

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

Ecole Nationale Polytechnique

THESE

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

Présentée au Laboratoire de Commande des Processus
DER de Génie Electrique et Informatique
Pour l'obtention du titre

MAGISTER

EN AUTOMATIQUE

Par

TIDJANI Zakaria

Ingénieur d'état en Automatique de l'E.N.P

THEME

**COMMANDE DECENTRALISEE FLOUE
APPLIQUEE AUX ROBOTS MANIPULATEURS**

Soutenue publiquement en 1999 devant la commission d'examen :

**F. BOUDJEMA
D. BOUKHETALA
M. S. BOUCHERIT
E. M. BERKOUK
R. ILLOUL
M. TADJINE**

**Président
Rapporteur
Rapporteur
Examinateur
Examinateur
Examinateur**

Thèse préparée au L.C.P.- DER de Génie Electrique et Informatique
E.N.P, 10, Avenue Hassen Badi, 16200 El-Harrach, Alger

Avant-Propos

Ce travail a été effectué au sein du Laboratoire de Commande des Processus (L.C.P) de la DER Génie Electrique et Informatique de l'Ecole Nationale Polytechnique (E.N.P) sous la direction conjointe de Monsieur **M. S. BOUCHERIT**, Maître de conférence à l'E.N.P et Monsieur **D. BOUKHETALA**, Chargé de cours à l'E.N.P, qu'ils trouvent ici ma profonde gratitude pour leur aides, leurs conseils et l'encouragement continus qu'ils m'ont apportés tous au long de ce travail.

Monsieur **F. BOUDJEMA**, Maître de conférence au département Automatique me fait l'honneur de présider le jury d'examen de ma thèse je l'en remercie vivement mais aussi pour sa bienveillance, ses conseils et ses encouragements et pour toute l'attention qu'il a eu à mon égard.

J'exprime ma gratitude à Monsieur **E. M. BERKOUK**, Docteur à l'E.N.P pour l'honneur qu'il ma fait en acceptant de faire partie de ce jury.

Mes vifs remerciements vont aussi à Monsieur **R. ILLOUL**, Chargé de cours à l'E.N.P pour l'intérêt qu'il a porté à ce travail en me faisant l'honneur de participer à ce jury.

Je remercie également, monsieur **M. TADJINE**, Docteur à l'E.N.P, dont je suis très honoré de sa présence dans la jury de cette thèse.

A **A. CHEMSA**, **A. SERRAYE** et **A. ABIMILOUD**, je ne serai exprimer assez ma gratitude pour leur soutien inconditionnel qui a amené à l'achèvement de ce travail.

Que mes amis, **A.-R. TIDJANI**, **A.-K. TIDJANI**, **Om. TIDJANI**, **Ou. TIDJANI**, **S. BENDENYA**, **F. BELLAZREG**, **A.-B. BOUBEKEUR**, **A. SALEM**, **Y. BENYAHIA**, **R. ZOUBEIDI**, **D. OUAREGLI**, **A.-R. ROUGHI**, **Y. KEBDI**, **A. BERRACHED**, **T. MADANI**, **S. LABIOD**, **O. BOUHALI**, trouvent ici toute ma gratitude pour leur encouragements permanentes.

Que toute la famille présente à l'université, particulièrement **A.-E. TIDJANI**, **M. TIDJANI**, et **A. TIDJANI**, **T. TIDJANI**, et la grande famille de **BOURAOU** et de **TEMACINE**, ainsi que tous les ingénieurs collègues de la promotion Automatique 95, je tiens à adresser mes vifs remerciements.

A tous ceux qui ont contribué de près ou de loin à l'achèvement de ce travail, je les remercie.

Enfin, je dois certainement beaucoup à l'ensemble de mes enseignants, qu'il trouve ici ma profonde reconnaissance.

Zakaria TIDJANI

Dédicaces

Je dédie ce modeste travail :

A la mémoire de *mon père*

A *ma très chère mère*

A tous mes frères : *Nadjib, El-Hadj et Saouli*

A toutes *mes sœurs*

Sommaire

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Chapitre 0 : Introduction générale	1
Chapitre I : Systèmes et régulateurs à logique floue	
I.1 Introduction	5
I.2 Logique floue	5
I.2.1 Ensembles flous	6
I.2.2 Opérations sur les ensembles flous	7
I.2.3 Relations floues	8
I.2.4 Opération sur les relations floues	9
I.2.5 Composition Sup-Star	9
I.3 Théorie de la logique floue appliquée à la représentation et raisonnement linguistique	10
I.3.1 Variables linguistiques	10
I.3.2 Propositions floues	11
I.3.3 Conjonction et disjonction des propositions	12
I.3.4 Raisonnement flou	13
I.3.5 Mécanisme d'inférence	16
I.4 Application de la logique floue à la commande	19
I.4.1 Description des modules d'un régulateur flou	20
I.4.2 Modèles flous	25
I.4.3 Commande décentralisée floue	28
I.5 Conclusion	30
Chapitre II: Commande floue associée à un réglage à modes glissants	
II.1 Introduction	31
II.2 Régulateurs à modes glissants	32
II.3 Régulateur flou associé à un régulateur à modes glissants	34
II.3.1 Sélection de la surface de glissement	35
II.3.2 Nouvelles variables d'entrée	37
II.3.3 Schéma de commande à base de régulateur flou	38
II.3.4 Description des modules du régulateur flou	39
II.3.5 Surface de commande	41
II.3.6 Discussion	42
II.4 Application à la commande d'un bras de robot manipulateur	44
II.4.1 Choix du type des régulateurs	44
II.4.2 Dimensionnement des régulateurs flous de type PD et PID	45
II.4.3 Résultats de simulation	47
II.4.3.1 Application sur le robot PUMA560	48
II.4.3.2 Tests et performances	48
II.5 Conclusion	49

Chapitre III : Commande décentralisée floue adaptative continue

III.1 Introduction	55
III.2 Modèle dynamique décentralisé	56
III.3. Première approche	57
III.3.1 Commande décentralisée floue adaptative (CDFA)	57
III.3.1.1 Structure de commande	57
III.3.1.2 Application au robot PUMA560	61
III.3.2 Commande décentralisée floue adaptative robuste (CDFAR)	70
III.3.2.1 Structure de commande	70
III.3.2.2 Application au robot PUMA560	73
III.3.3 Commande décentralisée floue adaptative modifiée (CDFAM)	82
III.3.3.1 Structure de commande	82
III.3.3.2 Application au robot PUMA560	83
III.4 Deuxième approche	90
III.4.1 Commande décentralisée floue adaptative simple (CDFAS)	90
III.4.1.1 Structure de commande	90
III.4.1.2 Application au robot PUMA560	95
III.4.2 Commande décentralisée floue adaptative Robuste (CDFAR)	104
III.4.2.1 Structure de commande	104
III.4.2.2 Application au robot PUMA560	107
III.4.3 Commande décentralisée floue adaptative avec zone morte (CDFAZM)	115
III.4.3.1 Structure de commande	115
III.4.3.2 Application au robot PUMA560	119
III.5 Conclusion	120

Chapitre IV : Commande décentralisée floue adaptative discrète

IV.1 Introduction	127
IV.2 Modèle discret décentralisé du robot manipulateur	128
IV.2.1 Modèle dynamique discret centralisé	128
IV.2.2 Modèle dynamique discret décentralisé	130
IV.3 Synthèse de loi de commande	130
IV.3.1 Formulation du problème de poursuite	131
IV.3.2 Structure de commande	132
IV.3.3 Loi d'adaptation à gain constant	133
IV.3.3.1 Structure de la loi d'adaptation	133
IV.3.3.2 Application au robot PUMA560	137
IV.3.4 Loi d'adaptation à gain décroissant	146
IV.3.4.1 Structure de la loi d'adaptation	146
IV.3.4.2 Application au robot PUMA560	157
IV.4 Conclusion	159

Conclusion générale et perspectives

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

167

Annexe A

Présentation du robot PUMA560 à trois degrés de libertés , modèle dynamique et trajectoires de référence

171

Annexe B

Démonstration des propositions de la première approche de la commande décentralisée floue adaptative continue

175

Annexe C

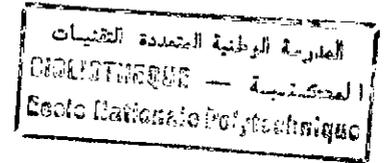
Cas particulier de la classe \mathcal{S}

183

Bibliographie

185

Notation



- a_{ij}^i (\hat{a}_{ij}^i) : Gains (gains estimés) de la matrice (matrice estimée) du modèle flou.
- c_i, d_i et f_i : Fonctions non linéaires de la dynamique du modèle du robot.
- d_n (d_p) : Variables de distance normale (parallèle) à la surface de glissement normalisée.
- i : Indice de l'articulation.
- k_{pi}, k_{vi} : Gain Adaptatifs d'action proportionnelle-dérivée.
- k_{oi}, k_{ii}, k_{ri} : Gains constants.
- nf^i, np^i et nr^i : nombre de fonctions d'appartenance, des gains adaptatifs dans chaque règle floue et des règles floues
- p_{2i}, p_{3i} : gains constants.
- q, \dot{q}, \ddot{q} : vecteurs des positions articulaires, vitesses et accélérations.
- $q_i, \dot{q}_i, \ddot{q}_i$: position articulaire, vitesse et accélération de la $i^{ème}$ articulation.
- $q_i^d, \dot{q}_i^d, \ddot{q}_i^d$: position articulaire désirée, vitesse désirée et accélération désirée de la $i^{ème}$ articulation.
- $q^d, \dot{q}^d, \ddot{q}^d$: vecteur des positions désirées, vitesses désirées et accélérations désirées.
- $m_{ii}(q)$: élément diagonal de la matrice d'inertie du modèle du robot.
- n : nombre des degrés de liberté du robot.
- r_i : erreur filtrée ou pondérée.
- $u(t)$: vecteur des commandes des différentes articulations.
- $u_i(t)$: commande de la $i^{ème}$ articulation.
- A_i^*, \hat{A}_i : Matrice des gains (estimés, des gains) des conséquences des règles floues.
- $\tilde{A}_i = A_i^* - \hat{A}_i$: erreur d'estimation de la matrice A_i^* .
- F_j : fonction d'appartenance associée à la variable linguistique F_j .
- $C_l(k, T_0) \quad l=0...3$: matrices du modèle dynamique discret du modèle du robot.
- D_n, D_p : variables floues associées respectivement aux variables $d_n; d_p$.
- $E_i(k)$: vecteur d'erreurs successives de la position.
- $E_i(t)$: Vecteur de l'erreur de poursuite en position et en vitesse.
- N_{x_i} : facteur de normalisation de la variable x_i .
- $P_{ii}^i(k), P_{2i}(k)$: matrice et gains d'adaptation décroissants.
- S_{Ni} : Surface de glissement normalisée.
- T_0 : période d'échantillonnage.
- $V_j(t)$: fonction de Lyapunov.
- Z_i : vecteur augmenté des variables d'entrées du modèle flou de Sugeno.
- $\alpha_{ii}, \beta_{ii}, \gamma_{ii}, \gamma_{m1i}, \alpha_{2i}, \beta_{2i}, \gamma_{2i}, \gamma_{m2i}$: gains d'adaptation constants.
- Γ_{ii}, Γ_{2i} : matrices d'adaptation constantes.

- Γ_i, γ_i : matrice et gain d'adaptation constants dans le cas continu et variable dans le cas discret.
- $\delta_{0i}, \delta_{1i}, \delta_{2i}$: paramètres constants.
- Δ : largeur de la zone morte.
- $\lambda_j^i, \lambda_{jN}^i, \lambda_I, \lambda_{jN}$: paramètres des surfaces de glissement normalisée et non normalisée.
- $\lambda_{\max}(\cdot)$: valeur propre maximale de la matrice (\cdot) .
- \mathfrak{S} : classe de matrices particulières (voir Annexe C).
- $\varepsilon_i(t), \hat{\varepsilon}_i(t)$: erreur de reconstruction et signal auxiliaire.
- $\tilde{\varepsilon}_i(t) = \varepsilon_i(t) - \hat{\varepsilon}_i(t)$: erreur entre erreur de reconstruction et signal auxiliaire.
- $\bar{\varepsilon}_i, \bar{\varepsilon}_j$: facteurs en pourcentage utilisés en normalisation.
- $\mu_{ij}^i(\cdot)$: la $j^{\text{ème}}$ fonction d'appartenance associée à la variable x_i de la $i^{\text{ème}}$ articulation.
- $\|(\cdot)\|$: norme Euclidienne de la matrice (\cdot) .

Abréviations

- CDFA : Commande Décentralisée Floue Adaptative.
- CDFAD : Commande Décentralisée Floue Adaptative Discrète.
- CDFAM : Commande Décentralisée Floue Adaptative Modifiée.
- CDFAR : Commande Décentralisée Floue Adaptative Robuste.
- CDFAS : Commande Décentralisée Floue Adaptative Simple.
- CDFAZM : Commande Décentralisée Floue Adaptative avec Zone Morte
- SRP : Strictement Réelle Positive.

Liste des schémas de commande

figure (II.4)	: Schéma de commande à base de régulateur flou	38
figure (III.8)	: Schéma de principe de la CDFAR	72
figure (III.15)	: Schéma de principe de la CDFAM	83
figure (III.20)	: Schéma de principe de la CDFAS	94
figure (III.27)	: Schéma de principe de la CDFAR	107
figure (III.33)	: Schéma de principe de la CDFAZM	118
figure (IV.1)	: Schéma de principe de la CDFAD à gain constant	136
figure (IV.13)	: Schéma de principe de la CDFAD à gain décroissants	157

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Introduction

Introduction générale

Avec le besoin pressant d'augmenter la productivité et la distribution de produit finis de qualité constante, l'industrie s'intéresse en plus vers l'automatisation des procédés. A présent, la plupart des tâches automatisées sont réalisées par des machines spécialisées conçues de manière à exécuter une fois prédéterminées dans un procédé industriel. L'inflexibilité et le coût excessif de ces machines ont amené les gens à s'intéresser à l'emploi de robots capables d'exécuter une variété de tâches dans un environnement plus flexible et avec un coût de production plus bas.

Les robots industriels sont capables de remplacer des opérateurs pour effectuer des tâches différentes. Les robots industriels d'aujourd'hui sont en effet des appareils actionnés mécaniquement sous le contrôle d'un ordinateur. Leurs performances surhumaines telles que l'infatigabilité, l'insensibilité aux environnements hostiles, la répétabilité des tâches, la précision et la rapidité d'actions poussent sans cesse au développement de la robotisation dans l'industrie.

De point de vue commande, les robots manipulateurs sont caractérisés par une dynamique non linéaire avec un couplage important entre les articulations. Dans les schémas de commande classiques des robots manipulateurs, chaque articulation est contrôlée indépendamment via une boucle de régulation. Les coefficients du régulateur sont fixes. Ces schémas de commande sont souhaitables pour réaliser les tâches de déplacement d'objets ("*Pick and place task*") dont les robots sont souvent chargés. Cependant, ces schémas apparaissent inefficaces dans le cas de présence d'incertitude sur les paramètres décrivant les propriétés dynamique de la charge portée par l'élément terminal [SER89].

Vue l'incessible croissance de l'utilisation des robots dans des tâches et des milieux de plus en plus diverses, un important effort a été consacré au développement de nouvelles techniques de commande performante. Ces recherches se sont fondées par l'apparition de deux approches principales de commande [SLO88] :

- Les techniques de commande à base du modèle ("*Model based techniques*");
- Les techniques de commande à base de performance ("*Performance based techniques*");.

Dans les techniques de commande à base du modèle (telle que la méthode du couple calculée "*Computed torque method*"), le calcul de la commande nécessite une parfaite connaissance du modèle du robot et de ses paramètres. Cette approche conduit à une loi de commande dont les performances sont fortement liées à la fidélité du modèle dynamique utilisé pour décrire le comportement du robot. Les paramètres de ce modèle sont difficiles à mesurer avec précision et peuvent varier dans les conditions réelles (telles que les paramètres de la charge). Des erreurs de modélisation ou une variation de charge peuvent avoir des effets néfastes sur les performances du système puisqu'elles contribuent directement au calcul de la commande.

Lorsque le concepteur du système de commande cherche une meilleure robustesse vis-à-vis d'éventuelles variations paramétriques, la technique de commande robuste s'avère intéressante. Ce type de commande est basé sur une connaissance complète ou partielle de la structure du modèle du robot et des variations de ses paramètres. Plusieurs travaux portés sur la commande robuste tels que [GUZ86][HA87] ont permis d'aboutir à des algorithmes assurant une bonne poursuite de trajectoires de référence. Cependant des oscillations intempestives (phénomène du "*chattering*") ainsi qu'une activation maximale de la commande (des valeurs excessives) sont souvent constatées ce qui est indésirable pour le système de commande.

Parmi les techniques de commande à base de performance, on trouve la commande adaptative. Grâce à leur capacité d'adaptation, les schémas de commande adaptative permettent de prendre en

charge les systèmes dont les modèles ou l'environnement sont mal connus à priori ou susceptibles de varier de façon imprévisible. L'utilisation des techniques adaptatives dans la commande des robots manipulateurs devient de plus en plus répandue vue les avantages intéressants qu'elle offre à savoir la compensation des incertitudes dans le modèle dynamique du robot, la réduction de la dépendance avec des informations à priori sur le modèle, la possibilité d'adaptation rapide à toutes variations imprévues et surtout le potentiel d'une amélioration continue des performances lorsque le robot est opérationnel.

Plusieurs schémas de commande adaptative développés pour le cas du robot sont basés sur des hypothèses et des approximations [SLO88] telles que la linéarisation locale, l'invariance des paramètres, les dynamiques découplées. D'autre part, Dans [CRA86][LEW93] on propose des lois de commande adaptative où la convergence de l'erreur de poursuite est assurée sans considérer les approximations précédentes. Dans [CRA86] le calcul de la commande nécessite le calcul de l'inverse de l'estimé de la matrice d'inertie ce qui rend le temps de calcul de la commande fastidieux. Dans [SLO88][LEW93] les auteurs proposent des lois de commande où interviennent deux actions principales : une action de stabilisation réalisée par un régulateur de type proportionnel-dérivé (PD) et une deuxième action adaptative pour compenser les dynamiques inconnues.

La plupart des schémas de commande sont développés dans un contexte centralisé [SER89] tels que [DUB79][SER87][SLO88] dans le cas continu et [HOR82][TAR89] dans le cas discret. Dans ce type de schéma, le régulateur est amené à traiter un volume important d'informations issues des différentes articulations. Dans le but de soulager le calcul dans le régulateur et d'offrir une meilleure robustesse vis-à-vis des informations erronées, l'élaboration d'un schéma de commande décentralisé s'avère utile. Dans ce cas, chaque articulation est considérée comme un sous-système indépendant ayant son propre régulateur. Ce dernier agit seulement à partir des mesures locales. Caractérisée par une implémentation facile, la commande décentralisée présente certaine fiabilité et une robustesse aux défaillances des capteurs et des actionneurs. Ces avantages ont poussé certains chercheurs à consacrer leurs travaux pour ce type de commande dont nous citons [FU92] [CHO88][SER89][SPO95c].

Autres que l'approche des techniques conventionnelles basées sur une description mathématique du système de commande, les efforts des chercheurs se sont consentis pour comprendre les aspects structurels et fonctionnels des systèmes biologiques et en particulier les mécanismes de la pensée humaine. Cela a conduit au développement de nouvelles approches qui peuvent tenir compte des non linéarités et des incertitudes inhérentes au système réel. Une des plus attrayantes n'est plus que l'approche linguistique floue. Basée sur l'imitation des aspects approximatifs et qualificatifs du raisonnement humain, les méthodes à base de linguistique font intervenir des citations conditionnelles linguistiques du type "Si-Alors" pour résoudre des problèmes de décision (contrôle) ou pour décrire le comportement dynamique du système inconnu ou mal défini [LOU97]. La plupart de ces méthodes sont basées sur le modèle de Mamdani. Ce dernier consiste à formuler un ensemble de règles de décision en termes linguistiques utilisant des ensembles flous pour décrire l'action à prendre en fonction des valeurs linguistiques des variables d'entrée. Ce modèle est très répandu lors de disponibilité d'une description linguistique du comportement que doit réaliser le régulateur [DRI93][WAT97]. D'autres méthodes sont développées en se basant sur un modèle appelé modèle de Sugeno. Ce dernier consiste en une formulation où les règles de décision sont caractérisées par une antécédente linguistique et une conséquence numérique. Le modèle flou de Sugeno est souvent utilisé lors de nécessité d'identification, d'approximation ou de compensation de fonctions non linéaires inconnues.

En effet, des études ont montré que les systèmes à base de logique floue sont dotés de la propriété d'approximation et de généralisation de fonctions non linéaires [MEN95]. De ce fait, ces systèmes constituent un outil de plus, à côté des systèmes à réseaux de neurones, utile dans la commandes des systèmes non linéaires complexes tels que les robots manipulateurs

[COM96][KWA97][TRE97]. Ainsi, un algorithme d'adaptation des paramètres du modèle flou peut être développé ce qui permet d'éviter la procédure de dimensionnement.

L'objectif de notre travail présenté dans ce mémoire est d'étudier et de porter une modeste contribution au développement de la technique de commande décentralisée floue en vue de la commande en poursuite de trajectoires de références appliquée aux robots manipulateurs. Pour l'application par simulation nous avons considéré le célèbre robot PUMA560.

A cet effet, notre mémoire est divisée en quatre chapitre :

Le premier chapitre présente les concepts de base des systèmes et des régulateurs à logique floue. Après une formulation mathématique des différents outils de la logique floue, nous développons son application pour représenter et raisonner sur des expressions linguistiques. A la suite, nous traitons l'aspect du régulateur flou et ses différentes contributions dans les schémas de commande décentralisée.

Dans le deuxième chapitre, nous développons un schéma de commande floue à structure décentralisée. Basée sur un modèle de type Mamdani, le régulateur utilise deux variables de distances calculées à partir d'une surface de glissement prédéterminée. L'ensemble des règles floues sont déduites du principe des régulateurs à modes glissants.

Dans le troisième chapitre nous proposons des lois de commande décentralisée floue adaptative continue. Les lois de commande proposées font intervenir trois actions principales : une action PD, une action de compensation des termes de couplages via un modèle flou de Sugeno adaptatif et une troisième action à travers un signal auxiliaire pour remédier aux erreurs de reconstruction. Selon que l'action PD utilise des gains adaptatifs ou constants, nous distinguons deux approches. L'analyse de la stabilité et la bornitude des différents signaux est effectuée via la théorie de Lyapunov.

Dans le quatrième et dernier chapitre nous proposons deux schémas de commande décentralisée floue adaptative discrète. Dans les lois de commande interviennent trois actions principales : une action de stabilisation réalisée par un filtre linéaire, une action de compensation des termes inconnus effectuée par un modèle flou de Sugeno adaptatif et une troisième action pour palier aux erreurs de reconstruction. Dans le premier schéma, les paramètres adaptatifs de la de commande sont ajustés via une loi d'adaptation à gain constant. Une loi d'adaptation à gain décroissant est adoptée dans la deuxième loi de commande. Une fonction de Lyapunov candidate et l'approche d'hyperstabilité sont utilisées pour vérifier la stabilité des schémas de commande.

Enfin, notre mémoire est clôturée par une conclusion générale et des perspectives pour une éventuelle continuité du présent travail.

Chapitre I

*Systemes et régulateurs
à logique floue*

Chapitre I :

Systèmes et régulateurs à logique floue

I.1 Introduction

La majorité des systèmes industriels complexes sont difficiles à contrôler. Cette difficulté découle généralement de la complexité du système dynamique dont il faut assurer la conduite non linéaire, dimensionnalité élevée, objectifs et contraintes de commande multiples, erreurs de modélisation, bruit de mesures, perturbations...

Pour faire face à ces difficultés, de nouvelles stratégies de commande, basées sur l'expertise de l'opérateur humain ont été élaborées. Le point clef de ces stratégies réside dans l'utilisation du calcul symbolique pour l'élaboration de l'algorithme de commande. Cette méthodologie ouvre une nouvelle voie pour l'utilisation d'heuristiques en remplaçant la modélisation du régulateur par une loi de commande basée sur des connaissances linguistiques acquises d'un opérateur du procédé.

En 1965, Zadeh [ZAD65] publie le premier article concernant une nouvelle méthode pour décrire les incertitudes non-probabilistiques, ce qui l'appellera les ensembles flous [JAN95]. Dans cette méthode, il établit le lien entre les ensembles flous et la modélisation non-probabilistique de l'incertitude grâce à la théorie des possibilités dans laquelle les ensembles flous servent à représenter les idées de valeurs incomplètement spécifiées, malconnues. Ensemble flou et théorie de possibilité sont à la base d'une méthodologie de représentation et d'utilisation de connaissances imprécises vagues et incertaines que Zadeh appelle théorie de raisonnement approché et qui est plus connu sous le vocable de logique floue.

Dans ce chapitre, nous allons examiner les principales notions de la logique floue. Ensuite, nous passons à l'application de ces notions pour la représentation des expressions linguistiques. Ce chapitre est organisé comme suit : dans la première section nous allons examiner les concepts de la logique floue. Ensuite, dans la deuxième section, nous passons en revue l'application de ces concepts pour la représentation des expressions linguistiques. Via l'étude d'un régulateur flou linguistique, nous allons présenter dans la dernière section, l'application de la logique floue dans la commande. Enfin, nous terminons par une conclusion.

I.2 Logique floue

Dans la perspective des sciences de l'ingénieur, on peut admettre que dans les situations où les méthodes traditionnelles de modélisations à partir d'observations physiques s'avèrent non-satisfaisantes, les sciences subjectives particulièrement la logique floue peuvent rendre beaucoup de services, dès lors que des connaissances sur la façon de résoudre un problème, de piloter un grand système, d'effectuer un réglage...etc sont disponibles. Cela suppose que l'on sache définir des méthodes rigoureuses de représentation des connaissances. En pratique, la résolution d'un problème concret passera par l'utilisation conjointe de méthodes objectives traditionnelles et de méthodes subjectives exploitantes des connaissances. La théorie de la logique floue offre un cadre formel qui n'existait pas auparavant, pour mettre en œuvre de telles méthodes de façon rigoureuses. Dans cette

section sont introduite les principales notions liées aux ensembles et relations flous ainsi que leurs utilisations pour représenter et raisonner sur des expressions linguistiques.

1.2.1 Ensembles flous

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

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

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

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

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

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

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

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

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

Exemple : La propriété "x est très proche de 0.2" définie sur l'univers discret $U_x = \{-1, -0.6, -0.2, 0.2, 0.6, 1\}$.

peut être représentée par un ensemble flou $\underline{F} = \int_U \mu_F(x) / x$ dont la fonction caractéristique est donnée par la table (I.1), une représentation graphique est illustrée à la figure (I.1).

x	-1	-0.6	-0.2	0.2	0.6	1
$\mu_F(x)$	0	0	0.5	1	0.5	0

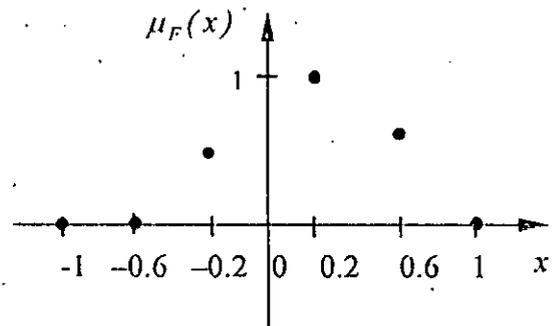
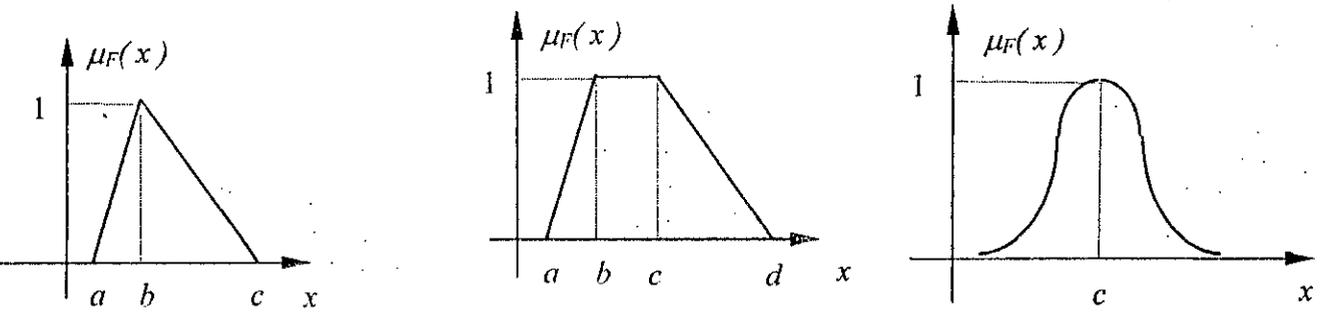


table (I.1) : Fonction caractéristique de la Relation "x est très proche de 0.2".

figure (I.1) : Fonction caractéristique de la relation "x est très proche de 0.2"

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

La figure (I.2) montre l'allure de ces différentes fonctions (a, b, c, d des paramètres).



$$\begin{aligned} \text{Triangle}(x, a, b, c) &= \max(\min(\frac{x-a}{b-a}, \frac{c-x}{b-c}), 0) \\ \text{Trapezoïde}(x, a, b, c, d) &= \max(\min(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}), 0) \\ \text{Gaussienne}(x, \sigma, c) &= \exp(-(\frac{x-c}{\sigma})^2) \end{aligned}$$

figure(I.2) : Exemple de fonctions d'appartenance

Vu l'efficacité de calcul, les fonctions triangulaires et trapézoïdales sont les plus utilisées, particulièrement dans les implémentations en temps réel. Cependant, le passage par les points spécifiés par les paramètres $a, b, c,$ et d nécessite l'introduction d'un lissage. Cela est évité dans le cas des fonctions gaussiennes ou sigmoïdes.

I.2.2 Opérations sur les ensembles flous

Puisque les fonctions caractéristiques des ensembles flous sont graduelles, l'extension de certaines opérations définies sur les ensembles classiques au cas des ensembles flous pose certaines difficultés [JAN95]. Donc, il n'est pas possible d'appliquer directement la logique propositionnelle qui manipule des ensembles à fonctions caractéristiques binaires (0 ou 1). Pour palier à ce problème, on introduit les définitions de trois outils nécessaires pour le traitement des ensembles flous.

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

a°/ Définition de norme triangulaire (T-norme) : [DRJ93]

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

- T.1 $a \otimes b = b \otimes a$
- T.2 $(a \otimes b) \otimes c = a \otimes (b \otimes c)$
- T.3 $a \leq c$ et $b \leq d \Rightarrow a \otimes b \leq c \otimes d$
- T.4 $a \otimes 1 = a$

b°/ Définition de norme co-triangulaire (T-conorme ou S-norme)

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

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

c°/ Définition du complément

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

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

d°/ Définitions de l'intersection d'ensembles flous

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

$$\mu_F(x) = \mu_{F_1 \cap F_2}(x) = \mu_{F_1}(x) \otimes \mu_{F_2}(x) \tag{I.1}$$

Exemple: Parmi les opérations d'intersection :

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

e°/ Définition de l'union d'ensembles flous

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

$$\mu_F(x) = \mu_{F_1 \cup F_2}(x) = \mu_{F_1}(x) \oplus \mu_{F_2}(x) \tag{I.2}$$

Exemple : Parmi les opérations d'union :

- ♦ Opérateur maximum noté "max" : $\mu_{F_1 \cup F_2}(x) = \max(\mu_{F_1}(x), \mu_{F_2}(x))$
- ♦ Opérateur somme algébrique : $\mu_{F_1 \cup F_2}(x) = \mu_{F_1}(x) + \mu_{F_2}(x) - \mu_{F_1}(x) \times \mu_{F_2}(x)$

f°/ Définition de complément d'ensemble flou

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

$$\mu_{\overline{F}_1}(x) = \overline{\mu_{F_1}(x)} \tag{I.3}$$

Exemple: Opérateur de complément : $\mu_{\overline{F}_1}(x) = 1 - \mu_{F_1}(x)$

1.2.3 Relations floues

En logique classique, une relation représente la présence ou l'absence d'association, d'interaction ou d'interconnexion entre les éléments de deux ensembles ou plus. En logique floue, une relation traduit le degré de présence ou d'absence d'association, d'interaction ou d'interconnexion entre les éléments de deux ensembles flous ou plus.

Soit, $U_{x_i} \ i=1, \dots, n$: des univers de discours. Une relation floue R définie sur l'espace de produit $U_{x_1} \times U_{x_2} \times \dots \times U_{x_n} \hat{=} \prod_{i=1}^n U_{x_i}$ est un ensemble flou caractérisé par la fonction d'appartenance μ_R telle que :

$$\underline{R} = \int_{\prod_{i=1}^n U_{x_i}} \mu_R(x_1, \dots, x_n) / (x_1, \dots, x_n)$$

Exemple : Considérons la relation binaire R : "x est très proche de y" définie sur l'univers $U_x \times U_y$, tel que: $U_x = \{0.1, 0.2, 0.3\}$; $U_y = \{0, 0.1, 0.2\}$. La fonction d'appartenance de \underline{R} peut être décrite par la table (I.2).

$y \downarrow \quad x \rightarrow$	0.1	0.2	0.3
0	0.5	0.25	0
0.1	1	0.5	0.25
0.2	0.5	1	0.5

table (I.2) : table de la fonction μ_R

1.2.4 Opérations sur les relations floues

Vu que les relations floues sont définies par des ensembles flous, il est possible d'utiliser les opérations d'intersection, d'union et de complément pour combiner plusieurs relations floues définies sur le même univers de discours.

Soit R et S deux relations floues définies sur l'univers $U_x \times U_y$. L'intersection de R et S est définie par l'ensemble flou : $\underline{R} \cap \underline{S}$ tel que :

$$\underline{R} \cap \underline{S} = \int_{U_x \times U_y} \mu_{R \cap S}(x, y) / (x, y) = \int_{U_x \times U_y} \mu_R(x, y) \otimes \mu_S(x, y) / (x, y) \quad (I.4)$$

L'union des deux relations R et S est définie par l'ensemble flou $\underline{R} \cup \underline{S}$ tel que :

$$\underline{R} \cup \underline{S} = \int_{U_x \times U_y} \mu_{R \cup S}(x, y) / (x, y) = \int_{U_x \times U_y} \mu_R(x, y) \oplus \mu_S(x, y) / (x, y) \quad (I.5)$$

\otimes et \oplus dénotent respectivement la T-norme et la S-norme.

1.2.5 Composition Sup-Star

La composition Sup-Star est une opération indispensable pour effectuer des combinaisons entre ensembles et relations flous. Elle est basée sur les principes de l'Extension Cylindrique et de la Projection [DRI93].

1.2.5.1 Principe d'Extension Cylindrique

Il permet l'extension de l'univers de discours d'un ensemble ou d'une relation floue à un univers de dimension plus grande.

Soit l'univers de discours $U = \prod_{i=1}^n U_{x_i}$ et soit V un sous-univers de U tel que: $V = \prod_{m=1}^k U_{x_{i_m}}$, $i_m \leq n, k < n$, sur lequel on définit une relation floue \underline{R} .

$$\underline{R} = \int_V \mu_R(x_{i_1} \dots x_{i_k}) / (x_{i_1} \dots x_{i_k})$$

L'extension cylindrique de \underline{R} sur U est défini par :

$$Ce_U(\underline{R}) = \int \mu_R(x_{i_1} \dots x_{i_k}) / (x_1 \dots x_n) \quad (I.6)$$

Grâce à l'introduction de la notion de l'extension cylindrique, il sera possible de réaliser les opérations d'union, d'intersection entre ensembles et relations flous. Il suffit d'effectuer des extensions sur un même univers contenant tous les univers des ensembles et relations, et d'appliquer l'opération envisagée.

1.2.5.2 Principe de Projection

A partir d'une relation ou ensemble flou défini sur un univers donné, la projection permet d'extraire une relation définie sur un univers de dimension inférieure.

Soit la relation R définie sur l'univers $U = \prod_{i=1}^n U_{x_i}$. La projection de R sur l'univers $V = \prod_{m=1}^k U_{x_{i_m}}$, $i_m \leq n, k < n$, est la relation $Proj_V(R)$ définie par :

$$Proj_V(\underline{R}) = \int_{V^{x_{i_1} \dots x_{i_k}}} Sup \mu_R(x_1 \dots x_n) / (x_{i_1} \dots x_{i_k}) \quad (I.7)$$

où $\prod_{m=1}^l U_{x_{i_m}}$ est l'univers complémentaire de V dans U ($l+k=n$).

1.2.5.3 Composition Sup-Star

La combinaison d'ensemble flou avec une relation floue est appelée *composition*. Soit un ensemble flou \underline{A} défini sur $\prod_{i=1}^n U_{x_i}$ et \underline{R} relation définie sur $\prod_{i=1}^n U_{x_i} \times \prod_{i=1}^m U_{y_i}$. La composition Sup-Star de \underline{A} et \underline{R}

est l'ensemble flou \underline{B} défini sur $\prod_{i=1}^m U_{y_i}$ tel que :

$$\underline{B} = \underline{A} \circ \underline{R} = Proj_{\prod_{i=1}^m U_{y_i}} (Ce_{\prod_{i=1}^n U_{x_i} \times \prod_{i=1}^m U_{y_i}} (\underline{A}) \cap \underline{R}) \quad (1.8)$$

donc,
$$\underline{B} = \int_{\prod_{i=1}^m U_{y_i}} Sup_{(x_1 \dots x_n)} (\mu_{\underline{A}}(x_1 \dots x_n) \otimes \mu_{\underline{R}}(x_1 \dots x_n, y_1 \dots y_m)) / (y_1 \dots y_m) \quad (1.9)$$

1.3 Théorie de la logique floue appliquée à la représentation et raisonnement linguistique

L'utilisation des ensembles et des relations floues offre une base pour une voie systématique de manipulation de concept vague et imprécis. En particulier, on peut représenter et raisonner sur des expressions linguistiques ou des mots du langage naturel.

Considérons l'expression verbale suivante: [DRI93]

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

Dans cette expression interviennent les notions clefs suivantes :

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

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

1.3.1 Variables linguistiques

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

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

1. X : le nom symbolique de la variable. Exemple : E pour l'erreur ΔE pour la variation d'erreur.
2. LX : l'ensemble des valeurs de la variable X . Exemple : $LE = \{N, Z, P\}$ où N , Z et P indiquent respectivement, négative, nulle et positive.
3. U_X : l'univers de discours dans lequel la variable de symbole X prend ses valeurs. Si la variable est normalisée entre -1 et +1 alors $U_X = [-1 +1]$.
4. M_X : fonction sémantique, qui attribue à chaque valeur de X son sens. Ce dernier est caractérisé par un ensemble flou défini sur l'univers de discours U_X , donc :

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

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

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

Exemple : Considérons le cas de la variable Erreur, on a $LE = \{N, Z, P\}$. La fonction M_X consiste à attribuer à ces valeurs des ensembles flous $\underline{N}, \underline{Z}, \underline{P}$. La figure (I.3) montre une possibilité de choix d'ensembles.

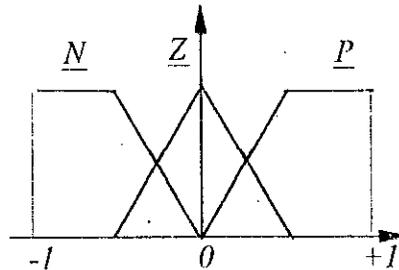


figure (I.3) : Exemple d'ensembles flous associés à la variable de l'erreur

Remarque:

L'attribution de fonctions d'appartenance aux valeurs d'une variable linguistique est une opération subjective, donc elle diffère d'une personne à l'autre. Cela est dû au fait que les concepts abstraits (tel que la description linguistique) sont de nature imprécise.

Comme indique la figure (I.3), dans les zones de chevauchement, la variable floue prend plus d'une valeur à la fois mais avec des degrés d'appartenance différents, ce qui reflète l'imprécision sur les frontières de ces valeurs. Dans le cas général, le passage entre des valeurs floues voisines (qui se chevauchent) est graduel (progressif).

I.3.2 Propositions floues

On distingue deux types de propositions :

I.3.2.1 Proposition floue simple

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

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

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

NB : le symbole d'un ensemble flou \underline{NB} associé à la valeur négative grande.

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

I.3.2.2 Proposition floue composée

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

Exemple

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

A partir de ces valeurs, on peut construire les propositions floues composées suivantes :

"E est NG ou E est PP"; "E est NG et ΔE est P"; "E est PM ou ΔE est N"; "E n'est pas P".

En logique floue, de telles propositions peuvent être quantifiées en utilisant les notions : conjonction, disjonction et négation.

1.3.3 Conjonction et disjonction des propositions

1.3.3.1 Conjonction

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

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

est la proposition composée P telle que :

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

Le sens de la proposition P est représenté par un ensemble flou \underline{F} résultat de l'intersection des ensembles flous $\underline{F}_1, \underline{F}_2, \dots$, et \underline{F}_n . D'après (I.1), on obtient :

$$\mu_F(x) = \mu_{F_1}(x) \otimes \mu_{F_2}(x) \otimes \dots \otimes \mu_{F_n}(x) \quad x \in U_x \quad (I.10)$$

on écrit :

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

1.3.3.2 Disjonction

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

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

est la proposition composée P telle que :

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

Le sens de la proposition P est représenté par un ensemble flou \underline{F} résultat de l'union des ensembles flous $\underline{F}_1, \underline{F}_2, \dots$, et \underline{F}_n . D'après (I.2), on obtient :

$$\mu_F(x) = \mu_{F_1}(x) \oplus \mu_{F_2}(x) \oplus \dots \oplus \mu_{F_n}(x) \quad x \in U_x \quad (I.11)$$

on écrit :

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

1.3.3.3 Addition de conjonction et disjonction

Considérons la proposition composée suivante :

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

Contrairement au sens de la conjonction décrit précédemment, dans ce cas les variables $(X_i)_{i=1..n}$ dans la proposition P sont définies sur des univers différents $(U_{x_i})_{i=1..n}$. Le sens de la proposition P est

représenté par une relation R définie sur l'univers $\prod_{i=1}^n U_{x_i}$. Elle est obtenue en utilisant le principe d'extension cylindrique et l'opérateur d'intersection. D'après (I.1) et (I.3), on a :

$$R = \bigcap_{i=1}^n Ce(\underline{F}_i) = \int_{\prod_{i=1}^n U_{x_i}} \bigotimes_{i=1}^n \mu_{F_i}(x) / (x_1, \dots, x_n) \quad (I.12)$$

où,

$$\bigcap_{i=1}^n Ce(\underline{F}_i) \hat{=} Ce(\underline{F}_1) \bigcap_{\prod_{i=1}^n U_{x_i}} Ce(\underline{F}_2) \bigcap \dots \bigcap_{\prod_{i=1}^n U_{x_i}} Ce(\underline{F}_n)$$

De la même façon, dans le cas de disjonction, le sens de la proposition P suivante :

$$P : "X_1 \text{ est } F_1 \text{ ou } X_n \text{ est } F_2 \text{ ou } \dots \text{ } X \text{ est } F_n"$$

est représenté par la relation floue :

$$\underline{R} = \bigcup_{i=1}^n Ce(\underline{F}_i) = \int_{\prod_{i=1}^n U_{x_i}} \mu_{\underline{R}}(x) / (x_1 \dots x_n) \quad (I.13)$$

où,

$$\bigcup_{i=1}^n Ce(\underline{F}_i) \hat{=} Ce(\underline{F}_1) \cup Ce(\underline{F}_2) \cup \dots \cup Ce(\underline{F}_n)$$

I.3.3.4 Négation

La négation d'une proposition P donnée par :

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

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

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

où \bar{F} est l'ensemble flou complément de l'ensemble flou F .

I.3.4 Raisonnement flou

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

Dans le raisonnement flou interviens trois niveaux fondamentaux :

1. Implication floue;
2. Inférence floue;
3. Agrégation floue.

I.3.4.1 Implication Floue

Les règles de la forme *SI-ALORS* qui manipulent des variables linguistiques (ou flou), sont appelées règles floues. L'utilisation des opérations des ensembles et de relations floues nous a permis dans la section précédente (§ I.3), d'associer des ensembles flous à des propositions floues complexes. Donc, il sera toujours possible d'écrire une règle de la forme :

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

sous la nouvelle forme :

$$SI X \text{ est } F \text{ ALORS } Y \text{ est } B \quad (I.14)$$

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

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

donc,

$$\underline{F} = \int_{\prod_{i=1}^n U_{x_i}} \mu_{\underline{F}}(x_1 \dots x_n) / (x_1 \dots x_n) = \int_{\prod_{i=1}^n U_{x_i}} \mu_{F_i}(x_i) / (x_1 \dots x_n) \quad (I.15)$$

Le sens de (I.15) est représenté par une relation d'implication floue définie sur l'univers de discours $\prod_{i=1}^n U_{x_i} \times U_y$. Cette relation mesure le degré de vérité de la règle floue, donc de l'implication entre antécédente et conséquence.

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

Sur la base de ces deux relations, nous avons introduit plusieurs définitions de l'implication floue, parmi lesquelles nous citons [DRJ93] :

- **La norme de Klanc-Dienes** : basé sur la relation $\bar{p}_1 \vee p_2$, l'opérateur "et" est réalisé par T-norme "maximum". Elle est définie par la fonction d'appartenance suivante :
 $\mu_{F \rightarrow B}(x_1 \dots x_n, y) = \max(1 - \mu_F(x_1 \dots x_n), \mu_B(y))$
- **La norme de Lukasiewicz** : basé sur la relation d'implication floue est donnée par la relation:
 $\mu_{F \rightarrow B}(x_1 \dots x_n, y) = \min(1, 1 - \mu_F(x_1 \dots x_n) + \mu_B(y))$
- **La norme de Zadeh** : basée sur la relation $(p_1 \wedge p_2) \vee p_2$, l'opérateur et est utilisé par T-norme "somme-bornée", la relation d'implication floue est donnée par la relation :
 $\mu_{F \rightarrow B}(x_1 \dots x_n, y) = \max(\min(\mu_F(x_1 \dots x_n), \mu_B(y)), 1 - \mu_F(x_1 \dots x_n))$

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

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

Elles sont définies par les relations suivantes :

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

1.3.4.2 Inférence Flou :

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

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

1. MTG :

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

2. MPG

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

MTG trouve son domaine d'application dans les systèmes experts, tandis que le MPG est utilisé dans les systèmes de commande à base de logique floue.

Dans MPG, l'ensemble inféré B' est le résultat de l'inférence (ou la composition) de l'ensemble flou F' et de la relation de l'implication floue, notée R . Pour obtenir B' , on utilise la composition Sup-Star définie par (I.9), on aura :

$$B' = F' \circ R = \int_{y \in U, x \in U} \text{Sup}(\mu_{F'}(x) \otimes \mu_R(x, y)) / y \tag{I.16}$$

Dans le cas de plusieurs variables dans l'antécédentes, la règle floue est la condition s'écrivent :
 $SI X_1 \text{ est } F_1 \text{ et } \dots \text{ et } X_n \text{ est } F_n \text{ ALORS } Y \text{ est } B$
 $X_1 \text{ est } F_1' \text{ et } X_2 \dots \text{ et } X_n \text{ est } F_n'$

L'ensemble flou inféré par MPG est donné par la fonction d'appartenance suivante :

$$\underline{B}' = \int_{y \in U_y} \text{Sup}_{(x_1, \dots, x_n) \in \prod_{i=1}^n U_{x_i}} \left(\bigotimes_{i=1}^n \mu_{F_i}(x_i) \otimes \mu_B(x_1 \dots x_n, y) \right) / y \quad (I.17a)$$

donc,

$$\underline{B}' = \int_{y \in U_y} \mu_{B'}(y) / y = \int_{y \in U_y} \text{Sup}_{(x_1, \dots, x_n) \in \prod_{i=1}^n U_{x_i}} \left[\left(\bigotimes_{i=1}^n \mu_{F_i}(x_i) \right) \otimes \left(\left(\bigotimes_{i=1}^n \mu_{F_i}(x_i) \right) \otimes \mu_B(y) \right) \right] / y \quad (I.17b)$$

Cas Particulier :

Dans le cas de fonctions singletons :

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

L'expression de l'ensemble inféré devient simple si :

a. Les T-normes sont choisis "produit", on obtient :

$$\underline{B}' = \int_{U_y} \left(\prod_{i=1}^n \mu_{F_i}(x_i^*) \right) \times \mu_B(y) / y$$

dont \underline{B}' est une image de \underline{B} compressée par le facteur $\prod_{i=1}^n \mu_{F_i}(x_i^*) < 1$

b. Les T-normes sont choisis "minimum", on aura :

$$\underline{B}' = \int_{U_y} \min(\mu_{F_1}(x_1^*), \mu_{F_2}(x_2^*), \dots, \mu_{F_n}(x_n^*), \mu_B(y)) / y$$

donc \underline{B}' est une image de l'ensemble \underline{B} après troncature.

Remarque :

Dans l'expression de l'ensemble inféré (I.17b), le T-norme est utilisé pour :

- ◆ Evaluer le connecteur "et" dans l'antécédente.
- ◆ Evaluer l'implication entre l'antécédente et la conséquence
- ◆ Evaluer la composition (relation (I.9)).

dans ces trois niveaux, les T-normes peuvent être choisis distincts.

1.3.4.3 Agrégation des règles :

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

Considérons les règle sous la forme symbolique suivante :

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

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

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

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

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

$$\underline{B}' = \int_{U_y} \bigoplus_{l=1}^k (\mu_{B^{(l)}}(y)) / y \quad (I.18a)$$

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

$$\underline{B}' = \int_{U_y} \bigoplus_{i=1}^k \text{Sup}_{(x_1, \dots, x_n)} \left[\left(\bigotimes_{i=1}^n \mu_{F_i'}(x_i) \right) \otimes \left(\left(\bigotimes_{i=1}^n \mu_{F_i''(x_i)} \right) \otimes \mu_{B''(y)} \right) \right] / y \quad (I.18b)$$

I.3.5 Mécanisme d'inférence :

D'après l'expression (I.18b), l'ensemble inféré dépend essentiellement des opérateurs de composition adoptés. Vu la richesse et la diversité des opérateurs possibles, on introduit la notion de méthode ou mécanisme d'inférence. Elle détermine la réalisation des différents opérateurs "et", "ou" et "implication" dans une inférence, permettant ainsi un traitement numérique de cette dernière.

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

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

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

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

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

X_1 est F'_1 et X_2 est F'_2

avec la condition :

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

Les ensembles flous NG, EZ et PG sont illustrés par la figure (I.4).

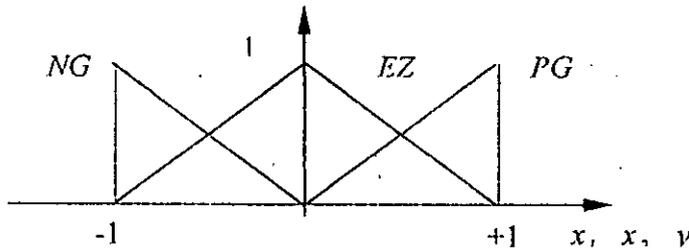


figure (I.4) : Ensembles flous associés à NG, EZ et PG

Il faut noter que les propositions floues simples de l'antécédente de la deuxième règle sont liées par l'opérateur "ou" et non pas par l'opérateur "et". Donc, l'ensemble flou inféré de cette règle ne peut être calculé directement à partir de la relation (I.18b)

I.3.5.1 Méthode d'inférence Max-min :

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

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

Dans ce cas l'expression de l'ensemble inféré (I.18b) devient :

$$\mu_{B'}(y) = \max_{i=1, \dots, k} \left[\text{Sup}_{(x_1, \dots, x_n)} \left(\min \left(\mu_{F_i'}(x_1), \dots, \mu_{F_i'}(x_n), \mu_{F_i''(x_1)}, \dots, \mu_{F_i''(x_n)}, \mu_{B''(y)} \right) \right) \right] \quad (I.19a)$$

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

$$\mu_{B'}(y) = \max_{i=1, \dots, k} \left[\min \left(\mu_{F_i''(x_1^*)}, \dots, \mu_{F_i''(x_n^*)}, \mu_{B''(y)} \right) \right] \quad (I.19b)$$

Une représentation graphique de l'application de cette méthode pour inférer les deux règles et la condition présentées précédemment, est illustrée par la figure (I.5).

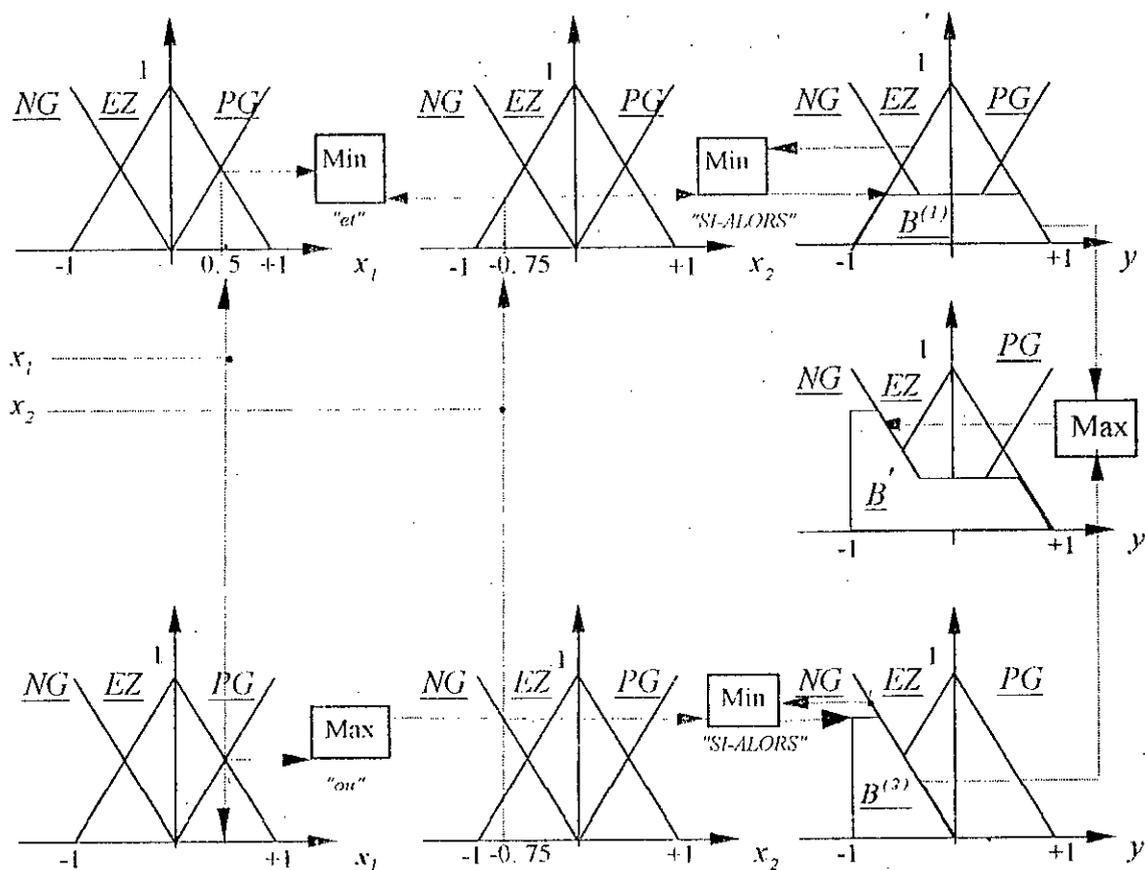


figure (1.5) : Méthode d'inférence "Max-Min".

