

# République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et  
de la Recherche Scientifique

*Ecole Nationale Polytechnique*



وزارة التعليم العالي  
و البحث العلمي  
المدرسة الوطنية المتعددة التقنيات

---

Projet de fin d'étude :  
En vue de l'obtention du diplôme d'ingénieur  
d'état en automatique.

---

## *Thème :*

**Commande adaptative floue du bras  
de robot manipulateur PUMA 560  
en position/orientation**

Proposé par :

**Mr. H. CHKIREB**

Etudié par :

**M<sup>elle</sup>. Sarah MECHHOUD**

*Promotion 2009*

## ملخص :

إن العمل المقدم في هذه المذكرة يتمثل في تطبيق آلية التحكم الغامضة على الذراع الآلية بيما 560 في الفضاء العملي. من أجل هذا قمنا أولاً بالتطرق إلى استخراج النماذج الهندسية الحركية و الديناميكية لذراعنا الآلية. ثم انتقلنا إلى تحديد نوعين من المسارات العملية الواجب إتباعها. حتى نبين فعالية تقنية التحكم المتبعة تم على سبيل المقارنة عرض نوعين آخرين من آلية التحكم المتبعة العزم المحسوب و الطريقة التلاؤمية. استعملنا مبدأ ليابونوف لنبرهن على توازن الجسم في الحلق المغلقة. و قد رافقت كل آلية مجموعة من المحاكاة على ذراعنا الآلية.

كلمات مفتاحية: الذراع الآلية بيما 560 الفضاء العملي المسار العملي النماذج الهندسية الحركة الديناميكية العزم المحسوب آلية التحكم التلاؤمية

## RESUME:

Le travail présenté dans ce mémoire est la commande en position/orientation dans l'espace libre du robot PUMA 560 en utilisant principalement la loi de commande adaptative floue centralisée et décentralisée (directe et indirecte). Dans cette fin, on a commencé par présenter les différentes modélisations du robot : géométrique, cinématique et dynamique. Ensuite, on est passé à la génération de deux types de trajectoires cartésiennes. Pour mettre en évidence l'intérêt de la commande adaptative floue, nous avons appliqué à titre de comparaison deux autres types de commande : la commande par couple calculé et la commande adaptative. Pour étudier la stabilité du système en boucle fermée nous avons utilisé le principe de Lyapunov. Pour valider le travail, des simulations ont accompagné chaque loi de commande.

Mots clés : PUMA 560, modélisation d'un bras de robot, espace opérationnel, trajectoire cartésienne, couple calculé, commande adaptative, commande adaptative floue, principe de Lyapunov

## ABSTRACT:

This thesis presents the position/orientation control of an industrial manipulator in the Cartesian space. The robot in question is the PUMA 560 arm and the control law on which we focused is the adaptive robust fuzzy control. We first present the different models of our robot: geometrical (direct and inverse), kinematic (first order, second order and for both direct and inverse representation) and dynamic. The end effector of the robot is lead to follow two kinds of prescribed trajectories.

To attain our goal, we applied three control laws: we first began with the computed torque, and to make more robust we have choose the adaptive law, then we have passed to the adaptive robust fuzzy control. The global asymptotic stability for each control law was checked with Lyapunov's principal.

To illustrate the validity of this work, simulations are presented and results are discussed, with comparative study between the motion control systems developed.

**Key words:** PUMA 560, robot modelling, Cartesian trajectory, computed torque, direct adaptive control, centralised adaptive robust control, decentralised adaptive control, Lyapunov's principal.

# Remerciements

*Je tiens à remercier en premier lieu « ALLAH » Le Tout Puissant, Qui m'a donné la force le courage et la volonté pour mener à bien ce modeste travail*

*J'exprime ma profonde gratitude, mon grand respect et ma sincère reconnaissance à mon promoteur monsieur **H. CHKIREB** professeur à l'Ecole Nationale Supérieure Polytechnique pour avoir assumé la lourde responsabilité de m'encadrer, de m'avoir orienté et conseiller tout au long de ce travail ainsi pour la confiance qu'il m'a accordée*

*Je remercie l'ensemble de mes enseignants d'automatique de l'Ecole Nationale Supérieure Polytechnique*

*Je remercie vivement tous mes enseignants et encadreurs de l'Ecole Nationale Préparatoire aux Etudes d'Ingéniorat*

*Enfin, je remercie toute personne qui m'a soutenu de près ou de loin tout au long de mon parcours pour la réalisation de ce travail.*

# Dédicace

*Je dédie ce modeste travail :*

*A mes très chers parent que j'aime par dessus tout , qui par leurs sacrifices, leur amour, leur patience, leur soutien et leurs encouragements m'ont permis de toujours persévérer, de viser haut et surtout de toujours donner le meilleurs de moi-même malgré tous*

*A ma très chère « petite » sœur « Samiha » qui m'a toujours encouragé, qui a toujours su comment supporter mes folies et qui a toujours était là pour moi*

*A mes très adorables frères : Housseem Eddine, Fares, Yasser et Samer pour qui j'exprime mon amour et ma joie d'être leurs sœur*

*A ma très chère sœur « Nadia » que j'estime énormément et qui m'inspire le courage la volonté et la patience*

*A tous mes amis de L'ENPEI*

*A tous mes amis de L'ENP*

*A toutes mes amies de la cité universitaire « EL ALIA » avec qui j'ai appris comment s'accrocher à ses rêves malgré toutes les souffrances*

# Table des matières

## Table des matières

### Liste des figures et des tableaux

<b>INTRODUCTION GENERALE</b> -----	1
<b>CHAPITRE I : Commande des mouvements par couple calculé</b>	
1.1. Introduction-----	3
1.2. Modélisation du robot PUMA 560-----	3
1.2.1. Modélisation géométrique-----	4
1.2.1.1. Modèle géométrique direct-----	4
1.2.1.2. Modèle géométrique inverse-----	7
1.2.2. Modélisation cinématique-----	9
1.2.2.1. Modèle cinématique direct du premier ordre-----	9
1.2.2.1.a. Calcul de la matrice jacobienne de base-----	9
1.2.2.1.b. Analyse de l'espace de travail du robot PUMA 560-----	10
1.2.2.1.c. Analyse des singularité du robot PUMA 560-----	11
1.2.2.2. Modèle cinématique du second ordre-----	13
1.2.2.3. Modèle cinématique inverse-----	15
1.2.3. Modélisation dynamique-----	15
1.3. Génération des mouvements-----	17
1.3.1. Contraintes de la génération des mouvements-----	17
1.3.2. Description de la trajectoire planaire-----	18
1.3.3. Présentation et modélisation de la trajectoire spatiale-----	22
1.4. Synthèse de la loi de commande par couple calculé-----	24
1.4.1. Hypothèses-----	24
1.4.2. Synthèse de la loi de commande pour le modèle nominal-----	25
1.4.3. Etude de la robustesse de la commande par découplage non linéaire-----	27
1.5. Simulations et interprétations-----	28
1.5.1. Cas idéal : sans erreurs de modélisation-----	29
1.5.2. En présence d'une erreur de modélisation de 50%-----	31
Conclusion-----	32

## **CHAPITRE II : Synthèse de la loi de commande adaptative**

2.1. Introduction	33
2.2. Commande adaptative par découplage non linéaire	35
2.3. Commande adaptative passive	37
2.4. Simulations et interprétations	39
2.4.1. Commande par découplage non linéaire adaptative	40
2.4.1.1. Cas idéal : sans erreurs de modélisation	40
2.4.1.2. En présence d'une erreur de modélisation de 50%	42
2.4.2. Commande passive	43
2.4.2.1. Cas idéal : sans erreur de modélisation	43
2.4.2.2. En présence d'une erreur de modélisation de 50%	45
Conclusion	46

## **CHAPITRE III : Synthèse de la loi de commande adaptative floue**

3.1. Introduction	47
3.2. Commande adaptative centralisée	48
3.2.1. Commande adaptative centralisée indirecte par les systèmes flous	51
3.2.1.1. Position du problème	51
3.2.1.2. Simulations et interprétations	54
3.2.1.2.1. Cas idéal : sans erreur de modélisation	55
3.2.1.2.2. En présence d'une erreur de modélisation de 50%	56
3.2.2. Commande adaptative directe centralisée floue	57
3.2.2.1. Position du problème	57
3.2.2.2. Simulations et interprétations	60
3.2.2.2.1. Cas idéal : sans erreur de modélisation	60
3.2.2.2.2. En présence d'une erreur de modélisation de 50%	62
3.3. Commande adaptative floue décentralisée	63
3.3.1. Commande adaptative indirecte décentralisée floue	63
3.3.1.1. Position du problème	63
3.3.1.2. Simulations et interprétations	68
3.3.1.2.1. Cas idéal : sans erreur de modélisation	68
3.3.1.2.2. En présence d'une erreur de modélisation de 50%	69

3.3.2. Commande adaptative décentralisée directe floue-----	71
3.3.2.1. Position du problème-----	72
3.3.2.2. Simulations et interprétations-----	73
3.3.2.2.1. Cas idéal : sans erreur de modélisation-----	74
3.3.2.2.2. En présence d'une erreur de modélisation de 50%-----	75
Conclusion-----	76
<b>CONCLUSION GENERALE</b> -----	77
<b>ANNEXE A</b> : Paramètres du robot PUMA 560-----	79
<b>ANNEXE B</b> : Systèmes flous et notions sur la stabilité des systèmes-----	81
<b>BIBLIOGRAPHIE</b> -----	95

## Liste des figures et tableaux:

### Liste des figures :

#### CHAPITRE I :

- Fig.1.1.** Systèmes de coordonnées et paramètres de Denavit-Hartenberg affectées au PUMA 560
- Fig.1.2.** Multiplicité de solutions du MGI pour le robot PUMA 560.
- Fig.1.3.** Espace de travail du robot PUMA.
- Fig.1.4.** Singularité d'épaule du robot PUMA.
- Fig.1.5.** Singularité du coude du robot PUMA.
- Fig.1.6.** Singularité du poignet rotule.
- Fig.1.7.** Différents repères associés au robot et à son environnement.
- Fig.1.8.** Schéma représentatif de la trajectoire planaire désirée.
- Fig.1.9.** Profil de vitesse et d'accélération souhaité pour la trajectoire planaire.
- Fig.1.10.** Génération de la trajectoire spatiale.
- Fig.1.11.** Structure de commande avec correction dans l'espace articulaire
- Fig.1.12.** Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande par découplage non linéaire en absence de variation des paramètres.
- Fig.1.13.** Signaux d'erreurs en position et orientation pour la trajectoire planaire avec la commande par découplage non linéaire en absence de variation des paramètres
- Fig.1.14.** Signaux des erreurs de position et d'orientation pour la trajectoire 3D avec la commande par découplage non linéaire en absence de variation des paramètres
- Fig.1.15.** Réponse du système et signaux de commande pour la trajectoire planaire avec la commande par couple calculé pour une variation des paramètres de 50%
- Fig.1.16.** Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la commande par découplage non linéaire en absence de variation des paramètres

#### Chapitre II :

- Fig.2.1.** Commande par découplage non linéaire adaptative (d'après Craig)
- Fig. 2.2 :** Schéma du régulateur adaptatif passif (d'après Slotine)
- Fig.2.3.** Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande par découplage non linéaire adaptative appliquée au système nominal

**Fig.2.4.** Signaux d'erreurs en position et orientation pour la trajectoire planaire avec la commande par découplage non linéaire adaptative appliquée au système nominal.

**Fig.2.5.** Réponse du système et signaux de commande pour la trajectoire planaire avec la commande par découplage non linéaire adaptative pour une variation des paramètres de 50%

**Fig.2.6.** Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la commande par découplage non linéaire adaptative avec une variation des paramètres de 50%.

**Fig.2.7.** Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande adaptative passive appliquée au système nominal

**Fig.2.8.** Signaux d'erreurs en position et orientation pour la trajectoire planaire avec la commande adaptative passive appliquée au système nominal.

**Fig.2.9.** Réponse du système et signaux de commande pour la trajectoire planaire avec la commande adaptative passive pour une variation des paramètres de 50%

**Fig.2.10.** Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la commande adaptative passive pour une variation des paramètres de 50%.

### **Chapitre III :**

**Fig.3.1.** Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande adaptative floue centralisée indirecte appliquée au système nominal

**Fig.3.2.** Signaux des erreurs en position et orientation pour la trajectoire planaire avec la commande adaptative floue centralisée indirecte appliquée au système nominal.

**Fig.3.3.** Réponse du système et signaux de commande pour la trajectoire planaire avec la commande adaptative floue indirecte centralisée pour une variation des paramètres de 50%

**Fig.3.4.** Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la commande adaptative floue indirecte centralisée avec une variation des paramètres de 50%.

**Fig.3.5.** Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande adaptative floue centralisée directe appliquée au système nominal

**Fig.3.6.** Signaux des erreurs en position et orientation pour la trajectoire planaire avec la commande adaptative floue centralisée directe appliquée au système nominal.

**Fig.3.7.** Réponse du système et signaux de commande pour la trajectoire planaire avec la commande adaptative floue directe centralisée pour une variation des paramètres de 50%

**Fig.3.8.** Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la commande adaptative floue directe centralisée avec une variation des paramètres de 50%.

**Fig.3.9.** Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande adaptative décentralisée indirecte appliquée au système nominal

**Fig.3.10.** Signaux des erreurs en position et orientation pour la trajectoire planaire avec la commande adaptative décentralisée indirecte appliquée au système nominal.

**Fig.3.11.** Réponse du système et signaux de commande pour la trajectoire planaire avec la commande adaptative décentralisée indirecte pour une variation des paramètres de 50%

**Fig.3.12.** Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la Commande adaptative décentralisée indirecte avec une variation des paramètres de 50%.

**Fig.3.13.** Réponse du système et signaux de commande pour la trajectoire planaire avec la commande adaptative décentralisée directe pour le système nominal.

**Fig.3.14.** Signaux des erreurs en position et orientation pour la trajectoire planaire avec la commande adaptative décentralisée directe appliquée au système nominal.

**Fig.3.15.** Réponse du système et signaux de commande pour la trajectoire planaire avec la commande adaptative décentralisée directe pour une variation des paramètres de 50%

**Fig.3.16.** Signaux des erreurs en position et orientation pour la trajectoire planaire avec la commande adaptative décentralisée directe appliquée au système nominal.

## **Liste des tableaux :**

**Tableau I.1.** Paramètres de Denavit-Hartenberg du bras de robot PUMA 560.

**Tableau A.1.** Butées du Robot PUMA 560

**Tableau A.2.** Masse de chaque corps constituant le robot PUMA 5

**Tableau A.3.** Tableau A.3. Coordonnées du centre de gravité de chaque corps du robot PUMA 560

**Tableau A.4.** Les principaux moments d'inertie de chaque corps du robot PUMA 560

**Tableau A.5.** Couple articulaire maximal pour chaque liaison du robot PUMA 560

## INTRODUCTION GENERALE

Depuis l'avènement des technologies, l'homme a voulu se décharger des tâches répétitives et inintelligibles en cherchant à automatiser tout processus de production et cela en développant des systèmes de réglages automatiques et de gestion des tâches. L'avancement de telles idées a donné naissance à la robotique. Cette branche a permis le regroupement de plusieurs domaines de connaissances tels que l'automatique, l'électronique, l'électrotechnique et la mécanique. Les objectifs principaux de la conception des robots industriels sont de réaliser des tâches avec rapidité et une grande précision. De nombreux travaux de recherche, basés essentiellement sur les stratégies de commande et de leur implémentation, ont vu le jour.

La problématique de la commande des robots manipulateurs, systèmes non linéaires à grande dimension interconnectés, est à l'heure actuelle l'une des préoccupations majeures et l'un des axes de recherches privilégiés. En effet, la complexité de la dynamique des robots, due essentiellement à l'interconnexion entre les différents axes du robot mais aussi aux variations paramétriques par rapport à la charge manipulée généralement inconnue, rend le contrôle pratiquement impossible par les régulateurs standards (classiques).

Par ailleurs, les exigences de l'industrie concernant les performances statiques et dynamiques en rapidité et en précision imposent des manipulateurs de plus en plus légers, plus stables et plus productifs.

L'introduction du formalisme de la géométrie différentielle et en particulier la technique de linéarisation entrée-sortie (connue aussi sous le nom de découplage non linéaire ou encore la commande par couple calculé « computed torque ») a connu un grand succès. Toutefois, cette loi de commande ne peut être utilisée que pour un modèle parfaitement connu. Pour pallier ce problème, plusieurs approches de commandes robustes (adaptatives ou modes de glissement) ont été introduites. Ces approches, suppose que le modèle du système non linéaire peut être exprimé sous la forme de fonctions non linéaires de formes connues mais à paramètres (constants ou lentement variables) inconnus. Malheureusement, il est difficile de décrire toutes les dynamiques d'un système à partir de fonctions connues, notamment, pour un système physique complexe. Une solution à ce problème est l'approximation de la dynamique du robot par un système flou où les relations entre entrées, sorties et états du système sont exprimées par des règles floues de types «Si-alors ».

Ainsi, l'objectif de ce présent travail porte sur l'utilisation des systèmes flous dans le cadre de la commande adaptative en position orientation du bras manipulateur rigide série PUMA 560.

Le robot PUMA 560 (Programmable Universal Machine for Assembly) est un bras manipulateur à chaîne ouverte simple à six articulations rotoïdes, de porteur anthropomorphe (RRR) et poignet rotule (poignet à 3 axes concourants). Disposant, ainsi, de six degrés de libertés (6 DOF) lui permettant, pourvu qu'il ne sort pas de son espace de travail, de positionner et d'orienter son organe terminal (3 degrés de liberté pour le positionnement, les 3 autres pour l'orientation) dans n'importe quelle configuration dans l'espace tridimensionnel, où il est non redondant.

Afin d'atteindre notre objectif, nous allons développer en premier lieu la synthèse de la loi de commande par couple calculé. Ce concept est la base de développement de plusieurs approches ultérieures. Aussi les différents niveaux de modélisation du robot et la tâche qu'il doit exécuter seront présentés.

Dans le deuxième chapitre, afin d'amener le robot à accomplir sa tâche en dépit de toutes modification des paramètres (intrinsèques ou externes), la commande adaptative est proposée. Son but est d'ajuster ou d'estimer en ligne les paramètres utilisés dans la loi de commande.

De nos jours, on peut considérer que la commande adaptative a atteint une bonne maturité et plusieurs stratégies sont proposées dans la littérature. Dans notre travail, nous allons nous intéresser à la commande par découplage non linéaire adaptative et la commande passive.

La contrainte majeure dont souffre cette approche (la commande adaptative) est le fait qu'elle suppose connue la forme du modèle du système, hors ce n'est pas le cas pour des systèmes complexes interconnectés tels que les robots. Dans cette perspective, nous allons utiliser la propriété d'approximateurs universels des systèmes flous pour la synthèse de lois de commandes plus performantes. C'est l'objectif du troisième chapitre où nous présentons la commande adaptative floue centralisée et décentralisée.

Chaque chapitre sera accompagné d'un ensemble de simulations qui mettent en relief les performances et/ou les limites de chaque loi de commande.

Le système et sa commande sont supposés continus. Dans la pratique, la commande étant réalisée par ordinateur, il faut tenir compte d'un retard pour l'acquisition des informations et pour le calcul des consignes, cela relève de la théorie des systèmes discrets et sort ainsi du cadre de notre travail.

---

# 1

## COMMANDE DES MOUVEMENTS PAR COUPLE CALCULE

---

### 1. Introduction :

Un robot manipulateur ou simplement un manipulateur est une chaîne cinématique constituée de deux corps distincts, un (ou plusieurs) organe terminal et une structure mécanique articulée. Chacun possédant son propre espace. L'espace dans lequel est représentée la situation de toute la chaîne cinématique constituant la structure articulée est dit espace articulaire, il est aussi connu sous le nom d'espace de configuration, sa dimension est égale au nombre de variables articulaires indépendantes et correspond au nombre de degrés de liberté de la structure mécanique.

L'espace opérationnel est celui dans lequel est représentée la situation de l'organe terminal. La situation la plus simple consiste à utiliser les coordonnées cartésiennes dans  $R^3$  pour la position. Pour l'orientation, différentes représentations sont utilisées : les cosinus directeurs, les angles d'Euler et les quaternions [1].

Dans ce chapitre, on traite le problème de la commande par découplage non linéaire du robot dans l'espace libre, cette commande sera développée en vue de réaliser deux applications : la première simule la soudure sur un plan horizontal de deux plaques métalliques suivant un arc de cercle fermé. Dans la deuxième, La pointe du robot doit tracer une courbe définie dans le premier huitième d'une sphère.

Pour commander le robot ou pour simuler son comportement on doit disposer de modèles. Plusieurs niveaux de modélisation sont possibles selon les objectifs, les contraintes de la tâche et les performances recherchées. Ces modèles, qu'ils soient utilisés en simulation ou en commande, doivent représenter le comportement géométrique, cinématique et dynamique du robot de façon réaliste.

### 2. Modélisation du robot PUMA 560:

La modélisation du bras de robot permet la détermination des relations directes et inverses entre les coordonnées généralisées et les coordonnées cartésiennes et leurs dérivés respectives.

Dans le but de simplifier la modélisation des robots on admet les hypothèses suivantes :

- Les liaisons du manipulateur sont rigides.
- Les jeux dans les articulations sont négligeables.
- Les capteurs ont un gain unitaire et de dynamique négligeable.

Les différents principes utilisés pour la modélisation du robot PUMA 560 sont :

- La notation de Denavit-Hartenberg pour le modèle géométrique direct [2].
- La méthode de Pieper pour le modèle géométrique inverse [1], [2], [3].
- Le principe du jacobien de base pour le modèle cinématique direct du premier et second ordre [1], [2], [4].
- Le modèle dynamique du robot PUMA 560 utilisé dans ce travail est tiré des travaux d'O. Khatib, B. Armstrong et J. Burdick de l'université de Stanford [5].

## 2.1. Modélisation géométrique :

L'objectif est d'exprimer la situation de l'organe terminal du robot : position et orientation, en fonction des variables articulaires du mécanisme et inversement. Dans le premier cas, le modèle qui en découle est dit : « le modèle géométrique direct : MGD » et dans le second c'est : « le modèle géométrique inverse : MGI ».

### 2.1.1. Modèle géométrique direct :

Le modèle géométrique direct (MGD) exprime la situation de l'organe terminal en fonction des variables articulaires :  $X = f(q)$ . Il peut être représenté par la matrice homogène  $T_0^6$  :

$$T_0^6(q) = T_1^0(q) * T_2^1(q) * T_3^2(q) * T_4^3(q) * T_5^4(q) * T_6^5(q) \quad (1.1)$$

Où :  $T_i^{i-1}$  représente la matrice de transformation homogène définissant le repère  $\{R_i\}$  dans le repère  $\{R_{i-1}\}$ .

La représentation du robot PUMA 560 suivant la convention de Denavit-Hartenberg est donnée par la figure 1.1, les valeurs numériques sont regroupées dans le tableau 1 ci-dessous.

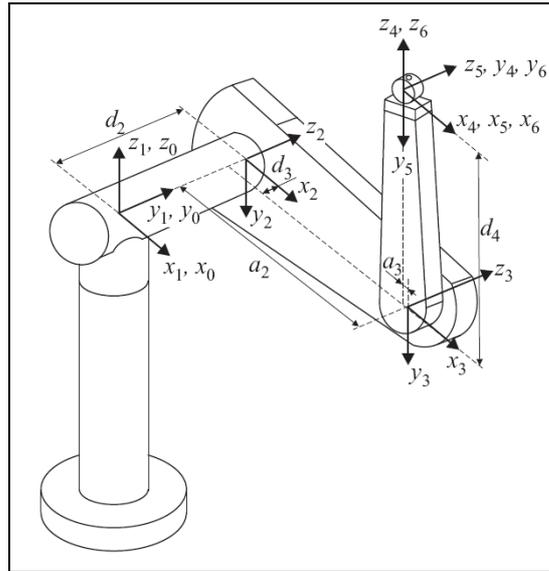


Fig.1.1. Système de coordonnées et paramètres de Denavit-Hartenberg affectés au PUMA 560

Numéro de la liaison « i »	$\alpha_{i-1}$ [degrés]	$\theta_{i-1}$ variable	$a_{i-1}$ [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

Tableau I.1. Paramètres de Denavit-Hartenberg du bras de robot PUMA 560

La distance  $d_6$  dépend de l'effecteur (l'outil attaché au poignet sur la dernière articulation pour la manipulation des objets). Dans notre application, la distance entre la fin de l'effecteur et l'axe du poignet est supposée égale à  $L_6 = 0.05625m$ .

Les transformations homogènes permettant l'élaboration du modèle géométrique direct sont les suivantes :

$$T_1^0 = \begin{bmatrix} c1 & -s1 & 0 & 0 \\ s1 & c1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_2^1 = \begin{bmatrix} c2 & -s2 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ -s2 & -c2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_3^2 = \begin{bmatrix} c3 & -s3 & a_3 \\ s3 & c3 & 0 & -d_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^3 = \begin{bmatrix} c4 & -s4 & 0 & a_3 \\ 0 & 0 & -1 & -d_4 \\ s4 & c4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_5^4 = \begin{bmatrix} c5 & -s5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s5 & -c5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_6^5 = \begin{bmatrix} c6 & -s6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s6 & c6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Avec :  $c_i = \cos(\theta_i)$  ;  $s_i = \sin(\theta_i)$  ;  $c_{ij} = \cos(\theta_i + \theta_j)$  ;  $s_{ij} = \sin(\theta_i + \theta_j)$

La matrice de transformation  $T_6^0$  définit le modèle géométrique direct, elle s'obtient par la relation (1.1). Grâce au calcul symbolique qu'offre le logiciel MATLAB<sup>®</sup>, l'expression de  $T_6^0$  est facilement obtenue :

$$T_6^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Avec :

$$r_{11} = [(c_1 c_{23} c_4 - s_1 s_4) c_5 - c_1 s_{23} s_5] c_6 - (c_1 c_{23} s_4 + s_1 c_4) s_6$$

$$r_{12} = [(c_1 c_{23} c_4 - s_1 s_4) c_5 - c_1 s_{23} s_5] c_6 - (c_1 c_{23} s_4 + s_1 s_4) c_6$$

$$r_{13} = [(c_1 c_{23} c_4 - s_1 s_4) s_5 + c_1 s_{23} s_5]$$

$$r_{21} = [(s_1 c_{23} c_4 + c_1 c_4) c_5 - s_1 s_{23} s_5] c_6 + (-s_1 c_{23} s_4 + c_1 s_4) s_6$$

$$r_{22} = -[(s_1 c_{23} c_4 + c_1 s_4) c_5 - s_1 s_{23} s_5] s_6 + (-s_1 c_{23} s_4 + c_1 c_4) c_6$$

$$r_{23} = [(s_1 c_{23} c_4 + c_1 c_4) s_5 + s_1 s_{23} c_5]$$

$$r_{31} = [(-s_{23} c_4 c_5 + c_{23} s_5) c_6 + s_4 s_{23} s_6]$$

$$r_{32} = [(s_{23} c_4 c_5 - c_{23} s_5) s_6 + s_4 s_{23} c_6]$$

$$r_{33} = s_{23} c_4 c_5 + c_{23} c_5$$

$$p_x = a_2 c_1 c_2 - (d_2 + d_3) s_1 + a_3 c_1 c_{23} + d_4 c_1 s_{23}$$

$$p_y = a_2 s_1 c_2 + (d_2 + d_3) c_1 + a_3 s_1 c_{23} + d_4 s_1 s_{23}$$

$$p_z = -a_2 s_2 - a_3 s_{23} - d_4 c_{23}$$

### 2.1.2. Modèle géométrique inverse :

Le modèle géométrique inverse (MGI) permet de calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Plus précisément, c'est la forme explicite, qui lorsqu'elle existe, donne toutes les solutions possibles dans l'espace articulaire (il y a rarement unicité de solution) à une situation donnée de l'effecteur dans l'espace opérationnel.

Plusieurs méthodes de calculs sont disponibles dans la littérature [1], [2], [3], [4], dans notre travail on considère la méthode de Paul qui facilite énormément le calcul pour les robots à poignet rotule (le cas du PUMA 560 et de la plupart des robots industriels). Dans ce type de structure, la position du centre de la rotule est uniquement fonction des trois premières variables articulaires :  $q_1$ ,  $q_2$  et  $q_3$ . Cela, assure le découplage position/orientation et conduit à considérer deux ensembles d'équations à trois inconnues. La résolution de ses systèmes d'équations donne les solutions suivantes :

$$\begin{cases} c1 = \frac{P_y(d_2 - d_3) \pm P_x \sqrt{P_x^2 + P_y^2 - (d_2 - d_3)^2}}{P_x^2 + P_y^2} \\ s1 = \frac{-P_x(d_2 - d_3) \pm P_y \sqrt{P_x^2 + P_y^2 - (d_2 - d_3)^2}}{P_x^2 + P_y^2} \\ q_1 = \text{atan2}(s1, c1) \end{cases}$$

$$\begin{cases} c2 = \frac{YZ \pm X \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2} \\ s2 = \frac{XZ \pm Y \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2} \\ q_2 = \text{atan2}(s2, c2) \end{cases} \quad \text{Avec : } \begin{cases} X = 2 a_2 P_x \\ Y = -2 a_2 (c1 P_x + s1 P_y) \\ Z = a_3^2 + d_4^2 - (c1 P_x + s1 P_y)^2 - P_z^2 - a_2^2 \end{cases}$$

$$\begin{cases} c3 = \frac{(c2 (c1 P_x + s1 P_y) - s2 P_z - a_2) d_4 - a_3 (-s2 (c1 P_x + s1 P_y) - c2 P_z)}{d_4^2 + a_3^2} \\ s3 = \frac{-(c2 (c1 P_x + s1 P_y) - s2 P_z - a_2) a_3 - d_4 (-s2 (c1 P_x + s1 P_y) - c2 P_z)}{d_4^2 + a_3^2} \\ q_3 = \text{atan2}(s3, c3) \end{cases}$$

$$\begin{cases} c4 = c1 c2 r_{13} + s1 c23 r_{23} - s23 r_{23} \\ s4 = -s1 r_{13} + s1 c23 r_{23} \\ q_4 = \text{atan2}(s4, c4) \end{cases} \quad \begin{cases} c5 = c1 s23 r_{13} + s1 s23 r_{23} + c23 r_{33} \\ s5 = r_{13} [c1 c23 c4 - s1 s4] + r_{23} s23 c4 \\ q_5 = \text{atan2}(s5, c5) \end{cases}$$

$$\begin{cases} c6 = -r_{12} (c1 c23 s4 + s1 c4) + r_{21} (-s1 c23 s4 + c1 c4) + r_{31} s23 s4 \\ s6 = -r_{12} [c1 c23 s4 - s1 c4] + r_{32} s23 s4 + r_{22} (-s1 c23 s4 + c1 c4) \\ q_6 = \text{atan2}(s6, c6) \end{cases}$$

Notons que : **atan2** est une fonction mathématique, qui à partir de ses deux arguments calcule l'angle correspondant et le donne entre  $[-180^\circ, 180^\circ]$ , avec une précision uniforme sur tout le domaine de définition.

Les confusions majeures que pose le modèle géométrique inverse sont :

1. La non unicité du modèle géométrique inverse. Pour le robot PUMA 560, il existe huit solutions [6], la figure 1.2 illustre quatre de ses solutions : coudé vers le bas (a, b) ou vers le haut (c, d), épaule vers l'avant (a, c), vers l'arrière (b, d).
2. Le traitement par incrément peut amener à des imprécisions
3. Des singularités mécaniques ou mathématiques apparaissent.

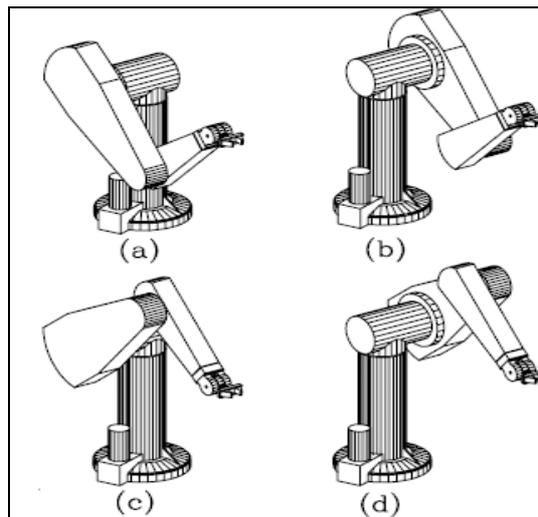


Fig.1.2. Multiplicité de solutions du MGI.

Cependant, ils sont contournables de la manière suivante :

1. S'agissant du premier inconvénient, étant donné que le robot se déplacera à partir d'une position connue vers une autre aussi connue, il est superflu de calculer toutes les configurations qui seraient impossibles à atteindre.
2. Grâce aux nouveaux calculateurs, une haute précision des calculs ne pose plus de problèmes.
3. Quant au problème des singularités, il existe plusieurs méthodes pour les éviter.

## 2.2. Modélisation cinématique :

Le modèle cinématique exprime les relations entre vitesses et accélérations articulaires de chaque joint et vitesses et accélérations cartésiennes d'un point de la chaîne cinématique, généralement l'organe terminal. Ce modèle est donc un modèle par accroissement élémentaire : chaque variation élémentaire de la vitesse/accélération d'une articulation implique la variation de celle de l'organe terminal, et inversement.

