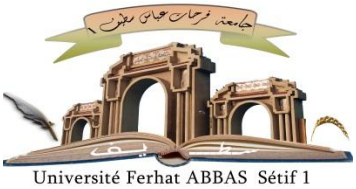
الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

Ministère de L'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITÉ FERHAT ABBAS - SETIF1

FACULTÉ DE TECHNOLOGIE

THÈSE

Présentée au Département de d'électronique

Pour l'obtention du diplôme de

DOCTORAT

Par

BOUTALBI Oussama

THÈME

Commande et Planification du Mouvement des Systèmes Non-linéaires Robotisés

Soutenue le ……./……./…… devant le Jury:

CHEMALI Hamimi	**Professeur**	**Univ. Ferhat Abbas Sétif 1**	**Président**
BENMAHEMMED Khier	**Professeur**	**Univ. Ferhat Abbas Sétif 1**	**Directeur de thèse**
ABDESSAMAD Foudil	**Professeur**	**Univ. Batna**	**Examinateur**
BOUMEHREZ Mohamed	**Professeur**	**Univ. Biskra**	**Examinateur**
BOUHARATI Sadek	**Professeur**	**Univ. Ferhat Abbas Sétif 1**	**Examinateur**

Je rends mes profondes gratitudes à ALLAH qui m'a aidé à réaliser ce modeste travail.

J'exprime mes profondes gratitudes à mes parents pour leurs encouragements, leurs soutiens et pour les sacrifices qu'ils ont enduré.

Je dédie ce modeste travail à mes frères et mes sœurs.

Je remercie mon encadreur Pr. K. Benmahammed pour les efforts qu'il a déployés, pour m'aider, conseiller, encourager et diriger.

Je remercie aussi tout le corps enseignant et administratif qui a contribué à ma formation universitaire.

Sans oublier toute ma famille et mes amis.

Remerciements

Je remercie Allah, le tout puissant, le miséricordieux, de m'avoir appris ce que j'ignorais, de m'avoir donné la santé et tout dont je nécessitais pour l'accomplissement de cette thèse.

Je remercie, en deuxième position, ma chère mère et mon cher père et mes soeurs et frères, sans lesquels je n'aurai jamais pu arriver à terme dans la rédaction, et qui m'ont toujours entouré d'affection et de soutien dans les moments pénibles comme dans les moments de joie.

Je remercie également, mon directeur de thèse Prof. KHEIR BENMAHAMMED pour son encadrement, ses directives et sa disponibilité. Aussi, je le remercie pour ses encouragements, son indulgence, et sa sympathie tout au long de la réalisation du présent travail au sein de Laboratoire des systèmes intelligents.

En outre, je remercie le prof H. CHEMALI président du jury, qui a accepté de présider l'honorable jury, malgré ses nombreuses occupations.

Également, je remercie les autres membres du jury Prof F. ABDEESSEMAD, Prof M. BOUMAHREZ, Prof M. TAIBI et Prof S.BOUHARATI pour la pertinence de leurs remarques avisées, qui vont permettre l'amélioration de la qualité du manuscrit et l'essor vers de nouvelles perspectives.

Je tiens à exprimer mes plus vifs remerciements à Dr.BOUBEZOULA Mabrouk, Dr. BOUKEZATA Boualem et Dr. HENNI Khadidja pour leurs conseils au cours de l'élaboration de cette thèse.

Je remercie aussi, tous ceux qui ont contribué, de près ou de loin, à ma formation du primaire au poste graduation.

Enfin, je ne peux pas oublier de remercier tous mes amis sans exception..

O. BOUTALBI
Sétif, le 19/10/2019

Table des matières

Remerciements i

Table des matières iii

Introduction générale **1**

1 Planification des mouvements pour les systèmes robotisés **7**

1.1 Introduction 8

1.2 Représentation mathématique d'un système robotisé 8

1.2.1 Représentation géométrique 8

1.2.2 Représentation Cinématique et Dynamique 10

1.3 Planification du mouvement : Problèmes et Terminologies 12

1.3.1 Planification sous contraintes globales 12

1.3.2 Planification sous des contraintes différentielles 13

1.3.3 Planification de la trajectoire 14

1.4 Planification du chemin 14

1.4.1 Spécification du chemin 16

1.4.2 définition du chemin 18

1.5 Profil du mouvement 22

1.5.1 Mouvement point à point (PTP) 24

1.5.2 Optimisation du mouvement le long d'un chemin prédéfini 27

1.5.3 Mise en échelle des trajectoires " Trajectories scaling" 28

1.6 Conclusion 29

2 Commande des systèmes non-linéaires **31**

2.1 Introduction 32

2.2 Commande des systèmes robotisés 33

2.3 Analyse des systèmes non-linéaires 34

2.3.1 Stabilité 34

2.3.2 Commandabilité 37

2.3.3 Robustesse 37

2.4 Linéarisation par Bouclage 37

2.4.1 Systèmes linéaires 38

2.4.2 Système non-linéaires 39

2.5 Commande par mode glissant 43

2.6 Commande Adaptative 45

2.7 Conclusion 47

3 Commande d'un robot mobile à roue **49**
3.1 Introduction . . . 50
3.2 Modélisation du système et formulation du problème . . . 51
3.2.1 Modélisation du système . . . 51
3.2.2 Formulation du problème . . . 53
3.3 Planification du mouvement . . . 54
3.3.1 Mouvement Point a point . . . 54
3.3.2 Évitement d'obstacles . . . 58
3.4 Commande en suivi de trajectoire . . . 62
3.5 Validation numérique des résultats . . . 63
3.5.1 Test 1 : Tâche de mouvement PTP . . . 64
3.5.2 Test 2 : Manipulation de robot dans un environnement encombré . 66
3.5.3 Test 3 : Évaluation de la commande de rétroaction . . . 71
3.6 Conclusion . . . 73

4 Commande d'un Portique à trois dimensions **75**
4.1 Introduction . . . 76
4.2 Modélisation du système et formulation du problème . . . 77
4.2.1 Modélisation du système, caractéristique et contraintes de commande . . . 77
4.2.2 Linéarisation par bouclage . . . 78
4.3 Planification du mouvement . . . 79
4.3.1 Planification du chemin . . . 79
4.3.2 Définition du profil de mouvement . . . 80
4.4 Suivi des Trajectoires . . . 81
4.4.1 Conception du contrôleur . . . 81
4.4.2 Analyse de stabilité . . . 83
4.5 Validation numérique des résultats . . . 85
4.5.1 planification de trajectoires . . . 85
4.5.2 Analyse de contrôle de suivi . . . 85
4.5.3 Analyse d'anti-ballant . . . 87
4.6 Conclusion . . . 90

5 Commande d'un bras manipulateur **92**
5.1 Introduction . . . 93
5.2 Modélisation du système et formulation du problème . . . 93
5.3 Conception du Twisting contrôleur sous des conditions d'incertitudes . . 94
5.3.1 Conception du contrôleur . . . 95
5.3.2 Analyse de stabilité . . . 96
5.4 Commande avec des paramètres inconnus . . . 96
5.4.1 Conception du contrôleur . . . 97
5.4.2 Analyse de stabilité . . . 98
5.5 Tests pratiques . . . 100
5.5.1 Prototype d'essai . . . 100
5.5.2 Développement et tests des contrôleurs . . . 101
5.5.3 Discussion des résultats . . . 107
5.6 Conclusion . . . 111

Conclusion générale **113**

Bibliographie **I**

Table des figures **XXI**

Liste des tableaux **XXIII**

Introduction générale

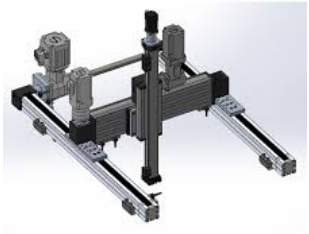
Ce travail de doctorat a été préparé au sein de l'équipe de traitement d'images du Laboratoire des Systèmes Intelligents (LSI), Université Ferhat Abbas Setif -1-, Setif, Algérie. Le thème de cette thèse concerne la problématique que pose le développement et la conception des stratégies de commande avancées pour des systèmes robotisés. La robotique traite de l'automatisation de systèmes mécaniques dotés de capacités de détection, d'actionnement et de traitement [LaValle2006]. Les robots peuvent être classifiés en **robots industriels** et **robots de service** en fonction de leurs utilisation [ISO 8373 :2012] :

— **Robot industriel** est un manipulateur, multi-applications, re-programmable, commandé automatiquement, programmable sur trois axes, ou plus, qui peut être fixe ou mobile, destiné à être utilisé dans des applications d'automatisation industrielle. Les applications d'automatisation industrielle incluent la fabrication, le contrôle, le conditionnement, et l'assemblage, mais ne s'y limitent pas. Des exemples génériques des robots industriels sont illustrés dans Fig.1.

(a) Robot Scara (OMRON)

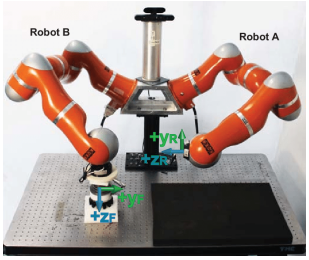
(b) Robot Articulé (Kuka)

(c) Robot Parallèle (Adept)



(d) Robot Cartésien

(e) Manipulateur Mobile

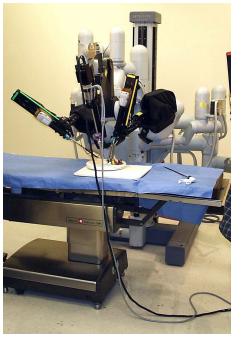


(f) Kuka Dual Arm

Figure 1 – Exemples génériques de robots industrielle

— **Robot de service** est un robot qui exécute des tâches utiles pour des humains ou des appareillages, excluant les applications d'automatisation industrielle. Tandis que les robots articulés utilisés sur des lignes de production sont des robots industriels, les

(a) Robots Humanoïde



(b) Robot chirurgicale

(c) fauteuil roulant avec bras manipulateurs (Jaco)

(d) Robot maçon (HADRIAN)

(e) Automatisation Agriculture

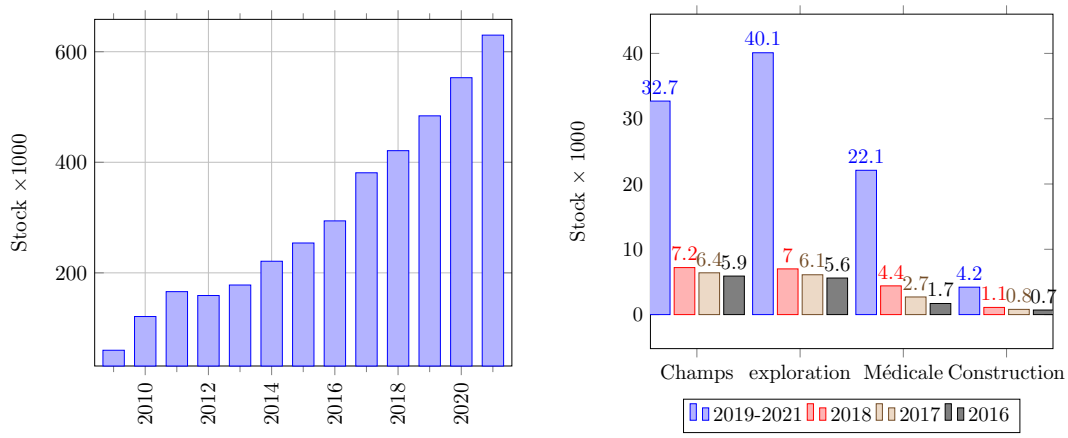
(f) Robot d'inspection sous marins

Figure 2 – Exemples de robots destinés pour des applications services

robots articulés similaires utilisés pour servir de la nourriture sont des robots de service. Des exemples pour des services des systèmes robotiques sont vus dans Fig.2.

Selon la **Fédération Internationale de Robotique (IFR)** [IFR], le stock mondial de la robotique industrielle a réalisé une croissance importante de entre les années 2015 et 2020, comme le montre la Fig.3.a. L'IFR prédit qu'il atteint le nombre de 6 million d'unités dans l'année 2020. De même pour les robots de service, Fig.3.b. Les statistiques prédisent que le stock mondial des robots de service sera doublé dans les années. Les robots n'ont pas seulement pénétré le monde industriel, ils sont aussi entrain de pénétrer notre vie quotidienne et notre culture, et certains d'entre eux participent au renouvellement de la vision que nous l'avons nous même. Les robots sont donc partout : dans les usines et dans les champs, au fond des mers et dans l'espace, dans les jardins et les salons. Ils ont une importance grandissante, et d'aucuns prédisent qu'ils seront au XXIe siècle ce que la voiture fut au XXe siècle.

Cette révolution annoncée, dont l'impact sur la science, l'industrie et la société sera potentiellement très grand, va cependant de pair avec un certain nombre de grands défis à résoudre, tant technologiques qu'économique. Le défi majeur peut se résumer à savoir comment peut on mettre au point des techniques permettant à un robot d'accomplir des tâches d'une manière efficace et sécurisée ? Chaque application apporte ses propres exigences en fonction de la tâche associée, la nature de l'environnement où le robot exécute cette tâche et les limites physiques de la structure mécanique. Le diagramme représenté la Fig.4, résume les différentes fonctionnalités nécessaires à l'automatisation des systèmes



(a) Robots industriels

(b) Robots de services

Figure 3 – Estimation annuelle du stock mondial des robots

mécaniques. De nombreux problèmes complexes ont été découverts dans les domaines de conception, de modélisation, de perception de l'environnement, de navigation, de localisation, de planification des mouvements, de contrôle et de programmation [Siciliano2016]. Les travaux de cette thèse sont focalisés sur le traitement et l'analyse des problèmes que pose les tâches de commande et de planification des mouvements du robot prenant en compte des différentes limites reliées aux systèmes lui même et son environnement de fonctionnement afin d'optimiser les performances du système robotisé. Le problème combiné de planification et de suivi des trajectoires peut être formulé comme une fonction qui gère un compromis entre des différents critères d'efficacité et la réponse à cette question dépendra de différents facteurs :

1. Le premier facteur qui influence la conception de la loi de commande qui répondra le mieux à ce problème est peut-être le plus évident ; les exigences techniques de la tâche associée au robot. Les performances souvent les plus caractéristiques :
   - **La rapidité** : critère global qui englobe les vitesses, les accélérations et les jerks (dérivée de l'accélération) des différents axes du robot, les temps de stabilisation aux points d'arrêt, les pertes de temps générées par les points de passage, les retards de synchronisation entre les mouvements et les procédés associés... ;
   - **L'adaptabilité** : Ce point est en relation direct avec la définition d'un système robotique exigeant différentes taches sous différentes conditions de fonctionnement.
   - **La précision et la sécurité** : ces deux critères représentent la capacité d'un système robotique à réaliser des tâches effectivement en toute sécurité, c.à.d, sans dépasser les limites du robot.
2. Un deuxième facteur important qui influence la réponse à ce problème général est le comportement dynamique du système. Dans les systèmes de fabrication, par exemple, cela devient évident lorsque les machines sont conçues pour être plus légères et plus rapides. Une conséquence direct de ces améliorations est que des dynamiques indésirables, telles que les vibrations, deviennent souvent un problème qui doit sérieusement considéré l'hors de conception d'une loi de commande.

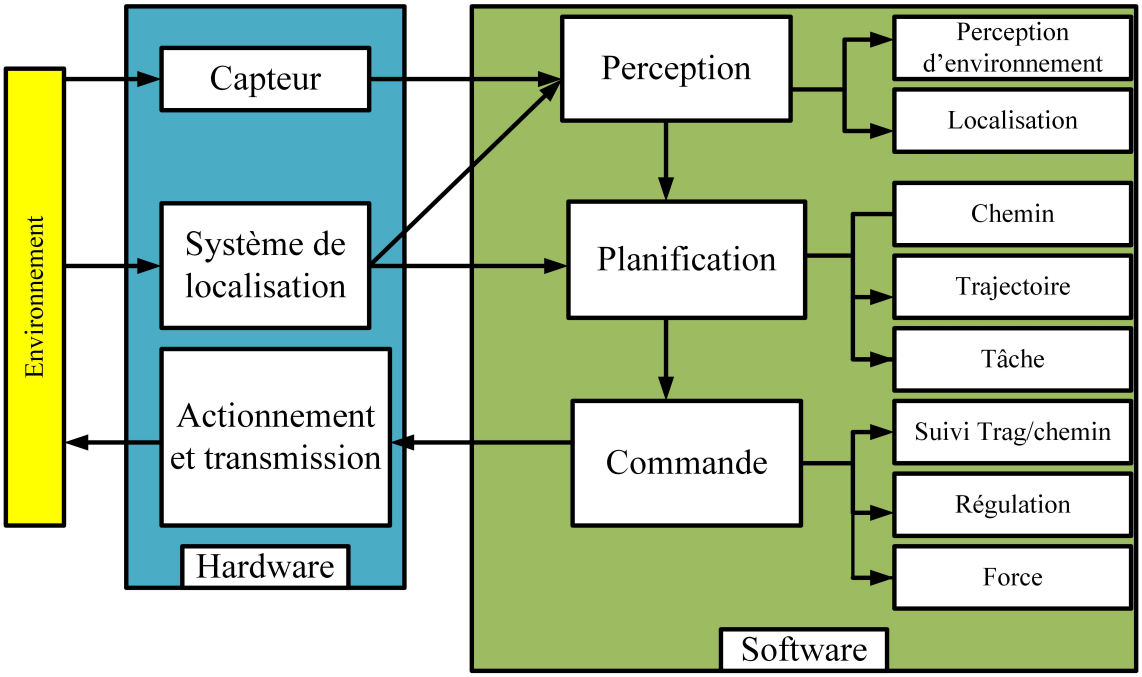


Figure 4 – Automatisation d'un système robotisé

3. Un troisième facteur à prendre en compte est les limites physiques du système que ce soit géométrique, cinématique ou dynamique. Les commandes de mouvement utilisées pour ces systèmes doivent être conçues pour rester dans les capacités des actionneurs du système. D'autres systèmes peuvent avoir des exigences supplémentaires , telles que la bornitude d'espace de travail ou l'efficacité énergétique de la loi proposée. Toute stratégie utile pour créer des lois de commande doit répondre efficacement à ces préoccupations pratiques.
4. Outre que la prise en compte du comportement de sortie souhaitée, de la dynamique du système, des limites physiques de système et les variations dans l'environnement de fonctionnement, de nombreux systèmes de commande peuvent être sensibles à d'autres variables. Par exemple, pour les systèmes dont la dynamique est instable, incertaine ou totalement inconnue, les performances fournies par les commandes d'entrée doivent rester robustes face à ces changements.

Bien que le problème de contrôle du mouvement, tel qu'il est posé ci-dessus, reste encore assez large, mais il a permis d'identifier un certain nombre de questions reliées à l'efficacité de la solution proposée. Est que cette solution est optimale (dans un sens bien défini) ? Est qu'elle est réalisable ? Est que le système de commande permet d'assurer la robustesse malgré la présence des phénomènes non considérés à la phase initiale de conception de contrôle ?

Pour répondre à ce nombre de questions, nous commençons cette thèse par une révision approfondie (review) sur les différents algorithmes proposés dans la littérature pour traiter le problème de planification sous différentes conditions d'optimalité et différentes exigences. Notre étude concentre essentiellement sur une approche dite découplée qui consiste à décomposer la phase de planification en deux étapes, une phase pour la définition de chemin sûr et sans collisions et une phase pour la génération des trajectoires le long de chemin planifié.

Le deuxième chapitre aborde la problématique de suivi exact et précis des trajectoires

générées sous la considération de différents types d'incertitudes. Nous avons également basé sur l'étude des principes de linéarisation par bouclage, de commande par mode glissant et la commande adaptative pour atteindre des niveaux élevés de précision et de robustesse.

Les trois dernier chapitres traitent des exemples typiques d'applications de robots. Dans le premier exemple, nous considérons en particulier la classe des systèmes nonholonomes pour le développement d'un algorithme qui permet d'optimiser le mouvement d'un robot mobile à roues. Dans cet exemple on a exploité la propriété de platitude pour la représentation des chemins lisses avec une courbature minimale. Le problème d'optimisation sous des contraintes cinématiques est considéré dans un deuxième temps. La notion de platitude est aussi adaptée pour résoudre le problème de suivi des trajectoires. Les résultats de ce chapitre sont confirmés par des tests pratiques en utilisant le prototype de test «PIONNER 3DX ».

Dans le deuxième exemple on considère le problème de commande d'un portique à trois dimensions. Ce type de systèmes largement utilisé dans les sites de construction pour la manutention des charges lourdes est principalement caractérisés par une forme flexible multivariable dite sous actionnée. Les principales étapes pour étendre l'approche de planification définie dans le chapitre trois sont initialement discutées. Deuxièmement, une commande par mode glissant d'ordre deux est développée pour assurer le suivi stable précis et robuste des trajectoires générées ainsi que l'atténuation des indésirables oscillations de la charge. La stabilité du système est analysée au sens de Lyapunov et confirmée par résultats simulés sous l'environnement Matlab/Simulink.

Le troisième exemple considère un bras manipulateurs comme exemple d'étude. Les robots manipulateur, quant à eux, sont riches en trajectoires réalisables pour des tâches spécifiques, mais le choix entre des tâches efficaces ou optimales est difficile. Nous traitons principalement la problématique de suivi des trajectoires dans le cas où les paramètres dynamiques du système sont inconnus. Un algorithme hybride de deux techniques, commande adaptative et commande par mode glissant d'ordre supérieur est développé pour dépasser les défis des incertitudes paramétriques et non paramétriques. Après une analyse théoriques des performances du contrôleur proposé, les résultats des ce chapitre sont confirmés par des tests pratiques en utilisant le prototype de test du laboratoire (Cyton GAMMA 1500).

On a terminé ce travail par une conclusion générale.

1

Planification des mouvements pour les systèmes robotisés

Ce chapitre vise à mettre l'accent sur les différentes approches de modélisation et de planification des mouvements des systèmes robotisés.

Sommaire

1.1 Introduction 8
1.2 Représentation mathématique d'un système robotisé 8
1.2.1 Représentation géométrique 8
1.2.2 Représentation Cinématique et Dynamique 10
1.3 Planification du mouvement : Problèmes et Terminologies 12
1.3.1 Planification sous contraintes globales 12
1.3.2 Planification sous des contraintes différentielles 13
1.3.3 Planification de la trajectoire 14
1.4 Planification du chemin 14
1.4.1 Spécification du chemin 16
1.4.2 définition du chemin 18
1.5 Profil du mouvement 22
1.5.1 Mouvement point à point (PTP) 24
1.5.2 Optimisation du mouvement le long d'un chemin prédéfini . . . 27
1.5.3 Mise en échelle des trajectoires " Trajectories scaling" 28
1.6 Conclusion 29

1.1 Introduction

L'objectif d'un planificateur du mouvement est de guider automatiquement un robot dans un environnement jonché d'obstacles et de choisir les commandes à appliquer au cours de l'exécution de la tâche du système. Des exigences techniques et économiques imposent souvent l'optimisation d'une fonction de coût sur le mouvement du robot, telles que : le temps d'exécution d'une tâche ou la quantité d'énergie requise pour assurer une manipulation efficace du robot. Le planificateur de mouvement doit pris en compte les différentes limites physiques de système, telles que les limites en couples ou vitesses générées par les moteurs d'actionnement. Il existe deux modes de fonctionnement du planificateur de trajectoire : hors-ligne, en exploitant des connaissances préalables sur l'environnement de fonctionnement du robot et de la tâche associée, ou en ligne en utilisant différents types de capteurs qui vont guider le robot vers sa position désirée et détecter la présence d'obstacles. Par conséquent, de nombreuses solutions peuvent être envisagées pour résoudre le problème de la planification des mouvements.

Le présent chapitre aborde les différentes techniques développées dans le domaine de la planification des mouvements des systèmes robotisés. Des notions de base pour représenter le mouvement du robot sont données dans la section 2. Les principaux problèmes liés à la planification des mouvements, sous différentes contraintes, sont abordés dans la section 3. La section 4 survole certaines solutions significatives de la planification des chemins lisses sous différentes considérations. Les méthodes d'optimisation de la vitesse du robot, tout au long de son chemin préalablement défini, sont abordées dans la section 5.

1.2 Représentation mathématique d'un système robotisé

Le but de cette partie est d'introduire quelques notions de base dans les domaines de modélisation, de planification et de commande des systèmes robotisés. La structure mécanique de robots typiques constitue des systèmes à plusieurs degrés de liberté et à une dynamique extrêmement non linéaire et couplée. Une base mathématique pour la modélisation de la géométrie, la cinématique et la dynamique des systèmes robotiques est bien établie dans la littérature et peut être trouvée dans de nombreux manuels tels que [LaValle2006, Siciliano2010, Khalil2000].

1.2.1 Représentation géométrique

Il existe une grande variété d'approches et de techniques de modélisation géométrique d'un robot A et de l'espace de travail $W \in R^n$, où $n = \{2,\ 3,\ \cdots\}$ [LaValle2006]. Le choix dépend en général de l'application et de la difficulté du problème. Considérant les problèmes classiques de planification dans le but d'assurer des mouvements sans collisions entre le robot et les obstacles et sans considération de la cinématique ni la dynamique du système, les obstacles et les robots seront modélisés comme des sous-ensembles fermés de forme limite ou solide (Polygonal, Polyhedral, Semi-Algebraic Models,...) dans l'espace de travail W.

Comme le montre la Fig.1.1, une description complète de la position d'un corps rigide, et par conséquent du mouvement de ce corps, dans une référence inertielle RI est obtenue par la considération d'un vecteur de six degrés de liberté en minimum connus sous le nom

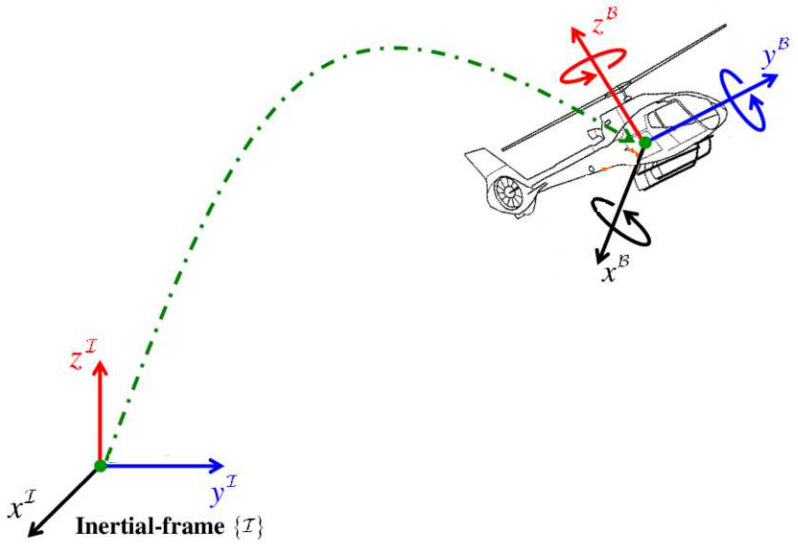


Figure 1.1 – Représentation du mouvement d'un corps rigide dans l'espace 3D

"vecteur de coordonnés opérationnels" définis par [Siliciano2010] :

— Un vecteur de position $P = [x\ y\ z]$ qui spécifie la position de la référence *RC* (Référence reliée au corps) par rapport à la référence *RI*.

— Une spécification convenable de l'orientation du corps par rapport au référentielle *RI*. Cette spécification peut être définie par une représentation **minimale** (Angle d'Euler) ou **redondante** (matrice d'orientation, orientation Axe/Angle ou Quaternions).

La connaissance du vecteur de position r (qui regroupe les représentations de position et d'orientation) peut être obtenue indirectement à partir de la connaissance de certains paramètres $q_j (j = 1, \cdots m)$ si la relation suivante (1.1) est connue :

$$r = f(q) \tag{1.1}$$

La relation (1.1) représente le modèle géométrique direct (*MGD*) d'un système robotisé. On peut définir aussi :

— Le vecteur de **configuration** ou **vecteur de coordonnées généralisé** noté q, qui donne une spécification complète de la position pour chaque point du robot dans le *RI*.

— **Espace de configuration** ou **Espace d'état**, noté Q ou **C-space** : c'est l'espace de toutes les configurations possibles pour le système, $q \in C$.

— **Degrés de liberté** (*dll*) : c'est la dimension de l'espace de configuration, ou le nombre minimum de paramètres nécessaire pour la spécification de la configuration du robot.

— **Espace de configuration collision, noté** Q_{osb} : est défini comme l'ensemble des configurations où le robot rencontre un obstacle.

— **Espace de configuration libre** : l'espace libre ou l'espace de configuration libre Q_{free} est défini comme l'ensemble des configurations sans collisions avec obstacles dans l'espace Q.

Remarque 1 *Selon leurs **degrés de mobilité**, les robots peuvent être classifiés en **robot mobile** et **robot fixe** (bras manipulateurs). Les robots mobiles sont distingués par la présence d'une plateforme mobile qui permet au robot d'exécuter des tâches complexes dans des environnements non structurés. La convention de Denavit-Hartenberg, largement connue dans le domaine des robots à chaine cinématique ouverte ou fermé, permet de caractériser la position relative de deux solides avec seulement quatre paramètres (au lieu de six). Ceci au prix de certaines restrictions dans le choix des systèmes de coordonnées (repères). En effet, on rencontre deux définitions qui diffèrent légèrement [Siliciano2010, Khalil2000] :*

- *La convention de Denavit-Hartenberg, qui fut introduite en 1955, par Jacques Denavit et Richard S. Hartenberg*
- *La convention de Denavit-Hartenberg modifiée, appelée aussi convention de Khalil-Kleinfinger3, qui est préconisée depuis 1964.*

Les deux notations sont couramment utilisées pour la modélisation des chaînes cinématiques constituées par des corps solides reliés entre eux par des articulations rotoïdes (pivots) ou prismatiques (glissières). Pour chaque articulation, on peut définir l'axe de sa liaison et sa variable articulaire.

Remarque 2 *D'un point de vue mécanique, on peut distinguer essentiellement deux types de systèmes [Tchoń2015, Tchon2018] :*

- ***Les systèmes holonomes** : dans lesquels les liaisons existant entre points matériels prennent la forme :*

$$f(q_j,t) = 0 \tag{1.2}$$

  et peuvent donc être résolues d'une manière explicite, afin de fournir la position des différents points du système à partir de coordonnées généralisées que l'on peut varier de façon indépendante sans violation des liaisons.

- ***Les systèmes non-holonomes** : dans lesquels les coordonnées ne peuvent être variées de façon indépendante, mais doivent vérifier des liaisons non intégrables à priori. Les liaisons s'expriment alors sous forme différentielle.*

$$\sum a_{kj} d_{qj} + b_k dt = 0, k = 1, \cdots s \tag{1.3}$$

  Dans le cas le plus général, les coefficients a_{ji} et b_i peuvent dépendre des q_i et de t.

1.2.2 Représentation Cinématique et Dynamique

Le modèle dynamique qui exprime la relation entre les couples/forces en fonction du vecteur des coordonnées généralisées et ses dérivés, peut être abordé de deux manières, le formalisme Newton-Euler ou Lagrange-Euler. Le Lagrangien L est défini par la différence entre l'énergie cinétique du robot et son énergie potentielle. Dans le cas général, le modèle dynamique d'un robot consistant de n coordonnées généralisées est donné comme [Siliciano2010] :

$$\begin{cases} \frac{d}{dt}(\frac{\partial L}{\partial \dot{q}}) - \frac{\partial L}{\partial q} + A^T \lambda = E\tau \\ A(q)\dot{q} = 0 \end{cases} \tag{1.4}$$

d'où, $q \in R^n$ est le vecteur des coordonnées généralisées. $A(q)$ est une matrice de l contraintes non-holonome avec $A(q)\dot{q} = 0$. $\tau \in R^m$ est le vecteur de commande, $m \leq n$. λ est un vecteur du multiplicateur de Lagrange. E est une matrice constante.

La contrainte non-holonome implique que le vecteur de vitesses généralisées $\dot{q}$ est dans un espace nul de la matrice $A(q)$, ce qui signifie qu'il est possible de trouver un vecteur des vitesses auxiliaire $V = [v_1\ v_2 \cdots\ v_m]^T$ tel que :

$$\dot{q} = S(q)V \tag{1.5}$$

d'où $S(q)$ est l'espace nul de la matrice A, c.à.d $S^T\ .A^T = 0$. L'équation (1.5) représente le modèle cinématique d'un système non-holonome.

Pour un système holonome, le modèle d'équation (1.4) se réduit à :

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}}) - \frac{\partial L}{\partial q} = E\tau \tag{1.6}$$

et le modèle cinématique direct peut être obtenu dans ce cas comme :

$$\dot{r} = J\dot{q} \tag{1.7}$$

d'où $J_{n\times m}$ est la matrice Jacobien définie :

$$J = \frac{\partial f(q)}{\partial q}$$

Remarque 3 *Si $m < n$ le système (1.6) est dit sous actionné. Cette propriété représente une contrainte additionnelle qui augmente les défis reliés avec les problèmes de planification et de commande des systèmes robotisés. On note ici que les systèmes non-holonome ont la même nature que celle des systèmes sous-actionnés. Les portiques et les robots à structures flexibles sont des exemples typiques des systèmes mécaniques sous-actionnés.*

Remarque 4 *Dans ce qui suit, nous considérons le problème de planification des mouvements dans un espace Cartésien (ou espace opérationnel), cependant la commande est effectuée en considérant l'espace des coordonnées généralisées. Pour cela nous introduisons les notions de modèle géométrique inverse et modèle cinématique inverse :*

— *Modèle géométrique inverse (MGI) : c'est le modèle qui représente le vecteur des coordonnées généralisées q en fonction du vecteur des positions opérationnelles :*

$$q = f^{-1}(r) \tag{1.8}$$

— *Modèle cinématique inverse (MCI) : c'est le modèle qui représente le vecteur des vitesses généralisées q en fonction du vecteur des vitesses opérationnelles :*

$$\dot{q} = J^{-1}(q)\dot{r} \tag{1.9}$$

Les équations (1.8 et 1.9) montrent bien la nature de deux classes de problèmes, largement connus dans le domaine planification, surtout pour les systèmes articulaires, le problème de singularité et le problème de redondance. Le problème de singularité implique qu'il existe des situations (dans l'espace articulaire) qui doivent être éliminées pour assurer la faisabilité de la solution proposée dans l'espace opérationnelle. Le problème de redondance implique qu'il existe un défi d'optimisation pour choisir une meilleure solution lorsque l'inverse de la matrice f et/ou J accepte plusieurs solutions.

1.3 Planification du mouvement : Problèmes et Terminologies

Après la courte introduction de la modélisation des systèmes robotisés, qui nous a permis d'introduire un certain nombre de caractéristiques des différents systèmes robotiques, cette section met l'accent sur les différentes catégories des problèmes de la planification. En fonction du type, de la structure mécanique et de l'application associée au robot, deux problèmes de base peuvent être essentiellement distingués : le problème de planification sous des contraintes **globales** et le problème de planification sous des contraintes **locales** [Kavraki2008]. Les contraintes globales dans l'espace de configuration $\mathbb{C}$ ont été considérées sous la forme d'obstacles et éventuellement les limites sur les articulations. Les contraintes locales sont modélisées avec des équations différentielles et sont donc appelées contraintes différentielles, celles-ci limitent les vitesses et les accélérations, en tout point du fait de considérations cinématiques, telles que les roues en contact et les considérations dynamiques.

1.3.1 Planification sous contraintes globales

Cette classe représente le problème de base de la planification du mouvement, précisément noté dans ce cas comme un problème de *planification d'un chemin sans collisions*, le robot est modélisé comme un corps rigide qui se déplace dans l'espace de configuration. Le but du planificateur consiste à définir tous les chemins assurant un mouvement sans collisions du robot entre deux configurations distinctes $q_I \in C_{free}$ et $q_F \in C_{free}$ [Kavraki1998, Kingston2018]. La solution, si elle existe, est un chemin P sans collision.
Pour résoudre ce type problème, différentes techniques ont été développées telles que, les méthodes de **décomposition cellulaire** [Schwartz1983a, Schwartz1983b, Schwartz1984], les **champs potentiels** [Khatib1986] et les méthodes de **construction des graphes** [Nilsson1969].
La notion de **complétude** différencie ces approches. On distingue des planificateurs complets, des planificateurs complets en résolution et des planificateurs de complétude probabiliste [Şucan2012]. Les méthodes de planification complètes (ou méthodes exactes) garantissent de trouver une solution ou d'indiquer qu'il n'y en a pas. C'est la première famille de méthode développée pour résoudre des problèmes de planification de chemin. Les méthodes complètes en résolution sont basées sur la discrétisation de l'espace des configurations. Elles garantissent de trouver un chemin solution s'il existe, mais uniquement à une résolution donnée. Les méthodes probabilistes garantissent quant à elles de trouver une solution en un temps fini si elle existe, mais elle ne permet pas de confirmer si une solution n'existe pas. Contrairement aux méthodes exactes et complètes en résolution, les méthodes probabilistes ne sont pas basées sur une construction préalable de l'espace des configuration. On distingue deux catégories de méthodes probabilistes : les techniques basées sur la construction de réseaux probabilistes et les techniques de diffusion (ou incrémentales).
Un planificateur de trajectoire peut fonctionner **en mode hors-ligne**, en exploitant des connaissances préalables sur l'environnement de fonctionnement des robots et de la tâche associée, ou **en ligne**, en utilisant différents types de capteurs qui permettent de guider le robot vers sa position désirée et de détecter différente variation dans l'environnement de

fonctionnement. La planification a priori ou hors-ligne, résultante en mouvement déterministe et prévisible, est sollicitée dans plusieurs applications robotiques : véhicules guidées automatiquement (AGV) [Gonzalez2016, Braghin2008], systèmes levage et de manutention [Blajer2007], les robots manipulateurs dans des applications industriels [Shin1985] et beaucoup d'autres applications. Dans ce cas, l'algorithme de la planification produit un trajet complet du point de départ au point objectif, avant que le robot commence son mouvement.
En plus des approches mentionnées ci dessus, Il convient de noter aussi les techniques d'intelligence artificielle, telles que la logique floue, les réseaux de neurones, les algorithmes génétiques, $\cdots$, etc, qui ont connus un grand succès dans le domaine de planification que ce soit pour l'amélioration des performances des techniques classiques ou la conception de nouveaux algorithmes de mouvement, pour un aperçu voir [Patle2019].

1.3.2 Planification sous des contraintes différentielles

Bien qu'il existe de nombreux méthodes pour résoudre le problème de planification sous des contraintes globale, l'inclusion des contraintes intrinsèques du robot au problème de la planification entraîne la modification de l'espace de recherche, qui n'est plus limité à C_{free} comme défini précédemment. Une configuration valide est non seulement celle qui assure l'évitement d'obstacles, mais également celle qui satisfait des contraintes supplémentaires reliées à la cinématique et la dynamique du robot [Kavraki2008].
Soit un robot $A(q,\dot{q},u)$, d'où : $\dot{q}$ est le vecteur des vitesses instantanées du vecteur des configurations q, u est le vecteur de commande, choisi à partir de l'espace des entrées U.
Les contraintes différentielles **cinématiques** peuvent être généralement modélisées soit :

— Implicitement sous la forme :

$$g(x,\dot{x}) = 0 \tag{1.10}$$

— Explicitement sous la forme paramétrique :

$$\dot{q} = f(x,u) \tag{1.11}$$

avec $x = q$.

De même, ce concept peut être étendu prenant en compte les **dynamiques** du système en introduisant le vecteur d'espace de la phase x définie comme :

$$x = [q \;\; \dot{q}]^T \tag{1.12}$$

Pour obtenir :

$$\begin{cases} g(x,\dot{x}) = 0 \\ \dot{x} = f(x,u) \end{cases} \tag{1.13}$$

Une extension de la notion d'évitement d'obstacles est étendue dans l'espace d'état X en introduisant la notion de la région des obstacles inévitables X_{ric} définie comme :

— X_{obs} est défini comme si et seulement si $q \in q_{obs}$ pour $x = [q\ \dot{q}]$.

— $X_{ric} = \{x(0) \in X |\ for\ any\ u \in U_{\text{inf}}, \exists t > 0\ such\ that\ x(t) \in X_{obs}\}$, d'où X_{ric} (region of inevitable collisions) représente la zone où le robot est en collision ou ne peut pas éviter un obstacle. Cette zone peut être considérée comme un invisible obstacle qui se développe proportionnellement avec la vitesse du robot.

Sous la rubrique générale de la planification sous contraintes différentielles, de nombreuses catégories importantes de problèmes ont fait l'objet d'une attention considérable dans la littérature scientifique :

— **Les contraintes sur l'organe terminal dans l'espace Cartésien** : pour les bras manipulateurs à chaines ouverte ou fermée.

— **Les contraintes non-holonomes** : pour les robots mobiles.

— **Les contraintes kinodynamique** : le problème de planification sous des contraintes sur les vitesses et les accélérations s'appelle un problème de planification "Kino-Dynamic".

1.3.3 Planification de la trajectoire

Les algorithmes de planification du chemin discutés dans la section précédente ignorent généralement les spécifications de la tâches et se concentrent principalement sur les translations et les rotations nécessaires au déplacement du robot pour aller d'un point à un autre. Des travaux plus restrictifs, cependant, prennent en compte d'autres aspects, tels que l'optimalité de la solution proposée et la qualité physique des signaux de commande générée. On se place ici dans un problème de la planification des trajectoires. La planification de trajectoire fait généralement référence au problème d'extraction de la solution d'un algorithme de planification de chemin pour le robot et de la détermination de son déplacement (à quel instant le robot occupe une position donnée) dans cette solution d'une manière qui respecte les limites mécaniques du robot et assure l'optimisation d'un critère donné. La trajectoire planifiée doit répondre aussi à d'autres exigences reliées à la tâche associée au robot, telles que l'optimisation du temps ou de l'énergie nécessaire pour l'exécution de la tâche et l'assurance des conditions acceptables pour le fonctionnement du système. Pour résoudre cette problématique, deux approches peuvent être distinguées [Khalil2000] :

1. **Approche couplée :** notée aussi approche "kinodynamique".
   Une trajectoire est définie directement dans l'espace d'état du système (ou espace de coordonnées généralisées) en utilisant, par exemple, des méthodes de contrôle optimales, une programmation dynamique ou des méthodes probabilistes aléatoires.
2. **Approche découplée :** notée aussi découplage Chemin-Trajectoire.
   Dans des environnements connus ou partiellement connus, cette approche permet de simplifier considérablement le problème de la planification des mouvements pour les robots mobiles ainsi que pour les robots industriels. L'optimisation des trajectoires planifiées est assurée en premier lieu par le choix approprié du chemin dans lequel le robot doit se déplacer, puis par un choix convenable d'une fonction temporelle qui assure la synchronisation des mouvements des différents axes du robot. Cette fonction est appelée **"Profil du mouvement"**.

1.4 Planification du chemin

Dans la rubrique de planification des trajectoires avec une approche découplée, on définit un chemin entre une configuration initiale X_I et une configuration finale X_F comme une courbe relativement lisse $P(u)$ qui représente le déplacement du robot A dans son environnement. La Fig.1.2 montre un ensemble de solutions possible pour le déplacement

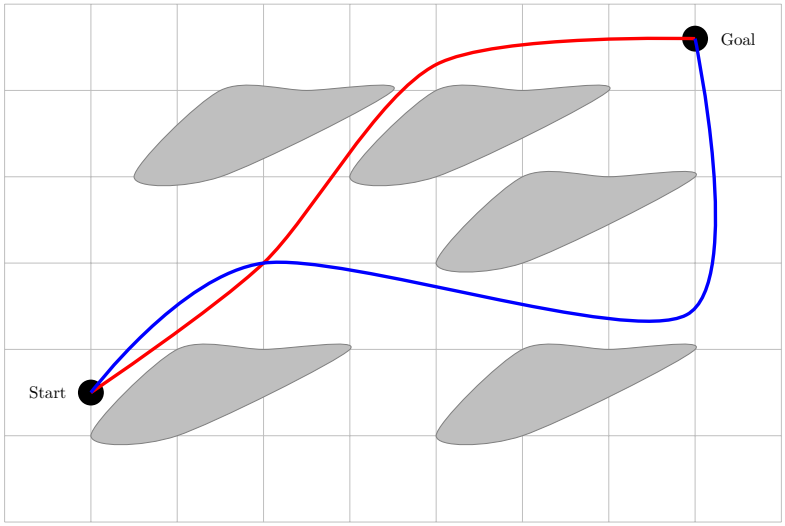


Figure 1.2 – Chemins lisses dans le plan

de robot. Il s'agit donc d'un problème de reconstruction et de sélection des solutions les plus sophistiqués pour un fonctionnement efficace du robot. La sélection est faite soit en ligne ou hors ligne à partir des données obtenues de l'un des algorithmes de planification classique. La conception et la fabrication assistée par ordinateur (CAGD, Computer Aided Geometric Design) conduit à définir et à manipuler des courbes continues dans l'espace Cartesien en $2D$ ou $3D$. Ces courbes sont le plus souvent décrites sous forme paramétrique à l'aide des fonctions polynomiales ou rationnelles :

$$\begin{cases} P(u) = [x(u)\ y(u)\ z(u)]^T \\ P(u_{min}) = X_I,\ P(u_{max}) = X_F \\ \forall \eta \in u : P(\eta) \in C_{free},\ g(q,\dot{q}) = 0 \end{cases} \tag{1.14}$$

d'où : $u \in [u_{min}; u_{max}]$ est un paramètre quelconque, P est une fonction choisie telle que $\left\|\vec{P}'(u)\right\| = \left\|\frac{dP}{du}\right\| \neq 0\ \forall u$. La troisième égalité dans l'équation (1.14) est introduite pour assurer le respect des contraintes géométrique et cinématique durant le mouvement du robot.
Le choix du paramètre d'interpolation u et de la fonction $P(u)$ donne aux problèmes de planification (1.14) plusieurs degrés de liberté pour résoudre différents types de problèmes avec différentes contraintes de mouvement.
Nous supposerons également une autre paramétrisation $s(u)$ résultant de l'inversion de l'équation (1.15) quelle que soit la nature du paramètre u. Cela donne ce qu'on appelle une paramétrisation en fonction de la longueur d'arc définie comme :

$$s(u) = \int_{u_0}^{u} \left\|\vec{P}'(u)\right\| du \tag{1.15}$$

d'où $u \geq u_0$. Ce choix permet de garantir que :

$$s(u) \geq s(u_0)$$
$$\left\|\vec{P}'(s)\right\| = 1$$

Ce choix permet aussi de simplifier la phase de modélisation du profil du mouvement et du robot tout au long du trajet, voir chapitre .3 et [Boutalbi2019]. Nous référons à $P(u)$ lorsque nous discuterons une courbe avec une paramétrisation régulière arbitraire, et nous ferons référence à $P(s)$ lorsque nous discuterons d'une courbe paramétrée en fonction de longueur d'arc s.
Pour résoudre le problème d'inversement (1.15), deux situations doivent être discutées :

1. Comment obtenir le paramètre s si la valeur de u est connue
2. et inversement, comment obtenir u si la valeur de s est connue.

La solution à ces deux problèmes n'est pas toujours évidente et fait l'objet de nombreuses recherches, on cite par exemple [Sharpe1982, Guenter1990].

1.4.1 Spécification du chemin

D'une manière générale, la conception et la définition d'un chemin lisse s'appuient sur des données préalables. Celles-ci proviennent par exemple d'un des algorithmes de planification discutés dans les sections précédentes. On parle alors de reconstruction des chemins sous la considération des contraintes spécifiques, telles que la continuité, les limitations sur la courbure et ses dérivées, l'évitement de collision avec les obstacles, $\cdots etc$. La forme à définir ou à reconstruire passera alors par des configurations spécifiques (interpolation) ou à proximité de ces configurations (approximation), Figure.1.3.

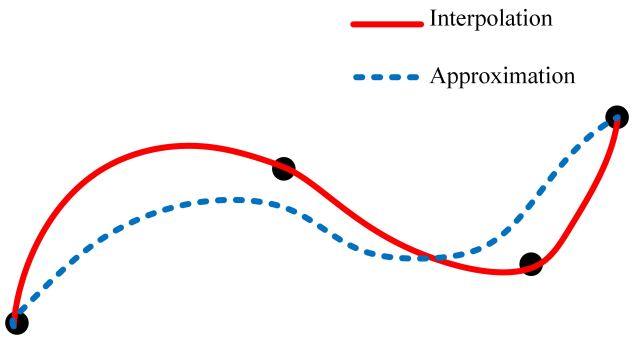


Figure 1.3 – Interpolation vs Approximation

Autre que les points de passages, les définitions de courbe peuvent inclure :

1. Des spécifications géométriques telles que la direction tangente, la courbure du chemin, $\cdots etc$.
2. Des spécifications sur les vitesses est les accélération aux points de départ et d'arrivée, Figure.1.4.

