote l'ensemble des parties de  $D_{q'}$ ,  
On suppose que pour tout  $x \in G_e, R_e(x) \neq \Phi$ .

Pour revenir à l'exemple précédant (fontaine à deux bassins):

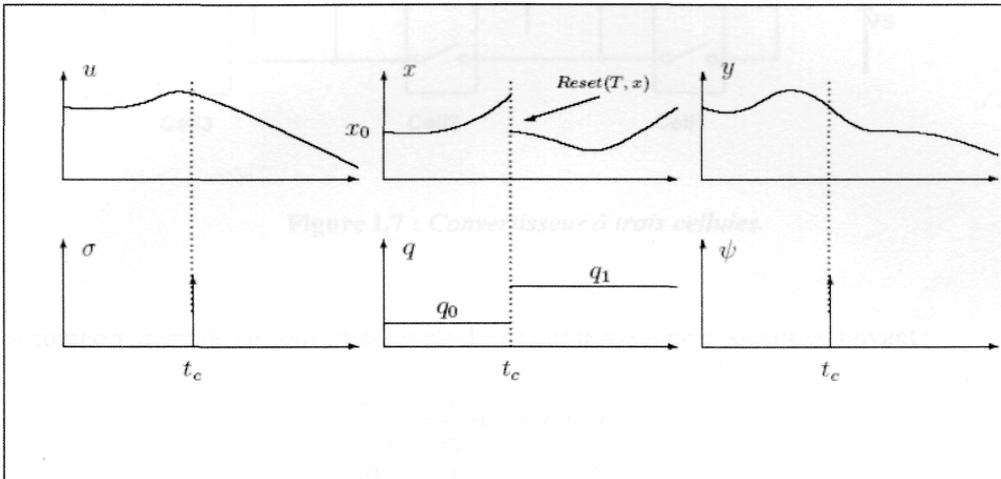
L'automate hybride associé à son fonctionnement est donnée comme suit :



**3.1.3. Exécution d'un système hybride :**

Un système hybride est caractérisé à chaque instant par une évolution de l'état discret  $q(t)$  et une évolution de l'état continu  $x(t)$ . La paire  $(q, x)$  compose le vecteur d'état hybride (respectivement les paires  $(\sigma, u)$  et  $(\psi, y)$  composent les entrées hybrides et les sorties hybrides). La variable discrète  $q(t)$  est constante entre deux instants de commutation successifs  $t_i, t_{i+1}$ . Par conséquent  $q(t)$  est constant par intervalles de temps  $I_i = [t_i, t_{i+1}]$ ,  $\{t_i, i = 1, \dots, N\}$  où  $N$  est le nombre de commutations.

Une trajectoire temporisée est une séquence finie ou infinie d'intervalles de temps  $I = \{I_i\}, i = 1, \dots, N$ . Suivant les valeurs de  $N$  et de  $I_i$ . Ainsi, une exécution d'un système hybride est représentée par des séquences d'intervalles de temps sur lesquels le vecteur hybride  $(q(t), x(t))$  évolue. La figure ci-dessous illustre un exemple d'exécution d'un SDH où nous avons mentionné l'évolution des trois variables hybrides  $(x, q), (\sigma, u)$  et  $(\psi, y)$ . L'entrée discrète et la sortie discrète sont indiquées par des impulsions (franchissement instantané).



A l'instant  $t_0=0$ , l'état hybride initial est  $(q_0, x_0)$  tel que  $x_0 \in Inv(q_0)$  et  $q_0 \in Q$ . Sur l'intervalle de temps  $I_0 = [t_0, t_c]$ , l'état discret est constant  $q(t) = q_0$  et  $x(t)$  évolue suivant l'équation  $\dot{x}(t) = f_{q_0}(x(t), u(t))$  avec  $x_0$  comme condition initiale. Enfin, lorsqu'à l'instant  $t_c$  une transition  $T = (q_0, \sigma, q_1)$  (due à l'arrivée d'une entrée discrète  $\sigma \in \Sigma$ ) se produit, la variable discrète  $q(t)$  prend alors une nouvelle valeur  $q_1$ . La sortie discrète devient  $\psi = \gamma(T)$  et la variable continue  $x$  est actualisée à une nouvelle valeur  $Reset(T, x)$ . Nous répétons alors le même processus avec une nouvelle dynamique

$$\dot{x}(t) = f_q(x(t), u(t)).$$

### 3.2. Commande hybride stabilisante de l'unicycle [6] :

La commande proposée dans ce chapitre fait partie des classes de commandes hybrides qui utilisent les dynamiques continues et les événements discrets.

Nous présentons dans ce qui suit une approche de commande hybride proposée par [Hespanha 1998] appliquée sur modèle cinématique de l'unicycle présenté sous la forme d' « intégrateur non holonome ».

#### 3.2.1. L'intégrateur non holonome (de Brockett1983) :

La transformation des modèles cinématiques des robots mobiles en formes canoniques permet de développer des commandes générales et systématiques.

Dans cette section, nous présentons une des formes canoniques fréquemment utilisée, c'est l'intégrateur non-holonome.

Considérons le changement de coordonnées suivant :

$$\begin{aligned} z_1 &= \theta \\ z_2 &= x \cos \theta + y \sin \theta \\ z_3 &= x \sin \theta - y \cos \theta \end{aligned} \quad (3.1)$$

Et les nouvelles entrées de commandes :

$$\begin{aligned} u_1 &= w \\ u_2 &= v - z_3 u_1 \end{aligned} \quad (3.2)$$

Le modèle cinématique de l'unicycle s'écrit en forme chaînée comme suit :

$$\begin{aligned} \dot{z}_1 &= u_1 \\ \dot{z}_2 &= u_2 \\ \dot{z}_3 &= z_2 u_1 \end{aligned} \quad (3.3)$$

Pour aboutir à la forme désirée, considérons un nouveau changement de coordonnées :

$$\begin{aligned} x_1 &= z_1 \\ x_2 &= z_2 \\ x_3 &= -2z_2 + z_1 z_2 \end{aligned} \quad (3.4)$$

La dynamique du système (3.4) s'écrit sans la forme d'intégrateur non-holonome :

$$\begin{aligned}\dot{x}_1 &= u_1 . \\ \dot{x}_2 &= u_2 . \\ \dot{x}_3 &= x_1 u_2 - x_2 u_1 .\end{aligned}\tag{3.5}$$

Selon [4] tous les modèles cinématiques des systèmes complément non-holonomes à trois variables d'état et deux entrées de commande peuvent être représentés sous la forme (3.5).

### 3.2.2. Commande hybride de l'intégrateur non-holonome

Les lois de commande hybride proposées par [4] sont construites comme suit :

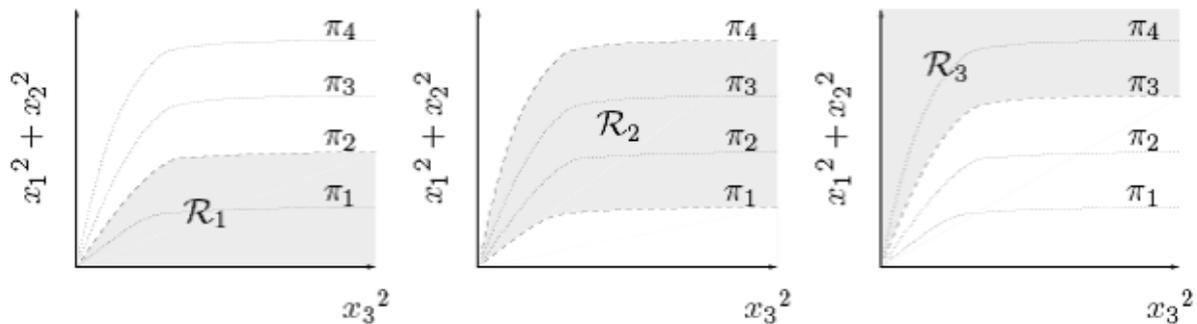
1- Définir quatre fonctions continues, monotones  $p_j$

$p_j: [0, +\infty] \rightarrow \mathcal{R}, j \in \{1,2,3,4\}$  . Ayant les propriétés suivantes :

- $p_j(0) = 0$  et  $0 < p_1(\cdot) < p_2(\cdot) < p_3(\cdot) < p_4(\cdot)$ .
- $p_1$  et  $p_2$  sont bornées.
- Si  $\emptyset \rightarrow 0$  exponentiellement alors  $\frac{\emptyset}{p_1(\emptyset)} \rightarrow 0$  et  $p_4(\emptyset) \rightarrow 0$  exponentiellement assi.

