

## THESE

présentée au Laboratoire de Commande des Processus  
en vue de l'obtention du titre de

### Docteur d'Etat

en Automatique  
par

**Djamel BOUKHETALA**

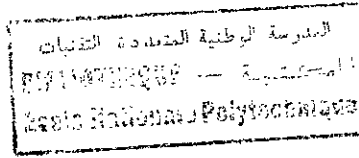
Magister en Automatique de l'ENP

### Thème

## Contribution à la commande décentralisée des systèmes complexes interconnectés. Applications en robotique

Soutenue publiquement le 02 Octobre 2002 devant le jury composé de:

D. BERKANI	Professeur à l'E.N.P	Président
M.S. BOUCHERIT	Maître de conférences à l'E.N.P	Rapporteur
N.K. M'SIRDI	Professeur à l'Université de Versailles (France)	Rapporteur
F. BOUDJÉMA	Professeur à l'ENP	Examineur
A. GUESSOUM	Professeur à l'Université de Blida	Examineur
M. TADJINÉ	Maître de conférences à l'E.N.P	Examineur
M. ATTARI	Maître de conférences à l'U.S.T.H.B	Invité



## THESE

présentée au Laboratoire de Commande des Processus  
en vue de l'obtention du titre de

### Docteur d'Etat

en Automatique  
par

**Djamel BOUKHETALA**

Magister en Automatique de l'ENP

### Thème

## Contribution à la commande décentralisée des systèmes complexes interconnectés. Applications en robotique

Soutenue publiquement le 02 Octobre 2002 devant le jury composé de:

D. BERKANI	Professeur à l'E.N.P	Président
M.S. BOUCHERIT	Maître de conférences à l'E.N.P	Rapporteur
N.K. M'SIRDI	Professeur à l'Université de Versailles (France)	Rapporteur
F. BOUDJEMA	Professeur à l'ENP	Examineur
A. GUESSOUM	Professeur à l'Université de Blida	Examineur
M. TADJINE	Maître de conférences à l'E.N.P	Examineur
M. ATTARI	Maître de conférences à l'U.S.T.H.B	Invité

تعتبر الأنظمة المتعددة التقنيات المعقدة إحدى اهتمامات الباحثين في مجال التحكم الآلي في الوقت الراهن. تتميز تلك الأنظمة ببنيتها المركبة والمعقدة ، كما يجعل طرق التحكم العادية غير صالحة لها لا سيما أثناء تطبيقها في الأوساط الصناعية، مما أدى الى التفكير في بنية تحكم جديدة يؤدي هذا التقسيم إلى انخفاض مستوى دقة و نجاعة التحكم عند استعمال الطرق المعروفة في مجال التحكم المركزي. العمل المقدم في هذه الأطروحة يندرج في هذا الإطار، حيث قمنا بإدخال تقنيات حديثة في هذا المجال كالتحكم ذي البنية المتغيرة، المنطق الغامض و الشبكات العصبونية الاصطناعية مما أدى بنا الى تطوير عدة طرق جديدة:

- الطريقة الأولى تم من خلالها اقتراح مساحة انزلاق لاحتية ذات صيغة عامة تشمل معظم الصيغ الموجودة حاليا، على إثرها اقترحنا قانون تحكم ذو بنية متغيرة
- الطريقة الثانية تتعلق بتطوير نظام تحكم تلاؤمي باستعمال المنطق الغامض. حيث استعمل هذا الأخير في تقريب جزء لاحتية من نموذج النظام.
- الطريقة الثالثة تعتبر توسيع الى مجال التحكم اللامركزي لطريقة تحكم تلاؤمي عن طريق الشبكات العصبونية الاصطناعية كانت قد طورت للتحكم المركزي.

كل الطرق المقترحة تم البرهان عنها نظريا كما تم تطبيقها عن طريق المحاكاة على ذراع آلي من نوع PUMA560 أين أثبتت النتائج المحصل عليها نجاعتها. قمنا أيضا بتطبيق مجري للطريقة الأولى على ذراع آلي حقيقي حيث أثبتت هذه التقنية قدرة عالية في التحكم. أخيرا قمنا باستعمال الشبكات العصبونية الاصطناعية عن طريق التدريب لطريقة العزم المحسوب المعروفة في مجال الآلية ثم استعملناها في التحكم اللامركزي. نتائج المحاكاة العددية أثبتت مدى قدرة الشبكات العصبونية الاصطناعية على تحسين مستوى أداء هذه الطريقة وكذا دور اللامركزية في تسهيل اختبار و تدريب الشبكات العصبونية.

**كلمات مفاتيح:** الأنظمة المتشابكة، التحكم اللامركزي، تحكم ذو بنية متغيرة، منطق غامض، الشبكات العصبونية الاصطناعية، ذراع آلي

**Abstract:**

Due to the complexity of the dynamic models and the high number of variables involved in the control schemes of the interconnected systems, design and implementation of the centralized controller are generally not evident. To overcome such problems, a decentralized structure has been proposed. In this structure, the control system is decomposed into a number of interconnected subsystems, in which, a local controller is designed independently using only local available signals, whereas centralized control theory is not applied directly because of the interconnection between subsystems.

This work deals with the development of a new decentralized control schemes using advanced tools such as variable structure theory, fuzzy logic and artificial neural networks (ANN) where some new approaches are proposed:

First, a new decentralized variable structure control scheme is proposed for robot manipulators. Furthermore, a new class of first order nonlinear sliding surfaces is given, on which the developed control law is based.

In the second part, a decentralized fuzzy logic adaptive controller is developed in which the fuzzy model is used to approximate some non linear functions.

An extension to the decentralized case of an adaptive centralized neural network controller is introduced in the third part. Indeed, in each controller the ANN is used to compensate a certain non linear terms of the model.

In all of these approaches, the stability is proved via Lyapunov theory. In order to highlight the tracking performances and the robustness of the proposed controllers, simulations are performed on the PUMA560 robot. Experimental results are also obtained on a three degree of freedom manipulator using the variable structure scheme.

Finally, to avoid a drawback of the computed torque method, decentralized neural networks are used to supervise this control law. The advantages of the decentralization in terms of design and training are clearly shown.

**Key words:** Interconnected systems, decentralized control, variable structure control, fuzzy logic, artificial neural networks, robot manipulator

**Résumé:**

La commande des systèmes complexes constitue à l'heure actuelle une des préoccupations des chercheurs en automatique. Généralement, les méthodes développées pour l'analyse et la conduite des systèmes, reposent sur des structures de commande centralisées ce qui engendre souvent des problèmes aussi bien en théorie qu'en pratique. Les difficultés proviennent de la structure de ces systèmes qui sont généralement constitués de plusieurs sous systèmes interconnectés. Pour pallier à ces inconvénients, une nouvelle structure de commande a été proposée. C'est la commande décentralisée. Il s'agit de décomposer la structure complexe en sous systèmes interconnectés dont chacun est commandé par une unité de commande locale. Par ailleurs, la décentralisation introduit une contrainte supplémentaire à cause de la présence des interconnexions

Le travail présenté dans cette thèse rentre dans ce contexte. En effet, nous nous sommes intéressés à la commande décentralisée appliquée en robotique. L'objectif principal de nos travaux est l'introduction des techniques de l'automatique moderne, telles que la théorie de la commande à structure variable, la théorie de la logique floue et les réseaux de neurones artificiels, pour la synthèse des lois de commande décentralisées.

En premier lieu, une nouvelle classe de surface de glissement non linéaire de forme générale est proposée. Une loi de commande décentralisée à structure variable est ensuite élaborée. Dans la seconde partie, un contrôleur adaptatif flou décentralisé est proposé où le modèle flou est utilisé pour l'approximation d'une certaine fonction non linéaire. Dans la troisième partie, l'extension d'une méthode de commande adaptative par réseaux de neurones au cas décentralisé est proposée. Pour les trois lois de commandes proposées, la stabilité est démontrée en utilisant la théorie de Lyapunov. En outre, ces méthodes sont validées par simulations sur un robot de type PUMA560. La commande à structure variable est aussi implantée expérimentalement sur un robot à trois degrés de liberté. En dernier lieu, les réseaux de neurones sont utilisés pour éliminer les défauts de la méthode du couple calculé. Les avantages de la décentralisation sont également mis en évidence.

**Mots clés :** Systèmes interconnectés, commande décentralisée, commande à structure variable, logique floue, réseaux de neurones artificiels, robot manipulateur

## Dédicace

*A mon épouse comme témoignage de ma profonde affection*

*A mes fils Mohsen et Nazih*

*A mes parents*

*A Smail et Soraya*

*A Haïthem et Rached*

*A mes amis*

*A toute la famille.*

## Avant propos



Ce travail a été effectué au Laboratoire de Commande des Processus de l'Ecole Nationale Polytechnique en collaboration avec le Laboratoire de Robotique de Paris de l'Université de Versailles Saint Quentin (France), sous la direction conjointe de Monsieur M.S. BOUCHERIT Maître de conférence à l'ENP et Monsieur N.K. M'SIRDI Professeur à l'UVSQ, qu'ils trouvent ici ma profonde gratitude pour leurs aides et encouragements.

J'adresse mes vifs remerciements au Professeur D.BERKANI pour l'honneur qu'il m'a fait en acceptant la présidence du jury de cette thèse.

Mes remerciements s'adressent également au Professeur F. BOUDJEMA pour l'importance qu'il a accordé à ces travaux en acceptant de participer au jury ainsi que pour son soutien et encouragement permanents. Ces qualités humaines, ces compétences scientifiques et son esprit critique ont contribué beaucoup à l'aboutissement de ce travail.

Que le Professeur A. GUESSOUM de l'université de Blida, soit vivement remercié pour avoir accepté d'être membre du jury malgré ses occupations.

Je remercie vivement Monsieur M. TADJINE Maître de conférence à l'ENP pour avoir accepté de faire partie du jury et pour le grand intérêt porté aux résultats de cette thèse.

Je tiens à exprimer mes vifs remerciements à Monsieur M. ATTARI Maître de conférence à l'USTHB pour avoir participé au jury.

J'adresse mes remerciements à T.MADANI, Z. TIDJANI, A. MELAHI et tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

Que SMAIL et SORAYA reçoivent ici mes sincères reconnaissances pour leurs aides et soutien que je n'oublierai jamais tant que je suis de ce monde.

Tous les membres du LCP et du Département de Génie Electrique reçoivent ici l'expression de ma profonde gratitude.

D. BOUKHETALA

# Table des matières

Introduction générale.....	1
<b>CHAPITRE I : Bases préliminaires sur les systèmes à structure variable, les systèmes flous et les réseaux de neurones artificiels.</b>	
I.0. Introduction.....	4
<b>I.1. Systèmes à structure variable.....</b>	<b>4</b>
I.1.1. Systèmes de réglage à structure variable.....	5
I.1.2. Surface de glissement.....	6
I.1.3. Solution des systèmes d'équations différentielles à second membre discontinu.....	7
I.1.3.1. Méthode de Fillpov.....	7
I.1.3.1. Méthode d'Utkin.....	8
I.1.4. Condition d'existence du mode de glissement.....	9
I.1.5. Commande par mode de glissement.....	10
<b>I.2. Notions fondamentales sur la logique floue.....</b>	<b>12</b>
I.2.1. Ensembles flous.....	12
I.2.2. Opérations sur les ensembles flous.....	13
I.2.3. Logique floue appliquée à la représentation et raisonnement linguistique.....	14
I.2.3.1. Variables linguistiques.....	15
I.2.3.2. Propositions floues.....	16
I.2.3.3. Conjonction et disjonction des propositions.....	16
I.2.3.4. Raisonnement flou.....	17
I.2.4. Application de la logique floue à la commande.....	21
I.2.4.1. Description des éléments d'un régulateur flou.....	22
I.2.4.2. Modèles flous.....	27
<b>I.3. Réseaux de neurones artificiels.....</b>	<b>29</b>
I.3.1. Le neurone artificiel.....	29
I.3.2. Les réseaux de neurones statiques.....	30
I.3.3. Apprentissage des réseaux de neurones artificiels.....	31
I.3.4. Les réseaux de neurones et l'approximation des fonctions.....	33
I.3.5. Les réseaux de neurones et la commande.....	34
<b>I.4. Conclusion.....</b>	<b>36</b>

## CHAPITRE II : Commande décentralisée à structure variable des robots manipulateurs.

II.1. Introduction.....	37
II.2. Description du modèle dynamique du robot manipulateur.....	38
II.3. Synthèse de la loi de commande.....	39
II.3.1. Sélection de la surface de glissement.....	40
II.3.2. Synthèse de la commande décentralisée à structure variable.....	42
II.4. Résultats de simulation.....	46
II.5. Résultats expérimentaux.....	52
II.6. Conclusion.....	57

## CHAPITRE III : Commande décentralisée adaptative par logique floue

III.1 Introduction.....	58
III.2. Modèle dynamique décentralisé.....	59
III.3. Commande adaptative décentralisée par logique floue.....	60
III.4. Résultats de simulation.....	71
III.5. Conclusion.....	79

## CHAPITRE IV : Commande adaptative décentralisée par réseaux de neurones

IV.1. Introduction.....	80
IV.2. Modèle dynamique décentralisé.....	81
IV.3. Commande Adaptative par réseaux de neurones adaptatifs.....	82
IV.3.1. Approximation des fonctions non linéaires par les réseaux de neurones à couches.....	82
IV.3.2. Détermination de la structure de commande.....	85
IV.4. Résultats de simulation.....	90
IV.5. Conclusion.....	93

## CHAPITRE V : Commande supervisée par réseaux de neurones ..... 94

## Conclusion générale..... 112

## Annexe..... 115

## Références bibliographiques..... 117

المدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

## Introduction générale



## Introduction générale

La commande des systèmes complexes constitue à l'heure actuelle une des préoccupations majeures des chercheurs en automatique. Généralement, les outils mathématiques classiques, développés pour l'analyse et la conduite des systèmes, reposent en majorité sur des structures de commande centralisées où tout le traitement de l'information se fait dans une même unité de commande. En effet, la centralisation engendre souvent des problèmes aussi bien en théorie qu'en pratique. Les difficultés proviennent d'une part, de la structure de ces systèmes qui sont généralement constitués de plusieurs sous systèmes interconnectés, parfois géographiquement éloignés, et d'autre part de leur dimension élevée.

Pour pallier à ces inconvénients, une nouvelle structure de commande a été proposée. C'est la commande décentralisée. Il s'agit de décomposer la structure complexe en sous systèmes interconnectés dont chacun est muni de capteurs et d'actionneurs qui lui sont propres, et commandé par une unité de commande locale.

L'intérêt de cette approche de commande est la simplification des algorithmes de commande ainsi que la rapidité dans le traitement des données. Par ailleurs, la décentralisation introduit une contrainte supplémentaire à cause de la présence des interconnexions entre les sous systèmes, ce qui engendre en général la détérioration des performances globales du système lorsque des lois de commande locales sont synthétisées par les approches centralisées.

En effet, plusieurs chercheurs se sont intéressés aux problèmes liés à la commande décentralisée. Les premiers résultats sur l'aspect analyse virent le jour vers le début des années soixante dix, lorsque Wang et Davison [WAN73] avaient étudié le problème de la stabilisation décentralisée linéaire en introduisant la notion de modes fixes. Ces derniers, apparaissent comme une mauvaise conséquence de la décentralisation. Suite à cela, plusieurs résultats sont obtenus sur la caractérisation et l'élimination de ces modes [BOU92], [BOU93a],[BOU93b]. En outre, pour l'aspect synthèse, un nombre important de résultats a été obtenu dans le domaine de la théorie de la commande décentralisée durant ces dernières années, allant de la commande linéaire jusqu'à la commande adaptative [BEN88], [GAV89], [OSS89], [SHI92], [WEN94], [VES97].

Parmi les domaines privilégiés pour l'application de la commande décentralisée, on peut citer entre autres : Les réseaux d'énergie électriques [DAV78],[JAI96],[JAI97a] et les robots manipulateurs [GAV87], [CHOI88],[SER87],[SER89],[FU92],[MAD99a],[BOU00a],[TAN00], [BOU01]. Néanmoins, certains résultats obtenus dans ce sens présentent des limites en terme de performances.

Le travail présenté dans cette thèse rentre dans ce contexte. En effet, nous nous sommes intéressés à la commande décentralisée appliquée en robotique. Ainsi, le robot manipulateur est considéré comme un système complexe interconnecté composé de plusieurs sous systèmes dont chacun caractérise une articulation. Cette dernière est commandée en poursuite de trajectoire par une unité de commande locale afin de garantir les performances requises.

L'objectif principal de nos travaux est l'introduction des techniques de l'automatique moderne pour la synthèse des lois de commande décentralisées des robots. Ainsi, nous proposons de nouvelles approches en utilisant la théorie de la commande à structure variable, la théorie de la logique floue et les réseaux de neurones artificiels.

Le premier chapitre est consacré à la présentation succincte des outils théoriques juste nécessaires pour la compréhension des développements qui suivent. Nous commençons d'abord par la théorie de la commande à structure variable où nous donnons les définitions de base ainsi que les caractéristiques de cette commande. Nous donnons, ensuite, des notions de base sur la théorie de la logique floue et son application dans le domaine de la commande. Enfin, nous présentons une description des réseaux de neurones artificiels ainsi que leurs utilisations en commande.

L'objet du second chapitre est la commande décentralisée à structure variable des robots. Nous commençons par donner une formulation décentralisée du modèle du robot en mettant en évidence les sous systèmes et les interconnexions entre eux. Une nouvelle classe de surface de glissement non linéaire de forme générale est ensuite proposée. La stabilité de cette surface est démontrée en utilisant la théorie de Lyapunov. A partir du modèle décentralisé du robot et la surface de glissement, une nouvelle loi de commande décentralisée à structure variable est élaborée. Afin de valider l'approche proposée, cette loi est d'abord appliquée en simulation sur un robot de type PUMA560, puis implantée expérimentalement sur un robot articulé à trois degrés de liberté du Laboratoire de Robotique de Paris. Des tests de performances, en terme de poursuite de trajectoire et de robustesse de la commande, sont présentés à la fin du chapitre.

Le troisième chapitre concerne l'utilisation de la théorie de la logique floue pour la commande décentralisée des robots manipulateurs. Une nouvelle approche de commande adaptative décentralisée par logique floue est proposée. Cette approche s'articule sur une représentation mathématique d'un modèle flou de type Sugeno. La stabilité du schéma proposé est démontrée. Le schéma de commande établi est validé par simulation sur un robot de type PUMA560. Des tests de performances en poursuite de trajectoire ainsi que de robustesse sont également donnés.

Une structure de commande adaptative décentralisée par réseaux de neurones artificiels est proposée dans le quatrième chapitre. La méthode proposée constitue une extension au cas décentralisée d'une méthode déjà proposée pour le cas centralisé [LEW93a]. Dans le développement de la loi de commande, nous utilisons la propriété d'approximation universelle

des fonctions non linéaires par les réseaux de neurones artificiels à couches. La stabilité du système bouclé est aussi démontrée. Afin de tester la validité de la méthode ainsi que ces performances en poursuite de trajectoires et sa robustesse vis à vis des variations paramétriques, des simulations sont effectuées sur le modèle du robot PUMA560.

Dans le dernier chapitre, nous proposons une version décentralisée de la méthode du couple calculé, que nous supervisons par des réseaux de neurones statiques afin d'éliminer le défaut de cette méthode, à savoir sa sensibilité aux variations paramétriques. Pour mettre en exergue les avantages de la supervision par réseaux de neurones ainsi que ceux de la décentralisation de la commande, la méthode est appliquée en simulation sur un robot manipulateur cylindrique.

# CHAPITRE I

## Bases préliminaires sur les systèmes à structure variable, les systèmes flous et les réseaux de neurones artificiels

### I.0. Introduction

Le but de ce chapitre est de rappeler quelques notions et outils nécessaires à la compréhension du contenu de ce mémoire. Il est scindé en trois parties :

Dans une première partie, nous donnons brièvement quelques éléments de base sur la théorie des systèmes à structure variable en mettant en évidence leurs configurations de base ainsi que leurs propriétés. La seconde partie est consacrée à la théorie de la logique floue, où nous donnons quelques définitions mathématiques sur les ensembles flous et leurs caractéristiques. Nous terminons cette partie par donner les différentes structures de base de la commande à l'aide de modèles flous. En dernier lieu, nous introduisons les réseaux de neurones artificiels en se limitant aux réseaux statiques à couches. Leur structure, leur algorithme d'apprentissage ainsi que leur utilisation en commande des systèmes sont données d'une manière succincte. Les approches développées dans les chapitres suivants sont essentiellement basées sur ces notions.

### I.1. Systèmes à Structure Variable

La théorie de la commande à structure variable a vu le jour à l'issue des travaux du mathématicien soviétique Filippov F. (1960) en résolvant le problème de l'équation différentielle à second membre discontinu. Ces travaux ont été suivis par des recherches avancées menées par le professeur Emil-Yanov en 1967 avec son équipe ainsi que le professeur Utkin V. I. (1978). Cette théorie est à la base de beaucoup de résultats de recherche actuelle dans le domaine de la commande des systèmes [GUZ86], [HUN93], [LEU91], [MAT88], [MOR85], [MAD99a], [MAD99b], [MAD00],...

La commande par mode de glissement est un cas particulier des systèmes à structure variable. Elle est caractérisée par la commande à commutation à haute fréquence (théoriquement infinie), afin d'amener la trajectoire d'état d'un système vers une surface appelée surface de glissement. Cette dernière doit être synthétisée pour avoir une dynamique désirée. La commutation du signal de commande est faite selon une certaine logique, afin d'amener le système vers un point d'équilibre.

Parmi les principales propriétés des modes glissants nous pouvons citer:[BOU91]

- La trajectoire d'état du système en mode de glissement appartient à une surface de commutation inférieure à celle de l'espace d'état. par conséquent l'ordre des équations différentielles décrivant le fonctionnement du système en mode de glissement est réduit (réduction d'ordre) :
- La dynamique du système en mode de glissement est fixée par le choix approprié de la surface de glissement :
- La théorie des modes glissants appartient bien aux systèmes dont les commandes sont discontinues ;
- La théorie des systèmes à structure variable, particulièrement les modes glissants, présente une robustesse vis à vis des perturbations externes et des variations paramétriques :
- La mise en pratique de cette technique étant liée exclusivement au développement de l'électronique de puissance qui apporte des circuits de commutations à haute fréquence.

### I.1.1. Système de réglage à structure variable

La figure I.1.1, donne une représentation générale englobant toutes les configurations de base des systèmes à structure variable

Cette représentation est caractérisée par trois blocs élémentaires constituant la commande globale, à savoir :

- le premier bloc (I) contient deux vecteurs de commandes auxiliaires, l'un des deux délivre le signal  $u_1$  selon le signe de la loi de commutation  $\sigma(x, x_d, t)$  ;
- le second bloc (II) fournit la commande  $u_2$ . Cette commande est formée en multipliant le vecteur d'état désiré par une matrice d'anticipations  $\psi^+(x_d, t)$  ou  $\psi^-(x_d, t)$  par le biais du signe de la loi de commutation ;
- le troisième bloc (III) fournit la commande  $u_3$ . Cette dernière est formée en multipliant le vecteur d'état  $x$  par une matrice de contre réaction  $\phi^+(x_d, t)$  ou  $\phi^-(x_d, t)$  toujours selon le signe de loi de commutation.

La commande résultante est décrite par la forme non linéaire suivante :

$$u = u_1 + u_2 + u_3 = \begin{cases} u^+ = \gamma^+(t) + \psi^+(x_d, t)x_d + \phi^+(x, t)x & \text{si } \sigma(x, x_d, t) > 0 \\ u^- = \gamma^-(t) + \psi^-(x_d, t)x_d + \phi^-(x, t)x & \text{si } \sigma(x, x_d, t) < 0 \end{cases} \quad (\text{I.1.1})$$

### I.1.2. Surface de glissement

L'une des principales notions de la commande à structure variable est la notion de surface de glissement, appelée aussi loi de commutation. Cette surface est caractérisée par un hyperplan dans l'espace d'état (une ligne pour un système de deuxième ordre, un plan pour un système de troisième ordre, etc...). La commande à structure variable consiste à synthétiser une surface de glissement stable, et ramener la trajectoire d'état du système vers cette surface afin d'atteindre le point d'équilibre. Cela se fait par le biais d'une commande discontinue oscillant à haute fréquence (fréquence théoriquement infinie). Les commutations par rapport à cette surface présentent des pics très fins (théoriquement négligeables). Le choix approprié de cette surface de glissement est primordial. Cependant, il n'existe pas de méthodes systématiques pour ce choix.

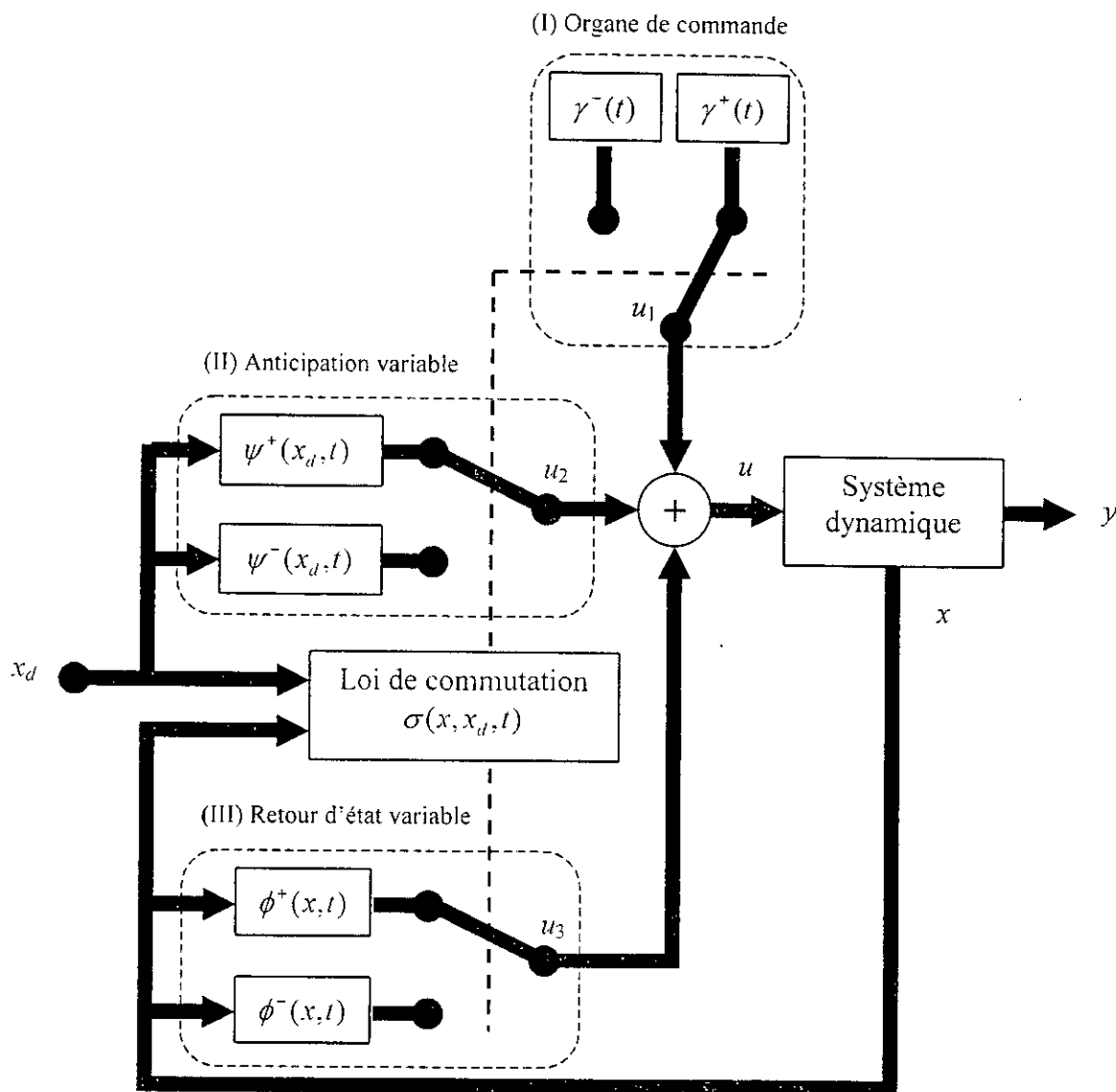


Figure I.1.1. Configurations de base des systèmes à structure variable.

# **CHAPITRE I**

Bases préliminaires sur les systèmes à structure variable, les systèmes flous et les réseaux de neurones artificiels

En général, la surface de glissement est choisie linéaire par rapport aux variables d'états. Pour un système ayant  $m$ -entrées, le vecteur de la surface de glissement linéaire peut prendre la forme suivante

$$\sigma = S x + S_d x_d \quad (I.1.2)$$

avec  $x \in \mathbb{R}^n$  vecteur d'état du système ;  $x_d \in \mathbb{R}^n$  vecteur d'état désiré ;  $S \in \mathbb{R}^{m \times n}$  est le vecteur des gains définissant la dynamique de la surface de glissement ;  $S_d \in \mathbb{R}^{m \times n}$  est vecteur des gains de la consigne.

Dans le chapitre suivant, nous allons proposer une surface non linéaire de forme générale pour la commande décentralisée des robots.

### I.1.3. Solution des systèmes d'équations différentielles à second membre discontinu

#### I.1.3.1. Méthode de Phillipov

Ce premier résultat fut obtenu par Phillipov F.(1960) en démontrant l'existence et l'unicité de la solution des systèmes dynamiques à structure variable en régime glissant idéal. Le principe de la méthode est le suivant :

Soit un système non-linéaire décrit par l'équation d'état

$$\dot{x} = f(x, u, t) \quad (I.1.3)$$

avec une commande discontinue

$$u = \begin{cases} u^+ & \text{si } \sigma > 0 \\ u^- & \text{si } \sigma < 0 \end{cases} \quad (I.1.4)$$

La trajectoire d'état en régime glissant s'écrit

$$\dot{x} = f^0 = \mu f^+ + (1 - \mu) f^- \quad \text{avec} \quad 0 \leq \mu \leq 1 \quad (I.1.5)$$

avec  $\mu$  dépend de la direction de la grandeur des vecteurs colonne  $f^+$ ,  $f^-$  et le rang du vecteur gradient du  $\sigma$ .



$$\mu = \frac{\text{grad} [\sigma] f^-}{\text{grad} [\sigma] (f^- - f^+)} \quad (\text{I.1.6})$$

Substituant l'expression de  $\mu$  dans l'équation (I.1.5) on trouve :

$$\dot{x} = \frac{\text{grad} [\sigma] f^-}{\text{grad} [\sigma] (f^- - f^+)} f^+ - \frac{\text{grad} [\sigma] f^+}{\text{grad} [\sigma] (f^- - f^+)} f^- \quad (\text{I.1.7})$$

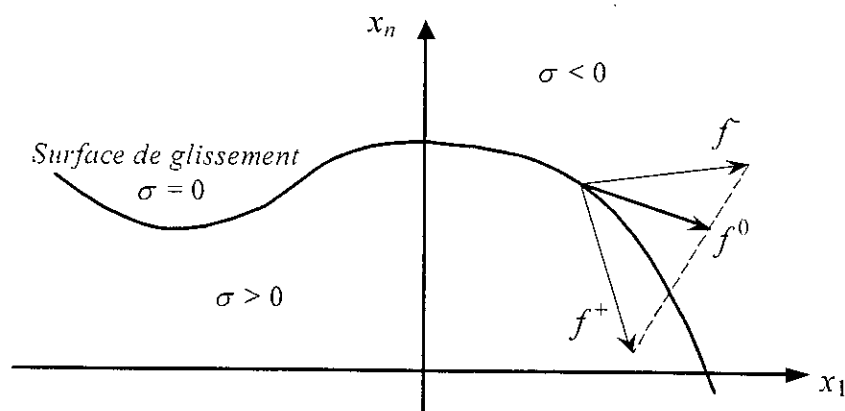


Figure I.1.2. Représentation des champs de vecteur pour la méthode de Fillipov.

La méthode de Fillipov consiste à trouver le résultat de la vitesse du champ de vecteur  $f^0$  qui est piloté par la combinaison convexe de deux champs de vecteurs  $f^+$  en dessous et  $f^-$  en dessus dans l'intersection de la surface est la tangente en un point donné.

### I.1.3.2 Méthode d'Utkin

Cette méthode est basée sur la commande équivalente (Utkin, 1978). Nous donnons ici juste un aperçu sur cette approche.

Soit le système non-linéaire décrit par l'équation d'état :

$$\dot{x} = f(x, t) + g(x, t) u \quad (\text{I.1.8})$$

En régime de glissement idéal la loi de commutation s'annule (expression de la surface de glissement est vérifiée), implicitement il n'y a pas de variation dans la loi de commutation. En effet,  $\dot{\sigma}(x, \dot{x}, t) = 0$

avec :

$$\dot{\sigma}(x, \dot{x}, t) = \left( \frac{\partial \sigma}{\partial x} \right) \dot{x} + \frac{\partial \sigma}{\partial t} \quad (\text{I.1.9})$$

L'introduction de l'équation (I.1.8) dans (I.1.9) donne

$$\dot{\sigma}(x, \dot{x}, t) = \left( \frac{\partial \sigma}{\partial x} \right) \left( f(x, t) + g(x, t) u_{eq} \right) + \frac{\partial \sigma}{\partial t} = 0 \quad (\text{I.1.10})$$

avec  $u_{eq}$  est la commande équivalente en mode glissant.

D'après (I.1.10), la commande équivalente est donnée s'écrit

$$u_{eq} = - \left[ \left( \frac{\partial \sigma}{\partial x} \right) g(x, t) \right]^{-1} \left\{ \left( \frac{\partial \sigma}{\partial x} \right) f(x, t) + \frac{\partial \sigma}{\partial t} \right\} \quad (\text{I.1.11})$$

avec la condition nécessaire de *transversalité* :

$$\det \left[ \left( \frac{\partial \sigma}{\partial x} \right) g(x, t) \right] \neq 0 \quad (\text{I.1.12})$$

L'équation d'état du système équivalent en mode de glissement idéal sera donc:

$$\dot{x} = \left\{ I - g(x, t) \left[ \left( \frac{\partial \sigma}{\partial x} \right) g(x, t) \right]^{-1} \left( \frac{\partial \sigma}{\partial x} \right) \right\} f(x, t) - g(x, t) \left[ \left( \frac{\partial \sigma}{\partial x} \right) g(x, t) \right]^{-1} \frac{\partial \sigma}{\partial t} \quad (\text{I.1.13})$$

#### I.1.4. Condition d'existence du mode de glissement

Le domaine du régime de glissement, appelé aussi région d'attraction, est une partie de l'espace d'état globale dans lequel, pour toutes conditions initiales, le vecteur vitesse de la trajectoire d'état à une direction vers la surface de glissement.

La condition d'existence du mode de glissement est liée à la convergence de la trajectoire d'état vers le point d'équilibre dans le domaine du régime de glissement. L'existence du régime de glissement peut être déterminée en utilisant la théorie de la stabilité de Lyapunov. Ceci se résume par :

$$\sigma \dot{\sigma} < 0 \text{ (attractivité)}, \quad \sigma = 0 \text{ et } \dot{\sigma} = 0 \text{ (invariance)} \quad (\text{I.1.14})$$

### I.1.5. Commande par mode de glissement

La synthèse de la commande à structure variable se fait en deux étapes:

- sélection d'une surface de glissement,
- construction d'une commande amenant la système vers le point d'équilibre sur cette surface.

La commande peut prendre diverses formes. L'une des formes les plus connues est

$$u = u_{eq} + u_N \quad (I.1.15)$$

$u_{eq}$  désigne la commande équivalente,  $u_N$  est une commande discontinue variant autour d'une moyenne, avec l'hypothèse que le système initial est en régime de glissement idéal.

L'utilisation de cette loi de commande donne

$$\dot{\sigma}(x) = \frac{\partial \sigma}{\partial x} \left[ f(x,t) + g(x,t) u_{eq} \right] + \frac{\partial \sigma}{\partial x} g(x,t) u_N \quad (I.1.16)$$

En régime de glissement idéal, la commande appliquée au système est la commande équivalente

$u_{eq}$

d'où :

$$\frac{\partial \sigma}{\partial x} \left[ f(x,t) + g(x,t) u_{eq} \right] = 0 \quad (I.1.17)$$

Il en résulte d'après (I.1.17) que

$$\dot{\sigma}(x) = \frac{\partial \sigma}{\partial x} g(x,t) u_N \quad (I.1.18)$$

Le choix de la surface de glissement est arbitraire. Pour des raisons de simplification, on peut sélectionner une surface de glissement vérifiant la relation

$$\frac{\partial \sigma}{\partial x} g(x,t) = I \quad (I.1.19)$$

avec  $I$  est une matrice identité .

Dans ce cas, (I.1.18) et (I.1.19), la commande discontinue  $u_N$  s'identifie à la dérivé de la surface de glissement c'est à dire  $\dot{\sigma}(x) = u_N$ . Cette commande doit être choisie afin que  $\sigma_i \dot{\sigma}_i < 0$  quand  $\sigma_i \neq 0$  pour  $i = 1 \dots m$  soit vérifiée.

Il existe plusieurs formes pour cette commande discontinue, nous citons entre autres:

$$\bullet \quad u_N = \begin{cases} \gamma_i \operatorname{sgn}(\sigma_i) & \text{pour } \sigma_i \neq 0 \\ 0 & \text{pour } \sigma_i = 0 \end{cases} \quad (\text{I.1.20})$$

avec  $\gamma_i < 0$ .

La condition d'existence du régime de glissement est vérifiée

$$\sigma_i \dot{\sigma}_i = \gamma_i \sigma_i \operatorname{sgn}(\sigma_i) < 0 \quad (\text{I.1.21})$$

$$\bullet \quad u_N(x) = \begin{cases} \gamma_i(x) \operatorname{sgn}(\sigma_i) & \text{pour } \sigma_i \neq 0 \\ 0 & \text{pour } \sigma_i = 0 \end{cases} \quad (\text{I.1.22})$$

avec  $\gamma_i(x) < 0$ .

d'où

$$\sigma_i \dot{\sigma}_i = \gamma_i(x) \sigma_i \operatorname{sgn}(\sigma_i) < 0 \quad (\text{I.1.23})$$

$$\bullet \quad u_N(x) = \phi x \quad (\text{I.1.24})$$

avec

$$\phi_{ij} = \begin{cases} \alpha_{ij} & \text{si } x_j \sigma_i > 0 \\ \beta_{ij} & \text{si } x_j \sigma_i < 0 \end{cases} \quad (\text{I.1.25})$$

et

$$\alpha_{ij} < 0 \text{ et } \beta_{ij} > 0 \quad (\text{I.1.26})$$

d'où

$$\sigma_i \dot{\sigma}_i = \sigma_i (\phi_{i1} x_1 + \dots + \phi_{in} x_n) < 0 \quad (\text{I.1.27})$$

$$\bullet \quad u_N = -L\sigma \quad (\text{I.1.28})$$

avec  $L$  une matrice constante  $m \times m$  définie positive.

d'où

$$\sigma^T \dot{\sigma} = -\sigma^T L \sigma < 0 \quad (\text{I.1.29})$$

$$\bullet \quad u_N = \frac{\sigma}{\|\sigma\|} \rho \quad (\text{I.1.30})$$

avec  $\rho < 0$

d'où

$$\sigma^T \dot{\sigma} = \|\sigma\| \rho < 0 \quad (\text{I.1.31})$$

## 1.2. Notions fondamentales sur la logique floue

La logique floue a été introduite en 1965 par L.A. ZADEH. Ses principes ont été appliqués en 1974 par E.H. MAMDANI à la construction d'un premier contrôleur flou. Mais, ce n'est que depuis quelques années que la commande floue a connu, essentiellement au Japon, un essor remarquable ; Elle a été appliquée à des problèmes aussi divers que la purification de l'eau, la fabrication du ciment, etc...

L'attitude des automaticiens à l'égard de la commande floue a d'abord été réservée : après avoir affirmé, pendant des décennies, la nécessité d'identifier le mieux possible un système pour pouvoir construire une commande ayant des performances satisfaisantes, fallait-il faire confiance à cette méthode nouvelle, qui prétendait remplacer les commandes basées sur l'identification, par des techniques s'appuyant sur le savoir faire humain plutôt que sur des équations. Actuellement, cette attitude a évolué. On peut dire qu'aujourd'hui la commande constitue un outil précieux pour la commande des systèmes difficilement identifiables ou dont les paramètres subissent des variations brutales.

Dans ce qui suit, nous donnons quelques définitions de base liées à la notion de la logique floue [TID99].

### 1.2.1. Ensembles flous

La notion de l'ensemble flou provient du constat établi par Zadeh qui stipule que : « *très souvent les classes des objets rencontrées dans le monde physique, ne possèdent pas de critères d'appartenance bien définis* ». Le concept de l'ensemble flou permet de considérer des classes d'objets dont les frontières ne sont pas clairement déterminées, par l'introduction d'une fonction caractéristique (fonction d'appartenance des objets à la classe) prenant des valeurs courantes entre 0 et 1, contrairement aux ensembles booléens dont la fonction caractéristique ne prend que deux valeurs possibles 0 ou 1.

En logique classique un élément  $x$  soit il appartient à un ensemble  $A$  ou non. Donc, on peut associer à la propriété *appartenir* la fonction caractéristique suivante :

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

En logique floue, un ensemble dit *flou*, est la généralisation de la notion de l'ensemble en logique classique. La propriété d'appartenance est définie par la fonction caractéristique suivante :

$$\begin{array}{l} \mu_F : U_x \rightarrow [0, 1] \\ x \rightarrow \mu_F(x) \end{array}$$

où,  $U_x$  est l'univers de discours de  $x$ .  $\mu_F(x)$  est appelée fonction d'appartenance.

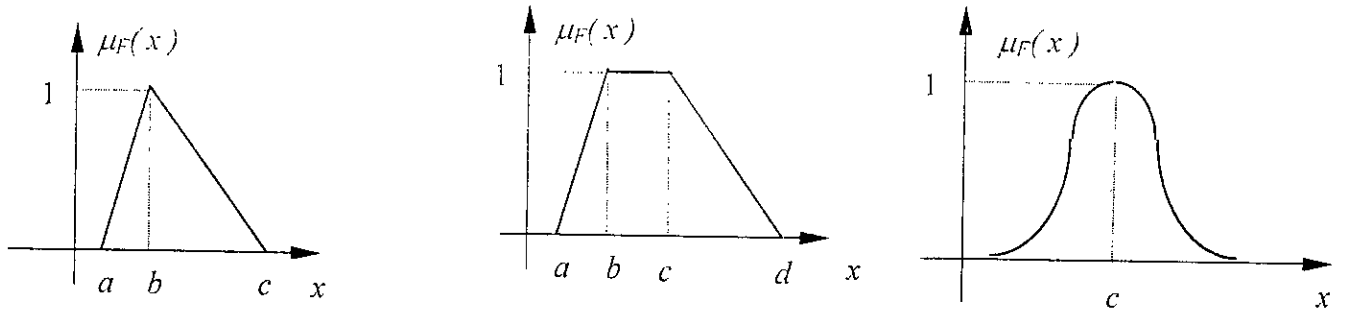
L'ensemble flou noté  $\underline{F}$  est défini par l'ensemble des couples  $(x, \mu_F(x))$ . Donc :

$$\underline{F} = \int_x \mu_F(x) / x$$

Le symbole  $\int_x$  indique la collection sur tout l'intervalle  $U_x$ .

Parmi les fonctions d'appartenance les plus utilisées, on cite [JAN95]: les fonctions triangulaires, trapézoïdales, gaussiennes, sigmoïde ...

La figure (I.2.1) montre l'allure de quelques unes.



$$\text{Triangle}(x, a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{b-c}\right), 0\right)$$

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

$$\text{Gaussienne}(x, \sigma, c) = \exp\left(-\left(\frac{x-c}{\sigma}\right)^2\right)$$

Figure I.2.1: Exemples de fonctions d'appartenance

## I.2.2. Opérations sur les ensembles flous

Puisque les fonctions caractéristiques des ensembles flous sont graduelles, l'extension de certaines opérations définies sur les ensembles classiques au cas des ensembles flous pose certaines difficultés [JAN95]. Donc, il n'est pas possible d'appliquer directement la logique propositionnelle qui manipule des ensembles à fonctions caractéristiques binaires. Pour palier à ce problème, on introduit les définitions de trois outils nécessaires pour le traitement des ensembles flous.

Soit, les nombres réels :  $a, b, c$  et  $d \in [0, 1]$ .

- **Norme triangulaire ( T-norme )**: [DRI93]

La norme triangulaire de  $a$  et  $b$ , notée  $a \otimes b$ , dénote une classe de fonctions binaires qui peuvent représenter l'opération d'**intersection** entre ensembles. Elle satisfait les propriétés suivantes:

$$T.1 \quad (a \otimes b) \otimes c = a \otimes (b \otimes c)$$

$$T.2 \quad a \leq c \text{ et } b \leq d \Rightarrow a \otimes b \leq c \otimes d$$

$$T.3 \quad a \otimes 1 = a$$

$$T.4 \quad a \otimes b = b \otimes a$$

L'intersection de deux ensembles flous  $\underline{F}_1$  et  $\underline{F}_2$  est l'ensemble  $\underline{F} = \underline{F}_1 \cap \underline{F}_2$  défini sur l'univers  $U_x$  de fonction d'appartenance :

$$\mu_F(x) = \mu_{F_1 \cap F_2}(x) = \mu_{F_1}(x) \otimes \mu_{F_2}(x) \quad (I.2.1)$$

Parmi les opérations d'intersection on cite:

- Opérateur minimum noté "min" :  $\mu_{F_1 \cap F_2}(x) = \min(\mu_{F_1}(x), \mu_{F_2}(x))$
- Opérateur produit noté "x" :  $\mu_{F_1 \cap F_2}(x) = \mu_{F_1}(x) \times \mu_{F_2}(x)$

### - Norme co-triangulaire ( T-conorme ou S-norme )

La conorme triangulaire de  $a$  et  $b$ , notée  $a \oplus b$ , dénote une classe de fonctions binaires qui peuvent représenter l'opération d'**union** entre ensembles. Elle satisfait les propriétés suivantes :

- S.1  $a \oplus b = b \oplus a$
- S.2  $(a \oplus b) \oplus c = a \oplus (b \oplus c)$
- S.3  $a \leq c$  et  $b \leq d \Rightarrow a \oplus b \leq c \oplus d$
- S.4  $a \oplus 0 = a$

L'union de deux ensembles flous  $\underline{F}_1$  et  $\underline{F}_2$  est l'ensemble  $\underline{F} = \underline{F}_1 \cup \underline{F}_2$  défini sur l'univers de discours  $U_x$ , de fonction d'appartenance :

$$\mu_{\underline{F}}(x) = \mu_{\underline{F}_1 \cup \underline{F}_2}(x) = \mu_{\underline{F}_1}(x) \oplus \mu_{\underline{F}_2}(x) \quad (I..2.2)$$

Parmi les opérations d'union :

- Opérateur maximum noté "max" :  $\mu_{\underline{F}_1 \cup \underline{F}_2}(x) = \max(\mu_{\underline{F}_1}(x), \mu_{\underline{F}_2}(x))$
- Opérateur somme algébrique :  $\mu_{\underline{F}_1 \cup \underline{F}_2}(x) = \mu_{\underline{F}_1}(x) + \mu_{\underline{F}_2}(x) - \mu_{\underline{F}_1}(x) \times \mu_{\underline{F}_2}(x)$

### - Complément

L'opérateur de complément noté ( $\bar{\quad}$ ), doit au moins satisfaire les propriétés suivantes :

- C.1  $a(\bar{0}) = 1$
- C.2  $a < b \Rightarrow \bar{a} > \bar{b}$
- C.3  $\overline{\bar{a}} = a$

Le complément d'un ensemble flou  $\underline{F}_1$  est l'ensemble flou  $\overline{\underline{F}_1}$  défini sur l'univers de discours  $U_x$  dont la fonction d'appartenance est donnée par :

$$\mu_{\overline{\underline{F}_1}}(x) = \overline{\mu_{\underline{F}_1}(x)} \quad (I.2.3)$$

- Opérateur de complément :  $\mu_{\overline{\underline{F}_1}}(x) = 1 - \mu_{\underline{F}_1}(x)$

### I.2.3. Logique floue appliquée à la représentation et raisonnement linguistique

En utilisant les ensembles et les relations floues, on peut représenter et raisonner sur des expressions linguistiques ou des mots du langage naturel.

