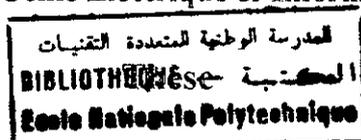


République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

## ECOLE NATIONALE POLYTECHNIQUE

DER de Génie Electrique & Informatique



Présentée par:

**Salim LABIOD**

Ingénieur d'Etat en Automatique de l'ENP

Pour obtenir le titre de

**MAGISTER EN AUTOMATIQUE**

Sujet

**Commande Adaptative par les Systèmes Flous;  
application aux robots manipulateurs**

Soutenue publiquement le *17/05/98* devant le jury composé de:

F. Boudjema	Maître de Conférences, ENP	Président
M. S. Boucherit	Maître de Conférences, ENP	Rapporteur
H. Chekireb	Chargé de Cours, ENP	Rapporteur
D. Boukhetala	Chargé de Cours, ENP	Examineur
R. Iloul	Chargé de Cours, ENP	Examineur
M. Tadjine	Docteur, ENP	Examineur

# AVANT PROPOS



*Les travaux présentés dans cette thèse ont été effectués au sein du Laboratoire de Commande des Processus (LCP) de la Direction d'Enseignement et de la Recherche en Génie Electrique & Informatique de l'Ecole Nationale Polytechnique d'Alger (ENP) sous la direction conjointe de Monsieur M. S. BOUCHERIT, Maître de Conférences à l'ENP, et Monsieur H. CHEKIREB, Chargé de Cours à l'ENP, qu'ils trouvent ici l'expression de toute ma gratitude pour les conseils avisés et l'encouragement constant qu'ils n'ont cessé de me prodiguer tout au long de cette étude.*

*Mes remerciements les plus sincères sont adressés à Monsieur F. BOUDJEMA, Maître de Conférences à l'ENP, pour l'honneur qu'il m'a fait en acceptant la présidence du jury de cette thèse.*

*J'exprime mes vifs remerciements à Monsieur D. BOUKHETALA, Chargé de Cours à l'ENP, d'avoir accepté de faire partie de ce jury. Qu'il trouve ici ma respectueuse gratitude.*

*Que Monsieur R. ILOUL, Chargé de Cours à l'ENP, trouve ici ma respectueuse gratitude pour avoir bien voulu examiner ce travail en acceptant de siéger dans ce jury.*

*Je remercie également Monsieur M. TADJINE, Docteur à l'ENP, pour l'intérêt qu'il a porté à ce travail en me faisant l'honneur de participer à ce jury. Qu'il trouve ici ma profonde gratitude.*

*Que mes amis O. BOUHALI, S. BOUMAZA, A. DJAHBAR, Z. TIDJANI, C. BOUBAKIR, E. BOUHASSOUN, K LAIDI, M. YEDDOU trouvent ici toute m'a gratitude pour leur compréhension, leur aide, leur ambiance bien sympathique et leur soutien moral et matériel qui n'ont cessé de m'apporter tout au long de l'élaboration de ce travail.*

*Je ne peut citer tout le monde, mais tous ceux qui m'ont aidé d'une façon ou d'une autre soient assurés de ma reconnaissance. Je les remercie tous, particulièrement ma famille, mes amis et le personnel de la Bibliothèque de l'ENP.*

*Enfin, je ne pourrais terminer ces remerciements sans une pensée à l'ensemble de mes enseignants, qui sont à l'origine de tout mon savoir.*

S. Labiod

# *TABLE DES MATIERES*

	<i>Page</i>
<i>INTRODUCTION GENERALE</i> .....	1
CHAPITRE 1 <u><i>SYSTEMES FLOUS</i></u>	
1.1 INTRODUCTION .....	4
1.2 ENSEMBLES FLOUS ET LOGIQUE FLOUE .....	5
1.2.1 Ensemble net .....	5
1.2.2 Ensemble flou .....	5
1.3 LES OPERATEURS EN LOGIQUE FLOUE .....	6
1.4 VARIABLES LINGUISTIQUES .....	7
1.5 FONCTION D'APPARTENANCE .....	8
1.6 IMPLICATION FLOUE .....	9
1.7 RAISONNEMENT FLOU .....	9
1.8 STRUCTURE D'UN SYSTEME FLOU .....	10
1.8.1 Base des règles floues .....	11
1.8.2 Moteur d'inférence floue .....	11
1.8.3 Fuzzification .....	12
1.8.4 Défuzzification .....	13
1.9 ANALYSE ET DESCRIPTION DU SYSTEME FLOU DE SUGENO .....	15
1.9.1 Architecture du système flou de Sugeno .....	16
1.10 ADAPTATION DES SYSTEMES FLOUS .....	17
1.11 INITIALISATION DES PARAMETRES DES SYSTEMES FLOUS .....	20
1.12 LA LOGIQUE FLOUE POUR LA COMMANDE DES SYSTEMES .....	21
1.13 CONCLUSION .....	22

---

CHAPITRE 2

**COMMANDE FLOUE ADAPTATIVE PAR INVERSION ITERATIVE**

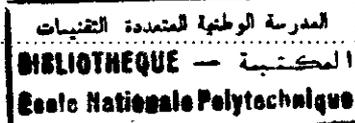
2.1 INTRODUCTION .....	23
2.2 APPRENTISSAGE ET INVERSION DES SYSTEMES FLOUS .....	24
2.3 COMMANDE PAR INVERSION ITERATIVE .....	24
2.4 COMMANDE PAR INVERSION ITERATIVE DES ROBOTS MANIPULATEURS .....	26
2.4.1 Modélisation du robot .....	26
2.4.2 Calcul de la commande .....	27
2.4.3 Résultats de simulation .....	29
2.5 CONCLUSION .....	35

CHAPITRE 3

**COMMANDE FLOUE ADAPTATIVE INDIRECTE**

3.1 INTRODUCTION .....	36
3.2 COMMANDE FLOUE ADAPTATIVE BASEE SUR L'IDENTIFICATION INVERSE .....	37
3.2.1 Modélisation du robot .....	37
3.2.2 Calcul de la commande .....	38
3.2.3 Résultats de simulation .....	39
3.3 COMMANDE FLOUE ADAPTATIVE BASEE SUR L'IDENTIFICATION DIRECTE .....	42
3.3.1 Modélisation du robot .....	42
3.3.2 Calcul de la commande .....	43
3.3.3 Résultats de simulation .....	44
3.4 COMMANDE FLOUE ADAPTATIVE INDIRECTE BASEE SUR LA MINIMISATION DE L'ERREUR EN VITESSE .....	47
3.4.1 Modélisation du robot .....	47
3.4.2 Calcul de la commande .....	50
3.4.3 Résultats de simulation .....	50
3.5 COMMANDE FLOUE ADAPTATIVE INDIRECTE BASEE SUR LA MINIMISATION DE L'ERREUR FILTREE .....	53
3.5.1 Modélisation du robot .....	53
3.5.2 Calcul de la commande .....	56
3.5.3 Résultats de simulation .....	56

3.6 CONCLUSION.....	59
---------------------	----



## CHAPITRE 4

### COMMANDE FLOUE ADAPTATIVE DIRECTE

4.1 INTRODUCTION .....	60
4.2 COMMANDE FLOUE ADAPTATIVE LINEARISANTE.....	61
4.2.1 Formulation du problème .....	61
4.2.2 Approximation par les systèmes flous .....	62
4.2.3 Synthèse de la commande .....	63
4.2.4 Analyse de la stabilité .....	64
4.2.5 Résultats de simulation.....	66
4.3 COMMANDE FLOUE ADAPTATIVE DIRECTE STABLE .....	69
4.3.1 Formulation du problème.....	69
4.3.2 Approximation par les systèmes flous .....	70
4.3.3 Synthèse de la commande .....	70
4.3.4 Analyse de la stabilité .....	71
4.3.5 Résultats de simulation.....	73
4.4 COMMANDE FLOUE ADAPTATIVE DIRECTE STABLE ETENDUE.....	76
4.4.1 Formulation du problème.....	76
4.4.2 Approximation par les systèmes flous .....	77
4.4.3 Synthèse de la commande .....	77
4.4.4 Analyse de la stabilité .....	79
4.4.5 Résultats de simulation.....	80
4.5 COMMANDE FLOUE ADAPTATIVE DIRECTE STABLE GENERALISEE .....	83
4.5.1 Structure de la commande .....	83
4.5.2 Analyse de la stabilité .....	84
4.5.3 Résultats de simulation.....	86
4.6 CONCLUSION.....	89
 <b>CONCLUSION GENERALE</b> .....	 90
 <b>ANNEXE</b> .....	 93
 <b>REFERENCES BIBLIOGRAPHIQUES</b> .....	 94

# *INTRODUCTION GENERALE*

Au cours des deux dernières décennies, un grand nombre de publications a été consacré au problème du calcul de la commande des robots manipulateurs. Au début, la commande était classique [BOI89], [SLO91], [VIB87], elle reposait sur la théorie de l'automatique linéaire, c'est-à-dire chaque liaison est représentée par un modèle linéaire et commandée indépendamment des autres liaisons. Grâce à sa facilité d'implantation, cette approche a remporté un grand succès auprès des industriels. Mais la nécessité d'atteindre de hautes performances, a imposé la prise en compte de la dynamique globale du robot manipulateur dans la synthèse de la commande [GUE95]. Dans cette approche, le robot manipulateur est représenté par des équations différentielles de seconde ordre non linéaires. Cependant, il s'est avéré que les performances des robots se dégradent en présence des variations paramétriques. Pour pallier à ce problème, la commande des robots manipulateurs a été orientée vers la commande dynamique adaptative [DAN89], [SAA94], [SLO88]. Les études théoriques développées de la commande adaptative considèrent deux schémas de commande:

- la commande adaptative directe, qui consiste à ajuster directement les paramètres du contrôleur en fonction des mesures d'erreur de performances.
- la commande adaptative indirecte, qui repose sur l'identification du procédé à chaque instant, ensuite remplacer les paramètres réels du procédé par leurs estimés dans le calcul de la commande.

Les approches de la commande adaptative développées dans la littérature nécessitent l'estimation des paramètres d'un modèle de connaissance complet du robot manipulateur. De surcroît, ces techniques adaptatives assurent la convergence de l'erreur uniquement en présence des incertitudes structurelles. Pour résoudre de tels problèmes, la proposition de nouvelles techniques de commande s'avère nécessaire. En effet, la recherche s'est orientée vers des techniques basées sur l'apprentissage à partir de données numériques et non sur un modèle de connaissance paramétré. L'une des approches les plus attractives est celle basée sur les systèmes flous.

La théorie des ensembles flous a été conçue au milieu des années soixante à l'université de Berkeley en Californie par le professeur iranien Lotfi A. Zadeh. En effet, les systèmes flous sont basés sur l'imitation des aspects approximatifs et qualitatifs du raisonnement humain. Des implications linguistiques floues de type 'si-alors' sont utilisées pour décrire le comportement d'un système mal défini. Cette théorie était mal perçue à l'époque du fait que le mode de penser était plus stricte et technique. De plus, le terme choisi « flou » n'a pas contribué à rendre son acceptation facile.

Zadeh a introduit les notions de la logique floue dans le domaine de l'automatique dès 1968 [TON84]. Le premier contrôleur flou a été implémenté par Mamdani et Assilian en 1974 [MAM77]. Néanmoins, si les concepts sont un peu anciens, le véritable essor des systèmes flous en automatique ne remonte qu'aux dix dernières années. Cet essor a notamment été favorisé par le succès de l'applications des systèmes flous à divers secteurs de l'industrie (électroménager, audiovisuel, engins mobiles, ...) [MEN95]; particulièrement grâce à l'effort de promotion des japonais. Devant cette passion des japonais pour les nouvelles technologies, et plus particulièrement pour la logique floue, des groupes de recherches spécialisés ont vu le jour sur tous les continents. C'est ainsi que divers développements sont réalisés, qu'ils soient fondamentaux, tels que les théorèmes d'approximation universelle, ou plus technique, tels que les méthodes d'apprentissage. Ces apports ont un impact direct sur les applications des systèmes flous à l'automatique.

Ainsi, les travaux présentés dans ce mémoire porte sur l'utilisation des systèmes flous dans le cadre d'une application liée à la commande adaptative avec poursuite d'une trajectoire de référence des robots manipulateurs. Ce sont des travaux de simulation effectués sur un robot manipulateur à trois degrés de liberté de type PORTOS.

Le premier chapitre est consacré à l'étude de la logique floue, une logique qui permet de traiter mathématiquement des données incertaines et imprécises. Au début, sont exposées les concepts de bases de la logique flous. Par la suite, nous présentons les systèmes flous, ainsi que leurs principales composantes et particulièrement celles du système flou de Sugeno. Après, quelques algorithmes d'optimisation et d'apprentissage sont fournis, suivi d'un panorama sur la commande floue.

Le but du deuxième chapitre est le développement d'une stratégie de commande adaptative basée sur la technique d'inversion itérative des systèmes flous. Dans cette approche, le calcul de la commande s'effectue en deux étapes:

- l'estimation des paramètres d'un modèle flou du système,
  
- le calcul de la commande à appliquer au robot par la technique d'inversion itérative des systèmes flous.

Le troisième chapitre présente la commande floue adaptative indirecte des robots manipulateurs. En premier lieu, nous proposons deux stratégies de commande adaptatives. La première est basée sur l'identification de la dynamique inverse du robot. La deuxième est basée sur l'identification de la dynamique directe. Ces deux approches nécessitent la mesure de l'accélération réelle des articulations. En seconde lieu, nous proposons de plus deux autres techniques de commande indirectes, qui ne nécessitent pas la mesure de l'accélération réelle des liaisons. Celles-ci se basent sur la théorie de Lyapunov dans l'élaboration de la loi d'adaptation paramétrique. En effet, la première technique minimise l'erreur en vitesse, par contre, la seconde minimise l'erreur en vitesse et celle en position. Dans ce chapitre, la loi de commande est calculée en combinant la méthode du couple calculé et le principe d'équivalence certaine.

Le quatrième chapitre aborde le problème de l'application des systèmes flous à la commande adaptative directe des robots manipulateurs. En effet, nous proposons quatre stratégies de commande. Dans ces dernières, la loi de commande est calculée afin de garantir la stabilité en boucle fermée ainsi que la convergence asymptotique vers zéro de l'erreur de poursuite.

# CHAPITRE 1

## *SYSTEMES FLOUS*

### 1.1 INTRODUCTION

C'est à Lotfi Zadeh, spécialiste renommé en automatique et en théorie des systèmes, à l'université de Berkeley, que revient le mérite d'avoir établi les bases théoriques de la logique floue [KOS92]. L'intérêt de la logique floue réside dans sa capacité à traiter l'imprécis, l'incertain et le vague. Elle est issue de la capacité de l'homme à décider et agir de façon pertinente malgré le flou des connaissances disponibles. En effet, la logique floue a été introduite pour approcher le raisonnement humain à l'aide d'une représentation adéquate des connaissances.

L'imitation des aspects qualitatifs et approximatifs du raisonnement humain a donné naissance à un autre type de système de commande qui trouvent de puissantes applications, surtout dans les situations où le système à commander est mal défini mathématiquement et/ou en présence d'une expertise qui puisse fournir des règles de commande en termes d'implications vagues et floues.

Ces dernières années, la logique floue a été appliquée à divers domaines. Cependant, la commande floue est le domaine qui suscite le plus de curiosité et s'avère être le champ d'application le plus actif à travers le monde [JAN95], [MEN95]. Parmi les nombreuses applications de cette discipline, on cite: l'ajustement des images TV, la reconnaissance de manuscrit, l'analyse des stocks, la commande d'un hélicoptère et l'autostationnement d'un modèle de voiture [LEE90], [MEN95].

Dans ce chapitre, nous exposons un bref rappel sur la théorie des ensembles flous et un aperçu général sur la logique floue, en insistant sur les idées utilisées en commande floue. Ensuite, nous présentons la structure des systèmes flous. Une analyse des systèmes flous de Sugeno est donnée, ainsi que leurs techniques d'apprentissage. Dans la dernière section, nous passons en revue quelques applications de la logique floue dans la commande.

## 1.2 ENSEMBLES FLOUS ET LOGIQUE FLOUE

La logique floue, introduite en 1965, peut être vue comme une extension de la logique booléenne. De plus, elle permet de traiter des variables linguistiques dont les valeurs sont des mots ou expressions du langage naturel. Cette partie fournit une introduction générale et un résumé des concepts de base essentiels pour l'étude de la logique floue.

### 1.2.1 ENSEMBLE NET

Un ensemble  $A$  sur un référentiel  $U$  dont les limites sont définies précisément s'appelle ensemble net ou ordinaire (figure 1.1 (a)). La description d'un ensemble net consiste en un énoncé des éléments qui appartiennent à cet ensemble, et qui le constituent. Une alternative pour le définir est de donner la condition pour laquelle un élément  $u \in A$ . Alors, l'ensemble  $A$  peut être défini tel que  $A = \{ u \mid u \text{ vérifie la condition} \}$ . De plus, on peut introduire une fonction caractéristique pour  $A$ , notée  $\mu_A(u)$  telle que:

$$\mu_A(u) = \begin{cases} 1 & u \in A \\ 0 & u \notin A \end{cases} \quad (1.1)$$

### 1.2.2 ENSEMBLE FLOU

La notion d'ensemble flou provient du constat que " très souvent, les classes d'objets rencontrés dans le monde physique ne possèdent pas de critères d'appartenance bien définis " [MEN95].

Mathématiquement, un ensemble flou  $A$  est défini sur un référentiel  $U$ , par une fonction d'appartenance, notée  $\mu_A$ , qui appliquée à un élément  $u$  de  $U$ , retourne un degré d'appartenance  $\mu_A(u)$  de  $u$  à  $A$  (figure 1.1 (b)). Un ensemble flou peut être représenté comme un ensemble de paires ordonnées:  $A = \{ (u, \mu_A(u)) \mid u \in U \}$ .

Si  $U$  est discret,  $A$  est représenté par:

$$A = \sum_{u_i \in U} \mu(u_i) / u_i \quad (1.2)$$

Si  $U$  est continu,  $A$  est représentée par:

$$A = \int_U \mu(u) / u \quad (1.3)$$

Les ensembles flous ont le grand avantage de constituer une représentation mathématique des termes linguistiques largement utilisés dans l'expression de connaissances expertes, qualitatives et qui sont manipulées par la logique floue.

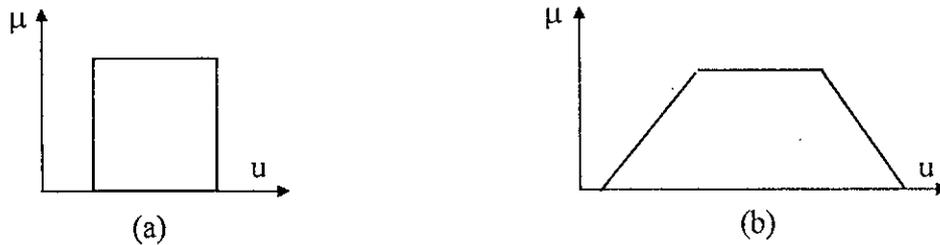


Figure 1.1. Exemples de fonctions d'appartenance: (a) logique classique; (b) logique floue.

### 1.3 LES OPERATEURS EN LOGIQUE FLOUE

Disposant d'une représentation formelle de classes floues sous la forme de fonctions d'appartenance, il est naturel de chercher à faire avec les ensembles flous ce que l'on fait avec les ensembles classiques. Zadeh a ainsi généralisé les opérateurs des ensembles classiques au cas flou. Les opérateurs appliqués aux ensembles flous sont définis à partir de leurs fonctions d'appartenance.

Soit A et B deux ensembles flous définis dans le référentiel U avec les fonctions d'appartenance  $\mu_A$  et  $\mu_B$  respectivement.

**Définition 1: Union (Disjonction):** La fonction d'appartenance  $\mu_{A \cup B}$  de l'union  $A \cup B$  ou A OR B est définie pour tout  $u \in U$  par:

$$\mu_{A \cup B}(u) = \mu_A(u) \dot{+} \mu_B(u) \quad (1.4)$$

Le symbole " $\dot{+}$ " représente la co-norme triangulaire [LEE90]. En effet, le Max est la co-norme triangulaire la plus utilisée dans le domaine de la commande:

$$\mu_A(u) \dot{+} \mu_B(u) = \text{Max}(\mu_A(u), \mu_B(u)) \quad (1.5)$$

**Définition 2: Intersection (Conjonction):** La fonction d'appartenance  $\mu_{A \cap B}$  de l'intersection  $A \cap B$  ou A AND B est définie pour tout  $u \in U$  par:

$$\mu_{A \cap B}(u) = \mu_A(u) * \mu_B(u) \quad (1.6)$$

Le symbole "\*" représente la norme triangulaire [LEE90]. Deux normes triangulaires sont le plus souvent utilisées pour évaluer l'intersection floue:

- Min:  $\mu_A(u) * \mu_B(u) = \text{Min}(\mu_A(u), \mu_B(u))$

- Produit algébrique:  $\mu_A(u) * \mu_B(u) = \mu_A(u) \times \mu_B(u)$

**Définition 3: Complémentation:** La fonction d'appartenance  $\mu_{\bar{A}}$  du complément de l'ensemble A; notée  $\bar{A}$  ou NOT A, est définie pour tout  $u \in U$  par:

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u) \quad (1.7)$$

**Définition 4: Produit cartésien:** Soit  $A_1, \dots, A_n$  des ensembles flous respectivement définis sur  $U_1, \dots, U_n$ , le produit cartésien de  $A_1, \dots, A_n$  est un ensemble flou défini sur  $U_1, \dots, U_n$ , de fonction d'appartenance:

$$\mu_{A_1 \times \dots \times A_n}(u_1, \dots, u_n) = \mu_{A_1}(u_1) * \dots * \mu_{A_n}(u_n) \quad (1.8)$$

**Définition 5: Relation floue:** Une relation floue représente le degré de présence, ou d'absence d'une association entre les éléments de deux ou de plusieurs ensembles flous. Une relation floue d'ordre n est un ensemble flou défini sur  $U_1 \times \dots \times U_n$  par l'expression suivante:

$$R_{U_1 \times \dots \times U_n} = \left\{ \left( (U_1, \dots, U_n), \mu_R(u_1, \dots, u_n) \right) \mid (u_1, \dots, u_n) \in U_1 \times \dots \times U_n \right\} \quad (1.9)$$

**Définition 6: Composition des règles floues:** Soit R et S deux relations floues définies respectivement dans  $U \times V$  et  $V \times W$ . La composition de R et S est un ensemble flou, symbolisé par  $R \circ S$ , de fonction d'appartenance:

$$\mu_{R \circ S}(u, w) = \left\{ (u, w), \sup_{v \in V} (\mu_R(u, v) * \mu(v, w)) \right\} \quad (1.10)$$

## 1.4 VARIABLE LINGUISTIQUE

C'est une variable dont les valeurs ne sont pas des nombres, mais des mots ou phrases exprimés en langage naturel [LEE90], [MEN95]. La raison pour laquelle on utilise cette représentation, est que le caractère linguistique est moins spécifique que le caractère numérique.

Une variable linguistique est généralement représentée par un triplet  $(x, T(x), U)$ , dans lequel x est le nom de la variable linguistique (vitesse, position, erreur, ...),  $T(x)$  est l'ensemble des valeurs linguistiques; qui sont utilisés pour caractériser x. Par exemple, si l'erreur est considérée comme variable linguistique, définie sur le référentiel  $U = [-10, +10]$ , ses valeurs linguistiques peuvent être définis comme suit:

$T(\text{erreur}) = \{ \text{Négative Grande (NG)}, \text{Négative Petite (NP)}, \text{Environ Zéro (EZ)}, \text{Positive Petite (PP)}, \text{Positive Grande (PG)} \}$ .

Ces symboles linguistiques peuvent être considérés comme des ensembles flous dont les fonctions d'appartenance sont représentées sur la figure 1.2.

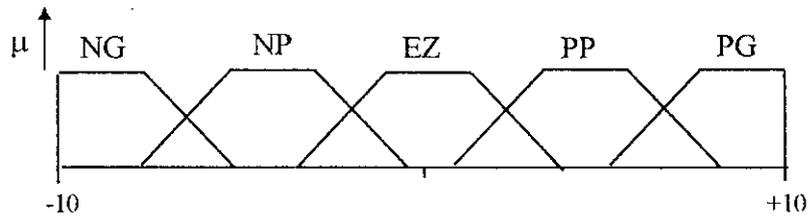


Figure 1.2. Représentation de la variable linguistique *erreur*.

## 1.5 FONCTION D'APPARTENANCE

Afin de permettre un traitement numérique des variables linguistiques dans la prise de décisions floues sur ordinateur, une définition des variables linguistiques à l'aide de fonctions d'appartenance s'impose. Dans ce contexte, on associe à chaque valeur de la variable linguistique une fonction d'appartenance désignée par  $\mu_A(x)$ , où  $x$  est la variable linguistique, tandis que  $A$  indique l'ensemble concerné. Une valeur précise de  $\mu_A(x)$ , sera désignée par le degré ou le facteur d'appartenance. Il est à noter que l'ensemble des éléments de  $x$  pour lesquels  $\mu_A(x) > 0$ , est appelé *support* de  $A$ . Le plus souvent, on utilise pour les fonctions d'appartenance les fonctions suivantes (figure 1.3) [BUH94], [JAN95], [LEE90]:

- *Fonction triangulaire*: Elle est définie par trois paramètres  $\{a, b, c\}$ , qui déterminent les coordonnées des trois sommets.

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (1.11)$$

- *Fonction trapézoïdale*: Elle est définie par quatre paramètres  $\{a, b, c, d\}$ :

$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \quad (1.12)$$

- *Fonction gaussienne*: Elle est définie par deux paramètres  $\{\sigma, m\}$ :

$$\mu(x) = \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right) \quad (1.13)$$

- *Fonction sigmoïdale*: Une fonction sigmoïdale est définie par deux paramètres  $\{a, c\}$ :

$$\mu(x) = \frac{1}{1 + \exp(-a(x-c))} \quad (1.14)$$

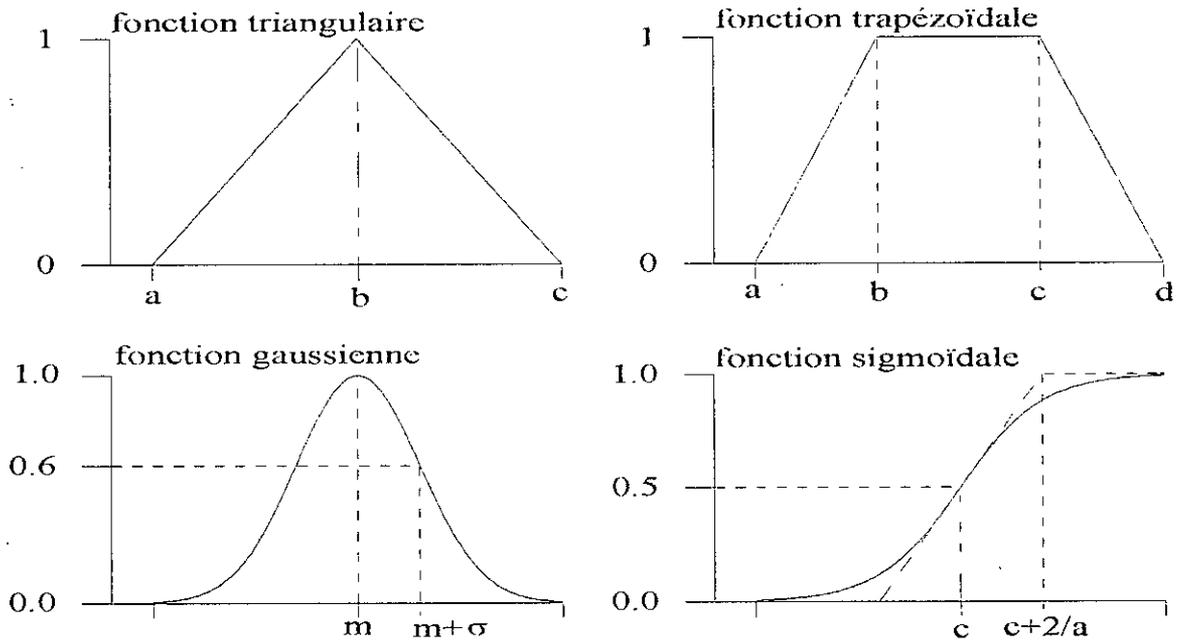


Figure 1.3. Formes usuelles des fonctions d'appartenance.

## 1.6 IMPLICATION FLOUE

L'implication floue est un opérateur qui permet d'évaluer le degré de vérité d'une règle R de la forme 'Si x est A alors y est B' à partir des valeurs de la prémisse d'une part, et de celle de la conclusion d'autre part.

$$\mu_R(x, y) = \text{Imp}(\mu_A(x), \mu_B(y)) \quad (1.15)$$

Les opérateurs les plus utilisés en commande floue sont les implications de Mamdani et de Larsen.

- Implication de Mamdani

$$\mu_R(x, y) = \min(\mu_A(x), \mu_B(y))$$

- Implication de Larsen

$$\mu_R(x, y) = \mu_A(x) \times \mu_B(y)$$

## 1.7 RAISONNEMENT FLOU

On rappelle qu'en logique classique le modus ponens permet, à partir de la règle 'Si x est A alors y est B' et du fait 'x est A', de conclure le fait 'y est B', qui sera ajouté à la base des faits. Zadeh a étendu ce principe au cas flou, principe que l'on appelle alors modus ponens généralisé. Le modus ponens et le modus ponens généralisé se résument comme suit:

	<i>Modus Ponens</i>	<i>Modus Ponens généralisé</i>
<i>Fait</i>	$x \text{ est } A$	$x \text{ est } A'$
<i>Règle</i>	$\text{Si } x \text{ est } A \text{ Alors } y \text{ est } B$	$\text{Si } x \text{ est } A \text{ Alors } y \text{ est } B$
<i>Déduction</i>	$y \text{ est } B$	$y \text{ est } B'$

A partir de la règle 'Si A alors B' et du fait A', on déduit un nouveau fait B' qui est caractérisé par un ensemble flou dont la fonction d'appartenance est donnée par:

$$\mu_{B'}(y) = \text{SUP}_x (\mu_{A'}(x) * \mu_R(x, y)) \quad (1.16)$$

Les fonctions d'appartenance  $\mu_{A'}$  et  $\mu_R$  caractérisent respectivement le fait A' et la règle.

## 1.8 STRUCTURE D'UN SYSTEME FLOU

Un système flou est un système à base de connaissance particulière. L'architecture de base d'un système flou (figure 1.4) se compose de quatre modules principaux [BUH94], [JAN95], [MEN95], [PED93], [WAN92b]:

- La base des règles floues, ou base de connaissances, contient les règles floues décrivant le comportement du système; elle est le cœur du système entier dans le sens où tous les autres composants sont utilisés pour interpréter et combiner ces règles pour former le système final.
- Le moteur d'inférence floue transforme, à l'aide des techniques de raisonnement flou la partie floue issue de la fuzzification en une nouvelle partie floue.
- La fuzzification transforme la grandeur physique d'entrée en une grandeur floue.
- La défuzzification transforme la grandeur floue issue de l'inférence en une grandeur physique.

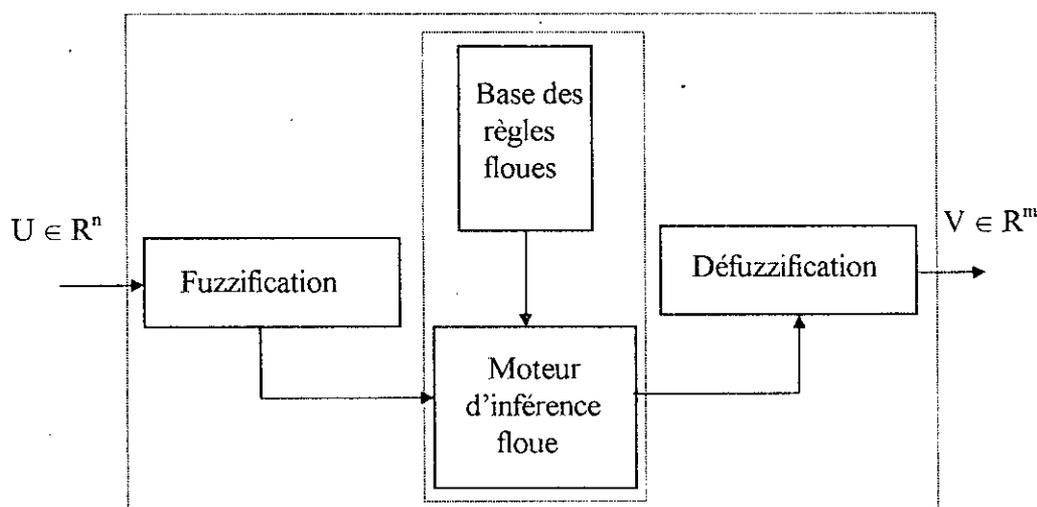


Figure 1.4. Configuration de base d'un système flou.

### 1.8.1 Base des règles floues

Une base des règles floues  $R$  est une collection de règles floues de la forme SI-ALORS.  $R=[R_1, \dots, R_M]$ , telle que:

$$R_i : \text{SI } x_1 \text{ est } A_{i1} \text{ et } \dots \text{ et } x_n \text{ est } A_{in} \text{ ALORS } y \text{ est } B_i$$

ou sous la forme de Sugeno:

$$R_i : \text{SI } x_1 \text{ est } A_{i1} \text{ et } \dots \text{ et } x_n \text{ est } A_{in} \text{ ALORS } y_i = f_i(x)$$

avec  $f_i(x)$  est un polynôme.

### 1.8.2 Moteur d'inférence floue

Le moteur d'inférence floue utilise la base des règles floues pour effectuer une transformation à partir des ensembles flous dans l'espace d'entrée vers les ensembles flous dans l'espace de sortie en se basant sur les opérations de la logique floue. L'antécédent de la règle  $R_i$  définit un produit cartésien de  $A_{i1}, \dots, A_{in}$ , et la règle elle-même  $R_i$  est vue comme une implication. Soit  $A_x$  un ensemble flou dans  $U$ , alors chaque règle  $R_i$  détermine un ensemble flou  $B'_i = A_x \circ R_i$  dans  $V$ . La fonction d'appartenance de  $B'_i$  est donnée par la règle compositionnelle (1.16) [MEN95]:

$$\mu_{B'_i}(y) = \sup_{x \in A_x} (\mu_{A_x}(x) * \mu_{R_i}(x, y)) \quad (1.17)$$

L'ensemble des  $M$  règles constituant la base des règles floues sont liées par l'opérateur de disjonction "ou". Ainsi, l'ensemble flou final  $B' = A_x \circ R$ , est donné par la relation [MEN95]:

$$\begin{cases} B' = B'_1 \dot{+} \dots \dot{+} B'_M \\ \mu_{B'}(y) = \mu_{A_x \circ R_1}(y) \dot{+} \dots \dot{+} \mu_{A_x \circ R_M}(y) \end{cases} \quad (1.18)$$

Dans le jeu de règles du système flou interviennent les opérateurs flous ET (AND) et OU (OR). L'opérateur AND s'applique aux variables à l'intérieur d'une règle, tandis que l'opérateur OR lie les différentes règles. Plusieurs types de raisonnement flou ont été proposés dans la littérature suivant la réalisation des opérateurs flous ET et OU et le type des règles floues utilisées. Les moteurs d'inférence floue les plus utilisés peuvent être classés dans trois types [HOR92], [JAN92], [LEE90], [LIN91], [SUG88b], [WAN92a].

