

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



Ecole Nationale Polytechnique



DEPARTEMENT AUTOMATIQUE
Laboratoire de Commande des Processus

MEMOIRE DE MAGISTER

Option : **Robotique et Productique**

Présenté par

Mohamed Lamine BENSEDDIK

Ingénieur d'Etat en Automatique de l'ENP/Alger

Thème

**Commande et navigation visuelle d'un robot mobile à roues :
Application au suivi de ligne et à l'évitement d'obstacles**

Soutenu publiquement le 13 décembre 2011 devant le jury suivant :

F. BOUDJEMA
M. TADJINE
L. NEZLI
B. HEMICI
H. CHEKIREB
A. ZELLAGUI

Professeur à l'ENP
Professeur à l'ENP
MCA à l'ENP
MCA à l'ENP
Professeur à l'ENP
Directeur CRD-GN

Président
Rapporteur
Examineur
Examineur
Invité
Invité

Année 2011

ملخص

تتناول هذه المذكرة الأطروحة المتعلقة بالتحكم في روبوت متحرك بعجلات ذو طبيعة غير هولونوميكية وذلك باستعمال الرؤية، أين قمنا باختيار نوعين من المهام وهما تتبع خط مرسوم على الأرض و التجوال مع تقادي الحواجز. لهذا الغرض، اتخذنا نظاما يتألف من كاميرا تكون موضوعة في مقدمة الروبوت و موجهة نحو الأرض بحيث تسمح بمجال رؤية مقبول.

في بادئ الأمر، أولينا اهتمامنا بإشكالية التحكم، حيث قمنا بتقديم وتجريب عن طريق المحاكاة لتقنيات تنتمي إلى استراتيجيتي تحكم مختلفتين، وهما ملاحقة المسار وتتبع الطريق. فيما يخص ملاحقة المسار، اقترحنا ثلاث طرق، الأولى وهي المنظومة الخطية ذات الرجوع الديناميكي، التي مكنت من الحصول على نظام خطي والذي يمكن التحكم فيه بواسطة منظم تناسبي مشتق بسيط. الطريقة الثانية تعتمد التحكم اللاخطي المتغير في الزمن وفيها تقدمنا بتقنيتين، واحدة تعتمد على مفهوم ليايونوف للنظم المستقرة، والأخرى تستعمل ميزة الاستقرار الخاصة بالنظم المتسلسلة. أما بالنسبة للطريقة الثالثة، فقد قمنا باستخدام نظام تحكم يحوي على نظام لمراقبة اتجاه الروبوت. فيما يخص تتبع الطريق، تقدمنا بتقنية تحكم لاخطي تشبه تلك المستعملة في ملاحقة المسار، أين يقوم الروبوت، مزودا بسرعة خطية معطاة، بتتبع معلم متحرك على طول الطريق. بالنسبة للجزء التطبيقي، قمنا باختبار هذه التقنية الأخيرة باعتبار أنها الأكثر توافقا مع السياق المطروح.

من ناحية أخرى، وفي إطار هذا العمل، تم استخدام الرؤية بالاعتماد أساسا على تقنيات معالجة الصور، من أجل استخراج أو حساب المسار الذي يحقق الهدف المرجو. بالنسبة لتطبيقية تتبع الخط المرسوم، اقتصر العمل على تحديد واستخراج الخط في المستوي المتعلق بالصورة، ومن ثم تحويله إلى المستوي الديكارتي. أما بالنسبة للتطبيقية الثانية، التي يقوم فيها الروبوت بالتجوال مع تقادي الحواجز، تعلق الأمر ببناء نموذج للمحيط نغرق من خلاله بين الحواجز وبين المنطقة الشاغرة. انطلاقا من هذه النقطة، قمنا بعملية التخطيط للمسار باستعمال أسلوب حقول الكمون، و من ثم طبقنا على الروبوت تقنية التحكم المختارة والتي أعطت نتائج مرضية كما يتبين من المحاكاة.

كلمات مفتاحية : روبوت متحرك بعجلات، نظام غير هولونوميكي، تقادي الحواجز، ملاحقة المسار، تتبع الطريق، المنظومة الخطية ذات الرجوع الديناميكي، تحكم لاخطي متغير في الزمن، تحكم مبني على مراقبة اتجاه الروبوت، تخطيط المسار، رؤية اصطناعية، معالجة الصور.

Résumé

Ce mémoire de magister traite la problématique de la commande par vision d'un robot mobile à roues non holonome, où nous avons choisi deux types de tâches, le suivi d'une ligne tracée sur le sol, et la navigation avec évitement d'obstacles. Pour ce faire, nous avons considéré un système dans lequel une caméra est positionnée à l'avant du robot et pointant vers le sol tout en permettant un champ de vision acceptable.

Dans un premier lieu, nous nous sommes intéressés au problème de la commande, pour lequel nous avons présenté et simulé des techniques appartenant à deux stratégies différentes, à savoir, la poursuite de trajectoires et le suivi de chemins. S'agissant de la poursuite de trajectoire, nous avons proposé trois méthodes. La première est la linéarisation par bouclage dynamique, qui a permis d'obtenir une linéarisation exacte ainsi qu'un découplage du système qui peut être désormais commandé par un simple Proportionnel dérivé. La deuxième méthode est la commande par retour d'état non linéaire non stationnaire, dans laquelle nous avons présenté deux techniques, une basée sur la stabilité au sens de Lyapunov, l'autre exploite la propriété de stabilité des systèmes en cascade. Dans la troisième et dernière méthode nous avons utilisé une commande avec observateur d'orientation du robot. S'agissant du suivi de chemin, nous avons proposé une technique de commande non linéaire semblable à celle développée dans le cadre de la poursuite, où le robot, animé d'une vitesse linéaire donnée, suit un repère mobile situé sur le chemin. Pour la partie applicative, on a choisi la technique par suivi de chemin puisqu'elle correspond le mieux à notre contexte.

Par ailleurs, dans le cadre de ce travail, la vision a été utilisée, en se basant essentiellement sur les techniques de traitement d'images, pour l'extraction ou le calcul de la trajectoire qui correspond à l'objectif visé. Dans le cas du suivi de ligne, il s'agissait de détecter et d'extraire dans le plan image la ligne à suivre, puis à transformer le résultat vers l'espace cartésien. Pour ce qui est de la deuxième application, le robot étant amené à naviguer tout en évitant les éventuels obstacles, il a été question de construire un modèle de l'environnement dans lequel on distingue les obstacles de la région navigable. A partir de ce point on a procédé à la planification d'une trajectoire via la méthode des champs de potentiel. Etant donné la trajectoire de référence, nous avons appliqué au robot la technique de commande choisie, qui a donné des résultats satisfaisants, comme démontré par simulations.

Mots clés : robot mobile à roues, système non holonome, évitement d'obstacles, poursuite de trajectoire, suivi de chemin, linéarisation par bouclage dynamique, commande non linéaire non stationnaire, commande par observation de l'orientation, planification de trajectoire, vision artificielle, traitement d'images.

Abstract

This thesis addresses the problem of vision control applied to a non holonomic wheeled mobile robot, in two different types of tasks, line following, and navigation with obstacle avoidance. To do this, we have considered a system in which a camera is positioned at the front of the robot and points towards the ground while allowing an acceptable field of view.

First of all, we examined the control problem, for which we presented and simulated techniques belonging to two different strategies, namely, trajectory tracking and path following. For the trajectory tracking problem, we proposed three methods. The first one is the dynamic feedback linearization, which yielded an exact linearization and decoupling of the system that can now be controlled by a simple proportional derivative. The second method is the nonlinear non-stationary state feedback control in which we presented two techniques; one based on stability in the sense of Lyapunov, the other uses the property of stability of cascaded systems. In the third and final method we used an observer based control of robot orientation. For the path following problem, we proposed a nonlinear control technique similar to that developed for the trajectory tracking, where the robot, driven by a given linear velocity, follows a moving frame located on the path. For the application part, we chose this latter technique because it is best suited to our context.

Furthermore, as part of this work, the vision was used, mainly on base of image processing techniques, to extract or calculate the trajectory which corresponds to the task at hand. In the case of line following, the purpose was to detect and extract in the image plane the line to follow, and then transform it to the Cartesian space. For the second application, in which the robot is required to navigate while avoiding any obstacles; it was about to build a model of the environment in which we distinguish obstacles from navigable area. Based on this, we have processed a path planning via potential fields method. Given the reference trajectory, we applied the chosen control technique to the robot, which gave satisfactory results, as demonstrated by simulations.

Key words: wheeled mobile robot, non holonomic system, obstacle avoidance, trajectory tracking, path following, dynamic feedback linearization, flat system, nonlinear non stationary control, orientation observer based control, path planning, artificial vision, image processing.

Avant-propos

Les travaux présentés dans ce mémoire ont été effectués au Laboratoire de Commande des Processus du Département Automatique de l'Ecole Nationale Polytechnique d'Alger, sous la direction de Monsieur Mohamed TADJINE, Professeur à l'ENP.

Je tiens à remercier chaleureusement Mr. Mohamed TADJINE et lui adresser ma profonde gratitude pour avoir accepté de diriger mes travaux, pour m'avoir fait profiter de ses compétences scientifiques, ainsi que pour son esprit d'analyse et la qualité de ses conseils.

Je souhaite remercier Monsieur Farès BOUDJEMA d'avoir accepté la présidence du jury de soutenance, et Messieurs Lezhar NEZLI et Boualem HEMICI qui ont eu la gentillesse de faire partie de mon jury de soutenance et d'avoir pris le temps pour examiner cette thèse.

Mes remerciements vont également à Monsieur C. BENDJELFI du BEM, ainsi qu'à Messieurs M. BELOUCHERANI, D. BOUKHETALA, et M. BOUCHERIT de l'ENP, qui n'ont pas hésité à m'apporter leur soutien, notamment dans les démarches administratives.

J'aimerais particulièrement remercier Monsieur A. ZELLAGUI, Directeur du Centre de Recherche Développement-GN, pour sa confiance et son appui qui m'a permis de continuer dans cette voie, qui est celle de la connaissance et du savoir.

Je n'oublie évidemment pas mes collègues Khalil, Mesbah, Hacene, et amis Abdelmoumene et Kamel pour leur encouragement et leur soutien moral durant cette épreuve.

Je remercie particulièrement ma femme Djamila, pour son soutien et sa patience.

Enfin, je veux remercier très sincèrement mes deux parents Benyounes et Saïda, et je souhaite qu'ils soient fiers de leur fils.

Table des matières

Introduction générale	1
Chapitre 1 : Contexte et problématique	
1.1. Cadre général	5
1.2. Problématique	5
1.3. Approche adoptée pour les applications choisies	8
1.3.1. Application 1 : suivi de ligne	9
1.3.2. Application 2 : navigation avec évitement d'obstacles	10
1.3.3. Exécution de la trajectoire de référence	12
1.4. Hypothèses de travail	12
1.5. Organisation du mémoire	13
Chapitre 2 : Modélisation et état de l'art sur la commande des robots mobiles à roues non holonomes	
2.1. Introduction	16
2.2. Modélisation des robots mobiles à roues	16
2.2.1. Rappels sur la notion de non holonomie	16
2.2.2. La contrainte de roulement sans glissement	18
2.2.3. Modélisation d'un robot de type unicycle	20
2.2.3.1. Description	20
2.2.3.2. Modèle cinématique	21
2.2.4. Modélisation d'un robot de type tricycle	23
2.2.4.1. Description	23
2.2.4.2. Modèle cinématique	24
2.2.5. Modélisation d'un robot de type voiture	25
2.2.5.1. Description	25
2.2.5.2. Modèle cinématique	25
2.2.6. Modèle cinématique généralisé	26

3.3.4.1. Modèle cinématique de l'erreur de poursuite	47
3.3.4.2. Commande n° 1	47
3.3.4.3. Commande n° 2	48
3.3.4.4. Génération de la trajectoire et du modèle de référence	49
3.3.5. Commande avec observateur de l'orientation du robot	51
3.4. Suivi de chemin	52
3.4.1. Formulation du problème du suivi de chemin	52
3.4.2. Synthèse de la commande	54
3.5. Modèle dynamique du robot et synthèse du régulateur interne	56
3.6. Simulations et résultats	57
3.6.1. Technique de linéarisation par bouclage	58
3.6.2. Technique de commande non stationnaire continue	66
3.6.3. Commande avec observateur de l'orientation du robot	70
3.6.4. Commande par suivi de chemin	71
3.6.5. Choix de la technique la plus adaptée à notre contexte	74
3.7. Conclusion	74

Chapitre 4 : Introduction à la vision artificielle en robotique mobile

4.1. Introduction	76
4.2. Détection et modélisation de l'environnement par vision	76
4.2.1. Rappels sur la notion d'image	76
4.2.1.1. Image numérique	77
4.2.2. Les capteurs de vision	77
4.2.2.1. Types de caméras	77
4.2.2.2. Modélisation d'une caméra	78
4.2.2.2.1. Présentation du modèle sténopé	78
4.2.2.2.2. Matrice de projection perspective	79
4.2.2.2.3. Transformation affine caméra/image	80
4.2.2.2.4. Paramètres intrinsèques	81
4.2.2.2.5. Paramètres extrinsèques	82
4.2.2.3. Calibrage d'une caméra	83
4.2.3. Techniques de traitement d'image en robotique mobile	83
4.2.3.1. Techniques de bas-niveau	84
4.2.3.1.1. Lissage local	84
4.2.3.1.1.1. Moyennage	84

4.2.3.1.1.2. Filtres médians	86
4.2.3.1.1.3. Autres filtres	86
4.2.3.1.2. Amélioration d'images	86
4.2.3.1.2.1. Modification d'histogramme	87
4.2.3.1.3. Détection de contours	88
4.2.3.1.3.1. Approches classiques	88
4.2.3.1.3.2. Approches analytiques	90
4.2.3.2. Techniques de segmentation	92
4.2.3.3. Techniques de haut-niveau	92
4.3. Planification de trajectoire et évitement d'obstacles	93
4.3.1. Méthode de champs de potentiel	93
4.3.1.1. Présentation de l'algorithme	94
4.3.1.1.1. Potentiel attractif	95
4.3.1.1.2. Potentiel répulsif	95
4.3.1.1.3. Planification de trajectoire induite par les champs de potentiel	96
4.3.2. Autres méthodes de planification	97
4.3.2.1. Méthode Steering Angle Field	97
4.3.2.2. Méthode de la fenêtre dynamique	97
4.3.2.3. Méthode du diagramme de proximité	97
4.4. Conclusion	97

Chapitre 5 : Simulations et résultats

5.1. Introduction	99
5.2. Architecture de commande	99
5.3. Description du système	100
5.3.1. Description de la caméra utilisée	101
5.3.2. Transformation de passage de l'espace Image vers l'espace cartésien	101
5.4. Calibrage de la caméra.	103
5.5. Présentation de la chaîne de traitement	103
5.5.1. Acquisition d'images	104
5.5.2. Phase de prétraitements	104
5.5.3. Détection de contours	104
5.5.4. Transformation de coordonnées du plan image vers l'espace cartésien	105
5.5.5. Affinage de trajectoire par courbes de Bézier	105
5.6. Application au suivi d'une ligne tracée sur le sol	106

5.6.1. Chaîne de traitement d'image	106
5.6.1.1. Prétraitements	106
5.6.1.2. Détection de contours	107
5.6.1.3. Extraction de la ligne dans le plan image	107
5.6.2. Transformation de la ligne vers l'espace cartésien	108
5.6.3. Simulation du suivi de la ligne	109
5.7. Application à l'évitement d'obstacle	111
5.7.1. Chaîne de traitement d'image	111
5.7.1.1. Prétraitements	111
5.7.1.2. Détection de contours	112
5.7.1.3. Segmentation de la scène.....	112
5.7.1.4. Décomposition en cellules.....	113
5.7.2. Transformation de la scène vers l'espace cartésien	113
5.7.3. Planification de trajectoire par champs de potentiel	114
5.7.4. Simulation du suivi de la trajectoire de référence.....	115
5.8. Conclusion.....	117
Conclusion générale	119
Annexes	
A Techniques de linéarisation par bouclage	122
B Modélisation dynamique du robot unicycle	130
Bibliographie	

Table des figures

Figure I.1 : Shakey, le premier robot mobile (1969)	1
Figure I.2 : Exemples d'applications en robotique mobile	2
Figure 1.1 : Schéma fonctionnel d'un système de commande en robotique mobile	8
Figure 1.2 : Robot mobile dans une application de suivi de ligne	9
Figure 1.3 : Etapes de calcul de la trajectoire désirée dans une application de suivi de ligne	10
Figure 1.4 : Robot mobile dans une application d'évitement d'obstacles	10
Figure 1.5 : Etapes de calcul de la trajectoire désirée dans une application d'évitement d'obstacles	10
Figure 1.6 : Architecture de commande d'un robot mobile de type unicycle	12
Figure 2.1 : Roue verticale qui roule sans glisser	18
Figure 2.2 : Exemples de robots de type unicycle	21
Figure 2.3 : Robot mobile de type unicycle	21
Figure 2.4 : Exemples de robots de type tricycle	24
Figure 2.5 : Robot mobile de type tricycle	24
Figure 2.6 : Exemples de robots de type voiture	25
Figure 2.7 : Robot mobile de type voiture	26
Figure 2.8 : Classification des techniques de commande des robots mobiles à roues	30
Figure 3.1 : Schématisation du problème de poursuite de trajectoire	36
Figure 3.2 : Système non linéaire bouclé équivalent à un système linéaire	40
Figure 3.3 : Architecture de commande dédiée à la poursuite de trajectoire par la méthode de linéarisation par bouclage	45
Figure 3.4 : Erreurs de poursuite par rapport à un robot virtuel	46
Figure 3.5 : Architecture de commande dédiée à la poursuite de trajectoire par la méthode directe	49
Figure 3.6 : Schéma-bloc de l'architecture de commande d'un unicycle avec observateur de l'orientation	51

Figure 3.7 : Suivi de chemin pour l'unicycle	52
Figure 3.8 : Architecture de commande dédiée au suivi de chemin	56
Figure 3.9 : Schéma bloc de la boucle interne avec réglage PI	57
Figure 3.10 : Poursuite réussie. (a) trajectoire circulaire. (b) en forme de huit. (c) ligne droite .	59
Figure 3.11 : Poursuite de trajectoire. (a) trajectoire du robot. (b) erreurs de poursuite	60
Figure 3.12 : Allure des entrées de commande du robot	61
Figure 3.13 : Allure des signaux électriques appliqués aux moteurs	62
Figure 3.14 : Système de commande soumis aux différentes perturbations	62
Figure 3.15 : Comportement du robot pour différentes amplitudes du bruit	63
Figure 3.16 : Comportement du robot soumis à une perturbation échelon d'amplitude $m=0,24$. 63	
Figure 3.17 : Allure des entrées de commande du robot	64
Figure 3.18 : Allure des signaux électriques appliqués aux moteurs	64
Figure 3.19 : Comportement du robot soumis à un bruit de modèle blanc de variance $\sigma = 10^{-4}$...	65
Figure 3.20 : Allure des entrées de commande et des signaux électriques du robot	65
Figure 3.21 : Echec de la poursuite de ligne droite pour la commande n°2	66
Figure 3.22 : Comportement du robot et erreurs de poursuite. (a) cde n°1. (b) cde n°2	67
Figure 3.23 : Allure des entrées de commande du robot. (a) cde n°1. (b) cde n°2	68
Figure 3.24 : Allure des signaux électriques appliqués aux moteurs. (a) cde n°1. (b) cde n°2 ..	69
Figure 3.25 : Comportement du robot et erreurs de suivi	70
Figure 3.26 : Erreur d'estimation	70
Figure 3.27 : Suivi réussi. (a) trajectoire circulaire. (b) en forme de huit. (c) ligne droite	71
Figure 3.28 : Comportement du robot et erreurs de suivi	72
Figure 3.29 : Allure des entrées de commande du robot	73
Figure 4.1 : Technologies utilisées dans les caméras numériques	78
Figure 4.2 : Modèle sténopé d'une caméra	79
Figure 4.3 : Image dans le repère UV	81
Figure 4.4 : Modèle sténopé et système de coordonnées	82
Figure 4.5 : Application d'un filtre sur un voisinage 3x3	85
Figure 4.6 : Exemples de filtres par détection de gradient	89
Figure 4.7 : Champs de potentiel attractif et répulsif	94
Figure 4.8 : Evitement d'obstacles par la méthode de champs de potentiel	96
Figure 5.1 : Architecture adoptée pour la commande par vision du robot unicycle	99
Figure 5.2 : Description du système robot-caméra	100
Figure 5.3 : Modèle géométrique de la caméra	101
Figure 5.4 : Exemples de prises effectuées pour le calibrage de la caméra	103

Figure 5.5 : Construction d'une courbe de Béziérs	105
Figure 5.6 : Application du suivi de ligne. (a) observateur externe. (b) vue par le robot	106
Figure 5.7 : Prétraitements de la scène	106
Figure 5.8 : Détection de contours. (a) Sobel. (b) Prewitt. (c) Canny	107
Figure 5.9 : Extraction de la ligne dans le plan image	107
Figure 5.10 : Génération de la trajectoire dans l'espace cartésien	108
Figure 5.11 : Simulation du suivi de ligne. (a) trajectoire du robot. (b) erreurs de suivi	109
Figure 5.12 : Allure des entrées de commande du système	110
Figure 5.13 : Allure des signaux électriques appliqués aux moteurs	110
Figure 5.14 : Application d'évitement d'obstacles. (a) observateur externe. (b) vue par le robot	111
Figure 5.15 : Prétraitements de la scène	111
Figure 5.16 : Détection de contours	112
Figure 5.17 : Segmentation de la scène	113
Figure 5.18 : Décomposition de la scène en cellules	113
Figure 5.19 : Transformation de la scène vers l'espace cartésien	114
Figure 5.20 : Planification de trajectoire par la méthode de champs de potentiel	114
Figure 5.21 : Simulation du suivi de ligne. (a) trajectoire du robot. (b) erreurs de suivi	115
Figure 5.22 : Allure des entrées de commande du système	116
Figure 5.23 : Allure des signaux électriques appliqués aux moteurs	116

Introduction générale

Il existe diverses définitions pour désigner un robot mobile, mais d'une manière générale, on peut le présenter comme une machine capable de réaliser de façon autonome un déplacement d'un point à un autre dans un environnement non parfaitement connu à priori. Pour se faire, le robot doit disposer des fonctionnalités de perception, de décision sur le mouvement à effectuer, et de réalisation de ce mouvement.

L'histoire nous raconte que l'intérêt pour les robots mobiles est intervenu peu de temps après que les robots manipulateurs eurent été introduits, et on a pu assister à la création du premier robot mobile doté d'une intelligence artificielle et capable d'interagir avec son environnement vers la fin des années 60.

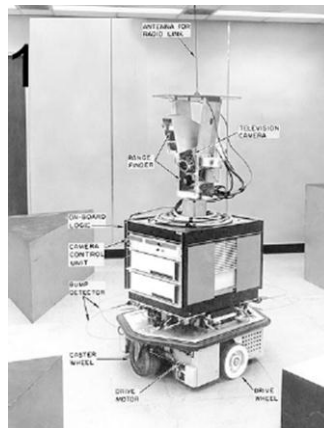


Figure I.1 : Shakey, le premier robot mobile (1969)

Cependant, durant les deux décennies qui suivirent, et contrairement aux manipulateurs qui ont envahi plusieurs secteurs de l'industrie, les robots mobiles étaient encore contraints à vivre dans les laboratoires, car jusqu'alors, ils ne pouvaient évoluer que dans un environnement adapté à leurs besoins, parfaitement connu et immuable. Dans cet environnement, soit une

mission était planifiée à partir d'un modèle complet de l'environnement, soit le robot se contentait de suivre un mécanisme de guidage.

Au cours des années 90, grâce aux avancées technologiques qu'a connues l'électronique, notamment la miniaturisation et le développement de plateformes intégrées, de nombreux laboratoires à travers le monde ont pu travailler sur la robotique mobile lui consacrant un nombre très important de thèmes de recherche, ce qui a conduit à son développement. Les objectifs visés étaient bien différents, les robots mobiles pouvaient se mouvoir dans des environnements plus complexes, partiellement connus, et qui peuvent se modifier avec le temps.

De nos jours, la robotique mobile connaît une réelle expansion avec des applications de plus en plus attrayantes, dont on peut citer à titre d'exemples :

- l'exploration planétaire ;
- l'exploration ou l'intervention dans des milieux hostiles pour l'être humain (zone radioactive, polaire, fonds sous-marins, présence de feu...), ou simplement difficiles d'accès ;
- les opérations militaires délicates telles que le déminage ou l'accompagnement de troupes ;
- les missions de maintenance, de surveillance ou de protection ;
- l'industrie (ex. les transports routiers) ;
- les loisirs.



Figure I.2 : Exemples d'applications en robotique mobile

Dans l'industrie, les applications concernent essentiellement le transport ou la distribution, que ce soit dans les usines, les mines, les hôpitaux ou les ateliers. On peut prendre l'exemple du convoyage dans un milieu souvent encombré qui consiste au stockage ou à l'expédition de pièces de rechange, de médicaments, de déchets, ou encore le nettoyage automatique des sols. On peut aussi citer le convoyage automatique de véhicules utilisé dans les transports routiers.

Dans toutes ces applications, les robots sont amenés à interagir avec le milieu dans lequel ils évoluent pour pouvoir accomplir leurs missions. De plus, il s'avère vital que les robots, en marge de l'objectif fixé par leur mission, puissent détecter et éviter les éventuels obstacles, statiques ou dynamiques, qui pourraient entraver leur déplacement.

En robotique mobile, l'interaction avec l'environnement repose sur l'utilisation de capteurs dits extéroceptifs qui permettent au robot de créer une représentation de cet environnement, chose qu'il ne peut effectuer via les capteurs proprioceptifs qui ne donnent une information que sur son état interne (ex. capteur de vitesse de rotation des roues).

Plusieurs types de capteurs extéroceptifs peuvent être utilisés, dont les plus répandus sont les capteurs à ultrasons, les télémètres laser, et les caméras. Ces dernières sont de loin les plus utilisées, du fait qu'elles se rapprochent des méthodes utilisées par les êtres humains, et grâce aussi à la richesse des informations qu'elles peuvent contenir comparativement aux autres types de capteurs.

Dans le cadre de ce mémoire, nous nous sommes intéressés à la problématique de la commande d'un robot mobile doté d'un système de vision et exécutant des missions de type suivi de ligne et navigation avec évitement d'obstacles.

Chapitre 1

Contexte et problématique

Dans ce premier chapitre, nous exposons le cadre général dans lequel s'inscrit le travail de cette thèse, la problématique traitée, ainsi que l'approche proposée.

1.1. Cadre Général

La robotique mobile est un domaine de recherche pluridisciplinaire qui fait intervenir à la fois plusieurs domaines techniques, allant de la mécanique, passant par l'automatique, l'informatique, l'intelligence artificielle, et la perception par ordinateur. De nos jours, cette branche connaît un engouement sans précédent et se retrouve dans plusieurs secteurs d'activité, en industrie manufacturière, dans le spatial, le militaire, l'automobile et même les loisirs.

Dans la majeure partie de ces applications, les robots mobiles à roues sont privilégiés, puisque ces derniers présentent des caractéristiques très intéressantes inégalées par les autres types de robots, dont on peut citer, la faible complexité de conception, la bonne manœuvrabilité et la grande capacité à supporter et à transporter des charges importantes.

Dans la littérature, la commande de ce type de robots a reçu un intérêt particulier, et un nombre important de travaux lui ont été consacrés. Ces travaux ont découlé sur plusieurs approches, qui dépendent non seulement de la tâche à effectuer, mais aussi du type de robot considéré.

Par ailleurs, la plupart des applications robotiques d'aujourd'hui impliquent une interaction du robot avec le milieu de son évolution puisqu'elles consistent généralement, à faire naviguer le robot dans un environnement dynamique, partiellement connu ou inconnu, et qui peut être fortement encombré. Parmi les types de navigation qui existent, la navigation visuelle est certainement la plus utilisée. Celle-ci peut être définie comme étant l'ensemble des fonctions permettant à un robot mobile de planifier et d'exécuter des déplacements dans un environnement quelconque, perçu par une ou plusieurs caméras embarquées.

Ce mémoire a pour cadre général la commande et la navigation visuelle en robotique mobile, et se focalise sur la commande d'un robot à roues de type unicycle muni d'une caméra monoculaire, dans des applications de suivi de ligne et d'évitement d'obstacles.

1.2. Problématique

La problématique que nous essayons de traiter dans le cadre de cette thèse concerne la commande d'un robot mobile à roues non holonome évoluant dans un environnement intérieur inconnu à priori, où la vision est utilisée pour la planification de la trajectoire permettant de satisfaire les objectifs indiqués. Cette problématique est loin d'être facile à résoudre et constitue encore un défi pour les chercheurs dans ce domaine.

En effet, hormis les contraintes d'ordre technologique, liées entre autres, à l'imprécision des capteurs et à l'autonomie énergétique, l'élaboration de la commande des robots mobiles s'avère une tâche très complexe, lesquels, comparés aux bras manipulateurs (prévus pour

travailler exclusivement dans des espaces connus et de manière répétitive), sont destinés à évoluer de manière autonome dans des environnements peu ou pas structurés.

Une des contraintes majeures qui pèsent sur la commande des robots mobiles réside dans le fait que la plupart de ces robots font partie des systèmes non holonomes, c'est-à-dire des systèmes ayant moins de degrés de commande que de degrés de liberté. Dans le cas des robots à roues, la non holonomie est due à la condition de roulement sans glissement des roues sur le sol, qui se traduit en pratique, par le fait qu'ils ne peuvent pas se déplacer latéralement de manière instantanée.

Bien que commandables, ces systèmes sont fortement non linéaires avec des caractéristiques qui les rendent difficiles à commander dans certaines situations [51]. Tout d'abord, le linéarisé autour de tout point d'équilibre n'est pas commandable, ce qui empêche l'utilisation des techniques de commande linéaire pour stabiliser asymptotiquement une configuration fixe. De plus, selon Brockett [9], la stabilisation asymptotique d'une configuration fixe est impossible à obtenir via un retour d'état stationnaire continu. Toutefois, le linéarisé le long de certaines trajectoires de référence est commandable, ce qui permet, dans ce cas, de construire des retours d'état linéaires asymptotiquement stabilisants.

Cette situation a contraint le développement de techniques et de lois de commande qui dépendent de la tâche visée, ou encore, de la trajectoire à exécuter, d'où la classification des travaux effectués dans ce domaine selon le type de trajectoire, à savoir :

- la stabilisation de points fixes ;
- la poursuite de trajectoires (ou stabilisation de trajectoires) ;
- et le suivi de chemins.

En pratique, on se trouve souvent dans une situation où la tâche à exécuter correspond à une poursuite de trajectoire ou à un suivi de chemin, car généralement, il ne suffit pas de spécifier la configuration finale que doit atteindre le robot mobile, mais il est aussi indispensable de maîtriser le mouvement qu'il devra effectuer pour atteindre cette configuration (ex. : navigation avec évitement d'obstacles).

Dans ce cas, si la trajectoire n'est pas indiquée, il est nécessaire de passer par une étape préalable de planification de cette trajectoire, qui consiste à déterminer un mouvement permettant au robot de se déplacer entre deux configurations données tout en respectant un certain nombre de contraintes et de critères, lesquels sont dus à plusieurs facteurs et dépendent du type de robot considéré, de l'environnement, et de la nature de la tâche à exécuter. Pour ce qui est des contraintes relatives au robot, celles-ci concernent sa géométrie, sa cinématique et sa dynamique et leur prise en compte peut être complexe selon l'architecture considérée (simple plateforme mobile, véhicules à remorques, main articulée, etc.), quant aux contraintes relatives à

l'environnement, elles concernent essentiellement la présence potentielle d'obstacles fixes ou mobiles dans cet environnement.

D'autre part, pour évoluer dans un environnement qui est partiellement ou non connu et/ou dynamique, le robot doit pouvoir interagir avec cet environnement, dans ce cas, les capteurs proprioceptifs s'avèrent insuffisants, et il devient nécessaire d'intégrer dans la boucle de commande de l'information issue de capteurs extéroceptifs permettant d'acquérir une connaissance plus au moins précise de l'environnement d'évolution, en d'autres termes, de construire un modèle du monde.

Ce modèle peut être représenté de différentes manières et avec différents niveaux de détail, en fonction de la tâche à réaliser, du type d'environnement, ainsi que de la résolution et du champ de perception des capteurs montés sur le robot [27].

L'utilisation des caméras pour percevoir l'environnement (vision artificielle) est une méthode attractive, car elle se rapproche des méthodes utilisées par l'être humain, aussi, comparées aux autres types de capteurs, les caméras offrent l'information la plus riche et la plus dense sur l'environnement : résolution, information géométrique et radiométrique, etc. En robotique mobile, les caméras sont de plus en plus utilisées et connaissent plusieurs applications, entre autres : le suivi d'objets et l'évitement d'obstacles, etc.

Cependant, la richesse des informations contenues dans les images fournies par les caméras avec les artefacts qu'elles peuvent contenir (reflets, zones saturées, occultations, etc.), font que la tâche de reconnaissance et d'interprétation de l'information visuelle soit extrêmement compliquée, et constitue encore une voie de recherche très explorée.

Ajouter à cela, la difficulté de la mesure de la configuration courante d'un robot mobile. En effet, la qualité de l'information issue des capteurs proprioceptifs se dégrade continuellement au fur et à mesure des déplacements à cause de l'intégration temporelle des mesures effectuées par ces capteurs, d'où le recours dans certains cas, à l'utilisation de capteurs extéroceptifs, notamment dans une logique de fusion de données multi-capteurs, ou encore à l'estimation de la configuration via des observateurs.

En se référant aux points évoqués, on peut situer les développements relatifs à notre problématique sur trois niveaux différents, à savoir :

- 1- Perception et interprétation de l'environnement ;
- 2- Planification de la trajectoire correspondant à l'objectif visé ;
- 3- Commande du robot.

Le schéma de la figure (1.1) permet d'illustrer ces étapes au sein d'un système de commande d'un robot mobile.

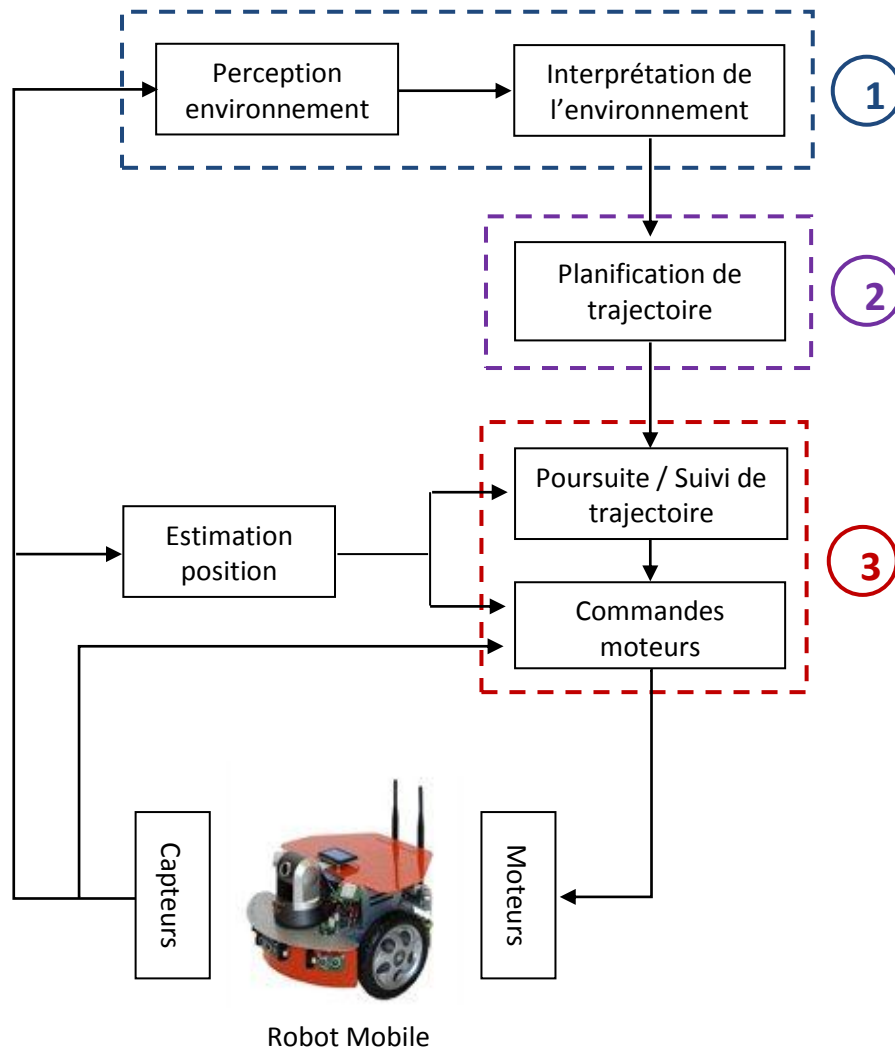


Figure 1.1 : Schéma fonctionnel d'un système de commande en robotique mobile

1.3. Approche adoptée pour les applications choisies

Pour aborder la problématique qu'on s'est fixée, nous avons choisi des applications dans lesquelles un robot de type unicycle, muni d'une caméra monoculaire pointant vers le sol avec un angle lui permettant un champ de vision acceptable, réalise des missions de types suivi de ligne et évitement d'obstacles, dans un environnement intérieur inconnu à priori.

L'approche adoptée consiste à faire exécuter au robot un algorithme assurant la réalisation successive des tâches suivantes :

- acquisition d'une image 2D de la scène courante ;
- traitements d'image : redressement, filtrage, etc. ;
- interprétation ou extraction d'une description de la scène. Il s'agit d'identifier dans l'image, les entités de l'environnement utiles pour la navigation (ligne à suivre, zones

navigables, obstacles). La méthode utilisée repose essentiellement sur les techniques de détection de contours ;

- transformation de la scène de l'espace image vers l'espace 3D (construction d'un modèle du monde) ;
- planification de la trajectoire requise par la tâche que doit exécuter le robot ;
- exécution de la trajectoire : via les techniques de commande en boucle fermée.

Afin de permettre une navigation continue du robot dans son environnement, l'algorithme décrit précédemment sera exécuté en boucle, avec une période qui sera déterminée suivant certains paramètres, ex : la profondeur du champ visuel du robot (donnée par l'angle d'inclinaison de la caméra), la puissance de calcul, la résolution des capteurs, etc.

1.3.1. Application 1 : suivi de ligne

Dans cette application, il s'agit de faire suivre au robot une ligne que nous avons tracée auparavant sur le sol. La trajectoire décrite par cette ligne doit être admissible, c'est-à-dire réalisable par le robot (figure 1.2).

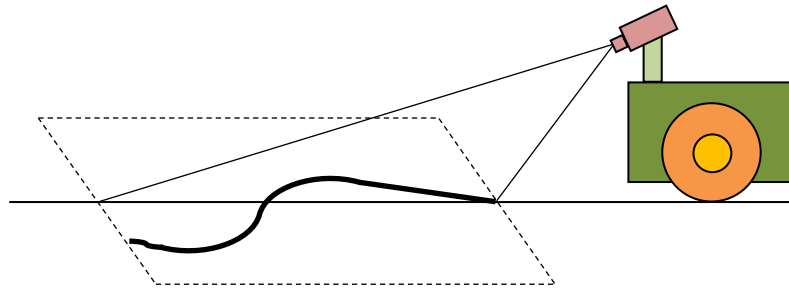


Figure 1.2 : Robot mobile dans une application de suivi de ligne

Dans ce cas précis, il n'y a pas lieu de faire une planification de trajectoire proprement dite, et cette étape se résumera à transformer la ligne extraite dans l'espace image en une trajectoire dans l'espace 3D, avec quelques traitements supplémentaires (échantillonnage, interpolation) pour la rendre exploitable par la partie commande, responsable de l'exécution et du suivi de la trajectoire.

L'architecture de la partie du système permettant d'obtenir la trajectoire de référence est donnée dans le schéma suivant (figure 1.3).

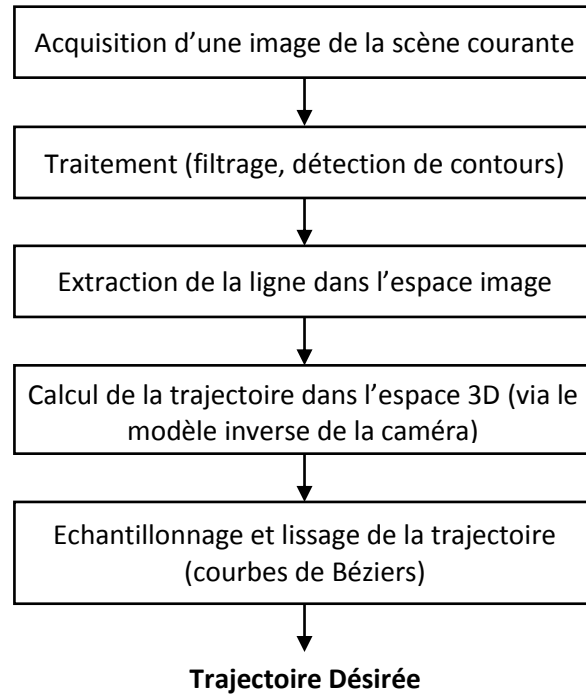


Figure 1.3: Etapes de calcul de la trajectoire désirée dans une application de suivi de ligne

1.3.2. Application 2 : navigation avec évitement d'obstacles

Dans cette application, il s'agit de faire naviguer un robot en évitant des obstacles, comme illustré dans la figure (1.4).

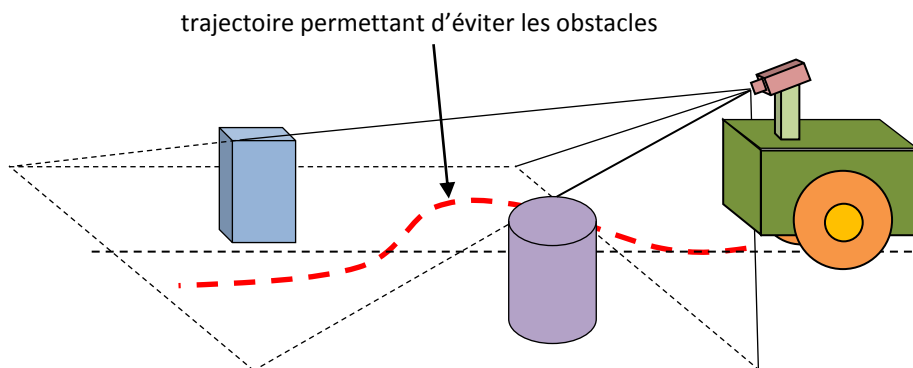


Figure 1.4: Robot mobile dans une application d'évitement d'obstacles

Une fois que le robot ait acquis une image de la scène courante, cette image passe par des prétraitements (filtrage, réhaussement de contraste, ...) la préparant pour l'étape d'interprétation, qui permettra de distinguer, dans l'espace image, la zone libre, de celle correspondant aux obstacles. Pour cette étape, nous avons utilisé une technique de segmentation basée sur la détection des contours.

A partir de ce point, on procèdera, dans la partie planification, à la transformation vers l'espace 3D des coordonnées des entités détectées, puis au calcul d'une trajectoire évitant les obstacles.

Plusieurs techniques peuvent être utilisées pour résoudre cette dernière problématique (voir chapitre 4.3). Dans le cadre de ce travail, nous avons choisi la méthode de champs de potentiel, dans laquelle une trajectoire est calculée suivant le champ résultant de la sommation, en tout point, du champ attractif, causé par la configuration cible, et de champs répulsifs causés par les obstacles. Dans notre cas, puisqu'il s'agit de navigation continue, nous avons choisi la configuration cible que doit atteindre le robot comme étant le plus loin point de la scène accessible par le robot.

De la même manière que pour l'application précédente, on présente l'architecture de la partie du système permettant d'obtenir la trajectoire de référence (figure 1.5).

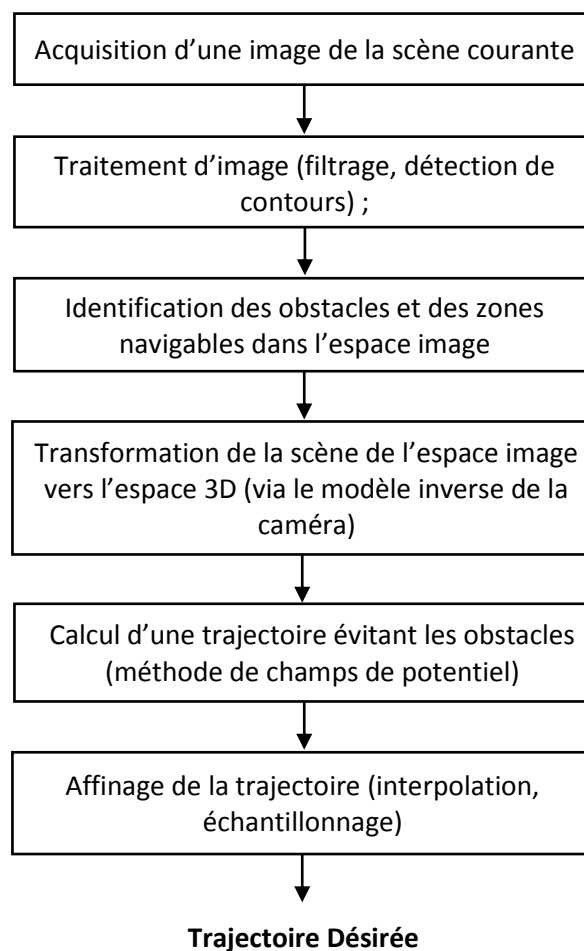


Figure 1.5: Etapes de calcul de la trajectoire désirée dans une application d'évitement d'obstacles

1.3.3. Exécution de la trajectoire de référence

Pour ce qui est de l'exécution de la trajectoire, cette étape est réalisée en recourant aux techniques de commande en boucle fermée, qui permettent de garantir une certaine précision du suivi de la trajectoire désirée.

Pour se faire, nous avons utilisé des techniques de commande non linéaire, dont les développements sont présentés dans le chapitre 3.

Pour les deux applications, la stratégie de commande implémentée repose sur la synthèse d'un régulateur en cascade (figure 1.6).

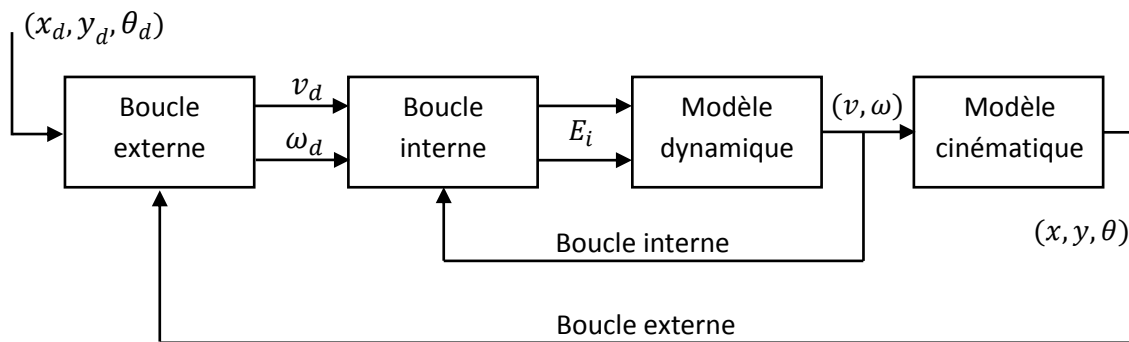


Figure 1.6: Architecture de commande d'un robot mobile de type unicycle

La boucle externe permet de calculer les vitesses linéaire et angulaire (v_d, ω_d) que doit avoir le robot pour exécuter la trajectoire désirée (x_d, y_d, θ_d).

