



Mémoire de Magister

Spécialité : **Automatique**

Option : **Systemes Intelligents de Commande et Robotique**

Présenté par

OULMAS Youssef

Ingénieur en Automatique de l'UMMTO

Thème

Commande Adaptative Floue de la Position et de l'Orientation des Robots Manipulateurs

Soutenu publiquement le 27 juin 2006 devant le jury composé des membres :

Président :	M. TADJINE	Maître de Conférences, ENP
Rapporteur :	H. CHEKIREB	Maître de Conférences, ENP
Examineurs :	L. NEZLI	Docteur, ENP
	R. ILLOUL	Chargé de Cours, ENP
	O. STIHI	Chargé de Cours, ENP

الملخص

في الصناعة الحديثة، الأذرع الآلية تحض بمكانة هامة. هذه الأذرع كثيرا ما تستعمل لقيام ببعض المهام كالتلحيم، الطلاء... الخ، التي تتطلب الحركة عبر الزمن و الفضاء للأداة. هذه المهام صعبة الانجاز خاصة بالنسبة لاذرع ذات أكثر من ثلاث درجات الحرية، و هذا لكون ديناميكية هذه الأذرع معقدة و جد متشابكة. يهدف هذا العمل إلى التحكم التلاؤمي الغامض لموضع و توجيه أذرع آلية. أولا محدودية تقنيات التلاؤمية الغير الغامضة المطبقة على الأذرع الآلية قد ذكرت. ثم نضرة وجيزة عن الجمل الغامضة قد عرضت. هذه الأخيرة تستعمل من جهة لتمثيل الديناميكية المجهولة للأذرع الآلية و هذا للحصول على قانون التحكم التلاؤمي الغامض المركزي و الغير المركزي الغير المباشر، ومن جهة أخرى تستعمل مباشرة لتمثيل قانون التحكم التلاؤمي الغامض المركزي أو الغير المركزي المباشر. هذه القوانين طبقت عن طريق المحكات على الذراع الآلي PUMA 560 في مهمة التلحيم.

كلمات مفتاحيه : التحكم في الأذرع الآلية، الجمل الغامضة، التحكم الغامض، التحكم التلاؤمي، التحكم الموضعي و التوجيهي للأذرع الآلية.

RESUME

Dans l'industrie moderne, les robots manipulateurs occupent une place primordiale. Ces derniers sont souvent appelés à effectuer certaines tâches, telle que la soudure, la peinture...etc., qui nécessitent l'évolution dans le temps et dans l'espace de la position/orientation de l'effecteur. Ces tâches sont difficiles à réaliser surtout pour les robots ayant plus de trois degrés de liberté. En effet, la dynamique des robots est complexe et fortement couplée de plus, le plus souvent elle est entachée d'incertitudes paramétriques. L'objectif de ce travail porte sur la commande adaptative floue de la position/orientation des robots manipulateurs. Tout d'abord ont été évoquées les limites des techniques non floues de commande adaptative classique qui sont appliquées à la commande des robots. Puis, a été exposé un bref aperçu sur les systèmes flous. Ces derniers sont utilisés, en premier lieu pour estimer la dynamique inconnue du robot manipulateur afin de développer la commande adaptative floue centralisée et décentralisée indirecte, et en second lieu pour approximer directement les lois de commande adaptative floue centralisée et décentralisée. Ces lois de commande ont été appliquées par simulation au cas du robot PUMA 560 exécutant une tâche de soudage.

Mots clé : Commande des robots, systèmes flous, commande floue, commande adaptative, commande en position/orientation des robots.

ABSTRACT

In modern industry, the robot manipulator takes a paramount place. This last often has to carry out certain tasks, such as the welding, painting... etc, which require the evolution in the time and the space of the position/orientation of its effector. These tasks are very difficult to realize especially for robots having more than three degree of freedom. Indeed, the dynamic of robots is complex and hardly coupled moreover it is involved with parametric uncertainties. So, the aim of this work concerns with the fuzzy adaptive control of the position/orientation of the robots manipulators. First of all, the limits of not fuzzy techniques of classical adaptive control, which are applied to robots control, were evoked. Then, a short outline on the fuzzy systems was given. These latter, are used in first to estimate the unknown dynamics of the robot manipulator in order to develop the centralized and decentralized indirect fuzzy adaptive control, and in second step to approximate directly the centralized and decentralized fuzzy adaptive control laws. These control laws are applied by simulation in the case of the robot PUMA 560 executing a welding task.

Key words: Robots control, fuzzy systems, fuzzy control, adaptive control, robots position and orientation control.

Remerciements

Le travail présenté dans ce mémoire a été effectué au Laboratoire de Commande des Processus de Génie Electrique de l'Ecole Nationale Polytechnique.

Je tiens à remercier Monsieur H. CHEKIREB, Maître de Conférences à l'ENP, pour la proposition du sujet de ce mémoire et pour ses conseils tout au long de ce travail.

Je remercie les membres de jury, qui m'ont fait l'honneur de participer au jugement de ce travail.

Pour leur encouragement, leur soutien moral et la patience qu'ils m'ont manifestés durant mes études, je remercie tous les membres de ma famille.

Enfin, Je remercie tous mes amis (es).

SOMMAIRE

SOMMAIRE

INTRODUCTION GENERALE	1
CHAPITRE 1 MOTIVATION ET POSITION DU PROBLEME	3
1.1 INTRODUCTION	3
1.2 TYPE DE TACHE.....	3
1.2.1 DESCRIPTION MATHEMATIQUE DE LA TACHE DANS L'ESPACE CARTESIEN	4
1.3 LE ROBOT	7
1.4 TRANSFORMATIONS NECESSAIRES A LA COMMANDE DES ROBOTS.....	9
1.4.1 MODELE GEOMETRIQUE DIRECT ET INVERSE DU ROBOT PUMA 560..	9
1.4.2 MODELE CINEMATIQUE DIRECT ET INVERSE DU ROBOT PUMA 560....	9
1.4.3 MODELE CINEMATIQUE INVERSE DU DEUXIEME ORDRE (MCI2).....	10
1.5 LIMITES DES TECHNIQUES DE COMMANDE CLASSIQUES DES ROBOTS ...	11
1.5.1 COMMANDE PID	11
1.5.2 COMMANDES NECESSITANT LA CONNAISSANCE DU MODELE DYNAMIQUE.....	12
1.5.3 COMMANDE ROBUSTE	15
1.5.4 COMMANDE ADAPTATIVE	18
1.6 CONCLUSION.....	21
CHAPITRE 2 SYSTEMES FLOUS	22
2.1 INTRODUCTION	22
2.2 SYSTEME FLOU.....	22
2.2.1 BASE DES REGLES.....	23
2.2.2 FUZZIFICATION	26
2.2.3 INFERENCE	27
2.2.4 DEFUZZIFICATION	28
2.3 SYSTEME FLOU DE TAKAGI-SUGENO.....	28
2.4 ADAPTATION DES PARAMETRES DU SYSTEMES FLOU DE SUGENO	31
2.5 CONCLUSION.....	31
CHAPITRE 3 COMMANDE ADAPTATIVE FLOUES CENTRALISEE	32
3.1 INTRODUCTION	32
3.2 COMMANDE ADAPTATIVE FLOUE INDIRECTE	33
3.2.1 APPROXIMATION PAR LES SYSTEMES FLOUS	33
3.2.2 SYNTHESE DE LA LOI DE COMMANDE.....	35
3.2.3 ANALYSE DE LA STABILITE.....	37
3.2.4 RESULTATS DE SIMULATION.....	39
▪ Robustesse vis-à-vis des perturbations extérieures.....	44
▪ Robustesse vis-à-vis des variations paramétriques	46
3.3 COMMANDE ADAPTATIVE FLOUE DIRECTE.....	51
3.3.1 APPROXIMATION PAR LES SYSTEMES FLOUS	51

3.3.2 SYNTHÈSE DE LA LOI DE COMMANDE.....	52
3.3.3 ANALYSE DE LA STABILITÉ.....	55
3.3.4 RESULTATS DE SIMULATION.....	56
▪ Robustesse vis-à-vis des perturbations extérieures.....	61
▪ Robustesse vis-à-vis des variations des paramètres du robot.....	64
3.4 CONCLUSION.....	68
CHAPITRE 4 COMMANDE ADAPTATIVE FLOUE DECENTRALISEE	69
4.1 INTRODUCTION.....	69
4.2 COMMANDE ADAPTATIVE DECENTRALISEE FLOUE INDIRECTE.....	70
4.2.1 ESTIMATION PAR LES SYSTEMES FLOUS.....	72
4.2.2 SYNTHÈSE DE LA LOI DE COMMANDE.....	72
4.2.3 ANALYSE DE LA STABILITÉ.....	73
4.2.4 RESULTATS DE SIMULATION.....	76
▪ Robustesse vis-à-vis des perturbations extérieures.....	81
▪ Robustesse de la loi de commande vis-à-vis des variations des paramètres.....	84
4.3 COMMANDE ADAPTATIVE DECENTRALISEE FLOUE DIRECTE.....	88
4.3.1 SYNTHÈSE DE LA LOI DE COMMANDE.....	90
4.3.2 ANALYSE DE LA STABILITÉ.....	91
4.3.3 RESULTATS DE SIMULATION.....	93
▪ Robustesse vis-à-vis des perturbations extérieures.....	97
▪ Robustesse vis-à-vis des variations paramétriques.....	99
4.4 CONCLUSION.....	102
REFERENCES BIBLIOGRAPHIQUES.....	106
ANNEXE 1.....	110
PARAMETRES DU MODELE GEOMETRIQUE ET DYNAMIQUE DU PUMA 560..	110
TERMES DE FROTTEMENTS VISQUEUX.....	117
ANNEXE 2.....	119
MODELE GEOMETRIQUE DIRECT ET INVERSE DU PUMA 560.....	119
MODÈLE CINÉMATIQUE DIRECT ET INVERSE DU PUMA 560.....	122

***INTRODUCTION
GENERALE***

INTRODUCTION GENERALE

La commande des robots manipulateurs est l'une des préoccupations majeures des recherches en robotique. En effet, un robot manipulateur est caractérisé par un comportement purement non linéaire, de plus, Certaines tâches qui lui sont confiées sont délicates et exigent une très grande précision sous des trajectoires rapides.

Dans le cas où le modèle exact du robot est parfaitement connu, plusieurs stratégies de commande peuvent être appliquées. Cependant, en pratique, cette condition idéale n'est jamais tout à fait remplie vu les différentes perturbations agissant sur le robot manipulateur, et les incertitudes du modèle, d'où la nécessité d'adapter la commande.

Durant ces trois dernières décennies, en vue d'améliorer les performances des robots manipulateurs, des recherches avancées ont permis de développer de nouvelles techniques de commande non linéaire telle que la commande adaptative pour les applications aux robots manipulateurs. L'implantation des fonctions d'auto-adaptation dans les régulateurs est rendue possible grâce à l'essor de la micro-informatique et le recours aux techniques d'identification.

Lorsqu'on cherche à identifier des modèles non linéaires, certaines informations sur le système sont difficiles à incorporer dans le modèle paramétrique standard. Des travaux de recherche sur les systèmes flous montrent que ces informations sont très faciles à approximer par les systèmes flous [Taka-85] [Moud-95] [Spoo-96]. La problématique de la commande adaptative floue est basée sur la propriété d'approximation universelle des systèmes flous et sur le théorème de stabilité de Lyapunov. En effet, ceux-ci sont capables d'approximer, avec un degré de précision arbitraire fixé, n'importe quelle dynamique non linéaire sur un ensemble compact [Wang-92] [Buck-92].

D'une manière générale, les contrôleurs adaptatifs flous sont divisés en deux grandes classes : une s'appelle la commande adaptative floue directe et l'autre la commande adaptative floue indirecte. Dans la commande adaptative floue directe, le système flou est vu comme un contrôleur adaptatif. Cependant, dans la commande adaptative floue indirecte, les systèmes flous sont utilisés pour modéliser le système réel et le contrôleur est construit en supposant que les

systèmes flous représentent approximativement le vrai système, ces deux tâches se font séquentiellement.

Le mémoire est organisé en quatre chapitres qui peuvent être résumés comme suit :

Le premier chapitre est dédié à la présentation de la tâche de soudage sur une trajectoire spirale, à confier au robot ainsi que la présentation du robot lui-même puis on a justifié le recours à la commande adaptative floue.

Le deuxième chapitre est dédié aux systèmes flous. Aussi, on a donné la structure d'un type du système flou qui est directement lié aux techniques de commande appliquées au robot à savoir les systèmes flous de Takagi-Sugeno.

Dans le troisième chapitre, nous appliquons deux techniques de commande adaptative floue centralisées pour la commande de la position et de l'orientation des robots manipulateurs, à savoir la commande directe et la commande indirecte. Si théoriquement il est toujours possible de représenter la dynamique d'un système non linéaire par un système flou, en pratique cette représentation peut nécessiter l'utilisation d'un très grand nombre de règles floues. Dans de pareils cas l'utilisation de la commande adaptative décentralisée floue s'impose, celle-ci fait l'objet du quatrième chapitre. Ce dernier chapitre est divisé, quant à lui, en deux parties : l'approche décentralisée indirecte est présentée en premier lieu, suivi de l'approche directe.

Les applications de ses commandes au robot sont effectuées par simulation. Pour chaque approche on a effectué des simulations avec le robot sans charge, puis on a testé la robustesse des lois de commande vis-à-vis des perturbations extérieures (charge) et vis-à-vis des perturbations paramétriques.

CHAPITRE 1 :

MOTIVATION ET POSITION DU PROBLEME

MOTIVATION ET POSITION DU PROBLEME

1.1 INTRODUCTION

Les robots sont en train de révolutionner l'industrie moderne. Ils s'avèrent particulièrement précieux dans de nombreuses applications industrielles [Khal-99], en particulier la manutention, la peinture, la soudure...etc. Ces opérations tendent à être robotisées, soit pour décharger l'homme des tâches fastidieuses, soit pour accroître de manière importante la productivité et la flexibilité des ateliers de production ou parce que les tâches demandent une précision ou une dextérité impossible d'atteindre humainement.

Pour réaliser ces tâches, le robot agit dans l'environnement grâce à son organe terminal. Cette action d'une manière générale est caractérisée par l'évolution dans le temps et dans l'espace de la position et de l'orientation de cet organe, appelé aussi effecteur.

1.2 TYPE DE TACHE

Dans ce mémoire, nous nous intéressons à l'une des tâches précitées qui est la soudure. Cette tâche consiste à ce que le robot effectue la soudure de forme spirale sur un cylindre d'origine (0.5, 0.5,-0.1) m par rapport au repère de la base du robot, de 0.30m de diamètre et de 0.15m de hauteur dans le sens contraire aux aiguilles d'une montre, tout en maintenant l'outil orienté radialement vers l'axe central du cylindre. Ce dernier est choisi parallèle à l'axe "z" du repère de la base (0, x, y, z), comme le montre la figure 1.1. La position et les dimensions du cylindre sont prises d'une manière à faire éviter au robot les positions singulières tout en maintenant les positions articulaires loin des butées mécaniques.

Le problèmes de contrôle des robots manipulateurs est généralement décomposé en trois étapes : la planification des trajectoires, la génération des trajectoires et la poursuite des trajectoires.

1.2.1 DESCRIPTION MATHÉMATIQUE DE LA TACHE DANS L'ESPACE CARTESIEN

La génération des mouvements est la fonctionnalité du contrôleur qui effectue les calculs nécessaires pour obtenir les consignes à poursuivre par le robot à partir des instructions données. Ces consignes prennent généralement la forme de trajectoires temporelles de position dans l'espace des articulations du robot ou dans l'espace de la tâche.

Dans le cas de la génération des mouvements dans l'espace articulaire, les consignes sont calculées et synchronisées sur la trajectoire la plus lente afin que tous les axes passent en même temps sur les points imposés. Un inconvénient majeur de ce type de mouvement est la non prévisibilité de la trajectoire de l'effecteur dans l'espace physique de la tâche résultant du calcul de trajectoire fait au niveau articulaire. Cependant, l'absence d'appel au modèle géométrique inverse du robot pour la commande favorise la rapidité d'exécution de la tâche [Kha-99].

La génération de la trajectoire dans l'espace de la tâche implique une trajectoire en position cartésienne avec une orientation de l'effecteur par rapport à un repère de base du robot. A la différence de l'approche articulaire, dans laquelle les contraintes de vitesse et d'accélération sont associées à chaque axe, la génération dans l'espace de la tâche est généralement fondée sur des données cinématiques de vitesse et d'accélération cartésiennes, en normes euclidiennes. La trajectoire d'orientation de l'effecteur est généralement traitée en termes d'angles d'Euler et synchronisée sur la trajectoire cartésienne.

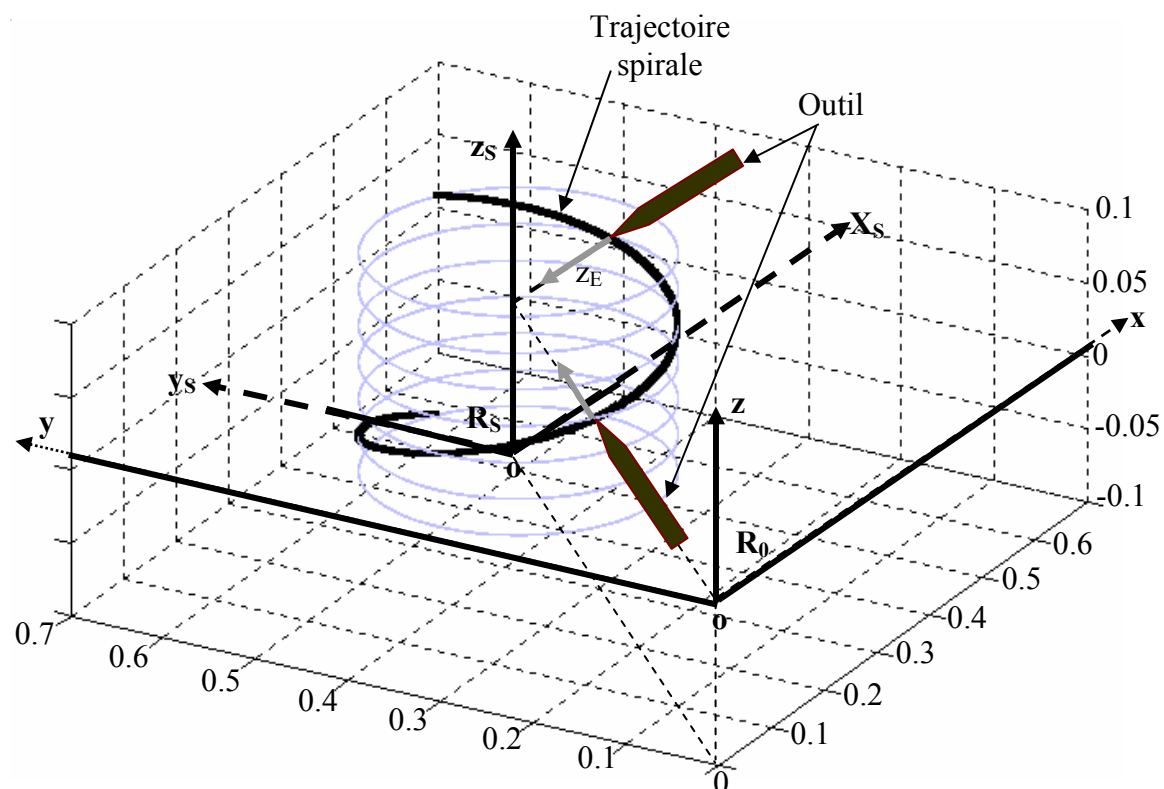


Fig.1.1 La trajectoire spirale à parcourir dans l'espace cartésien

On note que R_0 est le repère de la base du robot et R_S est le repère de la station.

Deux paramètres suffisent pour décrire l'évolution de la position et de l'orientation de l'outil lors de l'exécution de cette tâche : l'angle φ pour décrire la projection de la position de l'outil par rapport au plan (o_s, x_s, y_s) du repère de la station et sa position verticale h selon l'axe z (Figure 1.2 et Figure 1.3)

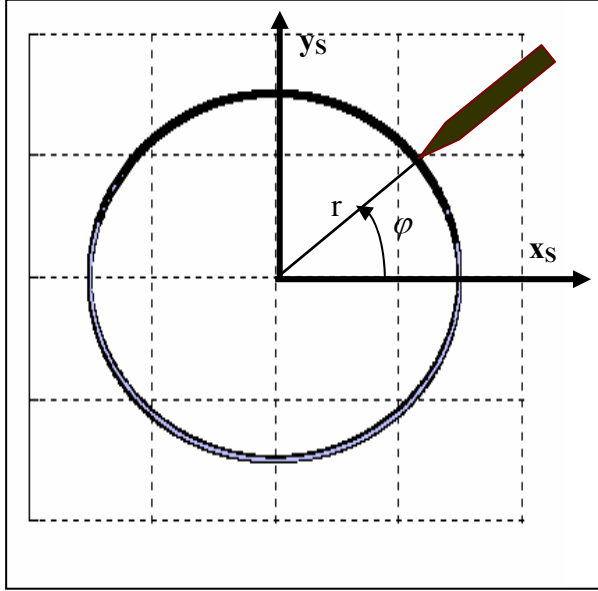


Fig.1.2 Projection de la trajectoire sur le plan (o, x, y) du R_S

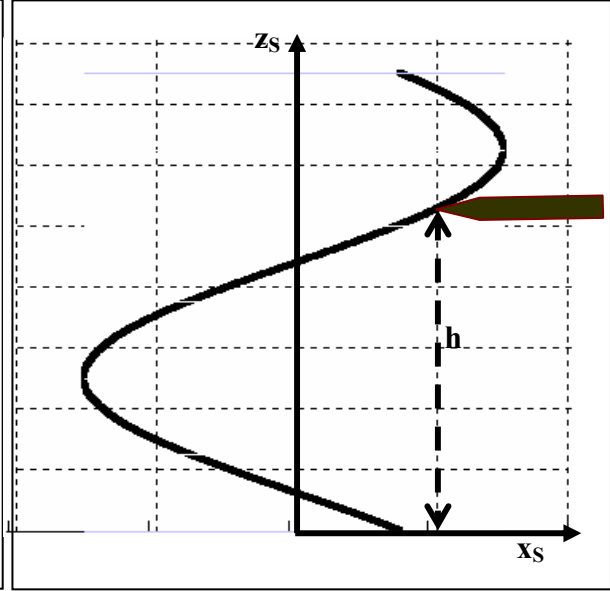


Fig.1.3 Projection de la trajectoire sur le plan (o, x, z) du R_S

Ainsi, les équations décrivant la position cartésienne de l'effecteur par rapport au repère de la base sont :

$$\begin{aligned} x &= x_0 + r \cos(\varphi) \\ y &= y_0 + r \sin(\varphi) \\ z &= z_0 + \frac{h}{2\pi}(\varphi - \varphi_0) \end{aligned} \quad (1.1)$$

où (x_0, y_0, z_0) est la position de l'origine du repère de la station R_S par rapport à celui de la base R_0

En adoptant la paramétrisation d'Euler zyz par les angles (α, β, γ) [Khal-99] [Yosh-90], l'orientation de l'effecteur par rapport au repère de la base est décrite par :

$$\alpha = \varphi + \pi ; \quad \beta = \frac{\pi}{2} ; \quad \gamma = 0 \quad (1.2)$$

$$x^d = [x \ y \ z \ \alpha \ \beta \ \gamma]^T$$

Les vitesses cartésiennes se déduisent de la position et de l'orientation cartésienne par simple dérivation par rapport au temps

$$\begin{aligned}\dot{x} &= \frac{dx}{dt} = -r \sin(\varphi) \dot{\varphi} \\ \dot{y} &= \frac{dy}{dt} = r \cos(\varphi) \dot{\varphi}\end{aligned}\tag{1.3}$$

$$\dot{z} = \frac{dz}{dt} = \frac{h}{2\pi} \dot{\varphi}$$

$$\dot{\alpha} = \frac{d\alpha}{dt} = \dot{\varphi}$$

$$\dot{\beta} = \frac{d\beta}{dt} = 0\tag{1.4}$$

$$\dot{\gamma} = \frac{d\gamma}{dt} = 0$$

$$\dot{x}^d = [\dot{x} \ \dot{y} \ \dot{z} \ \dot{\alpha} \ \dot{\beta} \ \dot{\gamma}]^T$$

Les accélérations cartésiennes sont :

$$\ddot{x} = \frac{d^2x}{dt^2} = -r(\cos(\varphi) \dot{\varphi}^2 + \sin(\varphi) \ddot{\varphi})$$

$$\ddot{y} = \frac{d^2y}{dt^2} = r(-\sin(\varphi) \dot{\varphi}^2 + \cos(\varphi) \ddot{\varphi})\tag{1.5}$$

$$\ddot{z} = \frac{d^2z}{dt^2} = \frac{h}{2\pi} \ddot{\varphi}$$

$$\ddot{\alpha} = \frac{d^2\alpha}{dt^2} = \ddot{\varphi}$$

$$\ddot{\beta} = \frac{d^2\beta}{dt^2} = 0\tag{1.6}$$

$$\ddot{\gamma} = \frac{d^2\gamma}{dt^2} = 0$$

$$\ddot{x} = [\ddot{x} \ \ddot{y} \ \ddot{z} \ \ddot{\alpha} \ \ddot{\beta} \ \ddot{\gamma}]^T$$

avec : $h \in [-10 \ 5] \text{ cm}$ et $\varphi \in [0 \ 2\pi]$, évolue selon une fonction polynomiale cubique tel que :

$$\varphi = \varphi_0 + 6 \frac{\pi}{t_f^2} t^2 - 4 \frac{\pi}{t_f^3} t^3$$

$$\dot{\varphi} = \frac{d\varphi}{dt} = 12 \frac{\pi}{t_f^2} t - 12 \frac{\pi}{t_f^3} t^2 \quad (1.7)$$

$$\ddot{\varphi} = \frac{d^2\varphi}{dt^2} = 12 \frac{\pi}{t_f^2} - 24 \frac{\pi}{t_f^3} t$$

En adoptant la paramétrisation d'Euler zyz par les angles (α, β, γ) , l'évolution de l'orientation désirée de l'outil est représentée sur la figure 1.3

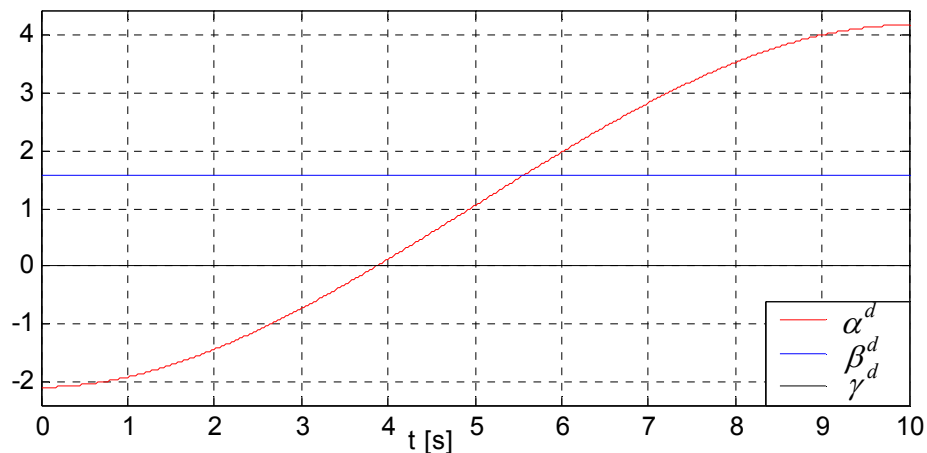


Fig.1.3 Orientation désirée de l'outil (unité : rd)

1.3 LE ROBOT

Un robot manipulateur est généralement considéré comme un ensemble de corps rigides connectés en série par des articulations, avec une extrémité au sol, et l'autre libre (effecteur ou élément terminal).

Vu sa structure mécanique (figure 1.4), le robot PUMA 560 se prête bien pour ce genre de tâches. Celui-ci est caractérisé par six liaisons de type rotoïdes (6R) lui conférant six degrés de liberté, trois premières liaisons pour le positionnement dans l'espace et trois autres (wrist) pour son orientation [Seli-92] et [Arms-86].

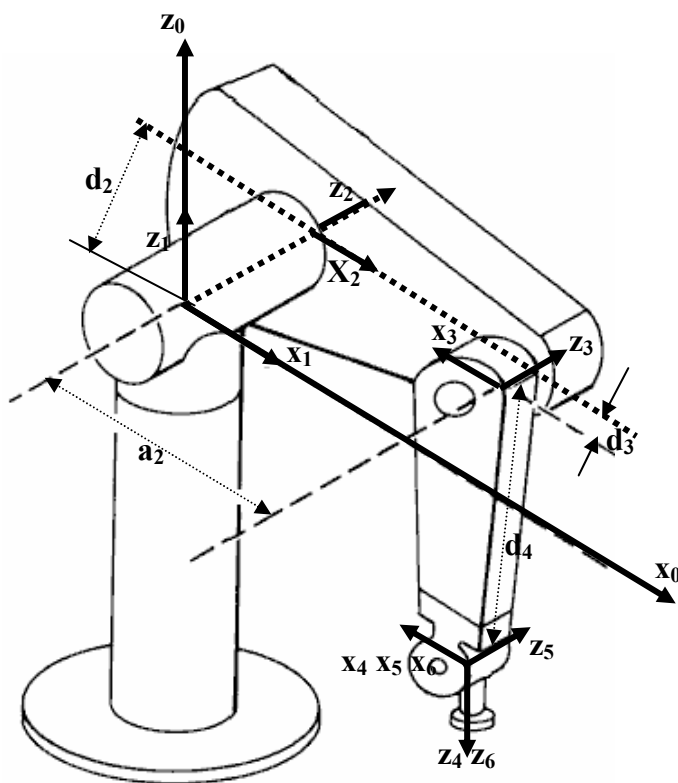


Fig.1.4 Architecture du Puma 560 avec les systèmes de coordonnées attachés

Les paramètres de Denavit-Hartenberg apparaissant à la figure 1.4 sont présentés au tableau 1.1

i	α_{i-1} (degré)	q_i	a_{i-1} (mètres)	d_i (mètres)
1	0	q_1	0	0
2	-90	q_2	0	0.2435
3	0	q_3	0.4318	-0.0934
4	90	q_4	-0.0203	0.4331
5	-90	q_5	0	0
6	90	q_6	0	0

Tab.1.1 Paramètres de Denavit-Hartenberg

Le couple maximal que peut développer chaque moteur est donné au tableau 1.2

	Art 1	Art 2	Art 3	Art 4	Art 5	Art 6
Couple max [N.m]	97.6	186.4	89.4	24.2	20.1	21.3

Tab.1.2 couples max développés par chaque moteur

1.4 TRANSFORMATIONS NECESSAIRES A LA COMMANDE DES ROBOTS

Les tâches à confier aux robots sont décrites tantôt dans l'espace cartésien tantôt dans l'espace articulaire, la correction peut se faire aussi bien dans l'espace articulaire que cartésien. Ainsi, le recours aux transformations ci-dessous est incontournable.

1.4.1 MODELE GEOMETRIQUE DIRECT ET INVERSE DU ROBOT PUMA 560

Le modèle géométrique direct du robot PUMA 560 permet d'exprimer les coordonnées opérationnelles (position et orientation) de l'effecteur en fonction des coordonnées articulaires du robot : $X=f(q)$ [annexe 2]. Ce dernier est calculé en utilisant les notations de Denavit et Hartenberg.

Le modèle géométrique inverse, quant à lui, exprime les coordonnées articulaires en fonction des coordonnées opérationnelles $q=g(X)$ [annexe 2].

En utilisant le MGI [annexe 2], l'image des trajectoires cartésiennes désirées dans l'espace articulaire sont des trajectoires lisses et continues (figure 1.5).

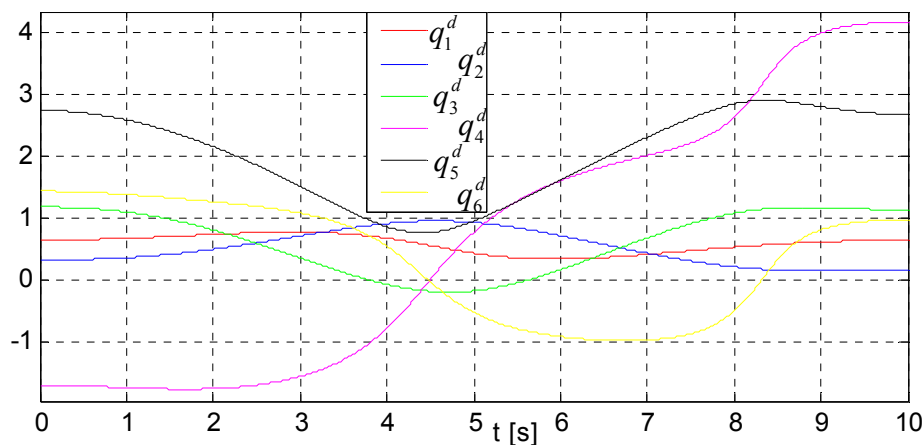


Fig.1.5 Image de la trajectoire désirée dans l'espace articulaire (unité : rd)

1.4.2 MODELE CINEMATIQUE DIRECT ET INVERSE DU ROBOT PUMA 560

Le modèle cinématique direct décrit les vitesses opérationnelles en fonction des vitesses articulaires et il est défini par :

$$\dot{X}=J(q)\dot{q} \quad (1.8)$$

où $J(q)$ désigne la matrice jacobienne du robot. Le problème revient au calcul du jacobien.

Dans le cas des robots à moins de trois degrés de libertés, ce dernier peut être obtenu en dérivant directement le MGD. Mais, pour les robots à plus de trois degrés de liberté tel que le PUMA 560, les méthodes récursives s'imposent [Seli-92] [Yosh-90] [Khal-89] [Craig-86]

La méthode récursive [annexe 2] consiste à calculer le jacobien de base J_V , qui exprime les vitesses réelles de l'effecteur (vitesses de translation et vitesses de rotation), sans passer par les dérivées partielles.

Le modèle cinématique inverse (MCI) décrit les vitesses articulaires en fonction des vitesses opérationnelles [annexe 2]

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

En utilisant le MCI, les vitesses articulaires désirées sont données à la figure 1.6. Celles-ci sont en dehors des butées articulaires et des singularités).

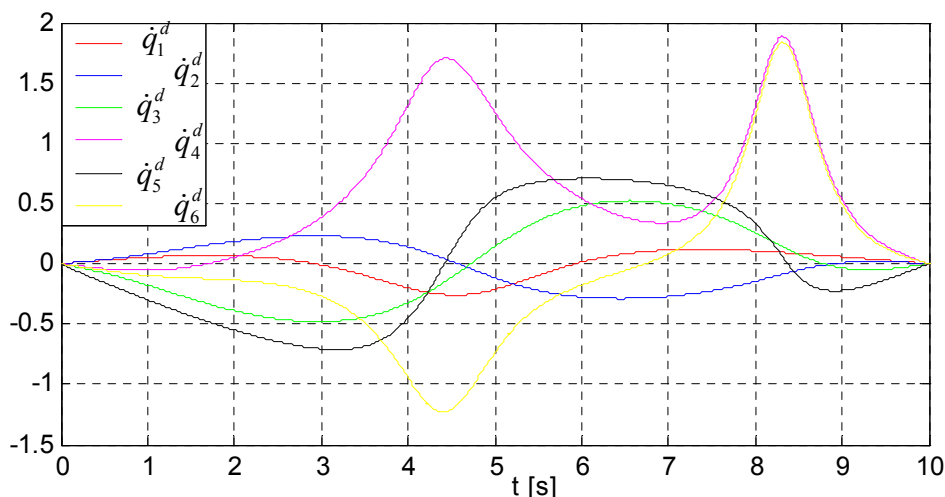


Fig.1.6 Vitesses articulaires désirées (unité : rd/s)

1.4.3 MODELE CINEMATIQUE INVERSE DU DEUXIEME ORDRE (MCI2)

Le modèle cinématique inverse du deuxième ordre (MCI2) s'exprime comme suit

$$\ddot{q} = J^{-1}(\ddot{X} - \dot{J}\dot{q}) \tag{1.10}$$

avec $\dot{J}(q, \dot{q}) = \frac{d}{dt} J(q)$ tel que \dot{J} est la dérivée temporelle du jacobien.

Comme il est déjà mentionné, la commande en position et en orientation des robots peut se faire dans l'espace de la tâche ou dans l'espace des articulations du robot. Etant donné qu'il est plus simple de régler les paramètres du correcteur axe par axe, la commande dans l'espace des articulations semble plus naturelle. Pour le calcul de cette commande, le contrôleur

recupère les consignes des positions articulaires à partir des coordonnées cartésiennes par les transformations ci-dessus. Ensuite, le contrôleur récupère les valeurs des positions des axes du robot à partir des capteurs de position et de vitesse placés sur ses articulations. La différence entre la position désirée et la position réelle de chaque axe du robot est envoyée au correcteur qui calcule la commande qui doit être appliquée à l'actionneur de l'axe considéré.

1.5 LIMITES DES TECHNIQUES DE COMMANDE CLASSIQUES DES ROBOTS

Les robots manipulateurs sont asservis avec des lois de commande appropriées pour que les actionneurs produisent les mouvements désirés. Le contrôle des robots est cependant un problème non linéaire car leur modèle dynamique inverse, ou tout simplement le modèle dynamique d'un tel mécanisme est habituellement décrit par l'équation matricielle suivante :

$$A(q)\ddot{q} + H(q, \dot{q}) = \Gamma \quad (1.11)$$

ou sous la forme explicite suivante [Arms-86] :

$$A(q)\ddot{q} + B(q)[\dot{q} \dot{q}] + C(q)[\dot{q}^2] + G(q) = \Gamma \quad (1.12)$$

où Γ est le couple moteur, $A(q)$ est une matrice de dimension $n \times n$ définissant l'inertie du robot, B est un vecteur de dimension $n \times 1$ représentant les termes centrifuge, C est le vecteur représentant les termes de Coriolis. $G(q)$ est un vecteur de dimension $n \times 1$ représentant les termes relatifs à la gravité. Chaque élément de $A(q)$ et $G(q)$ est une fonction qui dépend du vecteur des positions de tous les axes du robot. Chaque élément de B et C est, quant à lui, une fonction des vecteurs des positions et de vecteur des vitesses [Craig-86]. De plus, il est possible de rajouter un modèle de frottement $F(q, \dot{q})$ qui peut être fonction des positions et des vitesses des axes.

Les éléments des matrices A , B , C , G et F représentant le modèle du robot PUMA 560 sont donnés en [Annexe 1] [Arms-86] [Pete-94].

1.5.1 COMMANDE PID

Dans l'industrie, la commande des robots se fait généralement au moyen de cartes d'axes assurant l'asservissement en position de chacun des axes du robot grâce à des correcteurs type PID (correcteur Proportionnel – Intégrative – Dérivative) à gains constants ou variables. Dans ce cas, les positions et vitesses articulaires désirées sont comparées avec les positions et vitesses obtenues à partir de capteurs placés à la sortie du système. Cette architecture de commande est illustrée à la figure 1.7

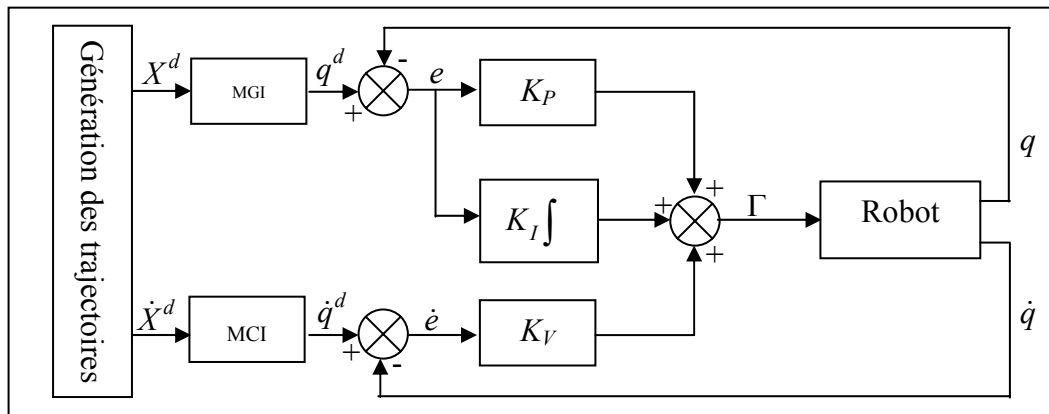


Fig.1.7 Schéma de commande PID

La loi de commande PID est donnée par :

$$\Gamma = K_V \dot{e} + K_P e + K_I \int e dt \quad (1.13)$$

avec $q^d = g(X^d)$ et $\dot{q}^d = J(q^d)^{-1} \dot{X}^d$

Les matrices K_p , K_v et K_I sont diagonales définies positives de dimension $(n \times n)$ dont les éléments représentent respectivement les gains proportionnels, dérivés et intégraux des articulations. q^d et \dot{q}^d sont les vecteurs des positions et vitesses articulaires désirées.

Les avantages de l'architecture de contrôle de la figure 1.7 sont la facilité d'implémentation et le faible coût en calcul.

Il est bien connu que les contrôleurs PID, lorsqu'ils sont convenablement ajustés, fournissent des solutions satisfaisantes pour beaucoup d'applications pratiques sans pour autant avoir besoin d'une description détaillée de la dynamique du système. Cependant, dans le domaine de la robotique cela est valable uniquement pour les mouvements lents où les actionneurs sont munis de réducteurs de vitesse.

Certaines tâches confiées aux robots sont délicates et exigent une très grande précision dynamique sous des trajectoires rapides, excluant ainsi l'utilisation de la commande PID.

1.5.2 COMMANDES NECESSITANT LA CONNAISSANCE DU MODELE DYNAMIQUE

Pour remédier aux problèmes liés au schéma de commande PID, il est naturel de concevoir les systèmes de commandes qui prennent en compte tout ou partie des forces d'interaction dynamiques, notamment celle basée sur la linéarisation et le découplage du système à commander [khal-99]. Elle est connue sous le nom de commande linéarisante ou commande par couple calculé. L'élaboration d'une loi de commande qui linéarise et découple les équations est simplifiée par le fait que le nombre d'actionneurs est égale aux nombre de

variables articulaires et que le modèle dont on dispose est un modèle inverse qui exprime l'entrée Γ du système en fonction du vecteur d'état (q, \dot{q}) et de \ddot{q} . Ces propriétés font que les équations du robot définissent un système plat [Flies-92].

Les premiers travaux portant sur cette technique de commande ont donné lieu à la conception de la commande linéarisante à paramètres connus et constants (figure 1.8).

En posant $e = q^d - q$ et $\dot{e} = \dot{q}^d - \dot{q}$

La dynamique de l'erreur en boucle fermée est :

$$\dot{e} = Ae + Bw \quad (1.14)$$

avec

$$e = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}; A = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

$$w = -\mathbf{A}(q)^{-1}[\Gamma - \mathbf{H}(q, \dot{q})] + \ddot{q}^d$$

Le problème est réduit à la recherche d'une commande linéaire qui assure les performances désirées en boucle fermée, i.e. recherche de F, G, H et J dans (1.15)

$$\begin{aligned} \dot{Z} &= FZ + Ge \\ w &= HZ + Je \end{aligned} \quad (1.15)$$

où

$$\begin{aligned} w(t) &= [(sI - F)^{-1}G + J]e(t) \\ &\equiv C(s)e(t) \end{aligned}$$

La notation ci-dessus indique que $w(t)$ est la sortie du système $C(s)$ quant l'entrée $e(t)$ est appliquée. Le retour d'état statique suivant est souvent utilisé :

$$\begin{aligned} F &= G = H = 0 \\ J &= K \\ w &= k_v \dot{e} + k_p e \equiv Ke \end{aligned} \quad (1.16)$$

L'application de la loi de commande non linéaire

$$\Gamma = \mathbf{A}(q)v + \mathbf{H}(q, \dot{q}) \quad (1.17)$$

avec $v = \ddot{q}^d + w$

donne le système (1.18) en boucle fermée

$$\ddot{e} + K_v \dot{e} + K_p e = 0 \quad (1.18)$$

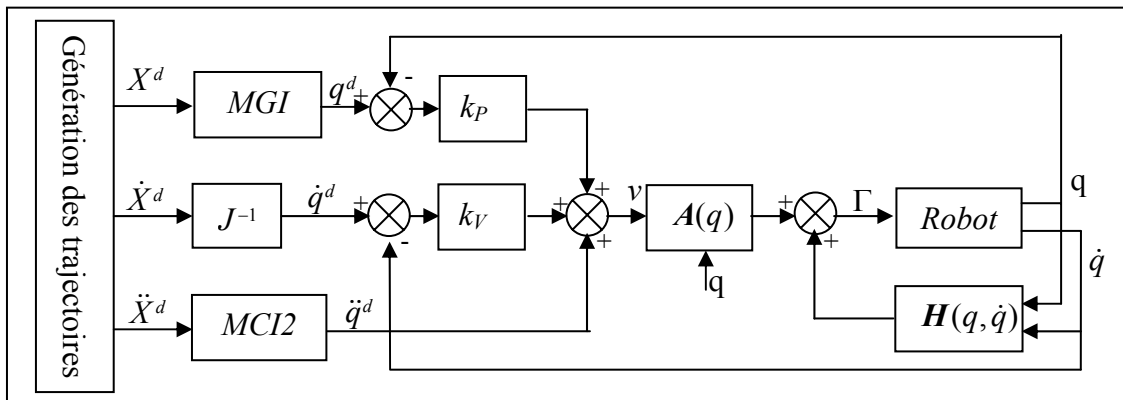


Fig.1.8 Schéma de commande par couple calculé

La commande par découplage non linéaire d'un robot découle d'une exploitation de la forme inverse de son modèle dynamique. D'autres techniques se basant sur la connaissance du modèle ont été appliquées aux robots. On cite entre autres les techniques de commande passive [Pade-88] et [Slot-87]. De telles lois de commande permettent de modifier l'énergie naturelle du robot pour qu'il satisfasse les objectifs désirés.

La mise en oeuvre de ces méthodes exige le calcul du modèle dynamique en ligne et la connaissance des valeurs des paramètres inertiels et de frottements.

La modélisation réalisée à partir de lois physiques, consiste à déterminer la structure des équations qui régissent le comportement du robot, et aussi, à fixer, a priori la valeur de ses paramètres (masses, inerties, frottements...etc.).

Plusieurs techniques sont utilisées pour calculer les paramètres inertiels d'un robot [khal-99]

▪ Le calcul

A partir de considérations géométriques sur les corps constitutifs du robot et de l'hypothèse d'une répartition uniforme des masses, on peut obtenir une évaluation des paramètres inertiels du mécanisme.

▪ Mesures

On procède à des essais expérimentaux sur chacun des corps du robot pris isolément [arms-86] :

- Les masses des corps sont obtenues par simple pesée ;
- Les positions des centres de masse sont repérées par les points d'équilibre des corps ; les moments principaux d'inertie sont obtenus par la méthode du pendule.

La grande difficulté vient du fait qu'en réalité les robots industriels sont difficilement démontables. De plus, les robots sont amenés à manipuler des objets de formes variées et dont

les caractéristiques sont inconnues. La meilleure solution est de recourir à des techniques d'identification.

❖ PRINCIPE D'IDENTIFICATION DES PARAMETRES DYNAMIQUES DES ROBOTS

L'identification des paramètres dynamiques des robots a fait l'objet de nombreuses recherches [khal-99] [yosh-90]. Les méthodes proposées présentent de nombreux points communs, notamment

- l'utilisation d'un modèle de connaissance linéaire vis-à-vis des paramètres dynamiques inconnus (modèle dynamique, modèle de puissance ou modélisation des efforts exercés sur la base) ;
- La construction d'un système linéaire surdéterminé par échantillonnage du modèle à différents instants au cours d'un mouvement du robot ;
- L'estimation des paramètres par la méthode de régression linéaire (moindres carrés ordinaires ou autres variantes).

Dans ce cas la loi de commande (1.17) devient

$$\Gamma = \hat{A}(q)v + \hat{H}(q, \dot{q}) \quad (1.18)$$

où les fonctions $\hat{A}(q)$ et $\hat{H}(q, \dot{q})$ sont les estimés de $A(q)$ et $H(q, \dot{q})$.

Comme on l'a déjà mentionné, l'implémentation de loi de commande par découplage non linéaire (à paramètres fixes) nécessite la connaissance complète et très fine du modèle du robot. Dans une telle situation cette stratégie de commande présente des performances très satisfaisantes en terme de poursuite. Cependant, dans la pratique cette exigence est très difficile à satisfaire vu les non linéarités et le couplage du modèle, ajouter à cela les perturbations agissant sur le robot. Dans de telles conditions, cette technique de commande s'avère inefficace et reste presque inutilisable. Afin d'éviter les inconvénients liés aux schémas de commande classiques, il est naturel d'appliquer les techniques de commande dites robustes

1.5.3 COMMANDE ROBUSTE

Plusieurs familles de commande robuste ont été appliquées aux robots. Celles-ci peuvent être divisés en trois grandes classes. La commande robuste linéaire multivariable, la commande robuste à structure variable et la commande robuste passive [Abda-91].

❖ APPROCHE LINEAIRE MULTIVARIABLE

Pour contourner les problèmes d'incertitude des fonction $A(q)$ et $H(q, \dot{q})$, la loi de commande (1.18) est adoptée, où les fonctions $\hat{A}(q)$ et $\hat{H}(q, \dot{q})$ sont les estimés de $A(q)$ et $H(q, \dot{q})$.

$$\Gamma = \hat{A}(\ddot{q}^d + w) + \hat{H} \quad (1.19)$$

L'approche robuste linéaire multivariable se résume à la construction du contrôleur linéaire $C(s)$ tel que le système en boucle fermée (1.20) reste stable dans tous les sens en présence de perturbations [Corl-81]. En d'autres termes, choisir $C(s)$ tel que l'erreur $e(t)$ dans (1.20) reste stable

$$\dot{e} = Ae + B(w + \eta) \quad (1.20)$$

$$w(t) = C(s)e(t) \quad (1.21)$$

avec

$$\eta = E(w + \ddot{q}^d) + A^{-1}\Delta H$$

et

$$E = A^{-1}\hat{A} - I_n$$

$$\Delta H = \hat{H} - H$$

L'approche linéaire multivariable est illustrée à la figure 1.9

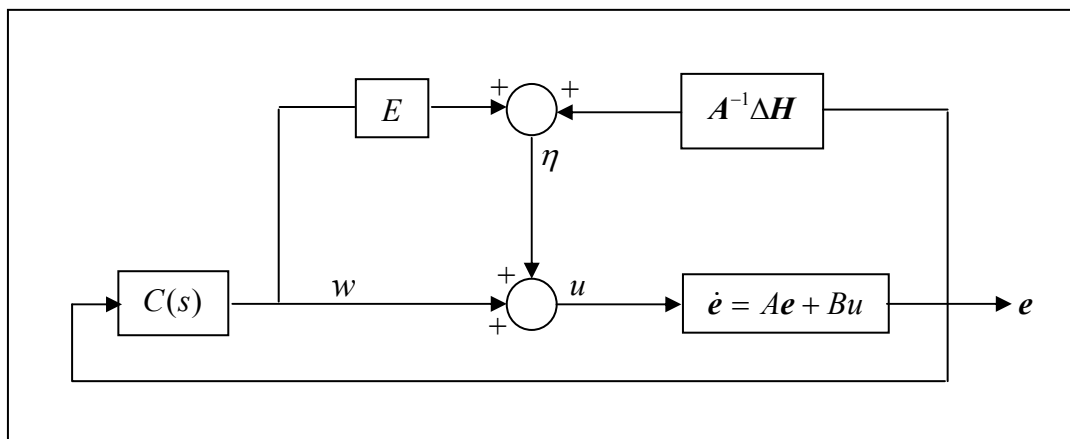


Fig.1.9 Commande robuste linéaire multivariable

Le terme non linéaire η est en fonction de e et w . Il représente les perturbations de la linéarisation exacte de l'erreur dynamique, qui sont causées par les incertitudes sur le modèle, perturbations extérieures, variation des paramètres et éventuellement les erreurs de mesure.

