



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

ÉCOLE NATIONALE POLYTECHNIQUE

Département automatique

Laboratoire de Commande des Processus



MÉMOIRE DE PROJET DE FIN D'ÉTUDES
Pour l'obtention du diplôme
D'INGÉNIEUR D'ÉTAT EN AUTOMATIQUE

**COMMANDE NON LINÉAIRE
DANS
LE DOMAINE DE POSITION :
APPLICATION AUX SYSTÈMES
ROBOTISÉS**

Etudié par :

DAHMANI Mohamed Lamine

BENKHELLA Imene

Proposé et dirigé par :

Pr. M.TADJINE (ENP)

Dr A.REZOUG (CDTA)

Soutenu publiquement le 20 juin 2017 devant un jury composé de

Président	M. Mohamed Seghir BOUCHRIT	Professeur	ENP
Encadreur	M. Mohamed TADJINE	Professeur	ENP
Co-Encadreur	M. Amar REZOUG	Docteur d'état	CDTA
Examineur	M. Rachid ILLOUL	Maître de Conférences A	ENP

ENP 2017



ÉCOLE NATIONALE POLYTECHNIQUE

Département automatique

Laboratoire de Commande des Processus



MÉMOIRE DE PROJET DE FIN D'ÉTUDES
Pour l'obtention du diplôme
D'INGÉNIEUR D'ÉTAT EN AUTOMATIQUE

**COMMANDE NON LINÉAIRE
DANS
LE DOMAINE DE POSITION :
APPLICATION AUX SYSTÈMES
ROBOTISÉS**

Etudié par :

DAHMANI Mohamed Lamine

BENKHELLA Imene

Proposé et dirigé par :

Pr. M.TADJINE (ENP)

Dr A.REZOUG (CDTA)

Soutenu publiquement le 20 juin 2017 devant un jury composé de

Président	M. Mohamed Seghir BOUCHRIT	Professeur	ENP
Encadreur	M. Mohamed TADJINE	Professeur	ENP
Co-Encadreur	M. Amar REZOUG	Docteur d'état	CDTA
Examineur	M. Rachid ILLOUL	Maître de Conférences A	ENP

ENP 2017

ملخص

في هذا العمل، قمنا بتطوير سلسلة من التحكمات غير الخطية لتطبيقات تتبع كفاف مخصصة خصيصاً للروبوتات الصناعية وصياغتها في مجال الموقف. ويتم تقييم التحكم في مجال الموقع، بمساهمته الملحوظة في الحد من خطأ تتبع الكفاف، بتطبيقه على هيكلين وروبوتين: مناور الروبوت والروبوت المتحرك. وكانت قوانين التحكم التي تم الحصول عليها بعد تحول نموذج النظام قادرة على الحد بشكل كبير من خطأ تتبع الكفاف عن طريق الحفاظ نسبياً على نفس مستوى السيطرة كما نظيرها توليفها في المجال الزمني

كلمات مفتاحية: التحكم غير الخطية، مجال الموقف، خطأ تتبع كفاف

Abstract :

In this work, WE developed a series of nonlinear controls for contour tracking applications dedicated specifically to industrial robots and their formulation in the position domain. The position domain control in the position domain, with its remarkable contribution to the reduction of the contour tracking error, is evaluated by its application to two robot structures: a robot manipulator and mobile robot. The control laws obtained after transformation of the system model were able to considerably reduce the contour tracking error by keeping relatively the same level of control as its counterpart synthesized in time domain.

Keywords: nonlinear control, position domain, contour tracking error

Résumé :

Le travail présenté dans ce mémoire concerne le développement de commandes non linéaires pour des applications de suivi de contours dédiées particulièrement aux robots industriels et leur formulation dans le domaine de position. La méthode de position commande dans le domaine de position, avec leur contribution remarquable à la diminution de l'erreur de suivi de contour, est évaluée par son application à deux structure de robots : robot manipulateur et robot mobile. Les commandes obtenues après transformation du modèle du système étaient capables de réduire considérablement l'erreur de suivi de contour en conservant relativement le même niveau de commande que son homologue synthétisée dans l'espace temporel.

Mots-clés : commandes non linéaires, domaine de position, erreur de suivi de contour

Remerciements

Tout d'abord, nous remercions Allah, le tout puissant de nous avoir accordés le savoir, le droit chemin, l'opportunité de poursuivre nos études et la force pour réaliser ce travail.

Le présent travail a été effectué sous la direction de Messieurs M. Tadjine et A.Rezzoug que nous remercions vivement pour leur encadrement, leurs directives, leurs conseils précieux et surtout la confiance qu'ils nous ont accordés.

Nous tenons également à remercier Messieurs R.Illoul et M.S.Boucherit, membres du jury qui ont accepté de lire et d'évaluer notre travail.

Nos sincères remerciements et notre profonde gratitude vont à tous nos enseignants du département du génie automatique en particulier et du génie électrique en général de l'Ecole Nationale Polytechnique pour les connaissances et le savoir faire qu'ils nous ont transmis pendant les cinq années de notre formation.

Nous tenons à remercier de manière particulière nos chers parents, frères et sœurs respectives "que Dieu les protège" qui ont toujours été là pour nous et qui nous ont soutenus pendant toutes ces années d'études. On espère qu'ils trouvent dans ce mémoire toute notre reconnaissance et tout notre amour.

Enfin, nous remercions toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

TABLE DE MATIÈRES

Liste des figures

Liste des tableaux

Notations et Symboles

Abréviations

Introduction	13
1 Généralités	16
1.1 Commande décentralisée	17
1.1.1 Contrôle PID/PD	18
1.1.2 Commande par mode glissant	18
1.2 Commande coordonnée	19
1.2.1 Commande maître / esclave	20
1.2.2 Commande à événements orientés (Event Based Control)	20
1.3 Autres techniques de commande	21
1.3.1 Commande répétitif	21
1.3.2 Commande par apprentissage itératif CAI(ILC)	21
1.3.3 Objectifs de la CAI	23
1.3.4 Commande par Forwarding	25
1.3.5 Les étapes de la synthèse	26
2 La commande dans le domaine de position	29
2.1 Erreur de poursuite et erreur de contour	30

2.2	Calcul de l'erreur de contour	32
2.2.1	Contour linéaire	32
2.2.2	Contour circulaire	33
2.2.3	Contour non linéaire (forme libre)	34
2.3	Génération de trajectoire	35
2.3.1	Polynôme d'ordre 5	35
2.3.2	Contours Linéaires	36
2.3.3	Contours non linéaires	36
2.3.4	Contour circulaire	37
2.4	Commande dans le domaine de position	37
2.4.1	Dérivation relative et transformation du domaine temporel vers le domaine de position	37
2.4.2	Modèle dynamique temporel d'un robot à n ddl	38
2.4.3	Modèle dynamique dans le domaine de position	39
2.5	Remarques	40
3	Synthèse des lois de commandes	42
3.1	Système robotique dans le domaine de position	43
3.1.1	Robot manipulateur	43
3.1.2	Propriétés et hypothèses du système dynamique	44
3.1.3	Robot mobile	45
3.1.4	Modèle dynamique dans le domaine de position	48
3.2	La commande PD-MG	49
3.2.1	Synthèse de la de commande pour le robot manipulateur	49
3.2.2	Commande dans le domaine temporel PD-MG	50
3.2.3	Commande dans le domaine de position PDP-MG	50
3.2.4	L'analyse de stabilité	52
3.2.5	Synthèse de la de commande pour le robot mobile	54
3.3	La commande par Forwarding	54
3.3.1	Synthèse des des lois de commande pour un robot manipulateur	54
3.4	Commande par Mode glissant	58
3.4.1	Synthèse de la loi de commande pour le robot manipulateur	58
3.4.2	Synthèse de la de commande pour le robot mobile	59
3.5	Synthèse de la commande CAI-MG	61
3.5.1	Étude de convergence [46]	62
3.5.2	Synthèse de la loi de commande pour le robot manipulateur	64
3.5.3	Synthèse de la loi de commande pour le robot mobile	65

4 Simulation et Résultats	66
4.1 Configuration de la simulation	67
4.1.1 Robot Manipulateur Planaire	67
4.1.2 Robot Mobile	68
4.2 Contours de références utilisés durant les simulations	68
4.2.1 Contours Linéaires	69
4.2.2 Contour Ligne	69
4.2.3 Contours non linéaires	71
4.3 Transformation de trajectoires	73
4.4 Résultats des simulations	75
4.4.1 Robot manipulateur planaire	75
4.4.2 Robot mobile	94
4.5 Synthèse comparative	106
4.5.1 Étude quantitative	106
4.5.2 Test de Robustesse	109
4.5.3 Commande par Forwarding	109
4.5.4 Commande par MG	111
4.5.5 Commande par CAI-MG	113
4.5.6 Étude qualitative	115
Conclusion générale	116
Bibliographie	119
Annexes	124

LISTE DES FIGURES

1.1	Commande décentralisée [7]	17
1.2	Surface de glissement[11]	19
1.3	Contrôle coordonné [7]	19
1.4	Schéma de la commande orientée événements[7]	21
1.5	Schéma bloc de la CAI [24]	24
2.1	Schéma de commande dans le domaine de position [7]	30
2.2	L'erreur de poursuite et l'erreur de contour [33]	31
2.3	Modèle de l'erreur pour un contour linéaire [1]	32
2.4	Modèle de l'erreur pour un contour non linéaire [7]	34
2.5	Schéma de commande dans le domaine de position PD [6]	40
2.6	Schéma de commande dans le domaine de position [7]	40
3.1	Schéma d'un robot manipulateur dans le domaine de position[5]	43
3.2	Robot Mobile uni-cycle [43]	45
3.3	Poursuite de trajectoires admissibles[43]	49
3.4	Schéma blocs pour la commande par MG standard et le PDC-MG.	50
4.1	Robot Manipulateur Planaire à 3.d.d.[7]	67
4.2	Robot Mobile uni-cycle [43]	68
4.3	Contour Ligne (Espace de travail)	70
4.4	Positions et vitesses désirées des articulations	70
4.5	Contour Cercle (Espace de travail)	71
4.6	Positions et vitesses désirées des articulations (contour circulaire)	72
4.7	Lookup Table (interp1) (The MathWorks, Inc)	73
4.8	Positions et vitesses désirées des articulations (contour ligne)	73

4.9	Positions et vitesses désirées des articulations (contour cercle)	74
4.10	Contour Arc (Espace de travail)	74
4.11	Contour Sinus (Espace de travail)	75
4.12	Résultat de suivi du contour (contour Ligne)	76
4.13	Commandes dans domaine temporel/position (Contour Ligne)	77
4.14	L'erreur du Contour Ligne (PD-MG)	77
4.15	Résultat de suivi du contour (contour Cercle)	78
4.16	Commandes PD-MG dans domaine temporel/position (Contour Cercle)	79
4.17	L'erreur du Contour Cercle (PD-MG)	79
4.18	Résultat de suivi du contour (contour Ligne)	81
4.19	Commandes Forwarding dans domaine temporel/position (Contour Ligne)	81
4.20	L'erreur du Contour Ligne (Forwarding)	82
4.21	Résultat de suivi du contour (contour Cercle)	82
4.22	Commandes par Forwarding dans domaine temps/position (Contour Cercle)	83
4.23	L'erreur du Contour Cercle (Forwarding)	83
4.24	Résultat de suivi du contour (contour Ligne)	85
4.25	Commandes par MG dans domaine temporel/position (Contour Ligne)	85
4.26	L'erreur du Contour Ligne (Mode Glissant)	86
4.27	Résultat de suivi du contour (contour cercle)	87
4.28	Commandes MG dans domaine temporel/position (Contour cercle)	87
4.29	L'erreur du Contour Cercle (MG)	88
4.30	Résultat de suivi du contour (contour Ligne)	89
4.31	Commandes par CAI-MG dans domaine(Contour Ligne)	90
4.32	L'erreur du Contour Ligne (CAI-MG)	90
4.33	Evolution de $\sup Ec $ au cours des itérations (contour ligne)	91
4.34	Résultat de suivi du contour (contour Cercle)	92
4.35	Commandes par CAI-MG dans domaine(Contour Cercle)	92
4.36	L'erreur du Contour Cercle (CAI-MG)	93
4.37	Evolution de $\sup Ec $ au cours des itérations (contour cercle)	93
4.38	Résultat de suivi du contour (contour arc)	95
4.39	Commandes dans domaine temporel/position (Contour arc)	96
4.40	L'erreur du Contour arc (PD-MG)	96
4.41	Résultat de suivi du contour (contour sinus)	97
4.42	Commandes PD-MG dans domaine temporel/position (Contour sinus)	98
4.43	L'erreur du Contour sinus (PD-MG)	98
4.44	Résultat de suivi du contour (contour arc)	100
4.45	Commandes par MG dans domaine temporel/position (Contour arc)	100

4.46	L'erreur du Contour arc (Mode Glissant)	101
4.47	Résultat de suivi du contour (contour sinus)	101
4.48	Commandes MG dans domaine temporel/position (Contour sinus)	102
4.49	L'erreur du Contour sinus (MG)	102
4.50	Résultat de suivi du contour (contour arc)	104
4.51	Commandes par CAI-MG dans domaine(Contour arc)	104
4.52	L'erreur du Contour arc (CAI-MG)	105
4.53	Evolution de $\sup Ec $ au cours des itérations (contour ligne)	105
4.54	Résultat de suivi du contour (contour Ligne)	109
4.55	Commandes Forwarding dans domaine temporel/position (Contour Ligne)	110
4.56	L'erreur du Contour Ligne (Forwarding)	110
4.57	Résultat de suivi du contour (contour Ligne)	111
4.58	Commandes MG dans domaine temporel/position (Contour Ligne)	112
4.59	L'erreur du Contour Ligne (MG)	112
4.60	Résultat de suivi du contour (contour Ligne)	113
4.61	Commandes CAI-MG dans (Contour Ligne)	113
4.62	L'erreur du Contour Ligne (CAI-MG)	114

LISTE DES TABLEAUX

4.1	Paramètres structurels du robot manipulateur série[5]	67
4.2	Paramètres du Robot Mobile [43]	68
4.3	Positions initiales et finales de l'effecteur pour le contour ligne.	69
4.4	Les paramètres du contour circulaire	71
4.5	Paramètres du régulateur PD-MG	76
4.6	Paramètres du régulateur Forwarding	80
4.7	Paramètres du régulateur Mode glissant	84
4.8	Paramètres du CAI-MG	89
4.9	Paramètres du régulateur PD-MG	95
4.10	Paramètres du régulateur Mode glissant	99
4.11	Paramètres du CAI-MG	103
4.12	Les performances de suivi pour le contour ligne	107
4.13	Les performances de suivi pour le contour cercle	108
4.14	Tableau d'appréciations des performances des commandes	115

Notations et Symboles

Paramètres	Définition
A_i	Matrice de contrainte.
$C(q, \dot{q})$	Vecteur de Coriolis et de forces centrifuges.
C_y, C_x	Les gains de contrôle à couplage transverse.
$F(t, q, \dot{q})$	Vecteur de forces des frottements.
$G(q)$	Vecteur des forces gravitationnelles.
I_i	Inertie du i ème corps.
I	Matrice d'identité.
J	Matrice Jacobienne.
K_d	Matrice de gain dérivé.
K_i	Matrice de gain inertielle.
K_p	Matrice de gain proportionnel.
S	Complément de Schur.
$V(e_s(q_m), \dot{e}_s(q_m))$	Fonction candidate de Lyapunov dans le domaine de position.
$\frac{dV}{dqm}$	Dérivé relative de de la fonction Lyapunov par rapport au mouvement maitre.
e	Erreur de poursuite.
\dot{e}	Vitesse de l'erreur de poursuite.
e_c	Erreur de suivi du contour.
e'	Erreur de position relative.
g	l'accélération de la pesanteur.
l_i	Longueur de l' i -ème corps.
q	Position angulaire.
\dot{q}	Vitesse angulaire.
\ddot{q}	Accélération angulaire.
sat	La fonction saturation .
$sign$	la fonction signe .
τ	Les couples des articulations .

Abréviations

Abréviation

Définition

PD/PID :

Proportionnel Dérivé/ Proportionnel Intégral Dérivé.

CNC :

Commande par numérisation informatisée .

CCC :

Commande par couple transverse

MG :

Modes Glissants.

PD-MG :

Proportionnel Dérivé- Modes Glissants.

PDP-MG :

Proportionnel Dérivé- Modes Glissants
dans le domaine de position.

CAI :

Commande par Apprentissage Itératif.

d.d.l :

Degrés de liberté.

FD :

Commande par Forwarding

CDP

Commande dans le domaine de position

INTRODUCTION GÉNÉRALE

Introduction Générale

Dans plusieurs opérations industrielles comme le soudage, l'assemblage et le coupage et médicales comme les robots chirurgicaux, le suivi de contour est l'un des problèmes courants auquel font face les robots manipulateurs et robots industriels. L'objectif du suivi de contour est de poursuivre une trajectoire de référence indépendante du temps avec une erreur de contour minimale. Par conséquent, la réduction des erreurs de suivi de contour demeure une préoccupation primordiale surtout dans les applications de suivi des contours qui nécessitent une haute précision afin d'assurer la qualité des produits finaux. A cet effet, de nombreuses méthodes de commande ont été développées dans le domaine temporel pour remédier aux problèmes de suivi de contour. Parmi ces commandes, on cite notamment la Proportionnelle-Intégrale-Dérivée (PID), le CCC introduit par Koren [1] et la commande orientée événement. Des techniques plus avancées ont été proposées par la suite telle que : la commande adaptative [2] et la commande robuste [3], etc.

La majorité des commandes évoquées sont fréquemment synthétisées dans le domaine articulaire, alors que la commande de l'organe terminal, assurant l'exécution de la tâche principale du robot a été peu étudiée. En effet, la commande en poursuite de trajectoire pour un bras manipulateur consiste à contrôler le mouvement de l'effecteur (organe terminal) suivant une trajectoire désirée de manière efficace et précise. Afin d'obtenir des exigences élevées de performance de contour [4], introduisent la notion de domaine de position comme nouvelle façon de représenter les systèmes. Cette nouvelle formulation va permettre d'avoir une manière différente d'analyse et de synthèse des lois de commande et par conséquent améliorer les performances et la robustesse du système commandé. Contrairement au domaine temporel que les automaticiens ont l'habitude d'utiliser, la "commande dans le domaine de position" consiste à analyser et synthétiser des commandes dans l'espace opérationnel du robot dont le facteur du temps n'est pas primordial. Pour développer des commandes dans le domaine de position, le mouvement du système poly-articulaire est considéré comme un mouvement de maître-esclaves où le mouvement du maître qu'est généralement l'axe lié à l'articulation de base du robot est utilisé comme une variable indépendante. Et les mouvements des esclaves -qui sont les axes du bras- sont décrits comme fonctions du mouvement du maître et ceux imposés par les contours à suivre [5] [6].

Les modèles dynamiques de ces systèmes sont ainsi représentés dans le domaine de position et développés en fonction de la réponse du maître grâce à une cartographie unique entre le domaine temporel et le domaine de position. Les commandes développées dans le domaine de position sont appliquées au suivi de contours linéaires et non linéaires. Ensuite, une

étude comparative est effectuée. Les résultats de simulation démontrent l'efficacité des commandes dans le nouvel espace par rapport à celles temporelles en matière de suivi de contour.

Théoriquement, cette nouvelle approche assure une meilleure poursuite de contour. En effet, en formulant le problème de commande dans le domaine de position, l'erreur de poursuite de l'axe de référence ne contribue pas à l'erreur globale du contour. Partant du fait que la commande PID soit la plus populaire dans les applications de la théorie de commande. Ouyang et al ont tout d'abord proposé des commandes de type PID et PD dans le domaine des positions appliquées notamment sur des machines CNC ainsi que des robots manipulateurs. Ils ont démontré l'efficacité de cette nouvelle approche vis-à-vis de l'amélioration des performances en matière de poursuite de contours linéaires et non linéaires.

Dans le présent travail, on s'intéresse au développement de commandes non linéaires pour le suivi de contours par un robot manipulateur et un robot mobile dans le domaine de position et à la comparaison de leurs performances par rapport à leurs homologues temporelles.

Dans le premier chapitre, nous donnons des généralités sur les approches de commandes classiques et modernes développées dans le cadre de l'amélioration du suivi de contour et de poursuite de trajectoires pour les robots industriels.

Le deuxième chapitre offre une méthodologie de formulation d'un problème de commande d'un système structuré maître-esclave selon la nouvelle méthode introduite par Ouyang et al et dite approche de commande dans le domaine de position. Cette méthodologie s'avère très performante lorsqu'il s'agit de l'amélioration du suivi de contour dans les processus d'usinage nécessitant une très haute précision.

Dans le troisième chapitre, nous concevons des lois de commande non linéaires basées sur le modèle dynamique du système robotisé et effectuons leur transformation dans le domaine de position. Ensuite, nous étudions les conditions de convergence de chaque loi synthétisée.

Le quatrième et dernier chapitre traite la simulation des commandes proposées ainsi que l'interprétation des résultats obtenus. Nous l'achevons par une étude comparative détaillée couvrant plusieurs aspects quantitatifs et qualitatifs des méthodes de commande considérées.

Enfin, une conclusion générale donnera une synthèse du travail effectué et résumera les principaux résultats obtenus ainsi que les perspectives envisagées.

CHAPITRE 1

GÉNÉRALITÉS

Chapitre 1 : Généralités

Introduction

Dans ce chapitre nous allons faire le tour d'horizon sur quelques approches de commande adoptées pour l'amélioration des performances d'un système robotisé asservi. Les techniques les plus anciennes avaient pour objectif la minimisation des erreurs de poursuite au niveau articulaire tandis que celles récentes tendent à donner plus d'importance aux performances globales du suivi de contour. Nous exposerons à la fin des notions élémentaires associées aux applications de suivi de contour.

1.1 Commande décentralisée

Cette catégorie de contrôleurs comprend des systèmes de contrôle où chaque actionneur (ou toute autre partie du manipulateur) est commandé indépendamment par sa propre boucle de commande (contrôleur, capteurs, etc). Leur principe de fonctionnement est schématisé dans la figure(1.1). Les contraintes, les trajectoires souhaitées, et les performances désirées sont fournies par l'opérateur, mais chaque actionneur agit parallèlement aux autres, en tant qu'une unité indépendante, sans se rendre compte de l'existence du reste du mécanisme [4].

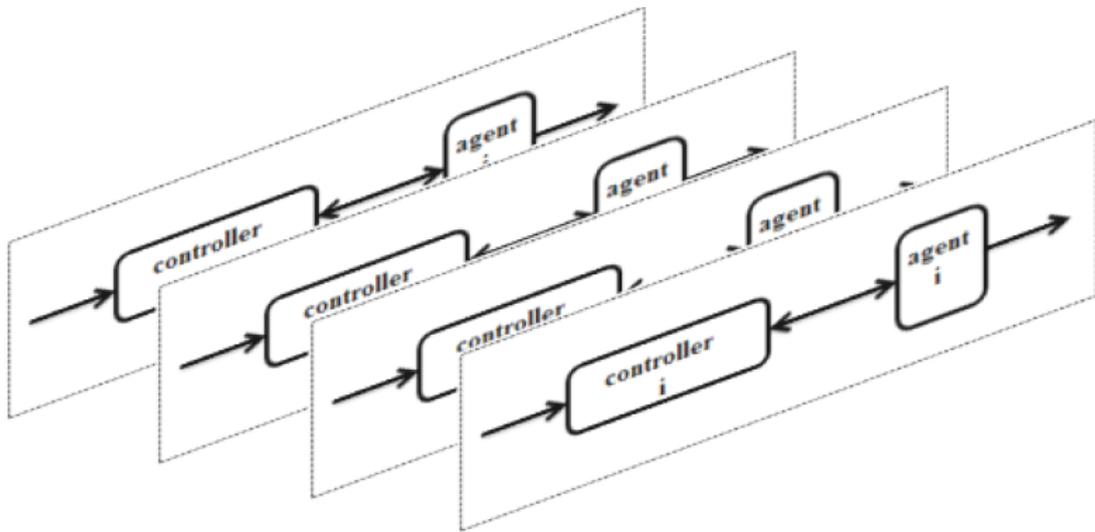


FIGURE 1.1 – Commande décentralisée [7]

1.1.1 Contrôle PID/PD

Le contrôle PID et ses autres versions (PI et PD) ont été les plus populaires parmi les techniques de commandes appliquées aux robots industriels. En fait, englobant plus de 90% des applications industrielles, les contrôleurs PID dominent le monde du contrôle à rétroaction [8]. Pour un robot à plusieurs ddl, la commande PID est utilisée indépendamment pour chaque axe ou variable de mouvement, le contrôleur PID pour un axe i ou une variable d'état i prend la forme suivante

$$\tau_i = K_{P_i}e_i + K_{D_i}\dot{e}_i + K_{I_i} \int e_i dt \quad (1.1)$$

Où $e = q_{d_i} - q_i$ est l'erreur de suivi du mouvement de l'axe i , et K_P, K_D, K_I sont des matrices de gains constantes et définies positives.

Les principaux avantages de ce contrôleur sont sa nature simple et linéaire et sa facilité de mise en œuvre. En dépit de sa simplicité, le contrôle PID est assez puissant pour traiter des problèmes pratiques tels que la saturation des actionneurs et la liquidation de l'intégrateur. En revanche, le contrôle PID demeure inapproprié pour certains cas en raison de son incapacité à s'adapter à la dynamique du système. Les propriétés dynamiques d'un système robotisé varient avec le temps pour des raisons comme l'usure, le frottement, les perturbations, ou simplement des changements dans les conditions d'opération, ce qui crée un besoin de redimensionnement du correcteur avec chaque nouvelle tâche.

1.1.2 Commande par mode glissant

Pour compenser les incertitudes du système dynamique, une loi de commande dite par mode glissant (MG) a été proposée par Bartolini [9]. Le concept d'une surface de glissement est introduit, en fonction de l'erreur de suivi du système comme suit :

$$s_i = \left(\frac{d}{dt} + \lambda\right)e_i \quad (1.2)$$

Où e est l'erreur de suivi pour chaque joint et λ est une constante positive. Par définition, les surfaces de glissement représentent la réponse fréquentielle d'un système linéaire passe-bas de premier ordre avec une fréquence de coupure $fc = \frac{\lambda_i}{2\pi}$. En conséquence, l'objectif de ce schéma de commande est de ramener les variables commandables du système le plus proche possible de cette surface de glissement, c'est-à-dire $s_i = 0$ [10]. Comme constaté à la figure (1.2)[11], des commutations à haute fréquence représentant le phénomène dit de broutement (chattering) peuvent être présentes dans la dynamique du système. Pour minimiser ce phénomène, une couche limite est définie pour lisser la discontinuité du signal de commande autour de la surface de glissement [3].

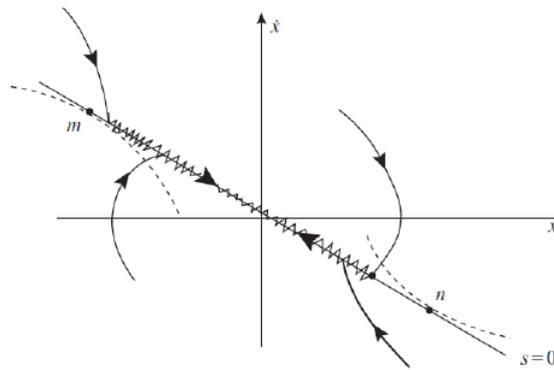


FIGURE 1.2 – Surface de glissement[11]

Cette approche de commande produit de bons résultats en présence de charges variables et/ou de dynamiques inconnues non modélisées et est beaucoup plus robuste que les contrôleurs susmentionnés. À l'inverse, les contrôleurs MG connaissent constamment des broutements et des pertes d'énergie élevées et sont généralement plus difficiles à réaliser que le PID à cause de leur complexité [3].

1.2 Commande coordonnée

Les systèmes de commande coordonnée tentent d'interconnecter le mouvement de chaque actionneur et de relier son comportement au comportement du reste des dispositifs du système. Cette coordination peut être une corrélation simple des trajectoires des différents agents comme elle peut prendre la forme d'une influence collective sur le signal de contrôle de chaque agent. Un schéma illustrant ce type de commande est représenté dans la figure(1.3).L'application des approches de commande décentralisées pour robots les rend plus «conscients» de leur structure entière, ce qui affecte positivement leur performance [12].

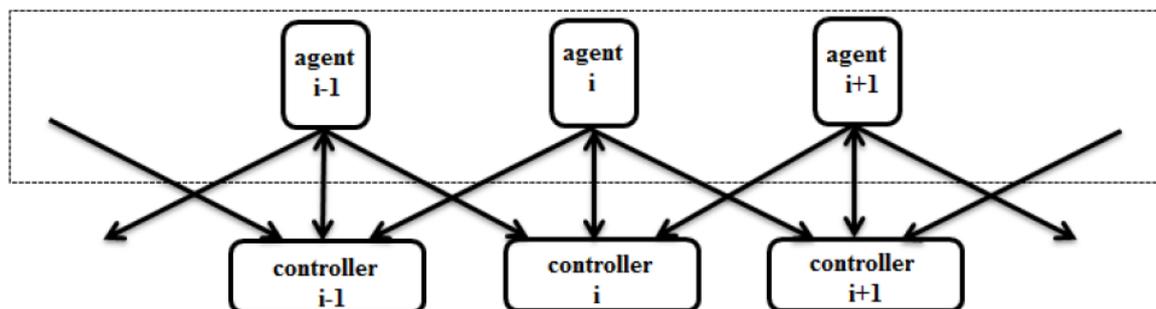


FIGURE 1.3 – Contrôle coordonné [7]

1.2.1 Commande maître / esclave

Les premières tentatives d'élaboration de commandes coordonnées en robotique ont porté sur le contrôle maître-esclave de deux manipulateurs robotiques [13]. Au niveau articulaire du robot, un autre contrôleur maître-esclave a été introduit [14] où un chemin souhaité peut être défini pour réaliser la tâche du maître et une position relative est utilisée pour commander le mouvement de l'esclave. Pour coordonner les deux mouvements, le contrôleur utilise la position actuelle du maître et définit le changement requis dans la commande du mouvement de l'esclave via une matrice de prédiction d'erreur. La coordination du système est effectuée par un ordinateur dédié à cette tâche, et qui est responsable de recevoir les données de position des encodeurs. Une approche similaire a été adoptée par [16] qui ont utilisé un Contrôleur de type PD + feedforward. La trajectoire du maître est librement définie, et l'entrée de commande du mouvement de l'esclave est basée sur la position réelle du maître.

$$\tau_s = M_s(q_s)\dot{q}_{mm} + C_s(q_s, \dot{q}_s) \quad (1.3)$$

Où : M et C et g sont des paramètres dynamiques du manipulateur, K_d et K_p sont des gains de réglage, et les indices m et s se réfèrent aux mouvements du maître et de l'esclave.

On remarque que la position actuelle du mouvement maître est incorporée dans le terme feedforward du contrôleur et l'erreur du système esclave est utilisée dans le segment PD. Dans les cas où la position et l'erreur ne peuvent pas être mesurées directement, les observateurs peuvent être utilisés pour estimer leurs valeurs [16]. L'inconvénient majeur de ce type de contrôleur est leur concentration sur une formation de plusieurs robots ainsi que leur indépendance complète de la coordination des mouvements communs entre eux. En outre, la prise en considération du mouvement esclave comme relatif et proportionnel au mouvement du maître crée des difficultés lorsqu'il s'agit de trajectoires et de tâches compliquées [7].

1.2.2 Commande à événements orientés (Event Based Control)

Une alternative à l'échantillonnage périodique des systèmes dynamiques par des capteurs est fondée sur le contrôle à base d'événements. Au lieu d'échantillonner l'état du système sur un intervalle de temps défini, les systèmes orientés événements échantillonnent l'état uniquement lorsque le signal de mesure traverse un niveau de seuil prédéfini [17].

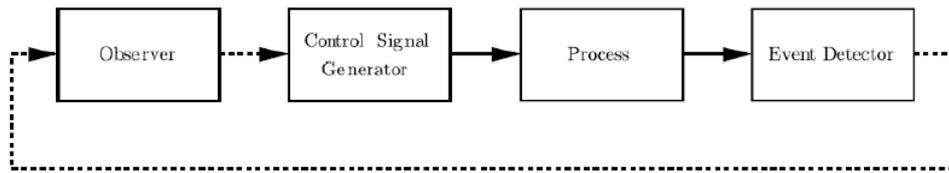


FIGURE 1.4 – Schéma de la commande orientée événements[7]

Plus précisément, lors du fonctionnement d'un système, le détecteur d'événement génère un signal lorsque cet événement se produit, et l'envoie à l'observateur responsable des estimations alimentant le générateur de signal de commande [18]. Il est clair que si le signal de mesure ne franchit pas le seuil, la rétroaction ne se matérialise pas pour cette mesure. En raison de sa nature, ce contrôleur réagit plus rapidement aux changements subis par le système que les correcteurs déclenchés par échantillonnage temporel périodique. Aussi, les systèmes orientés événements nécessitent moins de puissance de traitement que les systèmes à échantillonnage périodique, ce qui les rend très attractifs pour des situations où des performances de calcul élevées sont requises

1.3 Autres techniques de commande

1.3.1 Commande répétitif

Le contrôle répétitif (CR) similaire à la commande par apprentissage itératif (Iterative Learning Control [ILC]) utilise également la répétitivité du processus pour atteindre la stabilisation des performances du système. Proposé par [19], la commande répétitive peut être exprimé comme suit :

Où $u(t)$ est l'entrée du système et $e(t)$ est l'erreur de sortie du système.

Contrairement à la CAI, la CR est utilisée dans des processus périodiques et continu pour un meilleur suivi ou rejet de signaux exogènes. Pour cela, au lieu de conserver les mêmes conditions initiales pour chaque cycle, le correcteur répétitif définit les valeurs finales du cycle précédent comme les nouvelles conditions initiales pour la nouvelle itération [20].

1.3.2 Commande par apprentissage itératif CAI(ILC)

Le contrôle par apprentissage itératif (CAI/ILC) consiste à affiner les performances du système robotique pour une tâche spécifique par répétition. La CAI découle du fait que la majorité des systèmes industriels doivent effectuer à plusieurs reprises les mêmes tâches ou des tâches similaires.

Pendant le processus, les signaux d'entrée et de sortie du système sont enregistrés dans la mémoire du système. L'erreur entre les deux signaux est utilisée pour construire le signal d'entrée pour le nouveau cycle. Contrairement aux contrôleurs conventionnels qui tentent de diminuer l'erreur du système au fur et à mesure que le temps augmente, la CAI diminue l'erreur du système lorsque le nombre d'itérations augmente.

Principe de la CAI/ILC

Considérons un système dynamique non linéaire fonctionnant d'une façon répétitive sur un intervalle de temps fixe $[0, T]$ et soit $y_d(t)$ la sortie désirée. L'équation d'état décrivant le système à la k^{ime} itération est donnée par :

$$\begin{cases} x_k(t) = f(x_k(t), u_k(t)) \\ y_k(t) = g(x_k(t), u_k(t)) \end{cases} \quad (1.4)$$

Où : $x_k(t)$, $y_k(t)$ et $u_k(t)$ sont respectivement l'état, la sortie et la commande du système. L'erreur de poursuite à la k^{ime} itération est définie par :

$$e_k(t) = y_d(t) - y_k(t) \quad (1.5)$$

Postulats

P1 : Le temps que dure chaque itération (cycle, itération, pass...) est fixe, et est noté T , tel que $T > 0$.

P2 : La sortie désirée $y_d(t)$ est connue a priori pour $t \in [0, T]$.

P3 : A chaque début d'un cycle, la condition initiale sur l'état est vérifiée (répétitivité de l'état initial), c-à-d : à chaque itération $x_k(0) = x^0$ pour $k = 1, 2, 3, \dots$

P4 : L'invariance de la dynamique du système est assurée tout au long de ces itérations.

P5 : La sortie $y_k(t)$ est mesurable et par conséquent, l'erreur de poursuite peut être utilisée dans le calcul de la loi de commande de l'itération suivante c-à-d $u_{k+1}(t)$.

P6 : le système est inversible, c-à-d, il existe une entrée unique $u_d(t)$ qui peut amener le système à produire la sortie désirée $y_d(t)$.

Forme générale d'une commande CAI/ILC

Une loi de la CAI se présente sous la forme générale suivante :

$$u_{k+1}(t) = f(u_k(t), e_{k+1}(t), e_k(t), e_{k-m}(t)) \quad (1.6)$$

Avec : $m \geq 1$

Il convient de noter que f peut dépendre à la fois des erreurs, et/ou de leurs dérivées, et/ou de

leurs intégrales obtenues lors des différents cycles[22]. L'algorithme linéaire, le plus général, trouvé en bibliographie est[21] :

$$u_{k+1}(t) = u_k(t) + \phi e_k(t) + \gamma \dot{e}_k(t) + \psi \int e_k(t) dt \quad (1.7)$$

Cet algorithme montre que la commande répétitive est de type *PID*.

Le problème fondamental de cette approche de commande réside dans détermination d'une forme récursive f , cependant, l'exploit ne consiste pas seulement en l'obtention d'une telle forme récursive, mais aussi en la garantie de la convergence de l'erreur à une vitesse satisfaisante .

1.3.3 Objectifs de la CAI

L'objectif principal de la CAI est d'atteindre des performances désirées (stabilité et convergence) et de les maintenir (robustesse) en présence d'incertitudes dans le système à commander.

Il est démontré par la suite que la condition de stabilité d'un système sous la loi de commande CAI sont établies en terme de certaines normes du système. Ainsi, la stabilité du système (1.4)est assurée si[22] :

$$\|e_{k+1}(t)\| \leq \|e_k(t)\| \quad (1.8)$$

Dans un sens plus strict si :

$$\|e_{k+1}(t)\| \leq \theta \|e_k(t)\| \quad (1.9)$$

Remarques Le postulat *P3* se trouve parfois non vérifié, dans ce cas il s'agit d'une commande répétitive. La seule différence est que la commande CAI nécessite une initialisation sur l'état par la même condition pour chaque cycle, tandis que pour la commande répétitive dont la trajectoire de référence est supposée périodique, la condition initiale pour le cycle courant est égale à la condition finale du cycle précédent.

Afin d'évaluer la stabilité du système commandé ainsi que la robustesse de la loi de commande d'une façon plus réaliste, les effets des erreurs de modélisation et les bruits de mesure sur la sortie du système sont considérés. Par conséquent, la version révisée des postulats (*P2P5*)est[22] :

P'2 : La sortie désirée $y_d(t)$ est connue a priori pour $t \in [0, T]$ peut varier d'un cycle à l'autre, mais à condition que cette variation soit bornée.

P'3 : A chaque début d'un cycle, la condition initiale suivante est vérifiée :

$$x_k(0) = x^0 + \delta_k \quad (1.10)$$

Avec : $\|\delta_k\| \leq b_{x0}$

P'4 :L'invariance de la dynamique du système peut ne pas être assurée, on suppose donc qu'il existe une erreur de modélisation inconnue mais bornée en norme.

$$x_k(0) = x^0 + \delta_k \quad (1.11)$$

P'5 :A chaque itération, la sortie du système est supposée mesurable mais peut inclure un bruit de mesure spécifié.

Ainsi, la loi de la CAI est dite robuste, si sous les considérations (P2' à P5', la stabilité du système reste garantie. CAI en boucle ouverte / CAI en boucle fermée) Afin de bien illustrer les concepts liés à la CAI en boucle ouverte et en boucle fermée, considérons le schéma bloc de la figure (1. ?) traduisant la CAI suivante :

$$u_{k+1}(t) = u_k(t) + \phi_{k+1}(t) * e_{k+1}(t) + \phi_k(t) * e_k(t) + \dots + \phi_{k-m}(t) * e_{k-m}(t) \quad (1.12)$$

Avec : $m \geq 1$

D'après le schéma, la loi précédente peut être écrite sous la forme :

$$u_{k+1}(t) = u_{ff}(t) + u_{fb}(t) \quad (1.13)$$

$u_{ff}(t)$: la partie du contrôleur due au "feed forward", la réaction en amont par les cycles passés.

$u_{fb}(t)$: la partie du contrôleur due au "current cycle feedback", le retour par cycle courant[23].

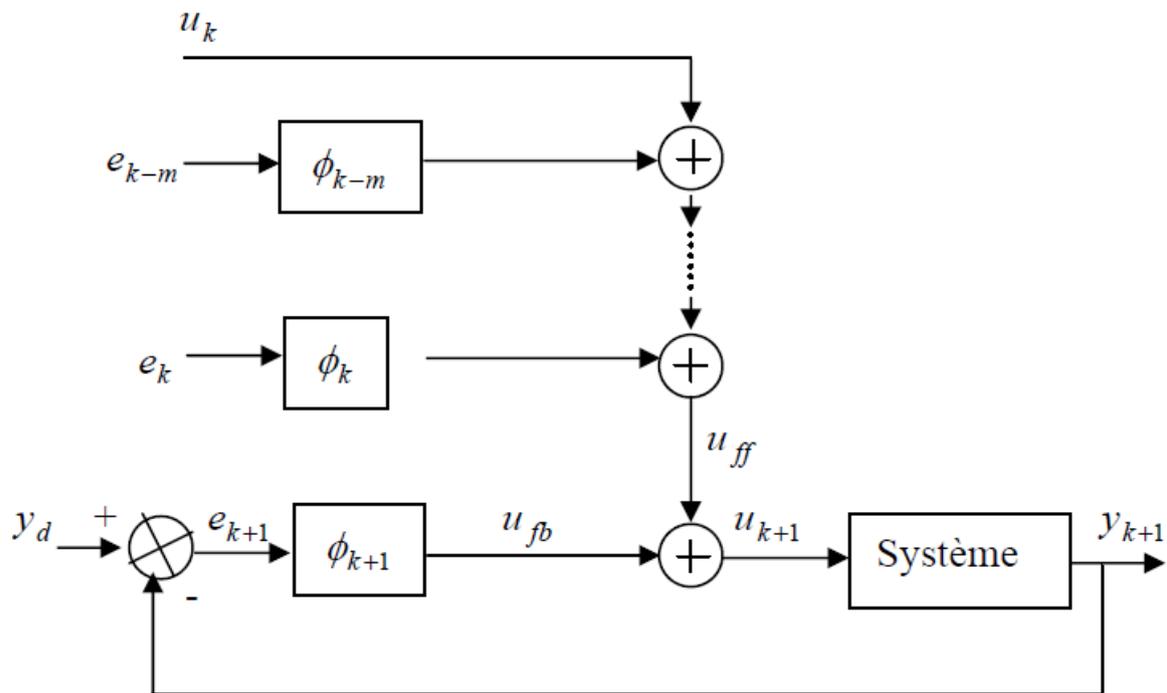


FIGURE 1.5 – Schéma bloc de la CAI [24]

CAI en boucle ouverte (off-line) Ce type de CAI est obtenu en posant le gain $\phi_{k+1}(t)$ de l'équation (1.12) égal à zéro, la loi de commande CAI devient[24] :

$$u_{k+1}(t) = u_{ff}(t) \quad (1.14)$$

$$u_{k+1}(t) = u_k(t) + \phi_k(t) * e_k(t) + \dots + \phi_{k-m}(t) * e_{k-m}(t) \quad (1.15)$$

Ce schéma de commande est souvent utilisé mais requiert que le système à commander soit stable en boucle ouverte[25].