- **Type 1 (méthode de Tsukamoto):** Dans ce cas, des fonctions monotoniques sont associées aux variables de sortie. La sortie totale est une moyenne pondérée des degrés de confiances de chaque règle floue et des valeurs des fonctions des variables de sortie.

- Type 2 (*méthode de Mamdani*): Dans ce type, les conclusions sont symboliques ou linguistiques et le calcul de la sortie nécessite l'utilisation d'une méthode de défuzzification.
- Type 3 (*méthode de Sugeno*): Les règles floues de type Sugeno sont utilisées. En effet, la conclusion de chaque règle floue a une forme polynomiale ou plus générale. La sortie finale est égale à la moyenne pondérée de la sortie de chaque règle floue.

La figure 1.5 illustre les trois types du raisonnement flou pour un système flou à deux entrées et une base de connaissances de deux règles floues. On constate que les différences viennent de la spécification de la partie conclusion d'une part, et de la méthode de défuzzification d'autre part.

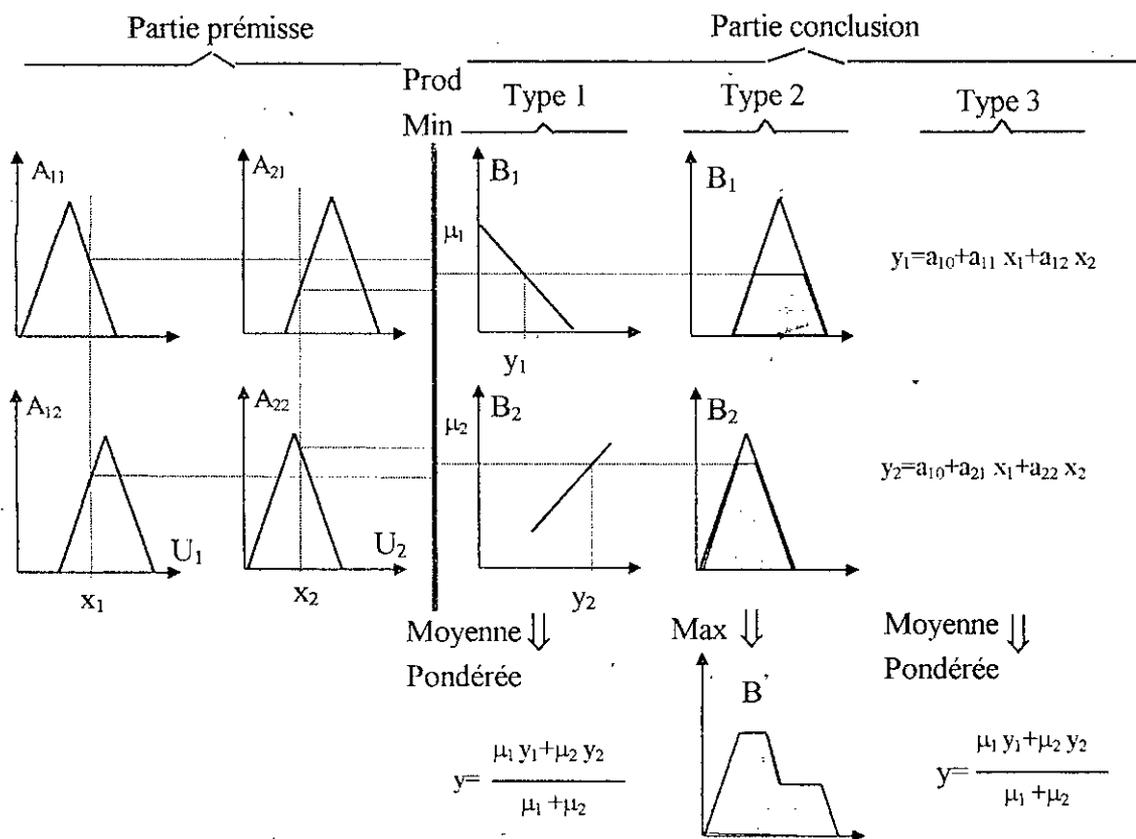


Figure 1.5. Types de règles floues et de moteurs d'inférence floue.

### 1.8.3 Fuzzification

La fuzzification consiste à relier le point numérique  $x_0 = (x_{10}, \dots, x_{n0})^T$  de  $U$  à l'ensemble flou  $A_x = [A_{x1}, \dots, A_{xn}]$  dans  $U = U_1 \times \dots \times U_n$ , où  $A_{xi}$  est un ensemble flou dans  $U_i$ . Il existe deux méthodes de fuzzification suivant la définition de  $A_x$  [LEE90], [MEN95].

- $A_x$  est un singleton flou défini par:

$$\mu_{A_x}(x) = \begin{cases} 1 & \text{si } x = x_0 \\ 0 & \text{si } x \neq x_0 \end{cases} \quad (1.19)$$

Dans ce cas, on considère que la valeur de  $x$  est précise et certaine (Figure 1.6 (a)).

- $A_x$  est un ensemble flou de fonction d'appartenance:  $\mu_{A_x}(x_0) = 1$  et  $\mu_{A_x}(x)$  décroît lorsque  $x$  s'éloigne de  $x_0$ .

Dans ce cas, est prise en compte le comportement de la variable autour de la valeur  $x_0$ . Par exemple, l'erreur de mesure est modélisée par une fonction d'appartenance triangulaire (Figure 1.6 (b)).

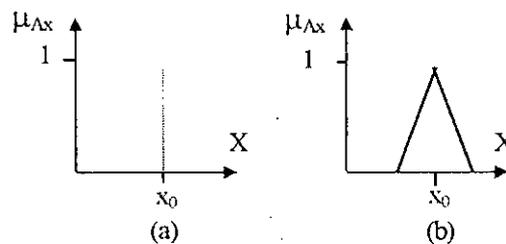


Figure 1.6. Deux stratégies de fuzzification.

### 1.8.4 Défuzzification

Le rôle de la défuzzification est de transformer la partie floue issue de l'inférence en une grandeur numérique. Malheureusement, il n'y a pas une procédure systématique pour choisir la stratégie de défuzzification. Cependant, la défuzzification est un art plus qu'une science [MEN95]. Comme on s'intéresse à l'application de la logique floue en commande, un critère de choix d'une méthode de défuzzification est la simplicité du calcul. Ce critère a conduit aux défuzzificateurs suivants: méthode du maximum, méthode des maxima, méthode du centre de gravité, méthode des hauteurs pondérées et la méthode des hauteurs pondérées modifiée [BUH94], [JAN95], [LEE90], [MEN95].

#### 1.8.4.1 Méthode du maximum

La méthode du maximum examine l'ensemble flou  $B$  issu de l'inférence et choisit comme sortie la valeur  $y$  pour laquelle  $\mu_B(y)$  est un maximum. Cependant, ce défuzzificateur présente un certain inconvénient lorsqu'il existe plusieurs valeurs pour lesquelles  $\mu_B(y)$  est un maximum.

### 1.8.4.2 Méthode de la moyenne des maxima

Ce défuzzificateur examine l'ensemble flou  $B'$  issu de l'inférence et détermine en premier temps les valeurs  $y_i$  pour lesquelles  $\mu_{B'}(y_i)$  est un maximum. Ensuite, on calcule la moyenne de ces valeurs comme résultat de défuzzification.

### 1.8.4.3 Méthode du centre de gravité

Dans ce cas, on calcule le centre de gravité  $\bar{y}$  de l'ensemble flou résultant  $B'$  et considère cette valeur comme résultat de défuzzification.

$$\bar{y} = \frac{\int_S y \mu_{B'}(y) dy}{\int_S \mu_{B'}(y) dy} \quad (1.20)$$

où  $S$  représente le support de  $B'$ . Souvent, on utilise la version discrète de l'intégral:

$$\bar{y} = \frac{\sum_{i=1}^I y_i \mu_{B'}(y_i)}{\sum_{i=1}^I \mu_{B'}(y_i)} \quad (1.21)$$

Il est à noter que le centre de gravité est généralement difficile à calculer. De ce fait, cette méthode est la plus coûteuse en temps de calcul.

### 1.8.4.4 Méthode des hauteurs pondérées

La valeur réelle de la sortie est donnée par la relation suivante :

$$\bar{y} = \frac{\sum_{i=1}^M \mu_i \bar{y}_i}{\sum_{i=1}^M \mu_i} \quad (1.22)$$

où

$\mu_i$  : représente le degré de confiance ou d'activation de la règle  $R_i$ ,

$\bar{y}_i$  : le centre de gravité de l'ensemble flou de la variable de sortie associée à la règle  $R_i$ .

Cette méthode est très simple à implémenter. En effet, les centres de gravité des fonctions d'appartenance sont connus a priori. Cependant, elle souffre d'un inconvénient du fait qu'elle n'utilise pas la forme entière des fonctions d'appartenance de la conclusion. Elle utilise seulement le centre  $\bar{y}_i$

du support de la fonction d'appartenance de la conclusion, sans tenir compte du fait que la fonction d'appartenance est étroite ou large.

#### 1.8.4.5 Méthode des hauteurs pondérées modifiée

Contrairement à la méthode précédente, la sortie du système flou est calculée en tenant compte du support de la fonction d'appartenance de la conclusion, par l'expression :

$$\bar{y} = \frac{\sum_{i=1}^M (\mu_i \bar{y}_i / \delta_i^2)}{\sum_{i=1}^M (\mu_i / \delta_i^2)} \quad (1.23)$$

où  $\delta$  est une mesure de l'étendue de la fonction d'appartenance de la variable de sortie associée à la règle  $R_i$ . Pour des fonctions d'appartenance triangulaires ou trapézoïdales,  $\delta$  peut être le support du triangle ou du trapèze, et pour une fonction gaussienne,  $\delta$  peut être sa variance.

### 1.9 ANALYSE ET DESCRIPTION DU SYSTEME FLOU DE SUGENO

Il existe un grand nombre de possibilités de réalisation de systèmes flous. En effet, pour chaque module constitutif d'un système flou (figure 1.4), il existe une multitude de choix différents, et chaque combinaison de ces choix engendre une classe de systèmes flous. Dans ces travaux, nous nous intéressons au modèle flou de type Sugeno, développé initialement par Sugeno et Takagi pour la modélisation des systèmes à partir de données numériques [SUG85]. Une règle typique dans le système flou de Sugeno est de la forme suivante:

$$R_i: \text{ SI } x_1 \text{ est } F_1^i \text{ et...et } x_n \text{ est } F_n^i \text{ ALORS } y_i = f(x)$$

où  $F_i^j$ , sont des ensembles flous dans l'espace d'entrée, tandis que  $f(x)$  est une fonction numérique dans l'espace de sortie. En général,  $f(x)$  est une fonction polynomiale en fonction des variables d'entrées, mais elle peut être aussi une fonction arbitraire tant qu'elle puisse décrire convenablement le comportement du système étudié. Si  $f(x)$  est un polynôme d'ordre un, on a affaire donc à un système de Sugeno d'ordre un. Si par contre,  $f(x)$  est un polynôme d'ordre zéro, on a donc un système de Sugeno d'ordre zéro. Etant donnée que chaque règle possède une conclusion numérique, la sortie totale du système flou est obtenue par le calcul d'une moyenne pondérée, et de cette manière, le temps consommé par la procédure de défuzzification est évité.

Comme notre but est de développer une loi d'adaptation des paramètres du système flou. Il est donc indispensable de donner l'expression analytique de la sortie du système flou de Sugeno, et de présenter ensuite l'algorithme d'estimation des paramètres du système flou pour approximer une fonction non linéaire quelconque à partir de données numériques.

### 1.9.1 Architecture du système flou de Sugeno

Notons par  $x_1, \dots, x_n$  les entrées du système flou, et par  $y$  sa sortie. Pour chaque  $x_i$  est associé  $m_i$  ensembles flous  $F_i^j$  dans  $U_i$ , tel que pour  $x_i$  quelconque de  $U_i$ , il existe au moins un degré d'appartenance  $\mu_{F_i^j}(x_i) \neq 0$ , où  $i = 1, 2, \dots, n$ , et  $j = 1, 2, \dots, m_i$ . La base des règles du système flou comporte  $M = \prod_{i=1}^n m_i$  règles telles que:

$$R_l: \text{SI } x_1 \text{ est } F_1^{l_1} \text{ et } \dots \text{ et } x_i \text{ est } F_i^{l_i} \text{ et } \dots \text{ et } x_n \text{ est } F_n^{l_n} \text{ ALORS } y_l(x) = a_0^l + a_1^l x_1 + \dots + a_n^l x_n$$

avec  $l=1, \dots, M$ ,  $i = 1, \dots, n$  et  $1 \leq l_i \leq m_i$ .

Chaque règle floue  $R_l$  corresponde à une combinaison de  $(F_1^{l_1}, \dots, F_i^{l_i}, \dots, F_n^{l_n})$  ensembles flous; où  $i = 1, \dots, n$  et  $1 \leq l_i \leq m_i$ . En effet, la base de connaissances comporte toutes les combinaisons possibles des ensembles flous des variables d'entrée.

La sortie du système flou est donnée par la relation suivante [LEU96], [JAN95], [KOS92]:

$$y = \frac{\sum_{l=1}^M \mu_l y_l}{\sum_{l=1}^M \mu_l} \quad (1.24)$$

avec

$$\mu_l = \prod_{i=1}^n \mu_{F_i^{l_i}}(x_i); 1 \leq l \leq M$$

et qui représente le degré de confiance ou d'activation de la règle  $R_l$ .

Dans le cas où les paramètres des prémisses sont fixés a priori par le concepteur. Alors, les seuls paramètres ajustables seront ceux de la conclusion. Ainsi, La sortie finale peut être écrite sous la forme suivante:

$$y = W(x) A \quad (1.25)$$

où  $A$ : est une matrice regroupant les paramètres  $a_i^l$ ,

$W$ : est une matrice de fonctions floues de base de la forme:

$$W_{ij}(x) = \frac{\mu_i x_j}{\sum_{l=1}^M \mu_l} \quad (1.26)$$



des systèmes flous répondant à des critères de performance et d'optimisation. Pour dériver l'algorithme d'apprentissage des systèmes flous, nous optons pour la technique dite filtre de Kalman étendu; adopter à partir de la théorie de l'identification. Les algorithmes basés sur le filtre de Kalman étendu ont été récemment introduits pour estimer les paramètres internes des réseaux neuronaux [WON96]. En effet, c'est des algorithmes d'entraînement fonctionnant en ligne, où les paramètres sont ajustés immédiatement après la présentation de chaque exemple d'entraînement.

Considérons un système flou caractérisé par un vecteur de paramètre  $\theta$ . Soit un ensemble de données entrée-sortie  $(x(k), d(k))$ . Notre objectif est de trouver les valeurs du vecteur  $\theta$  pour que la sortie du système flou approche le mieux possible la sortie désirée  $d(k)$ ; c'est-à-dire pour avoir  $d(k) = y(x(k); \theta)$ . L'approche du filtre de Kalman étendu consiste à linéariser à tout instant le système flou autour du vecteur estimé  $\hat{\theta}$ . Ceci revient à écrire:

$$\begin{cases} d(k) = y(x(k); \hat{\theta}(k-1)) + \psi^T(k) (\theta - \hat{\theta}(k-1)) \\ \psi(k) = \left. \frac{\partial y(x(k); \theta)}{\partial \theta} \right|_{\hat{\theta}(k-1)} \end{cases} \quad (1.28)$$

en posant:

$$z(k) = \psi^T(k) \theta \quad (1.29)$$

l'équation (1.29) devient:

$$z(k) = d(k) - y(x(k); \hat{\theta}(k-1)) + \psi^T(k) \hat{\theta}(k-1) \quad (1.30)$$

Nous sommes ainsi ramenés au problème d'estimation récursif de l'état du système linéaire constitué de l'équation d'état (1.29) et de l'équation de mesure (1.30). La forme générale, bien connue, de la solution est [AST89], [GOO84], [LJU87]:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P(k) \psi(k) \{z(k) - \psi^T(k) \hat{\theta}(k-1)\} \quad (1.31)$$

En remplaçant  $z(k)$  par son expression, nous aboutissons à l'algorithme d'estimation des paramètres  $\theta$  donné par la relation suivante:

$$\begin{cases} \hat{\theta}(k) = \hat{\theta}(k-1) + P(k) \psi(k) e(k) \\ e(k) = d(k) - y(x(k); \hat{\theta}(k-1)) \end{cases} \quad (1.32)$$

où  $P(k)$  est le gain de l'algorithme d'estimation. Les différentes méthodes d'identification récursives citées dans la littérature se distinguent essentiellement par le choix de ce gain, qui dans tous les cas, doit être pris égal à une matrice positive. Les méthodes, les plus souvent utilisées dans la littérature sont: la méthode du gradient, la méthode du gradient modifiée et la méthode des moindres carrés récursifs [AST89], [GOO84], [JAN95], [MEN95]:

- *Méthode du gradient simple*

C'est la méthode la plus simple à mettre en œuvre. Le gain  $P(k)$  est choisi égal au produit d'un gain scalaire par la matrice identité:

$$P(k) = \alpha I, \quad \alpha > 0 \quad (1.33)$$

- *Méthode du gradient modifiée*

Dans cette méthode, le gain  $P(k)$  est choisi variable. Il est donné par de la relation suivante:

$$P(k) = \frac{\alpha_1 I}{\alpha_2 + \psi^T(k) \psi(k)}, \quad \alpha_1 > 0, \alpha_2 > 0 \quad (1.34)$$

- *Méthode des moindres carrés récursifs*

Dans ce cas, le gain d'estimation  $P(k)$  est donné par la relation récursive suivante:

$$P(k) = P(k-1) + \frac{\psi(k) P(k-1) \psi^T(k)}{1 + \psi^T(k) P(k-1) \psi(k)} \quad (1.35)$$

avec  $P(0) = \alpha I$  où  $\alpha > 0$

Nous remarquons que ces méthodes nécessitent le calcul du gradient  $\psi = \frac{\partial y}{\partial \theta}$ . En effet, ce gradient est calculé par la méthode de la rétropropagation utilisée dans les réseaux neuronaux [FRE92], [FUK92], [HEN97], [JAN95], [HAM97], [MEN95], [MIL92], [NAR91].

Dans un système flou de Sugeno, nous avons  $\theta = [c \ v \ a]^T$ , nous avons donc:

$$\frac{\partial y}{\partial \theta} = \left[ \frac{\partial y}{\partial c} \ \frac{\partial y}{\partial v} \ \frac{\partial y}{\partial a} \right]^T \quad (1.36)$$

En utilisant l'expression de la sortie du système flou donnée par l'équation (1.24), il vient:

$$\frac{\partial y}{\partial c_i^j} = \sum_{k \in I_{c_i^j}} \frac{\partial y}{\partial \mu_k} \frac{\partial \mu_k}{\partial \mu_{F_i^j}} \frac{\partial \mu_{F_i^j}}{\partial c_i^j} \quad (1.37)$$

où  $I_{c_i^j}$  est l'ensemble des indices ( $l$ ) des règles floues ( $R_l$ ) dont lesquelles apparaît l'ensemble flou  $F_i^j$ .

$$\frac{\partial y}{\partial \mu_k} = \frac{y_k - y}{\sum_{l=1}^M \mu_l} \quad (1.38)$$

$$\frac{\partial \mu_k}{\partial \mu_{F_i^j}} = \prod_{l=1}^n \mu_{F_l^k}(x_l) \quad (1.39)$$

Pour une fonction d'appartenance gaussienne (1.27), on a:

$$\frac{\partial \mu_{F_i^j}}{\partial c_i^j} = v_i^{j2} (x_i - c_i^j) \mu_{F_i^j}(x_i) \quad (1.40)$$

Après calcul, il vient:

$$\frac{\partial y}{\partial c_i^j} = \frac{v_i^{j2} (x_i - c_i^j) \sum_{k \in I_{c_i^j}} \mu_k (y_k - y)}{\sum_{l=1}^M \mu_l} \quad (1.41)$$

De la même manière, on trouve:

$$\frac{\partial y}{\partial v_i^j} = \frac{-v_i^j (x_i - c_i^j)^2 \sum_{k \in I_{c_i^j}} \mu_k (y_k - y)}{\sum_{l=1}^M \mu_l} \quad (1.42)$$

La dérivée de la sortie par rapport aux paramètres des conséquences est donnée par:

$$\frac{\partial y}{\partial a_i^j} = \frac{\mu_j x_i}{\sum_{l=1}^M \mu_l} \quad (1.43)$$

## 1.11 INITIALISATION DES PARAMETRES DES SYSTEMES FLOUS

Nous avons vu que le système flou développé dans cette partie est un modèle paramétré, dont les paramètres sont ceux des fonctions d'appartenance associées aux prémisses et ceux des fonctions numériques associées aux conclusions. La procédure d'apprentissage dans un système flou consiste à trouver un ensemble de valeurs optimales des paramètres par un algorithme d'apprentissage. La qualité de chaque algorithme d'apprentissage dépend des valeurs initiales choisies pour les paramètres. Ainsi, une bonne initialisation conduit à de bonnes performances. Etant donné que les paramètres d'un système flou sont associés avec des fonctions d'appartenance, qui ont généralement un sens physique, il est possible de choisir de bonnes valeurs initiales.

Dans le cas d'un système flou de Sugeno, pour les fonctions d'appartenance des prémisses, les moyennes sont choisies sur la connaissance de l'univers de définition des variables d'entrée, et les supports sont choisis pour permettre un recouvrement satisfaisant des fonctions d'appartenance [JAN95]. En ce qui concerne les paramètres des conclusions, ils sont souvent fournis à partir d'une expertise ou de l'expérience d'un spécialiste.

## 1.12 LA LOGIQUE FLOUE POUR LA COMMANDE DES SYSTEMES

L'automatique fournit une large famille de méthodes pour synthétiser les lois de commande lorsque l'on dispose d'un modèle suffisamment précis du système. Ce modèle n'est pas toujours facile à obtenir car les systèmes comportent des non-linéarités et peuvent posséder des paramètres variables dans le temps. De plus, ils sont soumis à des perturbations.

Une alternative intéressante pour résoudre ce problème, consiste à étudier les méthodes utilisées par l'opérateur humain dans le contrôle des systèmes industriels complexes qui, dans certains cas, ont donné de meilleurs résultats que les techniques mathématiques élaborées. Dès lors, la logique floue, introduite par Zadeh en 1965, peut être vue comme un moyen de traduire cette expertise par un ensemble de règles à partir duquel est induite la commande à appliquer au système. Le premier contrôleur flou a été implémenté par l'équipe de Mamdani en 1974 [MAM77], [PED93], [TON94].

Un contrôleur flou classique est un simple moteur d'inférence floue chargé d'exécuter une base de règles floues [LEE90]. Cette base de règles relie directement des descriptions linguistiques de l'écart entre consigne et sortie réelle du système à des descriptions linguistiques des entrées du système. Les valeurs linguistiques qualifiant l'écart à la consigne sont modélisées par des ensembles flous. Les valeurs linguistiques décrivant les entrées du système sont également modélisées par des ensembles flous.

Les règles utilisées dans un système de commande floue peuvent être obtenues par apprentissage du contrôleur lui-même. Un ensemble de règles grossièrement déterminées est, en général, initialement fourni. Il sera affiné en utilisant un mécanisme d'adaptation. Le premier contrôleur flou apprenant lui-même à déterminer son fonctionnement a été proposé par Procyk et Mamdani sous le nom de Self-Organizing-Process-Controller (S.O.C.) [MAM77]. Ce contrôleur a été appliqué à la commande d'un robot manipulateur par Scharf et Mandic en 1985 [SCH85], et par Tanscheit et Scharf en 1988 [TAN88]. L'apprentissage se fait alors en ligne, les règles étant adaptées en cours de fonctionnement, et pouvant donc suivre les modifications du système à contrôler. L'apprentissage peut également se faire hors-ligne, le contrôleur n'étant effectivement utilisé que lorsque l'ajustement a conduit à une erreur tolérable. Les S.O.C. semblent néanmoins aujourd'hui supplantés par les contrôleurs à base de systèmes flous adaptatifs, qui tentent de combiner les capacités d'apprentissage des réseaux neuronaux aux capacités d'interprétabilité propres à la plupart des systèmes relevant de la logique floue.

Ainsi, actuellement un système flou peut manipuler non seulement des informations linguistiques fournies par un expert, mais il peut ajuster ses paramètres en utilisant un algorithme d'apprentissage. Il en résulte donc, que toutes les techniques de commande neuronales, sont directement applicables aux contrôleurs flous.

### 1.13 CONCLUSION

Dans la première partie de ce chapitre, nous avons présenté les notions de bases de la théorie de la logique floue. En effet, nous avons insisté sur les idées les plus utilisées en commande floue. Par la suite, nous avons introduit les systèmes flous, où une structure de base est donnée avec une description succincte de chacun de ses composants.

L'architecture d'un système flou est déterminée par une meilleure compréhension des ensembles flous et des opérateurs flous. Nous avons constaté qu'il n'existe pas un seul type de système flou, mais il y'en a plusieurs. Un utilisateur des systèmes flous doit décider sur la méthode de défuzzification, le type des fonctions d'appartenance, le type des règles floues, la méthode du raisonnement flou et la stratégie de défuzzification.

Les systèmes flous exigent la disponibilité d'une expertise. Par conséquent, les performances de ces derniers sont étroitement liées aux techniques d'acquisition de connaissances et la justesse des informations acquises.

Pour pallier à ces inconvénients, ont été développées des méthodes d'apprentissage utilisables dans le cas où aucune expertise n'est disponible. De ce fait, les systèmes flous sont devenus des systèmes dotés d'une capacité d'apprentissage. Ainsi, les systèmes flous ont supplanté les réseaux neuronaux. En effet, les systèmes flous peuvent incorporer non seulement des informations linguistiques mais, ils peuvent également ajuster leurs paramètres internes par des algorithmes d'apprentissage. En outre, l'adaptation d'un système flou peut être plus rapide que celle d'un réseau de neurones du fait que les paramètres d'un système flous peuvent être initialisés par de bonnes valeurs; car ils sont associés à des fonctions d'appartenance qui ont généralement un sens physique

Les systèmes flous et les réseaux neuronaux peuvent être utilisés pour résoudre le même type de problèmes. Les deux, sont des approximateurs généraux, ils peuvent approximer n'importe quelle fonction à partir de données numériques. Cependant, il faut noter que les deux approches souffrent du problème d'absence de méthodes systématiques pour choisir la structure optimale du système flou ou du réseau neuronal pour résoudre un problème particulier.

## CHAPITRE 2

# ***COMMANDE FLOUE ADAPTATIVE PAR INVERSION ITERATIVE***

### 2.1 INTRODUCTION

La commande adaptative des robots manipulateurs constitue à l'heure actuelle l'une des principales préoccupations des chercheurs en robotique. Cet intérêt est étroitement lié à la difficulté de commander un robot dont les performances désirées à l'exécution des tâches sont de plus en plus exigeantes. En effet, la présence des couplages entre les liaisons complique la conception de la commande. D'une part, l'utilisation des algorithmes de commande linéaires reste limitée et d'autre part, la synthèse des lois de commande adaptatives performantes nécessite le modèle de connaissance du robot manipulateur [GUE95], [SLO91], [BOI89]. Pour toutes ces raisons, la recherche de nouvelles techniques de commande s'avère nécessaire.

Le développement récent de la micro-informatique, et l'apparition des systèmes flous avec leur capacité d'apprentissage ont stimulé la commande adaptative par l'introduction de nouvelles architectures de commande.

L'apprentissage des systèmes non linéaires paramétrés, entre autres les systèmes flous, est effectué par l'algorithme de la rétropropagation. Pour ce dernier, l'erreur de sortie est rétropropagée à travers le système pour ajuster les poids internes afin de minimiser l'erreur de sortie [FRE92]. En effet, le même mécanisme d'apprentissage par adaptation des paramètres peut être utilisé pour inverser itérativement les systèmes non linéaires [HOS92]. Dans cette méthode, les erreurs à la sortie du système sont attribuées aux erreurs des signaux d'entrée, plutôt qu'aux erreurs des paramètres. Ainsi, la méthode d'inversion itérative des systèmes non linéaires procède par une recherche dans l'espace des entrées. Par contre, la méthode de la rétropropagation procède à travers une recherche dans l'espace des paramètres.

Dans ce chapitre, nous allons traiter la question relative à la commande floue adaptative des robots manipulateurs en utilisant la technique d'inversion itérative des systèmes non linéaires.

## 2.2 APPRENTISSAGE ET INVERSION DES SYSTEMES FLOUS

Dans ce chapitre, nous considérons le modèle flou de Sugeno d'ordre un. Ainsi, en notant les entrées du système flou par  $x_1, \dots, x_n$ , la sortie  $y$  du système flou est alors donnée par:

$$y(x) = \frac{\sum_{k=1}^M \mu_k y_k}{\sum_{k=1}^M \mu_k} \quad (2.1)$$

avec

$$\mu_k = \prod_{j=1}^n \mu_{F_j^k}(x_j) \quad (2.2)$$

$$y_k = a_0^k + a_1^k x_1 + \dots + a_n^k x_n \quad (2.3)$$

Pour une entrée donnée  $x(t)$ , le système flou est entraîné par un algorithme d'apprentissage afin de minimiser l'erreur entre la sortie désirée et la sortie du système flou. L'algorithme d'adaptation consiste à trouver le gradient de la fonction d'erreur par rapport aux paramètres, et d'ajuster ces derniers par une loi de la forme suivante:

$$\theta(t+1) = \theta(t) + p(t) \frac{\partial y}{\partial \theta} (y_r(t+1) - y(t+1)) \quad (2.4)$$

où  $\theta$  est le vecteur des paramètres ajustables du système flou,  $p(t)$  est le gain d'adaptation, et  $y_r$  est la sortie de référence ou désirée.

Par contre, l'inversion itérative du système flou génère le vecteur d'entrée,  $x$ , lequel donne une sortie, la plus proche que possible de la sortie désirée. La minimisation de l'erreur de sortie profite de l'avantage de dualité existante entre les paramètres et les entrées du système flou. L'algorithme d'adaptation est utilisé à nouveau pour obtenir le vecteur d'entrée désiré [HOS92]:

$$x^{(n+1)}(t) = x^{(n)}(t) + p(t) \frac{\partial y}{\partial x} (y_r(t+1) - y(t+1)) \quad (2.5)$$

## 2.3 COMMANDE PAR INVERSION ITERATIVE

L'utilisation des modèles inverses des systèmes présente un grand potentiel pour la commande [HUN92], [HUN91], [NAR90], [PSA88]. En effet, les modèles inverses des systèmes dynamiques ont une utilité immédiate dans la commande des systèmes dynamiques et particulièrement les systèmes non linéaires. Le problème majeur qui se pose dans l'identification inverse se manifeste quand l'inverse du système n'est pas bien défini. Pour remédier à ce problème, l'inverse du système est calculé par une technique itérative [HOS92]. Dans ce cas, une procédure

réursive est utilisée pour trouver l'inverse du système à chaque point de fonctionnement. Cette technique est particulièrement utile dans les systèmes singuliers qui admettent uniquement une inversion locale, et non sur l'espace globale de fonctionnement.

Considérons les systèmes dynamiques non linéaires décrits par une équation aux différences de la forme suivante:

$$y(t + 1) = f(y(t), y(t - 1), \dots, y(t - n), u(t), \dots, u(t - m)) \tag{2.6}$$

A l'instant  $t$ , l'objectif est de trouver une entrée  $u(t)$  pour laquelle la sortie du modèle  $y(t+1)$  est égale à la référence  $y_r(t+1)$ . Pour cela, le calcul de la commande  $u(t)$ , est obtenu par la méthode d'inversion itérative donnée par la relation suivante:

$$u^{(n+1)}(t) = u^{(n)}(t) + p(t) \frac{\partial y}{\partial u} (y_r(t + 1) - y(t + 1)) \tag{2.7}$$

La commande  $u(t-1)$  est utilisée comme valeur initiale à l'instant  $t$ .

La représentation par schéma bloc de cette procédure est donnée à la figure 2.1:

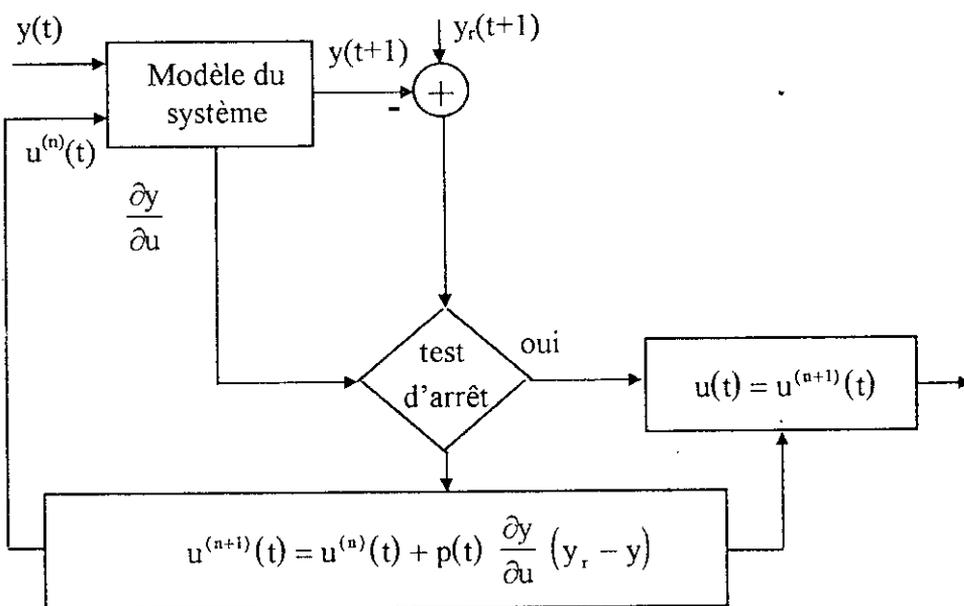


Figure 2.1. Procédure d'inversion itérative des systèmes non linéaires.

Dans le cas où le modèle dynamique du système est mal connu, la commande est calculée par la procédure d'inversion itérative en utilisant un modèle du système estimé en ligne.