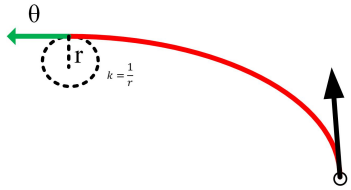


Figure 1.4 – Un chemin avec des spécifications sur les positions finales, les directions et la courbure dans le point b

Le caractère lisse du chemin est mesuré en fonction de la continuité des courbes planifiées. Deux types de continuités peuvent être distinguées :

1. **Continuité Paramétrique** : Considérons la courbe définie par morceaux pour un paramètre u varie de 0 à 3 (Fig. 1.5). La courbe est composée de trois pièces ou segments polynomiaux. Pour que la courbe soit considérée C^1 (continuité paramétrique d'ordre un), les premières dérivées des segments doivent être égales aux jonctions :

$$\vec{P}'_0(1) = \vec{P}'_1(0) \quad \text{et} \quad \vec{P}'_1(1) = \vec{P}'_2(0) \tag{1.16}$$

Pour que la courbe soit continue paramétrique d'ordre deux (C^2), les premières et secondes dérivées doivent être égales :

$$\vec{P}''_0(1) = \vec{P}''_1(0) \quad \text{et} \quad \vec{P}''_1(1) = \vec{P}''_2(0) \tag{1.17}$$

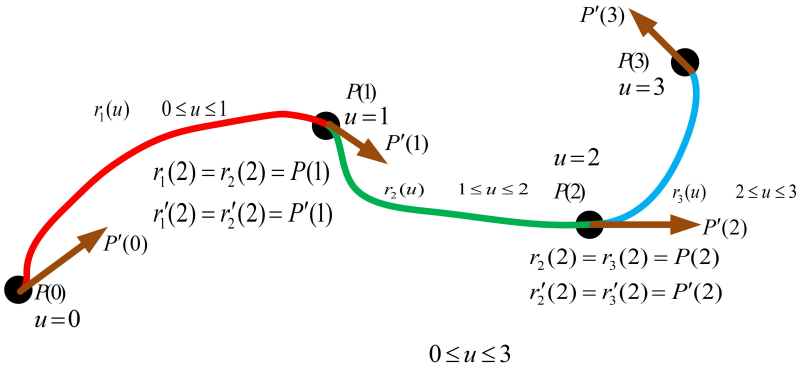


Figure 1.5 – Interpolation avec une continuité paramétrique C^1

2. **Continuité Géométrique** : Une fonction qui permet d'interpoler deux points donnés avec des vecteurs tangents attribués (unité) est appelée interpolation G^1. De plus, si les courbures données aux deux points sont également satisfaites, on parle alors d'interpolation Hermite de continuité G^2, d'où les conditions des dérivées aux limites sont exprimées sous la forme :

$$\vec{P}'_i(0) = m_i(0)\hat{t}_i(0) \quad \vec{P}'_i(1) = m_i(1)\hat{t}_i(1) \tag{1.18}$$

avec m_1 et m_2 sont des paramètres quelconques. L'utilisation de la continuité géométrique a produit deux degrés de liberté (m_1 et m_2) supplémentaires par segment de courbe $P(u)$. Ces degrés de liberté peuvent être utilisés comme poignées de forme pour contrôler la forme de la courbe sans perte de continuité géométrique.

Outre les propriétés permettant à un planificateur de définir des chemins faisables à base des fonctions définies par morceaux, nous avons également besoin des propriétés facilitant le processus de planification d'un tel chemin et de le modifier partiellement pour forcer les paramètres locaux d'un tel chemin à satisfaire des contraintes bien définies dans une zone spéciale d'espace de travail. C'est ce qu'on appelle la propriété de **Localisme**. Le Localisme est une propriété qui devient importante lors de la modification d'un segment d'une courbe à plusieurs segments. Nous dirons qu'une courbe a un Localisme si les modifications apportées à un seul segment de la spline n'affectent qu'un voisinage limité de ce segment particulier et n'entraînent pas de modifications pour tous les segments de la spline. Cette propriété offre la possibilité de modifier localement la forme d'une spline (par exemple, pour contourner un obstacle) sans qu'il soit nécessaire de reconstruire sa forme globale.

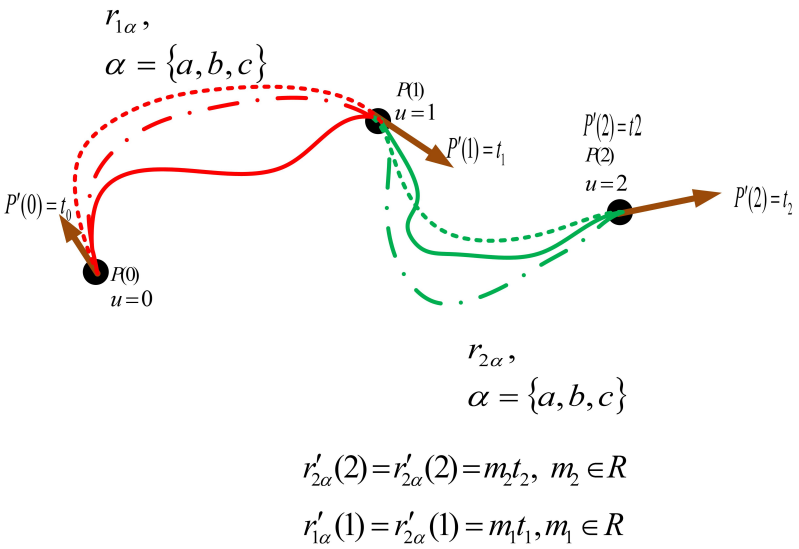


Figure 1.6 – Interpolation avec une continuité paramétrique G^1

1.4.2 définition du chemin

Étant donné une représentation appropriée d'un système robotisé et une map pour l'environnement dans lequel il évolue, le problème de conception des chemins sous forme paramétrique (1.14) est généralement reformulé comme un problème d'optimisation défini comme :

SPPProblem :

Étant donnée une représentation appropriée d'un système robotique et un plan de l'espace de travail, définir le type et l'ordre minimum de l'équation paramétrique $P(u)$ ***qui minimise la fonction de coût*** J ***:***

$$min\ J$$

sous les considérations :

(i)- Des conditions aux limites.

(ii)-Degré de continuité, géométrique ou paramétrique.

Le problème (Smooth Path Planning Problem) est basé sur le principe de définir une fonction P qui respecte les conditions (i et ii) et assure l'optimisation d'une certaine fonction de coût noté J.

Considérant la fonction de coût J, des applications classiques (souvent pour des corps rigides sans consiération de contrainte non-holonome) nécessitent de définir le chemin le plus court, car l'exécution de mouvements redondants prend plus de temps, plus d'énergie, $\cdots$. Dans autres applications, le chemin doit être forcé de garder une certaine marge de sécurité minimale par rapport aux obstacles afin de réduire les risques de collision avec ces obstacles [Geraerts2007, Braghin2008, Chan2008]. L'optimisation de la courbure du chemin planifié représente aussi un défi fondamental dans plusieurs applications

robotiques sous des contraintes non-holonome [Kanayama1989, Berglund2008]. La minimisation du régime vibratoire représente un critère fondamental qui doit être considéré lors de la conception d'un chemin faisable pour un système avec flexibilité dans sa structure [Cartmell2003]. Donc, un problème typique de planification de chemin est défini dans le plus souvent des cas comme une fonction de l'un de ces critères ou la combinaison entre eux.

En ce qui concerne avec le problème de définitions de la fonction P, plusieurs algorithmes ont été proposés dans la littérature. Ces algorithmes peuvent être principalement classifiés en trois catégories :

Fonctions d'interpolations

En raison de l'équilibre qu'elle possède entre la simplicité et la flexibilité, cette famille des courbes est largement utilisée pour assurer que le chemin planifié passe par tous les points de passage, voir par exemple [de Boor, Farin1997]. Généralement, cette classe de fonction est disponible dans la littérature sous différentes représentations (polynomiales, rationnelles, à base des fonctions Bersteins, à base des fonctions Hermites, $\cdots$) ce qui lui donne des degrés supplémentaires pour résoudre différents types de problèmes selon le critère de performance considéré. Les fonctions d'interpolation peuvent être classifiées en interpolation paramétrique (avec contraintes paramétriques sur les dérivées des points de passage) et interpolation géométrique (avec des contraintes géométriques dans les points de passage).

Les fonctions **"d'interpolation paramétrique"** sont initialement introduites dans de nombreux domaines de la robotique pour résoudre le problème de planification pour un système holonome. Le chemin est obtenu dans ce cas comme des segments définis par morceaux à base de fonctions polynomiales de Lagrange ou des fonctions splines [Siliciano2010, Khalil2000]. Ces segments sont définis pour interpoler des points de passages ou pour relier des segments spécifiques, tels que les lignes droites, les segments circulaires, $\cdots$. L'allure du chemin planifié est contrôlée par le choix des points de passages (nombre et positions), le type de la fonction d'interpolation et son degré.

La notion **d'interpolation géométrique** est initialement introduite dans le domaine de CAGD (Computer Aided Geomtric Design) par Barsky et Beatty [Barsky1983] pour la conception d'une famille de courbe notée $\beta - splines$. Cette notion de continuité géométrique est par la suite introduite dans [Piazzi2002, Lo Bianco2004a, Piazzi2007, Lo Bianco2010, Ghilardelli2014] pour définir une famille, connue sous le nom $\eta^i - Spline$, $i = \{1,2,3,4\}$, de fonctions polynomiales d'ordre 3, 5, 7 et 9 assurant une continuité géométrique G^1, G^2, G^3 et G^4 respectivement pour définir le chemin d'un système non-holonome. Une généralisation de cette approche pour résoudre le problème de planification dans l'espace Cartésien est donnée dans [Sebbane2014]. Il existe aussi une autre classe de fonctions d'interpolations géométrique connue sous le non **"d'interpolation Hermite"** [Farin2002]. Les fonctions Hermite sont principalement basées sur le principe de relier deux points sous la considération des exigences sur l'orientation et la courbure (et ses dérivées) dans le point de départ, ainsi que pour le point final. Des chemins plus complexes sont obtenus par l'interpolation de plusieurs segments. Ces techniques ont été étendues dans plusieurs directions pour résoudre différents types de problèmes selon un critère défini [Milica2018, Gašper Jaklic2011, Lu2018, Lu2015].

Fonction d'approximations

Cette famille des courbes est basée sur le principe d'interpoler les points de départ et d'arrêt et de passer de proximité des autres points (connue dans ce cas comme des points de contrôle). Quelques types courants de courbes apparaissant dans le domaine d'informatique et largement utilisés pour la planification des mouvements des systèmes robotisés sont les courbes de Bézier, les courbes B-spline et les courbes B-spline rationnels non uniformes (NURBS) (voir par exemple [Bartels1987, Farin1997], $\cdots$ et autres ouvrages de CAGD).

Les courbes de Bézier sont des fonctions pionnières dans cette classe. Elles sont des courbes polynomiales paramétriques décrites pour la première fois en 1962 par Pierre Bézier (ingénieur Arts et Métiers et Supélec à la régie Renault dans les années 1950) qui les utilisait pour concevoir des pièces d'automobiles. L'algorithme de Casteljau a permis d'inférer plusieurs propriétés très utiles dans le champ de la robotique [Farin2002]. Nous pouvons citer essentiellement les propriétés d'enveloppe convexe (convex hull), l'invariance sous des transformations affines, contrôle des conditions d'interpolation et degré élevé de continuité (C^{∞}).

La forme des courbes de Bézier est obtenue par un certain nombre de points de contrôle. Cette propriété facilite la résolution des problèmes d'interpolation entre deux points avec la considération des contraintes géométriques et paramétriques. Elle permet aussi de résoudre le problème d'interpolation de plusieurs points de passages ou d'interpoler des segments spécifiques. Dans ce cas, le chemin du robot est défini comme un ensemble de segments reliés entre eux. Chaque segment est contrôlé par un ensemble de points de contrôle [Yang2008a, Yang2008b, Yang2010, Walton2003, Dai2015, JAKLIC2010]. La propriété de "d'enveloppe convexe" indique qu'une courbe reste dans la hull convexe de ses points de contrôle (Figure.1.7). Notez que cette propriété s'applique également et uniquement aux courbes approximatives. Elle rend cette classe de fonction très utile pour manipuler les systèmes robotisés dans des milieux largement encombrés avec des obstacles et sous différentes considérations.

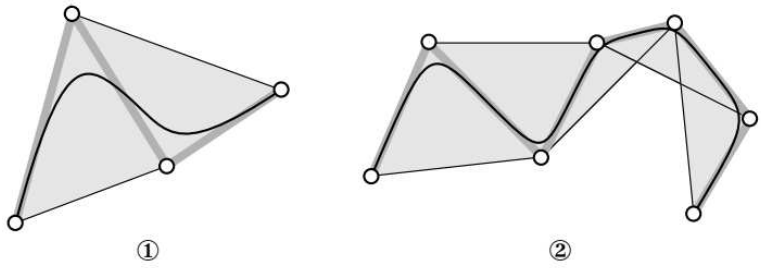


Figure 1.7 – La propriété d'enveloppe convexe pour les courbes d'approximation

Le principal inconvénient des courbes de Bézier est l'influence globale des points de contrôle sur l'ensemble de la courbe. Cela présente deux problèmes majeurs : (1) chaque modification d'un point de contrôle (insertion, déplacement, suppression) modifie globalement l'allure du chemin et (2) le temps de calcul nécessaire pour un grand ensemble de points de contrôle. Deux solutions peuvent être distinguées, l'interpolation d'un ensemble de courbe de Bezier comme a été déjà motionné ou l'introduction d'une classe de fonction approximative dite B-Spline. Le problème de continuité est sérieusement posé dans le premier cas.

Comme les fonctions de Bézier-Splines, ces chemins sont des courbes approximatives dans lesquelles les polynômes de Bernstein sont remplacés par les polynômes B-spline $B(k,d)$, d'où k est le nombre de points de passage(knot point), d est le nombre de points de contrôle (Control point) pour chaque segment [Farin2002]. Ces fonctions sont disponibles sous différentes formes (polynomiale, rationnelle,$\cdots$) préservent les principales caractéristiques des fonctions de Bézier avec une contrôlabilité locale du chemin planifié. Elles sont largement utilisées pour différentes applications de la robotique (on cite par exemple [Sankar1988, Thompson1987, Wang1988, Wang1990, Wu2009]). On peut facilement noter la popularité particulière de cette classe pour résoudre différents types de problèmes (optimisation de : la longueur du chemin, de la courbure, de la sécurité considérant l'évitement d'obstacles, en $2D$ tan que $3D$) sous différente considération (tan-que géométrique, paramétrique ou dynamiques).

Fonctions Basées sur primitives

Ce genre de fonction a été proposée essentiellement pour résoudre les problèmes de planification pour les robots mobiles planaires, i.e. $W = \mathbb{R}^2$, avec peu d'applications pour les systèmes robotisés dans l'espace Cartesien $3D$. Au début, les recherches visaient des solutions optimales pour la génération des chemins joignants des points attribués et avec la considération d'un modèle simplifié du robot mobile connu sous le nom de "voiture de Dubin". Les chemins sont constitués de segments rectilignes et d'arcs de cercle assurant ainsi la bornitude de la courbure, tandis que, la minimisation de la longueur de la courbe représentait le critère d'optimisation à respecter [Dubins1957, Dubins1961, Reeds1990]. Cependant, aux points de jonction entre les segments de ligne et les arcs de cercle, la courbure du trajet est discontinue, de sorte qu'un robot mobile qui suit ces trajets doit s'arrêter afin de réorienter ses roues. Nelson [Nelson1989], qui avait proposé deux primitives de trajectoire, des fonctions d'ordre cinq pour les manœuvres de changement du sens de mouvement et des splines polaires pour traverser des virages symétriques. Dans la même période, Kanayama et Hartman [Kanayama1989] ont également proposé la planification avec des trajectoires de courbure continue. Ils ont conçu la spirale cubique, un chemin primitif qui minimise l'intégrale de la variation de courbure mesurée le long de la courbe. Par la suite, Delingette et al. [Delingette1991] ont proposé une «spline intrinsèque», une primitive qui permet de définir un chemin avec une courbure continue définie comme une fonction polynomiale de la longueur de l'arc. Un axe de recherche commençant par Boissonnat et al. [Boissonnat1994] et continué dans [Scheuer1998] et [Kito2003] a mis en évidence l'opportunité de planifier des chemins avec une contrainte sur la dérivée de la courbure. Fraichard et Scheuer [Fraichard2004] ont présenté une méthode de pilotage, appelée *CCSteer*, conduisant à chemins composés de segments de ligne, d'arcs de cercle et de clothoïdes où le chemin global a une courbure bornée continue ainsi q'une dérivée de courbure bornée. De son côté, Reuter [Reuter1998] est allé plus loin afin d'éviter les mouvements brusques, en proposant une approche de lissage pour obtenir des chemins avec une continuité G^3, c'est-à-dire que la courbure et la dérivée de courbure sont continues le long du trajet du robot.

Comme résultat de ces travaux, un chemin optimal pour résoudre le problème de "car de Debun" est composé de segments de type linéaire, arc circulaire et des primitives avec une continuité géométrique G^1 (clothoïdes) ou G^2 (cornues spirales). Meek et Walton [Meek1989, Meek1992, Meek2004] ont proposé des algorithmes pour la conception des chemins avec trois types de primitives (arc de cercle, ligne droite et clothoïdes)

pour assurer des chemins avec une meilleure continuité géométrique, $G^{\{2/3\}}$. L'utilisation de ces primitives permet de concevoir des chemins avec une courbure continue, limitée et localement monotone (linéaire en fonction de la longueur d'arc). Cette approche souffre des problèmes de manipulations de ces primitives dans des milieux largement encombrés [Meek1989, Walton1996] et le problème d'évaluation numérique (en termes de précision et temps de calcul) des intégrales de Freisnel [Mielenz1997a, Mielenz1997b]. Plusieurs solutions ont été proposées dans la littérature pour relever ces défis. Des clothoides contrôlées avec des courbes de Bézier [Walton2012, Walton2003, Chen2017], de B-spline [Walton2005] et de cubique rationnelle [Dietz2004, Dietz2008] ont été proposées. La capacité des fonctions polynomiales $\eta - splines$ pour l'approximation des primitives, lignes droites, arcs de cercle, clothoïdes et cornue spirale, est discutée dans [Lo Bianco2004a,Lo Bianco2010]. Le problème d'interpolation Hermite G^1 sous la contrainte de minimisation de la longueur d'arc a été considéré dans [Bertolazzi2012].
Dans l'intention de définir des fonctions assurant des transitions lisses entre des primitives de type arc de cercle et ligne droite , on peut aussi mentionné les travaux de [Farouki2012, Habib2007, Walton2007] qui sont basés sur un type de courbes noté " Hodographe de péthaghore (PH -Pythagorean-hodograph-)". Les courbes (PH) ont une propriété distinctive de la vitesse paramétrique, c'est-à-dire la dérivée de la longueur de l'arc par rapport aux paramètres de la courbe, est simplement un polynôme, plutôt que la racine carrée d'un polynôme [Farouki1990, Farouki2008]. Cette propriété clé confère plusieurs avantages de calcul importants aux courbes de PH, en termes de mesure de longueur d'arc, de contrôle de mouvement en temps réel et de planification des trajectoires spatiales, pour plus de détails les lecteurs peuvent consulter [Farouki2014, Farouki2017, Giannelli2016].
Retournant vers le problème général de conception des primitives spéciales pour la planification des chemins lisses, des fonctions logistiques sont utilisées dans les dernières années par [Upadhyay2016, Upadhyay2018] pour la conception d'une nouvelle classe de primitives en forme *S* et sous la considération des contraintes tangentielles pour un système robotique planaire.

1.5 Profil du mouvement

Le deuxième défi lié au problème de la planification du mouvement est de spécifier l'évolution dynamique du robot à chaque point du chemin par une fonction temporelle, nommée profil de mouvement. Un profil de mouvement est défini comme un module qui permet de fixer les positions, les vitesses ou les accélérations -comme une fonction de temps- des axes du robot afin qu'il puisse réaliser la tâche spécifiée, tout en prenant en compte la géométrie du chemin planifié et les différentes contraintes dynamiques reliées au système. On peut supposer qu'un algorithme de planification de trajectoire prend en entrée : le chemin planifié (la fonction solution du problème "SPPProblem"), les contraintes cinématiques et dynamiques du robot. La sortie de cet algorithme est la trajectoire des articulations ou système robotique, exprimée sous la forme d'une séquence de valeurs de position, de vitesse et d'accélération. L'idée proposée le long de cette thèse consiste à choisir le temps t pour que le robot (ou la charge manipulée) arrive à une distance s spécifiée, où s : indique ici la longueur d'arc de la courbe, en utilisant les deux faits suivants :

— La distance parcourue par le robot peut être exprimée sous forme d'une intégration numérique de sa vitesse pour sélectionner une fonction de vitesse appropriée, ap-

pelée $\xi(t)$ pour $0 \leq t \leq T$, qui amène le robot à une configuration précise s à tout instant t :

$$s(t) = \int_0^t \xi(t)dt \quad ,0 \leq t \leq T \tag{1.19}$$

— Ensuite, en utilisant le fait que la longueur d'arc s est liée à n'importe quelle forme paramétrique par la relation :

$$\frac{ds}{du} = \left\| \frac{dr_d(u)}{du} \right\| \tag{1.20}$$

où : $\left\| \frac{dr_d(u)}{du} \right\|$: indique la norme euclidienne 3D de la courbe paramétrée.

pour sélectionner le paramètre u qui amène la charge à cette longueur d'arc s.

En utilisant les deux faits précédents, la trajectoire planifiée et ses dérivées temporelles successives sont obtenues sous la forme :

$$\begin{cases} r_d(\mathrm{t}) = r_d(s(t)) \\ \dot{r}_d(t) = s\frac{dr_d(s)}{ds} \\ \dot{r}_d(t) = \dot{s}\frac{dr_d(s)}{ds} + \dot{s}^2\frac{d^2r_d(s)}{ds^2} \\ \dots \end{cases} \tag{1.21}$$

Ainsi, le problème de la conception du profil de mouvement se transforme en problème d'optimisation pour trouver la fonction de vitesse $\xi(t)$ qui assure un transfert rapide du système, tout au long d'un chemin ayant une longueur totale égale à L, en tenant compte de la dynamique du robot. Ce problème peut être formulé comme suit :

STPProblem :

Définir le profil $\xi(t)$ ***qui minimise la fonction de coût*** JJ ***:***

$$min\ JJ$$

sous les contraintes :

(i)- Cinématique/ dynamique du système. (ii)- Chemin planifié. (iii)- Limites physiques du robot.

Noter $\xi^{(k)}$ comme la fonction qui représente la dérivée d'ordre k de la fonction $\xi(t)$. La solution de l'optimisation du problème *STPProblem* nécessite la connaissance préalable de :

— La forme de la fonction de coût JJ à optimiser.

— La forme générale du chemin planifié, telle que la longueur et la courbure.

— Le paramètre k qui spécifie le degré de lissage du profil de mouvement généré $\xi(t)$. Au moins, des profils de mouvement C^2, $(k = 1)$ sont nécessaires pour assurer un déplacement lisse pour le robot.

— Les conditions aux limites du profil de mouvement :

$$\begin{cases} \xi(0) = \xi_0, \quad \xi(T) = \xi_f \\ \dot{\xi}(0) = \dot{\xi}_0, \dot{\xi}(T) = \dot{\xi}_f \\ \vdots \\ \xi^{(k)}(0) = \xi_0^{(k)}, \xi(T) = \xi_f^{(k)} \end{cases} \tag{1.22}$$

— Les limites maximales du profil de mouvement et leurs dérivées temporelles :

$$\xi(t) \leq \xi_{max}, \ldots, \xi^{(k)}(t) \leq \xi_{max}^{k} \tag{1.23}$$

Remarque 5 *Quelle que soit la dimension du système considéré, l'approche découplée permet de transformer le problème de commande en un problème unidimensionnelle qui spécifie la vitesse d'une charge (robot mobile ou l'organe terminal d'un système articulé) tout au long d'un chemin bien défini.*

Les problèmes les plus souvent considérés dans la littérature peuvent être classifiés en trois catégories [Biagiotti2008, Gasparetto2012] :

— **Mouvement en temps minimum :** Depuis plus de 60 ans [Pontryagin1962, Hermes1969, Kahn1971], la planification de la trajectoire en temps minimal est une question qui fait l'objet de nombreuses discussions dans la littérature scientifique, principalement dans l'intérêt industriel de réduire les cycles de production et l'augmentation de productivité. Pour ces raisons, il est possible de trouver une des centaines de travaux scientifiques consacrés à la présentation d'algorithmes de planification de trajectoire et la minimisation de l'indice de performance donné par le temps d'exécution, voir par exemple [Tuttle2007] .

— **Mouvement en énergie minimum :** Le terme énergie, dans la plupart des situations où il est utilisé dans l'optimisation de trajectoire, ne correspond pas à une quantité physique mesurée en joules, mais il fait partie intégrante des couples au carré et mesure l'effort des actionneurs. Cependant, dans la littérature, il est possible de trouver également des algorithmes de planification dans lesquels l'indice d'optimisation n'est qu'une énergie au sens strict. La planification de la trajectoire basée sur des critères énergétiques présente de nombreux aspects intéressants [Vergnano2012, Wigstrom2013, Salan2015, Hausler2016]. D'une part, il produit des trajectoires lisses, faciles à suivre et réduisant les contraintes sur les actionneurs et la structure du manipulateur. D'autre part, cette méthode permet d'économiser l'énergie, ce qui n'est pas une simple implication économique, car cette caractéristique peut être requise par des applications spécifiques, dans lesquelles la source d'énergie est limitée par des facteurs techniques, comme les applications de la robotique dans l'espace, en plongée Exploration ou pour des tâches militaires.

— **Mouvement en Jerk minimum :** Ce problème est apparu essentiellement si le temps d'exécution de la tâche est fixé préalablement. La génération des trajectoires ne nécessite pas de valeurs discontinues des couples d'articulation [Béarée2013, Biagiotti2012], un tel résultat est obtenu en imposant des limites supérieures au taux de variation d'accélération. Ce type de solutions nécessite le calcul de la dynamique du troisième ordre du manipulateur. La minimisation de la secousse produit des résultats positifs, tels que : réduire l'erreur pendant la phase

de suivi de trajectoire, réduire l'excitation des fréquences de résonance, limiter les contraintes sur la structure du manipulateur et sur les actionneurs, ce qui permet des mouvements naturels et coordonnés.

Outre les problèmes mentionnés ci-dessus, certains critères d'optimisation hybrides ont également été proposés dans la littérature, par exemple l'énergie et le temps d'exécution des tâches planifiés. Un planificateur de trajectoire peut aussi distingué en deux catégories selon la tâche associée, les mouvements point à point et les mouvements le long d'un chemin prédéfini.

1.5.1 Mouvement point à point (PTP)

Dans ce cas, le problème général d'optimisation consiste à déterminer la variable de commande pour un système physique, de façon à réaliser un mouvement d'un état à un autre selon un critère donné (temps, énergie, jerk), tout en étant soumis aux contraintes fixes imposées par le système (vitesse, accélération, couple, $\cdots$) et aux conditions imposées par le point de départ et d'arrêt. Si un chemin est défini préalablement, la seule connaissance requise dans ce cas est sa longueur totale. Puisque ce type de mouvement est caractérisé par des contraintes fixe tout le long du chemin, le fonctionnement du robot est caractérisé généralement par trois phases, une phase d'accélération afin que l'actionneur atteigne sa vitesse maximale, une phase de mouvement à une vitesse constante (maximale) et une phase de décélération pour que le robot atteigne sa vitesse désirée dans le point final. Selon les exigences du système, les phases d'accélération et de décélération peuvent être aussi définies comme un ensemble de sous phases (par exemple, accélération linéaire, constante, décélération linéaire). Ce type de mouvement est caractérisé alors par les instants dans lesquels le système commute d'une phase à une autre et le type de mouvement dans chaque phase. On peut distinguer les approches suivantes pour la synthèse de ce type de mouvement :

Mouvement Bang-Bang :

Le principe de conception d'une loi de mouvement "Bang-Bang" consiste à saturer la variable de commande du système (dans notre cas l'accélération de référence), ou une de ses dérivées, en commutant un certain nombre de fois d'un niveau maximal à un niveau minimal autorisés. La $n^{ème}$ dérivée pour ce type de profil peut être réalisée par la sommation d'échelons, retardés dans le temps, de différentes amplitudes :

$$s_{ref}^{(n)}(t) = \left|s_{ref}^{(n)}\right|_{\max} \cdot \left(\sum_{i=1}^{n} A_i \cdot \mathrm{H}(t - T_i)\right) \tag{1.24}$$

avec $|s_{ref}^{(n)}|$ représente la limite fixée pour la dérivée $n^{ème}$ de la position, H la fonction de Heavyside, T_i les instants de commutation des échelons d'amplitude A_i. L'instant initial T_0 sera pris égale à 0 et l'instant final sera noté $T_n = T_f$. Les limitations sur les dérivées étant supposées symétriques, les amplitudes A_i vérifient $A_i \in \{2, 1, -1, -2\}$. Un profil de mouvement de degré n est défini essentiellement par $2n+1$ instants de commutation. Cette famille de profil représente un résultat naturel d'application du principe de Pontryagin [Pontryagin1962]. Clairement, ce profil est caractérisé par les instants de commutation d'un état à un autre. Ces instants sont obtenus sous la considération des différentes limites sur les dérivées du mouvement et les conditions aux points de limites. Un exemple de

conception d'un profil de mouvement d'ordre 2, sous la considération des limites sur la vitesse et l'accélération, est donné dans le chapitre 3. Cet exemple montre la complexité de dimensionnement d'un profil de mouvement avec un degré élevé de continuité et sous différentes considérations dans les points de départ et d'arrivée. Pour dépasser ce défi, on peut distinguer deux approches, des approches dites *récursive*s et des approches *basées sur des filtres*.

Les méthodes récursives sont basées sur le principe de saturer la $n^{ème}$ dérivée du profil. Les $(n-1)^{me}$ dérivées sont obtenues alors par intégration successive. La $(n)^{me}$ dérivée reste saturée jusqu'à ce que la $(n-1)^{ème}$ atteint sa valeur maximale pour se mettre à zéro. De la même manière, la dérivée de $(n-1)^{me}$ reste saturée jusqu'à ce que la $(n-2)^{ème}$ atteint sa valeur maximale, dans ce cas la $(n)^{ème}$ dérivée est saturée à sa valeur minimale . Cet algorithme se répète récursivement, jusqu'à ce que le profil atteint sa position désirée [Biagiotti2008,Lambrechts2005]. Les principaux avantages de ces algorithmes sont la garantie : *(i)* de l'optimisation de la solution obtenue en termes de temps d'exécution de la tâche et *(ii)* de la récursivité, ce qui permet de résoudre des problèmes de planification en temps réel sous des conditions asymétriques dans les limites max des dérivées [Kröger2006, Kim2007, Kröger2010]. Pour des raisons de complexité mathématique, l'application de ces algorithmes reste limitée. Un algorithme qui assure une continuité C^3 et C^4 sous la considération des limites symétriques et conditions initiales et finales nulles est donné dans [Liu2002, Lambrechts2005]. Un algorithme basé sur une méthode de recherche binaire est proposé dans [Ezair2014] pour définir des lois de mouvement avec un degré supérieur de continuité, mais sans garantir l'optimalité de la solution proposée et avec une augmentation considérable de la complexité numérique.

Contrairement aux méthodes récursives, l'idée de la conception des profils de mouvement à partir des **filtres dynamiques** consiste à reconstruire le profil souhaité à partir des lois pré-calculées et préprogrammées (step, ramp, trapézoïdal, $\cdots$). Les filtres dynamiques pour la génération des trajectoires en temps réel peuvent être conçus de différentes manières avec des niveaux de performances et de complexité très différents. Cependant, on peut observer que les configurations de tels filtres sont basées sur deux schémas principaux : *la technique "Input Shaping" (IS)* [Singer1990, Tuttle2007] et *les filtres moyens mobiles avec une réponse impulsionnelle finie* [Kim1994, Jeon2000].

L'idée générale de conception des profils à partir de la technique (IS) consiste à convoluer l'entrée originale du système (la position désirée) avec un ensemble d'impulsions spécifiques, appelé "Shaper", afin de produire une nouvelle entrée avec les performances requises. Ces performances sont exprimées généralement en fonction de : Vibration résiduelle, robustesse, optimum, l'amplitude des impulsions et la simplicité pour une implémentation en temps réel [Murphy1992,Singh1993,Singer1999,Park2006,Singhose2009]. Tandis que, l'implémentation de cette approche dans des problèmes typique est basée sur le modèle dynamique du système et la connaissance de l'objectif associé (position désirée), plusieurs approches, plus ou moins complexes, ont été proposées pour résoudre différente classe de problème tel que : la commande sous des conditions d'incertitudes du modèle [Park2006, Cole2011] et la mise à jour dynamique du " Shaper" afin d'atteindre des objectifs de planification en ligne du profil de mouvement [Thomsen2019]. Une autre limite de cette technique est qu'elle ne permet pas de contrôler l'accélération, la vitesse et d'autres dérivées du profil au point de départ et d'arrivée, ce qui la rend inapplicable dans de nombreux domaines de conception des profils avec conditions spécifier dans le point de limites.

Les avantages des techniques de filtrage, qui permettent de moduler correctement le

spectre de fréquence d'une loi de mouvement, sont combinés aux caractéristiques des trajectoires en un temps optimal, dont les paramètres sont généralement définis dans le seul but de rendre les trajectoires conformes à des limites données de vitesse, accélération, jerk, $\cdots$. Le point clé est l'équivalence entre les trajectoires optimales avec des contraintes sur les n premières dérivées et la sortie d'une chaîne de n filtres à moyenne mobile. Par conséquent, dans ce cas, les filtres ne sont pas utilisés pour générer en ligne une nouvelle trajectoire à partir de positions initiales et finales comme les planificateurs IS, mais pour lisser une trajectoire donnée à partir d'un profil d'une vitesse trapézoïdale (Bang-Zéro-Bang en accélération) comme dans [Biagiotti2012, Béarée2013] ou à partir d'un profil d'une accélération trapézoïdale [Béarée2014]. Cette méthode permet de réduire la complexité des planificateurs récursive et de générer des mouvements sous plusieurs contraintes d'asymétriques et conditions initiales non-zéro.

Mouvement polynomial et harmonique

Le problème d'optimisation (sans contraintes) ainsi défini peut se formuler sous une forme quadratique classique : déterminer $s(t)$ minimisant la fonctionnelle J donnée par :

$$J = \int_0^{T_\mathrm{f}} \left(\frac{\mathrm{d}^n s(t)}{\mathrm{d}t^n} \right)^2 \mathrm{d}t \tag{1.25}$$

Une autre solution possible de ce problème est un polynôme d'ordre $2n-1$ définie comme :

$$s(t) = a_m s^{(2n-1)}(t) + a_{m-1} s^{(2n-2)}(t) + \cdots + a_0 \tag{1.26}$$

dont les coefficients a_i seront fonction des conditions aux points extrêmes [**?**,K.Khalil2002].

Comme le mouvement résultant consiste toujours d'un basculement entre des phases où l'accélération sera positive puis négative, cela laisse tout naturellement penser à la forme périodique des fonctions trigonométriques comme des solutions plus lisse pour la définition d'une autre classe de profil de mouvement. A priori, la fonction sinus vérifie les contraintes d'une loi de mouvement. La loi de mouvement en accélération sera ainsi définie par le paramètre d'accélération maximale Am et la durée du mouvement T_f :

$$\ddot{s}(t) = \dot{\xi}(t) = A_\mathrm{m} \sin\left(2\pi \frac{t}{T_\mathrm{f}} \right) \quad (0 \leqslant t \leqslant T_\mathrm{f}) \tag{1.27}$$

On peut imaginer de nombreuses lois de mouvement à base de fonctions trigonométriques ; par exemple, la loi en sinus carré d'accélération [Boutalbi2017, Béarée2007].

Remarque 6 *Les algorithmes décrits dans cette section représentent les plus efficaces et les plus simples solutions pour garantir l'optimalité du mouvement généré. Cependant, elles ne considèrent pas les contraintes reliées aux chemins planifiés ni à la dynamique du système au cours du mouvement. Ce dernier problème est pratiquement dépassé dans plusieurs cas par la considération de certaines caractéristiques intrinsèques des systèmes robotiques, évidemment, la caractéristique de linéarisation par bouclage statique et dynamique. En raison de l'intérêt double de cette approche dans les cadres de la planification du mouvement et du suivi des trajectoires générées, nous discuterons ce point au chapitre 2. Le problème de la planification avec définition préalable du chemin est discuté dans la section suivante.*

1.5.2 Optimisation du mouvement le long d'un chemin prédéfini

Plusieurs approches ont été proposées dans la littérature pour inclure la nature du chemin planifié et la dynamique du système directement dans le processus de planification, voir par exemple [Kant1986, Kang Shin1985, Bobrow1985, Balkan1998, Palleschi2019], $\cdots$. Une approche qui attire l'attention des chercheurs est celle introduite par Bobrow [Bobrow1985] dans la fin des années 80 du dernier siècle. Dans cet algorithme, proposé initialement pour les bras manipulateurs avec une structure rigide, les limites sur les couples sont définies comme une fonction des variables articulaires (q et $\dot{q}$) et des variables cinématiques (s et $\dot{s}$) ce qui permet de tenir en compte tous les limites l'hors de la conception des trajectoires [Bobrow1985, Shiller1989, Pfeiffer1987]. Le mouvement du système est défini en basculant l'accélération entre les niveaux minimaux et maximaux. Contrairement au profil **PTP** (bang-bang en accélération), les limites d'accélération ne sont plus constantes dans ce cas, mais sont définies comme une fonction de la position et la vitesse actuelles du robot. Cette méthode a été étendue pour inclure des contraintes d'ordres élevés telles que les forces dynamiques maximales que la charge utile peut tolérer, les limites sur les dérivées d'ordre supérieures pour la vitesse (Jerk et Snap) et les couples. Cette approche a connue également une grande pénétration dans plusieurs domaines d'application de la robotique, tels que les systèmes flexibles, les robots mobiles (planaire, aérienne, sous marin), les systèmes chaotiques [Shiller1989, Tran2009, Antonelli2009, Lipp2014]. Tandis que le problème de planification consiste principalement à définir les instants dans lesquels la $n^{ème}$ dérivée du profil de déplacement du robot commute d'un niveau à un autre, cette approche est proposée de côte à côte avec plusieurs algorithmes d'optimisation sous différentes considérations (minimisation du temps, d'énergie, Jerk) et différents niveaux de complexité. Comme toutes les méthodes conçues à base d'un modèle de référence, cette approche souffre le problème de connaissance des paramètres dynamiques (masses, Inerties, centres de masses, $\cdots$) ce qui représente un défi fondamental dans le domaine de la robotique.

On peut citer aussi les travaux de Lo Bianco [Lo Bianco2004b, Lo Bianco2004c, Lo Bianco2005, Lo Bianco2006, Lo Bianco2013] qui a proposé un algorithme d'optimisation qui a comme entrées la courbure d'un chemin planaire et un profil de mouvement a vitesse bien définie à partir des fonctions polynomiales pour obtenir une vitesse qui permet de minimiser le jerk le long du mouvement d'un car de Dubin dans un temps bien spécifier. L'inconvénient principal de cette approche est qu'elle s'applique seulement pour les robots mobiles et ne considère que les contraintes cinématiques sans considération des limites dynamiques (couples et forces).

1.5.3 Mise en échelle des trajectoires " Trajectories scaling"

Une autre notion reliée aux problèmes de planification des mouvements pour les systèmes robotiques est la notion de "mise en échelle " ou de "scaler" une trajectoire (**"Trajectory scaling"**) [Gerelli2009, Lo Bianco2017]. Cette opération est effectuée généralement à l'aide d'un contrôleur secondaire pour modifier une trajectoire prédéfinie afin d'atteindre des objectifs non inclus dans le problème initial de planification, pour vérifier la faisabilité d'une trajectoire prédéfinie dans des conditions de fonctionnement réelles ou pour assurer que les commandes délivrées par le robot reste toujours dans ces conditions normales de fonctionnement lorsque la commande ou l'une de ces dérivées sont saturées au cours du fonctionnement du robot [Moon1991, De Luca2002, Chevalle-

reau2003].
Les deux dernières classes s'intéressent beaucoup plus avec le problème de faisabilités des trajectoires optimales. On peut interpréter cet intérêt par trois raisons principales. Premièrement, les planificateurs optimaux ne prévoient pas les effets des perturbations externes qui agissent le fonctionnement réel d'un système robotique et, en tant que tels, ne gèrent pas les incertitudes des modèles, surtout pour les méthodes développées à partir du modèle du système telles que la IS (input Shaping) et la méthode de **Bobrow**. Par conséquence, les perturbations et les erreurs de modélisation sont ensuite compensées par une rétroaction qui génère des forces généralisées qui, étant différentes de ceux initialement prévus, pourraient saturer les actionneurs. Deuxièmement, ils ne peuvent pas être utilisés dans des applications nécessitant une planification en temps réel, telle que, par exemple, le suivi d'un objet mobile dont la trajectoire n'est pas connue a priori. Finalement, une loi de commande optimale implique une exploitation maximale d'énergie du système, ces à dire, fonctionner le robot dans ces limites physiques ce qui augmente les risques de saturer ces actionneurs. Pratiquement, on peut distinguer des saturations cinématiques ou dynamiques. Les buts d'un scaler peuvent être obtenus à partir de la modification de la loi du mouvement ou, rarement, par la modification du chemin initial de mouvement [Lange2014]. Cette approche est vue également comme un moyen équivalent à la commande par rétroaction pour assurer la précision d'exécution des tâches planifiées et de manipuler des problèmes d'incertitudes et des perturbation externes [Gerelli2009,Sun2016,Lange2016]. Des scalers hors ligne ont été proposés pour le traitement du problème des incertitudes de modélisation [Hollerbach1984, Wigstrom2011]. Le comportement réel du système robotique est étudié à l'avance par l'exécution de trajectoires de test qui sont ensuite utilisées pour générer des trajectoires *quasi-optimaux* dans un temps qui évitent les saturations des actionneurs. La situation devient plus complexe lorsque les trajectoires doivent être modifiées en ligne, par exemple dans le cas de suivi d'un robot de référence ou pour gérer le problème d'imprécision dans le temps d'échantillonnage. Des propositions sont apparues dans [Chiaverini1999,Sugie2003,Antonelli2003,Lo Bianco2017] pour des robots soumis à des contraintes cinématiques ou pour des systèmes soumis à des limites dynamiques (couples, forces, ...) dans [Dahl1990, Lo Bianco2011].
Cette technique peut être aussi utilisée en exploitant les avantages des planificateurs PTP en termes d'efficacité et de simplicité de calcul pour inclure totalement ou partiellement la dynamique du système et/ou le chemin planifié. Les profils de mouvement Bang-Bang 'ordre n obtenus à partir d'un système de filtrage ne sont que des exemples typiques d'application de ce principe dans le but est d'inclure des exigences sur des dérivées d'ordre supérieur à partir d'un profil un peu simple. La graduation des trajectoires peut être obtenue par le forcement d'une trajectoire prédéfinie a passer d'un ensemble de points de passage avec des contraintes bien définies sur la vitesse, l'accélération, $\cdots$ [Biagiotti2008]. Un autre point de vue consiste à modifier les limitations physiques sur la vitesse et ces dérivées en fonction du chemin planifié [Boutalbi2019]. Cette méthode permet d'approcher des solutions optimales avec des niveaux simplifiés ou gradués de complexité.

1.6 Conclusion

La planification de trajectoire est un problème fondamental pour les applications robotiques. La capacité de générer des trajectoires avec des caractéristiques données est

un élément clé pour garantir l'efficacité des résultats obtenus en termes d'optimalité, de simplicité et d'adaptabilité. Deux problèmes de base ont été identifiés dans ce chapitre, planification du chemin et génération des trajectoires. Le découplage des tâches de planification, Chemin-Trajectoire, a permis d'identifier un certains nombre de solutions pour résoudre le problème global de planification des mouvements efficaces avec la considération de différents types de critères sous différentes contraintes.

2

Commande des systèmes non-linéaires

Ce chapitre survole les techniques de commande des systèmes non-linéaires incertains. Les approches de linéarisation par bouclage, les commandes par glissant d'ordre deux et les commandes adaptatives seront introduites.

Sommaire

2.1 Introduction 32
2.2 Commande des systèmes robotisés 33
2.3 Analyse des systèmes non-linéaires 34
2.3.1 Stabilité 34
2.3.2 Commandabilité 37
2.3.3 Robustesse 37
2.4 Linéarisation par Bouclage 37
2.4.1 Systèmes linéaires 38
2.4.2 Système non-linéaires 39
2.5 Commande par mode glissant 43
2.6 Commande Adaptative 45
2.7 Conclusion 47

2.1 Introduction

Après avoir planifier une tâche pour un système robotisé, la question posée, immédiatement, concerne le problème d'exécution des trajectoires planifiées. La précision, la stabilité et la fidélité d'exécution des tâches posent alors les grands défis de ce problème. La commande par rétroaction permet de répondre à ce problème globalement [Aström2008]. Le contrôle par rétroaction utilise des éléments de détection, de décision, d'actionnement et, généralement, des connaissances préalables sur la structure et/ou les paramètres du modèle commandé pour atteindre son objectif principal. Les performances du contrôleur reposent donc sur la fidélité des éléments de détection et d'actionnement attachés avec le système de commande ainsi que le degré de connaissance des paramètres de systèmes sous contrôle. On se place alors dans un problème de commande sous des conditions d'incertitudes dites paramétriques (relier aux degrés de connaissance des paramètres du système) et non paramétriques (ou non modélisées).

Dans le domaine d'automatique, il existe deux méthodes principales pour la conception d'un contrôleur efficace sous des conditions d'incertitudes :

1. Les méthodes de contrôle robustes : avec une structure généralement fixe, ces méthodes sont conçues pour compenser l'effet d'incertitudes, à condition que les paramètres incertains ou les perturbations soient bornées. dans cette classe on peut citer les approches de commande par mode glissant [Utkin2009a], la commande H_∞, la commande synergique, $\cdots$.

2. Les méthodes adaptatifs : où le contrôleur apprend et ajuste sa stratégie en fonction des données mesurées. Ces méthodes sont généralement introduites dans les cas où les paramètres du systèmes sont inconnus ou variants dans le temps [Sastry1989, Ioannou2006, Astrom1994].

D'un autre point de vu, les stratégies de commande par rétroaction peuvent être classifier en fonction de leur méthode de conception en deux grandes catégories : les méthodes basées sur modèles et les méthodes non basées sur modèles [Fless2009, Fliess2013]. Comme ces dernières n'exigent pas la connaissance du modèle, elles sont plus simples et plus robustes, mais ne garantissent pas l'optimalité des signaux de commande générés. Dans le domaine de commandes basées sur modèles, on peut distinguer plusieurs approches qui ont des grands succès dans le domaine de la robotique, telles que le backstepping, la passivité et la linéarisation par bouclage statique/dynamique, $\cdots$ [Slotine1991, Sastry1989, Khalil2000].
Après de courtes notions de bases sur l'étude des performances des systèmes robotisés et systèmes non linéaires (autonome et non-autonome), nous introduisons dans le reste de ce chapitre des techniques de commande avancées pour assurer la stabilité des systèmes non-linéaires. Premièrement, on introduit le concept de linéarisation exact des systèmes non linéaires. La linéarisation des processus mécaniques robotisés revient dans la plupart des cas à des propriétés intrinsèques du système telles que la passivité et la platitude. Pour un comportement plus robuste des systèmes commandés, les algorithmes de commande par mode glissant d'ordre deux sont ensuite donnés. Le concept de commande adaptative est introduit dans la dernière section pour résoudre le problème de commande sous incertitudes paramétriques plus grand.

2.2 Commande des systèmes robotisés

Selon la terminologie de modélisation dans le chapitre1, un système robotique peut être représenté soit par un modèle cinématique de la forme (1.7), dynamique de la forme (1.6) ou la combinaison des deux (1.4). A partir de ça, on peut distinguer pratiquement des systèmes commandés cinématiquement (vitesses ou accélérations dans plusieurs cas) et des robots commandés dynamiquement (en couples ou forces). Le choix entre eux dépend généralement de l'application en main et les capacités du système. Le modèle cinématique est plus simple que le modèle dynamique. En particulier, il ne fait pas intervenir un certain nombre de fonctions matricielles dont la connaissance précise repose sur celle des paramètres inertiels du robot et de ses actionneurs (masse, moments d'inertie, coefficients de reconduction des moteurs électriques, etc...). Pour de nombreuses applications, il n'est pas nécessaire de connaitre l'ensemble de ces termes avec précision. De plus, pour des robots à motorisation électrique (servomoteurs), il est fréquent de disposer d'un asservissement *"bas niveau"* en vitesse sur les moteurs, dont la grandeur en entrée est une vitesse désirée, et ayant pour rôle d'asservir la vitesse du moteur à celle désirée. Si cet asservissement est bien conçu, l'écart entre ces deux vitesses est toujours faible, de sorte que la vitesse désirée peut à son tour être interprétée comme une variable de commande libre. Finalement, s'il s'avéré que l'asservissement en vitesse évoque dans le point précédent, dont le rôle est de découpler la partie dynamique de la partie cinématique du robot, n'est pas présent, l'équation (1.5) peut alors être incorporée au modèle, en considérant que les couples fournis par les moteurs sont les variables de commande dont on dispose. Malgré ces avantages d'une commande cinématique, leur application reste limitée par la fréquence de fonctionnement (ou la vitesse de fonctionnement du robot). L'incorporation du modèle dynamique du robot dans la phase de conception du contrôleur permet d'assurer des niveaux élevés de précision pour l'exécution des tâches planifiées.