Quant à la boucle interne, qui est commune à toutes les techniques de commande présentées dans le chapitre 3, elle permet de générer les signaux électriques (E_i) à appliquer au niveau des moteurs des roues motrices du robot, pour assurer que ce dernier suive le profil de vitesse indiqué par la boucle externe. Le régulateur utilisé dans cette boucle est de type PID.

1.4. Hypothèses de travail

Les développements réalisés dans le cadre de cette thèse, s'appuient sur les hypothèses suivantes :

- les roues du robot sont rigides et roulent sans glisser sur le sol ;
- le centre de gravité du robot est situé au milieu de l'axe passant par les roues motrices ;
- le sol est considéré comme parfaitement plat ;
- les obstacles sont supposés statiques et inconnus à l'avance. Le cas où l'environnement change, peut être pris en charge sous certaines conditions (chap. 5).

1.5. Organisation du mémoire

Comme énoncé dans la problématique, les travaux effectués dans le cadre de ce mémoire portent sur trois aspects : la perception et l'interprétation de l'environnement, la planification de trajectoire, et la commande du robot.

L'axe principal de notre étude a porté sur ce dernier point, i.e. la commande des robots mobiles à roues non holonomes, pour laquelle nous avons consacré deux chapitres. Puis nous sommes passés à la présentation des outils et méthodes utilisées pour la perception et l'interprétation de l'environnement dans le cadre de la vision artificielle, suivies par les techniques de planification de trajectoire utilisées en robotique mobile.

Plus précisément, ce mémoire est organisé comme suit :

Le chapitre 2 se présente sous forme d'un état de l'art sur la commande des robots mobiles à roues non holonomes, dans lequel nous avons introduit quelques notions théoriques associées aux robots à roues et leur modélisation, ainsi que les différentes approches proposées pour la commande de ces robots.

Dans le chapitre 3, nous avons présenté deux stratégies de commande utilisées en robotique mobile, à savoir : la poursuite de trajectoire et le suivi de chemin, pour lesquelles nous avons implémenté et présenté les résultats de simulation de quelques techniques que nous avons appliqué sur un modèle du robot unicycle. S'agissant de la poursuite de trajectoire, nous avons proposé trois techniques : la linéarisation par bouclage dynamique qui permet d'obtenir un système équivalent complètement linéaire et découplé, la commande par retour d'état non linéaire non stationnaire, dite méthode directe, et une troisième commande dans laquelle cette dernière technique est couplée avec un observateur de l'orientation du robot qui est supposée non mesurée. Pour ce qui est du suivi de chemin, nous avons proposé une technique de commande non linéaire semblable à celle développée dans le cadre de la poursuite, dans laquelle le robot, animé d'une vitesse linéaire donnée, suit un repère mobile situé sur le chemin.

Dans le chapitre 4, nous avons introduit quelques notions fondamentales sur la perception basée sur la vision par ordinateur, notamment celles relatives aux caméras et leur modélisation, ainsi que les différentes techniques de traitement d'images permettant l'interprétation de l'information visuelle. Aussi, nous avons présenté les différentes méthodes utilisées pour la planification de trajectoire en robotique mobile dans le cadre de l'évitement réactif d'obstacles, en l'occurrence la méthode des champs de potentiel, que nous avons utilisé pour notre application.

Le chapitre 5 a été consacré à la mise en œuvre de l'approche adoptée pour la navigation visuelle, représentée par les deux applications choisies, où nous avons d'abord commencé par donner une description du système complet (robot + caméra), puis nous avons présenté en détail

la chaîne de traitement complète relative à la planification de trajectoire par vision, depuis l'acquisition d'image de la scène observée, jusqu'à l'obtention de la trajectoire de référence dans l'espace 3D. Par la suite, cette chaîne de traitement a été appliquée pour les deux cas de figures préconisés dans notre approche, à savoir, le suivi de ligne et la navigation avec évitement d'obstacle. Cette partie a été réalisée sous MATLAB, moyennant une caméra de type webcam. Enfin, la partie commande et exécution de trajectoire est testée et validée en environnement de simulation MATLAB.

Finally, this thesis is completed by a conclusion which constitutes a recapitulation of the main results obtained as well as research perspectives in the domain of the control of mobile robots with non-holonomic wheels, notably the techniques based on vision.

Chapitre 2

**Modélisation et état de l'art sur la commande
des robots mobiles à roues non holonomes**

2.1. Introduction

Ce chapitre est dédié à la modélisation et la présentation des différentes techniques de commande des robots mobiles à roues non holonomes. Tout d'abord nous donnons quelques notions fondamentales concernant la cinématique relative à ce type de robots, notamment la notion de non holonomie et la condition de roulement sans glissement des roues sur le sol, qui permettent d'obtenir une écriture simplifiée des équations décrivant le mouvement de ces robots, dont nous présentons les modèles cinématiques relatifs aux plateformes les plus répandus, à savoir, l'unicycle, le tricycle et le robot de type voiture.

Ensuite nous présentons un état de l'art des différentes techniques de commande utilisées pour piloter ces robots, lesquelles sont classées selon le type de stabilisation qu'on désire obtenir, soit la stabilisation de configurations fixes, la stabilisation de trajectoires non stationnaires, et le suivi de chemin.

2.2. Modélisation des robots mobiles à roues non holonomes

2.2.1. Rappels sur la notion de non holonomie

Définition 2.1

Étant donné un système mécanique dont l'espace de configuration est une variété différentielle Q de dimension n , on appelle contrainte cinématique une contrainte sur les vitesses du type :

$$\langle a(q), \dot{q} \rangle = 0, \forall q \in \mathcal{U}(q_0) \quad (2.1)$$

avec $\mathcal{U}(q_0)$ un voisinage du point $q_0 \in Q$ et $a(\cdot)$ une forme différentielle (ou champ de covecteurs) de Q dans $\mathbb{R}^n$.

Définition 2.2

Une contrainte cinématique $\langle a(q), \dot{q} \rangle = 0$ est dite contrainte intégrable s'il existe une fonction régulière $h : Q \rightarrow \mathbb{R}$ telle que $dh = a$. Dans ce cas la contrainte est équivalente à la relation statique $h(q) = cte$.

Notons que la propriété d'intégrabilité est une propriété locale. Dans la pratique, un critère qui permet de repérer l'existence de contraintes intégrables est donné par le résultat suivant.

Théorème 2.1 (théorème d'intégrabilité de Frobenius)

Soient un système mécanique sur une variété Q de dimension n , soumis à $k \leq n$ contraintes cinématiques indépendantes

$$\langle a_i(q), \dot{q} \rangle = 0, \forall q \in \mathcal{U}(q_0), \forall i = 1, \dots, k \quad (2.2)$$

avec $A(q) = (a_1(q), \dots, a_k(q))$ de rang k pour tout $q \in \mathcal{U}(q_0)$

et $\{X_1, \dots, X_{n-k}\}$ une famille de champ de vecteurs sur $\mathcal{U}(q_0)$ orthogonaux aux a_i . Si $\text{Lie}(X_i)(q)$ est de dimension $n - p$ pour tout $q \in \mathcal{U}(q_0)$ alors il existe p contraintes intégrables (au sens où l'on peut trouver des fonctions scalaires $h_1, \dots, h_p$ indépendantes et des fonctions $\lambda_{1,1}, \dots, \lambda_{p,n}$ telles que $dh_i(q) = \sum_{j=1}^n \lambda_{i,j}(q)a_j(q)$).

Définition 2.3

Un système non holonome S est un système mécanique soumis à un ensemble de contraintes cinématiques :

$$\langle a_i(q), \dot{q} \rangle = 0, \forall q \in \mathcal{U}(q_0), \forall i = 1, \dots, k \quad (2.3)$$

indépendantes et non intégrables (au sens où il n'existe pas de contrainte intégrable).

En particulier, on appelle modèle cinématique du système le modèle :

$$\dot{q} = \sum_{i=1}^{n-k} X_i(q)v_i \quad (2.4)$$

avec $\{X_1, \dots, X_{n-k}\}$ une famille de champs de vecteurs telle que $\langle a_i(q), X_j(q) \rangle = 0, \forall i = 1, \dots, k, \forall j = 1, \dots, n - k$.

La variable $v = (v_1, \dots, v_{n-k}) = (v_1, \dots, v_m) \in \mathbb{R}^m$ est assimilable à une variable de commande en vitesse. On appelle m (la dimension de l'espace des vitesses instantanées possibles) le nombre de degrés de liberté du système.

Les contraintes décrites par (2.2) peuvent s'écrire sous la forme :

$$A(q)\dot{q} = 0 \quad (2.5)$$

où $A(q)$ est une matrice $m \times n$.

L'intégrabilité de ces contraintes est directement liée à l'involutivité de la distribution $D(q)$ définie par :

$$D(q) = \ker(A(q)) \quad (2.6)$$

On rappelle qu'une distribution $D(q)$ est involutive, si et seulement si, le crochet de Lie de 2 champs de vecteur appartenant à $D(q)$ est à son tour un champ appartenant à $D(q)$. Lorsque $D(q)$ n'est pas involutive, on appelle fermeture involutive de $D(q)$, notée $\bar{D}(q)$, la plus petite distribution involutive contenant $D(q)$. On a alors les résultats suivants (conséquences du théorème de Frobenius) :

- Si $D(q)$ est involutive, alors les m contraintes définies par (2.5) sont toutes intégrables, l'état q du système S peut être réduit à un vecteur de dimension $n - m$.
- Si $D(q)$ n'est pas involutive, et $\dim\{\bar{D}(q)\} = n - m_1$, avec $0 < m_1 < m$, alors m_1 contraintes sont intégrables, et l'état du système S peut être réduit à un vecteur de dimension $n - m_1$. Le système reste néanmoins assujéti à $m - m_1$ contraintes non holonomes.
- Si $D(q)$ n'est pas involutive, et $\dim\{\bar{D}(q)\} = n$, alors aucune des m contraintes n'est intégrable. L'état du système S ne peut être réduit, et aucune des contraintes cinématiques ne peut être éliminée.

2.2.2. La contrainte de roulement sans glissement

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

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

En pratique le contact se fait sur une surface, ce qui engendre bien évidemment de légers glissements, notamment lorsque les roues ne sont pas réellement rigides (ex. pneu). Dans le cadre de notre travail, nous supposons que ces glissements sont négligeables, et que la condition de r.s.g est vérifiée.

Pour exprimer analytiquement la condition de r.s.g, considérons une roue verticale sur le plan qui roule sans glisser sur un sol plan, comme illustré sur la Figure 2.1.

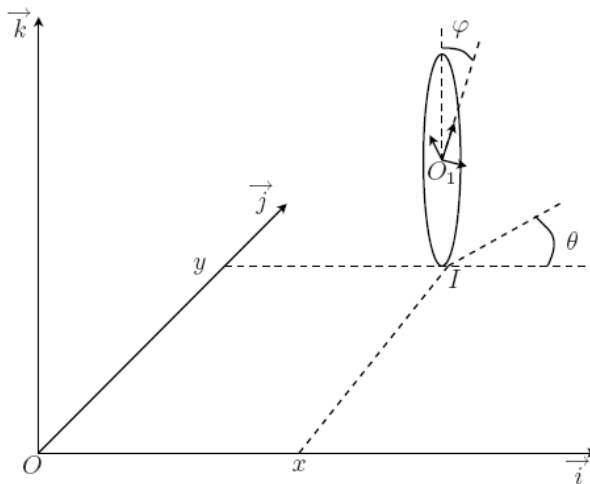


Figure 2.1 : Roue verticale qui roule sans glisser

L'hypothèse de roulement sans glissement se traduit par une vitesse nulle au point de contact I entre la roue et le sol.

avec les notations de la figure (2.1), on a :

$$\begin{aligned}\vec{V}(I/R_O) &= \dot{x}\vec{i} + \dot{y}\vec{j} + (\dot{\theta}\vec{k} + \dot{\phi}(-\sin\theta\vec{i} + \cos\theta\vec{j})) \wedge (-r\vec{k}) \\ &= (\dot{x} - r\dot{\phi}\cos\theta)\vec{i} + (\dot{y} - r\dot{\phi}\sin\theta)\vec{j} \\ &= 0\end{aligned}$$

où r est le rayon de la roue et (x,y) les coordonnées du point O_1 dans le repère fixe $R_O(O, \vec{i}, \vec{j}, \vec{k})$. On en déduit deux contraintes :

$$\begin{cases} \dot{x} - r\dot{\phi}\cos\theta = 0 \\ \dot{y} - r\dot{\phi}\sin\theta = 0 \end{cases} \quad (2.7)$$

Qu'on peut réécrire:

$$\begin{cases} \dot{x}\cos\theta + \dot{y}\sin\theta = r\dot{\phi} \\ -\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \end{cases} \quad (2.8)$$

Il est intéressant de noter qu'en introduisant $v = r\dot{\phi}$ la vitesse de roulement de la roue et $\omega = \dot{\theta}$ sa vitesse de rotation autour de l'axe $\vec{k}$, on forme le modèle suivant :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (2.9)$$

D'autre part, et pour bien expliciter la nature non holonome du roulement sans glissement de la roue sur le sol, dont la configuration est donnée par $q = (x, y, \theta, \phi)^T \in N = SE(2) \times \mathbb{S}^1$, les équations (2.7) peuvent être réécrites comme suit :

$$\begin{cases} \dot{x}\cos\theta + \dot{y}\sin\theta - r\dot{\phi} = 0 \\ -\dot{x}\sin\theta + \dot{y}\cos\theta = 0 \end{cases} \quad (2.10)$$

Ses dernières sont de la forme $\langle a_i(q), \dot{q} \rangle = 0$ avec les covecteurs suivants :

$$\begin{cases} a_1(q) = (\cos\theta \sin\theta \ 0 \ -r) \\ a_2(q) = (-\sin\theta \cos\theta \ 0 \ 0) \end{cases} \quad (2.11)$$

Il apparaît clairement que ces deux covecteurs sont indépendants. Nous obtenons un modèle cinématique du type (S_0) avec $m = 2$, et

$$X_1(q) = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \\ 1/r \end{pmatrix}, \quad X_2(q) = X_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad (2.12)$$

qui sont indépendants. Les covecteurs correspondent bien à des contraintes non holonomes. La non intégrabilité de ces contraintes peut se vérifier en montrant que la fermeture involutive de la distribution engendrée par X_1 et X_2 est de rang maximal, c'est-à-dire ici de rang $n = \dim N = 4$. Tel est le cas puisque la matrice

$$\left(X_1(q), X_2(q), [X_1, X_2](q), [X_2, [X_1, X_2]](q) \right) = \begin{pmatrix} \cos \theta & 0 & \sin \theta & \cos \theta \\ \sin \theta & 0 & -\cos \theta & \sin \theta \\ 0 & 1 & 0 & 0 \\ 1/r & 0 & 0 & 0 \end{pmatrix} \quad (2.13)$$

est inversible en tout point.

La réduction de la configuration de la roue à ses trois premières composantes conduit à un système dont les champs de commande :

$$X_1(q) = \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix}, \quad X_2(q) = X_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (2.14)$$

sont linéairement indépendants en tout point.

2.2.3. Modélisation d'un robot de type unicycle

2.2.3.1. Description

On désigne par unicycle un robot actionné par deux roues coaxiales (parallèles), non orientables, commandées indépendamment, et possédant éventuellement un certain nombre de roues folles assurant sa stabilité. Ce type de robot est très répandu en raison de sa simplicité de construction et de ces propriétés cinématiques intéressantes. Dans la figure ci-après, on présente quelques exemples de robots de ce type, qui sont les plus connus dans le domaine de la recherche scientifique.

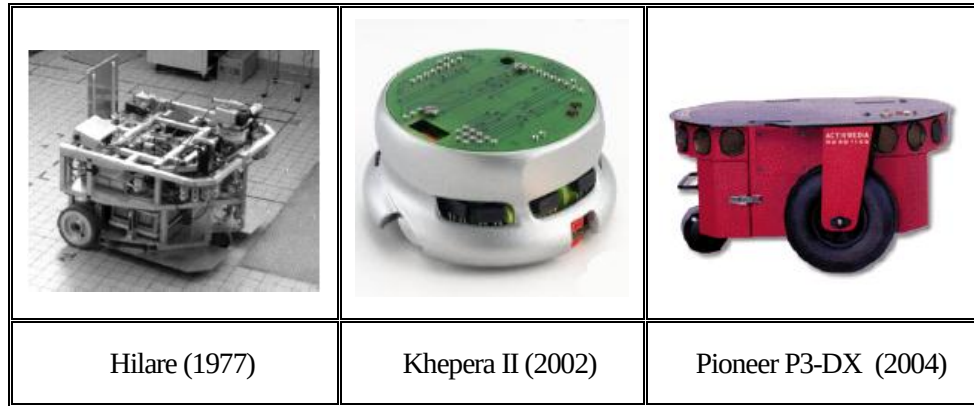


Figure 2.2 : Exemples de robots de type unicycle

2.2.3.2. Modèle cinématique

Une représentation simplifiée d'un robot unicycle est illustrée dans la figure 2.3. Les éventuelles roues folles qui permettent de stabiliser la plateforme ont été omises dans le schéma car elles n'affectent pas la cinématique du robot.

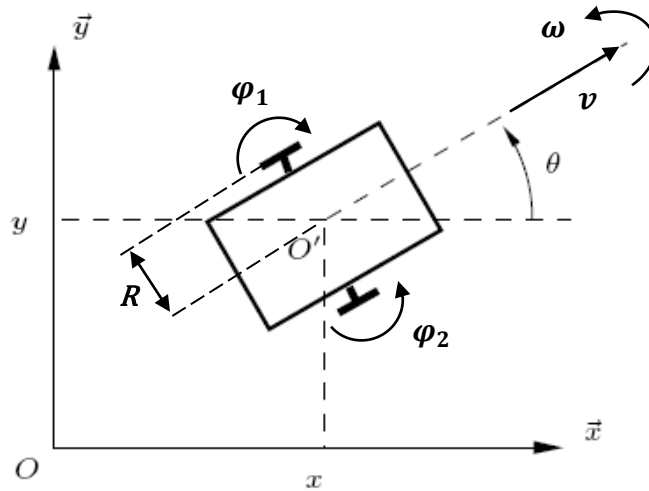


Figure 2.3 : Robot mobile de type unicycle

Soit le point $\hat{O}(x, y)$ situé au milieu de l'axe des deux roues motrices, θ l'orientation du robot, r le rayon des roues, $2R$ la distance entre les roues, φ_1 et φ_2 les positions angulaires des roues gauche et droite respectivement, et v et ω les vitesses linéaire et angulaire du robot. Le vecteur de configuration du robot est donné par $q = (x, y, \theta, \varphi_1, \varphi_2)^T \in \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{S}^2$.

Les contraintes de roulement sans glissement appliquées à chacune des deux roues donnent :

- Pour la roue gauche :

$$\dot{x} - R\dot{\theta}\cos\theta - r\dot{\phi}_1\cos\theta = 0 \quad (2.15)$$

$$\dot{y} - R\dot{\theta}\sin\theta - r\dot{\phi}_1\sin\theta = 0 \quad (2.16)$$

- Pour la roue droite :

$$\dot{x} + R\dot{\theta}\cos\theta - r\dot{\phi}_2\cos\theta = 0 \quad (2.17)$$

$$\dot{y} + R\dot{\theta}\sin\theta - r\dot{\phi}_2\sin\theta = 0 \quad (2.18)$$

En écrivant ces contraintes sous la forme (2.4), la matrice de contraintes $A(q)$ est alors donnée par :

$$A(q) = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -R\cos\theta & -R\sin\theta & R\cos\theta & R\sin\theta \\ -r\cos\theta & -r\sin\theta & 0 & 0 \\ 0 & 0 & -r\cos\theta & -r\sin\theta \end{pmatrix} \quad (2.19)$$

Cette matrice est seulement de rang 3 (les quatre contraintes ne sont pas indépendantes) puisque :

$$\sin\theta a_1(q) - \sin\theta a_3(q) - \cos\theta a_2(q) + \cos\theta a_4(q) = 0 \quad (2.20)$$

Il suffit donc de ne considérer que trois de ces contraintes, les trois premières par exemple. Le test d'intégrabilité par application du Théorème de Frobenius est réalisé avec

$$A(q) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ -R\cos\theta & -R\sin\theta & R\cos\theta \\ -r\cos\theta & -r\sin\theta & 0 \\ 0 & 0 & -r\cos\theta \end{pmatrix} \quad (2.21)$$

$$(X_1(q), X_2(q)) = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \\ 1/r & -R/r \\ 1/r & R/r \end{pmatrix} \quad (2.22)$$

On peut vérifier que $\dim(\text{Lie}(X_i)(q)) = 4 = n - 1$, ce qui implique l'existence d'une contrainte intégrable.

En effet, on a :

$$\begin{cases} \dot{x} \cos \theta + \dot{y} \sin \theta - R\dot{\theta} = r\dot{\varphi}_1 \\ \dot{x} \cos \theta + \dot{y} \sin \theta + R\dot{\theta} = r\dot{\varphi}_2 \end{cases} \quad (2.23)$$

on obtient donc :

$$2R\dot{\theta} = r(\dot{\varphi}_2 - \dot{\varphi}_1) \quad (2.24)$$

d'où :

$$2R\theta = r(\varphi_2 - \varphi_1) + \text{cte} \quad (2.25)$$

Cette dernière relation montre qu'il est par exemple possible d'exprimer φ_2 en fonction de φ_1 et θ . Dans la pratique, comme la valeur de l'angle φ_1 ne présente pas d'intérêt majeur, on ne retient que le vecteur $q = (x, y, \theta)^T \in Q = \mathbb{R}^2 \times \mathbb{S}^1$ pour caractériser la configuration du véhicule.

Par conséquent, on peut décrire la cinématique de l'unicycle par le modèle cinématique donné par :

$$\dot{q} = \bar{X}(q)u, \quad \text{avec } \bar{X}(q) = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix}, \quad u = (v, \omega)^T \quad (2.26)$$

où : $v = \frac{r}{2}(\dot{\varphi}_1 + \dot{\varphi}_2)$, et $\omega = \frac{r}{2R}(\dot{\varphi}_2 - \dot{\varphi}_1)$.

Modèle qu'on peut récrire sous forme $\dot{q} = f(q, u)$:

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

avec : $q = (x, y, \theta)^T$ et $u = (v, \omega)^T$.

La similarité de ce modèle avec le modèle (2.9) de la roue verticale, justifie l'appellation de robot de type unicycle.

2.2.4. Modélisation d'un robot de type tricycle

2.2.4.1. Description

Un robot de type tricycle est une plateforme mobile constituée de deux roues coaxiales fixes et d'une roue centrée orientable placée sur l'axe longitudinal de la plateforme. Son

mouvement est assuré par la combinaison d'une commande en vitesse longitudinale et une orientation de la roue directrice. Dans la figure 2.4 ci-après, on présente quelques exemples de robots de type tricycle.

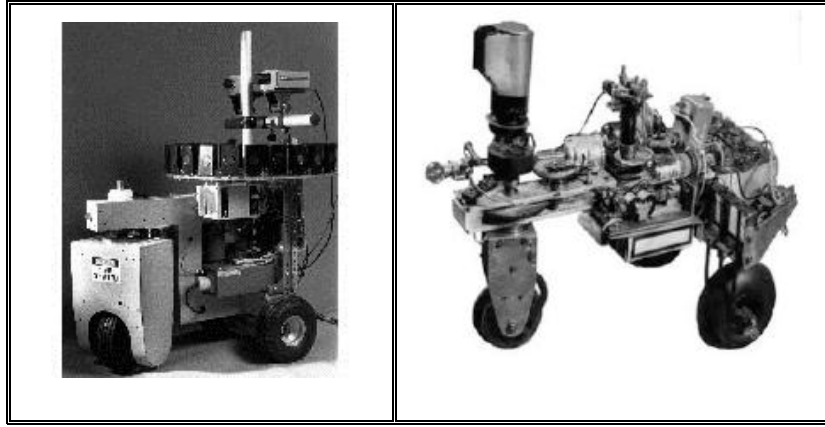


Figure 2.4 : Exemples de robots de type tricycle

2.2.4.2. Modèle cinématique

Considérons le schéma de la figure 2.5 décrivant la pose d'un tricycle dans un repère de référence.

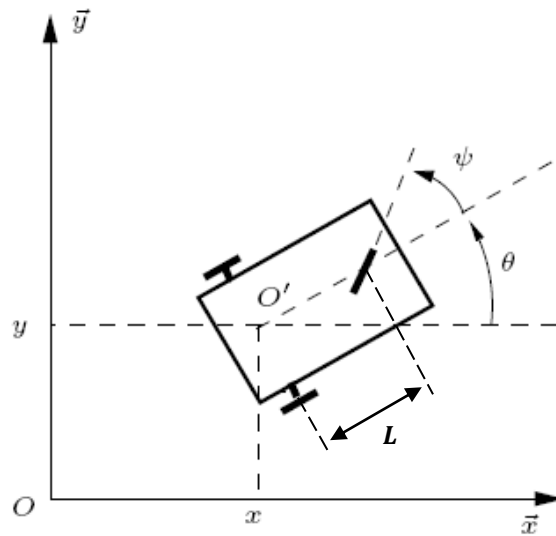


Figure 2.5 : Robot mobile de type tricycle

Le modèle cinématique correspondant à un robot tricycle est donné par :

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = v \frac{\tan \psi}{L} \\ \dot{\psi} = \omega_{\psi} \end{cases} \quad (2.28)$$

Qu'on peut récrire sous la forme :

$$\dot{q} = \bar{X}(q)u \quad \text{avec } q = (x, y, \theta, \psi)^T, \quad \bar{X}(q) = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ \frac{\tan \psi}{L} & 0 \\ 0 & 1 \end{pmatrix}, \quad u = (v, \omega_\psi)^T \quad (2.29)$$

où L est la distance entre les axes des roues avant et arrières, ψ est l'angle de braquage formé par la roue avant et l'axe principal de la plateforme, v est la vitesse linéaire et ω_ψ est la vitesse angulaire selon l'axe vertical de la roue directrice par rapport au corps du robot.

2.2.5. Modélisation d'un robot de type voiture

2.2.5.1. Description

Un robot de type voiture possède deux roues coaxiales au niveau de ces deux trains avant et arrière. Généralement les roues du train avant sont des roues directrices et les roues du train arrière motrices, mais plusieurs autres configurations peuvent être envisagées. Ce type de plateformes est le plus répandu dans les applications industrielles.



Figure 2.6 : Exemples de robots de type voiture

2.2.5.2. Modèle cinématique

Généralement, dans l'étude de la cinématique des plateformes de type voiture on introduit la notion de roue directrice centrale, qui est une roue virtuelle correspondant à la roue directrice d'un tricycle équivalent. Cette astuce permet de simplifier les équations en faisant abstraction du mécanisme de couplage des roues directrices servant à respecter les contraintes de roulement sans glissement et en ne considérant qu'un seul angle de direction.

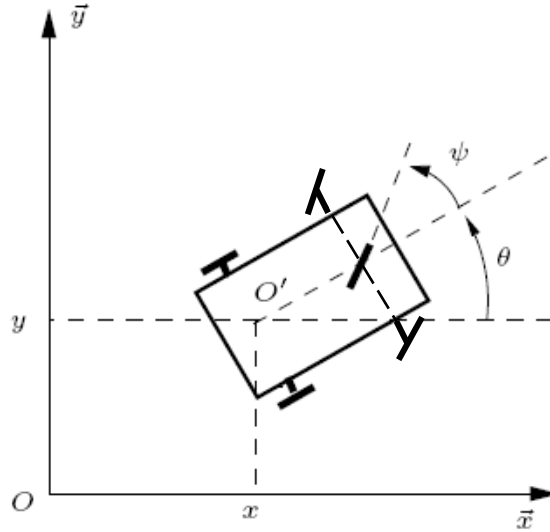


Figure 2.7 : Robot mobile de type voiture

Dans ce cas, le modèle (2.29) caractérisant la cinématique du tricycle pourra être adopté pour un robot de type voiture.

Pour plus de détails sur la modélisation des plateformes mobiles, consulter [1].

2.2.6. Modèle cinématique généralisé

Les modèles cinématiques associés aux plateformes mobiles étudiées (unicycle, tricycle et voiture) s'écrivent tous sous la forme :

$$\dot{q} = G(q)u \quad (2.30)$$

C'est une classe de systèmes dits sans dérives qui possèdent des propriétés particulières vis-à-vis de la commande.

2.3. Etat de l'art sur la commande des robots mobiles à roues

Dans cette partie, on va essayer de présenter un état de l'art des techniques proposées pour traiter la problématique de la commande des robots mobiles à roues non holonomes.

Pour aborder cette problématique dans le sens des travaux qui ont été réalisés dans la littérature, ces derniers peuvent être classifiés selon le type de stabilisation qu'on désire obtenir (complète ou partielle), mais plus souvent, la classification est opérée sur le type de trajectoire à exécuter, soit : la stabilisation de points fixes, la poursuite de trajectoire, et le suivi de chemin.

2.3.1. Stabilisation de points fixes

Le problème de stabilisation de points fixes consiste à amener un robot mobile sur une position cible avec une orientation désirée, ce qui implique, par exemple pour un robot de type unicycle, de devoir commander trois variables (x, y, θ) , représentant la position et l'orientation du robot, avec seulement deux entrées (v, w) , qui représentent les vitesses linéaire et angulaire du robot.

La résolution de ce problème s'avère difficile, bien plus que la poursuite et le suivi de trajectoire. Deux principaux points en sont la cause :

- le système obtenu en linéarisant les équations représentant le modèle cinématique d'un robot mobile autour de tout point d'équilibre (configuration fixe) n'est pas commandable, car le nombre de sorties est supérieur à celui des entrées, ce qui empêche l'utilisation des techniques de commande linéaire pour résoudre ce problème ;
- il n'existe pas de retours d'état continus autonomes permettant la stabilisation asymptotique de configurations fixes (résultat avancé et prouvé par Brockett dans [9]), ce qui exclut l'utilisation des techniques de commande non linéaire classiques.

Pour contourner ces difficultés, plusieurs solutions ont été proposées, qu'on peut classer sous quatre catégories :

2.3.1.1. Les retours d'état discontinus

Cette approche proposée initialement par Bloch à la fin des années 80 [7], [8], qu'on trouve aussi dans les travaux de Canudas de Wit et Sordalen [12], [2] et [63], repose sur l'utilisation d'une commande discontinue (ou continue par morceaux), afin d'amener le robot à la configuration cible par le biais de configurations intermédiaires. Bien que cette technique puisse garantir la convergence vers le point d'équilibre, la stabilité de ce point, et donc la stabilité asymptotique ne peut être assurée puisque la discontinuité intervient précisément au point que l'on cherche à stabiliser.

Les points faibles de cette approche résident dans le fait qu'elle ne donne pas des résultats satisfaisants en termes de stabilité et de robustesse aux erreurs de modélisation, aussi pour rallier deux configurations infiniment proches, le robot peut avoir une trajectoire ample à décrire [46].

2.3.1.2. Les retours d'état non stationnaires

Cette approche a été initialement proposée par Samson, C. [65] dans le cadre de la stabilisation d'une configuration fixe d'un robot mobile de type unicycle. Contrairement à la première, celle-ci permet de préserver la continuité de la commande, et repose en outre sur des retours d'état dépendant explicitement du temps. Dans ce cadre, plusieurs travaux ont été proposés, notamment dans [62], [18].

L'inconvénient des commandes de ce type est qu'elles ne permettent pas d'obtenir une convergence exponentielle [56], mais seulement polynomiale (lente). Aussi, elles ne garantissent pas à priori la robustesse de la propriété de stabilité par rapport à des erreurs provenant de dynamiques non modélisées. Néanmoins, cette propriété de robustesse a pu être montrée dans quelques cas particuliers [48].

2.3.1.3. Les retours d'état hybrides

Les retours d'état "hybrides" impliquent l'injection de commandes en boucle ouverte itérées périodiquement. Ils sont dits hybrides car ils sont continus par rapport au temps, et discontinus par rapport à l'état. Ce type de commande permet l'obtention de la convergence exponentielle et la robustesse vis-à-vis des erreurs de modélisation [6], [57]. Cependant, cette propriété de convergence peut être perdue en présence de retards dans l'application de la commande ou de fluctuations de la période d'itération.

2.3.1.4. La commande par fonctions transverses

Du fait des problèmes de stabilité rencontrés dans les approches précédentes, les chercheurs ont pensé à alléger l'objectif de stabilisation vers une notion de stabilité pratique, dans laquelle l'objectif n'est plus de stabiliser le robot sur une configuration cible avec une précision infinie, mais d'amener son état dans un voisinage de cette configuration [46]. Pour adresser cette tâche, des techniques de commande par fonctions transverses (voir [58] et [67], [55], [54]) ont été introduites, avec pour objectif d'éliminer, ou tout au moins, de maîtriser le comportement oscillant du robot.

2.3.2. Poursuite de trajectoires

Contrairement à la stabilisation de configurations fixes, pour laquelle l'approximation linéaire du modèle du robot autour des points d'équilibre n'est pas commandable, les systèmes linéaires associés à la poursuite de trajectoires admissibles non stationnaires peuvent être commandables, ce qui permet d'envisager la stabilisation locale du système d'erreur via des retours d'état linéaires ou non linéaires.

En plus de ces approches locales, plusieurs techniques de commande non linéaire ont été proposées pour résoudre la problématique de poursuite de trajectoire de façon globale. Aussi, de même que pour la stabilisation de points fixes, les retours discontinus et /ou non stationnaires ont été proposés pour la poursuite de trajectoire.

Les approches proposées pour résoudre cette problématique peuvent être classées en deux catégories :

2.3.2.1. Approche locale

Cette approche consiste à synthétiser des retours d'état linéaires (à gains variables) stabilisant le système d'erreur linéarisé [66]. Les matrices d'état et de commande de ces

systèmes linéaires étant fonction du temps, les analyses de stabilité sont en général difficiles et souvent basées sur certaines hypothèses sur les trajectoires de référence. L'utilisation de techniques non linéaires peut permettre de relâcher les hypothèses en question ou de garantir un plus grand domaine de stabilité.

2.3.2.2. Approche globale

Dans laquelle on peut distinguer deux types :

2.3.2.2.1. Approche directe

Dans cette approche le travail s'effectue directement sur le système d'erreur initial, i.e. non linéaire, et repose sur l'exploitation des fonctions de Lyapunov [36], [37], et [74]). La principale difficulté de cette approche consiste à trouver une fonction de Lyapunov satisfaisante. Pour faciliter la résolution de ce problème, quelques travaux utilisent une approche backstepping comme dans [35] ou [21], consistant à contrôler successivement plusieurs tâches associées à la poursuite de trajectoire (et par voie de conséquence de décomposer la construction de la fonction de Lyapunov). Par exemple la commande peut être décomposée en deux tâches, assurant la convergence de l'écart angulaire, puis celle de l'écart latéral [46].

2.3.2.2.2. Approche par linéarisation exacte entrée/sortie

Cette approche, aussi désignée linéarisation par retour d'état dynamique, tire profit des propriétés de platitude des systèmes non holonomes [22], [24], [68].

Toutes ces approches ne souffrent pas des limitations propres aux méthodes de stabilisation de points fixes : elles permettent d'obtenir des convergences de type exponentielles et une marge de robustesse par rapport aux erreurs de modèle dès lors que la dynamique de la trajectoire de référence est suffisante, on parle aussi de conditions « d'excitation persistante ». [51].

2.3.3. Suivi de chemins

Le problème du suivi de chemin consiste à suivre un repère en déplacement sur une courbe avec une vitesse d'avancement donnée [13]. Il entre dans la catégorie des techniques de stabilisation partielle de la configuration du robot (illustrée par exemple dans [66]), où il est question de stabiliser une partie seulement de la situation en un point fixe (la position seule ou bien une composante de la position et l'orientation). Il s'agit d'une problématique relativement simple par rapport aux deux premières.

Les modèles dédiés au suivi de trajectoire peuvent tous être convertis sous une forme dite chaînée (voir [59], [47]), et par un changement d'échelle des temps, il devient possible d'obtenir une expression linéaire pour la partie du robot décrivant l'écart à la trajectoire [64].

Pour terminer cette section, on présente, dans le schéma suivant (figure 2.8), un récapitulatif de l'ensemble des techniques de commande mentionnées:

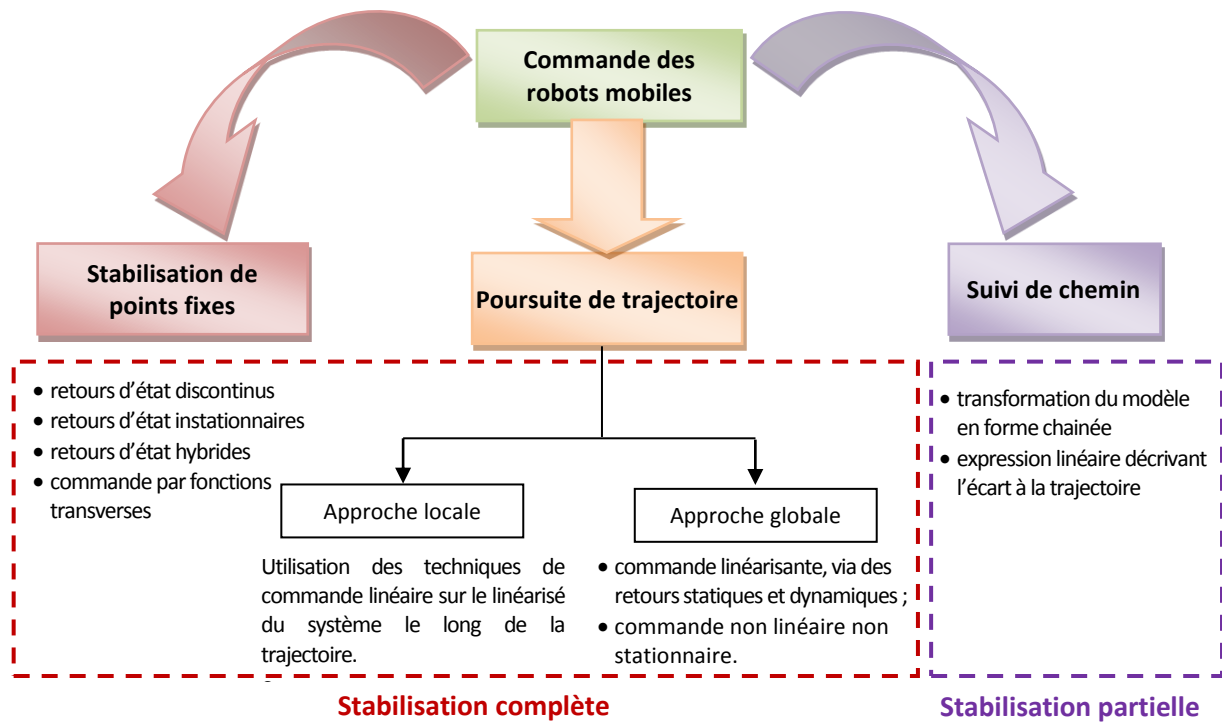


Figure 2.8 : Classification des techniques de commande des robots mobiles à roues

2.4. Conclusion

Dans ce chapitre nous avons mis en évidence la nature non holonome des robots mobiles à roues. Cette propriété possède des conséquences non seulement sur la cinématique du robot, qui se voit interdit certaines manœuvres au sein de son espace de configuration, mais aussi sur sa commande qui a reçu un intérêt particulier de la part des chercheurs, comme le démontre l'état de l'art présenté. Il s'avère que la stabilisation de configurations fixes est le problème le plus difficile à traiter, où des techniques de commande complexes de type discontinu ou non stationnaires (variant dans le temps) ont été proposées pour le résoudre. Pour ce qui est de la poursuite de trajectoire et du suivi de chemin, l'utilisation des techniques de commande linéaire est rendue possible par le fait que le linéarisé du modèle du robot le long de trajectoire non stationnaire soit commandable. Toutefois, des techniques non linéaires ont été proposées pour résoudre ces problématiques de façon globale.

Chapitre 3

Poursuite et suivi de trajectoires

3.1. Introduction

Après avoir exposé dans le chapitre précédent un état de l'art des stratégies de commande proposées pour stabiliser les robots mobiles non holonomes, nous allons maintenant nous intéresser aux techniques de poursuite de trajectoire et de suivi de chemin, appliquées à la commande d'un robot mobile de type unicycle, où nous commençons d'abord par étudier les propriétés du modèle cinématique y associé.

Dans le cadre de la poursuite de trajectoire, nous proposons trois méthodes. La première méthode est la linéarisation par bouclage dynamique, qui permet d'obtenir un système équivalent complètement linéaire et découplé. Celle-ci est rendue possible grâce à la propriété de platitude dont jouit le robot unicycle (y compris les autres robots mobiles non holonomes). Nous présentons dans ce chapitre quelques notions liées à cette propriété et la méthode pour aboutir à la linéarisation exacte, tout juste avant de l'implémenter sur le modèle de l'unicycle.

La deuxième méthode est la commande par retour d'état non linéaire non stationnaire, dans laquelle on définit dans le repère mobile lié au robot, l'erreur de poursuite par rapport à un robot virtuel du même type exécutant la trajectoire de référence. Dans ce cadre, on présente deux lois de commande assez connues dans la littérature, la première est basée sur le critère de stabilité de Lyapunov, et la deuxième exploite la propriété de stabilité des systèmes en cascades.

La troisième technique présentée suppose l'orientation du robot non mesurée. Elle est basée sur la loi de commande issue de la méthode n°2, à laquelle on intègre un observateur de l'orientation du robot.

Pour ce qui est du suivi de chemin, nous proposons une technique de commande non linéaire semblable à celle développée dans le cadre de la poursuite, où le robot, animé d'une vitesse linéaire donnée, suit un repère mobile situé sur le chemin.

Enfin, pour valider l'ensemble des techniques appliquées au modèle de l'unicycle, nous présenterons les résultats de simulation pour différentes trajectoires de références, et en présence de perturbations sur le système.

3.2. Propriétés du modèle cinématique du robot unicycle

3.2.1. Commandabilité

Définition 3.1

Soit un système de commande

$$\dot{q} = \sum_{i=1}^m X_i(q)v_i, \quad m < n \quad (3.1)$$

sur une variété Q de dimension n . On considère comme commandes admissibles l'ensemble V des fonctions $v = (v_1, \dots, v_m)$ définies et continues par morceaux sur un intervalle $[0, T_v]$ de $\mathbb{R}_+$ et telles que $\|v(t)\| \leq 1 \forall t \in [0, T_v]$. On note $q_v(t, t_0, q_0)$ la valeur en t de la solution du système pour la commande v et la condition initiale q_0 en $t = t_0$.

Soit q_0 un point d'équilibre du système associé à la commande $v = 0$. On dit que le système est commandable en q_0 si l'ensemble :

$$\{q_v(t, 0, q_0) : v \in V, t \in [0, T_v]\}$$

contient un voisinage de q_0 . Si ce voisinage est Q tout entier, le système est dit globalement commandable.

Le résultat suivant fournit une condition suffisante et nécessaire pour qu'un système du type (3.1) soit commandable.

Théorème 3.1 (Chow) [16]

Soit le système (3.1) sur une variété Q de dimension n . Alors si

$$\dim \text{Lie}(X_i)(q_0) = n \quad (3.2)$$

Le système est commandable en q_0 . Si (3.2) est vérifiée pour tout q_0 et Q est une variété connexe, alors le système est globalement commandable.

La condition (3.2) est connue comme condition de rang de l'algèbre de Lie (CRAL). Appliquons ce résultat sur un robot de type unicycle.

On a le modèle (2.26) associé à l'unicycle, qu'on peut récrire :

$$\dot{q} = X_1(q)v + X_2(q)\omega \quad (3.3)$$

avec $q = [x \ y \ \theta]^T$, $X_1(q) = [\cos \theta \ \sin \theta \ 0]^T$ et $X_2(q) = [0 \ 0 \ 1]^T$

Par simple calcul du crochet de Lie des champs de vecteurs X_1 et X_2 il advient que la matrice :

$$[X_1(q), X_2(q), [X_1(q), X_2(q)]] = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ \sin \theta & 0 & \cos \theta \\ 0 & 1 & 0 \end{pmatrix} \quad (3.4)$$

est inversible $\forall q$, donc $\dim \text{Lie}(X_1, X_2)(q) = 3 = n, \forall q$.

d'où le système est globalement commandable.

3.2.2. Commandabilité du linéarisé

3.2.2.1. Autour d'un point d'équilibre

En dépit du fait que le modèle de l'unicycle soit globalement commandable, son linéarisé autour de tout point d'équilibre n'est pas commandable.

En effet, l'approximation linéaire du système (2.27) autour d'un point d'équilibre $q_0 = (x_0, y_0, \theta_0)^T$ est donnée par :

$$\dot{q} = Aq + Bu \quad (3.5)$$

avec :

$$A = \frac{\partial f}{\partial q}(q_0, 0) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \frac{\partial f}{\partial u}(q_0, 0) = \begin{pmatrix} \cos \theta_0 & 0 \\ \sin \theta_0 & 0 \\ 0 & 1 \end{pmatrix}$$

Si par exemple on considère le critère de commandabilité selon Kalman, pour que le système (3.5) soit localement commandable, il faut que $\text{rang} [B \ AB \ A^2B] = n = 3$, où n la dimension du vecteur d'état du système.

Dans notre cas, $\text{rang}[B \ AB \ A^2B] = \text{rang}[B] = 2 < n$.

3.2.2.2. Le long de trajectoire non stationnaire

En procédant de la même manière que précédemment, l'approximation linéaire du système (2.27) autour d'une trajectoire se caractérise par :

$$A = \begin{pmatrix} 0 & 0 & -v \sin \theta \\ 0 & 0 & v \cos \theta \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix}$$

$$[B \ AB \ A^2B] = \begin{pmatrix} \cos \theta & 0 & -v \sin \theta \\ \sin \theta & 0 & v \cos \theta \\ 0 & 1 & 0 \end{pmatrix} \quad (3.6)$$

Cette matrice n'est de rang plein que sous la condition $v \neq 0$. Donc, le linéarisé du système le long de trajectoire est commandable à condition que la trajectoire soit non stationnaire.

3.2.3. Stabilisabilité

Un aspect essentiel du problème de stabilisation des systèmes non holonomes est lié à l'impossibilité d'obtenir une stabilisation asymptotique par le biais de retours d'état continus et stationnaires. Ceci découle d'un résultat important énoncé par Brockett [9], connu sous le nom de condition de Brockett.

Théorème 3.2 [9] :

Considérons le système non linéaire $\dot{q} = f(q, u)$, où f est différentiable et $f(0, 0) = 0$. Une condition nécessaire pour qu'il existe un retour d'état $u(q)$ continu qui rende l'origine du système bouclé localement asymptotiquement stable est que l'image par f de tout voisinage de l'origine de $\mathbb{R}^{n \times m}$ soit un voisinage de 0 dans $\mathbb{R}^n$.