## 2.4 COMMANDE ADAPTATIVE PAR INVERSION ITERATIVE DES ROBOTS MANIPULATEURS

Dans cette section, nous proposons une stratégie de commande adaptative des robots manipulateurs basée sur la méthode d'inversion itérative des systèmes flous. L'élaboration de la commande s'effectue en deux étapes. La première étape consiste à estimer les paramètres d'un modèle flou du robot manipulateur. Dans la seconde, la commande à appliquer à chaque articulation du robot est calculée itérativement.

### 2.4.1 Modélisation du robot

Modéliser un système, c'est chercher un modèle paramétré dont le comportement approche celui du système. En effet, la modélisation se compose de deux principales phases. La première phase consiste à rassembler les connaissances disponibles sur le comportement dynamique du système. Si les connaissances sur le système sont précises, le modèle peut être l'expression de lois physiques. Si ces connaissances sont rudimentaires, on est conduit à choisir des modèles de représentation du type boîte noire. Dans la seconde phase de la modélisation, aussi dite identification, les paramètres du modèle choisi sont estimés de façon à minimiser un critère.

La dynamique directe de chaque articulation du robot est représentée par un modèle flou de Sugeno d'ordre un. Ce choix est motivé par la facilité du calcul de la commande. Ainsi, le modèle de la  $i$ -ème articulation est écrit sous la forme:

$$\ddot{q}_i = f_i(q, \dot{q}, \tau_i; \theta_i) \quad (2.8)$$

Une règle floue de la base de connaissance de la fonction  $f_i(\cdot)$  a la forme suivante:

$$R_k : \text{Si } x_1 \text{ est } F_1^{k1} \text{ et } \dots \text{ et } x_j \text{ est } F_j^{kj} \text{ et } \dots \text{ et } x_n \text{ est } F_n^{kn} \text{ Alors } \ddot{q}_k = a_0^k + a_1^k x_1 + \dots + a_n^k x_n$$

où  $x = [q \ \dot{q} \ \tau_i]$  est le vecteur d'entrée. Alors que  $\tau_i$  est la commande exercée sur la  $i$ -ème articulation du robot. Le vecteur  $\theta_i$  regroupe tous les paramètres du système flou; c'est-à-dire, les paramètres des fonctions d'appartenance et ceux des conclusions.

La sortie finale du système flou est donnée par la relation:

$$\ddot{q}_i = \frac{\sum_{k=1}^{M_i} \mu_k \ddot{q}_k}{\sum_{k=1}^{M_i} \mu_k} \quad (2.9)$$

avec

$$\mu_k = \prod_{j=1}^n \mu_{F_j^k}(x_j) \quad (2.10)$$

alors que  $M_i$  est le nombre de règle du système flou  $f_i(\cdot)$ .

Après le choix du modèle de représentation, la seconde phase de la modélisation consiste à estimer les paramètres de ce modèle. Pour cela, nous considérons dans cette partie un algorithme d'apprentissage de type gradient modifié. Cet algorithme ajuste les paramètres  $\theta_i$  du système flous  $f_i(\cdot)$  afin de minimiser le carré de l'erreur instantanée entre l'accélération réelle du robot manipulateur et la sortie du modèle flou, définie par:

$$e_i(t) = \ddot{q}_i(t) - \ddot{\hat{q}}_i(t) \quad (2.11)$$

La loi d'adaptation est donnée par l'équations suivante:

$$\theta_i(t+1) = \theta_i(t) + p(t) \frac{\partial f_i(q, \dot{q}, \tau_i; \theta_i)}{\partial \theta_i} (\ddot{q}_i - \ddot{\hat{q}}_i) \quad (2.12)$$

Il est à noter que l'algorithme d'identification nécessite la mesure de l'accélération. Cette dernière est souvent évaluée par une différentiation de la vitesse ou par une double différentiation de la position, mais ceci n'est pas souhaitable dans le cas des mesures bruitées [SAA94].

#### 2.4.2 Calcul de la commande

La commande exercée sur la  $i$ -ème articulation du robot est calculée afin de minimiser l'erreur de poursuite. Pour cela, la méthode d'inversion itérative du système flou est utilisée pour trouver la commande nécessaire. La procédure d'optimisation est donnée par l'équation suivante:

$$\tau_i^{(n+1)}(t) = \tau_i^{(n)}(t) + p(t) J_{\tau_i}(t) (\tau_i'(t) - f_i(q, \dot{q}, \tau_i^{(n)}; \theta_i)) \quad (2.13)$$

avec

$$J_{\tau_i} = \frac{\partial f_i(q, \dot{q}, \tau_i; \theta_i)}{\partial \tau_i} \quad (2.14)$$

$$p(t) = \frac{1}{1 + J_{\tau_i}^2}$$

$$\tau_i'(t) = \ddot{q}_{di}(t) + k_v(\dot{q}_{di} - \dot{q}_i) + k_p(q_{di} - q_i)$$

alors que  $k_v$  et  $k_p$  sont des constantes positives.

Le gradient  $J_{\tau_i}$  est calculé analytiquement à partir de l'expression du système flou  $f_i(\cdot)$  en utilisant le mécanisme de la rétropropagation. Il est donné par l'expression suivante:

$$J_{\tau_i} = \sum_{k=1}^{M_i} \frac{\partial \ddot{q}_i}{\partial \mu_k} \frac{\partial \mu_k}{\partial \mu_{F_n^{kn}}(\tau_i)} \frac{\partial \mu_{F_n^{kn}}(\tau_i)}{\partial \tau_i} + \sum_{k=1}^{M_i} \frac{\partial \ddot{q}_i}{\partial \ddot{q}_k} \frac{\partial \ddot{q}_k}{\partial \tau_i} \quad (2.15)$$

Dans le cas des fonctions d'appartenance gaussienne, cette expression devient:

$$J_{\tau_i} = \frac{\sum_{k=1}^{M_i} \mu_k (\ddot{q}_k - \ddot{q}_i) v_n^{kn^2} (\tau_i - c_n^{kn}) + \sum_{k=1}^{M_i} \mu_k a_n^k}{\sum_{k=1}^{M_i} \mu_k} \quad (2.16)$$

Le calcul de ce gradient nécessite évidemment un temps de calcul non négligeable. Etant donné qu'il est bien connu que la matrice d'inertie d'un robot quelconque est définie positive. Il en résulte que la dérivée de l'accélération de chaque liaison par rapport à son entrée de commande est positive. Ainsi, si nous assumons que le modèle flou identifié est suffisamment bon, nous pouvons utiliser cette propriété pour avoir une expression plus simple de la recherche de la valeur de commande. Nous avons donc:

$$\operatorname{sgn}\left(\frac{\partial \ddot{q}_i}{\partial \tau_i}\right) = \operatorname{sgn}\left(\frac{\partial \ddot{q}_i}{\partial \tau_i}\right) = 1 \quad (2.17)$$

C'est ainsi que la relation (2.13) devient:

$$\tau_i^{(n+1)}(t) = \tau_i^{(n)}(t) + p(t) \left( \tau_i'(t) - f_i(q, \dot{q}, \tau_i^{(n)}; \theta_i) \right) \quad (2.18)$$

La procédure de recherche à l'instant  $t$  est initialisée par la valeur de la commande à l'instant précédent. Pour arrêter cette procédure, nous pouvons utiliser l'une des conditions suivantes:

- l'erreur  $e_i(t)$  est inférieure en valeur absolue d'une valeur de tolérance.
- le gradient  $J_{\tau}$  est inférieur à un seuil.
- le nombre d'itération atteint un nombre maximal d'itération permis, c'est-à-dire, le temps de calcul est limité.

La figure 2.2 illustre schématiquement le principe de base de cette stratégie de commande:

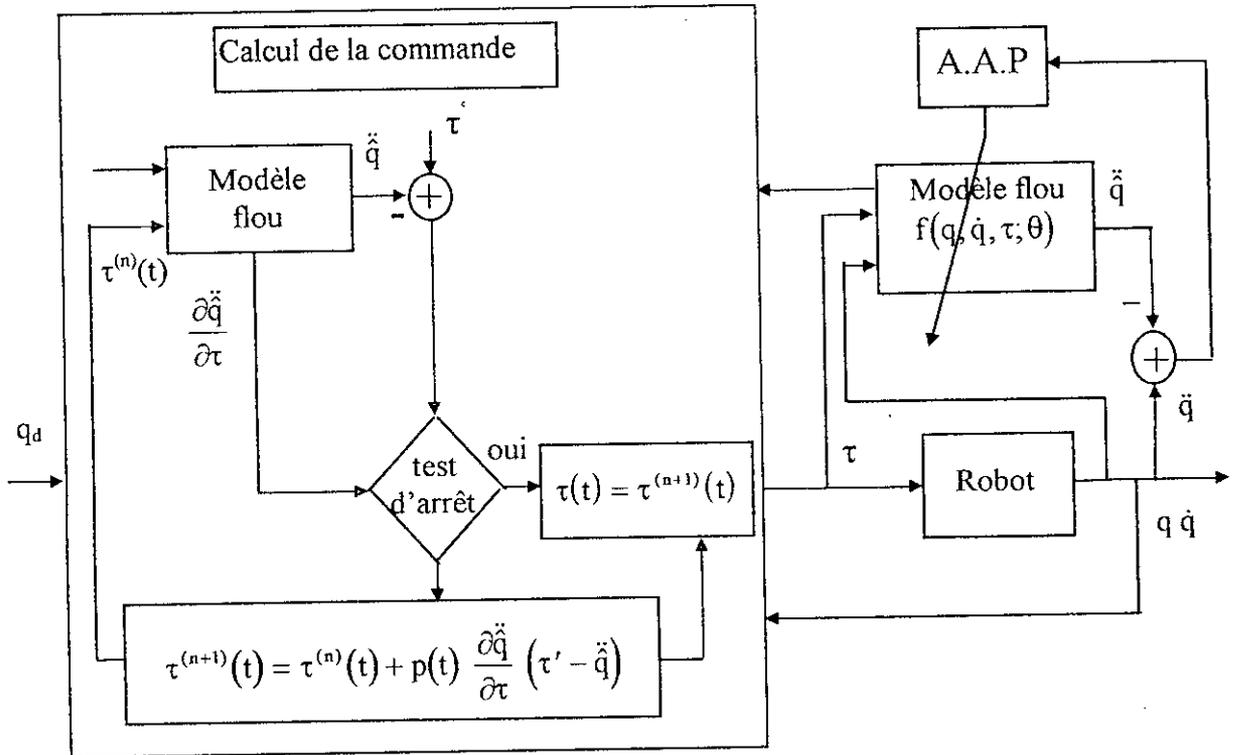


Figure 2.2. Principe de la commande adaptative des robots manipulateurs par la méthode d'inversion itérative.

### 2.4.3 Résultats de simulation

Nous proposons dans cette section l'application de cette stratégie de commande à un robot manipulateur de type PORTOS à trois degrés de liberté. Ce robot est constitué de trois articulations, il est caractérisé par deux articulations translationnelles  $q_1$  et  $q_3$  et une articulation rotationnelle  $q_2$  [GUE95], [ANNEXE]. Dans ce travail, nous avons opté pour une trajectoire de référence cycloïdale donnée par l'expression (2.19). Cette dernière assure la continuité en position, en vitesse et en accélération, et est donnée par l'expression suivante [SER89]:

$$\begin{cases} q(t) = q_i + (q_f - q_i) \left( \frac{t}{2\pi T} - \frac{1}{2\pi} \sin\left(\frac{2\pi t}{T}\right) \right), & t \leq T \\ q_f, & t > T \end{cases} \quad (2.19)$$

où  $q_i$  et  $q_f$  sont respectivement la valeur initiale et la valeur finale de la trajectoire, et  $T$  la durée de la trajectoire.

Le long de ce travail, nous choisissons les paramètres de synthèse suivants: Le pas de simulation est pris égal à 5 ms, la durée de la trajectoire est prise égale à 2.5 s, le robot PORTOS se déplace du repos de la position initiale  $q(0) = [0 \ 0 \ 0]$  à la configuration finale  $q(2.5) = [1 \ 1.5 \ 0.7]$ .

La simulation du robot est effectuée en modélisant chacune des liaisons du robot par un système flou. Le tableau 2.1 résume la structure des systèmes flous utilisés.

Articulation	1	2	3
Entrées du système flou	$\dot{q}_1, q_1, \tau_1$	$\dot{q}_2, q_2, \dot{q}_3, \tau_2$	$\dot{q}_2, q_3, \dot{q}_3, \tau_3$
Nombre de règles floues	8	16	16

**Tableau 2.1.** Structure des systèmes flous utilisés.

Les coefficients  $k_v$  et  $k_p$  sont pris respectivement égal à 40 et 400 pour toutes les articulations.

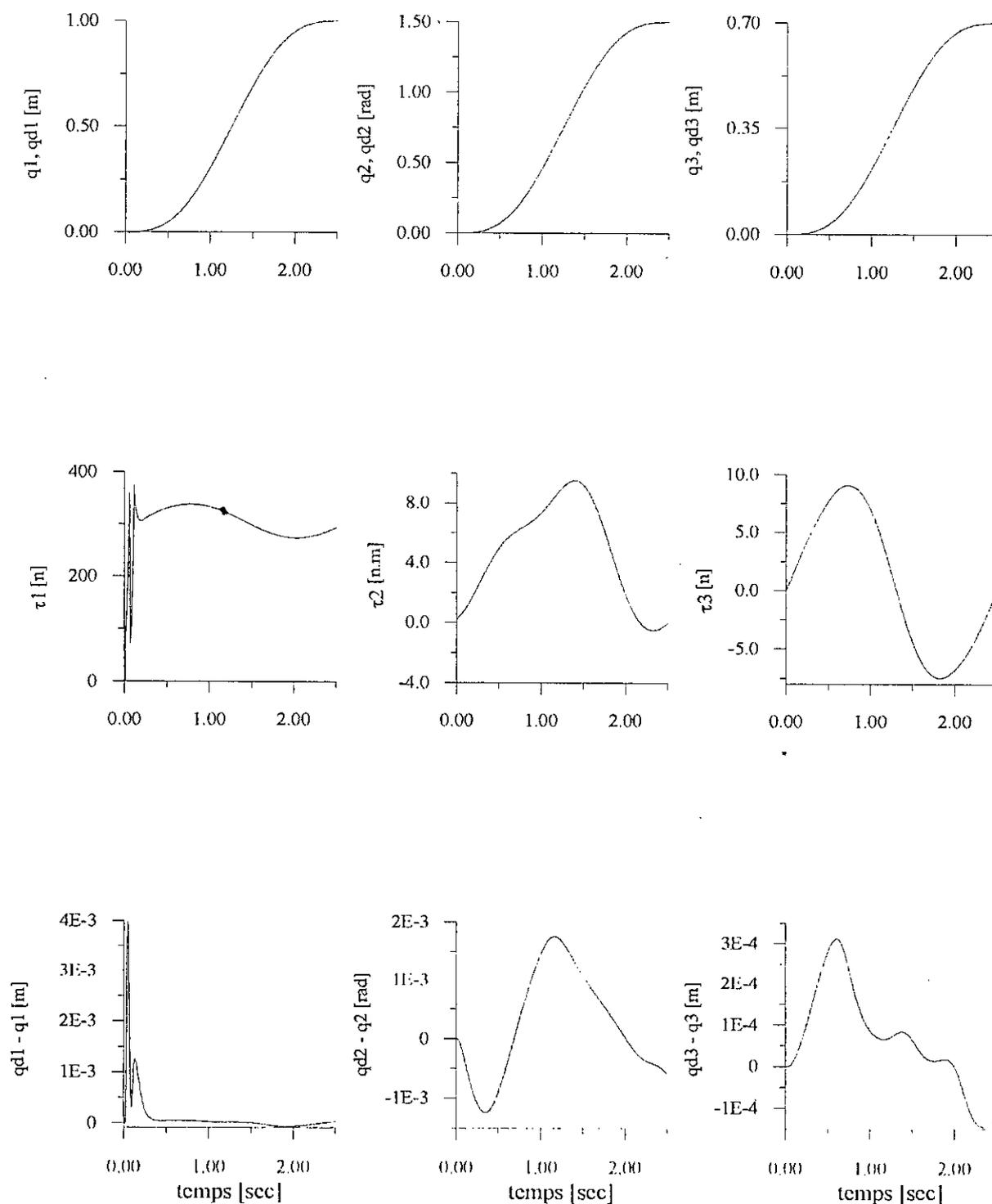
En premier lieu nous avons simulé en utilisant la commande donnée par la relation (2.13). Les réponses du robot manipulateur sont illustrées par la figure 2.3. Cette dernière représente l'évolution de la position, de la commande et de l'erreur de poursuite. Nous remarquons que les erreurs de poursuite sont très faibles, les grandeurs de commandes de la deuxième et de la troisième articulation sont lisses, mais la grandeur de commande de la première articulation présente des oscillations au début.

Pour tester les performances de ce schéma de commande vis-à-vis des variations paramétriques, une variation brusque de 50% de tous les paramètres est appliquée à l'instant 1.5 s. Les réponses obtenues sont illustrées par la figure 2.4. Ces réponses montrent que cette perturbation paramétrique n'a pas affecté les performances de poursuite. Néanmoins, la grandeur de commande de la première articulation a oscillé au début de l'application de la perturbation paramétrique.

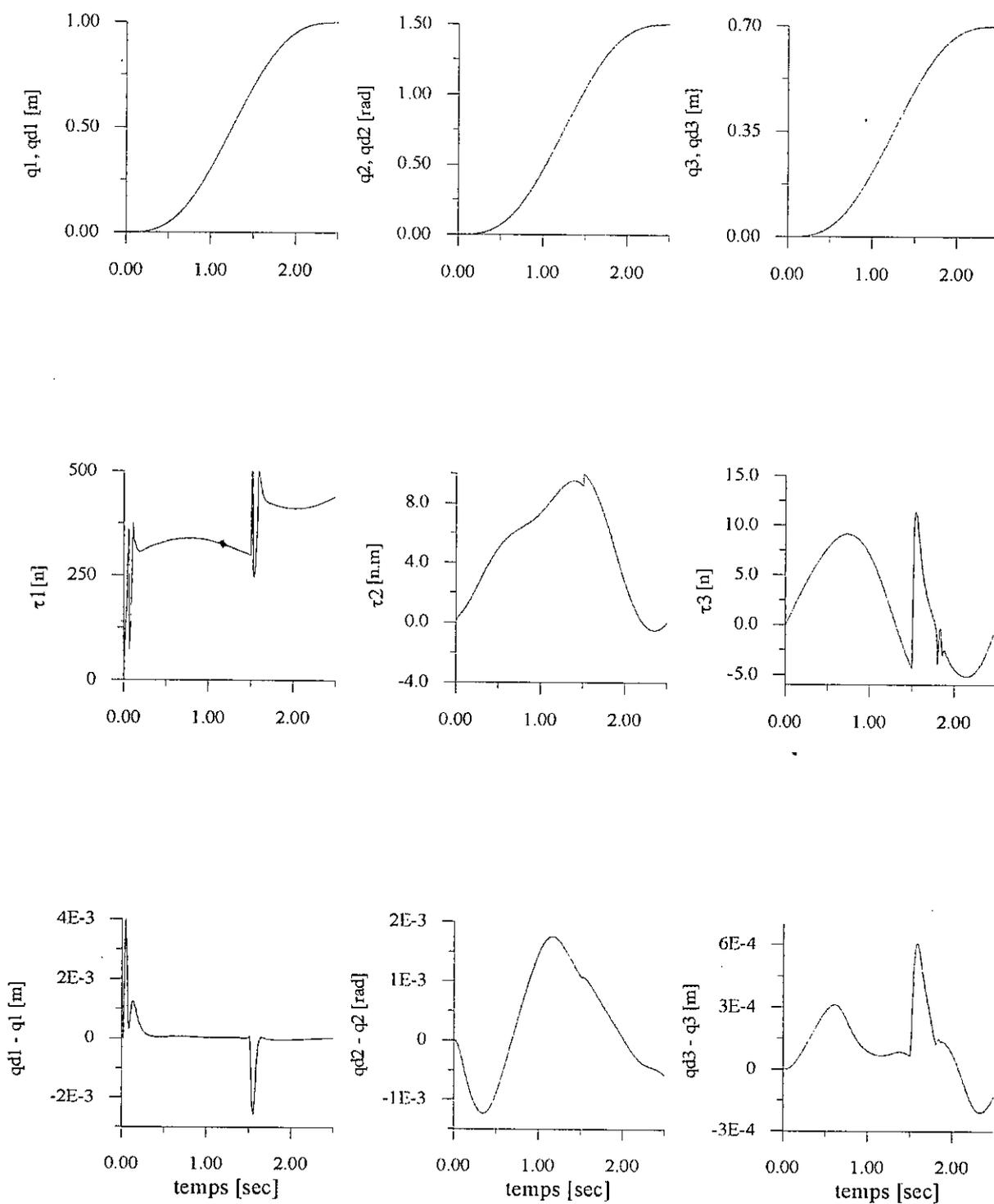
En second lieu, nous avons utilisé la loi de commande donnée par l'équation (2.18); c'est-à-dire, nous avons considérée le signe du gradient. Les réponses du robot manipulateur sont illustrées par la figure 2.5. Nous remarquons que les erreurs de poursuite sont très faibles. Néanmoins, les grandeurs de commandes de la première et de la deuxième articulation présentent des oscillations sur leur grandeur de commande.

Les résultats de simulation avec variations paramétriques sont donnés par la figure 2.6. Ces réponses montrent la robustesse de cette approche par rapport à cette perturbation paramétrique.

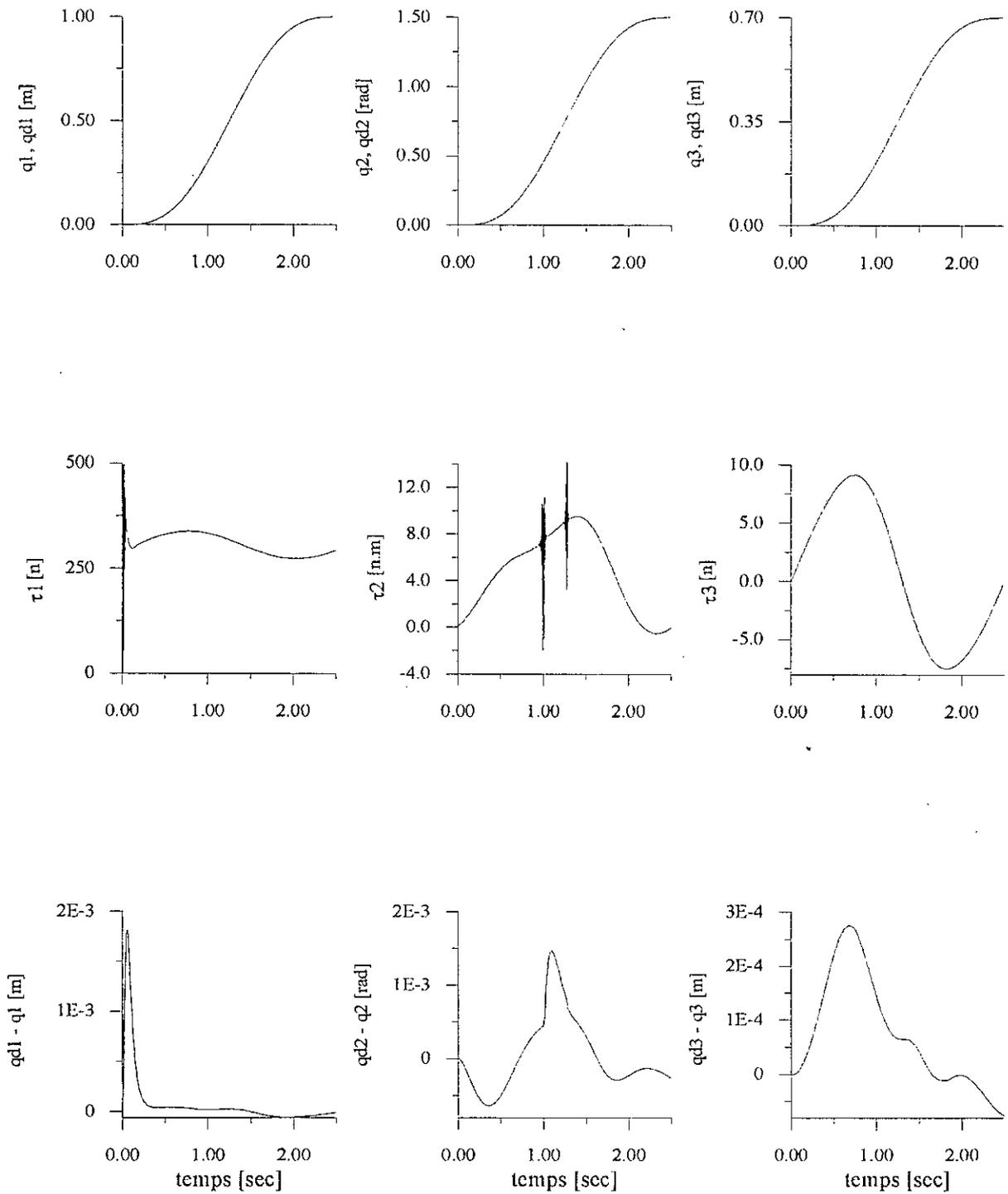
Nous remarquons que la première variante donne des signaux de commande plus lisses par rapport à ceux de la deuxième.



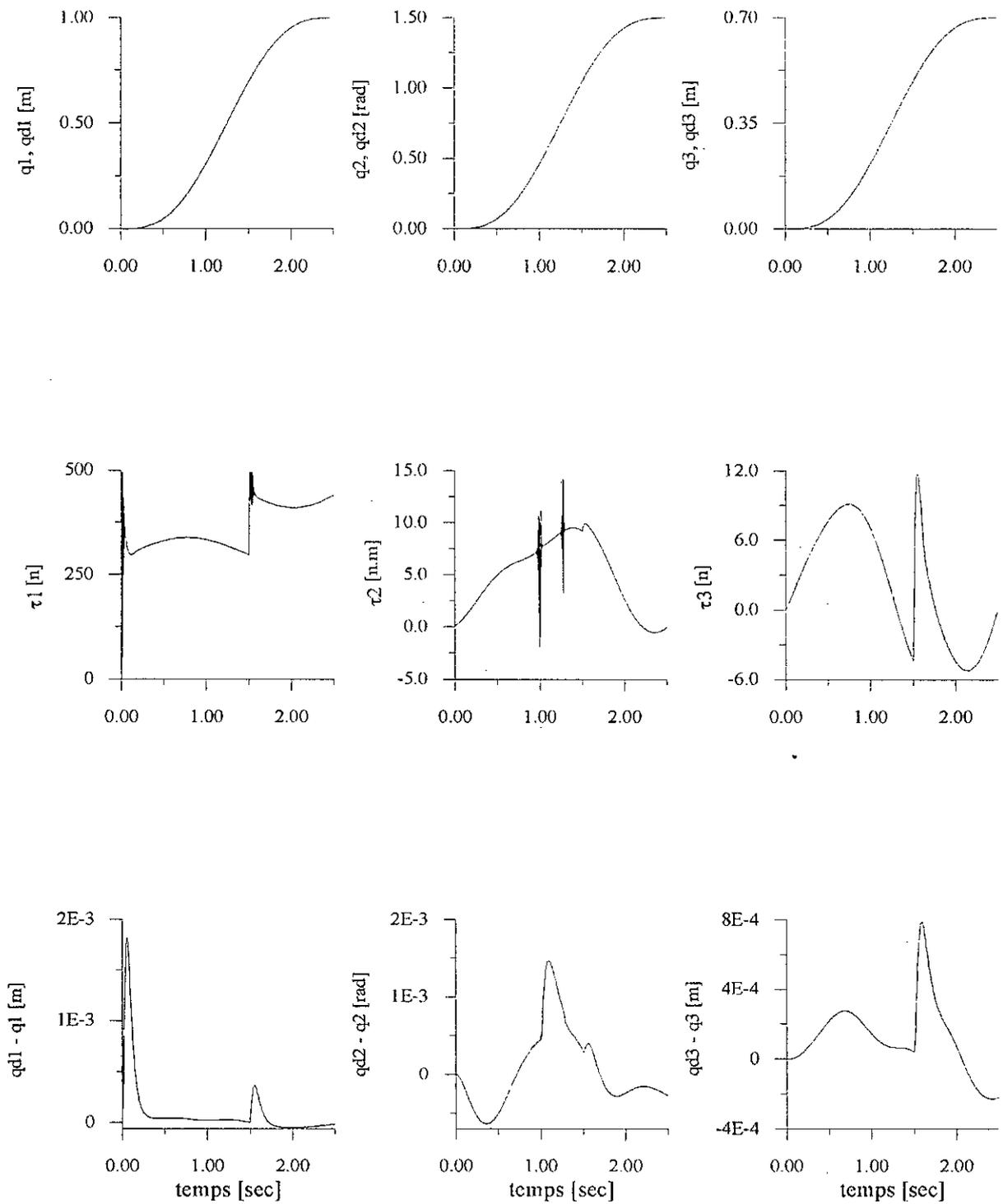
**Figure 2.3.** Résultats de simulation sans variations paramétriques pour la commande floue adaptative par inversion itérative.



**Figure 2.4.** Résultats de simulation avec variations paramétriques pour la commande floue adaptative par inversion itérative.



**Figure 2.5.** Résultats de simulation sans variations paramétriques pour la commande floue adaptative par inversion itérative utilisant le signe du gradient.



**Figure 2.6.** Résultats de simulation avec variations paramétriques pour la commande floue adaptative par inversion itérative utilisant le signe du gradient.

## 2.5 CONCLUSION

Dans ce chapitre, nous nous sommes intéressés à l'application de la technique d'inversion itérative des systèmes flous pour la commande adaptative des bras manipulateurs.

En premier lieu, nous avons présenté la méthode d'inversion itérative des systèmes non linéaires et plus particulièrement celle des systèmes flous. Nous avons remarqué que cette technique est une autre formulation du problème d'entraînement des systèmes non linéaires. En effet, au lieu d'attribuer les erreurs de sortie aux paramètres internes, on les attribue aux signaux d'entrée.

En second lieu, Nous avons développé la stratégie de commande adaptative des robots par la méthode d'inversion itérative. Cette stratégie procède en deux étapes. Dans la première étape, on identifie le modèle du robot manipulateur par les systèmes flous. Cependant, la qualité de l'identification dépend du modèle de représentation choisi et de l'efficacité de l'algorithme d'adaptation utilisé. Dans la seconde, la commande se calcule par la procédure d'inversion itérative en utilisant le modèle flou identifié.

Il est également présenté une variante de cette approche de commande où on utilise la positivité de la matrice d'inertie pour éviter le problème du calcul de la dérivée de la sortie du modèle flou par rapport à la commande.

L'avantage principal de cette stratégie de commande est qu'elle permet de commander les systèmes fortement non linéaires; même ceux qui possèdent des singularités, car l'inversion est effectuée à chaque point de fonctionnement. En outre, elle permet d'éviter les problèmes de modélisation.

L'inconvénient majeur de cette stratégie réside dans son utilisation d'une méthode itérative pour le calcul de l'inverse. Cependant, la procédure d'inversion peut prendre beaucoup de temps afin d'arriver à une solution optimale. Ainsi, pour remédier à ce problème, on impose le nombre maximal d'itération.

La simulation effectuée sur le robot manipulateur PORTOS a montré des performances de poursuite remarquables même en présence des variations paramétriques. En effet, nous avons remarqué que les erreurs de poursuite sont très faibles. En ce qui concerne les grandeurs de commande, nous avons enregistré des petites oscillations sur la grandeur de commande de la première articulation; et qui apparaissent au début et à l'instant de l'application de la perturbation paramétrique, ainsi que des oscillations sur la commande de la deuxième articulation dans le cas d'utilisation du signe du gradient. Néanmoins, cette méthode de commande souffre également de l'inconvénient de l'utilisation de l'accélération du robot qui est difficile à estimer dans les milieux bruités.

## CHAPITRE 3

# *COMMANDE FLOUE ADAPTATIVE INDIRECTE*

### 3.1 INTRODUCTION

Dans ce chapitre, nous nous intéressons au schéma de commande indirect. Ce dernier repose sur l'idée d'apprentissage du système à commander par le biais de l'identification des paramètres d'un modèle de ce système. Le fonctionnement de la commande adaptative indirecte est comme suit [AST89], [SLO91]: à chaque instant, l'estimateur envoie un ensemble des estimés des paramètres du système au contrôleur, ensuite le calculateur évalue ceux du contrôleur en prenant les estimés pour les vrais paramètres du système. Cette idée est souvent appelée le principe de l'équivalence certaine. La méthode indirecte nécessite évidemment un volume de calcul important, mais son domaine d'application est très large [AST89], [GOO84], [SLO91].

La première étape de toute synthèse de commande est le choix d'un modèle paramétré du système commandé [RIV95]. Ce modèle peut être un modèle de connaissance découlant d'une analyse physique des phénomènes mis en jeu dans le système. Une seconde démarche consiste à construire un modèle de représentation de type boîte noire. Généralement, les paramètres de ces modèles n'ont pas de signification physique. Les systèmes flous, ensembles de règles, forment une famille de fonctions non linéaires, qui permet de construire, par apprentissage, une très large classe de modèles et de correcteurs [JAN93], [JAN95], [KOS92], [WAN95].

Dans ce chapitre, nous allons présenter et développer des techniques de commande adaptatives indirectes utilisant les systèmes flous. En premier lieu, nous présentons deux méthodes de commande adaptatives. La première est basée sur l'identification inverse. La seconde est basée sur l'identification directe. En second lieu, deux autres approches de commande floues adaptatives indirectes sont proposées. Ces deux dernières utilisent la théorie de Lyapunov pour l'obtention des lois d'adaptation. Notons que pour ces deux méthodes la mesure de l'accélération n'est pas nécessaire.

## 3.2 COMMANDE FLOUE ADAPTATIVE BASEE SUR L'IDENTIFICATION INVERSE

### 3.2.1 Modélisation du robot

Compte tenu de l'équation représentative de la dynamique inverse de tout robot [ANNEXE], [PAU81], [SLO91], [VIB87], le modèle choisi pour la  $i$ -ème articulation est de la forme suivante:

$$\hat{\tau}_i = f_i(q, \dot{q}, \ddot{q}_i; \theta_i) \quad (3.1)$$

Cette relation constitue seulement un modèle pour la commande. Ce modèle respecte cependant la structure essentielle des robots issue des lois de la mécanique. La fonction  $f_i(\cdot)$  est un système flou de Sugeno d'ordre un avec des descriptions linguistiques de la forme suivante:

$$R_k : \text{Si } x_1 \text{ est } F_1^{k1} \text{ et } \dots \text{ et } x_n \text{ est } F_n^{kn} \text{ Alors } \hat{\tau}_k = a_0^k + a_1^k x_1 + \dots + a_n^k x_n + b^k \ddot{q}_i$$

Avec  $x = [q \ \dot{q}]$  est le vecteur d'entrée, et  $\ddot{q}_i$  est l'accélération réelle de la  $i$ -ème articulation du robot.

