

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



ÉCOLE NATIONALE POLYTECHNIQUE
Département d'Automatique

Laboratoire de Commande des Processus



THÈSE DE DOCTORAT

Présentée par :

Chems Eddine BOUDJEDIR

Ingénieur et Master en Automatique de l'ENP

En vue de l'obtention du titre de

Docteur

en Automatique

Option : Automatique des Systèmes Industriels

Thème

**Contribution à la Commande par Apprentissage Itératif
des Robots Parallèles de Type Delta.
Théorie, Simulation et Expérimentation**

Soutenue publiquement, le 01/ 12/ 2019 devant le jury :

Président	BOUDJEMA Farès	Professeur	ENP, Alger
Directeur	BOUKHETALA Djamel	Professeur	ENP, Alger
Co-Directeur	BOURI Mohammed	Maitre de Conférence A	EPFL, Lausanne
Examineurs	TOUMI Redouane	Professeur	USTHB, Alger
	BENALIA Atallah	Professeur	Univ. Laghouat
	ILLOUL Rachid	Maitre de Conférence A	ENP, Alger

ENP 2019

Thèse préparée au Laboratoire de Commande des Processus (LCP) - École Nationale Polytechnique (ENP)
10, Avenue des Frères Oudek, Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie

En collaboration avec le Laboratoire des Systèmes Robotiques (LSRO) de l'École Polytechnique Fédérale de Lausanne
(EPFL), Station 9, 1015 Lausanne, Suisse.

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



ÉCOLE NATIONALE POLYTECHNIQUE
Département d'Automatique

Laboratoire de Commande des Processus



THÈSE DE DOCTORAT

Présentée par :

Chems Eddine BOUDJEDIR

Ingénieur et Master en Automatique de l'ENP

En vue de l'obtention du titre de

Docteur

en Automatique

Option : Automatique des Systèmes Industriels

Thème

**Contribution à la Commande par Apprentissage Itératif
des Robots Parallèles de Type Delta.
Théorie, Simulation et Expérimentation**

Soutenue publiquement, le 01/ 12/ 2019 devant le jury :

Président	BOUDJEMA Farès	Professeur	ENP, Alger
Directeur	BOUKHETALA Djamel	Professeur	ENP, Alger
Co-Directeur	BOURI Mohammed	Maitre de Conférence A	EPFL, Lausanne
Examineurs	TOUMI Redouane	Professeur	USTHB, Alger
	BENALIA Atallah	Professeur	Univ. Laghouat
	ILLOUL Rachid	Maitre de Conférence A	ENP, Alger

ENP 2019

Thèse préparée au Laboratoire de Commande des Processus (LCP) - École Nationale Polytechnique (ENP)
10, Avenue des Frères Oudek, Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie

En collaboration avec le Laboratoire des Systèmes Robotiques (LSRO) de l'École Polytechnique Fédérale de Lausanne
(EPFL), Station 9, 1015 Lausanne, Suisse.

ملخص - الهدف من هذا العمل هو تحسين جودة نظام التحكم بالتعلم المرحلي المخصص لتتبع المسارات للأنظمة اللاخطية المعرضة الى الاضطرابات الخارجية و اخطاء النمذجة. لقد استعمل شرط المحاذاة العملي مكان شرط اعادة الضبط الكلاسيكي. من اجل الاخذ بعين الاعتبار كل الحالات الممكنة من مسارات الالتقاط و الوضع تم اقتراح ثلاث قوانين تحكم من اجل المسارات المعادة وقانونا تحكم من اجل المسارات الغير معادة. لقد تم دراسة استقرار طرق التحكم المقترحة باستعمال طريقة ليابونوف ولقد بينت هذه الدراسة ان خطأ التتبع ومعدل خطأ التتبع يؤول إلى صفر من خلال الاعادات. لقد تم التحقق من فاعلية قوانين التحكم المقترحة عن طريق التجارب العملية التي تم تنفيذها على الروبوت المتوازي من نوع *DeltaISIR88* مع ادراج دراسة مقارنة مع قوانين تحكم اخرى.

كلمات مفتاحية - التحكم بالتعليم المرحلي، الشرط المحاذاة، المسارات الغير معادة، طريقة ليابونوف، الروبوت المتوازي دلتا.

Abstract—The main objective of this work is to improve the iterative learning control performances of the trajectory tracking problem in the MIMO systems subjected to external disturbances and model uncertainty. The practical alignment condition has been considered in this work instead of the classical resetting condition. In order to consider all the possible pick and place operations, five control laws have been proposed for repetitive trajectories and nonrepetitive trajectories. Asymptotic convergence has been proven for the five controllers using Lyapunov theory. It is shown that the tracking error and the tracking error rate converge to zero through iterations. Experiments and simulations have been carried out on the parallel Delta ISIR88 robot to demonstrate the effectiveness of the proposed approaches. Several comparative studies have been also presented.

Keywords : Iterative Learning Control, Alignment Condition, Nonrepetitive Trajectories, Lyapunov Theory, Delta Robot ISIR88.

Résumé—Le travail présenté dans cette thèse concerne l'amélioration des performances de la commande par apprentissage itératif (CAI) dédiée aux systèmes multivariabes soumis aux perturbations externes et avec incertitude de modélisation. La condition classique de réinitialisation de l'erreur a été remplacée par la condition pratique d'alignement. Afin de généraliser l'utilisation de la CAI lors des opérations de prise et de pose, trois lois de commande ont été proposées dans le cas des trajectoires répétitives et deux lois de commande dans le cas des trajectoires non répétitives. La convergence asymptotique à travers les itérations a été prouvée pour les cinq lois développées en utilisant la théorie de Lyapunov. Des simulations ont été effectuées avec l'ensemble des approches proposées. Des implantations expérimentales ont été également réalisées sur le robot Delta ISIR88 montrant la faisabilité et l'efficacité des lois de commande proposées. Des études comparatives ont été aussi menées.

Mots-clés : Commande par Apprentissage Itératif, Condition d'Alignement, Trajectoires non Répétitives, Théorie de Lyapunov, Robot Delta ISIR88.

Remerciement

Les travaux de recherche présentés dans cette thèse ont été effectués au Laboratoire de Commande des Processus (LCP) de l'École Nationale Polytechnique (ENP) en collaboration avec le Laboratoire des Systèmes Robotiques (LSRO) de l'École Polytechnique Fédérale de Lausanne (EPFL).

Je voudrais tout d'abord remercier M. Djamel BOUKHETALA, Professeur à l'ENP et Directeur de cette thèse ainsi que M. Mohammed BOURI, Maître de Conférence A à l'EPFL et Co-Directeur de cette thèse, de m'avoir proposé ce sujet ainsi que leurs précieux conseils dans le domaine de la recherche scientifique, leurs précieuses directives et leurs suivis continus tout au long de la période de la recherche.

Mes remerciements vont également à M. Farès BOUDJEMA, Professeur à l'ENP, pour son support et son appui moral tout au long de mes études à l'ENP. Qu'il soit convaincu de notre sincère reconnaissance pour avoir accepté de juger ce travail et m'avoir fait l'honneur de présider le jury de soutenance.

Je tiens à exprimer toute ma gratitude à M. Redouane TOUMI, Professeur à l'Université des Sciences et de la Technologie Houari Boumedienne (USTHB), M. Atallah BENALIA, Professeur à l'Université de Laghouat, et M. Rachid ILLOUL, Maître de Conférence A à l'ENP de l'honneur qu'ils m'ont fait en acceptant de juger ce présent travail de recherche et de participer au jury en qualité d'examineurs.

Je tiens à remercier tous les enseignants ayant contribué à ma formation et ce depuis le tronc commun jusqu'à la dernière année de graduation. J'adresse également mes sincères remerciements à ma famille, mes amis et l'ensemble des doctorants/docteurs au sein du Laboratoire de Commande des Processus (LCP), qui m'ont aidé de prêt ou de loin dans l'élaboration de cette thèse.

Table des Matières

Liste des Figures

Liste des Tableaux

Introduction Générale 10

Chapitre 1: Introduction à la Commande par Apprentissage Itératif 14

1.1	Introduction	14
1.1.1	CAI exemples simples	14
	Exemple 1 :	14
	Exemple 2 :	15
1.2	Définitions	18
1.2.1	Commande par apprentissage itératif en boucle ouverte	18
1.2.2	Commande par apprentissage itératif en boucle fermée	18
1.3	Hypothèses fondamentales de la commande CAI	19
1.4	Avantages et inconvénients de la CAI	19
	Avantages :	20
	Inconvénients :	20
1.5	État de l'art sur la commande par apprentissage itératif	20
1.5.1	CAI classique	20
1.5.2	CAI adaptative	21
1.5.3	CAI robuste	22
1.5.4	CAI basée-modèle	22
1.5.5	Application pratique de la CAI	22
1.5.6	Rappels mathématiques	22
1.6	Conclusion	23

Chapitre 2: Robot Parallèle du Type Delta 24

2.1	Généralités sur les robots parallèles	24
2.1.1	Définition	24
2.1.2	Historique	24
2.1.3	Caractéristiques des robots parallèles	25
2.2	Robot Delta	27
2.2.1	Historique	27
2.2.2	Spécifications du Robot Delta	27
2.2.3	Types de robot Delta	28
	Delta linéaire	28
	Delta à actionnement rotatif	29
2.3	Description du robot Delta ISIR88 disponible au laboratoire de commande des processus (LCP)	29

2.4	Modélisation du Robot Delta	30
2.4.1	Modèle géométrique direct	30
2.4.2	Modèle géométrique inverse	35
2.4.3	Modèle dynamique	36
	Cinématique de vitesse	36
	Cinématique d'accélération	38
	La dynamique de l'actionneur	38
	Paramètres dynamiques	39
	Calcul du modèle dynamique en utilisant le principe du travail virtuel	39
2.5	Conclusion	41
Chapitre 3: Commande par Apprentissage Itératif à Trajectoires Répétitives des Robots		42
3.1	Introduction	42
3.2	Propriétés générales des robots manipulateurs et hypothèses	42
3.2.1	Propriétés générales des robots manipulateurs rigides	42
3.2.2	Hypothèses de travail	43
3.3	Commande par apprentissage itératif à base de modèle dynamique	44
3.3.1	Conception de la commande	44
3.3.2	Analyse de la convergence	45
3.3.3	Résultats de simulation	48
3.4	Commande combinée PD-CAI type PD	55
3.4.1	Introduction	55
3.4.2	Conception de la commande	55
3.4.3	Analyse de la convergence	56
3.4.4	Résultats expérimentaux	60
3.4.5	Comparaison entre la commande combinée PD-CAI type PD et la commande par couple a priori	62
	La conception de la commande avec couple a priori	68
	Relation entre la commande CCP et la commande PD plus CAI	69
3.4.6	Résultats expérimentaux	69
3.5	Commande par apprentissage itératif adaptative	75
3.5.1	Conception de la commande	75
3.5.2	Analyse de la convergence	76
3.5.3	Résultats expérimentaux	81
3.6	Conclusion	88
Chapitre 4: Commande par Apprentissage Itératif à Trajectoires Non Répétitives		89
4.1	Introduction	89
4.2	Formulation du problème	90
4.3	Commande CAI basée modèle pour les systèmes MIMO à trajectoires non répétitives	92
4.3.1	Conception de la commande	92
4.3.2	Influence du vecteur $\hat{d}_{k-1}(t_k)$	93
4.3.3	Analyse de la convergence	93
4.3.4	Résultats de simulation	96
4.4	Commande par apprentissage itératif sans modèle pour les systèmes non linéaires MIMO	103

4.4.1	Conception de la commande	103
4.4.2	Analyse de la convergence	105
4.4.3	Résultats expérimentaux	109
4.5	Conclusion	115
	Conclusion Générale	116
	Bibliographie	118

Liste des Figures

1.1	Schéma simple de la commande CAI	18
1.2	Schéma de la CAI en boucle ouverte	18
1.3	Schéma de la CAI en boucle fermée	19
2.1	Chaîne cinématique fermée	24
2.2	Mécanisme de Pollard	25
2.3	Simulateur de Vol : application de la Plateforme de Stewart	25
2.4	FlexPicker de ABB	26
2.5	Robot Delta de l'EPFL	27
2.6	Barres parallèles du robot Delta	28
2.7	Delta linéaire	28
2.8	Delta à actionnement rotatif	29
2.9	Schéma mécanique du robot Delta ISIR88 disponible au LCP	29
2.10	Modèle géométrique direct du robot Delta	31
2.11	Un bras du Delta	31
2.12	Vue du dessus de la base du Delta	31
2.13	Schéma d'un bras du Delta	32
2.14	Schéma simplifié d'un bras du Delta	32
2.15	Sphère de centre P et de rayon l_2 passant par C_1	33
2.16	Actionneur avec réducteur	38
2.17	Force gravitationnelle agissant sur la partie supérieure du bras d'un robot Delta	40
3.1	Schéma de la commande par apprentissage itératif à base de modèle dynamique	45
3.2	Robot Delta	48
3.3	Poursuite de trajectoire dans l'espace opérationnel	49
3.4	Critère RMSE suivant les itérations	50
3.5	Critère MaxAE suivant les itérations	50
3.6	Évolution de l'erreur de poursuite en fonction des itérations $k=1,10,30,50,90$. (a) articulation 1, (b) articulation 2, (c) articulation 3	52
3.7	Entrées de commande pour les itérations $k=1,10,30,50,90$. (a) articulation 1, (b) articulation 2, (c) articulation 3	53
3.8	Critère RMSE suivant les itérations sous une charge supplémentaire de 200 g	54
3.9	Critère MaxAE suivant les itérations sous une charge supplémentaire de 200 g	54
3.10	Entrées de commande sous une charge supplémentaire de 200 g. (a) articulation 1, (b) articulation 2	54
3.11	Photographie du banc d'expérimentation avec Robot Delta ISIR88	61
3.12	Évolution des critères en expérimentation suivant les itérations. (a) RMSE, (b) MaxAE	61
3.13	Poursuite de trajectoire dans l'espace opérationnel en expérimentation	62

3.14	Évolution de l'erreur de poursuite en fonction des itérations. (a) articulation 1, (b) articulation 2, (c) articulation 3	63
3.15	Couples de commandes expérimentaux (a) articulation 1, (b) articulation 2, (c) articulation 3	64
3.16	Évolution des critères en expérimentation suivant les itérations sous une accélération de 2G. (a) RMSE, (b) MaxAE	65
3.17	Évolution de l'erreur de poursuite en fonction des itérations. (a) articulation 1, (b) articulation 2, (c) articulation 3	66
3.18	Couples de commandes expérimentaux sous une accélération de 2G. (a) articulation 1, (b) articulation 2, (c) articulation 3	67
3.19	Schéma de la commande PD plus CAI	68
3.20	Schéma de la commande PD plus CCP	68
3.21	Poursuite de trajectoire dans l'espace opérationnel en expérimentation	70
3.22	Évolution de l'erreur de poursuite en fonction des itérations k=1-8. (a) articulation 1, (b) articulation 2, (c) articulation 3	71
3.23	Évolution de l'erreur de poursuite en fonction des itérations k=41-48. (a) articulation 1, (b) articulation 2, (c) articulation 3	72
3.24	Couples de commandes expérimentaux pour k=1-8. (a) articulation 1, (b) articulation 2, (c) articulation 3	73
3.25	Couples de commandes expérimentaux pour k=41-48. (a) articulation 1, (b) articulation 2, (c) articulation 3	74
3.26	Critère RMSE suivant les itérations	75
3.27	Schéma de la commande par apprentissage itératif adaptative	76
3.28	Poursuite de trajectoire dans l'espace opérationnel en expérimentation	82
3.29	Critère RMSE suivant les itérations	82
3.30	Évolution de l'erreur de poursuite en fonction des itérations. (a) articulation 1, (b) articulation 2, (c) articulation 3	83
3.31	Couples de commandes expérimentaux. (a) articulation 1, (b) articulation 2, (c) articulation 3	84
3.32	Critère RMSE suivant les itérations sous une accélération de 1.5 G	85
3.33	Évolution de l'erreur de poursuite en fonction des itérations sous une accélération de 2G. (a) articulation 1, (b) articulation 2, (c) articulation 3	86
3.34	Couples de commandes expérimentaux sous une accélération de 2G. (a) articulation 1, (b) articulation 2, (b) articulation 3	87
4.1	Trajectoire désirée. (a) articulation 1, (b) articulation 2, (c) articulation 3	97
4.2	Poursuite de trajectoire dans l'espace opérationnel	98
4.3	Critère MaxAE suivant les itérations pour le 1-ère cas. (a) articulation 1, (b) articulation 2, (c) articulation 3	99
4.4	Entrées de commande pour le 1-ère cas. (a) articulation 1, (b) articulation 2, (c) articulation 3	100
4.5	Critère MaxAE suivant les itérations pour le 2-ème cas. (a) articulation 1, (b) articulation 2, (c) articulation 3	101
4.6	Entrées de commande pour le 2-ème cas. (a) articulation 1, (b) articulation 2, (c) articulation 3	102
4.7	Trajectoire désirée. (a) articulation 1, (b) articulation 2, (c) articulation 3	110
4.8	Poursuite de trajectoire dans l'espace opérationnel en expérimentation	111
4.9	Évolution de l'erreur de poursuite en fonction du temps. (a) articulation 1, (b) articulation 2, (c) articulation 3	112
4.10	Évolution de l'erreur de poursuite en fonction des itérations. (a) articulation 1, (b) articulation 2, (c) articulation 3	113
4.11	Couples de commandes expérimentaux (a) articulation 1, (b) articulation 2, (c) articulation 3	114

Liste des Tableaux

2.1	Paramètres physiques du robot	30
3.1	Performances à travers les itérations	51
3.2	Performances à travers les itérations sous une charge supplémentaire de 200 g	51
3.3	Performances de poursuite	62
3.4	Performances de poursuite sous une accélération de 2 G.	62
3.5	Performances de poursuite	70
4.1	Durée d'itérations	109

Introduction Générale

AVEC le développement technologique actuel dans le domaine de l'automatisation industrielle, les chaînes de production deviennent de plus en plus robotisées. Lors du processus de fabrication, les robots sont souvent amenés à exécuter des tâches répétitives. Ces tâches sont généralement à exécuter avec un degré de précision élevé. En effet, ces robots doivent être dotés de systèmes de commande assez évolués afin de satisfaire les performances exigées par le processus de fabrication.

Habituellement les lois de commande utilisées sont à paramètres fixes et/ou adaptatifs. Les performances en termes de poursuite de trajectoire ne s'améliorent pas avec la répétition des tâches et pourront même se détériorer après un certain nombre de cycles d'opérations. Afin de remédier à cet inconvénient, une nouvelle technique de commande avait vu le jour. Cette approche de commande, appelée commande par apprentissage itératif (CAI), est capable d'améliorer les performances de poursuite avec la répétitions de la tâche exécutée en exploitant à chaque cycle d'opération les erreurs des cycles précédents. Après un certain nombre de répétitions, le système de commande arrive à minimiser l'erreur de poursuite.

L'idée de base de la commande par apprentissage itératif a été publiée en 1978 au Japon par Uchiyama [1]. Sachant que cet article était rédigé en Japonais, le travail n'a pas été exploité par les chercheurs et par conséquent n'a pas connu de continuité. Quelques années après, la commande par apprentissage itératif avait été introduite pour la première fois d'une manière explicite par Arimoto en 1984 [2], où il avait proposé une CAI classique de type PD et PID. Dans cette publication, il démontrait que l'erreur de poursuite peut converger en utilisant uniquement les erreurs des itérations précédentes. En se basant sur ce résultat intéressant, beaucoup de chercheurs s'étaient penchés sur la question afin d'apporter les améliorations nécessaires. Ainsi, dans [3], Arimoto a proposé une CAI de type PI pour une classe de systèmes robotiques soumis à des perturbations. Il avait été démontré que la trajectoire converge vers un voisinage de la trajectoire désirée et l'erreur de poursuite reste dans une boule dont la borne dépend de l'amplitude des perturbations et des erreurs initiales.

Dans l'objectif d'améliorer l'approche proposée en termes de rejet de perturbations et de poursuite de trajectoires, plusieurs travaux avaient été publiés. Ainsi, dans [4] une commande CAI combinée avec un régulateur à structure variable avait été synthétisée en se basant sur la théorie de Lyapunov. Dans [5], Tayebi avait présenté une CAI adaptative, où la loi de commande est destinée pour résoudre le problème de poursuite de trajectoire des robots manipulateurs. L'efficacité de la commande CAI adaptative avait été montrée aussi bien dans le travail de Tayebi que dans d'autres travaux tels que, [6–8], où, il avait été prouvé que l'erreur de poursuite converge vers zéro avec l'augmentation du nombre d'itérations. Néanmoins, la CAI adaptative est restrictive à une classe particulière de

modèles caractérisés par une fonction linéaire à paramètres inconnus combinée avec une fonction non linéaire connue. Il est à noter que ce type de représentation n'est pas toujours évident, notamment pour les systèmes à structures complexes.

Dans [9], une commande par apprentissage à commutation adaptative avait été proposée. Cette commande est la combinaison entre un régulateur PD classique et une commande CAI de type PD par anticipation (Feedforward). L'approche proposée assure une rapidité dans la convergence. Cependant, la synthèse de la loi de commande élaborée utilise la condition de réinitialisation de l'erreur appelée *resetting condition*.

La condition initiale dans chaque cycle joue un rôle fondamental dans la théorie de la CAI. Dans ce contexte, le travail publié dans [10] avait étudié la CAI sous différentes conditions initiales. Parmi ces conditions, on peut trouver : la condition de réinitialisation de l'erreur, la condition d'alignement de l'erreur (*alignment condition*), la condition qui considère l'erreur initiale constante ainsi que la condition qui considère l'erreur initiale aléatoire et bornée. Il est à remarquer que cette étude concernait uniquement les systèmes non linéaires SISO (single-input-single-output).

Dans le travail de Bouakrif et al. [11], une CAI basée-modèle avait été proposée pour résoudre le problème de la poursuite de trajectoire des robots manipulateurs exécutant des tâches répétitives sous l'effet des perturbations externes. Il avait été démontré que l'erreur de poursuite converge vers zéro sous certaines conditions suffisantes de stabilité. Cependant, l'implémentation de ces lois de commande exige la connaissance des vitesses articulaires du robot. Ce problème avait été traité par la même équipe dans [12] où, une CAI basée-modèle est combinée avec un observateur de vitesse articulaire. La stabilité des deux schémas proposés avait été prouvée sous la condition de réinitialisation de l'erreur.

Récemment, plusieurs commandes CAI robustes sont développées. Généralement, les commandes sont proposées pour les systèmes linéaires à temps discret. On peut citer le travail de Meng [13], où, les auteurs ont développé une CAI de type proportionnel. Dans cette publication le gain d'apprentissage varie en fonction des itérations afin de garantir la stabilité et la robustesse de la commande. Une commande similaire avait été proposée dans [14], où une étude plus approfondie concernant l'influence du gain d'apprentissage sur les performances de convergence est présentée. Cependant, la majorité des CAI robustes supposent la connaissance des bornes maximales des fonctions non linéaires du système, ce qui n'est pas toujours évident en pratique. De plus, la stabilité est démontrée uniquement dans le contexte entrée bornée-sortie bornée (BIBO stability). Dans ce cas, l'erreur de poursuite est toujours bornée et ne converge pas vers zéro.

Il est à remarquer que la grande majorité des commandes développées dans les travaux suscités ne sont pas validées expérimentalement.

Les travaux de recherche présentés dans cette thèse constituent une continuité des travaux cités précédemment. En effet, nous proposons plusieurs approches de commande CAI afin de surmonter les contraintes classiques de la CAI et améliorer les résultats existant dans la littérature. Ainsi, trois lois de commande sont synthétisées pour les systèmes effectuant des trajectoires répétitives. Deux autres sont développées pour les systèmes effectuant des trajectoires non répétitives.

La présente thèse est structurée en quatre chapitres :

Dans le premier chapitre, la notion de la commande par apprentissage itératif est donnée et suivie par deux exemples simples pour clarifier son idée. Ensuite les hypothèses fondamentales de la CAI sont présentées, et suivies par les avantages et les inconvénients de cette approche. Puis nous donnons un état de l'art sur la commande par apprentissage itératif.

Dans le deuxième chapitre, nous parlons des robots parallèles en général et le robot Delta en particulier, où nous donnons le modèle géométrique direct et inverse ainsi que le modèle dynamique du robot Delta ISIR88 disponible au Laboratoire de Commande des Processus (LCP).

Dans le troisième chapitre, nous proposons trois lois de la CAI à trajectoires répétitives. La première loi est développée sous la condition classique de réinitialisation de l'erreur, où elle utilise le modèle dynamique pour compenser le couplage entre les articulations, ce qui permet d'améliorer les performances, notamment, dans le cas des mouvements à haut cadence. Cette approche est applicable aux robots manipulateurs caractérisés par des modèles non linéaires incertains, contrairement aux approches présentées dans [13–15], où les lois de commandes sont développées pour les systèmes linéaires à temps discret. De plus, cette commande convient aussi bien pour les systèmes à perturbations répétitives que non répétitives, contrairement à celles développées dans [9, 11] où la loi de commande est valable uniquement lorsque la perturbation est répétée. La seconde loi de commande [16] est la commande PD combinée avec la commande CAI type PD que nous désignons dans la suite de cette thèse par PD plus CAI type PD. Cette commande est synthétisée avec la condition d'alignement au lieu de la condition classique de réinitialisation de l'erreur. Une comparaison entre cette approche, qui est partiellement indépendante du modèle dynamique, et celle par couple a priori, qui est basée sur le modèle, a été effectuée en utilisant une trajectoire répétitive. La convergence asymptotique de la commande proposée a été prouvée en utilisant la théorie de Lyapunov. Dans le contexte de l'analyse de la stabilité, les travaux présentés dans [17] concernent l'analyse de la convergence avec un régulateur à paramètres fixes. Dans [18], la même commande a été reprise avec des gains adaptatifs. Dans [9], les auteurs proposent une commande à gain adaptatif avec un terme de robustification. Toutes ces approches considèrent la condition de réinitialisation lors de la démonstration de la stabilité. Notre contribution par rapport à ces travaux réside dans le fait que nous avons utilisé la condition d'alignement. La méthode proposée est validée expérimentalement sur le robot Delta ISIR88 disponible au laboratoire LCP de l'École Nationale Polytechnique. La troisième loi de commande constitue une amélioration de la deuxième méthode, où nous introduisons des gains adaptatifs afin d'accélérer la convergence, ainsi qu'un terme de robustification pour compenser l'effet des perturbations externes. La théorie de Lyapunov est utilisée pour prouver la stabilité asymptotique. Cette commande est applicable aux robots manipulateurs caractérisés par des modèles non linéaires incertains, contrairement aux approches présentées dans [13–15], où les lois de commande sont restreintes aux systèmes linéaires à temps discret. Par rapport aux travaux présentés dans [9, 11, 18, 19], la condition d'alignement est utilisée au lieu de la condition de réinitialisation de l'erreur. En outre, cette commande peut fonctionner aussi bien en présence de perturbations répétitives que non répétitives.

Dans le quatrième chapitre, nous proposons deux lois de CAI pour des applications ayant des trajectoires non répétitives. La première loi de commande peut fonctionner efficacement avec une trajectoire non répétitive à temps d'exécution et à amplitude variables d'un cycle à l'autre, ce qui n'est pas le cas des travaux présentés dans [6, 20–24]. En plus, l'approche proposée convient pour commander les systèmes MIMO non linéaires incertains soumis à des perturbations répétitives et non répétitives et/ou avec erreurs de modélisation. La stabilité asymptotique par rapport aux itérations est démontrée sous la condition d'alignement en utilisant le théorème de Lyapunov. L'efficacité de la commande est validée par simulation. La deuxième et dernière loi de commande développée est fondée sur l'idée et les hypothèses de la loi précédente [25], sauf que la commande est entièrement indépendante du modèle dynamique. Cette loi a été validée expérimentalement sur le robot ISIR88. Les résultats obtenus ouvrent de nouvelles perspectives dans le domaine de la CAI en poursuite de trajectoires non répétitives des systèmes industriels.

sans connaissance a priori sur le modèle dynamique.

Enfin, nous terminons par une conclusion générale englobant les perspectives et les investigations futures à poursuivre.

Chapitre 1

Introduction à la commande par apprentissage itératif

1.1 Introduction

Dans le milieu industriel, un grand nombre de machines se trouvent effectuer des tâches répétitives imposées par le processus de fabrication. Les robots manipulateurs en font partie. Ces derniers doivent exécuter les opérations qui leur sont confiées en répondant à un certain nombre d'exigences technologiques fortement liées à la qualité de la production. La précision dans l'exécution des tâches est un paramètre primordial.

En général, les robots sont dotés de systèmes de commande conventionnels de type PD à paramètres fixes. Dans certains cas, les performances en termes de précision se détériorent avec l'augmentation du nombre de répétition de la tâche. Ceci a motivé les chercheurs pour investiguer dans la conception de lois de commande permettant de remédier à cet inconvénient. Parmi les approches proposées, on trouve la commande par apprentissage itératif. Cette technique de commande permet d'améliorer la précision avec la répétition de la tâche. En effet, lors du cycle présent, la commande actuelle utilise l'erreur du cycle précédent de telle manière à faire diminuer l'erreur de poursuite au fur et mesure que la tâche est répétée.

Dans ce chapitre, nous allons présenter l'idée de base de la commande par apprentissage itératif (CAI) ainsi que les développements théoriques qui lui sont liés.

1.1.1 CAI exemples simples

Pour clarifier l'idée de base de la commande CAI, nous donnons quelques exemples simples [26].

Exemple 1 :

Considérons le système suivant :

$$y(t) = g(t)u(t) \quad (1.1)$$

où, $u(t)$ est la commande, $y(t)$ est la sortie, la fonction $g(t) \neq 0$, pour $t \in [0, T]$. La trajectoire désirée est notée par $y_d(t)$, avec $t \in [0, T]$.

Si $g(t)$ est connue a priori, la loi de commande sera triviale et elle est donnée comme suit :

$$u_d(t) = \frac{y_d(t)}{g(t)}, \quad t \in [0, T] \quad (1.2)$$

Cependant, cette loi commande est en boucle ouverte. Pour cela, le système peut ne pas être performant. De plus, la connaissance exacte de $g(t)$ n'est pas garantie. Pour cela, nous allons considérer $0 < \alpha_1 \leq g(t) \leq \alpha_2$, où α_1 et α_2 sont deux constantes connues.

La loi de commande itérative suivante est donnée pour un système effectuant la même tâche dans un intervalle fini.

$$u_{k+1}(t) = u_k + q\Delta y_k(t), \quad t \in [0, T]. \quad (1.3)$$

avec $k \in \mathbb{Z}_+$ représente le nombre d'itérations, q est le gain d'apprentissage, et $\Delta y_k(t) = y_d(t) - y_k(t)$ est l'erreur de poursuite. Cette loi de commande va être exécutée plusieurs fois afin d'obtenir $u_k = u_d$, ce qui nous donne $y_k = y_d$.

Pour prouver ce résultat, on considère l'erreur de poursuite suivante :

$$\begin{aligned} \Delta y_{k+1} &= y_d - y_{k+1} = y_d - g u_{k+1} = y_d - g(u_k + q\Delta y_k) \\ &= (y_d - g u_k) - qg\Delta y_k = (1 - qg)\Delta y_k \end{aligned} \quad (1.4)$$

par conséquent

$$|\Delta y_{k+1}| \leq |1 - qg| \cdot |\Delta y_k| \quad (1.5)$$

Comme nous avons, $0 < \alpha_1 \leq g(t) \leq \alpha_2 \leq \infty$, alors avec un propre choix de gain d'apprentissage $q\alpha_2^{-1}$ nous pouvons obtenir :

$$0 \leq |1 - qg| \leq \frac{\alpha_2 - \alpha_1}{\alpha_2} = \gamma < 1 \quad (1.6)$$

et

$$\frac{\Delta y_{k+1}}{\Delta y_k} \leq \gamma < 1 \quad \forall k \in \mathbb{Z}_+ \quad (1.7)$$

Ce qui nous donne

$$\lim_{k \rightarrow \infty} |\Delta y_k| \leq \lim_{k \rightarrow \infty} \gamma^{k+1} |\Delta y_0| \rightarrow 0 \quad (1.8)$$

Comme $y_0(t)$ et $y_d(t)$ sont finis pour $t \in [0, T]$, on obtient finalement

$$y_k(t) \rightarrow y_d(t) \text{ pour } k \rightarrow \infty \text{ et } t \in [0, T] \quad (1.9)$$

Exemple 2 :

On considère, maintenant, le système linéaire dynamique suivant :

$$\begin{aligned} \dot{x} &= ax + by \quad x(0) = x_0 \\ y &= cx + du \end{aligned} \quad (1.10)$$

Où, a, b, c , et d sont des paramètres inconnus à temps invariant. La question qui se pose ici, est ce que la loi de commande donnée par (1.3) reste capable de réaliser une poursuite de trajectoire ($\Delta y_k \rightarrow 0$ pour $k \rightarrow \infty$) ?

Avant de répondre à cette question, on suppose que, $x(0) = x_0, \forall k \in \mathbb{Z}_+$. Cette condition suppose que le système démarre toujours de la même position initiale quelque soit l'itération. Le gain d est supposé borné dans un intervalle connu $0 < \alpha_1 \leq d \leq \alpha_2 < \infty$.

Commençant par calculer l'erreur de poursuite de trajectoire Δy_{k+1}

$$\begin{aligned}
 \Delta y_{k+1} &= y_d - y_{k+1} = y_d - cx_{k+1} - du_{k+1} \\
 &= y_d - cx_{k+1} - d(u_k + q\Delta y_k) \\
 &= y_d - (cx_{k+1} + d(u_k)) - qd\Delta y_k - c(x_{k+1} - x_k) \\
 &= (1 - qd)\Delta y_k - c(x_{k+1} - x_k)
 \end{aligned} \tag{1.11}$$

On peut constater qu'il y a un terme supplémentaire $c(x_{k+1} - x_k)$ par rapport à l'exemple précédent. Afin d'évaluer l'effet de ce terme, on définit $\Delta x_k(t) = x_{k+1}(t) - x_k(t)$:

$$\Delta x_k(t) = \Delta x_k(0) + \int_0^t e^{a(t-\tau)} b [u_{k+1}(\tau) - u_k(\tau)] d\tau. \tag{1.12}$$

En utilisant la condition $x(0) = x_0 \forall k \in \mathbb{Z}_+$ et la loi de commande donnée par (1.3), l'équation (1.12) devient :

$$\Delta x_k(t) = \int_0^t e^{a(t-\tau)} qb\Delta y_k(\tau) d\tau. \tag{1.13}$$

Pour une fonction $f(t)$ On définit :

$$|f|_s = \max |f(t)|, \quad t \in [0, T] \tag{1.14}$$

Donc, selon (1.14) l'erreur de poursuite maximale de la k-ème itération est $|\Delta y_k|_s$. Ainsi, la borne maximale de Δx_k est évaluée comme :

$$\begin{aligned}
 |\Delta x_k|_s &\leq \int_0^t e^{a(t-\tau)} |qb| |\Delta y_k(\tau)|_s d\tau \\
 &\leq |qb| |\Delta y_k(\tau)|_s \int_0^T e^{a(t-\tau)} d\tau \\
 &\leq w_k |\Delta y_k(\tau)|_s
 \end{aligned} \tag{1.15}$$

où

$$w_k \triangleq \begin{cases} |qb| \frac{e^{|a|T} - 1}{|a|} & a \neq 0 \\ |qb|T & a = 0 \end{cases} \tag{1.16}$$

Considérant (1.11) et (1.15), on peut écrire :

$$|\Delta y_{k+1}|_s \leq |1 - qd|_s |\Delta y_k|_s + |c|w_k |\Delta y_k|_s \tag{1.17}$$

Comme on peut toujours choisir q tel que $|1 - qd|_s \leq \gamma < 1$, alors il existe une constante positive δ qui satisfait $\gamma + \delta < 1$. Il est évident que si l'intervalle $[0, T_1] \subset [0, T]$, avec T_1 donné par :

$$T_1 \leq \begin{cases} \frac{1}{|a|} \ln \left[1 + \frac{\delta |a|}{|qcb|} \right] & a \neq 0 \\ \frac{\delta |a|}{|qcb|} T & a = 0 \end{cases} \tag{1.18}$$

alors

$$|\Delta y_{k+1}|_s \leq (\gamma + \delta) |\Delta y_k|_s \tag{1.19}$$

Quand on compare le système dynamique de l'exemple 2 avec le système statique de l'exemple 1, on remarque qu'il existe une condition supplémentaire $\Delta x_k(0) = 0$. Cette condition constitue la valeur initiale dans une équation différentielle. Pour cela, elle a un rôle important dans la convergence de l'erreur de poursuite. Nous remarquons aussi que

l'intervalle $[0, T]$ est limité par (1.18), cette limitation n'existe pas dans le système statique.

Nous allons déduire la condition de convergence de la loi de commande. Pour cela la condition de réinitialisation de l'erreur $x_k(0) = x_d(0)$ est considérée. Cette condition suppose que dans chaque itération l'état initial de la trajectoire désirée coïncide avec celui du système.

Supposons qu'il existe une loi de commande u_d que génère y_d et x_d

$$\begin{aligned} \dot{x}_d &= ax_d + by_d, & x_d(0) &= x_0 \\ y_d &= cx_d + du_d \end{aligned} \quad (1.20)$$

L'objectif est de démontrer que la loi de CAI (1.3) assure la convergence de la trajectoire.

on a

$$\begin{aligned} \Delta u_{k+1} &= u_d - u_{k+1} = u_d - u_k - q\Delta y_k \\ &= \Delta u_k - qc\Delta x_k - qd\Delta u_k \\ &= (1 - qd)\Delta u_k - qc\Delta x_k \end{aligned} \quad (1.21)$$

où, $\Delta x_k = x_d - x_k$.

L'erreur dynamique est donnée par

$$\begin{aligned} \Delta \dot{x}_k &= \dot{x}_d - \dot{x}_k = a(x_d - x_k) + b(u_d - u_k) \\ &= a\Delta x_k + b\Delta u_k \end{aligned} \quad (1.22)$$

On tenant compte la condition de réinitialisation de l'erreur et (1.22), on trouve

$$\begin{aligned} \Delta x_k(t) &= \Delta x_k(0) + \int_0^t e^{a(t-\tau)} b\Delta u_k(\tau) d\tau. \\ &= \int_0^t e^{a(t-\tau)} b\Delta u_k(\tau) d\tau. \end{aligned} \quad (1.23)$$

De la même manière de (1.15) on trouve

$$|\Delta x_k|_s \leq \tilde{w}_k |\Delta u_k|_s \quad (1.24)$$

où

$$\tilde{w}_k \triangleq \begin{cases} |b| \frac{e^{|a|T} - 1}{|a|} & a \neq 0 \\ |b|T & a = 0 \end{cases} \quad (1.25)$$

Comme conséquence de (1.21) on obtient

$$|\Delta u_{k+1}|_s \leq |1 - qd|_s |\Delta u_k|_s + |c| \tilde{w}_k |\Delta u_k|_s \quad (1.26)$$

On remarque qu'il y a une analogie entre (1.26) et (1.19). Comme résultat, on aura la même propriété de convergence, où $u_k(t)$ va converger vers $u_d(t)$ si l'intervalle T est choisi par l'équation (1.18).

Si les paramètres $a(t)$, $b(t)$, $c(t)$ et $d(t)$ sont inconnus à temps variant et si $|a|_s$, $|b|_s$, $|c|_s$ et $|d|_s$ sont finis, la seule différence entre le système à temps invariant et à temps variant sera l'intervalle de convergence T_1 .

$$T_1 \leq \begin{cases} \frac{1}{|a|_s} \ln \left[1 + \frac{\delta |a|_s}{|q| |c|_s |b|_s} \right] & a \neq 0 \\ \frac{\delta |a|}{|q| |c|_s |b|_s} T & a = 0 \end{cases} \quad (1.27)$$

D'après l'exemple 1 et 2, nous pouvons constater que la CAI permet la convergence

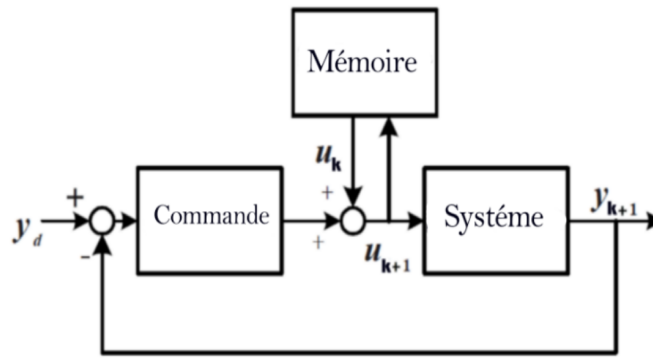


Figure 1.1 – Schéma simple de la commande CAI

de l'erreur avec l'augmentation du nombre d'itérations k . La synthèse de cette loi de commande ne nécessite pas la connaissance du modèle dynamique du système. Ceci constitue un avantage fondamental de cette commande.

La figure 1.1 illustre un schéma simple de la commande CAI.

1.2 Définitions

1.2.1 Commande par apprentissage itératif en boucle ouverte

Une loi de commande par apprentissage itératif en boucle ouverte est donnée par

$$u_{k+1} = f(u_k, e_k, t), \quad k = 1, 2, 3, \dots \quad (1.28)$$

1.2.2 Commande par apprentissage itératif en boucle fermée

Une loi de commande par apprentissage itératif en boucle fermée est donnée par

$$u_{k+1} = f(u_k, e_{k+1}, t), \quad k = 1, 2, 3, \dots \quad (1.29)$$

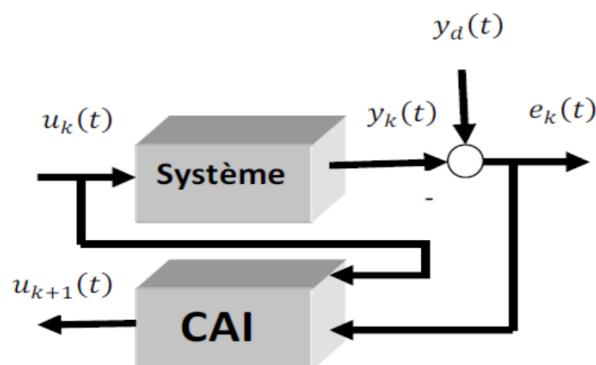


Figure 1.2 – Schéma de la CAI en boucle ouverte

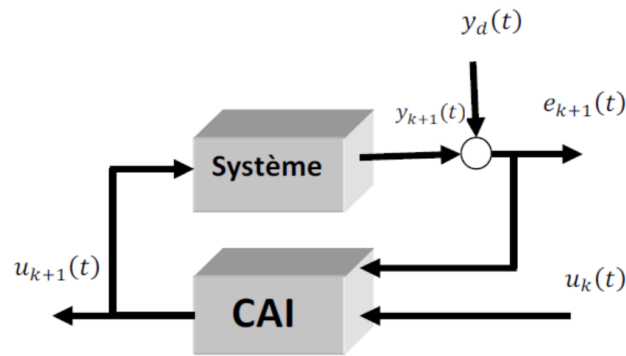


Figure 1.3 – Schéma de la CAI en boucle fermée

1.3 Hypothèses fondamentales de la commande CAI

La théorie de la CAI est basée sur deux hypothèses fondamentales, à savoir la condition de réinitialisation de l'erreur et la répétition de la trajectoire désirée (voir [27, 28]).

1- Hypothèse de réinitialisation de l'erreur : Cette hypothèse suppose que, dans chaque itération, l'état initial de la trajectoire désirée doit être égal à celui du système. Cette condition est utilisée dans plusieurs travaux [2, 3, 9, 18]. D'autres résultats ont été présentés pour remplacer la condition de réinitialisation de l'erreur par d'autres conditions. Parmi ces conditions, on peut trouver la condition de l'erreur fixe qui suppose que l'erreur de poursuite est constante à chaque itération [29]. Il y a également la condition d'alignement de l'erreur (alignement condition) qui considère l'état final de l'itération précédente comme étant l'état initial de l'itération actuelle [30]. Une autre condition considère l'initialisation de l'erreur par une valeur aléatoire et bornée [7].

Une des conditions suivantes est utilisée afin de représenter le problème de la condition initial de chaque itération [29] :

- $e_k(0) = 0$.
- $\sum_{i=1}^{\infty} e_k^2(0) = C$, où, C est une constante.
- $|e_k(0)| = C \neq 0$, où, C est une constante.
- $e_k(0)$ est aléatoire et bornée.
- $e_k(0) = e_{k-1}(T)$.

2- Hypothèse de la trajectoire répétitive : L'hypothèse de la trajectoire répétitive considère que le système effectue une tâche répétitive selon la même trajectoire définie sur un intervalle de temps fixe. Récemment, l'hypothèse de la trajectoire répétitive a été remplacée dans quelques travaux par une hypothèse moins contraignante qui considère la trajectoire non répétitive [6, 20, 21, 23, 24].

1.4 Avantages et inconvénients de la CAI

Les principaux avantages et inconvénients de la commande CAI se résument par les points suivants :

Avantages :

- La CAI est développée pour réaliser une poursuite de trajectoire et améliorer les performances d’une itération à l’autre contrairement à la plupart des commandes classiques.
- Elle possède une structure simple.
- Elle nécessite peu de connaissance sur le modèle dynamique. En générale, seules certaines bornes sont nécessaires.

Inconvénients :

- La condition de réinitialisation de l’erreur utilisée dans plusieurs lois de commande peut être très difficile à satisfaire en pratique.
- La trajectoire désirée doit être répétitive sur un intervalle de temps fini et fixe pour chaque itération.

1.5 État de l’art sur la commande par apprentissage itératif

L’idée de base de la commande par apprentissage itératif a été publiée en 1978 au Japon par Uchiyama [31]. Sachant que cet article était rédigé en Japonais, le travail n’a pas été exploité par les chercheurs et par conséquent n’a pas connu de continuité. Quelques années après, la commande par apprentissage itératif avait été introduite pour la première fois d’une manière explicite par Arimoto en 1984 [2], où il avait proposé une CAI classique de type PD et PID. Dans cette publication il démontrait que l’erreur de poursuite peut converger en utilisant uniquement les erreurs des itérations précédentes. En se basant sur ce résultat intéressant, beaucoup de chercheurs s’étaient penchés sur la question afin d’apporter les améliorations nécessaires [5, 8, 15, 32–34].

Dans le développement de la CAI, on peut trouver quatre catégories principales : la CAI classique, la CAI adaptative, la CAI robuste et la CAI basée-modèle.

1.5.1 CAI classique

Les premières lois de commande CAI ont été développés par Arimoto dans 1984 [2], où il est proposé la CAI pour commander un robot manipulateur série [3].

En général, le modèle d’un système dynamique lors d’un cycle d’exécution k , est mis sous la forme suivante :

$$\dot{x}_k(t) = f(x_k(t)) + g(x_k(t))u_k(t), \quad t \in [0, T] \quad (1.30)$$

où x_k est l’état, $f(x_k)$ est une fonction non linéaire, $g(x_k)$ est le gain d’entrée et u_k est la commande.

Les trois lois de commande CAI développées par Arimoto sont exprimées par les expressions suivantes :

- **CAI type PD :**

$$u_k(t) = u_{k-1}(t) + \Gamma e_k(t) + \Lambda \dot{e}_k(t) \quad (1.31)$$

où, $e_k = x_r - x_k \in \mathbb{R}^n$ est l'erreur de poursuite, x_r est la trajectoire désirée, \dot{e}_k est la dérivée de l'erreur de poursuite par rapport au temps, tandis que, Λ et Γ sont des matrices constantes diagonales définies positives.

– **CAI type D :**

$$u_k(t) = u_{k-1}(t) + \Lambda \dot{e}_k(t) \quad (1.32)$$

– **CAI type PID :**

$$u_k(t) = u_{k-1}(t) + \Gamma e_k(t) + \Lambda \dot{e}_k(t) + \psi \int_0^t e(\tau) d\tau \quad (1.33)$$

où, ψ est une matrice diagonale définie positive.

L'avantage principal de ces CAI classiques réside dans le fait que leur développement est indépendant du modèle. Elle utilise uniquement les erreurs passées de poursuite pour améliorer les erreurs actuelles. Cependant, ces lois ne sont pas développées en présence des perturbation externes. Ces perturbations et les variations paramétriques peuvent dégrader significativement les performances.

1.5.2 CAI adaptative

Dans l'objectif d'améliorer l'approche proposée en termes de rejet de perturbations et de poursuite de trajectoires, plusieurs travaux avaient été publiés. Dans [35] une commande CAI combinée avec un régulateur à structure variable avait été synthétisée en se basant sur la théorie de Lyapunov. Dans [5], Tayebi avait présenté une CAI adaptative, où la loi de commande est destinée pour résoudre le problème de poursuite de trajectoire des robots manipulateurs. L'efficacité de la commande CAI adaptative avait été montrée aussi bien dans le travail de Tayebi que dans d'autres travaux tels que, [5–9, 18, 20], où il a été prouvé que l'erreur de poursuite converge avec l'augmentation des itérations.

La majorité des CAI adaptatives sont développées pour la classe des robots données par le modèle suivant :

$$M(q_k)\ddot{x}_k + C(x_k, \dot{x}_k)\dot{x}_k + G(x_k) = u_k + d_k(t) \quad (1.34)$$

où, M est la matrice d'inertie, C est la matrice des forces de Coriolis et centrifuges, G est le vecteur des forces de gravitation et d représente les perturbations externes.

La loi de CAI adaptative est donnée par [5] :

$$u_k(t) = K_p e_k(t) + K_d(t) \dot{e}_k(t) + \phi(x_k, \dot{x}_k, \ddot{x}_k) \hat{\theta}_k(t) \quad (1.35)$$

avec

$$\hat{\theta}_k(t) = \hat{\theta}_{k-1}(t) + \Gamma \phi(x_k, \dot{x}_k, \ddot{x}_k) \dot{e}_k(t) \quad (1.36)$$

où le terme initial $\hat{\theta}_0(t) = 0$, la matrice $\phi = [\Psi(x_k, \dot{x}_k), \text{sgn}(\Psi)]$, avec Ψ satisfaisant la relation suivante $C(x_k, \dot{x}_k)\dot{x}_d + G(x_k) = \Psi(x_k, \dot{x}_k)\zeta(t)$, tel que Ψ est connue et le ζ est inconnu.