Exemple 1 *Pour illustrer les performances et les limites de chaque méthode, soit l'exemple d'un système dynamique de la forme :*

$$M\ddot{x} + b = F \tag{2.1}$$

Le but est de concevoir la commande F pour que x faire la poursuite d'une trajectoire désirée définie comme :

$$x_d = a\ sin(wt), 0 \leq t \leq t_f$$

dans un temps fini t_f.

Pour cela deux classes de commande ont été proposées :

— *Une commande cinématique : dans ce cas, on a supposé qu'elle existe une rétroaction de bas niveau définie comme :*

$$F = -K(\dot{x} - u)$$

où u est une vitesse de référence, K est un gain de contrôle choisi suffisamment large. Cette rétroaction est généralement équivalente à une boucle interne qui permet de réguler la vitesse d'un système motorisé. Sous cette considération la dynamique (2.1) se résume à :

$$\dot{x} = u$$

Le but de contrôle consiste à définir la vitesse u pour que $x - x_d \to 0$.

— *Une commande dynamique : Le processus est commandé dans ce cas directement par la commande F sans rétroaction interne, on peut définir alors :*

$$F = \ddot{x}_{ref} - k_1(x - x_{ref}) - k_2(\dot{x} - \dot{x}_{ref})$$

La figure 2.1 montre des résultats simulés sous l'environnement Matlab/Simulink pour $M = 5$ et $b = 0$ et différentes gammes de pulsation w. Cette figure montre les limites de la commande cinématique pour la gamme des vitesses très élevées, tandis que la commande dynamique préserve presque les même performances quelque soit la fréquence de fonctionnement.

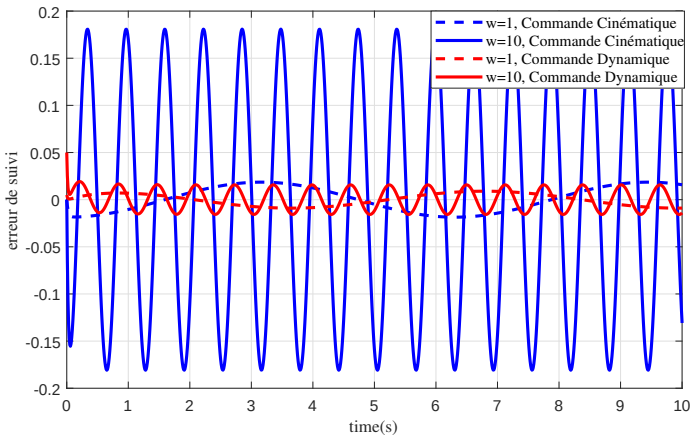


Figure 2.1 – Commandes dynamiques et cinématiques d'un système de deuxième ordre

2.3 Analyse des systèmes non-linéaires

Dans cette partie on introduit quelques notions de base pour évaluer les performances des systèmes non-linéaires. Différentes notions de stabilité au sens de Lyapunov sont introduites. La méthode de Lyapunov est largement connue dans le domaine de commande des systèmes non linéaires puisque elle permet d'analyser la stabilité du système sans connaitre explicitement la solution de l'équation différentielle qui représente l'évolution de système sous contrôle(2.3). En plus de l'analyse de la stabilité du système, elle est également utilisée comme un moyen pour concevoir des contrôleurs stables via le concept de Contrôle avec une Fonction de Lyapunov (*"CLF : Control Lyapunov Function"*) pour un système commandé de la forme (2.2) [Khalil2000, Slotine1991]. Des définitions pour les notions de commandabilité et de robustesse sont aussi introduites dans cette partie.

2.3.1 Stabilité

Quelque soit la nature de la commande pour un système robotique, on adopte la forme générale suivante pour représenter l'évolution dynamique d'un processus non-linéaire commandé :

$$\dot{x} = \frac{d}{dt}x = f(x,u,t), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m \tag{2.2}$$

d'où x est le vecteur d'état du système, u est le vecteur de commande, t est le temps. $f \in [0,\infty] \times D \to \mathbb{R}^n$ est une fonction continue par morceau localement Lipchitz dans $f \in [0,\infty] \times D \to R^n, D \subset R$ est le domaine qui contient l'origine.

Un système non commandé est donné sous la forme :

$$\dot{x} = \frac{d}{dt}x = f(x,t), \quad x \in \mathbb{R}^n \tag{2.3}$$

On verra plus loin que l'on peut se ramener à cette dernière situation (2.3) dans le cas de systèmes robotisés à commande u, par exemple au voisinage de points d'équilibre ou d'orbites périodiques, la loi de commande en boucle ouverte ou fermée ayant été fixée au préalable.

L'équation (2.3) est dite autonome ou stationnaire pour exprimer que la fonction $f(x)$ en tous point ne dépend pas l'instant de passage t. Le cas où l'équation différentielle de la vitesse $f(x,t)$ est exprimée explicitement comme une fonction du temps, le système est dit "non-autonome ou in-stationnaire".

Système autonome

Soit un système autonome de la forme :

$$\dot{x} = f(x) \tag{2.4}$$

On définit [Khalil2002] :

Définition 1 *Un point d'équilibre (ou point singulier) du champ f ou, par extension, du système (2.4), est un point $\bar{x} \in X$ tel que $f(\bar{x}) = 0$.*

Définition 2 *On dit que le point $\bar{x}$ est stable au sens de Lyapounov, ou L-stable(Lyapunov stable), si pour tout voisinage U_1 de $\bar{x}$ il existe un voisinage U_2 de $\bar{x}$ contenu dans U_1 tel que toutes les trajectoires partant d'un point quelconque de U_2 à l'instant $t = 0$ restent dans U_1 pour tout $t \geq 0$, i.e. :*

$$\forall \varepsilon > 0, \exists \delta(\varepsilon) > 0 \text{ telque } \|x_0 - \bar{x}\| \leq \delta \Longrightarrow \|x(t,x_0) - \bar{x}\| \leq \varepsilon, \forall t \geq 0$$

Définition 3 *On dit que le point $\bar{x}$ est Lyapounov-asymptotiquement stable, ou L- asymptotiquement stable, s'il est L-stable et si toutes les trajectoires partant d'un point quelconque d'un voisinage de $\bar{x}$ à l'instant $t = 0$ convergent vers $\bar{x}$ lorsque $t \to \infty$.*

Définition 4 *On dit que le point d'équilibre $\bar{x}$ est Lyapunov-exponentiellement stable s'il existe deux nombre α et λ sachant que :*

$$\forall t \geq 0, \quad \|x(t)\| \leq \alpha \|x(0)\| e^{-\lambda t}$$

dans un voisinage U_1 du point d'équilibre $\bar{x}$.

Définition 5 *Si la stabilité asymptotique (ou exponentielle) est vérifiée pour tous les états initiaux, i.e. $U_1 = R$, le point d'équilibre est dit globalement asymptotiquement (ou exponentiellement) stable.*

Théorème 1 *S'il existe une fonction V définie dans un voisinage U de $\bar{x}$, telle que :*

1. *$V(x) = 0$ pour $x = \bar{x}$ et $V(x) \geq 0, \forall x \neq \bar{x}$.*
2. *$\frac{d}{dt}V \leq 0, \forall x \neq \bar{x}$. Alors, le point d'équilibre est stable. V est noté dans ce cas sous le nom de **" Fonction de Lyapunov "**.*
3. *Si les conditions de stabilité sont satisfais et la condition (2) est forcée telle que : $\frac{d}{dt}V < 0, \forall x \neq \bar{x}$, alors, le point d'équilibre est asymptotiquement stable.*
4. *De plus, si : $V \to \infty$ pour $||x|| \to \infty$, alors le points d'équilibre est globalement asymptotiquement stable et la fonction V est dite dans ce cas **fonction de Lyapunov propre ou stricte**.*

Système non-autonome

Ces notions de stabilité peuvent être aussi étendues avec la considération d'un système non autonome de la forme :

$$\dot{x} = f(x,t) \tag{2.5}$$

Considérons $x = 0$ comme un point d'équilibre pour le système (2.5). Ensuite, on peut définir :

Définition 6 *Le point d'équilibre $x = 0$ de (2.5) est stable (au sens de Lyapunov) si :*

$$\forall \varepsilon > 0, \exists \delta(\varepsilon, t_0) : \|x(t_0)\| < \delta \Longrightarrow \|x(t)\| < \varepsilon, \forall t \geq t_0 \geq 0$$

Définition 7 *On dit que le point d'équilibre est uniformément stable si :*

$$\forall \varepsilon > 0, \exists \delta(\varepsilon) : \|x(t_0)\| < \delta \Longrightarrow \|x(t)\| < \varepsilon, \forall t \geq t_0 \geq 0$$

Les théorèmes suivants sont suffisants pour prouver la stabilité uniforme du système (2.5) :

Théorème 2 *Soit $V : [0,\infty] \times D \to R$ est une fonction continue différentiable telle que :*

$$W_1(x) \leq V(x) \leq W_2(x) \tag{2.6}$$

$$\dot{V} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x,t) \leq 0 \tag{2.7}$$

$\forall t \geq 0$ et $\forall x \in D$,d'où $W_1(x)$ et $W_2(x)$ sont des fonctions définies positives continues sur D.

Ainsi, le point d'équilibre est uniformément stable.

Théorème 3 *Supposons que les hypothèses du théorème.2 soient satisfaites et la condition de (2.7) renforcée comme :*

$$\dot{V} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x,t) \leq -W_3(x) \tag{2.8}$$

avec $W_3(x)$ est une fonction définie positive continue sur D. Ensuite, le point d'équilibre est uniformément asymptotiquement stable.
Si $D = \mathbb{R}^n$ et $W_1(x)$ n'est pas limité radialement, alors $x = 0$ est globalement uniformément asymptotiquement stable.

2.3.2 Commandabilité

Un autre paramètre qui caractérise l'évolution dynamique d'un système non-linéaire est la notion de commandabilité. Cette notion représente un point d'intersection entre les domaines de commande et de planification.

Définition 8 *[Bonnans2005] Le système (2.3) est dit commandable en temps $T \geq 0$, si est seulement si, pour p et $q \in \mathbb{R}^n$, il existe une trajectoire $t \in [0;T] \rightarrow u(t) \in \mathbb{R}^m$, dite aussi commande en boucle ouverte qui amène le système de l'état $x(0) = p$ à $x(t) = q$, c.à.d, telle que la solution du problème de Cauchy :*

$$\begin{aligned} &\dot{x} = f(x,u(t)) \text{ pour } t \in [0,T] \\ &x(0) = p \end{aligned} \tag{2.9}$$

vérifie $x(T) = q$.

Le système est dit simplement commandable lorsqu'il est commandable au moins en un temps $T \geq 0$.

2.3.3 Robustesse

En plus de la stabilité, un deuxième intérêt fondamental de la commande par rétro-action est d'assurer la robustesse du système commandé. **La robustesse** est la propriété des systèmes dynamiques de résister différentes variations du système sans dépasser les limites de tolérance prédéterminées au voisinage d'un comportement dynamique nominal [Weinmann1991]. On peut distinguer les variations suivantes : faible connaissance des processus, modèles d'ordre réduit, paramètres dynamiques incertains ou variables dans le temps, non-linéarités telles que le retard de réponse ou mauvaise connaissance de période d'échantillonnage, erreurs ou conditions initiales non nulles est inconnues, bruits internes ou externes additifs inconnus, influence de l'environnement.

2.4 Linéarisation par Bouclage

Dans le chapitre.1 (*Remarque.6*) nous avons mentionné l'existence d'une méthode qui permet d'inclure explicitement la dynamique d'un système robotisé dans le cadre de la génération optimale d'une trajectoire pour un système linéaire. Cette méthode est basée sur le principe de transformer le problème de commande d'une forme complexe non linéaire en une forme linéaire simple et commandable définie comme :

$$\dot{z} = Az + Bv, \quad z \in \mathbb{R}^{n'}, \quad v \in \mathbb{R}^m \tag{2.10}$$

via un bouclage v (dynamique/statique) non singulier et un difféomorphisme z d'espace d'état (ou de sortie) étendu. Les systèmes non-linéaires s'apparaissent dans ce cas comme des analogues des systèmes linéaires. Cette approche fournit une simplification de l'analyse, de planification et du contrôle pour les systèmes non-linéaires transformables en (2.10). Cependant, le problème de classification des systèmes non linéaires transformables en (2.10) et des transformations (z et v) nécessaires pour atteindre cet objectif pose un grand défi [Charlet1989, Krener2014]. Derrière cette problématique de transformation

entre les systèmes (2.5) et (2.10), plusieurs questions seront posées : es-qu'elle existe une telle transformation ? Quelle type s'agit elle ? Et es-qu'elle est réversible ? On peut donc reformuler cette problématique en disant que le but consiste à déterminer a quelle condition il existe un changement de coordonnées (dit aussi difféomorphisme s'il est réversible) et un bouclage approprié qui transforment localement (ou globalement) les trajectoires d'un système non linéaire en un système linéaire. Ce problème a été étudié dans la littérature scientifique de différents points de vue. D'un point de vu pratique, la réponse à une telle question permet de simplifier considérablement les problèmes de conception et de développement des commandes optimales, stables et robustes pour les systèmes non-linéaires. D'un point de vu mathématique, cette approche est basée sur la notion d'équivalence entre deux systèmes. L'intérêt de cette équivalence est illustré dans cette section en premier lieu pour un système linéaire commandable transformable en forme canonique puis sur un système non-linéaire.

2.4.1 Systèmes linéaires

Pour introduire l'idée d'équivalence entre deux systèmes et l'importance de la notion de commondabilité d'un système linéaire (2.10), on considère le système linéaire régi par la représentation d'état :

$$\dot{x} = Ax + Bu, \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m \tag{2.11}$$

avec A et B sont des matrices constantes. Alors, les résultats suivants peuvent être introduits [Bonnans2005] :

Théorème 4 *Le système (2.11) ou le pair $(A;B)$ est commandable, alors les déclarations suivantes sont équivalentes :*

1. *La matrice de commandabilité $M_c = \left[A\ AB \cdots A^{n-1}B\right]$ est a plein rang, ou $rang(M_c) = n$.*
2. *La matrice du Grammien de commandabilité $W_c = \int_0^t e^{A\xi} BB^T e^{A^T\xi} d\xi$ est définie positive pour n'importe quelle t.*
3. *Test de Popov-Belivitch-Hautus (PBH) :*
   - *La matrice $\lambda I - AB$ est à plein rang pour tous λ.*
   - *Prenant lambda une valeur propre quelconque et x le vecteur propre de gauche correspondant de A, i.e. $x^T A = x^T \lambda$, alors $x^T B \neq 0$.*

Théorème 5 *Tous système linéaire commandable à n états et m entrées, correspond à une paire $(A;B)$ est équivalent à sa forme canonique :*

$$y_1^{(\alpha_1)} = v_1, y_2^{(\alpha_2)} = v_2, \cdots, y_m^{(\alpha_m)} = v_m \tag{2.12}$$

via un changement d'état (difféomorphisme) de la forme :

$$z = Mx \tag{2.13}$$

et un bouclage de la forme :

$$u = Kz + Nv \tag{2.14}$$

avec : $z = \left[y_1, y_1^{(1)}, \cdots, y_1^{(\alpha_1-1)}, \cdots, y_m, y_m^{(1)}, \cdots, y_m^{(\alpha_m-1)}\right]$ d'où, α_i sont des entiers positives. M et N sont des matrices réversibles de dimension $(n \times n)$ et $(m \times m)$ respectivement. Les m quantités y sont appelées sorties de ***Brunovsky****.*

Théorème 6 ***(Placement de pôles)*** *Si le système est commandable, on peut placer les valeurs propres de la matrice $A+BK$ du système bouclé par un bouclage d'état $u = Kx$ adéquat, de façon arbitraire dans le plan complexe.*

Les conséquences de ces résultats sont très importantes dans la théorie de contrôle, puisque à l'aide de la forme canonique, on peut facilement planifier des trajectoires optimales sous différentes contraintes différentielles et de concevoir des commandes pour stabiliser le système commandable.

2.4.2 Système non-linéaires

On peut noter que la notion d'équivalence entre un système linéaire commandable et un système sous forme canonique permet de faciliter le traitement du problème de commande sous des contraintes. Nous allons voir dans cette partie qu'il est possible de généraliser cette propriété dans de nombreux cas des systèmes non linéaires, mais malheureusement pas pour tous. Dans la plupart des cas, cette équivalence est fournie comme un résultat naturel de certains propriétés physiques du système telles que la platitude, la passivité et d'autres.

Lemma 1 *[Charlet1989, Bonnans2005] Les deux proposition suivantes sont équivalentes :*

1. *Le système étendu*

$$\begin{aligned} \dot{x} &= f(x,u) \\ \dot{u} &= \overline{u} \end{aligned} \tag{2.15}$$

   est linéarisable par bouclage statique ($\overline{u}$ est ici la commande).

2. *Le système*

$$\dot{x} = f(x,u) \tag{2.16}$$

   est linéarisable par bouclage statique.

L'énoncé de *Lemma1* implique qu'il est toujours possible, et sans perdre de généralité, de considérer le problème de linéarisation d'un système affine de la commande de la forme (2.17) au lieu d'un système de la forme (2.2).

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2 + \cdots + g_m(x)u_m \tag{2.17}$$

Un système non-linéaire de la forme (2.17) peut être aussi défini par le paire $(f;\mathscr{G})$ où $\mathscr{G} = sp\{g_1, g_2, \cdots, g_m\}$.

Définition 9 *Deux sous systèmes $(f,\mathscr{G})$ et $(\tilde{f},\tilde{\mathscr{G}})$ sont équivalents par rétroaction si :*

1. $\mathscr{G} = \tilde{\mathscr{G}}$
2. $f+g = \tilde{f}$ *pour un champ vecteur* $g \in \mathscr{G}$ *où* $g(0) = 0,, f - \tilde{f} \in \mathscr{G}$

La première transformation proposée dans la littérature pour atteindre l'objectif de transformation d'un système non linéaire (2.17) en forme linéaire commandable avec paire (A,B) est donnée comme un changement de coordonnées dans l'espace d'état de la forme (voir par exemple [Cheng1988]) :

$$x = \phi(z), \quad x \in \mathbb{R}^n, \quad z \in \mathbb{R}^{n'} \tag{2.18}$$

d'où $\phi(z)$ est connue sous le nom du **Difféoromophisme**. On note ici que $n^{'} = n$ et $u = v$. Elle s'agit d'une linéarisation sans bouclage de l'entrée.
Dans [Brockett1978], il a été proposé d'élargir la classe de transformations (2.18) en ajoutant également des transformations à retour d'état (β est une matrice $m \times m$ réversible constante).

$$u = \alpha(x) + \beta v, \quad v \in \mathbb{R}^m \tag{2.19}$$

Cette approche a été ensuite généralisée dans [Isidori1986a, Isidori1986b] en introduisant les transformations de retour d'état :

$$u = \alpha(x) + \beta(x)v, \quad v \in \mathbb{R}^m \tag{2.20}$$

avec $\beta(x)$ est une $m \times m$ matrice inversible. Les conditions nécessaires et suffisantes qui identifient les systèmes (2.17) peuvent être transformées en (2.10) par un changement non linéaire de coordonnées de la forme (2.20) et une rétroaction d'état de la forme (2.19) sont données dans [Isidori1995].

Afin de mesurer le degré de non-linéarité de (2.17), les systèmes peuvent être transformés par (2.19) et (2.20) en systèmes linéaires du type suivant, avec $(A;B)$ est une paire contrôlable :

$$\begin{aligned} \dot{z}^{(1)} &= Az^{(1)} + Bv, & z^{(1)} &\in \mathbb{R}^p \\ \dot{z}^{(2)} &= \gamma\left(z^{(1)}, z^{(2)}\right) + \delta\left(z^{(1)}, z^{(2)}\right)v, & z^{(2)} &\in \mathbb{R}^{n-p} \end{aligned} \tag{2.21}$$

ont été étudiés d'abord dans [Krener1983] dans le cas $m = 1$, puis dans [Marino1986] dans le cas général (voir aussi [Lewis1985] et [Byrnes1986]). En particulier, il est montré qu'un ensemble d'indices de contrôlabilité $(k_i, \cdots, k_m)$ est associé à n'importe quel système non linéaire (2.17) et que l'entier $p = \sum_{i=1}^{m} k$ donne la dimension du plus grand sous-système linéarisable par rétroaction statique de la forme (2.20).
La stabilité et la robustesse de ces systèmes sont alors exprimées d'une part par la partie linéarisée et d'autre part par la partie non linéaire (zéro dynamic). Cependant le comportement de cette partie non linéaire est inconnue a priori, on ne peut constater que la stabilité ou l'instabilité du système bouclé. En outre, cette stabilité dépend du choix des coordonnées qui sont effectivement linéarisées.

Une généralisation des transformations de retour d'état statique (2.20) est donnée par les transformations de retour d'état dynamique :

$$\begin{aligned} \dot{w} &= a(x,w) + B(x,w)v, \quad w \in \mathbb{R}^q \\ u &= \alpha(x,w) + \beta(x,w)v, \quad v \in \mathbb{R}^m \end{aligned} \tag{2.22}$$

avec q est l'ordre du compensateur.

La compensation dynamique (2.22) a été introduite dans [Singh1980] pour l'étude du découplage entrée-sortie du systèmes non linéaires (2.17) avec des sorties $y = h(x), y \in R^s$, $s \in \mathbb{N}$. Dans [Isidori1986b, Charlet1989, Isidori1995], il existe des conditions suffisantes dans lesquelles un système (2.17) est simultanément découplé et linéarisable en mode entrée-sortie par un compensateur dynamique (2.22) et un difféomorphisme de la forme :

$$z = \phi(x,w) \tag{2.23}$$

D'une manière générale, vérifier qu'un système est linéarisable par rétroaction semble être un problème très difficile et reste ouvert jusqu'à présent. Il n'existe pas une méthode

crucial pour assurer si un système ou une partie est linéarisable ou non. Une des raisons majeures est qu'on ne sait pas si la dimension du retour dynamique peut être limitée a priori. Notez que le fait de prendre en compte les retours non réversibles augmente la complexité du problème, même pour les retours statiques. Jusqu'à présent, seuls quelques cas très spéciaux ont été examinés. En particulier, il existe une belle caractérisation géométrique en termes de crochet de Lie de systèmes linéarisables par rétroaction statique inversible [Slotine1991, Isidori1995].
Nous mentionnons également des résultats intéressants de [Isidori1986a] qui affirme les conditions nécessaires et suffisantes pour qu'un système soit linéarisable par bouclage statique, les résultats de [Charlet1989] qui donne des conditions nécessaires pour que des classes spécifiques de systèmes affines de la forme (2.17) soit linéarisable par bouclages dynamiques et finalement le résultat de Fliess [Fliess1993, Fliess1995] et son co-auteurs pour les systèmes désignés comme des **systèmes plats**.

Théorème 7 *[Isidori1986a] **(CNS pour bouclage statique)** Un système (f,G) est linéarisable par bouclage statique de la forme (2.20) si et seulement si :*

1. *Les distributions $\mathscr{G}$ sont involutives et de rang constant pour tous $i = 0,1,2,\cdots$ (stable par le crochet de Lie), d'où :*

$$\mathscr{G} = sp\{g_1, g_2, \ldots, g_m\}$$

2. $\dim(\mathscr{G}_{n-1}) = n \,\text{dans}\, U_0$

Théorème 8 ***(CS pour bouclage dynamique pour des classes spécifiques des systèmes non-linéaires)** [Charlet1989]*

1. *Soit un système affine (f,G) avec $m = 1$. Ce système est linéarisable avec bouclage statique, alors il est linéarisable par bouclage dynamique.*
2. *Soit système (f,G) affine tel que*
   (a) *$m = n-1 > 1$, $\sum_{i=1}^{m} k_i = n-1$*
   (b) *l'approximation linéaire autour de point d'équilibre 0 :*

$$\dot{x} = \left.\frac{\partial f}{\partial x}\right|_{x=0} x + G(0)u = Fx + Gu \quad (\ rang\ G = n-1)$$

alors, il est linéarisable par bouclage dynamique d'ordre $n-2$.

Définition 10 ***(Système plat)** Mathématiquement, on dit qu'un système $X \times U \times \mathbb{R}^m_\infty, \overline{f}$ à m entrées est différentiéllement plat, ou, plus brièvement plat, s'il est localement équivalent au système trivial $(\mathbb{R}^m_\infty, \tau_m)$, où m est le champ de vecteur trivial sur $\mathbb{R}^m_\infty$ muni des coordonnées $(y, \dot{y}, \ddot{y}, \cdots)$:*

$$\tau_m = \sum_{j\geq 0}\sum_{i=1}^{m} y_i^{j+1} \frac{\partial}{\partial y_i^{(j)}} \tag{2.24}$$

Le vecteur $y = (y_1, y_2, \cdots, y_m)$ est appelé vecteur des sorties plates.

Les principaux résultats concernant l'identification et la linéarisation des systèmes plats sont résumés dans le théorème suivant :

Théorème 9 *Soit un système non-linéaire plat avec y comme un vecteur de sorties plates, alors :*

1. *le nombre de composante d'une sortie plate est égal au nombre d'entrées indépendantes.*
2. *Ce système est localement accessible.*
3. *Ce système est linéarisable par un bouclage dynamique endogène. Inversement tout système linéarisable par un bouclage dynamique endogène est plat.*
4. *En outre, si le système admet une représentation d'état de dimension n a m entrées, ils existent des entiers $r_1, r_2, \cdots, r_m$ avec $\sum_{i=1}^{m} r_i \geq n$ tels que x et u soient données par :*

$$\begin{aligned} x &= \phi_0\left(y_1, \dot{y}_1, \cdots, y_1^{(r_1)}, \cdots, y_m, \dot{y}_m, \cdots, y_m^{(r_m)}, \cdots\right) \\ u &= \phi_1\left(y_1, \dot{y}_1, \cdots, y_1^{(r_1+1)}, \cdots, y_m, \dot{y}_m, \cdots, y_m^{(r_m+1)}, \cdots\right) \end{aligned} \tag{2.25}$$

   et tels que le système bouclé soit difféomorphe au système linéaire commandable sous forme canonique :

$$\begin{aligned} y^{(r_1+1)} &= v_1 \\ &\vdots \\ y^{(r_m+1)} &= v_m \end{aligned} \tag{2.26}$$

5. *Un système linéaire condamnable possède toujours une sortie plate.*

On peut noter donc que les sorties plates (ou le concept de linéarisation par bouclage) sont la prolongation au domaine non linéaire des sorties de Brunovsky et de la mise sous forme canonique commandable d'un système linéaire stationnaire. Le concept de platitude est très utilisé dans les situations qui exigent la génération des trajectoires optimales. Les sorties plates d'un systèmes non-linéaire permettent surtout de planifier très simplement les mouvements faisables d'un système robotisé pour aller d'un état vers un autre, tous en respectant des limites différentielles sur le mouvement du système (planification de trajectoires sous contraintes cinématiques). Elles permettent également de construire le bouclage ("feedback") qui assure le suivi asymptotique d'une trajectoire de référence arbitraire (stabilisation par placement de pôles).
L'un des principaux problèmes est l'existence d'un critère calculable pour décider si le système non linéaire (2.17) est plat. Cela signifie qu'il n'existe pas de méthode systématique pour construire une sortie plate. La situation est en fait un peu analogue à celle des fonctions de Lyapunov. Elles sont très utiles à la fois d'un point de vue théorique bien qu'il n'y ait pas de méthodologie de calcul proprement dite. Cependant, de nombreux systèmes réels sont plats, la notion de platitude a donc un grand intérêt pratique.
Le problème de suivi des trajectoires pour un système plat (linéarisable par bouclage) peut être résolu avec une loi de commande basée sur le principe d'équivalence par bouclage dynamique endogène. En réalité, cette équivalence est exprimée comme une fonction des sorties plates, ses dérivées et d'autre paramètres dynamiques (mass, inertie, ...) du système. Le succès dans la réalisation des objectifs du contrôleur dépend beaucoup de ce qui est pris en compte dans le modèle considéré, la précision des différenciateurs temps réel associé avec le contrôleur pour estimer les dérivées des sorties plates et la certitude de connaissance des paramètres du système. Malgré de l'efficacité des contrôleurs basés sur une linéarisation dynamique endogène dans le rejet des perturbations externes, comme le montrent [Hagenmeyer2003], le problème de robustesse vis-à-vis les variations paramétriques et non-paramétriques reste non résolu par des techniques de commande conventionnelles est fortement posé dans ce cas. De plus, le problème est plus compliqué dans

le cas où les paramètres du système sont totalement inconnus ou variant dans le temps. On peut noter aussi que le concept de platitude forme une approche de linéarisation dite (entrées-sortie plates) qui partage avec les techniques de linéarisation entrée-sortie le problème de stabilité du dynamique interne dénoté " dynamique de zéro ", essentiellement dans le cas où les sorties plates du système ne se confondent pas avec les sorties du système.
Malgré ces critiques reliés principalement à la robustesse de cette approche, elle peut être vue comme un bon point de départ vers des solutions plus efficace. Cette approche peut répondre au problème de planification de mouvement (ou commande en boucle ouverte dans le contexte de contrôle). Elles permettent d'assurer la génération des commandes lisses qui répondent aux différentes exigences et limites de commandes d'un système robotique. De plus, les sorties plates d'un système physique ont, dans la plupart des cas, un sens physique ce qui permet d'aider le choix de la surface de glissement dans un contrôleur à base de mode glissant pour le suivi des trajectoires dans des systèmes avec des paramètres incertains ou mal connus ou le choix d'une loi de commande adaptative dans le cas où les paramètres de système sont totalement inconnus. Les fondements de ces deux dernières techniques sont le sujet des deux sections suivantes.

2.5 Commande par mode glissant

La commande par mode glissant, introduite par V.Utkin dans la fin des années 70 du dernier siècle [Utkin1977, Utkin1993b], implique la sélection d'une variable de glissement dans l'espace d'état du système et une commande discontinue assurant des performances spécifiques au système. Cette approche est caractérisée par une grande précision de régulation, une petite sensibilité face au différents types d'incertitudes, tels que les variations paramétriques du système et les perturbations externes. Elle assure aussi une convergence en temps fini de la surface de glissement vers zéro. Le chattering est connu comme le défi fondamental face à l'implémentation pratique de cette technique. Initialement, l'application de cette technique est limitée pour les systèmes à degré relatif égale à un par rapport à la surface de glissement choisie. Une nouvelle génération des modes glissants connue sous le nom de Mode Glissant d'ordre élevé (HOSMC :*High Order Sliding Mode Control*) est introduite dans les années 90 pour étendre le concept de mode glissant conventionnel au système avec degré relatif supérieure ou égale à deux [Levant1987,Emel'yanov1993]. Afin de synthétiser une commande par mode glissant d'ordre r pour un système non-linéaire de la forme (2.2), on doit choisir une surface de glissement telle que :

$$\sigma^{(r)} = f(\sigma,\cdots,\sigma^{(r)},t) + g(\sigma,\cdots,\sigma^{(r)},t)u \tag{2.27}$$

d'où $r \in \mathbb{R}$ représente dans ce cas le degré relative de la commande u par rapport à la surface de glissement , σ et $\dot{\sigma}$, $\cdots$ sont les états du système (surface de glissement), $x_1 = \sigma$ est la sortie, u est l'entrée de commande. $f(\sigma,\cdots,\sigma^{(r)},t)$ et $g(\sigma,\cdots,\sigma^{(r)},t)$ sont des fonctions inconnues supposées bornées.

Dans ce chapitre nous avons focalisés notre étude à la conception d'un contrôleur robuste pour les systèmes vérifiant $r = 2$. La surface de glissement est donnée dans ce cas comme :

$$\ddot{\sigma} = f(\sigma,\dot{\sigma},t) + g(\sigma,\dot{\sigma},t)u \tag{2.28}$$

On suppose aussi que la première et la deuxième dérivées des fonctions f et g sont bornées.
L'objectif fondamental de la commande par mode glissant d'ordre deux est la conception d'une commande discontinue qui assure la stabilisation en temps fini de l'origine ($\sigma = \dot{\sigma} = 0$) du système (2.28), malgré la présence des fonctions d'incertitudes. Plusieurs algorithmes existants dans la littérature sont des cas particuliers de la loi de commande par mode glissant d'ordre deux généralisé définie comme [Levant2007] :

$$\begin{aligned} u = &- r_1 \operatorname{sign}\left(\mu_1 \dot{\sigma} + \lambda_1 |\sigma|^{1/2} \operatorname{sign}\sigma\right) \\ &- r_2 \operatorname{sign}\left(\mu_2 \dot{\sigma} + \lambda_2 |\sigma|^{1/2} \operatorname{sign}\sigma\right), r_1, r_2 > 0 \end{aligned} \tag{2.29}$$

d'où r_1 et r_2 sont des paramètres de commandes. On peut extraire à partir de (2.29) les lois de commande suivantes :

1. Le Twisting algorithme, d'où la commande u est donnée sous la forme :

$$u = -r_1 \operatorname{sign}(x_1) - r_2 \operatorname{sign}(x_2) \tag{2.30}$$

   Cet algorithme peut être obtenue à partir de la loi (2.29) par le choix :$\mu_1 = 0, \lambda_1 = 1, \mu_2 = 1$ et $\lambda_2 = 0$.

2. Considérant le cas où : $\mu_1 = \mu_2, \lambda_1 = \lambda_2$, la loi de commande (2.29) s'écrit alors :

$$u = -\alpha \operatorname{sign}\left(\dot{\sigma} + \beta |\sigma|^{1/2} \operatorname{sign}\sigma\right) \tag{2.31}$$

   avec $\alpha = r_1 + r_2, \beta = \lambda_1/\mu_1$. Cette loi de commande est connue sous le nom Mode glissant terminal.

3. Algorithme sous-optimale : Cet algorithme est donné sous la forme

$$u = -r_1 \operatorname{sign}(\sigma - \sigma^*/2) + r_2 \operatorname{sign}\sigma^*, \quad r_1 > r_2 > 0 \tag{2.32}$$

   d'où, σ^* est la valeur actuelle de σ détectée dans l'instant le plus proche quand $\dot{\sigma} = 0$, avec $\dot{\sigma}(0) = 0$.

Théorème 10 *[Levant2003] Les lois de commande (2.30,2.31,2.32) assurent une convergence en temps fini (stable en temps finis) de toute trajectoire (2.27) vers la surface de glissement $\sigma = \dot{\sigma} = 0$. Le temps de convergence t_{conv} est une fonction de conditions initiales $\sigma(0)$ et $\dot{\sigma}(0)$.*

L'énoncé du théorème.10 représente un résultat naturel de la propriété d'homogénéité des algorithmes (2.30,2.31,2.32) [Levant2007] et d'autre algorithme d'ordre supérieur [Levant2005, Levant2011, Livne2014, Levant2016]. Cette propriété a une grande importance dans le domaine de conception et d'analyse des commandes par mode glissant d'ordre supérieur. Également, on peut noter que pour cette classe de système avec un degré d'homogénéité négative, la stabilité asymptotique (au sens de Lyapunov) du système implique la convergence de la sortie du système vers la surface de glissement dans un temps fini. On doit noter aussi la robustesse de cette classe de système face aux perturbations. Cela implique qu'un système homogène (avec degré d'homogénéité négatif) perturbé assure l'existence d'un mode glissant réel autour de la surface de glissement choisie avec une

précision dépend des borne d'incertitudes. La convergence vers la surface de glissement est établie dans un temps fini.

Une autre classe des algorithmes de commande par mode glissant qui attire l'attention des chercheurs dans le domaine de commande par mode glissant d'ordre deux est connue sous le nom de Super-Twisting [Moreno2012] . Cette loi de commande est appliquée principalement pour les systèmes avec un degré relatif égale à un. Elle permet d'atténuer le phénomène de chattering dans ce cas. La loi de commande pour cet algorithme est définie comme :

$$u = -m_1\sqrt{|\sigma|}\,\mathrm{sign}(\sigma) + v, \dot{v} = -m_2\,\mathrm{sign}(\sigma) \tag{2.33}$$

dans le cas où la surface de glissement est donnée comme : $\dot{\sigma} = f() + g()u$, avec m_1 et m_2 sont des paramètres positifs.
Puisque la loi de commande (2.33) n'exige pas la dérivée de la surface de glissement, elle conçue aussi un bon moyen de dérivation en temps réel d'une fonction $F(t)$ [Levant1998].

Remarque 7 *Les lois de commande (2.30,2.31,2.32) peuvent être aussi utiliser pour la commande d'un système avec un degré relatif égal à un dans le but de minimiser le phénomène de chattering. L'idée dans ce cas consiste à considérer, $\dot{u}$, l'extension de la commande u, comme une nouvelle entrée de commande.*

Remarque 8 *Dans le domaine de commande par mode glissant d'ordre deux (ou généralement le mode glissant d'ordre superieur), il est toujours préférable de concevoir des différenciateurs exacts et robustes pour l'estimation de la dérivée de la surface de glissement σ. Le contrôleur est connu donc sous le nom* ***mode glissant par retour de sortie*** *[Levant2003].*

L'analyse des performances des contrôleurs (2.30,2.31,2.32 et 2.33) en termes de stabilité, de précision de régulation et de temps de convergence a attirée un grand nombre de recherches dans le cadre des systèmes (SISO). Ces performances sont en générale en compromis avec le phénomène de chattering qui représente le principal inconvénient de la mise en œuvre pratique des approches en mode glissant pour la commande des systèmes mécaniques robotisés. Ce compromis est non seulement lié à la dynamique fortement couplée et non-linéaire du système, mais également à la difficulté d'identifier un modèle approprié incluant toutes les dynamiques du système avec une connaissance exacte des paramètres de ce modèle. Dans le reste de ce mémoire on s'intéressera beaucoup plus avec le twisting algorithme, d'où l'extension des résultats de théorème pour des système Multi-Entrée Multi-Sortie (MIMO) pour des applications typiques dans le domaine de robotique.

2.6 Commande Adaptative

Une autre technique qu'attire l'attention dans le domaine de commande sous des incertitudes est la commande adaptative. Cette technique qui a une grande histoire dans le domaine de commande des systèmes incertains [Åström2014], a été établie comme un outil indispensable pour certaines classes de problèmes de contrôle des systèmes robotisés [Zhang2017]. La commande adaptative signifie l'ajustement du système de commande pour que le comportement de contrôleur soit conforme sur des nouvelles imprévisibles circonstances [Ioannou2006].

Pour une classe des commandes adaptatives, le contrôleur est défini comme la combinaison d'un estimateur de paramètres, qui génère des estimations de paramètres dans chaque pas d'échantillonnage, avec une loi de contrôle afin de stabiliser des classes de processus dont les paramètres sont complètement inconnus et/ou variant dans le temps de manière imprévisible. Cette classe est connue sous le nom **"Commande adaptative basée sur un identificateur"**. Le choix de l'estimateur des paramètres, de la loi de commande et la manière dont ils sont combinés conduisent à deux classes :

1. **Approche indirecte :** Les paramètres du système sont identifiés en ligne et utilisés pour ajuster les gains de contrôleurs de régulation.
2. **Approche directe :** Le modèle du système est défini comme une fonction d'un vecteur de paramètres inconnu θ^*. L'idée de concevoir d'une telle commande est :
   - — Spécifier une structure de commande qui permet de stabiliser (atour d'un point d'équilibre ou une trajectoire) le système dans le cas où les paramètres du système sont totalement connus.
   - — Obtenir l'équation de l'erreur de suivi.
   - — Trouver une fonction de Lyapunov et l'utiliser pour développer une loi de mise à jour des paramètres θ^* telle que l'erreur converge vers zéro.

Le choix de la nature d'estimateur permet de distinguer deux classes d'identificateurs :

1. Identificateurs basés sur un modèle de référence : La commande adaptative définie dans ce cas est dite MRAC (Model Reference Adaptive control) [Ioannou2006]. Elle exige une structure spécifique, généralement linéaire, pour le modèle du système commandé.
2. Identificateur intelligent : Le module d'identification des paramètres est basé sur les techniques d'intelligence artificiel tels que la logique floue et les réseaux de neurones [Lewis1985, Lewis1995, Lewis1999].

La deuxième classe des commandes adaptatives, non-basée sur identificateurs, consiste à remplacer le système d'identification des paramètres avec une loi d'ajustement des gains d'un contrôleur spécifique (PID par exemple [Astrom2006]), ou en commutant la loi de commande dans un groupe de contrôleurs fixes. La structure de la commande adaptative est composée d'une boucle de rétroaction et d'un régulateur à gains ajustable. Les commandes à gains programmés -gain sheduling- et la commande multi-modèles sont des cas spécifiques des commandes adaptatives non-basées sur identificateurs. Cette classe est largement utilisée dans le framework de mode glissant pour l'ajustement automatique des gains d'un contrôleur discontinu, mode glissant d'ordre un, le Super-Twisting et le twisting algorithmes, afin d'optimiser les performances des systèmes commandés en terme de précision (de suivi ou de régulation), de stabilité et d'atténuation du phénomène de chattering [Plestan2010, Utkin2013b, Utkin2013a, Taleb2013, Roy2019]. Reprenons par exemple l'algorithme classique de Twisting défini par l'équation (2.30), une loi d'adaptation des gains de commande r_1, r_2 est définie comme [Taleb2013] :

$$r_2 = \frac{1}{2} r_1 = r \tag{2.34}$$

$$\dot{r} = \begin{cases} -\alpha\lambda r & \text{if } r >_m \\ -\alpha\lambda_m & \text{if } r_{mm} < r \leqslant r_m, \quad K(0) \geqslant r_{mm} \\ \lambda_m & \text{if } r \leqslant r_{mm} \end{cases} \tag{2.35}$$

$$\alpha(t) = \begin{cases} 1 & \text{if } \forall t_j \in [t - N_t\tau, t] : \left|\sigma\left(t_j\right)\right| \leqslant \mu r\left(t_j\right)\tau^2 \\ -1 & \text{if } \exists t_j \in [t - N_t\tau, t] : \left|\sigma\left(t_j\right)\right| > \mu r\left(t_j\right)\tau^2 \end{cases} \tag{2.36}$$

$$r_m > r_{mm} > 0, \quad q > \frac{1+\beta}{1-\beta} \tag{2.37}$$

avec λ_m est un constant positif. Cette loi de commande (2.30, 2.34–2.37) permet d'ajuster les paramètres du contrôleur afin d'obtenir un régime de glissement réel autour de la surface de glissement en temps finis dans le cas où les bornes des incertitudes sont totalement inconnues. Elle permet également de minimiser le phénomène de chattering.

2.7 Conclusion

Dans ce chapitre nous introduisons quelques techniques de commande pour aborder la problématique de commande des systèmes non-linéares sous des conditions d' incertitudes.
La classification des systèmes robotiques selon le critère de linéarisabilité par bouclage permet de simplifier considérablement les tâches de planification et de conception de contrôle.
La commande robuste est utilisée lorsqu'il y a une grande dynamique non modélisée (c.-à-d. des incertitudes structurelles), des niveaux élevés de bruit avec une fréquence relativement élevée ou des changements de paramètres rapides et imprévisibles. En revanche, pour des variations de paramètres lentes ou largement prévisibles, avec une structure de modèle relativement bien connue et des niveaux de bruit limités (surtout en terme de fréquence), le contrôle adaptatif peut constituer un outil très utile pour un contrôle de haute performance.
D'une manière très simple, l'objectif du contrôle robuste est de concevoir une loi de contrôle qui garantie les spécifications de performances pour une plage donnée de valeurs des paramètres. Un contrôle robuste nécessite donc des informations a priori sur les paramètres. Le contrôle adaptatif ne nécessite aucune information a priori sur les paramètres dynamiques, mais nécessite un modèle contenant des informations sur la manière dont les paramètres affectent la dynamique du système.
Malgré les avancements dans ce domaine, la mise en ouvre pratique de ces techniques pour des applications typiques des systèmes robotisés conforte un certains nombre de défis fondamentaux reliés au système et à la tâche associée.

3

Commande d'un robot mobile à roue

En se basant sur les notions présentées dans les chapitres1 et 2, ce chapitre propose une stratégie de contrôle pour une manipulation efficace de robots mobiles non-holonomes dans des environnements encombrés d'obstacles statiques.

Sommaire

3.1 Introduction 50
3.2 Modélisation du système et formulation du problème 51
3.2.1 Modélisation du système 51
3.2.2 Formulation du problème 53
3.3 Planification du mouvement 54
3.3.1 Mouvement Point a point 54
3.3.2 Évitement d'obstacles 58
3.4 Commande en suivi de trajectoire 62
3.5 Validation numérique des résultats 63
3.5.1 Test 1 : Tâche de mouvement PTP 64
3.5.2 Test 2 : Manipulation de robot dans un environnement encombré 66
3.5.3 Test 3 : Évaluation de la commande de rétroaction 71
3.6 Conclusion 73

3.1 Introduction

Dans ce chapitre nous considérons le problème de commande d'un robot mobile différentielle comme un exemple typique d'applications pour le développement des stratégies de planification et de commande dans des milieux encombrés d'obstacles. Le contrôle du mouvement des robots mobiles à roues (WMRs) a attiré l'attention des recherches en raison des défis théoriques du problème et des applications croissantes des WMR dans de nombreux domaines tant industriels que services [Ping2013, Gonzalez2016, Kim2017, Klančar2017, Yang2018, Boutalbi2019]. Avec différents modèles et structures [Campion2008], la classe des robots mobiles à roues présente de nombreux avantages pratiques, notamment une structure simple et stable mécaniquement, un grand degré de mobilité, une efficacité énergétique, un faible coût de fabrication, $\cdots$ etc. D'un point de vue théorique, les WMRs différentielles (à deux roues différentielles, DWMR) sont des exemples typiques de systèmes non linéaires sous-actionnés avec des contraintes holonome et non-holonome.

Un algorithme basé sur le principe de découplage Chemin-Trajectoire est développé dans ce chapitre pour atteindre une planification efficace des taches associées au robot. Premièrement, un espace de travail libre est considéré. Les limites géométriques de la conception de mouvement, dans ce cas, sont réduites à la contrainte non holonome. Les chemins faisables sont obtenus en utilisant des fonctions d'interpolation cubique avec des tangentes contrôlées aux positions de départ et cible. Une fonction dépendante de la courbure est ensuite suggérée pour spécifier la vitesse du robot le long du chemin désiré tout en satisfaisant les limites physiques du système et en minimisant le temps de parcours de la tâche. Le profil de mouvement proposé est obtenu en filtrant les profils d'accélération limités classiques (bang-bang en accélération) à l'aide de filtres FIR pour obtenir des signaux de commande lisses [Lambrechts2005, Biagiotti2012, Béarée2013]. Les limites maximales d'accélération et de vitesse du profil de mouvement sont fixées en fonction de la courbure maximale du chemin.

Deuxièmement, une extension de l'algorithme ci-dessus est détaillée pour une manipulation efficace du robot dans des espaces de travail encombrés d'obstacles. Ici, le chemin du robot est spécifié par un ensemble de points de passage successifs afin d'éviter les collisions avec des obstacles. Un algorithme efficace est proposé pour estimer les tangentes, les courbures et les dérivés des courbures du chemin à tous points de passage. Ensuite, le chemin complet, avec une courbure contrôlée, est obtenu à l'aide de l'algorithme d'interpolation η^3 qui donne des courbes lisses avec une continuité géométrique du troisième ordre (G^3) [Lo Bianco2010, Piazzi2007]. La méthode proposée pour la conception des profils de mouvement accorde une attention particulière à l'optimisation du temps d'exécution de la tâche du robot. Le profil de mouvement est représenté par une chaîne de profils locaux, liés entre eux de manière lisse. Ensuite, chacun des profils élémentaires est conçu de manière à répondre aux contraintes du système, prendre en compte les caractéristiques locales de la courbe décrivant le chemin désiré du robot et assurer que le robot se déplace jusqu'à où il atteint la position cible.
Par rapport aux approches conventionnelles de planification découplée où les limites maximales des vitesses et des accélérations du profil de mouvement sont supposées être constantes le long du chemin planifié, voir par exemple [Khalil2000, Blajer2007], la méthode proposée dans ce travail suggère un profil de mouvement multi-segments avec des limites locales des accélérations et des vitesses adaptées aux changements de la courbure du chemin planifié. Bien qu'elle garantisse le respect de toutes les contraintes du système, cette méthode conduit à une optimisation significative du temps de cycle pour l'exécution

des tâches requises.

Finalement, la méthode de contrôle basée sur le concept de platitude est introduite pour le développement d'un contrôleur de suivi par une rétroaction appropriée, qui utilise les trajectoires planifiées comme signaux d'entrées [Fliess1993]. L'utilisation de cette approche est motivée par son efficacité en termes de stabilité de mouvement, de précision de suivi et de la simplicité du cinématique du système ce qui garanti une grande robustesse de l'approche [Veslin2011, Luviano-Juárez2014, Oriolo2002].