Considérons l'expression suivante: [DRI93]

*" Si c'est le cas où l'erreur actuelle a la propriété d'être négative grande et la variation de l'erreur actuelle a la propriété d'être positive grande, alors c'est une raison pour changer la commande en l'incrémentant pour qu'elle ait la propriété négative petite".*

Dans cette expression interviennent les notions clefs suivantes :

- Des variables linguistiques : Erreur, Variation d'erreur
- Propriétés des variations linguistiques : négative grande, positive grande . . .
- Des connecteurs linguistiques : "et", "ou" "à la propriété"
- Propositions :
  - "Erreur actuelle . . . négative grande"
  - "Variation de l'erreur . . . positive grande"
- Conditionnement : Si . . . alors . . .

Dans le but de représenter l'expression précédente à l'aide de la logique floue, on introduit une représentation symbolique des différentes notions citées ci-dessus. L'association d'ensembles et de relations floues à la représentation symbolique permettra de quantifier l'expression.

### 1.2.3.1. Variables linguistiques

C'est l'unité fondamentale pour représenter les connaissances pour être traitées par le raisonnement à logique floue. Une variable linguistique est une variable dont les valeurs sont définies par des termes du langage naturel.

Chaque variable linguistique est déterminée par quatre caractéristiques  $(X, LX, U_X, M_X)$  où

$X$  : le nom symbolique de la variable. Exemple :  $E$  pour l'erreur  $\Delta E$  pour la variation d'erreur.

$LX$  : l'ensemble des valeurs de la variable  $X$ . (par exemple :  $LE = \{N, Z, P\}$  où  $N, Z$  et  $P$  indiquent respectivement, négative, nulle et positive).

$U_X$  : l'univers de discours dans lequel la variable de symbole  $X$  prend ses valeurs. Si la variable est normalisée entre -1 et +1 alors  $U_X = [-1 +1]$ .

$M_X$  : fonction sémantique, qui attribue à chaque valeur de  $X$  son sens. Ce dernier est caractérisé par un ensemble flou défini sur l'univers de discours  $U_X$ , donc :

$$M_X \quad LX \longrightarrow \underline{LX}$$

où

$$\underline{LX} = \int_{U_X} \mu_{LX}(x_i) / x_i$$

Donc la variable  $X$  peut être considérée comme une variable floue dont les valeurs sont les éléments  $\underline{LX}$ .

**Exemple :** Considérons le cas de la variable Erreur, on a  $LE = \{N, Z, P\}$ . La fonction  $M_X$  consiste à attribuer à ces valeurs des ensembles flous  $\underline{N}, \underline{Z}, \underline{P}$ . La figure (1.3) montre une possibilité de choix d'ensembles.

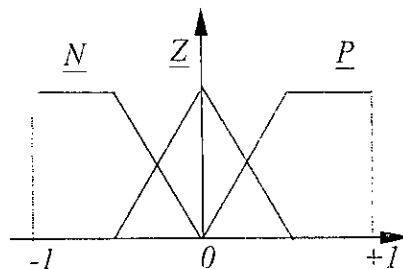


Figure 1.2.2. Exemple d'ensembles flou associés à la variable erreur



Dans les zones de chevauchement (figure I.2.2.), la variable floue prend plus d'une valeur à la fois mais avec des degrés d'appartenance différents, ce qui reflète l'imprécision sur les frontières de ces valeurs. Le passage entre des valeurs floues voisines qui se chevauchent est généralement graduel.

### 1.2.3.2. Propositions floues

Il existe deux types de propositions, simples et composées

#### - Propositions floues simples

Considérons l'expression : "*Erreur à la propriété d'être négative grande*". Une représentation symbolique possible est la suivante:

$$"E \text{ est } NB",$$

où "*est*" réalise le sens de "à la propriété d'être";

*NB* : le symbole d'un ensemble flou *NB* associé à la valeur négative grande.

Dans ce cas la représentation symbolique "*E est NB*", est appelée proposition floue simple.

#### - Proposition floue composée

La combinaison de plusieurs propositions floues simples par les connecteurs "*et*", "*ou*" et "*SI-ALORS*" permet de générer des propositions floues composées.

#### Exemple :

Considérons les valeurs de l'erreur  $LE = \{NB, NM, NP, Z, PP, PM, PG\}$  et de sa variation  $L\Delta E = \{N, Z, P\}$ , où *NB* désigne négative grande, *NM* : négative moyenne, *NP* : négative petite, *Z* : nulle, *PP* : positive petite, *PM* : positive moyenne, *PG* : positive grande.

A partir de ces valeurs, on peut construire les propositions floues composées suivantes : "*E est NG ou E est PP*"; "*E est NG et  $\Delta E$  est P*"; "*E est PM ou  $\Delta E$  est N*"; "*E n'est pas P*".

De telles propositions peuvent être quantifiées en utilisant les notions : conjonction, disjonction et négation.

### 1.2.3.3. Conjonction et disjonction des propositions

#### - Conjonction

La conjonction des propositions  $P_1, P_2, \dots, P_n$  définie par

$$P_1 : "X \text{ est } F_1", \dots, P_n : "X \text{ est } F_n"$$

est la proposition composée  $P$  telle que :

$$P : "X \text{ est } F_1 \text{ et } X \text{ est } F_2 \text{ et } \dots \text{ et } X \text{ est } F_n"$$

Le sens de la proposition  $P$  est représenté par un ensemble flou  $\underline{F}$  résultat de l'intersection des ensembles flous  $\underline{F}_1, \underline{F}_2, \dots$  et  $\underline{F}_n$ . on obtient :

$$\mu_{\underline{F}}(x) = \mu_{F_1}(x) \otimes \mu_{F_2}(x) \otimes \dots \otimes \mu_{F_n}(x) \quad x \in U_x \quad (I.2.4)$$

on écrit :

$$\mu_{\underline{F}}(x) \hat{=} \bigotimes_{i=1}^n \mu_{F_i}(x)$$

### - Disjonction

La disjonction des propositions  $P_1, P_2, \dots, P_n$  définies par :

$$P_1 : "X \text{ est } F_1", \dots, P_n : "X \text{ est } F_n"$$

est la proposition composée  $P$  telle que :

$$P : "X \text{ est } F_1 \text{ ou } X \text{ est } F_2 \text{ ou } \dots \text{ ou } X \text{ est } F_n"$$

Le sens de la proposition  $P$  est représenté par un ensemble flou  $\underline{F}$  résultat de l'union des ensembles flous  $\underline{F}_1, \underline{F}_2, \dots$  et  $\underline{F}_n$ . D'après (I.2.2), on obtient :

$$\mu_{\underline{F}}(x) = \mu_{F_1}(x) \oplus \mu_{F_2}(x) \oplus \dots \oplus \mu_{F_n}(x) \quad x \in U_x \quad (I.2.5)$$

on écrit :

$$\mu_{\underline{F}}(x) \hat{=} \bigoplus_{i=1}^n \mu_{F_i}(x)$$

### - Négation

La négation d'une proposition  $P$  donnée par :

$$P : "X \text{ est } F" \text{ est la proposition } P : "X \text{ n'est pas } F"$$

dont le sens est représenté par l'ensemble flou  $\overline{F}$  tel que :

$$\overline{F} = \int_{U_x} \mu_{\overline{F}}(x) / x$$

où  $\overline{F}$  est l'ensemble flou complément de l'ensemble flou  $\underline{F}$ .

### 1.2.3.4. Raisonnement flou

En général, plusieurs valeurs de variables linguistiques convenablement définies par des fonctions d'appartenances sont liées entre elles par des règles afin de tirer des conclusions. Ces règles utilisent souvent les connecteurs *SI-ALORS*. La procédure utilisée pour déduire des conclusions à partir d'un ensemble de règles et de conditions, représentés par des ensembles et relations floues, est appelée raisonnement flou [MEN95][JAN95].

Dans le raisonnement flou intervient trois niveaux fondamentaux :  
Implication floue, inférence floue et agrégation floue.

### 1.2.3.4.1. Implication floue

Les règles de la forme *SI-ALORS* qui manipulent des variables linguistiques (ou flou), sont appelées règles floues. L'utilisation des opérations des ensembles et de relations floues nous a permis précédemment d'associer des ensembles flous à des propositions floues complexes. Donc, il sera toujours possible d'écrire une règle de la forme :

$$SI X_1 \text{ est } F_1 \dots \text{ et } X_n \text{ est } F_n \text{ ALORS } ( Y \text{ est } B )$$

sous la nouvelle forme :

$$SI X \text{ est } F \text{ ALORS } Y \text{ est } B \quad (1.2.6)$$

où,  $X = [X_1 \dots X_n]$  vecteur des variables d'entrée.

$\underline{F}$  : ensemble flou défini sur l'univers de discours  $\prod_{i=1}^n U_{x_i}$

$$\underline{F} = \int_{\prod_{i=1}^n U_{x_i}} \mu_{\underline{F}}(x_1 \dots x_n) / (x_1 \dots x_n) = \int_{\prod_{i=1}^n U_{x_i}} \bigotimes_{i=1}^n \mu_{F_i}(x_i) / (x_1 \dots x_n) \quad (1.2.7)$$

Le sens de (1.2.7) est représenté par une relation d'implication floue définie sur l'univers de discours  $\prod_{i=1}^n U_{x_i} \times U_y$ . Cette relation mesure le degré de vérité de la règle floue, donc de l'implication entre antécédente et conséquence.

En logique classique une relation d'implication possède plusieurs représentations dont les résultats sont identiques. Considérons les deux variables logiques binaire  $p_1$  et  $p_2$ . La relation d'implication  $p_1 \Rightarrow p_2$  est équivalente aux deux relations  $\bar{p}_1 \vee p_2$  ou  $(p_1 \wedge p_2) \vee \bar{p}_1$ , avec  $\vee, \wedge, \bar{\quad}$ , désignent respectivement, les opérateurs *et, ou, complément* de la logique classique.

Sur la base de ces deux relations, nous avons introduit plusieurs définitions de l'implication floue, parmi lesquelles nous citons [DRI93] :

- **La norme de Klane-Dienes** : basé sur la relation  $\bar{p}_1 \vee p_2$ , l'opérateur « *et* » est réalisé par T-norme « *maximum* ». Elle est définie par la fonction d'appartenance suivante :

$$\mu_{F \rightarrow B}(x_1 \dots x_n, y) = \max(1 - \mu_F(x_1 \dots x_n), \mu_B(y))$$

- **La norme de Lukasiewicz** : basé sur la relation d'implication floue est donnée par la relation :

$$\mu_{F \rightarrow B}(x_1 \dots x_n, y) = \min(1, 1 - \mu_F(x_1 \dots x_n) + \mu_B(y))$$

- **La norme de Zadeh** : basée sur la relation  $(p_1 \wedge p_2) \vee p_2$ , l'opérateur *et* est utilisé par T-norme « *somme-bornée* », la relation d'implication floue est donnée par la relation :

$$\mu_{F \rightarrow B}(x_1 \dots x_n, y) = \max(\min(\mu_F(x_1 \dots x_n), \mu_B(y)), 1 - \mu_F(x_1 \dots x_n))$$

La littérature rapporte deux autres normes très utilisées : la norme de Mamdani et la norme de Larsen. En réalité, elles ne sont pas des implications mais plutôt des conjonctions, car elles traduisent d'une certaine manière une notion de progressivité de la forme :

$$\text{plus } x \text{ est } F \text{ plus } y \text{ est } B.$$

Elles sont définies par les relations suivantes :

- **La norme de Mamdani** :  $\mu_{F \rightarrow B}(x_1 \dots x_n, y) = \min(\mu_F(x_1 \dots x_n), \mu_B(y))$

- **La norme de Larsen :**  $\mu_{F \rightarrow B}(x_1, \dots, x_n, y) = \mu_F(x_1, \dots, x_n) \times \mu_B(y)$

### 1.2.3.4.2. Inférence Floue

La procédure utilisée pour déduire des conclusions à partir d'un ensemble de règles floues et d'une ou de plusieurs conditions est appelée *Inférence floue* (raisonnement flou, ou encore raisonnement approximatif) [JAN95].

En logique floue, il y a deux importantes règles d'inférence : Le Modus Tollens Généralisé "MTG" et le Modus Ponens Généralisé "MPG"

#### MTG :

condition :  $Y \text{ est } B'$   
 règle floue :  $SI X \text{ est } F \text{ ALORS } Y \text{ est } B$   
 ensemble de résultat :  $X \text{ est } F'$

#### MPG

condition :  $X \text{ est } F''$   
 règle floue :  $SI X \text{ est } F \text{ ALORS } Y \text{ est } B$   
 ensemble de résultat :  $Y \text{ est } B'$

### 1.2.3.4.3. Agrégation des règles

Lorsque la base de connaissance comporte plusieurs règles liées par un opérateur "ou", l'ensemble flou inféré  $B'$  est obtenu après une opération appelée *agrégation* des règles. En d'autres termes, l'agrégation des règles consiste à considérer des ensembles flous inférés par chaque règle pour en créer un autre, représentatif de l'ensemble des règles.

Considérons les règles sous la forme symbolique suivante :

$$SI (X_1 \text{ est } F_1^{(l)} \text{ et } X_2 \text{ est } F_2^{(l)} \dots \text{et } X_n \text{ est } F_n^{(l)}) \text{ ALORS } (Y \text{ est } B^{(l)}), \quad l = 1, \dots, k$$

L'inférence de chacune de ces règles avec la condition :

$$X_1 \text{ est } F_1 \text{ et } X_2 \text{ est } F_2 \dots \text{et } X_n \text{ est } F_n$$

produit des ensembles inférés :  $B^{(l)}, \quad l = 1, \dots, k.$

Si les différentes règles sont liées par des connecteurs "ou", l'agrégation des ensembles inférés est réalisée par un S-norme. Après agrégation, on obtient l'ensemble flou donné par :

$$\underline{B'} = \int_{U_y} \bigoplus_{l=1}^k (\mu_{B^{(l)}}(y)) / y \quad (I.2.8a)$$

En remplaçant (I.17b) dans l'expression de  $\underline{B'}$ , on obtient :

$$\underline{B'} = \int_{U_y} \bigoplus_{l=1}^k \text{Sup}_{(x_1, \dots, x_n)} \left[ \left( \bigotimes_{i=1}^n \mu_{F_i}(x_i) \right) \otimes \left( \left( \bigotimes_{i=1}^n \mu_{F_i^{(l)}}(x_i) \right) \otimes \mu_{B^{(l)}}(y) \right) \right] / y \quad (I.2.8b)$$

### 1.2.3.5. Mécanisme d'inférence

D'après l'expression (I.2.8b), l'ensemble inféré dépend essentiellement des opérateurs de composition adoptés. Vu la richesse et la diversité des opérateurs possibles, on introduit la notion de méthode ou mécanisme d'inférence. Elle détermine la réalisation des différents opérateurs "et", "ou" et "implication" dans une inférence, permettant ainsi un traitement numérique de cette dernière.

Les mécanismes les plus utilisés en commande à base de logique floue, sont : [BUH94]

- A. Méthode d'inférence maximum-minimum "Max-min";
- B. Méthode d'inférence maximum-produit "Max-pro";
- C. Méthode d'inférence somme-produit "Som-pro"

Afin de mettre en évidence les différentes étapes du déroulement de traitement numérique, on considère l'ensemble des deux règles suivantes :

Si  $X_1$  est  $PG$  et  $X_2$  est  $EZ$  alors  $Y$  est  $EZ$      ou

Si  $X_1$  est  $EZ$  ou  $X_2$  est  $NG$  alors  $Y$  est  $NG$

avec la condition :

$X_1$  est  $F'_1$  est  $X_2$  est  $F'_2$

où,  $F'_i(x_i) = \{x_i^*\}$  ;  $i = 1, 2$  ; avec  $x_1^* = 0.5$ ,  $x_2^* = -0.75$

Les ensembles flous  $NG$ ,  $EZ$  et  $PG$  sont illustrées par la figure (I.2.3).

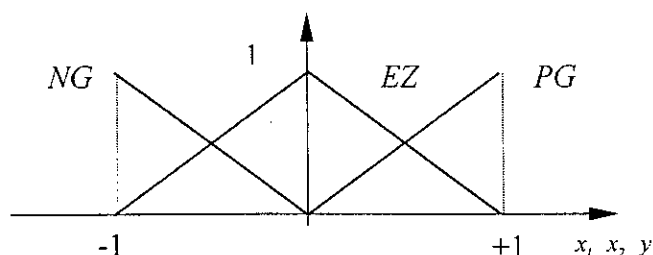


Figure I.2.3. Ensembles flous associés à  $NG$ ,  $EZ$  et  $PG$

Il faut noter que les propositions floues simples de l'antécédente de la deuxième règle sont liées par l'opérateur "ou" et non pas par l'opérateur "et". Donc, l'ensemble flou inféré de cette règle ne peut être calculé directement à partir de la relation (I.2.8b)

#### 1.2.3.5.1. Méthode d'inférence Max-min

Dans la méthode Max-min, on adopte les choix suivants :

- Le connecteur "et" est réalisé par le T-norme : "minium"
- Le connecteur "ou" est réalisé par le S-norme : "maximum"
- L'implication est réalisée par le T-norme : "minimum"

Dans ce cas l'expression de l'ensemble inféré (I.2.8b) devient :

$$\mu_{B'}(y) = \max_{l=1..k} \left[ \text{Sup}_{(x_1, \dots, x_n)} \left( \min \left( \mu_{F_1^l}(x_1), \dots, \mu_{F_n^l}(x_n), \mu_{F_1^l}(x_1), \dots, \mu_{F_n^l}(x_n), \mu_{B^l}(y) \right) \right) \right] \quad (I.2.9a)$$

Dans le cas particulier où  $F_i^l(x) = \{x_i^*\}$ ,  $i = 1, \dots, n$  :

$$\mu_{B'}(y) = \max_{l=1..k} \left[ \min \left( \mu_{F_1^l}(x_1^*), \dots, \mu_{F_n^l}(x_n^*), \mu_{B^l}(y) \right) \right] \quad (I.2.9b)$$

### I.2.3.5.2. Méthode d'inférence Max-prod

Dans cette méthode,

- Le connecteur "et" est réalisé par le T-norme "minimum"
- Le connecteur "ou" est réalisé par le S-norme "maximum"
- L'implication est réalisée par le T-norme produit. ( norme de Mamdani ).

L'expression de l'ensemble inféré devient :

(I.2.10.a)

Si  $F_i(x) = \{x_i^*\}$ ,  $i = 1, \dots, n$  on obtient :

$$\mu_{B'}(y) = \max_{l=1..k} \left[ \min \left( \mu_{F_1^l}(x_1^*), \dots, \mu_{F_n^l}(x_n^*) \right) \times \mu_{B^l}(y) \right] \quad (I.2.10.b)$$

### I.2.3.5.3. Méthode d'inférence Som-prod

Dans cette méthode, on adopte le choix suivant :

- Le connecteur "et" est réalisé par le T-norme "produit"
- Le connecteur "ou" est réalisé par le S-norme "somme moyenne"
- L'implication est réalisée par le T-norme "produit".

L'expression de l'ensemble inféré s'écrit :

$$\mu_{B'}(y) = \sum_{l=1}^k \text{Sup}_{(x_1, \dots, x_n)} \left[ \left[ \mu_{F_1^l}(x_1) \times \dots \times \mu_{F_n^l}(x_n) \right] \times \left[ \left( \mu_{F_1^l}(x_1) \times \dots \times \mu_{F_n^l}(x_n) \right) \times \mu_{B^l}(y) \right] \right] / k \quad (I.2.11.a)$$

Dans le cas particulier  $F_i(x) = \{x_i^*\}$ ,  $i = 1, \dots, n$  on obtient :

$$\mu_{B'}(y) = \sum_{l=1}^k \left[ \mu_{B^l}(y) \times \left[ \prod_{i=1}^n \mu_{F_i^l}(x_i^*) \right] \right] / k \quad (I.2.11.b)$$

## I.2.4. Application de la logique floue à la commande

L'utilisation de la logique floue dans la commande a créé une nouvelle stratégie permettant d'exploiter les connaissances subjectives, ou linguistiques. Ces dernières sont représentées sous forme de règles linguistiques conditionnelles. La théorie des ensembles flous, introduite par Zadeh en 1965, peut être vue comme un moyen de traduire ces connaissances par un ensemble de règles floues à partir desquelles on va inférer la commande à appliquer au processus.

Le figure (I.2.4) montre un régulateur à base à logique floue (ou régulateur flou) dans une boucle de commande. Son schéma synoptique est illustré à la figure (I.2.5)[DRI93].

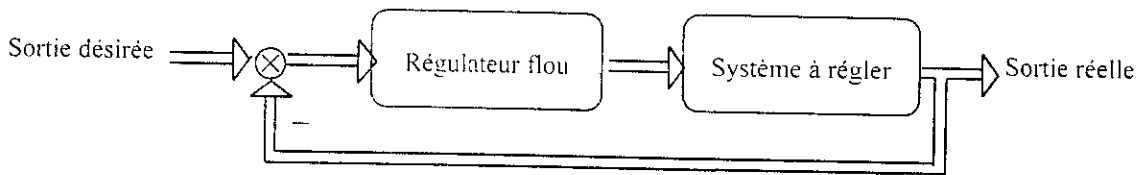


Figure I.2.4. Schéma d'un système bouclé avec régulateur flou

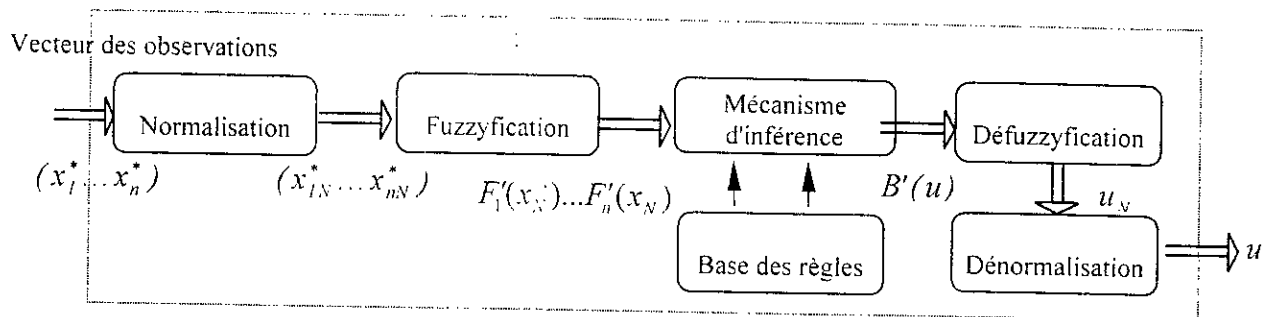


Figure I.2.5 : Schéma d'un régulateur flou

### I.2.4.1. Description des éléments d'un régulateur flou

#### I.2.4.1.1. Module de normalisation

Le vecteur d'entrée du module de normalisation  $(x_1^* \dots x_n^*)$ , qu'on appellera vecteur d'observations, est de composantes dépendantes des états du processus.

Ce module transforme les valeurs physiques actuelles du vecteur d'observation dans un intervalle normalisé, souvent entre -1 et 1. Une transformation linéaire est d'expression :

$$x_{iN}^* = \frac{x_i^*}{N_{x_i}} \quad i = 1, \dots, n \quad (\text{I.2.13})$$

où,  $(x_{iN}^*)_{i=1 \dots n}$  sont les variables d'observations normalisées.

$N_{x_i}$  : facteur de normalisation.

#### I.2.4.1.2. Module de fuzzyfication

Ce module d'interfaçage permet d'attribuer à chaque composante du vecteur d'observation normalisée un ensemble flou. Cette dernière dépendra de la valeur actuelle de la variable d'observation normalisée  $(x_{iN}^*)_{i=1 \dots n}$

Dans le but de simplifier les calculs dans les modules situés en amont, on choisit souvent des fonctions d'appartenances de type singleton. Elles sont définies par :

$$F_i(x) = \{x_i^*\} = \begin{cases} 1 & \text{si } x_i = x_i^* \\ 0 & \text{si } x_i \neq x_i^* \end{cases} \quad i = 1, \dots, n$$

En présence de bruit dans les mesures  $x_i^*$ , ce choix est déconseillé [DRI93][MEN95]. Il sera préférable de choisir des fonctions d'appartenance non-singleton (triangulaire, trapézoïdale...). Cela permettra d'introduire un préfiltrage sur les mesures entachées de bruits [MEN95]. La figure(I.2.6) montre deux types de fuzzyfication : singleton et non-singleton.

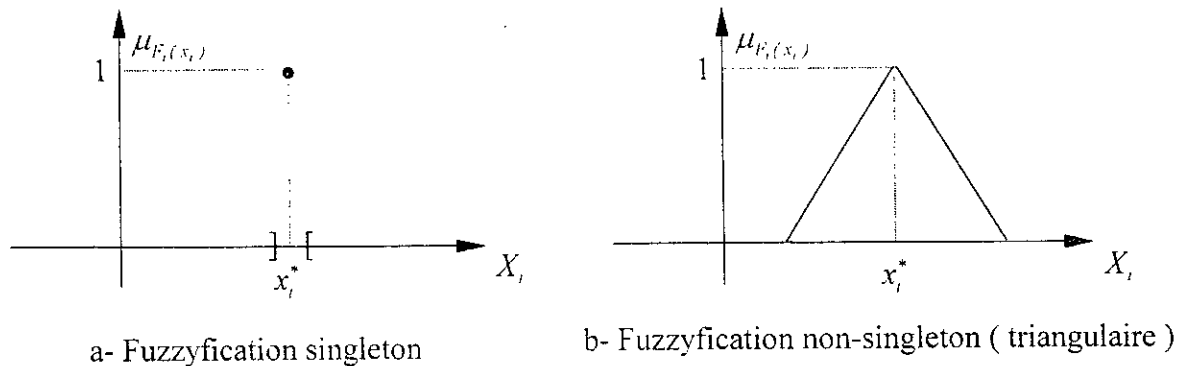


Figure I.2.6. Exemples de Fuzzyfication

### I.2.4.1.3. Module de la base des règles

#### a- Base de données liées aux règles

La façon d'agir que doit réaliser un régulateur flou est stockée sous forme de règles "SI-ALORS" liées par des opérateurs "ou". Souvent, ces règles s'écrivent comme suit :

$$\text{Si } (X_1 \text{ est } F_1^{(l)} \text{ et } X_2 \text{ est } F_2^{(l)} \dots \text{et } X_n \text{ est } F_n^{(l)}) \text{ ALORS } (U \text{ est } B^{(l)}) \quad l = 1, \dots, k$$

L'établissement de l'ensemble de ces règles, nécessite la détermination de :

- 1- La variable de sortie ;
- 2- Les valeurs linguistiques des variables d'observation et de sortie.

La sortie ou l'action de commande peut se traduire soit par une commande à appliquer directement au processus, soit par une variation de la commande.

Le nombre total des règles ( $nr$ ) est lié aux nombres des classes des variables d'observation  $nf_i$ .

par la relation suivante :

$$nr = \prod_{i=1}^n nf_i$$

où,  $nf_i$  : est le nombre de classes de la variable d'observation  $X_i$ .

Dans les régulateurs flous, on choisit souvent, un nombre de classes égale à 3, 5 ou 7, pour un régulateur à deux variables d'observations, on aura respectivement 9, 25, 49 règles. Des distributions équidistantes en cas de 3, 5 et 7 classes, sont montrées à la figure I.2.7.



#### 1.2.4.1.4. Module d'inférence

Les deux principales fonctions de ce module sont :

1. Inférer les ensembles flous représentatifs de la valeur actuelle du vecteur d'observation (issu du module de fuzzyfication) avec chaque règle.
2. Agrégation des différents ensembles inférés des différentes règles pour déduire un seul ensemble.

#### 1.2.4.1.5. Module de défuzzyfication

L'ensemble inféré créé par le module d'inférence ne peut être utilisé comme un signal de commande, car celui-ci présente une valeur réelle est non pas un ensemble flou. D'où la nécessité d'un module d'interface. Ce dernier permettra de déterminer une valeur réelle de la commande à partir de l'ensemble inféré. Cette opération est appelée *Défuzzyfication*.

Parmi les méthodes de défuzzyfication les plus utilisées, nous citons [LOU97] : Méthode du maximum, méthode de la moyenne des maxima, méthode du centre de gravité et méthode des hauteurs.

##### - Méthode du maximum

Dans cette méthode, le signal de commande normalisé est pris égale à la commande locale au point où la distribution de la fonction d'appartenance de l'ensemble flou inféré est maximale.

##### - Méthode de la moyenne des maxima

Elle consiste à prendre la moyenne des commandes associées aux points du maximum de l'ensemble inféré.

##### - Méthode du centre de gravité

Dans la méthode du centre de gravité, la commande est prise égale à la commande correspondante au centre de gravité de la surface au-dessous de la fonction d'appartenance de l'ensemble inféré. On utilise la relation :

$$u_N = \frac{\int_{u_a}^{u_b} u \mu_{B'}(u) du}{\int_{u_a}^{u_b} \mu_{B'}(u) du} \quad (I.2.13a)$$

Dans le cas particulier d'un fuzzyficateur singleton, et un mécanisme d'inférence de type "Som-prod", la fuzzyfication par la méthode du centre de gravité conduit à une expression simple. D'après les équations (I.2.11.b) et (I.2.13), la commande est donnée par :

$$u_N = \frac{\sum_{l=1}^k C^{(l)} S^{(l)} \left( \prod_{i=1}^n \mu_{F_i^{(l)}}(x_{iN}^*) \right)}{\sum_{l=1}^k S^{(l)} \left( \prod_{i=1}^n \mu_{F_i^{(l)}}(x_{iN}^*) \right)} \quad (I.2.13.b)$$

$C^{(l)}$  : la commande au point du centre de gravité de l'ensemble flou associé à la conséquence de la règle "l" :

$$C^{(l)} = \int_{U_u} u \mu_{B^{(l)}}(u)$$

$S^{(l)}$  : la surface au-dessous de la fonction d'appartenance de l'ensemble flou associé à la conséquence de la règle "l"

$$S^{(l)} = \int_{U_u} \mu_{B^{(l)}}(u) du$$

### - Méthode des hauteurs

Cette méthode utilise les ensembles flous inférés des différentes règles pour en déduire un signal de commande normalisé. Donc, l'opération d'agrégation réalisée par le mécanisme d'inférence est ignorée.

Dans la méthode des hauteurs, la commande défuzzyfiée est la moyenne pondérée des commandes partielles évaluées aux centres de gravité  $C^{(l)}$  des ensembles flous inférés des différentes règles ( $C^{(l)}$ ). La pondération est effectuée par les degrés d'appartenance à des ensembles inférés correspondants aux centres de gravités  $\mu_{B^{(l)}}(C^{(l)})$ . L'expression de la commande est :

$$u_N = \frac{\sum_{l=1}^k C^{(l)} \times \mu_{B^{(l)}}(C^{(l)})}{\sum_{l=1}^k \mu_{B^{(l)}}(C^{(l)})} \quad (I.2.14)$$

La figure (I.2.8) montre une représentation graphique de la méthode des hauteurs.

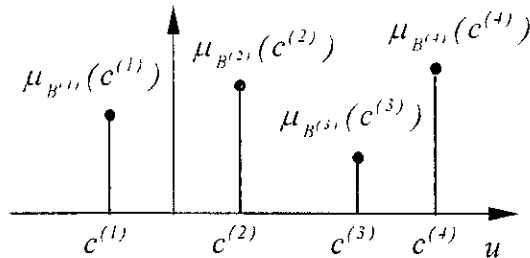


Figure (I.2.8) : Représentation graphique des hauteurs

#### 1.2.4.1.6. Dénormalisation

Généralement, le module de défuzzyfication délivre un signal de commande normalisé  $u_N$ , qui appartient à l'intervalle  $[-1, 1]$ . L'opérateur de dénormalisation transforme le signal de commande du domaine normalisé au domaine physique.

#### 1.2.4.2 Modèles flous

En se basant sur les notions de la théorie des ensembles flous, les systèmes d'inférence flous ont constitué un moyen efficace pour exploiter les deux types de connaissances (linguistique et numérique). Grâce à la propriété d'approximation des fonctions, les systèmes à logique floue permettent la modélisation de ces connaissances sous forme de modèles flous. Nous présentons ici deux types de modèles flous les plus utilisés en commande.

Selon la nature de la conséquence, et des règles floues utilisées, on distingue les modèles suivants :

##### a-Modèle de Mamdani

Dans ce modèle les conséquences des règles sont de type linguistique sous la forme :

$$SI (X_1 \text{ est } F_1^{(l)} \dots \text{ et } X_n \text{ et } F_n^{(l)}) \text{ ALORS } (Y \text{ est } B^{(l)}) \quad l=1 \dots k \quad (1.2.15)$$

Ce type de modèle est celui étudié jusqu'à maintenant. L'expression de sa sortie est obtenue en appliquant une des méthodes de défuzzyfication sur l'expression de l'ensemble inféré donnée par (1.2.11.b).

##### b-Modèle de Sugeno

Originellement développé par Takagi-Sugeno [SUG85], il était un effort pour développer une méthode systématique de génération des règles floues à partir d'un ensemble de données entrée-sortie. Les règles floues sont de la forme :

$$SI (X_1 \text{ est } F_1^{(l)} \dots \text{ et } X_n \text{ et } F_n^{(l)}) \text{ ALORS } (Y = f_l(x_1 \dots x_n)) \quad l=1 \dots k$$

Dans ce cas les conséquences des règles contiennent des fonctions numériques  $(f_l)_{l=1 \dots k}$ , qui dépendent des valeurs actuelles des variables d'observations  $(x_i)_{i=1 \dots n}$ . Les fonctions "f<sub>l</sub>" peuvent être choisies parmi les fonctions convenable pour la modélisation du système considéré dans les régions définies par les ensembles flous de l'antécédente de la règle "l".

A partir de l'ensemble des règles précédentes, l'expression de la sortie réalisée par un modèle flou de type Sugeno est d'expression suivante :

$$y(x_1, \dots, x_n) = \frac{\sum_{l=1}^k \left[ \prod_{i=1}^n \mu_{F_i^{(l)}}(x_i) \right] \times f_l(x_1, \dots, x_n)}{\sum_{l=1}^k \left[ \prod_{i=1}^n \mu_{F_i^{(l)}}(x_i) \right]} \quad (I.2.16)$$

Un modèle de type Sugeno peut être considéré comme un cas particulier des modèles de type Mamdani. En effet, si on considère les règles de Mamdani données par (I.2.15) où les ensembles flous des conséquences sont définis par des singletons tels que :

$$B^{(l)} = \{f_l(x_1, \dots, x_n)\}$$

Si on adopte un flouificateur singleton, un mécanisme d'inférence *Som-prod* et défuzzyfication par la méthode du centre de gravité, on obtient l'expression de sortie donnée par (I.2.16).

Un cas particulier, consiste à choisir  $(f_l)$ ,  $l=1, \dots, k$  des fonctions polynomiales d'ordre  $l$ . Le modèle ainsi obtenu est appelé *modèle de Sugeno d'ordre l*. Les fonctions  $(f_l)$   $l=1, \dots, k$  sont définies par :

$$f_l(x_1, \dots, x_n) = a_{0l} + \sum_{i=1}^n a_{il} x_i \quad l=1, \dots, k \quad (I.2.17a)$$

Où,  $(a_{ij})_{\substack{i=0, \dots, n \\ j=1, \dots, k}}$  représentent les paramètres des polynômes.

Posons :  $f = [f_1 \dots f_k]$ , alors sous forme matricielle, on écrit :

$$f = z^T A \quad z^T = [1 \ x_1 \ \dots \ x_n] \quad A = \begin{bmatrix} a_{01} & \dots & \dots & a_{0l} \\ a_{11} & \dots & \dots & a_{1l} \\ \vdots & \dots & \dots & \vdots \\ a_{n1} & \dots & \dots & a_{nl} \end{bmatrix} \quad (I.2.17b)$$

d'après (I.2.16), (I.2.17a) et (I.2.17.b), l'expression de la sortie  $y$  s'écrit :

$$\begin{cases} y = z^T A \xi \\ \xi^T = [\xi_1 \dots \xi_k] \end{cases} \text{ avec } \xi_j = \frac{\prod_{i=1}^n \mu_{F_i^{(j)}}(x_i)}{\sum_{l=1}^k \prod_{i=1}^n \mu_{F_i^{(l)}}(x_i)} \quad j=1, \dots, k \quad (I.2.18.a)$$

Si le polynôme est d'ordre 0, on obtient un modèle de Sugeno d'ordre 0. L'expression de la sortie  $y$  devient :

$$\begin{cases} y = z^T A \xi \\ z^T = [1] \\ A = [a_{01} \dots a_{0l}] \\ \xi^T = [\xi_1 \dots \xi_k] \end{cases} \text{ avec } \xi_j = \frac{\prod_{i=1}^n \mu_{F_i^{(j)}}(x_i)}{\sum_{l=1}^k \prod_{i=1}^n \mu_{F_i^{(l)}}(x_i)} \quad j=1, \dots, k \quad (I.2.18.b)$$

Selon les expressions de sorties (I.2.18.a) et (I.2.18.b), le choix des fonctions polynomiales permet d'obtenir un paramétrage linéaire par rapport aux paramètres des polynômes. Cette propriété intéressante sera exploitée dans les chapitres III pour le développement de l'algorithme de commande floue adaptative.

### I.3. Réseaux de neurones artificiels

Au cours des dernières années, l'une des évolutions les plus marquantes des réseaux de neurones formels était l'abandon de la métaphore biologique au profit de fondements théoriques solides dans le domaine des statistiques; on sait à présent que la propriété fondamentale des réseaux de neurones est l'approximation universelle parcimonieuse. De plus, le développement d'algorithmes performants pour l'entraînement et l'apprentissage des réseaux de neurones bouclés ou non bouclés, leur a ouvert de nouvelles perspectives d'utilisation. En particulier, les réseaux de neurones se sont avérés parfaitement adaptés dans le domaine de l'automatique comme éléments du système de commande de processus dynamiques non linéaires. Des travaux récents ont permis de replacer la mise en œuvre des réseaux de neurones comme modèle de processus ou correcteurs des systèmes dans le cadre général de la commande classique.

Dans cette partie nous donnerons une description très succincte des réseaux de neurones artificiels (RNA). Ainsi, nous présenterons seulement les notions utilisées dans les développements introduits aux chapitres IV et V. Pour plus de détails le lecteur pourra consulter les références spécialisées, tel que [FRE92],[KOS92],[KAR93][MIL90].

#### I.3.1. Le neurone artificiel

Un neurone artificiel est défini comme une unité fonctionnelle caractérisée par une fonction d'activation  $f$  décrivant l'état de sa sortie  $y$  en fonction de ses entrées  $x_i$ . (figure I.3.1).

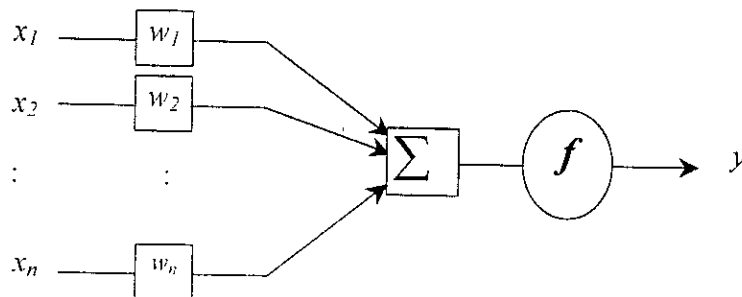


Figure I.3.1. Le modèle artificiel d'un neurone.

Ainsi, la sortie du neurone sera :

$$y = f\left(\sum_{i=1}^n w_i x_i + b\right) \quad (I.3.1)$$

où  $w_i$  sont les poids de pondération des connexions et  $b$  est le paramètre de biais.

La fonction non linéaire  $f$  permet donc la transformation du signal d'entrée  $x_i$  non borné, en un signal de sortie borné. Généralement, cette fonction est une sigmoïde (figure I.3.2), décrite par l'équation :

$$f(x) = \frac{1}{1 + \exp(-ax)} \quad (I.3.2)$$

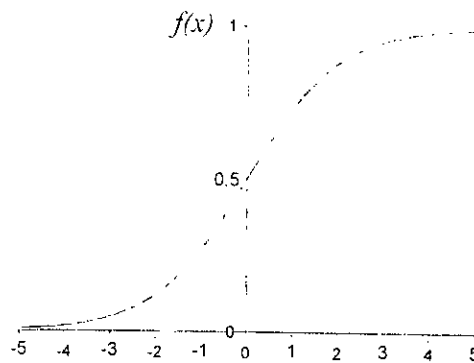


Figure I.3.2. La fonction de type sigmoïde

Un réseau de neurones est l'interconnexion pondérée de plusieurs neurones considérés comme éléments de base. Ce réseau effectue une application de l'espace des entrées dans l'espace des sorties d'une façon parallèle et distribuée. On peut décomposer un réseau de neurones en plusieurs champs de neurones, appelés couches. On distingue alors : la couche d'entrée, les couches cachées et la couche de sortie.

Plusieurs architectures de réseaux sont envisagées selon la topologie des connexions des neurones ainsi que la forme des fonctions d'activation des neurones. Parmi d'autres, on peut citer ; les réseaux statiques, les réseaux dynamiques, les réseaux récurrents et les réseaux de fonctions à bases radiales. Ici nous nous limitons aux réseaux statiques que nous allons utiliser dans les chapitres IV et V.

### I.3.2 Les réseaux de neurones statiques

Un RNA statique est caractérisé par la topologie de ses connexions qui relient les différentes couches du réseau et qui ne contient pas de boucles synaptiques (neuronale) fermées. Les fonctions d'activation des neurones sont généralement sigmoïdales ou de type tangente hyperbolique; néanmoins, elles peuvent prendre d'autres formes comme la fonction *sat* (*saturation*), ou la fonction *sgn* (*signe*) selon l'utilisation du réseau.

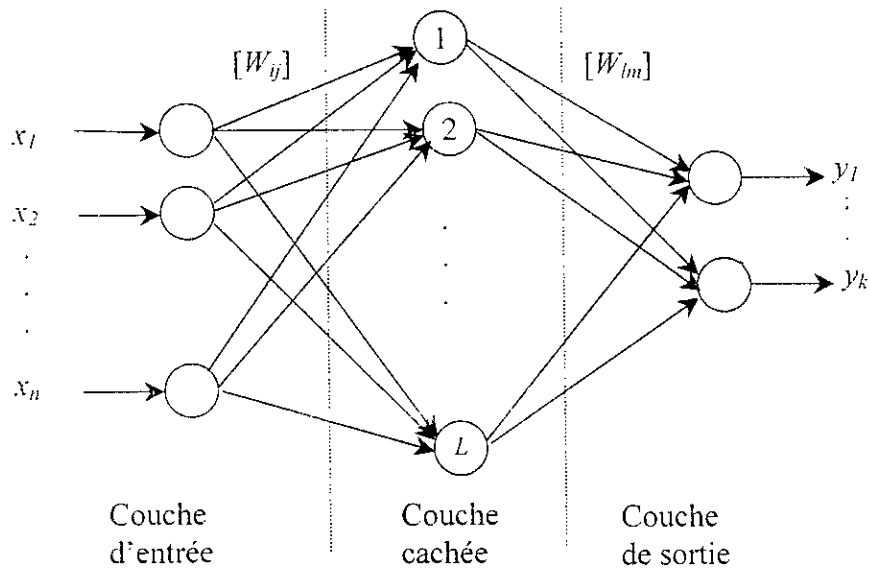


Figure I.3.3 Architecture d'un RNA statique.

La sortie du réseau sera

$$y_m = \sum_{l=1,L} w_{lm} \cdot f \left( \sum_{\substack{i=1,L \\ j=1,n}} w_{ij} \cdot x_j + b_l \right) + b_2 \quad (I.3.3)$$

ou sous une écriture matricielle :

$$y = W_{kL} \cdot f(W_{Ln} \cdot X_n + b1) + b2 \quad (I.3.4)$$

### I.3.3. Apprentissage des réseaux de neurones artificiels.

L'apprentissage des réseaux de neurones est l'ensemble des algorithmes d'ajustement de ses poids, pour lui permettre une application entre l'espace d'entrée qu'on lui présente et l'espace de sortie correspondant.

Trois classes d'apprentissages sont à distinguer :

**L'apprentissage supervisé :** disposant de deux espaces entrée/sortie, le réseau peut être entraîné, en ajustant ses paramètres systématiquement, à approximer au mieux les sorties désirées correspondantes à chaque vecteur d'entrée en se basant sur le principe *essai - erreur*.

**L'apprentissage non supervisé :** ce mode d'apprentissage est souvent utilisé lors du traitement des signaux stochastiques, où on ne dispose que de très faibles informations sur l'espace d'entrée. Le but de cet apprentissage est d'équiper le réseau du pouvoir de classification des données. ce qu'on appelle parfois « la séparation aveugle des sources ». Ainsi, les données

de la même classe vont, à l'issue de l'apprentissage, se retrouver réunies dans des groupes l'algorithme de base suivant est celui de la *Backpropagation*.