CAI en boucle fermée (on-line) Pour pallier aux inconvénient de la commande CAI en boucle ouverte, un autre schéma dit en boucle fermée a été proposé, il est obtenu en ajoutant au "schéma off-line" une contre réaction par le cycle courant, ainsi $\phi_{k+1}(t) \neq 0$ et la loi de commande s'exprime comme suit :

$$u_{k+1}(t) = u_{ff}(t) + u_{fb}(t) \quad (1.16)$$

$$u_{k+1}(t) = u_k(t) + \phi_{k+1}(t) * e_{k+1}(t) + \phi_k(t) * e_k(t) + \dots + \phi_{k-m}(t) * e_{k-m}(t) \quad (1.17)$$

1.3.4 Commande par Forwarding

La commande par Forwarding est une commande similaire au Backstepping dont l'idée générale est de construire une fonction de Lyapunov , ainsi que la commande d'une manière récursive et cela pour un système qui peut être écrit sous la forme suivante [26], [27], [28], [29] :

$$\begin{cases} \dot{x}_1 = f_1(x_2, \dots, x_n) + g_1(x_2, \dots, x_n)u \\ \dot{x}_2 = f_2(x_3, \dots, x_n) + g_2(x_3, \dots, x_n)u \\ \vdots \\ \dot{x}_{n-1} = f_{n-1}(x_n) + g_{n-1}(x_n)u \\ \dot{x}_n = u \end{cases} \quad (1.18)$$

L'objectif est de déterminer une loi de commande qui assure la stabilisation globale du système (1.18). Cela s'effectue en plusieurs étapes.

En premier lieu, nous essayons de stabiliser x_n à travers u . En prenant une fonction de Lyapunov V_n qui est définie positive sur \mathfrak{R} et on cherche à trouver un retour d'état $u = \alpha_n(x_n)$ pour stabiliser x_n .

L'étape suivante consiste à augmenter le système, de x_n à x_{n-1} , et remplacer la commande u par $u = \alpha_n(x_n) + v_{n-1}$. En prenant une fonction de Lyapunov V_{n-1} définie positive sur \mathfrak{R}^2 , on cherche à trouver un retour d'état $v_{n-1} = \alpha_{n-1}$ qui permet de stabiliser le système augmenté. On refait la même procédure jusqu'à la dernière étape, où une commande u qui permet d'avoir la stabilité globale du système.

1.3.5 Les étapes de la synthèse

La synthèse de la loi de commande pour le système (1.18) s'effectue en plusieurs étapes [26], [28], [29] :

Première étape

On prend le système :

$$\dot{x}_n = u \quad (1.19)$$

Soit la fonction de Lyapunov : $V_n = \frac{1}{2}x_n^2$ définie positive sur \mathfrak{R} .

Sa dérivée est donnée par : $\dot{V}_n = x_n u$.

Pour rendre \dot{V}_n définie négative, on peut prendre :

$$u = \alpha_n(x_n) = -\lambda L_g V_n = -\lambda_n x_n \quad \text{avec} \quad \lambda_n \geq 0 \quad (1.20)$$

Ce qui donne $\dot{V}_n = -\lambda_n x_n^2$ qui est définie négative sur \mathfrak{R} .

Deuxième étape

Soit le système augmenté :

$$\begin{cases} \dot{x}_{n-1} = f_{n-1}(x_n) + g_{n-1}(x_n)u \\ \dot{x}_n = u \end{cases} \quad (1.21)$$

Pour $u = \alpha_n(x_n)$, le système (1.21) devient :

$$\begin{cases} \dot{x}_{n-1} = \phi_{n-1}(x_n) \\ \dot{x}_n = -\lambda_n x_n \\ \phi_{n-1}(x_n) = f_{n-1}(x_n) - \lambda_n g_{n-1}(x_n)x_n \end{cases} \quad (1.22)$$

Le terme ϕ_{n-1} est appelé le terme d'interconnexion.

La solution dans l'espace 's' [27], du système précédent sera :

$$\begin{cases} \tilde{x}_n(s) = x_n(t)e^{-\lambda_n s} \\ \tilde{x}_{n-1}(s) = \int_0^t (f_{n-1}(\tilde{x}_n) - \lambda_n g_{n-1}(\tilde{x}_n)\tilde{x}_n) ds + x_{n-1}(t) \end{cases} \quad (1.23)$$

La nouvelle entrée de commande pour le système (1.21) sera [26], [27] :

$$u = \alpha_n(x_n) + v_{n-1} \quad (1.24)$$

La fonction de Lyapunov pour le système (1.21) est donnée par [27] :

$$\begin{aligned} V_{n-1} &= V_n + \frac{1}{2}x_{n-1}^2 + \int_0^\infty \tilde{x}_{n-1}(t)\phi_{n-1}(\tilde{x}_{n-1})ds \\ &= V_n + \frac{1}{2} \lim_{t \rightarrow \infty} \tilde{x}_{n-1}^2(s) \end{aligned} \quad (1.25)$$

Sa dérivée est [27] :

$$\begin{aligned} \dot{V}_{n-1} &= \dot{V}_{n-1}|_{u=\alpha_n(x_n)} + (L_g V_{n-1})v_{n-1} \\ &= -\lambda_n(L_g V_n)^2 + (L_g V_{n-1})v_{n-1} \end{aligned} \quad (1.26)$$

Donc pour rendre \dot{V}_{n-1} définie négative, on peut prendre :

$$v_{n-1} = -\lambda_{n-1}L_g V_{n-1} \quad \text{avec} \quad \lambda_{n-1} \geq 0 \quad (1.27)$$

Dans ce cas là, on aura : $\dot{V}_{n-1} = -\lambda_n(L_g V_n)^2 - \lambda_{n-1}(L_g V_{n-1})^2$. Cette fonction est définie négative sur \mathfrak{R}^2 . La commande qui assure la stabilisation globale du système (1.21) est alors [27] :

$$u = \alpha_{n-1}(x_n, x_{n-1}) = \alpha_n(x_n) + v_{n-1} \quad (1.28)$$

L'étape n

Dans cette étape, on considère le système complet décrit par l'équation (1.18). Pour $u = \alpha_2(x_2, \dots, x_n)$ (calculée dans l'étape (n -1)), le système (1.29) devient :

$$\left\{ \begin{array}{l} \dot{x}_1 = \phi_1(x_2, \dots, x_n) \\ \dot{x}_2 = f_2(x_3, \dots, x_n) + g_2(x_3, \dots, x_n)\alpha_2(x_2, \dots, x_n) \\ \vdots \\ \dot{x}_{n-1} = f_{n-1}(x_n) + g_{n-1}(x_n)\alpha_2(x_2, \dots, x_n) \\ \dot{x}_n = \alpha_2(x_2, \dots, x_n) \end{array} \right. \quad (1.29)$$

Avec le terme d'interconnexion :

$$\phi_1(x_2, \dots, x_n) = f_1(x_2, \dots, x_n) + g_1(x_2, \dots, x_n)\alpha_2(x_2, \dots, x_n) \quad (1.30)$$

On note $\tilde{x}_i(t)$ pour $i \in [1, n]$ la solution temporelle du système (1.29). La nouvelle entrée de commande pour (3.38) sera [30] :

$$u = \alpha_2(x - 2, \dots, x_n) + v_1 \quad (1.31)$$

La fonction de Lyapunov est alors [27] :

$$\begin{aligned} V_1 &= V_2 + \frac{1}{2}x_1^2 + \int_0^\infty \tilde{x}_1(s)\phi_1(\tilde{x}_2(t), \dots, \tilde{x}_n(s))ds \\ &= V_2 + \frac{1}{2} \lim_{s \rightarrow \infty} \tilde{x}_1^2(s) \end{aligned} \quad (1.32)$$

Sa dérivée est [27] :

$$\begin{aligned} \dot{V}_1 &= \dot{V}_1|_{u=\alpha_2} + (L_g V_1)v_1 \\ &= -\lambda_2(L_g V_2)^2 + (L_g V_1)v_1 \end{aligned} \quad (1.33)$$

Donc pour rendre \dot{V}_1 définie négative, on peut prendre le retour d'état non linéaire suivant :

$$v_1 = -\lambda_1 L_g V_1 \quad \text{avec} \quad \lambda_1 \geq 0 \quad (1.34)$$

Il en résulte :

$$\dot{V}_1 = -\lambda_2(L_g V_2)^2 - \lambda_1(L_g V_1)^2 \quad (1.35)$$

Cette fonction est définie négative donc la stabilité globale du système (1.18) est assurée. La commande finale est [27] :

$$u = \alpha_1(x_1, \dots, x_n) = \alpha_2(x_2, \dots, x_n) + v_1 \quad (1.36)$$

Conclusion

Dans ce chapitre, nous avons donné un aperçu sur différentes techniques et méthodes de commande de robots développées par la communauté scientifique. Le suivi de contour constitue l'objectif de commande le plus étudié pour les processus d'usinage. Une classification générale des types de contrôleurs a été donnée selon leur architecture centralisée ou coordonnée.

CHAPITRE 2

LA COMMANDE DANS LE DOMAINE DE POSITION

Chapitre 2 : La commande dans le domaine de position

Introduction

Dans ce chapitre, nous explorons les principes de la commande dénommée commande dans le domaine de position. Le modèle dynamique d'un système robotisé est transformé de l'espace temporel au domaine de position.

2.1 Erreur de poursuite et erreur de contour

La réponse dynamique d'un robot industriel asservi se caractérise par deux paramètres importants : la poursuite axiale de chaque moteur et la performance du suivi de contour dans le plan qui se traduit par les deux grandeurs : (l'erreur de poursuite et l'erreur de suivi de contour).

L'erreur de suivi ou de poursuite résulte d'une performance de suivi axiale individuelle, elle reflète la manière dont le contrôleur de chaque axe suit la position de référence et / ou la vitesse souhaitée. En revanche, une erreur de contour est définie comme l'erreur entre le contour de référence désiré et le contour réel dans une direction orthogonale. Celle-ci peut être interprétée comme étant la distance minimale entre le contour désiré et le contour parcouru par le système commandé. La différence entre l'erreur de suivi et l'erreur de contour peut être illustrée à travers la figure (2.1).

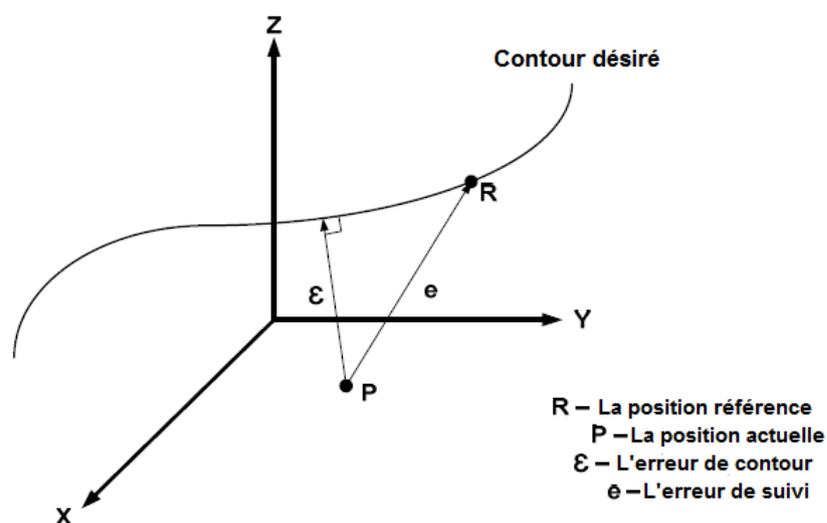


FIGURE 2.1 – Schéma de commande dans le domaine de position [7]

D'après les définitions précédentes, on peut voir que l'erreur de suivi de contour peut être considérée comme une combinaison de toutes les erreurs de suivi axiales dans un système poly-articulé. Diverses recherches [1] [30] [31] [32] ont indiqué que bien que des contrôles avancés ont été développés afin d'améliorer les performances de suivi, cela n'a pas nécessairement garanti une bonne performance de suivi de contour. Un exemple de cette conclusion est illustré à la figure (2.2). À partir de cette figure, les erreurs de suivi axiales (e_x et e_y) ont été considérablement réduites de la position actuelle P jusqu'à P''. Cependant, la variation de l'erreur de contournement ε à ε'' reste relativement faible.

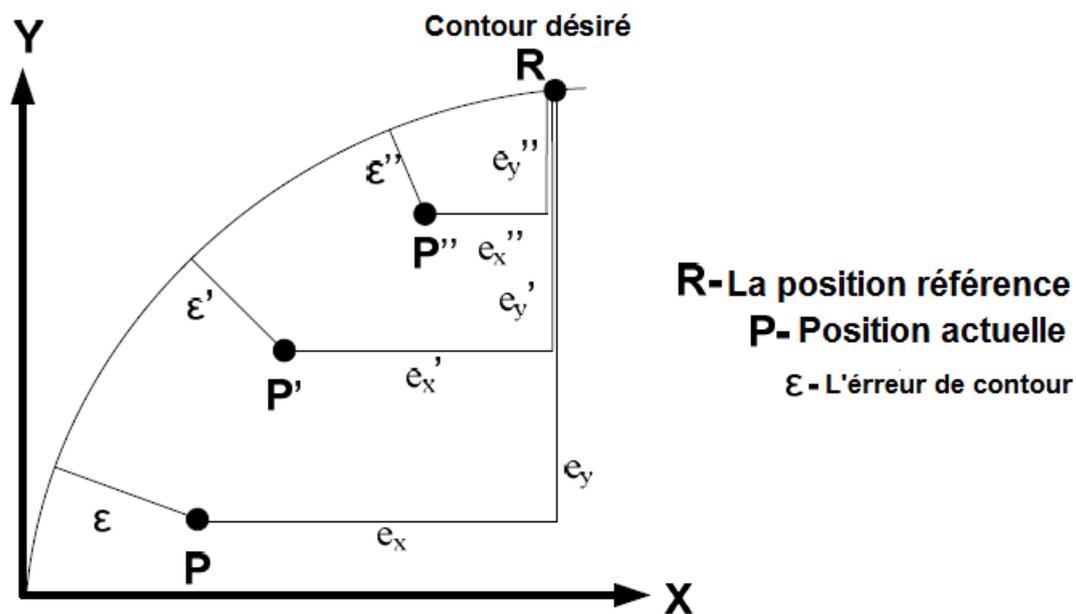


FIGURE 2.2 – L'erreur de poursuite et l'erreur de contour [33]

En conséquence, l'erreur de contour est considérée un indice plus important que les erreurs de poursuite dans la mesure de la qualité d'un produit usiné ou de la performance du suivi de contour. Diverses méthodes sont définies pour le calcul ou l'estimation de cette erreur selon que la forme du contour soit linéaire, non linéaire ou indéfinie. On exposera dans les paragraphes qui suivent les éléments de calcul de l'erreur de contours tels que définis par Koren.Y dans[33] .

2.2 Calcul de l'erreur de contour

2.2.1 Contour linéaire

Pour ce type de contour, l'erreur de contour peut être déterminée par la relation géométrique représentée dans la figure suivante et qui peut être exprimée comme suit :

$$\varepsilon = -E_x \sin(\theta) + E_y \cos(\theta) \quad (2.1)$$

Puisque : $\sin(\theta) = \frac{V_y}{V}$ et $\cos(\theta) = \frac{V_x}{V}$, (avec V : la vitesse désirée, l'équation précédente donne :

$$\varepsilon = \frac{-E_x V_y + E_y V_x}{V} \quad (2.2)$$

Cette équation a été introduite dans [1]. La comparaison de la figure donnée avec l'équation (2.1) donne les gain de couplage suivants :

$$C_x = \frac{V_y}{V} \quad \text{et} \quad C_y = \frac{V_x}{V} \quad (2.3)$$

En fait, les composantes V_x et V_y de la vitesse axiale dépendent de la pente de la coupe linéaire et sont ajustées au début de chaque segment.

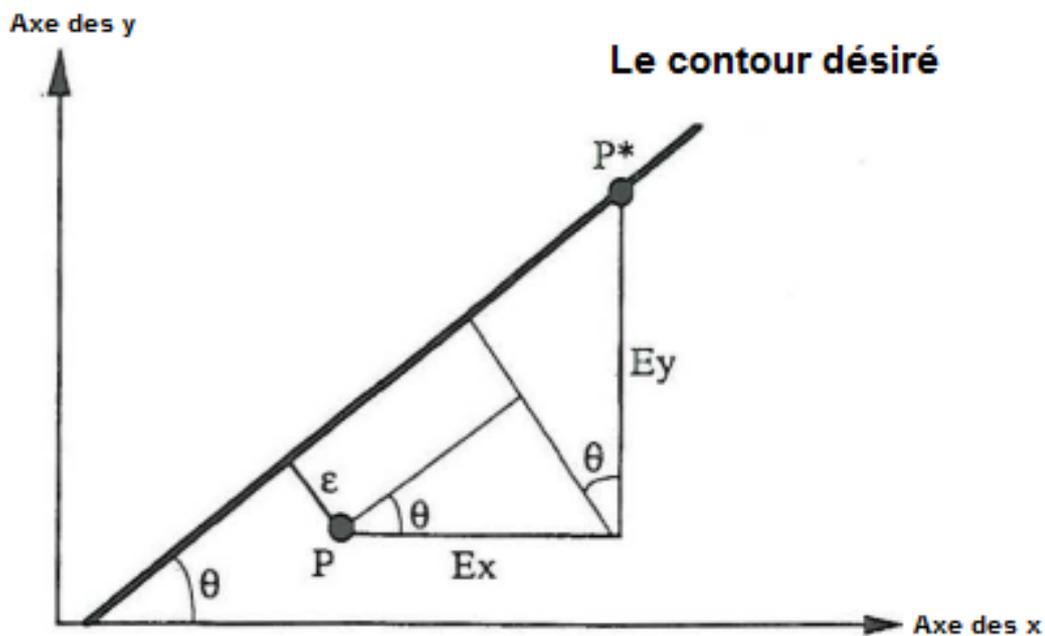


FIGURE 2.3 – Modèle de l'erreur pour un contour linéaire [1]

2.2.2 Contour circulaire

L'erreur de suivi de contour circulaire est la différence entre la distance séparant la position de l'organe terminal du centre du cercle et le rayon du cercle :

$$\varepsilon = \sqrt{(P_x - x_0)^2 + (P_y - x_0)^2} - R \quad (2.4)$$

Où : R est le rayon du contour circulaire

(x_0, y_0) correspond au coordonnées cartésiennes du centre du contour. (P_x, P_y) la position de l'organe terminal, qui peut être représenté en fonction de l'erreur de poursuite axiale et la position de référence par l'équation suivante :

$$P_x = R_x - E_x = R \sin \theta + x_0 - E_x \quad (2.5)$$

$$P_y = R_y - E_y = -R \cos \theta + y_0 - E_y \quad (2.6)$$

En remplaçant les équations (2.5) et (2.6) dans l'équation (2.4), on obtient :

$$\varepsilon = \sqrt{(R \sin \theta - E_x)^2 + (R \cos \theta - E_y)^2} - R \quad (2.7)$$

Étant donné que l'implémentation de l'équation (2.7) dans un système de commande en temps réel est relativement difficile, une simplification est proposée. Ainsi, le développement en série de Taylor à l'ordre 1 de l'erreur dans l'équation (2.7) donne :

$$\varepsilon = -E_x \left(\sin \theta - \frac{E_x}{2R} \right) + E_y \left(\cos \theta - \frac{E_y}{2R} \right) + O\left(\frac{\varepsilon^2}{R}\right) \quad (2.8)$$

A cet effet, si l'erreur de contour est très petite devant l'erreur de suivi axial, et si les erreurs axiales sont aussi petites devant le rayon du contour, le terme d'erreur d'ordre supérieur peut donc être négligé et l'erreur de contour est approximée par :

$$\varepsilon = -E_x C_x + E_y C_y \quad (2.9)$$

Avec : C_x et C_y définis par :

$$C_x = \sin \theta - \frac{E_x}{2R} \quad (2.10)$$

$$C_y = \cos \theta + \frac{E_y}{2R} \quad (2.11)$$

Il convient de noter que R est constant et que les gains sont calculés à chaque pas d'interpolation.

Dans ce calcul, on exploite les signaux d'entrée d'interpolation $\sin(\theta)$ et $\cos(\theta)$ ainsi que ceux des erreurs de poursuite axiale E_x et E_y . Ces signaux varient pendant l'opération, les équations (2.10) et (2.11) expriment le nouveau concept de gains variables.

Et la longueur de la projection de la distance (c'est-à-dire l'erreur de suivi) peut être calculée par :

$$L = \frac{\overrightarrow{r_2 r_1(t_k)} \cdot \overrightarrow{e_2(t_k)}}{\|\overrightarrow{e_2(t_k)}\|} \quad (2.15)$$

Supposons maintenant qu'un point $r_1(t'_k)$ existe sur le contour souhaité et la distance entre ce point et $r_2(t_k)$ est l'erreur de contour réelle. En supposant de plus que l'effecteur final a une vitesse désirée \vec{V} , le temps requis pour couvrir la distance L est :

$$\Delta t = \frac{L}{|\vec{V}|} \quad (2.16)$$

Celui-ci est également le temps requis pour se déplacer de $r_1(t'_k)$ à $r_1(t_k)$. D'où l'on peut retrouver le point inconnu $r_1(t'_k)$ par :

$$r_1(t'_k) = r_1(t_k - \Delta t) \approx r_1(t_k) - V \cdot \Delta t \quad (2.17)$$

Par conséquent, l'erreur de contour peut être calculé de manière similaire à celle correspondant à un contour linéaire :

$$e_c = -e_x \sin \phi + e_y \cos \phi \quad (2.18)$$

Où e_x et e_y sont les erreurs de suivi des axes cartésiens ϕ est est l'angle de référence entre le vecteur $\overrightarrow{r_1(t'_k)r_1(t_k)}$ et l'axe des abscisses x, calculé comme suit :

$$\phi = \arctan\left(\frac{r_{1y}(t'_k) - r_{1y}(t_k)}{r_{1x}(t'_k) - r_{1x}(t_k)}\right) \quad (2.19)$$

En ce qui concerne la vitesse V , si l'effecteur maintient une vitesse constante, $V_1(t'_k)$ est alors égale à $V_1(t_k)$. Généralement, l'effecteur d'un manipulateur robotique subit des variations significatives dans sa vitesse. Ainsi, une estimation peut être utilisée de sorte à définir la vitesse comme la moyenne des vitesses souhaitée $V_1(t_k)$ et réelle $V_2(t_k)$ à chaque instant t_k . Ceci est donc la raison principale pour laquelle la valeur d'erreur de contour résultante est considérée comme une estimation et non un calcul exact [34] .

2.3 Génération de trajectoire

2.3.1 Polynôme d'ordre 5

Pour asservir les diverses structures de robots qu'on traitera dans la suite de ce travail, on considère un problème de suivi de trajectoire, où la position souhaitée varie avec le temps. Dans ce cas, on ne peut pas simplement imposer une équation de vitesse de premier ordre, ce qui nécessiterait un signal de commande important pour satisfaire la loi de

commande, entraînerait un suivi inexact, et nuirait aux actionneurs lors de l'implémentation.

Un tel problème peut être résolu en choisissant une équation polynomiale d'ordre supérieur pour la planification de la trajectoire. On peut littéralement choisir l'ordre du polynôme aussi élevé que désiré pour obtenir une trajectoire lisse, qui assure un changement continu et graduel de la position, de la vitesse et de l'accélération. Ainsi, une fonction polynomiale du 5ème ordre du temps se révèle suffisante pour la simulation du système commandé.

Pour garantir que les trajectoires de référence soient lisses pour toutes les articulations du robot manipulateur, un polynôme d'ordre 5 est utilisé pour définir le contour désiré.

$$r(t) = 10\left(\frac{t}{T}\right)^3 - 15\left(\frac{t}{T}\right)^4 + 6\left(\frac{t}{T}\right)^5 \quad (2.20)$$

Où : t : le temps et T : la durée totale accordée à l'opération ou au mouvement .Puis, la vitesse et l'accélération sont donnés par :

$$\dot{r}(t) = \frac{1}{T} \left\{ 30\left(\frac{t}{T}\right)^2 - 60\left(\frac{t}{T}\right)^3 + 30\left(\frac{t}{T}\right)^4 \right\} \quad (2.21)$$

$$\ddot{r}(t) = \frac{1}{T^2} \left\{ 60\left(\frac{t}{T}\right) - 180\left(\frac{t}{T}\right)^2 + 120\left(\frac{t}{T}\right)^3 \right\} \quad (2.22)$$

2.3.2 Contours Linéaires

Avec les équations (2.21)-(2.22), un contour linéaire qu'on désire imposer au niveau de l'effecteur peut être défini par :

$$P_d = (P_f - P_i)r(t) \quad (2.23)$$

Où : P_d est le contour désiré, exprimé par des coordonnées cartésiennes (x_{ef}, y_{ef}) , et P_i et P_f sont les positions initiale et finale du contour.

Dans le cas de la planification d'une tâche pour un robot manipulateur, le modèle cinématique inverse donné en [annexe 1] est utilisé pour calculer la position et la vitesse articulaires de référence assurant la poursuite du contour désiré. L'orientation q_{ef} de l'effecteur est maintenue constante pour les simulation .

2.3.3 Contours non linéaires

Pour définir le contour non linéaire, un paramètre angulaire doit être introduit :

$$\theta(t) = r(t)(\theta_f - \theta_i) \quad (2.24)$$

2.3.4 Contour circulaire

Le contour circulaire est défini comme suit :

$$\begin{cases} x_{ef}(t) = x_c + R \cdot \cos(\theta(t)) \\ y_{ef}(t) = y_c + R \cdot \sin(\theta(t)) \end{cases} \quad (2.25)$$

2.4 Commande dans le domaine de position

Dans une synthèse de commande dans le domaine des positions, un système à plusieurs agents est subdivisé entre des agents maître et esclaves. Le mouvement dynamique de l'agent maître est échantillonné de façon équidistante et utilisé en tant que référence indépendante, de la même manière que le temps est utilisé dans le contrôle dans le domaine temporel. Les mouvements des agents esclaves sont définis comme des fonctions du mouvement principal et les exigences de contour de désiré.

La formulation de ces fonctions nécessite le développement du modèle dynamique du système dans le domaine des positions grâce à une transformation du domaine temporel vers le domaine de position ainsi que de sa représentation en tant que système maître-esclave. L'avantage majeur de la commande dans le domaine de position est notamment l'élimination de l'erreur de suivi de la trajectoire de référence pour l'élément maître, ce qui minimise la contribution de la dynamique de l'agent maître à l'erreur de contour du système global. Ceci s'explique par le fait que l'actualisation des calculs du correcteur développé dans le domaine de position est synchronisée avec l'échantillonnage des données relatives au mouvement de la variable maître. Par conséquent, afin de pouvoir commander le modèle esclave dans le domaine de position d'une manière efficace, la réponse temporelle du maître doit être mesurée avec une précision aussi élevée que possible.

2.4.1 Dérivation relative et transformation du domaine temporel vers le domaine de position

L'objectif d'une commande dans le domaine de position est de remplacer la dépendance temporelle du système par une dépendance à l'une de ses variables appelé variable de référence ou axe maître et notée " q_m " avec $q_m = q_1$. Les mouvements des autres variables, appelées esclaves, peuvent être représentés en fonction du mouvement de l'axe maître comme suit : $q_i = q_i(q_m), i = 2, 3, \dots, n$.

La première étape pour transformer un système dynamique en son équivalent dans le domaine de position est de développer une relation qui relie le domaine de position au domaine temporel. Cela se fait en présentant la dynamique dérivée du mouvement de l' i ème esclave

par rapport au mouvement de l'agent maître :

$$\dot{q}'_i = \frac{dq_i}{dq_m} = \frac{\dot{q}_i}{\dot{q}_m} \quad (2.26)$$

De l'équation (2.26) on comprend aisément que q' est le rapport de vitesse entre l'esclave et les vitesses de l'agent principal q_m qui décrit une relation de synchronisation entre les deux mouvements. Cette dérivée relative est appelée vitesse de position relative de l'agent par rapport à l'agent principal.

De la même manière, l'accélération de la position relative peut être définie comme la seconde dérivée relative :

$$q''_i = \frac{dq'_i}{dq_m} \quad (2.27)$$

De l'équation (2.26), la vitesse de l'axe i peut être définie comme suit :

$$\dot{q}_i = \dot{q}_m \dot{q}'_i \quad (2.28)$$

Par conséquent, l'équation (2.27) peut être exprimée comme :

$$\ddot{q}_i = \ddot{q}'_i \dot{q}_m^2 + \dot{q}'_i \ddot{q}_m \quad (2.29)$$

Les équations ci-dessus montrent le rapport entre mouvements absolus et relatifs. L'équation (2.28) relie la vitesse absolue dans le domaine temporel avec la dérivée relative dans le domaine de position, alors que l'équation (2.29) relie l'accélération absolue à l'accélération relative. Les deux équations sont utilisées pour transformer le modèle dynamique du domaine temporel vers le domaine de position.

2.4.2 Modèle dynamique temporel d'un robot à n ddl

Le modèle d'un robot à n ddl peut être représenté comme suit :

$$M(q)\ddot{q}(t) + C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q}) = B(q) \times \tau(t) \quad (2.30)$$

Où : $M(q)$, $C(q, \dot{q})$ et $G(q)$ sont la matrice d'inertie, les forces centrifuges et de Coriolis et le vecteur de gravitation respectivement et $B(q)$ est la matrice de transformation de l'entrée . Enfin, $F(t, q, \dot{q})$ est la matrice comportant les forces de frottement et τ le vecteur de couples de commande.

On opte pour une représentation en structure maître-esclave de la dynamique du robot comme décrit précédemment :

$$\begin{bmatrix} m_{mm} & M_{ms} \\ M_{sm} & M_{ss} \end{bmatrix} \begin{bmatrix} \ddot{q}_m \\ \ddot{q}_s \end{bmatrix} + \begin{bmatrix} c_{mm} & C_{ms} \\ C_{sm} & C_{ss} \end{bmatrix} \begin{bmatrix} \dot{q}_m \\ \dot{q}_s \end{bmatrix} + \begin{bmatrix} G_m \\ G_s \end{bmatrix} + \begin{bmatrix} F_m \\ F_s \end{bmatrix} = \begin{bmatrix} \tau_m \\ \tau_s \end{bmatrix} \quad (2.31)$$

L'indice m se réfère à l'agent maître et l'indice s se réfère aux agents esclaves.

2.4.3 Modèle dynamique dans le domaine de position

Dans le domaine de position, la position du maître est utilisée comme référence indépendante pour le suivi d'un contour défini. Par conséquent, le modèle dynamique pour les agents esclaves peut être réécrit dans le domaine de position en fonction de la référence par une transformation du domaine temporel vers le domaine de position :

En remplaçant dans le système (2.31) par les équations (2.28) et (2.29), un modèle dynamique pour les mouvements des esclaves dans le domaine de position est dérivé de la manière suivante :[34]

$$\dot{q}_m^2 M_{ss} q_s''(q_m) + (\ddot{q}_m M_{ss} + \dot{q}_m C_{ss}) q_s'(q_m) + \ddot{q}_m M_{sm} + \dot{q}_m C_{sm} + G_s + F_s = \tau_s(q_m) \quad (2.32)$$

L'équation ci-dessus représente la relation dynamique entre le mouvement principal, indiqué par l'indice m , et les mouvements des esclaves, indiqués par l'indice s

les dimensions du modèle dynamique dans le domaine de position sont $\{M_{sm}, G_s, F_s, \tau_s\} \in \mathbb{R}^{n-1}$, $\{M_s, C_{ss}\} \in \mathbb{R}^{(n-1) \times (n-1)}$ et $\{q_s, q_s', q_s''\} \in \mathbb{R}^{n-1}$

La dynamique de l'esclave dans le domaine de position peut être redéfinie en utilisant les paramètres suivant :

$$\begin{cases} \overline{M} = \dot{q}_m^2 M_{ss} \\ \overline{C} = \ddot{q}_m M_{ss} + \dot{q}_m C_{ss} \\ \overline{D} = M_{sm} \ddot{q}_m + \dot{q}_m C_{sm} + G_s + F_s \end{cases} \quad (2.33)$$

En utilisant l'équation (2.33), l'équation(2.32) peut être exprimée sous la forme compacte suivante :

$$\overline{M} q_s''(q_m) + \overline{C} q_s'(q_m) + \overline{D} = \tau_s(q_m) \quad (2.34)$$

A l'évidence, la non-linéarité du modèle du domaine temporel dans l'équation (2.32) est maintenue dans le domaine de la position.

À partir de cette équation, on peut déduire que la position du maître a une plus grande influence sur le contrôle du système que son erreur de suivi du contour désiré.

Il est entendu que la structure de contrôle du domaine de position requiert que le contrôle du mouvement maître fonctionne dans le domaine temporel. Dans ce sens, la commande dans le domaine de position est la combinaison de deux contrôleurs de natures différentes s'exécutant en séquence comme le montre les figures (2.5), (2.6).

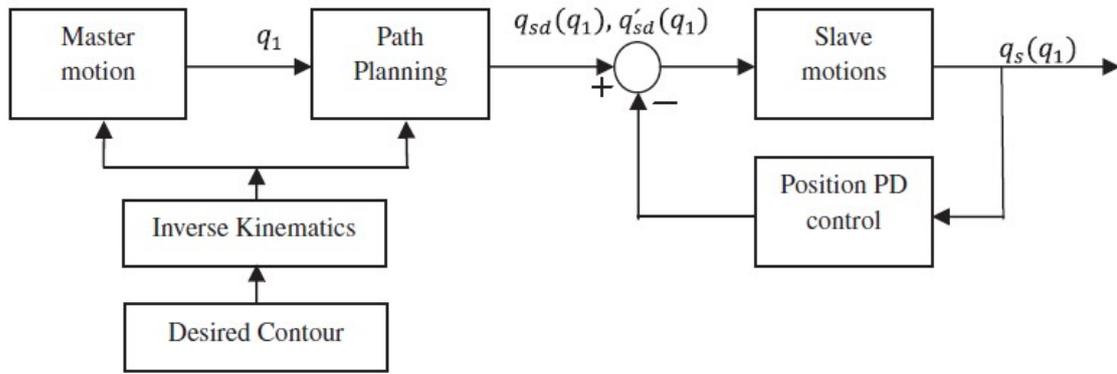


FIGURE 2.5 – Schéma de commande dans le domaine de position PD [6]

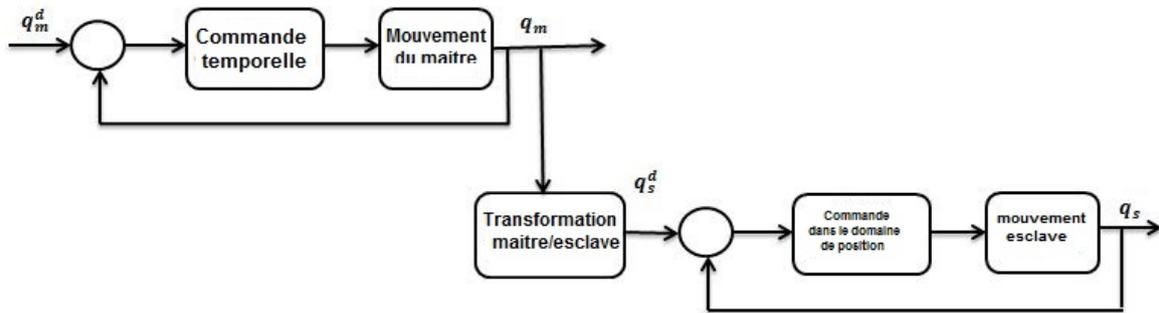


FIGURE 2.6 – Schéma de commande dans le domaine de position [7]

2.5 Remarques

On va récapituler les idées importantes de la commande dans le domaine de position en plus on va proposer quelques idées pour clarifier le concept dans les remarques suivantes :

R1 : Pour formuler le réglage d'un système dans le domaine de position, ce système supposé d'ordre n est transformé en un système dynamique d'ordre $n - 1$ dans le domaine de position tout en conservant la dynamique de son articulation maitre. En conséquence, l'erreur de contour diminue lors du passage vers le domaine de position pour deux raisons principales. La première étant l'annulation de la contribution de l'articulation maitre dans la création de l'erreur globale de suivi de contour. La deuxième raison est que la dynamique du maitre est conservée dans le domaine temporel sans intervenir dans l'expression de l'erreur de suivi de contour par l'effecteur.

R2 : Toutefois, les variations de l'articulation maitre doivent être strictement monotones par parties. Autrement, elle doit être similaire au temps "t" pour pouvoir intégrer l'équation différentielle ,et donc théoriquement on doit avoir $\dot{q}_m \neq 0$ pour chaque partie. La condition précédente se traduit numériquement par $\|\dot{q}_m\| > \varepsilon > 0$, tel que ε est la tolérance du calculateur.

R3 : Étant donné que le temps est toujours positif, alors que la variable maitre q_m peut être négative ce qui donne une solution pour l'équation différentielle non- significative voir instable , pour cela on va utilisé la théorie de Lyapounov afin de régler ce problème et ce dans le chapitre suivant portant sur la synthèse de commandes.

R4 : En particulier, dans le cas du robot planaire poly-articulaire il est primordial d'imposer une dynamique performante pour les articulations. En d'autre terme, comme on commande dans l'espace de travail pour suivre un contour désiré les articulation esclaves q_{d_i} doivent correspondre à la cinématique inverse . Simultanément, la variable maitre q_{m_d} est toujours monotone par partie du moment que celle-ci subit toute l'inertie du système. En effet, le couple τ_m appliqué au niveau de l'articulation maitre (la première articulation) vérifie $\tau_m \neq 0 \quad \forall t > 0$ et comme ce couple est l'image de l'accélération de cette articulation \ddot{q}_m , on a donc $\ddot{q}_m \neq 0 \quad \forall t > 0$. Ainsi, par intégration on a \dot{q}_m est strictement monotone ce qui implique que q_m doit être strictement monotone par partie On peut donc conclure que le robot manipulateur est choisi comme la structure la plus appropriée pour la commande dans le domaine de position pour les raisons évoquées précédemment.

Conclusion

Dans ce chapitre, nous avons exposé le concept de formulation d'un problème de suivi de contour dans le domaine de position. Nous avons introduit la loi de commande dans le domaine de position comme une alternative aux techniques de commande temporelles usuelles Cette approche fera l'objet de notre étude dans les prochains chapitres et sera comparée aux techniques de contrôle temporelles existantes.

CHAPITRE 3

SYNTHÈSE DES LOIS DE COMMANDES

Chapitre 3 : Synthèse des lois de commandes

Introduction

Dans ce chapitre, nous synthétisons quelques lois de commande non linéaire dédiées au suivi de contours par deux structures de robots différentes. En parallèle, nous formulons le problème de commande dans le domaine de position selon les transformations étudiées dans le chapitre précédent. Enfin, nous analysons la stabilité du système commandé pour chacune des méthodes considérées.

3.1 Système robotique dans le domaine de position

3.1.1 Robot manipulateur

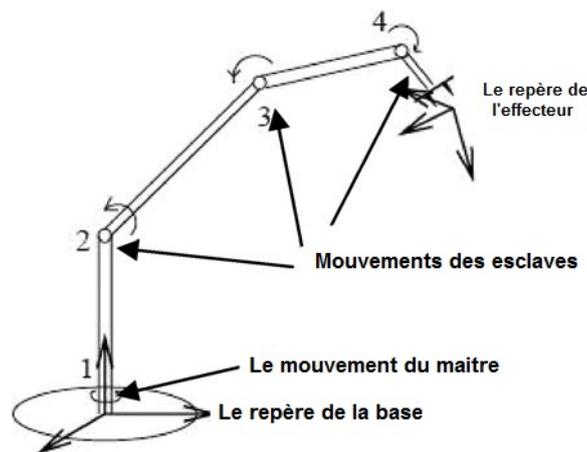


FIGURE 3.1 – Schéma d'un robot manipulateur dans le domaine de position[5]

Un robot manipulateur, illustré dans la figure(3.1), est utilisé à titre d'exemple pour décrire le concept de CDP. Dans la CDP, un principe de mouvement maître-esclave est utilisé . Le mouvement principal (mouvement articulation 1), mesuré par un capteur, est utilisé comme référence qui n'introduira aucune erreur à l'erreur de contour, et les mouvements des esclaves (les mouvements des articulations 2 à 4) sont décrits comme fonctions du mouvement du maître selon les exigences de la trajectoire du contour. Le mouvement maître fonctionne dans le domaine temporel, contrôlé par une loi de commande temporelle .

D'autre part, les mouvements des esclaves fonctionnent dans le domaine de position, en utilisant le mouvement principal comme référence au lieu du temps ([17]. On reprend le modèle dynamique dans le domaine temporel et dans le domaine de position (3.1) illustré dans la chapitre précédant (section 2.4.3)

Équation Dynamique du robot dans le domaine temporel

$$M(q)\ddot{q}(t) + C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q}) = \tau(t) \quad (3.1)$$

Équation Dynamique du robot dans le domaine de position

$$\overline{M}q_s''(q_m) + \overline{C}q_s'(q_m) + \overline{D} = \tau_s(q_m) \quad (3.2)$$

Avec

$$\begin{cases} \overline{M} = \dot{q}_m^2 M_{ss} \\ \overline{C} = \ddot{q}_m M_{ss} + \dot{q}_m C_{ss} \\ \overline{D} = M_{sm} \ddot{q}_m + \dot{q}_m C_{sm} + G_s + F_s \end{cases} \quad (3.3)$$

3.1.2 Propriétés et hypothèses du système dynamique

Certaines propriétés qu'on aura besoins pour les commandes et l'étude de stabilité des commandes du robot manipulateur décrit par les equations (3.1) et (3.2) sont données comme suit[36] [37] :

P1 : La matrice d'inertie $M(q) \in \mathfrak{R}^{n \times n}$ est symétrique et définie positive. $M_{ss}(q) \in \mathfrak{R}^{(n-1) \times (n-1)}$ est par conséquent symétrique et définie positive.

P2 : La matrice $\dot{M}(q) - 2C(q, \dot{q})$ est antisymétrique , de même la matrice $\dot{M}_{ss}(q) - 2C_{ss}(q, \dot{q})$,où $C(q, \dot{q})$ est la matrice des effets Coriolis et centrifuge.

P3 : les matrices d'inertie et des effets Coriolis et centrifuge satisfait l'équation suivante :

$$\begin{cases} \dot{M}(q) = C(q, \dot{q}) + C^T(q, \dot{q}) \\ \dot{M}_{ss}(q) = C_{ss}(q, \dot{q}) + C_{ss}^T(q, \dot{q}) \end{cases} \quad (3.4)$$

P4 : $M(q_d), C(q_d, \dot{q}_d), G(q_d)$, et $F(t, q_d, \dot{q}_d)$ sont bornées , ou q_d et \dot{q}_d sont la position désirée et la vitesse désirée respectivement , et ils sont bornées.

De plus, les notations suivantes seront utilisées $\lambda_m(M)$ et $\lambda_M(M)$ représentent la valeur propre minimale et maximale de la matrice M . et finalement deux suppositions sont utilisés lors de l'étude de stabilité .

S1 :La vitesse \dot{q}_m et l'accélération \ddot{q}_m du mouvement du maître sont bornées dans la région du contour désiré.

S2 :le contour désiré du mouvement des esclaves $q_{sd}(q_m)$ est borné et $q_{sd}(q_m) \in C^2(q_m)$.