### 2.2.1. Modèle cinématique direct du premier ordre :

Le modèle cinématique direct du premier ordre d'un robot-manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires [1]. Il est noté :

$$\dot{X} = J(q) \dot{q} \quad (1.2)$$

Où  $J(q)$  désigne la matrice jacobienne du mécanisme, égale à  $\frac{dX}{dq}$  et fonction de la configuration articulaire  $q$ . Pour le cas des robots série, elle est de dimension  $n \times n$ .

Ainsi, les vecteurs des vitesses de translation et de rotation  $V_n$  et  $\omega_n$  du repère  $\{R_n\}$  sont données par :  $\begin{bmatrix} V_n \\ \omega_n \end{bmatrix} = J(q) \dot{q}$ . L'outil principal dans la modélisation cinématique des robots manipulateurs est la matrice jacobienne.

#### 2.2.1.a. Calcul de la matrice jacobienne de base :

Il existe plusieurs méthodes pour calculer la matrice jacobienne. Hormis les méthodes de calcul symbolique, le calcul de la matrice jacobienne peut se faire en dérivant directement le modèle géométrique direct. Cette approche est facile à mettre en œuvre pour les robots à deux ou trois degrés de liberté, pour ceux ayant plus de trois, le calcul du jacobien de base est plus pratique.

Les expressions des colonnes de la matrice jacobienne du robot PUMA 560 sont construites en appliquant les principes détaillés dans [1], [3] et grâce au calcul symbolique dont dispose le logiciel MATLAB<sup>®</sup>. Elles sont données par :

$$\begin{cases} J_{11} = -(a_3 c_{23} + d_4 s_{23} + a_2 c_2)s_1 - (d_2 + d_3)c_1 \\ J_{21} = (a_3 c_{23} + d_4 s_{23} + a_2 c_2)c_1 - (d_2 + d_3)c_1 \\ J_{31} = 0 \\ J_{41} = 0 \\ J_{51} = 0 \\ J_{61} = 0 \end{cases}$$

$$\begin{cases} J_{12} = -a_3 c_1 s_{23} + d_4 c_1 c_{23} - a_2 c_1 s_2 \\ J_{22} = -a_3 s_1 s_{23} + d_4 s_1 c_{23} - a_2 s_1 s_2 \\ J_{32} = -a_3 c_{23} - d_4 s_{23} - a_2 c_2 \\ J_{42} = -s_1 \\ J_{52} = c_1 \\ J_{62} = 0 \end{cases} \quad \begin{cases} J_{13} = d_4 c_1 c_{23} - a_3 c_1 s_{23} \\ J_{23} = d_4 s_1 c_{23} - a_3 s_1 s_{23} \\ J_{33} = -d_4 s_{23} - a_3 c_{23} \\ J_{43} = -s_1 \\ J_{53} = c_1 \\ J_{63} = 0 \end{cases}$$

$$\begin{cases} J_{14} = 0 \\ J_{24} = 0 \\ J_{34} = 0 \\ J_{44} = c_1 s_{23} \\ J_{54} = s_1 s_{23} \\ J_{64} = c_{23} \end{cases} \quad \begin{cases} J_{15} = 0 \\ J_{25} = 0 \\ J_{35} = 0 \\ J_{45} = -c_1 c_{23} s_4 - s_1 c_4 \\ J_{55} = -s_1 c_{23} s_4 + c_1 c_4 \\ J_{65} = s_{23} s_4 \end{cases} \quad \begin{cases} J_{16} = 0 \\ J_{26} = 0 \\ J_{36} = 0 \\ J_{46} = c_1 (c_{23} c_4 s_5 + s_{23} c_5) - s_1 s_4 s_5 \\ J_{56} = s_1 (c_{23} c_4 s_5 + s_{23} c_5) + c_1 s_4 s_5 \\ J_{66} = -s_{23} c_4 s_5 + c_{23} c_5 \end{cases}$$

### 2.2.1.b. Analyse de l'espace de travail du robot PUMA 560:

L'espace de travail est l'ensemble des positions et/ou orientations accessibles par l'organe terminal. La géométrie de l'espace de travail dépend de la morphologie du robot. Ses frontières sont définies par les singularités, les butées articulaires et les obstacles existant dans le champ d'action du robot [1]. Les sous-espaces t-parcourables sont les domaines de l'espace de travail où toute trajectoire continue est réalisable. L'espace de travail du robot PUMA 560 est de dimension 6 et ne peut donc être représenté de façon simple. On se contente de donner ses projections dans l'espace des positions (figure 1.3) [3].

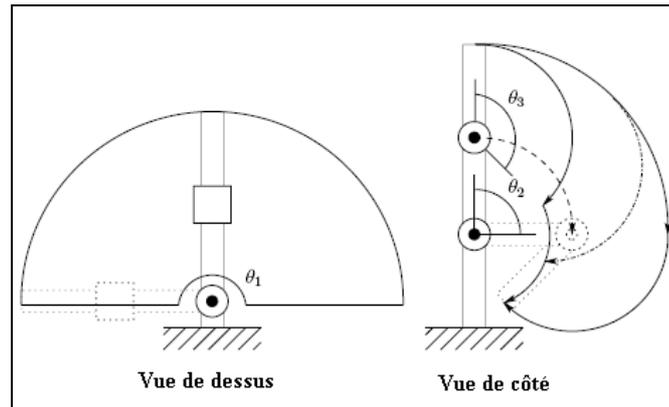


Fig.1.3. Espace de travail du robot PUMA

### 2.2.1.c. Analyse des singularités du robot:

Pour tous les robots, qu'ils soient redondants ou non, il se peut que dans certaines configurations dites *singulières*, le nombre de degrés de liberté de l'organe terminal devient inférieur à la dimension de l'espace opérationnel. Pour les robots à articulations rotoïdes, ce cas se présente lorsque deux liaisons se retrouvent confondues. Analytiquement, les singularités sont définies par l'équation  $\det(J) = 0$ . Les singularités du robot PUMA 560 sont données par:

$$\det(J) = a_2(a_3 c_{23} + d_4 s_{23} + a_2 c_2)(d_4 c_3 - a_3 s_3) s_5 = 0$$

Dont les solutions s'interprètent par les trois cas suivant [1] [5] [6] :

- **Singularité d'épaule :** C'est lorsque  $(a_3 c_{23} + d_4 s_{23} + a_2 c_2 = 0)$ . Dans cette configuration le centre du poignet est confondu avec l'axe  $z_0$  (figure 1.4). Dans ce cas, le modèle cinématique ne permet pas de commander une vitesse selon la normale au plan contenant les points  $O_2, O_3$  et  $O_4$ . La troisième ligne du jacobien est nulle.

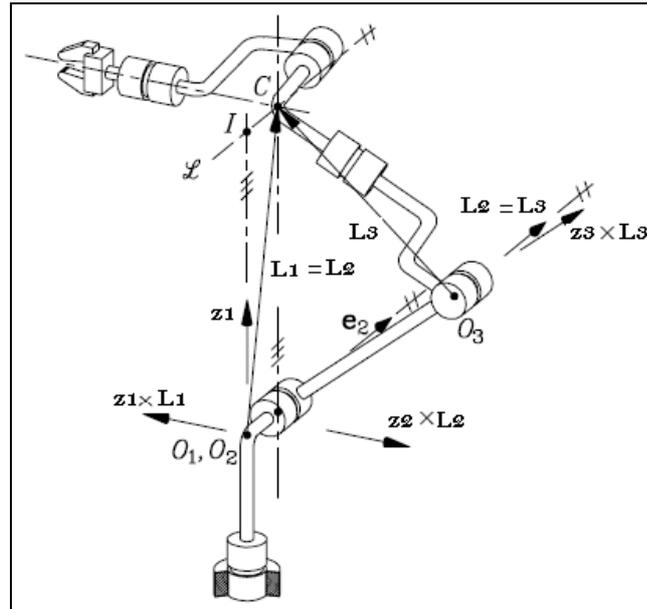


Fig.1.4. Singularité d'épaule du robot PUMA

- **Singularité du coude :** Quand  $(d_4 c_3 - a_3 s_3 = 0)$ . Dans cette configuration, le modèle cinématique est incapable de commander une vitesse radiale au centre du poignet. Le robot se trouve en extension maximale (figure 1.5). Le centre  $C$  du poignet se trouve dans le plan d'intersection des deux axes  $z_1$  et  $z_2$ . La deuxième ligne du jacobien est nulle.

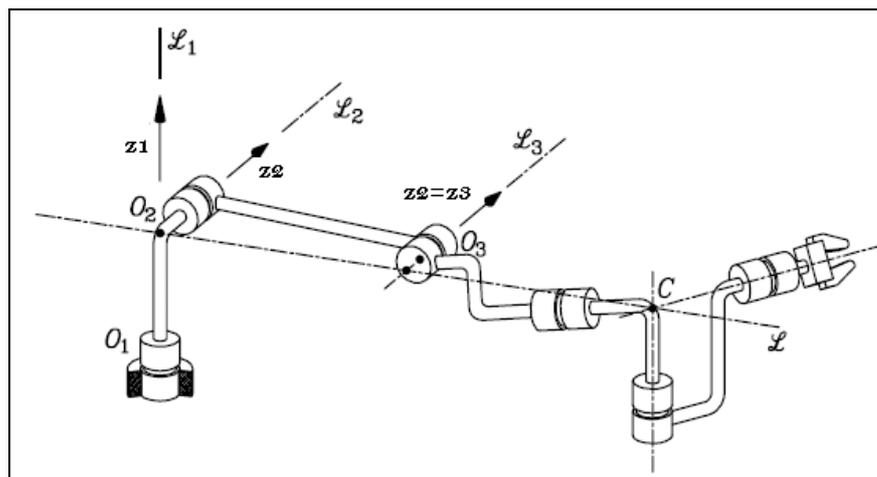


Fig.1.5. Singularité du coude du robot PUMA.

- **Singularité du poignet :** Lorsque  $(s_5 = 0)$ , les deux articulations rotoïdes commandant  $q_4$  et  $q_6$  ont leurs axes confondus, ce qui fait perdre un degré de liberté au robot (figure 1.6). Dans cette configuration, le modèle cinématique ne permet pas de commander une rotation autour de la normale au plan contenant les axes 4, 5 et 6. La quatrième et la sixième colonne du jacobien sont identiques.

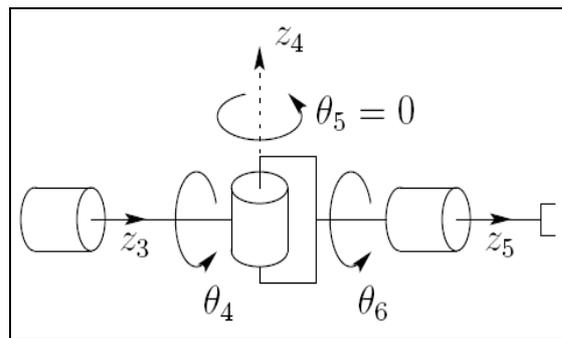


Fig.1.6. Singularité du poignet rotule

### 2.2.2. Modèle cinématique du second ordre :

Le modèle cinématique du deuxième ordre permet de calculer les accélérations des coordonnées opérationnelles en fonction des positions, vitesses et accélérations articulaires [1]. Dans le cas d'un mécanisme à chaîne ouverte simple, on l'obtient par dérivation du modèle cinématique direct :

$$\ddot{X} = J(q) \ddot{q} + \dot{J}(q) \dot{q} \quad (1.3)$$

Où :  $J(q) = \frac{dJ(q)}{dt}$

Le terme essentiel dans l'expression (1.3) est la dérivée de la matrice jacobienne. Elle est donnée par les expressions ci-dessous. Ces relations ont été établies en utilisant le calcul différentiel que fournit le logiciel Matlab sur des expressions symboliques.

$$\begin{cases} \dot{j}_{11} = \dot{q}_1[(d_2 + d_3) s_1 - (a_3 c_{23} + d_4 s_{23} + a_2 c_2) c_1] - s_1(\dot{q}_2 + \dot{q}_3)(-a_3 s_{23} + d_4 c_{23}) + \\ \quad + a_2 \dot{q}_2 s_1 s_2 \\ \dot{j}_{21} = \dot{q}_1[-(a_3 c_{23} + d_4 s_{23} + a_2 c_2) s_1 - (d_2 + d_3) c_1] + c_1 [(\dot{q}_2 + \dot{q}_3)(d_4 c_{23} - a_3 s_{23}) - a_2 \dot{q}_2 s_2] \\ \dot{j}_{31} = 0 \\ \dot{j}_{41} = 0 \\ \dot{j}_{51} = 0 \\ \dot{j}_{61} = 0 \end{cases}$$

$$\begin{cases} \dot{j}_{12} = \dot{q}_1[(d_2 + d_3) s_1 - (a_3 c_{23} + d_4 s_{23} + a_2 c_2) c_1] - s_1(\dot{q}_2 + \dot{q}_3)(-a_3 s_{23} + d_4 c_{23}) \\ \dot{j}_{22} = \dot{q}_1 c_1(-a_3 s_{23} + d_4 c_{23} - a_2 s_2) - c_1 [(\dot{q}_2 + \dot{q}_3)(d_4 c_{23} + a_3 s_{23}) + a_2 \dot{q}_2 s_2] \\ \dot{j}_{32} = (\dot{q}_2 + \dot{q}_3)(-d_4 c_{23} + a_3 s_{23}) + a_2 \dot{q}_2 s_2 \\ \dot{j}_{42} = -\dot{q}_1 c_1 \\ \dot{j}_{52} = -\dot{q}_1 s_1 \\ \dot{j}_{62} = 0 \end{cases}$$

$$\begin{cases} \dot{j}_{13} = -\dot{q}_1 s_1[-(a_3 s_{23} - d_4 c_{23})] - c_1(\dot{q}_2 + \dot{q}_3)[(a_3 c_{23} + d_4 s_{23})] \\ \dot{j}_{23} = \dot{q}_1 c_1(-a_3 s_{23} + d_4 c_{23}) - s_1 [(\dot{q}_2 + \dot{q}_3)(d_4 c_{23} + a_3 s_{23})] \\ \dot{j}_{33} = (\dot{q}_2 + \dot{q}_3)(-d_4 c_{23} + a_3 s_{23}) \\ \dot{j}_{43} = -\dot{q}_1 c_1 \\ \dot{j}_{53} = -\dot{q}_1 s_1 \\ \dot{j}_{63} = 0 \end{cases}$$

$$\begin{cases} \dot{j}_{14} = 0 \\ \dot{j}_{24} = 0 \\ \dot{j}_{34} = 0 \\ \dot{j}_{44} = -\dot{q}_1 s_1 s_{23} + (\dot{q}_2 + \dot{q}_3) c_1 c_{23} \\ \dot{j}_{54} = \dot{q}_1 c_1 s_{23} + (\dot{q}_2 + \dot{q}_3) s_1 c_{23} \\ \dot{j}_{64} = -(\dot{q}_2 + \dot{q}_3) s_{23} \end{cases}$$

$$\begin{cases} \dot{j}_{15} = 0 \\ \dot{j}_{25} = 0 \\ \dot{j}_{35} = 0 \\ \dot{j}_{45} = \dot{q}_1 (s_1 c_{23} s_4 - c_1 c_4) + (\dot{q}_2 + \dot{q}_3) c_1 s_{23} s_4 + \dot{q}_4 (s_1 s_4 - c_1 c_{23} c_4) \\ \dot{j}_{55} = -\dot{q}_1 (s_1 c_{23} s_4 + c_1 c_4) + (\dot{q}_2 + \dot{q}_3) s_1 s_{23} s_4 - \dot{q}_4 (c_1 s_4 + s_1 c_{23} c_4) \\ \dot{j}_{65} = (\dot{q}_2 + \dot{q}_3) c_{23} s_4 + \dot{q}_4 s_{23} c_4 \end{cases}$$

$$\begin{cases} \dot{j}_{16} = 0 \\ \dot{j}_{26} = 0 \\ \dot{j}_{36} = 0 \\ \dot{j}_{46} = -\dot{q}_1 (s_1 (c_{23} c_4 s_5 + s_{23} c_5) + c_1 s_4 s_5) - \dot{q}_4 s_1 c_4 s_5 - \dot{q}_5 s_1 s_4 c_5 + \\ \quad c_1 [(\dot{q}_2 + \dot{q}_3) (-s_{23} c_4 s_5 + c_{23} c_5) - \dot{q}_4 c_{23} s_4 s_5 + \dot{q}_5 (c_{23} c_4 c_5 - s_{23} s_5)] \\ \dot{j}_{56} = -\dot{q}_1 (c_1 (c_{23} c_4 s_5 + s_{23} c_5) - s_1 s_4 s_5) + \dot{q}_4 c_1 c_4 s_5 + \dot{q}_5 c_1 s_4 c_5 + \\ \quad s_1 [(\dot{q}_2 + \dot{q}_3) (-s_{23} c_4 s_5 + c_{23} c_5) - \dot{q}_4 c_{23} s_4 s_5 + \dot{q}_5 (c_{23} c_4 c_5 - s_{23} s_5)] \\ \dot{j}_{66} = -\dot{q}_4 s_{23} s_4 s_5 - (\dot{q}_2 + \dot{q}_3) (c_{23} c_4 s_5 + s_{23} c_5) - \dot{q}_5 (s_{23} c_4 c_5 + c_{23} s_5) \end{cases}$$

### 2.2.3. Modèle cinématique inverse :

L'objectif du modèle cinématique inverse est de calculer à partir d'une configuration  $q$  donnée, les vitesses articulaires  $\dot{q}$  qui assurent au repère terminal une vitesse opérationnelle  $\dot{X}$  imposée.

Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique direct en résolvant un système d'équations linéaires.

Le modèle cinématique inverse est donné par :  $\dot{q} = J(q)^{-1} \dot{X}$  (1.4)

Le calcul se fait directement par des instructions du logiciel Matlab, sans avoir besoin de développer les expressions analytiques.

### 2.3. Modélisation dynamique :

Le modèle dynamique (modèle dynamique inverse) des robots-manipulateur est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires. On le représente par la relation générale suivante :  $\Gamma = f(q, \dot{q}, \ddot{q}, f_e)$

Avec :

- $\Gamma$  vecteur des couples/forces des actionneurs.
- $f_e$  vecteur représentant l'effort extérieur qu'exerce le robot sur son environnement. Par la suite, comme notre centre d'intérêt est la commande en position/orientation du robot ce terme sera négligé.

Plusieurs formalismes ont été utilisés pour obtenir le modèle dynamique des robots, les plus souvent utilisés sont :

- Formalisme d'Euler-Lagrange.

- Le formalisme de Newton-Euler.
- La méthode de Kane.

Notons que pour un robot série ses approches sont équivalentes et aboutissent toutes à la même modélisation [2]. Dans la pratique, vue la complexité des calculs, des méthodes de simplifications ont été développées permettant la réduction du temps de calcul et de l'espace mémoire nécessaire [5] [6].

Le modèle dynamique d'un bras de robot rigide à  $n$  degrés de liberté est donné par un système d'équations différentielles du second ordre à  $n$  entrées formant le vecteur des couples généralisés  $\Gamma$  et  $n$  sortie composant le vecteur position  $q$ . Les équations de ce système sont représentées sous la forme matricielle suivante [2] [5] :

$$A(q)\ddot{q} + B(q)[\dot{q}\dot{q}] + C(q)[\dot{q}^2] + G(q) + F(\dot{q}) = \Gamma \quad (1.5)$$

Avec :

- $q \in R^n$ ,  $\dot{q} \in R^n$  et  $\ddot{q} \in R^n$  sont respectivement les positions, vitesses et accélérations articulaires.
- $A(q) \in R^{n \times n}$  : Matrice symétrique définie positive. C'est la matrice de l'énergie cinétique, appelée aussi matrice d'inertie.
- $B(q) \in R^{n \times n(n-1)/2}$  : Matrice des couples de Coriolis.
- $C(q) \in R^{n \times n}$  : Matrice des couples centrifuges.
- $G(q) \in R^n$  : Vecteur des couples de gravité.
- $F(\dot{q}) \in R^n$  : Vecteur des frottements secs et visqueux.
- $[\dot{q}\dot{q}] \in R^{n(n-1)/2}$  : Vecteur des produits des vitesses généralisées.
- $[\dot{q}^2] \in R^n$  : Vecteur des carrés des vitesses généralisées.

L'équation (1.5) est non linéaire par rapport aux coordonnées généralisées  $q$ ,  $\dot{q}$  et  $\ddot{q}$ . Toute fois, ses éléments  $A, B, C, G$  et  $F$  vérifient certaines propriétés fondamentales pour l'analyse du comportement et la synthèse de la commande:

➤ *Propriété 1 :*

La matrice d'inertie  $A(q)$  est symétrique définie positive et par conséquent, elle est inversible.

➤ *Propriété 2 :*

Les matrices  $A, B, C$  et les vecteurs  $F$  et  $G$  sont uniformément bornées [1] [7].

➤ *Propriété 3 :*

L'entrée de commande est indépendante pour chaque articulation.

➤ *Propriété 4 :*

Le vecteur de frottement est de la forme :  $F_i(\dot{q}_i) = f_{vi} \dot{q}_i + f_{si} \text{sign}(\dot{q}_i) \quad i = 1 \dots 6$

Avec :  $f_{si}$  et  $f_{vi}$  sont respectivement les coefficients des frottements secs et visqueux.

➤ *Propriété 5 :*

Le modèle dynamique est linéaire vis-à-vis des éléments des paramètres inertiels.

Cette propriété est mise à profit lors de l'identification et de la commande [1].

➤ *Propriété 6 :*

Le robot est un système passif qui dissipe de l'énergie.

Les propriétés 1 à 4 découlent de la nature physique du robot manipulateur. La propriété 3 est due au fait que les flexibilités des articulations et des structures n'ont pas été prises en compte. Dans ce cas de figure chaque degré de liberté est piloté par un actionneur.

Le modèle dynamique du robot PUMA 560 est tiré à partir des travaux effectués par O.Khatib, B.Armstrong et J.Burdick [5].

### 3. Génération de mouvements :

L'objectif de la génération de mouvements est le calcul des consignes articulaires destinées à réaliser une tâche traduite sous forme de positions successives de l'outil du robot. Elle peut s'interpréter comme étant un problème d'optimisation visant à minimiser la durée du mouvement en tenant compte de toutes les contraintes cinématiques.

La génération de mouvement peut s'exercer aussi bien dans l'espace articulaire que dans l'espace opérationnel selon la nature et la complexité de la tâche à réaliser. Dans ce chapitre, nous avons pris deux exemples de la génération de mouvements continus dans l'espace cartésien, le premier exemple est développé dans le plan et le second dans l'espace, et pour chacun on a supposé que la nature de la tâche à exécuter par le robot nécessite une vitesse de type trapézoïdal. Ce type de vitesse dit aussi loi Bang-Bang [1] [8] avec palier de vitesse est optimal en temps.

La première trajectoire à suivre est un arc de cercle fermé sur un plan horizontal, exécuté avec un angle de  $45^\circ$  entre la pointe de l'outil et l'axe  $OZ_0$  perpendiculaire au plan. La deuxième est une courbe définie entre deux points appartenant au premier huitième d'une sphère.

### 3.1. Contraintes de la génération de mouvement :

La génération de mouvement doit respecter certaines conditions afin d'assurer le fonctionnement normal du mécanisme. Parmi ces conditions on cite :

- La trajectoire doit être spécifiée relativement au repère lié à la station de travail (repère  $\{S\}$  dans la figure 1.7). Le passage entre repères se fait aisément grâce aux matrices de transformations homogènes.

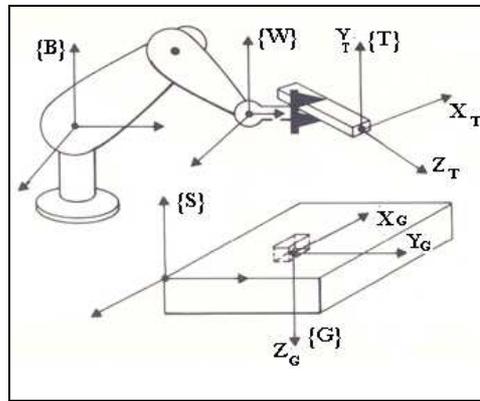


Fig.1.7. Différents repères associés au robot et à son environnement

- La loi de mouvement doit être lisse, i.e., la position et ses dérivées première et deuxième doivent être continues. Cela permet d'éviter les risques d'usure, les vibrations et les forces impulsives pouvant exciter les modes propres des actionneurs.
- La loi de mouvement doit satisfaire les contraintes temporelles de la tâche. Dans certaines applications le temps d'exécution et la vitesse de mouvement sont importants, un exemple est le tapis roulant qui s'immobilise pour une durée fixe permettant la soudure qui doit s'effectuer pendant cet instant.

### 3.3. Description de la trajectoire planaire :

Notre objectif est de générer une trajectoire dans l'espace cartésien, dans laquelle on impose au manipulateur une position et une orientation à suivre avec une vitesse et accélération données.

Le mouvement de la pointe du robot consiste à dessiner la droite  $[A, B]$  suivie du demi-cercle (c) comme l'illustre la figure 1.8. Cette trajectoire se situe dans le plan  $OXY$ , le cercle est repéré

par son centre de coordonnées  $(x_c, y_c)_{/R_0}$  et son rayon  $R$ . On impose une orientation à l'outil terminal de façon à ce qu'il soit oblique au plan OXY d'un angle  $45^\circ$ . En pratique, cela peut correspondre à une soudure que doit exécuter l'effecteur du robot suivant la trajectoire prédéfinie.

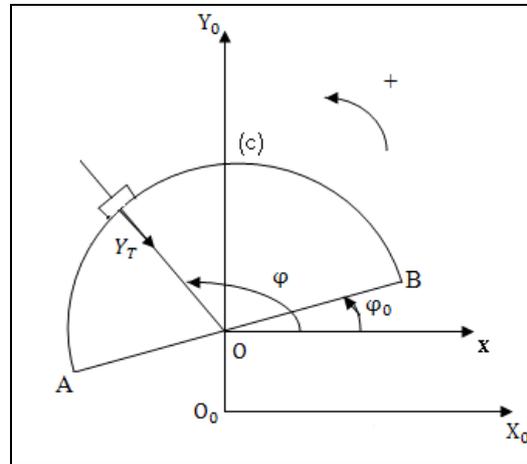


Fig.1.8. Schéma représentatif de la trajectoire désirée

On désire que l'outil terminal parcoure le cercle dans le sens de la flèche avec une vitesse de profil Bang-Bang pour chaque portion de la trajectoire et on prévoit un arrêt au point B pour que le bras puisse entamer le tracé du demi-cercle (c).

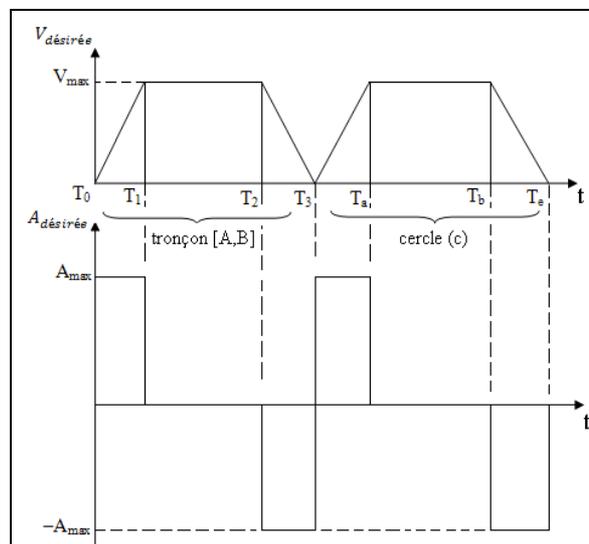


Fig.1.9. Profil de vitesse et d'accélération souhaité pour la trajectoire plane  
**Modélisation de la trajectoire :**

**a. Position, vitesse et accélération :**

La trajectoire représentée dans la figure 1.8 peut être modélisée par le système d'équations ci-dessous. Les équations sont données par rapport au repère absolu  $R_0$ :

➤ Pour la droite AB :

L'abscisse  $x$  est prise comme paramètre.

Position :

$$\begin{cases} y_T = \tan(\varphi_0) * (x - x_c) \\ x_T \in [-R \cos(\varphi_0) + x_c, R \cos(\varphi_0) + x_c] \end{cases}$$

Vitesse :

On a :  $V_{1,désirée} = \sqrt{\dot{x}_T^2 + \dot{y}_T^2} = \dot{x}_T * \cos(\varphi_0)$  donc il est normal d'exprimer la vitesse et l'accélération en fonction de  $x$ .

$$\dot{y}_T = \tan(\varphi_0) \dot{x}_T$$

Accélération :

$$\ddot{y}_T = \tan(\varphi_0) \ddot{x}_T$$

Loi horaire :

$$x_T(t) = \begin{cases} \cos(\varphi_0) * (A_{max} \frac{t^2}{2} - R * \cos(\varphi_0)) & T_0 \leq t \leq T_1 \\ \cos(\varphi_0) * (V_{1,désirée} * (t - \frac{1}{2}T_1) - R \cos(\varphi_0)) & T_1 \leq t \leq T_2 \\ \cos(\varphi_0) * (-\frac{A_{max}}{2}(T_3 - T_2)^2 + R \cos(\varphi_0)) & T_2 \leq t \leq T_3 \end{cases}$$

$$T_1 - T_0 = T_3 - T_2 = \frac{V_{max}}{A_{max}}$$

$$V_{max}^2 \leq 2 R \cos(\varphi_0) * A_{max}$$

➤ Pour le demi-cercle (c) :

L'angle  $\varphi$  est pris comme paramètre :

Position :

$$\begin{cases} x_T = x_c + R \cos(\varphi) \\ y_T = y_c + R \sin(\varphi) \\ \varphi(t) \in [\varphi_0, \varphi_0 + \pi] \end{cases}$$

Vitesse :

On désire que le robot suive dans cette partie une vitesse  $V_{2,désirée}$  de profil trapèze.

$$\left\{ V_{2,désirée} = \sqrt{x_T^2 + y_T^2} = R * \dot{\varphi}_{désirée} \right.$$

Comme la relation entre  $V$  et  $\varphi$  est linéaire, alors, dire que  $V_{désirée}$  suit une loi trapèze revient à le supposer pour  $\varphi$ . Ainsi il est légitime d'exprimer la loi de mouvement en fonction du paramètre  $\varphi$ .

$$\left\{ \begin{array}{l} \dot{x}_T = -R \dot{\varphi} \sin(\varphi) \\ \dot{y}_T = R \dot{\varphi} \cos(\varphi) \\ \dot{\varphi} = A_{2,désirée} / R \\ \varphi(t) \in [\varphi_0, \varphi_0 + \pi] \end{array} \right.$$

Accélération :

$$\left\{ \begin{array}{l} \ddot{x}_T = -R \ddot{\varphi} \sin(\varphi) - R \dot{\varphi}^2 \cos(\varphi) \\ \ddot{y}_T = R \ddot{\varphi} \cos(\varphi) - R \dot{\varphi}^2 \sin(\varphi) \\ \ddot{\varphi} = V_{désirée} / R \end{array} \right.$$

Loi horaire :

Le cercle sera tracé entre l'instant initial  $t_0 = T_0$  et l'instant final  $t_f = T_s$ . La loi horaire de  $\varphi$  sera :

$$\varphi(t) = \left\{ \begin{array}{ll} \ddot{\varphi} \frac{t^2}{2} + \varphi_0 & T_0 \leq t \leq T_a \\ \dot{\varphi}_{max} \left( t - \frac{T_a}{2} \right) + \varphi_0 & T_a \leq t \leq T_b \\ -\frac{\dot{\varphi}_{max}}{2} (T_s - t)^2 + (\varphi_0 + \pi) & T_b \leq t \leq T_s \\ \begin{array}{l} T_a = T_s - T_b = \frac{V_{max}}{A_{max}} \\ \dot{\varphi}_{max}^2 < \ddot{\varphi}_{max} * \pi \end{array} \end{array} \right.$$

**b. Orientation :**

L'orientation de l'organe terminal par rapport au repère absolu suivant cette trajectoire est déduite en appliquant le principe de Frenet-Serret de la géométrie différentielle [3] [10]:

$$R_E^0 = \begin{bmatrix} -\cos(45^\circ) * \cos(\varphi) & -\sin(\varphi) & \sin(45^\circ) \cos(\varphi) \\ -\cos(45^\circ) * \sin(\varphi) & \cos(\varphi) & \sin(45^\circ) * \sin(\varphi) \\ -\sin(45^\circ) & 0 & -\cos(45^\circ) \end{bmatrix}$$

**3.3. Présentation et modélisation de la trajectoire spatiale :**

Le mouvement de la pointe du robot consiste à suivre une courbe, bornée par deux points, sur la surface d'une sphère tout en restant perpendiculaire à cette surface comme indiqué à la figure 1.10. On note que ce mouvement se fait d'une manière continue.