La sortie finale du système flou  $f_i(\cdot)$  est donnée par la relation suivante:

$$\hat{\tau}_i = \frac{\sum_{k=1}^{M_i} \mu_k \hat{\tau}_k}{\sum_{k=1}^{M_i} \mu_k} \quad (3.2)$$

avec

$$\mu_k = \prod_{j=1}^n \mu_{F_j^{kj}}(x_j) \quad (3.3)$$

alors que  $M_i$  est le nombre de règles floues.

Les paramètres internes  $\theta_i$  du système flou  $f_i(\cdot)$ ; c'est-à-dire, les paramètres des fonctions d'appartenance et ceux des conclusions, sont calculés afin de minimiser l'erreur instantanée à l'entrée du robot, définie par:

$$e_i(t) = \tau_i(t) - f_i(q, \dot{q}, \ddot{q}_i; \theta_i) \quad (3.4)$$

L'adaptation des paramètres est effectuée par un algorithme de type gradient modifié. Celui-ci est donné par la relation suivante:

$$\theta_i(t+1) = \theta_i(t) + p(t) J_{\theta_i}(t) e_i(t+1) \quad (3.5)$$

où

$$J_{\theta_i} = \frac{\partial f_i(q, \dot{q}, \ddot{q}_i; \theta_i)}{\partial \theta_i}$$

et  $p(t)$  est le gain d'adaptation.

Il apparaît que l'algorithme d'adaptation nécessite la mesure de l'accélération du robot manipulateur. Cette dernière est calculée numériquement. Notons que cette approximation peut être néfaste dans le cas des mesures bruitées.

### 3.2.2 Calcul de la commande

Le but est de concevoir une commande de telle façon que le robot suive une trajectoire de référence. Cette loi de commande est calculée en utilisant le modèle flou de la dynamique inverse du robot et la méthode de commande dite du couple calculé. Ainsi, le contrôleur est défini comme un système flou pour lequel chaque règle floue correspond à celle similaire du modèle flou. Par conséquent, une règle du contrôleur flou proposé est de la forme:

$$R_k : \text{Si } \langle \text{prémisse} \rangle \text{ Alors } \tau_k = a_0^k + a_1^k x_1 + \dots + a_n^k x_n + b^k \tau'_i$$

avec

$$\tau'_i = \ddot{q}_{di} + k_v(\dot{q}_{di} - \dot{q}_i) + k_p(q_{di} - q_i)$$

alors que  $k_v$  et  $k_p$  sont des constantes positives, calculés de sorte que le polynôme  $p^2+k_v p+k_p$  soit stable.

La commande totale appliquée à la  $i$ -ème articulation est donnée par la relation suivante:

$$\tau_i = f_i(q, \dot{q}, \tau'_i; \theta_i) \quad (3.6)$$

soit:

$$\tau_i = \frac{\sum_{k=1}^{M_i} \mu_k \tau_k}{\sum_{k=1}^{M_i} \mu_k} \quad (3.7)$$

La loi d'adaptation de cette stratégie de commande est donnée par la figure 3.1:

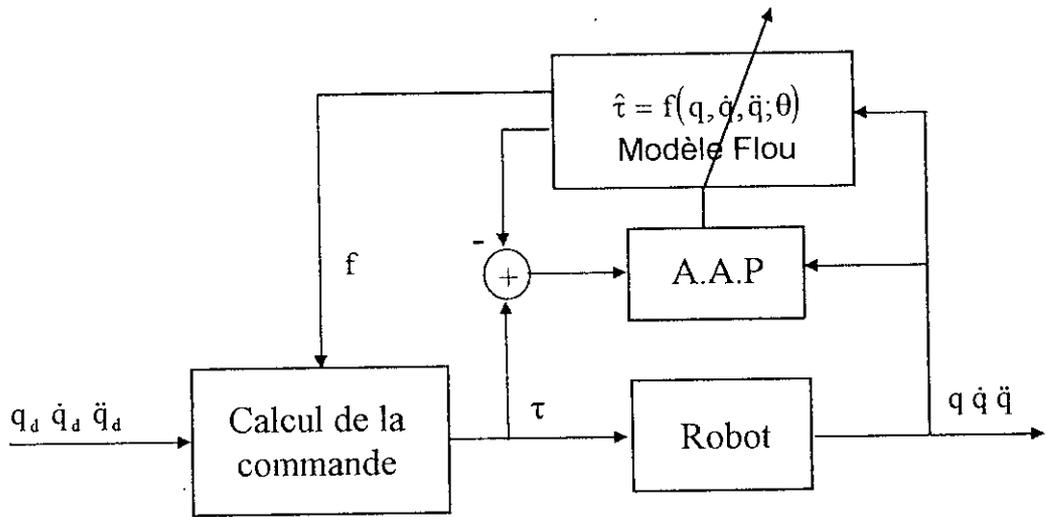


Figure 3.1. Structure de la commande floue adaptative basée sur l'identification inverse.

### 3.2.3 Résultats de simulation

Dans cette section, nous présentons les résultats de la simulation effectuée sur le robot manipulateur PORTOS [ANNEXE].

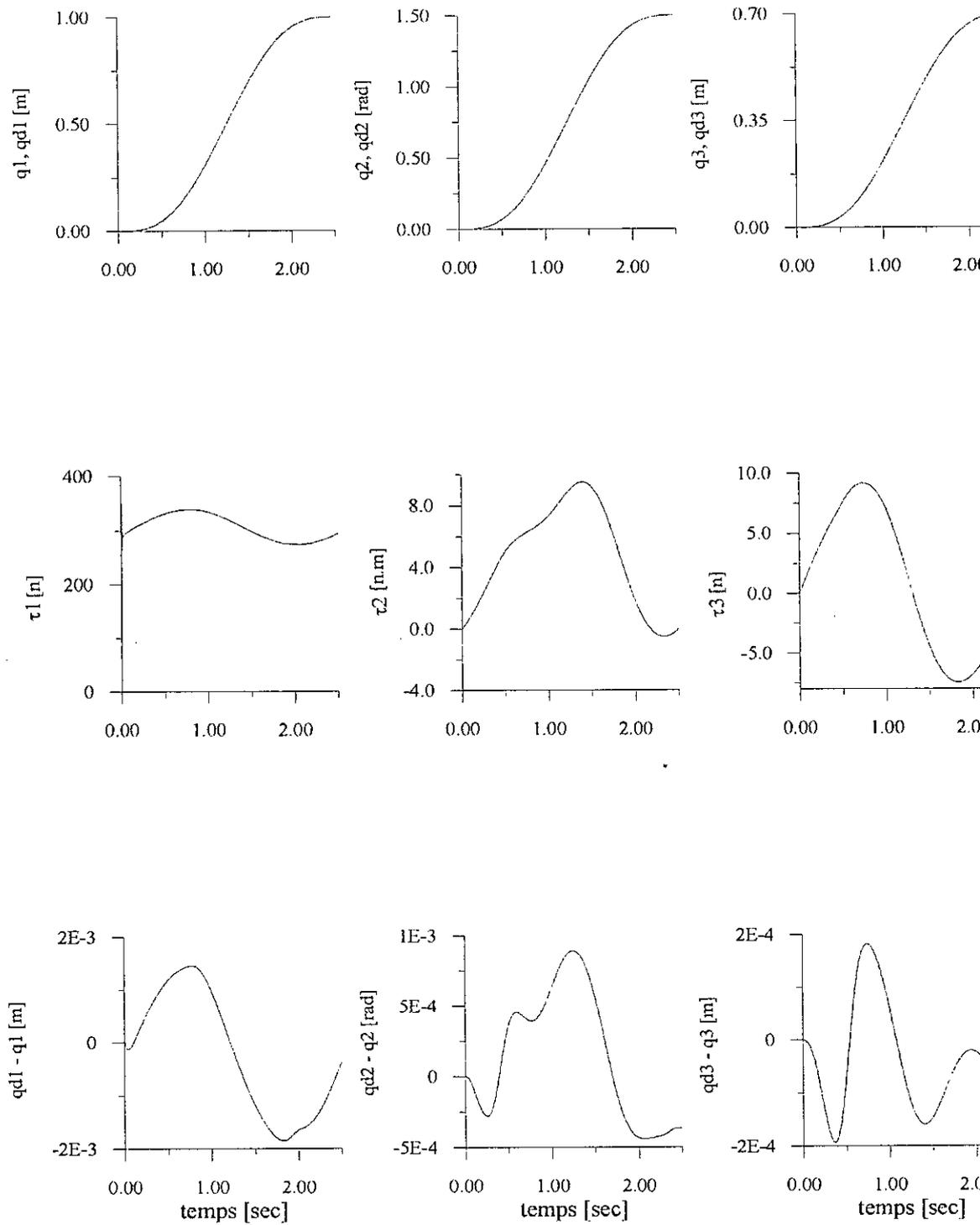
Le modèle de représentation de chaque articulation est constitué d'un système flou de Sugeno d'ordre un dont la structure est résumée par le tableau suivant:

Articulation	1	2	3
Entrées du système flou $f$	$\dot{q}_1, q_1$	$\dot{q}_2, q_3, \dot{q}_3$	$\dot{q}_2, q_3, \dot{q}_3$
Nombre de règles floues de $f$	4	8	8

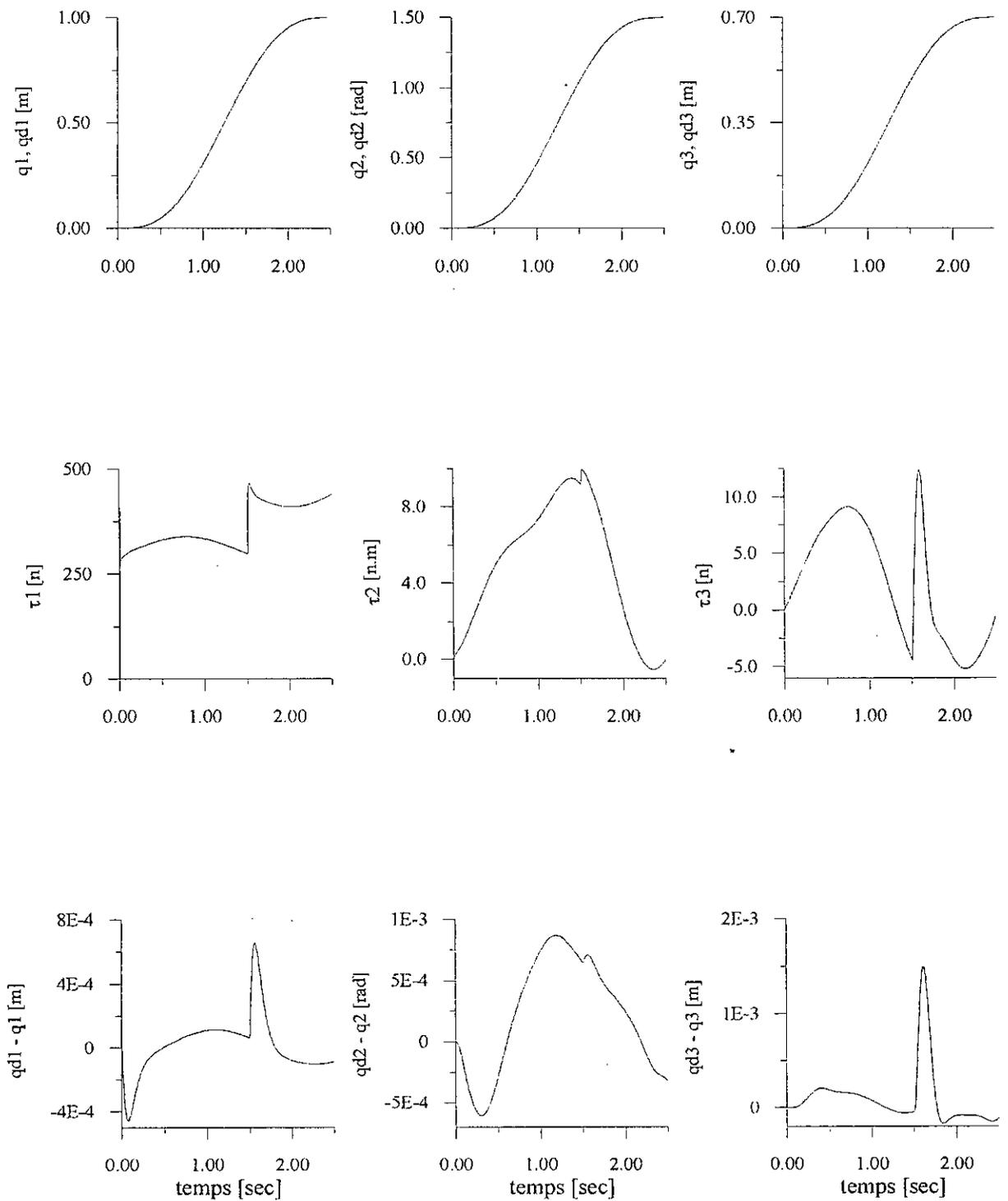
Tableau 2.1: Structure des systèmes flous utilisés.

Les paramètres  $k_v$  et  $k_p$ , sont respectivement choisis égal à 40 et 400 pour toutes les liaisons. Les résultats de simulation obtenus sont montrés à la figure 3.2. Cette figure illustre l'évolution de la position, la commande et l'erreur de poursuite de chaque articulation. Nous remarquons que les efforts de commande sont lisses, et que les erreurs de poursuite sont faibles.

Pour démontrer la capacité d'adaptation du schéma de commande lors de l'exécution des mouvements désirés face aux variations paramétriques, nous avons effectué une variation de 50% de tous les paramètres du robot à l'instant 1.5s. Les réponses obtenues sont représentées sur la figure 3.3. Nous constatons que cette perturbation due aux variations paramétriques n'a pas affecté les performances de poursuite.



**Figure 3.2.** Résultats de simulation sans variations paramétriques pour la commande floue adaptative directe basée sur l'identification inverse.



**Figure 3.3.** Résultats de simulation avec variations paramétriques pour la commande floue adaptative indirecte basée sur l'identification inverse.

### 3.3 COMMANDE FLOUE ADAPTATIVE BASEE SUR L'IDENTIFICATION DIRECTE

Dans l'approche exposée précédemment, nous nous sommes intéressés à une méthode de commande adaptative qui minimise l'erreur à l'entrée du robot manipulateur. Dans ce qui suit, est proposée une structure de commande adaptative basée sur l'identification de la dynamique directe du robot manipulateur et sur la méthode de commande dite du couple calculé.

#### 3.3.1 Modélisation du robot

La dynamique directe de chaque articulation du robot est représentée par un modèle flou de type Sugeno. Ce choix est motivé par la facilité de la détermination de la commande. Le modèle de la  $i$ -ème articulation est donc écrit sous la forme suivante [LAB97a], [LAB97b], [LAB98]:

$$\ddot{q}_i = f_i(q, \dot{q}, \tau_i; \theta_i) \quad (3.8)$$

Le système flou  $f_i(\cdot)$  est constitué d'un ensemble de règles floues de la forme suivante:

$$R_k : \text{Si } x_1 \text{ est } F_1^{k1} \text{ et } \dots \text{ et } x_n \text{ est } F_n^{kn} \quad \text{Alors} \quad \ddot{q}_k = a_0^k + a_1^k x_1 + \dots + a_n^k x_n + b^k \tau_i$$

avec  $x = [q \ \dot{q}]$  est le vecteur d'entrée,  $\tau_i$  est la commande exercée sur la  $i$ -ème articulation du robot, et  $\theta_i$  est un vecteur regroupant les paramètres des prémisses et ceux des conclusions.

La sortie finale du système flou est donnée par la relation suivante:

$$\ddot{q}_i = \frac{\sum_{k=1}^{M_i} \mu_k \ddot{q}_k}{\sum_{k=1}^{M_i} \mu_k} \quad (3.9)$$

avec

$$\mu_k = \prod_{j=1}^n \mu_{F_j^{kj}}(x_j) \quad (3.10)$$

où  $M_i$  est le nombre de règles du système flou  $f_i(\cdot)$ .

L'estimation des paramètres inconnus  $\theta_i$  du système flou se fait récursivement à l'aide d'un algorithme d'apprentissage du type gradient modifié. Dans cet algorithme, les paramètres  $\theta_i$  sont ajustés afin de minimiser l'erreur instantanée à la sortie entre l'accélération réelle et la sortie du modèle flou:

$$e_i(t) = \ddot{q}_i(t) - \ddot{\tilde{q}}_i(t) \quad (3.11)$$

La loi d'adaptation utilisée est donnée par la relation suivante:

$$\theta_i(t+1) = \theta_i(t) + p(t) J_{\theta_i}(t) e_i(t+1) \quad (3.12)$$

où

$$J_{\theta_i} = \frac{\partial f_i(q, \dot{q}, \tau_i; \theta_i)}{\partial \theta_i}$$

et  $p(t)$  est le gain d'estimation paramétrique.

### 3.3.2 Calcul de la commande

Le contrôleur est un système flou, défini de sorte que les antécédents des règles du modèle flou sont utilisés à nouveau pour générer la commande. Ainsi, une règle floue du contrôleur proposé est donnée par [LAB98]:

$$R_k : \text{Si } \langle \text{prémisse} \rangle \text{ Alors } \text{num}(\tau_i) = \tau'_i - y'_k \text{ et } \text{den}(\tau_i) = b^k$$

avec

$$\tau'_i = \ddot{q}_{di} + k_v(\dot{q}_{di} - \dot{q}_i) + k_p(q_{di} - q_i)$$

$$y'_k = a_0^k + a_1^k x_1 + \dots + a_n^k x_n$$

alors que  $k_v$  et  $k_p$  sont des constantes positives. Tandis que,  $\text{num}(\cdot)$  et  $\text{den}(\cdot)$  sont respectivement le numérateur et le dénominateur de la commande  $\tau_i$ .

La commande totale appliquée à la  $i$ -ème articulation est donnée par la relation suivante:

$$\tau_i = \frac{\text{num}(\tau_i)}{\text{den}(\tau_i)} \quad (3.13)$$

avec

$$\text{num}(\tau_i) = \frac{\sum_{k=1}^{M_i} \mu_k (\tau'_i - y'_k)}{\sum_{k=1}^{M_i} \mu_k} \quad (3.14)$$

et

$$\text{den}(\tau_i) = \frac{\sum_{k=1}^{M_i} \mu_k b^k}{\sum_{k=1}^{M_i} \mu_k} \quad (3.15)$$

d'où il vient:

$$\tau_i = \frac{\sum_{k=1}^{M_i} \mu_k (\tau'_i - y'_k)}{\sum_{k=1}^{M_i} \mu_k b^k} \tag{3.16}$$

La structure de commande adoptée est schématisée par la figure 3.4:

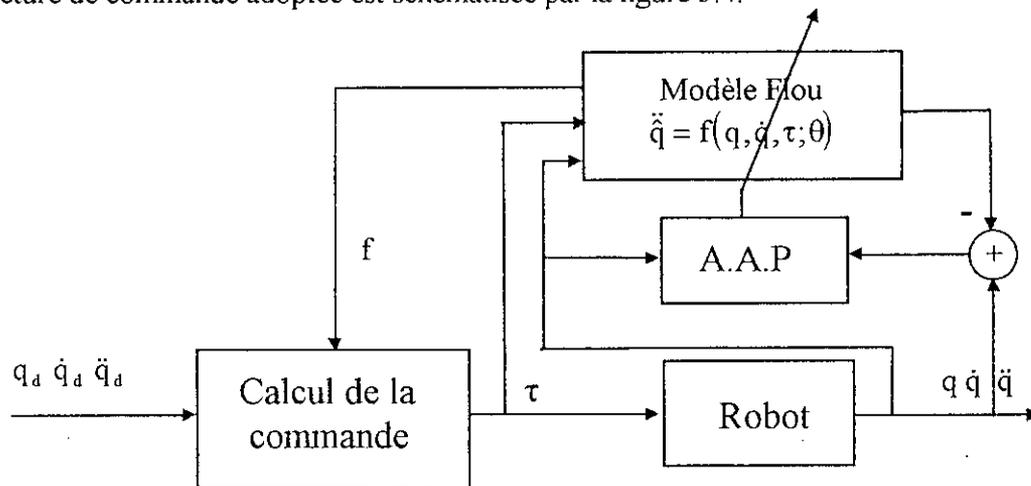


Figure 3.4. Principe de la commande floue adaptative basée sur l'identification directe.

### 3.3.3 Résultats de simulation

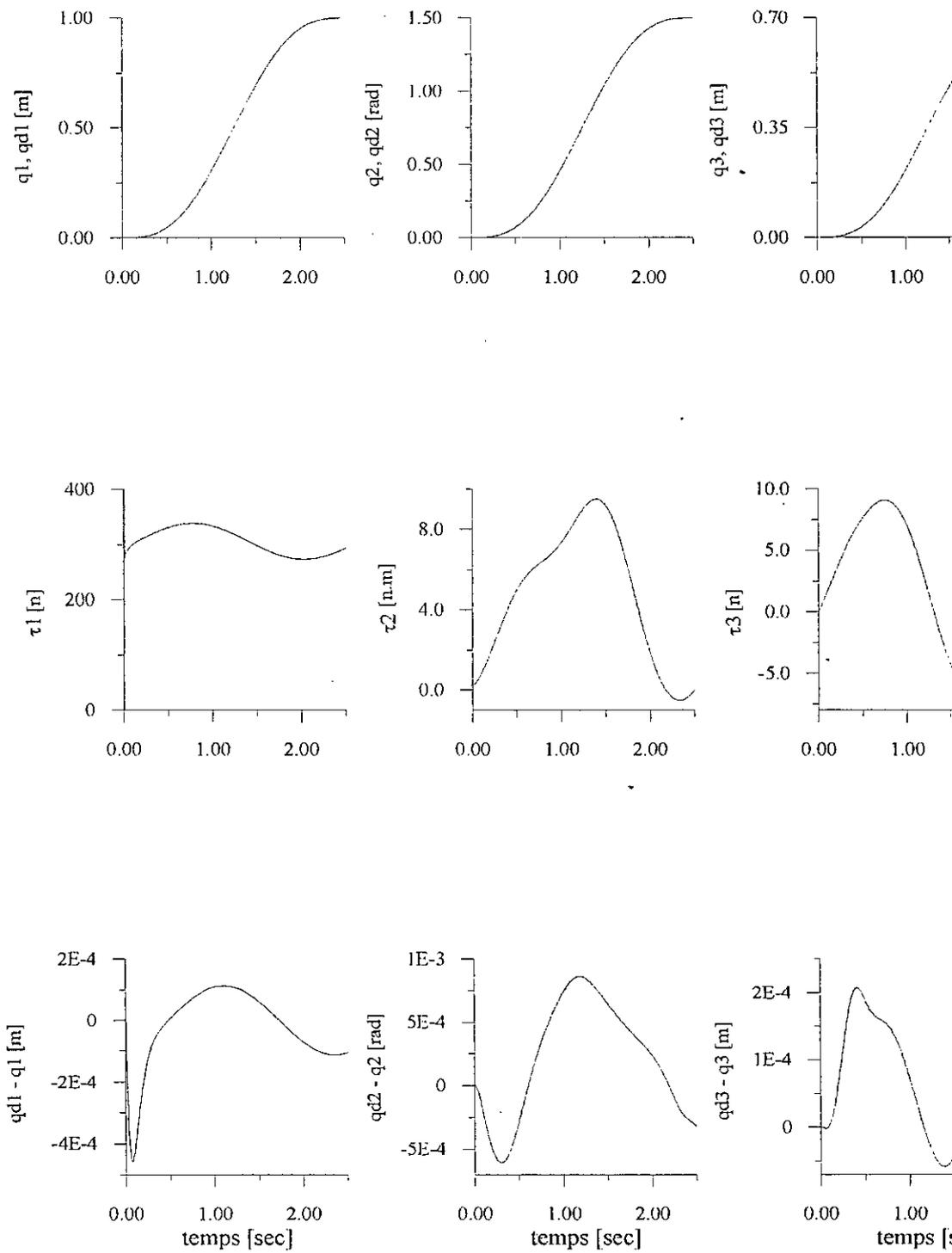
Nous procédons maintenant à l'application du schéma de commande présenté à la figure 3.4 sur le robot manipulateur PORTOS [ANNEXE]. La dynamique directe de chaque articulation est modélisée par un système flou dont la structure est résumée par le tableau 2.2:

Articulation	1	2	3
Entrées du SF f	$\dot{q}_1 \ q_1$	$\dot{q}_2 \ q_2 \ \dot{q}_3$	$\dot{q}_2 \ q_3 \ \dot{q}_3$
Nombre de règles floues de f	4	8	8

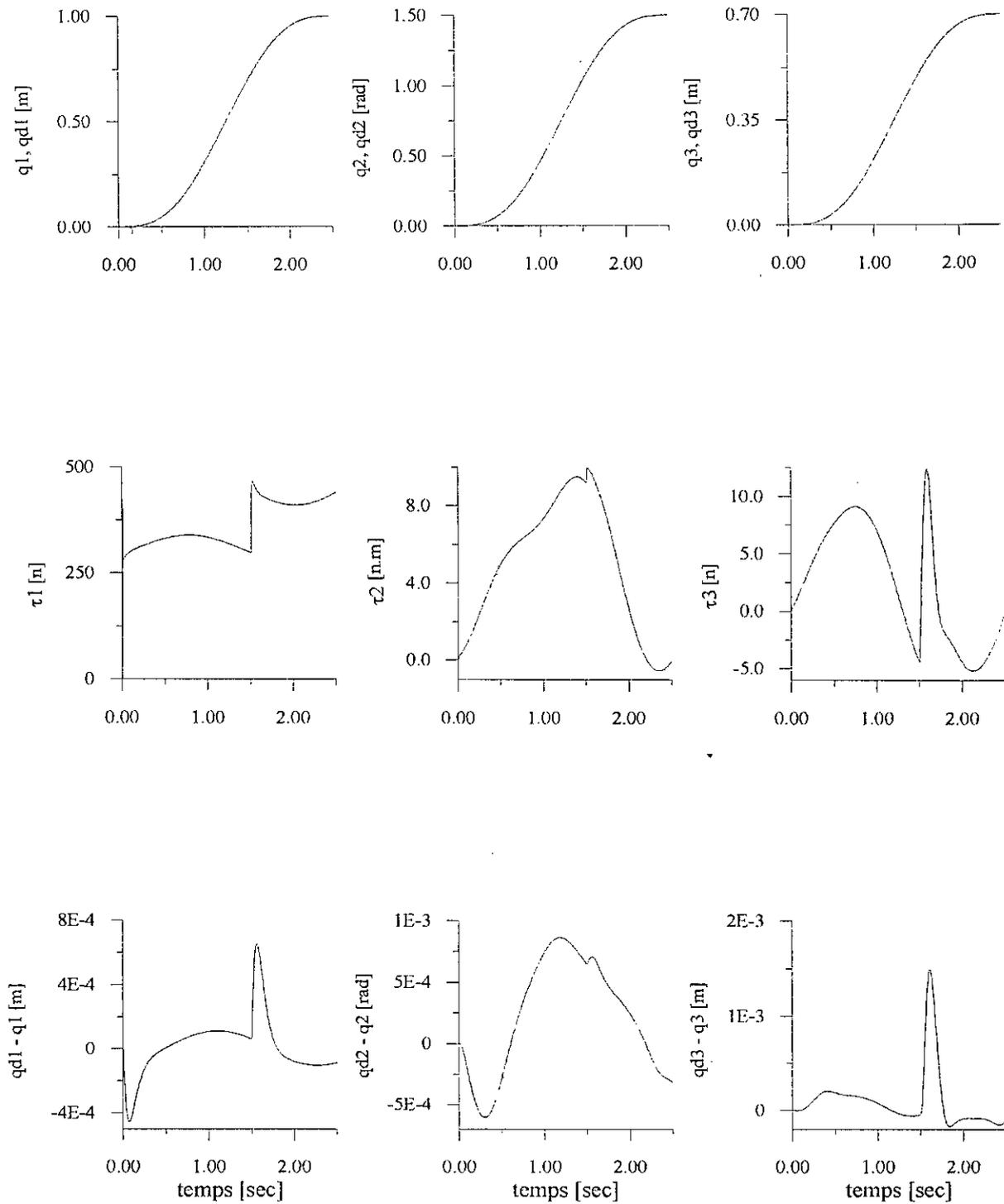
Tableau 2.2. Structure des systèmes flous utilisés.

Les paramètres  $k_v$  et  $k_p$ , sont respectivement choisis égal à 40 et 400 pour toutes les liaisons. Les résultats de simulation de la commande sans variation paramétrique sont montrés à la figure 3.5. Nous remarquons que les erreurs de poursuite sont bornées par de très faibles valeurs et que les grandeurs de commande sont lisses.

Pour démontrer la capacité d'adaptation du schéma de commande envers les variations paramétriques, nous effectuons une variation de 50% de tous les paramètres du robot à l'instant 1.5s. Les réponses obtenues sont représentées sur la figure 3.6. Nous constatons que cette perturbation paramétrique n'a pas affecté les performances de poursuite.



**Figure 3.5.** Résultats de simulation sans variations paramétriques pour la commande adaptative indirecte basée sur l'identification directe.



**Figure 3.6.** Résultats de simulation avec variations paramétriques pour la commande floue adaptative indirecte basée sur l'identification directe.

### 3.4 COMMANDE FLOUE ADAPTATIVE INDIRECTE BASEE SUR LA MINIMISATION DE L'ERREUR EN VITESSE

Dans les méthodes de commande exposées précédemment, l'ajustement des paramètres internes des systèmes flous nécessite la mesure de l'accélération réelle des articulations du robot. Or en général, les robots manipulateurs sont équipés au mieux de capteurs de position et de vitesse. De plus, le calcul numérique de l'accélération peut être néfaste en présence des bruits de mesure. Dans cette section, est proposée une approche de commande n'exigeant pas la mesure de l'accélération et l'ajustement des paramètres est basé sur la minimisation de l'erreur en vitesse.

#### 3.4.1 Modélisation du robot

Nous considérons la classe des systèmes dynamiques non linéaires MIMO dont l'équation dynamique est donnée par:

$$\ddot{q} = f(q, \dot{q}) + g(q) \tau \quad (3.17)$$

où

$q$  et  $\tau$  sont la sortie et l'entrée du système. Tandis que,  $f(q, \dot{q}) \in \mathbb{R}^n$  et  $g(q) \in \mathbb{R}^{n \times n}$  sont des fonctions non linéaires inconnues.

Notre but est de développer un modèle d'identification où les fonctions  $f(\cdot)$  et  $g(\cdot)$  sont remplacées par des modèles flous  $\hat{f}(q, \dot{q}, \theta_f)$  et  $\hat{g}(q, \theta_g)$ .

D'abord, nous écrivons (3.17) sous la forme suivante:

$$\ddot{q} = \hat{f}(q, \dot{q}, \theta_f) + \hat{g}(q, \theta_g) \tau + \varepsilon \quad (3.18)$$

avec  $\varepsilon$  est l'erreur de reconstruction donnée par:

$$\varepsilon = [f(q, \dot{q}) - \hat{f}(q, \dot{q}, \theta_f)] + [g(q) - \hat{g}(q, \theta_g)] \tau \quad (3.19)$$

telle que [JAG96], [MED97], [WAN95]:

$$\|\varepsilon\| \leq \bar{\varepsilon}$$

Nous utilisons le modèle d'identification suivant:

$$\ddot{\hat{q}} = -\alpha \dot{\hat{q}} + \alpha \hat{q} + \hat{f}(q, \dot{q}, \hat{\theta}_f) + \hat{g}(q, \hat{\theta}_g) \tau \quad (3.20)$$

où  $\alpha$  est une constante positive.

Le schéma d'identification est donné par la figure 3.7:

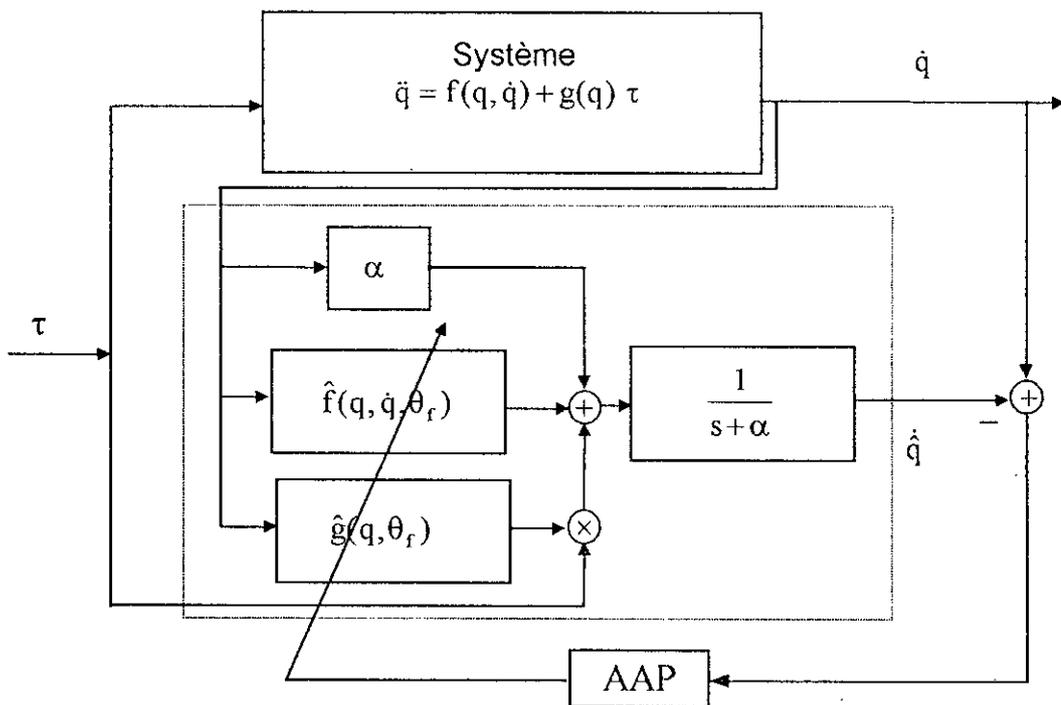


Figure 3.7. Schéma d'identification par les systèmes flous.

Le but de l'identification est de spécifier les systèmes flous  $\hat{f}(q, \dot{q}, \theta_f)$  et  $\hat{g}(q, \theta_g)$ , et de développer une loi d'adaptation des paramètres  $\theta_f$  et  $\theta_g$  afin que l'erreur

$$e = \dot{q} - \hat{\dot{q}}$$

soit aussi petite que possible.