### - L'algorithme de Backpropagation

Cet algorithme est l'un des plus utilisé pour l'ajustement des paramètres des réseaux de neurones et conduire ces derniers à savoir reconnaître les sorties désirées, en se basant sur la minimisation du critère quadratique  $J$  par la méthode du gradient [ FRE92 ].

$$J = \frac{1}{2} e^2 = \frac{1}{2} (y - \hat{y})^2 \quad (I.3.5)$$

Plusieurs variantes ont été dérivées pour accélérer sa convergence et améliorer ses performances. Dans ce qui suit, nous allons présenter le formalisme de l'algorithme appliqué sur un réseau de neurones à  $n$  entrées,  $m$  sorties et  $L$  couches cachées (figure I.3.3).

Disposant d'un ensemble d'apprentissage composé de  $k$  paires de vecteurs :  $(X_1, Y_1), (X_2, Y_2), \dots, (X_p, Y_p)$ , avec :

$X_p = (x_{p1}, x_{p2}, \dots, x_{pn}) \in \mathbb{R}^n$  un vecteur d'entrée

$Y_p = (y_{p1}, y_{p2}, \dots, y_{pn}) \in \mathbb{R}^m$  le vecteur de sortie correspondant désiré.

On note par

$W_{ij}$  : les poids des connexions entre les neurones  $i$  et  $j$ .

$net_{pj}^h$  : l'entrée totale du neurone  $j$  de la couche cachée  $h$  pour l'exemple  $p$ .

$net_{pj}^o$  : l'entrée totale du neurone  $j$  de la couche de sortie  $o$  pour l'exemple  $p$ .

$b_j^h$  : la polarisation du neurone  $j$  de la couche cachée  $h$ .

$f$  : la fonction d'activation.

$\mu$  : le pas d'apprentissage (ou pas d'adaptation).

On veut entraîner le réseau, par l'algorithme de Backpropagation, à approximer la fonction  $Y=F(X)$  sur tout l'ensemble d'apprentissage. Pour cela, on procède comme suit :

- 1- Appliquer un vecteur d'entrée  $X_p$  à l'entrée du réseau.
- 2- Calculer les entrées totales des neurones de la couche cachée :

$$net_{pj}^h = \sum_{i=1}^n W_{ji}^h x_{pi} + b_j^h \quad (I.3.6)$$

- 3- Calculer les sorties des neurones de la couche cachée :



$$I_{pj} = f_j^h(\text{net}_{pj}^h) \quad (I.3.7)$$

4- Calculer les entrées totales des neurones de la couche de sortie :

$$\text{net}_{pj}^o = \sum_{k=1}^L W_{kj}^o I_{pk} + b_k^o \quad (I.3.8)$$

5- Calculer les sorties réelles du réseau :

$$\hat{y}_{pk} = f_k^o(\text{net}_{pk}^o) \quad (I.3.9)$$

6- Calculer les termes d'erreur sur chaque unité de sortie :

$$e_{pk}^o = (y_{pk} - \hat{y}_{pk}) f_k^o{}'(\text{net}_{pk}^o) \quad (I.3.10)$$

7- Calculer les termes d'erreur sur chaque unité cachée :

$$e_{pj}^h = f_j^h{}'(\text{net}_{pj}^h) \sum_{k=1}^m e_{pk}^o W_{kj}^o \quad (I.3.11)$$

8- Ajuster les poids de la couche de sortie :

$$W_{kj}^o = W_{kj}^o + \mu e_{pk}^o I_{pj} \quad (I.3.12)$$

9- Ajuster les poids de la couche cachée :

$$W_{ji}^h = W_{ji}^h + \mu e_{pj}^h x_{pi} \quad (I.3.13)$$

10- Test d'arrêt

$$E_p = \frac{1}{2} \sum_{k=1}^m (e_{pk}^o)^2 \leq \text{erreur exigée.} \quad (I.3.14)$$

### 1.3.4. Les réseaux de neurones et l'approximation de fonctions

Les données présentées au réseau lors de l'apprentissage sont généralement limitées et sous forme d'échantillons représentatifs de l'espace d'entrée. Le rôle essentiel du réseau est de

pouvoir interpoler et extrapoler ces données, en établissant une fonction non linéaire qui recouvre tout l'espace d'entrée, et savoir reconstruire les liens entre de nouvelles entrées dans le même espace et leurs sorties correspondantes; ce qui est appelé « pouvoir de généralisation ».

Le caractère non linéaire des réseaux de neurones les prédispose à être les candidats naturels pour l'approximation de fonctions non linéaires, mais ils approximent aussi bien les fonctions linéaires.

L'emploi des réseaux de neurones plutôt que des techniques classiques pour l'approximation des fonctions peut se justifier par les avantages suivants :

- simplicité de mise en œuvre ( peu d'analyses mathématiques préliminaires ).
- capacité d'approximation universelle prouvée.
- possibilité de prendre le point de vue « processus = boîte noire ».
- robustesse par rapport à des défaillances internes du réseau.
- capacité d'adaptation.

Parfois, en voulant assurer la capacité d'approximation universelle, on est amené à augmenter le nombre de neurones dans le réseau ; se faisant, on observe une diminution des performances de généralisation à cause du phénomène de sur-paramétrisation. Un compromis est alors toujours à faire. Ceci constitue l'inconvénient principal des réseaux de neurones.

### 1.3.5. . Les réseaux de neurones et la commande

Vu le caractère non linéaire des réseaux de neurones, et grâce au fait qu'ils possèdent la capacité d'auto-organisation et de stockage implicite d'informations ainsi que d'autres propriétés, les automaticiens ont eu recours à l'utilisation des réseaux de neurones en identification et en commande des systèmes dynamiques.

Dans le cadre de la commande, un RNA est considéré comme un ensemble structuré d'unités agissant d'une façon distribuée et parallèle sur le processus dans le but de lui assurer une conduite souhaitable face à sa structure caractérisée par des non-linéarités, dimensionnalités élevées (multivariables), erreurs de modélisation, bruits de mesures ...

L'apprentissage peut se voir comme un moyen de synthétiser automatiquement une loi de commande sur la base du principe *essais - erreurs*. Si l'on considère plus particulièrement l'aspect *on line* de l'apprentissage, ce sont surtout les facultés d'optimisation des performances et l'adaptation à un environnement incertain qu'il faut souligner.

Parmi les stratégies de commandes par réseaux de neurones on peut citer:

- **La commande supervisée:** Comme son nom l'indique, cette technique de commande est basée sur l'utilisation d'un régulateur existant déjà (figure I.3.). La sortie de ce régulateur sera une fonction à approximer par le réseau. Une fois cette tâche accomplie, le réseau sera implémenté à la place de l'ancien régulateur pour conduire le système en imitant son

comportement. Cela se fait dans le but de pouvoir généraliser le comportement de la commande pour des cas qui ne se sont pas déjà présentés.

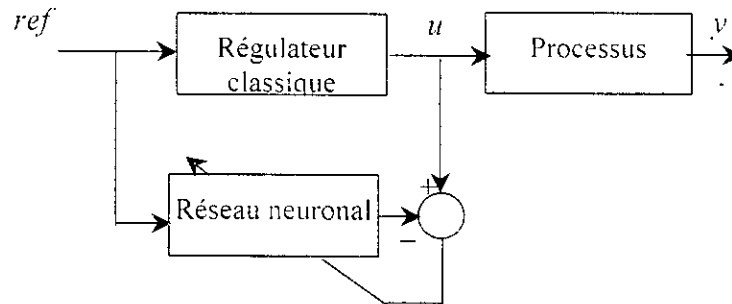


Figure I.3.4. Schéma de commande supervisée (phase d'apprentissage)

- **La commande inverse:** Cette approche procède en deux étapes successives; la première consiste à identifier le modèle inverse du processus en injectant la sortie du processus, résultant d'une certaine entrée  $u(t)$ , comme entrée du réseau qui sera entraîné par rétropropagation afin de produire la sortie la plus proche possible de  $u(t)$ . Comme étape de commande, le réseau sera placé devant le processus pour le conduire en boucle ouverte (figure I.3.5).

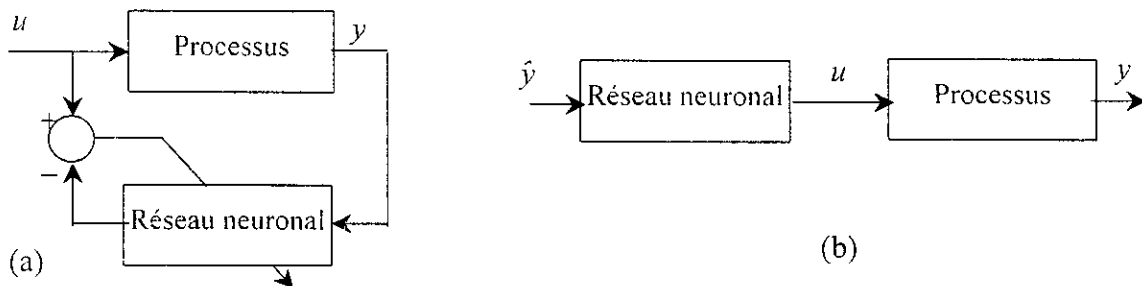


Figure I.3.5. Schéma de la commande inverse (a) phase d'apprentissage. (b) phase de commande.

- **La commande par identification neuronale du modèle direct:** C'est une commande indirecte qui consiste à identifier le modèle du processus et puis chercher une stratégie de commande en supposant que ce modèle coïncide parfaitement avec le processus réel « principe d'équivalence certaine ».

- **La commande par apprentissage spécialisé:** Dans cette approche, le réseau conduit directement le système en utilisant l'erreur de sortie (ou un indice de performance) pour ajuster ses poids en ligne, ce qui est généralement adopté dans la commande adaptative classique. On remarque qu'il ne faut pas nécessairement une phase d'apprentissage préliminaire, l'adaptation est continue en se focalisant directement sur le domaine d'intérêt.

#### I.4. Conclusion

Dans ce chapitre, nous avons présenté des généralités sur les différentes notions nécessaires à la compréhension des méthodes développées dans les chapitres suivants. En effet, ce chapitre est organisé en trois parties.

Dans une première partie, nous avons donné les principes de base des systèmes de commande à structure variable. Ces systèmes sont connus par la nature discontinue de leurs signaux de commande. En effet ces systèmes sont restés à l'étape théorique pendant plusieurs années à cause de la non disponibilité des composants permettant de les mettre en pratique. C'est avec le progrès technologique dans le domaine de l'électronique de commutation qu'ils ont connus un succès considérable dans diverses applications industrielles. A la base des notions théoriques introduites, nous allons proposer, dans le chapitre suivant une loi de commande à structure variable pour la commande décentralisée des robots manipulateurs.

Dans une deuxième partie, nous avons introduit la notion de logique floue et son application dans le domaine de la commande. La nature linguistique des variables manipulées permet d'exploiter l'expérience de l'opérateur humain dans l'élaboration des stratégies de commande. Deux types de contrôleurs flous sont présentés, le régulateur de Mamdani et celui de Sugeno. Dans le régulateur de Mamdani, les prémisses et les conclusions sont des variables floues, tandis que dans celui de Sugeno les entrées sont des variables linguistiques par contre les sorties sont des variables numériques. Grâce à l'expression analytique de la sortie de ce dernier type, les problèmes d'analyse tels que la stabilité des systèmes de commande pourront être abordés. En effet, le modèle de Sugeno sera utilisé dans le chapitre III pour le développement d'un schéma de commande adaptative décentralisée à base de logique floue.

La dernière partie, a été consacrée à une description des réseaux de neurones artificiels où nous avons mis en évidence leurs structures, leurs caractéristiques ainsi que leur application dans le domaine de la commande. Leur capacité d'approximation de fonctions non linéaires leur a ouvert un grand champ d'application, particulièrement, dans les problèmes de commande des systèmes non linéaires tels que les robots. En effet, à base de ces réseaux, nous allons proposer deux schémas de commande dans les chapitres IV et V.

## CHAPITRE II

### Commande décentralisée à structure variable des robots manipulateurs

#### II.1 Introduction

Durant la dernière décennie, la commande à structure variable a attiré l'attention de plusieurs chercheurs dans le monde. L'intérêt porté à cette technique est motivé par le développement technologique des circuits à commutation rapide permettant de réaliser les lois de commande de ce type. En effet, cette approche a été appliquée avec succès dans divers domaines industriels, tels que l'électronique de puissance [BOU90], les réseaux d'énergie électrique [AGG93],[AGG94] et la robotique [YEU88],[LEU91],[SU94],[ERB99]. L'avantage principal de cette commande réside dans sa robustesse vis à vis des variations paramétriques et des perturbations lorsque le régime glissant apparaît. Cette caractéristique la rend un outil très approprié pour compenser l'effet des interconnexions dans les systèmes complexes.

Dans le contexte de la commande décentralisée, un nombre limité de résultats a été obtenu. Cependant, dans [MOR85] une commande décentralisée à structure variable (CDSV) a été proposée, dans laquelle la loi de commande dépend des paramètres du modèle et des variables d'accélération. Matthews et al.[MAT88] ont proposé une loi CDSV nécessitant l'observation locale des interconnexions avec les autres sous systèmes, ce qui introduit des restrictions sur l'applicabilité de cette méthode. Une CDSV adaptative avec surface de glissement linéaire a été également proposée par Feng et al. dans [FEN95]. Récemment, Tang et al. [TAN00] ont développé une commande décentralisée robuste pour les systèmes mécaniques. Cependant, ce résultat, basé sur théorie de Lyapunov et quelques propriétés physiques du système, nécessite la connaissance du modèle nominal du système.

Dans le présent chapitre, une nouvelle méthode de commande décentralisée à structure variable des robots manipulateurs sera proposée. En outre, une nouvelle classe de surfaces de glissement non linéaires avec action intégrale sera proposée, à partir de laquelle la loi de commande est développée. Le robot manipulateur est considéré comme un système interconnecté où chaque articulation est vue comme un sous système commandé par une unité de commande locale. Ainsi, pour l'élaboration de la CDSV, une surface de glissement non linéaire locale est proposée pour chaque sous système. La stabilité d'une telle classe de surfaces est démontrée en utilisant la théorie de la stabilité de Lyapunov. Cependant, une loi de commande est synthétisée pour chaque sous système afin de garantir la stabilité du système global. Vue la nature discontinue du signal de commande par mode de glissement, le phénomène de broutement

(chattering) apparaît. Pour remédier à ce problème, gênant en pratique, une couche bornée appropriée sera introduite.

Afin de tester la faisabilité et la robustesse du schéma de commande proposé, des simulations ont été effectuées sur le modèle du Robot PUMA 560 dans lesquelles les valeurs numériques des paramètres ont été obtenues d'une manière optimale en utilisant les algorithmes génétique (AG) [GOL89].

Par ailleurs, la méthode proposée a été également implantée expérimentalement pour la commande en poursuite de trajectoire d'un robot articulé à trois degrés de liberté disponible au Laboratoire de Robotique de Paris. Les résultats expérimentaux seront présentés à la fin de ce chapitre.

## II.2. Description du modèle dynamique du robot manipulateur

Le modèle dynamique d'un manipulateur à  $n$ -degrés de liberté peut s'écrire comme une équation différentielle non linéaire vectorielle du second ordre [YEU88]:

$$M(q)\ddot{q} + B(q, \dot{q})\dot{q} + K(q, \dot{q})q + G(q) = u_p + u \quad (\text{II.1})$$

où  $q \in \mathcal{R}^n$ ,  $\dot{q} \in \mathcal{R}^n$  et  $\ddot{q} \in \mathcal{R}^n$  sont, respectivement, les vecteurs de position, de vitesse et d'accélération, tandis que  $M(q) \in \mathcal{R}^{n \times n}$  est une matrice d'inertie symétrique définie positive et bornée.  $B(q, \dot{q})\dot{q} + K(q, \dot{q})q \in \mathcal{R}^n$  représente des forces de Coriolis et centrifuge et  $G(q) \in \mathcal{R}^n$  est le vecteur de gravitation.  $u_p \in \mathcal{R}^n$  et  $u \in \mathcal{R}^n$  désignent respectivement le vecteur des perturbations externes et celui de commande.

On note ici que  $q$ ,  $\dot{q}$  sont généralement des fonctions continues dans le temps.

En général, les éléments  $M$ ,  $B$ ,  $K$  et  $G$  sont des fonctions compliquées et non linéaires par rapport aux coordonnées généralisées et des vitesses des articulations, ainsi qu'aux paramètres d'inertie liés aux charges manipulées par l'organe terminal. Sachant que les éléments de  $M$  et  $G$  sont fonctions des variables articulaires, le modèle dynamique de la forme (II.1) montre les interactions fortes entre les différentes articulations. En plus,  $M$  est une matrice non diagonale indiquant les couplages entre les accélérations des différentes articulations.

Etant donné le modèle dynamique (II.1), nous cherchons à synthétiser une commande décentralisée à structure variable pour le robot, dans laquelle chaque sous système est commandé par une unité de commande locale. Cette commande, en poursuite de trajectoire, doit assurer la stabilité du système global. Pour se faire, chaque articulation du manipulateur est vue comme un sous système interconnecté avec les autres sous systèmes à travers les termes de couplage liés aux forces de Coriolis, centrifuges et gravitationnelles.

Donc, le sous système défini par la  $i^{\text{ème}}$  articulation peut être décrit par le modèle dynamique local suivant:

$$\begin{aligned}
 m_{ii}(q) \ddot{q}_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^n m_{ij}(q) \ddot{q}_j(t) + b_{ii}(q, \dot{q}) \dot{q}_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^n b_{ij}(q, \dot{q}) \dot{q}_j(t) \\
 + k_{ii}(q, \dot{q}) q_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^n k_{ij}(q, \dot{q}) q_j(t) + g_i(q) = u_{pi}(t) + u_i(t) \quad i = 1, \dots, n
 \end{aligned} \tag{II.2}$$

où l'indice "i" est relatif à l' $i^{\text{ème}}$  sous système,  $m_{ii}(q)$  sont les éléments positifs diagonaux de la matrice d'inertie  $M$ .

Dans une forme plus simple (II.2) peut s'écrire comme suit:

$$\begin{aligned}
 m_{ii}(q) \ddot{q}_i(t) + b_{ii}(q, \dot{q}) \dot{q}_i(t) + k_{ii}(q, \dot{q}) q_i(t) \\
 + d_i(q, \dot{q}, \ddot{q}) = u_{pi}(t) + u_i(t) \quad i = 1, \dots, n
 \end{aligned} \tag{II.3}$$

avec

$$d_i(q, \dot{q}, \ddot{q}) = g_i(q) + \sum_{\substack{j=1 \\ j \neq i}}^n \left( m_{ij}(q) \ddot{q}_j(t) + b_{ij}(q, \dot{q}) \dot{q}_j(t) + k_{ij}(q, \dot{q}) q_j(t) \right) \tag{II.4}$$

L'équation (II.3) est le modèle dynamique entrée-sortie du  $i^{\text{ème}}$  sous-système avec des coefficients variables et fonctions des interconnexions,  $m_{ii}(q)$ ,  $b_{ii}(q, \dot{q})$ ,  $k_{ii}(q, \dot{q})$  et  $d_i(q, \dot{q}, \ddot{q})$ . La commande  $u_i(t)$  et la perturbation  $u_{pi}(t)$  sont les entrées. La position  $q_i(t)$  est la sortie du  $i^{\text{ème}}$  modèle dynamique.  $d_i(q, \dot{q}, \ddot{q})$  peut être vue comme une perturbation ( couple ou force) des autres sous systèmes sur l' $i^{\text{ème}}$  sous système. Il représente les termes de gravitation, de Coriolis et centrifuge de la  $i$ -ème articulation ainsi que l'effet du couplage inertiel des autres articulations.

### II.3. Synthèse de la loi de commande

Notre objectif consiste à synthétiser une loi de commande décentralisée à structure variable pour chaque sous système. En effet, le  $i^{\text{ème}}$  contrôleur génère la commande local  $u_i(t)$  en utilisant seulement les variables locales de la  $i^{\text{ème}}$  articulation de telle manière à ce que  $q_i(t)$  converge asymptotiquement vers la trajectoire désirée  $q_{di}(t)$ . Cela doit se faire en deux étapes:

- Sélectionner une surface de commutation telle qu'en mode de glissement le sous système ait le comportement dynamique désiré.

- Déterminer une loi de commande décentralisée garantissant l'atteignabilité de la surface de glissement et l'apparition du mode de glissement.

### II.3.1. Sélection de la surface de glissement

De nombreux résultats ont été obtenus sur l'application de la commande par mode de glissement aux robots manipulateurs durant les deux dernières décennies [YEU88], [LEU91],[SLO91]...Cependant, en majorités, ces méthodes considèrent des surfaces de glissement linéaires, invariantes dans le temps et sont proposées dans un contexte centralisé. Dans [HUN93], les auteurs ont mentionné que pour un système à  $m$ -entrées,  $m$ -fonctions de commutation et jusqu'à  $2^m - 1$  surfaces de glissement peuvent être distinguées. Par conséquent, des difficultés supplémentaires apparaissent à cause de l'existence des différents schémas de commutation pour atteindre la surface de glissement éventuelle. En effet, plusieurs solutions ont été proposées. Parmi lesquelles, le schéma de commutation décentralisée qui est vu comme la solution favorisée pour la commande des systèmes complexes interconnectés. Dans ce qui suit, nous proposons une nouvelle classe de surfaces de glissement pour la commande décentralisée.

Dans la théorie de la commande à structure variable des robots, la surface de glissement est choisie de telle sorte qu'en régime de glissement, l'erreur de poursuite converge vers zéro. En d'autres termes, la surface de glissement doit être asymptotiquement stable.

#### **Théorème:**

Considérons, pour le sous système  $i$ , la loi de commutation non linéaire du premier ordre avec un terme intégral non linéaire.

$$\sigma_i = \tilde{q}_i + \Lambda_{i,1}(\tilde{q}_i) + \Lambda_{i,2}(z_i) \quad (\text{II.5})$$

où  $\tilde{q}_i = q_{di} - q_i$  désigne l'erreur de poursuite,  $z_i = \int_0^t \tilde{q}_i(\tau) d\tau$  et  $\Lambda_j(.)$   $j = 1,2$  sont des

fonctions *Lipshitzienne* continues en leurs arguments.

La surface de glissement locale donnée par

$$\sigma_i = 0, \quad (\text{II.6})$$

est asymptotiquement stable si les conditions suivantes sont vérifiées:

$$x \Lambda_j(x) > 0 \quad j = 1,2 \quad \text{pour} \quad x \neq 0 \quad (\text{II.7})$$

où  $x$  est l'argument de  $\Lambda_j$  (c.a.d..  $x = \tilde{q}_i$  si  $j=1$ ,  $x = z_i$  si  $j = 2$ ).



Puisque  $(\Lambda_j(x), j = 1,2)$  sont continues, cela implique que  $\Lambda_j(0) = 0, j = 1,2$  (voir figure. II.1).

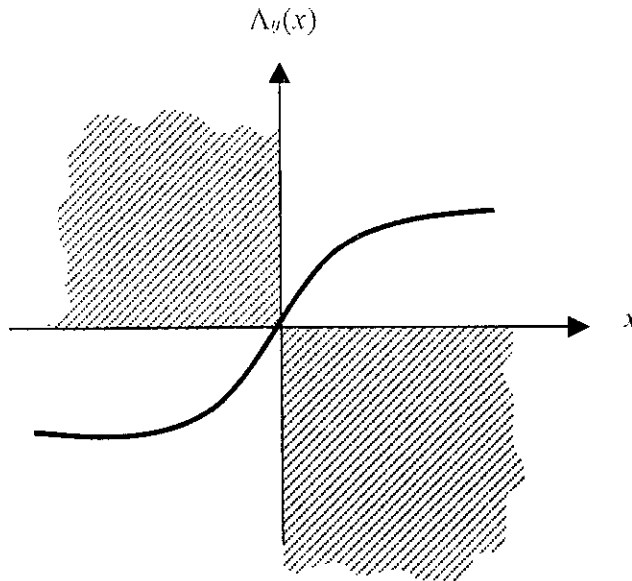


Figure II.1. forme possible des fonctions non linéaires  $\Lambda_j(x), j = 1,2$ .

**Preuve:**

Considérons la fonction de Lyapunov suivante:

$$V_i = \frac{1}{2} \tilde{q}_i^2 + \int_0^{\tilde{z}_i} \Lambda_{i2}(r) dr . \quad (\text{II.8})$$

dont la dérivée par rapport au temps est

$$\dot{V}_i = \tilde{q}_i \dot{\tilde{q}}_i + \dot{\tilde{z}}_i \Lambda_{i2}(\tilde{z}_i) \quad (\text{II.9})$$

de (II.5), (II.6) et (II.9), nous écrivons:

$$\begin{aligned} \dot{V}_i &= \tilde{q}_i (-\Lambda_{i1}(\tilde{q}_i) - \Lambda_{i2}(\tilde{z}_i)) + \tilde{q}_i \Lambda_{i2}(\tilde{z}_i) \\ &= -\tilde{q}_i \Lambda_{i1}(\tilde{q}_i) \end{aligned} \quad (\text{II.10})$$

De (II.7) et (II.10), l'inégalité  $\dot{V}_i < 0$  est toujours vérifiée pour  $\tilde{q}_i \neq 0$ . Donc  $\tilde{q}_i = 0$  est un point d'équilibre, d'où la stabilité asymptotique de la surface de glissement locale.

**Remarque 1:** Dans [SLO88] on a noté que  $\int_0^t \tilde{q}_i(\tau) d\tau$  peut être remplacé par  $\int \tilde{q}_i(\tau) d\tau$ , c.a.d., que l'intégrale peut être définie à une constante près. On peut choisir cette constante pour obtenir  $\sigma_i(0) = 0$  indépendamment de  $q_{di}(0)$  et  $\dot{q}_{di}(0)$  en posant:

$$\sigma_i = \tilde{q}_i + \Lambda_{i1}(\tilde{q}_i) + \Lambda_{i2}(z_i) - \tilde{q}_i(0) - \Lambda_{i1}(\tilde{q}_i(0)) \quad (\text{II.11})$$

Il est à souligner que les surfaces de glissement, linéaires ou non linéaires, rencontrées dans la littérature de la commande à structure variable en poursuite de trajectoire sont des cas particuliers (II.11) [YEU88], [LEU91], [MORG85], [SLO91], [YU96]-[TAN98]. Parmi lesquelles nous citons entre autres:  $\sigma = \dot{x} + 2\lambda x + \lambda^2 \int_0^t x(\tau) d\tau$ ,  $\sigma = \dot{x} + \Lambda(x)$  où  $\alpha x^2 \leq x \Lambda(x) \leq \beta x^2$  avec  $0 < \alpha \leq \beta < \infty$ .

### II.3.2. Synthèse de la commande décentralisée à structure variable

L'objectif de la synthèse d'une loi de commande décentralisée à structure variable est d'établir, pour chaque articulation, une loi de commande  $u_i(t)$  permettant de maintenir le régime de glissement. D'autre part, la sortie  $q_i(t)$  du  $i^{\text{ème}}$  sous système doit suivre le mieux possible la trajectoire désirée  $q_{di}(t)$  choisie arbitrairement.

Il est connu dans la théorie de la commande par mode de glissement, que les trajectoires d'état du système atteignent la surface de glissement globale  $\sigma^T = [\sigma_1, \dots, \sigma_n] = [0, \dots, 0]$  si la condition d'attractivité (d'atteignabilité)  $\sigma^T \dot{\sigma} < 0$  est satisfaite [UTK78]. Si cette condition est toujours vérifiée, le système en boucle fermée est asymptotiquement stable.

Pour satisfaire la condition d'attractivité du système global et assurer la stabilité asymptotique du système entier, une condition locale  $\sigma_i \dot{\sigma}_i < 0$  doit être vérifiée pour chacun des sous systèmes.

Pour l'établissement du schéma de commande décentralisée, nous nous basons sur les conditions d'attractivité locales. Afin d'obtenir ce dernier, nous multiplions la fonction positive  $m_{ii}(q)$  relative à l'inertie de l'articulation  $i$  (II.3) par la dérivée de la variable de commutation locale  $\sigma_i$ , d'où

$$m_{ii}(q) \dot{\sigma}_i = -p_{i1} \text{sgn}(\sigma_i) - p_{i2} \sigma_i \quad (\text{II.12})$$

où  $p_{i1}(q_i, \dot{q}_i, q_{di}, \dot{q}_{di}, \ddot{q}_{di})$  et  $p_{i2}(q_i, \dot{q}_i, q_{di}, \dot{q}_{di}, \ddot{q}_{di})$  sont des fonctionnelles positives et

$$\text{sgn}(\sigma_i) = \begin{cases} +1 & \text{si } \sigma_i > 0, \\ 0 & \text{si } \sigma_i = 0, \\ -1 & \text{si } \sigma_i < 0. \end{cases} \quad (\text{II.13})$$

En dérivant (II.5) par rapport au temps, nous obtenons

$$\dot{\sigma}_i = \frac{\partial \sigma_i}{\partial \tilde{q}_i} \frac{d\tilde{q}_i}{dt} + \frac{\partial \sigma_i}{\partial \dot{\tilde{q}}_i} \frac{d\dot{\tilde{q}}_i}{dt} + \frac{\partial \sigma_i}{\partial t} = \ddot{q}_i + \Lambda'_{i1}(\tilde{q}_i) \dot{\tilde{q}}_i + \Lambda'_{i2}(z_i) \tilde{q}_i \quad (\text{II.14})$$

où  $\Lambda'_{ij}(x)$  est la dérivée de  $\Lambda_{ij}(x)$  par rapport à son argument.

D'où nous écrivons

$$m_{ii}(q) \dot{\sigma}_i = m_{ii}(q) \ddot{q}_{ii} - m_{ii}(q) \ddot{q}_i + m_{ii}(q) \Lambda'_{i1}(\tilde{q}_i) \dot{\tilde{q}}_i + m_{ii}(q) \Lambda'_{i2}(z_i) \tilde{q}_i \quad (\text{II.15})$$

En utilisant le modèle dynamique de la  $i$ -ème articulation donné par (II.3) et en considérons l'équation (II.15), nous obtenons:

$$m_{ii}(q) \dot{\sigma}_i = \sum_{j=0}^2 \psi_{ij} q_{ii}^{(j)} + \sum_{j=0}^1 \phi_{ij} q_i^{(j)} + \gamma_i - u_i(t) \quad (\text{II.16})$$

l'indice "(j)" indique la  $j$ -ème dérivée par rapport au temps, et

$$\psi_{i0} = m_{ii}(q) \Lambda'_{i2}(z_i), \quad (\text{II.17.a})$$

$$\psi_{i1} = m_{ii}(q) \Lambda'_{i1}(\tilde{q}_i), \quad (\text{II.17.b})$$

$$\psi_{i2} = m_{ii}(q), \quad (\text{II.17.c})$$

$$\phi_{i0} = k_{ii}(q, \dot{q}) - m_{ii}(q) \Lambda'_{i2}(z_i), \quad (\text{II.17.d})$$

$$\phi_{i1} = b_{ii}(q, \dot{q}) - m_{ii}(q) \Lambda'_{i1}(\tilde{q}_i), \quad (\text{II.17.e})$$

$$\gamma_i = d_{ii}(q, \dot{q}, \ddot{q}) - u_{pi}(t), \quad (\text{II.17.f})$$

que nous pouvons écrire comme suit:

$$\psi_{ij} = \bar{\psi}_{ij} + \Delta\psi_{ij} \quad j = 0,1,2, \quad (\text{II.18.a})$$

$$\phi_{ij} = \bar{\phi}_{ij} + \Delta\phi_{ij} \quad j = 0,1, \quad (\text{II.18.b})$$

$$\gamma_i = \bar{\gamma}_i + \Delta\gamma_i \quad (\text{II.18.c})$$

avec  $\bar{\psi}_{ij}, \bar{\phi}_{ij}, \bar{\gamma}_i$  des valeurs moyennes de  $\psi_{ij}, \phi_{ij}, \gamma_i$ , et  $\Delta\psi_{ij}, \Delta\phi_{ij}, \Delta\gamma_i$  sont leurs variations respectives supposées bornées comme suit:

$$|\Delta\psi_{ij}| < \hat{\psi}_{ij} \quad j = 0,1,2 \quad (\text{II.19.a})$$

$$|\Delta\phi_{ij}| < \hat{\phi}_{ij} \quad j = 0,1 \quad (\text{II.19.b})$$

$$|\Delta\gamma_i| < \hat{\gamma}_i \quad (\text{II.19.c})$$

**Remarque 2 :** Puisque  $q, \dot{q}$  sont des fonctions continues, donc  $\dot{q}, \ddot{q}$  sont bornées et  $m_{ij}(q), b_{ij}(q, \dot{q}), k_{ij}(q, \dot{q}), d_i(q, \dot{q}, \ddot{q})$  sont aussi bornées. En outre,  $\Lambda_{ij}(\cdot)$   $j = 0,1,2$  sont des fonctions *Lipchitziennes* continues en leurs arguments. En effet, les suppositions précédentes sont valides.

Proposons la *i<sup>ème</sup>* loi de commande

$$u_i(t) = \sum_{j=0}^2 \psi_{ij}^\circ(t) q_{di}^{(j)} + \sum_{j=0}^1 \phi_{ij}^\circ(t) q_i^{(j)} + \gamma_i^\circ(t) + u_{ai}(t) \quad (\text{II.20})$$

avec  $u_{ai}(t)$  un terme de commande auxiliaire et

$$\psi_{ij}^\circ(t) = \bar{\psi}_{ij} + \chi_{ij} q_{di}^{(j)} \sigma_i \quad j = 0,1,2, \quad (\text{II.21.a})$$

$$\phi_{ij}^\circ(t) = \bar{\phi}_{ij} + \rho_{ij} q_i^{(j)} \sigma_i \quad j = 0,1, \quad (\text{II.21.b})$$

$$\gamma_i^\circ(t) = \bar{\gamma}_i + \mu_i \sigma_i \quad (\text{II.21.c})$$

où  $\chi_{ij}, \rho_{ij}$  et  $\mu_i$  des constantes arbitraires.

De (II.12), (II.16), (II.18), (II.20) et (II.21) nous obtenons:

$$\begin{aligned} -p_{i1} \operatorname{sgn}(\sigma_i) - p_{i2} \sigma_i &= \sum_{j=0}^2 (\Delta\psi_{ij} - \chi_{ij} q_{di}^{(j)} \sigma_i) q_{di}^{(j)} \\ &+ \sum_{j=0}^1 (\Delta\phi_{ij} - \rho_{ij} q_i^{(j)} \sigma_i) q_i^{(j)} \\ &+ (\Delta\gamma_i - \mu_i \sigma_i) - u_{ai}(t) \end{aligned} \quad (\text{II.22})$$

Les coefficients  $\chi_{ij}, \rho_{ij}, \mu_i$ , et  $u_{ai}(t)$  doivent être déterminés de façon à ce que (II.12) soit vérifiée.

Si  $\sigma_i > 0$ , alors

$$\begin{aligned}
m_u(q) \dot{\sigma}_i &= \sum_{j=0}^2 \Delta \psi_{ij} q_{di}^{(j)} + \sum_{j=0}^1 \Delta \phi_{ij} q_i^{(j)} + \Delta \gamma_i - u_{ai}(t) \\
&\quad - \left( \sum_{j=0}^2 \chi_{ij} (q_{di}^{(j)})^2 + \sum_{j=0}^1 \rho_{ij} (q_i^{(j)})^2 + \mu_i \right) \sigma_i, \\
&= -p_{i1} \operatorname{sgn}(\sigma_i) - p_{i2} \sigma_i < 0
\end{aligned} \tag{II.23}$$

Si  $\sigma_i < 0$ , alors

$$\begin{aligned}
m_u(q) \dot{\sigma}_i &= \sum_{j=0}^2 \Delta \psi_{ij} q_{di}^{(j)} + \sum_{j=0}^1 \Delta \phi_{ij} q_i^{(j)} + \Delta \gamma_i - u_{ai}(t) \\
&\quad - \left( \sum_{j=0}^2 \chi_{ij} (q_{di}^{(j)})^2 + \sum_{j=0}^1 \rho_{ij} (q_i^{(j)})^2 + \mu_i \right) \sigma_i, \\
&= -p_{i1} \operatorname{sgn}(\sigma_i) - p_{i2} \sigma_i > 0
\end{aligned} \tag{II.24}$$

En combinant les expressions précédentes nous obtenons:

$$\begin{aligned}
&\left( \sum_{j=0}^2 \Delta \psi_{ij} q_{di}^{(j)} + \sum_{j=0}^1 \Delta \phi_{ij} q_i^{(j)} + \Delta \gamma_i - u_{ai}(t) \right) \operatorname{sgn}(\sigma_i) \\
&\quad - \left( \sum_{j=0}^2 \chi_{ij} (q_{di}^{(j)})^2 + \sum_{j=0}^1 \rho_{ij} (q_i^{(j)})^2 + \mu_i \right) |\sigma_i| < 0
\end{aligned} \tag{II.25}$$

Pour satisfaire (II.25),  $\chi_{ij}$ ,  $\rho_{ij}$  et  $\mu_i$  doivent être choisis positifs et la commande auxiliaire  $u_{ai}(t)$  comme suit :

$$u_{ai}(t) = \operatorname{sgn}(\sigma_i) \left\{ \sum_{j=0}^2 \hat{\psi}_{ij} |q_{di}^{(j)}| + \sum_{j=0}^1 \hat{\phi}_{ij} |q_i^{(j)}| + \hat{\gamma}_i \right\} \tag{II.26}$$

En substituant  $u_{ai}(t)$  dans (II.20) par son expression (II.26) nous obtenons la commande locale:

$$\begin{aligned}
u_i(t) &= \sum_{j=0}^2 \psi_{ij}^\circ(t) q_{di}^{(j)} + \sum_{j=0}^1 \phi_{ij}^\circ(t) q_i^{(j)} + \gamma_i^\circ(t) \\
&\quad + \operatorname{sgn}(\sigma_i) \left\{ \sum_{j=0}^2 \hat{\psi}_{ij} |q_{di}^{(j)}| + \sum_{j=0}^1 \hat{\phi}_{ij} |q_i^{(j)}| + \hat{\gamma}_i \right\}
\end{aligned} \tag{II.27}$$

L'analyse de cette commande, permet de remarquer qu'elle est constituée de deux parties. La première est un régulateur à action PD<sup>2</sup> en feedforward variable, un PD variable en feedback et un signal variable. La seconde partie est liée à la loi de commutation locale, à un PD<sup>2</sup> en feedforward avec signaux en valeur absolue, à un PD en feedback avec signaux en valeur absolue et à un signal constant.

La loi de commande proposée est schématisée par la figure II.2 .

La commande à structure variable est discontinue de nature sur la surface de glissement. Cette caractéristique fait apparaître le phénomène de broutement (chattering). Ce dernier est

indésirable en pratique à cause de la fréquence élevée des oscillation du signal de commande. Pour réduire l'effet de ce phénomène, une couche bornée variable est introduite autour de la surface de commutation. A l'intérieur de cette couche, la commande est prise comme approximation continu de la fonction de commutation. Dans notre cas, la fonction signe (II.13) est remplacée par la fonction suivante, donnée dans [WAN96]:

$$M_{\delta}(\sigma_i) = \frac{\sigma_i}{|\sigma_i| + \delta_{i0} + \delta_{i1} |\tilde{q}_i|} \quad (\text{II.28})$$

où  $\delta_{i0}$  et  $\delta_{i1}$  sont deux scalaires positifs.

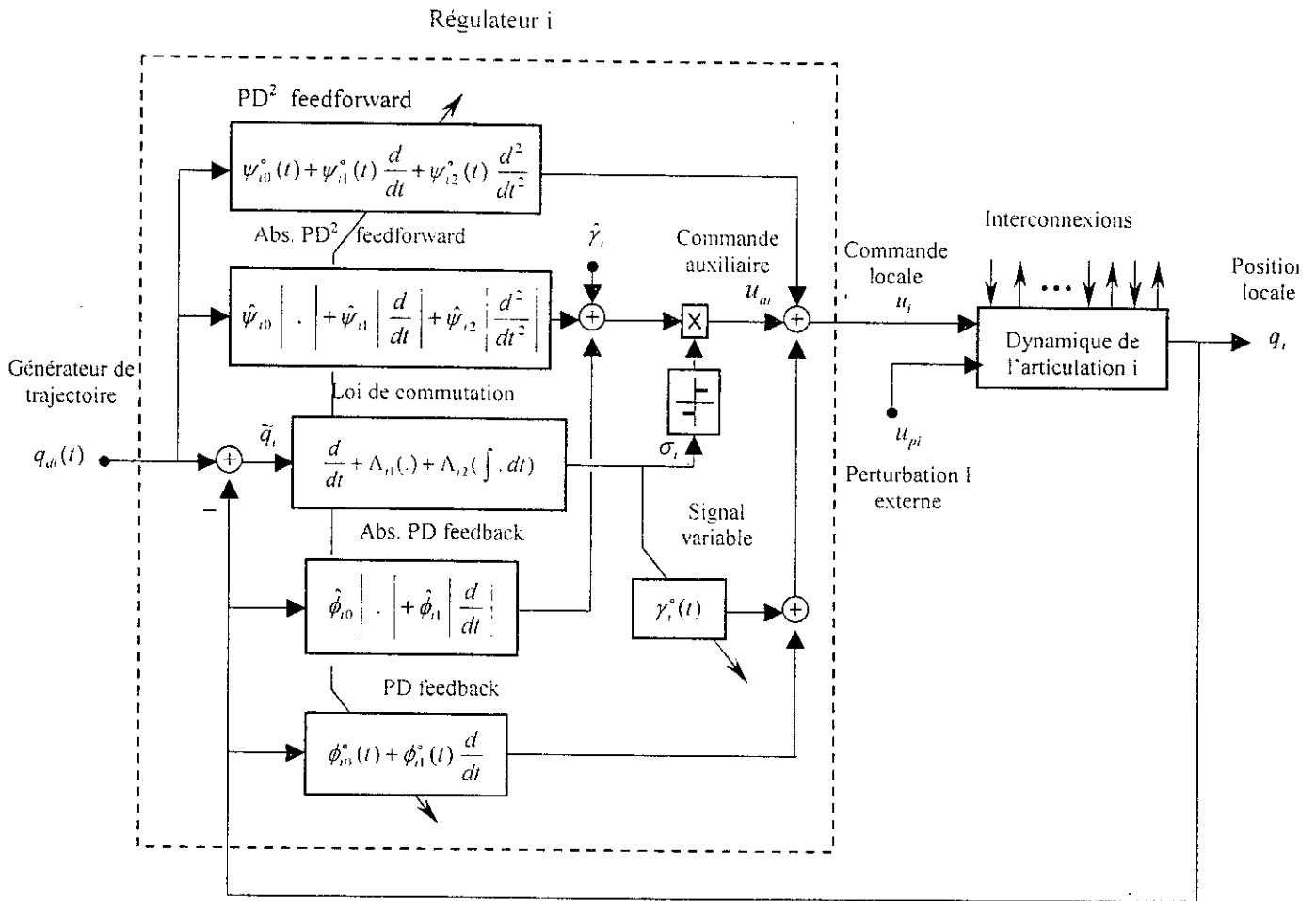


Figure II.2. Schéma de la commande décentralisée à structure variable des robots.

#### II.4. Résultats de simulation

Dans le but de valider la méthode proposée, une application de la loi de commande décentralisée à structure variable a été effectuée sur un Robot manipulateur de type PUMA 560 (figure II.3) en considérant seulement les trois premières articulations. La méthode de Range-

Kutta avec un pas de simulation de 0.0005 seconde est utilisée. Le modèle dynamique du robot étant donné en annexe.

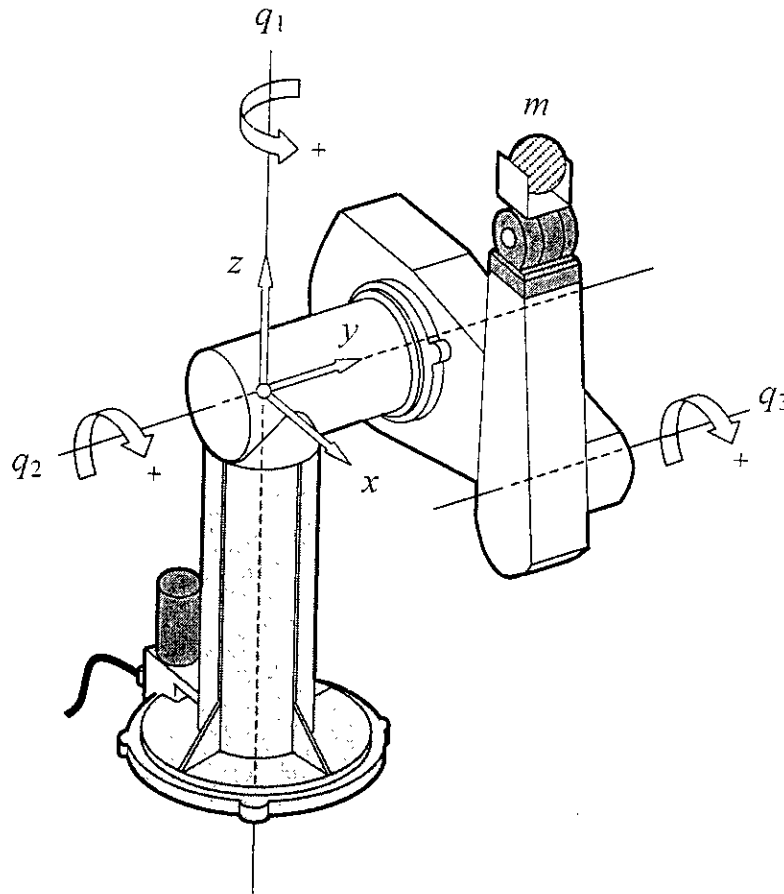


Figure II.3. Schéma du robot PUMA 560

Pour chaque contrôleur local, destiné à la  $i^{\text{ème}}$  articulation, les fonctions non linéaires suivante  $\Lambda_{ij}(\cdot)$  de la surface de commutation sont choisies en concordance avec le théorème énoncé:

$$\begin{aligned} \Lambda_{i1}(\tilde{q}_i) &= \lambda_i \tilde{q}_i + \alpha_i \tanh(\beta_i \tilde{q}_i), \\ \Lambda_{i2}(z_i) &= \eta_i z_i (1 + z_i^2) \end{aligned} \quad i = 1,2,3 \quad (\text{II.29})$$

avec  $\lambda_i$ ,  $\alpha_i$ ,  $\beta_i$  et  $\eta_i$  des constantes positives.