1.3.5.2 Méthode d'inférence Max-prod :

Dans cette méthode :

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

L'expression de l'ensemble inféré devient :

$$\mu_B(y) = \max_{l=1 \dots k} \left[\text{Sup}_{(x_1, \dots, x_n)} \left(\min \left(\min(\mu_{F_1}(x_1), \dots, \mu_{F_n}(x_n)), \min(\mu_{F_1^{(l)}}(x_1), \dots, \mu_{F_n^{(l)}}(x_n)) \times \mu_{B^{(l)}}(y) \right) \right) \right] \quad (1.20a)$$

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

$$\mu_B(y) = \max_{l=1 \dots k} \left[\min(\mu_{F_1^{(l)}}(x_1^*), \dots, \mu_{F_n^{(l)}}(x_n^*)) \times \mu_{B^{(l)}}(y) \right] \quad (1.20b)$$

La représentation graphique est illustrée à la figure (1.6)

1.3.5.3 Méthode d'inférence Som-prod :

Dans cette méthode, on adopte le choix suivant :

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

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

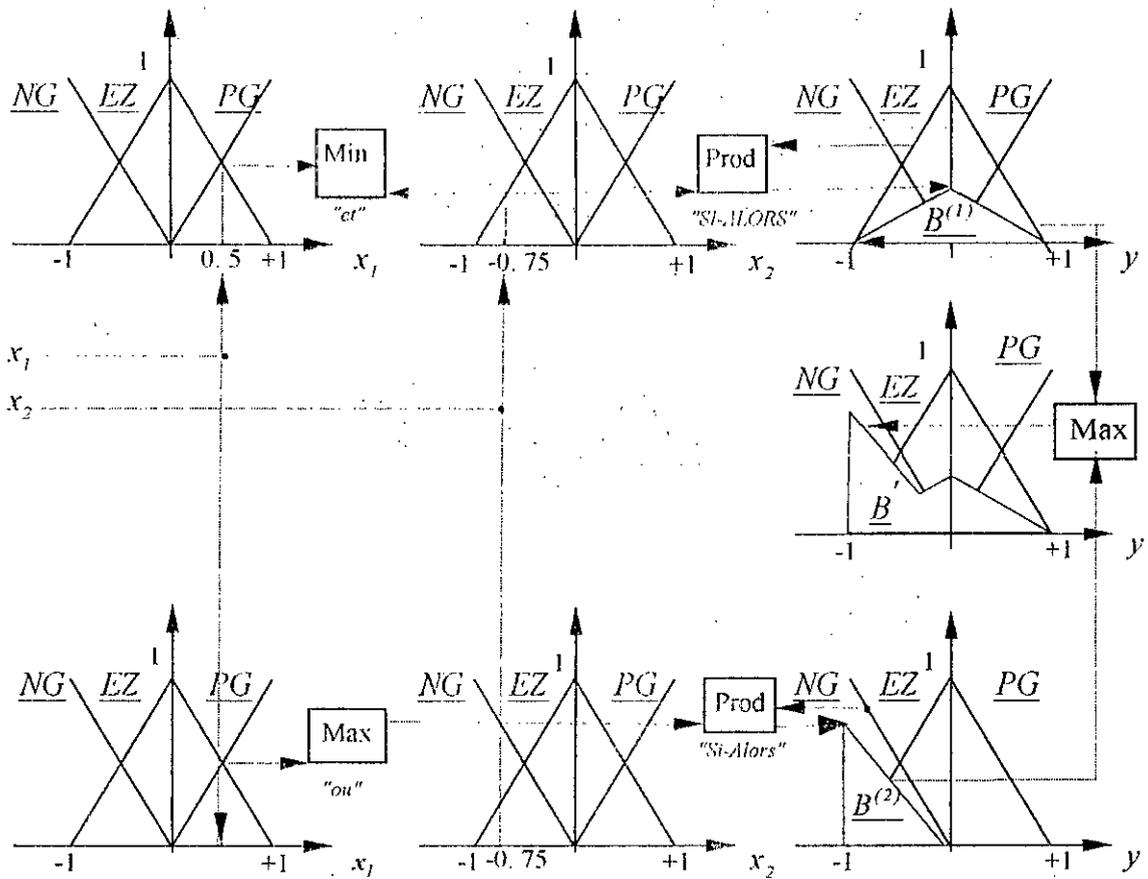


figure (1.6) : Méthode d'inférence "Max-Prod".

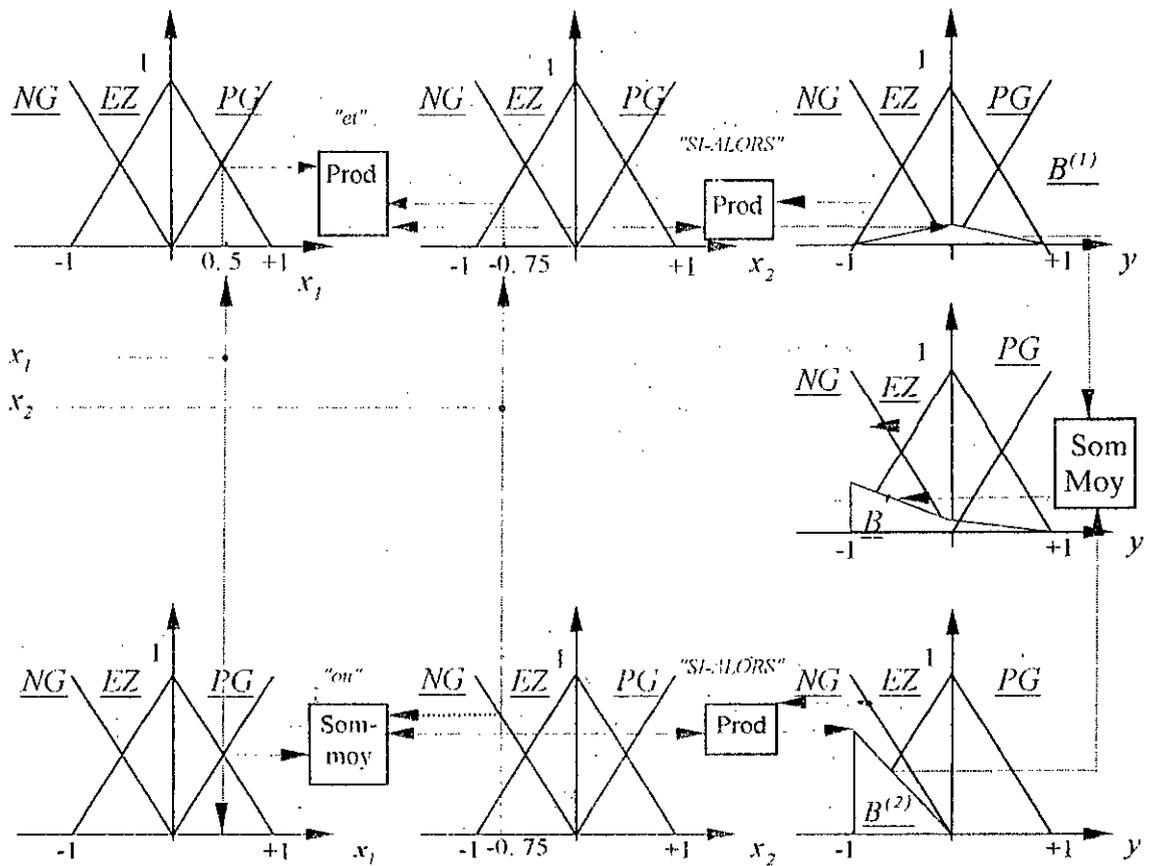


figure (1.7) : Méthode d'inférence "Som-Prod".

$$\mu_{B'}(y) = \sum_{l=1}^k \text{Sup}_{(x_1, \dots, x_n)} \left[\left[\mu_{F_1'}(x_1) \times \dots \times \mu_{F_n'}(x_n) \right] \times \left[\mu_{F_1'}(x_1) \times \dots \times \mu_{F_n'}(x_n) \right] \times \mu_{B'}(y) \right] / k \quad (I.21a)$$

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

$$\mu_{B'}(y) = \sum_{l=1}^k \left[\mu_{B'}(y) \times \left[\prod_{i=1}^n \mu_{F_i}(x_i^*) \right] \right] / k \quad (I.21b)$$

Les différentes étapes de l'inférence par la méthode Som-Prod, sont illustrées par la figure (I.7).

I.4 Application de la logique floue à la commande

La nécessité de réaliser des systèmes de commande complexes et sophistiqués a donné naissance à plusieurs techniques. Parmi ces techniques, on peut citer : la commande optimale, la commande adaptative, la commande robuste, la commande par réseaux de neurones... etc. Le développement de la plupart de ces techniques se base principalement sur des connaissances objectives (modèles mathématiques de connaissance ou de représentation), ces techniques permettent de réaliser de bonnes performances.

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

Le figure (I.8) montre un régulateur à base de la logique floue (ou régulateur flou) dans une boucle de commande. Son schéma synoptique est illustré à la figure (I.9)[DRI93].

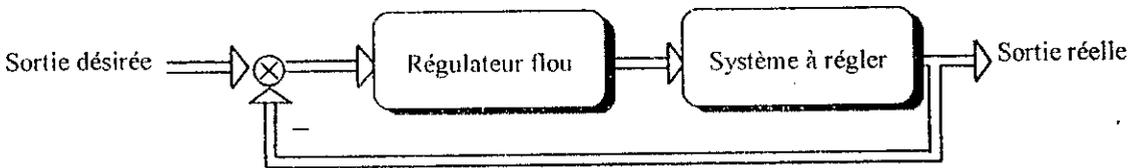


Figure (I.8) : Schéma d'un système bouclé avec régulateur flou

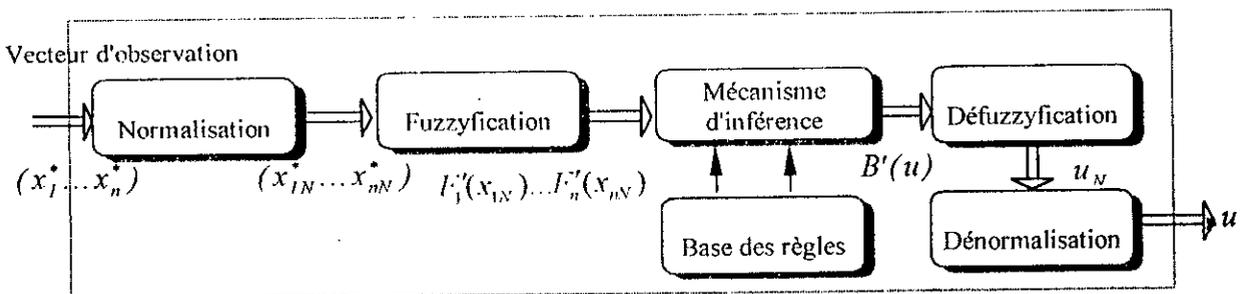


figure (I.9) : Schéma synoptique d'un régulateur flou

Dans ce schéma on distingue les modules suivants :

- ◆ Module de normalisation ;
- ◆ Module de fuzzyfication. ;
- ◆ Module de la base des règles ;
- ◆ Module d'inférence ;
- ◆ Module de défuzzyfication ;
- ◆ Module de dénormalisation.

1.4.1 Description des modules d'un régulateur flou

1.4.1.1 Module de normalisation :

Le vecteur d'entrée du module de normalisation $(x_1^* \dots x_n^*)$, qu'on appellera vecteur d'observation, est de composantes dépendantes des états du processus. Souvent, on choisit un vecteur d'observation à deux composantes (x_1^*, x_2^*) . Selon le choix de ce vecteur, on distingue les deux cas suivants :

1° Erreur entre la sortie désirée q^d et celle du procédé q et sa dérivée.

$$x_1^* = e(t) = q_d(t) - q(t), x_2^* = \dot{e}(t) = \dot{q}_d(t) - \dot{q}(t)$$

2° Erreur et sa variation entre deux instants d'échantillonnage :

$$x_1^*(t) = e(k) \hat{=} e(kT_0)$$

$$x_2^* = \Delta e(k) \hat{=} e(k) - e(k-1)$$

T_0 est la période d'échantillonnage avec $k = 1, 2, \dots$

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

$$x_{iN}^* = \frac{x_i^*}{N_{x_i}} \quad i = 1, \dots, n \tag{1.22}$$

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

N_{x_i} : facteur de normalisation.

1.4.1.2 Module de fuzzyfication :

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

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

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

En présence de bruit dans les mesures x_i^* , ce choix est déconseillé [DRI93][MEN95]. Il sera préférable de choisir des fonctions d'appartenance non-singleton (triangulaire, trapézoïdale...). Cela permettra d'introduire un préfiltrage sur les mesures entachées de bruits [MEN95]. La figure(1.10) montre deux types de fuzzyfication : singleton et non-singleton.

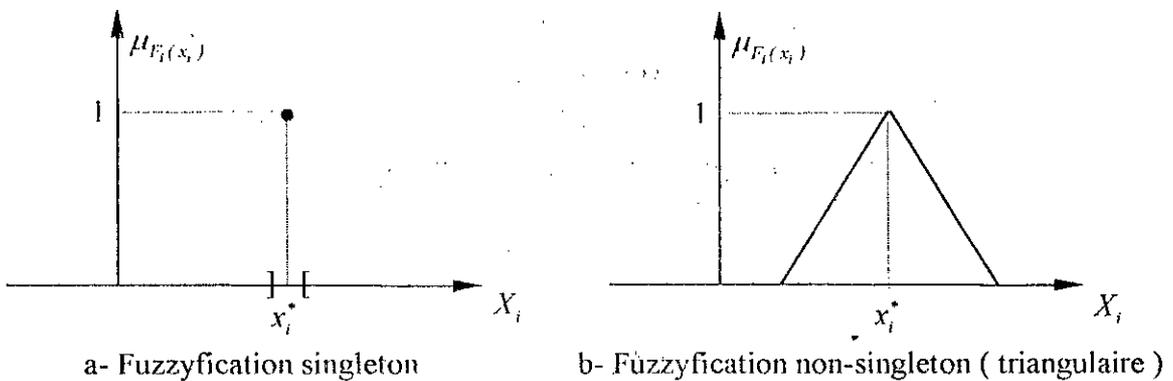


figure (1.10) : Exemples de Fuzzyfication

I.4.1.3 Module de la base des règles :

a°/ Base de données liées aux règles :

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

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

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

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

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

Quant aux variables linguistiques, le nombre des valeurs ainsi que leurs fonctions d'appartenance associées, dépendent essentiellement de :

- 1- La description disponible du comportement que doit réaliser le régulateur. Cette description est obtenue à partir d'un opérateur humain (un automaticien ou un expert).
- 2- La précision envisagée dans le réglage : une augmentation du nombre des valeurs (ou classes linguistiques des variables d'entrée et/ou de sortie) améliorent relativement la précision du réglage [MEN95].

Le nombre total des règles (nr) est lié aux nombres des classes des variables d'observation nf_i , par la relation suivante :

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

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

Dans les régulateurs flous, on choisit souvent, un nombre de classes égale à 3, 5 ou 7, pour un régulateur à deux variables d'observations, on aura respectivement 9, 25, 49 règles.

Des distributions équidistantes en cas de 3, 5 et 7 classes, sont montrées à la figue (I.11)

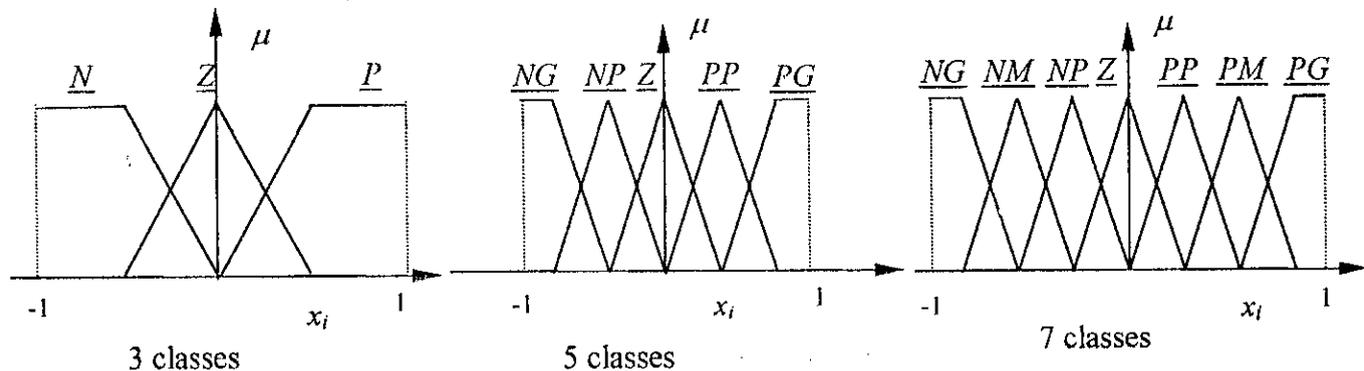


figure (I.11) : Distributions des différentes classes

De point de vue commande, le chevauchement entre deux ou plusieurs fonctions d'appartenance, permet de distribuer la décision (la commande) sur plusieurs règles, ce qui donne au régulateur flou une certaine robustesse [MEN95].

b°/ Dédution du jeu des règles :

Il existe plusieurs approches pour la synthèse de l'ensemble des règles de décision, sous forme d'expressions conditionnelles. Leur combinaison peut s'avérer nécessaire pour construire une approche efficace [LOU97][DRI93]. Parmi ces approches on cite :

- ♦ **Première approche :** La première approche considérée comme la plus utilisée, consiste à dériver les règles à partir des connaissances basées sur l'expérience d'un expert.

Généralement, elle produit une version prototype de règles propre à un domaine d'application spécifique. L'ajustement des fonctions d'appartenance ainsi que les règles peut devenir une étape nécessaire si des meilleures performances sont recherchées.

◆ **Deuxième approche :**

Dans cette approche, on utilise une description par des relations floues, comme un modèle flou du processus sous contrôle. Sur la base de ce modèle, on détermine l'ensemble des règles ou explicitement des relations floues représentatives de ces règles. Une application de cette approche peut être trouvée dans [DRI93].

◆ **Troisième approche :**

Sur la base d'un apprentissage, auto-organisation ou optimisation, cette approche consiste à créer un régulateur flou apte à générer et à modifier les règles floues de contrôle. Une première application est rapportée par Mamdani et Procyk dans [PRO79], qu'on a appelée "*Self Organisation Controller*" (SOC). On trouve plusieurs applications du SOC au problème de contrôle en robotique telles que [SCH85][TAN88]. Quant à l'apprentissage, on trouve celles présentées dans [JAN92][HOR92].

1.4.1.4 Module d'inférence :

Les deux principales fonctions de ce module sont :

1. Inférer les ensembles flous représentatifs de la valeur actuelle du vecteur d'observation (issu du module de fuzzyfication) avec chaque règle.
2. Agrégation des différents ensembles inférés des différentes règles pour déduire un seul ensemble. Ces deux fonctions peuvent être réalisées par un des mécanismes d'inférence présentés dans la section (§1.3.5), à savoir :

- ◆ Mécanisme d'inférence "*Max-min*"
- ◆ Mécanisme d'inférence "*Max-pro*"
- ◆ Mécanisme d'inférence "*Som-pro*"

1.4.1.5 Module de défuzzyfication :

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

Parmi les méthodes de défuzzyfication les plus utilisées, nous citons [LOU97] :

- ◆ Méthode du maximum ;
- ◆ Méthode de la moyenne des maxima ;
- ◆ Méthode du centre de gravité ;
- ◆ Méthode des hauteurs.

a°/ Méthode du maximum :

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

Remarque :

Bien que cette méthode simplifie la procédure de la fuzzyfication, elle présente certains inconvénients :

- 1- Elle ignore la distribution de la fonction d'appartenance de l'ensemble inféré. Seulement le point correspondant au maximum est considéré. Si l'ensemble inféré est celle montré à la figure (I.12a) la défuzzyfication par cette méthode, produit une commande normalisée $u_n = u_0$. Le reste de l'univers de discours est ignoré.

2- Si le maximum est atteint pour plusieurs points, un problème de choix de la commande se pose (voir figure (I.12b)).

3- Si un changement de domaine entre règles provoque des sauts entre les points correspondants au maximum, cela se traduit par des sauts de commande, ce qui est indésirable pour les circuits de réglage (voir figure (I.12c)) [BUH94].

b°/ Méthode de la moyenne des maxima :

Elle consiste à prendre la moyenne des commandes associées aux points du maximum de l'ensemble inféré. Cette méthode permet de résoudre le problème posé par la méthode du maximum lors de la présence de plusieurs maxima. Seulement, il existe le cas tel que montré à la figure (I.13), où la commande après défuzzification correspond à un degré d'appartenance à l'ensemble inféré nul, ce qui ne peut être justifié physiquement

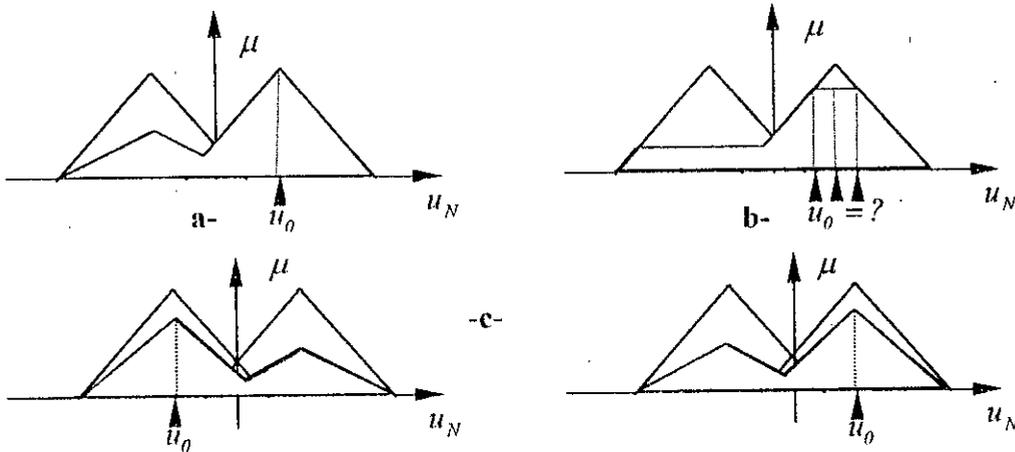


figure (I.12) : Représentation graphique des inconvénients de la méthode du maximum.

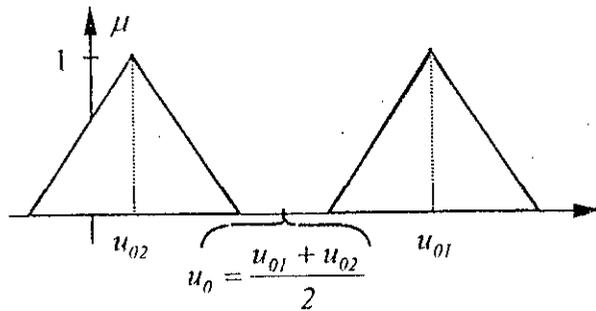


figure (I.13) : Cas particulier de défuzzification par la méthode de la moyenne des maxima.

c°/ Méthode du centre de gravité :

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

$$u_N = \frac{\int_{u_a}^{u_b} u \mu_B(u) du}{\int_{u_a}^{u_b} \mu_B(u) du} \tag{I.23a}$$

Dans le cas particulier d'un flouificateur singleton, et un mécanisme d'inférence de type "Somprod", la fuzzyfication par la méthode du centre de gravité conduit à une expression simple. D'après les équations (I.21b) et (I.23a), la commande est donnée par :

$$u_N = \frac{\sum_{l=1}^k C^{(l)} S^{(l)} \left(\prod_{i=1}^n \mu_{F_i^{(l)}}(x_{iN}^*) \right)}{\sum_{l=1}^k S^{(l)} \left(\prod_{i=1}^n \mu_{F_i^{(l)}}(x_{iN}^*) \right)} \quad (1.23b)$$

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

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

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

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

L'avantage d'utiliser l'expression (1.23a,b), c'est qu'une fois avoir choisit les fonctions d'appartenance associées à la commande, les paramètres $S^{(l)}, C^{(l)} l=1...k$ qui peuvent être déterminées a priori. Donc la détermination de la commande nécessite seulement le calcul des degrés d'activation des différentes

règles $\left(\prod_{i=1}^n \mu_{F_i^{(l)}}(x_{iN}^*) \right)_{l=1...k}$

d°/ Méthode des hauteurs :

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

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

$$u_N = \frac{\sum_{l=1}^k C^{(l)} \times \mu_{B^{(l)}}(C^{(l)})}{\sum_{l=1}^k \mu_{B^{(l)}}(C^{(l)})} \quad (1.24)$$

La figure (1.14) montre une représentation graphique de la méthode des hauteurs.

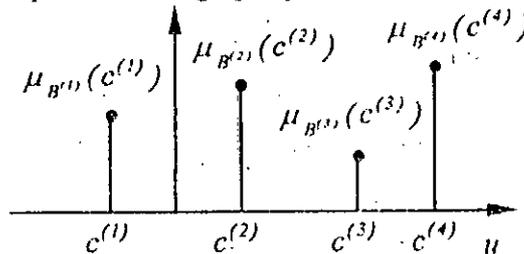


figure (1.14) : Représentation graphique des hauteurs

Remarque :

La défuzzyfication par la méthode du centre de gravité produit un lissage sur la commande, tous en consommant un temps de calcul important.

L'utilisation d'un fuzzyficateur de type singleton, un mécanisme d'inférence "Som-prod" réduit considérablement le calcul. Cependant la méthode du moyen des maxima fournit une meilleure performance transitoire alors que celle du centre de gravité est plus performante en régime permanent [LOU97].

1.4.1.6 Dénormalisation :

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

Dans la plupart des cas, on utilise une transformation linéaire (amplification ou atténuation). La commande effective est donnée par :

$$u = u_N \times N_u \quad (1.25)$$

où N_u est un facteur de dénormalisation qui dépend de l'application envisagée.

Remarque :

Si la variable de commande dans la conséquence des règles de décision est l'incrément de la commande Δx_m , l'expression de la commande effective devient :

$$u(k) = u(k-1) + \Delta u_N(k) \times N_{\Delta u} \quad (1.26)$$

avec, $N_{\Delta u}$: facteur de dénormalisation de Δu

$$u(k) \hat{=} u(t) \Big|_{t=kT} \quad \text{avec } T \text{ la période d'échantillonnage}$$

$$\Delta u_N(k) = \Delta u_N(t) \Big|_{t=kT} \quad \text{l'incrément de la commande normalisée.}$$

1.4.2 Modèles flous :

En se basant sur les notions de la théorie des ensembles flous, les systèmes d'inférence flous ont constitué un moyen efficace pour exploiter les deux types de connaissances (linguistique et numérique). Grâce à la propriété d'approximation des fonctions, les systèmes à logique floue permettent la modélisation de ces connaissances sous forme de modèles flous. Dans cette section, on présente les trois types de modèles flous les plus utilisés ainsi que leurs principales propriétés.

1.4.2.1 Types de modèles flous :

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

a°/ Modèle de Mamdani :

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

$$SI (X_1 \text{ est } F_1^{(l)} \dots \text{ et } X_n \text{ est } F_n^{(l)}) \text{ ALORS } (Y \text{ est } B^{(l)}) \quad l=1 \dots k \quad (1.27)$$

Ce type de modèle est celui étudié jusqu'à maintenant. L'expression de sa sortie est obtenue en appliquant une des méthodes de défuzzyfication sur l'expression de l'ensemble inféré donnée par (1.21b).

b°/ Modèle de Sugeno :

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

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

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

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

$$y(x_1, \dots, x_n) = \frac{\sum_{l=1}^k \left[\prod_{i=1}^n \mu_{F_i^{(l)}}(x_i) \right] \times f_l(x_1, \dots, x_n)}{\sum_{l=1}^k \left[\prod_{i=1}^n \mu_{F_i^{(l)}}(x_i) \right]} \quad (I.28)$$

Un modèle de type Sugeno peut être considéré comme un cas particulier des modèles de type Mamdani. En effet, si on considère les règles de Mamdani données par (I.27) où les ensembles flous des conséquences sont définis par des singletons tels que :

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

Si on adopte un flouificateur singleton, un mécanisme d'inférence *Som-prod* et défuzzification par la méthode du centre de gravité, on obtient l'expression de sortie donnée par (I.28).

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

$$f_l(x_1, \dots, x_n) = a_{0l} + \sum_{i=1}^n a_{il} x_i \quad l = 1, \dots, k \quad (I.29a)$$

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

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

$$f = z^T A \quad z^T = [1 \ x_1 \ \dots \ x_n] \quad A = \begin{bmatrix} a_{01} & \dots & \dots & a_{0k} \\ a_{11} & \dots & \dots & a_{1k} \\ \vdots & \dots & \dots & \vdots \\ a_{n1} & \dots & \dots & a_{nk} \end{bmatrix} \quad (I.29b)$$

d'après (I.28) et (I.29a,b), l'expression de la sortie y s'écrit :

$$\begin{cases} y = z^T A \xi \\ \xi^T = [\xi_1 \ \dots \ \xi_k] \end{cases} \text{ avec } \xi_j = \frac{\prod_{i=1}^n \mu_{F_i^{(j)}}(x_i)}{\sum_{l=1}^k \prod_{i=1}^n \mu_{F_i^{(l)}}(x_i)} \quad j = 1, \dots, k \quad (I.30a)$$

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

$$\begin{cases} y = z^T A \xi \\ z^T = [1] \\ A = [a_{01} \ \dots \ a_{0k}] \\ \xi^T = [\xi_1 \ \dots \ \xi_k] \end{cases} \text{ avec } \xi_j = \frac{\prod_{i=1}^n \mu_{F_i^{(j)}}(x_i)}{\sum_{l=1}^k \prod_{i=1}^n \mu_{F_i^{(l)}}(x_i)} \quad j = 1, \dots, k \quad (I.30b)$$

Selon les expressions de sorties (I.30a) et (I.30b), le choix des fonctions polynomiales permet d'obtenir un paramétrage linéaire par rapport aux paramètres des polynômes. Cette propriété intéressante sera exploitée dans les chapitres III et IV pour le développement d'algorithmes de commande floue adaptative.

Remarque

Dans un modèle de Sugeno d'ordre zéro, un meilleur lissage de la sortie en fonction des variables d'observations, est obtenu en élargissant les zones de chevauchement entre les ensembles flous de ces variables[JAN95].

c°/ Modèle de Tsukamoto :

Dans ce type de modèle, les conséquences des règles contiennent des variables linguistiques. Les règles ont les formes (I.27). La sortie est déterminée par l'expression suivante : [JAN95]

$$y = \frac{\sum_{l=1}^k \left[\prod_{i=1}^n \mu_{F_i^{(l)}}(x_i) \right] \times \mu^{-1_{B^{(l)}}} \left(\prod_{i=1}^n \mu_{F_i^{(l)}}(x_i) \right)}{\sum_{l=1}^k \left[\prod_{i=1}^n \mu_{F_i^{(l)}}(x_i) \right]} \quad (I.31)$$

où $\mu^{-1_{B^{(l)}}}(\cdot)$ est la fonction inverse de $\mu_{B^{(l)}}(\cdot)$, donc : $\mu^{-1_{B^{(l)}}}(\mu_{B^{(l)}}(y)) = y$

Pour garantir l'existence de la valeur $\mu_{B^{(l)}}^{-1} \left(\prod_{i=1}^n \mu_{F_i^{(l)}}(x_i) \right)$, il faut que $\mu_{B^{(l)}}(\cdot)$ soit définie sur l'intervalle $[0, 1]$, tandis que l'unicité de cette valeur est assurée si la fonction $\mu_{B^{(l)}}(\cdot)$ est monotone.

I.4.2.2 Propriétés des modèles flous :

Les modèles flous possèdent certaines propriétés intéressantes qui peuvent être exploitées en identification ou en commande, on note :

♦ Plusieurs études ont montré que certaines structures (à base de réseaux de neurones, à base des fonctions d'ondellettes...), sont capables d'approximer avec une précision arbitraire, n'importe quelle fonction non linéaire suffisamment lisse. Cette même propriété a été prouvée pour des modèles flous tels que [MEN95] :

- ♦ Modèle de Mamdani utilisant un fuzzyficateur singleton, implication et inférence produit, fonctions d'appartenance gaussiennes et une défuzzyfication par la méthode des hauteurs.
- ♦ Modèle de Mamdani utilisant un fuzzyficateur singleton, mécanisme d'inférence *Somprod*, et défuzzyfication par la méthode du centre de gravité.
- ♦ Modèle de Segeno : il est considéré comme un cas particulier du modèle précédant de Mamdani.

En général, un modèle flou peut être considéré comme un modèle global constitué de plusieurs modèles locaux chacun est activé dans une région définie par les ensembles flous de l'antécédente d'une règle. Pour montrer cela, considérons un modèle de Sugeno (donné par l'équation (I.28)), écrit sous la forme suivante :

$$y = \sum_{l=1}^k \left(\frac{\prod_{i=1}^n \mu_{F_i^{(l)}}(x_i)}{\sum_{j=1}^k \left[\prod_{i=1}^n \mu_{F_i^{(j)}}(x_i) \right]} \right) f_l(x_1, \dots, x_n) \quad (I.32a)$$

$$y = \sum_{l=1}^k \xi_l f_l(x_1, \dots, x_n) \quad (I.32b)$$

donc la sortie est une combinaison linéaire des différentes fonctions $(f_l)_{l=1, \dots, k}$, pondérées par les poids $(\xi_l)_{l=1, \dots, k}$. Chaque poids ξ_l est non nul seulement lorsque la règle "l" est activée, ou $\prod_{i=1}^n \mu_{F_i^{(l)}}(x_i) \neq 0$.

Donc, à facteur près, chaque fonction f_l peut être considérée comme un modèle local de la relation $y=y(x_1, \dots, x_n)$ dans une région définie par les ensembles flous de l'antécédente de la règle "l". Ainsi, l'équation (I.32a) peut être interprétée comme une manière d'*interpolation* entre ces différents modèles [PIC97].

- ♦ L'utilisation d'un fuzzyficateur non-singleton dans un modèle de Mamdani peut être interprété comme un préfiltrage des variables d'observation[MEN95].
- ♦ Le dénominateur dans l'expression de sortie du modèle flou, réalise une opération de normalisation, qui dépend des ensembles flous des antécédentes. Cette normalisation découle de la définition du

système d'inférence (fuzzyficateur, mécanisme d'inférence et défuzzyficateur). Cependant, à travers plusieurs travaux portés sur les systèmes à base de réseaux de neurones, l'introduction d'une telle normalisation (par abstraction) sur les fonctions d'activation a amélioré nettement l'apprentissage des réseaux [MEN95].

1.4.3 Commande décentralisée floue

Pour un système composé de plusieurs sous-systèmes interconnectés, la commande décentralisée a pour objectif de concevoir des régulateurs locaux, chacun réagit à partir des informations du sous-système correspondant. Ces régulateurs coopèrent pour stabiliser le système global.

Un robot manipulateur, peut être considéré comme la composition de plusieurs articulations couplées, chacune équipée d'un actionneur. Dans ce cas la commande décentralisée en poursuite de trajectoires ou de régulation doit assurer la stabilité du système global via des régulateurs associés à chaque articulation et n'ayant accès qu'aux mesures locales (position, vitesse, accélération,...).

Si les régulateurs sont à base de la logique floue, on obtient une Commande Décentralisée Floue. Le schéma de principe est montré dans la figure (I.15)

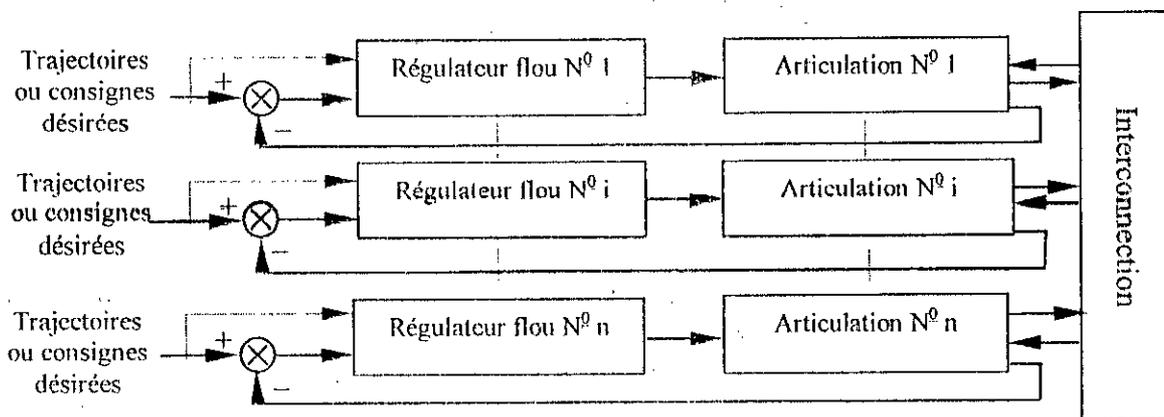


figure (I.15) : Schéma de principe de la Commande Décentralisée Floue.

En général, l'intervention du raisonnement flou, dans un régulateur s'effectue de deux manière principales [DRI93]:

1.4.3.1 Intervention indirecte :

Un modèle flou superviseur adapte les paramètres d'un régulateur classique sur la base d'un ensemble de règles floues prédéfinis. Dans [TZA90][ZHA93][KET92], les auteurs utilisent des modèles flous pour la supervision des gains proportionnel, intégral et dérivé d'un régulateur PID appliqué à plusieurs exemples de fonctions de transfert. La figure (I.16) montre le schéma de principe d'une supervision floue d'un PID.

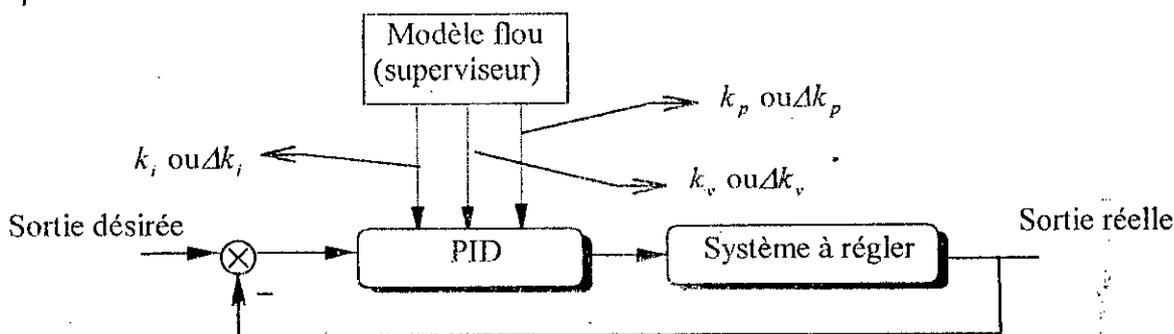


figure (I.16) : Schéma de principe d'une supervision floue d'un PID

I.4.3.2 Intervention directe :

Dans une intervention directe, le modèle flou se base sur un ensemble de règles floues établies préalablement, pour générer une commande. On distingue deux cas :

a°/ Régulateur flou simple :

Dans ce cas, le régulateur est constitué uniquement préalablement. (voir figure (I.17a)). Si des meilleurs performances sont recherchées, les paramètres du modèle flou (facteurs de normalisation et de dénormalisation, paramètres des fonctions d'appartenance,...) peuvent être adaptés hors ligne ou en temps réels. Pour cela, on utilise une des procédures d'optimisation (méthode du gradient [DRI93], méthode des moindres carrés [MEN95],...) pour ajuster les différents paramètres. La figure (I.17a,b) montre deux cas de régulateurs flous simples.

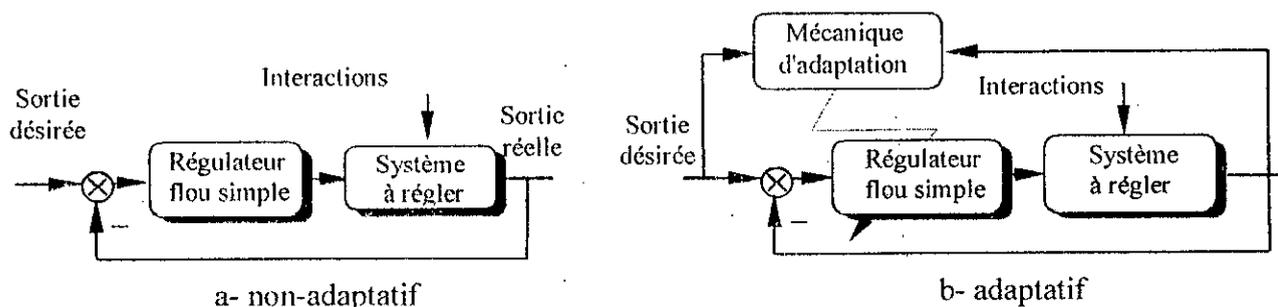


figure (I.17): Régulateur flou simple

b°/ Régulateur flou hybride :

La commande dans un régulateur flou hybride est générée après combinaison des sorties de plusieurs blocs :

- Bloc d'un modèle flou;
- Bloc dans la chaîne directe ("feedforward");
- Bloc dans la chaîne de retour ("feedback");

Une première idée consiste à attribuer au bloc de combinaison une fonction de commutation entre modèles flous ou entre modèle flou et régulateur conventionnel. Dans [KET92][LOU97], on présente une structure où le bloc de combinaison réalise une commutation entre deux modèles de type Mamdani (commande multi-échelles). Le premier modèle réalise un réglage grossier lorsque la variable d'observation (Erreur) est jugée importante. Lorsque ce dernier devient de faibles valeurs, on commute sur un réglage fin. La figure (I.18a) illustre le schéma de principe. Dans une deuxième structure, Loudini [LOU97] propose une commande qui résulte de la combinaison d'un régulateur classique de type PI intervenant pour des grandes valeurs de l'erreur et d'un régulateur flou de type Mamdani destiné aux petites valeurs de l'erreur. Le schéma de principe est illustré à la figure (I.18b).

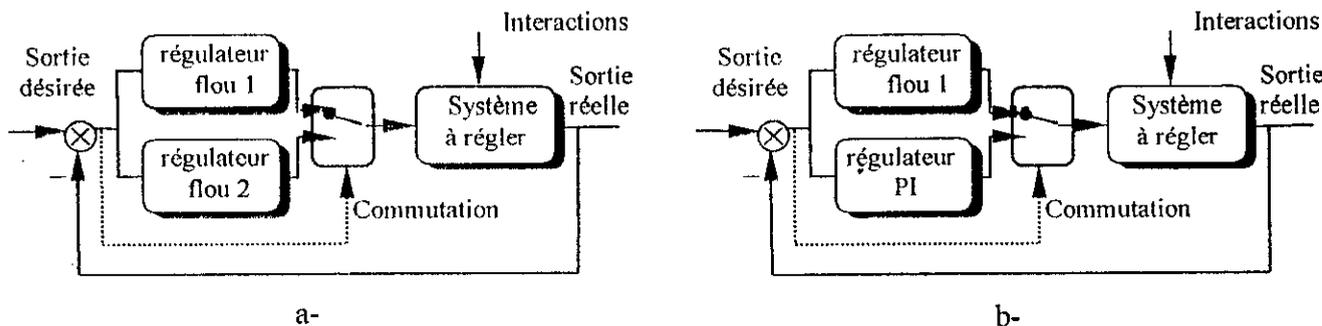


figure (I.18) : Exemples de régulateurs flous hybrides

Dans d'autres travaux [SPO95a,b] [TRE97], les auteurs synthétisent des lois d'adaptation pour l'ajustement de certains paramètres du modèle flou, tous en assurant la stabilité du système en boucle fermée. Le modèle flou intervient pour compenser des perturbations indésirables ou identifie des dynamiques inconnues, grâce à leur capacité d'approximation de fonctions non linéaires. La figure (1.19) illustre le schéma d'une commande à base de régulateur flou hybride.

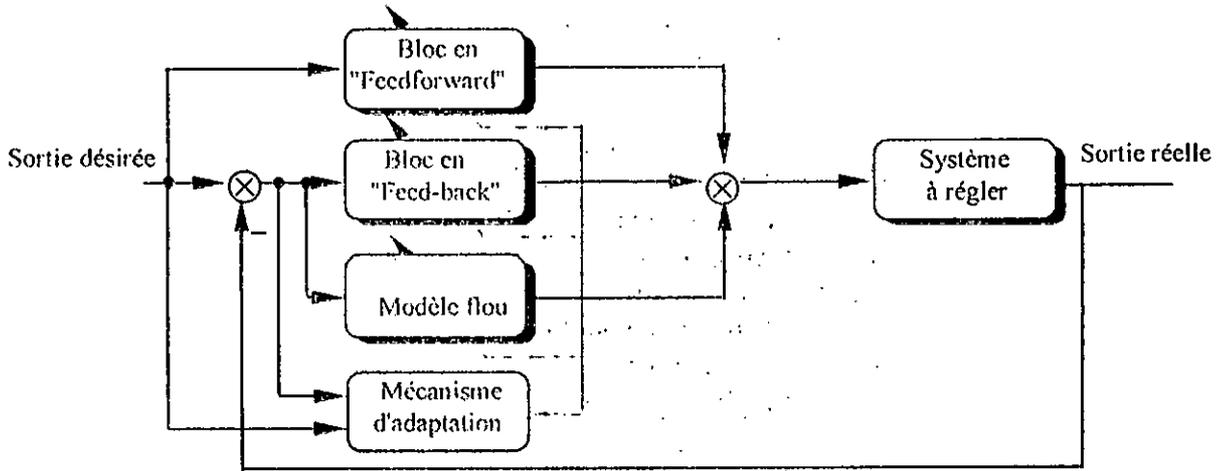


figure (1.19) : Un schéma de commande à base de régulateur flou hybride

1.5 Conclusion :

Dans ce chapitre, nous avons examiné les outils de base de la logique floue nécessaires à la compréhension des systèmes et régulateurs flous.

Selon la nature des conséquences des règles floues, on distingue deux principaux types de modèles flous :

- Le modèle de Mamdani dont les conséquences des règles sont linguistiques en terme de la commande.
- Le modèle de Sugeno, caractérisé par des règles dont les conséquences sont des fonctions numériques. Ces fonctions relient la commande aux variables d'observation.

Lorsqu'on dispose d'une description linguistique des actions d'un opérateur à modéliser, seul le modèle de Mamdani peut être utilisé. Cependant, une information numérique sur le comportement réalisé par un processus, par exemple des données entrée-sortie, peut être modélisée par les deux types de modèles. Dans ce cas, il est nécessaire d'adopter des modèles ajustables ayant la caractéristique d'approximation de fonctions. Si en plus les modèles apparaissent linéaires en paramètres à ajuster, il est tout à fait concevable d'utiliser les techniques classiques d'identification paramétrique (par exemple la technique des moindres carrés ou toutes leurs variantes possibles) et de profiter aussi de toute l'efficacité de ces méthodes pour obtenir un apprentissage rapide et précis.

Chapitre II

Commande floue associée à un réglage à modes glissants

Chapitre II :

Commande floue associée à un réglage à modes glissants

II.1 Introduction

Plusieurs études menées sur la commande en poursuite des robots manipulateurs, ont montré la capacité des régulateurs conventionnels de type PD (Proportionnel-Dérivé) à stabiliser la dynamique de l'erreur au risque de présence d'erreurs permanentes [LEW93]. Puisque la sortie d'un régulateur PD est nulle lorsque l'erreur et sa dérivée sont nulles, l'erreur permanente est nécessaire pour générer une commande suffisante au maintien d'équilibre de la liaison en régime permanent.

Pour éliminer cette erreur, on peut recourir à la compensation directe de la commande en régime permanent. Mais comme cette dernière dépend de la configuration du robot, elle doit être adaptée aux différentes configurations possibles ce qui est difficile à réaliser.

Une deuxième solution consiste à réaliser indirectement cette compensation par insertion d'une action d'intégration dans le régulateur pour obtenir un régulateur de type PID (proportionnel-intégral-dérivé). La présence de l'action d'intégration assure l'élimination des erreurs permanentes.

Autre que les stratégies de commande conventionnelles, la commande à base de la logique floue offre une alternative de synthèse des régulateurs, dans le sens où elle permet d'incorporer des connaissances linguistiques. La majorité des algorithmes de commande floue sont basés sur le schéma proposé par Mamdani pour les systèmes mono-entrée mono-sortie ("*SISO*") [LOU97]. Dans ce schéma, on utilise l'erreur et sa dérivée (ou sa variation) comme variables d'entrée du régulateur flou. Selon que la sortie du régulateur est considérée comme la commande à appliquer au processus, ou sa variation, on obtient respectivement un régulateur de type PD (PD-flou) ou régulateur de type PI (PI-flou) [DRI93].

La construction d'un régulateur PID flou peut s'effectuer de deux manières : Soit par la mise en parallèle d'un régulateur PD flou avec un PI flou, ou par l'adoption de l'intégrale de l'erreur comme une troisième variable d'entrée (en plus des variables de l'erreur et de sa dérivée). Dans ce cas l'élaboration des règles floues nécessite un raisonnement avec trois variables d'entrée ce qui est difficile à effectuer.

Cette partie de notre travail est consacrée principalement à la conception d'un régulateur flou permettant de simuler la fonction des régulateurs conventionnels de la famille PI, PD, PID⁽ⁿ⁾ ($n \geq 1$). Inspiré du principe des régulateurs à modes glissants avec couche bornée ("*Boundary Layer*"), nous proposons un régulateur flou dont la commande est déterminée en fonction de l'éloignement du vecteur d'état par rapport à une surface de glissement prédéfinie. Deux variables floues sont introduites permettant ainsi de quantifier cet éloignement dans un espace linguistique. Des régulateurs flous de type PI, PD et PID⁽ⁿ⁾ ($n \geq 1$) peuvent être obtenus par une simple modification de l'expression de la surface de glissement.

Le présent chapitre est organisé comme suivant :

Dans la première section, on présente le principe des régulateurs à modes glissants sans et avec couche bornée ("*Boundary Layer*"). Le régulateur flou proposé est introduit dans la deuxième section avec une description de ces différents modules. Des propriétés et des avantages du nouveau régulateur sont discutées dans la troisième section. Dans la quatrième section, nous présentons l'application du

régulateur flou étudié à la commande décentralisée des robots manipulateurs. Des résultats de simulation issus de l'application de la technique développée au robot PUMA560 sont présentés. Nous terminons le chapitre par une conclusion.

II.2 Régulateur à modes glissants

Lors de la conception d'une commande robuste pour un processus à perturbation ou à paramètres variables, l'utilisation des régulateurs à modes glissants s'avère utile [BUH86].

En se basant sur la connaissance des bornes supérieures de certains paramètres et fonctions dans le modèle du processus, le régulateur à modes glissants génère une commande permettant de ramener l'état du système vers un hyperplan (ou une surface) prédéfini dans l'espace de phase, et cela même en présence de perturbations externes ou de fluctuations au niveau des paramètres du processus. Lorsque l'état du système est maintenu sur cet hyperplan, la dynamique de convergence de la sortie du processus vers la sortie désirée peut être imposée via le choix des paramètres de la surface de glissement.

On considère le système mono-entrée mono-sortie, décrit par le modèle suivant :

$$\begin{cases} \dot{X}_s = f_s(X_s, u) + d(t) \\ y = h_s(X_s) \end{cases} \quad (\text{II.1})$$

avec, $X_s^T = [x_1 \dots x_n]$: vecteur d'état du processus. y : la sortie du processus ;

u : la commande du processus ;

$d(t)$: une perturbation externe;

f_s, h_s : des fonctions non linéaires.

Soit $X_d^T = [x_{d1} \dots x_{dn}]$, Un vecteur des états désirés (y compris la sortie y_d). Le dimensionnement d'un régulateur à modes glissants, de manière que les états du système X_s convergent vers les états désirés X_d , s'effectue en deux étapes :

1. Sélection de la surface de glissement :

La surface de glissement caractérise les systèmes de commande à base de régulateur à modes glissants. La sélection consiste à déterminer un hyperplan sur lequel l'état du système glisse vers l'état d'équilibre lorsque le système est soumis à une loi de commande issue d'un régulateur à modes glissants. En général, la surface de glissement est une fonction non linéaire qui dépend des états du système et des états désirés.

Un choix simple qui est souvent adopté, est celui d'une surface linéaire. Dans ce cas la surface s'écrit :

$$S(X_s, X_d) = \sum_{j=1}^n \lambda_{sj} x_j + \lambda_{dj} x_{dj} \quad (\text{II.2})$$

avec, $\lambda_{sj} \in \mathbb{R}$, $j = 1, \dots, n$: les gains d'intervention des états du système.

$\lambda_{dj} \in \mathbb{R}$, $j = 1, \dots, n$: les gains d'intervention des états désirés.

Pour des conditions initiales $x_j(0)$, $j = 1, \dots, n$ arbitraires, les gains λ_{sj} et λ_{dj} , $j = 1, \dots, n$ doivent être choisis de manière que chaque état x_j converge vers l'état désiré correspondant x_{dj} lorsque l'état du système est maintenu sur la surface de glissement donnée par $S(X_s, X_d) = 0$.

Pour montrer cela, considérons le cas particulier où :

$$(\dot{x}_j = x_{j+1}; \dot{x}_{dj} = x_{dj+1}; \lambda_{sj} = -\lambda_{dj} = -\lambda_j), \quad j = 1, \dots, n-1,$$

$$\text{donc, } S(X_s, X_d) = 0 \Leftrightarrow \lambda_{n-1} e^{(n-1)} + \lambda_{n-2} e^{(n-2)} + \dots + \lambda_1 e + \lambda_0 e = 0$$

où : $e = x_{d1} - x_1$: erreur entre sortie désirée et sortie réelle,

$e^{(j)}$: dérivée d'ordre j de l'erreur entre sortie désirée et sortie réelle.

Pour que chaque état x_j converge vers l'état désiré x_{d_j} , il faut que tous les racines de l'équation caractéristique suivante :

$$\lambda_{n-1}p^{n-1} + \lambda_{n-2}p^{n-2} + \dots + \lambda_1p + \lambda_0 = 0 \quad (\text{II.3})$$

aient des parties réelles strictement négatives.

2. Synthèse de la loi de commande :

Le choix de la surface de glissement permet d'imposer la dynamique de convergence vers l'état désiré sans qu'il assure la convergence vers la surface de glissement. Pour que l'état du processus soit toujours ramené vers la surface de glissement, la loi de commande est choisie de manière que cette surface constitue une zone attractive. Pour cela une condition suffisante dite *condition d'existence des régimes (ou modes) glissants*, consiste à choisir une loi de commande vérifiant l'inégalité suivante :

$$S(X_s, X_d) \times \dot{S}(X_s, X_d, u) \leq 0 \quad (\text{II.4})$$

avec, $\dot{S}(X_s, X_d, u) = \frac{dS}{dt}$ dérivée de la surface par rapport au temps.

Si cette inégalité est vérifiée avec une loi de commande discontinue qui change de structure selon le signe de la surface, on obtient une commande à modes glissants.