En général, la CAI adaptative est restrictive à une classe particulière de modèles caractérisés par une fonction linéaire à paramètres inconnus combinée avec une fonction non linéaire connue. Il est à noter que ce type de représentation n'est pas toujours évident, notamment pour les systèmes à structures complexes.

1.5.3 CAI robuste

Généralement ce type de commandes est proposé pour les systèmes linéaires à temps discret [13, 14, 36, 37]. On peut citer le travail de Meng [13], où les auteurs ont développé une CAI de type proportionnel. Dans cette publication le gain d'apprentissage varie en fonction des itérations pour garantir la stabilité et la robustesse de la commande. Une commande similaire a été proposée dans [14], où une étude plus approfondie concernant l'influence du gain d'apprentissage sur les performances de convergence est présentée.

La CAI robuste est développée pour la classe de systèmes suivants :

$$\begin{aligned}x_k(l+1) &= A_k(l)x_k(l) + B_k(l)u_k(l) + w_k(l) \\y_k(l) &= C_k(l)x_k(l) + v_k(l)\end{aligned}\tag{1.37}$$

La loi de CAI robuste est donnée par [5] :

$$u_{k+1}(l) = u_l + \Gamma_k(l)[r_k(l+1) - y_k(l+1)], l \in N-1\tag{1.38}$$

Où, $\Gamma_k(l)$ est une matrice variable au cours des itérations.

La majorité des CAI robustes supposent la connaissance des bornes maximales des fonctions non linéaires du système, ce qui n'est pas toujours évident en pratique. De plus, la stabilité est démontrée uniquement dans le contexte entrée bornée-sortie bornée (BIBO stability). Dans ce cas, l'erreur de poursuite est toujours bornée et ne converge pas vers zéro.

1.5.4 CAI basée-modèle

Comme la plupart des systèmes sont modélisables, la CAI basées-modèle dynamique utilise cette connaissance du modèle pour générer la commande. Elle a été introduite dans quelques application afin d'améliorer les performances de poursuite [11, 12, 19, 25]. Cependant, l'efficacité de la commande dépend du degré de connaissance du modèle dynamique, où une mauvaise modélisation pourra engendrer la dégradation les performances.

1.5.5 Application pratique de la CAI

La CAI a été implantée avec succès dans plusieurs domaines industriels, comme les robots manipulateurs rigides [15, 18, 38, 39], les robot manipulateurs flexibles [40] les processus chimiques [41], réglage de la température [42], procédé de soudage [43], moteur piézoélectrique [44], servo hydraulique [45] et le découpage au laser [46]. Il est à noter que la CAI est également utilisée pour ajuster les paramètres des régulateurs PID [47].

1.5.6 Rappels mathématiques

La norme euclidienne d'un vecteur x est définie par :

$$\|x\| = \sqrt{x^T x}\tag{1.39}$$

La norme d'une matrice A est définie par :

$$\|A\| = \sqrt{\lambda_{\max}(A^T A)} \quad (1.40)$$

avec $\lambda_{\max}(\cdot)$ est la plus grande valeur propre de (\cdot) .
L'inégalité de Schwartz est donnée comme suit :

$$|x^T y| \leq \|x\| \|y\|, \text{ pour tout } x, y \in \mathbb{R}^n \quad (1.41)$$

Les valeurs propres d'une matrice $A \in \mathbb{R}^{3 \times 3}$ sont déterminées par :

$$\det[\lambda_i \{A\} I - A] = 0 \text{ pour } i = 1, 2, \dots, n. \quad (1.42)$$

Le théorème de Rayleigh–Ritz est établi que pour toutes $x \in \mathbb{R}^3$:

$$\lambda_{\min}\{A\} \|x\|^2 \leq x^T A x \leq \lambda_{\max}\{A\} \|x\|^2 \quad (1.43)$$

1.6 Conclusion

Dans ce chapitre, nous avons présenté l'idée de base de la commande par apprentissage itératif. Deux exemples simples ont été introduits pour montrer l'avantage fondamental de la CAI qui réside dans sa capacité d'améliorer les performances sans avoir besoin d'utiliser directement le modèle dynamique de système. Il a été montré que la CAI est très appropriée pour les processus industriels grâce à sa simple structure.

Les hypothèses fondamentales utilisés dans le développement de la CAI ainsi ses avantages et les inconvénients ont été discutés. Une revue générale sur cette technique de commande a été présentée à travers les résultats consultés dans la littérature.

Chapitre 2

Robot Parallèle du Type Delta

2.1 Généralités sur les robots parallèles

2.1.1 Définition

Les architectures parallèles sont apparues il y a environ 50 ans avec les premiers simulateurs de vol. Elles sont utilisées depuis dans d'autres applications nécessitant la manipulation d'objets lourds avec de grandes accélérations ou l'assemblage dans les applications qui requièrent une grande précision et la maîtrise des efforts de contact ou encore la prise-dépose d'objets à grande cadence pour les applications d'emballage. Le concept a été repris récemment dans le domaine de la machine-outil pour des applications d'usinage à grande vitesse et à haute précision.

Un robot parallèle est un mécanisme en chaîne cinématique fermée figure 2.1, constitué d'un organe terminal mobile et d'une base fixe, relié entre eux par des chaînes cinématiques indépendantes. Chacune de ces chaînes compte au plus deux segments articulés afin que la structure reste commandable. La motorisation s'effectue avec des actionneurs simples, un pour chaque chaîne, généralement fixés sur la base.

2.1.2 Historique

L'historique des robots parallèles est controversé. Cependant, grâce aux travaux de Merlet et de Bonev [48], il existe une chronologie assez précise de l'apparition des robots parallèles. Tout s'abord, Merlet indique que la théorie sur les mécanismes parallèles a été explorée bien avant l'apparition du terme robot. Certains problèmes théoriques concernant les robots parallèles ont donc été résolus bien avant leur apparition. Ensuite, Bonev rappelle que le premier brevet déposé pour un mécanisme parallèle porte sur un mécanisme sphérique destiné à être utilisé comme plateforme de cinéma dynamique en 1928. Trop en avance sur son temps, cette machine n'a jamais été construite. Quelques

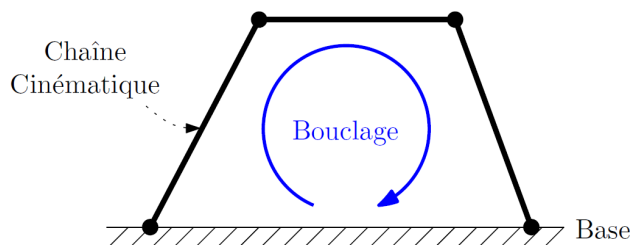


Figure 2.1 – Chaîne cinématique fermée

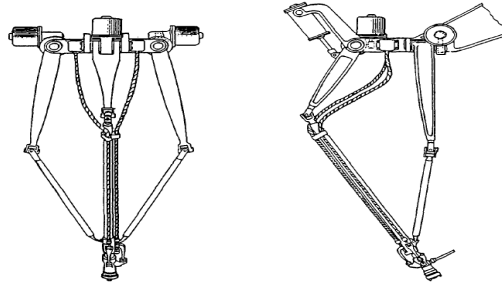


Figure 2.2 – Mécanisme de Pollard



Figure 2.3 – Simulateur de Vol : application de la Plateforme de Stewart

dizaines d'années plus tard, Williard L.V. Pollard [49] conçoit ce qui est considéré comme le premier robot parallèle industriel. Son fils fait breveter cette invention qui n'a, elle non plus, jamais été produite.

Finalement, le premier robot parallèle industriel ayant vu le jour est l'hexapode en forme octaèdre inventé par Eric Gough, également appelé plateforme de Gough [50]. Cet hexapode a été construit afin de tester des pneus pour la société Dunlop et le premier exemplaire a été produit en 1954. Pour beaucoup de chercheurs, l'invention de la plateforme de Gough marque le début des robots parallèles. L'autre événement majeur de la robotique parallèle est l'invention du premier robot parallèle léger, le robot Delta, par le professeur Clavel dans les années 80. Ce robot, qui est à ce jour le robot parallèle le plus vendu dans le monde, marque l'arrivée des robots parallèles rapides destinés, entre autres, à des tâches de conditionnement. Le robot Par4, est un parfait exemple de l'engouement de l'industrie pour les robots parallèles rapides. En effet, une version commerciale du robot Par4, le robot Quattro développé par Adept, a ensuite vu le jour et est actuellement le robot le plus rapide du marché (240 cycles par minute). Il existe un grand nombre de cinématiques différentes de robots parallèles et leurs applications possibles sont nombreuses : simulateur de vol, machine-outil, robot de prise et dépose, positionneur de haute précision, robot médical, capteurs d'effort, etc. Aujourd'hui encore, de nouveaux domaines s'ouvrent pour ces robots notamment grâce à la recherche sur les robots redondants, les robots à câbles ou les robots ultra rapides.

2.1.3 Caractéristiques des robots parallèles

Outre la fermeture de la chaîne cinématique, les robots parallèles présentent d'autres caractéristiques particulières. Plusieurs aspects qui différencient les robots parallèles et les robots sériés classiques peuvent être cités. Ces aspects permettent de mettre en évidence les caractéristiques des structures parallèles, leurs avantages et leurs inconvénients.



Figure 2.4 – FlexPicker de ABB

Les avantages :

– **Rigidité** : Un robot parallèle est un robot à chaîne cinématique bouclée. Cela lui confère une plus grande rigidité de positionnement, permettant ainsi d'améliorer la précision par rapport aux robots séries.

– **Rapport masse transportable/masse du robot** : Dans l'architecture série classique, l'organe terminal se trouve à l'extrémité de la chaîne articulée. Chaque actionneur doit avoir la puissance nécessaire pour mettre en mouvement tous les corps et actionneurs en aval. Dans les cas des robots parallèles, les actionneurs se situent le plus souvent au voisinage des points d'articulation de la base, si ce n'est sur la base elle-même, ce qui donne un rapport masse transportable/masse du robot beaucoup plus élevé.

– **Précision et répétabilité** : Pour les robots série une erreur de positionnement sur une ou plusieurs articulations a une répercussion sur la position finale de l'organe terminal. Ce problème d'accumulation des erreurs ne se présente pas pour les robots parallèles puisque leur morphologie leur assure une rigidité remarquable et une moindre dépendance des articulations en termes d'erreurs de positionnement

– **Comportement dynamique** : Compte tenu de leur structure, les robots série ont des performances dynamiques limitées. Avec les robots parallèles, le fait de diminuer l'inertie du robot contribue à l'augmentation de la rapidité et de la cadence de travail.

– **Technologie de construction** : L'un des avantages des manipulateurs parallèles est l'utilisation fréquente de plusieurs chaînes cinématiques identiques sur un robot unique (rentabilité, pièces de rechanges, . . .). Un autre avantage de certains manipulateurs parallèles (avec actionneurs fixe) est la plus grande liberté dans le choix des actionneurs : rotoïde, prismatiques, à entraînement direct ou indirect, à moteur électrique ou non électrique. De plus cela a pour effet de faciliter l'entretien et la réparation des moteurs. Le branchement de l'instrumentation (capteurs) et de l'alimentation est aussi plus aisé et enfin, la détérioration des fils de branchement est moindre pour les robots parallèles car on n'est pas contraint de les faire passer à travers toute la chaîne cinématique comme c'est le cas pour les robots séries.

Les inconvénients :

- Modèles géométriques directs (MGD) parfois difficiles à déterminer.
- Volume de travail limité en regard du volume total du mécanisme.
- Fort couplage entre le mouvement des différentes chaînes cinématiques.
- Présence de singularités qui conduisent à une perte de contrôle de la structure mobile.



Figure 2.5 – Robot Delta de l'EPFL

2.2 Robot Delta

2.2.1 Historique

Depuis le début des années 1980, des centaines de nouveaux modèles ont été proposés, et presque autant de demandes ont été déposées. Beaucoup de ces architectures parallèles sont véritablement novatrices, mais une seule a permis de développer le robot parallèle le plus réussi pour les applications industrielles : le robot Delta. Au cours de l'été 1985, le professeur Reymond Clavel de l'EPFL en Suisse, a eu l'idée d'utiliser des parallélogrammes pour construire un robot parallèle à 4 degré de liberté pour les opérations de transfert rapide [51]. Il a demandé une série de brevets, sur la base desquels plusieurs compagnies, dont ABB, ont obtenu des licences. Pendant la durée de vie de ses principaux brevets, plus de 10 000 unités ont été fabriquées. Aujourd'hui, le design du robot Delta n'est plus protégé et des dizaines de sociétés proposent leurs propres versions, y compris FANUC, Motoman, et Kawasaki.

2.2.2 Spécifications du Robot Delta

Le robot Delta possède les caractéristiques suivantes :

- Le Delta est un robot parallèle, ceci veut dire que la liaison entre sa base d'amarrage et son organe terminal se fait par le biais de plusieurs chaînes cinématiques (trois chaînes cinématiques identiques pour le Delta conventionnel), diminuant ainsi l'espace de travail et augmentant la rigidité et la précision du robot.
- La nacelle du robot (structure portant l'organe terminal) possède la propriété de garder constamment une configuration horizontale et ce d'une manière totalement passive, c'est-à-dire sans aucun asservissement ou contrôle de l'orientation de la nacelle (propriété obtenue par construction mécanique du robot).
- Les moteurs chargés d'actionner le robot sont fixés au niveau de la base d'amarrage du robot, ce qui a pour effet de diminuer considérablement les masses mobiles du robot et donc leurs inerties, permettant ainsi de grandes vitesses et accélérations dans l'utilisation de ce robot.
- Les barres parallèles ont constitué d'un parallélogramme. C'est cette particularité qui permet à la nacelle du robot Delta d'avoir des mouvements translationnels seulement.

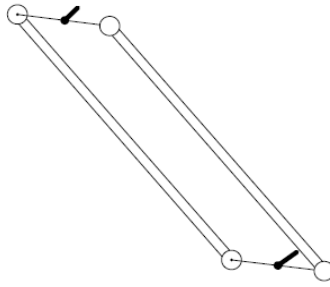


Figure 2.6 – Barres parallèles du robot Delta

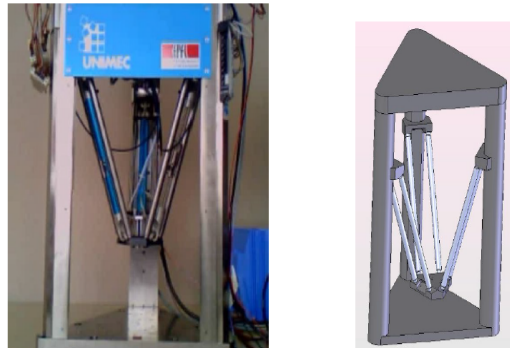


Figure 2.7 – Delta linéaire

2.2.3 Types de robot Delta

Le robot Delta est un robot à trois (quatre) degrés de liberté tel qu'il possédant trois (quatre) chaînes cinématiques indépendantes et identiques. La caractéristique essentielle qui est identique pour tous les robots Delta est l'utilisation de barres parallèles. Ces barres sont fixées entre elles à une petite barre de maintien par des liaisons rotules (voir figure 2.6). C'est cette particularité qui permet à la nacelle de rester parallèle à la base et d'avoir ainsi des mouvements translationnels seulement. Ces chaînes peuvent être actionnées de différentes façons. Le but de l'actionnement est de transmettre une force de traction/poussée sur les trois couples de barres parallèles. Cela peut être obtenu de différentes façons. Il existe deux types particuliers de robot Delta selon l'actionneur utilisé, le robot Delta linéaire dont les trois actionneurs des trois chaînes cinématiques sont de types prismatiques et le robot Delta à actionnement rotatif dont les actionneurs sont rotoïdes.

Delta linéaire

Ce type de robot Delta utilise trois actionneurs prismatiques, ces actionneurs sont fixés directement aux barres parallèles car ce type d'actionneur produit directement une force et peut donc transmettre une force de traction/poussée aux barres parallèles. Ces actionneurs prismatiques peuvent être du type pneumatique ou hydraulique (vérins), ou alors de type moteurs électriques couplés à une vis sans fin, une crémaillère ou une courroie. La figure 2.7 montre une telle structure. Ce type de robot Delta est cependant peu utilisé à cause des actionneurs prismatiques qui sont encombrants pour le cas des systèmes moteurs/crémaillères, et contraignant pour le cas des vérins.



Figure 2.8 – Delta à actionnement rotatif

Delta à actionnement rotatif

Ce type de robot Delta utilise trois actionneurs rotoïdes, ces actionneurs ne peuvent pas être fixés directement aux barres parallèles comme c'est le cas dans le Delta linéaire, car ces actionneurs produisent un couple et donc pour transmettre une force de traction/poussée aux barres parallèles on doit prévoir un bras couplé à l'axe de l'actionneur. Cet avant-bras est fixé à l'arbre du moteur d'un côté et à la barre de fixation des barres parallèles de l'autre (voir indication sur la figure 2.8). Cela permet d'avoir une configuration équivalente au cas du robot Delta linéaire [52].

2.3 Description du robot Delta ISIR88 disponible au laboratoire de commande des processus (LCP)

Ce robot parallèle à 4 degrés de liberté (x, y, z et φ), spécialement dédié à la manipulation de pièces légères, peut atteindre une cadence de travail deux à trois fois supérieure à celle des robots les plus rapides actuellement sur le marché à titre d'exemple, il peut assurer le transfert de 3 pièces par seconde par 3 mouvements aller et retour complets.

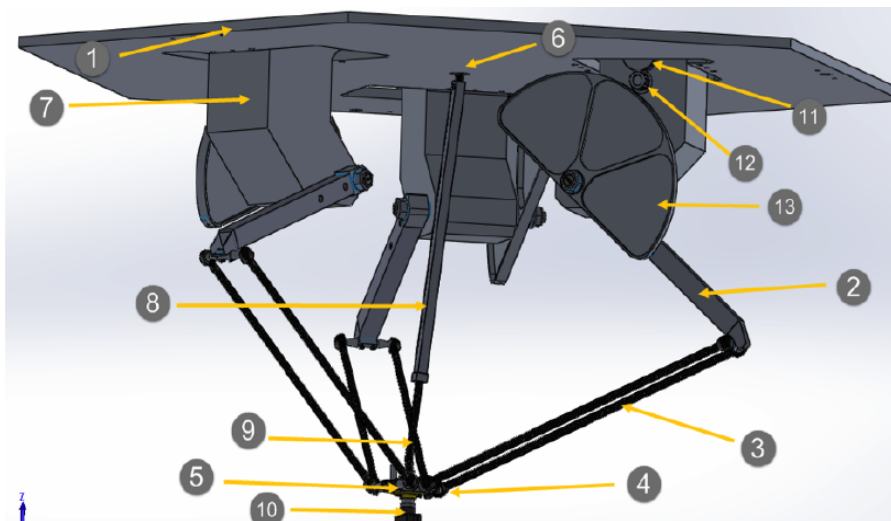


Figure 2.9 – Schéma mécanique du robot Delta ISIR88 disponible au LCP

Table 2.1 – Paramètres physiques du robot

paramètre	Valeur
Longueur du bras	0.205 m
Longueur de l'avant-bras	0.380 m
Masse de la nacelle plate	0.042 kg
Masse de bras	0.098 kg
Masse de barres parallèles	0.028 kg
Masse de coude	0.015 kg

La figure 2.9 nous servira pour la description du robot Delta-4 et de son fonctionnement. Ce robot a 4 degrés de liberté et il est principalement constitué d'une "base fixe" (1) solidaire d'un bâti support de l'installation (non représenté) et d'une plaque mobile (S), le nom donné à cette dernière pièce est "nacelle". La liaison entre la base fixe (1) et la nacelle (5) est assurée par 3 chaînes cinématiques, chacune d'elles est formée d'un "bras" (2) monté en articulation pivot sur la base fixe et de 2 "barres parallèles (avant-bras)" (3) munies chacune d'une articulation (4) à chaque extrémité l'ensemble formé de 2 barres parallèles et des 2 éléments de liaison au bras et à la nacelle sera nommé "parallélogramme". Chaque bras (2) est entraîné par un "moteur de bras" (7) qui aura le plus souvent la forme d'un ensemble moteur-réducteur-capteur. Le "préhenseur" (10) sera monté en pivotement sur la nacelle et entraîné par le moteur (6) par l'intermédiaire de l'arbre télescopique" (8) muni d'une articulation de type cardan (9) à chacune de ses extrémités. L'orientation de la nacelle est constamment assurée par les 3 parallélogrammes comportants chacun 2 petits têtes et 2 grands côtés formés par les barres parallèles, chaque petit côté solidaire de l'extrémité d'un bras reste constamment parallèle à l'axe de rotation du bras considéré les 3 paires de barres parallèles ne garantissent que les 3 petits côtés solidaires de la nacelle restent parallèles à ceux solidaires de l'extrémité des bras et donc parallèles aux axes de rotation des bras qui, par construction, sont situés dans un même plan. Les articulations aux extrémités des barres parallèles sont de type rotule. Chaque barre peut donc tourner autour de son axe longitudinal. Cette rotation ne perturbe pas le comportement de cette structure articulée formant le parallélogramme de l'espace. Une liaison par des ressorts et des étriers entre les 2 barres parallèles permet de simplifier la construction des rotules et limite les ébats en rotation des barres parallèles. La transmission entre l'un des moteurs et le bras correspondant n'est pas direct, en effet la transmission est assurée par un système de réduction constitué par une poulie (11), pignon (12) et une roue (13). afin d'obtenir un rapport de réduction égale 12 :1. Les valeurs des paramètres physiques du robot sont données par le tableau 2.1 [53].

2.4 Modélisation du Robot Delta

2.4.1 Modèle géométrique direct

Le modèle géométrique direct (MGD) du robot Delta permet d'exprimer la position translationnelle (x, y, z) de sa nacelle en fonction des trois angles articulaires $(\alpha_1, \alpha_2, \alpha_3)$ (figure 2.10).

Le calcul du MGD d'une structure parallèle telle que le robot Delta n'est pas systématique comme c'est le cas pour les structures série (transformations homogènes avec formulation de DH ou Dombre).

La symétrie ternaire que présente le Delta nous permet de travailler sur un tiers du

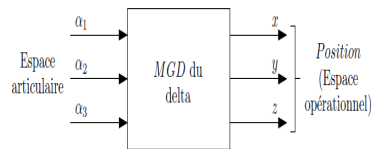


Figure 2.10 – Modèle géométrique direct du robot Delta

robot, c'est à dire un seul bras figure 2.11.

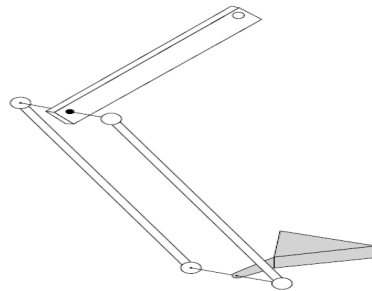


Figure 2.11 – Un bras du Delta

Un repère R_i est lié à chaque bras du Delta comme l'indique la figure 2.12 qui représente la vue de dessus de la base. Le travail se fera ensuite une façon indépendante sur chaque repère.

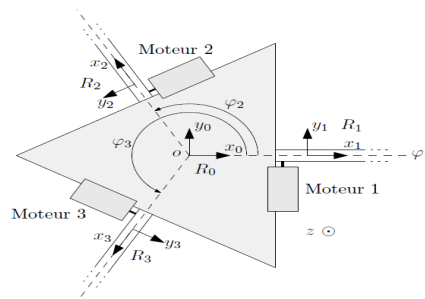


Figure 2.12 – Vue du dessus de la base du Delta

L'objectif est de trouver une relation entre les angles articulaires α_i correspondants à R_i et la position de la nacelle $(x, y, z)_{R_0}$ dans un sens ou dans l'autre (modèle direct ou inverse).

Le passage entre le repère absolu R_0 et les repères intermédiaires se fera par le biais des deux relations matricielles suivantes :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{R_0} = \begin{pmatrix} \cos \varphi_i & -\sin \varphi_i & 0 \\ \sin \varphi_i & \cos \varphi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_{R_i}$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{R_i} = \begin{pmatrix} \cos \varphi_i & \sin \varphi_i & 0 \\ -\sin \varphi_i & \cos \varphi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_{R_0}$$

Les valeurs prises par les φ_i sont $\varphi_1 = 0$, $\varphi_2 = \frac{2\pi}{3}$ et $\varphi_3 = \frac{4\pi}{3}$.

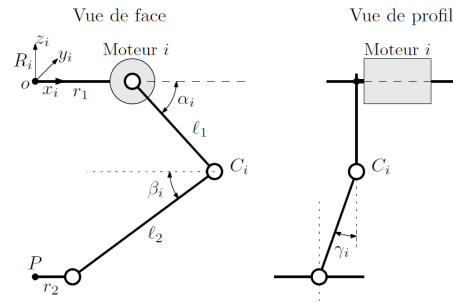


Figure 2.13 – Schéma d'un bras du Delta

La figure 2.13 donne le schéma annoté d'un bras du Delta en considérant les deux barres parallèles comme une seule (ce qui ne gêne pas pour le développement du modèle géométrique du moment que la nature des articulations est laissée comme telle).

Étant donné que la nacelle reste toujours horizontale et donc parallèle à la base (mouvement translationnel seulement), on peut la considérer comme étant un point matériel (l'orientation n'a pas d'importance puisque de toute façon elle n'est pas possible), situé au niveau du centre de gravité de la nacelle (c'est l'endroit où est placé l'organe terminal). Le schéma simplifié obtenu est présenté sur la figure 2.14.

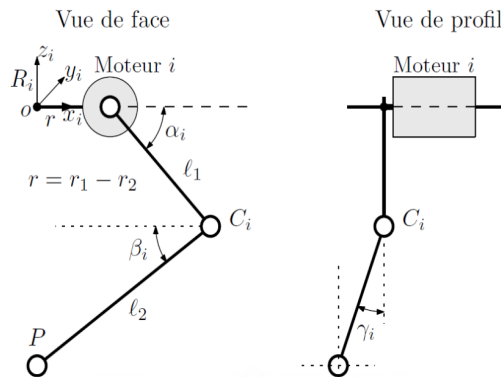


Figure 2.14 – Schéma simplifié d'un bras du Delta

Les points C_i vont servir de points intermédiaires dans le développement des modèles géométriques direct et inverse, ils ont comme coordonnées les quantités suivantes :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{C_i/R_i} = \begin{pmatrix} r + l_1 \cos \alpha_i \\ 0 \\ -l_1 \sin \alpha_i \end{pmatrix}_{C_i/R_i} \quad (2.1)$$

avec $i = 1, 2, 3$. Les points C_i appartiennent à des cercles de rayon l_1 centrés sur l'axe du moteur.

Des équations 2.1 en appliquant la rotation de repère :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{C_i/R_0} = \begin{pmatrix} \cos \varphi_i & -\sin \varphi_i & 0 \\ \sin \varphi_i & \cos \varphi_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r + l_1 \cos \alpha_i \\ 0 \\ -l_1 \sin \alpha_i \end{pmatrix} \quad (2.2)$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{C_i/R_0} = \begin{pmatrix} (r + l_1 \cos \alpha_i) \cos \varphi_1 \\ (r + l_1 \cos \alpha_i) \sin \varphi_1 \\ -l_1 \sin \alpha_i \end{pmatrix} \quad (2.3)$$

La jonction au niveau du point P étant une liaison rotule, le point C_i appartient à une sphère de rayon l_2 et de centre $P(x, y, z)_{R_0}$ (voir figure 2.15). Cette sphère est définie dans le repère absolu R_0 par l'équation :

$$(X - x)^2 + (Y - y)^2 + (Z - z)^2 = l_2^2 \quad (2.4)$$

où, X, Y et Z sont les coordonnées d'un point appartenant à la sphère et x, y et z sont les

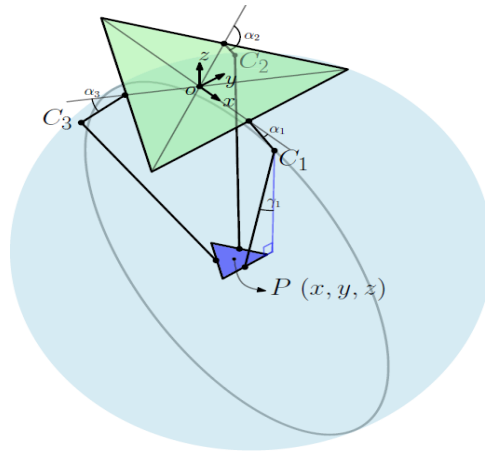


Figure 2.15 – Sphère de centre P et de rayon l_2 passant par C_1

coordonnées du centre de cette sphère (point P).

Les équations des cercles passant par C_i de rayon l_1 centrés sur l'axe des moteurs combinés aux équations de sphère centrée en P de rayon l_2 passant par les C_i permettent de trouver directement le MGD (les points C_i sont à l'intersection de la sphère et des cercles). La méthode présentée est celle développée par Clavel dans [54].

En remplaçant l'équation (2.3) dans l'équation de la sphère (2.4) on trouve :

$$\begin{aligned} x^2 - 2x(r + l_1 \cos \alpha_i) \cos \varphi_i + (r + l_1 \cos \alpha_i)^2 \cos^2 \varphi_i + \\ y^2 - 2y(r + l_1 \cos \alpha_i) \sin \varphi_i + (r + l_1 \cos \alpha_i)^2 \sin^2 \varphi_i + \\ z^2 + 2zl_1 \sin \alpha_i + l_1^2 \sin^2 \alpha_i = l_2^2 \end{aligned} \quad (2.5)$$

Ce qui donne :

$$\begin{aligned} x^2 - 2x(r + l_1 \cos \alpha_i) \cos \varphi_i + y^2 - 2y(r + l_1 \cos \alpha_i) \sin \varphi_i + \\ z^2 + 2zl_1 \sin \alpha_i = l_2^2 - l_1^2 - r^2 - 2rl_1 \cos \alpha_i \quad \text{avec } i = 1, 2, 3. \end{aligned} \quad (2.6)$$

En posant :

$$\begin{aligned} D_i &= -l_2^2 + l_1^2 + r^2 + 2rl_1 \cos \alpha_i \\ E_i &= 2(r + l_1 \cos \alpha_i) \cos \varphi_i \\ F_i &= 2(r + l_1 \cos \alpha_i) \sin \varphi_i \\ G_i &= -2l_1 \sin \alpha_i \end{aligned} \quad (2.7)$$

Donc nous avons 3 équations à 3 inconnues à résoudre :

$$x^2 - E_1 x + y^2 - F_1 y + z^2 - G_1 z = -D_1 \quad (2.8)$$

$$x^2 - E_2x + y^2 - F_2y + z^2 - G_2z = -D_2 \quad (2.9)$$

$$x^2 - E_3x + y^2 - F_3y + z^2 - G_3z = -D_3 \quad (2.10)$$

La résolution de ces équations en (x, y, z) nous donne le modèle géométrique direct.

En effectuant :

$$(E_2 - E_1)x + (F_2 - F_1)y + (G_2 - G_1)z = D_2 - D_1 \quad (2.11)$$

$$(E_3 - E_2)x + (F_3 - F_2)y + (G_3 - G_2)z = D_3 - D_2 \quad (2.12)$$

De l'équation (2.11) on obtient :

$$x = -\frac{F_2 - F_1}{E_2 - E_1}y - \frac{G_2 - G_1}{E_2 - E_1}z + \frac{D_2 - D_1}{E_2 - E_1} \quad (2.13)$$

En remplaçant cela dans l'équation (2.12) :

$$y = \left[\frac{-(E_3 - E_1)(G_2 - G_1) + (G_3 - G_1)(F_2 - F_1)}{-(F_3 - F_1)(E_2 - E_1) + (E_3 - E_1)(F_2 - F_1)} \right] z + \frac{-(D_3 - D_1)(E_2 - E_1) + (E_3 - E_1)(D_2 - D_1)}{-(F_3 - F_1)(E_2 - E_1) + (E_3 - E_1)(F_2 - F_1)} \quad (2.14)$$

donc

$$y = \frac{H_1}{H_2}z + \frac{H_3}{H_2} \quad (2.15)$$

avec

$$\begin{aligned} H_1 &= -(E_3 - E_1)(G_2 - G_1) + (G_3 - G_1)(F_2 - F_1) \\ H_2 &= -(F_3 - F_1)(E_2 - E_1) + (E_3 - E_1)(F_2 - F_1) \\ H_3 &= -(D_3 - D_1)(E_2 - E_1) + (E_3 - E_1)(D_2 - D_1) \end{aligned} \quad (2.16)$$

De l'équation (2.11) on obtient aussi :

$$y = -\frac{E_2 - E_1}{F_2 - F_1}x - \frac{G_2 - G_1}{F_2 - F_1}z + \frac{D_2 - D_1}{E_2 - E_1} \quad (2.17)$$

En remplaçant toujours dans l'équation (2.12) :

$$x = \left[\frac{-(G_3 - G_1)(F_2 - F_1) + (F_3 - F_1)(G_2 - G_1)}{-(F_3 - F_1)(E_2 - E_1) + (E_3 - E_1)(F_2 - F_1)} \right] z + \frac{-(D_3 - D_1)(F_2 - F_1) + (F_3 - F_1)(D_2 - D_1)}{-(F_3 - F_1)(E_2 - E_1) + (E_3 - E_1)(F_2 - F_1)} \quad (2.18)$$

donc

$$x = \frac{H_5}{H_4}z + \frac{H_3}{H_4} \quad (2.19)$$

avec

$$\begin{aligned} H_2 &= -(F_3 - F_1)(E_2 - E_1) + (E_3 - E_1)(F_2 - F_1) \\ H_4 &= -(D_3 - D_1)(F_2 - F_1) + (F_3 - F_1)(D_2 - D_1) \\ H_5 &= -(G_3 - G_1)(F_2 - F_1) + (G_3 - G_1)(F_2 - F_1) \end{aligned} \quad (2.20)$$

Il nous reste à calculer le z en remplaçant l'expression de x (2.19) et y (2.17) dans (2.8) on aura :

$$z = \frac{M - \sqrt{M^2 - 4LN}}{2L} \quad (2.21)$$

avec

$$\begin{aligned}
 L &= 1 + \frac{H_5^2 + H_1^2}{H_2^2} \\
 M &= -\frac{2(H_5H_4 + H_1H_3)}{H_2^2} + \frac{(E_1H_5 + F_1H_1)}{H_2} + G_1 \\
 N &= D_1 + \frac{H_4^2 + H_3^2}{H_2^2} - \frac{H_4H_1 + F_1H_3}{H_2} - \frac{(E_1H_4 + F_1H_3)}{H_2}
 \end{aligned} \tag{2.22}$$

Remarque 1 : La résolution de l'équation du second degré en z donne en réalité deux solutions, chacune d'entre-elles correspondant au centre d'une sphère passant par les trois points C_i , les centres de ces deux sphères (qui sont donc les solutions de l'équation) sont symétriques par rapport au plan constitué par les trois points C_i , la solution qui correspond à la position réelle de la nacelle sera donc celle qui est située en dessous de ce plan (en concordance avec l'espace de travail) et donc, celle avec un signe « - », l'autre solution est rejetée.

2.4.2 Modèle géométrique inverse

Le modèle géométrique inverse du robot Delta permet de trouver les trois angles articulaires afin d'avoir une position (x, y, z) souhaitée de sa nacelle. Plusieurs formulations ont été proposés pour le calcul du modèle géométrique inverse du robot Delta [54, 55].

On peut résoudre les équations (2.8) en $(\alpha_1, \alpha_2, \alpha_3)$ et trouver le MGI. Cette solution a été proposée dans [54].

En reprenant l'équation (2.6), on sépare les termes en $(\cos\alpha_i)$ et en $(\sin\alpha_i)$:

$$\begin{aligned}
 2l_1z\sin\alpha_i + (2rl_1 - 2l_1x\cos\varphi_i - 2l_1y\sin\varphi_i)\cos\alpha_i &= (l_2^2 - l_1^2 - r^2 + \\
 2r\sin\varphi_i - x^2 - y^2 - z^2 + 2rx\cos\varphi_i) &\quad \text{Avec } i = 1, 2, 3.
 \end{aligned} \tag{2.23}$$

On pose :

$$\begin{cases}
 A_i = -2l_1z \\
 B_i = 2rl_1 - 2l_1x\cos\varphi_i - 2l_1y\sin\varphi_i \\
 C_i = l_2^2 - l_1^2 - r^2 + 2r\sin\varphi_i - x^2 - y^2 - z^2 + 2rx\cos\varphi_i
 \end{cases}$$

Donc l'équation (2.23) devient :

$$-A_i\sin\alpha_i + B_i\cos\alpha_i = C_i \tag{2.24}$$

En posant :

$$t = \tan\left(\frac{\alpha_i}{2}\right) \Rightarrow \cos\alpha_i = \frac{1-t^2}{1+t^2} \quad \text{et} \quad \sin\alpha_i = \frac{2t}{1+t^2}$$

On aura :

$$(B_i + C_i)t^2 + 2A_it + (C_i - B_i) = 0 \quad \Rightarrow \quad t = \frac{-A_i \pm \sqrt{A_i^2 - C_i^2 + B_i^2}}{B_i + C_i}$$

donc

$$\tan\left(\frac{\alpha_i}{2}\right) = \frac{-A_i \pm \sqrt{A_i^2 - C_i^2 + B_i^2}}{B_i + C_i} \tag{2.25}$$

La solution est donnée par :

$$\alpha_i = 2\arctan\left(\frac{-A_i \pm \sqrt{A_i^2 - C_i^2 + B_i^2}}{B_i + C_i}\right) \quad \text{avec : } i=1, 2, 3. \quad (2.26)$$

Remarque 2 : La résolution de l'équation en $(\cos\alpha_i)$ et en $(\sin\alpha_i)$ donne deux solutions, chacune d'elles correspondent à un point d'intersection du cercle de C_i avec la sphère de centre $P(x, y, z)$. La solution qui correspond à l'angle réel du bras (en concordance avec l'espace de travail) sera celle qui donne l'angle de plus faible valeur, donc celle avec un signe « + » (2.26), l'autre solution est rejetée.

2.4.3 Modèle dynamique

Dans tout système physique, un bon modèle dynamique peut suffire pour caractériser et représenter complètement le fonctionnement de celui-ci. Pour une structure robotisée, l'utilisation du modèle dynamique est primordiale dans les étapes de conception et de commande du robot.

Dans le cas des robots manipulateurs, les équations dynamiques ou modèle dynamique donnent la relation entre l'effort des actionneurs, les accélérations et les trajectoires du mouvement résultant de l'organe terminal.

Le modèle dynamique d'un robot exprime cette relation dans un sens ou dans l'autre. C'est pour cela que nous avons deux cas : Modèle dynamique direct et Modèle dynamique inverse.

- **Modèle dynamique direct (MDD) :** Exprime la variation des positions, vitesses et accélérations articulaires résultant d'un certain effort exercé sur les actionneurs. Autrement dit, le MDD permet de prévoir le comportement du système (mouvement du robot) pour une force ou couple exercée.
- **Modèle dynamique inverse (MDI) :** Exprime la valeur de l'effort à exercer sur les actionneurs afin d'obtenir un certain profil sur les positions, vitesses et accélérations articulaires. Le MDI permet donc de trouver la force ou le couple à exercer pour avoir une trajectoire désirée.

Le MDD est utilisé afin d'introduire un modèle de simulation du système, ce modèle remplace ainsi le système réel et permet de faire différents tests en simulation prévoyant le comportement du robot.

Pour le MDI, il est souvent utilisé lors de la commande des robots dont fait partie le Delta (méthodes de commande basées sur l'utilisation de couple a priori) le MDI permet de prévoir à partir de la trajectoire désirée les couples articulaires nécessaires. Cette utilisation sera illustrée dans le chapitre relatif à la synthèse de lois de commande.

Cinématique de vitesse

Pour calculer le MDI du robot Delta on doit tous d'abord donné l'expression de la matrice Jacobienne. La matrice Jacobienne est une correspondance entre les vitesses dans l'espace opérationnel et les vitesses dans l'espace articulaire. Pour calculer la matrice Jacobienne du robot Delta, on peut utiliser l'ensemble des contraintes reliant l'espace opérationnel avec l'espace articulaire. Les trois équations de contraintes pour le robot Delta peut être choisi comme suit (figure 2.14) :

$$\|P_c P_{b_i}\|_2^2 - l_2^2 = 0 \quad i = 1, 2, 3. \quad (2.27)$$

où la longueur de l'avant-bras l_2 est supposée constante.

Soit s_i dénoté le vecteur $P_{c_i}P_{b_i}$. Comme la norme euclidienne peut être écrite comme $s_i^T s_i$, ainsi d'après la figure 2.13 le vecteur s_i peut être écrit comme suit :

$$s_i = O_i P_{b_i} - (O_i P_{a_i} + P_{a_i} P_{c_i}) = \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} - R_{0/i} \left(\begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} l_1 \cos \alpha_i \\ 0 \\ l_1 \sin \alpha_i \end{bmatrix} \right) \quad (2.28)$$

où

$$R_{0/i} = \begin{pmatrix} \cos \varphi_i & -\sin \varphi_i & 0 \\ \sin \varphi_i & \cos \varphi_i & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

On dérive l'équation (2.27) par rapport au temps :

$$s_i^T \dot{s}_i + \dot{s}_i^T s_i = 0 \quad (2.29)$$

À partir de (2.29) on trouve :

$$s_i^T \dot{s}_i = 0 \quad (2.30)$$

La dérivée de s_i est donnée par :

$$\dot{s}_i = \begin{bmatrix} \dot{x}_n \\ \dot{y}_n \\ \dot{z}_n \end{bmatrix} - R_{0/i} \begin{bmatrix} -l_1 \sin \alpha_i \\ 0 \\ l_1 \cos \alpha_i \end{bmatrix} \dot{\alpha}_i = \dot{X}_n - b_i \dot{\alpha}_i \quad i = 1, 2, 3 \quad (2.31)$$

avec

$$b_i = R_{0/i} \begin{bmatrix} -l_1 \sin \alpha_i \\ 0 \\ l_1 \cos \alpha_i \end{bmatrix} \quad (2.32)$$

ainsi

$$s_i^T \begin{bmatrix} \dot{x}_n \\ \dot{y}_n \\ \dot{z}_n \end{bmatrix} - s_i^T b_i \dot{\alpha}_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.33)$$

Ce qui peut être exprimé sous forme d'une matrice considérant tous les trois bras robotisés :

$$\begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix} \dot{X}_n - \begin{bmatrix} s_1^T b_1 & 0 & 0 \\ 0 & s_2^T b_2 & 0 \\ 0 & 0 & s_3^T b_3 \end{bmatrix} \dot{\alpha} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.34)$$

À partir de l'équation (2.34) la matrice Jacobienne pour un robot Delta est donnée par :

$$\dot{X}_n = J \dot{\alpha} \quad (2.35)$$

avec

$$J = \begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix}^{-1} \begin{bmatrix} s_1^T b_1 & 0 & 0 \\ 0 & s_2^T b_2 & 0 \\ 0 & 0 & s_3^T b_3 \end{bmatrix} \quad (2.36)$$

Remarque 3 : La matrice jacobienne J n'est pas seulement en fonction des cordonnés articulaires α , comme le cas des robots sériels, mais elle est aussi une fonction des cordonnés opérationnelles X_n .

Cinématique d'accélération

L'accélération cinématique représente la relation entre l'accélération dans l'espace articulaire et l'accélération dans l'espace opérationnel. Pour calculer cette représentation, on peut dériver l'équation (2.34) par rapport au temps comme suit :

$$\begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix} \ddot{X}_n + \begin{bmatrix} \dot{s}_1^T \\ \dot{s}_2^T \\ \dot{s}_3^T \end{bmatrix} \dot{X}_n - \left(\begin{bmatrix} s_1^T b_1 & 0 & 0 \\ 0 & s_2^T b_2 & 0 \\ 0 & 0 & s_3^T b_3 \end{bmatrix} \ddot{\alpha} + \begin{bmatrix} \dot{s}_1^T b_1 + s_1^T \dot{b}_1 & 0 & 0 \\ 0 & \dot{s}_2^T b_2 + s_2^T \dot{b}_2 & 0 \\ 0 & 0 & \dot{s}_3^T b_3 + s_3^T \dot{b}_3 \end{bmatrix} \dot{\alpha} \right) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.37)$$

Avec (2.35) et (2.36) on peut écrire :

$$\ddot{X}_n = \begin{bmatrix} s_1^T \\ s_2^T \\ s_3^T \end{bmatrix}^{-1} \left(- \begin{bmatrix} \dot{s}_1^T \\ \dot{s}_2^T \\ \dot{s}_3^T \end{bmatrix} J + \begin{bmatrix} \dot{s}_1^T b_1 + s_1^T \dot{b}_1 & 0 & 0 \\ 0 & \dot{s}_2^T b_2 + s_2^T \dot{b}_2 & 0 \\ 0 & 0 & \dot{s}_3^T b_3 + s_3^T \dot{b}_3 \end{bmatrix} \dot{\alpha} \right) \dot{\alpha} + J \ddot{\alpha} \quad (2.38)$$

Dans l'équation (2.38), la dérivée par rapport au temps de la matrice jacobienne est le terme multipliant par $\dot{\alpha}$. L'accélération de l'effecteur \ddot{X}_n est reliée à l'accélération articulaire $\ddot{\alpha}$ comme suit :

$$\ddot{X}_n = J \dot{\alpha} + J \ddot{\alpha} \quad (2.39)$$

La dynamique de l'actionneur

Dans la plupart des robots industriels, un réducteur est introduit entre chaque actionneur et bras articulaire (voir figure 2.16.), avec :

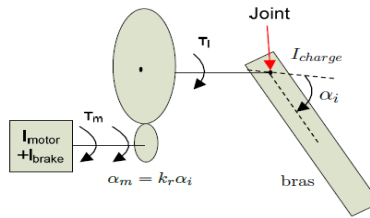


Figure 2.16 – Actionneur avec réducteur

$$\alpha_m = k_r \alpha_i \quad (2.40)$$

où α_m est le déplacement angulaire de l'arbre du moteur, α est le déplacement angulaire de bras de robot, tandis que la constante k_r est le rapport de réduction. τ_i est le couple de charge à l'axe du robot et τ_m est le couple produit par l'actionneur à l'axe de l'arbre de moteur.

Compte tenu de l'équation (2.40) on peut écrire :

$$\tau_m = \frac{\tau_i}{k_r} \quad (2.41)$$

Le modèle dynamique inverse d'un robot parallèle peut être calculer en utilisant différentes méthodes (ex : Lagrange ou Newton-Euler).

Pour le robot Delta, la complexité de la dynamique du modèle se pose principalement en raison du mouvement de l'avant-bras. Ce problème peut être simplifié si leurs inerties

de rotation sont négligées. Ainsi, la force entre la nacelle et le bras est dans une direction donnée par l'orientation de l'avant-bras. Le modèle pour le robot Delta est calculé avec la méthode Newton-Euler avec les hypothèses simplificatrices suivantes :

- Le moment d'inertie des avant-bras est négligé
- La masse de l'avant bras m_{ab} est répartie à ses deux extrémités : rm_{ab} sur sa partie supérieure (coude), et $(1 - r)m_{ab}$ sur sa partie inférieure (nacelle) ($0 < r < 1$) [56].
- Les effets de l'élasticité et de frottement sont négligés.

Paramètres dynamiques

Les paramètres suivants sont utilisés pour calculer le modèle dynamique du robot Delta. La masse totale agissant sur la nacelle est la somme de la masse de la nacelle m_n , la masse de la charge m_{charge} et les 3 masses qu'ont contribué chacune des 3 avant-bras $3(1 - r)m_{fb}$.

$$m_{nt} = m_n + m_{charge} + 3(1 - r)m_{fb} \quad (2.42)$$

r est le rapport de la masse de l'avant-bras. La valeur de r est choisie égale à $2/3$ [56]. La position du centre de masse pour chacune des parties supérieures de bras est calculée à l'aide de l'équation du centre de masse.

$$r_{Gb} = l_1 \frac{\frac{1}{2}m_b + m_c + rm_{fb}}{m_{br}} \quad (2.43)$$

avec

$$m_{br} = m_b + m_c + rm_{fb} \quad (2.44)$$

où m_b est la masse de la partie supérieure du bras, m_c est la masse du coude, m_{fb} est la masse de l'avant-bras.

L'inertie agit sur la partie supérieure de bras est la somme de l'inertie du moteur I_m , l'inertie de frein moteur I_{brake} et l'inertie du bras de I_{bc} . L'inertie du moteur et du frein moteur peut être exprimée comme l'inertie au niveau articulaire multiplié avec la quadratique de rapport de réduction. L'inertie au niveau articulaire est alors donnée par :

$$I_{bi} = I_m k r^2 + I_{brake} k r^2 + I_{bc} \quad (2.45)$$

où I_{bc} est la somme de l'inertie créée à partir de la partie supérieure du bras et de l'inertie de masse créée du point final de bras supérieur, ce qui est donné par :

$$I_{bc} = \frac{m_b}{3} l_1^2 + l_1^2 (m_c + rm_{fb}) = l_1^2 \left(\frac{m_b}{3} + m_c + rm_{fb} \right) \quad (2.46)$$

Calcul du modèle dynamique en utilisant le principe du travail virtuel

Il existe deux forces agissent sur la nacelle. La force de gravité G_n et la force inertielle (d'Alembert) F_n . Ses expressions sont données par :

$$G_n = m_{nt} (0 \quad 0 \quad -g)^T \quad F_n = m_{nt} \ddot{X}_n \quad (2.47)$$

La contribution de ses deux forces dans l'espace articulaire peut être calculer avec la transposée de la matrice Jacobienne :

$$\tau_n = J^T F_n = J^T m_{nt} \ddot{X}_n \quad (2.48)$$

$$\tau_{Gn} = J^T G_n = J^T m_{nt} (0 \quad 0 \quad -g)^T \quad (2.49)$$

Deux types de couples agissants sur les bras actionnés. Le couple τ_{Gb} produit par la force gravitationnelle de chaque bras et le couple τ_b produite à partir de l'inertie force (d'Alembert) agissant sur chaque bras. La contribution gravitationnelle peut être calculée comme la force qui agit perpendiculairement à la partie supérieure du bras à travers le centre de masse comme illustré à la figure 2.17.