Les valeurs numériques des paramètres de la surface de glissement ainsi que les autres coefficients intervenant dans la loi de commande sont calculés, d'une manière optimale, en utilisant les algorithmes génétiques (AG) [GOLD89]. L'algorithme de recherche minimise le critère quadratique suivant :

$$I = \sum_{i=1}^n \sum_{l=0}^{gn} (\xi_{\tilde{q}_i} \tilde{q}_i^2(l) + \xi_{u_i} u_i^2(l)) \quad (\text{II.30})$$

avec  $\xi_{\tilde{q}_i} = 1$ ,  $\xi_{u_i} = 10^{-9}$ ,  $i = 1, 2, 3$ ,  $gn=30$  (nombre de générations), et une taille de population de 20, en utilisant un codage réel. Les opérateurs de l'AG sont: Croisement en un point, sélection par la méthode de roulette de Casinos et taux de mutation adaptatif. Pour différents paramètres, les intervalles de recherche sont choisis comme suit:

$$\bar{\psi}_i, \bar{\gamma}_i, \hat{\psi}_i, \hat{\phi}_i, \hat{\gamma}_i, \lambda_i, \alpha_i, \beta_i, \eta_i \in [0, 30], \bar{\phi}_i \in [-30, 0], \lambda_i, \alpha_i, \beta_i, \eta_i \in [0, 50]$$

Les valeurs numériques de ces paramètres sont données dans la table II.1 et la table II.2. La couche bornée (II.28) est introduite avec  $\delta_{o1} = \delta_{o2} = \delta_{o3} = 0.05$ , et  $\delta_{i1} = \delta_{i2} = \delta_{i3} = 10$ .

Table II.1. Valeurs des paramètres utilisés en simulation

Paramètres	Régulateur de l'articulation i		
	i = 1	i = 2	i = 3
$\bar{\psi}_{i0}$	24.4634	3.6594	22.0900
$\bar{\psi}_{i1}$	15.8460	10.2474	15.7014
$\bar{\psi}_{i2}$	23.2730	6.8844	4.8582
$\bar{\phi}_{i0}$	-20.1370	-14.9532	-21.8337
$\bar{\phi}_{i1}$	-12.1608	-8.8183	-3.8333
$\bar{\gamma}_i$	13.2068	5.5234	17.9497
$\hat{\psi}_{i0}$	19.6542	21.7271	17.9497
$\hat{\psi}_{i1}$	13.1796	12.2295	22.0431
$\hat{\psi}_{i2}$	11.9077	18.3109	29.8559
$\hat{\phi}_{i0}$	5.5121	10.0477	14.8641
$\hat{\phi}_{i1}$	29.1391	28.6879	2.4520
$\hat{\gamma}_i$	0.3195	29.9021	27.4093
$\lambda_i$	4.5708	28.0419	24.3561
$\alpha_i$	46.4287	8.1614	48.7399
$\beta_i$	31.3327	12.8221	25.7787
$\eta_i$	46.9526	46.9425	40.1572



Table II.2. Valeurs des constantes utilisées en simulation

Constantes	Régulateur de l'articulation $i$		
	$i = 1$	$i = 2$	$i = 3$
$\chi_{i0}$	2.3424	20.0249	10.7051
$\chi_{i1}$	6.0082	5.2191	21.7284
$\chi_{i2}$	27.4842	16.5666	10.6851
$\rho_{i0}$	18.8342	27.7996	5.3662
$\rho_{i1}$	8.1009	27.6662	15.0264
$\mu_i$	23.2452	20.7153	17.5393

L'évolution de la fonction d'évaluation lors de la recherche par l'algorithme génétique est donnée par la figure II.4.

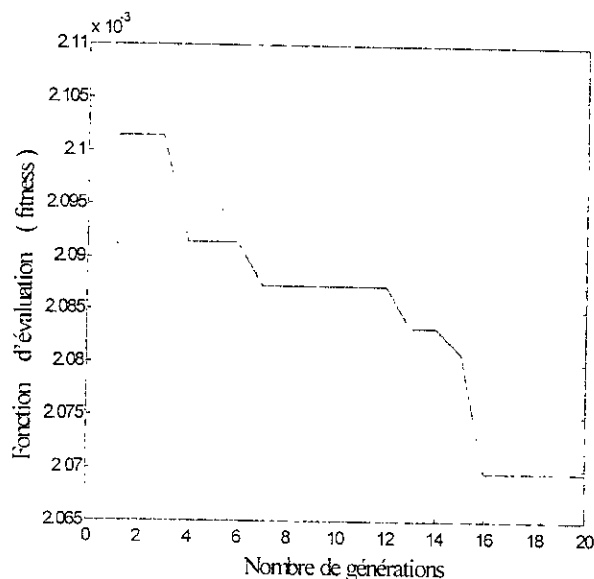


Figure II.4. Fonction d'évaluation

La trajectoire de référence est choisie pour amener les positions  $q_i(t)$  de l'état initial  $q_{di}(0)$  à l'état final  $q_{di}(t_f)$  en un temps  $t_f$  avec  $\dot{q}_{di}(0) = \dot{q}_{di}(t_f) = 0$ . Cette trajectoire cycloïdale [SER89] est donnée par :

$$\begin{aligned}
 q_{di}(t) &= q_{di}(0) + \frac{\Delta}{2\pi} [\omega t - \sin \omega t] \text{ rad,} & \text{pour } 0 \leq t \leq t_f \\
 &= q_{di}(t_f) \text{ rad} & \text{pour } t_f < t
 \end{aligned}
 \tag{II.31}$$

où  $\omega = 2\pi / t_f$  et  $\Delta_i = q_{di}(t_f) - q_{di}(0)$ .

Notons que cette trajectoire assure une continuité en position, vitesse et accélération. Pour cela, la configuration du robot varie d'une manière continue de la position initiale  $\{q_{d1}(0) = -50^\circ, q_{d2}(0) = -135^\circ, q_{d3}(0) = +135^\circ\}$  à la position finale  $\{q_{d1}(t_f) = +45^\circ, q_{d2}(t_f) = -85^\circ, q_{d3}(t_f) = +30^\circ\}$  en 1.5 secondes. Ce mouvement, appelé aussi test de Leahvy, permet d'exciter toutes les dynamiques. Il est souvent utilisé comme test standard pour le robot de type PUMA560.

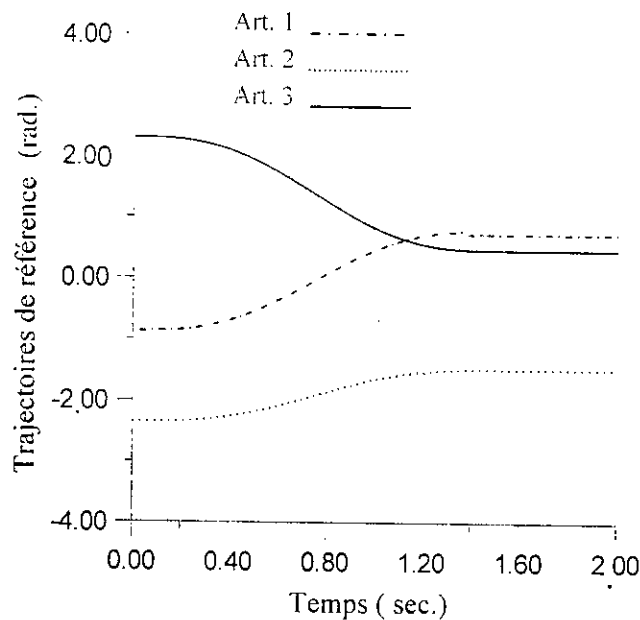


Figure II.5. Trajectoires de référence

Il est simulé un test de robustesse par rapport à la chute soudaine, à l'instant  $t = 0.75$  seconde, d'une masse de 5 kg manipulée par le robot en utilisant les lois de commande (II.27) avec introduction de la couche bornée. La figure II.5 montre les trajectoires de référence des différentes articulations. Les figure II.6 et II.7 montrent, respectivement, les erreurs de poursuite et les couples développés pour chaque articulation. A partir des ces figures, on peut remarquer que la robustesse est garantie.

Dans le but de constituer une base de comparaison, une commande à régulateur PD décentralisé classique

$$u_i = k_{pi} \tilde{q}_i + k_{vi} \dot{\tilde{q}}_i \quad i = 1,2,3 \quad (\text{II.32})$$

a été synthétisée. Les valeurs des paramètres de réglage utilisés en simulation sont [STE91]  $k_{pi} = 144, k_{vi} = 24$ .

Pour comparer les performances du la CDSV et celles du régulateur PD en terme d'erreur de poursuite maximale et d'amplitude maximale de la commande, une série de simulation a été effectuée pour différentes charges manipulées. Les figures II.8 et II.9 montrent que, par exemple

pour la seconde articulation, les maximums d'erreur et de commande pour la CDSV sont largement petits comparés à ceux du régulateur PD, et varient en fonction de la charge avec une pente plus petite. Cela confirme le caractère robuste appréciable de la loi de commande CDSV proposée.

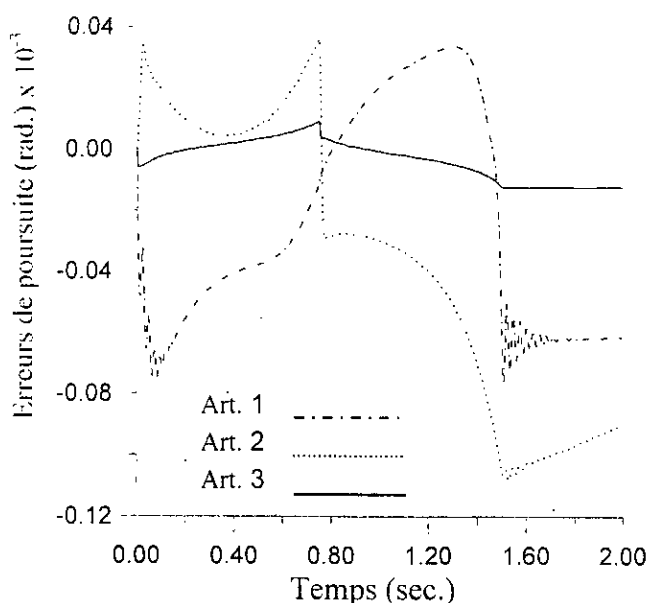


Figure II.6. Erreurs de poursuite

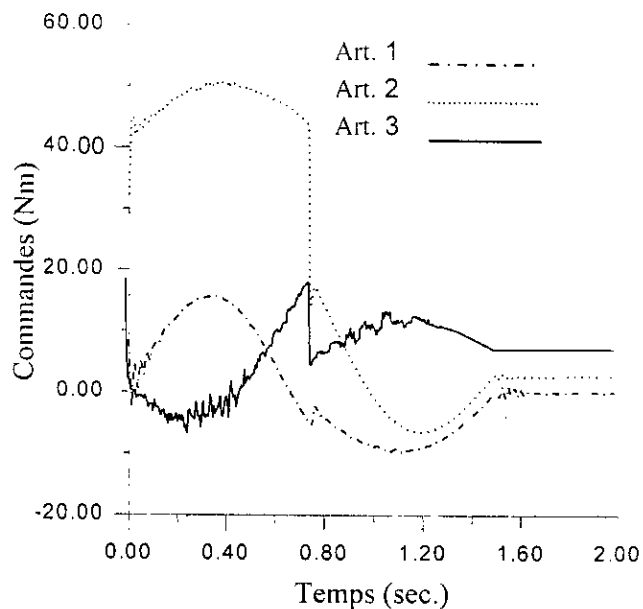


Figure II.7. Couples de commande

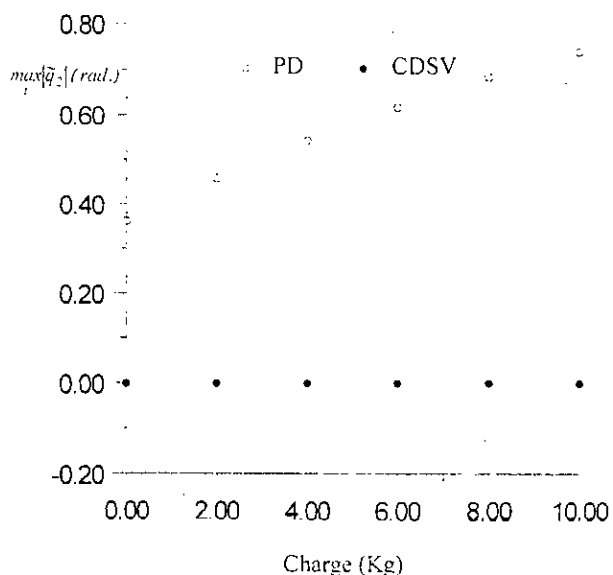


Figure II.8. Erreur maximale de la seconde articulation en fonction de la charge.

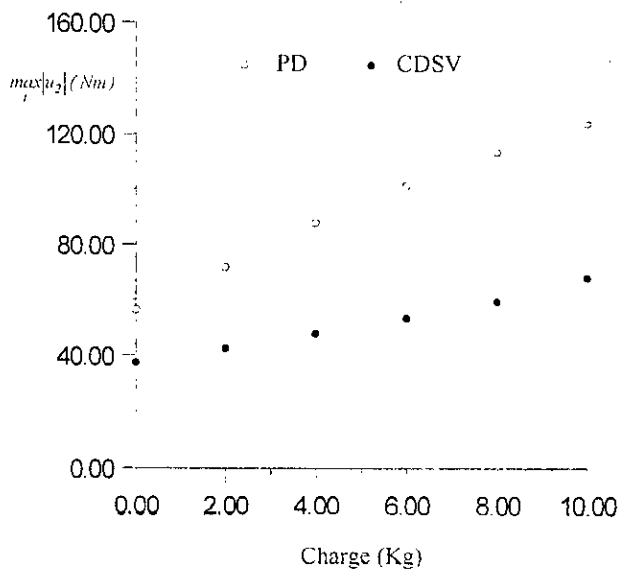


Figure II.9. Commande maximale de la seconde articulation en fonction de la charge

Constantes	Régulateur de l'articulation $i$		
	$i = 1$	$i = 2$	$i = 3$
$\chi_{i0}$	0.5	0.5	0.5
$\chi_{i1}$	0.5	0.5	0.5
$\chi_{i2}$	0.5	0.5	0.5
$\rho_{i0}$	1	1	1
$\rho_{i1}$	1	1	1
$\mu_i$	1	1	1

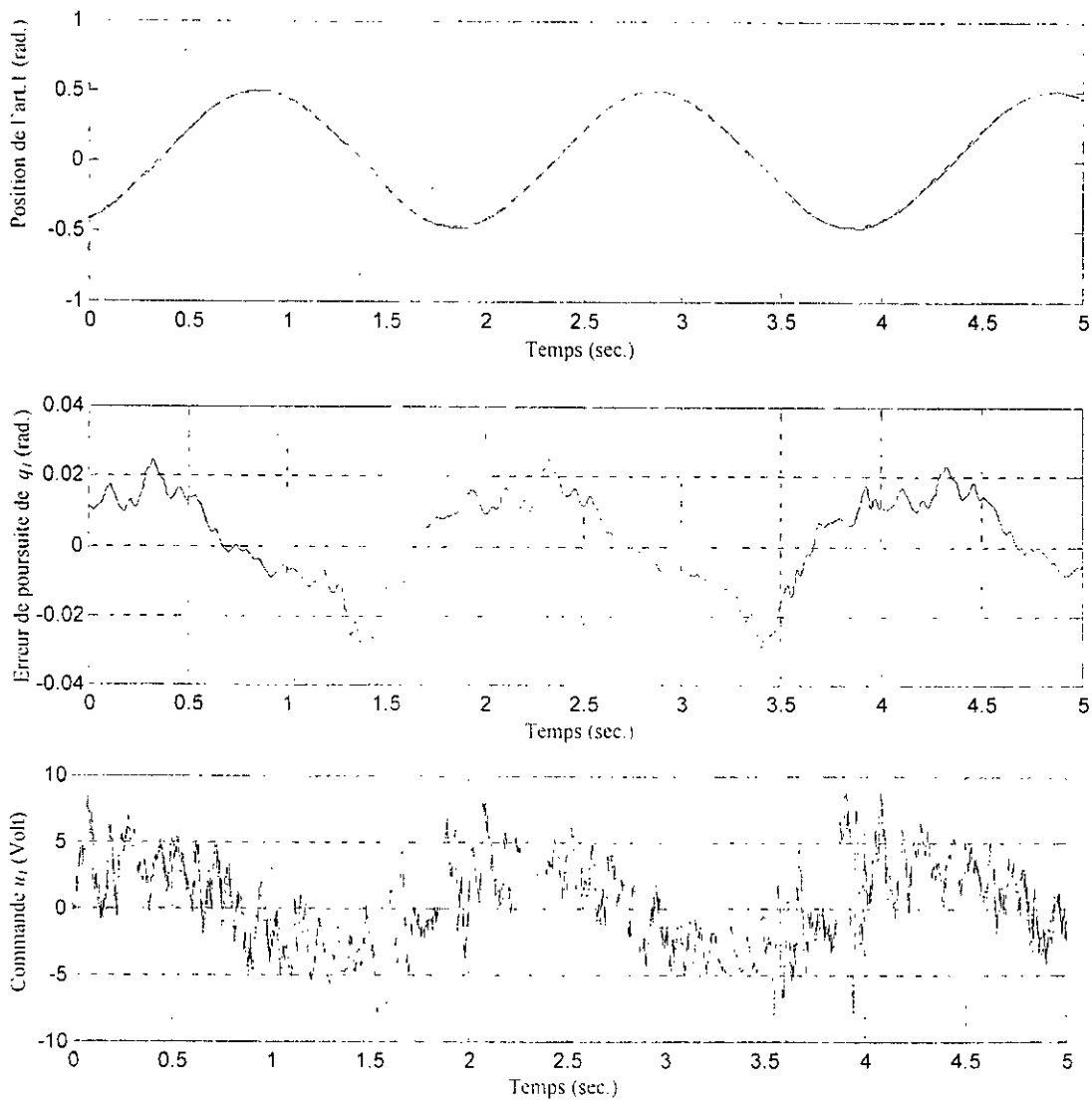


Figure 13a. Signaux expérimentaux de la trajectoire de sortie, de l'erreur de poursuite et de la commande relatifs à l'articulation 1.

Référence ----- Sortie \_\_\_\_\_

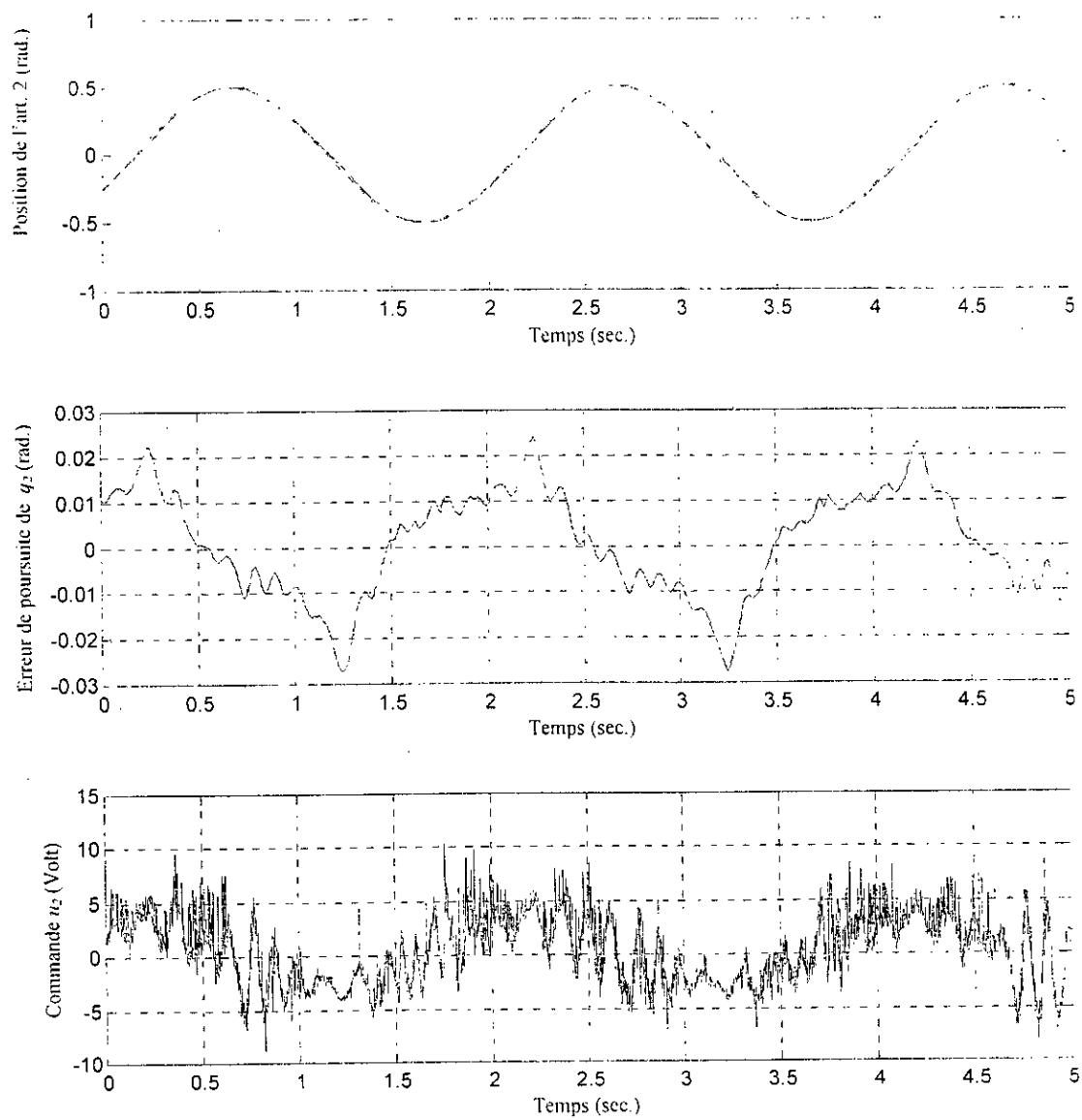


Figure 13b. Signaux expérimentaux de la trajectoire de sortie, de l'erreur de poursuite et de la commande relatifs à l'articulation 2.

Référence ----- Sortie \_\_\_\_\_

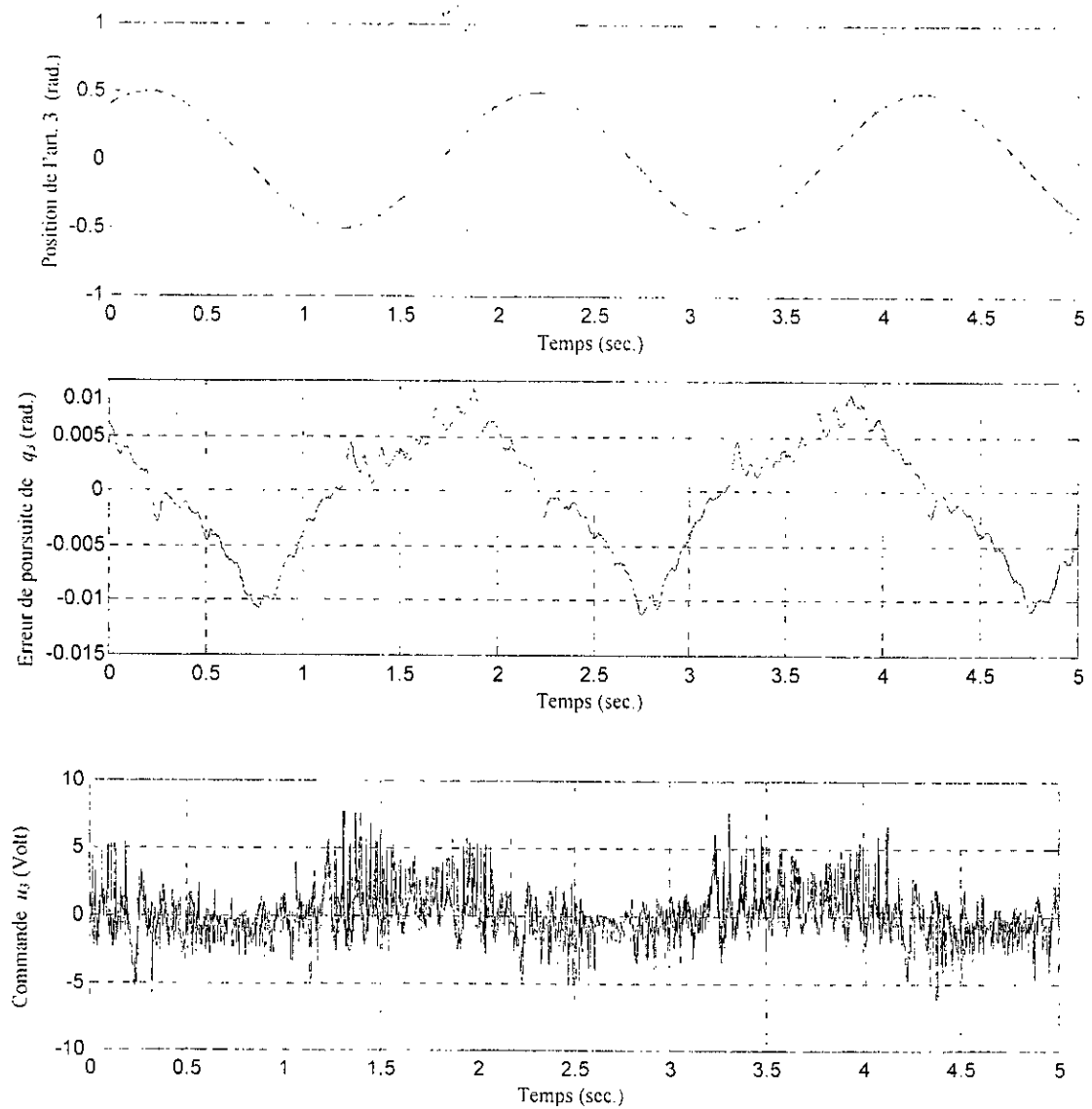


Figure 13c. Signaux expérimentaux de la trajectoire de sortie, de l'erreur de poursuite et de la commande relatifs à l'articulation 3.  
 Référence -----      Sortie ———

## II.6. Conclusion

Dans le présent chapitre, un nouveau schéma de commande décentralisée à structure variable a été proposé pour les robots manipulateurs. Pour cela, une nouvelle classe de surfaces de glissement non linéaires a été proposée dont la structure est facile à obtenir en choisissant des fonctions non linéaires appropriées. L'avantage principal de la décentralisation est la simplification aussi bien de la sélection de la surface de glissement que de la synthèse des lois de commande pour les systèmes multivariables. Le schéma de commande développé a été testé par simulation sur un robot PUMA 560 dont les paramètres de synthèse sont obtenus d'une manière optimale par les algorithmes génétiques. Les résultats de simulation ont montré clairement l'efficacité de la méthode ainsi que les bonnes performances en terme de poursuite de trajectoires. La robustesse de la commande via les variations paramétriques et la rupture de liaison au niveau d'une station de commande est aussi démontrée. La comparaison des performances avec celles du régulateur PD classique a permis de montrer la supériorité des performances de l'approche proposée. La faisabilité ainsi que les performances en poursuite de trajectoires sont aussi démontrées à travers les résultats expérimentaux obtenus sur le robot articulé à trois liaisons. Vu les résultats obtenus, l'approche de commande proposée constitue une bonne solution pour la commande robots et les systèmes mécaniques en général.

## **CHAPITRE III**

Commande décentralisée adaptative par  
logique floue



## CHAPITRE III

### Commande décentralisée adaptative par logique floue

#### III.1. Introduction

Actuellement, la commande à base de logique floue présente un intérêt particulier dans les milieux industriels. La conception de cette commande, basée sur des connaissances linguistiques, dépend essentiellement de l'expérience de l'opérateur humain chargé du fonctionnement de l'installation [DRI93][PIC97]. Les recherches menées actuellement sont consacrées aux développements de régulateurs pour des applications industrielles spécifiques [KIM94],[ZHA93],[TZA90][BOU00],[BOU99b]. Dans leurs majorités, les résultats obtenus n'offrent pas une méthodologie de synthèse générale permettant d'établir des schémas de commande fondés sur l'analyse de la stabilité du système bouclé. Cela est dû essentiellement au caractère linguistique des variables manipulées, ce qui limite l'utilisation de l'outil mathématique pour l'analyse [PIC97]. Cependant, des recherches récentes ont montré que certaines classes de systèmes flous peuvent être des approximateurs universels de fonctions non linéaires [MEN95]. Ce résultat, très intéressant, a donné de nouvelles orientations à l'utilisation des systèmes flous dans divers domaines. En effet, en automatique plusieurs travaux ont été réalisés en combinant les systèmes flous avec d'autres techniques telles que la commande adaptative [SPO95a][SPO95b][PIC97] et la commande linéarisante [TRE97].

Dans le présent chapitre, nous proposons un nouveau schéma de commande décentralisée adaptative à base de logique floue (CDALF), pour la commande en poursuite des robots manipulateurs. Le développement de la méthode proposée est basé principalement sur la propriété d'approximation des fonctions non linéaires par la logique floue, ainsi que l'approche de Seraji [SER89], dans lequel il proposa une loi de commande décentralisée adaptative au sens classique. En utilisant la théorie de stabilité de Lyapunov, nous montrons que la loi de commande proposée assure la convergence asymptotique des erreurs de poursuite ainsi que la bornitude des différents signaux intervenant dans la commande.

Afin de tester la faisabilité du schéma de commande proposé et évaluer ses performances en poursuite de trajectoire, une application à la commande d'un robot de type PUMA560 a été effectuée. Les résultats de simulation obtenus avec des tests de robustesse seront présentés à la fin de ce chapitre.

### III.2. Modèle dynamique décentralisé

Le modèle dynamique général d'un robot manipulateur rigide à un  $n$  degrés de liberté de l'équation (II.1) peut être représenté par un système d'équations non linéaires du second ordre sous la forme

$$M(q) \ddot{q} + B(q, \dot{q}) \dot{q} + K(q, \dot{q}) q + G(q) - u_p = u \quad (\text{III.1})$$

Comme il a été déjà mentionné au chapitre précédent, le problème de la commande décentralisée des robots manipulateurs consiste à élaborer un ensemble de régulateurs chacun d'entre eux est destiné pour une articulation. Le  $i^{\text{ème}}$  régulateur délivre la commande à partir des mesures locales  $q_i(t)$  et  $\dot{q}_i(t)$  et doit assurer une convergence asymptotique de l'erreur de poursuite [SER89]. Afin d'établir le schéma de commande décentralisée, on considère chaque articulation comme un sous système du robot. Chaque sous système est interconnecté avec le reste des sous systèmes à travers des couples et/ou des forces de couplage qui représentent des termes de couplage inertiels, de Coriolis et centrifuges, etc... Ainsi, le modèle dynamique du robot peut être considéré comme l'ensemble de  $n$  équations différentielles non linéaires du second ordre données par :

$$\begin{aligned} m_{ii}(q) \ddot{q}_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^n m_{ij}(q) \ddot{q}_j(t) + b_{ii}(q, \dot{q}) \dot{q}_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^n b_{ij}(q, \dot{q}) \dot{q}_j(t) + \\ k_{ii}(q, \dot{q}) q_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^n k_{ij}(q, \dot{q}) q_j(t) + g_i(q) - u_{pi}(t) = u_i(t) \quad i = 1, \dots, n \end{aligned} \quad (\text{III.2})$$

Si on prend

$$\begin{aligned} c_i(q, \dot{q}, \ddot{q}) = \sum_{\substack{j=1 \\ j \neq i}}^n m_{ij}(q) \ddot{q}_j(t) + b_{ii}(q, \dot{q}) \dot{q}_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^n b_{ij}(q, \dot{q}) \dot{q}_j(t) + \\ k_{ii}(q, \dot{q}) q_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^n k_{ij}(q, \dot{q}) q_j(t) + g_i(q) - u_{pi}(t) \quad i = 1, \dots, n \end{aligned} \quad (\text{III.3})$$

on obtient

$$m_{ii}(q) \ddot{q}_i + c_i(q, \dot{q}, \ddot{q}) = u_i(t) \quad (\text{III.4})$$

L'équation (III.4) représente le modèle dynamique de la  $i^{\text{ème}}$  articulation. Le terme  $c_i(q, \dot{q}, \ddot{q})$  peut être considéré comme une perturbation constituée par des termes locaux relatifs à l'articulation  $i$  ainsi que des termes de couplage avec les autres articulations.

Le schéma de commande décentralisée qu'on désire développer doit utiliser uniquement les mesures locales. Pour ce faire, nous allons nous baser sur le modèle de l'équation (III.4) relatif à chaque articulation.

En général, l'élaboration d'une loi de commande peut être effectuée principalement en deux étapes, à savoir: le choix de la structure de la commande et détermination d'une loi de commande assurant la stabilité du système bouclé. Dans notre cas, la commande décentralisée à base de logique floue, que nous allons développer, s'articule sur le modèle dynamique local exprimé en terme d'erreur de poursuite ainsi que la théorie de la stabilité de Lyapunov.

### III.3. Commande adaptative décentralisée par logique floue

Considérons un robot rigide à  $n$  degrés de liberté, dont le modèle dynamique décentralisé de chaque articulation est décrit par l'équation (III.4).

Les variables  $q_i$ ,  $\dot{q}_i$  et  $\ddot{q}_i$  représentent respectivement la position, la vitesse et l'accélération de la  $i^{\text{ème}}$  articulation.

Soit  $q_{di}(t)$ ,  $\dot{q}_{di}(t)$  et  $\ddot{q}_{di}(t)$ , respectivement la position, la vitesse et l'accélération désirées de la  $i^{\text{ème}}$  articulation et  $e_i(t) = q_{di}(t) - q_i(t)$ ,  $\dot{e}_i(t) = \dot{q}_{di}(t) - \dot{q}_i(t)$ ,  $\ddot{e}_i(t) = \ddot{q}_{di}(t) - \ddot{q}_i(t)$ , respectivement erreurs en position, en vitesse et en accélération;

D'une manière général, le problème de commande décentralisée des robots se pose comme suit:

Rechercher une loi de commande locale  $u_i = u_i(q_i, \dot{q}_i, \ddot{q}_i, q_{di}, \dot{q}_{di}, \ddot{q}_{di}, t)$  de manière à ce que l'erreur de poursuite  $e_i(t)$  converge asymptotiquement vers zéro.

D'après l'équation (III.4), on peut écrire:

$$m_i(q)\ddot{e}_i = l_i(q, \dot{q}, \ddot{q}, \ddot{q}_{di}) - u_i(t) \quad (III.5)$$

où

$$l_i(q, \dot{q}, \ddot{q}, \ddot{q}_{di}) = +c_i(q, \dot{q}, \ddot{q}) + m_i(q)\ddot{q}_{di} \quad (III.6)$$

Il faut noter que l'accélération de la  $i^{\text{ème}}$  articulation n'intervient pas dans le terme de couplage  $l_i$ . Elle est remplacée par l'accélération désirée  $\ddot{q}_{di}$ .

La fonction  $l_i$  peut être approximée comme suit :

$$l_i(q, \dot{q}, \ddot{q}, \ddot{q}_{di}) = f_i(q_i, \dot{q}_i, \ddot{q}_{di}) + \varepsilon_i^f(t) \quad i = 1, \dots, n \quad (\text{III.7})$$

Le premier terme est une fonction  $f_i$  dépendant uniquement des états locaux  $q_i, \dot{q}_i, \ddot{q}_{di}$ , tandis que second terme, notés  $\varepsilon_i^f(t)$ , est une erreur, appelée *erreur de reconstruction*. Cette erreur est le résultat de l'approximation du terme de couplage  $l_i$  par la fonction  $f_i$ . Elle dépend des états des autres articulations.

Considérons un modèle flou de type Sugeno (MFS), dont les variables d'entrée sont les grandeurs locales  $q_i, \dot{q}_i$  et  $\ddot{q}_{di}$ , et soit  $nf_1^i, nf_2^i$  et  $nf_3^i$  le nombre de fonctions d'appartenance associées respectivement aux variables  $q_i, \dot{q}_i$  et  $\ddot{q}_{di}$ . Ces fonctions sont notées  $(\mu_{1l_1}^i)_{l_1=1, \dots, nf_1^i}$ ,  $(\mu_{2l_2}^i)_{l_2=1, \dots, nf_2^i}$  et  $(\mu_{3l_3}^i)_{l_3=1, \dots, nf_3^i}$ . Le nombre de règles obtenues est  $nr^i = nf_1^i \times nf_2^i \times nf_3^i$ .

Utilisons ce modèle flou comme un approximateur universel (voir Chapitre I), pour écrire la fonction  $f_i$  sous la forme

$$f_i(q_i, \dot{q}_i, \ddot{q}_{di}) = Z_i^T A_i^* \xi^i + \varepsilon_i^f \quad (\text{III.8.a})$$

Le premier terme désigne la sortie du MFS,

avec

$$Z_i^T = \begin{cases} 1 & \text{si MFS est d'ordre 0} \\ [1 \quad q_i \quad \dot{q}_i \quad \ddot{q}_{di}] & \text{si MFS est d'ordre 1.} \end{cases} \quad (\text{III.8.b})$$

$A_i^*$  étant une matrice constante regroupant les gains des conséquences des règles floues. Elle est définie par [SPO95a,b]

$$A_i^* = \arg \min_{A_i \in \Omega_0^i} \left[ \sup_{q_i \in \Omega_1^i, \dot{q}_i \in \Omega_2^i, \ddot{q}_{di} \in \Omega_3^i} |Z_i^T A_i \xi^i - f_i(q_i, \dot{q}_i, \ddot{q}_{di})| \right] \quad (\text{III.8.c})$$

où  $\Omega_0^i, \Omega_1^i, \Omega_2^i$  et  $\Omega_3^i$  représentent, respectivement, les domaines admissibles des variables  $A_i, q_i, \dot{q}_i$  et  $\ddot{q}_{di}$ , avec :

$$A_i = \begin{cases} \begin{bmatrix} a'_{01} & \cdots & a'_{0nr'} \end{bmatrix} & \text{si MFS est d'ordre 0} \\ \begin{bmatrix} a'_{01} & \cdots & a'_{0nr'} \\ a'_{11} & \cdots & a'_{1nr'} \\ a'_{21} & \cdots & a'_{2nr'} \\ a'_{31} & \cdots & a'_{3nr'} \end{bmatrix} & \text{si MFS est d'ordre 1.} \end{cases} \quad (\text{III.8.d})$$

$\xi^i$  est un vecteur regroupant les degrés d'activation des différentes règles floues. Il est défini par

$$\xi^i = \begin{bmatrix} \xi^i_1 & \cdots & \xi^i_{nr'} \end{bmatrix}^T, \quad \xi^i_k = \frac{\mu'_{1k_1}(q_i) \times \mu'_{2k_2}(\dot{q}_i) \times \mu'_{3k_3}(\ddot{q}_i)}{\sum_{\substack{1 \leq l_1 \leq nr'_1 \\ 1 \leq l_2 \leq nr'_2 \\ 1 \leq l_3 \leq nr'_3}} \mu'_{1l_1}(q_i) \times \mu'_{2l_2}(\dot{q}_i) \times \mu'_{3l_3}(\ddot{q}_i)} \quad k = 1 \cdots nr' \quad (\text{III.8.e})$$

avec  $k = k_1 + \sum_{j=2}^3 (k_j - 1) \prod_{j=1}^j nr'_j$ ,  $k_1 = 1 \cdots nr'_1$ ,  $k_2 = 1 \cdots nr'_2$ ,  $k_3 = 1 \cdots nr'_3$ ,  $i = 1 \cdots n$

Dans (III.8.a), le terme  $\varepsilon^i_j$  est une constante qui représente une erreur due à l'approximation de  $f_i$  par le modèle MFS. Elle résulte de l'utilisation d'un nombre fini de fonctions d'appartenance.

D'après (III.8.c),  $A_i^*$  représente les gains optimaux des conséquences des règles floues des MFS, permettant de minimiser l'erreur de reconstruction  $\varepsilon^i_j$ . Cette dernière dépend du modèle flou adopté (le type, la structure et les paramètres).

Avant de donner la loi de commande, considérons les hypothèses suivantes :

#### Hypothèses:

- H0 : Les perturbations  $u_{p_i}(t)$   $i = 1, \dots, n$  sont bornées;
- H1 : Les trajectoires désirées  $q_{di}(t)$ ,  $\dot{q}_{di}(t)$  et  $\ddot{q}_{di}(t)$   $i = 1 \cdots n$  sont bornées;
- H2 : Les éléments diagonales de la matrice d'inertie  $m_{ii}(q)$ ,  $i = 1 \cdots n$ , varient lentement en fonction du temps;
- H3 : Les erreurs de reconstruction  $\varepsilon_1(t), \varepsilon_2(t), \dots, \varepsilon_n(t)$  varient lentement dans le temps.

En utilisant (III.6) et (III.8.a), l'équation (III.5) devient

$$m_{ii}(q)\ddot{e}_i = Z_i^T A_i^* \xi^i + \varepsilon^i_j + \varepsilon^i_j(t) - u_i(t) \quad i = 1 \cdots n \quad (\text{III.9})$$

Sur la base du modèle dynamique de l'erreur, donnée par l'équation (III.9), nous proposons la loi de commande décentralisée suivante :

$$u_i(t) = k_{pi}(t)e_i(t) + k_{vi}(t)\dot{e}_i(t) + Z_i^T \hat{A}_i(t)\xi_i' + \hat{\varepsilon}_i(t) \quad (\text{III.10})$$

où,

- $k_{pi}(t)e_i(t) + k_{vi}(t)\dot{e}_i(t)$  est un régulateur à actions proportionnelle et dérivé (PD) avec des gains  $k_{pi}$  et  $k_{vi}$  adaptatifs ;
- $Z_i^T \hat{A}_i(t)\xi_i'$  est la sortie d'un MFS estimant la fonction  $Z_i^T A_i^*(t)\xi_i'$ . La matrice  $\hat{A}_i$  est une estimation de  $A_i^*$ , déterminée en temps réel ;
- $\hat{\varepsilon}_i(t)$  est un signal auxiliaire utilisé pour compenser les erreurs  $\varepsilon_i^f$  et  $\varepsilon_i^l(t)$ .

avec les lois d'adaptation suivantes :

$$\dot{k}_{pi} = \bar{\alpha}_{1i} \bar{r}_i e_i + \bar{\alpha}_{2i} \frac{d}{dt}(\bar{r}_i e_i) \quad i = 1 \dots n \quad (\text{III.11.a})$$

$$\dot{k}_{vi} = \bar{\beta}_{1i} \bar{r}_i \dot{e}_i + \bar{\beta}_{2i} \frac{d}{dt}(\bar{r}_i \dot{e}_i) \quad i = 1 \dots n \quad (\text{III.11.b})$$

$$\dot{\hat{A}}_i(t) = \Gamma_{1i} Z_i \xi_i'^T \bar{r}_i + \Gamma_{2i} \frac{d}{dt}(Z_i \xi_i'^T \bar{r}_i) \quad i = 1 \dots n \quad (\text{III.11.c})$$

$$\dot{\hat{\varepsilon}}_i(t) = \bar{\gamma}_{1i} \bar{r}_i + \bar{\gamma}_{2i} \frac{d}{dt}(\bar{r}_i) \quad i = 1 \dots n \quad (\text{III.11.d})$$

avec l'erreur filtrée

$$\bar{r}_i = \bar{p}_{2i} e_i + \bar{p}_{3i} \dot{e}_i \quad i = 1 \dots n \quad (\text{III.11.e})$$

$$\bar{\alpha}_{1i}, \bar{\beta}_{1i}, \bar{\gamma}_{1i} > 0; \bar{\alpha}_{2i}, \bar{\beta}_{2i}, \bar{\gamma}_{2i} \geq 0 \text{ (scalaires)} ;$$

$$\Gamma_{1i} \text{ et } \Gamma_{2i} \text{ (matrices définies positives)} ;$$

$$\bar{p}_{2i}, \bar{p}_{3i} > 0 \text{ (scalaires).}$$

Dans ce cas, le problème de la commande décentralisée floue consiste à déterminer la fonction  $\hat{\varepsilon}_i(t)$  et les lois d'adaptation des gains  $k_{pi}$ ,  $k_{vi}$  et des paramètres du modèle flou  $\hat{A}_i$ , de manière à assurer la convergence asymptotique de l'erreur de poursuite  $e_i(t)$ . Seules les mesures de la  $i^{\text{ème}}$  articulation peuvent être utilisées.

Ainsi, nous énonçons le théorème suivant :

**Théorème:**

Etant donné un robot manipulateur à  $n$  degrés de liberté, dont le modèle dynamique de l'erreur de poursuite de chaque articulation peut être mis sous la forme (III.9), et en tenant compte des hypothèses H0, H1, H2 et H3.

Si on choisit la loi de commande (III.10) avec les lois d'adaptation (III.11,a,b,c,d) alors on aura les résultats suivants:

R1- Les états  $q_i, \dot{q}_i$  et  $\ddot{q}_i, i = 1 \dots n$  seront bornés ;

R2- Les variables  $k_p(t), k_v(t), \hat{A}_i(t)$  et  $\hat{\varepsilon}_i(t), i = 1 \dots n$ , seront bornées ;

R3- Les erreurs  $e_i(t) i = 1 \dots n$  seront bornées et convergent asymptotiquement vers zéro.

□

**Preuve :**

Pour simplifier la notation, nous allons omettre l'indice  $i$  relatif à l'articulation .

En appliquant l'équation de la loi de commande (III.10) dans l'équation du modèle dynamique de l'erreur (III.9), nous obtenons :

$$m\ddot{e} = -k_v\dot{e} - k_p e + Z^T \tilde{A} \xi + \tilde{\varepsilon} \quad (\text{III.12})$$

où,  $\tilde{A} = A^* - \hat{A}$  : erreur d'estimation de la matrice  $A^*$

$\tilde{\varepsilon} = \varepsilon_f + \varepsilon - \hat{\varepsilon}$  : erreur entre l'erreur de reconstruction  $(\varepsilon_f + \varepsilon)$  et le signal auxiliaire  $\hat{\varepsilon}$ .

posons  $X = [e \quad \dot{e}]^T$ . L'équation (III.12) devient :

$$\dot{X} = \begin{bmatrix} 0 & 1 \\ -\frac{k_p}{m} & -\frac{k_v}{m} \end{bmatrix} X + \frac{1}{m} \begin{bmatrix} 0 \\ Z^T \tilde{A} \xi + \tilde{\varepsilon} \end{bmatrix} \quad (\text{III.13})$$

Considérons un modèle de référence stable de l'erreur sous la forme

$$\ddot{e}_m + 2\bar{\eta}\omega\dot{e}_m + \omega^2 e_m = 0$$

où,  $\bar{\eta}$  et  $\omega$  sont respectivement, le coefficient d'amortissement et la pulsation propre du modèle de référence.