3.1.3 Robot mobile

Le contrôle des robots mobiles à roues (Wheeled Mobile Robots WMRs) a considérablement attiré l'attention de la communauté roboticienne au cours des dernières années. En effet, la forte utilisation des robots mobiles dans les domaines où l'être humain ne peut pas être présent, notamment dans les sites nucléaires à haut risque ou dans le cas de l'exploration spatiale, nécessite la mise en œuvre de lois de commande autonomes et performantes pour assurer les tâches assignées aux robots. Une propriété des modèles de robot mobile non-holonome (de type voiture ou uni-cycle) est qu'ils ne sont pas stabilisables par un retour d'état continu et stationnaire. Ce corollaire est le résultat des travaux de Brockett [38].

La plupart des recherches effectuées ont porté sur le problème de génération et de poursuite de trajectoires au niveau cinématique, c'est-à-dire en tenant compte des vitesses du système en tant qu'entrées de commande et en ignorant les caractéristiques dynamiques du système mécanique [39] [40] [41]. Deux des problèmes de commande les plus importants concernant les robots mobiles sont le suivi d'une trajectoire de référence et la stabilisation à une position fixe. Le problème de suivi a reçu des solutions y compris les techniques classiques de contrôle non linéaire [39] [40] [42]. L'idée de base est d'avoir un véhicule de référence qui génère une trajectoire à suivre par le robot mobile. Dans [39] [40], des entrées de commandes non linéaires en vitesse ont été définies afin d'annuler l'erreur de suivi tant que le robot de référence se déplaçait. En revanche, le problème de stabilisation autour d'une position fixe s'est montré plutôt compliqué. Ceci est dû à la violation de la condition de [38].

Modèle dynamique du robot mobile

On considère un robot mobile à trois roues qui se déplace sur un plan horizontal (figure (3.2)). Ce robot dispose de deux roues arrière à entraînement différentiel et d'une roue avant à roulette (roue folle). Le rayon des roues est dénoté r et la longueur de l'axe de la roue arrière est de $2l$. Les entrées du système sont deux couples τ_1 et τ_2 , fournis par deux moteurs attachés aux roues arrière.

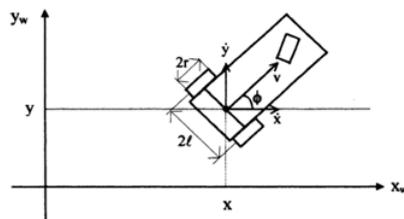


FIGURE 3.2 – Robot Mobile uni-cycle [43]

Le robot mobile est régi avec une équation dynamique du second ordre similaire à celle du robot manipulateur, avec comme seules différences, une matrice d'entrée B non unitaire et un terme supplémentaire du aux contraintes non holonomes.

Équation Dynamique du robot mobile dans le domaine temporel

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B(q)\tau - A^T(q)\lambda \quad (3.5)$$

Avec

$$M(q) = \begin{bmatrix} m & 0 & md \sin \phi \\ 0 & m & -md \cos \phi \\ md \sin \phi & -md \cos \phi & I \end{bmatrix} \quad (3.6)$$

$$C(q, \dot{q}) = \begin{bmatrix} 0 & 0 & md\dot{\phi} \cos \phi \\ 0 & 0 & md\dot{\phi} \sin \phi \\ 0 & 0 & 0 \end{bmatrix} \quad (3.7)$$

$$G(q) = 0 \quad (3.8)$$

$$B(q) = \frac{1}{r} \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & -l \end{bmatrix} \quad (3.9)$$

$$A(q) = \begin{bmatrix} -\sin \phi & \cos \phi & 0 \end{bmatrix} \quad (3.10)$$

Enfin : λ est le multiplicateur de Lagrange, donné par :

$$\lambda = -m\dot{\phi}(\dot{x} \cos \phi + \dot{y} \sin \phi) \quad (3.11)$$

Tels que :

d : la distance entre M , le centre de masse du robot et C , le centre géométrique du robot.

r : le rayon de chaque roue motrice.

l : la distance entre le centre géométrique et l'axe passant par les centres des deux roues du véhicule.

I : le moment d'inertie du robot.

m : la masse du robot. Dans notre cas, on prend : $d=0$. Les matrices du système deviennent :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B(q)\tau - A^T(q)\lambda \quad (3.12)$$

Avec

$$M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \quad (3.13)$$

$$C(q, \dot{q}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.14)$$

$$G(q) = 0 \quad (3.15)$$

$$B(q) = \frac{1}{r} \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & -l \end{bmatrix} \quad (3.16)$$

La simplification du modèle donne :

$$\begin{cases} \ddot{x} = \frac{\lambda}{m} \sin \phi + b_1 u_1 \cos \phi \\ \ddot{y} = -\frac{\lambda}{m} \cos \phi + b_1 u_1 \sin \phi \\ \ddot{\phi} = b_2 u_2 \end{cases} \quad (3.17)$$

$$\dot{x} \sin \phi - \dot{y} \cos \phi = 0 \quad (3.18)$$

Où $b_1 = \frac{1}{rm}$ et $b_2 = \frac{l}{rI}$. En outre, $u_1 = \tau_1 + \tau_2$ et $u_2 = \tau_1 - \tau_2$ sont les entrées de commande. L'équation (3.18) est la contrainte de non holonomie, provenant de l'hypothèse que les roues ne glissent pas. Le vecteur $q(t) = [x(t), y(t), \phi(t)]^T$ désigne la trajectoire (position et orientation) du robot par rapport à un espace de travail fixe. Ainsi, $q = [x, y, \phi]^T$ décrit la configuration du robot (posture) à tout instant t . Nous supposons également qu'à tout moment, la position du robot $q = [x, y, \phi]^T$ ainsi que sa dérivée $\dot{q} = [\dot{x}, \dot{y}, \dot{\phi}]^T$, sont mesurés en sortie du système commandé.

Remarque

Dans le modèle dynamique du robot mobile représenté ci-dessus, l'influence de la roue arrière qui est passive et auto-ajustable n'est pas prise en considération vu qu'il s'agit d'une roue libre. Ceci réduit considérablement la complexité de la synthèse du contrôleur. Néanmoins, cette roue libre peut constituer une source de distorsion particulièrement dans le cas de changement de la direction du robot(3.2). Finalement, on constate que le système étudié dans cette section est composé de deux sous-systèmes découplés comme suit :

Sous-système de direction(orientation) du robot

$$\ddot{\phi} = b_2 u_2 \quad (3.19)$$

Sous-système de déplacement linéaire du robot

$$\begin{cases} \ddot{x} = \frac{\lambda}{m} \sin \phi + b_1 u_1 \cos \phi \\ \ddot{y} = -\frac{\lambda}{m} \cos \phi + b_1 u_1 \sin \phi \end{cases} \quad (3.20)$$

3.1.4 Modèle dynamique dans le domaine de position

Tout d'abord, on représente le système du robot mobile dans une sous une maitre-esclave, avec :

$$\begin{cases} q_m(t) = \phi(t) \\ q_{s_1}(q_m) = x(q_m) \\ q_{s_2}(q_m) = y(q_m) \end{cases} \quad (3.21)$$

Ainsi, le modèle temporel précédent peut être formulé dans le domaine de position, avec les matrices suivantes :

$$\overline{M}q_s''(q_m) + \overline{C}q_s'(q_m) = \overline{B}\tau_s(q_m) - \overline{A}^T \lambda(q_m, \dot{q}_m) \quad (3.22)$$

Avec

$$\begin{cases} \overline{M} = \dot{q}_m^2 M_{ss} \\ \overline{C} = \ddot{q}_m M_{ss} \\ \overline{B} = B(q_m) \\ \overline{A} = A(q_m) \\ \lambda(q_m, \dot{q}_m) = -m\dot{q}_m^2 (q'_{s_1} \cos q_m + q'_{s_2} \sin q_m) \end{cases} \quad (3.23)$$

Tel que :

$$M_{ss} = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \quad (3.24)$$

Enfin, le modèle simplifié du robot mobile dans le domaine de position s'écrit :

$$\begin{cases} \ddot{\phi}(t) = b_2 u_2(t) \\ x''(q_m) = \frac{1}{\dot{q}_m^2} \left(\frac{\lambda(q_m)}{m} \sin q_m + b_1 u_1(q_m) \cos q_m - \ddot{q}_m x' \right) \\ y''(q_m) = -\frac{1}{\dot{q}_m^2} \left(\frac{\lambda(q_m)}{m} \cos q_m + b_1 u_1(q_m) \sin q_m - \ddot{q}_m y' \right) \end{cases} \quad (3.25)$$

Formulation du problème de poursuite de trajectoire

Considérons un robot mobile représenté par un modèle non linéaire décrit par l'équation (3.17). Étant donnée une trajectoire de référence admissible $q_r(t)$, l'objectif de la poursuite

de trajectoire est de déterminer la commande $u = [u_1, u_2]^T$ du système (3.2) en fonction de q, \dot{q}, q_R et du temps t telle que l'erreur de poursuite $q_e = q_R - q$ converge asymptotiquement vers zéro. Dans le cadre d'un robot mobile non holonome de type uni-cycle, le principe de la poursuite de trajectoire est illustré sur la figure (3.3)

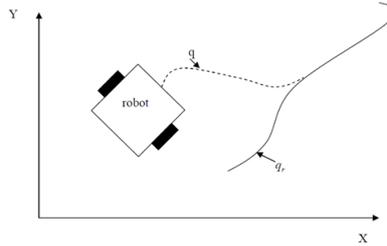


FIGURE 3.3 – Poursuite de trajectoires admissibles[43]

Dans les sections restantes du chapitre, on synthétisera les commandes considérées dans l'espace temporel et de position uniquement pour le sous système de déplacement linéaire. Le premier sous-système quant à lui sera commandé par un PD simple. Ainsi, pour le sous-système d'orientation, la loi de commande est :

$$u_2(t) = u_m(t) = K_{P_\phi} e_\phi + K_{D_\phi} \dot{e}_\phi \quad (3.26)$$

$$e_\phi = \phi_d(t) - \phi(t) \quad \dot{e}_\phi = \dot{\phi}_d(t) - \dot{\phi}(t) \quad (3.27)$$

3.2 La commande PD-MG

3.2.1 Synthèse de la de commande pour le robot manipulateur

Mr. Ouayang dans[5] a proposé une commande par PD combinée à une commande à structure variable dans le domaine de position. L'objectif de l'ajout de la commande discontinue est de rendre robuste la commande classique par PD .

3.2.2 Commande dans le domaine temporel PD-MG

$$\begin{cases} \tau(t) = K_P e(t) + K_D \dot{e}(t) + H \text{sign}(S) = \begin{bmatrix} \tau_1(t) \\ \tau_2(t) \\ \dots \\ \tau_n(t) \end{bmatrix} \\ S = \dot{e}(t) + \lambda e(t) \end{cases} \quad (3.28)$$

3.2.3 Commande dans le domaine de position PDP-MG

La nouvelle loi de commande PDP-MG est similaire à la commande PD + MG dans le temps [37] et elle est définie comme [44].

$$\begin{cases} \tau_m(t) = \tau_1(t) \\ \tau_s(q_m) = K_{P_s} e_s(q_m) + K_{D_s} e'_s(q_m) + H_s \text{sign}(s) \\ s = e'_s(q_m) + \lambda_s e_s(q_m) \end{cases} \quad (3.29)$$

Où $e_s(q_m)$ et $e'_s(q_m)$ représentent l'erreur de poursuite et la dérivée relative première de cette dernière ,avec $e_s(q_m) = q - sd(q_m) - q_s(q_m)$, s est la surface de glissement définie dans le domaine de position et $\text{sign}(s)$ est la fonction signe , K_{ps} et $K_{ds} \in \mathfrak{R}^{(n-1) \times (n-1)}$ représentent les matrices gain proportionnel et dérivé. $H_s \in \mathfrak{R}^{(n-1) \times (n-1)}$ est la matrice gain de la commande par MG, et $\lambda_s > 0$ est une matrice définie positive qui représente les constantes de glissement.

Respectivement le schéma de commande proposé par [5] est donné comme suit :

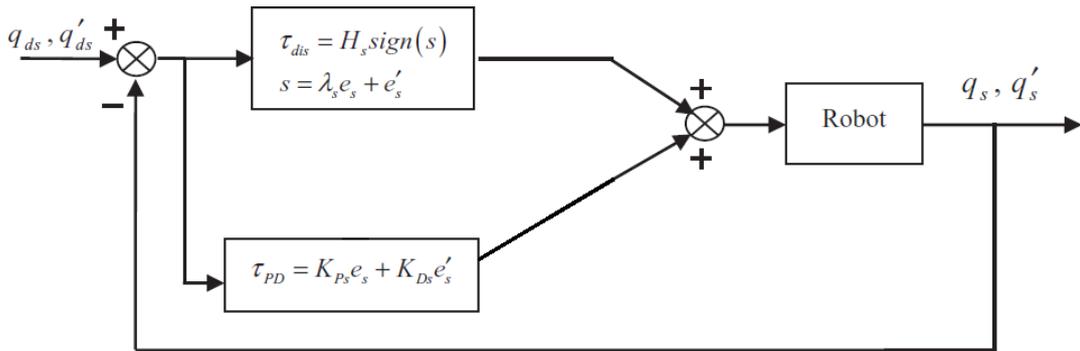


FIGURE 3.4 – Schéma blocs pour la commande par MG standard et le PDC-MG.

Remarque 1 : La figure (3.4) montre un schéma blocs de commande par mode de glissement et le PDC-MG proposé. Pour une commande MG standard, le couple (la commande)

combine deux parties : l'une est la commande équivalente τ_{eq} qui est basée sur le modèle dynamique normalisé d'un système robotique et l'autre est la commande de commutation discontinue qui est utilisée pour assurer l'attractivité de la trajectoire vers la surface de glissement $s = 0$.

Dans le PDC-MG proposé, comme le montre la figure (3.4), le couple équivalent τ_{eq} est remplacé par une commande τ_{PD} où aucune information de modèle dynamique n'est requise. De l'équation (3.2) et de la figure (3.4), on peut voir que la commande PDC-MG est une combinaison de d'une commande proportionnel dérivée et la commande par MG. la commande PD est utilisé pour stabiliser le modèle nominal du système et le commande par MG est utilisés pour assurer la robustesse a fin de compenser les incertitudes et les perturbations sur le système robotique à commander.

Remarque 2 :Il est entendu que la structure PDC exige que le mouvement principal fonctionne dans le domaine temporel et de position. Dans ce sens, PDC-MG est un système hybride avec deux différents domaines et des commandes séquentiels . Comme le mouvement maître remplace le temps pour les mouvements des esclaves , la trajectoire du maitre doit être impérativement une fonction strictement monotone. Par conséquent, les profils de contour complexes doivent être segmentés de telle sorte que, dans chaque segment, la trajectoire du mouvement maître soit alors strictement monotone croissante / décroissante (vitesse positive /négative).

Théorème [5] Étant donné un robot manipulateur rigide décrit dans le domaine de position par l'équation (3.2), si la loi PDC-MG dans l'équation (3.28) est appliquée pour commander le robot manipulateur à fin de suivre une contour désiré et les conditions de l'équation (3.29) doivent être satisfaites, le commande du robot manipulateur est globalement asymptotiquement stable pour le suivi du contour.

$$\begin{cases} K > 0, K_{Ds} > 0, H_s > \|\tau_d\| \\ K_{Ps} > \lambda_s^2 \bar{M}, K + \bar{C}^T > 0 \\ K_{Ps} - (\frac{1}{2}\lambda_- M(K + \bar{C}^T))I > 0 \\ K_{Ds} - \lambda_s \bar{M} - \frac{\lambda_s}{2} \lambda_M (K + \bar{C}^T)I > 0 \end{cases} \quad (3.30)$$

Proposition Supposons qu'une matrice Q symétrique est exprimée comme suit :

$$Q = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \quad (3.31)$$

Soit S le complément Schur d'une matrice A dans l'équation (3.30), ou :

$$S = C - B^T A^{-1} B \quad (3.32)$$

Alors, la matrice Q est positive définie si et seulement si A et S sont à la fois définis, c'est-à-dire si $A > 0$ et $S > 0$, Alors $Q > 0$ [46]. Pour prouver la stabilité de la loi PDC-MG proposée, la matrice L suivante est définie :

$$L = \begin{bmatrix} K_{P_s} & \lambda_s \overline{M} \\ \lambda_s \overline{M} & \overline{M} \end{bmatrix} \quad (3.33)$$

De (**P1**) et de l'équation (3.3), M_{ss} est symétrique et définie positive. Par conséquent, \overline{M} est symétrique et définie positive. L'application de l'équation (3.31) à la matrice L donne :

$$S = \overline{M} - \lambda_s^2 \overline{M} K_{P_s}^{-1} \overline{M} \quad (3.34)$$

De l'équation (3.29), $K_{P_s} > \lambda_s^2 \overline{M}$. D'où $S > 0$. Selon la proposition, on conclut que la matrice L est symétrique et positive définie, c'est-à-dire $L > 0$.

3.2.4 L'analyse de stabilité

En appliquant l'équation (3.28), le modèle dynamique de l'équation (3.2) peut être exprimé en fonction des erreurs.

$$\begin{aligned} \overline{M} e_s'' &= \tau_d - \overline{C} e_s' - K_{P_s} e_s - K_{D_s} e_s' - H \text{sign}(s) \\ \tau_d &= \overline{M} q_{sd}'' + \overline{C} q_{sd}' + \overline{D} \end{aligned} \quad (3.35)$$

Soit la fonction Lyapunov suivante :

$$V(e_s(q_m), e_s'(q_m)) = \frac{1}{2} \begin{pmatrix} e_s^T & e_s'^T \end{pmatrix} L \begin{pmatrix} e_s \\ e_s' \end{pmatrix} + \frac{\lambda_s}{2} e_s^T (K + K_{D_s}) e_s \quad (3.36)$$

Comme $L > 0$ et $K + K_{D_s} > 0$, cela signifie que $V(e_s(q_m), e_s'(q_m)) > 0$. Dans le domaine de position, la position de référence q_m du mouvement maître est une variable indépendante ayant une signification similaire de t dans le domaine temporel, avec e_s et e_s' étant des fonctions de la variable indépendante q_m . De l'équation (3.35), la dérivée relative de la fonction Lyapunov par rapport au mouvement principal du système est donnée par :

$$\begin{aligned} V' &= \begin{pmatrix} e_s^T & e_s'^T \end{pmatrix} L \begin{pmatrix} e_s' \\ e_s'' \end{pmatrix} + \frac{1}{2} \begin{pmatrix} e_s^T & e_s'^T \end{pmatrix} \begin{bmatrix} 0 & \lambda_s \overline{M}' \\ \lambda_s \overline{M}' & \overline{M}' \end{bmatrix} \begin{pmatrix} e_s \\ e_s' \end{pmatrix} + \lambda_s e_s^T (K + K_{D_s}) e_s' \\ &= \begin{pmatrix} e_s^T & e_s'^T \end{pmatrix} \begin{pmatrix} K_{P_s} e_s' + \frac{\lambda_s \overline{M}'}{2} e_s' + \lambda_s \overline{M} e_s'' \\ \frac{\lambda_s \overline{M}'}{2} e_s + \lambda_s \overline{M} e_s' + \frac{\overline{M}'}{2} e_s' + \overline{M} e_s'' \end{pmatrix} + \lambda_s e_s^T (K + K_{D_s}) e_s' \end{aligned} \quad (3.37)$$

Selon les propriétés **P2** et **P3**, il est prouvé que :

$$\begin{cases} e_s'^T (\frac{\overline{M}'}{2} - \overline{C}) e_s' = 0 \\ \overline{M}' = \overline{C} + \overline{C}^T \end{cases} \quad (3.38)$$

En appliquant les équations (3.34) et (3.36) dans l'équation (3.37), l'expression suivante peut être obtenue :

$$V' = -\lambda_s e_s^T K_{Ps} e_s - e_s'^T (K_{Ds} - \lambda_s \bar{M}) e_s' + \lambda_s e_s' + \lambda_s e_s^T (K + \bar{C}^T) e_s' + s^T (\tau_d - H_s \text{sign}(s)) \quad (3.39)$$

Pour faciliter l'analyse de stabilité, une matrice définie positive K constante est introduite pour s'assurer que la matrice $K + \bar{C}^T$ est positive définie. Selon l'équation (3.29), l'inégalité suivante est vérifiée : $0 < K + \bar{C}^T < \lambda_M(K + \bar{C}^T)I$. Donc, l'expression suivante peut être obtenue :

$$e_s^T (K + \bar{C}^T) e_s' \leq e_s^T \frac{\lambda_M(K + \bar{C}^T)}{2} I e_s + e_s'^T \frac{\lambda_M(K + \bar{C}^T)}{2} I e_s' \quad (3.40)$$

En appliquant l'équation (3.38) dans l'équation (3.39) et après simplification, l'inégalité suivante est vérifiée :

$$V' \leq -\lambda_s e_s^T (K_{Ps} - \frac{\lambda_M(K + \bar{C}^T)}{2} I) e_s - e_s'^T (K_{Ds} - \lambda_s \bar{M} - \frac{\lambda_M}{2} (K + \bar{C}^T) I) e_s' - (\lambda_s s^T) (H_s \text{sgn}(s) - \tau_d) \quad (3.41)$$

Selon l'équation (3.29), on peut prouver que :

$$\begin{cases} \lambda_s e_s^T (K_{Ps} - \frac{\lambda_M(K + \bar{C}^T)}{2} I) e_s \geq 0 \\ e_s'^T (K_{Ds} - \lambda_s \bar{M} - \frac{\lambda_s}{2} (K + \bar{C}^T) I) e_s' \geq 0 \\ s^T H_s \text{sgn}(s) = |s| H_s > |s| \|\tau_d\| \geq s^T \tau_d \end{cases} \quad (3.42)$$

D'après l'équation (3.41), on peut conclure que : $V' \leq 0$. L'égalité $V = 0$ est satisfait si et seulement si $e_s = e_s' = 0$. Puisque la fonction V est définie positive et V' est définie négative, on peut conclure que les mouvements des esclaves robot manipulateur dans l'équation (3.2) sont globalement asymptotiquement stables et les erreurs de position et de vitesse relative de la vitesse vont à zéro. Pour éviter le phénomène de broutement en raison de l'action de commutation de la fonction de signe [3], dans la loi PDC-MG proposée dans l'équation (3.28), la fonction de saturation peut être utilisée pour lisser la discontinuité de la commande avec une fine couche limite.

Le PDC-MG proposé de l'équation (3.28) peut être modifié comme suit :

$$\tau_s(q_m) = K_{Ps} e_s(q_m) + K_{Ds} e_s'(q_m) + H_s \text{sat}\{s, \Phi_s\} \quad (3.43)$$

Où Φ_s est une matrice diagonale constante qui détermine l'épaisseur de couche limite mince de la surface de glissement et $\text{sat}(\Delta)$ est une fonction de saturation définie comme :

$$\text{sat}\{s, \Phi_s\} = \begin{cases} \text{sign}(s) & \text{if } |s| > \Phi_s \\ s \setminus \Phi_s & \text{if } |s| \leq \Phi_s \end{cases} \quad (3.44)$$

3.2.5 Synthèse de la de commande pour le robot mobile

Commande dans le domaine temporel PD-MG

$$\begin{cases} u_1(t) = K_{P_x}e_x(t) + K_{P_y}e_y(t) + K_{D_x}\dot{e}_x(t) + K_{D_y}\dot{e}_y(t) + K_s \text{sign}(s) \\ s = \dot{e}_x(t) + \dot{e}_y(t) + \lambda_x e_x(t) + \lambda_y e_y(t) \end{cases} \quad (3.45)$$

Commande dans le domaine de position PDP-MG

La nouvelle loi de commande PDP-MG est similaire à la commande PD + MG dans le temps et elle est définie par :

$$\begin{cases} u_m(t) = u_2(t) \\ u_s(q_m) = K_{P_x}e_x(q_m) + K_{P_y}e_y(q_m) + K_{D_x}\dot{e}'_x(q_m) + K_{D_y}\dot{e}'_y(q_m) + K_s \text{sign}(s_s) \\ s_s = \dot{e}'_x(q_m) + \dot{e}'_y(q_m) + \lambda_x e_x(q_m) + \lambda_y e_y(q_m) \end{cases} \quad (3.46)$$

3.3 La commande par Forwarding

3.3.1 Synthèse des des lois de commande pour un robot manipulateur

Synthèse dans le domaine temporel

On reprend l'équation dynamique du robot manipulateur à n ddl .

$$M(q)\ddot{q}(t) + C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q}) = \tau(t) \quad (3.47)$$

Comme $M(q) > 0$ donc l'équation(3.47) devient :

$$\ddot{q}(t) = -M^{-1}(q)(C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q}) - \tau(t)) \quad (3.48)$$

on prends les états du système $\begin{pmatrix} q & \dot{q} \end{pmatrix}$ et $\tau(t) = \tilde{\tau}(t) + v$ le signal commande et posons $e_1 = q_d - q \in \mathfrak{R}^{n \times 1}$ et $e_2 = \dot{q}_d - \dot{q} \in \mathfrak{R}^{n \times 1}$ l'erreur sur la position et la vitesse des articulation ,respectivement q_d/\dot{q}_d représente la position/vitesse désirées pour un contour défini issu du modèle cinématique inverse [annexe 1].

La représentation d'état du système est la suivante :

$$\begin{bmatrix} \dot{e}_1(t) \\ \dot{e}_2(t) \end{bmatrix} = \begin{bmatrix} e_2 \\ \ddot{q}_d + M^{-1}(q)(C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q})) - M^{-1}(q)\tau(t) \end{bmatrix} \quad (3.49)$$

La synthèse du commande s'effectuera en deux étapes :

La première étape On ne considère que la deuxième équation du système (3.48) commandé seulement par $\tilde{\tau}(t)$.

Soit la fonction de Lyapounov : $V_2 = \frac{1}{2}e_2^T e_2$ Fonction définie positive sur \Re

Sa dérivée est : $\dot{V}_2 = \dot{e}_2^T e_2$.

Pour rendre \dot{V}_2 définie négative i.e $\dot{V}_2 = -e_2^T \lambda_2 e_2$ $\lambda_2 \geq 0$, on peut choisir la commande $\tilde{\tau}(t)$:

$$\tilde{\tau}(t) = M(q)(\ddot{q}_d + \lambda_2 e_2) + C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q}) \quad (3.50)$$

Cette commande va assurer la convergence de e_2 vers l'origine.

La deuxième étape En remplaçant la commande trouvée dans l'étape précédente dans le système (3.48), on obtient :

$$\begin{cases} \dot{e}_1 = e_2 \\ \dot{e}_2 = -\lambda_2 e_2 \end{cases} \quad (3.51)$$

La solution de ce système dans l'espace s est :

$$\begin{cases} \tilde{e}_2(s) = e^{-\lambda_2 s} e_2(t) \\ \tilde{e}_1(s) = \lambda_2^{-1} e^{-\lambda_2 s} e_2(t) + e_1(t) + \lambda_2^{-1} e_2(t) \end{cases} \quad (3.52)$$

Soit la fonction de Lyapounov :

$$\begin{aligned} V_1 &= V_2 + \frac{1}{2} \lim_{s \rightarrow \infty} \tilde{e}_1^2(s) \quad \text{avec} \quad V_1 > 0 \\ &= \frac{1}{2} [e_2^2 + (e_1 + \lambda_2^{-1} e_2)^2] \end{aligned} \quad (3.53)$$

On a $\tau(t) = \tilde{\tau}(t) + v$, la dérivée de V_1 sera :

$$\begin{aligned} \dot{V}_1 &= e_2^T \dot{e}_2 + (e_1 + \lambda_2^{-1} e_2)^T (\dot{e}_1 + \lambda_2^{-1} \dot{e}_2) \\ &= -e_2^T \lambda_2 e_2 + [\lambda_2^{-1} e_1 + ((\lambda_2^2)^{-1} + I)e_2]^T M^{-1} v \end{aligned} \quad (3.54)$$

Pour rendre \dot{V}_1 définie négative, on peut prendre :

$$v = M(q)[\lambda_2^{-1} e_1 + ((\lambda_2^2)^{-1} + I)e_2] \quad \text{avec} \quad \lambda_1 > 0 \quad (3.55)$$

la commande $\tau(t)$ devient :

$$\tau(t) = M(q)[\ddot{q}_d - \lambda_1 \lambda_2^{-1} e_1 - (\lambda_1 (\lambda_2^2)^{-1} + \lambda_1 + \lambda_2) e_2] + C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q}) \quad (3.56)$$

$$\text{Où } \lambda_1 = \begin{bmatrix} \lambda_{m1} & 0 \\ 0 & \lambda_{s1} \end{bmatrix} \text{ et } \lambda_2 = \begin{bmatrix} \lambda_{m2} & 0 \\ 0 & \lambda_{s2} \end{bmatrix}$$

Synthèse dans le domaine de position

En utilisant le même principe de la commande PD-MG dans le domaine de position , on va décomposer notre système en deux sous-système , un système maitre représentant la dynamique de la première articulation qu'on va utiliser sa commande temporelle calculée précédemment :

$$\tau_{temps}(t) = \begin{bmatrix} \tau_1(t) \\ \tau_2(t) \\ \vdots \\ \tau_n(t) \end{bmatrix} \quad (3.57)$$

Donc on prend :

$$\tau_m(t) = \tau_1(t) = [M(q)[\ddot{q}_d + \lambda_1 \lambda_2^{-1} e_1 + (\lambda_1 (\lambda_2^2)^{-1} + \lambda_1 + \lambda_2) e_2] + C(q, \dot{q}) \dot{q}(t) + G(q) + F(t, q, \dot{q})]^T \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (3.58)$$

Calcul de la commande des esclaves la commande des esclaves est calculé en utilisant le système dynamique dans le domaine de position :

$$\overline{M} q_s''(q_m(t)) + \overline{C} q_s'(q_m(t)) + \overline{D} = \tau_s(q_m(t)) \quad (3.59)$$

Où $q_m(t) \in \mathfrak{R}$, $q''_s, q'_s, q_s \in \mathfrak{R}^{n-1}$.On prends les états du système $\begin{pmatrix} q_s & q'_s \end{pmatrix}$ et $\tau_s(qm(t))$ le signal commande et posons $e_{s1} = q_{ds} - q_s \in \mathfrak{R}^{(n-1) \times 1}$ et $e_{s2} = \dot{q}'_{ds} - \dot{q}'_s \in \mathfrak{R}^{(n-1) \times 1}$ l'erreur sur la position et la vitesse des articulations , respectivement q_{ds}/\dot{q}'_{ds} représente la position/vitesse désirées des esclaves pour un contour définis issue en utilisant la transformation (temps/position) expliqué dans le chapitre 2 (2.4.1) .

La représentation d'état du système dans le domaine de position :

$$\begin{bmatrix} \dot{e}'_{1s}(qm(t)) \\ \dot{e}'_{2s}(qm(t)) \end{bmatrix} = \begin{bmatrix} e_{2s} \\ \dot{q}''_{ds} + \overline{M}^{-1}(q_s)(\overline{C}(q_s, \dot{q}'_s) \dot{q}'_s + \overline{D} - \overline{M}^{-1}(q_s) \tau_s(qm(t))) \end{bmatrix} \quad (3.60)$$

D'une manière équivalente à la synthèse de la commande Forwarding dans le domaine temporel.

La première étape On ne considère que la deuxième équation du système (3.48) commandé seulement par $\tilde{\tau}(t)$.

Soit la fonction de Lyapounov : $V_{s2} = \frac{1}{2} e_{s2}^T e_{s2}$ Fonction définie positive sur \mathfrak{R}

La dérivée relative est : $V'_{s2} = e_{s2}^T e'_{s2}$.

Pour rendre V'_{s2} définie négative et assurer que \dot{V}_{2s} on prend $V'_{s2} = -e_{2s}^T \tilde{\alpha} \lambda_{2s} e_{2s}$ avec $\lambda_{s2} \geq 0$, on peut choisir la commande $\tilde{\tau}_s(t)$:

$$\tilde{\tau}_s(t) = \overline{M}(q''_{sd} + \tilde{\alpha} \lambda_{s2} e_{s2}) + \overline{C}(q_s, q'_s) q'_s + \overline{D} \quad (3.61)$$

Où $\tilde{\alpha}$ est un opérateur définie comme suit :

$$\tilde{\alpha} = \tilde{\alpha}^{-1} = \begin{cases} 1 & \text{si } \dot{q}_m = 0 \\ \text{sign}(\dot{q}_m) & \text{si } \dot{q}_m \neq 0 \end{cases} \quad (3.62)$$

Cette commande va assurer la convergence de e_{s2} vers l'origine selon le théorème de stabilité de Lyapounov quand $t \rightarrow \infty \iff (q_{mi} \rightarrow q_{mf})$ car on a $\dot{V}_{2s} = \dot{q}_m V'_{2s} = -e_{2s}^T |\dot{q}_m| \lambda_{2s} e_{2s} < 0$

La deuxième étape En remplaçant la commande trouvée dans l'étape précédente dans le système (3.48), on obtient :

$$\begin{cases} \dot{e}_{s1} = e_{s2} \\ \dot{e}_{s2} = -\tilde{\alpha} \lambda_{s2} e_{s2} \end{cases} \quad (3.63)$$

La solution de ce système dans l'espace s est :

$$\begin{cases} \tilde{e}_{s2}(s) = e^{-\tilde{\alpha} \lambda_{s2} s} e_{s2}(q_m(t)) \\ \tilde{e}_{s1}(s) = \tilde{\alpha} \lambda_{s2}^{-1} e^{-\tilde{\alpha} \lambda_{s2} s} e_{s2}(q_m(t)) + e_{s1}(q_m(t)) + \tilde{\alpha} \lambda_{s2}^{-1} e_{s2}(q_m(t)) \end{cases} \quad (3.64)$$

Soit la fonction de Lyapounov :

$$\begin{aligned} V_{s1} &= V_{s2} + \frac{1}{2} \lim_{s \rightarrow \infty} \tilde{e}_{s1}^2(s) \quad \text{avec} \quad V_{s1} > 0 \\ &= \frac{1}{2} [e_{s2}^2 + (e_{s1} + \tilde{\alpha} \lambda_{s2}^{-1} e_{s2})^2] \end{aligned} \quad (3.65)$$

On a $\tau_s(t) = \tilde{\tau}_s(t) + v_s$, la dérivée de V_1 sera :

$$\begin{aligned} V'_{s1} &= e_{s2}^T e'_{s2} + (e_{s1} + \tilde{\alpha} \lambda_{s2}^{-1} e_{s2})^T (e'_{s1} + \tilde{\alpha} \lambda_{s2}^{-1} e'_{s2}) \\ &= -e_{s2}^T \tilde{\alpha} \lambda_{s2} e_{s2} + [\tilde{\alpha} \lambda_{s2}^{-1} e_{s1} + (((\tilde{\alpha} \lambda_{s2})^2)^{-1} + I) e_{s2}]^T \overline{M}^{-1} v_s \end{aligned} \quad (3.66)$$

Pour rendre V'_{s1} définie négative, on peut prendre :

$$v_s = \overline{M}(q_m(t)) \tilde{\alpha} \lambda_{s1} [\tilde{\alpha} \lambda_{s2}^{-1} e_{s1} + (((\tilde{\alpha} \lambda_{s2})^2)^{-1} + I) e_{s2}] \quad \text{avec} \quad \lambda_{s1} > 0 \quad (3.67)$$

la commande $\tau_s(t)$ devient :

$$\tau_s(t) = \overline{M}(q_m(t)) [q''_{sd} + \tilde{\lambda}_{s1} \lambda_{s2}^{-1} e_{s1} + (\tilde{\alpha} \lambda_{s1} (\lambda_{s2}^2)^{-1} + \tilde{\alpha} \lambda_{s1} + \tilde{\alpha} \lambda_{s2}) e_{s1}] + \overline{C}(q_s, q'_s) q'_s + \overline{D} \quad (3.68)$$

3.4 Commande par Mode glissant

3.4.1 Synthèse de la loi de commande pour le robot manipulateur

Synthèse dans le domaine temporel

$$\ddot{q}(t) = -M^{-1}(q)(C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q}) - \tau(t)) \quad (3.69)$$

on prends les états du système $\begin{pmatrix} q \\ \dot{q} \end{pmatrix}$ et $\tau(t)$ le signal commande et posons $e_1 = q_d - q \in \mathbb{R}^{n \times 1}$ et $e_2 = \dot{q}_d - \dot{q} \in \mathbb{R}^{n \times 1}$ l'erreur sur la position et la vitesse des articulation , respectivement q_d/\dot{q}_d représente la position/vitesse désirées pour un contour défini issue du modèle cinématique inverse [annexe 1].

La représentation d'état du système est la suivante :

$$\begin{bmatrix} \dot{e}_1(t) \\ \dot{e}_2(t) \end{bmatrix} = \begin{bmatrix} e_2 \\ \ddot{q}_d + M^{-1}(q)(C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q})) - M^{-1}(q)\tau(t) \end{bmatrix} \quad (3.70)$$

Clairement l'ordre relative du système $r = 2$, donc le choix de la surface de glissement est le suivant :

$$S = e_2 + \lambda e_1 \quad (3.71)$$

Où $\lambda = \begin{bmatrix} \lambda_m & 0 \\ 0 & \lambda_s \end{bmatrix} > 0$,la dérivée de la surface S est :

$$\dot{S} = \dot{e}_2 + \lambda \dot{e}_1 \quad (3.72)$$

Pour assurer la stabilité du système, il faut avoir $S(x)\dot{S}(x) \leq 0$ (pour satisfaire la condition d'attractivité). On peut choisir :

$$\dot{S} = -K \text{sign}(S) = \ddot{q}_d + M^{-1}(q)(C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q})) - M^{-1}(q)\tau(t) + \lambda e_2 \quad (3.73)$$

il en résulte :

$$\tau(t) = M(q)[\ddot{q}_d + \lambda e_2 + K \text{sign}(S)] + C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q}) \quad (3.74)$$

Synthèse dans le domaine de position

Similaire à la commande par MG , la commande du maître est calculé dans le domaine temporel :

$$\tau_m(t) = \tau_1(t) = [M(q)[\ddot{q}_d + \lambda e_2 + K \text{sign}(S)] + C(q, \dot{q})\dot{q}(t) + G(q) + F(t, q, \dot{q})]^T \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (3.75)$$

Calcul de la commande des esclaves la commande des esclaves est calculé en utilisant le système dynamique dans le domaine de position :

$$\overline{M}q_s''(q_m(t)) + \overline{C}q_s'(q_m(t)) + \overline{D} = \tau_s(q_m(t)) \quad (3.76)$$

on prends les états du système $(q_s \quad q_s')$ et $\tau_s(qm(t))$ le signal commande et posons $e_{s1} = q_{ds} - q_s \in \mathfrak{R}^{(n-1) \times 1}$ et $e_{2s} = q'_{ds} - q'_s \in \mathfrak{R}^{(n-1) \times 1}$ l'erreur sur la position et la vitesse des articulation , respectivement q_{ds}/q'_{ds} représente la position/vitesse désirée des esclaves pour un contour défini issu en utilisant la transformation (temps/position) expliqué dans le chapitre 2 (2.4.1) .

La transformation d'état du système dans le domaine de position :

$$\begin{bmatrix} e'_{1s}(qm(t)) \\ e'_{2s}(qm(t)) \end{bmatrix} = \begin{bmatrix} e_{2s} \\ q''_{ds} + \overline{M}^{-1}(q_s)(\overline{C}(q_s, q'_s)q'_s + \overline{D} - \overline{M}^{-1}(q_s)\tau_s(qm(t))) \end{bmatrix} \quad (3.77)$$

La surface de glissement a été choisie comme suit :

$$S_s = e_{s2} + \tilde{\alpha}\lambda_s e_{s1} \quad (3.78)$$

La dérivée relative de la surface S_s est :

$$S'_s = e'_{s2} + \tilde{\alpha}\lambda_s e'_{s1} \quad (3.79)$$

Pour assurer la stabilité du système, il faut avoir $S_s S'_s \leq 0$ (pour satisfaire la condition d'attractivité $S_s \dot{S}_s \leq 0$ dans le temps). On peut choisir :

$$S'_s = -\tilde{\alpha}K_s \text{sign}(S_s) = q''_{ds} + \overline{M}^{-1}(q_s)(\overline{C}(q_s, q'_s)q'_s + \overline{D} - \overline{M}^{-1}(q_s)\tau_s(qm(t))) + \tilde{\alpha}\lambda_s e_{s2} \quad (3.80)$$

il en résulte :

$$\tau_s(t) = \overline{M}(q_m(t))[q''_{sd} + \tilde{\alpha}\lambda_s e_{s2} + \tilde{\alpha}K_s \text{sign}(S_s)] + \overline{C}(q_s, q'_s)q'_s + \overline{D} \quad (3.81)$$

3.4.2 Synthèse de la de commande pour le robot mobile

Synthèse dans le domaine temporel

Les équations différentielles du système étant : Si on prend l'état du système $X = [xy\dot{x}\dot{y}]^T$ Les équations d'état et de sortie du système peuvent donc s'écrire :

$$(S) : \begin{cases} \dot{x} = f(x(t)) + g(x(t))u(t) \\ y = h(x(t)) \end{cases} \quad (3.82)$$

Avec :

$$(S) : \begin{cases} f(x(t)) = \begin{bmatrix} X_3 \\ X_4 \\ \frac{\lambda}{m} \sin \phi \\ -\frac{\lambda}{m} \cos \phi \end{bmatrix} \\ g(x(t)) = \begin{bmatrix} 0 \\ 0 \\ b_1 \cos \phi \\ b_1 \sin \phi \end{bmatrix} \\ h(x(t)) = \begin{bmatrix} x \\ y \end{bmatrix} \end{cases} \quad (3.83)$$

On pose $e_x = x_d - x$ et $e_y = y_d - y$ avec x_d et y_d sont les positions désirées selon les axes OX et OY respectivement dans le plan cartésien. On choisit la surface glissante suivante :

$$s = \dot{e}_x(t) + \dot{e}_y(t) + \lambda_x e_x(t) + \lambda_y e_y(t) \quad (3.84)$$

Avec : $\lambda_x > 0$ et $\lambda_y > 0$.