Pour certaines classes de systèmes non linéaires, la loi de commande est donnée par [DRI93] :

$$u = U_0 \text{sign}(S(X_s, X_d)) \quad (\text{II.5})$$

avec, U_0 : Une constante réelle,

$$\text{sign}(S) = \begin{cases} -1 & \text{si } S < 0 \\ 0 & \text{si } S = 0 \\ +1 & \text{si } S > 0 \end{cases}$$

D'après l'expression de la commande, on applique une commande constante de même signe que la surface. Dans le cas particulier d'un système du deuxième ordre ($n=2$), la figure (II.1) montre la trajectoire de l'état du système dans l'espace de phase lorsque l'équation caractéristique (II.3) possède des racines stables et la condition (II.4) est vérifiée. D'après cette représentation, on applique une commande positive (U_0) lorsque l'état de phase du système (état du système dans le plan de phase) est situé au-dessus de la surface de glissement. Dès que l'état passe dans la région située au-dessous de la surface de glissement, la commande devient négative ($u = -U_0$).

Grâce à ce changement de signe, l'état du système est ramené vers la surface de glissement même en présence de perturbations ou de variations paramétriques. Cela est accompagné de rapprochement vers l'origine ($\dot{e} = e = 0$) ce qui constitue le phénomène de glissement

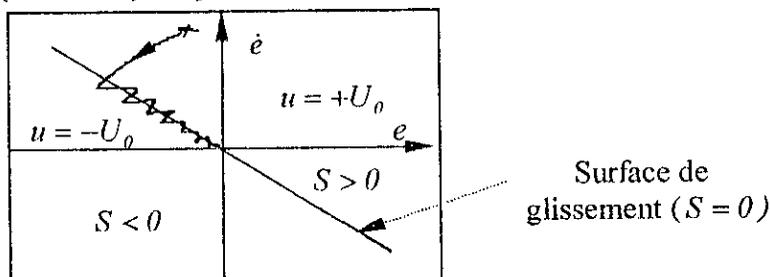


figure (II.1) : Trajectoire du système dans le plan de phase

Introduction d'une couche bornée:

La convergence des états du système vers les états désirés est assurée grâce à la structure particulière du régulateur à modes glissants. Cependant, cette structure caractérisée par une

discontinuité de la commande, peut provoquer des commutations intempestives entre les commandes positives et les commandes négatives lors du passage par la surface de glissement.

En pratique, ces commutations peuvent se traduire par des problèmes au niveau des actionneurs et du processus (par exemple excitation des dynamiques négligées). Pour palier à une telle situation, on peut remédier à l'adoption d'une zone autour de la surface de glissement dans laquelle l'évolution de la commande est progressive. Au-delà de cette zone, la commande est maintenue constante.

Si on considère une évolution linéaire en fonction de la surface, la loi de commande (II.5) devient :

$$u = U_0 \text{sat}_\Delta(S(X_s, X_d)) \tag{II.6}$$

où, sat_Δ une fonction de saturation définie par :

$$\text{sat}_\Delta(S) = \begin{cases} +1 & \text{si } S > \Delta \\ S/\Delta & \text{si } |S| \leq \Delta \\ -1 & \text{si } S < -\Delta \end{cases}$$

Δ : scalaire positif qui définit la largeur de la zone de linéarité. Cette zone appelée couche bornée est de largeur 2Δ . La loi de commande (II.6) est illustrée graphiquement à la figure (II.2).

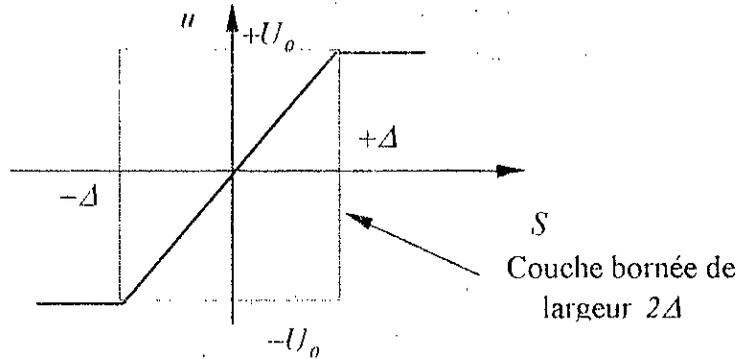


figure (II.2) : Evolution de la commande lors d'introduction de la couche bornée.

Grâce à l'utilisation de la fonction $\text{sat}_\Delta(\cdot)$, une transition brusque entre les deux valeurs $-U_0$ et $+U_0$ est remplacée par une transition progressive dans la zone de la couche bornée. Dans cette zone la commande varie linéairement en fonction de la surface, tandis qu'elle est maintenue constante en dehors de cette zone.

II.3 Régulateur flou associé à un régulateur à modes glissants

Par le biais de la logique floue, des connaissances linguistiques peuvent être exploitées pour la synthèse de régulateurs flous qui assurent l'asservissement d'un processus. Dans ce cas, une étude préliminaire pour la détermination d'un modèle mathématique précis de processus n'est plus nécessaire.

Dans le but d'exploiter des connaissances requises des techniques de commande conventionnelles, on associe aux régulateurs flous une structure similaire à celle des régulateurs classiques tels que PID ou régulateurs à modes glissants. Cela permet d'établir des directives de synthèse des règles floues utilisées dans l'algorithme de commande.

Considérons un système monovariante (mono-entrée mono-sortie) décrit par le modèle suivant:

$$\begin{cases} y^{(n)} = f_s(X_s, u) \\ y = h_s(X_s) \end{cases} \tag{II.7}$$

avec, X_s : vecteur d'état du système ; y : la sortie du système ; u : la commande à appliquée au système ; f_s, h_s : fonctions non linéaires.

Soit : $Y = [y(t) \dot{y}(t) \dots y^{(n-1)}(t)]^T$: Vecteur de la sortie du système et de ses (n-1) dérivées, supposées mesurables.

$Y_d = [y_d(t) \dot{y}_d(t) \dots y_d^{(n-1)}(t)]^T$: Vecteur de la sortie désirée et de ses (n-1) dérivées, supposées bornées

Le problème de commande consiste à déterminer la loi d'évolution de la commande u de telle manière que l'erreur de poursuite entre la sortie désirée et la sortie réelle converge asymptotiquement vers zéro.

Dans la section précédente, on a présenté une loi de commande à modes glissants modifiée par l'introduction d'une couche limite. Ainsi la commande obtenue varie linéairement en fonction de la surface dans la zone déterminée par la couche bornée tandis qu'elle est saturée en dehors de cette zone.

En réalité la fonction de saturation $sat_d(.)$ introduite dans la loi de commande (II.5), réalise une mesure de l'éloignement de l'état du système par rapport à la surface de glissement. Cette mesure est traduite par la valeur de la surface à laquelle appartient l'état actuel du système. C'est à partir de cette valeur que la commande est calculée. En effet, la distance entre l'état du système et la surface de glissement augmente suivant la valeur de la surface suscitée lorsque l'état s'éloigne de la surface de glissement. Dans le cas général, il est possible d'utiliser d'autres grandeurs que la valeur de la surface pour mesurer cette distance.

Dans cette section, on introduit une généralisation de la mesure de la distance entre une surface de glissement prédéfini et le vecteur d'état grâce à l'utilisation de deux variables de distance. L'association des variables floues aux variables de distance, nous permettra de construire un régulateur flou à base de modèle de Mamdani. La détermination des règles floues est basée sur le principe des régulateurs à modes glissants.

II.3.1 Sélection de la surface de glissement

Comme pour le cas des régulateurs à modes glissants, la sélection de la surface permet d'imposer la dynamique de l'erreur de poursuite. Un choix souvent adopté, est celui d'une surface linéaire avec intégrateur. Elle est donnée par :

$$S(e^{(n-1)}, \dots, e, (\int e)) = e^{(n-1)} + \lambda_{n-2}e^{(n-2)} + \dots + \lambda_1 \dot{e} + \lambda_0 e + \lambda_I \int_0^t e(\tau) d\tau \quad (II.8)$$

$e(t) = y_d(t) - y(t)$ est l'erreur entre la sortie désirée $y_d(t)$ et la sortie réelle $y(t)$;

$e^{(k)}$ désigne la dérivée d'ordre k de l'erreur $e(t)$ avec $k = 1, \dots, n-1$.

Afin d'assurer la stabilité de l'erreur, les paramètres $\lambda_k; \lambda_k \quad k = 0, \dots, n-2$ sont choisis de manière que l'équation caractéristique suivante :

$$p^n + \lambda_{n-2}p^{n-1} + \dots + \lambda_0 p + \lambda_I = 0 \quad (II.9)$$

possède des racines toutes à parties réelles négatives.

Afin de généraliser l'utilisation du schéma de commande que nous allons développer, introduisons les variables normalisées suivantes :

$$(\int e)_N = \frac{\int e(\tau) d\tau}{N_I}; e_N = \frac{e}{N_0}; e_N^{(k)} = \frac{e^{(k)}}{N_k}, \quad k = 1, \dots, n-1 \quad (II.10)$$

$N_I, N_k, k = 1, \dots, n-1$ sont des scalaires positifs non nuls représentant les facteurs de normalisation. L'expression d'une surface quelconque dans l'espace normalisé sera :

$$S_N = e_N^{(n-1)} + \lambda_{N_{n-2}} e_N^{(n-2)} + \dots + \lambda_{N_1} \dot{e}_N + \lambda_{N_0} e_N + \lambda_M (\int e)_N \quad (II.11)$$

Pour que la surface de glissement $S = 0$ soit invariante par rapport à la normalisation, on choisit :

$$\lambda_{N_1} = \frac{N_1}{N_{n-1}} \lambda_j; \lambda_{N_k} = \frac{N_k}{N_{n-1}} \lambda_k; k = 0, \dots, n-2 \quad (II.12)$$

Pour déterminer les différents facteurs de normalisation, on peut se baser sur l'imposition de valeurs maximales tolérables des différents variables normalisées.

Soit les valeurs maximales tolérables suivantes :

$$(\int e)_{max} = \max \left(\left| \int_0^t e(\tau) d\tau \right| \right); e_{max} = \max_i (|e(t)|); e_{max}^{(k)} = \max_i (|e^{(k)}(t)|), k = 1, \dots, n-1 \quad (II.13)$$

Ainsi, nous proposons de choisir :

$$N_1 = (\int e)_{max}; N_0 = e_{max}; N_k = e_{max}^{(k)} \quad k = 1, \dots, n-1 \quad (II.14)$$

dans ce cas, les variables normalisées atteignent les valeurs extrêmes ± 1 lorsque les variables non-normalisées prennent leurs valeurs maximales.

Quant aux valeurs maximales tolérables, nous proposons de les déterminer à partir de la sortie désirée et de ses $(n-1)$ dérivées selon les relations suivantes :

$$e_{max} = \max_i |y_d(t)| \times \frac{\bar{\epsilon}_0}{100}; e_{max}^{(k)} = \max_i |y_d^{(k)}(t)| \times \frac{\bar{\epsilon}_k}{100} \quad (II.15a)$$

avec, $0 < \bar{\epsilon}_0 \ll 1$: coefficient traduisant le pourcentage de l'erreur tolérée par rapport à la valeur maximale de la sortie;

$0 < \bar{\epsilon}_k \ll 1, k = 1, \dots, n-1$: coefficients traduisant le pourcentage de la $k^{ième}$ dérivée de l'erreur tolérée par rapport à la valeur de la $k^{ième}$ dérivée de la sortie désirée;

Les coefficients $\bar{\epsilon}_0, \bar{\epsilon}_k$ sont imposés par l'utilisateur en se basant sur la trajectoire désirée. Ils représentent le pourcentage admissible de cette trajectoire.

Quant à la normalisation de la variable intégrale de l'erreur, on peut s'intéresser à avoir une activation maximale de l'action d'intégration à l'instant $t = t_f$ avec t_f la durée de transition vers l'état final. Si on suppose qu'une erreur maximale $e(t) = e_{max} \quad 0 \leq t \leq t_f$ c'est produit le long de la transition, l'intégrale de l'erreur prend la valeur :

$$\int_{t=0}^{t_f} e(t) dt = e_{max} \times t_f \quad (II.15b)$$

si on choisit :

$$(\int e)_{max} = e_{max} \times t_f \quad (II.15c)$$

on obtient :

$$\left((\int e)_N \right)_{t_f} = \frac{\int_0^{t_f} e(t) d\tau}{N_1} = 1 \quad (II.15d)$$

ce qui correspond à une intervention maximale de l'intégration à l'instant $t = t_f$. Ce cas n'est toujours pas réalisable pour les deux raisons suivantes :

- 1- L'erreur ne peut pas demeurer à sa valeur maximale durant toute la trajectoire ($t = 0$ jusqu'à $t = t_f$).

2- L'erreur prend généralement des valeurs positives et négatives que l'une peut annuler

l'autre. Ce qui peut ramener la valeur de $\int_{t=0}^{t_f} e(t)dt < \left(\int e\right)_{\max}$.

d'où l'activation de l'action intégrale n'est pas au maximum.

Pour éviter de telle situation, on procède à une modification en prenant :

$$\left(\int e\right)_{\max} = e_{\max} \times t_f \times \bar{\epsilon}_I \tag{II.15e}$$

$0 < \bar{\epsilon}_I \ll 1$: coefficient choisit arbitrairement.

II.3.2 Nouvelles variables d'entrée :

Considérons un espace de dimension $(n+1)$ où l'état du système est représenté par le vecteur normalisé $Er_N = \left[\left(\int e\right)_N e_N \dot{e}_N \dots e_N^{(n-2)} e_N^{(n-1)} \right]^T$. Dans cet espace, la surface de glissement normalisée correspond à un hyperplan.

On définit par $A_N = [\lambda_{N1} \lambda_{N0} \dots \lambda_{Nn-2}]^T$ le vecteur normal à la surface de glissement. Le vecteur unitaire parallèle à ce vecteur est :

$$V_N = \frac{A_N}{\|A_N\|} \tag{II.16}$$

où, $\|\cdot\|$: désigne la norme Euclidienne.

Pour quantifier la distance entre le point de l'état et la surface de glissement normalisée, nous utilisons les variables de distance suivantes :

- d_n : La distance entre le point de l'état et la surface de glissement selon la direction du vecteur unitaire normal à la surface;
- d_p : La distance Euclidienne minimale, entre le point de l'état et le vecteur unitaire normal à la surface;

D'après la définition de d_n , cette distance peut être considérée comme la projection du vecteur d'erreur Er_N sur le vecteur V_N . Ainsi, on peut écrire $d_n = V_N^T \cdot Er_N$, d'où :

$$d_n = \frac{\lambda_{N1} \left(\int e\right)_N + \lambda_{N0} e_N + \dots + \lambda_{Nn-2} e_N^{(n-2)} + e_N^{(n-1)}}{\|A_N\|} \tag{II.17}$$

En utilisant la norme Euclidienne, la distance d_p est donnée par :

$$d_p = \sqrt{\|Er_N\|^2 - d_n^2} \tag{II.18}$$

Une représentation des variables de distance dans le cas d'un système d'ordre deux ($n=2$), est illustrée par la figure (II.3).

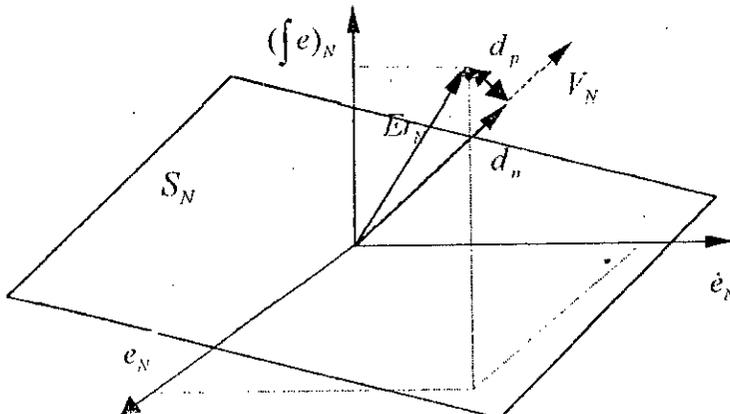


figure (II.3) : Représentation graphique des distances d_n et d_p pour $n=2$.

D'après les relations (II.11) et (II.17), on remarque que la variable d_n est une image de la surface normalisée car on peut écrire :

$$d_n = \frac{S_N}{\| \Lambda_N \|} \quad (II.19)$$

cette distance est positive lorsque le vecteur projection du vecteur Er_N sur le vecteur V_N est de même direction que ce dernier. Elle est négative si les deux vecteurs sont de directions opposées. Quant à la distance d_p , elle est positive.

II.3.3 Schéma de commande à base de régulateur flou :

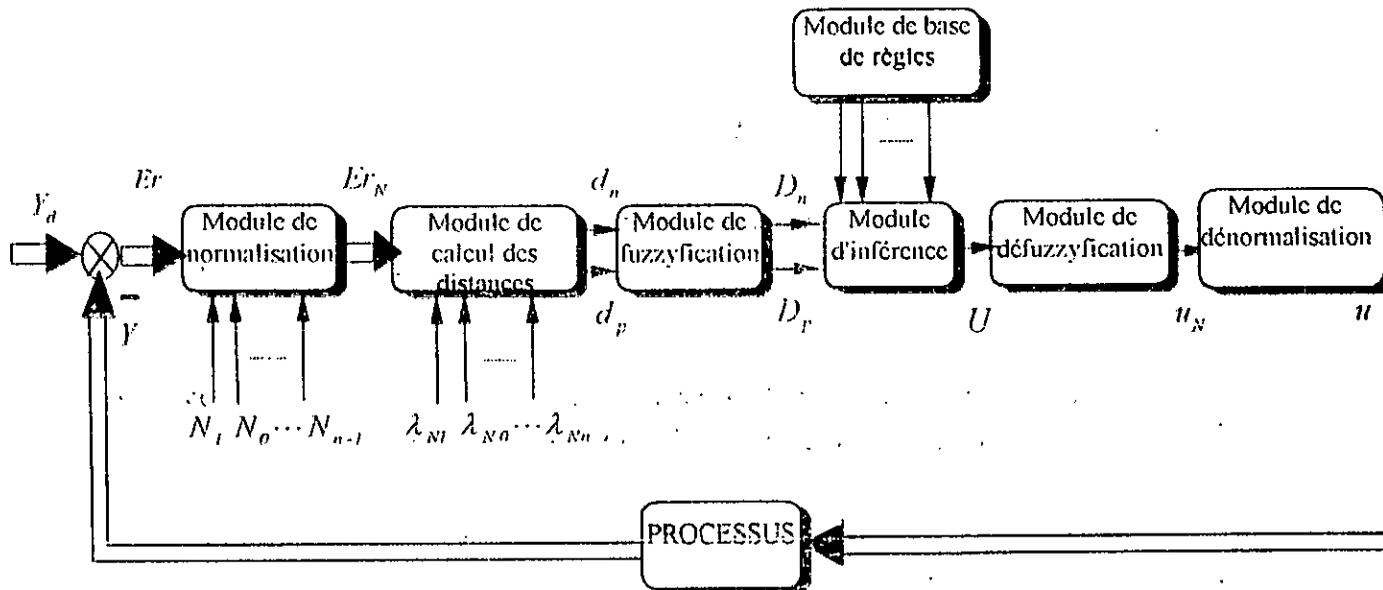
La plupart des régulateurs flous à base de modèle de Mamdani, utilisant des variables floues associées à l'erreur et sa dérivée (ou sa variation) comme variables d'entrée.

L'idée du régulateur flou que nous proposons consiste à utiliser les deux variables de distance d_n et d_p comme variables d'entrées d'un modèle de type Mamdani. Ainsi, on définit :

D_n : la variable floue associée à la variable de distance d_n ;

D_p : la variable floue associée à la variable de distance d_p .

Sur la base de ces deux variables floues introduites, le schéma de commande est celui montré par la figure (II.4).



figure(II.4) : Schéma de commande à base de régulateur flou

D'après ce schéma, le régulateur flou est composé des modules suivants :

- Module de normalisation;
- Module de calcul des distances;
- Module de fuzzification;
- Module de la base des règles;
- Module d'inférence;
- Module de défuzzification;
- Module de dénormalisation.

II.3.4 Description des modules du régulateur flou

II.3.4.1 Module de normalisation

En utilisant la relation (II.10), on effectue une normalisation du vecteur d'erreur

$$Er = \left[\int_0^t e(\tau) d\tau e^{-\lambda_1 t} \dots e^{-(n-2)t} e^{-(n-1)t} \right]^T$$

pour que toutes les composantes du vecteur normalisé

$Er_N = \left[\left(\int_0^t e(\tau) d\tau \right) e^{-\lambda_1 t} e^{-\lambda_2 t} \dots e^{-(n-2)t} e^{-(n-1)t} \right]^T$ prennent des valeurs dans l'intervalle $[-1, +1]$. Les paramètres de normalisation $N_1, N_k, k = 0, \dots, n-1$, sont déterminés à l'aide des relations (II.13) et (II.14).

II.3.4.2 Module de calcul des distances :

Via ce module, les composantes du vecteur normalisé Er_N , sont utilisées pour calculer les distances d_n et d_p . Pour cela il est nécessaire de suivre les étapes suivantes :

1. Choisir les paramètres $\lambda_1, \lambda_k, k = 0, \dots, n-2$ de telle sorte que l'équation caractéristique (II.9) possède des racines toutes à partie réelle négative;
2. Déterminer les paramètres $\lambda_{N1}, \lambda_{Nk}, k = 0, \dots, n-2$ à partir de la relation (II.12).
3. Calculer les distances d_n et d_p à partir des relations (II.17) et (II.18).

II.3.4.3 Module de fuzzyfication :

Pour que les informations sur l'état du système, sous forme des valeurs des distances d_n et d_p , puissent être manipulées, il est nécessaire de les transformer en valeurs floues. Pour cette raison, on effectue une fuzzyfication des deux variables d_n et d_p en utilisant des fonctions singleton ou non-singleton.

II.3.4.4 Module de la base des règles :

L'algorithme de commande d'un régulateur flou est stocké sous forme de règles linguistiques (ou règles floues). Chaque règle détermine une valeur floue pour chaque combinaison des valeurs floues des variables d'entrée.

Dans le cas du régulateur proposé, les variables floues d'entrées sont notées D_n et D_p , donc les règles de décision s'écrivent sous la forme suivante :

$$\text{SI } (D_n \text{ est } F_1^{(l)} \text{ et } D_p \text{ est } F_2^{(l)}) \text{ ALORS } (U \text{ est } B^{(l)}) \quad l = 1, \dots, k$$

où, k : le nombre des règles,

$F_1^{(l)}, F_2^{(l)}$ et $B^{(l)}$: représentent les valeurs floues attribuées, respectivement aux variables D_n, D_p et U dans la $l^{\text{ème}}$ règle.

Sur la base du principe des régulateurs à modes glissants avec couche bornée, présentés dans la section § (II.2), nous pouvons utiliser les directives de synthèse suivantes :

- ♦ La commande u doit être de même signe que celui de la variable de distance d_n ;
- ♦ Le module de la commande doit augmenter, si la distance d_n augmente;
- ♦ La commande augmente en module si la distance d_p augmente. Cela permettra d'atteindre d'une façon rapide la région centrale (région au voisinage de l'origine $d_n = d_p = 0$);
- ♦ Si l'état du système, en terme de d_n et d_p , est jugé relativement éloigné de la surface de glissement, on applique une commande constante maximale.

Les deux premières règles et la quatrième, sont déduites de la loi de commande d'un régulateur à modes glissants avec couche bornée, car la distance d_n et la surface S_N sont équivalentes.

L'utilisation de la variable d_p à travers la troisième directive, permet de prendre en considération l'éloignement du point de l'état par rapport à la surface de glissement dans une direction différente de celle de d_n . Cela attribue au régulateur flou proposé, la tâche de ramener le point de l'état directement vers l'origine et non pas vers la surface de glissement.

Considérons le cas particulier où on choisit :

- Huit classes pour chacune des deux variables D_n et U , elles sont notées :
 NG : Négative grande; NM : Négative moyenne. NP : Négative petite; NZ : Négative nulle;
 PG : Positive grande; PM : Positive moyenne. PP : Positive petite; PZ : Positive nulle.
- Quatre classes pour la variable D_p , notées :
 Z : Nulle; M : Moyenne ; P : Petite; G : Grande.

Une distribution gaussienne de ces différentes classes est illustrée par la figure (II.5).

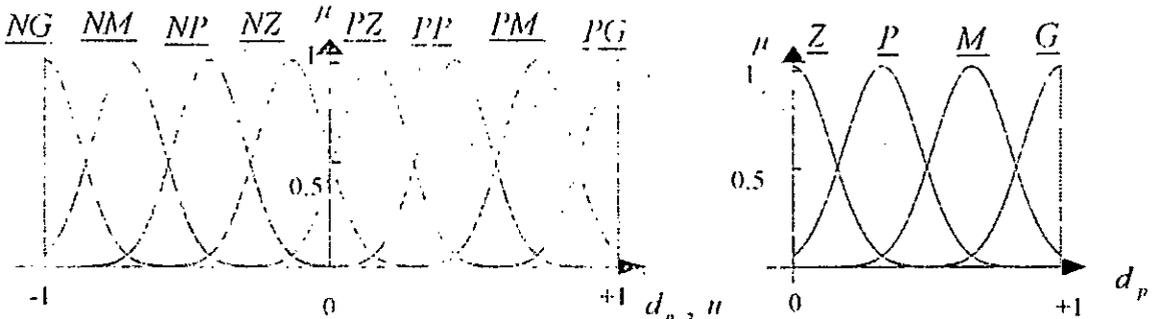


figure (II.5) : Distribution gaussienne des différentes classes des variables d'entrée et de sortie.

En suivant les directives mentionnées ci-dessus, on détermine l'ensemble des règles du régulateur flou. Elles sont regroupées dans la table (II.1).

D_n	NG	NM	NP	NZ	PZ	PP	PM	PG
D_p								
Z	NG	NM	NP	NZ	PZ	PP	PM	PG
P	NG	NG	NM	NP	PP	PM	PG	PG
M	NG	NG	NG	NM	PM	PG	PG	PG
G	NG	NG	NG	NG	PG	PG	PG	PG

Table (II.1) : table de régulateur flou de 32 règles de décision.

Dans cette table, on distingue deux régions principales :

1. Région de non-saturation (zone interne) : elle correspond à une zone de couche bornée qualitative (dans le plan linguistique);
2. Région de saturation (zone externe) : où la commande maintient la même valeur linguistique (NG ou PG), elle correspond à une zone de saturation qualitative.

II.3.4.5 Module d'inférence :

A travers ce module, les valeurs floues résultantes de la fuzzyfication sont inférées avec les règles de décision, pour produire une valeur floue de la commande. Pour cela, on utilise l'un des mécanismes d'inférence présentés dans le chapitre précédent, à savoir :

- Méthode d'inférence Max-min;
- Méthode d'inférence Max-prod;
- Méthode d'inférence Som-prod.

II.3.4.6 Module de défuzzyfication :

Ce module est nécessaire pour transformer la valeur floue produite par le module d'inférence en une valeur réelle de la commande.

Pour cela, on peut utiliser une des méthodes de défuzzyfication introduite dans le chapitre précédent, à savoir :

- Méthode du maximum ;
- Méthode de la moyenne des maxima ;
- Méthode du centre de gravité ;
- Méthode des hauteurs.

II.3.4.7 Module de dénormalisation :

En général, après défuzzyfication, la commande u_N doit subir une amplification ou atténuation pour que l'effort fournit par le régulateur soit adapté à l'application envisagée. C'est ainsi que la commande à l'entrée du processus est choisie :

$$u = u_N \times N_u \quad (II.20)$$

avec, N_u : facteur de dénormalisation.

La commande normalisée est généralement entre +1 et -1. Par conséquent, N_u doit être choisi supérieur à la commande maximale exigée par le processus.

II.3.5 Surface de commande :

Dans le cas général, un régulateur flou réalise une fonction non linéaire qui peut être caractérisée par une relation entrée-sortie [DR193]. Sa représentation dans l'espace entrée-sortie est appelée surface de commande. Cette relation dépend de méthodes de fuzzyfication, défuzzyfication et d'inférences adoptées.

Considérons le cas d'un régulateur où est utilisé :

- Un fuzzyficateur singleton;
- Un mécanisme d'inférence Som-prod
- Une défuzzyfication par la méthode du centre de gravité.

Dans ce cas la relation entre la sortie u et les variables d'entrées d_n, d_p est donnée par l'expression (Chapitre I, relation (I.23b)) :

$$u = N_u \times \frac{\sum_{l=1}^k \mu_{F_1^{(l)}}(d_n) \times \mu_{F_2^{(l)}}(d_p) \times S^{(l)} \times C^{(l)}}{\sum_{l=1}^k \mu_{F_1^{(l)}}(d_n) \times \mu_{F_2^{(l)}}(d_p)} \quad (II.21)$$

avec, $F_1^{(l)}$ et $F_2^{(l)}$: respectivement, les valeurs floues des variables D_n et D_p associées à la $l^{\text{ème}}$ règle
 $C^{(l)}$: centre de gravité de l'ensemble flou de la valeur de la commande associée à la $l^{\text{ème}}$ règle.
 $S^{(l)}$: surface au-dessous de l'ensemble flou de la valeur de la commande associée à la $l^{\text{ème}}$ règle.

En utilisant la table des règles (table (II.1)), avec la distribution gaussienne montrée par la figure (II.5), on obtient la surface de commande illustrée par la figure (II.6).

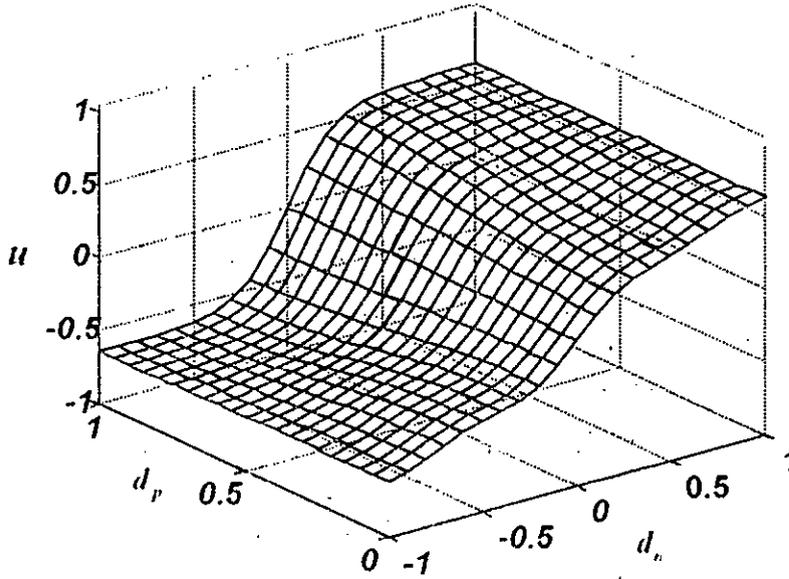


figure (II.6) : Exemple de surface de commande du régulateur flou

II.3.6 Discussion :

Le régulateur flou présenté est basé sur l'utilisation des variables de distance calculées dans un espace déterminé à partir de l'expression linéaire de la surface de glissement. Les deux variables de distance permettent de quantifier l'éloignement du point de l'état par rapport à la surface de glissement dans deux directions opposées.

En utilisant seulement deux variables floues, la structure particulière du régulateur proposé permet d'incorporer les mesures de l'erreur et de ses dérivées d'ordre supérieurs dans le calcul de la commande. Indépendamment de l'ordre maximal des dérivées de l'erreur utilisées dans l'expression de la surface, l'algorithme flou est basé sur les règles de décision regroupées dans la table (II.1).

D'autre part, le régulateur étudié peut être considéré comme une version floue modifiée des régulateurs classiques de type PID⁽ⁿ⁻¹⁾ (n > 1). Ces derniers englobent les actions proportionnel (P), intégral (I), et dérivée (D) d'ordre (n-1). La commande est d'expression :

$$u_{PID^{(n-1)}} = k_{n-1}e^{(n-1)} + k_{n-2}e^{(n-2)} + \dots + k_1\dot{e} + k_0e + k_I \int_0^t e(\tau)d\tau. \quad n > 1 \quad (II.22)$$

avec, $k_j, k_j, j = 0, \dots, n-1$ gains du régulateur PID⁽ⁿ⁻¹⁾.

Considérons le cas d'une surface S_N ayant la même expression que la commande $u_{PID^{(n-1)}}$. D'après la relation (II.19), la variable de distance d_n sera l'image de la commande $u_{PID^{(n-1)}}$. Ainsi si on choisit des règles floues est un modèle de Mamdani de telle manière qu'on assure une linéarité entre la commande u et la variable d_n , le régulateur flou devient similaire à un régulateur de type PID⁽ⁿ⁻¹⁾.

Cependant, l'utilisation des règles données par la table (II.1), permet d'introduire :

- Une activation maximale de la commande dans les zones éloignées de l'origine de l'espace de l'état (saturation de la commande).
- Une non-linéarité dans la loi de la commande via l'ajustement des fonctions d'appartenance des variables floues d'entrée et de sortie et / ou les opérateurs adoptés dans le modèle de Mamdani.

Le passage entre différents régulateurs flous de type PI, PD, PID peut être effectué par simple modification de l'expression de la surface. Ainsi, on obtient :

- Un régulateur flou de type PI, si on choisit, $S = e + \lambda_I \int_0^t e(\tau) d\tau$;
- Un régulateur flou de type PD, si on choisit, $S = \dot{e} + \lambda_D e$;
- Un régulateur de type PID, si on choisit, $S = \dot{e} + \lambda_D e + \lambda_I \int_0^t e(\tau) d\tau$.

Les régulateurs flous de type PI (PI-flou) présenté dans [DRI93], consiste à prendre l'erreur et sa dérivée comme deux variables d'entrées d'un modèle de type Mamdani. La sortie de ce dernier est considérée comme un incrément de la commande. Dans une deuxième structure on peut considérer deux variables d'entrée : l'erreur et son intégration. La commande est la sortie d'un modèle de Mamdani. Lorsqu'on introduit l'intégral de l'erreur comme troisième variable d'entrée l'opération de dressage de la table de décision devient une tâche difficile.

De même le régulateur PD-flou proposé dans [DRI93], consiste à prendre la sortie d'un régulateur PI-flou comme la commande à appliquée au processus. L'algorithme flou utilise la table de décision très célèbre de Mac Vichar. Ce régulateur, contrairement au régulateur de type PD proposé, ne permet pas l'ajustement directe de la surface de glissement (la courbe qui détermine la zone de changement de signe de la commande).

Dans [DRI93], on propose une structure d'un régulateur flou de type PID (PID-flou). Dans ce dernier, on utilise trois variables floues associées à l'erreur, sa dérivée et son intégration, avec un modèle de Mamdani. Ainsi les règles de décision détermine une valeur floue pour chaque combinaisons des valeurs floues des variables d'entrée. L'utilisation de trois variables floues d'entrée complique l'opération de dressage de la table de décision. La nouvelle structure proposée dans le présent travail permet de raisonner seulement sur deux variables floues d'entrées. Ainsi, on utilise encore la même table d'entrée (la table (II.1)).

Le régulateur flou présenté, utilise une surface de glissement linéaire. Cependant autres surfaces non-linéaire peuvent être adoptées. Dans ce cas, une modification appropriée de l'espace de phase sera nécessaire.

Pour montrer cela, considérons le cas particulier où on choisit une surface non-linéaire définie par :

$$S = \dot{e} + \lambda_D e + \lambda_I \int_0^t e(\tau) d\tau + \lambda_h \tanh(\alpha e) \quad (\text{II.23})$$

avec, \tanh : fonction tangente hyperbolique définie par : $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, $\forall x \in R$.

Dans [MAD97], on montre que la dynamique de l'erreur définie par la surface de glissement $S = 0$, est stable si on choisit : $\lambda_D > 0$; $\alpha, \lambda_I, \lambda_h \geq 0$. Après normalisation, l'expression de la surface devient :

$$S_N = \dot{e}_N + \lambda_{ND} e_N + \lambda_{NI} \left(\int e \right)_N + \lambda_{Nh} (\tanh(\alpha e))_N \quad (\text{II.24})$$

avec, $\dot{e}_N, e_N, (\int e)_N$ et $(\tanh(\alpha e))_N$ variables normalisées associées, respectivement, aux variables $\dot{e}, e, \int_0^t e(\tau)d\tau$ et $\tanh(\alpha e)$. Dans ce cas, on choisit un espace où le vecteur de l'erreur normalisé (Er_N) est donné par :

$$\tilde{Er}_N = \begin{bmatrix} \dot{e}_N & e_N & (\int e)_N & (\tanh(\alpha e))_N \end{bmatrix} \quad (II.25)$$

Le vecteur unitaire normal à la surface de glissement $S_N = 0$, dans le nouvel espace devient :

$$V_N = \frac{\tilde{\Lambda}_N}{\|\tilde{\Lambda}_N\|} \quad (II.26)$$

avec,

$$\tilde{\Lambda}_N = [\lambda_{Nh} \quad \lambda_{Ni} \quad \lambda_{No} \quad 1]^T$$

donc, les variables de distance d_n et d_p prennent les expressions suivantes :

$$d_n = \frac{\lambda_{Nh}(\tanh(\alpha e))_N + \lambda_{Ni}(\int e)_N + \lambda_{No}e + \dot{e}}{\|\tilde{\Lambda}_N\|} \quad (II.27)$$

$$d_p = \sqrt{\|\tilde{Er}_N\|^2 - d_n^2} \quad (II.28)$$

Donc, lors de l'utilisation de surfaces non linéaires, seule le module de calcul des distances doit être modifié, tandis que les règles floues sont toujours données par la table (II.1).

II.4 Application à la commande d'un bras de robot manipulateur

Un robot manipulateur est composé de plusieurs liaisons articulées de dynamiques interconnectées. Pour appliquer le régulateur proposé à la commande décentralisé en poursuite de trajectoires, on associe à chaque articulation un régulateur flou qui agit à partir des mesures locales de position et de vitesse. Les interconnexions seront considérées comme des perturbations.

II.4.1 Choix du type des régulateurs :

Notre choix s'est porté sur l'adoption de deux types de régulateurs, selon que la gravitation intervient ou non sur la dynamique de l'articulation considérée.

On utilise un régulateur flou de type PID lorsque l'articulation est sujette à des effets gravitationnels. Sinon, on utilise un régulateur flou de type PD.

Soit q_i^d et \dot{q}_i^d , respectivement, les trajectoires désirées en position et en vitesse de la $i^{ème}$ articulation ($i=1, \dots, n$), et q_i et \dot{q}_i , respectivement, les variables de position et de vitesse de la $i^{ème}$ articulation ($i=1, \dots, n$). Donc, les erreurs en positions et en vitesses seront données par :

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

$$\dot{e}_i(t) = \dot{q}_i^d(t) - \dot{q}_i(t)$$

Dans ce cas l'expression de la surface utilisée dans le régulateur de la $i^{ème}$ articulation, est donnée par :

$$S_i = \begin{cases} \dot{e}_i(t) + \lambda'_0 e_i(t) & \text{si le régulateur est de type PD} \\ \dot{e}_i(t) + \lambda'_0 e_i(t) + \lambda'_j \int_0^t e(\tau)d\tau & \text{si le régulateur est de type PID} \end{cases}$$

Le choix de ces deux types de régulateurs est motivé par les raisons suivantes :

1. une étude a montré que lors du problème de poursuite de trajectoires, un régulateur classique de type PD est capable de stabiliser la dynamique de l'erreur de poursuite des différentes articulations, si [LEW93]:

- Les valeurs initiales de l'erreur en position et en vitesse sont suffisamment petites ;
- Les gains proportionnels et dérivées sont choisis suffisamment grandes.

2. plusieurs algorithmes de commande développés pour les robots font intervenir les actions PD ou PID (voir [SER89], [LEW93] et [MAD97]).

3. l'introduction de l'action intégrale à pour objectif de compenser les effets de la gravitation, au moins en régime permanent (lorsque les trajectoires désirées deviennent constantes). Pour montrer cela, considérons le cas où l'équilibre d'une liaison nécessite l'application d'une commande constante $u = u_0$. Pour générer cette valeur de la commande, les variables de distance prennent les valeurs d_{n0}, d_{p0} .

Mais d'après l'expression de la distance d_n (relation (II.17)), l'erreur sera régit par l'équation dynamique suivante :

$$\dot{e}_{Ni} + \lambda'_{N0} e_{Ni} + \lambda'_{Ni} \left(\int e \right)_{Ni} = d_{n0} \cdot \left\| \left[\lambda'_{Ni} \lambda'_{N0} I \right] \right\| \quad (\text{II.29})$$

ou,

$$\dot{e}_i + \lambda'_0 e_i + \lambda'_i \int_0^t e(\tau) d\tau = d_{n0} \times \left\| \left[\lambda'_{Ni} \lambda'_{N0} I \right] \right\| \times N_i^i$$

avec, $i = 1, \dots, n$ et n désigne le nombre d'articulations.

Grâce à la présence de l'action intégrale, l'erreur converge asymptotiquement vers zéro même si $d_{n0} \neq 0$.

En absence de l'action intégrale, il apparaîtra une erreur permanente de l'ordre

$$e_{i0} = \frac{d_{n0} \cdot \left\| \left[\lambda'_{Ni} \lambda'_{N0} I \right] \right\| \cdot N_i^i}{\lambda'_0} \quad (\text{II.30})$$

II.4.2 Dimensionnement des régulateurs flous de type PD et PID :

Nous avons proposé d'associer à chaque articulation un des deux régulateurs flous de type PD ou PID. Leurs dimensionnement consiste à déterminer les paramètres des différents modules.

• Facteur de normalisation :

Considérons le cas de problème de poursuite de trajectoire de référence de forme cycloïde. Ces trajectoires sont définies par [SER89]:

$$q_i^d = \begin{cases} q_i^{dl} + (q_i^{df} - q_i^{dl}) \cdot \left(\frac{t}{t_f} - \sin(2\pi t/t_f) \right) & \text{si } t \leq t_f \\ q_i^{df} & \text{si } t > t_f \end{cases} \quad i=1 \dots n$$

q_i^{dl} : valeur désirée initiale de la position;

q_i^{df} : valeur désirée finale de la position;

t_f : durée de transition entre les positions q_i^{dl} et q_i^{df} .

En appliquant les relations (II.14) et (II.15), on obtient :

$$N_0^i = \max |q_i^{df} - q_i^{dl}| \times \frac{\bar{\varepsilon}_0^i}{100} \quad i = 1, \dots, n$$

$$N_f^i = \max |q_i^{df} - q_i^{dl}| \times \frac{2\pi}{t_f} \times \frac{\bar{\varepsilon}_f^i}{100} \quad i = 1, \dots, n$$

$$N_i^i = N_0^i \times t_f \times \bar{\varepsilon}_f^i \quad i = 1, \dots, n \quad (\text{pour le contrôleur PID})$$

où, $0 < \bar{\varepsilon}_0^i, \bar{\varepsilon}_j^i \ll 100$, $0 < \bar{\varepsilon}_j^i \ll 1$ coefficients arbitraires.

◆ Paramètres de la surface de glissement :

Pour déterminer les paramètres de la surface de glissement, nous proposons de baser sur l'imposition du temps d'annulation de l'erreur en régime permanent ($t \geq t_f$).

• Cas de régulateur flou de type PD :

L'équation caractéristique associée à la surface de glissement S_i , est donnée par :

$$p + \lambda_0^i = 0 \quad i = 1, \dots, n$$

Pour imposer un temps de réponse à 5% de l'ordre de $Tr^i(5\%)$, on choisit : [DAZ81]

$$\lambda_0^i = \frac{3}{Tr^i(5\%)} \quad i = 1, \dots, n$$

• Cas de régulateur flou de type PID :

L'équation caractéristique associée à la surface de glissement S_i , est donnée par :

$$p^2 + \lambda_0^i p + \lambda_1^i = 0 \quad i = 1, \dots, n$$

Pour avoir un pôle double $-\lambda^i$, on choisit :

$$\lambda_0^i = 2\lambda^i \quad \text{et} \quad \lambda_1^i = (\lambda^i)^2$$

Pour imposer un temps de réponse de l'ordre de $Tr^i(5\%)$, on choisit [DAZ81] :

$$\lambda^i = \frac{3}{Tr^i(5\%)} \quad i = 1, \dots, n$$

d'où,
$$\lambda_0^i = \frac{6}{Tr^i(5\%)} \quad \lambda_1^i = \frac{9}{(Tr^i(5\%))^2} \quad i = 1, \dots, n$$

◆ Les classes floues :

Le nombre et la distribution des valeurs floues des différentes variables d'entrée et de sortie sont arbitraires. Les valeurs des variables d_n , d_p et u appartiennent, respectivement aux intervalles $[-1 \ +1]$, $[-1 \ +1]$ et $[0 \ 1]$. Dans notre travail, les différentes fonctions d'appartenance sont choisies de formes gaussiennes équidistantes. L'intersection entre deux fonction adjacentes est de degrés d'appartenances égal à 0.5.

Soit nf_{D_n}, nf_{D_p} et nf_U , les nombres des classes floues associées respectivement aux variables D_n, D_p et U , et :

• $m_{D_n}^j$ $j = 1, \dots, nf_{D_n}$, $m_{D_p}^j$ $j = 1, \dots, nf_{D_p}$, et m_U^j $j = 1, \dots, nf_U$, centres des $j^{ème}$ fonctions gaussiennes, associées respectivement aux variables D_n, D_p et U .

• σ_{D_n} , σ_{D_p} , σ_U : les écart-types des fonctions gaussiennes, associées respectivement aux variables D_n, D_p et U .

On montre que :

$$\begin{aligned} \sigma_{D_n} &= \frac{2}{(nf_{D_n} - 1)\sqrt{2 \ln 2}} & m_{D_n}^j &= \frac{2(j-1)}{(nf_{D_n} - 1)} - j \quad j = 1, \dots, nf_{D_n} \\ \sigma_{D_p} &= \frac{1}{(nf_{D_p} - 1)\sqrt{2 \ln 2}} & m_{D_p}^j &= \frac{(j-1)}{(nf_{D_p} - 1)} \quad j = 1, \dots, nf_{D_p} \\ \sigma_U &= \frac{2}{(nf_U - 1)\sqrt{2 \ln 2}} & m_U^j &= \frac{2(j-1)}{(nf_U - 1)} - 1 \quad j = 1, \dots, nf_U \end{aligned}$$

Quant à notre application, nous avons choisi :

- Quatre classes pour chacune des deux variables D_n et U ($n_{D_n} = n_U = 4$), elles sont notées $\{NG, NP, PP, PG\}$.
- Deux classes pour la variable D_p ($n_{D_p} = 2$), notées $\{P, G\}$.

La figure (II.7) illustre la représentation graphique de ces différentes classes.

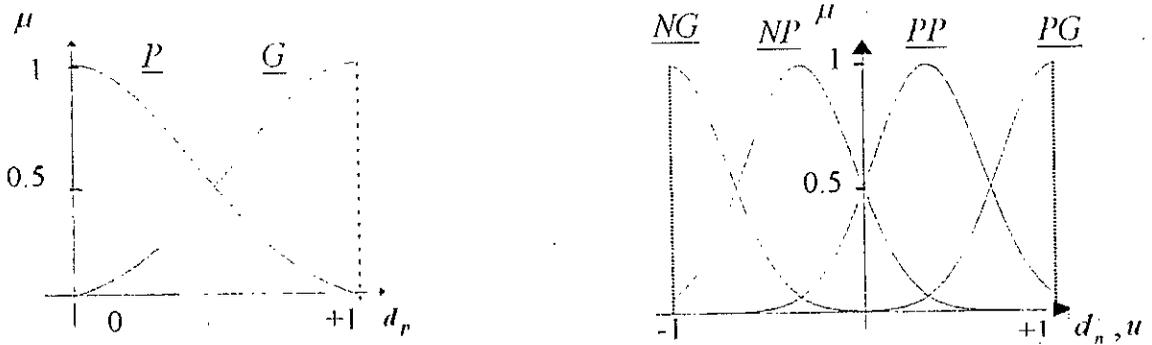


figure (II.7) : Distribution gaussienne équidistante des classes des variables d'entrée et de sortie, dans le cas $nf_{D_n} = nf_U = 4$ et $nf_{D_p} = 2$.

◆ **Table des règles floues :**

Considérons le cas précédent $n_{D_n} = n_U = 4$ et $n_{D_p} = 2$. En utilisant les directives de synthèse de règles décrites dans la section (II.2.3.4.4), on peut établir les règles de décision telles que regroupées dans la table (II.2).

D_n	NG	NP	PP	PG
D_p				
P	NG	NP	PP	PG
G	NG	NP	PP	PG

table (II.2) : Table des règles de décision dans le cas $nf_{D_n} = nf_U = 4$ et $nf_{D_p} = 2$.

◆ **Méthodes de fuzzyfication et de défuzzyfication et mécanisme d'inférence :**

Notre choix c'est porté sur :

- une fuzzyfication par des fonctions singleton.
- un mécanisme d'inférence de type Som-prod.
- une défuzzyfication par la méthode du centre de gravité.

◆ **Facteur de dénormalisation :**

Le facteur de dénormalisation est choisit supérieur à la valeur maximale du couple ou de la force exigée par l'articulation considérée.

II.4.3 Résultats de simulation :

Pour valider l'application des régulateur flous proposés dans la commande décentralisée des robots manipulateurs, nous avons procédé à la simulation. La résolution du système des équations différentielles a été réalisé par la méthode de Runge Kutta d'ordre 4 avec un pas constant $dt=0.001$ sec. Pour évoluer les intégrations dans la loi de commande, nous avons utilisé la méthode des trapèzes.

II.4.3.1 Application sur le robot PUMA560 :

Le robot PUMA560 présenté dans l'annexe A, possède six liaisons rotoïdes. Dans notre étude, nous supposons que les trois dernières articulations sont à l'arrêt. La première articulation agit dans un plan horizontal tandis que les deux autres fonctionnent dans des plans verticaux. La gravitation intervient seulement sur les deux dernières articulations.

Dans le but d'appliquer les régulateurs proposés, on associe à la première articulation un régulateur flou de type PD. Quant aux deux autres articulations, on utilise deux régulateurs de type PID. Les paramètres des trois régulateurs sont donnés par la table (II.3). Les différentes commandes sont initialisées par zéro.

Paramètres du régulateur	Symbole	Valeurs numériques		
		Première articulation $i=1$	Deuxième articulation $i=2$	Troisième articulation $i=3$
Normalisation	$\bar{\epsilon}_1^i ; \bar{\epsilon}_0^i \% ; \bar{\epsilon}_1^i \%$	- ; 0.28 ; 4.16	0.16 ; 0.28 ; 4.16	0.0350 ; 0.28 ; 4.16
	$N_1^i (rd.sec) ; N_0^i (rd) ; N_1^i (rd/sec)$	0 ; 88 10^{-4} ; 19.74 10^{-2}	2.4 10^{-4} ; 88 10^{-4} ; 17.45 10^{-2}	2.4 10^{-4} ; 88 10^{-4} ; 17.45 10^{-2}
Paramètres de la surface	$Tr^i (5\%) (sec)$	0.1500	0.1500	0.1500
	$\lambda_1^i ; \lambda_0^i$	0 ; 40	400 ; 40	400 ; 40
Facteurs de dénormalisation	$N_n^i (N.m)$	250	300	250

table (II.4) : Valeurs numériques des paramètres des trois régulateurs flous associés aux articulations du robot PUMA560.

II.4.3.2 Tests et performances :

Nous avons effectué les tests suivants :

Test de validation : Dans le cas de la trajectoire de Leahvy sans charge (défini dans annexe A), nous avons obtenu les résultats présentés dans la figure (II.8). Les trois articulations présentent une bonne poursuite avec des erreurs très faibles. Les commandes sont souples, et sans oscillations de même ordre de grandeur que les couples idéaux exigées (voir chapitre I). Les variables de distances sont caractérisées par des variations lisses. En régime permanent, elles se stabilisent à des valeurs qui permettent de compenser l'effet statique de la gravitation. Puisque cet effet n'intervient pas sur la première articulation, les distances d_{ni} et d_{pi} deviennent nulles en régime permanent.

Test de variation de charge : La figure (II.9) illustre les résultats obtenus dans le cas de l'ajout de charge de 2 Kg puis 4 Kg à l'instant initiale. On constate que le robot poursuit remarquablement la poursuite de la trajectoire avec des faibles erreurs. Pour compenser l'effet de cette augmentation de charge, les régulateurs flous, en particulier le troisième, génèrent des couples plus importants. Cela se manifeste aussi par l'augmentation des distances d_{ni} et d_{pi} . A l'instant $t=0.75 sec$ l'élément terminal relâche la charge transportée. On remarque une variation instantanée des couples pour atteindre les valeurs des commandes correspondantes à ces charges. La variation des distances d_{pi} est plus lisse par rapport à d_{ni} .

Test de variation Paramétrique : Dans ce test, nous avons effectué à l'instant $t=0.75 sec$ des variations brusques sur les paramètres physiques du modèle dynamique (les longueurs, les masses et les moments

d'inerties), en les augmentant de 50 % puis de 100 % de leurs valeurs initiales. Les résultats obtenus sont montrés à la figure (II.10). Les articulations effectuent la poursuite des trajectoires avec des faibles erreurs. La variation des couples est lisse, sauf à l'instant de changement des paramètres. Ce qui confirme la robustesse du régulateur.

Test de rupture de la commande : Pour montrer le caractère de décentralisation de la structure de commande, nous avons appliqué une rupture du deuxième régulateur à l'instant $t=0.75 \text{ sec}$ (la commande u_2 devient nulle). La figure (II.11) illustre les résultats obtenus. La première et troisième articulation continuent remarquablement la poursuite des trajectoires de référence avec des faibles erreurs. La première articulation, très affectée par cette rupture, génère un couple important pour compenser la grande inertie de la deuxième articulation.

Test de variation du pas d'échantillonnage : Pour avoir un aperçu sur le comportement des différents régulateurs, en cas d'un pas d'échantillonnage (T) plus important, nous l'avons augmenté à $T=10 \text{ ms}$ puis $T=15 \text{ ms}$. Pour obtenir un meilleur comportement, il était nécessaire de diminuer le gain de normalisation de la dérivée de l'erreur comme suit :

$$N_1^1 = 43.62 \cdot 10^{-2} \text{ rd/sec}; N_1^2 = 43.62 \cdot 10^{-2} \text{ rd/sec}; N_1^3 = 104.70 \cdot 10^{-2} \text{ rd/sec} \text{ pour } T_0=10 \text{ msec};$$

$$N_1^1 = 69.80 \cdot 10^{-2} \text{ rd/sec}; N_1^2 = 43.62 \cdot 10^{-2} \text{ rd/sec}; N_1^3 = 139.60 \cdot 10^{-2} \text{ rd/sec} \text{ pour } T_0=15 \text{ msec}.$$

Les résultats obtenus sont consignés sur la figure (II.12). Nous constatons d'assez bonnes performances si ce n'est une augmentation de l'erreur et l'apparition d'oscillations lors de démarrage. Les erreurs restent relativement faibles et s'annulent en régime permanent. Quant aux commandes, puisqu'elles sont initialisées à zéro, les régulateurs nécessitent un temps pour atteindre les valeurs des commandes à appliquer aux différentes articulations. Cela se manifeste par un régime transitoire. Sa durée est d'autant plus importante, lorsque le débit d'information diminue (pas d'échantillonnage augmente).