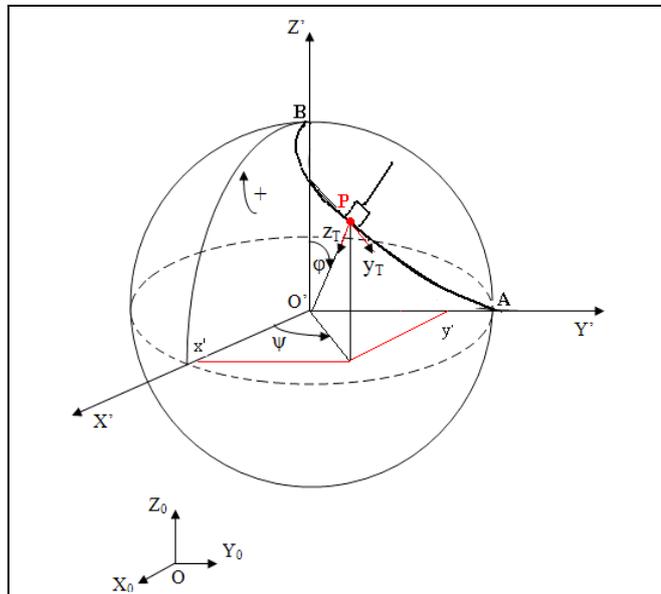


Fig.1.10. Génération d'une trajectoire spatiale

Il est clair que, la description de l'orientation de l'effecteur est plus aisée dans ce cas en utilisant les angles d'Euler  $(\alpha, \beta, \gamma)$ .

La sphère est repérée par son centre  $O'$  de coordonnées  $(x_c, y_c, z_c)$  et de rayon  $R$ . Le repère qui lui est associé  $R'(O', X'Y'Z')$  est parallèle au repère absolu  $R_0(O, X_0Y_0Z_0)$ . En fonction des angles  $\varphi$  et  $\psi$  la trajectoire est facilement décrite.

**Modélisation de la trajectoire désirée :**

**a. Position, vitesse et accélération :**

Position :

Un point se trouvant sur la courbe AB peut être repéré par le système d'équation suivant :

$$\begin{cases} x = R \sin(\varphi) \cos(\psi) + x_c \\ y = R \sin(\varphi) \sin(\psi) + y_c \\ z = R \cos(\varphi) + z_c \\ \varphi \in \left[-\frac{\pi}{2}, -\frac{\pi}{10}\right] \\ \psi \in \left[\frac{\pi}{10}, \frac{\pi}{2}\right] \end{cases}$$

Vitesse :

$$\begin{cases} \dot{x} = R(\cos(\varphi) \dot{\varphi} \cos(\psi) - \sin(\varphi) \sin(\psi) \dot{\psi}) \\ \dot{y} = R(\cos(\varphi) \dot{\varphi} \sin(\psi) + \sin(\varphi) \cos(\psi) \dot{\psi}) \\ \dot{z} = -R \sin(\varphi) \dot{\varphi} \end{cases}$$

Accélération:

$$\begin{cases} \ddot{x} = R(-\sin(\varphi) \dot{\varphi}^2 \cos(\psi) + \cos(\varphi) \ddot{\varphi} \cos(\psi) - \cos(\varphi) \dot{\varphi} \sin(\psi) \dot{\psi} - \cos(\varphi) \dot{\varphi} \sin(\psi) \dot{\psi} - \\ \quad - \sin(\varphi) \cos(\psi) \dot{\psi}^2 - \sin(\varphi) \sin(\psi) \ddot{\psi}) \\ \ddot{y} = R(-\sin(\varphi) \dot{\varphi}^2 \sin(\psi) + \cos(\varphi) \ddot{\varphi} \sin(\psi) + \cos(\varphi) \dot{\varphi} \cos(\psi) \dot{\psi} + \cos(\varphi) \dot{\varphi} \sin(\psi) \dot{\psi} + \\ \quad + \sin(\varphi) \cos(\psi) \dot{\psi}^2 + \sin(\varphi) \sin(\psi) \ddot{\psi}) \\ \ddot{z} = -R(\cos(\varphi) \dot{\varphi}^2 + \sin(\varphi) \ddot{\varphi}) \end{cases}$$

Loi horaire

On suppose que la tâche à exécuter par le robot nécessite des vitesses angulaires  $\dot{\varphi}, \dot{\psi}$  de type trapézoïdal, de l'instant initial  $t_0 = 0$  à l'instant final  $t_f = T_s$ , avec des contraintes maximales sur la vitesse et l'accélération. Ainsi la loi horaire s'exprimant par ces deux variables est donnée par :

$$\varphi(t) = \begin{cases} \ddot{\varphi} \frac{t^2}{2} - \frac{\pi}{2} & T_0 \leq t \leq T_a \\ \dot{\varphi}_{max} \left( t - \frac{T_a}{2} \right) - \frac{\pi}{2} & T_a \leq t \leq T_b \\ -\frac{\dot{\varphi}_{max}}{2} (T_0 - t)^2 - \frac{\pi}{10} & T_b \leq t \leq T_s \\ T_a = T_s - T_b = \frac{\dot{\varphi}_{max}}{\ddot{\varphi}_{max}} \\ \dot{\varphi}_{max}^2 < \ddot{\varphi}_{max} * \frac{2\pi}{5} \end{cases}$$

$$\psi(t) = \begin{cases} \ddot{\psi} \frac{t^2}{2} + \frac{\pi}{10} & T_0 \leq t \leq T_a \\ \dot{\psi}_{max} \left( t - \frac{T_a}{2} \right) + \frac{\pi}{10} & T_a \leq t \leq T_b \\ -\frac{\dot{\psi}_{max}}{2} (T_0 - t)^2 + \frac{\pi}{2} & T_b \leq t \leq T_s \\ T_a = T_s - T_b = \frac{\dot{\psi}_{max}}{\ddot{\psi}_{max}} \\ \dot{\psi}_{max}^2 < \ddot{\psi}_{max} * \frac{2\pi}{5} \end{cases}$$

### b. Angles d'Euler correspondant à l'orientation

La pointe du bras doit être perpendiculaire à la surface de la sphère cela revient à dire que  $Z_T$  est perpendiculaire à la courbe AB comme indiqué sur la figure 1.10. Le point  $P$  se trouve sur le plan de coupe repéré par l'angle  $\psi$  (figure 1.10), alors, l'angle  $\alpha = \psi$ . Pour qu'on ait  $Z_T$  entrant, la condition sur l'angle  $\beta$ , est alors :  $\beta = -(\pi - \varphi)$ . De plus, l'angle  $\gamma = 0$ . D'où l'orientation avec les angles d'Euler selon la représentation  $ZYZ$  :

$$\begin{cases} \alpha = \psi \\ \beta = \varphi - \pi \\ \gamma = 0 \end{cases}$$

## 4. Synthèse de la loi de commande par couple calculé :

La commande des robots-manipulateurs a fait l'objet de nombreux travaux. Les techniques de commande développées font appel à des méthodes de synthèse, établies en fonction des critères d'implémentation, de complexité de la tâche à effectuer (suivi de surface, transport d'objets), du comportement dynamique ou statique désiré (rapidité et précision) et d'insensibilité aux variations des paramètres.

### 4.1. Hypothèses :

Les hypothèses que nous allons admettre sont valables pour toutes les commandes que nous allons développer aussi bien dans ce chapitre que dans ceux qui vont suivre :

- Bien que les robots soient en général maintenant pilotés par des calculateurs numériques, donc avec une certaines périodes d'échantillonnage, nous supposons que cette période est suffisamment faible pour assimiler le système hybride (discret et continu) ainsi constitué à un système continu, et nous présenterons et utiliserons donc des équations à temps continu.
- Le robot aura comme variable d'entrée le vecteur  $\tau$  des couples appliqués aux articulations, et comme variable de sortie le vecteur  $q$  des positions articulaires. C'est d'ailleurs l'approche adoptée dans la majeure partie des ouvrages.
- Les actionneurs du robot sont piloté par des variateurs qui s'interpose entre les convertisseurs analogiques/numériques et les moteurs et ainsi permettent de l'équation dynamique (1.5) de décrire le système en entier avec ses deux aspects mécanique et électrique. La référence [5] expose en détail cette hypothèse.
- On suppose que les positions et vitesses articulaires sont bornées, mesurables et que les mesures ne sont pas bruitées.

Dans ce chapitre, la commande utilisée pour réaliser la poursuite en position et orientation, est la commande par découplage non linéaire (DNL) dans l'espace opérationnel avec correction du type PD (proportionnel dérivé) générée dans l'espace articulaire. Le passage entre l'espace opérationnel et l'espace articulaire est assuré par le modèle géométrique inverse, le modèle cinématique inverse du premier ordre et du second ordre. On donne ci-dessous le schéma représentatif de la structure de commande :

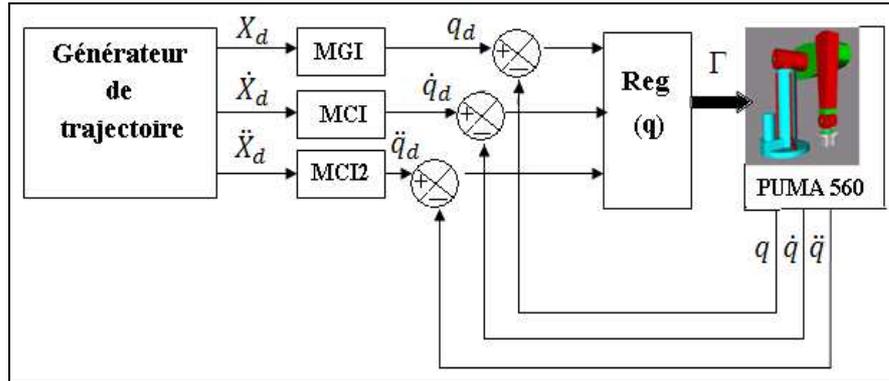


Fig.1.11. Structure de commande avec correction dans l'espace articulaire

L'équation dynamique (1.5) du système peut se mettre sous la forme :

$$\mathbf{u} = \mathbf{A}(q)\ddot{\mathbf{q}} + \mathbf{H}(q, \dot{\mathbf{q}}) \quad (1.6)$$

Où :  $\mathbf{u}$  est le vecteur de commande représentant le vecteur  $\mathbf{\Gamma}$  des couples articulaires, et

$$\mathbf{H}(q, \dot{\mathbf{q}}) = \mathbf{B}(q)[\dot{\mathbf{q}} \dot{\mathbf{q}}] + \mathbf{C}(q)[\dot{\mathbf{q}}^2] + \mathbf{g}(q).$$

Les forces de frottement sont considérées comme des perturbations.

Dans le cas général, le problème de linéarisation par retour d'état d'un système non linéaire n'est pas facile à résoudre. Cependant, dans le cas des robots manipulateurs rigides ce problème est simplifié par le fait que le nombre d'actionneurs est égal au nombre de variables articulaires et que le modèle dont on dispose est un modèle inverse qui exprime l'entrée  $\mathbf{\Gamma}$  du système en fonction du vecteur d'état  $(\mathbf{q}, \dot{\mathbf{q}})$  et de  $\ddot{\mathbf{q}}$ . Ces propriétés font que les équations du robot définissent un système *plat* dont les sorties *plates* sont les variables articulaires  $\mathbf{q}$ . Le choix du vecteur de commande est :

$$\mathbf{u} = \mathbf{A}(q)\mathbf{v} + \mathbf{H}(q, \dot{\mathbf{q}}) \quad (1.7)$$

Où  $\mathbf{v}$  est la nouvelle entrée du système, la combinaison des deux équations (1.6) et (1.7)

donne :

$$\mathbf{v} = \ddot{\mathbf{q}} \quad (1.8)$$

Si on choisit :

$$\underline{\mathbf{v}} = \ddot{\mathbf{q}}^d + K_v \dot{\mathbf{e}} + K_p \mathbf{e} \quad (1.9)$$

Avec :  $\mathbf{e} = \mathbf{q}^d - \mathbf{q}$  l'erreur de poursuite et  $\dot{\mathbf{e}}$  sa dérivée par rapport au temps.

Ainsi l'équation régissant le fonctionnement du système en boucle fermée sera donnée par :

$$\ddot{e} + K_v \dot{e} + K_p e = 0 \quad (1.10)$$

Avec :

$$K_v = \text{diag}(K_{v1}, K_{v2}, \dots, K_{vn})$$

$$K_p = \text{diag}(K_{p1}, K_{p2}, \dots, K_{pn})$$

Nous avons ainsi obtenu un système, qui plutôt que d'être représenté par une équation différentielle reliant son entrée et sa sortie, peut être représenté par une équation différentielle de l'erreur sur la variable à commander. Cela nous conduit à un système en boucle fermée complètement découplé, où chaque réponse articulaire sera celle d'un système linéaire du second ordre. Nous pouvons ainsi, en choisissant les gains  $K_v$  et  $K_p$  faire évoluer l'erreur de poursuite vers zéro avec la dynamique qu'on souhaite. Par exemple, le choix de  $K_v = \sqrt{2K_p}$  assure un régime apériodique optimal ( $\xi = 0.7$ ).

La méthode du couple calculé est ainsi très attirante du point de vue commande car le fort couplage dynamique non linéaire a été éliminé, néanmoins, la question importante qui se pose est que se passe-t-il si maintenant nous n'avons pas une connaissance parfaite des paramètres dynamiques  $A, B, C, G$  et  $F$  mais une connaissance approchée  $\hat{A}, \hat{B}, \hat{C}, \hat{G}$  et  $\hat{F}$  ?

#### 4.4. Etude de la robustesse de la commande par découplage non linéaire :

La loi de commande par couple calculé nécessite la connaissance parfaite des paramètres dynamique chose difficile à satisfaire en pratique car non seulement l'identification de ces paramètres nécessite un matériel adéquat et précis mais dans tout système physique il existe un certain degré d'incertitude conduisant à la variation des paramètres en fonction du temps.

Ceci est particulièrement vrai dans le cas d'un système assez compliqué comme le robot qui de plus est chargé de manipuler des objets de masses inconnues et/ou variables.

Soient :  $\hat{A}, \hat{B}, \hat{C}, \hat{G}$  et  $\hat{F}$  les valeurs approchées de  $A, B, C, G$  et  $F$ , l'équation (1.7) donnant la commande devient :

$$\bar{U} = \hat{A}(q) \underline{v} + \hat{H}(q, \dot{q}) \quad (1.11)$$

En remplaçant par (1.11) dans (1.5) l'équation du système en boucle fermée sera donnée par :

$$\ddot{e} + K_v \dot{e} + K_p e = \hat{A}^{-1} \{ (A - \hat{A}) \ddot{q} + (B - \hat{B}) [\dot{q} \ \dot{q}] + (C - \hat{C}) [\dot{q}^2] + G - \hat{G} + F - \hat{F} \} \quad (1.12)$$

Qu'on peut écrire sous la forme:

$$\ddot{\mathbf{e}} + K_v \dot{\mathbf{e}} + K_p \mathbf{e} = \boldsymbol{\eta} \quad (1.13)$$

D'après l'équation (1.13),  $\boldsymbol{\eta}$  est une fonction non linéaire de  $\mathbf{e}$  et donc si  $\boldsymbol{\eta}$  ne s'annule pas le système reste non linéaire couplé et même si on augmente les gains  $K_p$  et  $K_v$  on ne peut amener assurer la stabilité du système car cela conduit à l'accroissement de  $\boldsymbol{\eta}$ .

Aussi l'équation (1.13), montre qu'il existe une erreur en régime permanent  $\mathbf{e}_p$  entraînant un comportement non idéal du système. Elle est donnée par :

$$K_p \mathbf{e}_p = \hat{A}^{-1} \{(\mathbf{G} - \hat{\mathbf{G}}) + (\mathbf{F} - \hat{\mathbf{F}})\} \quad (1.14)$$

Cette erreur pouvant être réduite voire annulée, en ajoutant un terme intégrale dans la loi de commande représentée par l'équation (1.7). Quant à l'erreur en régime transitoire elle peut être compensée par l'utilisation de commandes robustes ou adaptatives.

#### 4.5. Simulations et interprétations :

Le logiciel utilisé pour la simulation est SIMULINK<sup>®</sup> vu les performances dont il dispose, programmation en architecture modulaire, une bibliothèque assez riche et une vitesse de simulation rapide relativement au MATLAB<sup>®</sup> [12]. Pour la résolution des équations différentielle, la méthode utilisée est l'ODE 45.

On suppose qu'initialement le robot est dans la configuration qui permet à son organe terminal de se positionner au point initial de la trajectoire et qu'on fin de tâche, il y revient. La trajectoire qui lui permet de revenir au point initial à partir du point d'arrêt est supposée générée d'une manière hors ligne qui sort du cadre de notre travail.

Dans un premier temps, on génère dans le cas idéal (sans variation des paramètres), la trajectoire dans le plan décrit dans la figure 1.7, puis la trajectoire spatiale figure 1.10. Par la suite, on introduit une variation des paramètres de 50% pour la trajectoire planaire à titre indicatif afin de mettre en évidence la défaillance de ce type de commande lorsque le modèle est mal connu.

### 4.5.1. Cas idéal : sans erreurs de modélisation :

De peur d'exciter les dynamiques non modélisées généralement de hautes fréquences et afin de s'éloigner de la fréquence de résonance inconnue du système, les paramètres de synthèse de la loi de commande sont choisis comme suit :

$$\begin{cases} \omega_0 = 70 \text{ rd/s} \\ \xi = 0.7 \end{cases} \Rightarrow \begin{cases} K_p = 4900 \\ K_v = 99 \end{cases}$$

#### *Simulation 1 : (trajectoire planaire)*

Les simulations ci-dessous concernent les résultats de l'application de la commande par couple calculé au système supposé de dynamique parfaitement connue.

Le rayon du cercle est de 0.45 m et le temps de simulation est de  $T_e = 14\text{s}$  et la vitesse est de 0.5m/s, une accélération de  $1\text{m/s}^2$ .

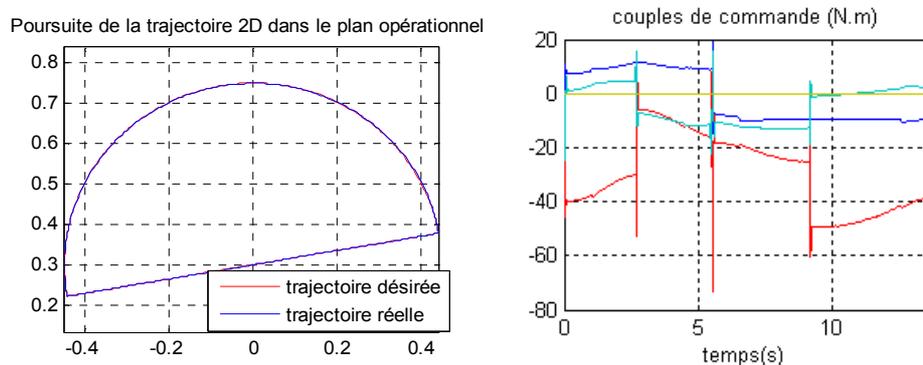


Fig.1.12. Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande par découplage non linéaire en absence de variation des paramètres.

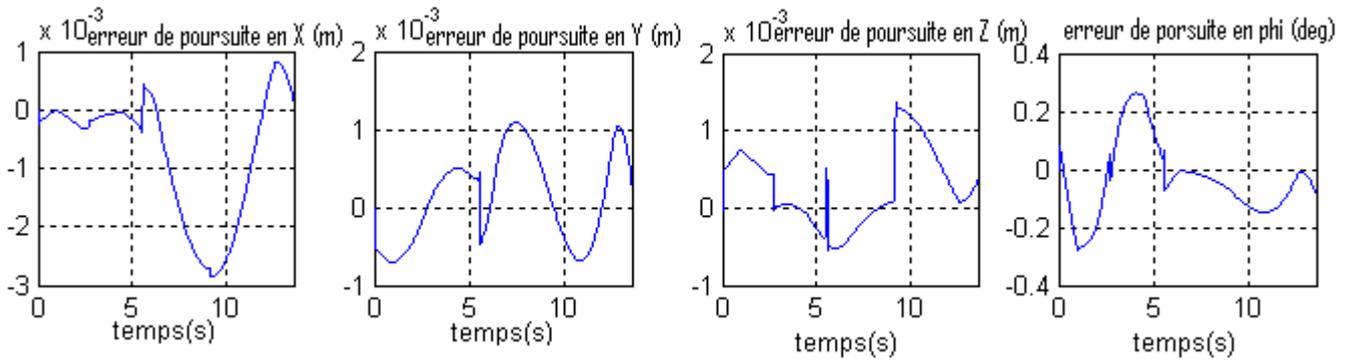


Fig.1.15. Signaux d’erreurs en position et orientation pour la trajectoire planaire avec la commande par découplage non linéaire en absence de variation des paramètres

On constate une parfaite poursuite de la trajectoire désirée (les trajectoires désirée et réelle sont superposées) ce qui est confirmé par les erreurs opérationnelles qui sont de l’ordre de 0.8% (dans la marge tolérée de 4mm) au maximum. Les diagrammes des signaux des erreurs de poursuites cartésiennes montrent qu’elles sont plus importantes dans le deuxième tronçon (arc cercle (c)). Ceci est dû à la nature de la trajectoire, il est évident qu’il est plus facile au robot de tracer une ligne droite plutôt qu’un arc de cercle incliné de 10°. Aussi on remarque que le maintien de l’orientation désirée est parfait vu que l’erreur sur l’angle  $\varphi$  ne dépasse pas 0.5°. La figure des couples articulaires, montre qu’ils sont admissibles par les actionneurs.

Simulation 2 : trajectoire spatiale :

Pour la trajectoire spatiale pour éviter les positions singulières on choisit un rayon est de 0.6m, le temps de simulation est de  $T_e=25s$ . La vitesse est de 0.1m/s. Les résultats sont les suivants :

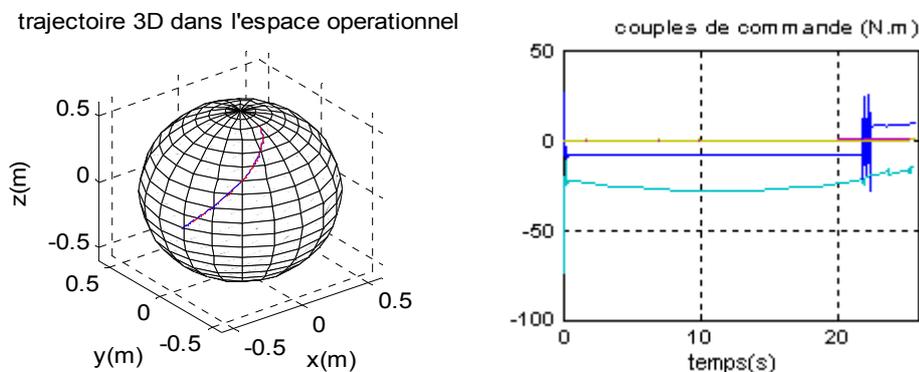


Fig.1.13. Réponse et signaux de commande pour la trajectoire 3D avec la commande par découplage non linéaire en absence de variation des paramètres

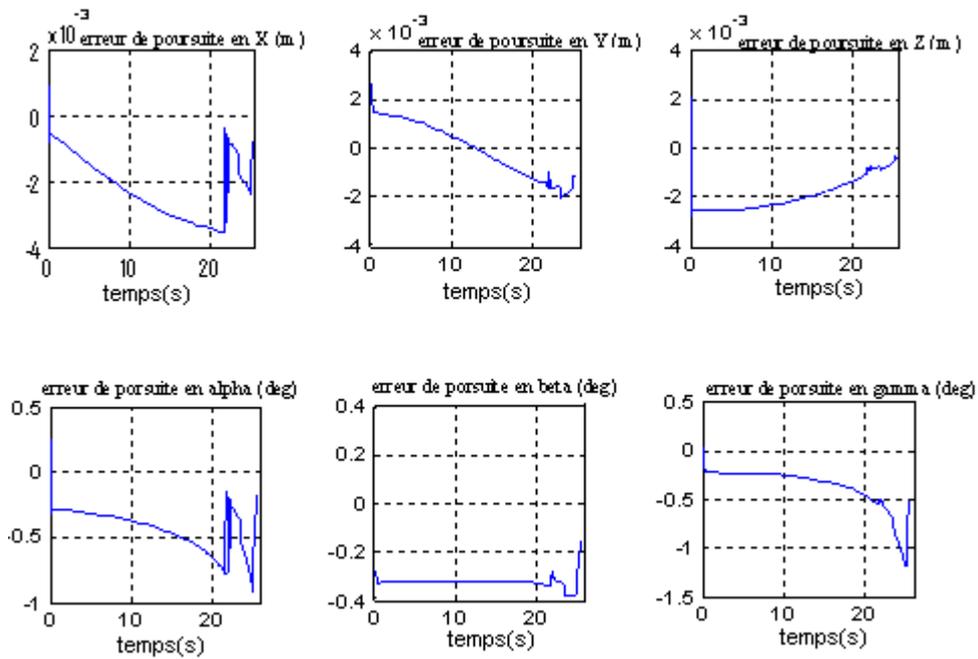


Fig.1.14. Signaux des erreurs de position et d'orientation pour la trajectoire 3D avec la commande par découplage non linéaire en absence de variation des paramètres

On constate une bonne poursuite de la trajectoire opérationnelle et des erreurs négligeables, de l'ordre de 0.4% au maximum pour la position et de 0.06% pour l'orientation.

Les couples de commande subissent une forte variation au démarrage, afin de permettre au robot d'acquiescer la configuration souhaité, puis ils redeviennent lisses. Cependant, ils restent tout au long de l'exécution de la tâche dans les limites tolérées.

#### 4.5.2. En présence d'une erreur de modélisation de 50% :

Pour étudier la robustesse de la commande, on fait varier les paramètres du système (accroissement de 50%) tout en laissant ceux de la commande fixes. Les résultats obtenus sont les suivants :

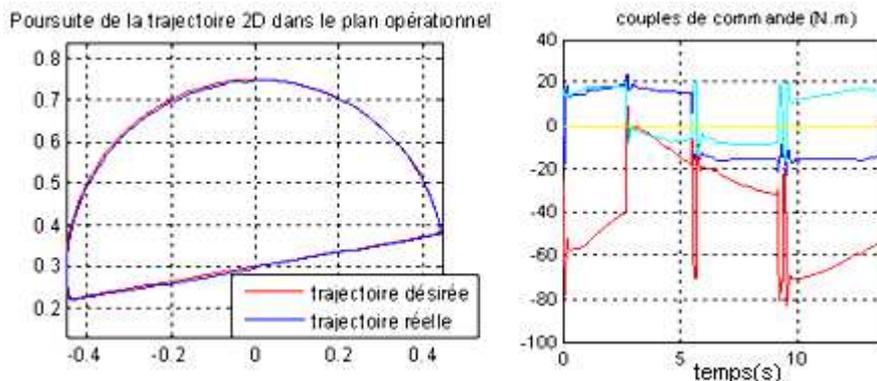


Fig.1.15. Réponse du système et signaux de commande pour la trajectoire planaire avec la commande par couple calculé pour une variation des paramètres de 50%

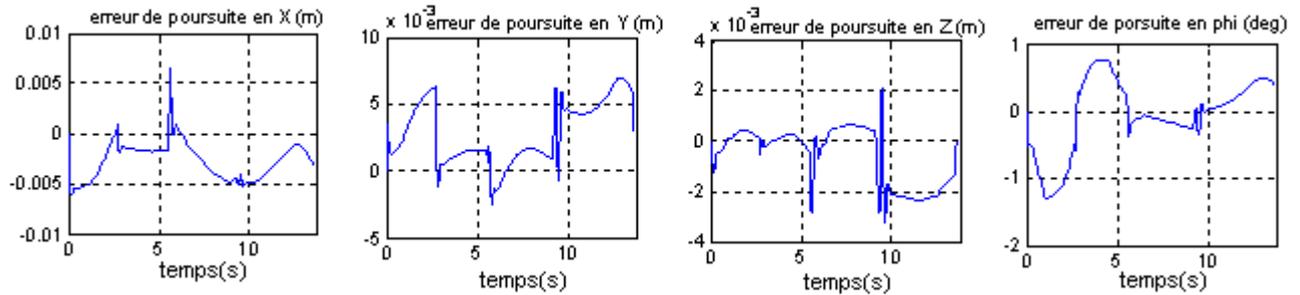


Fig.1.16. Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la commande par découplage non linéaire pour une variation des paramètres de 50%

D'après les figure 1.14 et 1.15, on constate qu'il y a une erreur de poursuite importante : les deux courbes (réelle et désirée ne sont plus superposées). Les erreurs en position dépassent les limites tolérées (4mm), elle es de l'ordre de 2.2%. Les couples de commandes sont beaucoup plus importants vu qu'ils atteignent les 80 N.m.

Cela prouve la non robustesse de la commande vis-à-vis les erreurs de modélisation, or dans la majorité des cas, le modèle du robot est mal connu, car cela dépend de la manière avec laquelle il est identifié [13].

### Conclusion :

Dans ce chapitre a été traité le problème de la commande par couple calculé du bras manipulateur PUMA 560.

Après la modélisation du bras, on est passé à la génération des mouvements dans l'espace opérationnel libre (pas de contact avec une quelconque surface), où le robot a été amené à suivre un arc de cercle fermé dans le plan horizontal, puis à tracer une courbe sur la surface d'une sphère, tout en respectant les contraintes cinématiques imposées.

En absence des erreurs de modélisation la commande par couple calculé est parfaite (l'erreur en position ne dépasse pas les 4mm et  $0.5^\circ$  pour l'orientation), mais le test de robustesse (variation des paramètres de 50%) a prouvé la limitation de ce type de commande, cela nous amène à envisager des méthodes plus robustes et qui seront le thème des chapitres suivants.

# 2

## SYNTHESE DE LA COMMANDE ADAPTATIVE

---

### 2.1. Introduction :

Le chapitre précédent nous a prouvé que lorsque les paramètres intrinsèques au robot sont mal connus ou bien ils ont évolué avec le temps, la commande par couple calculé devient insuffisante et la nécessité de synthèse d'une loi de commande robuste s'avère incontournable.

En effet, l'usure, la variation des réglages des mécanismes entraînent des modifications du comportement dynamique du robot, chose que la commande par découplage non linéaire ne prévoit pas vu que son hypothèse de base est la parfaite connaissance du modèle du robot.

Les incertitudes sont classées en deux catégories: incertitudes paramétriques et dynamiques négligées. Le premier type entre directement en jeu dans le modèle tandis que le second type porte sur l'ordre sous-estimé du système. La commande robuste est la première technique de commande de l'automatique traitant l'inévitable problème des incertitudes. Dans la synthèse de la loi de commande sont pris en compte un modèle nominal du procédé à contrôler mais aussi les incertitudes paramétriques liées au modèle. La structure du contrôleur robuste est finalement composée d'une partie "*nominale*"(ex. retour d'état...) mais aussi de termes additionnels permettant de compenser au mieux les incertitudes liées au modèle. Parmi les différentes techniques, la commande par modes de glissement est une technique particulièrement intéressante. Cependant cette commande souffre de quelques inconvénients : les modes glissant causent le broutement (le chattering, ou encore la réticence) inhérent à la fonction discontinue (i.e. la fonction Sgn). Notons que le chattering peut exciter les dynamiques haute-fréquence négligées menant parfois à l'instabilité. Des méthodes permettant de réduire ce phénomène ont été développées dans [11]. Aussi, les bornes sur les incertitudes doivent aussi être connues [11] [14].

Une alternative à la commande robuste est la commande adaptative. La commande adaptative offre l'avantage que les bornes des incertitudes ne sont pas exigées d'être connues, puisque en fait

ces incertitudes sont annulées en ligne d'une manière adaptative. Dans un schéma de commande adaptative, les paramètres du contrôleur (schéma direct) ou des fonctions non linéaires (schéma indirect) sont adaptés en ligne en utilisant les signaux qui sont disponibles dans le système.

Cependant, les méthodes de conception de la commande adaptative standard [11] sont limitées aux systèmes non linéaires qui peuvent être linéairement paramétrées (i.e. les non-linéarités du système ou la commande peuvent être exprimées sous forme de produit d'une fonction non linéaire connue par un vecteur de paramètres inconnus), le cas de notre robot, vu que, le modèle est linéaire par rapport aux paramètres inertiels.

On peut à l'heure actuelle considérer que la commande adaptative des robots rigides a atteint une bonne maturité comme en témoigne les nombreuses méthodes publiées sur le sujet [1], [11],[15],[16],[17]. Il s'en dégage cinq stratégies significatives des progrès réalisés dans ce domaine [1]:

1. Simplification du modèle dynamique : Dubowsky 79, Takegaki 81 ;
2. Application des techniques adaptatives développées pour les systèmes linéaires au cas des robots : Horowitz 80, Hsia 86, Nicosia 84
3. Proposition d'une commande adaptative non linéaire asymptotiquement stable [18].
4. Utilisation des propriétés de la passivité pour développer une loi de commande adaptative non linéaire asymptotiquement stable : Slotine 87, Sadegh 87, Landau 88, Kelley 88.
5. Utilisation pour l'adaptation d'un modèle qui n'est pas fonction des accélérations articulaires tel que le modèle dynamique filtré : Middleton 88, Li 89, ou le modèle énergétique EL Serafi 91.

Les lois de commande proposées dans les deux premières stratégies ne sont valables que pour des mouvements lents et ne prennent pas en compte le modèle complet du robot [1]. La loi de commande adaptative non linéaire de Craig dépend de deux hypothèses restrictives : la mesure des articulations articulaires et l'existence d'une borne de la matrice d'inertie estimée. Les lois de commandes de la quatrième et cinquième stratégie n'ont pas besoin de mesures des accélérations articulaires et sont, au moins de point de vue théorique, les plus intéressantes [1].