En général, le théorème de petit gain [Deso-75], théorème de passivité ou théorème de stabilité totale [Ande-89] sont évoqués pour construire $C(s)$. La majorité de ces contrôleurs sont conçus en utilisant la paramétrisation de Yula et le théorème de commande H^∞ [Vidy-85]

❖ COMMANDE ROBUSTE A STRUCTURE VARIABLE (CSV)

La philosophie de l'approche robuste à structure variable (CSV) est de forcer le signal d'erreur vers la surface de glissement [Abda-91]. Une fois que le système entre en mode de glissement, il devient insensible aux incertitudes du modèle et aux différentes perturbations [Utk-77] [Youn-78]. La première application de la commande à structure variable (CSV) aux robots manipulateurs a été effectuée par Young [Youn-78] où le problème de régulation de consigne (\dot{q}^d) est résolu en utilisant la loi de commande suivante :

$$\Gamma_i = \begin{cases} \Gamma_i^+ & \text{si } s_i(e_i, \dot{q}_i) > 0 \\ \Gamma_i^- & \text{si } s_i(e_i, \dot{q}_i) < 0 \end{cases} \quad (1.22)$$

où $i = 1, \dots, n$

et $s_i(e_i, \dot{q}_i) = c_i e_i + \dot{q}_i$, $c_i > 0$

Il a été montré qu'en utilisant la hiérarchie de surfaces de glissement (s_1, \dots, s_n) et en connaissant la plage de variation des incertitudes dans le modèle du robot, on peut trouver Γ^+ et Γ^- qui force le signal de l'erreur vers l'intersection des surfaces de glissement, qui glisse en suite vers zéro. Ce contrôleur élimine le couplage non linéaire des articulations en forçant le système vers le mode de glissant. D'autres techniques de commande à structure variables ont été appliqués [Nico-84], [Bail-87], [Barto-85] et [Morg-85]. Malheureusement, pour la plupart de ces schémas, la loi de commande est discontinue et peut, cependant, créer le phénomène de chattering qui peut exciter le système dans la dynamique non modélisable. Pour résoudre ce problème, Slotine a introduit dans la loi de commande un terme de saturation [Slot-85] [Slot-83].

$$\Gamma = \hat{A}u + \hat{H} \quad (1.23)$$

où u_i est défini comme suite :

$$u_i = \hat{u}_i - \Phi(q, \dot{q}, t) \quad (1.24)$$

La loi de commande est composée du terme de compensation non linéaire \hat{u}_i et d'une fonction de saturation $\Phi(q, \dot{q}, t)$

Les techniques de commande à structure variable donnent de bons résultats théoriques, mais elles ne sont pas appliquées réellement aux robots. Dans la pratique et pour éviter le phénomène de chattering, la stabilité asymptotique est souvent sacrifiée [Abda-91].

Les techniques de commande robustes considèrent que le modèle du robot est connu avec une certaine incertitude, les lois de commande robuste sont généralement des versions modifiées des lois de commande classique auxquelles s'ajoute un terme de robustesse. Pour des incertitudes importantes on doit utiliser les gains important qui peuvent limiter leurs applications pratiques.

1.5.4 COMMANDE ADAPTATIVE

Le modèle dynamique du robot ne pouvant être parfait, il été naturelle d'appliquer les théories de la commande adaptative pour calculer ou ajuster en ligne les valeurs des paramètres dynamiques utilisés dans la commande. Ceci est rendu réalisable grâce au développement de la micro-informatique et l'utilisation des techniques d'identification permettant d'implanter des fonctions d'auto adaptation (autoréglage) dans les régulateurs.

Les schémas de commande adaptative diffèrent selon l'objectif de commande et le signal qui forme les lois d'adaptation paramétriques. L'objectif du control détermine la structure fondamentale des paramètres à adapter en ligne, la loi d'adaptations peut être formées par le signal qui mesure l'erreur entre les paramètres estimés et les vrais paramètres du système (i, e erreurs de prédiction) ou l'erreur entre la sortie désirée et la sortie mesurée (i, e erreur de poursuite de sortie).

On ne peut pas citer dans le cadre de ce travail toutes les techniques adaptatives appliquées aux robots manipulateurs, rencontrées dans la littérature. Mais, nous nous contentons de citer quelques techniques les plus significatives.

En générale les techniques adaptatives classiques citées dans la littérature sont divisées en deux groupes : le premier groupe est la version adaptative du schéma de commande par couple calculé qui tend à linéariser et à découpler les équations du mouvement du robot manipulateur. Le second groupe utilise la propriété liée à la passivité des robots manipulateurs.

Considérons que le modèle dynamique du robot est linéaire par rapport aux paramètres dynamiques. Le modèle du robot (1.11) peut s'exprimer comme suit

$$\Gamma = Y(q, \dot{q}, \ddot{q})\theta \quad (1.25)$$

La loi de commande a la même structure que celle de la commande par découplage non linéaire :

$$\Gamma = \hat{A}(q)(\ddot{q}^d + K_v(\dot{q}^d - \dot{q}) + K_p(q^d - q)) + \hat{H}(q, \dot{q}) = Y(q, \dot{q}, \ddot{q})\hat{\theta} \quad (1.26)$$

En égalisant (1.11) et (1.26) on aura :

$$A(q)\ddot{q} + H(q, \dot{q}) = \hat{A}(q)(\ddot{q}^d + K_v(\dot{q}^d - \dot{q}) + K_p(q^d - q)) + \hat{H}(q, \dot{q}) \quad (1.27)$$

En ajoutant et en retranchant le terme $\hat{A}(q)\ddot{q}$ à gauche de l'équation (1.27), il vient

$$\hat{A}(q)(\ddot{e} + K_v\dot{e} + K_p e) = \tilde{A}(q)\ddot{q} + \tilde{H}(q, \dot{q}) = Y(q, \dot{q}, \ddot{q})\tilde{\theta} \quad (1.28)$$

Finalement, l'erreur dynamique peut s'écrire comme suit :

$$\ddot{e} + K_v\dot{e} + K_p e = \hat{A}^{-1}Y(q, \dot{q}, \ddot{q})\tilde{\theta} = \Phi\tilde{\theta} \quad (1.29)$$

Pour déterminer les lois d'adaptations, on réécrit l'équation (1.29) sous la forme d'état suivante :

$$\dot{e} = Ae + B\Phi\tilde{\theta} \quad (1.30)$$

avec

$$A = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}; B = \begin{bmatrix} 0 \\ I \end{bmatrix} \text{ et } e = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}$$

où 0 est une matrice nulle et I est la matrice d'identité de dimension $(n \times n)$

On choisit la fonction de Lyapunov suivante :

$$V = e^T P e + \tilde{\theta}^T \Lambda \tilde{\theta} \quad (1.31)$$

avec $\Lambda = \Lambda^T > 0$ est P est l'unique matrice définie positive solution de l'équation de Lyapunov

$$A^T P + P A = -Q \quad (1.32)$$

Par dérivation de la fonction de Lyapunov (1.31) par rapport et en utilisant la loi d'adaptation paratétrique suivante

$$\dot{\tilde{\theta}} = \Lambda^{-1} \Phi^T(q, \dot{q}, \ddot{q}) \hat{A}^{-1}(q) B^T P e \quad (1.33)$$

L'expression de \dot{V} devient

$$\dot{V} = -e^T Q e \leq 0 \quad (1.34)$$

Par conséquent, le vecteur e est borné et tend vers zéro.

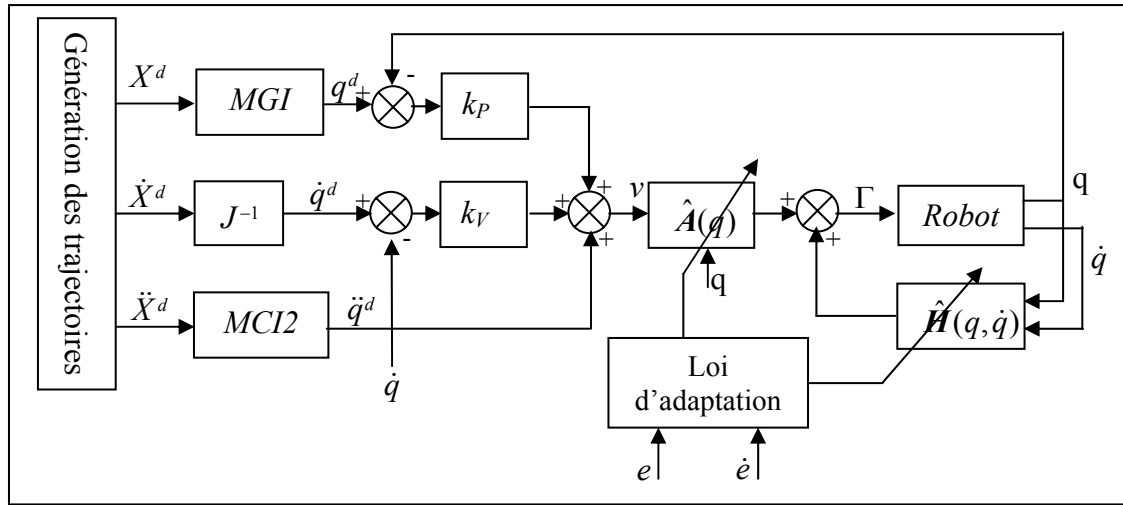


Fig.1.10 Commande par découplage non linéaire adaptative

Cette méthode contient deux hypothèses restrictives : la première est que l'on doit connaître les accélérations articulaires ; la seconde est que l'inverse de la matrice d'inertie estimée doit être bornée [Craig-86].

❖ COMMANDE ADAPTATIVE SANS INVERSION DE LA MATRICE D'INERTIE

Spong et Ortiga [Spon-88] ont proposé une approche ne nécessitant pas l'inversibilité de la matrice d'inertie. La même technique a été utilisée dans la commande robuste par Spong et Vidyasagar [Spon-87]

Dans cette approche la loi de commande est choisie de la forme suivante :

$$\Gamma = A_0(q)(v + \delta v) + H_0(q, \dot{q}) \quad (1.35)$$

avec $A_0 = A_0^T > 0$ et H_0 les estimés a priori de A et H respectivement avec des paramètres fixes.

En combinant (1.35) et (1.11) on aura une équation similaire à (1.28)

$$A_0(\ddot{e} + K_v \dot{e} + K_p e + \delta v) = \Delta A \ddot{q} + \Delta H = Y(q, \dot{q}, \ddot{q}) \Delta \theta \quad (1.36)$$

avec

$\Delta \theta = \theta_0 - \theta$ est un vecteur fixe dans R^n .

Finalement on écrit

$$\ddot{e} + K_v \dot{e} + K_p e = A_0^{-1} Y \Delta \theta - \delta v = \Phi_0 \Delta \theta - \delta v \quad (1.37)$$

On choisit la loi de control δv comme suit

$$\delta v = -\Phi_0 \Delta \hat{\theta} \quad (1.38)$$

et en choisissant la loi d'adaptation

$$\Delta \dot{\hat{\theta}} = -\Lambda^{-1} \Phi_0^T B^T P e \quad (1.39)$$

avec P satisfait l'équation 1.32 et choisissant la fonction de Lyapunov (1.31), on montre que tous les signaux sont bornés

❖ COMMANDE DYNAMIQUE SANS CALCUL DU VECTEUR ACCELERATION

Le problème de calcul du vecteur accélération a été résolu par Middleton [Midd-88] en introduisant une loi de commande filtrée. Cette technique a été utilisée aussi par Narendra [Nare-80].

Afin de développer l'algorithme adaptatif fondé sur le modèle dynamique complet, Slotine et Li [Slot-87] ont exploité la propriété de passivité du robot (système qui dissipe de l'énergie). De telles lois de commande permettent de modifier l'énergie naturelle du robot pour qu'il satisfasse les objectifs désirés. Cette stratégie ne conduit pas à un système linéaire en boucle fermée mais son élaboration ne dépend pas du vecteur accélération.

D'autres approches ne satisfaisant pas nécessairement le critères de stabilité globale telles que les approches adaptatives par linéarisation autour de la trajectoire, les techniques qui supposent des variations lentes des paramètres du système et celles n'incorporant pas les paramètres estimés dans leurs lois de commande on été appliquées aux robots [Bale-83] [Sing-85].

Face aux insuffisances des lois de commande adaptatives classiques décrites précédemment et en vue d'assurer de meilleures performances de suivi de trajectoire en position et en orientation en présence des incertitudes paramétriques et dynamiques, nous nous attachons dans ce mémoire à l'application des lois de commande adaptatives floues.

1.6 CONCLUSION

Dans ce chapitre nous avons en premier lieu défini la tâche à confier au robot. Cette tâche qui est le soudage, nécessite la poursuite de la trajectoire en position et en orientation. Puis on a exposé les raisons qui justifient l'utilisation de la commande adaptative floue en faisant un tour d'horizon des différentes techniques adaptatives et non adaptatifs non floues appliquées aux robots manipulateurs.

CHAPITRE 2 :

SYSTEMES FLOUS

SYSTEMES FLOUS

2.1 INTRODUCTION

Face à la complexité croissante des systèmes à commander, il devient de plus en plus ardu d'élaborer leurs modèles sous forme d'équations mathématiques. De ce fait, le calcul de la loi de commande qui répond de façon optimale à l'objectif du système, devient lui aussi de plus en plus difficile. Une alternative à la détermination explicite du modèle d'un système, est l'utilisation des systèmes flous. Ceux-ci sont basés sur la notion d'ensembles flous et sur le raisonnement approximatif proche du raisonnement humain.

Après l'introduction de la théorie des ensembles flous en 1965 par L. A. Zadeh et les travaux du pionnier M. Mamdani en 1974. Plusieurs travaux de recherche ont montré que l'application de la commande floue aux robots manipulateurs donne de bonnes performances en la comparant aux techniques de commande non floues [Kwan-99] [lin-97] [Yi-97] [Neo-95][Sun-95] [Kim-95] [Chek-03]

Le but principal de ce chapitre est de donner un bref aperçu sur la théorie des ensembles flous et les systèmes flous en se limitant à la présentation d'une classe des systèmes flous directement liée aux méthodes de commande appliquées dans ce mémoire, à savoir les systèmes flous de Takagi-Sugeno à conclusions fonctionnelles. Ces derniers, qui sont des approximateurs universels, sont utilisés dans la suite du mémoire pour la construction des lois de commande adaptatives floues.

2.2 SYSTEME FLOU

La construction d'un système flou consiste à trouver une relation de correspondance entre l'espace d'entrée et l'espace de sortie pour approximer, avec un degré de précision donné, la fonction ou la dynamique désirée. Si d'un point de vue mathématique les systèmes sont classés selon la nature des équations qui les caractérisent (linéaires, non linéaires, ...etc.), les systèmes flous sont quant à eux répertoriés selon leur nature structurelle. On distingue classiquement deux grandes familles de systèmes flous [Bouk-00] : Les systèmes flous à conclusions symboliques (systèmes de Mamdani) et les systèmes flous à conclusion fonctionnelles (systèmes flous de Takagi-Sugeno). Ces deux types de systèmes flous sont basés sur une collection de règles "Si ... Alors". Dans les deux cas, les prémisses des règles sont exprimées symboliquement. Seule l'expression des conclusions des règles permet alors

de dissocier les deux familles de systèmes. Les systèmes de Mamdani utilisent en effet des conclusions symboliques de même nature que les prémisses alors que les systèmes de Sugeno exploitent des conclusions numériques. Cette différence dans l'écriture des conclusions des règles permet de distinguer les deux types de systèmes à leur aspect externe. D'un point de vue interne, un mécanisme de calcul est associé à chaque famille de système. Pour les systèmes de Sugeno, celui-ci est purement numérique et s'exprime aisément de manière analytique selon une approche unique, commune à l'ensemble des systèmes de la famille. Par contre l'implantation des systèmes de Mamdani peut être envisagée de différentes façons. On aboutit alors à une typologie des systèmes de Mamdani telle que celle développée par Foulloy [Foul-92].

L'architecture principale d'un système flou (figure 2.1) se compose de quatre blocs principaux [Buhl-94] [Kasa-98] [Wang-94] [Labi-2005] :

- Base des règles
- Fuzzification
- Inférence
- Défuzzification

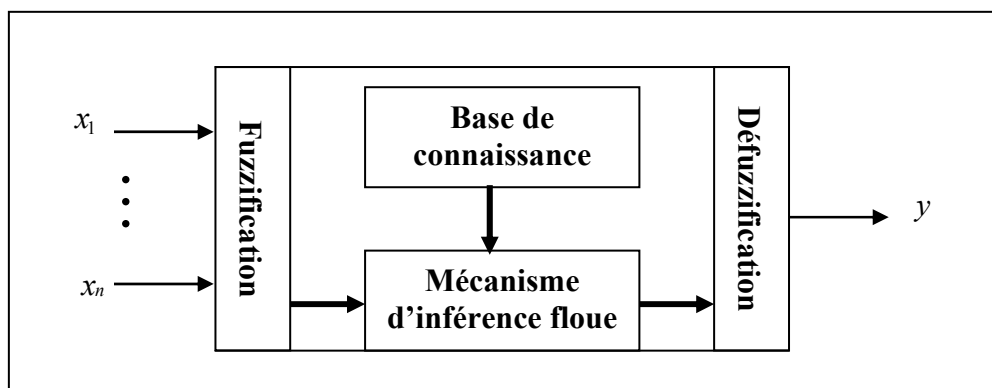


Fig.2.1 Configuration de base d'un système flou MISO

2.2.1 BASE DES REGLES

Un système flou multi-entrées mono-sortie (MISO) est une relation de correspondance non linéaire du vecteur d'entrée $x = [x_1 \ x_2 \ \dots \ x_n]^T \in R^n$ vers une sortie $y \in R$.

Soit un système flou MISO ayant n variables d'entrées (x_1, \dots, x_n) . Où chaque variable x_j ($j=1, n$) est caractérisée par m_j ensembles flous $(F_j^1, \dots, F_j^{m_j})$. Ainsi la base des règles floues est constituée par M règles $\left(\bigcup_{i=1}^M R_i \right)$ du type *Si – Alors* tel que :

$$M = \prod_{j=1}^n m_j$$

et la i^{eme} règle s'écrit sous la forme :

$$R_i : \mathbf{Si} \{x_1 \text{ est } F_1^{K_i} \text{ et } x_2 \text{ est } F_2^{K_i} \dots \text{ et } x_n \text{ est } F_n^{K_i}\} \mathbf{Alors } y \text{ est } C_i(y) \quad (2.1)$$

avec $k_j^i \in [1, \dots, m_j]$, $j = 1, \dots, n$ et $i = 1, \dots, M$

où $F_j^{k_j^i}$ est la j^{ieme} valeur linguistique associée à la variable linguistique d'entrée x_j et C_i est la valeur linguistique associée à la variable de sortie y .

La variable linguistique est la description de la variable dont la valeur numérique est imprécise, par des mots simples. Une variable linguistique est généralement représentée par le nom de la variable x (position, vitesse, erreur, ...), l'univers de discours "U" sur lequel elle est défini et les valeurs linguistiques T(x) qui décompose sa plage de variation.

L'univers de discours d'une variable couvrira l'ensemble des valeurs prises par cette variable.

Par exemple : $x = \text{erreur}$.

alors : $T(x) = \{negative\ grand, \text{environ zero}, positive\ grand\}$ et $U = [-10 \ 10]$.

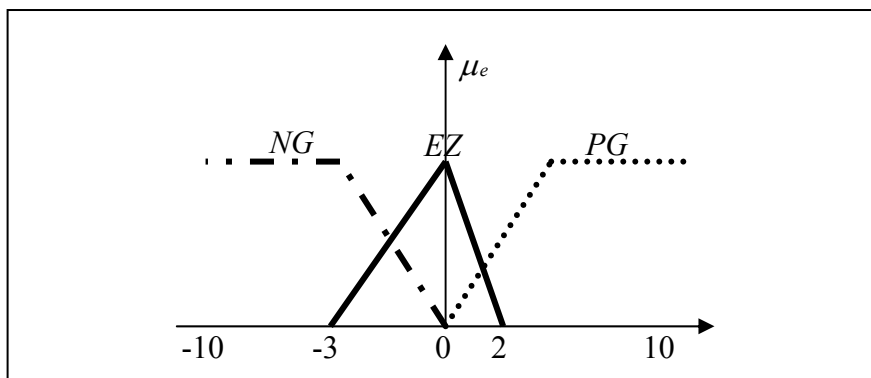


Fig.2.2 Représentation de la variable linguistique "erreur"

Les ensembles flous $F_j^{k_j^i}$ et C_i sont des ensembles flous définis par

$$F_j^{k_j^i} = \left\{ \left(x_j, \mu_{F_j^{k_j^i}}(x_j) \right) : x_j \in R \right\} \quad (2.2)$$

$$C_i = \left\{ \left(y, \mu_{C_i}(y) \right) : y \in R \right\} \quad (2.3)$$

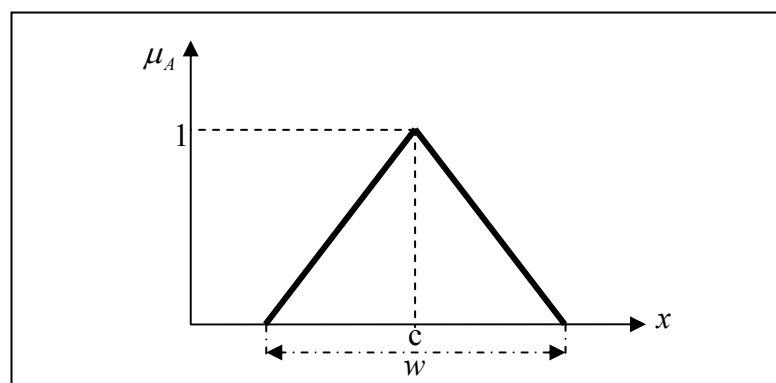
Les fonctions d'appartenance $\mu_{F_j^{k_j^i}}$ et $\mu_{C_i} \in [0 \ 1]$ determine le degrés d'appartenance de la variable linguistique x_j et y aux l'ensembles flous $F_j^{k_j^i}$ et C_i respectivement. Ces dernières

permettent un traitement numérique des variables linguistiques ce qui assure l'implémentation par ordinateur.

Les fonctions d'appartenance les plus utilisées sont [spoo-02] :

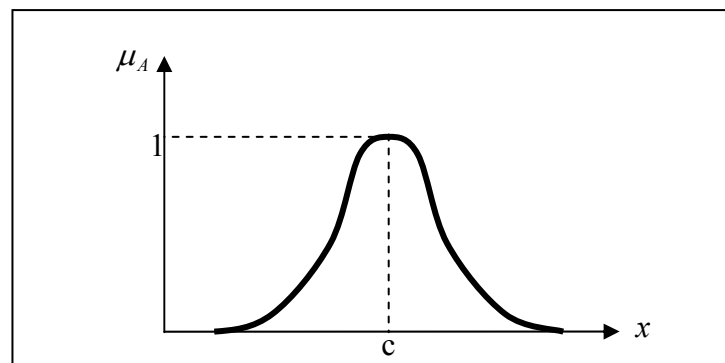
▪ **Triangulaire**

$$\mu_A(x) = \begin{cases} \max \left\{ 0, 1 + \frac{x-c}{0.5w} \right\} & \text{si } x \leq c \\ \max \left\{ 0, 1 + \frac{c-x}{0.5w} \right\} & \text{ailleurs} \end{cases} \quad (2.4)$$



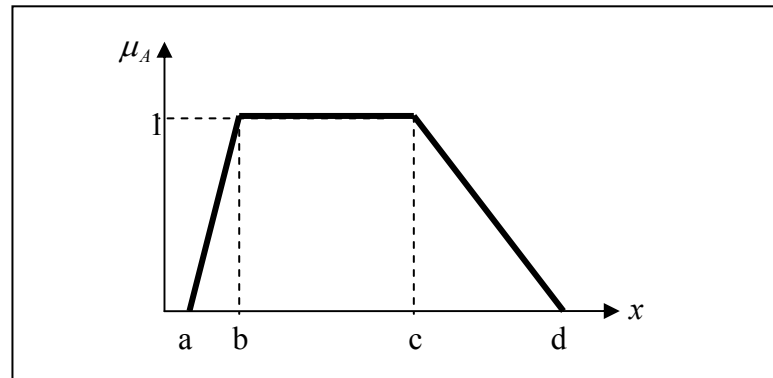
▪ **Gaussienne**

$$\mu_A(x) = \exp \left(-0.5 \left(\frac{x-c}{\sigma} \right)^2 \right) \quad (2.5)$$



▪ Trapézoïdale

$$\mu_A(x) = \max \left\{ \min \left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right\} \quad (2.6)$$



2.2.2 FUZZIFICATION

La fuzzification est simplement le processus d'obtention de la valeur de l'entrée x_j et le calcul de $\mu(x_j)$ pour chaque fonction d'appartenance μ

Deux approches de fuzzification sont souvent utilisées : fuzzification singleton et fuzzification non singleton.

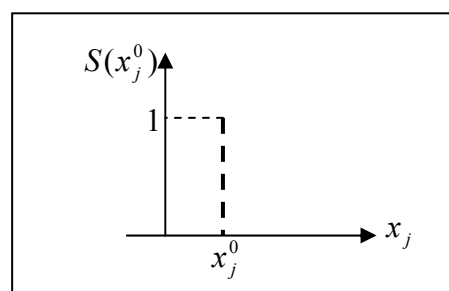
▪ Moteur d'inférence

Le moteur d'inférence permet de générer un ensemble flou tel que :

$$C'(y) = [S(x_1^0) * S(x_2^0) * \dots * S(x_n^0)] \circ \left[\bigcup_{i=1}^M R_i(x_1, \dots, x_n) \right] \quad (2.7)$$

où l'opérateur $*$ représente une t-norme et R_i l'implication floue liée à la $i^{\text{ème}}$ règle et $S(x_j^0)$ est l'ensemble singleton tel que :

$$S(x_j^0) = \begin{cases} 1 & \text{si } x_j = x_j^0 \\ 0 & \text{si } x_j \neq x_j^0 \end{cases} \quad (2.8)$$



Etant donné que l'opération de composition \circ est commutative par rapport à l'opérateur \bigcup , il vient

$$C'(y) = \bigcup_{i=1}^M \left[\left(S(x_1^0) * S(x_2^0) * \dots * S(x_n^0) \right) \circ R_i \right] \quad (2.9)$$

Comme la règle R_i est interprétée par :

$$R_i(x_1, \dots, x_n) = \left(F_1^{K_i}(x_1) * F_2^{K_i}(x_2) * \dots * F_n^{K_i}(x_n) * C_i(y) \right) \quad (2.10)$$

Et la règle R_i génère l'ensemble $C_i'(y)$ tel que :

$$C_i'(y) = \left(S(x_1^0) * S(x_2^0) * \dots * S(x_n^0) \right) \circ R_i \quad (2.11)$$

Qui s'écrit encore :

$$\begin{aligned} C_i'(y) &= \sup_{(x_1, \dots, x_n)} \left[\left(S(x_1^0) * \dots * S(x_n^0) \right) * \left(F_1^{K_i}(x_1) * \dots * F_n^{K_i}(x_n) * C_i(y) \right) \right] \\ &= \left[\sup_{x_1} S(x_1^0) * F_1^{K_i}(x_1) \right] * \dots * \left[\sup_{x_n} S(x_n^0) * F_n^{K_i}(x_n) \right] * C_i(y) \\ &= \left(F_1^{K_i}(x_1) * \dots * F_n^{K_i}(x_n) \right) * C_i(y) \end{aligned} \quad (2.12)$$

On pose

$$\alpha_i = F_1^{K_i}(x_1) * \dots * F_n^{K_i}(x_n) \quad (2.13)$$

D'où

$$C_i'(y) = \alpha_i * C_i(y) \quad (2.14)$$

α_i est le degré d'activation de la $i^{\text{ème}}$ règle.

D'où l'ensemble flou inféré est :

$$C'(y) = \bigcup_{i=1}^M (\alpha_i * C_i(y)) \quad (2.15)$$

2.2.3 INFERENCE

Si l'opérateur t-norme est interprété par le produit, il vient :

$$\alpha_i = F_1^{K_i}(x_1^0) \bullet \dots \bullet F_n^{K_i}(x_n) \quad (2.16)$$

Si * est interprété par le minimum :

$$\alpha_i = \min(F_1^{k_i}(x_1^0), \dots, F_n^{k_i}(x_n)) \quad (2.17)$$

La t-norme mesure simplement la conjonction dans les prémisses de chaque règle floue (c'est une opération de base de la logique floue).

2.2.4 DEFUZZIFICATION

Comme on l'a vu à la section précédente, les méthodes d'inférences fournissent un ensemble flou $C'(y)$. Il s'agit donc d'une information floue à partir de laquelle il faut extraire la variable de sortie y . Les actionneurs actuels, utilisés dans les boucles de commande ne s'accommodent pas de ce genre de décision, il convient de la transformer en une grandeur de commande précise: c'est le but de l'étape de défuzzification. La défuzzification combine les conclusions de toutes les règles et fournit une valeur numérique qui représente la sortie du système.

La méthode de défuzzification la plus utilisée en technologie est celle des hauteurs pondérées. Celle-ci s'exprime de la manière suivante :

$$y = \frac{\sum \alpha_i c_i}{\sum \alpha_i} \quad (2.18)$$

où c_i est la valeur de y pour laquelle $\mu_{C_i}(y)$ est maximale.

Il faut noter que pour définir un système flou (MIMO) d'ordre m il suffit de définir m systèmes flous (MISO).

2.3 SYSTEME FLOU DE TAKAGI-SUGENO

Les systèmes flous de Takagi-Sugeno utilisent les règles floues de la forme :

$$R_i : \text{Si } \{x_1 \text{ est } F_1^{K_i} \text{ et } x_2 \text{ est } F_2^{K_i} \dots \text{ et } x_n \text{ est } F_n^{K_i}\} \text{ Alors } y \text{ est } c_i(y) \quad (2.19)$$

Dans ce cas, les prémisses sont les mêmes que pour les systèmes flous standards définis ci-dessus, cependant, les conséquences ne sont pas des ensembles flous; ils sont des fonctions. En particulier, $c_i(y) = g_i(x)$ est la conséquence de la $i^{\text{ème}}$ règle.

Pour calculer la sortie du système flou de Takagi-sugeno avec la fuzzification par la moyenne pondérée on écrit :

$$y = \frac{\sum_{i=1}^M c_i \mu_i}{\sum_{i=1}^M \mu_i} \quad (2.20)$$

On note que le système flou peut être vu comme une interpolateur entre M fonctions de sortie.

Pour un système flou, on suppose que pour tout $x \in R^n$, on a $\sum_{i=1}^p \mu_i \neq 0$.

Parfois, les sorties des conséquences de chaque règle est une combinaisons linéaire des ensembles de fonctions linéaires Lipschisiennes, $\gamma_j(x) \in R$, $j = 1, 2, \dots, m-1$ tel que

$$c_i = g_i(x) = a_i^0 + a_i^1 \gamma_1(x) + \dots + a_i^{m-2} \gamma_{m-2}(x) + a_i^{m-1} \gamma_{m-1}(x) \quad (2.21)$$

$i = 1, \dots, M$.

Dans le cas particulier où $m-1 = n$, et $\gamma_j(x) = x_j$ on obtient le système de Takagi-sugeno d'ordre un.

Une règle typique utilisée par Sugeno pour construire ses systèmes est de la forme :

$$R_i : \mathbf{Si} \left\{ x_1 \text{ est } F_1^{K_1^i} \text{ et } x_2 \text{ est } F_2^{K_2^i} \dots \text{ et } x_n \text{ est } F_n^{K_n^i} \right\} \text{ Alors } c_i = a_i^0 + a_i^1 x_1 + \dots + a_i^{n-1} x_{n-1} + a_i^n x_n$$

où n est le nombre d'entrées de système, les a_i^j sont les coefficients décrivant le polynôme caractéristique de la conclusion.

En utilisant la méthode d'inférence singleton et la défuzzification par la moyenne pondérée, nous obtenons la sortie du système floue suivante :

$$y = \frac{\sum_{i=1}^M \left(\prod_{j=1}^n \mu_{F_n^{K_n^i}}(x_j) \right) \left(a_i^0 + \sum_{j=1}^n a_i^j x_j \right)}{\sum_{i=1}^M \left(\prod_{j=1}^n \mu_{F_n^{K_n^i}}(x_j) \right)} \quad (2.22)$$

$$\text{On pose } \alpha_i = \prod_{j=1}^n \mu_{F_n^{K_n^i}}(x_j)$$

On aura

$$y = \frac{\sum_{i=1}^M \alpha_i \left(a_i^0 + \sum_{j=1}^n a_i^j x_j \right)}{\sum_{i=1}^M \alpha_i} \quad (2.23)$$

où α_i représente le degré d'activation de la $i^{\text{ème}}$ règle.

La représentation schématique d'un système flou de Sugeno à deux entrées et deux ensembles flous pour chaque variable d'entrée est illustrée sur la figure 2.6

Dans ce cas l'équation (2.23) peut s'écrire :

$$y = W^T \theta \tag{2.24}$$

où W^T est le vecteur de régression donné par :

$$W^T = \begin{bmatrix} \frac{\alpha_1}{\sum_{i=1}^M \alpha_i} & \frac{\alpha_1 x_1}{\sum_{i=1}^M \alpha_i} & \dots & \frac{\alpha_1 x_n}{\sum_{i=1}^M \alpha_i} & \dots & \dots & \frac{\alpha_M}{\sum_{i=1}^M \alpha_i} & \frac{\alpha_M x_1}{\sum_{i=1}^M \alpha_i} & \dots & \frac{\alpha_M x_M}{\sum_{i=1}^M \alpha_i} \end{bmatrix}$$

et θ est le vecteur des paramètres ajustables donné par :

$$\theta = [a_1^0 \ a_1^1 \ \dots \ a_1^n \ \dots \ a_M^0 \ a_M^1 \ \dots \ a_M^n]^T$$

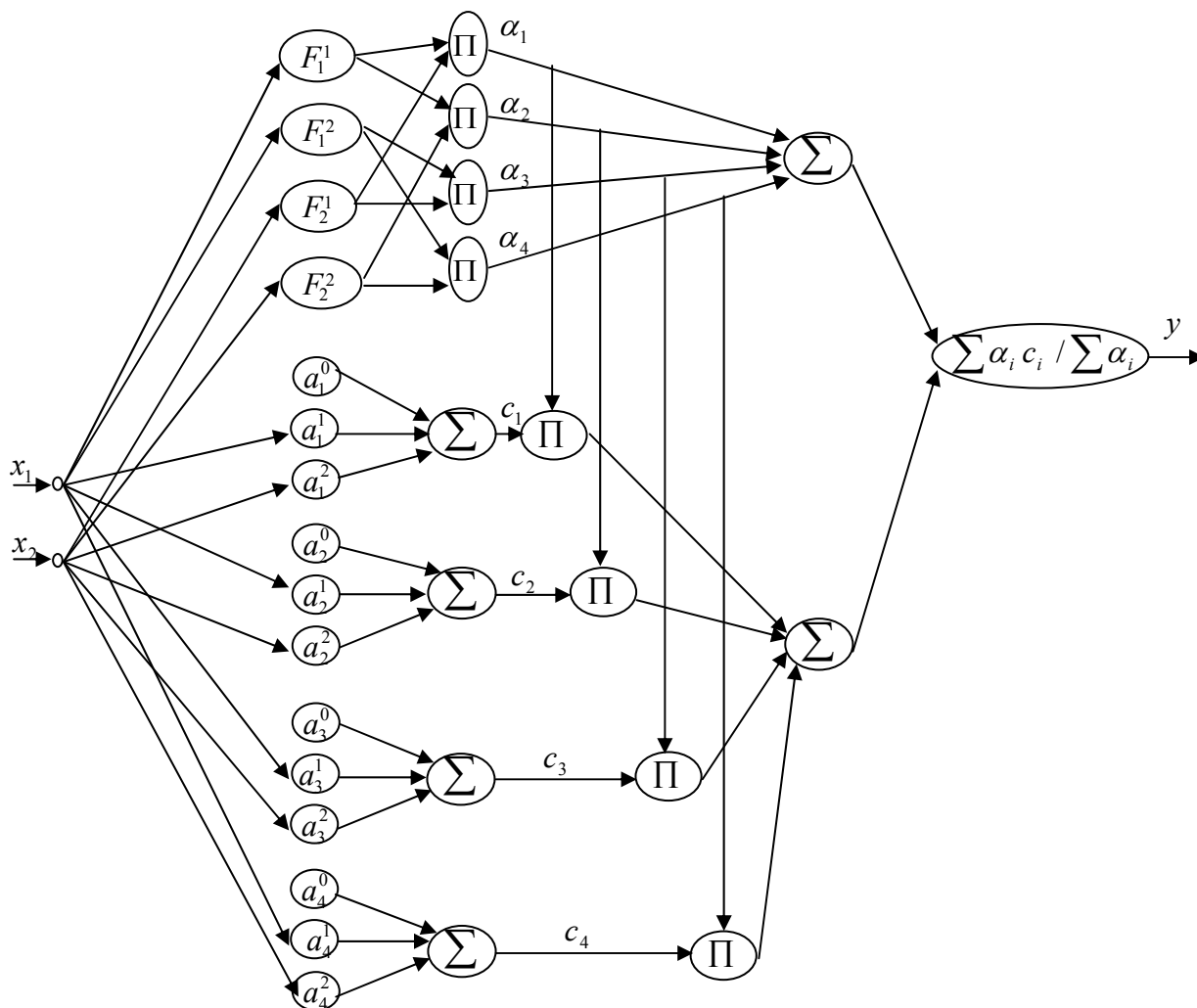


Figure 2.6 Structure du système flou de Sugeno à deux variables d'entrée définie chacune par deux ensembles flous

2.4 ADAPTATION DES PARAMETRES DU SYSTEMES FLOU DE SUGENO

D'une manière générale, le problème d'identification floue consiste à choisir une forme des règles floues appropriée puis à concevoir des lois d'ajustement de paramètres ou de structure afin que, pour une même entrée, la sortie du modèle flou identifié approche suffisamment la sortie du procédé. Dans ce cadre, on suppose que la sortie du procédé à identifier est bornée pour toute entrée admissible.

La procédure d'identification des modèles flous de Takagi-Sugeno à conclusion constante peut être décomposée en une première phase d'identification de structure (nombre de symboles flous et leur répartition sur les univers de discours des entrées) suivie d'une seconde identification de paramètres (conclusions numériques dans les règles floues).

L'objectif de l'identification de structure est de déterminer le nombre de symboles flous utilisés dans les prémisses de règles ainsi que leur répartition sur les univers de discours des entrées. Le choix des nombres de symboles correspondant amène à fixer le nombre de règles du modèle.

Dans la littérature d'identification linéaire [Foul-82][Born-94], pour un procédé linéaire donné, la structure du modèle peut être choisie (modèle AR, ARMA, ARMAX, ...etc.) dans la mesure où l'ordre du système est connu. Le problème d'identification est alors ramené à un problème d'estimation de paramètres. Cette procédure peut être adoptée dans l'identification floue des systèmes non linéaires.

L'identification de paramètres (conclusions numériques des règles) suppose une structure de base de règles figée. Dans ce cas, comme pour le cas linéaire, le problème d'identification est ramené à un problème d'estimation de paramètres.

Dans ce mémoire, on ne traitera pas la première tâche, c'est-à-dire que tout au long de ce travail, nous supposons que la structure du modèle flou (nombre de symboles flous et leur répartition sur les univers de discours des entrées) est disponible ou a été identifiée au préalable.

2.5 CONCLUSION

Dans ce chapitre, nous avons présenté les systèmes flous d'une manière générale tout en donnant les bases théoriques de la logique floue puis on a présenté l'architecture des systèmes flous de Takagi-Sugeno d'ordre un à conclusion constante. Ces derniers jouent un rôle très important dans les stratégies de commande floues qui seront détaillées dans les chapitres suivants.

Dans les chapitres suivant, nous présentons les stratégies de commande qui utilise les systèmes flous de Takagi-Sugeno d'ordre un pour la commande en position et en orientation du robot PUMA 560 étudié en chapitre 1.

CHAPITRE 3 :

COMMANDE ADAPTATIVE FLOUE CENTRALISEE

COMMANDE ADAPTATIVE FLOUE CENTRALISEE

3.1 INTRODUCTION

La commande des robots en position et en orientation constitue à l'heure actuelle l'une des préoccupations majeures des recherches en robotique. En effet la majorité des tâches confiées aux robots exigent une très grande précision sous des trajectoires rapides. Ces tâches sont très délicates vu la complexité de la dynamique résultante, sa non linéarité, ses multiples degrés de liberté, et les incertitudes paramétriques et dynamiques. Dans de pareilles circonstances, il est difficile d'obtenir les performances désirées avec des techniques où l'algorithme de commande est basé seulement sur le modèle dynamique explicite du robot. Une solution pour résoudre ce problème consiste à modéliser le robot par un modèle flou représenté par un nombre fini et raisonnable de règles floues fixé a priori, auquel on attache quelque fois une erreur d'approximation. Cependant, le modèle flou construit off line avec des paramètres fixes ne peut pas faire face au changement des paramètres. Par conséquent, les paramètres doivent être adaptés sans interruption lors du fonctionnement pour compenser cet effet indésirable. D'où l'appellation commande adaptative floue.

Une commande est dite adaptative si elle comporte des paramètres non fixés à l'avance mais modifiés en ligne.

D'une manière générale, les contrôleurs adaptatifs flous sont divisés en deux grandes classes : une s'appelle la commande adaptative floue directe et l'autre la commande adaptative floue indirecte. Dans la commande adaptative floue directe, le système flou est vu comme un contrôleur adaptatif. Cependant, dans la commande adaptative floue indirecte, les systèmes flous sont utilisés pour modéliser le système réel et le contrôleur est construit en supposant que le système flou représente approximativement le vrai système (principe d'équivalence certaine) [Labi-05] [Woo-04] [Chek-03] [Spoo-02].

Dans ce chapitre, nous nous intéressons à l'application de deux techniques de commande adaptatives floue centralisées aux robots manipulateurs. L'approche indirecte est

présentée en premier lieu puis l'approche indirecte. Ces deux approches utilisent le théorème de Lyapunov pour l'obtention des lois d'adaptation des paramètres.

3.2 COMMANDE ADAPTATIVE FLOUE INDIRECTE

Dans la technique adaptative floue indirecte, les systèmes flous sont utilisés pour estimer en premier lieu la dynamique du robot puis l'élaboration de la loi de commande est déduite en considérant les paramètres estimés des systèmes flous comme de vrais paramètres, ces deux étapes se font séquentiellement.

Le modèle dynamique d'un robot manipulateur à n degrés de liberté est donné sous la forme compacte suivante

$$A(q) \ddot{q} + H(q, \dot{q}) = \Gamma \quad (3.1)$$

ou encore

$$\ddot{q} = F(q, \dot{q}) + G(q) \Gamma \quad (3.2)$$

avec $F(q, \dot{q}) = -A(q)^{-1}H(q, \dot{q})$ et $G(q) = A(q)^{-1}$

où $q = [q_1 \ \cdots \ q_n]^T$ et $\dot{q} = [\dot{q}_1 \ \cdots \ \dot{q}_n]^T$ sont les vecteurs position et vitesse articulaire de robot supposés mesurables et $\ddot{q} = [\ddot{q}_1 \ \cdots \ \ddot{q}_n]^T$ vecteur accélération. $F(q, \dot{q}) \in R^{n \times 1}$ regroupe les termes d'inertie, de Coriolis, de centrifuge et de gravité et $G(q) \in R^{n \times n}$ regroupe les termes d'inertie et $\Gamma \in R^{n \times 1}$ est le vecteur des couples appliqués sur chaque articulation.

Le modèle (3.2) peut également s'écrire sous la forme explicite suivante

$$\begin{cases} \ddot{q}_1 = f_1(q, \dot{q}) + \sum_{j=1}^n g_{1j}(q) \Gamma_j \\ \vdots \\ \ddot{q}_n = f_n(q, \dot{q}) + \sum_{j=1}^n g_{nj}(q) \Gamma_j \end{cases} \quad (3.3)$$

On suppose que la fonction $G(q)$ est définie positive et satisfait

$$G(q) \geq \sigma_0 I_p, \quad \sigma_0 > 0.$$

3.2.1 APPROXIMATION PAR LES SYSTEMES FLOUS

Puisque le modèle dynamique du robot est incertain, les éléments des fonctions non linéaires $F(q, \dot{q})$ et $G(q)$, définis dans des ensembles compacts $\Omega_{q, \dot{q}}$ et Ω_q , peuvent être

approximées en utilisant les systèmes flous de Takagi-Sugeno d'ordre un de la manière suivante

$$\hat{f}_i(q, \dot{q}, \hat{\theta}_{f_i}) = w_{f_i}^T(q, \dot{q}) \hat{\theta}_{f_i} \quad (3.4)$$

$$\hat{g}_{ij}(q, \hat{\theta}_{g_{ij}}) = w_{g_{ij}}^T(q) \hat{\theta}_{g_{ij}} \quad (3.5)$$

avec $\hat{\theta}_{f_i}$ et $\hat{\theta}_{g_{ij}}$ sont les paramètres à adapter par un algorithme d'adaptation, w_{f_i} et $w_{g_{ij}}$ sont les fonctions floues de base

Soient

$$\theta_{f_i}^* = \arg \min_{\theta_{f_i}} \left\{ \sup_{q \in \Omega_q} |f_i(q, \dot{q}) - \hat{f}_i(q, \dot{q}, \hat{\theta}_{f_i})| \right\} \quad (3.6)$$

$$\theta_{g_{ij}}^* = \arg \min_{\theta_{g_{ij}}} \left\{ \sup_{q, \dot{q} \in \Omega_{q, \dot{q}}} |g_{ij}(q) - \hat{g}_{ij}(q, \hat{\theta}_{g_{ij}})| \right\} \quad (3.7)$$

où $\theta_{f_i}^*$ et $\theta_{g_{ij}}^*$ sont les paramètres optimaux de $\hat{\theta}_{f_i}$ et $\hat{\theta}_{g_{ij}}$ introduits uniquement pour faire l'étude théorique et $\varepsilon_{f_i}(q, \dot{q})$ et $\varepsilon_{g_{ij}}(q)$ sont les erreurs d'approximation définies par

$$\varepsilon_{f_i}(q, \dot{q}) = f_i(q, \dot{q}) - \hat{f}_i(q, \dot{q}, \theta_{f_i}^*) \quad (3.8)$$

$$\varepsilon_{g_{ij}}(q) = g_{ij}(q) - \hat{g}_{ij}(q, \theta_{g_{ij}}^*) \quad (3.9)$$

tel que $|\varepsilon_{f_i}| \leq \bar{\varepsilon}_{f_i}$ et $|\varepsilon_{g_{ij}}| \leq \bar{\varepsilon}_{g_{ij}}$

où les éléments de $\bar{\varepsilon}_{f_i}$ et $\bar{\varepsilon}_{g_{ij}}$ sont des constantes positives

posons

$$\hat{F}(q, \dot{q}, \hat{\theta}_F) = [\hat{f}_1(q, \dot{q}, \hat{\theta}_{f_1}) \quad \cdots \quad \hat{f}_n(q, \dot{q}, \hat{\theta}_{f_n})]^T \quad (3.10)$$

$$\hat{G}(q, \hat{\theta}_G) = \begin{bmatrix} \hat{g}_{11}(q, \hat{\theta}_{g_{11}}) & \cdots & \hat{g}_{1n}(q, \hat{\theta}_{g_{1n}}) \\ \vdots & \ddots & \vdots \\ \hat{g}_{n1}(q, \hat{\theta}_{g_{n1}}) & \cdots & \hat{g}_{nn}(q, \hat{\theta}_{g_{nn}}) \end{bmatrix} \quad (3.11)$$

$$\varepsilon_F(q, \dot{q}) = [\varepsilon_{f_1}(q, \dot{q}) \quad \cdots \quad \varepsilon_{f_n}(q, \dot{q})]^T \quad (3.12)$$

$$\varepsilon_G(q) = \begin{bmatrix} \varepsilon_{g_{11}}(q) & \cdots & \varepsilon_{g_{1n}}(q) \\ \vdots & \ddots & \vdots \\ \varepsilon_{g_{n1}}(q) & \cdots & \varepsilon_{g_{nn}}(q) \end{bmatrix} \quad (3.13)$$

De l'analyse précédente, il vient

$$F(q, \dot{q}) - \hat{F}(q, \dot{q}, \hat{\theta}_F) = \hat{F}(q, \dot{q}, \theta_F^*) - \hat{F}(q, \dot{q}, \hat{\theta}_F) + \varepsilon_F(q, \dot{q}) \quad (3.14)$$

$$G(q) - \hat{G}(q, \hat{\theta}_G) = \hat{G}(q, \theta_G^*) - \hat{G}(q, \hat{\theta}_G) + \varepsilon_G(q) \quad (3.15)$$

3.2.2 SYNTHÈSE DE LA LOI DE COMMANDE

Avant de développer la loi de commande, on définit les erreurs suivantes :

- Les erreurs de poursuite de chaque articulation

$$\begin{cases} e_1(t) = q_1^d(t) - q_1(t) \\ \vdots \\ e_n(t) = q_n^d(t) - q_n(t) \end{cases} \quad (3.16)$$

$$e(t) = [e_1(t) \quad \cdots \quad e_n(t)]^T$$

où q_i^d est la trajectoire désirée de la $i^{\text{ème}}$ articulation

- Les erreurs de poursuite filtrées

$$\begin{cases} s_1(t) = \left(\frac{d}{dt} + \lambda_1 \right) e_1 \\ \vdots \\ s_n(t) = \left(\frac{d}{dt} + \lambda_n \right) e_n \end{cases} \quad (3.17)$$

$$s(t) = [s_1(t), \dots, s_n(t)]^T$$

- Les dérivées des erreurs de poursuite filtrées

$$\begin{cases} \dot{s}_1 = v_1 - f_1(q, \dot{q}) - \sum_{j=1}^n g_{1j}(q) \Gamma_j \\ \vdots \\ \dot{s}_n = v_n - f_n(q, \dot{q}) - \sum_{j=1}^n g_{nj}(q) \Gamma_j \end{cases} \quad (3.18)$$

$$\text{avec } \begin{cases} v_1 = \ddot{q}_1^d + \lambda_1 \dot{e}_1 \\ \vdots \\ v_n = \ddot{q}_n^d + \lambda_n \dot{e}_n \end{cases}$$

$$v = [v_1(t) \cdots v_n(t)]^T$$

sous forme compacte

$$\dot{s} = v - F(q, \dot{q}) - G(q) \Gamma \quad (3.19)$$

La loi de commande indirecte est généralement donnée sous la forme suivante :

$$\Gamma = \Gamma_C = \hat{G}(q, \hat{\theta}_G)^{-1} \left(-\hat{F}(q, \dot{q}, \hat{\theta}_F) + v + K_0 s \right) \quad (3.20)$$

avec $K_0 = \text{diag}[k_{01} \cdots k_{0n}]$, et $k_{0i} > 0$

Cette approche repose sur l'estimation en ligne des paramètres des systèmes flous et le calcul de la loi de commande se fait par l'utilisation de ces derniers comme des vrais paramètres, sur la base du principe d'équivalence certaine.