Sous forme matricielle on a

$$\dot{X}_m = DX_m \quad (\text{III.14a})$$

avec,

$$X_m = [e_m \quad \dot{e}_m]^T; D = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\bar{\eta}\omega \end{bmatrix} \quad (\text{III.14b})$$

Le modèle précédent est stable. alors d'après le théorème de Kalman-Yagobovich, il existe une matrice  $P$  symétrique définie positive (SDP) et une matrice  $Q > 0$  (DP) telle que [SER89]

$$PD + D^T P = -Q \quad (\text{III.15})$$

D'autre part, soit le vecteur  $E_v$  défini par

$$E_v = X_m - X \quad (\text{III.16})$$

Si les conditions initiales sont tels que :  $e_m(0) = \dot{e}_m(0) = 0$ , alors  $\forall t \geq 0. X_m(t) = 0$ , nous aurons alors

$$E_v(t) = -X(t) \quad (\text{III.17})$$

D'après les équations (III.13), (III.14) et (III.16), nous écrivons

$$\dot{E}_v = DE_v + \begin{bmatrix} 0 & 0 \\ \frac{k_p}{m} - \omega^2 \frac{k_v}{m} - 2\bar{\eta}\omega & \end{bmatrix} X - \frac{1}{m} \left[ Z^T \tilde{A} \xi + \tilde{\varepsilon} \right] \quad (\text{III.18})$$

avec,  $m > 0$ .

Considérons, d'autre part la fonction de Lyapunov

$$V_l = \frac{1}{2} E_v^T \bar{P} E_v + \frac{1}{2} Q_1 \left( \frac{k_p}{m} - \omega^2 - k_p^* \right)^2 + \frac{1}{2} Q_2 \left( \frac{k_v}{m} - 2\bar{\eta}\omega - k_v^* \right)^2 + \frac{1}{2} Tr((\tilde{A} + A_0)^T \Gamma_0^{-1} (\tilde{A} + A_0)) + \frac{1}{2} \bar{\gamma}_0^{-1} (\tilde{\varepsilon} + \varepsilon_0)^2 \quad (\text{III.19})$$

où,

- $Tr(\cdot)$  désigne la trace de la matrice ( $\cdot$ );
- $(\cdot)^T$  dénote le transpose de la matrice ( $\cdot$ );
- $Q_1, Q_2, \bar{\gamma}_0 > 0$  des scalaires strictement positifs;
- $\Gamma_0 > 0$  matrice définie positive;
- $k_p^*, k_v^*, A_0$  et  $\varepsilon_0$  des fonctions introduites pour obtenir des lois d'adaptation de type proportionnelle-intégrale. Elle seront déterminées ultérieurement.

La fonction  $V_l$  est positive de forme quadratique. C'est une fonction proposée en se basant sur la fonction de Lyapunov utilisée dans [SER89] où nous introduisons un terme supplémentaire lié aux paramètres du modèle flou.

Si nous considérons que l'inertie (ou la masse)  $m$  varie lentement (hypothèse H2), elle peut être supposée constante. Dans ce cas, la dérivée de la fonction  $V_l$ , nous obtenons

$$\dot{V}_l = \frac{1}{2} \left[ \dot{E}_v^T \bar{P} E_v + E_v^T \bar{P} \dot{E}_v \right] + Q_1 \left( \frac{k_p}{m} - \omega^2 - k_p^* \right) \left( \frac{\dot{k}_p}{m} - \dot{k}_p^* \right) + Q_2 \left( \frac{k_v}{m} - 2\bar{\eta}\omega - k_v^* \right) \left( \frac{\dot{k}_v}{m} - \dot{k}_v^* \right) + Tr((\tilde{A}^T + \dot{A}_0) \Gamma_0^{-1} (\dot{\tilde{A}} + \dot{A}_0)) + \bar{\gamma}_0^{-1} (\tilde{\varepsilon} + \varepsilon_0) (\dot{\tilde{\varepsilon}} + \dot{\varepsilon}_0) \quad (\text{III.20})$$

En considérant les relations (III.15), (III.16) et (III.18), le premier terme de l'expression (III.20) devient



$$\frac{1}{2} [\dot{E}_v^T \bar{P} E_v + E_v^T \bar{P} \dot{E}_v] = -\frac{1}{2} E_v^T Q E_v - \bar{r} \left( \frac{k_p}{m} - \omega^2 \right) e - \bar{r} \left( \frac{k_v}{m} - 2\bar{\eta}\omega \right) \dot{e} + \frac{\gamma_0^{-1}}{m} \bar{\varepsilon} \bar{r} + \frac{1}{m} Z^T \tilde{A} \xi \bar{r} \quad (\text{III.21})$$

avec,  $\bar{r} = \bar{p}_2 e + \bar{p}_3 \dot{e}$ , où,  $\bar{p}_2$  et  $\bar{p}_3$  sont des éléments de la matrice  $\bar{P}$  tels que :  $\bar{P} = \begin{bmatrix} \bar{p}_1 & \bar{p}_2 \\ \bar{p}_2 & \bar{p}_3 \end{bmatrix}$ .

en remplaçant l'équation (III.21) dans (III.20) et après arrangement nous obtenons

$$\dot{V}_l(t) = \dot{V}_{11}(t) + \dot{V}_{12}(t) + \dot{V}_{13}(t) \quad (\text{III.22.a})$$

avec,

$$\dot{V}_{11} = -\frac{1}{2} E_v^T Q E_v + \left( \frac{k_p}{m} - \omega^2 \right) \left( Q_1 \left( \frac{k_p}{m} - k_p^* \right) - \bar{r} e \right) \quad (\text{III.22.b})$$

$$+ \left( \frac{k_v}{m} - 2\bar{\eta}\omega \right) \left( Q_2 \left( \frac{k_v}{m} - k_v^* \right) - \bar{r} \dot{e} \right) - Q_1 \left( \frac{k_p}{m} - k_p^* \right) k_p^* - Q_2 \left( \frac{k_v}{m} - k_v^* \right) k_v^*$$

$$\dot{V}_{12} = \tilde{\varepsilon} \left( \frac{1}{m} \bar{r} + \bar{\gamma}_0^{-1} (\tilde{\varepsilon} + \dot{\varepsilon}_0) \right) + \bar{\gamma}_0^{-1} \varepsilon_0 (\tilde{\varepsilon} + \dot{\varepsilon}_0) \quad (\text{III.22.c})$$

$$\dot{V}_{13} = \frac{1}{m} \bar{r} Z^T \tilde{A} \xi + Tr(\tilde{A}^T \Gamma_0^{-1} (\tilde{A} + \dot{A}_0)) + Tr(A_0^T \Gamma_0^{-1} (\tilde{A} + \dot{A}_0)) \quad (\text{III.22.d})$$

Si l'erreur de reconstruction  $\varepsilon$  varie lentement par rapport au signal auxiliaire  $\hat{\varepsilon}$  (hypothèse H3), nous pouvons écrire  $\tilde{\varepsilon} = \varepsilon - \hat{\varepsilon} = -\dot{\hat{\varepsilon}}$ , d'où

$$\dot{V}_{12} = \tilde{\varepsilon} \left( \frac{1}{m} \bar{r} - \bar{\gamma}_0^{-1} (\dot{\hat{\varepsilon}} - \dot{\varepsilon}_0) \right) - \bar{\gamma}_0^{-1} \varepsilon_0 (\dot{\hat{\varepsilon}} - \dot{\varepsilon}_0) \quad (\text{III.23})$$

D'autre part,  $\tilde{A} = \dot{A} - \dot{\hat{A}} = -\dot{\hat{A}}$  car  $A^*$  est une matrice constante. De plus, si nous utilisons la relation matricielle  $Tr(\Phi_1 \Phi_2) = Tr(\Phi_2 \Phi_1)$  [SPO95a], nous pouvons écrire

$$\dot{V}_{13} = Tr(\tilde{A}^T \left( \frac{1}{m} \bar{r} Z \xi^T - \Gamma_0^{-1} (\dot{\hat{A}} - \dot{A}_0) \right)) - Tr(A_0^T \Gamma_0^{-1} (\dot{\hat{A}} - \dot{A}_0)) \quad (\text{III.24})$$

Pour avoir  $\dot{V}_l \leq 0$ , nous choisissons

$$\begin{cases} Q_1 \left( \frac{k_p}{m} - k_p^* \right) = \bar{r} e \\ Q_2 \left( \frac{k_v}{m} - k_v^* \right) = \bar{r} \dot{e} \\ k_p^* = Q_1^* Q_1 \left( \frac{k_p}{m} - k_p^* \right) = Q_1^* \bar{r} e & Q_1^* \geq 0. \\ k_v^* = Q_2^* Q_2 \left( \frac{k_v}{m} - k_v^* \right) = Q_2^* \bar{r} \dot{e} & Q_2^* \geq 0. \end{cases} \quad (\text{III.25.a})$$

$$\begin{cases} \bar{\gamma}_0^{-1}(\dot{\hat{\varepsilon}} - \dot{\varepsilon}_0) = \frac{\bar{r}}{m} \\ \varepsilon_0 = \bar{\gamma}_0^* \bar{\gamma}_0^{-1}(\dot{\hat{\varepsilon}} - \dot{\varepsilon}_0) = \bar{\gamma}_0^* \frac{\bar{r}}{m} & \bar{\gamma}_0^* \geq 0 \\ \Gamma_0^{-1}(\dot{\hat{A}} - \dot{A}_0) = \frac{\bar{r}}{m} Z \xi^T \\ A_0 = \Gamma_0^* \Gamma_0^{-1}(\dot{\hat{A}} - \dot{A}_0) = \Gamma_0^* \frac{\bar{r}}{m} Z \xi^T & \Gamma_0^* \geq 0 \text{ (SDP)} \end{cases} \quad (\text{III.25.b})$$

Nous obtenons donc

$$\begin{aligned} \dot{V}_{11} &= -\frac{1}{2} E_v^T Q E_v - Q_1^* (\bar{r} e)^2 - Q_2^* (\bar{r} \dot{e})^2 \leq 0 \\ \dot{V}_{12} &= -\bar{\gamma}_0^{-2} \bar{\gamma}_0^* (\dot{\hat{\varepsilon}} - \dot{\varepsilon}_0)^2 \leq 0 \quad \text{car } \bar{\gamma}_0^* \geq 0 \\ \dot{V}_{13} &= -\text{Tr} \left( (\Gamma_0^{-1}(\dot{\hat{A}} - \dot{A}_0))^T \Gamma_0^{*T} (\Gamma_0^{-1}(\dot{\hat{A}} - \dot{A}_0)) \right) \leq 0, \text{ car } \Gamma_0^* \geq 0 \end{aligned}$$

et,

$$\begin{aligned} \dot{V}_1 &= -\frac{1}{2} E_v^T Q E_v - Q_1^* (\bar{r} e)^2 - Q_2^* (\bar{r} \dot{e})^2 - \bar{\gamma}_0^{-2} \bar{\gamma}_0^* (\dot{\hat{\varepsilon}} - \dot{\varepsilon}_0)^2 \\ &\quad - \text{Tr} \left( (\Gamma_0^{-1}(\dot{\hat{A}} - \dot{A}_0))^T \Gamma_0^{*T} (\Gamma_0^{-1}(\dot{\hat{A}} - \dot{A}_0)) \right) \leq 0 \end{aligned} \quad (\text{III.26})$$

Donc, les relations (III.25.a,b) assure la négativité de la fonction dérivée ( $\dot{V}_1 \leq 0$ ).

Les inéquations  $V_1 \geq 0$  et  $\dot{V}_1 \leq 0$  implique la stabilité au sens de Lyapunov, d'où la bornitude  $V_1(t)$ ,  $E(t)$  et  $\tilde{\varepsilon}(t)$  ainsi que la matrice  $\tilde{A}(t)$ .

D'autre part, considérons les deux familles de fonctions  $L_\infty$ ,  $L_2$  définies par :

$$\begin{aligned} L_\infty &= \{s(t) : |s(t)| < \infty\} : \text{ensemble des fonctions bornées ;} \\ L_2 &= \left\{ s(t) : \int_0^{+\infty} s(t)^2 dt < \infty \right\} : \text{ensembles des fonctions quadratiquement intégrables.} \end{aligned}$$

D'après la relation (III.17), on a :  $E = -[e \quad \dot{e}]^T$ , et sachant que les trajectoires désirées  $q_d, \dot{q}_d \in L_\infty$  (hypothèse H1), alors la bornitude de  $E$  implique que :  $q, \dot{q} \in L_\infty$  (première conséquence du théorème (R1)).

Pour montrer la convergence asymptotique de l'erreur  $E_v(t)$ , on utilise le corollaire suivant:

**Corollaire** [KRS95][LEW93][SPO95a][SPO95b] :

On considère une fonction réelle  $\bar{r}$  qui vérifie les conditions suivantes :

- $\bar{r}$  et  $\dot{\bar{r}} \in L_\infty$  ;
- $\bar{r} \in L_2$

alors cette fonction converge asymptotiquement vers zéro.

D'après le corollaire ci-dessus, des conditions suffisantes pour assurer la convergence asymptotique de l'erreur vers zéro sont

$$C1 : e(t), \dot{e}(t) \in L_\infty$$

$$C2 : e(t), \dot{e}(t) \in L_2$$

$$C3 : \dot{e}(t), \ddot{e}(t) \in L_\infty$$

La première condition C1 résulte directement de la bornitude de  $E_v(t)$ . Pour vérifier la condition C2, nous choisissons la matrice  $Q$  de forme diagonale :  $Q = \text{diag}\{2\bar{\rho}_1, 2\bar{\rho}_2\}$  avec,  $\bar{\rho}_1, \bar{\rho}_2 > 0$ . De l'équation (III.24), nous pouvons déduire :

$$\dot{V}_1 \leq -\frac{1}{2}\bar{\rho}_1 e^2 - \frac{1}{2}\bar{\rho}_2 \dot{e}^2 \quad (\text{III.27})$$

d'où,

$$\begin{cases} \dot{V}_1 \leq -\frac{1}{2}\bar{\rho}_1 e^2 \\ \dot{V}_1 \leq -\frac{1}{2}\bar{\rho}_2 \dot{e}^2 \end{cases} \quad \text{alors} \quad \begin{cases} \int_0^t e(\tau)^2 d\tau \leq \frac{2}{\bar{\rho}_1} (V(0) - V(\infty)) \\ \int_0^t \dot{e}(\tau)^2 d\tau \leq \frac{2}{\bar{\rho}_2} (V(0) - V(\infty)) \end{cases}$$

mais  $V \in L_\infty$ , alors  $e, \dot{e} \in L_2$ .

D'après l'équation (III.25.a), on peut déduire :

$$k_p(t) = Q_1^{-1} m \int_0^t \bar{r} e d\tau + Q_1^* m \bar{r} e + k_p(0) \quad (\text{III.28.a})$$

$$k_v(t) = Q_2^{-1} m \int_0^t \bar{r} \dot{e} d\tau + Q_2^* m \bar{r} \dot{e} + k_v(0) \quad (\text{III.28.b})$$

Sachant que  $e, \dot{e}, \bar{r} \in L_\infty$ ,  $e, \dot{e} \in L_2$  et en utilisant l'inégalité

$$\left| \int_0^t e \dot{e} d\tau \right| \leq \int_0^t e^2 d\tau + \int_0^t \dot{e}^2 d\tau \quad (\text{III.29})$$

nous déduisons que :  $k_p$  et  $k_v \in L_\infty$  (conséquence R2). D'autre part, puisque  $E(t), X(t), \tilde{A}$  et  $\tilde{e}$  sont bornées ( $\in L_\infty$ ) alors, d'après l'équation (III.18),  $\ddot{e} \in L_\infty$ . Donc la troisième condition (C3) est vérifiée. D'après le corollaire, les erreurs  $e(t)$  et  $\dot{e}(t)$  convergent asymptotiquement vers zéro (conséquence R3). D'après l'hypothèse H1,  $\ddot{q}_d \in L_\infty$  donc la bornitude de  $\ddot{e}$  ( $\ddot{e} \in L_\infty$ ) implique que  $\ddot{q} \in L_\infty$  (conséquence R1).

Pour montrer la bornitude des variables  $\hat{A}$  et  $\hat{\varepsilon}$ , nous avons

$$\begin{cases} \tilde{A} \in L_\infty \\ \tilde{\varepsilon} \in L_\infty \end{cases} \Rightarrow \begin{cases} \hat{A} \in L_\infty \\ \hat{\varepsilon} \in L_\infty \end{cases} \quad \begin{array}{l} \text{car } \tilde{A} = A^* - \hat{A} \text{ et } A^* \text{ est constante.} \\ \text{car } \tilde{\varepsilon} = \varepsilon - \hat{\varepsilon} \text{ et } \varepsilon \text{ dépend de variables d'états bornées} \end{array}$$

d'où la conséquence R3.

Pour déterminer  $\bar{p}_2$  et  $\bar{p}_3$ , nous utilisons la relation (III.15), nous obtenons

$$\bar{p}_2 = \frac{\bar{\rho}_1}{\omega^2}, \bar{p}_3 = \frac{1}{2\eta\omega} \left( \frac{\bar{\rho}_1}{\omega} + \bar{\rho}_2 \right) \quad (\text{III.30})$$

Puisque  $\eta, \omega$  des arbitraires positifs, il suffit de prendre  $\bar{p}_2, \bar{p}_3 > 0$ . Si de plus dans l'expression (III.14a,b) et (III.17a,b), nous choisissons :

$$\begin{cases} Q_1^{-1} = \frac{\bar{\alpha}_1}{m}, \quad Q_1^* = \frac{\bar{\alpha}_2}{m} & \bar{\alpha}_1 > 0, \bar{\alpha}_2 \geq 0 \\ Q_2^{-1} = \frac{\bar{\beta}_1}{m}, \quad Q_2^* = \frac{\bar{\beta}_2}{m}, & \bar{\beta}_1 > 0, \bar{\beta}_2 \geq 0 \\ \bar{\gamma}_0 = \bar{\gamma}_1 \quad m, \quad \bar{\gamma}_0^* = \bar{\gamma}_2 \quad m & \bar{\gamma}_1 > 0, \bar{\gamma}_2 \geq 0 \\ \Gamma_0 = \Gamma_1 \quad m, \quad \Gamma_0^* = \Gamma_2 \quad m, & \Gamma_1 > 0, \Gamma_2 \geq 0 \end{cases} \quad (\text{III.31})$$

Les lois d'adaptations (III.14a,b) et (III.17a,b) s'écrivent :

$$\begin{cases} k_p(t) = \bar{\alpha}_1 \int_0^t \bar{r} e d\tau + \bar{\alpha}_2 \bar{r} e + k_p(0) \\ k_v(t) = \bar{\beta}_1 \int_0^t \dot{\bar{r}} e d\tau + \bar{\beta}_2 \bar{r} e + k_v(0) \\ \hat{A}(t) = \Gamma_1 \int_0^t Z \xi^T \bar{r} d\tau + \Gamma_2 Z \xi^T \bar{r} + \hat{A}(0) \\ \hat{\varepsilon}(t) = \bar{\gamma}_1 \int_0^t \bar{r} d\tau + \bar{\gamma}_2 \bar{r} + \hat{\varepsilon}(0). \end{cases} \quad (\text{III.32})$$

L'utilisation des lois d'adaptation précédentes dans la loi de commande (III.9), pour chaque articulation assure la convergence asymptotique des erreurs de poursuite vers zéro de toutes les articulations, ainsi que la bornitude des différents signaux adaptatifs.



Le schéma de principe de la commande adaptative décentralisée par logique floue (CADLF), en utilisant la loi de commande (III.10) et (III.11a,b,c,d) est illustré par la figure (III.1).

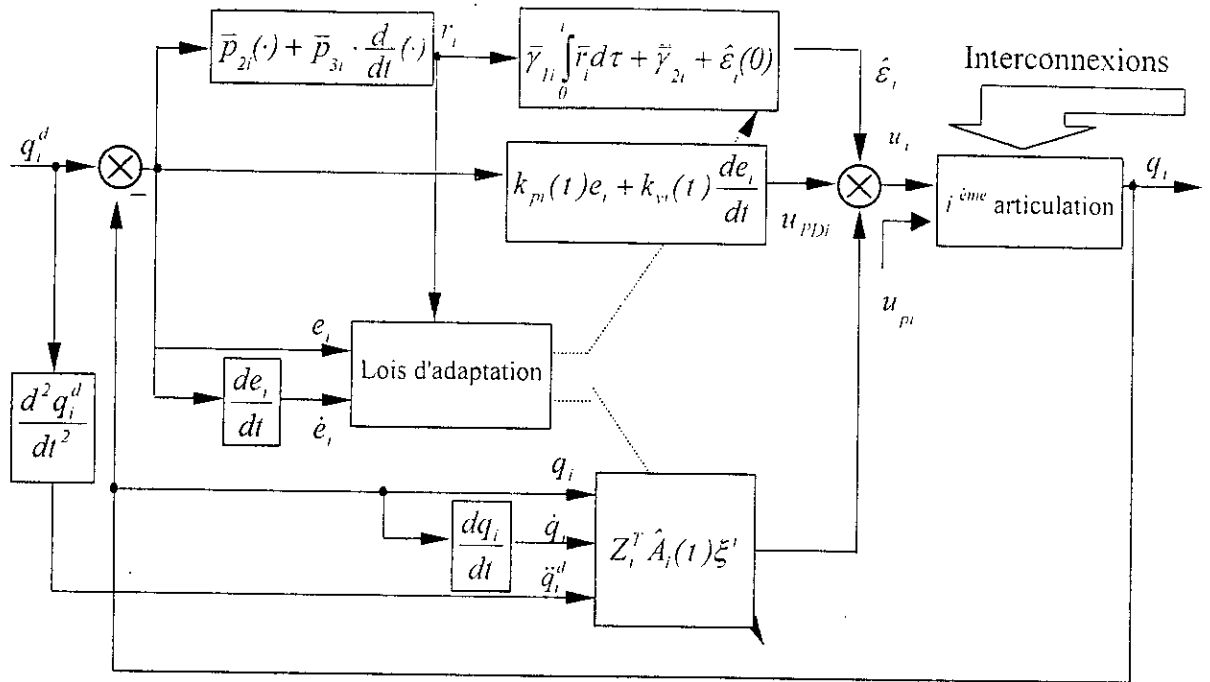


Figure III.1. Schéma de principe de la CADLF.

Il est à souligner que sous les hypothèses H0, H1, H2 et H3 et en vertu du théorème, l'utilisation de la loi de commande décentralisée floue adaptative (CADLF) proposée assure la bornitude et la convergence asymptotique des erreurs de poursuite vers zéro. La bornitude des commandes  $u_i(t)$   $i = 1 \dots n$  et des variables  $q_i, \dot{q}_i, \ddot{q}_i, k_{pi}, k_{vi}, \hat{A}_i$  et  $\hat{\epsilon}_i$ ,  $i = 1 \dots n$  est aussi garantie.

De plus, l'analyse de l'expression de la loi de commande  $u_i(t)$  proposée montre trois termes distincts :

- Un premier terme caractérise un retour d'état variable réalisé par une action proportionnelle-dérivée (PD), avec des gains  $k_{pi}, k_{vi}$  à adapter en temps réel dont la bornitude a été démontrée :
- Un second terme, caractérise la sortie d'un modèle flou adaptatif de type Sugeno (MFS). Il a été adopté afin de compenser les termes de couplages et des perturbations. Les gains des conséquences des règles floues sont adaptés en temps réel selon une loi de type proportionnelle-intégrale (PI).
- Un signal auxiliaire ( $\hat{\epsilon}_i$ ) qui représente un retour d'état réalisé par une action proportionnelle-intégrale-dérivée (PID) à gains fixes. Ce terme est nécessaire pour faire face à l'erreur de reconstruction.

**Remarque 1:**

Si on considère l'expression d'adaptation de la matrice  $\hat{A}_i$  (donnée par (III.11c)), on peut déduire

$$\dot{\hat{a}}'_{ij} = \Gamma'_{l+1} Z_i^T \xi'_j r_i + \Gamma'_{2l+1} \frac{d}{dt} (Z_i^T \xi'_j r_i), j = 1 \dots nr', i = 1 \dots n \quad (\text{III.33})$$

avec,  $l = \begin{cases} 0 & \text{si MFS est d'ordre 0.} \\ 0,1,2,3 & \text{si MFS est d'ordre 1.} \end{cases}$

$j = 1 \dots nr'$  indice de la règle ;  $i = 1 \dots n$  indice de l'articulation ;  $\Gamma'_{l+1}$  ;  $\Gamma'_{2l+1}$  : la ligne  $l+1$  des matrices  $\Gamma_1$ ,  $\Gamma_2$ , respectivement.

De cette expression, on remarque que les gains des conséquences des règles floues  $\hat{a}'_{ij}$ , subiront des modifications seulement lorsque les règles correspondantes sont activées ( $\xi'_j \neq 0$ ), car dans ce cas on obtient  $\dot{\hat{a}}'_{ij} \neq 0$ . Cependant les gains des règles inactivées gardent leurs valeurs constantes ( $\dot{\hat{a}}'_{ij} = 0$ ).

Cela reflète, d'une part, la propriété des modèles flous de produire des modèles locaux, qui dépendent de la partition des univers de discours des variables d'entrées  $q_i$ ,  $\dot{q}_i$  et  $\ddot{q}_i$  [PIC97]. D'autre part, ces modèles traduisent le principe de perturbation minimale " Minimal disturbance principle " [JAN95], qui stipule que l'adaptation ne doit pas seulement réduire l'erreur de sortie mais aussi minimiser les perturbations sur les sorties déjà utilisées dans l'adaptation.

L'identification effectuée par chaque règle (par l'adaptation de ses propres gains) et son intervention ne sont opérationnelles que lorsque cette même règle est activée. Donc, le modèle flou réalise une identification et une intervention selon des modèles locaux. Chaque modèle local correspond à une règle flou. L'expression particulière du vecteur d'activation ( $\xi'$ ) produit un lissage entre ces différents modèles.

**Remarque 2 :**

Il est à noter que le régulateur proposé est général dans le sens où il ne nécessite aucune information sur la structure où les paramètres du modèle dynamique. Cela est dû principalement à la présence d'une adaptation en temps réel des différents paramètres  $k_p, k_v, \hat{A}_i$  et  $\hat{e}_i, i = 1 \dots n$ .

**III.4. Résultats de simulation**

Afin de valider la méthode que nous avons proposée, nous l'avons appliqué pour la commande en poursuite de trajectoire du robot manipulateur PUMA 560 (Figure II.3) dont le modèle est donnée en annexe A. Seules les trois premières articulations sont considérées.

Les différents tests de simulation sont effectués en utilisant des trajectoires cycloïdales (équation II.31) avec le test de Leahy [LEA90]. La méthode de Runge-Kutta d'ordre 4, à pas fixe égal à 1 ms a été utilisée. Les intégrations dans la loi de commande sont calculées via la méthode des trapèzes.

Dans chaque régulateur associé à une articulation, nous avons adopté un modèle flou de type Sugeno d'ordre 0. Deux fonctions d'appartenance sont attribuées à chacune des trois variables d'entrées telles que montrée par la figure III.2. Ainsi, le modèle flou possède huit règles.

Les paramètres de la loi de commande sont donnés dans la table III.1 où  $1_{ij}$  est une matrice de dimensions  $i \times j$  dont les éléments sont tous égaux à 1.

Les différentes variables adaptatives de la loi de commande sont initialisées à zéro.

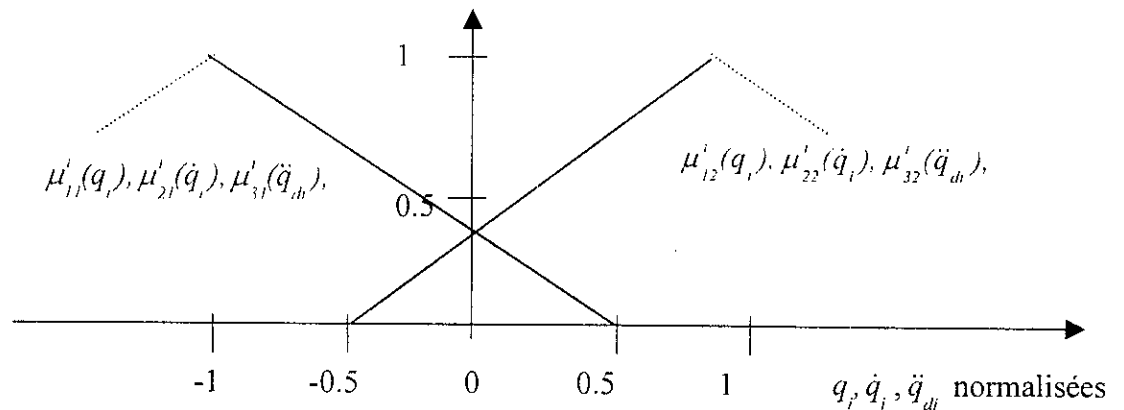


Figure II.2. Fonctions d'appartenance des variables d'entrées.

### Tests des performances

Pour valider le schéma de commande, un test à vide, sans variation paramétrique, a été effectué. Les figures III.3a,b montrent l'évolution des différents variables. On remarque que la poursuite des trajectoires de référence est assurée avec des erreurs faibles et inférieures à 0.2 degrés. Ces derniers s'annule en régime permanent. Quant aux signaux de commande, ils présentent des allures lisses. La contribution du terme flou dans la loi de commande apparaît clairement à travers le signal  $uf$ . Les allures des gains  $k_{pi}$ ,  $k_{vi}$  et  $a_{li}$  sont lisses.

Un test de variation de charge a été également effectuée. Dans ce test l'élément terminal manipule une charge de 2 Kg puis de 4 Kg. cette charge est relâchée à l'instant  $t=0.75$  sec. Les réponses obtenues sont montrées par la figure III.4. On constate que les articulations continuent remarquablement la poursuite des trajectoires.

Une variation paramétrique de +50% puis de +100 % est aussi introduite à l'instant  $t=0.75$  sec. La figure III.5 illustre les résultats obtenus. On remarque une bonne poursuite des trajectoires désirées. Les commandes sont adaptés rapidement dès l'introduction de la variation afin d'assurer la convergence des erreurs de poursuite.

Pour montrer l'importance de la décentralisation dans le schéma de commande, nous avons effectué le même test de rupture que celui du chapitre précédent en annulant la commande du second sous système à l'instant  $t=0.75$  sec. Les réponses obtenues sont montrées par la figure III.6. Nous constatons que la première et troisième articulation continue la poursuite des trajectoires avec une légère augmentation de l'erreur qui reste inférieure à 0.3 degré. Le système est soumis à des sollicitations de commandes plus grandes au niveau des autres articulations. Cette dernière, ayant une dynamique lente, impose un temps plus longs pour l'établissement du régime permanent.

Table ( III.1) : Paramètres du régulateur sous la loi CADLF

	Symboles des paramètres	Première articulation	Deuxième articulation	Troisième articulation
Erreur Filtrée $r_i$	$p_{1i} ; p_{2i}$	3000 ; 500	3000 ; 500	3000 ; 500
Action PD	$\alpha_{1i} ; \alpha_{2i}$	930 ; 9.3	930 ; 9.3	930 ; 9.3
Adaptative $k_{pi} e_i + k_{vi} \dot{e}_i$	$\beta_{1i} ; \beta_{2i}$	930 ; 9.3	930 ; 9.3	930 ; 9.3
Facteur de normalisation	$q_i ; \dot{q}_i ; \ddot{q}_i^d$	4 ; 5.33 ; 46	4 ; 5.33 ; 46	4 ; 5.33 ; 46
Modèle flou $Z_i^T \hat{A}_i(t) \xi_i$	$\Gamma_{1i} ; \Gamma_{2i}$	$30 \times \mathbf{1}_{1 \times 8} ; 0 \times \mathbf{1}_{1 \times 8}$	$50 \times \mathbf{1}_{1 \times 8} ; 0 \times \mathbf{1}_{1 \times 8}$	$50 \times \mathbf{1}_{1 \times 8} ; 0 \times \mathbf{1}_{1 \times 8}$



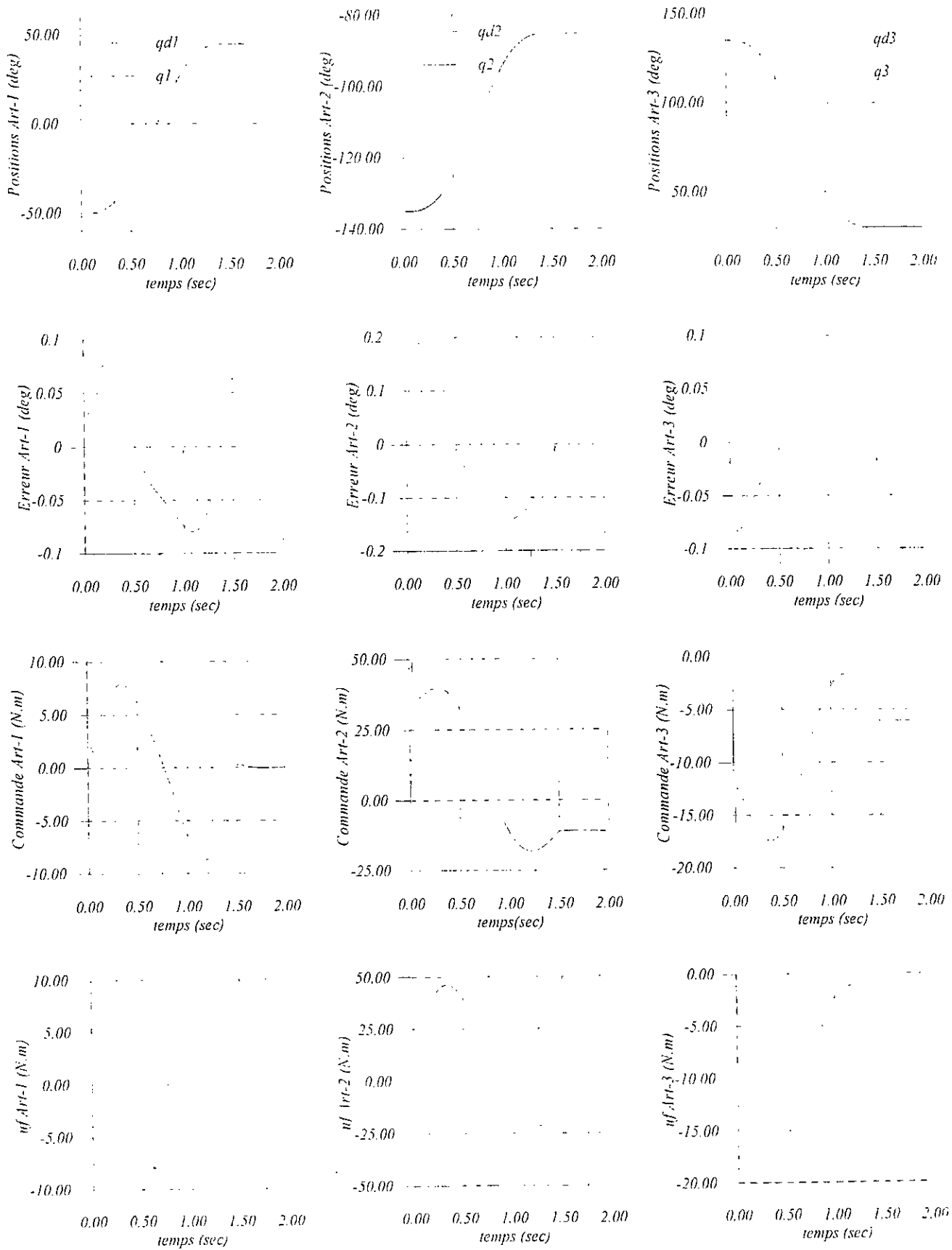


Figure III.3.a . Réponses temporelles des différentes articulations en utilisant la loi de commande CADLF (Test à vide)

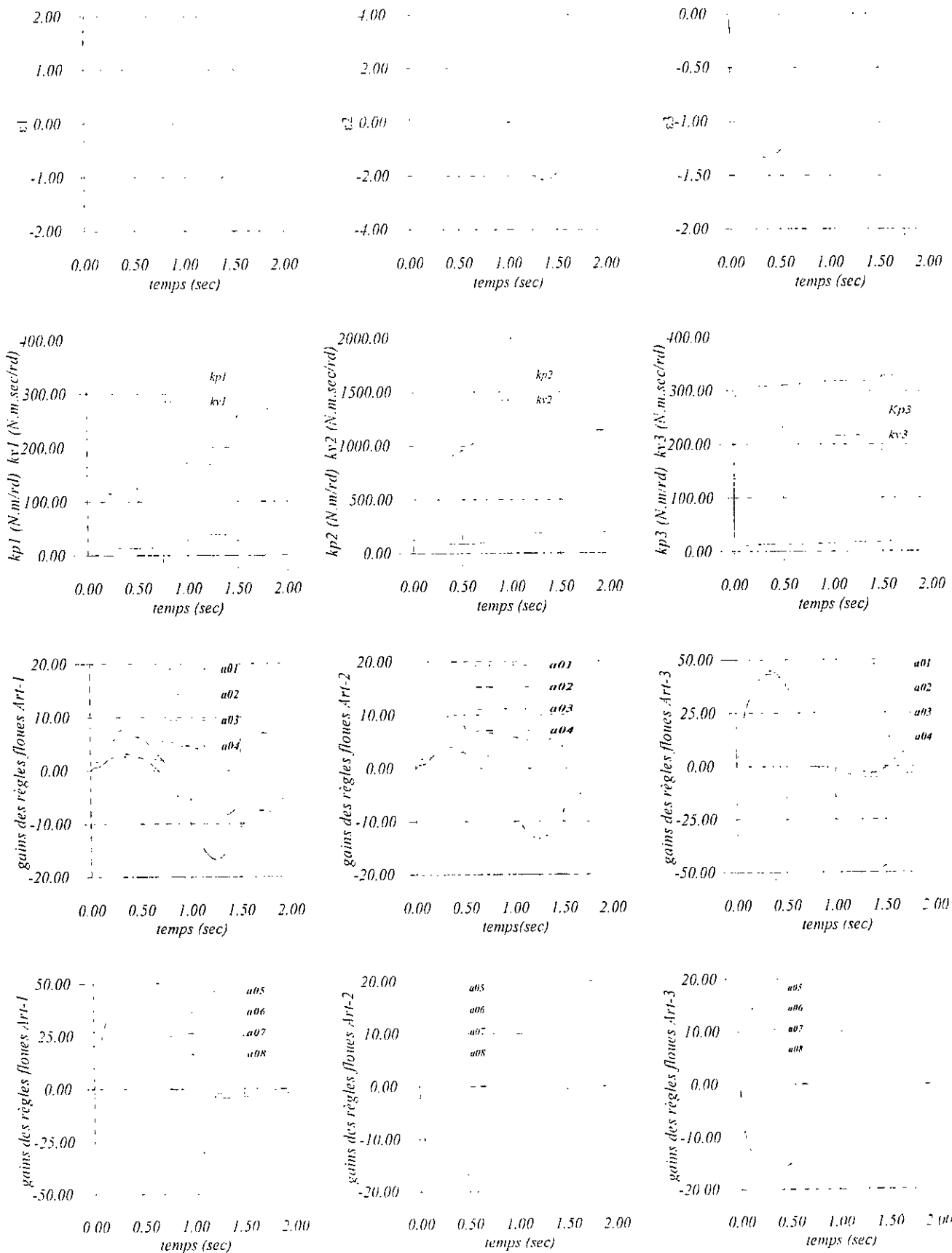


Figure III.3.b . Evolution des différentes signaux adaptatifs en utilisant la loi de commande CADLF (Test à vide)

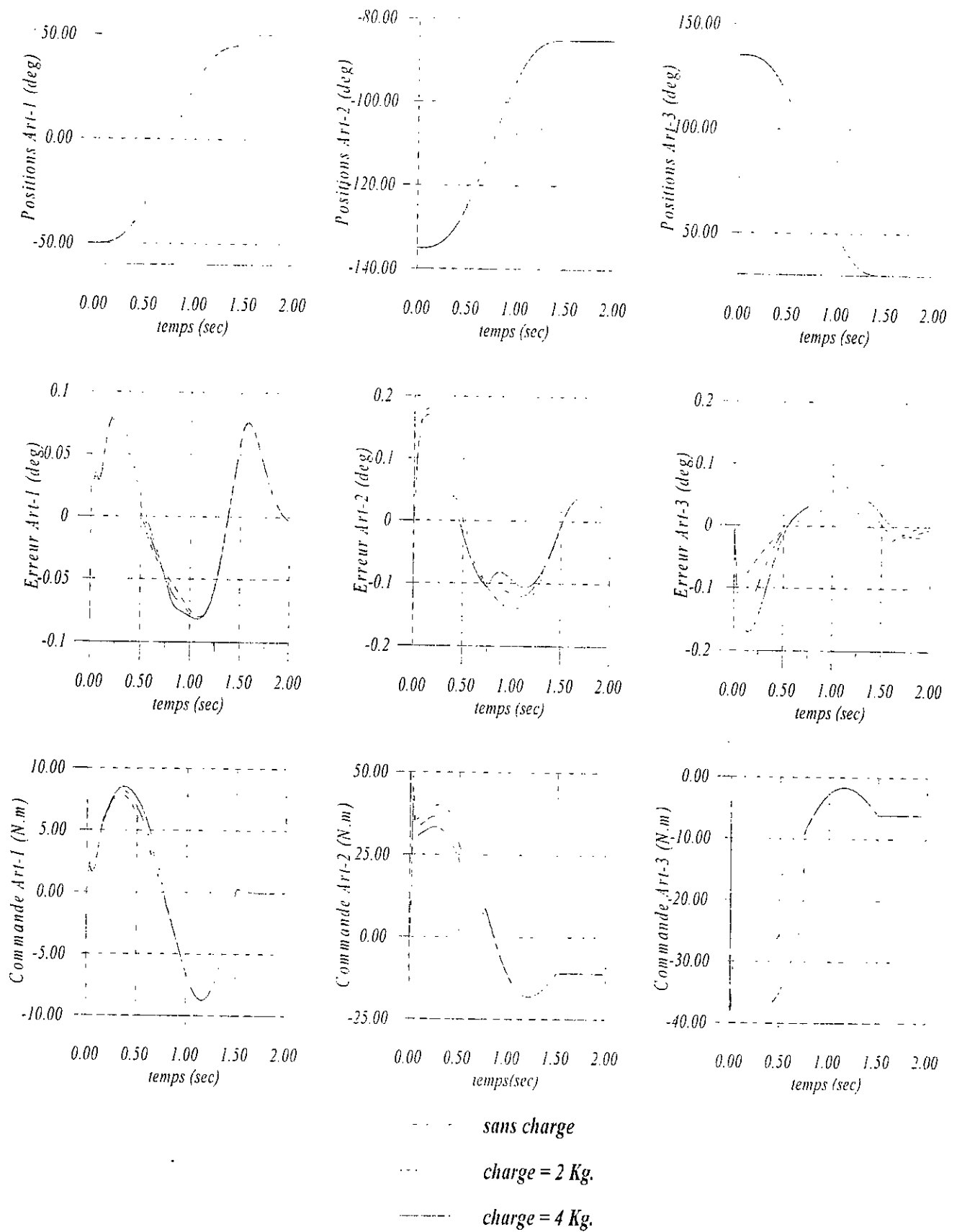
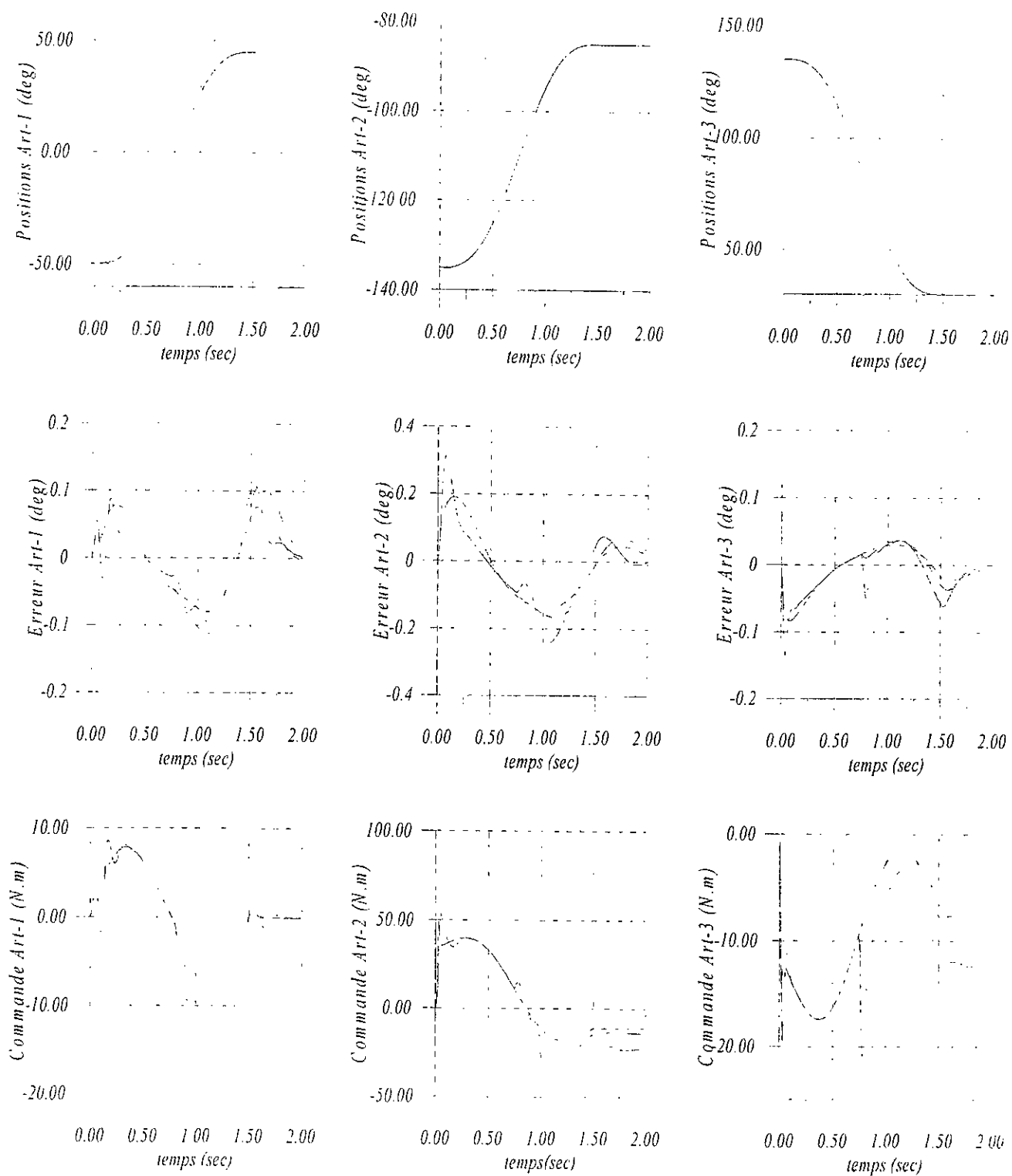


Figure III.4 . Réponses temporelles des différentes articulations en utilisant la loi de commande CADLF (test de variation de charge).