Dans la suite, nous présentons la loi de commande par découplage linéaire et la loi adaptative passive. Des simulations vont mettre en évidence l'intérêt de chaque commande.

## 2.2. Commande adaptative par découplage non linéaire [1]:

La première version d'une commande dynamique adaptative a été formulée par Craig en 1986. La loi de commande a la même structure que celle de la commande par découplage non linéaire de l'équation (1.10) et s'écrit sous la forme suivante :

$$U(t) = A(q, \hat{\lambda}) v(t) + H(q, \dot{q}, \hat{\lambda}) \quad (2.1)$$

Où  $\hat{\lambda}$  est le vecteur des paramètres dynamiques estimés et où :

$$\begin{cases} v(t) = \ddot{q}^d + K_v \dot{e} + K_p e \\ e = q^d - q \end{cases} \quad (2.2)$$

Dans la suite, cette loi de commande sera notée :

$$U(t) = \tilde{A}(q) v(t) + \tilde{H}(q, \dot{q}) \quad (2.3)$$

Grâce à la propriété (5) du modèle dynamique du bras de robot, l'équation (1.5) peut se mettre sous la forme :

$$\Gamma = \Phi(q, \dot{q}, \ddot{q}) \chi \quad (2.4)$$

$\chi$  étant le vecteur des paramètres inconnus du modèle.

Partant des équations (2.4), (2.2) et (2.1), l'équation dynamique d'erreur du système en boucle fermée peut s'écrire :

$$\ddot{e} + K_v \dot{e} + K_p e = \tilde{A}^{-1}(q) [\Phi(q, \dot{q}, \ddot{q}) \chi - \Phi(q, \dot{q}, \ddot{q}) \hat{\lambda}] = \tilde{A}^{-1}(q) \Phi(q, \dot{q}, \ddot{q}) \tilde{\lambda} \quad (2.5)$$

Où :  $\tilde{\lambda} = \hat{\lambda} - \chi$  est l'erreur d'estimation.

Pour calculer la loi d'adaptation, on réécrit l'équation (2.5) sous la forme d'état suivante :

$$\dot{x} = a x + b \tilde{A}^{-1}(q) \Phi(q, \dot{q}, \ddot{q}) \tilde{\lambda} \quad (2.6)$$

$$\text{Avec : } x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix}, a = \begin{bmatrix} 0 & I \\ -K_p & -K_v \end{bmatrix}, b = \begin{bmatrix} 0 \\ \Gamma \end{bmatrix} \quad (2.7)$$

Où : 0 et  $I$  sont respectivement la matrice nulle et la matrice identité de dimension  $(n \times n)$ .

$$K_p = \text{diag}(k_{p,1}, \dots, k_{p,n}) \quad K_v = \text{diag}(k_{v,1}, \dots, k_{v,n})$$

On choisit la fonction candidate de Lyapunov suivante :

$$V = x^T P x + \tilde{\chi}^T \Lambda \tilde{\chi} \quad (2.8)$$

Où :  $\Lambda = \text{Diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$  où les  $\lambda_j$  sont des constantes positives et  $m$  est le nombre de paramètres dynamique à adapter.

La matrice  $P$  est l'unique solution de l'équation de Lyapunov suivante :

$$a^T P + P a + Q = 0 \quad (2.9)$$

La différentiation par rapport au temps de la fonction candidate de Lyapunov  $V$  aboutit à :

$$\dot{V} = -x^T F x + 2 \tilde{\chi}^T [\dot{\Phi}^T(q, \dot{q}, \ddot{q}) \hat{A}^{-1}(q) b^T P x - \Lambda \dot{\tilde{\chi}}] \quad (2.10)$$

Avec :  $F = Q + a^T P$  matrice définie positive

En utilisant la loi d'adaptation donnée par :

$$\dot{\tilde{\chi}} = \Lambda^{-1} \dot{\Phi}^T(q, \dot{q}, \ddot{q}) \hat{A}^{-1}(q) b^T P x = \dot{\tilde{\chi}} \quad (2.11)$$

L'expression de  $\dot{V}$  devient :

$$\dot{V} = -x^T F x \leq 0 \quad (2.12)$$

Par conséquent, le lemme de Barbalat permet de conclure que la fonction  $\dot{V}$  tend uniformément vers zéro et donc  $x \rightarrow 0$ .

Comme le vecteur  $x$  a pour éléments  $e$  et  $\dot{e}$ , alors l'erreur et sa dérivée tendent aussi uniformément vers zéro. L'algorithme de commande adaptative donné par les relations (2.3) et (2.11) est donc globalement asymptotiquement stable.

Cette méthode présente l'avantage de faire une adaptation asymptotiquement stable non pas sur les paramètres du système mais sur ceux du modèle. Néanmoins, elle comporte 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 rester bornée. Craig *et al.* ont suggéré de projeter les paramètres estimés dans une région connue de l'espace des paramètres contenant les paramètres réels et dans laquelle l'inverse de la matrice d'inertie existe. Cependant, la connaissance et le calcul de telle région ne sont pas évidents.

Spong [2] et Ortega ont proposé une nouvelle version de la loi de commande adaptative par découplage non linéaire dans laquelle la condition sur la matrice  $\hat{A}^{-1}(q)$  est levée, mais les accélérations articulaires restent indispensables.

La figure ci-dessous illustre le principe de la commande par découplage non linéaire adaptative :

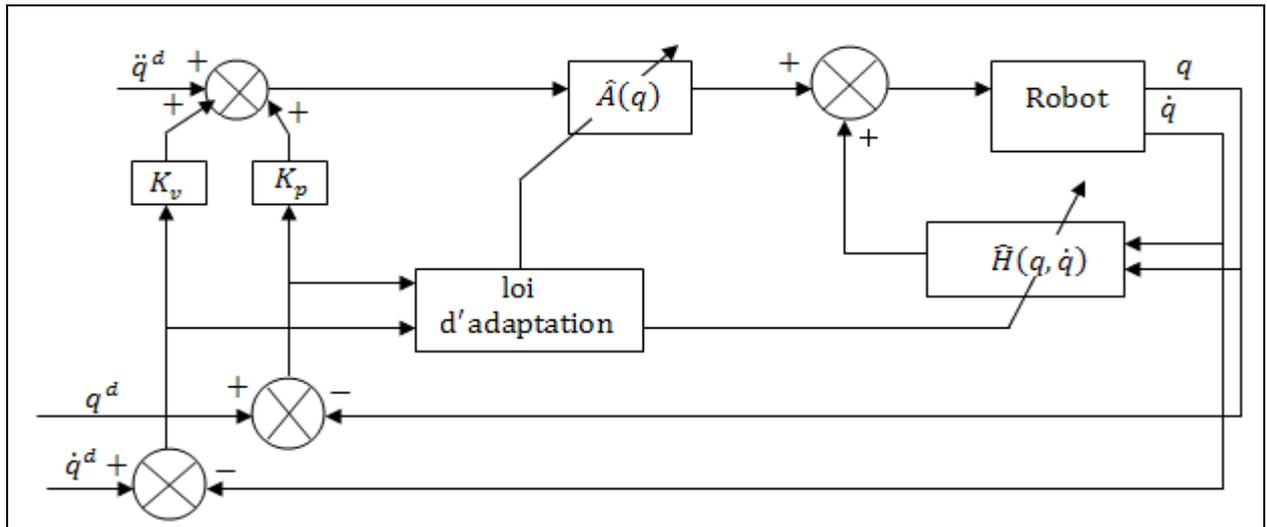


Fig.2.1. Commande par découplage non linéaire adaptative (d'après Craig)

### 2.3. Commande adaptative passive :

Afin de développer un algorithme adaptatif fondé sur le modèle dynamique complet, Slotine [11] et Lie ont exploité la propriété d'antisymétrie de la matrice  $(\dot{A} - 2c)$ . Tel que :

$$\left\{ \begin{array}{l} c = B(q) * Q_1 + C(q) * Q_2 \\ Q_1 = \begin{bmatrix} \dot{q}_2 & \dot{q}_3 & \dot{q}_4 & \dot{q}_5 & \dot{q}_6 & 0_{1 \times 10} \\ 0_{1 \times 5} & \dot{q}_3 & \dot{q}_4 & \dot{q}_5 & \dot{q}_6 & 0_{1 \times 6} \\ 0_{1 \times 9} & & \dot{q}_4 & \dot{q}_5 & \dot{q}_6 & 0_{1 \times 3} \\ 0_{1 \times 12} & & & \dot{q}_5 & \dot{q}_6 & 0_{1 \times 1} \\ 0_{1 \times 14} & & & & & \dot{q}_6 \end{bmatrix}^T \\ Q_2 = \text{Diag}(\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6) \end{array} \right. \quad (2.13)$$

Cette propriété est liée à la passivité du robot, d'où le nom de méthode de commande adaptative passive.

#### 2.3.1. Algorithme de base [11]:

Ayant une trajectoire désirée  $q^d$  (on admet que les consignes de position, vitesse et accélération sont bornées), et en supposant qu'on n'a aucune connaissance a priori des paramètres du système, le problème de la conception d'un régulateur adaptatif direct est la synthèse en même temps d'une loi de commande et d'une loi d'estimation des paramètres, telles que les sorties réelles  $q(t)$  du système parviennent à poursuivre celles désirées. Pour cette fin, on définit la surface de glissement

$\mathbf{s} = \left( \lambda + \frac{d}{dt} \right) \mathbf{e}$ , et le paramètre  $\dot{\mathbf{q}}_r = \dot{\mathbf{q}} + \Lambda \mathbf{e}$ .

La fonction de Lyapunov candidate choisit est la suivante :

$$V = \frac{1}{2} (\mathbf{s}^T A \mathbf{s} + \tilde{\chi}^T \eta^{-1} \tilde{\chi}) \quad (2.14)$$

Avec :  $A, \chi$  définis respectivement dans (1.5), (2.4).

$\eta = \text{diag}(\eta_1, \dots, \eta_n)$  matrice carrée symétrique de dimension  $(n \times n)$  définie positive.

$\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$  matrice symétrique définie positive.

La différentiation par rapport au temps de  $V$  en prenant en considération la propriété de passivité, et l'équation (2.13) donne :

$$\dot{V} = \mathbf{s}^T (\Gamma - A \ddot{\mathbf{q}}_r - \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_r - \mathbf{g}(\mathbf{q})) + \dot{\chi}^T \eta^{-1} \tilde{\chi} \quad (2.15)$$

En exploitant la propriété (5) du modèle dynamique du robot, les termes  $A, c$  et  $g$  dépendent linéairement de  $\chi$  et donc on peut définir une matrice non linéaire  $Y(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)$  telle que :

$$A(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}}_r + \mathbf{g}(\mathbf{q}) = Y(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r) \chi \quad (2.16)$$

En choisissant la loi de commande suivante :

$$\Gamma = \mathbf{U} = Y \tilde{\chi} - K_D \mathbf{s} \quad (2.17)$$

Cette loi de commande contient en plus d'un simple P.D, un terme de correction par anticipation  $Y \tilde{\chi}$ .

$$\text{Ainsi : } \dot{V} = \mathbf{s}^T Y \tilde{\chi} - \mathbf{s}^T K_D \mathbf{s} + \dot{\chi}^T \eta^{-1} \tilde{\chi} \quad (2.18)$$

Par conséquent, la loi d'adaptation paramétrique sera :

$$\dot{\tilde{\chi}} = -\eta Y^T \mathbf{s} \quad (2.19)$$

Cela conduit à :

$$\dot{V} = -\mathbf{s}^T K_D \mathbf{s} \leq 0 \quad (2.20)$$

En utilisant le lemme de Barbalat, il est clair que l'erreur converge vers la surface  $\mathbf{s} = \mathbf{0}$  et donc  $\mathbf{e}$  et  $\dot{\mathbf{e}}$  tendent vers zéro. Par conséquent, on a assuré simultanément l'asymptotique globale stabilité du système et la convergence des erreurs de poursuite.

Notons que, cette approche n'implique pas nécessairement l'estimation des paramètres du modèle, mais génère des valeurs qui permettent la réalisation de l'objectif de commande. Aussi, l'erreur de poursuite ne tend pas asymptotiquement vers zéro mais en temps fini [11] qui dépend des paramètres  $A, \eta$  et  $K_D$  et dont le choix est contraint par les la présence des dynamiques non modélisées hautes fréquences et par les bruits de mesure.

La loi d'adaptation paramétrique donnée dans (2.19) dépend uniquement des signaux d'erreurs de poursuite, par ailleurs, ces signaux ne sont pas la seule source d'information sur les paramètres du système. Ainsi une stratégie pour l'estimation des paramètres combinant les erreurs de prédiction et les erreurs de poursuite a été développée dans [11] et permet d'augmenter les performances du contrôleur adaptatif.

La figure ci-dessous illustre le principe de la commande adaptative passive :

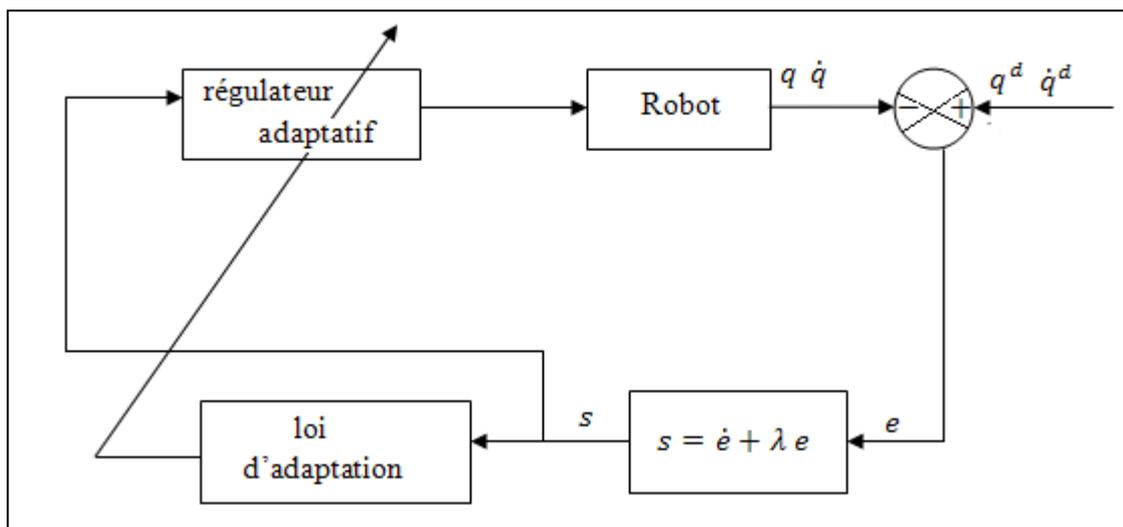


Fig.2.2. Schéma du régulateur adaptatif passif (d'après Slotine)

## 2.4. Simulations et interprétations :

On implémente la loi de commande par découplage non linéaire adaptative et la commande adaptative passive sous SIMULINK<sup>®</sup>. Pour la résolution des équations différentielle, la méthode utilisée est l'ODE 45.

On suppose qu'initialement le robot est dans la configuration qui permet à son organe terminal de se positionner au point initial de la trajectoire et qu'on fin de tâche, il y revient. La trajectoire qui lui permet de revenir au point initial à partir du point d'arrêt est supposée générée d'une manière off line.

Dans un premier temps, pour chaque commande on génère dans le cas idéal (sans variation des paramètres), la trajectoire planaire décrite dans la figure 1.10. Ensuite, on introduit une variation des paramètres de 50%. Dans la synthèse de la loi de commande, les forces de frottement ont été négligés, en effet, ils sont considérés comme des perturbations que doit rejeter le régulateur, cela constitue un test supplémentaire de la robustesse de la commande. Finalement, une comparaison entre les deux structures de commande sera établie.

### 2.4.1. Commande par découplage non linéaire adaptative:

Après plusieurs simulations notre choix des paramètres du régulateur s'est fixé sur les suivants :

- On a supposé qu'a priori qu'aucune connaissance des paramètres du modèle n'est disponible ce qui nous amène à envisager l'estimation de 30 paramètres : 25 constantes d'inertie et 5 constantes gravitationnelles, ainsi :  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_{30})$  et on prend :

$$\lambda_1 = \lambda_2 = \dots = \lambda_{30} = 1000$$

- Pour le choix de la dynamique de l'erreur on opte pour le régime apériodique avec un amortissement de 0.7. Par conséquent  $k_{v,i} = \sqrt{2k_{p,i}}$  et les gains choisis sont les suivants :

$$k_{p,1} = k_{p,2} = \dots = k_{p,6} = 4900 \quad \text{et} \quad k_{v,1} = k_{v,2} = \dots = k_{v,6} = 99$$

- Etant donné que, certains paramètres diagonaux nominaux du système sont de l'ordre de 1, on initialise  $\hat{A}^{-1}(q)$  par  $I_6$ .
- Pour la résolution de l'équation de Lyapunov on prend  $Q = 100I_6$ .

Dans, un premier temps, nous allons simuler la commande sur le modèle nominal, puis pour tester sa robustesse on envisage une variation des paramètres de 50%.

#### 2.4.1.1. Cas idéal : sans erreurs de modélisation :

Les simulations ci-dessous concernent les résultats de l'application de la commande par découplage non linéaire adaptative au modèle nominal du système avec comme référence la trajectoire planaire décrite dans la figure 1.7.

Le rayon du cercle est de 0.45m et le temps de simulation est de  $T_e = 14s$ .

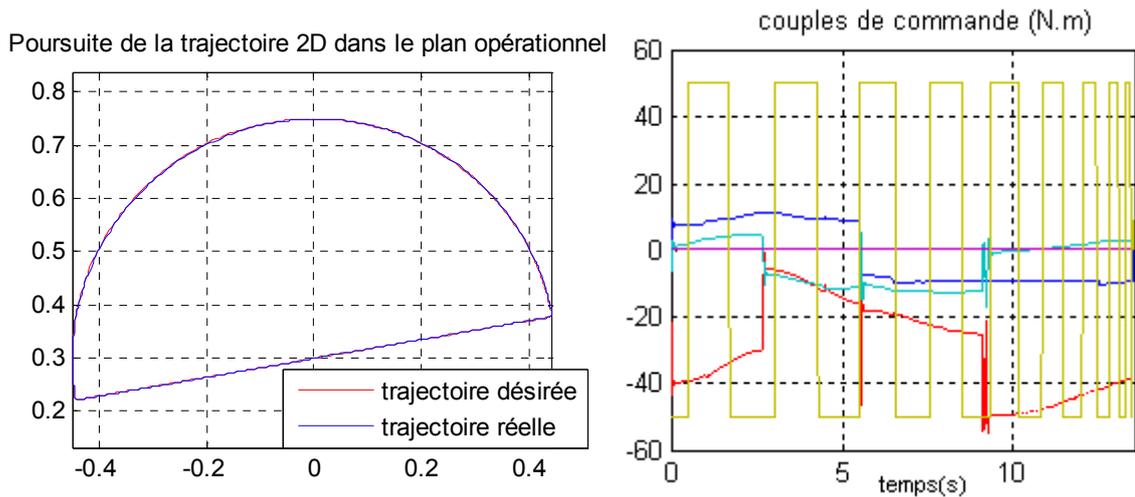


Fig.2.3. Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande par découplage non linéaire adaptatif appliquée au système nominal

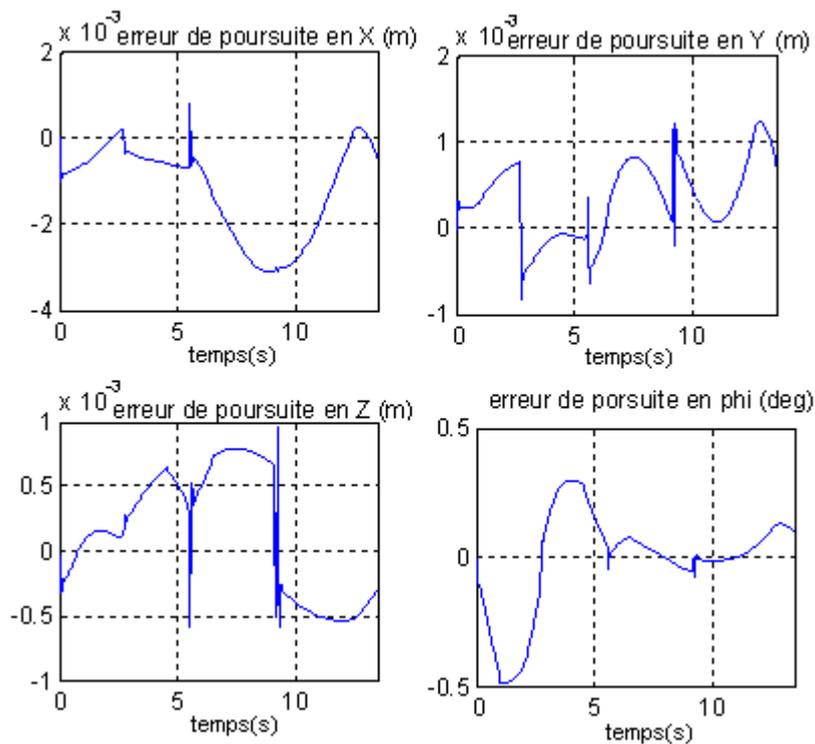


Fig.2.4. Signaux d'erreurs en position et orientation pour la trajectoire planaire avec la commande par découplage non linéaire adaptatif appliquée au système nominal.

La réponse du système montre qu'il y a une parfaite poursuite de la trajectoire tout en gardant des signaux de commande admissibles. Il est vrai que, par rapport à la commande par couple calculé, le quatrième couple articulaire subit des oscillations mais qui restent faibles, de l'ordre de 0.4Hz.

Les signaux d'erreur sont de l'ordre de 0.6% au maximum, en dépit des perturbations (les forces de frottement) et sont compris dans la zone tolérée.

### 2.4.1.2. En présence d'une erreur de modélisation de 50% :

Pour étudier la robustesse de la commande, on fait varier les paramètres du système (accroissement de 50%) tout en laissant ceux de la commande fixes. Les résultats de la simulation sont les suivants :

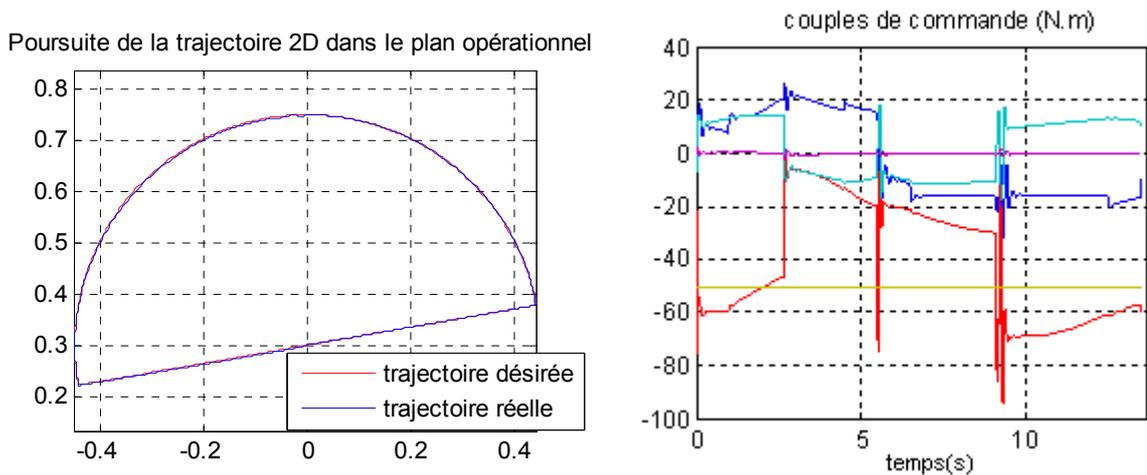


Fig.2.5. Réponse du système et signaux de commande pour la trajectoire planaire avec la commande par découplage non linéaire adaptative pour une variation des paramètres de 50%

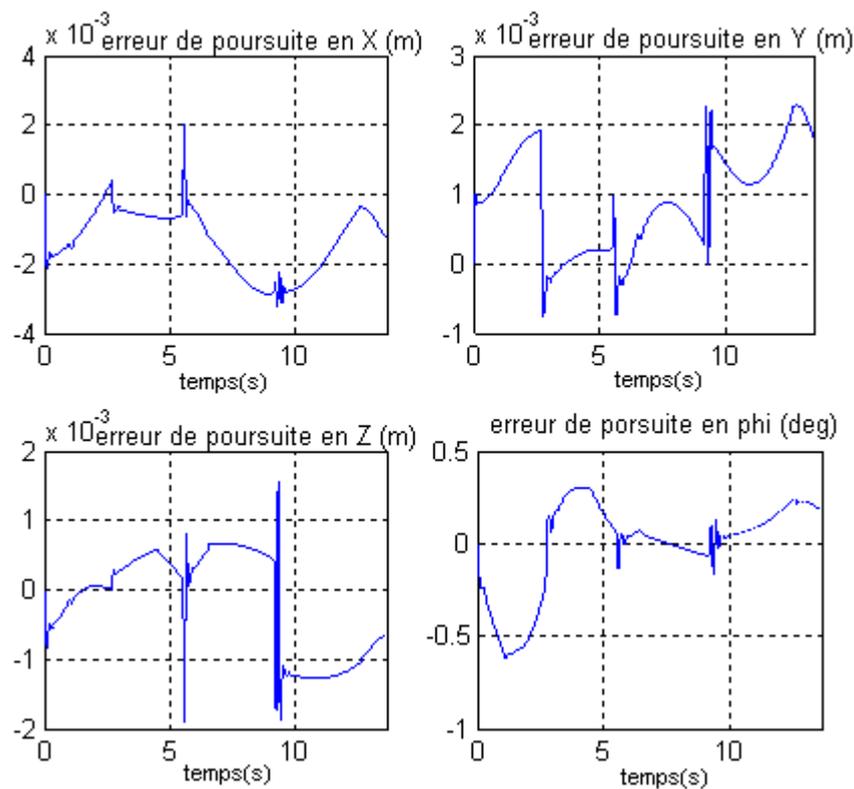


Fig.2.6. Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la commande par découplage non linéaire adaptative avec une variation des paramètres de 50%.

Les résultats de la simulation montrent, qu'il ya toujours une parfaite poursuite de la trajectoire avec les même plages d'erreurs de position et d'orientation, malgré une variation des paramètres de 50% et en présence des perturbations (les forces de frottement). Cela prouve la supériorité de cette commande par rapport à la commande par couple calculé. On remarque aussi qu'en présence de variation des paramètres, ce sont, la première, la deuxième et la sixième commande articulaire qui sont les plus sollicitées, plus particulièrement lors de l'exécution du tracé de l'arc du cercle, mais elles restent toujours dans les domaines admissibles d'amplitude et de fréquence. Cela démontre que, la nature de la tâche à exécuter à une grande influence sur les performances du régulateur.

### 2.4.2. Commande passive :

Les paramètres du régulateur sur lesquels notre choix s'est fixé sont :

- pour les  $\lambda_i$  on prend :  $\lambda_1 = \lambda_2 = \dots = \lambda_6 = 25$

En effet, le choix d'un  $\lambda_i$  supérieur augmente les performances de la poursuite mais en revanche, il accentue le phénomène du chattering, et donc un compromis s'impose. Après plusieurs essais, on a constaté que le choix d'un  $\lambda_i$  entre [20,30] permet une bonne poursuite tout en évitant l'excitation de la commande.

- Pour accélérer la rapidité de la simulation et éviter la divergence des paramètres, le gain d'adaptation doit être faible, on prend :  $\eta = 10^{-3} I_{30}$ .

Pour les gains du P.D on prend :  $k_{D,1} = k_{D,2} = \dots = k_{D,6} = 200$ . La présence du terme de robustesse «  $s$  » dans la commande diminue les gains.

#### 2.4.2.1. Cas idéal : sans erreurs de modélisation :

Les simulations ci-dessous concernent les résultats de l'application de la commande passive au modèle nominal du système avec comme référence la trajectoire planaire décrite dans la figure 1.10.

Le rayon du cercle est de 0.45 m et le temps de simulation est de  $T_e = 14s$ .

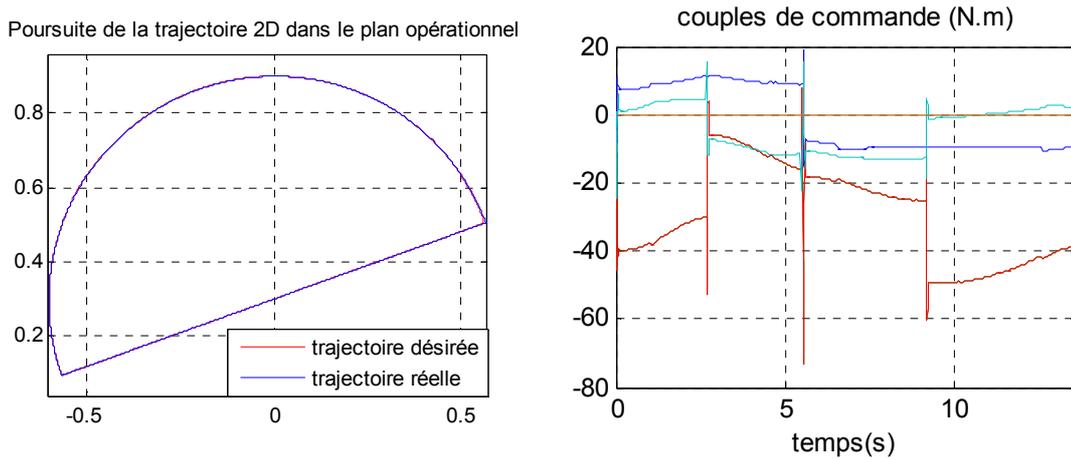


Fig.2.7. Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande adaptative passive appliquée au système nominal

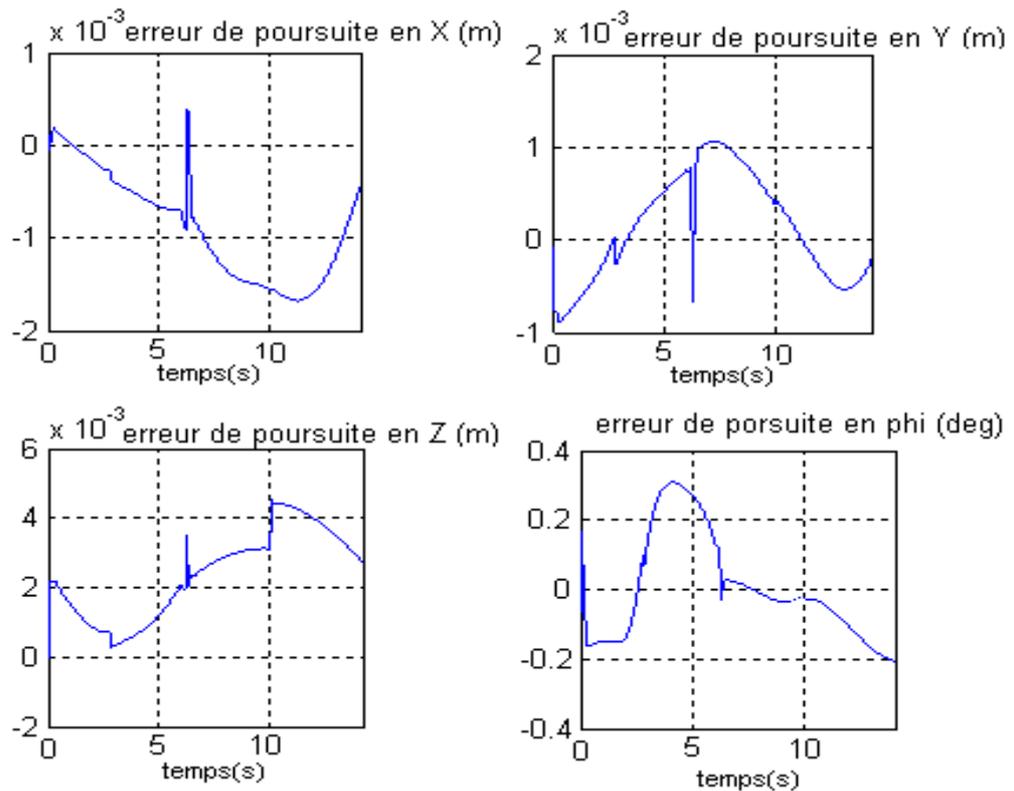


Fig.2.8. Signaux d'erreurs en position et orientation pour la trajectoire planaire avec la commande adaptative passive appliquée au système nominal.

Les résultats de la simulation indique qu'il y a une parfaite poursuite de la trajectoire, les erreurs de position et d'orientation sont de l'ordre de 1% au maximum, avec des signaux de commande admissibles.