Puisque le modèle du robot est considéré inconnu, il est difficile de maintenir les paramètres $\hat{\theta}_G$ dans une région où $\hat{G}(q, \hat{\theta}_G)$ est inversible

Pour remédier à ce problème, l'inverse matriciel $\hat{G}(q, \hat{\theta}_G)^{-1}$ est remplacé par l'inverse régularisé

$$\hat{G}^T(q, \hat{\theta}_G) \left[\varepsilon_0 I_p + \hat{G}(q, \hat{\theta}_G) \hat{G}^T(q, \hat{\theta}_G) \right]^{-1} \quad (3.21)$$

où ε_0 est une constante positive petite

La loi de commande équivalente (3.21) devient :

$$\Gamma_C = \hat{G}^T(q, \hat{\theta}_G) \left[\varepsilon_0 I_p + \hat{G}(q, \hat{\theta}_G) \hat{G}^T(q, \hat{\theta}_G) \right]^{-1} \left(-\hat{F}(q, \dot{q}, \hat{\theta}_F) + v + K_0 s \right) \quad (3.22)$$

Afin de compenser les erreurs introduite par l'approximation des fonctions $F(q, \dot{q})$ et $G(q)$ d'un coté et l'utilisation de l'inverse régularisé à la place de l'inverse matriciel et dans le but d'avoir une loi de commande ne dépendant d'aucune phase d'initialisation, on utilise une loi de commande comportant deux termes :

$$\Gamma = \Gamma_C + \Gamma_r \quad (3.23)$$

Γ_C : est la loi de commande équivalente modifiée donnée par :

$$\Gamma_C = \hat{G}^T(q, \hat{\theta}_G) \left[\varepsilon_0 I_P + \hat{G}(q, \hat{\theta}_G) \hat{G}^T(q, \hat{\theta}_G) \right]^{-1} \left(-\hat{F}(q, \dot{q}, \theta_F) + v + K_0 s + K_1 \hat{G}(q, \hat{\theta}_G) s \right) \quad (3.24)$$

où Γ_r est le terme de robustesse, utilisé pour compenser les erreurs de reconstruction. Il est donné par :

$$\Gamma_r = \frac{s |s^T| \left(\left| \hat{\varepsilon}_F + \hat{\varepsilon}_G \Gamma_C \right| + \left| \Gamma_0 \right| \right)}{\sigma_0 \|s\|^2 + \delta} \quad (3.25)$$

avec

$$\Gamma_0 = \varepsilon_0 \left[\varepsilon_0 I_P + \hat{G}(q, \hat{\theta}_G) \hat{G}^T(q, \hat{\theta}_G) \right]^{-1} \left(-\hat{F}(q, \dot{q}, \hat{\theta}_F) + v + K_0 s + K_1 \hat{G}(q, \hat{\theta}_G) s \right)$$

$\hat{\varepsilon}_F$ et $\hat{\varepsilon}_G$ sont les estimés de $\bar{\varepsilon}_F$ et $\bar{\varepsilon}_G$, $k_1 > 0$ et δ est un paramètre adaptatif défini ci-dessous

En utilisant le théorème de stabilité de Lyapunov, les paramètres du contrôleur flou sont adaptés par les lois d'adaptations suivantes

$$\dot{\hat{\theta}}_{fi} = -\eta_{fi} w_{fi}(q, \dot{q}) s_i \quad (3.26)$$

$$\dot{\hat{\theta}}_{gij} = -\eta_{gij} w_{gij}(q) s_i \left(\Gamma_{Cj} - k_1 s_j \right) \quad (3.27)$$

$$\dot{\hat{\varepsilon}}_F = \gamma_F |s| \quad (3.28)$$

$$\dot{\hat{\varepsilon}}_G = \gamma_G |s| \left| \Gamma_C^T \right| \quad (3.29)$$

$$\dot{\delta} = -\eta_0 \frac{|s^T| \left(\left| \hat{\varepsilon}_F + \hat{\varepsilon}_G \Gamma_C \right| + \left| \Gamma_0 \right| \right)}{\sigma_0 \|s\|^2 + \delta} \quad (3.30)$$

avec $\eta_{fi} > 0$, $\eta_{gij} > 0$, $\gamma_F > 0$, $\gamma_G > 0$, $\delta(0) > 0$ et $\eta_0 > 0$.

3.2.3 ANALYSE DE LA STABILITE

La théorie d'approximation universelle garantit la possibilité de modéliser la dynamique des systèmes non linéaires par des systèmes flous. Ces derniers sont utilisés pour concevoir des contrôleurs flous. Reste qu'une des propriétés fondamentales de la commande, à savoir la stabilité, doit être assurée.

L'objectif est d'assurer la stabilité des structures de commande dans le sens où tous les signaux d'entrée et de sortie restent bornés. Pour analyser la stabilité de la loi de commande (3.23), on utilise le théorème de stabilité de Lyapunov. Celui-ci est le plus utilisé dans la littérature de la commande floue.

D'une manière générale, la synthèse de Lyapunov consiste à sélectionner la candidature d'une fonction positive V puis à choisir des lois de commande ou d'adaptation assurant sa décroissance. De plus, la méthode de Lyapunov peut servir de base à la synthèse des lois de commande stabilisantes.

Des équations (3.21) et (3.23) on a

$$\dot{s} = v - F(q, \dot{q}) - (G(q) - \hat{G}(q, \hat{\theta}_G))\Gamma_C - \hat{G}(q, \hat{\theta}_G)\Gamma_C - G(q)\Gamma_r \quad (3.31)$$

par substitution (3.24) dans (3.31), on aura

$$\dot{s} = -k_0 s - k_1 \hat{G}(q, \hat{\theta}_G) s - (F(q, \dot{q}) - \hat{F}(q, \dot{q}, \hat{\theta}_F)) - (G(q) - \hat{G}(q, \hat{\theta}_G))\Gamma_C + \Gamma_0 - G(q)\Gamma_r \quad (3.32)$$

en utilisant (3.14) et (3.15) la relation (3.32) devient

$$\begin{aligned} \dot{s} = & -k_0 s - k_1 \hat{G}(q, \hat{\theta}_G) s - (\hat{F}(q, \dot{q}, \theta_F^*) - \hat{F}(q, \dot{q}, \hat{\theta}_F)) - (\hat{G}(q, \theta_G^*) - \hat{G}(q, \hat{\theta}_G))\Gamma_C \\ & - G(q)\Gamma_r + \Gamma_0 - \varepsilon_F(q, \dot{q}) - \varepsilon_G(q)\Gamma_C \end{aligned} \quad (3.33)$$

en ajoutant et en retranchant $k_1 \hat{G}(q, \theta_G^*) s$

$$\begin{aligned} \dot{s} = & -k_0 s - k_1 \hat{G}(q, \theta_G^*) s - (\hat{F}(q, \dot{q}, \theta_F^*) - \hat{F}(q, \dot{q}, \hat{\theta}_F)) - (\hat{G}(q, \theta_G^*) - \hat{G}(q, \hat{\theta}_G))(\Gamma_C - k_1 s) \\ & - G(q)\Gamma_r + H_0 - \varepsilon_F(q, \dot{q}) - \varepsilon_G(q)\Gamma_C \end{aligned} \quad (3.34)$$

en multipliant par s^T

$$\begin{aligned} s^T \dot{s} = & -s^T k_0 s - \sum_{i=1}^n w_{fi}^T(q, \dot{q}) \tilde{\theta}_{fi} s_i - \sum_{i=1}^n \sum_{j=1}^n w_{gij}^T(q) \tilde{\theta}_{gij} s_i (\Gamma_{Cj} - k_1 s_j) - s^T k_1 \hat{G}(q, \theta_G^*) s \\ & - s^T G(q)\Gamma_r + s^T \Gamma_0 - s^T \varepsilon_F(q, \dot{q}) - s^T \varepsilon_G(q)\Gamma_C \end{aligned} \quad (3.35)$$

Afin de démontrer la stabilité du système, nous considérons la fonction de Lyapunov candidate suivante

$$V = \frac{1}{2} s^T s + \frac{1}{2} \sum_{i=1}^n \frac{1}{\eta_{fi}} \tilde{\theta}_{fi}^T \tilde{\theta}_{fi} + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{1}{\eta_{gij}} \tilde{\theta}_{gij}^T \tilde{\theta}_{gij} + \frac{1}{2\gamma_F} \tilde{\varepsilon}_f^T \tilde{\varepsilon}_f + \frac{1}{2\gamma_G} \text{tr}(\tilde{\varepsilon}_G^T \tilde{\varepsilon}_G) + \frac{1}{2\eta_0} \delta^2 \quad (3.36)$$

avec $\tilde{\varepsilon}_F = \bar{\varepsilon}_F - \hat{\varepsilon}_F$, $\tilde{\varepsilon}_G = \bar{\varepsilon}_G - \hat{\varepsilon}_G$, $\tilde{\theta}_{fi} = \theta_{fi}^* - \hat{\theta}_{fi}$ et $\tilde{\theta}_{gij} = \theta_{gij}^* - \hat{\theta}_{gij}$

La dérivée temporelle de V est

$$\dot{V} = \mathbf{s}^T \dot{\mathbf{s}} - \sum_{i=1}^n \frac{1}{\eta_{fi}} \tilde{\theta}_{fi}^T \dot{\theta}_{fi} - \sum_{i=1}^n \sum_{j=1}^n \frac{1}{\eta_{gij}} \tilde{\theta}_{gij}^T \dot{\theta}_{gij} - \frac{1}{\gamma_F} \tilde{\boldsymbol{\varepsilon}}_f^T \dot{\boldsymbol{\varepsilon}}_f - \frac{1}{\gamma_G} \text{tr}(\tilde{\boldsymbol{\varepsilon}}_G^T \dot{\boldsymbol{\varepsilon}}_G) + \frac{1}{\eta_0} \delta \dot{\delta} \quad (3.37)$$

En utilisant (3.35), \dot{V} peut être bornée comme suit

$$\dot{V} \leq -\mathbf{s}^T k_0 \mathbf{s} - k_1 \mathbf{s}^T \hat{G}(q, \theta_G^*) \mathbf{s} + \dot{V}_1 + \dot{V}_2 \quad (3.38)$$

avec

$$\dot{V}_1 = -\sum_{i=1}^n \tilde{\theta}_{fi}^T \left(w_{fi}(q, \dot{q}) s_i + \frac{1}{\eta_{fi}} \dot{\theta}_{fi} \right) - \sum_{i=1}^n \sum_{j=1}^n \tilde{\theta}_{gij}^T(q) \left(w_{gij}(q) s_i (\Gamma_{cj} - k_1 s_j) + \frac{1}{\eta_{gij}} \dot{\theta}_{gij} \right) \quad (3.39)$$

$$\dot{V}_2 = -\mathbf{s}^T \hat{G}(q) \Gamma_r + |\mathbf{s}^T \Gamma_0| + |\mathbf{s}^T \bar{\boldsymbol{\varepsilon}}_F| + |\mathbf{s}^T \bar{\boldsymbol{\varepsilon}}_G| |\Gamma_c| - \frac{1}{\gamma_F} \tilde{\boldsymbol{\varepsilon}}_F^T \dot{\boldsymbol{\varepsilon}}_F - \frac{1}{\gamma_G} \text{tr}(\tilde{\boldsymbol{\varepsilon}}_G^T \dot{\boldsymbol{\varepsilon}}_G) + \frac{1}{\eta_0} \delta \dot{\delta} \quad (3.40)$$

A partir des lois d'adaptations (3.26) et (3.27), on a

$$\dot{V}_1 = 0 \quad (3.41)$$

et a partir de (3.25), (3.28), (3.29) et (3.30) on a

$$\dot{V}_2 \leq 0 \quad (3.42)$$

Pour le modèle du robot, il est connu que la matrice $G(q)$ est définie positive, alors on peut supposer que $\hat{G}(q, \theta_G^*)$ est, au moins, semi définie positive, de ce fait, (3.39) peut être simplifiée à

$$\dot{V} \leq -\mathbf{s} K_0 \mathbf{s} \quad (3.43)$$

D'où la bornitude de tous les signaux en boucle fermée et la convergence vers zéro de toutes les erreurs de poursuite

3.2.4 RESULTATS DE SIMULATION

Dans cette partie nous procédons à l'application de la commande adaptative floue indirecte, décrite précédemment, pour la commande de la position et de l'orientation du robot PUMA 560 à six degrés de liberté.

Dans ce cas les vecteurs d'état q et $[q \ \dot{q}]^T$ sont définis comme suit :

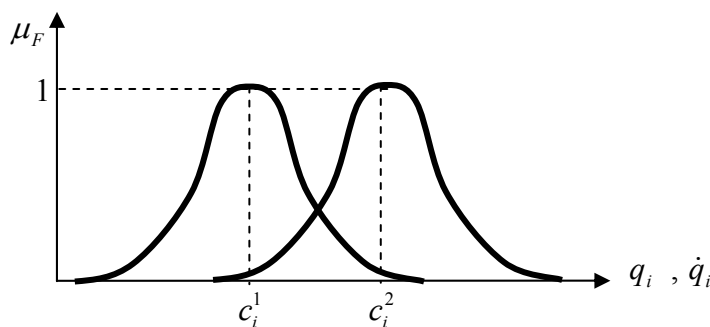
$$q = [q_1 \ \cdots \ q_6]^T \quad \text{et} \quad [q \ \dot{q}]^T = [q_1 \ \cdots \ q_6 \ \dot{q}_1 \ \cdots \ \dot{q}_6]^T$$

Etant donné que les éléments de $F(q,\dot{q})$ et $G(q)$ sont incertains, six (6) systèmes flous sont utilisés pour approcher les éléments de $F(q,\dot{q})$, et trente six (36) systèmes flous sont utilisés pour approcher les éléments de $G(q)$.

Dans le soucis de réduire le nombre de règles dans la construction du système flou, la structure du modèle dynamique du robot est supposée connue [Arms-86], seuls les paramètres du modèle sont inconnus. Dans ce cas, tous les systèmes flous utilisés pour estimer $F(q,\dot{q})$ admettent les variables $q_2, q_3, q_4, q_5, \dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4$ et \dot{q}_5 comme entrées et tous les systèmes flous utilisés pour estimer $G(q)$ ont les variables q_2, q_3, q_4, q_5 comme entrées. Afin de réduire le nombre de règles floues, qui peut être élevé pour un robot à six degrés de liberté, chaque variable d'entrée est définie uniquement par deux (2) fonctions d'appartenance gaussiennes de la forme :

$$\mu_{F_i^j}(q_i) = \exp\left(-0.5\left(\frac{q_i - c_i^j}{\sigma_i}\right)^2\right) \text{ où } i=2,\dots,5 \text{ et } j=1,2$$

$$\mu_{F_i^j}(\dot{q}_i) = \exp\left(-0.5\left(\frac{\dot{q}_i - \bar{c}_i^j}{\bar{\sigma}_i}\right)^2\right) \text{ où } i=1,\dots,6 \text{ et } j=1,2$$



où valeurs des paramètres $\sigma_i, \bar{\sigma}_i, c_i^j$ et \bar{c}_i^j sont choisies selon le tableau 3.1

q_i	c_i^1	c_i^2	σ_i	\dot{q}_i	\bar{c}_i^1	\bar{c}_i^2	$\bar{\sigma}_i$
				\dot{q}_1	-0.26	0.15	0.1
q_2	0.18	1	0.205	\dot{q}_2	-0.32	0.21	0.13
q_3	-0.2	1.2	0.35	\dot{q}_3	-0.42	0.58	0.25
q_4	-1.75	4.4	1.28	\dot{q}_4	-0.1	2	0.52
q_5	0.72	2.95	0.55	\dot{q}_5	-0.75	0.75	0.375

Tab.3.1 :Les paramètres des fonctions d'appartenance

Les coefficients de commande et d'adaptation sont donnés comme suit

$$K_0 = \text{diag}[220 \ 220 \ 220 \ 100 \ 100 \ 100], \quad \gamma_f = \gamma_g = 0.001, \quad \lambda_1 = \lambda_2 = \lambda_3 = 1600, \quad \lambda_4 = \lambda_5 = \lambda_6 = 1000, \\ k_1 = 100, \quad \eta_{fi} = 0.5, \quad \eta_{gij} = 0.005, \quad \sigma_0 = 0.001, \quad \varepsilon_0 = 0.1 \text{ et } \eta_0 = 0.001$$

Les résultats de simulation sont illustrés par les figures 3.1-3.4

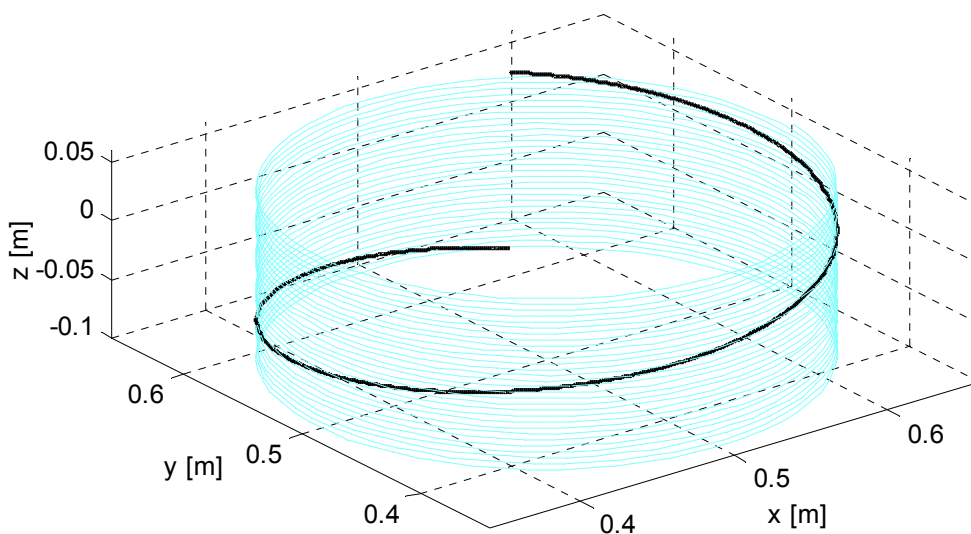


Fig.3.1.a Trajectoire cartésienne de l'outil par rapport au repère R_0

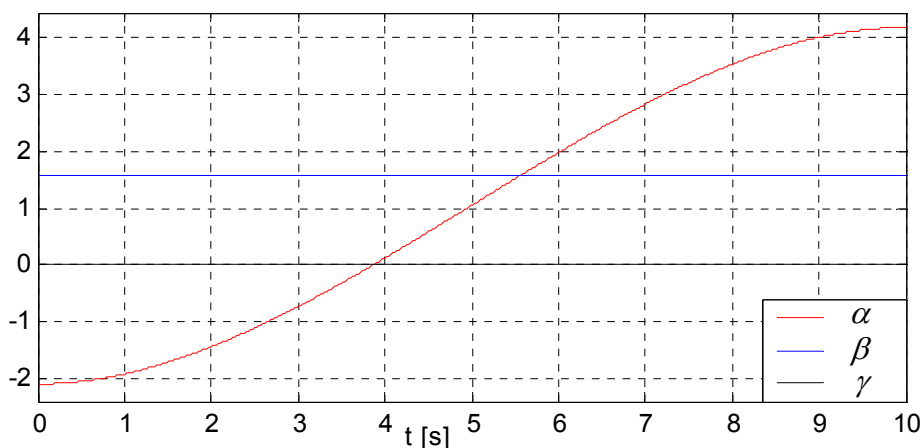


Fig.3.1.b Orientation de l'outil (unité : radian)

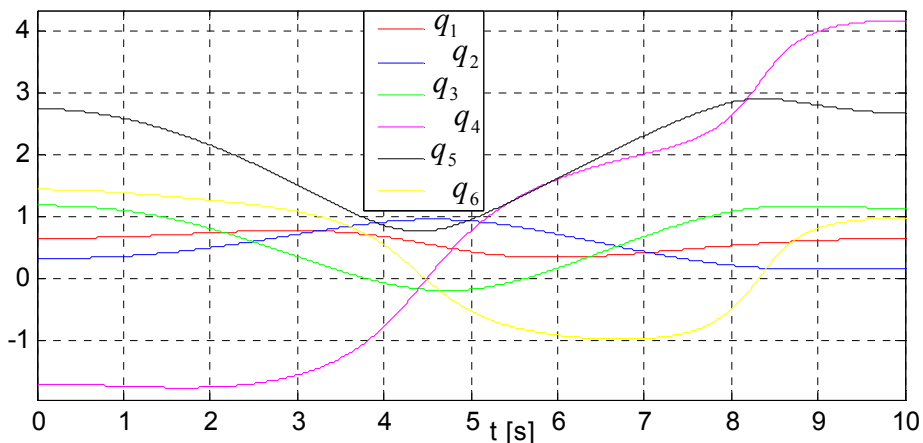


Fig.3.2 Trajectoires articulaires (unité : radian)

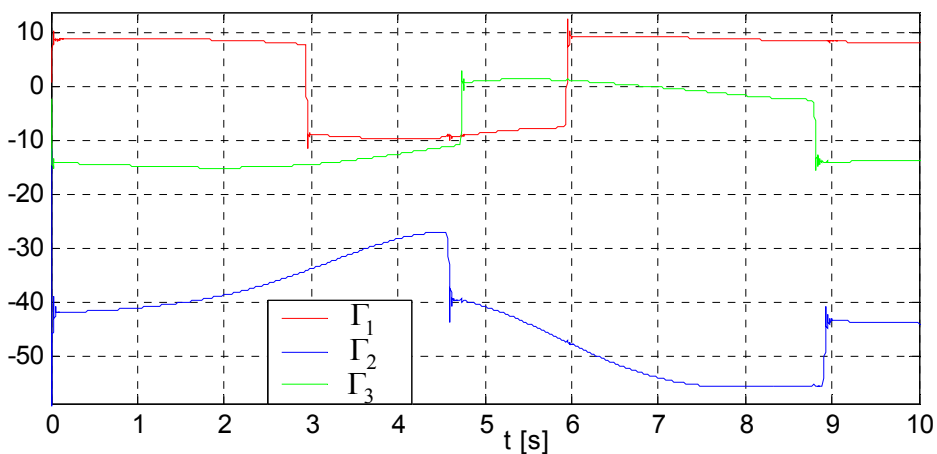


Fig.3.3.a Couples des trois premiers actionneurs (unité : N.m)

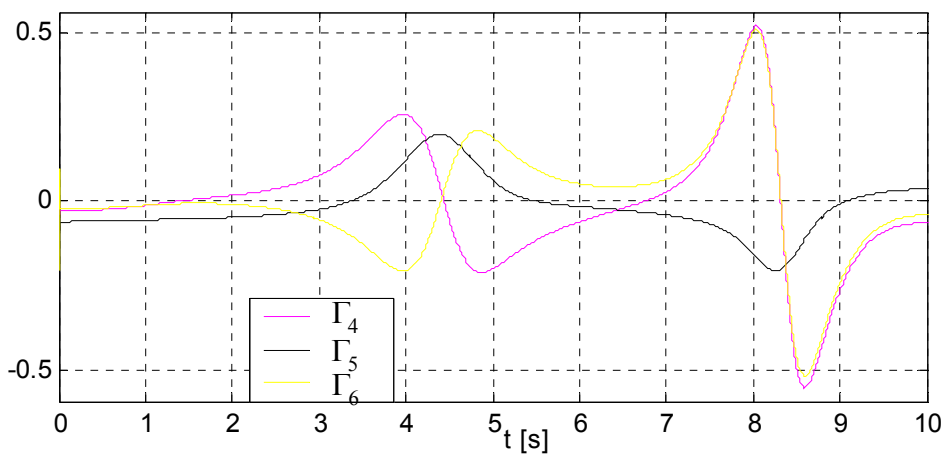


Fig.3.3.b Couples des trois derniers actionneurs (unité : N.m)

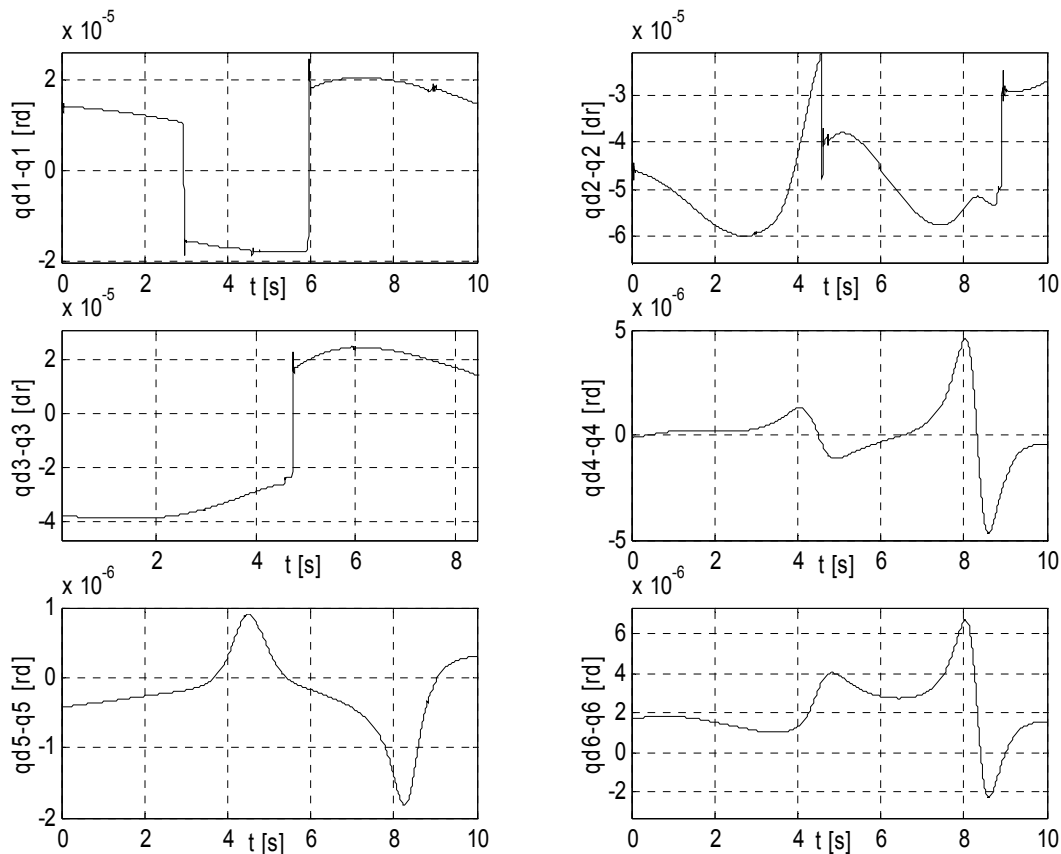


Fig.3.4.a Erreurs articulaires

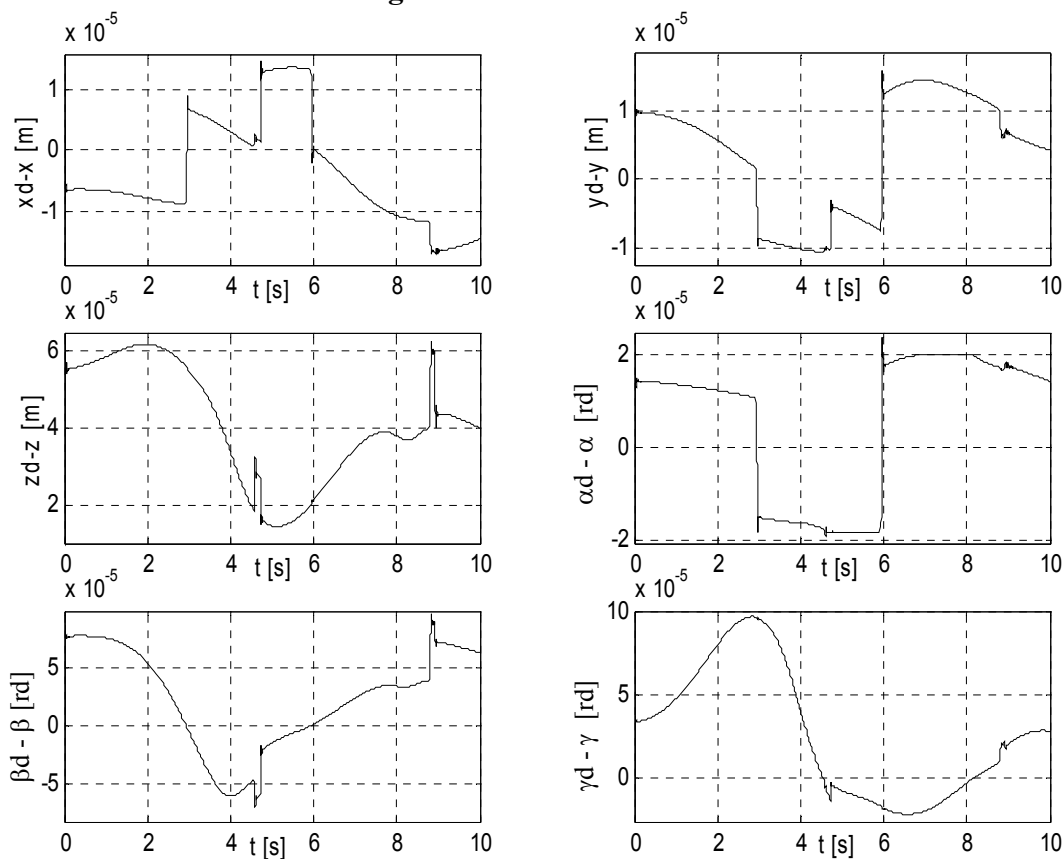


Fig.3.4.b Image des erreurs articulaires dans l'espace cartésien

▪ Robustesse vis-à-vis des perturbations extérieures

Les figures 3.5-3.3.7 montrent les résultats de simulation en appliquant au robot une charge de 2kg (masse de l'outil) à partir de l'instant $t = 5$ secondes.

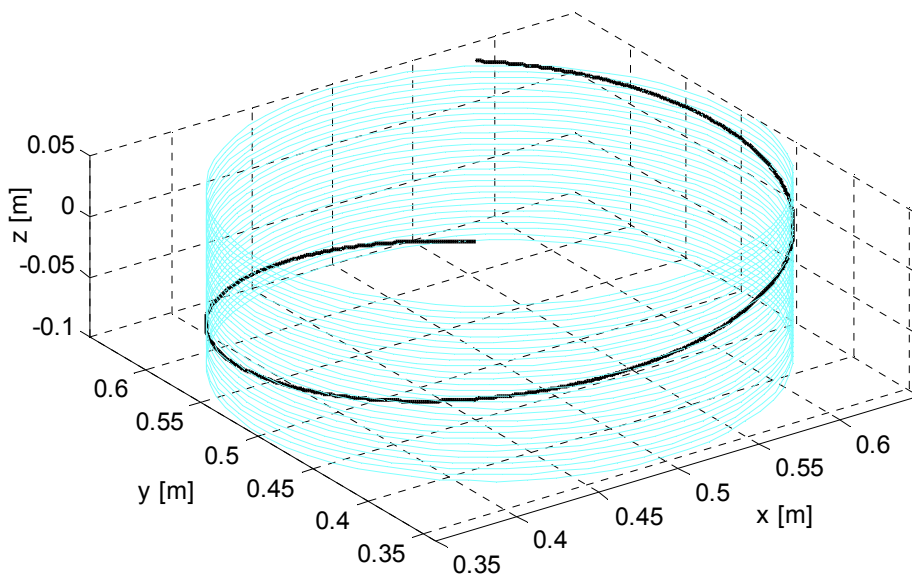


Fig.3.5.a Trajectoire cartésienne de l'outil

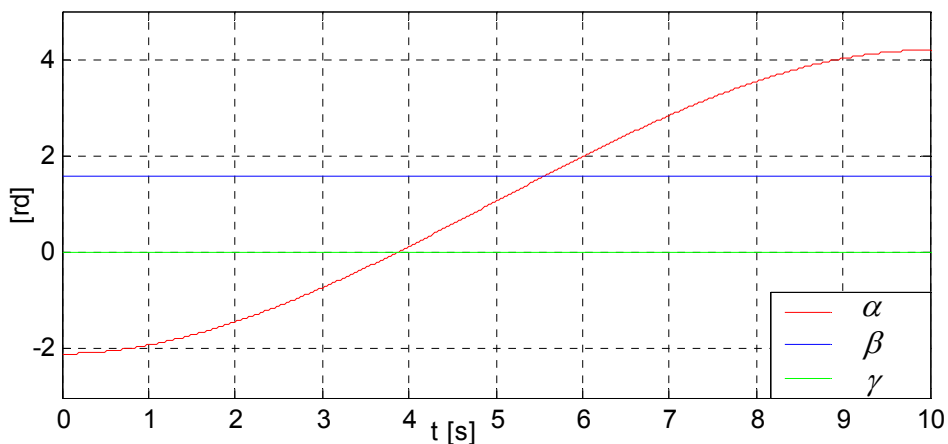


Fig.3.5.b Orientation de l'outil (unité : radian)

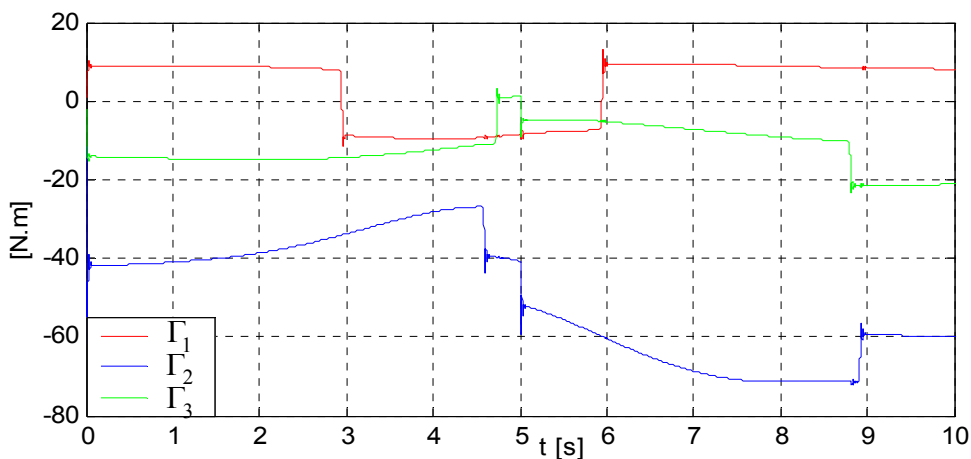


Fig.3.6.a Couples des trois premiers actionneurs (unité : N.m)

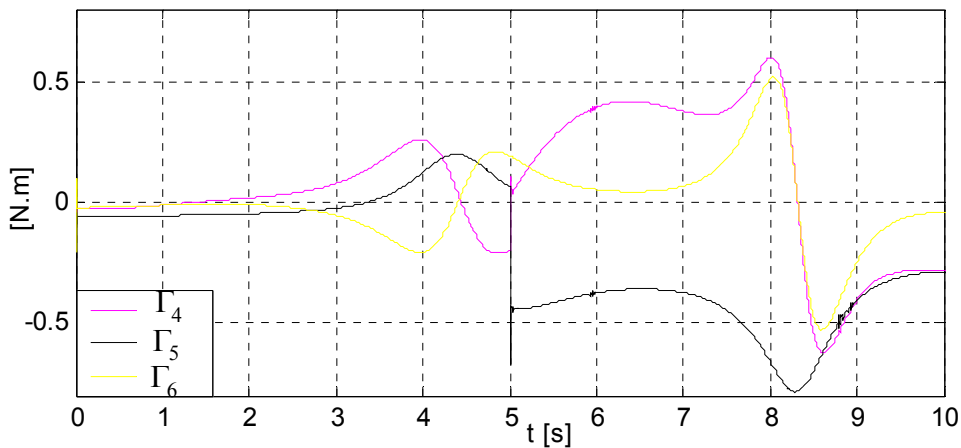


Fig.3.6.b Couples des trois derniers actionneurs (unité : N.m)

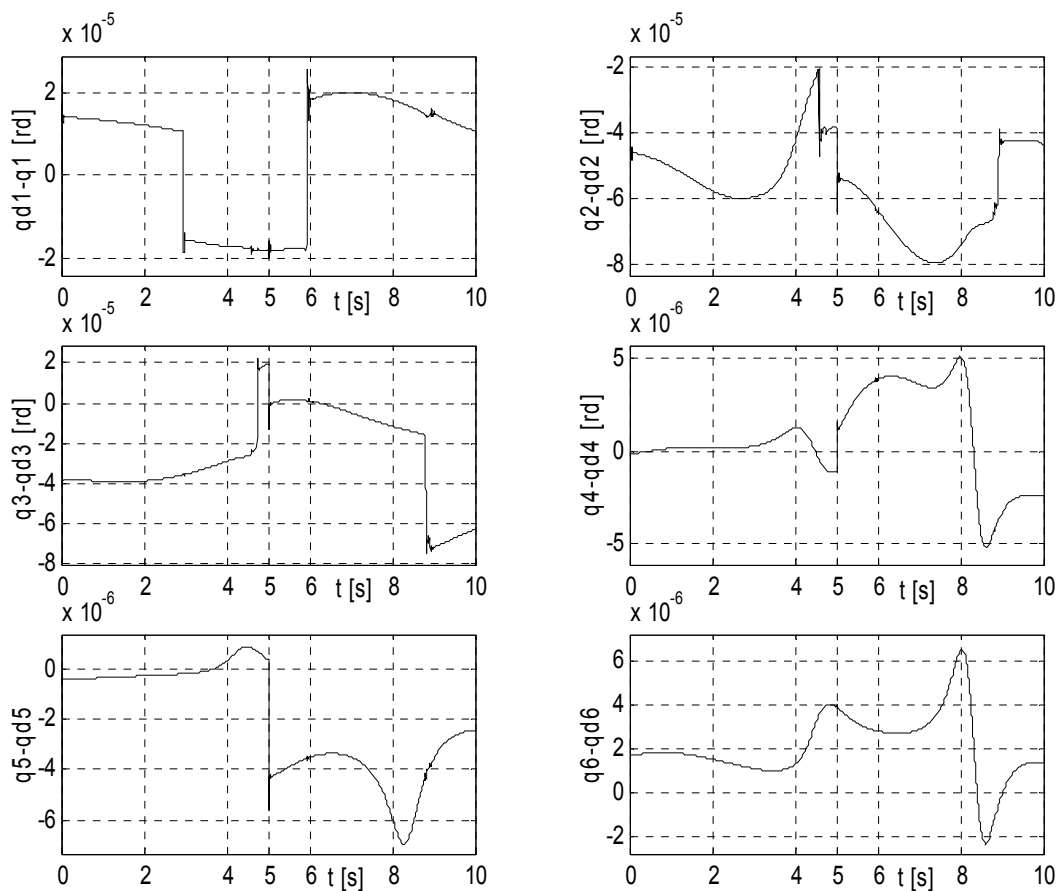


Fig.3.7.a Erreurs articulaires

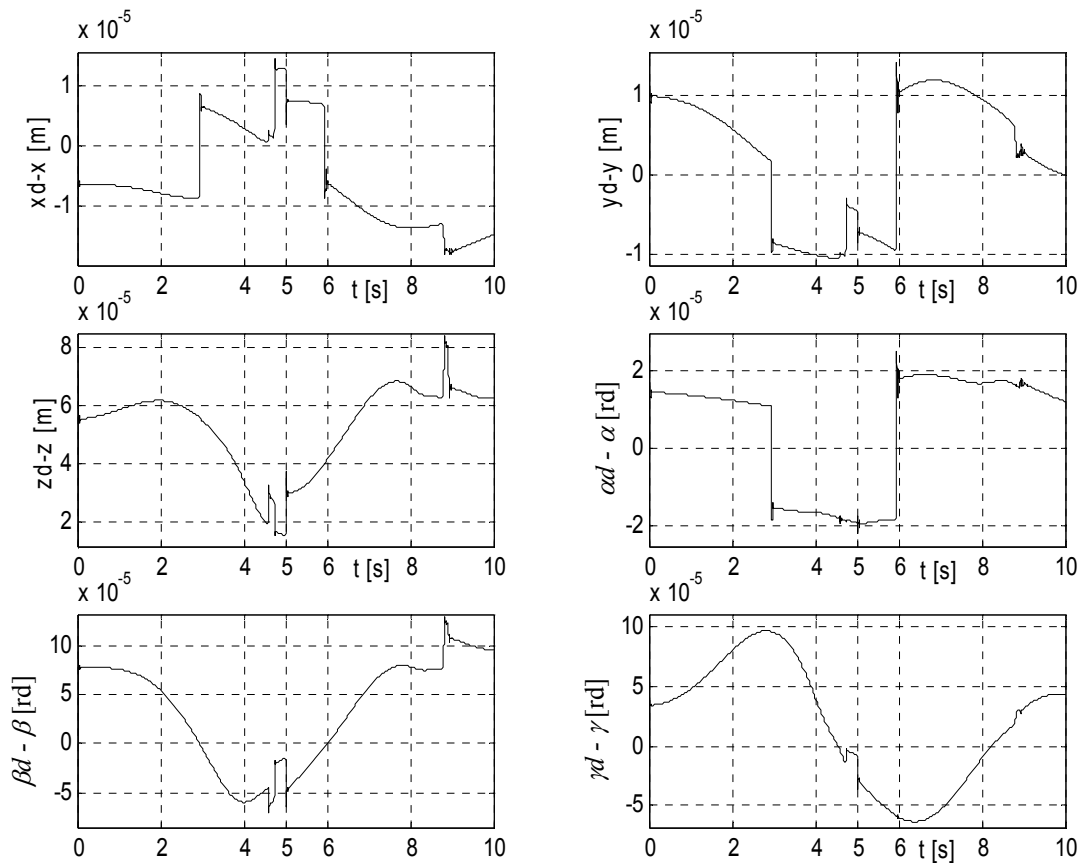


Fig.3.7.b Image des erreurs articulaires dans l'espace cartésien

▪ Robustesse vis-à-vis des variations paramétriques

Les figures 3.8-3.10 montrent les résultats de simulation en soumettant le robot à une variation de 100% de la matrice d'inertie.

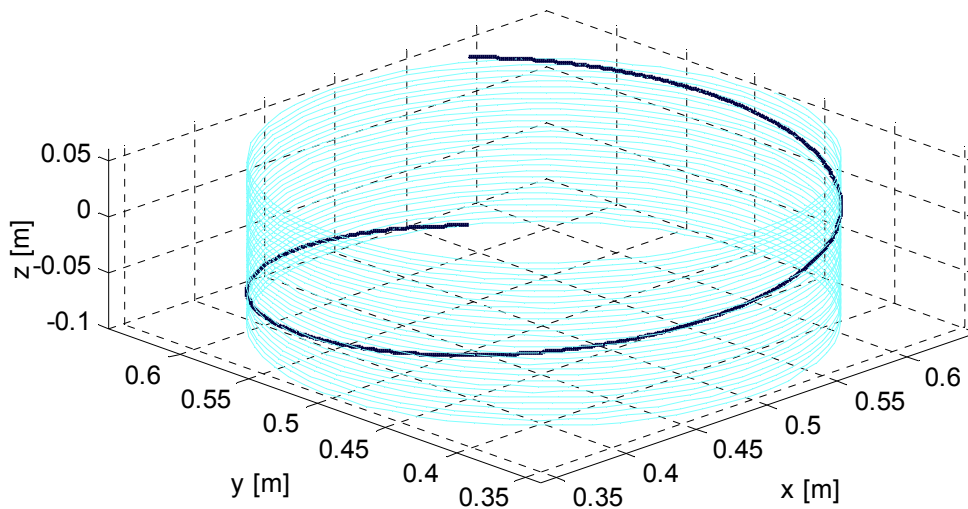


Fig.3.8.a Trajectoire de l'outil

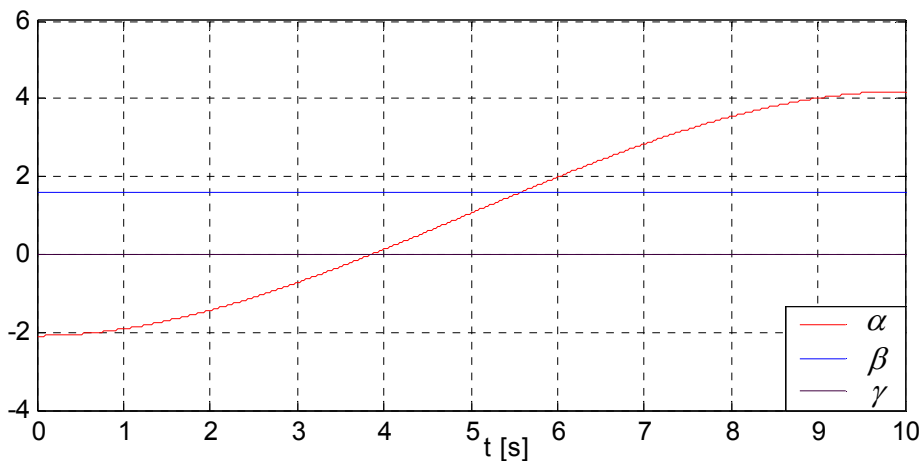


Fig.3.8.b Orientation de l'outil (unité : radian)

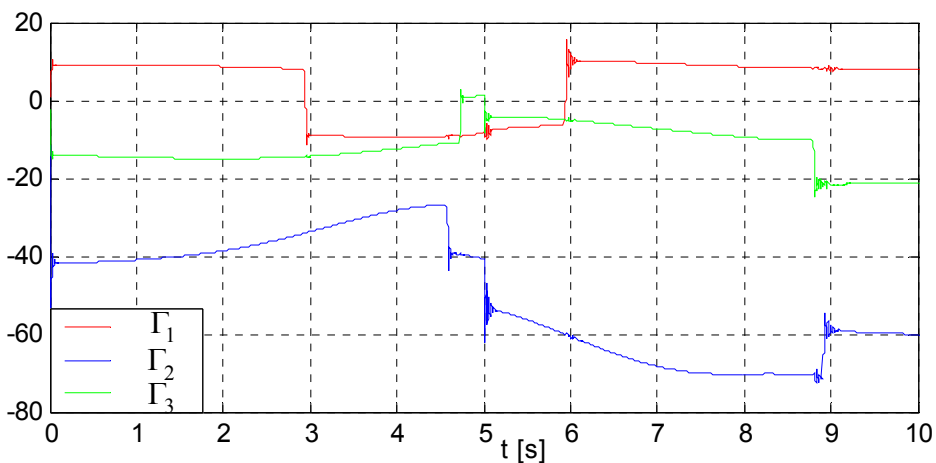


Fig.3.9.a Couples actionneurs des trois premières articulations (unité : N.m)

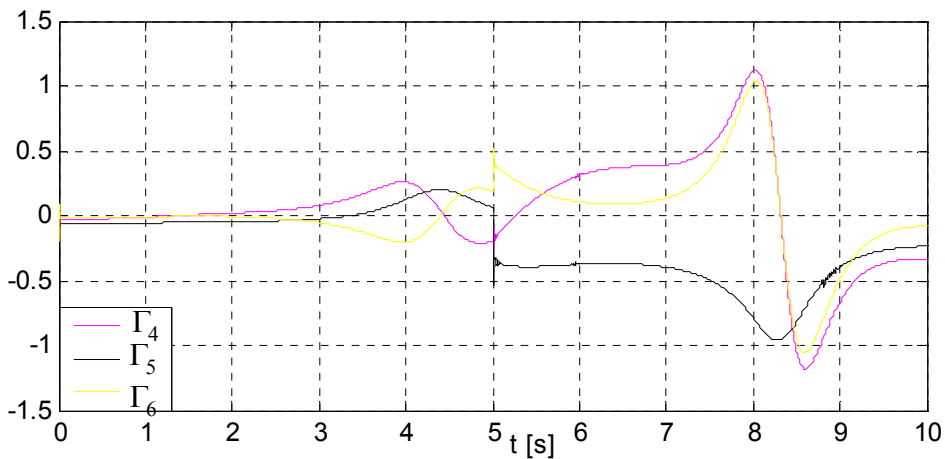


Fig.3.9.b Couples actionneurs des trois dernières articulations (unité : N.m)

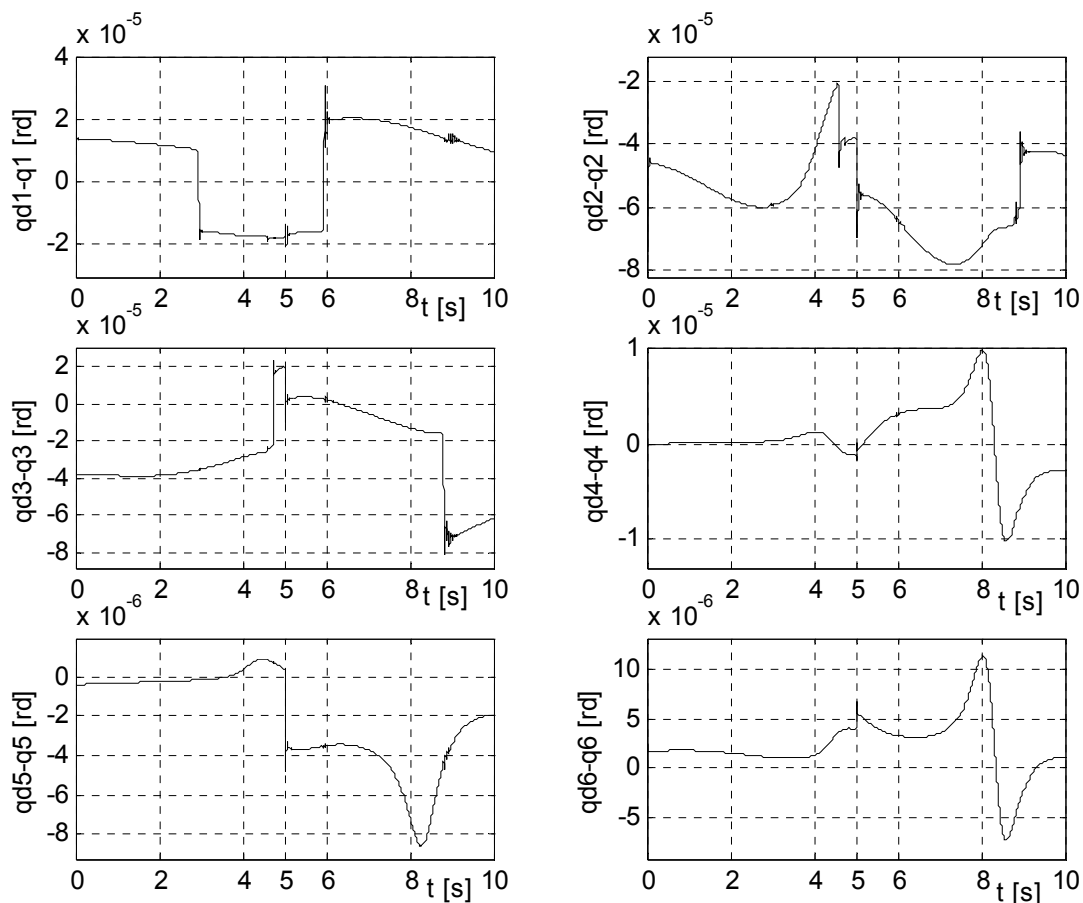


Fig.3.10.a Erreurs articulaires

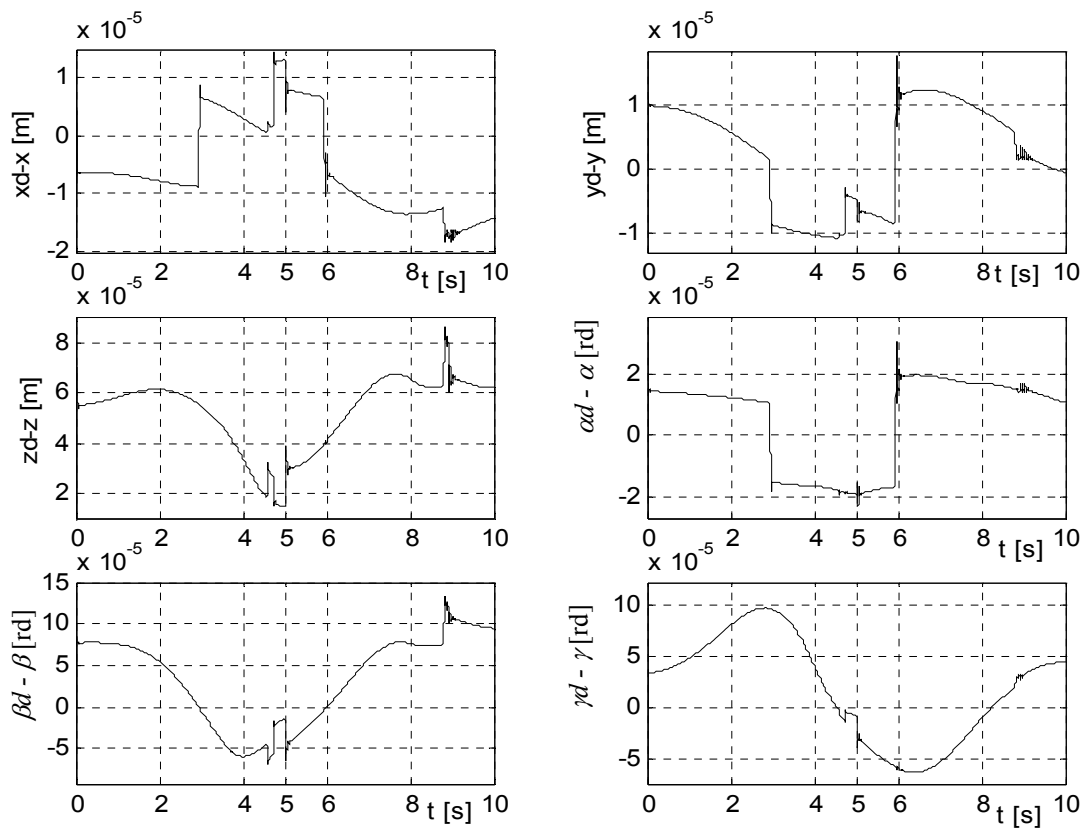


Fig.3.10.b Image des erreurs dans l'espace cartésien

Les tableaux 3.2 et 3.4 présentent respectivement les erreurs cartésiennes maximales et les couples articulaires maximaux pour différents tests

	Tests sans perturbations	Tests avec charge	Tests avec perturbations paramétriques
$x^d - x [mm]$	0.01	0.01	0.01
$y^d - y [mm]$	0.01	0.01	0.01
$z^d - z [mm]$	0.07	0.08	0.08
$\alpha^d - \alpha [rd]$	2.39×10^{-5}	2.5×10^{-5}	3.04×10^{-5}
$\beta^d - \beta [rd]$	9.68×10^{-5}	6.96×10^{-5}	1.35×10^{-4}
$\gamma^d - \gamma [rd]$	9.72×10^{-5}	9.72×10^{-5}	9.72×10^{-5}

Tab.3.2 Erreurs maximales dans l'espace de la tâche

	Tests sans perturbations	Tests avec charge	Tests avec perturbations paramétriques
$\Gamma_1 [N.m]$	12.68	13.15	15.88
$\Gamma_2 [N.m]$	61.96	72.27	72.57
$\Gamma_3 [N.m]$	15.51	23.24	24.63
$\Gamma_4 [N.m]$	0.55	0.60	1.17
$\Gamma_5 [N.m]$	0.20	0.78	0.96
$\Gamma_6 [N.m]$	0.51	0.52	1.05

Tab.3.3 Couples maximaux développés par les actionneurs

Les figures 3.1.a et 3.1.b montrent respectivement la poursuite en position et en orientation de l'outil lors de l'exécution de la tâche de soudage de forme spirale sur un cylindre.