— sans variation.

- - - variation paramétrique de +50 %.

... variation paramétrique de +100 %.

Figure III.5 . Réponses temporelles des différentes articulations en utilisant la loi de commande CADLF (test de variation paramétrique).

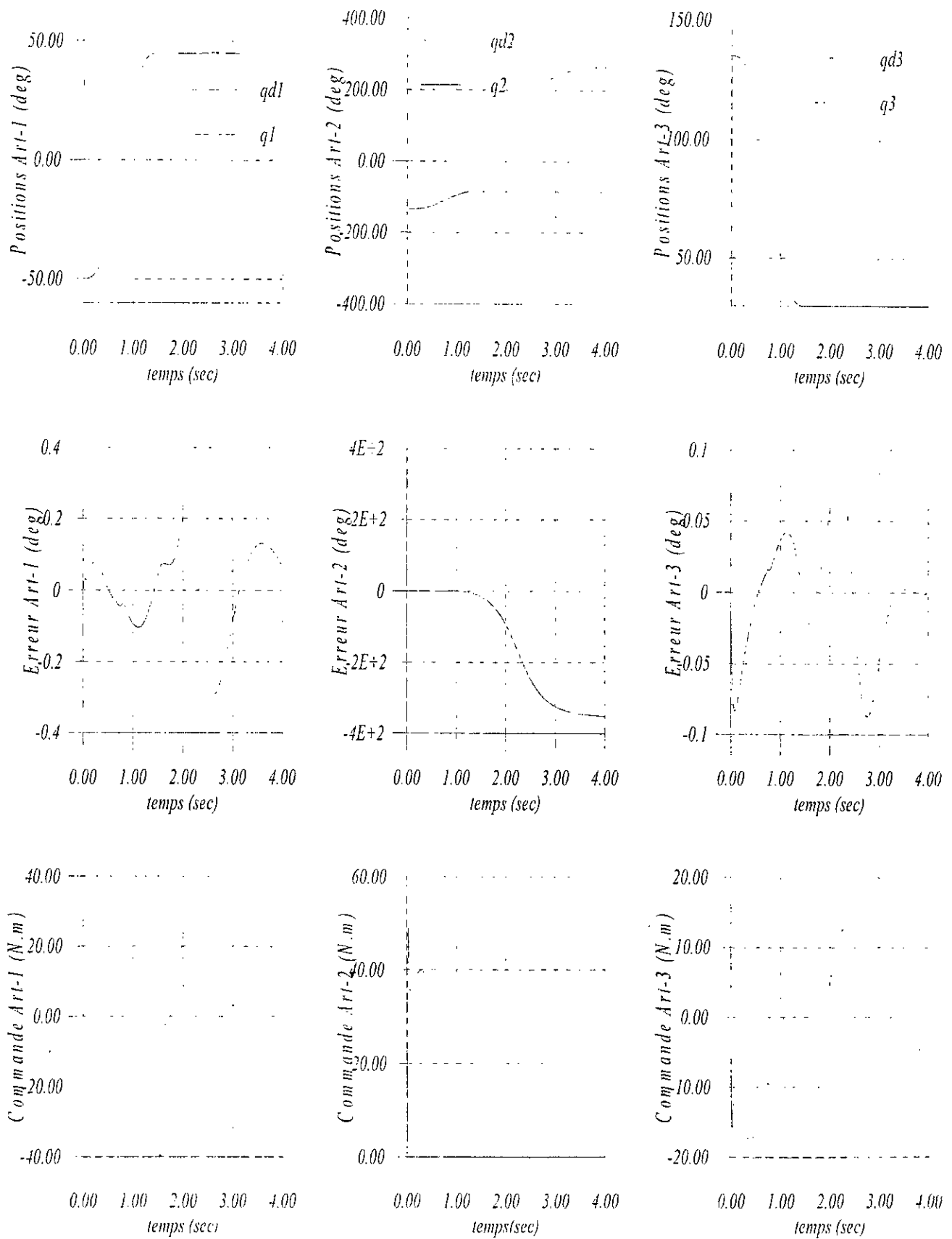


Figure III.6 . Réponses temporelles des différentes articulations en utilisant la loi de commande CADLF (test de rupture de liaison).

### III.5. Conclusion

Dans ce chapitre, nous avons proposé une nouvelle approche de commande adaptative décentralisée pour la commande des robots en utilisant la logique floue. Le développement de la méthode est basé sur l'approximation des fonctions non linéaires par le modèle flou de type Sugeno, ainsi que la théorie de stabilité de Lyapunov.

En effet, le schéma de commande proposé assure la stabilité et la convergence des différents signaux du système bouclé. L'utilisation de cet outil mathématique permet une certaine systématisation dans l'établissement de la loi de commande, ce qui n'est pas toujours le cas lors de l'utilisation de la logique floue. D'autre part, la loi de commande proposée présente un caractère générale du fait qu'on n'utilise pas le modèle explicite du système lors de la synthèse. Cette caractéristique intéressante rend possible l'utilisation de ce schéma pour n'importe quelle système mécanique pouvant être mis sous la forme (III.5).

Les résultats de simulation effectuée sur le robot PUMA 560 ont montré l'applicabilité de la méthode proposée. La convergence des différents signaux et paramètres a été vérifiée. Les tests de performances ont montré la robustesse de la commande vis à vis des variations paramétriques et de la décentralisation.

En fin, Il est à noter que cette méthode ouvre de nouvelles perspectives pour l'élaboration des schémas de commande par logique floue des systèmes non linéaires d'une manière générale, en utilisant l'analyse de la stabilité.

## **CHAPITRE IV**

Commande adaptative décentralisée par  
réseaux de neurones

## CHAPITRE IV

### Commande décentralisée adaptative par réseaux de neurones

#### IV .1. Introduction

L'apparition des réseaux de neurones artificiels a ouvert des perspectives prometteuses dans le développement technologique. Grâce à leurs capacités d'approximation des fonctions non linéaires, ils ont été utilisés avec beaucoup de succès dans le domaine de l'identification et de la commande des systèmes physiques, particulièrement les systèmes non linéaires [NAR90a], [NAR90b]. Parmi les méthodes développées dans ce sens, on peut citer entre autres :

- la commande supervisée [HUN92], [MIL90],[HAM97],[BOU01] [BOU98] [BOU99 ], Le réseau recopie une loi de commande existante afin d'améliorer les performances grâce aux capacités de généralisation dont il jouisse.
- La commande inverse [HUN92], [MIL90], [NAR90a]. Cette commande consiste à rechercher d'abord un modèle inverse du processus qu'on apprend au réseau. Ce dernier sera placé en série avec le processus à commander.
- La commande à régulateur auto-ajustable [CHE90], [BOU96a], [BOU96b],[BOU97], [BOU98a, [BOU98b]. Dans cette structure de commande, un modèle suffisamment précis du système doit être établi afin de l'utiliser pour la construction d'une stratégie de commande. Le réseau de neurone est entraîné pour modéliser le système. Les sorties des réseaux seront utilisées pour l'optimisation d'un critère, sur la base duquel la commande est développée.
- La commande adaptative avec modèle de référence (MRAC) [NAR90a]. Le réseau est utilisé pour que le système suive un modèle de référence. L'erreur de poursuite est utilisée pour ajuster les poids du réseau.

Il est à noter que pour toutes les approches citées ci dessus aucune analyse de la stabilité du système n'est établie. Cependant, l'apparition des travaux de Lewis [LEW93a],[LEW93b] a donné un grand souffle pour la commande par réseaux de neurones avec preuve de la stabilité. En se basant sur ses travaux, un certain nombre de résultats ont été obtenus [KWA95a], [KWA95b], [ZIA96], [SHU96a], [SHU96b].

Dans ce chapitre, nous proposons une nouvelle approche de commande décentralisée par réseaux de neurones adaptatives des robots manipulateurs. Cette technique de commande constitue une extension au cas décentralisée de celle déjà développée par Lewis dans le contexte centralisé. Dans l'approche proposée, nous utilisons la théorie d'approximation des fonctions par les réseaux de neurones. La stabilité du système en présence des réseaux de neurones sera démontrée en utilisant la théorie de Lyapunov. Afin de tester le faisabilité et les performances du schéma proposé, une application pour la commande en poursuite de trajectoire du robot PUMA560 est effectuée.



## IV.2. Modèle dynamique décentralisé

Etant donné le modèle dynamique décentralisé du robot de l'équation (III.2) mis sous la forme suivante :

$$m_{ii}(q) \ddot{q}_i(t) + b_{ii}(q, \dot{q}) \dot{q}_i(t) + g_i(q) + \bar{c}_i(q, \dot{q}, \ddot{q}) = u_i(t) \quad i = 1, \dots, n \quad (\text{IV.1})$$

Avec

$$\bar{c}_i(q, \dot{q}, \ddot{q}) = \sum_{\substack{j=1 \\ j \neq i}}^n m_{ij}(q) \ddot{q}_j(t) + \sum_{\substack{j=1 \\ j \neq i}}^n b_{ij}(q, \dot{q}) \dot{q}_j(t) + k_{ii}(q, \dot{q}) q_i(t) + \sum_{\substack{j=1 \\ j \neq i}}^n k_{ij}(q, \dot{q}) q_j(t) - u_{pi}(t) \quad (\text{IV.2})$$

$$i = 1, \dots, n$$

Ce modèle caractérise la dynamique de chaque articulation en tenant compte des interactions avec le reste du robot.

Ces équations peuvent être exprimées en terme d'erreur de poursuite filtrée

$$r_i(t) = \dot{e}_i(t) + \lambda_i e_i(t); \quad \lambda_i \in \mathfrak{R}^+ \quad (\text{IV.3})$$

d'où

$$\dot{r}_i(t) = \ddot{e}_i(t) + \lambda_i \dot{e}_i(t) \quad i = 1, \dots, n \quad (\text{IV.4})$$

En multipliant (IV.4) de part et d'autre par  $m_{ii}(q)$ , nous obtenons

$$m_{ii}(q) \dot{r}_i(t) = m_{ii}(q) \ddot{q}_{di}(t) - m_{ii}(q) \ddot{q}_i(t) \quad (\text{IV.5})$$

En remplaçant le terme  $m_{ii}(q) \ddot{q}_i(t)$  par son expression de (IV.1) et après quelques manipulations mathématiques nous obtenons la dynamique de l'erreur filtrée locale

$$m_{ii}(q) \dot{r}_i(t) = -\frac{1}{2} b_{ii}(q, \dot{q}) r_i(t) + f_i(x_i) + \bar{c}_i(q, \dot{q}, \ddot{q}) - u_i(t) \quad (\text{IV.6})$$

avec

$$f_i(x_i) = -\frac{1}{2} b_{ii}(q, \dot{q}) r_i(t) + b_{ii}(q, \dot{q}) (\dot{q}_{di}(t) + \lambda_i \dot{e}_i(t)) + m_{ii}(q) (\ddot{q}_{di}(t) + \lambda_i \dot{e}_i(t)) + g_i(q) \quad (\text{IV.7})$$

où  $x_i = (e_i, \dot{e}_i, q_{di}, \dot{q}_{di}, \ddot{q}_{di})^T$

Cette dernière équation représente une fonction couple non linéaire.

### IV.3. Commande décentralisée par réseaux de neurones adaptatifs

Le problème consiste à synthétiser une loi de commande locale à base de réseaux de neurones adaptatifs assurant ainsi la convergence asymptotique de l'erreur de poursuite et la bornitude des différents paramètres intervenant dans la loi de commande. Pour cela, nous proposons la loi de commande suivante :

$$u_i(t) = \hat{f}_i(x_i) + k_{v_i} r_i(t) + v_{r_i}(t) \quad (\text{IV.8})$$

$\hat{f}_i(x_i)$  est une estimation en temps réel de la fonction couple non linéaire

En remplaçant (IV.8) dans (IV.6) nous obtenons l'équation de l'erreur en boucle fermée

$$m_{ii}(q) \dot{r}_i(t) = -(k_{v_i} + \frac{1}{2} b_{ii}(\dot{q}, \ddot{q})) r_i(t) + f_i(x_i) - \hat{f}_i(x_i) - v_{r_i}(t) + \bar{c}_i(q, \dot{q}, \ddot{q}) \quad (\text{IV.9})$$

Sachant que les réseaux de neurones artificiels jouissent de la propriété d'approximateurs universels de fonctions non linéaires, nous allons les utiliser pour l'approximation de la fonction  $f_i(x_i)$  donnée dans (IV.9).

#### IV.3.1. Approximation des fonctions non linéaires par les réseaux de neurones à couches

Dans cette partie, nous montrons comment les réseaux à couches sont utilisés pour l'approximation des fonctions non linéaires  $f_i(x_i)$ . Ceci est basé principalement sur le théorème suivant :

**Théorème [BEL97]:**

Soit  $\phi(x)$  une fonction non constante, bornée, monotone et croissante. Soit  $K$  une région bornée de  $R^n$  et  $F(x_1, \dots, x_n)$  une fonction réelle continue définie sur  $K$ . Alors, pour tout  $\varepsilon > 0$ , il existe un entier  $N$  et des constantes réelles  $C_i$ ,  $\theta_i$  ( $i=1, \dots, N$ ) et  $W_{ij}$  ( $i=1, \dots, N$ ;  $j=1, \dots, N$ ) tels que :

$$\hat{F}(x_1, \dots, x_N) = \sum_{i=1}^N C_i \phi(\sum_{j=1}^N W_{ij} x_j + \theta_i) \quad (\text{IV.10})$$

satisfait

$$\max(|F(x_1, \dots, x_N) - \hat{F}(x_1, \dots, x_N)|) < \varepsilon$$

□

La forme de l'équation (IV.10) indique que la fonction peut être approximée par un réseau de neurones à trois couches dont la structure est donnée par la figure IV.1.

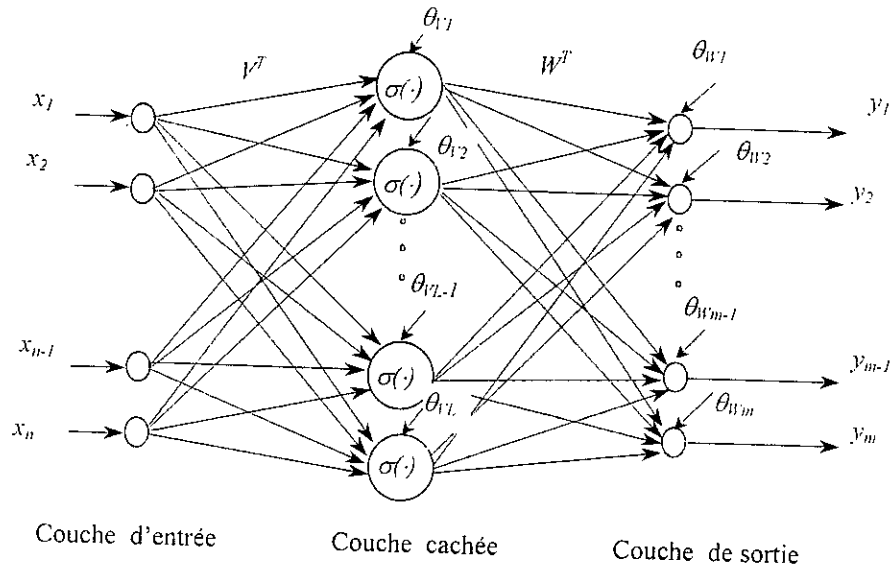


Figure IV.1. Structure d'un réseau de neurone à trois couches.

Chaque sortie du réseau est exprimée par

$$y_i = \sum_{j=1}^l \left[ W_{ij} \cdot \sigma \left[ \sum_{k=1}^n V_{jk} x_k + \theta_{Vj} \right] + \theta_{Wi} \right] \quad i=1, \dots, m \quad (\text{IV.11})$$

Sous forme matricielle l'expression (IV.11) devient

$$Y = W^T \cdot \sigma(V^T \cdot X + \Theta_V) + \Theta_W \quad (\text{IV.12})$$

Avec

$$Y \in \mathfrak{R}^m, X \in \mathfrak{R}^n, \quad \sigma(Z) = [\sigma(Z_1) \dots \sigma(Z_l)], Z \in \mathfrak{R}^l$$

- $\sigma(\cdot)$  : Fonction d'activation (sigmoïde).
- $V^T \in \mathfrak{R}^{l \times n}$  : Matrice des poids (entrée-couche cachée).
- $W^T \in \mathfrak{R}^{m \times l}$  : Matrice des poids (couche cachée-sortie).
- $\theta_{Vj} \quad \theta_{Wj}$  : Paramètres du biais.

Pour plus de détail sur les réseaux de neurones, le lecteur est amené à consulter le chapitre I ainsi que les références spécialisées entre autres [NAR 90a], [MIL92], [FRE92],...

Pour chaque loi de commande locale dans le schéma de commande en boucle fermée, un réseau de neurone est utilisé afin d'estimer en temps réel la fonction couple non linéaire  $f_i(x_i)$ .

En supposant qu'il existe des poids idéaux  $W_i$  et  $V_i$  du réseau permettant la reconstruction de la fonction  $f_i(x_i)$ , cette dernière peut être exprimée en terme de réseau comme suite:

$$f_i(x_i) = W_i^T \cdot \sigma_i(V_i^T \cdot x_i) + \varepsilon_i(x_i) \quad (\text{IV.13})$$

où  $\varepsilon_i(x_i)$  est une erreur de reconstruction supposée bornée

On désigne l'estimée de la fonction  $f_i(x_i)$  par

$$\hat{f}_i(x_i) = \hat{W}_i^T \cdot \sigma_i(\hat{V}_i^T \cdot x_i) \quad (\text{IV.14})$$

$\hat{W}_i$  et  $\hat{V}_i$  sont les estimées des poids  $W_i$  et  $V_i$  respectivement.

Les notations suivantes sont adoptées pour les erreurs d'estimation des différentes variables pour  $i=1, \dots, n$  :

- Erreur de l'estimation de la fonction

$$\tilde{f}_i(x_i) = f_i(x_i) - \hat{f}_i(x_i) \quad (\text{IV.15})$$

- Erreur de l'estimation des poids du réseau

$$\begin{cases} \tilde{V}_i = V_i - \hat{V}_i \\ \tilde{W}_i = W_i - \hat{W}_i \\ \tilde{Z}_i = Z_i - \hat{Z}_i \end{cases} \quad (\text{IV.16})$$

Avec  $Z_i^T = [W_i^T \ V_i^T]$ .

- Erreur d'estimation de la sortie de la couche cachée

$$\tilde{\sigma}_i = \sigma_i - \hat{\sigma}_i = \sigma_i(V_i^T \cdot x_i) - \sigma_i(\hat{V}_i^T \cdot x_i) \quad (\text{IV.17})$$

Le développement en série de Taylor de la fonction d'activation du réseau i donne

$$\sigma_i(V_i^T \cdot x_i) = \sigma_i(\hat{V}_i^T \cdot x_i) + \sigma_i'(\hat{V}_i^T \cdot x_i) \cdot \tilde{V}_i^T \cdot x_i + O_i(\tilde{V}_i^T \cdot x_i)^2 \quad (\text{IV.18})$$

où

$$\sigma_i'(\hat{z}_i) = \left. \frac{d(\sigma_i(z_i))}{dz_i} \right|_{z = \hat{z}_i}$$

En notant:  $\hat{\sigma}_i' = \sigma_i'(\hat{V}_i^T x_i)$ , nous aurons

$$\tilde{\sigma}_i = \sigma_i'(\hat{V}_i^T \cdot x_i) \cdot \tilde{V}_i^T \cdot x_i + O_i(\tilde{V}_i^T \cdot x_i)^2 = \hat{\sigma}_i' \cdot \tilde{V}_i^T \cdot x_i + O_i(\tilde{V}_i^T \cdot x_i)^2 \quad (\text{IV.19})$$

### IV.3.2. Détermination de la structure de commande

Afin d'établir le schéma de commande, nous donnons les hypothèses et propriétés suivantes nécessaires pour le développement de la méthode de commande.

#### Hypothèses :

**H1:** Les poids idéaux sont bornés c'est-à-dire qu'il existe des constantes  $V_{mi}$ ,  $W_{mi}$  et  $Z_{mi}$  tels que:

$$\begin{cases} \|V_i\|_F \leq V_{mi} \\ \|W_i\|_F \leq W_{mi} \\ \|Z_i\|_F \leq Z_{mi} \end{cases} \quad (\text{IV.20})$$

Où  $\|\cdot\|_F$  désigne la *norme Frobenienne* définie comme suit:

Soit les matrices  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{m \times n}$ , et  $X \in \mathbb{R}^n$

La norme *Frobenienne* a les propriétés suivantes:

$$\|A\|_F^2 = \text{trace}(A^T \cdot A) = \sum_{i,j} a_{ij}^2 \quad (\text{IV.21})$$

$$\langle A, B \rangle_F = \text{trace}(A^T \cdot B) \quad (\text{IV.22})$$

$$\|A \cdot X\|_F \leq \|A\|_F \cdot \|X\| \quad (\text{IV.23})$$

**H2:** Les trajectoires de référence sont bornés par une constante positive  $Q_{di}$ , c'est à dire

$$\left\{ |q_{di}|, |\dot{q}_{di}|, |\ddot{q}_{di}| \right\} \leq Q_{di} \quad (\text{IV.24})$$

**Propriétés :**

**P1:** Pour la fonction sigmoïde, le terme du 2<sup>ème</sup> ordre dans le développement en série de Taylor défini dans (IV.18) vérifie l'équation suivante :

$$\left\| O_i(\tilde{V}_i^T \cdot x_i) \right\|^2 \leq c_{i1} + c_{i2} \cdot \|\tilde{V}_i\|_F \quad (\text{IV.25})$$

$c_{i1}, c_{i2}$  des constantes positives. La preuve de cette propriété est donnée dans [LEW95].

Pour le bras manipulateur défini par (IV.1) et (IV.2) nous avons pour  $i=1, \dots, n$ :

**P2:**  $\bar{m}_{ii} \leq m_{ii}(q) \leq \bar{m}_{i2}$ ,  $\bar{m}_{i1}, \bar{m}_{i2}$  constantes positives.

**P3:**  $b_{ii}(q, \dot{q}_i) \leq b_{hi}(q_i) |\dot{q}_i|$  avec  $b_{hi}(q_i) \in C^1$ .

**P4:**  $\dot{m}_{ii}(q) = b_{ii}(q, \dot{q}_i)$ .

**P5:**  $u_{pi} < b_{pi}$ ,  $b_{pi}$  sont des constantes positives.

En remplaçant dans l'équation (IV.15)  $f_i(x_i)$  et  $\hat{f}_i(x_i)$  par leurs expressions respectives (IV.13) et (IV.14) et en tenant compte de (IV.18) nous aurons

$$\tilde{f}_i(x_i) = \tilde{W}_i^T \cdot \sigma_i(\hat{V}_i^T \cdot x_i) + \hat{W}_i^T \cdot \sigma_i'(\hat{V}_i^T \cdot x_i) \tilde{V}_i^T \cdot x_i + \tilde{W}_i^T \cdot \sigma_i'(\hat{V}_i^T \cdot x_i) \tilde{V}_i^T \cdot x_i + W_i^T \cdot O_i(\tilde{V}_i^T \cdot x_i)^2 + \varepsilon_i(x_i) \quad (\text{IV.26})$$

En remplaçant (IV.26) dans (IV.9) la dynamique de l'erreur en boucle fermée devient

$$m_{ii}(q) \dot{r}_i(t) = -(k_{wi} + \frac{1}{2} b_{ii}(q, \dot{q})) r_i(t) + \tilde{W}_i^T \cdot \sigma_i(\hat{V}_i^T \cdot x_i) + \hat{W}_i^T \cdot \sigma_i'(\hat{V}_i^T \cdot x_i) \tilde{V}_i^T \cdot x_i + \omega_i(t) - v_{ri}(t) \quad (\text{IV.27})$$

avec

$$\omega_i(t) = \tilde{W}_i^T \cdot \sigma_i'(\hat{V}_i^T \cdot x_i) \tilde{V}_i^T \cdot x_i + W_i^T \cdot O_i(\tilde{V}_i^T \cdot x_i)^2 + \varepsilon_i(x_i) + \bar{c}_i(q, \dot{q}, \ddot{q}) \quad (\text{IV.28})$$

Supposons que cette dernière fonction est majorée comme suit [LEW95] :

$$|\omega_i(t)| \leq S_i \varphi_i \quad (\text{V.29})$$

avec

$$S_i \varphi_i = \zeta_{i0} + \zeta_{i1} \|z_i\| + \zeta_{i2} \|z_i\|^2 + \zeta_{i3} \|\hat{W}_i\|_F + \zeta_{i4} \|\hat{V}_i\|_F + \zeta_{i5} \|\tilde{V}_i\|_F \|\hat{W}_i\|_F$$

et

$$z_i = [e_i \ r_i]^T$$

$$S_i = \left[ 1 \quad \frac{\|z_i\|}{\|z_i\|^2} \|\hat{W}_i\|_F \|\hat{V}_i\|_F \|\hat{W}_i\|_F \times \|\hat{V}_i\|_F \right]$$

$$\varphi_i = [\zeta_{i0} \ \zeta_{i1} \ \zeta_{i2} \ \zeta_{i3} \ \zeta_{i4} \ \zeta_{i5}]^T$$

Les éléments du dernier vecteur sont des constantes positives dépendant des caractéristiques physiques du robot et de la borne des perturbations et des trajectoires.

Le problème de la commande adaptative décentralisé par réseau de neurone consiste à synthétiser pour chaque articulation (sous système) une loi de commande adaptative de la forme (IV.8), en tenant compte de (IV.14), afin d'assurer la stabilité du système en boucle fermée.

Si on choisit le signal  $v_{Ri}(t)$  (terme robuste) et les lois d'adaptation des différentes variables comme suit :

$$v_{Ri} = \frac{r_i (S_i \hat{\varphi}_i)^2}{(S_i \hat{\varphi}_i)^T r_i + \delta_i} \quad (IV.30)$$

$$\begin{aligned} \dot{\delta}_i &= -\gamma_i \delta_i \\ \delta_i(0) &> 0 \text{ et } \gamma_i > 0 \end{aligned} \quad (IV.31)$$

$$\dot{\hat{\varphi}}_i = \Gamma_i S_i^T |r_i| = -\tilde{\varphi}_i \quad (IV.32)$$

et si l'algorithme d'ajustement des poids est réalisé par

$$\begin{cases} \dot{\hat{W}}_i = F_{wi} \cdot \hat{\sigma}'_i r_i^T \\ \dot{\hat{V}}_i = F_{vi} \cdot x_i \cdot (\hat{\sigma}'_i \cdot \hat{W}_i \cdot r_i)^T \end{cases} \quad (IV.33)$$

avec  $\Gamma_i = \Gamma_i^T > 0$ ,  $F_{wi} = F_{wi}^T > 0$ ,  $F_{vi} = F_{vi}^T > 0$

on peut démontrer que l'erreur de poursuite en boucle fermée tend asymptotiquement vers zéro est les poids des réseaux restent bornés.

Le schéma correspondant à la loi de commande développée est illustré par la figure IV.2.

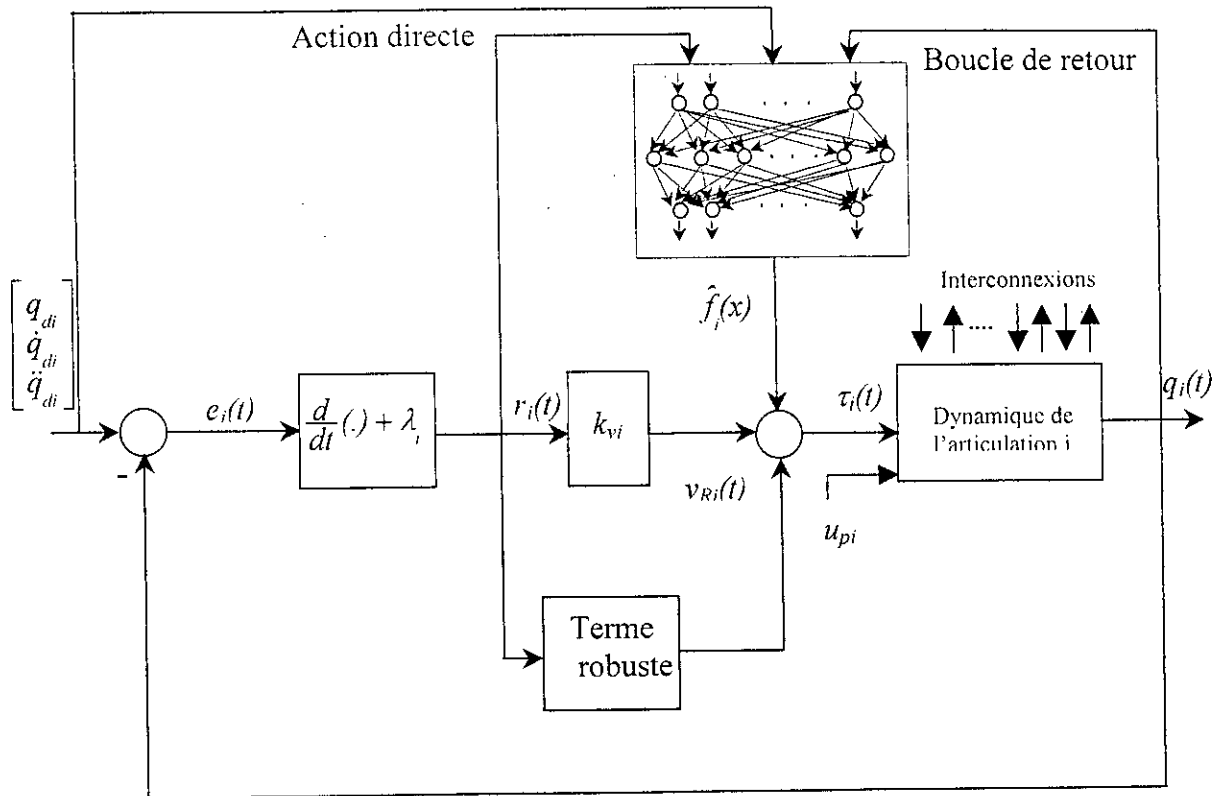


Figure IV.2. Schéma de structure de commande adaptative par réseau de neurones artificiel.

### Preuve de la stabilité

Considérons la fonction de Lyapunov suivante :

$$L_i = \frac{1}{2} m_{ii} r_i^2 + \frac{1}{2} \text{tr}(\tilde{W}_i^T F_{wi}^{-1} \tilde{W}_i) + \frac{1}{2} \text{tr}(\tilde{V}_i^T F_{vi}^{-1} \tilde{V}_i) + \frac{1}{2} \tilde{\varphi}_i^T \Gamma_i^{-1} \tilde{\varphi}_i + \frac{\delta}{\gamma_i} \quad (\text{IV.34})$$

En dérivant cette fonction par rapport au temps et en utilisant (IV.27), nous obtenons

$$\begin{aligned} \dot{L}_i = & -k_{vi} r_i^2 + \frac{1}{2} r_i (m_{ii} - b_{mii}) - r_i v_{Ri} - \delta_i - |r_i| S \tilde{\varphi}_i + r_i w_i + \text{tr}(\tilde{W}_i^T F_{wi}^{-1} \dot{\tilde{W}}_i + \tilde{W}_i^T \hat{\sigma}_i r_i) + \\ & \text{tr}(\tilde{V}_i^T F_{vi}^{-1} \dot{\tilde{V}}_i + \tilde{V}_i^T x_i (\hat{\sigma}_i \hat{W}_i r_i)^T \end{aligned} \quad (\text{IV.35})$$



En tenant compte de la propriété P4

De l'équation (IV.29), et sachant que  $\tilde{\varphi}_i = \varphi_i - \hat{\varphi}_i$ , nous pouvons écrire

$$r_i \omega_i - |r_i| S_i \varphi_i \leq 0 \quad (\text{IV.36})$$

Il résulte que

$$\begin{cases} L_i \leq -\frac{1}{2} k_{v_i} r_i^2 + L_{i1} + L_{i2} \\ L_{i1} = |r_i| S_i \hat{\varphi}_i - r_i v_{Ri} - \delta_i \\ L_{i2} = \text{tr}(\tilde{W}_i^T F_{v_i}^{-1} \tilde{W}_i + \tilde{W}_i^T \hat{\sigma}_{r_i}) + \text{tr}(\tilde{V}_i^T F_{v_i}^{-1} \tilde{V}_i + \tilde{V}_i^T x_i(\hat{\sigma}_{r_i} \hat{W}_{r_i})) \end{cases} \quad (\text{IV.37})$$

En remplaçant dans (IV.37)  $v_{Ri}$  par son expression (IV.30) nous aurons :

$$L_{i1} = \frac{\delta_i |r_i| S_i \hat{\varphi}_i}{(S_i \hat{\varphi}_i) |r_i| + \delta_i} - \delta_i \leq 0 \quad (\text{IV.38})$$

Sachant que  $\tilde{W}_i = W_i - \hat{W}_i$ ,  $\tilde{V}_i = V_i - \hat{V}_i$  et en appliquant les lois d'adaptation paramétrique

(IV.33) dans l'expression de  $L_{i2}$ , nous pouvons trouver aisément

$$\dot{L}_{i2} = 0 \quad (\text{IV.39})$$

des équations (IV.37), (IV.38), (IV.39) nous déduisons

$$\dot{L}_i \leq 0 \quad (\text{IV.40})$$

L'équation (IV.40) implique que  $L_i$ ,  $r_i$ ,  $W_i$ ,  $V_i$  et  $\varphi_i$  sont bornés. On peut aussi montrer que  $\dot{r}_i$  est bornée c'est à dire que  $r_i$  est uniformément continue.

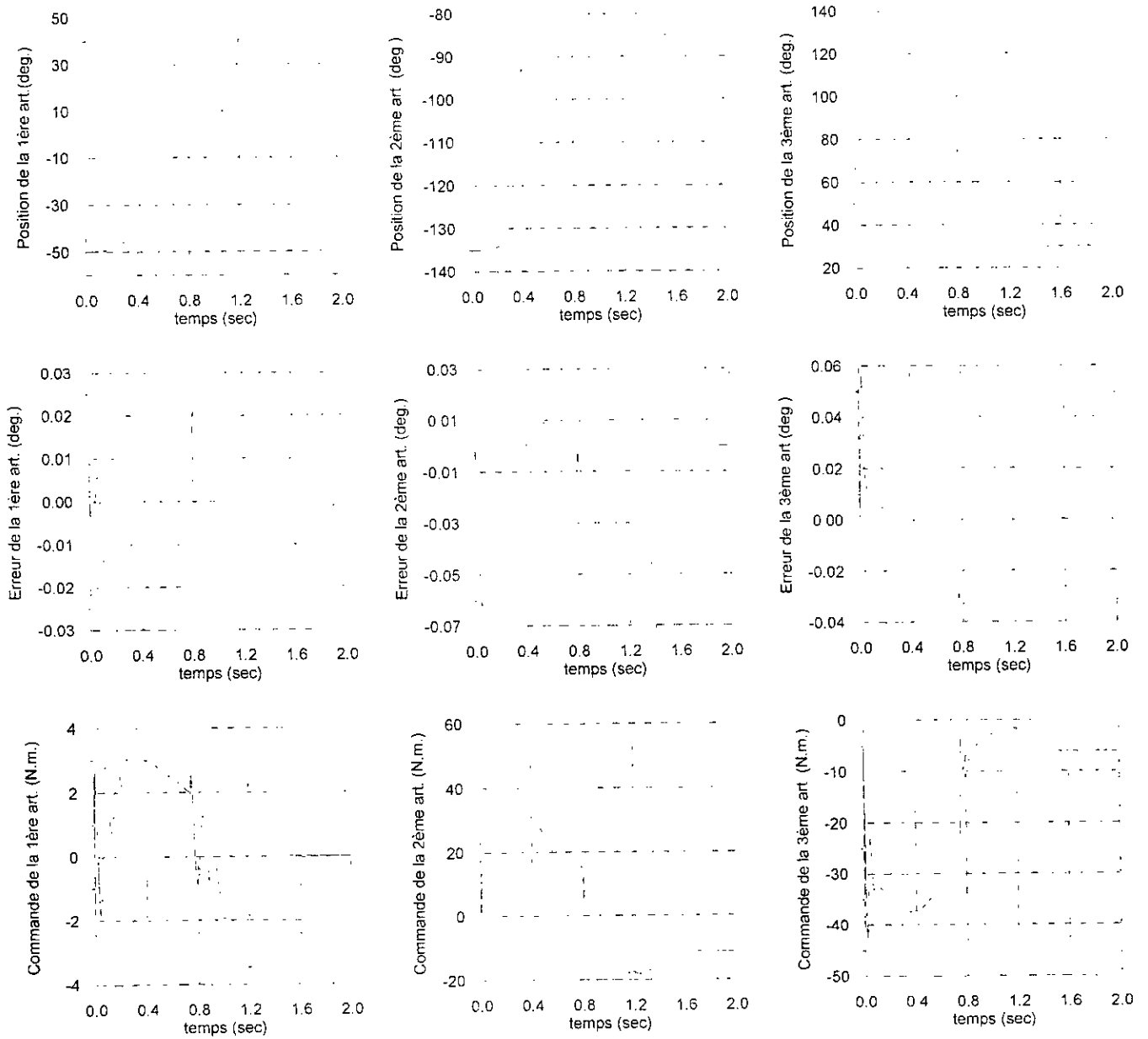


Figure IV.3. Réponses temporelles des différentes articulations en utilisant la commande adaptative par RNA (test de variation de la charge de 5 Kg).

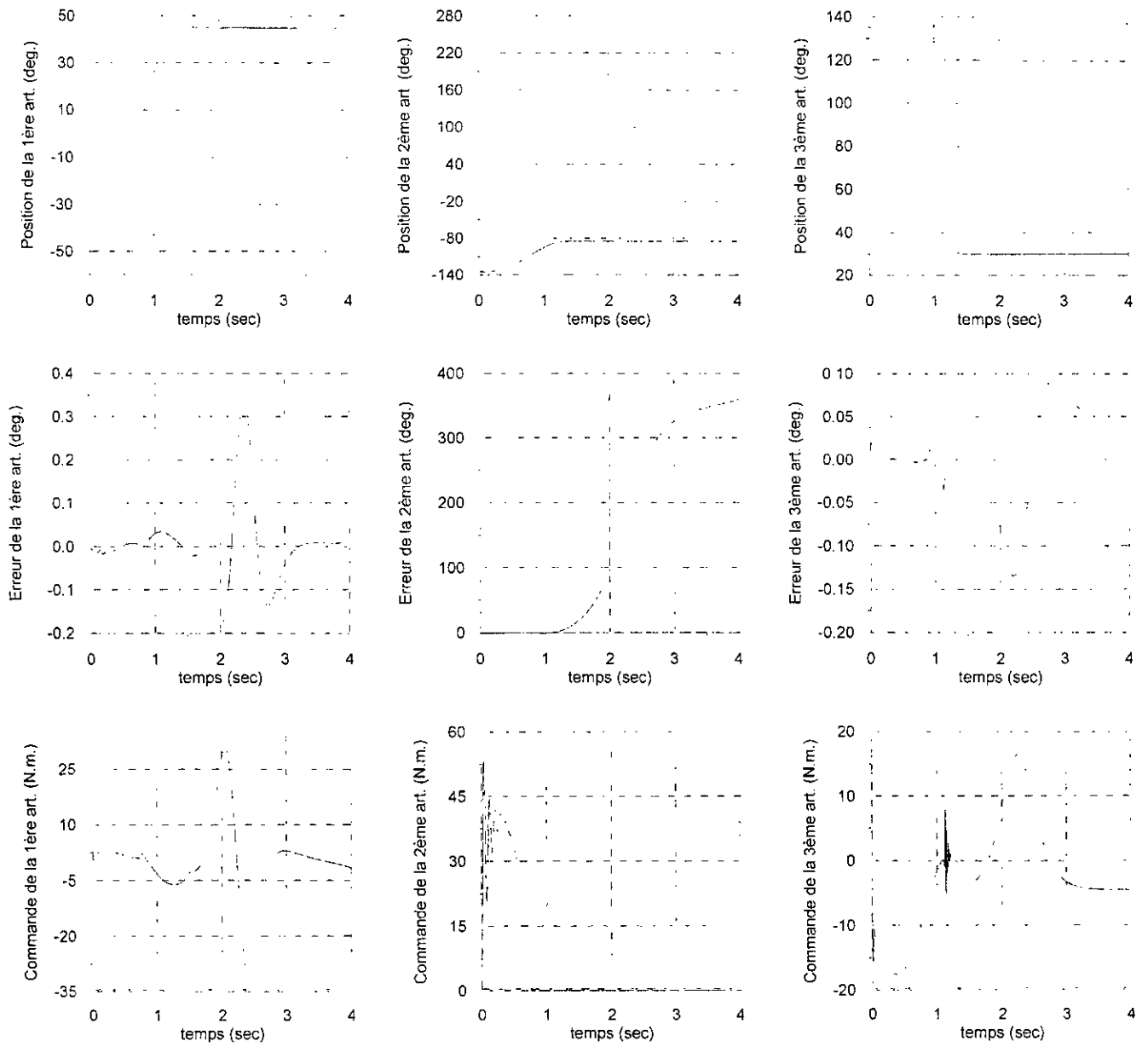


Figure IV.4. Réponses temporelles des différentes articulations en utilisant la commande adaptative par RNA (test de rupture de liaison)

#### IV.5. Conclusion

Dans ce chapitre, nous avons développé une nouvelle approche de commande décentralisée adaptative à base de réseaux de neurones à couches. La méthode proposée constitue une extension au cas décentralisée d'un schéma déjà développé pour la commande centralisée. Dans la structure proposée, chaque unité de commande locale utilise une loi de commande composée de trois termes. Le premier terme représente un régulateur à action proportionnelle et dérivée à gains fixes. Le second terme est un signal délivré par un réseau de neurones adaptatif estimant une fonction couple non linéaire. Le dernier terme est introduit pour compenser au niveau de chaque sous système l'effet des erreurs de constructions, des perturbations, des dynamiques non modélisées ainsi que l'effet des interconnexions avec les autres sous systèmes.

La stabilité du schéma de commande est prouvée en se basant sur des hypothèses et propriétés liées aux modèles du robot et la théorie de Lyapunov.

La loi de commande proposée a été appliquée pour la commande en poursuite de trajectoire d'un robot de type PUMA560 où la robustesse a été prouvée. Il est à noter que dans cette structure, les réseaux de neurones ne nécessitent aucun apprentissage *off line*. L'adaptation se fait on line et les poids pourront être initialisée à zéro.

En outre, contrairement aux méthodes de commandes adaptatives classiques, la synthèse de la commande ne nécessite pas la connaissance préliminaire du modèle du robot. On peut également envisager l'utilisation d'autres types de réseaux de neurones comme les réseaux à bases radiales RBF, les réseaux récurrents ou dynamiques [KOS92], [KAR93]. Néanmoins, la preuve de la stabilité doit se conformer aux caractéristiques de ces réseaux.

## **CHAPITRE V**

Commande décentralisée supervisée par réseaux  
de neurones

## Decentralized neural network controller for trajectory tracking of robot manipulators

D. BOUKHETALA, F. BOUDJEMA, M. DJEBIRI, M.S. BOUCHERIT

Laboratoire de Commande des Processus, Ecole Nationale Polytechnique, Alger, Algeria

N.K. M'SIRDI

Laboratoire de Robotique de Paris, Université de Versailles, Saint Quentin, France

*Manuscript received 2001.02.07, revised version 2001.05.09*

This paper deals with the decentralized supervised control of robot manipulators using neural networks. First, a control law is synthesized by using the computed torque method. Since this latter is a model-based control technique, it presents some drawbacks such as sensitivity to parameter variations. To avoid this problem, the control law is supervised using neural networks. Simulation results with robustness tests are provided to demonstrate whether the neural controller is capable to mimic the existing controller in one hand, and on the other hand, to compensate disturbances for which the existing controller did not show satisfactory results in terms of performances. A comparative study of both the decentralized structure and the centralized structure of the supervised control is also given.

### 1. INTRODUCTION

The control of robot manipulators is an expanding field of study since the increasing use of robotics and automation technology in manufacturing industry. In real world tasks, mechanical manipulators are required to perform high-speed and high-precision trajectory tracking while subjected to unexpected disturbances from the external environment. One problem specifically encountered with robot manipulators is their highly nonlinear coupled dynamics, which makes traditional linear control techniques practically inadequate. Therefore, several approaches have been proposed to overcome these problems, namely, adaptive and variable structure approaches which allows the robots to perform desired tasks under the constraint of parameter variation [1]–[5]. The computed torque method is also one of these approaches [6], [7]. It is a particular case of the feedback linearisation in which its concept is to algebraically transform a nonlinear system dynamics into a linear one, so that linear control techniques can be applied. However, the weakness of this method is that it is based on an *a priori* knowledge of the manipulator dynamics and parameters, i.e., if unmodelled dynamics occur or the robot parameters change, performances will be seriously deteriorated.

Neural networks have become a challenging alternative to solve such problems, handling their attractive features such as nonlinearity, parallel processing, learning and generalization capabilities. The ability of neural networks to approximate arbitrary nonlinear mappings makes them of great deal in control of nonlinear systems [8]–[12].

Ozaki et al. [13] proposed a nonlinear compensator using neural networks. Neural networks were used to compensate nonlinearities of the robot manipulators. They also proposed a learning scheme using a model of known dynamics of manipulators. Simulation results showed good tracking performances even with untrained trajectories. Recently, in [14] a robust neural network feedback scheme is developed for motion of robot manipulators in which the joint velocities are estimated by a neural network observer. In [15], a neural network scheme was proposed to the motion control systems with deadzone and good simulation results on a two-link robot arm were obtained.