La stabilité du système commandé est assurée si $s(X) \cdot \dot{s}(X) \leq 0$, le choix le plus souvent utilisé est le suivant :

$$\dot{s} = -K_s \text{sign}(s) = \ddot{x}_d + \ddot{y}_d + \lambda_x \dot{e}_x + \lambda_y \dot{e}_y - \frac{\lambda}{m} (\sin \phi - \cos \phi) - b_1 u_1 (\cos \phi + \sin \phi) \quad (3.85)$$

Par conséquent :

$$u_1(t) = \frac{K_s \text{sign}(s) + \ddot{x}_d + \ddot{y}_d + \lambda_x \dot{e}_x + \lambda_y \dot{e}_y - \frac{\lambda}{m} (\sin \phi - \cos \phi)}{b_1 (\cos \phi + \sin \phi)} \quad (3.86)$$

Synthèse dans le domaine de position

Ici, la stabilité est assurée si : $s(q_m) \cdot \dot{s}(q_m)(X) \leq 0$

La surface de glissement est définie par l'équation :

$$s(q_m) = \tilde{\alpha} \lambda_x e_x(q_m) + \tilde{\alpha} \lambda_y e_y(q_m) + \dot{e}'_x + \dot{e}'_y \quad (3.87)$$

et sa dérivée relative est :

$$\dot{s}'(q_m) = \tilde{\alpha} \lambda_x \dot{e}'_x(q_m) + \tilde{\alpha} \lambda_y \dot{e}'_y(q_m) + \ddot{e}''_x + \ddot{e}''_y \quad (3.88)$$

Afin d'assurer la stabilité par l'inégalité précédente, on prend :

$$\begin{aligned} s'(q_m) &= -\tilde{\alpha}K_s \text{sign}(s) = \tilde{\alpha}\lambda_x e'_x(q_m) + \tilde{\alpha}\lambda_y e'_y(q_m) + x''_d(q_m) + y''_d(q_m) \\ &\quad - \left(\frac{1}{\dot{q}_m^2} \left(\frac{\lambda(q_m)}{m} \sin q_m + b_1 u_1(q_m) \cos q_m - \ddot{q}_m x' \right) \right) + y''_d(q_m) \\ &\quad + \frac{1}{\dot{q}_m^2} \left(\frac{\lambda(q_m)}{m} \cos q_m + b_1 u_1(q_m) \sin q_m - \ddot{q}_m y' \right) \end{aligned} \quad (3.89)$$

D'où la commande résultante est :

$$\begin{aligned} u_1(q_m) &= \dot{q}_m^2 \frac{(\tilde{\alpha}K_s \text{sign}(s) + \tilde{\alpha}\lambda_x e'_x(q_m) + \tilde{\alpha}\lambda_y e'_y(q_m) + x''_d(q_m) + y''_d(q_m))}{b_1(\cos q_m + \sin q_m)} \\ &\quad - \frac{\frac{1}{\dot{q}_m^2} \left(\frac{\lambda(q_m)}{m} (\sin q_m - \cos q_m) - \ddot{q}_m (x' + y') \right)}{b_1(\cos q_m + \sin q_m)} \end{aligned} \quad (3.90)$$

3.5 Synthèse de la commande CAI-MG

L'association de la commande par apprentissage itératif et la commande par modes glissants a donné naissance à une nouvelle loi de commande appelée commande CAI-MG(SM-ILC).

Le principe de cette technique de commande peut se résumer en trois étapes :

- Définir une surface de glissement fonction de l'état du système :

$$S(x_k) = C^T \tilde{x}_k \quad (3.91)$$

Avec : $C^T = [c_1 \dots c_n]$, $\tilde{x}_k = [\tilde{x}_k^1 \dots \tilde{x}_k^n]^T$, $\tilde{x}_k^i = x_k^i - x_d^i$

Où : k désigne la k^{ime} itération et x_d est l'état désiré.

- Choisir une loi de commande à structure variable de la forme :

$$u_k(x_k, t) = u_{eq,k}(x_k, t) + \Delta u_k(x_k, t) \quad (3.92)$$

Où : $u_{eq,k}(t)$ est la commande équivalente (commande d'invariance) du mode de glissement obtenue en mettant $S(x_k) = 0$ et $\dot{x}_k = 0$. On obtient alors l'expression de $u_{eq,k}(t)$ comme suit :

$$u_{eq,k}(x_k, t) = - \left[\left[\left(\frac{\partial S(x_k)}{\partial x_k} g(x_k, t) \right) \right]^{-1} \left\{ \left(\frac{\partial S(x_k)}{\partial x_k} \right)^T f(x_k, t) + \frac{\partial S(x_k)}{\partial t} \right\} \right] \quad (3.93)$$

Avec : $\left[\left(\frac{\partial S(x_k)}{\partial x_k} g(x_k, t) \right) \right]^{-1} \neq 0$

Δu_k est une commande discontinue (commande d'attractivité) qui doit assurer la condition de régime glissant $S(\dot{x}_k)S(x_k) < 0$. La fonction la plus utilisée pour Δu_k est :

$$\Delta u_k = K_s \text{sign}(S(x_k)) \quad (3.94)$$

Où : K_s est un réel positif.

- Remplacer la commande discontinue Δu_k par une autre commande, soit v_k fonction de

la surface de commutation $S(x_k)$ et sa dérivée $S(\dot{x}_k)$, et qui sera évaluée itérativement en utilisant un algorithme de la commande CAI.

La relation ($u_k = u_{eq} + \delta u_k$) devient :

$$\begin{cases} u_k(x_k, t) = u_{eq,k}(x_k, t) + v_k(t) \\ v_k(t) = v_{k-1}(t) + \alpha_k(t)S(x_k) + \beta_k(t)S(\dot{x}_k) \end{cases} \quad (3.95)$$

Où : $\beta_k(t)$ est à déterminer à partir de la condition de convergence et $\alpha_k(t)$ est choisi librement pour agir sur la vitesse de convergence [Ahn Choi, 1993,1].

En remplaçant la relation ($u_k = u_{eq} + \delta u_k$) dans le (syst d'état), on obtient :

$$\begin{cases} \dot{x}_k(t) = F(x_k(t), t) + g(x_k(t), t)v_k(t) \\ y_k(t) = h(x_k(t), t) = c_k x_k \end{cases} \quad (3.96)$$

Avec : $F(x_k(t), t) = f(x_k(t), t) + g(x_k(t), t)u_{eq,k}(x_k(t), t)$.

3.5.1 Étude de convergence [46]

La démonstration de convergence de cet algorithme se fait comme décrit dans [22], et en considérant les hypothèses suivantes :

- H1 : On suppose qu'il existe une commande désirée $u_d(t)$ unique et bornée sur $[0, T]$ qui peut amener la sortie $y_k(t)$ vers la sortie désirée $y_d(t)$. D'où :

$$\begin{cases} \dot{x}_d(t) = F(x_d(t), t) + g(x_d(t), t)v_{d,k}(t) \\ y_d(t) = h(x_d(t), t) = c_k x_d(t) \end{cases} \quad (3.97)$$

- H2 : $G(x_k(t), t)$ est borné par b_G tel que :

$$b_G = \sup_{\forall x_k \in \mathfrak{R}^n} \sup_{t \in [0, T]} \|G(x_k(t), t)\| \quad (3.98)$$

- H3 : Les conditions de Lipschitz sont vérifiées par F et G :

$$\begin{aligned} \|F(x_1(t), t) - F(x_2(t), t)\| &\leq k_F \|x_1(t) - x_2(t)\| \quad \forall t \in [0, T] \quad \text{et} \quad x_1, x_2 \in \mathfrak{R}^n \\ \|G(x_1(t), t) - G(x_2(t), t)\| &\leq k_G \|x_1(t) - x_2(t)\| \\ \|F(x_d(t), t) - F(x_2(t), t)\| &\leq L_F(x_d, t) \|x_d(t) - x_2(t)\| \\ \|G(x_d(t), t) - G(x_2(t), t)\| &\leq L_G(x_d, t) \|x_1(t) - x_2(t)\| \end{aligned} \quad (3.99)$$

Où : L_F et L_G sont des fonctions continues et bornées, k_F et k_G sont des constantes de Lipschitz.

- H4 : $\|c\|$ et $\|\dot{c}\|$ sont bornés par b_c et $b_{\dot{c}}$

On note aussi :

$$b_{u_d} = \sup_{t \in [0, T]} \|u_d\| \quad b_{\dot{x}_k} = \sup_{t \in [0, T]} \|\dot{x}_k\| \quad (3.100)$$

On a :

$$\Delta v_{k+1}(t) = v_d(t) - v_{k+1}(t) = v_d(t) - v_k(t) - \alpha_{k+1}(t)S(x_{k+1}) - \beta_{k+1}(t)\dot{S}(x_{k+1}) \quad (3.101)$$

Pour des des raisons de clarté, on omet le paramètre t .

$$\begin{aligned} \Delta v_{k+1} &= \Delta v_k - \alpha_{k+1}(t)S(x_{k+1}) - \beta_{k+1}(t)\dot{S}(x_{k+1}) \\ &= \Delta v_k - \alpha_{k+1} * c * (x_d - x_{k+1}) - \beta_{k+1} * \dot{c} * (x_d - x_{k+1}) - \beta_{k+1} * c * (\dot{x}_d - \dot{x}_{k+1}) \\ &= \Delta v_k - \alpha_{k+1} * c * (x_d - x_{k+1}) - \beta_{k+1} * \dot{c} * (x_d - x_{k+1}) - \beta_{k+1} * c * (F(x_d, t) + G(x_d, t)v_d \\ &\quad - F(x_{k+1}, t) - G(x_{k+1}, t)v_{k+1}) \\ &= \Delta v_k - \alpha_{k+1}c(x_d - x_{k+1}) - \beta_{k+1}\dot{c}(x_d - x_{k+1}) - \beta_{k+1} * c * (F(x_d, t) - F(x_{k+1}, t)) \\ &\quad - \beta_{k+1}c(G(x_d, t) - G(x_{k+1}, t))v_d + G(x_{k+1}, t)v_{k+1}(1 + \beta_{k+1} * c * G(x_{k+1}, t))\Delta v_{k+1} \\ &= \Delta v_k - \alpha_{k+1} * c * (x_d - x_{k+1}) - \beta_{k+1} * \dot{c} * (x_d - x_{k+1}) - \beta_{k+1} * c * (F(x_d, t) - F(x_{k+1}, t)) \\ &\quad - \beta_{k+1} * c * (G(x_d, t) - G(x_{k+1}, t))v_d \end{aligned} \quad (3.102)$$

En tenant compte des hypothèses H3 et H4, on peut écrire :

$$\rho_{k+1}\|\Delta v_{k+1}\| \leq \|\Delta v_k\| + \alpha_{k+1} * b_c * \|\Delta x_{k+1}\| + \beta_{k+1} * b_{\dot{c}} * (\|\Delta x_{k+1}\| + \beta_{k+1} * b_c * (L_F + L_G v_d)) \quad (3.103)$$

La relation suivante entre $\|\Delta v_{k+1}\|$ et $\|\Delta x_{k+1}\|$ est trouvée dans [9] :

$$\|\Delta x_{k+1}\|_{\lambda} \leq o_1(\lambda^{-1})\|\Delta v_{k+1}\|_{\lambda} \text{ Avec : } o_1(\lambda^{-1}) = \frac{b_G o(\lambda^{-1})}{1 - o(\lambda^{-1})}(k_F + k_G b_{v_d}) \quad (3.104)$$

La norme λ de l'équation (rho*..) est :

$$\rho_{k+1}\|\Delta v_{k+1}\|_{-\lambda} \leq \|\Delta v_k\|_{\lambda} + \alpha_{k+1} * b_c * \|\Delta x_{k+1}\|_{\lambda} + \beta_{k+1} * b_{\dot{c}} * (\|\Delta x_{k+1}\|_{\lambda} + \beta_{k+1} * b_c * (L_F + L_G v_d)) \quad (3.105)$$

En remplaçant $\|\Delta x_{k+1}\|_{\lambda}$ dans (dernière éq), il vient :

$$\begin{aligned} \rho_{k+1}\|\Delta v_{k+1}\|_{\lambda} &\leq \|\Delta v_k\|_{\lambda} + (\alpha_{k+1} * b_c + \beta_{k+1} * b_{\dot{c}} + \beta_{k+1} * b_c * (L_F + L_G v_d))\|\Delta v_{k+1}\|_{\lambda} \\ \rho_{k+1}\|\Delta v_{k+1}\|_{\lambda} &\leq \|\Delta v_k\|_{\lambda} + \varpi o_1(\lambda^{-1})\|\Delta v_{k+1}\|_{\lambda} \\ \rho_{k+1}\|\Delta v_{k+1}\|_{\lambda} &\leq \|\Delta v_k\|_{\lambda} + \varepsilon \end{aligned} \quad (3.106)$$

Avec :

$$\varepsilon = \varpi o_1(\lambda^{-1})\|\Delta v_{k+1}\|_{\lambda} \quad \text{et} \quad \varpi = \sup_{t \in [0, T]} (\alpha_{k+1} * b_c + \beta_{k+1} * b_{\dot{c}} + \beta_{k+1} * b_c * (L_F + L_G v_d)) \quad (3.107)$$

Si on choisit λ suffisamment grand tel que :

$$\lim_{k \rightarrow \infty} o_1(\lambda^{-1}) \rightarrow 0 \quad (3.108)$$

$$\text{Et : } \frac{(1-\gamma)}{\|1+\beta_{k+1}cG_{k+1}\|} < 1$$

Alors : $\|\Delta v_k\|_\lambda \rightarrow 0$ quand $k \rightarrow \infty$. Ceci signifie que $v_k \rightarrow v_d$.

On a donc :

$$\lim_{k \rightarrow \infty} \|\Delta v_k\|_\lambda \rightarrow 0 \quad (3.109)$$

De (relation finale) , il découle

$$\lim_{k \rightarrow \infty} \|\Delta x_{k+1}\|_\lambda \rightarrow 0 \quad (3.110)$$

Par conséquent :

$$\lim_{k \rightarrow \infty} \|\Delta y_{k+1}\|_\lambda = \lim_{k \rightarrow \infty} \|e_k\|_\lambda \rightarrow 0 \quad (3.111)$$

L'application de cet algorithme au robot mobile donne pour :

$$e_k(t) = y_d(t) - y_k(t) = c\tilde{x} = S(\tilde{x}_k(t)) \quad (3.112)$$

avec la condition de convergence est donc :

$$\left\| \frac{1}{1 + \beta_k(t)cg(x_k(t), t)} \right\| \leq 1 \quad (3.113)$$

3.5.2 Synthèse de la loi de commande pour le robot manipulateur

On reprend la représentation d'état du système établie précédemment à l'itération k :

$$\begin{aligned} \begin{bmatrix} \dot{e}_{1k}(t) \\ \dot{e}_{2k}(t) \end{bmatrix} &= \begin{bmatrix} e_{2k} \\ \ddot{q}_d + M^{-1}(q_k)(C(q_k, \dot{q}_k)\dot{q}_k(t) + G(q_k) + F(t, q_k, \dot{q}_k)) - M^{-1}(q_k)\tau_k(t) \end{bmatrix} \\ &= F(q_k(t), t) + g(q_k(t), t)\tau_k(t) \end{aligned} \quad (3.114)$$

Avec les états du système à l'itération k sont $\begin{pmatrix} q_k & \dot{q}_k \end{pmatrix}$ et $\tau_k(t)$ le signal commande et $e_{1k} = q_d - q_k \in \mathfrak{R}^{n \times 1}$ et $e_{2k} = \dot{q}_d - \dot{q}_k \in \mathfrak{R}^{n \times 1}$ l'erreur sur la position et la vitesse des articulations avec :

$$\begin{cases} F(q_k(t), t) = \begin{bmatrix} e_{2k} \\ q_d + M^{-1}(q_k)(C(q_k, \dot{q}_k)\dot{q}_k(t) + G(q_k) + F(t, q_k, \dot{q}_k)) \end{bmatrix} \\ g(q_k(t), t) = \begin{bmatrix} 0 \\ -M^{-1} \end{bmatrix} \\ \tau_k(q_k, t) = u_{eq,k}(q_k, t) + v_k(t) \\ v_k(t) = v_{k-1}(t) + \alpha_k(t)S(q_k) + \beta_k(t)\dot{S}(q_k) \end{cases} \quad (3.115)$$

Tel que :

$$\begin{cases} \|A\|_\infty = \|(I - \beta_k(t)M(q_k)^{-1})^{-1}\|_\infty \leq 1 \\ \|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}| \end{cases} \quad (3.116)$$

La surface de glissement

On prend comme surface de glissement à l'itération k :

$$S(q_k) = \dot{e}_{2k} + \lambda e_{1k} \quad (3.117)$$

La commande équivalente

$$u_{eq,k}(q_k, t) = M(q_k)[\ddot{q}_d + \lambda e_{2k}] + C(q_k, \dot{q}_k)\dot{q}_k(t) + G(q_k) + F(t, q_k, \dot{q}_k) \quad (3.118)$$

3.5.3 Synthèse de la loi de commande pour le robot mobile

La surface de glissement pour l'itération k , est :

$$s_k = \lambda_x e_{x_k} + \lambda_y e_{y_k} + \dot{e}_{x_k} + \dot{e}_{y_k} \quad (3.119)$$

La commande est :

$$\begin{cases} u_{1k}(t) = u_{1eq_k}(t) + v_k(t) \\ u_{1eq_k}(t) = \frac{(\ddot{x}_d + \ddot{y}_d + \lambda_x \dot{e}_{x_k} + \lambda_y \dot{e}_{y_k} - \frac{\lambda}{m}(\sin \phi_k - \cos \phi_k))}{b_1(\cos \phi_k + \sin \phi_k)} \\ v_k(t) = v_{k-1}(t) + \alpha_k(t)s_k(q_k) + \beta_k(t)\dot{s}_k(q_k) \end{cases} \quad (3.120)$$

Conclusion

Dans ce chapitre, nous avons exposé le concept de formulation d'un problème de suivi de contour dans le domaine de position. Nous avons introduit des loi de contrôle comme une alternative de domaine de position à la loi de contrôle PSC de domaine de temps. Cette approche fera l'objet de notre étude dans les prochains chapitres et sera comparée aux techniques de contrôle temporelles existantes.

CHAPITRE 4

SIMULATION ET RÉSULTATS

Chapitre 4 : Simulation et Résultats

4.1 Configuration de la simulation

4.1.1 Robot Manipulateur Planaire

Un robot manipulateur virtuel composé de 3 articulations rotoides illustré dans la figure(4.1) est utilisé pour la simulation. Le robot manipulateur est supposé être composé de 3 segments de différentes tailles qui sont commandés avec le même type d'actionneurs. Ce type de configuration a été choisi parce qu'il était assez complexe pour montrer les capacités de la loi des commandes proposée. . Les paramètres structurels du robot manipulateur choisi figurent dans le tableau (4.1).

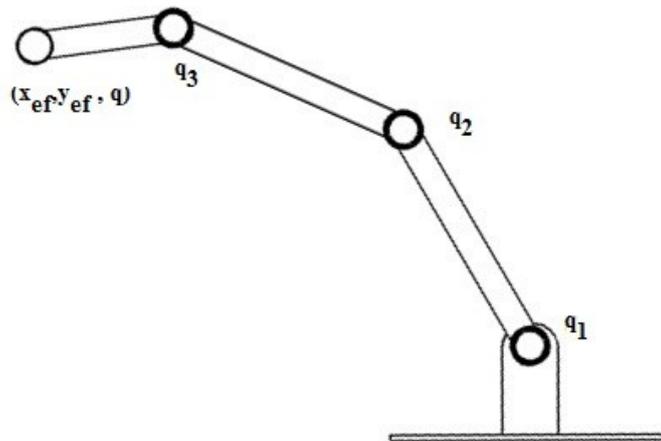


FIGURE 4.1 – Robot Manipulateur Planaire à 3.d.d.l[7]

Corps	Mass $m_i(\text{kg})$	Longueur $l_i(\text{m})$	Centre $r_i(\text{m})$	Inertie $I_i(\text{kgm}^2)$
1	1.00	0.50	0.25	0.10
2	1.00	0.50	0.25	0.10
3	0.50	0.30	0.15	0.05

TABLE 4.1 – Paramètres structurels du robot manipulateur série[5]

4.1.2 Robot Mobile

Soit le robot mobile de type uni cycle représenté dans la section, décrit dans la figure(4.2) dont le modèle dynamique est défini dans l'équation(-). Les paramètres mécaniques du robot sont regroupés dans le tableau (4.2)

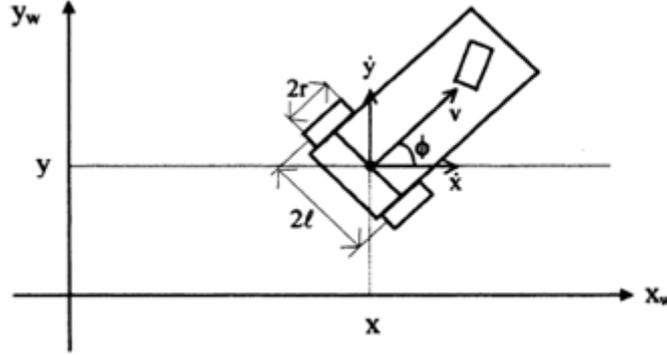


FIGURE 4.2 – Robot Mobile uni-cycle [43]

Masse m[kg]	Inertie I[kg.m ²]	Distance entre une roue et l'axe central du véhicule l[m]
4	3.75	0.15

TABLE 4.2 – Paramètres du Robot Mobile [43]

4.2 Contours de références utilisés durant les simulations

Pour garantir que les trajectoires de référence soient lisses pour toutes les articulations du robot manipulateur, un polynôme d'ordre 5 est utilisé pour définir pour le contour désiré, ses signaux de position, vitesse et accélération associés [10]. Le polynôme d'ordre est défini comme fonction du temps par :

$$r(t) = 10\left(\frac{t}{T}\right)^3 - 15\left(\frac{t}{T}\right)^4 + 6\left(\frac{t}{T}\right)^5 \quad (4.1)$$

Où : t : le temps et T : la durée totale accordée à l'opération ou au mouvement .Puis, la vitesse et l'accélération sont donnés par :

$$\dot{r}(t) = \frac{1}{T}\left\{30\left(\frac{t}{T}\right)^2 - 60\left(\frac{t}{T}\right)^3 + 30\left(\frac{t}{T}\right)^4\right\} \quad (4.2)$$

$$\ddot{r}(t) = \frac{1}{T^2}\left\{60\left(\frac{t}{T}\right) - 180\left(\frac{t}{T}\right)^2 + 120\left(\frac{t}{T}\right)^3\right\} \quad (4.3)$$

4.2.1 Contours Linéaires

Avec les équations (4.2)-(4.3), un contour linéaire qu'on désire imposer au niveau de l'effecteur peut être défini par :

$$P_d = (P_f - P_i)r(t) \quad (4.4)$$

Où : P_d est le contour désiré, exprimé par des coordonnées cartésiennes (x_{ef}, y_{ef}) , et P_i et P_f sont les positions initiale et finale du contour. L'orientation (q_{ef}) de l'effecteur est maintenue constante pour chaque segment. Les équations de la cinématique inverse du manipulateur robotique (annexe) sont utilisées pour calculer la position et les vitesses des des axes nécessaires pour suivre le contour défini.

4.2.2 Contour Ligne

Les figures (4.3) et (4.4) montrent le contour ligne ainsi que les trajectoires de niveau de l'effecteur et de l'articulation. Le tableau (4.3) montre respectivement les positions finale et initiale sur le niveau de l'effecteur final . Nous avons pris pour cette simulation une fréquence d'échantillonnage $1000H_z$, et un temps de simulation $t_s = 1.5s$.

	P_i	P_f
$\mathbf{x}_{ef}(\mathbf{m})$	0.50	0.70
$\mathbf{y}_{ef}(\mathbf{m})$	0.50	0.70
$\mathbf{q}_{ef}(\text{rad})$	1.047	1.047

TABLE 4.3 – Positions initiales et finales de l'effecteur pour le contour ligne.

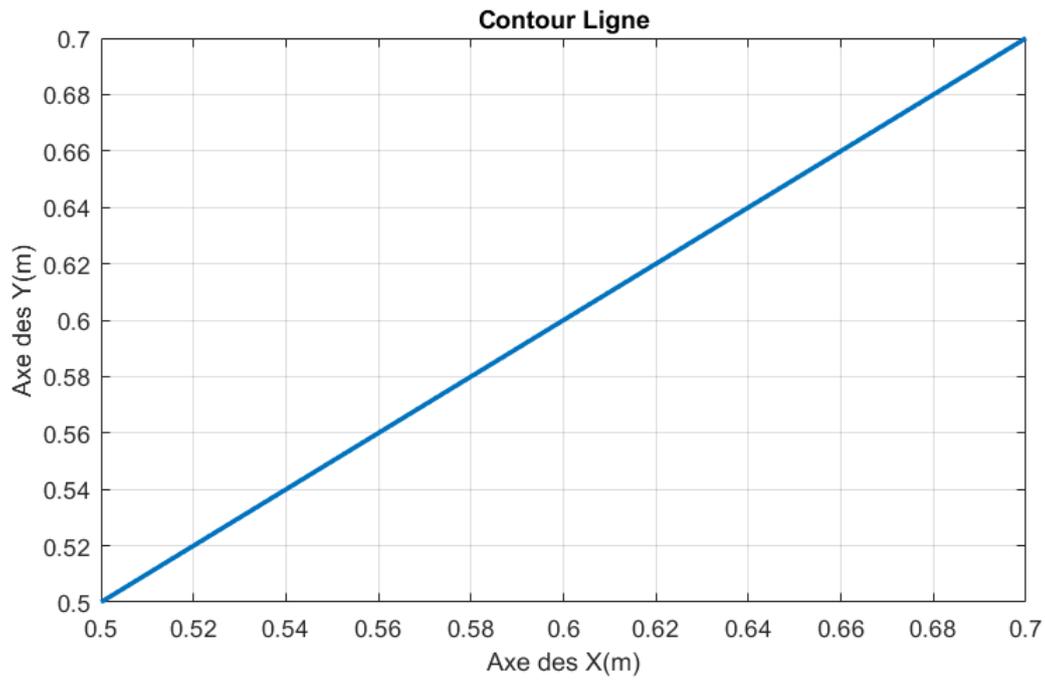


FIGURE 4.3 – Contour Ligne (Espace de travail)

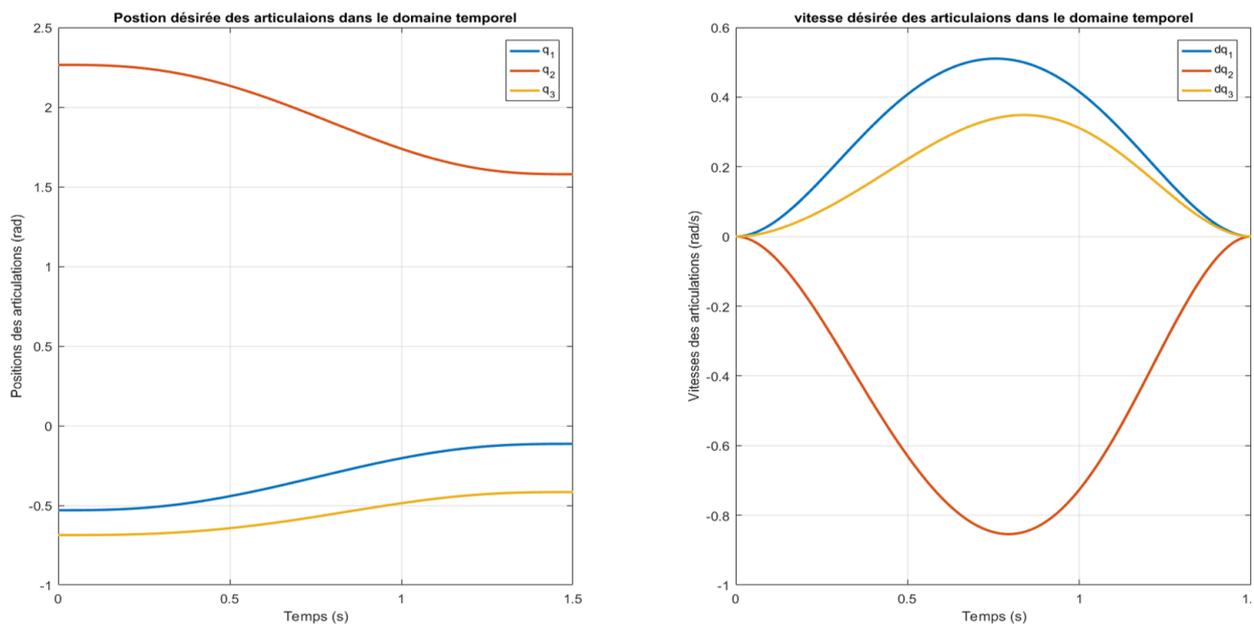


FIGURE 4.4 – Positions et vitesses désirées des articulations

4.2.3 Contours non linéaires

Contours pour robot planaire

Pour définir le contour non linéaire, un paramètre angulaire doit être introduit :

$$\theta(t) = r(t)(\theta_f - \theta_i) \quad (4.5)$$

Le contour circulaire est défini comme suit :

$$\begin{cases} x_{ef}(t) = x_c + R \cdot \cos(\theta(t)) \\ y_{ef}(t) = y_c + R \cdot \sin(\theta(t)) \end{cases} \quad (4.6)$$

Avec (x_c, y_c) le centre du cercle et R son rayon .

Pour cette simulation ,nous avons pris une fréquence d'échantillonnage $1000Hz$, et un temps de simulation $t_s = 5s$.

x_c	y_c	R
0.0[m]	0.0[m]	0.6[m]

TABLE 4.4 – Les paramètres du contour circulaire

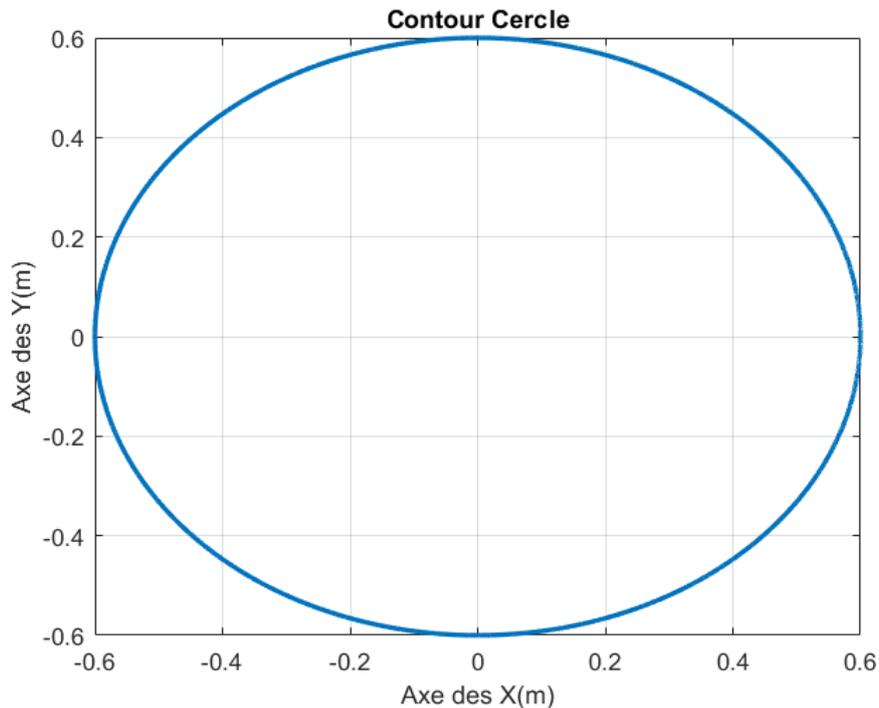


FIGURE 4.5 – Contour Cercle (Espace de travail)

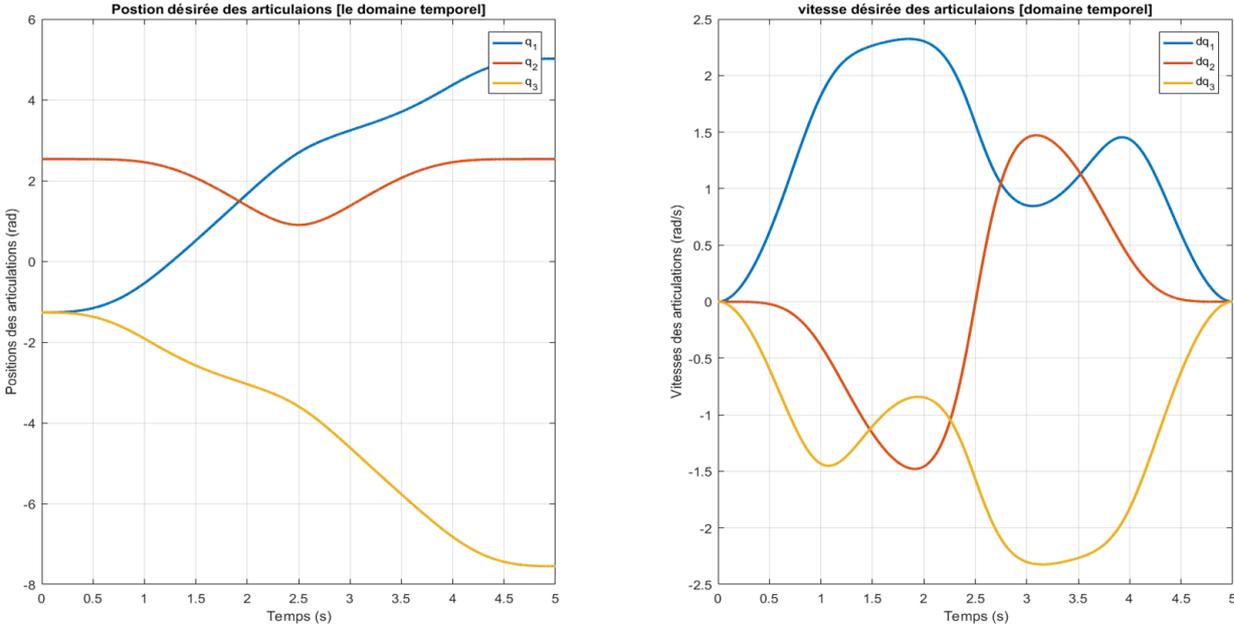


FIGURE 4.6 – Positions et vitesses désirées des articulations (contour circulaire)

4.3 Transformation de trajectoires

Comme mentionné précédemment, lors de l'élaboration de la commande dans le domaine de position, les fonctions temporelles traduisant la dynamique de référence des esclaves doivent être transformées en des fonctions dépendant de la position du maître.

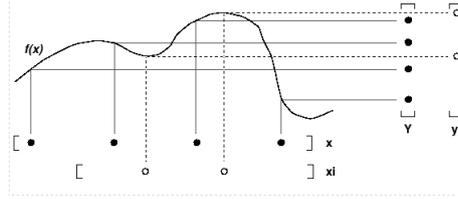


FIGURE 4.7 – Lookup Table (interp1) (The MathWorks, Inc)

Pour effectuer les simulations suivantes, « lookup tables » sont utilisées pour la redéfinition des trajectoires désirées dans le domaine des positions. Plus précisément, les trajectoires désirées sont initialement définies comme fonctions temporelles. Ensuite, ces trajectoires sont interpolées linéairement en fonction de la trajectoire réelle de l'agent maître résultant de son asservissement dans le domaine temporel. Le concept de cette transformation est illustré dans la figure (4.7)[17].

Les contours résultants dérivés de la procédure ci-dessus peuvent être observés dans les figures suivantes. Les figures (4.8) et (4.9) contiennent les trajectoires résultantes pour les contours ligne et cercle respectivement.

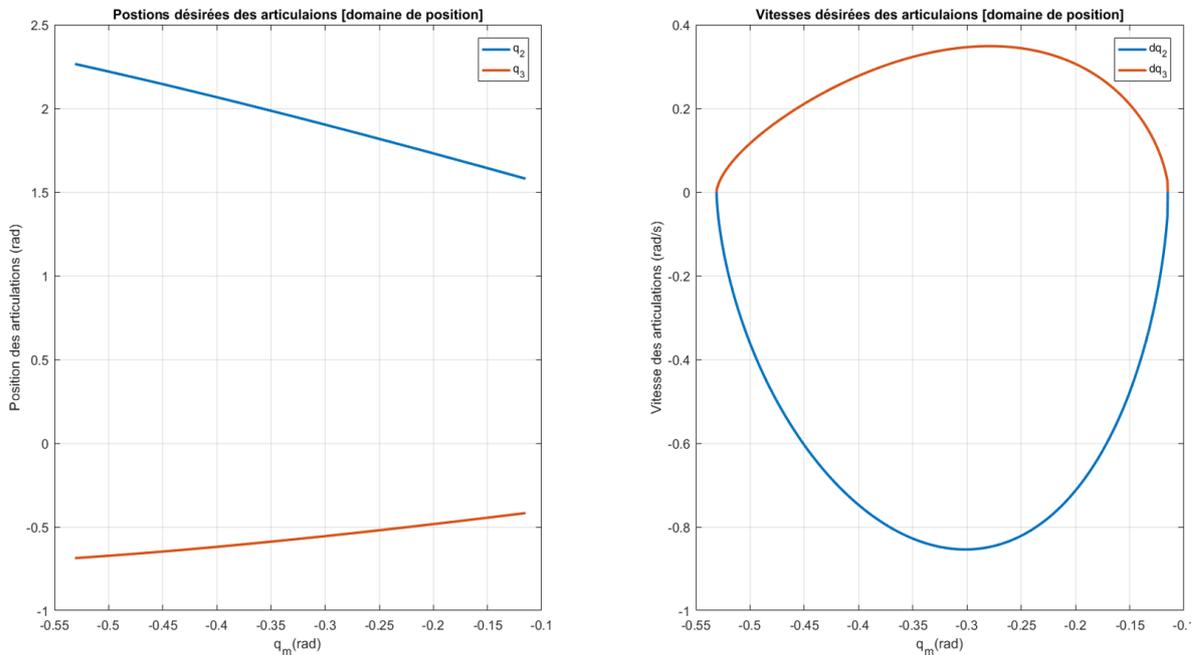


FIGURE 4.8 – Positions et vitesses désirées des articulations (contour ligne)

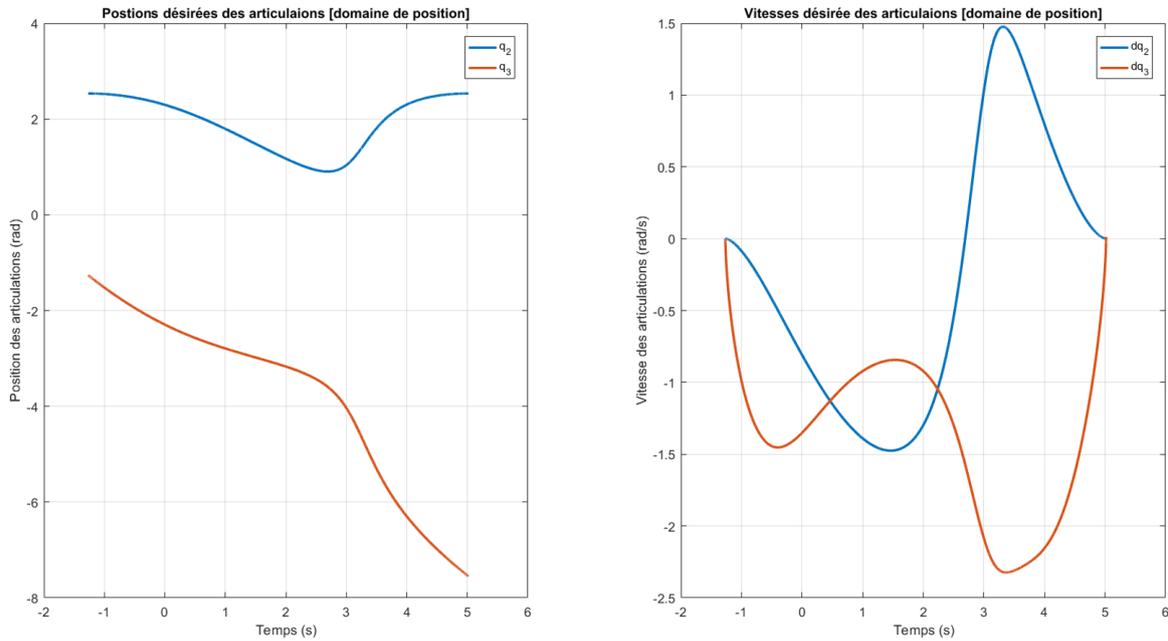


FIGURE 4.9 – Positions et vitesses désirées des articulations (contour cercle)

Contours pour robot mobile

Pour des raisons de faisabilité des contours désirés et d'évitement des singularités liées au signal de commande, nous imposerons des contours incomplets de sorte que la variable maître soit monotone et ne prenne pas de valeurs annulant le dénominateur de la loi de commande.