2- Partager  $\mathbb{R}^3$  en trois régions :

$$\begin{aligned}R_1 &= \{X \in \mathbb{R}^3: 0 \leq x_1^2 + x_2^2 < p_2(x_3^2)\} \\ R_2 &= \{X \in \mathbb{R}^3: p_1(x_3^2) < x_1^2 + x_2^2 < p_4(x_3^2)\} \\ R_3 &= \{X \in \mathbb{R}^3: p_3(x_3^2) < x_1^2 + x_2^2\}\end{aligned}$$



L'auteur dans a proposé les fonctions suivantes :

$$p_1(\emptyset) = 0.5(1 - e^{-\sqrt{\emptyset}}); \quad p_2 = 1.7p_1 ; \quad p_3 = 2.5p_1; \quad p_4 = 4p_1.$$

3- Définir une loi de commande pour chaque région  $R_j$  :

$$U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = g_q(x), \quad q \in Q = \{1,2,3\}$$

$$\text{Avec } g_1(x) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} ; g_2(x) = \begin{bmatrix} x_1 + \frac{x_2 x_3}{x_1^2 + x_2^2} \\ x_2 - \frac{x_1 x_3}{x_1^2 + x_2^2} \end{bmatrix}$$

$$g_3(x) = \begin{bmatrix} -k_1 x_1 + k_2 \frac{x_2 x_3}{x_1^2 + x_2^2} \\ -k_1 x_2 - k_2 \frac{x_1 x_3}{x_1^2 + x_2^2} \end{bmatrix}, \quad k_1, k_2 > 0 \quad (3.6)$$

4- Définir la fonction des transitions :

$$q = \emptyset(x, q^-), \quad t > t_0 \quad (3.7)$$

$$q^-(t_0) = q_0 \quad (\text{valeur d'initialisation}).$$

$$\text{Tel que : } \emptyset(x, j) = \begin{cases} j & \text{si } X \in LR_j \\ \max\{i \in Q : X \in LR_i\} & \text{si } X \notin R_j \end{cases} \quad (3.8)$$

Avec  $X \in LR^3, j \in Q = \{1,2,3\}$

- **Stabilité d'un système hybride :**

La définition usuelle de la stabilité au sens de Lyapounov s'étend naturellement aux systèmes hybrides

En effet, l'origine est un point d'équilibre stable (au sens de Lyapounov) du système hybride H défini par:  $\dot{x} = f_q(x)$ ,  $x \in X = LR^n, q = \emptyset(x, q^-), q \in Q$

Si : (i)  $f_j(0) = 0$  pour tout  $j \in Q$  tel que :  $\emptyset(0, j) = j$

(ii) pour tout  $\varepsilon > 0$  il existe  $\delta > 0$  tel que

$\forall x_0 \in X$  et  $\forall q_0 \in Q : \|x_0\| < \delta \Rightarrow$  La solution  $\{q, x\}$  du système H vérifie

$$\|x(t)\| < \varepsilon, t \geq t_0$$

Autrement dit, si pour n'importe quelles conditions initiales, la partie continue de  $x$  converge vers l'origine, alors l'origine est dit globalement asymptotiquement stable.

**Théorème** : considérons le système hybride  $H$  défini par (3.5), (3.6), (3.7) :

- (1) L'origine est un point d'équilibre pour  $H$  et il est globalement asymptotiquement stable.
- (2) la partie continue de  $X$  et le signal de commande  $U$  convergent vers zéro exponentiellement rapide.

### 3.2.3. Conditions de transitions :

Pour simplifier les écritures posons:  $S = x_1^2 + x_2^2$

Les conditions de transitions entre les différents modes de commande sont :

$$G(q_1, q_2) = \{X \in \mathbb{R}^3: [(s \geq p_2(x_3^2)) \wedge (S \leq p_3(x_3^2))]\}$$

$$G(q_2, q_1) = \{X \in \mathbb{R}^3: [S \leq p_1(x_3^2)]\}$$

$$G(q_1, q_3) = \{X \in \mathbb{R}^3: [S > p_3(x_3^2)]\}$$

$$G(q_3, q_1) = \{X \in \mathbb{R}^3: [S \leq p_1(x_3^2)]\}$$

$$G(q_2, q_3) = \{X \in \mathbb{R}^3: [S \geq p_4(x_3^2)]\}$$

$$G(q_3, q_2) = \{X \in \mathbb{R}^3: [(s \geq p_1(x)) \wedge (S \leq p_3(x_3^2))]\}$$

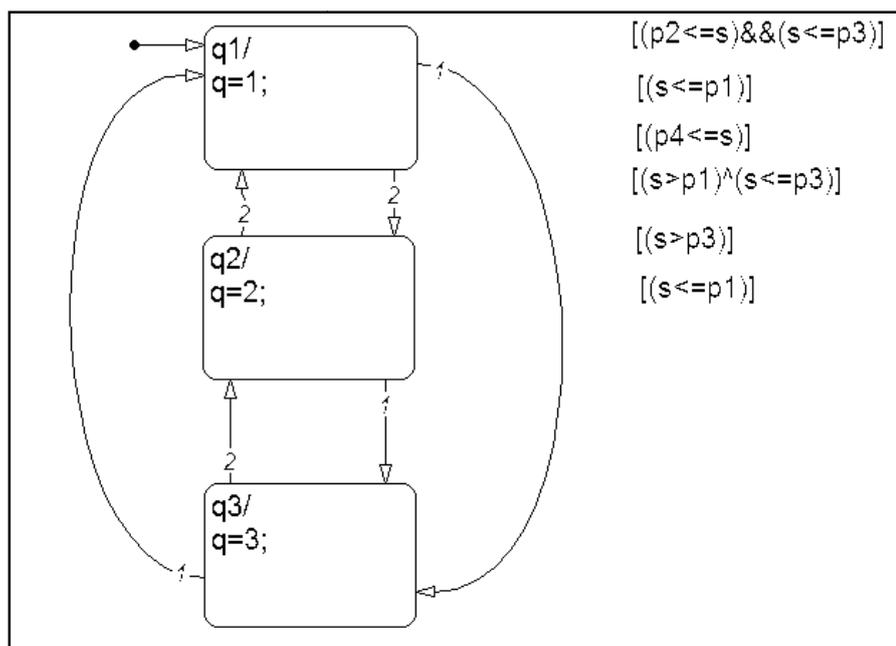


Figure 3.3: Schéma des transitions sous StateFlow

• **Simulations:**

Nous allons montrer le comportement du robot en appliquant la commande stabilisante  $(v,w)$  donnée précédemment .nous allons tester les performances du contrôleur à différentes conditions initiales .

Les valeurs numériques des gains sont choisies comme suit :  $k_1 = 0.6, k_2 = 0.9$  .

Aussi, nous nous intéressons à observer le comportement des commandes en translation et en rotation pendant le mouvement du robot et les transitions entre les trois modes.