Due to the complexity of the dynamic models and the high number of variables involved in the control scheme of the interconnected large scale systems, the design and implementation of the centralized controller are generally not evident. To overcome such problems, a decentralized structure has been proposed [16], [17]. In this structure, the control system is decomposed into a number of interconnected subsystems, for each one, a local controller is designed independently using only local available signals. The obtained decentralized controllers are also reliable in the sense that when some local controllers are out of order, the rest of the system can still be in operation. In the robotic industry, the decentralized control has attracted some attention and it has been widely accepted due to its ease of implementation and tolerance to failure [1], [2], [5]. In this case, the manipulator is considered as an interconnected system in which each joint is taken as a subsystem.

In this paper, we intend to use supervised learning [10] in order to train neural networks to approximate the control law of the computed torque method applied to a three-joint robot manipulator. This allows us to make use of generalization capabilities of neural networks to maintain good tracking performances even with parameter uncertainty and reference trajectories changes. Two approaches are proposed here, a decentralized approach where each joint is independently controlled with a local neural controller which has access only to local information, and a centralized approach where one neural controller has the charge to control all the joints of the manipulator. Simulation results with robustness tests are given. A comparison between the neural controller and the existing controller is given and the effectiveness of the decentralized neural controller is clarified. The advantages of the decentralized scheme relative to the centralized one is also given in terms of neural networks structure complexity and tracking performances.

## 2. COMPUTED TORQUE METHOD

A multi-joint robot manipulator is generally described by the dynamic equation

$$\tau = M(q)\ddot{q} + h(q, \dot{q}) + F, \quad (1)$$

where:  $q(t)$  is the  $n \times 1$  vector of joint positions,  $\dot{q}(t)$  is the  $n \times 1$  vector of joint velocities,  $\tau(t)$  is the  $n \times 1$  vector of applied joint torques,  $M(q)$  is the  $n \times n$  manipulator inertia matrix,  $h(q, \dot{q})$  is the  $n \times 1$  centrifugal, Coriolis and frictional torque vector,  $F$  is the  $n \times 1$  vector of unknown terms arising from unmodeled dynamics and external disturbances.

The idea of the computed torque method, also known as feedback linearization, is to cancel the nonlinearities in a nonlinear system so that the closed loop dynamics is in a linear form and then to impose desired linear dynamics.

For a robot manipulator, the control law of the computed torque method is given by the following equation:

$$\begin{cases} \tau = \hat{M}(q)u + \hat{h}(q, \dot{q}) \\ u = \ddot{q}_r + k_p(q_r - q) + k_v(\dot{q}_r - \dot{q}) = \ddot{q}_r + k_p e + k_v \dot{e}, \end{cases} \quad (2)$$

where:  $\hat{M}$  and  $\hat{h}$  are the estimates of  $M$  and  $h$  respectively used for the nonlinear compensation,  $q_r, \dot{q}_r, \ddot{q}_r$  are the  $n \times 1$  vectors of desired joint positions, velocities and acceleration respectively,  $k_p$  and  $k_v$  are  $n \times n$  constant diagonal matrices with positive elements in their diagonals.

The dynamic of error is obtained by using equations (1) and (2)

$$\hat{M}(\ddot{q}_r + k_p e + k_v \dot{e}) + \hat{h} = M \ddot{q} + h + F, \quad (3)$$

then

$$\ddot{e} + k_v \dot{e} + k_p e = \hat{M}^{-1} \left( (M - \hat{M}) \ddot{q} + h + \hat{h} + F \right). \quad (4)$$

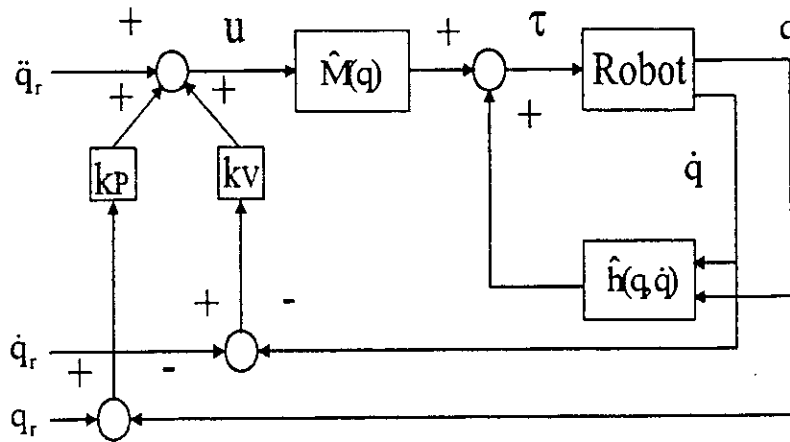


Fig. 1. Computed torque controller



It is easily seen that the tracking error  $e$  will not converge asymptotically to zero if  $\hat{M}$  and  $\hat{h}$  are not true estimates of  $M$  and  $h$  or if the vector of unknown terms  $F$  is not equal to zero. This can be the case in real situations where physical parameters of manipulator change or unmodeled dynamics occur due to nonlinear friction or high frequency modes. As a result, asymptotic stability is not assured and tracking performances will be severely affected. The computed torque control structure is illustrated in Fig. 1.

### 3. NEURAL NETWORK CONTROLLER

In an attempt to overcome restrictions of the computed torque method mentioned above, we intend in this section to implement this control technique in the form of a neural network. This is generally known as supervised control. Supervised control could be implemented in two stages:

- training stage: in this stage, the neural network learns the function input/output of the existing controller. When the latter controls the system, training examples are generated. Then a supervised learning algorithm is used in order to train the neural network,
- control stage: after training, the network controls the system and the existing controller is removed.

The general scheme of the supervised control is given in Fig. 2.

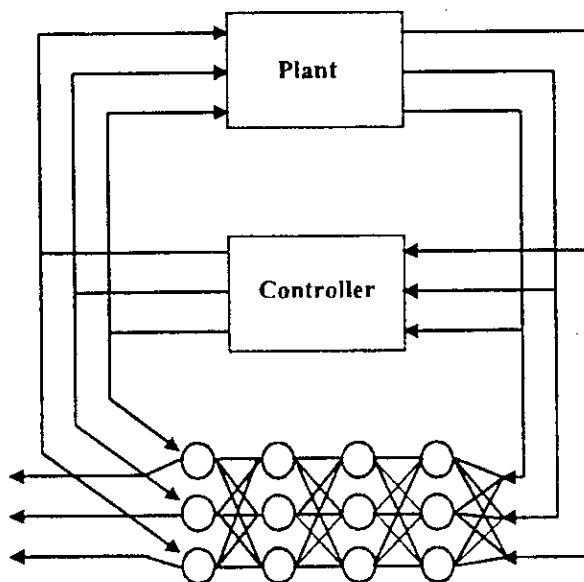


Fig. 2. The general scheme of the supervised control

### 3.1. Decentralized supervision of the computed torque law

This approach requires the use of a neural network for each joint  $i$ . The local neural controller will be trained to reproduce the local torque  $\tau_i$  while supplied with local information. The network dedicated to the joint  $i$  has one output and one hidden layer. The inputs to the  $i$ th network will depend on local variables appearing in the computed torque law giving  $\tau_i$ . So let us develop the expression of the computed torque law applied to a three-joint robot manipulator.

From (2) we have

$$\tau = \hat{M}(q) \left[ \ddot{q}_r + k_p e + k_v \dot{e} \right] + \hat{h}(q, \dot{q}). \quad (5)$$

For a robot manipulator [18]:

$$\hat{M}(q) = \begin{bmatrix} m_1 + m_2 + m_0 & 0 & 0 \\ 0 & J_2 + J_3 + m_3(q_3 - l_2)^2 + m_0 q_3^2 & 0 \\ 0 & 0 & m_3 + m_0 \end{bmatrix}, \quad (6)$$

$$\hat{h}(q, \dot{q}) = \begin{bmatrix} (m_1 + m_3 + m_0)g + f_1 \dot{q}_1 \\ 2(m_3(q_3 - l_2) + m_0 q_3) \dot{q}_2 \dot{q}_3 + f_2 \dot{q}_2 \\ -(m_3(q_3 - l_2) + m_0 q_3) \dot{q}_2^2 + f_3 \dot{q}_3 \end{bmatrix}, \quad (7)$$

where:  $m_0$  is the load supported by the end effector (payload),  $m_1$  is the mass of the first link,  $m_3$  is the mass of the third link,  $l_2$  is the distance between the gravity center of the third link and its extremity,  $J_2$  is the inertia of third link with regard to a vertical axis passing by its gravity center,  $J_3$  is the inertia of first link and its actuator,  $f_1, f_2$  and  $f_3$  are the viscous friction coefficients of the joint 1, 2 and 3 respectively,  $g$  is the constant of gravitation.

By substituting (6) and (7) in (5) we obtain

$$\tau_1 = (m_1 + m_3 + m_0) \left[ \ddot{q}_{r1} + k_p e_1 + k_v \dot{e}_1 \right] + (m_1 + m_3 + m_0)g + f, \quad (8a)$$

$$\tau_2 = \left( J_2 + J_3 + m_3(q_3 - l_2)^2 \right) \left[ \ddot{q}_{r2} + k_p e_2 + k_v \dot{e}_2 \right] + 2(m_3(q_3 - l_2) + m_0 q_3) \dot{q}_2 \dot{q}_3 + f_2 \dot{q}_2, \quad (8b)$$

$$\tau_3 = (m_3 + m_0) \left[ \ddot{q}_{r3} + k_p e_3 + k_v \dot{e}_3 \right] - (m_3(q_3 - l_2) + m_0 q_3) \dot{q}_2^2 + f_3 \dot{q}_3. \quad (8c)$$

From (8) we deduce that, in order to implement a decentralized control law and to provide the network with the necessary information to perform his task, we have to choose:  $e_1, \dot{e}_1, \dot{q}_1$  and  $\ddot{q}_{r1}$  as inputs for the neural network controlling the first joint (Fig. 3a),  $e_2, \dot{e}_2, \dot{q}_2$  and  $\ddot{q}_{r2}$  as inputs for the neural network controlling the second joint (Fig. 3a),  $e_3, \dot{e}_3, \dot{q}_3$  and  $\ddot{q}_{r3}$  as inputs for the neural network controlling the third joint 3 (Fig. 3b).

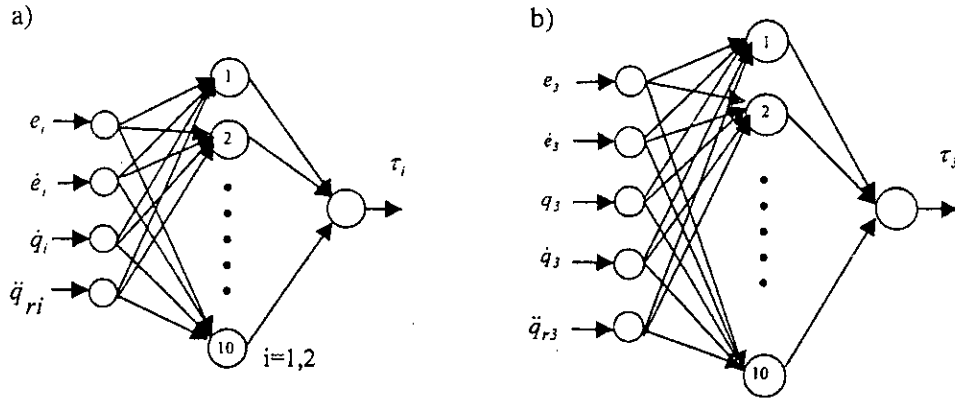


Fig. 3. Neural networks structure: a) for joint 1 and 2, b) for joint 3

Figure 4 shows the decentralized control scheme, for  $n$ -joint manipulator, when the training of the network has been performed. A constant gain  $K_i$  is introduced in each local feedback loop in order to adjust, if needed, the outputs of the neural controllers.

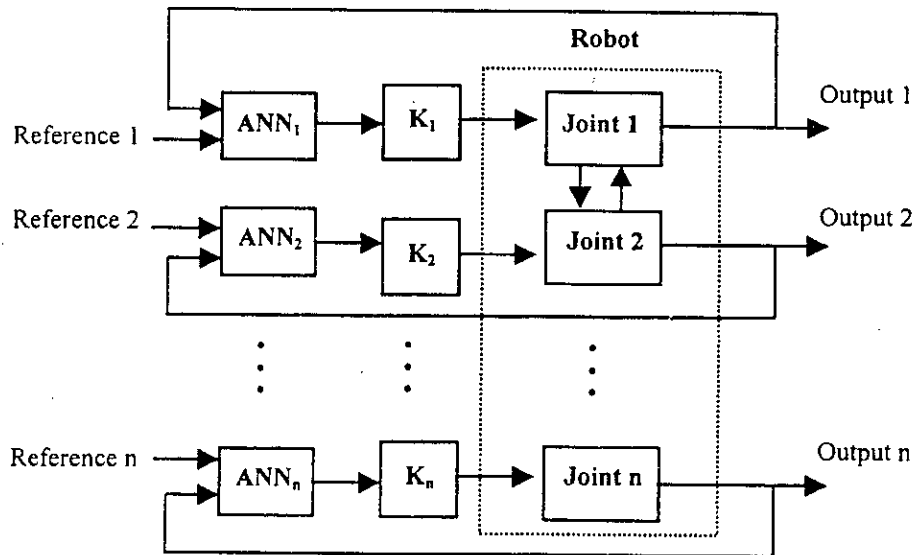


Fig. 4. Decentralized supervised control

### 3.2. Centralized supervision of the computed torque law

In this case, only one neural network will be used to control all joints of the manipulator. Thus, the number of neurons in the output layer of this network will be equal to the number of joints to be controlled. In order to supervise the computed torque law, the proposed neural network will require information from all the manipulator joints, and its inputs are all the variables appearing in the computed torque law (see Fig. 5).

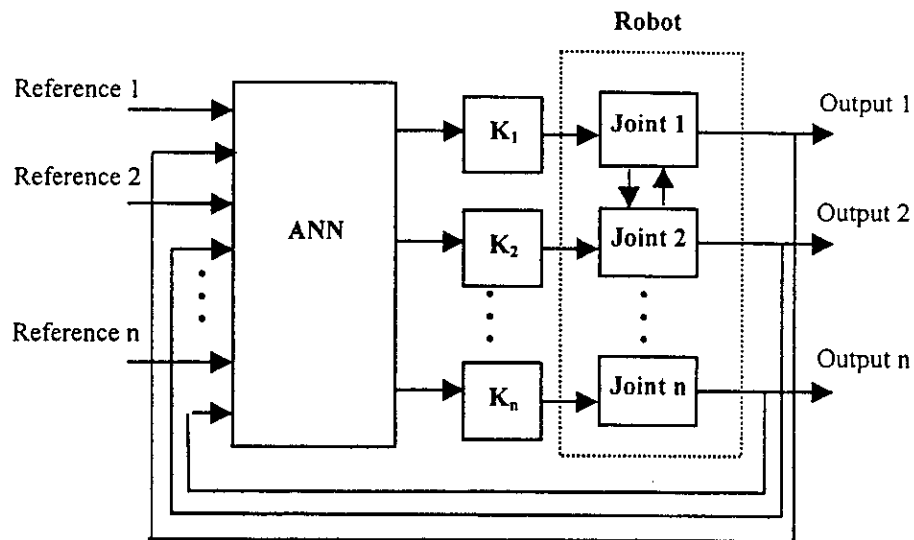


Fig. 5. Centralized supervised control

Therefore, for the control of the three joint robot manipulator, the network used has 13 inputs which are:  $e_1, \dot{e}_1, \dot{q}_1, \ddot{q}_{r1}, e_2, \dot{e}_2, \dot{q}_2, \ddot{q}_{r2}, e_3, \dot{e}_3, \dot{q}_3$  and  $\ddot{q}_{r3}$ , and 3 outputs which are the three joint torques  $\tau_1, \tau_2$  and  $\tau_3$ . Concerning the number of hidden layers and the number of neurons in each hidden layer, they will be determined in the next section depending on the network efficiency to control the manipulator.

Figure 5 shows the centralized control scheme once the training of the network has been accomplished. As in the decentralized case, the constant gains  $K_i$  are eventually incorporated in the feedback loop to adjust the outputs of the neural controller.

## 4. SIMULATION RESULTS

In this section, the two approaches proposed earlier are applied to the three joint robot manipulator (Fig. 6) of the following parameters:  $l_2 = 0.75$  m;  $m_1 = 20$  kg;  $m_3 = 10$  kg;  $k_1 = 100$  N·V<sup>-1</sup>;  $k_2 = 10$  N·V<sup>-1</sup>;  $k_3 = 10$  N·V<sup>-1</sup>;  $f_1 = 30$  N·s·m<sup>-1</sup>;  $f_2 = 7.825$  N·s·m<sup>-1</sup>;  $f_3 = 20$  N·s·m<sup>-1</sup>;  $J_2 = 2$  kg·m<sup>2</sup>;  $J_3 = 0.2$  kg·m<sup>2</sup>;  $g = 9.81$  m·s<sup>-2</sup>.

Firstly, the computed torque method is applied and training examples are generated. The constant diagonal matrices  $k_p$  and  $k_v$  used for this purpose are chosen as:  $k_{pi} = 20$ ,  $k_{vi} = 10$  ( $i = 1, 2, 3$ ). We must keep in mind that the training was performed supposing no payload ( $m_0 = 0$ ).

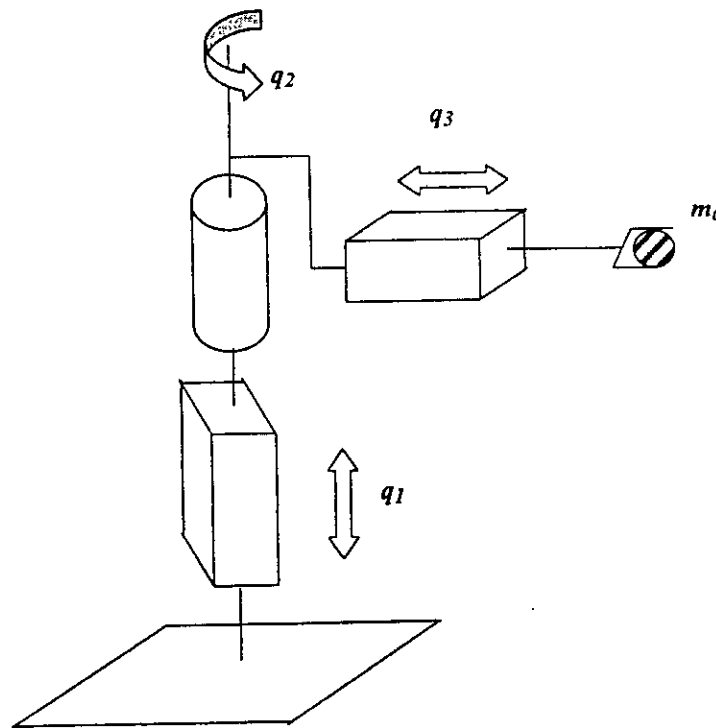


Fig. 6. The tree joint manipulator robot

#### 4.1. Decentralized approach

The structure of the three neural networks used in this approach has been already specified (see Fig. 3). Training has been performed using the backpropagation algorithm [8] over 150 examples. The learning rate was 0.2 for the first layer and 0.1 for the second layer. Weights were arbitrary initialized between 0.5 and  $-0.5$ . The constant gain  $K$ , mentioned above (see Fig. 4) were taken as 4, 1 and 1 for the first, second and third joint, respectively.

The number of iterations needed for successful training of the networks were 2400, 6000 and 600 for the first, second and third network, respectively.

Figure 7 illustrates a comparison between the decentralized neural controller and the computed torque controller when the manipulator picks up a 10 kg load with its end-effector and drops it at  $t = 1.5$  s, and by using the following well known standard cycloidal trajectories (see Fig. 9)

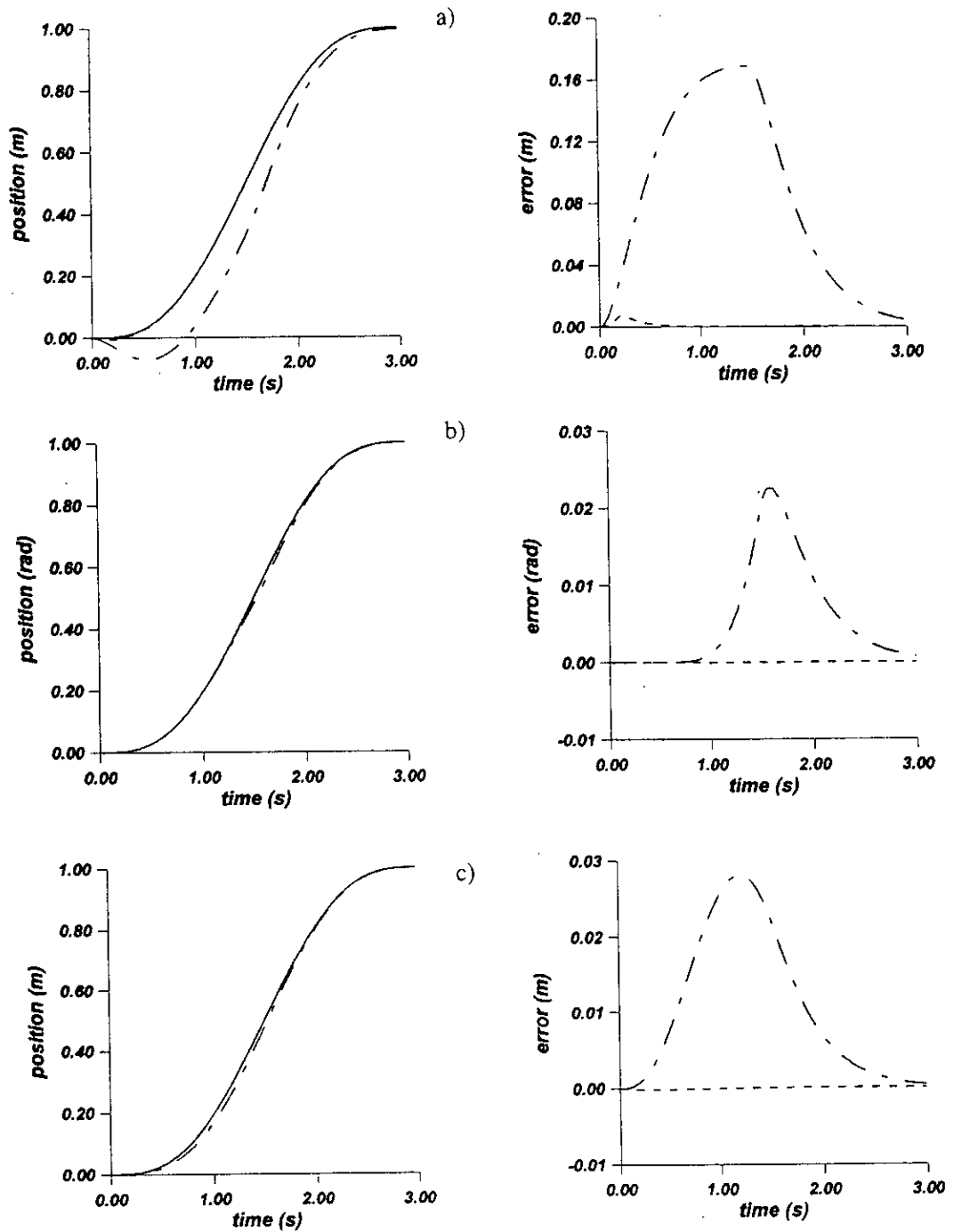


Fig. 7. Comparison between the decentralized supervised control and the computed torque control for cycloidal reference trajectories: a) joint 1, b) joint 2, c) joint 3; — reference, --- NN, - · - · - computed torque

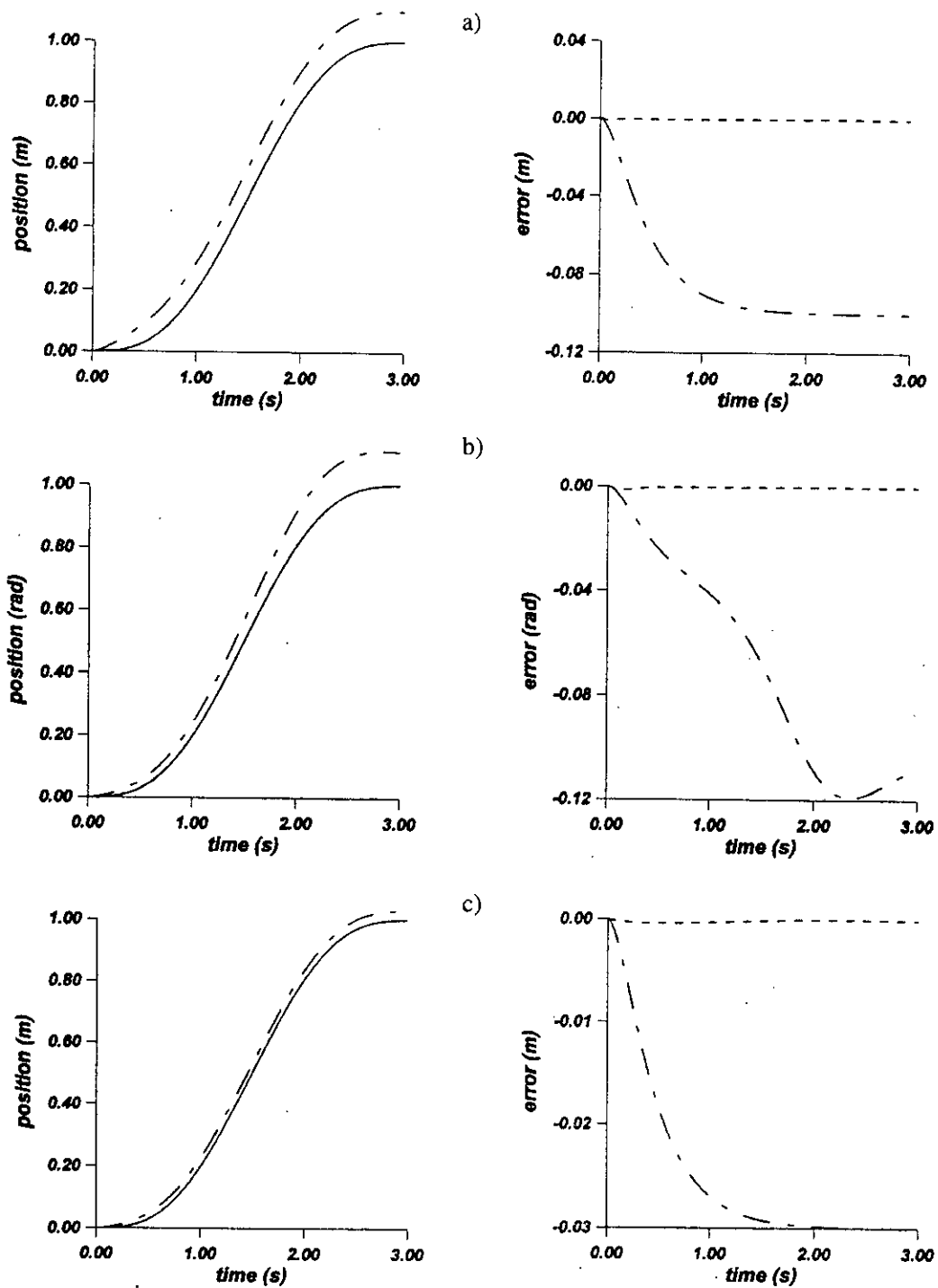


Fig. 8. Comparison between the decentralized supervised control and the computed torque control when  $F^T = [-60 \ -6 \ -6]$ : a) joint 1, b) joint 2, c) joint 3; notation as on Fig. 7

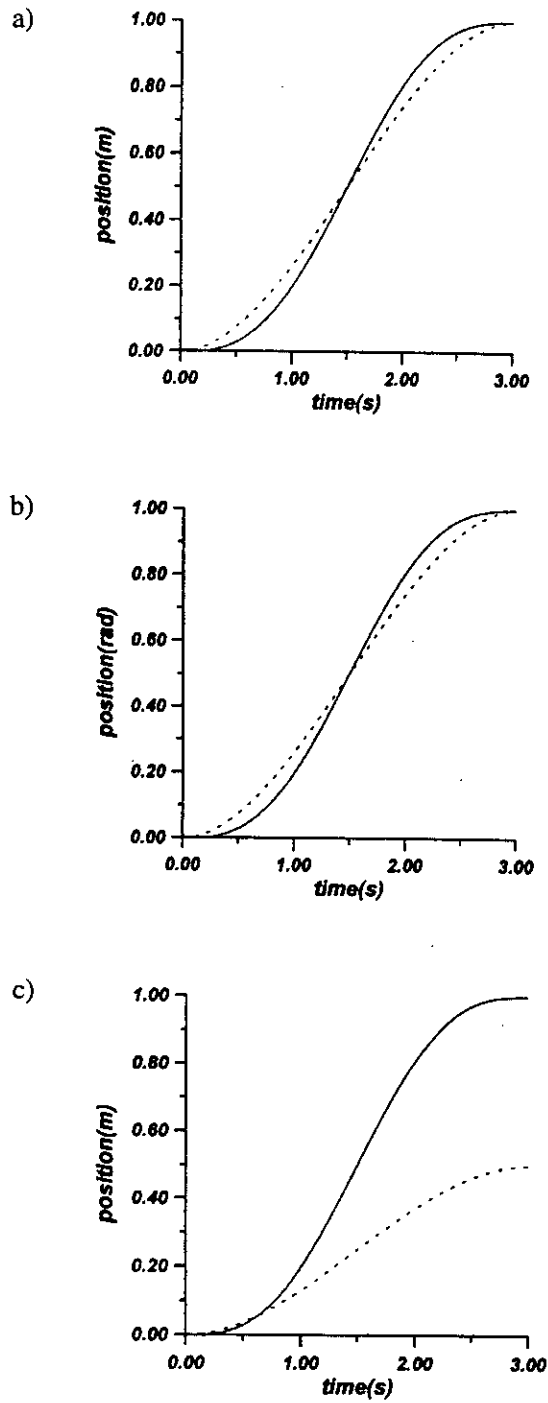


Fig. 9. Cycloidal and polynomial trajectories: a) joint 1, b) joint 2, c) joint 3;  
—— cycloidal, ---- polynomial



$$\begin{cases} q_{di}(t) = q_{di}(0) + \frac{q_{di}(t_f) - q_{di}(0)}{2\pi} \left( 2\pi \frac{t}{t_f} - \sin\left(2\pi \frac{t}{t_f}\right) \right) & \text{for } 0 \leq t \leq t_f \\ q_{di}(t_f) & \text{for } t < t_f \end{cases} \quad i = 1, 2, 3 \quad (9)$$

where  $t_f$  is the final time.

The neural controller demonstrates good tracking performances and better robustness because of the generalization capability of neural networks. This is not the case with computed torque method in which the performances are severely deteriorated when  $\hat{M}$  and  $\hat{h}$  are not true estimates of  $M$  and  $h$ .

Other robustness tests are provided in order to show the superiority of the neural controller compared to the computed torque controller. The term  $F$  (see eq. 1) previously supposed equal to zero, is now assigned a constant value, representing either unmodelled dynamics or external disturbances. Figure 8 shows tracking performances comparison between the decentralized neural approach and the computed torque approach when  $F^T = [-60 \ -6 \ -6]$  with the use of cycloidal reference trajectories. We can clearly see that the decentralized neural controller could compensate the term  $F$ , whereas the computed torque controller could not maintain good tracking performances since the model used in computing this control differs from the real model of the manipulator which now incorporates the term  $F$ .

It is important to note that the training set is obtained from the robot model without payload and disturbances. Whereas, performance tests are performed on the disturbed robot model in terms of changing payload, unmodelled dynamics and disturbances. This leads to a new set of examples which is different from the learning one.

In order to check the tracking capability of the decentralized neural network controllers when another reference trajectory is applied, the following standard polynomial trajectories, which are different from those used in the training phase (eq. 9) are chosen (see Fig. 9).

$$q_{di}(t) = -2\left(\frac{t}{t_f}\right)^3 + 3\left(\frac{t}{t_f}\right)^2 \quad i = 1, 2, 3. \quad (10)$$

Figure 10 shows the robustness of the decentralized neural controllers with regard to reference trajectories changes as well as in their forms (Fig. 10a, Fig. 10b) or in forms and amplitudes (Fig. 10c).

Note that due to the physical interconnections between joints of the manipulator, it is obvious that joints movements are correlated through these interconnections.

#### 4.2. Centralized approach

In this part and as specified previously, a neural network, which has 13 inputs and 3 outputs, is used. We have first chosen a network with one hidden layer whose number of neurons has been varied (10, 15 and then 20 neurons), but after many training cycles, the

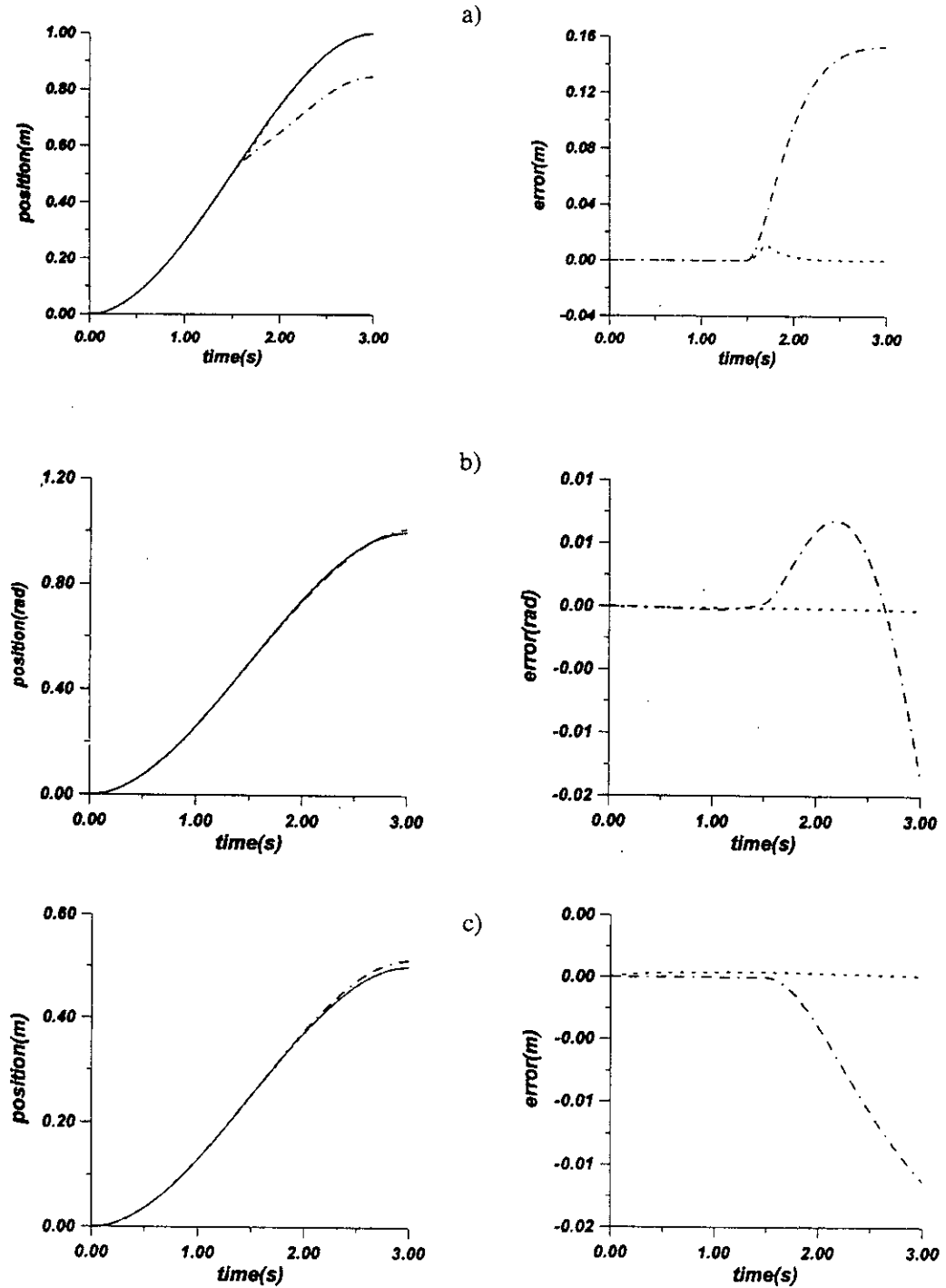


Fig. 10. Comparison between the decentralized supervised control and the computed torque control for polynomial reference trajectories: a) joint 1, b) joint 2, c) joint 3; notation as on Fig. 7

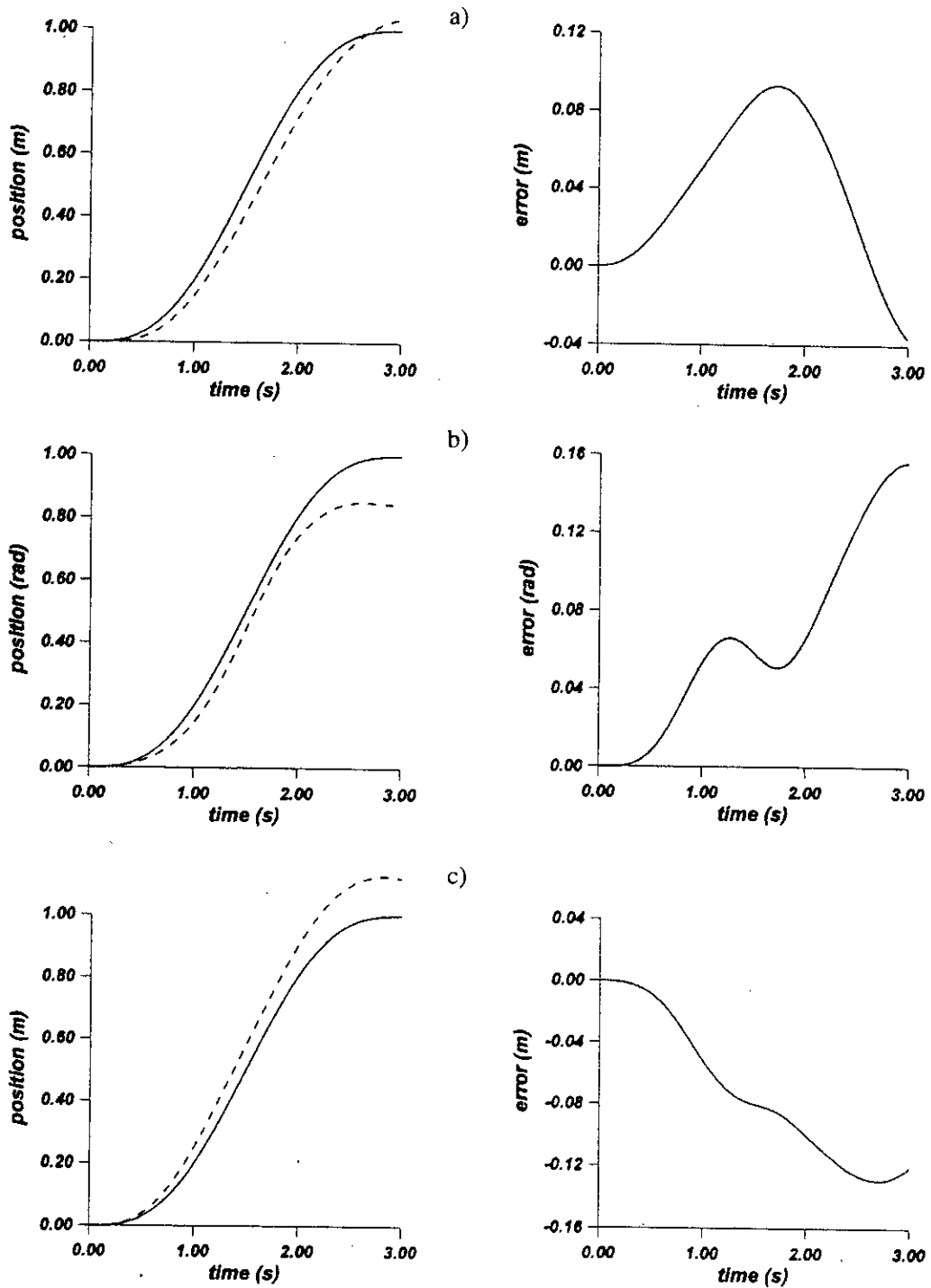


Fig. 11. Tracking performances of the centralized supervised control: a) joint 1, b) joint 2, c) joint 3; — reference, - - - output

network did not show satisfactory results in the control scheme. We have finally decided to increase the number of hidden layers to 2 instead of 1, with 10 neurons in each hidden layer. Training has been performed with the backpropagation algorithm [8] over 150 examples. Learning rates were 0.1, 0.08 and 0.06 for the first, second and third layers, respectively. Weights were arbitrary initialized between  $-0.5$  and  $0.5$ . The constant gains  $K_i$  were all set at 1.

Figure 11 shows the tracking performances after the network has been trained for 1500 iterations.

Since one of the aims of this paper is to compare centralized and decentralized neural network controllers in terms of design and training difficulties, tests are performed by using the same set of training data and the same training algorithm (i.e. the same conditions). Of course, it is clear that for the centralized case, if the number of the training data is increased and with another training algorithm such as efficient backpropagation, random optimization method or other methods, the performances may be improved.

We can deduce from the obtained results that in the conditions stated above, the centralized network was not even able to implement the computed torque law, not to mention robustness tests. In addition to these bad performances, it was not easy to choose a structure for the network and its training was time consuming and required relatively large memory in computer because of the large dimension of the network. This is the contrary of the decentralized approach, where better tracking performances were obtained and on the other hand, the three decentralized networks were of reduced dimension, so their training and implementation were easy to perform.

## 5. CONCLUSIONS

A decentralized supervised controller using neural networks has been presented. Such controller was able to implement the existing controller and even showed better robustness behavior. The neural controller, thus, overcame restrictions of the classical computed torque method, namely its close dependence on the manipulator model to compute the desired torque. However, the centralized neural controller was not able to implement the computed torque law, although the network dimension has been increased and training has been performed during many adaptation cycles what required more training data and more improved algorithms. Simulation results have shown the advantages of the neural approach with regard to the classical computed torque approach and, as well, the advantages of the decentralized approach with regard to the centralized approach either in design phase and in tracking performances or in implementation requirements.

## REFERENCES

1. Gavel T., Hsia T.C.: *Decentralized adaptive control of robot manipulators*. Proc. IEEE Int. Conf. on Robotics and Automation, Raleigh, NC, 1987, pp. 1230-1235.
2. Seraji H.: *Decentralized adaptive control of manipulator: theory, simulation and experimentation*. IEEE Trans. Robotics and Automation, RA-5, 1989, pp. 183-201.

3. Tang Y.: *Terminal sliding mode control for rigid robots*. Automatica, 34, 1998, pp. 51–56.
4. Lyou J.: *Note on decentralized adaptive controller design*. IEEE Trans. Autom. Control, AC-40, 1995, pp. 89–91.
5. Tang Y., Tomizuka M., Guerrero G., Montemayer G.: *Decentralized robust control of mechanical systems*. IEEE Trans. Autom. Control, AC-45, 2000, pp. 771–776.
6. Slotine J., Li W.: *Adaptive manipulator control: a case study*. IEEE Trans. Autom. Control, AC-33, 1988, pp. 995–1002.
7. Slotine J., Li W.: *Applied nonlinear control*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1991.
8. Miller W., Sutton R., Werbos P.: *Neural networks for control*. MIT Press, Cambridge, MA, 1992.
9. Guez, Selinsky T.: *A trainable neuromorphic controller*. J. Robotic Syst., Vol. 5, Apr. 1998, pp. 363–388.
10. Hamzi, Labiod S., Boukhetala D., Boudjema F.: *Supervised control using dynamical neural networks with application to an inverted pendulum*. IFAC-IFIP-IMACS Conference, Control of Industrial Systems, Vol. 2, May 1997, pp. 570–575.
11. Hunt K., Sbarbaro D., Zbikowsky R., Gawthrop P.: *Neural networks for control systems-A survey*. Automatica, Vol. 28, No. 6, 1992, pp. 1083–1112.
12. Fukuda T., Shibata T.: *Theory and applications of neural networks for industrial control systems*. IEEE Trans. Ind. Electron., Vol. 9, No. 6, Dec. 1992, pp. 472–489.
13. Ozaki T. et al.: *Trajectory control of robotic manipulators using neural networks*. IEEE Trans. Ind. Electron., Vol. 38, No. 3, Jun. 1991, pp. 195–202.
14. Young, Kim H., Lewis F.: *Neural network output feedback control of robot manipulators*. IEEE Trans. Robotics and Automation, RA-15, 1999, pp. 301–309.
15. Rastko R., Selmic, Lewis F.: *Deadzone compensation in motion control systems using neural networks*. IEEE Trans. Autom. Control, TAC-45, 2000, pp. 602–613.
16. Davison E., Tripathi N.: *The optimal decentralized control of large-power systems: load and frequency control*. IEEE Trans. Autom. Control, AC-23, 1978, pp. 312–325.
17. Jain S., Khorrami F., Fardanesh B.: *Decentralized control of large-scale power systems with unknown interconnections*. Int. J. Control, 63, 1996, pp. 591–608.
18. Stoten P.: *Generalized manipulator dynamics with regard to model reference adaptive control*. Int. J. Control, Vol. 50, No. 6, 1989, pp. 2249–2268.

#### ZDECENTRALIZOWANY STEROWNIK NEURONOWY ŚLEDZENIA TORU MANIPULATORÓW ROBOTÓW

Artykuł dotyczy zdecentralizowanego nadzoru sterowania manipulatorów robotów przy zastosowaniu sieci neuronowych. Podano syntetyczny opis metody sterowania opartej o obliczenia momentu obrotowego. Posługuje się ona modelem wrażliwym na zmienność wprowadzanych parametrów. Aby ograniczyć ten wpływ sterowanie manipulatorów nadzorowane jest za pomocą sieci neuronowych. Podano wyniki symulacji oraz rezultaty prób odporności na zakłócenia dla zademonstrowania, czy sterownik neuronowy jest w stanie zastąpić zwykły sterownik, wykazując jednocześnie zdolność kompensacji zakłóceń, której zwykły sterownik nie posiada. Podano analizę porównawczą zdecentralizowanego oraz scentralizowanego podejścia do zagadnienia nadzoru sterowania.

#### ДЕЦЕНТРАЛИЗИРОВАННЫЙ НЕЙТРОННЫЙ КОНТРОЛЛЕР НАБЛЮДАНИЯ ПУТИ МАНИПУЛЯТОРОВ РОБОТОВ

Статья касается децентрализованного наблюдения уравнения манипуляторов роботов с применением нейронных сетей. Подано синтетическое описание метода управления основанном на расчёте вращающего момента. Применяемый метод впечатлительный на изменение введённых

параметров. Чтобы ограничить это влияние управление манипуляторов контролирует при помощи нейронных сетей. Подано результаты симуляции и тоже результаты испытаний пробы устойчивости на помехи чтобы продемонстрировать, что нейронный контроллер есть в состоянии заступить обыкновенный контроллер, доказывая одновременно способность компенсации помех, которую обыкновенный контроллер не имеет. Подано сравнительный анализ децентрализованного и централизованного подхода к вопросу контроля управления.