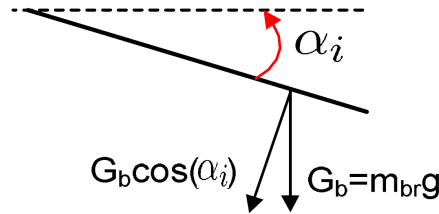


Figure 2.17 – Force gravitationnelle agissant sur la partie supérieure du bras d'un robot Delta

Nous donnons

$$\tau_{Gb} = r_{Gb} G_b (\cos\alpha_1 \quad \cos\alpha_2 \quad \cos\alpha_3)^T \quad (2.50)$$

où G_b est la force gravitationnelle agissant sur le centre de masse de chaque bras.

La contribution de chaque couple de la partie supérieure du bras sur les articulations peuvent être exprimée par :

$$\tau_b = I_b \ddot{\alpha} \quad (2.51)$$

Où I_b est la matrice d'inertie du bras dans l'espace articulaire et elle est donnée par :

$$I_b = \begin{bmatrix} I_{b1} & 0 & 0 \\ 0 & I_{b2} & 0 \\ 0 & 0 & I_{b3} \end{bmatrix} \quad (2.52)$$

Selon le principe d'Alembert la contribution de toutes les forces d'inertie doit être égale à la contribution de toutes les forces no inertielle. L'application de ce principe au niveau de l'articulation nous permet de trouver :

$$\tau + \tau_{Gn} + \tau_{Gb} = \tau_b + \tau_n \quad (2.53)$$

où τ est le vecteur des couples appliqués aux articulations. τ_n contenir le terme \ddot{X}_n ce qui peut être exprimé dans l'espace articulaire par l'équation (2.39), cela donne :

$$\tau = (I_b + m_{nt} J^T J) \ddot{\alpha} + J^T m_{nt} \dot{J} \dot{\alpha} - \tau_{Gn} - \tau_{Gb} \quad (2.54)$$

Ce qui peut être écrit comme suit :

$$\tau = M(\alpha) \ddot{\alpha} + C(\alpha, \dot{\alpha}) \dot{\alpha} + G(\alpha) \quad (2.55)$$

où $M(\alpha) \in \mathbb{R}^{3 \times 3}$ est la matrice d'inertie, $C(\alpha, \dot{\alpha}) \dot{\alpha} \in \mathbb{R}^3$ représente les forces de Coriolis et centrifuges, et $G(\alpha) \in \mathbb{R}^3$ est le vecteur des forces de gravitation agissant sur le manipulateur.

2.5 Conclusion

Dans ce chapitre, nous avons présenté les robots parallèles en général et le robot parallèle Delta en particulier. Quelques avantages et inconvénients des robots parallèles ont été évoqués. Par ailleurs, une description détaillée du robot Delta ISIR88 disponible au niveau de Laboratoire de Commandes des Processus a été présentée, avec son modèle géométrique direct, inverse ainsi que son modèle dynamique.

Chapitre 3

Commande par Apprentissage Itératif à Trajectoires Répétitives des Robots

3.1 Introduction

Le nombre de robots manipulateurs utilisés dans des tâches répétitives a augmenté durant les dernières années. Les trajectoires désirées lors de l'exécution des tâches sont souvent répétées sur un intervalle de temps donné. Un exemple de ses tâches est la ligne d'assemblage manufacturière, où le robot doit exécuter la même trajectoire d'une manière répétée. Afin d'exploiter ce phénomène de répétition dans l'amélioration de la commande, plusieurs schémas ont été introduits pour améliorer les performances en poursuite. Dans ce contexte, une commande par apprentissage à commutation adaptative a été proposée dans [9]. Cette commande est la combinaison d'un régulateur PD classique et d'une commande par anticipation (feedforward) CAI de type PD. L'approche proposée assure une rapidité dans la convergence. Dans [15], les auteurs ont implémenté la CAI pour commander un robot parallèle, où ils ont utilisés un modèle linéaire pour synthétiser la loi de commande. Tandis que, dans [18] les auteurs ont proposé une CAI robuste pour commander un actionneur magnétique à trois degrés de liberté, où la commande classique PD plus CAI type PD est combinée avec un terme de robustification pour compenser la perturbation extérieure non répétitive.

Dans ce chapitre, nous présentons trois lois de CAI pour réaliser la poursuite de trajectoire des robots manipulateurs rigides effectuant des trajectoires répétitives. La stabilité asymptotique du système en boucle fermée est démontrée pour les trois commandes en utilisant la théorie de Lyapunov. Ensuite nous allons présenter les résultats de simulation de la première loi de commande et les résultats expérimentaux pour la deuxième et troisième commande.

3.2 Propriétés générales des robots manipulateurs et hypothèses

3.2.1 Propriétés générales des robots manipulateurs rigides

On considère le modèle des robots manipulateurs rigides décrits par l'équation suivante :

$$M(q_k)\ddot{q}_k + C(q_k, \dot{q}_k)\dot{q}_k + G(q_k) + W_k = \tau_k \quad (3.1)$$

où, k est le nombre d'itérations, et W_k représente les perturbations extérieures qui peuvent être répétitives ou non répétitives.

En utilisant le modèle nominal, (3.1) devient :

$$M_n(q_k)\ddot{q}_k + C_n(q_k, \dot{q}_k)\dot{q}_k + G_n(q_k) + \tau_{d1_k} = \tau_k \quad (3.2)$$

τ_{d1_k} égale à :

$$\tau_{d1_k} = \Delta M(q_k)\ddot{q}_k + \Delta C(q_k, \dot{q}_k)\dot{q}_k + \Delta G(q_k) + W_k \quad (3.3)$$

Les propriétés suivantes sont valables pour les robots manipulateurs [57, 58]

(P1) La matrice d'inertie $M_n(q_k)$ est symétrique, définie positive, et bornée :

$$0 < \alpha \leq \|M_n(q_k)\| \leq \beta$$

où α , et β sont deux constantes.

(P2) Il existe une constante positive K_{M_n} , sachant que la matrice d'inertie $M_n(q_k)$ est globalement lipschitzienne :

$$\|M_n(q_{k+1}) - M_n(q_k)\| < K_{M_n}\|q_{k+1} - q_k\|$$

(P3) $G_n(q_k)$ est globalement lipschitzienne :

$$\|G_n(q_{k+1}) - G_n(q_k)\| \leq K_g\|q_{k+1} - q_k\|$$

où K_g est une constante positive.

(P4) Les bornes supérieures suivantes sont valables :

$$\|C_n(q_k, \dot{q}_k)\| \leq K_{c1}\|\dot{q}_k\|, \quad \|G_n(q_k)\| \leq K_G, \quad \forall q_k, \dot{q}_k \in \mathbb{R}^n$$

avec K_{c1} et K_G sont des constantes positives.

(P5) Pour un robot ayant uniquement des articulations rotationnelles, il existe deux constantes $K_{c1} > 0$ et $K_{c2} > 0$, sachant que

$$\|C_n(q_{k+1}, \dot{q}_{k+1})\dot{q}_{k+1} - C_n(q_k, \dot{q}_k)\dot{q}_{k+1}\| \leq \frac{K_{c1}\|\dot{q}_{k+1} - \dot{q}_k\|\|\dot{q}_{k+1}\| + K_{c2}\|q_{k+1} - q_k\|\|\dot{q}_{k+1}\|^2}{\|\dot{q}_{k+1}\|^2}$$

(P6) Les matrices d'inertie et celle des forces centrifuges et de Coriolis vérifient la relation suivante :

$$q_k^T (\dot{M}(q_k) - 2C(q_k, \dot{q}_k))q_k = 0, \quad \forall q_k \in \mathbb{R}^3$$

3.2.2 Hypothèses de travail

Considérons les hypothèses suivantes qui pourront ne pas être utilisées en même temps.

Hypothèse 1 : La vitesse articulaire du robot est bornée par une constante V_m et la perturbation $\tau_{d1_k}(t)$ est bornée, tel que

$$\|\dot{q}_k\| \leq V_m, \quad \|\tau_{d1_k} t\| \leq l_d \quad (3.4)$$

Hypothèse 2 : La trajectoire désirée ainsi que ses 1^{ère} et 2^{ème} dérivée sont bornés.

Hypothèse 3 : La condition de réinitialisation de l'erreur est :

$$q_k(0) = q_d(0), \quad \dot{q}_k(0) = \dot{q}_d(0), \quad \forall k \in \mathbb{Z}_+ \quad (3.5)$$

Hypothèse 4 : La condition d'alignement de l'erreur est :

$$q_{k-1}(T) = q_k(0), \quad \dot{q}_{k-1}(T) = \dot{q}_k(0), \quad \forall k \in \mathbb{Z}_+ \quad (3.6)$$

Hypothèse 5 : La trajectoire désirée est spatialement fermée (l'organe terminal démarre d'un point et retourne au même point à la fin la tâche) c-à-d :

$$q_d(0) = q_d(T), \quad \dot{q}_d(0) = \dot{q}_d(T) \quad (3.7)$$

Notre objectif est de développer une commande $\tau_k(t)$ qui assure la convergence de $q_k(t)$ et $\dot{q}_k(t)$ vers les trajectoires désirées $q_d(t)$ et $\dot{q}_d(t)$ respectivement, quand k tend vers l'infini, i.e. $\lim_{k \rightarrow \infty} \tilde{q}_k(t) = q_d(t) - q_k(t) = 0$, $\lim_{k \rightarrow \infty} \dot{\tilde{q}}_k(t) = \dot{q}_d(t) - \dot{q}_k(t) = 0$, $\forall t \in [0, T]$.

3.3 Commande par apprentissage itératif à base de modèle dynamique

La loi de commande CAI que nous allons développer est dédiée au problème de poursuite de trajectoire des robots manipulateurs effectuant des tâches répétitives sujet à des perturbations extérieures et/ou en présence d'erreurs de modélisation. Le schéma de commande est composé d'un régulateur utilisant partiellement le modèle dynamique afin de compenser une grande partie du couplage dans le robot. Il est composé aussi d'une CAI type PD pour améliorer les performances en poursuite à travers la répétition de la trajectoire ainsi qu'un terme de robustification qui sert à compenser l'effet des perturbations.

3.3.1 Conception de la commande

La loi de commande est donnée ci-dessous :

$$\tau_k = M_n(q_k)[K_p \tilde{q}_k + K_d \dot{\tilde{q}}_k + u_k] \quad (3.8)$$

avec

$$u_{k+1} = u_k + \Lambda \tilde{q}_k + \Gamma \dot{\tilde{q}}_k + \mu \operatorname{sgn}(\tilde{z}_k) \quad (3.9)$$

Les variables \tilde{z}_k et z_k sont définies par :

$$\tilde{z}_k = z_{k+1} - z_k \quad (3.10)$$

$$z_k(t) = \dot{\tilde{q}}_k(t) + \zeta \tilde{q}_k(t) \quad (3.11)$$

où, K_p, K_d, Γ , et λ sont des matrices diagonales définies positives, et μ, ζ sont des constantes positives.

Le schéma de commande proposé est illustré par la figure 3.1, où x_d représente la trajectoire désirée dans l'espace opérationnel et IGM indique le modèle géométrique inverse.

Comme on peut constater, le schéma de commande proposé 3.1 se compose de deux parties : un retour d'état représenté par une prémultiplication entre la matrice d'inertie et le régulateur PD, ce qui permet de compenser l'effet du couplage entre les actionneurs, contrairement aux régulateurs PD/PID et les CAI classiques [3] qui n'arrivent pas à compenser le couplage fort entre les articulations en engendrant la dégradation des

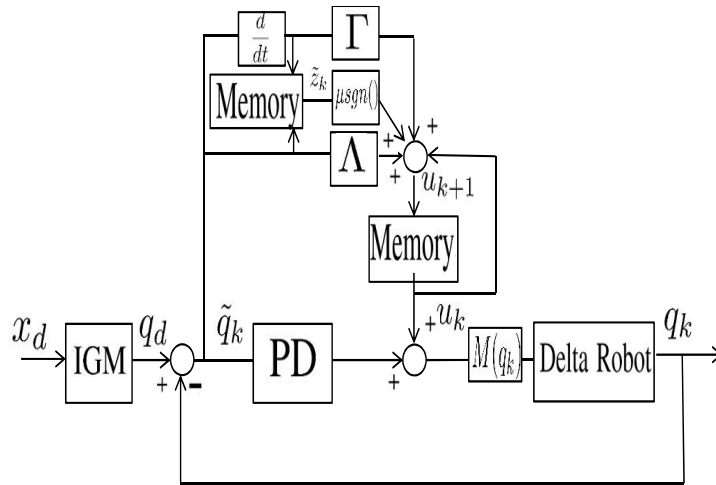


Figure 3.1 – Schéma de la commande par apprentissage itératif à base de modèle dynamique

performances, particulièrement lors des mouvements rapides. La deuxième partie, se compose d'un terme d'apprentissage représenté par une CAI de type PD plus la fonction signe qui est synthétisée en se basant sur l'analyse de la convergence, ce qui permet de prendre en charge l'effet des incertitudes du modèle et les perturbations externes, contrairement aux cas des PD/PID et CAI classique.

3.3.2 Analyse de la convergence

Pour simplifier l'écriture, on note :

$$M_n(q_k) = M_{n,k}, C_n(q_k, \dot{q}_k) = C_{n,k}, G_n(q_k) = G_{n,k}, \text{ et } \tau_{d1_k}(t) = \tau_{d1_k}$$

Dans la k-ème itération, l'équation (3.2) peut se réécrire comme suit

$$\ddot{q}_k = k_p \tilde{q}_k + k_d \dot{\tilde{q}}_k + u_k - M_{n,k}^{-1} C_{n,k} \dot{q}_k - M_{n,k}^{-1} G_{n,k} - M_{n,k}^{-1} \tau_{d1_k} \quad (3.12)$$

De même, dans la (k+1)-ème itération, nous avons

$$\ddot{q}_{k+1} = k_p \tilde{q}_{k+1} + k_d \dot{\tilde{q}}_{k+1} + u_{k+1} - M_{n,k+1}^{-1} C_{n,k+1} \dot{q}_{k+1} - M_{n,k+1}^{-1} G_{n,k+1} - M_{n,k+1}^{-1} \tau_{d1_{k+1}} \quad (3.13)$$

Pour la preuve de la convergence, nous supposons que

$$K_p = \zeta K_d \quad \text{et} \quad \Lambda = \zeta \Gamma \quad (3.14)$$

Nous définissons également la variable δq_k comme suit

$$\delta q_k = \tilde{q}_{k+1} - \tilde{q}_k \quad (3.15)$$

Théorème 1 : Considérons le système (3.2) vérifiant les hypothèses (1-3) et les propriétés (P1-P5), la loi de commande donnée par (3.8) garantit la convergence en position et en vitesse sur un intervalle de temps fini $[0, T]$ lorsque le nombre d'itérations k tend vers l'infini, i.e., $\lim_{k \rightarrow \infty} q_k(t) = q_d(t)$ et $\lim_{k \rightarrow \infty} \dot{q}_k(t) = \dot{q}_d(t) \forall t \in [0, T]$, si les gains de la commande sont sélectionnés comme suit :

$$2\lambda_{\min}(K_d - \zeta I) \geq \lambda_{\max}(\Gamma) - 2\zeta + \frac{a_1 + a_2}{\zeta} \geq 0 \quad (3.16)$$

$$2\lambda_{\min}(K_d - \zeta I) \geq \lambda_{\max}(\Gamma) + a_3 \geq 0 \quad (3.17)$$

$$4AB \geq C^2 \quad (3.18)$$

$$\mu - 2l_d\alpha^{-1} > \gamma \quad (3.19)$$

où

$$A = \zeta^2\lambda_{\max}(\Gamma) - 2\zeta^2\lambda_{\min}(K_d - \zeta I) - 2\zeta^3 + \zeta(a_1 + a_2) \quad (3.20)$$

$$B = \lambda_{\max}(\Gamma) - 2\lambda_{\min}(K_d - \zeta I) + a_3 \quad (3.21)$$

$$C = a_1 + a_2 + \zeta a_3 \quad (3.22)$$

et

$$a_1 = 2(\alpha^{-1}K_g + K_M\alpha^{-2}K_G) \quad (3.23)$$

$$a_2 = 2(\alpha^{-1}K_{c2}V_m^2 + \alpha^{-2}K_MK_{c1}V_m^2) \quad (3.24)$$

$$a_3 = 4\alpha^{-1}K_{c1}V_m \quad (3.25)$$

γ est une constante positive.

Preuve :

On considère la fonction de Lyapunov suivante

$$V_k(t) = \int_0^t z_k^T \Gamma z_k d\sigma \quad (3.26)$$

soit

$$\begin{aligned} \Delta V_k = V_{k+1} - V_k &= \int_0^t z_{k+1}^T \Gamma z_{k+1} d\sigma - \int_0^t z_k^T \Gamma z_k d\sigma \\ &= \int_0^t \tilde{z}_k^T \Gamma \tilde{z}_k + 2\tilde{z}_k^T \Gamma z_k d\sigma \end{aligned} \quad (3.27)$$

avec $\ddot{q}_{k+1} - \ddot{q}_k = \ddot{q}_d - \ddot{q}_{k+1} - \ddot{q}_d + \ddot{q}_k$

Soustrayant (3.13) de (3.12), assure l'égalité suivante

$$\begin{aligned} \ddot{q}_{k+1} - \ddot{q}_k &= -K_p(\tilde{q}_{k+1} - \tilde{q}_k) - K_d(\dot{\tilde{q}}_{k+1} - \dot{\tilde{q}}_k) - u_{k+1} + u_k \\ &\quad + M_{n,k+1}^{-1}G_{n,k+1} - M_{n,k}^{-1}G_k + M_{n,k+1}^{-1}C_{n,k+1}\dot{q}_{k+1} \\ &\quad - M_{n,k}^{-1}C_{n,k}\dot{q}_k + M_{n,k+1}^{-1}\tau_{d1_{k+1}} - M_{n,k}^{-1}\tau_{d1_k} \end{aligned} \quad (3.28)$$

Combinant (3.9), (3.10), (3.11), (3.14) et (3.15) on trouve

$$\begin{aligned} \dot{\tilde{z}}_k + (K_d - \zeta I)\tilde{z}_k + \zeta^2\delta q_k - (M_{n,k+1}^{-1}G_{n,k+1} - M_{n,k}^{-1}G_{n,k}) - \\ (M_{n,k+1}^{-1}C_{n,k+1}\dot{q}_{k+1} - M_{n,k}^{-1}C_{n,k}\dot{q}_k) - (M_{n,k+1}^{-1}\tau_{d1_{k+1}} - M_{n,k}^{-1}\tau_{d1_k}) + \mu \text{sgn}(\tilde{z}_k) = -\Gamma z_k \end{aligned} \quad (3.29)$$

Remplaçant (3.29) dans (3.27) on obtient

$$\begin{aligned} \Delta V_k = \int_0^t \tilde{z}_k^T \Gamma \tilde{z}_k - 2\tilde{z}_k^T \dot{\tilde{z}}_k - 2\tilde{z}_k^T (K_d - \zeta I)\tilde{z}_k - \\ 2\zeta^2\tilde{z}_k^T \delta q_k + 2\tilde{z}_k^T (M_{n,k+1}^{-1}G_{n,k+1} - M_{n,k}^{-1}G_{n,k}) + \\ 2\tilde{z}_k^T (M_{n,k+1}^{-1}C_{n,k+1}\dot{q}_{k+1} - M_{n,k}^{-1}C_{n,k}\dot{q}_k) + \\ 2\tilde{z}_k^T (M_{n,k+1}^{-1}\tau_{d1_{k+1}} - M_{n,k}^{-1}\tau_{d1_k} - \mu \text{sgn}(\tilde{z}_k)) d\sigma \end{aligned} \quad (3.30)$$

par conséquent

$$\begin{aligned} \Delta V_k = & \int_0^t \tilde{z}_k^T \Gamma \tilde{z}_k - 2\tilde{z}_k^T \dot{\tilde{z}}_k - 2\tilde{z}_k^T (K_d - \zeta I) \tilde{z}_k - \\ & 2\zeta^2 \tilde{z}_k^T \delta q_k + 2\tilde{z}_k^T [M_{n,k+1}^{-1} (G_{n,k+1} - G_{n,k}) + \\ & (M_{n,k+1}^{-1} - M_{n,k}^{-1}) G_{n,k}] + 2\tilde{z}_k^T [M_{n,k+1}^{-1} ((C_{n,k+1} - C_{n,k}) \dot{q}_{k+1} \\ & + C_{n,k} (\dot{q}_{k+1} - \dot{q}_k)) + (M_{n,k+1}^{-1} - M_{n,k}^{-1}) C_{n,k} \dot{q}_k] + \\ & 2\tilde{z}_k^T (M_{n,k+1}^{-1} \tau_{d1_{k+1}} - M_{n,k}^{-1} \tau_{d1_k} - \mu \operatorname{sgn}(\tilde{z}_k)) d\sigma \end{aligned} \quad (3.31)$$

L'hypothèse (1) et la propriété (P1) assurent que $\tilde{z}_k^T (M_{n,k+1}^{-1} \tau_{d1_{k+1}} - M_{n,k}^{-1} \tau_{d1_k}) \leq \|\tilde{z}_k^T\| (2l_d \alpha^{-1})$, de là on trouve

$$\int_0^t \tilde{z}_k^T (M_{n,k+1}^{-1} \tau_{d1_{k+1}} - M_{n,k}^{-1} \tau_{d1_k} - \mu \operatorname{sgn}(\tilde{z}_k)) d\sigma \leq \int_0^t \|\tilde{z}_k^T\| (2l_d \alpha^{-1} - \mu) d\sigma \quad (3.32)$$

Les propriétés (P1-P5), (3.32) et l'hypothèse (1) conduisent à

$$\begin{aligned} \Delta V_k \leq & \int_0^t \tilde{z}_k^T \Gamma \tilde{z}_k - 2\tilde{z}_k^T \dot{\tilde{z}}_k - 2\tilde{z}_k^T (K_d - \zeta I) \tilde{z}_k - \\ & 2\zeta^2 \tilde{z}_k^T \delta q_k + 2\|\tilde{z}_k^T\| (\alpha^{-1} K_g + K_M \alpha^{-2} K_G) \|\delta q_k\| + \\ & 2\|\tilde{z}_k^T\| (\alpha^{-1} K_{c2} V_m^2 + \alpha^{-2} K_M K_{c1} V_m^2) \|\delta q_k\| + \\ & 2\|\tilde{z}_k^T\| (2\alpha^{-1} K_{c1} V_m) \|\delta \dot{q}_k\| + 2\|\tilde{z}_k^T\| (2l_d \alpha^{-1} - \mu) d\sigma \end{aligned} \quad (3.33)$$

Remplaçant (3.11), (3.23), (3.24) et (3.25) dans (3.33), on obtient

$$\begin{aligned} \Delta V_k \leq & \int_0^t \delta \dot{q}_k^T \Gamma \delta \dot{q}_k + \zeta^2 \delta q_k^T \Gamma \delta q_k + 2\zeta \delta \dot{q}_k^T \Gamma \delta q_k - 2\tilde{z}_k^T \dot{\tilde{z}}_k - \\ & 2\zeta^2 \delta q_k^T (K_d - \zeta I) \delta q_k - 2\delta \dot{q}_k^T (K_d - \zeta I) \delta \dot{q}_k - \\ & 4\zeta \delta \dot{q}_k^T (K_d - \zeta I) \delta q_k - 2\zeta^2 \delta \dot{q}_k^T \delta q_k - 2\zeta^3 \|\delta q_k\|^2 + \\ & \zeta a_1 \|\delta q_k\|^2 + a_1 \|\delta q_k\| \|\delta \dot{q}_k\| + \zeta a_2 \|\delta q_k\|^2 + \\ & a_2 \|\delta q_k\| \|\delta \dot{q}_k\| + a_3 \|\delta \dot{q}_k\|^2 + \zeta a_3 \|\delta q_k\| \|\delta \dot{q}_k\| + \\ & 2\|\tilde{z}_k^T\| (2l_d \alpha^{-1} - \mu) d\sigma \end{aligned} \quad (3.34)$$

À l'aide de l'hypothèse (3), on peut obtenir

$$\begin{aligned} \Delta V_k \leq & -\|\tilde{z}_k\|^2 - \zeta^2 \|\delta q_k\|^2 - \zeta \delta q_k^T (2K_d - 2\zeta I - \Gamma) \delta q_k + \\ & \int_0^t \delta \dot{q}_k^T \Gamma \delta \dot{q}_k + \zeta^2 \delta q_k^T \Gamma \delta q_k - 2\zeta^3 \|\delta q_k\|^2 - \\ & 2\zeta^2 \delta q_k^T (K_d - \zeta I) \delta q_k - 2\delta \dot{q}_k^T (K_d - \zeta I) \delta \dot{q}_k + \\ & \zeta a_1 \|\delta q_k\|^2 + a_1 \|\delta q_k\| \|\delta \dot{q}_k\| + \zeta a_2 \|\delta q_k\|^2 + \\ & a_2 \|\delta q_k\| \|\delta \dot{q}_k\| + a_3 \|\delta \dot{q}_k\|^2 + \zeta a_3 \|\delta q_k\| \|\delta \dot{q}_k\| + \\ & 2\|\tilde{z}_k^T\| (2l_d \alpha^{-1} - \mu) d\sigma \end{aligned} \quad (3.35)$$

par conséquent

$$\begin{aligned} \Delta V_k \leq & -\|\tilde{z}_k\|^2 - \zeta^2 \|\delta q_k\|^2 - \zeta \lambda_{\min}(2K_d - 2\zeta I - \Gamma) \|\delta q_k\|^2 \\ & + \int_0^t [\lambda_{\max}(\Gamma) - 2\lambda_{\min}(K_d - \zeta I) + a_3] \|\delta \dot{q}_k\|^2 + \\ & [\zeta^2 \lambda_{\max}(\Gamma) - 2\zeta^3 - 2\zeta^2 \lambda_{\min}(K_d - \zeta I) \\ & + \zeta(a_1 + a_2)] \|\delta q_k\|^2 + [a_1 + a_2 + \zeta a_3] \|\delta q_k\| \|\delta \dot{q}_k\| + \\ & 2\|\tilde{z}_k^T\| (2l_d \alpha^{-1} - \mu) d\sigma \end{aligned} \quad (3.36)$$

L'utilisation de (3.20), (3.21), et (3.22) implique que

$$\begin{aligned} \Delta V_k \leq & -\|\tilde{z}_k\|^2 - \zeta^2 \|\delta q_k\|^2 - \zeta \lambda_{\min}(2K_d - 2\zeta I - \Gamma) \|\delta q_k\|^2 \\ & + \int_0^t A \|\delta q_k\|^2 + B \|\delta \dot{q}_k\|^2 + C \|\delta q_k\| \|\delta \dot{q}_k\| \\ & + 2\|\tilde{z}_k^T\| (2I_d \alpha^{-1} - \mu) d\sigma \end{aligned} \quad (3.37)$$

Ainsi, à partir de (3.19) on peut obtenir

$$\begin{aligned} \Delta V_k < & -\|\tilde{z}_k\|^2 - \zeta^2 \|\delta q_k\|^2 - \zeta \lambda_{\min}(2K_d - 2\zeta I - \Gamma) \|\delta q_k\|^2 \\ & + \int_0^t A (\|\delta q_k\| + \frac{C}{2A} \|\delta \dot{q}_k\|)^2 + (B - \frac{C^2}{4A}) \|\delta \dot{q}_k\|^2 - 2\gamma \|\tilde{z}_k^T\| d\sigma \end{aligned} \quad (3.38)$$

À partir de (3.16), (3.17), (3.18) et (3.19) on trouve

$$\Delta V_k < 0, \quad ie \quad V_{k+1} < V_k \quad (3.39)$$

De (3.39) nous concluons que lorsque k tend vers l'infini, V_k tend vers zéro, ce qui implique que $z_k \rightarrow 0$, et à partir de la définition de z_k (3.11), nous pouvons obtenir

$$\lim_{k \rightarrow \infty} \tilde{q}_k(t) = \lim_{k \rightarrow \infty} \dot{\tilde{q}}_k = 0, \quad \forall t \in [0, T]. \quad (3.40)$$

Remarque 1 : La CAI-basée modèle est applicable aux robots manipulateurs caractérisés par des modèles non linéaires incertains, contrairement aux approches présentées dans [13–15], où les lois de commande sont développées pour les systèmes linéaires à temps discret. De plus, cette commande convient pour les systèmes à perturbations répétitives et non répétitives, contrairement à celles développées dans [9, 11] où la loi de commande est valable uniquement lors de la répétition de la perturbation.

3.3.3 Résultats de simulation

Considérons le robot Delta de la figure 3.2 ayant les valeurs des paramètres données dans le tableau 2.1.

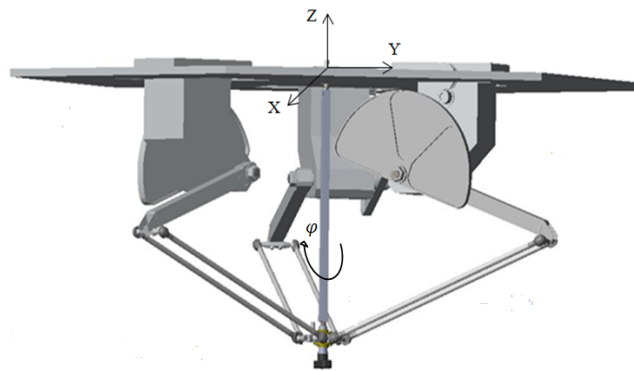


Figure 3.2 – Robot Delta

Les paramètres du robot sont données comme suit : $K_{c1} = 0.44 \text{ kgm}^2$, $K_{c2} = 2.675 \text{ kgm}^2$, $K_g = 0.354 \text{ kg.m}^2/\text{s}^2$, $K_G = 0.442 \text{ kg.m}^2/\text{s}^2$, $V_m = 5 \text{ rad/s}$, $\alpha = 0.3 \text{ kgm}^2$, $K_M = 0.09 \text{ kgm}^2$.

Les erreurs de modélisation sont données par : $\Delta M(q_k) = 0.1 * M(q_k)$, $\Delta C(q_k, \dot{q}_k) = 0.1 * C(q_k, \dot{q}_k)$, $\Delta G(q_k) = 0.1 * G(q_k)$. Les perturbations sont supposées varier dans le temps et dans les itérations [5] : $d1(t) = d2(t) = d3(t) = 0.2 \cdot \text{rand}(k) \sin 2\pi t$ N.m, où, $\text{rand}(k)$ est une fonction aléatoire prenant ses valeurs entre 0 et 1.

La trajectoire désirée est utilisée le long de l'axe x et de l'axe z , où, elle est représentée par un polynôme de cinquième degré avec une vitesse et accélération initiales et finales égale à zéro. La trajectoire désirée est exprimée comme suit :

$$x(t) = x_i + (x_f - x_i) \left(6 \frac{t^5}{t_f^5} - 15 \frac{t^4}{t_f^4} + 10 \frac{t^3}{t_f^3} \right) \quad (3.41)$$

où x_i est la position initiale, x_f est la position finale, et t_f est la durée du mouvement.

La trajectoire désirée démarre de la position initiale $(-0.15, 0, -0.37)$ m vers la position finale $(0.15, 0, -0.37)$ m, puis elle retourne à la position initiale. La hauteur de transition est égale à 0.04 m. Tous ces mouvements sont exécutés durant 0.4 seconde.

Afin d'évaluer les performances de poursuite, les critères de l'erreur quadratique moyenne (*the Root Mean Square Error (RMSE)*) et le critère de l'erreur maximale absolue (*the Maximum Absolute Error (MaxAE)*) seront utilisés. Leurs expressions sont données comme suit :

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y_{d_i})^2} \quad (3.42)$$

$$MaxAE = Max(|y_i - y_{d_i}|) \quad (3.43)$$

où y_d est la trajectoire désirée, y_i est la trajectoire actuelle et n désigne le nombre d'échantillons dans une itération.

Les gains du régulateur sont sélectionnés afin de minimiser les critères RMSE et MaxAE après 90 itérations. Les gains de la commande proposée sont choisis : $K_p = \text{diag}\{600, 600, 600\}$, $K_d = \text{diag}\{40, 40, 40\}$, $\Lambda = \text{diag}\{18.3, 18.3, 18.3\}$, $\Gamma = \text{diag}\{1.22, 1.22, 1.22\}$, et $\zeta = 15$, tandis que, les gains de PD sont données par : $K_{p(PID)} = \text{diag}\{12, 12, 12\}$, $K_{d(PID)} = \text{diag}\{0.18, 0.18, 0.18\}$, $K_{I(PID)} = \text{diag}\{2, 2, 2\}$, et les gains de PD plus CAI sont choisis : $K_{p(PD-CAI)} = \text{diag}\{8, 8, 8\}$, $K_{d(PD-CAI)} = \text{diag}\{0.08, 0.08, 0.08\}$, $\Lambda_{PD-CAI} = \text{diag}\{0.5, 0.5, 0.5\}$, $\Gamma_{PD-CAI} = \text{diag}\{0.02, 0.02, 0.02\}$.

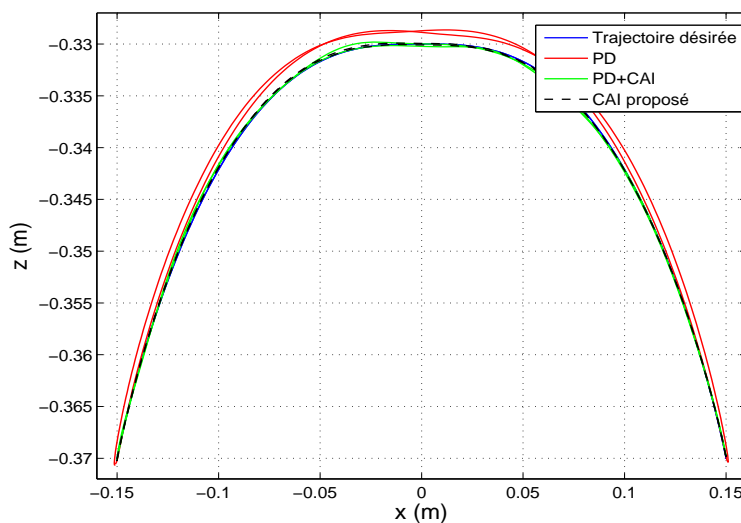


Figure 3.3 – Poursuite de trajectoire dans l'espace opérationnel

La figure 3.3 présente la poursuite de trajectoire dans l'espace opérationnel après 90 itérations sous la commande proposée, la commande PD, et PD plus CAI type PD. La

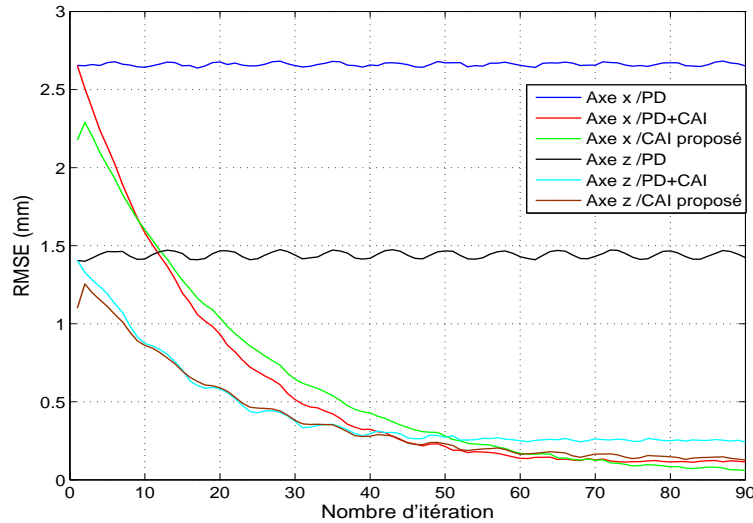


Figure 3.4 – Critère RMSE suivant les itérations

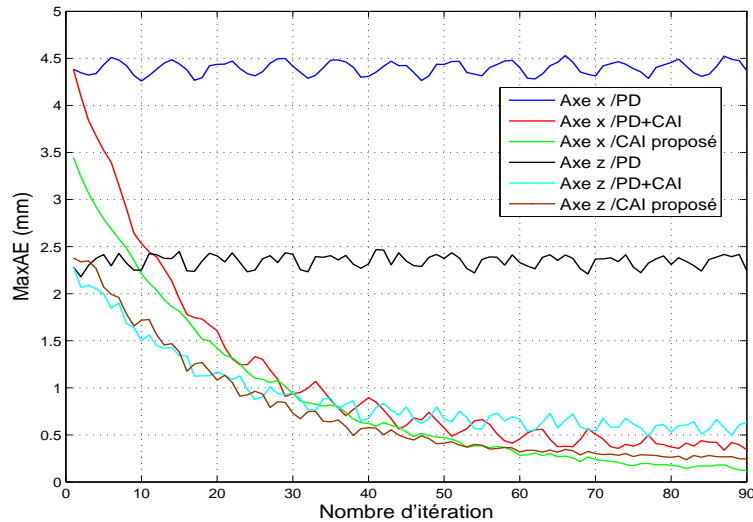


Figure 3.5 – Critère MaxAE suivant les itérations

figure 3.4 et 3.5 indiquent la progression avec l'augmentation du nombre d'itérations du RMSE et du MaxAE respectivement. On constate, que les performances sont améliorées d'une itération à l'autre, où le RMSE a diminué de 2,17 mm sur l'axe x et 1,10 mm sur l'axe z dans la première itération à 0,06 mm sur l'axe des x et 0,13 mm sur l'axe z dans la 90-ème itération, tandis que le régulateur PD fournit un RMSE égale à 1,48 mm sur l'axe des x et 0,96 mm sur l'axe z, et le PD plus CAI type PD donne un RMSE égal à 0,12 mm sur l'axe des x et 0,25 mm sur l'axe z. La figure 3.6, montre l'erreur de poursuite pour l'articulation 1, 2 et 3 respectivement (l'erreur de poursuite de la 3-ème articulation est semblable à la 2-ème à cause de la nature de la trajectoire), pour la 1-ère, 10-ème, 30-ème, 50-ème et la 90-ème itération. On peut voir, que la trajectoire effectuée suit la trajectoire désirée au cours des itérations malgré l'existence des perturbations externes. Le tableau 3.1 résume les performances obtenues en fonction du nombre d'itérations. La figure 3.7, représente le signal du couple de la 1-ère, 2-ème et la 3-ème articulation respectivement. Comme il est indiqué, le couple reste presque le même au cours des itérations, ce qui représente un avantage de la commande proposée, où, elle peut améliorer les performances de poursuite avec l'augmentation du nombre des itérations avec la même énergie du début.

Table 3.1 – Performances à travers les itérations

Itération	1	10	30	50	90
RMSE axe-x (mm)	2.17	1.60	0.65	0.28	0.06
MaxAE axe-x (mm)	3.43	2.21	0.93	0.48	0.13
RMSE axe-z (mm)	1.10	0.85	0.38	0.23	0.13
MaxAE axe-z (mm)	2.40	1.73	0.75	0.43	0.26

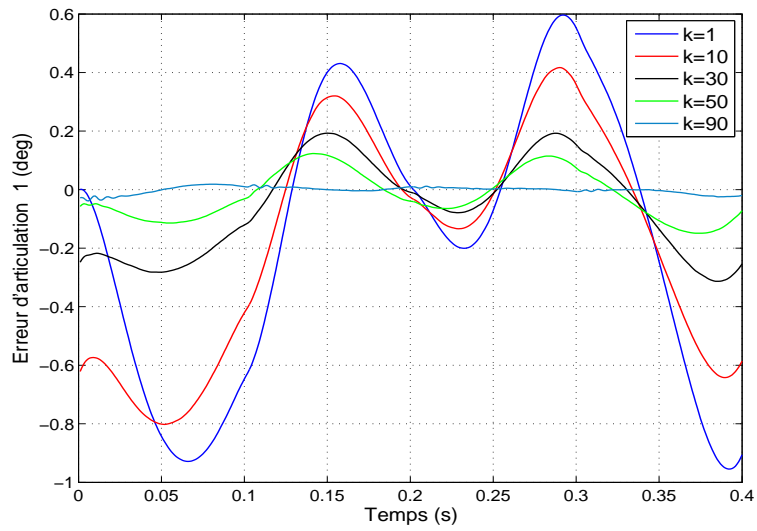
Test de robustesse

Une charge supplémentaire de 200 g a été rajoutée sur la nacelle, entre la 30-ème et 90-ème itération, afin de tester la poursuite de trajectoire en présence des charges utiles,

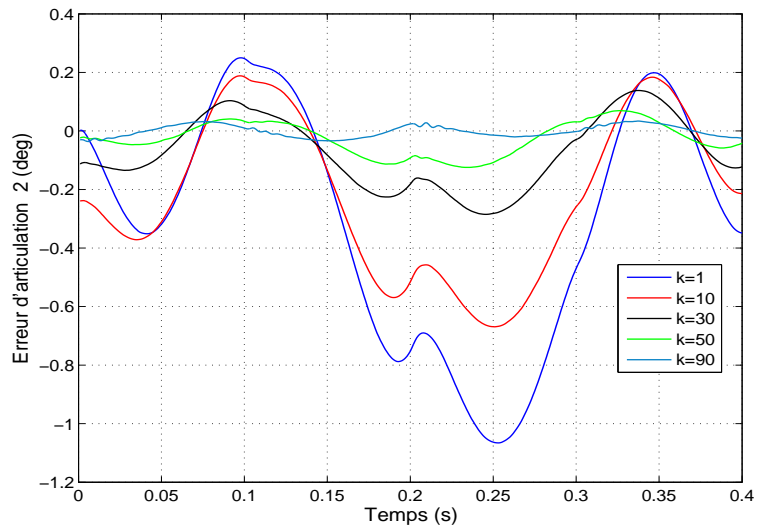
Les résultats de simulation sont présentés de la figure 3.8 jusqu'à la figure 3.10. On constate, que le régulateur PD a perdu ses performances après l'introduction de la charge supplémentaire, tandis que les performances du PD plus CAI type PD sont constantes entre la 60-ème et la 70-ème itération puis commencent à diverger. D'autre part, la commande proposée donne de meilleures performances, où, le RMSE et le MaxAE ont diminué avec un rythme plus rapide par rapport à la CAI classique et ils ont continué à diminuer même à la 90-ème itération. De plus, le couple du généré est plus petit que celui du PD et le PD plus CAI type PD, ce qui donne un grand avantage à l'approche proposée. Le tableau 3.2 résume les performances obtenues sous une charge supplémentaire de 200 g.

Table 3.2 – Performances à travers les itérations sous une charge supplémentaire de 200 g

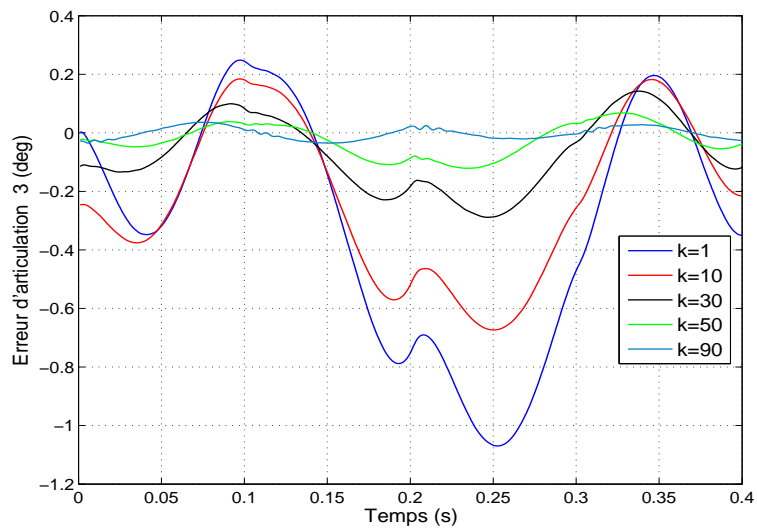
Itération	1	10	30	50	90
RMSE axe-x (mm)	2.17	1.60	2.76	1.22	0.20
MaxAE axe-x (mm)	3.43	2.21	4.02	1.85	0.33
RMSE axe-z (mm)	1.10	0.85	1.30	0.58	0.16
MaxAE axe-z (mm)	2.40	1.73	2.73	1.16	0.33



(a)

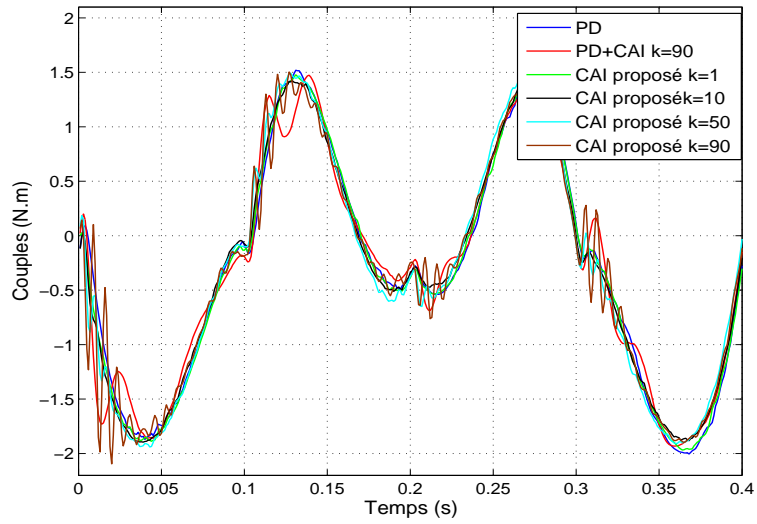


(b)

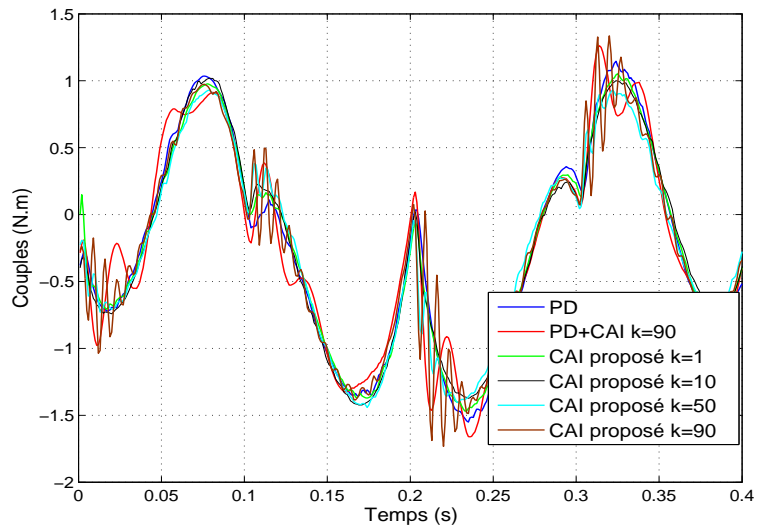


(c)

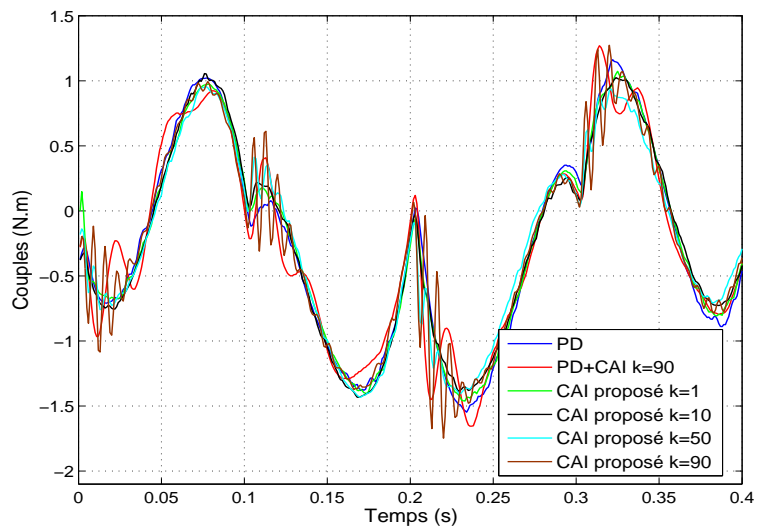
Figure 3.6 – Évolution de l'erreur de poursuite en fonction des itérations $k=1,10,30,50,90$. (a) articulation 1, (b) articulation 2, (c) articulation 3



(a)



(b)



(c)

Figure 3.7 – Entrées de commande pour les itérations $k=1,10,30,50,90$. (a) articulation 1, (b) articulation 2, (c) articulation 3

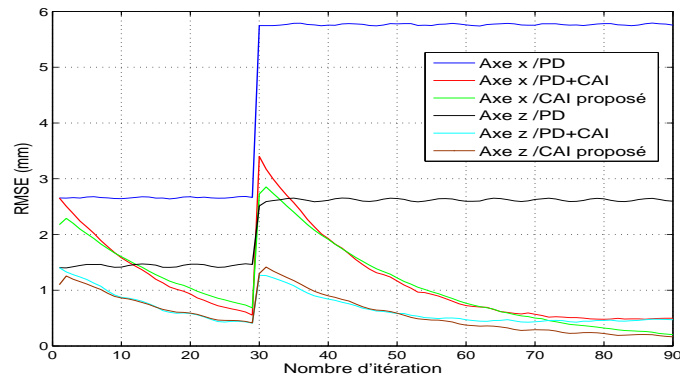


Figure 3.8 – Critère RMSE suivant les itérations sous une charge supplémentaire de 200 g

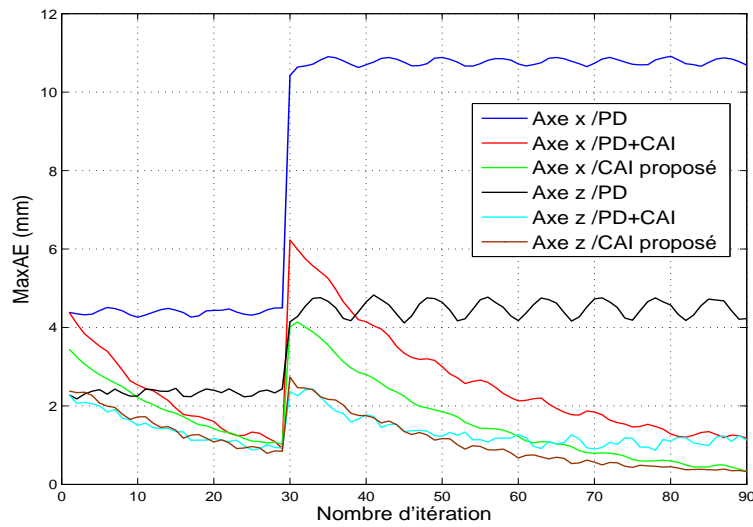


Figure 3.9 – Critère MaxAE suivant les itérations sous une charge supplémentaire de 200 g

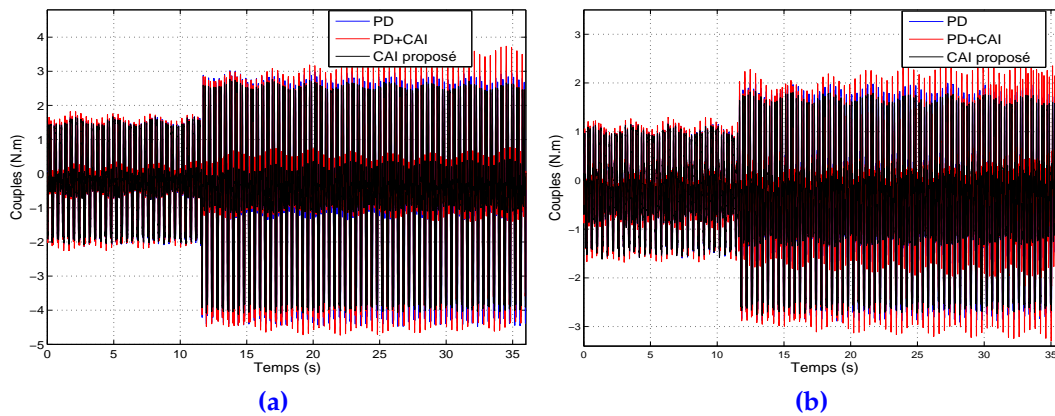


Figure 3.10 – Entrées de commande sous une charge supplémentaire de 200 g. (a) articulation 1, (b) articulation 2

3.4 Commande combinée PD-CAI type PD

3.4.1 Introduction

Le robot Delta est bien connu pour des opérations de prise et de pose qui exigent une haute précision aussi bien dans la phase statique que dans la phase dynamique. Étant donné les vitesses d'exécution élevées entraînant des effets de couplage, les commandes traditionnelles telles que PD/PID ne peuvent pas satisfaire les performances requises. Une CAI type PD est combinée avec un régulateur PD afin d'améliorer les performances d'une itération à l'autre sur un intervalle de temps fini.