Les trajectoires des six axes du robot (fig.3.2) lors de l'exécution de la tâche sont lisses et continues.

Les changements brutaux des amplitudes des couples actionneurs (fig.3.3.a, 3.6.a et 3.9.a) sont dûs aux frottements qui sont des fonctions non linéaires discontinus dépendant du signe de la vitesse de chaque articulation [Annexe 1].

Les oscillations des amplitudes des couples apparues à l'instant $t= 5$ secondes (fig.3.6.a, 3.6.b, 3.9.a et 3.9.b) sont dues aux perturbations. Ces dernières conduisent à des variations du modèle flou. Dans ce cas l'algorithme d'adaptation recherche un nouveau jeu de paramètres retrouvés, de manière à ce que l'outil suit sa trajectoire.

Les performances de la loi de commande se traduisent par des erreurs de suivi des trajectoires dans l'espace articulaire (fig.3.4.a, 3.7.a et 3.10.a) et leurs images respectives dans l'espace cartésien (erreurs de position/orientation : fig.3.4.b, 3.7.b et 3.10.b) .

Les tableaux 3.2 et 3.3 résument respectivement les erreurs maximales de position/orientation du robot et les couples maximaux développés par chaque actionneurs pour trois différents tests : sans perturbation, avec application d'une charge de 2kg à l'instant $t=5$ secondes et pour une variation de 100% de la matrice d'inertie à l'instant $t=5$ secondes. Cet instant est choisi tel qu'il correspond au passage des vitesses cartésiennes par leurs maximums. Ainsi, les faibles valeurs des erreurs enregistrées révèlent que cette loi de commande est robuste vis-à-vis des perturbations extérieures et paramétriques. De plus, il est bien clair que les couples développés lors de différents tests restent dans les limites admissibles.

3.3 COMMANDE ADAPTATIVE FLOUE DIRECTE

Contrairement à l'approche indirecte où l'estimation de la dynamique du robot par les systèmes flous et le calcul de la loi commande se font séquentiellement, l'approche directe utilise les systèmes flous pour modifier directement l'ensemble des paramètres du régulateur pour former un système stable en boucle fermée.

Dans cette partie, on présente une technique adaptative directe pour une classe de systèmes souvent rencontrée en robotique. Cette classe de système est donnée sous la forme suivante

$$\begin{cases} u = F(X)x^{(p)} + G(X) \\ y = x \end{cases} \quad (3.44)$$

où $x = [x_1, \dots, x_n]^T$ et $x^{(p)}$ est la $p^{\text{ième}}$ dérivée de la variable x , de plus les vecteurs $X = [x^{(p-1)^T}, \dots, x^T]^T$, $u = [u_1 \dots, u_n]^T$ et $y = [y_1, \dots, y_n]^T$ sont respectivement le vecteur d'état, le vecteur de commande et le vecteur de sortie. $F(X) \in R^{n.n}$ et $G(X) \in R^{n.1}$ sont des fonctions non linéaires.

Nous supposons que $F(X) \in R^{n.n}$ est une matrice définie positive, telle que sa dérivée par rapport au temps satisfait la condition

$$\|\dot{F}(X)\| \leq F_0 \|X\|$$

avec $F_0 > 0$.

3.3.1 APPROXIMATION PAR LES SYSTEMES FLOUS

Les fonctions non linéaires $F(X)$ et $G(X)$ peuvent être approximées en utilisant les systèmes flous de Takagi-Sugeno d'ordre un

$$F(X) = W^T(X)\theta_F^* + \varepsilon_F(X). \quad (3.45)$$

$$G(X) = W^T(X)\theta_G^* + \varepsilon_G(X). \quad (3.46)$$

avec θ_F^* et θ_G^* sont les paramètres optimaux et $\varepsilon_F(X)$ et $\varepsilon_G(X)$ sont les erreurs de reconstruction des fonctions $F(X)$ et $G(X)$ tel que :

$$|\bar{\varepsilon}_F| \leq \bar{\varepsilon}_F \quad \text{et} \quad |\bar{\varepsilon}_G| \leq \bar{\varepsilon}_G$$

W^T , θ_F^* et θ_G^* sont organisés comme suit

$$W^T = \begin{bmatrix} w^T & 0 & \cdots & 0 \\ 0 & w^T & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & w^T \end{bmatrix} \quad (3.47)$$

$$\theta_F^* = \begin{bmatrix} \theta_{f11}^* & \theta_{f12}^* & \cdots & \theta_{f1n}^* \\ \theta_{f21}^* & \theta_{f22}^* & \cdots & \theta_{f2n}^* \\ \vdots & \vdots & \cdots & \vdots \\ \theta_{fn1}^* & \theta_{fn2}^* & \cdots & \theta_{fnn}^* \end{bmatrix},$$

$$\theta_G^* = \left[(\theta_{g_1}^*)^T \quad (\theta_{g_2}^*)^T \quad \cdots \quad (\theta_{g_n}^*)^T \right]^T \quad (3.48)$$

3.3.2 SYNTHÈSE DE LA LOI DE COMMANDE

L'objectif de cette loi de commande est de forcer la sortie du système y à suivre la trajectoire de référence y_d .

Avant de donner la loi de commande, on définit les grandeurs suivantes :

- Le vecteur de l'erreur de poursuite

$$s(t) = [s_1(t), \dots, s_n(t)]^T \quad (3.49)$$

$$s_i = \left(\frac{\partial}{\partial t} + \lambda_i \right)^{(p-1)} \cdot e_i \quad (3.50)$$

$$= \lambda_i^{(p-1)} e_i + (p-1)\lambda_i^{(p-2)} \dot{e}_i + \cdots + (p-1)\lambda_i e_i^{(p-2)} + e_i^{(p-1)}$$

avec $e_i = x_i^d - x_i$ et $\lambda_i > 0$ tel que $i = 1, n$

la relation(3.50) peut s'écrire sous la forme compacte suivante

$$s_i = c_i^T Y_i \quad (3.51)$$

avec

$$Y_i = [e_i \quad \dot{e}_i \quad \cdots \quad e_i^{(p-2)} \quad e_i^{(p-1)}]^T$$

$$\mathbf{c}_i^T = [\lambda_i^{(p-1)} \quad (p-1)\lambda_i^{(p-2)} \quad \cdots \quad (p-1)\lambda_i \quad 1]$$

Par conséquence la relation (3.49) prend la forme suivante

$$\mathbf{s} = \mathbf{c}^T \mathbf{Y} \quad (3.52)$$

avec

$$\mathbf{c}^T = \text{diag}[\mathbf{c}_1^T \quad \mathbf{c}_2^T \quad \cdots \quad \mathbf{c}_{n-1}^T \quad \mathbf{c}_n^T]_{(n \times p, n)}$$

$$\mathbf{Y}^T = [\mathbf{Y}_1^T \quad \mathbf{Y}_2^T \quad \cdots \quad \mathbf{Y}_{(n-1)}^T \quad \mathbf{Y}_n^T]_{(p, n \times 1)}$$

La dynamique de s_i est donnée par

$$\dot{s}_i = \mathbf{c}_{ri}^T \mathbf{Y}_i + \mathbf{e}_i^{(p)} \quad (3.53)$$

avec

$$\mathbf{c}_{ri}^T = [0 \quad \lambda_i^{(p-1)} \quad (p-1)\lambda_i^{(p-2)} \quad \cdots \quad 0.5(p-1)(p-2)\lambda_i^2 \quad (p-1)\lambda_i]$$

et la dynamique de \mathbf{s} est alors :

$$\dot{\mathbf{s}} = \mathbf{c}_r^T \mathbf{Y} + \mathbf{e}^{(p)} \quad (3.54)$$

où

$$\mathbf{c}_r^T = \text{diag}[\mathbf{c}_{r1}^T \quad \mathbf{c}_{r2}^T \quad \cdots \quad \mathbf{c}_{rm}^T \quad \mathbf{c}_{rn-1}^T]_{(n \times p, n)}$$

$$\mathbf{e} = [e_1 \quad e_2 \quad \cdots \quad e_{n-1} \quad e_n]^T$$

- Dynamique de l'erreur filtrée en boucle fermée

De la relation (3.44) on obtient

$$\mathbf{x}^{(p)} = F^{-1}(X)(u(t) - G(X)) \quad (3.55)$$

$$\mathbf{e}^{(p)} = \mathbf{x}_d^{(p)} - F^{-1}(X)(u(t) - G(X)) \quad (3.56)$$

En remplaçant $\mathbf{e}^{(p)}$ (3.56) dans (3.54) on aura l'expression de l'erreur filtrée en boucle fermée

$$\dot{\mathbf{s}} = \mathbf{c}_r^T \mathbf{Y} + \mathbf{x}_d^{(p)} - F^{-1}(X)(u(t) - G(X)) \quad (3.57)$$

On définit le signal de référence filtré par

$$\mathbf{Y}_{ref} = \mathbf{x}_d^{(p)} + \mathbf{c}_r^T \mathbf{Y} \quad (3.58)$$

La relation(3.57) devient

$$\begin{aligned}\dot{\mathbf{s}} &= \mathbf{Y}_{ref} - F^{-1}(X)(u(t) - G(X)) \\ F(X)\dot{\mathbf{s}} &= F(X)\mathbf{Y}_{ref} + G(X) - u(t)\end{aligned}\quad (3.59)$$

En utilisant (3.45) et (3.46) la dynamique de l'erreur peut être transformée sous la forme suivante

$$F(X)\dot{\mathbf{s}} = W^T \theta_F^* \mathbf{Y}_{ref} + W^T \theta_G^* + \varepsilon_F(t) \mathbf{Y}_{ref} + \varepsilon_G(t) - u(t) \quad (3.60)$$

A partir des fonctions floues estimées est adoptée la loi de commande suivante

$$u(t) = K_d \mathbf{s} + W^T \hat{\theta}_F \cdot \mathbf{Y}_{ref} + W^T \hat{\theta}_G + \frac{1}{2} F_0 \|X\| \mathbf{s} + u_{glis} \quad (3.61)$$

où $\hat{\theta}_F$ et $\hat{\theta}_G$ sont les paramètres à estimer par des lois d'adaptation paramétriques.

Puisque un modèle, aussi parfait soit-il, ne sera jamais qu'une approximation de la réalité physique, un terme de glissement (u_{glis}) est associé à la loi de commande pour assurer la robustesse vis à vis des perturbations et des incertitudes.

Le terme de glissement est donné par

$$u_{glis} = (\bar{\varepsilon}_F \|\mathbf{Y}_{ref}\| + \bar{\varepsilon}_G) \cdot \text{sign}(\mathbf{s}) \quad (3.62)$$

avec $\bar{\varepsilon}_F \geq \sup_{t \geq 0} \|\varepsilon_F(t)\|$ et $\bar{\varepsilon}_G \geq \sup_{t \geq 0} \|\varepsilon_G(t)\|$

Il est à noter que dans la pratique, l'utilisation d'une loi de commande de mode de glissement de la forme (sgn) peut introduire des signaux hautes fréquences qui risquent d'exciter les dynamiques non modélisées, notamment quand l'amplitude de cette loi de commande est importante. Pour éviter ce problème, une solution consiste à remplacer la fonction signe (sgn) par une fonction de saturation (sat) définie comme suit

$$\text{sat}(x) = \begin{cases} 1 & \text{si } x \geq \zeta \\ x & \text{si } -1 < x < 1 \\ -1 & \text{si } x \leq -\zeta \end{cases} \quad (3.63)$$

Cette solution se traduit par une opération de lissage de la loi de mode de glissement (introduction d'une zone morte)

Les paramètres du contrôleur flou sont adaptés par les lois d'adaptations suivantes

$$\dot{\hat{\theta}}_F = \gamma_2 \cdot W \cdot \mathbf{s} \cdot (\mathbf{Y}_{ref})^T \quad (3.64)$$

$$\dot{\hat{\theta}}_G = \gamma_1 \cdot W \cdot s \quad (3.65)$$

où γ_1 et γ_2 sont des constantes positives.

Vue la complexité du modèle du robot, les fonctions F_0 , $\bar{\varepsilon}_F$ et $\bar{\varepsilon}_G$ sont difficiles à déterminer, Pour faire face à ce problème on utilise une loi de commande dont ces dernières sont adaptées en ligne

$$u(t) = K_d s + \frac{1}{2} \hat{F}_0 \|X\| s + W^T \hat{\theta}_F \cdot Y_{ref} + W^T \hat{\theta}_G + \hat{\varepsilon}_F \|Y_{ref}\| \cdot \text{sign}(s) + \hat{\varepsilon}_G \cdot \text{sign}(s) \quad (3.66)$$

où les paramètres du contrôleur flou sont adaptés par les lois d'adaptation (3.64) et (3.65) et les paramètres des fonctions limites sont mis à jour par

$$\dot{\hat{F}}_0 = \eta_1 \|X\| \|s\| \quad (3.67)$$

$$\dot{\hat{\varepsilon}}_F = \eta_2 \|Y_{ref}\| \|s\| \quad (3.73)$$

$$\dot{\hat{\varepsilon}}_G = \eta_2 \|s\| \quad (3.68)$$

où $\gamma_1, \gamma_2, \eta_1$ et $\eta_2 > 0$

3.3.3 ANALYSE DE LA STABILITE

L'objectif est d'assurer la stabilité des structures de commande dans le sens où tous les signaux d'entrée et de sortie restent bornés et l'erreur de poursuite tend asymptotiquement vers zéro.

Afin de démontrer la stabilité du système, nous considérons la fonction de Lyapunov candidate suivante :

$$V = \frac{1}{2} s^T F(X) s + \frac{1}{2\gamma_1} \tilde{\theta}_G^T \tilde{\theta}_G + \frac{1}{2\gamma_2} \text{trace}(\tilde{\theta}_F^T \tilde{\theta}_F) + \frac{1}{2\eta_1} (\tilde{F}_0)^2 + \frac{1}{2\eta_2} (\tilde{\varepsilon}_F)^2 + \frac{1}{2\eta_2} (\tilde{\varepsilon}_G)^2 \quad (3.69)$$

où $\tilde{\theta}_F = \theta_F^* - \hat{\theta}_F$, $\tilde{\theta}_G = \theta_G^* - \hat{\theta}_G$, $\tilde{F}_0 = F_0 - \hat{F}$, $\tilde{\varepsilon}_F = \bar{\varepsilon}_F - \hat{\varepsilon}_F$ et $\tilde{\varepsilon}_G = \bar{\varepsilon}_G - \hat{\varepsilon}_G$

En dérivant la fonction de Lyapunov par rapport au temps, nous obtenons :

$$\dot{V} = \frac{1}{2} s^T \dot{F}(X) s + s^T F(X) \dot{s} - \frac{1}{\gamma_1} \tilde{\theta}_G^T \dot{\tilde{\theta}}_G - \frac{1}{\gamma_2} \text{trace}(\tilde{\theta}_F^T \dot{\tilde{\theta}}_F) - \frac{1}{\eta_1} \tilde{F}_0 \dot{\tilde{F}}_0 - \frac{1}{\eta_2} \tilde{\varepsilon}_F \dot{\tilde{\varepsilon}}_F - \frac{1}{\eta_2} \tilde{\varepsilon}_G \dot{\tilde{\varepsilon}}_G \quad (3.70)$$

En remplaçant dans (3.70) $F(X)\dot{s}$ par son expression (3.60) et en introduisant l'expression de $u(t)$ (3.66) il vient

$$\begin{aligned} \dot{V} = & \frac{1}{2} \mathbf{s}^T \dot{F}(X) \mathbf{s} - \frac{1}{2} \mathbf{s}^T \hat{F}_0 \|X\| \mathbf{s} - \frac{1}{\eta_1} \tilde{F}_0 \dot{F}_0 + \mathbf{s}^T (\varepsilon_F \mathbf{Y}_{ref} + \varepsilon_G) - \mathbf{s}^T (\hat{\varepsilon}_F \|\mathbf{Y}_{ref}\| + \hat{\varepsilon}_G) \text{sign}(\mathbf{s}) - \\ & \frac{1}{\eta_2} \tilde{\varepsilon}_F \dot{\varepsilon}_F - \frac{1}{\eta_2} \tilde{\varepsilon}_G \dot{\varepsilon}_G + (\mathbf{s}^T W^T \tilde{\theta}_F \mathbf{Y}_{ref}) - \frac{1}{\gamma_2} \text{trace}(\tilde{\theta}_F^T \dot{\theta}_F) + \mathbf{s}^T W^T \tilde{\theta}_G - \frac{1}{\gamma_1} \tilde{\theta}_G^T \dot{\theta}_G \end{aligned} \quad (3.71)$$

En introduisant les lois d'adaptation paramétriques (3.64) et (3.65) dans (3.71) on arrive à l'expression

$$\begin{aligned} \dot{V} = & -K_d \mathbf{s}^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T \dot{F} \|X\| \mathbf{s} - \frac{1}{2} \mathbf{s}^T \hat{F}_0 \|X\| \mathbf{s} - \frac{1}{\eta_1} \tilde{F}_0 \dot{F}_0 + \mathbf{s}^T (\varepsilon_F \|\mathbf{Y}_{ref}\| + \varepsilon_G) - \\ & \mathbf{s}^T (\hat{\varepsilon}_F \|\mathbf{Y}_{ref}\| + \hat{\varepsilon}_G) \text{sign}(\mathbf{s}) - \frac{1}{\eta_2} \tilde{\varepsilon}_F \dot{\varepsilon}_F - \frac{1}{\eta_2} \tilde{\varepsilon}_G \dot{\varepsilon}_G \end{aligned} \quad (3.72)$$

Cette inégalité est toujours satisfaite

$$\begin{aligned} \dot{V} \leq & -K_d \mathbf{s}^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T \dot{F}_0 \|X\| \mathbf{s} - \frac{1}{2} \mathbf{s}^T \hat{F}_0 \|X\| \mathbf{s} - \frac{1}{\eta_1} \tilde{F}_0 \dot{F}_0 + \bar{\varepsilon}_F \|\mathbf{Y}_{ref}\| \|\mathbf{s}\| + \\ & \hat{\varepsilon}_F \|\mathbf{Y}_{ref}\| \|\mathbf{s}\| - \frac{1}{\eta_2} \tilde{\varepsilon}_F \dot{\varepsilon}_F + \bar{\varepsilon}_G \|\mathbf{s}\| - \hat{\varepsilon}_G \|\mathbf{s}\| - \frac{1}{\eta_2} \tilde{\varepsilon}_G \dot{\varepsilon}_G \end{aligned} \quad (3.73)$$

Ce qui est équivalent à

$$\begin{aligned} \dot{V} \leq & -K_d \mathbf{s}^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T \tilde{F}_0 \|X\| \mathbf{s} - \frac{1}{\eta_1} \tilde{F}_0 \dot{F}_0 + \tilde{\varepsilon}_F \|\mathbf{Y}_{ref}\| \|\mathbf{s}\| \\ & - \frac{1}{\eta_2} \tilde{\varepsilon}_F \dot{\varepsilon}_F + \tilde{\varepsilon}_G \|\mathbf{s}\| - \frac{1}{\eta_2} \tilde{\varepsilon}_G \dot{\varepsilon}_G \end{aligned} \quad (3.74)$$

En utilisant les lois d'adaptation des paramètres limite $\dot{\tilde{F}}_0$, $\dot{\tilde{\varepsilon}}_F$ et $\dot{\tilde{\varepsilon}}_G$ on obtient

$$\dot{V} \leq -K_d \mathbf{s}^T \mathbf{s} < 0, \forall \mathbf{s} \neq 0 \quad (3.75)$$

Donc S converge asymptotiquement vers zéros et tous les signaux d'entrée-sortie sont bornés.

3.3.4 RESULTATS DE SIMULATION

Dans cette partie, nous procédons à l'application de la commande adaptative centralisée floue directe au robot PUMA 560. L'identification entre l'équation (3.44) et le

modèle du robot (1.11), conduit à $F(X) = A(q)$, $G(X) = H(q, \dot{q})$, $u = \Gamma$, $x = q = [q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6]^T$, $X = [q \ \dot{q}]^T = [q_1 \ \dots \ q_6 \ \dot{q}_1 \ \dots \ \dot{q}_6]^T$ et $P=2$ ordre du système.

Afin de réduire le nombre de règles floues dans la construction du système flou de Sugeno, on considère à nouveau que la structure du modèle du robot est connue [Arms-96] et seuls les paramètres du modèle sont inconnus. Dans ce cas le vecteur X se réduit à $X = [q_2 \ \dots \ q_5 \ \dot{q}_1 \ \dots \ \dot{q}_5]^T$.

Etant donné que les éléments de $F(X)$ et $G(X)$ sont incertains, trente six (36) systèmes flous sont utilisés pour approcher les éléments de $F(X)$ et six (6) systèmes flous sont utilisés pour approcher les éléments de $G(X)$. Tous les systèmes flous utilisés pour estimer $F(X)$ et $G(X)$ ont comme entrée les variables $q_2, q_3, q_4, q_5, \dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5$. Chaque variable d'entrée est caractérisée uniquement par deux (2) fonctions d'appartenances gaussiennes ce qui permet de réduire la base des règles floues à une taille congrue (pour de tel système, celle-ci peut devenir facilement explosive si le nombre d'ensembles flous est plus élevé).

Les fonctions d'appartenances gaussiennes sont de la forme

$$\mu_{f_i^j}(q_i) = \exp\left(-0.5 \left(\frac{q_i - c_i^j}{\sigma_i}\right)^2\right) \text{ où } i=2, \dots, 5 \text{ et } j=1, 2$$

$$\mu_{f_i^j}(\dot{q}_i) = \exp\left(-0.5 \left(\frac{\dot{q}_i - \bar{c}_i^j}{\bar{\sigma}_i}\right)^2\right) \text{ } i=1, \dots, 6 \text{ et } j=1, 2$$

où les valeurs de σ_i , c_i^j , $\bar{\sigma}_i$ et \bar{c}_i^j et sont choisies selon le tableau 3.1

q_i	c_i^1	c_i^2	σ_i		\dot{q}_i	\bar{c}_i^1	\bar{c}_i^2	$\bar{\sigma}_i$
					\dot{q}_1	-0.26	0.15	0.1
q_2	0.18	1	0.205		\dot{q}_2	-0.32	0.21	0.13
q_3	-0.2	1.2	0.35		\dot{q}_3	-0.42	0.58	0.25
q_4	-1.75	4.4	1.28		\dot{q}_4	-0.1	2	0.52
q_5	0.72	2.95	0.55		\dot{q}_5	-0.75	0.75	0.375

Tab.3.4 Valeurs de paramètres des fonctions d'appartenance

Les coefficients de commande et d'adaptation sont donnés au tableau 3.5

Articulation	K_d	λ	γ_1	γ_2	η_1	η_2
N° 1	1000	50	2	2	2	0.02
N° 2	1000	50	2	2	2	0.02
N° 3	1000	50	2	2	2	0.02
N° 4	50	5	2	2	2	0.02
N° 5	60	0.01	2	2	2	0.05
N° 6	60	0.01	2	2	0.001	0.05

Tab.3.5 Coefficients de commande et d'adaptation.

Les resultats de simulation sont présentés par les figures 3.11-3.14.

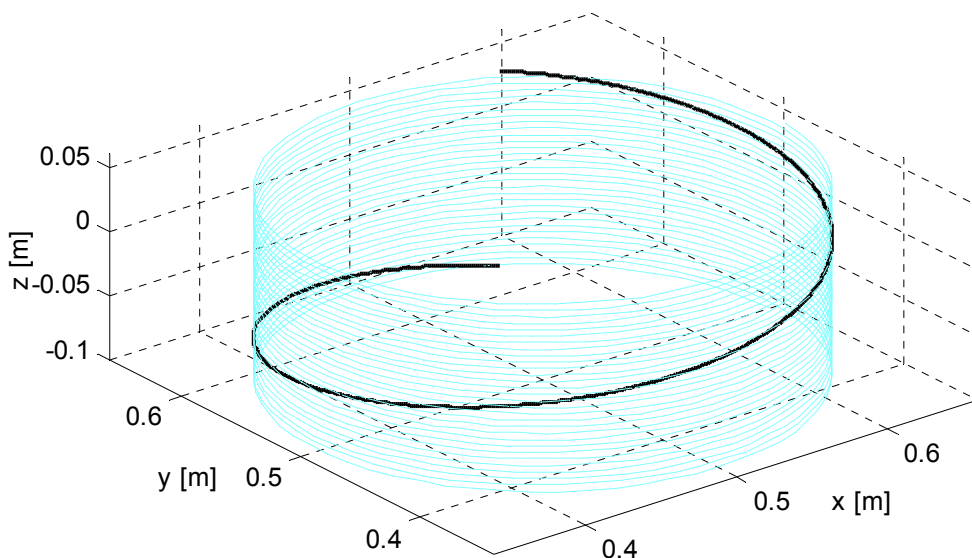


Fig.3.11.a Trajectoire cartésienne de l'outil

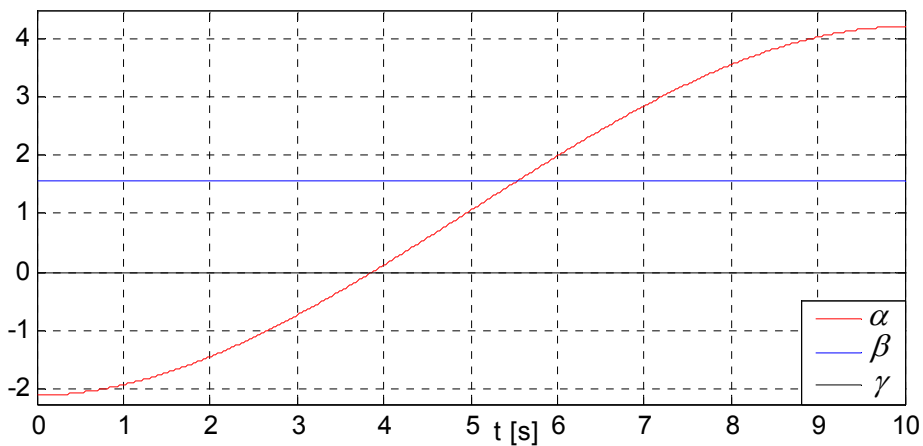


Fig.3.11.b Orientation de l'outil (unité : radian)

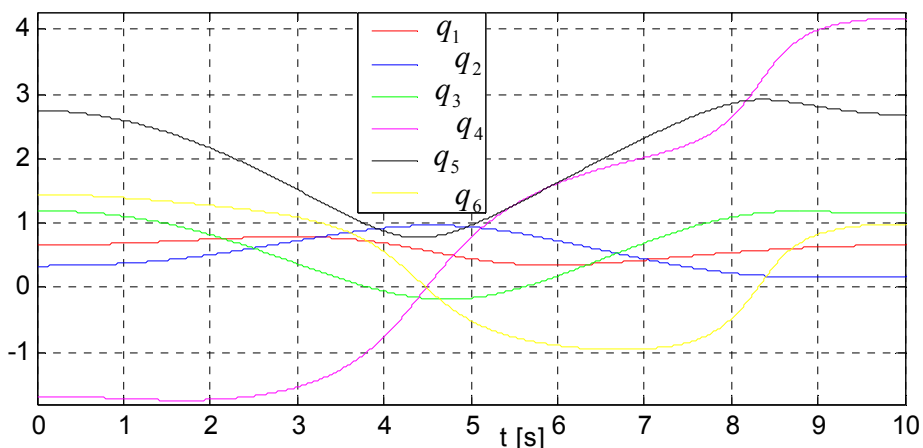


Fig.3.12 Trajectoires articulaires (unité : radian)

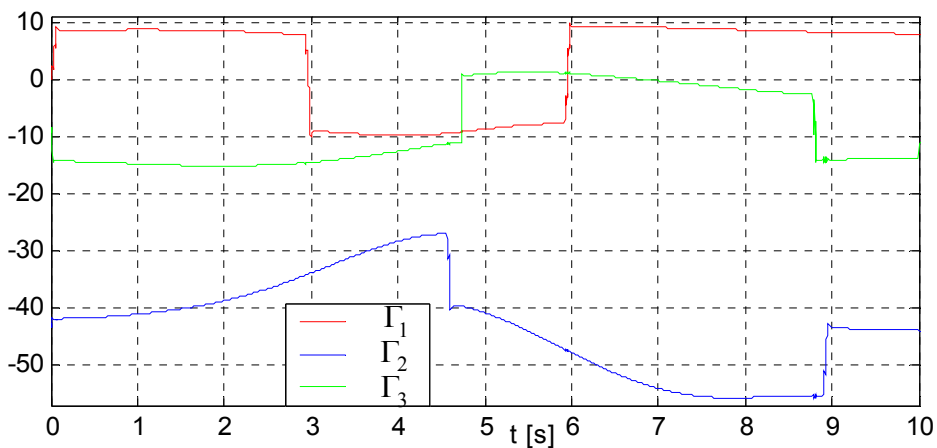


Fig.3.13.a Couples des trois premiers actionneurs (unité : N.m)

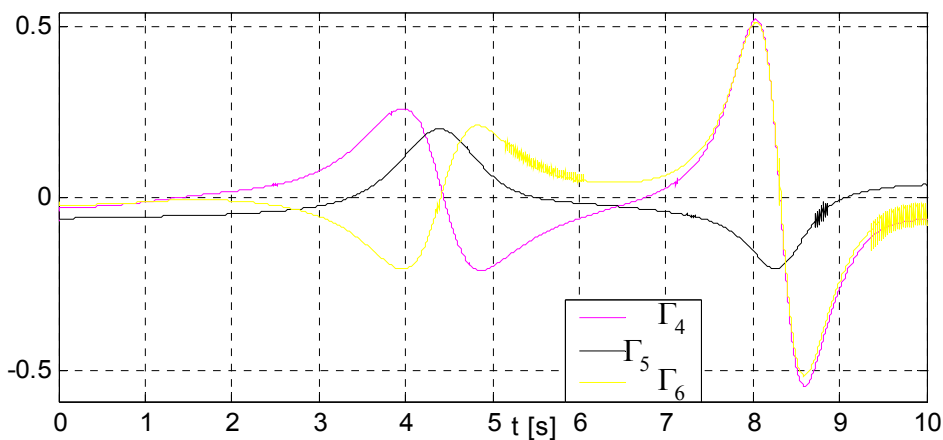


Fig.3.13.b Couples des trois premiers actionneurs (unité : N.m)

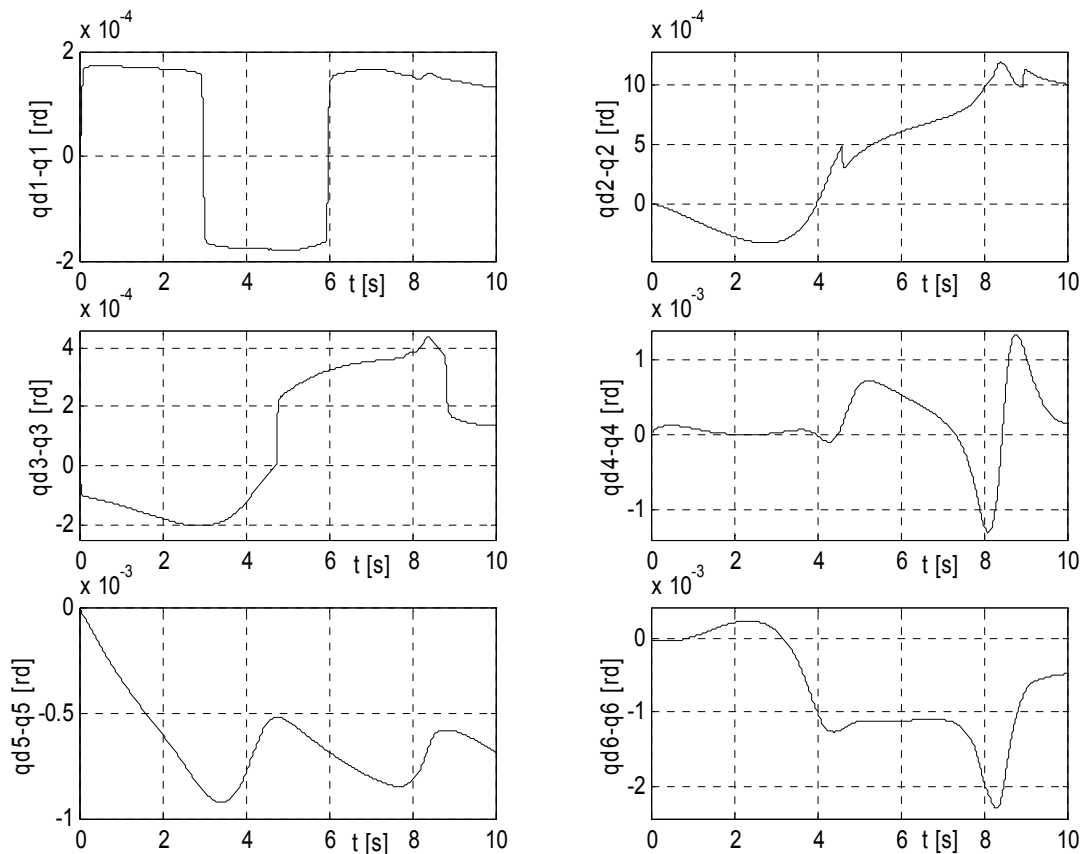


Fig.3.14.a Erreurs articulaires

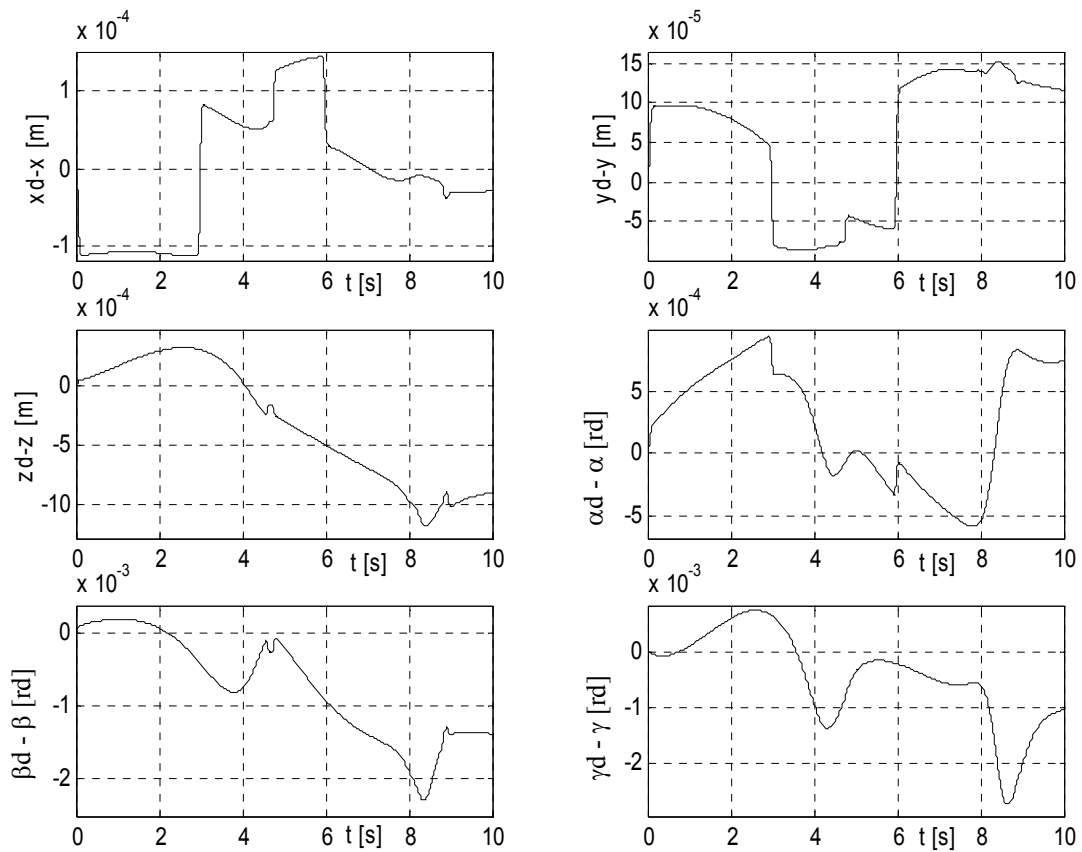


Fig.3.14.b Image des erreurs articulaires dans l'espace cartésien

▪ Robustesse vis-à-vis des perturbations extérieures

Afin de tester la robustesse de la commande vis-à-vis des perturbations extérieures, on applique au robot une charge de 2 kg (masse de l'outil) à partir de l'instant $t=5$ secondes (figure 3.15-3.17).

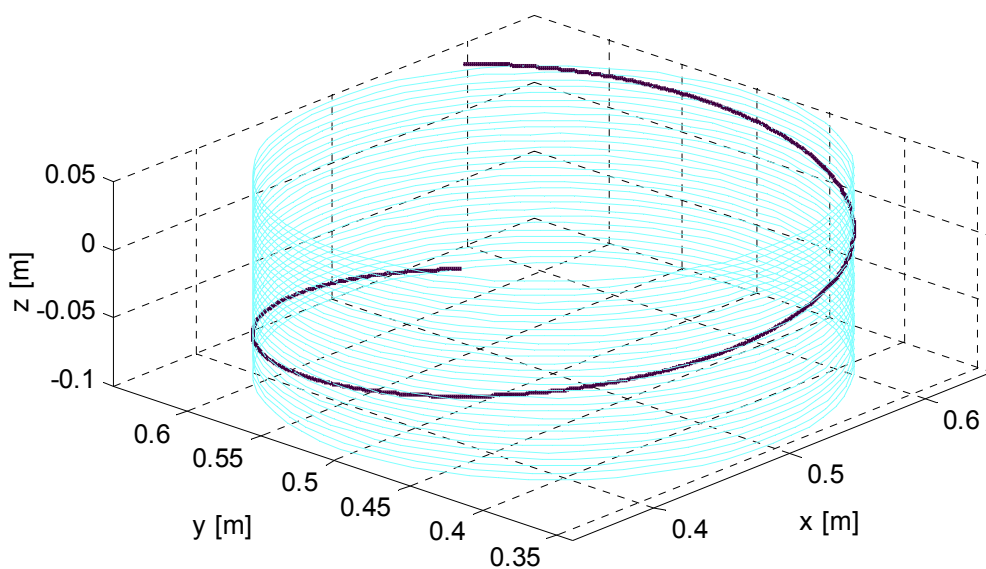


Fig.3.15.a Trajectoire de l'outil

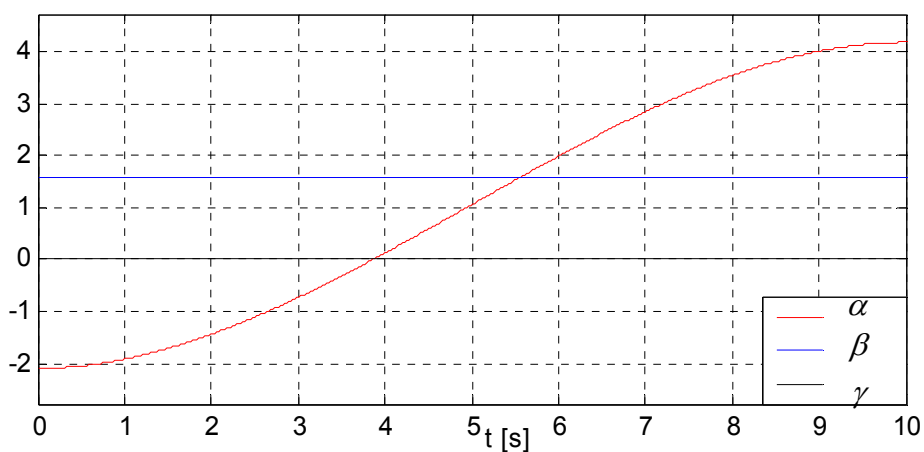


Fig.3.15.b Orientation de l'outil (unité : radian)

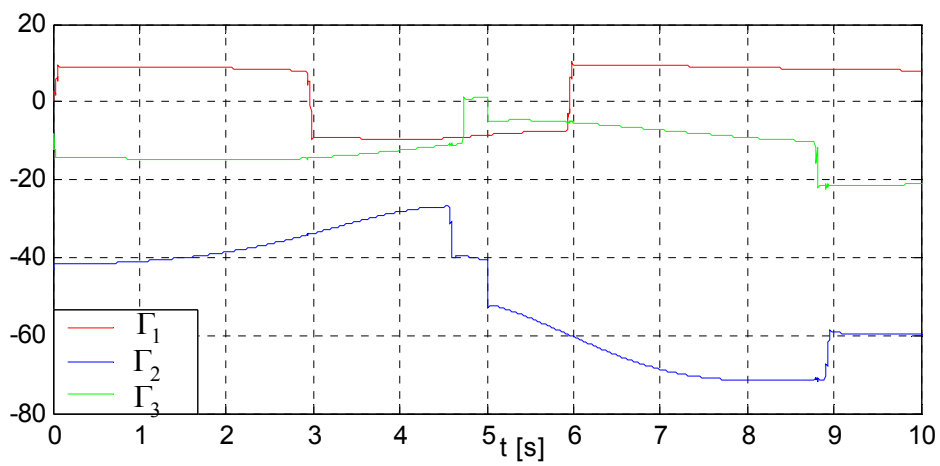


Fig.3.16.a Couples actionneurs des trois premières articulations (unité : N.m)

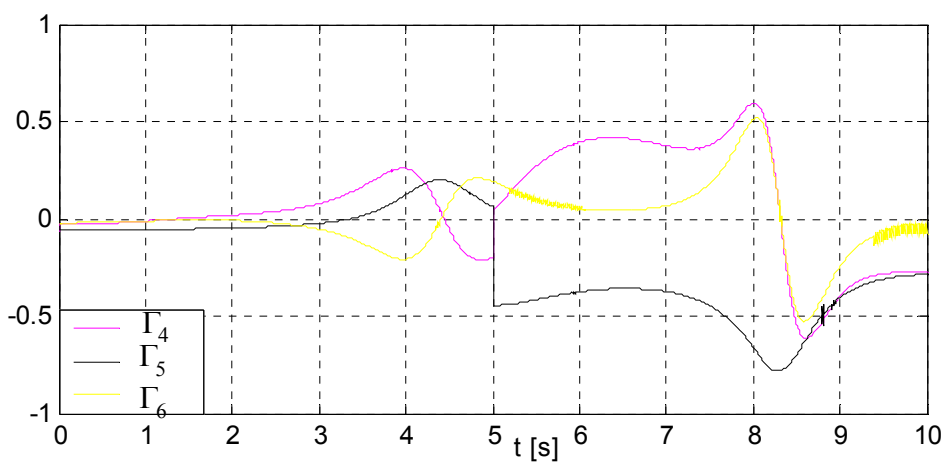


Fig.3.16.b Couples actionneurs des trois dernières articulations (unité : N.m)

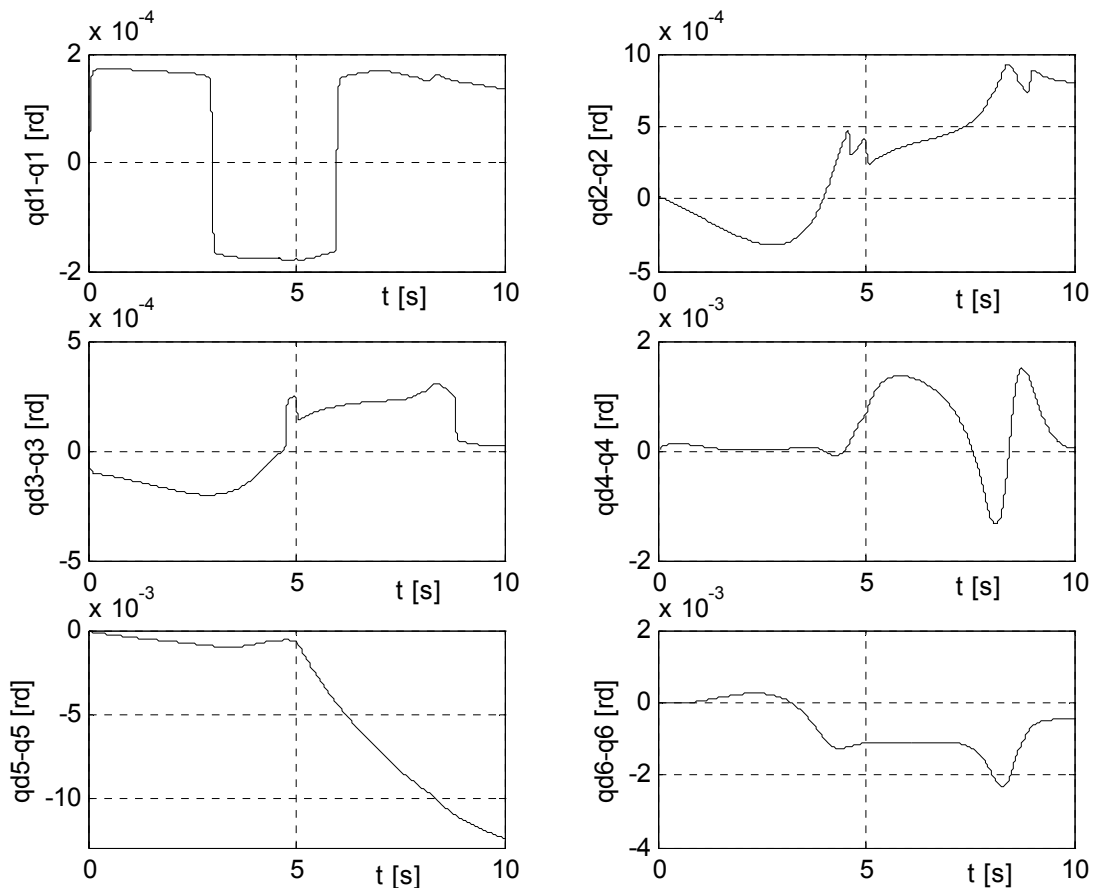


Fig.3.17.a Erreurs articulaires

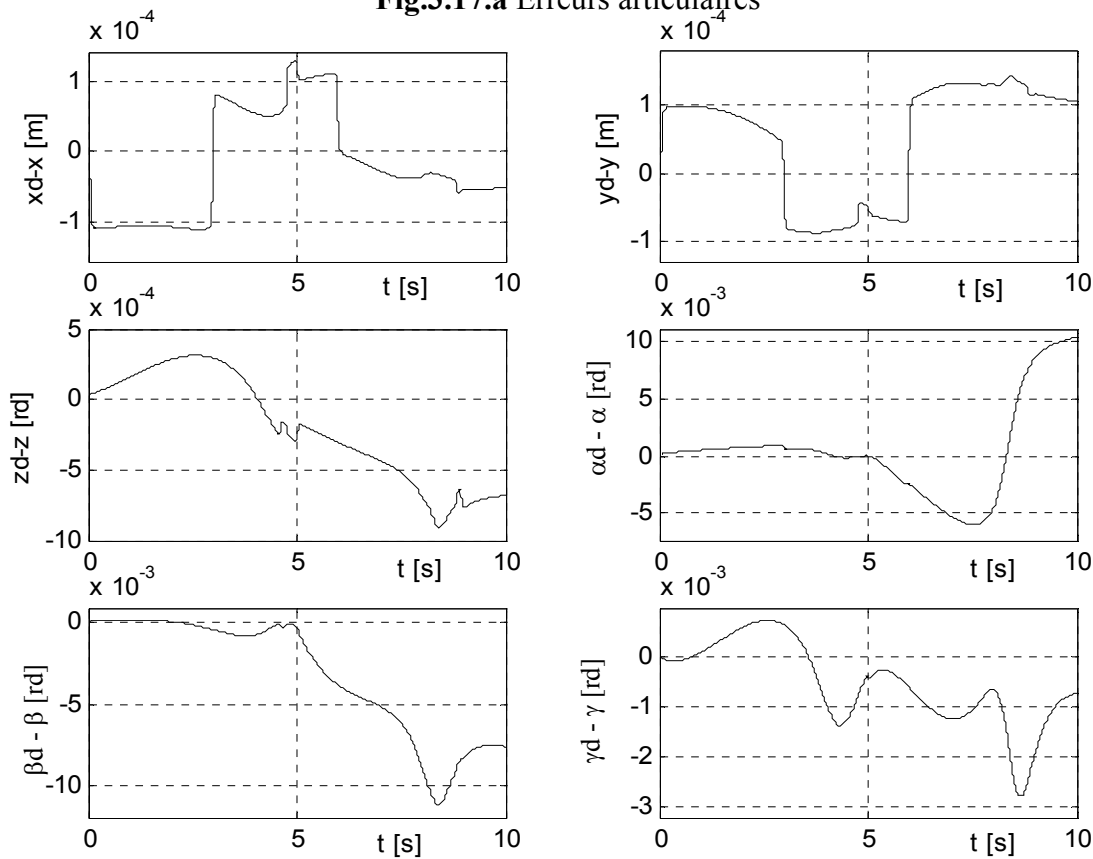


Fig.3.17.b Erreurs cartésiennes

▪ **Robustesse vis-à-vis des variations des paramètres du robot**

Pour tester la robustesse de la loi de commande adaptative floue directe on procède de la même manière que pour la loi de commande indirecte (application d'une variation de 100% à la matrice d'inertie à partir de l'instant $t= 5$ secondes). Les résultats sont donnés aux figures 3.18-3.20).