### 2.4.2.2. En présence d'une erreur de modélisation de 50% :

On fait varier les paramètres du système (accroissement de 50%) tout en laissant ceux de la commande fixes. Les résultats de la simulation sont les suivants :

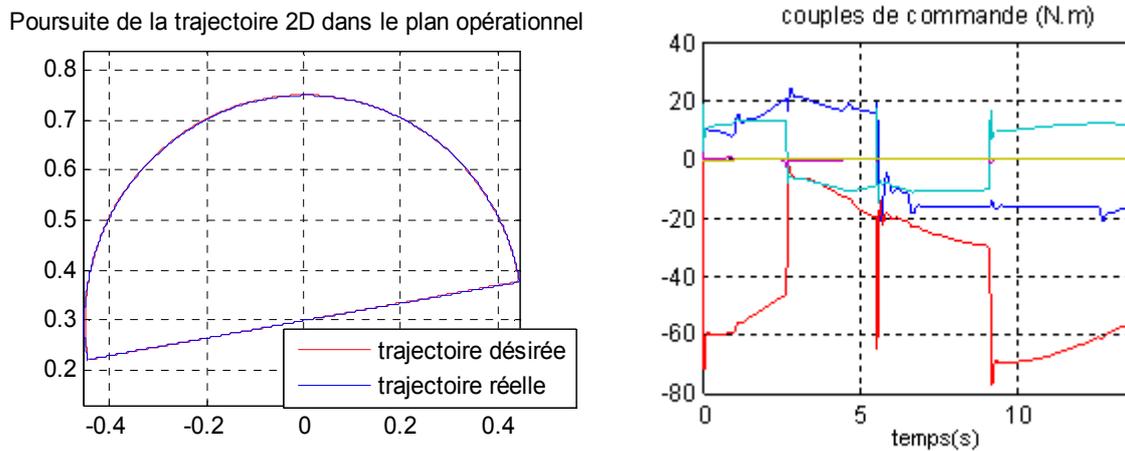


Fig.2.9. Réponse du système et signaux de commande pour la trajectoire planaire avec la commande adaptative passive pour une variation des paramètres de 50%

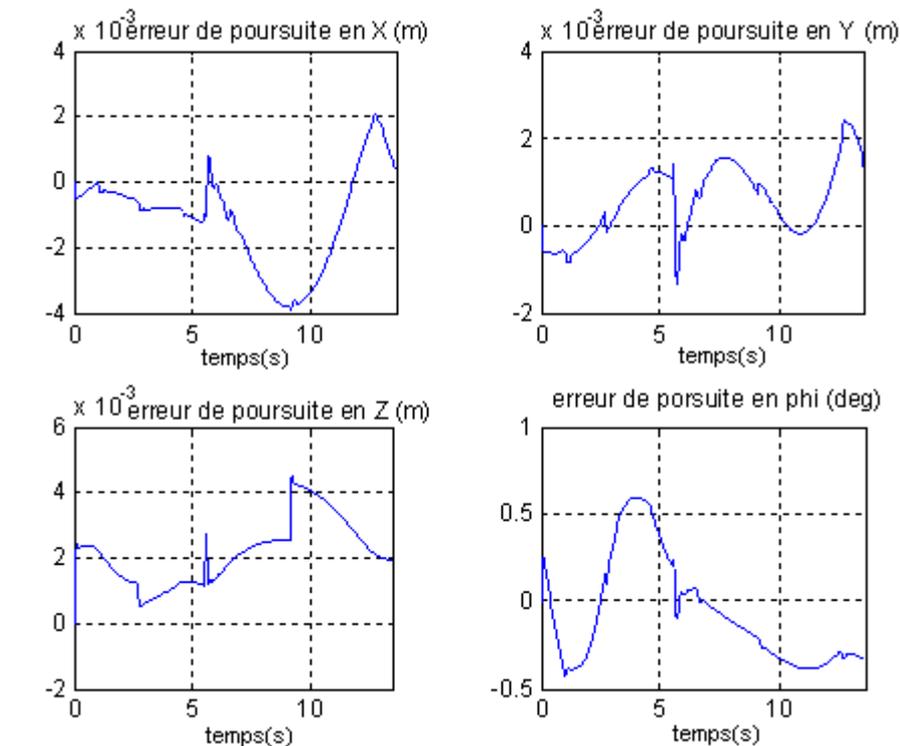


Fig.2.10. Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la commande adaptative passive pour une variation des paramètres de 50%.

On constate que malgré la variation des paramètres le régulateur adaptatif passif parvient toujours à garder les mêmes performances, à savoir une parfaite poursuite de la trajectoire avec les

même intervalles d'erreurs de position et d'orientation, avec des signaux de commande un peu plus important: ils atteignent les 80 N.M, particulièrement lors de l'exécution du tracé de l'arc du cercle, mais ils restent toujours dans les limites tolérées.

### **Conclusion :**

Le modèle dynamique d'un robot ne peut être parfait et il est naturel que les chercheurs appliquent les théories de la commande adaptative pour estimer ou ajuster en ligne les valeurs des paramètres dynamiques utilisés dans la commande.

Dans ce chapitre, nous avons traité deux types de commande adaptative, et qui sont les plus utilisées dans le monde de la robotique [1], à savoir la commande adaptative par découplage non linéaire et la commande adaptative passive. Dans les deux cas on a supposé que la forme du modèle est connu mais qu'on a aucune information a priori sur ces paramètres. Toutefois, une initialisation de l'estimée la matrice d'inertie est nécessaire dans la synthèse de la loi de commande adaptatif par découplage non linéaire, aussi pour cette commande une connaissance des accélérations articulaires est obligatoire, ce qui limite un peu son étendue. L'intérêt majeur de la loi de commande passive est son indépendance d'un calcul explicite du modèle du système.

L'application de la méthode indirecte de Lyapunov permet la synthèse de régulateurs stables tout en assurant la convergence des erreurs de poursuites vers zéro. Aussi, l'introduction du terme de glissement dans la commande passive augmente nettement ses performances.

Des simulations de ces lois de commande sur le robot PUMA 560 ont montré leur efficacité par rapport à la commande par couple calculé simple, en termes de poursuite, de signaux de commande et des marges de signaux d'erreurs (en position et en d'orientation). Ces simulations ont prouvé aussi que la commande indirecte (par découplage non linéaire adaptative) se basant sur l'estimation des fonctions du modèle afin de les injecter dans la commande l'emporte sur la commande directe passive. Toute fois, son inconvénient majeur reste la nécessité d'une initialisation.

Dans ce chapitre les commandes développées suppose que le modèle du système non linéaire peut s'exprimer sous la forme d'un produit de fonctions non linéaires connues par un vecteur de paramètres inconnus (constants ou lentement variable dans le temps). Cependant, il est difficile d'envisager tous les phénomènes qui rentrent en jeu dans le modèle d'un système complexe tel que le robot Une solution à ce problème peut être envisagée à partir de l'utilisation des systèmes flous comme approximateurs. Cette solution qui sera l'objet du chapitre suivant.

---

# 3

## SYNTHESE DE LA LOI DE COMMANDE ADAPTATIVE FLOUE

---

### 3.1. Introduction :

Dans le chapitre précédent, afin de palier le problème de la commande par couple calculé qui repose sur la connaissance parfaite du modèle, nous avons introduit la commande adaptative. Dans cette approche, on suppose que le modèle du système non linéaire peut s'exprimer sous la forme d'un produit de fonctions non linéaires connues par un vecteur de paramètres inconnus (constants ou lentement variable dans le temps). Malheureusement, il est souvent difficile de décrire tous les phénomènes mis en jeu dans un système à partir de fonctions non linéaires connues, notamment pour un système physique complexe tel que le robot. Une solution à ce problème peut être envisagée à partir d'une représentation de la dynamique du système non linéaire par un système flou où les relations entre les entrées, les sorties et les états sont exprimées par des règles floues de type « si-alors ». [19]

Les systèmes flous sont des systèmes à base de connaissances construits à partir des opérateurs de la logique floue. La logique floue a été conçue au milieu des années soixante à l'université de Berkeley Californie par le professeur iranien Lotfi A. Zadeh. Cette logique est basée sur l'imitation des aspects approximatifs et qualitatifs du raisonnement humain. La technique de commande floue est souvent présentée comme une technique robuste capable de compenser quasi naturellement les incertitudes de et/ou le manque de connaissances précises quant au processus à commander. Les systèmes flous sont des approximateurs universels [20] et fournissent une manière commode pour paramétriser les non-linéarités inconnues. En fait, ils peuvent approcher uniformément n'importe quelle fonction inconnue non linéaire continue sur un ensemble compact avec un degré de précision arbitraire. Notons qu'il existe d'autres approximateurs universels tels que : les réseaux de neurones (RN) multi couches, les fonctions radiales de base (radial basis fonctions (RBF)), les ondelettes, ....etc. [14]. Cependant, les systèmes flous sont les seuls capables d'incorporer d'une manière systématique la connaissance de l'expert et d'imiter ainsi des aspects approximatifs et qualitatifs du raisonnement humains [14]. La difficulté liée à la construction de la base des règles floues pour les

systèmes dynamiques complexes et le besoin d'améliorer la qualité d'approximation ont motivé Wang à introduire la commande floue adaptative [14] [21]. Dans ces schémas, la propriété d'approximation universelle des systèmes flous a été complètement exploitée, et la stabilité et la robustesse ont été étudiées par l'approche de Lyapunov. Depuis, plusieurs schémas de commande adaptative ont été proposés.

Dans ce chapitre, nous allons utiliser les systèmes flous dans le cadre d'une application liée à la commande adaptative centralisées et décentralisées pour la poursuite de la trajectoire de référence imposée à notre système non linéaire (le robot PUMA 560). On suppose que les non linéarités sont inconnues et que la sortie du système flou est supposé de type TS0 [21]. Pour faire face aux erreurs d'approximation floue (i.e. erreurs de reconstruction ou encore erreurs de modélisation) et perturbations externes bornées (forces de frottements), ces contrôleurs flous adaptatifs sont généralement augmentés par un terme robuste qui est la commande à mode glissant. La supposition clé dans ces schémas est que tous les états du système sont disponibles pour la mesure. Dans l'approche indirecte, pour contourner le problème de l'initialisation de l'estimée de la matrice d'inertie, nous utilisons l'inverse régulariser pour avoir une loi de commande bien définie avec une étude rigoureuse de la stabilité du système en boucle fermée [14] [21]. Et dans l'approche directe une loi d'adaptation paramétrique  $\sigma$ -modification de type PI est proposée.

De plus des simulations accompagneront chaque technique pour montrer et mettre en évidence ses performances.

### 3.2. Commande adaptative centralisée :

Dans ce qui suit nous allons exposer les approches de commande adaptative floue directe et indirecte développées dans [21-25] pour la classe des systèmes non linéaires à laquelle appartient notre robot.

Le modèle dynamique du robot peut être décrit par les équations différentielles suivantes :

$$\begin{cases} y_1^{(r_1)} = f_1(x) + \sum_{j=1}^p g_{1j}(x) u_j \\ \vdots \\ y_p^{(r_p)} = f_p(x) + \sum_{j=1}^p g_{pj}(x) u_j \end{cases} \quad (3.1)$$

Avec :  $p = 6$ ,  $y = q$  vecteur du variables articulaires,  $r_j$  le degré relatif de la sortie  $y_j$ , il est égale à deux,  $u = [u_1, \dots, u_p]^T$  est le vecteur d'entrée,  $x = [y_1, \dot{y}_1, \dots, y_p^{(r_1-1)}, \dots, y_p, \dot{y}_p, \dots, y_p^{(r_p-1)}]^T$  est le vecteur d'état supposé mesurable,  $y = [y_1, \dots, y_p]^T$  est le vecteur de sortie,  $f_i(x)$  et  $g_{ij}(x)$ ,  $i, j = 1, \dots, p$  sont des fonctions continues inconnues.

Posons :

$$Y^{(r)} = [y_1^{(r)}, \dots, y_p^{(r)}]^T$$

$$F(x) = [f_1(x), \dots, f_p(x)]^T$$

$$G(x) = \begin{bmatrix} g_{11}(x) & \dots & g_{1p}(x) \\ \vdots & \ddots & \vdots \\ g_{p1}(x) & \dots & g_{pp}(x) \end{bmatrix}$$

Par conséquent, le système dynamique (3.1) peut être réécrit sous la forme compacte suivante :

$$Y^{(r)} = F(x) + G(x)u \quad (3.2)$$

L'objectif de la commande consiste à synthétiser une loi de commande  $u(t)$  qui assure la bornitude de toutes les variables de la boucle fermée ainsi que le suivi pour les sorties du système  $y_1(t), \dots, y_p(t)$  des références  $y_{d1}(t), \dots, y_{dp}(t)$ .

**Hypothèse 3.1 :** La matrice  $G(x)$  est définie positive et satisfait :  $G(x) \geq \sigma_0 I_p$ , avec  $\sigma_0$  une constante positive donnée.

**Hypothèse 3.2 :** Chaque trajectoire de référence  $y_{di}(t)$  et ses dérivées sont supposées continues, bornées et connues.

Etant donné que la matrice d'inertie est définie positive (propriété 1 du modèle dynamique) et que les variables articulaires de référence et leurs dérivées sont bornées, les hypothèses 3.1 et 3.2 sont vérifiées. L'hypothèse sur  $G(x)$  est une condition suffisante pour la commandabilité du système.

Soient  $e_1(t), \dots, e_p(t)$  les erreurs de poursuite données par :

$$\begin{aligned} e_1(t) &= y_{d1}(t) - y_1(t) \\ &\vdots \\ e_p(t) &= y_{dp}(t) - y_p(t) \end{aligned} \quad (3.3)$$

Et  $s_1(t), \dots, s_p(t)$  les erreurs filtrées (surfaces de glissement de Slotine) définies par :

$$\begin{aligned} s_1(t) &= \left( \frac{d}{dt} + \lambda_1 \right) e_1(t) \\ &\vdots \\ s_p(t) &= \left( \frac{d}{dt} + \lambda_p \right) e_p(t) \end{aligned} \quad (3.4)$$

D'après (3.4), il vient que  $e_i(t) \rightarrow 0$  quand  $s_i(t) \rightarrow 0$ . Dans ce cas, l'objectif de la commande devient la synthèse d'une loi de commande assurant la convergence des surfaces de glissement  $s_i(t)$  vers zéro.

Les dérivées temporelles des erreurs filtrées sont données par :

$$\begin{aligned} \dot{s}_1 &= v_1 - f_1(x) - \sum_{j=1}^p g_{1j}(x) u_j \\ &\vdots \\ \dot{s}_p &= v_p - f_p(x) - \sum_{j=1}^p g_{pj}(x) u_j \end{aligned} \quad (3.5)$$

Avec :

$$\begin{aligned} v_1 &= y_{d1}^{(r_1)} + \beta_{1,r_1-1} e_1^{(r_1-1)} + \dots + \beta_{1,1} \dot{e}_1 \\ &\vdots \\ v_p &= y_{dp}^{(r_p)} + \beta_{p,r_p-1} e_p^{(r_p-1)} + \dots + \beta_{p,p} \dot{e}_p \end{aligned} \quad (3.6)$$

Et:

$$\beta_{i,j} = \frac{(r_i-1)!}{(r_i-j)! (j-1)!} \lambda_i^{r_i-j}, \quad i = 1, \dots, p, \quad j = 1, \dots, r_i - 1$$

En posant :

$$\begin{aligned} s(t) &= [s_1(t) \dots s_p(t)]^T \\ v(t) &= [v_1(t) \dots v_p(t)]^T \end{aligned}$$

On obtient la forme compacte suivante :

$$\dot{s} = v - F(x) - G(x)u \quad (3.7)$$

Si les fonctions  $F(x)$  et  $G(x)$  étaient connus alors, pour atteindre l'objectif qu'on s'est déjà fixé, on peut utiliser la loi de commande suivante :

$$u = G^{-1}(x)(-F(x) + v + K_0 s) \quad (3.8)$$

Avec :  $K_0 = \text{diag}(k_{01}, \dots, k_{0p})$  et  $k_{0i} > 0$  pour  $i = 1, \dots, p$

En effet, la substitution de (3.8) dans (3.7) on obtient :

$$\dot{s}(t) = -K_0 s(t) \quad (3.9)$$

Et par conséquent, si  $s(t) \rightarrow 0$  alors l'erreur  $e(t) \rightarrow 0$  ainsi que toutes ses dérivées.

Cependant, ces fonctions  $F(x)$  et  $G(x)$  sont supposées inconnues. Notre but est de les approcher par des systèmes flous.

### 3.2.1. Commande adaptative indirecte par les systèmes flous:

#### 3.2.1.1. Position du problème :

Nous proposons dans cette section l'utilisation des systèmes flous pour l'approximation des fonctions  $f_i(x)$  et  $g_{ij}(x)$  inconnues de la loi de commande (3.8).

Les fonctions non linéaires  $f_i(x)$  et  $g_{ij}(x)$  sont approchées par des systèmes flous type T-S :

$$\hat{f}_i(x, \theta_{f_i}) = w_{f_i}^T(x) \theta_{f_i}, \quad i = 1, \dots, p \quad (3.11)$$

$$\hat{g}_{ij}(x, \theta_{g_{ij}}) = w_{g_{ij}}^T(x) \theta_{g_{ij}}, \quad i, j = 1, \dots, p \quad (3.12)$$

Avec :  $w_{f_i}(x)$  et  $w_{g_{ij}}(x)$  sont des vecteurs de fonctions flous de base fixés par l'utilisateur,  $\theta_{f_i}$  et  $\theta_{g_{ij}}$  sont les vecteurs des paramètres ajustés.

$w_l = \frac{\mu_k}{\sum_{k=1}^N \mu_k}$  ;  $l = f_i, g_{ij}$  et  $\mu_k = \prod_{i=1}^n \mu_{F_i^k}(x_i)$ ,  $n$  étant le nombre des variables d'entrée et  $N$  le nombre totale des règles floues.

$$\text{Soient : } \theta_{f_i}^* = \arg \min_{\theta_{f_i}} \{ \sup_x |f_i(x) - \hat{f}_i(x, \theta_{f_i})| \} \quad (3.13)$$

$$\theta_{g_{ij}}^* = \arg \min_{\theta_{g_{ij}}} \{ \sup_x |g_{ij}(x) - \hat{g}_{ij}(x, \theta_{g_{ij}})| \} \quad (3.14)$$

Les paramètres optimaux de  $\theta_{f_i}$  et  $\theta_{g_{ij}}$ . Notons que les paramètres  $\theta_{f_i}^*$  et  $\theta_{g_{ij}}^*$  sont des constantes artificielles introduites uniquement pour faire l'étude théorique, et la connaissance de leurs valeurs n'est pas nécessaire pour l'implantation de la loi de commande.

$$\text{Notons par : } \tilde{\theta}_{f_i} = \theta_{f_i}^* - \theta_{f_i}, \quad \tilde{\theta}_{g_{ij}} = \theta_{g_{ij}}^* - \theta_{g_{ij}}$$

les erreurs d'estimation paramétrique, et par :

$$\varepsilon_{f_i}(x) = f_i(x) - \hat{f}_i(x, \theta_{f_i}^*) \quad (3.15)$$

$$\varepsilon_{g_{ij}}(x) = g_{ij}(x) - \hat{g}_{ij}(x, \theta_{g_{ij}}^*) \quad (3.16)$$

Les erreurs d'approximation minimales réalisables, et qui correspondent aux erreurs d'approximation obtenues lorsque les paramètres optimaux sont utilisés. En supposant que les systèmes flous sont choisis de sorte que la propriété d'approximation des systèmes flous n'est pas violée, il est raisonnable de supposer que les erreurs d'approximation sont bornées, i.e.

$$|\varepsilon_{f_i}(x)| \leq \bar{\varepsilon}_{f_i}, \quad |\varepsilon_{g_{ij}}(x)| \leq \bar{\varepsilon}_{g_{ij}}$$

Où  $\bar{\varepsilon}_{fi}$  et  $\bar{\varepsilon}_{gij}$  sont des constantes positives inconnues.

Posons :

$$\hat{F}(x, \theta_f) = [\hat{f}_1(x, \theta_{f1}) \dots \hat{f}_p(x, \theta_{fp})]^T$$

$$\hat{G}(x, \theta_g) = \begin{bmatrix} \hat{g}_{11}(x, \theta_{g11}) & \dots & \hat{g}_{1p}(x, \theta_{g1p}) \\ \vdots & \ddots & \vdots \\ \hat{g}_{p1}(x, \theta_{gp1}) & \dots & \hat{g}_{pp}(x, \theta_{gpp}) \end{bmatrix}$$

$$\varepsilon_f(x) = [\varepsilon_{f1}(x) \dots \varepsilon_{fp}(x)]^T$$

$$\varepsilon_g(x) = \begin{bmatrix} \varepsilon_{g11}(x) & \dots & \varepsilon_{g1p}(x) \\ \vdots & \ddots & \vdots \\ \varepsilon_{gp1}(x) & \dots & \varepsilon_{gpp}(x) \end{bmatrix}$$

$$\bar{\varepsilon}_f = [\bar{\varepsilon}_{f1} \dots \bar{\varepsilon}_{fp}]^T$$

$$\bar{\varepsilon}_g = \begin{bmatrix} \bar{\varepsilon}_{g11} & \dots & \bar{\varepsilon}_{g1p} \\ \vdots & \ddots & \vdots \\ \bar{\varepsilon}_{gp1} & \dots & \bar{\varepsilon}_{gpp} \end{bmatrix}$$

D'après l'analyse précédente, il vient que :

$$F(x) - \hat{F}(x, \theta_f) = \hat{F}(x, \theta_f^*) - \hat{F}(x, \theta_f) + \varepsilon_f(x) \quad (3.17)$$

$$G(x) - \hat{G}(x, \theta_g) = \hat{G}(x, \theta_g^*) - \hat{G}(x, \theta_g) + \varepsilon_g(x) \quad (3.18)$$

Maintenant, considérons la loi de commande  $u = u_c$  où  $u_c$  est une loi de commande équivalente

donnée par : 
$$u_c = \hat{G}^{-1}(x, \theta_g)(-\hat{F}(x, \theta_f) + v + K_0 s) \quad (3.19)$$

Cette loi de commande est obtenue à partir de (3.18) en substituant les fonctions réelles  $F(x)$  et  $G(x)$  par leurs approximations floues  $\hat{F}(x, \theta_f)$  et  $\hat{G}(x, \theta_g)$ .

La loi de commande (3.19) n'est pas définie lorsque la matrice  $\hat{G}(x, \theta_g)$  n'est pas inversible.

Pour éviter ce problème, on remplace  $\hat{G}^{-1}(x, \theta_g)$  par son inverse régularisé et ainsi la loi de commande équivalente devient:

$$u_c = \hat{G}^T(x, \theta_g) [\varepsilon_0 I_p + \hat{G}(x, \theta_g) \hat{G}^T(x, \theta_g)]^{-1} (-\hat{F}(x, \theta_f) + v + K_0 s) \quad (3.20)$$

Avec :  $\varepsilon_0$  une constante positive petite.

Notons que la loi donnée dans (3.20) ne peut garantir à elle seule la stabilité du système bouclé. Ceci est dû, d'un côté aux erreurs d'approximation des fonctions réelles  $F(x)$  et  $G(x)$  par des systèmes flous et d'un autre côté à l'erreur introduite par l'utilisation de l'inverse régularisé. Pour ces raisons, et afin d'avoir une loi de commande ne dépendant d'aucune phase d'initialisation nous proposons d'ajouter un terme de commande robuste et la loi de commande devient [21] :

$$u = u_c + u_r \quad (3.21)$$

Cette loi de commande est la somme de deux termes : le premier est la commande équivalente  $u_c$  modifiée et le second est la commande robuste  $u_r$  introduite pour palier le problème des erreurs de reconstruction.

La commande équivalente modifiée est donnée par :

$$u_c = \bar{G}^T(x, \theta_g) [\varepsilon_0 I_p + \bar{G}(x, \theta_g) \bar{G}^T(x, \theta_g)]^{-1} (-\bar{F}(x, \theta_f) + v + K_0 s + k_1 \bar{G}(x, \theta_g) s) \quad (3.22)$$

Et le terme de commande robuste par :

$$u_r = \frac{s |s^T| (\hat{\varepsilon}_f + \hat{\varepsilon}_g |u_c| + |u_0|)}{\sigma_0 \|s\|^2 + \delta} \quad (3.23)$$

Avec :

$$u_0 = \varepsilon_0 [\varepsilon_0 I_p + \bar{G}(x, \theta_g) \bar{G}^T(x, \theta_g)]^{-1} (-\bar{F}(x, \theta_f) + v + K_0 s + k_1 \bar{G}(x, \theta_g) s) \quad (3.24)$$

Et  $\hat{\varepsilon}_f$  et  $\hat{\varepsilon}_g$  sont respectivement les estimés de  $\bar{\varepsilon}_f$  et  $\bar{\varepsilon}_g$ ,  $k_1 > 0$ , et  $\delta$  un paramètre variable dans le temps défini ci-dessous.

L'étude de la stabilité du système en boucle fermée permet de choisir les lois d'adaptation suivantes [21] :

$$\dot{\theta}_{fi} = -\eta_{fi} w_{fi}(x) s_i \quad (3.25)$$

$$\dot{\theta}_{gij} = -\eta_{gij} w_{gij}(x) s_i (u_{ej} - k_1 s_j) \quad (3.26)$$

$$\dot{\hat{\varepsilon}}_f = \gamma_f |s| \quad (3.27)$$

$$\dot{\hat{\varepsilon}}_g = \gamma_g |s| |u_c^T| \quad (3.28)$$

$$\hat{\delta} = \frac{-\eta_0 |s^T| (\hat{\varepsilon}_f + \hat{\varepsilon}_g |u_c| + |u_0|)}{\sigma_0 |s|^2 + \delta} \quad (3.29)$$

$\eta_{fi} > 0, \eta_{gij} > 0, \gamma_f > 0, \gamma_g > 0, \eta_0 > 0$  et  $\delta(0) > 0$ .

**Théorème 3.1 :** Considérons le système non linéaire (3.1), et supposons que les hypothèses 3.1 et 3.2 sont satisfaites. Alors, la loi de commande définie par les équations (3.21) et (3.24) avec les lois d'adaptation (3.25) et (3.29) appliquées au système (3.1) garantissent la bornitude de tous les signaux de la boucle fermée et la convergence vers zéro des erreurs de poursuite.

Pour la démonstration du théorème voir [21].

### 3.2.1.2. Simulations et interprétations :

L'implantation de la commande adaptative indirecte centralisée floue pour la commande du robot PUMA 560 se fait sous SIMULINK®. Pour la résolution des équations différentielles, la méthode utilisée est l'ODE 45. Les conditions de simulation sont les même déjà mentionnée dans le chapitre de la commande adaptative conventionnelle.

Vu le nombre de calcul important que demande cette technique de commande nous nous sommes contentés d'implanter uniquement la loi de commande équivalente donnée dans (3.22)

Pour les paramètres du régulateur nous avons choisi :

- Nous avons pris six systèmes flous pour l'estimation de  $F(x)$ , et trente-six systèmes flous pour l'estimation de  $G(x)$ .
- Pour chaque variable  $x_i, i = 1, \dots, 12$ , on définit trois fonctions d'appartenance gaussienne, dont les centres et les variances sont choisis en se basant sur ceux des signaux de référence :

$$\mu_1 = \exp \left( -\frac{1}{2} \left( \frac{x+1}{0.5} \right)^2 \right), \mu_2 = \exp \left( -\frac{1}{2} \left( \frac{x-0.01}{0.5} \right)^2 \right), \mu_3 = \exp \left( -\frac{1}{2} \left( \frac{x-2}{0.5} \right)^2 \right)$$

- Les valeurs des paramètres ajustés sont prises égales à zéro, i.e.,  $\theta_f(0) = 0, \theta_g = 0$
- Les valeurs des paramètres de conception utilisées sont :  $\lambda_1 = \lambda_2 = \dots = \lambda_6 = 25,$   
 $K_0 = 400I_6, k_1 = 50, \eta_{fi} = \eta_{gij} = 0.0001$

### 3.2.1.2.1. Cas idéal : sans erreurs de modélisation :

Les simulations ci-dessous concernent les résultats de l'application de la commande adaptative floue indirecte au modèle nominal du système avec comme référence la trajectoire planaire décrite dans la figure 1.10.

Le rayon du cercle est de 0.45m et le temps de simulation est de  $T_e = 14s$

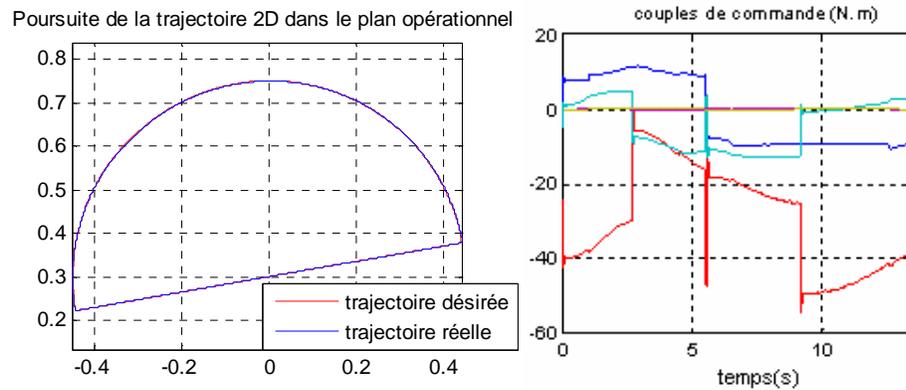


Fig.3.1. Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande adaptative floue centralisée indirecte appliquée au système nominal

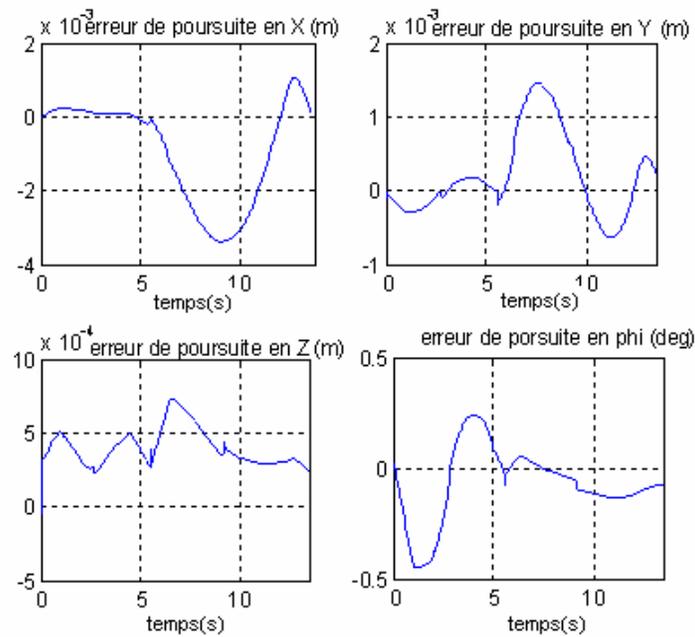


Fig.3.2. Signaux d'erreurs en position et orientation pour la trajectoire planaire avec la commande adaptative floue centralisée indirecte appliquée au système nominal.

La réponse du système montre qu'il y a une parfaite poursuite de la trajectoire tout en gardant des signaux de commande admissibles. Par rapport à la commande adaptative conventionnelle les

marges d'erreur sont identiques (de l'ordre de 0.6%). L'avantage principal de cette commande est sa capacité d'amener le système à exécuter sa tâche sans aucune connaissance a priori du système.

**3.2.1.2.2. En présence d'une erreur de modélisation de 50% :**

Afin d'étudier la robustesse de la commande, on fait varier les paramètres du système (accroissement de 50%) tout en laissant ceux de la commande fixes. Les résultats de la simulation sont les suivants :

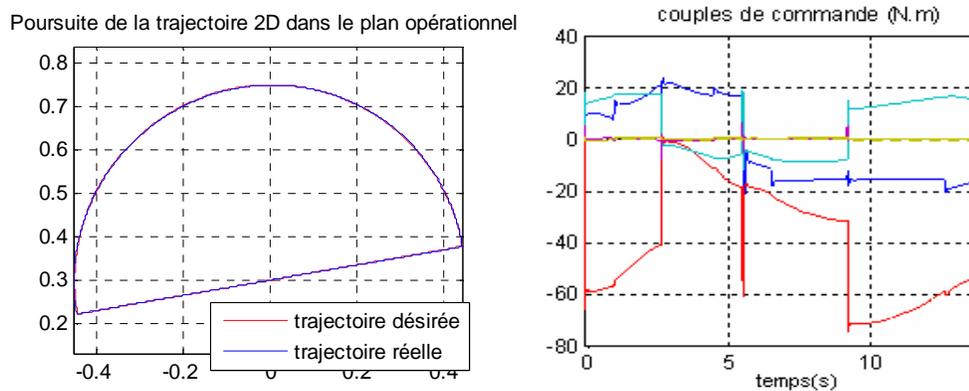


Fig.3.3. Réponse du système et signaux de commande pour la trajectoire planaire avec la commande adaptative floue indirecte centralisée pour une variation des paramètres de 50%

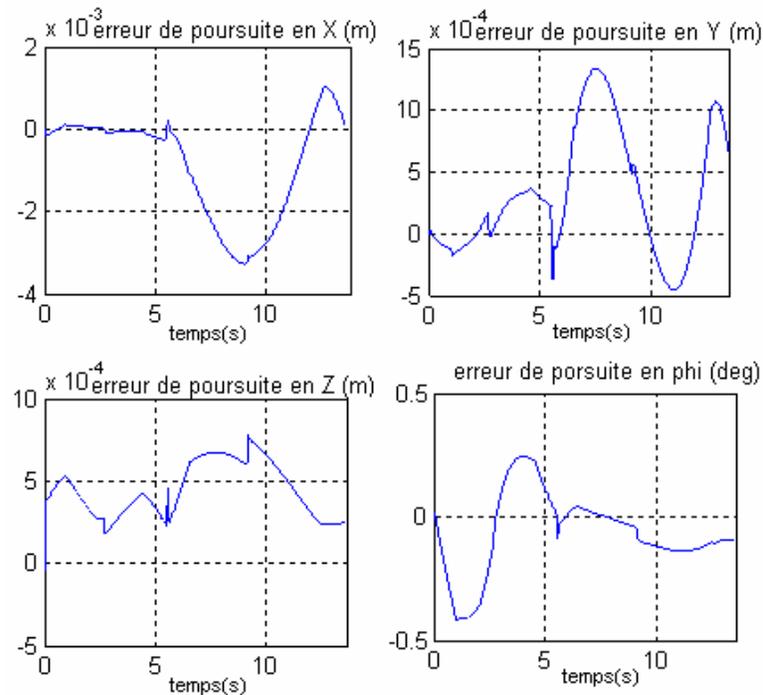


Fig.3.4. Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la commande adaptative floue indirecte centralisée avec une variation des paramètres de 50%.