II.5 Conclusion

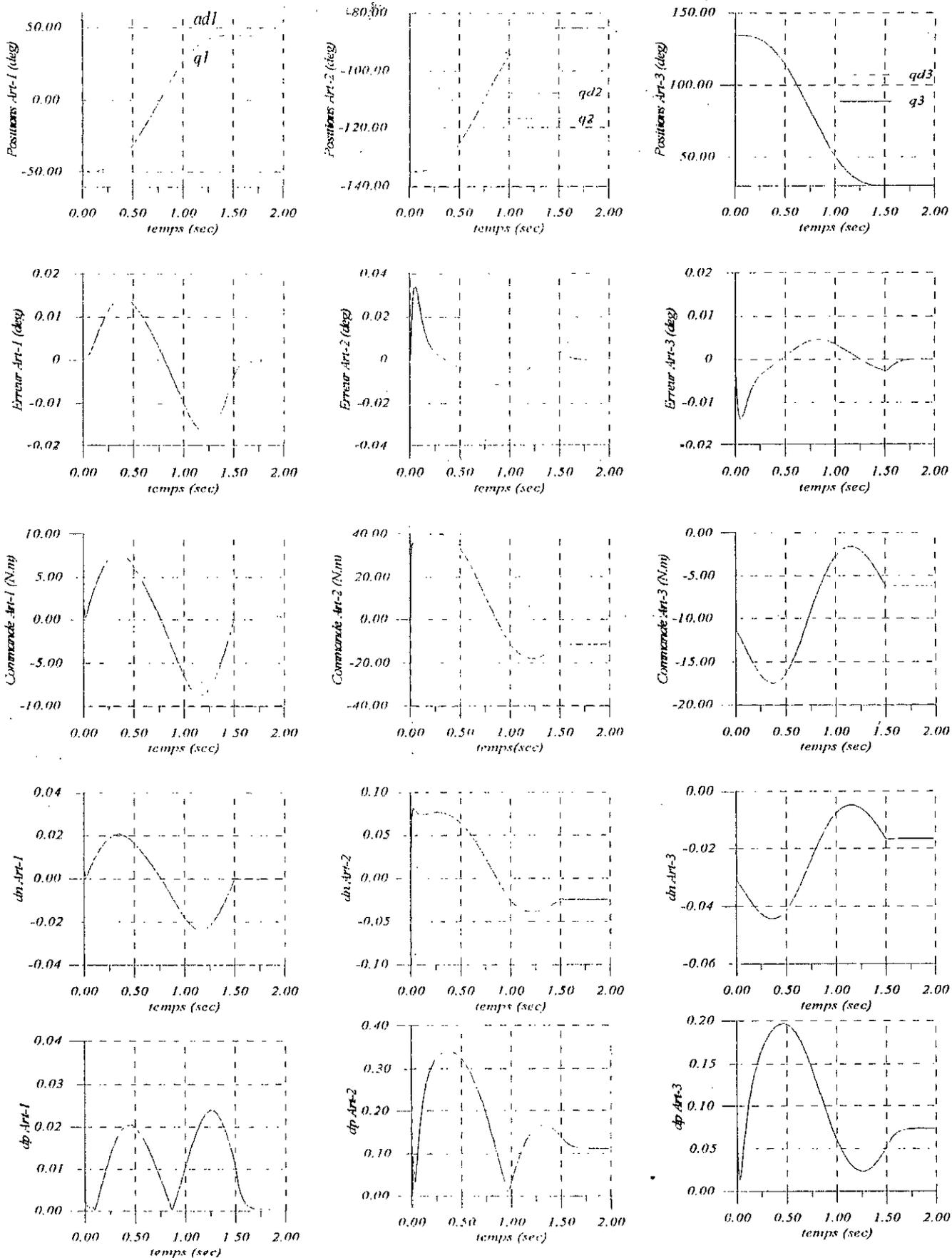
Dans ce chapitre nous avons présenté et appliqué un régulateur flou linguistique à la commande décentralisée d'un bras de robot manipulateur. Le régulateur est basé sur le modèle de Mamdani, dont les règles sont inspirées du principe des régulateurs à modes glissants.

En utilisant deux variables de distance, le régulateur étudié permet de :

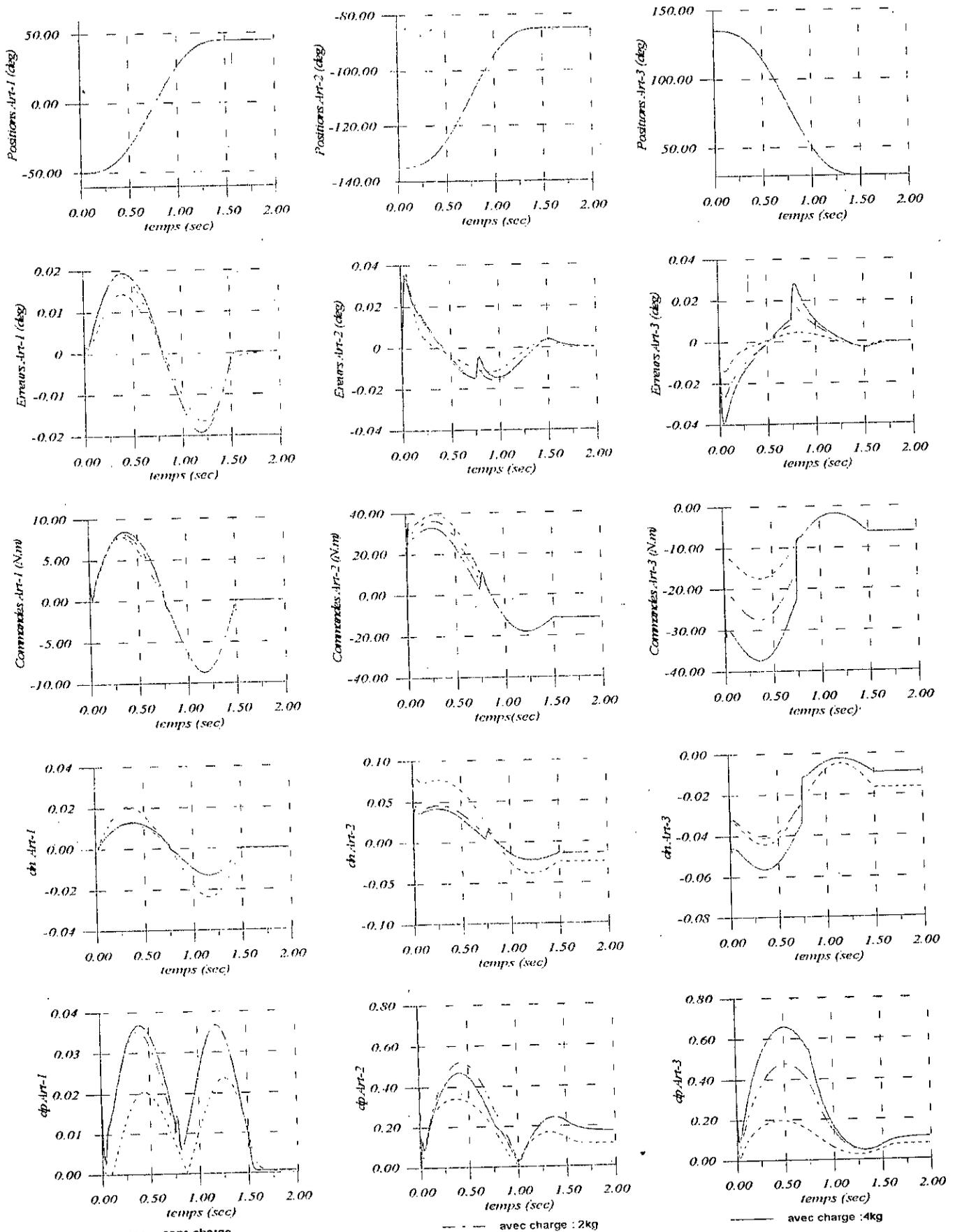
- Incorporer les dérivées d'ordre supérieures de l'erreur, dans le calcul de la commande, sans que la table de décision soit modifiée ou augmentée;
- Imposer une dynamique linéaire ou non linéaire de l'erreur à travers la sélection de la surface de glissement;
- Générer une famille de régulateurs flous de types PID⁽ⁿ⁾.

Dans le but de systématiser le dimensionnement du régulateur, nous avons proposé des directives pour la détermination des différents paramètres.

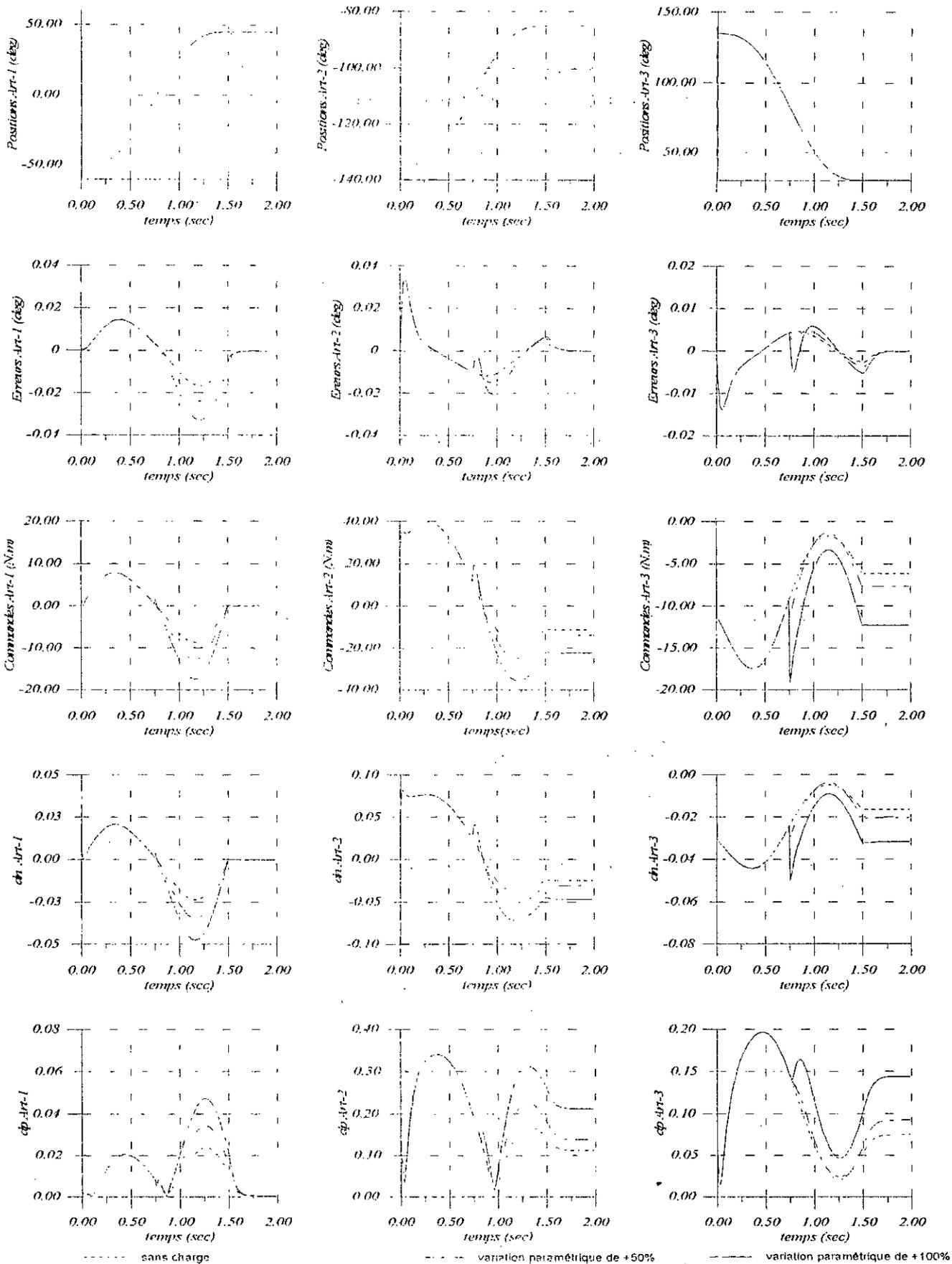
L'application du régulateur développé en commande décentralisée du robot manipulateur PUMA560, a donné de bons résultats (erreurs très faibles et commandes réalisables sans oscillations). Sous un dimensionnement correcte (ou optimale), les régulateurs associés aux différentes articulations ont montré une robustesse vis-à-vis de l'augmentation de charge, de paramètres et de rupture de commande. Cependant, les performances du régulateur restent sensibles au pas d'échantillonnage.



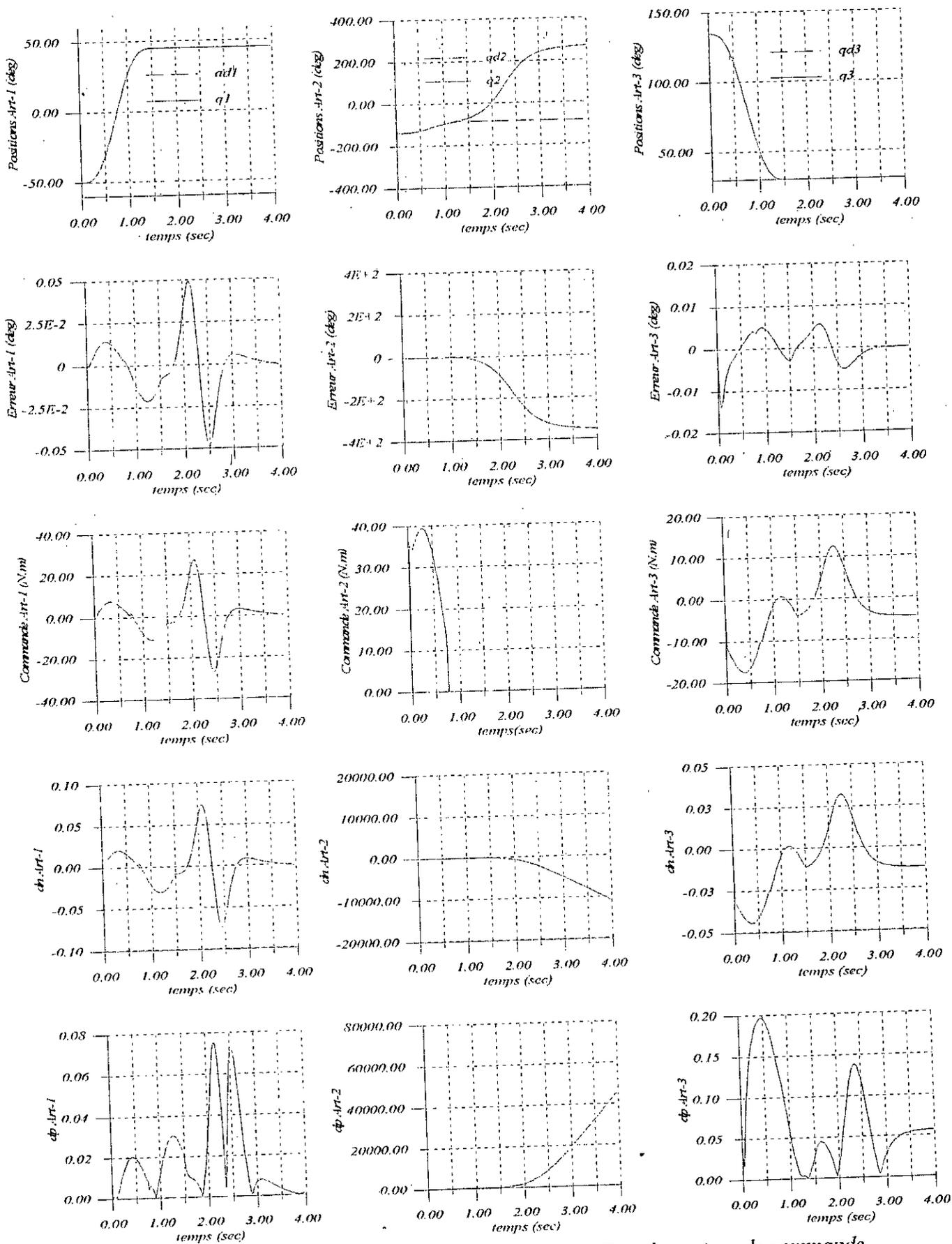
figure(II.8) : Réponses des différentes articulations : Test sans charge.



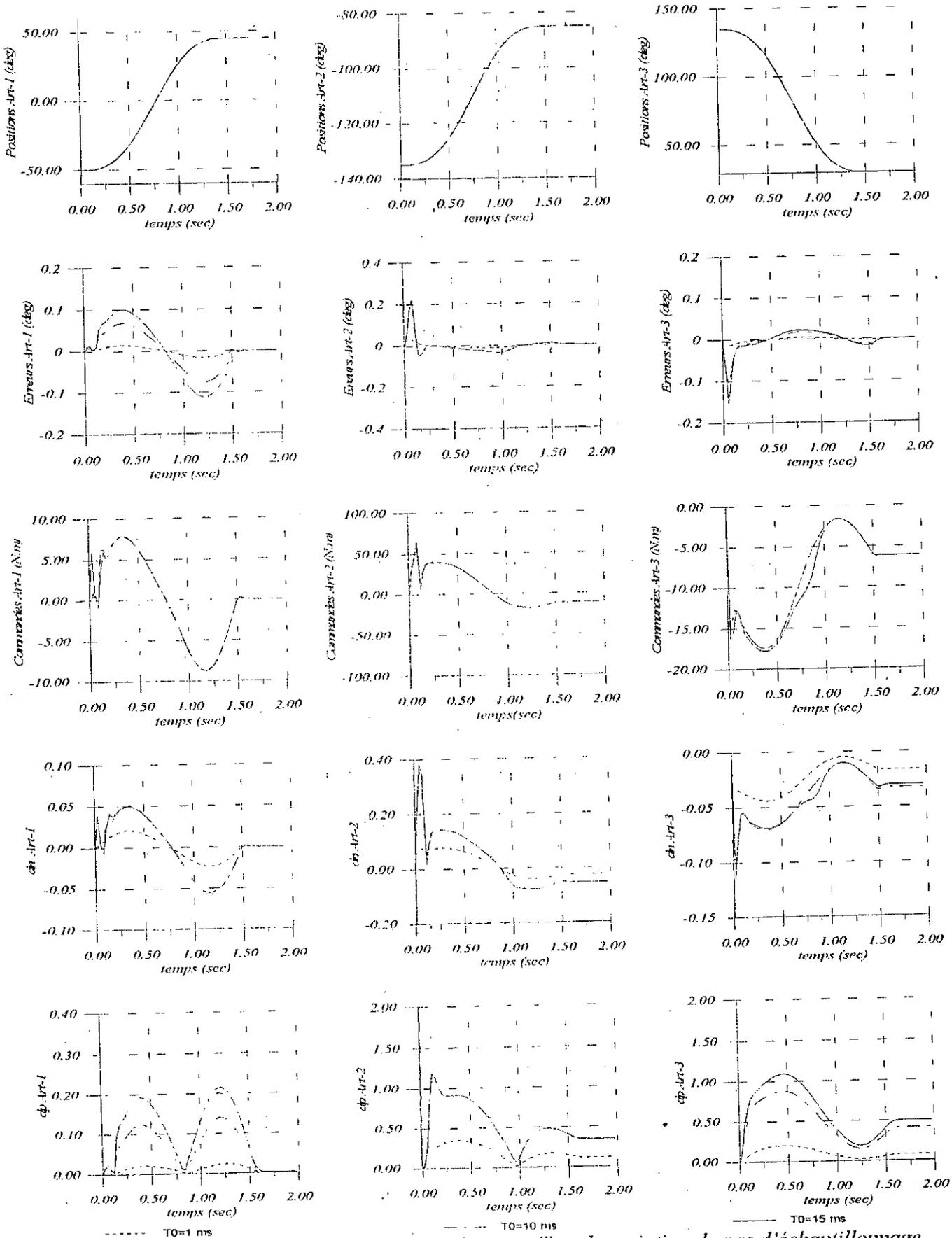
figure(II.9) : Réponses des différentes articulations : Test de variation de charge.



figure(II.10) : Réponses des différentes articulations : Test de variation paramétrique.



figure(II.11) : Réponses des différentes articulations : Test de rupture de commande.



figure(II.12) : Réponses des différentes articulations : Test de variation du pas d'échantillonnage.

Chapitre III

Commande décentralisée floue adaptative continue

Chapitre III :

Commande décentralisée floue adaptative continue

III.1 Introduction.

L'application des régulateurs à base de connaissances linguistiques, a constitué un moyen puissant dans la commande des processus complexes [PIC97]. La synthèse de la majorité de ces régulateurs est basée principalement sur l'expérience de l'opérateur ou l'ingénieur du processus dont dépendent fortement les performances du régulateur [DRI93].

La grande partie des travaux portés sur les régulateurs à base de connaissances linguistiques a été consacrée au développement de régulateurs particuliers destinés à des applications spécifiques [TZA90][ZHA93][KIM94]. Ces travaux n'offrent pas une méthodologie de synthèse permettant une analyse générale des performances des schémas de commande, en particulier leur stabilité.

En général, le maintien de la dépendance des systèmes flous avec des descriptions linguistiques limite la classe des outils mathématiques qu'on peut utiliser [PIC97]. Cependant grâce à des études effectuées sur les systèmes flous, on a montré que certaines classes ont la qualité d'être des approximations universelles de fonctions [MEN95]. Cette importante propriété a donné une nouvelle voie à l'utilisation des systèmes flous en commande.

A cet effet, plusieurs travaux sont orientés vers la combinaison des systèmes flous avec des techniques de commande telles que la commande adaptative continue et discrète [SPO95a][SPO95b][PIC97], et la commande linéarisante [TRE97]. Dans la plupart de ces schémas, le système flou intervient pour la compensation (ou l'approximation) de fonctions non linéaires. Le rôle de l'opérateur dans la détermination du système flou est remplacé par un auto-apprentissage ou auto-adaptation selon des lois qui assurent la stabilité globale du système.

Inspirer de cette idée, nous proposons dans ce chapitre deux approches de commande décentralisée floue adaptative (CDFA) en vue de la commande en poursuite des robots manipulateurs. La première approche est basée sur le travail de Seraji [SER89]. Nous développons la loi de commande CDFA où interviennent trois actions principales : une action de type PD adaptative pour la stabilisation de la dynamique de l'erreur ; une deuxième action de compensation des termes de couplages réalisée par un modèle flou de type Sugeno (MFS) adaptative. Les erreurs de reconstruction sont prises en charge par une action de type PID avec des gains fixes. Pour certaines classes de robots, il est possible de diminuer les dimensions du MFS, ainsi nous proposons la CDFA Robuste. Pour éviter les discontinuités dans la loi de CDFA Robuste, nous développons la loi CDFA Modifiée.

Quant à la deuxième approche, nous proposons une loi de CDFA simple où interviennent trois actions principales : Une action de stabilisation de type PID avec des gains constants, et deux actions de compensation réalisées par un MFS et un signal adaptatif. Les lois d'adaptations sont de type intégral avec des matrices et gains d'adaptation variables. Nous proposons la loi de CDFA Robuste pour diminuer les dimensions du MFS. Par la suite, nous proposons la loi de CDFA avec Zone Morte pour palier au problème de discontinuité dans la loi de commande précédente.

Dans les deux approches, la stabilité globale et la convergence asymptotique des erreurs de poursuite ainsi que la bornitude des différents signaux sont vérifiées via l'analyse de fonctions de Lyapunov candidates.

Le présent chapitre est organisé comme suit :

Dans la première section nous présentons le modèle dynamique indépendant de chaque articulation du robot. Sur la base de ce modèle, les techniques de commande de la première approche sont développées dans la deuxième section suivies par les résultats de simulation de l'application au robot PUMA 560. Les techniques de la deuxième approches sont développées dans la troisième section ainsi que les résultats de simulation. Le chapitre est terminé par une conclusion.

III.2 Modèle dynamique décentralisé : [MAD97]

Le modèle dynamique général d'un robot manipulateur rigide à un n degrés de liberté, peut être représenté par un système d'équations non linéaires du second ordre :

$$M(q)\ddot{q} + N(q, \dot{q}) + G(q) + H(\dot{q}) = u(t) \quad (\text{III.1})$$

avec,

$q \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$ et $\ddot{q} \in \mathbb{R}^n$: représentent, respectivement, les positions, les vitesses et les accélérations articulaires ;

$M(q) \in \mathbb{R}^{n \times n}$: matrice d'inertie.

$N(q, \dot{q}) \in \mathbb{R}^n$: couples et/ou forces de Coriolis et centrifuges ;

$G(q) \in \mathbb{R}^n$: vecteur des forces et/ou couples visqueux et secs ;

$u(t) \in \mathbb{R}^n$: vecteur des couples et/ou forces à appliquer aux articulations.

Les éléments de M , N , G et H sont des fonctions couplées très complexes qui dépendent de la configuration du robot (q), des vitesses de mouvements (\dot{q}) ainsi que les paramètres inertiels de la charge portée par l'élément terminal.

Le problème de la commande décentralisée des robots manipulateurs consiste à élaborer un ensemble de régulateurs indépendants aux différentes articulations. Le $i^{\text{ème}}$ régulateur délivre la commande à partir des mesures $q_i(t)$ et $\dot{q}_i(t)$ (trajectoire désirée) et assure une convergence asymptotique de $q_i(t)$ vers $q_i^d(t)$ [SER89]. Afin de développer un schéma de commande décentralisée, il est judicieux de considérer chaque articulation comme un sous-système du robot. Chaque sous système est interconnecté avec le reste des sous-systèmes à travers des couples et/ou des forces de couplage qui représentent des termes de couplage inertiels, de Coriolis et centrifuges, de frottement et de gravitation. Donc, le modèle dynamique du robot peut être considéré comme la collection de n équations différentielles non linéaires du second ordre donnée par :

$$[m_{ij}(q)\ddot{q}_j(t) + \sum_{j=1}^n m_{ij}(q)\ddot{q}_j(t)] + N_i(q, \dot{q}) + G_i(q) + H_i(\dot{q}) = u_i(t) \quad i=1 \dots n \quad (\text{III.2})$$

La notation par l'indice i réfère à la $i^{\text{ème}}$ articulation ($i^{\text{ème}}$ sous système). Le terme $m_{ij}(q)$ représente l'élément de la $i^{\text{ème}}$ ligne et la $j^{\text{ème}}$ colonne de la matrice $M(q)$. Les autres vecteurs sont décomposés de la manière suivante :

$$q = [q_1 \dots q_i \dots q_n]^T, \quad \dot{q} = [\dot{q}_1 \dots \dot{q}_i \dots \dot{q}_n]^T, \quad \ddot{q} = [\ddot{q}_1 \dots \ddot{q}_i \dots \ddot{q}_n]^T$$

$$G(q) = [G_1(q) \dots G_i(q) \dots G_n(q)]^T; \quad H(\dot{q}) = [H_1(\dot{q}) \dots H_i(\dot{q}) \dots H_n(\dot{q})]^T$$

$$N(q) = [N_1(q) \dots N_i(q) \dots N_n(q)]^T; \quad u(t) = [u_1(t) \dots u_i(t) \dots u_n(t)]^T$$

Posons

$$c_i(q, \dot{q}, \ddot{q}) = \sum_{\substack{j=1 \\ j \neq i}}^n m_{ij}(q) \ddot{q}_j + N_i(q, \dot{q}) + G_i(q) + H_i(\dot{q}) \quad (III.3)$$

on obtient :

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

Les équations (III.2) ou (III.3) et (III.4) représentent le modèle dynamique de la $i^{ème}$ articulation. Le terme $c_i(q, \dot{q}, \ddot{q})$ peut être considéré comme une perturbation constituant les termes de couplages avec les autres sous systèmes.

Les approches de synthèse, que nous allons développer, se basent sur le modèle relatif à chaque articulation et donné par (III.4). La loi de commande proposée utilise seulement les mesures locales de chaque articulation.

III.3 Première approche

Dans le cas général, la synthèse d'une loi de commande peut s'effectuer en deux étapes principales : choix de la structure de la commande et détermination d'une loi de commande assurant la stabilité du système.

La synthèse des schémas de commande décentralisée floue que nous proposons, considère le modèle dynamique décentralisé de chaque sous-système (articulation). Une écriture de ce dernier sous une nouvelle forme nous permettra de proposer une structure de la loi de commande. Cette loi est déterminée en utilisant la théorie de stabilité de Lyapunov.

III.3.1 Commande Décentralisée Floue Adaptative (CDFA)

III.3.1.1 Structure de commande

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

Les variables q_i, \dot{q}_i et \ddot{q}_i représentent respectivement la position, la vitesse et l'accélération de la $i^{ème}$ articulation. Soit $q_i^d(t), \dot{q}_i^d(t)$ et $\ddot{q}_i^d(t)$, respectivement la position, la vitesse et l'accélération désirées de la $i^{ème}$ articulation et :

- $e_i(t) = q_i^d(t) - q_i(t)$: erreur en position ;
- $\dot{e}_i(t) = \dot{q}_i^d(t) - \dot{q}_i(t)$: erreur en vitesse ;
- $\ddot{e}_i(t) = \ddot{q}_i^d(t) - \ddot{q}_i(t)$: erreur en accélération.

Le problème de commande décentralisée consiste à déterminer la commande $u_i = u_i(q_i, \dot{q}_i, \ddot{q}_i, q_i^d, \dot{q}_i^d, \ddot{q}_i^d, t)$ de manière que l'erreur $e_i(t)$ converge asymptotiquement vers zéro. D'après l'équation (III.4), on peut écrire :

$$m_{ii}(q) \ddot{e}_i = d_i(q, \dot{q}, \ddot{q}, \ddot{q}_i^d) - u_i(t) \quad (III.5)$$

où, $d_i(q, \dot{q}, \ddot{q}, \ddot{q}_i^d) = +c_i(q, \dot{q}, \ddot{q}) + m_{ii}(q) \ddot{q}_i^d$

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

Le terme d_i peut être décomposé en deux termes :

- Un terme qui peut être approximé par une fonction f_i dépendante seulement des états locaux (q_i, \dot{q}_i et \ddot{q}_i^d).
- Une partie englobant des termes non approximables notés $\varepsilon_i(t)$. Ce terme peut être vu comme l'erreur résultante de l'approximation du terme de couplage d_i par la fonction f_i .

est aussi appelé *erreur de reconstruction* [LEW93]. Cette erreur dépend des états des autres articulations, donc elle ne peut être bornée par les états de la $i^{ème}$ articulation.

Donc on écrit

$$d_i(q, \dot{q}, \ddot{q}, \ddot{q}_i^d) = f_i(q_i, \dot{q}_i, \ddot{q}_i^d) + \varepsilon_i(t) \quad i = 1, \dots, n \quad (III.6)$$

Considérons un modèle flou de type Sugeno (MFS), dont les trois variables d'entrée sont q_i, \dot{q}_i et \ddot{q}_i^d . Soit nf_1^i, nf_2^i et nf_3^i respectivement le nombre de fonctions d'appartenance associées aux variables q_i, \dot{q}_i et \ddot{q}_i^d . Ces fonctions sont notées $(\mu_{1l}^i)_{l=1, \dots, nf_1^i}, (\mu_{2l}^i)_{l=1, \dots, nf_2^i}$ et $(\mu_{3l}^i)_{l=1, \dots, nf_3^i}$. Dans ce cas, le nombre de règles obtenues est $nr^i = nf_1^i \times nf_2^i \times nf_3^i$.

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

$$\bullet \quad f_i(q_i, \dot{q}_i, \ddot{q}_i^d) = Z_i^T A_i^* \xi^i + \varepsilon_{f_i} \quad (III.7a)$$

Le premier terme est la sortie du MFS, où :

$$\bullet \quad Z_i^T = \begin{cases} 1 & \text{si MFS est d'ordre 0} \\ [1 \quad q_i \quad \dot{q}_i \quad \ddot{q}_i^d] & \text{si MFS est d'ordre 1.} \end{cases} \quad (III.7.b)$$

- A_i^* : une matrice constante qui regroupe les gains des conséquences des règles floues. Elle est définie par [SPO95a,b] :

$$A_i^* = \arg \min_{A_i \in \Omega_i^*} \left[\sup_{q_i \in \Omega_i^1, \dot{q}_i \in \Omega_i^2, \ddot{q}_i^d \in \Omega_i^3} |Z_i^T A_i \xi^i - f_i(q_i, \dot{q}_i, \ddot{q}_i^d)| \right] \quad (III.7c)$$

où $\Omega_0^i, \Omega_1^i, \Omega_2^i$ et Ω_3^i représentent, respectivement les domaines admissibles des variables A_i, q_i, \dot{q}_i et \ddot{q}_i^d , avec :

$$A_i = \begin{cases} [a_{0l}^i \cdots a_{0nr^i}^i] & \text{si MFS est d'ordre 0} \\ \begin{bmatrix} a_{0l}^i & \cdots & a_{0nr^i}^i \\ a_{1l}^i & \cdots & a_{1nr^i}^i \\ a_{2l}^i & \cdots & a_{2nr^i}^i \\ a_{3l}^i & \cdots & a_{3nr^i}^i \end{bmatrix} & \text{si MFS est d'ordre 1.} \end{cases} \quad (III.7.d)$$

- ξ^{iT} : vecteur des degrés d'activation des différentes règles floues, il est défini par :

$$\xi^{iT} = [\xi_1^i \cdots \xi_{nr^i}^i] \quad \text{et} \quad \xi_k^i = \frac{\mu_{1k_1}^i(q_i) \times \mu_{2k_2}^i(\dot{q}_i) \times \mu_{3k_3}^i(\ddot{q}_i^d)}{\sum_{\substack{1 \leq l_1 \leq nf_1^i \\ 1 \leq l_2 \leq nf_2^i \\ 1 \leq l_3 \leq nf_3^i}} \mu_{1l_1}^i(q_i) \times \mu_{2l_2}^i(\dot{q}_i) \times \mu_{3l_3}^i(\ddot{q}_i^d)} \quad k = 1 \cdots nr^i \quad (III.7e)$$

et, $k = k_1 + \sum_{j=2}^3 (k_j - 1) \prod_{j'=1}^j nf_{j'}^i$, $k_1 = 1 \cdots nf_1^i$, $k_2 = 1 \cdots nf_2^i$, $k_3 = 1 \cdots nf_3^i$, $i = 1 \dots n$

- Le deuxième terme ε_{f_i} est une constante qui représente l'erreur de reconstruction de f_i . Elle résulte de l'utilisation d'un nombre fini de fonctions d'appartenance.

D'après sa définition, A_i^* représente les gains optimaux des conséquences des règles floues des MFS, permettant d'obtenir une erreur de reconstruction ε_f^i minimale, elle même dépend du modèle flou adopté (le type, la structure et les paramètres).

D'après (III.6) (III.7a) l'équation (III.5) devient :

$$m_{ii}(q)\ddot{e}_i = Z_i^T A_i^* \xi^i + \varepsilon_f^i + \varepsilon_i(t) - u_i(t) \quad i = 1 \dots n \quad (III.8)$$

Sur la base du modèle dynamique de l'erreur, donnée par l'équation (III.8), nous proposons la loi de commande suivante :

$$u_i(t) = k_{pi}(t)e_i(t) + k_{vi}(t)\dot{e}_i(t) + Z_i^T \hat{A}_i(t)\xi^i + \hat{\varepsilon}_i(t) \quad (III.9)$$

où,

- $k_{pi}(t)e_i(t) + k_{vi}(t)\dot{e}_i(t)$: une action proportionnelle-dérivée (PD) avec des gains k_{pi} et k_{vi} adaptatifs ;
- $Z_i^T \hat{A}_i(t)\xi^i$: la sortie d'un MFS pour estimer la fonction $Z_i^T A_i^*(t)\xi^i$. La matrice \hat{A}_i (estimée de A_i^*) doit être déterminée en temps réel ;
- $\hat{\varepsilon}_i(t)$: signal auxiliaire utilisé pour compenser les erreurs ε_f^i et ε_i .

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

Hypothèses :

Avant de développer la loi de commande, il est nécessaire de considérer les hypothèses suivants :

H1 : Les trajectoires désirées $q_i^d(t)$, $\dot{q}_i^d(t)$ et $\ddot{q}_i^d(t)$ $i = 1 \dots n$ sont bornées ;

H2 : Les éléments diagonales de la matrice d'inertie $(m_{ii}(q))$ $i = 1 \dots n$ varient lentement en fonction du temps ;

H3 : Les erreurs de reconstruction $\varepsilon_1(t), \varepsilon_2(t), \dots, \varepsilon_n(t)$ varient lentement, respectivement, par rapport aux signaux auxiliaires $\hat{\varepsilon}_1(t), \hat{\varepsilon}_2(t), \dots, \hat{\varepsilon}_n(t)$.

Proposons les lois d'adaptation suivantes :

$$\dot{k}_{pi} = \alpha_{1i} r_i e_i + \alpha_{2i} \frac{d}{dt}(r_i e_i) \quad i = 1 \dots n \quad (III.10a)$$

$$\dot{k}_{vi} = \beta_{1i} r_i \dot{e}_i + \beta_{2i} \frac{d}{dt}(r_i \dot{e}_i) \quad i = 1 \dots n \quad (III.10b)$$

$$\dot{\hat{A}}_i(t) = \Gamma_{1i} Z_i \xi^{iT} r_i + \Gamma_{2i} \frac{d}{dt}(Z_i \xi^{iT} r_i) \quad i = 1 \dots n \quad (III.10c)$$

$$\dot{\hat{\varepsilon}}_i(t) = \gamma_{1i} r_i + \gamma_{2i} \frac{d}{dt}(r_i) \quad i = 1 \dots n \quad (III.10d)$$

avec,

$$r_i = p_{2i} e_i + p_{3i} \dot{e}_i \quad i = 1 \dots n$$

où,
 $\alpha_{1i}, \beta_{1i}, \gamma_{1i} > 0$; $\alpha_{2i}, \beta_{2i}, \gamma_{2i} \geq 0$ (des scalaires) ;

Γ_{1i} et Γ_{2i} deux matrices définies positives ;

- $p_{2i}, p_{3i} > 0$ (des scalaires).

Proposition 1:

Etant donné un robot manipulateur à n degrés de liberté, dont le modèle dynamique de l'erreur de poursuite de chaque articulation peut être mis sous la forme (III.8).

Si on choisit la loi de commande (III.9) et en tenant compte des hypothèses H1, H2 et H3, et les lois d'adaptation (III.10a,b,c,d) on aura :

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

R2 : Les variables $k_{pi}(t), k_{vi}(t), \hat{A}_i(t)$ et $\hat{e}_i(t), i = 1 \dots n$, sont bornées ;

R3 : Les erreurs $e_i(t) i = 1 \dots n$ sont bornées et convergent, au moins asymptotiquement vers zéro.

Démonstration

Pour la démonstration, on utilise le théorème de stabilité de Lyapunov avec un corollaire de convergence asymptotique [KRS95][LEW93]. Une fonction de Lyapunov candidate est proposée en se basant sur celle utilisée par Seraji [SER89].

Cette démonstration est donnée dans l'Annexe B. Démo1.

Schéma de principe

Le schéma de principe de la commande décentralisée floue adaptative (CDFA), en utilisant la loi de commande (III.9) et (III.10a,b,c,d) est illustré par la figure (III.1).

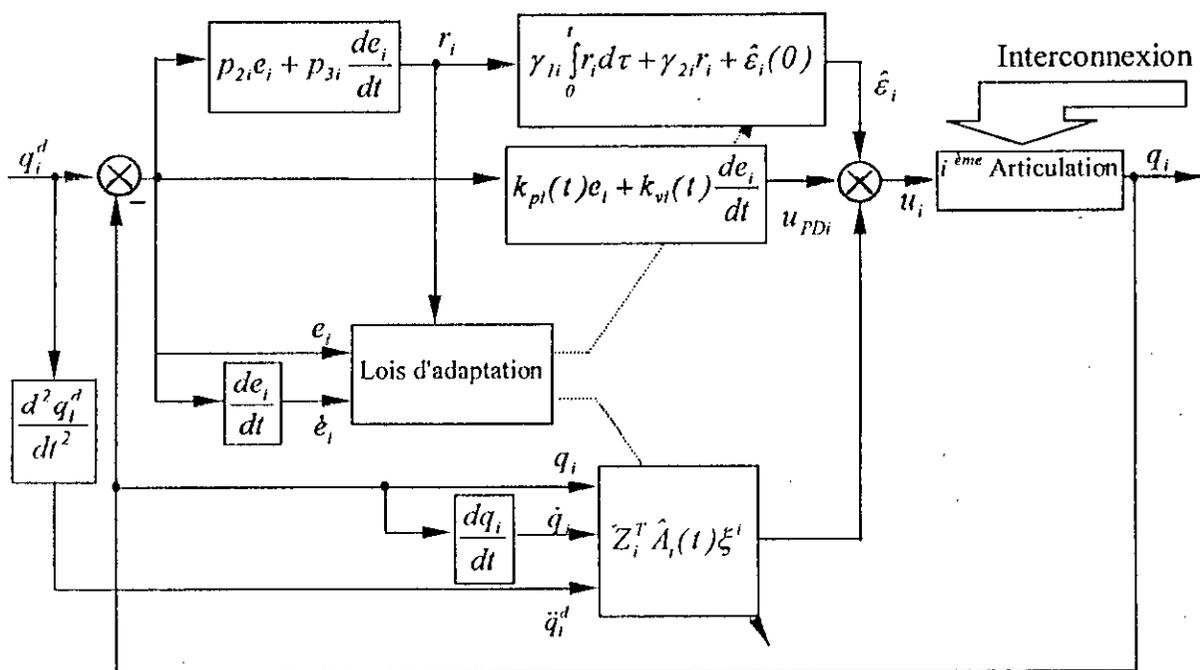


figure (III.1) : Schéma de principe de la CDFA.

Sous les hypothèses H1, H2 et H3 la Proposition 1 révèle que l'utilisation de la loi de commande décentralisée floue adaptative (CDFA) assure la bornitude et la convergence asymptotique des erreurs de poursuite vers zéro. La bornitude des commandes $u_i(t) i = 1 \dots n$ et des variables $q_i, \dot{q}_i, \ddot{q}_i, k_{pi}, k_{vi}, \hat{A}_i$ et $\hat{e}_i, i = 1 \dots n$ est aussi garantie.

Dans l'expression de la commande $u_i(t)$, on distingue trois termes :

1. Un premier terme caractérise un retour d'état variable réalisé par une action Proportionnelle-Dérivée (PD), avec des gains k_{pi}, k_{vi} à adapter en temps réel. La bornitude de ces gains a été démontrée ;

2. Un deuxième terme, qui est la sortie d'un modèle flou adaptatif de type Sugeno (MFS). Il a été adopté afin de compenser les termes de couplages. Les gains des conséquences des règles floues sont adaptés en temps réel selon une loi de type proportionnelle-intégrale (PI).

D'autre part, si on considère l'expression d'adaptation de la matrice \hat{A}_i (donnée par (III.10c)), on peut déduire :

$$\dot{\hat{a}}_y^i = \Gamma_{11,i}^i Z_i^T \xi_j^i r_i + \Gamma_{21,i}^i \frac{d}{dt} (Z_i^T \xi_j^i r_i), j = 1 \dots m^i, i = 1 \dots n \quad (\text{III.11})$$

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

$j = 1 \dots m^i$ indice de la règle ;

$i = 1 \dots n$ indice de l'articulation ;

$\Gamma_{ll,i}^i$ ($\Gamma_{2l,i}^i$) : la ligne l de la matrice Γ_{ll} (respectivement de Γ_{2l}).

De l'expression précédente, les gains \hat{a}_y^i des conséquences des règles floues, subiront des modifications seulement lorsque les règles correspondantes sont activées ($\xi_j^i \neq 0$), car dans ce cas on obtient $\dot{\hat{a}}_y^i \neq 0$. Cependant les gains des règles inactivées gardent leurs valeurs constantes ($\dot{\hat{a}}_y^i = 0$).

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

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

3. Un signal auxiliaire ($\hat{\varepsilon}_i$) qui représente un retour d'état réalisé par une action proportionnelle-intégrale-dérivée (PID) avec des gains fixes. Ce terme est nécessaire pour faire face à l'erreur de reconstruction.

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

III.3.1.2 Application au robot PUMA560

Nous avons appliqué la CDEFA pour la commande en poursuite du robot manipulateur PUMA560 présenté dans l'annexe A. Afin d'effectuer les différents tests, nous avons choisi la trajectoire de Leahvy de transition cycloïdal. Avec la méthode de simulation Runge-Kutta d'ordre 4, le pas de simulation est fixé à 1 ms. Les intégrations dans la loi de commande sont évaluées par la méthode des trapèzes.

Paramètres du régulateur :

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

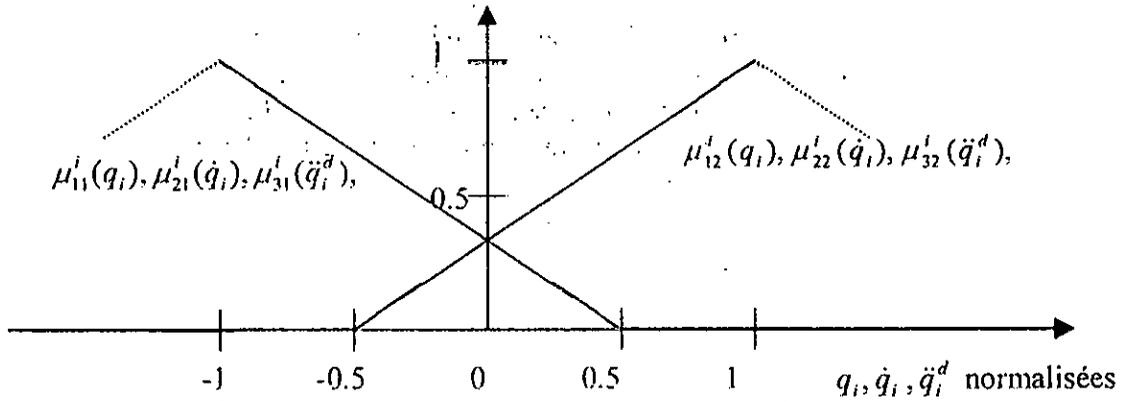


figure (III.2) Les fonctions d'appartenance des variables d'entrées

Les paramètres de la loi de commande sont regroupés dans la table (III.1)

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

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

Avec $\mathbf{1}_{i \times j}$ est la matrice de dimensions $i \times j$ d'éléments tous égaux à 1.

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

Résultats et performances

Test sans charge : les figures (III.3a,b) montrent l'évolution des différents variables obtenues à partir d'un test sans charge. On remarque que les articulations poursuivent les trajectoires désirées avec des faibles erreurs et des commandes lisses. Les erreurs sont inférieures à 0.2 deg et s'annulent en régime permanent. L'intervention du modèle flou est principale dans la génération de la commande (voir les sorties des modèles flous notées u_f). Les allures des gains k_{pi} , k_{vi} et a_{ij}^i sont lisses dès qu'ils sont adaptés selon une loi de type proportionnel-intégral. L'adaptation des gains a_{ij}^i n'est activée que dans des zones particulières des variables articulaires (principe de perturbation minimale).

Test de variation de charge : dans ce test l'élément terminal porte une charge de 2 Kg puis de 4 Kg jusqu'à l'instant $t=0.75$ sec où elle est relâchée. Les réponses obtenues sont montrées par la figure (III.4). On constate que les articulations continuent remarquablement la poursuite des trajectoires malgré la violation de l'hypothèse sur la variation lente de l'inertie du robots.

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

Test de rupture de commande : Pour montrer l'efficacité de la décentralisation dans le schéma de commande, nous appliquons une commande nulle à la deuxième articulation (chute libre) et nous observons le comportement des régulateurs associés aux deux autres articulations. Les réponses obtenues sont montrées par la figure (III.6). Nous constatons que la première et troisième articulation continue la poursuite des trajectoires avec une légère augmentation de l'erreur qui reste inférieure à 0.3 deg. Des commandes relativement considérables sont générées pour faire face à l'importante inertie de la liaison en rupture. Cette dernière, ayant une dynamique lente, impose un temps plus long pour l'établissement du régime permanent.

Test de variation du pas d'échantillonnage : Nous avons augmenté le pas d'échantillonnage à la valeur 10 ms puis à 15 ms. Pour maintenir les performances du schéma de commande, il était nécessaire de diminuer les valeurs des paramètres de notre régulateur comme suit :

$$\alpha_{1i} = \beta_{1i} = 460, \alpha_{2i} = \beta_{2i} = 0 \text{ pour } T = 10 \text{ msec};$$

$$\alpha_{1i} = \beta_{1i} = 100, \alpha_{2i} = \beta_{2i} = 0 \text{ pour } T = 15 \text{ msec};$$

La figure (III.7) montre les réponses obtenues. Nous constatons une augmentation relative des erreurs de poursuite qui restent inférieures à 0.5 deg. Ces erreurs s'annulent en régime permanent. Avec l'augmentation du pas d'échantillonnage, des ondulations de la commande apparaissent pour refléter une lente adaptation.