Cette condition implique que de nombreux systèmes non linéaires commandables ne sont pas asymptotiquement stabilisables par retour d'état continu, dont les robots mobiles non holonomes.

Néanmoins, un autre résultat [17] a permis de démontrer pour ces systèmes l'existence de retours d'état instationnaires $v(q, t)$ (i.e. dépendant explicitement du temps), qui stabilisent asymptotiquement une configuration fixe donnée. Ce résultat est valable pour tout système sans dérive commandable.

3.3. Poursuite de trajectoire :**3.3.1. Trajectoire admissible :**

On appelle trajectoire admissible (ou réalisable) d'un système de commande $\dot{q} = f(q, u)$ sur une variété U , toute fonction $q : \mathbb{R} \rightarrow U$ qui vérifie l'équation du système pour une commande donnée u_r de classe C^0 , i.e. $\dot{q}_r = f(q_r, u_r)$.

On vérifie aisément que les points fixes sont des trajectoires admissibles pour un robot de type unicycle modélisé par (2.27) pour lequel $f(q, 0) = 0, \forall q$. C'est d'ailleurs le cas pour tous les systèmes de la forme (2.30), appelés systèmes sans dérive.

3.3.2. Formulation du problème de poursuite de trajectoire

Considérons le système non linéaire suivant :

$$\dot{q} = f(q, u) \quad (3.7)$$

où $q \in Q \subset \mathbb{R}^n$ représente l'état du système, avec Q un ensemble ouvert de $\mathbb{R}^n$, $u \in \mathbb{R}^m$ est l'entrée de commande et $f : Q \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ est une application suffisamment différentiable.

Etant donnée une trajectoire de référence admissible $q_r(t)$, l'objectif de la poursuite de trajectoire consiste à trouver la commande u du système (3.7) telle que l'erreur de poursuite $q_e = q_r - q$ converge asymptotiquement vers zéro. Dans le cas d'un robot de type unicycle, le principe de la poursuite de trajectoire est illustré sur la figure 3.1.

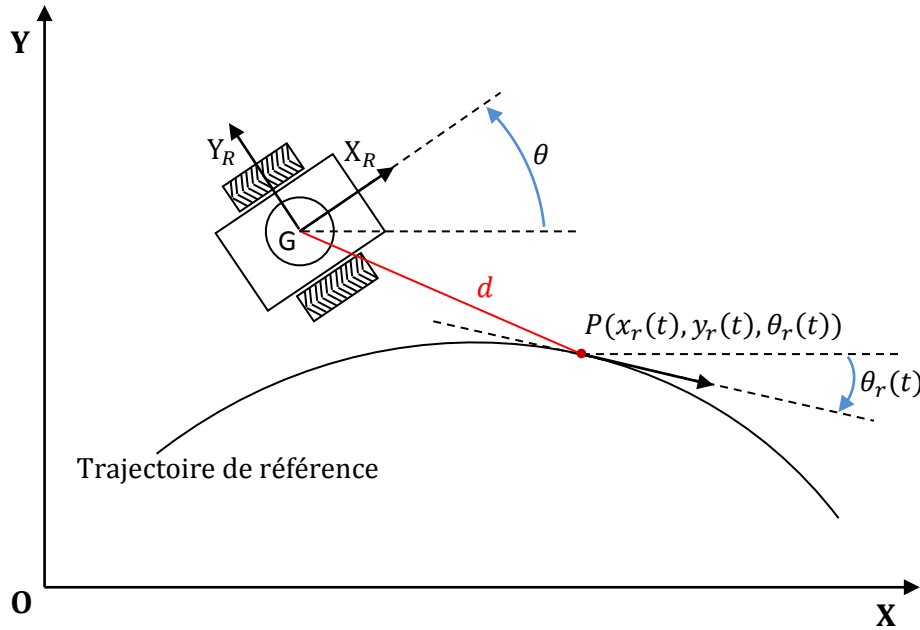


Figure 3.1 : Schématisation du problème de poursuite de trajectoire

Dans ce cas, il s'agit de faire asservir la position d'un point appartenant au robot sur un point P de la trajectoire, à tout instant. Généralement pour le choix du point appartenant au robot on prend le centre de gravité, ici représenté par G. L'objectif de la poursuite est alors de minimiser la distance séparant les deux points, et qui représente l'écart à la trajectoire, notée d , ainsi que l'écart d'orientation $|\theta_r(t) - \theta(t)|$.

Nous allons maintenant aborder la partie commande de l'unicycle, dans laquelle on va présenter trois types de commande qui entrent toutes dans le cadre des approches globales, c'est-à-dire qu'il s'agit de commandes non linéaires qui se basent sur le modèle initial du robot (non linéaire), contrairement à d'autres techniques qui reposent sur la commande du linéarisé le long de trajectoires non stationnaires (approches locales). Par ailleurs, notre intérêt a porté sur la technique de linéarisation par bouclage dynamique, pour laquelle nous présentons un bref état de l'art, ainsi que les développements nécessaires à sa mise en œuvre.

3.3.3. Commande linéarisante par bouclage dynamique

Dans cette section, on se propose de synthétiser une commande linéarisante au sens des entrées/sorties par bouclage dynamique appliquée à un robot de type unicycle, où on va démontrer qu'il s'agit d'un système non linéarisable au sens des entrées/état par bouclage statique. En contrepartie, bénéficiant de la propriété de platitude, ce système est complètement linéarisable par bouclage dynamique.

La linéarisation par bouclage est aujourd'hui bien connue en théorie des systèmes non linéaires. Le lecteur peut se référer, par exemple, aux ouvrages [31], [70].

Grâce à cette technique, un grand nombre de systèmes non linéaires peuvent être transformés en systèmes possédant un comportement entrée/état ou entrée/sortie linéaire à travers un choix approprié d'une loi de commande par retour d'état non linéaire [5].

A la base, la technique utilisée dans cette méthode avait pour objectif d'obtenir un découplage entrée/sortie par retour d'état, et a donc été d'abord introduite pour la commande des systèmes linéaires, grâce notamment aux travaux de Falb P.L et Wolovitch W.A. [19] qui donnèrent des conditions nécessaires et suffisantes pour son applicabilité. L'inconvénient de ce découplage est, dans certains cas, la perte d'observabilité d'une partie de l'état consécutive à des simplifications pôle-zéro dans le transfert en boucle fermée. L'étude de la dynamique des zéros finis de la matrice de transfert est alors nécessaire pour avoir la stabilité interne [44]. Afin d'y remédier, Wonham, W.M. [72] proposa une méthode pour le découplage linéaire entrée/sortie avec stabilité qui consistait à remplacer le bouclage statique par un bouclage dynamique.

Pour les systèmes non-linéaires, Isidori, A. *et al* [30] définirent les conditions équivalentes qui portaient sur la régularité de la matrice de découplage. Cette technique utilise un changement de coordonnées sur l'état et un retour d'état statique permettant à la fois de découpler et de linéariser les relations entrée/sortie. S'il apparaît un état non observable, son étude se fait à l'aide de la notion de dynamique des zéros. On utilisera alors les critères de stabilité sur la dynamique des zéros comme la stabilité exponentielle [10] ou la K-stabilité [15] pour évaluer la stabilité interne du système non linéaire bouclé.

De même que pour les systèmes linéaires, le bouclage dynamique fut aussi utilisé dans le cadre du découplage de systèmes non linéaires, où Nijmeijer, H. et Respondeck, W. [60] définirent les systèmes non linéaires découplables par ce type de bouclage.

3.3.3.1. Notion de platitude

Le concept de platitude d'un système non linéaire fut proposé par Fliess, M. *et al* [23]. C'est un concept qui introduit une notion d'équivalence entre un système non linéaire et un système linéaire commandable. Cette équivalence porte le nom d'équivalence par bouclage dynamique endogène dans le cadre de l'algèbre différentielle et le nom d'équivalence de Lie-Bäcklund dans le cadre de la géométrie différentielle.

Cette notion est plus générale que l'équivalence par difféomorphisme et bouclage statique car elle ne conserve pas nécessairement la dimension de l'état, et les sorties du système linéaire commandable équivalent mis sous la forme de Brunovsky constituent les sorties plates du système. Cependant, il n'existe pas actuellement de méthode constructive permettant de déterminer la platitude d'un système et d'en donner les sorties plates. La détermination des sorties candidates reste liée à une certaine pratique de la méthode, comme c'est le cas avec les fonctions de Lyapounov pour l'étude de la

stabilité. Toutefois, la connaissance de certains critères de platitude peut aider au choix des sorties plates [14].

3.3.3.1.1. Définition de la platitude pour un système non linéaire

Dans le contexte de l'algèbre différentielle, un système est vu comme un champ de vecteurs généré par un ensemble de variables (états et commandes). La caractéristique essentielle des systèmes plats est qu'il existe un vecteur de sorties plates, tel que ses composantes soient des fonctions différentielles des variables du système, et tel que toute variable du système (on ne fait pas de distinction entre les variables d'état et les entrées du processus) puisse s'exprimer à partir de ces sorties et d'un nombre fini de leurs dérivées sans intégration d'équations différentielles [73].

Pour définir les systèmes plats, considérons le système non linéaire régi par l'équation différentielle suivante :

$$\dot{x} = f(x, u) \quad (3.8)$$

où $x \in \mathbb{R}^n$ est l'état, $u \in \mathbb{R}^m$ est l'entrée de commande et f une fonction régulière de classe C^∞ de x et de u dont le rang de la matrice jacobienne $\frac{\partial f}{\partial u}$ est égal à m (i.e. le système admet effectivement m commandes indépendantes).

Définition 3.2

Un système non linéaire modélisé par (3.8) est dit différentiellement plat, si et seulement si, il existe un vecteur de sorties plates y de dimension égale à celle du vecteur de commande u ($y \in \mathbb{R}^m$), tel que :

- *L'état x et la commande u s'expriment en fonction de y et d'un nombre fini de ses dérivées;*
- *La sortie plate y s'exprime en fonction de l'état x , de la commande u , et d'un nombre fini r des dérivées de u .*

Ces conditions sont traduites par :

$$x = X(y, \dot{y}, \dots, y^{p-1}) \quad (3.9)$$

$$u = U(y, \dot{y}, \dots, y^{p-1}, y^p) \quad (3.10)$$

$$y = Y(x, u, \dot{u}, \dots, u^r) \quad (3.11)$$

où $X: (\mathbb{R}^m)^p \rightarrow \mathbb{R}^n$, $U: (\mathbb{R}^m)^{p+1} \rightarrow \mathbb{R}^m$, $Y: \mathbb{R}^n \times (\mathbb{R}^m)^{r+1} \rightarrow \mathbb{R}^m$ sont des fonctions régulières.

Cela revient à dire que tout le comportement dynamique du système (3.8) peut être décrit par le comportement dynamique de la sortie plate y . A $y(t)$ définie pour $t \in [0, T]$; les trajectoires sont de la forme :

$$\begin{aligned} x(t) &= X(y(t), \dot{y}(t), \dots, y^{p-1}(t)) \\ u(t) &= U(y(t), \dot{y}(t), \dots, y^{p-1}(t), y^p(t)) \end{aligned} \quad (3.12)$$

où p est un entier.

Remarques

- avec cette définition, les composantes du vecteur de sortie y sont différentiellement indépendantes, car il n'existe pas de relation différentielle entre les composantes $(y_1, \dots, y_m)$ qui soit indépendante de u ;
- Un système linéaire commandable est plat, et admet pour sorties plates les sorties de Brunovsky issues de la forme canonique de commandabilité.

Par la suite, la terminologie « systèmes plats », sera utilisée pour désigner la classe des modèles non linéaires respectant la définition (3.2).

3.3.3.2. Platitude et linéarisation

Les systèmes non linéaires plats ont la propriété d'être équivalents à des systèmes linéaires par bouclage dynamique endogène [23]. Avant d'introduire ce type de bouclage, rappelons d'abord la linéarisation par difféomorphisme et bouclage statique.

3.3.3.2.1. Linéarisation par difféomorphisme et bouclage statique

Pour le système dynamique donné par (3.8), le problème de linéarisation par difféomorphisme et bouclage statique consiste à trouver un changement de coordonnées (un difféomorphisme φ de classe C^∞) donné par :

$$z = \varphi(x), \quad \varphi(0) = 0 \quad (3.13)$$

et un bouclage statique d'état de la forme :

$$u = \alpha(x) + \beta(x)v \quad (3.14)$$

avec $\alpha(0) = 0$, et $\beta(x)$ inversible. α et β sont de classe C^∞ . Après bouclage et changement de coordonnées, on obtient un système linéaire commandable [28] de la forme :

$$\dot{z} = Az + Bv \quad (3.15)$$

où $z \in \mathbb{R}^n$ est l'état associé à la nouvelle entrée de commande v du système linéaire équivalent donné par (3.8) et représenté sur la figure 3.2.

Ce bouclage consiste, en effet, à passer de l'entrée v à l'entrée u et réciproquement, sans intégrer d'équations différentielles, grâce aux expressions suivantes :

$$\begin{aligned} u &= \alpha(x) + \beta(x)v \\ v &= \beta(x)^{-1}\alpha(x) + \beta(x)^{-1}u \end{aligned} \quad (3.16)$$

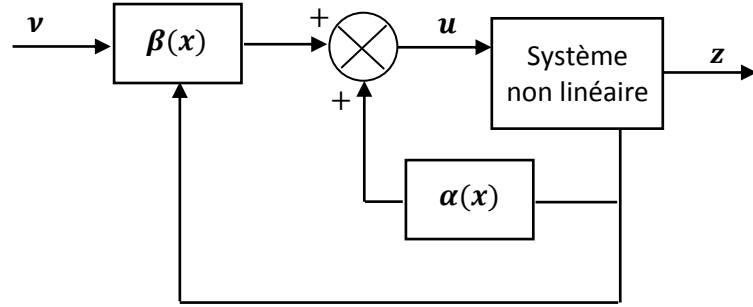


Figure 3.2 : Système non linéaire bouclé équivalent à un système linéaire

Des conditions nécessaires et suffisantes, d'existence d'un bouclage statique linéarisant, en termes de crochets de Lie pour un système affine en la commande ont été proposées dans [33] et [29].

Pour un système non linéaire ayant la représentation d'état suivante :

$$\dot{x} = f(x) + \sum_{i=1}^m g_i(x)u_i, \text{ avec } x \in \mathbb{R}^n \quad (3.17)$$

où $f, g_1, g_2, \dots, g_m$ sont des champs de vecteurs de classe C^∞ vérifiant :

$$f(0) = 0 \text{ et } \text{rang}\{g_1(0), g_2(0), \dots, g_m(0)\} = m \quad (3.18)$$

L'expression (3.16) sera donnée par :

$$\begin{aligned} \alpha(x) &= (L_g L_f^{(n-1)} z)^{-1} (L_f^{(n)} z) \\ \beta(x) &= (L_g L_f^{(n-1)} z)^{-1} \end{aligned} \quad (3.19)$$

où $L_f^{(n)} z$ est la dérivée $n^{\text{ième}}$ de Lie de la sortie mesurée z suivant le champ de vecteurs f .

A partir de (3.17) on définit les distributions suivantes :

$$\begin{aligned} G_0 &= \text{span}\{g_1, \dots, g_m\} \\ G_i &= G_{i-1} + \text{ad}_f^i G_{i-1} \quad \text{avec } i \geq 1 \end{aligned} \quad (3.20)$$

où span correspond à l'espace engendré par les vecteurs $g_1, \dots, g_m$ et $\text{ad}_f^i G_0$ correspond au crochet de Lie de f et de G_0 itéré i fois avec $\text{ad}_f^i = [f, \text{ad}_f^{i-1}]$, pour $i \geq 1$.

Le résultat suivant est issu de [33] et [29].

Théorème 3.3

Pour qu'un système non linéaire admette une linéarisation par bouclage statique, il faut et il suffit que dans un voisinage V du point d'équilibre $(x, u) = (0, 0)$:

- *La famille G_i soit une famille involutive et de rang constant dans V pour $i \in \{0, \dots, n-2\}$, où n la dimension du vecteur d'état ;*
- *G_{n-1} soit de rang n .*

Rappelons qu'une distribution G est dite involutive si, et seulement si, pour tout couple de champs de vecteurs f et g de G , le crochet de Lie de f et g appartient à G .

Notons également que la seconde condition du théorème est l'extension naturelle du critère de commandabilité dans le cas linéaire : $\text{rang}(B, AB, \dots, A^{n-1}B) = n$.

3.3.3.2.2. Linéarisation partielle par difféomorphisme et bouclage statique

Dans le cas où il n'est pas possible d'obtenir une linéarisation complète du système par bouclage statique et difféomorphisme (i.e. que le système non linéaire ne vérifie pas les conditions du théorème 3.3), on a démontré que le système en question peut être décomposé en une partie linéaire et une partie non linéaire, où Marino R. [49] donna la taille et la structure du plus grand sous-système linéarisable. La stabilisation n'est alors assurée que pour la partie linéarisable. Le comportement de la partie non linéarisable est a priori inconnue et on ne peut que constater la stabilité ou l'instabilité après bouclage.

Au vu de ce théorème on peut s'interroger sur la platitude d'un système partiellement linéarisable par bouclage statique. Dans le cas de systèmes mono-entrée partiellement linéarisables par bouclage d'état statique, Charlet B. et al [15] ont montré que l'extension au bouclage dynamique n'apporte rien de plus au problème de la linéarisation. Ces systèmes ne sont donc pas linéarisables et constituent des systèmes non plats. Par contre dans le cas multi-entrées il est possible de linéariser le comportement entrées/état par une extension dynamique endogène réalisée à l'aide d'un bouclage dynamique d'état. Le système étant alors linéarisable sur l'état, la stabilisation s'effectue sur le système linéaire équivalent. Ce type de bouclage est généralement appelé bouclage dynamique endogène car, contrairement au bouclage dynamique exogène, il n'existe pas de variables supplémentaires indépendantes du système d'origine.

3.3.3.2.3. Linéarisation par bouclage dynamique endogène

Une fois que l'on a résolu la principale difficulté, c'est à dire la détermination de sorties plates $(y_1, \dots, y_m)$ du système, les techniques classiques de découplage par bouclage dynamique ou quasi-statique permettent la construction du bouclage linéarisant et du difféomorphisme qui transforme le système en une chaîne d'intégrateurs purs formée à partir de $(y_1, \dots, y_m)$ [14].

On cherche à déterminer le bouclage dynamique endogène de la forme suivante:

$$\begin{cases} \dot{\xi} = a(x, \xi) + b(x, \xi)v \\ u = c(x, \xi) + d(x, \xi)v \end{cases} \quad \xi \in \mathbb{R}^p, v \in \mathbb{R}^m \quad (3.21)$$

et un difféomorphisme ϕ sur l'espace étendu z :

$$z = \phi(x, \xi) \quad z \in \mathbb{R}^{n+p} \quad (3.22)$$

De telle manière que le système étendu résultant suivant soit linéarisable par bouclage statique.

$$\dot{z} = \begin{pmatrix} \dot{x} \\ \dot{\xi} \end{pmatrix} = \begin{pmatrix} f(x, a(x, \xi) + b(x, \xi)v) \\ c(x, \xi) + d(x, \xi)v \end{pmatrix} \quad z \in \mathbb{R}^{n+p} \quad (3.23)$$

Alors en utilisant la technique de linéarisation par bouclage statique et difféomorphisme présentée précédemment sur ce système étendu on obtient un système linéaire équivalent que l'on peut mettre sous la forme canonique de Brunovsky suivante:

$$\begin{cases} y_1^{(k_1)} = v_1 \\ \vdots \\ y_m^{(k_m)} = v_m \end{cases} \quad (3.24)$$

avec $k_1, \dots, k_m$ les indices de commandabilité associées à $y_1, \dots, y_m$.

Remarques

- La linéarisation par bouclage n'a d'intérêt que si l'on sort du domaine ou l'approximation tangente est acceptable, car si un système est linéarisable par difféomorphisme et bouclage au voisinage d'un point d'équilibre, alors son linéarisé tangent est commandable en ce point. Ceci constitue une condition nécessaire mais pas suffisante. En effet il existe des systèmes dont le linéarisé tangent est commandable, mais qui ne sont pas linéarisables par bouclage ;
- La platitude peut être vue comme l'extension de la méthode dite du couple calculé au cas des systèmes ayant un vecteur de commande de dimension inférieure à la dimension du vecteur d'état. Les coordonnées généralisées q sont alors remplacées par les sorties plates, le bouclage statique est remplacé par un bouclage dynamique, et la dynamique du second ordre après bouclage est remplacée par $y^{(k)} = v$.

Dans ce qui suit, nous allons appliquer cette technique sur le robot unicycle.

3.3.3.3. Application de la linéarisation par bouclage au robot unicycle

On reprend le modèle décrivant la cinématique du robot unicycle :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (3.25)$$

Ce système non linéaire n'est pas linéarisable au sens des entrées/état par bouclage statique. En effet, la condition nécessaire et suffisante donnée par le théorème 3.2 n'est pas respectée, car la distribution G_1 n'est pas involutive.

Dans ce cas, si le système est plat, il est toujours possible d'obtenir une linéarisation complète par bouclage dynamique endogène. Pour ce faire, on procède comme suit :

- 1- On choisit un vecteur de sortie plate candidate de dimension 2 (égale à la dimension du vecteur de commande) ;
- 2- On procède à la dérivation successive des sorties choisies jusqu'à faire apparaître les entrées de commande. Si le système est inversible par rapport aux sorties choisies, l'algorithme s'arrête. Au cours de ce processus, on introduit des intégrations (compensateurs dynamiques) afin d'éviter d'avoir à dériver les entrées de commande. Cette extension permet de construire l'état ξ du compensateur.
- 3- Si la somme des ordres de dérivation des sorties est égal à la dimension de l'état augmenté donnée par $n + p$, p étant l'ordre du compensateur dynamique, alors on obtient une linéarisation complète de l'état.

Soit le vecteur de sortie plate candidate : $\eta = (x, y)^T$, correspondant à la position du robot.

- Première dérivation :

$$\begin{cases} \dot{\eta}_1 = \dot{x} = v \cos \theta \\ \dot{\eta}_2 = \dot{y} = v \sin \theta \end{cases} \quad (3.26)$$

Le système obtenu n'est pas inversible. Seulement une composante du vecteur de commande apparaît au terme de cette première dérivation.

- Dérivation d'ordre 2 :

On voit bien que si on procède directement à la dérivation d'ordre 2, on aura à dériver la composante v du vecteur de commande.

Dans ce cas, on ajoute un intégrateur sur cette entrée comme suit :

$$\begin{cases} v = \xi \\ \dot{\xi} = a \end{cases} \quad (3.27)$$

où : a est la nouvelle entrée de commande (correspond à l'accélération du robot)

On aura alors :

$$\begin{cases} \dot{\eta}_1 = \xi \cos \theta \\ \dot{\eta}_2 = \xi \sin \theta \end{cases} \quad (3.28)$$

et le système d'ordre 2 est donné par :

$$\begin{cases} \ddot{\eta}_1 = \dot{\xi} \cos \theta - \xi \dot{\theta} \sin \theta \\ \ddot{\eta}_2 = \dot{\xi} \sin \theta + \xi \dot{\theta} \cos \theta \end{cases} \quad (3.29)$$

Qu'on peut récrire sous forme matricielle, sachant que $\dot{\xi} = a$ et $\dot{\theta} = \omega$:

$$\ddot{\eta} = \begin{pmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\xi \sin \theta \\ \sin \theta & \xi \cos \theta \end{pmatrix} \begin{pmatrix} a \\ \omega \end{pmatrix} \quad (3.30)$$

avec $\begin{pmatrix} a \\ \omega \end{pmatrix}$ le nouveau vecteur de commande.

la matrice $\begin{pmatrix} \cos \theta & -\xi \sin \theta \\ \sin \theta & \xi \cos \theta \end{pmatrix}$ est non singulière à condition que $\xi \neq 0$. (condition d'excitation persistante). Ce qui met fin à la procédure de dérivation.

Si maintenant, on définit :

$$\begin{pmatrix} a \\ \omega \end{pmatrix} = \begin{pmatrix} \cos \theta & -\xi \sin \theta \\ \sin \theta & \xi \cos \theta \end{pmatrix}^{-1} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (3.31)$$

On obtient un système d'équations linéaires indépendantes :

$$\begin{cases} \ddot{\eta}_1 = u_1 \\ \ddot{\eta}_2 = u_2 \end{cases} \quad (3.32)$$

avec la dynamique du compensateur donnée par :

$$\begin{cases} \dot{\xi} = u_1 \cos \theta + u_2 \sin \theta \\ v = \xi \\ \omega = \frac{u_2 \cos \theta - u_1 \sin \theta}{\xi} \end{cases} \quad (3.33)$$

Dans notre cas, le compensateur ξ étant d'ordre 1, on a $n + p = 4$ qui est égal à la somme des ordres de dérivation des sorties η .

Dans ce cas, avec le nouveau vecteur d'état z du système augmenté tel que :

$$\begin{cases} z_1 = x \\ z_2 = y \\ z_3 = \dot{x} = \xi \cos \theta \\ z_4 = \dot{y} = \xi \sin \theta \end{cases} \quad (3.34)$$

On aboutit à une linéarisation complète du système, qui est désormais décrit par :

$$\begin{cases} \ddot{z}_1 = u_1 \\ \ddot{z}_2 = u_2 \end{cases} \quad (3.35)$$

Etant donné le système équivalent (3.35), le problème de poursuite de trajectoire revient à calculer la commande $u = (u_1 \ u_2)^T$ pour garantir un suivi d'une trajectoire de référence admissible $(x_d(t), y_d(t))$, telle que $v_d = \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \neq 0$ (condition d'excitation persistante, dictée par le fait que ξ ne doit pas tendre vers zéro).

Une commande de type Proportionnelle Dérivée permet d'assurer une stabilisation exponentielle du système double intégrateur (3.35), ce qui donnera pour u :

$$\begin{cases} u_1 = \ddot{x}_d + k_{p1}(x_d - x) + k_{d1}(\dot{x}_d - \dot{x}) \\ u_2 = \ddot{y}_d + k_{p2}(y_d - y) + k_{d2}(\dot{y}_d - \dot{y}) \end{cases} \quad (3.36)$$

avec : $k_{pi} > 0, k_{di} > 0$ pour $i = 1, 2$.

Enfin, les entrées de commande réelles du robot (v, ω) peuvent être déduites à partir du système d'équations du compensateur donné dans (3.33).

L'architecture de commande peut être schématisée comme suit (figure 3.3) :

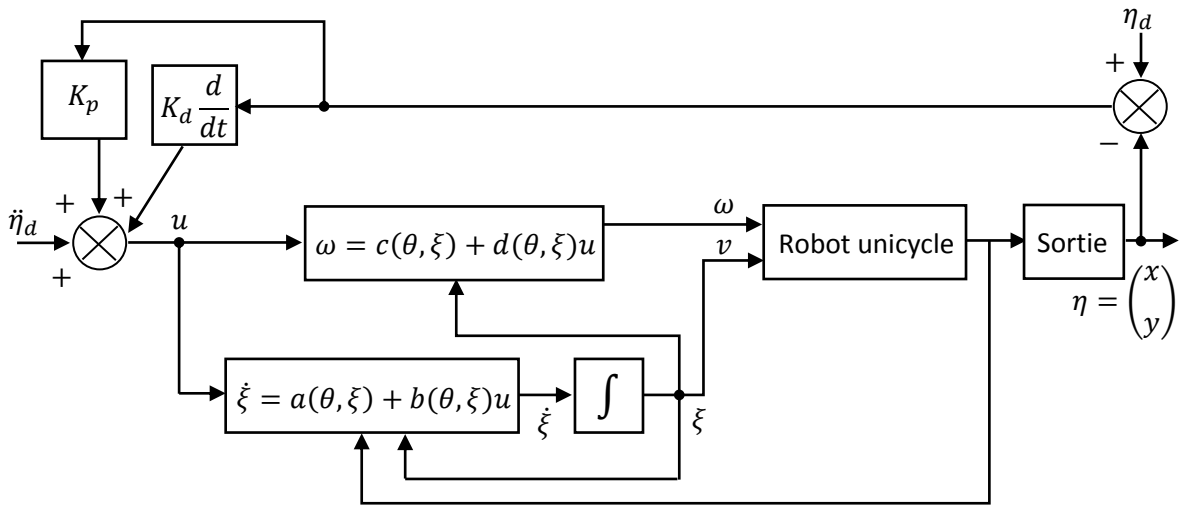


Figure 3.3 : Architecture de commande dédiée à la poursuite de trajectoire par la méthode de linéarisation par bouclage

3.3.4. Commande par retour d'état non linéaire non stationnaire continu

Dans cette approche, la problématique de poursuite de trajectoire est assimilée à une poursuite d'un robot virtuel exécutant la trajectoire de référence (figure 3.4).

Plus explicitement, l'objectif consiste à suivre une trajectoire de référence $q_r = (x_r y_r \theta_r)^T$ générée par le système :

$$\begin{cases} \dot{x}_r = v_r \cos \theta_r \\ \dot{y}_r = v_r \sin \theta_r \\ \dot{\theta}_r = \omega_r \end{cases} \quad (3.37)$$

qui correspond au modèle d'un robot virtuel du même type que le robot considéré, dans notre cas, un robot unicycle.

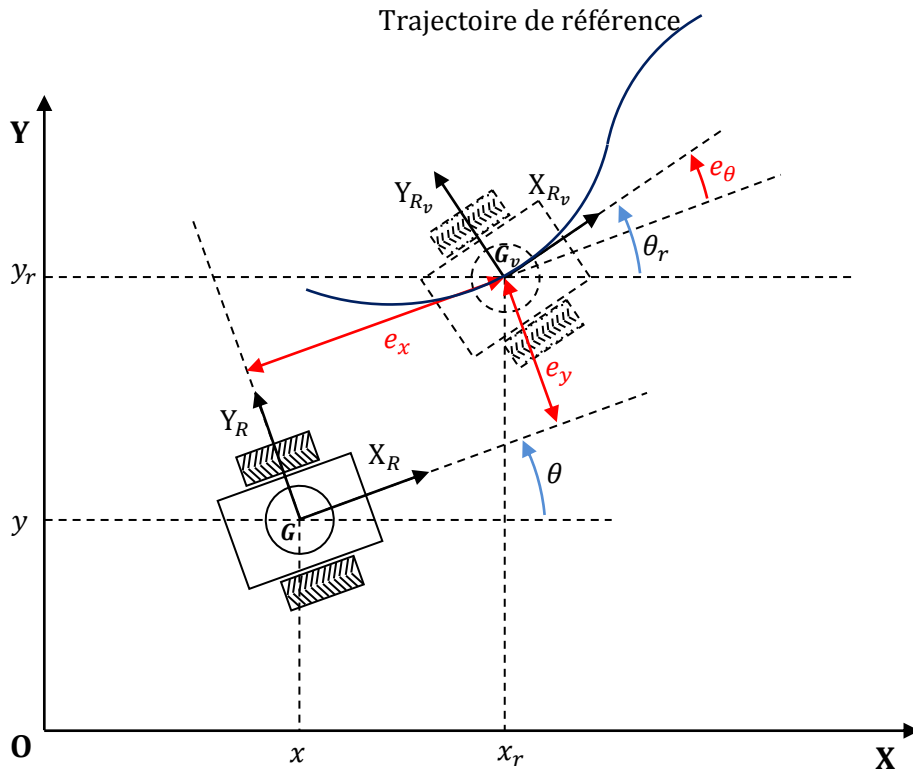


Figure 3.4 : Erreur de poursuite par rapport à un robot virtuel

L'erreur entre la configuration réelle et la configuration de référence, q_e est donnée par :

$$q_e = q_r - q = \begin{pmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{pmatrix} = \begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} \quad (3.38)$$

Maintenant, si on se place dans le repère mobile $\{GX_RY_R\}$ lié au robot, on peut définir l'erreur de poursuite e_p , qui s'exprime dans ce repère comme suit :

$$e_p = \begin{pmatrix} e_x \\ e_y \\ e_\theta \end{pmatrix} = T_e q_e = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} \quad (3.39)$$

où T_e est la matrice de transformation entre le repère $\{0 X Y\}$ et le repère lié au robot.

3.3.4.1. Modèle cinématique de l'erreur de poursuite

En dérivant le système (3.39), on obtient le système d'équations :

$$\dot{e}_p = \begin{pmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{pmatrix} = \begin{pmatrix} \omega e_y - v + v_r \cos e_\theta \\ -\omega e_x + v_r \sin e_\theta \\ \omega_r - \omega \end{pmatrix} \quad (3.40)$$

qui caractérise la dynamique de l'erreur de poursuite.

Le problème de poursuite peut être reformulé en termes de l'erreur e_p , comme suit :

Trouver une loi de commande permettant de stabiliser l'erreur de poursuite e_p à l'origine.

Pour ce faire, nous présentons deux lois de commande assez connues dans la littérature, qui serviront de modèles de comparaison avec la technique de linéarisation par bouclage développée dans le chapitre précédent.

3.3.4.2. Commande n° 1 : proposée dans [32]

Dans cette approche, on se base sur le critère de stabilité au sens de Lyapunov pour dériver la loi de commande stabilisante.

Soit le système (3.40), la fonction candidate de Lyapunov V est donnée par :

$$V = \frac{1}{2} e_x^2 + \frac{1}{2} e_y^2 + \frac{1}{c_2} (1 - \cos e_\theta), \text{ avec } c_2 > 0 \quad (3.41)$$

qui est une fonction définie positive.

En dérivant la fonction (3.41) on obtient :

$$\begin{aligned} \dot{V} &= \dot{e}_x e_x + \dot{e}_y e_y + \dot{e}_\theta \frac{1}{c_2} \sin e_\theta \\ &= (\omega e_y - v + v_r \cos e_\theta) e_x + (-\omega e_x + v_r \sin e_\theta) e_y + (\omega_r - \omega) \frac{\sin e_\theta}{c_2} \end{aligned}$$

$$\dot{V} = (-v + v_r \cos e_\theta) e_x + (\omega_r - \omega + c_2 v_r e_y) \frac{\sin e_\theta}{c_2} \quad (3.42)$$

Si on choisit les entrées de commande v et ω tel que :

$$\begin{cases} v = v_r \cos e_\theta + c_1 e_x \\ \omega = \omega_r + c_2 v_r e_y + c_3 \sin e_\theta \end{cases} \text{ avec : } c_1, c_3 > 0 \quad (3.43)$$

$\dot{V}$ devient :

$$\dot{V} = c_1 e_x^2 - \frac{c_3}{c_2} \sin^2 e_\theta < 0 \quad \forall e_x \neq 0, e_\theta \neq 0 \quad (3.44)$$

Du moment que $\dot{V}_1$ est semi définie négative et que e_y converge vers zéro, selon le théorème de Lasalle, le système décrit par (3.40) est globalement asymptotiquement stable.

3.3.4.3. Commande n° 2 : proposée dans [61], [34]

Soit le système (3.40). Si on choisit la loi de commande suivante :

$$\begin{cases} v = v_r + c_2 e_x - c_3 \omega_r e_y & c_1 > 0 \\ \omega = \omega_r + c_1 e_\theta & c_2 > 0, \quad c_3 > -1 \end{cases} \quad (3.45)$$

Dans ce cas, on obtient :

$$\begin{aligned} \underbrace{\begin{pmatrix} \dot{e}_x \\ \dot{e}_y \end{pmatrix} = \begin{pmatrix} -c_2 & (1+c_3)\omega_r \\ -\omega_r & 0 \end{pmatrix} \begin{pmatrix} e_x \\ e_y \end{pmatrix}}_{\dot{z}_1 = f_1(t, z_1)} + \underbrace{\begin{pmatrix} c_1 e_y + v_r \frac{\cos e_\theta}{e_\theta} \\ -c_1 e_x + v_r \frac{\sin e_\theta}{e_\theta} \end{pmatrix} e_\theta}_{g(t, z_1, z_2) z_2} \\ \underbrace{\dot{e}_\theta = -c_1 e_\theta}_{\dot{z}_2 = f_2(t, z_2)} \end{aligned} \quad (3.46)$$

Il apparaît clairement que le sous système $\dot{z}_2 = f_2(t, z_2)$ est globalement exponentiellement stable.

Pour démontrer la stabilité du sous système $\dot{z}_1 = f_1(t, z_1)$, le lemme suivant est utilisé :

Lemme 3.1 [40]

Soit le système :

$$\dot{x} = \begin{pmatrix} -c_1 & -c_2 \phi(t) \\ c_3 \phi(t) & 0 \end{pmatrix} x, \quad x \in \mathbb{R}^2 \quad (3.47)$$

Si $c_1 > 0$, $c_2, c_3 > 0$ et $\phi(t)$ est une excitation persistante, alors le système (3.47) est globalement exponentiellement stable.

Dans ce cas, pour que le système $\dot{z}_1 = f_1(t, z_1)$ soit globalement exponentiellement stable, il faut que ω_r soit une excitation persistante.

Pour finir, il est prouvé dans [69], à condition que v_r soit bornée, que le système (3.46) est globalement κ -exponentiellement stable, moyennant la supposition suivante sur $g(t, z_1, z_2)z_2$:

Il existe deux fonctions continues $k_1: \mathbb{R}^+ \rightarrow \mathbb{R}$, et $k_2: \mathbb{R}^+ \rightarrow \mathbb{R}$ telles que :

$$\|g(t, z_1, z_2)\| \leq k_1(\|z_1\|) + k_2(\|z_2\|)\|z_1\| \quad (3.48)$$

où $\|g(t, z_1, z_2)\|$ est la norme de Frobenius de la matrice $g(t, z_1, z_2)$.

L'architecture de commande, dans les deux lois précédentes, peut être schématisée comme suit (figure 3.5) :

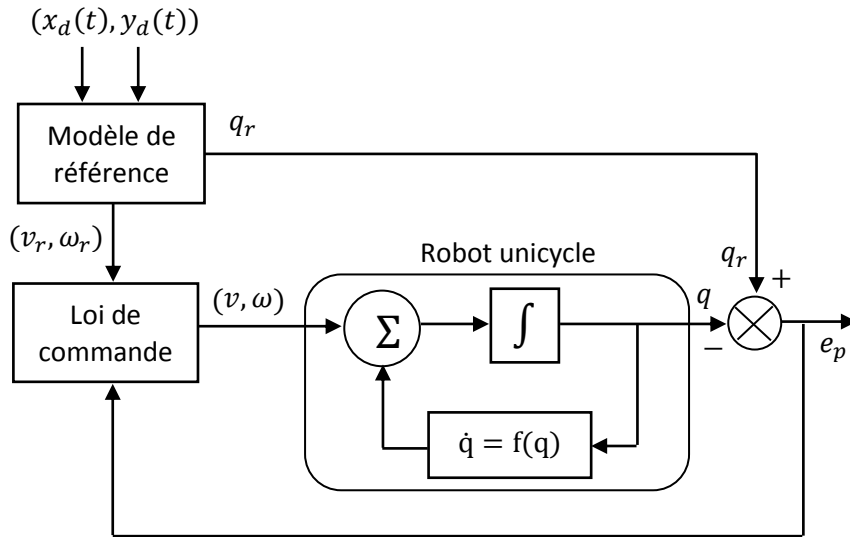


Figure 3.5 : Architecture de commande dédiée à la poursuite de trajectoire par la méthode directe

3.3.4.4. Génération de la trajectoire et du modèle de référence

Pour obtenir l'expression des commandes du robot (3.43) et (3.45), il nous reste à calculer le modèle de référence du robot virtuel qui exécute la trajectoire qu'on vise à suivre.

Etant donné une trajectoire désirée $(x_d(t), y_d(t))$ admissible, qu'on veut faire suivre par le robot. L'objectif maintenant est de calculer les variables d'état ainsi que les entrées de commande relatives au robot de référence $q_r = (x_r y_r \theta_r)^T$, $\dot{q}_r$, v_r , ω_r qui lui permettraient d'exécuter la trajectoire désirée.

Pour ce faire, on procède comme suit :

Un premier résultat intuitif est de poser :

$$\begin{cases} x_r(t) = x_d(t) \\ y_r(t) = y_d(t) \end{cases} \quad (3.49)$$

A partir de ce point, il est possible de calculer les autres variables en utilisant le modèle cinématique du robot.

En effet, puisque la trajectoire de référence est admissible, elle vérifie l'équation du système, et on peut écrire :

$$\dot{x}_r = v_r \cos \theta_r \quad (3.50)$$

$$\dot{y}_d = v_r \sin \theta_r \quad (3.51)$$

$$\dot{\theta}_r = \omega_r \quad (3.52)$$

Pour des conditions initiales $(x_r(t_0), y_r(t_0), \theta_r(t_0))$ données, v_r peut être obtenue à partir de (3.50) et (3.51) :

$$v_r = \pm \sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \quad (3.53)$$

où le signe de v_r dépend du choix de l'exécution de la trajectoire en marche avant ou en marche arrière.

En divisant l'équation (3.51) par (3.52), et en tenant compte du signe de v_r , l'orientation désirée du robot est donnée par :

$$\theta_r(t) = \text{ATAN2} \left\{ \frac{\dot{y}_r(t)}{v_r(t)}, \frac{\dot{x}_r(t)}{v_r(t)} \right\} \quad (3.54)$$

En dérivant les équations (3.50) et (3.51) et en combinant les résultats avec élimination du terme $\dot{v}_r(t)$, on obtient :

$$\dot{\theta}_r(t) = \frac{\ddot{y}_r(t)\dot{x}_r(t) - \ddot{x}_r(t)\dot{y}_r(t)}{v_r^2(t)} \quad (3.55)$$

Les équations (3.53-3.55) donnent les expressions des variables $(v_r, \theta_r, \dot{\theta}_r)$ nécessaires pour reproduire la trajectoire de référence. Ces expressions dépendent uniquement de la trajectoire (3.49) et de ses dérivées jusqu'au second ordre. Par conséquent, pour garantir sa reproductibilité, la trajectoire de référence devrait être au moins deux fois différentiable.

3.3.5. Commande avec observateur de l'orientation du robot

En robotique mobile, la mesure de l'orientation du robot s'avère difficile et coûteuse, et bien qu'il existe des capteurs pour mesurer l'orientation, on recourt souvent à son estimation via un observateur. Dans ce cas, le problème de commande consistera d'abord à construire un observateur pour l'orientation, puis au calcul de la loi de commande qui assurera la stabilité à l'origine du système global (système + observateur).

Dans ce qui suit, nous présentons un résultat proposé dans [34], qui est basé sur la loi de commande n°2, présentée dans la section précédente, à laquelle est intégré un observateur de l'orientation du robot. L'architecture de commande est donnée dans le schéma de la figure 3.6.

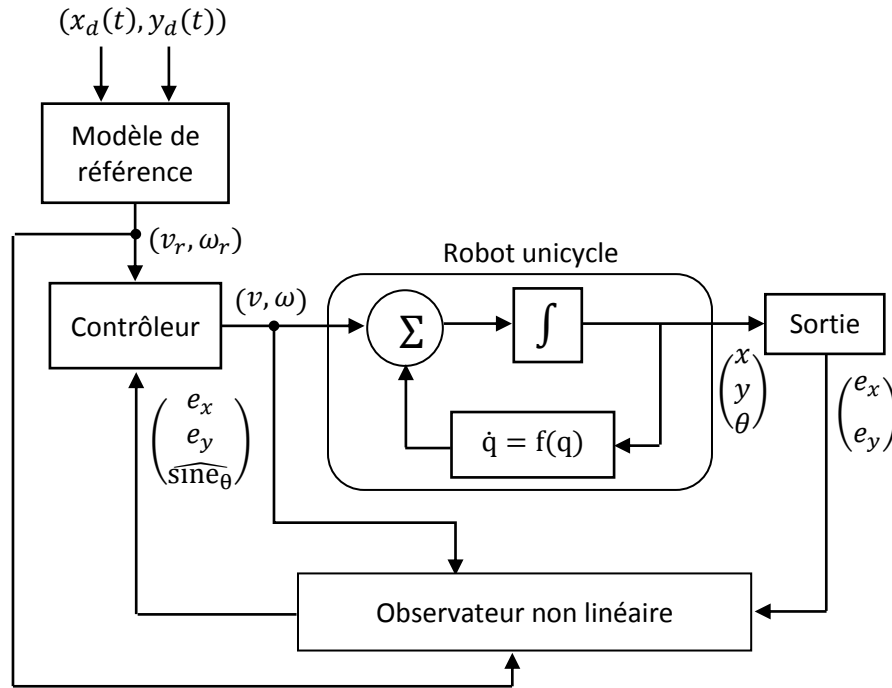


Figure 3.6 : Schéma-bloc de l'architecture de commande d'un unicycle avec observateur de l'orientation

Dans cette approche, Jakubiak et al proposent une modification de la loi de commande (3.45), où dans l'expression de ω , $(\sin e_\theta)$ remplace (e_θ) puisque cette dernière variable n'apparaît dans le système (3.40) que sous forme de cosinus et de sinus. La nouvelle loi de commande est donnée par :

$$\begin{cases} v = v_r + c_2 e_x - c_3 \omega_r e_y & c_2 > 0, c_3 > -1 \\ \omega = \omega_r + c_1 \sin e_\theta & c_1 > 0 \end{cases} \quad (3.56)$$

Ensuite, ils développent un observateur pour estimer le terme $(\sin e_\theta)$:

En posant :

$$z = \sin e_\theta - a v_r e_y \quad (3.57)$$

$$\hat{\phi} = \widehat{\sin e_\theta} \quad (3.58)$$

L'observateur est donné par :

$$\begin{cases} \dot{\hat{z}} = -a\dot{v}_r e_y + a v_r \omega e_x - a v_r^2 \hat{z} - a^2 v_r^3 e_y \\ \hat{\phi} = \hat{z} + a v_r e_y \end{cases} \quad (3.59)$$

Enfin, en récrivant (3.56) :

$$\begin{cases} v = v_r + c_2 e_x - c_3 \omega_r e_y & c_2 > 0, \quad c_3 > -1 \\ \omega = \omega_r + \frac{1}{2} \gamma a v_r^2 \hat{\phi} & a > 0, \quad 0 < \gamma < 1 \end{cases} \quad (3.60)$$

D'après [34], le système ainsi obtenu est localement exponentiellement stable à condition que : $e_\theta < \pi$, v_r et ω_r bornées et excitations persistantes, et $\dot{v}_r, \dot{\omega}_r$ bornées.

3.4. Suivi de Chemin

3.4.1. Formulation du problème

Le suivi de chemin (path following) consiste à minimiser l'écart entre un point G du robot et une courbe du plan, où la vitesse de déplacement du robot est prédéfinie et peut être constante, mais non nulle.