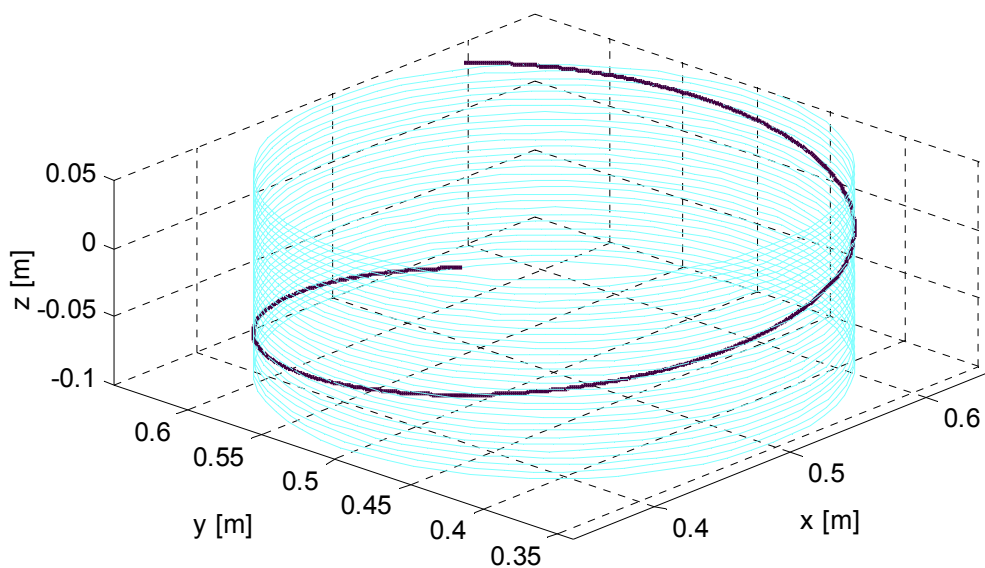


Fig.3.18.a Trajectoire de l'outil

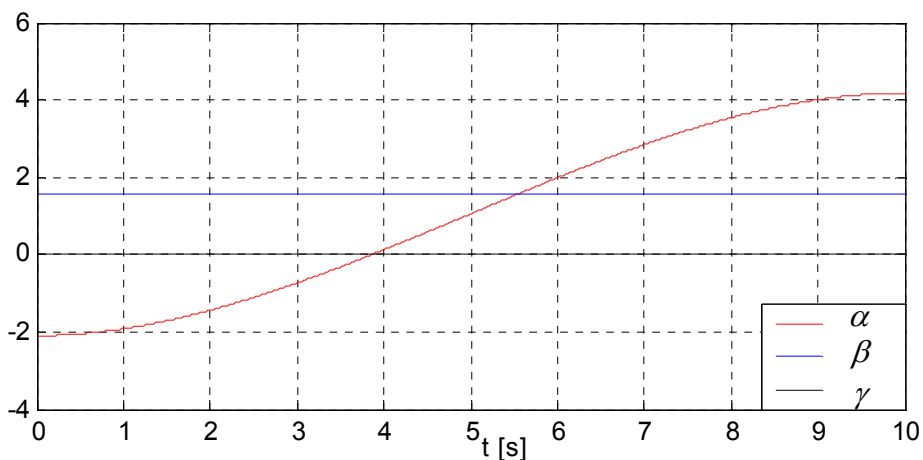


Fig.3.18.b Orientation de l'outil (unité : radian)

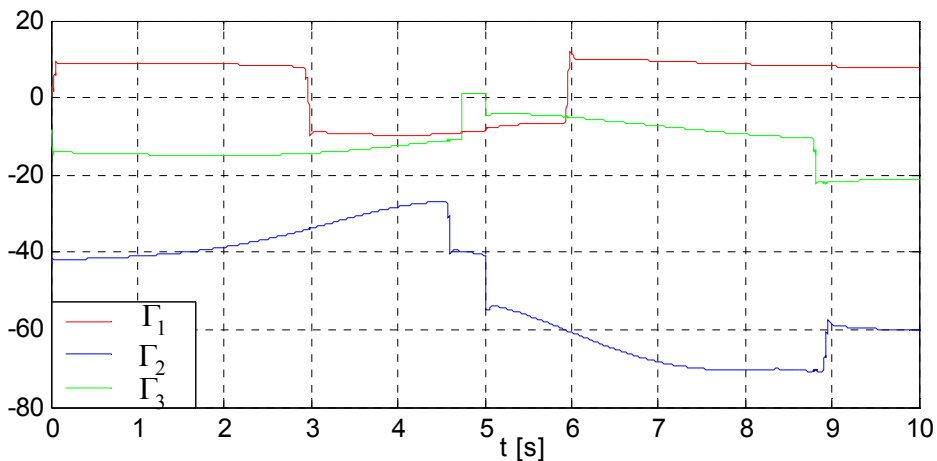


Fig.3.19.a Couples développés par les trois premières articulations (unité : N.m)

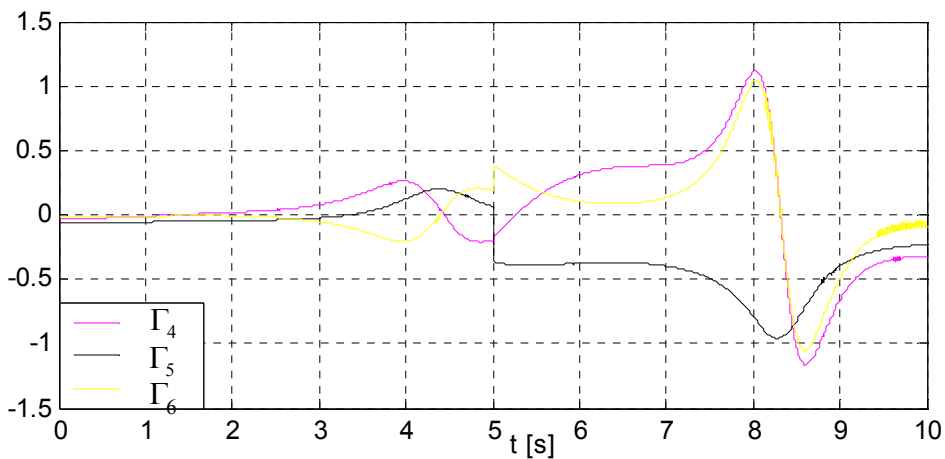


Fig.3.19.b Couples développés par les trois dernières articulations (unité : N.m)

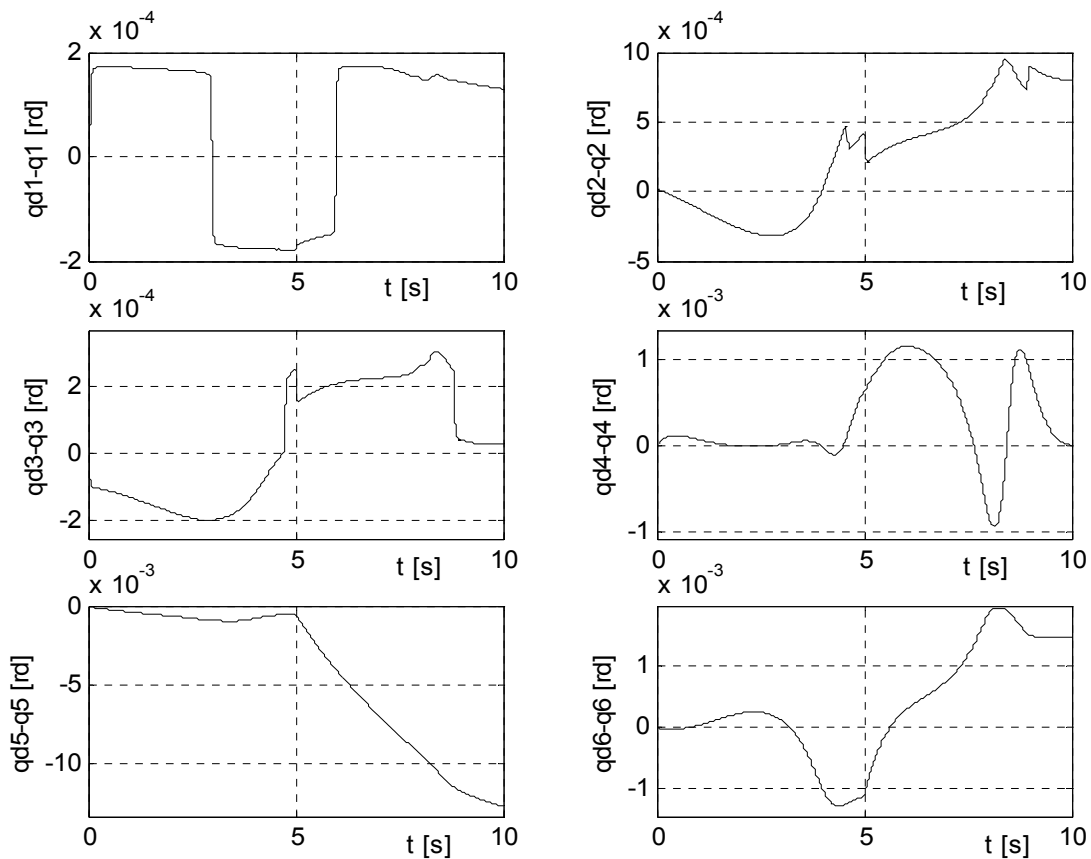


Fig.3.20.a Erreurs articulaires

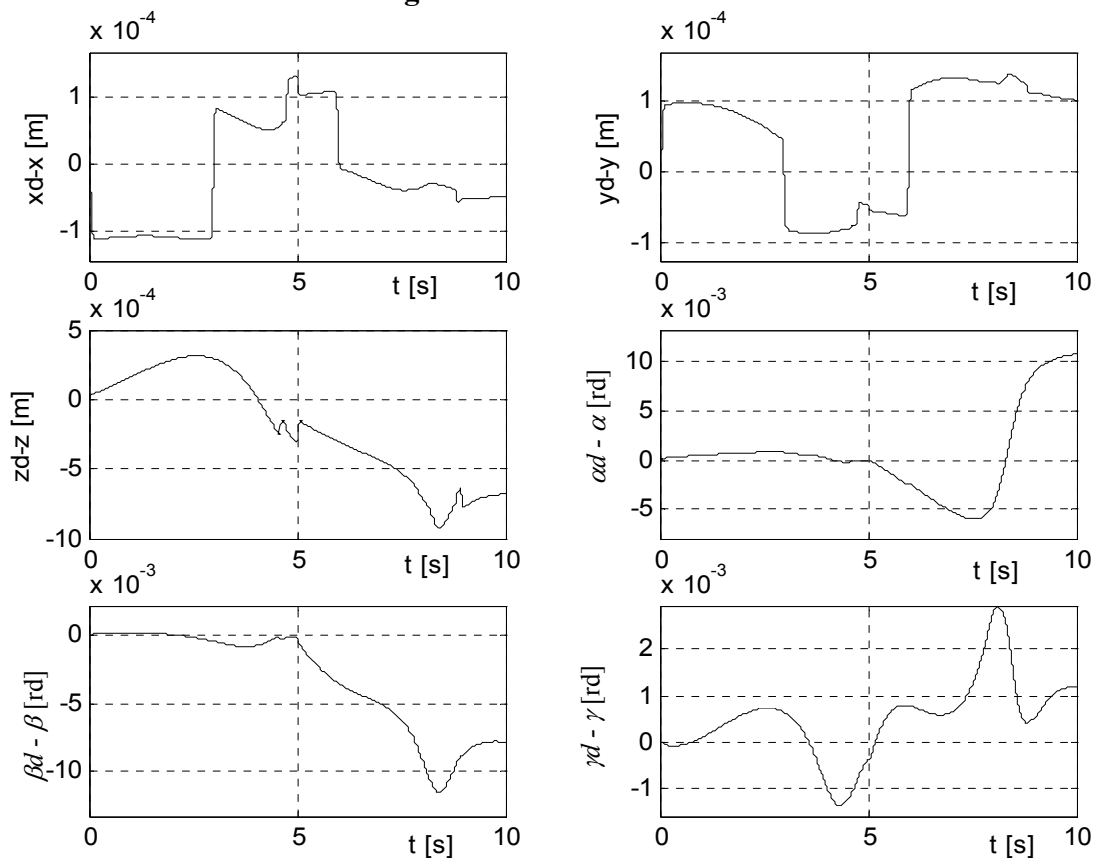


Fig.3.20.2 Erreurs cartésiennes

	Tests sans perturbations	Tests avec charge	Tests avec perturbations paramétriques
$x^d - x [mm]$	0.14	0.13	0.11
$y^d - y [mm]$	0.15	0.14	0.13
$z^d - z [mm]$	1.20	0.90	0.92
$\alpha^d - \alpha [rd]$	9.32×10^{-4}	1×10^{-2}	1×10^{-2}
$\beta^d - \beta [rd]$	2.3×10^{-3}	1×10^{-2}	1×10^{-2}
$\gamma^d - \gamma [rd]$	2.7×10^{-3}	2.8×10^{-3}	1.4×10^{-3}

Tab.3.6 Erreurs maximales dans l'espace de la tâche

	Tests sans perturbations	Tests avec charge	Tests avec perturbations paramétriques
$\Gamma_1 [N.m]$	10.02	10.22	15.88
$\Gamma_2 [N.m]$	55.68	71.79	72.57
$\Gamma_3 [N.m]$	14.99	22.43	24.63
$\Gamma_4 [N.m]$	0.54	0.61	1.17
$\Gamma_5 [N.m]$	0.20	0.78	0.96
$\Gamma_6 [N.m]$	0.51	0.52	1.05

Tab.3.7 Couples maximaux développés par les actionneurs

Les figures 3.11.a et 3.11.b révèlent une bonne poursuite en position/orientation de l'outil lors de l'exécution de la tâche.

Les trajectoires des six axes du robot (fig.3.12) lors de l'exécution de la tâche sont lisses et continues.

Les faibles ondulations apparues sur les couples actionneurs (fig.3.13.b, 3.16.b et 3.19.b) sont dues à la présence du terme de glissement dans la loi de commande.

La robustesse de cette loi de commande est validée par les faibles valeurs des erreurs maximales de poursuite (voir tab.3.6) obtenus lors des essais où le robot est soumis à des perturbations importantes, une charge de 2kg comme perturbation extérieures et une variation paramétrique de 100% de la matrice d'inertie. De plus, les couples maximaux par chaque enregistrés lors de ces différents tests restent dans les limites tolérées (Tab.3.7).

3.4 CONCLUSION

Dans ce chapitre, on a appliqué deux techniques de commande adaptatives floues centralisées au robot PUMA 560 à six degrés de liberté. Dans l'approche indirecte la loi de commande est basée sur la principe d'équivalence certaine, dans l'approche directe, la loi de commande est augmentée d'un terme de robustesse pour compenser l'effet des erreurs d'approximation.

Quoi que le modèle du robot est considéré inconnu, les résultats de simulation montrent bien l'efficacité des systèmes flous dans la commande de la position et de l'orientation des robots. Nous avons aussi constaté que ces deux stratégies de commande ne présentent pas uniquement une robustesse vis-à-vis des incertitudes, mais aussi une robustesse en présence d'un changement dans les paramètres du système.

L'inconvénient majeur de la commande adaptative floue centralisée pour la position et l'orientation des robots manipulateurs à plusieurs degrés de liberté est le nombre important de règles floues à utiliser pour construire les systèmes flous, ce qui augmente le temps de calcul et complique la conception de l'algorithme de commande. La solution à ce problème consiste en utilisation des techniques de commande adaptatives décentralisées.

CHAPITRE 4 :

COMMANDE ADAPTATIVE FLOUE DECENTRALISEE

COMMANDE ADAPTATIVE FLOUE DECENTRALISEE

4.1 INTRODUCTION

Au chapitre précédent, on a montré et vérifié qu'il est très laborieux de mettre en œuvre les méthodes centralisées dans le cas des robots à six degrés de liberté dont la dynamique non linéaire et fortement couplée est représenté par une classe des systèmes MIMO, En effet, pour construire un système flou, le nombre de règles floues augmente exponentiellement avec le nombre de variables d'entrée, ce qui complique l'algorithme de commande et augmente le temps de calcul. Par conséquent, il est très avantageux de concevoir des contrôleurs décentralisés pour de tels systèmes.

La structure de commande décentralisée considère le robot comme un ensemble de sous-systèmes interconnectés dont chacun caractérise une articulation. Chaque sous-système est commandé par une unité de commande qui n'a accès qu'aux informations locales [Labi-05] [Chia-02] [Liu-00] [Jin-98] [Spoo-96] [Tang-02].

La commande adaptative décentralisée présente plusieurs avantages tels que : La réduction du nombre de règles dans la conception des systèmes flous, ceux-ci permettent de surmonter le problème de manque de connaissance des non linéarités locales de chaque sous système. Aussi, la structure décentralisée rend possible l'implantation par des architectures parallèles.

Les approches de commande adaptatives floues décentralisées sont classifiées en deux catégories : les approches indirectes et les approches directes. Les approches indirectes se basent généralement sur l'approximation des non linéarités locales de chaque sous-système par les modèles flous. En revanche les approches directes se basent sur l'approximation par des modèles flous des lois de commandes stabilisatrices.

Ce chapitre est consacré à l'application de deux lois de commande adaptatives floues décentralisées aux robots manipulateurs. La première partie est réservée à l'approche indirecte et la deuxième partie est consacrée à l'approche directe. Les lois d'adaptations sont élaborées en se basant sur la théorie des modes glissants et la théorie de la stabilité au sens de Lyapunov. La synthèse des lois de commande ne nécessite pas la connaissance du modèle et n'utilise que l'information locale au niveau de chaque sous-système.

4.2 COMMANDE ADAPTATIVE DECENTRALISEE FLOUE INDIRECTE

Dans la commande adaptative floue indirecte, les systèmes flous sont utilisés pour approcher les fonctions incertaines de chaque sous-système (articulation) et à développer, à partir de ces approximations, la loi de commande adaptative décentralisée et les lois d'adaptations paramétriques afin d'atteindre les objectifs de commande.

Considérons le robot manipulateur à n degrés de liberté dont le modèle est donné par

$$\ddot{q} = F(q, \dot{q}) + G(q)\Gamma \quad (4.1)$$

Ce système peut être décomposé de n sous-systèmes \sum_i non linéaires donnés par

$$\sum_i \Rightarrow \ddot{q}_i = f_i(q_i, \dot{q}_i) + g_i(q_i, \dot{q}_i)\Gamma_i + \Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) \quad (4.2)$$

avec q_i est la sortie du sous système \sum_i qui caractérise la $i^{\text{ème}}$ articulation. ($i = 1, \dots, n$)..

Les variables d'état q_i et \dot{q}_i du sous-système \sum_i sont supposées mesurables et Γ_i est le couple appliqué à la $i^{\text{ème}}$ articulation, $f_i(q_i, \dot{q}_i)$ et $g_i(q_i, \dot{q}_i)$ sont des fonctions non linéaires continues incertaines, et $\Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n)$ représente l'effet des interconnexions entre le sous système \sum_i et les autre sous systèmes.

Pour chaque articulation \sum_i , on définit les erreurs de poursuites

$$e_i(t) = q_i^d(t) - q_i(t) \text{ et } \dot{e}_i(t) = \dot{q}_i^d(t) - \dot{q}_i(t) \quad (4.3)$$

et le signal de référence

$$v_i = \ddot{q}_i^d + k_{i,1}\dot{e}_i + k_{i,0}e_i$$

Posons

$$\mathbf{e}_i = [e_i \quad \dot{e}_i]^T$$

$$\mathbf{k}_i = [k_{i,0} \quad k_{i,1}]^T$$

d'où

$$v_i = \ddot{q}_i^d + \mathbf{k}_i^T \mathbf{e}_i \quad (4.4)$$

Par introduction de (4.4) dans (4.2), il vient

$$\dot{\mathbf{e}}_i = \mathbf{A}_i \mathbf{e}_i + \mathbf{b}_i \left\{ v_i - f_i(q_i, \dot{q}_i) - g_i(q_i, \dot{q}_i)\Gamma_i - \Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) \right\} \quad (4.5)$$

avec
$$A_i = \begin{bmatrix} 0 & 1 \\ -k_{i,0} & -k_{i,1} \end{bmatrix} \text{ et } b_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Les paramètres $k_{i,0}$ et $k_{i,1}$ sont choisis de sorte que la matrice A_i soit stable. Ainsi pour toute matrice donnée $Q_i = Q_i^T > 0$, il existe une matrice unique $P_i = P_i^T > 0$ solution de l'équation de Lyapunov

$$P_i A_i + A_i^T P_i = -Q_i \quad (4.6)$$

On suppose que les interconnexions satisfont la condition :

$$|\Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n)| \leq \delta_{i,0} + \sum_{j=1}^n \delta_{i,j} \|e_j\| \quad (4.7)$$

avec $\delta_{i,0}, \dots, \delta_{i,n}$ sont des constante positives inconnues, et $\|\bullet\|$ représente la norme Euclidienne.

Nous supposons que $g_i(q_i, \dot{q}_i) \geq \underline{g}_i \geq 0$

Dans le cas où les sous-systèmes sont isolés l'un de l'autre et que les fonctions $f_i(q_i, \dot{q}_i)$ et $g_i(q_i, \dot{q}_i)$ sont parfaitement connues, la loi de commande par couple calculé est directement applicable

$$\Gamma_i = \frac{1}{g_i} (v_i - f_i(q_i, \dot{q}_i)) \quad (4.8)$$

Par substitution de (4.8) dans (4.5), il vient

$$\dot{e}_i = A_i e_i \quad (4.9)$$

Du fait que la matrice A_i est stable, alors les erreurs de poursuite convergent vers zéro.

Dans la suite de cette partie, on considère que les fonctions $f_i(q_i, \dot{q}_i)$ et $g_i(q_i, \dot{q}_i)$ sont inconnues et que les interactions $\Delta_i(q_i, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n)$ sont non nulles. De ce fait, l'implantation de la loi de commande (4.8) s'avère impossible.

Notre objectif de commande est de synthétiser une loi de commande adaptative utilisant les systèmes flous pour chaque sous système en exploitant uniquement les variables d'état locales propre à chaque articulation.

4.2.1 ESTIMATION PAR LES SYSTEMES FLOUS

Les fonctions non linéaires $f_i(q_i, \dot{q}_i)$ et $g_i(q_i, \dot{q}_i)$ du sous-système \sum_i peuvent être approchées par les systèmes flous comme suit

$$\hat{f}_i(q_i, \dot{q}_i, \hat{\theta}_{fi}) = w_{fi}^T(q_i, \dot{q}_i) \hat{\theta}_{fi} \quad (4.10)$$

$$\hat{g}_i(q_i, \dot{q}_i, \hat{\theta}_{gi}) = w_{gi}^T(q_i, \dot{q}_i) \hat{\theta}_{gi} \quad (4.11)$$

avec $\hat{\theta}_{fi}$ et $\hat{\theta}_{gi}$ sont les vecteurs des paramètres ajustés, $w_{fi}(q_i, \dot{q}_i)$ et $w_{gi}(q_i, \dot{q}_i)$ sont des vecteurs de fonctions floues de base.

Les fonctions réelles $f_i(q_i, \dot{q}_i)$ et $g_i(q_i, \dot{q}_i)$ peuvent s'exprimer sous la forme

$$f_i(q_i, \dot{q}_i) = w_{fi}^T(q_i, \dot{q}_i) \theta_{fi}^* + \varepsilon_{fi}(q_i, \dot{q}_i) \quad (4.12)$$

$$g_i(q_i, \dot{q}_i) = w_{gi}^T(q_i, \dot{q}_i) \theta_{gi}^* + \varepsilon_{gi}(q_i, \dot{q}_i) \quad (4.13)$$

où $\varepsilon_{fi}(q_i, \dot{q}_i)$ et $\varepsilon_{gi}(q_i, \dot{q}_i)$ sont les erreurs d'approximation, θ_{fi}^* et θ_{gi}^* sont respectivement les paramètres optimaux de $\hat{\theta}_{fi}$ et $\hat{\theta}_{gi}$.

D'après la propriété d'approximateur universel caractérisant les systèmes flous utilisés, on peut alors supposer que les erreurs d'approximation sont bornées telles que :

$$|\varepsilon_{fi}(q_i, \dot{q}_i)| \leq \bar{\varepsilon}_{fi}, \quad |\varepsilon_{gi}(q_i, \dot{q}_i)| \leq \bar{\varepsilon}_{gi} \quad (4.14)$$

où $\bar{\varepsilon}_{fi}$ et $\bar{\varepsilon}_{gi} > 0$.

4.2.2 SYNTHÈSE DE LA LOI DE COMMANDE

La loi de commande est composée de deux termes : un terme de commande adaptatif, Γ_{fi} , et un terme de commande robuste, Γ_{ri} , introduit pour compenser les effets des erreurs de reconstruction et les effets d'interconnexions. Cette dernière est donnée comme suit

$$\Gamma_i = \Gamma_{fi} + \Gamma_{ri} \quad (1.15)$$

avec

$$\Gamma_{fi} = \frac{\hat{g}_i(q_i, \dot{q}_i, \hat{\theta}_{gi})}{\varepsilon_0 + \hat{g}_i^2(q_i, \dot{q}_i, \hat{\theta}_{gi})} \left\{ v_i - \hat{f}_i(q_i, \dot{q}_i, \hat{\theta}_{fi}) + \beta_i \hat{g}_i(q_i, \dot{q}_i, \hat{\theta}_{gi}) z_i \right\} \quad (4.16)$$

où ε_0 est une constante positive utilisée pour éviter la division par zéro lorsque $\hat{g}_i(q_i, \dot{q}_i) = 0$ et β_i est une autre constante positive, $z_i = b_i^T P_i e_i$ et P_i est solution de l'équation (4.6)

$$\Gamma_{ri} = \frac{1}{\underline{g}_i} \left(\rho_i (1 + |\Gamma_{fi}|) + |\Gamma_{0i}| \right) \text{sgn}(z_i) + \frac{1}{\underline{g}_i} \alpha_i z_i \quad (4.17)$$

où ρ_i et α_i sont des paramètres estimés en ligne, \underline{g}_i est une constante telle que $g_i \geq \underline{g}_i \geq 0$ et Γ_{0i} est donnée par :

$$\Gamma_{0i} = \frac{\varepsilon_0}{\varepsilon_0 + \hat{g}_i^2(q_i, \dot{q}_i, \hat{\theta}_{gi})} \left\{ v_i - \hat{f}_i(q_i, \dot{q}_i, \hat{\theta}_{fi}) + \beta_i \hat{g}_i(q_i, \dot{q}_i, \hat{\theta}_{gi}) z_i \right\} \quad (4.18)$$

Pour assurer les objectifs de la commande, on adopte les lois d'adaptation suivantes

$$\dot{\hat{\theta}}_{fi} = -\eta_{fi} w_{fi}(q_i, \dot{q}_i) z_i \quad (4.19)$$

$$\dot{\hat{\theta}}_{gi} = -\eta_{gi} w_{gi}(q_i, \dot{q}_i) z_i (\Gamma_{fi} - \beta_i z_i) \quad (4.20)$$

$$\dot{\rho}_i = \eta_{\rho i} |z_i| (1 + |\Gamma_{fi}|) \quad (4.21)$$

$$\dot{\alpha}_i = \eta_{\alpha i} z_i^2 \quad (4.22)$$

avec $\eta_{fi}, \eta_{gi}, \eta_{\rho i}$ et $\eta_{\alpha i} > 0$.

4.2.3 ANALYSE DE LA STABILITE

Considérons la fonction de Lyapunov candidate

$$V = \sum_{i=1}^n V_i \quad (4.23)$$

avec

$$V_i = e_i^T P_i e_i + \frac{1}{\eta_{fi}} \tilde{\theta}_{fi}^T \tilde{\theta}_{fi} + \frac{1}{\eta_{gi}} \tilde{\theta}_{gi}^T \tilde{\theta}_{gi} + \frac{1}{\eta_{\rho i}} \tilde{\rho}_i^2 + \frac{1}{\eta_{\alpha i}} \tilde{\alpha}_i^2 \quad (4.24)$$

où $\tilde{\theta}_{fi} = \theta_{fi}^* - \hat{\theta}_{fi}$, $\tilde{\theta}_{gi} = \theta_{gi}^* - \hat{\theta}_{gi}$, $\tilde{\rho}_i = \rho_i^* - \hat{\rho}_i$ et $\tilde{\alpha}_i = \alpha_i^* - \hat{\alpha}_i$

La dérivée par rapport au temps de (4.23) est

$$\dot{V}_i = \dot{e}_i^T P_i e_i + e_i^T P \dot{e}_i - \frac{2}{\eta_{fi}} \tilde{\theta}_{fi}^T \dot{\hat{\theta}}_{fi} - \frac{2}{\eta_{gi}} \tilde{\theta}_{gi}^T \dot{\hat{\theta}}_{gi} - \frac{2}{\eta_{\rho i}} \tilde{\rho}_i \dot{\hat{\rho}}_i - \frac{2}{\eta_{\alpha i}} \tilde{\alpha}_i \dot{\hat{\alpha}}_i \quad (4.25)$$

En utilisant la loi de commande (4.15), la dynamique de l'erreur (4.5) peut s'écrire sous la forme

$$\dot{e}_i = A_i e_i + b_i \left\{ v_i - f_i(q_i, \dot{q}_i) - (g_i(q_i, \dot{q}_i) - \hat{g}_i(q_i, \dot{q}_i, \hat{\theta}_{gi})) \Gamma_{fi} - \hat{g}_i(q_i, \dot{q}_i, \hat{\theta}_{gi}) \Gamma_{fi} - g_i(q_i, \dot{q}_i) \Gamma_{ri} - \Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) \right\} \quad (4.26)$$

en ajoutant et en retranchant $\beta_i \hat{g}_i(q_i, \dot{q}_i, \theta_{gi}^*)$ à (4.26), il vient

$$\dot{e}_i = A_i e_i + b_i \left\{ -w_{fi}^T(q_i, \dot{q}_i) \tilde{\theta}_{fi}^T - w_{gi}^T(q_i, \dot{q}_i) \tilde{\theta}_{gi}^T (\Gamma_{fi} - \beta_i z_i) - \beta_i \hat{g}_i(q_i, \dot{q}_i, \theta_{gi}^*) z_i + \Gamma_{0i} - \varepsilon_{fi}(q_i, \dot{q}_i) - \varepsilon_{gi}(q_i, \dot{q}_i) \Gamma_{fi} - g_i(q, \dot{q}) \Gamma_{ri} - \Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) \right\} \quad (4.27)$$

En remplaçant (4.27) dans (4.25) et en utilisant l'égalité $z_i = b_i^T P_i e_i$, il vient

$$\dot{V}_i = -e_i^T Q_i e_i - 2\beta_i \hat{g}_i(q_i, \dot{q}_i, \theta_{gi}^*) z_i^2 + \dot{V}_{i1} + \dot{V}_{i2} \quad (4.28)$$

avec

$$\dot{V}_{i1} = -2w_{fi}^T(q_i, \dot{q}_i) \tilde{\theta}_{fi}^T z_i - 2w_{gi}^T(q_i, \dot{q}_i) \tilde{\theta}_{gi}^T (\Gamma_{fi} - \beta_i z_i) z_i - \frac{2}{\eta_{fi}} \tilde{\theta}_{fi}^T \dot{\hat{\theta}}_{fi} - \frac{2}{\eta_{gi}} \tilde{\theta}_{gi}^T \dot{\hat{\theta}}_{gi} \quad (4.29)$$

$$\dot{V}_{i2} = 2z_i \left\{ \Gamma_{0i} - \varepsilon_{fi}(q_i, \dot{q}_i) - \varepsilon_{gi}(q_i, \dot{q}_i) \Gamma_{fi} - g_i(q_i, \dot{q}_i) \Gamma_{ri} - \Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) \right\} - \frac{2}{\eta_{\rho i}} \tilde{\rho}_i \dot{\hat{\rho}}_i - \frac{2}{\eta_{\alpha i}} \tilde{\alpha}_i \dot{\hat{\alpha}}_i \quad (4.30)$$

A partir des lois d'adaptations (4.19) et (4.20)

$$\dot{V}_{i1} = 0 \quad (4.31)$$

D'après (4.7)

$$\left| z_i \Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) \right| \leq \delta_{i0} |z_i| + \sum_{j=1}^n \delta_{i,j} |z_i| \|e_j\| \quad (4.32)$$

Puisque $0 \leq (2\bar{\alpha}a - b)^2$

alors l'inégalité suivante est toujours vérifiée

$$ab \leq \bar{\alpha}a^2 + \frac{1}{4\bar{\alpha}} b^2, \quad \bar{\alpha} > 0$$

L'équation (4.32) devient

$$\left| z_i \Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) \right| \leq \delta_{i0} |z_i| + \sum_{j=1}^n \left\{ \bar{\alpha}_j z_i^2 + \frac{\delta_{i,j}^2}{4 \bar{\alpha}_j} \|e_j\|^2 \right\} \quad (4.33)$$

En choisissant

$$\rho_i^* \geq \max(\bar{\varepsilon}_{fi} + \delta_{0,i}, \bar{\varepsilon}_{gi}) \text{ et } \alpha_i^* \geq \sum_{j=1}^n \bar{\alpha}_j$$

avec les lois d'adaptations (4.21) et (4.22), et le terme de commande robuste (4.17), l'équation (4.30) devient

$$\dot{V}_{i2} \leq \sum_{j=1}^n \frac{\delta_{i,j}^2}{2 \bar{\alpha}_j} \|e_j\|^2 \quad (4.34)$$

d'où la relation (4.28) peut être bornée par

$$\dot{V}_i \leq -e_i^T Q_i e_i - 2 \beta_i \hat{g}_i(q_i, \dot{q}_i, \theta_{gi}^*) z_i^2 + \sum_{j=1}^n \frac{\delta_{i,j}^2}{2 \bar{\alpha}_j} \|e_j\|^2 \quad (4.35)$$

De fait que $g_i(q_i, \dot{q}_i)$ est supposé positif, il est raisonnable de considérer que $\hat{g}_i(q_i, \dot{q}_i^*)$ l'est aussi

$$\dot{V}_i \leq -e_i^T Q_i e_i + \sum_{j=1}^n \frac{\delta_{i,j}^2}{2 \bar{\alpha}_j} \|e_j\|^2 \quad (4.36)$$

D'après (4.36) la dérivée de l'équation de Lyapunov (4.27) devient

$$\dot{V} \leq -\sum_{i=1}^n e_i^T Q_i e_i + \sum_{i=1}^n \frac{1}{2 \bar{\alpha}_j} \left(\sum_{j=1}^n \delta_{i,j}^2 \right) \|e_j\|^2 \leq -\sum_{i=1}^n \left\{ \lambda_{\min}(Q_i) - \frac{1}{2 \bar{\alpha}_i} \right\} \|e_i\|^2 \quad (4.37)$$

$\lambda_{\min}(Q_i)$ est la plus petite valeur propre de Q_i

en choisissons $\bar{\alpha}_i \geq \frac{1}{2(\lambda_{\min}(Q_i) - \lambda_{0i})} \sum_{j=1}^n \delta_{i,j}^2$, $\lambda_{0i} > 0$

l'équation (4.23) devient

$$\dot{V} \leq -\sum_{i=1}^n \lambda_{0i} \|e_i\|^2 < 0 \quad (4.38)$$

avec $\lambda_{0i} > 0$

Pour la simulation de cette technique de commande, on suppose que la fonction \underline{g}_i est inconnue. Dans ce cas, le terme de robustesse Γ_{ri} est modifié comme suit

$$\Gamma_{ri} = (\rho_i (1 + |\Gamma_{fi}|) + \rho_{\Gamma_i} |\Gamma_{0i}|) \operatorname{sgn}(z_i) + \alpha_i z_i \quad (4.39)$$

ρ_i est calculé par la loi d'adaptation suivante

$$\dot{\rho}_{\Gamma_i} = \eta_{\Gamma_i} |z_i| |\Gamma_{0i}| \quad (4.40)$$

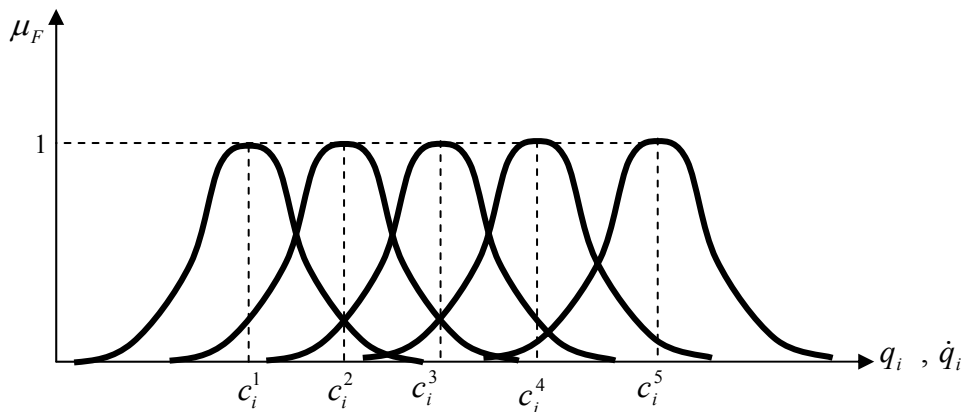
4.2.4 RESULTATS DE SIMULATION

Cette partie présente les résultats de simulation de la technique de commande adaptative floue décentralisée indirecte, décrite précédemment, pour la commande de la position et de l'orientation du robot PUMA 560 à six degrés de liberté.

Le système flou associé à chaque articulation i et ($i = 1, \dots, 6$) admet deux variables d'entrées (q_i et \dot{q}_i), où chacune est définie par cinq fonctions d'appartenances gaussiennes de la forme :

$$\mu_{F_i^j}(q_i) = \exp\left(-0.5 \left(\frac{q_i - c_i^j}{\sigma_i}\right)^2\right) \quad \text{où } j=1, \dots, 5$$

$$\mu_{F_i^j}(\dot{q}_i) = \exp\left(-0.5 \left(\frac{\dot{q}_i - \bar{c}_i^j}{\bar{\sigma}_i}\right)^2\right) \quad \text{où } j=1, \dots, 5$$



Les valeurs des paramètres des prémisses (c_i^j , σ_i et \bar{c}_i^j , $\bar{\sigma}_i$) sont données au tableau (4.1)

q_i	c_i^1	c_i^2	c_i^3	c_i^4	c_i^5	σ_i	\dot{q}_i	\bar{c}_i^1	\bar{c}_i^2	\bar{c}_i^3	\bar{c}_i^4	\bar{c}_i^5	$\bar{\sigma}_i$
q_1	0.3	0.42	0.55	0.68	0.8	0.05	\dot{q}_1	-0.28	-0.18	-0.08	1.02	0.12	0.04
q_2	0.16	0.37	0.58	0.79	1	0.08	\dot{q}_2	-0.3	-0.15	0	0.15	0.3	0.06
q_3	-0.2	0.15	0.5	0.85	1.2	0.14	\dot{q}_3	-0.5	-0.24	0.01	0.27	0.53	0.10
q_4	-1.75	-0.26	1.21	2.71	4.2	0.59	\dot{q}_4	-0.06	0.43	0.92	1.41	1.9	0.196
q_5	0.75	1.3	1.85	2.4	2.95	0.22	\dot{q}_5	-0.72	-0.34	0.04	0.42	0.8	0.15
q_6	-1	-0.38	0.24	0.86	1.48	0.24	\dot{q}_6	-1.25	-0.47	0.3	1.07	1.85	0.31

Tab.4.1 Valeurs des paramètres des prémisses

Les paramètres de synthèse sont choisis selon le tableau 4.2

\sum_i	k_i	P_i	β_i	η_{fi}	n_{gi}	η_{ci}	η_{pi}	$\eta_{\Gamma i}$
\sum_1	[5000 1000]	[150 50; 50 50]	10	1	1	0.001	0.001	0.001
\sum_2	[50000 500]	[150 20; 20 20]	10	1	1	0.001	0.001	0.001
\sum_3	[50000 300]	[10 5; 5 5]	10	1	1	0.001	0.001	0.001
\sum_4	[20 5]	[20 2; 2 2]	30	250	250	0.001	0.001	0.001
\sum_5	[20 30]	[15 5; 5 5]	10	250	250	0.001	0.001	0.001
\sum_6	[40 15]	[15 5; 5 5]	10	250	250	0.001	0.001	0.001

Tab.4.2 Paramètres de synthèse

Les résultats de simulations sont présentés aux figures 4.1-4.4

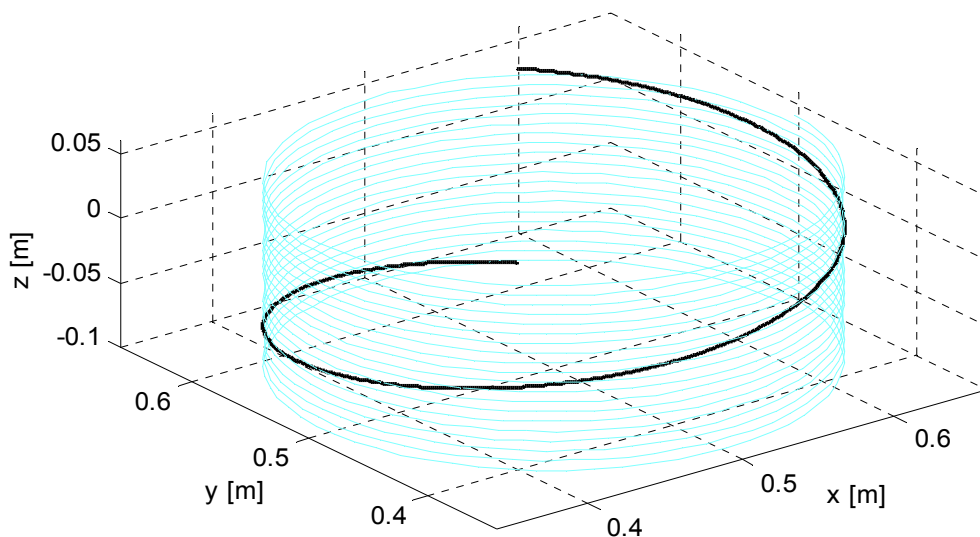


Fig.4.1.a Trajectoire de l'outil

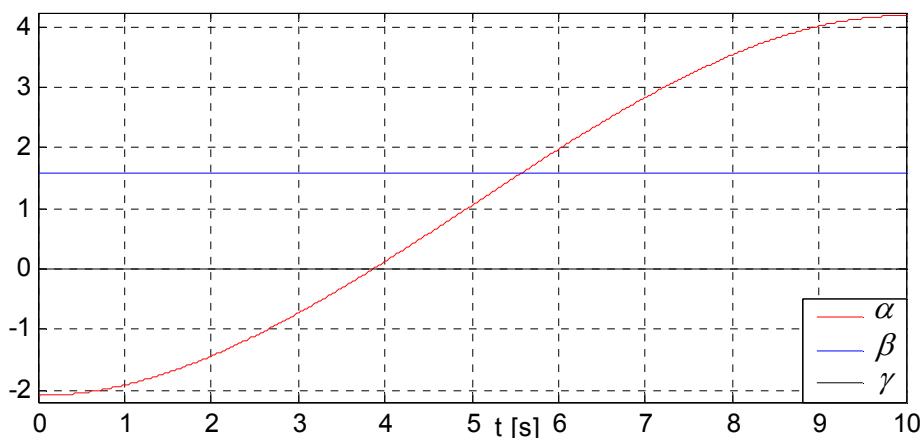


Fig.4.1.b Orientation de l'outil (unité : radian)

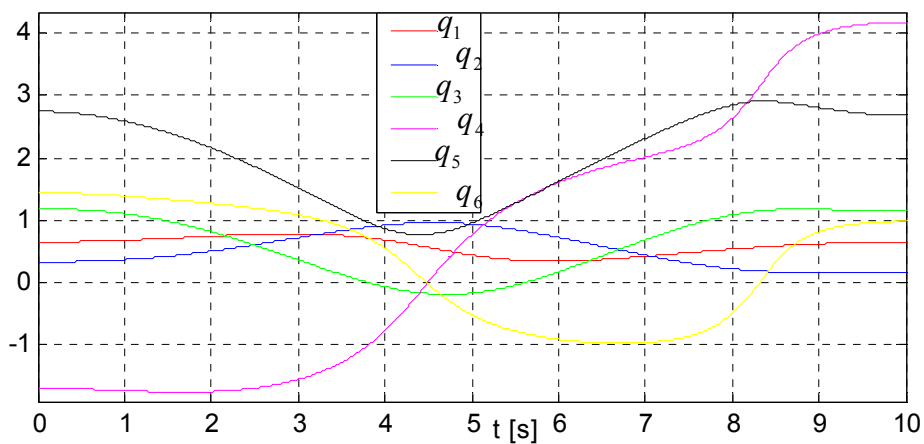


Fig.4.2 Trajectoires articulaires (unité : radian)

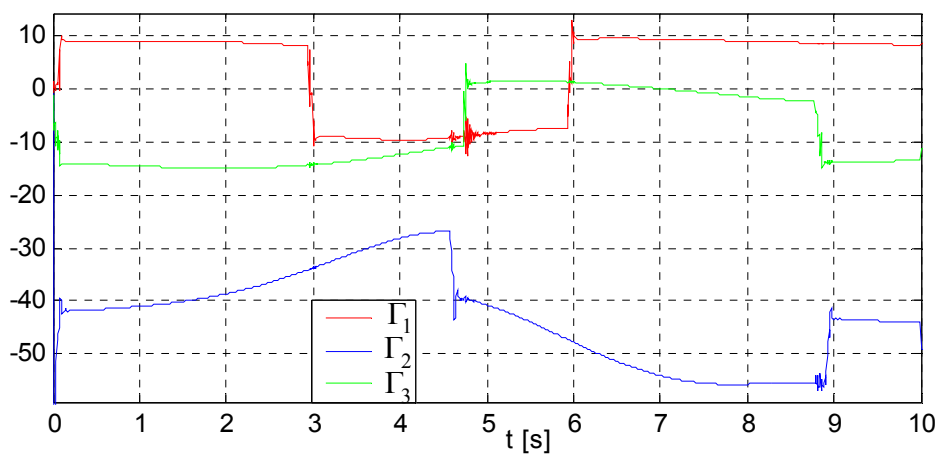


Fig.4.3.a couples appliqués aux trois premières articulations (unité : N.m)

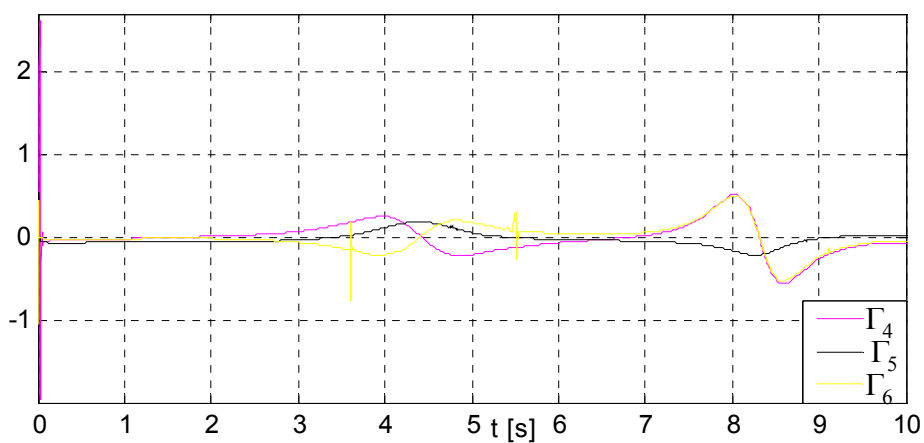


Fig.4.3.b Couples appliqués aux trois dernières articulations (unité : N.m)

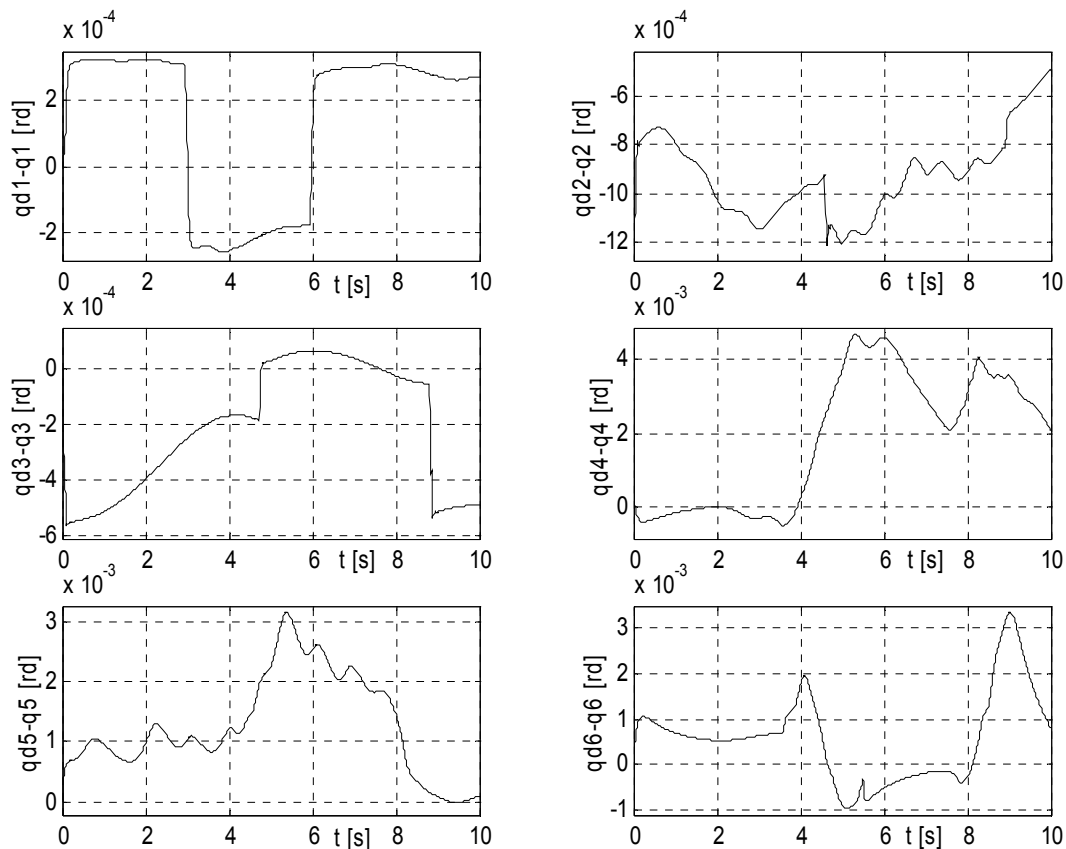


Fig.4.4.a Erreurs articulaires

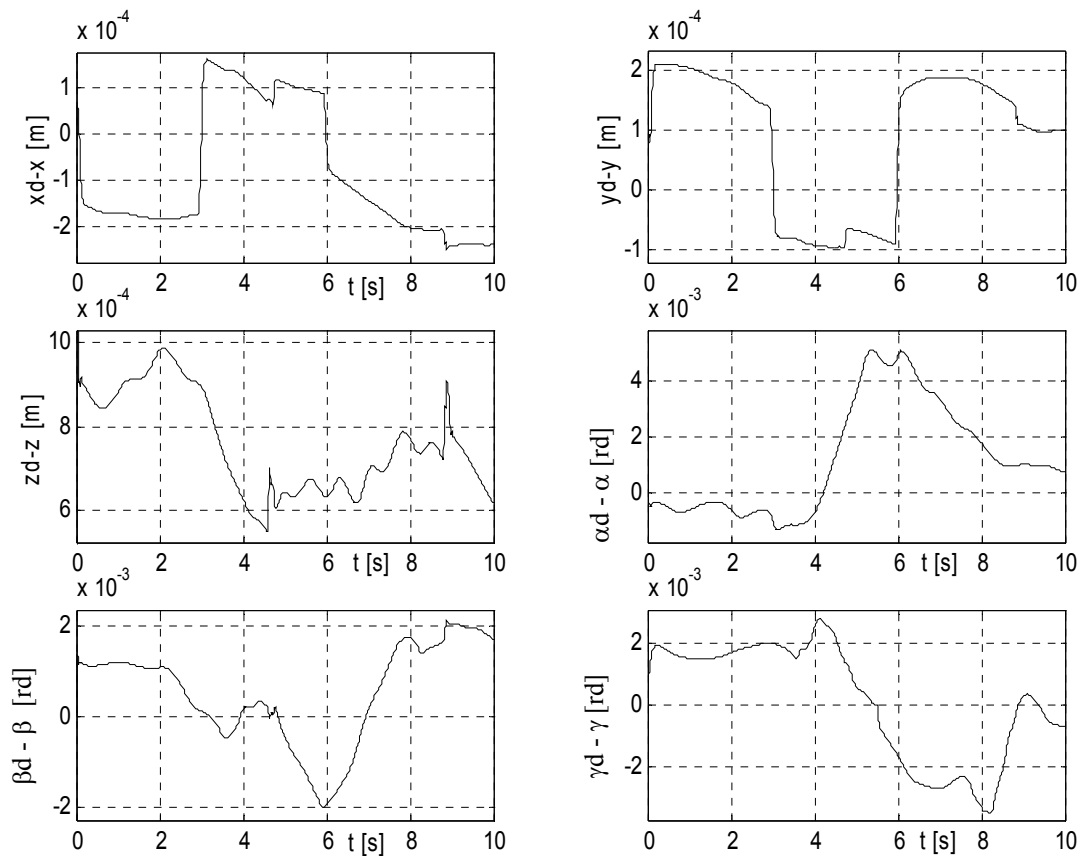


Fig.4.4.b Erreurs cartésiennes

▪ Robustesse vis-à-vis des perturbations extérieures

Les figures 4.5-4.4.7 illustrent les résultats de simulation en appliquant au robot une charge de 2 kg à l'instant $t = 5$ secondes.