La condition de réinitialisation de l'erreur est une hypothèse fondamentale dans le développement de la théorie de la CAI. Malheureusement, cette condition n'est pas nécessairement vérifiée en pratique, où il est difficile de réaliser une poursuite parfaite en début d'exécution de chaque itération. Au cours des années, un certain nombre de travaux ont été effectués pour remplacer cette condition par d'autres conditions telles que, la condition de l'erreur initiale aléatoire, la condition d'alignement et la condition de l'erreur initiale fixe [30, 59]. Pour être plus proche du comportement réel du robot, la condition d'alignement semble être la plus pratique et la plus adéquate pour les robots effectuant des mouvements continus, où l'état final de l'itération précédente est considéré comme état initial de l'itération actuelle. Néanmoins, l'hypothèse de la répétitivité de la trajectoire doit être vérifiée, ce qui est le cas dans les opérations de prise et de pose.

Une comparaison de cette approche, qui est partiellement indépendante du modèle dynamique, avec celle par couple a priori, qui est basée sur le modèle, a été effectuée en utilisant une trajectoire répétitive. La convergence asymptotique de la commande proposée a été prouvée en utilisant la théorie de Lyapunov. Dans le contexte de l'analyse de la stabilité, les travaux présentés dans [17] concernent l'analyse de la convergence avec un régulateur à paramètres fixes. Dans [18], la même commande a été reprise avec des gains adaptatifs. Dans [9] les auteurs proposent une commande à gain adaptatif avec un terme de robustification. Toutes ces approches considèrent la condition de réinitialisation lors de la démonstration de la stabilité. Notre contribution par rapport à ces travaux réside dans le fait que nous avons utilisé la condition d'alignement. La méthode proposée est validée expérimentalement sur le robot Delta ISIR88 disponible au laboratoire LCP.

3.4.2 Conception de la commande

La commande PD plus CAI type PD est exprimée comme suit :

$$\tau_k = K_p \tilde{q}_k + K_d \dot{\tilde{q}}_k + u_k \quad (3.44)$$

$$u_{k+1} = u_k + \Lambda \tilde{q}_k + \Gamma \dot{\tilde{q}}_k \quad (3.45)$$

où, K_p , K_d , Γ , et Λ sont des matrices diagonales définies positives.

L'expression de la loi de commande indique clairement sa simplicité et sa facilité de mise en œuvre pratique.

3.4.3 Analyse de la convergence

Considérons le modèle de robot suivant :

$$M(q_k)\ddot{q}_k + C(q_k, \dot{q}_k)\dot{q}_k + G(q_k) + d(t) = \tau_k \quad (3.46)$$

où $d(t)$ représente des perturbations externes répétitives.

En utilisant le développement de Taylor, l'équation (3.46) peut être linéarisé au tour la trajectoire désirée $(q_d, \dot{q}_d, \ddot{q}_d)$ comme suit :

$$M(t)\ddot{\tilde{q}}(t) + [C(t) + C_1(t)]\dot{\tilde{q}}(t) + F(t)\tilde{q}(t) + n(\ddot{\tilde{q}}, \dot{\tilde{q}}, \tilde{q}, t) - d(t) = H(t) - \tau(t) \quad (3.47)$$

$$\text{avec } M(t) = M(q_d(t))$$

$$C(t) = C(q_d(t), \dot{q}_d(t))$$

$$C_1(t) = \frac{\partial C}{\partial \dot{q}_k} \Big|_{q_d(t), \dot{q}_d(t)} \dot{q}_k(t)$$

$$F(t) = \frac{\partial M}{\partial q_k} \Big|_{q_d(t)} \ddot{q}_k(t) + \frac{\partial C}{\partial q_k} \Big|_{q_d(t), \dot{q}_d(t)} \dot{q}_k(t) + \frac{\partial G}{\partial q_k} \Big|_{q_d(t)}$$

$$H(t) = M(q_d(t))\ddot{q}_d(t) + C(q_d(t), \dot{q}_d(t))\dot{q}_d(t) + G(q_d(t))$$

Le terme $n_k(\ddot{\tilde{q}}, \dot{\tilde{q}}, \tilde{q}, t)$ contient les contributions d'ordre supérieur de $\ddot{\tilde{q}}, \dot{\tilde{q}}, \tilde{q}$, et par conséquent il a été négligé.

Dans la k -ème itération, l'équation (3.47) peut se réécrire comme suit

$$M(t)\ddot{\tilde{q}}_k(t) + [C(t) + C_1(t)]\dot{\tilde{q}}_k(t) + F(t)\tilde{q}_k(t) - d(t) = H(t) - \tau_k(t) \quad (3.48)$$

De même, dans la $(k+1)$ -ème itération, nous avons

$$M(t)\ddot{\tilde{q}}_{k+1}(t) + [C(t) + C_1(t)]\dot{\tilde{q}}_{k+1}(t) + F(t)\tilde{q}_{k+1}(t) - d(t) = H(t) - \tau_{k+1}(t) \quad (3.49)$$

Nous supposons que

$$\Lambda = \lambda \Gamma \quad (3.50)$$

où λ est une constante positive.

Théorème 2 [16] : Considérons la dynamique du robot (3.46) sous l'hypothèse d'alignement (4), les hypothèses (2, 5) et les propriétés (P1, P6). Alors, la loi de commande donnée par (3.44) garantit que l'erreur de poursuite en position et en vitesse converge vers zéro lorsque k tend vers l'infini sur un intervalle de temps fini $[0, T]$, i.e., $\lim_{k \rightarrow \infty} \tilde{q}_k(t) = \lim_{k \rightarrow \infty} \dot{\tilde{q}}_k(t) = 0 \forall t \in [0, T]$, si les gains sont sélectionnés pour tout $t \in [0, T]$ comme suit :

$$v_{p1} = \lambda_{\min}(2C_1 - \Gamma - 2M\lambda + 2K_d) > 0 \quad (3.51)$$

$$v_{p2} = \lambda_{\min}\left(2\frac{F}{\lambda} + 2\frac{K_p}{\lambda} - \Gamma + 2C - 2\frac{\dot{C}_1}{\lambda}\right) > 0 \quad (3.52)$$

$$v_{p1}v_{p2} \geq \left\| \frac{F}{\lambda} + \frac{K_p}{\lambda} - (C + C_1 + K_d - M\lambda) \right\|_{max}^2 \quad (3.53)$$

où $\lambda_{min}(\cdot)$ est la valeur propre minimal de matrice la (\cdot) , et $\|(\cdot)\|_{max} = \max\|(\cdot)\|$, pour $t \in [0, T]$.

Preuve

Soit la fonction de Lyapunov suivante :

$$V_k(t) = \int_0^t z_k^T \Gamma z_k d\sigma \quad (3.54)$$

où

$$z_k(t) = \dot{q}_k(t) + \lambda \tilde{q}_k(t) \quad (3.55)$$

soit

$$\Delta V_k = V_{k+1} - V_k, \tilde{z}_k = z_{k+1} - z_k, \delta q_k = \tilde{q}_{k+1} - \tilde{q}_k \quad (3.56)$$

par conséquent

$$\Delta V_k(t) = \int_0^t z_{k+1}^T \Gamma z_{k+1} d\sigma - \int_0^t z_k^T \Gamma z_k d\sigma = \int_0^t \tilde{z}_k^T \Gamma \tilde{z}_k + 2\tilde{z}_k^T \Gamma z_k d\sigma \quad (3.57)$$

En soustrayant(3.48) de (3.49), on trouve

$$\begin{aligned} M(t)(\ddot{q}_{k+1} - \ddot{q}_k) + [C(t) + C_1(t)](\dot{q}_{k+1} - \dot{q}_k) + F(t)(\tilde{q}_{k+1} - \tilde{q}_k) \\ = -K_p(\tilde{q}_{k+1} - \tilde{q}_k) - K_d(\dot{q}_{k+1} - \dot{q}_k) - \Lambda \tilde{q}_k - \Gamma \dot{q}_k \end{aligned} \quad (3.58)$$

En combinant les équations (3.55), (3.50), et (3.56) on obtient

$$\begin{aligned} M(t)\dot{\tilde{z}}_k + [C + C_1 - M\lambda + K_d]\tilde{z}_k \\ + [F + K_p - (C + C_1 + K_d - M\lambda)\lambda]\delta q_k = -\Gamma z_k \end{aligned} \quad (3.59)$$

À partir de (3.57) et (3.59) on trouve

$$\begin{aligned} \Delta V_k(t) = \int_0^t \tilde{z}_k^T \Gamma \tilde{z}_k d\sigma - 2 \int_0^t \tilde{z}_k^T M(t) \dot{\tilde{z}}_k d\sigma \\ - 2 \int_0^t \tilde{z}_k^T [C + C_1 - M\lambda + K_d] \tilde{z}_k d\sigma \\ - 2 \int_0^t \tilde{z}_k^T [F + K_p - (C + C_1 + K_d - M\lambda)\lambda] \delta q_k d\sigma \end{aligned} \quad (3.60)$$

En appliquant l'intégration par partie

$$\begin{aligned} \int_0^t \tilde{z}_k^T M(\sigma) \dot{\tilde{z}}_k d\sigma &= \tilde{z}_k^T M \tilde{z}_k \Big|_0^t - \int_0^t (\dot{\tilde{z}}_k^T M)' \tilde{z}_k d\sigma \\ &= \tilde{z}_k^T M \tilde{z}_k \Big|_0^t - \int_0^t \tilde{z}_k^T \dot{M} \tilde{z}_k - \int_0^t \tilde{z}_k^T \dot{M} \tilde{z}_k d\sigma \end{aligned} \quad (3.61)$$

La propriété (P6) conduit à $2 \int_0^t \tilde{z}_k^T M(\sigma) \dot{\tilde{z}}_k d\sigma = \tilde{z}_k^T(t)M(t)\tilde{z}_k(t) - \tilde{z}_k^T(0)M(0)\tilde{z}_k(0) - 2 \int_0^t \tilde{z}_k^T C \tilde{z}_k d\sigma$

De là, on obtient

$$\begin{aligned} \Delta V_k(t) = -\tilde{z}_k^T(t)M(t)\tilde{z}_k(t) + \tilde{z}_k^T(0)M(0)\tilde{z}_k(0) \\ - \int_0^t \tilde{z}_k^T (2C_1 - \Gamma - 2M\lambda + 2K_d) \tilde{z}_k d\sigma \\ - 2 \int_0^t \tilde{z}_k^T (F + K_p - (C + C_1 + K_d - M\lambda)\lambda) \delta q_k d\sigma \end{aligned} \quad (3.62)$$

La substitution de $\tilde{z}_k = \delta\dot{q}_k + \lambda\delta q_k$ dans (3.62) nous donne

$$\begin{aligned} \Delta V_k(t) = & -\tilde{z}_k^T(t)M(t)\tilde{z}_k(t) + \tilde{z}_k^T(0)M(0)\tilde{z}_k(0) \\ & - \int_0^t \delta\dot{q}_k^T(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta\dot{q}_k d\sigma \\ & - 2\lambda \int_0^t \delta q_k^T(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta\dot{q}_k d\sigma \\ & - \lambda^2 \int_0^t \delta q_k^T(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta q_k d\sigma \\ & - 2 \int_0^t \delta\dot{q}_k^T(F + K_P - (C + C_1 + K_d - M\lambda)\lambda)\delta q_k d\sigma \\ & - 2\lambda \int_0^t \delta q_k^T(F + K_P - (C + C_1 + K_d - M\lambda)\lambda)\delta q_k d\sigma \end{aligned} \quad (3.63)$$

Après une autre intégration par partie, on obtient

$$\begin{aligned} \int_0^t \delta q_k^T(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta\dot{q}_k d\sigma = & 2 \int_0^t \delta q_k^T(\dot{M}\lambda - \dot{C}_1)\delta q_k d\sigma \\ & - \int_0^t \delta q_k^T(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta\dot{q}_k d\sigma \\ & + \delta q_k^T(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta q_k \Big|_0^t \end{aligned} \quad (3.64)$$

La substitution de (3.64) dans (3.63) conduit à

$$\begin{aligned} \Delta V_k(t) = & -\tilde{z}_k^T(t)M(t)\tilde{z}_k(t) + \tilde{z}_k^T(0)M(0)\tilde{z}_k(0) \\ & - \lambda\delta q_k^T(t)(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta q_k(t) \\ & + \lambda\delta q_k^T(0)(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta q_k(0) - \int_0^t W_k d\sigma \end{aligned} \quad (3.65)$$

où

$$\begin{aligned} W_k = & \delta\dot{q}_k^T(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta\dot{q}_k \\ & + 2\lambda\delta\dot{q}_k^T\left(\frac{F}{\lambda} + \frac{K_p}{\lambda} - (C + C_1 + K_d - M\lambda)\right)\delta q_k \\ & + \lambda^2\delta q_k^T\left(2\frac{F}{\lambda} + 2\frac{K_p}{\lambda} - \Gamma + 2C - 2\frac{\dot{C}_1}{\lambda}\right)\delta q_k \end{aligned} \quad (3.66)$$

Prenons $D = \frac{F}{\lambda} + \frac{K_p}{\lambda} - (C + C_1 + K_d - M\lambda)$, et à partir de (3.51) et (3.52) on obtient

$$W_k \geq v_{p1}\|\delta\dot{q}_k\|^2 + 2\lambda\delta\dot{q}_k^T D\delta q_k + \lambda^2 v_{p2}\|\delta q_k\|^2 \quad (3.67)$$

L'inégalité de Cauchy-Schwartz conduite à

$$\delta\dot{q}_k^T D\delta q_k \geq -\|\delta\dot{q}_k\| \|D\|_{max}\|\delta q_k\| \quad (3.68)$$

À partir de (3.53), (3.67) et (3.68), on trouve

$$\begin{aligned} W_k & \geq v_{p1}\|\delta\dot{q}_k\|^2 - 2\lambda\|\delta\dot{q}_k\| \|D\|_{max}\|\delta q_k\| + \lambda^2 v_{p2}\|\delta q_k\|^2 \\ & = v_{p1}\left(\|\delta\dot{q}_k\| - \frac{\lambda}{v_{p1}}\|D\|_{max}\|\delta q_k\|\right)^2 \\ & + \lambda^2\left(v_{p2} - \frac{1}{v_{p1}}\|D\|_{max}^2\right)\|\delta q_k\|^2 \geq 0 \end{aligned} \quad (3.69)$$

prenons

$$w_k = v_{p1}\left(\|\delta\dot{q}_k\| - \frac{\lambda}{v_{p1}}\|D\|_{max}\|\delta q_k\|\right)^2 + \lambda^2\left(v_{p2} - \frac{1}{v_{p1}}\|D\|_{max}^2\right)\|\delta q_k\|^2 \quad (3.70)$$

par conséquent

$$\begin{aligned} \Delta V_k(t) \leq & -\tilde{z}_k^T(t)M(t)\tilde{z}_k(t) + \tilde{z}_k^T(0)M(0)\tilde{z}_k(0) \\ & - \lambda\delta q_k^T(t)(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta q_k(t) \\ & + \lambda\delta q_k^T(0)(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta q_k(0) - \int_0^t w_k d\sigma \end{aligned} \quad (3.71)$$

D'autre part, nous avons

$$V_{k+1}(t) = V_1(t) + \sum_{i=1}^k \Delta V_i(t) \quad (3.72)$$

La substitution de (3.71) dans (3.72) conduit à

$$\begin{aligned} V_{k+1}(t) \leq & V_1(t) + \sum_{i=1}^k [-\tilde{z}_i^T(t)M(t)\tilde{z}_i(t) + \tilde{z}_i^T(0)M(0)\tilde{z}_i(0)] \\ & + \sum_{i=1}^k [-\lambda\delta q_i^T(t)(2C_1 - \Gamma - 2M\lambda + 2K_d)(t)\delta q_i(t) \\ & + \lambda\delta q_i^T(0)(2C_1 - \Gamma - 2M\lambda + 2K_d)(0)\delta q_i(0)] \\ & - \sum_{i=1}^k \int_0^t w_i d\sigma \end{aligned} \quad (3.73)$$

Avec les hypothèses (2, 4, 5) on obtient

$$\begin{aligned} \tilde{q}_k(0) = \tilde{q}_{k-1}(T) & \implies \tilde{z}_k(0) = \tilde{z}_{k-1}(T), \delta q_k(0) = \delta q_{k-1}(T) \\ \dot{\tilde{q}}_k(0) = \dot{\tilde{q}}_{k-1}(T) & \implies \dot{\tilde{z}}_k(0) = \dot{\tilde{z}}_{k-1}(T), \delta \dot{q}_k(0) = \delta \dot{q}_{k-1}(T) \end{aligned} \quad (3.74)$$

et

$$q_d(T) = q_d(0) \implies \begin{aligned} M(0) &= M(T) \\ (2C_1 - \Gamma - 2M\lambda + 2K_d)(0) &= \\ (2C_1 - \Gamma - 2M\lambda + 2K_d)(T) \end{aligned} \quad (3.75)$$

À partir de (3.74) et (3.75) on obtient

$$\begin{aligned} \tilde{z}_{i-1}^T(T)M(T)\tilde{z}_{i-1}(T) &= \tilde{z}_i^T(0)M(0)\tilde{z}_i(0) \\ \delta q_{i-1}^T(T)(2C_1 - \Gamma - 2M\lambda + 2K_d)(T)\delta q_{i-1}(T) &= \\ \delta q_i^T(0)(2C_1 - \Gamma - 2M\lambda + 2K_d)(0)\delta q_i(0) & \quad i = 2 : k \end{aligned} \quad (3.76)$$

La substitution de (3.76) dans (3.73) conduit à

$$\begin{aligned} V_{k+1}(T) \leq & V_1(T) - \tilde{z}_k^T(T)M(T)\tilde{z}_k(T) \\ & - \lambda\delta q_k^T(T)(2C_1 - \Gamma - 2M\lambda + 2K_d)(T)\delta q_k(T) \\ & + \tilde{z}_1^T(0)M(0)\tilde{z}_1(0) \\ & + \lambda\delta q_1^T(0)(2C_1 - \Gamma - 2M\lambda + 2K_d)(0)\delta q_1(0) \\ & - \sum_{i=1}^k \int_0^T w_i(\sigma) d\sigma \end{aligned} \quad (3.77)$$

De là, on trouve

$$\begin{aligned} V_{k+1}(T) \leq & V_1(T) - \tilde{z}_k^T(T)M(T)\tilde{z}_k(T) - \lambda\delta q_k^T(T)v_{p1}\delta q_k(T) \\ & + \lambda\delta q_1^T(0)(2C_1 - \Gamma - 2M\lambda + 2K_d)(0)\delta q_1(0) \\ & + \tilde{z}_1^T(0)M(0)\tilde{z}_1(0) - \sum_{i=1}^k \int_0^T w_i(\sigma) d\sigma \end{aligned} \quad (3.78)$$

par conséquent

$$\begin{aligned} V_{k+1}(T) + \tilde{z}_k^T(T)M(T)\tilde{z}_k(T) + \lambda\delta q_k^T(T)v_{p1}\delta q_k(T) & \leq V_1(T) \\ + \tilde{z}_1^T(0)M(0)\tilde{z}_1(0) + \lambda\delta q_1^T(0)(2C_1 - \Gamma - 2M\lambda + 2K)(0)\delta q_1(0) & \\ - \sum_{i=1}^k \int_0^T w_i(\sigma) d\sigma & \end{aligned} \quad (3.79)$$

Le terme du côté droit de (3.79)

$$V_1(T) + \tilde{z}_1^T(0)M(0)\tilde{z}_1(0) + \lambda\delta q_1^T(0)(2C_1 - \Gamma - 2M\lambda + 2K_D)(0)\delta q_1(0)$$

est borné parce que nous avons

$$\tilde{z}_1(0) = z_2(0) - z_1(0) = \dot{\tilde{q}}_2(0) - \dot{\tilde{q}}_1(0) + \lambda\tilde{q}_2(0) - \lambda\tilde{q}_1(0),$$

où, $\tilde{q}_2(0) = \tilde{q}_1(T)$, $\dot{\tilde{q}}_2(0) = \dot{\tilde{q}}_1(T)$, par conséquent $\tilde{z}_1(0)$ est en fonction de la première erreur de position et de vitesse $(\dot{\tilde{q}}_1, \tilde{q}_1)$, de même pour $\delta q_1(0)$, par conséquent, le terme de droite est en fonction de $(\dot{\tilde{q}}_1, \tilde{q}_1)$. Selon la loi de commande donnée par (3.44), seul le régulateur PD fonctionne à la première itération, et il est connu que les gains positifs PD peuvent stabiliser un robot et conduire à une erreur statique non nulle, ce qui signifie que $(\dot{\tilde{q}}_1, \tilde{q}_1)$ sont bornées, ainsi que le terme sur le côté droit.

Comme le terme de gauche de l'équation (3.79) est positif, on peut avoir le résultat suivant :

$$\int_0^T w_k(\sigma) d\sigma \rightarrow 0 \quad \text{quand} \quad k \rightarrow \infty, \forall t \in [0, T] \quad (3.80)$$

À partir de l'équation (3.70) on peut conclure que

$$\|\delta\dot{q}_k\| \rightarrow 0, \|\delta q_k\| \rightarrow 0 \quad \text{quand} \quad k \rightarrow \infty \quad (3.81)$$

et puisque \tilde{q}_{k+1} et \tilde{q}_k sont deux vecteurs indépendants on trouve finalement :

$$\lim_{k \rightarrow \infty} \tilde{q}_k(t) = \lim_{k \rightarrow \infty} \dot{\tilde{q}}_k = 0, \forall t \in [0, T]. \quad (3.82)$$

3.4.4 Résultats expérimentaux

La commande est implantée sur un PC (Core Duo 2.8 Ghz) avec Windows XP et une extension en temps réel (RTX de Interval Zero Inc). Les actionneurs du robot sont des moteurs DC brushless (minertia motor mini series UGTMEM-03LB2), avec un rapport de transmission égale $r_g = 12$. L'algorithme de commande est programmé en langage C et exécuté avec une fréquence d'échantillonnage de 1 kHz. La figure 3.11 montre le banc d'essai réel.

Les gains du régulateur sont choisis comme suit : $K_p = \text{diag}\{2.2, 2.2, 2.2\}$, $K_d = \text{diag}\{0.0145, 0.0145, 0.0145\}$, $\Lambda = \text{diag}\{0.074, 0.074, 0.074\}$, $\Gamma = \text{diag}\{0.0015, 0.0015, 0.0015\}$. La valeur de la première itération u_1 est définie comme suit : $u_1 = [0, 0, 0]^T$.

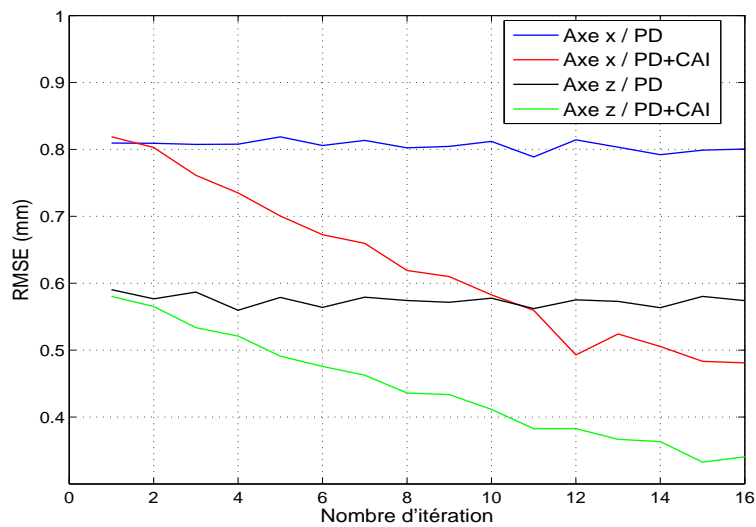
Nous donnons l'étude expérimentale dans deux cas.

Cas 1 : La trajectoire opérationnelle désirée est une semi-elliptique réalisée avec un profil parabolique et une accélération maximale de 10 m/s^2 . Il démarre à partir de la position initiale $(0.20, 0, -0.43)\text{m}$ pour atteindre la position finale $(-0.20, 0, -0.43)\text{m}$ et retourne à la position initiale pendant 0.88 secondes.

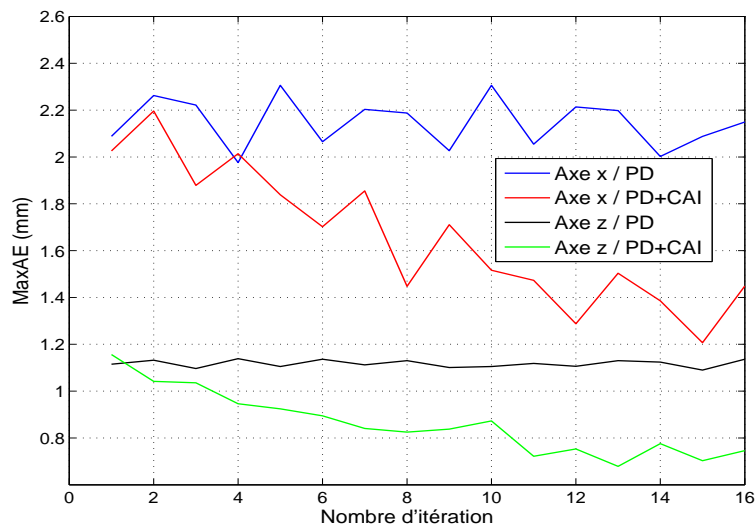
La figure 3.12 présente l'évolution au cours des itérations de RMSE et MaxAE sous la commande PD et PD plus CAI. Les deux régulateurs donnent les mêmes valeurs à la première itération. La CAI diminue clairement le RMSE et le MaxAE avec l'augmentation du nombre des itérations. D'autre part, le RMSE et le MaxAE sont pratiquement constants dans le cas où le PD fonctionne seul. La figure 3.13 montre la trajectoire dans l'espace opérationnel, et la figure 3.14 montre l'erreur de poursuite pour les articulations 1, 2 et 3 respectivement. On peut observer qu'avec l'augmentation du nombre des itérations l'erreur s'améliore d'environ 45% lors de l'utilisation de CAI (de 0.44° à 0.25° pour l'articulation 1 et de 0.25° à 0.14° pour l'articulation 2). À partir de la figure 3.15, les couples ont presque la même variation et amplitude. Le tableau 3.3 présente le MaxAE, le



Figure 3.11 – Photographie du banc d'expérimentation avec Robot Delta ISIR88



(a)



(b)

Figure 3.12 – Évolution des critères en expérimentation suivant les itérations. (a) RMSE, (b) MaxAE

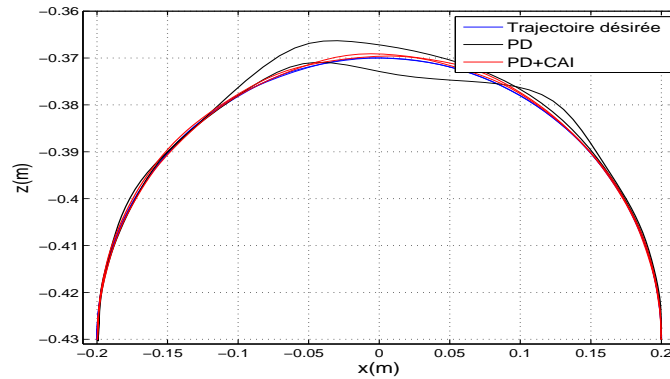


Figure 3.13 – Poursuite de trajectoire dans l'espace opérationnel en expérimentation

RMSE et l'erreur autour du point d'arrêt après 16 itérations. Il est clair que l'algorithme proposé permet d'améliorer les performances où le RMSE peut atteindre 0,48 mm dans l'axe des x au lieu de 0,80 mm obtenu par le régulateur PD.

Table 3.3 – Performances de poursuite

Commande	PD		PD+CAI	
	x	z	x	z
Axe				
RMSE (mm)	0.80	0.57	0.48	0.38
max e (mm)	2.15	1.14	1.44	0.74
Erreurs autour les points d'arrêt (mm)	0.26	0.10	0.1	0.0

Cas 2 : Afin de montrer l'efficacité du régulateur à une vitesses d'exécution élevée, une expérience similaire a été effectuée sous la même trajectoire désirée en 0.62 seconde et avec une accélération maximale de 20 m/s^2 .

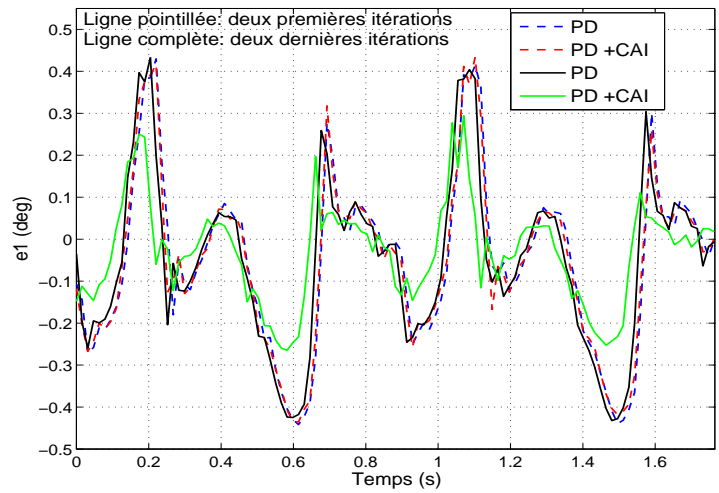
Les résultats expérimentaux sont présentés de la figure 3.16 jusqu'à la figure 3.18. Nous remarquons que le régulateur PD plus CAI engendre une diminution des erreurs de poursuite, du RMSE ainsi que du MaxAE à travers les itérations, tandis que le couple reste dans les valeurs admissibles (22 N.m). Le tableau 3.4 présente les performances, où on peut remarquer que malgré l'augmentation de l'accélération, la commande PD plus CAI reste celle donnant les meilleurs résultats.

Table 3.4 – Performances de poursuite sous une accélération de 2 G.

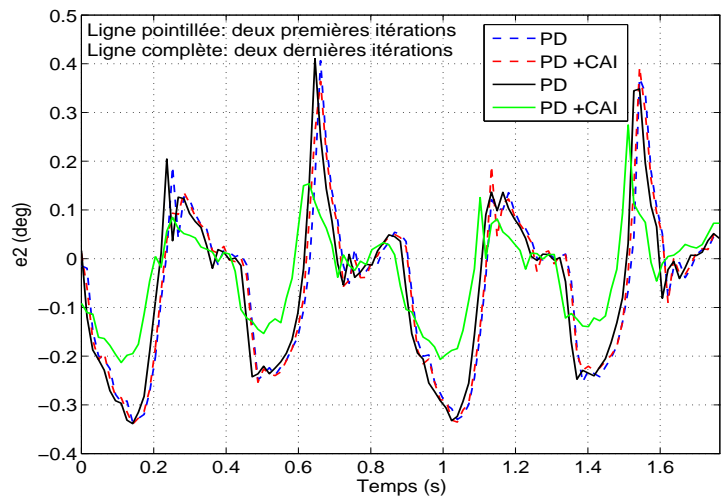
Commande	PD		PD+ILC	
	x	z	x	z
Axe				
RMSE (mm)	0.97	0.66	0.60	0.40
max e (mm)	10.31	5.67	9.28	4.75
Erreurs autour les points d'arrêt (mm)	0.70	0.30	0.50	0.20

3.4.5 Comparaison entre la commande combinée PD-CAI type PD et la commande par couple a priori

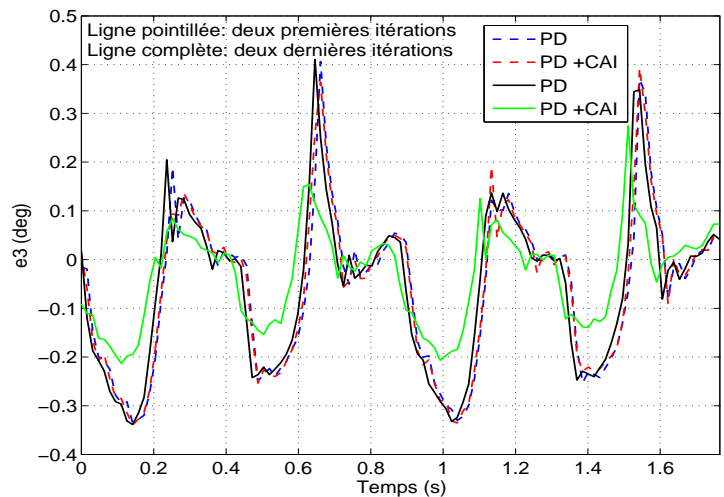
La commande par couple a priori (Feedforward) est basée sur le modèle dynamique du robot où elle utilise la position, la vitesse et l'accélération désirées. Une étude comparative



(a)

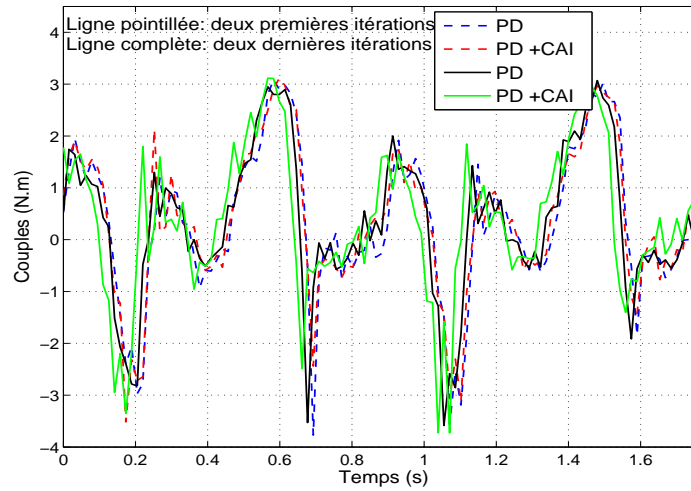


(b)

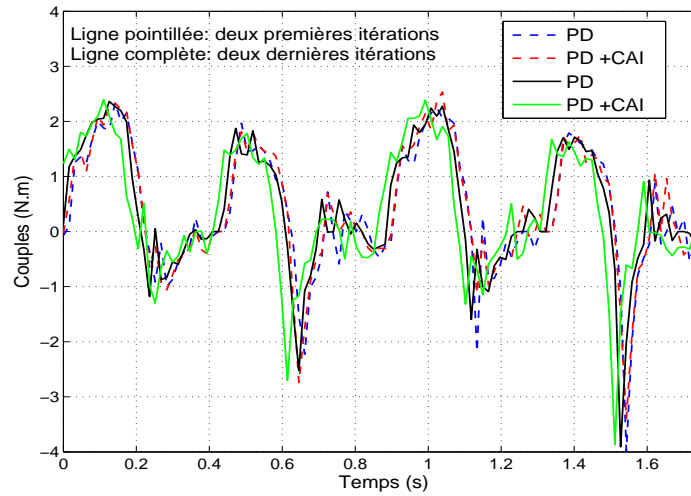


(c)

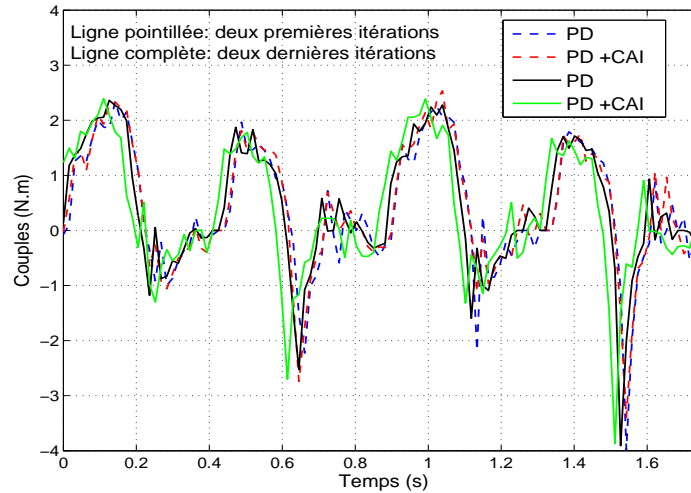
Figure 3.14 – Évolution de l’erreur de poursuite en fonction des itérations. (a) articulation 1, (b) articulation 2, (c) articulation 3



(a)

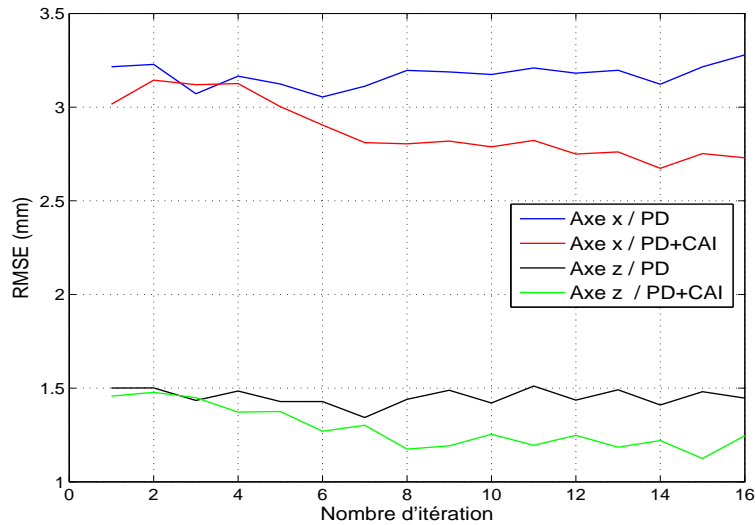


(b)

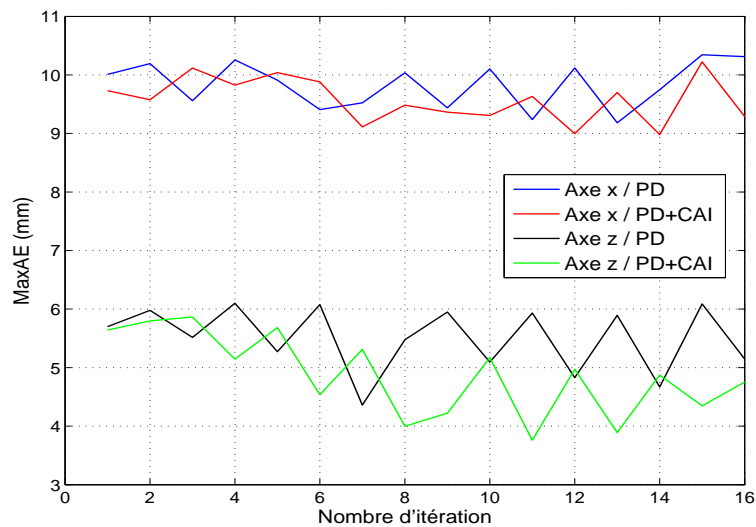


(c)

Figure 3.15 – Couples de commandes expérimentaux (a) articulation 1, (b) articulation 2, (c) articulation



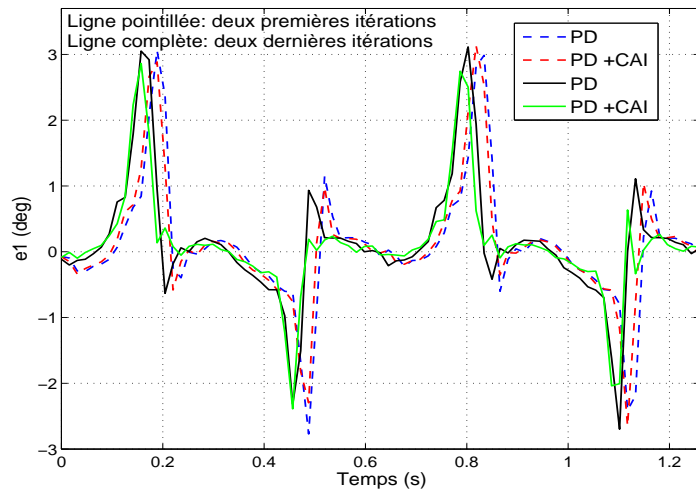
(a)



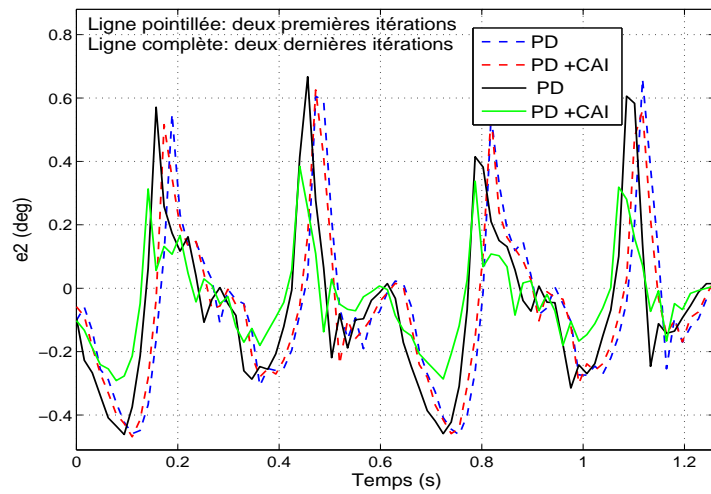
(b)

Figure 3.16 – Évolution des critères en expérimentation suivant les itérations sous une accélération de 2G. (a) RMSE, (b) MaxAE

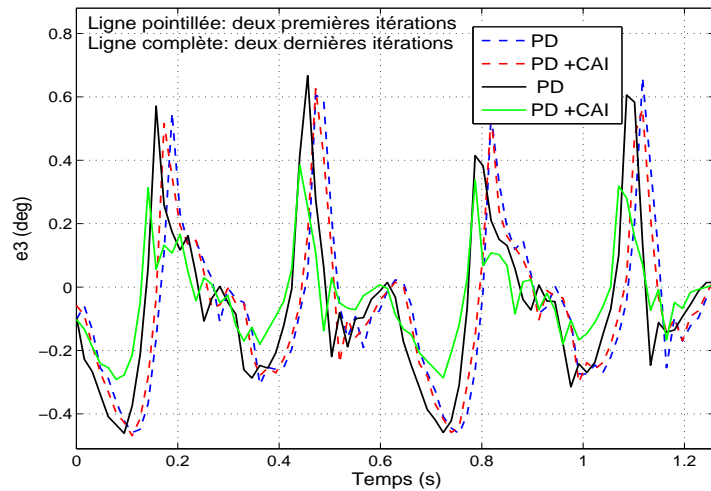
a été faite entre la commande par couple a priori (CCP) et la commande "computed torque" où le robot série MIT à bras d'entraînement direct a été considéré [60]. Les auteurs ont conclu que les deux commandes ont des performances similaires avec un petit avantage pour la CCP représenté par la possibilité de calculer la commande hors ligne. La CCP a été largement étudiée durant les trois dernières décennies, où elle a été mise en œuvre avec succès dans de nombreux robots tels que le bras vertical à deux degrés de liberté avec entraînement direct (Direct-Drive Vertical Arm) [61], le robot série MIT, le bras d'entraînement direct [60] et le robot COMAU [62]. Il a été montré, à travers les résultats expérimentaux, que cette commande permet d'améliorer les performances par rapport aux commandes PD/PID. La stabilité de la CCP a été prouvée dans [61], où l'erreur de poursuite converge sous certaines contraintes. L'inconvénient de la commande par couple a priori réside dans la nécessité de la connaissance exacte du modèle dynamique. Malheureusement, ceci n'est pas toujours évident pour les robots manipulateurs, car, entre autres, les frottements causés par les réducteurs, qui ne sont pas pris en compte dans le modèle dynamique, ont un important impact sur les performances.



(a)

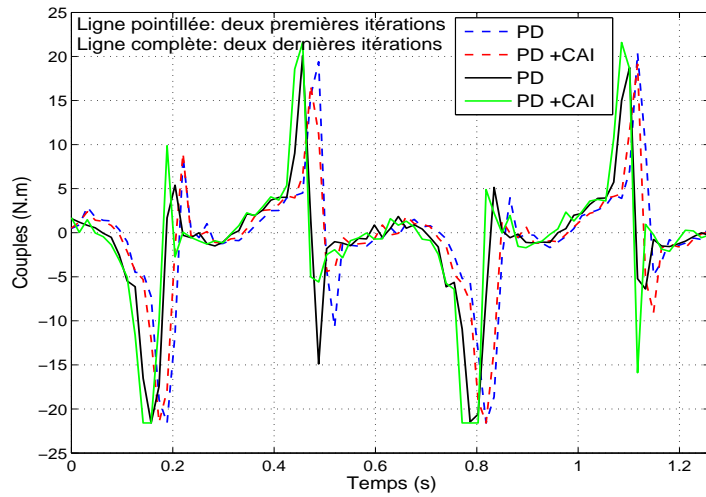


(b)

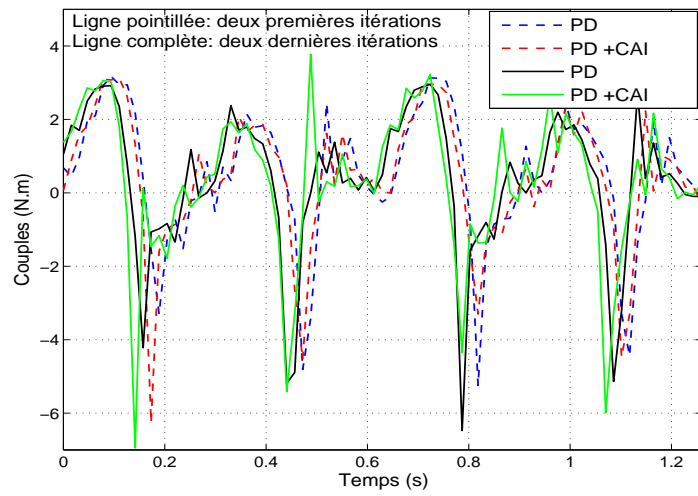


(c)

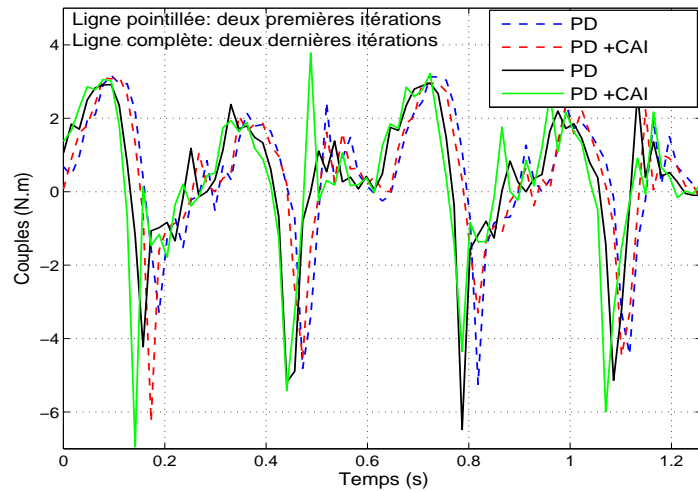
Figure 3.17 – Évolution de l’erreur de poursuite en fonction des itérations. (a) articulation 1, (b) articulation 2, (c) articulation 3



(a)



(b)



(c)

Figure 3.18 – Couples de commandes expérimentaux sous une accélération de 2G. (a) articulation 1, (b) articulation 2, (c) articulation 3

Relation entre la commande CCP et la commande PD plus CAI

Dans cette partie, nous allons expliquer la relation entre la commande CCP et la commande PD plus CAI.