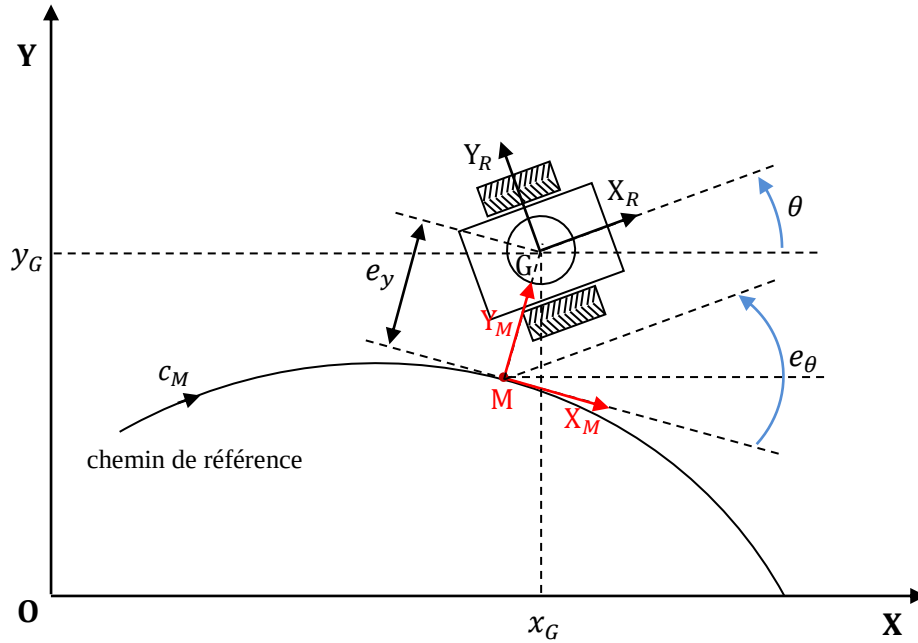


Figure 3.7 : Suivi de chemin pour l'unicycle

Cette problématique ressemble à celle présentée dans le cadre de la poursuite de trajectoire, où l'écart de poursuite est formulé par rapport à un repère mobile situé sur la courbe à

suivre, mais à une différence près. En effet, dans le cadre du suivi de chemin, le repère mobile n'évolue pas suivant une loi horaire indépendante, mais il est partiellement lié au robot.

Soit le repère mobile $\{M X_M Y_M\}$ défini sur le chemin, tel que l'axe X_M soit tangent au chemin et dirigé dans le sens du mouvement à suivre, et Y_M passe par le point G du robot (figure 3.7). On suppose que la distance MG est inférieure au rayon de courbure du chemin au point M, ainsi le chemin est défini de façon unique [64].

On peut maintenant définir l'erreur de suivi dans ce repère comme suit :

$$e_s = \begin{pmatrix} e_x \\ e_y \\ e_\theta \end{pmatrix} = T_s \times q_s = \begin{pmatrix} \cos \theta_M & \sin \theta_M & 0 \\ -\sin \theta_M & \cos \theta_M & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x - x_M \\ y - y_M \\ \theta - \theta_M \end{pmatrix} \quad (3.61)$$

où $q_s = (q - q_M)$ représente l'erreur entre la configuration du robot et le repère mobile exprimée dans $\{0 X Y\}$, et T_s la matrice de transformation entre le repère $\{0 X Y\}$ et le repère mobile $\{M X_M Y_M\}$.

La dynamique de l'erreur $\dot{e}_s$ est obtenue en dérivant l'équation (3.61).

En utilisant les relations trigonométriques suivantes :

$$\begin{aligned} \sin(a - b) &= \sin a \cos b - \cos a \sin b \\ \cos(a - b) &= \cos a \cos b + \sin a \sin b \end{aligned}$$

Et en tenant compte des contraintes :

$$\begin{aligned} v &= \dot{x} \cos \theta + \dot{y} \sin \theta \\ v_M &= \dot{x}_M \cos \theta_M + \dot{y}_M \sin \theta_M \end{aligned}$$

Cette dynamique s'écrit alors :

$$\dot{e}_s = \begin{pmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{pmatrix} = - \begin{pmatrix} v_M \\ 0 \\ \omega_M \end{pmatrix} + \begin{pmatrix} \cos e_\theta & -\sin e_\theta & 0 \\ \sin e_\theta & \cos e_\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ 0 \\ \omega \end{pmatrix} + \begin{pmatrix} 0 & \omega_M & 0 \\ -\omega_M & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} e_x \\ e_y \\ e_\theta \end{pmatrix} \quad (3.62)$$

$$\text{avec : } \omega = \dot{\theta} \text{ et } \omega_M = \dot{\theta}_M$$

Puisque l'axe Y_M passe par le point G du robot (figure 3.7), alors :

$$e_x(t) = \dot{e}_x = 0 \quad (3.63)$$

En utilisant ce dernier résultat dans la première équation de (3.62), on obtient :

$$-v_M + \cos e_\theta v + \omega_M e_y = 0 \quad (3.64)$$

En utilisant le fait que :

$$\omega_M = \frac{v_M}{\rho_M} = v_M c_M \quad (3.65)$$

où : ρ_M et c_M représentent respectivement, le rayon de courbure et la courbure du chemin au point M.

L'expression de ω_M est donnée alors par :

$$\omega_M = \frac{c_M v \cos e_\theta}{1 - c_M e_y} \quad (3.66)$$

avec $|c_M e_y| < 1$.

Finalement, en utilisant (3.63) dans les deux dernières équations de (3.62), on obtient la dynamique de l'erreur du suivi :

$$\begin{cases} \dot{e}_y = v \sin e_\theta \\ \dot{e}_\theta = e_\omega = \omega - \omega_M \end{cases} \quad (3.67)$$

avec ω_M donnée par (3.66).

Ainsi, le problème du suivi de chemin peut être reformulé comme suit :

Etant donné une vitesse d'avance du robot $v(t) > 0$, et sous la condition $|c_M e_y| < 1$, trouver une loi de commande ω pour le système (3.67) telle que :

$$\begin{aligned} \lim_{t \rightarrow \infty} e_y(t) &= 0 \\ \lim_{t \rightarrow \infty} e_\theta(t) &= 0 \end{aligned}$$

3.4.2. Synthèse de la commande

Pour la dérivation de la loi de commande, on utilise une approche basée sur le critère de stabilité au sens de Lyapunov, où la fonction de Lyapunov est construite par la méthode du backstepping [42].

On considère le système (3.67), et en utilise un changement de variable non linéaire défini comme suit :

$$\begin{cases} z_1 = e_y \\ z_2 = \sin e_\theta \end{cases} \quad (3.68)$$

avec une nouvelle entrée de commande u telle que :

$$u = e_\omega \cos e_\theta, \quad \text{avec } 0 \leq e_\theta < \pi/2 \quad (3.69)$$

On obtient ainsi un système écrit en fonction des nouvelles variables comme suit :

$$\begin{cases} \dot{z}_1 = v z_2 \\ \dot{z}_2 = u \end{cases} \quad (3.70)$$

Il s'agit d'un système linéaire en cascade, dont la synthèse de la commande u est facilitée par l'utilisation de la technique du backstepping.

Pour commencer, on essaye de stabiliser la première équation de (3.70).

On définit alors la fonction :

$$V_1 = \frac{1}{2} z_1^2 \quad (3.71)$$

Dont la dérivée est donnée par :

$$\dot{V}_1 = \dot{z}_1 z_1 = v z_2 z_1 \quad (3.72)$$

Si on prend comme commande virtuelle, la variable z_2 , telle que :

$$z_2 = z_2^d = -k_1 z_1, \quad \text{avec } k_1 > 0 \quad (3.73)$$

alors $\dot{V}_1 = -v k_1 z_1^2 < 0$ (définie négative).

Soit maintenant, la fonction augmentée V_2 définie comme suit :

$$V_2 = V_1 + \frac{1}{2} (z_2 - z_2^d)^2 \quad (3.74)$$

Sa dérivée est donnée par :

$$\begin{aligned} \dot{V}_2 &= \dot{z}_1 z_1 + (\dot{z}_2 - \dot{z}_2^d)(z_2 - z_2^d) \\ &= v z_2 z_1 + (u + k_1 v z_2)(z_2 + k_1 z_1) \end{aligned} \quad (3.75)$$

Si on choisit u , telle que :

$$u = -k_1 v z_2 - k_2 (z_2 + k_1 z_1) - v z_1, \quad \text{avec } k_2 > 0 \quad (3.76)$$

Alors, $\dot{V}_2$ devient :

$$\dot{V}_2 = -v k_1 z_1^2 - k_2 (z_2 + k_1 z_1)^2 < 0 \quad (3.77)$$

qui est définie négative.

Enfin, l'entrée de commande réelle ω du robot, peut être déduite en utilisant l'équation (3.69). Ce qui donne :

$$\omega = \frac{u}{\cos e_\theta} + \omega_M \quad (3.78)$$

L'architecture de commande peut être schématisée comme suit (figure 3.8) :

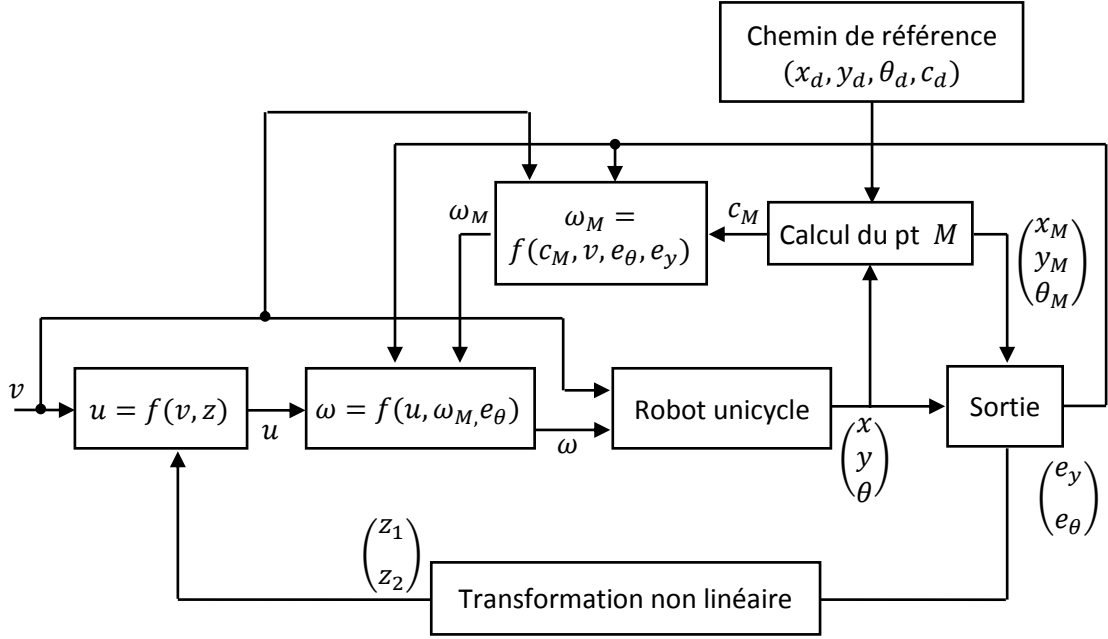


Figure 3.8 : Architecture de commande dédiée au suivi de chemin

3.5. Modèle dynamique du robot et synthèse du régulateur interne

Dans cette section, on va considérer le modèle dynamique relatif au robot unicycle.

Ce modèle, dont le détail des calculs se trouve en annexe B, est donné par :

$$\begin{cases} M\dot{v} = -K_v v + K_m E_{am} \\ J\dot{\omega} = -K_\omega \omega + K_d E_{ad} \end{cases} \quad (3.79)$$

où :

M : la masse du système (kg), J : son moment d'inertie (kg.m^2), $K_v, K_m, K_\omega, K_d = \text{ctes}$, E_{am}, E_{ad} tensions, moyenne respectivement différentielle (V).

Pour cette partie du système, dont les entrées de commande sont les tensions E_{am} et E_{ad} , qui sont l'image des entrées réelles du robot, on utilise une régulation de type Proportionnelle Intégrale (figure 3.9) pour obtenir les expressions qui permettent au robot de suivre les profils de vitesses linéaire et angulaire désirées, v_d et ω_d .

Ces expressions sont dans ce cas données par :

$$E_{am} = K_{p1}e_v + K_{i1} \int_0^t e_v(\tau) d\tau \quad (3.80)$$

$$E_{ad} = K_{p2}e_\omega + K_{i2} \int_0^t e_\omega(\tau) d\tau \quad (3.81)$$

avec : $e_v = v_d - v$, $e_\omega = \omega_d - \omega$, $K_{p1,2}, K_{i1,2} > 0$.

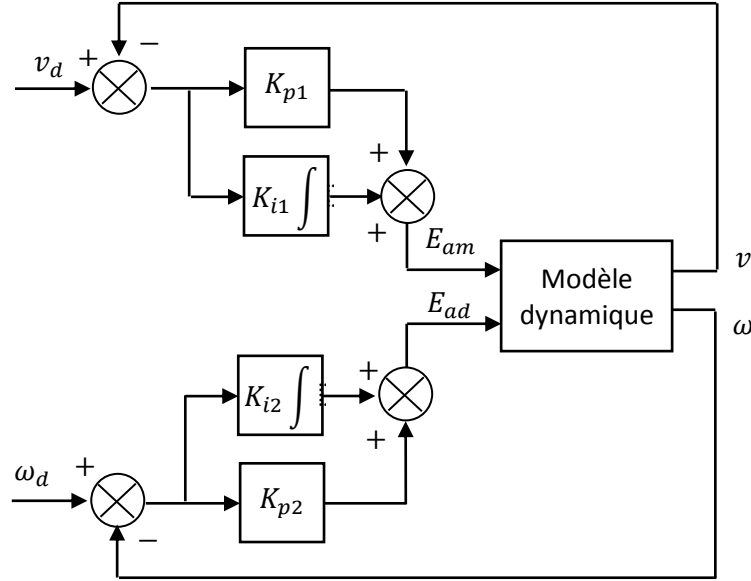


Figure 3.9 : Schéma bloc de la boucle interne avec réglage PI

3.6. Simulations et résultats

Dans cette section, on va tester, en simulation sous MATLAB, la validité des techniques de commande qui ont été présentées dans les paragraphes précédents.

Les simulations ont été effectuées en considérant le modèle dynamique du robot, où on a essayé de tenir compte des contraintes relatives à la charge et au frottement au niveau des deux roues motrices, ainsi qu'à la dynamique et la saturation des actionneurs.

Pour rappel, les techniques de commande dont il s'agit sont des techniques dédiées à la poursuite de trajectoire ou au suivi de chemin, c'est-à-dire que la trajectoire de référence est imposée, soit par choix de l'utilisateur, soit faisant partie d'un système plus global qui intègre une couche de planification de trajectoire dans le cadre d'une mission donnée (ex. navigation avec évitement d'obstacles).

Pour mener ces tests, notre choix pour les trajectoires de référence a porté sur trois types de trajectoires : circulaire, en forme de huit, et en ligne droite, débutant toutes à la configuration $(x, y, \theta)^T = 0_{3 \times 1}$.

De plus, pour tester la robustesse des techniques utilisées, des tests sont réalisés avec des perturbations de type bruit de modèle et bruit de mesure.

Ces perturbations peuvent être modélisées de la manière suivante :

❖ Bruit de modèle

Dans le cadre de cette étude, nous avons considéré que cette perturbation survient sur le modèle cinématique du robot. Elle est de la forme $b(t) = (b_1, b_2, b_3)'$, et intervient dans les équations du système (2.27) associé au robot unicycle, comme suit :

$$\begin{cases} \dot{x} = v \cos \theta + b_1(t) \\ \dot{y} = v \sin \theta + b_2(t) \\ \dot{\theta} = \omega + b_3(t) \end{cases} \quad (3.82)$$

Nous avons considéré deux classes de bruit de modèle, à savoir, un bruit aléatoire de type blanc gaussien de moyenne nulle et de variance $\sigma = (\sigma_1, \sigma_2, \sigma_3)'$, et un bruit déterministe de type échelon d'amplitude $m = (m_1, m_2, m_3)'$.

❖ Bruit de mesure

C'est une perturbation qui vient s'ajouter aux variables de sortie (x, y, θ) du système et qui est de type bruit blanc gaussien de moyenne nulle et de variance $\sigma' = (\sigma'_1, \sigma'_2, \sigma'_3)'$.

Dans le cadre de ce travail, on se limitera à l'étude de l'effet des perturbations sur la commande avec linéarisation par bouclage uniquement.

3.6.1. Résultats de simulation pour la technique de linéarisation par bouclage

On rappelle que la loi de commande permettant d'obtenir les consignes de vitesse du robot, v_d et ω_d , est donnée par les équations (3.33) et (3.36). Cette loi de commande existe tant que $v_d = \xi \neq 0$. Pour éviter la singularité, on définit un seuil minimum pour la variable v_d , soit 0.01 m/s . Dans ce cas, le robot se déplacera toujours en mode marche avant. Quant aux variables de commandes du modèle dynamique, e_{am} et e_{ad} , elles sont données par les équations (3.80) et (3.81).

En considérant le cas idéal, c'est-à-dire sans perturbations sur le système ou sur la sortie, on montre dans la figure suivante (figure 3.10) que la poursuite est assurée pour les trois types de trajectoires de référence choisies.

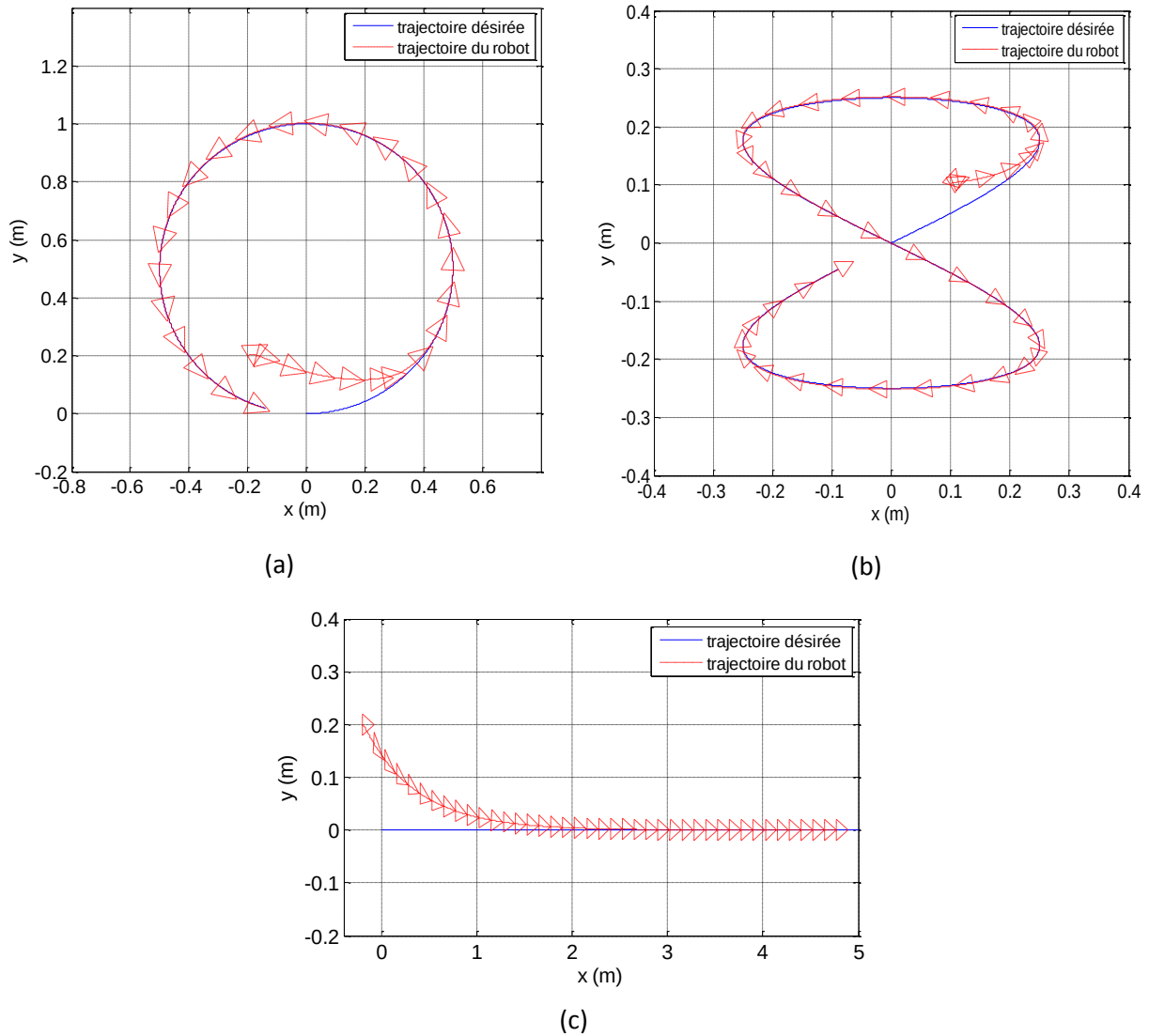


Figure 3.10 : Poursuite réussie. (a) trajectoire circulaire. (b) en forme de huit. (c) ligne droite

Les conditions initiales ont été prises comme suit :

$$(a) : x_0 = -0,2 \text{ m}, \quad y_0 = 0,2 \text{ m}, \quad \theta_0 = \pi/6 \text{ rad}$$

$$(b) : x_0 = 0,1 \text{ m}, \quad y_0 = 0,1 \text{ m}, \quad \theta_0 = \pi/6 \text{ rad}$$

$$(c) : x_0 = -0,2 \text{ m}, \quad y_0 = 0,2 \text{ m}, \quad \theta_0 = 0 \text{ rad}$$

Le robot a été représenté par un triangle afin de mieux apprécier le sens de déplacement et l'angle d'orientation.

3.6.1.1. Etude des performances

Cette section est dédiée à l'étude des performances de la technique de linéarisation par bouclage, où nous allons d'abord considérer le cas idéal, puis les cas où le système est soumis aux différentes perturbations, en prenant l'exemple de la trajectoire circulaire comme trajectoire de référence. Les valeurs des différents paramètres mis en jeu ont été choisies comme suit :

Conditions initiales du robot	Paramètres de la boucle interne	Paramètres de la boucle externe
$x_0 = -0,2 \text{ m}$	$K_{p1} = 10$ $K_{i1} = 0,9$	$k_{p1} = 100$ $k_{d1} = 20$
$y_0 = 0,2 \text{ m}$	$K_{p2} = 0.8$ $K_{i2} = 0,1$	$k_{p2} = 120$ $k_{d2} = 50$
$\theta_0 = \pi/6 \text{ rad}$		

3.6.1.1.1. Simulation dans le cas idéal (système sans perturbation)

Le comportement du robot est celui présenté dans la figure 3.10.a. A l'instant initial, afin d'éviter la singularité de la commande donnée par l'équation 3.33, la variable ξ est initialisée avec la valeur de la vitesse linéaire associée à la trajectoire de référence au point zéro. Nous remarquons tout de même qu'au démarrage, le robot prend une vitesse angulaire importante qui le fait pratiquement tourner sur lui-même (même remarque pour les cas b. et c.), ce qui implique de fortes demandes en courant au niveau des actionneurs. Pour y remédier, nous limitons la valeur absolue de la consigne en vitesse angulaire à 2 rad/s durant la phase de démarrage ([0 0,5]s). On obtient donc le résultat donné dans la figure (3.11.a). Cette astuce sera aussi adoptée pour les autres techniques de commande.

Les erreurs de poursuite selon x, y , ainsi que l'erreur d'orientation par rapport à la trajectoire de référence sont données dans la figure 3.11 b.

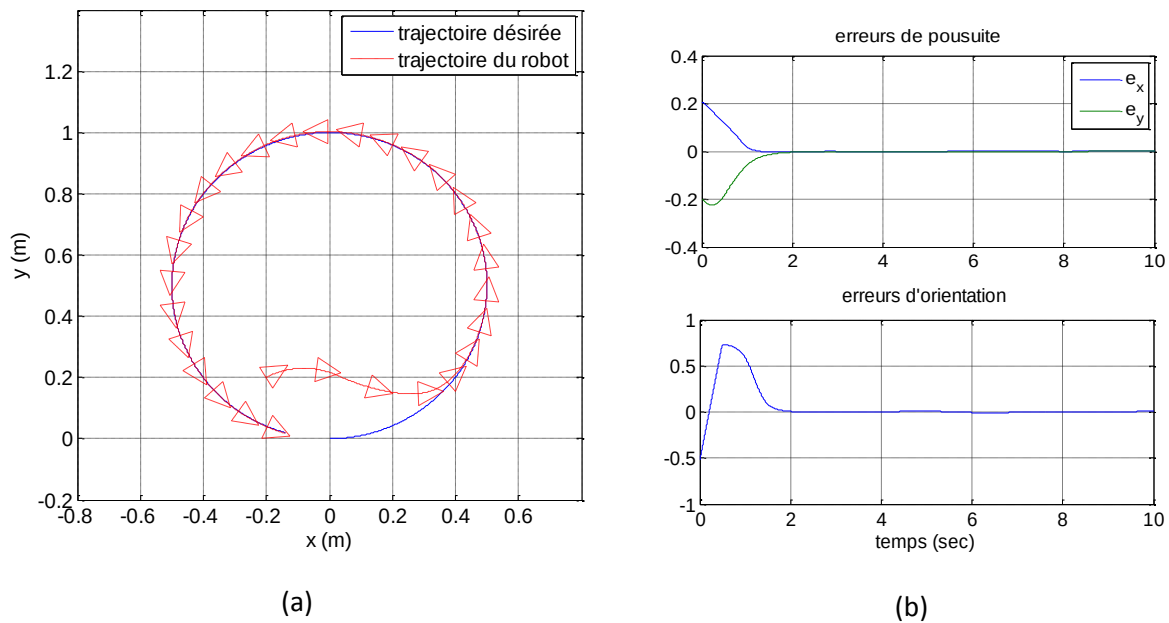


Figure 3.11 Poursuite de trajectoire. (a) trajectoire du robot. (b) erreurs de poursuite

La poursuite est assurée avec une précision de l'ordre de $10^{-3} m$ pour les deux variables de position, et de $10^{-3} rad$ pour l'orientation. Bien que la poursuite soit faite uniquement sur la position, l'orientation est implicitement régulée (propriété du système). Quant au temps de convergence, il est de 2 sec. Ce dernier facteur n'a de sens que lorsque nous aurons à comparer les performances de cette loi de commande avec les autres techniques pour les mêmes conditions initiales, puisqu'il dépend justement de ces conditions initiales (éloignement initial par rapport à la trajectoire) et des limites physiques du robot (vitesse max).

On s'intéresse maintenant aux entrées de commande du robot ($v, \omega, u_1, u_2, eam, ead$), qui correspondent aux entrées de commande des modèles cinématique et dynamique respectivement, lesquelles sont données dans la figure suivante (figure 3.12) :

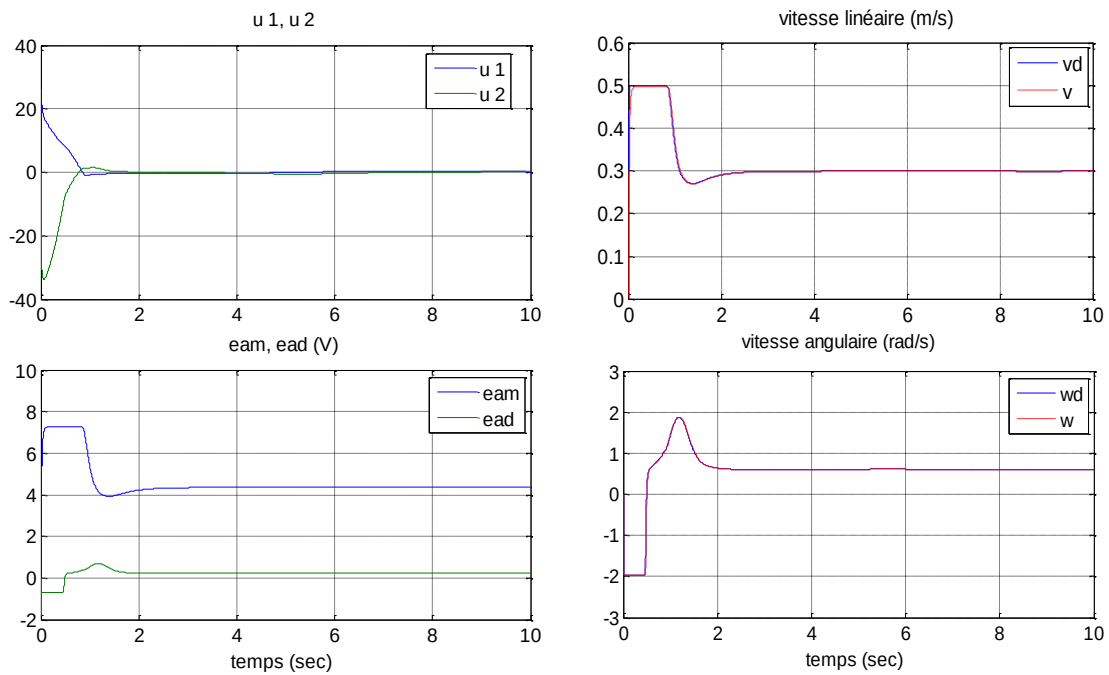


Figure 3.12 : Allure des entrées de commande du robot

La dynamique donnée aux commandes u_1 et u_2 a fait que les vitesses linéaire et angulaire du robot épousent les profils désirés quasi instantanément.

Afin d'éviter de saturer les actionneurs, nous avons limité les valeurs maximales des consignes des vitesses linéaire et angulaire comme suit :

$$v_{dmax} = 0,6 \times v_{max} \text{ du robot} = 0,5 \text{ m/s}$$

$$\omega_{dmax} = 0,25 \times \omega_{max} \text{ du robot} = 8,25 \text{ rad/s}$$

Nous présentons finalement, les signaux électriques appliqués sur les moteurs du robot pour générer les commandes précédentes (figure 3.13) :

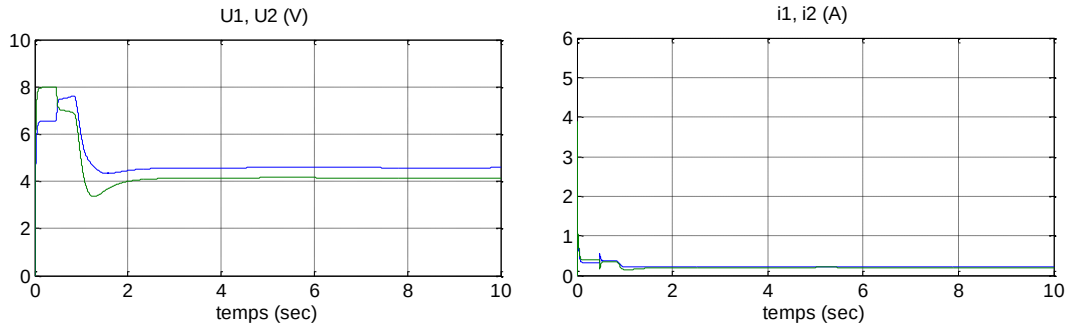


Figure 3.13 : Allure des signaux électriques appliqués aux moteurs

On note bien que ces signaux sont situés dans les limites des valeurs de fonctionnement normal des actionneurs.

3.6.1.1.2. Simulation avec perturbations

Le système de commande sous l'effet des différentes perturbations peut être représenté comme suit (figure 3.14) :

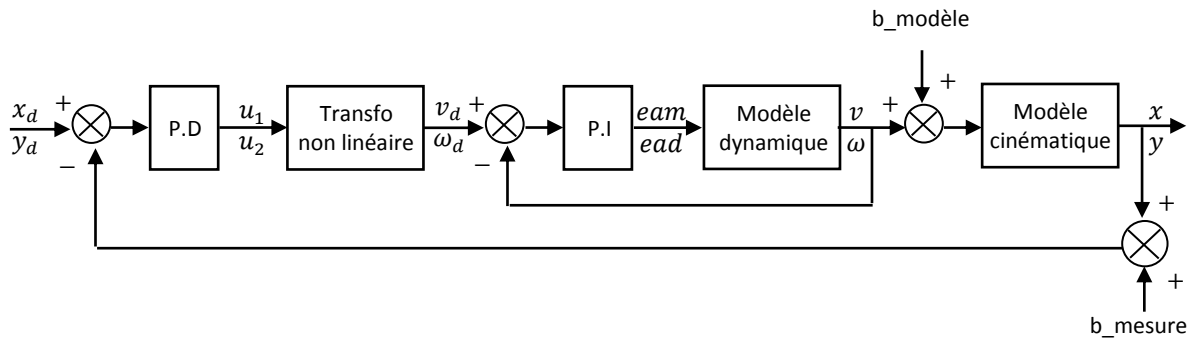


Figure 3.14 : Système de commande soumis aux différentes perturbations

En examinant ce système de commande, on voit que la sortie passe en premier lieu par le correcteur Proportionnel Dérivé, on peut tout de suite dire que cette technique de commande est inappropriée lorsqu'on fait face à des perturbations de type bruit blanc (que ce soit sur la mesure ou sur le modèle). En effet, cette loi de commande est très sensible à ce type de bruit, puisque l'action dérivée a pour effet d'amplifier les signaux hautes-fréquences. La diminution de gain k_d permettra d'atténuer cet effet, mais au détriment des autres performances (erreurs de poursuite, temps de convergence).

a) Système soumis à un bruit de modèle de type échelon

Nous avons effectué plusieurs tests avec différentes amplitudes du bruit, où nous avons trouvé que la commande pouvait tolérer jusqu'à $m = (0,24)_{3 \times 1}$.

On présente dans la figure suivante, deux cas où la poursuite n'est pas assurée, pour $m = (0,3)_{3 \times 1}$ et $m = (0,25)_{3 \times 1}$ respectivement.

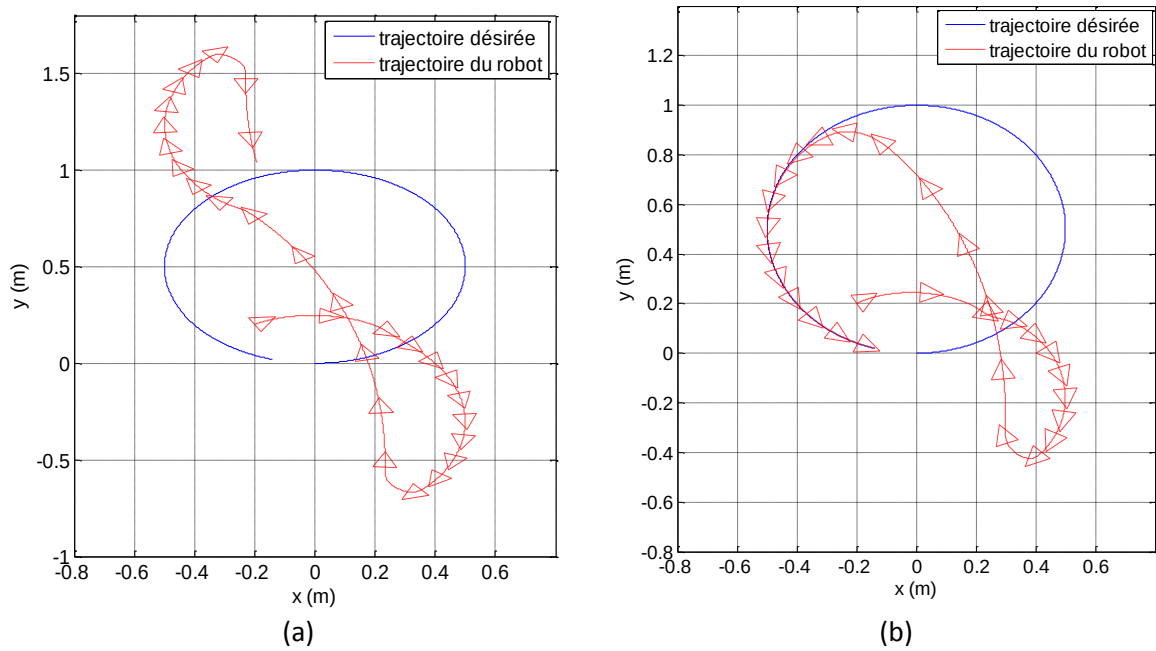


Figure 3.15 : comportement du robot pour différentes amplitudes du bruit.
(a) : $m = 0,3$. (b) : $m = 0,25$.

Dans le cas où $m = (0,24)_{3 \times 1}$, on présente dans les figures suivantes (3.16-3.18), le comportement du robot avec les erreurs de poursuite, les différentes entrées de commande, ainsi que les signaux électriques appliqués au niveau des actionneurs.

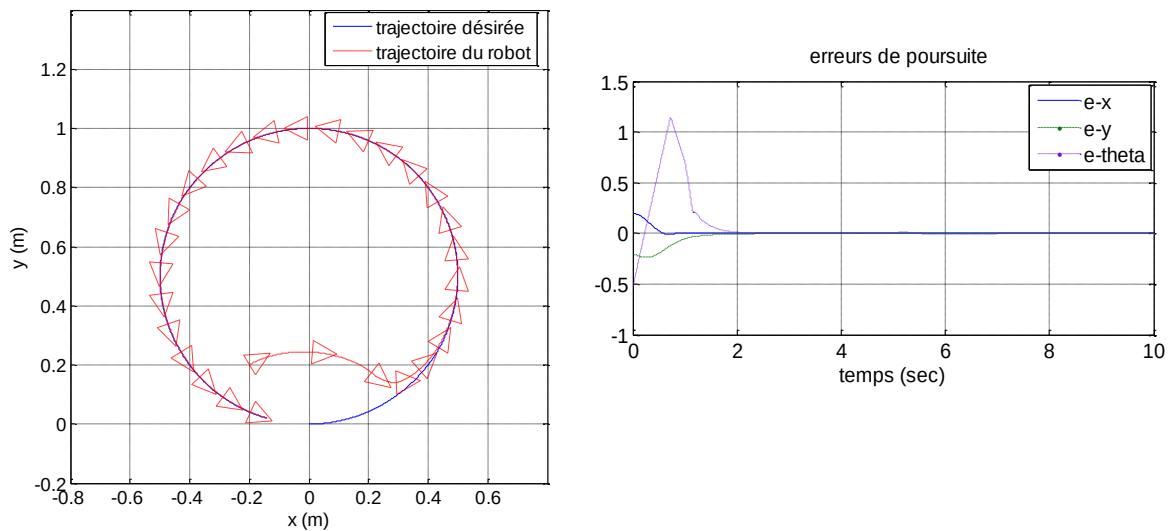


Figure 3.16 : comportement du robot soumis à une perturbation échelon d'amplitude $m = 0,24$.

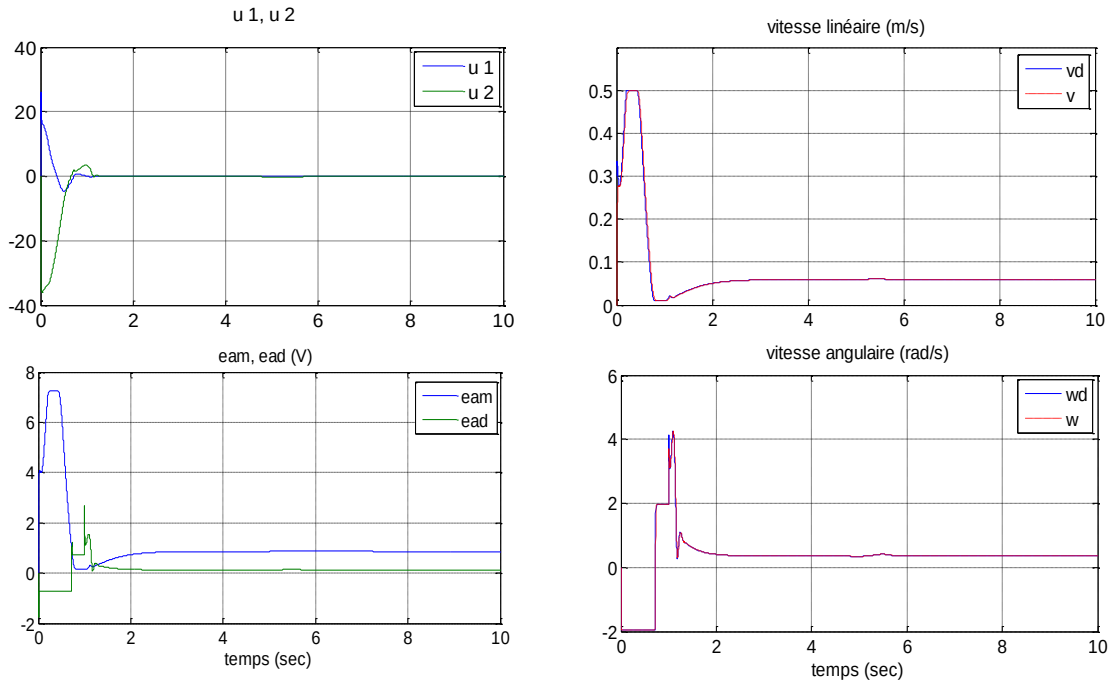


Figure 3.17 : allure des entrées de commande du robot

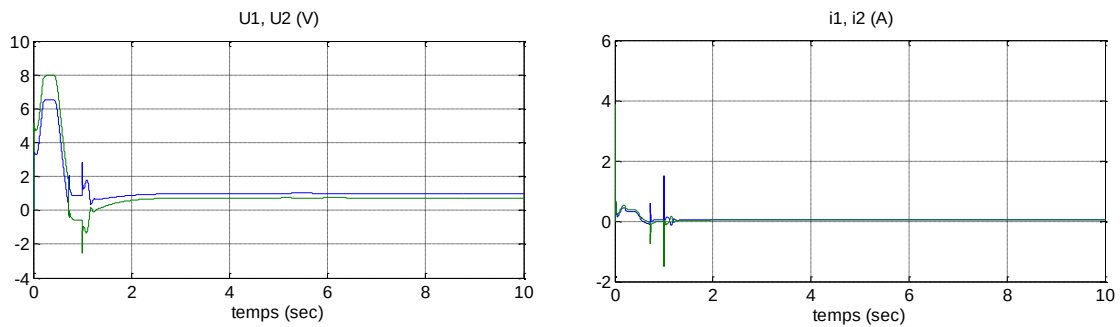


Figure 3.18 : allure des signaux électriques appliqués aux moteurs

b) Système soumis à un bruit de modèle de type bruit blanc gaussien

Dans ce cas, si on garde les mêmes valeurs pour les gains k_d , la variance σ du bruit ne doit pas dépasser 10^{-5} pour avoir des entrées de commande qui soient tolérables au niveau des actionneurs. Si par contre on diminue les gains de l'action dérivée, en posant par exemple : $k_{d1} = 5$ et $k_{d2} = 12,5$, on peut arriver jusqu'à une variance de 10^{-4} . Les résultats correspondant à cette dernière combinaison sont donnés dans les figures suivantes :

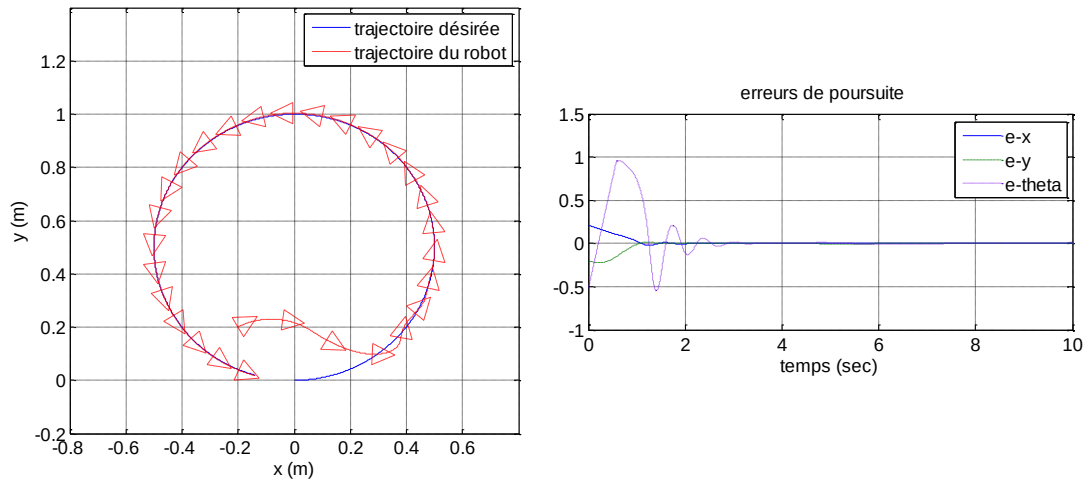


Figure 3.19 : comportement du robot soumis à un bruit de modèle blanc de variance $\sigma = 10^{-4}$.

Du fait de la diminution des gains k_d , le temps de convergence a augmenté puisqu'il est passé de 2 à 3 sec, et la précision de poursuite a diminué, 4×10^{-3} pour la position et 10^{-2} pour l'orientation.

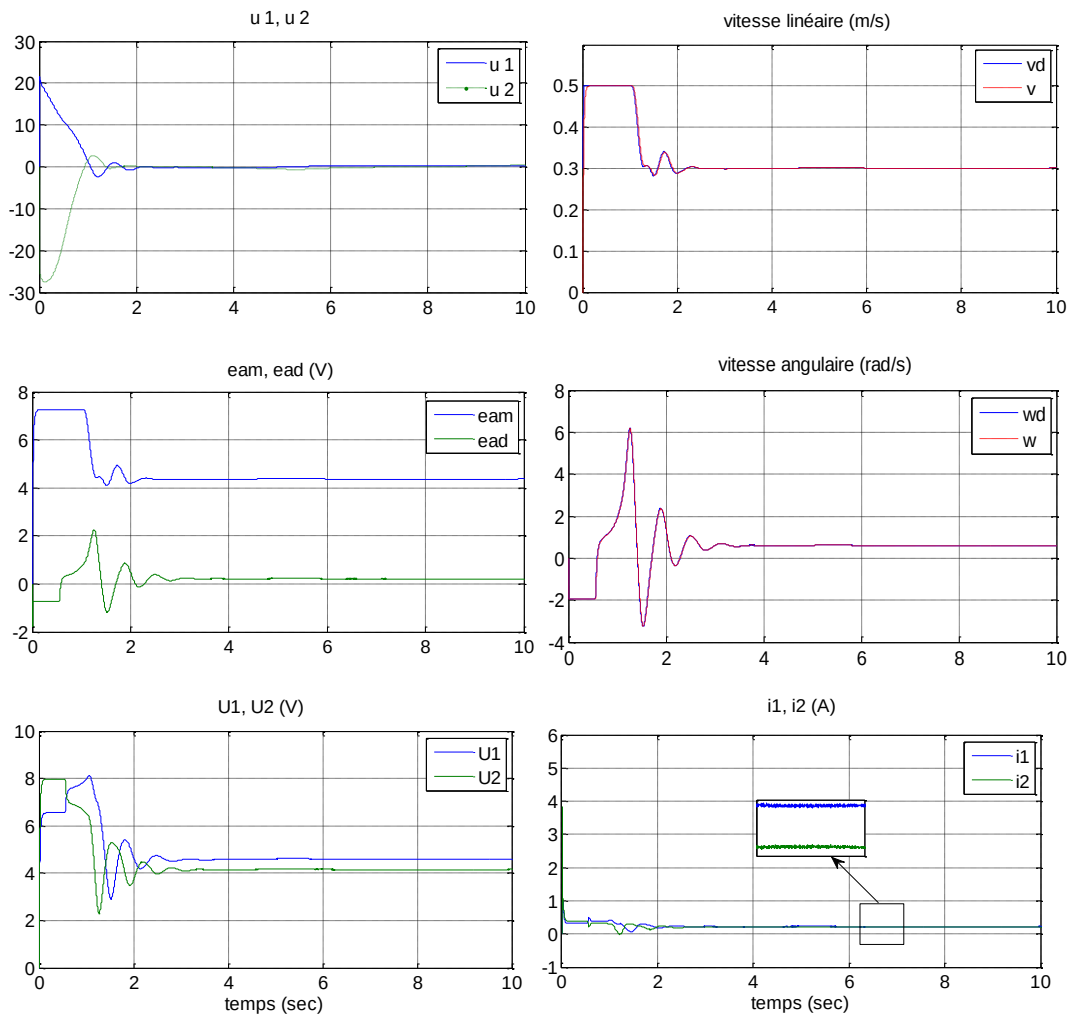


Figure 3.20 : allure des entrées de commande et des signaux électriques du robot

c) Système soumis à un bruit de mesure de type bruit blanc gaussien

La sensibilité de la commande à un bruit blanc sur la mesure est beaucoup plus importante que dans le cas d'un bruit de même type intervenant en entrée (point b), puisque dans ce dernier cas, le bruit passe par le système et se voit donc filtré. Par contre, le bruit de mesure est directement introduit dans le correcteur proportionnel dérivé, qui l'amplifie et aboutit à des commandes hautes fréquences qui peuvent détruire le système. Malgré une diminution significative des gains k_d ($k_{d1} = 2$, et $k_{d2} = 5$), la commande reste sensible à des bruits de mesure dont la variance est de l'ordre de 10^{-6} .