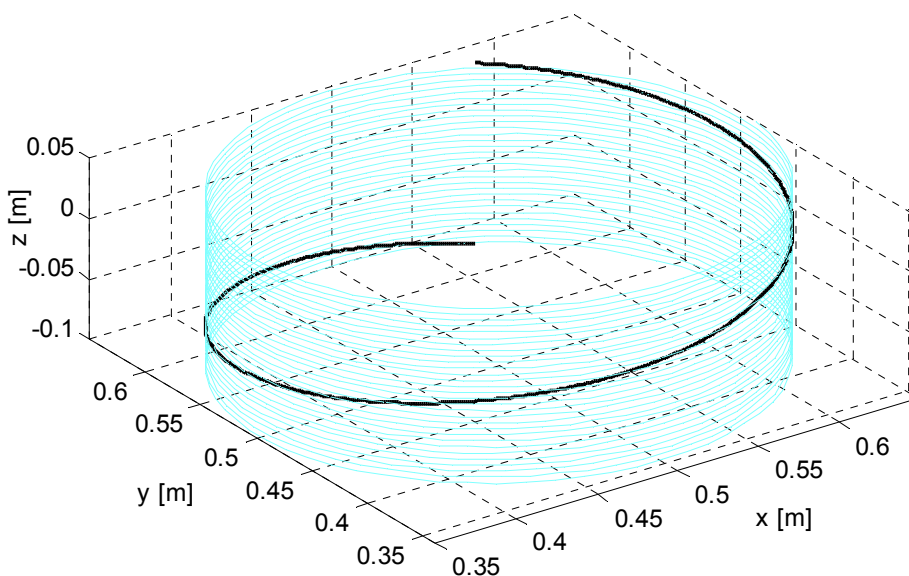


Fig.4.5.a Trajectoire de l'outil

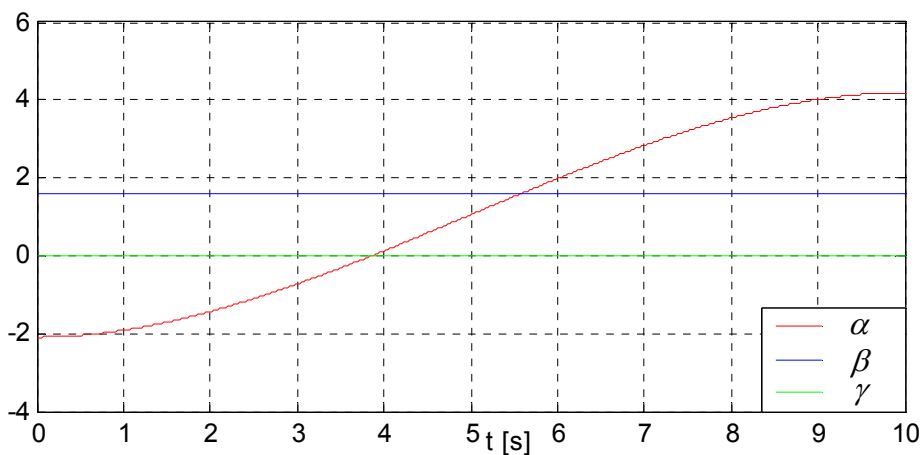


Fig.4.5.b Orientation de l'outil (unité : radian)

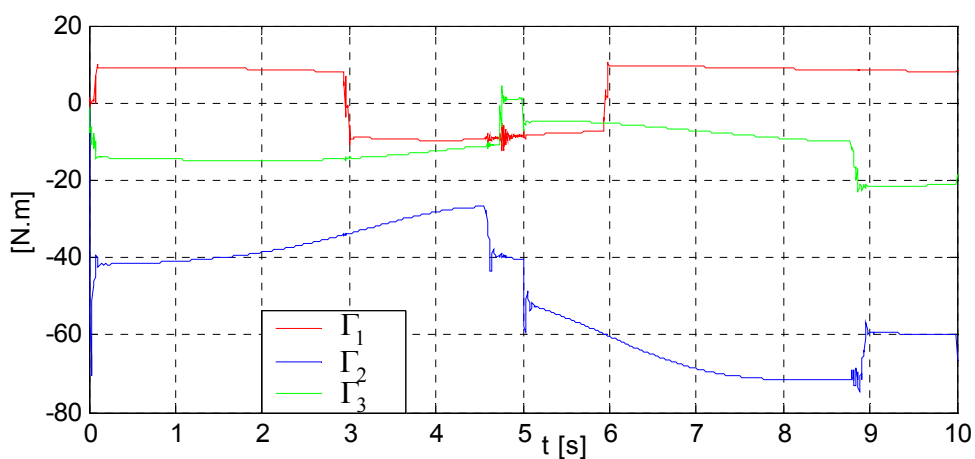


Fig.4.6.a Couples appliqués aux trois premières articulations (unité : N.m)

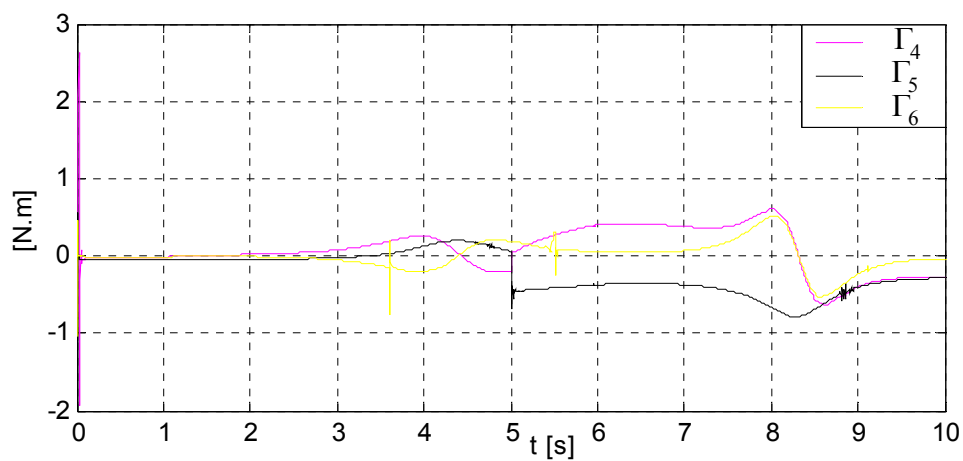


Fig.4.6.b Couples appliqués aux trois dernières articulations (unité : N.m)

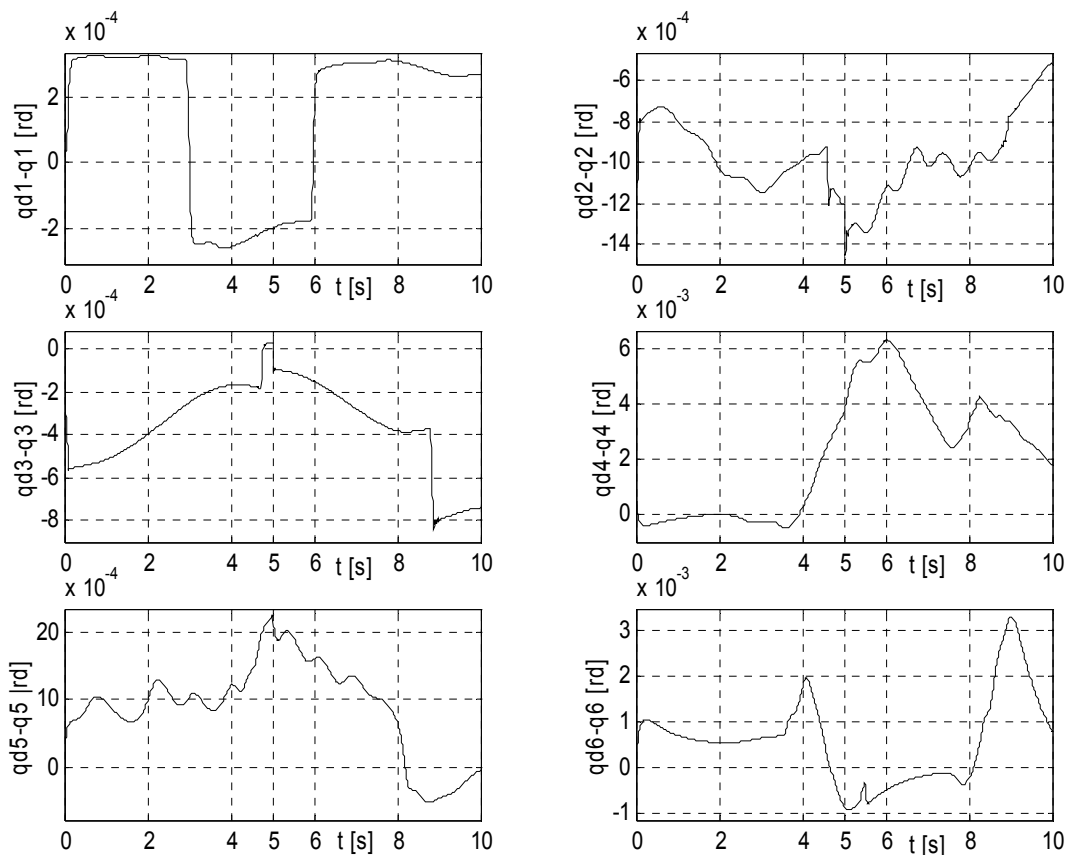


Fig.4.7.a Erreurs articulaires

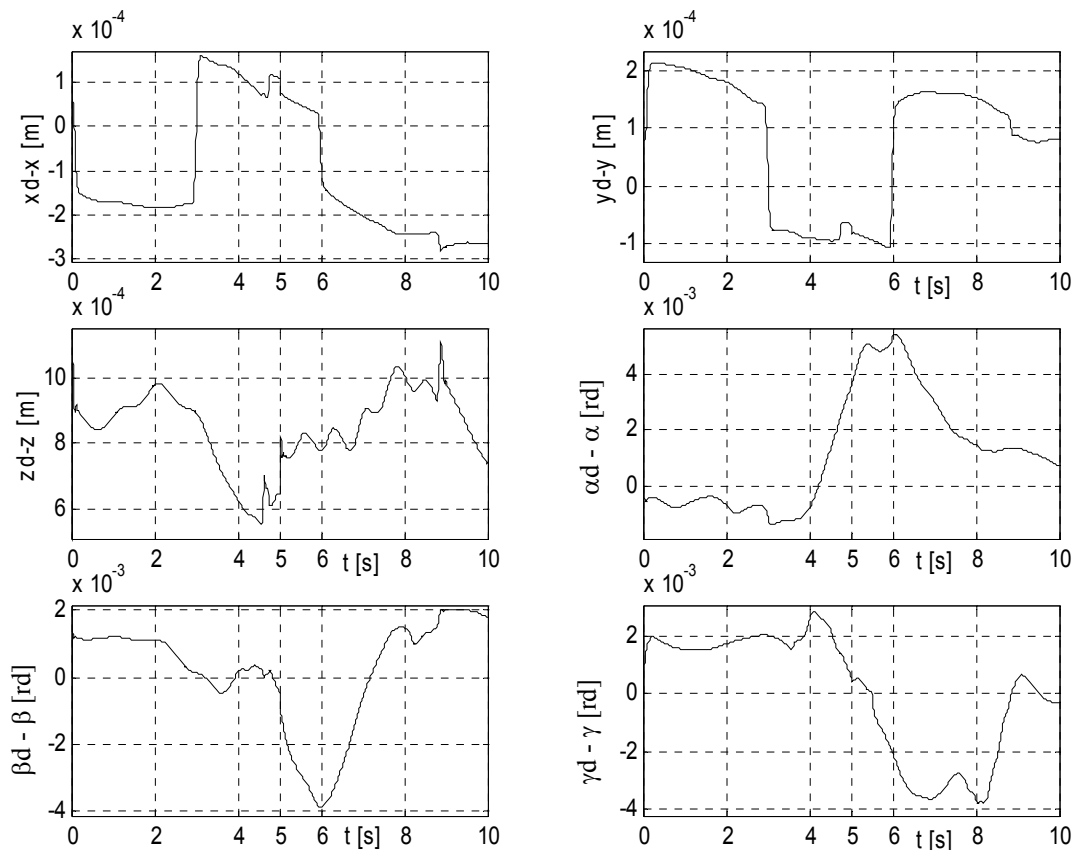


Fig.4.7.b Erreurs cartésiennes

▪ **Robustesse de la loi de commande vis-à-vis des variations des paramètres du robot**

Les figures 4.8-4.10 montrent les résultats de simulation en soumettant le robot à une variation de 100% de la matrice d'inertie.

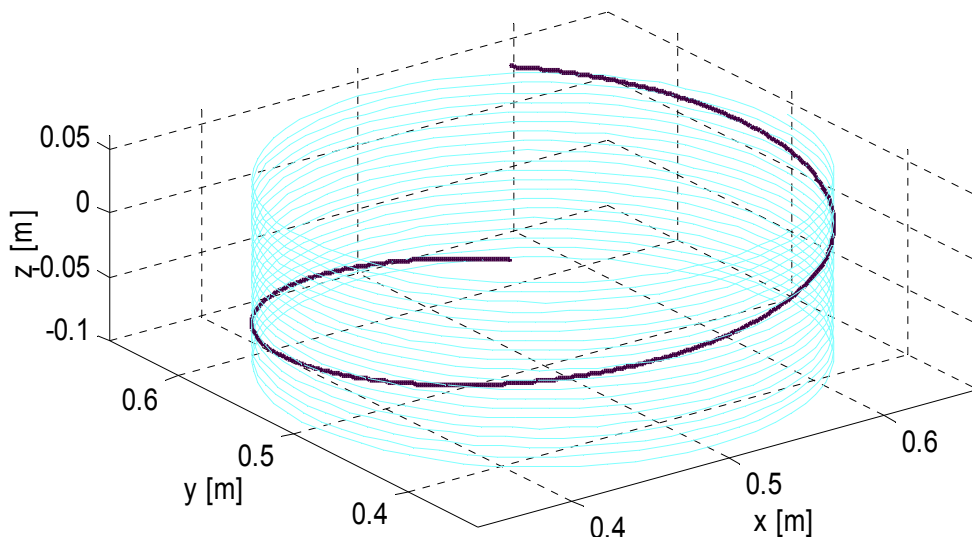


Fig.4.8.a Trajectoire de l'outil

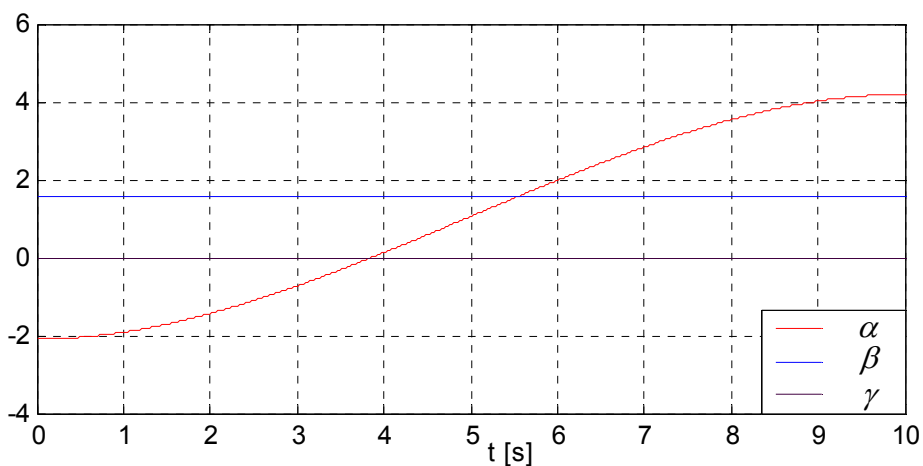


Fig.4.8.b Orientation de l'outil (unité : radian)

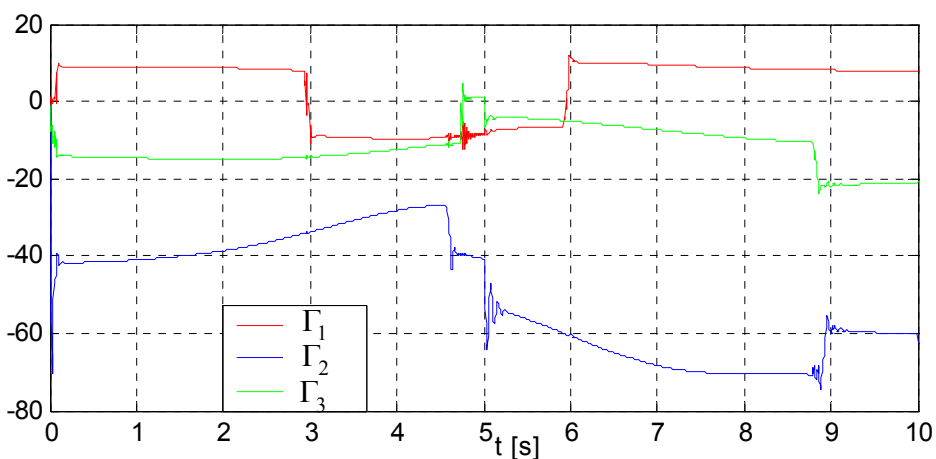


Fig.4.9.a Couples développés par les trois premiers actionneurs (unité : N.m)

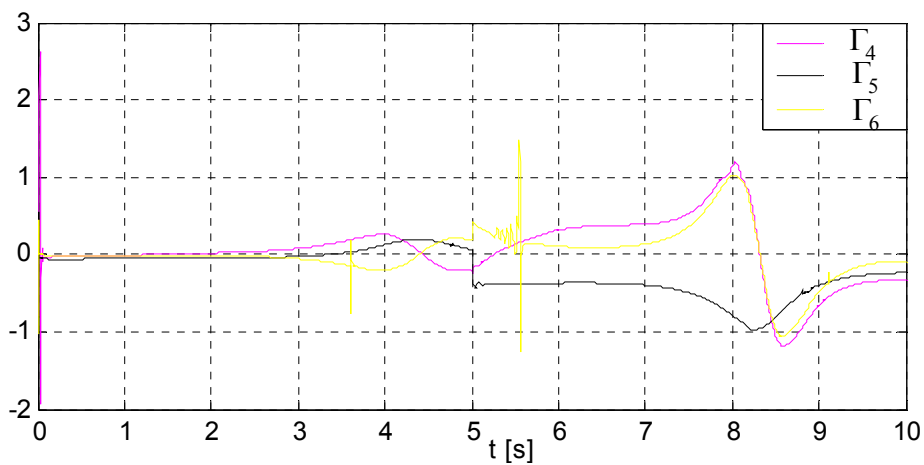


Fig.4.9.b Couples développés par les trois derniers actionneurs (unité : N.m)

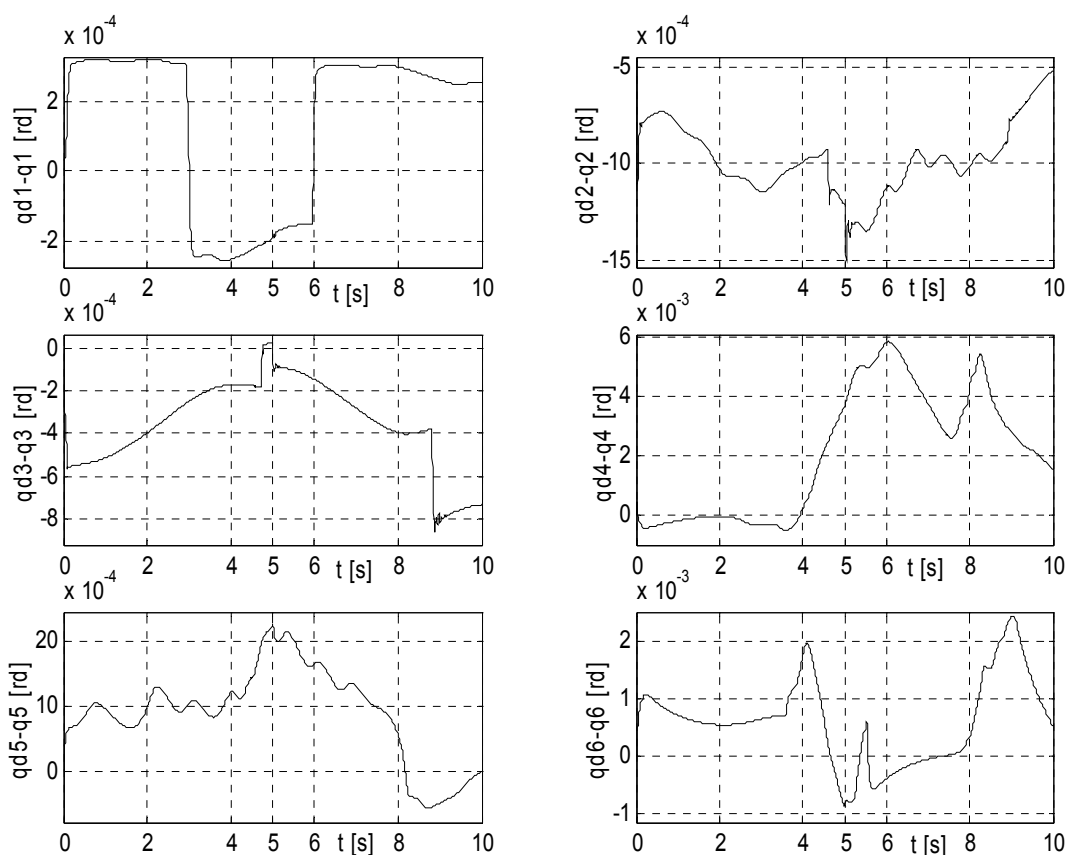


Fig.4.10.a Erreurs articulaires

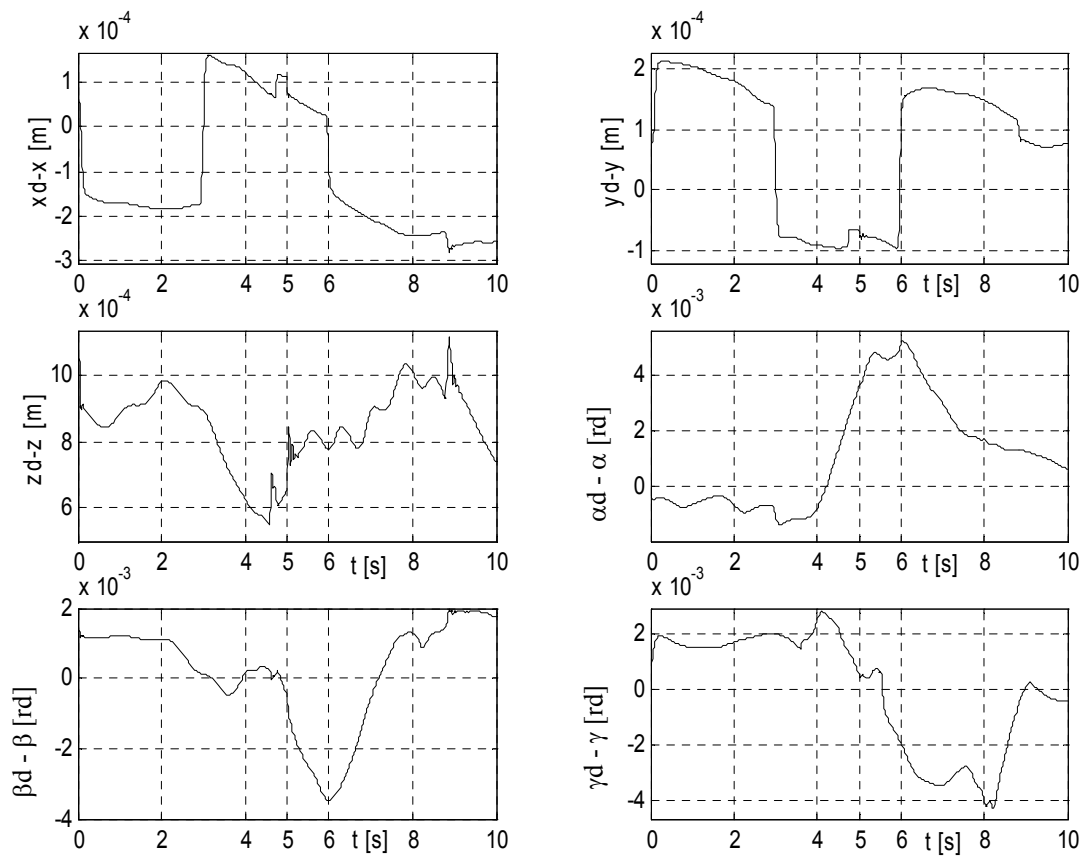


Fig.4.10.b Erreurs cartésiennes

	Tests sans perturbations	Tests avec charge	Tests avec perturbations paramétriques
$x^d - x [mm]$	0.24	0.15	0.27
$y^d - y [mm]$	0.21	0.21	0.21
$z^d - z [mm]$	1.10	1.10	1.10
$\alpha^d - \alpha [rd]$	5.1×10^{-3}	5.4×10^{-3}	5.2×10^{-3}
$\beta^d - \beta [rd]$	2.1×10^{-3}	3.2×10^{-3}	3.5×10^{-3}
$\gamma^d - \gamma [rd]$	3.5×10^{-3}	3.8×10^{-3}	4.3×10^{-3}

Tab.4.3 Erreurs maximales dans l'espace de la tâche

	Tests sans perturbations	Tests avec charge	Tests avec perturbations paramétriques
$\Gamma_1 [N.m]$	12.77	12.61	12.61
$\Gamma_2 [N.m]$	70.52	74.40	74.32
$\Gamma_3 [N.m]$	15.11	23.14	23.53
$\Gamma_4 [N.m]$	2.62	2.62	2.62
$\Gamma_5 [N.m]$	1.08	1.08	1.08
$\Gamma_6 [N.m]$	1.03	1.03	1.48

Tab.4.4 Couples maximaux développés par les actionneurs

La réponse du robot dans l'espace des tâches est illustrée par les figures 4.1.a, 4.10. Celle-ci montrent que malgré la réduction du nombre de règles nécessaires à la construction des systèmes flous par rapport aux approches centralisées, la loi de commande assure une bonne poursuite en position/orientation.

Les oscillations qui apparaissent sur les amplitudes des couples de commande (fig.4.3.b, 4.6.b et 4.9.b) au début de la tâche sont dues à l'initialisation des paramètres des systèmes flous, un bon choix de ces paramètres permet de contourner ce problème.

Les valeurs faibles des erreurs cartésiennes maximales (Tab.4.3) et les amplitudes des couples maximaux développés par chaque actionneur (Tab.4.4), confirment la robustesse de la loi de commande adaptative floue décentralisée indirecte vis-à-vis respectivement de l'application de la charge et des perturbations paramétriques.

4.3 COMMANDE ADAPTATIVE DECENTRALISEE FLOUE DIRECTE

L'approche adaptative décentralisée floue directe se basent sur l'approximation par des modèles flous des lois de commandes pour chaque sous système en utilisant uniquement les informations locales.

Considérons un robot manipulateur à n degrés de liberté dont le modèle est

$$\ddot{q} = F(q, \dot{q}) + G(q)\Gamma \quad (4.41)$$

où les variables q, \dot{q} , \ddot{q} , $f(q, \dot{q})$, $g(q)$ et Γ ont été définies dans la section précédente.

Ce système est composé de six sous-systèmes \sum_i non linéaires donnés par

$$\sum_i \Rightarrow \ddot{q}_i = f_i(q_i, \dot{q}_i) + g_i(q_i, \dot{q}_i)\Gamma_i + \Delta_i(q_i, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) \quad (4.42)$$

où q_i est la sortie du sous-système \sum_i et $i=1, \dots, n$

Les variables d'état q_i et \dot{q}_i du sous-système \sum_i sont supposées mesurables et Γ_i est le couple actionneur de la $i^{\text{ème}}$ articulation (entrée de commande). Les fonctions non linéaires $f_i(q_i, \dot{q}_i)$ et $g_i(q_i, \dot{q}_i)$ sont non linéaires continues incertaines, et le terme $\Delta_i(q_i, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n)$ représente l'effet des interconnexions entre le sous système \sum_i et les autre sous-systèmes $\sum_{j=1, j \neq i}^n$.

Pour chaque sous-système \sum_i , on définit l'erreur de poursuite

$$e_i(t) = q_i^d(t) - q_i(t)$$

et la surface de l'erreur de poursuite

$$S_i = \left(\frac{\partial}{\partial t} + \lambda_i \right) e_i, \lambda_i > 0 \quad (4.43)$$

ou, encore

$$S_i = \dot{e}_i + \lambda_i e_i \quad (4.44)$$

de la relation (4.44), on a

$$\dot{e}_i = -\lambda_i e_i + s_i \quad (4.45)$$

La dérivée temporelle de la surface de l'erreur est donnée par

$$\dot{s}_i = v_i - f_i(q_i, \dot{q}_i) - g_i(q_i, \dot{q}_i)\Gamma_i - \Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) \quad (4.46)$$

avec

$$v_i = \ddot{q}_i^d + \lambda_i \dot{e}_i$$

On suppose que les interconnexions satisfont

$$\left| \Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) \right| \leq g_i(q_i, \dot{q}_i) \left\{ \delta_{i,0} + \sum_{j=1}^n \delta_{i,j} (|e_j| + |s_j|) \right\} \quad (4.47)$$

avec $\delta_{i,0}, \dots, \delta_{i,N}$ sont des constantes positives inconnues.

et que les fonctions $g_i(q_i, \dot{q}_i)$ et $\dot{g}_i(q_i, \dot{q}_i)$ vérifient les conditions :

$$g_i(q_i, \dot{q}_i) \geq \bar{g}_i > 0$$

$$\left| \dot{g}_i(q_i, \dot{q}_i) \right| \leq D_i(q_i, \dot{q}_i)$$

où $D_i(q_i, \dot{q}_i)$ est une fonction positive continue.

Si on considère que les sous systèmes sont isolés l'un de l'autre et que les fonctions $f_i(q_i, \dot{q}_i)$, $g_i(q_i, \dot{q}_i)$ et $D_i(q_i, \dot{q}_i)$ sont parfaitement connues, alors la loi de commande suivante assure les objectifs de commande

$$\Gamma_i = \Gamma_i^* = k_i s_i + e_i + \frac{1}{g_i(q_i, \dot{q}_i)} (v_i - f_i(q_i, \dot{q}_i)) + \frac{D_i(q_i, \dot{q}_i)}{2 g_i^2(q_i, \dot{q}_i)} s_i \quad (4.48)$$

avec $k_i > 0$

Considérons la fonction de Lyapunov candidate suivante :

$$V_i = \frac{1}{2} e_i^2 + \frac{1}{2 g_i(q_i, \dot{q}_i)} s_i^2 \quad (4.49)$$

En dérivant (4.49) par rapport au temps en utilisant les équations (4.45), (4.46) et (4.48), on obtient

$$\dot{V}_i \leq -\lambda_i e_i^2 - k_i s_i^2 \leq 0$$

D'où les erreurs $e_i(t)$ et $s_i(t)$ convergent asymptotiquement vers zéro.

Dans le domaine de la robotique les fonctions $f_i(q_i, \dot{q}_i)$, $g_i(q_i, \dot{q}_i)$, $D(q_i, \dot{q}_i)$ et $\Delta_i(q_i, \dots, q_6, \dot{q}_1, \dots, \dot{q}_n)$ sont inconnues ou mal connues, l'application de la loi de commande (4.48) est très difficile.

L'objectif de cette section est d'approcher cette loi de commande par des systèmes flous.

4.3.1 SYNTHÈSE DE LA LOI DE COMMANDE

Supposons que chaque commande idéale (4.48) peut être estimée par les systèmes flous de la manière suivante :

$$\hat{\Gamma}_i^* = k_i s_i + e_i + w_i(q_i, \dot{q}_i) \hat{\theta}_i \quad (4.50)$$

où $\hat{\theta}_i$ est le vecteur des paramètres ajusté et $w_i(q_i, \dot{q}_i)$ est un vecteur de fonctions floues de base

La fonction réelle Γ_i^* peut être exprimée alors de la façon suivante :

$$\Gamma_i^*(q_i, \dot{q}_i, \theta_i^*) = k_i s_i + e_i + w_i^T(q_i, \dot{q}_i) \theta_i^* + \varepsilon_i(q_i, \dot{q}_i) \quad (4.51)$$

avec $\varepsilon_i(q_i, \dot{q}_i)$ sont les erreurs d'approximation, θ_i^* sont les paramètres optimaux de $\hat{\theta}_i$ minimisant $|\varepsilon_i(q_i, \dot{q}_i)|$

les erreurs d'approximation sont bornées telles que :

$$|\varepsilon_i(q_i, \dot{q}_i)| \leq \bar{\varepsilon}_i$$

où $\bar{\varepsilon}_i$ est un paramètre positif inconnu.

L'erreur d'identification de la commande idéale peut s'exprimer de la manière suivante :

$$\Gamma_i^* - \hat{\Gamma}_i^*(q_i, \dot{q}_i, \hat{\theta}_i) = w_i^T(q_i, \dot{q}_i) \tilde{\theta}_i + \varepsilon_i(q_i, \dot{q}_i) \quad (4.52)$$

avec $\tilde{\theta}_i = \theta_i^* - \hat{\theta}_i$

Chaque sous-système est commandé par la loi de commande locale donnée par :

$$\Gamma_i = \hat{\Gamma}_i(q_i, \dot{q}_i, \hat{\theta}_i) + \Gamma_{ri} = k_i s_i + e_i + w_i(q_i, \dot{q}_i) \hat{\theta}_i + \Gamma_{ri} \quad (4.53)$$

avec Γ_{ri} est le terme de commande robuste donné sous la forme :

$$\Gamma_{ri} = \rho_i \operatorname{sgn}(s_i) + \alpha_i s_i \quad (4.54)$$

avec ρ_i et α_i sont des paramètres estimés en ligne par des lois d'adaptations.

Pour assurer les objectifs de la commande, sont utilisées les lois d'adaptation suivantes

$$\dot{\hat{\theta}}_i = \eta_i w(q_i, \dot{q}_i) s_i \quad (4.55)$$

$$\dot{\rho}_i = \eta_{\rho i} |s_i| \quad (4.56)$$

$$\dot{\alpha}_i = \eta_{\alpha i} s_i^2 \quad (4.57)$$

avec $\eta_i, \eta_{\rho i}$ et $\eta_{\alpha i} > 0$

4.3.2 ANALYSE DE LA STABILITE

L'équation dynamique de l'erreur (4.46) peut être exprimée comme suit

$$\dot{s}_i = v_i - f_i(q_i, \dot{q}_i) - g_i(q_i, \dot{q}_i) \Gamma_i^* + g_i(q_i, \dot{q}_i) (\Gamma_i^* - \Gamma_i) - \Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) \quad (4.58)$$

par substitution de (4.48) dans (4.58), il vient

$$\dot{s}_i = g_i(q_i, \dot{q}_i) \left\{ -k_i s_i - e_i - \frac{D_i(q_i, \dot{q}_i)}{2g_i^2(q_i, \dot{q}_i)} s_i + (\Gamma_i^* - \Gamma_i) - \frac{\Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n)}{g_i(q_i, \dot{q}_i)} \right\} \quad (4.59)$$

de (4.52) et (4.53), il vient

$$\begin{aligned} \frac{\dot{s}}{g_i(q_i, \dot{q}_i)} &= -k_i s_i - e_i - \frac{D_i(q_i, \dot{q}_i)}{2g_i^2(q_i, \dot{q}_i)} s_i + w_i^T(q_i, \dot{q}_i) \tilde{\theta}_i \\ &\quad - \Gamma_{ri} + \varepsilon_i(q_i, \dot{q}_i) - \frac{\Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n)}{g_i(q_i, \dot{q}_i)} \end{aligned} \quad (4.60)$$

Considérons la fonction de Lyapunov candidate suivante

$$V = \sum_{i=1}^n V_i \quad (4.61)$$

avec

$$V_i = \frac{1}{2} e^2 + \frac{1}{2g_i(q_i, \dot{q}_i)} s_i^2 + \frac{1}{2\eta_i} \tilde{\theta}_i^T \tilde{\theta}_i + \frac{1}{2\eta_{\rho i}} \tilde{\rho}_i^2 + \frac{1}{2\eta_{\alpha i}} \tilde{\alpha}_i^2 \quad (4.62)$$

La dérivée de (5.62) par rapport au temps est :

$$\dot{V}_i = e \dot{e} + \frac{1}{g_i(q_i, \dot{q}_i)} s_i \dot{s}_i - \frac{\dot{g}_i(q_i, \dot{q}_i)}{2g_i^2(q_i, \dot{q}_i)} s_i^2 - \frac{1}{\eta_i} \tilde{\theta}_i^T \dot{\hat{\theta}}_i + \frac{1}{\eta_{\rho_i}} \tilde{\rho}_i \dot{\hat{\rho}}_i - \frac{1}{\eta_{\alpha_i}} \tilde{\alpha}_i \dot{\hat{\alpha}}_i \quad (4.63)$$

A partir de (4.45) et (4.60), il vient

$$\begin{aligned} \dot{V}_i \leq & -\lambda_i e_i^2 - k_i s_i^2 + s_i w_i^T(q_i, \dot{q}_i) \tilde{\theta}_i - s_i \Gamma_{r_i} + s_i \varepsilon_i(q_i, \dot{q}_i) - \\ & s_i \frac{\Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n)}{g_i(q_i, \dot{q}_i)} - \frac{1}{\eta_i} \tilde{\theta}_i^T \dot{\hat{\theta}}_i - \frac{1}{\eta_{\rho_i}} \tilde{\rho}_i \dot{\hat{\rho}}_i - \frac{1}{\eta_{\alpha_i}} \tilde{\alpha}_i \dot{\hat{\alpha}}_i \end{aligned} \quad (4.64)$$

En remplaçant Γ_{r_i} par son expression (4.54) et en utilisant les loi d'adaptations (4.55), (4.56) et (4.57), la relation (4.47) devient

$$\dot{V}_i \leq -\lambda_i e_i^2 - k_i s_i^2 + s_i \varepsilon_i(q_i, \dot{q}_i) - s_i \frac{\Delta_i(q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n)}{g_i(q_i, \dot{q}_i)} - \rho_i^* |s_i| - \alpha_i^* s_i^2 \quad (4.65)$$

Par substitution de (4.47) dans (4.64), il vient

$$\dot{V}_i \leq -\lambda_i e_i^2 - k_i s_i^2 + |s_i| \bar{\varepsilon}_i + \delta_{i,0} |s_i| + \sum_{j=1}^n \left\{ \bar{\alpha}_j s_i^2 + \frac{\delta_{i,j}^2}{2 \bar{\alpha}_j} (|e_j|^2 + |s_j|^2) \right\} - \rho_i^* |s_i| - \alpha_i^* s_i^2 \quad (4.66)$$

En choisissant

$$\rho_i^* \leq \bar{\varepsilon}_i + \delta_{i,0}$$

La fonction (4.66) devient

$$\dot{V}_i \leq -\lambda_i e_i^2 - k_i s_i^2 + \sum_{j=1}^n \left\{ \bar{\alpha}_j s_i^2 + \frac{\delta_{i,j}^2}{2 \bar{\alpha}_j} (|e_j|^2 + |s_j|^2) \right\} - \alpha_i^* s_i^2 \quad (4.67)$$

et la dérivée de la fonction de lyapunov (4.61) est bornée par

$$\dot{V} \leq -\sum_{i=1}^n \left\{ k_i s_i^2 + \left(\lambda_i - \frac{1}{2 \bar{\alpha}_i} \sum_{j=1}^n \delta_{j,i}^2 \right) |e_i|^2 \right\} + \sum_{i=1}^n \left\{ \left(\sum_{j=1}^n \bar{\alpha}_j + \frac{1}{2 \bar{\alpha}_i} \sum_{j=1}^n \delta_{j,i}^2 \right) - \alpha_i^* \right\} s_i^2 \quad (4.68)$$

il suffit de choisir

$$\bar{\alpha}_i \geq \frac{1}{2(\lambda_i - \lambda_{0,i})} \sum_{j=1}^n \delta_{j,i}^2 \quad (4.69)$$

avec $\lambda_{0,i} > 0$

$$\text{et } \alpha_i^* \geq \sum_{j=1}^n \bar{\alpha}_j + \frac{1}{2} \frac{1}{\bar{\alpha}_i} \sum_{j=1}^n \delta_{j,i}^2 \quad (4.70)$$

pour avoir

$$\dot{V} \leq -\sum_{i=1}^n \left\{ k_i s_i^2 + \lambda_{0,i} |e_i|^2 \right\} \leq 0 \quad (4.71)$$

ce qui implique la bornitude de tous les signaux dans le système bouclé.

4.3.3 RESULTATS DE SIMULATION

Dans cette section, nous présentons les résultats de simulation relative à la commande floue décentralisée directe au robot PUMA 560.