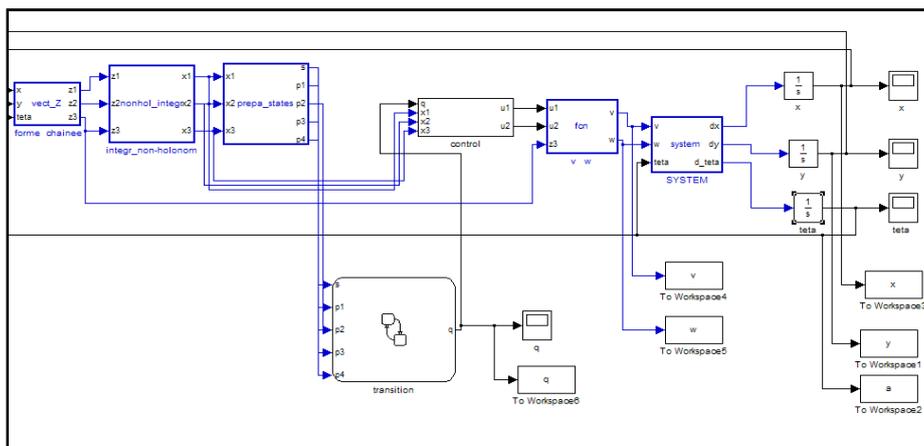


Figure3.4 : schéma bloc Simulink de la commande hybride

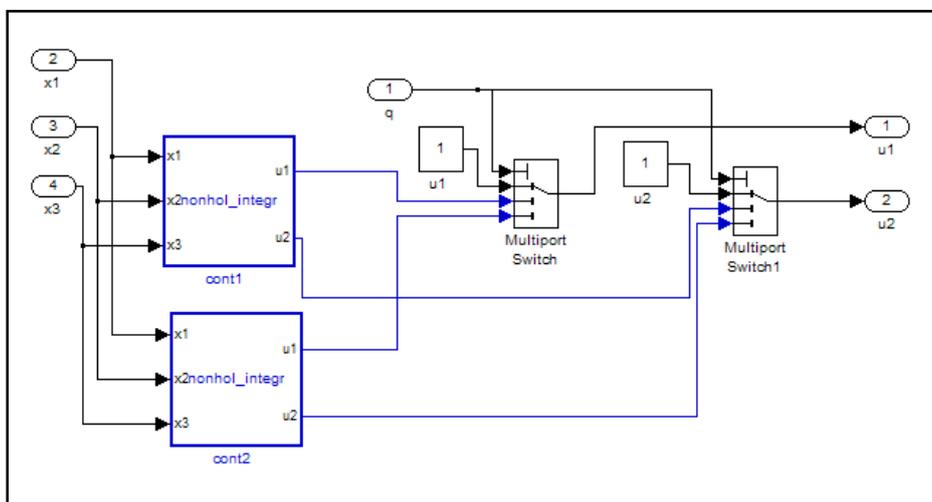


Figure3.5 : schéma bloc Simulink du bloc 'control'

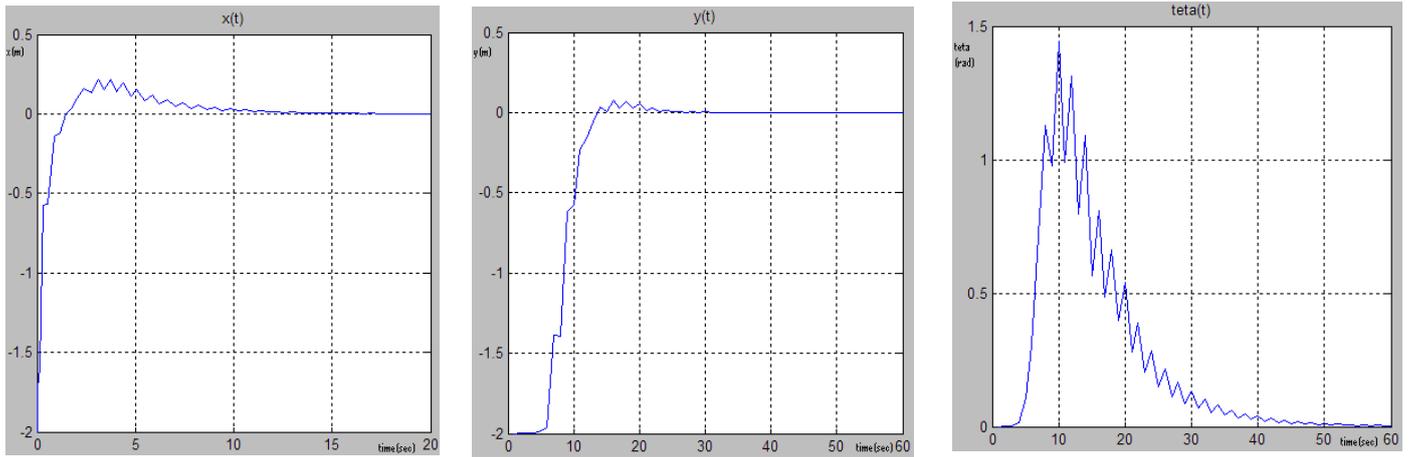


Figure3.6 : Evolution des variables d'état (état initial= $(-2,-2,0)$ )

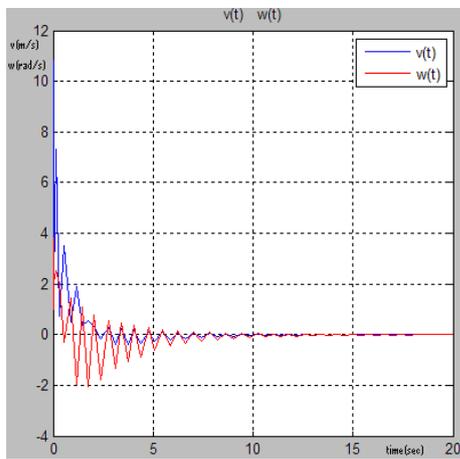


figure 3.7: Evolution des commandes (v,w)

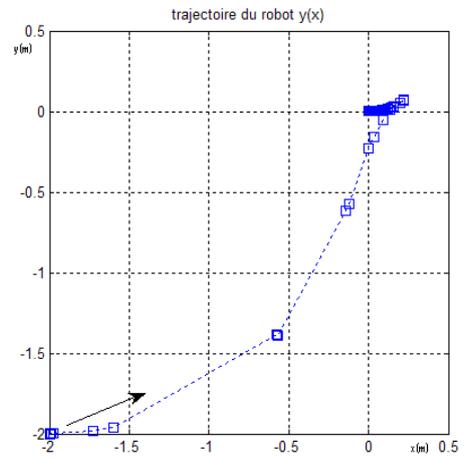


Figure 3.8:Trajectoire y(x)