Les résultats de la simulation montrent, qu'il ya toujours une parfaite poursuite de la trajectoire tout en gardant les même plages des erreurs de position et d'orientation cela est dû au fait que, loi de commande utilisée est indépendante de toute information concernant le modèle du système, ainsi la propriété d'approximateurs universels qui caractérise les systèmes flous est vérifiée.

### 3.2.2. Commande adaptative centralisée directe floue :

#### 3.2.2.1. Position du problème :

L'objectif de cette commande est d'utiliser les systèmes flous afin d'estimer la commande dans son ensemble tout en assurant la convergence des signaux d'erreurs et de leurs dérivées vers zéro.

Les hypothèses suivantes sont nécessaires [21]:

**Hypothèse 3.3 :** Dans l'ensemble compact  $D$ , chaque élément diagonal  $g_{ii}(x)$  est borné, non nul et sa dérivée temporelle est bornée par une fonction positive continue inconnue  $d_i(x)$ , c'est-à-dire,  $0 \leq |\dot{g}_{ii}(x)| < d_i(x)$ .

**Hypothèse 3.4 :** Chaque élément hors de la diagonale de la matrice  $G(x)$  vérifie :

$$\left| \frac{g_{ij}(x)}{g_{ii}(x)} \right| \leq \bar{g}_{ij}, \quad \forall x \in D$$

Où  $\bar{g}_{ij}$  est une constante positive inconnue.

Posons : 
$$G_d(x) = \text{diag}(g_{11}(x), \dots, g_{pp}(x))$$

$$D(x) = \text{diag}(d_1(x), \dots, d_p(x))$$

D'après [21], si les fonctions  $F(x)$ ,  $G(x)$  et  $D(x)$  étaient connues, l'objectif de la commande peut être atteint en choisissant la loi de commande idéale suivante :

$$u^* = [G_d^{-1}(x) G(x)]^{-1} \left( -G_d^{-1}(x)F(x) + G_d^{-1}(x)v + Ks + \frac{1}{2}G_d^{-1}(x)D(x)G_d^{-1}(x)s \right) \quad (3.30)$$

Où  $K = \text{diag}(k_1, \dots, k_p)$ , avec  $k_i > 0$  pour  $i = 1, \dots, p$ .

L'objectif est de pouvoir approcher la commande idéale (3.30) où les fonctions  $F(x)$ ,  $G(x)$  et  $D(x)$  sont supposées inconnues mais satisfaisant les hypothèses (3.1, 3.2, 3.3 et 3.4), afin de réaliser

une poursuite de trajectoires prédéfinies. Pour atteindre cet objectif, les systèmes flous sont utilisés afin d'estimer la commande dans son ensemble (approche directe).

On suppose que la loi de commande se décompose en une loi connue  $u_k^*$ , et d'une partie inconnue à estimer  $u_{ui}^*$ , i.e., la commande idéale (3.30) s'écrit [21]:

$$u^* = u_{ui}^*(\bar{x}) + u_k^*(\bar{x}) \quad (3.31)$$

Avec :  $\bar{x} = [x^T, s^T, v^T]^T$ .

Il est à noter que le terme  $u_k^*$ , dans l'équation (3.31) est introduit pour permettre à l'utilisateur d'exploiter toute information mathématique ou linguistique disponible sur la commande du système.

Le terme  $u_{ui}^*$  peut, par exemple, correspondre à une expertise préalable, une loi existante issue de l'application des méthodes classiques de commande, une loi issue du modèle nominal du système, etc. Le rôle de ce terme est d'améliorer les performances du contrôleur, l'étude de la stabilité de la loi de commande globale est indépendante de ce terme et donc par la suite on le prend égal à zéro.

Supposons que la partie inconnue  $u_{ui}^*$  est une fonction continue, chacune de ses composantes peut être approchée par un système flou, de la façon suivante :

$$u_{ui}(\bar{x}, \theta_i) = w_i^T(\bar{x}) \theta_i, \quad i = 1, \dots, p \quad (3.32)$$

Avec :  $\theta_i$  le vecteur des paramètres ajustés et  $w_i^T(\bar{x})$  un vecteur de fonctions floues normalisées de base supposé fixé par l'utilisateur.

La fonction réelle  $u_{ui}^*(\bar{x})$  peut être exprimée de la façon suivante :

$$u_{ui}^*(\bar{x}) = u_{ui}(\bar{x}, \theta_i^*) + \varepsilon_i(\bar{x}) \quad (3.33)$$

Avec :  $\varepsilon_i(\bar{x})$  l'erreur d'approximation de la partie inconnue par un système flou, et  $\theta_i^*$  le vecteur des paramètres optimaux de  $\theta_i$  minimisant l'erreur  $|\varepsilon_i(\bar{x})|$  et satisfaisant :

$$\theta_i^* = \arg \min_{\theta_i} \left\{ \sup_{\bar{x}} |u_{ui}^*(\bar{x}) - u_{ui}(\bar{x}, \theta_i)| \right\} \quad (3.34)$$

En posant :

$$\theta = [\theta_1, \dots, \theta_p]^T$$

$$w(\bar{x}) = \text{diag}(w_1(\bar{x}), \dots, w_p(\bar{x}))$$

$$\varepsilon(\bar{x}) = [\varepsilon_1(\bar{x}), \dots, \varepsilon_p(\bar{x})]^T$$

On peut écrire :

$$u_u(\bar{x}, \theta) = w^T(\bar{x})\theta$$

$$u_u^*(\bar{x}) - u_u(\bar{x}, \theta) = w^T(\bar{x})\bar{\theta} + \varepsilon(\bar{x})$$

Avec :  $\bar{\theta} = \theta^* - \theta$  l'erreur paramétrique.

**Hypothèse 3.5 :** L'erreur d'approximation est bornée :  $\|\varepsilon(\bar{x})\| \leq \bar{\varepsilon}$ .

**Hypothèse 3.6 :** Les paramètres optimaux  $\theta^*$  sont bornés :  $\|\theta^*\| \leq M$

Où  $M$  est une constante positive inconnue.

Du fait que les systèmes flous utilisés pour approcher la commande idéale sont supposés possédant la propriété d'approximation universelle, les hypothèses (3.5 et 3.6) sont raisonnables.

De plus on n'exige pas dans l'hypothèse 3.5 que la borne  $\bar{\varepsilon}$  soit connue et/ou petite, mais seulement qu'elle soit constante. L'hypothèse 3.6 est peu contraignante puisque les paramètres optimaux sont des constantes et que la borne  $M$  n'est pas explicite. Cependant, réduire les erreurs d'approximation peut entraîner un nombre important de fonctions floues de base ce qui peu compliquer les calculs.

L'approximation de la loi de commande (3.31) s'écrit alors :

$$u = u_u(\bar{x}, \theta) + u_k^*(\bar{x}) \quad (3.35)$$

L'estimation des paramètres suit la loi d'adaptation suivante [21] :

$$\dot{\theta}(t) = \eta_1 \int_0^t (w(\bar{x})s - \eta_2 \theta) d\tau + \eta_3 (w(\bar{x})s - \eta_2 \theta) \quad (3.36)$$

Qui peut s'écrire de la forme :

$$\dot{\theta}(t) = \eta_1 (w(\bar{x})s - \eta_2 \theta) + \dot{\varphi}(t) \quad (3.37)$$

Avec :

$$\dot{\varphi}(t) = \eta_3 (w(\bar{x})s - \eta_2 \theta) \quad (3.38)$$

Et:  $\eta_1$ ,  $\eta_2$  et  $\eta_3$  des constantes positives choisies par l'utilisateur.

Cette loi d'adaptation est de type « proportionnel intégral à  $\sigma$ -modification ». Cette forme de loi permet de rendre le contrôleur plus robuste et d'éviter le problème « d'explosion des paramètres » estimés [21].

Pour la démonstration voir [21].

**Théorème 3.2 :** Considérons le système (3.1) et supposons que les hypothèses 3.1-3.6 sont satisfaites. En utilisant la loi de commande définie par (3.32) et la loi d'adaptation donnée par (3.36) (ou (3.37) et (3.38)), tous les signaux du système en boucle fermée sont bornés [21].

Pour la démonstration voir [21].

### 3.2.2.2. Simulations et interprétations :

L'implantation de la commande adaptative directe centralisée floue pour la commande du robot PUMA 560 se fait sous SIMULINK<sup>®</sup>. Pour la résolution des équations différentielles, la méthode utilisée est l'ODE 45. Les conditions de simulation sont les mêmes déjà mentionnées dans le chapitre de la commande adaptative conventionnelle.

- Six systèmes flous sont utilisés pour estimer les lois de commande  $u_1, u_2, \dots, u_6$ .

Chaque système flou a pour entrée le vecteur  $\bar{x} = [q_1 \dot{q}_1 \dots q_6 \dot{q}_6 s_1 \dots s_6]^T$ . Notons que, les variables  $v_i, i = 1, \dots, 6$  ne sont pas incluses dans le vecteur d'état afin de réduire la taille des systèmes flous et, d'autre part, du fait que ses variables sont fonctions des erreurs de poursuite qui sont incluses dans les signaux  $s_i, i = 1, \dots, 6$ .

- Pour chaque variable d'entrée  $\bar{x}_i$  on définit trois fonctions d'appartenance gaussiennes données par :

$$\mu_1(\bar{x}_i) = \exp\left(-\frac{1}{2}\left(\frac{\bar{x}_i+1}{0.5}\right)^2\right), \mu_2(\bar{x}_i) = \exp\left(-\frac{1}{2}\left(\frac{\bar{x}_i-0.01}{0.5}\right)^2\right), \mu_3(\bar{x}_i) = \exp\left(-\frac{1}{2}\left(\frac{\bar{x}_i-2}{0.5}\right)^2\right)$$

- Les paramètres des systèmes flous utilisés sont initialisés à zéro.
- Les différents paramètres nécessaires à la synthèse de la loi de commande sont :

$$\lambda_1 = \lambda_2 = \dots = \lambda_6 = 25, \eta_1 = \eta_3 = 400, \eta_2 = 0.0001.$$

#### 3.2.2.2.1. Cas idéal : sans erreurs de modélisation :

Le rayon du cercle est de 0.45m et le temps de simulation est de  $T_s = 14s$ .

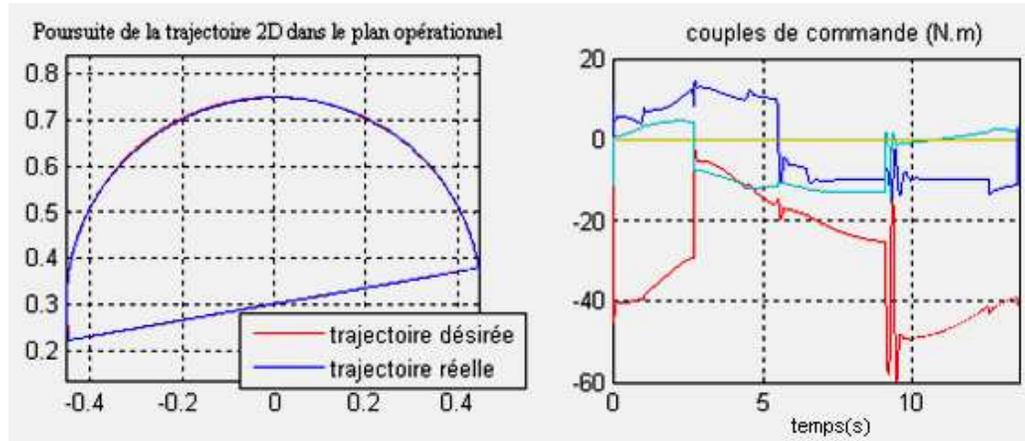


Fig.3.5. Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande adaptative floue centralisée directe appliquée au système nominal

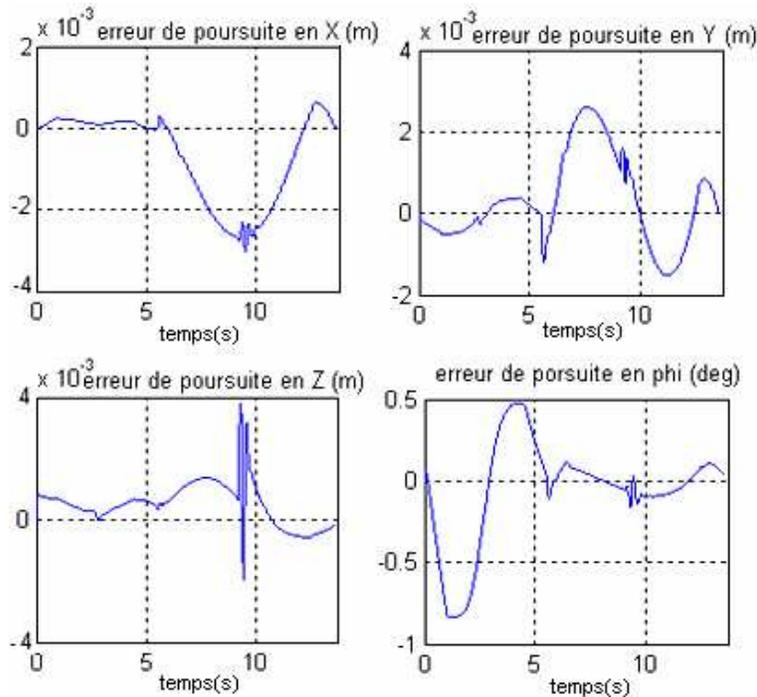


Fig.3.6. Signaux d'erreurs en position et orientation pour la trajectoire planaire avec la commande adaptative floue centralisée directe appliquée au système nominal.

Les résultats de la simulation montre qu'il y a une parfaite poursuite de la trajectoire et la convergence des erreurs de position et d'orientation vers zéro (de l'ordre de 0.8%), tout en gardant les signaux de commande dans les limites tolérées.

L'ordre des erreurs de poursuite est équivalent à celui la commande centralisée indirecte, mais le temps de calcul est beaucoup plus réduit aussi la programmation est plus aisée.

### 3.2.2.2.2. En présence d'une erreur de modélisation de 50% :

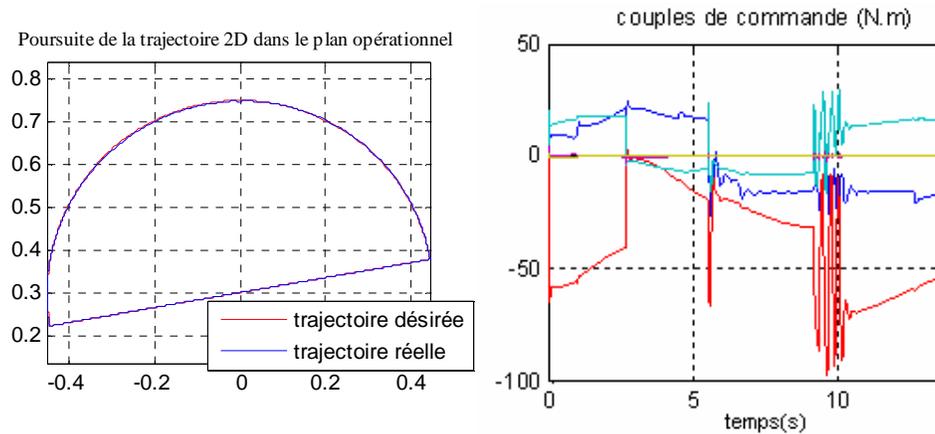


Fig.3.7. Réponse du système et signaux de commande pour la trajectoire planaire avec la commande adaptative floue directe centralisée pour une variation des paramètres de 50%

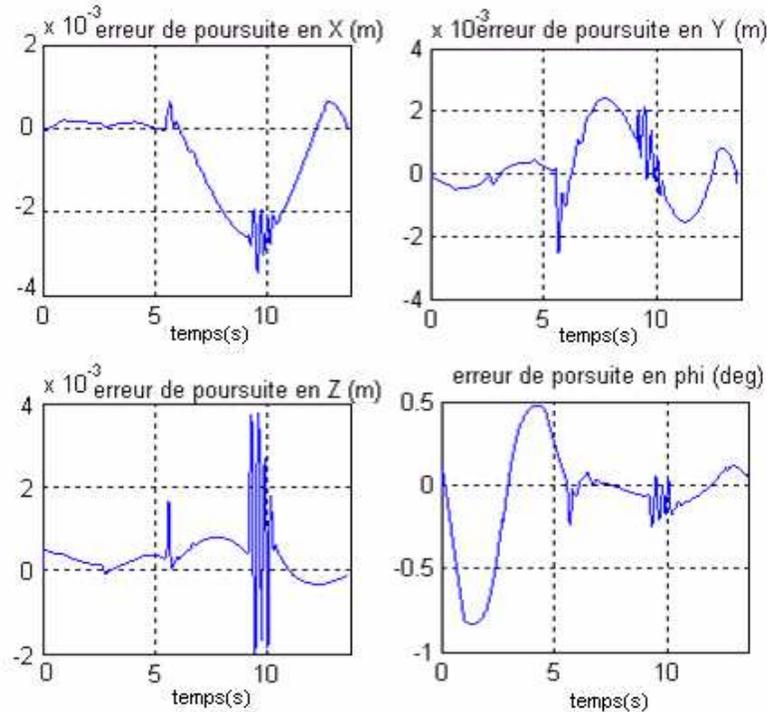


Fig.3.8. Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la commande adaptative floue directe centralisée avec une variation des paramètres de 50%.

Ces simulations approuvent la propriété d'approximateurs universels qui caractérise les systèmes flous. Malgré, la variation des paramètres les sorties du système en boucle fermée parviennent toujours à traquer les signaux de référence avec une erreur qui tend vers zéro, d'où la parfaite poursuite de la trajectoire.

### 3.3. Commande adaptative floue décentralisée :

Le problème de la commande décentralisée se pose lorsqu'on considère la commande des systèmes interconnectés composés de plusieurs sous-systèmes pouvant être géographiquement éloignés. En effet, la synthèse d'une loi de commande centralisée peut être complexe et fastidieuse, due à la nature modulaire du système.

Vu que le modèle du robot appartient à la classe des systèmes non linéaires interconnectés avec des sous-systèmes non linéaires incertains et des interconnexions inconnues, dans cette section, nous allons essayer de lui appliquer les approches déjà développées dans [21-28] dans le cadre de la commande adaptative floue décentralisée. Les approches proposées peuvent être classées dans deux catégories : l'approche directe et l'approche indirecte. L'approche indirecte se base sur l'approximation des non linéarités locales de chaque sous système par des modèles flous. En revanche, les approches directes se basent sur l'approximation par des systèmes flous des lois de commandes stabilisantes inconnues. La structure décentralisée rend l'implantation sur des architectures parallèles possibles, et de ce fait, le temps de calcul peut être considérablement réduit par rapport à la commande centralisée.

#### 3.3.1. Commande adaptative décentralisée indirecte floue :

##### 3.3.1.1. Position du problème :

Considérons un système non linéaire interconnecté  $\Sigma$  composé de  $N$  sous-systèmes non linéaires  $\Sigma_i, i = 1, \dots, N$ , donné par :

$$\Sigma_i \begin{cases} \dot{x}_{i1} = x_{i2}, \dot{x}_{i2} = x_{i3}, \dots, \dot{x}_{i(n_i-1)} = x_{in_i} \\ \dot{x}_{in_i} = f_i(x_i) + g_i(x_i)u_i + \Delta_i(x_1, \dots, x_N) \\ y_i = x_{i1} \end{cases} \quad (3.39)$$

Où encore :

$$\dot{x}_i^{(n_i)} = f_i(x_i) + g_i(x_i)u_i + \Delta_i(x_1, \dots, x_N) \quad (3.40)$$

Avec :  $x_i = [x_{i1}, \dots, x_{in_i}]^T \in R^{n_i}$  le vecteur d'état du sous systèmes  $\Sigma_i$  supposé mesurable,  $n_i$  le degré relatif de la variable d'état  $x_i$ ,  $u_i \in R$  est l'entrée de commande,  $y_i \in R$  est la sortie,  $f_i(x_i)$  et  $g_i(x_i)$  sont des fonctions non linéaires continues supposées inconnues, et  $\Delta_i(x_1, \dots, x_N)$  représente l'effet des interconnexions entre le sous systèmes  $\Sigma_i$  est autres sous systèmes.

L'objectif de commande est la synthèse des lois de commande adaptatives utilisant les systèmes flous pour chaque sous systèmes et ceci, en utilisant uniquement les informations locales (variables d'état locales) propre à chaque sous système, de telle sorte que chaque sortie  $y_i(t)$  suit le plus proche possible, une trajectoire de référence donnée  $y_{di}(t)$  tout en garantissant la bornitude de tous les signaux du système en entier.

Pour chaque sous-système  $\Sigma_i$ , on définit les erreurs de poursuites suivantes :

$$e_i(t) = y_{di}(t) - y_i(t), \dots, e_i^{(n_i-1)}(t) = y_{di}^{(n_i-1)}(t) - y_i^{(n_i-1)}(t) \quad (3.41)$$

Et le signal d'erreur :

$$v_i(t) = y_{di}^{(n_i)}(t) + k_{i,n_i-1}e_i^{(n_i-1)}(t) + \dots + k_{i,0}e_i(t) \quad (3.42)$$

Posons :

$$e_i = [e_i, \dot{e}_i, \dots, e_i^{(n_i-1)}]^T \quad (3.43)$$

$$k_i = [k_{i,0}, \dots, k_{i,n_i-1}]^T \quad (3.44)$$

D'où, on peut écrire :

$$v_i = y_{di}^{(n_i)} + k_i^T e_i \quad (3.45)$$

Par introduction de (3.45) dans (3.39), il vient que :

$$\dot{e}_i = A_i e_i + b_i \{v_i - f_i(x_i) - g_i(x_i)u_i - \Delta_i(x_1, \dots, x_N)\} \quad (3.46)$$

Avec :

$$A_i = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -k_{i,0} & -k_{i,1} & -k_{i,2} & \dots & -k_{i,n_i-1} \end{bmatrix} \in R^{n_i \times n_i} \quad (3.47)$$

$$b_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in R^{n_i} \quad (3.48)$$

Les paramètres de conception  $k_{i,0}, \dots, k_{i,n_i-1}$  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 :

$$P_i A_i + A_i^T P_i = -Q_i \quad (3.49)$$

Les hypothèses suivantes sont nécessaires par la suite :

**Hypothèse 3.8 :** le gain de la commande  $g_i(x_i)$  est non nul et de signe connu.

Notons que, dans [14] on propose une technique pour éliminer cette contrainte,

**Hypothèse 3.9 :** chaque trajectoire de référence  $y_{di}(t)$  et ses dérivées jusqu'à l'ordre  $n_i$  sont supposées bornées et connues.

**Hypothèse 3.10 :** Les interconnexions satisfont :  $|\Delta(x_1, \dots, x_N)| \leq \delta_{i,0} + \sum_{j=1}^N \delta_{i,j} \|e_j\|$  (3.50)

Avec :  $\delta_{i,0}, \dots, \delta_{i,N}$  sont des constantes positives inconnues, et  $\|\bullet\|$  représente la norme Euclidienne.

Si on suppose que les sous systèmes sont isolés l'un de l'autre, i.e.,  $\Delta(x_1, \dots, x_N) = 0$ , et que les fonctions non linéaires  $f_i(x_i)$  et  $g_i(x_i)$  sont parfaitement connues, alors pour atteindre les objectifs de commande, on opte pour la loi suivante [21]:

$$u_i = \frac{1}{g_i(x_i)} (v_i - f_i(x_i)) \quad (3.51)$$

Par substitution de (3.51) dans (3.46) on trouve :

$$\dot{e}_i = A_i e_i \quad (3.52)$$

Comme la matrice  $A_i$  est stable, on peut conclure que toutes les erreurs de poursuite convergent vers zéro.

Dans cette étude, les fonctions non linéaires  $f_i(x_i)$  et  $g_i(x_i)$  sont supposées inconnues et les interconnexions  $\Delta(x_1, \dots, x_N)$  supposées non nulles. De ce fait, l'implantation de la loi de commande (3.51) est difficile voire même impossible. Notre but, consiste alors à approcher les fonctions inconnues par des systèmes flous et de modifier la loi de commande (3.51) afin d'atteindre les objectifs de la commande.

Ces fonctions sont approchées par :

$$\hat{f}_i(x_i, \theta_{fi}) = w_{fi}^T(x_i) \theta_{fi} \quad (3.53)$$

$$\hat{g}_i(x_i, \theta_{gi}) = w_{gi}^T(x_i) \theta_{gi} \quad (3.53)$$

Avec :  $\theta_{fi}$  et  $\theta_{gi}$  sont les vecteurs des paramètres ajustés,  $w_{fi}(x_i)$  et  $w_{gi}(x_i)$  des vecteurs de fonctions floues de basse supposées convenablement choisis par l'utilisateur.

Les fonctions réelles  $f_i(x_i)$  et  $g_i(x_i)$  peuvent être exprimées sous la forme :

$$f_i(x_i) = w_{f_i}^T(x_i) \theta_{f_i}^* + \varepsilon_{f_i}(x_i) \quad (3.54)$$

$$g_i(x_i) = w_{g_i}^T(x_i) \theta_{g_i}^* + \varepsilon_{g_i}(x_i) \quad (3.55)$$

Avec :  $\varepsilon_{f_i}(x_i)$  et  $\varepsilon_{g_i}(x_i)$  sont les erreurs d'approximation,  $\theta_{f_i}^*$  et  $\theta_{g_i}^*$  sont respectivement les paramètres optimaux de  $\theta_{f_i}$  et  $\theta_{g_i}$  minimisant les fonctions  $|\varepsilon_{f_i}(x_i)|$  et  $|\varepsilon_{g_i}(x_i)|$

D'après la propriété d'approximation universelle des systèmes flous utilisés, on peut supposer la bornitude des erreurs d'approximation, i.e.

$$|\varepsilon_{f_i}(x_i)| \leq \bar{\varepsilon}_{f_i}, \quad |\varepsilon_{g_i}(x_i)| \leq \bar{\varepsilon}_{g_i}$$

Où  $\bar{\varepsilon}_{f_i}$  et  $\bar{\varepsilon}_{g_i}$  sont des paramètres positifs inconnus.

De l'analyse ci-dessus, on peut exprimer l'erreur d'identification des fonctions inconnus par

$$f_i(x_i) - \hat{f}_i(x_i, \theta_{f_i}^*) = w_{f_i}^T(x_i) \tilde{\theta}_{f_i} + \varepsilon_{f_i}(x_i) \quad (3.56)$$

$$g_i(x_i) - \hat{g}_i(x_i, \theta_{g_i}^*) = w_{g_i}^T(x_i) \tilde{\theta}_{g_i} + \varepsilon_{g_i}(x_i) \quad (3.57)$$

Avec :  $\tilde{\theta}_{f_i} = \theta_{f_i}^* - \theta_{f_i}$ ,  $\tilde{\theta}_{g_i} = \theta_{g_i}^* - \theta_{g_i}$  sont les erreurs paramétriques.

L'utilisation d'une loi de commande équivalente du type :

$$u_i = \frac{1}{\hat{g}_i(x_i, \theta_{g_i}^*)} (v_i - \hat{f}_i(x_i, \theta_{f_i}^*)) \quad (3.58)$$

Ne peut garantir à elle seule la stabilité du système bouclé. Ceci est dû aux erreurs de reconstruction, à l'existence des interconnexions entre les sous-systèmes, et au fait que cette loi de commande n'est pas définie lorsque :  $\hat{g}_i(x_i, \theta_{g_i}^*) = 0$ .

Pour éviter tous ces problèmes, nous proposons une loi de commande de la forme :

$$u_i = u_{f_i} + u_{r_i} \quad (3.59)$$

Cette loi de commande est la somme de deux termes : un terme de commande adaptatif,  $u_{f_i}$ , qui est une modification de la commande équivalente (3.58), et un terme de commande robuste,  $u_{r_i}$ , introduit pour compenser les effets des erreurs de reconstruction et les effets des interconnexions.

Afin d'avoir une loi de commande bien définie à tout moment, le terme adaptatif flou est défini comme suit :

$$u_{fi} = \frac{\hat{g}_i(x_i, \theta_{gi})}{\varepsilon_0 + \hat{g}_i^2(x_i, \theta_{gi})} \{v_i - \hat{f}_i(x_i, \theta_{fi}) + \beta_i \hat{g}_i(x_i, \theta_{gi}) z_i\} \quad (3.60)$$

Où  $\varepsilon_0$  est une constante positive petite,  $\beta_i$  une constante positive et :

$$z_i = b_i^T P_i e_i$$

Avec :  $P_i$  la solution de l'équation de Lyapunov (3.49).

Le terme de robustification est choisi comme suit :

$$u_{ri} = \frac{1}{\underline{g}_i} (\rho_i (1 + |u_{fi}|) + |u_{0i}|) \text{sign}(z_i) \frac{1}{\underline{g}_i} \alpha_i z_i \quad (3.61)$$

Avec :  $\rho_i$  et  $\alpha_i$  sont des paramètres estimés en ligne par des lois d'adaptation et  $u_{0i}$  est donnée par :

$$u_{0i} = \frac{\varepsilon_0}{\varepsilon_0 + \hat{g}_i^2(x_i, \theta_{gi})} \{v_i - \hat{f}_i(x_i, \theta_{fi}) + \beta_i \hat{g}_i(x_i, \theta_{gi}) z_i\} \quad (3.62)$$

Pour assurer les objectifs de commande, on adopte les lois d'adaptations suivantes :

$$\dot{\theta}_{fi} = -\eta_{fi} w_{fi}(x_i) z_i \quad (3.63)$$

$$\dot{\theta}_{gi} = -\eta_{gi} w_{gi}(x_i) z_i (u_{fi} - \beta_i z_i) \quad (3.64)$$

$$\dot{\rho}_i = \eta_{\rho i} |z_i| (1 + |u_{fi}|) \quad (3.65)$$

$$\dot{\alpha}_i = \eta_{\alpha i} z_i^2 \quad (3.66)$$

Avec :  $\eta_{fi}$ ,  $\eta_{gi}$ ,  $\eta_{\rho i}$  et  $\eta_{\alpha i}$  sont des constantes positives spécifiées par l'utilisateur.

**Théorème 3.3 :** Considérons le système non linéaire interconnecté (3.39). Supposons que les hypothèses 3.8-3.10 sont satisfaites. Alors, la loi de commande définie par les équations (3.60)-(3.62) avec les lois d'adaptation données par les relations (3.63)-(3.66) garantit la bornitude de tous les signaux de la boucle fermée et la convergence vers zéro de toutes les erreurs de poursuite, i.e.  $e_i(t) \rightarrow 0$  quand  $t \rightarrow \infty$  pour  $i = 1, \dots, N$ .

Pour la démonstration voir [21].

**3.3.1.2. Simulations et interprétations :**

Afin de montrer les avantages cette commande par rapport aux autres types étudiés précédemment, les conditions de simulation seront identiques. Aussi, on ne va considérer que la génération de la trajectoire planaire.

Chaque articulation est considérée comme un sous-système. Chaque fonction inconnue est approchée par un système flou à deux entrées  $q_i$  et  $\dot{q}_i$  où on définit pour chaque variable d'entrée deux fonctions d'appartenance :

$$\mu_1(x_{ij}) = \exp\left(-\frac{1}{2}(x_{ij} + 2)^2\right), \mu_2(x_{ij}) = \exp\left(-\frac{1}{2}(x_{ij} + 1)^2\right), \mu_3(x_{ij}) = \exp\left(-\frac{1}{2}(x_{ij} - 0.01)^2\right),$$

$$\mu_4(x_{ij}) = \exp\left(-\frac{1}{2}(x_{ij} - 1)^2\right), \mu_5(x_{ij}) = \exp\left(-\frac{1}{2}(x_{ij} - 2)^2\right), i = j = 1, \dots, 6,$$

Avec :  $x_1 = [q_1 \dot{q}_1], x_2 = [q_2 \dot{q}_2], x_3 = [q_3 \dot{q}_3], x_4 = [q_4 \dot{q}_4], x_5 = [q_5 \dot{q}_5], x_6 = [q_6 \dot{q}_6]$

➤ Les paramètres de synthèses sont choisis comme suit :  $\varepsilon_0 = 10^{-3}, k_i = [2000 \ 63],$

$$Q = 10I_2 \text{ et par conséquent, } P_i = \begin{bmatrix} 1590 & 0.25 \\ 0.25 & 0.8 \end{bmatrix}, \eta_{f_i} = \eta_{g_i} = \eta_{a_i} = \eta_{p_i} = \eta_{u_i} = 10^{-6}.$$

➤ Les conditions initiales sont toutes égales à zéro.