Six systèmes flous sont utilisés pour générer les commandes $\Gamma_1, \dots, \Gamma_6$. Chaque variable d'entrée (q_i et \dot{q}_i) est définie par cinq fonctions d'appartenances de la même manière que pour l'approche indirecte (Tab.4.1)

Les paramètres de synthèse sont imposés selon le tableau.4.5

\sum_i	k_i	λ_i	η_i	$\eta_{\rho i}$	$\eta_{\alpha i}$
\sum_1	600	100	1	0.01	0.01
\sum_2	600	100	1	0.01	0.01
\sum_3	400	100	1	0.01	0.01
\sum_4	20	80	1	0.01	0.01
\sum_5	20	80	1	0.01	0.01
\sum_6	20	80	1	0.01	0.01

Tab.4.5 Paramètres de synthèse

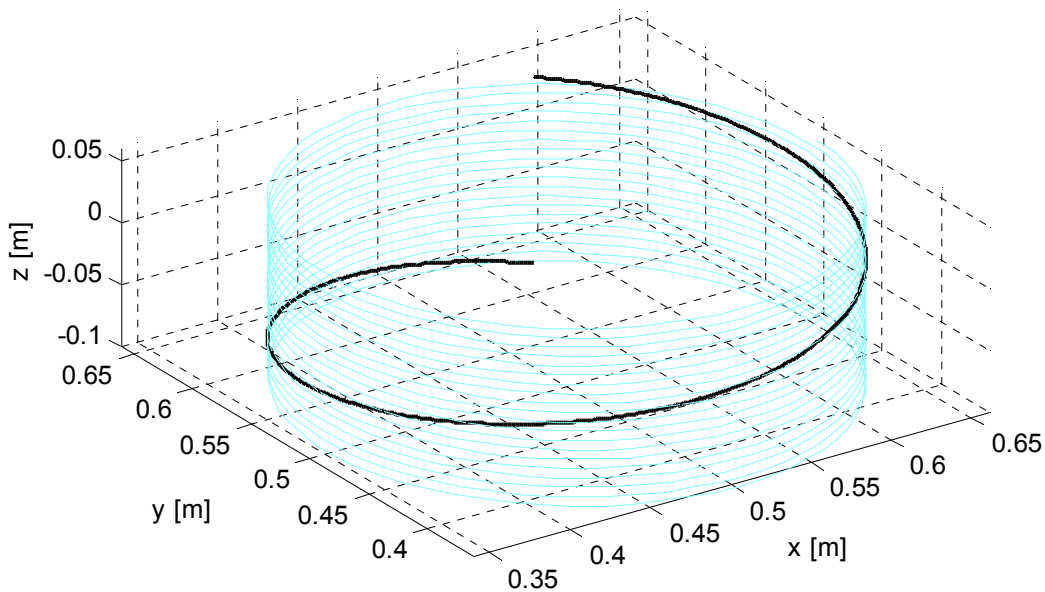


Fig.4.11.a Trajectoire de l'outil

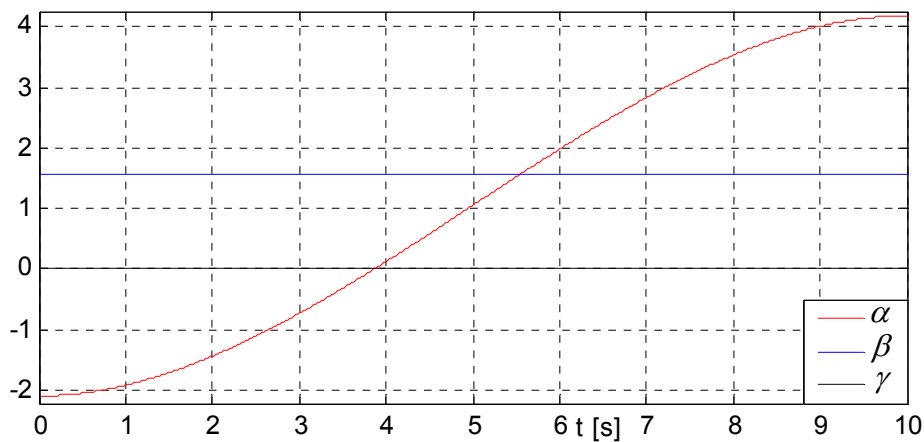


Fig.4.11.b Orientation de l'outil (unité : radian)

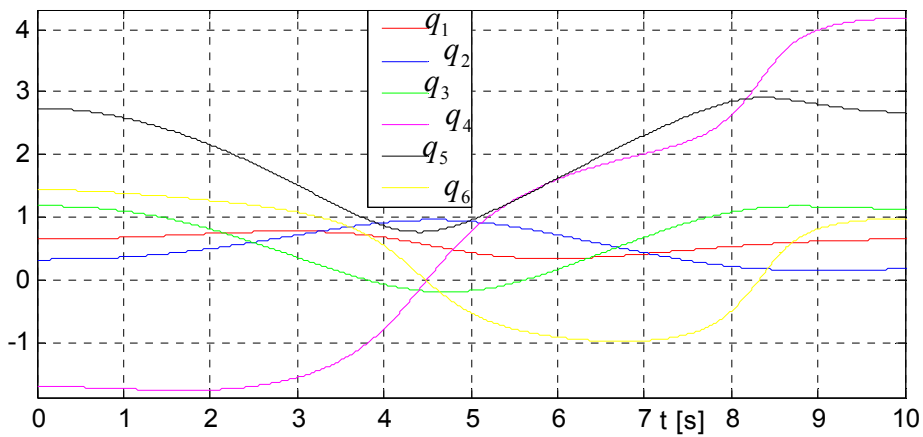


Fig.4.12 Trajectoires articulaires (unité : radian)

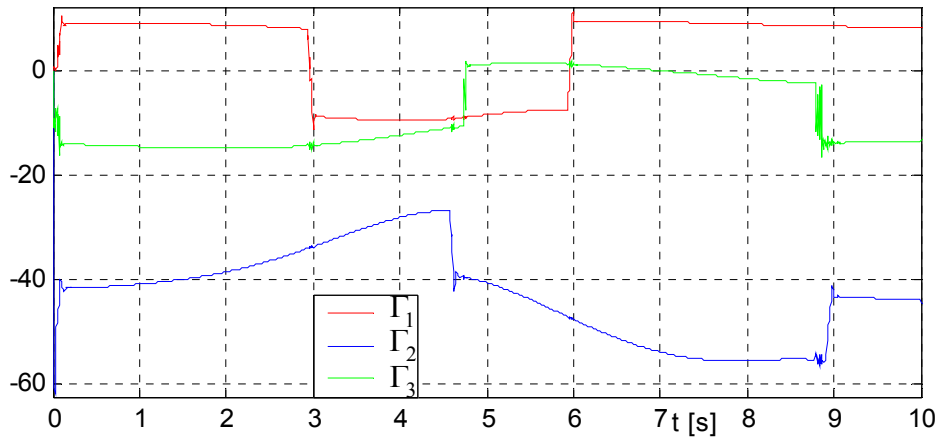


Fig.4.13.a Couples actionneurs (unité : N.m)

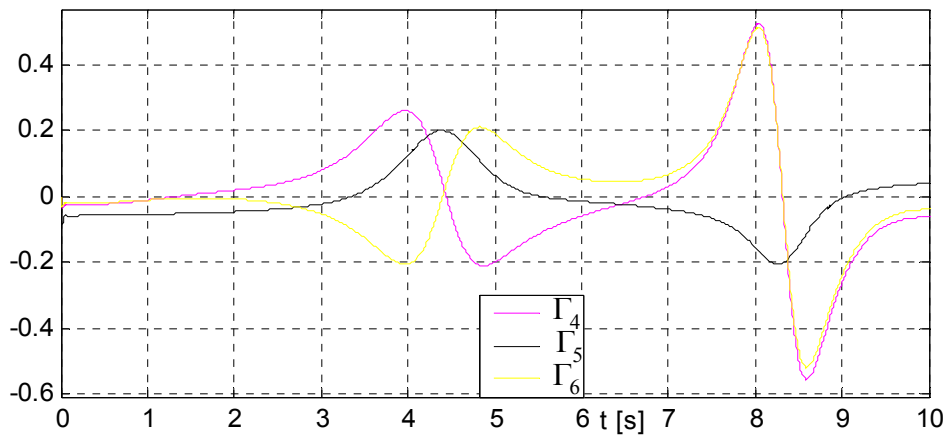


Fig.4.13.b Couples actionneurs (unité : N.m)

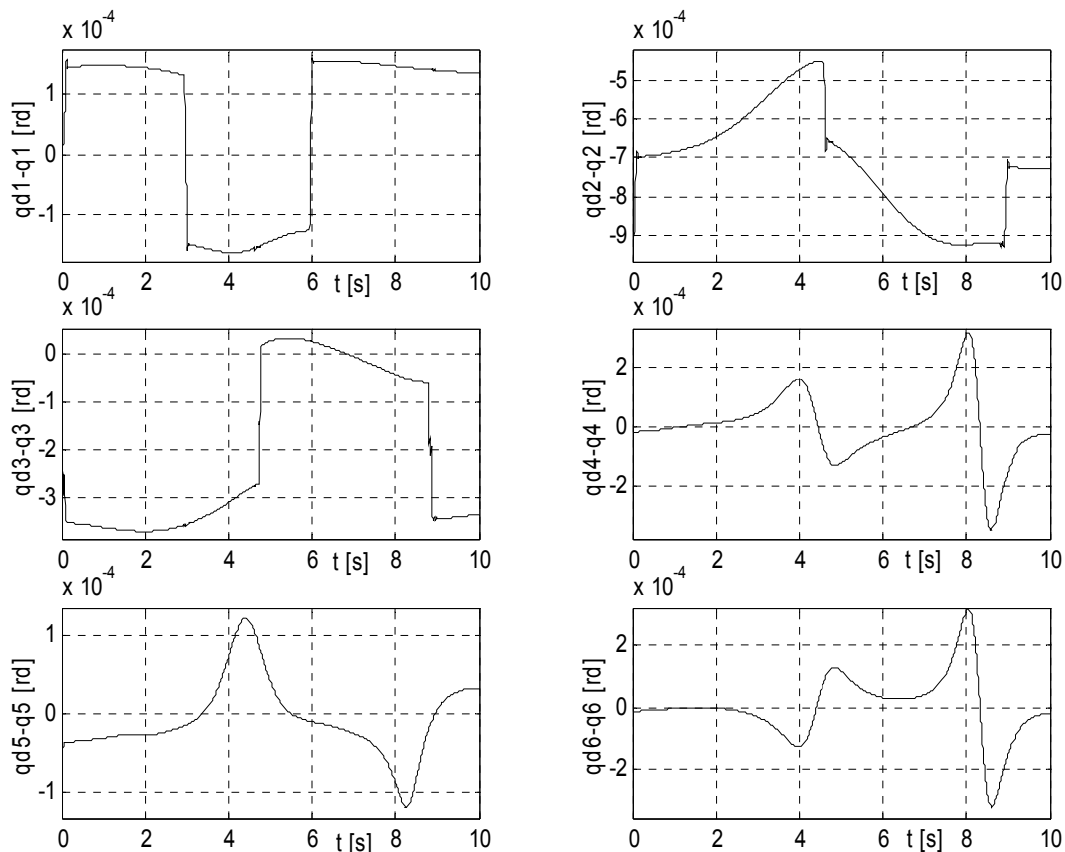


Fig.4.14.a Erreurs articulaires

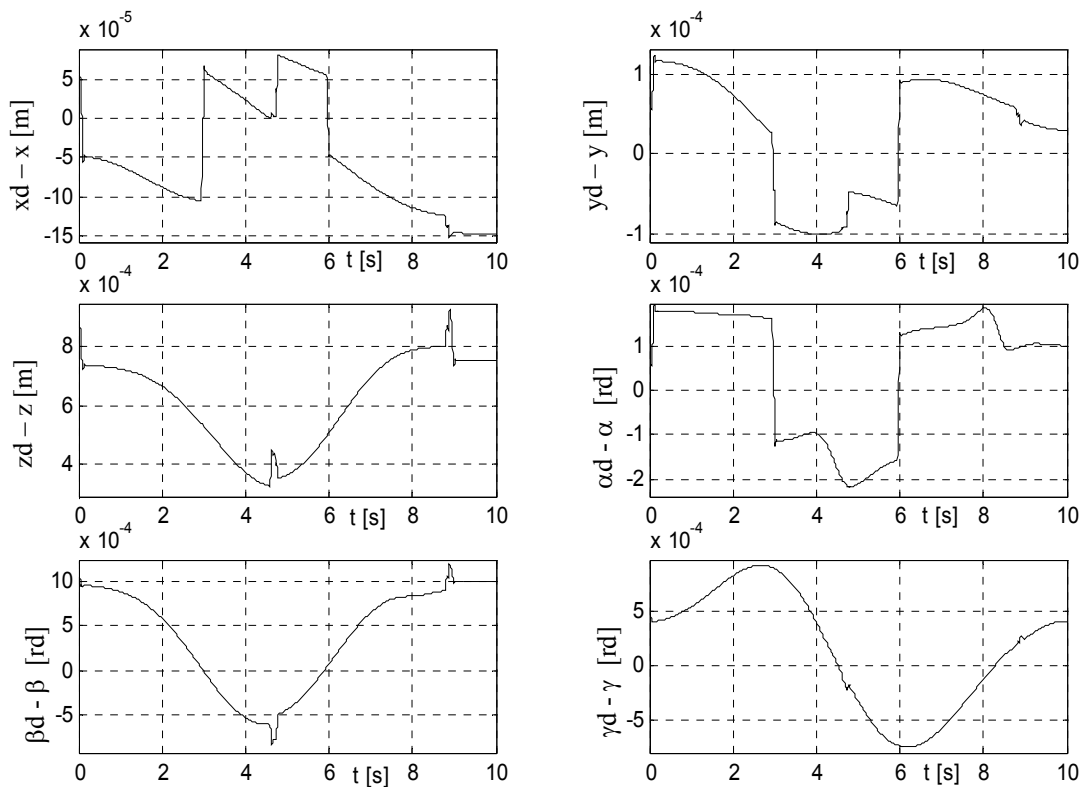


Fig.4.14.b Erreurs cartésiennes

▪ Robustesse vis-à-vis des perturbations extérieures

Résultats de simulation en appliquant une charge de 2 kg à $t=5$ secondes (fig.4.15-4.17).

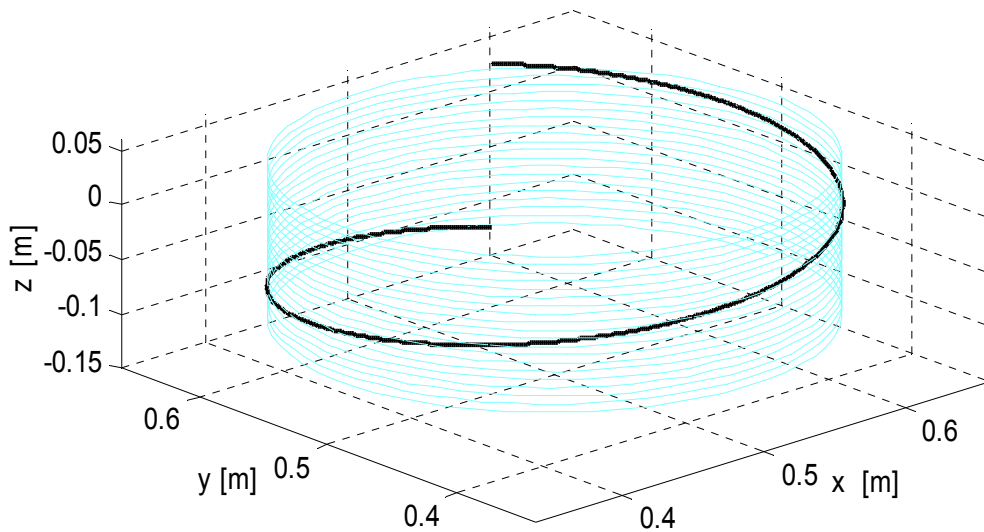


Fig.4.15.a Trajectoire de l'outil

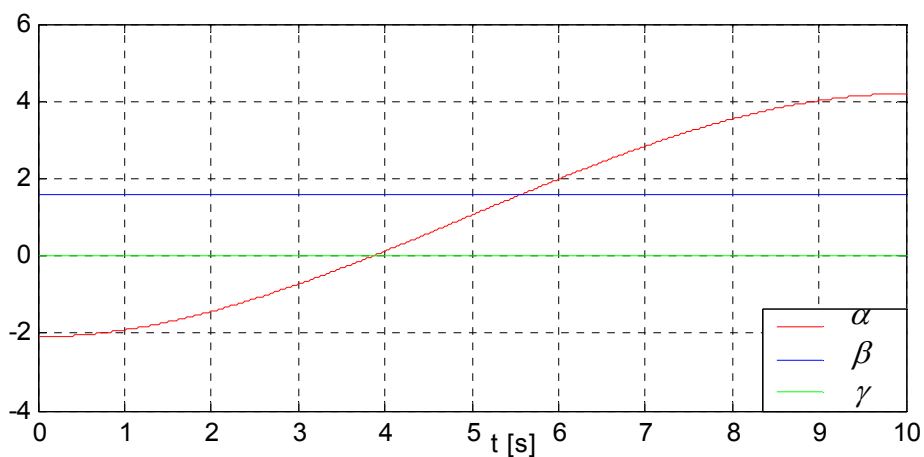


Fig.4.15.b Orientation de l'outil (unité : radian)

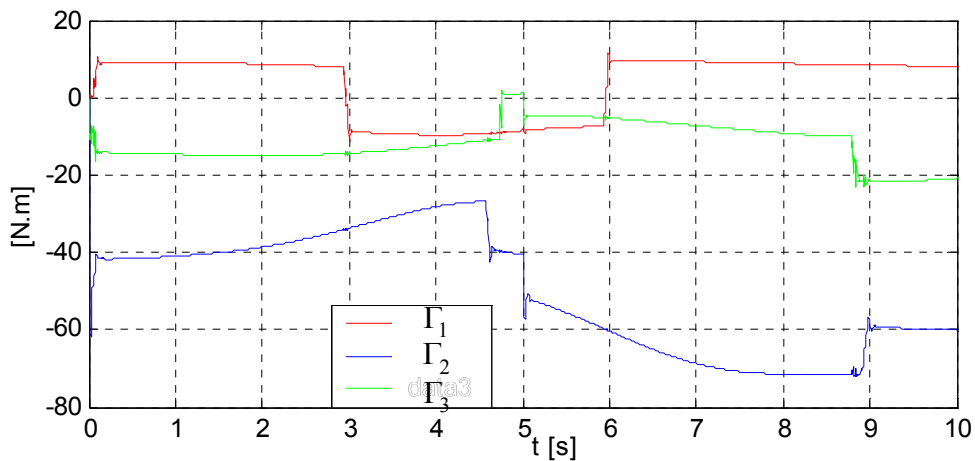


Fig.4.16.a Couples développés par les trois premiers actionneurs (unité : N.m)

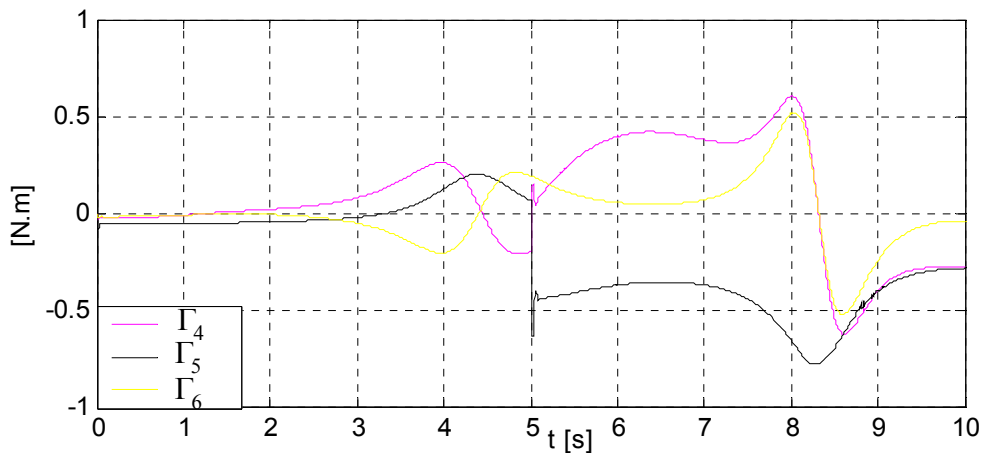


Fig.4.16.b Couples développés par les trois derniers actionneurs (unité : N.m)

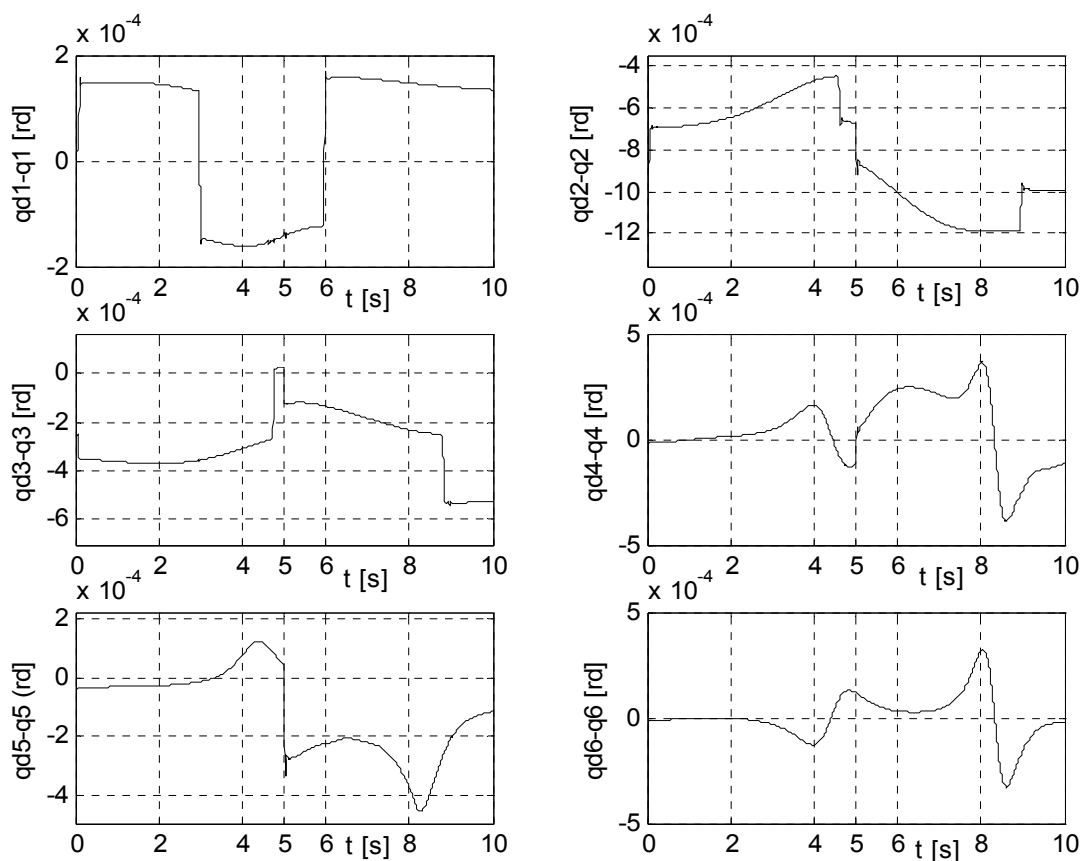


Fig.4.17.a Erreurs articulaires

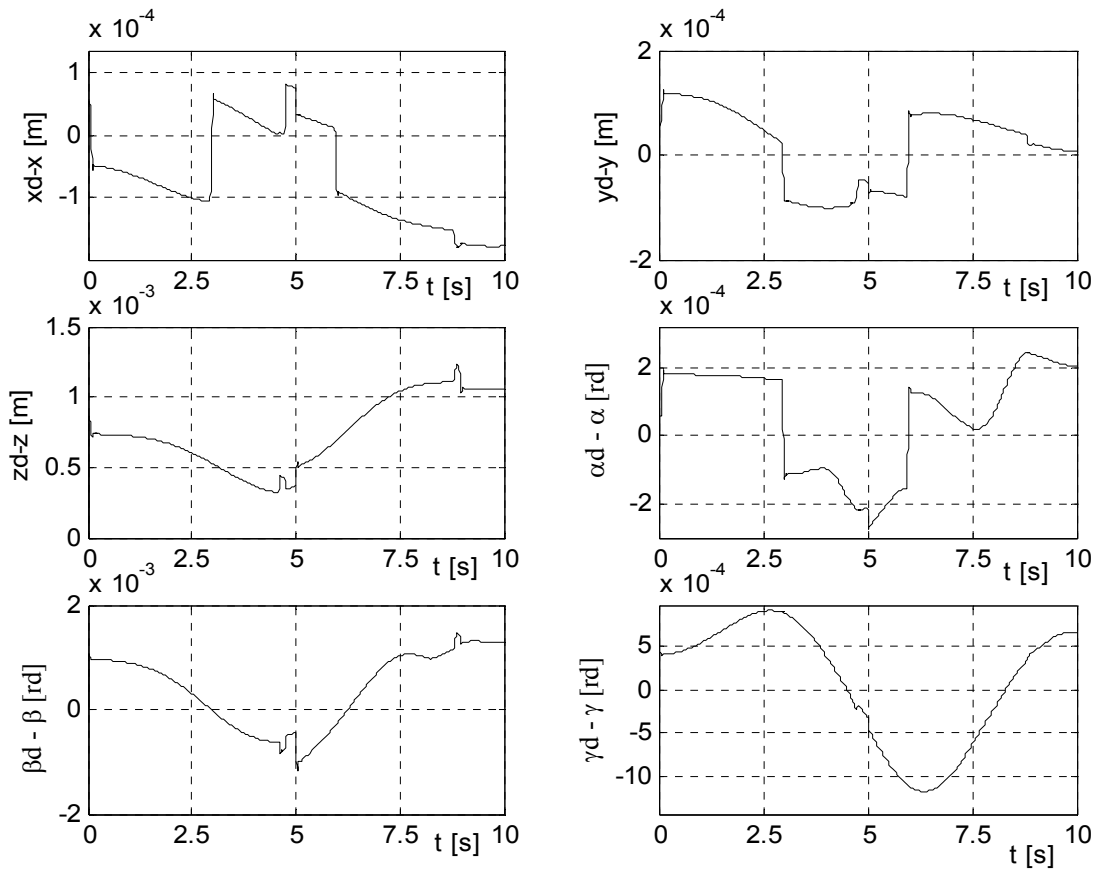


Fig.4.17.b Erreurs cartésiennes

▪ Robustesse vis-à-vis des variations paramétriques

Les figures 4.18-4.20 montrent les résultats de simulation en soumettant le robot à une variation de 100% de la matrice d'inertie.

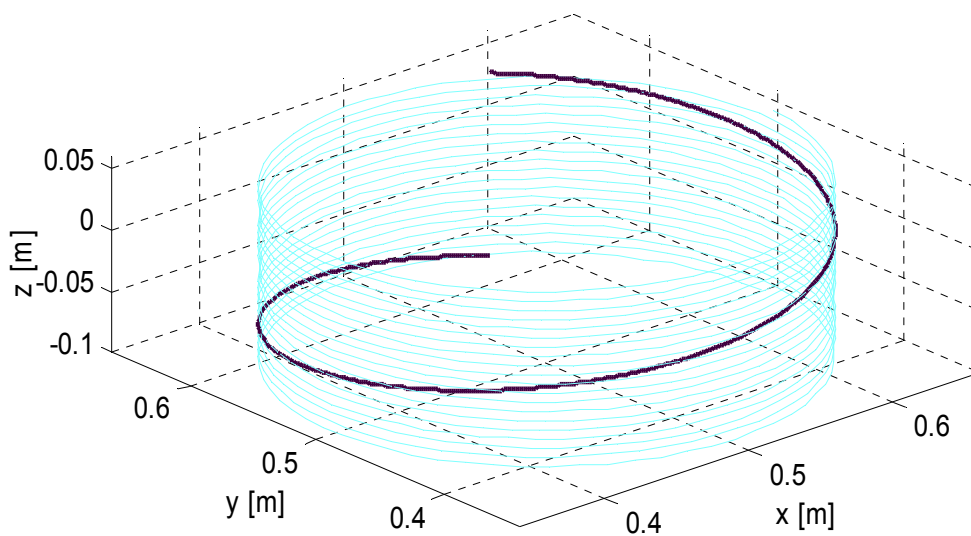


Fig.4.18.a Trajectoire de l'outil

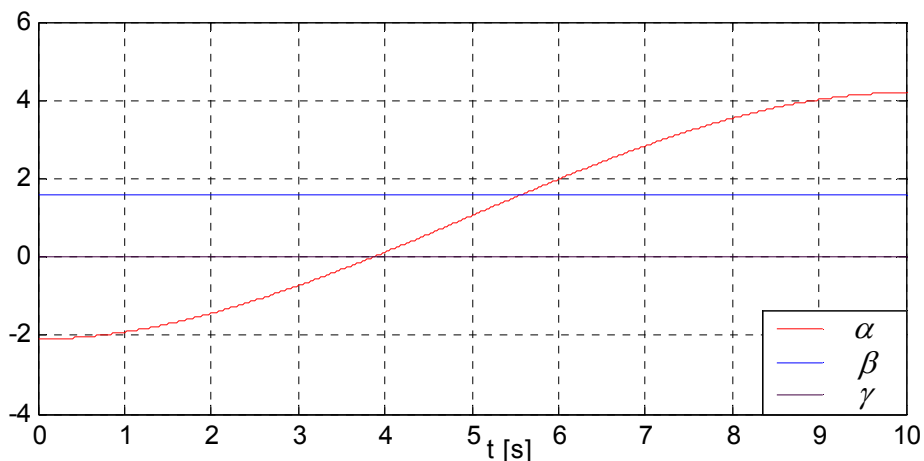


Fig.4.18.b Orientation de l'outil (unité : radian)

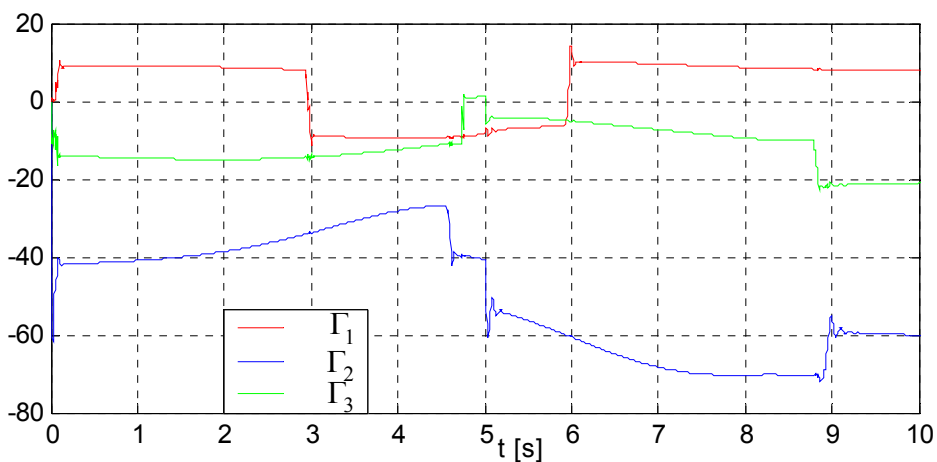


Fig.4.19.a Couples développés par les trois premiers actionneurs (unité : N.m)

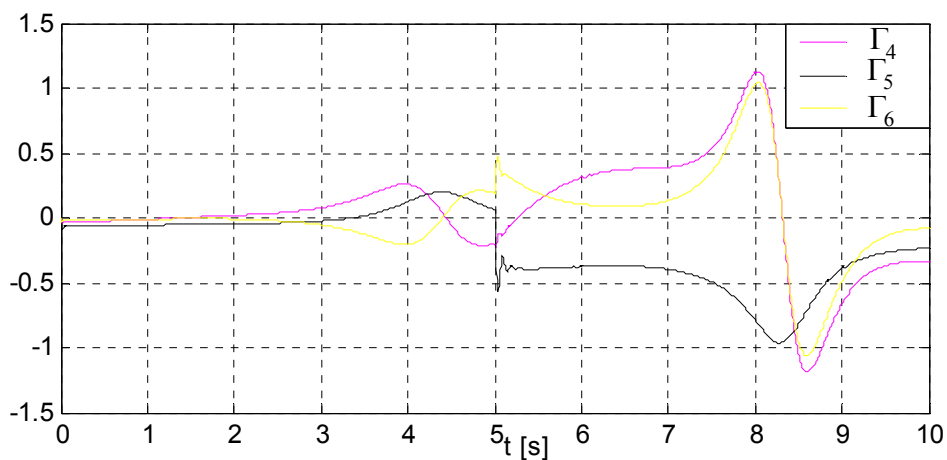


Fig.4.19.b Couples développés par les trois derniers actionneurs (unité : N.m)

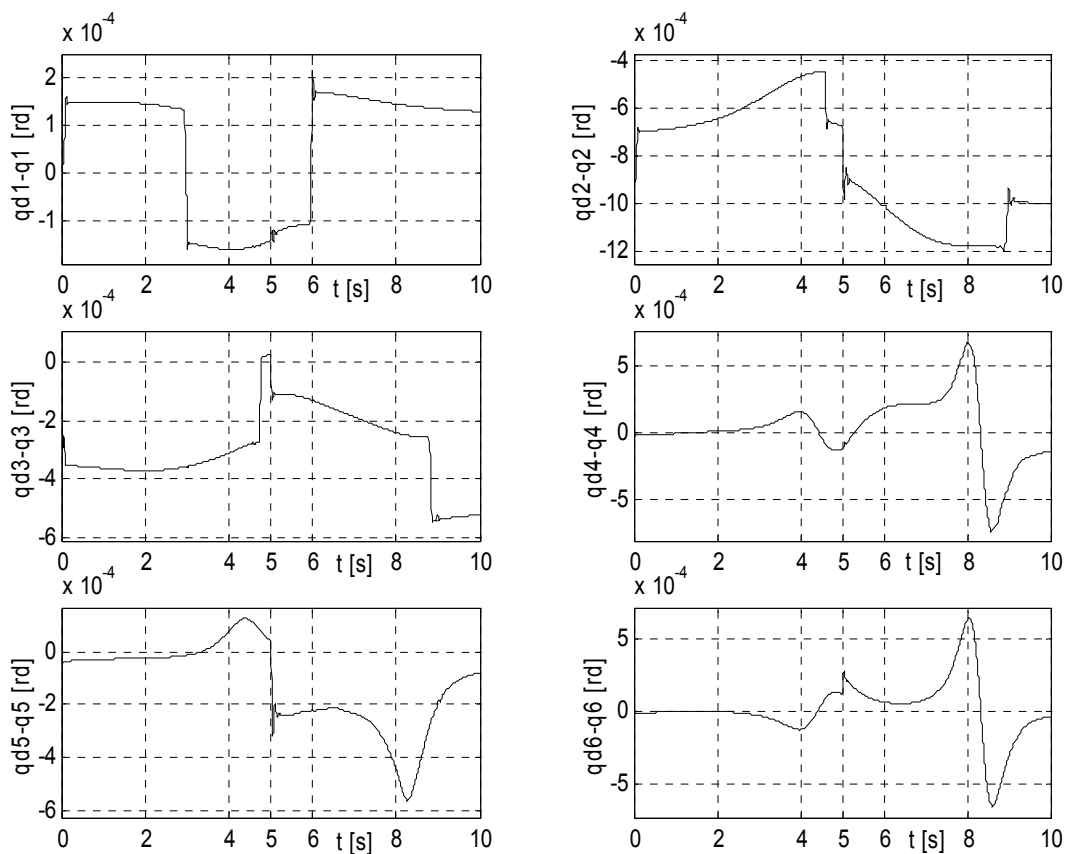


Fig.4.20.a Erreurs articulaires

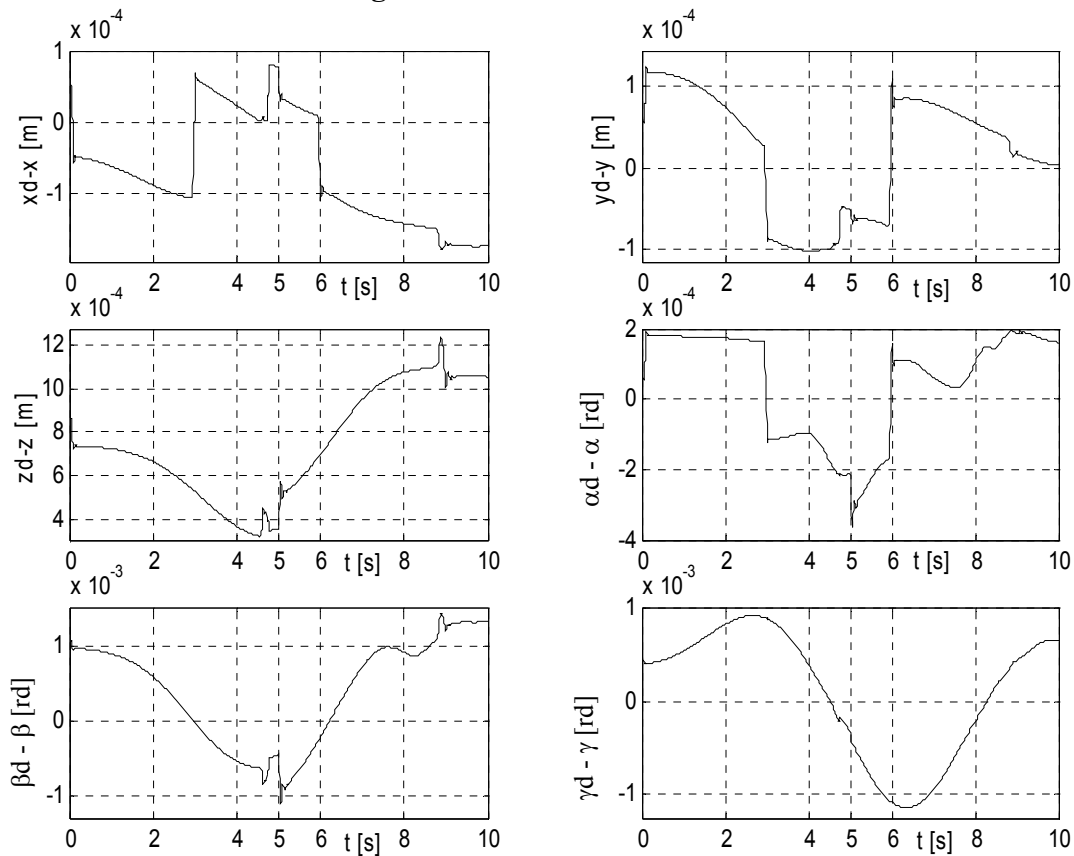


Fig.4.20.b Erreurs cartésiennes

	Tests sans perturbations	Tests avec charge	Tests avec perturbations paramétriques
$x^d - x [mm]$	0.15	0.17	0.17
$y^d - y [mm]$	0.12	0.12	0.12
$z^d - z [mm]$	0.92	1.2	1.2
$\alpha^d - \alpha [rd]$	2.16×10^{-4}	2.74×10^{-4}	3.63×10^{-4}
$\beta^d - \beta [rd]$	1.2×10^{-3}	1.2×10^{-3}	1.4×10^{-3}
$\gamma^d - \gamma [rd]$	9.21×10^{-4}	9.21×10^{-4}	9.21×10^{-4}

Tab.4.6 Erreurs maximales dans l'espace de la tâche

	Tests sans perturbations	Tests avec charge	Tests avec perturbations paramétriques
$\Gamma_1 [N.m]$	11.30	11.45	14.29
$\Gamma_2 [N.m]$	61.98	72.26	71.85
$\Gamma_3 [N.m]$	16.53	23.26	23.02
$\Gamma_4 [N.m]$	0.55	0.62	1.18
$\Gamma_5 [N.m]$	0.20	0.78	0.96
$\Gamma_6 [N.m]$	0.52	0.52	1.06

Tab.4.7 Couples maximaux développés par les actionneurs

Les courbes (figures 4.11.a et 4.11.b) montrent respectivement le mouvement du robot dans l'espace de travail lors de l'exécution de la tâche.

On constate que l'effecteur suit parfaitement sa trajectoire puisque les erreurs de poursuite dans l'espace cartésien restent stables (fig.4.14.b). De plus, les erreurs maximales de poursuite (Tab.4.6) et les valeurs maximales des couples (Tab.4.7) enregistrées lors de l'application des perturbations extérieures et des variations paramétriques montrent clairement que cette commande est robuste.

Les couples développés par les différents actionneurs restent dans les limites admissibles (Tab.4.7).

4.4 CONCLUSION

Dans ce chapitre, nous avons appliqué deux techniques de commande adaptatives floues décentralisées pour la poursuite de la trajectoire en position et en orientation du robot PUMA 560 à six degrés de liberté. Les lois d'adaptation paramétriques sont élaborées en utilisant le théorème de Lyapunov.

Dans l'approche directe les systèmes flous sont utilisés pour approcher les sous-systèmes réels qui caractérisent les articulations, une indépendamment des autres en utilisant uniquement les informations locales propre à chaque sous système. Par contre, dans l'approche indirecte, les systèmes flous sont utilisés pour générer directement les couples articulaires

Contrairement aux approches centralisées, le nombre de règles utilisées pour la conception des systèmes flous dans la commande décentralisée est nettement réduit. Les algorithmes de commande décentralisés sont moins complexes que ceux des approches centralisées

Les résultats de simulation montrent bien l'efficacité des lois de commande décentralisées floues pour la commande des robots.

CONCLUSION GENERALE

CONCLUSION GENERALE

Les travaux présentés dans ce mémoire ont pour objectif d'appliquer des techniques de commande adaptatives floues pour la poursuite de la trajectoire en position et en orientation du robot Puma 560. La commande des robots à six degrés de liberté n'est pas une tâche aisée. Cela est dû à la complexité de son modèle dynamique et aux différentes contraintes telles que les butées mécaniques, les positions singulières etc. La très grande majorité des résultats publiés dans ce domaine ont utilisés des robots à deux ou à trois degrés de liberté avec la génération des trajectoires dans l'espace de configuration.

Arrivé au terme de ce mémoire et avant d'en évoquer les perspectives, nous nous proposons de faire un rapide bilan des principaux résultats obtenus en résumant le contenu des quatre chapitres.

Dans le premier chapitre introductif, on a présenté la tâche à exécuter par le robot et qui consiste à l'opération du soudage de forme spirale sur un cylindre avec l'orientation de l'outil radialement vers l'axe central du cylindre. Le choix du robot est dicté par la nature de la tâche qui nécessite la poursuite en position et en orientation de l'outil dans l'espace. Cette tâche ne peut être réalisée que par un robot à six degrés de liberté. La génération de la trajectoire est effectuée dans l'espace cartésien, en tenant compte des positions singulières et les butées articulaires.

Dans la deuxième section du chapitre 1, on a fait un tour d'horizon des insuffisances des différentes techniques de commande classiques et adaptative non floues pour la commande des robots.

Dans le deuxième chapitre, l'architecture des systèmes flous en générale puis ceux de Takagi-Sugeno ont été présentés. Les systèmes flous ont été utilisés soit pour représenter la dynamique du robot Puma 560 (techniques indirectes) soit pour former directement le signal de commande (techniques directes).

Quoi que le nombre de règles utilisées dans la construction des systèmes flous est important (augmente exponentiellement en fonction de nombre d'entrées du robot), les résultats obtenus par les techniques de commande adaptative centralisées (chapitre 3) ont montré que le robot est

insensible aux différentes perturbations. Le volume de calcul résultant n'est plus un handicapé vue la vitesse d'exécution des processeurs actuels.

Bien que le nombre de règles utilisées pour la construction des systèmes flous, dans les approches de commande décentralisée, est nettement réduit (chapitre 4), les résultats obtenus par les différentes simulations ont montré l'efficacité de ces dernières dans la commande des robot en position et en orientation.

Il est difficile de faire une comparaison de point de vue performance entre les résultats obtenus par les différentes techniques de commandes appliquées par ce que les valeurs des paramètres de commande et d'adaptation ne sont pas optimaux, ajouter à cela que la qualité d'approximation obtenue par les système flous dépend fortement du nombre et de la forme des fonctions d'appartenance attribuées à chaque entrée.

Comme la simulation ne peut en aucun cas remplacer une application réelle. Les systèmes de commande appliqués dans ce mémoire ne pourront être validé qu'une fois testés sur des robots réels.

Comme perspective, il serait intéressant d'appliquer d'autres techniques telles que la commande adaptative neuro-floue et la commande adaptative floues par les algorithmes génétiques.

REFERENCES
BIBLIOGRAPHIQUES

REFERENCES BIBLIOGRAPHIQUES

- [Abda-91] C. Abdallah, D. Dawson, P. Dorato, et M. Jamshidi, 'Survey of Robust Control for Rigid Robots', IEEE Control System, pp. 24-30, 1992.
- [Ande-89] B. D. O. Anderson et al. 'Stability of Adaptive Systems: Passivity and averaging Techniques'. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [Arms-86] Brian Armstrong et Ossama Khatib, 'The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 arm'. Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, avril 1986, pp. 510-518.
- [Bail-87] E. Bailey et A. Arapostathis, 'Simple Sliding Mode Control Scheme Applied to Robot Manipulators', Int. J. Contr., Vol. 45, pp. 1197-1209, 1987.
- [Bale-83] Balestrino, A. De Maria, G. et Sciavicco, L., (1983), 'An Adaptive Model Following Control for Robot Manipulators', J. Dyn. Sys, Meas., and cont., Vol 105, 143-151.
- [Barto-85] G. Bartolini et T. Zolezzi, 'Variable Structure Nonlinear in the Control Law' IEEE Trans. Auto. Contr., Vol. AC-30, pp. 684, July 1985.
- [Bouk-00] Reda Boukezzoula, 'Commande Floue d'une Classe des Systèmes Non lineaire : Application au Probleme de Suivre de Trajectoire', these de doctorat, universite de savoie 2000.
- [Buck-92] Buckley J., 'Universal Fuzzy Controllers', Automatica, Vol. 28, No. 6, pp. 1245-1248, 1992.
- [Buhl-94] Hansruedi Bühler 'Réglage par Logique Floue', presses polytechniques et universitaires Romandes. 1994.
- [Chan-00] Yeong-Chan Chang, 'Robust Tracking Control for Nonlinear MIMO Systems via Fuzzy Approaches', Automatica 36 (2000) 1535-1545
- [Chek-03] H. Chekireb, M. Tadjine et D. Bouchaffra, 'Direct Adaptive Fuzzy Control of Nonlinear System Class with Applications', Control and intelligent systems, vol.31, No.2, 2003
- [Chia-02] Chiang-Cheng Chiang et Zu-Hung Kuo 'Decentralised Adaptive Fuzzy Controller Design of Large-Scale Nonlinear Systems with Unmatched Uncertainties', IEEE, 2002
- [Corl-81] M. J. Corless et G. Leitman, 'Continuous States Feedback Guaranteeing Uniform Ultimate Boundedness for Uncertain Dynamic Systems', IEEE Trans. Autom. Contr., Vol AC-26, pp. 1139-1144, 1981.
- [Craig-86] JOHN J. CRAIG 'Introduction to Robotics: Mechanics and Control', adison-wesley publishing company, Canada, 1986.
- [Deso-75] C. Desoer et M. Vidyasagar, 'Feedback Systems: Input-Output properties'. New York: Academic. 1975.

- [Flie-92] M Fliesse et J Lévine, Ph. Martin, and P. Rouchon. 'Sur les Systèmes non Linéaires Différentiellement plats '. C. R. Acad. Sci. Paris, I-315 :619-624, 1992.
- [Foul-92] Foulloy L., 'Contrôle qualitative et contrôle flou: Vers une Méthode d'écriture des Contrôleurs Flous', Actes des 12 ièmes journées internationales sur les systèmes experts et leurs applications", Avignon, France, 1992.
- [Foul-82] Foulard C., Gentil S. et Sandraz J.P., 'Commande et Régulation par Calculateur Numérique', Ed. Eyrolles, 1982.
- [Jin-98] Yaochu Jin, 'Decentralized Adaptive Fuzzy Control of Robot Manipulators', IEEE transactions on systems, man, and cybernetics—part b: cybernetics, vol. 28, no. 1, février 1998
- [Kasa-98] Nikola K. Kasabov, 'Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering', A Bradford Book The MIT Press Cambridge, Massachusettes London, England, Second printing, 1998
- [Khal-99] Wissama Khalil 'Modélisation Identification et Commande des Robot', Hermes science publication, Paris, 1988,1999.
- [Kim-95] Y.T. Kim et Z. Bien. 'Robust Self Learning Fuzzy Logic Controller', IEEE inter. Conf. On robotics and automation, pp. 1172, 1995.
- [Kwan-99] E. Kwan et M. liu. 'Robot Manipulators Tracking Using Hybrid fuzzy Logic Controllers', Proc.Aust. Conf. On robotics and Auto'. , pp. 155, 1999.
- [Labi-05] Salim Labiod 'Contribution à la Commande Adaptative Floue des Systèmes non lineaires', these de doctorat, ENP Algerie, 2005.
- [Lin-97] C. K. Lin et S. D. Wang. 'Robust Self-tuning Rotated Fuzzy Basis Function Controller for Robot arms', IEEE Proc. Control Theory Appl., Vol.155, No. 4, pp.293, 1997.
- [Liu-00] Ming Liu, 'Stability Analysis of Decentralized Adaptive Fuzzy Logic Control for Robot Arm Tracking', IEEE 2000
- [Morg-85] R. G. Morgan et U. Ozgunner, 'A Decentralised Variable Structure Control Algorithm for Robotic Manipulators' IEEE J. Rob. &Auto., Vol. RA-1, pp 57-65.1985
- [Moud-95] Moudgal V.G., Kwong W.A., Passino K.M. et Yurkovich Y. 'Fuzzy Learning Control for a Flexible-Link Robot', IEEE Trans. on Fuzzy Systems, Vol. 3, NO. 2, mai 1995
- [Nare-80] Narendra, K.S., Lin, Y., and Valvani, L. (1980), 'Stable Adaptive Control Designe : Part II, Proof of Stability', IEEE Trans. Automatic Control, AC-25, 440-448.
- [Neo-95] S. S. Neo et M. J. Er, 'Adaptive Fuzzy Control of Robot Manipulators', IEEE, 1995
- [Nico-84] S. Nicosia et P. Tomei, 'Model Reference Adaptive Control Algorithms for Industrial Robots', Automatica, pp. 635-644, Mai 1984.
- [Pade-88] B.E. Paden and S.S. Sastray, 'A Calculus for Computing Filipov's Differential Inclusion with Application to the Variable Structure Control of Robot Manipulators', IEEE Trans. Circ. Sys., Vol CAS-34, pp73-82, Jan. 1987.

- [Peter-94] Peter. I .Corke et Brian Armstrong ‘A Search for Consensus Among Model Parameters Reported for the PUMA 560 Robot. ICRA 1994: 1608-1613
- [Seli-92] J.m. selig ‘Introductory Robotics’,prentice hall, 1992.
- [Sing-85] Singh, S.N., (1985), ‘Adaptive Model Following Control of Non Linear Robotic Systems’, IEEE Trans. Auto. Contr., AC-30, No.11.
- [Slot-87] Slotine, J,-J.,and Li, W., (1987), ‘On the Adaptive Control of Robot Manipulateurs’, Int. J. of Robotics Research, Vol. 6, No. 3, pp. 49-59.
- [Slot-85] J-J.E. Slotine, ‘The Robust Control of Robot Manipulators,’ Int. J. Rob. Res., Vol. 4, pp. 49-564,1985.
- [Slot-83] J-J.E. Slotine et S.S. Sastry, ‘Tracking Control of Nonlinear Systems using Sliding Surface withe Applications to Robot Manipulators, ‘Int. L. Contr., Vol. 38, pp. 465-492, 1983.
- [Spon-87] M. W. Spong and M. Vidyasagar, ‘Robust Linear Compensator Design for non Linear Robotic Control’. IEEE J. Rob. Autom., Vol. Ra-3, pp. 345-351, Aug. 1987
- [Spoo-96] Jeffry T.Spooner and Kevin M. Passino, ‘Adaptive Control of a Class of Decentralised Nonlinear Systems’, IEEE Transactions on Automatic Control,Vol.41, N°2, 1996.
- [Spoo-96] Jeffrey T. Spooner et Kevin M. Passino, ‘Stable Adaptive control Using Fuzzy Systems and Neural Networks’, IEEE Transactions on Fuzzy Systems, vol. 4 no. 3 Aout 1996.
- [Spoo-02] Jeffrey T. Spooner, Manfredi Maggiore, Raul Ordonez, Kevin M. Passino ‘Stable Adaptive Control and Estimation for Nonlinear Systems :Neural and Fuzzy ApproximatorTechniques’.2002
- [Sun-96] F. C. Sun, Z. Q. Sun et G. Feng, ‘Design of Adaptive Fuzzy Sliding Mode Controller for Robot Manipulators’, IEEE 1996
- [Taka-85] Takagi T. et Sugeno M. ‘Fuzzy Identification of Systems and its Application to Modeling and Control’, IEEE Trans. on Syst. Man and Cybernetics. Vol. 15, pp.116-132, 1985.
- [Tang-02] Jiantao Tang et Shaocheng, Tong, ‘Output Feedback Fuzzy Adaptive Control for a Class of Large-Scale Nonlinear Systems’, Processings of the 4th World congress on Intelligent Control and Automation. June 10-14, 2002, Shanghai, P. R. China, pp 1930-1934.
- [Tong-03] Shaocheng Tong, ‘Adaptive Fuzzy Decentralized Control for a Class of Large-Scale Nonlinear Systems’, IEEE Trans. on Cybern. Man Sys, Part B, 2003.
- [Utki-77] V.I. Utkin, ‘Variable Structure Systems with Sliding Modes’, IEEE Trans. Auto. Contr., Vol. AC-22, pp.212-222, Apr.1977.
- [Vidy-85] M. Vidyasagar, ‘Control Systems Synthesis : a Factorization Approach’, Cambridge, MA, M.I.T. Press, 1985.
- [wang-94] Li-xin-Wang ‘Adaptive Fuzzy Systems and Control: design and stability analysis’, Prentice hall, 1994.

- [Wang-92]** Wang L.X. et Mendel J.M. 'Fuzzy Basis Functions, Universal Approximation and Orthogonal Least-Squares Learning', IEEE Trans. on Neural Networks, Vol. 3, No. 5, pp. 807-814, 1992.
- [Woo-04]** Chang-Woo Park et Young-Wan Cho, 'T-S Model Based Indirect Adaptive Fuzzy Control Using Online Parameter Estimation', IEEE transactions on systems, man, and cybernetics—part b: cybernetics, vol. 34, no. 6, décembre 2004
- [Yi-97]** S. Y. and M. J. Chung. 'A Robust Fuzzy Logic Controller for Robot Manipulators with Uncertainties', IEEE Trans. Syst. Man and Cybern., ~01.27, No.4, pp.706, 1997.
- [Yosh-90]** Tsuneo yoshikawa, 'Fondation of Robotics: Analyse and Control', The MIT press combridge, massachusetts, London, England.
- [Youn-78]** K-K.D. Young, 'Controller Design for a Manipulator using Theory of Variable Structure Systems' IEEE Trans. Syst., Man, and Cyber., Vol. SMC-8, pp. 210-218, 1978.