Contour Arc

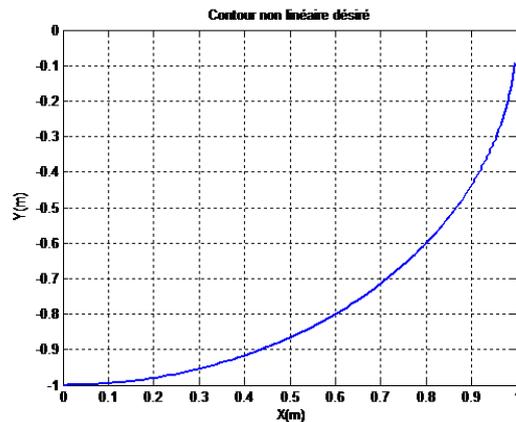


FIGURE 4.10 – Contour Arc (Espace de travail)

Contour Sinus

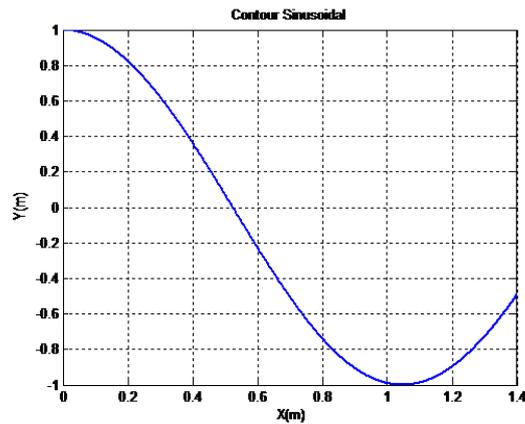


FIGURE 4.11 – Contour Sinus (Espace de travail)

4.4 Résultats des simulations

4.4.1 Robot manipulateur planaire

Pour ces simulations , nous avons ignoré les effets des frottements sur les articulations .
Le modèle dynamique du système devient :

$$M(q)\ddot{q}(t) + C(q, \dot{q})\dot{q}(t) + G(q) = \tau(t) \quad (4.7)$$

La commande PD-MG

On reprend les résultats établies lors du chapitre 3 :

Commande dans le domaine temporel

$$\begin{cases} \tau(t) = K_P e(t) + K_D \dot{e}(t) + H \text{sat}(s, \Phi) \\ s = \dot{e}_s(t) + \lambda e(t) \end{cases} \quad (4.8)$$

Commande dans le domaine de position

$$\begin{cases} \tau_m(t) = k_{pm}e_m(t) + k_{dm}\dot{e}_m(t) + h_m \text{sat}(s_m, \Phi_m) \\ \tau_s(q_m(t)) = K_{Ps}e_s(q_m) + K_{Ds}e'_s(q_m) + H_s \text{sat}(s, \Phi_s) \\ s_s = e'_s(q_m) + \lambda_s e_s(q_m) \end{cases} \quad (4.9)$$

tel que :

$$K_p = \begin{bmatrix} k_{pm} & 0 \\ 0 & K_{Ps} \end{bmatrix} \quad K_d = \begin{bmatrix} k_{dm} & 0 \\ 0 & K_{Ds} \end{bmatrix} \quad H = \begin{bmatrix} h_m & 0 \\ 0 & H_s \end{bmatrix} \quad \Phi = \begin{bmatrix} \Phi_m \\ \Phi_s \end{bmatrix} \quad (4.10)$$

les paramètres du régulateur :

les paramètres choisis sont regroupés dans le tableau suivant :

	Contour ligne	Contour Cercle
K_P	$\text{diag}([7700; 7500; 2100])$	$\text{diag}([6800; 4000; 4000])$
K_D	$\text{diag}([2000; 1200; 1200])$	$\text{diag}([4800; 7350; 8500])$
H	$\text{diag}([25; 15; 10])$	$\text{diag}([10; 10; 10])$
λ	$[65; 30; 20]$	$[20; 20; 20]$
Φ	$[0.25; 0.3; 0.25]$	$[0.08; 0.08; 0.08]$

TABLE 4.5 – Paramètres du régulateur PD-MG

Résultats de suivi du contour linéaire

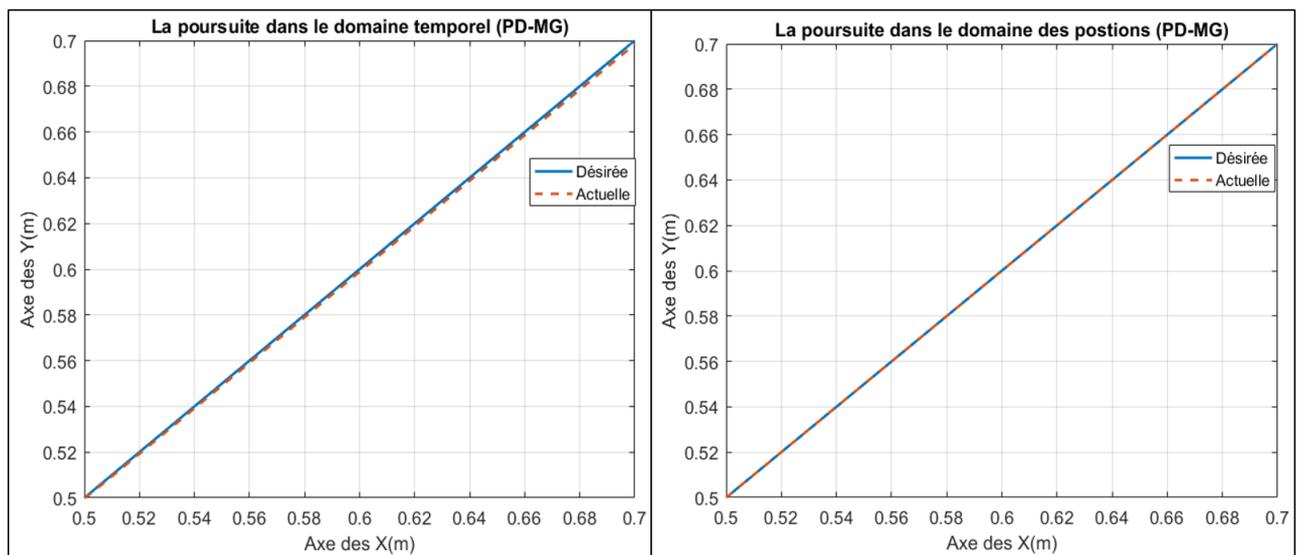


FIGURE 4.12 – Résultat de suivi du contour (contour Ligne)

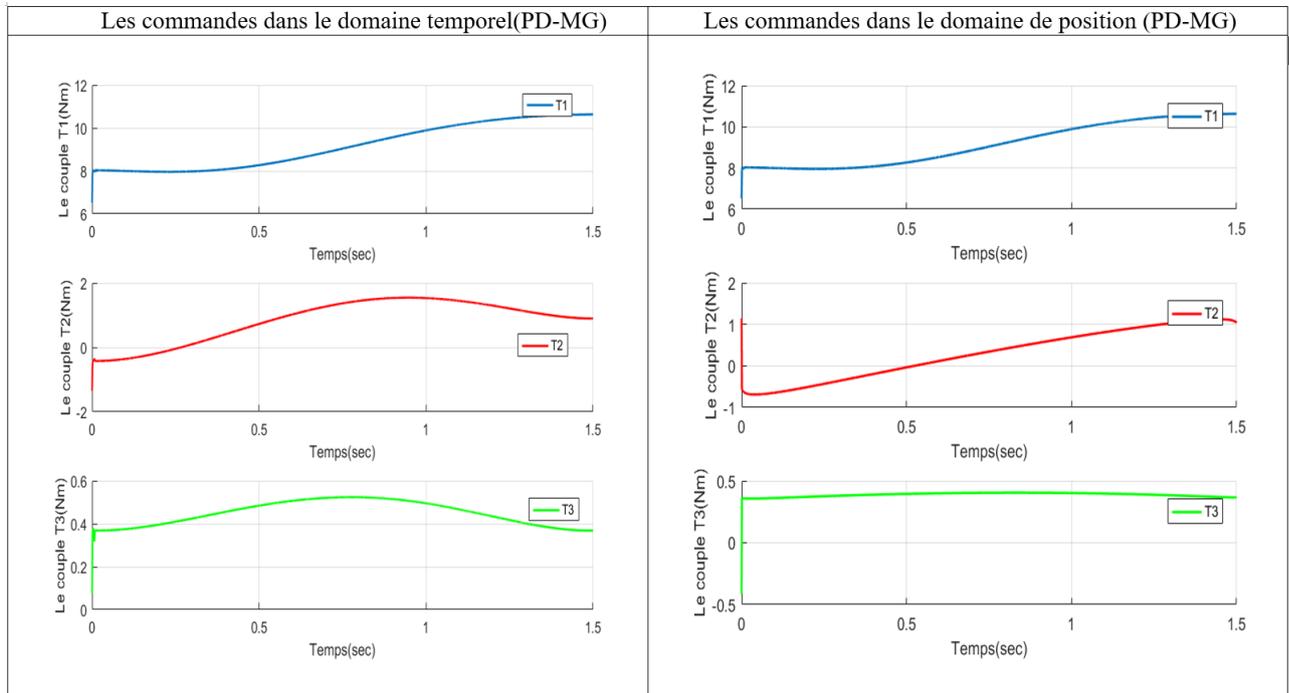


FIGURE 4.13 – Commandes dans domaine temporel/position (Contour Ligne)

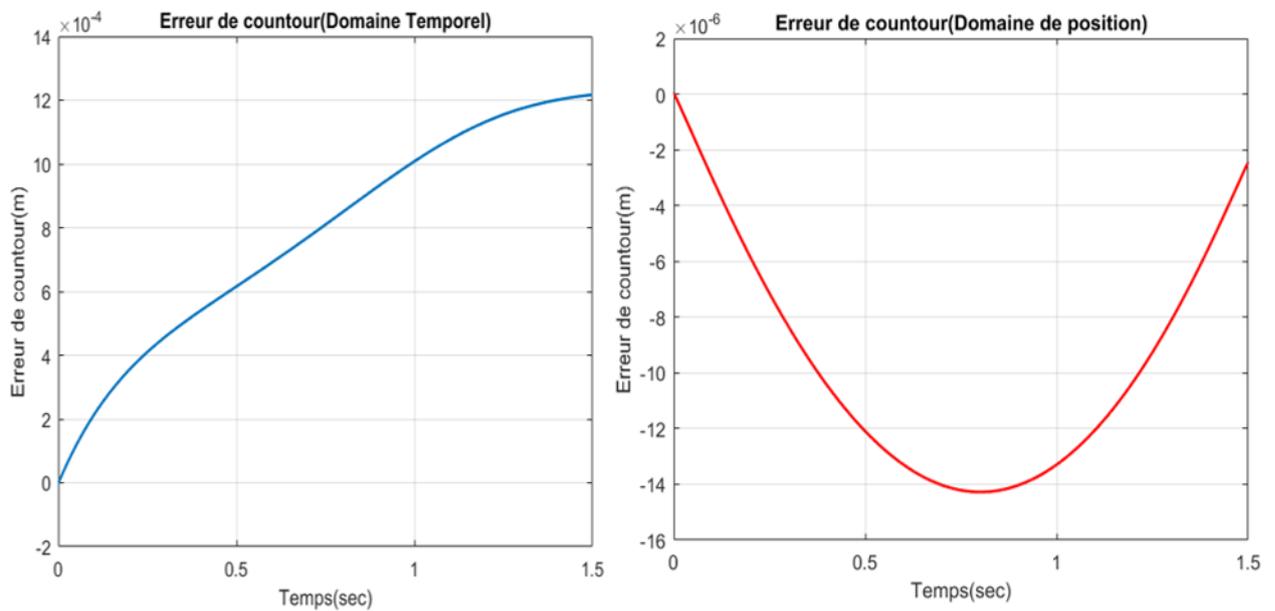


FIGURE 4.14 – L'erreur du Contour Ligne (PD-MG)

Commentaires

Les commandes dans le domaine de position ont pu produire une erreur de contour moyenne très petite par rapport aux commandes dans le domaine temporel approximativement 80 fois mieux que le PD-MG(temporel)(figure(4.14),ceci est due aux informations sur la dynamique du maître utilisé lors du calcul de la commande PD-MG(position) , on remarque aussi que les commandes (τ_2) (τ_3) (figure(4.13) présentent une variation brusque au début et à la fin du trajectoire ceci est expliqué par l'effet de l'action dérivé au début sachant que l'effecteur du robot(TABLE 4.1) déplace une distance $d \simeq 0.894m$ avec une vitesse relativement grande pour une durée de simulation $t_s = 1.5sec$.

Résultats de la simulation pour le contour circulaire

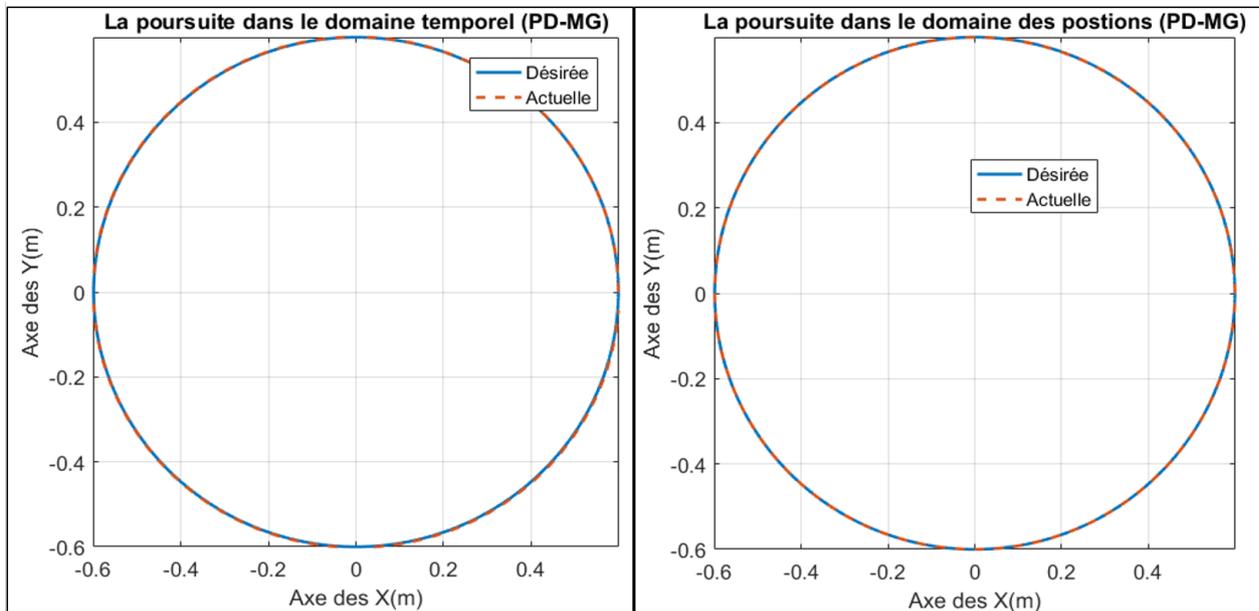


FIGURE 4.15 – Résultat de suivi du contour (contour Cercle)

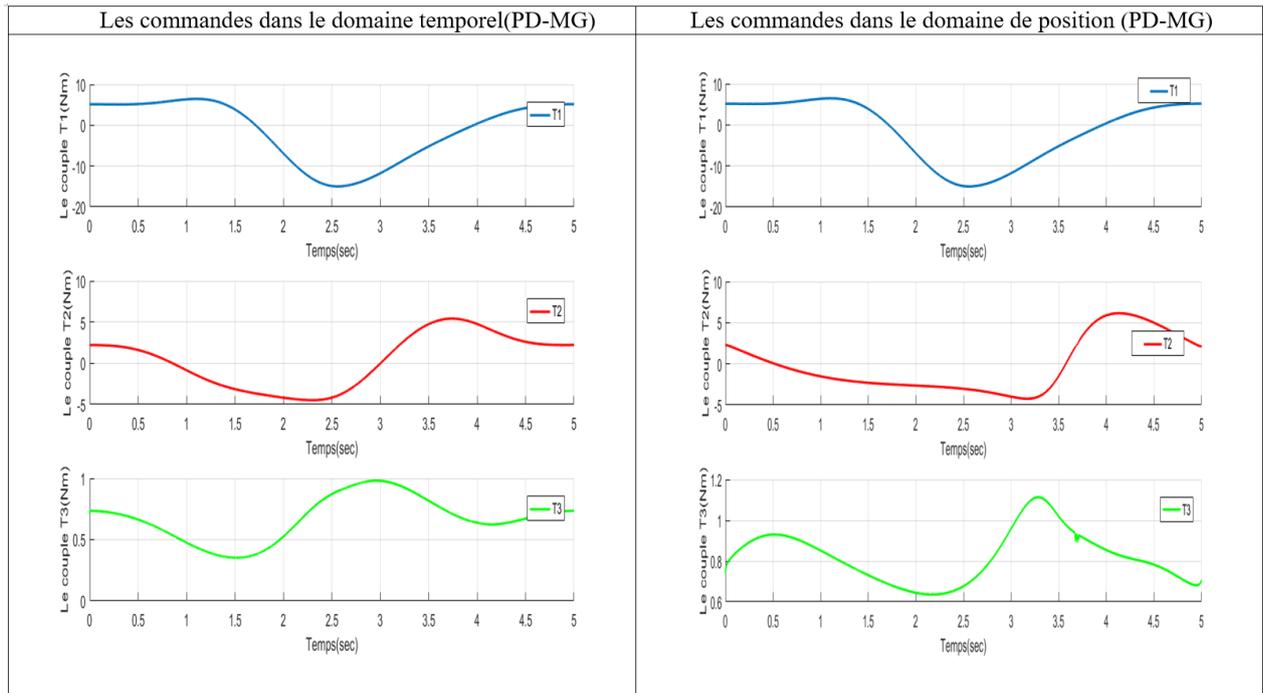


FIGURE 4.16 – Commandes PD-MG dans domaine temporel/position (Contour Cercle)

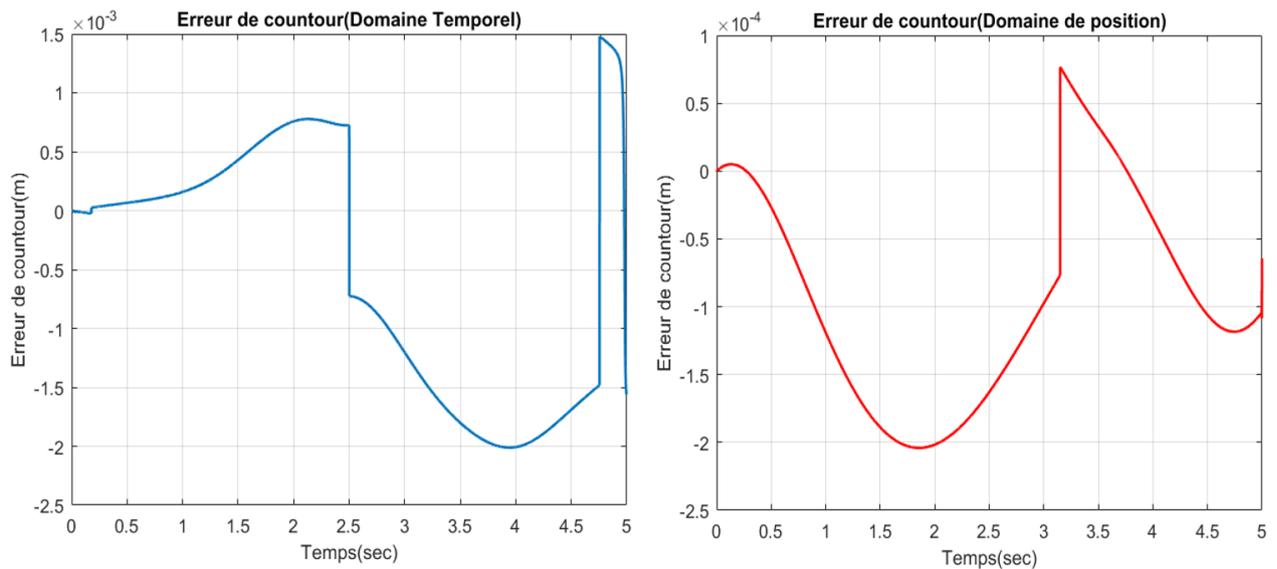


FIGURE 4.17 – L'erreur du Contour Cercle (PD-MG)

Commentaires

On remarque qu'avec le même signal commande on a pu diminuer la moyenne l'erreur de contour approximativement 7 fois (figure(4.17)), mais le choix des gains pour les régulateurs pour assurer la stricte monotonie de la trajectoire d'articulation maitre était un peu délicat par rapport au cas du contour ligne .Cette condition est nécessaire pour avoir une bonne interpolation lors du passage temps/position.

La commande par Forwarding

On reprend les résultats établies lors du chapitre 3 :

Commande dans le domaine temporel

$$\tau(t) = M(q)[\ddot{q}_d - \lambda_1 \lambda_2^{-1} e_1 - (\lambda_1 (\lambda_2^2)^{-1} + \lambda_1 + \lambda_2) e_2] + C(q, \dot{q}) \dot{q}(t) + G(q) = \begin{bmatrix} \tau_1(t) \\ \tau_2(t) \\ \tau_3(t) \end{bmatrix} \quad (4.11)$$

Commande dans le domaine de position

$$\begin{cases} \tau_m(t) = \tau_1(t) \\ \tau_s(t) = \bar{M}(q_m(t))[q''_{sd} + \lambda_{s1} \lambda_{s2}^{-1} e_{s1} + (\lambda_{s1} (\lambda_{s2}^2)^{-1} + \lambda_{s1} + \lambda_{s2}) e_{s1}] + \bar{C}(q_s, q'_s) q'_s + \bar{D} \end{cases} \quad (4.12)$$

tel que :

$$\lambda_1 = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \lambda_{s1} \end{bmatrix} \lambda_2 = \begin{bmatrix} \Lambda_2 & 0 \\ 0 & \lambda_{s1} \end{bmatrix} \quad (4.13)$$

les paramètres du régulateur :

les paramètres choisis sont regroupés dans le tableau suivant :

	Contour ligne	Contour Cercle
Λ_1	10	10
λ_{s1}	$\begin{bmatrix} 100\tilde{\alpha} & 0 \\ 0 & 100\tilde{\alpha} \end{bmatrix}$	$\begin{bmatrix} 5\tilde{\alpha} & 0 \\ 0 & 5\tilde{\alpha} \end{bmatrix}$
Λ_2	2	1
λ_{s2}	$\begin{bmatrix} 2\tilde{\alpha} & 0 \\ 0 & 2\tilde{\alpha} \end{bmatrix}$	$\begin{bmatrix} 1\tilde{\alpha} & 0 \\ 0 & 1\tilde{\alpha} \end{bmatrix}$

TABLE 4.6 – Paramètres du régulateur Forwarding

Résultats de suivi du contour ligne

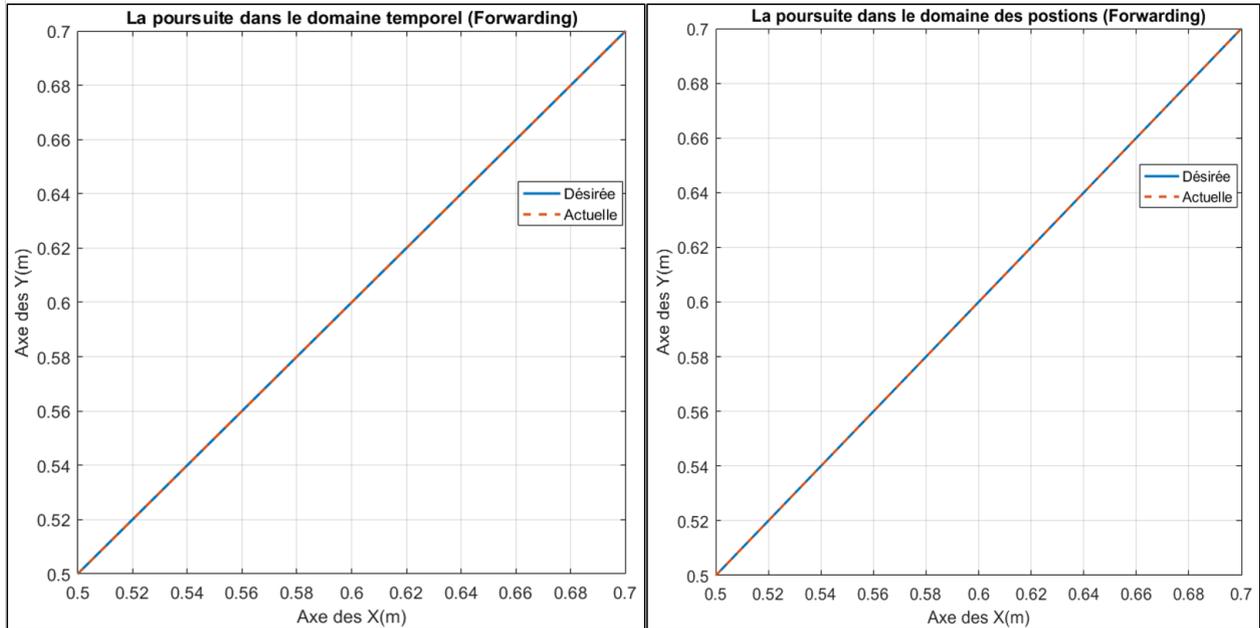


FIGURE 4.18 – Résultat de suivi du contour (contour Ligne)

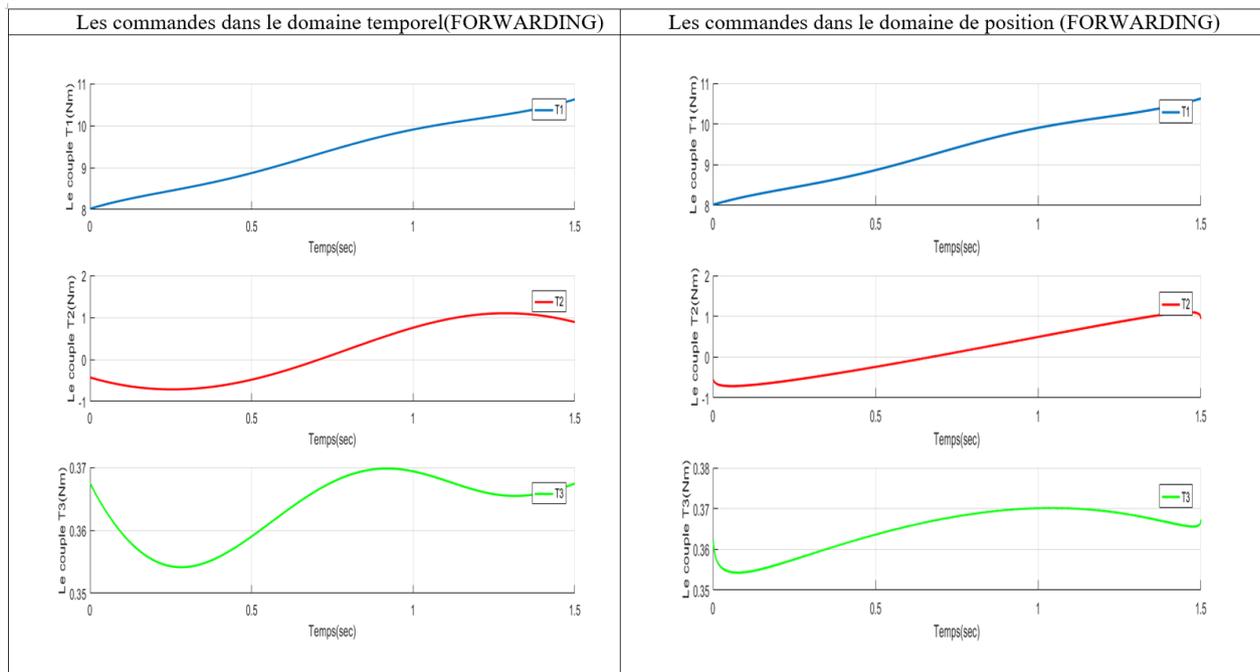


FIGURE 4.19 – Commandes Forwarding dans domaine temporel/position (Contour Ligne)

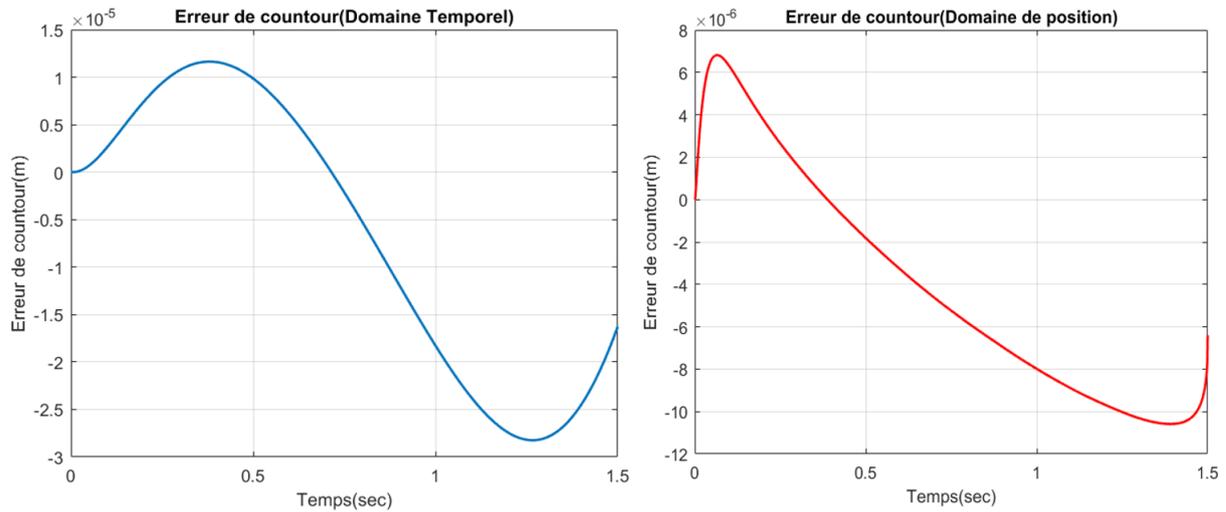


FIGURE 4.20 – L'erreur du Contour Ligne (Forwarding)

Commentaires

Les commandes dans le domaine de position ont pu produire une erreur de contour moyenne légèrement petite par rapport aux commandes (diminution de 60% (figure 4.20)) ,tout en gardant pratiquement le même signal commande(4.17) ,ce résultat reste intéressant vu qu'on obtient les même performance avec le PD-MG dans le domaine temporel (figures (4.13) et (4.14)).

Résultats de la simulation pour le contour circulaire

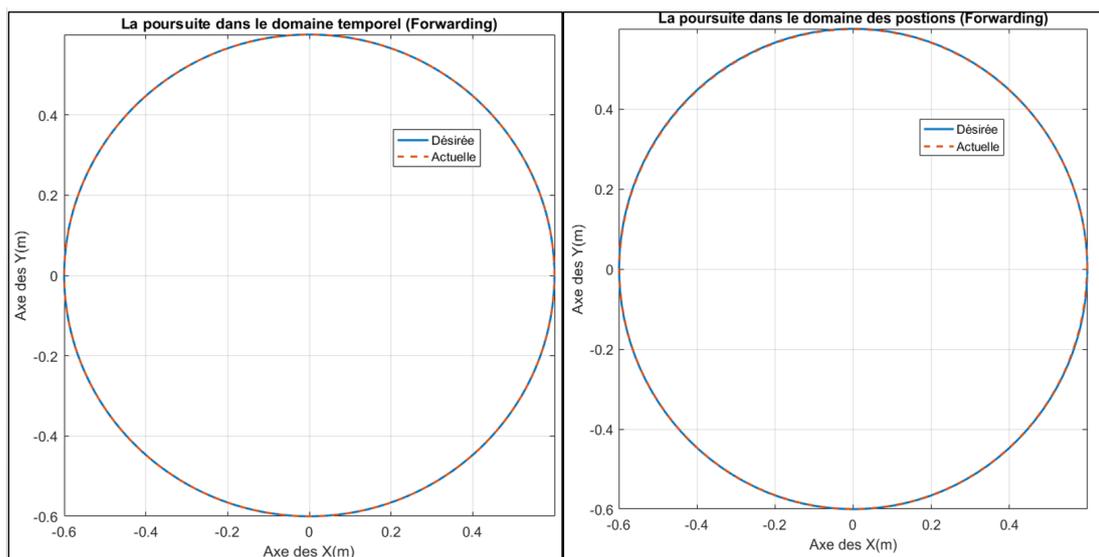


FIGURE 4.21 – Résultat de suivi du contour (contour Cercle)

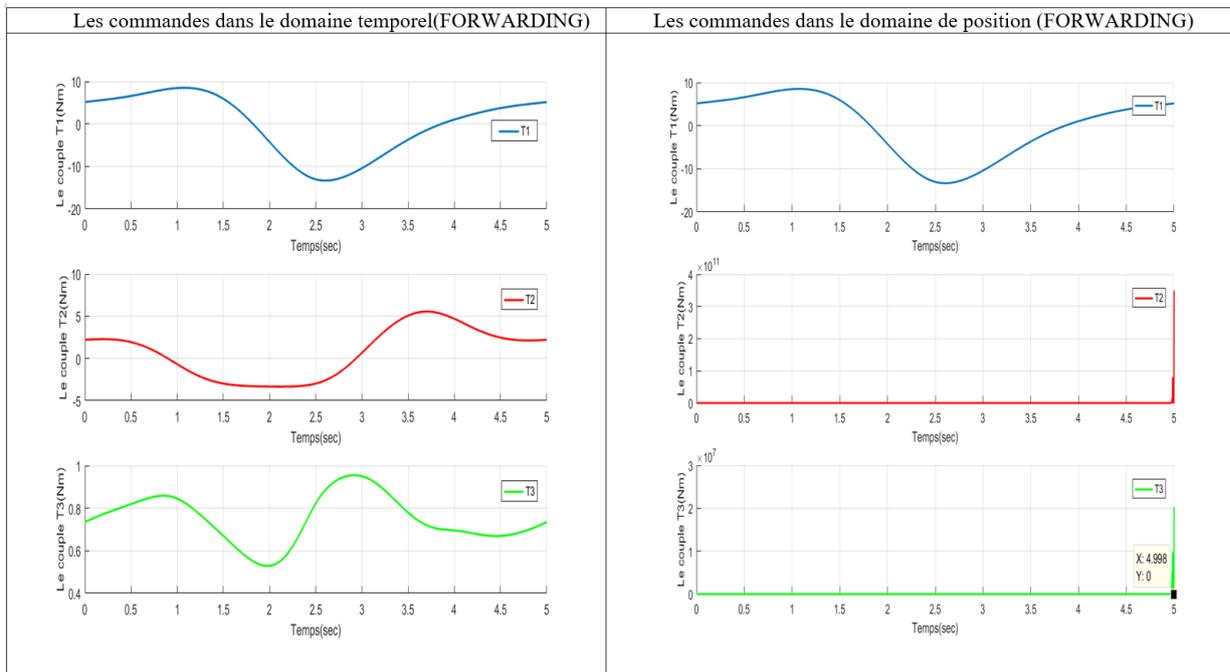


FIGURE 4.22 – Commandes par Forwarding dans domaine temps/position (Contour Cercle)

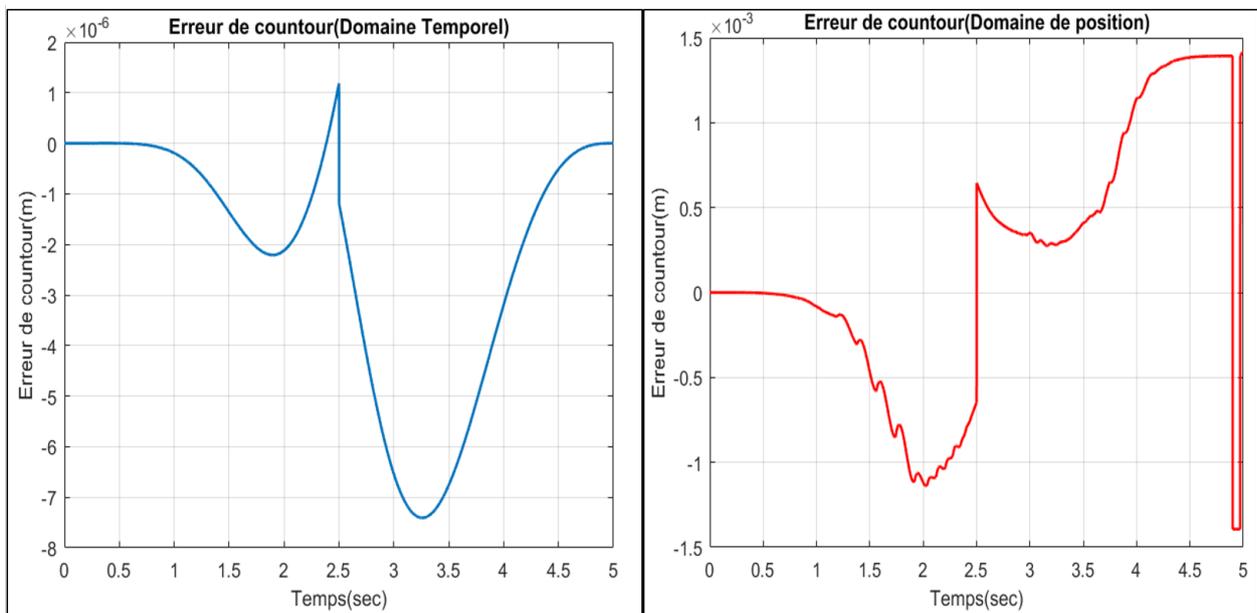


FIGURE 4.23 – L'erreur du Contour Cercle (Forwarding)

Commentaires

On remarque que le mauvais choix des gains a produit une erreur de transformation (Temps/Position) qui a donné un \dot{q}_m considéré nulle par le calculateur ,d'où la fonction interp1 de Matlab à donner une mauvais interpolation au niveau des derniers points d'échantillonnage ce qui produit des commandes (τ_2) (τ_3) (figure(4.22) dans le domaine de position qui sont inadmissibles à l'échantillonne ($i = 5000$ ou $t = 5s$).

La commande par mode glissant

On reprend les résultats établies lors du chapitre 3 :

Commande dans le domaine temporel

$$\tau(t) = M(q)[\ddot{q}_d + \lambda e_2 + K_{sat}(S, \Phi)] + C(q, \dot{q})\dot{q}(t) + G(q) = \begin{bmatrix} \tau_1(t) & \tau_2(t) & \tau_3(t) \end{bmatrix}^T \quad (4.14)$$

Commande dans le domaine de position

$$\begin{cases} \tau_m(t) = \tau_1(t) \\ \tau_s(t) = \bar{M}(q_m(t))[q''_{sd} + \lambda_s e_{s2} + K_s sat(S_s, \Phi_s)] + \bar{C}(q_s, q'_s)q'_s + \bar{D} \end{cases} \quad (4.15)$$

tel que :

$$\lambda = \begin{bmatrix} \lambda_m & 0 \\ 0 & \lambda_s \end{bmatrix} \quad K = \begin{bmatrix} k_m & 0 \\ 0 & K_s \end{bmatrix} \quad \Phi = \begin{bmatrix} \phi_m \\ \Phi_s \end{bmatrix} \quad (4.16)$$

les paramètres du régulateur :

les paramètres choisis sont regroupés dans le tableau suivant :

	Contour ligne	Contour Cercle
λ_m	10	35
λ_{s1}	$\begin{bmatrix} 20\tilde{\alpha} & 0 \\ 0 & 20\tilde{\alpha} \end{bmatrix}$	$\begin{bmatrix} 20\tilde{\alpha} & 0 \\ 0 & 20\tilde{\alpha} \end{bmatrix}$
k_m	10	1000
K_s	$\begin{bmatrix} 1000\tilde{\alpha} & 0 \\ 0 & 1000\tilde{\alpha} \end{bmatrix}$	$\begin{bmatrix} 1000\tilde{\alpha} & 0 \\ 0 & 1000\tilde{\alpha} \end{bmatrix}$
ϕ_m	0.08	0.08
Φ_s	$\begin{bmatrix} 0.08 \\ 0.08 \end{bmatrix}$	$\begin{bmatrix} 0.08 \\ 0.08 \end{bmatrix}$

TABLE 4.7 – Paramètres du régulateur Mode glissant

Résultats de suivi du contour ligne

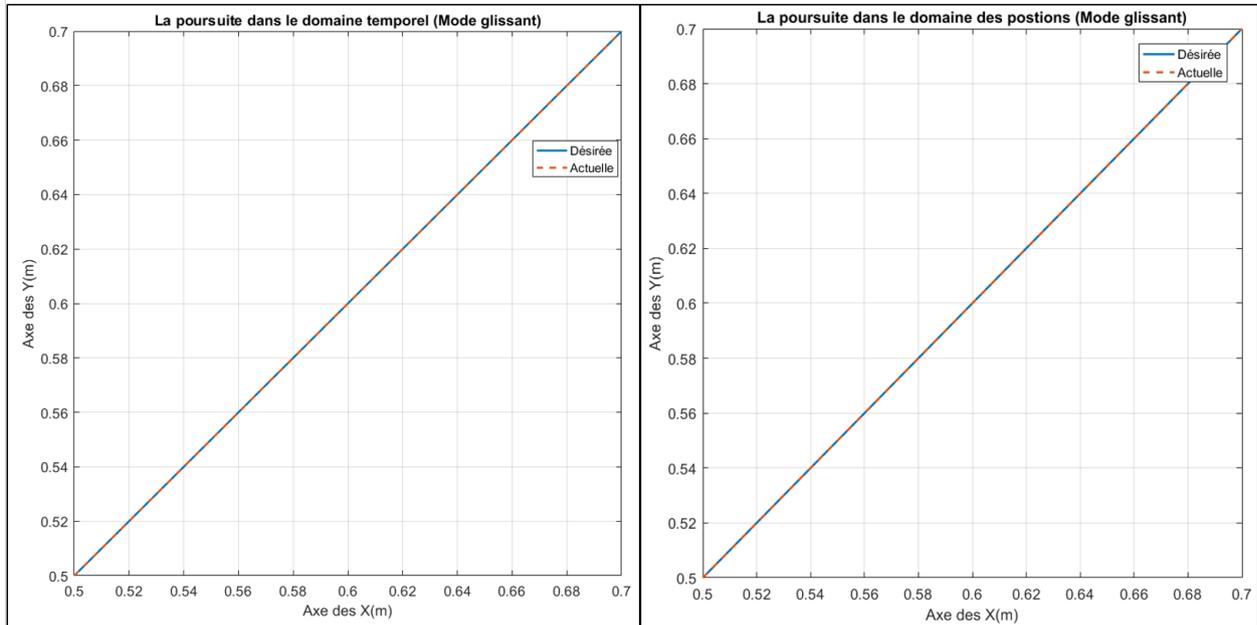


FIGURE 4.24 – Résultat de suivi du contour (contour Ligne)

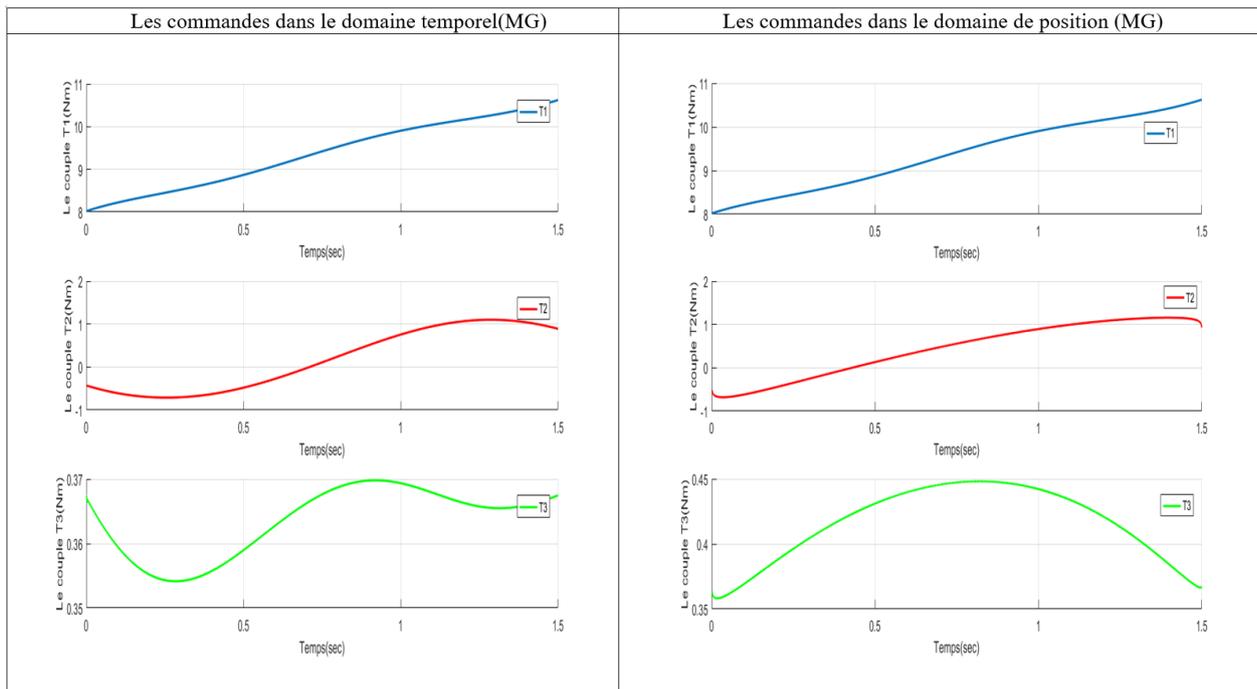


FIGURE 4.25 – Commandes par MG dans domaine temporel/position (Contour Ligne)

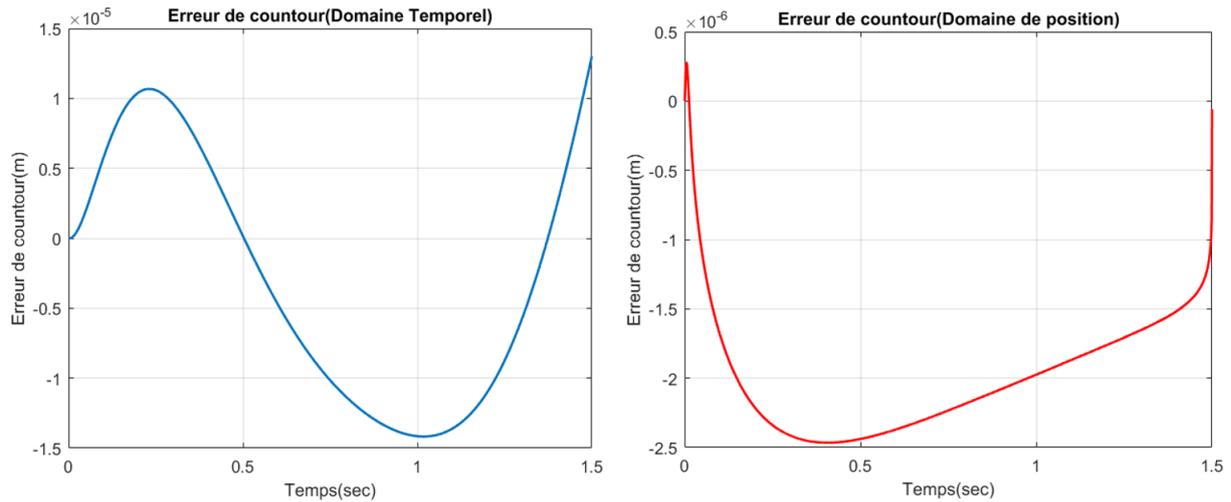


FIGURE 4.26 – L'erreur du Contour Ligne (Mode Glissant)

Commentaires Les commandes dans le domaine de position ont pu produire une erreur de contour légèrement petite par rapport aux commandes dans le domaine temporel(diminution de 60% (4.26)) ,tout en gardent pratiquement le même signal commande(4.25) ,ce résultat reste intéressant vu qu'on obtient les même performance avec le PD-MG dans le domaine temporel (figure (4.13) et (4.14)).