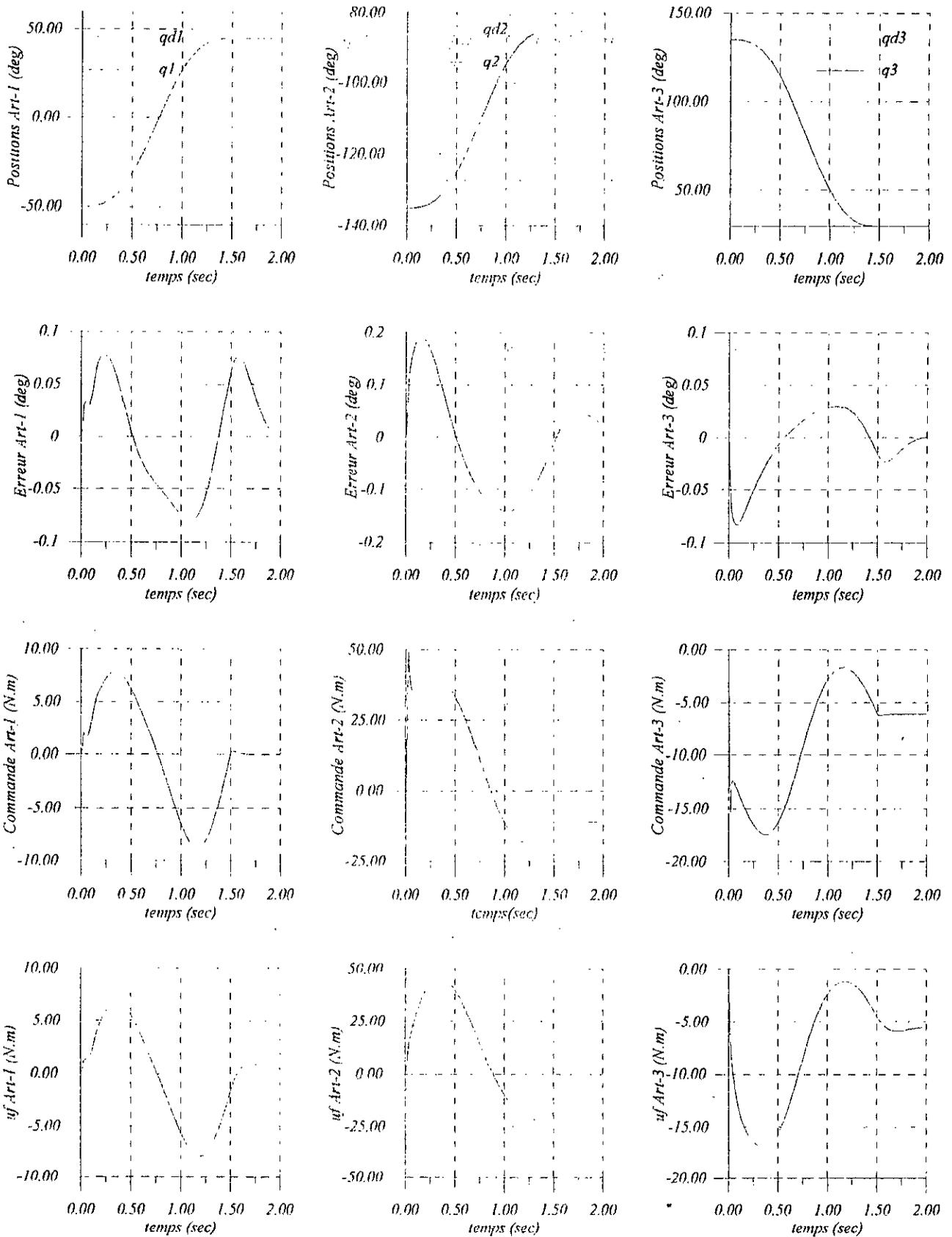


figure (III.3a) : Réponses des différentes articulations soumises à la loi CDFA :
Test sans charge

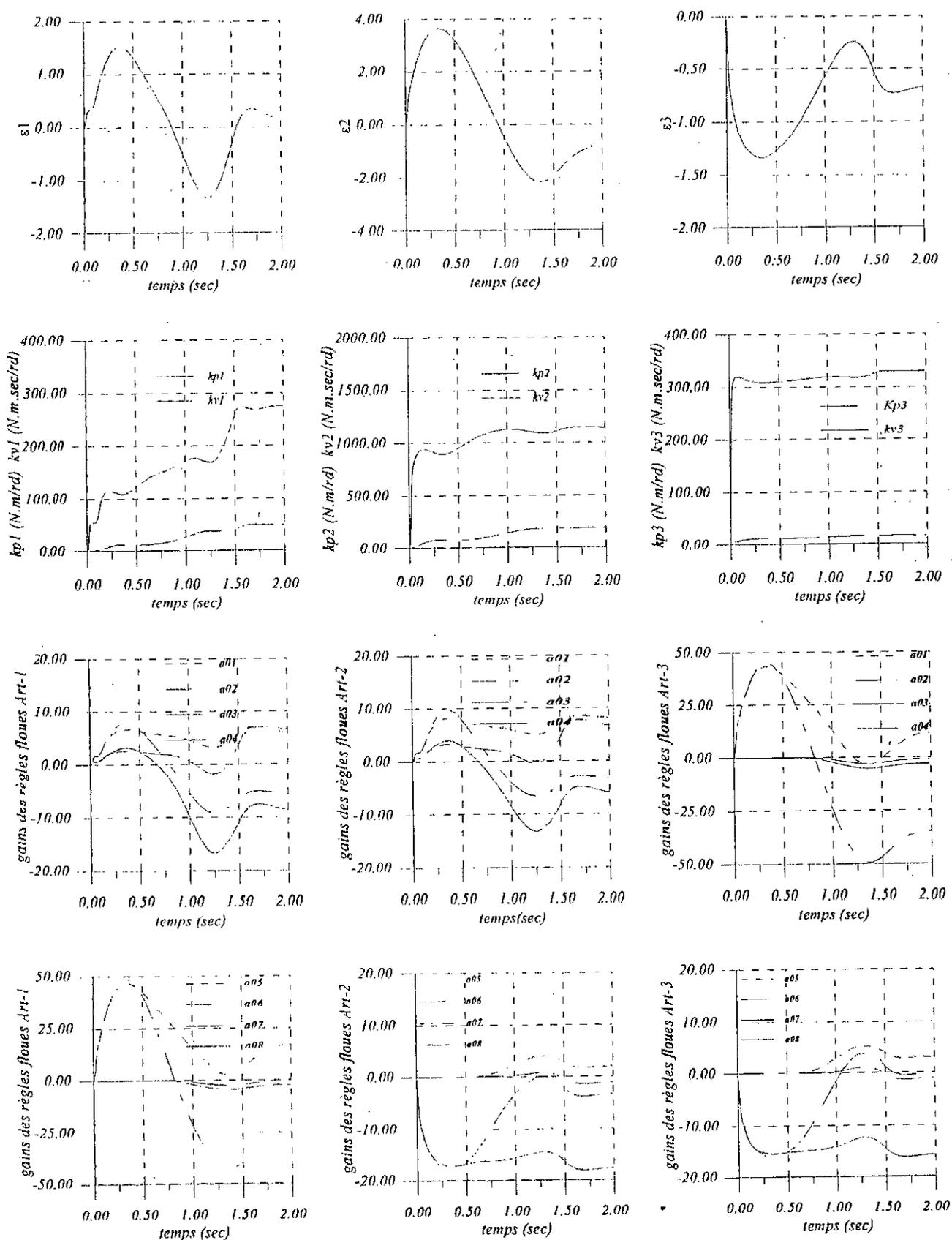
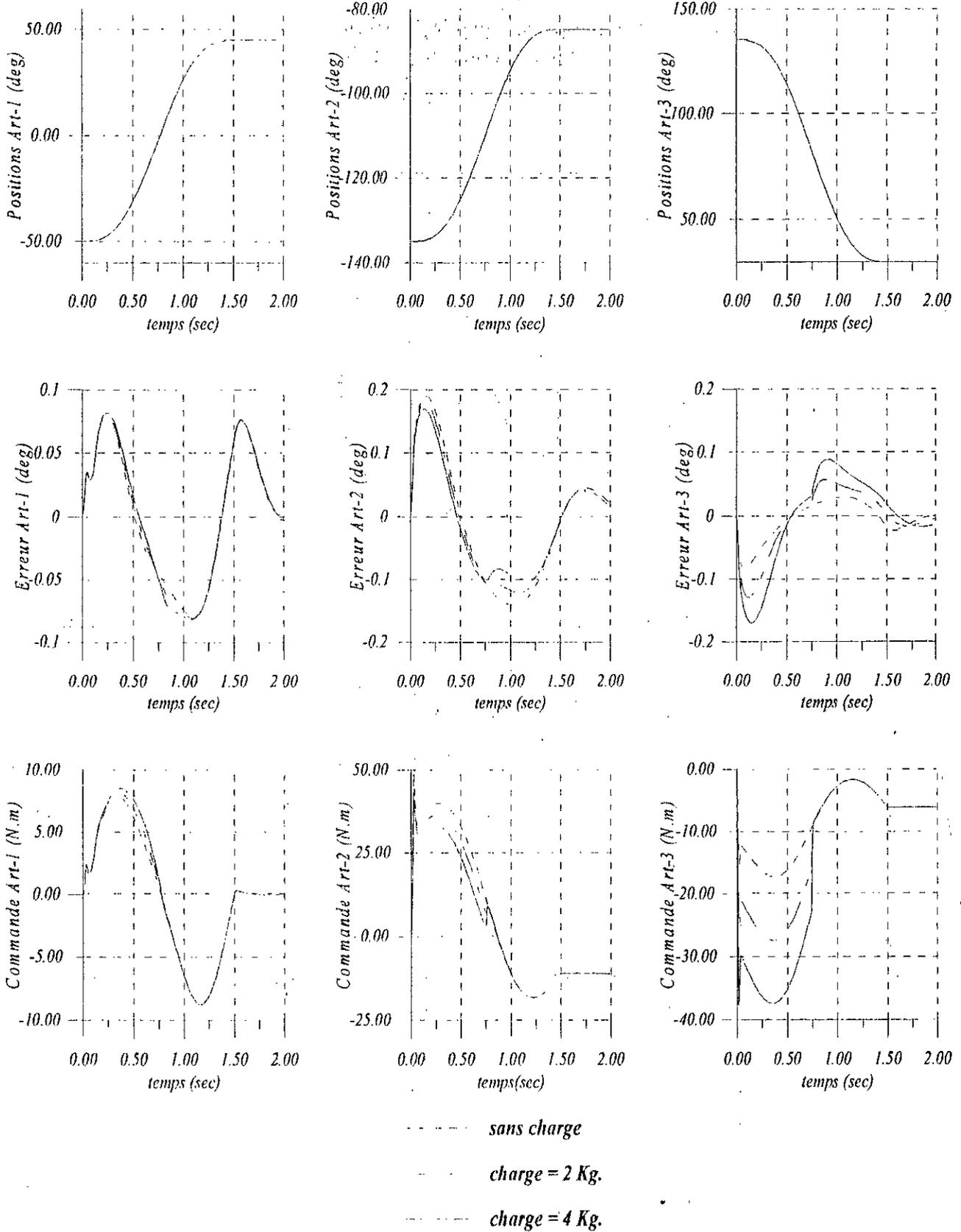


figure (III.3b) : Réponses des différentes articulations soumises à la loi CDFA :
Test sans charge



*figure (III.4) : Réponses des différentes articulations soumises à la loi CDFA :
Test de variation de charge*

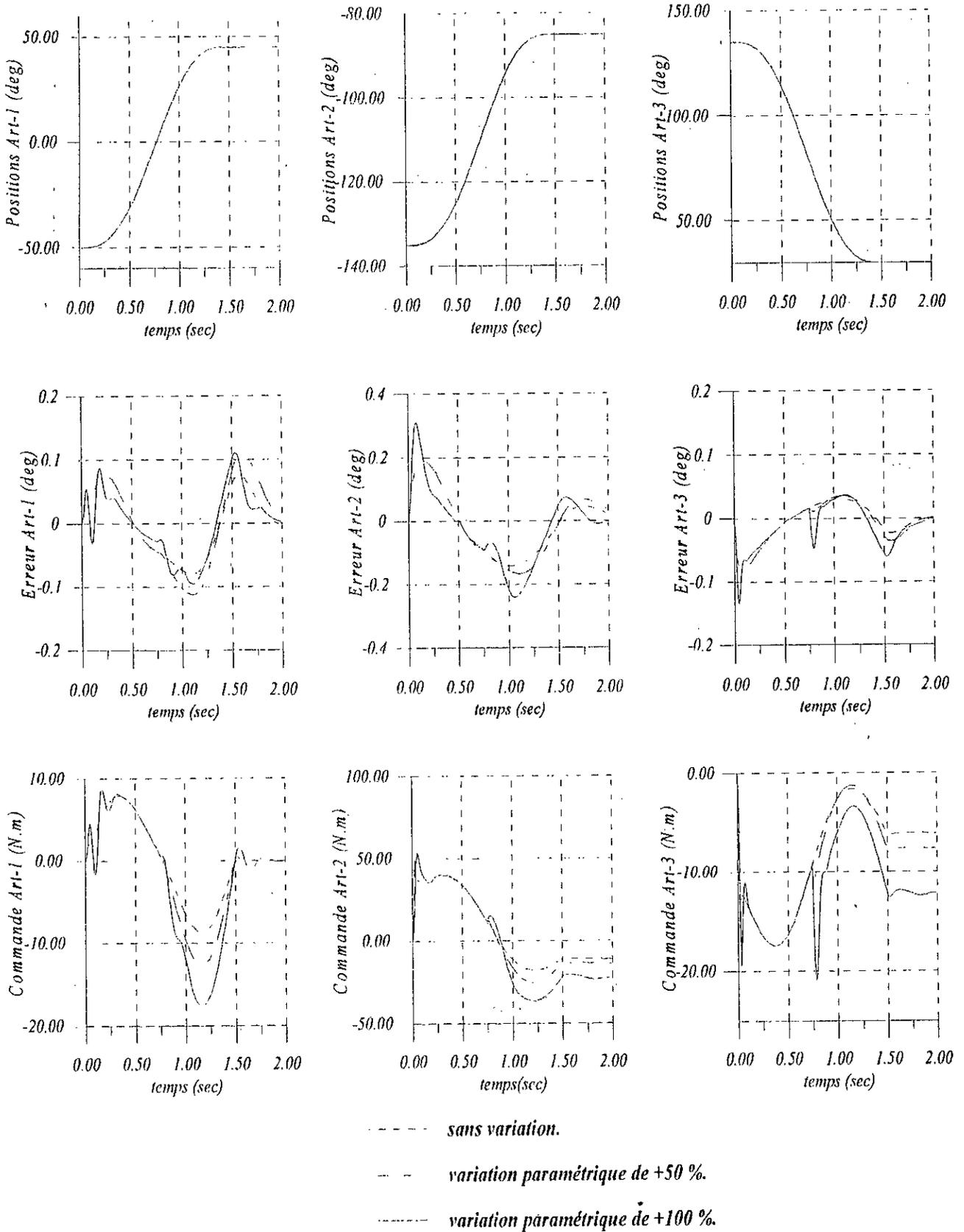
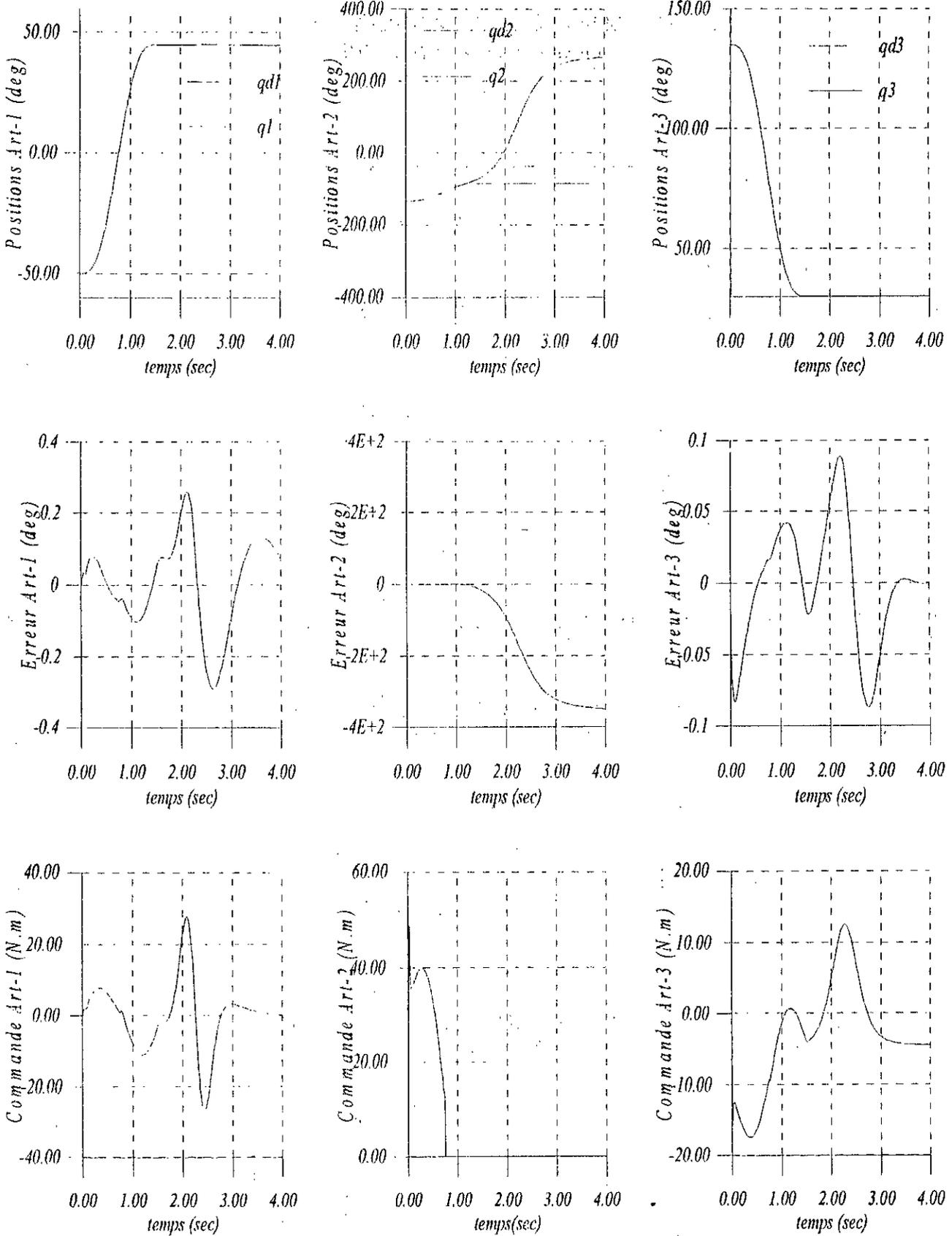


figure (III.5) : Réponses des différentes articulations soumises à la loi CDFA :
Test de variation paramétrique



**figure (III.6) : Réponses des différentes articulations soumises à la loi CDFA :
Test de rupture de commande.**

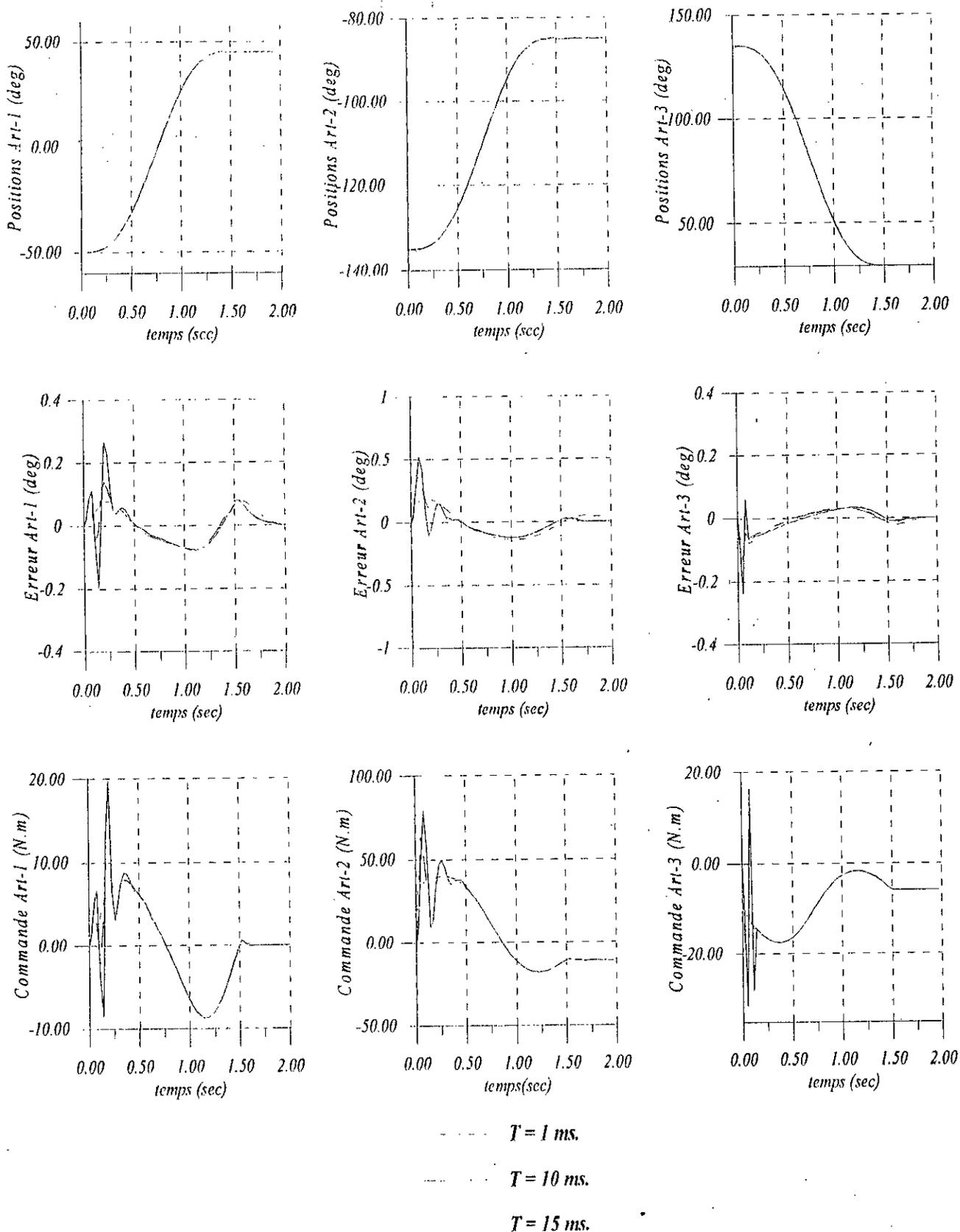


figure (III.7) : Réponses des différentes articulations soumises à la loi CDEA :
Test de variation du pas d'échantillonnage

III.3.2 Commande Décentralisée Floue Adaptative Robuste (CDFAR).

III.3.2.1 Structure de commande

Dans la loi de commande développée dans la section précédente, nous avons utilisé un modèle flou à trois variables d'entrées (q_i, \dot{q}_i et \ddot{q}_i^d) pour compenser le terme de couplage. Pour les articulations ayant des inerties ou des masses $m_{ii}(q)$ bornées, il est possible d'envisager un modèle flou à deux entrées seulement (q_i, \dot{q}_i). Cependant, il sera nécessaire d'introduire un terme supplémentaire de robustesse. Un cas particulier où les inerties $m_{ii}(q)$ sont bornées, est celui des robots ayant des articulations toutes rotoïdes.

On considère l'équation (III.5), on peut écrire:

$$m_{ii}(q)\ddot{e}_i = c_i(q, \dot{q}, \ddot{q}) + m_{ii}(q)\ddot{q}_i^d - u_i(t) \quad (III.12)$$

Avec les mêmes notations de la section précédente, utilisons un modèle flou de type Sugeno (MFS) à deux entrées q_i, \dot{q}_i pour estimer le terme de couplage $c_i(q, \dot{q}, \ddot{q})$. On écrit :

$$c_i(q, \dot{q}, \ddot{q}) = f_i(q_i, \dot{q}_i) + \varepsilon_i(t) \quad (III.13a)$$

où, $f_i(q_i, \dot{q}_i)$ est la partie de $c_i(q, \dot{q}, \ddot{q})$ approximable par un MFS. Donc, on écrit :

$$f_i(q_i, \dot{q}_i) = Z_i^T A_i^* \xi^i + \varepsilon_{fi} \quad (III.13b)$$

avec,

- $Z_i^T = \begin{cases} 1 & \text{si MFS est d'ordre 0} \\ [1 \quad q_i \quad \dot{q}_i] & \text{si MFS est d'ordre 1.} \end{cases} \quad (III.13c)$

- $A_i^* = \arg \min_{A_i \in \Omega_0^i} \left[\sup_{q_i \in \Omega_1^i, \dot{q}_i \in \Omega_2^i} |Z_i^T A_i \xi^i - f_i(q_i, \dot{q}_i)| \right] \quad (III.13d)$

avec,

$$A_i = \begin{cases} \begin{bmatrix} a_{01}^i & \dots & a_{0m}^i \end{bmatrix} & \text{si MFS est d'ordre 0} \\ \begin{bmatrix} a_{11}^i & \dots & a_{1m}^i \\ a_{21}^i & \dots & a_{2m}^i \end{bmatrix} & \text{si MFS est d'ordre 1.} \end{cases} \quad (III.13e)$$

- $\xi^i = [\xi_1^i \dots \xi_{m'}^i]^T$ et $\xi_k^i = \frac{\mu_{1k_1}^i(q_i) \times \mu_{2k_2}^i(\dot{q}_i)}{\sum_{\substack{1 \leq l_1 \leq m_1^i \\ 1 \leq l_2 \leq m_2^i}} \mu_{1l_1}^i(q_i) \times \mu_{2l_2}^i(\dot{q}_i)} \quad (III.13f)$

où, $k = k_1 + (k_2 - 1)nf_1^i, k_1 = 1 \dots nf_1^i, k_2 = 1 \dots nf_2^i, i = 1 \dots n$

On peut écrire :

$$m_{ii}(q)\ddot{e}_i(t) = Z_i^T A_i^* \xi^i + \varepsilon_i(t) + m_{ii}(q)\ddot{q}_i^d - u_i(t) \quad (III.14)$$

Nous proposons la loi de commande suivante :

$$u_i(t) = k_{pi}(t)e_i(t) + k_{vi}(t)\dot{e}_i(t) + Z_i^T \hat{A}_i \xi^i + \hat{e}_i(t) + u_{si}(t) \quad (III.15)$$

Le terme $u_{si}(t)$ a été ajouté pour introduire une robustesse vis à vis du terme $m_{ii}(q)\ddot{q}_i^d$.

En remplaçant la loi de commande (III.15) dans l'équation dynamique de l'erreur (III.14), on obtient :

$$m_{ii}(q)\ddot{e}_i(t) = -k_{pi}(t)e_i(t) - k_{vi}(t)\dot{e}_i(t) + Z_i^T \tilde{A}_i \xi^i + \tilde{\varepsilon}_i(t) + (m_{ii}\ddot{q}_i^d - u_{si}(t)) \quad (III.16)$$

avec, $\tilde{A}_i = A_i^* - \hat{A}_i$; $\tilde{\varepsilon}_i = \varepsilon_i^*(t) - \hat{\varepsilon}_i(t)$

Considérons l'hypothèse suivante :

Hypothèse

H4 : l'inertie (ou la masse) $m_{ii}(q)$ est bornée par une constante connue :

$$m_{ii}(q) \leq k_{Si}, \forall q$$

Proposition 2 :

Etant donné un robot manipulateur à n degrés de liberté dont le modèle dynamique de l'erreur de poursuite de chaque articulation peut être mis sous la forme (III.14).

Soit la loi de commande (III.15) et en tenant compte des hypothèses H1, H2, H3 et H4, et les lois d'adaptation (III.10a,b,c,d). On choisit :

$$u_{Si} = k_{Si} \left| \ddot{q}_i^d \right| \text{sign}(r_i), i = 1 \dots n \quad (\text{III.17})$$

avec, $r_i = p_2 \dot{e}_i + p_3 e_i$ $p_2, p_3 > 0$

sign : la fonction signe

alors, on aura :

R1. Les états q_i, \dot{q}_i et \ddot{q}_i sont bornés;

R2. Les variables $k_{pi}, k_{vi}, \hat{A}_i$ et $\hat{\varepsilon}_i$ $i = 1 \dots n$ sont bornées;

R3. Les erreurs de poursuite $e_i(t)$ $i = 1 \dots n$ sont bornées et convergentes au moins asymptotiquement vers zéro.

Démonstration

La démonstration est basée sur celle de la proposition précédente (proposition 1). Le lecteur pourra la consulter en Annexe B. Démo 2.

Schéma de principe

La loi de commande donnée par (III.15), (III.17) est menue d'une action de robustesse vis à vis du terme supplémentaire lié à l'inertie ($m_{ii}(q)$). Le schéma de principe de cette loi de commande décentralisée floue adaptative robuste (CDFAR) est donnée par la figure (III.8).

D'après la Proposition 2, sous les hypothèses H1, H2, H3 et H4, la loi de commande décentralisée floue adaptative robuste assure la bornitude et la convergence asymptotique des erreurs de poursuite de toutes les articulations. La bornitude des variables $q_i, \dot{q}_i, \ddot{q}_i$, des gains $k_{pi}, k_{vi}, \hat{A}_i$ du signal $\hat{\varepsilon}_i$ et de la commande u_i est garantie.

Cette loi de commande ne peut être appliquée qu'aux robots ayant des articulations dont les inerties $m_{ii}(q)$ $i = 1 \dots n$ peuvent être bornées par des constantes connues (hypothèse H4).

Grâce à l'introduction du terme de robustesse, on a réduit le nombre des variables d'entrées des modèles flous à deux (q_i, \dot{q}_i), ce qui permet de diminuer le nombre des règles floues, et par conséquent le nombre des gains \hat{a}_{ij}^i à adapter.

Remarques :

1°. La démonstration de la Proposition 1 (CDFA) et la Proposition 2 (CDFAR) a été effectuée en considérant chaque articulation indépendamment. Donc, on peut utiliser ces deux lois en vue de la commande décentralisée du même robot. La CDFAR est applicable seulement lorsque l'articulation considérée vérifie l'hypothèse H4.

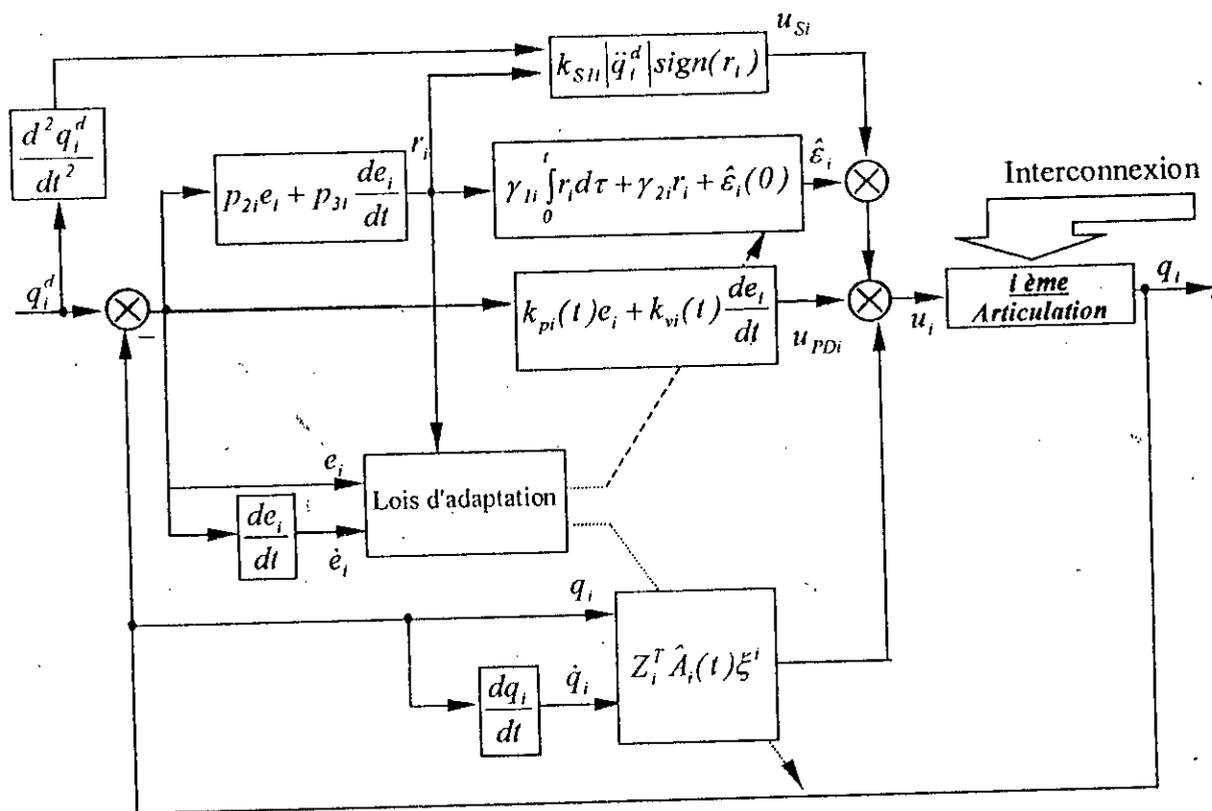


figure (III.8) : Schéma de principe de la CDFAR.

2°. La présence de la fonction $sign(\cdot)$ dans le terme de robustesse $u_{si}(t)$ peut provoquer des discontinuités de la commande $u_i(t)$, lorsque l'erreur filtrée $r_i(t)$ change de signe. Ce qui pose le problème de la validité des hypothèses (excitation des dynamiques négligées) et des problèmes aux niveaux des actionneurs (réalisation des commandes) ou ce qu'on appelle phénomène de "chattering". Pour palier à ces problèmes, une solution consiste à remplacer la fonction $sign$ par une fonction continue telle que :

- La fonction proposée dans [SLO83] :

$$C_{\delta_i}(r_i) = \frac{r_i}{|r_i| + \delta_i} \quad \delta_i : \text{petite constante}$$

positive.

- La fonction améliorée proposée dans [MAD97] :

$$M(r_i) = \frac{r_i}{|r_i| + \delta_{oi} / (1 + \delta_{ii} |e_i|)}$$

δ_{oi} : petite constante positive, δ_{ii} : grande constante positive.

- La fonction de saturation :

$$sat\left(\frac{r_i}{\Delta_i}\right) = \begin{cases} \frac{r_i}{\Delta_i} & |r_i| \leq \Delta_i \\ sign(r_i) & |r_i| > \Delta_i \end{cases}$$

Δ_i : Une petite constante positive.

Cette modification permet un lissage de la commande $u_i(t)$, cependant la convergence asymptotique des erreurs n'est plus assurée[SPO95c].

III.3.2.2 Application au robot PUMA560

Nous avons appliqué la technique de la CDFAR pour la commande en poursuite du robot manipulateur PUMA560.

Paramètres du régulateur :

Chaque régulateur local est menu d'un modèle flou à deux entrées q_i et \dot{q}_i . Nous avons adopté deux fonctions d'appartenances triangulaires pour chacune des deux variables d'entrées dont les allures sont montrées par la figure (III.2). Ainsi le modèle flou est caractérisé par quatre règles.

Les paramètres de la loi de commande sont regroupés dans la table (III.2)

	Symboles des paramètres	Première articulation	Deuxième articulation	Troisième articulation
Erreur Filtrée r_i	$p_{1i} ; p_{2i}$	300 ; 100	300 ; 100	300 ; 100
Action PD Adaptative $k_{pi}e_i + k_{vi}\dot{e}_i$	$\alpha_{1i} ; \alpha_{2i}$ $\beta_{1i} ; \beta_{2i}$	930 ; 9.3 93 ; 9.3	930 ; 9.3 93 ; 9.3	930 ; 9.3 93 ; 9.3
Facteur de normalisation	$q_i ; \dot{q}_i$	4 ; 5.33	4 ; 5.33	4 ; 5.33
Modèle flou $Z_i^T \hat{A}_i(t) \xi^i$	$\Gamma_{1i} ; \Gamma_{2i}$	$80.5 \times \mathbf{1}_{1 \times 4} ; 0 \times \mathbf{1}_{1 \times 4}$	$80.5 \times \mathbf{1}_{1 \times 4} ; 0 \times \mathbf{1}_{1 \times 4}$	$80.5 \times \mathbf{1}_{1 \times 4} ; 0 \times \mathbf{1}_{1 \times 4}$
Signal auxiliaire	$\gamma_{1i} ; \gamma_{2i}$	0.1 ; 0	0.1 ; 0	0.1 ; 0
Terme de robustesse	k_{s1i}	6	8.5	1.5

Table (III.2) : Paramètres du régulateur sous la loi CDFAR

Avec $\mathbf{1}_{i \times j}$ est la matrice de dimensions $i \times j$ d'éléments tous égaux à 1.

Les paramètres k_{s1i} sont déterminés à partir du modèle dynamique du robot.

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

Résultats et performances

Test sans charge : les résultats obtenus suite à un test sans charge sont montrés par la figure (III.9). Nous remarquons que les trois articulations effectuent la poursuite des trajectoires avec des erreurs maximales inférieures à 0.6 deg qui s'annule en régime permanent et des commandes oscillatoires.

Cela est dû à la discontinuité du terme de robustesse dans la loi de commande. Vue l'absence de l'effet gravitationnel, ces oscillations sont de plus persistantes dans le cas de la première articulation où la commande atteint 30 N.m. Dès que ces oscillations sont modulées par l'accélération désirée, elles disparaissent en régime permanent.

Test sans charge avec lissage: Dans le but d'obtenir une bonne poursuite avec des commandes réalisables, nous avons introduit un lissage dans la commande. Ainsi, nous avons remplacé la fonction *sign* du terme de robustesse par la fonction d'amélioration suivante :

$$C_{\delta}(r_i) = \frac{r_i}{|r_i| + \delta_i} \quad \delta_1 = 1.00 ; \delta_2 = \delta_3 = 0.08$$

Les résultats obtenus sont montrés par les figure (III.10a,b). Nous constatons que les commandes prennent des allures lisses avec des valeurs moins importantes que celles du cas sans lissage. Les erreurs de poursuite sont inférieures à 0.6 deg. L'ordre de grandeur des sorties des modèles flous (*uf*) indique l'importance de leurs interventions. Caractérisées par une variation lisse, les gains des règles flous prennent des valeurs constantes en régime permanent.

Test de variation de charge : une charge de 2 Kg puis de 4 Kg est transportée par l'élément terminal pour être relâchée à l'instant $t=0.75$ sec. La figure (III.11) montre les résultats obtenus. Nous constatons la robustesse du régulateur vis-à-vis de cette variation.

Test de variation paramétrique : Une variation sur tous les paramètres du modèle du robot de +50% puis de +100 % est introduite à l'instant $t=0.75$ sec. Les réponses obtenues sont illustrées par la figure (III.12). Nous remarquons une poursuite satisfaisante où les erreurs sont inférieures à 0.6 deg. Nous constatons une adaptation instantanée des commandes afin d'assurer une poursuite avec des faibles erreurs.

Test de rupture de commande : La figure (III.13) montre les résultats obtenus suite au test de rupture de la commande de la deuxième articulation. Nous remarquons que les deux autres articulations continuent la poursuite des trajectoires désirées avec une erreurs maximales inférieures à 0.6 deg.

Test d'association de la CDFFA et CDFAR: le but de ce test est de montrer l'applicabilité de l'association de la CDFFA avec la CDFAR pour la commande du même robot. A cet effet, la première articulation est soumise à une loi CDFFA dont les paramètres sont donnés par la table (III.1). Les deux autres articulations sont contrôlées par la loi CDFAR sans lissage de paramètres regroupés dans la table (III.2). Les résultats obtenus sont montrés par la figure (III.14). Nous constatons une bonne poursuite des trajectoires avec des faibles erreurs qui s'annulent en régime permanent. Les oscillations de la première articulation ont disparu.

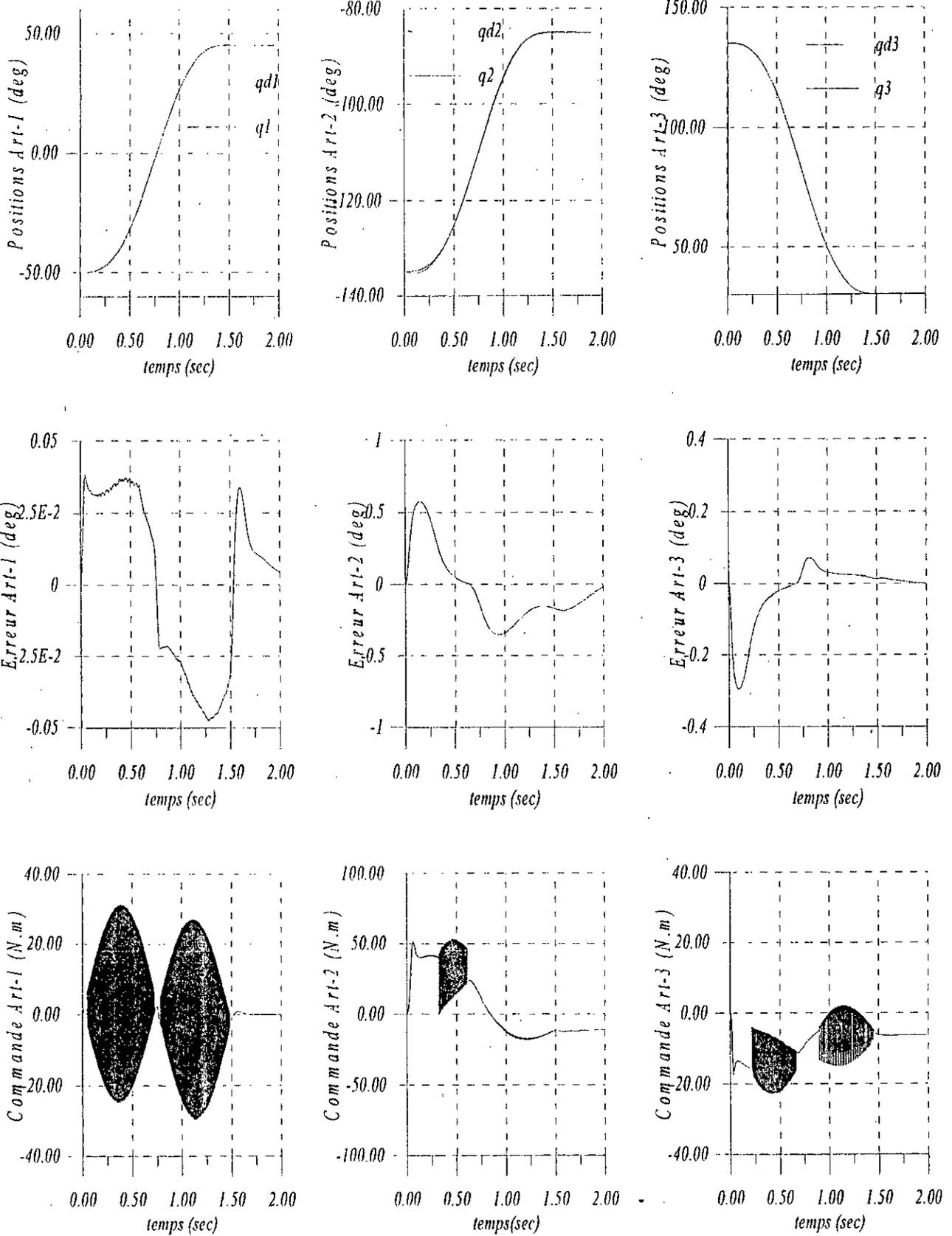
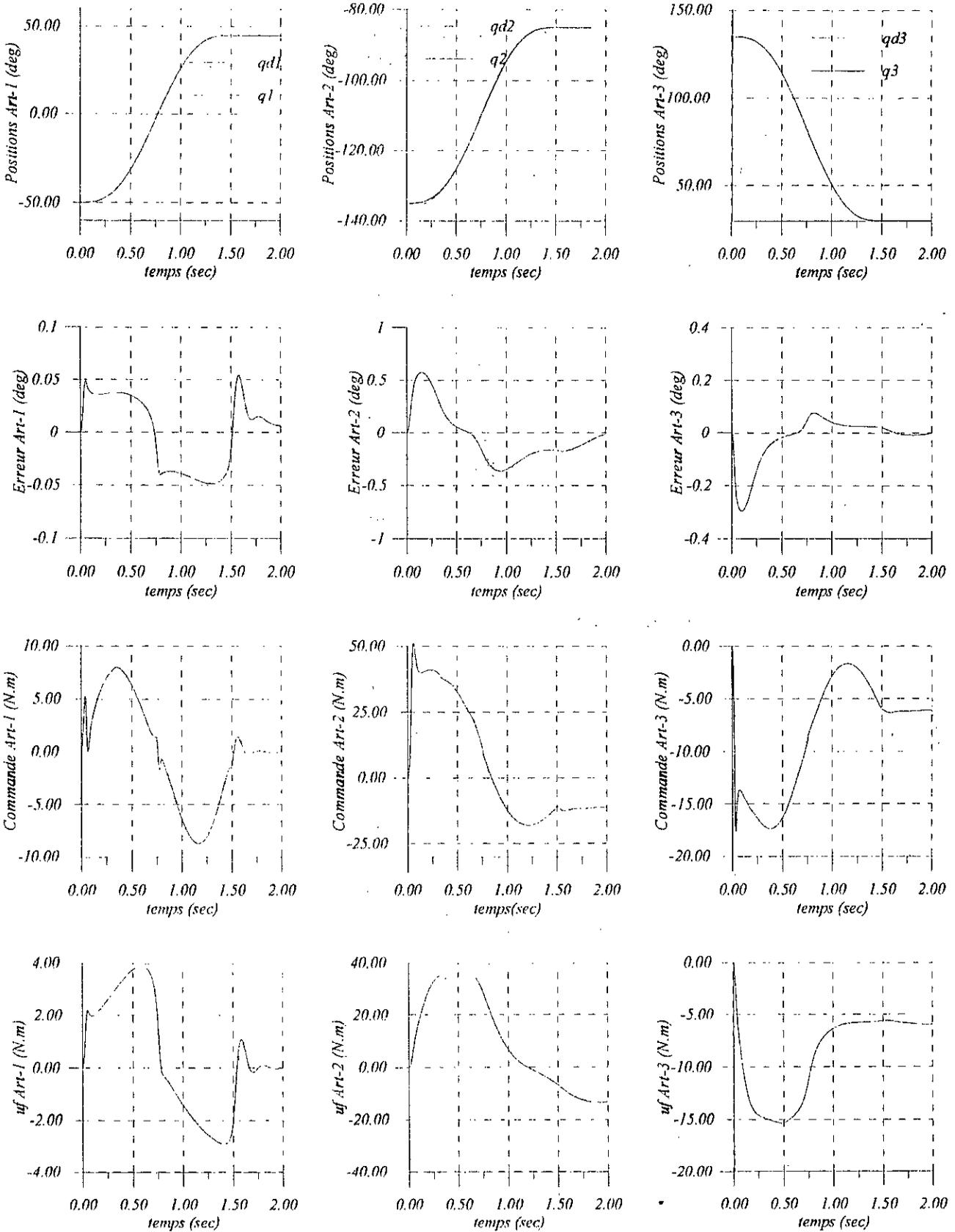


figure (III.9) : Réponses des différentes articulations soumises à la loi CDFAR :
Test sans charge



*figure (III.10a) : Réponses des différentes articulations soumises à la loi CDFAR :
Test sans charge avec lissage*

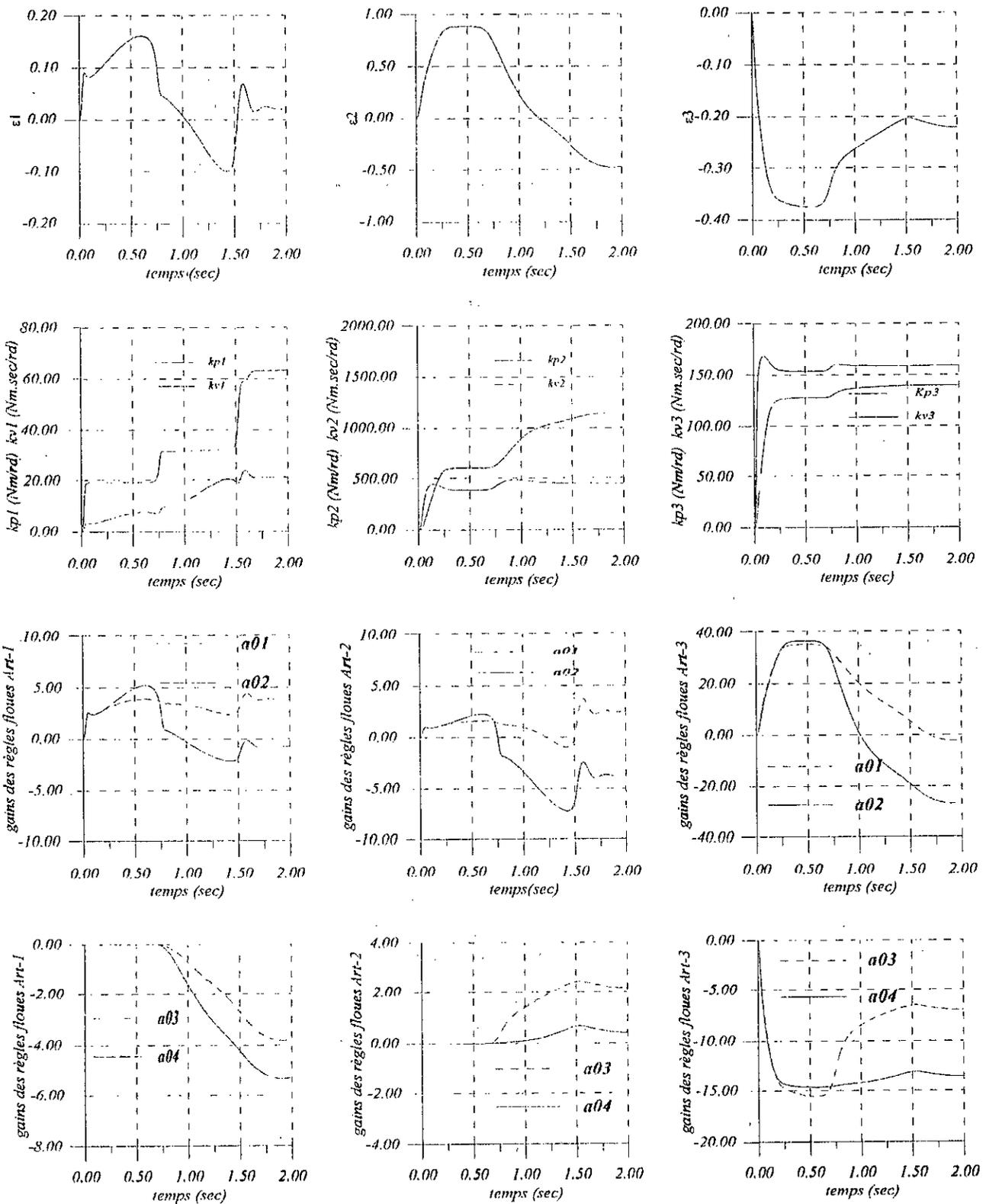
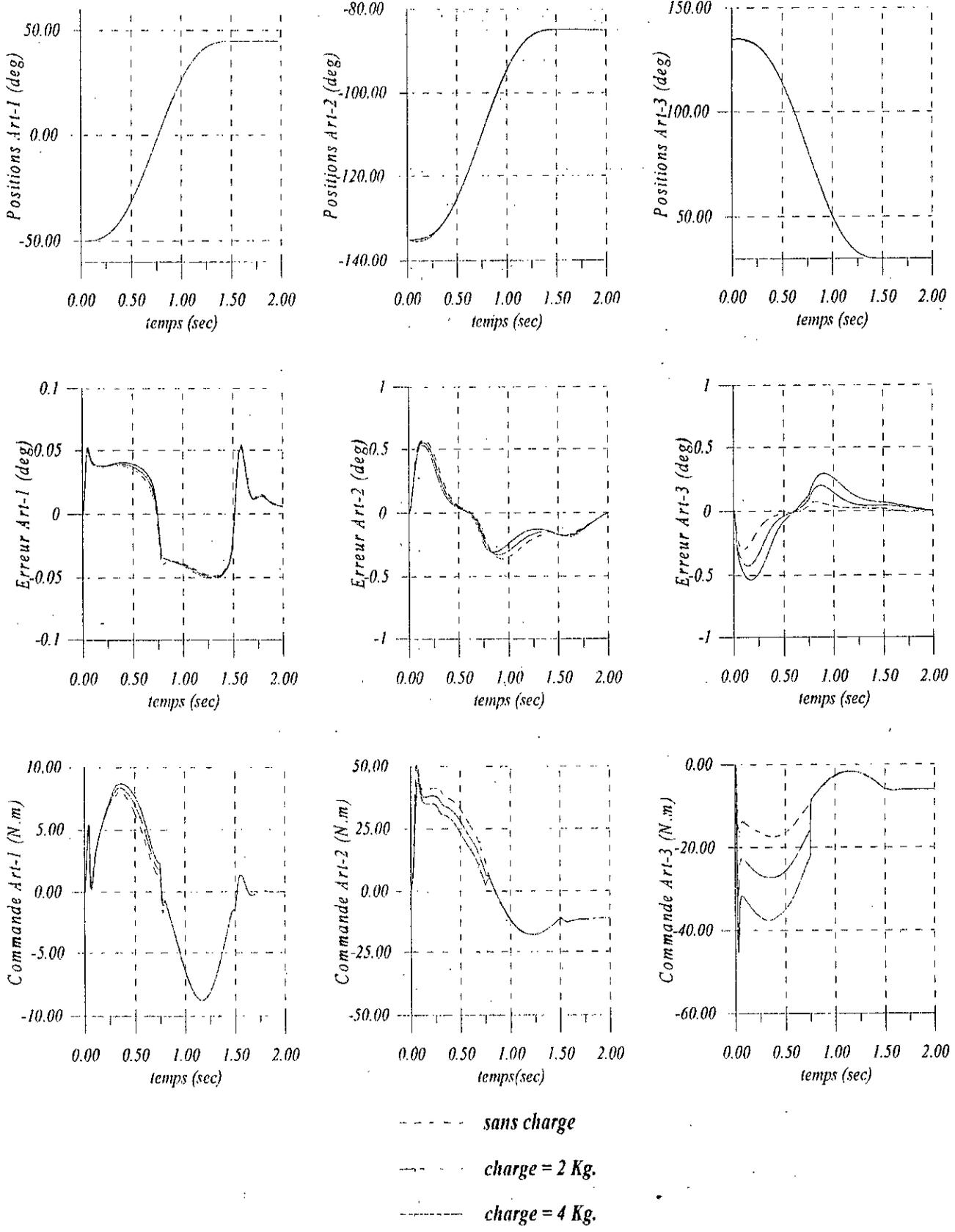


figure (III.10b) : Réponses des différentes articulations soumises à la loi CDFAR :
Test sans charge avec lissage



*figure (III.11) : Réponses des différentes articulations soumises à la loi CDFAR :
Test de variation de charge*

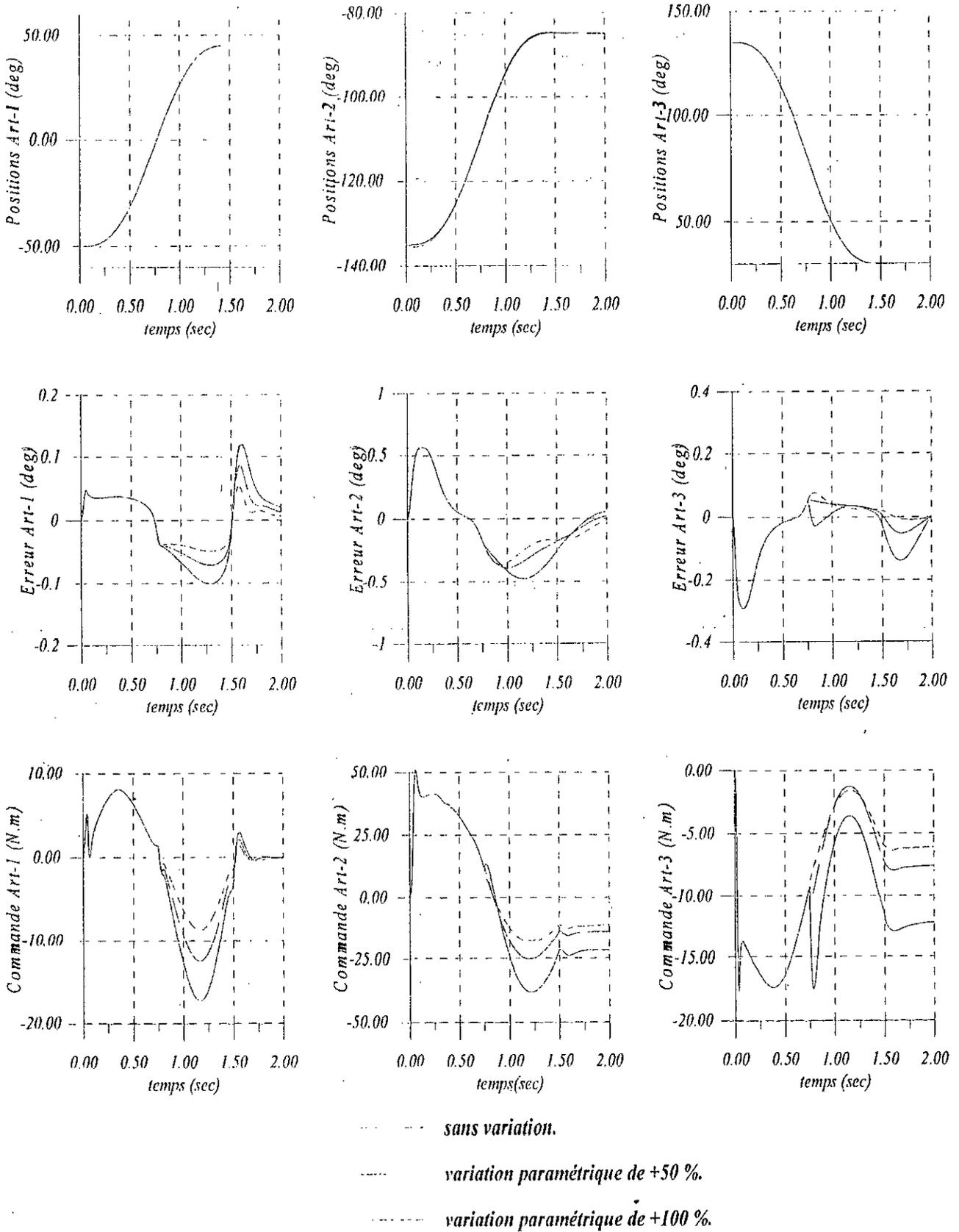
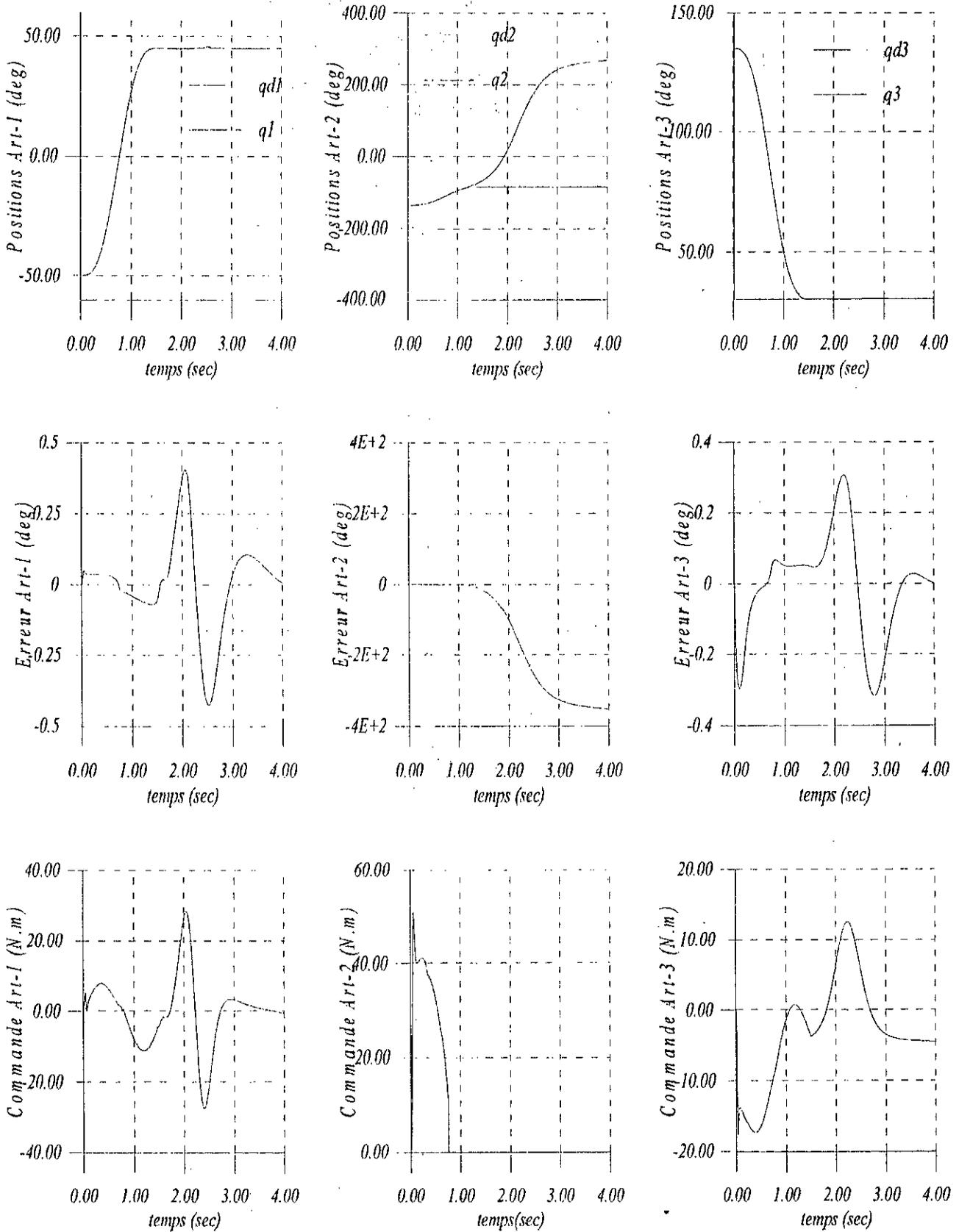
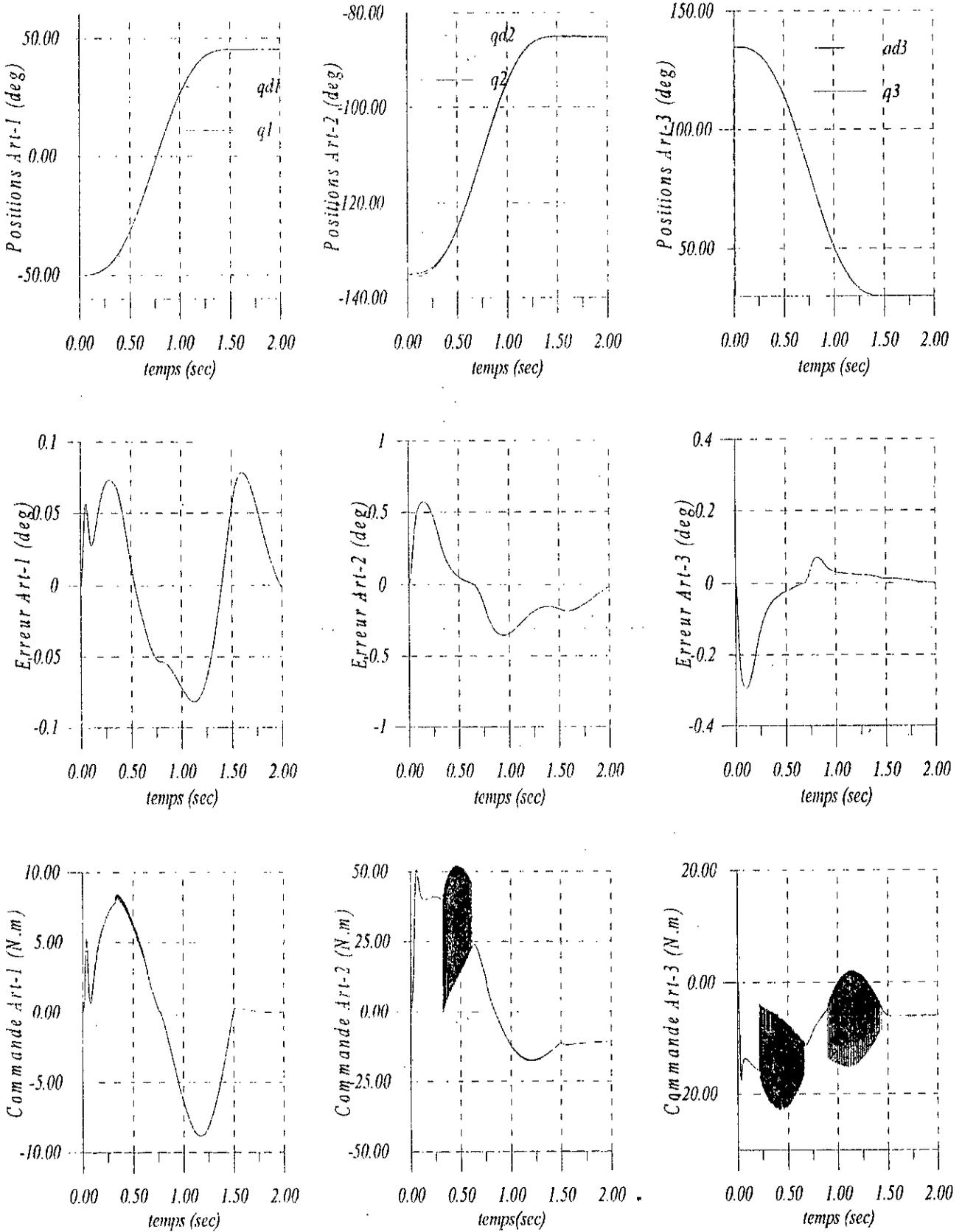


figure (III.12) : Réponses des différentes articulations soumises à la loi CDFAR :
Test de variation paramétrique



*figure (III.13) : Réponses des différentes articulations soumises à la loi CDFAR :
Test de rupture de commande*



*figure (III.14) : Réponses des différentes articulations:
Test d'association de la CDFA et la CDFAR*

III.3.3 Commande Décentralisée Floue Adaptative Modifiée (CDFAM).