La suite de ce chapitre est divisée comme suit : Le modèle cinématique du système considéré est donné dans la deuxième partie. La troisième section donne les détails de l'algorithme de planification du mouvement point à point (PTP). La quatrième section décrit les outils nécessaires à la planification des tâches dans des milieux encombrés d'obstacles statiques. L'algorithme de contrôle par rétroaction est brièvement décrit dans la section cinq. Des tests numériques obtenus avec la plate-forme de tests, PIONEER 3DX, en laboratoire sont décrits dans la section six. La dernière section conclut le chapitre.

3.2 Modélisation du système et formulation du problème

3.2.1 Modélisation du système

La Figure.3.1 montre un modèle simplifié d'un robot mobile différentielle. Il se comporte deux roues motrices opposées avec un rayon r et une longueur d'essieu l. La position du robot sur le plan est décrite par rapport à un référentiel inertiel fixe $F_R(O,x,y)$ par le vecteur des coordonnées généralisées définie comme $[x \;\; y \;\; \theta \;\; \phi_L \;\; \phi_R]^T$,d'ou $[x \;\; y]$ est le vecteur de position, $\theta \in [-\pi \;\; \pi]$ représente l'orientation du robot mesurée dans le sens anti-horaire par rapport à l'axe des Ox et $[\phi_L \;\; \phi_R]^T$ est le vecteur des déplacements angulaires des roues gauche et droite respectivement. Considérons le référentiel $F_b(O_b,x_b,y_b)$ qui est fixé au centre (C) du robot. L'équation du mouvement est obtenue sous la considération de trois types de contraintes :

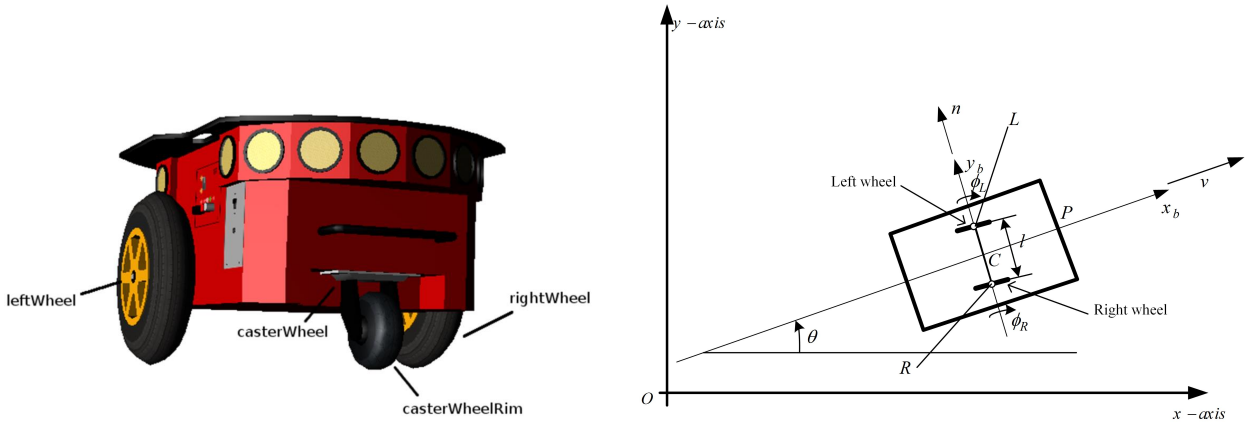
$$\begin{aligned} &\dot{y}\cos\theta - \dot{x}\sin\theta = 0 \\ &\dot{x}\cos\theta + \dot{y}\sin\theta + \tfrac{l}{2}\dot{\theta} - r\dot{\phi}_R = 0 \\ &\dot{x}\cos\theta + \dot{y}\sin\theta - \tfrac{l}{2}\dot{\theta} - r\dot{\phi}_L = 0 \end{aligned} \tag{3.1}$$

d'où :

1. La première égalité dans l'équation (3.1) se découle du fait que le vecteur vitesse du robot v est toujours défini le long de l'axe x_b. En autres termes, la composante latérale du vecteur vitesse (celle qui est normale aux roues ; c'est-à-dire le long de y_b) est nulle.
2. Les deux autres équations de contrainte sont obtenues en supposant que les roues roulent sans glissement.

Parmi les équations de contrainte données ci-dessus, il existe une contrainte holonome cachée qui peut être obtenue en soustrayant la troisième équation de la deuxième pour obtenir :

$$l\dot{\theta} = r\left(\dot{\phi}_L - \dot{\phi}_R\right) \text{ or } l\theta = r\left(\phi_R - \phi_L\right) + C \tag{3.2}$$



(a) Pioneer 3DX. (b) Modèle simplifié

Figure 3.1 – Robot mobile différentielle

Le système d'équation (3.1) peut être reformulé sous une forme matricielle :

$$\underbrace{\begin{bmatrix} \sin\theta & -\cos\theta & 0 & 0 & 0 \\ -\cos\theta & -\sin\theta & \frac{l}{2} & 0 & r \\ -\cos\theta & -\sin\theta & -\frac{l}{2} & r & 0 \end{bmatrix}}_{R(q)} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.3}$$

ou

$$R(q)\dot{q} = 0 \tag{3.4}$$

On peut facilement vérifier que la matrice $R(q)$ dans l'équation (3.3) est de rang complet, $rang(R) = 3$. Ce dernier implique que le vecteur $\dot{q}$ est un espace nul de la matrice $R(q)$, ce qui signifie aussi qu'il est possible de trouver un vecteur $V \in \mathbb{R}^n$ de vitesses auxiliaires v_1 et v_2 tel que :

$$\dot{q} = S(q)V \tag{3.5}$$

où $S(q) \in \mathbb{R}^{(m \times n)}$ c'est une matrice de rang complet et "span" l'espace nul de la matrice $R(q)$. On peut obtenir donc :

$$S(q) = \begin{bmatrix} \frac{r}{2}\cos\theta & \frac{r}{2}\cos\theta \\ \frac{r}{2}\sin\theta & \frac{r}{2}\sin\theta \\ \frac{r}{l} & -\frac{r}{l} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.6}$$

À partir des équations (3.4) et (3.5), on peut facilement montrer que le vecteur de vitesse

auxiliaire peut être défini comme :

$$V = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \tag{3.7}$$

Par une simple comparaison entre le modèle cinématique (3.5), (3.7) et le modèle cinématique bien connu des robots mobiles uni-cycles [Siliciano2010], le modèle DWMR peut être réorganisé comme suit :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.8}$$

$$\begin{bmatrix} v_R \\ v_L \end{bmatrix} = \begin{bmatrix} 1 & l \\ 1 & -l \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3.9}$$

d'ou, v et ω sont, respectivement, les vitesses de translation et de rotation . $v_R = r\dot{\phi}_R$ et $v_L = r\dot{\phi}_L$ sont les vitesses linéaires des roues gauche et droite.
Si on compare le modèle (3.8) avec la forme générale (2.17) d'un système non linéaire donnée dans le chapitre.2, on note que $f(x) = 0$. Cette classe de système, connue sous le nom des systèmes sans terme drift ou *"driftless system"*, a un certain nombre de propriétés :

— Le système (3.8) n'est plus linéarisable par un bouclage d'état statique. Cependant une partie (maximum deux états) du système peut être linéarisé par un bouclage statique pour transformer la dynamique en une forme chainée [Murray1991].

— Puisque le nombre de commande $n = 2$ et le nombre d'état $m = 3 \leq 4$, alors le système est plat et linéarisable par un bouclage d'état dynamique [Oriolo2002, Martinl1994].

— Contrairement au système (2.17) dans lequel le problème de stabilisation peut être vu comme un cas spéciale du problème générale de suivi des trajectoires [Slotine1991], ce formalisme n'est plus valide pour la classe des systèmes sous la forme (3.8). Une limitation importante de la stabilisation du système (3.8) est que la stabilité au sens de Lyapunov ne peut pas être obtenue en utilisant des lois de commande lisses et invariantes dans le temps, puisque $n \neq m$.

3.2.2 Formulation du problème

Équipé avec le modèle cinématique (3.8, 3.9), le but du reste de ce chapitre est la conception d'une stratégie de commande qui assure une manipulation efficace (en terme de temps d'exécution, simplicité, stabilité et sécurité) du robot dans des milieux encombrés. Les contraintes de commande sont définies dans ce cas comme :

— Contrainte géométriques : contraintes non-holonomes (3.1) et évitement d'obstacles.

— Contraintes cinématiques : les limites sur les vitesses et les accélérations des deux roues $v_{L(R)} \leq v_{max}$, $|a_{L(R)}| \leq amax$.

— L'exigence de développement des mouvement lisses avec des commandes lisses.

3.3 Planification du mouvement

3.3.1 Mouvement Point a point

Planification du chemin

Dans cette partie on considère le problème de planification du mouvement pour le robot mobile dans un espace de travail considéré libre, sans obstacle. Prenons en compte la contrainte non-holonome du système, le problème de planification peut être résolu convenablement avec la fonction d'interpolation définie comme [Siliciano2010] :

$$r_d(u) = [x_d(u) \;\; y_d(u)]^T \tag{3.10}$$

avec :

$$\begin{cases} x_d(u) = x_f u^3 - x_i(u-1)^3 + \alpha_x u^2(u-1) + \beta_x u(u-1)^2 \\ y_d(u) = y_f u^3 - y_i(u-1)^3 + \alpha_y u^2(u-1) + \beta_y u(u-1)^2 \end{cases} \tag{3.11}$$

Pour assurer le respect de la contrainte nonholonme et le guidage des orientations du robot dans les configurations initiale P_i et finale P_f, on doit imposer les conditions aux limites suivantes :

$$\begin{matrix} x_d'(0) = \eta_i \cos(\theta_i) \\ y_d'(0) = \eta_i \sin(\theta_i) \end{matrix} , \quad \begin{matrix} x_d'(1) = \eta_f \cos(\theta_f) \\ y_d'(1) = \eta_f \sin(\theta_f) \end{matrix} , \quad \begin{matrix} x_d(0) = x_i, x_d(1) = x_f \\ y_d(0) = y_i, y_d(1) = y_f \end{matrix} \tag{3.12}$$

d'ou :

— $P_i = [x_i \; y_i \; \theta_i]$ et $P_f = [x_f \; y_f \; \theta_f]$ sont les configurations initiale et finale, respectivement.

— $\eta_i, \eta_f \in \mathbb{R}$ *and* $\neq 0$ sont des paramètres de contrôle.

— Le paramètre d'interpolation u est choisi généralement comme : $0 \leq u \leq 1$.

— $(.)'$ représente la dérivé par rapport au paramètre u.

Ces conditions sont assurées si on défini :

$$\begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} = \begin{bmatrix} \eta \cos(\theta_f) - 3x_f \\ \eta \sin(\theta_f) - 3y_f \end{bmatrix} \tag{3.13}$$

$$\begin{bmatrix} \beta_x \\ \beta_y \end{bmatrix} = \begin{bmatrix} \eta \cos(\theta_i) + 3x_i \\ \eta \sin(\theta_i) + 3y_i \end{bmatrix} \tag{3.14}$$

d'où $\eta \in \mathbb{R}$ est un paramètre libre.

Planification de la trajectoire

Le deuxième défi relié au problème de planification de la trajectoire est comment spécifier la vitesse du robot le long de la trajectoire planifiée. Une solution directe consiste à définir la vitesse du robot comme fonction du temps $\psi(t)$ dans un intervalle de temps $0 \leq t \leq T$.

D'un point de vue physique, la distance parcourue par le robot, ou l'abscisse curviligne, est représentée comme l'intégrale de sa vitesse :

$$s(t) = \int_0^t \psi(t)dt, \text{ for } 0 \leq t \leq T \tag{3.15}$$

D'un autre point de vue, et comme été déjà vu dans le chapitre 1, l'abscisse curviligne de n'importe qu'elle forme paramétrique est :

$$s(u) = \int_0^u \| \frac{dr_d(u)}{du} \| du, , \text{ for } 0 \leq u \leq 1 \tag{3.16}$$

d'ou $\| \frac{dr_d(u)}{du} \|$ est la norme euclidien du courbe $r_d(u)$, donnée dans le cas d'une courbe planaire comme :

$$\| \frac{dr_d(u)}{du} \| = \sqrt{(\frac{dr_{dx}(u)}{du})^2 + (\frac{dr_{dy}(u)}{du})^2} \tag{3.17}$$

Donc, la formule nécessaire pour relier le chemin planifié $r_d(u)$, équations (3.10-3.14), et le profil de mouvement $\psi(t)$ (3.15,3.16) peut être vue comme la solution numérique de l'équation :

$$s - \int_0^u \| \frac{dr_d(u)}{du} \| du = 0 \tag{3.18}$$

En appliquant cet algorithme, la recherche de u pour n'importe quelle s dans n'importe quel instant t selon les équations (3.15-3.18), les fonctions temporelles du mouvement de robot sont obtenues :

$$\begin{cases} r_d(t) = r_d[u(s(t))] \\ \dot{r_d}(t) = \psi(t) \frac{dr_d(s)}{ds} \\ \ddot{r}_d(t) = \dot{\psi}(t) \frac{dr_d(s)}{ds} + \psi^2(t) \frac{d^2 r_d(s)}{ds^2} \\ \dots \end{cases} \tag{3.19}$$

$$r_d = [x_d(t)\ y_d(t)]^T \tag{3.20}$$

Les vitesses de translation et de rotation qui assurent le mouvement du robot le long du chemin (3.10-3.14) dans le cas nominal, sont définies comme :

$$\begin{cases} v_d(t) = \sqrt{(\dot{x}_d(t))^2 + (\dot{y}_d(t))^2} = v_d(s).\psi(t) \\ w_d(t) = \frac{\ddot{y}_d(t)\dot{x}_d(t) - \ddot{x}_d(t)\dot{y}_d(t)}{\dot{x}_d(t)^2 + \dot{y}_d(t)^2} = \frac{k(s)}{v_d(s)}.\psi(t) \end{cases} \tag{3.21}$$

d'ou :

— $k(s)$ représente la courbure du chemin.

— $v_d(s) = \sqrt{(x_d^{'}(s))^2 + (y_d^{'}(s))^2}$ est la vitesse de translation géométrique le long du chemin $r_d(s)$.

— $w_d(s) = \frac{y_d^{''}(s)x_d^{'}(s) - x_d^{''}(s)y_d^{'}(s)}{x_d^{'}(S)^2 + y_d^{'}(s)^2}$ est la vitesse de rotation géométrique le long du chemin $r_d(s)$.

Comme le chemin du robot est paramétré comme une fonction de l'abscisse curviligne, la vitesse de translation géométrique $v(s)$ satisfait :

$$v_d(s) = 1 \tag{3.22}$$

Ainsi :

$$\begin{cases} v_d(t) = \psi(t) \\ w_d(t) = k(s).\psi(t) \end{cases} \tag{3.23}$$

$$\begin{cases} \dot{v}_d(t) = \dot{\psi}(t) \\ \dot{w}_d(t) = \frac{\partial k}{\partial s}\psi^2(t) + k(s).\dot{\psi}(t) \end{cases} \tag{3.24}$$

Les équations (3.23-3.24) montrent bien la relation entre le chemin planifié, le profil de mouvement et l'évolution dynamique du robot.
Enfin, on peut noter aussi que la distance totale parcourue par le robot est donnée par :

$$L = \int_0^1 ||\frac{dr_d(u)}{du}||du = \int_0^T \psi(t)dt \tag{3.25}$$

Donc, on peut conclure qu'une planification convenable des mouvements (ou solution des problèmes *SPPProblem* et *STPProblem* dans le chapitre1) pour un système avec des contraintes non holonomes exige :

— La planification des chemins appropriés qui assure la continuité de la courbure de la courbe ainsi que sa dérivée. Cette courbe doit être G^3 au minimum.

— La définition d'un profil de mouvement $\psi(t)$ qui assure la commande de la vitesse du robot le long du chemin. Ce profil doit être C^1 au minimum.

— Les équations (3.23,3.24) montrent que la conception du profil du mouvement est limitée non seulement par les contraintes physique du système, mais aussi, avec les caractéristiques du courbe planifiée, c.à.d la courbure maximale du chemin. En d'autre terme, une grande vitesse pour le robot exige des chemins avec une courbure minimale.

Pour cela on a choisi de fixer η_i et η_f dans les équations (3.13, 3.14) comme :

$$\eta_i = \eta_f = \eta = ||P_f - P_i|| \tag{3.26}$$

d'où, $||P_f - P_i||$ représente la distance euclidien entre les points P_f et P_i. Ce choix assure une courbure minimale pour les chemins planifiés [Lo Bianco2010].

Conception du profil de mouvement

Dans cette étape on considère le problème de conception du profil de mouvement $\psi(t), 0 \leq t \leq T$ discuté dans les sous sections précédentes. Les défis fondamentaux dans cette partie sont la minimisation du temps d'exécution de la tâche, le respect des différentes contraintes reliées au système et à la tache planifiée ainsi que le respect des conditions aux limites, c.à.d :

$$\begin{cases} v_{L(R)} \leq v_{max}, a_{L(R)} \leq a_{max} \\ \psi(t_0) = v_0, \psi(T) = v_e \\ \dot{\psi}(t_0) = 0, \dot{\psi}(T) = 0 \end{cases} \tag{3.27}$$

d'ou v_0 et v_e sont les vitesses désirées pour le robot dans les configurations, initiale et finale, respectivement. a_{max} et v_{max} sont l'accélération et la vitesse linéaires maximales des roues gauche et droite. Ce choix doit assurer aussi un mouvement continu le long du chemin planifié.
Pour garantir le respect de toutes les contraintes, les différents limites du profile de mouvement sont fixées selon la Proposition suivante :

Proposition 1 *Les vitesses des roues gauche et droite ($v_R(t)$ et $v_L(t)$) satisfaisaient certainement les inégalités :*

$$\begin{cases} |v_R(t)| \leq v_{max}, \\ |v_L(t)| \leq v_{max} \end{cases} \tag{3.28}$$

si la fonction $\psi(t)$ respecte :

$$\psi(t) \leq \psi_{max} \tag{3.29}$$

où

$$\psi_{max} = \frac{v_{max}}{1 + l \ k_{max}} \tag{3.30}$$

avec k_{max} dénote le maximum de la valeur absolue de la courbure.

De plus, l'accélération maximale du profil de mouvement est fixée comme :

$$\dot{\psi}_{max} = a_{max} \tag{3.31}$$

Le problème se résume dans ce cas en un problème simple qui cherche à définir la fonction $\psi(t) \leq \psi_{max}, |\dot{\psi}|(t) \leq \ddot{\psi}_{max}$ qui assure un déplacement totale L en temps minimale T, prenant en compte les conditions aux limites définies comme $\psi(0) = v_0, \psi(T) = v_e$. La commande en temps optimal a pour objectif de déterminer la variable de commande qui pourra réaliser un mouvement du type point à point le plus rapidement possible (voir chapitre 1). L'approche proposée pour la génération d'un tel profil est illustré dans Fig.3.2. Elle repose sur le principe de filtrage d'un profil bang-zéro-bang en accélération pour assurer l'efficacité de planificateur en terme de temps d'exécution et de simplicité de calcul. Premièrement, une commande en temps minimale, vue dans Fig.3.3, est analytiquement obtenue selon l'énoncé d'algorithme1. Dans cet algorithme la distance S_{min} représente la distance minimale pour que le robot atteint sa vitesse maximale. Si la distance parcourue est inférieur à S_{min}, le profil de mouvement est composé dans ce cas de deux phases seulement, une phase d'accélération et une phase de décélération.

Le profil résultant est ensuite filtré par un Filtre Mobile Moyen FIR (Moving Avearage Filter) ce qui permet d'assurer un niveau élevé de continuité pour le profil de mouvement. Ce dernier est contrôlé par un temps T_j. Ainsi, le temps d'exécution de la tâche dans ce cas est donné par :

$$t_f^{'} = t_f + T_j \tag{3.32}$$

d'où, t_f représente le temps optimal d'exécution de la tache. T_j est donné comme :

$$T_j = \frac{\dot{\psi}_{max}}{\ddot{\psi}_{max}} \tag{3.33}$$

avec $\ddot{\psi}_{max}$ est le borne maximale du jerk.

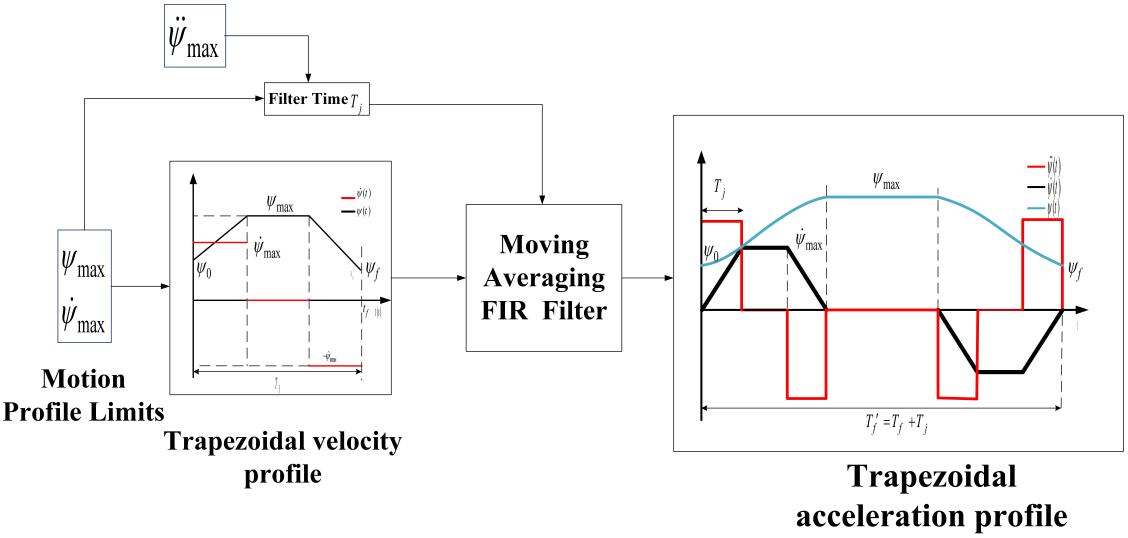


Figure 3.2 – Élaboration d'un profil de vitesse à jerk limité

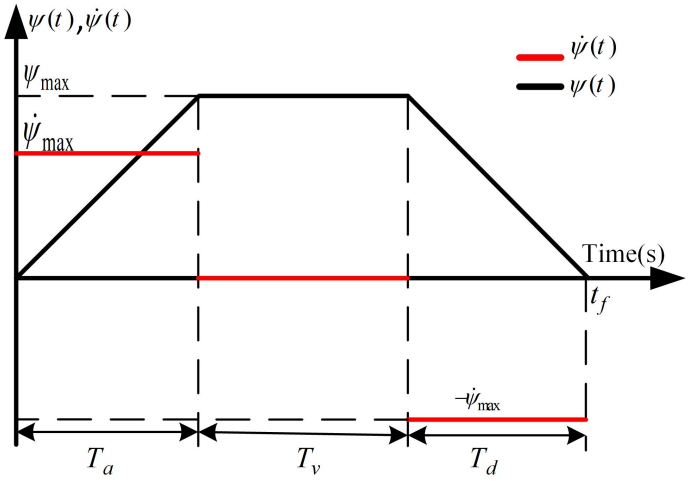


Figure 3.3 – Profile à accélération limitée

3.3.2 Évitement d'obstacles

Planification du chemin

Une extension de l'approche de planification étudiée dans la section précédente est donnée dans cette partie, pour une manipulation efficace du robot mobile dans des environnements jonchés d'obstacles statiques. Une série des points de passages est initialement sélectionnée dans l'espace libre du robot comme :

$$P = \{P_0(x_0, y_0), P_1(x_1, y_1), \ldots, P_N(x_N, y_N)\} \tag{3.34}$$

d'où, le nombre et les positions des points P_i sont choisis d'une manière à éviter tous les obstacles, le choix des point P_O et P_N doit être coïncider avec les positions initiales et finale du robot, respectivement. Le choix des tangentes dans les points initiale et finale doit être compatible avec l'orientation initiale et finale du robot. Une méthode pour estimer les orientations désirées pour le robot dans les points de passage est donnée sous la formule :

$$\theta_i = atan2(y_i^{'}, x_i^{'}) \tag{3.35}$$

Algorithm 1 : Motion profile for one manœuvrer sequence

Input : ψ_{max}, $\dot{\psi}_{max}$, L, v_0, v_e
Output : Optimal time t_f

Compute : $S_{min} = \frac{\psi_{max}^2}{\dot{\psi}_{max}}$
if $L > S_{min}$ **then**
  $T_a = \frac{\psi_{max} - v_0}{\dot{\psi}_{max}}$, $T_d = \frac{\psi_{max} - v_e}{\dot{\psi}_{max}}$, $T_v = \frac{L - S_{min}}{\psi_{max}}$
else
  $\psi_{max} = \sqrt{L\dot{\psi}_{max} + \frac{v_0^2 + v_e^2}{2}}$
  $T_a = \frac{\psi_{max} - v_0}{\dot{\psi}_{max}}$, $T_d = \frac{\psi_{max} - v_e}{\dot{\psi}_{max}}$, $T_v = 0$
end if
$t_f = T_a + T_v + T_d$

d'où, les tangentes $x_i^{'}$ et $y_i^{'}$ sont approximées en utilisant la formule de Fritich Carslon [Fritscht1980] :

$$\alpha_i^{'} = \begin{cases} m & \text{if } sign(d_{i-1}) = sign(d_i) \\ 0 & \text{otherwise} \end{cases} \tag{3.36}$$

avec :

$$\begin{cases} \alpha = \{x, y\} \\ m = 3(\delta_{i-1} + \delta i)[\frac{2\delta_i + \delta_{i-1}}{d_{i-1}} + \frac{2\delta_{i-1} + \delta_i}{d_i}]^{-1} \\ d_i = \frac{\alpha_{i+1} - \alpha_i}{\delta_i} \\ \delta_i = h_{i+1} - h_i \\ h_{i+1} = h_i + \Delta h_i \\ h_0 = 0 \\ \Delta h_i = ||P_{i+1} - Pi|| = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \end{cases} \tag{3.37}$$

Ensuite, un chemin cubique par morceaux est obtenu en interpolant les points de passage sélectionnés utilisant les équations (3.11–3.14) et les orientations obtenues à partir des équations (3.35–3.37).

Les conditions aux limites imposées (3.12) n'assurent qu'une continuité géométrique (G^1) dans le chemin planifié. Afin d'obtenir des chemins très lisses, l'algorithme de η^3-Splines proposé dans [Lo Bianco2010, Piazzi2007] peut être utilisé, ce qui garantit une continuité géométrique de troisième ordre du chemin (G^3), la continuité de la courbure et la dérivée de la courbure du chemin :

$$r_{di} = \begin{cases} x_{di}(u) = \sum_{j=0}^{7} \alpha_{ij} u^j \\ y_{di}(u) = \sum_{j=0}^{7} \beta_{ij} u^j \end{cases} \tag{3.38}$$

avec $0 \leq u \leq 1$, $0 \leq i \leq N$ et $0 \leq j \leq 7$.

Les paramètres d'interpolation α_{ij} et β_{ij} sont fixés pour assurer une courbure lisse du chemin planifié dans tous les points de passage comme montre l'équation suivante , [Lo Bianco2010, Piazzi2007], voir Annexe B :

$$\alpha_{ij} = f_1(x_i, x_{i+1}, ||P_i - P_{i-1}||, \theta_i, k_i, k_{i+1}, k_i^{'}, k_{i+1}^{'}, \eta)$$
$$\beta_{ij} = f_2(y_i, y_{i+1}, ||P_i - P_{i-1}||, \theta_i, k_i, k_{i+1}, k_i^{'}, k_{i+1}^{'}, \eta)$$

où k_i et $k_i^{'}$ sont des paramètres supplémentaires utilisés pour spécifier la courbure du chemin et la dérivée de la courbure aux points de passage P_i, respectivement.
En plus d'interpolation des points de passage, cette courbe polynomiale d'ordre sept bénéficie une paramétrisation polyvalente d'une forme simple par un vecteur supplémentaire à six degrés de liberté η, donné par le vecteur :

$$\eta = [\eta_1 \ \eta_2 \ \eta_3 \ \eta_4 \ \eta_5 \ \eta_6] \tag{3.39}$$

Dans le présent travail, les degrés de liberté sont réduits par le choix :

$$\eta = [\eta_1 \ \eta_2 \ 0 \ 0 \ 0 \ 0] \tag{3.40}$$

Comme on a vu précédemment, on peut aussi choisir : $\eta_1 = \eta_2 = ||P_i - P_{i-1}||$ pour minimiser la courbure du chemin planifié.

L'estimation nécessaire de la courbure du chemin et la dérivée de la courbure à tous les points de passage peut être obtenue selon la remarque suivante :

Remarque 9 *Pour trois points séquentiels P_{i-1}, P_i et P_{i+1}, une valeur approximative de la courbure au point de passage P_i est donnée [Liu2002] :*

$$k_i = 2\frac{det(P_i - P_{i-1}, P_{i+1} - P_i)}{||P_i - P_{i-1}||\ ||P_{i+1} - P_i||\ ||P_{i+1} - P_{i-1}||} \tag{3.41}$$

pour $i = 2, 3, \ldots, N-1$, où $det(.)$ est le déterminant de $(.)$.

On adopte une courbure (positive/négative) pour indiquer le virage/braquage (droite/gauche) dans l'espace de travail. La dérivée de la courbure au point de passage P_i cpeut être estimée récursivement comme : $k_0^{'} = 0$

$$k_i^{'} = \frac{k_i - k_{i-1}}{\Delta h_i}, i = 1, 2, \ldots, n \tag{3.42}$$

avec :

$$k_0^{'} = 0$$

où Δh_i représente la longueur du chemin entre deux points successifs P_i et P_{i-1}.

Finalement, le chemin défini par (3.38) peut être recalculé en fonction de la longueur d'arc $r_d(s)$ à l'aide de la formulation récursive suivante :

$$\begin{cases} s = \xi + s_{i-1} \\ s_0 = 0 \\ s_i = \int_0^1 ||r_{di}(u)|| du, i = 1, N \\ \xi = \int_0^u ||r_{di}(u)|| du, i = 1, N \end{cases} \tag{3.43}$$

où ξ représente le paramétrage local de la longueur d'arc de la courbe dans le i^{th} segment et s_i est la longueur d'arc du chemin à tous les points de passage. On peut noter que la longueur globale du chemin planifié peut être obtenue dans ce cas comme :

$$L = \sum_{i=0}^{N} s_i \tag{3.44}$$

Conception du profil de mouvement

Comme il a été détaillé dans la section précédente (Sect.3.3.1), l'algorithme de planification des mouvements nécessite, également, la définition d'une fonction temporelle appropriée qui assure des mouvements rapides du robot le long du chemin planifié. Une solution conventionnelle, qui est préférée dans de nombreux domaines robotiques tels que les manipulateurs articulés et les grues, consiste à définir le mouvement du robot comme une seule manœuvre allant d'une position de départ vers la cible par une trajectoire planifiée considérant l'approche détaillée dans la section précédente.
Cependant, et comme étant donné que les limites maximales du profil sont assignées en fonction des limites maximales de la courbure du chemin planifié, il est plus approprié de définir la fonction de vitesse comme une chaîne de profils locaux concaténés entre eux pour former un profil de mouvement complet et complexe. Chaque profil a été affecté à un segment spécifique du chemin désiré prenant en compte des caractéristiques locales de ce segment. Donc, les limites maximales de vitesse et d'accélération sont fixées en fonction de la courbure maximale du segment de mouvement. De plus, il est nécessaire de définir les vitesses limites des profils de mouvement à tous les points de passage, afin de s'assurer que le robot se déplace toujours, sans stop, jusqu'à il atteint sa configuration finale. Ces conditions aux limites doivent également satisfaire à l'exigence de continuité de la trajectoire conçue, ce qui implique que la vitesse et l'accélération cible du segment actuel doivent être sélectionnées comme vitesse et accélération de départ du segment suivant. De plus, pour une solution optimale, la vitesse assignée à chaque point doit prendre en compte les limites du robot ainsi que les caractéristiques du trajet local, c'est-à-dire le maximum local de la courbure. La solution proposée pour définir le profil de mouvement de la vitesse dans ce cas est montrée dans Algorithme2, où l'accélération désirée à tous les points de passage est fixée à zéro. Notez également que les vitesses désirées du robot aux positions de départ et finale sont choisies comme $v_0 = 0$ et $v_e = 0$. L'accélération maximale dans tous les segments a été fixée pour être égale à a_{max}.

Après avoir obtenu la fonction de la vitesse optimale à partir de **l'algorithme 2**, le paramétrage de la longueur de la courbe peut alors être obtenu comme l'intégrale de la fonction de la vitesse comme dans (3.16), et la représentation temporelle des trajectoires du robot est obtenue par (3.16–3.19) , en prenant en compte la représentation segmentée du chemin. Les trajectoires de référence sont trouvées dans ce cas en recherchant le segment i qui contient la valeur courante de $s(t)$ et en résolvant numériquement $u(t)$ de l'équation suivante :

$$\xi = s - s_{i-1} = \int_0^u ||r_{di}(u)||du \tag{3.45}$$

Algorithm 2 : Motion profile for multi-sequence system

Input : Desired via-points, desired robot path.
Output : Optimal motion profile $\psi(t)$

Find the maximum curvature at each segment.
Find the length L_i $(i = 1, ..., N)$ of each segment.
Find the maximum velocity v_{imax} may be reached at each segment according to **Proposition1**.
for i=1 :N **do**
  if i==1 **then**
    $v_{i0} = v_0 = 0$
  else
    $v_{i0} = v_{(i-1)e}$
  end if
  if i==N **then**
    $v_{Ne} = v_e = 0$
  else
    if $v_{(i+1)max} \leq v_{i\,max}$ **then** $v_{ie} = v_{(i+1)max}$ **end if**
    if $v_{(i+1)max} > v_{i\,max}$ **then** $v_{ie} = v_{(i)max}$ **end if**
  end if
  define $\psi(t, v_{i0}, v_{ie}, L_i)$ (see **algorithm1**)
  Smoothing the velocity of the motion profile using the moving averaging FIR filter, figure **??**.
end for

3.4 Commande en suivi de trajectoire

Après avoir obtenu les trajectoires de référence qui définissent la tâche du robot, il reste la conception de la commande par rétroaction appropriée, qui assure l'exécution précise des tâches planifiées. Les coordonnées cartésiennes des DWMRs sont connues sous le nom de sorties plates, on peut donc considérer la sortie suivante :

$$\sigma = \begin{bmatrix} \sigma_x \\ \sigma_y \end{bmatrix} = \begin{bmatrix} x_c - x_d \\ y_c - y_d \end{bmatrix} \tag{3.46}$$

d'où $[x_c\ y_c]^T$ et $[x_d\ y_d]^T$ représentent respectivement les vecteurs des positions actuelles/mesurées et désirées du robot.

Prenant la deuxième dérivée temporelle de la sortieσ pour obtenir :

$$\ddot{\sigma} = \begin{bmatrix} \ddot{e}_1 \\ \ddot{e}_2 \end{bmatrix} = \begin{bmatrix} -\ddot{x}_d + \dot{v}cos\theta - vwsin\theta \\ -\ddot{y}_d + \dot{v}sin\theta - vwcos\theta \end{bmatrix} \tag{3.47}$$

Si un nouveau vecteur de commande $u = [u_1\ u_2]^T$ est introduit comme :

$$u = \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} \dot{v}cos\theta - vwsin\theta \\ \dot{v}sin\theta + vwcos\theta \end{bmatrix} \tag{3.48}$$

ou :

$$v = \int u_x cos(\theta) + u_y sin(\theta) dt \tag{3.49}$$

$$w = \frac{-u_x sin(\theta) + u_y cos(\theta)}{v} \tag{3.50}$$

Elle réduit le système (3.47) à l'équation suivante :

$$\ddot{\sigma} = \begin{bmatrix} \ddot{\sigma}_x \\ \ddot{\sigma}_y \end{bmatrix} = \begin{bmatrix} -\ddot{x}_d + u_x \\ -\ddot{y}_d + u_y \end{bmatrix} \tag{3.51}$$

Considérant les trajectoires de référence $r_d(t)$ et le modèle dynamique (3.51), le problème de stabilisation est relativement simplifié en raison de la relation linéaire découplée entre la dynamique d'erreur de suivi et les nouvelles entrées de commande u_x et u_y. Ensuite, deux contrôleurs seront conçus pour les sous-systèmes indépendants σ_x et σ_y. Une solution populaire qui garantit la stabilisation du processus est donnée comme :

$$\begin{cases} u_x = +\ddot{x}_d - k_{p1} e_1 - k_{d1} \dot{e_1} \\ u_y = +\ddot{y}_d - k_{p2} e_2 - k_{d2} \dot{e_2} \end{cases} \tag{3.52}$$

Les polynômes caractéristiques de la dynamique des erreurs de suivi e_σ sont des polynômes de Hurwitz (stables) quand :

$$k_{pi}, k_{di} \geq 0 \tag{3.53}$$

Cette structure de contrôle, (3.46-3.53), formée d'une combinaison spécifique basée sur le concept de platitude différentielle d'une partie de feed-forward basée sur les trajectoires de référence planifiées et d'une simple partie de rétroaction PD linéaire qui prend en compte l'erreur de suivi.
Outre que sa simplicité de conception, ce contrôleur a prouvé sa robustesse face aux perturbations externes [Hagenmeyer2003]. On peut noter aussi qu'un algorithme basé sur le mode glissant d'ordre un avec un gain dépendant de l'état a été proposé dans [Boubezoula2018] pour améliorer les performances de suivi en terme de précision et de robustesse. De plus, cette loi de commande basée sur la notion de linéarisation par bouclage dynamique peut également répondre aux problèmes de suivi des trajectoires ainsi que la régulation de la position du robot à une configuration spécifique avec des commandes lisses [Oriolo2002].

3.5 Validation numérique des résultats

En utilisant le prototype PIONEER3DX du Laboratoire des Systèmes Intelligents, présenté dans la Fig 3.4, des essais pratiques ont été effectués pour évaluer les performances de l'approche de contrôle proposée. Le PIONEER 3DX est un exemple typique de plates-formes DWMR qui fonctionnent comme un serveur dans un environnement client-serveur. L'Interface Robotique Avancée pour Application (ARIA, C++) a été utilisée pour l'intégration numériques des commandes proposées. Trois exemples ont été considérés dans

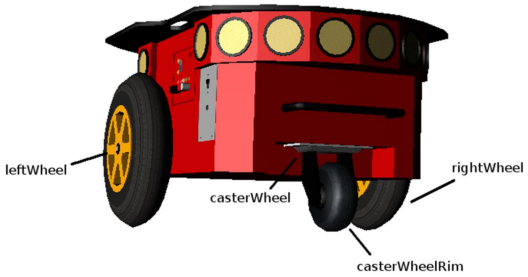


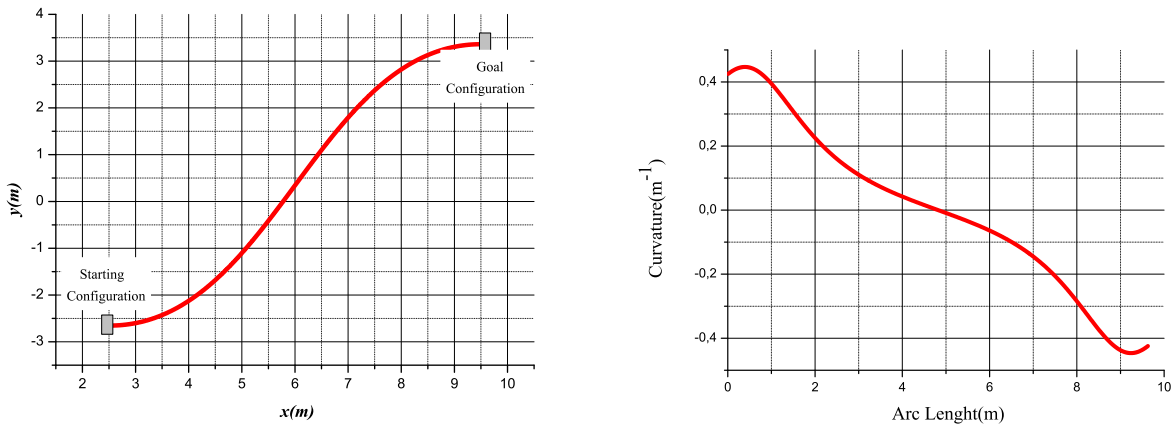
Figure 3.4 – PIONEER 3DX mobile robot.

cette partie pour illustrer les performances de l'approche proposée. Le premier exemple considère le problème de commande du mouvement PTP discuté dans Section. 3. Le deuxième exemple concerne la manipulation du robot dans un environnement encombré d'obstacles statiques. Le dernier test est concentré sur l'évaluation des performances du contrôleur par rétroaction proposé basé sur la platitude. Dans tous les essais, le temps d'échantillonnage a été fixé à $T_s = 0.01s$. Des différenciateurs numériques, basés sur l'algorithme super-twisting, ont été utilisés pour estimer les dérivés en ligne des coordonnées cartésiennes du robot [Levant1998].Les limites maximales de vitesse et d'accélération des roues gauche et droite ont été fixées dans tous les essais : $v_{max} = 0.3m/s$, $a_{max} = 0.1m/s^2$

3.5.1 Test 1 : Tâche de mouvement PTP

L'objectif de contrôle dans ce test est de déplacer le robot d'une configuration initiale $P_i = [2.5121, -2.6533, 0]$ à une configuration finale $P_f = [9.5040, 3.3621, 0]$ dans un espace de travail libre.

Le chemin planifié et sa courbure correspondante, qui répond aux contraintes nonholonome (3.10-3.14)), sont montrés dans la Fig. 3.5. Ce chemin est caractérisé par une longueur totale de $L = 9.6287m$ et une courbure maximale $k_{max} = 0.4469m^{-1}$. Il est également caractérisé par une courbure bornée et lisse.



(a) (b)

Figure 3.5 – Tâche planifiée : (a) Chemin désiré et (b) Courbure du chemin

Une tâche du mouvement de repos à repos a été considérée, c'est-à-dire que le robot devrait commencer son mouvement avec une vitesse nulle et l'achever avec une vitesse nulle. Selon la proposition1 et **Algorithm 1**, les caractéristiques principales du profil du mouvement, illustré à la figure 3.6, sont obtenues comme :
$\psi_{max} = 0.2536ms^{-1}$, $\dot{\psi}_{max} = 0.1ms^{-2}$, $T_a = T_d = 4s$ et $T_j = 4s$.
Cela implique que la vitesse maximale de translation pouvant être atteinte par le robot soit égale à $\psi_{max} = 0.2536ms^{-1}$. Ainsi, le temps d'exécution total de la tâche est obtenu comme suit : $T = 44.9s$.
La paramétrisation de la courbe par longueur d'arc est obtenue en intégrant le profil du mouvement de la vitesse présenté la Fig.3.6 selon l'équation (3.15). Ensuite, les trajectoires de référence temporelles, illustrées dans la figure 3.7, le long du chemin désiré sont obtenues par les équations (3.16- 3.19). Pour toute valeur donnée de s, le paramètre u a été résolu à partir de (3.18) en utilisant une méthode numérique non linéaire combinant les algorithmes de bissection et de Newton pour obtenir une paramétrisation temporelle précise du chemin désiré.

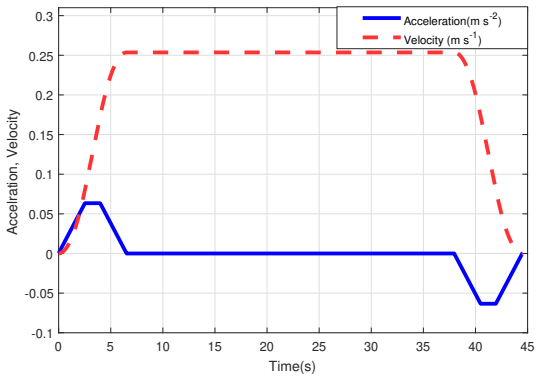


Figure 3.6 – Vitesse et accélération du profil de mouvement.

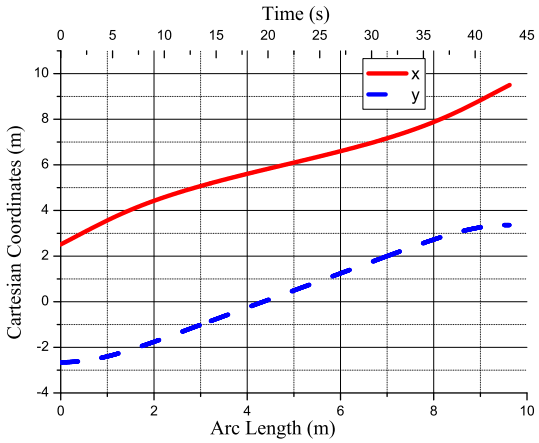
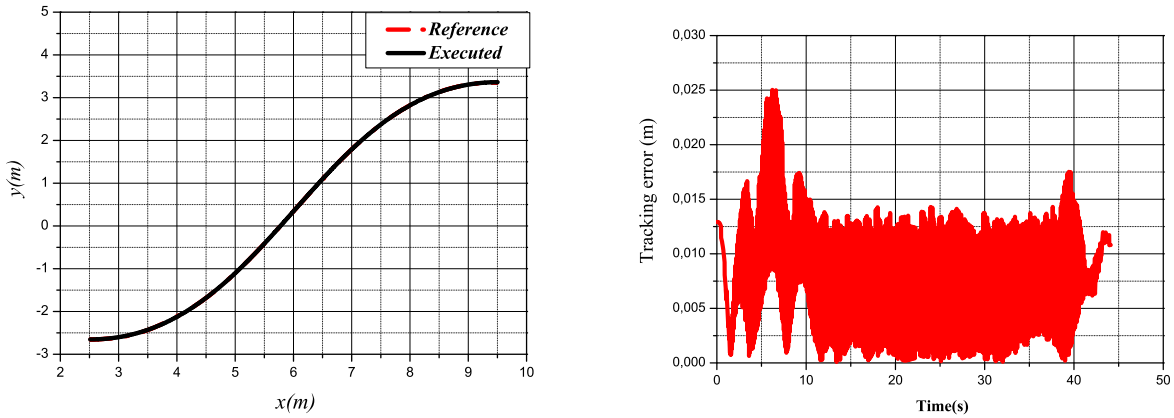


Figure 3.7 – Représentation paramétrique des Trajectoires de référence

La mise en œuvre en temps réel de la tâche ci-dessus a été testée pratiquement sur le prototype du laboratoire PIONEER 3DX. Les mouvements du robot le long de ces

trajectoires désirées ainsi que la norme cartésienne de l'erreur de suivi sont représentées dans Fig. 3.8, où la loi de commande (3.46–3.53) est utilisée pour permettre au robot de suivre les trajectoires désirées avec :

$k_{p1} = 1, k_{d1} = 0.7,$
$k_{p2} = 1, k_{d2} = 0.7.$



(a) (b)

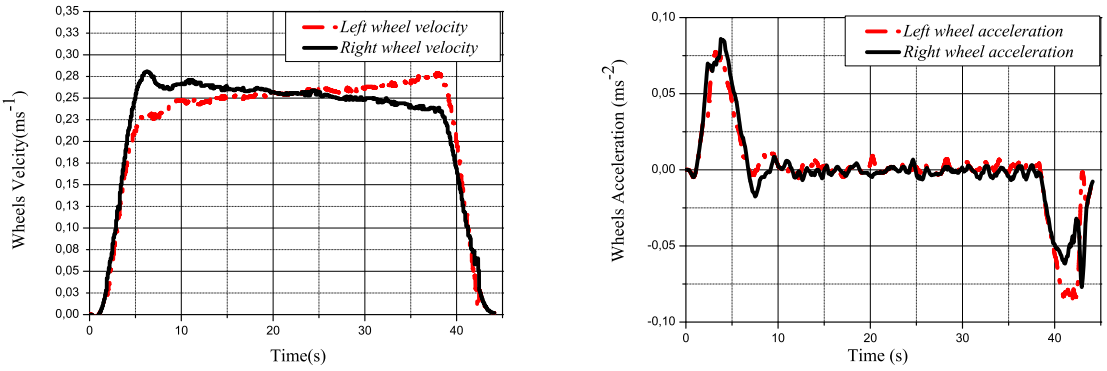
Figure 3.8 – Tâche exécutée (a) Mouvement du robots le long du chemin (b) Norme Cartésienne d'erreurs de suivie

Les vitesses mesurées ainsi que les accélérations estimées des deux roues, illustrées respectivement dans la Fig. 3.9.a et Fig. 3.9.b ; soulignent l'importance de l'algorithme de planification proposé en termes de qualité des signaux de commande. Les signaux de commande générés sont lisses et respectent toutes les limites physiques du robot, c.-à-d, les limites maximales des vitesses et des accélérations de déplacement des deux roues sont inférieures à $v_{max} = 0.3ms^{-1}$ et $a_{max} = 0.1ms^{-2}$, respectivement.
Comme conclusion de ce test, l'algorithme de planification proposé nous permet de définir des tâches réalisables avec un temps d'exécution prédéfini pour le robot, en satisfaisant toutes les exigences de contrôle, y compris la contrainte non-holonome, les limites physiques du système et l'exigence de signaux de contrôle lisse.