Considérons le modèle dynamique d'un robot où les perturbations externes et les frictions ne sont pas prises en compte.

D'une part, par l'application de la commande PD plus CCP on obtient

$$M(q_k)\ddot{q}_k + C(q_k, \dot{q}_k)\dot{q}_k + G(q_k) = K_p\tilde{q}_k + K_d\dot{\tilde{q}}_k + M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) \quad (3.84)$$

Si les gains du régulateur sont choisis comme indiqués dans [61], alors les erreurs de poursuite en position et en vitesse tendent asymptotiquement vers zéro. Comme résultat le couple va tendre asymptotiquement vers la commande CCP, ie :

$$\tau \rightarrow M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) \text{ quand } t \rightarrow \infty$$

D'autre part, par l'application de la commande PD plus CAI, on trouve

$$M(q_k)\ddot{q}_k + C(q_k, \dot{q}_k)\dot{q}_k + G(q_k) = K_p\tilde{q}_k + K_d\dot{\tilde{q}}_k + u_k \quad (3.85)$$

Si les gains du régulateur sont choisis comme indiqués dans [16], alors l'erreur de poursuite en position et en vitesse tendent asymptotiquement vers zéro. Comme résultat le couple va tendre asymptotiquement vers la commande CAI qui lui même tend asymptotiquement vers la commande CCP, ie :

$$\tau \rightarrow u_k \rightarrow M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + G(q_d) \text{ quand } t \rightarrow \infty$$

En conclusion, la commande CAI converge vers la commande CCP après un certain temps, ce qui signifie que les performances de la CCP seront meilleures que la CAI. Puisque le couple de la CCP commence par le même couple désiré qu'on va obtenir pour la poursuite parfaite, tandis que le couple de CAI démarre par une valeur aléatoire puis converge vers le couple désiré.

Considérons maintenant le modèle dynamique réel d'un robot manipulateur :

$$M(q_k)\ddot{q}_k + C(q_k, \dot{q}_k)\dot{q}_k + G(q_k) + d(t) = \tau_k \quad (3.86)$$

où $d(t)$ représente les perturbations externes et les frictions.

La question qui se pose ici est : est ce que, sous un environnement répétitif, les performances de la CCP restent toujours meilleures que celles de la CAI à travers les itérations, malgré l'influence du réducteur et des perturbations externes sur la dynamique du robot ?

Pour répondre à cette question les expérimentations suivantes ont été réalisées.

3.4.6 Résultats expérimentaux

La figure 3.21 montre la poursuite de trajectoire expérimentale dans l'espace opérationnel sous les commandes PD plus CAI et PD plus CCP. La figure 3.22 représente l'erreur de poursuite pour les huit premières itérations des articulations 1, 2 et 3 respectivement. On constate que, d'une part, l'erreur de poursuite du PD plus CCP est plus petite que le PD plus CAI. D'autre part, l'erreur de poursuite de PD plus CAI diminue avec l'augmentation

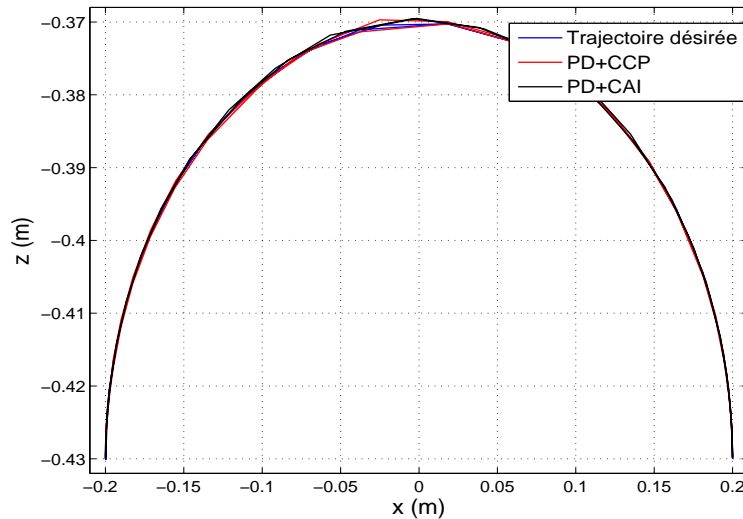


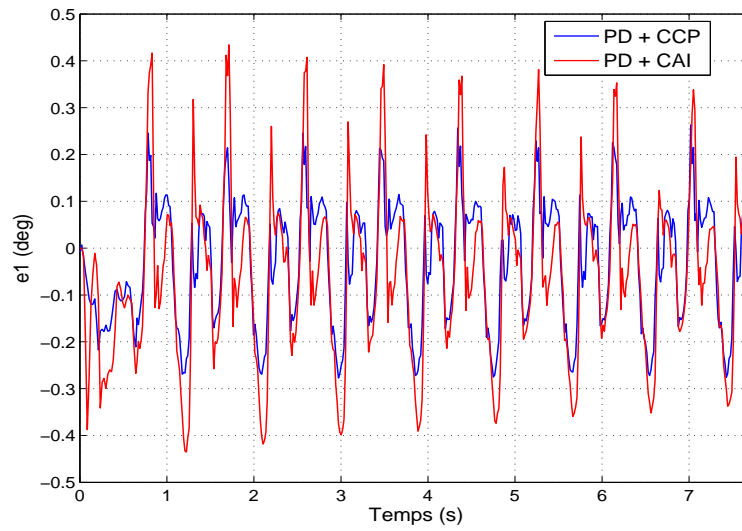
Figure 3.21 – Poursuite de trajectoire dans l’espace opérationnel en expérimentation

Table 3.5 – Performances de poursuite

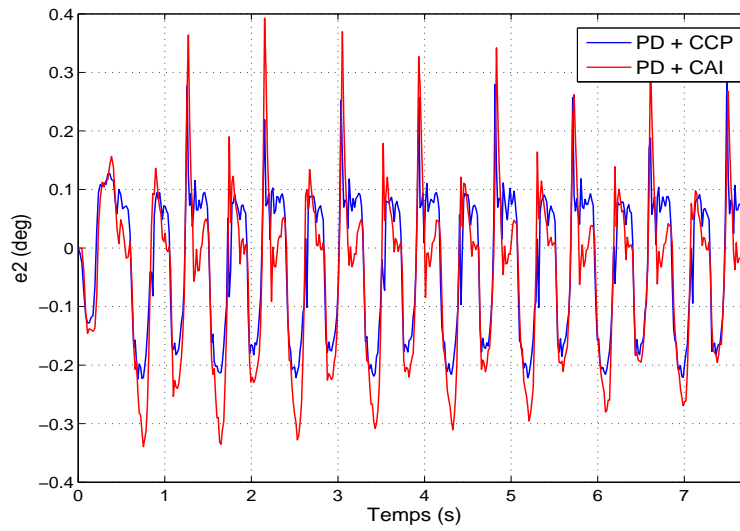
Itération		1	10	30	40	48
RMSE PD+CCP (mm)	x axis	0.49	0.48	0.47	0.48	0.47
	z axis	0.43	0.42	0.42	0.43	0.42
RMSE PD+CAI (mm)	x axis	0.81	0.58	0.39	0.37	0.38
	z axis	0.58	0.41	0.29	0.30	0.31

du nombre d’itérations.

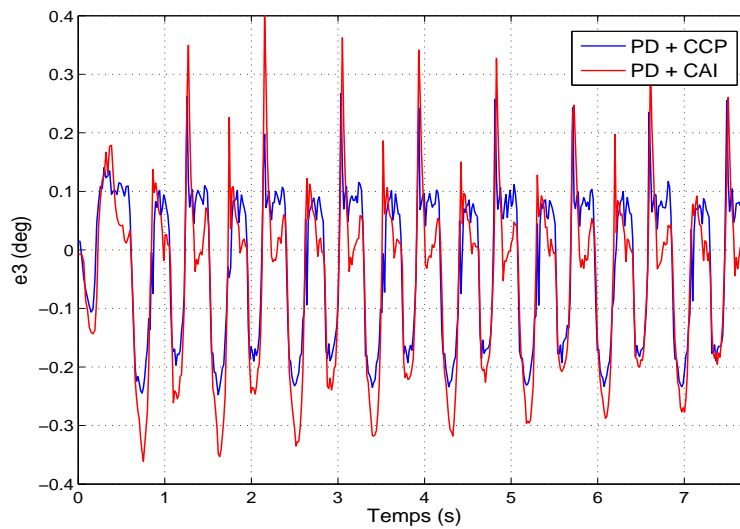
La figure 3.23 montre l’erreur de poursuite des articulations 1, 2 et 3 respectivement pour les itérations 41 à 48. Comme on peut le constater, les performances du PD plus CAI deviennent meilleures que celles du PD plus CCP. Les figures 3.24 et 3.25 indiquent les signaux des commandes générés avec l’augmentation du nombre d’itérations. On constate, que les deux régulateurs ont presque la même variation de l’amplitude pour les huit premières itérations et les huit dernières itérations. La figure 3.26 indique l’évolution du RMSE à travers les itérations. Les performances de poursuite sont résumées dans le tableau 3.5. Il est clair qu’au début, les performances de la CCP sont supérieures à celles de la CAI, cependant, les performances de la CAI deviennent meilleures que celles de CCP avec l’augmentation du nombre des itérations. Ceci peut être expliqué, par l’absence de modèle dynamique exact, où les performances du PD plus CCP sont influencées par les frictions des réducteurs, tandis que les performances de la commande PD plus CAI continuent à s’améliorer à travers les itérations en utilisant uniquement les informations des cycles précédents.



(a)

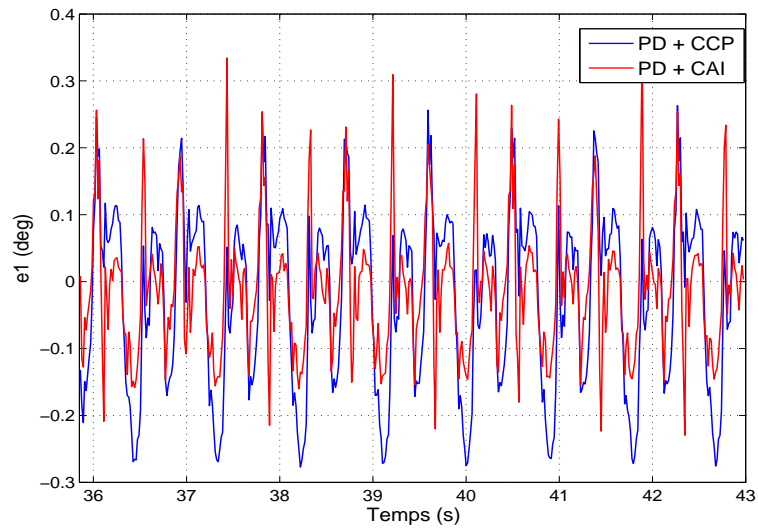


(b)

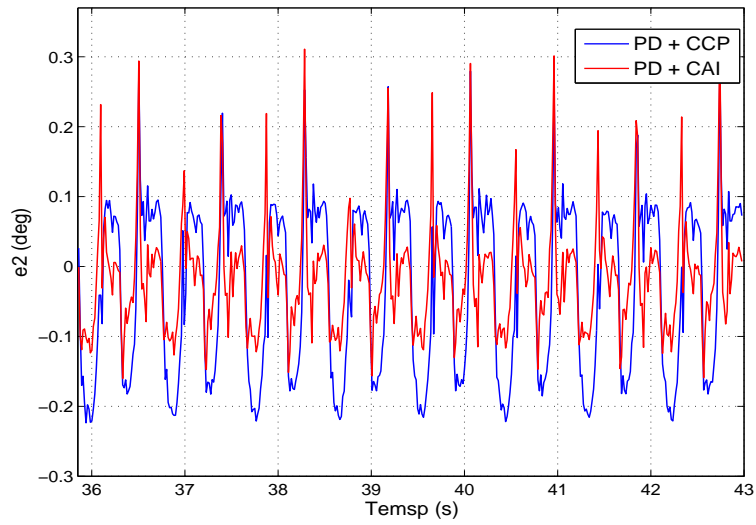


(c)

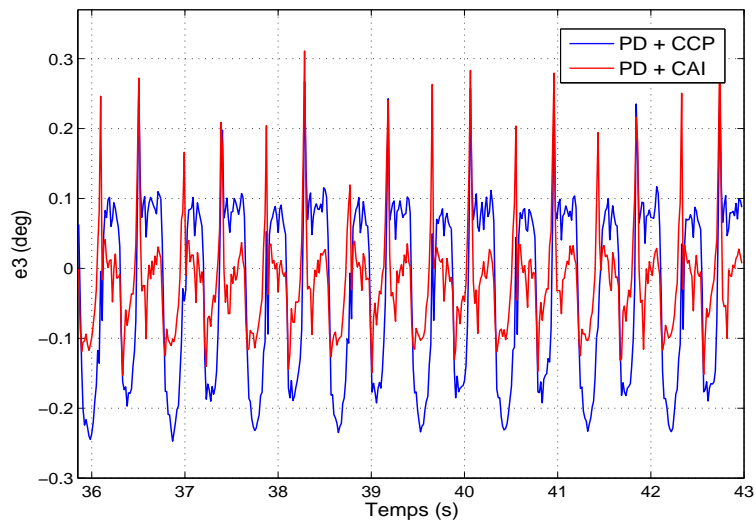
Figure 3.22 – Évolution de l’erreur de poursuite en fonction des itérations $k=1-8$. (a) articulation 1, (b) articulation 2, (c) articulation 3



(a)

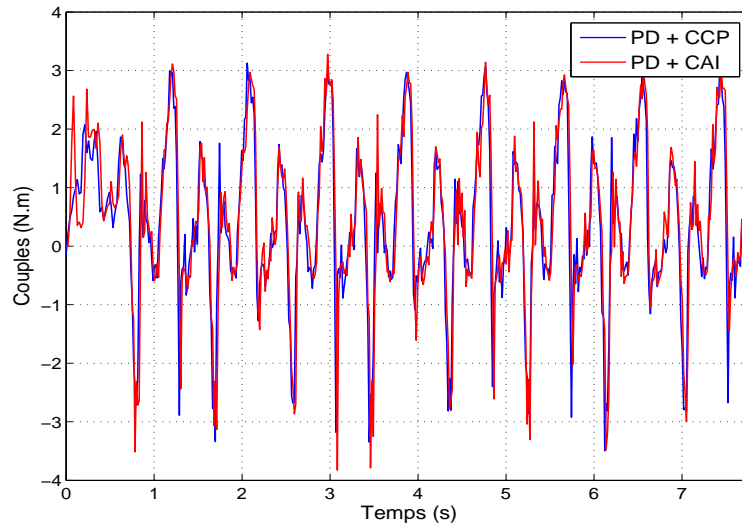


(b)

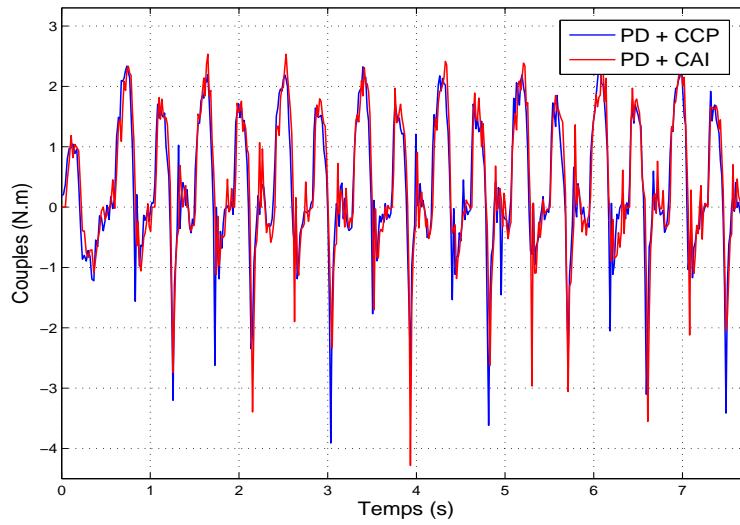


(c)

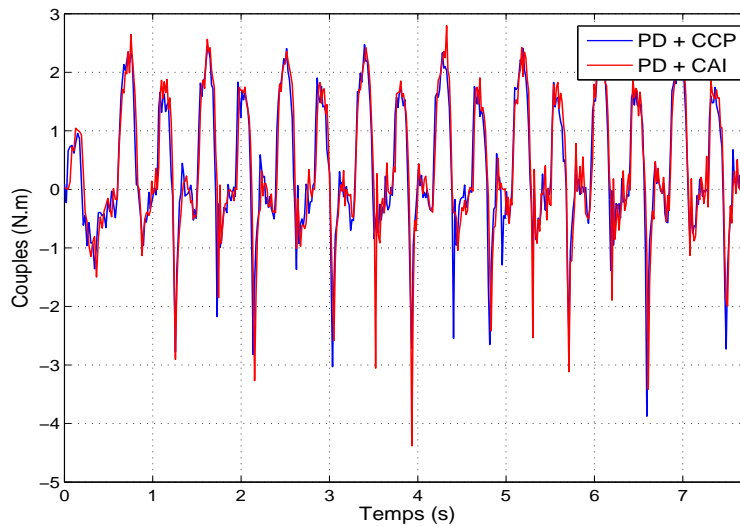
Figure 3.23 – Évolution de l’erreur de poursuite en fonction des itérations $k=41-48$. (a) articulation 1, (b) articulation 2, (c) articulation 3



(a)

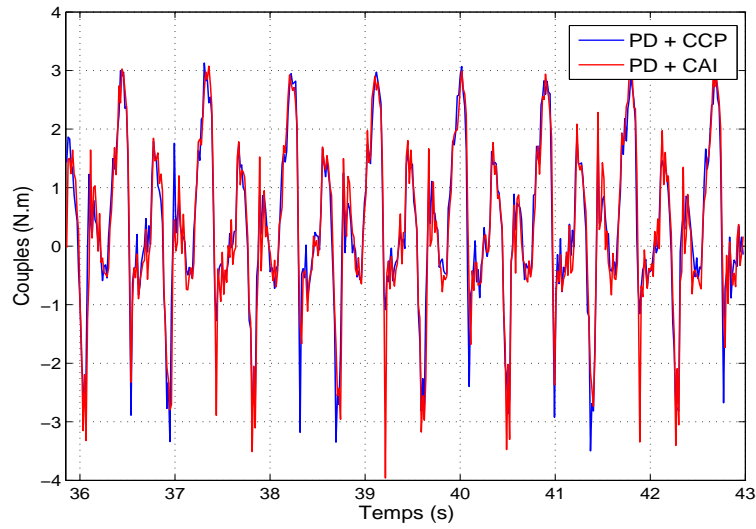


(b)

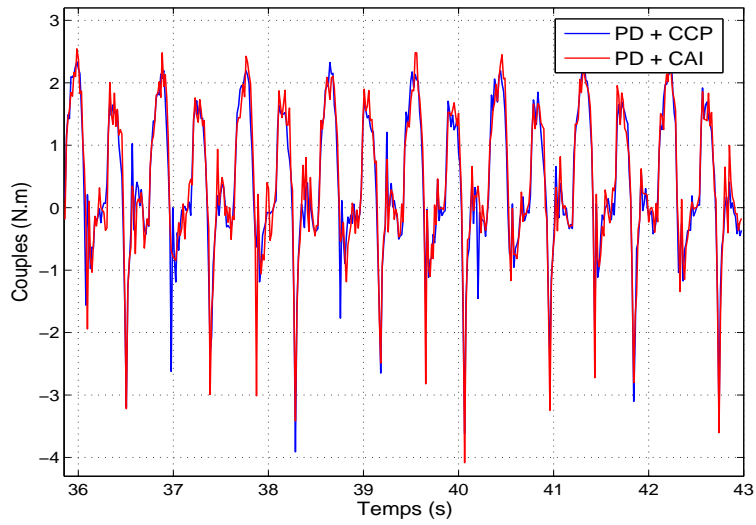


(c)

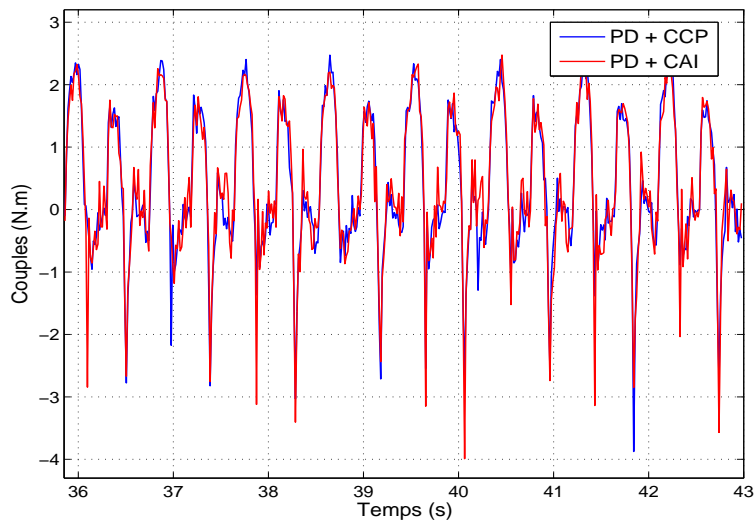
Figure 3.24 – Couples de commandes expérimentaux pour $k=1-8$. (a) articulation 1, (b) articulation 2, (c) articulation 3



(a)



(b)



(c)

Figure 3.25 – Couples de commandes expérimentaux pour $k=41-48$. (a) articulation 1, (b) articulation 2, (c) articulation 3

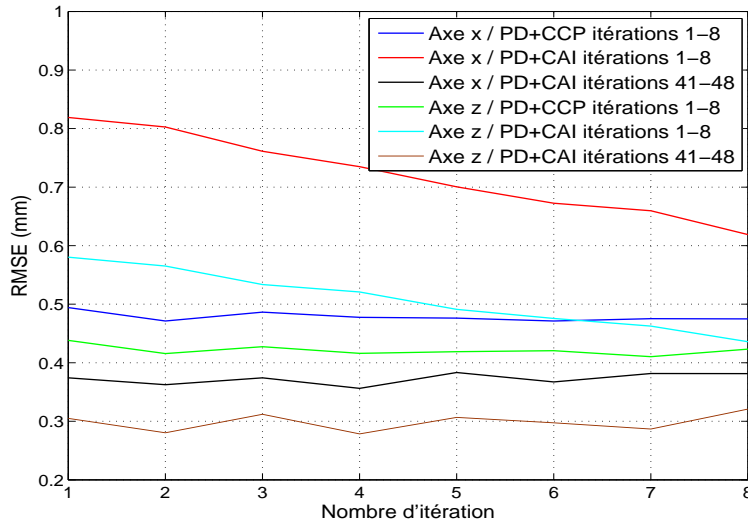


Figure 3.26 – Critère RMSE suivant les itérations

3.5 Commande par apprentissage itératif adaptative

La loi de commande que nous proposons est issue de la loi développée précédemment avec l'adaptation des gains de la partie PD et l'ajout d'un terme discontinu de robustification afin de compenser les incertitudes du modèle et les perturbations externes répétitives et non répétitives.

3.5.1 Conception de la commande

L'expression de la commande est donnée par :

$$\tau_k = K_p(t)\tilde{q}_k + K_d(t)\dot{\tilde{q}}_k + u_k \quad (3.87)$$

$$u_{k+1} = u_k + \Lambda\tilde{q}_k + \Gamma\dot{\tilde{q}}_k + \zeta \operatorname{sgn}(\tilde{z}_k) \quad (3.88)$$

avec

$$K_p(\tilde{q}_k) = \operatorname{diag}\{K_{p_1}(\tilde{q}_{k,1}), \dots, K_{p_n}(\tilde{q}_{k,n})\} \quad (3.89)$$

$$K_d(\dot{\tilde{q}}_k) = \operatorname{diag}\{K_{d_1}(\dot{\tilde{q}}_{k,1}), \dots, K_{d_n}(\dot{\tilde{q}}_{k,n})\} \quad (3.90)$$

où

$$K_{p_i}(\tilde{q}_{k,i}) = \begin{cases} K_{p0}|\tilde{q}_{k,i}|^{\alpha_1-1}, & |\tilde{q}_{k,i}| > \delta_1 \quad i = 1, \dots, n \\ K_{p0}\delta_1^{\alpha_1-1}, & |\tilde{q}_{k,i}| \leq \delta_1 \\ K_{pmax}, & K_{p_i}(\tilde{q}_{k,i}) \geq K_{pmax} \\ K_{pmin}, & K_{p_i}(\tilde{q}_{k,i}) \leq K_{pmin} \end{cases} \quad (3.91)$$

$$K_{d_i}(\dot{\tilde{q}}_{k,i}) = \begin{cases} K_{d0}|\dot{\tilde{q}}_{k,i}|^{\alpha_2-1}, & |\dot{\tilde{q}}_{k,i}| > \delta_2 \quad i = 1, \dots, n \\ K_{d0}\delta_2^{\alpha_2-1}, & |\dot{\tilde{q}}_{k,i}| \leq \delta_2 \\ K_{dmax}, & K_{d_i}(\dot{\tilde{q}}_{k,i}) \geq K_{dmax} \\ K_{dmin}, & K_{d_i}(\dot{\tilde{q}}_{k,i}) \leq K_{dmin} \end{cases} \quad (3.92)$$

L'indice i indique la i -ème ligne d'un vecteur. $\alpha_1 > 1$, $\alpha_2 > 1$, et δ_1 et δ_2 caractérisent l'erreur tolérée en position et en vitesse respectivement. Les valeurs de ces paramètres peuvent être déterminées en utilisant la procédure de réglage proposée dans [63]. Les constantes K_{p0} , K_{pmax} , K_{pmin} , K_{d0} , K_{dmax} , K_{dmin} , Γ , Λ , et ζ sont prises positives. Les vecteurs

z_k et \tilde{z}_k sont définis par (3.55), (3.56).

Remarque 2 : Les gains adaptatifs (3.91) ou (3.92) indiquent qu'un grand gain sera utilisé pour une petite erreur et un petit gain pour une grande erreur (ou sa variation).

Remarque 3 : Un aspect très important de ce régulateur réside dans sa facilité d'implémentation et d'intégration dans les robots industriels, où il utilise seulement les erreurs de poursuite et ses dérivées par rapport au temps.

Le schéma de la CAI adaptative robuste est illustré par la figure 3.27.

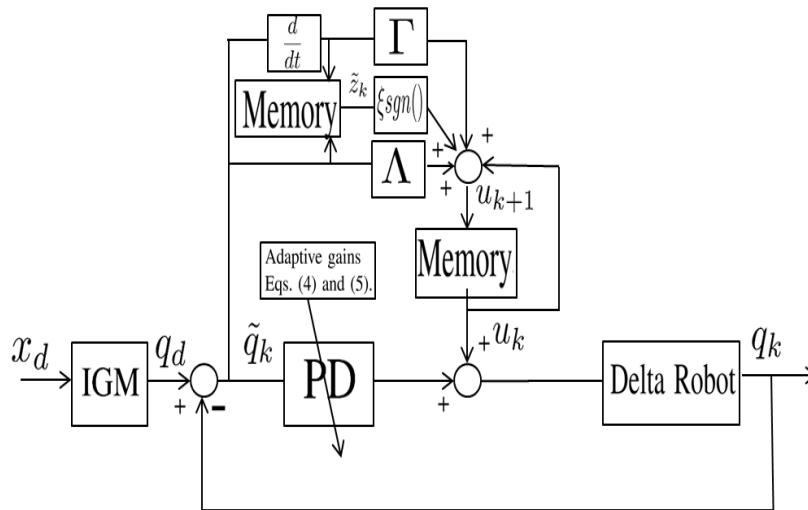


Figure 3.27 – Schéma de la commande par apprentissage itératif adaptative

3.5.2 Analyse de la convergence

Nous allons introduire le vecteur τ_{d2_k} qui contient les perturbations externes non répétitives, contrairement à τ_{d1_k} qui contient les perturbations externes répétitives.

Le modèle dynamique du robot est donné comme suit :

$$M_n(q_k)\ddot{q}_k + C_n(q_k, \dot{q}_k)\dot{q}_k + G_n(q_k) + \tau_{d1_k} + \tau_{d2_k} = \tau_k \quad (3.93)$$

Avec le développement de Taylor, l'équation (3.93) peut être linéarisé au tour la trajectoire désirée $(q_d, \dot{q}_d, \ddot{q}_d)$, comme suit :

$$\begin{aligned} M(t)\ddot{q}_k(t) + [C(t) + C_1(t)]\dot{q}_k(t) + F(t)\tilde{q}_k(t) + n(\ddot{q}_k, \dot{q}_k, \tilde{q}_k, t) \\ = H(t) - \tau_k(t) + \tau_{d1_k} + \tau_{d2_k} \end{aligned} \quad (3.94)$$

où $M(t) = M(q_d(t))$

$$C(t) = C(q_d(t), \dot{q}_d(t))$$

$$C_1(t) = \frac{\partial C}{\partial \dot{q}_k} \Big|_{q_d(t), \dot{q}_d(t)} \dot{q}_k(t)$$

$$F(t) = \frac{\partial M}{\partial q_k} \Big|_{q_d(t)} \ddot{q}_k(t) + \frac{\partial C}{\partial q_k} \Big|_{q_d(t), \dot{q}_d(t)} \dot{q}_k(t) + \frac{\partial G}{\partial q_k} \Big|_{q_d(t)}$$

$$H(t) = M(q_d(t))\ddot{q}_d(t) + C(q_d(t), \dot{q}_d(t))\dot{q}_d(t) + G(q_d(t))$$

Le terme $n_k(\ddot{q}_k, \dot{q}_k, q_k, t)$ contient les contributions d'ordre supérieur de \ddot{q}_k , \dot{q}_k , et q_k .

On pose $d_k(t) = \tau_{d1} + \tau_{d2_k} - n_k(t)$.

Dans la k -ème itération, l'équation (3.94) peut se réécrire comme suit

$$\begin{aligned} M(t)\ddot{q}_k(t) + [C(t) + C_1(t)]\dot{q}_k(t) + F(t)q_k(t) \\ = H(t) - \tau_k(t) + d_k(t) \end{aligned} \quad (3.95)$$

De même, dans la $(k+1)$ -ème itération, nous avons

$$\begin{aligned} M(t)\ddot{q}_{k+1}(t) + [C(t) + C_1(t)]\dot{q}_{k+1}(t) + F(t)q_{k+1}(t) \\ = H(t) - \tau_{k+1}(t) + d_{k+1}(t) \end{aligned} \quad (3.96)$$

Nous supposons que

$$\Lambda = \lambda\Gamma \quad (3.97)$$

Théorème 3 : Considérons la dynamique du robot (3.93) sous la condition d'alignement (4), les hypothèses (2,5) et les propriétés (P1) et (P6), alors, la loi de commande donnée par (3.87) garantit la convergence de l'erreur en position et en vitesse lorsque le nombre d'itérations k tend vers l'infini sur un intervalle de temps fini $[0, T]$, i.e., $\lim_{k \rightarrow \infty} \tilde{q}_k(t) = \lim_{k \rightarrow \infty} \dot{\tilde{q}}_k(t) = 0 \forall t \in [0, T]$, si les gains sont sélectionnés pour tous $t \in [0, T]$ comme suit

$$v_{p1} = \lambda_{\min}(2C_1 - \Gamma - 2M\lambda + 2K_{d\min}I) > 0 \quad (3.98)$$

$$v_{p2} = \lambda_{\min}\left(2\frac{F}{\lambda} + 2\frac{K_{p\min}I}{\lambda} - \Gamma + 2C - 2\frac{\dot{C}_1}{\lambda}\right) > 0 \quad (3.99)$$

$$v_{p1}v_{p2} \geq \left\| \frac{F}{\lambda} + \frac{K_p}{\lambda} - (C + C_1 - K_d - M\lambda) \right\|_{\max}^2 \quad (3.100)$$

$$\xi - \|d_{k+1} - d_k\| \geq \gamma \quad (3.101)$$

où, γ est une constante positive.

Preuve

Soit la fonction de Lyapunov suivante

$$V_k(t) = \int_0^t z_k^T \Gamma z_k d\sigma \quad (3.102)$$

La variation de $V_k(t)$ et δq_k à travers les itérations peut être calculée comme suit

$$\Delta V_k = V_{k+1} - V_k \quad \text{et} \quad \delta q_k = \tilde{q}_{k+1} - \tilde{q}_k \quad (3.103)$$

avec

$$\begin{aligned} \Delta V_k(t) &= \int_0^t z_{k+1}^T \Gamma z_{k+1} d\sigma - \int_0^t z_k^T \Gamma z_k d\sigma \\ &= \int_0^t \tilde{z}_k^T \Gamma \tilde{z}_k + 2\tilde{z}_k^T \Gamma z_k d\sigma \end{aligned} \quad (3.104)$$

Soustrayant (3.96) de (3.95), on obtient

$$\begin{aligned} M(t)(\ddot{q}_{k+1} - \ddot{q}_k) + [C(t) + C_1(t)](\dot{q}_{k+1} - \dot{q}_k) \\ + F(t)(q_{k+1} - q_k) = -K_p(q_{k+1} - q_k) - K_d(\dot{q}_{k+1} - \dot{q}_k) \\ - \Lambda \tilde{q}_k - \Gamma \dot{\tilde{q}}_k - \xi \text{sgn}(\tilde{z}_k) + d_{k+1} - d_k \end{aligned} \quad (3.105)$$

À partir de (3.55), (3.97) et (3.103) on trouve

$$\begin{aligned} M(t)\dot{\tilde{z}}_k + [C + C_1 - M\lambda + K_d]\tilde{z}_k \\ + [F + K_p - (C + C_1 + K_k - M\lambda)\lambda]\delta q_k + \zeta \operatorname{sgn}(\tilde{z}_k) \\ - (d_{k+1} - d_k) = -\Gamma z_k \end{aligned} \quad (3.106)$$

En utilisant (3.104) et (3.106) on obtient

$$\begin{aligned} \Delta V_k(t) = \int_0^t \tilde{z}_k^T \Gamma \tilde{z}_k d\sigma - 2 \int_0^t \tilde{z}_k^T M(\sigma) \dot{\tilde{z}}_k d\sigma \\ - 2 \int_0^t \tilde{z}_k^T [C + C_1 - M\lambda + K_d] \tilde{z}_k d\sigma \\ - 2 \int_0^t \tilde{z}_k^T [F + K_p - (C + C_1 + K_d - M\lambda)\lambda] \delta q_k d\sigma \\ + 2 \int_0^t \tilde{z}_k^T [d_{k+1} - d_k - \zeta \operatorname{sgn}(\tilde{z}_k)] d\sigma \end{aligned} \quad (3.107)$$

En appliquant l'intégration par partie

$$\begin{aligned} \int_0^t \tilde{z}_k^T M(\sigma) \dot{\tilde{z}}_k d\sigma &= \tilde{z}_k^T M \tilde{z}_k \Big|_0^t - \int_0^t (\tilde{z}_k M)' \tilde{z}_k d\sigma \\ &= \tilde{z}_k^T M \tilde{z}_k \Big|_0^t - \int_0^t \tilde{z}_k^T M \dot{\tilde{z}}_k - \int_0^t \tilde{z}_k^T \dot{M} \tilde{z}_k d\sigma \end{aligned} \quad (3.108)$$

La propriété (P6) assure que

$$2 \int_0^t \tilde{z}_k^T M(\sigma) \dot{\tilde{z}}_k d\sigma = \tilde{z}_k^T(t) M(t) \tilde{z}_k(t) - \tilde{z}_k^T(0) M(0) \tilde{z}_k(0) - 2 \int_0^t \tilde{z}_k^T C \tilde{z}_k d\sigma$$

ainsi

$$\begin{aligned} \Delta V_k(t) = -\tilde{z}_k^T(t) M(t) \tilde{z}_k(t) + \tilde{z}_k^T(0) M(0) \tilde{z}_k(0) \\ - \int_0^t \tilde{z}_k^T (2C_1 - \Gamma - 2M\lambda + 2K_d) \tilde{z}_k d\sigma \\ - 2 \int_0^t \tilde{z}_k^T (F + K_p - (C + C_1 + K_d - M\lambda)) \delta q_k d\sigma \\ + 2 \int_0^t \tilde{z}_k^T [d_{k+1} - d_k - \zeta \operatorname{sgn}(\tilde{z}_k)] d\sigma \end{aligned} \quad (3.109)$$

La substitution de $\tilde{z}_k = \delta \dot{q}_k + \lambda \delta q_k$ dans (3.109) conduit à

$$\begin{aligned} \Delta V_k(t) = -\tilde{z}_k^T(t) M(t) \tilde{z}_k(t) + \tilde{z}_k^T(0) M(0) \tilde{z}_k(0) \\ - \int_0^t \delta \dot{q}_k^T (2C_1 - \Gamma - 2M\lambda + 2K_d) \delta \dot{q}_k d\sigma \\ - 2\lambda \int_0^t \delta \dot{q}_k^T (2C_1 - \Gamma - 2M\lambda) \delta \dot{q}_k d\sigma \\ - \lambda^2 \int_0^t \delta \dot{q}_k^T (2C_1 - \Gamma - 2M\lambda + 2K_d) \delta q_k d\sigma \\ - 2 \int_0^t \delta \dot{q}_k^T (F + K_p - (C + C_1 - K_d - M\lambda)\lambda) \delta q_k d\sigma \\ - 2\lambda \int_0^t \delta \dot{q}_k^T (F + K_p - (C + C_1 + K_d - M\lambda)\lambda) \delta q_k d\sigma \\ + 2 \int_0^t \tilde{z}_k^T [d_{k+1} - d_k - \zeta \operatorname{sgn}(\tilde{z}_k)] d\sigma \end{aligned} \quad (3.110)$$

Après une autre intégration par partie, on obtient

$$\begin{aligned} \int_0^t \delta \dot{q}_k^T (2C_1 - \Gamma - 2M\lambda) \delta \dot{q}_k d\sigma = \\ 2 \int_0^t \delta \dot{q}_k^T (\dot{M}\lambda - \dot{C}_1) \delta q_k d\sigma + \delta \dot{q}_k^T (2C_1 - \Gamma - 2M\lambda) \delta q_k \Big|_0^t \\ - \int_0^t \delta \dot{q}_k^T (2C_1 - \Gamma - 2M\lambda) \delta \dot{q}_k d\sigma \end{aligned} \quad (3.111)$$

Comme $\tilde{z}_k^T (d_{k+1} - d_k) \leq \|\tilde{z}_k^T\| \|d_{k+1} - d_k\|$, et de (3.101), on trouve

$$\begin{aligned} \int_0^t \tilde{z}_k^T (d_{k+1} - d_k - \zeta \operatorname{sgn}(\tilde{z}_k)) d\sigma &\leq \int_0^t \|\tilde{z}_k^T\| (\|d_{k+1} - d_k\| - \zeta) d\sigma \\ &\leq - \int_0^t \gamma \|\tilde{z}_k^T\| d\sigma \end{aligned} \quad (3.112)$$

La substitution de (3.111) et (3.112) dans (3.110) nous donne

$$\begin{aligned} \Delta V_k(t) = & -\tilde{z}_k^T(t)M(t)\tilde{z}_k(t) + \tilde{z}_k^T(0)M(0)\tilde{z}_k(0) \\ & -\lambda\delta q_k^T(t)(2C_1 - \Gamma - 2M\lambda)\delta q_k(t) \\ & +\lambda\delta q_k^T(0)(2C_1 - \Gamma - 2M\lambda)\delta q_k(0) - \int_0^t W_k d\sigma \end{aligned} \quad (3.113)$$

où

$$\begin{aligned} W_k = & \delta\dot{q}_k^T(2C_1 - \Gamma - 2M\lambda + 2K_d)\delta\dot{q}_k + \\ & 2\lambda\delta\dot{q}_k^T\left(\frac{F}{\lambda} + \frac{K_p}{\lambda} - (C + C_1 - K_d - M\lambda)\right)\delta q_k + \\ & \lambda^2\delta q_k^T\left(2\frac{F}{\lambda} + 2\frac{K_p}{\lambda} - \Gamma + 2C - 2\frac{C_1}{\lambda}\right)\delta q_k + 2\gamma\|\tilde{z}_k^T\| \end{aligned} \quad (3.114)$$

prenons $H = \frac{F}{\lambda} + \frac{K_p}{\lambda} - (C + C_1 - K_d - M\lambda)$, de (3.98) et (3.99) on trouve

$$W_k \geq v_{p1}\|\delta\dot{q}_k\|^2 + 2\lambda\delta\dot{q}_k^T H\delta q_k + \lambda^2 v_{p2}\|\delta q_k\|^2 + 2\gamma\|\tilde{z}_k^T\| \quad (3.115)$$

L'inégalité de Cauchy-Schwartz conduit à

$$\delta\dot{q}_k^T H\delta q_k \geq -\|\delta\dot{q}_k\|\|H\|_{max}\|\delta q_k\| \quad (3.116)$$

À partir de (3.100), (3.115) et (3.116), on trouve

$$\begin{aligned} W_k \geq & v_{p1}\|\delta\dot{q}_k\|^2 - 2\lambda\|\delta\dot{q}_k\|\|H\|_{max}\|\delta q_k\| + \lambda^2 v_{p2}\|\delta q_k\|^2 \\ & + 2\gamma\|\tilde{z}_k^T\| = v_{p1}\left(\|\delta\dot{q}_k\| - \frac{\lambda}{v_{p1}}\|H\|_{max}\|\delta q_k\|\right)^2 \\ & + \lambda^2\left(v_{p2} - \frac{1}{v_{p1}}\|H\|_{max}^2\right)\|\delta q_k\|^2 + 2\gamma\|\tilde{z}_k^T\| \geq 0 \end{aligned} \quad (3.117)$$

prenons

$$\begin{aligned} w_k = & v_{p1}\left(\|\delta\dot{q}_k\| - \frac{\lambda}{v_{p1}}\|H\|_{max}\|\delta q_k\|\right)^2 \\ & + \lambda^2\left(v_{p2} - \frac{1}{v_{p1}}\|H\|_{max}^2\right)\|\delta q_k\|^2 + 2\gamma\|\tilde{z}_k^T\| \end{aligned} \quad (3.118)$$

De là, on peut avoir

$$\begin{aligned} \Delta V_k(t) \leq & -\tilde{z}_k^T(t)M(t)\tilde{z}_k(t) + \tilde{z}_k^T(0)M(0)\tilde{z}_k(0) \\ & -\lambda\delta q_k^T(t)(2C_1 - \Gamma - 2M\lambda)\delta q_k(t) \\ & +\lambda\delta q_k^T(0)(2C_1 - \Gamma - 2M\lambda)\delta q_k(0) \\ & - \int_0^t w_k d\sigma \end{aligned} \quad (3.119)$$

Sachant que

$$V_{k+1}(t) = V_1(t) + \sum_{i=1}^k \Delta V_i(t) \quad (3.120)$$

Ainsi, la substitution (3.119) dans (3.120) conduit à

$$\begin{aligned} V_{k+1}(t) \leq & V_1(t) + \sum_{i=1}^k [-\tilde{z}_i^T(t)M(t)\tilde{z}_i(t) + \tilde{z}_i^T(0)M(0)\tilde{z}_i(0)] + \\ & \sum_{i=1}^k [-\lambda\delta q_i^T(t)(2C_1 - \Gamma - 2M\lambda)\delta q_i(t) \\ & +\lambda\delta q_i^T(0)(2C_1 - \Gamma - 2M\lambda)\delta q_i(0)] \\ & - \sum_{i=1}^k \int_0^t w_i d\sigma \end{aligned} \quad (3.121)$$

Les hypothèses (2, 4, 5) assurent que

$$\begin{aligned} \tilde{q}_k(0) = \tilde{q}_{k-1}(T) &\implies \tilde{z}_k(0) = \tilde{z}_{k-1}(T), \delta q_k(0) = \delta q_{k-1}(T) \\ \dot{\tilde{q}}_k(0) = \dot{\tilde{q}}_{k-1}(T) &\implies \dot{\tilde{z}}_k(0) = \dot{\tilde{z}}_{k-1}(T), \delta \dot{q}_k(0) = \delta \dot{q}_{k-1}(T) \end{aligned} \quad (3.122)$$

et

$$q_d(T) = q_d(0) \implies \begin{aligned} M(0) &= M(T) \\ (2C_1 - \Gamma - 2M\lambda)(0) &= (2C_1 - \Gamma - 2M\lambda)(T) \end{aligned} \quad (3.123)$$

À partir de (3.122) et (3.123) on obtient

$$\begin{aligned} \tilde{z}_{i-1}^T(T)M(T)\tilde{z}_{i-1}(T) &= \tilde{z}_i^T(0)M(0)\tilde{z}_i(0) \\ \delta q_{i-1}^T(T)(2C_1 - \Gamma - 2M\lambda)(T)\delta q_{i-1}(T) &= \\ \delta q_i^T(0)(2C_1 - \Gamma - 2M\lambda)(0)\delta q_i(0) &\quad i = 2 : k \end{aligned} \quad (3.124)$$

En substituant (3.124) dans (3.121), on aura

$$\begin{aligned} V_{k+1}(T) \leq & V_1(T) - \tilde{z}_k^T(T)M(T)\tilde{z}_k(T) + \tilde{z}_1^T(0)M(0)\tilde{z}_1(0) \\ & - \lambda \delta q_k^T(T)(2C_1 - \Gamma - 2M\lambda)(T)\delta q_k(T) \\ & + \lambda \delta q_1^T(0)(2C_1 - \Gamma - 2M\lambda)(0)\delta q_1(0) - \sum_{i=1}^k \int_0^T w_i(\sigma) d\sigma \end{aligned} \quad (3.125)$$

donc

$$\begin{aligned} V_{k+1}(T) \leq & V_1(T) - \tilde{z}_k^T(T)M(T)\tilde{z}_k(T) - \lambda \delta q_k^T(T)v_{p1}\delta q_k(T) \\ & + \lambda \delta q_1^T(0)(2C_1 - \Gamma - 2M\lambda)(0)\delta q_1(0) \\ & + \tilde{z}_1^T(0)M(0)\tilde{z}_1(0) - \sum_{i=1}^k \int_0^T w_i(\sigma) d\sigma \end{aligned} \quad (3.126)$$

De là, on trouve

$$\begin{aligned} V_{k+1}(T) + \tilde{z}_k^T(T)M(T)\tilde{z}_k(T) + \lambda \delta q_k^T(T)v_{p1}\delta q_k(T) \leq & V_1(T) \\ + \tilde{z}_1^T(0)M(0)\tilde{z}_1(0) + \lambda \delta q_1^T(0)(2C_1 - \Gamma - 2M\lambda)(0)\delta q_1(0) & \\ - \sum_{i=1}^k \int_0^T w_i(\sigma) d\sigma & \end{aligned} \quad (3.127)$$

Le terme du côté droit de (3.127)

$$V_1(T) + \tilde{z}_1^T(0)M(0)\tilde{z}_1(0) + \lambda \delta q_1^T(0)(2C_1 - \Gamma - 2M\lambda)(0)\delta q_1(0)$$

est borné car nous avons

$$\tilde{z}_1(0) = z_2(0) - z_1(0) = \dot{\tilde{q}}_2(0) - \dot{\tilde{q}}_1(0) + \lambda \tilde{q}_2(0) - \lambda \tilde{q}_1(0),$$

où, $\tilde{q}_2(0) = \tilde{q}_1(T)$, $\dot{\tilde{q}}_2(0) = \dot{\tilde{q}}_1(T)$, par conséquent, $\tilde{z}_1(0)$ est en fonction de la première erreur de poursuite en position et en vitesse ($\tilde{q}_1, \dot{\tilde{q}}_1$), de même pour $\delta q_1(0)$. Par conséquent, le terme à droite est en fonction de $(\tilde{q}_1, \dot{\tilde{q}}_1)$. Selon la loi de commande donnée par (3.87), seul le régulateur PD fonctionne à la première itération, et il est connu que les gains positifs PD peuvent stabiliser le robot et conduire à une erreur statique non nulle, ce qui signifie que $(\tilde{q}_1, \dot{\tilde{q}}_1)$, sont bornés, ainsi le terme sur le côté droit est borné.

Comme le terme de gauche de l'équation (3.127) est positif, on peut avoir le résultat suivante :

$$\int_0^T w_k(\sigma) d\sigma \rightarrow 0 \quad \text{quand } k \rightarrow \infty, \forall t \in [0, T]$$

À partir de l'équation (3.118) on peut conclure que

$$\|\delta\dot{q}_k\| \rightarrow 0, \|\delta q_k\| \rightarrow 0 \quad k \rightarrow \infty$$

et puisque \tilde{q}_{k+1} et \tilde{q}_k sont deux vecteurs indépendants on trouve finalement :

$$\lim_{k \rightarrow \infty} \tilde{q}_k(t) = \lim_{k \rightarrow \infty} \dot{\tilde{q}}_k = 0, \forall t \in [0, T]. \quad (3.128)$$

Remarque 4 : Il faut noter que la fonction "signe" utilisée dans la commande proposée (3.88) pourrait conduire au phénomène de broutement. Afin de réduire ce phénomène dans les applications pratiques, la fonction saturation peut être utilisée au lieu de la fonction signe. En conséquence, l'erreur de poursuite converge vers un domaine autour de zéro avec un signal de commande plus lisse.

Remarque 5 : Par rapport aux travaux existants relatifs à l'application de la CAI utilisant des trajectoires répétitives, le régulateur proposé est plus général et peut couvrir la plupart de ces travaux [5, 7, 9, 11, 18, 19]. Où d'une part, il peut être appliqué aux systèmes non linéaires SISO/MIMO qui font l'objet des perturbations externes non répétitives et des incertitudes du modèle. D'autre part, il peut travailler efficacement sous l'hypothèse pratique d'alignement.