3.6.1.2. Evaluation et conclusion partielle

La technique de commande avec linéarisation par bouclage assure, en l'absence de perturbations, une bonne poursuite des trois types de trajectoires de références considérés, même pour des conditions initiales assez éloignées de la trajectoire de référence. La poursuite est aussi assurée dans le cas où le système est soumis à des perturbations de type échelon (bon rejet des perturbations de ce type). Néanmoins, cette technique de commande s'avère être très sensible vis-à-vis des perturbations hautes-fréquences (bruit blanc) intervenant en entrée (sur le système) ou en sortie (sur la mesure), et est donc inappropriée dans ces cas de figure.

3.6.2. Résultats de simulation pour la technique de commande non stationnaire continue

On rappelle que nous avons proposé deux lois de commande pour la partie représentant le modèle cinématique, qui sont données par les équations (3.43) et (3.45). Quant aux variables de commandes du modèle dynamique, eam et ead , elles sont données par les équations (3.80) et (3.81).

Nous commençons d'abord par démontrer que dans le cas de la commande n°2, la poursuite n'est pas assurée pour une trajectoire de référence de type ligne droite. En effet, nous avons vu que la stabilité du système $\dot{z}_1 = f_1(t, z_1)$ de (3.46) est conditionnée par le fait que ω_r soit une excitation persistante, ce qui n'est pas le cas pour une ligne droite. (voir figure 3.21 ci-après).

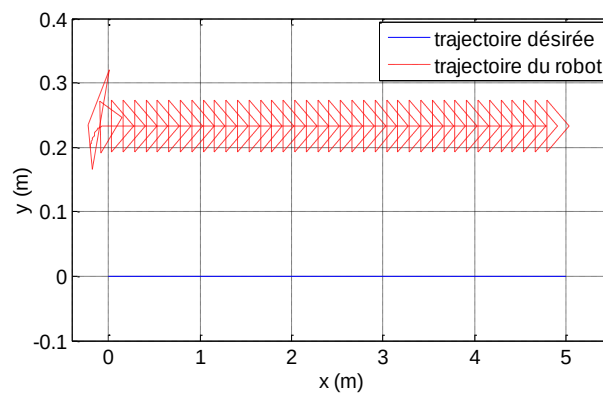


Figure 3.21 : Echec de la poursuite de ligne droite pour la commande n°2
($x_0 = -0,2 \text{ m}$, $y_0 = 0,2 \text{ m}$, $\theta_0 = \pi/6 \text{ rad}$)

Dans (3.45), la commande ω_d ne dépendant que de e_θ , une fois cette erreur éliminée, ω_d s'annule à son tour et le robot n'est alors animé que par une vitesse linéaire (il ne converge pas vers la trajectoire de référence).

3.6.2.1. Etude des performances

Dans cette section nous allons étudier les performances des deux lois de commande, que nous présentons sous forme d'une étude comparative, en prenant l'exemple de la trajectoire circulaire comme trajectoire de référence. Les valeurs des différents paramètres mis en jeu ont été choisies comme suit :

Conditions initiales du robot	Paramètres de la boucle interne		Paramètres de la boucle externe	
$x_0 = -0,2 \text{ m}$	$K_{p1} = 10$	$K_{i1} = 0,9$	Cde n°1	Cde n°2
$y_0 = 0,2 \text{ m}$			$c_1 = 16$	$c_1 = 20$
$\theta_0 = \pi/6 \text{ rad}$	$K_{p2} = 0.8$	$K_{i2} = 0,1$	$c_2 = 20$	$c_2 = 2.8$
			$c_3 = 2.2$	$c_3 = 6.6$

3.6.2.1.1. Simulation dans le cas idéal (système sans perturbation)

Le comportement du robot est celui présenté dans la figure 3.22. De la même manière que pour la commande précédente on limite la valeur de ω_d pendant la phase de démarrage.

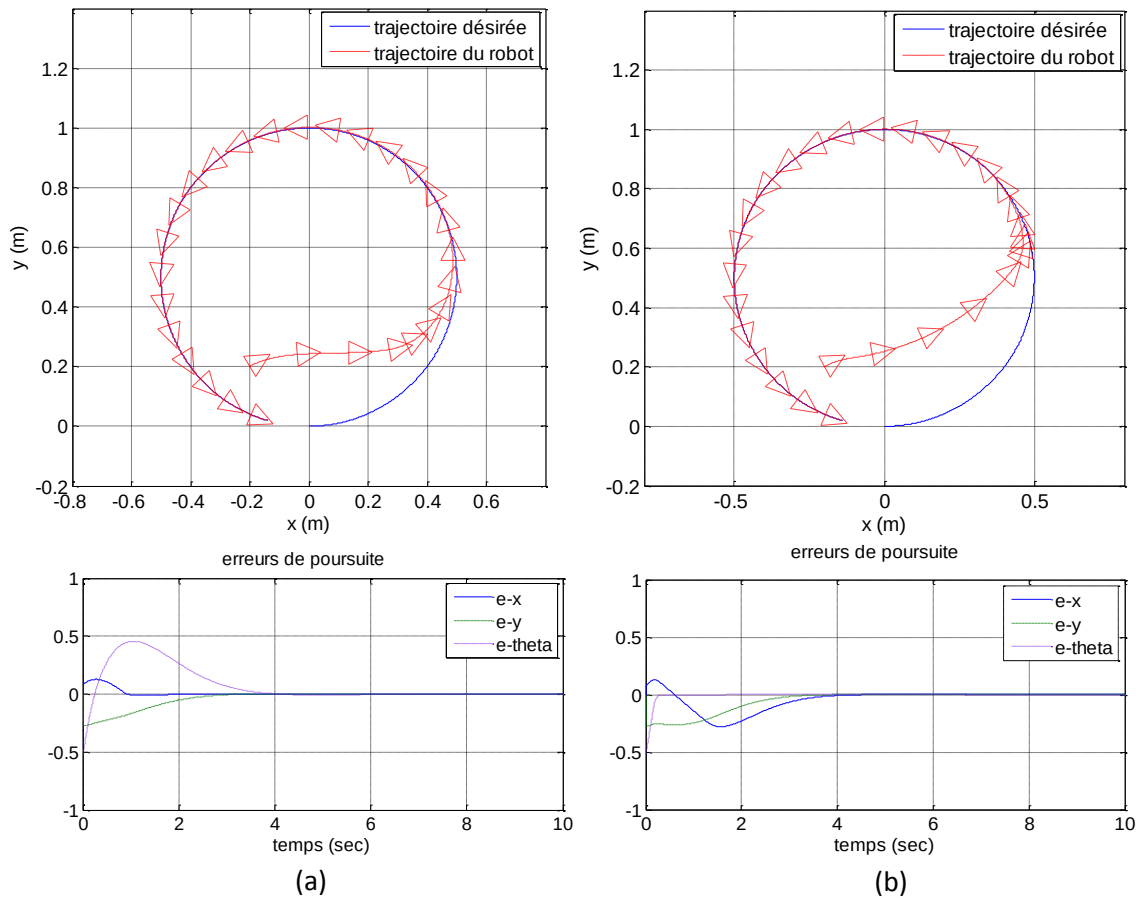


Figure 3.22 : comportement du robot et erreurs de poursuite. (a) commande n°1. (b) commande n°2

Le temps de convergence est sensiblement le même pour les deux lois de commande, et est estimé à 4 sec. Néanmoins, nous avons une meilleure poursuite dans le cas de la commande n°1, avec une erreur de l'ordre de $10^{-5} m$ pour la position et de $4 \times 10^{-4} rad$ pour l'orientation. A noter que les erreurs de poursuite sont décrites dans le repère mobile lié au robot, dont les expressions sont données par (3.39).

On s'intéresse maintenant aux entrées de commande du robot (v, ω, eam, ead), qui sont données dans la figure suivante (3.23) :

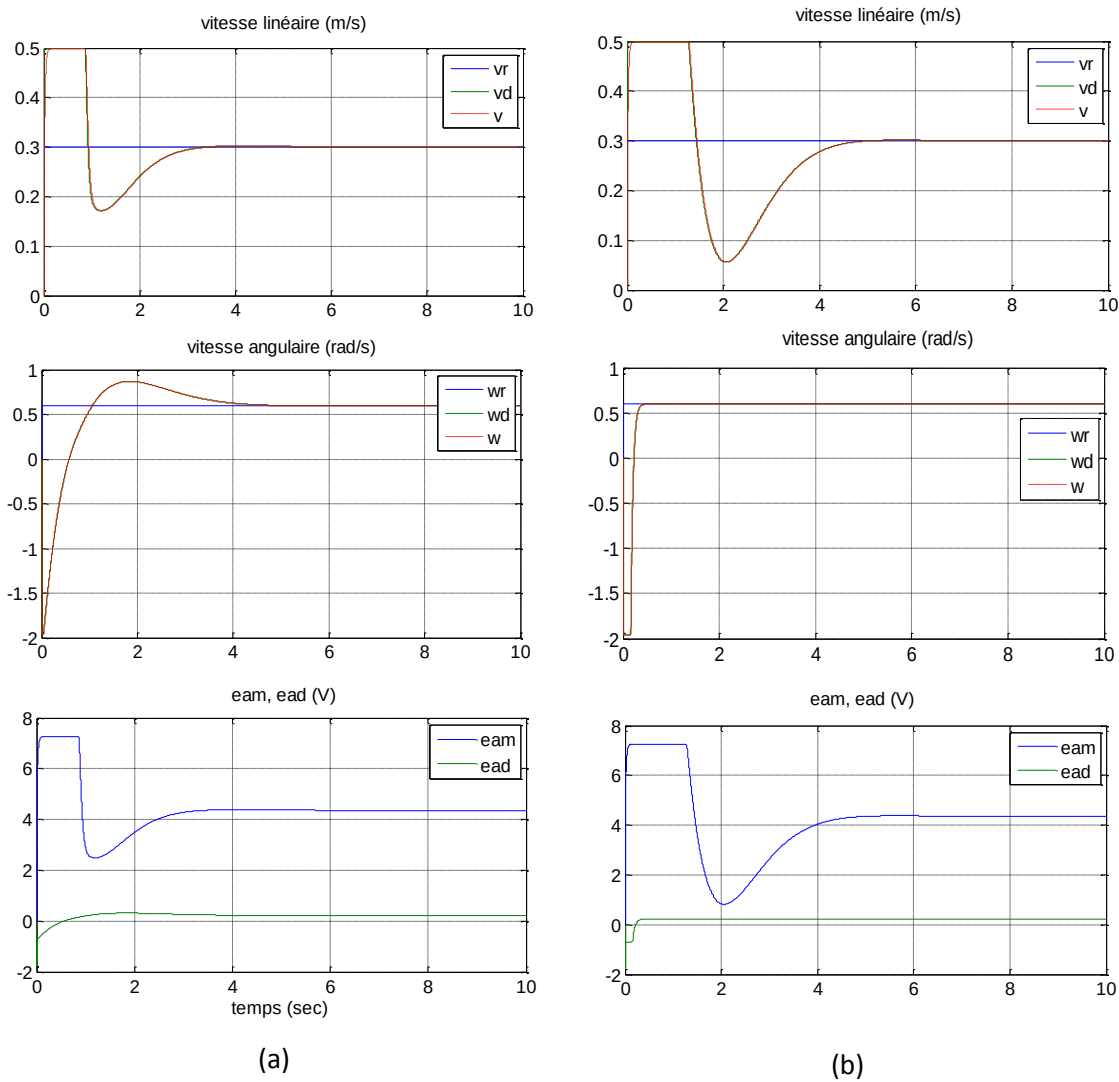


Figure 3.23 : Allure des entrées de commande du robot. (a) commande n°1. (b) commande n°2

On note bien que les variations des différentes entrées de commande sont semblables dans les deux cas à quelques différences près. En effet, pour la commande n°1 la correction de l'erreur d'orientation intervient en dernier lieu, alors que c'est le contraire pour la commande n°2. Ce choix a été pris à cause du fait que dans le cas de la commande n°1, un réglage rapide de l'erreur d'orientation, qui correspond à une augmentation de la valeur de c_3 , empêcherait une

bonne poursuite en position, particulièrement lorsque les erreurs initiales sont importantes. On essaye donc d'abord de ramener le robot dans un voisinage proche de la trajectoire de référence en réduisant l'erreur de position.

On note aussi, que la commande n°1 présente des consignes de vitesses dont les variations sont moins brusques, surtout à l'approche de la trajectoire de référence.

Finalement, nous présentons les signaux électriques appliqués sur les moteurs du robot pour générer les commandes précédentes (figure 3.24) :

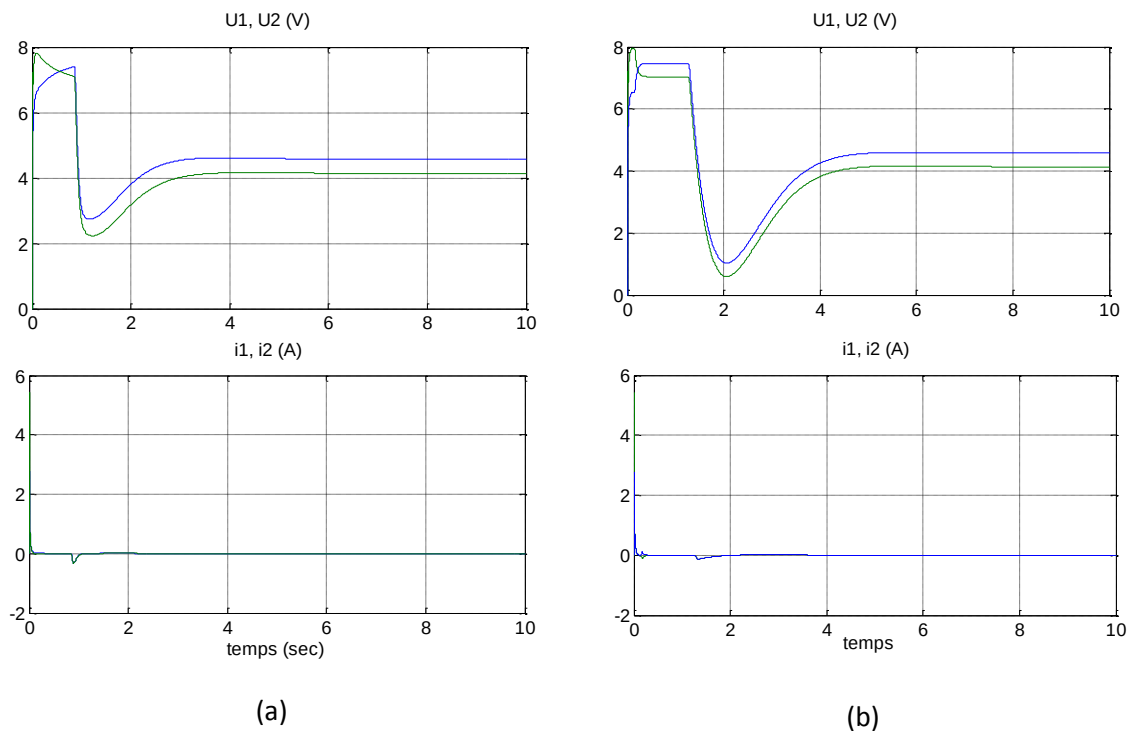


Figure 3.24 : allure des signaux électriques appliqués aux moteurs. (a) commande n°1. (b) commande n°2

3.6.2.2. Evaluation et conclusion partielle

Il apparaît clairement que la commande n°1 basée sur la stabilité au sens de Lyapunov est meilleure, puisque plus précise et moins contraignante par rapport à l'autre technique (permet de suivre n'importe quelle trajectoire). D'autre part, comparativement à la commande basée sur la linéarisation par bouclage, cette technique possède des avantages et des inconvénients. Pour ce qui est des avantages, elle présente des allures pour les entrées de commande qui sont moins brusques. S'agissant des inconvénients, le temps de convergence est deux fois plus grand. Ceci peut être expliqué par le fait que dans la technique de linéarisation par bouclage l'objectif est de converger vers la trajectoire, alors que pour l'autre, l'objectif est de converger vers les profils des vitesses du robot virtuel. Aussi, on a noté une relative difficulté pour le réglage des paramètres de la loi de commande dont l'objectif est moins explicite par rapport à la commande avec linéarisation par bouclage.

3.6.3. Résultats de simulation pour la commande avec observateur de l'orientation du robot

La loi de commande permettant d'obtenir les consignes de vitesse du robot, v_d et ω_d , est donnée par l'équation (3.60) et l'observateur est donné par (3.59). Cette loi est dérivée de la commande n°2 vue précédemment, donc les remarques y afférentes sont aussi valables pour cette loi, notamment l'échec de poursuite de trajectoire en ligne droite.

On rappelle que les variables de commande du modèle dynamique, e_{am} et e_{ad} , sont données par les équations (3.80) et (3.81).

En considérant le cas idéal, c'est-à-dire sans perturbations sur le système ou sur la sortie, on présente les résultats de simulations pour une trajectoire de référence de type circulaire (figure 3.25), avec comme paramètres :

Conditions initiales du robot	Paramètres de la boucle interne	Paramètres de la boucle externe
$x_0 = -0,2 \text{ m}$	$K_{p1} = 10$ $K_{i1} = 0,9$	$a = 100$ $c_2 = 2.8$
$y_0 = 0,2 \text{ m}$	$K_{p2} = 0.8$ $K_{i2} = 0,1$	$c_3 = 6.6$ $\gamma = 0.8$
$\theta_0 = \pi/6 \text{ rad}$		

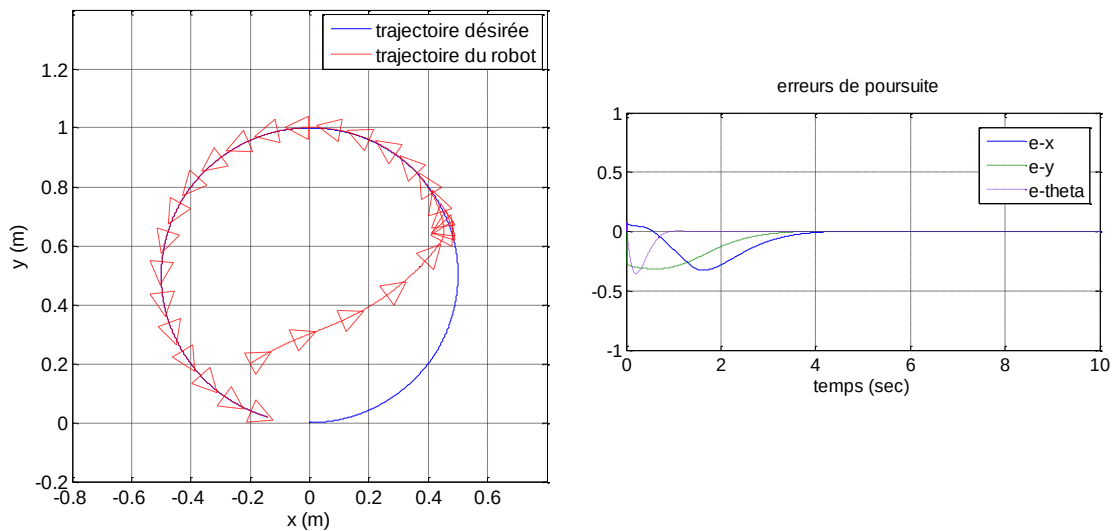


Figure 3.25 : comportement du robot et erreurs de suivi

Quant à l'erreur d'estimation, elle est donnée dans la figure suivante :

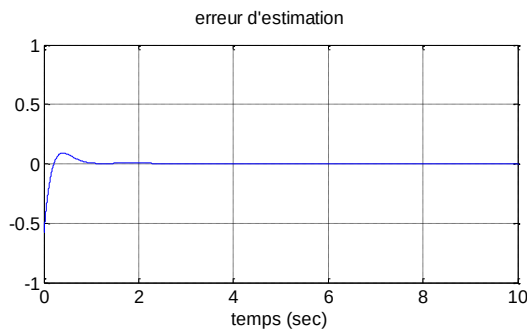


Figure 3.26 : erreur d'estimation

On rappelle que c'est une estimation effectuée sur $\sin \theta$, pour laquelle nous avons obtenu une erreur de l'ordre de 4×10^{-4} .

3.6.4. Résultats de simulation pour la technique de commande par suivi de chemin

La loi de commande permettant d'obtenir la consigne de vitesse angulaire du robot, ω_d , est donnée par l'équation (3.78) sachant (3.76) et (3.66). Quant à la vitesse d'avance on l'a fixée à $0.3m/s$. Pour arrêter le robot, on force la vitesse d'avance à zéro dès que le robot est suffisamment proche de la fin du chemin (on note eps la distance séparant le robot de la fin du chemin), ou après un certain temps écoulé.

En considérant le cas idéal, c'est-à-dire sans perturbations sur le système ou sur la sortie, on montre dans la figure suivante (figure 3.27) que le suivi est assuré pour les trois types de trajectoires de référence choisies :

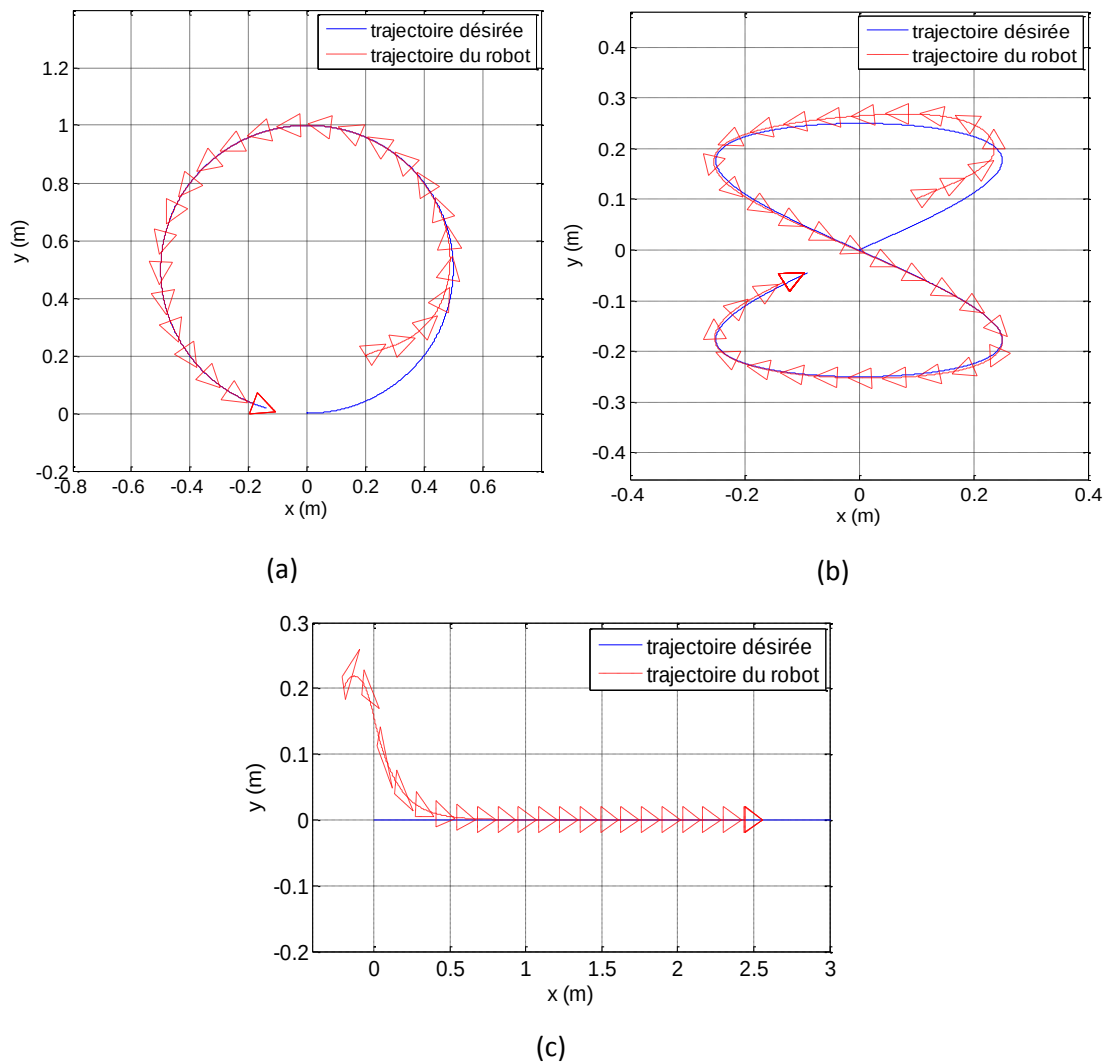


Figure 3.27 : Suivi réussi. (a) trajectoire circulaire. (b) en forme de huit. (c) ligne droite

Les conditions initiales ont été prises comme suit :

$$(a) : x_0 = -0,2 \text{ m}, \quad y_0 = 0,2 \text{ m}, \quad \theta_0 = \pi/6 \text{ rad}$$

$$(b) : x_0 = 0,1 \text{ m}, \quad y_0 = 0,1 \text{ m}, \quad \theta_0 = \pi/6 \text{ rad}$$

$$(c) : x_0 = -0,2 \text{ m}, \quad y_0 = 0,2 \text{ m}, \quad \theta_0 = \pi/6 \text{ rad}$$

Nous remarquons pour le cas du suivi de la trajectoire en forme de huit (figure 3.27.b), le robot met un certain temps pour converger vers la trajectoire. Ceci s'explique par le fait que la limitation qu'on a imposée à la vitesse angulaire pendant la phase de démarrage ne permet pas au robot de suivre la première partie de la trajectoire qui présente une courbure importante.

3.6.4.1. Etude des performances

Cette section est dédiée à l'étude des performances de la technique du suivi de chemin en prenant l'exemple de la trajectoire circulaire comme trajectoire de référence. Les valeurs des différents paramètres mis en jeu ont été choisies comme suit :

Cdts initiales du robot	Param. de la boucle interne	Param. de la boucle externe	Cdts pour arrêter le robot
$x_0 = 0,2 \text{ m}$ $y_0 = 0,2 \text{ m}$ $\theta_0 = \pi/6 \text{ rad}$	$K_{p1} = 10$ $K_{p2} = 0.8$ $K_{i1} = 0,9$ $K_{i2} = 0,1$	$K_1 = 7$ $K_2 = 3.6$	$eps = 5 \text{ cm}$ $t = 9 \text{ sec}$

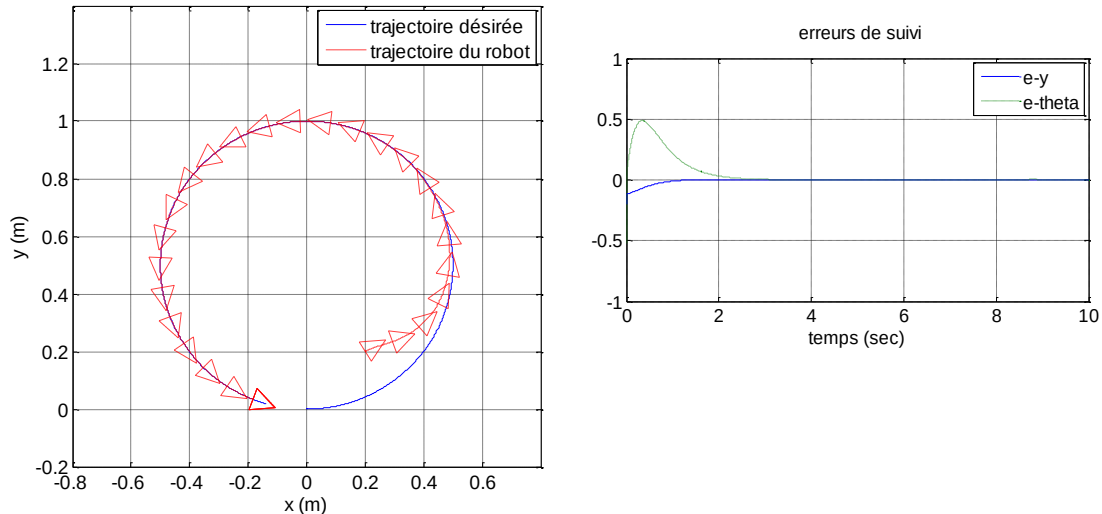


Figure 3.28 : comportement du robot et erreurs de suivi

Le suivi est assuré avec une très bonne précision, qui est de l'ordre de 10^{-4} m pour la position et de 10^{-3} rad pour l'orientation. Le temps de convergence est estimé à 3 sec, mais ne peut être comparé avec les autres techniques, puisqu'ici la vitesse d'avance du robot est imposée.

On présente maintenant aux entrées de commande du robot (v, ω, eam, ead), qui sont données dans la figure suivante (3.29) :

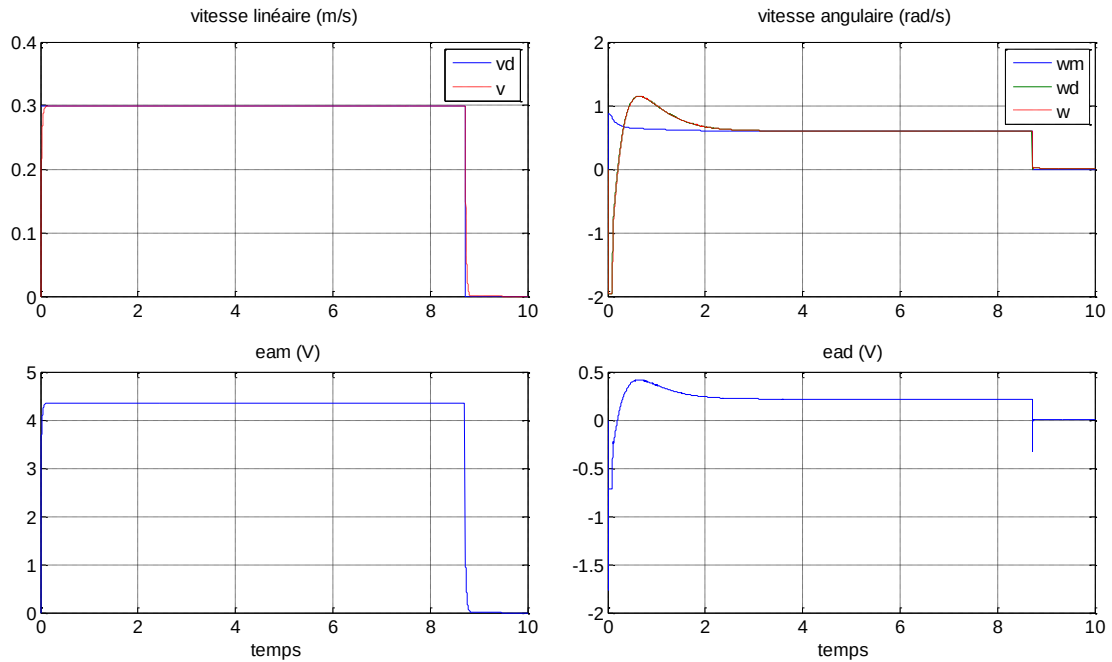


Figure 3.29 : Allure des entrées de commande du robot

Finalement, nous présentons les signaux électriques appliqués sur les moteurs du robot pour générer les commandes précédentes (figure 3.30) :

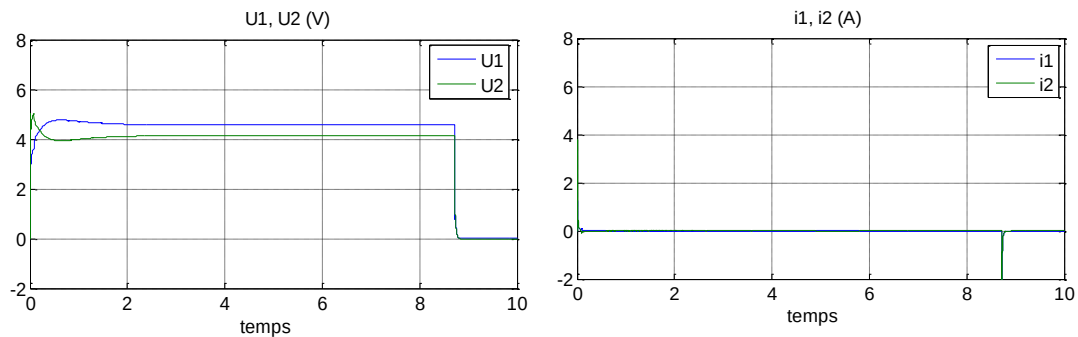


Figure 3.30 : allure des signaux électriques appliqués aux moteurs

Les deux pics de courants correspondent aux instants de démarrage et de freinage du robot.

3.6.4.2. Evaluation et conclusion partielle

La technique de commande par suivi de chemin donne de très bons résultats en matière de précision du suivi. En plus, cette technique permet d'avoir un contrôle permanent sur la vitesse d'avance du robot, qui peut être choisie constante ou répondant à un profil bien précis (par exemple en fonction de la courbure du chemin à suivre).

3.6.5. Choix de la technique la plus adaptée à notre contexte

Etant donné le contexte des applications que nous avons choisies, dans lesquelles le robot est amené à suivre une trajectoire, ou plus précisément une courbe, qui n'est pas connue à priori, et dont l'exécution ne tolère pas de dépassements, le plus sûr serait de choisir la technique de commande par suivi de chemin, où il nous reviendra de fixer la vitesse d'avance du robot, d'autant plus que cette technique offre une bonne précision, comparativement aux autres lois de commande.

3.7. Conclusion

Dans ce chapitre, nous nous sommes intéressés à la problématique de la commande du robot unicycle. Nous avons proposé des techniques de commande appartenant à deux stratégies qui correspondent à nos objectifs, à savoir, la poursuite de trajectoire et le suivi de chemin. Les simulations ont permis de dégager les points forts et les points faibles pour chacune de ces techniques, qui dans l'ensemble permettent toutes d'assurer une bonne précision pour les variables commandées.

Chapitre 4

Introduction à la vision artificielle en robotique mobile

4.1. Introduction

Chez l'être humain, la vision est le sens qui transmet le plus d'informations au cerveau. Elle permet de construire une représentation très riche de notre environnement et ainsi d'entreprendre une interaction intelligente avec cet environnement qu'il soit statique ou dynamique (éviter d'obstacles fixes et mobiles, rendez-vous avec autres agents mobiles, etc.).

De ce point de vue, il est intéressant de constater que les chercheurs en robotique ont rapidement essayé de faire bénéficier les robots de capacités de vision pour accomplir des tâches essentielles telles que la navigation dans un environnement inconnu [4].

Au cours des deux dernières décennies, les innovations technologiques concernant la fabrication de caméras et l'évolution des ordinateurs ont permis d'intégrer des systèmes complexes de vision dans les systèmes embarqués, que ce soit sur des robots mobiles pour la navigation autonome ou sur des véhicules pour l'aide à la conduite [3].

En général, les applications de vision pour la robotique passent par deux principales phases, qui consistent à l'extraction de certaines informations pertinentes de l'image, puis à l'utilisation de ces informations pour une tâche donnée (contrôle, cartographie, etc.). La première phase d'extraction de l'information est plutôt un problème de traitement d'image, tandis que la seconde n'est pas forcément spécifique aux applications de vision, par exemple, la planification d'une trajectoire à exécuter.

A noter, qu'en comparaison aux autres domaines applicatifs de la vision par ordinateur (contrôle industriel, télédétection, analyse médicale, ...), celui de la robotique mobile se heurte à plusieurs difficultés. L'environnement dans lequel le robot évolue est variable, l'éclairage est naturel, le traitement doit être embarqué et le temps de réaction doit être le plus bref possible. Il convient alors de faire un choix judicieux du contrôleur permettant de satisfaire les objectifs de la tâche à réaliser [39].

Dans ce chapitre, on se propose de présenter quelques notions liées à la vision artificielle et son utilisation dans le domaine de la robotique mobile, notamment celles entrant dans le cadre de la perception et de la modélisation de l'environnement (calibrage et modélisation de caméra, filtrage d'images, détection de contours, etc.), ainsi que quelques techniques de planification de trajectoire basée sur l'évitement réactif d'obstacles, utilisée dans notre application.

4.2. Détection et modélisation de l'environnement par vision

4.2.1. Rappels sur la notion d'image

Une image qu'on note $a(x, y)$ est obtenue à partir d'un signal continu bidimensionnel issu par exemple d'un appareil photo ou d'une caméra.

4.2.1.1. Image numérique

Une image numérique $a[m,n]$ décrite dans un espace discret 2D, est obtenue à partir d'une image analogique $a(x,y)$ par un processus d'échantillonnage, appelé numérisation.

Par ce processus, l'image analogique $a(x,y)$ est divisée en N lignes et M colonnes, où l'intersection entre une ligne et une colonne représente ce qu'on appelle un pixel.

Une image numérique est donc un signal fini bidimensionnel échantillonné à valeurs quantifiées dans un certain espace de couleurs, où on peut distinguer :

- Image en niveaux de gris :qui autorise un dégradé de gris entre le noir et le blanc pour chaque pixel de l'image. En général, on code le niveau de gris sur un octet (8 bits) soit 256 nuances de dégradé ;
- Image binaire :chaque pixel de l'image peut prendre uniquement la valeur 0 (noir) ou 1 (blanc). Le niveau de gris est donc codé sur un seul bit ;
- Image RVB : il s'agit d'une image couleur formée par la composition de trois images en niveaux de gris sur trois composantes. On définit donc trois plans de niveaux de gris, un rouge, un vert et un bleu. La couleur finale est obtenue par synthèse additive des ces trois composantes.

4.2.2. Capteurs pour la vision

Les systèmes de vision utilisés en robotique, et dans la vision par ordinateur en général, reposent tous sur l'utilisation d'un capteur de vision, qui peut être un appareil photo, mais dans la plupart des cas il s'agit d'une caméra. Cette dernière permet de fournir une représentation bidimensionnelle, l'image, de l'espace tridimensionnel qui représente la scène observée.

Dans certaines applications, une seule caméra ne suffit pas, et on recourt à l'utilisation d'un système dit de stéréovision qui consiste à observer une même scène à l'aide d'au moins deux caméras légèrement décalées l'une de l'autre, afin d'extraire une information de distance. Néanmoins, ces systèmes sont plus chers et plus complexes à mettre en place, car ils nécessitent des caméras bien calibrées et précisément positionnées.

4.2.2.1. Types de caméras

L'acquisition des images peut se faire à l'aide de plusieurs types de caméras. Etant donné que l'ère de l'analogique est révolue, la quasi-totalité des systèmes de vision artificielle d'aujourd'hui sont basés sur l'utilisation de caméras numériques.

Bien qu'étant de conception complexe, la composition d'une caméra numérique peut être simplifiée autour des composants suivants :

- Une lentille (système optique de la caméra) qui sert à diriger les rayons lumineux provenant de la scène vers le centre optique de la caméra par des techniques de diffraction, de réfraction et de réflexion;

- Un capteur placé à côté du centre optique de la caméra, qui mesure l'intensité de la lumière sur sa surface et la stocke dans des registres mémoires. Ce capteur est composé d'une matrice de petites cellules photosensibles permettant de transformer la lumière en énergie électrique ;
- Un dispositif électronique (système de capture), qui permet de construire les images numériques et de les acheminer vers l'unité de calcul.

Les types de caméras numériques les plus répandus sont basés sur des capteurs de type CCD ou CMOS, tous deux reposant sur le principe de l'effet photoélectrique des semi-conducteurs. A noter qu'il existe d'autres types de caméras plus évoluées, tel que la technologie X3, qui utilise trois photorécepteurs sur chaque cellule de la matrice du capteur au lieu d'une seule, chacune d'elles étant responsable de la capture d'une couleur r-v-b (figure 4.1).

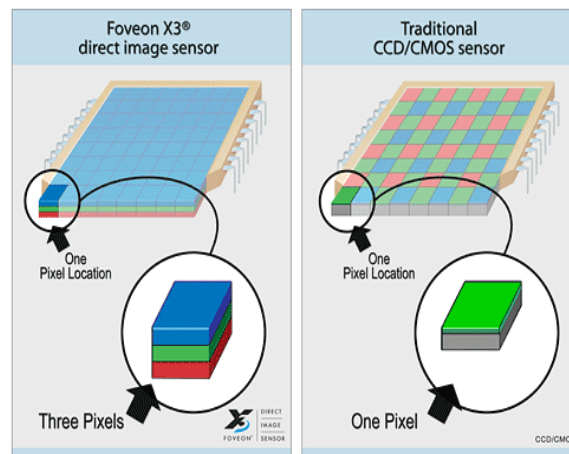


Figure 4.1 : Technologies utilisées dans les caméras numériques

Les caméras CCD souffrent, par rapport aux CMOS, d'un problème dit de débordement (overflow), où la saturation d'un pixel entraîne une altération de couleur des pixels voisins. Cependant, grâce à leur faible coût et une qualité acceptable même en milieu professionnel, les caméras CCD restent de loin les plus utilisées.

4.2.2.2. Modélisation d'une caméra [28]

Dans cette section, nous allons présenter le modèle mathématique le plus utilisé pour modéliser le processus de formation d'image à partir d'une caméra. Il s'agit bien d'un modèle simplifié, mais qui reste acceptable pour la majeure partie des applications, avec l'avantage d'un calcul plus facile du modèle inverse, nécessaire pour le calcul de la commande.

4.2.2.2.1. Présentation du modèle sténopé

La situation d'une caméra est définie par la position de son centre optique C et l'orientation de son axe optique.

Sur la figure 4.2, le centre optique de la caméra C est lié à l'origine O_C du repère caméra R_C et l'axe optique est confondu avec l'axe Z_C de ce repère. Physiquement, le plan image UV est un plan perpendiculaire à l'axe optique localisé à une distance f du centre optique, ou f est la distance focale. Le point C_i est appelé le point principal et correspond au point où l'axe optique traverse le plan image.

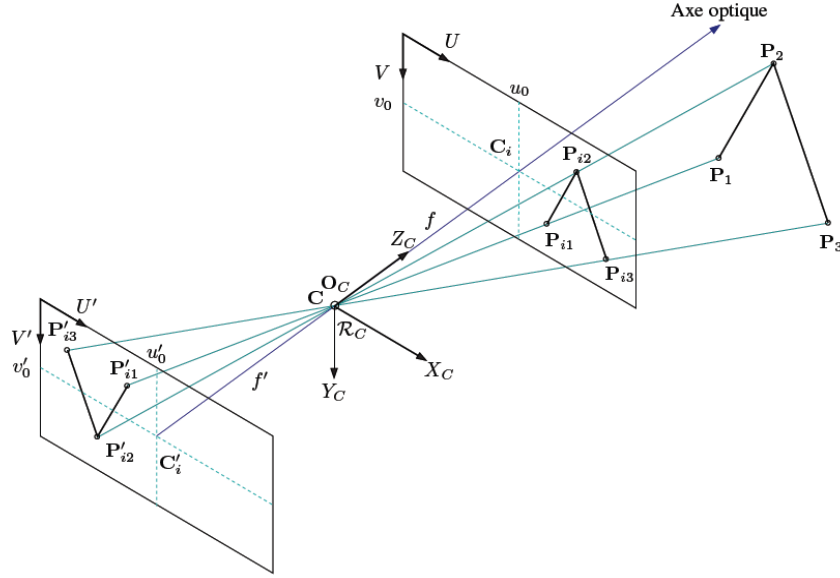


Figure 4.2 : Modèle Sténopé d'une caméra, avec le plan image réel UV et le plan image virtuel UV' .

Le modèle sténopé (pinhole en anglais) suppose que tous les rayons passent par le centre optique C .

Considérons un objet défini dans le repère caméra par trois points P_1, P_2 et P_3 . Leurs projections dans le plan image produisent les points P_{i1}, P_{i2} et P_{i3} . Ce qui produit une image inversée de l'objet.

Pour faciliter l'analyse des images, nous utilisons un plan image « virtuel » UV' situé à une distance focale f en avant du centre optique C . La projection des points de l'objet sur ce plan virtuel produit les points P_{i1}, P_{i2} et P_{i3} qui génèrent une image équivalente ayant la propriété d'être dans le même sens que l'objet.

4.2.2.2.2. Matrice de projection perspective

Nous nous intéressons à la projection et aux propriétés géométriques du système. Le point P_1 est défini par les coordonnées :

$$P_1 = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_C$$

Les coordonnées du point P_{i1} correspondent à la projection sur le plan image UV du point P_1 , elles s'expriment dans le repère caméra par :

$$P_{i1} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}_C$$

où x_i, y_i et z_i représentent les coordonnées dans le plan image :

$$x_i = f \frac{x}{z} y_i = f \frac{y}{z} z_i = f \quad (4.1)$$

En utilisant la représentation par des coordonnées homogènes, le système s'écrit :

$$s \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}_C = \begin{bmatrix} s x_i \\ s y_i \\ s \end{bmatrix}_C = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_C \quad (4.2)$$

où s représente la profondeur du point dans l'image.

La matrice de projection P est alors définie par :

$$P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.3)$$

Cette matrice peut être décomposée en un produit

$$P = K_p P_0 \quad (4.4)$$

où :

$$K_p \triangleq \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & f \end{bmatrix} P_0 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.5)$$

4.2.2.2.3. Transformation affine caméra/image

Considérons la figure 4.3. Les points image sont mesurés en pixels dans le repère UV associé au plan image. Le point principal C_i exprimé en coordonnées images est donné par (u_0, v_0) . Les pixels d'une caméra sont rarement carrés, soit λ_u la largeur et λ_v la hauteur d'un pixel.