III.3.3.1 Structure de commande

La loi de commande **CDFAR** développée dans la section précédente, bien qu'elle utilise un MFS à deux entrées seulement, elle présente deux inconvénients principaux : d'une part, la condition d'existence et de connaissance d'une borne supérieure des inerties $m_{ii}(q)$ et d'autre part la discontinuité de la commande. Pour développer une loi de commande sans ces deux inconvénients, nous proposons de remplacer le terme de robustesse $u_{si}(t)$ par un terme adaptatif $u_{mi}(t)$.

Proposons la loi de commande suivante :

$$u_i(t) = k_{pi}(t)e_i(t) + k_{vi}(t)\dot{e}_i(t) + Z_i^T \hat{A}_i(t)\xi^i + \hat{e}_i(t) + u_{mi}(t) \quad (\text{III.18a})$$

Choisissons le signal $u_{mi}(t)$ tel que :

$$u_{mi}(t) = k_{mi}(t)\ddot{q}_i^d \quad (\text{III.18b})$$

$$\dot{k}_{mi} = \gamma_{mi} r_i \ddot{q}_i^d + \gamma_{m2i} \frac{d}{dt}(r_i \ddot{q}_i^d), \quad \gamma_{mi} > 0, \gamma_{m2i} \geq 0 \quad (\text{III.18c})$$

Proposition 3 :

Etant donné un robot manipulateur à n degrés de liberté dont le modèle dynamique décentralisé de l'erreur de poursuite de chaque articulation peut être mis sous la forme (III.12).

Si on choisit la loi de commande (III.18a,b) et en tenant compte des hypothèses H1, H2 et H3 et les lois d'adaptations (III.10a,b,c,d), (III.18c) on aura :

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

R2 : Les paramètres $k_{pi}, k_{vi}, k_{mi}, \hat{A}_i$ et $\hat{e}_i, i = 1 \dots n$ et la commande $u_i(t), i = 1 \dots n$ sont bornées ;

R3 : Les erreurs de poursuite $e_i(t), i = 1 \dots n$ sont bornées et convergent, au moins asymptotiquement vers zéro.

Démonstration

La démonstration est basée sur une fonction de Lyapunov similaire à celle utilisée pour la démonstration de la Proposition 1 (Annexe B. Demo 1). Seulement, on a rajouté un terme supplémentaire nécessaire pour déduire la loi d'adaptation (III.18c) et pour assurer la bornitude des gains adaptatifs k_{mi} . Pour plus de détail voir Annexe B. Demo3.

Schéma de principe

La loi de commande décentralisée floue adaptative modifiée (**CDFAM**) donnée par (III.17a,b,c) et (III.10) est schématisée par la figure (III.15)

La loi de **CDFAM** garantissent la convergence asymptotique de l'erreur de poursuite tous en assurant, aussi, la bornitude des différents signaux. Cette loi de commande n'est plus discontinue et n'exige ni la connaissance ni même l'existence d'une borne supérieure des inerties $m_{ii}(q)$. Pour déterminer ces bornes il était nécessaire de prendre en considération les variations de charge et paramétriques. Cela n'est plus le cas avec la **CDFAM** car ces variations seront prises en charge via l'adaptation du gain $k_{mi}(t)$.

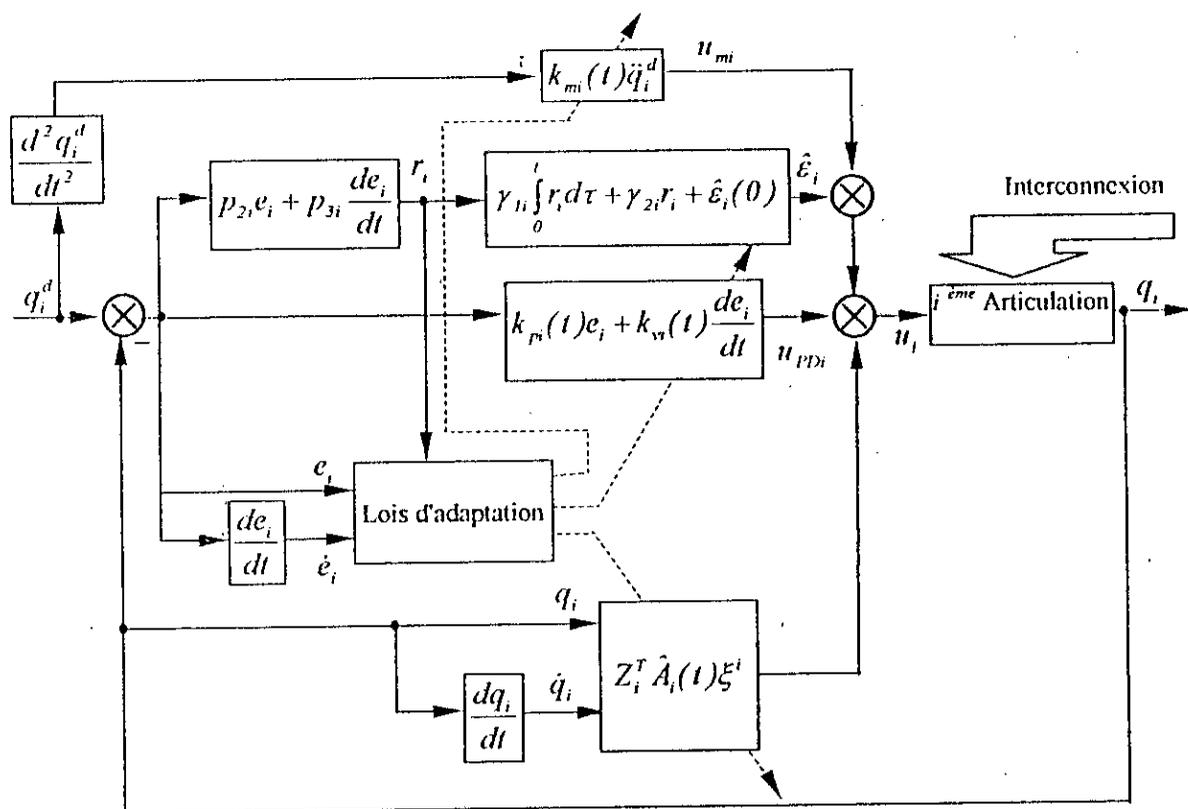


figure (III.15): Schéma de principe de la CDFAM.

III.3.3.2 Application au robot PUMA560

Dans cette section, nous présentons les résultats de l'application de la technique de CDFAM au cas du robot PUMA560 à trois degrés de liberté.

Paramètres du régulateur :

Chaque articulation est dotée d'un modèle flou de Sugeno d'ordre zéro (0). Deux fonctions d'appartenance triangulaires sont associées à chacune des variables d'entrée telles que montrées par la figure (III.2). Ainsi les modèles flous possèdent quatre règles.

Les paramètres de la loi de commande sont regroupés dans la table (III.3).

Avec I_{ij} est la matrice de dimensions $i \times j$ d'éléments tous égaux à 1.

Les valeurs initiales des différentes variables adaptatives de la loi de commande sont choisies nulles.

Tests et performances :

Test sans charge : Suite à un test sans charge nous avons obtenu les résultats montrés par les figures (IV.16a,b). Les trois articulations effectuent remarquablement la poursuite des trajectoires désirées.

Les erreurs sont inférieures à 0.05 deg et s'annulent en régime permanent. Les trois commandes prennent des variations lisses. Grâce à l'adoption d'une loi de type proportionnel-intégral, nous constatons une variation lisse des gains k_{mi} qui deviennent constante en régime permanent.

Test de variation de charge : Les réponses obtenues suite à l'introduction d'une charge de 2 Kg puis de 4 Kg est illustrée par la figure (III.17). Les erreurs de poursuite maximales sont inférieures à 0.05 deg, ce qui confirme la robustesse du régulateur.

	Symboles des paramètres	Première articulation	Deuxième articulation	Troisième articulation
Erreur Filtrée r_i	$p_{1i} ; p_{2i}$	3000 ; 300	3000 ; 300	3000 ; 300
Action PD Adaptative $k_{pi}e_i + k_{vi}e_i$	$\alpha_{1i} ; \alpha_{2i}$ $\beta_{1i} ; \beta_{2i}$	930 ; 9.3 930 ; 9.3	930 ; 9.3 930 ; 9.3	930 ; 9.3 930 ; 9.3
Facteur de normalisation	$q_i ; \dot{q}_i$	4 ; 5.33	4 ; 5.33	4 ; 5.33
Modèle flou $Z_i^T \hat{A}_i(t) \xi^i$	$\Gamma_{1i} ; \Gamma_{2i}$	$300 \times 1_{1 \times 4} ; 0 \times 1_{1 \times 4}$	$300 \times 1_{1 \times 4} ; 0 \times 1_{1 \times 4}$	$300 \times 1_{1 \times 4} ; 0 \times 1_{1 \times 4}$
Signal auxiliaire	$\gamma_{1i} ; \gamma_{2i}$	1 ; 0	1 ; 0	1 ; 0
Terme de robustesse	$\gamma_{m1i} ; \gamma_{m2i}$	0.001 ; 0.0001	0.01 ; 0.001	0.01 ; 0.001

Table (III.3) : Paramètres du régulateur sous la loi CDFAM

Test de variation paramétrique : Une variation sur tous les paramètres du modèle du robot de +50% puis de +100 % est introduite à l'instant $t=0.75$ sec. La figure (III.18) montre les réponses obtenues. Nous constatons la robustesse de notre régulateur vis-à-vis des variations appliquées.

Test de rupture de commande : Dans ce test nous appliquons une commande nulle à de la deuxième articulation à partir de l'instant $t=0.75$ (chute libre). D'après les résultats obtenues, montrés par la figure (III.19), nous constatons la robustesse des régulateurs locaux associés aux deux autres articulations.

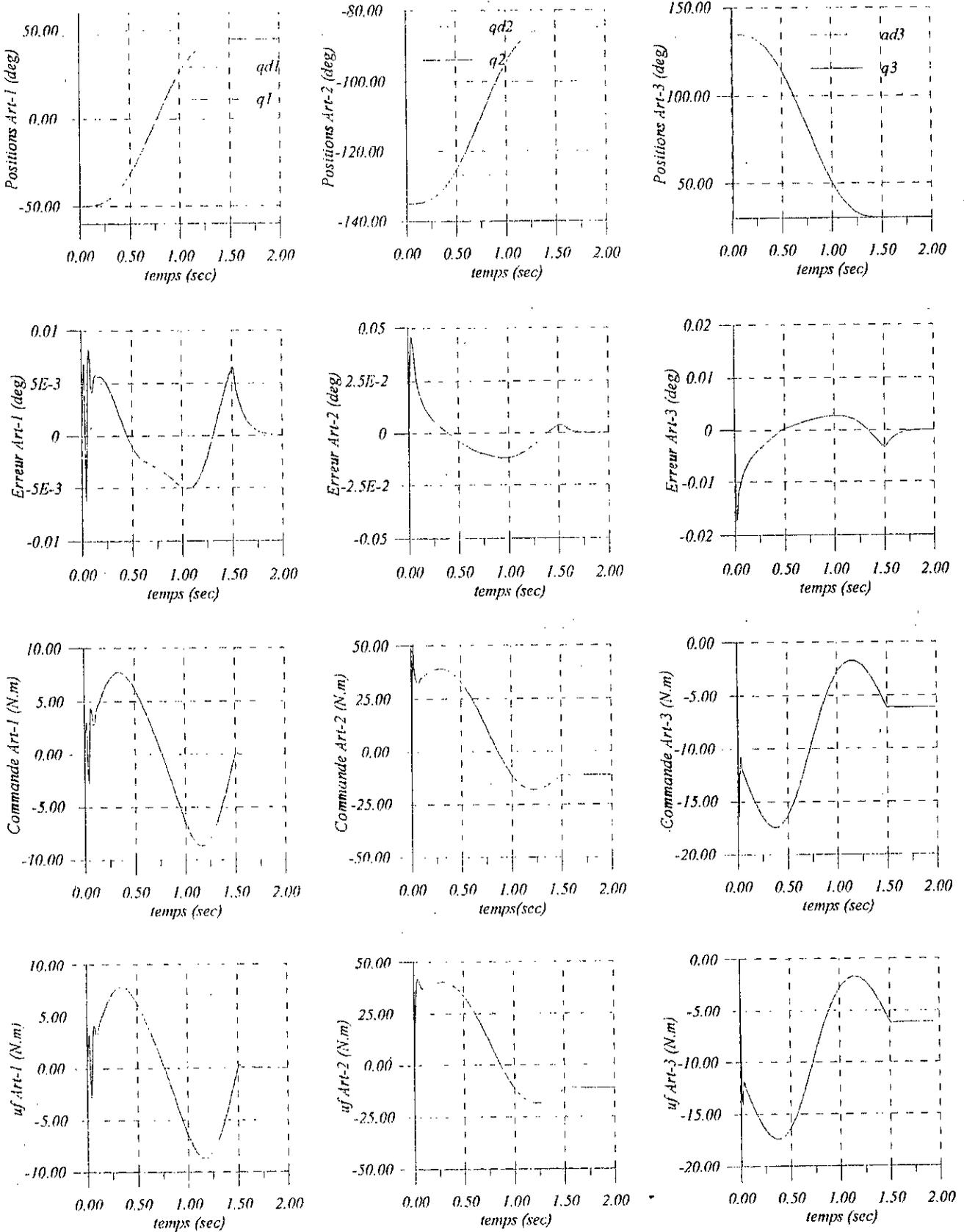


figure (III.16a) : Réponses des différentes articulations soumises à la loi CDFAM
 Test sans charge :

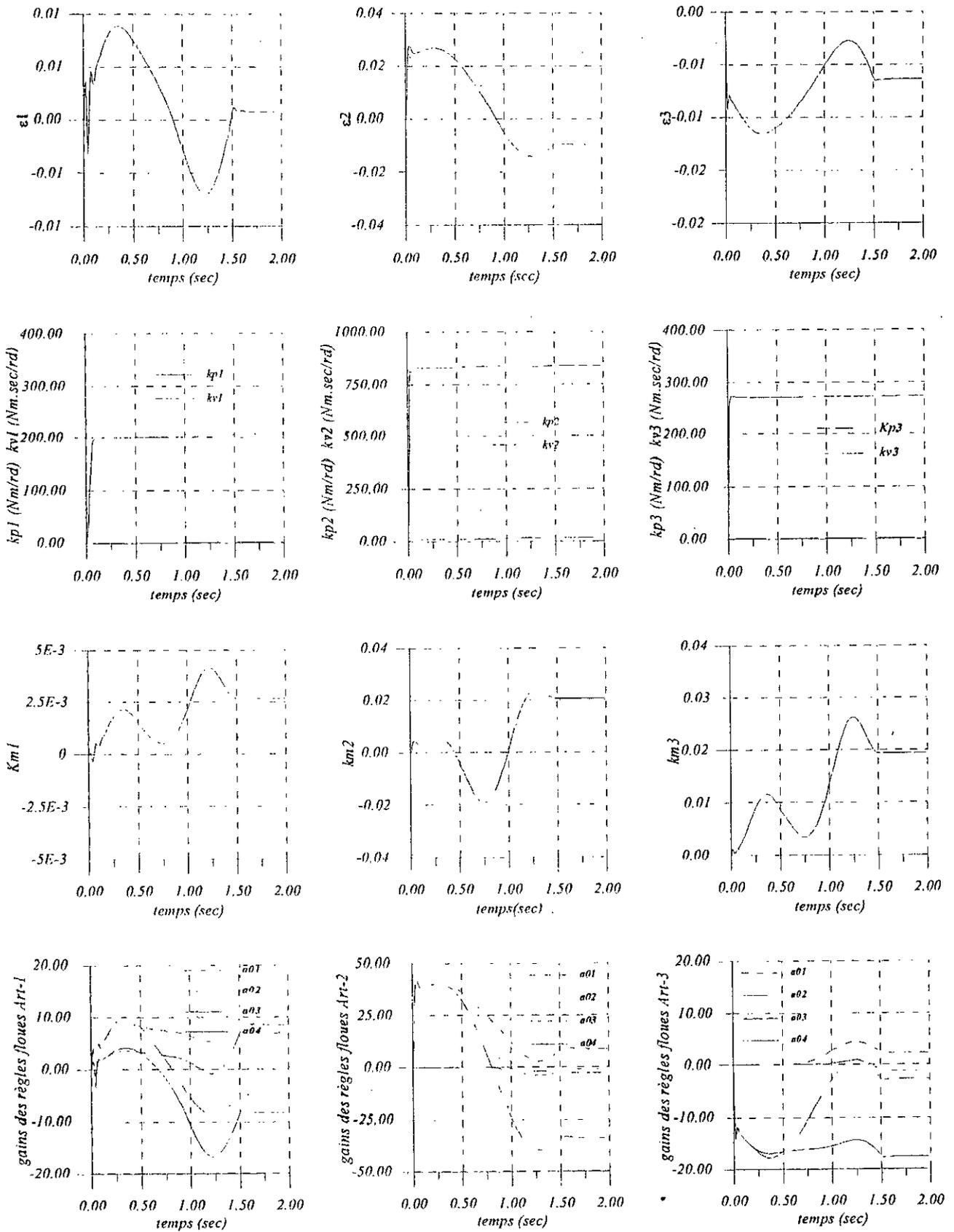


figure (III.16b) : Réponses des différentes articulations soumises à la loi CDFAM
Test sans charge

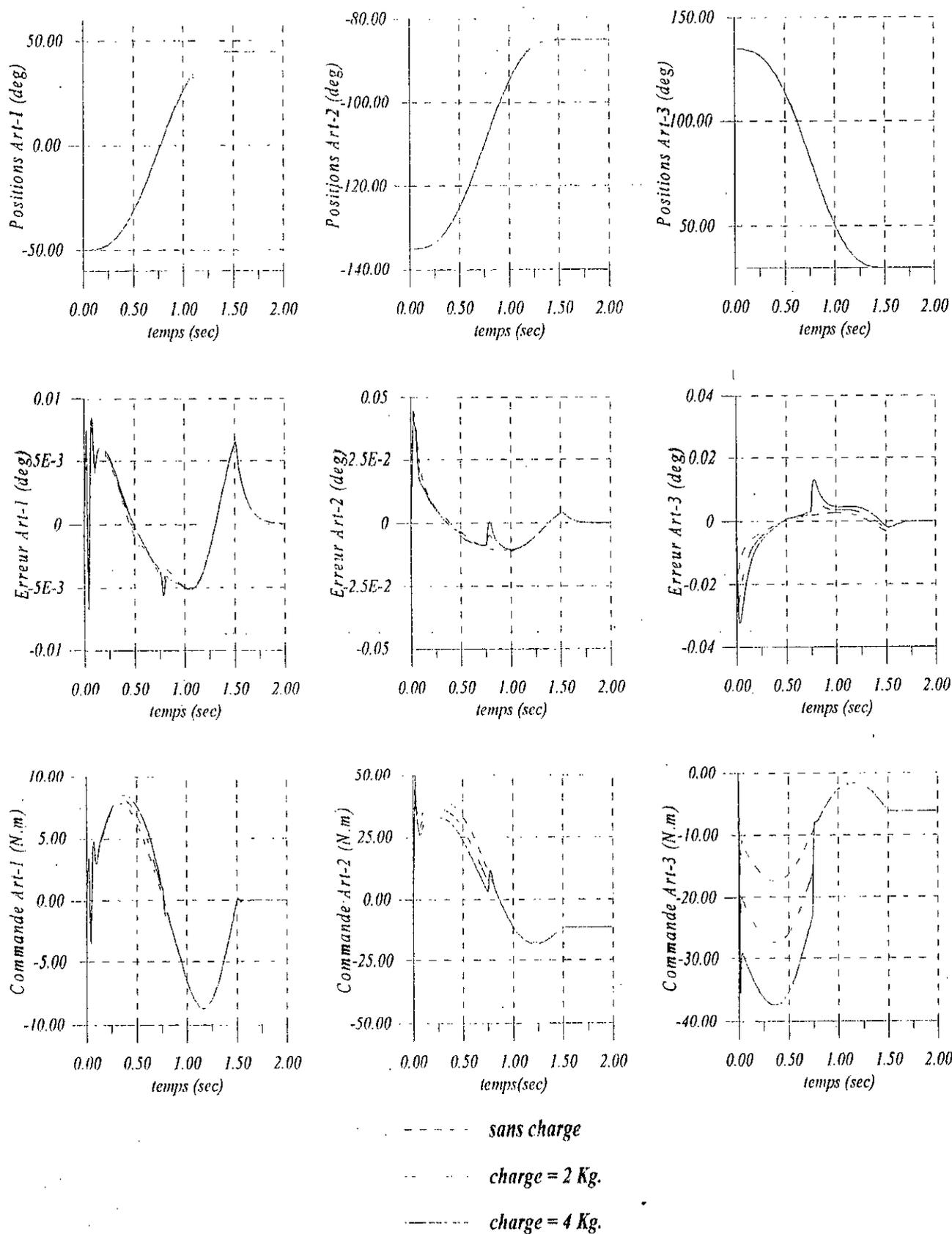


figure (III.17) : Réponses des différentes articulations soumises à la loi CDFAM
Test de variation de charge

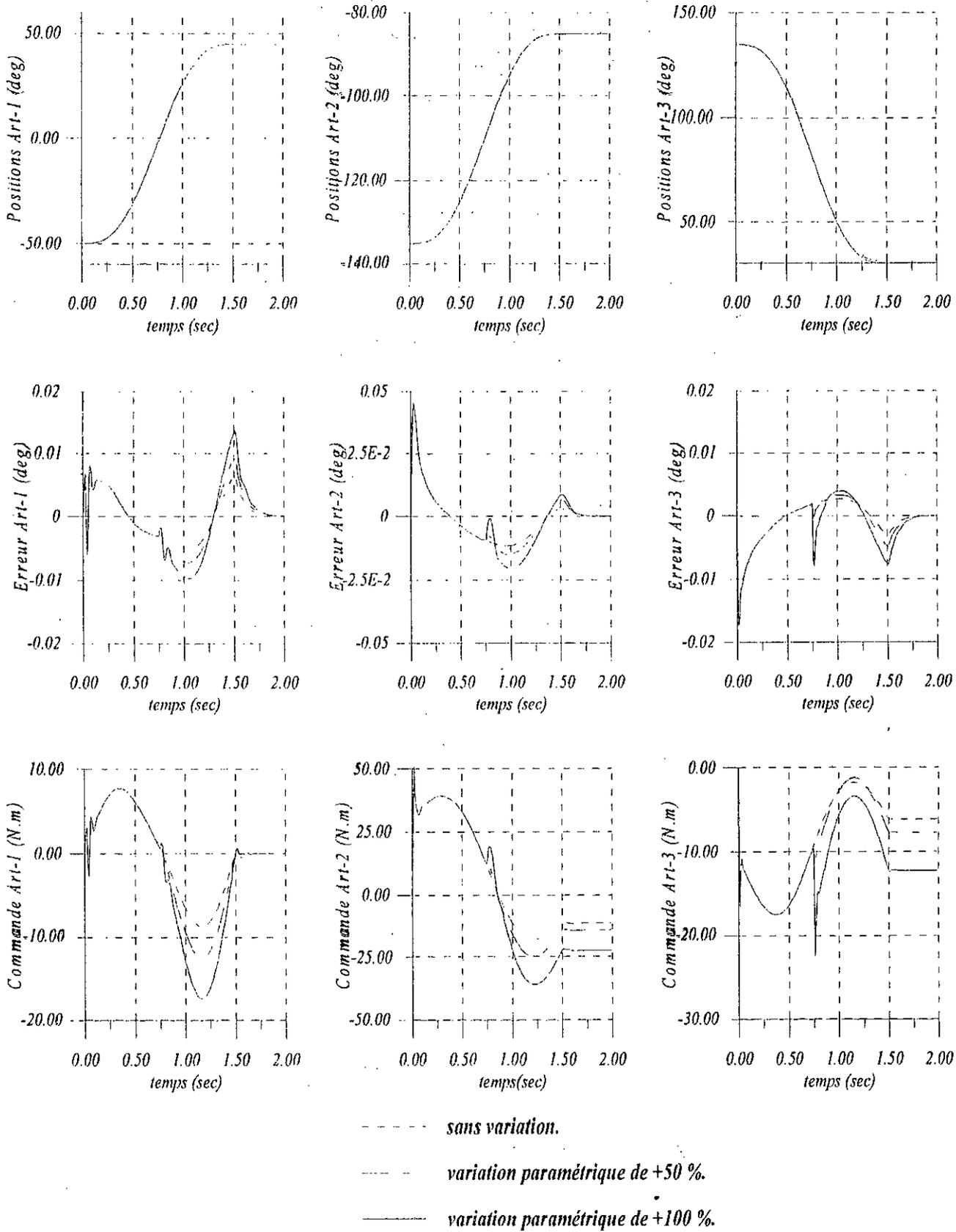


figure (III.15) : Réponses des différentes articulations soumises à la loi CDFAM
 Test de variation paramétrique

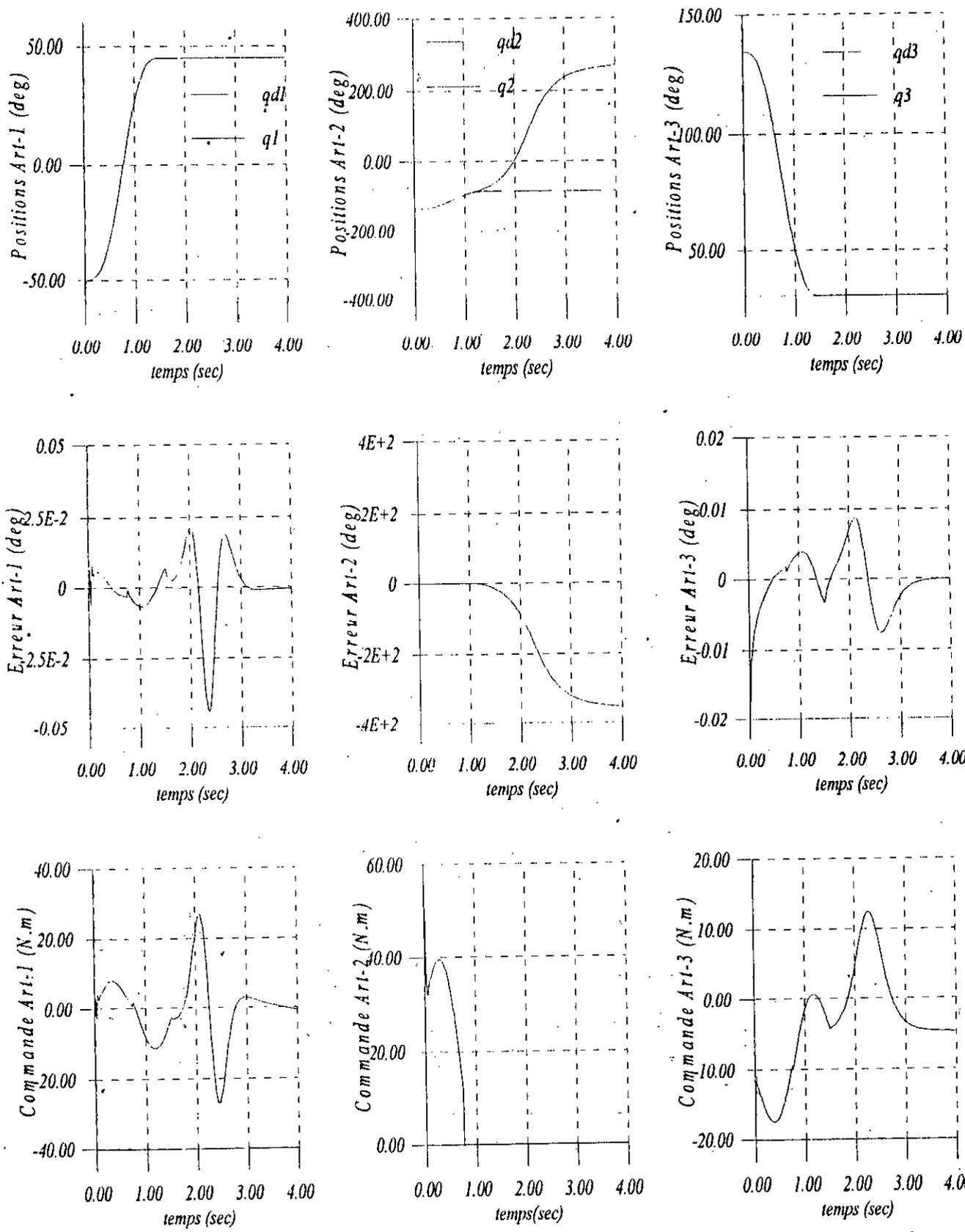


figure (III.19) : Réponses des différentes articulations soumises à la loi CDFAM
 Test de rupture de commande

III.4 Deuxième approche

Dans la première approche, nous avons développé des lois de commande décentralisée floue (CDFA, CDFAR et CDFAM) où la stabilité asymptotique de l'erreur ainsi que la bornitude des différents signaux sont assurés. Dans la structure intervient une action proportionnelle (PD) adaptative. Une hypothèse principale consiste à considérer que les inerties $m_n(q)$ sont lentement variables dans le temps.

Dans la deuxième approche, en basant toujours sur l'approche de l'erreur filtrée [LEW93] [SER89], nous proposons des lois de commande où interviennent trois actions principales :

- une action PD non adaptative ;
- un modèle flou de Sugeno (MFS) adaptatif pour compenser les termes d'interconnexions;
- signal auxiliaire pour compenser l'erreur de reconstruction.

Sans que l'hypothèse sur la variation des inerties $m_n(q)$ soit nécessaire, la convergence de l'erreur est assurée. Cependant la mesure de l'accélération réelle doit être disponible.

III.4.1 Commande Décentralisée Floue Adaptative Simple (CDFAS)

III.4.1.1 Structure de commande

Considérons le modèle dynamique de la $i^{ème}$ articulation donnée par (III.4).

$$\dot{m}_n(q)\ddot{q}_i = -c_i(q, \dot{q}, \ddot{q}) + u_i(t)$$

On sait que l'inertie $m_n(q)$ est toujours positive, donc on peut écrire :

$$m_n(q) = m_{0i} + \Delta m_n(q) \tag{III.19}$$

avec, m_{0i} constante positive connue.

Donc on peut écrire :

$$m_{0i}\ddot{e}_i = c_i(q, \dot{q}, \ddot{q}) + \Delta m_n(q)\ddot{q}_i + m_{0i}\ddot{q}_i^d - u_i(t) \tag{III.20}$$

$$m_{0i}\ddot{e}_i = d_i(q, \dot{q}, \ddot{q}) + m_{0i}\ddot{q}_i^d - u_i(t) \tag{III.21a}$$

$$d_i(q, \dot{q}, \ddot{q}) = c_i(q, \dot{q}, \ddot{q}) + \Delta m_n(q)\ddot{q}_i \tag{III.21b}$$

avec,

d'après l'équation (III.20b), le terme de couplage d_i dépend de tous les états locaux q_i, \dot{q}_i et \ddot{q}_i . Comme dans la première approche, nous supposons que le terme d'interconnexion d_i peut être exprimé sous la forme de la sortie d'un MFS ayant trois entrées q_i, \dot{q}_i et \ddot{q}_i . D'où, on écrit :

$$d_i(q, \dot{q}, \ddot{q}, \ddot{q}_i) = f_i(q_i, \dot{q}_i, \ddot{q}_i) + \varepsilon_i(t) \tag{III.22a}$$

où,

$$f_i(q_i, \dot{q}_i, \ddot{q}_i) = Z_i^T A_i^* \xi^i + \varepsilon_{f_i} \tag{III.22b}$$

Les expressions des variables liées au MFS (Z_i^T, A_i^* et ξ^{iT}) sont données par les équations (III.7a,b,c,d,e) où on remplace la variable \ddot{q}_i^d par \ddot{q}_i . Les erreurs $\varepsilon_i, \varepsilon_{f_i}$ sont respectivement, l'erreur d'approximation de d_i et l'erreur de reconstruction de f_i .

En remplaçant (III.22a, b) dans (III.21a), on obtient le modèle dynamique suivant :

$$m_{0i}\ddot{e}_i = Z_i^T A_i^* \xi^i + (\varepsilon_i(t) + \varepsilon_{f_i}) + m_{0i}\ddot{q}_i^d - u_i(t) \tag{III.22c}$$

D'après ce modèle, nous proposons la loi de commande :

$$u_i(t) = u_{PDi} + Z_i^T \hat{A}_i \xi^i + \hat{\varepsilon}_i(t) + m_{0i}\ddot{q}_i^d \tag{III.22d}$$

où : u_{PDi} : action Proportionnelle-Dérivée de stabilisation

$Z_i^T A_i^* \xi^i$: estimée de la fonction f_i ;

ε_{f_i} : signal auxiliaire.

En remplaçant la loi de commande (III.24) dans (III.23), on obtient le modèle dynamique de l'erreur en boucle fermée :

$$m_{0i}\ddot{e}_i = -u_{FDi} + Z_i^T \tilde{A}_i \xi^i + \tilde{e}_i(t) \tag{III.25}$$

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

$\tilde{e}_i = \varepsilon_{fi} + \varepsilon_i - \hat{\varepsilon}_i$: erreur entre erreur de reconstruction $\varepsilon_{fi} + \varepsilon_i$ et le signal auxiliaire $\hat{\varepsilon}_i$.

Définition

Soit \mathfrak{I} la classe de matrices carrées non-singulières, de dimension $l \geq 1$, définies par :

$$\mathfrak{I} = \{ \Gamma(t) \in \mathbb{R}^{l \times l}, l \geq 1 \}$$

avec,

D1: $\Gamma^{-1}(t) > 0$ matrice définie positive (pour $l > 1$), ou scalaire strictement positif (pour $l = 1$).

D2: $\Gamma^{-1}(t) \leq 0$ matrice semi-définie négative et bornée (pour $l > 1$), ou scalaire négative bornée ($l = 1$).

D3: $\Gamma^{-1}(t) \leq 0$ matrice d'éléments bornés (pour $l > 1$), ou scalaire borné (pour $l = 1$).

Cas particulier de la classe \mathfrak{I}

On montre dans l'Annexe C. que l'ensemble des matrices P définies par :

$$\dot{P} = \lambda P - \lambda P W_p P \tag{III.26}$$

où, λ : une constante positive ($\lambda \in \mathbb{R}_+$);

W_p : une matrice constante ($W_p \in \mathbb{R}^{l \times l}$);

P : une matrice à éléments variables ($P \in \mathbb{R}^{l \times l}$).

constitue un sous-ensemble de la classe \mathfrak{I} si :

♦ $P^{-1}(t=0)$: matrice définie positive lorsque $\lambda = 0$;

♦ $P^{-1}(t=0) - W_p$: matrice définie positive lorsque $\lambda \neq 0$;

Choisissons u_{FDi} et les lois d'adaptation suivantes :

$$u_{FDi} = \left(\frac{k_{ri} k_{0i}}{k_{1i}} \right) e_i + \left(\frac{m_{0i} k_{0i} + k_{ri} k_{1i}}{k_{1i}} \right) \dot{e}_i \tag{III.27}$$

$$\begin{cases} \dot{\hat{A}}_i = \Gamma_i(t) Z_i \xi^i r_i & \Gamma_i \in \mathfrak{I} \\ \dot{\hat{\varepsilon}}_i = \gamma_i(t) r_i & \gamma_i \in \mathfrak{I} \\ r_i = k_{0i} e_i + k_{1i} \dot{e}_i & k_{0i}, k_{1i}, k_{ri} > 0 \end{cases} \tag{III.28}$$

Proposition 4 :

Etant donné un robot manipulateur à n degrés de liberté dont le modèle dynamique de l'erreur de poursuite de l' $i^{ème}$ articulation $i=1 \dots n$ peut être décrit par l'équation (III.23).

Si on choisit la loi de commande (III.24), (III.27) et en tenant compte des hypothèses H1, H3 et les lois d'adaptation (III.28) on aura :

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

R2 : Les variables \hat{A}_i et $\hat{\varepsilon}_i, i=1 \dots n$ et la commande $u_i(t)$ sont bornées ;

R3 : Les erreurs de poursuite $e_i(t), i=1 \dots n$ sont bornées et convergent, au moins, asymptotiquement vers zéro.

Démonstration :

La démonstration de la Proposition est basée sur la théorie de stabilité de Lyapunov en utilisant aussi le lemme de Barbalat. Ce lemme stipule que si une fonction (de Lyapunov) V vérifie les trois conditions suivantes [SLO91] :

C1 : La fonction de Lyapunov V possède une borne inférieure (inférieurement bornée);

C2 : $\dot{V} \leq 0$;

C3 : \dot{V} est uniformément continue;

alors,

R : \dot{V} converge asymptotiquement vers zéro.

Dès que la démonstration est basée sur le modèle dynamique de chaque articulation indépendamment nous omettons l'indice de l'articulation pour ne pas encombrer l'écriture.

Considérons la fonction de Lyapunov candidate suivante :

$$V_l = \frac{1}{2} m_0 r^2 + \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \tilde{A}) + \frac{1}{2} \gamma_0^{-1} \tilde{\varepsilon}^2 \quad (\text{III.29})$$

où,

- $\text{Tr}(\cdot)$: désigne la trace de la matrice (\cdot);
- $\Gamma_0, \gamma_0 \in \mathfrak{S}$;
- r : erreur filtrée (ou pondérée) définie par :

$$r = k_0 e + k_1 \dot{e} \quad k_0, k_1 > 0 \quad (\text{III.30})$$

D'après la définition de V_l et la propriété D1 de la classe \mathfrak{S} , on a $V_l \geq 0$, d'où la condition C1 est vérifiée. En dérivant la fonction V_l par rapport au temps, on obtient :

$$\dot{V}_l = r(m_0 k_0 \dot{e} + m_0 k_1 \ddot{e}) + \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{\tilde{A}}) + \gamma_0^{-1} \tilde{\varepsilon} \dot{\tilde{\varepsilon}} + \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{\tilde{A}}) + \frac{1}{2} \gamma_0^{-1} \dot{\tilde{\varepsilon}}^2 \quad (\text{III.31})$$

d'après la définition de \tilde{A} et $\tilde{\varepsilon}$, on a :

$$\dot{\tilde{A}} = -\dot{\hat{A}} \quad (\text{car la matrice } A^* \text{ est constante})$$

$$\dot{\tilde{\varepsilon}} = -\dot{\hat{\varepsilon}} \quad (\text{d'après l'hypothèse H3}).$$

Si en plus, on remplace (III.25) dans (III.31), on obtient :

$$\dot{V}_l = r(m_0 k_0 \dot{e} + k_1 (-u_{PD} + Z^T \tilde{A} \xi + \tilde{\varepsilon})) - \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{\hat{A}}) - \gamma_0^{-1} \tilde{\varepsilon} \dot{\hat{\varepsilon}} + \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{\tilde{A}}) + \frac{1}{2} \gamma_0^{-1} \dot{\tilde{\varepsilon}}^2$$

donc,

$$\dot{V}_l = \dot{V}_{11} + \dot{V}_{12} + \dot{V}_{13} + \dot{V}_{14} \quad (\text{III.32a})$$

où,

$$\dot{V}_{11} = r(m_0 k_0 \dot{e} - k_1 u_{PD}) \quad (\text{III.32b})$$

$$\dot{V}_{12} = r k_1 Z^T \tilde{A} \xi - \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{\hat{A}}) \quad (\text{III.32c})$$

$$\dot{V}_{13} = r k_1 \tilde{\varepsilon} - \gamma_0^{-1} \tilde{\varepsilon} \dot{\hat{\varepsilon}} \quad (\text{III.32d})$$

$$\dot{V}_{14} = \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{\tilde{A}}) + \frac{1}{2} \gamma_0^{-1} \dot{\tilde{\varepsilon}}^2 \quad (\text{III.32e})$$

Pour avoir $\dot{V}_l \leq 0$:

$$1. \text{ On choisit } u_{PD} = \frac{1}{k_1} (m_0 k_0 \dot{e} + k_1 r) \quad (\text{III.33})$$

$$\text{donc, } u_{PD}(t) = \left(\frac{k_r k_0}{k_1} \right) e + \left(\frac{m_0 k_0 + k_r k_1}{k_1} \right) \dot{e}$$

On obtient : $\dot{V}_{11} = -k_r r^2 \quad k_r > 0$

2. On utilise les deux relations matricielles suivantes [SPO95a] :

$$\begin{cases} \text{Tr}(KL) = \text{Tr}(LK) \\ \text{Tr}(KL) = \text{Tr}(L^T K^T) \end{cases}$$

avec K, L deux matrices de dimensions convenables, pour écrire :

$$Z^T \tilde{A} \xi = \text{Tr}(Z^T \tilde{A} \xi) = \text{Tr}(\xi^T \tilde{A}^T Z) = \text{Tr}(\tilde{A}^T Z \xi^T)$$

donc \dot{V}_{12} devient :

$$\dot{V}_{12} = \text{Tr}(\tilde{A}^T (rk_1 Z \xi^T - \Gamma_0^{-1} \dot{\hat{A}}))$$

si nous choisissons

$$\dot{\hat{A}} = \Gamma Z \xi^T r, \quad \Gamma = k_1 \Gamma_0 \in \mathfrak{S} \quad (\text{III.34})$$

on aura : $\dot{V}_{12} = 0$

3. On choisit

$$\dot{\hat{\varepsilon}} = \gamma \cdot r, \quad \gamma = k_1 \gamma_0 \in \mathfrak{S} \quad (\text{III.35})$$

alors $\dot{V}_{13} = 0$.

4. D'autre part, selon la propriété D2 de la classe \mathfrak{S} , on a :

$$\dot{V}_{14} = \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \tilde{A}) + \frac{1}{2} \gamma_0^{-1} \tilde{\varepsilon}^2 \leq 0 \quad (\text{III.36})$$

On obtient :

$$\dot{V}_1 = -k_r r^2 + \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \tilde{A}) + \frac{1}{2} \gamma_0^{-1} \tilde{\varepsilon}^2 \leq 0 \quad (\text{III.37})$$

donc la deuxième condition (C2) du lemme est vérifiée.

Pour montrer que \dot{V} est uniformément continue, il suffit de montrer que sa dérivé \ddot{V} est bornée [SLO91], où :

$$\ddot{V}_1 = -2k_r r \cdot \dot{r} - \text{Tr}(\tilde{A}^T \dot{\Gamma}_0^{-1} \dot{\hat{A}}) - \gamma_0^{-1} \tilde{\varepsilon} \dot{\hat{\varepsilon}} + \frac{1}{2} \text{Tr}(\tilde{A}^T \ddot{\Gamma}_0^{-1} \tilde{A}) + \frac{1}{2} \gamma_0^{-1} \ddot{\varepsilon}^2 \quad (\text{III.38})$$

D'après la théorie de stabilité de Lyapunov les inégalités $V_1 \geq 0$, $\dot{V}_1 \leq 0$ assurent que les variables r , \tilde{A} et $\tilde{\varepsilon}$ sont bornées [SLO91]. Alors d'après (III.34) (III.35) les paramètres $\dot{\hat{A}}$ et $\dot{\hat{\varepsilon}}$ sont aussi bornés.

D'autre part, les dérivées $\dot{\Gamma}_0^{-1}$ et $\dot{\gamma}_0^{-1} \Gamma_0^{-1}$ et $\dot{\gamma}_0^{-1}$ sont bornées (propriétés D2 et D3).

Si on montre que \dot{r} est aussi bornée, on peut conclure que \dot{V}_1 est uniformément continue.

En effet, $\dot{r} = k_0 \dot{e} + k_1 \ddot{e}$

L'expression de l'erreur filtrée r peut être considérée comme l'équation d'un filtre stable (car $k_0, k_1 > 0$) dont l'état est l'erreur (e) et l'entrée est l'erreur filtrée (r). Cette dernière étant bornée, alors on en déduit que e et \dot{e} , et d'après l'hypothèse H1, q et \dot{q} sont aussi bornées (définition de la stabilité). En utilisant les équations (III.25) (III.27), la bornitude des variables e , \dot{e} , \tilde{A} et $\tilde{\varepsilon}$ garantie celle de \ddot{e} , et d'après l'hypothèse H1 l'accélération \ddot{q} est aussi bornée.

Vu que e et \ddot{e} sont bornées alors la fonction \dot{V} est uniformément continue ainsi la troisième condition du lemme est vérifiés. Donc, \dot{V}_1 et r converge vers zéro.

Puisque $k_0, k_1 > 0$, on conclut que e et \dot{e} converge asymptotiquement vers zéro.

D'après cette analyse on a :

- q, \dot{q} et \ddot{q} sont bornées (résultat R1)
- e et \dot{e} convergent asymptotiquement vers zéro (résultat R2).

Pour montrer la bornitude de \hat{A} et $\hat{\varepsilon}$ on a :

$\Rightarrow \begin{cases} \hat{A} \text{ est bornée car } A^* \text{ est une matrice contante;} \\ \hat{\varepsilon} \text{ est bornée car } \varepsilon_f \text{ et } \varepsilon \text{ sont bornées puisqu'ils dépendent d'états dont on a montré la bornitude.} \end{cases}$

D'après la définition $u(t)$ est bornée car e, \dot{e}, \hat{A} et $\hat{\varepsilon}$ sont bornées (résultat R3).

La démonstration est la même pour toutes les articulations.

Schéma de principe :

Le schéma de principe de la commande décentralisée floue adaptative simple (CDFAS) donnée par (III.24), (III.27) et (III.28) est montré par la figure (III.20).

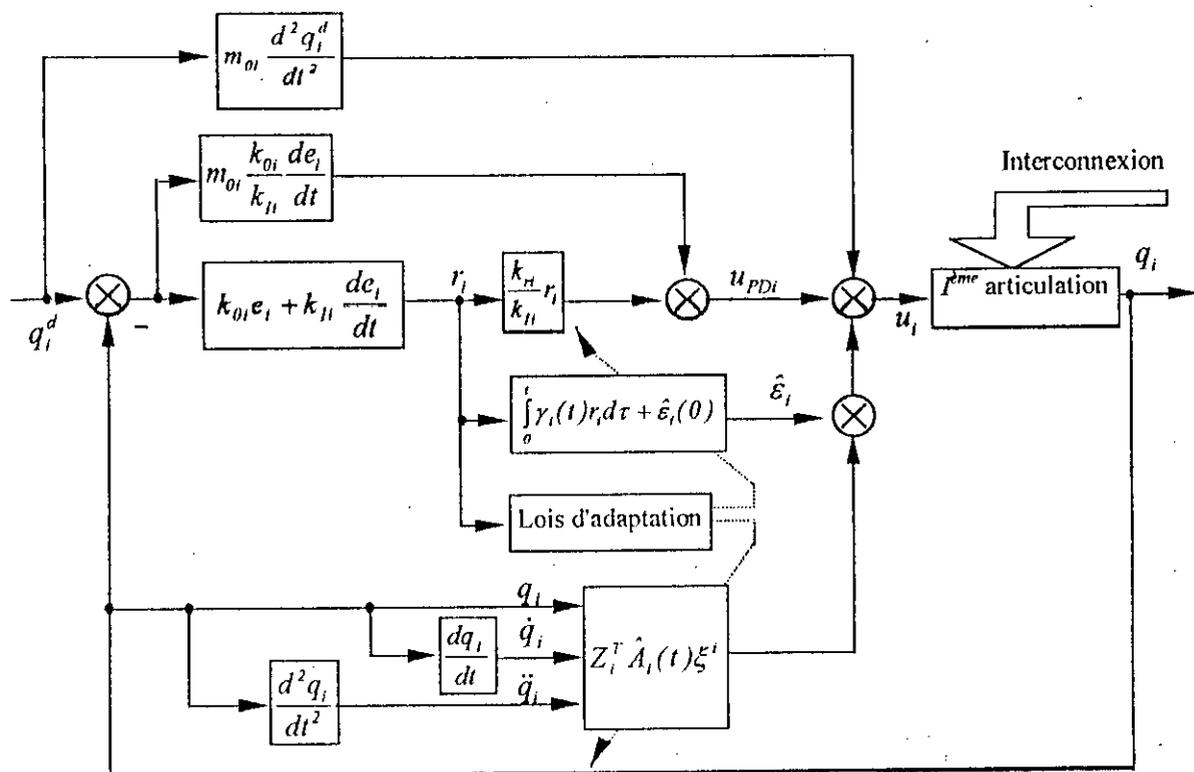


figure (III.20) : Schéma de principe de la CDFAS.

Sous les hypothèses H1, H3, la Proposition précédente, révèle que l'utilisation de la CDFAS assure la bornitude et la convergence asymptotique des erreurs de poursuite des différentes articulations. La bornitude des commande $u_i(t)$ ainsi que des variables $q_i, \dot{q}_i, \ddot{q}_i, \hat{A}_i$ et $\hat{\varepsilon}_i$ est aussi garantie. L'hypothèse exigeant une variation lente des moments d'inerties ($m_{ii}(q)$) n'est plus nécessaire. Cependant la mesure de l'accélération \ddot{q}_i doit être disponible. Dans l'expression de la commande $u_i(t)$, on distingue trois actions principales :

1. Une première action caractérisée par un retour d'état ("Feedback") réalisé par le terme u_{PDi} et $\hat{\varepsilon}_i$. Il traduit une intervention de type Proportionnelle-Dérivée (PD) avec des gains constants. Le deuxième terme ($\hat{\varepsilon}_i$) est adaptatif dont la loi d'adaptation est de type intégral avec un gain d'adaptation γ_i variable;
2. Une deuxième action qui est caractérisée par une intervention directe ("Feedforward") via

l'accélération désirée \ddot{q}_i^d . En générale, une telle intervention sert à produire une commande permettant une meilleure poursuite des trajectoires spécifiées, la compensation des effets de perturbations et d'effectuer une anticipation de la variation des trajectoires désirées [SLO91].