ANNEXES

ANNEXE 1

PARAMETRES DU MODELE GEOMETRIQUE ET DYNAMIQUE DU PUMA 560

Paramètres inertiels

$$I_1 = I_{zz} + m_1 r_{y1}^2 + m_2 d_2^2 + (m_4 + m_5 + m_6) a_3^2 + m_2 r_{z2}^2 + (m_3 + m_4 + m_5 + m_6) (d_2 + d_3)^2 + I_{xx2} + I_{yy3} + 2 m_2 d_2 r_{z2} + m_2 r_{y2}^2 + m_3 r_{z3}^2 + 2 m_3 (d_2 + d_3) r_{z3} + I_{zz4} + I_{yy5} + I_{zz6};$$

$$I_2 = I_{zz2} + m_2 (r_{x2}^2 + r_{y2}^2) + (m_3 + m_4 + m_5 + m_6) a_2^2;$$

$$I_3 = -I_{xx2} + I_{yy2} + (m_3 + m_4 + m_5 + m_6) a_2^2 m_2 r_{x2}^2 - m_2 r_{y2}^2;$$

$$I_4 = m_2 r_{x2} (d_2 + r_{z2}) + m_3 a_2 r_{a3} + (m_3 + m_4 + m_5 + m_6) a_2 (d_2 + d_3);$$

$$I_5 = -m_3 a_2 r_{y3} + (m_4 + m_5 + m_6) a_2 d_4 + m_4 a_2 r_{z4};$$

$$I_6 = I_{zz3} + m_3 r_{y3}^2 + m_4 a_3^2 + m_4 (d_4 + r_{z4})^2 + I_{yy4} + m_5 a_3^2 + m_5 d_4^2 + I_{zz5} + m_6 a_3^2 + m_6^2 + m_6 r_{z6}^2 + I_{xx6};$$

$$I_7 = m_3 r_{y3}^2 + I_{xx3} - I_{yy3} + m_4 r_{z4}^2 + 2 m_4 d_4 r_{z4} + (m_4 + m_5 + m_6) (d_4^2 + a_3^2) + I_{yy4} - I_{zz4} + I_{zz3} - I_{yy5} + m_6 r_{z6}^2 - I_{zz6} + I_{xx6};$$

$$I_8 = -m_4 (d_2 + d_3) (d_4 + r_{z4}) - (m_5 + m_6) (d_2 + d_3) d_4 + m_3 r_{y3} r_{z3} + m_3 (d_2 + d_3) r_{y3};$$

$$I_9 = m_2 r_{y2} (d_2 + r_{z2});$$

$$I_{10} = 2 m_4 a_3 r_{z4} + 2 (m_4 + m_5 + m_6) a_3 d_4;$$

$$I_{11} = -2 m_2 r_{x2} r_{y2};$$

$$I_{12} = (m_4 + m_5 + m_6) a_2 a_3;$$

$$I_{13} = (m_4 + m_5 + m_6) a_3 (d_2 + d_3);$$

$$I_{14} = I_{zz4} + I_{yy5} + I_{zz6};$$

$$I_{15} = m_6 d_4 r_{z6};$$

$$I_{16} = m_6 a_2 r_{z6};$$

$$I_{17} = I_{zz5} + I_{xx6} + m_6 r_{z6}^2;$$

$$I_{18} = m_6 (d_2 + d_3) r_{z6};$$

$$I_{19} = I_{yy4} - I_{xx4} + I_{zz5} - I_{yy5} + m_6 r_{z6}^2 + I_{xx6} - I_{zz6};$$

$$I_{20} = I_{yy5} - I_{xx5} - m_6 r_{z6}^2 + I_{zz6} - I_{xx6};$$

$$I_{21} = I_{xx4} - I_{yy4} + I_{xx5} - I_{zz5};$$

$$I_{22} = m_6 a_3 r_{z6};$$

$$I_{23} = I_{zz6};$$

avec

$$I_1 = 1.43 \quad \pm 0.05$$

$$I_2 = 1.75 \quad \pm 0.07$$

$$I_3 = 1.38 \quad \pm 0.5$$

$$I_4 = 6.9 * 10^{-1} \quad \pm 0.20 * 10^{-1}$$

$$I_5 = 3.72 * 10^{-1} \quad \pm 0.31 * 10^{-1}$$

$$I_6 = 3.33 * 10^{-1} \quad \pm 0.16 * 10^{-1}$$

$$I_7 = 2.98 * 10^{-1} \quad \pm 0.29 * 10^{-1}$$

$$I_8 = -1.34 * 10^{-1} \quad \pm 0.14 * 10^{-1}$$

$$\begin{array}{llll}
I_9 = & 2.38*10^{-2} & \pm 1.20*10^{-2} & \\
I_{11} = & -1.42*10^{-2} & \pm 0.70*10^{-2} & \\
I_{13} = & -3.79*10^{-3} & \pm 0.90*10^{-3} & \\
I_{15} = & 1.25*10^{-3} & \pm 0.30*10^{-3} & \\
I_{17} = & 6.42*10^{-4} & \pm 3.00*10^{-4} & \\
I_{19} = & 3.00*10^{-4} & \pm 14.0*10^{-4} & \\
I_{21} = & -1.00*10^{-4} & \pm 6.00*10^{-4} & \\
I_{23} = & 4.00*10^{-5} & \pm 2.00*10^{-5} & \\
I_{10} = & -2.13*10^{-2} & \pm 0.22*10^{-2} & \\
I_{12} = & -1.10*10^{-2} & \pm 0.11*10^{-2} & \\
I_{14} = & 1.64*10^{-3} & \pm 0.07*10^{-3} & \\
I_{16} = & 1.24*10^{-3} & \pm 0.30*10^{-3} & \\
I_{18} = & 4.31*10^{-4} & \pm 1.30*10^{-4} & \\
I_{20} = & -2.02*10^{-4} & \pm 8.00*10^{-4} & \\
I_{22} = & -5.80*10^{-5} & \pm 1.50*10^{-5} &
\end{array}$$

et

$$\begin{array}{llll}
I_{m1} = & 1.14 & \pm 0.27 & \\
I_{m3} = & 8.27*10^{-1} & \pm 0.93*10^{-1} & \\
I_{m5} = & 1.79*10^{-1} & \pm 0.14*10^{-1} & \\
I_{m2} = & 4.71 & \pm 0.54 & \\
I_{m4} = & 2.00*10^{-1} & \pm 0.16*10^{-1} & \\
I_{m6} = & 1.93*10^{-1} & \pm 0.16*10^{-1} &
\end{array}$$

Les constantes gravitationnelles

$$\begin{array}{l}
G_1 = -g ((m_3 + m_4 + m_5 + m_6) a_2 + m_2 r_{x2}) ; \\
G_2 = g (m_3 r_{y3} - (m_4 + m_5 + m_6) d_4 - m_4 r_{z4}) ; \\
G_3 = g m_2 r_{y2} ; \\
G_4 = -g (m_4 + m_5 + m_6) a_3 ; \\
G_5 = -g m_6 r_{z6} ;
\end{array}$$

avec

$$\begin{array}{llll}
G_1 = & -37.2 & \pm 0.50 & \\
G_3 = & 1.02 & \pm 0.50 & \\
G_5 = & -2.82*10^{-2} & \pm 0.56*10^{-2} & \\
G_2 = & -8.44 & \pm 0.20 & \\
G_4 = & 2.49*10^{-1} & \pm 0.25*10^{-1} &
\end{array}$$

Les éléments de la matrice d'inertie (6 × 6)L'unité des éléments est [kg·m²].**1^{ère} ligne**

$$\begin{array}{l}
a_{11} = I_{m1} + I_1 + I_3 * c_2^2 + I_7 * s_{23}^2 + I_{10} * s_{23} * c_{23} + I_{11} * s_2 * c_2 + I_{20} * (\sin(q_5))^2 * (\sin(q_2 + q_3))^2 * (1 + (\cos(q_4))^2) - 1) - 2 * \sin(q_2 + q_3) * \\
\cos(q_2 + q_3) * \cos(q_4) * \sin(q_5) * \cos(q_5) + I_{21} * \sin(q_2 + q_3)^2 * \cos(q_4)^2 + 2 * (I_5 * \cos(q_2) * \sin(q_2 + q_3) + I_{12} * \cos(q_2) * \\
\cos(q_2 + q_3) + I_{15} * (\sin(q_2 + q_3))^2 * \cos(q_5) + \sin(q_2 + q_3) * \cos(q_2 + q_3) * \cos(q_4) * \sin(q_5) + I_{16} * \cos(q_2) * \\
(\sin(q_2 + q_3) * \cos(q_5) + \cos(q_2 + q_3) * \cos(q_4) * \sin(q_5)) + I_{18} * \sin(q_4) * \sin(q_5) + I_{22} * (\sin(q_2 + q_3) * \cos(q_2 + q_3) * \cos(q_4) * \\
\sin(q_5) + \cos(q_2 + q_3)^2 * \cos(q_4) * \cos(q_5))) ; \\
\approx 2.57 + 1.38 * \cos(q_2)^2 + 0.30 * \sin(q_2 + q_3)^2 + 7.44 * 10^{-1} * \cos(q_2) * \sin(q_2 + q_3) ; \\
a_{12} = I_4 * \sin(q_2) + I_8 * \cos(q_2 + q_3) + I_9 * \cos(q_2) + I_{13} * \sin(q_2 + q_3) - I_{15} * \cos(q_2 + q_3) * \sin(q_4) * \sin(q_5) + I_{16} * \sin(q_2) * \\
\sin(q_4) * \sin(q_5) + I_{18} * (\sin(q_2 + q_3) * \cos(q_4) * \sin(q_5) - \cos(q_2 + q_3) * \cos(q_5)) + I_{19} * \sin(q_2 + q_3) * \sin(q_4) * \\
\cos(q_4) + I_{20} * \sin(q_4) * (\sin(q_2 + q_3) * \cos(q_4) * \cos(q_5)^2 + \cos(q_2 + q_3) * \sin(q_5) * \cos(q_5)) + I_{22} * \sin(q_2 + q_3) * \\
\sin(q_4) * \sin(q_5) ; \\
\approx 6.90 * 10^{-1} * \sin(q_2) - 1.34 * 10^{-1} * \cos(q_2 + q_3) + 2.38 * 10^{-2} * \cos(q_2) ; \\
a_{13} = I_8 * \cos(q_2 + q_3) + I_{13} * \sin(q_2 + q_3) - I_{15} * \cos(q_2 + q_3) + I_{19} * \sin(q_2 + q_3) * \sin(q_4) * \cos(q_4) + I_{18} * (\sin(q_2 + q_3) * \\
\cos(q_4) * \sin(q_5) - \cos(q_2 + q_3) * \cos(q_5)) + I_{22} * \sin(q_2 + q_3) * \sin(q_4) * \sin(q_5) + I_{20} * \sin(q_4) * (\sin(q_2 + q_3) *
\end{array}$$

$$\cos(q4) * \cos(q5)^2 + \cos(q2+q3) * \sin(q5) * \cos(q5));$$

$$\approx -1.34 * 10^{-1} * \cos(q2+q3) + -3.97 * 10^{-3} * \sin(q2+q3);$$

$$a_{14} = I_{14} * \cos(q2+q3) + I_{15} * \sin(q2+q3) * \cos(q4) * \sin(q5) + I_{16} * \cos(q2) * \cos(q4) * \sin(q5) + I_{18} * \cos(q2+q3) * \sin(q4) * \sin(q5) - I_{20} * (\sin(q2+q3) * \cos(q4) * \sin(q5) * \cos(q5) + \cos(q2+q3) * \sin(q5)^2) + I_{22} * \cos(q2+q3) * \cos(q4) * \sin(q5)$$

$$\approx 0;$$

$$a_{15} = I_{15} * \sin(q2+q3) * \sin(q4) * \cos(q5) + I_{16} * \cos(q2) * \sin(q4) * \cos(q5) + I_{17} * \sin(q2+q3) * \sin(q4) + I_{18} * (\sin(q2+q3) * \sin(q5) - \cos(q2+q3) * \cos(q4) * \cos(q5)) + I_{22} * \cos(q2+q3) * \sin(q4) * \cos(q5)$$

$$\approx 0;$$

$$a_{16} = I_{23} * (\cos(q2+q3) * \cos(q5) - \sin(q2+q3) * \cos(q4) * \sin(q5))$$

$$\approx 0;$$

2^{ème} ligne

$$a_{21} = a_{12}$$

$$a_{22} = I_{m2} + I_2 + I_6 + I_{20} * \sin(q4)^2 * \sin(q5)^2 + I_{21} * \sin(q4)^2 + 2 * (I_5 * \sin(q3) + I_{12} * \cos(q3) + I_{15} * \cos(q5) + I_{16} * (\sin(q3) * \cos(q5) + \cos(q3) * \cos(q4) * \sin(q5))) + I_{22} * \cos(q4) * \sin(q5));$$

$$\approx 6.79 + 7.44 * 10^{-1} * \sin(q3);$$

$$a_{23} = I_5 * \sin(q3) + I_6 + I_{12} * \cos(q3) + I_{16} * (\sin(q3) * \cos(q5) + \cos(q3) * \cos(q4) * \sin(q5)) + I_{20} * \sin(q4)^2 * \sin(q5)^2 + I_{21} * \sin(q4)^2 + 2 * (I_{15} * \cos(q5) + I_{22} * \cos(q4) * \sin(q5));$$

$$\approx 0.333 + 3.72 * 10^{-1} * \sin(q3) - 1.110 * 10^{-2} * \cos(q3);$$

$$a_{24} = -I_{15} * \sin(q4) * \sin(q5) + I_{16} * \sin(q3) * \sin(q4) * \sin(q5) + I_{20} * \sin(q4) * \sin(q5) * \cos(q5)$$

$$\approx 0;$$

$$a_{25} = I_{15} * \cos(q4) * \cos(q5) + I_{16} * (\cos(q3) * \sin(q5) + \sin(q3) * \cos(q4) * \cos(q5))$$

$$\approx 0;$$

$$a_{26} = I_{23} * \sin(q4) * \sin(q5)$$

$$\approx 0;$$

3^{ème} ligne

$$a_{31} = a_{13};$$

$$a_{32} = a_{23};$$

$$a_{33} = I_{m3} + I_6 + I_{20} * \sin(q4)^2 * \sin(q5)^2 + I_{21} * \sin(q4)^2 + 2 * (I_{15} * \cos(q5) + I_{22} * \cos(q4) * \sin(q5));$$

$$\approx 1.16;$$

$$a_{34} = -I_{15} * \sin(q4) * \sin(q5) + I_{20} * \sin(q4) * \sin(q5) * \cos(q5);$$

$$\approx 0;$$

$$a_{35} = I_{15} * \cos(q4) * \cos(q5) + I_{17} * \cos(q4) + I_{22} * \sin(q5) \approx 1.25 * 10^{-3} * \cos(q4) * \cos(q5);$$

$$a_{36} = I_{23} * \sin(q4) * \sin(q5)$$

$$\approx 0;$$

4^{ème} ligne

$$a_{41} = a_{14};$$

$$a_{42} = a_{24};$$

$$a_{43} = a_{34};$$

$$a_{44} = I_{m4} + I_{14} - I_{20} * \sin(q5)^2 \approx 0.20;$$

$$a_{45} = 0;$$

$$a_{46} = I_{23} * \cos(q5) \\ \approx 0;$$

5^{ème} ligne

$$a_{51} = a_{15};$$

$$a_{52} = a_{25};$$

$$a_{53} = a_{35};$$

$$a_{54} = a_{45};$$

$$a_{55} = I_{m5} + I_{17} \approx 0.18;$$

$$a_{56} = 0;$$

6^{ème} ligne

$$a_{61} = a_{16};$$

$$a_{62} = a_{26};$$

$$a_{63} = a_{36};$$

$$a_{64} = a_{46};$$

$$a_{65} = a_{56};$$

$$a_{66} = I_{m6} + I_{23} \\ \approx 0.19;$$

Eléments de la matrice de Coriolis (6 × 15)

L'unité des éléments [Kg·m²]

1^{ère} ligne

$$b_{112} = 2 * (-I_3 * \sin(q2) * \cos(q2) + I_5 * \cos(q2 + q2 + q3) + I_7 * \sin(q2 + q3) * \cos(q2 + q3) - I_{12} * \sin(q2 + q2 + q3) + I_{15} * (2 * \sin(q2 + q3) * \cos(q2 + q3) * \cos(q5) + (1 - 2 * \sin(q2 + q3)^2) * \cos(q4) * \sin(q5)) + I_{16} * (\cos(q2 + q2 + q3) * \cos(q5) - \sin(q2 + q2 + q3) * \cos(q4) * \sin(q5)) + I_{21} * \sin(q2 + q3) * \cos(q2 + q3) * \cos(q4)^2 + I_{20} * ((1 + \cos(q4))^2) * \sin(q2 + q3) * \cos(q2 + q3) * \sin(q5)^2 - (1 - 2 * \sin(q2 + q3)^2) * \cos(q4) * \sin(q5) * \cos(q5)) + I_{22} * ((1 - 2 * \sin(q2 + q3)^2) * \cos(q5) - 2 * \sin(q2 + q3) * \cos(q4) * \sin(q5))) + I_{10} * (1 - 2 * \sin(q2 + q3)^2) + I_{11} * (1 - 2 * \sin(q2)^2); \\ \approx -2.76 * \sin(q2) * \cos(q2) + 7.44 * 10^{-1} * \cos(q2 + q2 + q3) + 0.60 * \sin(q2 + q3) * \cos(q2 + q3) - 2.13 * 10^{-2} * (1 - 2 * \sin(q2 + q3)^2);$$

$$b_{113} = 2 * (I_5 * \cos(q2) * \cos(q2 + q3) + I_7 * \sin(q2 + q3) * \cos(q2 + q3) - I_{12} * \cos(q2) * \sin(q2 + q3) + I_{15} * (2 * \sin(q2 + q3) * \cos(q2 + q3) * \cos(q5) + (1 - 2 * \sin(q2 + q3)^2) * \cos(q4) * \sin(q5)) + I_{20} * ((1 + \cos(q4))^2) * \sin(q2 + q3) * \cos(q2 + q3) * \sin(q5)^2 - (1 - 2 * \sin(q2 + q3)^2) * \cos(q4) * \sin(q5) * \cos(q5)) + I_{22} * ((1 - 2 * \sin(q2 + q3)^2) * \cos(q5) - 2 * \sin(q2 + q3) * \cos(q4) * \sin(q5))) + I_{10} * (1 - 2 * \sin(q2 + q3)^2); \\ \approx 7.44 * 10^{-1} * \cos(q2) * \cos(q2 + q3) + 0.60 * \sin(q2 + q3) * \cos(q2 + q3) + 2.20 * 10^{-2} * \cos(q2) * \sin(q2 + q3) - 2.13 * 10^{-2} * (1 - 2 * \sin(q2 + q3)^2);$$

$$b_{114} = 2 * (-I_{15} * \sin(q2 + q3) * \cos(q2 + q3) * \sin(q4) * \sin(q5) - I_{16} * \cos(q2) * \cos(q2 + q3) * \sin(q4) * \sin(q5) + I_{18} * \cos(q4) * \sin(q5) - I_{20} * (\sin(q2 + q3)^2 * \sin(q5)^2 * \sin(q4) * \cos(q4) - \sin(q2 + q3) * \cos(q2 + q3) * \sin(q4) * \sin(q5) * \cos(q5)) - I_{22} * \cos(q2 + q3)^2 * \sin(q4) * \sin(q5) - I_{21} * \sin(q2 + q3)^2 * \sin(q4) * \cos(q4)); \\ \approx -2.50 * 10^{-3} * \sin(q2 + q3) * \cos(q2 + q3) * \sin(q4) * \sin(q5) + 8.060 * 10^{-4} * \cos(q4) * \sin(q5) - 2.48 * 10^{-}$$

$$\begin{aligned}
& 3*\cos(q2)*\cos(q2+q3)*\sin(q4)*\sin(q5); \\
b_{115} = & 2*(I_{20}*(\sin(q5)*\cos(q5))*(\cos(q4)^2*(1-\cos(q2+q3)^2)-\cos(q2+q3)^2)-\sin(q2+q3)*\cos(q2+q3)* \\
& \cos(q4)*(1-2*\sin(q5)^2))-I_{15}*(\sin(q2+q3)^2*\sin(q5)-\sin(q2+q3)*\cos(q2+q3)*\cos(q4)*\cos(q5))- \\
& I_{16}*\cos(q2)*(\sin(q2+q3)*\sin(q5)-\cos(q2+q3)*\cos(q4)*\cos(q5))+I_{18}*\sin(q4)*\cos(q5)+I_{22}* \\
& (\cos(q2+q3)^2*\cos(q4)*\cos(q5)-\sin(q2+q3)*\cos(q2+q3)*\sin(q5))); \\
& \approx -2.50*10^{-3}*(\sin(q2+q3)^2*\sin(q5)-\sin(q2+q3)*\cos(q2+q3)*\cos(q4)*\cos(q5))-2.48*10^{-3} \\
& *3*\cos(q2)*(\sin(q2+q3)*\sin(q5)-\cos(q2+q3)*\cos(q4)*\cos(q5))+8.60*10^{-4}*\sin(q4)*\cos(q5); \\
b_{116} = & 0; \\
b_{123} = & 2*(-I_8*\sin(q2+q3)+I_{13}*\cos(q2+q3)+I_{15}*\sin(q2+q3)*\sin(q4)*\sin(q5)+I_{18}*(\cos(q2+q3)*\cos(q4)* \\
& \sin(q5)+\sin(q2+q3)*\cos(q5))+I_{19}*\cos(q2+q3)*\sin(q4)*\cos(q4)+I_{20}*\sin(q4)*(\cos(q2+q3)* \\
& \cos(q4)*\cos(q5)^2-\sin(q2+q3)*\sin(q5)*\cos(q5))+I_{22}*\cos(q2+q3)*\sin(q4)*\sin(q5)); \\
& \approx 2.67*10^{-1}*\sin(q2+q3)-7.58*10^{-3}*\cos(q2+q3); \\
b_{124} = & -I_{18}^2*\sin(q2+q3)*\sin(q4)*\sin(q5)+I_{19}*\sin(q2+q3)*(1-(2*\sin(q4)^2))+I_{20}*\sin(q2+q3)*(1- \\
& 2*\sin(q4)^2*\cos(q5)^2)-I_{14}*\sin(q2+q3); \\
& \approx 0; \\
b_{125} = & I_{17}*\cos(q2+q3)*\sin(q4)+I_{18}^2*(\sin(q2+q3)*\cos(q4)*\cos(q5)+\cos(q2+q3)*\sin(q5))+I_{20}*\sin(q4)*(\cos(q2+ \\
& q3)*(1-2*\sin(q5)^2)-\sin(q2+q3)*\cos(q4)*2*\sin(q5)*\cos(q5)); \\
& \approx 0; \\
b_{126} = & I_{23}*(\sin(q2+q3)*\cos(q5)+\cos(q2+q3)*\cos(q4)*\cos(q5)); \\
& \approx 0; \\
b_{134} = & b_{124}; \quad b_{135} = b_{125}; \quad b_{136} = b_{126}; \\
b_{145} = & 2*(I_{15}*\sin(q2+q3)*\cos(q4)*\cos(q5)+I_{16}*\cos(q2)*\cos(q4)*\cos(q5)+I_{18}*\cos(q2+q3)*\sin(q4)*\cos(q5)+I_{22}* \\
& \cos(q2+q3)*\cos(q4)*\cos(q5))+I_{17}*\sin(q2+q3)*\cos(q4)-I_{20}*(\sin(q2+q3)*\cos(q4)*(1- \\
& 2*\sin(q5)^2)+2*\cos(q2+q3)*\sin(q5)*\cos(q5)); \\
& \approx 0; \\
b_{146} = & I_{23}*\sin(q2+q3)*\sin(q4)*\sin(q5); \\
& \approx 0; \\
b_{156} = & -I_{23}*(\cos(q2+q3)*\sin(q5)+\sin(q2+q3)*\cos(q4)*\cos(q5)); \\
& \approx 0; \\
\mathbf{2^{\text{eme}} \text{ ligne}} \\
b_{212} = & 0; \quad b_{213} = 0; \\
b_{214} = & I_{14}*\sin(q2+q3)+I_{19}*\sin(q2+q3)*(1-(2*\sin(q4)^2))+2*(-I_{15}*\cos(q2+q3)*\cos(q4)*\sin(q5)+ \\
& I_{16}*\sin(q2)*\cos(q4)*\sin(q5)+I_{20}*(\sin(q2+q3)*(\cos(q5)^2*\cos(q4)^2-0.5)+\cos(q2+q3)* \\
& \cos(q4)*\sin(q5)*\cos(q5))+I_{22}*\sin(q2+q3)*\cos(q4)*\sin(q5)); \\
& \approx 1.64*10^{-3}*\sin(q2+q3)-2.50*10^{-3}*\cos(q2+q3)*\cos(q4)*\sin(q5)+2.48*10^{-3}*\sin(q2)*\cos(q4)*\sin(q5)+ \\
& 0.30*10^{-3}*\sin(q2+q3)*(1-(2*\sin(q4)^2)); \\
b_{215} = & 2*(-I_{15}*\cos(q2+q3)*\sin(q4)*\cos(q5)+I_{22}*\sin(q2+q3)*\sin(q4)*\cos(q5)+I_{16}*\sin(q2)*\sin(q4)*\cos(q5))- \\
& I_{17}*\cos(q2+q3)*\sin(q4)+I_{20}*(\cos(q2+q3)*\sin(q4)*(1-2*\sin(q5)^2)-2*\sin(q2+q3)*\sin(q4)* \\
& \cos(q4)*\sin(q5)*\cos(q5)); \\
& \approx -2.50*10^{-3}*\cos(q2+q3)*\sin(q4)*\cos(q5)+2.48*10^{-3}*\sin(q2)*\sin(q4)*\cos(q5)-6.42*10^{-4} \\
& *\cos(q2+q3)*\sin(q4); \\
b_{216} = & -b_{126}; \\
b_{223} = & 2*(-I_{12}*\sin(q3)+I_3*\cos(q3)+I_{16}*(\cos(q3)*\cos(q5)-\sin(q3)*\cos(q5)-\sin(q3)*\cos(q4)*\sin(q5)));
\end{aligned}$$

$$\begin{aligned} &\approx 2.20*10^{-2}*\sin(q3)+7.44*10^{-1}*\cos(q3); \\ b_{224} &= 2*(I_{16}*\cos(q3)*\sin(q4)*\sin(q5)+I_{20}*\sin(q4)*\cos(q4)*\sin(q5)^2+I_{21}*\sin(q4)\cos(q4)-I_{22}*\sin(q4)*\sin(q5)); \\ &\approx -2.48*10^{-3}*\cos(q3)*\sin(q4)*\sin(q5); \\ b_{225} &= 2*(-I_{15}*\sin(q5)+I_{16}*(\cos(q3)*\cos(q4)*\cos(q5)-\sin(q3)*\sin(q5))+I_{20}*\sin(q4)^2*\sin(q5)*\cos(q5)+I_{22}*\cos(q4)*\cos(q5)); \\ &\approx -2.50*10^{-3}*\sin(q5)+2.48*10^{-3}*(\cos(q3)*\cos(q4)*\cos(q5)-\sin(q3)*\sin(q5)); \\ b_{226} &= 0; \quad b_{234}=b_{224}; \quad b_{235}=b_{225}; \quad b_{236}=0; \\ b_{245} &= 2*(-I_{15}*\sin(q4)*\cos(q5)-I_{16}*\sin(q3)*\sin(q4)*\cos(q5))-I_{17}*\sin(q4)+I_{20}*\sin(q4)*(1-2*\sin(q5)^2); \\ &\approx 0; \\ b_{246} &= I_{23}*\cos(q4)*\sin(q5); \\ &\approx 0; \\ b_{256} &= I_{23}*\sin(q4)*\cos(q5); \\ &\approx 0; \end{aligned}$$

3^{ème} ligne

$$\begin{aligned} b_{312} &= 0; \quad b_{313}=0; \\ b_{314} &= 2*(I_{15}*\cos(q2+q3)*\cos(q4)*\sin(q5)+I_{22}*\sin(q2+q3)*\sin(q4)*\sin(q5)+I_{20}*(\sin(q2+q3)*(\cos(q5)^2*\cos(q4)^2-0.5)+\cos(q2+q3)*\cos(q4)*\sin(q5)*\cos(q5)))+I_{14}*\sin(q2+q3)+I_{19}*\sin(q2+q3)*(1-2*\sin(q4)^2)); \\ &\approx -2.50*10^{-3}*\cos(q2+q3)*\cos(q4)*\sin(q5)+1.64*10^{-3}*\sin(q2+q3)+0.30*10^{-3}*\sin(q2+q3)*(1-2*\sin(q4)^2); \\ b_{315} &= 2*(-I_{15}*\cos(q2+q3)*\sin(q4)*\cos(q5)+I_{22}*\sin(q2+q3)*\sin(q4)*\cos(q5))-I_{17}*\cos(q2+q3)*\sin(q4)+I_{20}*\sin(q4)*(\cos(q2+q3)*(1-2*\sin(q5)^2)-2*\sin(q2+q3)*\cos(q4)*\sin(q5)*\cos(q5)); \\ &\approx -2.50*10^{-3}*\cos(q2+q3)*\sin(q4)*\cos(q5)-6.42*10^{-4}*\cos(q2+q3)*\sin(q4); \\ b_{316} &= -b_{136}; \quad b_{323}=0; \\ b_{324} &= 2*(I_{20}*\sin(q4)*\cos(q4)*\sin(q5)^2+I_{21}*\sin(q4)*\cos(q4)-I_{22}*\sin(q4)*\sin(q5)) \approx 0; \\ b_{325} &= 2*(-I_{15}*\sin(q5)+I_{20}*\sin(q4)^2*\sin(q5)*\cos(q5)+I_{22}*\cos(q4)*\cos(q5)); \\ &\approx -2.50*10^{-3}*\sin(q5); \\ b_{326} &= 0; \quad b_{334}=b_{324}; \quad b_{335}=b_{325}; \quad b_{336}=0; \\ b_{345} &= -I_{15}*2*\sin(q4)*\cos(q5)-I_{17}*\sin(q4)+I_{20}*\sin(q4)*(1-2*\sin(q5)^2); \\ &\approx -2.50*10^{-3}*\sin(q4)*\cos(q5); \\ b_{346} &= b_{246}; \quad b_{356}=b_{256}; \end{aligned}$$

4^{ème} ligne

$$\begin{aligned} b_{412} &= -b_{214}; \quad b_{413}=-b_{314}; \quad b_{414}=0; \\ b_{415} &= -I_{20}*(\sin(q2+q3)*\cos(q4)*(1-2*\sin(q5)^2)+2*\cos(q2+q3)*\sin(q5)*\cos(q5))-I_{17}*\sin(q2+q3)*\cos(q4); \\ &\approx -6.42*10^{-4}*\sin(q2+q3)*\cos(q4); \\ b_{416} &= -b_{146}; \quad b_{423}=-b_{324}; \quad b_{424}=0; \\ b_{425} &= I_{17}*\sin(q4)+I_{20}*\sin(q4)*(1-2*\sin(q5)^2); \\ &\approx 6.24*10^{-4}*\sin(q4); \\ b_{426} &= -b_{246}; \quad b_{434}=0; \quad b_{435}=b_{425}; \quad b_{436}=-b_{346}; \end{aligned}$$

$$b_{445} = -I_{20} * 2 * \sin(q5) * \cos(q5) \\ \approx 0;$$

$$b_{446} = 0;$$

$$b_{456} = -I_{23} * \sin(q5) \\ \approx 0;$$

5^{ème} ligne

$$b_{512} = -b_{215}; \quad b_{513} = -b_{315}; \quad b_{514} = -b_{415}; \quad b_{515} = 0; \quad b_{516} = -b_{156};$$

$$b_{523} = -b_{325}; \quad b_{524} = -b_{425}; \quad b_{525} = 0; \quad b_{526} = -b_{256};$$

$$b_{534} = b_{524}; \quad b_{535} = 0; \quad b_{536} = -b_{356};$$

$$b_{545} = 0; \quad b_{546} = -b_{456};$$

$$b_{556} = 0;$$

6^{ème} ligne

$$b_{612} = b_{126}; \quad b_{613} = b_{136}; \quad b_{614} = b_{146}; \quad b_{615} = b_{156}; \quad b_{616} = 0;$$

$$b_{623} = 0; \quad b_{624} = b_{246}; \quad b_{625} = b_{256}; \quad b_{626} = 0;$$

$$b_{634} = b_{624}; \quad b_{635} = b_{625}; \quad b_{636} = 0;$$

$$b_{645} = b_{456}; \quad b_{646} = 0; \quad b_{656} = 0;$$

Eléments de la matrice centrifuge (6 × 6)

L'unité des éléments [Kg·m²].

$$c_{11} = 0;$$

$$c_{12} = I_4 * \cos(q2) - I_8 * \sin(q2+q3) - I_9 * \sin(q2) + I_{13} * \cos(q2+q3) + I_{15} * \sin(q2+q3) * \sin(q4) * \sin(q5) + I_{16} * \cos(q2) * \sin(q4) * \sin(q5) + I_{18} * (\cos(q2+q3) * \cos(q4) * \sin(q5) + \sin(q2+q3) * \cos(q5)) + I_{19} * \cos(q2+q3) * \sin(q4) * \cos(q4) + I_{20} * \sin(q4) * (\cos(q2+q3) * \cos(q4) * \cos(q5))^2 - \sin(q2+q3) * \sin(q5) * \cos(q5)) + I_{22} * \cos(q2+q3) * \sin(q4) * \sin(q5); \\ \approx 6.90 * 10^{-1} * \cos(q2) + 1.34 * 10^{-1} * \sin(q2+q3) - 2.38 * 10^{-2} * \sin(q2);$$

$$c_{13} = 0.5 * b_{123};$$

$$c_{14} = -I_{15} * \sin(q2+q3) * \sin(q4) * \sin(q5) - I_{16} * \cos(q2) * \sin(q4) * \sin(q5) + I_{18} * \cos(q2+q3) * \cos(q4) * \sin(q5) + I_{20} * \sin(q2+q3) * \sin(q4) * \sin(q5) * \cos(q5) - I_{22} * \cos(q2+q3) * \sin(q4) * \sin(q5) \\ \approx 0;$$

$$c_{15} = -I_{15} * \sin(q2+q3) * \sin(q4) * \sin(q5) - I_{16} * \cos(q2) * \sin(q4) * \sin(q5) + I_{18} * (\sin(q2+q3) * \cos(q5) + \cos(q2+q3) * \cos(q4) * \sin(q5)) - I_{22} * \cos(q2+q3) * \sin(q4) * \sin(q5) \\ \approx 0;$$

$$c_{16} = 0;$$

$$c_{21} = -2.1 * b_{112};$$

$$c_{22} = 0;$$

$$c_{23} = 0.5 * b_{223};$$

$$c_{24} = -I_{15} * \cos(q4) * \sin(q5) - I_{16} * \sin(q3) * \cos(q4) * \sin(q5) + I_{20} * \cos(q4) * \sin(q5) * \cos(q5) \approx 0;$$

$$c_{25} = -I_{15} * \cos(q4) * \sin(q5) + I_{16} * (\cos(q3) * \cos(q5) - \sin(q3) * \cos(q4) * \sin(q5)) + I_{22} * \cos(q5) \approx 0;$$

$$c_{26} = 0;$$

$$c_{31} = -0.5 * b_{113};$$

$$c_{32} = -c_{23}; \quad c_{33} = 0;$$

$$c_{34} = -I_{15} * \cos(q4) * \sin(q5) + I_{20} * \cos(q4) * \sin(q5) * \cos(q5) \\ \approx -1.25 * 10^{-3} * \cos(q4) * \sin(q5);$$

$$c_{35} = -I_{15} * \cos(q4) * \sin(q5) + I_{22} * \cos(q5) \\ \approx c_{34};$$

$$c_{36} = 0;$$

$$c_{41} = -0.5 * b_{114};$$

$$c_{42} = -0.5 * b_{224};$$

$$c_{43} = 0.5 * b_{423};$$

$$c_{44} = 0; \quad c_{45} = 0; \quad c_{46} = 0;$$

$$c_{51} = -0.5 * b_{115};$$

$$c_{52} = -0.5 * b_{225};$$

$$c_{53} = 0.5 * b_{523};$$

$$c_{54} = -0.5 * b_{445};$$

$$c_{55} = 0;$$

$$c_{56} = 0;$$

$$c_{61} = 0; \quad c_{62} = 0; \quad c_{63} = 0; \quad c_{64} = 0; \quad c_{65} = 0; \quad c_{66} = 0;$$

Les termes de la gravité

L'unité des éléments est (Newton-mètre)

$$g_1 = 0;$$

$$g_2 = G_1 * \cos(q2) + G_2 * \sin(q2+q3) + G_3 * \sin(q2) + G_4 * \cos(q2+q3) + G_5 * (\sin(q2+q3) * \cos(q5) + \cos(q2+q3) * \cos(q4) * \sin(q5));$$

$$g_3 = G_2 * \sin(q2+q3) + G_4 * \cos(q2+q3) + G_5 * (\sin(q2+q3) * \cos(q5) + \cos(q2+q3) * \cos(q4) * \sin(q5));$$

$$g_4 = -G_5 * \sin(q2+q3) * \sin(q4) * \sin(q5);$$

$$g_5 = G_5 * (\cos(q2+q3) * \sin(q5) + \sin(q2+q3) * \cos(q4) * \cos(q5));$$

$$g_6 = 0;$$

TERMES DE FROTTEMENTS VISQUEUX

Les termes de frottements secs et visqueux sont décrits selon [Arms-88], pour l'articulation i , comme suit :

$$F_i(t) = \begin{cases} S_i^- + V_i^- \dot{q}_i & \text{si } \dot{q}_i < 0 \\ S_i^+ + V_i^+ \dot{q}_i & \text{si } \dot{q}_i > 0 \end{cases}$$

La variable \dot{q}_i est la vitesse angulaire de l'articulation i . Lorsque la vitesse articulaire est nulle $S_i=1$ [Craig-86] ce qui implique que $F_i=1$ puisque les frottements visqueux sont nuls. Les frottements secs S_i et visqueux V_i des trois premières articulations sont donnés ci-après :

$$\begin{array}{lll} S_1^- = -8.26 & ; & S_2^- = -11.34 & ; & S_3^- = -5.57 & ; \\ S_1^+ = 8.43 & & S_2^+ = 12.77 & & S_3^+ = 5.93 & \\ \\ V_1^- = 3.45 & ; & V_2^- = 8.53 & ; & V_3^- = 3.02 & ; \\ V_1^+ = 4.94 & & V_2^+ = 7.67 & & V_3^+ = 3.27 & \end{array}$$

L'indice supérieur (- ou +) indique le signe de la vitesse articulaire correspondante.

ANNEXE 2

MODELE GEOMETRIQUE DIRECT ET INVERSE DU PUMA 560

Le modèle géométrique direct (MGD) du robot permet d'exprimer les coordonnées opérationnelles (position/orientation) de l'effecteur en fonction des coordonnées articulaires ($q_1 \dots q_6$).

Les transformations homogènes qui décrivent le MGD sont :

$$\begin{aligned}
 {}^0T_1 &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ c_1 & s_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; & {}^1T_2 &= \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; & {}^2T_3 &= \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^3T_4 &= \begin{bmatrix} c_4 & -s_4 & 0 & a_3 \\ 0 & 0 & -1 & -d_4 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; & {}^4T_5 &= \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; & {}^5T_6 &= \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^6T_E &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{A2.1}$$

avec : ${}^{i-1}T_i$ est la transformation homogène qui transforme le repère $i-1$ vers le repère i

En utilisant les propriétés des transformations homogènes, on obtient le modèle géométrique direct du robot par :

$${}^6T_E = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_E = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \vdots & P_x \\ r_{21} & r_{22} & r_{23} & \vdots & P_y \\ r_{31} & r_{32} & r_{33} & \vdots & P_z \\ \dots & \dots & \dots & \vdots & \dots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix} = \begin{bmatrix} \vdots & P_x \\ {}^0R_E & \vdots & P_y \\ \vdots & P_z \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix} \tag{A2.2}$$

où

$$r_{11} = c_1 c_{23} (c_4 c_5 c_6 - s_4 s_6) - c_1 c_{23} s_5 c_6 - s_1 (s_4 c_5 c_6);$$

$$r_{12} = -c_1 c_{23} (c_4 c_5 s_6 + s_4 c_6) + c_1 s_{23} s_5 s_6 - s_1 (-s_4 c_5 s_6 + c_4 c_6);$$

$$r_{13} = c_1 c_{23} c_4 s_5 + c_1 s_{23} c_5 - s_1 s_4 s_5;$$

$$\begin{aligned}
r_{21} &= s_1 c_{23} (c_4 c_5 c_6 - s_4 s_6) - s_1 s_{23} s_5 c_6 + c_1 (s_4 c_5 c_6 + c_4 s_6); \\
r_{22} &= -c_1 c_{23} (c_4 c_5 s_6 + s_4 c_6) + s_1 s_{23} s_5 s_6 + c_1 (-s_4 c_5 s_6 + c_4 c_6); \\
r_{23} &= s_1 c_{23} c_4 s_5 + s_1 s_{23} c_5 + c_1 s_4 s_5; \\
r_{31} &= -s_{23} (c_4 c_5 c_6 - s_4 s_6) - c_{23} s_5 c_6; \\
r_{32} &= s_{23} (c_4 c_5 s_6 + s_4 c_6) + c_{23} s_5 s_6; \\
r_{33} &= -s_{23} c_4 s_5 + c_{23} c_5; \\
P_x &= -s_1 (d_2 + d_3) + c_1 c_2 a_2 + c_1 c_{23} (a_3 + c_4 s_5 L_6) + c_1 s_{23} (d_4 + c_5 L_6) - s_1 s_4 s_5 L_6; \\
P_y &= c_1 (d_2 + d_3) + s_1 c_2 a_2 + s_1 c_{23} (a_3 + c_4 s_5 L_6) + s_1 s_{23} (d_4 + c_5 L_6) + c_1 s_4 s_5 L_6; \\
P_z &= -s_2 a_2 - s_{23} (a_3 + c_4 s_5 L_6) + c_{23} (d_4 + c_5 L_6);
\end{aligned} \tag{A2.3}$$

avec $c_i = \cos(q_i)$; $s_i = \sin(q_i)$; $c_{ij} = \cos(q_i + q_j)$; $s_{ij} = \sin(q_i + q_j)$.

En utilisant la paramétrisation d'Euler zyz , la matrice de rotation 0R_E peut être exprimée uniquement en fonction de trois paramètres (α, β, γ) . Cette paramétrisation correspond à une rotation du repère de la base R_0 par rapport à z_0 d'un angle α définissant le repère R' , puis une rotation de ce dernier par rapport à y' d'un angle β définissant le repère R'' et en fin une rotation de ce dernier par rapport à z'' d'un angle γ .

La matrice de rotation associée à cette paramétrisation est donnée par :

$$R_{zyz} = \begin{bmatrix} c_\alpha c_\beta c_\gamma - s_\alpha s_\gamma & -c_\alpha c_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta \\ s_\alpha c_\beta c_\gamma + c_\alpha s_\gamma & -s_\alpha c_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta \\ -s_\beta c_\gamma & s_\beta s_\gamma & c_\beta \end{bmatrix} \tag{A2.4}$$

Par identification avec la matrice 0R_E , on obtient :

$$\begin{aligned}
\beta &= A \tan 2 \left(\sqrt{r_{31}^2 + r_{32}^2}, r_{33} \right); \\
\alpha &= A \tan 2 \left(\frac{r_{23}}{s_\beta}, \frac{r_{13}}{s_\beta} \right); \\
\gamma &= A \tan 2 \left(\frac{r_{32}}{s_\beta}, -\frac{r_{31}}{s_\beta} \right);
\end{aligned} \tag{A2.5}$$

En tenant compte de cette paramétrisation, le modèle géométrique direct s'écrit :

$$X = f(q) = [P_x \ P_y \ P_z \ \alpha \ \beta \ \gamma]^T \quad (\text{A2.6})$$

Le modèle géométrique inverse permet d'exprimer dans l'espace articulaire une position décrite dans l'espace opérationnel.

Puisque les trois derniers axes du PUMA 560 se coupe en un point, l'utilisation de la méthode analytique développée par Pieper [Khal-99] pour résoudre le problème de géométrie inverse est judicieux. Afin de simplifier les calculs, on ramène le problème au niveau du poignet (repère R_0)

$${}^0T_6 = {}^0T_E {}^6T_E^{-1} \quad (\text{A2.7})$$

On obtient :

$$\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} - L_6 \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} \quad (\text{A2.8})$$

Le modèle géométrique inverse est donné ci-après :

$$q_1 = \text{Atan} 2(w_x, w_y) - \text{Atan} 2(d_2 + d_3, \sqrt{w_x^2 + w_y^2 - (d_2 + d_3)^2})$$

$$q_3 = -\text{Atan} 2(a_3, d_4) + \text{Atan} 2(k, \sqrt{a_3^2 + d_4^2 - k^2})$$

avec $k = (w_x^2 + w_y^2 + w_z^2 - a_2^2 - a_3^2 - d_4^2 - (d_2 + d_3)^2) / 2a_2$

$$q_2 = \text{Atan} 2(k_1, k_2) - q_3$$

avec :

$$k_1 = \frac{[-(a_3 + a_2)w_z + (c_1 w_x + s_1 w_y)(a_2 s_3 + d_4)]}{[w_z^2 + (c_1 w_x + s_1 w_y)^2]}$$

et

$$k_2 = \frac{[(d_4 + a_2 s_3)w_z + (c_1 w_x + s_1 w_y)(a_2 c_3 + a_3)]}{[w_z^2 + (c_1 w_x + s_1 w_y)^2]}$$

$$q_4 = \text{Atan} 2(k_3, k_4) \quad (\text{A2.9})$$

avec : $k_3 = -s_1 r_{13} + c_1 r_{23}$ et $k_4 = c_1 c_{23} r_{13} + s_1 c_{23} r_{23} - s_{23} r_{33}$

$$q_5 = \text{Atan} 2(k_5, k_6)$$

avec : $k_5 = r_{13}(c_1 k_2 c_4 - s_1 s_4) + r_{23}(s_1 k_2 c_4 + c_1 s_4) - r_{33} k_1 c_4$

$$\text{et} \quad k_6 = r_{13} c_1 s_{23} + r_{23} s_1 s_{23} + r_{33} c_{23}$$

$$q_6 = \text{Atan2}(k_7, k_8)$$

$$\text{avec :} \quad k_7 = -r_{11} (c_1 c_{23} s_4 + s_1 c_4) + r_{21} (-s_1 c_{23} s_4 + c_1 c_4) + r_{31} s_{23} s_4$$

$$\text{et} \quad k_8 = -r_{12} (c_1 c_{23} s_4 + s_1 c_4) + r_{22} (-s_1 c_{23} s_4 + c_1 c_4) + r_{32} s_{23} s_4$$

MODÈLE CINÉMATIQUE DIRECT ET INVERSE DU PUMA 560

Le modèle cinématique direct décrit les vitesses opérationnelles en fonction des vitesses articulaires.

$$\text{Il est défini par :} \quad \dot{X} = J(q) \dot{q} \quad (\text{A2.10})$$

où $J(q)$ désigne la matrice jacobienne du robot. Le problème revient au calcul du jacobin. Dans le cas des robots à moins de trois degrés de libertés, ce dernier peut être obtenu en dérivant directement MGD. Mais, pour les robots à plus de trois degrés de liberté, les méthodes récursives s'imposent [Yosh-90] [Khal-89] [Craig-98] [Seli-92].

Etant donné le vecteur de position et de l'orientation de l'outil défini par :

$$X = [P_x \quad P_y \quad P_z \quad \alpha \quad \beta \quad \gamma]^T \quad (\text{A2.11})$$

sa dérivé par rapport au temps est :

$$\dot{X} = [\dot{P}_x \quad \dot{P}_y \quad \dot{P}_z \quad \dot{\alpha} \quad \dot{\beta} \quad \dot{\gamma}]^T \quad (\text{A2.12})$$

La méthode récursive consiste à calculer le jacobien de base J_V , qui exprime les vitesses réelles de l'effecteur (vitesses de translation et vitesses de rotation), sans passer par les dérivées partielles

Soit un robot série possédant n articulations et dont les variables articulaires sont notées $q_1 \dots q_n$ et $q = [q_1 \quad q_2 \quad \dots \quad q_n]$ le vecteur des coordonnées articulaires. Soit ${}^0V_{0n}$ la vitesse de l'origine du repère lié à l'organe terminal exprimée dans le repère de base et ${}^0\omega_{0n}$ les coordonnées du vecteur vitesse de rotation de l'organe terminal dans le repère de base.

$$\begin{bmatrix} {}^0V_{0n} \\ {}^0\omega_{0n} \end{bmatrix} = J_V \dot{q} \quad (\text{A2.13})$$

On applique la loi de décomposition des vitesses linéaires

$${}^0V_{0n} = {}^0V_{0n}^n = {}^0V_{01}^n + {}^0V_{12}^n + \dots + {}^0V_{n-1,n}^n = \sum_{i=1}^n {}^0V_{i-1,i}^n \quad (\text{A2.14})$$

où ${}^0V_{i-1,i}^n$ est la vitesse du repère n lors du mouvement du repère $i-1$ par rapport au repère i , exprimé dans le repère o .

Puisque tous les axes q_i sont rotoïdes alors :

$${}^0V_{i-1,i}^n = {}^0\omega_{i-1,i} \times {}^0(O_{i-1} O_i) \quad (\text{A2.15})$$

$$\text{Dans ce cas, on a aussi : } {}^0V_{i-1,i}^n = \dot{q}_i {}^0z_{i-1} \times {}^0({}^0R_{i-1}^{i-1} O_i) \quad (\text{A2.16})$$

où O_i est l'origine du repère i , ${}^0(O_{i-1} O_i)$ est le vecteur $O_{i-1} O_i$ exprimé dans le repère o , z est l'axe de rotation et 0R_i est la matrice de rotation du repère i par rapport au repère o et

$${}^0z_{i-1} = {}^0R_{i-1} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

On applique la loi de composition des vitesses angulaires :

$${}^0\omega_{0n} = {}^0\omega_{01} + {}^0\omega_{12} + \dots + {}^0\omega_{n-1,n} = \sum_{i=1}^n {}^0\omega_{i-1,i} \quad (\text{A2.17})$$

Puisque tous les axes q_i sont rotoïdes alors :

$${}^0\omega_{i-1} = {}^0z_{i-1} \dot{q}_i \quad (\text{A2.18})$$

De (A2.15), (A2.16), (A2.17), (A2.18) on tire le jacobien

$$J_V = [J_{V1} \ J_{V2} \ \dots \ J_{Vn}] \quad (\text{A2.19})$$

$$\text{avec } J_{Vi} = \begin{bmatrix} {}^0z_{i-1} \times ({}^0R_{i-1}^{i-1} O_n) \\ {}^0z_{i-1} \end{bmatrix} \quad (\text{A2.20})$$

En utilisant la paramétrisation d'Euler zyz , on aura la relation qui lie le vecteur \dot{X} aux vitesses réelles de l'effecteur [Craig-86] [Khal-99] :

$$\dot{X} = \begin{bmatrix} 1 & 0 & 0 & \vdots & 0 & 0 & 0 \\ 0 & 1 & 0 & \vdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \vdots & 0 & 0 & 0 \\ \dots & \dots & \dots & \vdots & \dots & \dots & \dots \\ 0 & 0 & 0 & \vdots & 0 & -s_\alpha & c_\alpha s_\beta \\ 0 & 0 & 0 & \vdots & 0 & c_\alpha & s_\alpha s_\beta \\ 0 & 0 & 0 & \vdots & 1 & 0 & c_\beta \end{bmatrix}^{-1} \begin{bmatrix} {}^0V_{0n} \\ {}^0\omega_{0n} \end{bmatrix} \quad (\text{A2.21})$$

où

$$\dot{X} = T^{-1} \begin{bmatrix} {}^0V_{0n} \\ {}^0\omega_{0n} \end{bmatrix} \quad (\text{A2.22})$$

$$\dot{X} = T^{-1} J_V \dot{q} = J \dot{q}$$