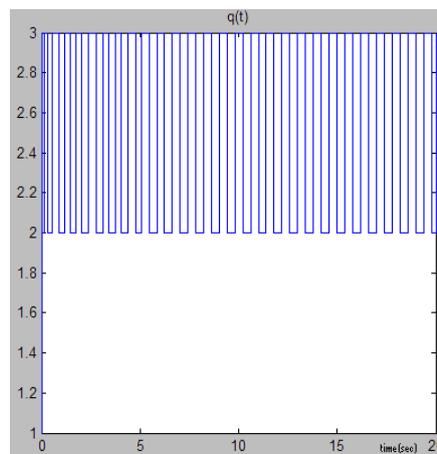
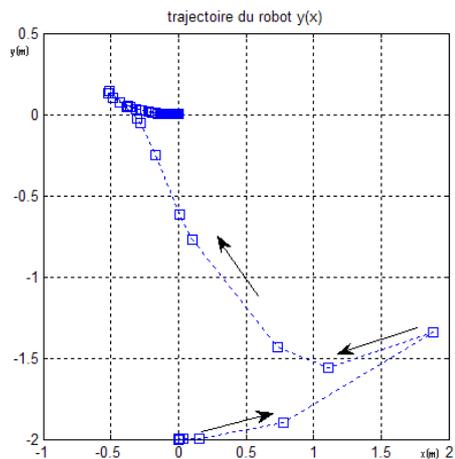
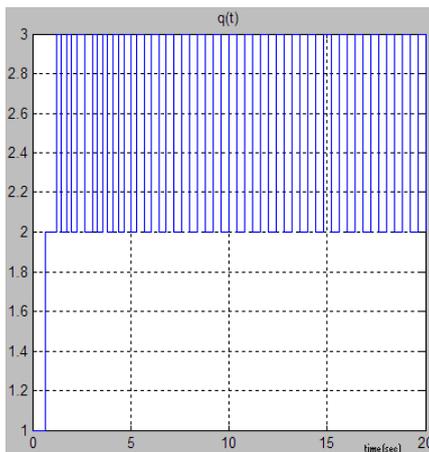
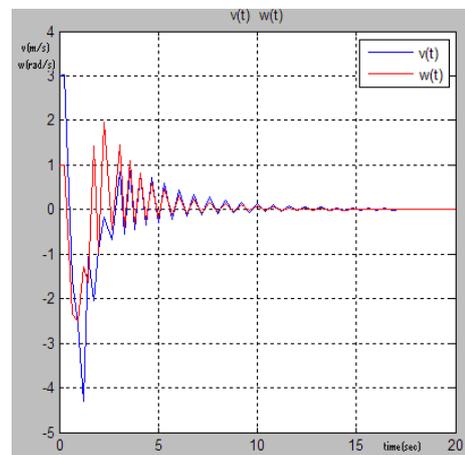
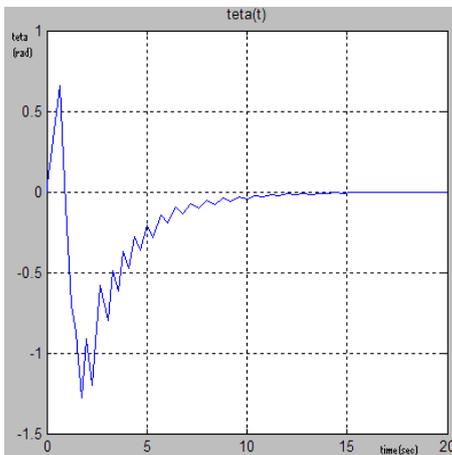
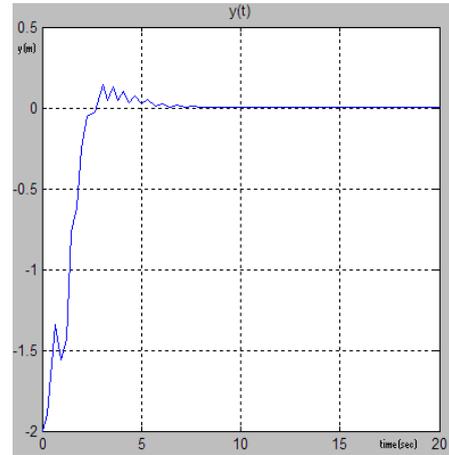
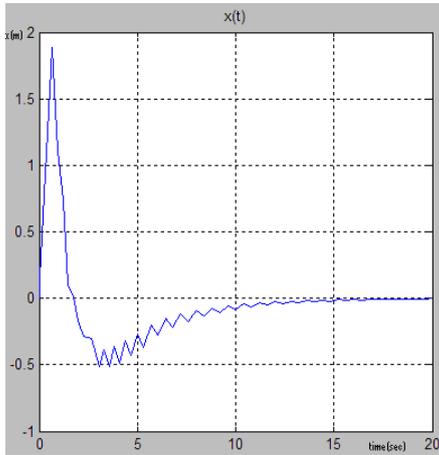
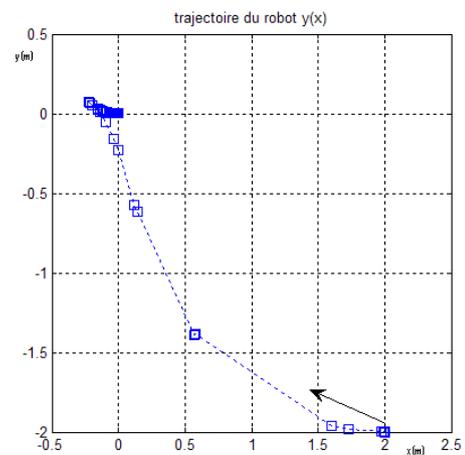
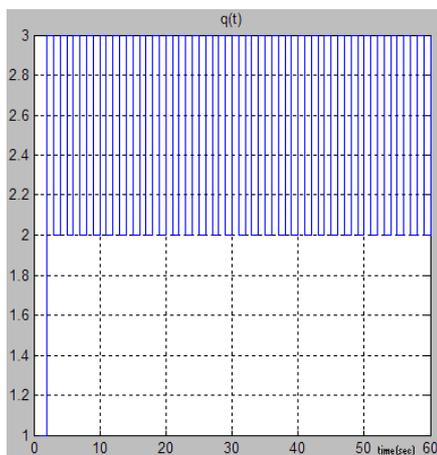
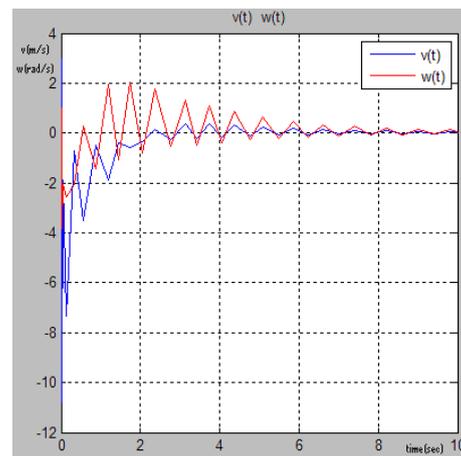
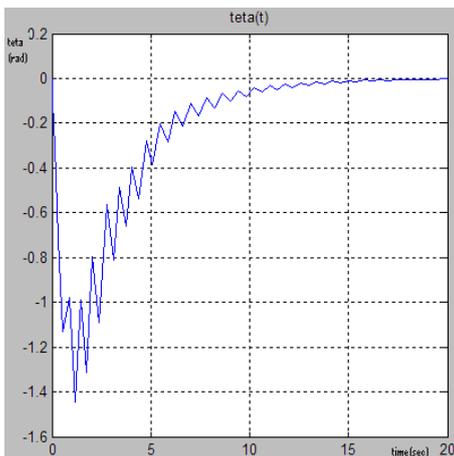
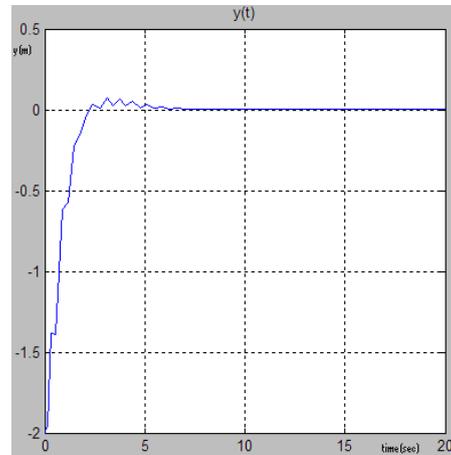
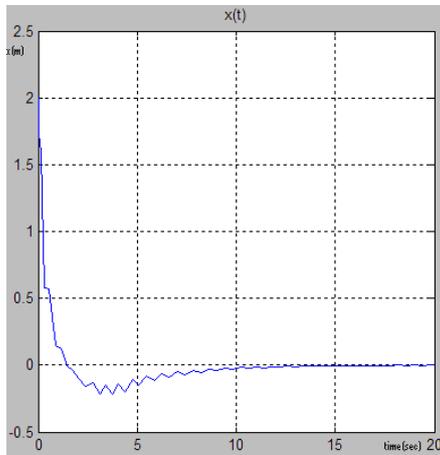


Figure 3.9:Evolution des transitions

Pour la position initiale  $(x_0, y_0, \theta_0) = (0, -2, 0)$  les résultats de simulation sont donnés par :



Pour la position initiale  $(x_0, y_0, \theta_0) = (2, -2, 0)$  les résultats de simulation sont donnés par



### 3.3. Stabilisation par une commande hybride dynamique [7] :

En vu de simplifier les lois de commande, la plupart des approches négligent les dynamiques du second ordre de l'unicycle. Ces approches considèrent que le robot est piloté par le système de vitesses (modélisation et commande cinématique) au lieu des couples moteurs.

Dans cette section on applique une loi de commande hybride sur le modèle dynamique simplifier du robot en vu de le stabiliser autour le l'origine.

Cette commande est valable pour tout système non holonome qu'on peut mettre sous la forme d'un « double intégrateur non holonome ».

#### 3.3.1. Le double intégrateur non holonome(DIN) :

Considérons le modèle dynamique simplifiés de l'unicycle, sous les hypothèses suivantes :

- Roulement sans glissement.
- Masses et moments d'inertie des roues sont négligeables.
- Le centre de masse du robot se situe au milieu de l'axe qui rejoint les deux roues arrière.