3.5.2 Test 2 : Manipulation de robot dans un environnement encombré

Ce test a pour but d'examiner le problème compliqué de la manipulation du robot dans des environnements encombrés. Une image réelle de la carte de l'environnement de test est illustrée dans la Fig. 3.10 (un plan des locaux du laboratoire des systèmes intelligents). L'objectif de contrôle dans ce cas a été fixé pour définir les commandes nécessaires permettant au robot d'atteindre la position cible spécifiée 'G' à partir de sa position de départ 'S'. Pour cet objectif, onze points de passage, marqués à l'aide des points bleus de la Fig. 3.10, sont sélectionnés.

Les conditions d'interpolation listées dans le Tableau.3.1 sont obtenues à partir des équations (3.34–3.42) et utilisées pour la génération du chemin désiré du robot, illustré



(a) (b)

Figure 3.9 – Signaux de commande (a) Vitesses des roue gauche et droite (b) Accélérations des roue gauche et droite

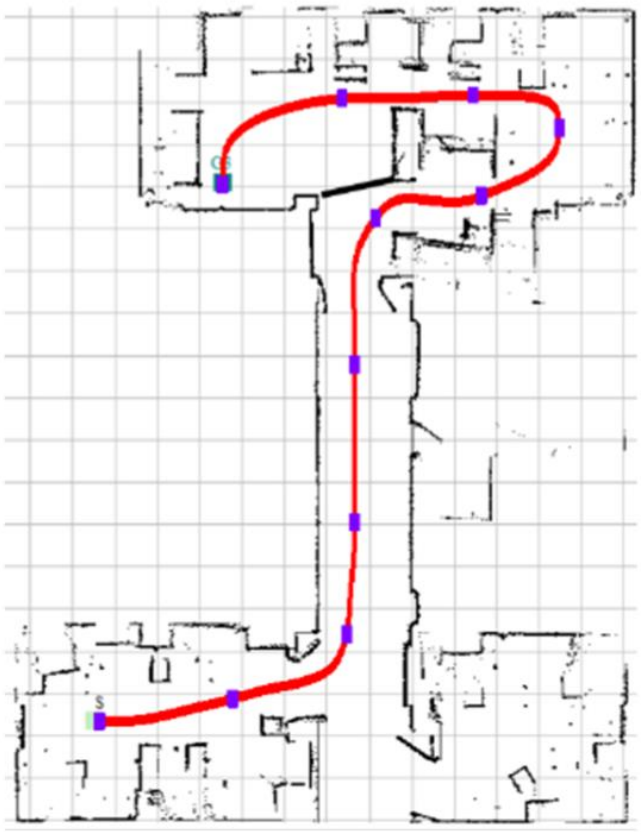
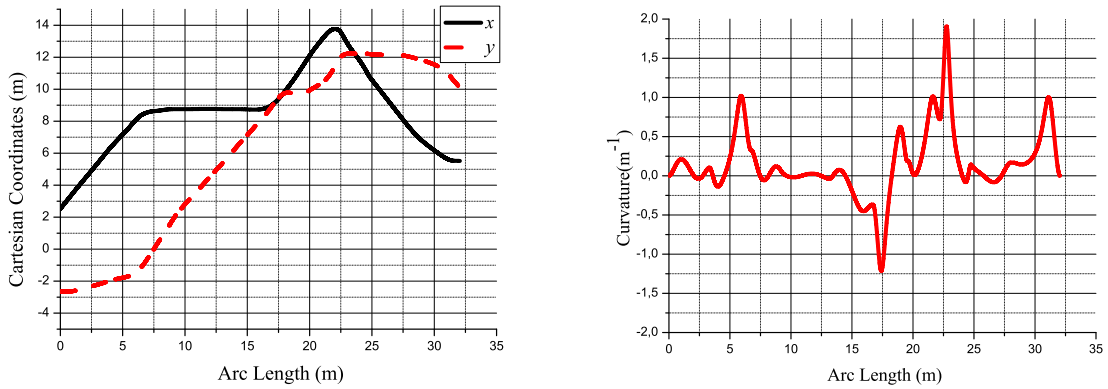


Figure 3.10 – Plan de l'environnement des tests (les locaux de laboratoire des systèmes intelligent), "S : point de départ, G, point d'arrivé et chemin désiré.



(a) (b)

Figure 3.11 – Chemin panifié (a) Coordonnées Cartésiennes (b) Courbure.

Tableau 3.1 – Conditions d'interpolation pour le deuxième exemple

Point P_i	k_i	k'_i	θ_i	v_i
START $[2.525, 2.654]$	0	0	0	0
$[5.779, 2.118]$	0.1047	0.0317	0.2571	0.2160
$[9.112, 0.591]$	0.3263	0.0700	1.3680	0.2160
$[9.020, 2.065]$	0.0235	-0.1137	1.5708	0.2668
$[8.696, 5.809]$	-0.0392	-0.0167	1.5708	0.2560
$[8.753, 9.296]$	-0.3700	-0.0939	0.9051	0.2052
$[11.855, 9.836]$	0.1907	0.2110	0.3477	0.2052
$[13.661, 11.444]$	0.7255	0.2145	1.5708	0.1738
$[11.662, 12.234]$	0.1413	-0.2605	3.1416	0.1738
$[8.441, 12.141]$	0.1689	0.0085	-3.0817	0.2171
GOAL $[5.519, 10.098]$	0	0	-1.5708	0

dans la Fig. 3.10, avec une ligne continue, le long de ces points de passage. La forme paramétrique correspondant de la longueur d'arc des coordonnées cartésiennes du chemin planifié et de sa courbure sont résumées dans la Fig. 3.11, qui illustre le caractère lisse du chemin planifié et la bornitude de sa courbure.
On peut noter que le chemin obtenu est caractérisé par des courbes fortement incurvées dans des passages étroits inévitables.
La dernière colonne du tableau. 3.1 résume les vitesses désirées au différents point de passage requises dans l'algorithme 2 pour la conception du profil de mouvement en vitesse, affecté dans ce chemin prévu.

Tableau 3.2 – Caractéristiques du profil de mouvement pour la deuxième tâche

Seg	Max.Courbure	Longueur	Max.Vitesse	Temps d'exécution (T_i)
1	0.1047	3.3114	0.2777	15.8200
2	0.3263	3.4176	0.2160	15.8100
3	0.3269	2.6658	0.2668	10.4100
4	0.0392	3.7442	0.2956	13.1600
5	0.4511	3.6080	0.2560	14.5300
6	1.2122	2.7976	0.2052	13.6200
7	1.0131	2.6783	0.2165	12.9000
8	1.9068	2.5566	0.1738	14.7000
9	0.1689	3.2235	0.2819	12.9300
10	1.0024	4.0518	0.2171	21.7400

La tâche de mouvement est composée dans ce cas de dix segments récapitulés dans le tableau. 3.2. En appliquant l'algorithme 2, le profil de mouvement de la vitesse qui décrit le mouvement du robot le long du chemin planifié est montré dans la Fig.3.12 , avec une ligne continue et un temps d'exécution total de $T = \sum T_i = 145.61s$. L'algorithme proposé permet de définir un profil de vitesse dépendant de la courbure. Il fournit des vitesses élevées pour le robot, dans les parties à faible courbure du chemin et vice versa.
Pour une étude comparative, la trajectoire du robot le long de ce chemin peut également être définie comme une manœuvre repos-à-repos utilisant l'approche classique, avec un chemin parcourue d'une longueur maximale $L = \sum L_i = 32.0547m$ et une courbure maximale $k_{max} = max(k_{imax}) = 1.9068m^{-1}$. Comme le montre la Fig. 3.12 avec une ligne pointillée, le maximum du profil de mouvement attribué à la tâche dans ce cas est obtenu comme suit $\psi_{max} = 0.1738ms^{-1}$.

En conséquence, le temps d'exécution de la tâche proposée est augmenté à $T = 190.87s$ par rapport à $T = 145.61s$ pour la réalisation de la tâche selon l'algorithme de planification proposé. Fig. 3.13 décrit le mouvement en temps réel du robot le long de ce chemin et la norme d'erreur de suivi cartésienne, en utilisant les mêmes paramètres du contrôleur comme dans le premier test et en considérant le profil de mouvement à vitesse optimisée.

Les vitesses de commande dans les deux roues le long du chemin exécuté sont représentées dans la Fig. 3.14. Des signaux de commande lisses sont fournis pour assurer le mouvement du robot le long du chemin défini avec un niveau de précision acceptable. L'algorithme de planification proposé amène le robot dans ses capacités maximales, vitesses maximales des roues gauche et droite, pour atteindre un temps d'exécution optimisé de la tâche, sans dépasser les limites physiques du robot.
Les résultats obtenus lors du deuxième test confirment les capacités de l'approche de planification proposée lorsque le robot est manipulé dans des environnements encombrés. Outre que la définition des mouvements sans collisions, l'algorithme de génération de la vitesse dépendante de la courbure, l'algorithme 2, optimise considérablement le temps nécessaire à l'exécution de la tâche complète.

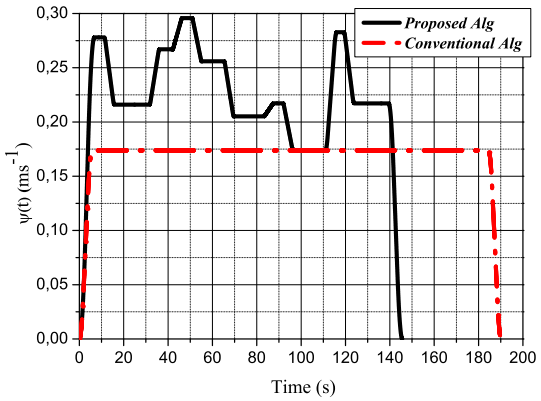
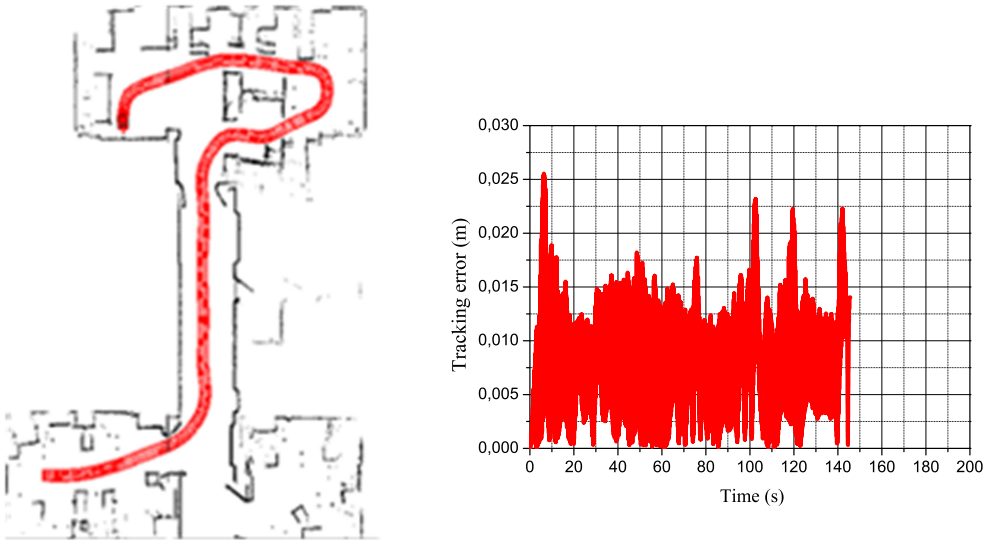
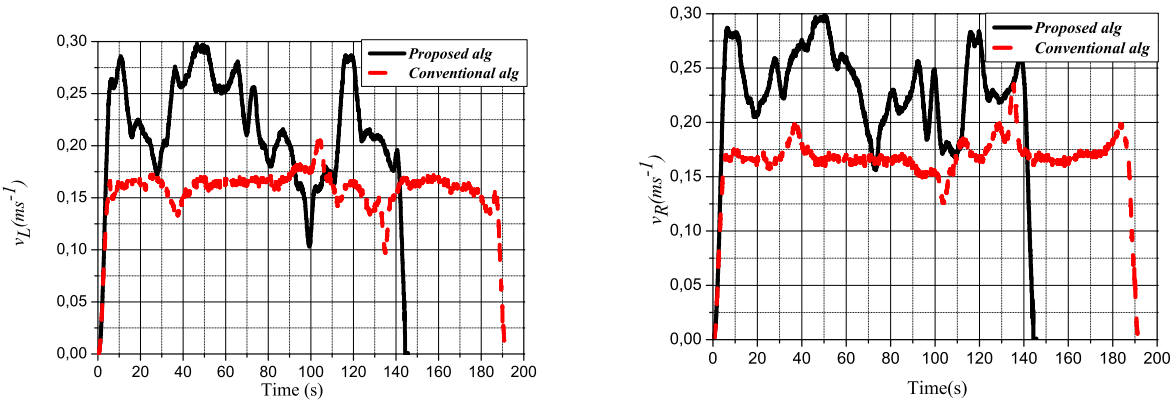


Figure 3.12 – Profil de mouvement selon les deux algorithmes.



(a) (b)

Figure 3.13 – Tâche exécutée (a) Mouvement du robot (b) Norme Cartésienne de l'erreur



(a) (b)

Figure 3.14 – Signaux de commande pour le test 2 (a) vitesse de la roue gauche (b) Vitesse de roue droite

3.5.3 Test 3 : Évaluation de la commande de rétroaction

Le but de ce test est d'évaluer les performances de l'algorithme de contrôle proposé en termes de précision de suivi et de robustesse face aux incertitudes des modèles et aux perturbations externes. Pour cette raison, deux contrôleurs seront considérés pour une étude comparative :

1) Le contrôleur proposé dans les équations (3.46–3.53) et avec les paramètres de contrôle fixés comme suit :

$k_{p1} = 1, k_{d1} = 0.7$

$k_{p2} = 1, k_{d2} = 0.7.$

2) Un algorithme de contrôle non linéaire est défini comme suit [Oriolo2002] :

$$\begin{cases} v = v_d cos(\theta_d - \theta) + \\ K_1((x - x_d)cos(\theta) + (y - y_d)sin(\theta)) \\ \omega = \omega_d K_2 \dfrac{sin(\theta_d - \theta)}{(\theta_d - \theta)}((y - y_d)cos(\theta) + (x - x_d)sin(\theta)) \\ + K_3(\theta_d - \theta) \end{cases} \tag{3.54}$$

Un choix approprié des paramètres de contrôle K_i sont donné comme suit :

$K_1 = K_3 = 2\zeta\sqrt{w_d^2 + bv_d^2}$

$K_2 = b \mid v_d(t) \mid$

où les paramètres ζ et b sont fixés dans ce travail comme suit :

$\zeta = 0.9$, $b = 2.$

Les vitesses de translation et rotations souhaitées $v_d(t)$ et $\omega_d(t)$ son obtenues à partir des équations suivantes :

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

$\omega_d(t) = \dfrac{\ddot{y}(t)\dot{x}(t) - \ddot{x}(t)\dot{y}(t)}{(\dot{x}^2(t) + \dot{y}^2(t))}$

Considérons la même tâche spécifiée dans le premier test (Test1). Ensuite, trois expériences ont été étudiées dans ce test :

(a) La première correspond au cas nominal.

(b) La seconde considère que l'entrée de commande du robot v est soumise à des signaux de perturbations externes additives inconnues δv mais bornées, avec des limites maximales $\delta v < 0.04 ms^{-1}$.

c) La troisième expérience traite le problème de robustesse dans le cas de présence des incertitudes dans la mesure des positions réelles du robot x_c et y_c.

La norme cartésienne des erreurs de suivi dans toutes les investigations sont illustrées dans les Figs.3.15, 3.16 et 3.17, respectivement.

Après avoir analysé les Figs. 3.15, 3.16 et 3.17, on peut facilement noter les principales caractéristiques du contrôleur par rétroaction, basé sur la platitude, proposée que l'on peut résumer par le haut niveau de précision de suivi, la robustesse du système et l'insensibilité de la réponse du robot à la dynamique non modélisée et aux perturbations externes.

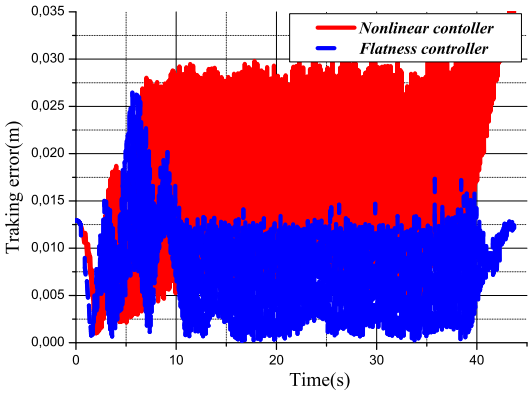


Figure 3.15 – Normes Cartésiennes de l'erreur sous les deux contrôleurs dans le cas nominal.

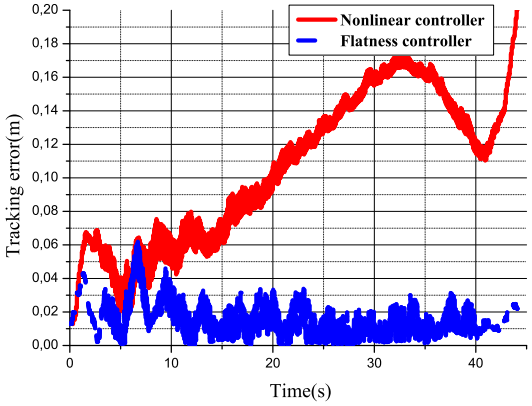


Figure 3.16 – Normes Cartésiennes de l'erreur sous les deux contrôleurs sous des perturbations exogènes.

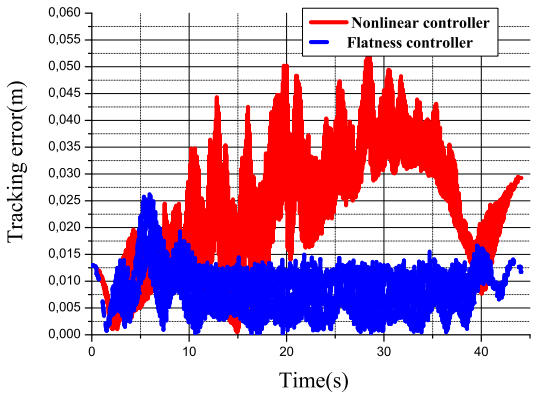


Figure 3.17 – Normes Cartésiennes de l'erreur sous les deux contrôleurs sous des incertitudes de mesure.

3.6 Conclusion

Dans ce chapitre, nous avons développé un algorithme efficace pour le contrôle des mouvements rapides et sûrs pour un robot mobile à roues avec des contraintes non holonome. Une analyse primaire du problème de planification de mouvement sans obstacles (PTP) pour les systèmes robotiques avec des contraintes non-holonome a montrée la relation inverse entre la courbure du chemin planifié et les limites maximales du profil de mouvement décrivant la vitesse du robot le long de ce chemin. Une extension de l'approche de planification découplée a été proposée dans ce chapitre pour une manipulation efficace des robots mobiles non-holonome dans des environnements encombrés d'obstacles statiques. Tout d'abord, un chemin multi-segments lisse et sans collision (free collision) a été défini pour le robot en interpolant un nombre suffisant de points de passage sélectionnés à l'aide de fonctions polynomiales d'ordre sept. Ensuite, et afin de minimiser le temps de cycle de la tâche exécutée, le profil de mouvement qui spécifie la vitesse du robot le long du chemin désiré a été défini comme une chaîne de fonctions élémentaires dépendantes de la courbure qui assurent un mouvement continu du robot jusqu'à où il atteigne sa configuration cible. Tenant en compte les caractéristiques locales du chemin planifié et en appliquant un filtre de moyenne mobile aux profils d'accélération limités classiques avec des vitesses non nulles aux points de passage, chaque fonction de vitesse élémentaire a été conçue pour assurer une solution presque optimale dans le temps pour la tâche de planification de trajectoire, ainsi qu'une continuité suffisante des profils de vitesse et d'accélération globales. Un algorithme de contrôle par rétroaction basé sur la platitude a été proposé pour assurer la robustesse et la précision durant l'exécution des tâches planifiées malgré les incertitudes du modèle et les perturbations externes. L'efficacité de l'approche proposée a été testée et validée pratiquement à l'aide du prototype de laboratoire *PIONEER3DX*.

4

Commande d'un Portique à trois dimensions

Dans ce chapitre, nous considérons le problème de commande d'un portique de levage à 5 dll. Après la génération des trajectoires de références, une commande par mode glissant d'ordre deux est introduite pour assurer l'exécution des taches planifiées.

Sommaire

4.1 Introduction 76
4.2 Modélisation du système et formulation du problème 77
4.2.1 Modélisation du système, caractéristique et contraintes de commande 77
4.2.2 Linéarisation par bouclage 78
4.3 Planification du mouvement 79
4.3.1 Planification du chemin 79
4.3.2 Définition du profil de mouvement 80
4.4 Suivi des Trajectoires 81
4.4.1 Conception du contrôleur 81
4.4.2 Analyse de stabilité 83
4.5 Validation numérique des résultats 85
4.5.1 planification de trajectoires 85
4.5.2 Analyse de contrôle de suivi 85
4.5.3 Analyse d'anti-ballant 87
4.6 Conclusion 90

4.1 Introduction

Ce chapitre a pour but d'étendre les approches discutées dans le chapitre 3 pour manipuler des systèmes mécaniques avec une structure flexible, les ponts roulants ou portiques. Dans les sites industriels et de construction (installations de traitement des déchets nucléaires, chantiers navals, aéroports, usines automobiles, nouveaux bâtiments, etc.), les grues (portiques) sont utilisées pour transporter de grandes et volumineuses charges utiles via un câble relié à une plate-forme motorisée [Lee1998, Sawodny2002, M.Abdel-Rahman2003, Omar2004, Ramli2017]. La structure flexible des ponts roulants leur permet d'accomplir différentes tâches dans des grands espaces de travail. Cependant, cette flexibilité peut être vue comme une source de problèmes lors du mouvement du pont, principalement, l'excitation paramétrique des oscillations de la charge. Le comportement oscillatoire de la charge est provoqué par différents facteurs tels que (i) la sensibilité de la structure mécanique aux accélérations et décélérations de la plate-forme mobile, (ii) les dynamiques non modélisées, comme les forces de frottement et les perturbations externes, comme le vent et (iii) les paramètres variables du système, comme la longueur du câble [Cartmell2003, Peña2012]. Ainsi, le problème immédiat dans toute conception de commande est le développement de techniques de commande avancées qui tiennent en compte des différentes sources de comportement des oscillations de charge pour atteindre un niveau élevé de précision et de sécurité malgré la nature incertaine et sous-actionnée du système.

Un premier niveau du développement de la commande pour les systèmes de type pont roulant consiste à générer des trajectoires de référence appropriées qui assurent le transfert sûr de la charge dans son environnement aussi rapidement que possible et sans collision avec les obstacles, tout en tenant compte des oscillations de charge pendant les phases d'accélération et de décélération du mouvement. En conséquence, une approche de planification découplée est proposée dans ce chapitre pour assurer une planification efficace de tâches désirées. Elle s'agit de concevoir un chemin lisse et libre d'obstacles le long duquel la charge doit se déplacer, puis d'attribuer la vitesse de la charge en utilisant un profil de mouvement approprié. Ce dernier doit être soigneusement conçu pour éviter l'excitation des oscillations de charge à hautes fréquences. Le deuxième niveau de la conception du contrôle nécessite le développement d'une loi de commande par rétroaction appropriées pour assurer une exécution précise des tâches planifiées.

D'un point de vue contrôle, plusieurs approches, y compris des techniques d'intelligence artificielles [Lewis2002] et des algorithmes de contrôle par mode glissant (SMC) [Utkin2009b], ont été développées pour traiter le problème de contrôle robuste des systèmes sous-actionnés et incertains. Particulièrement, les algorithmes conventionnels de la théorie de SMC sont appliqués pour contrôler le comportement des ponts roulants [Liu2012, Collado2013, Bartolini2003]. Les techniques de SMC conventionnelles se caractérisent par leur simplicité de conception et leur robustesse face à une classe des perturbations [Utkin1993a]. Cependant, cette classe de contrôleurs (SMC) n'est appliquée qu'aux systèmes ayant un degré relatif un, ce qui limite les degrés de liberté de la conception de la commande et dégrade les performances des systèmes. Les développements récents de la théorie SMC prouvent que le SM d'ordre supérieur est plus suffisant que les algorithmes conventionnels en termes de stabilité, de robustesse et de précision du système (Chapitre 2). Motivé par le succès des applications des modes glissants d'ordre deux (SOSMC) dans de nombreux domaines, ce chapitre traite la conception d'un SOSMC pour obtenir des performances de suivi et de positionnement élevées pour un pont rou-

lant. En utilisant le fait que la dynamique du pont roulant est partiellement linéarisable par rétroaction, une surface de glissement a été définie de telle sorte que sa régulation assure la limitation de l'erreur de suivi ainsi que l'atténuation de toute oscillation de charge causée par un comportement externe. Un algorithme Twisting dépend de la variable d'état (State Dependent Twisting SDT) est ensuite introduit dans une deuxième étape comme un compensateur discontinu pour améliorer les performances du système en termes de précision du mouvement, robustesse contre les incertitudes structurées et non structurées et atténuation des oscillations. Les gains du contrôleur sont définis comme fonction dépendant de l'erreur afin de réduire les phénomènes de chattering et d'adoucir les signaux de commande. En plus de sa simplicité de conception et de ses caractéristiques de robustesse, cet algorithme permet une convergence en temps fini de la variable d'état à zéro. De plus, en cas de discrétisation, ce contrôleur garantit une précision quadratique de régulation par rapport à la période d'échantillonnage. Une preuve formelle de la stabilité asymptotique de l'approche de contrôle proposée est établie en utilisant la théorie de Lyapunov.

Les grandes parties de ce chapitre sont les suivantes. Le modèle dynamique du pont roulant et quelques propriétés standard pour cette classe de système sont présentés à la section 2. L'algorithme de planification de trajectoire proposé est donné dans la section 3. La section 4 donne les détails de la loi de commande proposée. Les résultats de la simulation, illustrent les avantages et l'efficacité de la stratégie proposée, sont présentés à la section 5. La section 6 conclut le chapitre.

4.2 Modélisation du système et formulation du problème

4.2.1 Modélisation du système, caractéristique et contraintes de commande

On considère le système de pont roulant à $5ddl$ illustré dans Fig.4.1, piloté par un vecteur de force définie comme suit $\tau = [f_x \quad f_y \quad f_l\ ]^T$. En se basant sur le formalisme de Lagrange, le modèle dynamique pour ce système peut se mettre sous la forme :

$$M(q)\ \ \ddot{q} + C(q,\ \dot{q}\ )\ \ \dot{q} + G(q) = B\ \tau \tag{4.1}$$

avec $q(t) = [q_u\ q_a]^T$ et $\dot{q}(t) = [\dot{q}_u\ \dot{q}_a]^T$ sont les vecteurs coordonnées généralisés et des vitesses respectivement, $q_a \in \mathbb{R}^n (n = 3)$ représente la partie actionnée du système composée de coordonnées cartésiennes x, y de la charge utile mesurées dans la référence inertielle fixe $OXYZ$ et la longueur l du câble de levage, $q_u \in \mathbb{R}^m (m = 2)$ représente la partie non actionnée du système composée de deux angles (θ_x et θ_y) représentent la projection de l'angle d'oscillation θ sur les plans $X_T Y_T$ et $X_T Z_T$ ($X_T Y_T Z_T$ est le système de coordonnées qui se déplace avec le chariot " figure1 "). $M(q)$ est la matrice d'inertie, $C(q,\dot{q})$ est la matrice centripète et de Coriolis, $G(q)$ est le vecteur de gravité. B est une matrice constante donnée comme : $B = [b_1\ b_2]^T$ d'où $b_1 = 0_{[m\times n]}$ et $b_2 = I_{[n\times n]}$.
Le modèle dynamique (4.1) conserve certaines propriétés standard de la dynamique lagrangienne mécanique, y compris la propriété définie positive symétrique de la matrice d'inertie M, l'anti-symétrique de la matrice $\dot{M} - C(q,\dot{q})$ et la bornitude de la matrice $G(q)$.

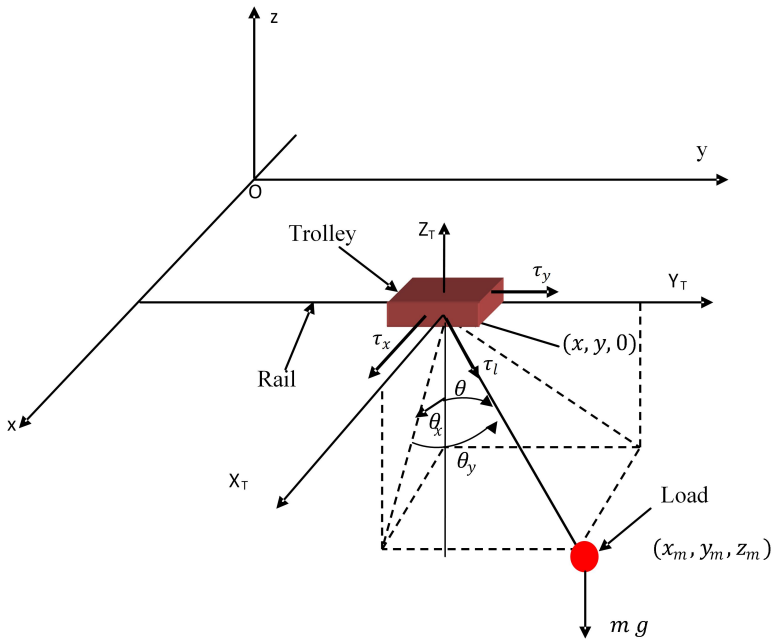


Figure 4.1 – Portique de trois dimensions

4.2.2 Linéarisation par bouclage

Linéarisation par bouclage statique

Pour les systèmes sous-actionnés, tels qu'un pont roulant, nous pouvons réorganiser les coordonnées généralisées décrites par (4.1) sous la forme suivante :

$$\begin{cases} m_{11}\ddot{q}_u + m_{12}\ddot{q}_a + f_1(q,\dot{q}) = 0 \\ m_{21}\ddot{q}_u + m_{22}\ddot{q}_a + f_2(q,\dot{q}) + \delta d = \tau \end{cases} \tag{4.2}$$

avec $\delta d \in \mathbb{R}^n$ représente un vecteur borné inconnu introduit ici pour décrire toutes les incertitudes du système, y compris les perturbations externes et les dynamiques non modélisées.
La matrice d'inertie M est définie positive, alors, le terme m_{11} est 2×2 est inversible. Donc, nous pouvons résoudre $\ddot{q}_u$ à partir de la première équation dans (4.2) comme :

$$\ddot{q}_u = -m_{11}^{-1}(m_{12}\ddot{q}_a + f_1(q,\dot{q})) \tag{4.3}$$

puis nous substituons l'expression résultante (4.3) dans la deuxième équation de (4.2) pour obtenir :

$$\bar{m}_{22}\ddot{q}_a + \bar{f}_2 = \tau \tag{4.4}$$

où : $\bar{m}_{22} = m_{22} - m_{21}m_{12}m_{11}^{-1}$ et $\bar{f}_2 = f_2 - m_{21}m_{11}^{-1}f_1$.
Une commande linéarisante peut donc être définie pour l'équation (4.4) selon :

$$\tau = \bar{m}_{22}v + \bar{f}_2 \tag{4.5}$$

d'où $v \in \mathbb{R}^n$ est un contrôleur auxiliaire a déterminé. Ensuite, le système complet peut être écrit comme suit :

$$\begin{cases} m_{11}\ddot{q}_u + f_1 = -m_{12}v \\ \ddot{q}_a = v + \delta d \end{cases} \tag{4.6}$$

Puisque la relation entrée-sortie de q_a à v dans (4.6) est linéaire, la partie active de (4.2) a été complètement linéarisée. Cependant, si l'on considère le vecteur d'état complet, seule une linéarisation partielle a été réalisée par un bouclage statique.

Linéarisation par bouclage dynamique

En plus de la propriété de la linéarisation partielle par bouclage statique (4.3–4.6), il a été montré dans [D'Andréa-Novel1990, Fliess1995, Van Nieuwstadt1998] que cette classe de système est complètement linéarisable par un bouclage dynamique d'ordre deux. Le vecteur de coordonnées Cartésiennes $[x \;\; y \;\; z]^T$ est défini dans ce cas comme un vecteur de sorties plates.

4.3 Planification du mouvement

En utilisant le modèle dynamique (4.1, 4.2 et 4.6), le problème de la planification de trajectoire consiste à trouver un vecteur 3D $r_d(t) = [x_d(t) \;\; y_d(t) \;\; z_d(t)]^T$ des positions désirées qui amènent la charge à sa position final le plus rapidement possible et sans collision avec les obstacles. La conception doit tenir compte la dynamique du système décrit dans l'équation (4.2), c'est-à-dire qu'elle doit assurer la bornitude du vecteur q_u qui décrit le comportement vibratoire de la charge. Pour cela on doit répondre aux problèmes *SPPProblem* et *STPProlem* définis dans le chapitre1, c.à.d. de spécifier le type et le degré des fonctions utilisées pour la conception du chemin de la charge et du profil du mouvement. Un résultat direct de la propriété de platitude du vecteur des coordonnées Cartésiennes pour cette classe de système implique que :

— (i) l'ordre minimum de la fonction qui spécifie le chemin de la charge, dans l'espace plat, est égal à cinq.

— (ii) l'ordre minimum de la fonction qui spécifie la vitesse de la charge le long de ce chemin est égal à cinq.

4.3.1 Planification du chemin

Comme point de départ, on considère une séquence de N configurations désirées $\{P_i / P_i = [x_i \;\; y_i \;\; z_i], \;\; i = 0, 1, \cdots, \}$ que la charge doit atteindre, où P_0 et P_N coïncident avec les positions initiale et cible de la charge, respectivement et toutes les autres configurations $\{P_i, i = 1, ..., N-1\}$ sont sélectionnées depuis l'espace libre de la charge pour éviter toute collision de la charge dans son environnement. Le processus de planification du chemin est focalisé sur la transformation de la séquence des configurations prédéfinies en une courbe lisse. Une approche standard pour trouver une telle courbe peut être exprimée sous la forme d'une procédure en deux étapes (Algorithme 3 : 3D-Curve P arameterization) [Boutalbi2015].

Algorithm 3 : 3D-Curve P arameterization

Input :Samples of N way-points $P_i = [x_i\ y_i\ z_i]$.
Output : $3D$ Parameterized curve $r_d(u)$.

1: Curve parameterization :
   — Set : $u_0 = 0$.
   — For $(i = 1, i < N, i++)$ set $u_{i+1} = u_i + |p_i - p_{i+1}|^{\alpha}$, $\alpha = 0.5$
2: Between any two successive points, find the interpolation function such that :

$$r_d(u_i) = p_i, r_d^{(k)}(u_{i-1}) = r_d^{(k)}(u_i)$$

3: return $r_d(u)$

La première étape de **l'algorithme 3** se réfère au choix des valeurs des paramètres d'interpolation u. Il est bien connu qu'elles ont une grande influence sur la forme de la courbe obtenue [Floater2008]. Une approche simple, fréquemment utilisée dans la pratique, consiste à considérer $u_0 = 0$ puis à calculer récursivement :

$$u_{i+1} = u_i + |p_i - p_{i+1}|^{\alpha}, i = 0, 1, \ldots, N-1 \tag{4.7}$$

où $|.|$ dénote la norme euclidienne dans $\mathbb{R}^3$.
Le cas particulier de $\alpha = 1$ est connu sous le nom de paramétrisation en fonction de longueur corde *"chordal parametrisation"*. Elle donne une bonne approximation de l'arc entre deux points distincts P_i et P_{i+1}. Les paramètres d'interpolation u_i sont définie dans ce cas comme :

$$\alpha = 1, \text{ et } u_i = |p_{i-1} - p_i| = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \tag{4.8}$$

La deuxième étape **d'Algorithme 3** consiste à trouver une fonction appropriée (d'ordre 5) qui assure une courbe lisse à travers la séquence des valeurs de paramètres prédéfinies u_i. La courbe le long de laquelle la charge doit se déplacer est spécifiée en N segments par les équations suivantes :

$$r_d(u) = \begin{cases} r_{d1}(u) & u_0 \le u \le u_1 \\ r_{d2}(u) & u_1 \le u \le u_1 \\ \vdots & \vdots \\ r_{dN}(u) & u_{N-1} \le u \le u_N \end{cases} \tag{4.9}$$

Ainsi, $r_d(u)$ est un polynôme d'ordre 5 paramétré sur chaque intervalle $[u_i\ u_{i+1}]$, $i = 0, 1, \cdots$. où $r_d^{(k)}(u_{i-1}) = r_d^{(k)}(u_i), k = 1, \cdots, 5$ pour $i = 1, 2, \cdots$, (conditions de continuité dans les points de passage).

4.3.2 Définition du profil de mouvement

En appliquant la même méthode discutée dans les chapitres 1 et 3, le profil de mouvement est obtenu directement à partir d'un profil de mouvement Bang-Zéro-Bang en accélération. En utilisant deux filtres de mouvement moyen (FIR), comme le montre la figure

4.2. Un mouvement lisse (avec un snap limité) est obtenu. Le temps total d'exécution de la tache est caractérisé dans ce cas par le temps optimal t_f du mouvement Bang-Zéro-Bang conventionnel, le temps T_j du premier filtre et le temps T_s du deuxième filtre.

$$t'_f = t_f + T_j + T_s \tag{4.10}$$

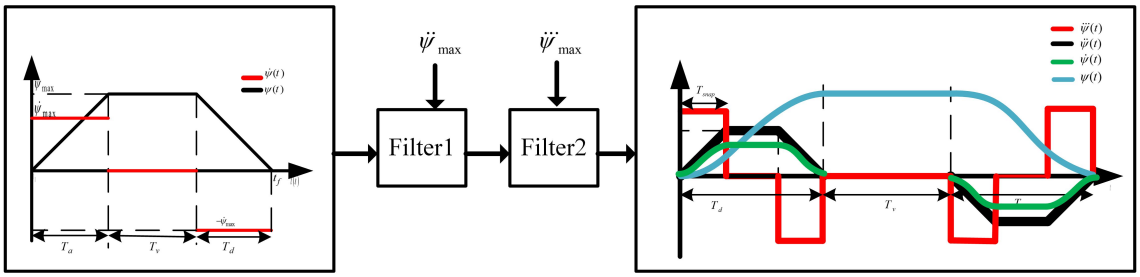


Figure 4.2 – Profil de vitesse à snap limité

4.4 Suivi des Trajectoires

4.4.1 Conception du contrôleur

L'objectif principal de cette partie est le développement d'une loi de contrôle appropriée qui assure l'exécution précise des tâches planifiées et l'élimination des oscillations de charge peuvent être générées par des perturbations externes.

Étant donné un vecteur de trajectoire désirées $q_{ad}(t) = r_d(t) = [x_d(t)\ y_d(t)\ l_d(t)]^T$, on définit l'erreur de suivi comme :

$$\begin{cases} \eta_1 = q_u \\ \eta_2 = \dot{q}_u \\ e_1 = q_a - q_{ad}(t) \\ e_2 = \dot{q}_a - q_{ad}(t) \end{cases} \tag{4.11}$$

Prenons la première dérivée de (4.11) pour obtenir :

$$\begin{cases} \dot{e}_1 = e_2 \\ \dot{e}_2 = -\ddot{q}_{ad}(t) + v + d_0 \\ \dot{\eta}_1 = \eta_2 \\ \dot{\eta}_2 = \ddot{q}_u \end{cases} \tag{4.12}$$

où : $d_0 = \delta d$.

La question principale de la conception de la commande par rétroaction est de savoir comment définir les entrées de commande qui assurent la stabilisation des erreurs de suivi $e_1(t)$ et $e_2(t)$ ainsi que les oscillations de charge η_1 , η_2 à zéro.

Remarque 10 *Comme le système est linéarisable par bouclage dynamique, une rétroaction endogène similaire à la commande (3.52) peut être définie pour assurer le suivi des trajectoires. Cependant, la mise en ouvre pratique de cette loi de commande exige dans ce cas la quatrième dérivée de vecteur d'erreur. Dans ce travail, on exploite l'efficacité de commande par mode glissant pour assurer la stabilité, la robustesse et la précision d'exécution de la tache planifiée en utilisant seulement la deuxième dérivée du vecteur d'erreur de suivi.*

Le processus de stabilisation du système (4.11) par mode glissant nécessite la définition des sorties qui prennent en compte le vecteur d'erreur de poursuite e ainsi que le vecteur décrivant les oscillations de la charge η. En conséquence, une forme appropriée de la surface de glissement μ peut être choisie comme [Boutalbi2017] :

$$\mu = e_1 + k\left[\eta_1 + h\int_0^t \eta_2 dx\right] \tag{4.13}$$

avec $\mu \in \mathbb{R}^n$, h est un paramètre de réglage positif. La matrice $k \in \mathbb{R}^{(n\times m)}$ est définie comme :

$$k = \begin{bmatrix} k_\theta & 0 \\ 0 & k_\beta \\ 0 & 0 \end{bmatrix}$$

avec k_θ et k_β sont des facteurs de couplage positifs qui jouent un rôle important dans le contrôleur dynamique.
La commande v a un degré relatif égal à deux par rapport à la sortie μ dans (4.13). Prenons la deuxième dérivée de μ, on obtient :

$$\ddot{\mu} = -q_{ad}(t) + \left(I + km_{11}^{-1}m_{12}\right)v + km_{11}^{-1}\left(f_1 + m_{12}\eta_2\right) + d_0 \tag{4.14}$$

avec $I_{(n\times n)}$ est la matrice unitaire.
Puisque la matrice $\left(I + km_{11}^{-1}m_{12}\right)$ est inversible, une nouvelle commande v est introduite sous la forme :

$$v = \ddot{q}_{ad}(t) + \left(I + km_{11}^{-1}m_{12}\right)^{-1}\left[U + km_{11}^{-1}\left(f_1 + m_{12}\eta_2\right)\right] \tag{4.15}$$

L'algorithme de commande dans (4.15) est composé d'une commande en boucle ouverte $\ddot{q}_{ad}(t)$ et d'une partie de linéarisation par rétroaction. Cette structure de commande illustre clairement l'importance de planification des signaux de commande lisses. Elle réduit le système (4.14) à :

$$\ddot{\mu} = U + d_0 \tag{4.16}$$

Nous pouvons réécrire le système de (4.16) comme un système de trois double-intégrateurs découplés non-autonomes perturbés, comme :

$$\begin{cases} \ddot{\mu}_x = u_x + d_{0x} \\ \ddot{\mu}_y = u_y + d_{0y} \\ \ddot{\mu}_l = u_l + d_{0l} \end{cases} \tag{4.17}$$

où u_x, u_y et u_l sont des nouvelles entrées de commande devraient être conçues pour stabiliser la dynamique $\ddot{\mu}_x, \ddot{\mu}_y$ et $\ddot{\mu}_l$ respectivement, malgré les incertitudes du modèle et

les perturbations externes. La sortie μ_α a un degré relatif égal à deux par rapport à l'entrée de commande u_α (pour $\alpha = (x,\ y,\ l)$) . Par conséquent, une commande par mode glissant d'ordre deux pour l'entrée u_α peut être définie comme suit [Emeryanov1996, Levant2007] :

$$\begin{aligned} u_\alpha &= -M_{1\alpha} sign(\sigma_1) - M_{1\alpha} sign(\sigma_2) \\ \sigma_1 &= \mu_\alpha,\ \ \sigma_2 = \dot{\sigma}_1 = \dot{\mu}_\alpha \end{aligned} \tag{4.18}$$

Cette loi de commande garantit que le système est uniformément globalement asymptotiquement stable si les gains $M_{1\alpha}$ et $M_{1\alpha}$ sont correctement choisis. Dans le présent chapitre, on a choisi de les définir comme des fonctions dépendantes des états (σ_1 et σ_2) comme suit :

$$\begin{aligned} M_{1\alpha} &= k_{p\alpha}|\sigma_1| + r_{1\alpha} \\ M_{2\alpha} &= k_{d\alpha}|\sigma_2| + r_{2\alpha} \end{aligned} \tag{4.19}$$

d'où $k_{p\alpha},\ \ k_{d\alpha}$ sont des paramètres positifs et $r_{1\alpha}\ \ , r_{2\alpha}$ sont des constantes positives suffisamment petites.
Ce dernier choix pour les gains de réglage par mode glissant (4.19) est adopté dans cette forme pour minimiser le phénomène de chattering face l'implémentation des techniques de mode glissant.
La loi de contrôle (4.18) est reformulée comme suit :

$$u_\alpha = -k_{\alpha p}\sigma_1 - k_{d\alpha}\sigma_2 - r_{1\alpha} sign(\sigma_1) - r_{2\alpha} sign(\sigma_2) \tag{4.20}$$

qui est équivalent à l'algorithme PD-T (Proportionnel-Dérivé Twisting) proposé par de nombreux auteurs [Orlov2005, Vázquez2013].

4.4.2 Analyse de stabilité

Pour prouver la stabilité du système (4.17) sous l'entrée de commande (4.18-4.19), on considère la fonction Candidate de Lyapunov définie comme :

$$V = \frac{1}{2}\sigma^T P\sigma + r_1 k_p \varepsilon |\sigma_1| \tag{4.21}$$

d'où :

— $P = \begin{bmatrix} \varepsilon k_p + k_d & 1 \\ 1 & \varepsilon \end{bmatrix}$,

— ε est une petite constante positive..

— $\sigma = \begin{bmatrix} \sigma_1 & \sigma_2 \end{bmatrix}^T$

Notez également que la dynamique (4.17) sous la loi de commande (4.18-4.19) peut être réécrite sous forme d'espace d'états de la forme :

$$\dot{\sigma} = B_1\sigma + B_2 sign(\sigma) + C \tag{4.22}$$

avec :

$$B_1 = \begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix}, B_2 = \begin{bmatrix} 0 & 0 \\ -r_1 & -r_2 \end{bmatrix}, C = \begin{bmatrix} 0 & c \end{bmatrix}^T$$

$$\sigma = \begin{bmatrix} \sigma_1 & \sigma_2 \end{bmatrix}^T, sign(\sigma) = \begin{bmatrix} sign(\sigma_1) & sign(\sigma_2) \end{bmatrix}^T$$

Prendre la dérivée temporelle de l'équation (4.21) le long de la solution du système (4.22) pour obtenir :

$$\dot{V} = -\sigma^T Q \sigma - r_1|\sigma_1| - r_2\varepsilon|\sigma_2| - r_2\sigma_1 sign(\sigma_2) + c\sigma_1 + \varepsilon\sigma_2 c \tag{4.23}$$

avec :

$$Q = \begin{bmatrix} k_p & 0 \\ 0 & \varepsilon k_d - 1 \end{bmatrix}$$

Ensuite, les conditions suffisantes pour accepter V comme une fonction de Lyapunov, c'est-à-dire que V est une fonction définie positive et que sa dérivée temporelle est une fonction semi-définie négative, peuvent être choisies comme suit :

$$\begin{cases} r_2 \geq c, \\ r_1 \geq r_2 + c, \\ k_p \geq \frac{1}{\varepsilon^2} \\ k_d \geq \frac{1}{\varepsilon} \end{cases} \tag{4.24}$$

En outre, le système de comparaison suivant peut être écrit comme suit :

$$\begin{cases} V_1 \leq V \leq V_2, \\ \dot{V} \leq -V_3 \end{cases} \tag{4.25}$$

avec :

$$\begin{cases} V_1 = \lambda_{min}[P]||\sigma||^2 \\ V_2 = \lambda_{max}[P]||\sigma||^2 + r_1\varepsilon||\sigma|| \\ V_3 = \lambda_{min}[Q]||\sigma||^2 + r_m||\sigma|| \\ r_m = min(r_1 - r_2 - c, \varepsilon(r_1 - c)) \end{cases} \tag{4.26}$$

$\lambda_{min/max}[.]$:représente la plus petite valeur propre de la matrice $[.]$ qui est un nombre positif. $|()|$ représente la première norme du vecteur $()$.

En utilisant les théorèmes 2 et 3, les inégalités (4.25) et (4.26) prouvent la stabilité asymptotique globale uniforme du système perturbé (4.22) et le théorème suivant peut être énoncé :

Théorème 11 *La loi de commande proposée (4.18–4.19) rend le système (4.17)* ***globalement uniformément asymptotiquement stable*** *si les conditions suivantes sont respectées : :*

$$\begin{cases} r_2 \geq c, \\ r_1 \geq r_2 + c, \\ k_p \geq \frac{1}{\varepsilon^2} \\ k_d \geq \frac{1}{\varepsilon} \end{cases} \tag{4.27}$$

pour un ε constant positif suffisamment petit .

4.5 Validation numérique des résultats

Des simulations sous l'environnement Matlab/Simulink, ont été réalisées pour évaluer les performances de l'approche proposée, à l'aide du modèle du pont roulant $5-ddl$ décrit dans Fig.4.1. Les paramètres du pont roulant sont pris comme suit :

- Masse du chariot : $m_c = 10kg$
- Masse du rail mobile : $m_w = 20kg$,
- Masse de la charge utile :$m_p = 10kg$.
- La gravité a été fixée : $g = 10m/s^{-2}$.
- Câble de masse négligeable et de longueur variable.