**3.3.1.2.1. Cas idéal : sans erreurs de modélisation :**

Le rayon du cercle est de 0.45m et le temps de simulation est de  $T_e = 14s$

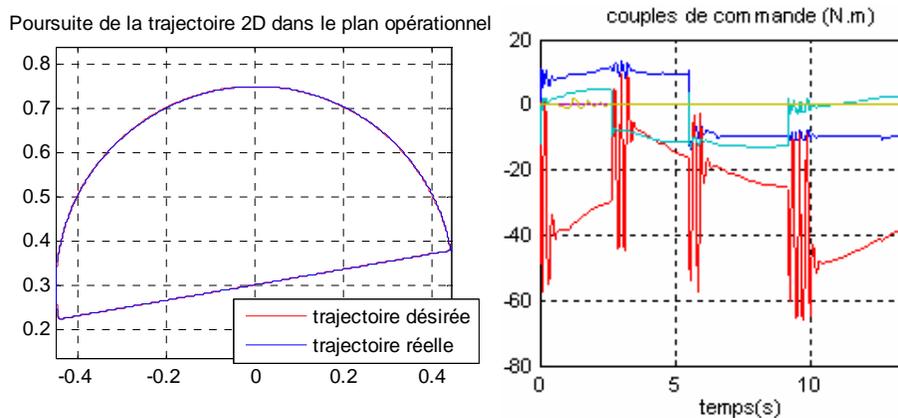


Fig.3.9. Réponse du système et signaux de commandes (couples articulaires) pour la trajectoire planaire avec la commande adaptative décentralisée indirecte appliquée au système nominal

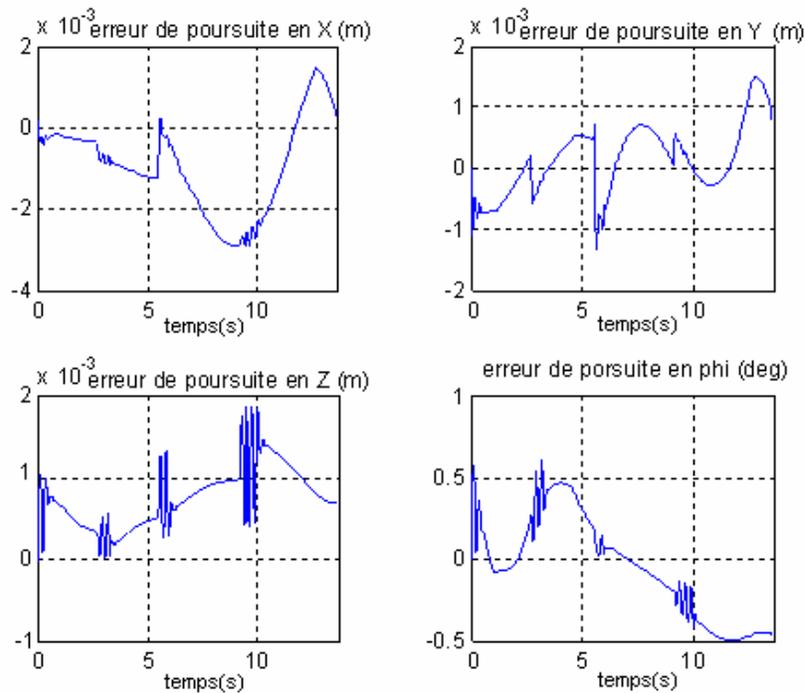


Fig.3.10. Signaux d'erreurs en position et orientation pour la trajectoire planaire avec la commande adaptative décentralisée indirecte appliquée au système nominal.

D'après ces simulations, on parvient à avoir les mêmes performances de la commande adaptative centralisée indirecte, voire une marge d'erreurs identique avec des couples de commandes analogues. Cependant, l'intérêt majeur de ce type de contrôleur est l'approche décentralisation de la commande, qui permet de considérer le système en un ensemble de sous systèmes indépendant et parvenir à réduire aussi bien les erreurs de reconstructions que l'effet des interconnexions.

### 3.3.1.2.2. En présence d'une erreur de modélisation de 50% :

Les résultats sont les suivants :

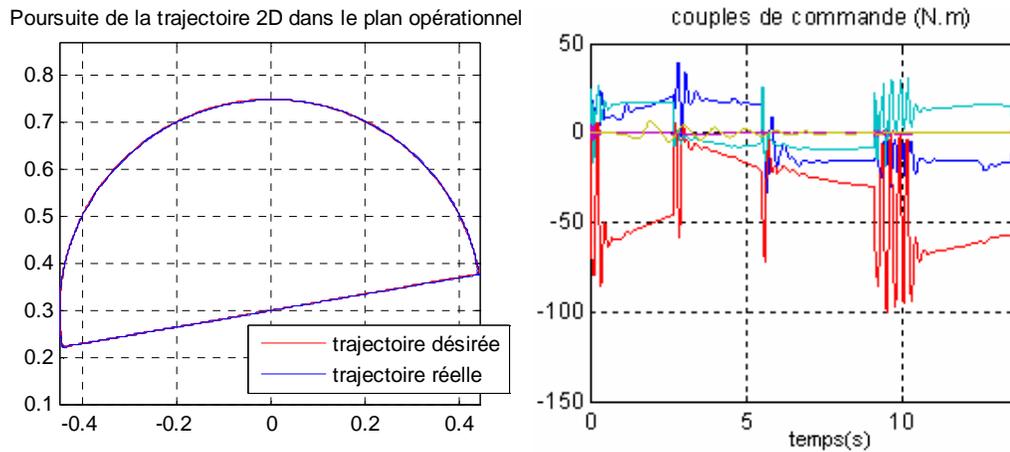


Fig.3.11. Réponse du système et signaux de commande pour la trajectoire plane avec la commande adaptative décentralisée indirecte pour une variation des paramètres de 50%

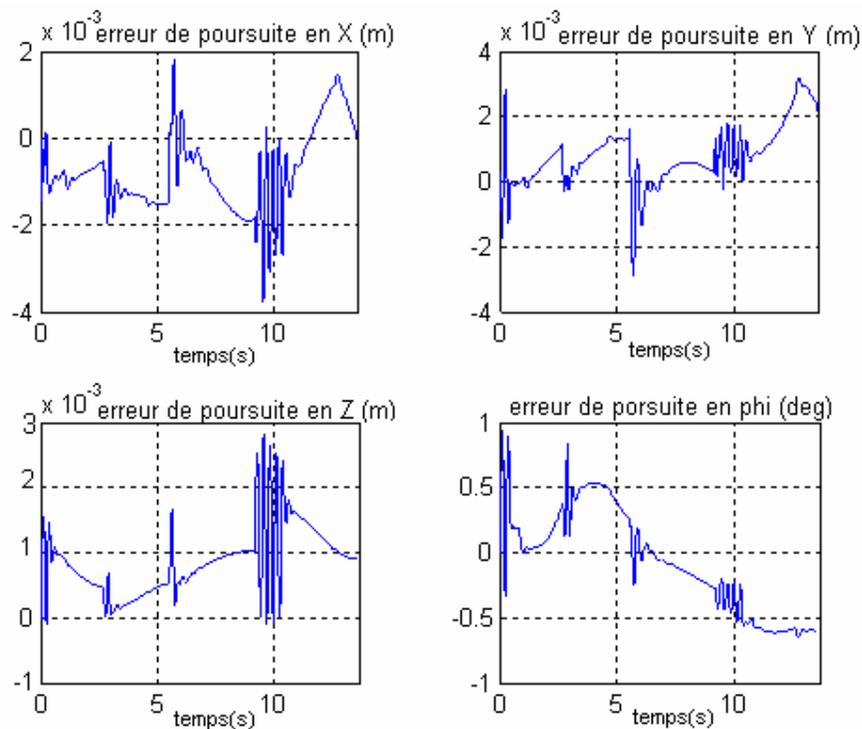


Fig.3.12. Signaux des erreurs de position et d'orientation pour la trajectoire plane avec la Commande adaptative décentralisée indirecte avec une variation des paramètres de 50%.

La commande est robuste vis-à-vis la variation des paramètres. Les marges d'erreurs sont identiques à celles de la commande centralisée, et on parvient toujours à atteindre l'objectif fixé avec une loi de commande plus simple (par rapport à la commande centralisée) ce qui favorise son implantation sur un calculateur adéquat.

### 3.3.2. Commande adaptative décentralisée directe floue :

#### 3.3.2.1. Position du problème :

On considère la classe des systèmes non linéaire interconnectés définie dans (3.39) et les signaux d'erreurs déjà mentionnés dans (3.41).

Pour chaque sous système  $\Sigma_i$ , nous définissons la surface de glissement  $s_i(t)$  suivante :

$$s_i(t) = \left( \frac{d}{dt} + \lambda_i \right)^{n_i-1} e_i(t), \quad \lambda_i > 0 \quad (3.67)$$

L'équation (3.67) peut être écrite sous la forme :

$$s_i(t) = e_i^{(n_i-1)}(t) + k_{i,n_i-1} e_i^{(n_i-2)}(t) + \dots + k_{i,2} \dot{e}_i(t) + k_{i,1} e_i(t) \quad (3.68)$$

Avec :

$$k_{i,j} = c_{n_i-1}^{j-1} \lambda_i^{n_i-j} = \frac{(n_i-1)!}{(n_i-j)! (j-1)!} \lambda_i^{n_i-j}$$

L'équation (3.68) peut s'écrire sous la forme d'état suivante :

$$\dot{e}_i = A_i e_i + b_i s_i \quad (3.69)$$

Avec :  $e_i$ ,  $A_i$  et  $b_i$  définies dans (3.43), (3.47) et (3.48) respectivement.

Comme  $A_i$  est choisi stable, alors 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 :

$$P_i A_i + A_i^T P_i = -Q_i \quad (3.70)$$

La dérivée temporelle de la surface d'erreur (3.68) est donnée par :

$$\dot{s}_i = v_i - f_i(x_i) - g_i(x_i)u_i - D_i(x_1, \dots, x_N) \quad (3.71)$$

Avec :

$$v_i = y_{di}^{(n_i)} + k_{i,n_i-1} e_i^{(n_i-1)} + \dots + k_{i,2} \dot{e}_i \quad (3.72)$$

On admet les hypothèses suivantes :

**Hypothèse 3.11 :** Le gain de commande  $g_i(x_i)$  est non nul, borné et de signe connu. De plus, sa dérivée temporelle est bornée par une fonction positive continue  $D_i(x_i)$ .

**Hypothèse 3.12 :** Chaque trajectoire de référence  $y_{di}(t)$  et ses dérivées jusqu'à l'ordre  $n_i$  sont supposées continues, bornées et connues.

**Hypothèse 3.13 :** Les interconnexions satisfont

$$|\Delta_i(x_1, \dots, x_N)| \leq g_i(x_i) \left\{ \delta_{i,0} + \sum_{j=1}^N \delta_{i,j} (\|e_j\| + |s_j|) \right\} \quad (3.73)$$

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

Si on suppose que les sous-systèmes sont isolés l'un de l'autre, i.e.  $\Delta_i(x_1, \dots, x_N) = \mathbf{0}$ , et que les fonctions non linéaires  $f_i(x_i)$ ,  $g_i(x_i)$  et  $D_i(x_i)$  sont parfaitement connues, alors la loi de commande idéale suivante assure les objectifs de commande [21]:

$$u_i = u_i^* = K_i s_i + \frac{1}{g_i(x_i)} (v_i - f_i(x_i)) + \frac{D_i(x_i)}{2 g_i^2(x_i)} s_i + 2b_i^T P_i e_i \quad (3.74)$$

Mais du fait que, les fonctions  $f_i(x_i)$ ,  $g_i(x_i)$  et  $D_i(x_i)$  sont inconnues et que les interconnexions existent entre les différents sous systèmes, il est clair que la commande idéale (3.74) ne peut être implantée. Dans cette section, nous montrons comment construire un contrôleur adaptatif direct décentralisé en utilisant les systèmes flous. A travers ces systèmes flous, nous allons approximer la loi de commande (3.74).

La loi de commande (3.74) peut être approchée par un système flou de la façon suivante :

$$\hat{u}_i^*(x_i, \theta_i) = K_i s_i + 2 b_i^T P_i e_i + w_i^T(x_i) \theta_i \quad (3.75)$$

Avec :  $\theta_i$  est le vecteur des paramètres ajustés,  $w_i(x_i)$  un vecteur de fonctions floues de base supposé convenablement choisi par l'utilisateur.

La fonction réelle  $u_i^*$  peut être exprimée sous la forme :

$$u_i^*(x_i, \theta_i^*) = K_i s_i + 2 b_i^T P_i e_i + w_i^T(x_i) \theta_i^* + \varepsilon_i(x_i) \quad (3.76)$$

Avec :  $\varepsilon_i(x_i)$  sont les erreurs d'approximation,  $\theta_i^*$  sont les paramètres optimaux de  $\theta_i$  minimisant la fonction  $|\varepsilon_i(x_i)|$ . D'après la propriété d'approximation universelle des systèmes flous utilisés, on peut supposer la bornitude des erreurs d'approximation, i.e.

$$|\varepsilon_i(x_i)| \leq \bar{\varepsilon}_i$$

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

De l'analyse ci-dessus, on peut exprimer l'erreur d'identification de la commande idéale par :

$$u_i^* - \hat{u}_i^*(x_i, \theta_i^*) = w_i^T(x_i) \bar{\theta}_i + \varepsilon_i(x_i) \quad (3.77)$$

Avec :  $\tilde{\theta}_i = \theta_i^T - \hat{\theta}_i$  est le vecteur des erreurs paramétriques.

A partir de l'approximation floue (3.75) de la commande idéale (3.74), nous proposons pour chaque sous système une loi de commande de la forme [21]:

$$u_i = \hat{u}_i^*(x_i, \theta_i) + u_{ri} = K_i s_i + 2 b_i^T P_i e_i + w_i^T(x_i) \theta_i + u_{ri} \quad (3.78)$$

Avec :  $u_{ri}$  est un terme de commande robuste introduit pour compenser les effets des erreurs de reconstruction et les effets des interconnexions. Ce terme est choisi comme suit :

$$u_{ri} = \rho_i \text{sign}(s_i) + \alpha_i s_i \quad (3.79)$$

Avec :  $\rho_i$  et  $\alpha_i$  sont des paramètres estimés en ligne par les lois d'adaptation suivantes :

$$\dot{\theta}_i = \eta_i w_i(x_i) s_i \quad (3.80)$$

$$\dot{\rho}_i = \eta_{\rho_i} |s_i| \quad (3.81)$$

$$\dot{\alpha}_i = \eta_{\alpha_i} s_i^2 \quad (3.82)$$

Avec :  $\eta_i, \eta_{\rho_i}$  et  $\eta_{\alpha_i}$  sont des constantes positives spécifiées par l'utilisateur.

**Théorème 3.4 :** Considérons le système non linéaire interconnecté (3.39). Supposons que les hypothèses 3.11-3.13 sont satisfaites. Alors la loi de commande définie par les équations (3.78) et (3.79) avec les lois d'adaptation données par les équations (3.80)-(3.82) garantit la bornitude de tous les signaux de la boucle fermée et la convergence vers les zéro de toutes les erreurs de poursuites.

Pour la démonstration revoir [21].

### 3.3.2.2. Simulations et interprétations :

Dans cette partie, nous allons appliquer la loi de commande adaptative décentralisée directe au modèle du robot PUMA 560 en gardant les mêmes conditions de simulations déjà mentionnées dans le chapitre précédent. La trajectoire planaire est prise comme exemple de référence à traquer.

- Six systèmes flous sont générés pour les commandes  $u_1, \dots, u_6$  avec deux entrées  $q_i$  et  $\dot{q}_i$  pour chaque système, et pour chaque variable on définit cinq fonctions d'appartenance

$$\mu_1(x_{ij}) = \exp\left(-\frac{1}{2}(x_{ij} + 2)^2\right), \mu_2(x_{ij}) = \exp\left(-\frac{1}{2}(x_{ij} + 1)^2\right), \mu_3(x_{ij}) = \exp\left(-\frac{1}{2}(x_{ij} - 0.01)^2\right),$$

$$\mu_4(x_{ij}) = \exp\left(-\frac{1}{2}(x_{ij} - 1)^2\right), \mu_5(x_{ij}) = \exp\left(-\frac{1}{2}(x_{ij} - 2)^2\right), i = j = 1, \dots, 6,$$

Avec :  $x_1 = [q_1 \dot{q}_1], x_2 = [q_2 \dot{q}_2], x_3 = [q_3 \dot{q}_3], x_4 = [q_4 \dot{q}_4], x_5 = [q_5 \dot{q}_5], x_6 = [q_6 \dot{q}_6]$

- Les conditions initiales sont égales à zéro.
- Les paramètres de synthèse sont choisis comme suit :

$$\lambda_i = 50, \eta_{fi} = 10^{-6}, \eta_{ai} = 10^{-6}, \eta_{pi} = 10^{-6}, K_i = 80, P_i = \begin{bmatrix} 25 & 50 \\ 50 & 50 \end{bmatrix}$$

### 3.3.2.2.1. Cas idéal : sans erreurs de modélisation :

Sur le modèle nominal on applique la loi de commande adaptative décentralisée directe. Les résultats sont les suivants :

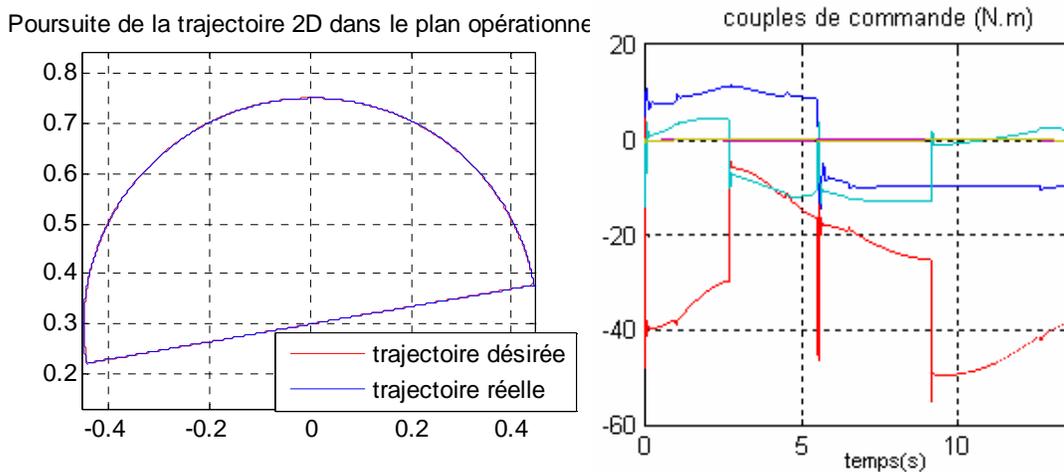


Fig.3.13. Réponse du système et signaux de commande pour la trajectoire planaire avec la commande adaptative décentralisée directe pour le système nominal.

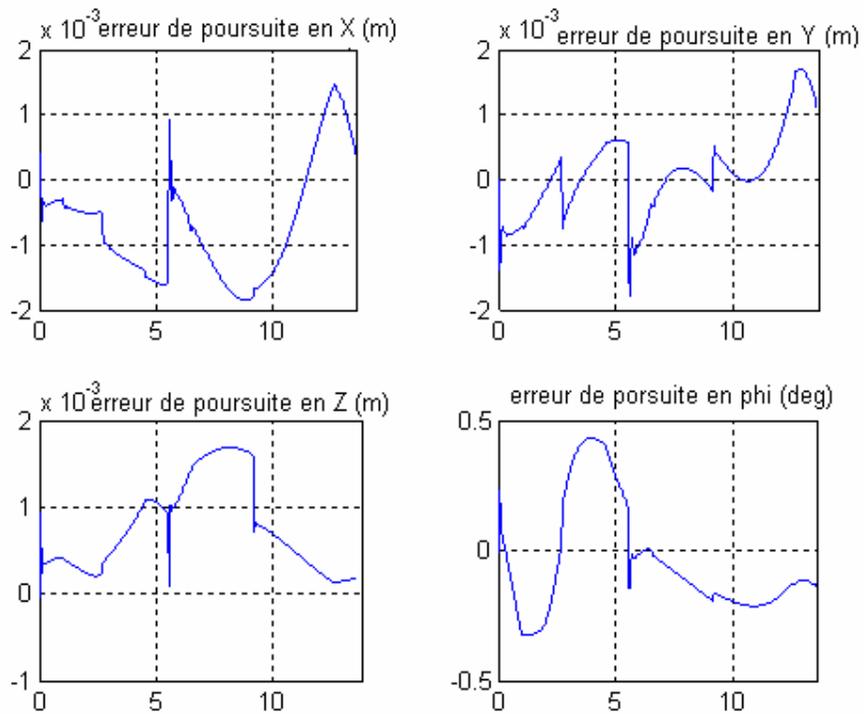


Fig.3.14. Signaux d’erreurs en position et orientation pour la trajectoire planeaire avec la commande adaptative décentralisée directe appliquée au système nominal.

Le contrôleur répond aux recommandations exigées : Parfaites poursuites de la trajectoire avec des erreurs d’ordre inférieur par rapport aussi bien à la commande précédente que par rapport à la même commande mais centralisée.

**3.3.2.2.2. En présence d’une erreur de modélisation de 50% :**

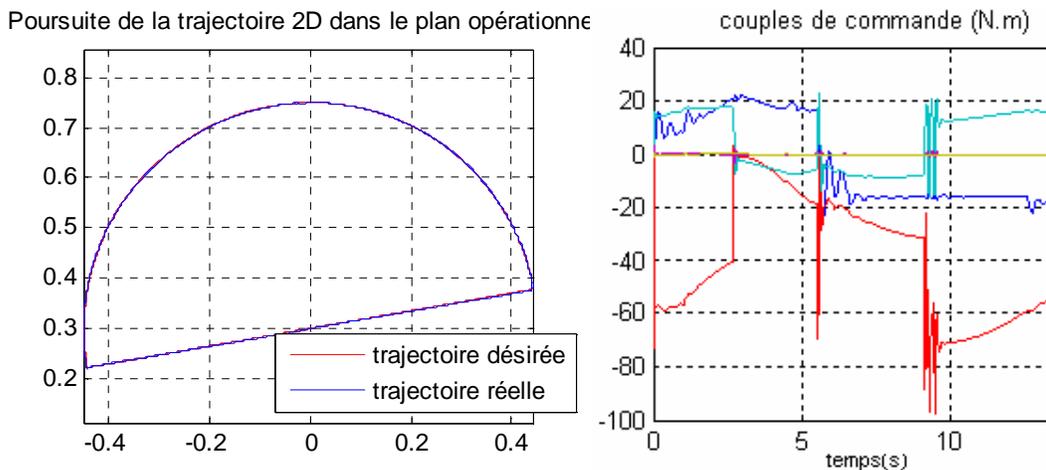


Fig.3.15. Réponse du système et signaux de commande pour la trajectoire planeaire avec la commande adaptative décentralisée directe pour une variation des paramètres de 50%

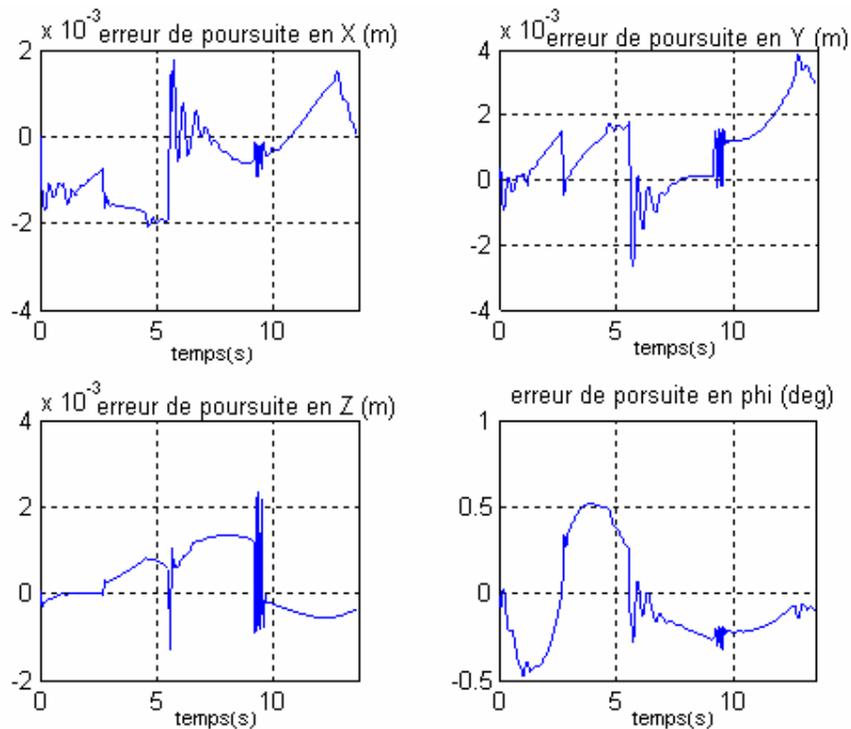


Fig.3.16. Signaux des erreurs de position et d'orientation pour la trajectoire planaire avec la commande adaptative décentralisée directe avec une variation des paramètres de 50%.

En dépit de la variation des paramètres, en appliquant cette loi de commande, l'outil de l'organe terminal parvient à exécuter la tâche imposée avec une bonne précision. La marge d'erreur est la même et les commandes articulaires sont admissibles.

### Conclusion :

Dans ce chapitre, nous avons développé la commande adaptative floue centralisée et décentralisée, directe et indirecte pour notre système non linéaire MIMO qui est le robot PUMA 560.

Dans un premier temps, nous avons opté à commander le système dans sa globalité, en supposant a priori aucune information disponible concernant le modèle du système. La propriété d'approximateurs universels dont se réjouissent les systèmes flous nous a permis dans l'approche indirecte d'approximer les fonctions du système nécessaires dans la loi de commande, et d'approximer la loi de commande (elle même) dans l'approche directe.

On est passé par la suite à supposer qu'on a la possibilité de décomposer le système en un ensemble de sous système qu'on rendra indépendant en appliquant la loi de commande décentralisée. Cette loi de commande, considère que chaque articulation est commandée indépendamment des autres et grâce aux systèmes flous nous avons pu éliminer aussi bien les erreurs d'approximation (de reconstruction) que les influences mutuelles des interconnexions.

Des simulations ont montré la robustesse de ce type de commande (adaptative floue) et en comparant les différents résultats, on peut conclure d'une part que, l'approche directe est meilleure que celle indirecte vu qu'il minimise la marge d'erreur tout en nécessitant un temps de simulation plus réduit. Cela revient à l'absence de la phase d'initialisation que nécessite l'approche indirecte. D'autre part, les deux types de commande floue centralisée et décentralisée donnent des résultats similaires mais l'avantage principal de la commande décentralisée est sa simplicité et son temps d'exécution minime.

## CONCLUSION GENERALE

Bien que la commande de la plus parts des robots industriels actuels soit encore conçue à partir des théories de l'automatique linéaire, des méthodes plus avancées tenant compte du caractère non linéaire interconnecté des structures articulées doivent être envisagée pour les applications exigeant de grandes performances dynamique telles que la précision et la rapidité.

Dans cette perspective, l'objectif de notre mémoire était l'application d'une nouvelle loi de commande sophistiqué sur le bras de robot manipulateur PUMA 560 en vue de l'amener à exécuter une tâche déterminée dans l'espace libre. La loi de commande adoptée est : la commande adaptative floue centralisée et décentralisée avec ses deux approches (directe et indirecte).

Pour atteindre notre objectif, on été amené à introduire quelques commandes à titre de comparaison et afin de démontrer la suite logique de la loi de commande choisie.

Dans le premier chapitre, on s'est intéressé à la commande par découplage non linéaire. Cette loi de commande découle de l'application des principes de la géométrie différentielle, et a connu un grand succès. C'est la base sur laquelle se sont développées plusieurs approches de commande. Son principe est la linéarisation et le découplage des équations du modèle.

Aussi dans ce chapitre, le modèle du système et la tâche à accomplir ont y été introduits. Des simulations sur SIMULINK ont prouvé les performances de cette commande appliquée au modèle nominal, mais une variation des paramètres les détériorât. Ceci est prévisible vue que, l'inconvénient majeur de cette loi de commande est son hypothèse d'une parfaite connaissance du modèle.

Ainsi, dans le deuxième chapitre, une nouvelle loi plus robuste a été introduite : il s'agit de la loi de commande adaptative. Parmi les différentes stratégies qui existent notre choix s'est fixé sur deux d'entre elles : la commande par découplage non linéaire adaptative et la commande adaptative passive. Les résultats de simulation ont donnent l'avantage à la commande par découplage non linéaire adaptative.

Toute fois, ces méthodes supposent une connaissance de la dynamique du système, chose qu'on ne peut admettre pour un système complexe tel que le robot, pour qui, l'élaboration du modèle nécessitait plusieurs hypothèses et négligeait plusieurs phénomènes.

Une solution à ce problème peut être envisagée en utilisant les systèmes flous pour l'approximation des dynamiques du robot. Ainsi, le troisième chapitre, traitât la commande adaptative floue centralisée et décentralisée.

Dans ces techniques de commande, les systèmes flous ont été utilisés pour l'approximation des dynamiques inconnues du système dans l'approche indirect, et pour l'approximation des lois de commandes stabilisantes dans l'approche direct. Le problème des erreurs de reconstruction, inhérent à ce type de stratégies est traité par l'utilisation des modes de glissements comme termes robustes dans la loi de commande de base.

Dans l'approche centralisée le système est considéré en entier, l'avantage est de parfaites performances, cependant, un temps d'exécution important. C'est pourquoi on est passé à la commande décentralisée dont la programmation est plus simple tout en respectant les contraintes imposées.

Pour toutes les lois de commande élaborées, l'étude de la stabilité du système bouclé était faite par l'approche de Lyapunov.

Concernant les perspectives futures de ce travail on propose :

- D'appliquer les principes étudiés sur un robot flexible ;
- D'utiliser d'autres types d'approximateurs universels tels que : les réseaux de neurones, les fonctions radiales de base et les ondelettes,
- D'utiliser les nouveaux algorithmes d'optimisation tels que : les algorithmes génétiques et le recuit simulé pour la planification en ligne des trajectoires.
- Et pour la partie programmation on propose de développer un logiciel de simulation qui soit identique à la réalité en utilisant par exemple la toolbox SimMechanics qu'offre le logiciel SIMULINK, et la réalité virtuelle que fournit le logiciel SolidWorks.

## Annexe A

### Paramètres du Robot PUMA 560

#### 1. Buté du robot PUMA 560 :

Les butées du Robot PUMA 560 selon [5] sont représentées dans le tableau A.1

	$q_i$ min (deg)	$q_i$ max (deg)
$q_1$	-170	170
$q_2$	-225	45
$q_3$	-70	255
$q_4$	-135	135
$q_5$	-100	100
$q_6$	-180	180

Tableau A.1. Butées du Robot PUMA 560

#### 2. Paramètres inertiels du robot PUMA 560 :

Dans ce qui suit, nous allons donner dans des tableaux les différents paramètres inertiels du robot PUMA 560. Pour chaque corps on donne : sa masse, les coordonnées de son centre de gravité par rapport au repère lié à sa liaison, les principaux moments d'inertie et les couples articulaires maximaux.

	Corps 2	Corps 3	Corps 4	Corps 5	Corps 6
Masse (kg)	17.4	4.8	0.82	0.34	0.09

Tableau A.2. Masse de chaque corps constituant le robot PUMA 560

	$r_x$ (m)	$r_y$ (m)	$r_z$ (m)
Corps 2	0.068	0.006	-0.016
Corps 3	0	-0.07	0.014
Corps 4	0	0	-0.019
Corps 5	0	0	0
Corps 6	0	0	0.032

Tableau A.3. Coordonnées du centre de gravité de chaque corps du robot PUMA 560

	$I_{xx}$ (kg.m <sup>2</sup> )	$I_{yy}$ (kg.m <sup>2</sup> )	$I_{zz}$ (kg.m <sup>2</sup> )	$I_{moteur}$ (kg.m <sup>2</sup> )
Corps1	-	-	0.35	1.14
Corps 2	0.13	0.524	0.539	4.71
Corps 3	0.066	0.0125	0.086	0.83
Corps 4	0.0018	0.0018	0.0013	0.2
Corps 5	0.0003	0.0003	0.0004	0.179
Corps 6	0.00015	0.00015	0.00004	0.193

Tableau A.4. Les principaux moments d'inertie de chaque corps du robot PUMA 560

	Liaison 1	Liaison 2	Liaison 3	Liaison 4	Liaison 5	Liaison 6
Couple maximal(N.m)	97.6	186.4	89.4	24.2	20.1	21.3

Tableau A.5. Couple articulaire maximal pour chaque liaison du robot PUMA 560

## ANNEXE B

### Systèmes flous et notions de la stabilité des systèmes

Dans cette partie, nous présentons les notions de base de la logique floue et les outils mathématiques aux quels on a fait appel.

#### I. Systèmes flous :

##### 1. Ensembles flous

En logique floue, un ensemble flou est la généralisation de la notion de l'ensemble en logique classique (binaire).