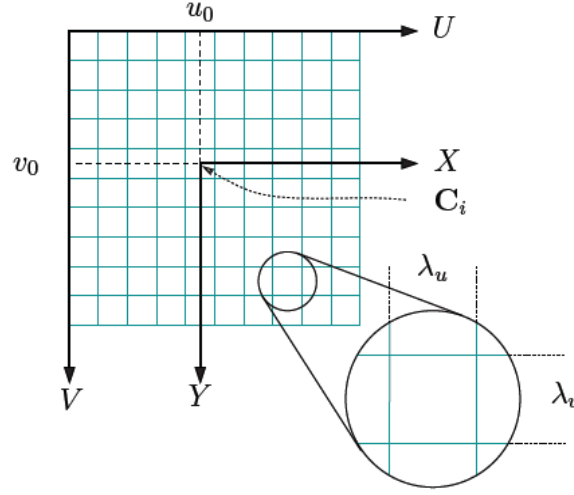


Figure 4.3 : Image dans le repère UV

Les coordonnées du point de l'image mesurées en pixel sont données par :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} K_u & 0 & u_0 \\ 0 & K_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \quad (4.6)$$

où :

$$K_u = \frac{1}{\lambda_u} K_v = \frac{1}{\lambda_v} \quad (4.7)$$

La matrice K_A définit une transformation affine :

$$K_A \triangleq \begin{bmatrix} K_u & 0 & u_0 \\ 0 & K_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

4.2.2.2.4. Paramètres intrinsèques

En faisant une projection perspective suivie d'une transformation affine, on aboutit à :

$$u = K_u f \frac{x}{z} + u_0 \quad v = K_v f \frac{y}{z} + v_0 \quad (4.9)$$

en multipliant les matrices $K_A K_p P_0$, on abouti à une représentation par matrice homogène :

$$\begin{bmatrix} s_u \\ s_v \\ s \end{bmatrix}_C = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_C \quad (4.10)$$

où :

$$\alpha_u = fK_u\alpha_v = fK_v$$

Ce modèle de la caméra sténopé ne comporte que des paramètres intrinsèques $(\alpha_u, \alpha_v, u_0, v_0)$ et suppose :

- un montage mécanique parfait, tel que la matrice CCD du capteur soit orthogonale à l'axe optique ;
- des lentilles et les axes XC et YC soient alignées aux lignes et colonnes de la matrice CCD ;
- les pixels ont une forme rectangulaire ;
- les lentilles ne présentent pas de distorsion.

Pour l'estimation de ces paramètres, il est nécessaire de passer par un calibrage de la caméra.

4.2.2.2.5. Paramètres extrinsèques

Lorsqu'on utilise la caméra comme capteur, les mesures que l'on peut obtenir sont exprimées dans le repère caméra. Pour avoir un modèle complet de la caméra, il faut prendre en compte un repère de référence, par exemple le repère monde R_W (figure 4.4). Une transformation qui exprime le changement de repère de la caméra au monde ${}^C T_W$ définit les paramètres extrinsèques. En utilisant la transformation inverse ${}^W T_C$, le modèle complet de la caméra devient :

$$\begin{bmatrix} s_u \\ s_v \\ s \end{bmatrix}_C = \begin{bmatrix} \alpha_u & 0 & u_0 & 0 \\ 0 & \alpha_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^W A_C & {}^W P_C \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_W \quad (4.11)$$

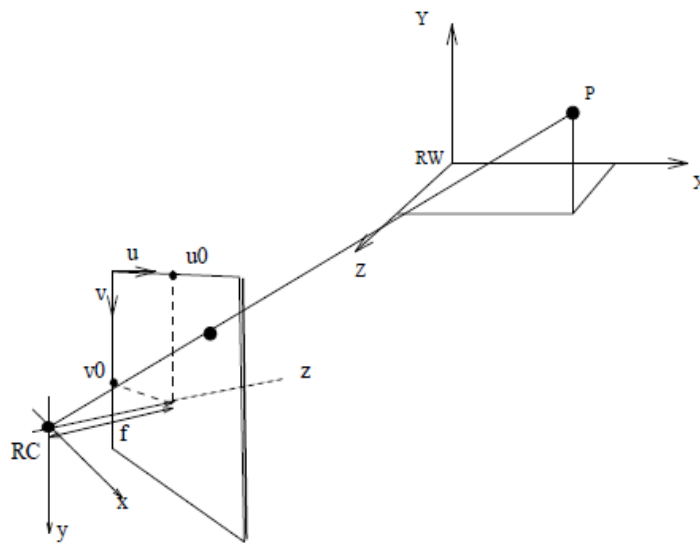


Figure 4.4 : Modèle sténopé et système de coordonnées

4.2.2.3. Calibrage d'une caméra

Le calibrage d'une caméra, ou d'un système de vision en général, consiste à déterminer la relation mathématique existant entre les coordonnées tridimensionnelles des points d'une scène et les coordonnées bidimensionnelles de ces mêmes points projetés, détectés dans une image. Ce qui revient à calculer les paramètres intrinsèques et extrinsèques de la caméra.

La détermination de cette relation est une étape incontournable en vision, en particulier pour la reconstruction, où il est alors nécessaire d'inférer une information tridimensionnelle à partir de données extraites de l'image 2D.

En général, un système permettant de calibrer une caméra est composé par :

- une mire de calibrage (grille ou objet étalon) généralement constituée de points de référence dont on connaît parfaitement les coordonnées 3D dans un repère propre à celle-ci ; un système d'acquisition d'images pour la numérisation et la mémorisation des images de la mire ;
- un algorithme de mise en correspondance des points 2D détectés dans les images avec leurs homologues de la mire ;
- Un algorithme de calcul de la matrice de transformation perspective de la caméra, liant le repère associé à la mire de calibrage à celui de l'image.

A noter que le calibrage est un processus statique qui s'effectue hors-ligne, avant l'utilisation de la caméra pour une tâche visuelle précise. Une fois la caméra calibrée, ses paramètres doivent rester fixes en cours d'utilisation. Chaque fois que l'on désire modifier la mise au point, la distance focale voire l'ouverture de l'objectif, la caméra devra être recalibrée.

S'agissant des méthodes de calibrage, les plus connues utilisent le coin de Moravec, qui repose sur l'utilisation d'une mire composée de carrés noirs sur fond blanc afin d'obtenir le meilleur gradient possible.

On trouve également une autre technique de calibration qui utilise la correspondance entre des ellipses caractérisées par leurs moments géométriques, ou encore la technique qui permet d'extraire les paramètres intrinsèques par rotation de la caméra et en effectuant une recherche d'éléments géométriquement invariants.

4.2.3. Les techniques de traitement d'images en robotique mobile

Les techniques de traitement d'images utilisées en robotique mobile peuvent être classées en trois catégories, selon le niveau d'abstraction :

- Les processus de bas-niveau, qui nécessitent très peu d'informations sur le contenu des images. Il s'agit de processus de filtrage, d'amélioration d'images et de détection de contours, processus qu'on peut regrouper sous l'appellation : prétraitement d'images et extraction d'indices ;

- Les processus de niveau intermédiaire, qui englobent les techniques de segmentation d'images en régions ;
- Les processus de haut-niveau, qui peuvent nécessiter des informations sur le contenu des images, et dans lesquels on trouve, la reconstruction tridimensionnelle (modélisation de l'environnement), la reconnaissance de formes, l'analyse de mouvement, etc.

Dans le cadre de ce mémoire, la vision artificielle est utilisée pour des fins de planification de trajectoire (voir approche proposée, chap.1). Ainsi, pour l'application impliquant le suivi d'une ligne tracée sur le sol, il s'agit de détecter et d'extraire les contours correspondants à la ligne à suivre. Quant à l'application de navigation avec évitement d'obstacles, il s'agit plutôt de créer une représentation de l'environnement (position des obstacles), en localisant les segments d'image correspondants aux obstacles. A cet effet, dans cette section, l'intérêt est orienté essentiellement vers les techniques de bas-niveau qui permettent de répondre en grande partie à notre problématique, et qui sont donc présentées dans le détail. Les autres catégories sont quant à elles sommairement introduites.

4.2.3.1. Les techniques de bas-niveau

4.2.3.1.1. Lissage local

Le lissage local (ou filtrage) consiste à supprimer le bruit présent dans une image en étudiant, pour chaque pixel, les valeurs d'intensité sur son voisinage.

Afin de procéder au lissage, il convient tout d'abord de définir un produit de convolution. En une dimension, on a :

$$(f * g)(n) = \sum_{i=0}^n f(i) \cdot g(n - i) \quad (4.11)$$

où : f et g sont deux fonctions.

En généralisant, on obtient, en deux dimensions :

$$(f * g)(n, m) = \sum_{i=0}^n \sum_{j=0}^m f(i, j) \cdot g(n - i, m - j) \quad (4.12)$$

4.2.3.1.1.1. Moyennage

Une première classe d'approche est basée la redondance d'informations. La nouvelle valeur d'un pixel est calculée par moyennage des valeurs sur un voisinage.

Cette opération linéaire peut être vue comme la convolution discrète de l'image par un masque :

$$I(i, j) = \sum_{(m,n) \in V} h(m, n) \cdot I(i - m, j - n) \quad (4.13)$$

$$\sum_{(m,n) \in V} h(m, n) = 1 \quad (4.14)$$

Où I est l'intensité de l'image d'origine, I' est l'intensité de l'image filtrée, V est le voisinage utilisé et h est le masque de convolution.

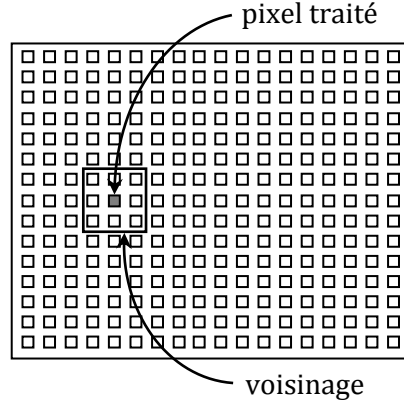


Figure 4.5 : Application d'un filtre sur un voisinage 3×3

Par exemple, pour le moyennage sur un voisinage 3×3 on utilise le masque suivant :

$$h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.15)$$

Bien qu'il permette d'éliminer les dégradations locales de faibles dimensions (dimensions des objets dans l'image supérieures à celles des dégradations), le moyennage a tendance à rendre l'image floue (dégradation de contours).

Une amélioration du filtre moyenne consiste à jouer sur les valeurs des coefficients du masque :

$$h_1 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.16)$$

C'est le cas des filtres binomiaux pour lesquels les valeurs des coefficients sont générées par le triangle de Pascal :

$$h_1 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (4.17)$$

4.2.3.1.1.2. Filtres médians

Les filtres de moyennage ont tendance à rendre l'image floue et donc à perdre de l'information sur les contours caractérisés par des fortes variations d'intensité.

Pour diminuer cet effet, on ne moyenne plus sur le voisinage mais on prend la valeur médiane sur ce voisinage. C'est le filtre médian.

Exemple pour un voisinage 3×3 :

$$I = \begin{bmatrix} 2 & 12 & 12 \\ 2 & 12 & 60 \\ 2 & 2 & 12 \end{bmatrix} \quad (4.18)$$

Le filtre médian est un filtre non linéaire qu'on peut appliquer de manière itérative. Il permet d'éliminer le bruit impulsionnel, tout en préservant les contours.

4.2.3.1.1.3. Autres filtres

Dans l'objectif d'améliorer les caractéristiques du filtre et en particulier la préservation des contours dans l'image, on peut utiliser des filtres plus évolués.

Il en existe plusieurs. Nous donnons ici l'exemple d'un filtre faisant intervenir des variations d'intensité.

Soit $d(i, j, k, l)$ la variation d'intensité entre le pixel (i, j) et (k, l) :

$$d(i, j, k, l) = \begin{cases} |I(i, j) - I(k, l)| & \text{si } I(i, j) \neq I(k, l) \\ 1/2 & \text{sinon} \end{cases} \quad (4.19)$$

Alors les coefficients du masque de convolution sont déterminés par :

$$h(m, n) = \frac{1/d(i, j, i + m, j + n)}{\sum_{(m, n) \in V} 1/d(i, j, i + m, j + n)} \quad (4.20)$$

Avec ce type de filtre :

- la distance est plus importante sur un contour que dans une région homogène ;
- les points fortement bruités ont peu d'influence ($h(m, n)$ petit) ;
- si le pixel est sur un contour, les pixels voisins de même région auront un poids important alors que les pixels voisins de région différente auront peu de poids. Le contour est de cette manière préservé.

4.2.3.1.2. Amélioration d'images

L'amélioration d'images consiste à modifier les caractéristiques visuelles de l'image de manière à en faciliter son interprétation. Il peut s'agir de rehausser les contrastes, d'accentuer

certaines intensités pour mettre en valeur une région, etc. Les histogrammes sont fréquemment utilisés pour effectuer ce type d'opérations.

4.2.3.1.2.1. Modification d'histogramme

L'histogramme d'une image est la fonction qui associe à une valeur d'intensité i le nombre de pixels dans l'image ayant cette valeur.

Pour modifier les caractéristiques de l'image (accentuer les contrastes en général), une approche générale consiste à modifier son histogramme par application d'une fonction qui associe à chaque valeur d'intensité dans l'image une nouvelle valeur.

Soit i , $i < MaxInt$ les valeurs d'intensité de l'image traitée, on considère alors les transformations du type :

$$i = T(i) \quad (4.21)$$

qui donnent une nouvelle valeur d'intensité i pour chaque valeur i de l'image. On suppose que la fonction T est telle que :

- $T(i)$ est monotone (souvent croissante) sur l'intervalle. Cette condition assurant que l'ordre des intensités est préservé après transformation ;
- $0 \leq T(i) \leq MaxInt$ pour $0 \leq i \leq MaxInt$ qui garantit que la nouvelle image est cohérente avec les niveaux d'intensité autorisés ;
- La transformation inverse satisfait les deux conditions précédentes.

4.2.3.1.2.1.1. Transformations linéaires

Les transformations linéaires d'histogrammes sont nombreuses et variées. Elles permettent d'accentuer une zone d'intensité ou de modifier la répartition des valeurs d'intensité. Un exemple de transformation linéaire est le recadrage.

Il consiste à modifier l'intervalle des valeurs d'intensité (la dynamique) de façon à obtenir pour l'image améliorée un intervalle de valeurs maximal.

Si $[i_1, i_2]$ est l'intervalle de l'image traitée, alors la transformation équivalente s'écrit simplement :

$$T(i) = MaxInt (i - i_1)/(i_2 - i_1) \quad (4.22)$$

4.2.3.1.2.1.2. Transformations non linéaires

Une modification d'histogramme très répandue pour augmenter le contraste de manière automatique est la linéarisation d'histogramme. Le principe est de transformer l'image de manière à obtenir un histogramme plat, soit une distribution uniforme des intensités. Cela revient à maximiser l'entropie de l'image et donc à obtenir théoriquement une image présentant une information maximale.

Pour une image majoritairement claire la linéarisation a tendance à augmenter la dynamique de la partie sombre de l'histogramme au détriment de la partie claire.

4.2.3.1.3. Détection des contours

4.2.3.1.3.1. Approches classiques

Ces approches ont grandement contribué à l'évolution du traitement d'image, mais sont de moins en moins utilisées, à l'avantage d'autres méthodes plus récentes et plus performantes. Parmi ces approches on retrouve :

4.2.3.1.3.1.1. Détecteurs de gradient par filtrage

Ces détecteurs reposent tous sur une recherche d'un extremum de la dérivée première (ou d'un passage par zéro d'une dérivée seconde), celle-ci étant calculée de diverses manières, mais généralement par un filtrage passe haut précédé d'un léger filtrage passe-bas pour s'affranchir des bruits. La somme des connaissances sur ce sujet se trouve dans [71].

Pour illustrer cette méthode, on peut citer l'exemple de filtrage de Wiener [53] basé sur l'utilisation d'un filtre récursif de la forme :

$$h_0(z_1, z_2) = \frac{\sum_i^n \sum_j^n b_{i,j} z_1^{-i} z_2^{-j}}{1 + \sum_i^n \sum_j^n a_{i,j} z_1^{-i} z_2^{-j}} \quad (4.23)$$

où les coefficients $a_{i,j}$ et $b_{i,j}$ sont déterminés à partir de 3 paramètres spécifiques de l'image à traiter : λ , ρ et ϵ . λ est le paramètre poissonien de densité de contours (il exprime la distance moyenne entre 2 contours lors d'un balayage de l'image), ρ la constante de la loi exponentielle de l'auto-corrélation de l'image et ϵ la puissance du bruit. Comme un tel filtre ne filtre que du coin supérieur gauche au coin inférieur droit, on lui associe les 3 autres filtres obtenus par rotation de $\pi/2$, pour obtenir un filtre final. Dans la pratique pour assurer des temps de calcul raisonnables, on choisit des filtres d'ordre 1.

4.2.3.1.3.1.2. Détecteurs de gradient par masques

Reposent sur l'utilisation de filtres de dérivation à partir d'estimateurs locaux de l'image ou de ses dérivées. Ces estimées sont obtenues à l'aide de masques appliqués sur des fenêtres 2×2 ou 3×3 pixels, dont les coefficients sont antisymétriques et la somme nulle.

Les filtres les plus utilisés sont :

<table><tr><td>1</td><td></td></tr><tr><td>-1</td><td></td></tr></table>	1		-1		<table><tr><td>1</td><td></td></tr><tr><td></td><td>-1</td></tr></table>	1			-1	<table><tr><td>1</td><td></td><td>-1</td></tr><tr><td>1</td><td></td><td>-1</td></tr><tr><td>1</td><td></td><td>-1</td></tr></table>	1		-1	1		-1	1		-1	<table><tr><td>1</td><td></td><td>-1</td></tr><tr><td>2</td><td></td><td>-2</td></tr><tr><td>1</td><td></td><td>-1</td></tr></table>	1		-1	2		-2	1		-1
1																													
-1																													
1																													
	-1																												
1		-1																											
1		-1																											
1		-1																											
1		-1																											
2		-2																											
1		-1																											
gradient	Roberts	Prewitt	Sobel																										

Figure 4.6 : Exemples de filtres par détection de gradient

Les filtres 3×3 sont un peu moins précis (c'est-à-dire que les contours qu'ils détectent sont moins bien localisés et souvent épais), mais les images ainsi obtenues sont généralement plus fiables et permettent des post-traitements plus poussés, ils sont également centrés sur un pixel et non entre des pixels (filtres à phase nulle).

L'application de ces filtres s'effectue comme suit : le filtre est centré en chaque pixel successivement (au coin supérieur gauche pour les filtres 2×2 , au point central pour les filtres 3×3). Le produit du masque par les valeurs des pixels correspondant étant fait, la valeur absolue de la somme est retenue. Puis le masque est tourné de $\pi/2$ autour de son centre et la même mesure est répétée. Les deux mesures sont alors ajoutées et leur somme constitue la mesure du gradient en ce point. On peut également mesurer l'orientation ϕ du contour au point donné en faisant le rapport des réponses des deux filtres.

L'avantage avec ces filtres, c'est qu'ils sont particulièrement simples à mettre en œuvre, et très peu gourmands en temps de calcul.

4.2.3.1.3.1.3. Détecteurs du passage par zéro du Laplacien

Ces méthodes exploitent le fait que le passage par zéro du laplacien correspond au maximum du gradient dans la direction du gradient, et permet de mettre en évidence les extremums de la dérivée.

Ces méthodes tirent en outre profit du fait que les zéros de la dérivée seconde constituent un réseau de lignes fermées. Il en est de même du réseau des lignes de crête du gradient, mais le premier est plus aisément détecté à partir d'un simple étiquetage des zones positives et négatives. Mais l'estimation de la dérivée seconde étant très sensible aux bruits, il convient de filtrer l'image avant d'en mesurer le laplacien. Cela conduit au filtrage suivant :

$$\psi * \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \right) = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) [\psi * f] \quad (4.24)$$

où $f(x, y)$ est une image supposée continue, et ψ un filtre passe-bas.

Cela exprime qu'une image de contours est obtenue par filtrage de l'image par la dérivée seconde d'un filtre passe-bas, puis détection des zéros de la fonction ainsi obtenue.

Les filtres les plus utilisés pour ces filtrages passe-bas sont les filtres gaussiens [50]. Le filtre obtenu par convolution avec le laplacien d'une gaussienne est connu sous le nom de LOG. Un autre filtre très utilisé et très proche du LOG est le DOG (*Difference of Gaussians*). Les gaussiennes sont bien sûr affectées des facteurs adéquats pour que la somme des parties positives du filtre soit égale à la somme des parties négatives (valeur nulle à la fréquence 0 de la fonction de transfert). Sa mise en œuvre peut tirer profit des diverses qualités de la gaussienne : séparabilité, noyau autoreproducteur, limite de suites de polynômes, approximation par des splines, etc., ce qui permet d'en accélérer la réalisation.

4.2.3.1.3.1.4. Détecteurs par masques adaptés

Dans ces techniques, on recherche en chaque point de l'image la présence d'une configuration conforme à un gabarit appartenant à un dictionnaire de contours. Pour cela, on définit une distance entre un contour type et une fenêtre de l'image.

Les filtres adaptés les plus utilisés sont le filtre de Kirsch, le filtre de Nevatia et Babu et les boussoles directionnelles de Prewitt.

4.2.3.1.3.2. Approches analytiques

Ces approches sont apparues dans le milieu des années 80 avec les travaux de Torre [71], Shen et Castan [69], Canny [11]. Elles ont permis une meilleure compréhension des conditions d'une bonne détection des contours, ce qui conduit à des détecteurs de bonne qualité.

4.2.3.1.3.2.1. Filtre de Canny

Le filtre de Canny repose sur 3 critères :

- garantir une bonne détection, c'est-à-dire une réponse forte même à de faibles contours ;
- garantir une bonne localisation ;
- assurer que pour un contour il n'y aura qu'une seule détection (éviter les effets de rebonds dus, par exemple, à la troncature des filtres).

Ces 3 critères s'expriment par l'optimisation conjointe de 3 fonctionnelles qui permettent de définir le filtre linéaire optimal pour la détection d'une marche d'escalier sous l'hypothèse d'un bruit additif indépendant du signal.

Si l'on considère que le filtre a pour réponse impulsionnelle $\psi(x)$, ces fonctionnelles s'écrivent :

$$\Sigma = \frac{\int_0 \psi(x)dx}{\sqrt{\int_- \psi^2(x)dx}} \quad (4.25)$$

$$\Lambda = \frac{|\psi(0)|}{\sqrt{\int_{-} \psi^2(x) dx}} \quad (4.26)$$

$$\frac{|\psi(0)|}{\sqrt{\int_{-} \psi^2(x) dx}} = k \frac{\int_{-}^0 \psi(x) dx}{\sqrt{\int_{-} \psi^2(x) dx}} \quad (4.27)$$

La résolution du système est assez complexe (on maximise le produit $\Sigma\Lambda$ sous la contrainte du 3^{ème} terme constant). Par ailleurs, Canny souhaite obtenir un filtre monodimensionnel (dans la direction orthogonale au contour), à réponse impulsionnelle finie (RIF) ; celui-ci a alors une expression complexe, composée de 4 termes, chacun combinant des lignes exponentielles et trigonométriques :

$$\psi(x) = a_1 e^{x/\sigma} \sin \omega x + a_2 e^{x/\sigma} \cos \omega x + a_3 e^{-x/\sigma} \sin \omega x + a_4 e^{-x/\sigma} \cos \omega x$$

Où les coefficients a_i et ω sont déterminés à partir de la taille du filtre. Le paramètre σ est un paramètre d'échelle qui indique en-deçà de quelle distance deux contours parallèles seront confondus en un seul. Canny montre que la dérivée d'une gaussienne est une bonne approximation de son filtre.

$$\psi(x) = -x \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad (4.28)$$

C'est donc la réduction à 1 dimension du filtre proposé par Marr à 2 dimensions (le maximum de la dérivée de la gaussienne est obtenu pour le passage par zéro du laplacien de gaussienne). Ce filtre donne des résultats de bonne qualité. Le critère de Canny (critère $\Sigma\Lambda$) vaut 1,12 pour le filtre RIF optimal, 0,97 pour le filtre gaussien.

4.2.3.1.3.2.2. Filtre de Deriche

Le détecteur de Deriche répond exactement aux mêmes critères de qualité que celui de Canny, mais possède une réponse impulsionnelle infinie (filtre RII). Il a pu donc être synthétisé de façon récursive particulièrement efficace [Deriche, 1987]. Le filtre de Deriche a une expression générale de la forme :

$$\psi(x) = -c x \exp(-\alpha |x|) \quad (4.29)$$

avec :

$$c = \frac{(1 - \exp(-\alpha))^2}{\exp(-\alpha)} \quad (4.30)$$

Avec les critères même de Canny il est supérieur au filtre de Canny ($\Sigma\Lambda = 2$). Le paramètre α de Deriche représente alors l'inverse de l'écart type σ de la gaussienne du filtrage de Canny ($\alpha = \frac{\sqrt{\pi}}{\sigma}$).

4.2.3.1.3.2.3. Filtre de Shen et Castan

Il possède également de très bonnes qualités de détection, et s'apparente à la même famille des filtres exponentiels (mais comme il n'est pas dérivable à l'origine, il ne se prête pas à une comparaison avec le critère de Canny) :

$$\psi(x) = c. \operatorname{sgn}(x) \exp(-\alpha|x|) \quad (4.31)$$

Le filtre de Shen et Castan, en raison de sa forme à l'origine, donne une localisation très précise des contours, mais il est sensible aux bruits.

4.2.3.2. Techniques de segmentation

La technique de segmentation consiste à décomposer des scènes complexes en des entités identifiables et plus aisément traitables que la scène ou l'image d'origine [25]. Elle peut être complète, dans ce cas les objets identifiés sont séparés de l'arrière-plan de façon individuelle, c'est-à-dire que chaque objet possède un identifiant qui le distingue, ou partielle, dans ce cas la quantité d'information extraite est réduite dans le but d'accélérer les traitements.

On distingue trois types de segmentation :

- la segmentation par classification, qui repose sur la mesure des similarités entre les valeurs des pixels, dont le seuillage représente la forme la plus simplifiée ;
- la segmentation basée sur les contours, dans laquelle on recherche les contours dans l'image, qui seront ensuite considérés comme frontières entre les régions de l'image ;
- la segmentation basée sur les régions, dans laquelle on recherche les similarités locales entre groupes de pixels voisins. Dans cette technique on retrouve des concepts tels que fusionnement, fractionnement, et accroissement de régions.

La segmentation représente la problématique la plus difficile à résoudre dans toute la chaîne de traitement, car elle dépend de plusieurs paramètres, tels que le choix du capteur, l'illumination, l'arrière-plan, qui font qu'il n'existe pas de solution universelle permettant sa mise en œuvre.

4.2.3.3. Techniques de haut-niveau

Une fois les informations extraites sous forme d'un ensemble d'entités, la dernière étape consiste à prendre une décision en fonction de ces informations. Selon l'application visée, cette décision peut être une simple valeur booléenne (l'image fait partie de la catégorie visée ou non),

une commande moteur (ce sera le cas pour l'évitement d'obstacles) ou un ensemble plus complexe de traitements (mise à jour d'une carte par exemple).

4.3. Planification de trajectoire et évitement d'obstacles

La planification de trajectoire revêt un aspect important en robotique mobile comparée à d'autres classes de systèmes, où seule la commande en boucle fermée est suffisante.

En effet, lorsque le robot est sujet à des contraintes qui restreignent sa mobilité (non holonomie) et/ou qu'il soit amené à se déplacer vers une situation désirée dans un environnement présentant des obstacles, il devient nécessaire de planifier une trajectoire qui tienne compte de ces contraintes, c'est-à-dire une trajectoire admissible et sans collision. Etant donnée cette trajectoire, la phase de commande consistera à élaborer des lois qui en garantissent le suivi avec une certaine précision, voir [3].

Selon la situation, la planification de trajectoire peut être réalisée hors ligne (off-line) si l'environnement est supposé parfaitement connu et stable, ou en ligne (on-line) lorsque l'environnement est inconnu à priori, incertain, et/ou dynamique. Dans le cadre de ce travail, on s'intéresse à ce deuxième type de planification, étant donné qu'il correspond aux objectifs de notre application.

Les méthodes entrant dans ce cadre sont dites méthodes d'évitement réactif d'obstacles sans trajectoire de référence, dont la méthode de champs de potentiel, largement utilisée dans la littérature.

4.3.1. Méthode de champs de potentiel

La méthode de champs de potentiel a été initialement introduite par Khatib et utilisée pour la commande et l'évitement d'obstacles pour un bras manipulateur [41], et a été étendue par la suite pour la planification de mouvement.

En robotique mobile, cette méthode consiste à assimiler le robot à une particule contrainte à se déplacer dans un champ de potentiel fictif obtenu par la composition d'un premier champ attractif au but et d'un ensemble de champs répulsifs modélisant la présence d'obstacles dans l'espace du robot (figure 4.7). Les déplacements de celui-ci sont alors calculés itérativement par un algorithme de descente du gradient du potentiel obtenu.

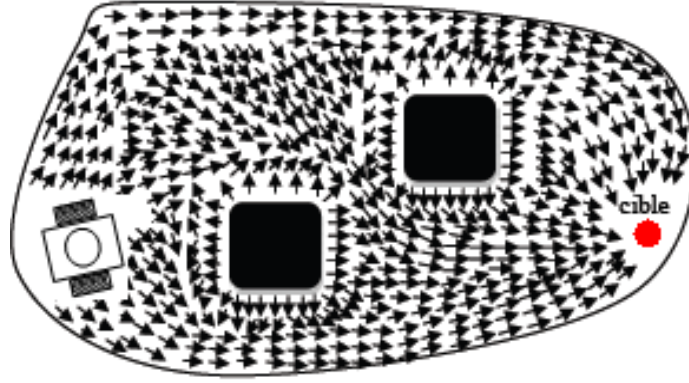


Figure 4.7 : champs de potentiel attractif et répulsif

A noter que cette méthode est sujette à des minima locaux engendrant des configurations de blocage ou d'oscillation. Ces minima sont généralement liés à la géométrie et à la répartition des obstacles dans l'espace de travail du robot (ex : passage étroit) et aussi aux coefficients de pénalités qui leur sont associés pendant la construction du champ de potentiel [38].

Afin de remédier à ces problèmes, Barraquand et Latombe proposent dans [43] d'intégrer à la minimisation du potentiel appliqué au robot l'exploration d'une représentation hiérarchique en bitmap de l'espace des configurations. L'évitement ou plutôt la sortie d'un minimum local est effectuée par l'application de mouvements aléatoires, dits browniens.

4.3.1.1. Présentation de l'algorithme

Dans cette approche, le robot est considéré dans l'espace de configuration $\mathcal{C}$ comme une particule soumise à un champ de potentiel virtuel $U(q)$, avec q l'état du robot (typiquement, pour un robot mobile $q = (x, y)$).

A chaque itération, la force virtuelle $F(q) = -\nabla U(q)$ induite par le champ de potentiel, indiquera la direction à suivre par le robot. Cette force est donnée par :

$$F(q) = -\nabla U(q) = (\partial U / \partial x, \partial U / \partial y)^T, \text{ avec } q = (x, y) \quad (4.32)$$

Le champ de potentiel est défini comme la somme d'un champ attractif $U_{att}(q)$, agissant sur le robot de manière à l'amener dans la configuration désirée, et d'un champ répulsif $U_{rep}(q)$, tenant le robot éloigné des obstacles.

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (4.33)$$

4.3.1.1.1. Potentiel attractif

Le potentiel attractif est défini comme une fonction indépendante de l'ensemble des configurations répulsives. Il doit atteindre son minimum lorsque le robot est dans la configuration désirée.

Il peut être simplement défini sous forme d'une parabolique :

$$U_{att}(q) = \frac{1}{2} \xi \rho_{goal}^2(q) \quad (4.34)$$

avec ξ un scalaire positif, et $\rho_{goal}(q)$ la distance euclidienne : $\|q - q_{goal}\|$ séparant le robot de la configuration désirée q_{goal} .

La fonction $U_{att}(q)$ est donc positive ou nulle, et atteint son minimum à q_{goal} , où $U_{att}(q_{goal}) = 0$.

La force F_{att} est obtenue en appliquant l'équation (4.). Elle est différentiable partout dans C et on a :

$$F_{att}(q) = -\nabla U_{att}(q) = -\xi \rho_{goal}(q) \nabla \rho_{goal}(q) = -\xi (q - q_{goal}) \quad (4.35)$$

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

$$U_{att}(q) = \xi \rho_{goal}(q) \quad (4.36)$$

On aura alors une force de type :

$$F_{att}(q) = -\xi \nabla \rho_{goal}(q) = -(q - q_{goal}) / \|q - q_{goal}\| \quad (4.37)$$

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

4.3.1.1.2. Potentiel répulsif

Le potentiel répulsif est choisi indépendamment de la configuration désirée, décroissant quand le robot s'éloigne des configurations indésirables et prenant de grandes valeurs lorsqu'il est dans le voisinage d'une d'entre elles. En d'autres termes, il doit servir à créer une barrière de potentiel autour des obstacles, sans pour autant affecter le mouvement du robot lorsque celui-ci est suffisamment loin des obstacles.

Une des formes correspondant à ces objectifs consiste à choisir :

$$\begin{cases} U_{rep}(q) = \frac{1}{2}\eta(1/\rho(q) - 1/\rho_0)^2 & \text{si } \rho(q) \leq \rho_0 \\ = 0 & \text{si } \rho(q) > \rho_0 \end{cases} \quad (4.38)$$

ρ_0 est appelée la distance d'influence des obstacles. La fonction U_{rep} est positive ou nulle et tend vers l'infini lorsqu'on se rapproche de la frontière des obstacles.

On a dans ce cas :

$$\begin{cases} F_{rep}(q) = \eta(1/\rho(q) - 1/\rho_0)^2 & \text{si } \rho(q) \leq \rho_0 \\ = 0 & \text{si } \rho(q) > \rho_0 \end{cases} \quad (4.39)$$

4.3.1.1.3. Planification de trajectoire induite par les champs de potentiel

Une fois les potentiels définis, plusieurs techniques sont possibles. Les deux plus courantes sont, pour des espaces de configuration à faibles dimensions (< 5) :

- La descente de gradients ;
- La recherche du chemin de moindre coût.

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

$$x(q_i + 1) = x(q_i) + \delta_i \partial U / \partial x(x, y)$$

$$y(q_i + 1) = y(q_i) + \delta_i \partial U / \partial y(x, y)$$

L'algorithme s'arrêtera lorsque le robot arrive à une distance de q_{goal} inférieure à δ_i .

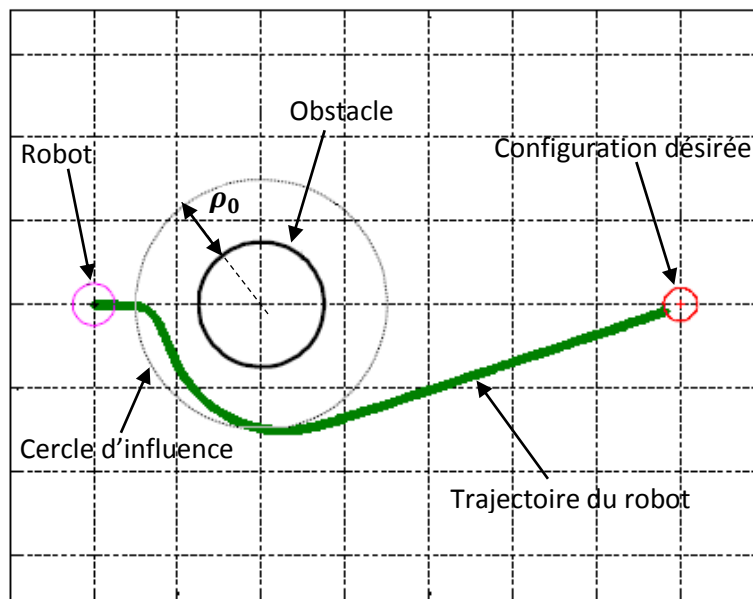


Figure 4.8 : Evitement d'un obstacle par la méthode de champs de potentiel

4.3.1.2. Autres méthodes de planification

En plus de la méthode de champs de potentiel, il existe d'autres méthodes d'évitement réactif d'obstacles sans trajectoire de référence, dont on cite [45] :

4.3.1.2.1. Méthode Steering Angle Field

Cette approche a été proposée par [20] pour un robot de type unicycle. Il s'agit de calculer pour un ensemble de vitesses linéaires des domaines de l'angle de braquage (SAF) qui n'entraînent pas de collision avec les obstacles perçus. Cette méthode utilise une discrétisation en grille du plan de travail autour du robot. Elle exploite la possibilité de pré-calculer et de stocker dans des tables de recherche les SAF pour chaque cellule obstacle de la grille.

4.3.1.2.2. Méthode de la fenêtre dynamique

Cette technique proposée dans [26] travaille dans l'espace des commandes du robot. La taille du domaine de recherche des vitesses accessibles (c'est à dire n'entraînant pas de collisions) est réduite par la prise en compte explicite de la dynamique (les capacités d'accélération) du système. Les commandes envoyées au robot sont le résultat de la maximisation sur ce domaine de recherche d'une fonction de coût liée à la position but. Cette méthode a été développée pour des robots se déplaçant à des vitesses élevées (1m/s) dans des environnements intérieurs encombrés.

4.3.1.2.3. Méthode du diagramme de proximité

Cette approche proposée par Minguez [52] repose sur un diagramme de proximité des obstacles, mis à jour au fur et à mesure du déplacement du robot. En fonction de l'allure de ce diagramme (nombre de passages sans obstacles, taille des passages, etc.), un comportement adapté (exploration, avancement vers le but, retour en arrière, etc.) est sélectionné. La direction de mouvement la plus prometteuse par rapport au but est alors choisie. Cette méthode a été appliquée à un véhicule de type unicycle grâce à une expression des obstacles dans l'espace des points accessibles en un mouvement élémentaire (un arc de cercle en l'occurrence), obtenue par une transformation dans l'espace dit «Ego-cinématique».

4.4. Conclusion

Dans ce chapitre nous avons introduit et présenté les différents aspects de la vision artificielle qui sont utilisés en robotique mobile. Ceux-ci concernent principalement les techniques permettant la perception et l'interprétation de l'environnement dans lequel un robot mobile est amené à évoluer. La perception est réalisée à travers la caméra, cette dernière joue un rôle très important dans le processus et un soin particulier doit être apporté à sa modélisation et son calibrage. Nous avons aussi présenté quelques techniques de planification réactive de trajectoires, notamment la méthode de champ de potentiel, qu'on utilisera dans la partie applicative pour planifier une trajectoire évitant les obstacles présents dans l'environnement du robot.

Chapitre 5

Applications et résultats

5.1. Introduction

Ce chapitre est dédié à la mise en œuvre de l'approche préconisée pour traiter la problématique de la commande du robot unicycle par vision. Comme indiqué dans le chapitre 1, nous proposons deux types d'applications, la première concerne le suivi d'une ligne tracée sur le sol, et la deuxième traite le problème de navigation avec évitement d'obstacles. Pour ce faire, nous commençons d'abord par présenter l'architecture de commande adoptée. Par la suite, nous donnons une description du système robot+caméra, où nous mettons en évidence les différents calculs liés au calibrage et à la modélisation de la caméra ainsi que les transformations permettant de passer entre repères caméra-robot-monde.

Ensuite, après une présentation globale de la chaîne de traitement permettant d'extraire l'information utile dans l'espace image, nous appliquons cette chaîne de traitement dans le cadre des deux applications choisies, où nous calculons dans chaque cas, le chemin de référence permettant de répondre à l'objectif associé.

A partir de ce point, nous utilisons la technique du suivi de chemin pour commander le robot et simuler l'exécution des trajectoires obtenues.

5.2. Architecture de commande

Dans le cadre de ce travail, la vision est utilisée pour la capture et l'interprétation de la scène représentant l'environnement du robot afin de calculer une trajectoire de référence à suivre. Une fois cette trajectoire obtenue, l'asservissement est opéré sur la configuration du robot, qui est supposée mesurable, pour l'amener à cette trajectoire de référence.

La fréquence à laquelle la boucle de vision est activée dépend du temps nécessaire pour le calcul et l'exécution de la trajectoire désirée. On explicitera cette situation dans le cadre de la navigation avec évitement d'obstacles où cette période est estimée à 4 secondes, (2sec) pour le traitement et (2sec) pour l'exécution de la trajectoire. L'architecture de commande, qui est commune aux deux applications, est donnée dans le schéma de la figure 5.1.

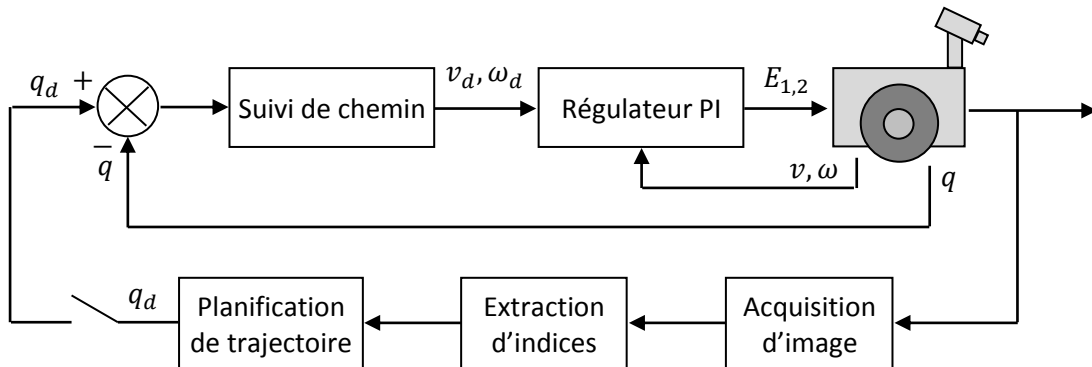


Figure 5.1 : Architecture adoptée pour la commande par vision du robot unicycle

La boucle de commande est scindée en deux blocs, le premier, calculé à la base du modèle cinématique du robot, permet d'obtenir les consignes de vitesses linéaire et angulaire nécessaire pour le suivi de la trajectoire de référence, et le deuxième calcule les tensions de commande à appliquer au niveau des actionneurs pour reproduire les profils de vitesses. Cette architecture a l'avantage d'être plus robuste, puisque l'influence des perturbations qui peuvent survenir dans chaque boucle est locale, et sa correction n'incombe que la loi de commande y afférente.

5.3. Description du système

Le système envisagé dans le cadre de ce travail se compose d'un robot mobile de type unicycle, muni d'une caméra monoculaire. La caméra est disposée à une hauteur H du sol et pointe vers ce dernier avec un angle α par rapport à l'axe horizontal. Le choix de la valeur de l'angle α est directement lié à la longueur du champ de vision qu'on désire avoir pour notre système.

Le repère lié au robot a pour origine le point G situé à une hauteur h du sol, et correspondant au centre de l'axe des roues motrices, considéré comme le centre de gravité du robot. A l'instant initial, ce repère est défini par rapport au repère monde $\{O, \vec{X}, \vec{Y}, \vec{Z}\}$ tel que $\vec{X}_R = \vec{Y}$, $\vec{Y}_R = -\vec{X}$ et $\vec{OG} = h\vec{Z}$. Quant au repère de la caméra $\{O_c, \vec{X}_c, \vec{Y}_c, \vec{Z}_c\}$, sa situation par rapport au repère du robot peut être définie par la matrice de transformation homogène T telle que :

$$T = \text{trans}(d, 0, H - h) \times \text{Rot}\left(y, \frac{\pi}{2} + \alpha\right) \times \text{Rot}\left(z, \frac{\pi}{2}\right)$$

$$T = \begin{pmatrix} 0 & \cos \alpha & -\sin \alpha & d \\ 1 & 0 & 0 & 0 \\ 0 & -\sin \alpha & -\cos \alpha & H - h \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.1)$$

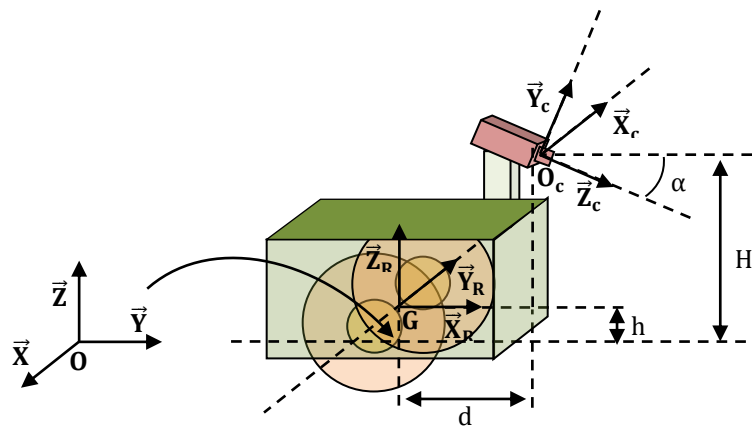


Figure 5.2 : Description du système robot-caméra

Remarque : ne disposant pas d'un robot mobile, l'expérimentation est réalisée avec un dispositif tel que la caméra est installée dans la configuration indiquée.

5.3.1. Description de la caméra utilisée

La caméra utilisée est une webcam-USB de type CMOS, qui présente les caractéristiques suivantes :

- résolution jusqu'à (1028×1024) pixels ;
- taux de transmission vidéo : 320×240 à 30 images/sec, et 640×480 à 15 images/sec ;
- formats de sortie : YUY2;
- rapport signal sur bruit supérieur à 48dB ;
- ajustement automatique de la brillance ;
- ajustement automatique du contraste ;
- réglage manuel de la netteté ;
- microphone intégré.



5.3.2. Transformation de passage de l'espace Image vers l'espace cartésien

Dans cette section, nous allons décrire la transformation permettant de calculer les coordonnées 3D des points du monde réel notées (x_w, y_w, z_w) , à partir de leurs représentations dans le plan Image (coordonnées pixéliques), notées (x_c, y_c) .

Pour caractériser cette transformation, on utilise le modèle du sténopé présenté dans le chapitre 4. Puisque la caméra pointe vers le sol, si on suppose que tous les points qui nous intéressent appartiennent au sol, et sont de ce fait décrits uniquement par la paire (x_w, y_w) , ce modèle peut simplement être obtenu grâce à des transformations géométriques de base (figure 5.3).

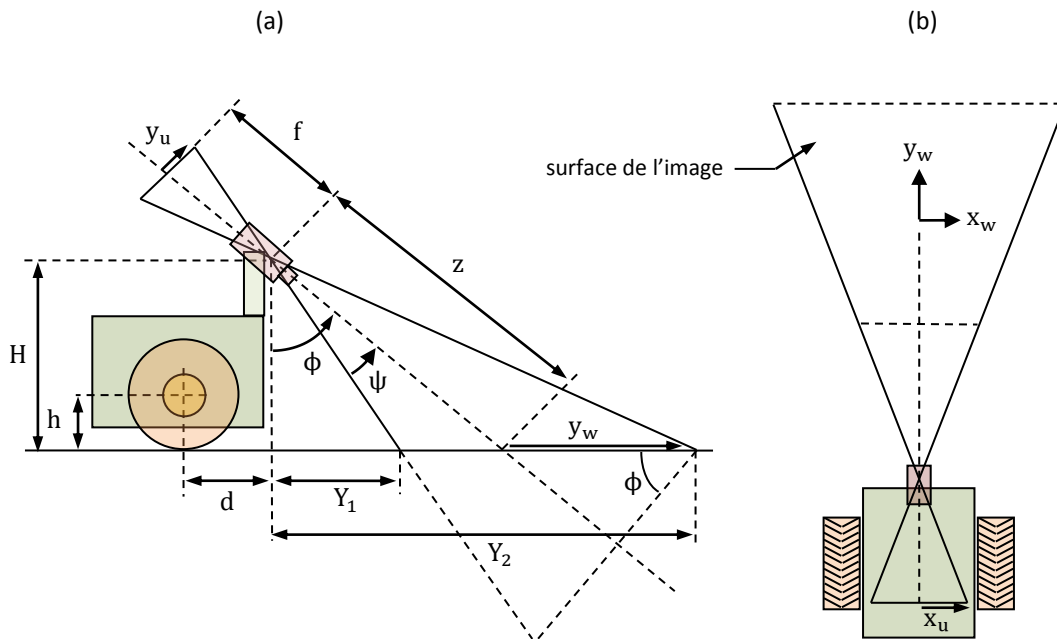


Figure 5.3 : Modèle géométrique de la caméra. (a) coupe transversale. (b) vue de dessus

D'après le schéma de la figure (5.3), on définit :

$$\frac{y_u}{f} = \frac{y_w \cos \phi}{z + y_w \sin \phi} \quad (5.2)$$

$$\frac{x_u}{f} = \frac{x_w}{z + y_w \sin \phi} \quad (5.3)$$

$$z = \frac{H}{\cos \phi} \quad (5.4)$$

où H est la hauteur de la caméra, f sa distance focale et $\phi = \frac{\pi}{2} - \alpha$, son angle d'inclinaison par rapport à la verticale.