Nous supposons que les fonctions estimées générées par les systèmes flous peuvent être écrites de la manière suivante:

$$\hat{f}(q, \dot{q}, \hat{\theta}_f) = W_f(q, \dot{q}) \hat{\theta}_f \tag{3.21}$$

$$\hat{g}(q, \hat{\theta}_g) = W_g(q) \hat{\theta}_g \tag{3.22}$$

alors que  $\hat{\theta}_f$  et  $\hat{\theta}_g$  sont les paramètres à calculer par un algorithme d'adaptation approprié. Tandis que  $W_f(\cdot)$  et  $W_g(\cdot)$  sont des fonctions floues de base [MEN95].

Les paramètres du modèle flou sont ajustés par une loi d'adaptation donnée par les équations (3.23) et (3.24) [AST89], [NAR87]:

$$\dot{\hat{\theta}}_r = \eta_r W_r^T(q, \dot{q}) e - k \|e\| \hat{\theta}_r \quad (3.23)$$

$$\dot{\hat{\theta}}_g = \eta_g W_g^T(q) e \tau - k \|e\| \hat{\theta}_g \quad (3.24)$$

avec  $\eta_r$ ,  $\eta_g$  et  $k$  sont des constantes positives.

Pour démontrer la bornitude de l'erreur d'équation  $e$ , considérons la fonction de Lyapunov suivante:

$$V = \frac{1}{2} e^T e + \frac{1}{2} (\tilde{\theta}_r^T \eta_r^{-1} \tilde{\theta}_r) + \frac{1}{2} (\tilde{\theta}_g^T \eta_g^{-1} \tilde{\theta}_g) \quad (3.25)$$

où

$$\tilde{\theta}_r = \theta_r - \hat{\theta}_r \text{ et } \tilde{\theta}_g = \theta_g - \hat{\theta}_g$$

En utilisant (3.17) et (3.20), on obtient la dynamique de l'erreur:

$$\dot{e} = -\alpha e + W_r(q, \dot{q}) \tilde{\theta}_r + W_g(q) \tilde{\theta}_g \tau + \varepsilon \quad (3.26)$$

En dérivant  $V$  et en utilisant (3.26), on obtient:

$$\dot{V} = -\alpha \|e\|^2 + e^T W_r(q, \dot{q}) \tilde{\theta}_r + e^T W_g(q) \tilde{\theta}_g \tau - \tilde{\theta}_r^T \eta_r^{-1} \dot{\hat{\theta}}_r - \tilde{\theta}_g^T \eta_g^{-1} \dot{\hat{\theta}}_g + e^T \varepsilon \quad (3.27)$$

En utilisant (3.23) et (3.24), il vient:

$$\dot{V} \leq -\alpha \|e\|^2 + k \|e\| \left\{ (\tilde{\theta}_r^T \eta_r^{-1} \hat{\theta}_r) + (\tilde{\theta}_g^T \eta_g^{-1} \hat{\theta}_g) \right\} + \bar{\varepsilon} \|e\| \quad (3.28)$$

sachons que:

$$\tilde{\theta}^T \hat{\theta} = \frac{1}{2} \theta^T \theta - \frac{1}{2} \hat{\theta}^T \hat{\theta} - \tilde{\theta}^T \tilde{\theta} \quad (3.29)$$

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

$$\dot{V} \leq -\alpha \|e\|^2 - k \|e\| \left[ \left\{ \frac{1}{\eta_r} (\tilde{\theta}_r^T \tilde{\theta}_r) + \frac{1}{\eta_g} (\tilde{\theta}_g^T \tilde{\theta}_g) \right\} - \left\{ \frac{1}{\eta_r} (\theta_r^T \theta_r) + \frac{1}{\eta_g} (\theta_g^T \theta_g) + \frac{\bar{\varepsilon}}{k} \right\} \right] \quad (3.30)$$

Alors, si

$$\frac{1}{\eta_f}(\tilde{\theta}_f^T \tilde{\theta}_f) + \frac{1}{\eta_g}(\tilde{\theta}_g^T \tilde{\theta}_g) \geq \frac{1}{\eta_f}(\theta_f^T \theta_f) + \frac{1}{\eta_g}(\theta_g^T \theta_g) + \frac{\bar{\epsilon}}{K} \quad (3.31)$$

On aura:

$$\dot{V} \leq 0 \quad (3.32)$$

Nous concluons que l'erreur d'identification  $e$  est bornée ainsi que tous les signaux du système [NAR87], [POL96], [SLO91].

### 3.4.2 Calcul de la commande

La loi de commande est calculée en utilisant le principe de l'équivalence certaine. Elle est donnée par:

$$\tau = \hat{g}(q, \hat{\theta}_g)^{-1} \{ \tau' - \hat{f}(q, \dot{q}, \hat{\theta}_f) \} \quad (3.33)$$

avec

$$\tau' = \ddot{q}_d + k_v(\dot{q}_d - \dot{q}) + k_p(q_d - q) \quad (3.34)$$

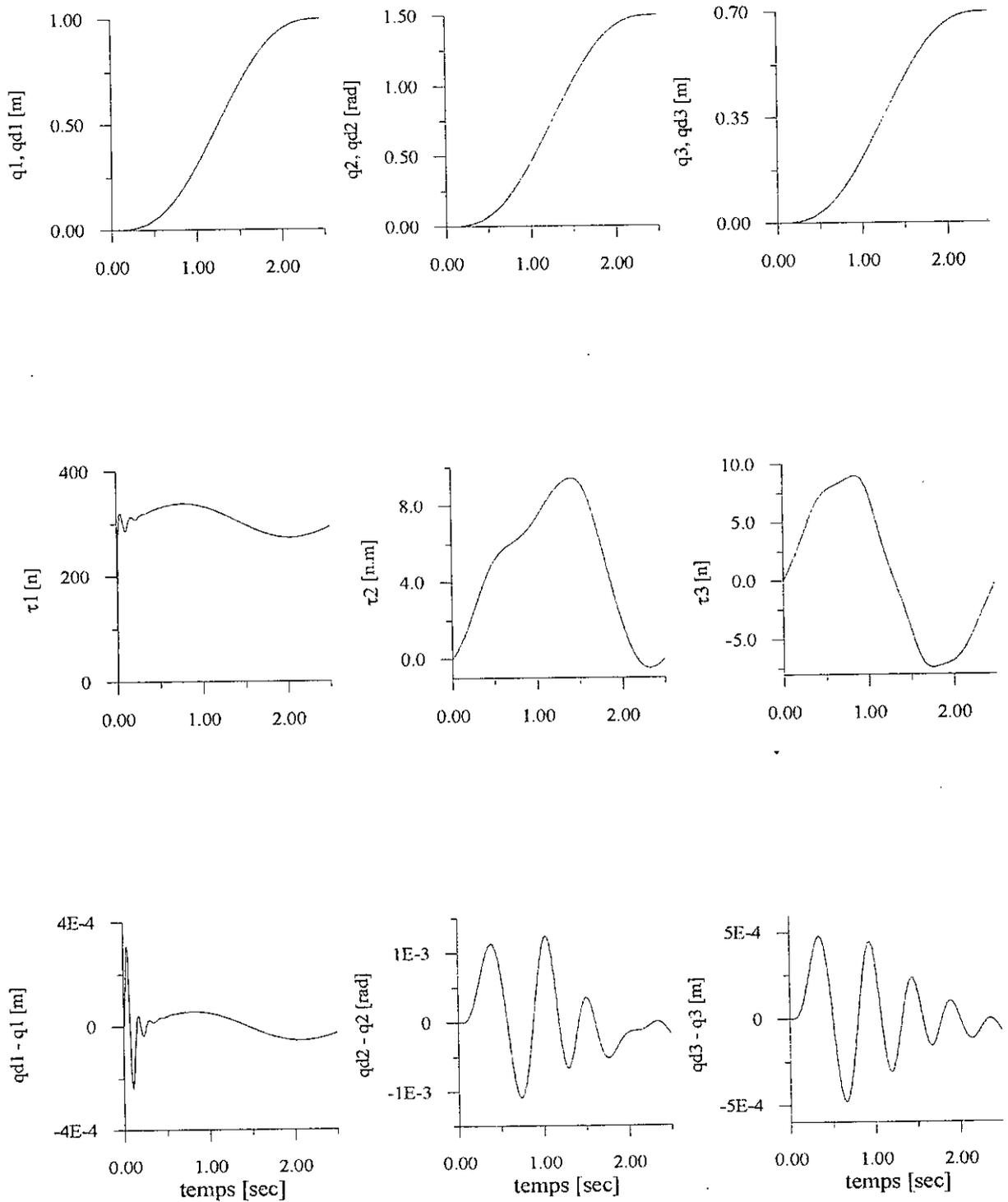
où  $k_v$  et  $k_p$  sont des constantes positives.

### 3.4.3 Résultats de simulation

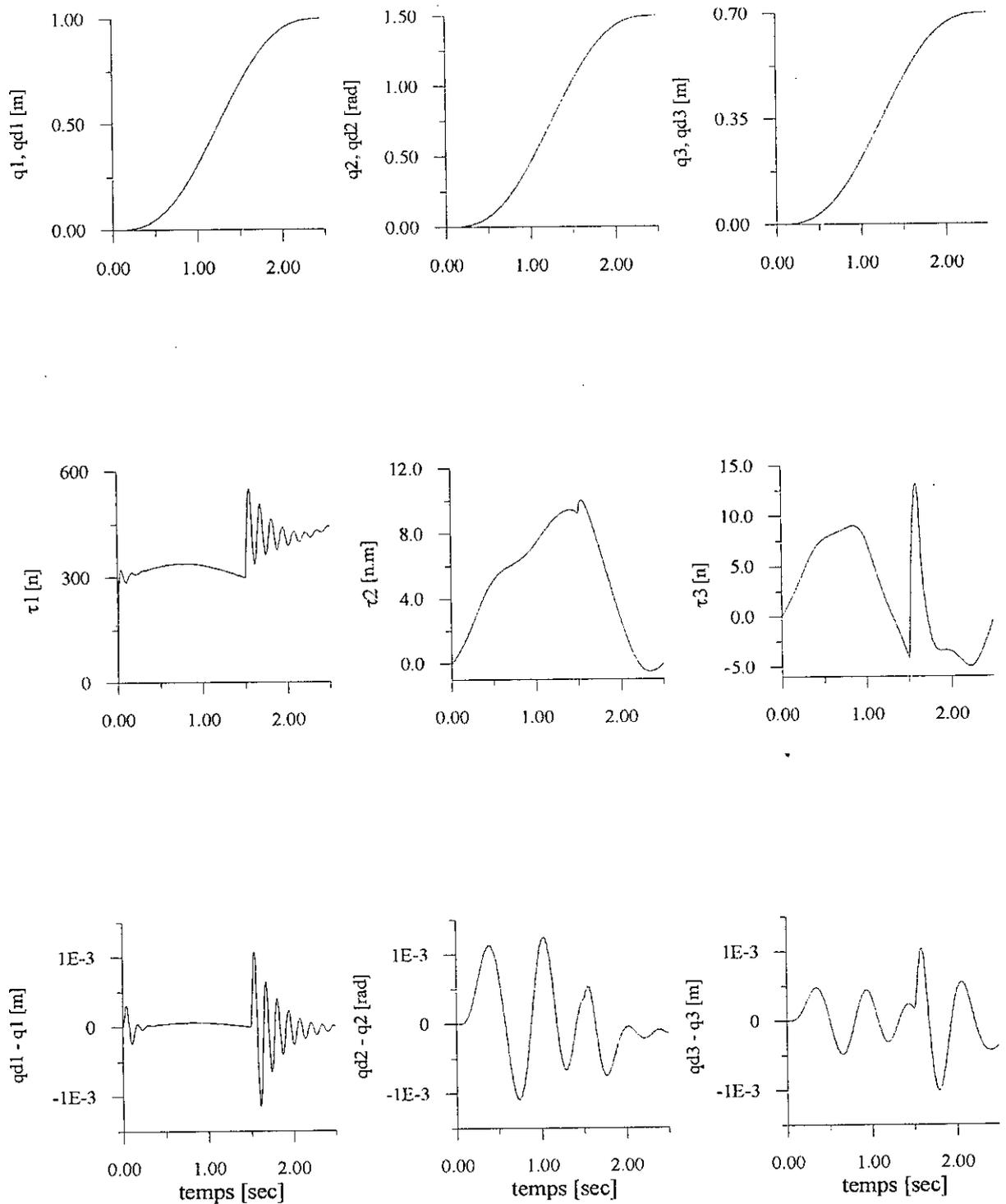
Nous présentons les résultats de la simulation numérique concernant la commande du bras manipulateur par la méthode exposée précédemment.

Les systèmes flous de Sugeno d'ordre un sont utilisés pour approximer le modèle du bras manipulateur. Les paramètres  $k_v$  et  $k_p$ , sont respectivement choisis égal à 40 et 400 pour toutes les liaisons. Les autres paramètres sont choisis comme suit:  $\alpha = 5$ ,  $k = 0.5$ ,  $\eta_f = 5$ ,  $\eta_g = 5$ . Les résultats de simulation obtenus (figure 3.8) révèlent des performances de poursuite remarquables.

Pour mettre en évidence la capacité d'adaptation du schéma de commande vis-à-vis des variations paramétriques, nous provoquons une variation de 50% de tous les paramètres du robot à l'instant 1.5s. Les réponses obtenues sont représentées à la figure 3.9. Nous remarquons clairement que cette variation paramétrique n'a pas affecté les performances de poursuite. Néanmoins, des oscillations apparaissent sur la grandeur de commande de la première articulation lors de la variation paramétrique.



**Figure 3.8.** Résultats de simulation sans variations paramétriques pour la commande floue adaptative indirecte basée sur la minimisation de l'erreur en vitesse.



**Figure 3.9.** Résultats de simulation avec variations paramétriques pour la commande floue adaptative indirecte basée sur la minimisation de l'erreur en vitesse.

### 3.5 COMMANDE FLOUE ADAPTATIVE INDIRECTE BASEE SUR LA MINIMISATION DE L'ERREUR FILTREE

Dans cette section, nous proposons une autre approche de commande adaptative indirecte où l'ajustement des paramètres est basé sur la minimisation de l'erreur filtrée.

#### 3.5.1 Modélisation du robot

Considérons de nouveau la classe des systèmes dynamiques non linéaires MIMO dont l'équation dynamique est donnée par:

$$\ddot{q} = f(q, \dot{q}) + g(q) \tau \quad (3.35)$$

avec  $q$  et  $\tau$  sont respectivement le vecteur de sortie et celui de commande.

Dans cette approche, nous utilisons le modèle d'identification suivant:

$$\ddot{\hat{q}} = -\alpha \dot{\hat{q}} - \beta \hat{q} + \alpha \dot{q} + \beta q + \hat{f}(q, \dot{q}, \hat{\theta}_f) + \hat{g}(q, \hat{\theta}_g) \tau \quad (3.36)$$

où  $\alpha$  et  $\beta$  sont des constantes positives.

Les paramètres du modèle flou sont ajustés par la loi d'adaptation suivante:

$$\dot{\hat{\theta}}_f = \eta_f W_f^T(q, \dot{q}) e_1 - k \|e_1\| \hat{\theta}_f \quad (3.37)$$

$$\dot{\hat{\theta}}_g = \eta_g W_g^T(q) e_1 \tau - k \|e_1\| \hat{\theta}_g \quad (3.38)$$

alors que  $\eta_f$ ,  $\eta_g$  et  $k$  sont des constantes positives

avec  $e_1$  est l'erreur filtrée donnée par la relation suivante:

$$e_1(t) = \dot{e}(t) + \psi e(t) \quad (3.39)$$

où  $\psi$  est une constante positive et  $e$  est l'erreur d'identification donnée par:

$$e(t) = q(t) - \hat{q}(t)$$

Le schéma d'identification est donné par la figure 3.10:

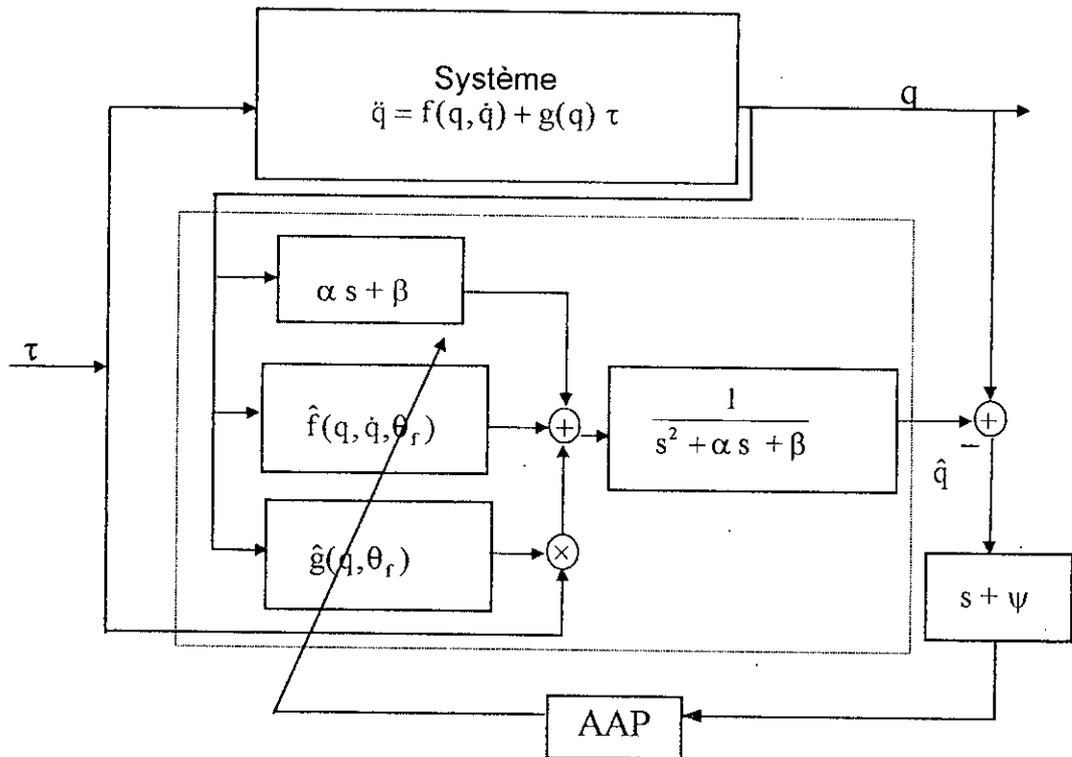


Figure 3. 10. Structure d'identification par les systèmes flous par la minimisation de l'erreur en position et celle en vitesse.

La dynamique de l'erreur est donnée par l'équation suivante:

$$\ddot{e} + \alpha \dot{e} + \beta e = W_f(q, \dot{q}) \tilde{\theta}_f + W_g(q) \tilde{\theta}_g \tau + \varepsilon \tag{3.40}$$

où

$$\tilde{\theta}_f = \theta_f - \hat{\theta}_f \text{ et } \tilde{\theta}_g = \theta_g - \hat{\theta}_g$$

Les coefficients  $\alpha$ ,  $\beta$  et  $\psi$  sont choisis de telle sorte que la fonction de transfert (3.41):

$$\frac{p + \psi}{p^2 + \alpha p + \beta} \tag{3.41}$$

soit strictement réelle positive [AST89], [GUE95].

Ce choix assure l'existence des matrices symétriques et définies positives  $P_i$  et des matrices définies positives  $Q_i$  telles que [AST89]:

$$\begin{cases} A_i^T P_i + P_i A_i = -Q_i \\ P_i B_i = C_i^T, \quad i = 1, \dots, n \end{cases} \tag{3.42}$$

Où les matrices  $A_i$ ,  $B_i$  et  $C_i$  sont les matrices de la représentation d'état de l'équation de l'erreur (3.40). On obtient donc:

$$x_i = \begin{bmatrix} e_i^T & \dot{e}_i^T \end{bmatrix} \quad (3.43)$$

$$A_i = \begin{bmatrix} 0 & 1 \\ -\beta & -\alpha \end{bmatrix}, B_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C_i = [1 \quad \psi] \quad (3.44)$$

$$\dot{x} = A x + B \{ W_f(q, \dot{q}) \tilde{\theta}_f + W_g(q) \tilde{\theta}_g \tau + \varepsilon \} \quad (3.45a)$$

$$e_i = C x \quad (3.45b)$$

Où  $A$ ,  $B$  et  $C$  sont des matrices diagonales par bloc ( avec  $A_i$ ,  $B_i$  et  $C_i$  sur leurs diagonales respectivement ).

L'équation (3.42) devient alors:

$$\begin{cases} A^T P + P A = -Q \\ P B = C^T \end{cases} \quad (3.46)$$

où

$$P = \text{diag}( P_1, \dots, P_n ) \text{ et } Q = \text{diag}( Q_1, \dots, Q_n )$$

Dans le but de démontrer la stabilité de ce schéma d'identification, nous proposons la fonction de Lyapunov suivante:

$$V = x^T P x + \tilde{\theta}_f^T \eta_f^{-1} \tilde{\theta}_f + \tilde{\theta}_g^T \eta_g^{-1} \tilde{\theta}_g \quad (3.47)$$

La dérivée temporelle de  $V$  est donnée par:

$$\dot{V} = x^T P \dot{x} + \dot{x}^T P x - 2 \left( \tilde{\theta}_f^T \eta_f^{-1} \dot{\tilde{\theta}}_f \right) - 2 \left( \tilde{\theta}_g^T \eta_g^{-1} \dot{\tilde{\theta}}_g \right) \quad (3.48)$$

Après calcul, on démontre que la dérivée de la fonction de Lyapunov vérifie:

$$\dot{V} \leq -x^T Q x - k \|C\| \|x\| \left[ \left\{ \frac{1}{\eta_f} (\tilde{\theta}_f^T \tilde{\theta}_f) + \frac{1}{\eta_g} (\tilde{\theta}_g^T \tilde{\theta}_g) \right\} - \left\{ \frac{1}{\eta_f} (\theta_f^T \theta_f) + \frac{1}{\eta_g} (\theta_g^T \theta_g) + \frac{\bar{\varepsilon}}{k} \right\} \right] \quad (3.49)$$

Alors, si la condition suivante est vérifiée.

$$\frac{1}{\eta_f} (\tilde{\theta}_f^T \tilde{\theta}_f) + \frac{1}{\eta_g} (\tilde{\theta}_g^T \tilde{\theta}_g) \geq \frac{1}{\eta_f} (\theta_f^T \theta_f) + \frac{1}{\eta_g} (\theta_g^T \theta_g) + \frac{\bar{\varepsilon}}{k} \quad (3.50)$$

La dérivée de la fonction de Lyapunov est négative:

$$\dot{V} \leq 0 \quad (3.51)$$

Nous concluons que l'erreur d'identification  $e$  est bornée ainsi que tous les signaux du système [NAR87], [POL96], [SLO91].

### 3.5.2 Calcul de la commande

La loi de commande est déterminée par la même méthode utilisée précédemment. Elle est donnée par:

$$\tau = \hat{g}(q, \hat{\theta}_g)^{-1} \{ \tau' - \hat{f}(q, \dot{q}, \hat{\theta}_f) \} \quad (3.52)$$

avec

$$\tau' = \ddot{q}_d + k_v(\dot{q}_d - \dot{q}) + k_p(q_d - q)$$

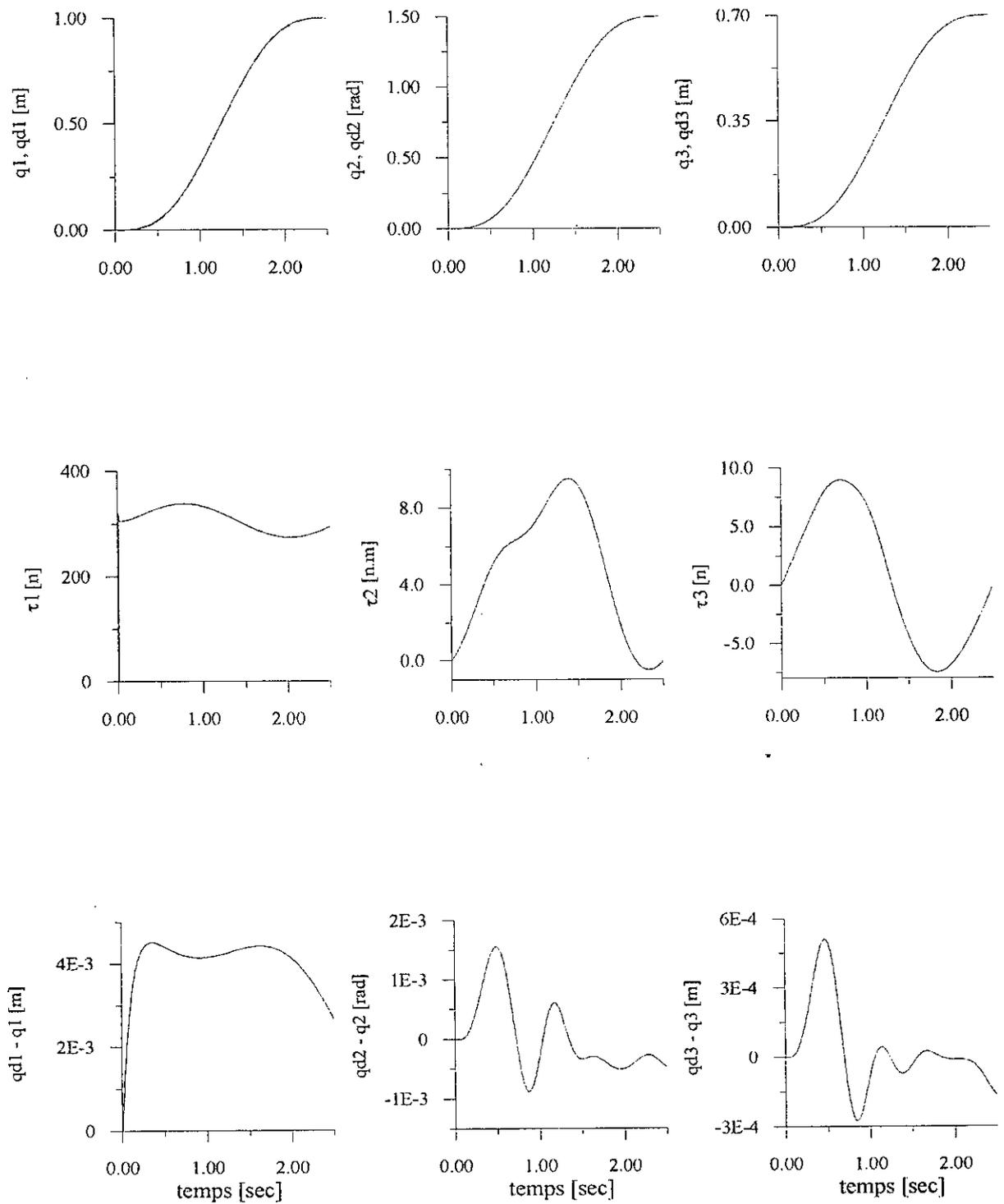
alors que  $k_v$  et  $k_p$  sont des constantes positives.

### 3.5.3 Résultats de simulation

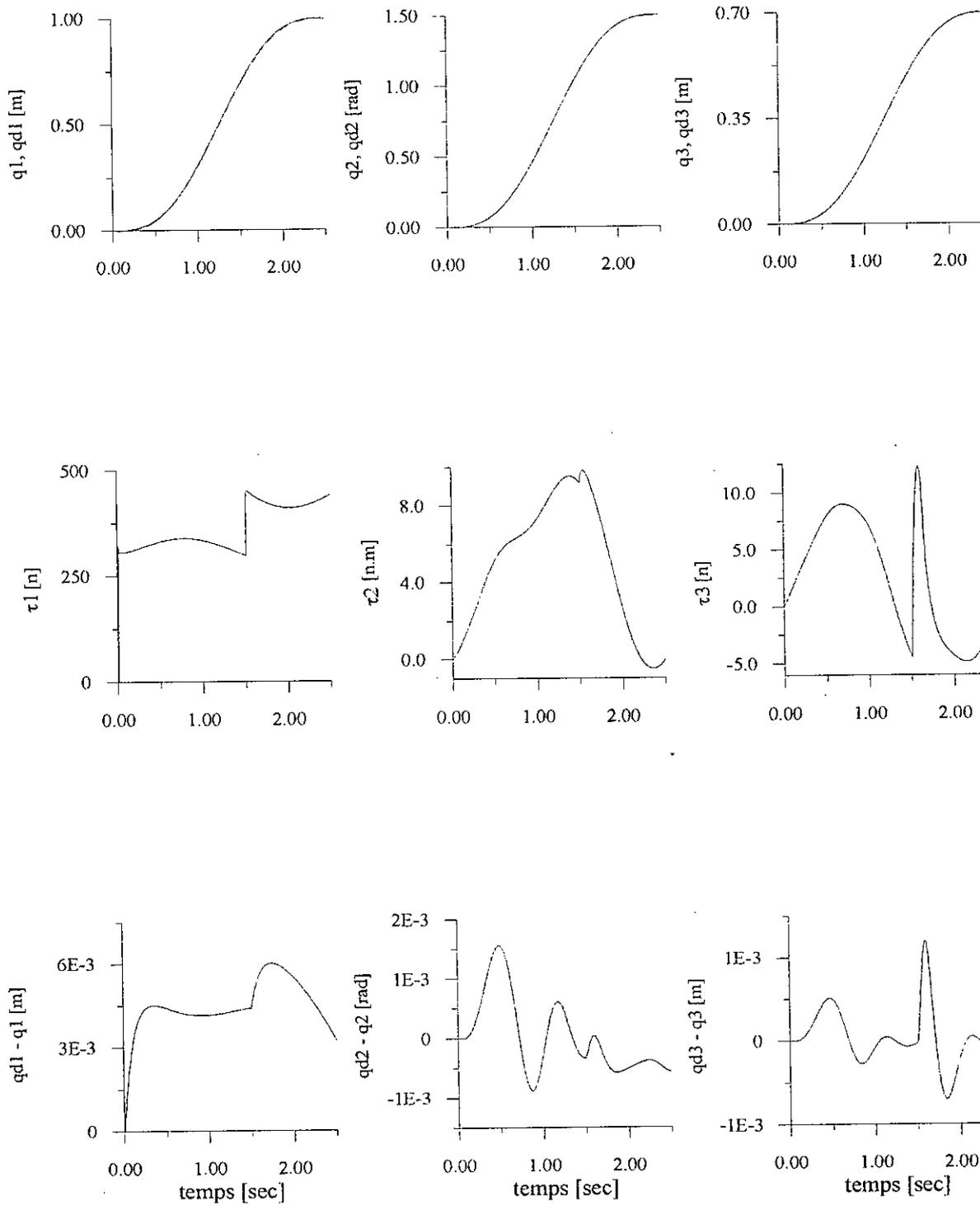
Nous présentons les résultats de la simulation effectuée sur le bras manipulateur PORTOS.

Les mêmes paramètres de synthèse que ceux de la méthode précédente sont considérés, avec  $\alpha=5$  et  $\beta=10$ . Les résultats de simulation obtenus sont montrés en figure 3.11. Cette figure représente l'évolution de la position, la commande et de l'erreur de poursuite pour chacune des articulations du robot. Nous constatons que la technique donne des performances de poursuite remarquables.

Pour tester la capacité d'adaptation du schéma de commande vis-à-vis des variations paramétriques, nous effectuons une variation de 50% de tous les paramètres du robot à l'instant 1.5s. Les réponses obtenues sont représentées sur la figure 3.12. Ces réponses montre que la commande est insensible à cette perturbation paramétrique.



**Figure 3.11.** Résultats de simulation sans variations paramétriques pour la commande floue adaptative indirecte basée sur la minimisation de l'erreur filtrée.



**Figure 3.12.** Résultats de simulation avec variations paramétriques pour la commande floue adaptative indirecte basée sur la minimisation de l'erreur filtrée.

### 3.6 CONCLUSION

Dans ce chapitre, nous avons étudié et développé des stratégies de commande adaptatives indirectes utilisant les systèmes flous, appliquées à la commande en poursuite d'une trajectoire de référence des robots manipulateurs. Ainsi, nous nous sommes intéressés à la commande floue basée sur l'identification de la dynamique inverse directe, la commande floue adaptative basée sur la minimisation de l'erreur en vitesse, et à la commande floue adaptative basée sur la minimisation de l'erreur filtrée.

La simulation effectuée sur le robot manipulateur PORTOS montre l'efficacité des techniques de commande floues proposées. En effet, nous constatons que les différentes commandes conduisent à de faibles erreurs de poursuite. De plus, les résultats montrent que ces techniques de commande sont robustes par rapport aux variations paramétriques. A partir des résultats de simulation, nous constatons également que les grandeurs de commande sont lisses. Néanmoins, dans le cas de la commande adaptative basée sur la minimisation de l'erreur en vitesse, des oscillations apparaissent sur la grandeur de commande de la première articulation lors de l'application de la perturbation paramétrique.

Enfin, cette étude nous a permis de formuler les remarques suivantes:

- Contrairement à la commande adaptative classique des robots manipulateurs, l'utilisation des systèmes flous pour la modélisation du robot a permis d'éviter les problèmes de modélisation.
- Les performances de la commande dépendent du nombre de règles floues choisi. Bien sûr, celles-ci peuvent être améliorées en augmentant le nombre de règles. Néanmoins, un compromis doit être fait, car plus le nombre de règles est important, plus le correcteur est lent.
- Les deux premières approches de commandes, à savoir l'approche basée sur l'identification inverse et celle basée sur l'identification directe, sont d'une part des approches de commande à structure décentralisée et d'autre part, le correcteur est un système flou dont les règles sont construites à partir de celles du modèle flou. De plus, ces deux approches nécessitent la connaissance de l'accélération réelle du robot. Cette dernière est calculée numériquement, ce qui peut être néfaste dans le cas des mesures bruitées.
- Dans les trois dernières techniques de commande, le calcul de la loi de commande contient une division par un terme. Ceci peut provoquer une division par zéro lors de l'adaptation des paramètres de ces termes. Pour remédier à ce problème, on peut utiliser les algorithmes de projection pour l'adaptation des paramètres [KHA96].

## CHAPITRE 4

# *COMMANDE FLOUE ADAPTATIVE DIRECTE*

### 4.1 INTRODUCTION

Dans ce chapitre, nous nous intéressons au schéma de commande adaptatif direct, où les paramètres du contrôleur sont ajustés directement de telle façon à minimiser une certaine norme de l'erreur de sortie. Les systèmes flous sont directement applicables à la commande adaptative des systèmes non linéaires, car ils sont des éléments non linéaires adaptatifs. En effet, les systèmes flous sont souvent utilisés pour la modélisation des systèmes dans la commande indirecte. En revanche, l'utilisation des systèmes flous dans la commande adaptative directe reste très restreinte [JAN95]. Dans le schéma de commande adaptatif direct, la difficulté principale réside dans la manière d'utiliser l'erreur de sortie pour l'ajustement des paramètres du contrôleur, car le système est interposé entre la sortie à commander et celle du correcteur.