3.5.3 Résultats expérimentaux

Cette loi de commande a été mise en œuvre pratiquement sur le robot Delta ISIR88 disponible au laboratoire LCP (voir figure 3.11).

Les gains de régulateur sont choisis comme suit : $K_{p0} = \text{diag}\{2.2, 2.2, 2.2\}$, $K_{d0} = \text{diag}\{0.0145, 0.0145, 0.0145\}$, $\alpha_1 = 0.4$, $\alpha_2 = 0.6$, $\delta_1 = 0.2$, $\delta_2 = 0.7$, $\Lambda = \text{diag}\{0.074, 0.074, 0.074\}$ et $\Gamma = \text{diag}\{0.0015, 0.0015, 0.0015\}$. La valeur de u_1 a été définie comme suit : $u_1 = [0, 0, 0]^T$. Nous donnons l'étude expérimentale dans deux cas.

Cas 1 : La trajectoire désirée démarre à partir de la position initiale (0.20, 0, -0.43)m pour atteindre la position finale (-0.20, 0, -0.43)m et retourne vers la position initiale pendant 0.88 secondes. Les résultats expérimentaux sont comparés avec ceux obtenus par la commande PD plus CAI type PD.

la loi de commande donnée par (3.88) est implantée et les résultats sont comparés avec la commande PD et la commande PD plus CAI type PD, afin de montrer l'efficacité et les performances de poursuite de l'approche proposée.

La figure 3.28 montre la poursuite de trajectoire dans l'espace opérationnel sous la commande proposée. La figure 3.29 présente l'évolution du RMSE avec l'augmentation du nombre des itérations sous la commande PD, la commande PD plus CAI type PD, et la CAI proposée. On peut observer que le RMSE diminue dans les deux approches de la CAI, par exemple, le RMSE de l'axe x diminue de 0,81 mm à 0,49 mm avec la CAI classique et diminue de 0,49 mm à 0,29 mm avec la CAI proposée. D'autre part, le RMSE du régulateur PD a une valeur fixe autour de 0,82 mm. Les figures 3.30 et 3.31 montrent les erreurs de poursuite et les signaux du couple pour les articulations 1, 2 et 3 respectivement. On constate, que l'erreur de poursuite des trois articulations diminue à travers les itérations pour les deux approches de CAI. En outre, avec la CAI proposée les erreurs de poursuite convergent plus rapidement que la commande PD plus CAI type PD. De plus, les couples ont presque la même variation et la même amplitude.

Cas 2 : Afin de montrer l'efficacité de la commande proposée avec des vitesses d'exécution élevées, une expérience similaire a été effectuée, où la même trajectoire désirée sera exécutée cette fois en 0.72 second avec une accélération maximale de $15m/s^2$.

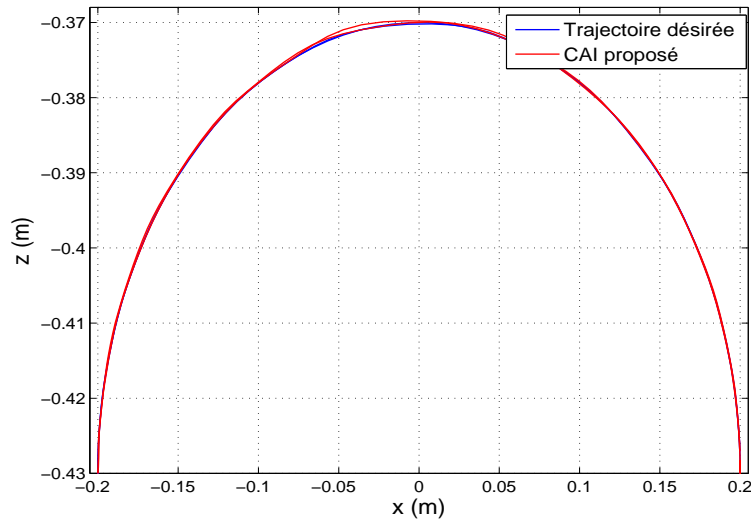


Figure 3.28 – Poursuite de trajectoire dans l'espace opérationnel en expérimentation

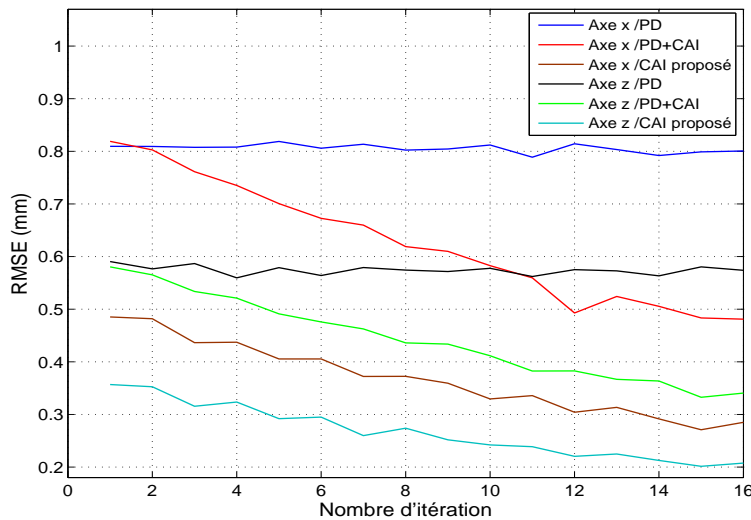
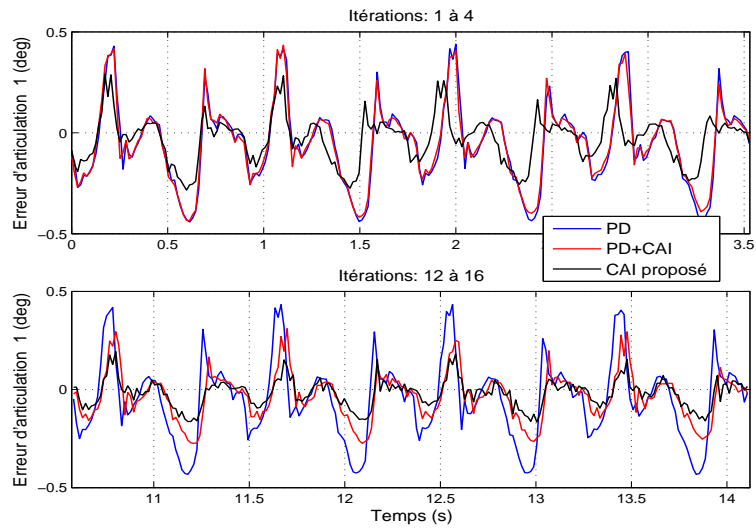
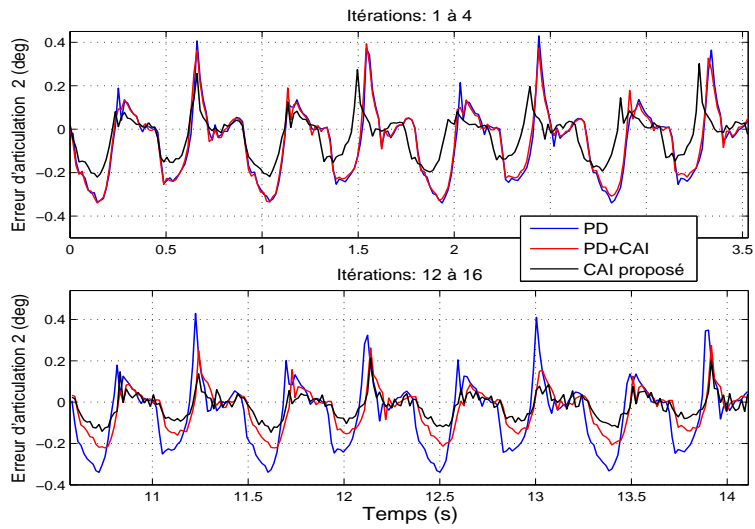


Figure 3.29 – Critère RMSE suivant les itérations

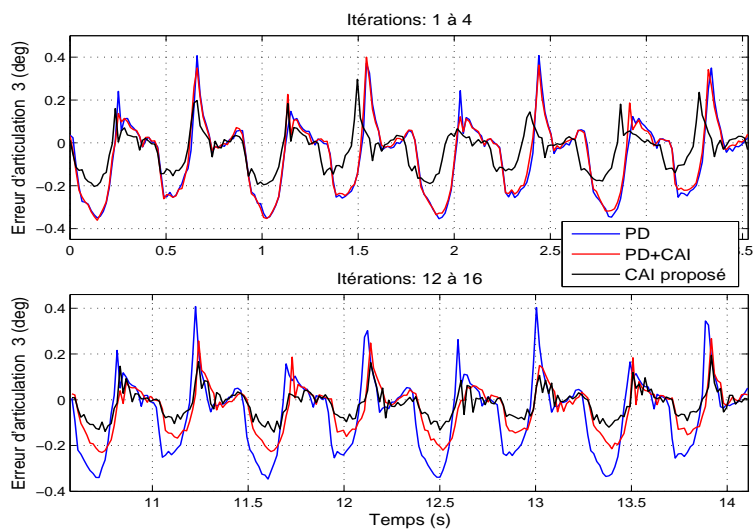
Les résultats expérimentaux sont présentés de la figure 3.32 jusqu'à la figure 3.34. On constate que la CAI proposée reste celle qui donne les meilleurs résultats, où l'erreur de poursuite démarre avec une petite valeur par rapport à la commande PD plus CAI type PD puis elle diminue à travers les itérations avec un couple acceptable.



(a)



(b)



(c)

Figure 3.30 – Évolution de l’erreur de poursuite en fonction des itérations. (a) articulation 1, (b) articulation 2, (c) articulation 3

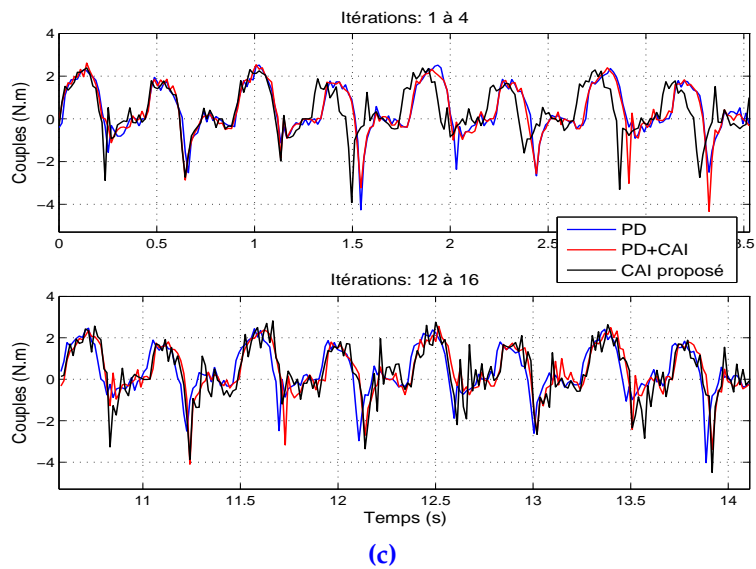
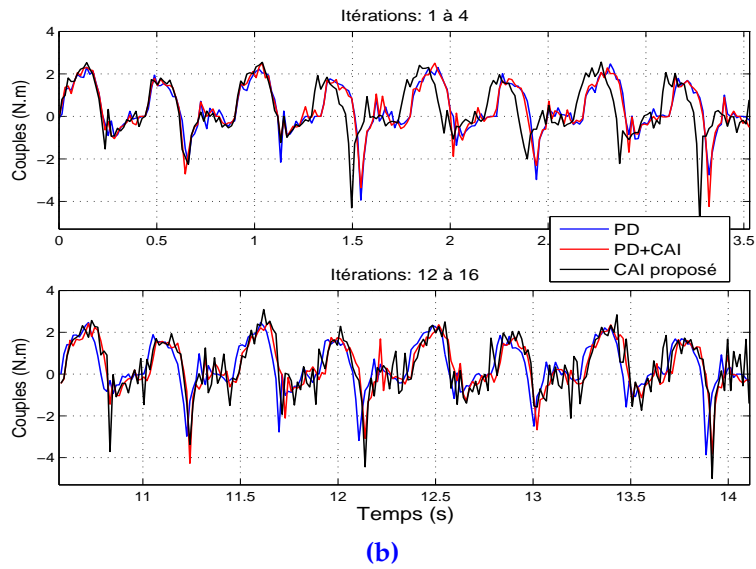
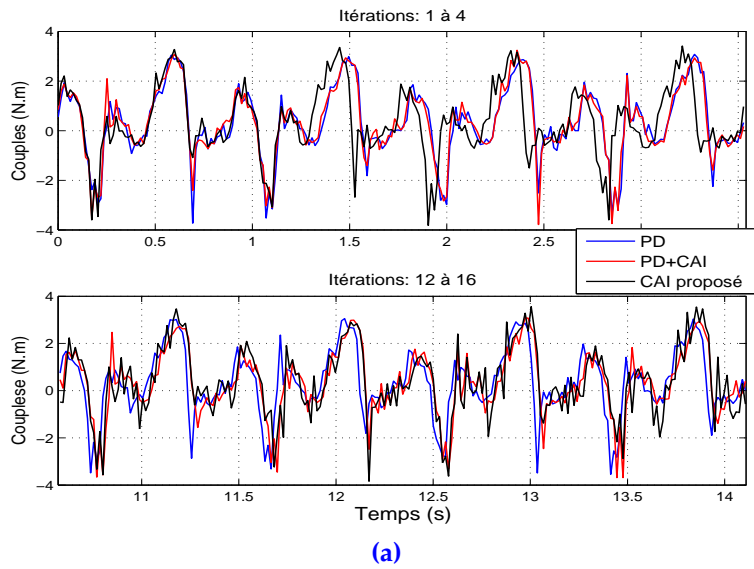


Figure 3.31 – Couples de commandes expérimentaux. (a) articulation 1, (b) articulation 2, (c) articulation 3

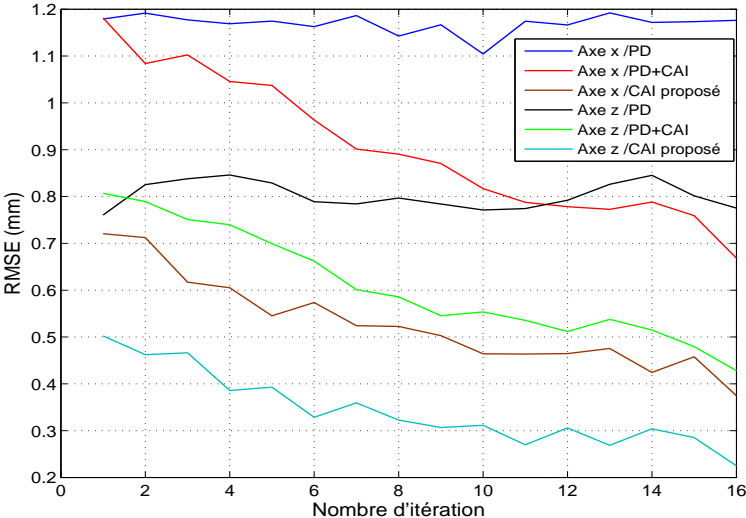
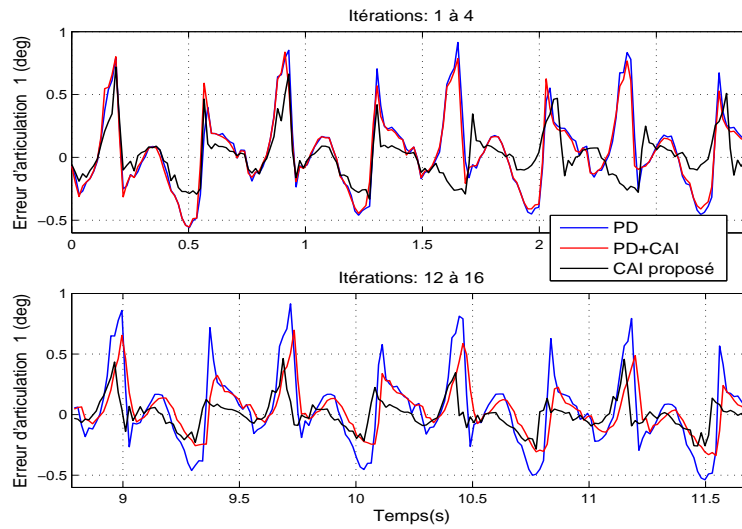
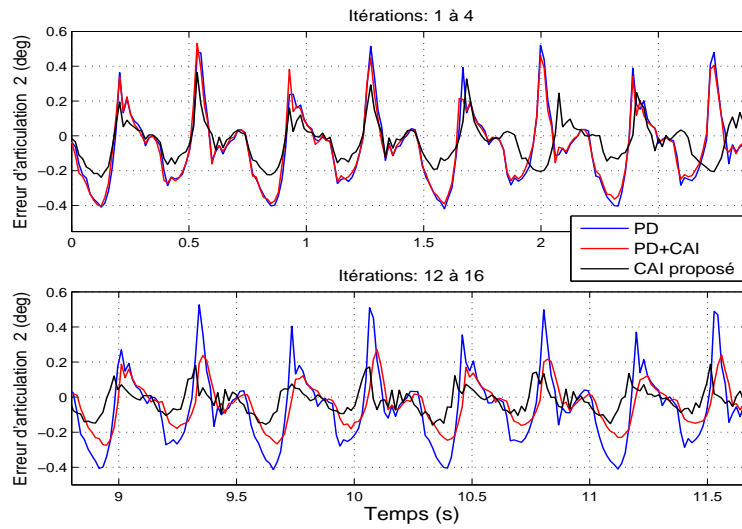


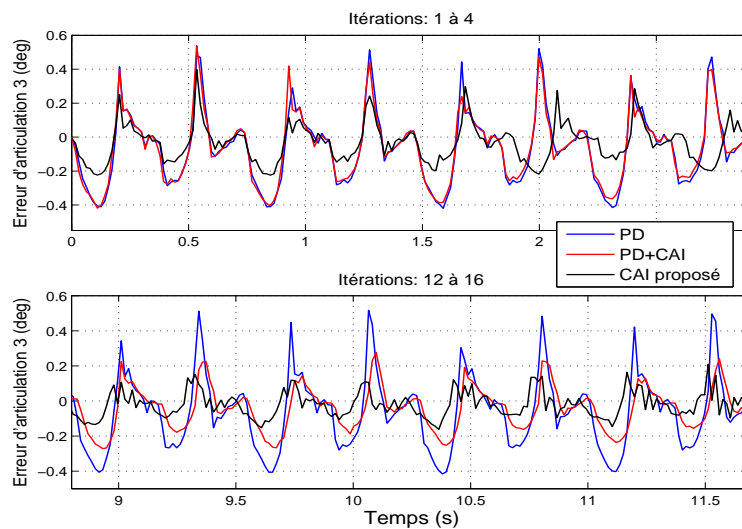
Figure 3.32 – Critère RMSE suivant les itérations sous une accélération de 1.5 G



(a)

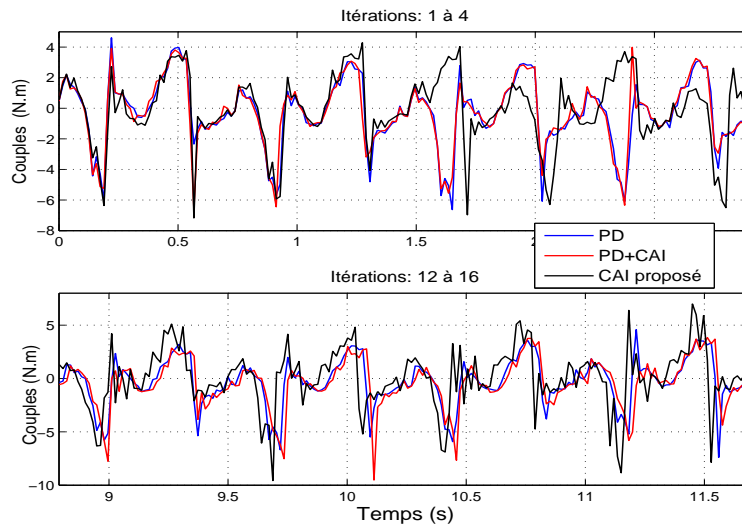


(b)

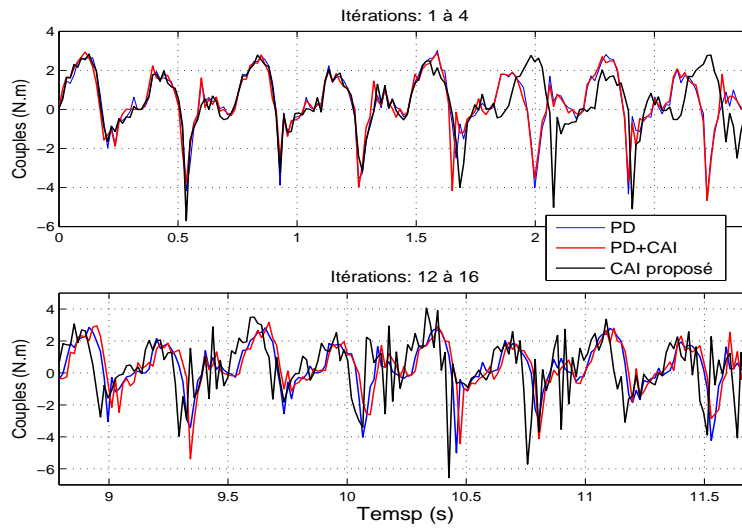


(c)

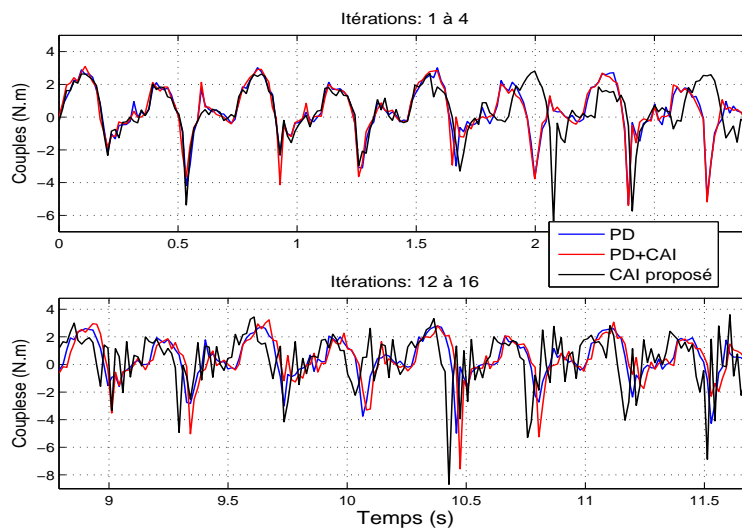
Figure 3.33 – Évolution de l'erreur de poursuite en fonction des itérations sous une accélération de 2G. (a) articulation 1, (b) articulation 2, (c) articulation 3



(a)



(b)



(c)

Figure 3.34 – Couples de commandes expérimentaux sous une accélération de 2G. (a) articulation 1, (b) articulation 2, (c) articulation 3

3.6 Conclusion

Dans ce chapitre, nous avons traité le problème de poursuite de trajectoire des robots manipulateurs exécutant des tâches répétitives par la proposition de trois schémas de commande par apprentissage itératif. Le premier schéma est synthétisé sous la condition de réinitialisation de l'erreur. Cette loi de commande utilise le modèle dynamique pour minimiser l'effet du couplage entre les articulations, la CAI de type PD pour améliorer les performances avec des trajectoires répétitives et un terme de robustification pour compenser les perturbations extérieures. Le deuxième schéma est constitué de la commande classique PD plus CAI type PD. Ce schéma peut fonctionner sous la condition d'alignement qu'est plus pratique par rapport à la condition de réinitialisation de l'erreur. Pour le troisième schéma, nous avons repris le deuxième schéma avec des gains adaptatifs afin d'accélérer la convergence et un terme de robustification afin de rejeter les perturbations extérieures et augmenter la robustesse. L'analyse de la convergence a été effectuée pour les trois schémas proposés en se basant sur la méthode de Lyapunov, où il a été démontré, pour les trois commandes, que l'erreur de poursuite converge vers zéro quand le nombre d'itérations tend vers l'infini.

Les résultats de simulation ont été obtenus pour le premier schéma ainsi que, des résultats expérimentaux pour le deuxième et troisième schémas proposés, où une étude comparative a été présentée par rapport au régulateur PD. Les résultats expérimentaux et les simulations ont validé l'analyse de la convergence où l'erreur approche zéro avec l'augmentation du nombre d'itérations. De plus, nous avons démontré à travers les expérimentations que, dans un environnement répétitif, la commande partiellement indépendante de modèle PD plus CAI type PD est plus performante que la commande par couple a priori qui est basée sur le modèle dynamique.

Chapitre 4

Commande par Apprentissage Itératif à Trajectoires Non Répétitives

4.1 Introduction

Lors du développement théorique de la CAI, deux hypothèses fondamentales sont imposées, à savoir la réinitialisation de l'erreur et la répétition de la trajectoire désirée. La réinitialisation de l'erreur suppose qu'à chaque itération la trajectoire désirée et la trajectoire réelle démarrent de la même position. En réalité cette hypothèse n'est pas vérifiée en pratique. De nombreux travaux ont été proposés pour remplacer cette condition (voir introduction générale). La condition d'alignement semble être la plus raisonnable et la plus pratique pour les mouvements continus. Par conséquent, elle a été introduite dans de nombreuses stratégies de la CAI [30][64].

L'hypothèse de la trajectoire répétitive considère que le système effectue toujours la même trajectoire sur un intervalle de temps fixe. Cette hypothèse est difficile à vérifier en pratique, puisqu'en général, les systèmes poursuivent des trajectoires composées de sous trajectoires successives de profil similaires ou non et de caractéristiques temporelles et d'amplitude différentes [65]. Peu de travaux sur la CAI ont été proposés pour éliminer l'hypothèse de la trajectoire répétitive. Dans [21], des commandes CAI type D, type PD et type PID ont été présentées pour commander un système MIMO, ainsi qu'un autre schéma pour le système MISO (multi-input-single-output) a été développé dans [22]. Dans les deux travaux, la trajectoire de référence est supposée variable en amplitude d'une itération à l'autre. Cependant, la trajectoire a un intervalle de temps fixe avec la condition de la réinitialisation de l'erreur est considérée dans le travail [22]. Dans [23], les auteurs proposent un schéma de la CAI pour un système SISO sans prendre en compte les incertitudes et les perturbations externes. De plus, la loi de commande est en boucle ouverte, ce qui va restreindre le domaine d'application de cette approche dans le milieu industriel. Dans [6] et [24], les auteurs présentent une commande avec trajectoire désirée à intervalle de temps fixe, amplitude variable et avec position initiale coïncidant avec celle de la trajectoire réelle pour chaque itération.

Afin de surmonter ces contraintes, nous avons développé dans ce chapitre deux commandes CAI pour les systèmes (MIMO) non linéaires incertains soumis à des perturbations externes et ayant des trajectoires non répétitives. La commande proposée fonctionne sous la condition d'alignement et avec trajectoire non répétitive en intervalle de temps, en amplitude et en position initiale et finale. Une transformation en temps a été introduite et combinée avec la méthode de Lyapunov pour synthétiser une loi de commande garantissant la convergence asymptotique de l'erreur de poursuite sur l'axe des itérations.

4.2 Formulation du problème

Considérons un système non linéaire MIMO avec trajectoires non répétitives, décrit par l'équation d'état suivante :

$$\begin{cases} \dot{x}_{1,k} = x_{2,k} \\ \dot{x}_{2,k} = f_k(x_k) + B_k(x_k)u_k(t_k) + \eta_k(t_k) \\ y_k = x_{1,k}, \quad t_k \in [0, T.T_k] \end{cases} \quad (4.1)$$

où, $x_k = [x_{1,k}, x_{2,k}]^T \in \mathbb{R}^n$ est le vecteur d'état, $u \in \mathbb{R}^{\frac{n}{2}}$, vecteur des entrées de commande et $\eta_k \in \mathbb{R}^{\frac{n}{2}}$ est le vecteur des perturbations externes. $f(x_k) \in \mathbb{R}^{\frac{n}{2}}$ est une fonction non linéaire, $B(x_k) \in \mathbb{R}^{\frac{n}{2} \times \frac{n}{2}}$ est la matrice de gain d'entrée du système, $y \in \mathbb{R}^{\frac{n}{2}}$ est le vecteur des sorties mesurées, et $T.T_k$ est l'intervalle de temps de la k-ème itération, où T_k peut être différente d'une itération à l'autre.

Le système (4.1) peut être récrit en utilisant le modèle nominal suivant :

$$\begin{cases} \dot{x}_{1,k} = x_{2,k} \\ \dot{x}_{2,k} = f_n(x_k) + B_n(x_k)u_k(t_k) + d_k(x_{1,k}, x_{2,k}, \dot{x}_{2,k}, t_k) \\ y = x_{1,k}, \quad t_k \in [0, T.T_k] \end{cases} \quad (4.2)$$

Où, $d_k(x_{1,k}, x_{2,k}, \dot{x}_{2,k}, t_k)$ est le vecteur des perturbations augmentées, qui peut être une combinaison des différentes erreurs de modélisation et perturbations externes.

Dans notre travail, nous avons pris le modèle dynamique du robot Delta. Ainsi, les perturbations augmentées peuvent être exprimées comme suit :

$$d_k(x_{1,k}, x_{2,k}, \dot{x}_{2,k}, t_k) = -M_n(x_k)^{-1}(\Delta M \dot{x}_{2,k} + \Delta C x_{2,k} + \Delta G - W_k(t))$$

où Δ indique l'erreur de modélisation et $W_k(t_k)$ est la perturbation externe agissant sur le robot manipulateur, alors que, $f_n(x) = -M_n^{-1}(C_n x_2 + G_n)$, et $B_n = M_n^{-1}$.

La trajectoire désirée dans le k-ème itération est exprimée comme $x_{r,k}(t_k) \in \mathbb{R}^{\frac{n}{2}}$, $t_k \in [0, T.T_k]$.

Définissons l'erreur de poursuite en position et en vitesse comme :

$$e_{1,k}(t_k) = x_{r,k}(t_k) - x_{1,k}(t_k) \quad (4.3)$$

$$e_{2,k}(t_k) = \dot{x}_{r,k}(t_k) - \dot{x}_{2,k}(t_k), \quad t_k \in [0, T.T_k] \quad (4.4)$$

Notre objectif est de concevoir une commande CAI $u_k(t_k)$ garantissant la convergence asymptotique, sous une trajectoire non répétitive, de l'erreur de poursuite en position et en vitesse quand le nombre d'itérations k augmente infiniment c-à-d : $e_{1,k}(t_k) = \lim_{k \rightarrow \infty} e_{2,k}(t_k) = 0, \forall t_k \in [0, T.T_k]$.

Définition : Dans ce travail, la trajectoire désirée est supposée différente d'une itération à l'autre avec les caractéristiques suivantes :

a) La durée d'une itération, $T.T_k$ peut être différente dans chaque itération comme, $t_{\text{nouveau}} \in [0, T.T_{\text{nouveau}}]$. Contrairement aux travaux présentés dans [21, 22, 24] et [66], où à travers les itérations la trajectoire désirée a toujours un intervalle de temps fixe.

b) L'amplitude de la trajectoire désirée peut être différente d'une itération à l'autre comme, $x_{r,\text{nouvelle}}(t_{\text{nouvelle}}) = \alpha x_{r,\text{ancien}}(t_{\text{nouvelle}}) + \beta$. Ce qui signifie qu'après avoir changer l'échelle de temps de la référence précédente, nous pouvons ajuster son amplitude avec différentes échelles.

c) La position initiale et finale de la trajectoire désirée peut être différente d'une itération à l'autre. Ce qui signifie que α et β non seulement peuvent être constantes mais peuvent aussi varier. Contrairement aux travaux présentés dans [20, 23] où la trajectoire désirée a toujours la même position initiale et finale.

Pour les systèmes MIMO, la trajectoire désirée $x_{r,k}(t_k)$ peut s'exprimer comme :

$$x_{r,k}(t_k) = [x_{r,k,1}(t_{k,1}), x_{r,k,2}(t_{k,2}), \dots, x_{r,k,\frac{n}{2}}(t_{k,\frac{n}{2}})]^T \quad t_{k,i} \in [0, TT_{k,i}], i = 1 : \frac{n}{2} \quad (4.5)$$

L'équation (4.5) signifie que chaque composante de la trajectoire désirée a sa durée de fonctionnement indépendante des autres composantes.

Nous définissons la transformation de temps t comme suit : $t = \frac{t_{k,i}}{T_{k,i}}$, où $t \in [0, T]$ et $i = 1 : \frac{n}{2}$.

L'utilisation d'une transformation de temps est commune dans le domaine de la robotique et de la chimie en raison de ses avantages, tels que la linéarisation d'un système non linéaire qui ne peut pas être linéarisée en échèle de temps ordinaire [67]. Plus importante, cette transformation ne modifie pas la stabilité du système comme il est démontrée dans [68]. Elle est donc utile, dans le sens de la facilitation de l'analyse de la convergence, où elle nous aide à synthétiser une loi de commande pour les systèmes MIMO exécutant des trajectoires non répétitives.

On définit la variable $\rho_{k,i}(t) = t_{k,i} = T_{k,i}t$, pour $i = 1 : \frac{n}{2}$. Aussi, nous définissons la matrice $\rho_k(t)$ comme suit :

$$\rho_k(t) = \text{diag}\{\rho_{k,1}(t), \dots, \rho_{k,\frac{n}{2}}(t)\}.$$

Le modèle dynamique (4.2) peut se réécrire avec en considérant la transformation de temps t , comme suit :

$$\begin{cases} \dot{x}_{1,k}(t) = \dot{\rho}_k(t)x_{2,k}(\rho_k(t)) \\ \dot{x}_{2,k}(t) = \dot{\rho}_k(t)(f_n(x_k) + B_n(x_k)u_k(\rho_k(t)) + d_k(\rho_k(t))) \\ y = x_{1,k}, \quad t \in [0, T] \end{cases} \quad (4.6)$$

où le vecteur $x_{2,k}(\rho_k(t))$ est utilisé pour simplifier l'expression suivante :

$$x_{2,k}(\rho_k(t)) = [x_{2,k,1}(\rho_{k,1}(t)), \dots, x_{2,k,\frac{n}{2}}(\rho_{k,\frac{n}{2}}(t))]^T$$

De même pour les vecteurs $u(\rho_k(t))$, $d_k(\rho_k(t))$, $e_{1,k}(\rho_k(t))$ et $e_{2,k}(\rho_k(t))$.

On définit les vecteurs suivants comme :

$$\tilde{q}_{1,k}(t) \triangleq e_{1,k}(\rho_k(t)), \tilde{q}_{2,k}(t) \triangleq e_{2,k}(\rho_k(t)),$$

$$\theta_k(t) \triangleq d_k(\rho_k(t)), X_{r,k}(t) \triangleq x_{r,k}(\rho_k(t)), v_k(t) \triangleq u_k(\rho_k(t)).$$

Ainsi, (4.6) peut se réécrire comme :

$$\begin{cases} \dot{x}_{1,k}(t) = \dot{\rho}_k(t)x_{2,k}(\rho_k(t)) \\ \dot{x}_{2,k}(t) = \dot{\rho}_k(t)(f_n(x) + B_n(x)v_k(t) + \theta_k(t)) \\ y = x_{1,k}, \quad t \in [0, T] \end{cases} \quad (4.7)$$

Tandis que, les erreurs dynamiques (4.3) et (4.4) sont données par :

$$\dot{\tilde{q}}_{1,k}(t) = \dot{\rho}_k(t)\dot{e}_{1,k}(\rho_k(t)) = \dot{\rho}_k(t)\tilde{q}_{2,k}(t), \quad t \in [0, T] \quad (4.8)$$

$$\dot{\hat{q}}_{2,k}(t) = \dot{\rho}_k(t)\dot{e}_{2,k}(\rho_k(t)) = \dot{\rho}_k(t)^{-1}\ddot{X}_{r,k}(t) - \dot{\rho}_k(t)\dot{x}_{2,k}(\rho_k(t)) \quad (4.9)$$

Nous donnons les hypothèses suivantes :

Hypothèse 1 : La condition d'alignement est satisfaite :

$$x_{1,k-1}(TT_{k-1}) = x_{1,k}(0), \quad x_{2,k-1}(TT_{k-1}) = x_{2,k}(0) \quad \forall k \in \mathbb{Z}_+$$

Hypothèse 2 : La matrice $B_n(x_k)$ est inversible, $\forall k \in \mathbb{Z}_+$

Hypothèse 3 : Il existe une fonction positive lisse inconnue $\theta(t) \in \mathbb{R}$ qui satisfait :

$$\|\theta_k(t)\|_1 \leq \theta(t) \quad (4.10)$$

Remarque 1 : Comme le mouvement est continu, la trajectoire désirée travaille aussi sous l'hypothèse d'alignement, i.e.,

$$x_{r,k-1}(TT_{k-1}) = x_{r,k}(0), \quad \dot{x}_{r,k-1}(TT_{k-1}) = \dot{x}_{r,k}(0) \quad \forall k \in \mathbb{Z}_+$$

Remarque 2 : L'hypothèse 3, indique que les perturbations augmentées dépendant des itérations, peuvent être bornées par une fonction positive inconnue et indépendante des itérations. Cette hypothèse est beaucoup plus réalisable par rapport à la bornitude des perturbations par une fonction connue ou une constante comme dans [69].

4.3 Commande CAI basée modèle pour les systèmes MIMO à trajectoires non répétitives

4.3.1 Conception de la commande

La loi de commande proposée pour la k-ème itération est exprimée par :

$$u(t_k) = B_n^{-1}(x_k)(-f_n(x_k) + \ddot{x}_{r,k}(t_k) + \lambda e_{2,k}(t_k) + \Phi_k \hat{d}_k(t_k) + K(e_{2,k}(t_k) + \lambda e_{1,k}(t_k))) \quad (4.11)$$

$$\hat{d}_k(t_k) = \hat{d}_{k-1}(t_k) + \Gamma \Phi_k \dot{\rho}_k(e_{2,k}(t_k) + \lambda e_{1,k}(t_k)), t_k \in [0, T.T_k] \quad (4.12)$$

où $\Phi_k \in \mathbb{R}^{\frac{n}{2} \times \frac{n}{2}}$.

L'expression de Φ_k est donné par :

$$\Phi_k = \text{diag}\{\text{sgn}(e_{2,k,1}(t_{k,1}) + \lambda e_{1,k,1}(t_{k,1})), \dots, \text{sgn}(e_{2,k,\frac{n}{2}}(t_{k,\frac{n}{2}}) + \lambda e_{1,k,\frac{n}{2}}(t_{k,\frac{n}{2}}))\} \quad (4.13)$$

La loi de commande donnée par (4.11) et (4.12) peut se réécrire en respectant la transformation de temps t comme suit :

$$v_k(t) = B_n^{-1}(x_k)(-f_n(x_k) + \dot{\rho}_k^{-2}\ddot{X}_{r,k}(t) + \Phi_k \hat{\theta}_k(t) + \lambda \tilde{q}_2(t) + K(\tilde{q}_2(t) + \lambda \tilde{q}_1(t))) \quad (4.14)$$

$$\hat{\theta}_k(t) = \hat{\theta}_{k-1}(t) + \Gamma \Phi_k \dot{\rho}_k(\tilde{q}_2(t) + \lambda \tilde{q}_1(t)), t \in [0, T] \quad (4.15)$$

$\hat{d}_k(t_k)$, $\hat{\theta}_k(t)$ désignent respectivement la CAI à t_k , et à t . Où, la matrice de retour d'état K est diagonale définie positive, et les gains d'apprentissage Γ et λ sont des constantes positives.

4.3.2 Influence du vecteur $\hat{d}_{k-1}(t_k)$

À partir de l'équation (4.12) nous avons le vecteur $\hat{d}_{k-1}(t_k)$, où $t_k \in [0, T.T_k]$. D'autre part, seule les données de $\hat{d}_{k-1}(t_{k-1})$ où $t_{k-1} \in [0, T.T_{k-1}]$ sont disponibles. Pour faire face à ce problème, nous avons trois scénarios :

a) $T_{k-1} = T_k$: Dans ce cas, le problème ne se pose pas car $\hat{d}_{k-1}(t_k)$ va utiliser toutes les données de la (k-1)-ème itération incluses dans $[0, T.T_{k-1}]$ pour fournir les données requises de la k-ème itération incluses dans $[0, T.T_k]$.

b) $T_{k-1} > T_k$: Dans ce cas, les données disponibles dans la (k-1)-ème itération sont plus importante que nécessaire, et $\hat{d}_{k-1}(t_k)$ va les utiliser jusqu'à $t_k = T.T_k$.

c) $T_{k-1} < T_k$: Dans ce cas, $\hat{d}_{k-1}(t_k)$ va utiliser toutes les données de la (k-1)-ème itération incluses dans $[0, T.T_{k-1}]$ puis nous pouvons compléter les données manquantes jusqu'à $T.T_k$ par trois choix :

$$\text{c.1)} \hat{d}_k(t_k) = \hat{d}_{k-1}(T.T_{k-1}) + \Gamma \Phi_k \dot{\rho}_k(e_{2,k}(t_k) + \lambda e_{1,k}(t_k)), \forall t_k \in [T.T_{k-1}, T.T_k]. \quad (4.16)$$

$$\text{c.2)} \hat{d}_k(t_k) = \Gamma \Phi_k \dot{\rho}_k(e_{2,k}(t_k) + \lambda e_{1,k}(t_k)), \forall t_k \in [T.T_{k-1}, T.T_k]. \quad (4.17)$$

$$\text{c.3)} \hat{d}_k(t_k) = AVR(\hat{d}_{k-1}(t_k \in [0, T.T_{k-1}])) + \Gamma \Phi_k \dot{\rho}_k(e_{2,k}(t_k) + \lambda e_{1,k}(t_k)), \forall t_k \in [T.T_{k-1}, T.T_k]. \quad (4.18)$$

où AVR représente la fonction "Average" (moyenne).

Remarque 3 : Les équations (4.16), (4.17) et (4.18) indiquent que, $\forall t_k \in [T.T_{k-1}, T.T_k]$, nous pouvons choisir arbitrairement la valeur de $\hat{d}_{k-1}(t_k)$ sans effet sur la stabilité du système, où l'erreur de poursuite va tendre vers zéro lorsque le nombre d'itérations k approche l'infini. Toutefois, ce choix peut avoir une influence sur les performances à travers les itérations, comme la vitesse de convergence et la douceur de la commande.

4.3.3 Analyse de la convergence

Théorème 4 [25] : Considérons le système MIMO non linéaire incertain (4.2), sous la commande (4.11) et (4.12). Si les hypothèse 1-3 sont satisfaites, alors l'erreur de poursuite de la trajectoire non répétitive converge vers zéro quand le nombre d'itérations k augmente à l'infini sur un intervalle de temps fini $[0, T.T_k]$, i.e., $\lim_{k \rightarrow \infty} e_{1,k}(t_k) = \lim_{k \rightarrow \infty} e_{2,k}(t_k) = 0$, $\forall t_k \in [0, T.T_k]$.

Preuve

Soit la fonction de Lyapunov suivante

$$w_k(t) = \frac{1}{2} \Gamma (\tilde{q}_{2,k}(t) + \lambda \tilde{q}_{1,k}(t))^T (\tilde{q}_{2,k}(t) + \lambda \tilde{q}_{1,k}(t)) + \frac{1}{2} \int_0^t \varphi_k^T \varphi_k d\tau \quad (4.19)$$

où $\varphi_k(t) = \vartheta(t) - \hat{\theta}_k(t)$, et le vecteur $\vartheta(t) \in \mathbb{R}^n$, s'exprime par $\vartheta(t) = [\theta(t), \theta(t), \dots, \theta(t)]^T$.

De plus, on a

$$\Delta w_k(t) = w_k(t) - w_{k-1}(t) \quad (4.20)$$

par conséquent

$$\begin{aligned} \Delta w_k(t) = & \frac{1}{2} \Gamma (\tilde{q}_{2,k}(t) + \lambda \tilde{q}_{1,k}(t))^T (\tilde{q}_{2,k}(t) + \lambda \tilde{q}_{1,k}(t)) - \\ & \frac{1}{2} \Gamma (\tilde{q}_{2,k-1}(t) + \lambda \tilde{q}_{1,k-1}(t))^T (\tilde{q}_{2,k-1}(t) + \lambda \tilde{q}_{1,k-1}(t)) + \\ & \frac{1}{2} \int_0^t \varphi_k^T \varphi_k - \varphi_{k-1}^T \varphi_{k-1} d\tau \end{aligned} \quad (4.21)$$

Nous avons

$$\begin{aligned} \frac{1}{2}\Gamma(\tilde{q}_{2,k}(t) + \lambda\tilde{q}_{1,k}(t))^T(\tilde{q}_{2,k}(t) + \lambda\tilde{q}_{1,k}(t)) &= \frac{1}{2}\Gamma(\tilde{q}_{2,k}(0) + \lambda\tilde{q}_{1,k}(0))^T(\tilde{q}_{2,k}(0) + \lambda\tilde{q}_{1,k}(0)) \\ &+ \Gamma \int_0^t (\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T (\dot{\tilde{q}}_{2,k} + \lambda\dot{\tilde{q}}_{1,k}) d\tau \end{aligned} \quad (4.22)$$

À partir de (4.7), (4.8), (4.9) et (4.22) on obtient

$$\begin{aligned} \frac{1}{2}\Gamma(\tilde{q}_{2,k}(t) + \lambda\tilde{q}_{1,k}(t))^T(\tilde{q}_{2,k}(t) + \lambda\tilde{q}_{1,k}(t)) &= \\ \frac{1}{2}\Gamma(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})(0)^T(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})(0) &+ \\ \Gamma \int_0^t (\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T (\dot{\rho}_k^{-1} \ddot{X}_{r,k} - \dot{\rho}_k(f_n(x_k) + B_n(x_k)v_k + \theta_k) + \lambda\dot{\rho}_k \tilde{q}_{2,k}) d\tau \end{aligned} \quad (4.23)$$

Nous avons aussi

$$\frac{1}{2} \int_0^t \varphi_k^T \varphi_k - \varphi_{k-1}^T \varphi_{k-1} d\tau = \int_0^t (\hat{\theta}_k - \vartheta)^T (\hat{\theta}_k - \hat{\theta}_{k-1}) - \frac{1}{2} (\hat{\theta}_k - \hat{\theta}_{k-1})^T (\hat{\theta}_k - \hat{\theta}_{k-1}) d\tau \quad (4.24)$$

Substituant (4.23) et (4.24) dans (4.21) on trouve

$$\begin{aligned} w_k(t) - w_{k-1}(t) &= \frac{1}{2}\Gamma(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T(0)(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})(0) - \\ \frac{1}{2}\Gamma(\tilde{q}_{2,k-1} + \lambda\tilde{q}_{1,k-1})^T(t)(\tilde{q}_{2,k-1} + \lambda\tilde{q}_{1,k-1})(t) &+ \\ \Gamma \int_0^t (\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T (\dot{\rho}_k^{-1} \ddot{X}_{r,k} - \dot{\rho}_k(f_n(x_k) + B_n(x_k)v_k + \theta_k - \lambda\tilde{q}_{2,k})) d\tau & \\ + \int_0^t (\hat{\theta}_k - \vartheta)^T (\hat{\theta}_k - \hat{\theta}_{k-1}) - \frac{1}{2} (\hat{\theta}_k - \hat{\theta}_{k-1})^T (\hat{\theta}_k - \hat{\theta}_{k-1}) d\tau \end{aligned} \quad (4.25)$$

L'équation (4.10) de l'hypothèse 3 et (4.13) garantissent

$$\begin{aligned} (\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T(-\theta_k) &\leq (\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T \text{sign}(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})\theta \\ &= (\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T (\Phi_k \vartheta) \end{aligned} \quad (4.26)$$

Remplaçant (4.26) dans (4.25) conduit à

$$\begin{aligned} w_k(t) - w_{k-1}(t) &\leq \frac{1}{2}\Gamma(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T(0)(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})(0) - \\ \frac{1}{2}\Gamma(\tilde{q}_{2,k-1} + \lambda\tilde{q}_{1,k-1})^T(t)(\tilde{q}_{2,k-1} + \lambda\tilde{q}_{1,k-1})(t) &+ \\ \Gamma \int_0^t (\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T (\dot{\rho}_k^{-1} \ddot{X}_{r,k} - \dot{\rho}_k(f_n(x_k) + B_n(x_k)v_k - \lambda\tilde{q}_{2,k}) + \Phi_k \rho_k \vartheta) d\tau & \\ + \int_0^t (\hat{\theta}_k - \vartheta)^T (\hat{\theta}_k - \hat{\theta}_{k-1}) - \frac{1}{2} (\hat{\theta}_k - \hat{\theta}_{k-1})^T (\hat{\theta}_k - \hat{\theta}_{k-1}) d\tau \end{aligned} \quad (4.27)$$

À partir de la loi de commande donnée par (4.14) et (4.15), on obtient

$$\begin{aligned} w_k(t) - w_{k-1}(t) &\leq \frac{1}{2}\Gamma(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T(0)(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})(0) - \\ \frac{1}{2}\Gamma(\tilde{q}_{2,k-1} + \lambda\tilde{q}_{1,k-1})^T(t)(\tilde{q}_{2,k-1} + \lambda\tilde{q}_{1,k-1})(t) &- \\ \int_0^t \Gamma(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T \dot{\rho}_k K(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k}) d\tau - & \\ \frac{1}{2} \int_0^t \Gamma^2(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k})^T \dot{\rho}_k^2(\tilde{q}_{2,k} + \lambda\tilde{q}_{1,k}) d\tau \end{aligned} \quad (4.28)$$

Nous avons aussi

$$w_k(t) = w_1(t) + \sum_{i=2}^k \Delta w_i(t) \quad (4.29)$$

Ainsi, la substitution de (4.28) dans (4.29) donne

$$\begin{aligned}
w_k(t) \leq & w_1(t) + \sum_{i=2}^k \frac{1}{2} \Gamma(\tilde{q}_{2,i} + \lambda \tilde{q}_{1,i})^T(0)(\tilde{q}_{2,i} + \lambda \tilde{q}_{1,i})(0) - \\
& \frac{1}{2} \Gamma(\tilde{q}_{2,i-1} + \lambda \tilde{q}_{1,i-1})^T(t)(\tilde{q}_{2,i-1} + \lambda \tilde{q}_{1,i-1})(t) - \\
& \int_0^t \Gamma(\tilde{q}_{2,i} + \lambda \tilde{q}_{1,i})^T \dot{\rho}_i K(\tilde{q}_{2,i} + \lambda \tilde{q}_{1,i}) d\tau - \\
& \frac{1}{2} \int_0^t \Gamma^2(\tilde{q}_{2,i} + \lambda \tilde{q}_{1,i})^T \dot{\rho}_i^2(\tilde{q}_{2,i} + \lambda \tilde{q}_{1,i}) d\tau
\end{aligned} \tag{4.30}$$

À l'aide de l'hypothèse 1, on trouve

$$\begin{aligned}
& \frac{1}{2} \Gamma(\tilde{q}_{2,i} + \lambda \tilde{q}_{1,i})(0)^T(\tilde{q}_{2,i} + \lambda \tilde{q}_{1,i})(0) = \\
& \frac{1}{2} \Gamma(\tilde{q}_{2,i-1} + \lambda \tilde{q}_{1,i-1})(T)^T(\tilde{q}_{2,i-1} + \lambda \tilde{q}_{1,i-1})(T) \quad i = 2 : k
\end{aligned} \tag{4.31}$$

À partir de (4.30) et (4.31) on conclut

$$w_k(T) \leq w_1(T) - \Gamma \sum_{i=2}^k \dot{\rho}_{i_m} K_m \int_0^T \|\tilde{q}_{2,i} + \lambda \tilde{q}_{1,i}\|^2 d\tau \tag{4.32}$$

où K_m et $\dot{\rho}_{i_m}$ indiquent la valeur propre minimale des matrices K et $\dot{\rho}_i$ respectivement.