En combinant les équations (5.2), (5.3) et (5.4), on obtient :

$$y_w = -\frac{y_u H}{\cos \phi (f \cos \phi + y_u \sin \phi)} \quad (5.5)$$

$$x_w = \frac{x_u}{f} \left(\frac{H}{\cos \phi} + y_w \sin \phi \right) \quad (5.6)$$

avec :

$$\begin{aligned} x_u &= (x_c - C_x) d_x \\ y_u &= (y_c - C_y) d_y \end{aligned} \quad (5.7)$$

où : C_x et C_y représentent les coordonnées du centre de l'image, et d_x et d_y les dimensions horizontale et verticale d'un élément de la matrice du capteur.

On définit aussi :

$$\begin{aligned} Y_1 &= H \tan(\phi - \psi) \\ Y_2 &= H \tan(\phi + \psi) \end{aligned} \quad (5.8)$$

où Y_1 correspond à la distance sur le sol que la caméra ne peut percevoir, et $Y_2 - Y_1$ correspond à la longueur du champ de vision de la caméra.

Les coordonnées x_w, y_w données par (5.6) et (5.5) sont définies par rapport au repère de la caméra. Il ne nous reste maintenant qu'à écrire ces coordonnées dans le repère monde. D'après les descriptions données dans les figures 5.2 et 5.3, on obtient :

$$\begin{aligned} {}^0x_w &= x_w \\ {}^0y_w &= y_w + Y_1 + d \end{aligned} \quad (5.9)$$

Pour notre application, nous avons pris :

$H = 126 \text{ mm}$, $h = 20 \text{ mm}$, $d = 76 \text{ mm}$

$\alpha = 35^\circ$, $\phi = 55^\circ$,

d'où : $z = 219,6 \text{ mm}$, $\psi = 19^\circ$, $X_1 = 91,5 \text{ mm}$, $X_2 = 439,4 \text{ mm}$.

avec $C_x = 320$ et $C_y = 240$.

Les paramètres restants, à savoir f , d_x et d_y seront déterminés à l'issu de l'opération de calibrage de la caméra, présentée dans la section suivante.

5.4. Calibrage de la caméra

Le calibrage de la caméra a été effectué avec « Camera calibration toolbox » de MATLAB, qui utilise un algorithme basé sur l'extraction des coins d'une mire plane composée de carrés noirs sur fond blanc (en forme d'un échiquier). Le calibrage nécessite au moins 20 prises dans lesquelles la mire doit être disposée à peu près à la même distance avec différents angles d'inclinaison par rapport à la caméra. Pour notre part, nous avons utilisé une mire avec des carrées de 10 mm de coté.

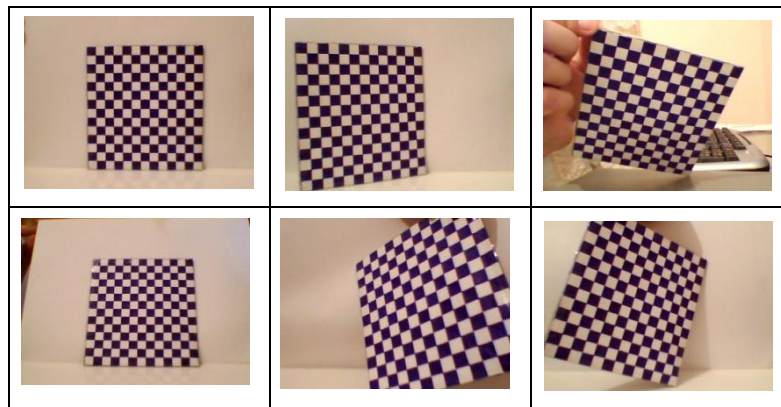


Figure 5.4 : exemples de prises effectuées pour le calibrage de la caméra

Cette opération nous a permis d'obtenir les paramètres intrinsèques de la caméra, à savoir :

- distance focale : $f = 4,8 \text{ mm}$
- point principal : $(311, 256) \text{ pixels}$
- dimensions d'un élément de la matrice du capteur : $(6,9 \times 7,01) \mu\text{m}$

Afin de simplifier les calculs, les facteurs de distorsion ont été négligés.

5.5. Présentation de la chaîne de traitement

Dans les points qui suivent, nous allons présenter les différentes phases de traitement qui sont communes aux deux applications. Ceci nous permettra, lors de l'exposition des résultats

relatifs à chaque application, d'éviter la répétition et de s'atteler uniquement aux développements spécifiques à chaque application.

5.5.1. Acquisition d'images

Les images produites par la caméra utilisée sont des images couleurs de type JPEG, dont on a fixé la résolution à 640×480 pixels, qui est une résolution acceptable que ce soit en termes de qualité ou de lourdeur des calculs.

Une fois la caméra branchée sur le PC, la procédure d'acquisition d'image est réalisée sous MATLAB de façon automatique, en exécutant le code suivant :

```
obj=videoinput('winvideo','1','YUY2_640x480');  
preview(obj);  
pause(3);  
set(obj,'ReturnedColorspace','RGB');  
mat=getsnapshot(obj);  
delete(obj);
```

L'image acquise est sauvegardée dans la variable « mat ».

5.5.2. Phase de prétraitements

Cette phase porte essentiellement sur l'amélioration et le filtrage d'images. Pour des raisons de simplicité, les images acquises sont d'abord converties en niveaux de gris codés sur 8 bits (256 nuances de gris), ce qui nous permet d'utiliser certaines fonctions de traitement d'images disponibles sous MATLAB.

La première opération consiste à réaliser un ajustement de contraste, qui revient à redistribuer les valeurs de l'intensité des pixels de l'image de sorte à occuper toute la gamme de nuances possibles (256 valeurs), ce qui donne une image plus contrastée.

En second lieu, on effectue une opération de filtrage, afin de réduire le bruit, et ainsi préparer l'image à l'étape d'extraction d'information. Dans notre cas, nous avons utilisé un filtre de type médian. A noter que le filtrage a pour conséquence de rendre l'image un peu floue.

5.5.3. Détection de contours

Cette phase est la plus importante du processus de traitement d'image. En effet, c'est au cours de cette étape qu'on procède à l'extraction de l'information utile de notre scène. Dans le cas de la première application, le résultat est direct, puisque les contours détectés correspondent à la ligne qu'on veut extraire de l'image. Pour ce qui est de la deuxième application, les contours détectés permettent de délimiter les obstacles, et renseignent donc sur leurs positions dans le plan image. Nous avons testé trois types de filtres, dont les fonctions sont disponibles dans « Image toolbox » de MATLAB à savoir, les filtres de Prewitt, de Sobel, et de Canny. C'est ce dernier que nous avons choisi vu sa qualité de détection supérieure comparativement aux deux autres.

5.5.4. Transformation de coordonnées du plan image vers l'espace cartésien

Une fois les entités utiles détectées dans le plan image, on procède au calcul des coordonnées cartésiennes correspondantes, dans le repère monde. Ces calculs sont effectués en utilisant les équations (5.5-5.9). Pour ce qui est de la première application, il s'agit d'un calcul direct de la ligne à suivre. Quant à la deuxième, on calcule les coordonnées cartésiennes correspondant aux différents obstacles et aux limites de la scène observée.

5.5.5. Affinage de trajectoire par courbes de Bézier

Comme nous allons le voir, les trajectoires calculées par transformation depuis le plan image sont des trajectoires saccadées et peuvent même présenter des discontinuités, dues aux déformations subies dans l'image et aux algorithmes de traitement. Par ailleurs, les trajectoires calculées dans l'espace cartésien ne sont pas toujours adaptées au besoin, comme c'est souvent le cas en robotique mobile où la trajectoire à suivre doit être particulièrement lisse et ne présentant pas de discontinuité par rapport à la courbure. Pour résoudre ce problème, on prévoit généralement un lissage par courbes de Bézier, dont on explique brièvement le principe.

Principe du lissage par courbes de Bézier

La courbe de Bézier est une technique rapide efficace pour construire une trajectoire lisse et uniforme à partir d'une séquence de points. A partir de $n + 1$ points de contrôle $P_0, P_1, P_2, \dots, P_n$ une courbe de Bézier se définit comme :

$$C(t) = \sum_{i=0}^n B_{n,i}(t)P_i \quad (5.10)$$

où $t \in [0, 1]$ et les coefficients $B_{n,i}$ sont des polynômes de Bernstein de la forme :

$$B_{n,i}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \quad (5.11)$$

Le point qui correspond à t est la moyenne pondérée des points de contrôle où les coefficients de pondération sont $B_{n,i}(t)$, voir figure (5.5) ci-après.

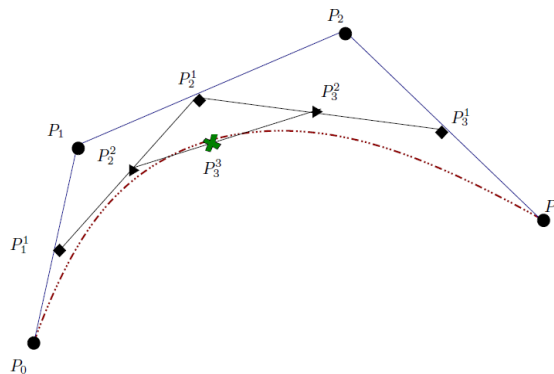


Figure 5.5 : construction d'une courbe de Bézier

Nous allons à présent présenter en détail les développements relatifs à chaque application avec résultats et simulations de la partie commande et exécution de trajectoire.

5.6. Application au suivi d'une ligne tracée sur le sol

Dans cette application, on souhaite faire suivre par le robot la ligne qu'on a tracée auparavant sur le sol (figure 5.6).

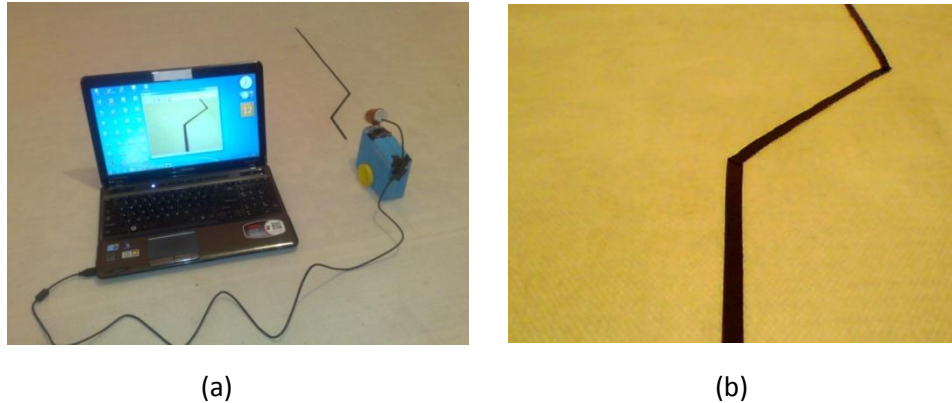


Figure 5.6 : Application du suivi de ligne. (a) observateur externe. (b) vue par le robot

Dans ce cas, la trajectoire de référence est indiquée (pas besoin de planification), et la chaîne de traitement consistera d'abord à récupérer cette ligne dans le plan image, puis à la convertir vers l'espace cartésien, où elle sera lissée pour enfin être prête à se faire exécuter par le robot via la commande en boucle fermée.

5.6.1. Chaîne de traitement d'image

5.6.1.1. Prétraitements

On présente dans la figure (5.7) le résultat de l'opération de prétraitement effectuée sur l'image d'origine (figure 5.6 (b)), où :

- (a) correspond au résultat de la transformation de l'image d'origine en niveaux de gris codés sur 8 bits ;
- (b) résultat après ajustement de contraste ;
- (c) résultat après application d'un filtre médian.

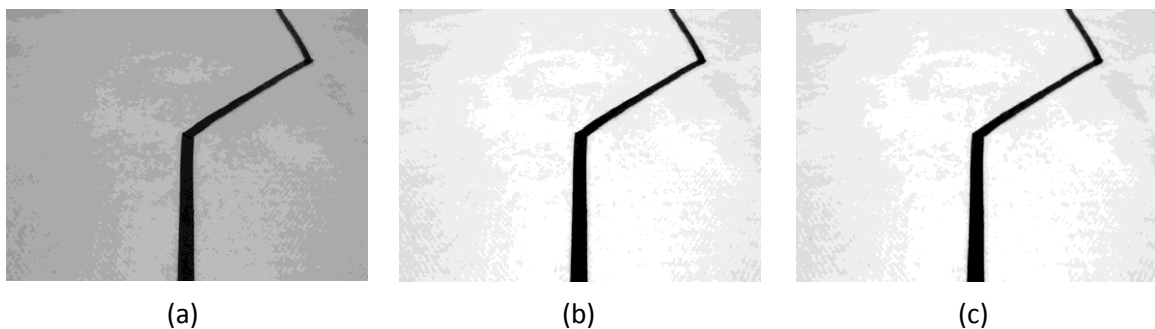


Figure 5.7 : prétraitements de la scène

5.6.1.2. Détection de contours

L'application des techniques de détection de contours sur l'image (5.7.c), donne les résultats suivants (figure 5.8) :

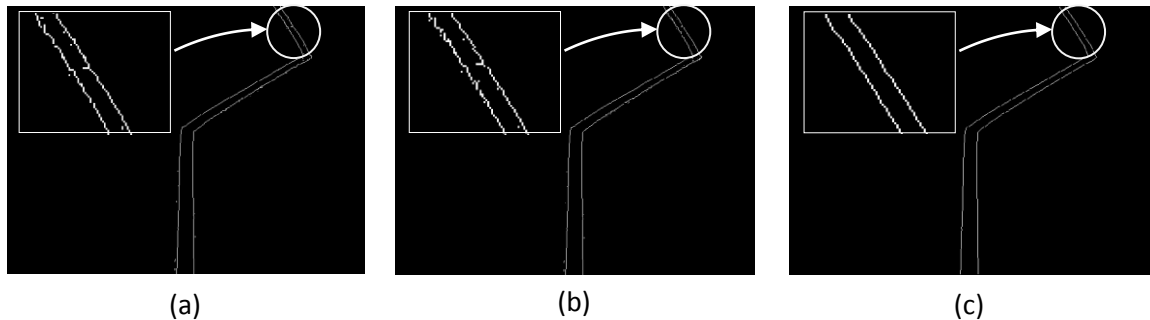


Figure 5.8 : détection de contours. (a) Sobel. (b) Prewitt. (c) Canny

Il apparaît clairement que le résultat obtenu avec le filtre de Canny (figure 5.8.c) est meilleur. De plus, bien qu'il soit possible pour les deux autres filtres, de réduire la détection de faux contours en introduisant un seuil de détection, la valeur de ce dernier dépend des caractéristiques de l'image traitée et s'obtient donc par tâtonnement, ce qui est inadapté à notre contexte. Ce qui justifie notre choix pour le filtre de Canny.

5.6.1.3. Extraction de la ligne dans le plan image

Le résultat de l'étape précédente nous a permis de localiser, dans le plan image, la bande correspondante à la ligne qu'on a tracée sur le sol. En effet, celle-ci est située, en chaque ligne de l'image, entre les deux pixels blancs représentant les contours détectés. Afin de réduire cette bande en un vecteur de points, on calcule, en chaque ligne de l'image, le point milieu de la bande en cette ligne (figure 5.9.a). On obtient ainsi la ligne de référence dans le plan image (figure 5.9.b).

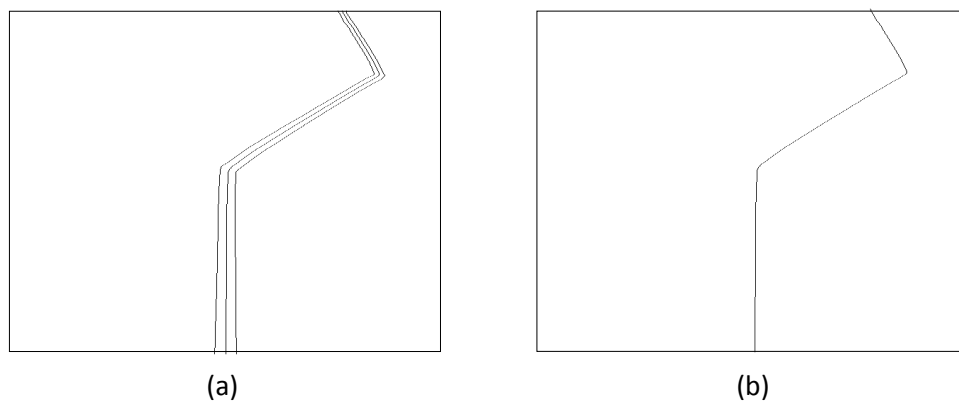


Figure 5.9 : Extraction de la ligne dans le plan image. (a) calcul de la ligne. (b) ligne extraite

5.6.2. Transformation de la ligne vers l'espace cartésien

La ligne extraite dans le plan image peut être considérée comme un vecteur de points V , tel que : $V(i) = j$, où i représente l'indice de la ligne, et j celui de la colonne d'un point appartenant à V dans l'image.

Le calcul des coordonnées cartésiennes dans le repère monde des points caractérisés par le vecteur V est effectué en utilisant les équations (5.5-5.9), en notant que :

$$\begin{cases} x_c = V(i) = j \\ y_c = i \end{cases} \quad (5.12)$$

La ligne obtenue dans l'espace cartésien (figure 5.10.a) présente des irrégularités, ce qui nécessite un échantillonnage suivi par un lissage par courbe de Bézier (figure 5.10.b).

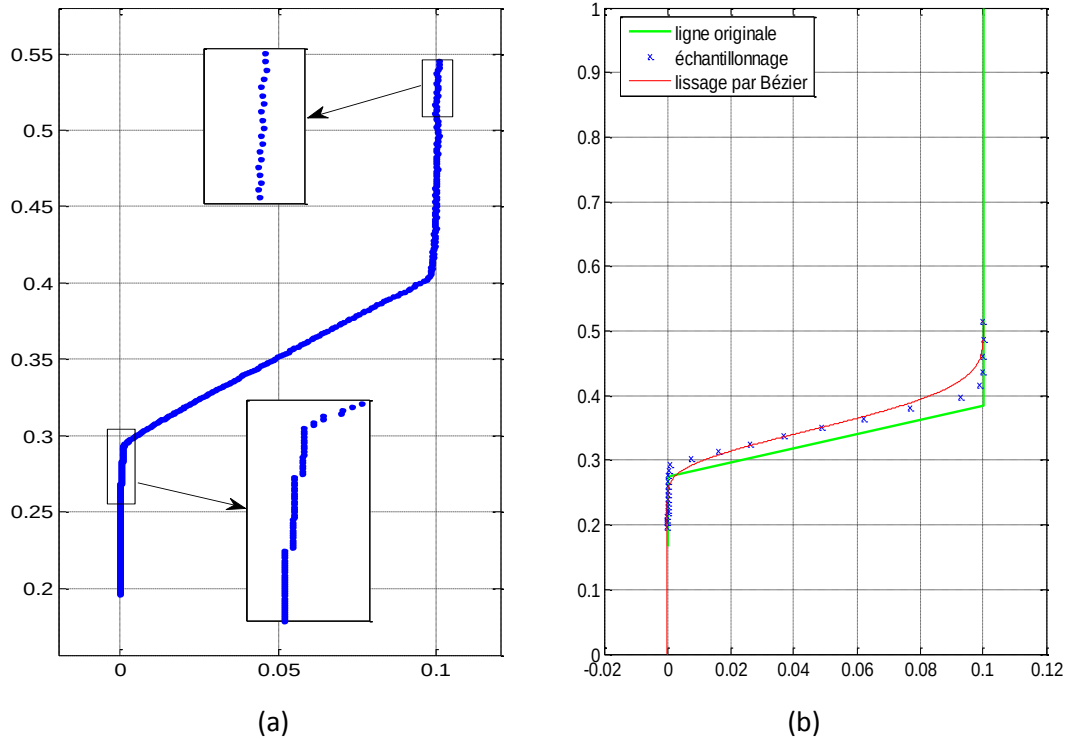


Figure 5.10 : Génération de trajectoire dans l'espace cartésien

La courbe ainsi obtenue est décrite par rapport au point $(x, y) = (0, 0)$ qui correspond aux origines des repères monde et robot. Puisque cette courbe débute théoriquement à une distance égale à $(Y_1 + d)$ selon l'axe des y , nous générons un ensemble de points permettant de couvrir cette distance, que nous concaténons avec cette même courbe. De cette manière, le robot suivra une courbe composée de la courbe calculée, précédée d'une ligne droite permettant d'atteindre cette courbe.

Dans la pratique, bien qu'il soit extrêmement difficile d'obtenir une description fidèle des paramètres du modèle de la caméra, notamment la valeur de la distance focale, l'orientation de la caméra par rapport au repère du robot, sans oublier les effets de distorsion qu'on a négligés, la reproduction de la ligne d'origine (figure 5.10.b) a été assez réussie, ceci grâce au fait qu'on s'est fixé une longueur de champ de vision assez réduite (jusqu'à 55 cm seulement), ce qui permet de réduire les effets des erreurs de modélisation.

5.6.3. Simulation du suivi de la ligne

Pour réaliser le suivi de la courbe décrite précédemment, nous appliquons au robot la technique de commande par suivi de chemin, où nous avons choisi une vitesse d'avance de 0.3 m/s , ce qui permettra un temps d'exécution de moins de 2 secondes.

Les résultats de simulation sont les suivants :

Nous présentons d'abord la trajectoire exécutée par le robot avec les erreurs de suivi selon x , y et θ par rapport à la courbe de référence (figure 5.11).

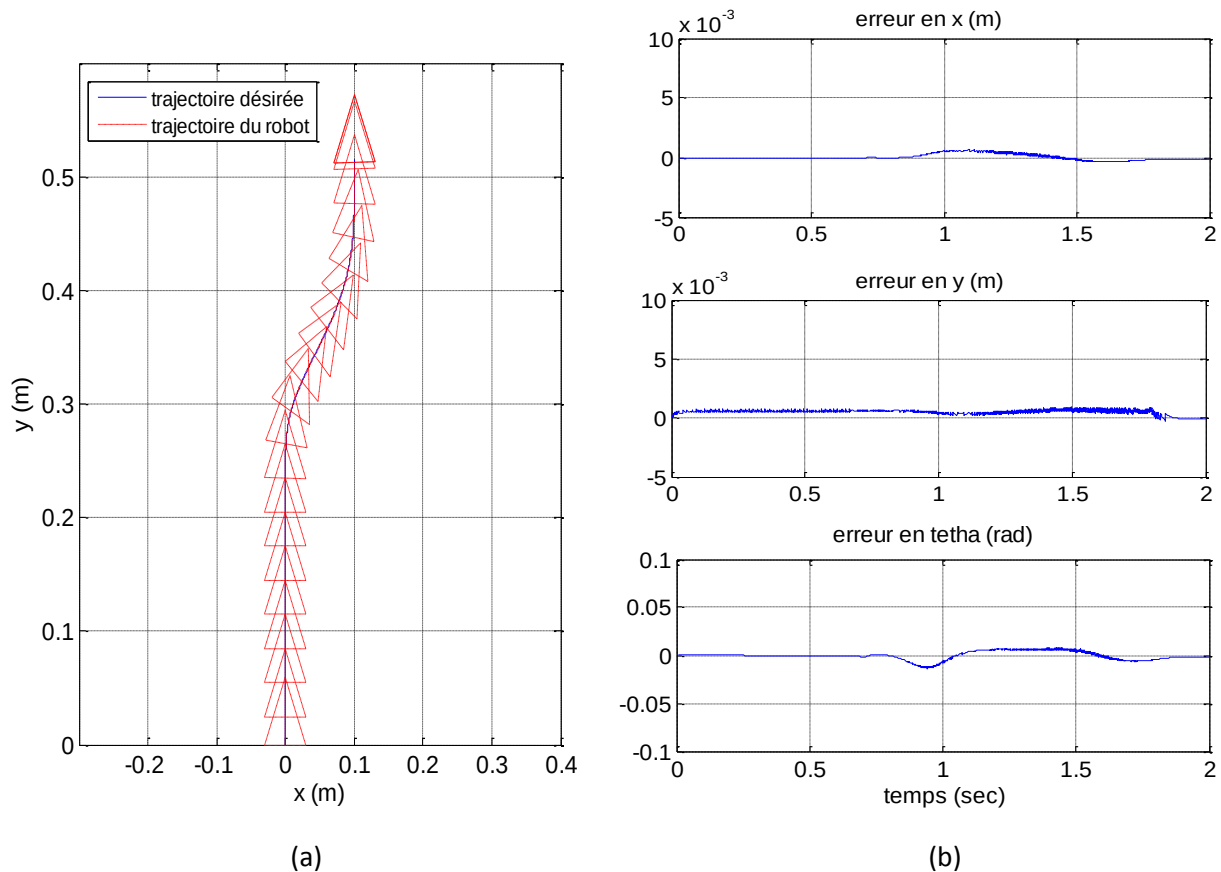


Figure 5.11 : simulation du suivi de ligne. (a) trajectoire du robot. (b) les erreurs de suivi

Le suivi est assuré avec une bonne précision, de l'ordre de 10^{-4} m pour la position, et de 10^{-2} rad pour l'orientation.

Les différentes entrées de commande des modèles cinématique et dynamique du robot (v, ω, eam, ead) sont données dans la figure suivante :

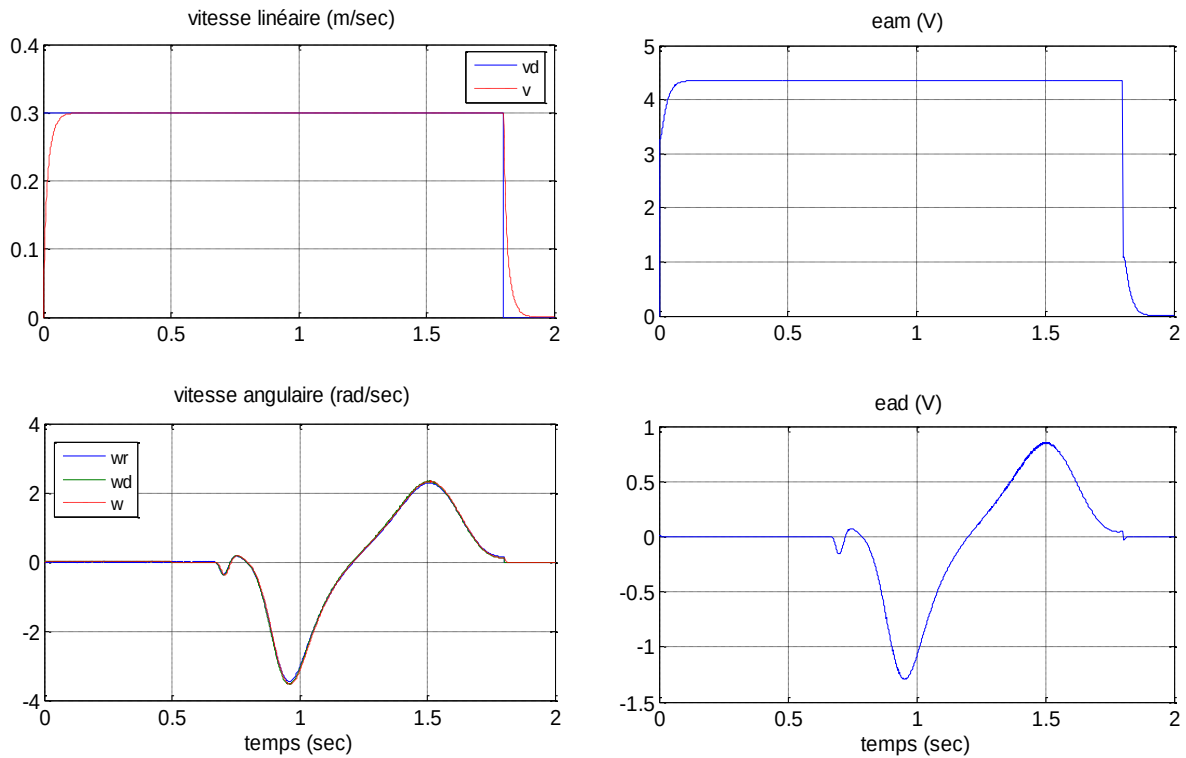


Figure 5.12 : allure des entrées de commande du système

Finalement, nous présentons les signaux électriques appliqués sur chacun des moteurs du robot pour générer ces commandes :

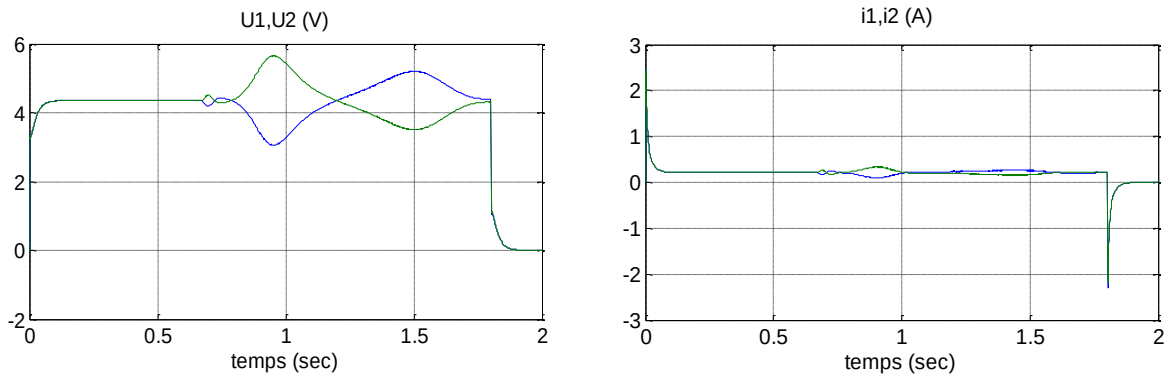


Figure 5.13 : allure des signaux électriques appliqués aux moteurs

Toutes les entrées de commandes sont bien situées dans les limites des valeurs de fonctionnement normal du système. Les pics de courant correspondent aux deux instants de démarrage et de freinage du robot.

5.7. Application à la navigation avec évitement d'obstacles

Dans cette application, on souhaite faire naviguer le robot tout en évitant les éventuels obstacles présents dans son environnement, capturé via la caméra embarquée (figure 5.14).

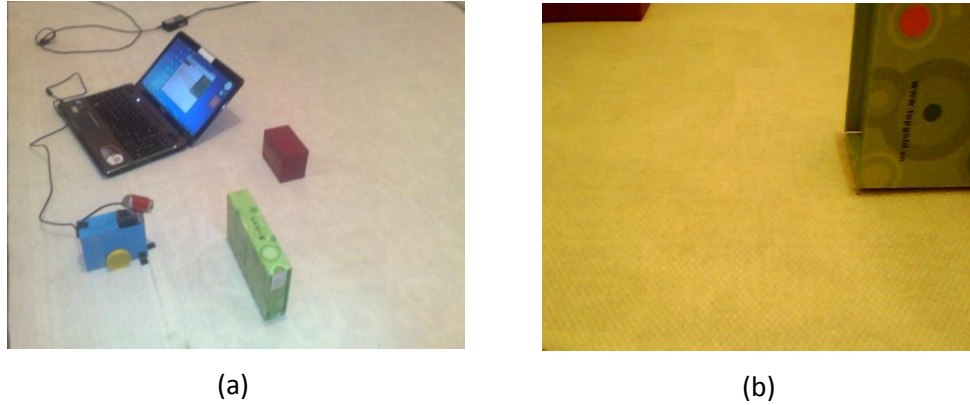


Figure 5.14 : Application d'évitement d'obstacles. (a) observateur externe. (b) vue par le robot

Dans ce cas, la trajectoire de référence sera planifiée de sorte à atteindre le plus haut point accessible de la scène tout en évitant les obstacles, où la partie traitement d'image nous permettra d'obtenir une description de la scène en termes de position des obstacles. Une fois la scène convertie dans l'espace cartésien, on procède à la planification par la méthode du champ de potentiel. Enfin, le suivi de la trajectoire obtenue sera réalisé via la commande en boucle fermée.

5.7.1. Chaîne de traitement d'image

En procédant de la même manière que pour l'application précédente, la chaîne de traitement se présente comme suit :

5.7.1.1. Prétraitements

Les résultats sont donnés dans la figure (5.15).

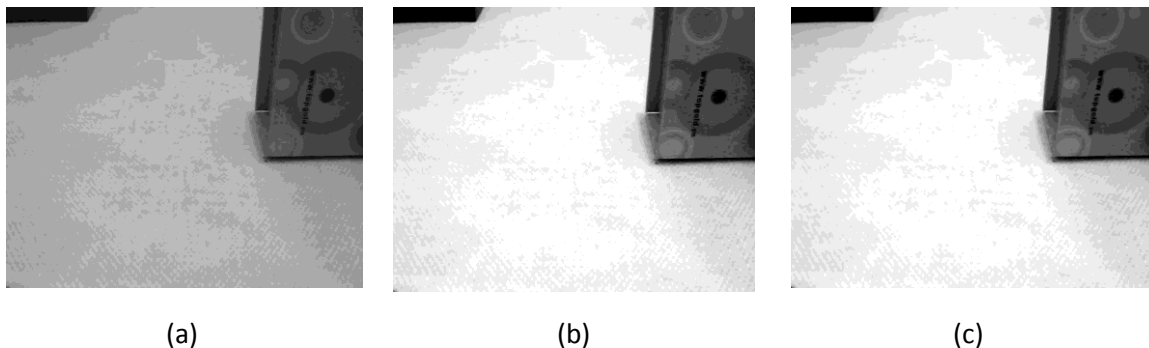


Figure 5.15 : prétraitements de la scène

5.7.1.2. Détection de contours

L'application des techniques de détection de contours sur l'image (5.15.c), donne les résultats suivants (figure 5.16) :

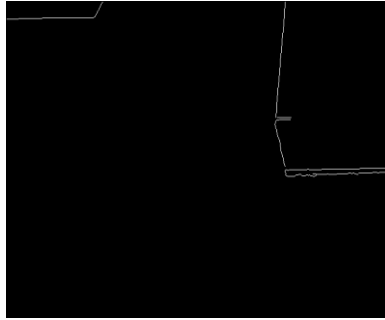


Figure 5.16 : détection de contours par filtre de Canny

5.7.1.3. Segmentation de la scène

Dans cette application, il nous est nécessaire de passer par une segmentation de l'image pour distinguer les régions correspondant aux obstacles de celles libres. Pour ce faire, nous procédons comme suit :

On balaye, de bas en haut, l'image de la figure (5.16) et dès détection d'un contour (i.e. un pixel blanc), qui correspondrait théoriquement à la détection d'un obstacle (plus exactement, la partie qui est en contact avec le sol), on considère que le pixel en question ainsi que tous les pixels de l'image qui lui sont supérieurs comme faisant partie de l'obstacle, auxquels on affectera la valeur 0 (noir). Tous les pixels inférieurs au pixel considéré se verront affectés la valeur 1 (blanc). Ainsi, on obtient une image binaire dont les zones de pixels noirs correspondent aux obstacles, et celle de pixels blancs à la zone sans obstacle. Cette dernière zone ne peut pas encore être qualifiée de navigable, il faut d'abord vérifier que le robot puisse passer au travers de cette zone.

Cette considération permet d'alléger significativement les calculs sans pour autant détériorer les résultats, et n'intervient qu'au cas où les obstacles ne sont pas suffisamment hauts pour obstruer la vision de la caméra par rapport aux zones situées derrière ces obstacles. De plus, puisque dans notre application, l'objectif est celui d'atteindre le plus loin point accessible de la scène dans les limites du champ de vision de la caméra (navigation locale), les zones situées derrière les obstacles (qu'ils soient suffisamment haut ou pas) n'ont que peu d'intérêt.

L'application de l'algorithme décrit ci-dessus donne le résultat suivant :



Figure 5.17 : segmentation de la scène

5.7.1.4. Décomposition en cellules

Afin de faciliter le traitement et réduire le temps de calcul, on procède à une décomposition de la scène représentée par la figure 5.17 en cellules carrées de côté noté a . le traitement s'effectuera désormais sur des blocs de $(a \times a) \text{pixel}^2$.

De plus, les irrégularités des contours délimitant les obstacles se verront disparaître en utilisant l'astuce suivante : En comparant les cellules avec l'image segmentée (par convolution), si un des pixels d'une cellule correspond à un obstacle (valeur 0), alors toute la cellule se verra attribuée l'étiquette obstacle. Ceci a pour conséquence d'augmenter les dimensions des obstacles, ce qui permet d'avoir une marge de sécurité supplémentaire par rapport aux obstacles réels. Dans la figure suivante (5.18), nous présentons le résultat de cette décomposition, avec $a = 20$.

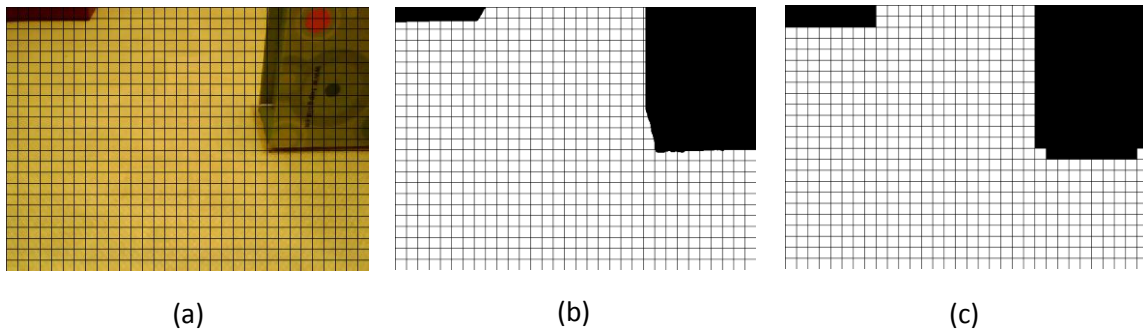


Figure 5.18 : décomposition de la scène en cellules

5.7.2. Transformation de la scène vers l'espace cartésien

En utilisant les équations 5.5-5.9, on procède à la transformation de notre scène vers l'espace cartésien, par le calcul des coordonnées de tous les points d'intérêt, à savoir les points délimitant la scène, ainsi que les points délimitant les obstacles. Le résultat obtenu est donné dans la figure suivante :

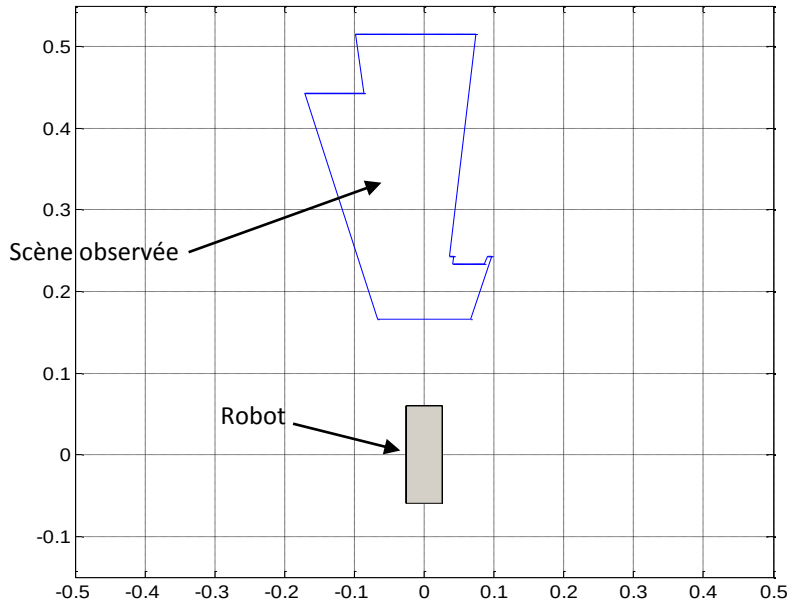


Figure 5.19 : transformation de la scène vers l'espace cartésien

5.7.3. Planification de trajectoire par champs de potentiel

Nous utilisons l'algorithme présenté dans le chapitre 4.3.1 pour la planification d'une trajectoire qui permet d'éviter les obstacles. Le champ de potentiel attractif est généré par la position cible correspondant au plus loin point de la scène accessible par le robot. Les champs de potentiel répulsifs sont générés par les obstacles. On a choisi une distance d'influence ρ égale à 8 cm.

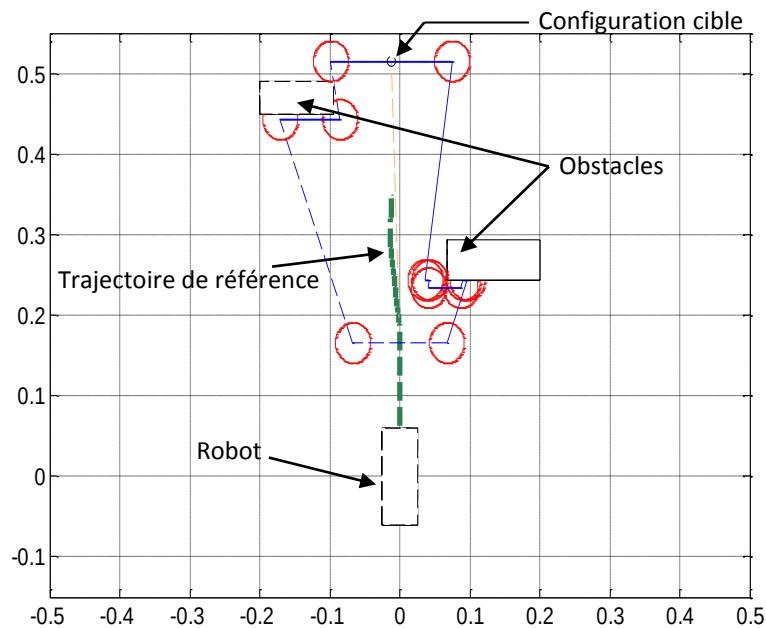


Figure 5.20 : planification de trajectoire par la méthode de champs de potentiel

5.7.4. Simulation du suivi de la trajectoire de référence

Pour réaliser le suivi de la courbe décrite précédemment, nous appliquons au robot la technique de commande par suivi de chemin, où nous avons choisi une vitesse d'avance de 0.3 m/s , ce qui permettra un temps d'exécution de moins de 2 secondes.

Les résultats de simulation sont les suivants :

Nous présentons d'abord la trajectoire exécutée par le robot avec les erreurs de suivi selon x, y et θ par rapport à la courbe de référence (figure 5.21).

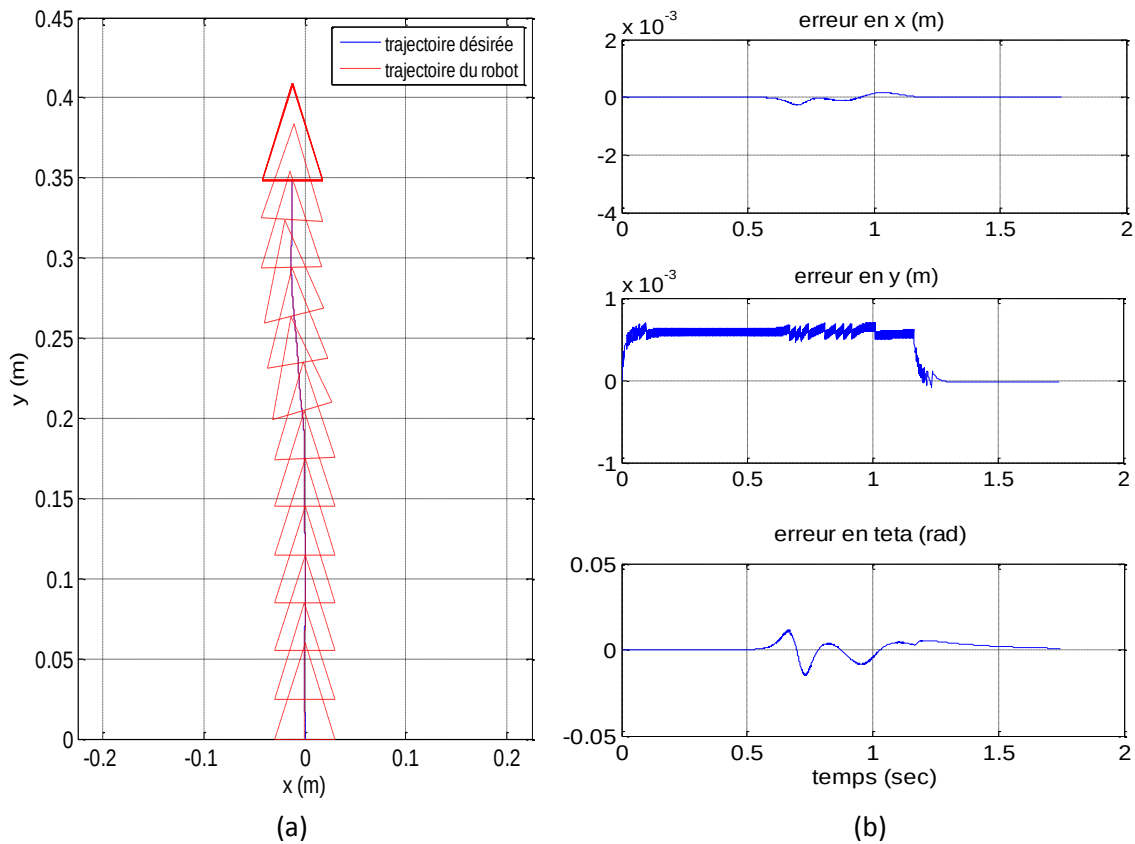


Figure 5.21 : simulation du suivi de ligne. (a) trajectoire du robot. (b) les erreurs de suivi

Le suivi est assuré avec une bonne précision, de l'ordre de 10^{-3} m pour la position, et de $2 \times 10^{-2} \text{ rad}$ pour l'orientation.