À travers la comparaison entre les commandes PD-MG et MG ,on remarque qu'avec le même signal de commande avec on a pu diminuer la moyenne de l'erreur de contour approximativement 10 fois , le résultat dans le domaine temporel (4.24) est clairement meilleur que celui du PD-MG (4.14) cela est grâce à la commande équivalente du mode glissant qui utilise le modèle du système cependant le choix des gains du PD-MG était hyper difficile .

Dans le domaine de position les deux commandes PD-MG et MG présentent pratiquement les mêmes performances concernant l'erreur contour.

Résultats de la simulation pour le contour circulaire

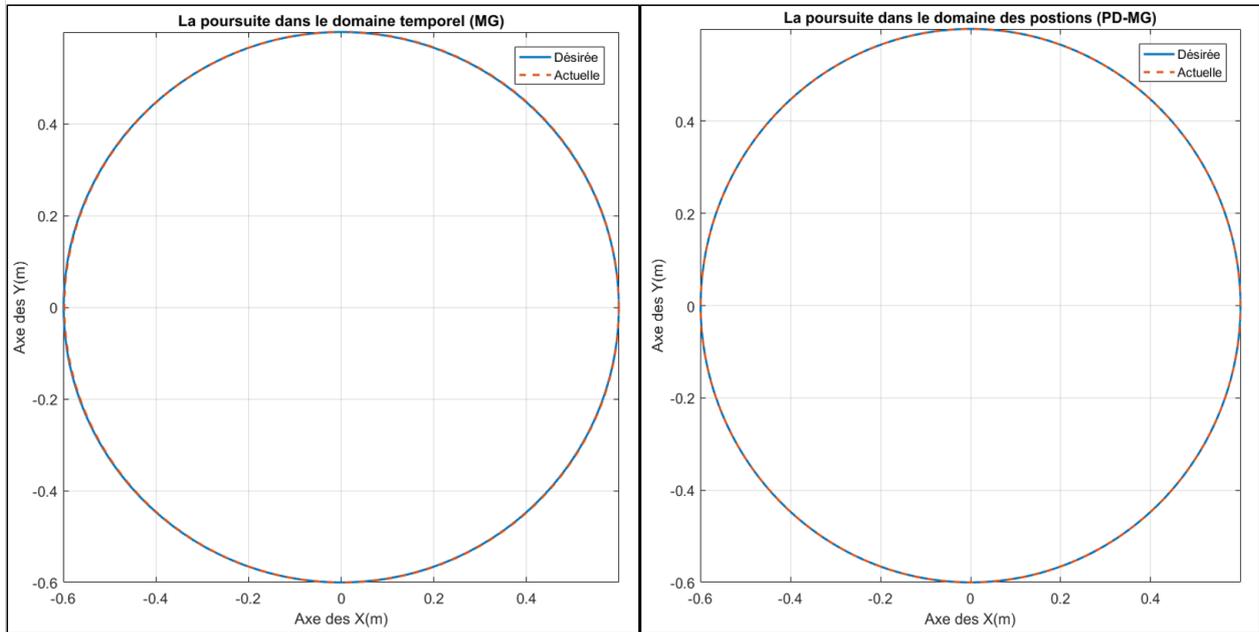


FIGURE 4.27 – Résultat de suivi du contour (contour cercle)

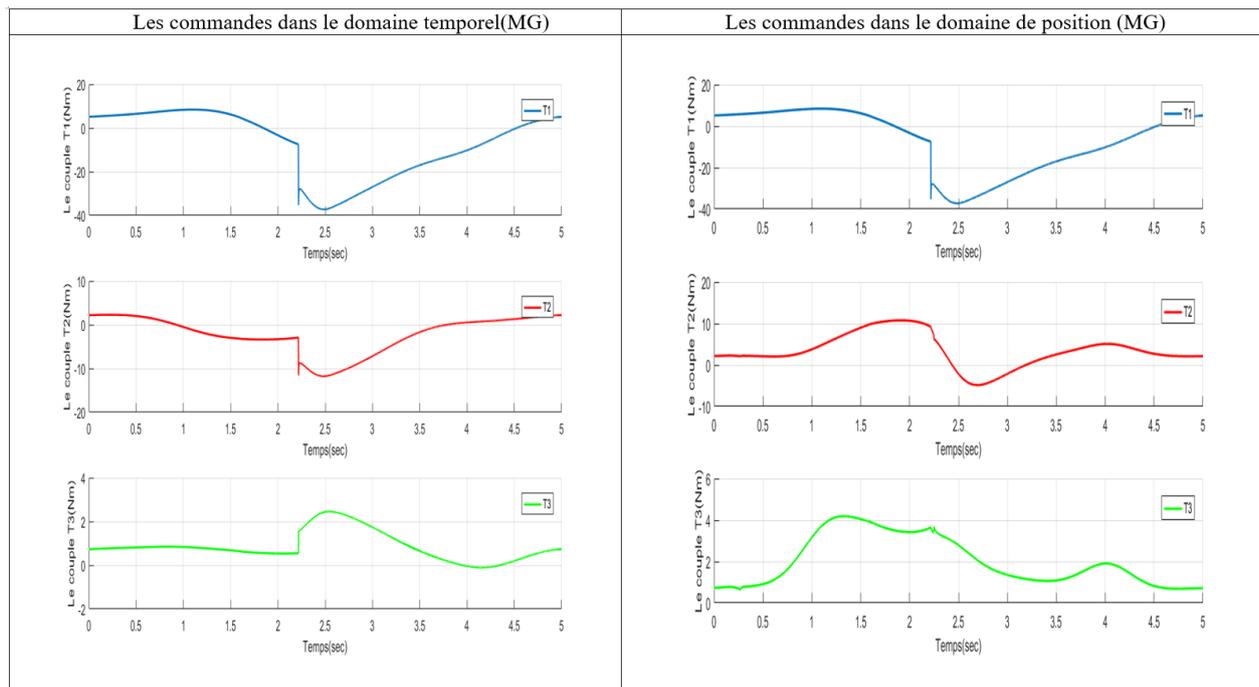


FIGURE 4.28 – Commandes MG dans domaine temporel/position (Contour cercle)

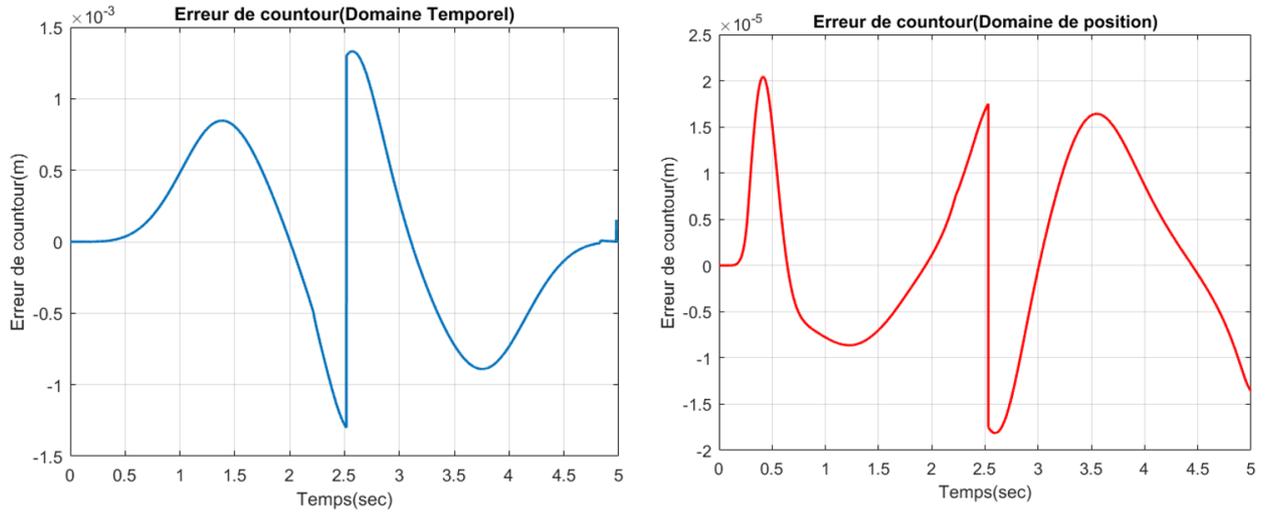


FIGURE 4.29 – L'erreur du Contour Cercle (MG)

Commentaires

On remarque l'amélioration de l'allure des couples τ_2 et τ_3 (figure(4.28)) dans le domaine de position (élimination des variations brusques des signaux de commande τ_2 et τ_3 qui pourraient nuire au système commandé dans le domaine temporel). Par ailleurs, l'erreur de contour dans le domaine de position est inférieure 90 fois (figure(4.29)) à celle du domaine temporel. En revanche, le niveau du signal de commande τ_3 augmente légèrement (figure(4.28)).

Commande par CAI-MG(Temps)

On reprend la commande trouvée dans le chapitre 3 :

$$\begin{cases} \tau_k(q_k, t) = u_{eq,k}(q_k, t) + v_k(t) \\ u_{eq,k}(q_k, t) = M(q_k)[\ddot{q}_d + \lambda e_{2k}] + C(q_k, \dot{q}_k)\dot{q}_k(t) + G(q_k) + F(t, q_k, \dot{q}_k) \\ v_k(t) = v_{k-1}(t) + \alpha_k(t)S(q_k) + \beta_k(t)\dot{S}(q_k) \\ S(q_k) = \dot{e}_{2k} + \lambda e_{1k} \end{cases} \quad (4.17)$$

Tel que :

$$\begin{cases} \|A\|_\infty = \|(I - \beta_k(t)M(q_k)^{-1})^{-1}\|_\infty \leq 1 \\ \|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}| \end{cases} \quad (4.18)$$

les paramètres du régulateur :

les paramètres choisis sont regroupées dans le tableau suivant :

	Contour ligne	Contour Cercle
λ	$\begin{bmatrix} 15 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 15 \end{bmatrix}$	$\begin{bmatrix} 35 & 0 & 0 \\ 0 & 35 & 0 \\ 0 & 0 & 35 \end{bmatrix}$
$\alpha_{\mathbf{k}}(\mathbf{t})$	$\begin{bmatrix} 0.75 & 0 & 0 \\ 0 & 0.75 & 0 \\ 0 & 0 & 1.65 \end{bmatrix}$	$\begin{bmatrix} 0.525 & 0 & 0 \\ 0 & 0.525 & 0 \\ 0 & 0 & 1.155 \end{bmatrix}$
$\beta_{\mathbf{k}}(\mathbf{t})$	$\begin{bmatrix} 10^{-6} & 0 & 0 \\ 0 & 10^{-6} & 0 \\ 0 & 0 & 10^{-6} \end{bmatrix}$	$\begin{bmatrix} 10^{-6} & 0 & 0 \\ 0 & 10^{-6} & 0 \\ 0 & 0 & 10^{-6} \end{bmatrix}$

TABLE 4.8 – Paramètres du CAI-MG

Résultat de la simulation pour le contour ligne

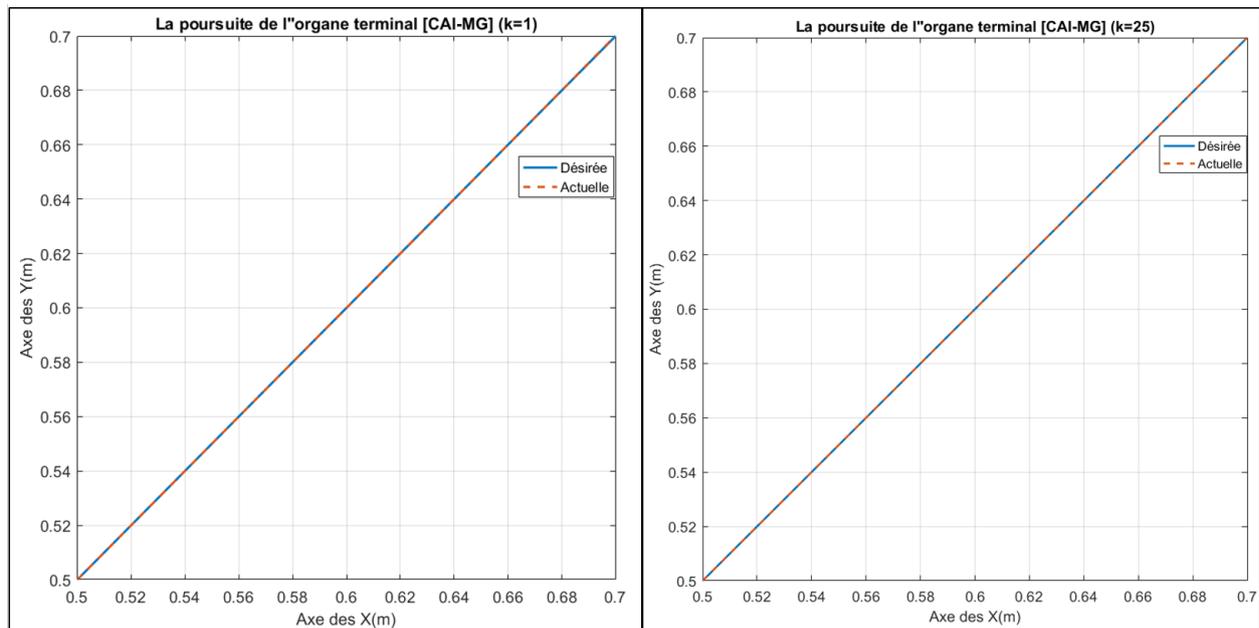


FIGURE 4.30 – Résultat de suivi du contour (contour Ligne)

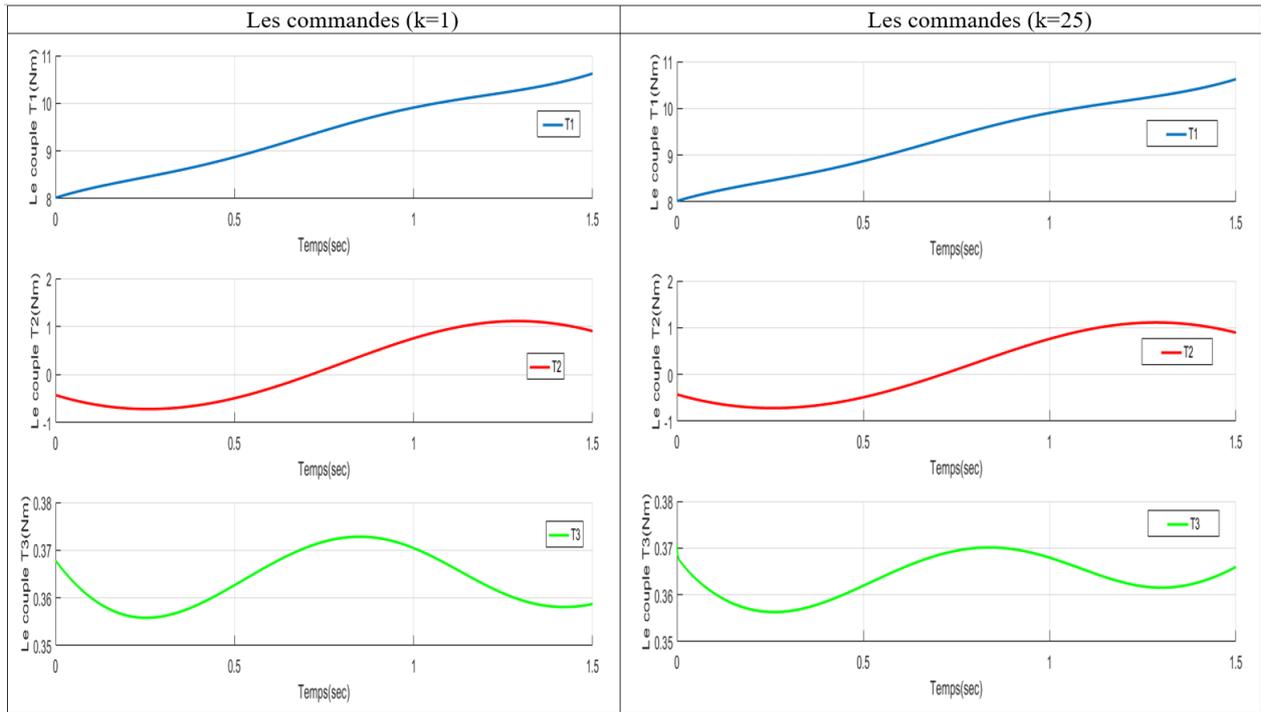


FIGURE 4.31 – Commandes par CAI-MG dans domaine(Contour Ligne)

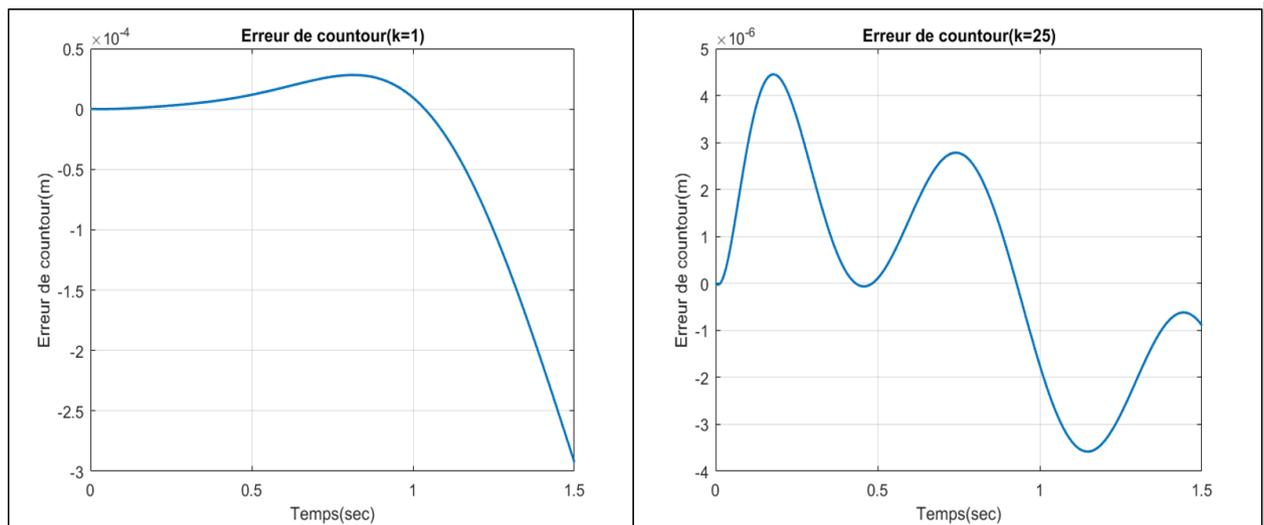


FIGURE 4.32 – L'erreur du Contour Ligne (CAI-MG)

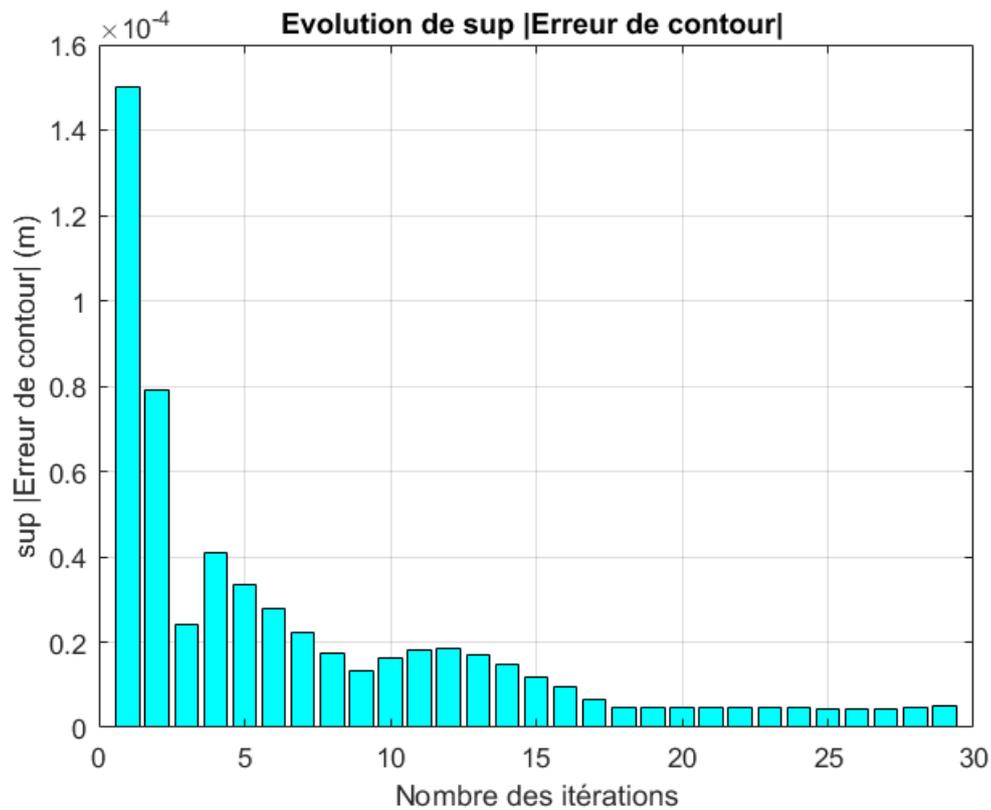


FIGURE 4.33 – Evolution de $\sup |E_c|$ au cours des itérations (contour ligne)

Commentaires

On remarque qu'après seulement 25 itérations on a pu améliorer la moyenne de l'erreur du contour approximativement 90 fois (figure(4.32) ,tout en gardant pratiquement la même énergie de commande(figure (4.31)).

On pu également confirmer l'étude de stabilité de Lyapounov établie lors du chapitre 3 (section 3.5.1) ,la figure (4.33) montre que $\sup(E_c)$ converge exponentiellement vers le zéro quand $k \rightarrow \infty$.

Cependant le choix des gains $\alpha_k(t)$ n'était pas évident pour assurer une stabilité globalement exponentielle.

Résultats de la simulation pour le contour circulaire

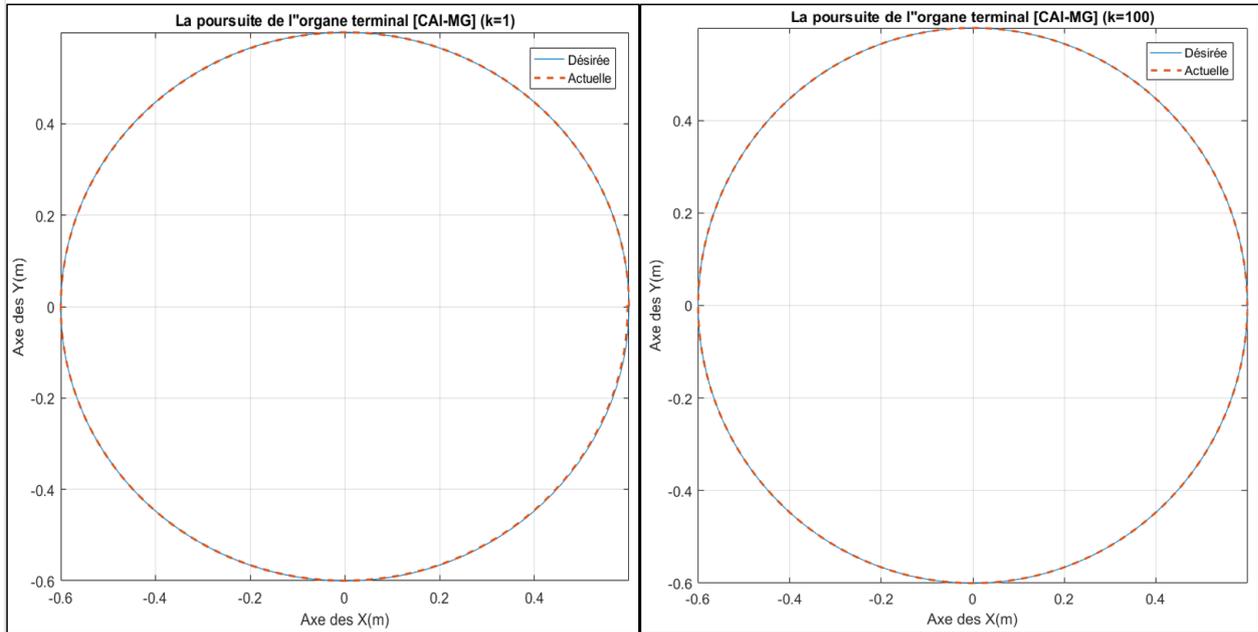


FIGURE 4.34 – Résultat de suivi du contour (contour Cercle)

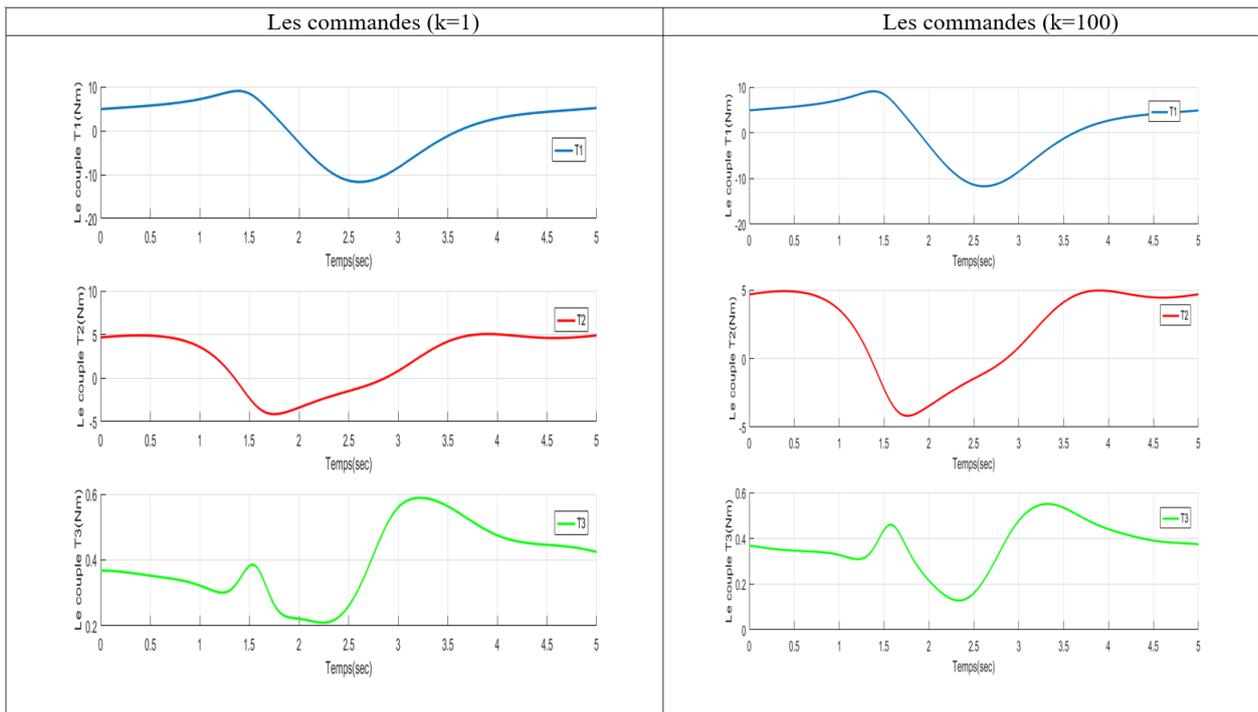


FIGURE 4.35 – Commandes par CAI-MG dans domaine(Contour Cercle)

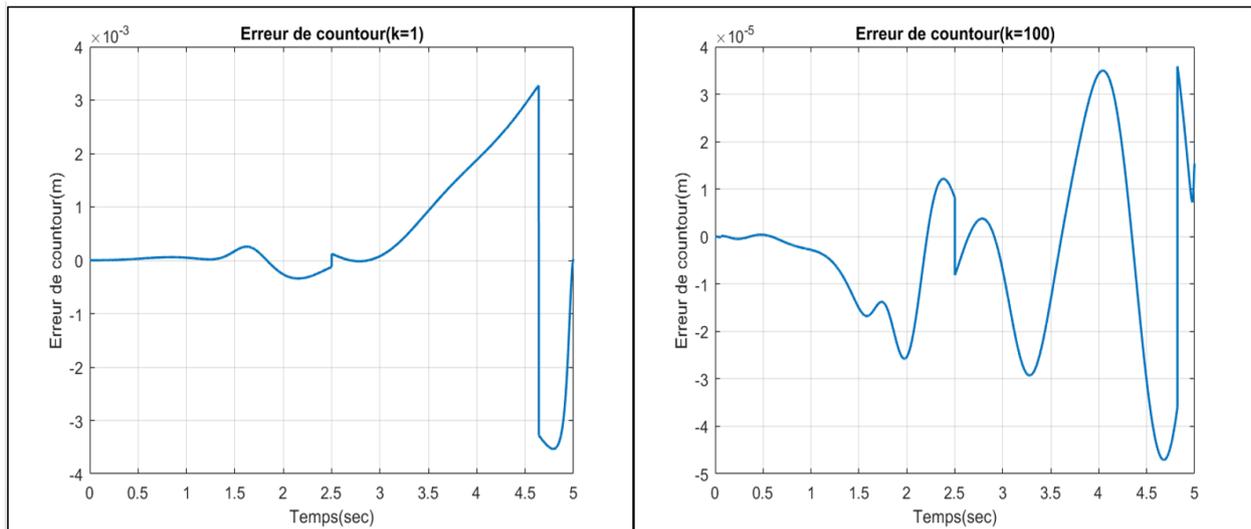
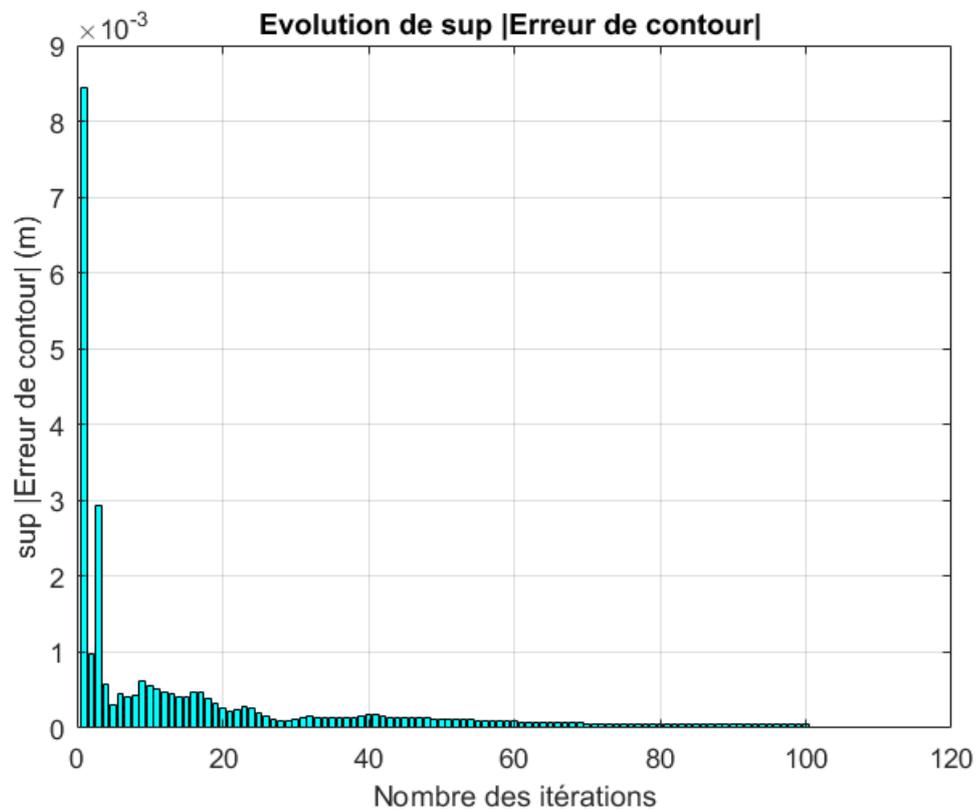


FIGURE 4.36 – L'erreur du Contour Cercle (CAI-MG)

FIGURE 4.37 – Evolution de $\sup |Ec|$ au cours des itérations (contour cercle)

Commentaires

D'une manière similaire au contour ligne ,on a pu on utilisant la commande CAI-MG diminuer la moyenne de l'erreur du contour approximativement plus de 100 fois (figure(4.36) ,tout en gardent pratiquement la même énergie de commande(figure (4.35)),avec une légère diminutions dans les commandes (τ_2 , τ_3 pour $k = 100$) .

La stabilité globale asymptotique (exponentielle) de Lyapounov démontré lors du chapitre 3 (section 3.5.1) peut se voir dans la figure (4.37) , le $\sup(E_c)$ converge exponentiellement vers le zéro quand $k \rightarrow \infty$. cependant le choix des gains $\alpha_k(t)$ n'est pas toujours évident pour assurer une stabilité globalement exponentielle .

4.4.2 Robot mobile

On reprend le modèle dynamique du système :

$$\begin{cases} \ddot{x} = \frac{\lambda}{m} \sin \phi + b_1 u_1 \cos \phi \\ \ddot{y} = -\frac{\lambda}{m} \cos \phi + b_1 u_1 \sin \phi \\ \ddot{\phi} = b_2 u_2 \end{cases} \quad (4.19)$$

La commande PD-MG

On reprend les résultats établies lors du chapitre 3 :

Commande dans le domaine temporel

$$\begin{cases} u_1(t) = K_{P_x} e_x(t) + K_{P_y} e_y(t) + K_{D_x} \dot{e}_x(t) + K_{D_y} \dot{e}_y(t) + K_s \text{sign}(s) \\ s = \dot{e}_x(t) + \dot{e}_y(t) + \lambda_x e_x(t) + \lambda_y e_y(t) \end{cases} \quad (4.20)$$

Commande dans le domaine de position

$$\begin{cases} u_m(t) = u_2(t) \\ u_s(q_m) = K_{P_x} e_x(q_m) + K_{P_y} e_y(q_m) + K_{D_x} \dot{e}_x(q_m) + K_{D_y} \dot{e}_y(q_m) + K_s \text{sat}(s_s, \phi_s) \\ s_s = \dot{e}_x(q_m) + \dot{e}_y(q_m) + \lambda_x e_x(q_m) + \lambda_y e_y(q_m) \end{cases} \quad (4.21)$$

les paramètres du régulateur :

les paramètres choisis sont regroupées dans le tableau suivant :

	Contour arc	Contour sinusoïdale
K_{Px}	69	69
K_{Dx}	0.69	0.69
K_{Py}	69	69
K_{Dy}	0.69	0.69
K_s	1	1
λ_x	3	3
λ_y	3	3
ϕ_s	1	1

TABLE 4.9 – Paramètres du régulateur PD-MG

Résultats de suivi du contour arc

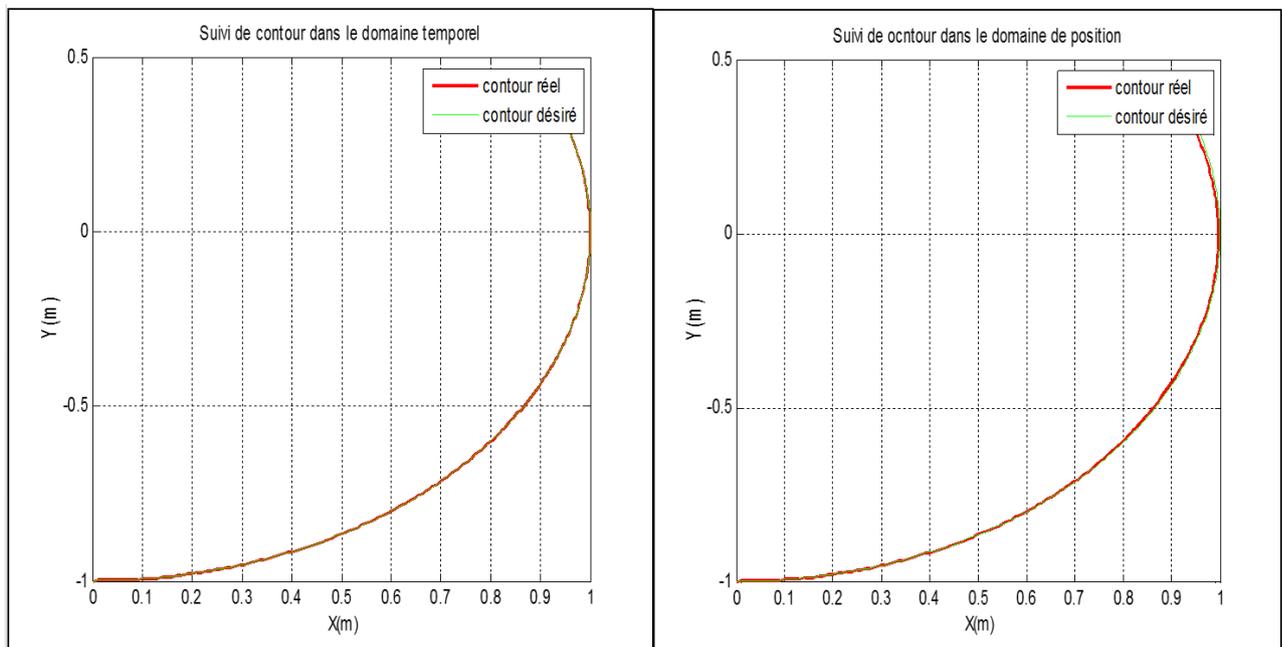


FIGURE 4.38 – Résultat de suivi du contour (contour arc)

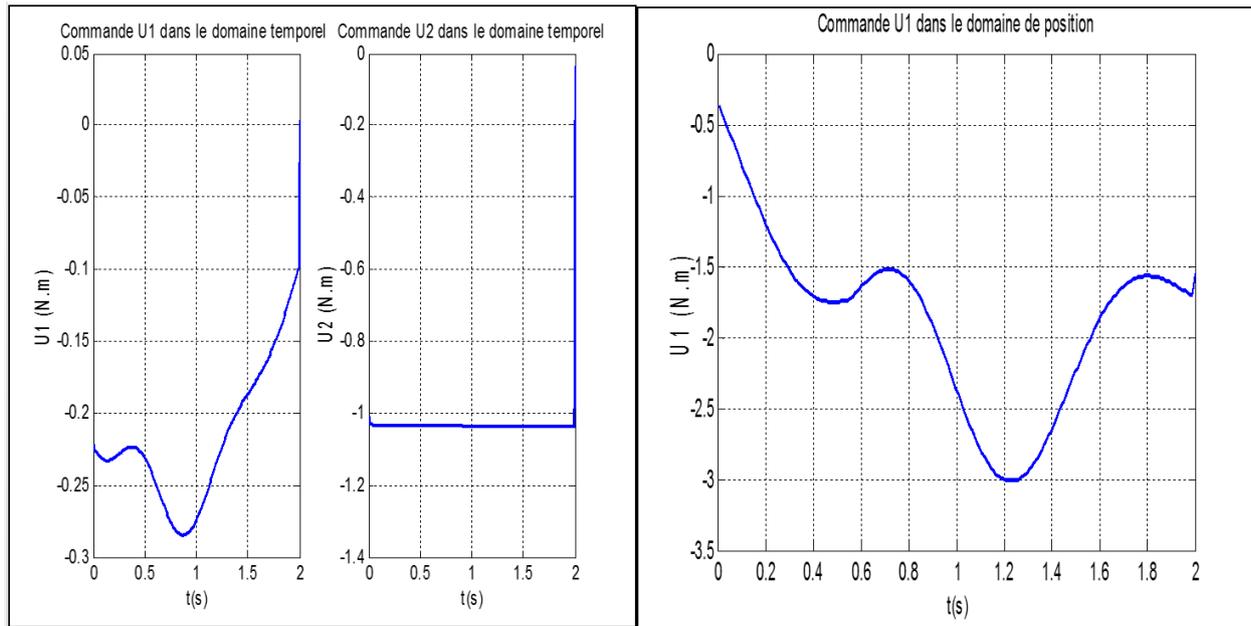


FIGURE 4.39 – Commandes dans domaine temporel/position (Contour arc)

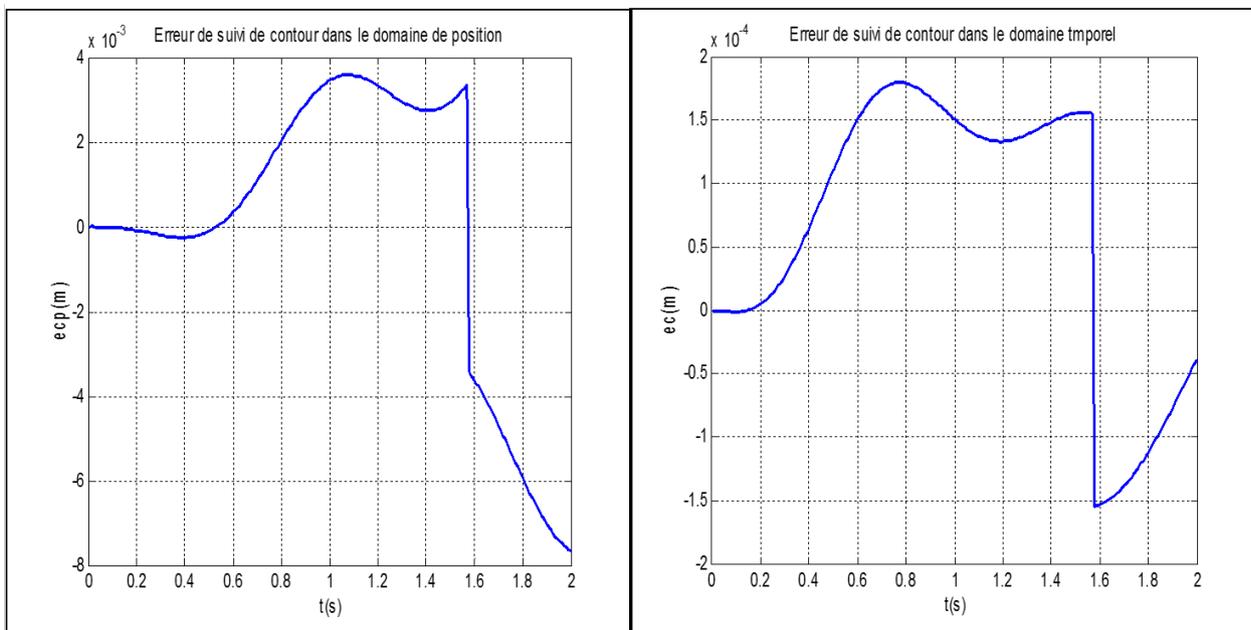


FIGURE 4.40 – L'erreur du Contour arc (PD-MG)

Commentaires

On constate une légère augmentation de l'erreur. En effet, le signal d'erreur garde pratiquement le même ordre de grandeur avec une légère diminution de sa valeur moyenne sur l'intervalle de simulation. Contrairement au cas du robot manipulateur, ici la variable maître peut être asservie à poursuivre une référence strictement monotone. Toutefois, non seulement la variable doit varier dans un intervalle défini afin que le signal de commande ne converge pas ; mais aussi la stricte monotonie de cette variable d'orientation ne permet pas de poursuivre un contour faisable tel qu'une trajectoire en 8.