Le modèle du système s'écrit alors avec les équations :

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

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

$$\dot{\theta} = w$$

$$m\dot{v} = F$$

$$I\dot{w} = N \quad (3.8)$$

Où

- $(x, y)$  sont les coordonnées cartésiennes du centre de masse.
- $\theta$  est l'orientation du robot par rapport à l'axe des x.
- $v, w$  sont les vitesses linéaire et angulaire du robot respectivement.
- $m, I$  : sont la masse et le moment d'inertie du robot.
- Les entrées de commande sont :  
 $F$  : Force de traction( pushing force)  
 $N$  : couple d'orientation (Steering torque) qui sont reliées aux couples moteur  $T_1, T_2$  des deux roues par :

$$F = \frac{1}{r}(T_1 + T_2)$$

$$N = \frac{L}{r}(T_1 - T_2)$$

Où  $r$  est le rayon d'une roue et  $2L$  est la distance entre les roues arrières.

Soit la transformation d'état suivante :

$$z_1 = \theta$$

$$z_2 = x \cos \theta + y \sin \theta$$

$$z_3 = x \sin \theta - y \cos \theta$$

$$u_1 = \frac{N}{I}$$

$$u_2 = \frac{F}{m} - \frac{N}{I}z_3 - w^2z_2$$

Le modèle dynamique du robot s'écrit sous la forme chaînée étendue comme suit :

$$\ddot{z}_1 = u_1$$

$$\ddot{z}_2 = u_2$$

$$\dot{z}_3 = z_2 \dot{z}_1$$

Pour aboutir à la forme d'un double intégrateur non-holonome, on utilise un changement de variable :

$$x_1 = z_1$$

$$x_2 = z_2$$

$$x_3 = -2z_3 + z_1z_2$$

Le système (3.8) s'écrit alors comme :

$$\ddot{x}_1 = u_1$$

$$\ddot{x}_2 = u_2$$

$$\dot{x}_3 = x_1\dot{x}_2 - x_2\dot{x}_1 \quad (3.9)$$

C'est la forme du double intégrateur non-holonome (DIN) dont le vecteur d'état est  $x = (x_1, \dot{x}_1, x_2, \dot{x}_2, x_3)$ . c'est une extension de l'intégrateur non-holonome.

### 3.3.2. Commande hybride du (DIN) :

Les lois de commande ressoudent le problème du convergence globale et stabilisation du (DIN) vers un petit voisinage de l'origine selon la démarche suivante :

1. Définir une projection du  $W(\cdot)$  vecteur d'état

$$(x_1, x_2, x_3, \dot{x}_3) \in \mathbb{R}^4 \text{ Dans } \Omega \subset \mathbb{R}^2 \text{ tel que } W(x) = [w_1, w_2] = [s^2; x_1^2 + x_2^2]$$

$$\text{où } s = \dot{x}_3 + \lambda x_3, \lambda > 0$$

On note que :

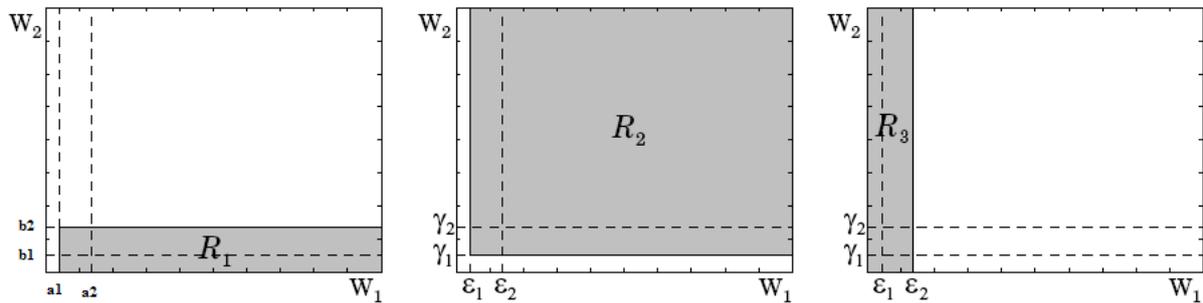
- $W(0) = 0$
- Si  $W(x) \rightarrow 0$  lorsque  $t \rightarrow \infty$  alors  $x \rightarrow 0$  aussi lorsque  $t \rightarrow \infty$ .

2. Partager  $\Omega \subset \mathbb{R}^2$  en trois régions (voire figure ci-dessous) :

$$R_1 = \{(w_1, w_2) \in \Omega : (w_1 > a_1) \wedge (w_2 \leq b_2)\}$$

$$R_2 = \{(w_1, w_2) \in \Omega : (w_1 > a_1) \wedge (w_2 \geq b_1)\}$$

$$R_3 = \{(w_1, w_2) \in \Omega : w_1 \leq a_2\}$$



où  $a_2 > a_1 > 0$  et  $b_2 > b_1 > 0$  (eg :  $a = 0.001, a = 0.2, b = 1, b = 2$ ).

3. Définir trois lois de commande pour chaque région :

$$U = [u_1 u_2]^T = g_q(x), q \in Q = \{1, 2, 3\}$$

Tel que :

$$\begin{aligned} g_1(x) &= \begin{bmatrix} -\lambda \dot{x}_1 + x_1 \\ -\lambda \dot{x}_2 + x_2 \end{bmatrix} \\ g_2(x) &= \begin{bmatrix} -\lambda \dot{x}_1 + x_1 + x_2 s \\ -\lambda \dot{x}_2 + x_2 - x_1 s \end{bmatrix} \\ g_3(x) &= \begin{bmatrix} -\lambda \dot{x}_1 - x_1 \\ -\lambda \dot{x}_2 - x_2 \end{bmatrix} \end{aligned} \quad (3.10)$$

4. Définir la fonction de transition :

$$\begin{aligned} q &= \emptyset(x, q^-), \quad t > t_0 \\ q^-(t_0) &= q_0 \quad (\text{Valeur d'initialisation}). \end{aligned} \quad (3.11)$$

$$\text{Tel que :} \quad \emptyset(x, j) = \begin{cases} j & \text{si } x \in lR_j \\ \max\{i \in Q: x \in lR_i\} & \text{si } x \notin R_j \end{cases} \quad (3.12)$$

Avec  $x \in lR^5, j \in Q = \{1,2,3\}$

Par cette fonction on aboutit aux transitions suivantes :

$$q_{12} = [(w_1 > a_2) \&\& (w_2 > b_2)]$$

$$q_{13} = [(w_1 \leq a_1) \|\| ((w_1 \leq a_2) \&\& (w_2 > b_2))]$$

$$q_{21} = [(w_1 > a_2) \&\& (w_2 < b_1)]$$

$$q_{23} = [(w_1 \leq a_1) \|\| ((w_1 \leq a_2) \&\& (w_2 < b_1))]$$

$$q_{31} = [(w_1 > a_2) \&\& (w_2 < b_1)]$$

$$q_{32} = [(w_1 > a_2) \&\& (w_2 \geq b_1)]$$

• **Théorème :**

Considérons le système hybride  $H$  défini par (3.9) (3.10) et (3.11). Soit  $\{x(t), q(t)\}$  une solution de  $H$ , alors :

- 1-  $\{x(t), q(t)\}$  est unique  $\forall t \geq t_0$
- 2- Pour n'importe quelle condition initiale  $\{x_0, q_0\}$  il existe un temps fini  $T \geq t_0$  tel que pour  $t > T$  les variables d'état  $x_1, \dot{x}_1, x_2, \dot{x}_2$  convergent exponentiellement à zéro.
- 3- L'origine est un point d'équilibre stable (stabilité au sens de Lyapounov)