Les différentes entrées de commande des modèles cinématique et dynamique du robot (v, ω, eam, ead) sont données dans la figure suivante :

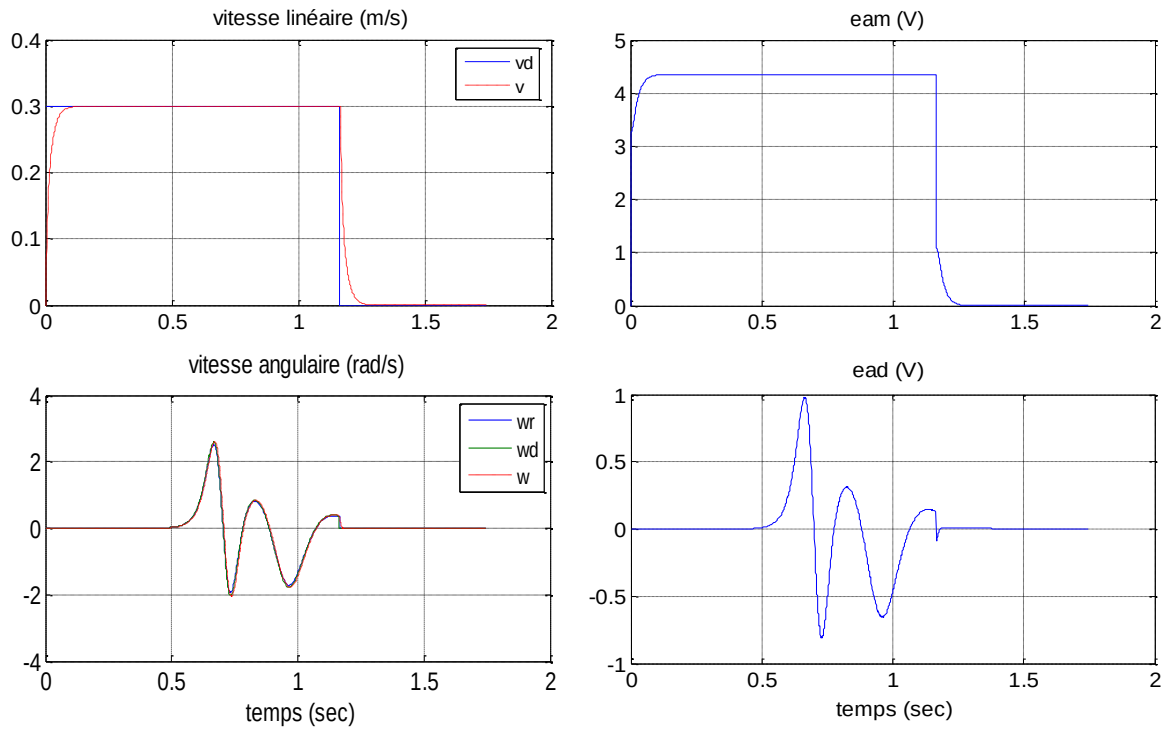


Figure 5.22 : allure des entrées de commande du système

Finalement, nous présentons les signaux électriques appliqués sur chacun des moteurs du robot pour générer ces commandes :

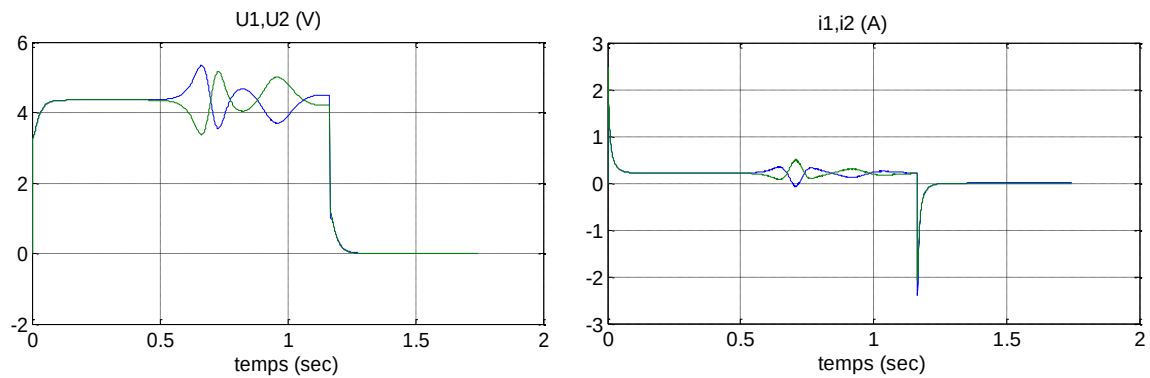


Figure 5.23 : allure des signaux électriques appliqués aux moteurs

Toutes les entrées de commandes sont bien situées dans les limites des valeurs de fonctionnement normal du système. Les pics de courant correspondent aux deux instants de démarrage et de freinage du robot.

5.8. Conclusion

Dans ce chapitre, nous avons présenté les résultats relatifs aux applications choisies, où les techniques de traitement d'image ont été utilisées afin de calculer la trajectoire appropriée à chaque tâche. Cette partie expérimentale a été réalisée par le biais d'une webcam, dont nous avons d'abord effectué le calibrage et la modélisation. Cette dernière opération s'est vue facilitée du fait de la disposition particulière que nous avons adoptée pour la caméra. En outre, l'expérimentation a été réalisée en milieu intérieur dans des conditions d'éclairage favorables.

Ayant obtenu la trajectoire de référence, s'en est suivit l'application des techniques de commande en boucle fermée pour en assurer l'exécution. Dans ce cas, on a simulé le comportement du robot commandé par la technique du suivi de chemin, qui a donné des résultats satisfaisants.

Conclusion générale

Nous nous sommes intéressés dans ce mémoire à la problématique de la commande par vision d'un robot mobile non holonome de type unicycle, où nous nous sommes attachés plus précisément à deux types d'applications, à savoir le suivi d'une ligne tracée sur le sol et l'évitement d'obstacles qui s'avère être incontournable lorsqu'il s'agit de faire de la navigation.

A cet effet, nous avons dû faire appel simultanément aux techniques de commande en boucle fermée, aux techniques de vision artificielle, et aux techniques de planification de trajectoires.

Dans le premier chapitre, nous avons présenté le contexte dans lequel s'inscrit notre travail, en mettant le point sur la problématique traitée, où nous avons mis en évidence la difficulté et le nombre assez important des contraintes y afférentes, notamment la nature non holonome du système considéré. Enfin nous avons présenté l'approche proposée pour résoudre cette problématique qui est une approche hiérarchique itérative.

Le chapitre 2 a constitué un état de l'art sur les robots mobiles non holonomes et leur commande, où dans un premier temps, on s'est intéressé à la présentation des bases théoriques relatives aux systèmes non holonomes, et plus particulièrement les plateformes mobiles pour lesquelles nous avons dérivé les modèles cinématiques des types les plus répandus, à savoir l'unicycle, le tricycle et la voiture. Ensuite nous avons effectué un tour d'horizon sur les stratégies et techniques proposées dans la littérature pour résoudre la problématique de la commande de ce type de robots.

Dans le chapitre 3, nous avons traité l'aspect commande du robot, où nous avons choisi les deux stratégies de commande adaptées à notre contexte, à savoir : la poursuite de trajectoire et le suivi de chemin, puisqu'elles permettent de maîtriser le comportement du robot tout au long de la tâche, chose qu'on ne peut obtenir en utilisant la stratégie de stabilisation de configurations fixes.

Dans ce cadre, nous avons proposé plusieurs techniques de commande non linéaires, comme suit :

- Poursuite de trajectoire :
 - linéarisation par bouclage dynamique ;
 - commande par retour d'état non linéaire non stationnaire ;
 - commande avec observateur de l'orientation du robot
- Suivi de chemin :
 - commande non linéaire avec repère du robot partiellement lié au chemin.

Les simulations effectuées sous MATLAB ont permis de valider les différentes techniques présentées, où nous avons trouvé que la technique de linéarisation par bouclage combinée à un proportionnel dérivé offrait un meilleur temps de convergence vers la trajectoire de référence avec une précision acceptable, cependant elle s'avère être très sensible vis-à-vis des bruits sur le système. En contrepartie, la technique de commande non stationnaire basée sur le critère de stabilité de Lyapunov a permis d'obtenir la meilleure précision de poursuite mais avec un temps de convergence plus important. Finalement, l'étude comparative a guidé notre choix pour la technique la plus adaptée à nos applications, qui a porté sur la technique entrant dans le cadre du suivi de chemin, qui offrait une bonne précision du suivi, et le contrôle sur la vitesse d'avance du robot.

Dans le chapitre 4, nous avons introduit quelques notions fondamentales sur la perception basée sur la vision par ordinateur, notamment celles relatives aux caméras et leur modélisation, ainsi que les différentes techniques de traitement d'images permettant l'interprétation de l'information visuelle. Aussi, nous avons présenté les différentes méthodes utilisées pour la planification de trajectoire en robotique mobile dans le cadre de l'évitement réactif d'obstacles, en l'occurrence la méthode des champs de potentiel, que nous avons utilisé pour notre application.

Le chapitre 5 a été consacré à la mise en œuvre de l'approche adoptée pour la navigation visuelle, représentée par les deux applications choisies, où nous avons d'abord commencé par donner une description du système complet (robot + caméra), puis nous avons présenté en détail la chaîne de traitement complète relative à la planification de trajectoire par vision, depuis l'acquisition d'image de la scène observée, jusqu'à l'obtention de la trajectoire de référence dans l'espace 3D. Par la suite, cette chaîne de traitement a été appliquée pour les deux cas de figures préconisés dans notre approche, à savoir, le suivi de ligne et la navigation avec évitement d'obstacle. Cette partie a été réalisée sous MATLAB, moyennant une caméra de type webcam. Enfin, nous avons utilisé la technique de commande par suivi de chemin pour assurer au robot l'exécution de la trajectoire ainsi obtenue, laquelle a été testée et validée en environnement de simulation MATLAB.

Pour ce qui est des perspectives de recherche envisageables en complément à ce travail, on peut prévoir :

- L'extension des résultats sur des plateformes de type plus complexe, ex. voiture avec remorque ;
- L'utilisation de lois de commandes robustes (modes glissants, logique floue, etc.) pour s'affranchir des phénomènes non modélisés, tel que le glissement ;
- La proposition d'une approche de navigation visuelle en temps réel, basée sur le traitement du flux optique.

Annexe A

Techniques de linéarisation par bouclage

A.1. Principe de la linéarisation par bouclage :

Considérons un système non linéaire décrit par l'expression suivante :

$$\begin{cases} \dot{x} = f(x) + g(x)u \\ y = h(x) \end{cases} \quad (\text{A.1})$$

dont le nombre d'entrées et sorties sous tous deux égaux à m .

L'idée de la linéarisation par bouclage est de boucler le système par une commande du type $u = r(x, v)$ où v est la nouvelle entrée, aussi de dimension m .

Afin d'effectuer ce bouclage, il va falloir exprimer les dérivées successives de chacune des sorties y_i en fonction de l'état et de l'entrée. On s'arrête de dériver y_i dès que les entrées commencent à apparaître dans les expressions des dérivées. Nous obtenons ainsi une équation du type :

$$\begin{pmatrix} y_1^{(k_1)} \\ \vdots \\ y_m^{(k_m)} \end{pmatrix} = A(x)u + b(x) \quad (\text{A.2})$$

où k_i désigne le nombre de fois qu'il a fallu dériver y_i pour voir apparaître une entrée. Dans ce cas, k_i est appelé degré relatif correspondant à la sortie y_i .

Sous hypothèse que la matrice $A(x)$ soit inversible, le bouclage suivant :

$$u = A^{-1}(x)(v - b(x)) \quad (\text{A.3})$$

forme un système linéaire de m entrées à m sorties décrit par les équations différentielles :

$$\begin{cases} y_1^{(k_1)} = v_1 \\ \vdots \\ y_m^{(k_m)} = v_m \end{cases} \quad (\text{A.4})$$

Ce système linéaire est complètement découplé (chaque entrée v_i agit sur une seule sortie y_i).

A.2. Linéarisation par retour d'état d'un système monovarié :

$$\text{Soit le système (A.1), avec } x \in \mathbb{R}^n, y \in \mathbb{R}, u \in \mathbb{R}. \quad (\text{A.5})$$

A.2.1. Linéarisation exacte entrée/état :

Le but de la méthode est de trouver pour le système précédemment défini un bouclage de la forme :

$$u = \alpha(x) + \beta(x)v \quad (\text{A.6})$$

et un changement de variable :

$$z = \phi(x) \quad (\text{A.7})$$

de façon à obtenir des relations linéaires entre le nouvel état z et la nouvelle entrée v calculée à partir de la dynamique désirée du système.

Définition A.1 :

Un système monovarié de la forme (A.5) est linéarisable au sens entrée/état s'il existe une région $\Omega \subset \mathbb{R}^n$, un difféomorphisme $\phi : \Omega \rightarrow \mathbb{R}^n$ et une loi de commande non linéaire $u = \alpha(x) + \beta(x)v$ telles que les nouvelles variables d'état $z = \phi(x)$ et la nouvelle entrée v forme un système linéaire invariant dans le temps de la forme canonique suivante :

$$\dot{z} = A(z) + Bv \quad (\text{A.8})$$

$$\text{où } A(n \times n) = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ et } B(n \times 1) = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix}$$

L'équation (A.8) est obtenue dans le cas où le degré relatif du système (A.5) est égal à la dimension de l'espace d'état, autrement dit dans les nouvelles coordonnées, l'équation d'état prend la forme normale :

$$\begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \vdots \\ \dot{z}_n = b(z) + a(z)u \end{cases} \quad (\text{A.9})$$

Le nouveau vecteur de coordonnées z et la loi de commande non linéaire u s'appellent respectivement l'état linéarisant et la commande linéarisante du système.

Lorsque la sortie y du système est une fonction non linéaire de la nouvelle variable d'état z , le schéma-bloc du système linéarisé entrée/état est le suivant (figure A.1) :

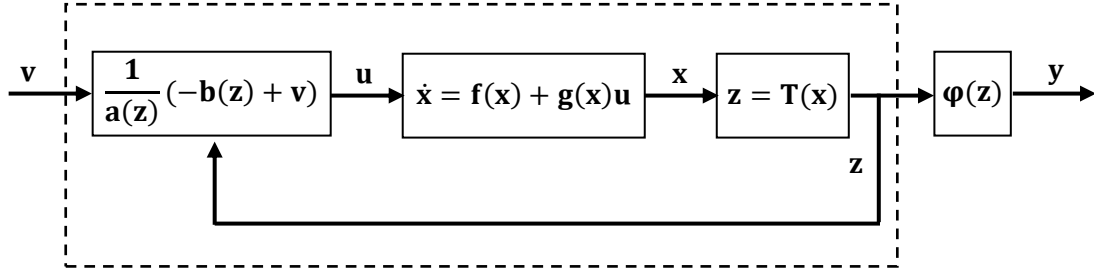


Figure A.1 : schéma-bloc de linéarisation entrée/état d'un système SISO non linéaire

A.2.1.1. Conditions pour une linéarisation exacte entrée/état :

Les conditions à satisfaire pour qu'un système soit linéarisable par retour d'état sont fournies dans le théorème suivant [70].

Théorème A.1 :

Le système (A.1) est linéarisable entrée/état si et seulement si il existe une région $\Omega \subset \mathbb{R}^n$ telle que les conditions suivantes soient satisfaites :

- 1- Les champs de vecteurs $\{g, ad_f g, \dots, ad_f^{n-1} g\}$ sont linéairement indépendants sur Ω
- 2- L'ensemble $\{g, ad_f g, \dots, ad_f^{n-2} g\}$ est involutif.

La première condition peut être interprétée comme une simple condition de commandabilité du système (A.1). Et la deuxième condition signifie que le crochet de Lie de deux champs de vecteurs quelconques de l'ensemble $\{g, ad_f g, \dots, ad_f^{n-2} g\}$ peut s'exprimer comme une combinaison linéaire des autres champs de vecteurs de cet ensemble.

A.2.1.2. Etapes de linéarisation entrée/état :

La linéarisation au sens entrée/état est réalisée en quatre étapes. La première consiste à construire les champs de vecteurs $\{g, ad_f g, \dots, ad_f^{n-1} g\}$. La deuxième consiste à vérifier si les conditions de commandabilité et d'involutivité sont satisfaites. La troisième consiste à choisir le premier élément z_1 du nouveau vecteur d'état z de façon à respecter les conditions :

$$\nabla_{z_1} ad_f^i g = 0 \quad i = 1, \dots, n-2 \quad (A.10)$$

$$\nabla_{z_1} ad_f^{n-1} g \neq 0 \quad (A.11)$$

La quatrième étape consiste à construire le nouveau vecteur d'état $z = (z_1 L_f z_1 L_f^{n-1} z_1)^T$ ainsi que la commande u définie en (A.6), tel que :

$$\alpha(x) = -\frac{L_f^n z_1}{L_g L_f^{n-1} z_1}, \quad \beta(x) = -\frac{1}{L_g L_f^{n-1} z_1}$$

A.2.2. Linéarisation entrée/sortie :

Lorsque la linéarisation au sens entrée/état n'est pas réalisable, il est possible de faire une linéarisation au sens entrée/sortie en dérivant successivement la sortie du système jusqu'à retrouver l'entrée u . En posant la nouvelle entrée v égale à la dérivée la plus élevée de la sortie, on obtient une relation linéaire entre y et v .

A.2.2.1. Génération de la relation entrée/sortie linéaire :

On dérive la sortie y par rapport au temps pour obtenir :

$$y^{(1)} = L_f h(x) + L_g L_f^0 h(x) u$$

Si $L_g L_f^0 h(x) \neq 0$ on s'arrête, sinon on dérive encore jusqu'à obtenir :

$$y^{(r)} = L_f^r h(x) + L_g L_f^{r-1} h(x) u \quad \text{avec} \quad L_g L_f^{r-1} h(x) \neq 0 \quad (\text{A.12})$$

Où $L_f h$ et $L_g h$ sont les dérivées de Lie de h le long de f et g respectivement. L'entier r est degré relatif du système. Il est le plus petit entier qui vérifie :

$$L_g L_f^i h(x) = 0 \quad i = 0, \dots, r-2 \quad (\text{A.13})$$

$$L_g L_f^{r-1} h(x) \neq 0 \quad (\text{A.14})$$

et il correspond au nombre de dérivations par rapport au temps de la sortie pour retrouver l'entrée u .

Si on applique la loi de commande de la forme :

$$u = \frac{1}{L_g L_f^{r-1} h(x)} (-L_f^r h(x) + v) \quad (\text{A.15})$$

au système (A.5) on obtient une relation linéaire entre la sortie et l'entrée v :

$$y^{(r)} = v \quad (\text{A.16})$$

A.2.2.2. Forme normale et dynamique interne :

Si l'entier r est égal à l'ordre du système ($r = n$) alors la linéarisation au sens entrée/sortie résulte en une linéarisation complète au sens entrée/état.

Dans ce cas la forme normale se présente comme suit :

$$\begin{cases} z_1 = \phi_1(x) = h(x) \\ z_2 = \phi_2(x) = L_f h(x) \\ z_3 = \phi_3(x) = L_f^2 h(x) \\ \dots \\ z_n = \phi_n(x) = L_f^{n-1} h(x) \end{cases} \quad (\text{A.17})$$

et l'équation d'état est celle donnée dans (A.9).

Dans le cas où $r < n$, la linéarisation est partielle et seules la sortie y ainsi que ses $(r - 1)$ dérivées successives sont linéairement liées avec la nouvelle entrée v . Cette partie du système est appelée dynamique externe.

$$\begin{cases} z_1 = \phi_1(x) = h(x) \\ z_2 = \phi_2(x) = L_f h(x) \\ z_3 = \phi_3(x) = L_f^2 h(x) \\ \dots \\ z_r = \phi_r(x) = L_f^{r-1} h(x) \end{cases} \quad (\text{A.18})$$

Les $(n - r)$ variables restantes du système sont rendues inobservables par la linéarisation, et forment ce qu'on appelle dynamique interne ou dynamique des zéros, dont la stabilité est étroitement liée à celle du système.

Dans ce cas, il est toujours possible de trouver $(n - r)$ fonctions $\phi_{r+1}, \dots, \phi_n$ telles que l'application $\phi(x) = (\phi_1(x) \dots \phi_n(x))$ soit un difféomorphisme. Pour plus de détails voir C. Byrnes et A. Isidori [10].

A.3. Extension des résultats au cas multivariable :

Dans cette section, nous décrivons la théorie de linéarisation par retour d'état appliquée aux systèmes multivariables.

Soit le système multivariable défini par :

$$\dot{x} = f(x) + \sum_{i=1}^m g_i(x)u_i \quad (\text{A.19})$$

avec $x \in \mathbb{R}^n$, $f, g_1, \dots, g_m$ des champs de vecteurs C définis dans $\mathbb{R}^n$.

Définition A.2 :

Soit un système de la forme (A.19), avec un vecteur de sortie $y = (h_1(x)h_2(x)h_m(x))^T$.

Ce système possède les degrés relatifs $r_1, \dots, r_m$ au voisinage Ω d'un point d'opération x_0 si :

- $\sum_{j=1}^m L_{g_j} L_f^k h_i u_j = 0$ pour $0 \leq k \leq r_i - 1$ et $1 \leq i, j \leq m$ $\forall x \in \Omega$
- La matrice de dimension $(m \times m)$

$$A(x) = \begin{pmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & L_{g_2} L_f^{r_1-1} h_1(x) & \dots & \dots & L_{g_m} L_f^{r_1-1} h_1(x) \\ L_{g_1} L_f^{r_2-1} h_2(x) & L_{g_2} L_f^{r_2-1} h_2(x) & & & L_{g_m} L_f^{r_2-1} h_2(x) \\ \dots & \dots & \dots & \dots & \dots \\ L_{g_1} L_f^{r_m-1} h_m(x) & L_{g_2} L_f^{r_m-1} h_m(x) & \dots & \dots & L_{g_m} L_f^{r_m-1} h_m(x) \end{pmatrix} \quad (\text{A.20})$$

est inversible $\forall x \in \Omega$.

Le degré relatif r_i est égal au nombre de dérivations nécessaires de la sortie y_i pour retrouver au moins une entrée u_j . Ceci est traduit par :

$$\sum_{j=1}^m L_{g_j} L_f^k h_i u_j = 0 \quad \text{pour } 0 \leq k \leq r_i - 1$$

et $\exists j$ tel que $L_{g_j} L_f^{r_i-1} h_i(x) \neq 0$

A.3.1. Linéarisation exacte entrées/état :

Lemme A.1 :

Le système (A.19) est linéarisable au sens entrées/état si et seulement si il existe un voisinage Ω et m fonctions réelles définies dans Ω tel que le système :

$$\begin{cases} \dot{x} = f(x) + \sum_{i=1}^m g_i(x)u_i \\ y = (h_1(x)h_2(x) \dots h_m(x))^T \end{cases} \quad (\text{A.21})$$

possède un degré relatif global $r = r_1 + r_2 + \dots + r_m = n$.

S'il est possible de trouver un vecteur de sortie tel que le degré relatif du système $r = n$ alors on peut définir un difféomorphisme :

$$z = \Phi(x) \quad \text{avec} \quad \Phi(0) = 0 \quad (\text{A.22})$$

et un vecteur de commande de la forme :

$$u_i = \alpha_i(x) + \sum_{j=1}^m \beta_{ij}(x) v_j \quad \text{avec} \quad \alpha(0) = 0 \quad i = 1, \dots, m \quad (\text{A.23})$$

Telle que si appliquée au système (A.21), le nouveau vecteur d'état z et les nouvelles entrées v forment un système linéaire de la forme (A.17).

$$\begin{aligned} \text{Avec} \quad & A(n \times n) = \text{diag}(a_1, \dots, a_m) \\ & B(n \times m) = \text{diag}(b_1, \dots, b_m) \\ & v = (v_1 \dots v_m)^T \end{aligned}$$

où :

$$a_j = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}_{(k_j \times k_j)}, \quad b_j = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \\ 0 \end{pmatrix}_{(k_j \times 1)}$$

k_j est l'indice caractéristique associé à la sortie y_j . Si on choisit $y_j = (1 \ 0 \ \dots \ 0)z_j$, les indices caractéristique k_j deviennent égaux aux degrés relatifs r_j , et le degré relatif global du système devient $k = k_1 + k_2 + \dots + k_m = n$.

A.3.2. Linéarisation entrées/sorties

Lorsque le vecteur de sortie est déjà choisi, on vérifie si le degré relatif global r est égal à l'ordre du système n . Si tel est le cas, le système est linéarisable au sens entrées/état, sinon il est possible de réaliser une linéarisation partielle au sens entrées/sorties.

Soit le système multivariable donné par (A.19), avec les degrés relatifs $r_1, r_2, \dots, r_m$.

On peut écrire :

$$y_i^{(r_i)} = L_f^{r_i} h_i + \sum_{i=1}^m L_{g_i} L_f^{r_i-1} h_i u_i \quad (\text{A.24})$$

$$\begin{pmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{pmatrix} = \begin{pmatrix} L_f^{r_1} h_1 \\ \vdots \\ L_f^{r_m} h_m \end{pmatrix} + A(x) \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} \quad (\text{A.25})$$

Si la matrice $A(x)$ est non singulière, la loi de commande linéarisante (A.23) est telle que :

$$\alpha(x) = -A(x)^{-1} \begin{pmatrix} L_f^{r_1} h_1 \\ \vdots \\ L_f^{r_m} h_m \end{pmatrix}, \quad \beta(x) = A(x)^{-1}, \quad u = \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix}, \quad v = \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix}$$

L'application de cette loi de commande permet d'avoir une relation entrées/sorties linéaire :

$$\begin{pmatrix} y_1^{(r_1)} \\ \vdots \\ y_m^{(r_m)} \end{pmatrix} = \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix}$$

Si $r = n$, le système est donc linéarisable au sens entrées/état et le nouveau vecteur z d'état est de la forme :

$$z = (z_1 \ z_2 \ \dots \ z_n)^T, \quad \text{avec} \quad z_i = (z_{i_1} \ \dots \ z_{i_{r_i}})^T = (y_i \ \dots \ y_i^{r_i-1})^T \quad i = 1, \dots, n$$

Si $r < n$, le système est alors partiellement linéarisable et il existe des dynamiques internes rendues inobservables par la linéarisation. Comme pour le cas monovarié, le vecteur $(z_1 \ z_2 \ \dots \ z_m)^T$ forme la partie linéaire du nouveau vecteur d'état du système. Pour les $(n - r)$ variables restantes du système $(z_{r+1} \ \dots \ z_n)^T$, il est toujours possible de les trouver de façon à former une transformation de coordonnées $\Phi(x) = z$.

De plus, si la distribution $\{g_1, \dots, g_m\}$ est involutive dans Ω , alors chaque élément z_i avec $i = r + 1, \dots, n$ peut être choisi pour satisfaire la condition :

$$L_{g_j} z_i(x) = 0 \quad i = r + 1, \dots, n \quad j = 1, \dots, m \quad \forall x \in \Omega \quad (\text{A.26})$$

La description du système dans ses nouvelles coordonnées devient alors :

$$\left\{ \begin{array}{l} z_{1_1} = z_{1_2} \\ \vdots \\ z_{1_{r_1}} = L_f^{r_1} h_1 + \sum_{j=1}^m L_{g_j} L_f^{r_1-1} h_1 u_j \\ \vdots \\ z_{m_1} = z_{m_2} \\ \vdots \\ z_{m_{r_m}} = L_f^{r_m} h_m + \sum_{j=1}^m L_{g_j} L_f^{r_m-1} h_m u_j \end{array} \right. \quad (\text{A.27})$$

Annexe B

Modélisation dynamique du robot unicycle

B.1. Description du modèle :

Le modèle dynamique adopté pour l'unicycle repose sur l'utilisation d'un système de motorisation identique pour chacune des deux roues motrices du robot basé sur un moteur à courant continu, dont la représentation au niveau d'une roue est donnée dans la figure ci-après.

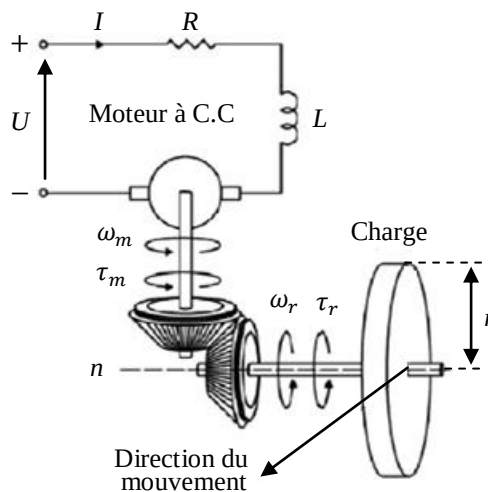


Figure B. 1 : représentation de la motorisation pour une roue du robot

Où :

U : tension d'alimentation du moteur (V)

I : courant (A)

n : rapport de réduction

r : rayon de la roue (m)

ω_m : vitesse de rotation du moteur (rad/sec)

τ_m : couple moteur (N.m)

ω_r : vitesse de rotation de la roue (rad/sec)

τ_r : couple au niveau de la roue (N.m)

B.2. Modélisation :**B.2.1. Equations du moteur :**

$$\begin{cases} U = RI + K_e \omega_m \\ \tau_m = J \dot{\omega}_m + \tau_f \end{cases} \quad (\text{B.1})$$

Avec :

K_e : constante électromagnétique du moteur

J : moment d'inertie vu coté moteur

τ_f : couple résistant

B.2.2. Au niveau d'une roue :

$$\begin{cases} U = \frac{R}{nK_t\eta} \tau_r + \frac{nK_e}{\alpha} \omega_r \\ \tau_r = J_r \dot{\omega}_r + \tau_f \end{cases} \quad (\text{B.2})$$

Où :

$$I = \frac{\tau_r}{K_t}$$

$$K_t = K_e = K$$

η : coefficient de perte de couple

α : coefficient de perte en vitesse de rotation

$J_r = \frac{1}{2} \left(\frac{M}{2} \right) r^2$: moment d'inertie au niveau de la roue, avec M masse du robot.

$\tau_f = r \cdot F_f = r \cdot \frac{Mg}{2} \mu$, avec μ coefficient de frottement roue-sol.

En régime permanent : $\tau_r = \tau_f$

$$\Rightarrow \omega_{r \max} = \frac{\alpha}{nK} \left(U - \frac{R}{nK\eta} \tau_f \right) \quad (\text{B.3})$$

A l'arrêt : $\omega_r = 0$

$$\Rightarrow \tau_{r \max} = \frac{nK}{R} \eta U \quad (\text{B.4})$$

D'où :

$$v_{r \max} = r \omega_{r \max} \quad (\text{B.5})$$

B.2.3. Modèle du robot :

Les équations précédentes établies pour les deux roues motrices du robot, notées g (gauche) et d (droite) donnent :

- Pour la roue gauche :

$$\begin{cases} U_g = \frac{R}{nK\eta} \tau_g + \frac{nK_e}{\alpha} \omega_g \\ \tau_g = J_r \dot{\omega}_g + \tau_f \end{cases}$$

$$\text{D'où : } U_g = \frac{RJ_r}{nK\eta} \dot{\omega}_g + \frac{nK}{\alpha} \omega_g + \frac{RJ_r}{nK\eta} \tau_f$$

En posant : $U'_g = U_g - \frac{RJ_r}{nK\eta} \tau_f$, on obtient une représentation de la partie linéaire du fonctionnement du moteur, où le terme constant $\frac{RJ_r}{nK\eta} \tau_f$, représente la tension nécessaire pour vaincre le couple résistant au démarrage.

Ce qui permet d'écrire :

$$U'_g = \frac{RJ_r}{nK\eta} \dot{\omega}_g + \frac{nK}{\alpha} \omega_g \quad (\text{B.6})$$

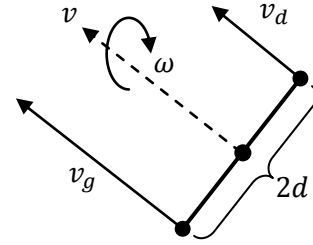
- Pour la roue droite :

$$\begin{cases} U_d = \frac{R}{nK\eta} \tau_d + \frac{nK_e}{\alpha} \omega_d \\ \tau_r = J_r \dot{\omega}_d + \tau_f \end{cases}$$

Avec des développements similaires que précédemment, on obtient :

$$U'_d = \frac{RJ_r}{nK\eta} \dot{\omega}_d + \frac{nK}{\alpha} \omega_d \quad (\text{B.7})$$

D'autre part, nous avons :



La vitesse linéaire du robot v est donnée par :

$$v = \frac{1}{2}(v_g + v_d) = \frac{r}{2}(\omega_g + \omega_d) \quad (\text{B.8})$$

Et la vitesse angulaire du robot ω est donnée par :

$$\omega = \frac{1}{2d}(v_g - v_d) = \frac{r}{2d}(\omega_g - \omega_d) \quad (\text{B.9})$$

Maintenant, on calcule $\frac{(B.6)+(B.7)}{2}$:

$$\begin{aligned} \frac{U'_g + U'_d}{2} &= \frac{RJ_r}{nK\eta} \left(\frac{\dot{\omega}_g + \dot{\omega}_d}{2} \right) + \frac{nK}{\alpha} \left(\frac{\omega_g + \omega_d}{2} \right) \\ Eam &= \frac{RJ_r}{nK\eta r} \dot{v} + \frac{nK}{\alpha} v \end{aligned} \quad (\text{B.10})$$

Eam représentant la tension moyenne des deux actionneurs.

En multipliant par $(\frac{2nK\eta}{Rr})$, on obtient :

$$\left(\frac{2J_r}{r^2}\right)\dot{v} = \frac{2nK\eta}{Rr}Eam - \frac{2n^2K^2\eta}{Rr^2\alpha}v \quad (\text{B. 11})$$

$$M\dot{v} = K_mEam - K_vv \quad (\text{B. 12})$$

De même, on calcule $\frac{(B.6)-(B.7)}{2}$:

$$\frac{U'_g - U'_d}{2} = \frac{RJ_r}{nK\eta} \left(\frac{\dot{\omega}_g - \dot{\omega}_d}{2}\right) + \frac{nK}{\alpha} \left(\frac{\omega_g - \omega_d}{2}\right)$$

$$Ead = \frac{RJ_r d}{nK\eta r} \dot{\omega} + \frac{nKd}{\alpha r} \omega \quad (\text{B. 13})$$

Ead représentant la tension différentielle des deux actionneurs.

En multipliant par $(\frac{2nKd\eta}{Rr})$, on obtient :

$$\left(\frac{2J_r d^2}{r^2}\right)\dot{\omega} = \frac{2nK\eta}{Rr}Ead - \frac{2n^2K^2\eta}{Rr^2\alpha}\omega \quad (\text{B. 14})$$

$$J_{eq}\dot{\omega} = K_d Ead - K_\omega \omega \quad (\text{B. 15})$$

B.2.4. Application numérique :

Paramètres du robot	
M	2 Kg
r	0.02 m
d	0.025 m
μ	0.5
Paramètres des moteurs	
U	12 V
R	1.33 Ω
K_e	0.046
K_t	0.046
n	5.9
η	0.772
α	0.935

Bibliographie

- [1] Alexander, J.C. and Maddocks J.H. « On the kinematics of wheeled mobile robots », in *International Journal of Robotics and Res* vol. 8, no. 5, pp. 15-27, 1989.
- [2] Astolfi, A. « Discontinuous control of nonholonomic systems ». *Systems and Control Letters*, 27:37–45, 1996.
- [3] Avina, J. G. « *Navigation visuelle d'un robot mobile dans un environnement d'extérieur semi-structuré* ».Thèse de Doctorat, Institut National Polytechnique de Toulouse, France, 2005.
- [4] Barate, M. R. « *Apprentissage de fonctions visuelles pour un robot mobile par programmation génétique* ».Thèse de Doctorat, Université Pierre et Marie Curie, France, 2008.
- [5] Belkacem, S. « *Contribution à la commande directe du couple de la machine à induction* ». Thèse de Doctorat, Université de Batna, Algérie, 2011.
- [6] Bennani, M. K. and Rouchon, P. « Robust stabilization of flat and chained systems ». In *European Control Conference (ECC)*, pp 2642–2646, 1995.
- [7] Bloch, A. and McClamroch, N. « Control of mechanical systems with classical nonholonomic constraints ». In *IEEE Conference on Decision and Control*, Tampa,USA, 1989.
- [8] Bloch, A., Reyhanoglu, M., and McClamroch, N. « Control and stabilization of nonholonomic dynamic systems ». *IEEE Transactions on Automatic Control*, 37:1746–1757, 1992.
- [9] Brockett, R. « Asymptotic stability and feedback stabilization ». In *Brockett, R., Millman, R., et Sussmann, H., editors, Differential geometric control theory*, pp. 181–195. Boston, MA : Birkhauser, 1983.
- [10] Byrnes C. and Isidori A., « Asymptotic expansions, root-loci and the global stability of nonlinear feedback systems », *Algebraic and Geometric Methods in control theory*, Eds Reidel, Dordrecht- Boston, 1986.
- [11] Canny, 1986.
- [12] Canudas-De-Wit, C. and Sordalen, O. « Exponential stabilization of mobile robots with nonholonomic constraints”. *IEEE Transactions on Automatic Control*, 37:1791–1797, 1992.
- [13] Canudas-De-Wit, C., Siciliano, B., and Bastin, G. « *Theory of Robot Control* ». Springer-Verlag, Berlin, 1996.

-
- [14] Cazaurang, F. « *Commande robuste des systèmes plats, application à la commande d'une machine synchrone* ». Thèse de Doctorat, Université de Bordeaux I, France, 1997.
 - [15] Charlet, B. « *Quelques problèmes de stabilisation robuste des systèmes non linéaires* », Thèse de doctorat, Ecole des Mines de Paris, 1989.
 - [16] Chow, W. L. « Uber systeme von linearen partiellen differential-gleichungen erster ordnung ». *Math. Ann.*, 117:98–105, 1939.
 - [17] Coron, J.-M. « Global asymptotic stabilization for controllable systems without drift ». *Mathematics of Control, Signals and Systems*, 5:295–312., 1992.
 - [18] Coron, J.-M. and d'Andrea Novel, B. « Smooth stabilizing time-varying control laws for a class of nonlinear systems ». In *IFAC Nonlinear Control Systems Design Symp. (NOLCOS)*, pages 649–654, 1992.
 - [19] Falb P. L. and Wolovitch W.A., « Decoupling in the Design and Synthesis of Multivariable Control Systems ». *I.E.E.E. Transactions on Automatic Control*, AC-12, pp. 651-659, 1967.
 - [20] Feiten, W., Bauer, R. and Lawitzky, G. « Robust obstacle avoidance in unknown and cramped environments ». *International Conference on Robotics and Automation*, 1994.
 - [21] Fierro, R. and Lewis, F. L. « Control of a nonholonomic mobile robot: Backstepping the kinematics into dynamics ». *Journal of Robotic Systems*, 14(3): 149-163, 1997.
 - [22] Fliess M., Lévine J., Martin Ph. and Rouchon P., « Design of trajectory stabilizing feedback for driftless flat systems ». In *European Control Conference*, pp.1882-1887, Rome, 1995.
 - [23] Fliess M., Lévine J., Martin Ph. and Rouchon P., « Sur les systèmes non linéaires différentiellement plats ». *C.R Acad. Sci. Paris*, I-315, pp.619-624,1992.
 - [24] Fliess, M., Levine, J., Martin, P., and Rouchon, P. « Flatness and defect of nonlinear systems: Introductory theory and examples ». *International Journal of Control*, 61:1327–1361, 1995.
 - [25] Fofi, D. « *Navigation d'un véhicule Intelligent à l'aide d'un capteur de vision en lumière structurée et codée* ». Thèse de Doctorat, Université Picardie Jules Verne, France, 2001.
 - [26] Fox, D., Burgard, W. and Thrun, S. « The Dynamic Window Approach to Collision Avoidance ». *IEEE Robotics and Automation Magazine*, vol. 4, no 1, pages 23– 33, March 1997.
 - [27] Hernandez, A. M. « *Vision dynamique pour la navigation d'un robot mobile* ». Thèse de doctorat, Institut National Polytechnique, Toulouse, France, 2004. (04)
 - [28] Herrera, I. « *Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service* ». Thèse de doctorat, Université Toulouse III, France, 2007.
 - [29] Hunt, L. R., Su R. and Meyer G. « Design for multi-input nonlinear systems Differential geometric control theory », *R. Brockett, R. Mllman, H. Sussmann eds*, pp. 268-298, Birkäuser, 1983.
 - [30] Isidori A., Krener A.J., Gori-Giorgi C., and Monaco S., « Nonlinear decoupling via feedback », *I.E.E.E. Transactions on Automatic Control*, AC-26, pp. 331- 345, 1981.
 - [31] Isidori, A. « *Nonlinear control systems* ». Springer-Verlag, Berlin, 2nd edition, 1989.

-
- [32] Jahoo Koo, Seungjoon Choi, Daehyun Ji and Sangchul Won, « Observer-Based Trajectory Tracking Control for a Wheeled Mobile Robot », *Asian Control Conference*, 2009.
 - [33] Jakubczyk, B. and Respondek, W. « On Linearization of control systems ». *Bull. Acad. Pol. Sci, Ser. Sci. Math*, Vol 28, 9-10, pp. 517-522, 1980.
 - [34] Jakubiak, E. Lefeber, K. Tchon and Nijmeijer, H. « Two observer based Tracking Algorithms for a Unicycle Mobile Robot », *International Journal of Mathematical Computational Science*, Vol. 12, No. 4, pp. 513-522, 2002.
 - [35] Jiang, Z.-P. and Nijmeijer, H. « Tracking control of mobile robots: a case study in backstepping ». *Automatica*, 33(7): 1393-1399, 1997.
 - [36] Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T. « A stable tracking control method for an autonomous mobile robot ». In *IEEE International Conference on Robotics and Automation*, 1990.
 - [37] Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T. « A stable tracking control method for non holonomic mobile robot ». In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, p 236-241, Osaka, Japan, 1991.
 - [38] Kenzai, A. « *Planification de trajectoires de robots mobiles via des méthodes ensemblistes* ». Thèse de DEA, Université d'Angers, France, 2005.
 - [39] Khadraoui, J. « *La commande référencée vision pour le guidage automatique de véhicules* ». Thèse de Doctorat, Université Blaise Pascal, France, 1996.
 - [40] Khalil, H. « *Nonlinear Systems, 2nd Edition* ». Upper Saddle River, NJ: Prentice Hall, 1996.
 - [41] Khatib, O. « Real-Time Obstacle Avoidance for Manipulators and Mobile Robots ». *International Journal on Robotics Research*, vol. 5, no 1, pages 90–98, Spring 1986.
 - [42] Kritic, M., Kokotovic, P., Kanellakopoulos, I.: « *Nonlinear and adaptive control design* », John Wiley & Sons, Inc., ISBN 0-471-12732-9, New York, 1995.
 - [43] Latombe, J. C. « Robot motion planning ». *Kluwer Academic Press*, 1991.
 - [44] Lavigne, M. L. « *Outils d'analyse et de synthèse de lis de commande robustes des systèmes dynamiques plats* ». Thèse de Doctorat, Université de Bordeaux I, France, 2003.
 - [45] Lefebvre, O. « *Navigation autonome sans collision pour robots mobiles non holonomes* ». Thèse de Doctorat, Université Institut National Polytechnique de Toulouse, France, 2006.
 - [46] Lenain, R. « *Contribution à la modélisation et à la commande de robots mobiles en présence de glissements* ». Thèse de Doctorat, Université Blaise Pascal-Clermont II, France, 2005.
 - [47] Leroquais, W. « *Modélisation et commande de robots mobiles à roues en présence de pseudo-glissements* ». Thèse de Doctorat, Ecole Nationale Supérieure des Mines, Paris, France, 1998.
 - [48] Maini, M., Morin, P., Pomet, J.-B., and Samson, C. « On the robust stabilization of chained systems by continuous feedback ». In *IEEE Conf. on Decision and Control (CDC)*, pages 3472–3477, 1999.

-
- [49] Marino, R. « On the largest feedback linearizable subsystem ». *System Control Letters* 7, pp. 345-351, 1986.
 - [50] Marr and Hildreth, 1980.
 - [51] Maya Mendez, M. « *Commande référencée capteur des robots non holonomes* ». Thèse de Doctorat, Ecole nationale supérieure des Mines de Paris, 2007.
 - [52] Minguez, J. and Montano, L. « Nearness diagram navigation (ND): a new real time collision avoidance approach ». *International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, 2000.
 - [53] Modestino and Fries, 1977.
 - [54] Morin P. and Samson C. « Practical stabilization of driftless systems on Lie groups: the transverse function approach ». *IEEE Transactions on Automatic Control*, Vol.49, n° 9, p 1496-1508.
 - [55] Morin P. and Samson, C. « A characterization of the Lie algebra rank condition by transverse periodic function ». *SIAM Journal on Control and Optimization*, Vol.40, n°4, p 1227-1249, 2001.
 - [56] Morin, P. « *Sur la stabilisation par retour d'état instationnaire* ». Thèse de Doctorat, Ecole des Mines de Paris, France, 1996.
 - [57] Morin, P. and Samson, C. « Exponential stabilization of nonlinear driftless systems with robustness to unmodeled dynamics ». *Control, Optimization & Calculus of Variations*, 4 :1-36, 1999.
 - [58] Morin, P. and Samson, C. « Practical stabilization of a class of nonlinear systems. Application to chain systems and mobile robots ». In *IEEE conference on Decision and Control*, volume 3, pages 2989-2994, Orlando (Florida, USA), 2000.
 - [59] Murray, R. M. and Sastry Shankar, S. « Steering nonholonomic systems in chained form ». In *Conference on Decision and Control*, p 1121-1126, Brighton, England, 1991.
 - [60] Nijmeijer, H., and Respondek W., « Dynamic input-output decoupling of nonlinear control systems », *I.E.E.E. Transactions on Automatic Control*, AC- 33, pp. 1065-1070, 1988.
 - [61] Panteley, E. « On Global Uniform Asymptotic Stability of Nonlinear Time-varying Systems in Cascade », *Systems and Control Letters*, Vol. 33, No.2, pp. 131-138, 1998.
 - [62] Pomet, J. B. « Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift ». *Systems and Control Letters*, 18 :147-158, 1992.
 - [63] Pourboghrat, F. « Exponential stabilization of nonholonomic mobile robots ». *Computers and Electrical Engineering*, 28(5) : 349-359, 2002.
 - [64] Samson, C. « Control of chained systems: Application to path following and time-varying point-stabilization of mobile robots ». In *IEEE Transactions on Automatic Control*, (40):64-77, 1995.
 - [65] Samson, C. « Velocity and torque feedback control of a nonholonomic cart ». In *International Workshop Nonlinear Adaptive Control: Issues in Robotics*, France, 1990.
 - [66] Samson, C. and Ait-Abderrahim, K. « Feedback control of a nonholonomic wheeled cart in cartesian space ». In *IEEE Conf. on Robotics and Automation (ICRA)*, pages 1136-1141, 1991.

- [67] Samson, C. and Morin, P. « Practical and asymptotic stabilization of chained systems by the transverse function control approach ». *SIAM Journal on Control and Optimization*, 43(1): 32-57, 2005.
- [68] Sekhavat, S. and Hermosillo, J. « Cycab bi-steerable cars: A new family of differentially flat systems ». *Advanced Robotics*, 16(5): 445-462, 2002.
- [69] Shen and Castan, 1986.
- [70] Slotine, J. E. and Li, W. « *Applied nonlinear control* », Prentice-hall, 1991.
- [71] Torre and Poggio, 1986.
- [72] Wonham W.M. and Morse A.S, « Decoupling and pole assignment in linear multivariable systems: a geometric approach », *SIAM J. Control* 8 (1), pp.1-18, 1970.
- [73] Zerar, M. « *Contribution à la caractérisation LPV d'une classe de systèmes non linéaires pour la synthèse de lois de poursuite robuste, application à un système spatial* ». Thèse de Doctorat, Université de Bordeaux I, France, 2006.
- [74] Zhang, J. R., Xu, S. J. and Rachid, A. « Path tracking control of vehicles based on Lyapunov approach ». In *American Control Conference*, vol 3, p 2132-2137, Anchorage, Alaska, USA, 2002.