Nous allons vérifier la bornitude de $w_1(T)$.

À partir de la définition de $w_k(t)$, nous avons

$$w_1(t) = \frac{1}{2} \Gamma(\tilde{q}_{2,1}(t) + \lambda \tilde{q}_{1,1}(t))^T(\tilde{q}_{2,1}(t) + \lambda \tilde{q}_{1,1}(t)) + \frac{1}{2} \int_0^t \varphi_1^T \varphi_1 d\tau \tag{4.33}$$

La dérivée par rapport au temps de $w_1(t)$ est donnée comme suit

$$\dot{w}_1(t) = \Gamma(\tilde{q}_{2,1} + \lambda \tilde{q}_{1,1})^T(\dot{\tilde{q}}_{2,1} + \lambda \dot{\tilde{q}}_{1,1}) + \frac{1}{2} \varphi_1^T \varphi_1 \tag{4.34}$$

En utilisant (4.15), (4.24) et sachant que $\hat{\theta}_0 = 0$, on peut avoir

$$\begin{aligned}
\frac{1}{2} \varphi_1^T \varphi_1 &= \frac{1}{2} (\varphi_1^T \varphi_1 - \varphi_0^T \varphi_0) + \frac{1}{2} \varphi_0^T \varphi_0 \\
&= (\hat{\theta}_1 - \vartheta) (\Gamma \Phi_1 \dot{\rho}_1 (\tilde{q}_{2,1} + \lambda \tilde{q}_{1,1})) \\
&\quad - \frac{1}{2} \Gamma^2 (\tilde{q}_{2,1} + \lambda \tilde{q}_{1,1})^T \dot{\rho}_1^2 (\tilde{q}_{2,1} + \lambda \tilde{q}_{1,1}) + \frac{1}{2} \vartheta^T \vartheta
\end{aligned} \tag{4.35}$$

par conséquent

$$\begin{aligned}
\dot{w}_1(t) &= -(\tilde{q}_{2,1} + \lambda \tilde{q}_{1,1})^T (\Gamma \dot{\rho}_1 K + \frac{1}{2} \Gamma^2 \dot{\rho}_1^2) (\tilde{q}_{2,1} + \lambda \tilde{q}_{1,1}) \\
&\quad + \frac{1}{2} \vartheta^T \vartheta \\
&\leq \frac{1}{2} \vartheta^T \vartheta
\end{aligned} \tag{4.36}$$

Comme ϑ est positif continu sur l'intervalle de temps $[0, T]$, donc on peut avoir une constante telle que

$$L = \frac{1}{2} \int_0^T \vartheta^T \vartheta d\tau \leq \infty$$

D'où $w_1(t)$ peut être borné comme suit

$$0 \leq w_1(t) = w_1(0) + \int_0^t \dot{w}_1(\tau) d\tau \leq w_1(0) + \frac{1}{2} \int_0^t \vartheta^T \vartheta d\tau \leq w_1(0) + L \leq \infty \tag{4.37}$$

Comme, $w_1(0) = \frac{1}{2}\Gamma\|\lambda(x_{r,1}(0) - x_{1,1}(0))\|^2$ est borné, car $x_{r,1}(0)$ et $x_{1,1}(0)$ sont bornés (le système ne s'est pas encore déplacé). Ainsi $w_1(t)$ est borné sur $t \in [0, T]$, par conséquent, la bornitude de $w_1(T)$.

Comme le côté gauche de l'équation (4.32) est positif et $w_1(T)$ est borné, on conclu

$$\|\tilde{q}_{2,k}(t) + \lambda\tilde{q}_{1,k}(t)\|^2 \rightarrow 0 \quad \text{quand } k \rightarrow \infty, \forall t \in [0, T] \quad (4.38)$$

À partir de (4.38) on obtient

$$\tilde{q}_{2,k}(t) + \lambda\tilde{q}_{1,k}(t) \rightarrow 0 \quad \text{quand } k \rightarrow \infty, \forall t \in [0, T] \quad (4.39)$$

Comme nous avons $\tilde{q}_{2,k}(t) = \dot{\rho}_k^{-1}\dot{\tilde{q}}_{1,k}(t)$, alors on peut trouver

$$\dot{\tilde{q}}_{1,k} + \dot{\rho}_k\lambda\tilde{q}_{1,k} \rightarrow 0 \quad (4.40)$$

Finalement on obtient

$$\tilde{q}_{1,k}(t) \rightarrow 0 \text{ et } \tilde{q}_{2,k}(t) \rightarrow 0 \text{ pour } t \in [0, T] \text{ quand } k \rightarrow \infty \quad (4.41)$$

et

$$e_{1,k}(t_k) \rightarrow 0 \text{ et } e_{2,k}(t_k) \rightarrow 0 \text{ pour } t_k \in [0, T.T_k] \text{ quand } k \rightarrow \infty \quad (4.42)$$

Remarque 4 : Par rapport aux travaux existants relatif à l'application de la CAI sur les trajectoires non répétitive, cette loi de commande est plus générale et peut couvrir la plupart de ces travaux [6, 20–24, 66]. Elle peut fonctionner efficacement avec une trajectoire non répétitive avec temps d'exécution et amplitude variables d'un cycle à l'autre. En outre, l'approche proposée convient pour commander les systèmes MIMO non linéaires incertains soumis à des perturbations répétitives et non répétitives et/ou avec erreurs de modélisation.

4.3.4 Résultats de simulation

Considérons le modèle dynamique nominal du robot Delta décrit par l'équation suivante :

$$\ddot{q}_k = M_n(q_k)^{-1}(-C_n(q_k, \dot{q})\dot{q}_k - G_n(q_k) + \tau_k) + d_k(t_k) \quad (4.43)$$

où, le terme de perturbation augmenté avec les erreurs de modélisation $d_k(t_k)$ est donné par :

$$d_k(t_k) = -M_n(q_k)^{-1}(\Delta M(q_k)\ddot{q}_k + \Delta C(q_k, \dot{q})\dot{q}_k + \Delta G(q_k) - W_k(t_k)) \quad (4.44)$$

Les erreurs de modélisation sont choisies selon les expressions :

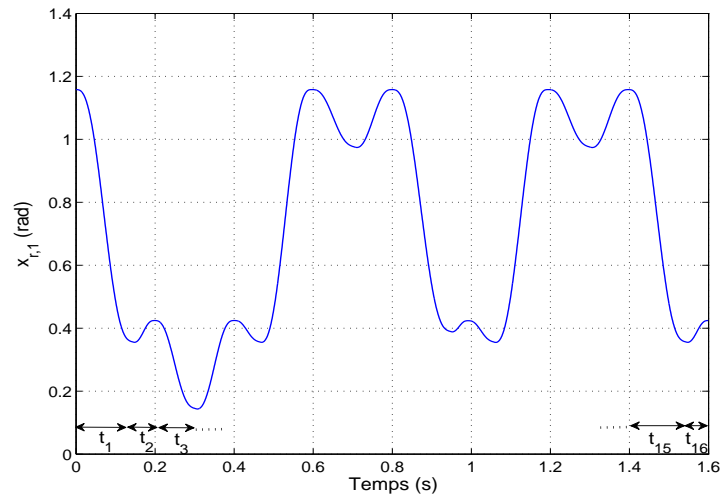
$$\Delta M(q_k) = 0.1 * M(q_k) \quad \Delta C(q_k, \dot{q}_k) = 0.1 * C(q_k, \dot{q}_k) \quad \Delta G(q_k) = 0.1 * G(q_k)$$

Le couple de perturbation externe $W_k(t_k)$ est défini par : $W_k(t_k) = rand(t_k)$.

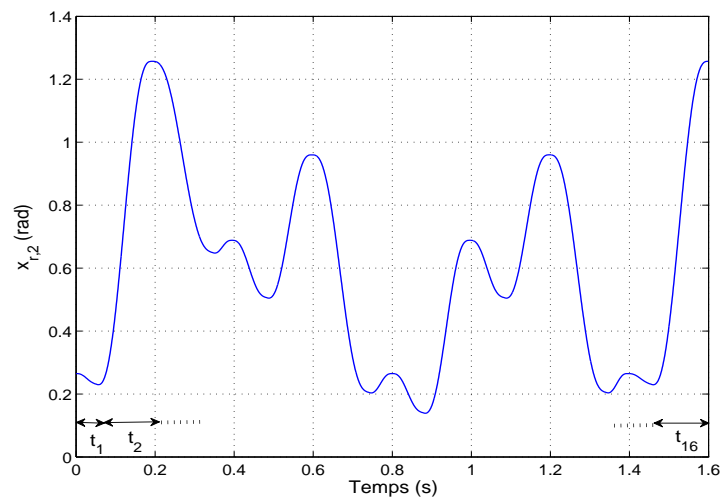
La trajectoire désirée utilisée pour les trois articulations est un polynôme de cinquième degré avec une vitesse et accélération initiale et finale égale à zéro à chaque itération. Son expression est donnée par :

$$x_{r,k}(t_k) = x_i + (x_f - x_i)\left(6\left(\frac{t_k}{T_k}\right)^5 - 15\left(\frac{t_k}{T_k}\right)^4 + 10\left(\frac{t_k}{T_k}\right)^3\right), \quad (4.45)$$

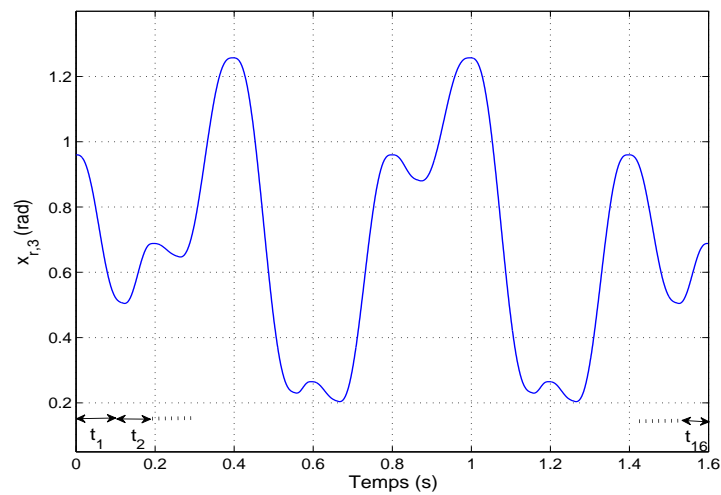
$$t_k \in [0, T.T_k].$$



(a)



(b)



(c)

Figure 4.1 – Trajectoire désirée. (a) articulation 1, (b) articulation 2, (c) articulation 3

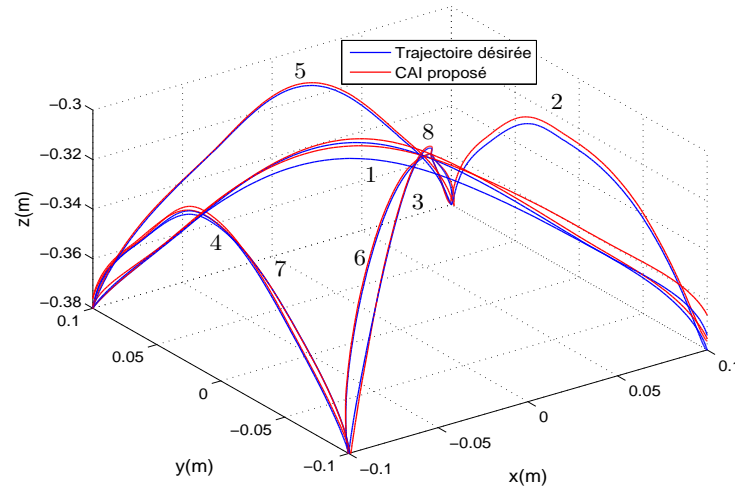


Figure 4.2 – Poursuite de trajectoire dans l'espace opérationnel

où, $T.T_k$ est la durée d'itération qui peut être différente d'une articulation à l'autre, et d'une itération à l'autre.

La figure 4.1 montre les trajectoires désirées des trois articulations. Il est tout à fait clair que la trajectoire désirée est différente dans les trois articulations et chacune possède 16 trajectoires partielles qui sont différentes. La figure 4.2 présente la trajectoire de prise et de pose dans l'espace opérationnel.

En se basant sur (4.11), (4.12) et (4.43), la loi de la commande appliquée au robot Delta est donnée par :

$$u_k(t_k) = M_n(q_k)(\ddot{x}_r(t_k) + \Phi_k \hat{d}_k(t_k) + \lambda e_{2,k}(t_k) + K(e_{2,k}(t_k) + \lambda e_{1,k}(t_k))) + C_n(q_k, \dot{q}_k) \dot{q}_k + G_n(q_k) \quad (4.46)$$

$$\hat{d}_k(t_k) = \hat{d}_{k-1}(t_k) + \Gamma \Phi_k \hat{\rho}_k(e_{2,k}(t_k) + \lambda e_{1,k}(t_k)), t_k \in [0, T.T_k] \quad (4.47)$$

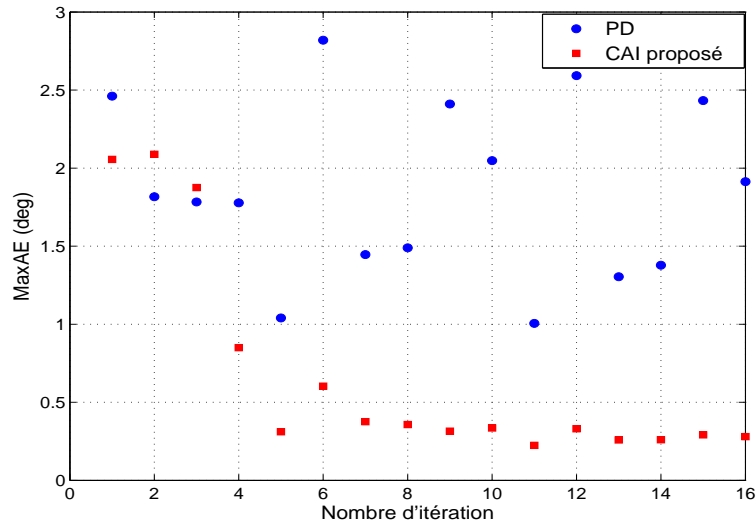
Les gains du régulateur sont sélectionnés comme $K = \text{diag}\{15, 15, 15\}$, $\Gamma = 30$, et $\lambda = 80$. Nous donnons la simulation dans deux cas :

Cas 1 : Les lois de commandes (4.11), (4.12) et (4.16) et la commande PD sont appliquées afin de démontrer l'efficacité de la commande proposée.

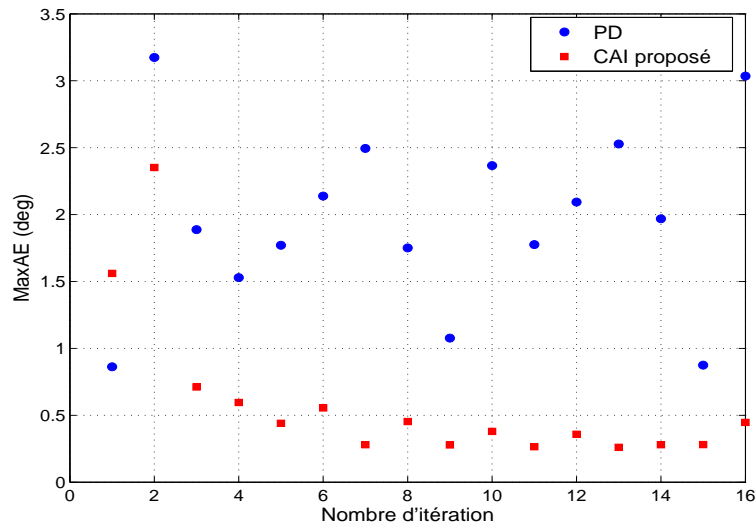
La figure 4.3 présente l'évolution du MaxAE à travers les itérations sous la commande proposée et le régulateur PD. Il est clair que, d'une part, la CAI proposée peut traiter efficacement la trajectoire non répétitive et les perturbations externes, où le MaxAE diminue pour les trois articulations. D'autre part, le régulateur PD donne les résultats attendus, où le MaxAE oscille pour les trois articulations à cause des perturbations externes et de la haute cadence du mouvement. La figure 4.4 indique les signaux de commande. On constate, que la CAI a des couples admissibles avec une faible amplitude par rapport au régulateur PD.

Cas 2 : Pour montrer l'influence du vecteur $d_k(t_k)$ sur les performances, une comparaison entre les trois lois de CAI (4.16), (4.17) et (4.18) est faite.

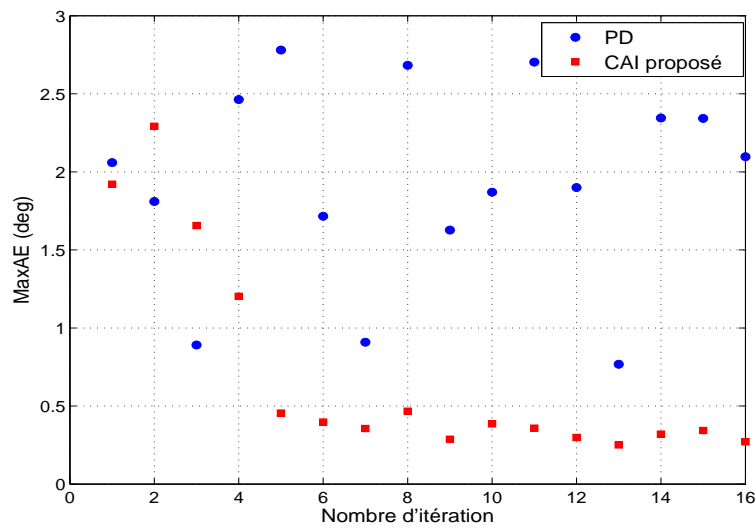
La figure 4.5 présente le MaxAE sous les trois algorithmes, et la figure 4.6 montre les signaux des commandes. On constate, que les régulateurs (4.16) et (4.18) donnent des performances et des couples similaires. D'autre part, on constate que les performances avec le régulateur (4.17) se sont dégradées, où la vitesse de convergence a diminué consi-



(a)

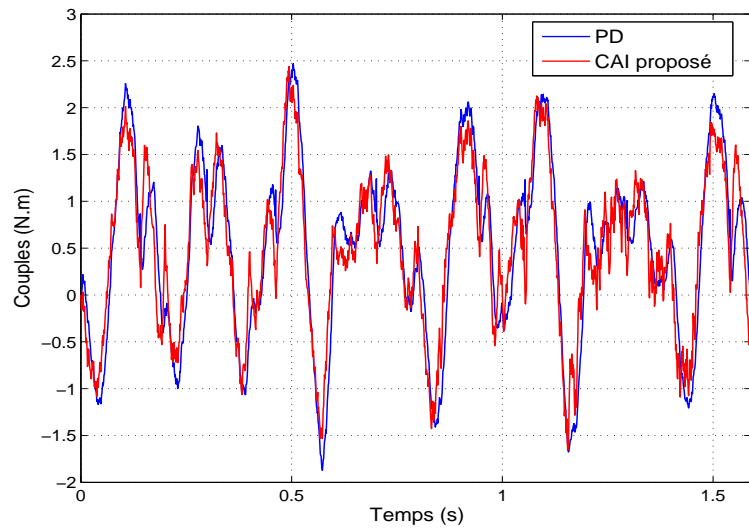


(b)

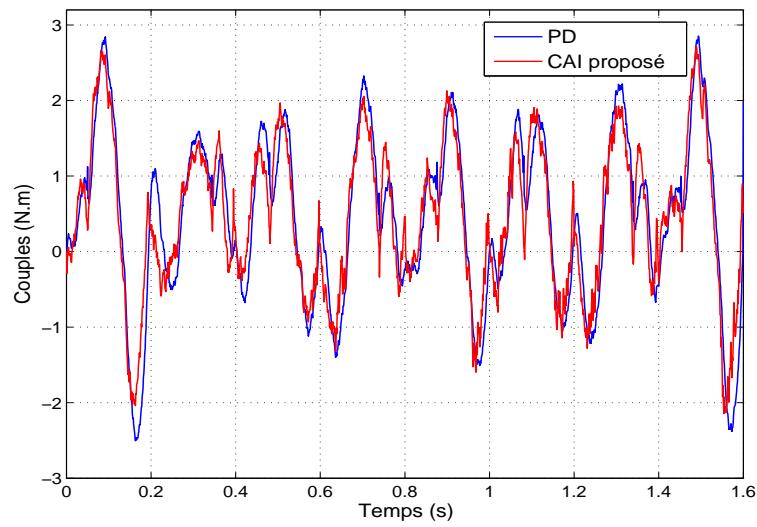


(c)

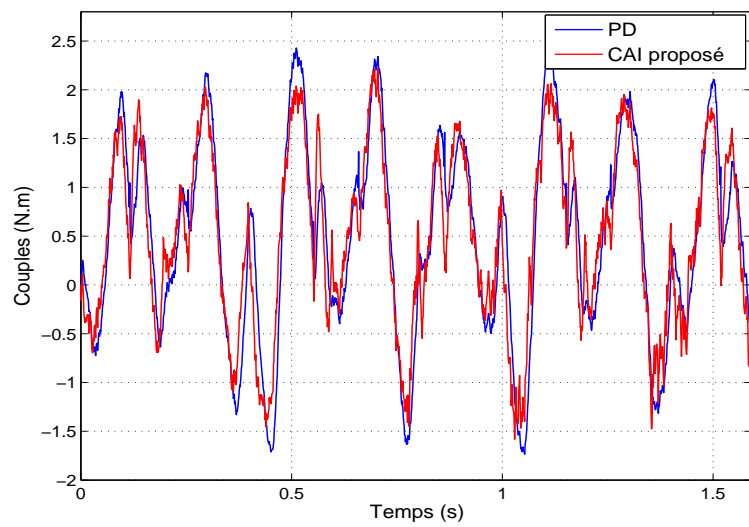
Figure 4.3 – Critère MaxAE suivant les itérations pour le 1-ère cas. (a) articulation 1, (b) articulation 2, (c) articulation 3



(a)

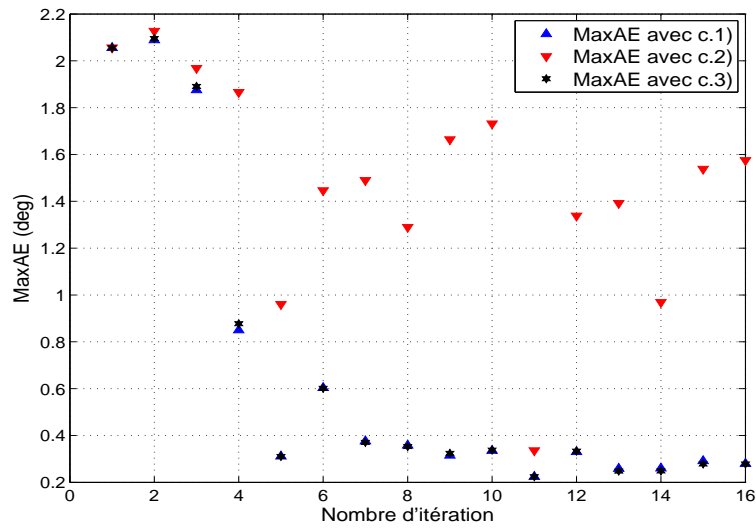


(b)

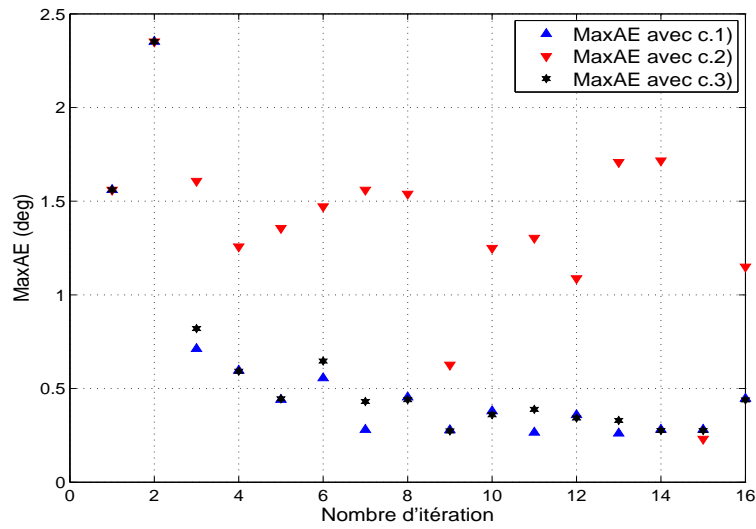


(c)

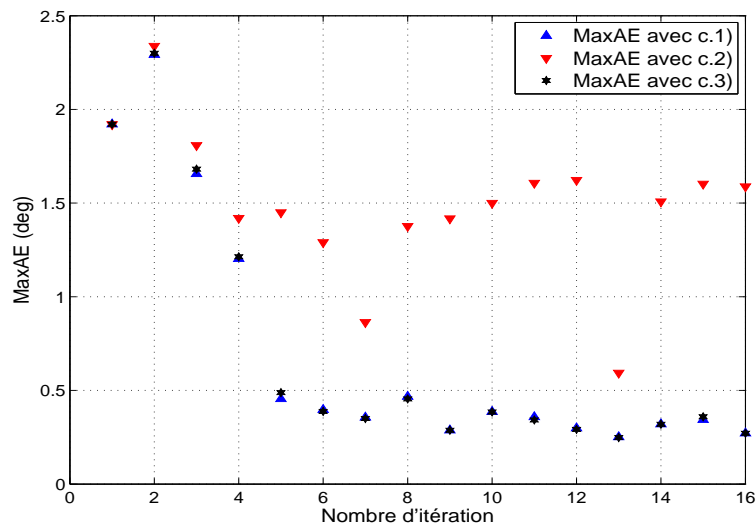
Figure 4.4 – Entrées de commande pour le 1-ère cas. (a) articulation 1, (b) articulation 2, (c) articulation



(a)

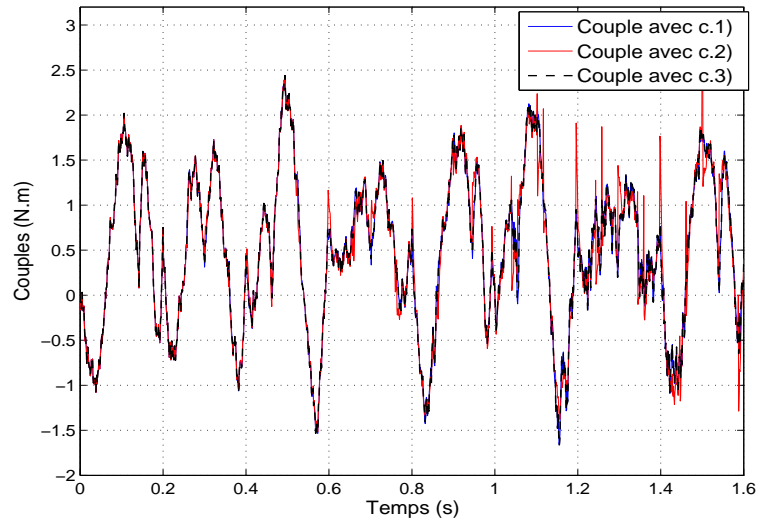


(b)

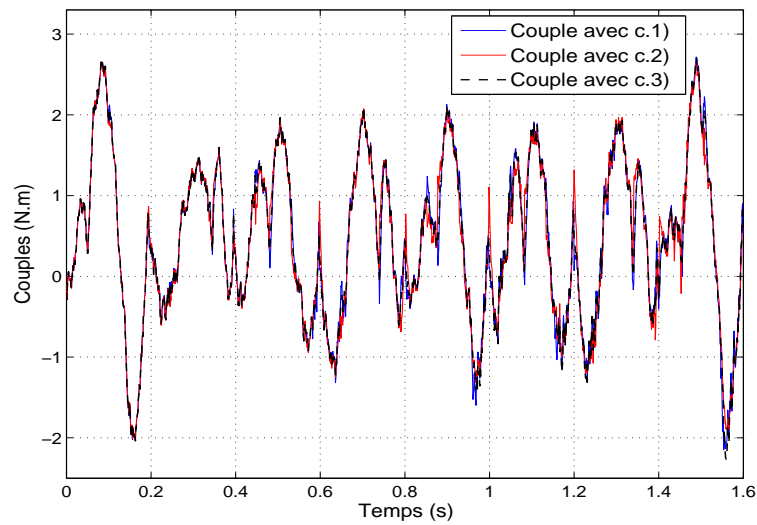


(c)

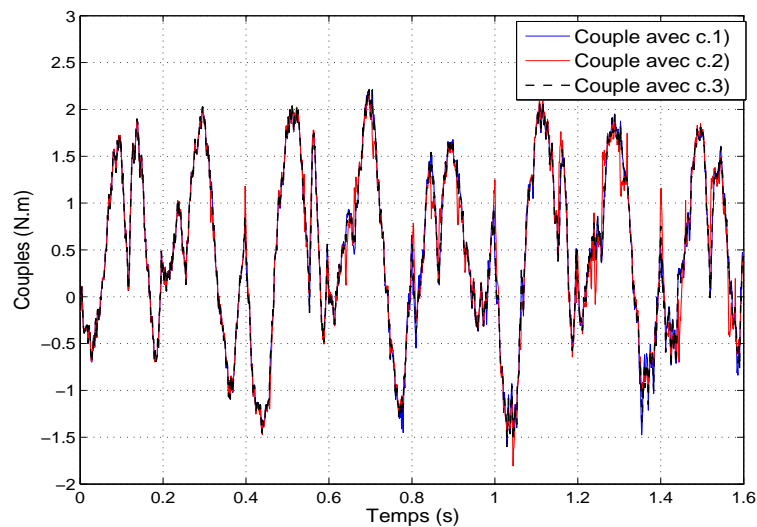
Figure 4.5 – Critère MaxAE suivant les itérations pour le 2-ème cas. (a) articulation 1, (b) articulation 2, (c) articulation 3



(a)



(b)



(c)

Figure 4.6 – Entrées de commande pour le 2-ème cas. (a) articulation 1, (b) articulation 2, (c) articulation

dérablement malgré le fait qu' il a un couple presque comme (4.16) et (4.18). Cependant, même avec la convergence lente de (4.17), l'erreur de poursuite pour les trois algorithmes approche zéro lorsque le nombre d'itérations augmente.

4.4 Commande par apprentissage itératif sans modèle pour les systèmes non linéaires MIMO

Pour des raisons de sécurité, beaucoup de systèmes dynamiques fonctionnant en temps réel sont soumis à des contraintes sur leurs variables d'état. Dans les robots manipulateurs, ces contraintes sont souvent imposées sur les positions et les vitesses articulaires afin de préserver leur structure mécanique et éviter les positions de singularité. Peu de travaux de recherche sur la CAI sous contraintes sont disponibles dans la littérature. On peut citer à titre d'exemple la CAI prédictive [46], la CAI adaptative [7], et la CAI optimal [36]. Cependant, ces approches sont développées uniquement dans le cas d'une trajectoire répétitive, ce qui limite leur champs d'application.

Dans cette partie, nous proposons une technique de commande CAI sous contraintes sur les grandeurs d'état et avec trajectoire non répétitive dans un environnement perturbé. Nous montrons que l'erreur de poursuite s'améliore avec les itérations malgré le changement de la forme de la trajectoire d'une itération à l'autre en utilisant uniquement les erreurs des itérations passées sans pour autant avoir besoin du modèle dynamique.

Par rapport aux travaux développés dans le domaine de la commande CAI à trajectoire non répétitive [6, 6, 13, 18, 21–23], l'approche que nous proposons évite l'hypothèse classique de réinitialisation de l'erreur. En outre, la trajectoire peut varier en amplitude, en temps d'exécution et en coordonnées de départ et d'arrivée. Une transformation dans l'échelle de temps est utilisée pour faciliter l'analyse de la convergence. L'approche développée est validée expérimentalement sur le robot Delta ISIS88.

4.4.1 Conception de la commande

Considérons le système MIMO de second ordre décrit par l'équation suivante :

$$\begin{cases} \dot{x}_{1,k}(t_k) = x_{2,k}(t_k) \\ \dot{x}_{2,k}(t_k) = f_k(x_k(t_k)) + B_k(x_k(t_k))u_k(t_k) + \eta_k(t_k) \\ y_k(t_k) = x_{1,k}(t_k), \quad t_k \in [0, T.T_k] \end{cases} \quad (4.48)$$

La trajectoire désirée dans la k-ème itération est définie comme suit

$$x_{r,k}(t_k) = [x_{r,k,1}(t_k), \dots, x_{r,k,\frac{n}{2}}(t_k)] \in \mathbb{R}^{\frac{n}{2}}$$

L'erreur de poursuite en position $e_{1,k}(t_k) = [e_{1,k,1}(t_k), \dots, e_{1,k,\frac{n}{2}}(t_k)] \in \mathbb{R}^{\frac{n}{2}}$ et l'erreur de poursuite en vitesse $e_{2,k}(t_k) = [e_{2,k,1}(t_k), \dots, e_{2,k,\frac{n}{2}}(t_k)] \in \mathbb{R}^{\frac{n}{2}}$ sont définis par :

$$e_{1,k}(t_k) = x_{r,k}(t_k) - x_{1,k}(t_k) \quad (4.49)$$

$$e_{2,k}(t_k) = \dot{x}_{r,k}(t_k) - x_{2,k}(t_k), \quad t_k \in [0, T.T_k] \quad (4.50)$$

La transformation de temps t est exprimée comme suit : $t = \frac{t_k}{T_k}$, où $t \in [0, T]$.

La variable $\rho_k \in \mathbb{R}$ est définie comme suit :

$$\rho_k(t) = t_k = T_k t \quad (4.51)$$

Ainsi, le système (4.48) peut se réécrire avec la transformation de temps t , comme suit :

$$\begin{cases} \dot{x}_{1,k}(\rho_k(t)) = x_{2,k}(\rho_k(t))\dot{\rho}_k(t) \\ \dot{x}_{2,k}(\rho_k(t)) = (f_k(x_k(\rho_k(t))) + B_k(x_k(\rho_k(t)))u_k(\rho_k(t)) + \theta_k(\rho_k(t)))\dot{\rho}_k(t) \\ y_k = x_{1,k}(\rho_k(t)), \quad t \in [0, T] \end{cases} \quad (4.52)$$

On définit les vecteurs : $X_k(t) \triangleq x_k(\rho_k(t))$, $X_{1,k}(t) \triangleq x_{1,k}(\rho_k(t))$, $X_{2,k}(t) \triangleq x_{2,k}(\rho_k(t))$, $\tilde{q}_{1,k}(t) \triangleq e_{1,k}(\rho_k(t))$, $\tilde{q}_{2,k}(t) \triangleq e_{2,k}(\rho_k(t))$, $d_k(t) \triangleq \eta_k(\rho_k(t))$, $X_{r,k}(t) \triangleq x_{r,k}(\rho_k(t))$, $v_k(t) \triangleq u_k(\rho_k(t))$.

Ainsi, (4.52) peut se réécrire comme suit :

$$\begin{cases} \dot{X}_{1,k}(t) = X_{2,k}(t)\dot{\rho}_k(t) \\ \dot{X}_{2,k}(t) = (f_k(X_k) + B_k(X_k)v_k(t) + d_k(t))\dot{\rho}_k(t) \\ y_k = X_{1,k}(t), \quad t \in [0, T] \end{cases} \quad (4.53)$$

La dynamique des erreurs (4.49) et (4.50) en respectant à la transformation t est donnée par :

$$\dot{\tilde{q}}_{1,k}(t) = \dot{e}_{1,k}(\rho_k(t))\dot{\rho}_k(t) = \tilde{q}_{2,k}(t)\dot{\rho}_k(t), \quad t \in [0, T] \quad (4.54)$$

$$\dot{\tilde{q}}_{2,k}(t) = \dot{e}_{2,k}(\rho_k(t))\dot{\rho}_k(t) = \ddot{X}_{r,k}(t)\dot{\rho}_k(t)^{-1} - \dot{X}_{2,k}(t) \quad (4.55)$$

Les définitions des normes de vecteur/ matrice suivantes sont utilisées :

$$\|Y\|_\infty = \max|\varepsilon_i|, \quad \|A\|_{m_\infty} = n \max(a_{ij})$$

où, $Y = [\varepsilon_1, \dots, \varepsilon_n]^T \in \mathbb{R}^n$, et $A = [a_{ij}] \in \mathbb{R}^{n \times n}$.

Nous allons utiliser l'hypothèse (1) et les hypothèses suivantes pour démontrer la convergence. Ces hypothèses sont faites par rapport à la transformation t afin de les utiliser directement dans l'analyse de la convergence.

Hypothèse 4 : La matrice $B_k(X_k(t))$ est inversible, bornée, et définie positive ou négative. Sans perte de généralité, nous supposons $B_k(X_k(t))$ définie positive et $B_k^{-1}(X_k(t))$ est différentiable et sa dérivée par rapport au temps est bornée. ie.

$$\|B_k^{-1}(X_k(t))\|_{m_\infty} \leq l_b, \quad \left\| \frac{dB_k^{-1}(X_k(t))}{dt} \right\|_{m_\infty} \leq l_{bd} \forall t \in [0, T], \forall k \in \mathbb{Z}_+$$

où, l_b , et l_{bd} sont des constantes positives inconnues.

Hypothèse 5 : La fonction non linéaire $f_k(X_k(t))$ satisfait le croisement linéaire de l'état (*the linear growth condition*)

$$\|f_k(X_k(t))\|_\infty \leq c_1 \|x_k\|_\infty + c_2, \quad \forall t \in [0, T], \text{ and } \forall k \in \mathbb{Z}_+$$

où c_1 et c_2 sont des constantes positives inconnues.

Hypothèse 6 : Il existe une fonction inconnue, lisse et positive $d(t) \in \mathbb{R}$ satisfait :

$$\|d_k(t)\|_\infty \leq d(t) \quad \forall k \in \mathbb{Z}_+ \quad (4.56)$$

Hypothèse 7 : La trajectoire désirée $X_{r,k}(t_k)$ et ses dérivées jusqu'à la n-ème sont continues, et bornées, ie.

$$\|X_{r,k}(t)\|_\infty \leq l_{xr}, \quad \|\dot{X}_{r,k}(t)\|_\infty \leq l_{xr}$$

,

$$\|\ddot{X}_{r,k}(t)\|_\infty \leq l_{xr}, \quad \forall t \in [0, T], \forall k \in \mathbb{Z}_+$$

où l_{xr} est une constante positive inconnue.

Hypothèse 8 : L'état $X_k(t)$ est sous la contrainte

$$\|X_k(t)\|_\infty \leq l_x, \|\dot{X}_k(t)\|_\infty \leq l_x,$$

$$\|\ddot{X}_k(t)\|_\infty \leq l_x, \forall t \in [0, T], \forall k \in \mathbb{Z}_+$$

où l_x est une constante positive connue ou inconnue.

Remarque 5 : L'hypothèse 5 semble être très raisonnable et pratique, en particulier quand c_1 et c_2 sont inconnus. Cette hypothèse a été proposée dans beaucoup de littératures [6].

Remarque 6 : L'hypothèse 8 indique que le système est soumis aux contraintes d'état. Les limites de ces contraintes ne sont pas nécessairement connues et le régulateur proposé peut fonctionner effectivement sous des contraintes inconnues.

L'objectif de cette partie est de concevoir une CAI telle que :

1. Les systèmes non linéaires MIMO soumis aux perturbations non répétitives peuvent être considérés.
2. L'itération actuelle peut bénéficier des itérations précédentes malgré la variation dans la durée d'itération, dans l'amplitude et dans la position initiale et finale de la trajectoire désirée.
3. La commande peut fonctionner efficacement indépendamment du modèle et/ou de ses bornées maximales.
4. L'erreur de poursuite en position et en vitesse convergent vers zéro avec l'augmentation du nombre des itérations.

La loi CAI proposée est donnée comme suit :

$$u_k(t_k) = Ks_k(t_k) + \zeta(s_k(t_k))\hat{\theta}_k(t_k), t_k \in [0, T.T_k] \quad (4.57)$$

$$\hat{\theta}_k(t_k) = \hat{\theta}_{k-1}(t_k) + \Gamma\zeta^T(s_k(t_k))s_k(t_k)\dot{\rho}_k \quad (4.58)$$

où, $\hat{\theta}_0 = 0 \in \mathbb{R}^2$, $s_k(t_k) = e_{2,k}(t_k) + \lambda e_{1,k}(t_k)$, et $\zeta(s_k(t_k)) = [s_k(t_k), \text{sgn}(s_k(t_k))] \in \mathbb{R}^{\frac{n}{2} \times 2}$. La matrice de retour d'état $K \in \mathbb{R}^{\frac{n}{2} \times \frac{n}{2}}$ est définie positive, et la matrice gain d'apprentissage $\Gamma \in \mathbb{R}^{2 \times 2}$ est définie positive et λ est une constante positive.

La loi de commande donnée par (4.57) et (4.58) peut s'exprimer en respectant la transformation t comme suit :

$$v_k(t) = Kz_k(t) + \zeta(z_k(t))\hat{\theta}_k(t) \quad (4.59)$$

$$\hat{\theta}_k(t) = \hat{\theta}_{k-1}(t) + \Gamma\zeta^T(z_k(t))z_k(t)\dot{\rho}_k, t \in [0, T] \quad (4.60)$$

où,

$$z_k(t) = \tilde{q}_{2,k}(t) + \lambda\tilde{q}_{1,k}(t) \quad (4.61)$$

Et $z_k(t) \triangleq s_k(t_k)$, $\hat{\theta}_k(t) \triangleq \hat{\theta}_k(t_k)$.

4.4.2 Analyse de la convergence

Théorème 5 : Considérons le système MIMO non linéaire incertain (4.1), sous la loi de commande (4.57) et (4.58). Si les hypothèses 1, 4-8, sont satisfaites, alors l'erreur de

poursuite de la trajectoire non répétitive décrite dans la **définition** converge vers zéro quand le nombre d'itérations k augment infiniment sur un intervalle de temps fini $[0, T.T_k]$, i.e., $\lim_{k \rightarrow \infty} e_{1,k}(t_k) = \lim_{k \rightarrow \infty} e_{2,k}(t_k) = 0 \forall t_k \in [0, T.T_k]$.