Les incertitudes non structurées sont simulées en tenant compte des perturbations externes inconnues et des forces de frottement mécaniques dans le chariot mobile, la rail mobile et le mécanisme de levage qui sont modélisées à l'aide du modèle de frottement de **LuGre** [Åström2008], car il permet de capturer la plupart du comportement de frottement qui a été, en général, observé expérimentalement dans les systèmes de commande. Les limites maximales des accélérations et des vitesses de mouvement sont fixées comme suit : $\xi_{max} = 1.2m.s^{-1}$, $\dot{\xi}_{max} = 0.6m.s^{-2}$. Toutes les simulations ont été effectuées avec un temps d'échantillonnage fixé à : $t_s = 0.01s$.

4.5.1 planification de trajectoires

Dans les tests proposés, on considère la tâche de déplacer la charge de sa position de départ $P_A = [x_A\ y_A\ l_A]$ à la position cible $P_B = [x_B\ y_B\ l_B]$ sans oscillations.
La courbe en $3D$ qui représente le chemin sans collisions le long duquel la charge doit se déplacer, est représentée dans la Fig.4.3.a. On l'obtient via **Algorithme3** en interpolant la séquence de points de passage définie dans tableau 4.1.

La longueur totale du chemin planifié est $L = 47,3158m$. La fonction temporelle qui spécifie la vitesse de la charge le long du chemin planifiée est illustrée dans Fig.4.3.b,c. Elle est obtenue en appliquant l'algorithme de génération de trajectoire décrit dans la Fig.4.2 et en considérant un mouvement de repos à repos, c'est-à-dire que le système commence son mouvement avec une vitesse et une accélération nulle et le complète avec une vitesse et une accélération nulles.

Les principales caractéristiques de ce profil de mouvement sont les suivantes : $T = 49.4200s$,$T_J = 5s$, $T_{snap} = 5s$. Ce dernier paramètre est introduit pour définir le retard du deuxième filtre dans l'algorithme de planification proposé. Comme on le verra plus loin, ce choix limite les oscillations de charge causées par l'accélération du chariot à un

Tableau 4.1 – Points de passages et paramètres d'interpolation

Point	Coordonnées	$u_i(m)$
P_A	(1, 2, 1)	00.0000
2	(8, 3, 2)	06.0828
3	(15, 4, 3)	13.2242
4	(18, 7, 4)	17.5831
5	(18, 9, 5)	19.8192
6	(18, 11, 6)	22.0552
7	(14, 14, 6)	27.0552
8	(10, 17, 5)	32.1542
9	(10, 18, 4)	33.5685
10	(10, 21, 3)	36.7307
11	(13, 24, 2)	41.0896
P_B	(18, 23, 1)	46.2858

niveau inférieur de $0,15rad$. La paramétrisation par longueur d'arc ainsi que la représentation temporelle des trajectoires générées, obtenues par Algorithme2 et Algorithme3, sont illustrés dans Fig.4.3.d.

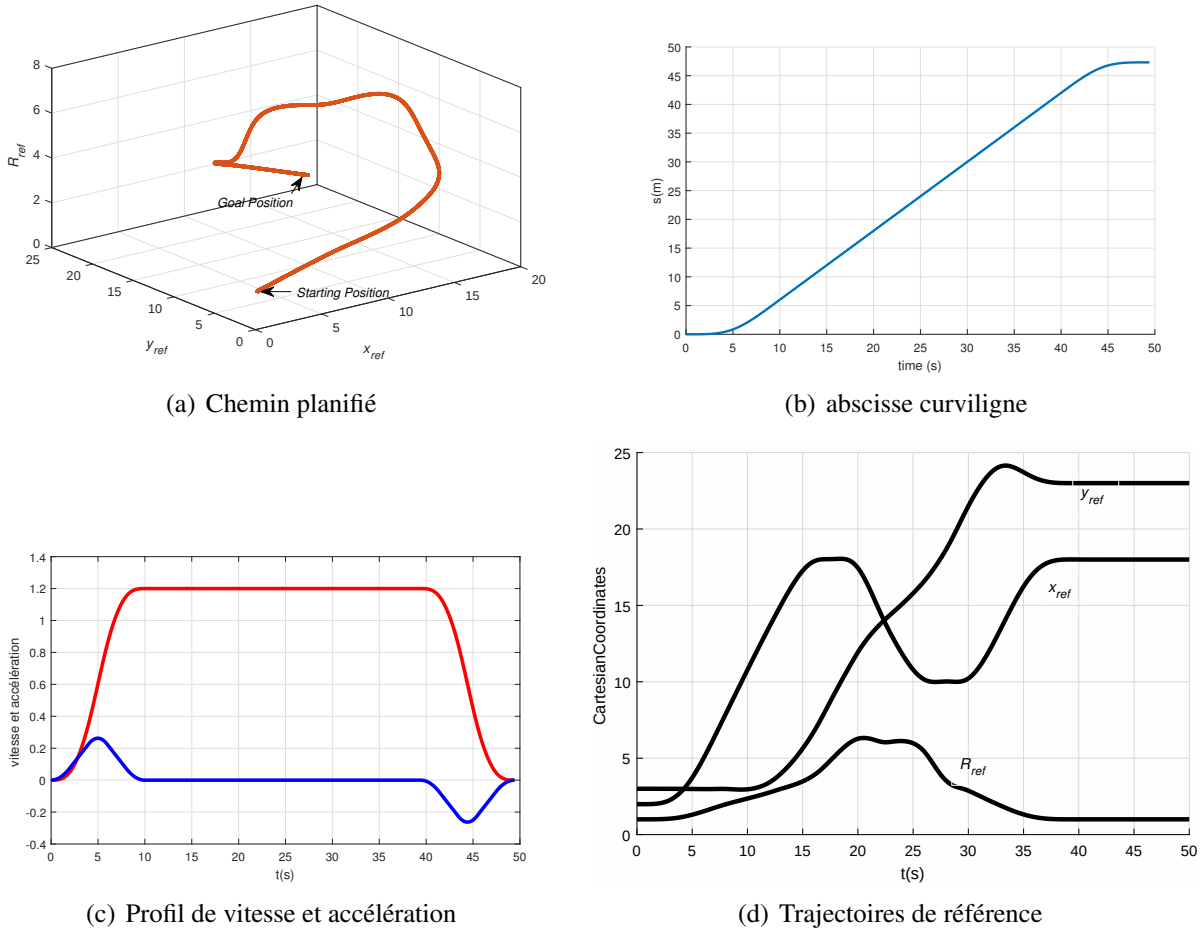
4.5.2 Analyse de contrôle de suivi

Le but de ce test est d'évaluer les performances de l'approche de commande proposé en termes de précision de poursuite, d'atténuation des oscillations de la charge et de la douceur des signaux de commande en considérant la même tâche planifiée ci-dessus. Pour la surface de glissement donnée dans (4.13), nous considérons les paramètres suivants :

$$K_\theta = 3, K_\beta = 2, h = 1$$

Ensuite, pour une étude comparative, trois approches sont utilisées pour la stabilisation de la dynamique (4.17) :

1. Un contrôleur PD avec les paramètres suivants :
   $k_{px} = 9, \quad k_{dx} = 6$
   $k_{py} = 9, \quad k_{dy} = 6$
   $k_{pl} = 20, \quad k_{dl} = 9$
2. Un algorithme de commande par mode glissant conventionnel définie dans [Boutalbi2017], avec des paramètres de contrôle fixés comme :
   $v_\alpha = -M_\alpha sign(\sigma_{1\alpha} + \lambda_\alpha \sigma_{2\alpha})$
   $M_\alpha = \psi_\alpha |\sigma_{1\alpha}| + \Psi_\alpha |\sigma_{2\alpha}| + \delta_{0\alpha}$
   $\alpha = [x, y, l]$
   où $\lambda_\alpha, \psi_\alpha, \Psi_\alpha$ and $\delta_{0\alpha}$ sont des paramètres choisis comme



(a) Chemin planifié

(b) abscisse curviligne

(c) Profil de vitesse et accélération

(d) Trajectoires de référence

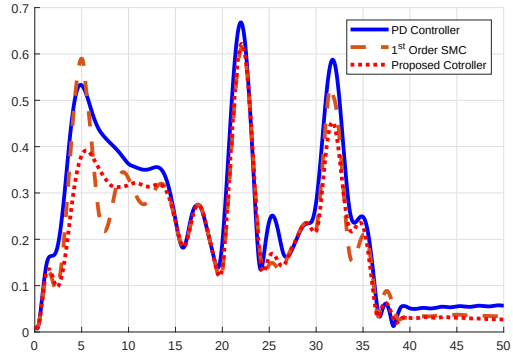
Figure 4.3 – Tâche planifiée

$\lambda_x = 1, \psi_x = 10, \Psi_x = 6, \delta_x = 4$
$\lambda_y = 1, \psi_y = 10, \Psi_y = 7, \delta_y = 4$
$\lambda_l = 1, \psi_l = 20, \Psi_l = 5, \delta_l = 4$

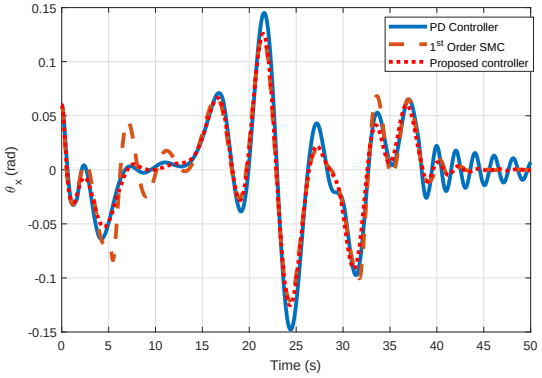
3. Un algorithme SDT discuté ci-dessus (4.20, 4.19) où les paramètres de contrôle sont fixés comme suit :
   $k_{px} = 9, \quad k_{dx} = 6, \quad r_{1x} = 10, \quad r_{2x} = 5$
   $k_{py} = 9, \quad k_{dy} = 6, \quad r_{1y} = 10, \quad r_{2y} = 5$
   $k_{px} = 20, \quad k_{dx} = 9, \quad r_{1x} = 4, \quad r_{2x} = 2$

Le comportement du système simulé sous les contrôleurs ci-dessus est montré dans la Fig.4.4. Les normes cartésiennes correspondantes des erreurs de suivi, comme le montre la Fig.4.4.c, mettent en évidence la grande précision des approches de contrôle discontinues proposées par rapport au contrôleur PD, où la méthode proposée réduit le RMS, Root Mean Square, de l'erreur de suivi avec $RMS_{STD} = 0.222m$, alors que les valeurs RMS des erreurs de suivi sont $RMS_{MGC} = 0.2463m$ et $RMS_{PD} = 0.2967$ pour les contrôleurs basés sur le mode glissant conventionnelle et le PD classique, respectivement. On note que la norme cartésienne de l'erreur de suivi est donnée sous la forme :

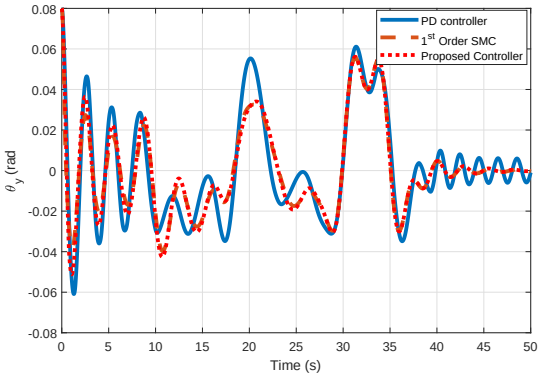
$$e_N = \sqrt{(x-x_d)^2 + (y-y_d)^2 + (l-l_d)^2} \tag{4.28}$$



(a) Norme Cartésien d'erreur.



(b) Angle d'oscillation θ_x



(c) Angle d'oscillation θ_y

Figure 4.4 – Performances de suivi

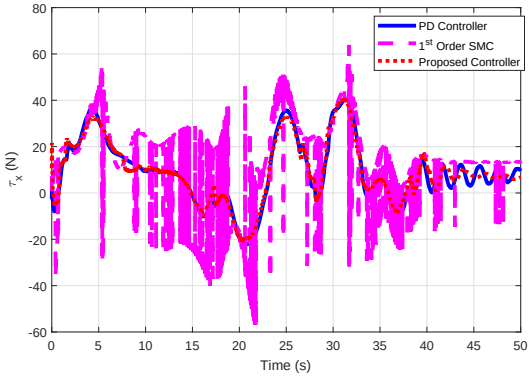
De plus, les Figs.4.4.a et 4.4.b illustrent les performances positives de l'algorithme SDT proposé dans l'atténuation des indésirables oscillations de la charge .

Les entrées de commande correspondantes sont illustrées par Fig.4.5. En outre des résultats ci-dessus liés à la précision de poursuite et l'atténuation importante des oscillations indésirables de la charge pour l'approche SDT proposée, on peut clairement noter que le signal de commande est lisse pour l'approche SDT proposée par rapport au mode glissant classique (d'ordre1).

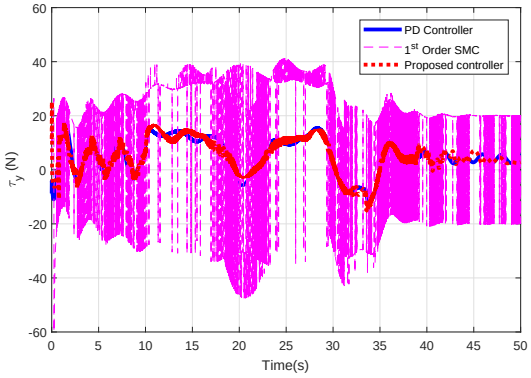
4.5.3 Analyse d'anti-ballant

Le but de cet essai est de mettre en évidence l'importance du contrôleur anti-ballant proposé dans le comportement général du système. Par conséquent, deux contrôleurs ont été considérés pour une étude comparative :

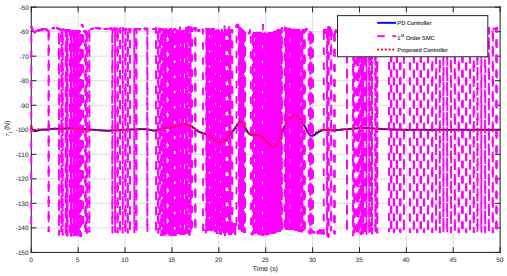
1. Un contrôleur SDT basé sur la surface de glissement défini dans (4.13) avec les paramètres suivants :
$$k_\theta = 3, \quad k_\beta = 1, \quad h = 1$$
2. Un contrôleur SDT basé sur la surface de glissement sans la partie d'anti-ballant, c'est-à-dire :
$$k_\theta = 0, \quad k_\beta = 0, \quad h = 0$$



(a) Commande de "Travelling"



(b) Commande de "Traverse"



(c) Commande de "levage"

Figure 4.5 – Signaux de commande

alors

$$\mu = e$$

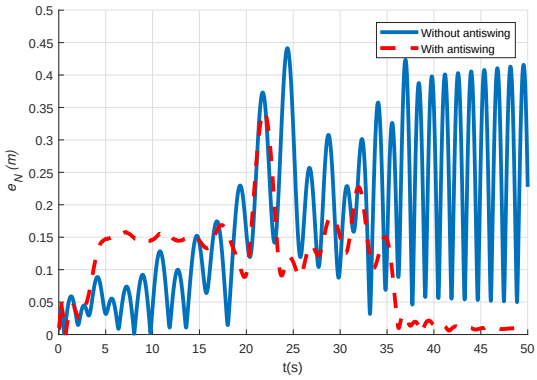
Dans les deux tests, les paramètres de contrôle par rétroaction sont fixés comme suit :

$$k_{px} = 9, \;\; k_{dx} = 6, \;\; r_{1x} = 10, \;\; r_{2x} = 5$$
$$k_{py} = 9, \;\; k_{dy} = 6, \;\; r_{1y} = 10, \;\; r_{2y} = 5$$
$$k_{pl} = 20, \;\; k_{dl} = 9, \;\; r_{1l} = 4, \;\; r_{2l} = 2$$

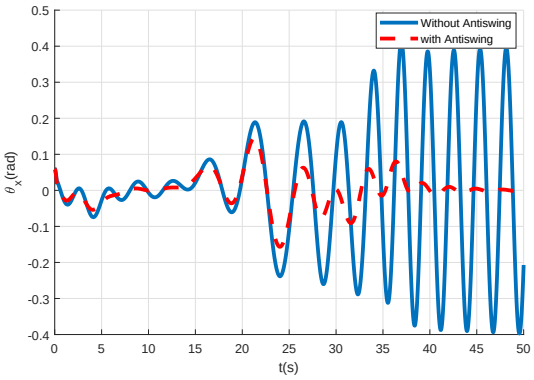
La norme cartésienne des erreurs de poursuite et les balancements qui caractérisent le mouvement du pont roulant en appliquant les contrôleurs ci-dessus sont illustrées dans Fig.4.6.

On constate clairement l'importance de la partie anti-ballant dans le mouvement du système. Sans la partie anti-ballant, le mouvement de la charge est caractérisé par des oscillations lentement amortis. Ces oscillations sont naturellement provoquées par les phases d'accélération et de décélération caractérisant le mouvement du système le long des axes x et y, ainsi que la nature courbé du chemin planifié. L'introduction du système d'anti-ballant permet d'atténuer de manière significative les oscillations de la charge et d'éliminer l'excitation de la résonance paramétrique du système pendant le mouvement de la charge, comme le montrent les Figs.4.6.b et 4.6.c. Les oscillations de la charge sont limitées dans ce cas (avec un anti-ballant) à celles provoquées durant les phases d'accélération et décélération du chariot et rail mobiles. De plus, l'atténuation des balancements

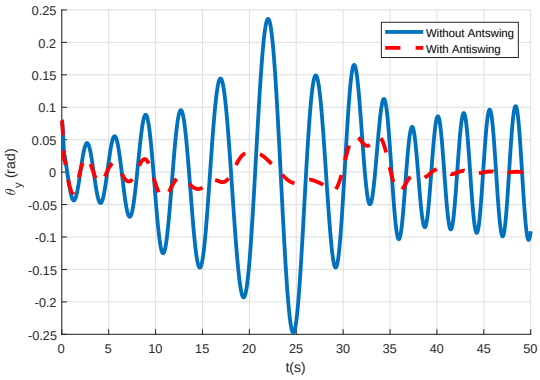
de la charge par l'intermédiaire de la partie anti-ballant améliore significativement la précision de suivi et surtout dans la position final (cible) de la charge, 4.6.b.



(a) Norme cartésien d'erreur de suivi



(b) Angle d'oscillation θ_x



(c) Angle d'oscillation θ_y

Figure 4.6 – Performances du système d'anti-ballant

4.6 Conclusion

Dans ce chapitre, nous avons proposé une stratégie de contrôle pour une manipulation efficace des charges lourdes à l'aide d'un système de pont roulant à plusieurs degrés de liberté. Il est composé de deux tâches principales, la génération de trajectoires de référence et la conception de la commande par rétroaction. Les trajectoires de référence sont générées à l'aide d'un algorithme de planification découplé. Il vise d'abord à concevoir un chemin sans collisions, puis de calculer une fonction temporelle le long du chemin soumis à des contraintes différentielles. L'idée clé de la conception de la commande par rétroaction consiste à (i) définir la surface de glissement comme une sortie couplée qui assure à la fois le suivi précis des trajectoires prédéfinies, l'atténuation de toute balancement de la charge peut être confrontée pendant la séquence de transfert et élimine l'excitation paramétrique des oscillations de la charge (ii) puis à concevoir un contrôleur discontinu de second ordre basé sur l'algorithme Twisting qui force la stabilisation de la sortie spécifiée à zéro malgré les incertitudes du modèle et les perturbations externes. La stabilité asymptotique de l'approche de contrôle proposée est établie en utilisant le théorème de Lyapunov. Des tests numériques sont effectués pour illustrer les performances de l'approche de contrôle proposée.

5

Commande d'un bras manipulateur

Dans ce chapitre nous proposons une commande adaptative pour la manipulation des système robotiques avec des paramètres dynamiques totalement inconnus.

Sommaire

5.1 Introduction 93
5.2 Modélisation du système et formulation du problème 93
5.3 Conception du Twisting contrôleur sous des conditions d'incertitudes 94
5.3.1 Conception du contrôleur 95
5.3.2 Analyse de stabilité 96
5.4 Commande avec des paramètres inconnus 96
5.4.1 Conception du contrôleur 97
5.4.2 Analyse de stabilité 98
5.5 Tests pratiques 100
5.5.1 Prototype d'essai 100
5.5.2 Développement et tests des contrôleurs 101
5.5.3 Discussion des résultats 107
5.6 Conclusion 111

5.1 Introduction

Dans ce chapitre nous considérons le problème de commande d'un bras manipulateur avec plusieurs degrés de liberté. Les bras manipulateurs sont des exemples typiques de systèmes non linéaires avec une dynamique fortement couplée. Ce type de système peut être utilisé pour exécuter des différentes tâches sous différentes conditions d'incertitudes.

Le problème d'adaptation de l'approche découplée chemin-trajectoire, pour ce type de système sous différentes exigences et différentes contraintes est largement discuté dans la littérature [Bobrow1985, Lo Bianco2013, Lo Bianco2017]. Différentes approches, que ce soit adaptatives [Slotine1987, Grimm1990, Brogliato1991] ou robustes [Abdallah1991, de Wit1996, Liu2000, Spong2004, Sage2010], ont été aussi proposées pour résoudre les problèmes de suivi des trajectoires prédéfinies.

Le but principal du reste de ce chapitre est d'étendre le concept de commande par mode glissant d'ordre deux défini dans le chapitre 4 pour la manipulation des bras manipulateurs. Cela est motivé :

— d'une part, par les propriétés prometteuses de ce type de contrôleur, telles que la convergence en temps finis, la précision de régulation et la robustesse face aux incertitudes non paramétriques.

— d'autre part, par les exigences strictes des nouvelles applications des systèmes robotiques dans différents domaines.

Tout d'abord, le twisting algorithme dépendant de l'état (SDTA) est proposé pour une manipulation robuste du robot avec des paramètres dynamiques incertains. Cet algorithme est ensuite étendu pour la manipulation du système robotique sous la considération du cas où les paramètres dynamiques du robot sont totalement inconnus. Une loi d'adaptation est définie pour estimer les paramètres du système en assurant la convergence de l'erreur de suivi vers zéro et l'atténuation de phénomène de Chattering. Ce chapitre est supporté par une analyse théorique et pratique des performances du contrôleur proposé.

5.2 Modélisation du système et formulation du problème

Les Bras de robots à chaîne ouverte, ou bras série, sont des exemples bien connus de systèmes mécaniques à dynamique non linéaire et fortement couplée. Essentiellement, ils sont construits en reliant des corps rigides, appelés segments, au moyen d'articulations, de sorte qu'un mouvement relatif entre les segments adjacents devient possible. L'actionnement des articulations, généralement par des moteurs électriques, permet au robot de se déplacer et d'exercer des forces/couples d'une manière désirée. En appliquant une technique de modélisation standard telle que les formalismes d'Euler-Lagrange ou de Newton-Euler, l'équation de mouvement pour un bras manipulateur série à n segments, avec n variables articulaires (rotoïde/prismatique), $q_1, q_2, ...q_n$, (q_i, décrit la i^{eme} variable articulaire) et n actionneurs peut être décrite sous forme matricielle comme suit [Spong2004] :

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau \tag{5.1}$$

où :
$q \in R^n$ est le vecteur de coordonnées généralisées donné comme :$q = [q_1\ q_2\ ...q_n]^T$

$\tau \in R^n$ est le vecteur de couple défini comme :$\tau = [\tau_1 \ \tau_2 \ ...\tau_n]^T$
$D(q)$ est une matrice $(n \times n)$ représente la matrice d'inertie.
$C(q,\dot{q}) \in R^{n \times n}$ représente les forces de Coriolis et de centrifuge.
$G(q) \in R^n$ est un vecteur qui contient les termes de gravité.
$F(\dot{q}) \in R^n$ est un vecteur qui contient les frottements des articulations (ce terme sera négligé dans la phase de conception de la commande).

On peut noter les propriétés suivantes :
Propriété 1 :

$$\begin{cases} \forall \xi \in R^n m_1 ||\xi|| \leq \xi D(q) \xi^T \leq m_2 ||\xi|| \\ m_2 > m_1 > 0 \end{cases} \tag{5.2}$$

d'ou les paramètres m_1 et m_2 sont des constants positifs et $||\xi||$ est la norme euclidienne du vecteur ξ.

Propriété 2 : Les matrices d'inertie et centripète satisfont la propriété d'anti-symétrie suivante :

$$\forall \xi \in R^n : \xi (C(q,\dot{q}) - \frac{1}{2}\dot{D}(d)) \xi^T = 0 \tag{5.3}$$

Propriété 3 : L'équation dynamique (5.1) du robot manipulateur peut être écrite sous la forme linéaire :

$$Y(q,\dot{q},\ddot{q})\theta = D(q)\ddot{q} + C(q,\dot{q}) + G(q) + F(\dot{q}) = \tau \tag{5.4}$$

où $Y \in R^{n \times p}$ est une matrice de régression qui ne dépend que des grandeurs cinématiques $(q,\ \dot{q},\ \ddot{q})$ et $\theta \in R^p$ est un vecteur de coefficients dynamiques ou paramètres de base.
Du fait que la combinaison linéaire (5.4) , notamment la matrice Y et θ, n'est pas unique, la forme linéarisé (5.4) est utilisée en général pour résoudre les problèmes de contrôle adaptatif ou l'estimation des paramètres inertiels des bras manipulateurs.

Propriété 4 : Limitation de la gravité et des vecteurs de frottement :

$$||G(q)|| \leq c_G \tag{5.5}$$

$$F(\dot{q}) \leq c_{1F} + c_{2F}||\dot{q}|| \leq c_F \tag{5.6}$$

Pour certaines constantes positives $c_G,\ c_{1F},\ c_{2F},\ c_F$.

Propriété 5 : Le vecteur de coordonnées généralisées q dans la dynamique du robot (5.1) est prouvé comme un vecteur de sorties plates, ce qui implique que le système peut être linéarisé via une loi de commande par rétroaction appropriée [Rigatos2012].

5.3 Conception du Twisting contrôleur sous des conditions d'incertitudes

Dans cette section, un algorithme de commande par mode glissant d'ordre deux, basé sur le twisting algorithme développé dans le chapitre précédent, est conçu pour assurer la manipulation robuste du bras de robot sous des conditions d'incertitudes paramétriques et non paramétriques. La stabilité du système est prouvée par la méthode de Lyapunov. La

tâche de contrôle de base du robot consiste à calculer le vecteur de couple requis pour les actionneurs afin d'exécuter des tâches prédéfinies à partir du vecteur désiré des variables articulaires et de leurs vitesses, noté par $q_d(t)$ et $\dot{q}_d(t)$. Les trajectoires faisables doivent satisfaire à certaines exigences de base, telles que : les limites maximales des actionneurs et un certain degré de continuité, qui doivent être d'au moins C^2.

5.3.1 Conception du contrôleur

La conception de SMC consiste toujours à choisir une surface de glissement appropriée, puis à définir une loi de commande discontinue efficace qui impose l'existence d'un mode réel glissant (en temps fini) au voisinage de la surface de glissement choisie.
Introduisons le vecteur d'erreur de suivi défini comme :

$$e = q - q_d \tag{5.7}$$

Tenant en compte de la dynamique du robot (5.1), la deuxième dérivée de l'erreur (5.7) est donnée comme suit :

$$\ddot{e} = -\ddot{q}_d - D^{-1}(q)(C(q,\dot{q}) + G(q) + F(\dot{q})) + D^{-1}\tau \tag{5.8}$$

En se basant sur le fait que le vecteur de coordonnées généralisées q dans la dynamique (5.1) est un vecteur de sortie plate pour le système étudié, Propriété 5, une rétroaction linéarisante peut être définie comme [Rigatos2012, Fliess1995] :

$$\tau = \hat{D}(q)\ddot{q}_d + \hat{D}(q)U + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) \tag{5.9}$$

$$\begin{cases} \hat{D} = D + \tilde{D} \\ \hat{C} = C + \tilde{C} \\ \hat{G} = G + \tilde{G} \end{cases} \tag{5.10}$$

Le système se compose d'une boucle de compensation non linéaire interne et d'une boucle externe avec un signal de commande exogène $U \in R^n$. Où $\hat{(.)}$ représente une valeur approximative du paramètre $(.)$, et la quantité inconnue $\tilde{(.)}$ est introduite pour décrire les incertitudes du modèle dans la matrice $(.)$. Cet écart entre le modèle nominal (5.1) et le modèle (5.9,5.10) peut se découler de différents facteurs.
En substituant cette loi de contrôle dans la dynamique d' erreurs (5.8), on obtient :

$$\begin{cases} \ddot{e} - U + \hat{W}(e,t) = 0 \\ \hat{W}(e,t) = D^{-1}(\tilde{D}\, e \; + \; \tilde{C} + \tilde{G}) \end{cases} \tag{5.11}$$

où les limites d'incertitudes dans le vecteur $\hat{W}(e,t) \in R^{(n\times 1)}$ dépend du niveau des incertitudes de connaissance des paramètres .
La dynamique résultante dans (5.11) est non pas seulement indépendante des variables articulaires et leurs vitesses, mais elle est également découplée au niveau des articulations. Ainsi, le problème de contrôle est transformé en un simple problème de conception pour un système linéaire contrôlable ; composé de n double intégrateurs découplés.

Une commande équivalente à celle définie dans le chapitre précédent peut être donc introduite pour assurer la convergence de l'erreur de suivi vers zéro ainsi qu'un bon suivi des trajectoires planifiées.

$$u_i = -k_{pi}e_i - k_{di}\dot{e}_i - r_{1i}sign(e_i) - r_{2i}sign(\dot{e}_i) \tag{5.12}$$
$$i = \{1, \cdots, n\}$$

5.3.2 Analyse de stabilité

On peut citer le théorème suivant :

Théorème 12 *La loi de commande (5.8, 5.12) rend l'erreur de suivi (5.7)* ***globalement uniformément asymptotiquement stable*** *si les conditions suivantes sont respectées : :*

$$\begin{cases} r_2 \geq c, \\ r_1 \geq r_2 + c, \\ k_p \geq \frac{1}{\varepsilon^2} \\ k_d \geq \frac{1}{\varepsilon} \end{cases} \tag{5.13}$$

pour un ε constant positif suffisamment petit. c est un constant positif sachant que $|\hat{w}_i| < c$, $i = 1, \cdots, n$.

Preuve

L'analyse de stabilité est similaire à celle détaillé dans le cas de stabilisation de la surface de glissement (4.13) sous la commande (4.20) dans le chapitre précédent. La surface de glissement est résumée dans ce cas en :

$$\mu = e \tag{5.14}$$

ce qui implique la convergence de l'erreur de suivi et sa dérivée vers zéro en temps fini.

5.4 Commande avec des paramètres inconnus

Le but de cette partie est l'extension de l'approche discutée ci-dessus pour la manipulation robuste de robots avec grandes incertitudes ou des systèmes avec des paramètres dynamiques totalement inconnus. Les principaux avantages de l'algorithme SDT (5.12) sont le niveau élevé de la précision de suivi de trajectoire, la réponse dynamique rapide et la robustesse face aux incertitudes du modèle. Les performances du contrôle de l'AT sont principalement limitées par l'exigence fondamentale de la meilleure connaissance des limites des incertitudes, ce qui est une tâche difficile d'un point de vue pratique. Ce problème peut être compensé par l'augmentation et la saturation des gains du régulateur discontinu M_1 et M_2 (r_1 et r_2). Cette dernière solution peut exciter les phénomènes indésirables du chattering, en particulier dans le cas où les paramètres du système sont inconnus. L'idée principale pour l'atténuation du chattering, dans ce chapitre, consiste à réduire le niveau d'incertitude en supportant l'algorithme platitude-SDT discuté avant par un algorithme d'identification des paramètres dynamiques du système. Cette stratégie garantit l'existence d'un mode de glissement réel au voisinage de la surface de glissement avec des gains relativement faibles, ce qui permet une application en toute sécurité des contrôleurs discontinus et sans provoquer de phénomènes de chattering.

5.4.1 Conception du contrôleur

Une approche bien connue pour estimer les paramètres dynamiques de robots manipulateurs consiste à considérer l'erreur de suivi filtrée définie comme suit [Slotine1987, Slotine1988] :

$$\begin{cases} s = \dot{q} - \dot{q}_r \\ v = \dot{q}_r = \dot{q}_d - Ae \end{cases} \tag{5.15}$$

avec $A \in R^{n\times n}$ est une matrice diagonale définie positive.
Soustrayant les deux côtés de la dynamique du robot dans (5.1) de, $D(q)\dot{v} + C(q,\dot{q})v + G(q)$, pour obtenir :

$$D(q)\dot{s} + C(q,\dot{q})s = \tau - D(q)\dot{v} - C(q,\dot{q})v - G \tag{5.16}$$

Ensuite, reconsidérant la loi de commande linéarisante basée sur la propriété de platitude dans l'équation (5.9) pour obtenir :

$$D(q)\dot{s} + C(q,\dot{q})s = \hat{D}(q)\ddot{q}_d + \hat{D}U + \hat{C}(q,\dot{q})\dot{q} + \hat{G} - D(q)\dot{v} - C(q,\dot{q})v - G \tag{5.17}$$

En substituant $\ddot{q}_d$ et $\dot{q}$ avec leur valeur correspondante de (5.15), on peut obtenir :

$$D(q)\dot{s} + C(q,\dot{q})s = U + \tilde{D}(q)\dot{v} + \tilde{C}(q,\dot{)}v + \tilde{G} + \hat{W} \tag{5.18}$$

avec :

$$\begin{cases} \tilde{D}(q) = D(q) - \hat{D} \\ \tilde{C}(q,\dot{q}) = C(q,\dot{q}) - \hat{C}(q,\dot{q}) \\ \tilde{G}(q) = G(q) - \hat{G}(q) \end{cases} \tag{5.19}$$

$\hat{W}(e,\dot{e},\dot{q},\ddot{q})$ représente un vecteur additif inconnu pour décrire toutes les perturbations externes et les erreurs de modélisation qui affectent le système.
En se basant sur la propriété 2, l'erreur dynamique de (5.18) peut être réécrite comme suit :

$$\begin{cases} D(q)\dot{s} + C(q,\dot{q})s = U + Y(q,\dot{q},v,\dot{v})\tilde{\theta} + \hat{W} \\ \tilde{\theta} = \hat{\theta} - \theta \end{cases} \tag{5.20}$$

où : $\hat{\theta}$ est le vecteur d'estimations en temps réel du vecteur θ des paramètres constants.
Les principaux défis de la conception d'une commande adaptative dans la suite de cette partie consiste à définir un estimateur des paramètres en ligne approprié $f(Y,s)$, qui fournit des estimations des paramètres inconnus $\hat{\theta}$ à chaque instant, et une loi de commande appropriée U qui garantit la convergence de l'erreur de suivi à zéros, c.-à-d. la stabilité de l'erreur dynamique (5.20).
Prenant en compte que la loi de commande dans la dynamique (5.20) a un degré relatif deux par rapport au vecteur d'erreur de suivi e, un TA peut être défini comme suit :

$$\begin{cases} U = U_c + \Delta U \\ U_c = -K_D s \\ \Delta U = -R_1 sign(e) - R_2 sign(\dot{e}) \end{cases} \tag{5.21}$$

avec $K_D = diag(k_{d1}, k_{d2}, ..., k_{dn})$, $R_1 = diag(r_{11}, r_{12}, ..., r_{1n})$ and $R_2 = diag(r_{21}, r_{22}, ..., r_{2n})$ étant des matrices diagonales positives constantes et :
$sign(e) = [sign(e_1)\ sign(e_2)\ ...sign(e_n)]^T$
$sign(\dot{e}) = [sign(\dot{e_1})\ sign(\dot{e_2})\ ...sign(\dot{e_n})]^T$
La partie continue U_c de la loi de commande (5.21) est également reformulée sous la forme d'un contrôleur PD classique en tant que :

$$U_c = -K_P\, e - K_D\, \dot{e} \tag{5.22}$$

où, la matrice définie positive K_p est donnée par :

$$K_P = AK_D \tag{5.23}$$

La loi de commande de l'équation (5.21) prend la forme générale (pour un système MIMO : *Multi Input Multi Output*) d'un algorithme SDT défini dans la section précédente.
De plus, le choix appropriée de la loi d'adaptation doit garantir que l'estimation du vecteur des paramètres $\hat{\theta}$ et la vitesse d'adaptation $\dot{\hat{\theta}}$. Dans ce chapitre, nous conservons le même choix de la loi de mise à jour proposée dans [Slotine1987, Slotine1988], donnée comme suit :

$$\dot{\hat{\theta}} = -\Gamma Y^T(q, \dot{q}, v, \dot{v})s \tag{5.24}$$

avec Γ est une matrice diagonale définie positive et $\hat{\theta}_0 \geq 0$.
Ce choix garantit que les paramètres estimés $\theta(t)$ et la vitesse d'adaptation $\dot{\theta}(t)$ sont bornés.

5.4.2 Analyse de stabilité

Théorème 13 *L'algorithme de commande défini dans (5.21) ainsi que la loi de mise à jour donnée dans (5.24) rendent le système* ***globalement uniformément asymptotiquement stable*** *si les conditions suivantes sont respectées :*

$$\begin{cases} r_2 i \geq c_i, \\ r_1 i \geq r_{2i} + c_i, \\ K_D, A \geq 0 \\ \Gamma \geq 0 \end{cases} \tag{5.25}$$

Preuve

Pour prouver la stabilité de l'algorithme de contrôle proposé, nous considérons la fonction de Lyapunov suivante :

$$W = \frac{1}{2}s^T Ds + \sum_{i=1}^{n} r_{1i}k_{pi}|e_i| + \frac{1}{2}\tilde{\theta}^T\Gamma\tilde{\theta} \tag{5.26}$$

avec r_{1i} and k_{pi} sont les éléments diagonaux des matrices diagonales R_1 et K_p, respectivement. Prenant la dérivée de la fonction (5.26) pour obtenir :

$$\dot{W} = s^T D\dot{s} + \frac{1}{2}s^T \dot{D}s + \sum_{i=1}^{n} r_{1i}k_{pi}\ \ \dot{e}_i\ \ sign(e_i) + \tilde{\theta}^T\Gamma\dot{\hat{\theta}} \tag{5.27}$$

ou :

$$\dot{W} = s^T D\dot{s} + \frac{1}{2} s^T \dot{D} s + \dot{e}^T \ R_1 K_P \ sign(e) + \tilde{\theta}^T \Gamma \dot{\hat{\theta}} \tag{5.28}$$

substituant $(D\dot{s})$ par son équivalent de l'équation (5.20) pour obtenir :

$$\dot{W} = s^T [Y(q,\dot{q},v,\dot{v})\tilde{\theta} - C(q,\dot{q})s + U + \hat{W}] + \frac{1}{2} s^T \dot{D} s + \dot{e}^T R_1 K_P \ sign(e) + \tilde{\theta}^T \Gamma \dot{\hat{\theta}} \tag{5.29}$$

ou :

$$\dot{W} = s^T U - s^T [C(q,\dot{q}) - \frac{1}{2}\dot{D}]s + s^T \hat{W} + \dot{e}^T R_1 K_P \ sign(e) + \tilde{\theta}\Gamma\dot{\hat{\theta}} + s^T Y(q,\dot{q},v,\dot{v})\tilde{\theta} \tag{5.30}$$

En utilisant la **Propriété 2** et en définissant la loi de mise à jour définie dans (5.24) pour trouver :

$$\dot{W} = s^T U + s^T \hat{W} + \dot{e}^T R_1 K_P \ sign(e) - \tilde{\theta}^T \Gamma \Gamma^{-1} Y(q,\dot{q},v,\dot{v})s + s^T Y(q,\dot{q},v,\dot{v})\tilde{\theta} \tag{5.31}$$

ou :

$$\dot{W} = s^T U_c + s^T \Delta U + s^T \hat{W} + \dot{e}^T R_1 K_P \ sign(e) \tag{5.32}$$

Introduire les lois de commande U_c et ΔU comme dans (5.21) pour obtenir :

$$\dot{W} = -s^T K_D s - (\dot{e} + Ae)^T K_D [R_1 sign(e) + R_2 sign(\dot{e}) - \hat{W}] + \dot{e}^T R_1 K_P \ sign(e) \tag{5.33}$$

ou :

$$\dot{W} = -s^T K_D s - \dot{e}^T K_D (R_2 sign(\dot{e}) - \hat{W}) - e^T A K_D [R_1 sign(e) + R_2 sign(\dot{e}) - \hat{W}] \tag{5.34}$$

Ainsi, les conditions suffisantes pour garantir la stabilité du système sont données comme suit :

$$\begin{cases} r_2 i \geq c_i, \\ r_1 i \geq r_{2i} + c_i, \\ K_D, A \geq 0 \end{cases} \tag{5.35}$$

De plus, exploitant la stabilité du système, *Propriété 6* du modèle dynamique et le fait que les matrices A, R_1, K_D et Γ sont définies diagonales positives, le système de comparaison suivant peut être noté :

$$0 \leq W_1 \leq W \leq W_2 \tag{5.36}$$

$$\dot{W} \leq -W_3 \tag{5.37}$$

avec :

$$W_1 = \lambda_{min}(D)||s|| \tag{5.38}$$

$$W_2 = \lambda_{max}(D)||s|| + \lambda_{max}(\Gamma)||\hat{\theta}|| + \lambda_{max}(AR_1K_D)|e| \tag{5.39}$$

$$W_3 = \lambda_{min}(K_D)||s|| + \lambda_{min}(M_1)|\dot{e}| + \lambda_{min}(M_2)|e| \tag{5.40}$$

d'où M_1 and M_2 sont des matrices diagonales définies comme :

$$M_1 = K_D diag(r_{2i} - c_i)$$

$$M_2 = K_P diag(r_{2i} - r_{1i} - c_i)$$

Alors la stabilité asymptotique du système peut être prouvée en utilisant les théorèmes 1 et 2.

5.5 Tests pratiques

Une validation expérimentale de la méthode de contrôle proposée dans ce chapitre et une comparaison avec des contrôleurs existants sont analysées dans cette partie, à l'aide du prototype du laboratoire Bras Manipulateur *Lightweight Robai Gamma 1500*. Ici, on peut noter que les performances des approches de contrôle adaptatif sont liées aux différents facteurs, tels que les tâches spécifiées, les incertitudes paramétriques et les conditions initiales. Par conséquent, deux trajectoires de référence ont été considérées dans les essais qui suivent, les signaux de référence sinusoïdaux et les références polynomiales conventionnelles. Théoriquement, les trajectoires de référence sinusoïdales sont riches en harmoniques, ce qui permet une meilleure estimation des paramètres du système. De plus, des tests sur les incertitudes paramétriques sont effectués en attachant une charge supplémentaire à l'extrémité du robot.
Dans la conception des lois de commande discontinues, la fonction de signe est approximée en utilisant la fonction saturation définie comme :

$$sat(x) = \begin{cases} sign(x) & \text{if } |x| \geq x_0 \\ x, & \text{otherwise} \end{cases} \tag{5.41}$$

pour un constant positif x_0, qui est fixé dans tous les tests suivants comme $x_0 = 0.01$.
De plus, on note que les dérivées temporelles du vecteur d'erreur de suivi sont estimées en temps réel à l'aide des différenciateurs numériques, basés sur le contrôleur de mode glissant Super-Twisting [Levant1998]. Le temps d'échantillonnage est fixé comme, $T_s = 0.01s$.

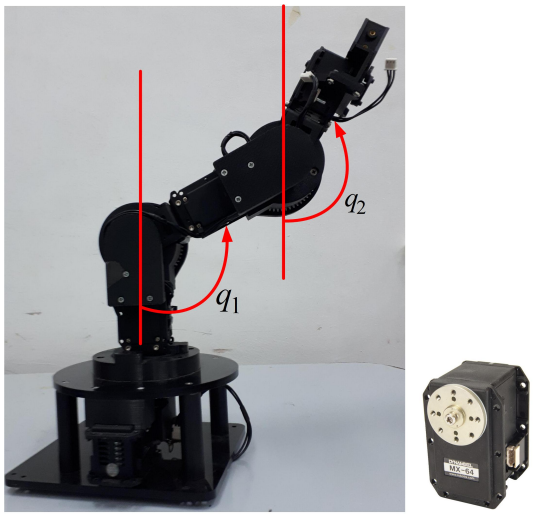


Figure 5.1 – Prototype de test : Cyton Gamma 1500 robot et Servomoteur Dynamixel

5.5.1 Prototype d'essai

Le prototype de test est illustré dans Fig.5.1. Il s'agit d'un modèle réduit, avec deux degrés de liberté 2DoFs, du modèle "Robai Gamma 1500". Le mouvement du robot est réalisé à l'aide de deux servomoteurs DC MX 64, de la famille Dynamixel, illustrés à droite de la Fig.5.1 et avec des réducteurs dont le rapport de réduction est $k_r = \frac{60}{5}$. Ces

servos sont connectés et contrôlés par un protocole de communication série. Les servos dynamixel de la série MX64 sont équipés d'un codeur de position sans contact (magnétique) avec une résolution de position de 12 *bits* (4096) et peuvent fonctionner dans la plage de 360 *degrés* avec une résolution de 0.088^o. Ils offrent également un couple de 9 *bits* (1023 $units/6N.m$) comme contrôle d'entrée. La bibliothèque *Matlab Dynamixel* de Robotis a été utilisée pour l'intégration numérique des algorithmes de contrôle développés.

5.5.2 Développement et tests des contrôleurs

En négligeant toutes les forces de frottement, le modèle dynamique du prototype d'essai est donné sous la forme simplifiée :

$$\begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

d'où :

$$\begin{gathered} d_{11} = \theta_1, \\ d_{12} = d_{21} = \theta_3 cos(q_2 - q_1) + \theta_4 sin(q_2 - q_1), \\ d_{22} = \theta_2 \\ c_{11} = c_{12} = 0, \\ c_{12} = \theta_4 cos(q_2 - q_1) - \theta_3 sin(q_2 - q_1)\dot{q}_2, \\ c_{21} = -\theta_4 sin(q_2 - q_1) + \theta_3 sin(q_2 - q_1)\dot{q}_1 \\ g_1 = g_2 = 0 \end{gathered}$$

q_1 et q_2 sont les angles de jonction mesurés directement à partir des codeurs de servomoteurs, comme la montre Fig.5.1. Les paramètres inconnus θ_i sont liés de façon non linéaire aux paramètres physiques du système, y compris les masses, les longueurs des axes, les centres de masses *COMs*, les moments d'inertie et la dynamique des actionneurs. Notez ici que les termes gravitationnels et de frottement sont négligés, ce qui simplifie le choix de la matrice de régression Y dans la suite du processus de conception du contrôle. Selon les inégalités dans (5.5,5.6), les dynamiques négligées satisfont aux conditions *d'hypothèse1*.
Définir une entrée de commande comme dans (5.21-5.25), pour obtenir :

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = Y(q, \dot{q}, v, \dot{v}) \begin{bmatrix} \hat{\theta}_1 & \hat{\theta}_2 & \hat{\theta}_3 & \hat{\theta}_3 \end{bmatrix}^T$$

où, la matrice de régression est donnée sous la forme :

$$Y = \begin{bmatrix} \dot{v}_1 & 0 & \dot{v}_2 cos(q_2 - q_1) - \dot{q}_2 v_2 sin(q_2 - q_1) & \dot{v}_2 sin(q_2 - q_1) - \dot{q}_2 v_2 cos(q_2 - q_1) \\ 0 & \dot{v}_2 & \dot{v}_1 cos(q_2 - q_1) - \dot{q}_1 v_1 sin(q_2 - q_1) & \dot{v}_1 sin(q_2 - q_1) - \dot{q}_1 v_1 cos(q_2 - q_1) \end{bmatrix}$$

Ainsi, on peut définir une loi d'estimation des paramètres comme dans l'équation (5.24) et l'algorithme de contrôle de suivi comme dans (5.21).
Afin de bien évaluer les performances de l'approche de contrôle proposée, une étude comparative sera considérée dans les prochains essais, où quatre contrôleurs ont été utilisés :

1. En ce qui concerne la dynamique des erreurs dans (5.8) et en utilisant les propriétés de bornitude des paramètres dynamiques du robot, un algorithme SDT sans modèle (*free model*) peut être conçu directement sans considération de la loi d'estimation adaptative (5.24) ni de la rétroaction linéarisante (5.9), comme suit :

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \ddot{q}_{1d} - k_{p11}e_1 - k_{d12}\dot{e}_1 - r_{11}(sign(e_1) + \frac{1}{2}sign(\dot{e}_1)) \\ \ddot{q}_{2d} - k_{p21}e_2 - k_{d22}\dot{e}_2 - r_{21}(sign(e_2) + \frac{1}{2}sign(\dot{e}_2)) \end{bmatrix} \tag{5.42}$$

   Les paramètres r_{11} et r_{21} des deux régulateurs doivent être sélectionnés dans ce cas pour rejeter toutes les effet des incertitudes de modélisation et de perturbation externe. En raison du haut niveau d'incertitude dans ce cas, on peut utiliser la méthode suggérée dans [Taleb2013] pour ajuster dynamiquement les gains de contrôle discontinus r_{11} et r_{21} et assurer l'existence d'un vrai mode de glissement dans le voisinage des surfaces de glissements e_1 et e_2. Dans ce chapitre, nous avons choisi de les fixer comme :

$$k_{p1} = 9, k_{d1} = 6, r_{11} = 4, r_{12} = 2$$
$$k_{p2} = 9, k_{d2} = 6, r_{21} = 4, r_{21} = 2$$

2. Un contrôleur adaptatif conventionnel avec une loi de mise à jour des paramètre comme dans (5.24) et un contrôleur PD classique [Slotine1988] :

$$U = U_c = -K_D s$$

   avec des paramètres de commande fixés comme :

$$A = \begin{bmatrix} 3.5 & 0 \\ 0 & 3.5 \end{bmatrix}, K_D = \begin{bmatrix} 2.4913 & 0 \\ 0 & 2.4913 \end{bmatrix}$$

3. Un contrôleur Twisting-Adaptive (5.9, 5.15) avec une loi de mise à jour sous la forme (5.24) et un algorithme de contrôle de suivi des trajectoires basé sur l'approche de contrôle proposée (eq.5.21), comme :

$$\begin{cases} U = U_c + \Delta U \\ U_c = -K_D s \\ \Delta U = -R_1 sign(e) - R_2 sign(\dot{e}) \end{cases}$$

   avec :

$$A = \begin{bmatrix} 3.5 & 0 \\ 0 & 3.5 \end{bmatrix}, K_D = \begin{bmatrix} 2.4913 & 0 \\ 0 & 2.4913 \end{bmatrix}, R_1 = \begin{bmatrix} 1.8 & 0 \\ 0 & 1.8 \end{bmatrix}, R_2 = \begin{bmatrix} 0.9 & 0 \\ 0 & 0.9 \end{bmatrix}$$

   Dans les deux régulateurs adaptatifs, les entrées de commande τ_1 et τ_2 sont ensuite obtenues par la transformation donnée en (5.9). Les paramètres de la loi de mise à jour sont fixés comme suit :

$$\hat{\theta}(0) = \begin{bmatrix} 0.3 & 0.3 & 0.05 & 0.002 \end{bmatrix}^T$$

$$\Gamma = \begin{bmatrix} 0.20 & 0 & 0 & 0 \\ 0 & 0.20 & 0 & 0 \\ 0 & 0 & 0.20 & 0 \\ 0 & 0 & 0 & 0.20 \end{bmatrix}$$

4. En plus, et en exploitant le fait que le vecteur des paramètres θ peut être prédit a priori en utilisant les résultats des essais précédents, on peut également proposer un TSD basé sur la notion de platitude selon les équations (5.9- 5.12).