Les méthodes existantes de la synthèse des correcteurs flous ne garantissent pas nécessairement la stabilité. Actuellement, la recherche est orientée vers l'étude de la stabilité des contrôleurs flous. Les outils de synthèse et d'analyse nécessaires pour l'étude de la stabilité de la commande des systèmes non linéaires paramétrés, entre autres les systèmes flous, sont fournis par la théorie de la commande adaptative robuste, la théorie de l'approximation, et la théorie de la stabilité des systèmes non linéaires [POL96], [SAN92].

Le but de ce chapitre, est de développer des lois de commandes adaptatives directes des robots manipulateurs. En effet, les systèmes flous sont utilisés pour approximer le modèle du bras manipulateur. Et afin de compenser les effets des erreurs de reconstruction, nous introduisons un terme du mode de glissement dans la loi de commande [SLO88]. En premier lieu, nous développons la commande floue adaptative linéarisante. En second lieu, une technique de commande adaptative directe stable par les systèmes flous est étudiée. Par la suite, une extension de cette approche à une classe des systèmes non linéaires est proposée ainsi qu'une autre stratégie de commande plus générale.

## 4.2 COMMANDE FLOUE ADAPTATIVE LINEARISANTE

### 4.2.1 Formulation du problème

Considérons la classe des robots manipulateurs rigides ayant des dynamiques décrites par l'équation suivante [SLO88], [SLO91]:

$$H(q) \ddot{q} + C(q, \dot{q}) \dot{q} + E(q, \dot{q}) = \tau \quad (4.1)$$

où

$q$ : vecteur des  $n$  coordonnées articulaires du robot.

$\dot{q}$ ,  $\ddot{q}$ : respectivement les dérivées premières et secondes de  $q$ .

$\tau$ : vecteur contenant les couples ou les forces de commande.

$H(q)$ : matrice d'inertie symétrique et définie positive.

$C(q, \dot{q})\dot{q}$ : représente les forces de coriolis et les forces centripètes.

$E(q, \dot{q})$ : vecteur regroupant les forces de gravités et les frottements.

Pour faciliter la synthèse de la loi de commande, le modèle du robot manipulateur est mis sous la forme compacte suivante :

$$H(q) \ddot{q} + G(q, \dot{q}) = \tau \quad (4.2)$$

avec

$$G(q, \dot{q}) = C(q, \dot{q}) \dot{q} + E(q, \dot{q}) \quad (4.3)$$

Dans le cas d'une connaissance exacte du modèle du robot manipulateur, la méthode de la commande linéarisante dite du couple calculé est directement applicable [ISI89], [SLO91], [GUE95]. Cette dernière est donnée par la relation suivante:

$$\tau = H(q)\tau' + G(q, \dot{q}) \quad (4.4)$$

où

$$\tau'(t) = \ddot{q}_d(t) + k_v \dot{e}(t) + k_p e(t)$$

alors que  $k_v$  et  $k_p$  sont choisis de telle sorte que le polynôme  $p^2 + k_v p + k_p$  soit stable.

Une version adaptative de la loi de commande (4.4) a été étudiée dans le cas où le modèle du bras manipulateur est méconnu [GUE95].

Le problème de la commande adaptative linéarisante par les systèmes flous consiste à déterminer une loi de commande semblable à celle donnée par la relation (4.4). En effet les systèmes flous seront utilisés pour approximer le modèle inconnu du robot manipulateur. De ce fait, cette technique n'exige pas un modèle de connaissance du robot. Pour faciliter la synthèse de la commande floue, le modèle non linéaire du robot manipulateur est écrit sous la forme compacte suivante:

$$H(q) \ddot{q} + G(q, \dot{q}) = f(q, \dot{q}, \ddot{q}) \quad (4.5)$$

#### 4.2.2 Approximation par les systèmes flous

Nous considérons dans cette section le problème d'approximation des fonctions non linéaires par les systèmes flous du type Sugeno de la forme  $W(x)\theta$  ; c'est-à-dire les systèmes flous dont les paramètres des prémisses sont fixés a priori et les paramètres de la partie conclusion sont les seuls à ajuster.

Nous supposons qu'il existe un système flou de la forme  $W_f(q, \dot{q}, \ddot{q})\theta$  de telle sorte que la dynamique du robot manipulateur (4.5) est écrite sous la forme suivante:

$$f(q, \dot{q}, \ddot{q}) = W_f(q, \dot{q}, \ddot{q}) \theta + \varepsilon_f \quad (4.6)$$

où  $\varepsilon_f$  est l'erreur de reconstruction de la fonction  $f(\cdot)$ , et qui est due à l'approximation de la fonction réelle par un système flou, telle que [JAG96], [POL96]:

$$|\varepsilon_f| \leq \bar{\varepsilon} \quad (4.7)$$

L'estimation  $\hat{f}(q, \dot{q}, \ddot{q})$  générée par le système flou est donnée par l'expression suivante:

$$\hat{f}(q, \dot{q}, \ddot{q}) = W_f(q, \dot{q}, \ddot{q}) \hat{\theta} \quad (4.8)$$

tel que:

$$\hat{f}(q, \dot{q}, \ddot{q}) = \hat{H}(q) \ddot{q} + \hat{G}(q, \dot{q}) \quad (4.9)$$

alors que  $\hat{\theta}$  est le vecteur des paramètres à estimer par une loi d'adaptation appropriée, et  $W_f(\cdot)$  est une matrice contenant des fonctions floues de base [MEN95].

### 4.2.3 Synthèse de la commande

La commande appliquée au robot manipulateur est calculée en utilisant le modèle flou estimé selon la relation suivante:

$$\tau = \hat{H}(q) (\tau' + u_{gl}) + \hat{G}(q, \dot{q}) \quad (4.10)$$

où  $u_{gl}$  est un terme du mode de glissement utilisé pour compenser l'effet de l'erreur de reconstruction. Son expression est donnée par l'équation suivante:

$$u_{gl} = \text{sgn}(s) \left| \hat{H}^{-1} \right| k_{gl} \quad (4.11)$$

avec

$$\text{sgn}(s) = \text{diag}(\text{sign}(s_i))_{i=1, \dots, n} \quad (4.12a)$$

et

$$k_{gl} = \bar{\varepsilon} \quad (4.12b)$$

où  $s$  est l'erreur filtrée donnée par:

$$s(t) = \dot{e}(t) + \lambda e(t) \quad (4.13)$$

avec

$$e(t) = q_d(t) - q(t)$$

alors que  $\lambda$  est une constante positive choisie de telle sorte que la fonction de transfert suivante:

$$F(s) = \frac{p + \lambda}{p^2 + k_v p + k_p} \quad (4.14)$$

soit strictement réelle positive [AST89], [GUE95].

Les paramètres du système flou sont estimés par l'algorithme d'adaptation suivant:

$$\dot{\hat{\theta}} = \Gamma W_f^T(q, \dot{q}, \ddot{q}) \hat{H}^{-1} s \quad (4.15)$$

où  $\Gamma$  est une matrice constante définie positive.

Les lois de commande et d'adaptation définies ci-dessus garantissent la bornitude des signaux du système en boucle fermée ainsi que la convergence globale de l'erreur de trajectoire et celle en vitesse tant que les trajectoires désirées sont bornées. Cependant, l'inconvénient majeur de cette stratégie est qu'elle nécessite la mesure de l'accélération réelle des articulations et l'inversion de la matrice d'inertie estimée.

La structure de commande adoptée est donnée par la figure 4.1 :

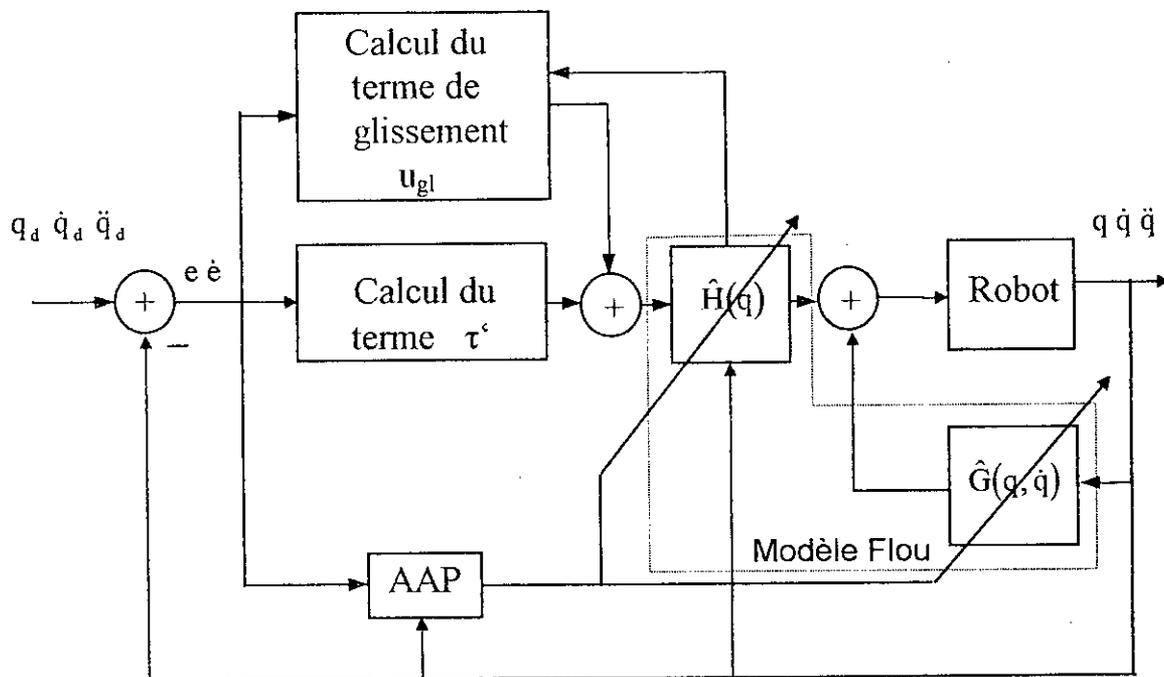


Figure 4.1. Structure de la commande floue adaptative linéarisante.

#### 4.2.4 Analyse de la stabilité

Dans cette section nous allons utiliser l'approche de Lyapunov pour démontrer la stabilité du système bouclé.

En utilisant la loi de commande (4.10) et le modèle dynamique du robot manipulateur (4.1), la dynamique de l'erreur est donnée par:

$$\ddot{e}(t) + k_v \dot{e}(t) + k_p e(t) = -\hat{H}^{-1} \{ \hat{H}(q) \ddot{q} + \hat{G}(q, \dot{q}) - H(q) \ddot{q} - G(q, \dot{q}) \} - u_{gl} \quad (4.16)$$

Ainsi,

$$\ddot{e}(t) + k_v \dot{e}(t) + k_p e(t) = -\hat{H}^{-1} W_r(q, \dot{q}, \ddot{q}) \tilde{\theta} + \hat{H}^{-1} \varepsilon_r - u_{gl} \quad (4.17)$$

où l'erreur paramétrique est définie comme suit:

$$\tilde{\theta} = \hat{\theta} - \theta, \text{ ainsi } \ddot{\tilde{\theta}} = \ddot{\hat{\theta}} \quad (4.18)$$

En utilisant le signal de l'erreur filtrée donné par:

$$s(t) = \dot{e}(t) + \lambda e(t) \quad (4.19)$$

Le choix approprié des coefficients  $k_v$ ,  $k_p$  et  $\lambda$  assure l'existence des matrices symétriques définies positives  $P_i$  et des matrices définies positives  $Q_i$  tel que [AST89]:

$$\begin{cases} A_i^T P_i + P_i A_i = -Q_i \\ P_i B_i = C_i^T, \quad i = 1, \dots, n \end{cases} \quad (4.20)$$

où les matrices  $A_i$ ,  $B_i$  et  $C_i$  sont les matrices de la représentation d'état de l'équation de l'erreur (4.17). On obtient donc:

$$x_i = \begin{bmatrix} e_i^T & \dot{e}_i^T \end{bmatrix} \quad (4.21)$$

$$A_i = \begin{bmatrix} 0 & 1 \\ -k_p & -k_v \end{bmatrix}, \quad B_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_i = \begin{bmatrix} 1 & \lambda \end{bmatrix} \quad (4.22)$$

$$\dot{x} = A x + B \left\{ -\hat{H}^{-1} W_f(q, \dot{q}, \ddot{q}) \tilde{\theta} - u_{gl} + \hat{H}^{-1} \varepsilon_f \right\} \quad (4.23a)$$

$$s = C x \quad (4.23b)$$

où  $A$ ,  $B$  et  $C$  sont des matrices diagonales par bloc ( avec  $A_i$ ,  $B_i$  et  $C_i$  sur leurs diagonales respectivement ).

L'équation (4.20) devient alors:

$$\begin{cases} A^T P + P A = -Q \\ P B = C^T \end{cases} \quad (4.24)$$

où

$$P = \text{diag}( P_1, \dots, P_n ) \text{ et } Q = \text{diag}( Q_1, \dots, Q_n )$$

Afin de démontrer la stabilité de ce schéma de commande, nous considérons la fonction de Lyapunov suivante:

$$V = x^T P x + \tilde{\theta}^T \Gamma^{-1} \tilde{\theta} \quad (4.25)$$

La dérivée temporelle de la fonction  $V$  est donnée par:

$$\dot{V} = \dot{x}^T P x + x^T P \dot{x} + 2 \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} \quad (4.26)$$

A partir des relations (4.23a), (4.23b) et (4.24), l'expression de la dérivée de  $V$  devient:

$$\dot{V} = -x^T Q x + 2 s^T \hat{H}^{-1} \varepsilon_f - 2 s^T u_{gl} - 2 s^T \hat{H}^{-1} W_f(q, \dot{q}, \ddot{q}) \tilde{\theta} + 2 \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} \quad (4.27)$$

En utilisant la relation (4.15), il vient:

$$\dot{V} = -x^T Q x + 2 s^T \hat{H}^{-1} \varepsilon_r - 2 s^T u_{gl}$$

D'où on peut écrire:

$$\dot{V} \leq -x^T Q x + 2 |s^T| |\hat{H}^{-1}| \bar{\varepsilon} - 2 s^T u_{gl}$$

En utilisant l'expression du terme de glissement (4.11), on obtient:

$$\dot{V} \leq -x^T Q x \tag{4.28}$$

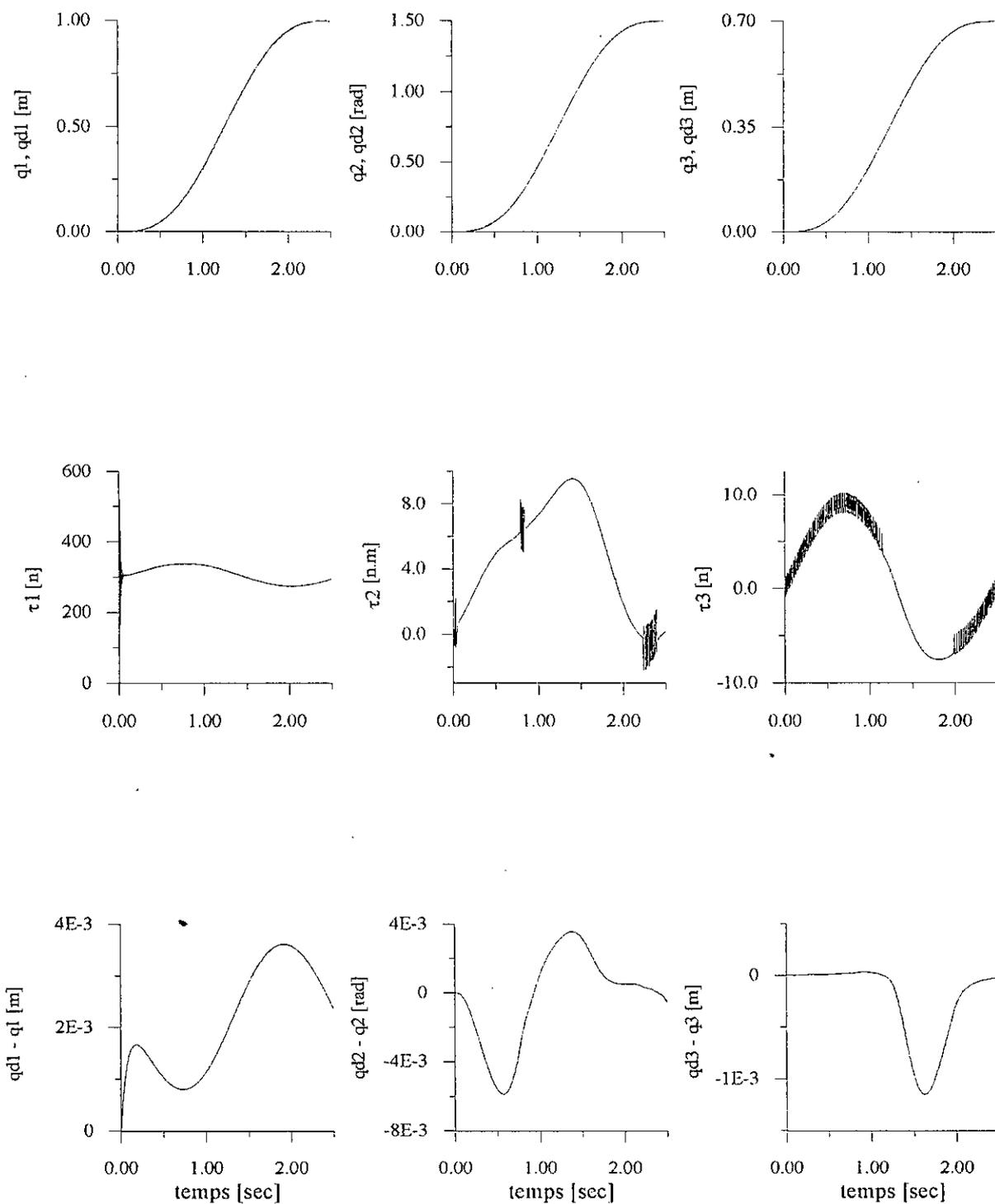
Par conséquent, tous les signaux du système sont bornés. De plus, l'erreur de trajectoire et celle en vitesse convergent asymptotiquement vers zéro [MED97], [POL96], [SLO91], [GIL88].

#### 4.2.5 Résultats de simulation

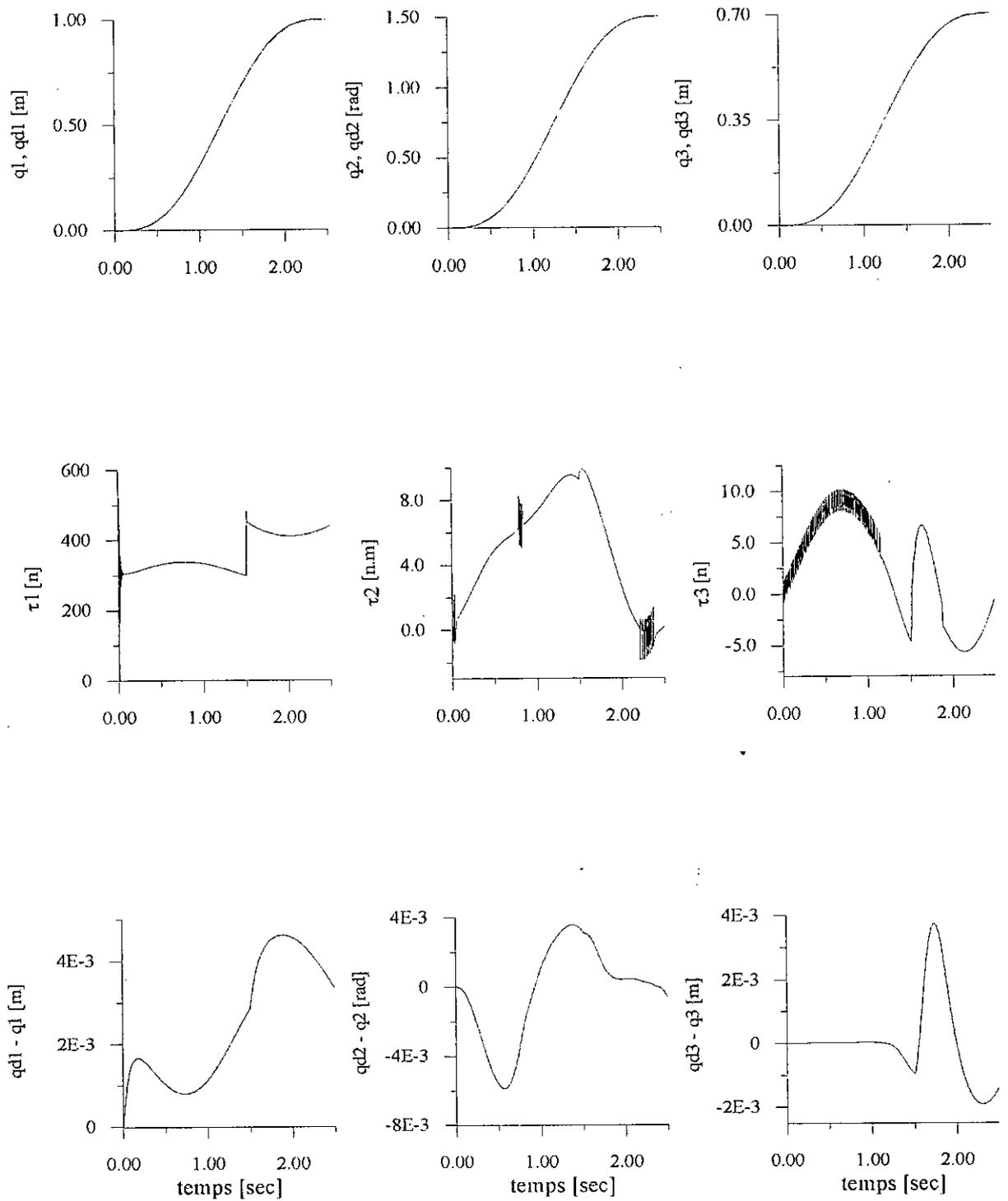
Après avoir étudié le développement de la commande floue adaptative linéarisante, nous proposons dans cette section l'application de cette stratégie de commande au robot manipulateur PORTOS [ANNEXE].

Le modèle dynamique inconnu du robot manipulateur est approximé par des systèmes flous de Sugeno d'ordre un. Les paramètres de synthèse sont choisis comme suit:  $k_v = 20$ ,  $k_p = 100$ ,  $\lambda = 5$ ,  $k_{gl} = [0.1, 1.5, 1.5]^T$ ,  $\Gamma = \text{diag}(10, \dots, 10)$ . Les résultats de simulation obtenus sont montrés en figure 4.2. Cette figure illustre l'évolution de la position, la commande et l'erreur de chaque articulation. D'après les erreurs de poursuite, il apparaît clairement qu'il y a une bonne qualité de poursuite.

Pour mettre en évidence la capacité d'adaptation du schéma de commande lors de l'exécution des mouvements désirés face aux variations paramétriques, nous avons effectué une variation de 50% de tous les paramètres du robot à l'instant 1.5. Les réponses obtenues sont représentées à la figure 4.3. Ces réponses témoignent que les variations paramétriques n'ont pas affecté les performances de poursuite. Cependant, il faut noter l'existence d'un léger effet de chattering sur l'effort de commande du au terme du mode de glissement.



**Figure 4.2.** Résultats de simulation sans variations paramétriques pour la commande floue linéarisante adaptative.



**Figure 4.3.** Résultats de simulation avec variations paramétriques pour la commande floue linéarisante adaptative.

### 4.3 COMMANDE FLOUE ADAPTATIVE DIRECTE STABLE

#### 4.3.1 Formulation du problème

Considérons la classe des robots manipulateurs décrite par le modèle dynamique suivant [SLO91]:

$$H(q) \ddot{q} + C(q, \dot{q}) \dot{q} + E(q, \dot{q}) = \tau \quad (4.29)$$

et vérifiant la condition suivante [SLO88], [SLO91]:

$$x^T (\dot{H} - 2C)x = 0, \quad \forall x \in \mathbb{R}^n \quad (4.30)$$

Lorsque tous les termes de (4.29) sont exactement connus, la loi de commande qui permet d'avoir une poursuite asymptotique d'une trajectoire de référence  $q_d$ , avec  $\dot{q}_d$  et  $\ddot{q}_d$  connues et bornées, est donnée par [SLO88], [SLO91], [SAA94]:

$$\tau = k_d s + H(q) \ddot{q}_r + C(q, \dot{q}) \dot{q}_r + E(q, \dot{q}) \quad (4.31)$$

où

$$\begin{aligned} e &= q_d - q \\ s &= \dot{e} + \lambda e \\ \dot{q}_r &= \dot{q}_d + \lambda e \end{aligned} \quad (4.32)$$

alors que  $k_d$  et  $\lambda$  sont des constantes positives.

Dans la plus part des applications pratiques, les termes  $H$ ,  $C$  et  $E$  de l'équation du mouvement ne sont pas précisément connus. Une alternative pour commander le robot, est de rendre adaptative la stratégie de commande (4.31). Pour synthétiser la loi de commande, le modèle non linéaire du robot est paramétrisé de la manière suivante [SLO91]:

$$H(q) \ddot{q}_r + C(q, \dot{q}) \dot{q}_r + E(q, \dot{q}) = W(q, \dot{q}, \dot{q}_r, \ddot{q}_r) \theta \quad (4.33)$$

où  $W(\cdot)$  est une matrice des fonctions non linéaires connues et  $\theta$  est un vecteur des paramètres physiques du robot. Slotine et Li [SLO88] ont proposé les lois de commande et d'adaptation suivantes:

$$\tau = k_d s + W(q, \dot{q}, \dot{q}_r, \ddot{q}_r) \hat{\theta} \quad (4.34)$$

$$\dot{\hat{\theta}} = \Gamma W^T s \quad (4.35)$$

où  $\hat{\theta}$  est une estimation adaptative des constantes physiques  $\theta$  et  $\Gamma$  est une matrice constante définie positive contrôlant la vitesse d'adaptation.

Cette loi de commande exige une connaissance parfaite du modèle dynamique du robot avec simplement des connaissances insuffisantes sur les paramètres. De plus, la convergence des erreurs n'est pas assurée en présence des incertitudes non structurales. Pour résoudre de tels problèmes, une approche de commande floue est proposée. Dans ce cas, l'équation (4.29) est mise sous une forme compacte de la manière suivante:

$$H(q) \ddot{q}_r + C(q, \dot{q}) \dot{q}_r + E(q, \dot{q}) = f(x) \quad (4.36)$$

avec  $x^T = [q^T, \dot{q}^T, \ddot{q}_r^T, \ddot{q}_r^T]$

### 4.3.2 Approximation par les systèmes flous

L'architecture de commande présentée dans cette partie utilise les systèmes flous de Sugeno; où la sortie est linéairement dépendante des paramètres, pour approximer la fonction inconnue  $f(x)$ . Ainsi, nous supposons qu'il existe des systèmes flous de la forme  $W_f(x)\theta$  tel que:

$$f(x) = W_f(x) \theta + \varepsilon_f \quad (4.37)$$

où  $W_f(x)$  est une matrice de fonctions floues de base [MEN95], et  $\theta$  est un vecteur de paramètres optimaux. Tandis que,  $\varepsilon_f$  est l'erreur de reconstruction de la fonction  $f(x)$ , tel que [MED97], [POL96], [WAN92b]:

$$|\varepsilon_f| \leq \bar{\varepsilon} \quad (4.38)$$

L'estimation  $\hat{f}(x)$  générée par le système flou est donnée par:

$$\hat{f}(x) = W_f(x) \hat{\theta} \quad (4.39)$$

où  $\hat{\theta}$  est le vecteur des paramètres à estimer par un algorithme d'adaptation approprié.

### 4.3.3 Synthèse de la commande

A partir de la fonction floue estimée, nous écrivons l'expression du contrôleur:

$$\tau = k_d s + W_f(x) \hat{\theta} + K \operatorname{sgn}(s) \quad (4.40)$$

où  $K \operatorname{sgn}(s)$  est un terme de mode de glissement qui compense l'effet de l'erreur de reconstruction tel que:

$$K \operatorname{sgn}(s) = [k_i \operatorname{sgn}(s_i)]_{i=1, \dots, n}^T \quad (4.41)$$

avec

$$k = \bar{\varepsilon} \quad (4.42)$$

Les paramètres du système flou sont estimés par l'algorithme d'adaptation suivant:

$$\hat{\theta} = \Gamma W_r^T(x) s \quad (4.43)$$

où  $\Gamma$  est une matrice constante définie positive.

La structure de commande adoptée est donnée par la figure 4.4:

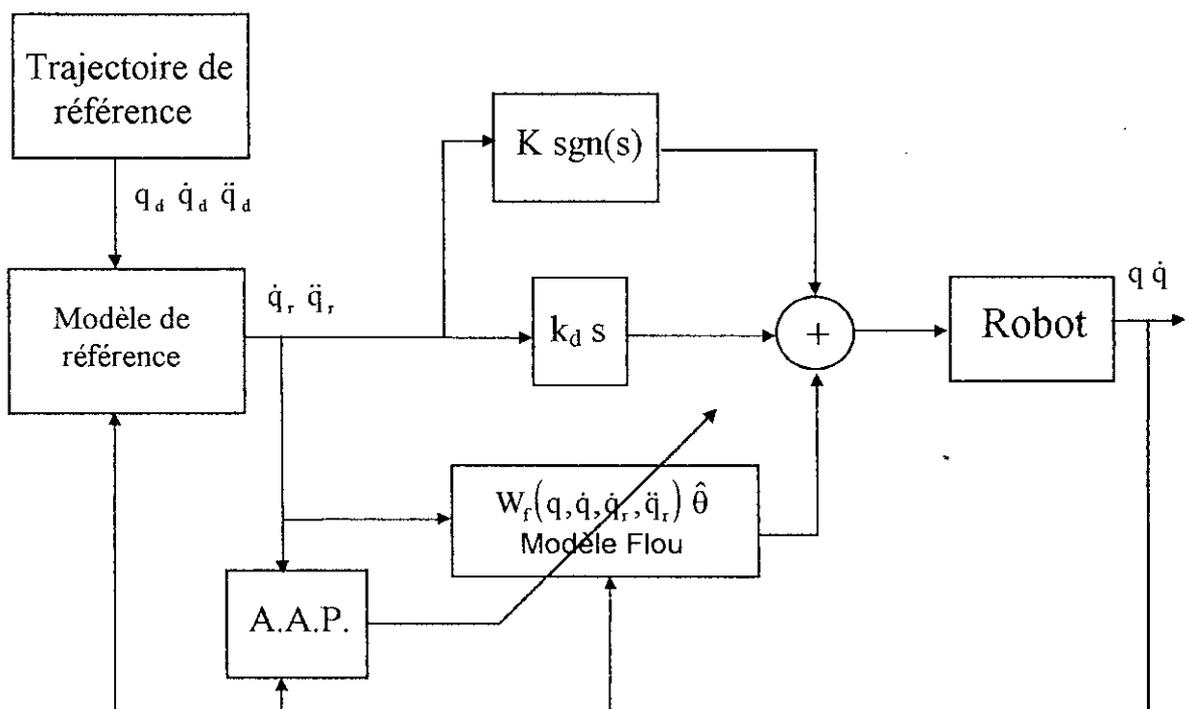


Figure 4.4. Schéma de la commande floue adaptative stable des robots manipulateurs.

### 4.3.4 Analyse de la stabilité

Dans cette section nous utilisons l'approche de Lyapunov pour prouver la stabilité du système bouclé.

On définit l'erreur paramétrique comme suit:

$$\tilde{\theta} = \hat{\theta} - \theta, \text{ ainsi } \dot{\tilde{\theta}} = \dot{\hat{\theta}}$$

En utilisant la loi de commande (4.40), la dynamique de l'erreur filtrée  $s$  est donnée par:

$$H \dot{s} + C s + k_d s = f(x) - \hat{f}(x) - K \operatorname{sgn}(s) \quad (4.44)$$

En utilisant (4.37) et (4.39), il vient:

$$H \dot{s} + C s + k_d s = -W_f(x) \tilde{\theta} + \varepsilon_f - K \operatorname{sgn}(s) \quad (4.45)$$

Considérons la fonction candidate de Lyapunov suivante:

$$V = \frac{1}{2} s^T H s + \frac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta} \quad (4.46)$$

La dérivée temporelle de cette fonction est donnée par l'expression suivante:

$$\dot{V} = \frac{1}{2} s^T \dot{H} s + s^T H \dot{s} + \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} \quad (4.47)$$

En utilisant (4.45), il vient:

$$\dot{V} = \frac{1}{2} s^T \dot{H} s - s^T C s - s^T k_d s - W_f(x) \tilde{\theta} + \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} + s^T \varepsilon_f - s^T K \operatorname{sgn}(s) \quad (4.48)$$

Posons:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 + \dot{V}_3 + \dot{V}_4 \quad (4.49)$$

avec

$$\dot{V}_1 = -s^T k_d s$$

$$\dot{V}_2 = \frac{1}{2} s^T \dot{H} s - s^T C s$$

$$\dot{V}_3 = s^T W_f(x) \tilde{\theta} - \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}}$$

$$\dot{V}_4 = s^T \varepsilon_f - s^T K \operatorname{sgn}(s)$$

La première composante est négative puisque  $k_d$  est une constante positive par définition:

$$\dot{V}_1 \leq 0$$

Suivant la propriété (4.30), nous obtenons:

$$\dot{V}_2 = 0$$

La loi d'adaptation des paramètres du modèle flou (4.43) donne:

$$\dot{V}_3 = 0$$

En utilisant l'expression des gains du mode de glissement (4.41), nous pouvons écrire:

$$\dot{V}_4 \leq 0 \quad (4.50)$$

Ce qui entraîne:

$$\dot{V} \leq -s^T k_d s \quad (4.51)$$

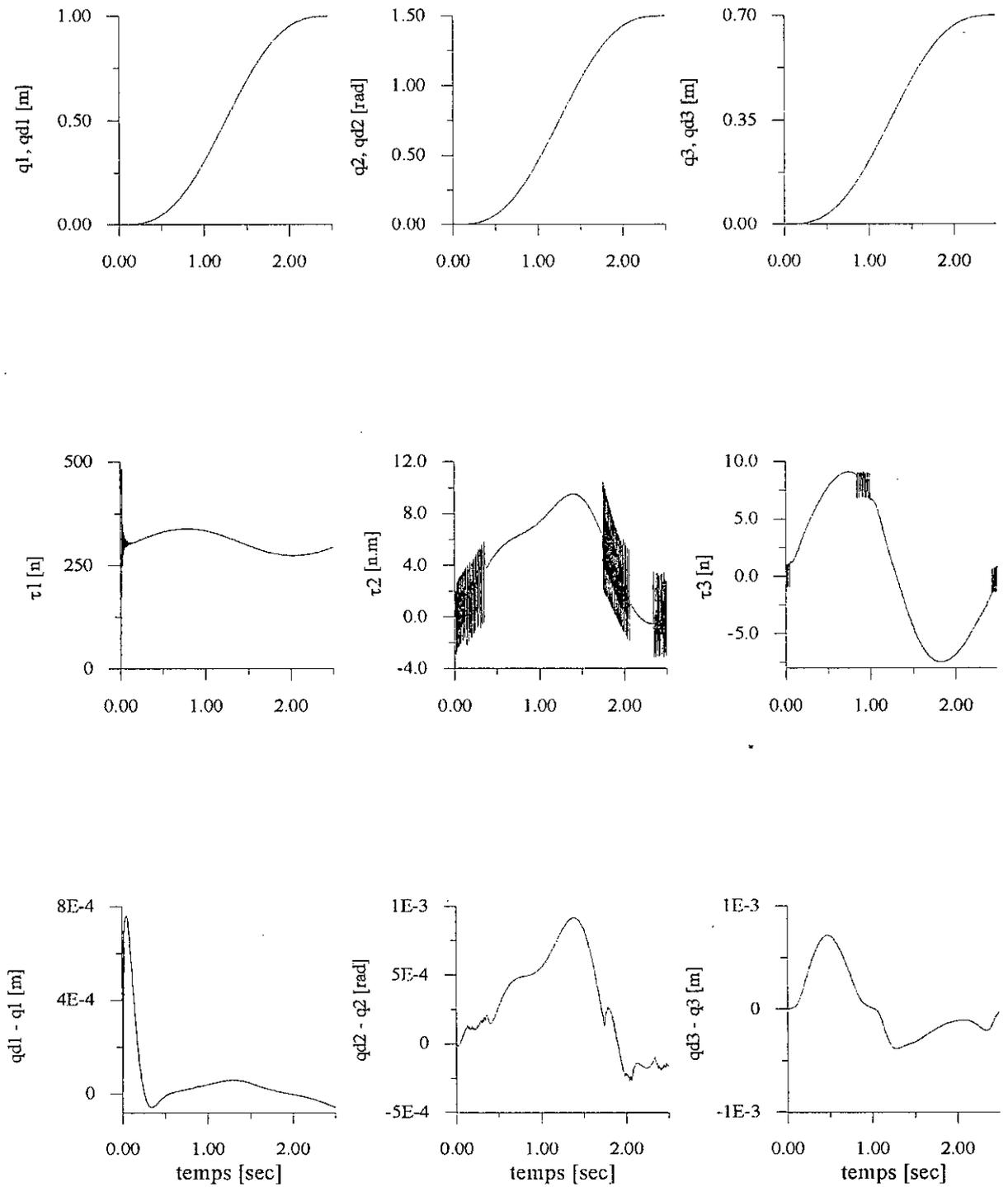
Par conséquent,  $s$  tend vers zéro et d'après (4.32),  $e$  tend vers zéro comme la sortie d'un système linéaire stable du premier ordre à entrée  $s$  nulle [MED97], [POL96], [SLO88], [SLO91].