La troisième action caractérisée par l'intervention des états du système à travers un modèle flou de type Sugeno adaptatif. Ce dernier nécessite la mesure de l'accélération en plus de la position et la vitesse. Les lois d'adaptation des gains des conséquences des règles floues sont de type intégral avec des matrices d'adaptations $\Gamma_i(t)$ variables. Dans ces lois d'adaptation où le principe de perturbation minimale est respecté, plusieurs types de matrices et de gains d'adaptations peuvent être adoptés. Un cas particulier de matrices à gains décroissant est celui donnée par l'expression (III.26). En effet, on montre dans l'annexe C que la matrice inverse de la solution des équations différentielles (III.26), est donnée par :

$$P^{-1}(t) = [P^{-1}(0) - W_p] \exp(-\lambda t) + W_p \quad (III.39)$$

Donc la matrice inverse $P^{-1}(t)$ décroît exponentiellement pour converger vers la matrice W_p . Dans notre cas, l'utilisation de gains ou matrices d'adaptation variables ($\gamma_i(t)$, $\Gamma_i(t)$) appartenants à la classe \mathfrak{T} améliore les performances du régulateur lorsque les variations du terme de couplage (ou A_i^*, ε_i) ne peuvent être négligées par rapport aux variations des paramètres adaptatifs $\hat{A}_i, \hat{\varepsilon}_i$ (hypothèse H3 n'est pas vérifiée). Une telle situation peut se produire, lorsque les paramètres du système sont variables ou lors de poursuite à des vitesses importantes. Pour montrer cela, on considère l'expression de la fonction de Lyapunov donnée par (III.29):

$$V_i = \frac{1}{2} m_0 r^2 + \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \tilde{A}) + \frac{1}{2} \gamma_0^{-1} \tilde{\varepsilon}^2 \quad (III.40)$$

Puisque l'hypothèse H3 n'est pas vérifiée, on écrit :

$$\tilde{A} = \hat{A} - A^* \quad \text{et} \quad \tilde{\varepsilon} = \hat{\varepsilon} - \varepsilon \quad (III.41)$$

dans ce cas l'expression (III.37) devient :

$$\dot{V}_i = -k_r r^2 + \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \tilde{A}) + \frac{1}{2} \gamma_0^{-1} \tilde{\varepsilon}^2 + \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{A}^*) + \gamma_0^{-1} \tilde{\varepsilon} \dot{\varepsilon} \leq 0 \quad (III.42)$$

Puisque le signe des deux derniers termes ne peuvent être connus, leur présences introduit une perturbation sur la négativité de \dot{V}_i , d'où la condition suffisante de la stabilité n'est pas vérifiée. Cette perturbation peut être compensée grâce à la négativité du deuxième et troisième terme. Cette compensation est d'autant meilleur lorsqu'on choisit : γ_0^{-1} et les valeurs propres de Γ_0^{-1} négatives grandes et/ou γ_0^{-1} et les valeurs propres de Γ_0^{-1} positives petites. Entre les deux, le choix devient contradictoire si on s'intéresse à une plage de gains d'adaptation non importante. Pour cela, il est nécessité d'adopter un compromis de choix.

Il faut noter que le problème d'initialisation des différents gains adaptatifs, qui persiste dans d'autre approches, ne se pose pas. Si $\hat{\varepsilon}_i(t=0)$ et $\hat{A}_i(t=0)$ sont choisis nuls, l'action proportionnelle dérivée (PD), (réalisée par le terme u_{pd}) stabilise la dynamique de l'erreur par intérim. En effet, une étude révèle que les gains proportionnels, et dérivés doivent être suffisamment larges, et l'erreur filtrée doit être initialement ($r(t=0)$) suffisamment petite [LEW93].

III.4.1.2 Application au robot PUMA560

Nous avons appliqué la CDFAS pour la commande en poursuite du robot PUMA560 à trois degrés de liberté présenté dans l'annexe A. Nous avons choisis la trajectoire de Leahvy avec une transition cycloïdale afin d'imposer une continuité sur les position, la vitesse, l'accélération et par conséquent celle de la commande. Avec la méthode de simulation Runge-Kutta d'ordre 4, le pas de

simulation est fixé à 1 ms. Les intégrations dans la loi de commande sont évaluées par la méthode des trapèzes.

Paramètres du régulateur :

Chaque articulation est dotée d'un régulateur local indépendant dont le modèle flou de type Sugeno est d'ordre zéro. Deux fonctions d'appartenance triangulaires sont associées à chacune des trois variables d'entrées (q_i , \dot{q}_i et \ddot{q}_i) telles que montrées par la figure (III.21). Ainsi le modèle flou est caractérisé par huit règles.

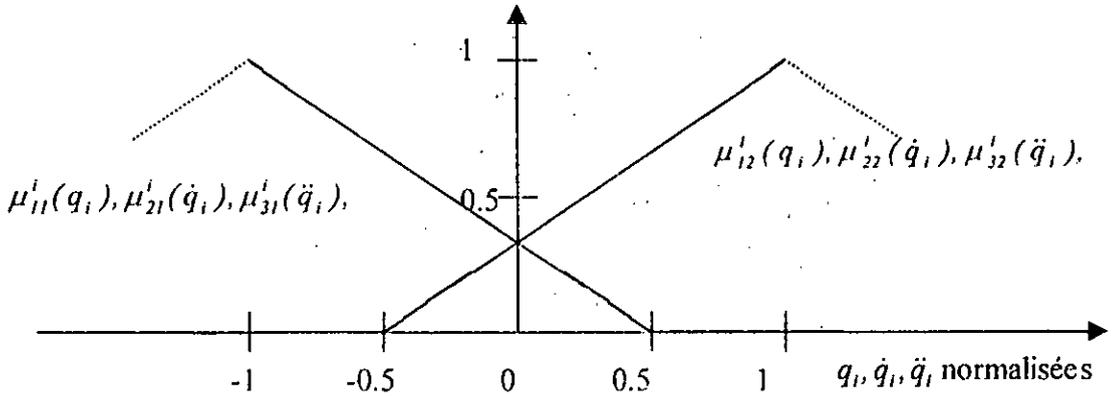


figure (III.21) : fonctions d'appartenance des variables d'entrées

Les gains d'adaptation $\gamma_i(t)$ et $\Gamma_i(t)$ sont choisis telles que les gains inverses sont donnés par (III.39).

Les paramètres de la loi de commande sont regroupés dans la table (III.4)

	Symboles des paramètres	Première articulation	Deuxième articulation	Troisième articulation
	m_{0i}	3	4,2	0,65
Action PD	$k_{0i} ; k_{ui} ; k_{fi}$	200 ; 5 ; 2000	200 ; 5 ; 2000	200 ; 5 ; 2000
Facteur de normalisation	$q_i ; \dot{q}_i ; \ddot{q}_i$	4 ; 5,33 ; 46	4 ; 5,33 ; 46	4 ; 5,33 ; 46
Modèle flou $Z_i^T \hat{A}_i(t) \xi^i$	$\lambda_i ; \Gamma_i(0) ; W_{pi}^{-1}(0)$	2 ; 4545 ; 5000	2 ; 4545 ; 5000	2 ; 4545 ; 5000
Signal auxiliaire $\hat{\varepsilon}_i(t)$	$\lambda_i ; \gamma_i(0) ; W_{pi}^{-1}(0)$	2 ; 9,09 ; 10	2 ; 9,09 ; 10	2 ; 9,09 ; 10

Table (III.4) : Paramètres du régulateur sous la loi CDFAS

Les inerties m_{0i} sont déterminées à partir du modèle dynamique du robot donné dans l'annexe A.

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

Résultats et performances

Test sans charge : Suite à un test sans charge, nous avons obtenu les réponses illustrées par les figures (III.22a,b). On remarque que les articulations effectuent une très bonne poursuite des trajectoires désirées. Les erreurs de poursuite sont quasiment nulles (inférieure à 0.008 deg.). Avec des

commandes lisses l'intervention des modèles flous est primordiale (voir leurs sorties notées u_f). Les gains des règles floues sont caractérisés par une variation lisse où on constate de plus le caractère d'adaptation localisée.

Test de variation de charge : Pour tester la capacité d'adaptation de notre régulateur, nous avons introduit une charge de masse 2 Kg puis de 4 kg qui est relâchée à l'instant $t=0.75$ sec. Les réponses obtenues sont illustrées par la figure (III.23). Malgré la brusque variation de charge, nous constatons que les articulations continuent remarquablement la poursuite des trajectoires désirées avec des erreurs maximales inférieures à 0.02 deg. Ces erreurs convergent asymptotiquement vers zéro.

Test de variation paramétrique : Pour montrer les performances du régulateur en cas de variation paramétrique, nous avons augmenté les valeurs de tous les paramètres du modèle dynamique avec un rapport de 50 % puis de 100 %. D'après les réponses illustrées par la figure (III.24), Nous constatons une légère augmentation des erreurs de poursuite qui restent inférieures à 0.02 deg.

Test de rupture de commande : Suite à une rupture de commande de la deuxième articulation à partir de l'instant $t=0.75$ sec (commande nulle), nous avons obtenu les réponses montrées par la figure (III.25). Avec des erreurs de poursuite maximales inférieures à 0.02 deg, les deux autres articulations font preuve d'une bonne robustesse malgré le couplage important avec la liaison en rupture. En effet, les valeurs relativement élevées des commandes générées reflète l'importance du couplage existant entre articulations.

Test de variation du pas d'échantillonnage : Dans ce test nous portons le pas d'échantillonnage à la valeur 10 ms puis 15 ms. Pour maintenir la stabilité du système de commande, nous avons diminué les paramètres du régulateur comme suivant :

Cas $T=10$ ms :

$$k_{oi} = 100, k_{ii} = 2, k_{ri} = 50, (\Gamma_i(0) = 454.4, W_{pi}^{-1}(0) = 500), (\gamma_i(0) = 0.9, W_{pi}^{-1}(0) = 1) \text{ avec } i=1..3 ;$$

Cas $T=15$ ms :

$$k_{oi} = 30, k_{ii} = 2, k_{ri} = 50, (\Gamma_i(0) = 454.4, W_{pi}^{-1}(0) = 500), (\gamma_i(0) = 0.9, W_{pi}^{-1}(0) = 1) \text{ avec } i=1..3 ;$$

Les réponses obtenues sont illustrées par la figure (III.26). Nous remarquons une poursuite satisfaisante avec des erreurs qui augmentent en fonction du pas d'échantillonnage et s'annulent en régime permanent. Nous constatons qu'après un régime transitoire les commandes deviennent presque identiques.

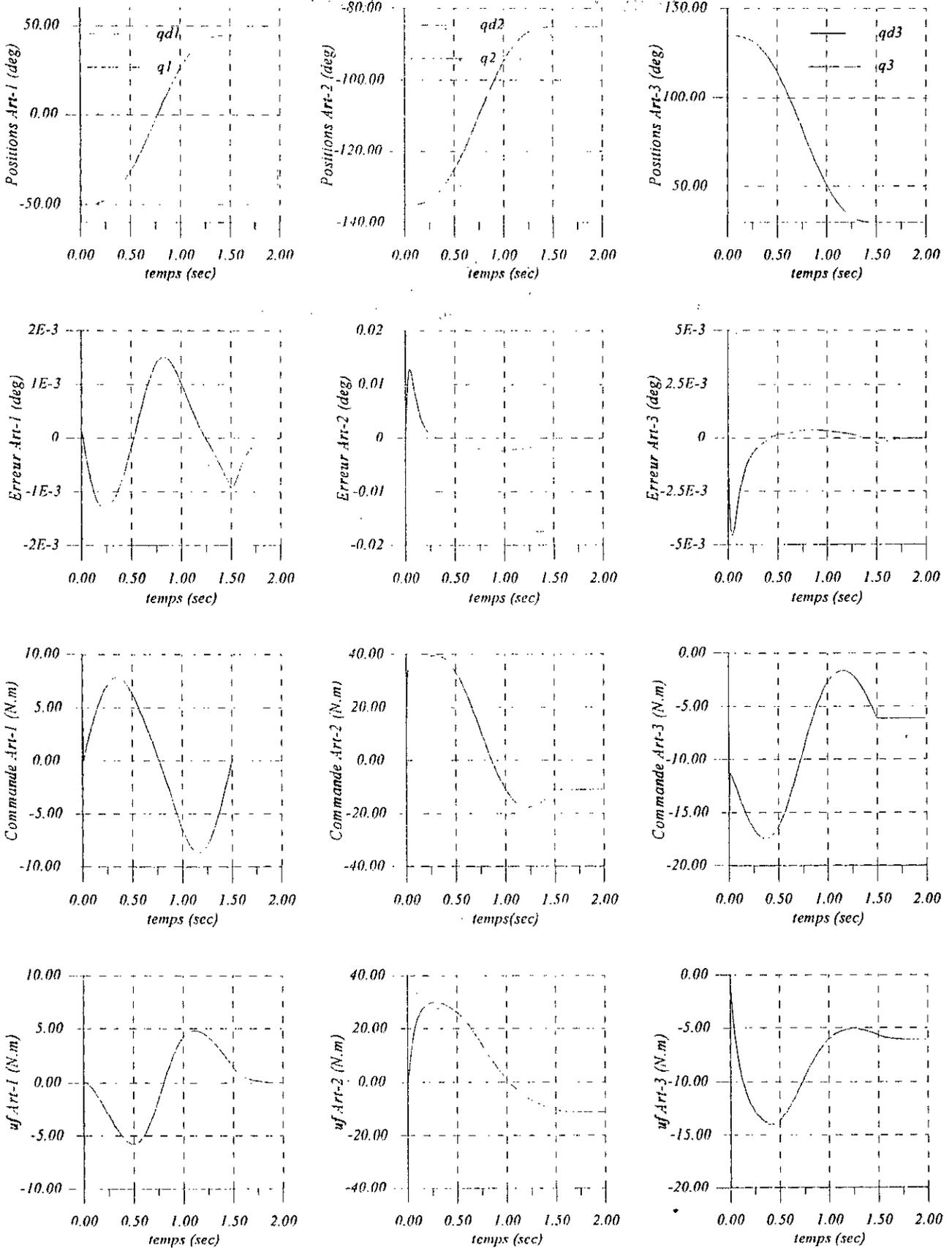


figure (III.22a) : Réponses des différentes articulations soumises à la loi CDFAS :
Test sans charge

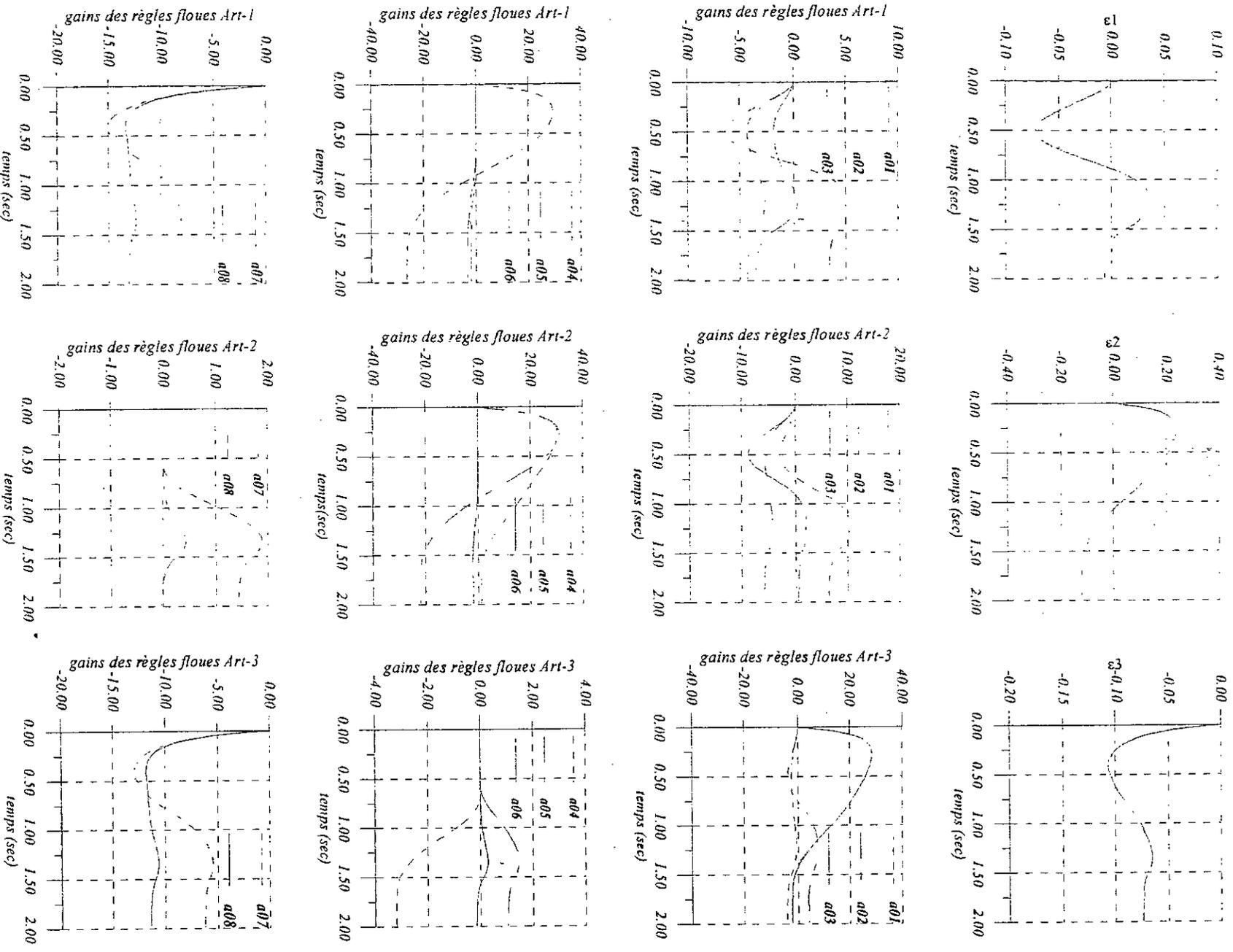


figure (III.22b) : Réponses des différentes articulations soumises à la loi CDFAS :
Test sans charge

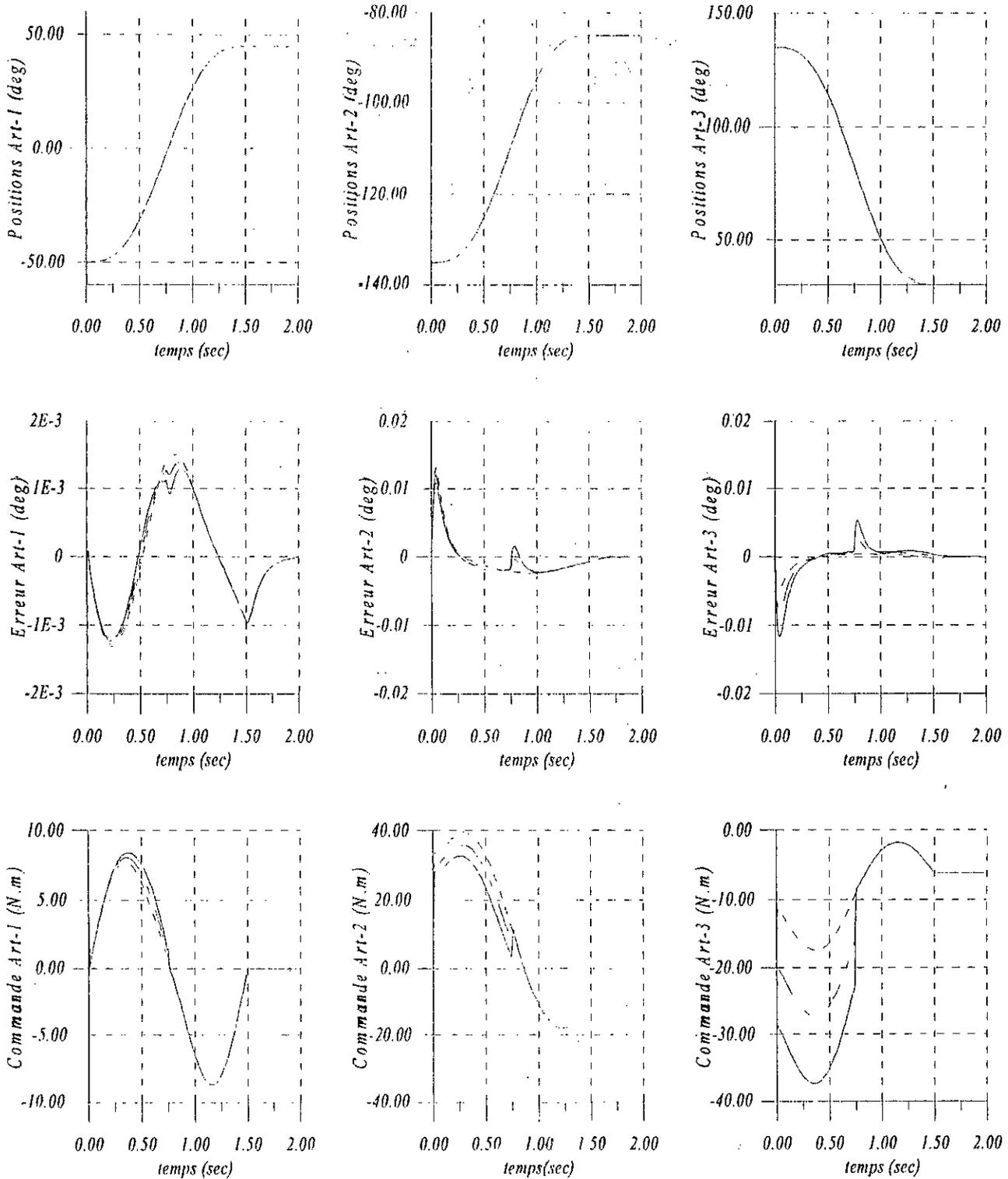
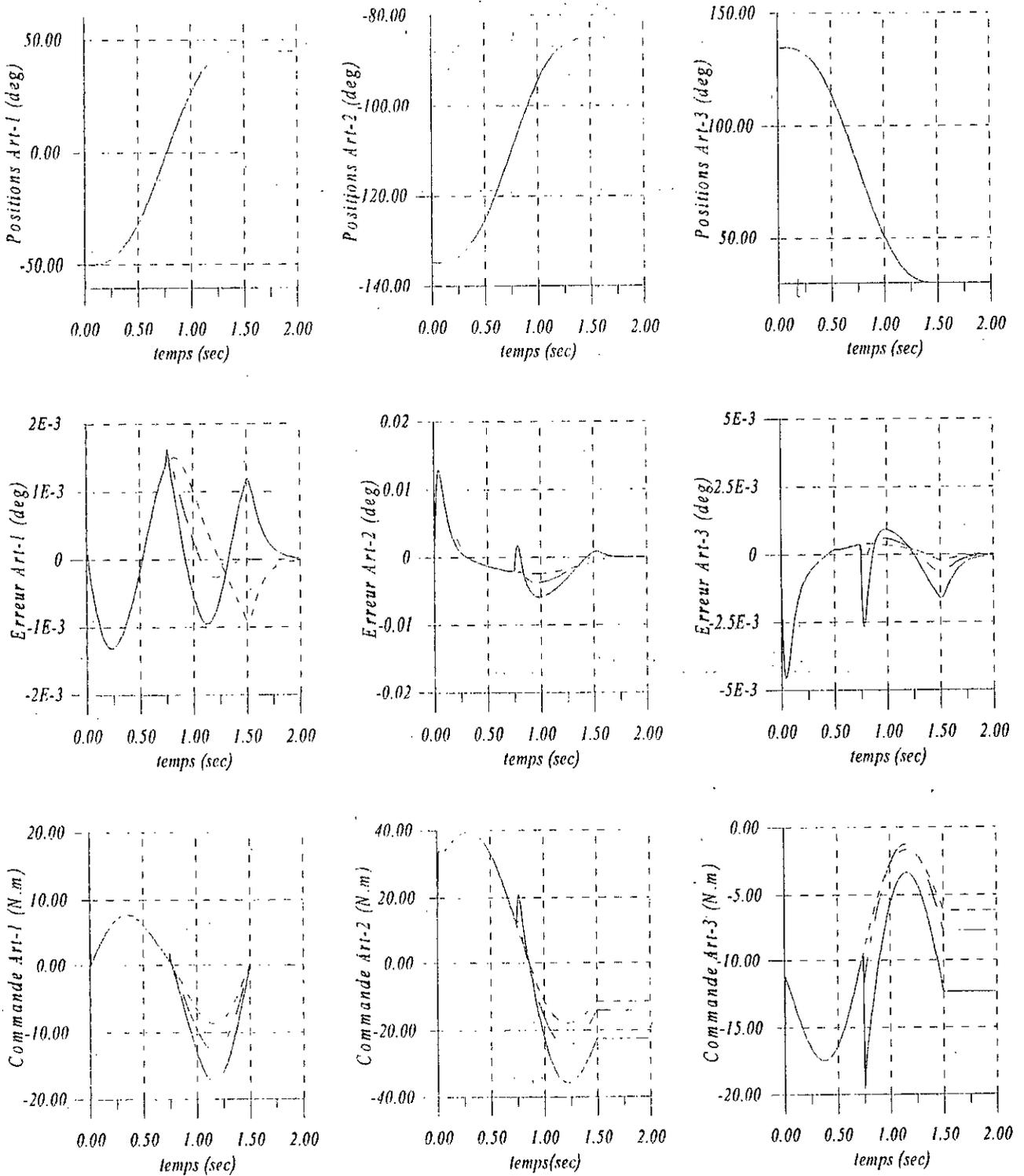


figure (III.23) : Réponses des différentes articulations soumises à la loi CDFAS ;
Test de variation de charge



sans variation.

--- variation paramétrique de +50 %.

-.-.- variation paramétrique de +100 %.

figure (III.24) : Réponses des différentes articulations soumises à la loi CDFAS :
Test de variation paramétrique

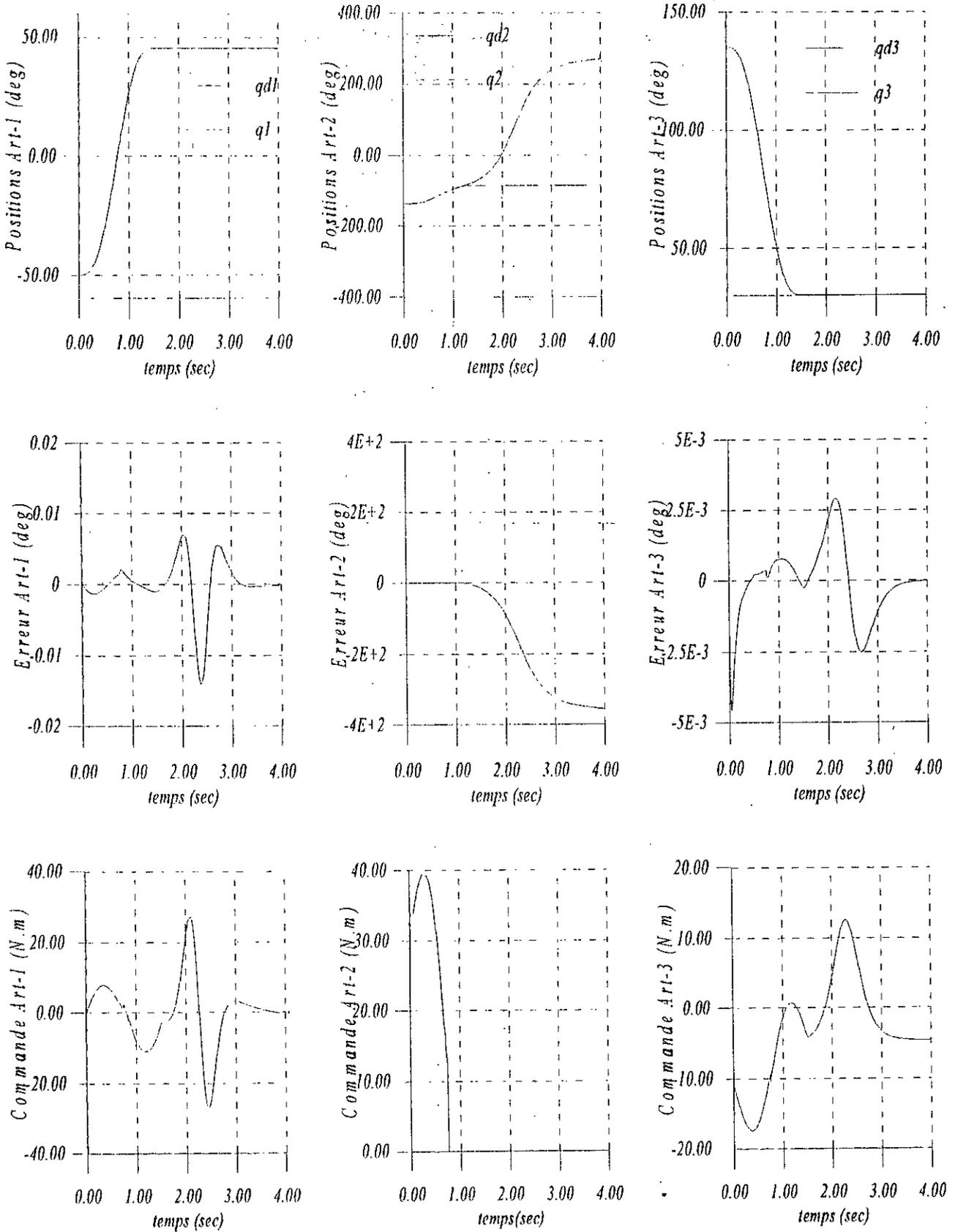


figure (III.25) : Réponses des différentes articulations soumises à la loi CDFA :
Test de rupture de commande

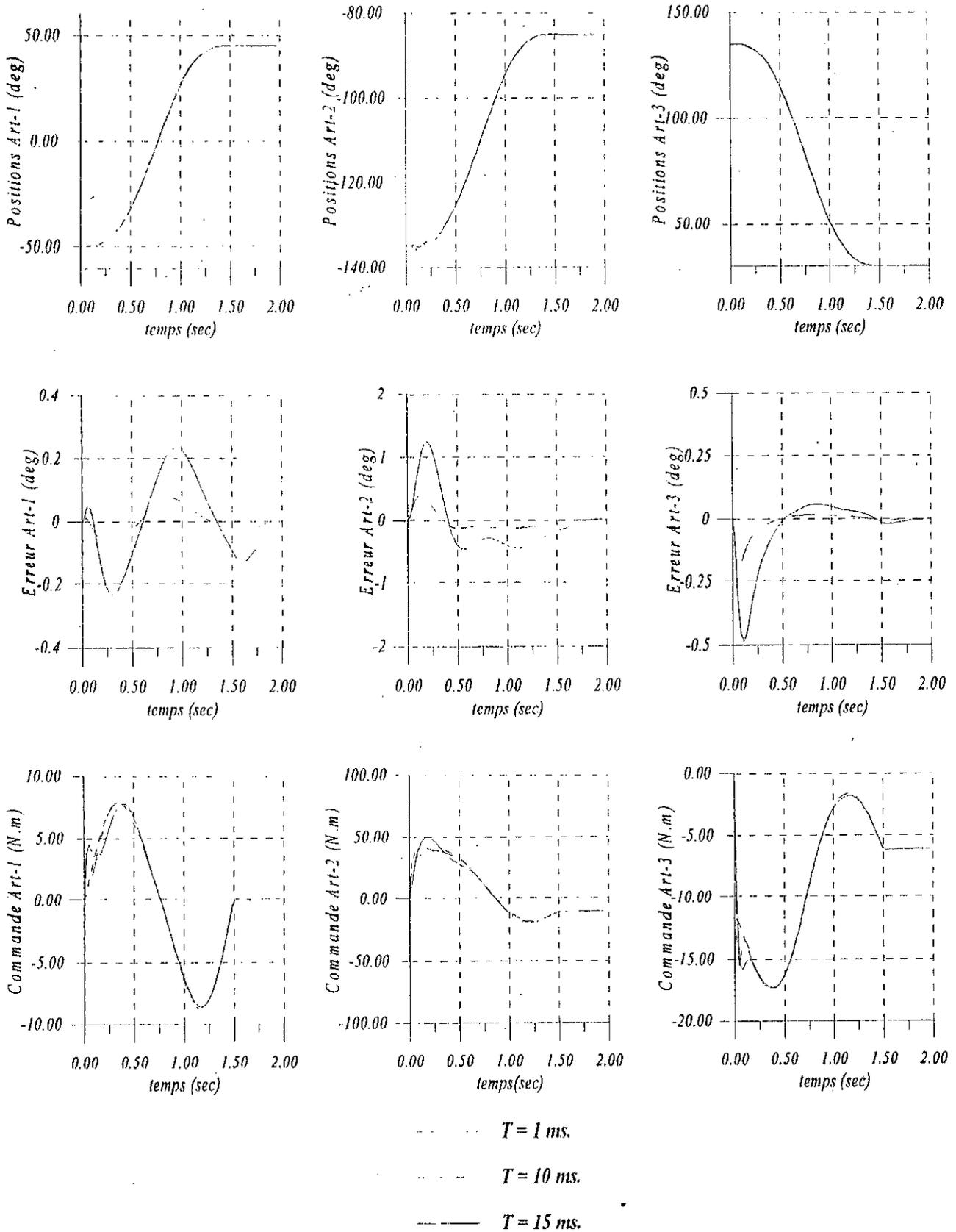


figure (III.26) : Réponses des différentes articulations soumises à la loi CDFAS :
Test de variation du pas d'échantillonnage

III.4.2 Commande Décentralisée Floue Adaptative Robuste (CDFAR)

III.4.2.1 Structure de commande

Dans la loi de CDFAR proposée précédemment, la compensation des termes de couplages est assurée via un modèle flou de Sugeno adaptatif ayant trois entrées q_i, \dot{q}_i et \ddot{q}_i . Dans le cas d'adaptation d'un grand nombre de fonctions d'appartenance pour chacune des variables d'entrées, le nombre de règles à traiter, et des gains à adapter devient important. Ce nombre peut être minimisé considérablement si le nombre de variables d'entrées est réduit.

Comme dans la loi de CDFAR présentée dans la section (§III.3.2), nous proposons d'adopter un modèle flou à deux entrées seulement q_i, \dot{q}_i et d'introduire un terme supplémentaire de robustesse. De la même manière que dans les techniques de commande robuste, cette modification exige l'existence et la connaissance de bornes supérieures de certains paramètres liés à la dynamique du robot [BUH86].

Considérons le modèle dynamique de l'erreur de poursuite de la $i^{ème}$ articulation décrit par l'équation (III.20) :

$$m_{oi}\ddot{e}_i = c_i(q, \dot{q}, \ddot{q}) + \Delta m_{ii}\ddot{q}_i + m_{oi}\ddot{q}_i^d - u_i(t) \tag{III.43}$$

Dès que le terme $-c_i(q, \dot{q}, \ddot{q})$ n'est pas fonction de l'accélération \ddot{q}_i , utilisons un modèle flou de Sugeno à deux entrées q_i, \dot{q}_i pour approximer ce terme. On écrit :

$$c_i(q, \dot{q}, \ddot{q}) = Z_i^T A_i^* \xi^i + \varepsilon_i(t)$$

avec,

- $Z_i^T A_i^* \xi^i$: la sortie de MFS, où Z_i^T, A_i^* et ξ^i sont données, respectivement, par les équations (III.13c,d,e,f) ;
- $\varepsilon_i(t)$: erreur d'approximation.

Dans ce cas, le modèle dynamique de l'erreur (III.43), s'écrit

$$m_{oi}\ddot{e}_i = Z_i^T A_i^* \xi^i + \varepsilon_i(t) + \Delta m_{ii}\ddot{q}_i + m_{oi}\ddot{q}_i^d - u_i(t) \tag{III.44}$$

On considère la classe des robots manipulateurs, dont les termes d'inerties $\Delta m_{ii} \quad i = 1 \dots n$, peuvent être bornées par des constantes connues. On peut écrire :

$$|\Delta m_{ii}(q)| \leq k_{s2i} \quad \forall q, i = 1 \dots n$$

d'après l'équation (III.44), nous proposons la loi de commande suivante :

$$u_i(t) = Z_i^T \hat{A}_i \xi^i + \hat{\varepsilon}_i(t) + u_{si}(t) + m_{oi}\ddot{q}_i^d + u_{rDi}(t) \tag{III.45}$$

u_{si} : Un terme de robustesse introduit pour compenser l'effet du terme $\Delta m_{ii}\ddot{q}_i$.

En remplaçant l'expression de $u_i(t)$ dans (III.44), on obtient le modèle dynamique de l'erreur en boucle fermée :

$$m_{oi}\ddot{e}_i = Z_i^T \tilde{A}_i \xi^i + \tilde{\varepsilon}_i(t) + \Delta m_{ii}\ddot{q}_i - u_{si}(t) - u_{rDi}(t) \tag{III.46}$$

Proposition 5 :

Etant donné un robot manipulateur de n degrés de liberté, dont le modèle dynamique de l'erreur de poursuite de la $i^{ème}$ articulation peut être écrit sous la forme (III.43).

Soit la loi de commande (III.45) et en tenant compte des hypothèses H1 et H3 et les lois d'adaptation (III.28).

Si on choisit :

$$u_{si}(t) = k_{s2i} \cdot |\dot{q}_i| \cdot \text{sign}(r_i) \tag{III.47}$$

on aura :

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

R2 : Les paramètres \hat{A}_i et $\hat{\varepsilon}_i, i = 1 \dots n$ et les commandes $u_i(t) i = 1 \dots n$ sont bornées;

R3: Les erreurs de poursuite $e_i(t), i = 1 \dots n$ sont bornées et convergent, au moins asymptotiquement vers zéro.

Démonstration :

Pour démontrer la Proposition, nous utilisons la méthode directe de Lyapunov. Considérons la fonction de Lyapunov suivante :

$$V_2 = \frac{1}{2} m_0 r^2 + \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \tilde{A}) + \frac{1}{2} \gamma_0^{-1} \tilde{\varepsilon}^2 \quad (\text{III.48})$$

avec, $\Gamma_0, \gamma_0 \in \mathfrak{S}$,

$$r = k_0 e + k_1 \dot{e} \quad k_0, k_1 > 0$$

La fonction V_2 est positive ($V_2 \geq 0$). En utilisant l'expression (III.46), la dérivée \dot{V}_2 s'écrit :

$$\begin{aligned} \dot{V}_2 = r(m_0 k_0 \dot{e} + k_1 Z^T \tilde{A} \xi + k_1 \dot{\tilde{\varepsilon}} + k_1 (\Delta m \ddot{q} - u_s(t)) - k_1 u_{PD}) + \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{\tilde{A}}) + \gamma_0^{-1} \dot{\tilde{\varepsilon}} \tilde{\varepsilon} \\ + \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{\tilde{A}}) + \frac{1}{2} \gamma_0^{-1} \dot{\tilde{\varepsilon}}^2 \end{aligned} \quad (\text{III.49})$$

donc,

$$\dot{V}_2 = \dot{V}_{21} + \dot{V}_{22} + \dot{V}_{23} + \dot{V}_{24} + \dot{V}_{25} \quad (\text{III.50a})$$

où,

$$\dot{V}_{21} = r(m_0 k_0 \dot{e} - k_1 u_{PD}) \quad (\text{III.50b})$$

$$\dot{V}_{22} = \text{Tr}(\tilde{A}^T (r k_1 Z \xi - \Gamma_0^{-1} \dot{\tilde{A}})) \quad (\text{III.50c})$$

$$\dot{V}_{23} = \tilde{\varepsilon} (r k_1 - \gamma_0^{-1} \dot{\tilde{\varepsilon}}) \text{ car } \dot{\tilde{\varepsilon}} = -\dot{\tilde{\varepsilon}} \quad (\text{hypothèse H2}) \quad (\text{III.50d})$$

$$\dot{V}_{24} = k_1 r (\Delta m \ddot{q} - u_s(t)) \quad (\text{III.50e})$$

$$\dot{V}_{25} = \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{\tilde{A}}) + \frac{1}{2} \gamma_0^{-1} \dot{\tilde{\varepsilon}}^2 \quad (\text{III.50f})$$

Pour obtenir $\dot{V}_2 \leq 0$, on choisit :

$$\begin{cases} k_0 m_0 \dot{e} - k_1 u_{PD} = -k_r r & \text{avec } k_r \geq 0 \\ r k_1 Z \xi - \Gamma_0^{-1} \dot{\tilde{A}} = 0 \\ r k_1 - \gamma_0^{-1} \dot{\tilde{\varepsilon}} = 0 \\ u_s(t) = k_{S2} |\ddot{q}| \text{sign}(r) \end{cases} \quad (\text{III.51})$$

On obtient :

$$\dot{V}_{21} = -k_r r^2 \text{ et } \dot{V}_{22} = \dot{V}_{23} = 0$$

$$\dot{V}_{24} = \Delta m \dot{q} r - k_{S2} |\dot{q} r| \leq 0 \text{ car } k_{S2} \geq |\Delta m|$$

D'après la propriété D2 de la classe \mathfrak{S} , on a : $\dot{V}_{25} \leq 0$

donc,

$$\dot{V}_2 = -k_r r^2 + \Delta m \dot{q} r - k_{S2} |\dot{q} r| + \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{\tilde{A}}) + \frac{1}{2} \gamma_0^{-1} \dot{\tilde{\varepsilon}}^2 \leq 0 \quad (\text{III.52})$$

Puisque $V_2 \geq 0, \dot{V}_2 \leq 0$, d'après la théorie de stabilité de Lyapunov, on déduit que V, r (et par conséquent e, \dot{e}), \tilde{A} et $\tilde{\varepsilon}$ sont bornées.

Pour montrer la convergence asymptotique de l'erreur de poursuite vers zéro, il suffit de montrer, en plus, que \dot{V}_2 est uniformément continue. Vu que $\dot{V}_2(t)$ dépend de l'accélération \ddot{q} , l'utilisation de la condition suffisante (\ddot{V}_2 bornée) reste une opération délicate (l'apparition de la dérivée de \ddot{q} dans l'expression de \ddot{V}_2). Pour palier à cette situation, nous proposons d'utiliser le corollaire suivant :

Corollaire [KRS95][LEW93][SPO95a][SPO95b]: On considère une fonction réelle r qui vérifie les conditions suivantes :

- r et \dot{r} sont bornées;
- r est quadratiquement intégrable : $(\int_0^{+\infty} r^2(t) dt < +\infty)$.

alors cette fonction converge asymptotiquement vers zéro.

D'après (III.46); la bornitude de \tilde{A} , $\tilde{\varepsilon}$ et r assure que \ddot{e} est bornée et par conséquent \dot{r} . Selon (III.52), on peut déduire :

$$\int_0^{+\infty} r^2(t) dt < \frac{1}{k_r} (V_2(0) - V_2(\infty)) < +\infty \text{ car } V_2 \text{ est bornée.}$$

d'où la convergence asymptotique de l'erreur filtrée r , et par conséquent celui de e et \dot{e} , vers zéro (résultat R3). Si on choisit $\Gamma = \Gamma_0 k_l$, $\gamma = \gamma_0 k_l$ de la relation (III.51), on écrit :

$$u_{PD} = \left(\frac{k_r k_0}{k_l} \right) e + \left(\frac{m_0 k_0 + k_r k_l}{k_l} \right) \dot{e}$$

$$\dot{\hat{A}} = \Gamma(t) Z \xi^T r \quad \Gamma \in \mathfrak{S} \text{ matrice définie positive.}$$

$$\dot{\hat{\varepsilon}} = \gamma(t) r \quad \gamma \in \mathfrak{S} \text{ constante positive.}$$

- Selon l'hypothèse H1, la bornitude de e , \dot{e} et \ddot{e} implique celle de q , \dot{q} et \ddot{q} .
- \tilde{A} et $\tilde{\varepsilon}$ bornées, implique que \hat{A} bornée (car A^* est constante) et $\hat{\varepsilon}$ bornée car les états (positions, vitesses, accélérations) sont bornés (R2).

La démonstration est valable pour toutes les articulations du robot manipulateur.

Schéma de principe :

Le schéma de principe de la loi de Commande Décentralisée Floue Adaptative Robuste (CDFAR) donnée par (III.45), (III.47) avec les lois d'adaptation (III.28) est montré à la figure (III.27).

D'après la Proposition 5, l'utilisation de la loi de commande de CDFAR, assure la convergence asymptotique des erreurs de poursuite des différentes articulations. La bornitude des variables q_i , \dot{q}_i et \ddot{q}_i est garantie via une adaptation des paramètres \hat{A}_i et ε_i de type intégral.

En introduisant un terme supplémentaire de robustesse, la loi de CDFAR utilise un modèle flou à deux entrées seulement. Dès que la loi de la commande nécessite la connaissance de bornes supérieures des inerties Δm_{ii} (ou $m_{ii}(q)$), $i = 1 \dots n$, la CDFAR reste restreinte à une classe particulière des robots manipulateurs dont les robots rotationnels font partie.

Vu la nature de la fonction $sign$, la commande $u_i(t)$ est discontinue lorsque l'erreur filtrée $r_i(t)$ change de signe. Cela peut provoquer des commutations intempestives ("chattering") de la commande, d'où la nécessité de remplacer la fonction $sign$ par une fonction continue au voisinage de l'origine $r_i = 0$ (voir la remarque dans la section (§III.3.2)).

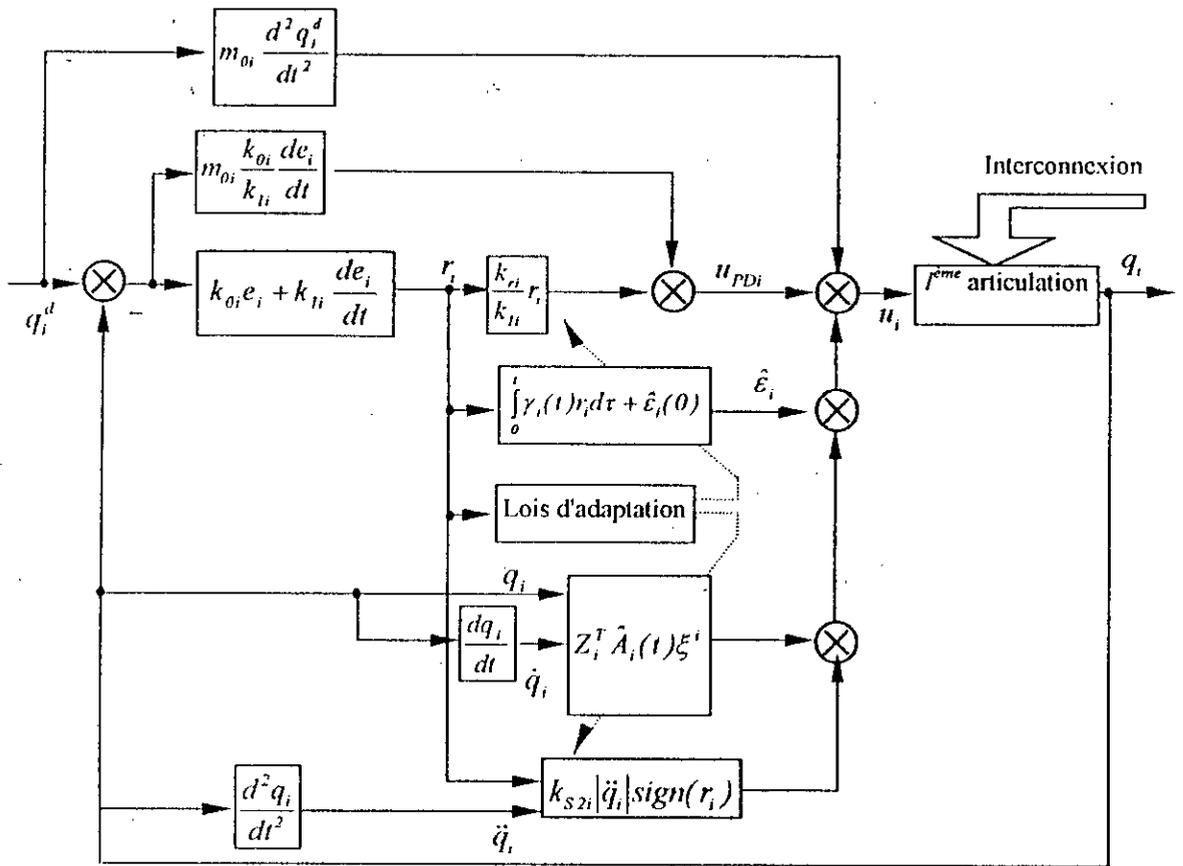


figure (III.27): Schéma de principe de la CDFAR.

III.4.2.2 Application au robot PUMA560

Dans cette section, nous présentons les résultats obtenus suite à l'application de la technique de CDFAR pour la commande en poursuite du robot manipulateur PUMA560.

Paramètres du régulateur :

Dans la loi de commande, nous avons choisis un modèle flou de Sugeno d'ordre zéro ayant les deux entrées q_i , \dot{q}_i . Deux fonctions d'appartenance triangulaires sont associées à chacune des variables d'entrées telles que montrées par la figure (III.21). Ainsi, on aura quatre règles floues dans chaque régulateur local. Les gains d'adaptation $\gamma_i(t)$ et $\Gamma_i(t)$ sont choisis telles les gains inverses sont donnés par (III.39).

Les paramètres de la loi de commande sont regroupés dans la table (III.5)

Les inerties m_{oi} et les paramètres k_{s2i} sont déterminées à partir du modèle dynamique du robot donné dans l'annexe A.

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

Résultats et performances

Test sans charge : Nous avons soumis le robot à un test sans charge en suivant la trajectoire de Leahvy avec transition cycloïdale. Nous avons obtenu les résultats montrés par la figure (III.28). Nous remarquons une très bonne poursuite des trajectoires où les erreurs maximales sont inférieures à 0.4 deg. Les erreurs s'annulent en régime permanent. Ayant des valeurs relativement importantes, les commandes sont oscillatoires en particulier celle de la première articulation.

	Symboles des paramètres	Première articulation	Deuxième articulation	Troisième articulation
	m_{0i}	3.16	6.9	1.3
Action PD	$k_{0i} ; k_{1i} ; k_{2i}$	20 ; 5 ; 60	20 ; 5 ; 40	20 ; 5 ; 40
Facteur de normalisation	$q_i ; \dot{q}_i$	4 ; 5.33	4 ; 5.33	4 ; 5.33
Modèle flou $Z_i^T \hat{\Lambda}_i(t) \xi^t$	$\lambda_i ; \Gamma_i(0) ; W_{pi}^{-1}(0)$	2 ; 454.5 ; 500	2 ; 454.5 ; 500	2 ; 454.5 ; 500
Signal auxiliaire $\hat{e}_i(t)$	$\lambda_i ; \gamma_i(0) ; W_{pi}^{-1}(0)$	2 ; 0.9 ; 1	2 ; 0.9 ; 1	2 ; 0.9 ; 1
Terme de robustesse	k_{S2i}	3	4.2	0.65

Table (III.5) : Paramètres du régulateur sous la loi CDFAR

Test sans charge avec lissage : Dans le but de réaliser une bonne poursuite avec des commandes réalisables, nous avons remplacé la fonction discontinue *sign* de la loi de commande par la fonction continue suivante :

$$C_{\delta}(r_i) = \frac{r_i}{|r_i| + \delta_i} \quad \delta_1 = 1.00 ; \delta_2 = \delta_3 = 0.08$$

Les figures (III.29a,b) illustrent les réponses obtenues. Les erreurs de poursuite sont maintenues inférieures à 0.1 deg. Nous constatons que les commandes sont devenues lisses avec à l'exception d'ondulations de faible amplitude qui apparaissent en régime permanent. Ces ondulations, qui se reflètent aussi sur les gains des règles floues sont dues au terme de robustesse u_{S1} .

Test de variation de charge : L'élément terminal porte une charge de 2 Kg puis de 4 Kg jusqu'à l'instant $t=0.75$ sec. Les réponses montrées par la figure (III.30) confirment les bonnes performances de notre régulateur vis-à-vis des variations massiques introduites.

Test de variation paramétrique : Suite à une variation paramétrique de + 50% puis de +100% nous avons obtenu les résultats illustrés par la figure (III.31). Nous constatons une adaptation instantanée des commandes.

Test de rupture de commande : Pour montrer l'efficacité de l'indépendance des commandes des différentes articulations, nous appliquons une commande nulle (chute libre) à la deuxième articulation à partir de l'instant $t=0.75$ sec. D'après les réponses montrées par la figure (III.32), les erreurs de poursuite sont inférieures à 0.05 deg et s'annule en régime permanent. Des commandes relativement considérables sont générées pour de faire face au couplage important entre articulation.