   Les paramètres suivants peuvent être obtenus grâce à des trajectoires de référence et des entrées de commande appropriées :

$$\theta_{1pred} = 0.1517, \theta_{2pred} = 0.1240, \theta_{3pred} = 0.0867, \quad \theta_{4pred} = 0.0647$$

   Ensuite, une loi de contrôle sous la forme (5.9, 5.12) est conçue. Les paramètres de la loi de contrôle sont fixés pour respecter les conditions du théorème.12 comme :

$$k_{p1} = 9, k_{d1} = 6, r_{11} = 2.8, r_{12} = 1.5$$
$$k_{p2} = 9, k_{d2} = 6, r_{21} = 2.8, r_{21} = 1.5$$

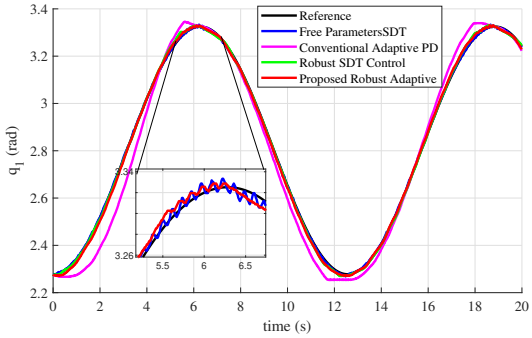
Test1 : Références sinusoïdales

Le but de cette tâche est de déplacer le robot le long d'un ensemble de trajectoires de référence sinusoïdales périodiques :

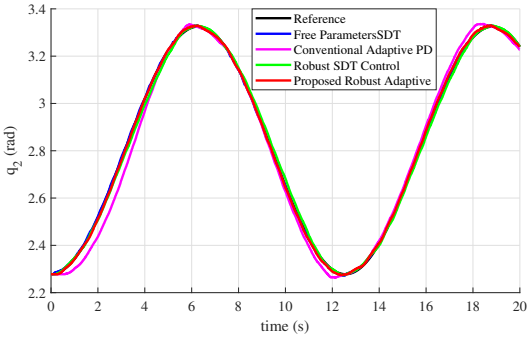
$$q_1(t) = q_{10} + a(1 - cos(\omega t))$$
$$q_2(t) = q_{20} + b(1 - cos(\omega t))$$

avec, $q_{10} = q_{20} = \frac{13\pi}{18}$,$a = b = 1$ and $\omega = 0.5$.

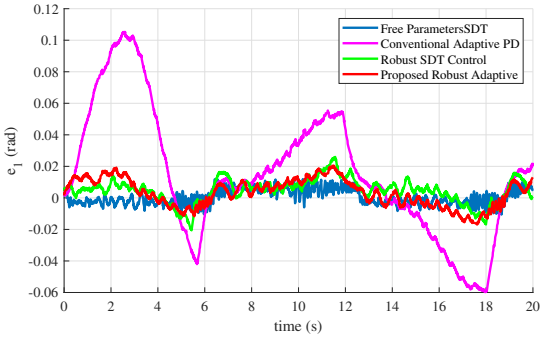
La motivation derrière ce choix de trajectoires de référence sinusoïdales est leur propriété d'excitation, ce qui permet une meilleure estimation des paramètres dynamiques du système. Le comportement global du système sous les contrôleurs ci-dessus dans une plage de temps $T = 20s$ est illustré dans la Fig.5.2.
Les résultats obtenus mettent en évidence le niveau élevé de précision qui caractérise le suivi des trajectoires de référence pour les algorithmes basés sur le Twisting algorithme par rapport au contrôleur Adaptive PD classique. Toutefois, on peut noter aussi le chattering qui caractérise l'algorithme de commande par mode glissant d'ordre deux sans modèle (Fig.5.2.a,b) par rapport au mode glissant basé sur la technique de linéarization par bouclage ou le twisting algorithme basé sur un estimateur de paramètres. De plus, l'évolution des paramètres estimés sous la loi de commande continue (PD classique) et la commande discontinue (Le SDT algorithme) prouve que l'approche proposée permet au contrôleur d'atteindre la précision de suivi requise par un faible niveau de paramètres estimés.
Afin de vérifier les performances de l'approche de contrôle proposée tenant en compte les incertitudes paramétriques, les mêmes essais ont été répétées en considérant que la seconde articulation du robot est attachée avec une charge externe de poids important, donnée comme $m_l = 475g$. L'évolution dynamique du système est reportée dans la Fig.5.3.



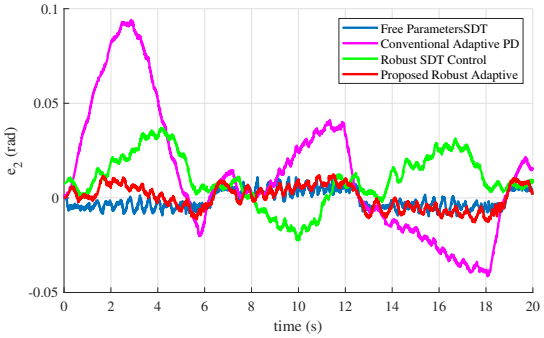
(a) Évolution de la deuxième articulation.



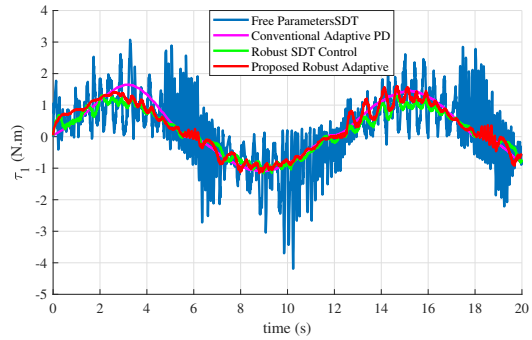
(b) Évolution de la deuxième articulation.



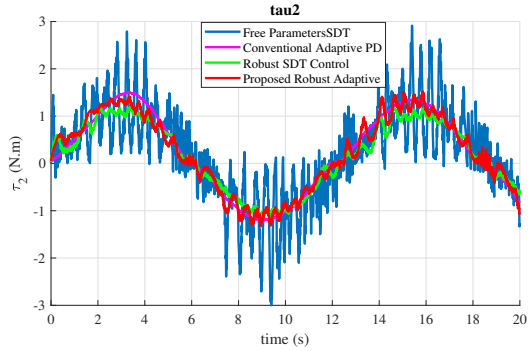
(c) Erreur de suivi dans la première articulation.



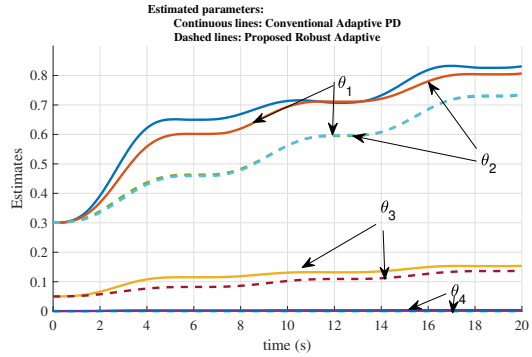
(d) Erreur de suivi dans la deuxième articulation.



(e) Couple appliqué dans la première articulation.

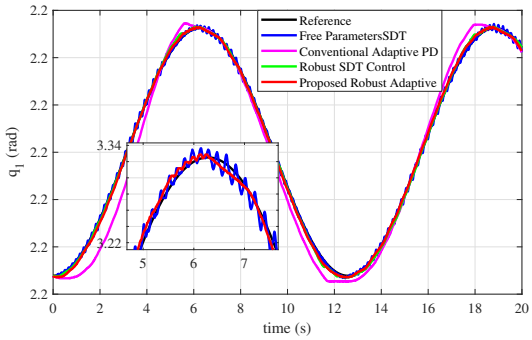


(f) Couple appliqué dans la deuxième articulation.

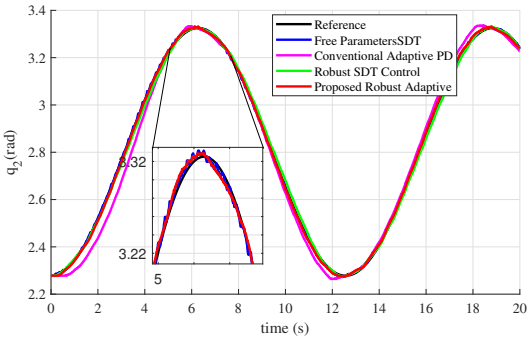


(g) Paramètres estimés.

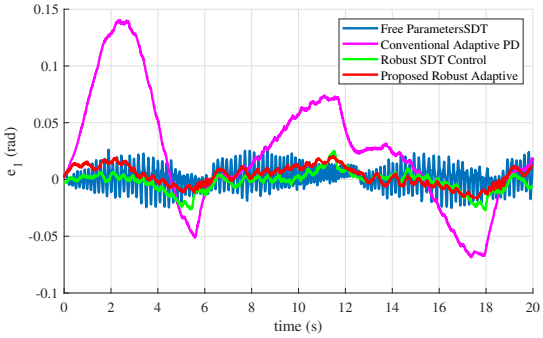
Figure 5.2 – Évolution du robot sans charge attachée et des trajectoires sinusoïdales.



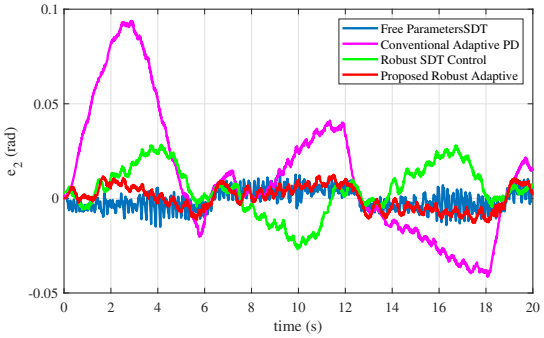
(a) Évolution de la première articulation.



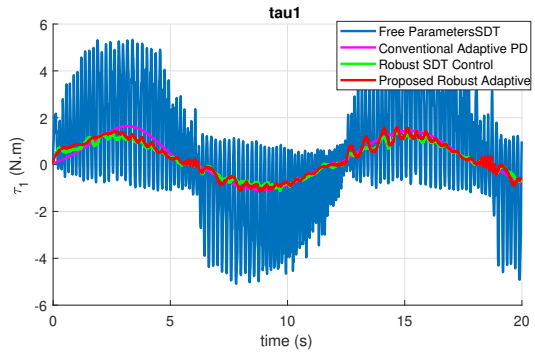
(b) Évolution de la première articulation.



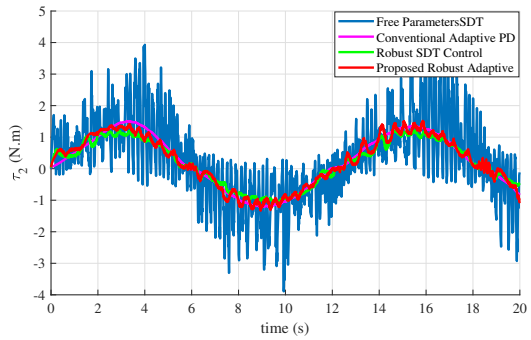
(c) Erreur de suivi dans la première articulation.



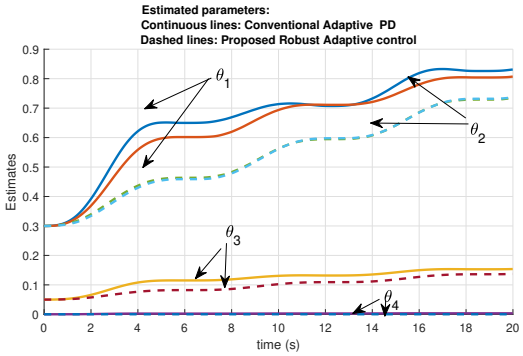
(d) Erreur de suivi dans la deuxième articulation.



(e) Couple appliqué dans la première articulation.



(f) Couple appliqué dans la deuxième articulation.



(g) Paramètres estimés.

Figure 5.3 – Évolution du robot avec une charge attachée et des trajectoires sinusoïdales.

Pour comparer les résultats de différents tests, le comportement d'erreur de suivi est également quantifié par les valeurs quadratiques moyennes (RMS) de la norme euclidienne du vecteur d'erreur de suivi défini comme suit :

$$e_N = \sqrt{e_1^2 + e_2^2}$$

Les valeurs du *RMS* sont résumés dans Fig.5.6.
Les résultats obtenus confirment les capacités de l'approche de contrôle adaptatif Twisting proposée par rapport à d'autres techniques. Tout d'abord, on peut noter clairement le niveau élevé de chattering qui caractérise les entrées de commande dérivée sous le Twisting algorithme sans modèle et ses effets sur la sortie du système. Deuxièmement, la réponse de l'algorithme SDT basé sur la linéarisation par bouclage statique est caractérisée par importantes erreurs de suivi par rapport au test précédent, principalement dans la deuxième articulation du robot. De plus, la méthode classique de contrôle adaptatif du PD est caractérisée par une dégradation de plus en plus importante de la précision de la poursuite. Les performances obtenues sous l'approche de contrôle proposée (qui combine le mode glissant avec la loi d'adaptation des paramètres) sont similaires à celles de l'essai précédent avec un signal de contrôle relativement lisse et sans affecter la stabilité du système.
Outre les résultats ci-dessus concernant le niveau élevé de précision, ce test prouve également la robustesse de l'approche proposée face aux incertitudes du modèle.

Test2 : Robustesse face aux variations des conditions initiales

Le but de ce test est de vérifier les performances de l'algorithme de contrôle proposé lorsque la tâche assignée au robot consiste à suivre des trajectoires polynomiales de la forme générale :

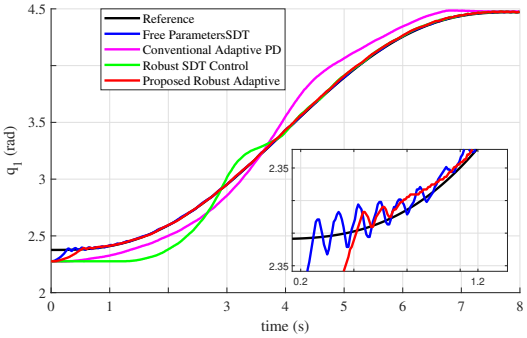
$$q_{di}(t) = a_{5i}t^5 + a_{4i}t^4 + a_{3i}t^3 + a_{2i}t^2 + a_{1i}t + a_{0i}$$

pour $i = \{1,2\}$ et $0 \leq t \leq t_f$. Les paramètres d'interpolation a_{ji} dans la fonction ci-dessus sont choisis pour assurer un mouvement repos à repos, un mouvement avec des vitesses et des accélérations nulles dans les points de départ et d'arrivé, pour le robot à partir de son point initial $q_i = [\frac{13\pi}{18} \ \frac{13\pi}{18}]^T$ à la position finale $q_f = [\frac{25\pi}{18} \ \frac{25\pi}{18}]^T$ dans un temps précis t_f en respectant les limites physiques du système (voir chapitre1). Pour cela on à fixé le temps d'exécution $t_f = 8s$. Afin d'explorer le comportement transitoire de l'approche proposée, cet ensemble d'expériences a été réalisé en considérant des incohérences dans les conditions initiales, c'est-à-dire en démarrant la tâche avec une erreur d'état initial par rapport à la trajectoire attribuée, où la configuration initiale du robot a été donnée $q_{ci} = q_i + [0.1 \ 0.1]^T$ ce qui conduit à un vecteur d'erreurs initiales $e_i = [0.1 \ 0.1]^T$. Les résultats obtenus en répétant les mêmes tests, avec et sans charge attachée, sont donnés respectivement par les Fig.5.4 et Fig.5.5. Comme on le voit, l'incohérence initiale entre la configuration du robot et celle souhaitée est rapidement amortie et renforcée par une réaction appropriée du couple τ, des signaux de commande lisses, en appliquant l'approche proposée. Malgré le convergence rapide, le comportement du Twisting algorithme sans modèle est caractérisé par des fréquences élevées et des signaux de commande non lisses, surtout dans le cas de présence d'une charge supplémentaire dans l'organe terminal du robot. Le twisting algorithme basé sur la platitude est caractérisé par un temps de convergence important par rapport à l'approche proposée. De plus, l'analyse du comportement

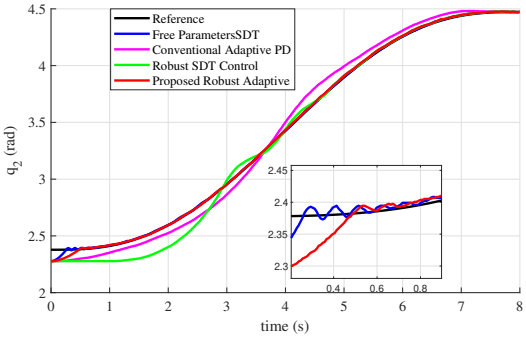
du robot en phase permanente, une fois que le robot a atteint les trajectoires souhaitées, confirme les résultats ci-dessus (de test1) concernant la précision et la robustesse de l'approche proposée.

5.5.3 Discussion des résultats

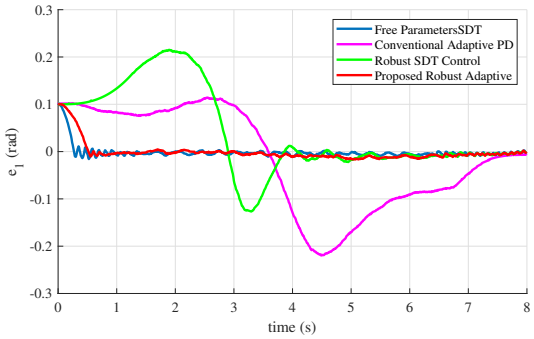
Tableau.5.1 et l'histogrammes dans Fig.5.6 résument les principaux résultats explorés à partir des tests ci-dessus pour évaluer les performances des approches de contrôle proposées en termes de précision de suivi, de régularité des signaux de contrôle et de robustesse du système face aux incertitudes paramétriques et non paramétriques. Les algorithmes basés sur la technique de commande par mode glissant préservent généralement des meilleures performances en termes de précision de suivi et de robustesse face aux incertitudes. Cependant, le phénomène de chattering limite les capacités du TA à un certain niveau d'incertitude. L'idée proposée pour l'atténuation du chattering, apparue sérieusement dans les algorithmes basés dans le cas de TA-sans modèle (5.42), consiste à réduire le niveau d'incertitudes en soutenant le contrôleur discontinu avec une loi de commande appropriée. Au départ, la loi de contrôle SDT basée sur la platitude donnée par (5.9,5.12), sur la base de certaines valeurs prédéfinies des paramètres du système, prouve leur capacité à traiter le problème des incertitudes non paramétriques. Cependant, cette approche ne résout pas le problème des incertitudes non paramétriques dues à la présence/l'absence d'une charge supplémentaire sur l'effecteur terminal du robot, ce qui représente l'une des tâches principales des manipulateurs. L'approche de contrôle combiné (adaptatif-Twisting) définie par (5.9,5.22 et 5.24) représente une solution efficace pour résoudre ce problème. Elle préserve un niveau de précision acceptable avec des signaux de commande relativement lisses et sans vibration dans la sortie du système, quelle que soit la tâche assignée au robot. Ce contrôleur est principalement composé d'une boucle interne avec un estimateur de paramètres en ligne approprié et d'une boucle externe avec une loi de contrôle à rétroaction discontinue. Les résultats obtenus ont prouvé le comportement complémentaire de chaque technique sur l'autre. Ici, on peut noter que des techniques intelligentes telles que la logique floue et les réseaux de neurones peuvent être utilisées pour remplacer la loi de mise à jour (5.24) afin de constituer un modèle de contrôleurs permettant de manipuler des bras de manipulateur plus complexes [Lewis1995], [Lewis1999].
De plus, étant donné que le contrôle par rétroaction proposé est basé sur une loi de commande discontinue, cette approche garantit une convergence en temps fini de l'erreur de suivi vers la surface de glissement spécifiée. Les résultats obtenus dans *Test2* supportent cette idée. Cette propriété des contrôleurs discontinus représente l'une des caractéristiques les plus importantes de la théorie du contrôle robuste et de nombreux auteurs se sont intéressés à ce domaine de recherche. Pour les modes glissants basés sur le TA, l'analyse de la propriété de convergence en temps fini et l'estimation du temps de convergence ont été effectuées d'abord par un raisonnement géométrique [Levant2007]. Les travaux suivants sur l'analyse de la convergence en temps fini des systèmes en mode glissant d'ordre élevé reposent principalement sur le principe d'homogénéité [Bacciotti1999] [Polyakov2009] [Oza2011] [Shtessel2007] [Orlov2005].



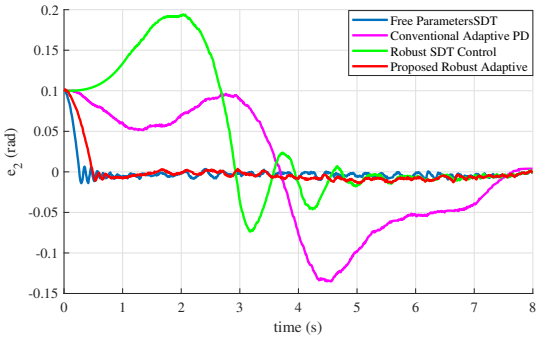
(a) Évolution de la première articulation.



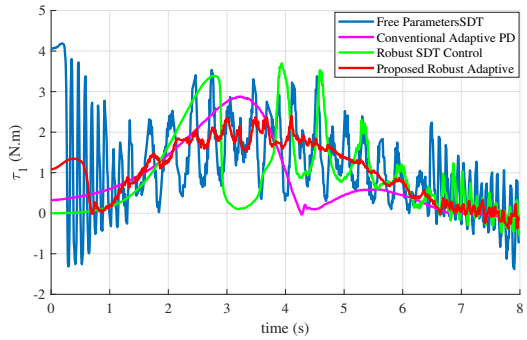
(b) Évolutions la deuxième articulation.



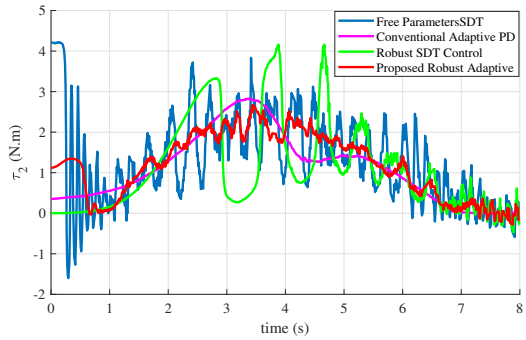
(c) Erreur de suivi dans la deuxième articulation.



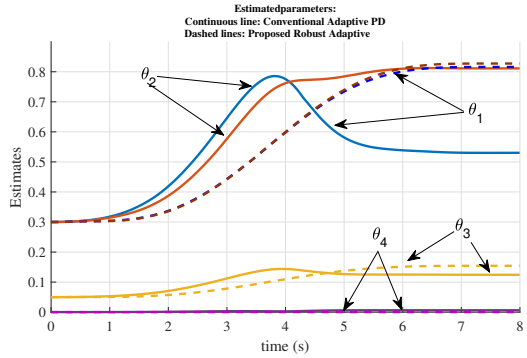
(d) Erreur de suivi dans la deuxième articulation.



(e) Couple appliqué dans la première articulation.

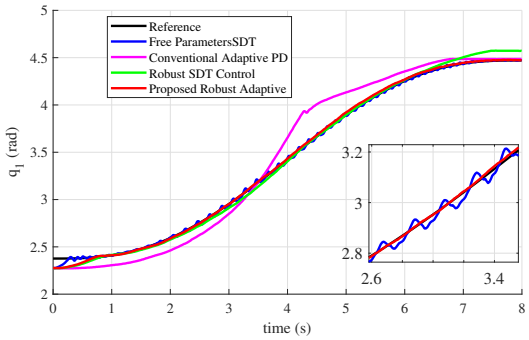


(f) Couple appliqué dans la deuxième articulation.

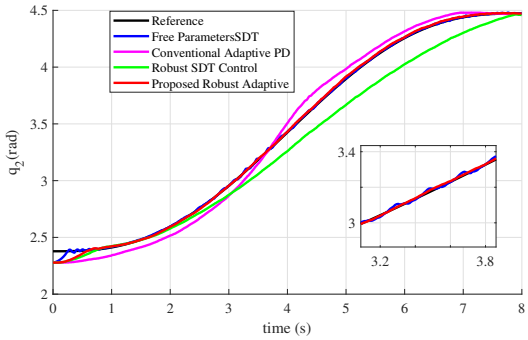


(g) Paramètres estimés.

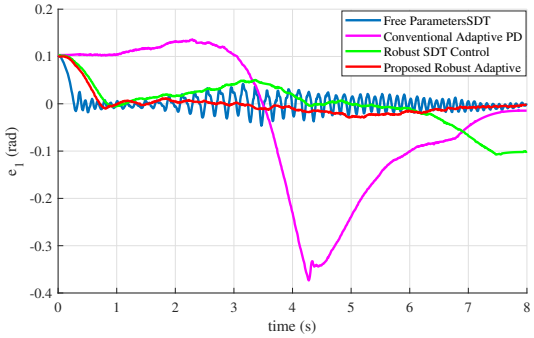
Figure 5.4 – Évolution de robot sans charge attachée avec des trajectoires polynomiales.



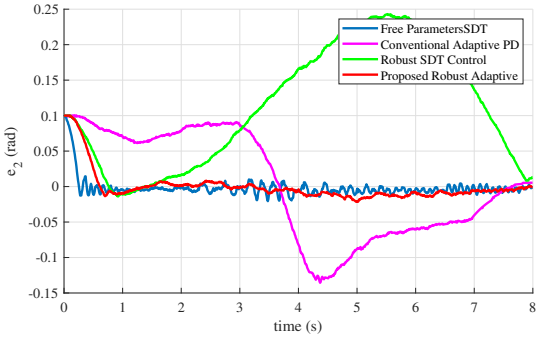
(a) Évolution de la première articulation.



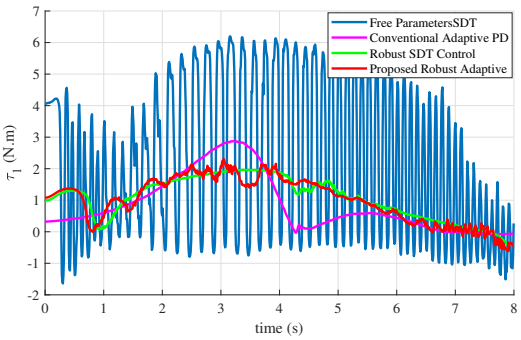
(b) Évolution la deuxième articulation.



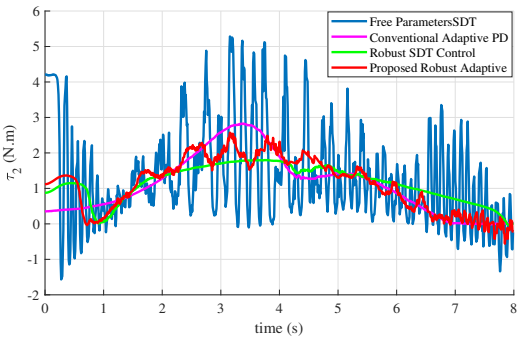
(c) Erreur de suivi dans la première articulation..



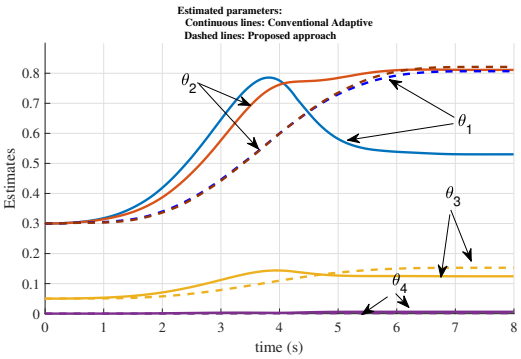
(d) Erreur de suivi dans la deuxième articulation.



(e) Couple dans la première articulation.



(f) Couple dans la deuxième articulation.



(g) Paramètres estimés.

Figure 5.5 – Évolution du robot avec une charge attachée le long d'une trajectoire polynomiale.

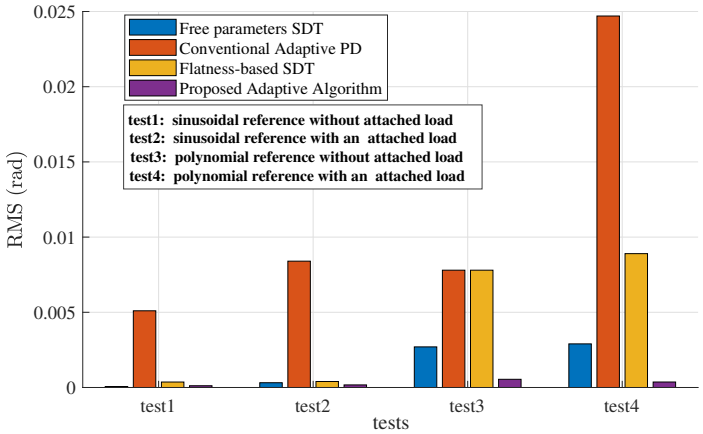


Figure 5.6 – Analyse d'erreur de suivi

Tableau 5.1 – Évaluation des performances des contrôleurs proposés

	Précision	**Chattering**	**Sensibilité**	**Convergence**
Twisting sans modèle	très élevé	très élevé	bas	très rapide
Platitude-TA	élevé	moyen	très élevé	lent
Adaptive PD	très bas	sans chattering	très élevé	très lent
Approche combinée (Adaptative-Twisting)	très élevé	bas	très bas	très rapide

5.6 Conclusion

Un schéma de contrôle adaptatif-robuste a été proposé dans ce chapitre pour assurer des bonnes performances de suivi pour les robots manipulateurs avec des paramètres dynamiques totalement inconnus. Les avantages de l'approche proposée s'expriment principalement par le niveau élevé de précision de suivi, la rapidité de convergence de l'erreur de suivi vers zéro et de la robustesse par rapport aux incertitudes paramétriques et non paramétriques. De plus, le soutien de l'algorithme de twisting avec une loi de contrôle adaptative permet de lisser les commandes discontinues et d'éviter les phénomènes indésirables de chattering quelle que soit la tâche assignée au robot. L'efficacité de l'approche proposée a été démontrée par une étude théorique en utilisant la théorie de Lyapunov. Des tests pratiques ont prouvés le comportement complémentaire de chaque technique sur l'autre.

Conclusion générale et perspectives

Le travail mené tout au long de cette thèse a permis d'apporter un certain nombre de solutions aux problèmes de commande et de planification des mouvements des systèmes robotisés. D'une manière générale, la commande de ce type de système est exprimée par une fonction qui gère un compromis entre les critères de rapidité, de précision et de sécurité. Les approches proposées dans ce mémoire permettent de répondre à ce compromis dans sa globalité.

Ce travail commence par des présentations générales pour les systèmes robotisés en vue de contourner la problématique d'autonomie de ces systèmes d'une part et d'élaborer les différentes techniques de planification et de commandes sous différentes contraintes et avec différentes considérations.

Une révision des avancements dans le domaine de planification sous les considérations géométriques, cinématiques et dynamiques est donnée dans le premier chapitre. Cette révision a permis d'identifier un certain nombre de problème de base en fonction de variation du système, de ces limites et de la tâche associée. Principalement, l'approche de planification avec des techniques découplées est introduite. Cette approche prend son utilité de sa nature qui permet d'incorporer différents types de contraintes aux différents niveaux de conception de la trajectoire. Essentiellement, la phase de planification du chemin permet d'inclure toutes les contraintes géométriques du problème et de considérer différents critères d'optimalité. La phase de génération des trajectoires a pour but d'optimiser le fonctionnement du robot en terme de différents critères de rapidité prenons en comptes la géométrie du chemin planifié ainsi que la dynamique et la cinématique du système d'autre part.

Le chapitre II a permis d'explorer les principes, les défis et les intérêts de différentes techniques de commande pour l'asservissement des systèmes non linéaires incertains. L'approche de linéarisation par bouclage est introduite comme un outil fondamental pour la commande des systèmes robotisés en vue de leur utilité non seulement dans la phase de suivi des trajectoires mais aussi dans la phase de génération d'une trajectoire le long d'un chemin prédéfini. Une exposition des principaux avancements et développements dans les techniques de linéarisation par bouclage statique/dynamique, les techniques de commande par mode glissant et les approches de commande adaptatives sont introduites.

Dans le chapitre III, nous considérons le problème de commande d'un robot mobile à roues. L'approche proposée est adaptée à la fois à l'optimisation du mouvement du système robotisé en terme de temps d'exécution, de complexité de calcul et à l'assurance de précision d'exécution des tâches planifiées :

— D'abord, un chemin qui évite les obstacles et minimise la courbure du trajet est défini.

— Ensuite, un algorithme qui assure l'optimisation du temps de parcours du chemin est proposé sous la considération des différentes limites physiques du robot ainsi que la minimisation du complexité de l'algorithme.

— Finalement, la notion de platitude est exploitée pour la conception d'une commande robuste pour résoudre le problème de suivi des trajectoires planifiées .

Les résultats de ce chapitre sont confirmés par des tests pratiques en utilisant le prototype de test « «PIONNER 3DX ».

Dans le chapitre IV on a fixé le problème de commande d'un portique à trois dimensions. Ce type de systèmes est distingué principalement par une structure flexible sous actionné, ce qui pose plusieurs problèmes reliés principalement à la stabilité, robustesse et précision d'exécution des tâches. L'approche découplée est étendue pour résoudre le problème de planification pour ce type de système. Un algorithme de commande par mode glissant d'ordre deux a été ensuite proposé pour gérer le compromis entre l'atténuation des indésirables oscillations de la charge et la précision de suivi des trajectoires tous en gardant la stabilité du système. La stabilité du système est analysée au sens de Lyapunov et confirmée par des résultats simulés sous l'environnement Matlab/Simulink.

Dans le chapitre V, on a concentré principalement sur le problème de suivi des trajectoires pour un bras manipulateur avec des paramètres inertiels totalement inconnues. La commande proposée a permis de couvrir une grande classe d'incertitude par le développement d'une technique hybride basée sur les concepts de linéarisation par bouclage, commande adaptative et le mode glissant d'ordre supérieur. Après une analyse théorique des performances du contrôleur proposé, les résultats de ce chapitre sont confirmé par des tests pratiques en utilisant le prototype de test (Cyton GAMMA 1500).

Comme perspectives, on envisage :

— D'étendre l'approche de planification proposée pour la manipulation des systèmes robotiques à structure plus compliquée tels que les robots parallèles et flexibles.

— L'estimation exacte du temps de convergence de la loi de commande proposée dans les chapitres IV, V.

—

Bibliographie

[Abdallah1991] C. Abdallah, D. Dawson, P. Dorato et M. Jamshidi. *Survey of Robust Contro for Rigid Robots*. IEEE Control Systems, pages 24–30, 1991.

[Antonelli2003] Gianluca Antonelli, Stefano Chiaverini et Giuseppe Fusco. *A new on-line algorithm for inverse kinematics of robot manipulators ensuring path tracking capability under joint limits*. IEEE Transactions on Robotics and Automation, vol. 19, no. 1, pages 162–167, feb 2003.

[Antonelli2009] Gianluca Antonelli, Stefano Chiaverini, Cataldo Curatella et Alessandro Marino. *Constrained motion planning for industrial robots*. In 2009 IEEE International Conference on Automation and Logistics, numéro August, pages 1934–1939. IEEE, aug 2009.

[Astrom1994] Karl Johan Astrom et Bjorn Wittenmark. Adaptive control. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1994.

[Astrom2006] Karl J. Astrom et Tore HÄgglund. Advanced PID Control. 2006.

[Aström2008] Karl Johan Aström et Richard M. Murray. Feedback Systems : An Introduction for Scientists and Engineers . PRINCETON UNIVERSITY PRESS PRINCETON AND OXFORD, 41 William Street, Princeton, New Jersey 08540, princeton edition, dec 2008.

[Åström2014] Karl Åström. *History of Adaptive Control*. In John Baillieul et Tariq Samad, editeurs, Encyclopedia of Systems and Control, numéro Bellman 1961, pages 1–9. Springer London, London, 2014.

[Bacciotti1999] Andrea Bacciotti et Francesca Ceragioli. *Stability and Stabilization of Discontinuous Systems*. ESAIM : Control, Optimisation and Calculus of Variations, vol. 4, no. June, pages 361–376, 1999.

[Balkan1998] Tuna Balkan. *A dynamic programming approach to optimal control of robotic manipulators*. Mechanics Research Communications, vol. 25, no. 2, pages 225–230, mar 1998.

[Barsky1983] Brian A. Barsky et John C. Beatty. *Local control of bias and tension in beta-splines*. Proceedings of the 10th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1983, vol. 2, no. 2, pages 193–218, 1983.

[Bartels1987] Richard H. Bartels, John C. Beatty et Brian A. Barsky. An Introduction to Splines for use in Computer Graphics and Geometric Modeling. Numeéro 541. Morgan Kaufmann Publishers, Inc., United States of America, 1987.

[Bartolini2003] Giorgio Bartolini, Alessandro Pisano et Elio Usai. *Output-feedback Control of Container Cranes : A Comparative Analysis*. Asian Journal of Control, vol. 5, no. 4, pages 578–593, 2003.

[Béarée2007] Richard Béarée et Pierre-Jean Barre. *Adaptation de la loi de mouvement aux systèmes de positionnement à dynamique élevée*. Mécanique & Industries, vol. 8, no. 5, pages 407–417, sep 2007.

[Béarée2013] Richard Béarée et Adel Olabi. *Dissociated jerk-limited trajectory applied to time-varying vibration reduction*. Robotics and Computer-Integrated Manufacturing, vol. 29, no. 2, pages 444–453, 2013.

[Béarée2014] Richard Béarée. *New Damped-Jerk trajectory for vibration reduction*. Control Engineering Practice, vol. 28, no. 1, pages 112–120, 2014.

[Berglund2008] Tomas Berglund, Andrej Brodnik, Håkan Jonsson, Kent Mrozek, M. Staffanson et Inge Söderkvist. *Planning Smooth Paths among Obstacles using Minimum Curvature Variation B-splines*. 2008.

[Bertolazzi2012] Enrico Bertolazzi et Marco Frego. *Fast and Accurate Clothoid Fittin*. ACM Trans. Math. Softw., vol. V, 2012.

[Biagiotti2008] Luigi Biagiotti et Claudio Melchiorri. Trajectory Planning for Automatic Machines and Robots. Springer-Verlag, Berlin Heidelberg, 2008.

[Biagiotti2012] Luigi Biagiotti et Claudio Melchiorri. *FIR filters for online trajectory planning with time- and frequency-domain specifications*. Control Engineering Practice, vol. 20, no. 12, pages 1385–1399, 2012.

[Blajer2007] W. Blajer et K.Kołodziejczyk. *Motion planning and control of gantry cranes in cluttered work environment*. IET Control Theory Appl., vol. 1, no. 5, pages 1370–1379, 2007.

[Bobrow1985] J.E. Bobrow, S. Dubowsky et J.S. Gibson. *Time-Optimal Control of Robotic Manipulators Along Specified Paths*. The International Journal of Robotics Research, vol. 4, no. 3, pages 3–17, sep 1985.

[Boissonnat1994] J.-D. Boissonnat, A. Cérézo et J. Leblond. *A Note on Shortest Paths in the Plane Subject to a Constraint on the Derivative of the Curvature*. Rapport technique INRIA-2160, INRIA, Sophia-Antipolis, SOPHIA-ANTIPOLIS Cedex (France), 1994.

[Bonnans2005] F. Bonnans et P. Rouchon. Commande et optimisation de systèmes dynamiques. Mathématiques appliquées. Éditions de l'École Polytechnique, 2005.

[Boubezoula2018] Mabrouk Boubezoula, Abdelouhab Hassam et Oussama Boutalbi. *Robust-flatness Controller Design for a Differentially Driven Wheeled Mobile Robot Mabrouk Boubezoula* . International Journal of Control, Automation and Systems, vol. 16, no. 4, pages 1895–1904, 2018.

[Boutalbi2015] Oussama Boutalbi. *Commande par mode glissant du second ordre d ' un portique de trois dimensions Dédicaces*. Thèse de Doctorat,

UNIVERSITE FERHAT ABBAS – SETIF-1- UFAS (ALGERIE), 2015.

[Boutalbi2017] O. Boutalbi, K. Benmahammed et M. Boubezoula. *Robust feedback linearizing controller design for an overhead crane system*. In 2017 6th International Conference on Systems and Control (ICSC), pages 640–645, May 2017.

[Boutalbi2019] Oussama Boutalbi, Khier Benmahammed, Khadidja Henni et Boualem Boukezata. *A high-performance control algorithm based on a curvature-dependent decoupled planning approach and flatness concepts for non-holonomic mobile robots*. Intelligent Service Robotics, vol. 12, no. 2, pages 181–196, apr 2019.

[Braghin2008] F. Braghin, F. Cheli, S. Melzi et E. Sabbioni. *Race driver model*. Computers and Structures, vol. 86, no. 13-14, pages 1503–1516, 2008.

[Brockett1978] R.W. Brockett. *Feedback Invariants for Nonlinear Systems**. IFAC Proceedings Volumes, vol. 11, no. 1, pages 1115 – 1120, 1978. 7th Triennial World Congress of the IFAC on A Link Between Science and Applications of Automatic Control, Helsinki, Finland, 12-16 June.

[Brogliato1991] Bernard Brogliato et Ioan-dore Landau. *Adaptive Motion Control of Robot Manipulators : A Unified Approach Based on Passivity*. International Journal of Robust and Adaptive Control Journal, vol. 1, pages 187–202, 1991.

[Byrnes1986] Christopher I. Byrnes et Anders Lindquist, editeurs. Theory and applications of nonlinear control systems. Elsevier North-Holland, Inc., New York, NY, USA, 1986.

[Campion2008] Guy Campion et Woojin Chung. *Wheeled Robots*. In Springer Handbook of Robotics, pages 391–410. Springer-Verlag Berlin Heidelberg, 2008.

[Cartmell2003] M P Cartmell. *On the Need for Control of Nonlinear Oscillations in Machine Systems*. Meccanica, vol. 38, pages 185–212, 2003.

[Chan2008] Nicholas Chan, James Kuffner et Matthew Zucker. *Improved Motion Planning Speed and Safety using Regions of Inevitable Collision*. In 17th CISM-IFToMM Symposium on Robot Design, Dynamics, and Control (RoManSy'08), July 2008.

[Charlet1989] B. Charlet, J. Lévine et R. Marino. *On dynamic feedback linearization*. Systems & Control Letters, vol. 13, no. 2, pages 143–151, aug 1989.

[Chen2017] Yong Chen, Yiyu Cai, Jianmin Zheng et Daniel Thalmann. *Accurate and Efficient Approximation of Clothoids Using Bézier Curves for Path Planning*. IEEE Transactions on Robotics, vol. 33, no. 5, pages 1242–1247, 2017.

[Cheng1988] D Cheng, A. Isidori, W. Respondek et T. J. Tarn. *Exact linearization of nonlinear systems with outputs*. Mathematical Systems Theory, vol. 21, no. 1, pages 63–83, dec 1988.

[Chevallereau2003] C. Chevallereau. *Time-scaling control for an underactuated biped robot*. IEEE Transactions on Robotics and Automation, vol. 19, no. 2, pages 362–368, apr 2003.

[Chiaverini1999] Stefano Chiaverini et Giuseppe Fusco. *A new inverse kinematics algorithm with path tracking capability under velocity and acceleration constraints*. In Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304), volume 3, pages 2064–2069. IEEE, 1999.

[Cole2011] Matthew O T Cole. *A discrete-time approach to impulse-based adaptive input shaping for motion control without residual vibration*. Automatica, vol. 47, no. 11, pages 2504–2510, 2011.

[Collado2013] Joaquin Collado, Leonid Fridman et V Carlos. *Super Twisting Control of a Parametrically Excited Overhead Crane $*. Journal of the Franklin Institute, page 9, 2013.

[Dahl1990] O. Dahl et L. Nielsen. *Torque-limited path following by online trajectory time scaling*. IEEE Transactions on Robotics and Automation, vol. 6, no. 5, pages 554–561, 1990.

[Dai2015] Zhiwu Dai, Xingjun Sheng, Jian Hu, Hui Wang et Dingguo Zhang. *Design and Implementation of Bézier Curve Trajectory Planning in DELTA Parallel Robots*. In International Conference on Intelligent Robotics and Applications, volume 9245, pages 420–430, 2015.

[D'Andréa-Novel1990] B. D'Andréa-Novel et J. Lévine. *Modelling and Nonlinear Control of an Overhead Crane*. In Robust Control of Linear Systems and Nonlinear Control, pages 523–529. Birkhäuser Boston, Boston, MA, 1990.

[de Boor] Carl de Boor. A Practical Guide to Splines. Springer-Verlag, New York, revised ed edition.

[De Luca2002] Alessandro De Luca et Riccardo Farina. *Dynamic scaling of trajectories for robots with elastic joints*. In Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), volume 3, pages 2436–2442. IEEE, 2002.

[de Wit1996] Carlos Canudas de Wit, Bruno Siciliano et Georges Bastin. Theory of Robot Control. Springer-Verlag London, 1 st edtio edition, 1996.

[Delingette1991] H. Delingette, M. Hebert et K. Ikeuchi. *Trajectory Generation with Curvature Constraint Based on Energy Minimization*. In IEEE/RSJ International Workshop on Intelligent Robots and Systems lROS '91. Nov. 3-5, 1991, Osaka, Japan. IEEE Cat. No. 91TH0375-6, numéro 91, pages 206–211, 1991.

[Dietz2004] Donna A. Dietz et Bruce Piper. *Interpolation with Cubic Spirals*. Computer Aided Geometric Design, vol. 21, no. 2, pages 165–180, 2004.

[Dietz2008] Donna A. Dietz, Bruce Piperb et Elena Sebe. *Rational Cubic Spirals*. Computer-Aided Design, vol. 12, no. 1, pages 145–157, 2008.

[Dubins1957] Lester Eli Dubins. *On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal*

Positions and Tangents. The American Journal of Mathematics, vol. 79, no. 3, pages 497–516, 1957.

[Dubins1961] Lester Eli Dubins. *On Plane Curves With Curvature*. Pacific Journal of Mathematics, vol. 11, no. 2, pages 470–484, 1961.

[Emel'yanov1993] Stanislav Vasil'evich Emel'yanov, Sergei Konstantinovich Korovin et A Levant. *Sliding modes of higher orders in control systems*. Differentsial'nye Uravneniya, vol. 29, no. 11, pages 1877–1899, 1993.