#### 4.3.5 Résultats de simulation

Pour la validation de la stratégie de commande proposée, une simulation a été réalisée sur le bras manipulateur PORTOS.

Un système flou à huit règles est utilisé pour approximer le modèle inconnu. Les paramètres de synthèse sont choisis comme suit:  $k_d = 10$ ,  $\bar{\varepsilon} = [0.1, 1.5, 1.5]^T$ ,  $\Gamma = \text{diag}(10, \dots, 10)$ . La réponse du robot est illustrée par la figure 4.5. Cette dernière représente l'évolution de la position, de la commande et de l'erreur de poursuite. Nous remarquons, à partir des courbes des erreurs, les bonnes qualités de poursuite.

L'influence d'une variation paramétrique appliquée à l'instant 1.5s, est montrée à la figure 4.6. Dans ce test, tous les paramètres du robot sont supposés augmenter de 50% de leurs valeurs réelles. Nous remarquons que cette perturbation paramétrique n'a pas affecté les performances de poursuite. Toutefois, la grandeur de commande présente un léger effet de chattering.



**Figure 4.5.** Résultats de simulation sans variations paramétriques pour la commande floue adaptative directe stable.

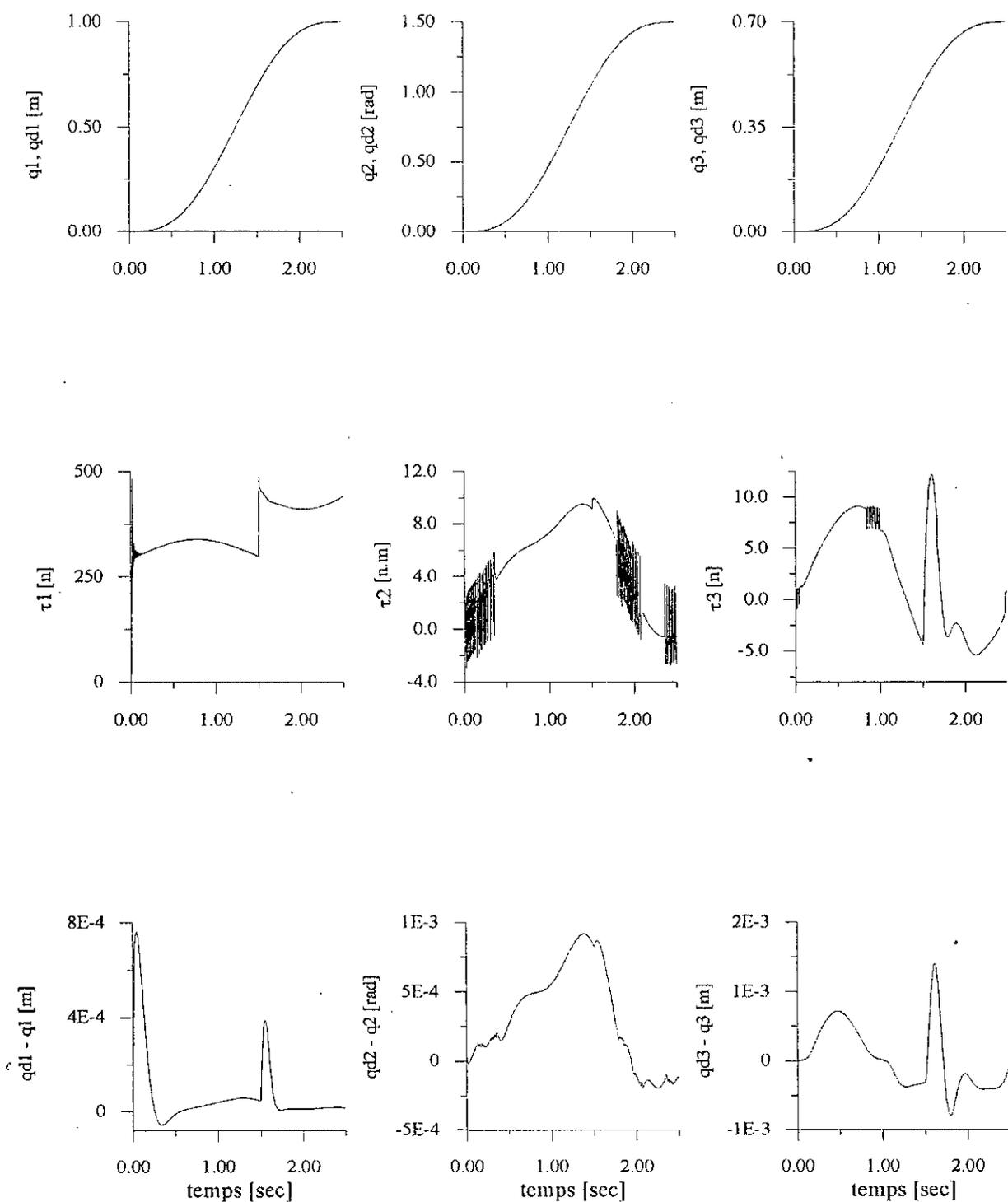


Figure 4.6. Résultats de simulation avec variations paramétriques pour la commande floue adaptative directe stable.

#### 4.4 COMMANDE FLOUE ADAPTATIVE DIRECTE STABLE ETENDUE

La stratégie de commande présentée dans la section précédente est appliquée uniquement aux robots manipulateurs. Dans cette section nous proposons une extension de cette approche à une classe des systèmes non linéaires.

##### 4.4.1 Formulation du problème

Nous considérons la classe des systèmes dynamiques non linéaires MIMO dont l'équation dynamique est écrite sous la forme suivante:

$$F(x) y^{(n)} + G(x) = u \quad (4.52)$$

où

$x \in \mathbb{R}^{m \cdot n}$  est le vecteur d'état,  $u, y \in \mathbb{R}^m$  sont l'entrée et la sortie du système,  $F(x) \in \mathbb{R}^{m \times m}$ ,  $G(x) \in \mathbb{R}^m$  sont des fonctions non linéaires. Nous supposons que  $F(x)$  est une matrice définie positive, telle que sa dérivée temporelle vérifie la condition:

$$\|\dot{F}(x)\| \leq F_0 \|x\| \quad (4.53)$$

avec

$F_0$  est une constante positive connue.

On définit comme suit:

- Le vecteur de l'erreur de poursuite:

$$e_0 = [e \ \dot{e} \ \dots \ e^{(n-1)}]^T \in \mathbb{R}^{m \cdot n}$$

avec  $e = y_d - y$

- L'erreur de poursuite filtrée:

$$s = \left(1 \frac{d}{dt} + \lambda\right)^{n-1} e \quad (4.54)$$

qui peut être écrite sous la forme:

$$s = C e_0$$

où

$$C = [\lambda^{n-1} \ (n-1)\lambda^{n-2} \ \dots \ 1]$$

- Le signal de référence

$$y_r^{(n)} = y_d^{(n)} + C_r e_0 \quad (4.55)$$

avec

$$C_r = [0 \quad \lambda^{n-1} \quad (n-1)\lambda^{n-2} \quad \dots \quad (n-1)\lambda]$$

alors que  $y_d^{(n)}$  est la n-ième dérivée de la trajectoire désirée.

#### 4.4.2 Approximation par les systèmes flous

Les fonctions non linéaires  $F(x)$  et  $G(x)$  de (4.52) sont inconnues, les systèmes flous; approximateurs universels, sont utilisés pour la construction de ces fonctions. Nous supposons qu'il existe des systèmes flous de la forme  $W(x)\theta$ , tel que:

$$F(x) = W_f(x)\theta_f + \varepsilon_f \quad (4.56)$$

$$G(x) = W_g(x)\theta_g + \varepsilon_g \quad (4.57)$$

avec  $\varepsilon_f$  et  $\varepsilon_g$  sont les erreurs de reconstruction des fonctions  $F(x)$  et  $G(x)$  tel que:

$$|\varepsilon_f| \leq \bar{\varepsilon}_f \quad (4.58a)$$

et

$$|\varepsilon_g| \leq \bar{\varepsilon}_g \quad (4.58b)$$

Notons que  $\bar{\varepsilon}_f$  est une matrice et  $\bar{\varepsilon}_g$  est un vecteur.

Les fonctions estimées générées par les systèmes flous sont données par:

$$\hat{F}(x) = W_f(x)\hat{\theta}_f \quad (4.59)$$

$$\hat{G}(x) = W_g(x)\hat{\theta}_g \quad (4.60)$$

Où  $\hat{\theta}_f$  et  $\hat{\theta}_g$  sont les paramètres à calculer par un algorithme d'adaptation approprié.

#### 4.4.3 Synthèse de la commande

A partir des fonctions floues estimées, nous choisissons la loi de commande suivante:

$$u = k_d s + \frac{1}{2} F_0 \|x\| s + W_f(x)\hat{\theta}_f y_r^{(n)} + W_g(x)\hat{\theta}_g + K \operatorname{sgn}(s) \quad (4.61)$$

où  $K$  est le gain du terme du mode de glissement donné par:

$$K \operatorname{sgn}(s) = [k_i \operatorname{sign}(s_i)]^T_{i=1,\dots,n} \quad (4.62)$$

et

$$K = \bar{\varepsilon}_r |y_r^{(n)}| + \bar{\varepsilon}_g$$

Les paramètres du contrôleur flou sont ajustés par la loi d'adaptation suivante:

$$\hat{\theta}_r = \eta_r W_r^T(x) s y_r^{(n)} \quad (4.63)$$

$$\hat{\theta}_g = \eta_g W_g^T(x) s \quad (4.64)$$

où  $\eta_r$  et  $\eta_g$  sont des constantes positives.

La structure de commande adoptée est illustrée par la figure 4.7:

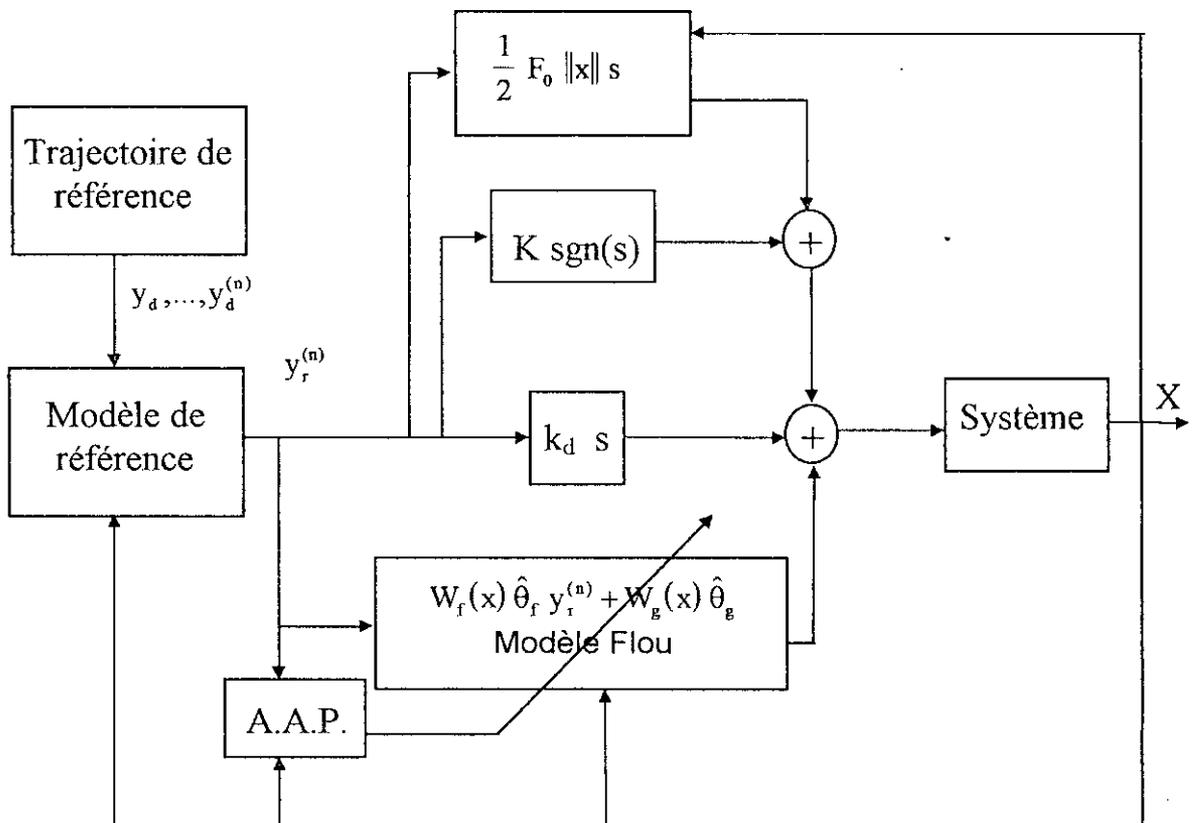


Figure 4.7. Structure de la commande adaptative directe stable étendue.

#### 4.4.4 Analyse de la stabilité

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

$$V = \frac{1}{2} s^T F s + \frac{1}{2} (\tilde{\theta}_f^T \eta_f^{-1} \tilde{\theta}_f) + \frac{1}{2} (\tilde{\theta}_g^T \eta_g^{-1} \tilde{\theta}_g) \quad (4.6)$$

où

$$\tilde{\theta}_f = \hat{\theta}_f - \theta_f \text{ et } \tilde{\theta}_g = \hat{\theta}_g - \theta_g$$

La dynamique de l'erreur filtrée  $s$  en boucle fermée est donnée par:

$$F \dot{s} = -k_d s - \frac{1}{2} F_0 \|x\| s - W_f(x) \tilde{\theta}_f y_r^{(n)} - W_g(x) \tilde{\theta}_g - K \operatorname{sgn}(s) + \varepsilon_f y_r^{(n)} + \varepsilon_g \quad (4.6)$$

En dérivant la fonction de Lyapunov nous obtenons:

$$\dot{V} = \frac{1}{2} s^T \dot{F} s + s^T F \dot{s} + \tilde{\theta}_f^T \eta_f^{-1} \dot{\hat{\theta}}_f + \tilde{\theta}_g^T \eta_g^{-1} \dot{\hat{\theta}}_g \quad (4.6)$$

En utilisant (4.66), il vient:

$$\begin{aligned} \dot{V} = & \frac{1}{2} s^T \dot{F} s - s^T k_d s - \frac{1}{2} s^T F_0 \|x\| s - s^T W_f(x) \tilde{\theta}_f y_r^{(n)} - s^T W_g(x) \tilde{\theta}_g + \\ & \tilde{\theta}_f^T \eta_f^{-1} \dot{\hat{\theta}}_f + \tilde{\theta}_g^T \eta_g^{-1} \dot{\hat{\theta}}_g + s^T \varepsilon_f y_r^{(n)} + s^T \varepsilon_g - s^T K \operatorname{sgn}(s) \end{aligned} \quad (4.6)$$

L'expression finale de  $\dot{V}$  est mise sous la forme suivante:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 + \dot{V}_3 + \dot{V}_4$$

où

$$\dot{V}_1 = -s^T k_d s$$

$$\dot{V}_2 = \frac{1}{2} s^T \dot{F} s - \frac{1}{2} s^T F_0 \|x\| s$$

$$\dot{V}_3 = -s^T W_f(x) \tilde{\theta}_f y_r^{(n)} + \tilde{\theta}_f^T \eta_f^{-1} \dot{\hat{\theta}}_f - s^T W_g(x) \tilde{\theta}_g + \tilde{\theta}_g^T \eta_g^{-1} \dot{\hat{\theta}}_g$$

$$\dot{V}_4 = s^T \varepsilon_f y_r^{(n)} + s^T \varepsilon_g - s^T K \operatorname{sgn}(s)$$

La première composante est négative, car  $k_d$  est une constante positive.

$$\dot{V}_1 \leq 0$$

Suivant la propriété (4.53), il vient:

$$\dot{V}_2 \leq 0$$

En utilisant la loi d'adaptation (4.63) et (4.64), il vient:

$$\dot{V}_3 = 0$$

Selon le choix des gains du mode de glissement (4.62), on a:

$$\dot{V}_4 \leq 0$$

Alors, la dérivée temporelle de la fonction de Lyapunov vérifie:

$$\dot{V} \leq -s^T k_d s \quad (4.69)$$

Nous concluons donc que  $s$  converge asymptotiquement vers zéro et que tous les signaux du système sont bornés. Cependant, comme  $e$  étant la sortie d'un système linéaire stable, il vient que,  $e^{(i)} \rightarrow 0$ ;  $i=0, \dots, n-1$ , [MED97], [SLO91].

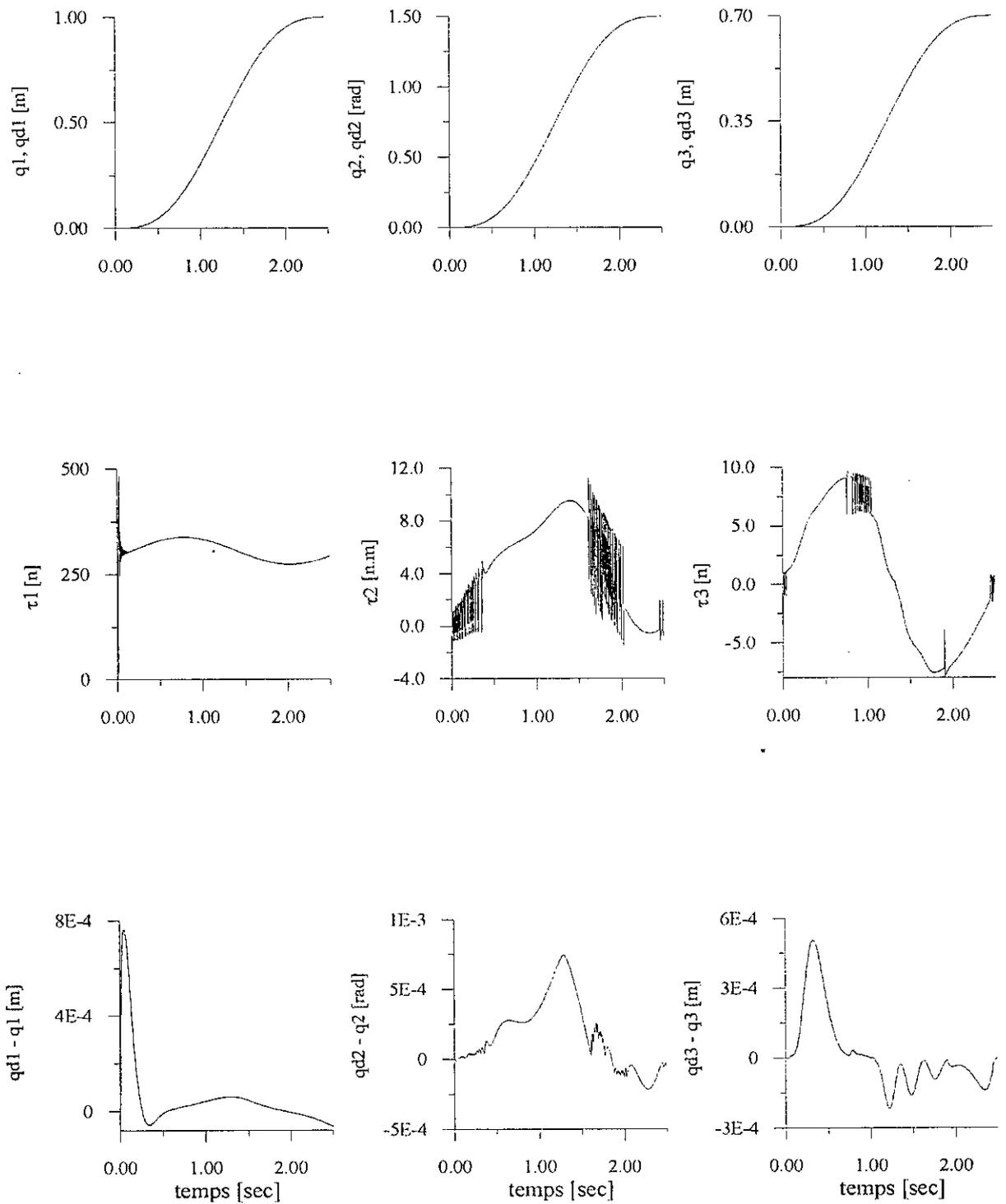
#### 4.4.5 Résultats de simulation

Cette partie présente des résultats de simulation numérique réalisés sur le bras manipulateur PORTOS.

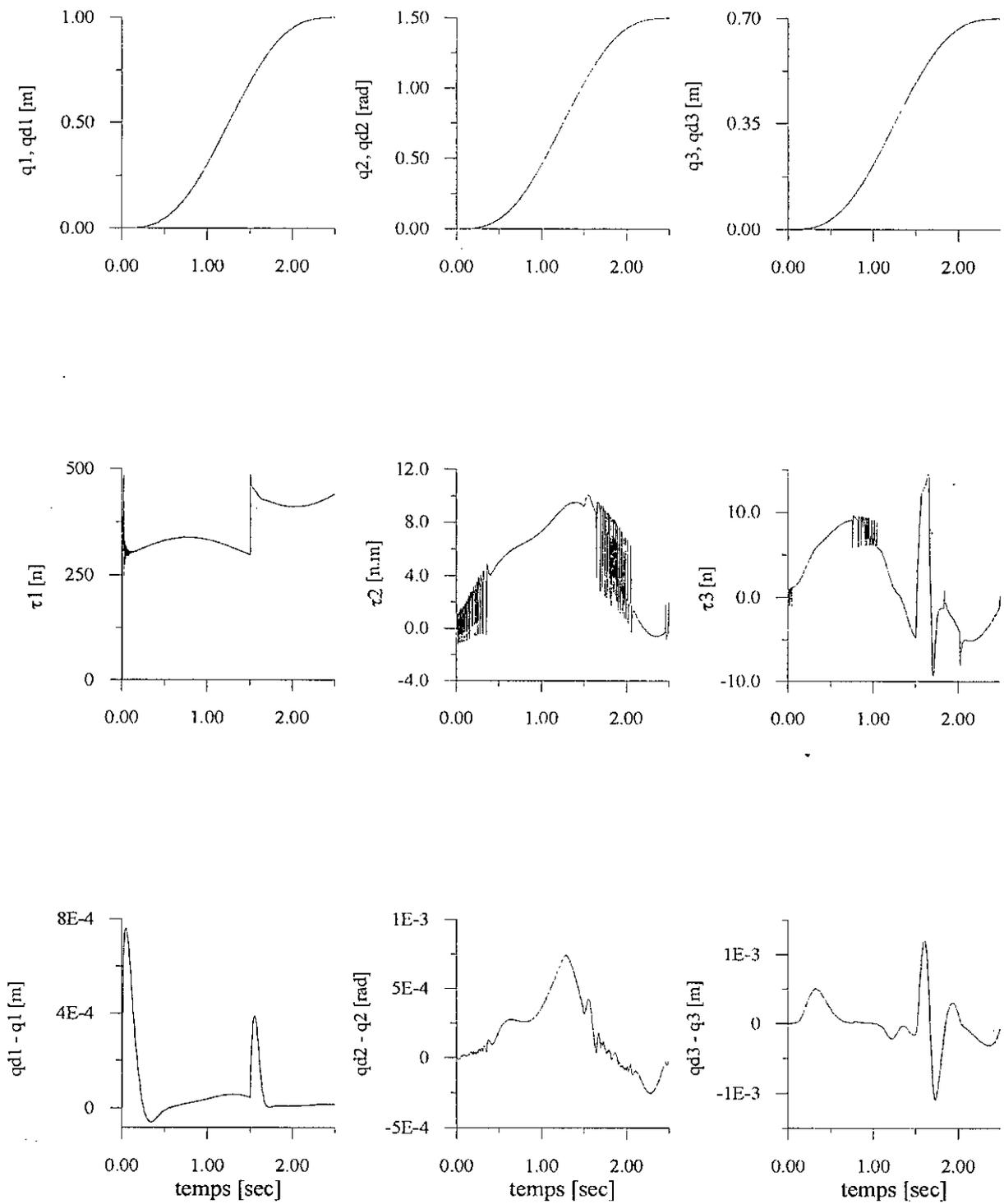
Nous choisissons les mêmes structures des systèmes flous utilisés dans la section précédente. Les paramètres de synthèse sont choisis comme suit:  $k_d = 10$ ,  $\bar{\epsilon}_f = \text{diag}(0.1, 1, 1)$ ,  $\bar{\epsilon}_g = [0.1, 1, 1]^T$ ,  $\eta_f = 10$ ,  $\eta_g = 10$  et  $F_0 = 5$ .

Les réponses du robot sans variation paramétrique sont illustrées par la figure 4.8. Nous constatons qu'il y'a une poursuite parfaite. Cependant, la grandeur de commande présente un léger effet de chattering qui est du à la composante du mode de glissement.

Les réponses du robot manipulateur à une perturbation paramétrique sont présentées dans la figure 4.9. Dans ce test, une variation soudaine de 50% de tous les paramètres du robot est appliquée à l'instant 1.5s. Nous constatons que cette technique de commande est robuste vis-à-vis des variations paramétriques.



**Figure 4.9.** Résultats de simulation sans variations paramétriques pour la commande floue adaptative directe stable étendue.



**Figure 4.10.** Résultats de simulation avec variations paramétriques pour la commande floue adaptative directe stable étendue.

## 4.5 COMMANDE FLOUE ADAPTATIVE DIRECTE STABLE GENERALISEE

Les méthodes de commandes étudiées dans les sections précédentes nécessitent la spécification de quelques paramètres de synthèse telles que les bornes des erreurs de reconstruction. Afin de résoudre de tels problèmes, nous proposons dans cette partie une approche de commande basée sur l'estimation en ligne de ces paramètres.

### 4.5.1 Structure de la commande

Considérons encore la classe des systèmes non linéaires MIMO régie par l'équation dynamique (4.52). De plus, nous supposons que:

- la constante  $F_0$  est inconnue,
- l'erreur de reconstruction vérifie:

$$\begin{aligned} \|\varepsilon_r\| &\leq \bar{\varepsilon}_r \\ \|\varepsilon_g\| &\leq \bar{\varepsilon}_g \end{aligned} \quad (4.70)$$

où les bornes  $\bar{\varepsilon}_r$  et  $\bar{\varepsilon}_g$  sont inconnues.

A partir des fonctions floues estimées, nous choisissons la loi de commande suivante:

$$u = k_d s + \frac{1}{2} \hat{F}_0 \|x\| s + W_f(x) \hat{\theta}_f y_r^{(n)} + W_g(x) \hat{\theta}_g + \hat{k}_f \|y_r^{(n)}\| \operatorname{sgn}(s) + \hat{k}_g \operatorname{sgn}(s) \quad (4.71)$$

où  $\hat{F}_0$ ,  $\hat{k}_f$  et  $\hat{k}_g$  sont des paramètres ajustés par les lois d'adaptations suivantes:

$$\dot{\hat{F}}_0 = \eta \|x\| \|s\| \quad (4.72)$$

$$\dot{\hat{k}}_f = \eta \|y_r^{(n)}\| \|s\| \quad (4.73)$$

$$\dot{\hat{k}}_g = \eta \|s\| \quad (4.74)$$

alors que  $\eta$  est une constante positive contrôlant la vitesse d'adaptation de ces coefficients.

Les paramètres  $\theta_f$  et  $\theta_g$  du correcteur flou sont ajustés par la loi d'adaptation suivante:

$$\dot{\hat{\theta}}_r = \eta_r W_r^T(x) s y_r^{(n)} \tag{4.75}$$

$$\dot{\hat{\theta}}_g = \eta_g W_g^T(x) s \tag{4.76}$$

où  $\eta_r$  et  $\eta_g$  sont des constantes positives.

La structure de commande adoptée est illustrée à la figure 4.10:

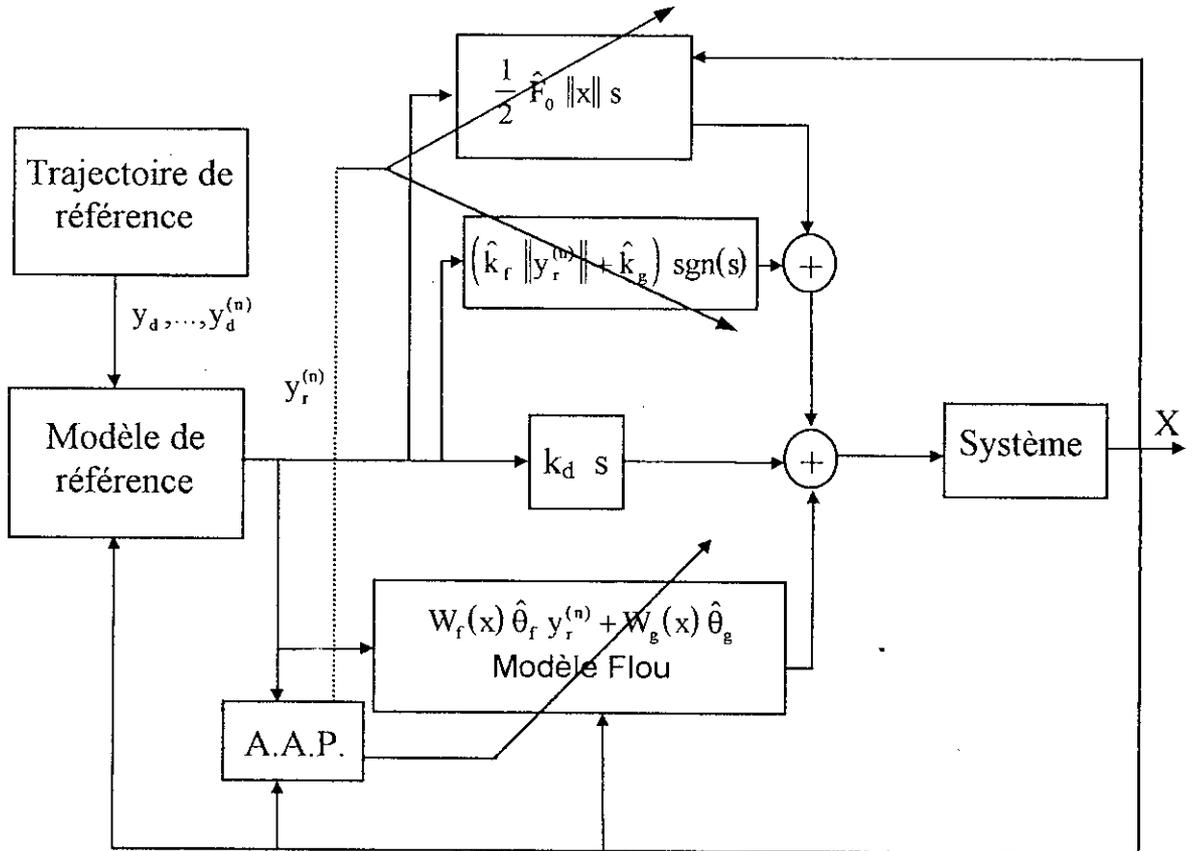


Figure 4.10. Structure de la commande adaptative directe stable généralisée.