A noter que le choix des gains de réglage demeure relativement délicat vu qu'il se fait empiriquement par la méthode d'essai et d'erreur.

Résultats de la simulation pour le contour sinus

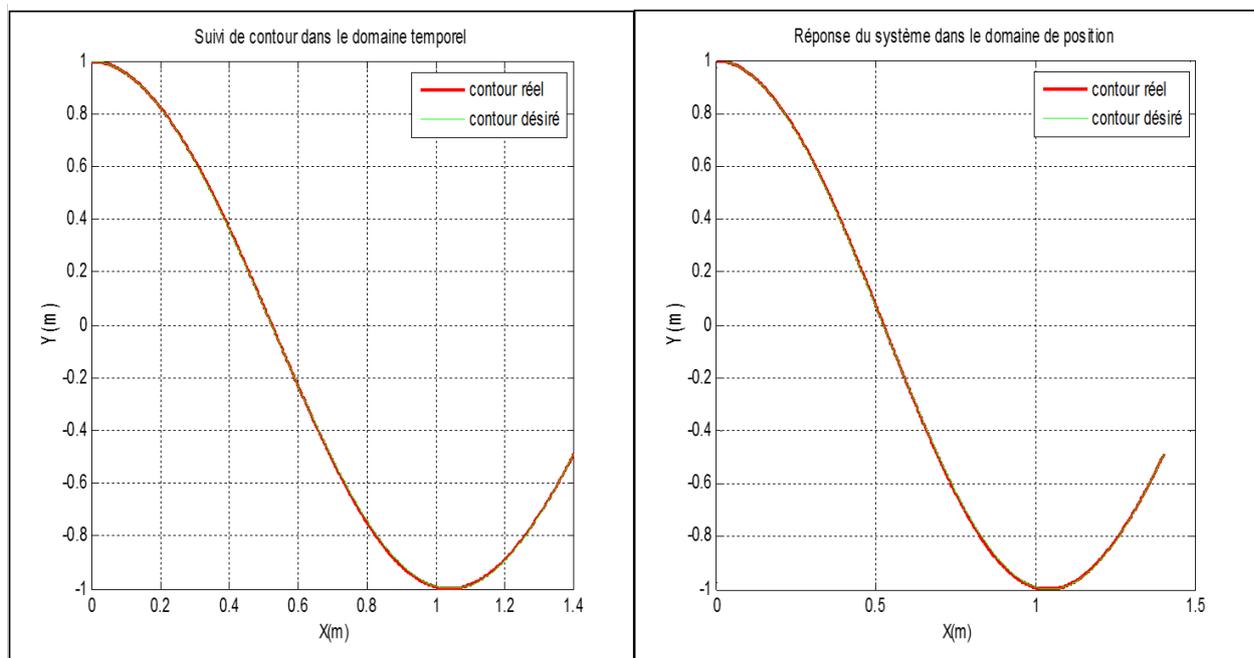


FIGURE 4.41 – Résultat de suivi du contour (contour sinus)

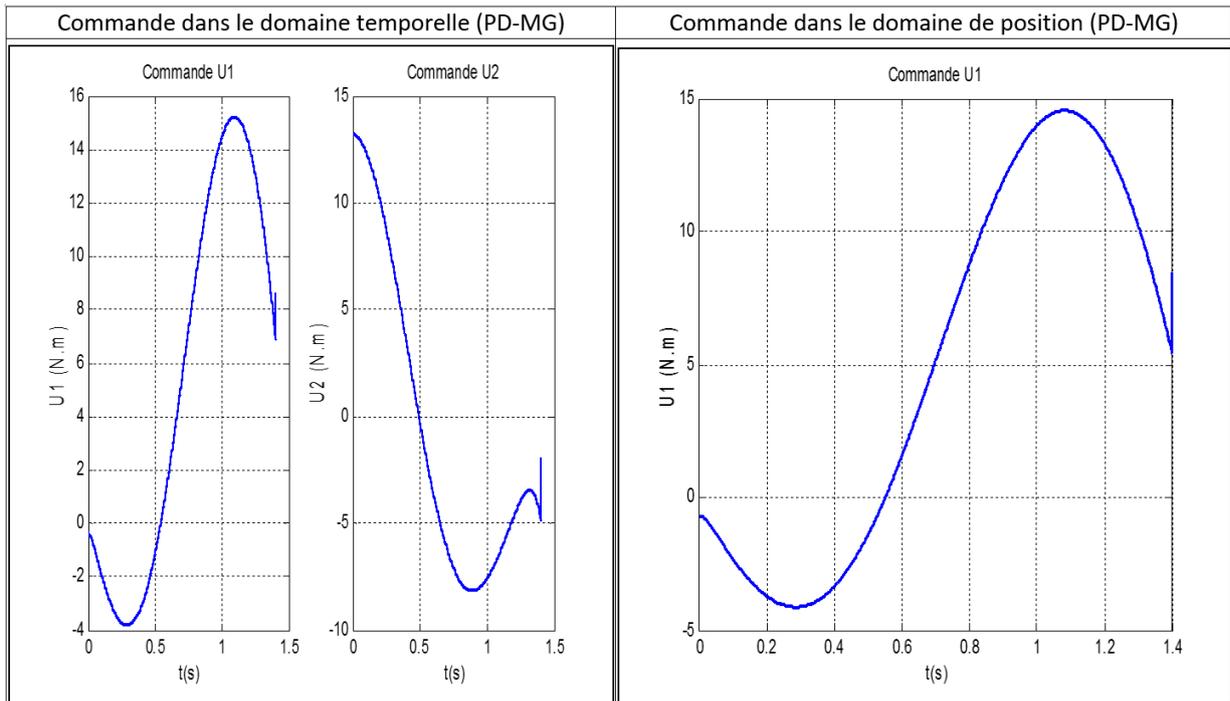


FIGURE 4.42 – Commandes PD-MG dans domaine temporel/position (Contour sinus)

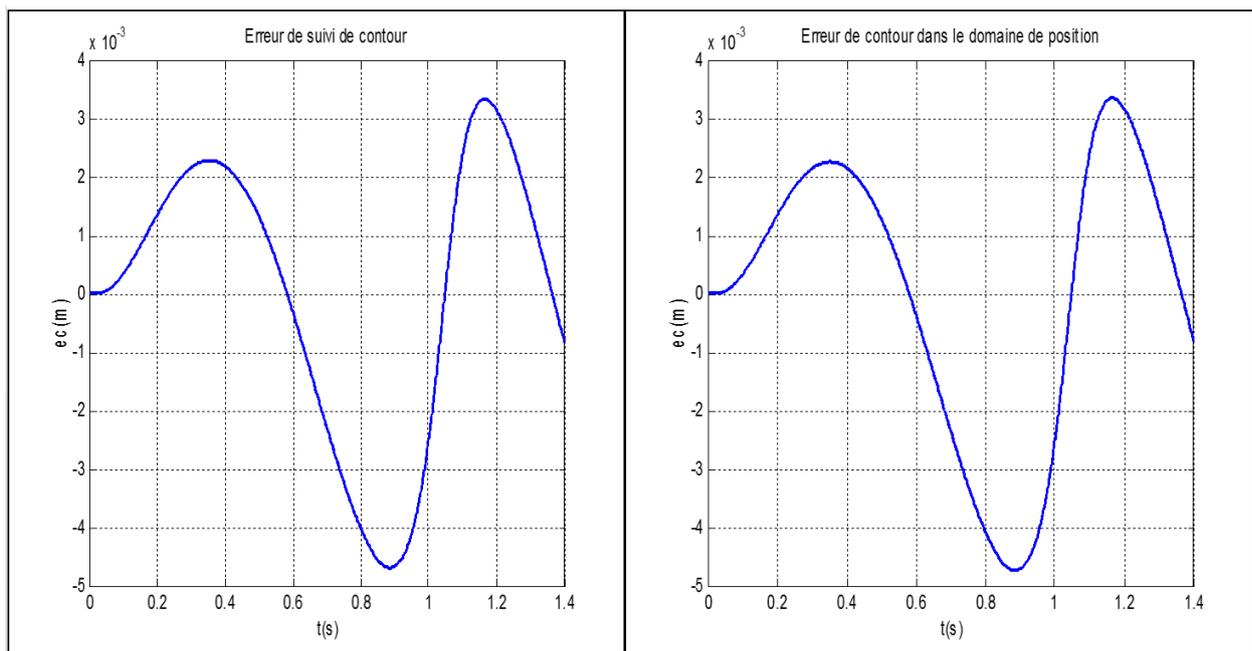


FIGURE 4.43 – L'erreur du Contour sinus (PD-MG)

Commentaires

On remarque que l'erreur moyenne diminue dans le domaine de position et les oscillations haute fréquence que subit le signal de commande $U1$ dans le domaine temporel sont éliminées dans le domaine de position.

La commande par mode glissant

On reprend les résultats établies lors du chapitre 3 :

Commande dans le domaine temporel

$$u_1(t) = \frac{K_s \text{sat}(s, \phi_s) + \ddot{x}_d + \ddot{y}_d + \lambda_x \dot{e}_x + \lambda_y \dot{e}_y - \frac{\lambda}{m}(\sin \phi - \cos \phi)}{b_1(\cos \phi + \sin \phi)} \quad (4.22)$$

Commande dans le domaine de position

$$u_1(q_m) = \dot{q}_m^2 \frac{(\tilde{\alpha} K_s \text{sat}(s, \phi_s) + \tilde{\alpha} \lambda_x \dot{e}'_x(q_m) + \tilde{\alpha} \lambda_y \dot{e}'_y(q_m) + \ddot{x}'_d(q_m) + \ddot{y}'_d(q_m))}{b_1(\cos q_m + \sin q_m)} - \frac{\frac{1}{\dot{q}_m^2} \left(\frac{\lambda(q_m)}{m} (\sin q_m - \cos q_m) - \ddot{q}_m (x' + y') \right)}{b_1(\cos q_m + \sin q_m)} \quad (4.23)$$

les paramètres du régulateur

les paramètres choisis sont regroupés dans le tableau suivant :

	Contour ligne	Contour Cercle
λ_x	39	39.6
λ_y	39.6	19.6
\mathbf{K}_s	3	39
ϕ_s	0.3	0.3

TABLE 4.10 – Paramètres du régulateur Mode glissant

Résultats de suivi du contour arc

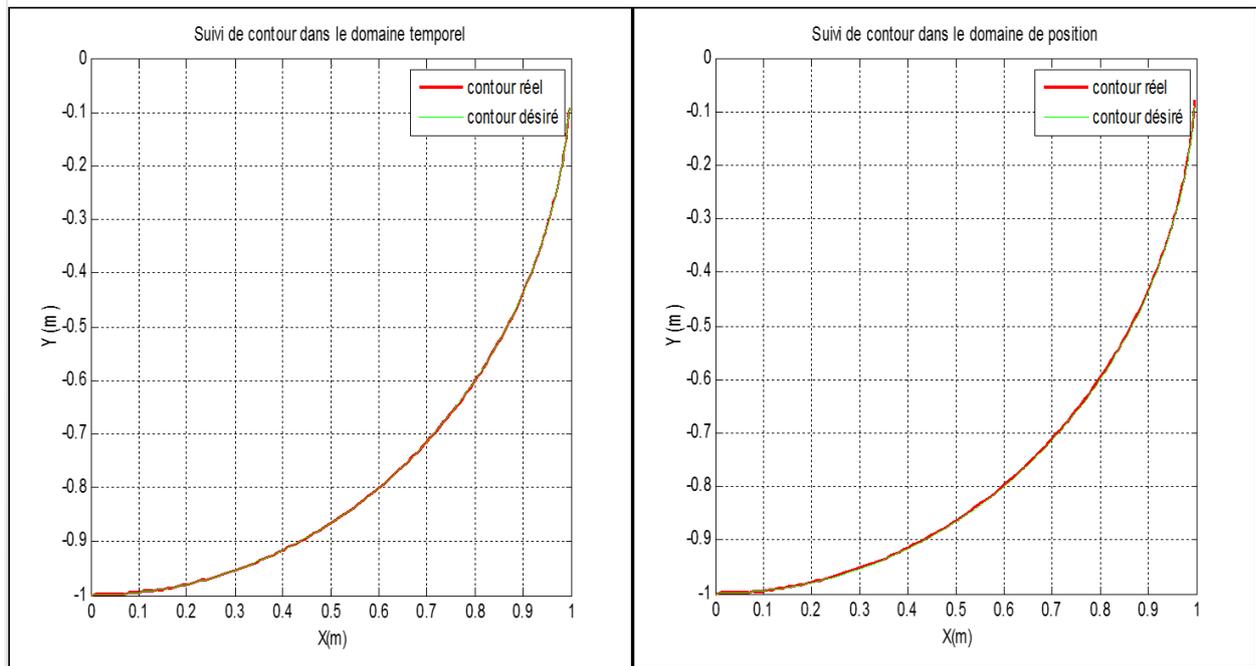


FIGURE 4.44 – Résultat de suivi du contour (contour arc)

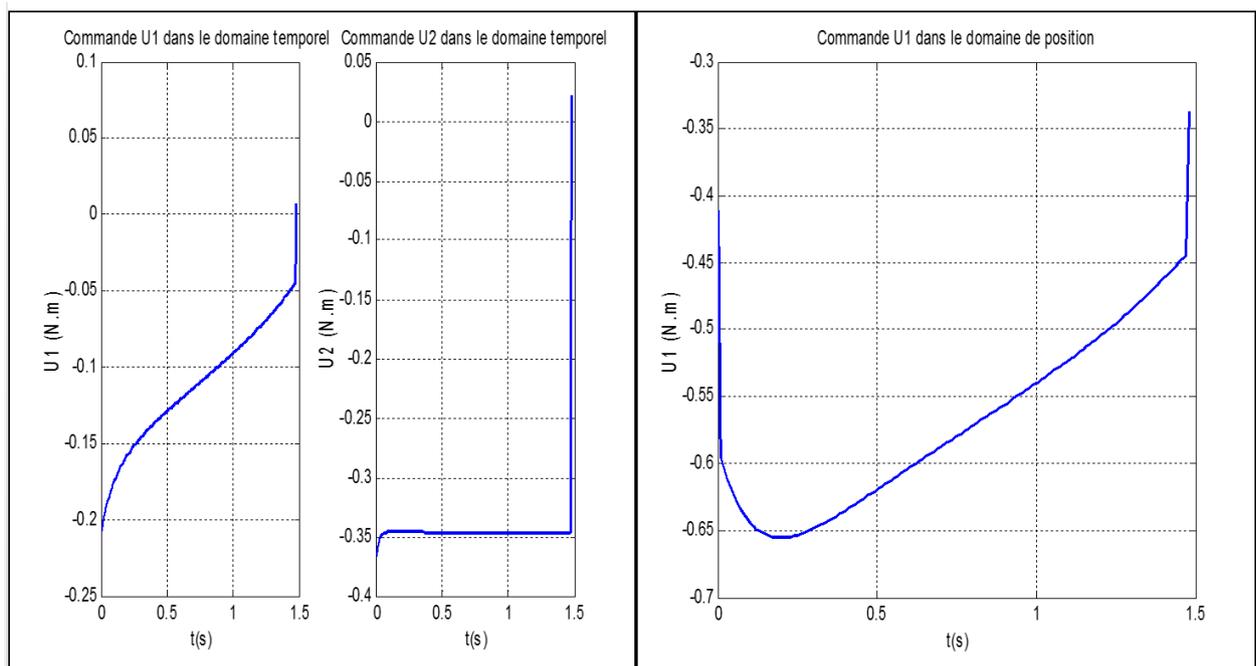


FIGURE 4.45 – Commandes par MG dans domaine temporel/position (Contour arc)

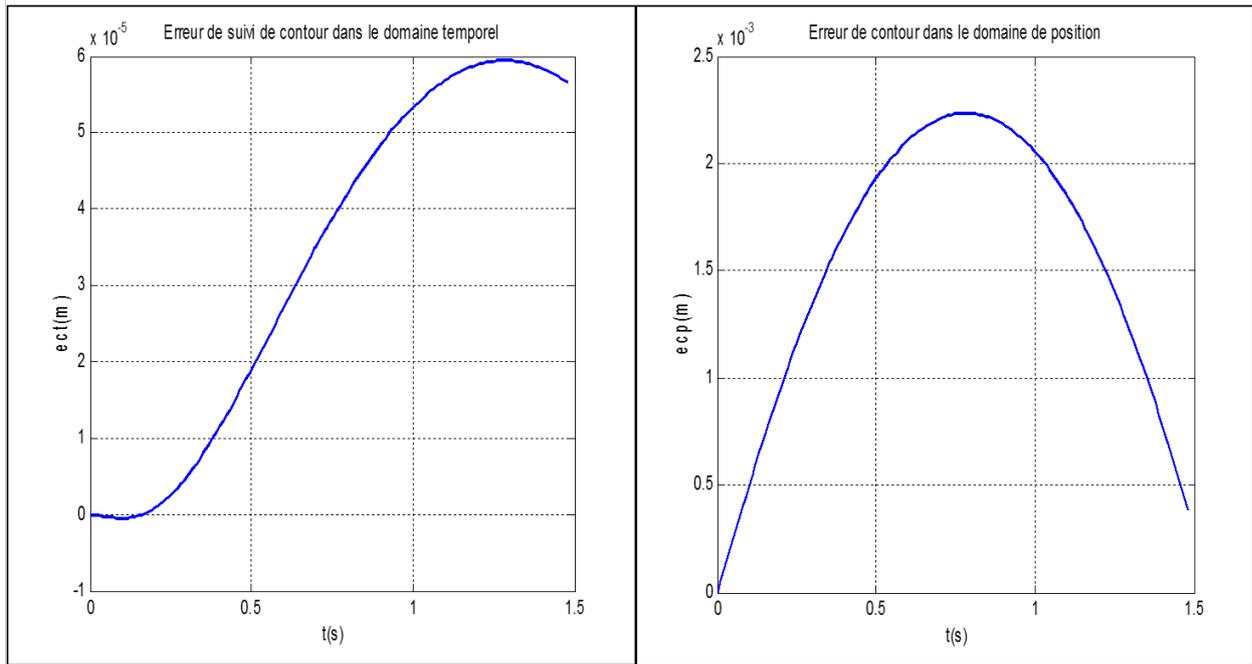


FIGURE 4.46 – L'erreur du Contour arc (Mode Glissant)

Commentaires

Dans le cas d'un contour circulaire, le signal de commande augmente considérablement. Ceci n'affecte pourtant pas le signal d'erreur qui conserve relativement sa même valeur moyenne dans le domaine de position.

Résultats de la simulation pour le contour sinus

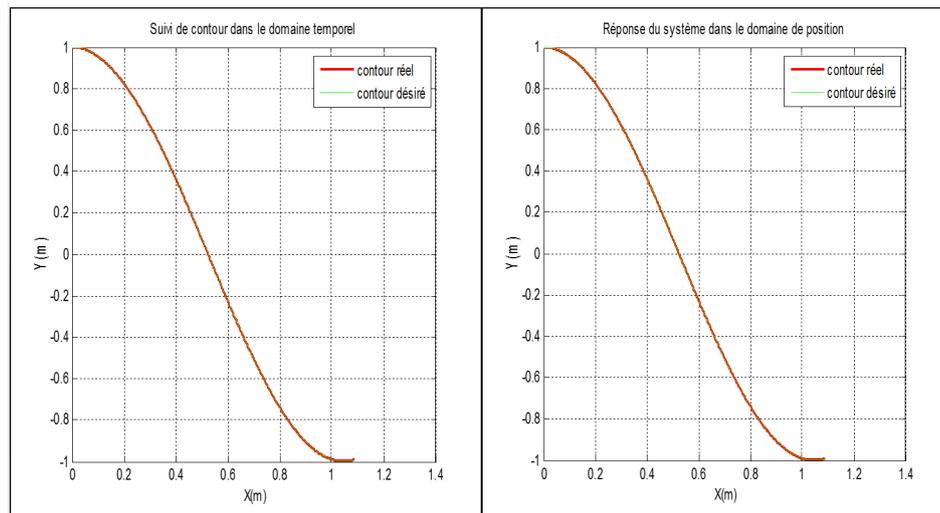


FIGURE 4.47 – Résultat de suivi du contour (contour sinus)

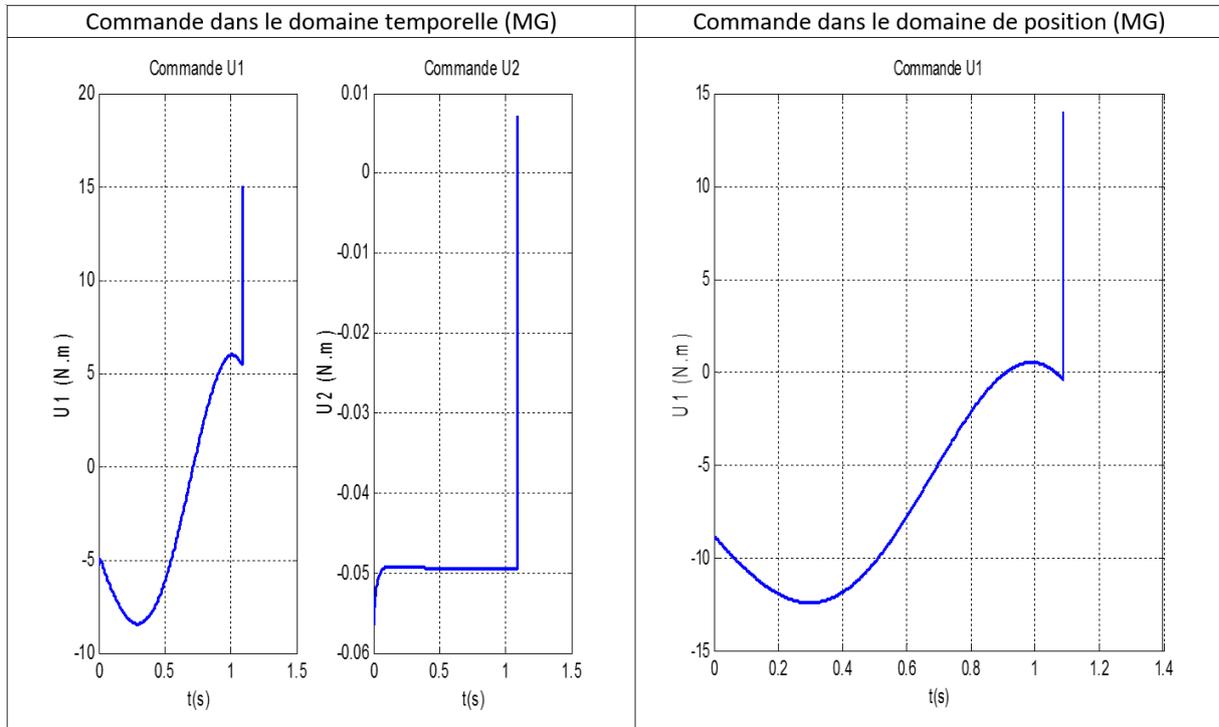


FIGURE 4.48 – Commandes MG dans domaine temporel/position (Contour sinus)

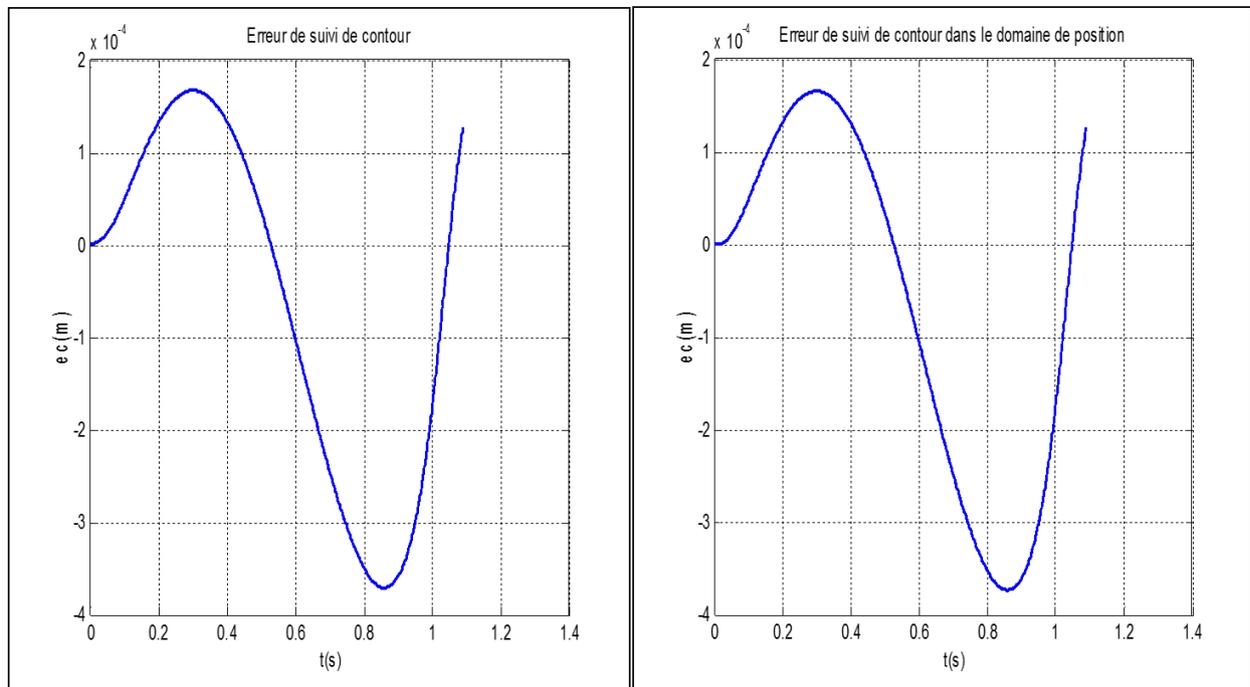


FIGURE 4.49 – L'erreur du Contour sinus (MG)

Commentaires Ici, également les performances du système commandé sont pratiquement inchangées lors du passage vers le domaine de position.

Commande par CAI-MG(Temps)

On reprend la commande trouvée dans le chapitre 3 :

$$\begin{cases} u_{1k}(t) = u_{1eqk}(t) + v_k(t) \\ u_{1eqk}(t) = \frac{(\ddot{x}_d + \ddot{y}_d + \lambda_x \dot{e}_{x_k} + \lambda_y \dot{e}_{y_k} - \frac{\lambda}{m}(\sin \phi_k - \cos \phi_k))}{b_1(\cos \phi_k + \sin \phi_k)} \\ v_k(t) = v_{k-1}(t) + \alpha_k(t)s_k(q_k) + \beta_k(t)\dot{s}_k(q_k) \\ s_k = \lambda_x e_{x_k} + \lambda_y e_{y_k} + \dot{e}_{x_k} + \dot{e}_{y_k} \end{cases} \quad (4.24)$$

les paramètres du régulateur :

les paramètres choisis sont regroupés dans le tableau suivant :

	Contour arc	Contour sin
λ_x	29.9	-
λ_y	29.9	-
$\alpha_k(\mathbf{t})$	1	-
$\beta_k(\mathbf{t})$	10^{-9}	-

TABLE 4.11 – Paramètres du CAI-MG

Résultat de la simulation pour le contour arc

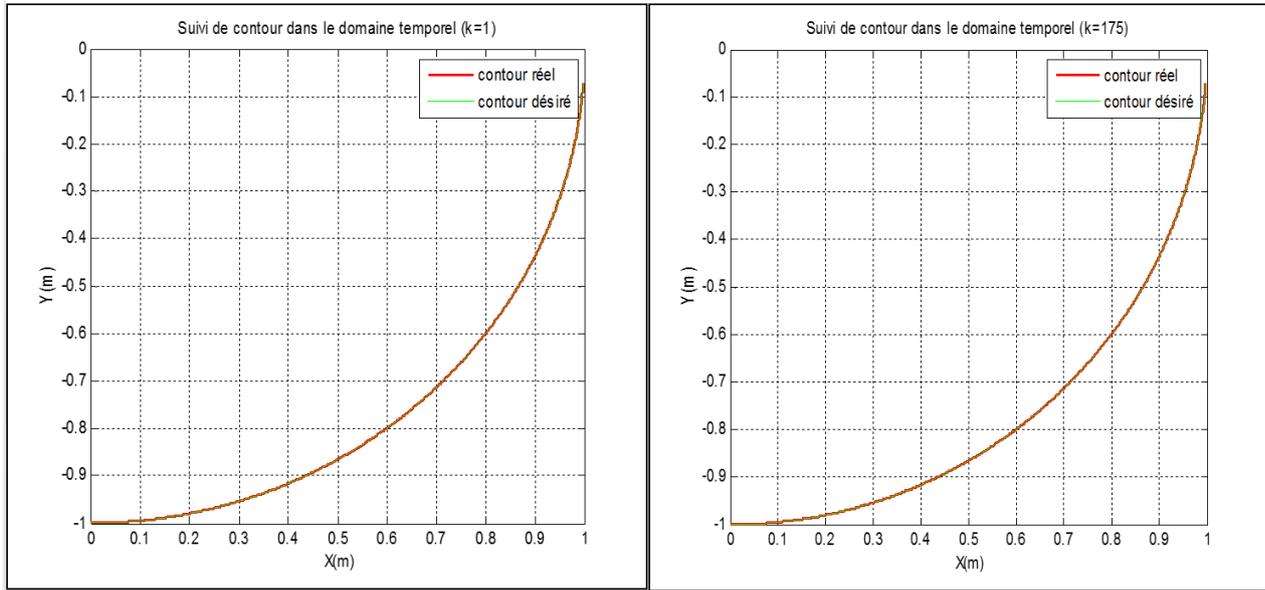


FIGURE 4.50 – Résultat de suivi du contour (contour arc)

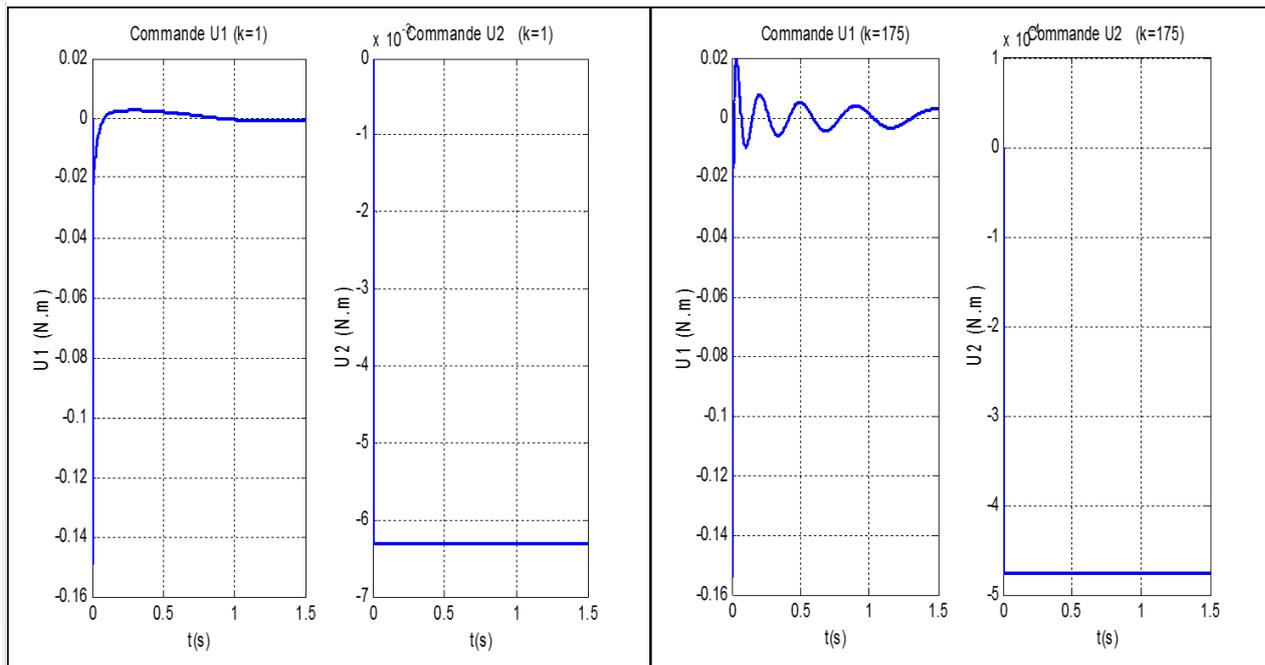


FIGURE 4.51 – Commandes par CAI-MG dans domaine (Contour arc)

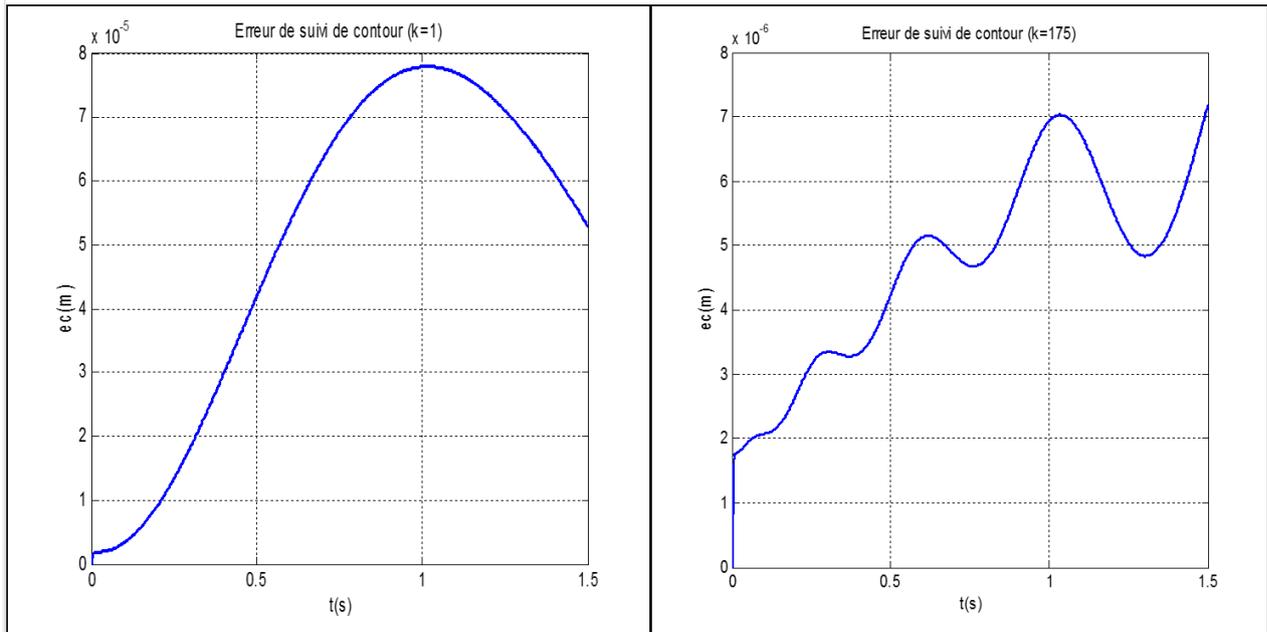
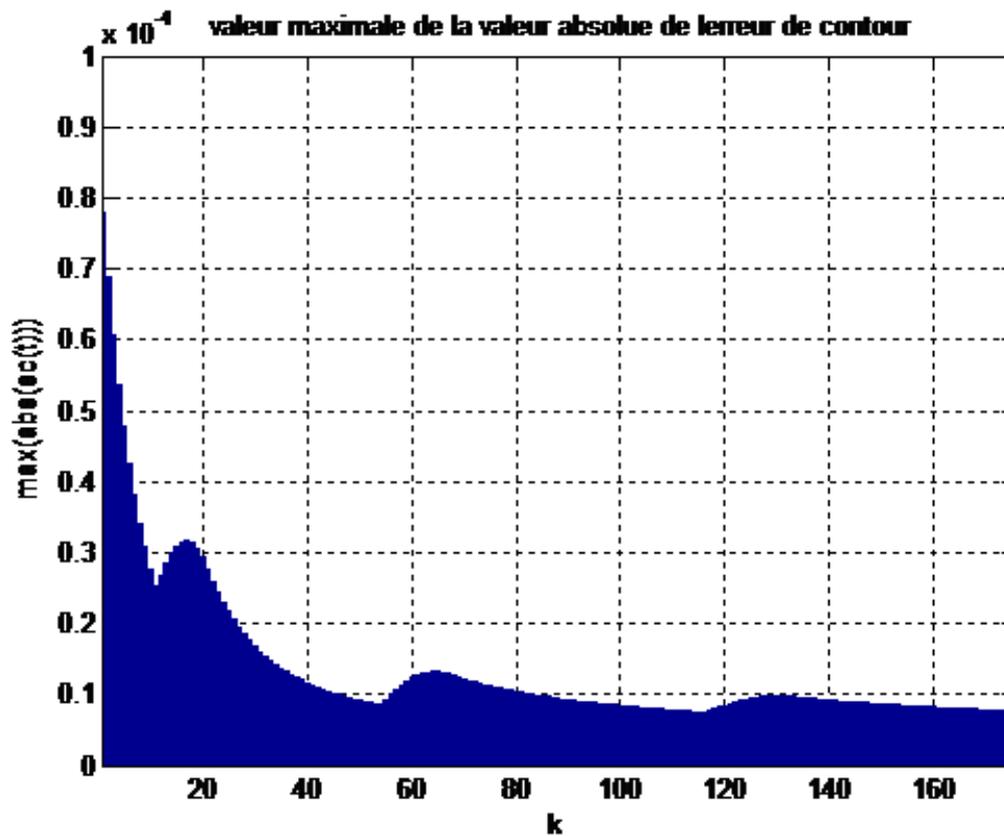


FIGURE 4.52 – L'erreur du Contour arc (CAI-MG)

FIGURE 4.53 – Evolution de $\sup |Ec|$ au cours des itérations (contour ligne)

Commentaires On remarque que le max absolue l'erreur de contour diminue d'une manière exponentielle ,la moyenne de l'erreur subit une diminution de 10 fois en 175 itérations ce qui prouve la la convergence de la CAI-MG étudiée lors du chapitre 3 , le signal commande reste toujours admissible malgré il devient très peu oscillatoire .

4.5 Synthèse comparative

Robot manipulateur

A fin d'évaluer les performances des lois de commandes élaborées dans le chapitre 3 et simulées dans ce chapitre , cette étude est divisée en deux partie : une étude quantitative où on procède à un test benchmark en utilisant des critères de performances usuelles , et par la suite on va étudier la robustesse du système. Une étude qualitative où on va résumer l'étude quantitative en un tableau d'appréciations qui va comparer les performances des lois de commandes traitées .

4.5.1 Étude quantitative

Pour cette synthèse comparative nous avons choisi 5 critères d'optimisation et de performances notamment :

L'intégrale de la valeur absolue de l'erreur (IAE)

Ce critère est utilisé pour mesurer l'erreur cumulé lors du passage de l'organe terminal vers le contour désiré

$$J_{IAE} = \int_0^{\infty} |e(t)|dt \quad (4.25)$$

L'intégrale du temps la valeur absolue de l'erreur (ITAE)

Ce critère est utilisé pour mesurer la rapidité de la réponse du système . il est définit comme suit :

$$J_{ITAE} = \int_0^{\infty} t|e(t)|dt \quad (4.26)$$

L'intégrale du carrée de la commande(ISV)

Ce critère est utiliser pour mesurer l'énergie de commande :

$$J_{ISV} = \sum_{i=1}^{i=m} \int_0^{\infty} u_i^2(t)dt \quad (4.27)$$

L'intégrale du carrée de la commande(CISV)

Ce critère est utilisé pour mesurer l'énergie consommée par les actionneurs dans le cas d'une commande répétitive (k itération) (commande dans le domaine de position / commande itérative) :

$$J_{CISV} = \sum_{i=1}^{i=k} \sum_{j=1}^{j=m} \int_0^{\infty} u_{kj}^2(t) dt \quad (4.28)$$

Temps de simulation (Ts)

Pour ce test nous allons utiliser les fonctions matlab "tic" et "toc" , il est important de dire que les simulations sont faites avec MATLAB R2016B , Processeur I7 RAM 16Gb , Disque dur SSD.

Performances dans le cas du contour Ligne

Commande	IAE	ITAE	IVS	CIVS	Ts
PD-MG (Temps)	0.0012	0011	129.204	- -	47.69
PD-MG (Position)	1.4614e-05	1.1521e-05	128.0999	257.3039	174.39
FD (Temps)	1.9761e-05	1.9531e-05	133.4201	- -	3.08
FD (Position)	8.8854e-06	8.3718e-06	133.1991	266.6192	24.30
MG (Temps)	1.2236e-05	9.9216e-06	133.4089	- -	1.64
MG (Position)	2.9723e-06	2.1504e-06	133.5293	266.9382	76.2
CAI-MG(k=20)	3.1106e-06	2.4997e-06	133.4216	2.8018e+03	11.46
CAI-MG(k=25)	2.4713e-06	1.6496e-06	133.4185	3.4689e+03	13.78

TABLE 4.12 – Les performances de suivi pour le contour ligne

— excellent résultat

— bon résultat

— mauvais résultat

Performances dans le cas du contour cercle

Commande	IAE	ITAE	IVS	CIVS	Ts
PD-MG (Temps)	0.0047	0.0164	334.7821	- -	280.78
PD-MG (Position)	5.1094e-04	0.0013	1145.1968	1.4800e+03	610.87
FD (Temps)	1.9761e-05	1.9531e-05	133.4201	- -	3.08
FD (Position)	- -	- -	- -	- -	- -
MG (Temps)	0.0023	0.0061	1.4135e+03	- -	1.64
MG (Position)	4.0084e-05	1.0370e-04	1.4655e+03	2.8790e+03	323.43
CAI-MG(k=100)	6.7844e-05	2.3211e-04	274.7526	2.8572e+04	176.13
CAI-MG(k=200)	3.5575e-05	1.0827e-04	274.7237	5.5221e+04	324.129

TABLE 4.13 – Les performances de suivi pour le contour cercle

— excellent résultat — bon résultat — mauvais résultat

Commentaires

On remarque que la commande PD-MG en temps et en position est la commande la moins performante car l'erreur cumulée(CIVS) est relativement importante .La réponse est relativement lente(ITAE) ceci est due à la difficulté du choix des gains vu que la commande est indépendante du modèle mais ceci facilite la synthèse et l'implémentation . Les commandes MG(Position) et CAI-MG(k=25) donne les meilleures performances , le choix entre les deux commandes dépendent de la tâche à effectuer , si la tâche est répétitive (plus de 25/200 itérations(countours ligne/cercle)) le choix optimale est la CAI-MG(k=25) vu qu'elle donne une erreur de contour relativement petite une réponse rapide et enfin une énergie de commande minimale cependant l'énergie cumulée(CISV) est importante et ceci va surcharger les actionneurs .

la MG(Position) donne un résultat pratiquement similaire avec une énergie cumulé relativement petite (seulement 2 itérations) , l'énergie de la commande est importante (cas du contour cercle) vue la difficulté du choix des gains pour assurer la stricte monotonie de la trajectoire du maître.