## **Conclusion générale**

## Conclusion générale

L'ensemble des travaux présentés dans cette thèse concerne la commande décentralisée des systèmes complexes interconnectés appliquée aux robots manipulateurs. La majorité des études réalisées dans ce cadre utilisent les techniques classiques linéaires ou adaptatives.

Notre objectif a été d'essayer de donner de nouvelles orientations à cette commande, et ce en introduisant les outils modernes de l'automatique, tels que les systèmes à structure variable, les systèmes flous et les réseaux de neurones artificiels.

Dans une première partie, nous avons donné un aperçu général sur les outils utilisés pour le développement de l'ensemble des méthodes que nous avons proposé.

En premier lieu, la théorie des systèmes à structure variable ainsi que son utilisation en commande est introduite. Cette approche de commande est caractérisée par la nature discontinue du signal de commande et de sa variation à haute fréquence, ce qui nécessite l'utilisation de nouvelles technologies de l'électronique de commutation pour sa mise en œuvre pratique. Notre choix de cette méthode pour la commande décentralisée des systèmes interconnectés est motivé par sa caractéristique intéressante de robustesse par rapport aux variations paramétriques et aux perturbations externes. En effet, dans les systèmes interconnectés, la présence des interconnexions entre les sous systèmes provoque parfois d'importantes perturbations au niveau des stations de commandes locales.

Nous avons ensuite présenté quelques concepts mathématiques de base sur la théorie de la logique floue en mettant en évidence ses caractéristiques et son application dans le domaine de l'automatique. Le caractère linguistique des variables manipulées lui permet d'être utilisée pour la modélisation des systèmes complexes dont les modèles sont difficiles à obtenir, ce qui a ouvert un vaste champ d'application pour cette technique dans les milieux industriels.

La notion de réseaux de neurones artificiels est introduite en dernier lieu. Leur structure, leur algorithme d'apprentissage ainsi que leur utilisation en commande ont été présentées. Ces systèmes intelligents, réputés par leur capacité d'approximation de fonctions non linéaires et leur capacité de généralisation, ont connu un succès considérable dans divers domaines d'application, en particulier dans la commande des systèmes non linéaires.

Dans la seconde partie, nous avons abordé le problème de la commande décentralisée à structure variable des robots. Une formulation décentralisée du modèle du robot est d'abord donnée. Ce dernier est considéré comme un système interconnecté où chaque articulation représente un sous système. Une nouvelle classe de surfaces de glissement non linéaires est



ensuite proposée pour chaque sous système. Ces surfaces sont facilement synthétisables par un simple choix approprié de certaines fonctions non linéaires. Il est à noter que la plupart des surfaces rencontrées dans la littérature appartiennent à cette classe.

Sur la base du modèle décentralisé du robot et de la surface de glissement proposée, une nouvelle loi de commande décentralisée à structure variable a été élaborée. Le développement de cette dernière s'articule sur certaines caractéristiques physiques du robot. Cependant, la synthèse de la commande ne nécessite pas la connaissance exacte du modèle du système, ce qui facilite énormément sa mise en œuvre.

Cette loi de commande a été testée en simulation pour la commande en poursuite de trajectoire d'un robot manipulateur de type PUMA560. Les différents paramètres intervenant dans la commande ont été calculés d'une manière optimale en utilisant les algorithmes génétiques. Les résultats obtenus ont montré la faisabilité de la commande et sa robustesse vis à vis des variations paramétriques. Un test simulant une panne au niveau d'un sous système a montré aussi l'aptitude des autres contrôleurs locaux à y remédier. Une comparaison avec un régulateur Proportionnel-Dérivé en terme d'erreur de poursuite et du signal de commande a montré clairement la supériorité des performances de l'approche proposée.

Par ailleurs, nous avons implanté expérimentalement le schéma proposé pour la commande en poursuite de trajectoire d'un bras de robot manipulateur à trois degrés de liberté disponible au Laboratoire de Robotique de Paris ( France). Une carte DSP de type DS1103 PPC, installée sur un Pentium II à 200MHz a été utilisée. Les résultats obtenus sont jugés très satisfaisants.

La troisième partie de ce travail a été consacrée à l'utilisation de la logique floue pour la commande décentralisée adaptative des robots. Dans le développement de la loi de commande, un modèle flou de Sugeno a été utilisé pour compenser certaines variables non linéaires. Notre choix est porté sur ce modèle à cause de la forme analytique de son expression de sortie. Cette caractéristique nous a permis de démontrer la stabilité du système bouclé par la théorie de Lyapunov.

Afin de valider le schéma de commande proposé, nous avons simulé cette commande sur le modèle du robot PUMA560. A la lumière des résultats obtenus, cette méthode offre de très bonnes performances en poursuite de trajectoires, et une grande robustesse par rapport aux variations paramétriques et à la décentralisation. Il est à noter que l'approche proposée peut être reformulée pour la commande d'une large classe de systèmes dynamique non linéaires.

Dans la quatrième partie, nous avons proposé une approche de commande décentralisée adaptative par réseaux de neurones artificiels. Cette méthode constitue une extension d'une méthode déjà introduite par F.R. Lewis [LEW93a] pour le cas centralisé. La stabilité globale a été démontrée en utilisant la théorie de Lyapunov. L'avantage de cette méthode par rapport à

celles déjà développées dans le domaine de la commande adaptative par réseaux de neurones artificiels est qu'elle ne nécessite pas d'apprentissage *off line* du réseau de neurone. L'adaptation des poids du réseau s'effectue *on line* selon des lois appropriées et leur initialisation ne pose aucun problème.

Cette technique a été appliquée en simulation sur le robot PUMA560. Les résultats obtenus montrent clairement l'efficacité de la commande et sa robustesse par rapport aux variations paramétriques.

Dans la cinquième et dernière partie du travail, nous avons utilisé les réseaux de neurones pour la supervision d'une commande dite à couple calculé. Cette commande repose principalement sur le modèle du robot, ce qui permet la diminution de sa robustesse lorsque les paramètres du robot changent. En effet la capacité de généralisation des réseaux de neurones artificiels nous a permis d'éliminer ce défaut. La simplicité de synthèse et d'apprentissage des réseaux de neurones dans la structure décentralisée est mise en évidence à travers des résultats de simulation sur un robot cylindrique.

Enfin, l'étude présentée dans cette thèse constitue, à notre connaissance, une des premières tentatives de l'introduction des techniques modernes de l'automatique dans la commande décentralisée des systèmes interconnectés, particulièrement les robots manipulateurs.

# Annexe

## Annexe

### Modèle du robot PUMA560.

Nous présentons le modèle dynamique du robot en supposant que les trois dernières articulations sont à l'arrêt (Figure II.3).

Afin de faciliter l'écriture, les abréviations suivantes sont adoptées pour les fonctions trigonométriques :  $C_i$ ,  $S_i$ ,  $C_{ij}$ ,  $S_{ij}$  et  $C_{ijk}$  désignent respectivement  $\cos(q_i)$ ,  $\sin(q_i)$ ,  $\cos(q_i + q_j)$ ,  $\sin(q_i + q_j)$  et  $\cos(q_i + q_j + q_k)$ .

Le modèle dynamique des trois premières articulations du robot est :

$$M(q) \ddot{q} + B(q, \dot{q}) \dot{q} + G(q) = u - m J^T(q) [ J(q) \ddot{q} + \dot{J}(q, \dot{q}) \dot{q} + g ]$$

où  $q^T = [q_1, q_2, q_3]$  est le vecteur des coordonnées généralisées,  $m$  la charge (masse),  $J(q)$  la matrice Jacobiéenne et  $g^T = [0 \ 0 \ +9.81]$ .

Les expressions des éléments des matrices  $M$ ,  $B$  et  $G$  sont: [ARM86]:

Matrice d'inertie  $M(q)$

$$m_{11} \approx 2.57 + 1.38 C_2^2 + 0.30 S_2 S_3 + 7.44 \times 10^{-1} C_2 S_{23} ,$$

$$m_{12} = m_{21} \approx 6.90 \times 10^{-1} S_2 - 1.34 \times 10^{-1} C_{23} + 2.38 \times 10^{-2} C_2 ,$$

$$m_{13} = m_{31} \approx -1.34 \times 10^{-1} C_{23} - 3.97 \times 10^{-3} S_{23} ,$$

$$m_{22} \approx 6.79 + 7.44 \times 10^{-1} S_3 ,$$

$$m_{23} = m_{32} \approx 0.333 + 3.72 \times 10^{-1} S_3 - 1.10 \times 10^{-2} C_3 ,$$

$$m_{33} \approx 1.16 .$$

Les éléments de Coriolis et centrifuge  $B(q, \dot{q})$  :

$$\begin{aligned}
b_{11} &\approx (-2.76 S_2 C_2 + 7.44 \times 10^{-1} C_{223} + 0.60 S_2 C_3 \\
&\quad - 2.13 \times 10^{-2} (1 - 2 S_2 S_3)) \dot{q}_2, \\
b_{12} &\approx (6.90 \times 10^{-1} C_2 + 1.34 \times 10^{-1} S_{23} - 2.38 \times 10^{-2} S_2) \dot{q}_2 \\
&\quad + (2.67 \times 10^{-1} S_{23} - 7.58 \times 10^{-3} C_{23}) \dot{q}_3, \\
b_{13} &\approx (7.44 \times 10^{-1} C_2 C_{23} + 0.60 S_2 C_3 + 2.20 \times 10^{-2} C_2 S_{23} \\
&\quad - 2.13 \times 10^{-2} (1 - 2 S_2 S_3)) \dot{q}_1 \\
&\quad + 0.5 (2.67 \times 10^{-1} S_{23} - 7.58 \times 10^{-3} C_{23}) \dot{q}_3, \\
b_{21} &\approx -0.5 (-2.76 S_2 C_2 + 7.44 \times 10^{-1} C_{223} + 0.60 S_2 C_3 \\
&\quad - 2.13 \times 10^{-2} (1 - 2 S_2 S_3)) \dot{q}_1, \\
b_{22} &\approx (2.20 \times 10^{-2} S_3 + 7.44 \times 10^{-1} C_3) \dot{q}_3, \\
b_{23} &\approx 0.5 (2.20 \times 10^{-2} S_3 + 7.44 \times 10^{-1} C_3) \dot{q}_3, \\
b_{31} &\approx -0.5 (7.44 \times 10^{-1} C_2 C_{23} + 0.60 S_2 C_3 \\
&\quad + 2.20 \times 10^{-2} C_2 S_{23} - 2.13 \times 10^{-2} (1 - 2 S_2 S_3)) \dot{q}_1, \\
b_{32} &\approx -0.5 (2.20 \times 10^{-2} S_3 + 7.44 \times 10^{-1} C_3) \dot{q}_2, \\
b_{33} &= 0.
\end{aligned}$$

Terme de gravitation  $G(q)$

$$\begin{aligned}
g_1 &= 0, \\
g_2 &\approx -37.2 C_2 - 8.4 S_{23} + 1.02 S_2, \\
g_3 &\approx -8.4 S_{23} + 0.25 C_{23}.
\end{aligned}$$

La matrice Jacobienne  $J(q)$

$$\begin{aligned}
j_{11} &= -S_1(a_2 C_2 + a_3 C_{23}) - (d_2 + d_3) C_1 - d_4 S_1 S_{23}, \\
j_{12} &= -C_1(a_2 S_2 + a_3 S_{23}) + d_4 C_1 C_{23}, \\
j_{13} &= -a_3 C_1 S_{23} + d_4 C_1 C_{23}, \\
j_{21} &= C_1(a_2 C_2 + a_3 C_{23}) - (d_2 + d_3) C_1 + d_4 C_1 S_{23}, \\
j_{22} &= -S_1(a_2 S_2 + a_3 S_{23}) + d_4 S_1 C_{23}, \\
j_{23} &= -a_3 S_1 S_{23} + d_4 S_1 C_{23}, \\
j_{31} &= 0, \\
j_{32} &= -(a_2 C_2 + a_3 C_{23}) - d_4 S_{23}, \\
j_{33} &= -a_3 C_{23} - d_4 S_{23}.
\end{aligned}$$

où

$$a_2 = 0.4319 \text{ m. } a_3 = -0.0203 \text{ m,}$$

$$b_2 = 0.2435 \text{ m. } b_3 = -0.0934 \text{ m et } b_4 = 0.4331 \text{ m.}$$

## **Références bibliographiques**

## Références bibliographiques

- [AGG93] M. E. Aggoune, M. E. F. Boudjema, A. Bensenousi, A. Hellal, M. R. Elmesai, & S. V. Vadari, Design of adaptive variable structure voltage regulator using artificial neural network, *Proc. 2ed IEEE Conf. in Control Applications*, Vancouver, Canada, 1993, 337-343.
- [AGG94] M. E. Aggoune, , F. Boudjema, A. Bensenousi, A. Hellal, M. R. Elmesai, & S. V. Vadari , Design of variable structure voltage regulator using pole assignment technique. *IEEE Trans. Autom. Control*, AC-39, 1994, 2106-2110.
- [ARM86] B. Armstrong, , O. Khatib & J. Burdick ,The explicit dynamic model and inertial parameters of the PUMA 560 arm, *Proc. 1986 IEEE Int. Conf. On Robotics and Automation*, San Francisco. CA, 1986, 510-518.
- [AST89] K. J. Aström.B. Wittenmark *Adaptive Control*, Adisson-Wesly, New York, 1989
- [AST90] K.J.Aström , *Computer controlled systems*. Edition Prentice Hall ,1990.
- [BAC97a] T.Bachir-Bouyadjara, M.S. Boucherit, **D.Boukhetala**, Commande Adaptative Décentralisée des Systèmes Fortement Interconnectés. *Proceeding of the IEEA*, Vol. 2, pp 46-50. Dec.1997.
- [BAC97b] R .Bachir-Bouyadjara, **D.Boukhetala**, M.S. Boucherit , Commande Décentralisée à Structure Variable d'un robot. *CMSES'97 Saida Algérie*. pp 15-21. Mai.1997.
- [BAC98] R .Bachir-Bouyadjara, *Commande adaptative décentralisée des robots manipulateurs* , Thèse de Magister, Ecole Nationale Polytechnique , (ENP 98).
- [BEL97] A. Belmehdi et al, Commande Linéarisante par Feedback Utilisant les Réseaux de Neurones Multicouches. Application à un Bras de Robot à Trois Degrés de Liberté. *Proceeding of the IEEA*, Vol. 2, pp 88-92. Dec.1997.
- [BEN88] A. Benhammou, *Contribution à l'Etude de la Commande Adaptative Décentralisée des Systèmes Interconnectés*. Thèse de Doctorat d'état, université Paul Sabatier de Toulouse, 1988.
- [BEN91] A . Benallegue , *Contribution à la commande dynamique adaptative des robots manipulateurs rapides*. Thèse de doctorat de l'université PARIS VI, novembre 1991.
- [BER93] H. Berghuis and H. Nijmeijer, Global regulation of robots using only position measurements. *Systems and Control Letters*, Vol. 21, pp. 289-293. 1993.
- [BOI88] J. D. Boissonnat et al. *Techniques de la Robotique(Tome 1)*, Editions Hermès. Paris, 1988.
- [BOU00a] **D. Boukhetala** , F. Boudjema ,A. Melahi, A. Boulahia et N.K. M'Sirdi, Commande décentralisée par logique floue appliquée à un bras de robot manipulateur. *ICEL '00 International Conference on Electrotechnics*, Oran 2000, Nov.13-15,2000 pp.337-341.
- [BOU00b] A. Boulahia. *Commandes floue et neuro-floue appliquées à un bras de robot*, thèse de Magister, Thèse de Magister. Ecole Militaire Polytechnique (EMP ex. ENIT ), 2000
- [BOU01] **D. Boukhetala** , F. Boudjema, M. Djebiri , M.S. Boucherit and N.K. M`Sirdi, Decentralized neural network controller for trajectory tracking of robot manipulator. *Archives of Electrical Engineering*, Journal of Poland Academy vol.L N° 3.2001, pp.297-311.
- [BOU90] F. Boudjema, & J. L. Abatut , Sliding-mode - a new way to control series resonant converters, In *Proc. 1990 IEEE Conf. Ind. Electron. Society*, Pacific Grove, California. USA, 1990. 938-943.
- [BOU92] **D. Boukhetala** et A.Belmehdi, Commande Décentralisée des Systèmes Structurés. *Séminaire ,Signaux, systèmes et Automatismes SSA'92 Blida* 13-15 Dec.1992.
- [BOU91] F. Boudjema,*Commande par mode de glissement: Application aux convertisseurs*

- électriques*, Thèse de l'université de Paul Sabatier de Toulouse, France Mai 1991.
- [BOU93a] **D. Boukhetala** et A. Belmehdi, Caractérisation des Modes Fixes dans la Commande Décentralisée des Grands Systèmes. *Colloque Maghrébin sur l'Automatique et l'Electronique Industrielle COMAEI Djelfa*, 1993. pp 27-38.
- [BOU93b] **D. Boukhetala**, *Commande décentralisée des systèmes structurés*, thèse de Magister. (ENP 93).
- [BOU94] **D. Boukhetala** et F. Boudjema, Synthèse des Observateurs d'Etat dans les Systèmes Décentralisés. *Colloque sur l'Electrotechnique et l'Automatique CEA '94 Algiers*. Nov. 29-30 1994, Vol. 1. pp 154-159.
- [BOU96a] **D. Boukhetala**, O. Bouhali, et F. Boudjema, Commande Adaptative à Structure Décentralisée par Réseaux de Neurones Statiques d'un Bras de Robot Manipulateur. *Deuxième Conférence Maghrébine sur l'Automatique, l'Electrotechnique et l'Electronique Industrielle COMAEI'96 Tlemcen* 3-5 Decembre 1996, AJOT. Vol. 1, pp 165-170.
- [BOU96b] **D. Boukhetala**, M. Gareh and M.S. Boucherit, Decentralized Direct Adaptive Control of interconnected Systems. *Algerian Journal of Technology (AJOT)*, International Publications, Vol. 12 1996. ISSN 1111-357 X, pp 27-30.
- [BOU97a] O. Bouhali, **D. Boukhetala**, F. Boudjema et M. S. Boucherit, Commande Adaptative directe par Réseaux de Neurones Gaussiens Appliquée à un Pendule. *Proceeding of the IEEA*, Vol. 1, pp 76-80. Dec. 1997.
- [BOU97b] **D. Boukhetala**, O. Bouhali, F. Boudjema, N.K. M'Sirdi, Commande Adaptative Décentralisée des Robots Manipulateur en utilisant les Réseaux de Neurones Artificiels: Etude Comparative. *Algerian Journal of Technology (AJOT)*, International Publications, Série B, vol .13, No 1, 1997, ISSN 1111-357 X . pp 94-101.
- [BOU98a] O. Bouhali, **D. Boukhetala**, F. Boudjema, N.K. M'Sirdi, Une Approche de Commande Adaptative Décentralisée par Réseaux de Neurones Récurrents des Robots manipulateurs. *Troisième Conférence Maghrébine sur l'Automatique, l'Electrotechnique et l'Electronique Industrielle COMAEI'98 Bejaia* 8-9 Decembre 1998, AJOT, pp 184-187.
- [BOU98b] O. Bouhali, **D. Boukhetala**, F. Boudjema, N.K. M'Sirdi, Commande Adaptative Indirecte Décentralisée par RNA du Robot Manipulateur PUMA 560. *ICEL'98 International Conference on Electrotechnics*, 5-7 Oct 1998, Oran . pp 26-30.
- [BOU98c] O. Bouhali, *Etude et Développement de Méthodes de commande adaptative centralisée et décentralisée en utilisant différents types de réseaux de neurones artificiels : Application au robot PUMA560*, Thèse de Magister, Ecole Nationale Polytechnique, (ENP) 1998.
- [BOU98d] O. Bouhali, **D. Boukhetala**, F. Boudjema et N. K. M'Sirdi, "Commande Neuronale Adaptative Décentralisée du Robot PUMA560, CNP'98, 30-31 mai 1998 Tizi-Ouzou, Algérie, pp 200-204.
- [BOU99a] **D. Boukhetala**, F. Boudjema, M. Djebiri, M.S. Boucherit et N.K. M'SIRDI Decentralized supervised control using neural networks with application to robotic manipulators. *AJOT international publications*, Vo 14 n° 1 1999, pp 132-137.
- [BOU99b] **D. Boukhetala**, F. Boudjema, A. Melahi, A. Boulahia, Application de la logique floue pour la commande d'un robot. *Séminaire National de Mathématiques (CNM'99)*, Annaba, 17-18 Mai 1999.
- [BUH86] H. Bühler, *Réglage par mode de glissement*. Presse polytechnique et universitaire Romande, Lausanne (Suisse), 1986.
- [BUH94] H. Bühler, *Réglage par logique floue*. Presse polytechnique et universitaire Romande, Lausanne (Suisse). 1994.
- [CAN92] C. Canudas de Wit and N. Fixot, Adaptive control of robot manipulators via velocity estimated feedback. *IEEE Transactions on Automatic Control*. Vol. 37, no. 8. pp. 1234-1237, 1992.
- [CHA95a] F. C. Chang and H. K. Khalil, Adaptive Control of a Class of Nonlinear Discrete-Time Systems Using Neural Networks. *IEEE Trans. on Automatic Control*, Vol. 40, N° 5, MAY 1995.



- [CHA95b] N. Chatry and al, Double Backpropagation for the Gradient Calculation in Dynamic Feedback Networks, *IFAC-IFIP-IMACS Conference, Control of Industrial systems*, pp 535-538, 20-22 May 1997, Belfort France.
- [CHA96] W.-D. Chang, L.-C. Fu and J.-H. yang, Adaptive robust neural-network based control for SISO systems. *13<sup>th</sup> Triennial world Congress*. San-Frasisco, USA, pp. 163-168, 1996.
- [CHE90] F. C. Chen, Backpropagation Neural Networks fo Nonlinear Self-Tuning Adaptive Control. *IEEE Control Syst. Mag.*, vol. 10, pp. 44-48, 1990.
- [CHO88] Y.K. Choi and Z. Bien. Decentralized control scheme for control of a multi-arm-type robot, *International Journal of Control*, Vol. 48, no. 4, pp. 1715-1722, 1988.
- [CHU96] V. K. Chu. & M. Tomizuka, Sliding mode control with nonlinear sliding surfaces, *Proc. 13th IFAC Triennial World Congress*, San Francisco, USA, 1996, pp 481-486.
- [COL95] R. Colbaugh and K. Glass, Adaptive tracking control of manipulators using only position measurements, *Proceedings of the 34<sup>th</sup> Conference On Decision And Control*, New-Orleans, pp. 51-54. LA-Dec. 1995.
- [COM96] S. Commuri and F. L. Lewis, "Design and stability analysis of adaptive-fuzzy controllers for a class of nonlinear systems", *IEEE proceedings of the 35<sup>th</sup> Conference on Decision and Control*, Kobe (Japan), Dec. 1996.
- [CRA86] J. J. Craig, P. Hsu and S. Sastry, "Adaptive control of mechanical manipulators", International Conference on Robotics Automation, San-Francisco, CA, 1986.
- [DAV78] E. J. Davison, & N. Tripathi, The optimal decentralized control of large-power systems: load and frequency control, *IEEE Trans. Autom. Control*, AC-23, 1978, pp 312-325.
- [DRI93] D. Driankov, H. Hellendoor and M. Reinfrank, *An introduction to fuzzy control*. Springer-Verlag, Berlin Heidelberg, 1993
- [DUB79] S. Dubowsky and D. T.Desforges, The application of model referenced adaptive control to robotic manipulators. *ASME Journal of Dynamic systems, Measurements and Control*, Vol. 101, no. 3, pp. 193-200, 1979.
- [ERB99] K. Erbatur, M. O. Kaynak, & A. Sabonovic, A study on robustness property of sliding mode controllers: A novel design and experimental investigations, *IEEE Trans. Indus. Elect.*, IE-46, 1999, pp 1012-1018.
- [FEN95] G. Feng, & Y. A. Jiang, Variable structure based decentralized adaptive control, *IEE Proc. Control Theory Application*, 142, September 1995, pp 439-443.
- [FER98] J. J. Fernandez, and I. D. Walker, Biologically inspired robot grasping using genetic programming, *Proc. IEEE Int. Conf. On Robotics and Automation*. Leuven, Belgium, 1998. pp 3032-3039.
- [FRE92] J. A. Freeman, D. M. Skapura, *Neural Networks* Adisson-Wesley Publishing Company, Inc, 1992.
- [FU87] K. S. Fu. R. C. Gonzalez and C. S. G. Lee, *Robotics: control, sensing, vision and intelligence*. Edition Mc Graw-Hill. 1997.
- [FU92] L.C. Fu, Robot adaptive decentralized control of robot manipulators, *IEE Trans On Autom and Contr*, Vol. 37, pp. 106-110. Jan. 1992.
- [FUR96] T. Furuta, & K. Tomiyama, Sliding mode controller with time-varying hyperplane, *Proc. IROS 96*, 1996, pp 576-581.
- [GAV87] D. T. Gavel, & T. C. Hsia, Decentralized adaptive control of robot manipulators, *Proc. IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC, 1987, pp 1230-1235

- [GAV89] D. T. Gavel, & D. D. Siljak, Decentralized adaptive control: Structural conditions for stability, *IEEE Trans. Autom. Control*, AC-34, 1989, pp 413-426.
- [GEO92] G. George, *Sur des méthodes de commande non linéaires adaptatives : Aspects échantillonné et applications*. Doctorat ès science, Université de Paris XI Orsay, France, 1992.
- [GOL89] D.E. Goldberg, *Genetic Algorithms in search, optimization and machine learning*, Addison Wesley, Reading, MA, 1989.
- [GOO84] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*, Englewood Cliffs, NJ :Prentice Hall, 1984.
- [GUZ86] C. Guzzella and H. P. Geering, Model following variable structure control for a class of uncertain mechanical systems, *IEEE Conference on Decision and Control*, Athens, Greece, 1986.
- [HA87] I. J. Ha and E. G. Gilbert, Robust tracking in nonlinear systems, *IEEE Trans on Autom and Contr*, Vol. AC-32, pp. 763-771, Sep. 1987.
- [HAM96a] B. Hamzi, **D. Boukhetala** and F. Boudjema, Remotely Operated Vehicle Attitude and Position Control:Application of Geometric Control Theory, *Deuxième Conférence Maghrébine sur l'Automatique, l'Electrotechnique et l'Electronique Industrielle COMAEI'96* Tlemcen 3-5 Decembre 1996, AJOT. Pp 205-210.
- [HAM96b] B. Hamzi, M.S. Boucherit, **D. Boukhetala**, F. Boudjema, Commande Adaptative par Reseaux de Neurones d'un Robot Manipulateur, *Algerian Journal of Technology (AJOT)*, International Publications ,vol .12 ,1996. ISSN 1111-357 X. pp 40-45.
- [HAM97] B. Hamzi,S. Labiod, **D. Boukhetala** and F.Boudjema, Supervised Control Using Dynamical Neural Networks with Application to an Inverted Pendulum, *International IFAC-IFIP-IMACS Conference on Control of Industrial Systems*, 20-22 May 1997, Belfort , France.
- [HOR82] R. Horwitz and M. Tomizuka, Discrete time model reference adaptive control of mechanical manipulator, *Computer in Engineering Robot and Robotics*, ASME, 2, pp. 107-112, 1982.
- [HOR92] S.-I. Horikawa, T. Furuhashi and Y. Uchikawa, On fuzzy modelling using fuzzy neural networks with the back-propagation algorithm, *IEEE Transactions on Neural Networks*, Vol. 3, no. 5,pp. 801-806, Sept. 1992.
- [HUN92] K.J. Hunt and al, Neural Networks for Control Systems-A Survey, *Automatica*, vol. 28. NO. 6, pp. 1083-1112, 1992.
- [HUN93] J. Y. Hung, , W. Gao, & J. C. Hung, Variable structure control: a survey, *IEEE Trans. Ind. Electron.*, IE-40, 1993, 2-22.
- [IOA86] P. A. Ioannou, Decentralized adaptive control of interconnected systems, *IEEE Trans. Autom. Control*, AC-34, 1986, 291-298.
- [JAI96] S. Jain, F. Khorrami & B. Fardanesh, Decentralized control of large-scale power systems with unknown interconnections, *Int. J. Control*, 63, 1996, pp 591-608.
- [JAI97a] D.S. Jain *et al*, Decentralized Adaptive Control of a Class of Large-Scale Nonlinear Systems, *IEEE Trans. Auto. Contr.*, Vol. 42 pp 729-735. No. 05, May 1997.
- [JAI97b] D.S. Jain *et al*, Decentralized Adaptive Output Feedback Design for Large-Scale Interconnected Nonlinear Systems, *IEEE Trans. Auto. Contr.*, Vol. 42 pp 136-154. No. 02, 1997.

- [JAN92] J.-S. R. Jang. Self learning fuzzy controllers based on temporal back propagation, *IEEE Transactions on Neural Networks*. Vol. 3, no. 5, pp. 714-723, Sept. 1992.
- [JAN95] J.-S. R. Jang, Neuro fuzzy modelling and control, *Proceedings of the IEEE*, Vol. 83, no.3, pp. 378-404, Mar. 1995.
- [KET92] R. Ketata, *Méthodologies des régulations numériques incluant la logique floue*, Thèse de doctorat de l'institut des sciences appliquées de Toulouse, L.A.A.S, France, 1992.
- [KHA96] H. K. Khalil, Adaptive Output Feedback Control of Nonlinear Systems Represented by input-Output Models, *IEEE Trans. on Automatic Control*, Vol. 41, N° 2, pp 177-188, Feb. 1996.
- [KHE94] S. Khemaissia and A. S. Morris, Review of Artificial Neural Networks and Choice of Radial Basis Function Networks for System Identification, *Technologies Advances*, N° 6, September 1994.
- [KIM94] K.-C. Kim and J.-H. Kim, Multicriteria fuzzy control, *Journal of Intelligent and Fuzzy Systems*, Vol. 2, pp. 279-288, 1994.
- [KOI83] Koivo, A. J. and Guo, T. H. (1983) Adaptive linear Controller for Robotic Manipulators. *IEEE Transactions on Automatic control*, Vol. AG-28, No. 2.
- [KOS92] A. Kosco, *Neural network and fuzzy systems*, Edition Prentice Hall 1992.
- [KRA93] A. Krakasoglu et al, Identification and Decentralized Adaptive Control using dynamical Neural Networks with Application to Robotic Manipulators, *IEEE, Trans. Neural Networks*, Vol. 4, No.6, Nov. 1993.
- [KRS95] M. Krstic, I. Kanellakopoulos and P. Kokotovic, *Nonlinear and Adaptive Control Design*, John Wiley & Sons, Inc., 1995.
- [KUS93] J. G. Kuschewski, and al, Application of Feedforward Neural Networks to Dynamic System Identification and Control, *IEEE Transactions on Control Systems Technology*. Vol. 1, No. 1, March 1993.
- [KWA95a] C. M. Kwan and al, Robust Neural Network Control of Flexible-Joint Robots, *Proceeding of the 34<sup>th</sup> Conference in decision & Control*, pp. 1296-1301, December 1995.
- [KWA95b] C. M. Kwan and al, Robust Adaptive Control of Robots Using Neural Network : Global Tracking Stability, *Proceeding of the 34<sup>th</sup> Conference in decision & Control*, pp. 1846-1850, December 1995.
- [KWA97] C. Kwan and all., Robust robot control using fuzzy CMAC, *Proceedings of IFAC/IFIP/IMACS Conference*, Belfort, France, 1997.
- [LAB97] S. Labiod, H. Chekireb et M. S. Boucherit, Commande Floue Adaptative d'un Robot Manipulateur, *Proceeding des 17<sup>èmes</sup> journées Tunisiennes d'Electrotechnique et d'Automatique*, 5 et 6 Nov. 1997, Nabeul-Tunisie.
- [LAL94] Lallemand, *Robotique : aspects fondamentaux*, Edition Masson, 1994.
- [LAN79] I. D. Landau, *Commande Adaptative Aspects Pratiques et Théoriques*, Masson. 1979.
- [LAN79] Y. D. Landau, *Adaptive control : Model reference approach*, Marcel Dekker, New York, 1979.
- [LEA90] M. B. Leahy, Model based auxiliary input control : development and experimental analysis, *Proceedings Of the 20<sup>th</sup> Conference On Decision and Control*, Honolulu, Hawai. Dec. 1990.
- [LEU91] T. P. Leung, , Q. J. Zhou, & Yi Su Chun, An adaptive variable structure model following control design for robot manipulators, *IEEE Trans. Autom. Control*, AC-36, 1991, pp 347-353.
- [LEW93a] F. L. Lewis and al, Neural Net Robot Controller With Guaranteed Tracking Performances, *Proc. IEEE Int. Symp. Intellegent Control*, 1993.

- [LEW93b] F. L. Lewis, A. Yesildirek and k. Liu, Neural net robot controller with guaranteed stability, *IEEE Transaction on Robotics and Automation*, Vol. 5, no. 2, pp. 103-108, 1993.
- [LEW95] F. L. Lewis, C. M. Kwan, D. M. Dawson, robust adaptive control of robots using neural network : global tracking stability. *IEEE conference on decision & control, New Orleans*, 1995.
- [LOU97] M. Loudini, *Modélisation, analyse et méthodologies de commande linguistique floue d'un bras manipulateur de robot flexible*, Thèse de Magister, Ecole Nationale Polytechnique (ENP), Alger, 1997.
- [LYO95] J. Lyou, , Note on decentralized adaptive controller design, *IEEE Trans. Autom. Control*, AC-40, 1995, 89-91.
- [MAD00] T. Madani, *Différentes approches de Commande décentralisée à structure variable appliquées en robotique*, Thèse de Magister à l'Ecole Nationale Polytechnique (ENP), Alger, 2000.
- [MAD99a] T. Madani, **D. Boukhetala** et F. Boudjema, Réglage Décentralisé non Linéaire par Mode de Glissement des Robots Manipulateurs, *Séminaire Nationale sur l'Automatique et les Signaux SNAS'99*, Annaba, 9-10 Nov 1999. pp. 177-181.
- [MAD99b] T. Madani, **D. Boukhetala** et F. Boudjema, Commande Adaptative Décentralisée à Structure Variable avec Modèle de Référence des Robots Manipulateurs, *2<sup>ème</sup> Séminaire Nationale en Génie Electrique SNGE'99*, Biskra, 22-24 Nov 1999. pp. 67-72.
- [MAT88] G. P. Mathews, & R. A. DeCarlo, Decentralized tracking for a class of interconnected non linear systems using variable structure control, *Automatica*, 24, 1988, 187-193.
- [MEL01] A. Melahi, *Application de la commande par la logique Floue aux processus complexes*, thèse de Magister, (ENP 2001)
- [MEN95] T. M. Mendel, Fuzzy logic systems for engineering : A tutorial, *Proceedings of the IEEE*, Vol. 83, no. 3, pp. 345-377, Mar. 1989.
- [MIL92] W. T. Miller, R. Sutton, P.J. Werbos, *Neural Networks for Control*, The MIT Press, Cambridge, Massachusetts, London, 1992.
- [MOR85] R. G. Morgan, & Ü. Özgüner, A decentralized variable structure control algorithm for robotic manipulators, *IEEE J. Robotics and Automation*, RA-1, 1985, 57-65.
- [NAR90a] K.S. Narendra and K. Parthasarathy, Identification and Control of Dynamical Systems using Neural Networks," *IEEE Trans, Neural Networks*, vol. 1, pp. 4-27, 1990.
- [NAR90b] K.S. Narendra and K. Parthasarathy, Gradient Methods for the Optimisation of Dynamical Systems Containing Neural Networks, *IEEE Trans, Neural Networks*, vol. 2, NO. 2, pp. 252-262, 1990.
- [NED97] S. Nedjari, M.S. Boucherit, **D.Boukhetala** et H. Boukhari, Commande Adaptative Décentralisée Appliquée à un Bras de Robot Manipulateur, *IEEA, International Conference of the University of Batna*. Dec 97.
- [NEU85] C.-P. Neuman and V. D. Tourassis, Discrete dynamic robot models, *IEEE Transaction on Systems, Man and Cybernetics*, Vol. SMC-15, no. 2, Mar./Apr. 1985.
- [NIC90] S. Nicosia and P. Tomei, Robust control by using only joint positions measurements. *IEEE Transactions on Automatic Control*, Vol. 37, no. 8, pp. 1234-1237.
- [OSS89] K. A. Ossman, Indirect Adaptive Control for Interconnected Systems. *IEEE Trans. on Automatic Control*, Vol. 34, N° 8, pp 908-911, August 1989.
- [OZA91] T. Ozaki et al, Trajectory Control of Robotic Manipulators, *IEEE Trans. On Industrial Electronics*, vol. 38, NO. 3, 1991.
- [PAU81] R. P. Pau et al, Kinematic Control Equations for Simple Manipulators, *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, no. 6, pp. 445-449, June 1981.

- [PAU83] R.P. Paul, *Robots manipulators : mathématiques, programming and control*, MIT press 1983.
- [PAU86] R. P. Paul. *Robots manipulators: mathematics, programming and control*, MIT press, 1986.
- [PIC97] J. Pico and P. Moya, Localized BF-type networks for identification and adaptive control of discrete-time nonlinear systems, *3<sup>rd</sup> IFAC Symposium on Intelligent Components and Instruments for Control Applications-SICICA'97*, Annecy, France, pp. 653-658, June. 9-11, 1997.
- [PRO79] J. J. Procyk and E. H. Mamdani, A linguistic self-organizing process controller, *Automatica*, Vol. 15, pp. 15-30, 1979.
- [REN95] J. M. Renders, *Algorithmes génétiques et réseaux de neurones*, Edition HERMES, Mars 1995.
- [SAA94] M. Saad, L. A. Dessait, P. Bigras and K. El-haddad, Adaptive versus neural network Control : Application to Robotics, *Int. J. of Adaptive Cont. And Sign. Processing*, Vol. 8, pp 223-236, 1994.
- [SAD90] N . Sadegh , R . Horowitz, Stability and robustness analysis of a class of adaptive controllers for robotic manipulators, *Int. J. Robot. Res. ,* Vol 9, 1990.
- [SAN92] R. M. Sanner and J-J E. Slotine, "Gaussian Networks for Direct Adaptive Control," *IEEE Trans, Neural Networks*, vol. 3, N°6, Nov 1992.
- [SCH85] E. M. Scharf and N. J. Mandic, The application, of a fuzzy controller to the control of a multi degree-of-freedom robot arm", *Industrial Applications of fuzzy control*, M. Sugeno (ed.), Elsevier, Science publishers B.V. (North-Holland), 1985.
- [SER87] H. Seraji, A new approach to adaptive control of manipulators, *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 109, no. 3, pp. 193-202, Nov., 1987.
- [SER89] H. Seraji, Decentralized Adaptive Control of Manipulators: Theory, Simulation, and Experimentation, *IEEE Transactions in Robotics and Automation*, Vol. 5, No. 2, pp 183-201, April 1989.
- [SHI92] L. Shi, & S. K. Singh, Decentralized adaptive controller design for large-scale systems with higher order interconnections, *IEEE Trans. Autom. Control*, AC-37, pp 1106-1118. August. 1992.
- [SHU96a] D. Shukla and F. W. Paul, Computationally Efficient Control of Nonlinear Systems using Orthonormal Activation Function Based Neural Networks, *IFAC 1996, 13<sup>th</sup> Triennial World Congress*, San Francisco, USA, pp 55-60.
- [SHU96b] S. Ge. Shuzhi, Robust Adaptive Control of Robots Based on Static Neural Networks, *IFAC 1996, 13<sup>th</sup> Triennial World Congress*, San Francisco, USA, pp 139-144.
- [SIN85] S. N. Singh, Adaptive model following control of nonlinear robotic systems, *IEEE Transactions on Automatic Control*, 30, pp 1099-1100, 1985.
- [SLO83] J.-J. E. Slotine and S. Sastry, Tracking control of nonlinear systems using sliding surfaces with application to robot manipulators, *International Journal of Control*, Vol. 38, pp. 465-492, 1983.
- [SLO88] J. J. Slotine and W. Li, Adaptive Manipulator Control : A case study, *IEEE Trans. Auto. Contr.*, Vol.33, No. 11, pp. 995-1002, Nov. 1988.
- [SLO91] J. J. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ :Prentice Hall, 1991.

- [SPO95a] J. T. Spooner and K. M. Passino, Stable indirect adaptive control using fuzzy systems and neural networks, *IEEE Proceedings of the 34<sup>th</sup> Conference On Decision and Control*, New-Orleans, pp. 243-248, LA-Dec. 1995.
- [SPO95b] J. T. Spooner and K. M. Passino, Stable direct adaptive control using fuzzy systems and neural networks, *IEEE Proceedings of the 34<sup>th</sup> Conference On Decision and Control*, New-Orleans, pp. 249-254, LA-Dec. 1995.
- [SPO95c] J. T. Spooner and K. M. Passino, Adaptive control of a class of decentralized nonlinear systems, *Proceedings of the 34<sup>th</sup> Conference On Decision and Control*, New-Orleans, pp. 1135-1139, LA-Dec. 1995.
- [STE91] T.C. Steve Hsia, Ty A. Lasky, & Zhengyu Guo, Robust independent joint controller design for industrial robot manipulators, *IEEE Trans. Autom. Control*, AC-38, 1991, 21-25.
- [STO87] D.Stoten, Modeling and Control of Elastic Joint Robots, *ASME, Journal of Dynamics System Measurement and Control*, vol. 109, pp.310-319,1987.
- [SU94] C. Y. Su, & Y. Stepanenko, Adaptive sliding mode control of robot manipulators: general sliding manifold case, *Automatica*, 30, 1994, 1497-1500.
- [SUG85] M. Sugeno and T. Takagi, Derivation of fuzzy control rules from human operator's control actions", IFAC Fuzzy Information, France, 1985.
- [SUN94] Sunid Elanayar V. T. and Yung C. Shin, Radial Basis function Neural Network for Approximation and Estimation of Nonlinear Stochastic Dynamic systems, *IEEE Tran. on Neural Networks*, Vol.5, N° 4, july 1994.
- [TAN00] Y. Tang, M. Tomizuka, G. Guerrero, & G. Montemayor, Decentralized robust control of mechanical systems, *IEEE Trans. Autom. Control*, AC-45, 2000, 771-776.
- [TAN88] R. Tansheit and E. M. Sharf, Experiments with the use of rule-based self-organising controller for robotics application, *Fuzzy Sets and Systems* 26, pp. 195-214, North-Holland, 1988.
- [TAN98] Y. Tang, Terminal sliding mode control for rigid robots, *Automatica*, 34, 1998, 51-56.
- [TAR89] M. Tarokh, Approach to pole assignment by centralized and decentralized output feedback. *IEE Proc.*, 136, March 1989, pp 89-97.
- [TAR90] M. Tarokh, A discrete-time adaptive control scheme for robot manipulators, *Journal Of Robotic Systems*, 7(2), pp. 145-166, 1990.
- [TID99] Z.Tidjani, *Commande décentralisée floue:Appliquée aux robots manipulateurs* Thèse de Magister, Ecole Nationale Polytechnique, (ENP), 1999.
- [TRE97] A. Trebi-Ollennu and B. A.White. Robust output tracking for MIMO nonlinear systems : An adaptive fuzzy systems approach. *IEE Proceedings of Control Theory Application*, Vol. 144. no. 6, pp. 537-544, Nov. 1997.
- [TZA90] S. Tzafestas. N.P. Papanikolopoulos, Incremental fuzzy expert PID control, *IEEE transactions On Industrial Electronics*. Vol. 37, no. 5, Oct. 1990.
- [UTK78] V. I. Utkin, *Sliding Modes and their Application in Variable Structure Systems*. MIR Moscow. 1978.
- [VES97] V. Vesely. Robust Adaptive Decentralised Control for a Class of Complex Systems. *IFAC-IFIP-IMACS Conference. Control of Industrial Systems*, vol. 3, pp 69-74, 20-22 May 1997, Be fort, France
- [VIB87] C. Vibel *Robots : Principe et Contrôle*. Ellipse. Paris. 1987.
- [WAN73] S. H. Wang & E. J. Davison, On stabilization of decentralized control systems, *IEEE Trans. Autom. Control*, AC-18. 1973, 473-478.

- [WAN73] S. H. Wang, & E. J. Davison, On stabilization of decentralized control systems, *IEEE Trans. Autom. Control*, AC-18, 1973, 473–478.
- [WAN96] J. D. Wang, T. L. Lee & Y. T. Juang, New methods to design an integral variable structure controller, *IEEE Trans. Autom. Control*, AC-41, 1996, 140–143.
- [WAR92] K. Warwick, G. W. Irwin and K. G. Hunt. *Neural Networks for Control and Systems*. Peter Pergrinus LTD, on behalf of the Institution of Electrical Engineers. February 1992.
- [WAT97] M. Watheq El Kharachi and M. Sheirah, Fuzzy logic controllers : A comparative study, *Proceedings IFAC/IFIP/IMACS Conference*, pp. 209-215, Belfort, France, May 1997.
- [WEN94] C. Wen, Decentralized adaptive regulation. *IEEE Trans. Autom. Control*, AC-39, pp 1994, 2163–2166.
- [YAN96] X . Yang, Neural Network Model and System Used for Nonlinear Control. *IFAC*, 1996.
- [YEU88] K. S. Yeung, & Y. P. Chen, A new controller design for manipulators using the theory of variable structure systems, *IEEE Trans. Autom. Control*, AC-33, 1988, pp 200–206.
- [YU96] X. Yu, & M. Zhihong, On finite time mechanism: terminal sliding modes, *Proc. 1996 IEEE Int. Workshop on Variable Structure Systems*, Tokyo Japan, 1996, pp 164–167.
- [YUA91] J. Yuan and Y. Stepanenko, Robust control of robotic manipulators without velocity feedback, *International Journal of Robust and Nonlinear Control*, Vol. 1, pp. 203-213, 1991.
- [ZAD65] L. Zadeh, *Fuzzy sets*, Information and Control, Vol. 8, 1965.
- [ZHA93] Z.-Y. Zhao, M. Tomizuka and J. Isaka, Fuzzy gain scheduling of PID controllers, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, no. 5, Sep./Oct. 1993.
- [ZIA96] S. M. Ziauddin and A. M. S. Zalzala, Model-Based Neural Networks Compensation for Uncertainties of Robotic Arms, *IFAC 1996. 13<sup>th</sup> Triennial World Congress*, San Francisco, USA, pp 157-162.