### 4.5.2 Analyse de la stabilité

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

$$V = \frac{1}{2} s^T F s + \frac{1}{2} (\tilde{\theta}_r^T \eta_r^{-1} \tilde{\theta}_r) + \frac{1}{2} (\tilde{\theta}_g^T \eta_g^{-1} \tilde{\theta}_g) + \frac{1}{2 \eta_{f_0}} \tilde{F}_0^2 + \frac{1}{2 \eta_r} \tilde{k}_r^2 + \frac{1}{2 \eta_g} \tilde{k}_g^2 \tag{4.77}$$

où

$$\tilde{\theta}_r = \hat{\theta}_r - \theta_r \text{ et } \tilde{\theta}_g = \hat{\theta}_g - \theta_g$$

$$\tilde{F}_0 = \hat{F}_0 - F_0, \quad \tilde{k}_f = \hat{k}_f - k_f \text{ et } \tilde{k}_g = \hat{k}_g - k_g \quad (4.78)$$

alors que

$$k_f = \bar{\varepsilon}_f \text{ et } k_g = \bar{\varepsilon}_g \quad (4.79)$$

La dynamique de l'erreur filtrée  $s$  est donnée par:

$$F \dot{s} = -k_d s - \frac{1}{2} \hat{F}_0 \|x\| s - W_f(x) \tilde{\theta}_f y_r^{(n)} - W_g(x) \tilde{\theta}_g - \left( \hat{k}_f \|y_r^{(n)}\| + \hat{k}_g \right) \text{sgn}(s) + \varepsilon_f y_r^{(n)} + \varepsilon_g \quad (4.80)$$

La dérivée de la fonction de Lyapunov est donnée par:

$$\dot{V} = \frac{1}{2} s^T \dot{F} s + s^T F \dot{s} + \tilde{\theta}_f^T \eta_f^{-1} \dot{\hat{\theta}}_f + \tilde{\theta}_g^T \eta_g^{-1} \dot{\hat{\theta}}_g + \frac{1}{\eta} \tilde{F}_0 \dot{\hat{F}}_0 + \frac{1}{\eta} \tilde{k}_f \dot{\hat{k}}_f + \frac{1}{\eta} \tilde{k}_g \dot{\hat{k}}_g \quad (4.81)$$

En utilisant l'expression de la dynamique de l'erreur filtrée (4.80), la dérivée de  $V$  devient alors:

$$\begin{aligned} \dot{V} = & \frac{1}{2} s^T \dot{F} s - s^T k_d s - \frac{1}{2} s^T \hat{F}_0 \|x\| s - s^T W_f(x) \tilde{\theta}_f y_r^{(n)} - s^T W_g(x) \tilde{\theta}_g + \\ & \tilde{\theta}_f^T \eta_f^{-1} \dot{\hat{\theta}}_f + \tilde{\theta}_g^T \eta_g^{-1} \dot{\hat{\theta}}_g + s^T \varepsilon_f y_r^{(n)} + s^T \varepsilon_g - s^T \left( \hat{k}_f \|y_r^{(n)}\| + \hat{k}_g \right) \text{sgn}(s) \end{aligned} \quad (4.82)$$

L'expression finale de  $\dot{V}$  est mise sous la forme suivante:

$$\dot{V} = \dot{V}_1 + \dot{V}_2 + \dot{V}_3 + \dot{V}_4$$

où

$$\dot{V}_1 = -s^T k_d s$$

$$\dot{V}_2 = \frac{1}{2} s^T \dot{F} s - \frac{1}{2} s^T \hat{F}_0 \|x\| s - \frac{1}{\eta} \tilde{F}_0 \dot{\hat{F}}_0$$

$$\dot{V}_3 = -s^T W_f(x) \tilde{\theta}_f y_r^{(n)} + \tilde{\theta}_f^T \eta_f^{-1} \dot{\hat{\theta}}_f - s^T W_g(x) \tilde{\theta}_g + \tilde{\theta}_g^T \eta_g^{-1} \dot{\hat{\theta}}_g$$

$$\dot{V}_4 = s^T \varepsilon_f y_r^{(n)} + s^T \varepsilon_g - s^T \left( \hat{k}_f \|y_r^{(n)}\| + \hat{k}_g \right) \text{sgn}(s) + \frac{1}{\eta} \tilde{k}_f \dot{\hat{k}}_f + \frac{1}{\eta} \tilde{k}_g \dot{\hat{k}}_g$$

$k_d$  est une constante positive, donc:

$$\dot{V}_1 \leq 0$$

En utilisant la loi d'adaptation (4.72), il vient:

$$\dot{V}_2 \leq 0$$

En utilisant la loi d'adaptation (4.75) et (76), il s'en suit:

$$\dot{V}_3 = 0$$

Selon le choix des lois d'adaptation des gains du mode de glissement (4.73) et (4.74), on a:

$$\dot{V}_4 \leq 0$$

Par conséquent:

$$\dot{V} \leq -s^T k_d s \quad (4.83)$$

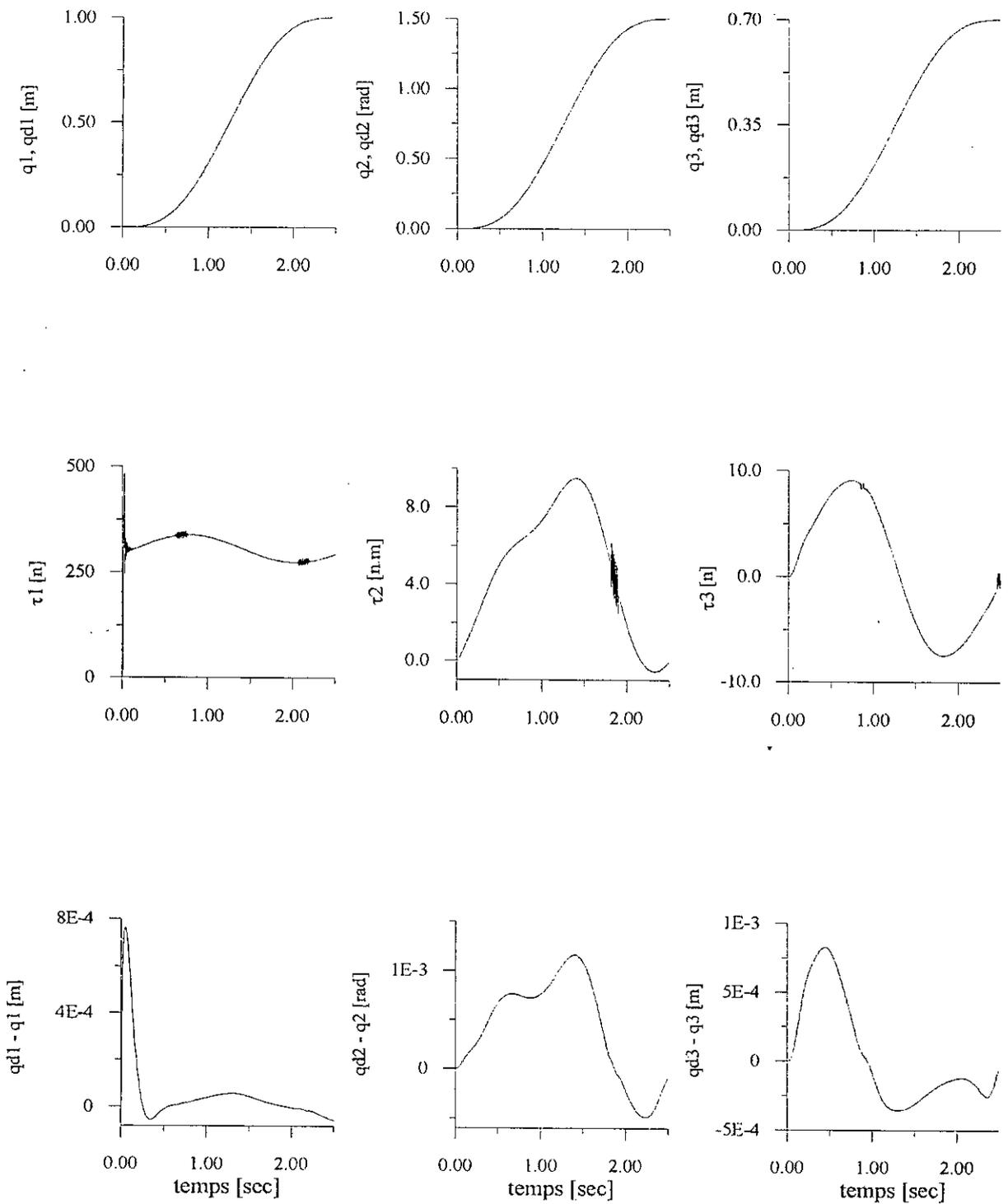
Nous déduisons donc que  $s$  converge asymptotiquement vers zéro et que tous les signaux du système sont bornés. Cependant, comme  $e$  étant la sortie d'un système linéaire stable, il vient que,  $e^{(i)} \rightarrow 0$ ;  $i=0, \dots, n-1$ , [MED97], [SLO91].

### 2.5.3 Résultats de simulation

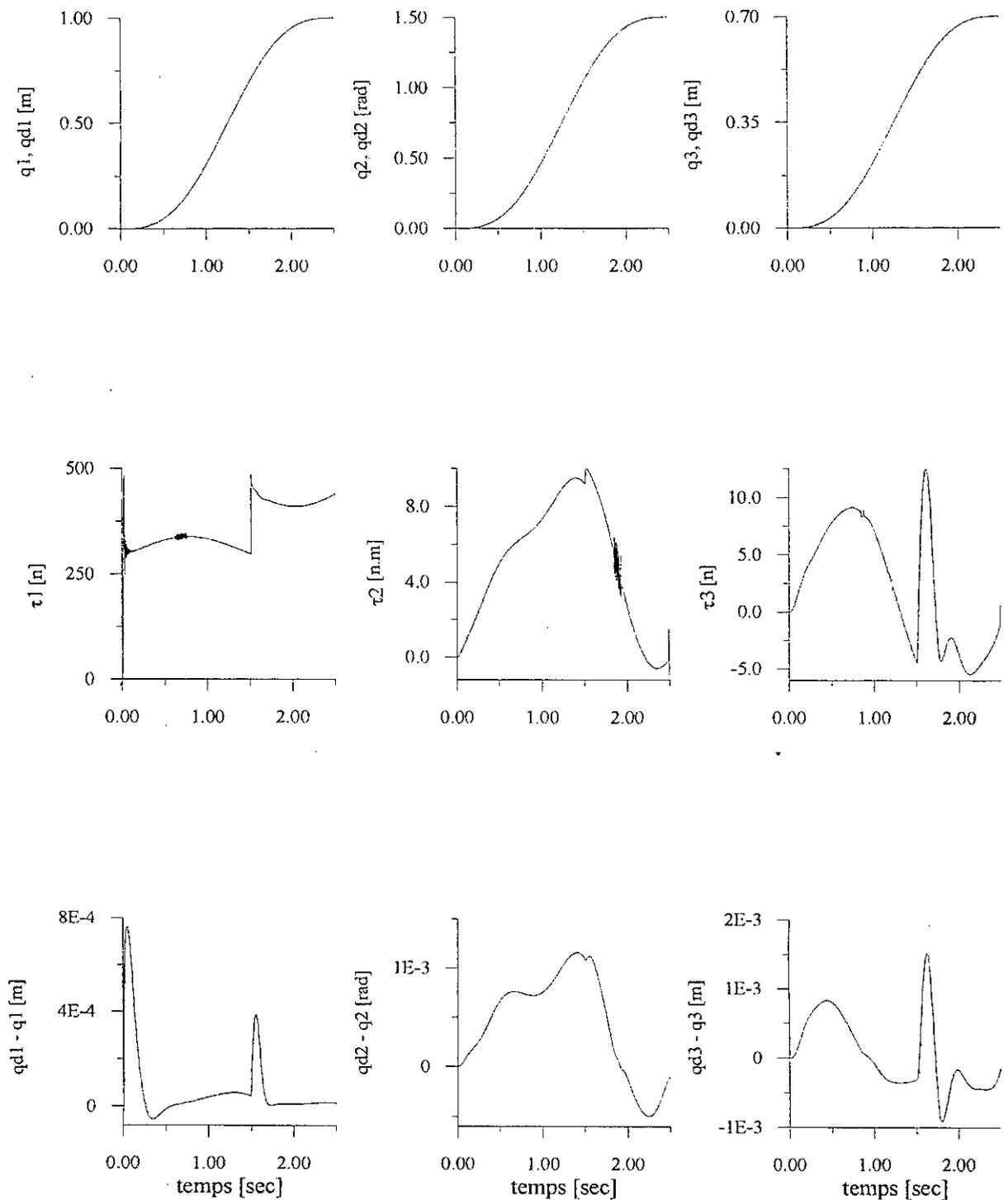
Cette partie présente des résultats de la simulation numérique effectuée sur le robot manipulateur PORTOS.

Le robot manipulateur est modélisé par les mêmes systèmes flous que ceux de la section précédente. Les paramètres de synthèse sont choisis comme suit:  $k_d = 10$ ,  $\eta_f = 10$ ,  $\eta_g = 10$  et  $\eta = 1$ . La réponse du robot sans variation paramétrique est illustrée par la figure 4.11. Ces résultats montrent que l'algorithme d'adaptation est efficace. Nous constatons qu'il y'a une amélioration au niveau de la grandeur de commande. En effet, nous remarquons que l'effet de chattering est devenu presque nul.

Pour tester la capacité d'adaptation de cette stratégie en présence des variations paramétriques, une perturbation paramétrique brusque de 50% est appliquée à l'instant 1.5s. La réponse à une telle situation est donnée en figure 4.12. Nous constatons la robustesse de la méthode vis-à-vis des variations paramétriques.



**Figure 4.11.** Résultats de simulation sans variations paramétriques pour la commande floue adaptative directe stable généralisée.



**Figure 4.12.** Résultats de simulation avec variations paramétriques pour la commande floue adaptative directe stable généralisée.

## 4.6 CONCLUSION

Dans ce chapitre, nous avons investi les possibilités offertes, par les systèmes flous de type Sugeno, en termes de capacité d'approximation des fonctions non linéaires continues, afin de développer des stratégies de commande adaptatives directes pour les robots manipulateurs rigides. Ainsi, nous nous sommes intéressés à la commande floue adaptative linéarisante, la commande floue adaptative directe stable, la commande floue adaptative directe stable étendue et à la commande floue adaptative directe stable généralisée.

L'idée conductrice dans notre analyse repose sur l'utilisation des systèmes flous pour la construction des correcteurs adaptatifs, puisque, le modèle du robot est supposé inconnu. De cette façon, le problème est transformé à un problème de commande adaptative robuste. Le problème de l'erreur de reconstruction, du à la modélisation du robot par des systèmes flous, est rigoureusement étudié.

En effet, pour l'étude de la stabilité, d'une part nous avons utilisé une paramétrisation linéaire des systèmes flous et d'autre part, nous avons utilisé des outils théoriques puissants, à savoir la théorie de stabilité de Lyapunov, la théorie de l'approximation et celle des modes glissants. Nous avons constaté que le choix de la fonction de Lyapunov et du terme de mode de glissement sont d'une importance primordiale pour assurer la convergence de l'erreur de poursuite.

Les stratégies de commande proposées ont prouvé leur efficacité à travers les performances enregistrées tout au long des simulations numériques effectuées sur le robot manipulateur PORTOS. Cette efficacité se traduit par des faibles erreurs de poursuite et par une bornitude de la grandeur de commande. Néanmoins, ces techniques de commande présentent un effet de chattering sur la grandeur de commande. En effet, nous constatons que la grandeur de commande de la dernière approche est la plus lisse comparée à celle des autres techniques. De surcroît, cette étude nous a permis de formuler les remarques suivantes:

- Dans les trois premières approches de commande, le choix du gain du terme de glissement est intimement lié aux bornes des erreurs de reconstruction. Cependant, il faut noter que ces bornes sont choisies par tâtonnement. Ainsi, un compromis doit être fait entre la taille des systèmes flous utilisés et le terme du mode de glissement. En effet, plus la taille est petite, plus est important le terme du mode de glissement; ce qui se répercute par un effet de chattering important sur la commande. En revanche, si la taille est grande, l'algorithme de commande devient très lent.
- Le problème du choix des bornes de l'erreur de reconstruction est résolu dans la dernière approche par la modification de la loi de commande et de la fonction de Lyapunov. Néanmoins, nous constatons que le gain de glissement augmente tant que l'erreur de poursuite n'est pas nulle.
- La commande floue adaptative linéarisante présente un inconvénient majeur par rapport aux autres approches. En effet, elle nécessite la mesure de l'accélération réelle des liaisons du robot ainsi que l'inversion de la matrice d'inertie estimée.

# ***CONCLUSION GENERALE***

Dans ces travaux, nous avons étudié et développé des stratégies de commande adaptatives utilisant les systèmes flous, appliquées à la commande en poursuite d'une trajectoire de référence des robots manipulateurs. Ainsi, nous nous sommes intéressés à la commande adaptative basée sur la technique d'inversion itérative des systèmes flous, à la commande adaptative indirecte et à la commande adaptative directe

Dans le premier chapitre, une large recherche bibliographique a été faite dont le but d'étudier les systèmes flous en qualité d'approximateur universel des fonctions non linéaires. A l'issue de cette recherche nous avons constaté que ces systèmes sont parfois difficiles à construire. En effet, la détermination d'une architecture optimale par des méthodes systématiques précises n'est pas encore bien établie. Cependant, ils sont des systèmes non linéaires adaptatifs, capables de manipuler non seulement des données numériques mais également des informations linguistiques. En automatique, l'utilisation des systèmes flous comme éléments de base pour la modélisation et la commande des systèmes nous épargne, dans la majorité des cas, de la connaissance détaillée du modèle du système, et on se contente souvent de son comportement entrée-sortie.

Dans le deuxième chapitre, nous avons développé une technique de commande adaptative basée sur l'inversion itérative des systèmes flous. Cette technique utilise un modèle flou pour identifier en ligne la dynamique directe de chacune des liaisons du robot, combiné avec la technique d'inversion itérative pour déterminer la valeur de la commande. La technique d'inversion itérative profite de l'avantage de dualité entre les paramètres et les entrées du système flou pour donner les expressions du calcul de la commande. Les paramètres du modèle flou sont ajustés par l'algorithme du gradient modifié. Cependant, il faut noter que cette approche présente deux inconvénients majeurs. D'une part, elle utilise une procédure itérative pour le calcul de la commande, aussi la recherche peut prendre beaucoup de temps avant de converger vers une solution satisfaisante, d'autre part elle nécessite la mesure de l'accélération réelle des articulations; or les robots sont équipés au mieux de capteurs de position et de vitesse. Le calcul numérique de l'accélération peut conduire à des résultats néfastes dans le cas des mesures bruitées.

Le troisième chapitre a été consacré au développement de quatre stratégies de commande adaptatives indirectes. Les deux premières techniques proposées sont basées respectivement sur l'identification de la dynamique inverse et directe de chaque articulation du robot. La loi de commande, dans ces deux techniques, est un système flou dont les règles sont

construites à partir de celles du modèle flou. Ces deux approches souffrent également du problème de la mesure de l'accélération des liaisons. Pour pallier à cet inconvénient, nous avons proposé, comme première alternative, une stratégie d'identification du modèle dynamique du robot en minimisant l'erreur en vitesse ainsi qu'une seconde alternative basée sur la minimisation de l'erreur filtrée. Dans ces deux techniques de commande, la théorie de l'approximation et celle de Lyapunov sont investies pour établir une loi d'adaptation paramétrique assurant la bornitude de l'erreur d'équation et des paramètres des systèmes flous

Les stratégies de commande étudiées dans le deuxième et le troisième chapitre ne garantissent pas la convergence vers zéro de l'erreur de poursuite et nécessitent un temps de calcul relativement grand; vu qu'elles procèdent en deux étapes pour le calcul de la commande.

Le dernier chapitre est consacré au développement de techniques de commande garantissant la convergence asymptotique de l'erreur de poursuite. Pour atteindre cet objectif, nous avons utilisé des théories puissantes telles que la théorie de l'approximation, la théorie de Lyapunov et celle des modes glissants. La commande adaptative linéarisante par les systèmes flous était la première stratégie proposée. Nous avons constaté que cette approche nécessite la mesure de l'accélération du robot d'une part, et l'inversion de la matrice d'inertie estimée d'autre part. Dans le but de pallier à ces inconvénients, la commande adaptative directe stable a été proposée. Néanmoins, cette technique impose une condition sur le modèle dynamique du système commandé. Afin d'étendre l'utilisation de cette approche de commande à d'autre classe des systèmes non linéaires, nous avons développé la commande appelée la technique de commande floue adaptative directe stable étendue. Ces trois dernières méthodes de commande nécessitent la spécification de certains paramètres de synthèse telles que les bornes de l'erreur de reconstruction. Le choix de ces paramètres est subjectif. Ainsi, pour surmonter ce problème, une dernière technique de commande a été proposée sous le nom de commande floue adaptative directe stable généralisée. Cette technique est plus générale que la troisième car, elle ne nécessite pas la spécification des termes de majoration et des bornes des erreurs de reconstruction. En effet, elle utilise un mécanisme d'adaptation pour estimer ces paramètres en ligne.

La simulation effectuée sur le robot PORTOS montre l'efficacité des systèmes flous utilisés dans la commande et la modélisation des robots manipulateurs. En effet, nous constatons que les différentes commandes conduisent à de faibles erreurs de poursuite. De plus, les résultats montrent que les techniques de commande adoptées sont robustes par rapport aux variations paramétriques du système. Cependant, nous remarquons que les commandes dotées d'un terme de mode de glissement présentent un léger effet de chattering sur la grandeur de commande. L'utilisation des systèmes flous comme éléments de base pour l'approximation du modèle dynamique fortement non linéaire du bras manipulateur a conduit à une amélioration remarquable des performances de la commande adaptative.

Les techniques de commande adaptatives proposées dans ce mémoire sont particulièrement puissantes par rapport aux approches classiques. D'une part, elles ne requièrent pas un modèle de connaissance du système commandé d'autre part, elles peuvent être vues comme des contrôleurs universels des robots manipulateurs rigides. D'ailleurs, elles ont un caractère général car, on peut remplacer le système flou de Sugeno par d'autres types de systèmes flous ou de réseaux neuronaux. De plus, elles permettent également d'incorporer des connaissances linguistiques sur le système commandé.

A l'issue de ces travaux, cette thèse ouvre de nouvelles perspectives de recherche parmi lesquelles nous citons:

- Application de ces techniques de commande à d'autres types de systèmes non linéaires, tels que les robots à jonctions flexibles, les robots flexibles, les machines électriques et les systèmes chimiques, ... etc.
- Commande supervisée par les systèmes flous.
- Commande adaptative décentralisée stable par les systèmes flous.
- Développement de méthodes d'optimisation des systèmes flous évolutifs.
- Utilisation des algorithmes génétiques pour l'optimisation des systèmes flous.

# ANNEXE

## 1. PRESENTATION DU ROBOT PORTOS

Dans cette étude, nous avons utilisé un robot manipulateur de classe quatre appelé PORTOS [GUE95]. Ce robot est caractérisé par une articulation rotationnelle  $q_2$  et deux articulations translationnelles  $q_1$  et  $q_3$ .

Le schéma du robot PORTOS est donné par la figure suivante:

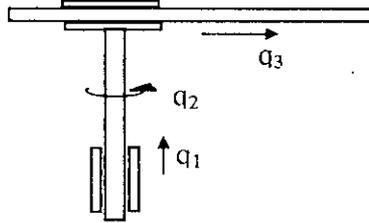


Figure An.1. Schéma simplifié du robot PORTOS.

## 2. Modèle dynamique du robot manipulateur PORTOS

Les formalismes de Newton-Euler ou de Lagrange peuvent être utilisés indifféremment pour obtenir le modèle mathématique d'un robot manipulateur. Le modèle du robot écrit sous une forme compacte est donné par [SLO88], [SLO91]:

$$\tau = H(q) \ddot{q} + C(q, \dot{q}) \dot{q} + E(q, \dot{q}) \quad (\text{An.1})$$

où

$q, \dot{q}$  et  $\ddot{q}$ : vecteur des coordonnées articulaires du robot et ses dérivées.

$\tau$ : vecteurs des commandes.

$H(q)$ : matrice d'inertie symétrique et définie positive.

$C(q, \dot{q})\dot{q}$ : vecteur regroupant les forces de coriolis et les forces centrifuges.

$E(q, \dot{q})$ : vecteur regroupant les forces de gravité et les forces de frottement.

Le modèle dynamique de ce robot est donné par les équations suivantes [GUE95]:

$$\begin{cases} \tau_1 = (m_1 + m_3) \ddot{q}_1 + f_1 \dot{q}_1 + (m_1 + m_3) g \\ \tau_2 = m_3 \left( \frac{l_2^2}{3} - l_2 q_3 + q_3^2 \right) \ddot{q}_2 + 2 m_3 \left( q_3 - \frac{l_2}{2} \right) \dot{q}_2 \dot{q}_3 + f_2 \dot{q}_2 \\ \tau_3 = m_3 \ddot{q}_3 - m_3 \left( q_3 - \frac{l_2}{2} \right) \dot{q}_2^2 + f_3 \dot{q}_3 \end{cases} \quad (\text{An.2})$$

Les paramètres du robot sont [GUE95]:  $m_1 = 20\text{Kg}$ ,  $m_3 = 10\text{Kg}$ ,  $l_2 = 0.75\text{m}$ ,  $f_1 = 30\text{Nsm}^{-1}$ ,  $f_2 = 7.825\text{Nmrad}^{-1}$ ,  $f_3 = 20\text{Nsm}^{-1}$ .

# REFERENCES BIBLIOGRAPHIQUES

- [AST89] K. J. Aström and H. Wittenmark, *Adaptive Control*. Addison-wesley, Reading, MA, 1989.
- [BOI89] J. D. Boissonnat, B. Faverjon and J. P. Merlet, *Techniques de la Robotique: Architectures et Commandes-Tome 1. Traité des nouvelles Technologies, Serie Robotique*, Hermes, Paris, 1989.
- [BUH94] H. Buhler, *Le réglage par logique floue*. Presses Polytechniques Rommandes, 1994.
- [DAN89] B. D'Andrea, *Commande non Linéaire des Robots. Traité des Nouvelles Technologies, Serie Automatique*, Paris, 1989.
- [FRE92] J. A. Freeman and D. M. Skapura, *Neural Networks*. Addison-Wisley Publishing Company, Inc, 1992.
- [FUK92] T. Fukuda and T. Shibata, "Theory and applications of neural networks for industrial control systems," *IEEE Trans. Ind. Electr.*, vol. 39, no. 6, pp. 472-489, Dec. 1992.
- [GIL88] J. C. Gille, P. Decaulne and M. Pélegrin, *Systèmes asservis non linéaires*, Dunod, 1988.
- [GOO84] G. C. Goodwin and K. S. Sin, *Adaptive Filtring Prediction and Control*. Englewood Cliffs, NJ: Prentice Hall, 1984.
- [GUE95] L. Guenfaf, "Etude de differentes strategies de commande adaptative: application à un robot manipulateur," Thèse de Magister, ENP, 1995.
- [HAM97] B. Hamzi, S. Labiod, F. Boudjema and D. Boukhetala, "Supervised control of an inverted pendulum by using dynamic neural networks," *Proc. of the 3rd CSS'97*, vol. 3, pp. 125-129, Belfort, France, Mai 1997.
- [HEN97] M. M. Heniche, "Sur l'utilisation des réseaux de neurones artificiels et des systèmes flous pour la linéarisation et la commande de processus chimiques non linéaires," Thèse de Magister, ENP, 1997.
- [HOR92] S. I. Horikawa, T. Furuhashi and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with back-propagation algorithm," *IEEE Trans. Neural Net.*, vol. 3, no.5, pp. 801-806, Sep. 1992.
- [HOS92] D. A. Hoskin, J. N. Hwang, and J. Vagners, "Iterative inversion of neural networks and its application to adaptive control," *IEEE Trans. Neural Net.*, vol. 3, no. 2, pp. 292-301, Mar. 1992.

- [HUN91] K. J. Hunt and D. Sbarbaro, "Neural networks for nonlinear internal control," *Proc. IEE*, Pt. D, vol. 138, pp. 431-438, 1991.
- [HUN92] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P.J. Gawthrop, "Neural networks for control systems: A survey," *Automatica*, vol. 28, pp. 1083-1112, 1992.
- [ISI91] A. Isidori, *Nonlinear Control Systems*. New York, Springer Verlag, 1989.
- [JAG96] S. Jagannathan and F. L. Lewis, "Discrete-time neural net controller for a class of nonlinear dynamical systems," *IEEE Trans. Aut. Contr.*, vol. 41, no. 11, pp. 1693-1699, Nov. 1996.
- [JAN91] J. S. Jang, "Rule extraction using generalized neural networks," *Proc. 4th IFSA World Congress*, pp. 82-86, July 1991.
- [JAN92] J. S. R. Jang, "Self-learning fuzzy controllers based on temporel backpropagation," *IEEE Trans. Neural Networks*, vol. 3, pp. 714-723, Sept. 1992.
- [JAN95] J. S. R. Jang and C. T. Sun, "Neuro-Fuzzy modeling and control," *Proc. IEEE*, vol. 83, no. 3, pp. 378-406, Mar. 1995.
- [KHA96] H. K. Khalil, "Adaptive output feedback control of nonlinear systems represented by input-output models," *IEEE Trans. Automat. Contr.*, vol. 41, no. 2, pp. 177-188, Feb. 1996.
- [KOS92] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach*. Englewood Cliffs, NJ: Prentice Hall, 1992.
- [LAB97a] S. Labiod, M. Loudini, M. S. Boucherit et H. Chekireb, "Commande floue adaptative d'un robot PUMA," *Actes des Deuxièmes Journées Scientifiques et Techniques de l'INH (JST'97)*, vol. 3, pp. 370-373, Boumerdes, 5 & 6 Mai 1997.
- [LAB97b] S. Labiod, H. Chekireb et M. S. Boucherit, "Commande floue adaptative d'un robot manipulateur," *Proc. des 17èmes Journées Tunisiennes d'Electrotechnique et d'Automatique (JTEA '97)*, Nabeul, Tunisie, 3 & 4 Novembre 1997.
- [LAB98] S. Labiod, H. Chekireb et M. S. Boucherit, "Commande floue adaptative indirecte d'un robot manipulateur," *Acceptée au CNP '98*, Tizi-ouzou, 30 & 31 Mai 1998.
- [LEE90] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller-Part I and Part II," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 2, pp. 404-435, Mar./Apr. 1990.
- [LEU96] F. H. F. Leung, L. K. Wong and P. K. S. Tam, "Fuzzy model controller for inverted pendulum," *Electronics Letters*, vol. 32, no. 18, pp. 1683-1685, August 1996.
- [LIN91] C. T. Lin and C. S. G. LEE, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Computers*, vol. 40, no. 12, pp. 1320-1336, 1991.
- [LJU87] L. Ljung, *System identification: Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
- [MAM77] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Trans. Computers*, vol. 26, no. 12, pp. 1182-1191, Dec. 1977.

- [MED97] D. Y. Meddah and A. Benallegue, "A robust nonlinear controller using neural networks," *Proc. of the 3rd SICICA'97*, Jun. 1997.
- [MEN95] J. M. Mendel, "Fuzzy systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, no. 3, pp. 347-377, Mar. 1995.
- [MIL92] W. T. Miller, R. S. Sutton and P. T. Werbos, *Neural networks for control*. MIT Press, 1992.
- [NAR87] K. S. Narandra and A. M. Annaswamy, "A new adaptive law for robust adaptation without persistent excitation," *IEEE Trans. Automat. Contr.*, vol. AC-32, no. 2, pp. 134-145, Feb. 1987.
- [NAR90] K. S. Narandra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4-27, Mar. 1990.
- [NAR91] K. S. Narandra and K. Parthasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 252-262, 1991.
- [PAU81] R. P. Paul, *Robot Manipulators: Mathematics, Programming and Control*. Cambridge, MA, MIT Press, 1981.
- [PED93] W. Pedrycz, *Fuzzy control and fuzzy systems*, Wiley, New York, 1993.
- [POL96] M. M. Polycarpou, "Stable adaptive neural control scheme for nonlinear systems," *IEEE Trans. Aut. Contr.*, vol. 41, no. 3, pp. 447-451, Mar. 1996.
- [PRO78] T. J. Procyk and E. H. Mamdani, "A linguistic self-organizing process controller," *Automatica*, vol. 15, pp. 15-30, 1978.
- [PSA88] D. Psaltis, A. Sederis, and A. Yamamura, "A multilayer neural network controller," *IEEE Control Syst. Magazine*, vol. 8, no. 4, pp. 17-21, Apr. 1988.
- [RIV95] I. Rivals, "Modélisation et commande de processus par réseaux de neurones; application au pilotage d'un véhicule autonome," Thèse de Doctorat de l'Université Paris 6. 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 Sig. Processing*, vol. 8, pp. 223-236, 1994.
- [SAN92] R. M. Sanner and J. J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol. 3, pp. 837-863, Mar. 1990.
- [SCH85] E. M. Scharf and N. T. Mandic, "The application of a fuzzy controller to the control of a multi-degree-of-freedom robot arm," in: *M. Sugeno. Ed., Industrial Applications of fuzzy control* (North-Holland, Amsterdam, 1985).
- [SER89] H. Seraji, "Decentralized adaptive control of manipulators: Theory, simulation, and experimentation," *IEEE Trans. Automat. Contr.*, vol. 5, no. 2, pp. 183-201, Apr. 1989.
- [SLO88] J. J. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Trans. Automat. Contr.*, vol. 33, no. 11, pp. 995-1002, NOV. 1988.

- [SLO91] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice Hall, 1991.
- [SUG85] M. Sugeno and G. T. Kang, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. 15, pp. 116-132, 1985.
- [SUG88a] M. Sugeno and G. T. Kang, "Structure identification of fuzzy models," *Fuzzy Sets and Systems*, vol. 28, pp. 15-33, 1988.
- [SUG88b] M. Sugeno, "Special issue on fuzzy control," *Fuzzy Sets and Systems*, vol. 26, pp. 149-223, 1988.
- [TAN88] R. Tanscheit and E. M. Scharf, "Experiments with the use of a rule-based self-organizing controller for robotics applications," *Fuzzy Sets and Syst.*, vol. 26, pp. 195-214, 1988.
- [TON84] R. M. Tong, "A retrospective view of fuzzy control systems," *Fuzzy Sets and Systems*, vol. 14, pp. 199-210, 1984.
- [VIB87] C. Vibet, *Robots: Principes et contrôle*, Ellipse, Paris, 1987.
- [WAN92a] L. X. Wang and J. M. Mendel, "Generating fuzzy rules from numerical data, with applications," *IEEE Trans. Syst., Man, and Cybernetics*, vol. SMC-22, pp. 1414-1427, 1992.
- [WAN92b] L. X. Wang and J. M. Mendel, "Back-propagation fuzzy systems as nonlinear dynamic system identifiers," *Proc. IEEE Int. Conf. on Fuzzy Systems*, pp. 1409-1418, San Diego, CA, 1992.
- [WAN95] L. X. Wang, "Design and analysis of fuzzy identifiers of nonlinear dynamic systems," *IEEE Trans. Automat. Contr.*, vol. 40, no. 1, Jan. 1995.
- [WON96] K. W. Wong, C. S. Leung, P. F. Sun and L. W. Chan, "On-line training and pruning for recursive least squares algorithms," *Electronics Letters*, vol. 32, no. 23, pp. 2152-2153, Aug. 1996.