- **Résultats de simulation :**

Le paramètre de contrôle  $\lambda = 1$  et les trois régions sont définis par :

$$a_1 = 0.001, a_2 = 0.2, b_1 = 1, b_2 = 2$$

Pour le robot, on considère que :  $m = 1, I = 1$

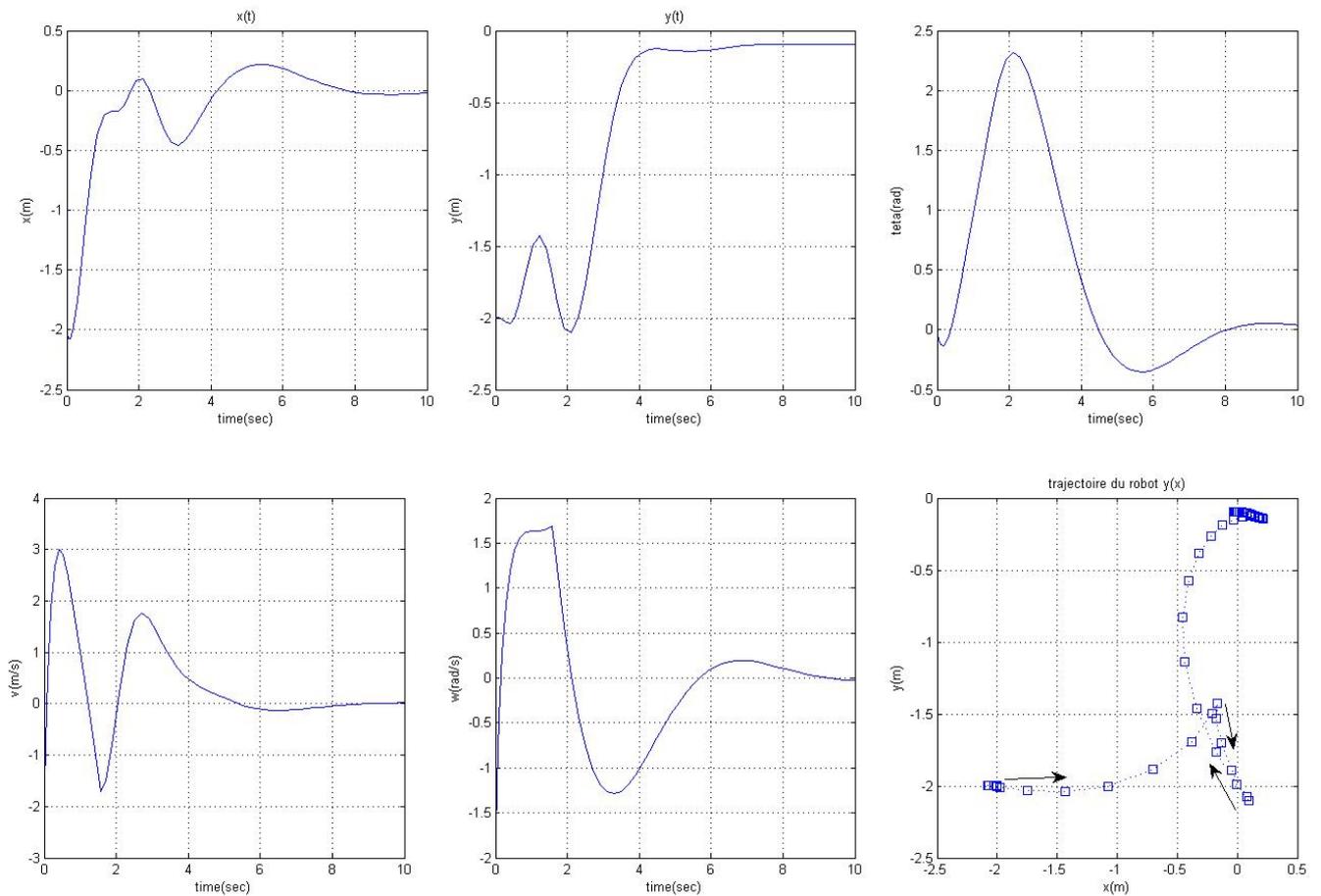


Figure 3.10 : Résultats de simulation, condition initiale  $(-2, -2, 0, 0)$

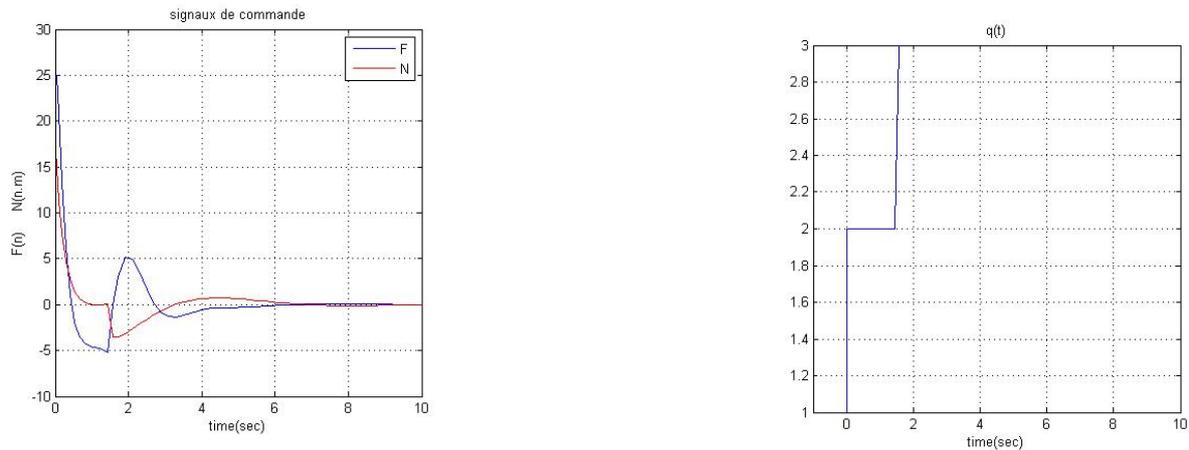


Figure 3.11 :Evaluation des commandes

- **Conclusion :**

Dans ce chapitre, deux méthodes de commande hybride ont été synthétisées et testées afin d'assurer la stabilisation du robot mobile unicycle, la première est applicable sur le modèle cinématique en forme d'intégrateur non-holonome, la deuxième concerne la stabilisation du modèle dynamique simplifiée en forme du DIN.

La première commande s'effectue en trois modes. Chaque mode possède une loi de commande. la fonction des transitions garantie le passage entre ces modes.

# Chapitre4: Poursuite de Trajectoire du Robot Mobile

## Chapitre 4 : Poursuite de Trajectoire du Robot Mobile

- **Introduction :**

La résolution du problème de la poursuite d'une trajectoire nécessite la combinaison de deux commandes : une commande nominale (feedforward) et une commande stabilisante appliquée au système d'erreur [4].

#### 4.1. Génération des commandes (feedforward)[4] :

Soit le vecteur d'état nominal  $q_d=(x_d(t), y_d(t), \theta_d(t))$  avec

$t \in [0, T]$  (Possibilité de  $T \rightarrow \infty$ ).

Ce vecteur représente le mouvement cartésien désiré. Il doit satisfaire les contraintes non holonomes de l'unicycle, plus précisément, celles du modèle cinématique, i.e.:

$$\dot{x}_d = v_d \cos \theta_d \quad (i)$$

$$\dot{y}_d = v_d \sin \theta_d \quad (ii)$$

$$\dot{\theta} = w_d .$$

De (i) et (ii) l'état  $\theta_d(t)$  peut être calculé par la relation :

$$\theta_d(t) = \text{atan2}(\dot{y}_d, \dot{x}_d) + k\pi, k = 0.1 \quad (4.1)$$

où:

$\text{atan2}$  : est la fonction tangente inverse du quatrième quadrant, (n'est définie uniquement que si ses deux arguments sont nuls).

$k = 0$  (Respect,  $k = 1$ ) correspond au marche en avant (respect, en arrière).

Supposons que le mouvement cartésien  $(x_d(t), y_d(t))$  est deux fois différentiable sur  $[0, T]$ , les commandes nominales (feedforward) sont données par :

$$v_d(t) = \pm \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \quad (4.2)$$

$$w_d(t) = \frac{\dot{y}_d(t) \dot{x}_d(t) - \ddot{x}_d(t) \dot{y}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \quad (4.3)$$

L'équation (4.3) s'obtient en dérivant l'équation (4.1) par rapport au temps.