[Emeryanov1996] S V Emeryanov et S K Korovin. *High-Order Sliding Modes in Control Systems*. Computational Mathematics and Modelefing,, vol. 7, no. 3, pages 294–318, 1996.

[Ezair2014] Ben Ezair, Tamir Tassa et Zvi Shiller. *Planning high order trajectories with general initial and final conditions and asymmetric bounds*. International Journal of Robotics Research, vol. 33, no. 6, pages 898–916, 2014.

[Farin1997] Gerald Farin. Curves and Surfaces for Computer-Aided Geometric Design A Practical Guide. Academic Press, Boylston Street, Chestnut Hill, MA 02167, USA, fourth edi edition, 1997.

[Farin2002] Gerald Farin. Curves and Surfaces for CAGD A Practical Guide. MORGAN KAUFMANN PUBLISHERS, United States of America, fifth edit edition, 2002.

[Farouki1990] R. T. Farouki et T Sakkalis. *Pythagorean Hodographs*. IBM J. RES. DEVEL, vol. 34, no. 5, pages 736–752, 1990.

[Farouki2008] Rida T. Farouki. Pythagorean -Hodograph Curves : Algebra and Geometry Inseparable. Springer-Verlag, Berlin Heidelberg, first edit edition, 2008.

[Farouki2012] R. T. Farouki et C A Neff. *Hermite Interpolation by Pythagorean Hodograph Quintics*. Mathematics of Computation, vol. 64, no. 212, pages 1589–1609, 2012.

[Farouki2014] Rida T Farouki, Carla Manni et Maria Lucia Sampoli. *Shape-preserving interpolation of spatial data by Pythagorean-hodograph quintic spline curves*. IMA Journal ofNumerical Analysis, pages 1–21, 2014.

[Farouki2017] Rida T Farouki, Carlotta Giannelli, Duccio Mugnaini et Alessandra Sestini. *Path planning with Pythagorean- hodograph curves for unmanned or autonomous vehicles*. J Aerospace Engineering, vol. 0, no. 0, pages 1–12, 2017.

[Fless2009] Michel Fless et Cédric JOIN. *MODEL-FREE CONTROL AND INTELLIGENT PID CONTROLLERS : TOWARDS A POSSIBLE TRIVIALIZATION OF NONLINEAR CONTROL ?* IFAC Proceedings Volumes, vol. 42, no. 10, pages 1531–1550, 2009.

[Fliess1993] M. Fliess, J. Levine, P. Martin et P. Rouchon. *On Differentially Flat Nonlinear Systems*. In IFAC Symposia Series, numéro 7, pages 159–163, 1993.

[Fliess1995] Michel Fliess, Jean Lévine, Philippe Martin et Pierre Rouchon. *Flatness and defect of non-linear systems : introductory theory and examples*. International Journal of Control, vol. 61, no. 6, pages 1327–1361, 1995.

[Fliess2013] Michel Fliess et Cédric Join. *Model-free control*. International Journal of Control, vol. 86, no. 12, pages 2228–2252, 2013.

[Floater2008] Michael S Floater. *On the deviation of a parametric cubic spline interpolant from its data polygon*. Computer Aided Geometric Design, vol. 25, no. 0167, pages 148–156, 2008.

[Fraichard2004] Thierry Fraichard et Alexis Scheuer. *From Reeds and Shepp's to Continuous-Curvature Paths*. IEEE Transactions on Robotics, vol. 20, no. 6, pages 144–149, 2004.

[Fritscht1980] F N Fritscht et R E Carlson. *Monotone Piecewise Cubic Interpolation*. SIAM Journal on Numerical Analysis, vol. 17, no. 2, pages 238–246, 1980.

[Gasparetto2012] Alessandro Gasparetto, Paolo Boscariol, Albano Lanzutti et Renato Vidoni. *Trajectory Planning in Robotics*. Mathematics in Computer Science, vol. 6, no. 3, pages 269–279, 2012.

[Gašper Jaklic2011] Gašper Jaklic et Emil Zˇagar. *Curvature variation minimizing cubic Hermite interpolants*. Applied Mathematics and Computation, vol. 218, pages 3918–3924, 2011.

[Geraerts2007] Roland Geraerts et Mark H. Overmars. *Creating High-quality Paths for Motion Planning*. International Journal of Robotics Research, vol. 26, no. 8, pages 845–863, 2007.

[Gerelli2009] Oscar Gerelli et Guarino Corrado Lo Bianco. *Nonlinear Variable Structure Filter for the Online Trajectory Scaling*. IEEE Transactions on Industrial Electronics, vol. 56, no. 10, pages 3921–3930, oct 2009.

[Ghilardelli2014] Fabio Ghilardelli, Gabriele Lini et Aurelio Piazzi. *Path Generation Using 4$-Splines for a Truck and Trailer Vehicle*. IEEE Transactions on Automation Science and Engineering, vol. 11, no. 1, pages 187–203, jan 2014.

[Giannelli2016] Carlotta Giannelli, Duccio Mugnaini et Alessandra Sestini. *Path Planning with Obstacle Avoidance by G 1 Ph Quintic Splines*. Computer-Aided Design, vol. 75-76, pages 47–60, 2016.

[Gonzalez2016] David Gonzalez, Joshué Pérez, Vicente Milanés et Fawzi Nashashibi. *A Review of Motion Planning Techniques for Automated Vehicles*. Ieee Transactions on Intelligent Transportation Systems, vol. 17, no. 4, pages 1135–1145, 2016.

[Grimm1990] Wolfgang M Grimm. *Robot non-linearity bounds evaluation techniques for robust control*. International Journal Of Adaptive Control And Signal Processign, vol. 4, no. June, pages 501–522, 1990.

[Guenter1990] Brian Guenter et Richard Parent. *Computing the arc length of parametric curves*. IEEE Computer Graphics and Applications, vol. 10, no. 3, pages 72–78, may 1990.

[Habib2007] Zulfiqar Habib et Manabu Sakai. *G 2 Pythagorean Hodograph Quintic Transition between Two Circles with Shape Control*. Computer Aided Geometric Design, vol. 24, pages 252–266, 2007.

[Hagenmeyer2003] Veit Hagenmeyer et Emmanuel Delaleau. *Robustness analysis of exact feedforward linearization based on di fferential atness*. Automatica, vol. 39, pages 1941–1946, 2003.

[Hausler2016] Andreas J. Hausler, Alessandro Saccon, Antonio Pedro Aguiar, John Hauser et Antonio M. Pascoal. *Energy-Optimal Motion Planning for Multiple Robotic Vehicles With Collision Avoidance*. IEEE Transactions on Control Systems Technology, vol. 24, no. 3, pages 867–883, may 2016.

[Hermes1969] Henry Hermes et Joseph P. Lassalle. Functional analysis and time optimal control. ACADEMIC PRESS New York, New York, 1969.

[Hollerbach1984] J. M. Hollerbach. *Dynamic Scaling of Manipulator Trajectories*. Journal of Dynamic Systems, Measurement, and Control, vol. 106, no. 1, pages 102–106, mar 1984.

[IFR] *International Federation of Robotics*.

[Ioannou2006] Petros Ioannou et Fidan Baris. Adaptive control tutorial (advances in design and control). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[Isidori1986a] A. Isidori. *Control of Nonlinear Systems Via Dynamic State-Feedback*. In Algebraic and Geometric Methods in Nonlinear Control Theory, numéro 2, pages 121–145. Springer Netherlands, Dordrecht, 1986.

[Isidori1986b] A. Isidori, C. Moog et A. Luca. *A sufficient condition for full linearization via dynamic state feedback*. In 1986 25th IEEE Conference on Decision and Control, numéro December, pages 203–208. IEEE, dec 1986.

[Isidori1995] Alberto Isidori. Nonlinear Control Systems. Communications and Control Engineering. Springer London, London, 1995.

[JAKLIC2010] GASPER JAKLIC, JERNEJ KOZAK, MARJETA KRAJNC, VITO VITRIH et EMIL ZAGAR. *Hermite Geometric Interpolation by Rational Bezier Spatial Curves*. Society, vol. 20, no. 6, pages 2994–3013, 2010.

[Jeon2000] Jae Wook Jeon et Young Youl Ha. *A generalized approach for the acceleration and deceleration of industrial robots and CNC machine tools*. IEEE Transactions on Industrial Electronics, vol. 47, no. 1, pages 133–139, 2000.

[Kahn1971] M. E. Kahn et B. Roth. *The Near-minimum-time Control of Open-loop Articulated Kinematic Chains*. Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, vol. 93, no. 3, pages 164–172, 1971.

[Kanayama1989] Yutaka Kanayama et Bruce I B.I. Hartman. *Smooth Local Path Planning for Autonomous Vehicles*. In Proceedings, 1989 International Conference on Robotics and Automation, pages 1265–1270, 1989.

[Kang Shin1985] Kang Shin et N. McKay. *Minimum-time control of robotic manipulators with geometric path constraints*. IEEE Transactions on Automatic Control, vol. 30, no. 6, pages 531–541, jun 1985.

[Kant1986] Kamal Kant et Steven W Zucker. *Toward Efficient Trajectory Planning : The Path-Velocity Decomposition*. The International Journal of Robotics Research, vol. 5, no. 3, pages 72–89, sep 1986.

[Kavraki1998] L.E. Kavraki, M.N. Kolountzakis et J.-C. Latombe. *Analysis of probabilistic roadmaps for path planning*. IEEE Transactions on Robotics and Automation, vol. 14, no. 1, pages 166–171, feb 1998.

[Kavraki2008] Lydia E. Kavraki et Steven M. LaValle. Motion planning, pages 109–131. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[Khalil2000] Wissama Khalil et E Dombre. Modeling Identification and Control of Robots. Butterworth-Heinemann, butterwort edition, 2000.

[Khalil2002] Hassan K Khalil. Nonlinear systems ; 3rd ed. Prentice-Hall, Upper Saddle River, NJ, 2002.

[Khatib1986] Oussama Khatib. *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots*. The International Journal of Robotics Research, vol. 5, no. 1, pages 90–98, mar 1986.

[Kim1994] Dong Il Kim, Jae Wook Jeon et Sungkwun Kim. *Software acceleration/deceleration methods for industrial robots and CNC machine tools*. Mechatronics, vol. 4, no. 1, pages 37–53, 1994.

[Kim2007] Doang Nguyen Kim, I. Ming Chen et Teck Chew Ng. *Planning Algorithms for S-curve Trajectories*. In IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, volume 1, 2007.

[Kim2017] Yunjeong Kim et Byung Kook Kim. *Time-Optimal Trajectory Planning Based on Dynamics for Differential-Wheeled Mobile Robots With a Geometric Corridor*. IEEE Transactions on Industrial Electronics, vol. 64, no. 7, pages 5502–5512, 2017.

[Kingston2018] Zachary Kingston, Mark Moll et Lydia E. Kavraki. *Sampling-Based Methods for Motion Planning with Constraints*. Annual Review of Control, Robotics, and Autonomous Systems, vol. 1, no. 1, pages 159–185, 2018.

[Kito2003] Tomomi Kito, Jun Ota, Rie Katsuki, Takahisa Mizuta, Tamio Arai, Tsuyoshi Ueyama et Tsuyoshi Nishiyama. *Smooth Path Planning by Using Visibility Graph-like Method*. In Proceedings - IEEE International Conference on Robotics and Automation, volume 3, pages 3770–3775, Taipei, Taiwan, 2003. IEEE.

[K.Khalil2002] Hassan K.Khalil. Nonlinear Systems. Prentice Hall, 3rd editio edition, 2002.

[Klančar2017] G. Klančar, A. Zdešar, S. Blažič et I. Škrjanc. 'Wheeled Mobile Robotics : From Fundamentals Towards Autonomous Systems'. Butterworth-Heinemann, 2017.

[Krener1983] Arthur Krener, Alberto Isidori et Witold Respondek. *Partial and robust linearization by feedback*. In The 22nd IEEE Conference on Decision and Control, volume 1, pages 126–130. IEEE, 1983.

[Krener2014] A J Krener. *Feedback Linearization of Nonlinear Systems*. In John Baillieul et Tariq Samad, editeurs, Encyclopedia of Systems and Control, numéro 4, pages 1–14. Springer London, London, 2014.

[Kröger2006] Torsten Kröger, Adam Tomiczek et Friedrich M. Wahl. *Towards on-line trajectory computation*. IEEE International Conference on Intelligent Robots and Systems, pages 736–741, 2006.

[Kröger2010] Torsten Kröger. *On-Line Trajectory Generation in Robotic Systems : Basic Concepts for Instantaneous Reactions to Unforeseen (Sensor) Events*. Springer Tracts in Advanced Robotics, vol. 58, no. 1, pages 1–230, 2010.

[Lambrechts2005] Paul Lambrechts, Matthijs Boerlage et Maarten Steinbuch. *Trajectory Planning and Feedforward Design for Electromechanical Motion Systems*. Control Engineering Practice, vol. 13, no. 2, pages 145–157, 2005.

[Lange2014] Friedrich Lange et Michael Suppa. *Predictive path-accurate scaling of a sensor-based defined trajectory*. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 754–759. IEEE, may 2014.

[Lange2016] Friedrich Lange et Alin Albu-Schaffer. *Path-Accurate Online Trajectory Generation for Jerk-Limited Industrial Robots*. IEEE Robotics and Automation Letters, vol. 1, no. 1, pages 82–89, jan 2016.

[LaValle2006] Steven M. LaValle. Planning algorithms, volume 9780521862. Cambridge University Press, 2006.

[Lee1998] Ho-hoon Lee. *Modeling and Control of a Three-Dimensional Overhead*. Journal of Dynamic Systems, Measurement, and Control, ASME, vol. 120, no. December, pages 471–476, 1998.

[Levant1987] A Levant. *Higher order sliding modes and their application for controlling uncertain processes*. Institute for System Studiesof the USSR Acadamy of Science, Moskau, Diss, 1987.

[Levant1998] Arie Levant. *Robust Exact Differentiation via Sliding Mode Technique*. Automatica, vol. 34, no. 3, pages 379–384, 1998.

[Levant2003] Arie Levant. *Higher-order sliding modes, differentiation and output-feedback control*. International Journal of Control, vol. 76, no. 9-10, pages 924–941, jan 2003.

[Levant2005] Arie Levant. *Homogeneity approach to high-order sliding mode design*. Automatica, vol. 41, no. 5, pages 823–830, may 2005.

[Levant2007] ARIE Levant. *Sliding order and sliding accuracy in sliding mode control*. International Journal of Control, vol. 58, N°.6, pages 37–41, 2007.

[Levant2011] Arie Levant, Mohammed Taleb et Franck Plestan. *Twisting-controller Gain Adaptation*. In 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC) Orlando, FL, USA, volume 1, pages 7015–7020, 2011.

[Levant2016] Arie Levant, Denis Efimov, Andrey Polyakov et Wilfrid Perruquetti. *Stability and robustness of homogeneous differential inclusions*. In 2016 IEEE 55th Conference on Decision and Control (CDC), numéro Cdc, pages 7288–7293. IEEE, dec 2016.

[Lewis1985] J. H. Lewis. *The Kronecker indices of affine-nonlinear control systems*. In 1985 24th IEEE Conference on Decision and Control, numéro December, pages 371–374. IEEE, dec 1985.

[Lewis1995] Frank L Lewis, Kai Liu et Aydin Yesildirek. *Neural Net Robot Controller with Guaranteed Tracking Performance*. IEEE transaction On Neural Networks, vol. 6, no. 3, pages 703–715, 1995.

[Lewis1999] F L Lewis. *Nonlinear Network Structures for Feedback Control*. Asian Journal of Control,, vol. 1, no. 4, pages 205–228, 1999.

[Lewis2002] F. Lewis, J. Campos et R. Selmic. Neuro-fuzzy control of industrial systems with actuator nonlinearities. Society for Industrial and Applied Mathematics, 2002.

[Lipp2014] Thomas Lipp et Stephen Boyd. *Minimum-time speed optimisation over a fixed path*. International Journal of Control, vol. 87, no. 6, pages 1297–1311, jun 2014.

[Liu2000] C Liu et J. J. E. Slotine. *Adaptive Tracking Control for Robots with Unknown Kinematic and Dynamic Properties*. The International Journal of Robotics Research, vol. Vol. 25, N, no. 639798, pages 283–296, 2000.

[Liu2002] G. H. Liu, Y. S. Wong, Y. F. Zhang et H. T. Loh. *Adaptive fairing of digitized point data with discrete curvature*. Computer-Aided Design, vol. 34, no. 4, pages 309–320, April 2002.

[Liu2012] D T Liu et W P Guo. *Tracking Control for an Underactuated Two-Dimensional Overhead Crane*. Journal of Applied Research and Technology, vol. 10, no. 4, pages 597–606, 2012.

[Livne2014] Miki Livne et Arie Levant. *Accuracy of disturbed homogeneous sliding modes*. In 2014 13th International Workshop on Variable Structure Systems (VSS), numéro 5, pages 1–6. IEEE, jun 2014.

[Lo Bianco2004a] Guarino Corrado Lo Bianco, Aurelio Piazzi et Massimo Romano. *Smooth Motion Generation for Unicycle Mobile Robots Via Dynamic Path Inversion*. IEEE Transactions on Robotics, vol. 20, no. 5, pages 884–891, 2004.

[Lo Bianco2004b] Guarino Corrado Lo Bianco, Aurelio Piazzi et Massimo Romano. *Smooth motion generation for unicycle mobile robots via dynamic path inversion*. IEEE Transactions on Robotics, vol. 20, no. 5, pages 884–891, 2004.

[Lo Bianco2004c] Guarino Corrado Lo Bianco, Aurelio Piazzi et Massimo Romano. *Smooth motion generation for unicycle mobile robots via dynamic path inversion*. IEEE Transactions on Robotics, vol. 20, no. 5, pages 884–891, 2004.

[Lo Bianco2005] Guarino Corrado Lo Bianco et Massimo Romano. *Bounded velocity planning for autonomous vehicles*. 2005 IEEE/RSJ International

Conference on Intelligent Robots and Systems, IROS, no. 2, pages 4068–4073, 2005.

[Lo Bianco2006] Guarino Corrado Lo Bianco. Optimal velocity planning for autonomous vehicles under kinematic constraints, volume 8. IFAC, 2006.

[Lo Bianco2010] Guarino Lo Bianco Corrado et Oscar Gerelli. *Generation of Paths with Minimum Curvature Derivative with ? ?3-splines*. IEEE Transactions on Automation Science and Engineering, vol. 7, no. 2, pages 249–256, 2010.

[Lo Bianco2011] Guarino Corrado Lo Bianco et Oscar Gerelli. *Online Trajectory Scaling for Manipulators Subject to High-Order Kinematic and Dynamic Constraints*. IEEE Transactions on Robotics, vol. 27, no. 6, pages 1144–1152, dec 2011.

[Lo Bianco2013] Guarino Corrado Lo Bianco. *Minimum-jerk velocity planning for mobile robot applications*. IEEE Transactions on Robotics, vol. 29, no. 5, pages 1317–1326, 2013.

[Lo Bianco2017] Guarino Corrado Lo Bianco et Fabio Ghilardelli. *A scaling algorithm for the generation of jerk-limited trajectories in the operational space*. Robotics and Computer-Integrated Manufacturing, vol. 44, no. June 2016, pages 284–295, apr 2017.

[Lu2015] Lizheng Lu. *A Note on Curvature Variation Minimizing Cubic Hermite Interpolants*. Applied Mathematics and Computation, vol. 259, pages 596–599, 2015.

[Lu2018] Lizheng Lu, Chengkai Jiang et Qianqian Hu. *Planar cubic G1 and quintic G2 Hermite interpolations via curvature variation minimization*. Computers and Graphics (Pergamon), vol. 70, pages 92–98, 2018.

[Luviano-Juárez2014] Alberto Luviano-Juárez, John Cortés-Romero et Hebertt Sira-Ramírez. *Trajectory Tracking Control of a Mobile Robot Through a Flatness-Based Exact Feedforward Linearization Scheme*. Journal of Dynamic Systems, Measurement, and Control, vol. 137, no. 5, page 051001, 2014.

[M.Abdel-Rahman2003] Eihab M.Abdel-Rahman, Ali H.Nayfeh. et Masoud N.Ziyad. *Dynamics and Control of Cranes : A Review*. Journal of Vibration and Control, vol. 9, pages 863–908, 2003.

[Marino1986] Riccardo Marino. *On the largest feedback linearizable subsystem*. Systems & Control Letters, vol. 6, no. 5, pages 345–351, jan 1986.

[Martinl1994] Philippe Martinl et Pierre Rouchon. *Mathematics of Control , Signals , and Systems Feedback Linearization and Driftless Systems* . Mathematics of Control, Signal and Systems, vol. 7, pages 235–254, 1994.

[Meek1989] D. S. Meek et D. J. Walton. *The Use of Cornu Spirals in Drawing Planar Curves of Controlled Curvature*. Journal of Computational and Applied Mathematics, vol. 25, no. 1, pages 69–78, 1989.

[Meek1992] D. S. Meek et D. J. Walton. *Clothoid Spline Transition Spirals*. Mathematics of Computation, vol. 59, no. 199, page 117, 1992.

[Meek2004] D. S. Meek et D. J. Walton. *A Note on Finding Clothoids*. Journal of Computational and Applied Mathematics, vol. 170, no. 2-2, pages 433–453, 2004.

[Mielenz1997a] Klaus D. Mielenz. *Computation of Fresnel Integrals*. Journal of Research of the National Institute of Standards and Technology, vol. 102, no. 3, pages 363–365, 1997.

[Mielenz1997b] Klaus D. Mielenz. *Computation of Fresnel Integrals*. Journal of Research of the National Institute of Standards and Technology, vol. 102, no. 3, pages 363–365, 1997.

[Milica2018] Lucian Milica, Alexandru Năstase et Gabriel Andrei. *Optimal path planning for a new type of 6RSS parallel robot based on virtual displacements expressed through Hermite polynomials*. Mechanism and Machine Theory, vol. 126, pages 14–33, 2018.

[Moon1991] S.B. Moon et Shaheen Ahmad. *Time scaling of cooperative multi-robot trajectories*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, no. 4, pages 900–908, 1991.

[Moreno2012] Jaime A. Moreno et Marisol Osorio. *Strict Lyapunov Functions for the Super-Twisting Algorithm*. IEEE Transactions on Automatic Control, vol. 57, no. 4, pages 1035–1040, apr 2012.

[Murphy1992] B.R. Murphy et I. Watanabe. *Digital shaping filters for reducing machine vibration*. IEEE Transactions on Robotics and Automation, vol. 8, no. 2, pages 285–289, apr 1992.

[Murray1991] R.M. Murray et S.S. Sastry. *Steering nonholonomic systems in chained form*. In [1991] Proceedings of the 30th IEEE Conference on Decision and Control, pages 1121–1126. IEEE, 1991.

[Nelson1989] Winston Nelson. *Continuous-curvature Paths for Autonomous Vehicles*. pages 1260–1264, 1989.

[Nilsson1969] Nils J. Nilsson. *A Mobile Automaton : An Application of Artificial Intelligence Techniques*. Rapport technique January, OSD or Non-Service DoD Agency, jan 1969.

[Omar2004] Frrag Omar, Fakhri KARRAY, Otman BASIR et L. YU. *Autonomous Overhead Crane System Using a Fuzzy Logic Controller*. Journal of Vibration and Control, vol. 10, pages 1255–1270, 2004.

[Oriolo2002] G Oriolo, A De Luca et M Vendittelli. Wmr control via dynamic feedback linearization : Design, implementation, and experimental validation, volume 10. 2002.

[Orlov2005] Y Orlov. *Finite Time Stability and Robust Control Synthesis of Uncertain Switched Systems*. SIAM J.CONTROL OPTIM., vol. 43, no. 4, pages 1253–1271, 2005.

[Oza2011] Harshal B Oza, Yury V Orlov et Sarah K Spurgeon. *Settling Time Estimate for a Second Order Sliding Mode Controller : A Homogeneity Approach*. In Preprints of the 18th IFAC World Congress Milano (Italy) August 28, 2011.

[Palleschi2019] Alessandro Palleschi, Manolo Garabini, Danilo Caporale et Lucia Pallottino. *Time-Optimal Path Tracking for Jerk Controlled Robots*. IEEE Robotics and Automation Letters, vol. 4, no. 4, pages 3932–3939, oct 2019.

[Park2006] Juyi Park, Pyung Hun Chang, Hyung Soon Park et Eunjeong Lee. *Design of learning input shaping technique for residual vibration suppression in an industrial robot*. IEEE/ASME Transactions on Mechatronics, vol. 11, no. 1, pages 55–65, 2006.

[Patle2019] B.K. Patle, Ganesh Babu L, Anish Pandey, D.R.K. Parhi et A. Jagadeesh. *A review : On path planning strategies for navigation of mobile robot*. Defence Technology, vol. 15, no. 4, pages 582–606, aug 2019.

[Peña2012] J Peña, Rob H B Fey et Henk Nijmeijer. *An Introduction to Parametric Resonance*. In Parametric Resonance in Dynamical Systems, pages 1–13. Springer, New York, NY, eds edition, 2012.

[Pfeiffer1987] Friedrich Pfeiffer et Rainer Johanni. *A concept for manipulator trajectory planning*. IEEE Journal on Robotics and Automation, vol. 3, no. 2, pages 115–123, apr 1987.

[Piazzi2002] Aurelio Piazzi, Guarino Corrado Lo Bianco et Massimo Bertozzi. *Quintic G 2 -Splines for the Iterative Steering of Vision-Based Autonomous Vehicles*. IEEE Transactions On Intelligent Transportation Systems, vol. 3, no. 2, pages 27–36, 2002.

[Piazzi2007] A. Piazzi, C.G. Lo Bianco et M. Romano. *Splines for the Smooth Path Generation of Wheeled Mobile Robots*. IEEE Transactions on Robotics, vol. 23, no. 5, pages 1089–1095, 2007.

[Ping2013] Ronald Ping, Man Chan, Karl A Stol et C Roger Halkyard. *Review of modelling and control of two-wheeled robots*. Annual Reviews in Control, vol. 37, no. 1, pages 89–103, 2013.

[Plestan2010] Franck Plestan, Yuri Shtessel, V. Brégeault et Alexander Poznyak. *New methodologies for adaptive sliding mode control*. International Journal of Control, vol. 83, no. 9, pages 1907–1919, sep 2010.

[Polyakov2009] Andrei Polyakov et Alex Poznyak. *Lyapunov function design for finite-time convergence analysis : Twisting controller for second-order sliding mode realization*. Automatica, vol. 45, no. 2, pages 444–448, 2009.

[Pontryagin1962] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze et E. F. Mishchenko. The Mathematical Theory of Optimal Processes. U. S. A., john wiley edition, 1962.

[Ramli2017] Liyana Ramli, Z Mohamed, Auwalu M Abdullahi, H I Jaafar et Izzuddin M Lazim. *Control Strategies for Crane Systems : A Comprehensive Review*. Mechanical Systems and Signal Processing, vol. 95, pages 1–23, 2017.

[Reeds1990] J. A. Reeds et L. A. Shepp. *Optimal Paths for a Car That Goes Both Forwards and Backwards*. Pacific Journal of Mathematics, vol. 145, no. 2, pages 367–393, 1990.

[Reuter1998] Johannes Reuter. *Mobile Robots Trajectories with Continuously Differentiable Curvature : An Optimal Control Approach*. IEEE International Conference on Intelligent Robots and Systems, vol. 1, no. October, pages 38–42, 1998.

[Rigatos2012] G G Rigatos. *Adaptive Fuzzy Control for Non-linear Dynamical*. IET Control Theory and Applications, vol. 6, no. August 2011, pages 2644–2656, 2012.

[Roy2019] Spandan Roy, Simone Baldi et Leonid M. Fridman. *On adaptive sliding mode control without a priori bounded uncertainty*. Automatica, no. xxxx, page 108650, oct 2019.

[Sage2010] H G Sage, M F De Mathelin et E Ostertag. *Robust control of robot manipulators : A survey*. International Journal of Control, vol. 72 :16, no. May 2013, pages 37–41, 2010.

[Salan2015] Serge Salan, Evan Drumwright et King-Ip Lin. *Minimum-Energy Robotic Exploration : A Formulation and an Approach*. IEEE Transactions on Systems, Man, and Cybernetics : Systems, vol. 45, no. 1, pages 175–182, jan 2015.

[Sankar1988] P. V. Sankar et L. A. Ferrari. *Simple Algorithms and Architectures for B-Spline Interpolation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 2, pages 271–276, 1988.

[Sastry1989] Shankar Sastry et Marc Bodson. Adaptive control : Stability, convergence, and robustness. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[Sawodny2002] Oliver Sawodny, Harald Aschemann et Stephan Lahres. *An Automated Gantry Crane As a Large Workspace Robot*. Control Engineering Practice, vol. 10, pages 1323–1338, 2002.

[Scheuer1998] A. Scheuer et Ch. Laugier. *Planning Sub-optimal and Continuous-Curvature Paths for Car-Like Robots*. In Proceedings of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, numéro October, pages 25–31, Victoria, B.C., Canada October, 1998. IEEE.

[Schwartz1983a] Jacob T. Schwartz et Micha Sharir. *On the "piano movers'" problem I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers*. Communications on Pure and Applied Mathematics, vol. 36, no. 3, pages 345–398, may 1983.

[Schwartz1983b] Jacob T. Schwartz et Micha Sharir. *On the Piano Movers' Problem : III. Coordinating the Motion of Several Independent Bodies : The Special Case of Circular Bodies Moving Amidst Polygonal Barriers*. The International Journal of Robotics Research, vol. 2, no. 3, pages 46–75, sep 1983.

[Schwartz1984] Jacob T. Schwartz et Micha Sharir. *On the piano movers' problem : V. The case of a rod moving in three-dimensional space amidst polyhedral obstacles*. Communications on Pure and Applied Mathematics, vol. 37, no. 6, pages 815–848, nov 1984.

[Sebbane2014] Yasmina Bestaoui Sebbane. Planning and Decision Making for Aerial Robots, volume 71. 2014.

[Sharpe1982] Richard J Sharpe et Richard W Thorne. *Numerical method for extracting an arc length parameterization from parametric curves*. Computer-Aided Design, vol. 14, no. 2, pages 79–81, mar 1982.

[Shiller1989] Zvi Shiller et Steven Dubowsky. *Robot Path Planning with Obstacles, Actuator, Gripper, and Payload Constraints*. The International Journal of Robotics Research, vol. 8, no. 6, pages 3–18, dec 1989.

[Shin1985] Kang G. Shin et Neil D. Mckay. *Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints*. IEEE Transactions on Automatic Control, vol. 30, no. 6, pages 531–541, 1985.

[Shtessel2007] Yuri B Shtessel, Ilya A Shkolnikov et Arie Levant. *Smooth Second-order Sliding Modes : Missile Guidance Application*. Automatica, vol. 43, pages 1470–1476, 2007.

[Siciliano2016] Bruno Siciliano et Oussama Khatib, editeurs. Springer Handbook of Robotics. Springer International Publishing, Berlin, Heidelberg, 2016.

[Siliciano2010] B Siliciano, L Sciavicco, L Villani et G. Oriolo. Robotics Modelling, Planning and Control. 2009 Springer-Verlag London Limited, London, 2010.

[Singer1990] N. C. Singer et W. P. Seering. *Preshaping command inputs to reduce system vibration*. Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, vol. 112, no. 1, pages 76–82, 1990.

[Singer1999] N. Singer, W. Singhose et W. Seering. *Comparison of Filtering Methods for Reducing Residual Vibration*. European Journal of Control, vol. 5, no. 2-4, pages 208–218, jan 1999.

[Singh1980] S. Singh. *Decoupling of invertible nonlinear systems with state feedback and precompensation*. IEEE Transactions on Automatic Control, vol. 25, no. 6, pages 1237–1239, dec 1980.

[Singh1993] T. Singh et S. R. Vadali. *Robust Time-Delay Control*. Journal of Dynamic Systems, Measurement, and Control, vol. 115, no. 2A, pages 303–306, jun 1993.

[Singhose2009] William Singhose. *Command shaping for flexible systems : A review of the first 50 years*. International Journal of Precision Engineering and Manufacturing, vol. 10, no. 4, pages 153–168, oct 2009.

[Slotine1987] Jean-jacques E Slotine et Weiping Li. *On the Adaptive Control of Robot Manipulators*. The International Journal of Robotics Research, vol. 06 N° 03, pages 49–59, 1987.

[Slotine1988] J. J. E. Slotine et Weiping Li. *Adaptive Manipulator Control : A Case Study*. IEEE Transactions on Automatic Control, vol. 33, no. 11, 1988.

[Slotine1991] Jean-Jacques E Slotine et Weiping Li. Applied Nonlinear Control. Prentice Hall, Englewood Cliffs, New Jersey 07632, prentice-h edition, 1991.

[Spong2004] Mark W. Spong, Seth Hutchinson et M. Vidyasagar. Robot dynamics and control. John Wiley & Sons, second edition, 2004.

[Åström2008] Karl Johan Åström et Carlos Canudas-de Wit. *Revisiting the LuGre friction model*. IEEE Control Systems, vol. 28, no. 6, pages 101–114, dec 2008.

[Sugie2003] Toshiharu Sugie, Kenji Fujimoto et Yutaka Kito. *Obstacle avoidance of manipulators with rate constraints*. IEEE Transactions on Robotics and Automation, vol. 19, no. 1, pages 168–174, feb 2003.

[Sun2016] Weichao Sun, Songyuan Tang, Huijun Gao et Jie Zhao. *Two Time-Scale Tracking Control of Nonholonomic Wheeled Mobile Robots*. IEEE Transactions on Control Systems Technology, vol. 24, no. 6, pages 2059–2069, nov 2016.

[Taleb2013] Mohammed Taleb, Arie Levant et Franck Plestan. *Pneumatic actuator control : Solution based on adaptive twisting and experimentation*. Control Engineering Practice, vol. 21, no. 5, pages 727–736, 2013.

[Tchoń2015] Krzysztof Tchoń, Adam Ratajczak et Ida Góral. *Lagrangian Jacobian inverse for nonholonomic robotic systems*. Nonlinear Dynamics, vol. 82, no. 4, pages 1923–1932, dec 2015.

[Tchon2018] Krzysztof Tchon et Joanna Ratajczak. *General Lagrange-Type Jacobian Inverse for Nonholonomic Robotic Systems*. IEEE Transactions on Robotics, vol. 34, no. 1, pages 256–263, feb 2018.

[Thompson1987] Stuart E. Thompson et Rajnikant V. Patel. *Formulation of Joint Trajectories for Industrial Robots Using B-Splines*. IEEE Transactions on Industrial Electronics, vol. IE-34, no. 2, pages 192–199, 1987.

[Thomsen2019] Dan Kielsholm Thomsen, Rune Søe-Knudsen, David Brandt, Ole Balling et Xuping Zhang. *Smooth online time-varying input shaping with fractional delay FIR filtering*. Control Engineering Practice, vol. 88, no. March, pages 21–37, jul 2019.

[Tran2009] Dinh Quoc Tran et Moritz Diehl. *An application of sequential convex programming to time optimal trajectory planning for a car motion*. In Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, volume 0, pages 4366–4371. IEEE, dec 2009.

[Tuttle2007] Timothy Douglas Tuttle. *Creating Time-optimal Commands for Linear Systems*. Thèse de Doctorat, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2007.

[Şucan2012] Ioan A. Şucan, Mark Moll et Lydia Kavraki. *The open motion planning library*. IEEE Robotics and Automation Magazine, vol. 19, no. 4, pages 72–82, 2012.

[Upadhyay2016] Saurabh Upadhyay et Ashwini Ratnoo. *Continuous-Curvature Path Planning With Obstacle Avoidance Using Four Parameter Logistic*

Curves. IEEE Robotics and Automation Letters, vol. 1, no. 2, pages 609–616, 2016.

[Upadhyay2018] Saurabh Upadhyay et Ashwini Ratnoo. *On Existence and Synthesis of Smooth Four Parameter Logistic Paths Inside Annular Passages*. IEEE Robotics and Automation Letters, vol. 3, no. 4, pages 4375–4382, 2018.

[Utkin1977] Vadim I. Utkin. *Variable Structure Systems with Sliding Modes*. IEEE Transactions on Automatic Control,, vol. AC-22, no. c, pages 212–222, 1977.

[Utkin1993a] Vadim I Utkin. *Sliding Mode Control Design Principles and Applications to Electric Drives*. IEEE Transactions on Industrial Electronics, vol. 40, no. 1, pages 23–36, 1993.

[Utkin1993b] V.I. Utkin. *Sliding mode control design principles and applications to electric drives*. IEEE Transactions on Industrial Electronics, vol. 40, no. 1, pages 23–36, feb 1993.

[Utkin2009a] VADIM I. Utkin, Guldner Jurgen et Jingxin Shi. Sliding Mode Control in Electro-Mechanical Systems. CRC Press Taylor & Francis Group, second edi edition, 2009.

[Utkin2009b] V.I. Utkin, Guldner Jurgen et Jingxin Shi. Sliding Mode Control in Electro-Mechanical Systems. CRC Press Taylor & Francis Group, second edi edition, 2009.

[Utkin2013a] Vadim I Utkin et Alex S Poznyak. *Adaptive sliding mode control with application to super-twist algorithm : Equivalent control method* . Automatica, vol. 49, no. 1, pages 39–47, 2013.

[Utkin2013b] Vadim I. Utkin et Alexander S. Poznyak. Adaptive sliding mode control, pages 21–53. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Van Nieuwstadt1998] Michiel J. Van Nieuwstadt et Richard M. Murray. *Real-time trajectory generation for differentially flat systems*. International Journal of Robust and Nonlinear Control, vol. 8, no. 11, pages 995–1020, sep 1998.

[Vázquez2013] Carlos Vázquez, Joaquin Collado et Leonid Fridman. *Control of a Parametrically Excited Crane : A Vector Lyapunov Approach*. IEEE Transactions on control systems technology, pages 1–9, 2013.

[Vergnano2012] Alberto Vergnano, Carl Thorstensson, Bengt Lennartson, Petter Falkman, Marcello Pellicciari, Francesco Leali et Stephan Biller. *Modeling and Optimization of Energy Consumption in Cooperative Multi-Robot Systems*. IEEE Transactions on Automation Science and Engineering, vol. 9, no. 2, pages 423–428, apr 2012.

[Veslin2011] Elkin Veslin, Jules Slama, Max Suell Dutra et Omar Lengerke. *Motion planning on Mobile Robots using Differential Flatness*. IEEE Latin America Transactions, vol. 9, no. 7, pages 1006–1011, 2011.

[Walton1996] D. J. Walton et D. S. Meek. *A planar cubic Bézier spiral*. Journal of Computational and Applied Mathematics, vol. 72, no. 1, pages 85–100, 1996.

[Walton2003] D. J. Walton, D. S. Meek et J. M. Ali. *Planar G2 transition curves composed of cubic Bézier spiral segments*. Journal of Computational and Applied Mathematics, vol. 157, no. 2, pages 453–476, 2003.

[Walton2005] D. J. Walton et D. S. Meek. *A Controlled Clothoid Spline*. Computers and Graphics (Pergamon), vol. 29, no. 3, pages 353–363, 2005.

[Walton2007] D J Walton et D S Meek. *G 2 Curve Design with a Pair of Pythagorean Hodograph Quintic Spiral Segments*. Computer Aided Geometric Design, vol. 24, pages 267–285, 2007.

[Walton2012] D. J. Walton et D. S. Meek. *A further generalisation of the planar cubic Bzier spiral*. Journal of Computational and Applied Mathematics, vol. 236, no. 11, pages 2869–2882, 2012.

[Wang1988] Kesheng Wang. *B-splines Joint Trajectory Planning*. Computers in Industry, vol. 10, no. 2, pages 113–122, 1988.

[Wang1990] Chi-hsu Wang et JONG-GIN HORNG. *Constrained Minimum-Time Path Planning for Robot Manipulators Via Virtual Knots of the Cubic B-Spline Functions*. IEEE TRANSACTIONS ON AUTOMATIC CONTROL, vol. 35, no. 5, pages 573–577, 1990.

[Weinmann1991] Alexander Weinmann. Uncertain Models and Robust Control. Springer Vienna, Vienna, 1991.

[Wigstrom2011] Oskar Wigstrom et Bengt Lennartson. *Energy optimization of trajectories for high level scheduling*. In 2011 IEEE International Conference on Automation Science and Engineering, volume 2011, pages 654–659. IEEE, aug 2011.

[Wigstrom2013] Oskar Wigstrom, Bengt Lennartson, Alberto Vergnano et Claes Breitholtz. *High-Level Scheduling of Energy Optimal Trajectories*. IEEE Transactions on Automation Science and Engineering, vol. 10, no. 1, pages 57–64, jan 2013.

[Wu2009] Wenxiang Wu, Shiqiang Zhu et Songguo Liu. *Smooth joint trajectory planning for humanoid robots based on B-splines*. In 2009 IEEE International Conference on Robotics and Biomimetics, ROBIO 2009, pages 475–479, 2009.

[Yang2008a] Kwangjin Yang et Salah Sukkarieh. *3d Smooth Path Planning for a Uav in Cluttered Natural Environments*. In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pages 794–800, 2008.

[Yang2008b] Kwangjin Yang et Salah Sukkarieh. *Planning Continuous Curvature Paths for Uavs Amongst Obstacles*. Proceedings of the 2008 Australasian Conference on Robotics and Automation, ACRA 2008, 2008.

[Yang2010] Kwangjin Yang et Salah Sukkarieh. *An Analytical Continuous-curvature Path-smoothing Algorithm*. IEEE Transactions on Robotics, vol. 26, no. 3, pages 561–568, 2010.

[Yang2018] Hongjiu Yang, Mingchao Guo, Yuanqing Xia et Lei Cheng. *Trajectory Tracking for Wheeled Mobile Robots Via Model Predictive Control with Softening Constraints*. IET Control Theory & Applications, vol. 12, pages 206–214, 2018.

[Zhang2017] Dan Zhang et Bin Wei. *A review on model reference adaptive control of robotic manipulators*. Annual Reviews in Control, vol. 43, pages 188–198, 2017.

Table des figures

1 Exemples génériques de robots industrielle 1
2 Exemples de robots destinés pour des applications services 2
3 Estimation annuelle du stock mondial des robots 3
4 Automatisation d'un système robotisé 4

1.1 Représentation du mouvement d'un corps rigide dans l'espace 3D 9
1.2 Chemins lisses dans le plan 15
1.3 Interpolation vs Approximation 16
1.4 Un chemin avec des spécifications sur les positions finales, les directions et la courbure dans le point b 16
1.5 Interpolation avec une continuité paramétrique C^1 17
1.6 Interpolation avec une continuité paramétrique G^1 18
1.7 La propriété d'enveloppe convexe pour les courbes d'approximation . . . 20

2.1 Commandes dynamiques et cinématiques d'un système de deuxième ordre 34

3.1 Robot mobile différentielle 52
3.2 Élaboration d'un profil de vitesse à jerk limité 58
3.3 Profile à accélération limitée 58
3.4 PIONEER 3DX mobile robot. 64
3.5 Tâche planifiée : (a) Chemin désiré et (b) Courbure du chemin 64
3.6 Vitesse et accélération du profil de mouvement. 65
3.7 Représentation paramétrique des Trajectoires de référence 65
3.8 Tâche exécutée (a) Mouvement du robots le long du chemin (b) Norme Cartésienne d'erreurs de suivie 66
3.9 Signaux de commande (a) Vitesses des roue gauche et droite (b) Accélérations des roue gauche et droite 67
3.10 Plan de l'environnement des tests (les locaux de laboratoire des systèmes intelligent), "S : point de départ, G, point d'arrivé et chemin désiré. 67
3.11 Chemin panifié (a) Coordonnées Cartésiennes (b) Courbure. 68
3.12 Profil de mouvement selon les deux algorithmes. 70
3.13 Tâche exécutée (a) Mouvement du robot (b) Norme Cartésienne de l'erreur 70
3.14 Signaux de commande pour le test 2 (a) vitesse de la roue gauche (b) Vitesse de roue droite 71
3.15 Normes Cartésiennes de l'erreur sous les deux contrôleurs dans le cas nominal. 72
3.16 Normes Cartésiennes de l'erreur sous les deux contrôleurs sous des perturbations exogènes. 72
3.17 Normes Cartésiennes de l'erreur sous les deux contrôleurs sous des incertitudes de mesure. 73

4.1 Portique de trois dimensions 77
4.2 Profil de vitesse à snap limité 81
4.3 Tâche planifiée 87
4.4 Performances de suivi 88
4.5 Signaux de commande 89
4.6 Performances du système d'anti-ballant 90

5.1 Prototype de test : Cyton Gamma 1500 robot et Servomoteur Dynamixel . 100
5.2 Évolution du robot sans charge attachée et des trajectoires sinusoïdales. . 104
5.3 Évolution du robot avec une charge attachée et des trajectoires sinusoïdales. 105
5.4 Évolution de robot sans charge attachée avec des trajectoires polynomiales. 108
5.5 Évolution du robot avec une charge attachée le long d'une trajectoire polynomiale. 109
5.6 Analyse d'erreur de suivi 110

Liste des tableaux

3.1 Conditions d'interpolation pour le deuxième exemple 68
3.2 Caractéristiques du profil de mouvement pour la deuxième tâche 69

4.1 Points de passages et paramètres d'interpolation 86

5.1 Évaluation des performances des contrôleurs proposés 110

الملخص: تتناول هذه الأطروحة مشكلات التخطيط والتحكم في حركة الأنظمة الميكانيكية الآلية مع الأخذ بعين الاعتبار معايير وقيود مختلفة للكفاءة. بعد تنقيحات واسعة النطاق في أساليب التخطيط والتحكم في الأنظمة الآلية، ركزت دراستنا على ثلاثة أنواع من الأنظمة، روبوتات متحركة ذات عجلات مع قيود كلية وغير كلية، رافعات ثلاثية الأبعاد ذات بنية مرنة، وأذرع مناورة مع معلمات ديناميكية غير معروفة. تستند خوارزميات التخطيط المقترحة إلى مقاربة منفصلة تتمثل في تحديد المسار الذي يجب أن يتحرك فيه الروبوت بأمان، ثم إلزامه بقانون زمني معين من أجل ضمان الكفاءة في تحقيق الأهداف المرجوة (تحسين الوقت ،الطاقة ...). يتم اعتماد تتبع المسارات المرجعية من خلال تطبيق تقنيات التخطيط الكلي بواسطة التغذية الخلفية ، التحكم في وضع انزلاق عالي المستوى وأدوات تحكم تكيفية لتغطية مجموعة واسعة من حالات عدم اليقين. تم التأكد من فعالية الطرق المقترحة من خلال دراسات نظرية واختبارات عملية.

الكلمات المفتاحية: الأنظمة غير الخطية الآلية ، تخطيط و تحسين الحركة ، المقاربة المنفصلة، تتبع المسار، التحكم القوي والتكيف.

Résumé : Cette thèse aborde les problèmes de planification et de commande du mouvement des systèmes mécaniques automatisés sous la considération de différents critères d'optimalité et différentes contraintes de fonctionnement. Après des révisions approfondies sur les méthodes de planifications et de commandes des systèmes robotiques, on a concentré notre études sur trois types de systèmes, les robots mobiles a roues avec des contraintes holonomes et non-holonomes, un portique de trois dimensions avec une structure flexible, et les bras manipulateurs avec des paramètres dynamiques inconnus. Les algorithmes de planification proposés sont basés sur une approche découplée qui consiste à définir le chemin dans lequel le robot doivent se déplace en toute sécurité puis à assigne une loi horaires à ce chemin pour atteindre des objectifs désirés (optimisation du temps de cycle, d'énergie et du jerk) prenant en compte la dynamique/cinématique du système. Le suivi des trajectoires de référence générées est adopté en appliquant des techniques linéarisations par bouclage, de commande par mode glissant d'ordre supérieur et de commande adaptative pour couvrir une large gamme d'incertitudes. Des études théoriques et des tests pratiques sont inclus pour confirmer l'efficacité des méthodes proposées.

Mots clés : systèmes non linéaires robotisés, planification et optimisation de mouvement, approche découplée, suivi des trajectoires, commande robuste et adaptative, mode glissant.

Abstract: This thesis addresses the problems of motion planning and control for automated mechanical systems under consideration of different optimality criterions and different operating constraints. After extensive reviews to the planning and control methods of robotic systems, our study focused on three types of systems, a wheeled mobile robots with holonomic and non-holonomic constraints, a three-dimensional overhead crane system with a flexible structure, and manipulator arms with unknown dynamical parameters. The proposed planning algorithms are based on a decoupled approach which consists on defining the path in which the robot must move safely and then assigns a timing law to achieve desired objectives (optimization of cycle time, jerk,…) taking into account the dynamics/kinematics of the system. The tracking of the generated reference trajectories is achieved by using feedback linearizing control techniques, higher order sliding mode control approaches and adaptive control techniques to cover a wide range of uncertainties. Theoretical studies and practical tests are included to confirm the effectiveness of the proposed methods.

Key words: robotic nonlinear systems, motion planning, decoupled approach, trajectory tracking, robust and adaptive control, Sliding Mode.