Mathématiquement, un ensemble flou  $F$  est défini sur un référentiel (un univers de discours)  $X$  et par une fonction d'appartenance  $\mu_F(x)$  qui prend des valeurs dans l'intervalle  $[0,1]$ . Il peut être représenté comme un ensemble de paires ordonnées :

$$F = \{(x, \mu_F(x) \mid x \in X\}$$

La fonction d'appartenance  $\mu_F(x)$  quantifie le degré d'appartenance de chaque élément de  $X$  à  $F$  :

$$\mu_F(x) = \begin{cases} \in ]0,1] & \text{si } x \in F \\ 0 & \text{sinon} \end{cases}$$

##### I.1.1. Fonction d'appartenance:

Les fonctions d'appartenances utilisées pour décrire les ensembles flous possèdent quatre caractéristiques principales :

- 1) **Le type** : Elles peuvent être à titre d'exemple triangulaires, trapézoïdales, gaussiennes ou sigmoïdales, etc.

➤ **Fonction triangulaire** : Elle est définie par trois paramètres (a, b, c) et donnée par :

$$\mu_F(x) = \max \left[ \min \left( \frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right]$$

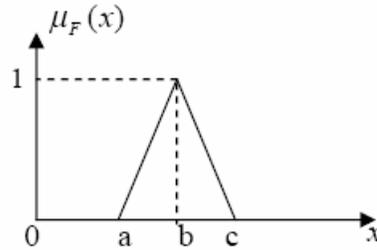


Fig.1. Fonction triangulaire

➤ **Fonction trapézoïdale** : Elle est définie par quatre paramètres (a, b, c, d) et donnée par :

$$\mu_F(x) = \max \left[ \min \left( \frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right]$$

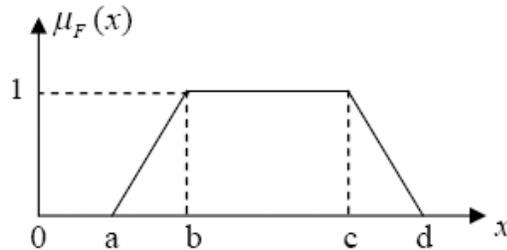


Fig.2. Fonction trapézoïdale

➤ **Fonction gaussienne** : Elle est définie par deux paramètres ( $\sigma, m$ ) :

$$\mu_F(x) = \exp \left( -\frac{1}{2} \left( \frac{x-m}{\sigma} \right)^2 \right)$$

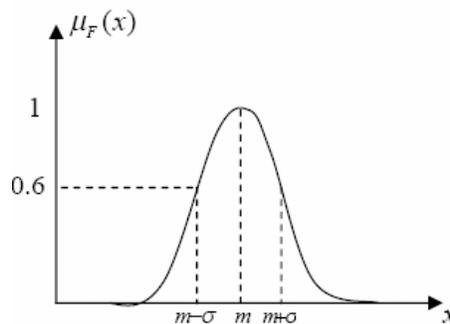


Fig.3. Fonction gaussienne

- **Fonction sigmoïdale** : Elle est définie par deux paramètres (  $a$ ,  $c$  ) :

$$\mu_F(x) = \frac{1}{1 + \exp(-a(x - c))}$$

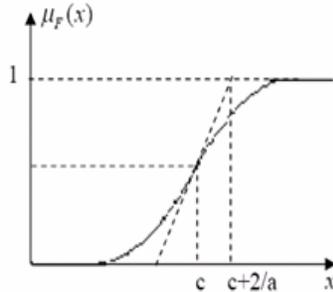


Fig.4. Fonction sigmoïdale

Il y a également deux ensembles flous spéciaux :

- **Un ensemble flou singleton** : il est équivalent à un nombre flou et défini par :

$$\mu_F(x) = \begin{cases} 1 & \text{si } x = x_0 \\ 0 & \text{ailleurs} \end{cases}$$

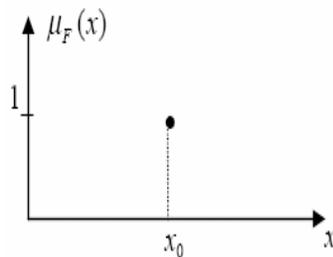


Fig.5. Fonction singleton

- **Un ensemble flou universel** : il est défini par :

$$\mu_F(x) = 1 \quad \forall x \in F$$

i.e. tous les degrés d'appartenance sont égaux à 1.

**2) Le support** : Il définit l'ensemble des éléments de  $X$  qui appartiennent au moins un peu à  $F$ . Il est noté  $S(F)$ . Autrement dit, c'est la partie de  $X$  sur laquelle le degré d'appartenance à  $F$  n'est pas nul:

$$S(F) = \{x \in X, \mu_F(x) \neq 0\}$$

3) **Le noyau** : L'ensemble des éléments appartenant de façon absolue à  $F$  (i.e. ayant un degré 1) est appelé le noyau de  $F$ , noté par  $N(F)$  :

$$N(F) = \{x \in X, \mu_F(x) = 1\}$$

Si  $F$  est un ensemble classique sur  $X$ , il donc est normalisé et identique à son support et à son noyau.

4) **La hauteur** : notée  $H(F)$ , c'est le plus fort degré avec lequel un élément de  $X$  appartient à  $F$ . Autrement dit, c'est la valeur maximale prise par sa fonction d'appartenance:

$$H(F) = \sup_{x \in X} \mu_F(x)$$

On dit que  $F$  est normalisé si sa hauteur  $H(F)$  est égale à 1.

### I.1.2. Formalisme Linguistique :

Dans le formalisme défini par Zadeh, les propriétés définies par les ensembles flous se nomment les termes (labels) linguistiques ou les termes flous. Chaque terme représente un sous-ensemble de valeurs numériques et caractérise ainsi la variable floue (ou linguistique). Le domaine sur lequel ces termes et ces variables sont définis constitue *l'univers de discours*. Le découpage de cet univers de discours par les termes linguistiques est appelé *une partition floue*. Lorsque la somme des degrés d'appartenance sur la totalité de l'univers de discours est égale à 1, nous parlons alors de *partition floue forte*.

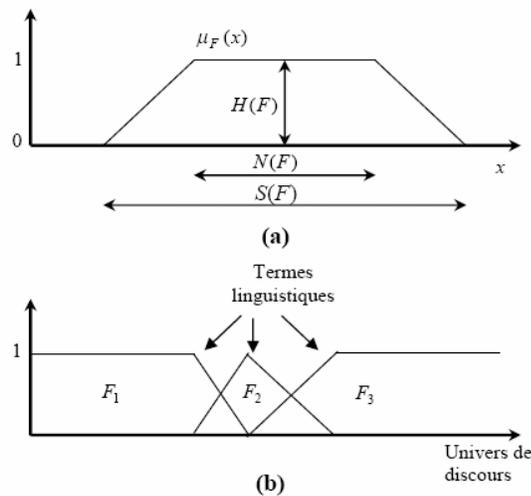


Fig.6. Caractéristiques d'un ensemble flou et partition floue forte.

### I.1.3 Opérations de base sur les ensembles flous :

La théorie mathématique sur les ensembles flous définit de nombreux opérateurs sur ces ensembles et sur les fonctions d'appartenance. Soient  $F_1$  et  $F_2$  deux ensembles flous définis sur l'ensemble  $X$  :

- **Complément** : Le complément de  $F_1$  noté  $\bar{F}_1$  est défini par :

$$\forall x \in X, \mu_{\bar{F}_1}(x) = 1 - \mu_{F_1}(x)$$

- **Intersection** : Elle est réalisée par n'importe quelle T-norme (norme triangulaire notée T). La plus ordinaire est l'opérateur min :

$$\forall x \in X, \mu_{F_1 \cap F_2}(x) = T(\mu_{F_1}(x), \mu_{F_2}(x)) = \min(\mu_{F_1}(x), \mu_{F_2}(x))$$

- **Union** : Elle est réalisée par n'importe quelle T-conorme (conorme triangulaire notée T). La plus courante est l'opérateur max :

$$\forall x \in X, \mu_{F_1 \cup F_2}(x) = \perp(\mu_{F_1}(x), \mu_{F_2}(x)) = \max(\mu_{F_1}(x), \mu_{F_2}(x))$$

- **Cardinalité** : Elle évalue le degré global avec lequel les éléments de  $X$  appartiennent à  $F_1$ . Elle est définie par :

$$|F| = \sum_{x \in X} \mu_F(x)$$

La cardinalité d'un ensemble classique est égale au nombre d'éléments qui le composent.

### I.1.4. Interprétation des ensembles flous :

Il existe deux interprétations possibles des sous-ensembles flous. La première concerne le passage du numérique au symbolique (Fuzzification) et la seconde le passage du symbolique au numérique (Défuzzification) [19].

## I.2. Règles floues :

Les règles floues, appelées aussi règles d'inférence, sont de la forme "Si *prémisses* Alors *conclusions*". Elles sont très proches du langage naturel.

Plus généralement, un système d'inférence floue est composé de  $N$  règles floues de type :

**$R_i$  : Si  $x \in X_1^i$  et ... et  $x \in X_n^i$  alors  $y$  est  $Y^i$**

Avec :

- "**Si  $x \in X_1^i$  et ... et  $x \in X_n^i$** " la partie prémisses de la règle «  $i$  »
- " **$y$  est  $Y^i$** " la conclusion (conséquence) associée
- $X_j^i$  une caractéristique floue de la variable  $x_j$  dont la fonction d'appartenance est notée  $\mu_{X_j^i}$ .

Les prémisses et les conclusions constituent des propositions floues exprimées par " $x$  est  $X$ " ou par une conjonction ou disjonction de prédicats de cette forme. Un système d'inférence flou est alors une collection de relations entre propositions floues, ayant comme particularité la déduction de conclusions même quand les prédicats en prémisses ne sont qu'en partie satisfaits. La règle d'inférence utilisée en logique floue est la généralisation du Modus Ponens (MP) qui prend en considération l'aspect graduel inhérent aux ensembles flous.

La conclusion de chaque règle  $i$  est obtenue en déterminant la fonction d'appartenance  $\mu_{Y^i}$ .

Ce calcul est réalisé à partir de la règle  $R_i$  et de  $\mu_{X_j^i}$ .

La conclusion finale est formée par l'agrégation des conclusions de chaque règle. Pour le calcul de cette conclusion, quatre étapes sont nécessaires :

1. calcul des propositions,
2. calcul des relations,
3. compositions des règles avec les faits observés, et
4. agrégations des conclusions des règles.

Une cinquième opération appelée la défuzzification est indispensable, si le résultat souhaité est non flou.

### **I.2.1 Calcul des propositions floues :**

Le calcul d'une proposition floue consiste à déterminer sa fonction caractéristique  $\mu_P$ .

Lorsque la proposition est composée de conjonctions ou de disjonctions de plusieurs prédicats, il est indispensable de définir les opérateurs correspondants. Nous notons trois implémentations principales pour ces opérateurs logiques.

Soient  $X_1^i$  et  $X_2^i$  deux termes flous opérant sur les univers de discours  $X_1$  et  $X_2$  et qui :

composent la proposition  $P$  :

➤ la conjonction :  $P = "x_1 \text{ est } X_1^i \text{ et } x_2 \text{ est } X_2^i"$  est définie par :  $\forall x_1 \in X_1 \text{ et } x_2 \in X_2 :$

$$\mu_P = \text{ET}(X_1^i, X_2^i) = \begin{cases} \min(\mu_{X_1^i}(x_1), \mu_{X_2^i}(x_2)), & \text{Mamdani} \\ \mu_{X_1^i}(x_1) \times \mu_{X_2^i}(x_2), & \text{Larsen} \\ \max(0, \mu_{X_1^i}(x_1) + \mu_{X_2^i}(x_2) - 1) & \text{Lukasiewicz} \end{cases}$$

➤ la disjonction :  $P = "x_1 \text{ ou } X_1^i \text{ et } x_2 \text{ est } X_2^i"$  est définie par :  $\forall x_1 \in X_1 \text{ et } x_2 \in X_2 :$

$$\mu_P = \text{OU}(X_1^i, X_2^i) = \begin{cases} \max(\mu_{X_1^i}(x_1), \mu_{X_2^i}(x_2)), & \text{Mamdani} \\ \mu_{X_1^i}(x_1) + \mu_{X_2^i}(x_2) - (\mu_{X_1^i}(x_1) \cdot \mu_{X_2^i}(x_2)), & \text{Larsen} \\ \min(1, \mu_{X_1^i}(x_1) + \mu_{X_2^i}(x_2)) & \text{Lukasiewicz} \end{cases}$$

Après le calcul de toutes les propositions floues, l'étape suivante consiste à calculer la contribution de chaque règle afin de pouvoir, dans un troisième temps, composer la proposition correspondante aux faits observés avec chaque règle.

## I.2.2 Calcul des relations :

Une règle floue  $R$  définit une relation entre  $X$  et  $Y$ . Cette relation constitue une distribution conditionnelle qui associe à chaque  $x \in X$  l'ensemble flou  $R(x)$  des images possibles de  $x$  dans  $Y$  et est caractérisée par sa fonction d'appartenance  $\mu_{R(x,y)}$ . Elle peut être modélisée selon trois classes de fonctions d'implication :

$$\mu_{R(x,y)} = I(\mu_X(x), \mu_Y(y)) \text{ avec } I: [0,1] \times [0,1] \rightarrow [0,1]$$

- **S-implication** :  $I_s(u, v) = S(n(u), v)$  où  $S$  est une T-conorme et  $n$  une fonction de négation définie sur  $[0,1]$  ( $n(u) = 1 - u$  par exemple).
- **R-implication** : Ces implications proviennent du formalisme de la logique intuitionniste.
- **Implication T-norme** : Bien que ce modèle ne vérifie pas toutes les propriétés de l'implication, il est possible de l'utiliser tout en prenant garde au choix de l'opérateur d'agrégation.

### I.2.3 Inférence floue :

L'inférence floue permet de combiner la connaissance que nous avons de la règle  $R$ , par exemple  $\mu_R(x,y), \forall x \in X, \forall y \in Y$  avec la donnée concernant la variable  $x$ . Cette composition définie par le MPG (Modus Ponens Généralisé) est la règle compositionnelle d'inférence sup-T, où T est la norme triangulaire. La fonction d'appartenance ainsi inférée est donnée par :

$$\forall y \in Y, \mu_{Y'}(y) = \sup_{x \in X} T(\mu_{X'}(x), \mu_R(x, y))$$

La norme triangulaire doit être choisie de telle sorte que le MPG soit compatible avec le Modus Ponens classique.

Après avoir déduit les conclusions des règles du système d'inférence floue, l'étape suivante consiste à les agréger pour calculer la fonction d'appartenance finale.

### I.2.4 Agrégation des règles :

Après le calcul des conclusions intermédiaires fournies par chaque règle de la base du système d'inférence floue, la conclusion finale est le résultat de la combinaison de toutes ces conséquences locales. Elle peut être effectuée de trois manières :

**1- Par intersection :** l'ensemble flou final  $Y'$  est défini comme étant l'intersection de tous les ensembles inférés.

**2- Par union :** l'ensemble flou final  $Y'$  est issu de l'union des ensembles flous locaux.

**3- Par certaines méthodes de défuzzification, ou par le calcul du barycentre** des conclusions inférées (quand celles-ci sont de nature précise, comme dans le cas du système flou de Takagi-Sugeno).

### I.2.5 Défuzzification :

Le résultat de l'agrégation est généralement un ensemble flou. Or, si l'on désire une sortie non floue, il faut transformer cette partie floue issue de l'inférence en une valeur physique.

Cette opération est appelée défuzzification. Plusieurs stratégies de défuzzification existent.

Les plus répandues sont :

- la méthode du centre de gravité,
- la méthode du maximum,
- la méthode de la hauteur, et
- la méthode de l'aire, ...etc.

### I.3. Modèles de TS :

Un modèle flou (système flou) est une représentation du comportement du système par les ensembles flous. Il peut remplacer un système d'équations différentielles ou une fonction non linéaire analytique utilisées dans la modélisation classique. D'un point de vue structurel, les modèles flous peuvent être classés en deux classes essentielles :

- les modèles linguistiques (les modèles de Mamdani),
- les modèles à conséquences fonctionnelles (les modèles de Takagi-Sugeno (TS)).

Vu que le type de modèle utilisé dans ce travail est celui de Takagi-Sugeno, nous allons donner ci-dessous les principes de ce type.

Les modèles linguistiques ne permettent pas d'intégrer directement des connaissances objectives sur le système qui ne sont pas exprimées par des ensembles flous. Ainsi, des connaissances exprimées sous forme analytique décrivant la structure physique du système, par exemple les équations de conservation d'énergie ou du bilan de masses ne peuvent pas être prises en compte directement. Sugeno et ses collaborateurs ont proposé une classe de modèles permettant de représenter ce type d'information. Les modèles de cette classe sont appelés *les modèles de Takagi-Sugeno (TS)*.

La base de  $N$  règles pour un vecteur d'entrée  $x$  de dimension  $n$  est de la forme :

**Si  $x \in X_1^i$  et ... et  $x \in X_n^i$  alors  $y = F_i(x)$ ,  $i = 1, \dots, N$**

Où  $F_i(x)$  peut être une fonction analytique non linéaire quelconque.

Si  $F_i(x)$  est une fonction linéaire, dans ce cas le modèle est de type TS d'ordre 1 (TS1) :

$$F_i(x) = \sum_{j=1}^n a_j^i x_j + b^i$$

Lorsque  $F_i(x) = b_i$  le modèle est de type TS d'ordre zéro (TS0).

Chacune des règles représente un modèle local sur une région floue d'entrée ou sur un sous-espace d'entrée. Dans chaque région, le modèle flou est défini par la fonction  $F_i$  qui relie les entrées à la sortie numérique. Le modèle global est constitué par interpolation des modèles locaux.

L'inférence floue, dans ce modèle, est composée de trois étapes:

- 1- calcul des degrés d'activation de chaque règle avec l'opération de conjonction implémentée par le

$$w_i(x) = \prod_{j=1}^n \mu_{X_j^i}(x_j) \quad (i = 1, \dots, N)$$

2- calcul des sorties individuelles : la sortie de chaque règle  $i$  est donnée par sa conséquence, c'est à dire:

$$y^i = F_i(x)$$

3- agrégation des sorties individuelles : la valeur finale de la sortie résultante de l'ensemble des règles est donnée par la moyenne des sorties individuelles pondérées par le degré d'activation des règles, soit:

$$y = \frac{\sum_{i=1}^N w_i(x) F_i(x)}{\sum_{i=1}^N w_i(x)} \quad (1)$$

Les avantages intrinsèques au système flou de TS1 sont les suivants :

- Ils permettent de décrire des situations où la structure physique du système est bien connue, par exemple, une fonction linéaire exprimant la relation entrée-sortie.
- Ils permettent d'approcher le comportement d'un système complexe par une collection des modèles locaux linéaires.
- Les modèles TS ont une capacité de représentation importante. Le nombre de règles nécessaires pour approcher un système à certain degré de précision est généralement plus faible que dans le cas des modèles linguistique. L'identification du modèle à partir de données (dans le cas hors ligne) est facile et nécessite moins de données que pour les modèles linguistiques.
- La relation (1) relie directement la variable de sortie aux variables d'entrée. Ceci permet une application facile des techniques d'optimisation et d'adaptation pour identifier les différents paramètres du modèle.

Toutefois, ces modèles ont aussi des inconvénients :

- De part la nature fonctionnelle des conclusions, les règles n'ont pas de signification sémantique claire comme dans le cas des modèles linguistiques.
- L'interpolation entre les modèles locaux peut engendrer une erreur importante qui réduit la précision globale du modèle.
- L'intégration de la connaissance humaine formulée par des règles linguistiques n'est pas aisée.

Tout au long de ce mémoire, comme la majorité des travaux relatifs à la commande adaptative floue, c'est le système flou de TS0 qui sera utilisé pour estimer la commande idéale ou les fonctions non linéaires inconnues.

En introduisant la notion des fonctions floues de base [WAN94], la sortie du système flou de TS0 peut être écrite comme suit :

$$y(x) = \theta^T \Psi(x)$$

Avec :  $\theta = [b^1, \dots, b^N]^T$  est le vecteur des paramètres de la partie conclusion, et  $\Psi(x) = [v_1(x), \dots, v_N(x)]^T$  est le vecteur des fonctions floues de base, dont chacune des composantes est donnée par :

$$v_i = \frac{w_i(x)}{\sum_{i=1}^N w_i(x)}$$

Il est important de noter que si nous choisissons une partition floue forte pour les variables d'entrée, le vecteur des fonctions floues de base se réduit  $v_i = w_i(x)$ . Dans le cas où la partition floue forte n'est pas choisie, nous devons assurer tout le temps que  $\sum_{i=1}^N w_i(x) \neq 0$  i.e. que le système flou est toujours bien défini.

#### I.4. Théorème des approximateurs universels :

Deux raisons principales amènent à incorporer les systèmes flous dans un contrôleur adaptatif :

- Ils ont la propriété d'approximation universelle,
- La possibilité d'intégrer dans les contrôleurs des informations linguistiques ou mathématiques disponibles issues d'une expertise.

Dans la littérature floue, on dispose d'un nombre important de travaux montrant que les systèmes flous sont bien des approximateurs universels [14] [21].

Les algorithmes de commande floue proposés dans ce mémoire sont basés sur le théorème suivant dit *théorème d'approximateurs universels*.

**Théorème 1.1 :** Soit  $f(x)$  une fonction non linéaire continue définie sur un ensemble compact  $\Omega_x$  et pour toute constante positive  $\varepsilon$ , il existe un système flou  $y(x)$  de la forme :

$$\sup_{x \in \Omega_x} |f(x) - \theta^T \Psi(x)| < \varepsilon$$

## II. Rappel sur la stabilité :

Les définitions et les résultats donnés dans cette section sont particulièrement utiles dans l'analyse des schémas de commande adaptative.

## II.1 Définitions et Stabilité de Lyapunov

La notion de stabilité d'un système dynamique caractérise le comportement de ses trajectoires autour des points d'équilibre. L'analyse de la stabilité d'un système dynamique permet donc d'étudier l'évolution de sa trajectoire d'état lorsque l'état initial est proche d'un point d'équilibre. La stabilité au sens de Lyapunov est une théorie générale valable pour toute équation différentielle. Cette notion signifie que la solution d'une équation différentielle initialisée au voisinage d'un point d'équilibre en reste suffisamment proche.

Considérons la classe des systèmes non linéaires décrits par l'équation dynamique:

$$\dot{x} = f(x, t), x(t_0) = x_0 \quad (2.1)$$

Où  $x \in R^n$  et  $f: R^n \times R^+ \rightarrow R^n$  continue. On désigne par  $x_e$  un point d'équilibre de (2.1), tel que  $(x_e, t) = 0, \forall t \geq t_0$ , et par  $x(t, t_0, x_0)$  la solution à l'instant  $t \geq t_0$  du système (2.1), initialisée en  $x_0$  à l'instant  $t_0$ .

Nous supposons que le système (2.1) possède un unique point d'équilibre  $x_e$ . Ceci nous mène à présenter les définitions de la stabilité du système (2.1) autour de l'origine.

**Définition 1.1 (Stabilité):** L'origine est un point d'équilibre stable au sens de Lyapunov pour (2.1) si  $\forall \varepsilon > 0, \forall t \geq t_0$ , il existe un scalaire positif  $\delta(\varepsilon, t_0)$  tel que :

$$\|x_0\| \leq \delta(\varepsilon, t_0) \Rightarrow |x(t, t_0, x_0)| \leq \varepsilon$$

On dit que l'origine est instable dans le cas contraire.

**Définition 1.2 (Stabilité uniforme) :** L'origine est un point d'équilibre uniformément stable pour (2.1) si  $\forall \varepsilon > 0$  il existe un scalaire positif  $\delta(\varepsilon)$  tel que :

$$\|x_0\| \leq \delta(\varepsilon) \Rightarrow |x(t, t_0, x_0)| \leq \varepsilon$$

**Définition 1.3 (Stabilité UUB) :** Les solutions de (2.1) sont dites *uniformément ultimes bornées* (Uniformly Ultimately Bounded UUB) s'il existe des constantes  $b$  et  $c$  telles que, pour tout  $\alpha \in [0, c]$ , il existe un temps positif  $T = T(\alpha)$  (indépendant de  $t_0$ ) tel que, pour tout  $x_0 \in B_\alpha$  et  $\forall t_0 \geq 0$  on a :

$$\|x_0\| \leq \alpha \Rightarrow \|x(t, t_0, x_0)\| \leq b \quad \forall t \geq t_0 + T$$

Lorsque ceci est vrai pour toute constante positive  $\alpha$ , les solutions sont dites *globalement uniformément ultimes bornées*.

**Définition 1.4 (Attractivité) :** L'origine est un point d'équilibre attractif pour (2.1) si  $\forall \varepsilon > 0$ , il existe un scalaire positif  $\delta(t_0)$  tel que :

$$\|x_0\| \leq \delta(t_0) \Rightarrow \lim_{t \rightarrow \infty} x(t, t_0, x_0)$$

Lorsque  $\delta(t_0) = +\infty$ , on dit que l'origine est globalement attractive.

**Définition 1.5 (Stabilité asymptotique)** : L'origine est un point d'équilibre asymptotiquement (respectivement globalement asymptotiquement) stable pour (2.1) s'il est stable et attractif (respectivement globalement attractif).

**Définition 1.6 (Stabilité exponentielle)** : L'origine est un point d'équilibre localement exponentiellement stable pour (2.1) s'il existe deux constantes strictement positives  $\alpha$  et  $\beta$  telles que :

$$\|x(t, t_0, x_0)\| \leq \alpha \exp(-\beta(t - t_0)), \quad \forall t \geq t_0, \alpha > 0, \beta > 0$$

Lorsque  $B_r = B_n$ , on dit que l'origine est globalement exponentiellement stable.

Notons que l'utilisation des définitions précédentes, pour démontrer la stabilité de (2.1) autour de son point d'équilibre, exige la résolution explicite de l'équation différentielle (2.1), ce qui est souvent très difficile voir impossible dans la plupart des cas. De ce fait, la méthode directe de Lyapunov permet de contourner cet obstacle. Cette méthode consiste à définir une fonction particulière dont l'existence garantit la stabilité.

### Méthode directe de Lyapunov :

La méthode directe de Lyapunov permet d'analyser la stabilité d'un système autour de son point d'équilibre sans le résoudre explicitement. L'existence d'une fonction particulière fournit des informations sur la stabilité du système.

**Définition 1.7** : Soit  $V(x, t) : \mathbb{R}^n \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$  une fonction continue.  $V$  est dite propre définie positive si :

1.  $\forall t \in \mathbb{R}^+, \forall x \in \mathbb{R}^n, x \neq 0, V(x, t) > 0$
2.  $\forall t \in \mathbb{R}^+ V(x, t) = 0 \Rightarrow x = 0$
3.  $\forall t \in \mathbb{R}^+, \lim_{x \rightarrow \infty} V(x, t) \rightarrow \infty$

**Définition 1.8 (Fonction de Lyapunov)** : Une fonction  $V(x, t)$  de classe  $C1$  est une fonction de Lyapunov locale (respectivement globale) au sens large pour le système (2.1) si elle est propre définie positive et s'il existe un voisinage de l'origine  $v_0$  tel que  $x \in v_0$  (respectivement  $x \in \mathbb{R}^n$ ) :

$$\dot{V}(x, t) = \frac{\partial v(x, t)}{\partial t} + \frac{\partial v(x, t)}{\partial x} f(x, t) < 0$$

Si  $\dot{V}(x,t) < 0$ , alors  $V(x,t)$  est appelée fonction de Lyapunov au sens strict pour (2.1).

**Définition 1.9 (Méthode directe de Lyapunov) :** Si le système (2.1) admet une fonction de Lyapunov locale au sens large (respectivement au sens strict) alors l'origine est un point d'équilibre localement stable (respectivement asymptotiquement stable).

Ce résultat peut être validé globalement pour  $x \in \mathbb{R}^n$ .

**Définition 1.10 (Stabilité exponentielle) :** L'origine de (2.1) est localement exponentiellement stable s'il existe des constantes  $\alpha, \beta$  et  $\gamma$  et une fonction  $V(x,t)$  :

$v_0 \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$  de classe  $C^1$  telles que,  $\forall x \in v_0$  :

1.  $\alpha \|x\|^p \leq V(x,t) \leq \beta \|x\|^p$ ;
2.  $\dot{V}(x,t) \leq -\gamma V(x,t)$

Si  $v_0 = \mathbb{R}^n$ , alors l'origine de (2.1) est globalement exponentiellement stable.

**Remarque 1.1 :** En choisissant la fonction de Lyapunov quadratique  $V(x) = x^T P x$ ,  $P = P^T$  le système linéaire :  $\dot{x}(t) = Ax(t)$  est globalement exponentiellement stable à l'origine si et seulement si  $P$  est la solution de l'équation  $A^T P + P A = -Q$ , pour une matrice  $Q$  symétrique et définie positive.

### II.3.Lemme de Barbalat :

Le lemme suivant sera utilisé intensivement à travers cette thèse, afin de démontrer la convergence de l'erreur de poursuite vers zéro.

**Lemme 1.1 :** Si  $f, \dot{f} \in L_\infty$  et  $f \in L_p$ , pour  $p \in [1, \infty[$  alors  $f \rightarrow 0$ , quand  $t \rightarrow \infty$ .

Le résultat du lemme 1.1 est u cas spécial d'un résultat plus général donné par le lemme de Barbalat indiqué ci-dessous.

**Lemme 1.2 : (Lemme de Barbalat) :**

*Si  $\lim_{t \rightarrow \infty} \int_0^t f(\tau) d\tau$  existe et est définie, et  $f(t)$  est une fonction uniformément continue, alors  $\lim_{t \rightarrow \infty} f(t) = 0$ .*

## **BIBLIOGRAPHIE :**

- [1] W. Khalil, E. Dombre, “Modélisation identification et commande des robots”, 2<sup>e</sup> édition, HERMES Science Publication, Paris, 1999.
- [2] Mark W. Spong and M. Vidiyasagr, “Robot Dynamics and Control”, QuinnWoodbine, USA, 1981
- [3] J. Angeles, “Fundamentals of Robotic Mechanical Systems”, Third edition, Springer Science +Business Media, LLC, 2007.
- [4] Frank L. Lewis, Darren M. Dawson, Chaouki T. Abdallah, “Robot Manipulator Control”, second edition, Marcel Dekker , Inc, USA, 2006.
- [5] B. Armstrong, O. Khatib and J. Burdick, “The explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm”, CH2282-2/86/0000/05101.00 © 1986 IEEE.
- [6] Thomas R. Kurfess, “ROBOTIC AND AUTOMATION HANDBOOK”, CRC PRESS, 2005
- [7] R. Kelly, V. Santibanez, A. Loria, “Control of Robot Manipulator in Joint Space”, Springer-Verlag London Limited 2005.
- [8] *Sous la direction de W. Khalil ; “Commande des robots manipulateurs”, LAVOISIER 2002*
- [9] J. Norberto Pires, “Industrial Robot Programming”, Springer Science+Business Media, LLC, 2005
- [10] Sam Cubero, “Industrial Robotics: Theory, Modelling and Control”, Advanced Robotic Systems International, 2007
- [11] J.J. Slotine, W. Lie, “Applied Nonlinear Control”, Printice Hall, 1990
- [12] Peter I. Corke, “Robotic toolbox”, The MathWorks, Inc, 2007
- [13] B. Armstrong, Peter I. Corke, “A Meta-study of PUMA 560 dynamics”, University of Wisconsin Milwankee, USA 1994

- [14] A. BOULEKROUNE, “Commande adaptative floue avec et sans observateur des systèmes non linéaires incertains”, Thèse de doctorat en science en Automatique, Ecole Nationale Polytechnique, 2008/2009
- [15] L. Wang; “Design and analysis of Fuzzy Identifiers of Nonlinear Dynamic Systems”, IEEE Trans. Automatic Control, vol. 40, no. pp 11-23, 1995
- [16] K. Ching Chou, S. Huang, “An adaptive fuzzy controller for robot manipulators”, Science direct, Mechatronics, 2004
- [17] H. Xu and A. Ioannou, “Robust Adaptive Control for a Class of MIMO Nonlinear Systems With Guaranteed Error Bounds”, IEEE Trans. Automatic Control, vol. 48, no. pp 728-742, 2003
- [18] J. Jantez, “Foundation of fuzzy control”, John Wesley Sons, 2007
- [19] K. TANAKA and H. Wang, “Fuzzy control systems design and analysis”, John Wesley & Sons, 2001
- [20] F. Hsu, L. Fu, “Nonlinear control of Robot Manipulators Using Adaptive Fuzzy Sliding Mode Control”, IEEE Trans. Automatic Control, vol.4, no. pp 156-161, 1995
- [21] S. Labiod, “Contribution à la commande adaptative floue des systèmes non linéaires ”, Thèse de doctorat en Automatique, Ecole Nationale Polytechnique, 2004
- [22] H. Chekireb, M. Tadjine, D. Bouchaffra, “Direct Adaptive Fuzzy Control of Nonlinear System Class with Applications”
- [23] Y. Jin, “Decentralized adaptive Fuzzy Control of Robot Manipulators”, IEEE Trans. Systems, Man and Cybernetics, vol. 28, no. pp 47-57, 1998