Le choix du signe de  $v_d(t)$  va déterminer le sens du mouvement : en avant ou en arrière.

## 4.2. Lois de commandes stabilisant la trajectoire :

On définit le vecteur d'erreur  $e$  comme étant l'erreur de poursuite avec une rotation locale d'angle  $\theta$  :

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{bmatrix} \quad (4.4)$$

Pour simplifier de calcul la dynamique de l'erreur  $e$ , considérons le changement de variables suivant :

$$v = v_d \cos e_3 - u_1 \quad (4.6)$$

$$w = w_d - u_2 \quad (4.7)$$

La dynamique de l'erreur  $e$  devient alors :

$$\dot{e} = \begin{bmatrix} 0 & w_d & 0 \\ -w_d & 0 & v_d \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} v_d + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (4.8)$$

La linéarisation de ce système autour de la trajectoire référence, i.e, autour

de  $e = 0$  donne le système linéarisé variant dans le temps suivant :

$$\dot{e} = A(t)e + BU \quad (4.9)$$

Où :

$$\text{La matrice dynamique : } A(t) = \begin{bmatrix} 0 & w_d(t) & 0 \\ -w_d(t) & 0 & v_d(t) \\ 0 & 0 & 0 \end{bmatrix}$$

$$\text{Matrice : } B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

### 4.3. Contrôlabilité de la trajectoire :

#### **Définition :**

Soit le système  $\Sigma: \dot{x} = Ax + Bu$ ,  $x \in \mathbb{R}^3$ , et un point  $\bar{x}$  dans l'espace d'état.

Le système est commandable en  $\bar{x}$  s'il existe une commande admissible  $\bar{u}$  permettant de transférer le vecteur d'état de la position du repos vers  $\bar{x}$  en un temps admissible.

-le système est dit complétement commandable si et seulement si il est commandable en tout point de  $\mathbb{R}^n$ .

- Le test de commandabilité complète s'effectue à travers le calcul du rang d'une matrice  $C$  appelée matrice de commandabilité.

#### **Critère de Kalman :**

Soit la matrice de commandabilité  $C = [B, AB, \dots, A^{n-1}B]$ , le système  $\Sigma$  est complétement commandable ssi  $\text{rang}(C) = n$

Effectuons le test de contrôlabilité sur le système dynamique linéarisé de l'erreur (4.9), soit  $C$  la matrice de contrôlabilité correspondante :

$$C = [B, AB, A^2B] = \begin{bmatrix} 1 & 0 & 0 & 0 & w_d^2 & v_d w_d \\ 0 & 0 & -w_d & v_d & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Le système (4.9) est contrôlable ( $\text{rang}(C) = 3$ ) sous la contrainte

$v_d \neq 0$  Ou  $w_d \neq 0$ .

#### **4.4. Commande non linéaire de poursuite :**

Nous présentons dans cette section une commande non linéaire assurant la stabilisation du système d'erreur (4.8) :

$$\begin{aligned} u_1 &= -\bar{k}_1 e_1 . \\ u_2 &= -k_2 v_d(t) \frac{\sin e_3}{e_3} e_2 - \bar{k}_3 e_3 . \end{aligned} \quad (4.11)$$

Ou : la constante  $k_2 > 0$ .

$\bar{k}_i = k_i(v_d(t), w_d(t))$  ,  $i = 1,2$ , sont deux fonctions positives et continues à deux variables  $v_d, w_d$  .

#### **Théorème :**

*Assuming that  $v_d$  and  $w_d$  are bounded with bounded derivatives, and that  $v_d(t) \rightarrow 0$  Or  $w_d(t) \rightarrow 0$ . When  $t \rightarrow \infty$ , the control law (4.11) globally asymptotically stabilizes the origin  $e = 0$ .*

Par une analyse linéaire du système linéaire (4.9), l'auteur dans [4] a donné le choix des gains comme suit :

$$\bar{k}_1 = \bar{k}_2 = 2\xi \sqrt{w_d^2(t) + b v_d^2(t)} .$$

$$k_2 = b . \quad \text{Avec } b > 0 \text{ et } \xi \in [0,1] .$$

**4.5. Simulations :**

A partir de l'étude réalisée précédemment, nous illustrons le comportement du robot (observer les états  $x, y, \theta$ ), donc la trajectoire du robot, en appliquant la commande stabilisante donnée par les expressions (4.11). Aussi, nous nous intéressons à observer le comportement des commandes en rotation et en translation pendant le mouvement du robot.

nous fixons d'abord la valeur des différents gains de chaque commande :