Preuve

Considérons la fonction de Lyapunov suivante

$$w_k(t) = \frac{1}{2} z_k^T(t) B_k^{-1}(X_k(t)) z_k(t) + \frac{1}{2} \int_0^t \tilde{\vartheta}_k^T \Gamma^{-1} \tilde{\vartheta}_k d\tau \quad (4.62)$$

où, $\tilde{\vartheta}_k(t) = \vartheta(t) - \hat{\vartheta}_k(t)$, et le vecteur ϑ s'exprime par

$$\vartheta = [\alpha(t), \beta(t)]^T \in \mathbb{R}^2 \quad (4.63)$$

où $\alpha(t)$, et $\beta(t)$ sont donnés comme suit

$$\alpha(t) = \frac{n}{4} \dot{\rho}_k^{-1} l_{bd} \quad (4.64)$$

$$\beta(t) = \frac{n}{2} (l_b l_{xr} \dot{\rho}_k^{-2} + l_b l_x c_1 + l_b c_2 + l_b d(t) + l_b \lambda_m l_{xr} + l_b \lambda_m l_x) \quad (4.65)$$

λ_m indique la valeur propre maximale de la matrice λ .
Considérons la différence de $w_k(t)$ à la k -ème itération

$$\begin{aligned} \Delta w_k(t) &= w_k(t) - w_{k-1}(t) \\ &= \frac{1}{2} z_k^T(t) B_k^{-1}(X_k(t)) z_k(t) - \frac{1}{2} z_{k-1}^T(t) B_{k-1}^{-1}(X_{k-1}(t)) z_{k-1}(t) \\ &\quad + \frac{1}{2} \int_0^t (\tilde{\vartheta}_k^T \Gamma^{-1} \tilde{\vartheta}_k - \frac{1}{2} \tilde{\vartheta}_{k-1}^T \Gamma^{-1} \tilde{\vartheta}_{k-1}) d\tau \end{aligned} \quad (4.66)$$

Puisque nous avons

$$\begin{aligned} \frac{1}{2} z_k^T(t) B_k^{-1}(X_k(t)) z_k(t) &= \frac{1}{2} z_k^T(0) B_k^{-1}(X_k(0)) z_k(0) + \\ &\quad \int_0^t z_k^T B_k^{-1} \dot{z}_k d\tau + \frac{1}{2} \int_0^t z_k^T \frac{dB_k^{-1}}{dt} z_k d\tau \end{aligned} \quad (4.67)$$

Donc, à partir de (4.53), (4.54), (4.55), (4.61) et (4.67) nous pouvons obtenir

$$\begin{aligned} \frac{1}{2} z_k^T(t) B_k(X_k(t))^{-1} z_k(t) &= \frac{1}{2} z_k^T(0) B_k^{-1}(X_k(0)) z_k(0) + \\ &\quad \int_0^t z_k^T (B_k^{-1} \ddot{X}_{r,k} \dot{\rho}_k^{-1} - B_k^{-1} f_k \dot{\rho}_k - B_k^{-1} d_k \dot{\rho}_k + \frac{1}{2} \frac{dB_k^{-1}}{dt} z_k + B_k^{-1} \lambda \tilde{q}_{2,k} \dot{\rho}_k - \\ &\quad v_k \dot{\rho}_k - B_k^{-1} d_k \dot{\rho}_k + \frac{1}{2} \frac{dB_k^{-1}}{dt} z_k + \\ &\quad B_k^{-1} \lambda \tilde{q}_{2,k} \dot{\rho}_k) d\tau \end{aligned} \quad (4.68)$$

Les définitions des normes de vecteurs et de matrices mènent à

$$\begin{aligned} & z_k^T (B_k^{-1} \ddot{X}_{r,k} \dot{\rho}_k^{-1} - B_k^{-1} f_k \dot{\rho}_k - B_k^{-1} d_k \dot{\rho}_k + \frac{1}{2} \frac{dB_k^{-1}}{dt} z_k + B_k^{-1} \lambda \tilde{q}_{2,k} \dot{\rho}_k) \\ & \leq \| \dot{\rho}_k z_k^T (B_k^{-1} \ddot{X}_{r,k} \dot{\rho}_k^{-2} - B_k^{-1} f_k - B_k^{-1} d_k + \frac{1}{2} \frac{dB_k^{-1}}{dt} z_k \dot{\rho}_k^{-1} + B_k^{-1} \lambda \tilde{q}_{2,k}) \|_{\infty} \\ & \leq \frac{n}{2} \| \dot{\rho}_k z_k^T \|_{\infty} \| (B_k^{-1} \ddot{X}_{r,k} \dot{\rho}_k^{-2} - B_k^{-1} f_k - B_k^{-1} d_k + \frac{1}{2} \frac{dB_k^{-1}}{dt} z_k \dot{\rho}_k^{-1} + \\ & \quad B_k^{-1} \lambda \tilde{q}_{2,k}) \|_{\infty} \\ & \leq \frac{n}{2} \| \dot{\rho}_k z_k^T \|_{\infty} (\| B_k^{-1} \|_{m_{\infty}} \| \ddot{X}_{r,k} \dot{\rho}_k^{-2} \|_{\infty} + \| B_k^{-1} \|_{m_{\infty}} \| f_k \|_{\infty} + \\ & \quad \| B_k^{-1} \|_{m_{\infty}} \| d_k \|_{\infty} + \frac{1}{2} \frac{dB_k^{-1}}{dt} \|_{m_{\infty}} \| z_k \dot{\rho}_k^{-1} \|_{\infty} + \| B_k^{-1} \lambda \|_{m_{\infty}} \| \tilde{q}_{2,k} \|_{\infty}) \end{aligned} \quad (4.69)$$

Selon les hypothèses (4-8) nous pouvons obtenir

$$\begin{aligned}
& z_k^T (B_k^{-1} \ddot{X}_{r,k} \dot{\rho}_k^{-1} - B_k^{-1} f_k \dot{\rho}_k - B_k^{-1} d_k \dot{\rho}_k + \frac{1}{2} \frac{dB_k^{-1}}{dt} z_k + B_k^{-1} \lambda \tilde{q}_{2,k} \dot{\rho}_k) \\
& \leq \frac{n}{2} \|\dot{\rho}_k z_k^T\|_\infty (l_b l_{xr} \dot{\rho}_k^{-2} + l_b c_1 \|x_k\|_\infty + l_b c_2 + l_b d(t) + \\
& \frac{1}{2} \dot{\rho}_k^{-1} l_{bd} \|z_k\|_\infty + l_b \lambda_m \|X_{2,r}\|_\infty + l_b \lambda_m \|X_{2,k}\|_\infty) \\
& \leq \|\dot{\rho}_k z_k^T\|_\infty (\frac{n}{2} l_b l_{xr} \dot{\rho}_k^{-2} + \frac{n}{2} l_b l_x c_1 + \frac{n}{2} l_b c_2 + \frac{n}{2} l_b d(t) + \\
& \frac{n}{2} \dot{\rho}_k^{-1} l_{bd} \|z_k\|_\infty + \frac{n}{2} l_b \lambda_m l_{xr} + \frac{n}{2} l_b \lambda_m l_x)
\end{aligned} \tag{4.70}$$

Par conséquent, à partir de (4.63), (4.64), (4.65), et (4.70) on obtient

$$\begin{aligned}
& z_k^T (B_k^{-1} \ddot{X}_{r,k} \dot{\rho}_k^{-1} - B_k^{-1} f_k \dot{\rho}_k - B_k^{-1} d_k \dot{\rho}_k + \frac{1}{2} \frac{dB_k^{-1}}{dt} z_k + B_k^{-1} \lambda \tilde{q}_{2,k} \dot{\rho}_k) \\
& \leq \|\dot{\rho}_k z_k^T\|_\infty (\beta + \alpha \|z_k^T\|_\infty) \leq \dot{\rho}_k z_k^T (\beta \text{sgn}(z_k) + \alpha z_k) \leq \dot{\rho}_k z_k^T \xi(z_k) \vartheta
\end{aligned} \tag{4.71}$$

D'après le dernier terme de (4.66), on peut obtenir

$$\begin{aligned}
& \frac{1}{2} \int_0^t \tilde{\vartheta}_k^T \Gamma^{-1} \tilde{\vartheta}_k d\tau - \frac{1}{2} \int_0^t \tilde{\vartheta}_{k-1}^T \Gamma^{-1} \tilde{\vartheta}_{k-1} d\tau = \\
& \int_0^t (\hat{\vartheta}_k - \vartheta)^T \Gamma^{-1} (\hat{\vartheta}_k - \hat{\vartheta}_{k-1}) - \frac{1}{2} (\hat{\vartheta}_k - \hat{\vartheta}_{k-1})^T \Gamma^{-1} (\hat{\vartheta}_k - \hat{\vartheta}_{k-1}) d\tau
\end{aligned} \tag{4.72}$$

La substitution de la loi de commande (4.60) dans (4.72) conduit à

$$\begin{aligned}
& \frac{1}{2} \int_0^t \tilde{\vartheta}_k^T \Gamma^{-1} \tilde{\vartheta}_k - \frac{1}{2} \int_0^t \tilde{\vartheta}_{k-1}^T \Gamma^{-1} \tilde{\vartheta}_{k-1} d\tau = - \int_0^t \dot{\rho}_k z_k^T \xi(z_k) \tilde{\vartheta}_k \\
& \quad - \frac{1}{2} \dot{\rho}_k^2 z_k^T \xi(z_k) \Gamma^T \xi^T(z_k) z_k d\tau
\end{aligned} \tag{4.73}$$

Puis, par la combinaison de (4.66), (4.71) et (4.73) nous obtenons

$$\begin{aligned}
& w_k(t) - w_{k-1}(t) \leq \frac{1}{2} z_k^T(0) B_k^{-1} (X_k(0)) z_k(0) + \\
& \int_0^t \dot{\rho}_k z_k^T (\xi(z_k) \vartheta - v_k - \xi(z_k) \tilde{\vartheta}_k) d\tau - \\
& \int_0^t \frac{1}{2} \dot{\rho}_k^2 z_k^T \xi(z_k) \Gamma^T \xi^T(z_k) z_k d\tau - \frac{1}{2} z_{k-1}^T B_{k-1}^{-1} (X_{k-1}) z_{k-1}
\end{aligned} \tag{4.74}$$

En appliquant la loi de commande (4.59), l'inégalité (4.74) devient

$$\begin{aligned}
& w_k(t) - w_{k-1}(t) \leq \frac{1}{2} z_k^T(0) B_k^{-1} (X_k(0)) z_k(0) - \int_0^t \dot{\rho}_k z_k^T K z_k d\tau \\
& - \int_0^t \frac{1}{2} \dot{\rho}_k^2 z_k^T \xi(z_k) \Gamma^T \xi^T(z_k) z_k d\tau - \frac{1}{2} z_{k-1}^T B_{k-1}^{-1} (X_{k-1}) z_{k-1}
\end{aligned} \tag{4.75}$$

Notons que $w_k(t)$ peut être écrit comme suit

$$w_k(t) = w_1(t) + \sum_{i=2}^k \Delta w_i(t) \tag{4.76}$$

Par conséquent, en utilisant (4.75) on aura

$$\begin{aligned}
& w_k(t) \leq w_1(t) + \sum_{i=2}^k (\frac{1}{2} z_i^T(0) B_i^{-1} (X_i(0)) z_i(0) - \\
& \frac{1}{2} z_{i-1}^T(t) B_{i-1}^{-1} (X_{i-1}(t)) z_{i-1}(t) - \int_0^t \dot{\rho}_i z_i^T K z_i - \\
& \frac{1}{2} \dot{\rho}_i^2 z_i^T \xi(z_i) \Gamma^T \xi^T(z_i) z_i d\tau)
\end{aligned} \tag{4.77}$$

En basant sur l'hypothèse 1 nous pouvons obtenir

$$\frac{1}{2} z_i^T(0) B_i^{-1} (X_i(0)) z_i(0) = \frac{1}{2} z_{i-1}^T(T) B_{i-1}^{-1} (X_{i-1}(T)) z_{i-1}(T), i = 2 : k \tag{4.78}$$

Par substitution de (4.77) dans (4.78) on trouve

$$w_k(T) \leq w_1(T) - \sum_{i=2}^k \dot{\rho}_i K_m \int_0^T \|z_i\|^2 d\tau \quad (4.79)$$

où K_m indique la valeur propre minimale de la matrice K .

Nous allons maintenant vérifier la bornitude de $w_1(T)$.
À partir de la définition de $w_k(t)$, nous avons

$$w_1(t) = \frac{1}{2} z_1^T(t) B_1^{-1}(X_1(t)) z_1(t) + \frac{1}{2} \int_0^t \tilde{\vartheta}_1^T \Gamma^{-1} \tilde{\vartheta}_1 d\tau \quad (4.80)$$

La dérivée par rapport au temps de $w_1(t)$ est donnée comme suit

$$\dot{w}_1(t) = z_1^T(t) B_1^{-1}(X_1(t)) \dot{z}_1(t) + \frac{1}{2} z_1^T(t) \frac{dB_1^{-1}(X_1(t))}{dt} z_1(t) + \frac{1}{2} \tilde{\vartheta}_1^T \Gamma^{-1} \dot{\tilde{\vartheta}}_1 \quad (4.81)$$

En utilisant (4.60), (4.73) et sachant que $\hat{\vartheta}_0 = [0, 0]^T$, on peut avoir

$$\begin{aligned} \frac{1}{2} \tilde{\vartheta}_1^T \dot{\tilde{\vartheta}}_1 &= \frac{1}{2} (\tilde{\vartheta}_1^T \dot{\tilde{\vartheta}}_1 - \tilde{\vartheta}_0^T \dot{\tilde{\vartheta}}_0) + \frac{1}{2} \tilde{\vartheta}_0^T \dot{\tilde{\vartheta}}_0 \\ &= -\dot{\rho}_1 z_1^T \zeta(z_1) \tilde{\vartheta}_1 - \frac{1}{2} \dot{\rho}_1^2 z_1^T \zeta(z_1) \Gamma^T \zeta^T(z_1) z_1 + \frac{1}{2} \vartheta^T \vartheta \end{aligned} \quad (4.82)$$

par conséquent

$$\begin{aligned} \dot{w}_1(t) &\leq -\dot{\rho}_1 z_1^T K z_1 - \frac{1}{2} \dot{\rho}_1^2 z_1^T \zeta(z_1) \Gamma^T \zeta^T(z_1) z_1 + \frac{1}{2} \vartheta^T \vartheta \\ &\leq \frac{1}{2} \vartheta^T \vartheta \end{aligned} \quad (4.83)$$

Comme $\vartheta^T \vartheta$ est positif continu sur l'intervalle de temps $[0, T]$, donc on peut avoir une constante telle que

$$L = \frac{1}{2} \int_0^T \vartheta^T \vartheta d\tau \leq \infty$$

D'où, $w_1(t)$ peut être borné comme suit

$$\begin{aligned} 0 \leq w_1(t) &= w_1(0) + \int_0^t \dot{w}_1(\tau) d\tau \leq w_1(0) + \frac{1}{2} \int_0^t \vartheta^T \vartheta d\tau \leq \\ &w_1(0) + L \leq \infty \end{aligned} \quad (4.84)$$

Comme, $w_1(0) = \frac{1}{2} \lambda(X_{r,1}(0) - X_{1,1}(0))^T B(X_0(0)) \lambda(X_{r,1}(0) - X_{1,1}(0))$ est borné, car $X_{r,1}(0)$, $B(X_0(0))$ et $X_{1,1}(0)$ sont bornés (le système ne s'est pas encore déplacé). Ainsi $w_1(t)$ est borné sur $t \in [0, T]$, par conséquent, la bornitude de $w_1(T)$.

Comme la partie gauche de l'équation (4.79) est positive et $w_1(T)$ est borné, on conclut

$$\|z_k\|^2 \rightarrow 0 \quad \text{quand } k \rightarrow \infty, \forall t \in [0, T] \quad (4.85)$$

À partir de (4.85) on obtient

$$\tilde{q}_{2,k}(t) + \lambda \tilde{q}_{1,k}(t) \rightarrow 0 \quad \text{quand } k \rightarrow \infty, \forall t \in [0, T] \quad (4.86)$$

Comme nous avons $\tilde{q}_{2,k}(t) = \dot{\tilde{q}}_{1,k}(t)\dot{\rho}_k^{-1}$, alors on peut trouver

$$\dot{\tilde{q}}_{1,k} + \lambda\tilde{q}_{1,k}\dot{\rho}_k \rightarrow 0 \quad (4.87)$$

Comme résultat

$$\tilde{q}_{1,k}(t) \rightarrow 0 \text{ et } \tilde{q}_{2,k}(t) \rightarrow 0 \text{ pour } t \in [0, T] \text{ quand } k \rightarrow \infty \quad (4.88)$$

Finalement, on obtient

$$e_{1,k}(t_k) \rightarrow 0 \text{ et } e_{2,k}(t_k) \rightarrow 0 \text{ pour } t_k \in [0, T.T_k] \text{ quand } k \rightarrow \infty \quad (4.89)$$

4.4.3 Résultats expérimentaux

Cette loi de commande a été mise en œuvre pratiquement sur le robot Delta ISIR88.

Pour examiner les résultats de la CAI proposée d'une façon comparative, des expériences ont été également exécutées pour un régulateur PID de la forme :

$$\tau_k(t_k) = K_p e_{1,k}(t_k) + K_i \int_0^{t_k} e_{1,k}(\sigma) d\sigma + K_d e_{2,k}(t_k). \quad (4.90)$$

Les gains de PID ont été ajustés en utilisant le procédé essai-erreur jusque à obtenir une bonne poursuite de trajectoire et des performances semblables à celles de la CAI proposée. Les gains du PID ont été choisis comme suit : $K_p = \text{diag}\{2.4, 2.4, 2.4\}$, $K_d = \text{diag}\{0.0135, 0.0135, 0.0135\}$ et $K_i = \text{diag}\{80, 80, 80\}$.

Les gains de la commande proposée ont été sélectionnés comme suit : $K = \text{diag}\{0.0135, 0.0135, 0.0135\}$, $\lambda = \text{diag}\{195, 195, 195\}$, et $\Gamma = [0.9, 2; 9, 1] \times 10^{-6}$.

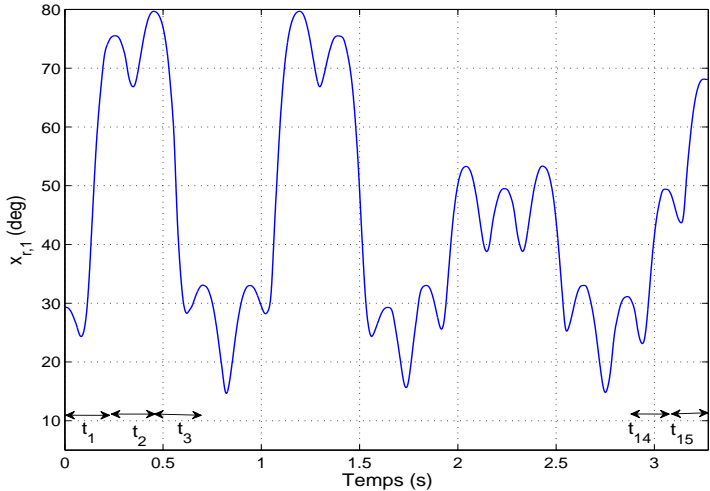
La figure 4.7 montre les trajectoires désirées dans l'espace articulaire et la figure 4.8 montre la poursuite de trajectoires dans l'espace opérationnel. Comme on peut remarquer, les trajectoires imposées varient d'une itération à l'autre en échelle de temps, en amplitude et en position initiale et finale. Le tableau 3.3 représente la durée de chaque itération.

Table 4.1 – Durée d'itérations

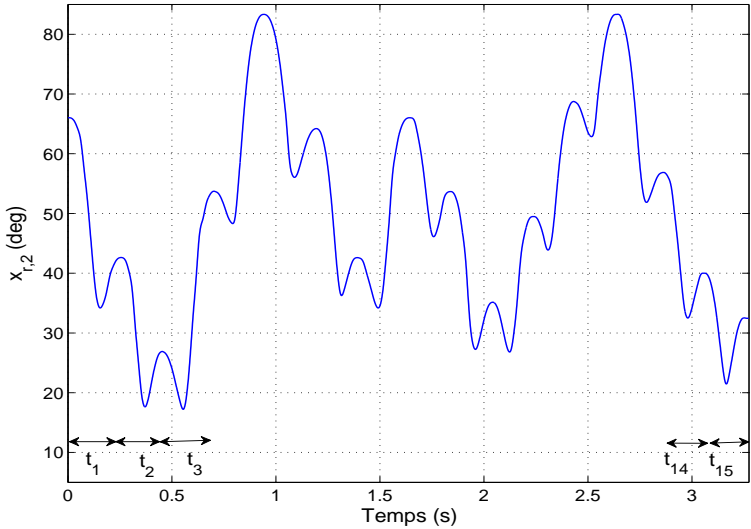
Itération	1	2	3	4	5
Durée (s)	0.255	0.197	0.255	0.237	0.255
Itération	6	7	8	9	10
Durée (s)	0.197	0.255	0.197	0.205	0.197
Itération	11	12	13	14	15
Durée (s)	0.197	0.205	0.226	0.201	0.193

La figure 4.9 montre les erreurs articulaires le long de l'axe du temps avec la commande PID et la CAI sous l'algorithme (4.16) pour le scénario c. La figure 4.10 montre le MaxAE au long de l'axe des itérations pour les trois articulations avec la commande PID et la CAI proposée. On peut observer qu'avec la commande proposée, le MaxAE a diminué au cours des itérations pour les trois articulations malgré la variation de la trajectoire désirée en durée d'itération, en amplitude et en positions initiales et finales ce qui n'est pas le cas pour la commande PID. On peut observer aussi que les algorithmes (4.16), (4.17), et (4.18) donnent des performances similaires.

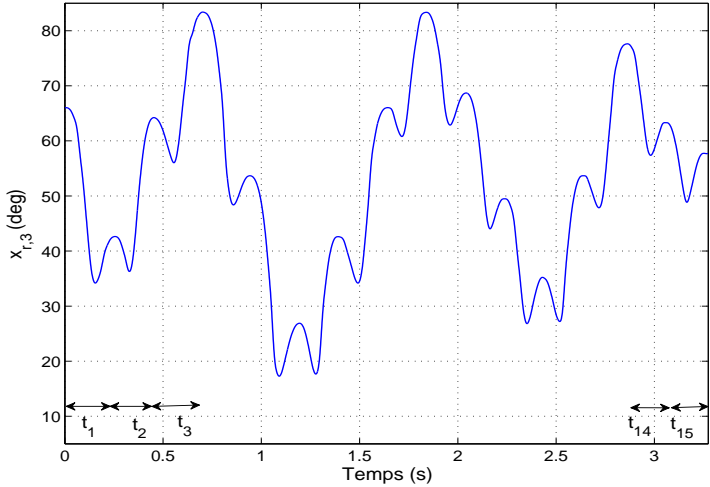
À partir de la figure 4.10 et 4.11, la commande PID et la CAI avec l'algorithme (4.16) possèdent de petits couples similaires à ceux de la CAI avec l'algorithme (4.17), et (4.18).



(a)



(b)



(c)

Figure 4.7 – Trajectoire désirée. (a) articulation 1, (b) articulation 2, (c) articulation 3

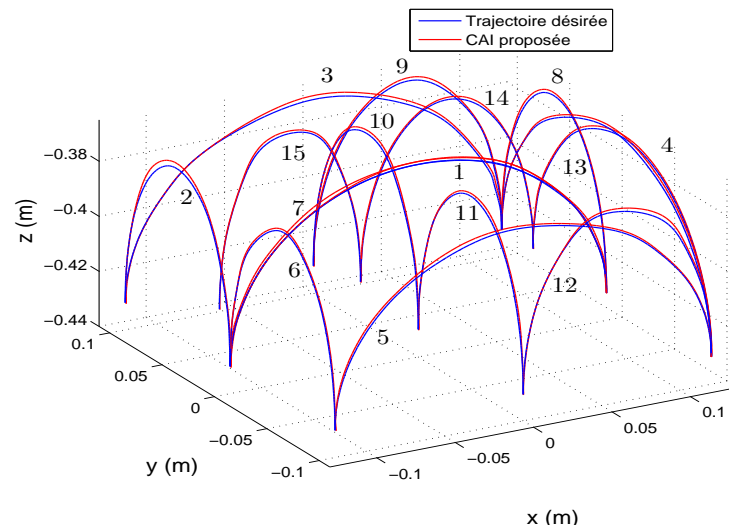
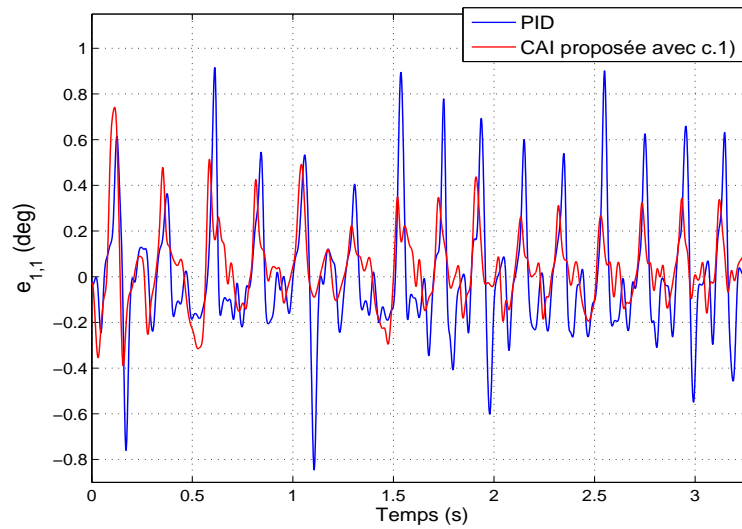


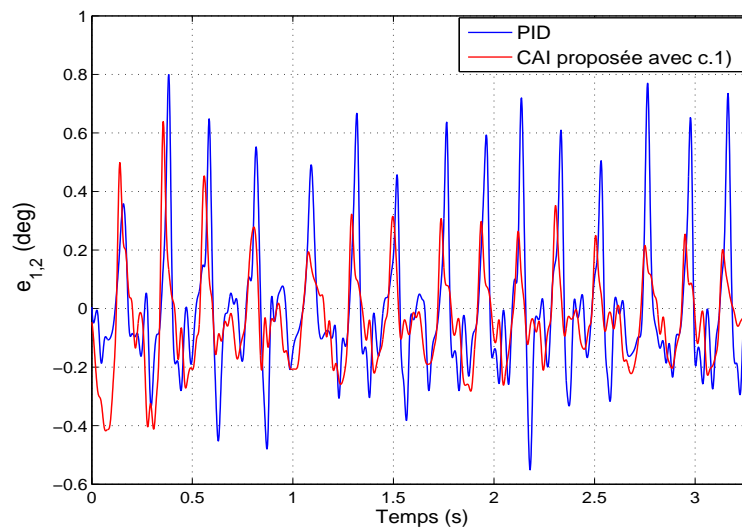
Figure 4.8 – Poursuite de trajectoire dans l'espace opérationnel en expérimentation

En outre, le signal de commande pour tous les algorithmes est borné sans apparition de broutement.

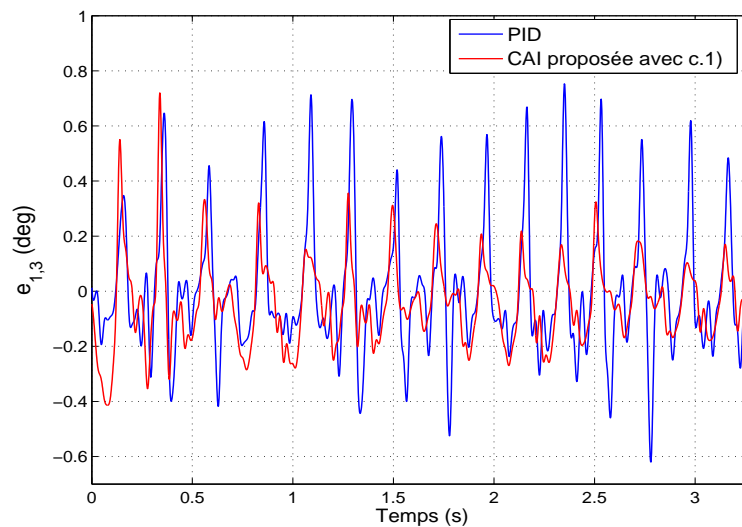
Selon les résultats expérimentaux, la commande CAI non-basée-modèle peut assurer de meilleures performances par rapport à la commande PID qui est partiellement dépendante du modèle, où l'erreur de poursuite sous les algorithmes (4.16), (4.17), et (4.18) approche zéro le long de l'axe des itérations.



(a)

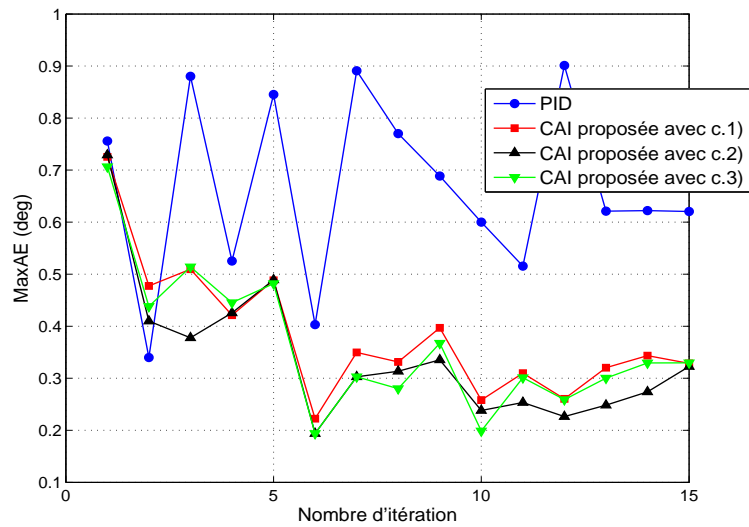


(b)

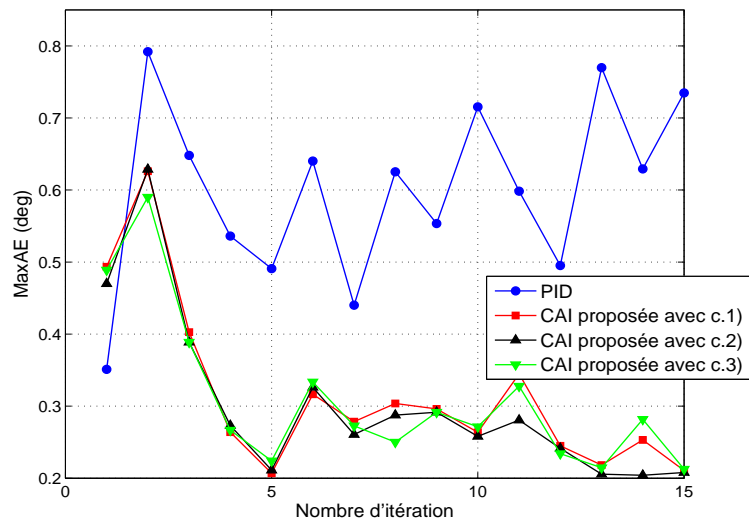


(c)

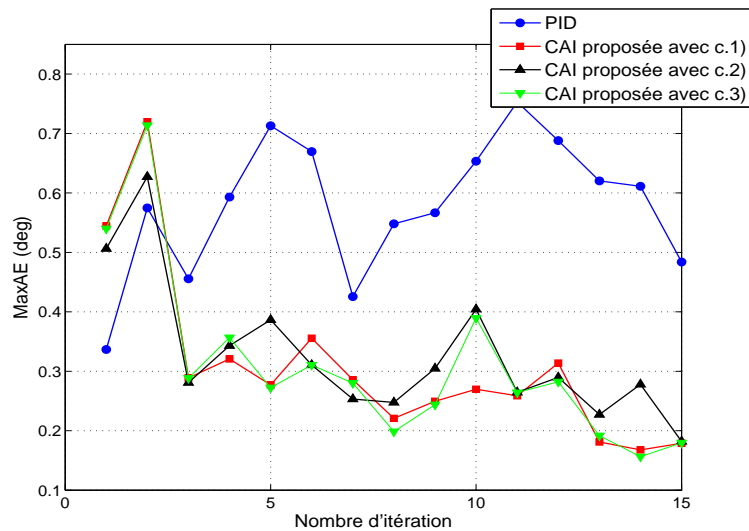
Figure 4.9 – Évolution de l'erreur de poursuite en fonction du temps. (a) articulation 1, (b) articulation 2, (c) articulation 3



(a)

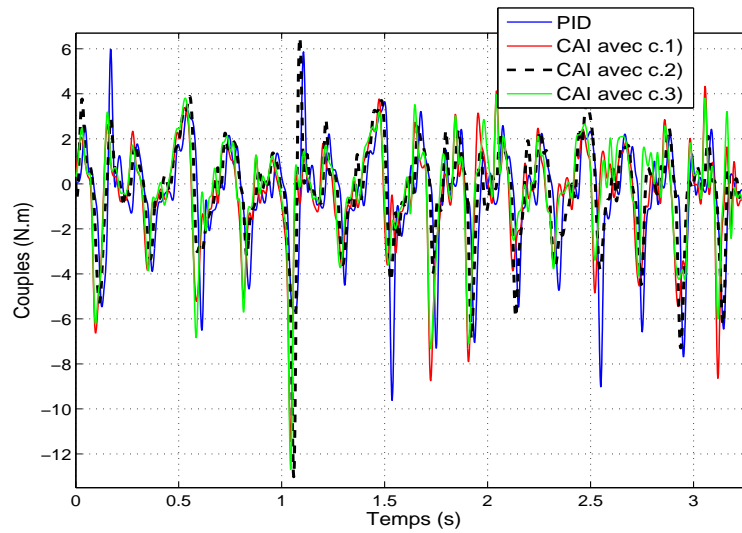


(b)

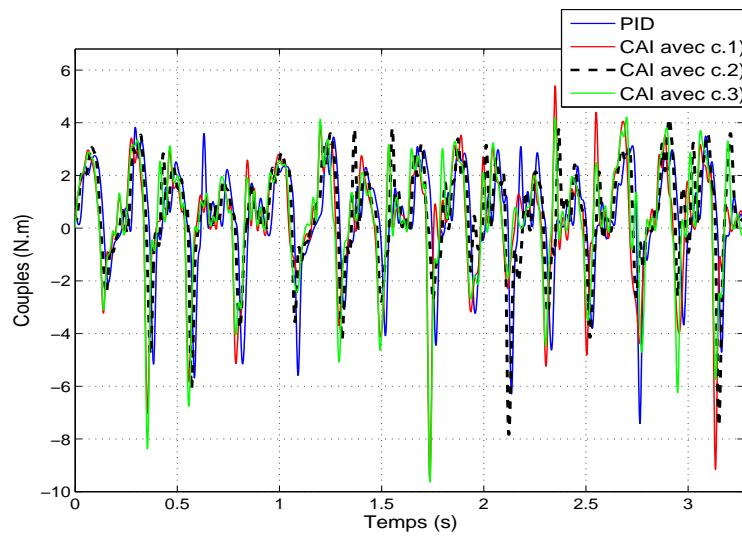


(c)

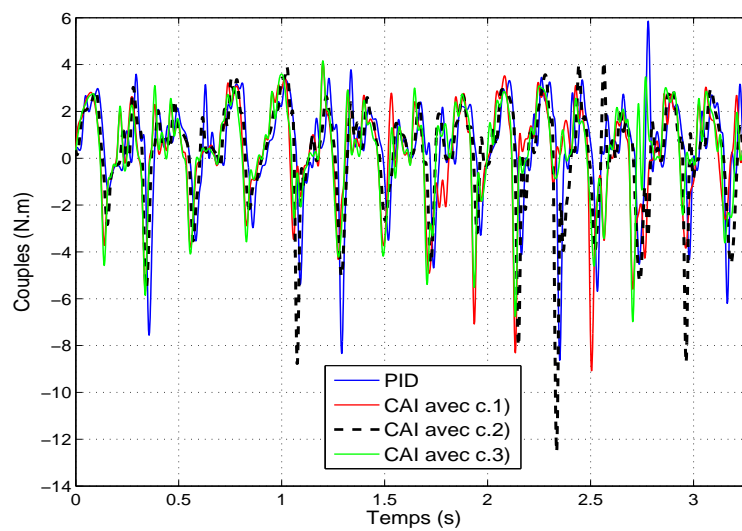
Figure 4.10 – Évolution de l'erreur de poursuite en fonction des itérations. (a) articulation 1, (b) articulation 2, (c) articulation 3



(a)



(b)



(c)

Figure 4.11 – Couples de commandes expérimentaux (a) articulation 1, (b) articulation 2, (c) articulation

4.5 Conclusion

Dans ce chapitre, nous avons proposé deux commandes de type CAI pour les systèmes MIMO incertains soumis à des perturbations externes et effectuant des trajectoires non répétitives. La condition traditionnelle de réinitialisation de l'erreur est remplacée par la condition d'alignement qui est plus pratique. Pour prendre en considération des trajectoires non répétitives, une transformation de temps est introduite, où la conception de la commande et l'analyse de la convergence deviennent plus facile. L'avantage principal des commandes proposées réside dans leurs capacités de bénéficier des itérations précédentes, malgré la variation de la référence à chaque itération en durée, en amplitude ainsi qu'en position initiale et finale. Les simulations et les expérimentations des opérations de prise et de pose s'effectuent sur un robot parallèle de type Delta qui est caractérisé par une dynamique non linéaire et un fort couplage entre ces articulations. Il est démontré que les CAI proposées diminuent les erreurs de poursuite à travers les itérations malgré l'influence de la trajectoire non répétitive et les perturbations externes, tout en gardant le signal de commande dans limites des valeurs physiquement admissibles.

Conclusion Générale

LES travaux de recherche présentés dans cette thèse concernent le développement de lois de commande par apprentissage itératif pour les systèmes non linéaires à trajectoires répétitives et non répétitives. Le domaine d'application le plus indiqué pour ce type de commandes est celui de la robotique. A travers notre recherche bibliographique, nous avons pu déceler un certain nombre de problèmes inhérents à cette approche de commande. Nos contributions vont dans le sens de l'amélioration de ces techniques afin de surmonter les problèmes liés à certaines hypothèses souvent contraignantes. Ainsi, nous avons développé plusieurs approches que nous avons pu mettre en œuvre pratiquement sur un robot réel disponible au Laboratoire de commande des processus.

L'ensemble des contributions peu être résumé comme suit :

- Une loi de commande utilisant le modèle dynamique a été proposée sous la condition classique de réinitialisation de l'erreur. Il a été montré, à travers des résultats simulations, que cette commande est plus performante que les commandes classiques PD plus CAI type PD. La stabilité asymptotique de l'erreur de poursuite est prouvée en utilisant la théorie de Lyapunov.
- La stabilité asymptotique de la commande classique PD plus CAI type PD a été prouvée sous la condition pratique d'alignement. Cette approche a été également validée expérimentalement. Nous avons, ensuite, montré pratiquement que la commande PD plus CAI type PD qui dépend indirectement du modèle et plus performante que la commande par couple a priori qui est entièrement basée sur le modèle dynamique du système.
- Une modification de la commande PD plus CAI type PD a été proposée, en introduisant des gains adaptatifs et un terme de robustification afin d'accélérer la convergence et compenser l'effet des perturbations externes. Une comparaison expérimentale entre la commande proposée et la commande classique PD plus CAI type PD a été effectuée. Les résultats obtenus montrent clairement la supériorité de l'approche proposée.
- Une autre commande utilisant le modèle dynamique du système est proposée pour éviter les hypothèses classiques de la CAI à savoir ; la répétition de la trajectoire et la réinitialisation de l'erreur. Les résultats de simulation obtenus ont montré que la commande prend en charge effectivement des trajectoires non répétitives.
- La dernière commande développée est valable aussi bien pour les trajectoires répétitives que non répétitives et évite l'hypothèse de réinitialisation de l'erreur. En outre, la synthèse de la commande ne nécessite aucune connaissance sur le modèle dynamique. Les résultats expérimentaux sur le robot ISIR88 ont montré la faisabilité et l'efficacité de l'approche proposée.

Pour conclure, l'ensemble d'approches étudiées et développées on permis d'ouvrir

d'un champ d'investigation dans le domaine de la commande par apprentissage itérative. Comme perspective nous proposons l'amélioration de la dernière approche de commande par l'introduction d'un algorithme d'adaptation des paramètres de la loi de commande avec analyse de la stabilité. En outre, nous proposons l'implantation pratique de l'ensemble de commande proposées sur d'autre type de systèmes dynamiques adéquats.

Bibliographie

- [1] M. Uchiyama. "Formation of high-speed motion pattern of a mechanical arm by trial". *Trans. Soc. Instrum. Control Engineering (Japan)*, vol. 14, no. 6, pp. 706-712, 1978.
- [2] S. Arimoto, S. Kawamura, and F. Miyazaki. "Can mechanical robots learn by themselves?". in : H. Hanafusa, H. Inoue (Eds.), *Robotics Research, Second International Symposium*, MIT Press, Cambridge, MA, pp. 127-134, 1984.
- [3] S. Arimoto. "Robustness of learning control for robot manipulators". *IEEE Int. Conf. Robotics and Automation*, Cincinnati, pp. 1528-1533, 1990.
- [4] J. X. Xu and Z. Qu. "Robust iterative learning control for a class of nonlinear systems". *Xu*, vol. 34, no. 8, pp. 983-988, 1998.
- [5] A. Tayebi. "Adaptive iterative learning control for robot manipulators". *Automatica*. 40, pp 1195-1203. 2004.
- [6] X. D. Li, M. M. Lv, and J. K. L. Ho. "Adaptive ILC algorithms of nonlinear continuous systems with non-parametric uncertainties for nonrepetitive trajectory tracking". *Inter Journal of Systems Science*, vol. 47, no. 10, pp 2279-2289, 2016.
- [7] Q. X. Yu, Z. S. Hou, and R. H. Chi. "Adaptive Iterative Learning Control for Nonlinear Uncertain Systems with Both State and Input Constraints". *Journal of the Franklin Institute*, vol. 353, no. 15, pp 2279-2289, 2016.
- [8] J. M. Wei, Y. Zhang, M. Sun, and B. L. G. "Adaptive iterative learning control of a class of nonlinear time-delay systems with unknown backlash-like hysteresis input and control direction". *ISA Trans*, vol. 70, pp 79-92, 2017.
- [9] P.R. Ouyang, W.J. Zhang, and M. Gupta. "An adaptive switching learning control method for trajectory tracking of robot manipulators". *Mechatronics*, 16. pp 51-61. 2006.
- [10] Jian-Xin Xu, R. Yan, and Y. Chen. "On initial conditions in iterative learning control". *Proceedings of the American Control Conference, USA*, June. 2006.
- [11] F. Bouakrif, D. Boukhetala, and F. Boudjema. "Iterative learning control schemes for robot manipulators". *The Mediterranean Journal of Measurement and Control*, 3(3) :104-112 2007.
- [12] F. Bouakrif, D. Boukhetala, and F. Boudjema. "Velocity observer-based iterative learning control for robot manipulators". *International Journal of Systems Science*, vol. 44, no.2, pp. 214-222, 2013.
- [13] D. Meng and K. L. Moore. "Robust Iterative Learning Control for Nonrepetitive Uncertain Systems". *IEEE Trans on automatic control*, vol. 62, no. 2, pp. 907-913, 2017.
- [14] B. Altın, J. Willems, T. Oomen, and K. Barton. "Iterative Learning Control of Iteration-Varying Systems via Robust Update Laws with Experimental Implementation". *Control Engineering Practice*, vol. 62, pp. 36-45, 2017.
- [15] H. Abdellatif, M. Feldt, and B. Heimann. "Application study on iterative learning control of high speed motions for parallel robotic manipulator". *Proceedings of IEEE International Conference on Control Applications*, Germany, October, 2006.
- [16] C. E. Boudjedir, M. Bouri, and D. Boukhetala. "Iterative learning control for trajectory tracking of a parallel Delta robot". *at - Automatisierungstechnik*, vol. 67, no. 2, pp. 145-156, 2019. .
- [17] T. Y Kuc, K. h. Nam, and Jin S . Lee. "An Iterative Learning Control of Robot Manipulators". *IEEE Trans on robotics and automation*, vol. 7, no. 6, 1991.
- [18] L. Zhang, W. Chen, J. Liu, and C. Wen. "A robust adaptive iterative learning control for trajectory tracking of permanent-magnet spherical actuator". *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1 pp 291-301, 2016.
- [19] F. Bouakrif and M. Zasadzinski. "High order iterative learning control to solve the trajectory tracking problem for robot manipulators using Lyapunov theory". *Transactions of the Institute of Measurement and Control*, DOI :10.1177/0142331217741958, 2018.
- [20] X. Li and J.-X. Xu. "On Adaptive Iterative Learning From Tracking Tasks with Different Magnitude and Time Scales". *Chinese Control and Decision Conference (CCDC)*, pp. 4191-4196. doi : 10.1109/CCDC.2016.7531717, 2016.

- [21] S. S. Saab, W. G. Vogt, and M. H. Mickle. "Learning control algorithms for tracking slowly varying trajectories". *IEEE Trans. Syst. Man Cybern. B, Cybern*, vol. 27, no. 4, pp. 657-670, 1997.
- [22] J. X. Xu. "On Iterative Learning From Different Tracking Tasks in the Presence of Time-Varying Uncertainties". *IEEE Trans. Syst. Man Cybern. B, Cybern*, vol. 34, no. 1, pp. 589-597, 2004.
- [23] J.-X. Xu and T. Zhu. "Dual-scale direct learning control of trajectory tracking for a class of nonlinear uncertain systems". *IEEE Trans. Automat. Contr*, vol. 44, no. 10, pp. 1884-1888, 1999.
- [24] X. Jin. "Iterative learning control for non-repetitive trajectory tracking of robot manipulators with joint position constraints and actuator faults". *Intern Journal of Adaptive Control and Signal Processing*, vol. 31, no. 6, pp 859-875, 2017.
- [25] **C. E. Boudjedir**, D. Boukhetala, and M. Bouri. "Iterative learning control of multivariable uncertain nonlinear systems with nonrepetitive trajectory". *Nonlinear Dynamics*, vol. 95, no. 3, pp. 2197-2208, 2019. <https://doi.org/10.1007/s11071-018-4685-0>.
- [26] J.-X. Xu and Y. Tan. "Linear and Nonlinear Iterative Learning Control". *Lecture Notes in Control and Information Science*, Vol. 291, Hiedelberg, Germany : Springer-Verlag. 2003.
- [27] H.S. Ahn, Y. Chen, and K.L. Moore. "Iterative learning control : brief survey and categorization". *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev*, vol. 37, no. 6, pp. 1099-1121, Nov. 2007.
- [28] Jian-Xin Xu. "A survey on iterative learning control for nonlinear systems". *International Journal of Control*, Vol. 84, no. 7, pp 1275-1294, 2011.
- [29] Jian-Xin Xu, R. Yan, and Y. Chen. "On initial conditions in iterative learning control". *Proceedings of the American Control Conference, USA*, June. 2006.
- [30] Z. Yang and C. W. Chan. "Conditional iterative learning control for non-linear systems with non-parametric uncertainties under alignment condition". *IET Control Theory & Applications*, vol. 3, no. 11, pp. 1521-1527, 2009.
- [31] M. Uchiyama. "Formation of high-speed motion pattern of a mechanical arm by trialt". *Trans. Soc. Instrum. Control Engineering (Japan)*, vol. 14, no. 6, pp. 706-712, 1978.
- [32] Y. S. Lu and X. W. Wang. "Sliding-mode repetitive learning control with integral sliding-mode perturbation compensation ". *ISA Trans*, vol. 48, no. 02, pp. 156-165, 2009.
- [33] **C. E. Boudjedir**, D. Boukhetala, and M. Bouri. "Fuzzy logic iterative learning control for trajectory tracking of parallel kinematic manipulators". 5th International conference on electrical engineering, algeria, pp. 1-6. doi : 10.1109/ICEE-B.2017.8192041, 2017.
- [34] **C. E. Boudjedir**, D. Boukhetala, and M. Bouri. "Iterative Learning Control of a Parallel Delta Robot". In : M. Chadli, S. Bououden, S. Ziani, I. Zelinka, (eds) *Advanced Control Engineering Methods in Electrical Engineering Systems. ICEECA 2017. Lecture Notes in Electrical Engineering*, vol 522. Springer, DOI : doi.org/10.1007/978-3-319-97816-1-6.
- [35] J. X. Xu and Z. Qu. "Robust iterative learning control for a class of nonlinear systems". *Xu*, vol. 34, no. 8, pp. 983-988, 1998.
- [36] L. A. Castañeda, A. L. Juárez, and I. Chairez. "Robust trajectory tracking of a delta robot through adaptive active disturbance rejection control". *IEEE Transactions on control systems technology*, vol. 23, no. 4, pp 1387-1398, 2015.
- [37] M. F. Heertjes, B. V. der Velden, and T. Oomen. "Constrained Iterative Feedback Tuning for Robust Control of a Wafer Stage System". *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 1, pp. 56-66, 2016.
- [38] P. Chirarattananon, K. Y. Ma, and R. J Wood. "Perching with a robotic insect using adaptive tracking control and iterative learning control". *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1185-1206, 2016.
- [39] X. Li, Q. Ren, and Jian-Xin Xu. "Precise speed tracking control of a robotic fish via iterative learning control". *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4 pp 2221-2228, 2015.
- [40] D. W. Wang. "A simple iterative learning controller for manioulators with flexible joints". *At*, vol. 31, no. 5, pp. 1198-1204, 1995.
- [41] K. S. Lee, S. H. Bang, S. Yi, J. S. Son, and S. C. Yoon. "Iterative learning control of heat up phase for a batch polymerization reactor". *Journal of process control*, 6(4), pp 255-262 , 1996.

- [42] J. Y. Choi and H. M. Do. "A learning approach of wafer temperature control in rapid thermal processing system". IEEE transaction on semiconductor manufacturing vol. 14, no. 1, pp. 1-10, 2001.
- [43] K. L. Moore and A. Mathews. "Iterative learning control with non-standard assumptions applied to the control of gas-metal arc welding". In : iterative learning control- Analysis, design, integration and applications, pp. 335-350, Kluwer academic press. Boston, USA, 1998.
- [44] K. K. Tan, T. H. Lee, and H. X. Zhou. "Micro-positioning of linear piezoelectronic motors based on learning nonlinear pid controller". IEEE/ASME Transactions on Mechatronics vol. 6, no. 4, pp. 428-436, 2002.
- [45] J. Y. Choi and H. M. Do. "On the realization of an accurate hydraulic servo through an iterative learning control". Mechatronics vol. 2, no. 1, pp. 75-88, 1992.
- [46] C. H. Tsai and C. J. Chen. "Application of iterative path revision technique for laser cutting with controlled fracture". Opt. Laser Eng., vol. 41, no. 1, pp. 189-204, , 2004.
- [47] J. X. Xu and D. Huang. "Optimal Tuning of PID Parameters Using Iterative Learning Approach". IEEE 22nd International Symposium on Intelligent Control, Singapore, pp. 226-231. 2007.
- [48] J-P. Merlet and D. Daney. "Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace.". IEEE, 2005.
- [49] W.L.V.Pollard. "Position controlling apparatus". US Patent No. 2,286,571, June 16, 1942.
- [50] V.E.Gough. "Contribution to discussion of papers on research in automobile stability, control and tyre performance". Proc. Auto Div. Inst.Mech. Eng, pp. 392-394, 1956.
- [51] R. Clavel. "Device for movement and displacing of an element in space". US Patent n : 4976582, date of patent : December 11, 1990.
- [52] **C. E. Boudjedir**, D. Boukhetala, and M. Bouri. "Nonlinear PD control of a Parallel Delta robot : Experimentals Results". the International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM), Algiers, 2018, DOI : 10.1109/CISTEM.2018.8613618.
- [53] **C. E. Boudjedir**, D. Boukhetala, and M. Bouri. "Nonlinear PD plus sliding mode control with application to a parallel delta robot". Journal of Electrical Engineering, vol. 69, no. 5, pp. 329-336, 2018. .
- [54] R. Clavel. "Conception d'un robot parallèle rapide à 4 degré de liberté". [Doctoral Thesis]. Swiss Federal Institute of Technology Lausanne (EPFL), 1991.
- [55] A. Codourey. "Contribution à la commande des robots rapides et précis - Application au robot Delta à entraînement direct.". Thèse de doctorat, Ecole Polytechnique Federale de Lausanne - EPFL, 1991.
- [56] A. Codourey. "Dynamic modelling and mass matrix evaluation of the delta parallel robot for axes decoupling control". the International Conference on Intelligent Robots and Systems, vol. 3, pp. 1211-1218, 1996.
- [57] R. Kelly, V. Santibáñez, and A. Loría. "Control of Robot Manipulators in Joint Space". Advanced textbooks in control and signal processing. Springer, Berlin, 2005.
- [58] F. Bouakrif and M. Zasadzinski. "Trajectory tracking control for perturbed robot manipulators using iterative learning method". Int J Adv Manuf Technol, doi :10.1007/ s00170-016-8550-3, 2016.
- [59] X. Jin, D. Huang, and Jian-Xin Xu. "Iterative Learning Control for systems with nonparametric uncertainties under alignment condition". 51st IEEE Conference on Decision and Control, 10-13, USA. Dec 2012.
- [60] C.H. An, C.G. Atkeson, J.D. Griffiths, and J.M. Hollerbach. "Experimental evaluation of feedforward and computed torque control". IEEE Trans. on Robotics and Automation, vol. 5, no. 3, 368-373. 1989.
- [61] V. Santibáñez and R. Kelly. "PD control with feedforward compensation for robot manipulators : analysis and experimentation". Robotica, vol. 19, pp. 11-19, 2001.
- [62] F. Caccavale and P. Chiacchio. Identification of dynamic parameters and feedforward control for a conventional industrial manipulator". Control Engineering Practice, vol. 2, no. 6, pp. 1039-1050, 1994.
- [63] W. W.Shang, S. Cong, Z. X. Li, and S. L. Jiang. "Augmented Nonlinear PD Controller for a Redundantly Actuated Parallel Manipulator". Advanced Robotics, vol. 23, no. 12-13, pp 1725-1742, 2009.
- [64] M. Yu, W. Zhou, and B. Liu. "On iterative learning control for MIMO nonlinear systems in the presence of time-iteration-varying parameters". Nonlinear Dynamics, vol. 89, no.04, pp. 2561-2571, 2017.

- [65] G. S. Natal, A. Chemori, and F. Pierrot. *"Dual-Space Control of Extremely Fast Parallel Manipulators : Payload Changes and the 100G Experiment"*. IEEE Trans. Control Syst. Technol, vol 23, no. 4, pp 1520-1535, 2015.
- [66] X. Jin. *"Non-Repetitive Trajectory Tracking for Joint Position Constrained Robot Manipulators Using Iterative Learning Control"*. IEEE 55th Conference on Decision and Control (CDC), USA, 2016.
- [67] T. Sugie, Y. Kito, and K. hujimoto. *"Obstacle Avoidance of Manipulators with Rate Constraints"*. IEEE Transactions on Robotics and Automation, vol. 19, no. 1, pp. 168-174, 2003.
- [68] M. Sampei and K. Furuta. *"On time scaling for nonlinear systems : Application to linearization"*. IEEE Transactions on Automatic Control, vol. 31, no. 5, pp. 459-462, 1986.
- [69] J. Huang, S. Ri, L. Liu, Y. Wang, J. Kim, and G. Pak. *"Nonlinear Disturbance Observer-Based Dynamic Surface Control of Mobile Wheeled Inverted Pendulum"*. IEEE Transactions on Control Systems Technology, vol. 23, no. 6, pp. 2400-2407, Nov. 2015.