4.5.2 Test de Robustesse

Rejet perturbation et erreur de modélisation

Pour cette étude nous allons prendre le contour ligne seulement , et les commandes (Forwarding) (MG) et (CAI-MG)

$$M(q)\ddot{q}(t) + C(q, \dot{q})\dot{q}(t) + G(q) = \tau(t) + d(t) \quad (4.29)$$

avec $d(t)$ est e vecteur contenant les dynamiques non modélisées et les perturbations externes inconnues [47], tel que :

$$d(t) = \begin{cases} 0 & t < 0.5s \\ \begin{bmatrix} rand(a)\sin(2\pi t) \\ rand(b)\sin(2\pi t) \\ rand(c)\sin(2\pi t) \end{bmatrix} & t > 0.5s \end{cases} \quad (4.30)$$

4.5.3 Commande par Forwarding

Comme la commande par forwarding n'est pas généralement robuste on prend $a = 0.05$ $b = 0.01$ et $c = 0.001$,avec le même régulateur utilisé dans la partie simulation .

Résultats de suivi du contour ligne

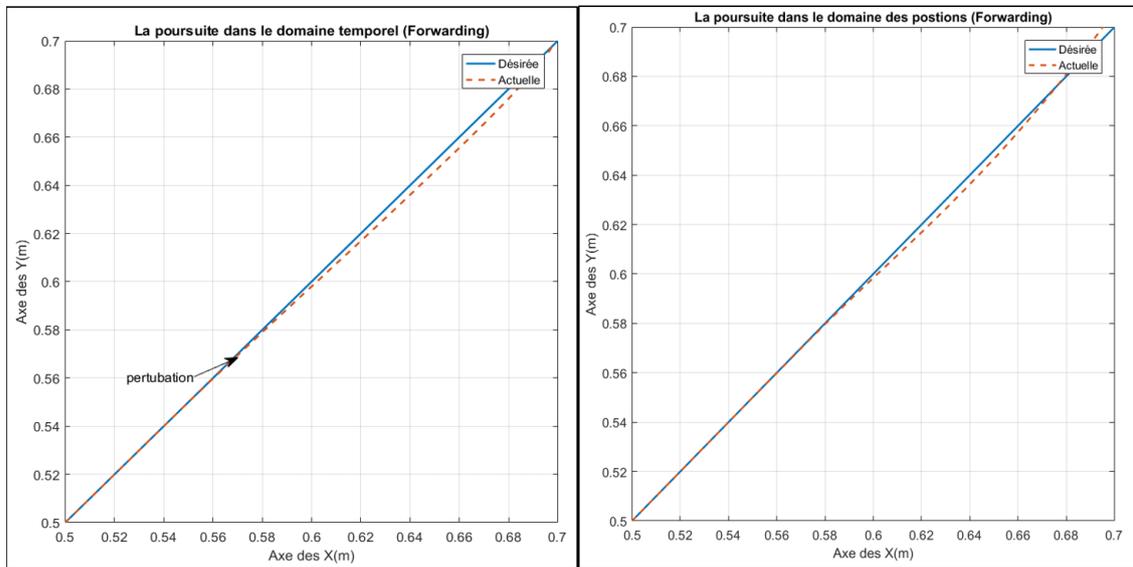


FIGURE 4.54 – Résultat de suivi du contour (contour Ligne)

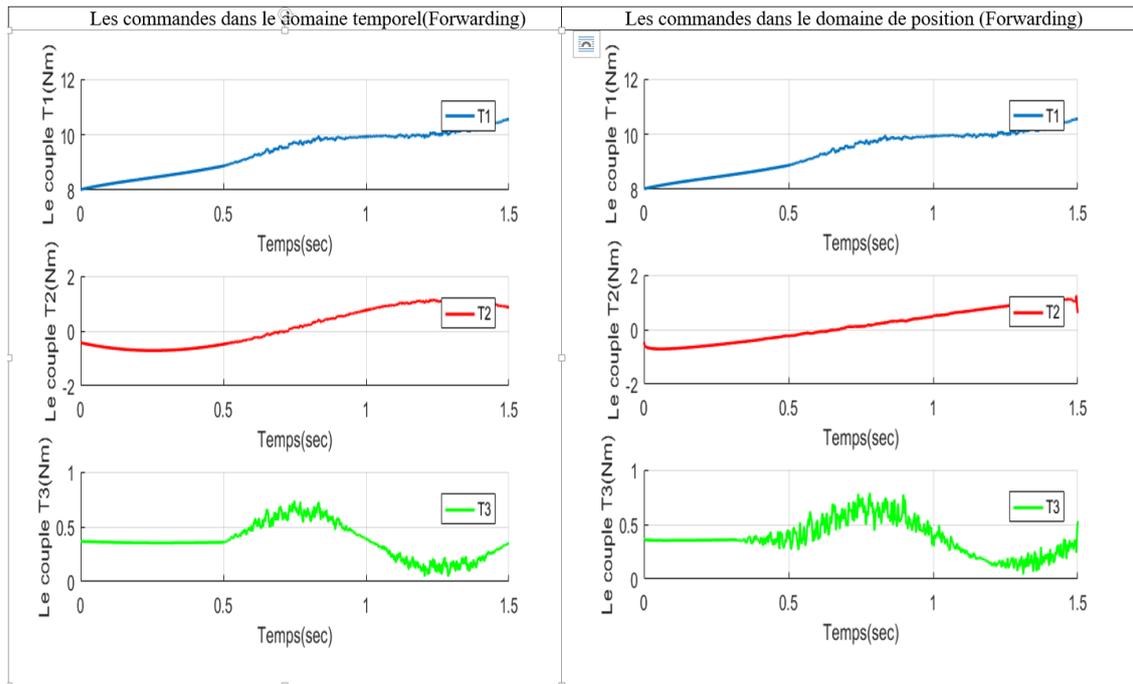


FIGURE 4.55 – Commandes Forwarding dans domaine temporel/position (Contour Ligne)

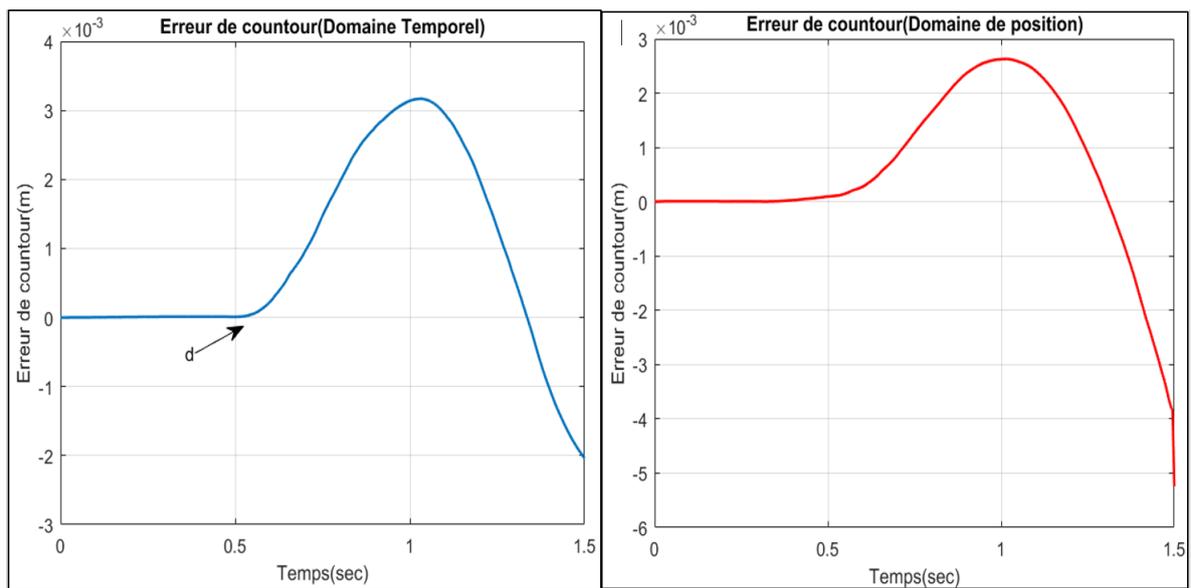


FIGURE 4.56 – L'erreur du Contour Ligne (Forwarding)

4.5.4 Commande par MG

On prend $a = 0.1$ $b = 0.1$ et $c = 0.1$ (car la méthode est robuste dans le domaine temporel), avec le même régulateur utilisé dans la partie simulation.

Résultats de suivi du contour ligne

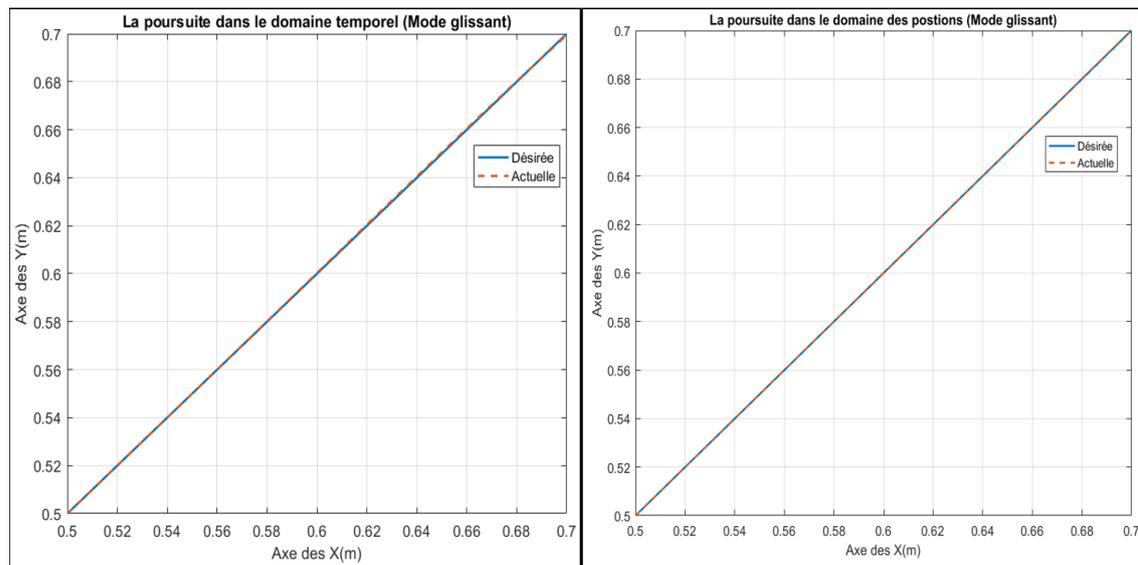


FIGURE 4.57 – Résultat de suivi du contour (contour Ligne)

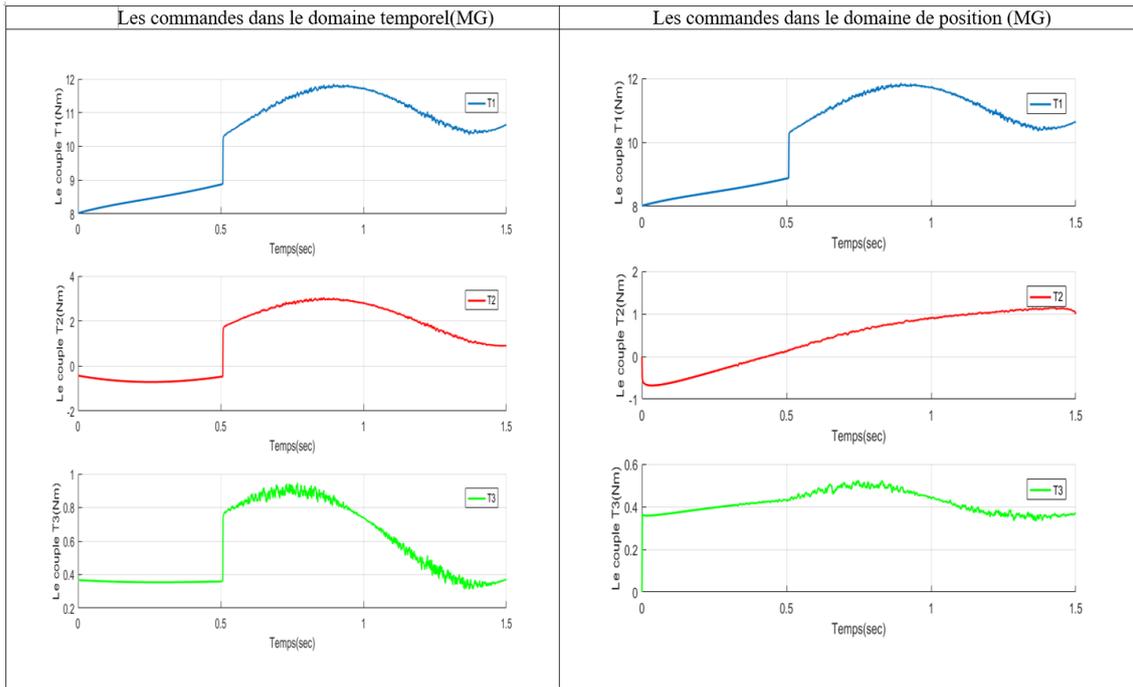


FIGURE 4.58 – Commandes MG dans domaine temporel/position (Contour Ligne)

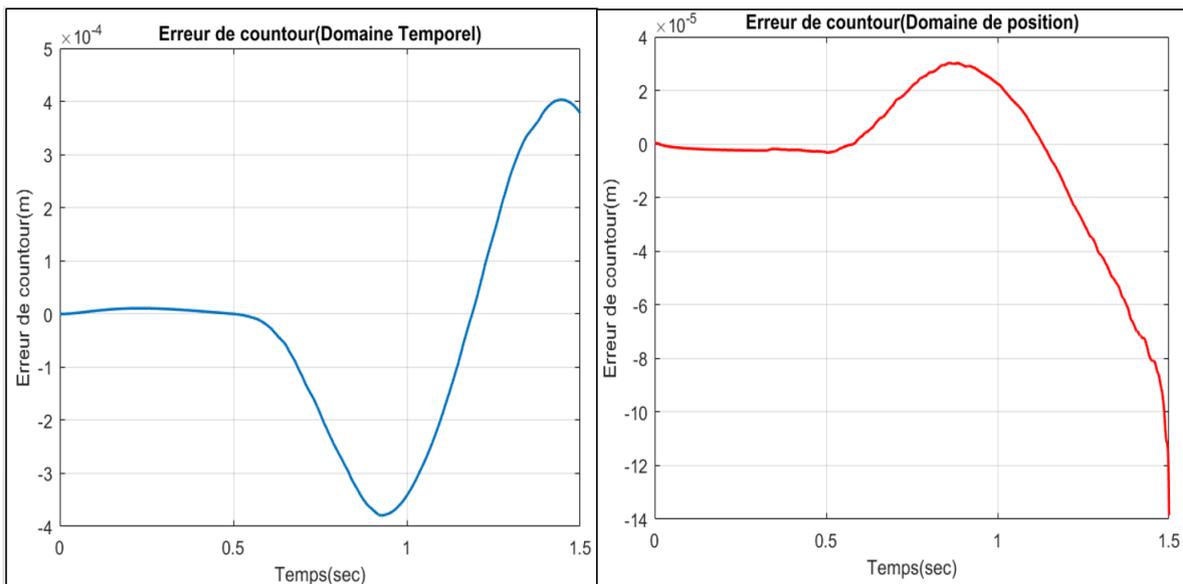


FIGURE 4.59 – L'erreur du Contour Ligne (MG)

4.5.5 Commande par CAI-MG

On prend $a = 1$ $b = 1$ et $c = 1$ (pour pouvoir tester l'effet de l'apprentissage jumelé avec la mode glissant robuste), avec le même régulateur utilisé dans la partie simulation .

Résultats de suivi du contour ligne

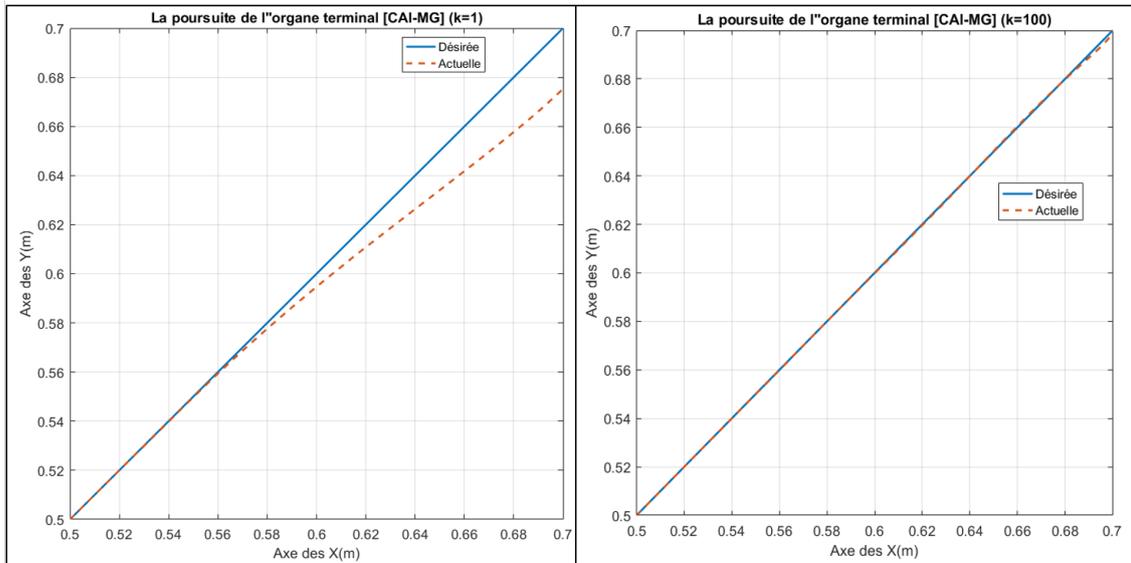


FIGURE 4.60 – Résultat de suivi du contour (contour Ligne)

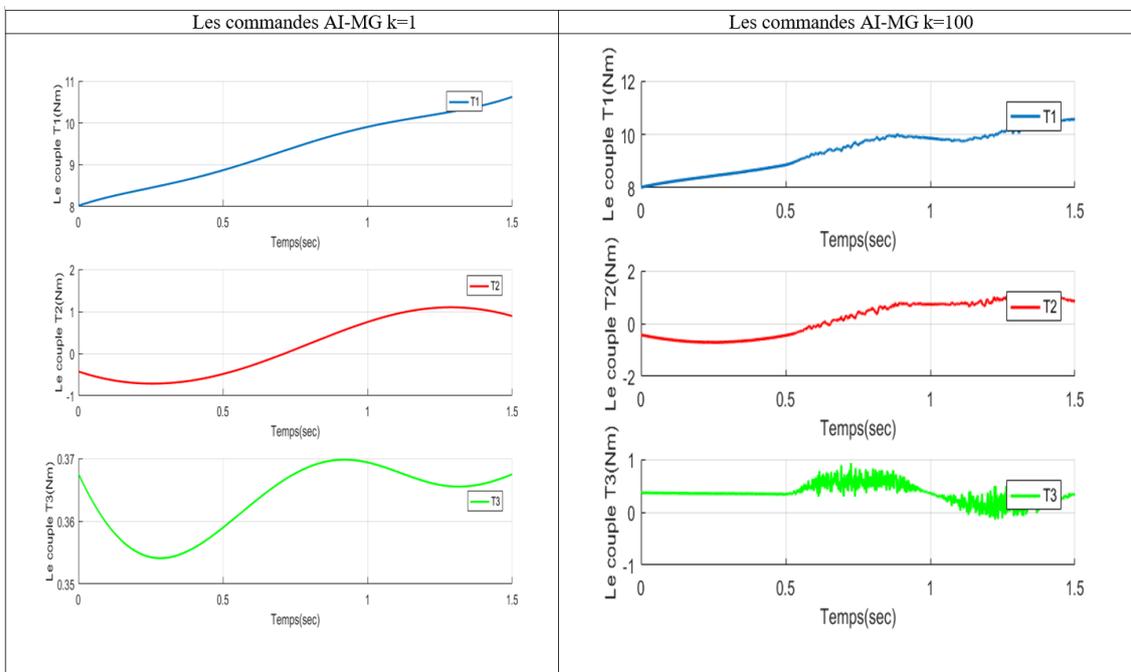


FIGURE 4.61 – Commandes CAI-MG dans (Contour Ligne)

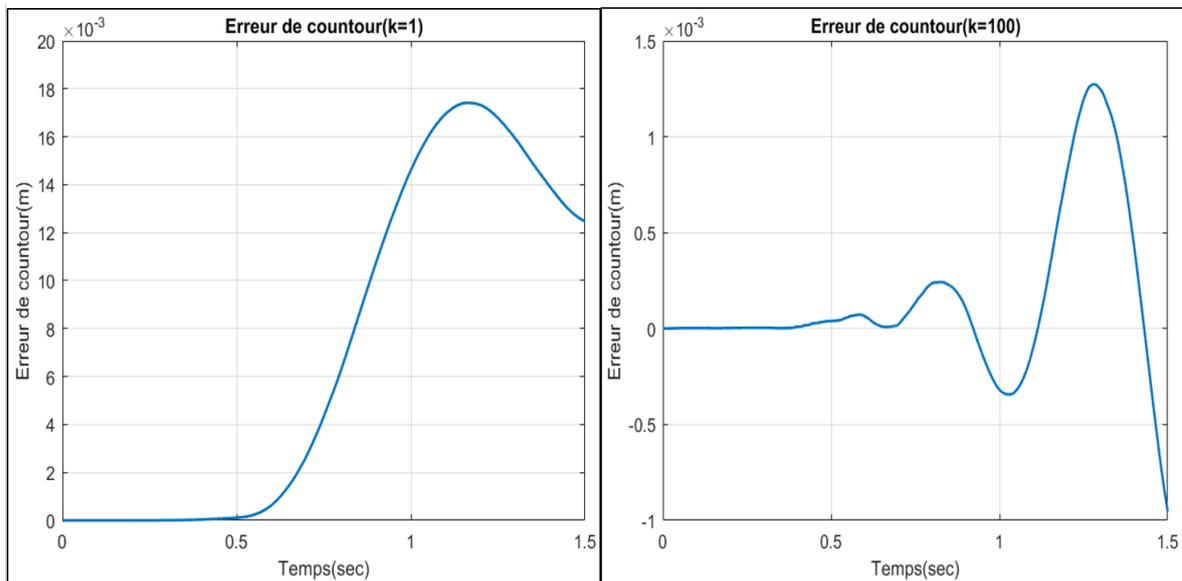


FIGURE 4.62 – L’erreur du Contour Ligne (CAI-MG)

Commentaires

Comme la commande par Forwarding n’est pas robuste dans le domaine temporelle (figure (4.54)), la commande dans le domaine de position à améliorer l’erreur de contour par rapport au domaine temporel (figure (4.56)) mais le résultat reste insuffisant pour dire que la commande par Forwarding(en position) est robuste.

La commande par MG est robuste dans le domaine temporelle et position (figure (4.57)), cependant on remarque que la commande temporelle présente des variations de hautes fréquences (figure (4.58)), l’erreur de contour diminue considérablement (10 fois) dans le domaine de position (figure (4.59)) en plus la commande devient plus lisse et admissible (figure (4.58)). Au fur et à mesure on augmente les itérations la commande par CAI-MG devient de plus en plus robuste (figure (4.60)), l’erreur de contour s’améliore lentement (figure (4.62)), et on risque d’avoir à un effet de broutement donc une commande inadmissible ((figure (4.61)).

4.5.6 Étude qualitative

Tableau d'appréciations des performances des commandes

Partant des résultats de l'étude quantitative ,on peut présenter des appréciations dans le tableau suivant :

Commande	COMPLEX	E.C	R.S	E.CO	E.CCO	ROB
PD-MG (Temps)	++	--	--	++	*	++
PD-MG (Position)	+	++	++	+	-	+++
FD (Temps)	+	++	++	++	*	--
FD (Position)	--	-	++	++	-	-
MG (Temps)	++	+	++	++	*	++
MG (Position)	+	+++	+++	+	-	+++
CAI-MG	++	++	+++	+++	---	++

TABLE 4.14 – Tableau d'appréciations des performances des commandes

+++ Très bien ++ Bien + Assez bien

--- Très mauvais -- Mauvais - Passable

COMPLEX : Complexité de synthèse et d'implémentation.

EC : Erreur de contour .

R.S : Rapidité du système commandé.

E.CO : Énergie de commande(ISV).

E.C.CO : Énergie cumulée de commande(CISV) (cas domaine de position et CAI).

ROB : Robustesse du système commandé.

Conclusion

Dans ce chapitre ,nous avons simulé les 3 commandes (PD-MG,FD,MG) en temps et positions pour le robot manipulateur , et la commande (CAI) en temps pour les deux contours , nous avons ainsi comparé les performances de chaque commande d'une manière quantitative et qualitative ,et enfin nous avons donné un tableau d'appréciations qui résume le travail effectué dans ce chapitre .

CONCLUSION GÉNÉRALE

Conclusion Générale

Le travail présenté dans ce mémoire concerne le développement de commandes non linéaires appliquées au suivi de contours pour les robots industriels dans les domaines temporel et de position.

Les robots manipulateurs en particulier ont de nombreuses applications, de la production électrique et électronique aux interventions chirurgicales et à l'exploration spatiale. Un domaine de recherche important pour la robotique est le développement de systèmes de contrôle avancés dans le but d'améliorer la performance du suivi de contour, et c'est l'objectif principal de ce travail. En effet la motivation du choix de l'approche basée sur le domaine de position est qu'elle contribue efficacement à la réduction des erreurs de suivi de contour notamment dans les processus industriels et la robotique médicale exigeant une très haute performance de suivi de contour. On a expliqué pourquoi la méthode en position .

Dans ce mémoire, des contrôleurs non linéaires ont été proposés dans le domaine de position et comparés à leurs contrôleurs équivalents synthétisés dans le domaine temporel et ce pour deux structures différentes de robots. Cela a été réalisé en transformant la dynamique du système robotisé du domaine temporel vers le domaine de position par une formulation particulière. Dans le domaine de position, l'un des signaux d'états ou variables du système est pris comme référence (maître) et échantillonné à une nouvelle fréquence relative à sa propre dynamique. L'agent maître remplacera la variable indépendante temps 't' pour les reste des états et variables constitutives du système. Par conséquent, l'agent maître du système génère une erreur de suivi nulle et sa contribution donc à la création d'erreur de suivi de contour au niveau de l'organe termina est éliminée. Seules les erreurs de suivi des agents esclaves contribuent à l'erreur globale de suivi de contour par le système considéré. La stabilité du contrôleur de domaine de position proposé a été démontrée avec la méthode de Lyapunov. En plus de la comparaison entre les contrôleurs temporels et leurs homologues formulés dans le domaine de position, une autre approche de commandes des robots dite commande par apprentissage itératif a été synthétisée et simulée à une analyse comparative a été effectuée pour tous les contrôleurs et pour le suivi de contours linéaires et no linéaires.

Les résultats indiquent que des performances très améliorées ont été obtenues par les contrôleurs synthétisés dans le domaine de position par rapport à leurs équivalents temporels. Cela se traduit par des erreurs de suivi inférieures pour les variables esclaves. Ce résultat a été appuyé par une étude comparative avancée concluant sur les aspects quantitatifs et qualitatifs de performance des deux approches de commandes temporelle et celle dans le domaine de position. En conclusion, nous pouvons dire que les résultats obtenus lors de l'application ont confirmé l'avantage et l'apport de l'approche de commande dans le domaine de position pour l'amélioration du suivi de contour. Les résultats obtenus nous permettent de conclure que les algorithmes de formulation de la nouvelle méthode de commande sont très efficaces pour et nous encouragent à l'explorer dans d'autres applications industrielles.

Ce travail peut faire l'objet de travaux futurs tels que :

Tout d'abord, des expériences pratiques devraient être effectuées pour valider les simulations et étudier toutes les difficultés imprévues.

Le contrôleur doit également être utilisé dans différentes configurations de robots (parallèles, spatiaux, etc.) ainsi que pour des formations de robots coordonnés.

Enfin, les contrôleurs dans le domaine de position utilisent des trajectoires de références différentes de leurs homologues définies dans le domaine temporel. Par conséquent, une technique d'optimisation de planification de la trajectoire transformée devrait être développée afin de définir des trajectoires dans le domaine de position qui produiraient les meilleures performances pour le contrôleur de domaine de position.

BIBLIOGRAPHIE

- [1] Koren, Y. *Cross-Coupled Biaxial Computer Control for Manufacturing Systems*. ASME Journal of Dynamic Systems, Measurement, and Control, 1980, 265-272.
- [2] Tomei, P. *Adaptive PD controller for robot manipulators*. IEEE Transactions on Robotics Automation, 1991, Vol. 7, p 565-570.
- [3] Slotine, J. E. and Li, W. *Applied Nonlinear Control* : New Jersey : Prentice Hall, 1991, 461 p. ISBN 978-0130408907
- [4] P.R. Ouyang and T. Dam. *Position Domain PD Control for Contour Tracking*. Proceedings of the ASME 2010 International Mechanical Engineering Congress and Exposition (IMECE 2010), pp. 657-66 ,2010.
- [5] W.H. Yue, V. Pano, P.R. Ouyang, and Y.Q. Hu *Model-Independent Position Domain Sliding Mode Control for Contour Tracking of Robotic Manipulator*. International Journal of Systems Science, March, 2016.
- [6] Ouyang, P.R. and Pano, V., and Acob, J. *Position Domain Contour Control for Multi-DOF Robotic System*. Mechatronics, 2013, Vol. 23, No. 8, 1061-1071.
- [7] Pano, Vangjel. *Position Domain Synchronization Control For Robotic Manipulators*. 112 p Thèse de Maîtrise : Génie Aérospatial : Ontario, Canada, Ryerson University Toronto : 2011.
- [8] Astrom, K. J., and Hagglund, T. *The Future of PID Control..* Engineering Practice, 1163-1175 ,2001.

-
- [9] Bartolini, G., Pisano, A., Punta, E., and Usai, E. *A Survey of Applications of Second- Order Sliding Mode Control to Mechanical Systems.* International Journal of Control, 875-892 ,2003.
- [10] Dombre, E., and Khalil, W. *Robot Manipulators : Modeling, Performance Analysis and Control.* Newport Beach : Wiley Publishing ,2006.
- [11] Sabanovic, A., Fridman, M., and Spurgeon, S. *Variable Structure Systems, from principles to implementation.* London, UK : The Institution of Engineering and Technology ,2004.
- [12] Ouyang, P. R., and Dam, T. *Position Domain PD Control : Stability and Comparison.* The 6th IEEE Conference on Industrial Electronics and Applications, 962-976 ,2011.
- [13] Yokokohji, Y., and Yoshikawa, T. *Bilaterl Control of Master-Slave Manipulators for Ideal Kinesthetic Coupling-Formulation and Experiment.* IEEE Transactions on Robotics and Automation, 605-620 ,1994.
- [14] Alford, C., and Belyeu, S. *Coordinated control of two robot arms.* Proceedings of IEEE International Conference in Robotics and Automation, 468-473 ,1984.
- [15] E. Curcio, E. Drioli. *Membrane distillation and related operations—a review.* Purif. Rev. 34 35–86 ,2005.
- [16] Rodriguez-Angeles, A., and Nijmeijer, H. *Coordination of two robot manipulators based on position measurements only.* International Journal of Control, 1311-1323 ,2001.
- [17] Ouyang, P. R., Pano, V., and Dam, T. *PID Contour Tracking Control in Position Domain.* 21st IEEE International Symposium on Industrial Electronics, 1297-1302 ,2012.
- [18] Astrom, K. J. *Event Based Control.* Analysis and Design of Nonlinear Control Systems : In Honor of Alberto Isidori, 127-147 ,2007.
- [19] Inoue, T., Nakano, M., Matsumoto, S., and Baba, H. *High Accuracy Contour Control of Proton Synchrotron Magnet Power Supply.* Proceeding of the 8th World Conference of IFAC, 216-221 ,1981.
- [20] Wang, Y., Gao, F., and Doyle, F. J. *Survey on Iterative Learning Control, Repetitive control, and Run-to-Run Control.* Journal of Process Control, 1589-1600 ,2009.

-
- [21] Hammoud Saari. *Commande répétitive de procédés échantillonnés à déphasage non minimal.* ,1996.
- [22] Y. Chen, and C. Wen. *Iterative Learning Control : Convergence, Robustesse and Control.* Springer, LNCIS 248 ,1999.
- [23] Amann, N., D.H. Owens and E. Rogers. *Iterative Learning Control : Convergence, Robustesse and Control.* Springer, LNCIS 248 ,1999.
- [24] Farah BOUAKRIF. *Commande par Apprentissage Itératif des Robots Manipulateurs.* Thèse de Magistère : Automatique : Alger, Ecole Nationale Polytechnique : 2008.
- [25] Sholten Owens and E. Rogers. *Iterative Learning Control : Convergence, Robustesse and Control.* Springer, LNCIS 248 ,2000.
- [26] R. Sepulchre, M. Jankovic and P. Kokotovic. *Constructive Nonlinear Control.* London : Springer-Verlag, 1997, 313 p.ISBN 978-1-4471-1245-7.
- [27] R. Sepulchre, M. Jankovic and P.V. Kokotovic. *Interlaced systems and recursive designs for global stabilization.* Proc. of the 4th European Control Conference, 1997, Brussels, WE-E-A (6 pages).
- [28] RW. Respondek and I. Tall. "*Strict feedforward form and symmetries of nonlinear control systems*". 43rd IEEE Conference on Decision and Control, pp. 1611-1616, Atlantis, Paradise Island, Bahamas ,2004.
- [29] W. Su and M. Fu. "*Robust Nonlinear Control : Beyond Backstepping Nonlinear Forwarding*". Proceeding of the 38th conference on Decision and Control, Phoenix, Arizon, USA ,1999.
- [30] Fang RW, Chen JS. *Cross-coupling control for a direct-drive robot.* JSMA Ser C 2002 ;45(3) :749–57 ,2002.
- [31] Barton KL, Alleyne AG. *A cross-coupled iterative learning control design for precision motion control.* IEEE Control Syst Technol 2008 ;16(6) :1218–30 ,2008.
- [32] Wu J, Wang JS, Wang LP, Li TM. *Dynamics and control of a planar 3-DOF parallel manipulator with actuation redundancy..* Mechan Mach Theory ;44(4) :835–49. ,2009.

-
- [33] Koren, Y., and Lo, C.-C. *Variable-Gain Cross-Coupling Controller for Contouring*. ASME Journal of Dynamic Systems, Measurement, and Control, 40(1), 265-272 ,1991.
- [34] Cheng, M. L. *Motion Controller Design for Contour-Following Tasks Based on Real- Time Contour Error Estimation*. AIEEE Transactions on Industrial Electronics, 1686- 1695 ,2007.
- [35] Ouyang, P.R., Dam, T., and Pano, V. *Cross-coupled PID control in position domain for contour tracking*. Robotica, 33(6), 1351–1374 ,2015.
- [36] Craig, J. *Introduction to Robotics : Mechanics and Control*. Upper Saddle River : Peason Prentice Hall ,2005.
- [37] Ouyang, P.R., Acob, J.,and Pano, V. *PD with sliding mode control for trajectory tracking of robotic system*. Robotics and Computer-Integrated Manufacturing, 30(2), 189–200 ,2014.
- [38] Brockett RW. *Asymptotic stability and feedback stabilization*. In : Brockett RW, Millman RS, Sussmann HJ, editors. Differential Geometric Control Theory, Boston, MA : Birkhauser p. 181–91 ,1983.
- [39] Kanayama Y, Kimura Y, Miyazaki F, Noguchi T. *A stable tracking control method for an autonomous mobile robot*. vol. 1. Proceedings of IEEE International Conference on Robotics and Automation, Cincinnati, Ohio, p. 384–9 ,1990.
- [40] Canudas de Wit C, Khenouf H, Samson C, Sordalen OJ. *Nonlinear control design for mobile robots*. In : Zheng YF, editor. Recent trends in Mobile robots, World Scientific, p. 121–56 ,1993.
- [41] Guldner J, Utkin VI. *Stabilization of nonholonomic mobile robot using Lyapunov functions for navigation and sliding mode control*. Control-Theory Adv Technol ;10(4) :635–47 ,1994.
- [42] Sarkar N, Yun X, Kumar V. *Control of mechanical systems with rolling constraints : application to dynamic control of mobile robots*. Int J Robot Res ;13(1) :55–69 ,1994.
- [43] Farzad Pourboghrat *, Mattias P. Karlsson. *Adaptive control of dynamic mobile robots with nonholonomic constraints* . Computers and Electrical Engineering 28 ;p 241–253 ,2002.

- [44] Ouyang, P.R., Dam, T., and Pano, V. *Cross-coupled PID control in position domain for contour tracking* . Robotica, 33(6), 1351–1374. ,2015.
- [45] Arendt, W., and Schleuch, W.P. *Mathematical analysis of evolution, information and complexity* . Weiheim :Wiley-VCH ,2009.
- [46] H.S. Ahn, C.H. Choi, and K.B. Kim *Iterative Learning Control for a Class of Nonlinear Systems*. Automatica,1993, Vol. 29, No.6, p. 1575-1578.
- [47] Tayebi, A. *Adaptive iterative learning control for robot manipulators*. Automatica ,2004, Vol. 40,p 1195-1203.

ANNEXES

ANNEXE 1

Modèle cinématique et dynamique du robot manipulateur

Modèle géométrique direct

$$\begin{cases} x_{ef} = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) \\ y_{ef} = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) \\ q_{ef} = q_1 + q_2 + q_3 \end{cases}$$

Modèle géométrique inverse

La position

$$\begin{cases} x_2 = x_{ef} - l_3 \cos(q_{ef}) \\ y_2 = y_{ef} - l_3 \sin(q_{ef}) \\ \cos(q_2) = \frac{x_2^2 + y_2^2 - l_1^2 - l_2^2}{2l_1 l_2} \\ \sin(q_2) = \sqrt{1 - (\cos(q_2))^2} \end{cases}$$

Alors,

$$\begin{cases} q_2 = \text{atan2}(\cos(q_2), \sin(q_2)) \\ q_1 = \text{atan2}(y_2, x_2) - \text{atan2}(l_1 \sin(q_2) + l_2 \cos(q_2)) \\ q_3 = q_{ef} - q_1 - q_2 \end{cases}$$

Modèle cinématique inverse

$$\begin{cases} \dot{q}_1 = \frac{\cos(q_1+q_2)}{l_1 \sin(q_2)} \dot{x}_{ef} + \frac{\sin(q_1+q_2)}{l_1 \sin(q_2)} \dot{y}_{ef} + \frac{l_3 \sin(q_3)}{l_1 \sin(q_2)} \dot{q}_{ef} \\ \dot{q}_2 = -\frac{l_2 \cos(q_1+q_2) + l_1 \cos(q_1)}{l_1 l_2 \sin(q_2)} \dot{x}_{ef} - \frac{l_2 \sin(q_1+q_2) + l_1 \sin(q_1)}{l_1 l_2 \sin(q_2)} \dot{y}_{ef} - \frac{l_3 \sin(q_2+q_3) + l_3 \cos(q_3)}{l_1 l_2 \sin(q_2)} \dot{q}_{ef} \\ \dot{q}_3 = \frac{\cos(q_1)}{l_2 \sin(q_2)} \dot{x}_{ef} + \frac{\sin(q_1)}{l_2 \sin(q_2)} \dot{y}_{ef} + \left(\frac{l_3 \sin(q_2+q_3)}{l_2 \sin(q_2)} + 1 \right) \dot{q}_{ef} \end{cases}$$

Modèle dynamique

La matrice d'inertie

$$\left\{ \begin{array}{l} M_{11} = I_1 + I_2 + I_3 + m_1 r_1^2 + m_2 (l_1^2 + r_2^2 + 2l_1 r_2 \cos(q_2)) + m_3 (l_1^2 + l_2^2 + r_3^2 + 2l_1 l_2 \cos(q_2) \\ \quad + 2l_2 r_3 \cos(q_3) + 2l_1 r_3 \cos(q_2 + q_3)) \\ M_{12} = M_{21} = I_2 + I_3 + m_2 (r_2^2 + l_1 r_2 \cos(q_2)) + m_3 (l_3^2 + r_3^2 + l_1 l_2 \cos(q_2) \\ \quad + 2l_2 r_3 \cos(q_3) + l_1 r_3 \cos(q_2 + q_3)) \\ M_{13} = M_{31} = I_3 + m_3 (r_3^2 + l_2 r_3 \cos(q_3) + l_1 r_3 \cos(q_2 + q_3)) \\ M_{22} = I_2 + I_3 + m_2 r_2^2 + m_3 (l_2^2 + r_3^2 + 2l_2 r_3 \cos(q_3)) \\ M_{23} = M_{32} = I_3 + m_3 (l_2^2 + r_3^2 + 2l_2 r_3 \cos(q_3)) \\ M_{33} = I_3 + m_3 r_3^2 \end{array} \right.$$

La matrice des effets Coriolis et centrifuge

$$C_{11} = \begin{bmatrix} 0 \\ m_2 l_1 r_2 \sin(q_2) - m_3 (l_1 r_2 \sin(q_2) + l_1 r_3 \sin(q_2 + q_3)) \\ -m_3 (l_1 r_2 \sin(q_2) + l_1 r_3 \sin(q_2 + q_3)) \end{bmatrix}^T \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$C_{12} = \begin{bmatrix} m_2 l_1 r_2 \sin(q_2) - m_3 (l_1 r_2 \sin(q_2) + l_1 r_3 \sin(q_2 + q_3)) \\ -m_2 l_1 r_2 \sin(q_2) - m_3 (l_1 l_2 \sin(q_2) + l_1 r_3 \sin(q_2 + q_3)) \\ m_3 (l_1 r_3 \sin(q_3) + l_1 r_3 \sin(q_2 + q_3)) \end{bmatrix}^T \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$C_{13} = \begin{bmatrix} -m_3 (l_1 r_2 \sin(q_2) + l_1 r_3 \sin(q_2 + q_3)) \\ m_3 (l_2 r_3 \sin(q_3) + l_1 r_3 \sin(q_2 + q_3)) \\ -m_3 (l_2 r_3 \sin(q_3) + l_1 r_3 \sin(q_2 + q_3)) \end{bmatrix}^T \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$C_{21} = \begin{bmatrix} m_3 (l_2 r_3 \sin(q_3) + l_1 r_3 \sin(q_2 + q_3)) + m_2 l_1 r_2 \sin(q_2) \\ 0 \\ m_3 l_2 r_3 \sin(q_3) \end{bmatrix}^T \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$C_{22} = \begin{bmatrix} 0 \\ 0 \\ -m_3 l_2 r_3 \sin(q_3) \end{bmatrix}^T \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$C_{23} = \begin{bmatrix} m_3 l_2 r_3 \sin(q_3) \\ -m_3 l_2 r_3 \sin(q_3) \\ -m_3 l_2 r_3 \sin(q_3) \end{bmatrix}^T \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$C_{31} = \begin{bmatrix} m_3(l_2 r_3 \sin(q_3) + l_1 r_3 \sin(q_2 + q_3)) \\ m_3 l_2 r_3 \sin(q_3) \\ 0 \end{bmatrix}^T \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$C_{31} = \begin{bmatrix} m_3 l_2 r_3 \sin(q_3) \\ m_3 l_2 r_3 \sin(q_3) \\ 0 \end{bmatrix}^T \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

$$C_{31} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}$$

La matrice des effets dûs à la gravité

$$G_{11} = g(m_1 r_1 \cos(q_1) + m_2(l_1 \cos(q_1) + r_2 \cos(q_1 + q_2)) + m_3(l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) + r_3 \cos(q_1 + q_2 + q_3)))$$

$$G_{12} = g(m_2 r_2 \cos(q_1 + q_2) + m_3(l_2 \cos(q_1 + q_2) + r_3 \cos(q_1 + q_2 + q_3)))$$

$$G_{13} = g m_3 r_3 \cos(q_1 + q_2 + q_3)$$