$b=10 ; \xi = 0.7$

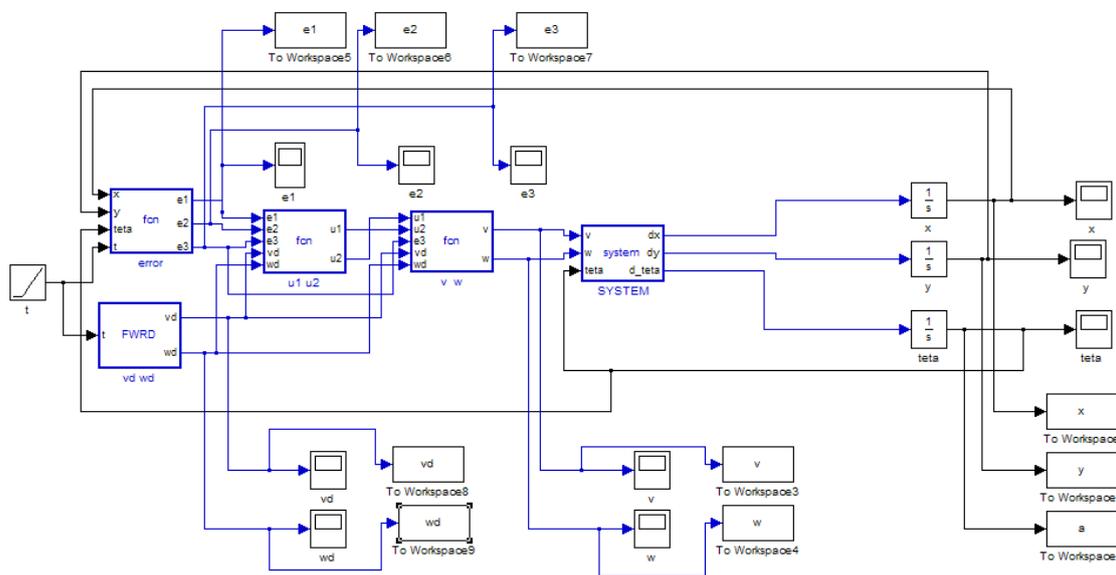
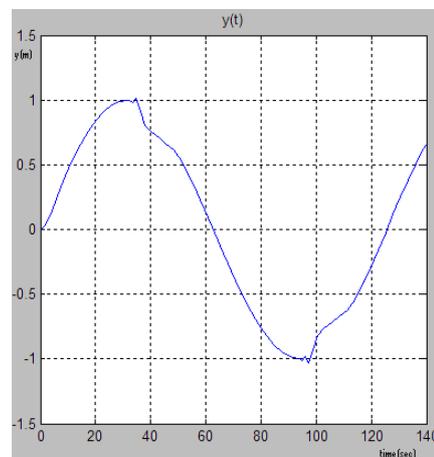
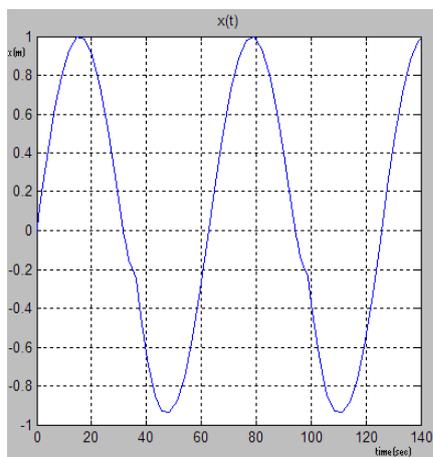
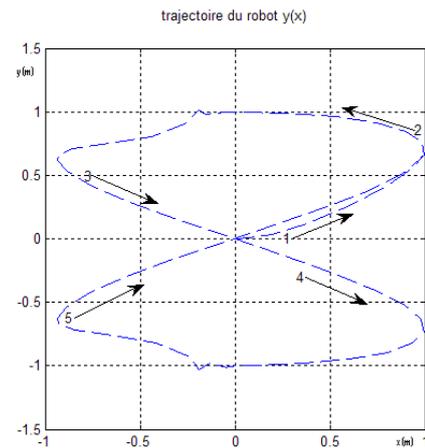
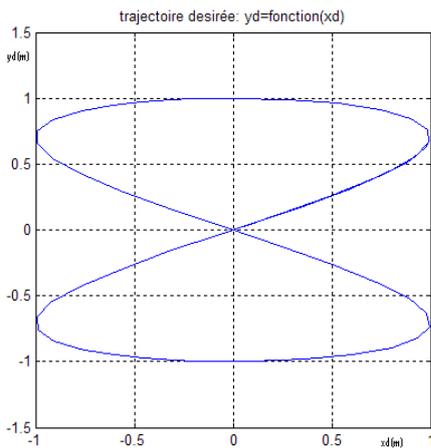
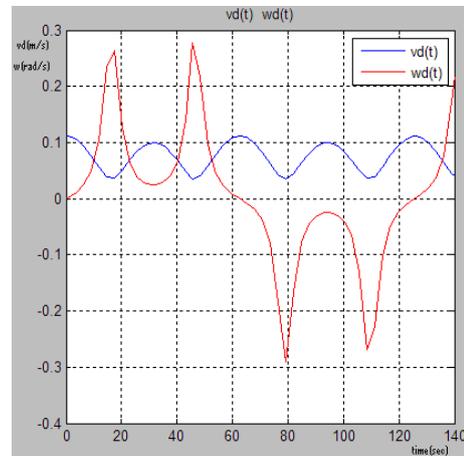
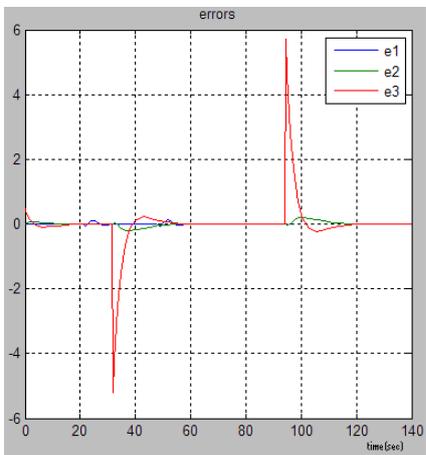
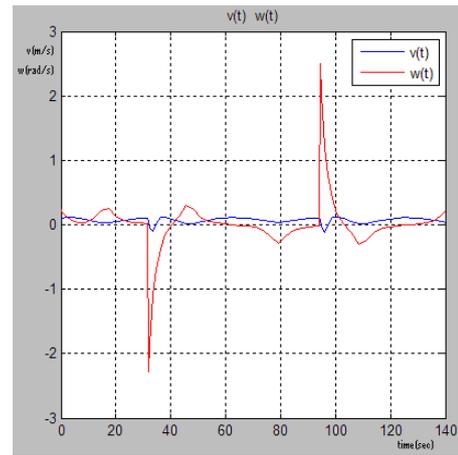
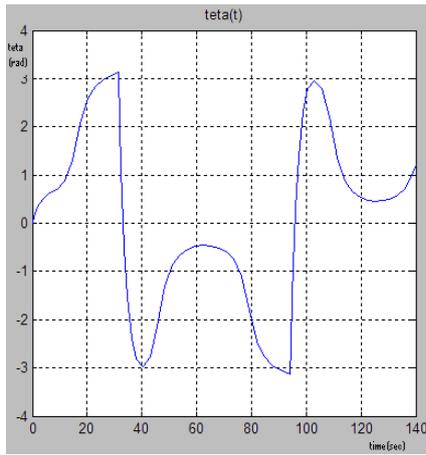


Figure4.1. : Schéma Bloc de simulation





On constate d'après les résultats obtenus ci-dessus que le robot suit la trajectoire de référence  $y_d=fonction(x_d)$  avec peu d'erreurs survenues lors du changement des directions.

## Conclusion Générale :

Dans ce travail, nous nous sommes intéressés aux problèmes fondamentaux de mouvement d'un robot mobile à roues de type unicycle.

L'objectif était, en premier temps, d'élaborer des techniques de commande en vue de stabiliser le système non-holonyme autour de l'origine. Ensuite on s'est intéressé à un problème très intéressant qui est la poursuite de trajectoire. afin d'atteindre les objectifs, nous avons présenté, dans le premier chapitre, les bases théoriques relatives aux systèmes non-holonyme (robot mobiles).en effet, nous avons établi un état d'art sur les robots mobiles à roues.

Nous avons ensuite dans le deuxième chapitre présenté notre première problématique qui est la stabilisation du robot unicycle autour de l'origine. Dans ce cadre, nous avons développé deux algorithmes de stabilisation : le premier algorithme, propose une loi de commande par retour d'état discontinue et invariant dans le temps appliquée à la forme chaînée du robot. Le deuxième algorithme consiste à appliquer la technique du Backstepping. Egalement, une troisième loi de commande est appliquée au modèle cinématique du robot en coordonnées polaires. Les simulations montrent que l'objectif de stabilisation est atteint.

Le troisième chapitre est consacré à l'application des lois de commande hybrides, toujours concernant la première problématique, stabilisant le robot autour de l'origine. Ce chapitre étudie la commande cinématique hybride ainsi celle appliquée au modèle dynamique simplifié du robot. Chacune des commande s'applique en trois modes.

Après s'être concentré sur la stabilisation autour d'un point, nous avons abordé le problème de poursuite de trajectoire, présenté dans le dernier chapitre. Réaliser cette tâche revient à stabiliser le système d'erreurs de poursuite autour de zéro. Cela est assuré par un retour d'état non linéaire. Les simulations ont montré les performances de ce contrôleur.

**Perspectives :**

Les concepts théoriques introduits dans ce travail peuvent conduire à d'autres applications futures, par exemple : stabilisation d'un robot mobile de type voiture (algorithmes du deuxième chapitre)

En plus, dans ce travail on s'est intéressé au problème de mouvement sans collision. Il serait intéressant d'étudier la tâche de poursuite assurant l'évitement de collision en présence d'obstacles. Pour se faire l'approche hybride présentée en [9] et les algorithmes de planification des chemins optimaux [10] offrent d'autres solutions.

La réalisation d'une plate-forme mobile d'expérimentation, programmable, dotée de systèmes de perception est un sujet intéressant pour l'implémentation des algorithmes de commande.

### ***Bibliographie:***

- [1] Bayle .b, « robotique mobile ».rapport de l'école nationale supérieure de physique de Strasbourg, Université Louis Pasteur, Strasbourg, 2007
- [2] Ronald, S. Illah, R, N. « Introduction to autonomous Mobile robots », the MIT press, London 2004.
- [4] Tayebi, A., Tadjdine, M., and Rashid. , A. "Invariant Manifold approach for stabilization of Non homonymic chaired Systems: Application to a Mobile robot". Non linear Dynamics.Vol.24:167-181; 2001.
- [5] A.de Luca, and G. Oriolo,"control of wheeled Mobile Robots: An Experimental Overview" .Dipartimento di Informatica, Universita degli studi di Roma "La Sesienza", Italy
- [6] Hespanha, J. P. stabilization of non homonymic integrators via logic-based switching. In: proc. 13<sup>th</sup> World Congress of IFAC. vol. E. S. Francisco, CA. pp.167-172;1996.
- [7] Agniar. A. P. and Pascal. A. M"Stabilization of un extended Non homonymic Integrator Via logic-basic hybrid control.Vienna, 2000.
- [8] Benbelkacem Samir, " stabilisation de systèmes non-holonomes par asservissement visuelle:  
Cas des robots mobiles à roues ».mémoire de Magistère de ENP, Alger, 2008.
- [9] Magnus, E. «Behavior Based robotics using hybrid automata ». Royal Institute of technology, Stockholm.