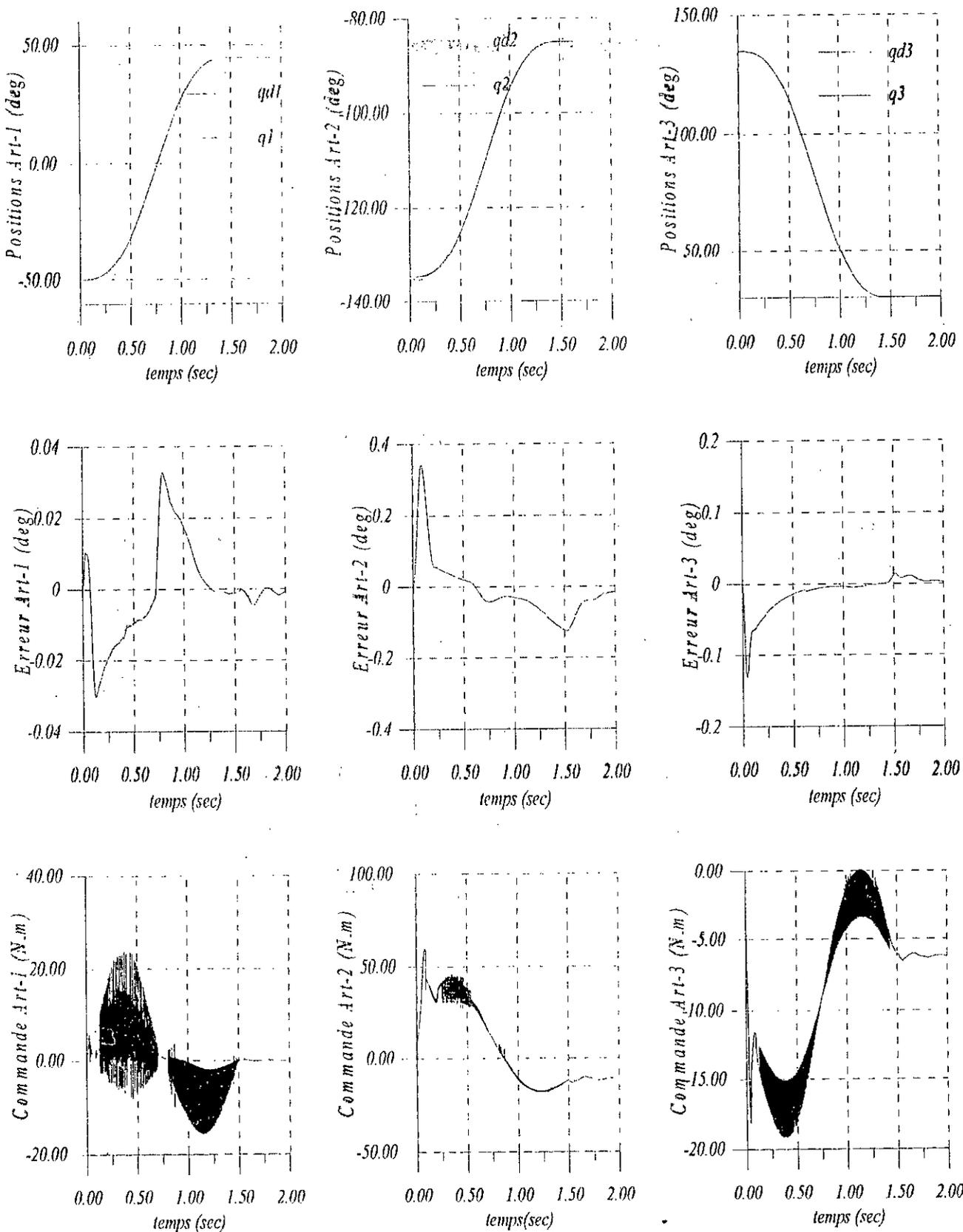


figure (III.28) : Réponses des différentes articulations soumises à la loi CDFAR :
 Test sans charge

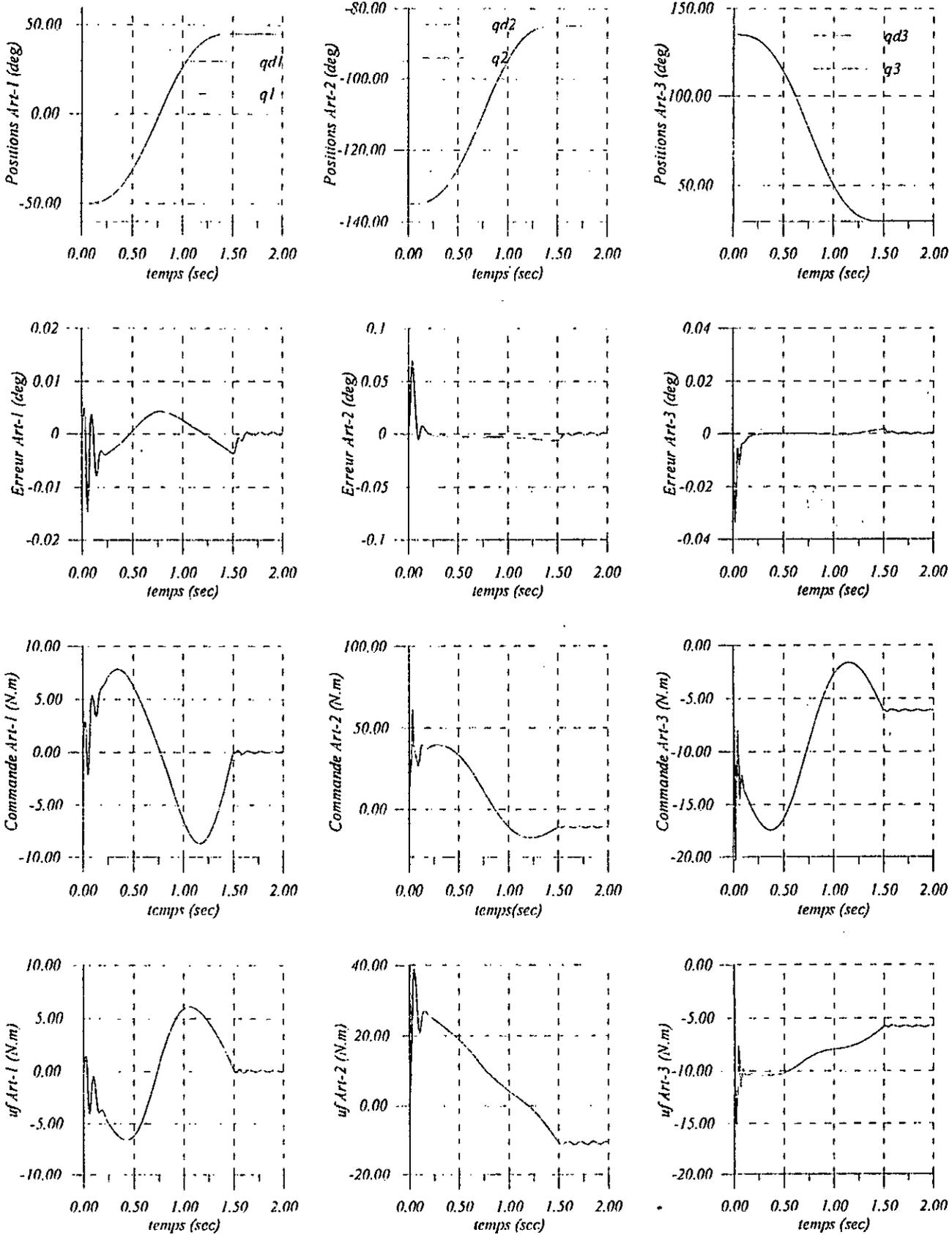


figure (III.29a) : Réponses des différentes articulations soumises à la loi CFAR :
Test sans charge avec lissage

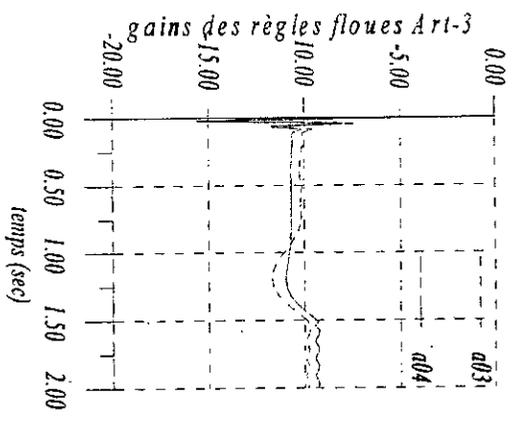
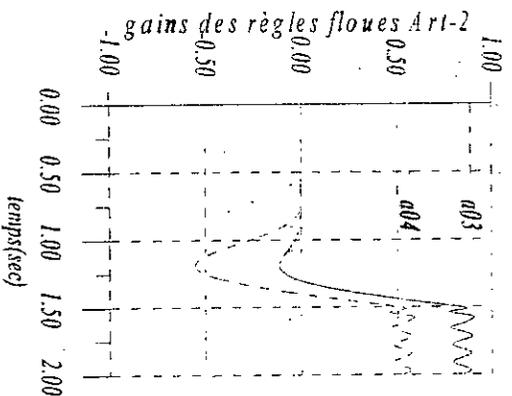
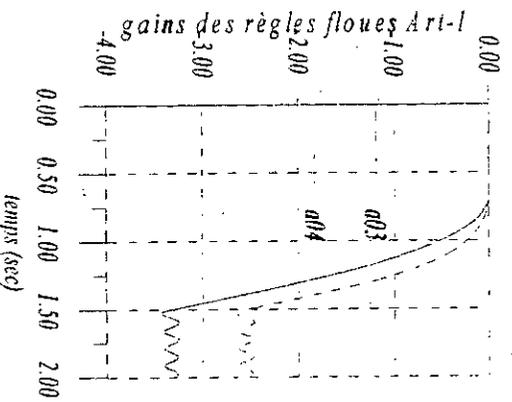
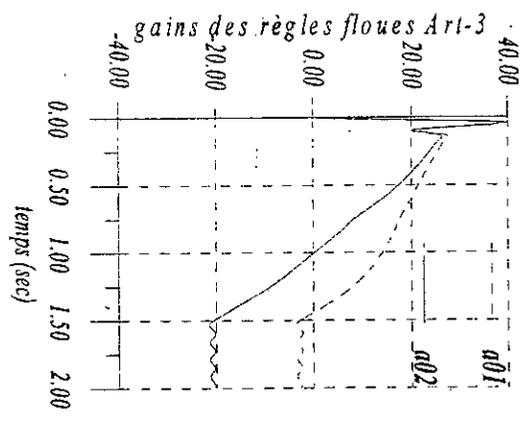
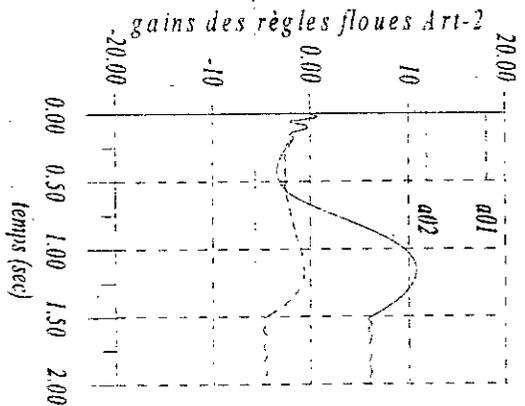
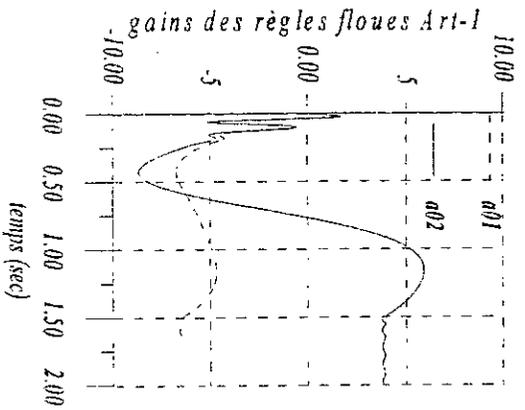
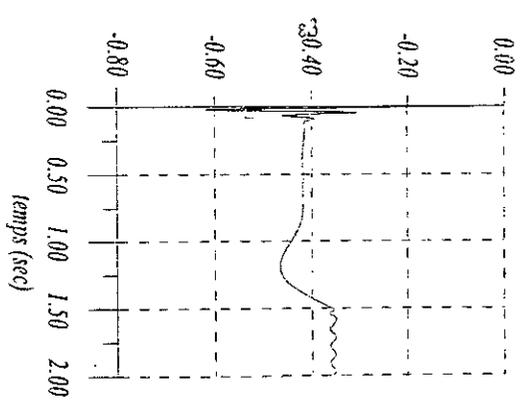
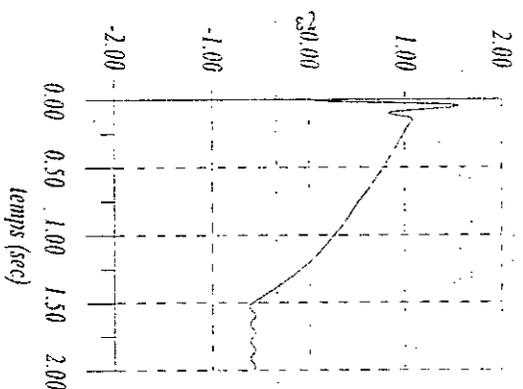
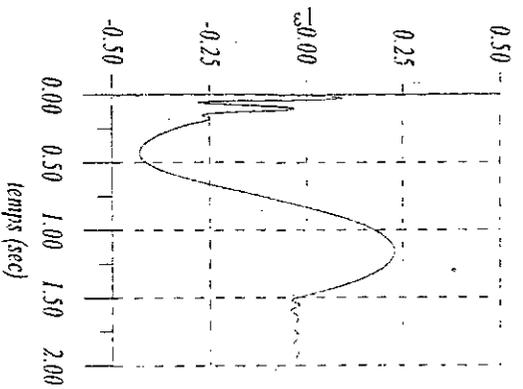


figure (III.29b) : Réponses des différentes articulations soumises à la loi CDFAR :
Test sans charge avec lissage

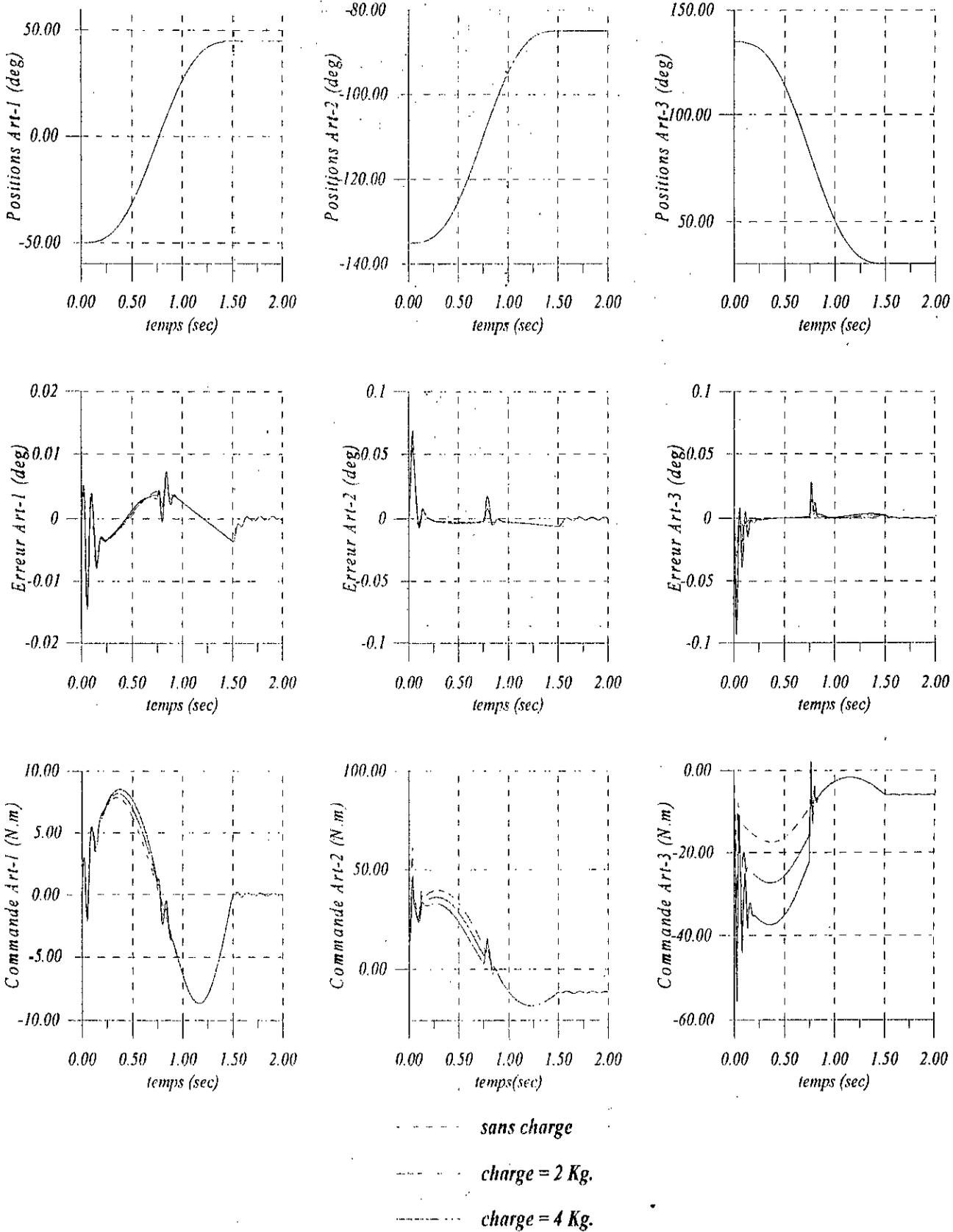
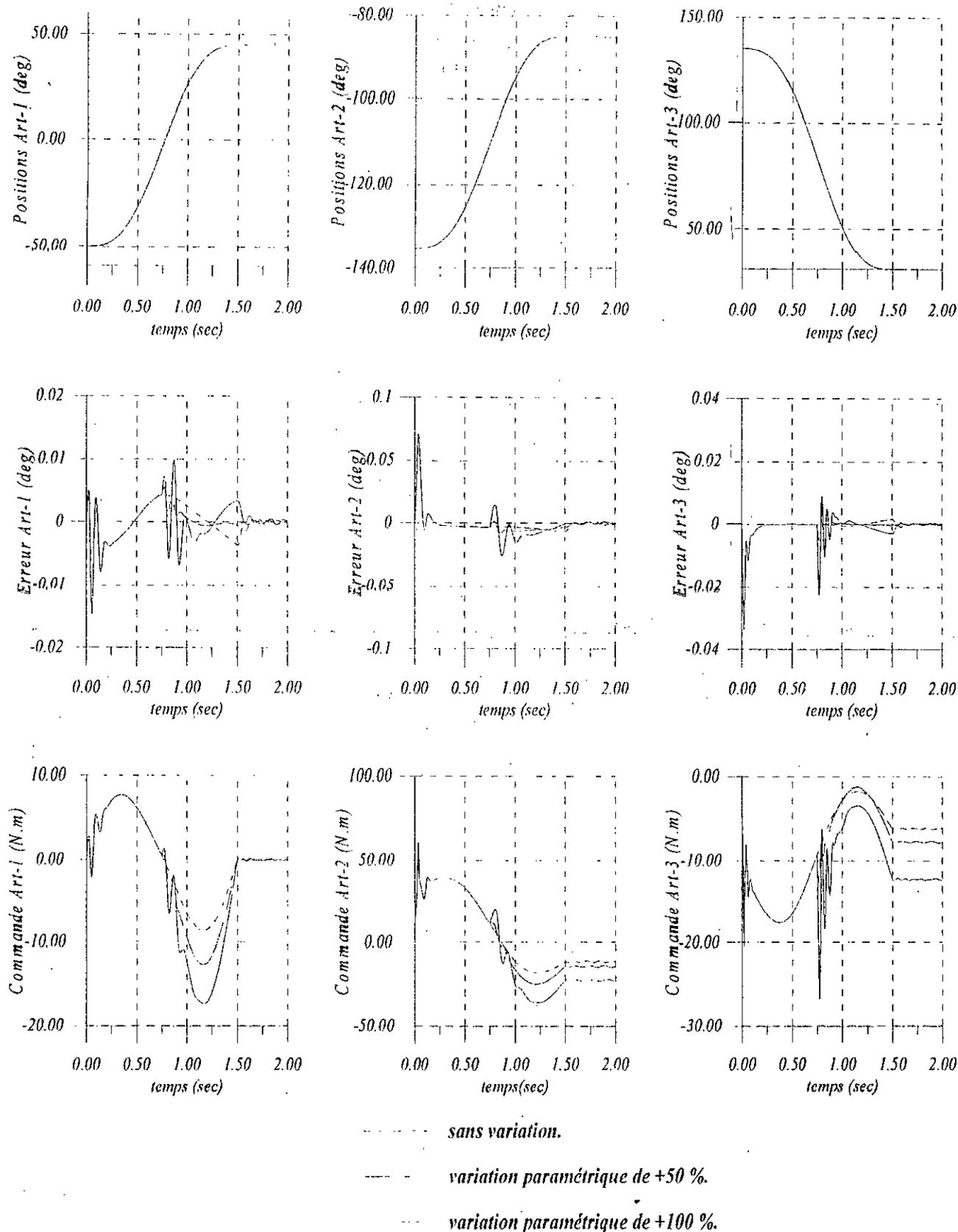


figure (III.30) : Réponses des différentes articulations soumises à la loi CDFAR :
Test de variation de charge



*figure (III.31) : Réponses des différentes articulations soumises à la loi CDFAR :
 Test de variation paramétrique*

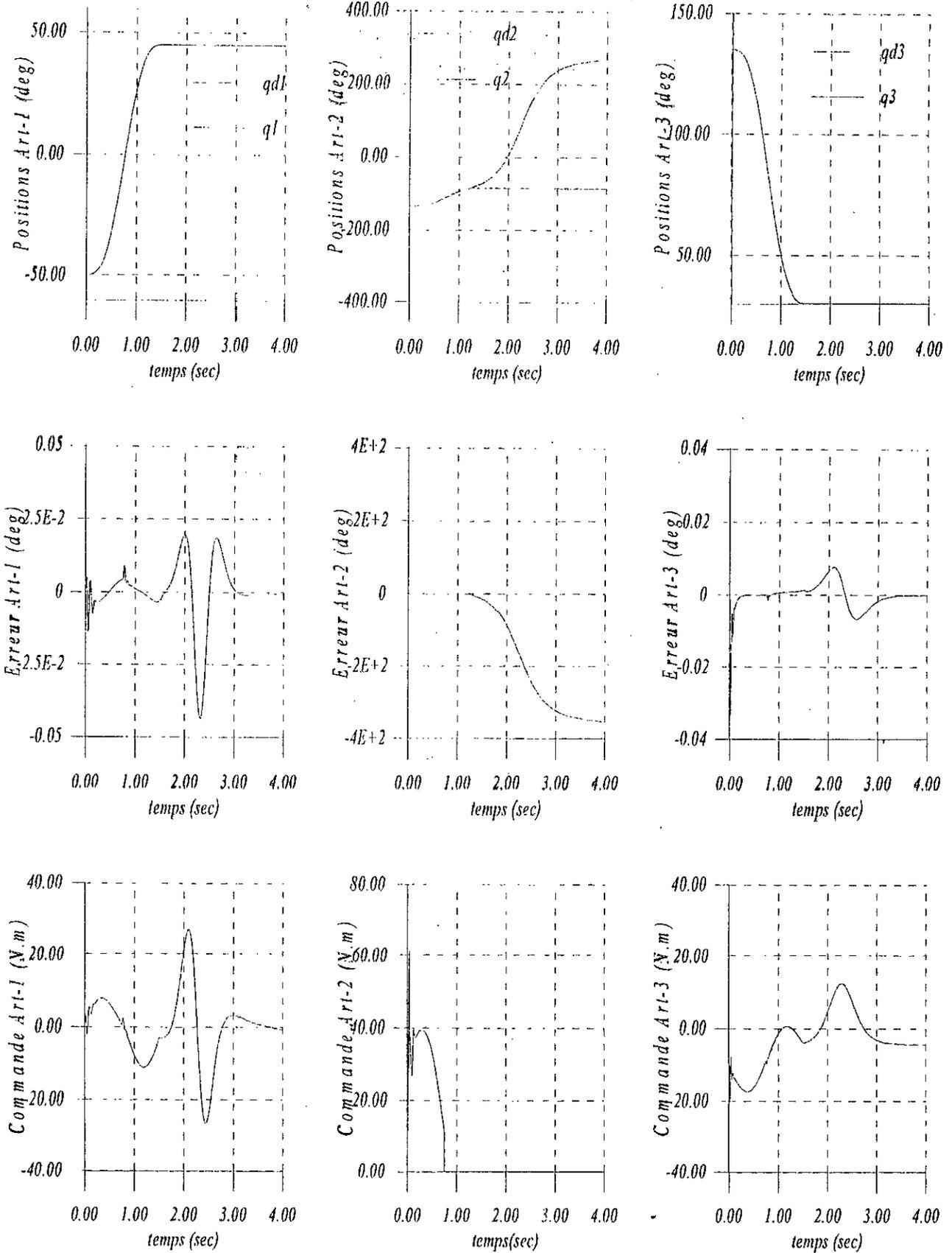


figure (III.32) : Réponses des différentes articulations soumises à la loi CDFAR :
Test de rupture de commande

III.4.3 Commande Décentralisée Floue Adaptative avec Zone Morte (CDFAZM)

III.4.3.1 Structure de commande

La loi de **CDFAR** proposée précédemment assure la stabilité asymptotique de l'erreur via une commande bornée. Cependant vu la nature de la fonction *sign*, la commande $u_i(t)$ est discontinue lorsque l'erreur filtrée $r_i(t)$ change de signe. Ce qui peut provoquer des effets indésirables tels que le phénomène de "chattering".

Pour palier au problème de discontinuité, trois solutions sont possibles :

1. Une première solution consiste à remplacer la fonction *sign* par une fonction continue. Cette solution n'assure plus la convergence asymptotique de l'erreur [SPO95c].
2. Une deuxième solution consiste à remplacer le terme de robustesse par un terme adaptatif tel qu'on a effectué dans la loi de **CDFAM**. Dans ce cas la variation de l'inertie Δm_{ii} est supposée lente par rapport à celle du gain d'adaptation adopté.
3. Une troisième solution, que nous adoptons ici, consiste à utiliser la technique de la zone morte [SLO91][SPO95b][CHA96]. Elle est basée sur l'observation que la discontinuité de la commande se provoque au faible valeur de l'erreur filtrée lorsque cette dernière change de signe. En pratique, une telle situation se manifeste en présence de bruits de mesure ou de perturbation même de faible niveau. Le principe de cette technique consiste à désactiver la mesure de l'erreur au voisinage du zéro ($r = 0$). En contre partie, cette solution engendre une dégradation de la précision statique [SLO91].

On considère l'erreur filtrée avec zone morte r_{Ai} , définie par :

$$r_{Ai} = r_i - \Delta_i \text{sat}\left(\frac{r_i}{\Delta_i}\right) \quad i = 1 \dots n \tag{III.53}$$

où, $r_i = k_{0i}e + k_{1i}\dot{e} \quad k_{0i}, k_{1i} \geq 0$

Δ_i : constante positive permettant de définir la largeur de la zone morte.

sat : fonction de saturation définie par :
$$\text{sat}(x) = \begin{cases} x & \text{si } |x| \leq 1 \\ \text{sign}(x) & \text{si } |x| > 1 \end{cases}$$

L'introduction de la zone morte, offre les propriétés utiles suivantes :

P1 : si $|r_i| \leq \Delta_i$ alors $r_{Ai} = \dot{r}_{Ai} = 0$;

P2 : si $|r_i| > \Delta_i$ alors $|r_{Ai}| = |r_i| - \Delta_i$;

P3 : $r_{Ai} \text{sat}\left(\frac{r_i}{\Delta_i}\right) = |r_{Ai}|$.

Considérons d'autre part, le modèle dynamique de l'erreur donné par (III.44), et proposons la loi de commande suivante :

$$u_i(t) = Z_i^T \hat{A}_i \xi_i^t + \dot{\varepsilon}_i(t) + m_{0i} \ddot{q}_i^d + u_{SAi}(t) + u_{PDi}(t) \tag{III.54}$$

en boucle fermée, on obtient :

$$m_{0i} \ddot{e}_i = Z_i^T \tilde{A}_i \xi_i^t + \tilde{\varepsilon}_i(t) + \Delta m_{ii} \ddot{q}_i - u_{SAi}(t) - u_{PDi}(t) \tag{III.55}$$

où le terme robustesse u_{SAi} et le terme de l'action PD sont, respectivement, donnée par :

$$u_{SAi}(t) = k_{SAi} |\dot{q}_i| \text{sat}\left(\frac{r_i}{\Delta_i}\right) \tag{III.56}$$

$$u_{rD\Delta}(t) = \left(\frac{k_{ri}k_{oi}}{k_{ji}} \right) e_i + \left(\frac{m_{oi}k_{oi} + k_{ri}k_{ji}}{k_{ji}} \right) \dot{e}_i - \frac{k_{ri}}{k_{ji}} \Delta_i \operatorname{sat}\left(\frac{r_i}{\Delta_i}\right) \quad (\text{III.57})$$

avec, k_{oi} ; k_{ji} et k_{ri} des constants positifs.

Considérons la loi d'adaptation :

$$\begin{cases} \dot{\hat{A}}_i = \Gamma_i(t) Z_i \xi^{i^T} r_{\Delta i} & \Gamma_i \in \mathfrak{S} \\ \dot{\hat{e}}_i = \gamma_i(t) r_{\Delta i} & \gamma_i \in \mathfrak{S} \end{cases} \quad (\text{III.58})$$

Proposition 6 :

Etant donné un robot manipulateur à n degrés de liberté dont le modèle dynamique de l'erreur de poursuite de chaque articulation peut être écrit sous la forme (III.44).

Si on choisit la loi de commande (III.54), (III.56), (III.57) et en tenant compte des hypothèses H1, H2 et des lois d'adaptation (III.58) on aura :

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

R2 : Les paramètres estimés \hat{A}_i, \hat{e}_i et la commande $u_i(t)$ sont bornées ;

R3 : Les erreurs de poursuite $e_i(t) i = 1 \dots n$ sont bornées et convergent, au moins, asymptotiquement vers un voisinage de zéro. En plus, on a :

$$\exists t_0 > 0 \quad \begin{cases} |e_i(t)| \leq \frac{\Delta_i}{k_{oi}}, \quad \forall t \geq t_0 \\ |\dot{e}_i(t)| \leq \frac{2\Delta_i}{k_{ji}} \quad \forall t \geq t_0 \end{cases} \quad (\text{III.59})$$

Démonstration :

Considérons la fonction de Lyapunov candidate suivante :

$$V_3 = \frac{1}{2} m_{oi} r_{\Delta}^2 + \frac{1}{2} \operatorname{Tr}(\tilde{A}^T \Gamma_0^{-1} \tilde{A}) + \frac{1}{2} \gamma_0^{-1} \tilde{\varepsilon}^2 \quad (\text{III.60})$$

selon la valeur de r on distingue les deux cas suivants :

Premier cas : $|r| \leq \Delta$

D'après la propriété P1 :

$$\dot{V}_3 = -\operatorname{Tr}(\tilde{A}^T \Gamma_0^{-1} \dot{\hat{A}}_i) - \frac{1}{2} \gamma_0^{-1} \dot{\tilde{\varepsilon}} \tilde{\varepsilon} = 0 \quad (\text{III.61})$$

donc V_3 , et par conséquent $\tilde{A}, \tilde{\varepsilon}, \hat{A}, \hat{e}$ et $\varepsilon(t)$, sont bornées.

Deuxième cas : $|r| \geq \Delta$

Après dérivation de l'expression de la fonction V_3 , et en utilisant le modèle de l'erreur donné par (III.43), on obtient :

$$\dot{V}_3 = \dot{V}_{31} + \dot{V}_{32} + \dot{V}_{33} + \dot{V}_{34} + \dot{V}_{35} \quad (\text{III.62a})$$

avec,

$$\dot{V}_{31} = r_{\Delta} (m_{oi} k_{oi} \dot{e} - k_{ri} u_{rD\Delta}) \quad (\text{III.62b})$$

$$\dot{V}_{32} = \left(r_{\Delta} k_{ji} \tilde{A} Z \xi - \operatorname{Tr}(\tilde{A} \Gamma_0^{-1} \dot{\hat{A}}) \right) \quad (\text{III.62c})$$

$$\dot{V}_{33} = \tilde{\varepsilon} (r_{\Delta} k_{ji} - \gamma_0^{-1} \dot{\hat{e}}) \quad (\text{III.62d})$$

$$\dot{V}_{34} = r_{\Delta} (\Delta m \dot{q} - u_{5\Delta}) \quad (\text{III.62e})$$

$$\dot{V}_{35} = \frac{1}{2} \text{Tr}(\tilde{A}^T \Gamma_0^{-1} \tilde{A}) + \frac{1}{2} \gamma_0^{-1} \tilde{\varepsilon}^2 \tag{III.62f}$$

En utilisant les expressions (III.56), (III.57), et (III.58), on aura :

$$\dot{V}_{31} = -k_r r_A^2, \dot{V}_{32} = \dot{V}_{33} = 0 \quad \text{avec} \quad \Gamma = \Gamma_0 k_l \in \mathfrak{I} \quad \text{et} \quad \gamma = \gamma_0 k_l \in \mathfrak{I}$$

En utilisant l'expression du terme u_{SA} (donnée par (III.56) et la propriété P3, on obtient :

$$\dot{V}_{34} = (\Delta m \ddot{q}r_A - k_{S2} |\ddot{q}r_A|) \leq 0 \quad \text{car} \quad k_{S2} > |\Delta m|$$

et d'après la propriété D2 : $\dot{V}_{35} \leq 0$

donc, la dérivée \dot{V}_3 devient : $\dot{V}_3 \leq -k_r r_A^2 \leq 0$

$V_3 \geq 0$ et $\dot{V}_3 \leq 0$ implique que V_3 et par conséquent r_A, \tilde{A} et $\tilde{\varepsilon}$ sont bornées. Pour montrer la convergence asymptotique de l'erreur r_A on utilise le corollaire présenté dans la section précédente.

Puisque r_A est bornée, il reste à montrer la condition d'intégralité r_A^2 et de bornitude de la dérivée \dot{r}_A .

D'après l'expression de \dot{V} , l'erreur r_A est quadratiquement intégrable car :

$$\int_0^{+\infty} r_A^2 dt < \frac{1}{k_r} (V_3(0) - V_3(+\infty)) < \infty \tag{III.63}$$

En considérant l'expression (III.55), \dot{r}_A est bornée car r_A, \tilde{A} et $\tilde{\varepsilon}$ sont bornées. D'où la convergence asymptotique de r_A vers zéro, ou :

$$\exists t_0 \geq 0 \quad |r(t)| \leq \Delta \quad \forall t \geq t_0 \tag{III.64}$$

Pour déterminer la zone de convergence de l'erreur et de sa dérivée, on considère le cas général d'un filtre linéaire stable d'entrée x et de sortie y et de fonction de transfert $N(s)$.

Si : $|y(0)| < x_{max} \|N(s)\|_1$

alors [SPO95b] : $|y(t)| \leq x_{max} \|N(s)\|_1 \quad \forall t > 0$

avec, x_{max} : la borne supérieure de l'entrée, donc $x_{max} = \max_t |x(t)|$

$\|N(s)\|_1$: la norme-1 de la fonction de transfert $N(s)$ définie par :

$$\|N(s)\|_1 \triangleq \int_{-\infty}^{+\infty} |g_n(t)| dt \tag{III.65}$$

avec $g_n(t)$ la réponse impulsionnelle de $N(s)$.

Dans notre cas, on choisit : $x(t) = r(t)$, $y(t) = e(t)$, donc la fonction de transfert est donnée par :

$$N(s) = \frac{1}{k_0 + k_1 s} \tag{III.66}$$

on obtient : $\exists t_0 > 0 \quad |e(t)| \leq \frac{\Delta}{k_0}, \quad \forall t \geq t_0$

D'autre part : $\dot{e} = \frac{r - k_0 e}{k_1}$

en appliquant l'inéquation triangulaire, on aura :

$$\exists t_0 > 0 \quad |\dot{e}(t)| \leq \frac{2\Delta}{k_1} \quad \forall t \geq t_0 \tag{III.67}$$

Quant à la bornitude des paramètres : \hat{A} et \hat{e} , on a :

$$\begin{cases} \tilde{A} = A^* - \hat{A} & \text{est bornée} \\ \tilde{\varepsilon} = \varepsilon_f + \varepsilon - \hat{\varepsilon} & \text{est bornée} \end{cases} \Rightarrow \begin{cases} \hat{A} & \text{car la matrice } A^* \text{ est bornée;} \\ \hat{\varepsilon} & \text{car tous les états sont bornés.} \end{cases}$$

Schéma de principe :

Le schéma de principe de la CDFAZM donnée par (III.54), (III.56), (III.57) est montré par la figure (III.33).

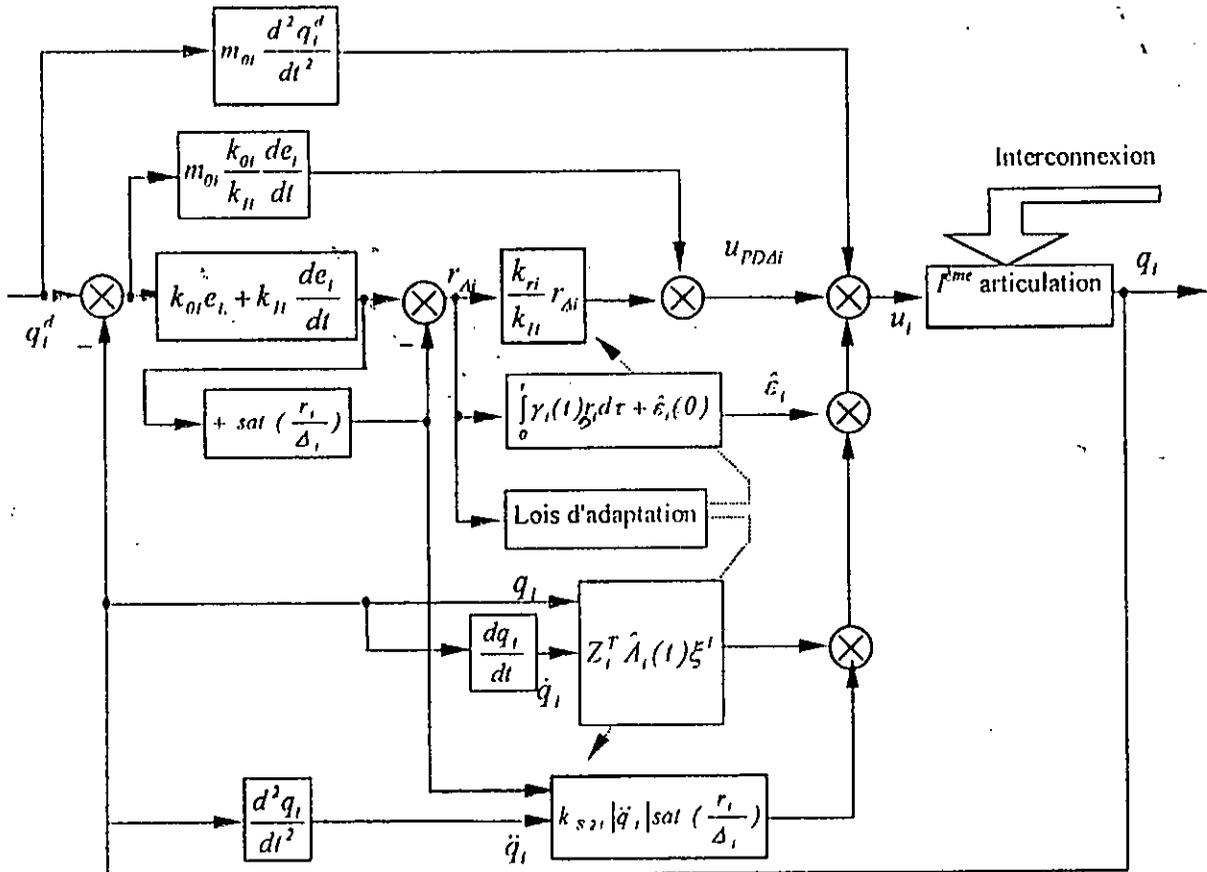


figure (III.33) : Schéma de principe de CDFAZM

L'application de la technique de zone morte sur la loi de CDFAR, nous a permis de construire la loi de CDFAZM. Cette dernière assure la convergence asymptotique de l'erreur de poursuite vers une zone voisine du zéro. La taille de cette zone peut être réduite arbitrairement via l'ajustement des gains k_{oi} et Δ_i .

Par rapport à la loi de CDFAR, trois modifications principales ont été apportées :

Le terme de robustesse est caractérisé par une variation sans discontinuité. La fonction *sign* est remplacée par la fonction continue *sat* ;

L'adaptation des paramètres \hat{A}_i et $\hat{\varepsilon}_i$ est désactivée dans la zone morte ($|r_i| \leq \Delta_i$);

Le terme $u_{PD\Delta i}$ est de structure variable. En effet, on peut distinguer les deux cas suivants:

$$u_{PD\Delta i}(t) = \begin{cases} \left(\frac{k_{ri} k_{oi}}{k_{ii}} \right) e_i + \left(\frac{m_{oi} k_{oi} + k_{ri} k_{ii}}{k_{ii}} \right) \dot{e}_i - \frac{k_{ri}}{k_{ii}} \Delta_i & \text{si } |r_i| > \Delta_i \\ \left(\frac{m_{oi} k_{oi}}{k_{ii}} \right) \dot{e}_i & \text{si } |r_i| \leq \Delta_i \end{cases}$$

En dehors de la zone morte, le terme $u_{PD\Delta t}$ réalise une action proportionnelle-dérivée (PD) en parallèle avec une action constante. Lors d'entrée dans la zone morte, ce terme se transforme en une dérivation pure.

III.4.3.2 Application au robot PUMA560

Dans cette section, nous présentons les résultats obtenus suite à l'application de la technique de CDFAZM pour la commande en poursuite du robot manipulateur PUMA560.

Paramètres du régulateur :

Dans la loi de commande nous avons choisis des modèles flous identiques à ceux adoptés dans l'application de la CDFAR.

Les paramètres de la loi de commande sont regroupés dans la table (III.6).

	Symboles des paramètres	Première articulation	Deuxième articulation	Troisième articulation
	m_{oi}	3.16	4.2	0.65
Zone morte	Δ_i	0.08	0.04	0.04
Action PD	$k_{oi} ; k_{vi} ; k_{ri}$	600 ; 5 ; 2000	600 ; 5 ; 2000	600 ; 5 ; 2000
Facteur de normalisation	$q_i ; \hat{q}_i$	4 ; 5.33	4 ; 5.33	4 ; 5.33
Modèle flou $Z_i^T \hat{A}_i(t) \xi^i$	$\lambda_i ; \gamma_i(0) ; W_{pi}^{-1}(0)$	2 ; 4545 ; 5000	2 ; 4545 ; 5000	2 ; 4545 ; 5000
Signal auxiliaire $\hat{e}_i(t)$	$\lambda_i ; \gamma_i(0) ; W_{pi}^{-1}(0)$	2 ; 9.09 ; 10	2 ; 9.09 ; 10	2 ; 9.09 ; 10
Terme de robustesse	k_{s2i}	3	4.2	0.65

Table (III.6) : Paramètres du régulateur sous la loi CDFAZM

Les inerties m_{oi} et les paramètres k_{s2i} sont déterminées à partir du modèle dynamique du robot donné dans l'annexe A.

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

Résultats et performances

Test sans charge : Nous avons soumis le robot à un test sans charge en suivant la trajectoire de Leahvy avec transition cycloïdale. Nous avons obtenu les résultats montrés par les figures (III.34a,b). Nous remarquons une très bonne poursuite des trajectoires désirées. Les erreurs maximales sont inférieures à 0.008 deg. Les commandes sont devenues lisses grâce à l'introduction de la zone morte sur les erreurs filtrées. Cependant des erreurs constantes sont maintenues en régime permanent 0.0025 deg, 0.004 deg et 0.004 deg, respectivement pour la première, deuxième et troisième articulation. Ces erreurs résiduelles vérifient l'inégalité (III.59). Nous constatons que l'intervention du modèle flou sur la commande est primordiale dans la deuxième et troisième articulation contrairement au cas de la première.

Test de variation de charge : Une charge de 2 Kg puis de 4 Kg est portée par l'élément terminal jusqu'à l'instant $t=0.75$ sec où elle est relâchée. Les réponses sont montrées par la figure (III.35). Nous constatons que les erreurs restent pratiquement invariables.

Test de variation paramétrique : Suite à une variation paramétrique de + 50% puis de +100% nous avons obtenu les résultats illustrés par la figure (III.36). Les erreurs de poursuite sont maintenues inférieures à 0.005 deg. Les erreurs permanentes sont indépendantes des variations introduites.

Test de rupture de commande : Pour montrer l'efficacité de la décentralisation dans notre régulateur, une rupture de la commande de la deuxième articulation est appliquée à partir de l'instant $t=0.75$ sec (commande nulle). D'après les réponses obtenues, montrées par la figure (III.37), nous constatons la robustesse de notre régulateur. En effet, à l'exception d'une léger augmentation des erreurs de poursuite, nous remarquons une très bonne poursuite malgré le couplage important existant entre articulations.

III.5 Conclusion :

Dans ce chapitre, nous nous sommes intéressés à la commande décentralisée floue adaptative continue appliquée aux robots manipulateurs.

Les techniques de commande de la première approche CDFA, CDFAR et CDFAM se sont avérées judicieuses au vu des résultats de simulation. En effet, les multiples tests effectués via la trajectoire de Leahvy [LEA90] ont montré des performances fortes intéressantes (des faibles erreurs de poursuites qui s'annulent en régime permanent). Bien que la technique de CDFAR soit applicable à une classe particulière des robots manipulateurs, elle permet de diminuer les dimensions des modèles flous. L'introduction d'un lissage dans la loi de CDFAR, nous a permis d'éviter une activation excessive et des oscillations intempestives des commandes. Ce qui peut être atteint en utilisant la loi de CDFAM.

L'application des techniques de commande de la deuxième approche (CDFAS, CDFAR, et CDFAZM) nous ont permis d'obtenir de très bons résultats où les erreurs de poursuite sont quasiment nulles. Les différentes réponses présentées confirme la robustesse relative des régulateurs construits vis-à-vis des variations massiques et paramétriques. La technique de CDFAZM assure la convergence des erreurs de poursuite vers un voisinage de zéro de taille qui peut être réduite considérablement.

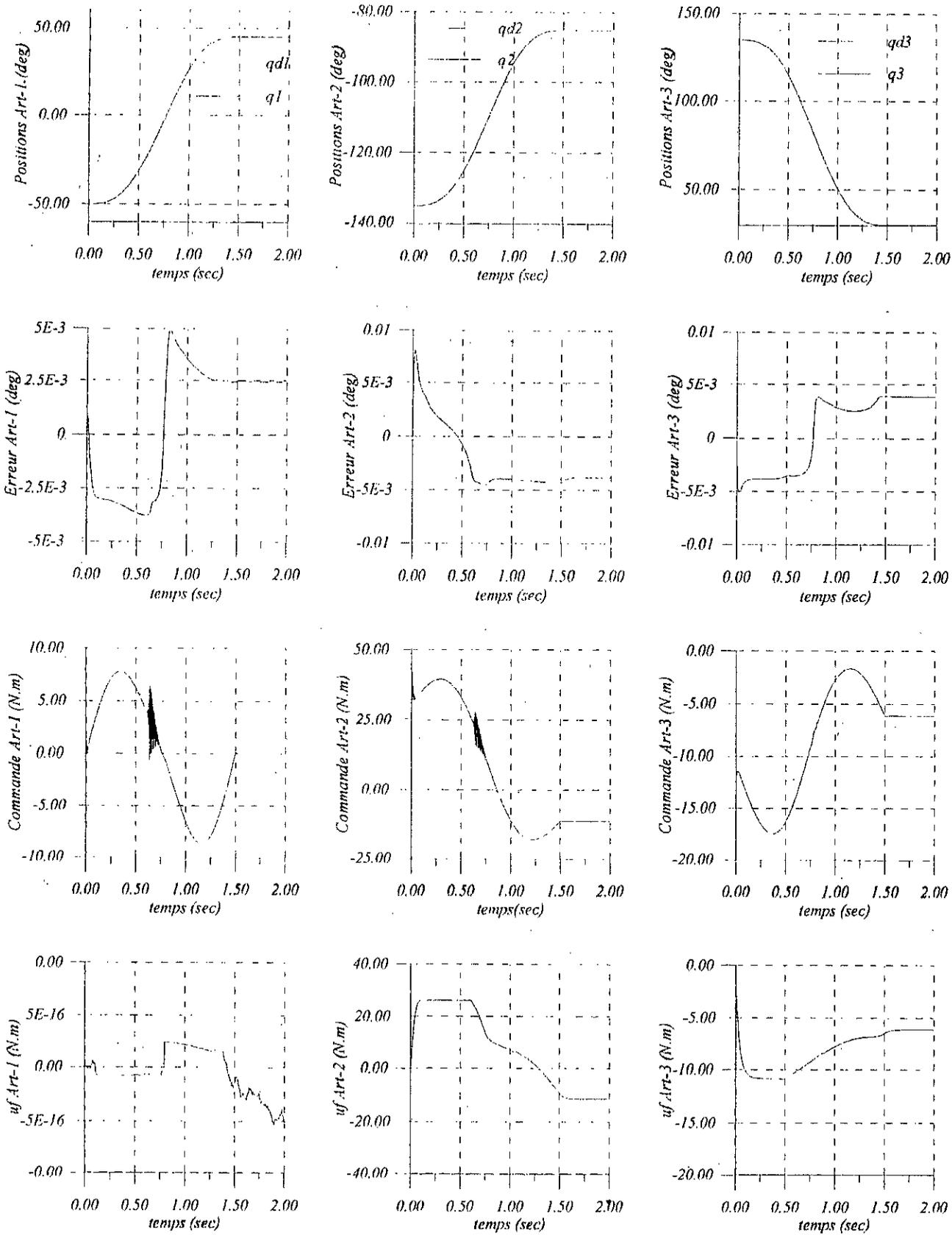


figure (III.34a) : Réponses des différentes articulations soumises à la loi CDFAZM :
Test sans charge

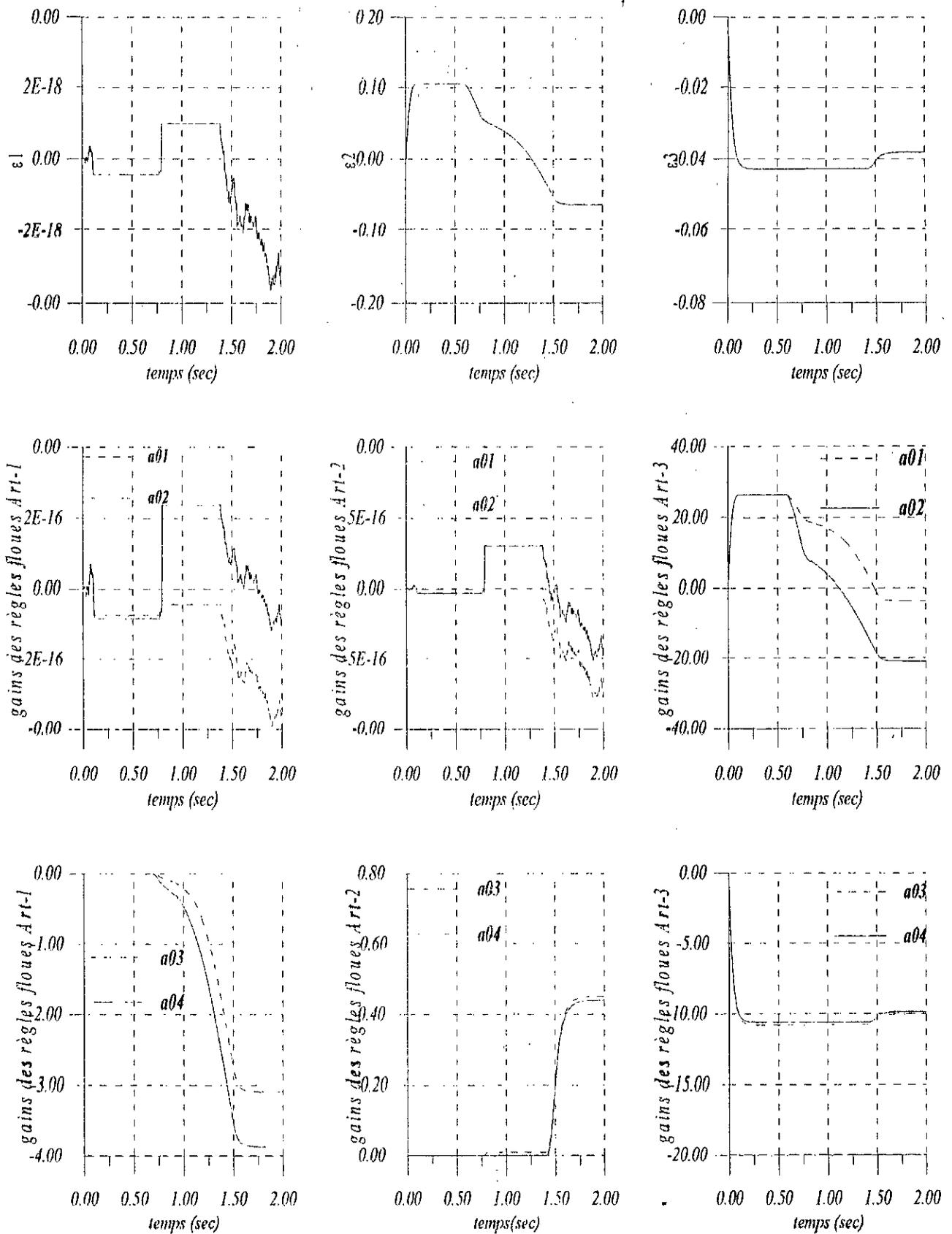


figure (III.34b) : Réponses des différentes articulations soumises à la loi CDEFAZM :
Test sans charge

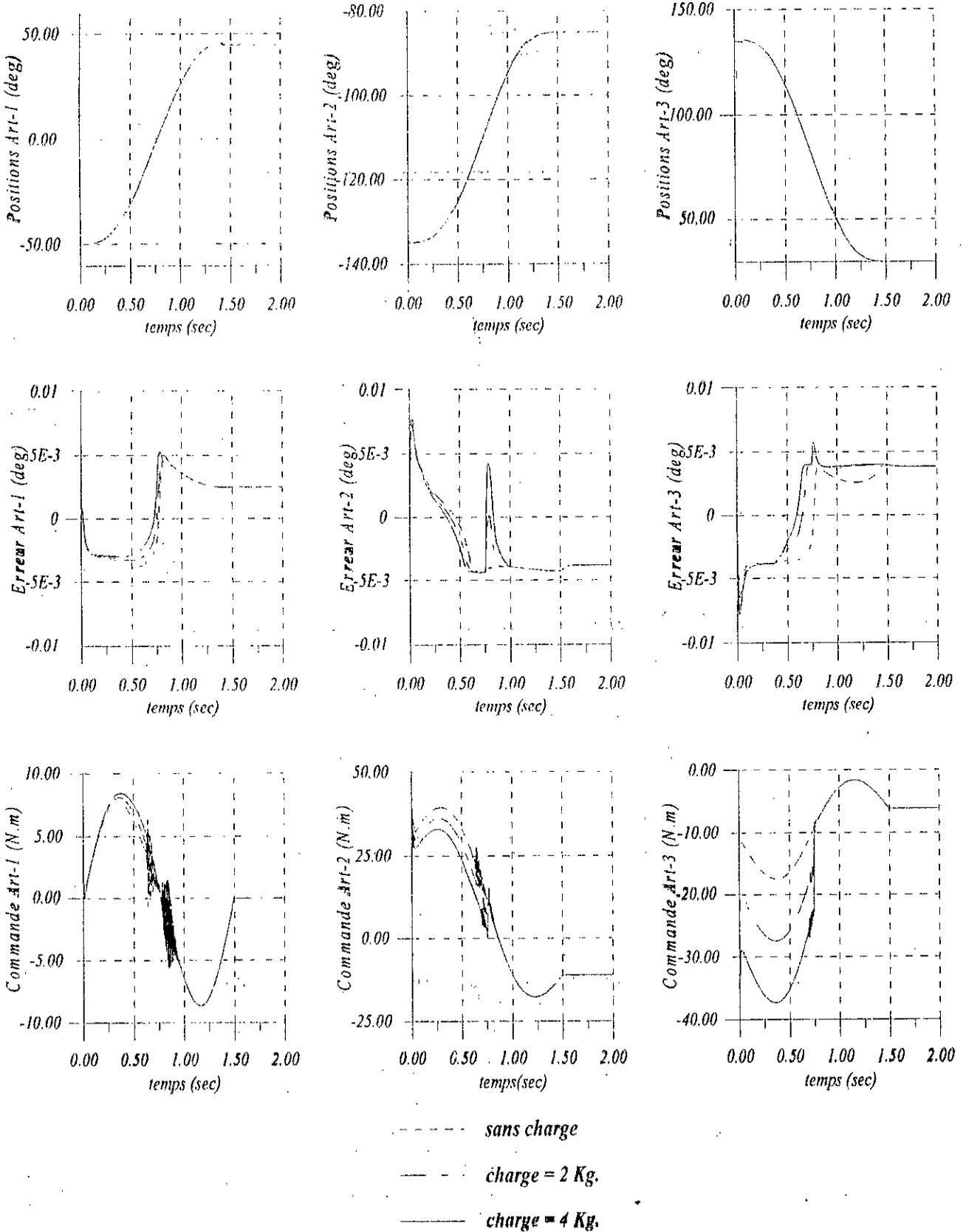
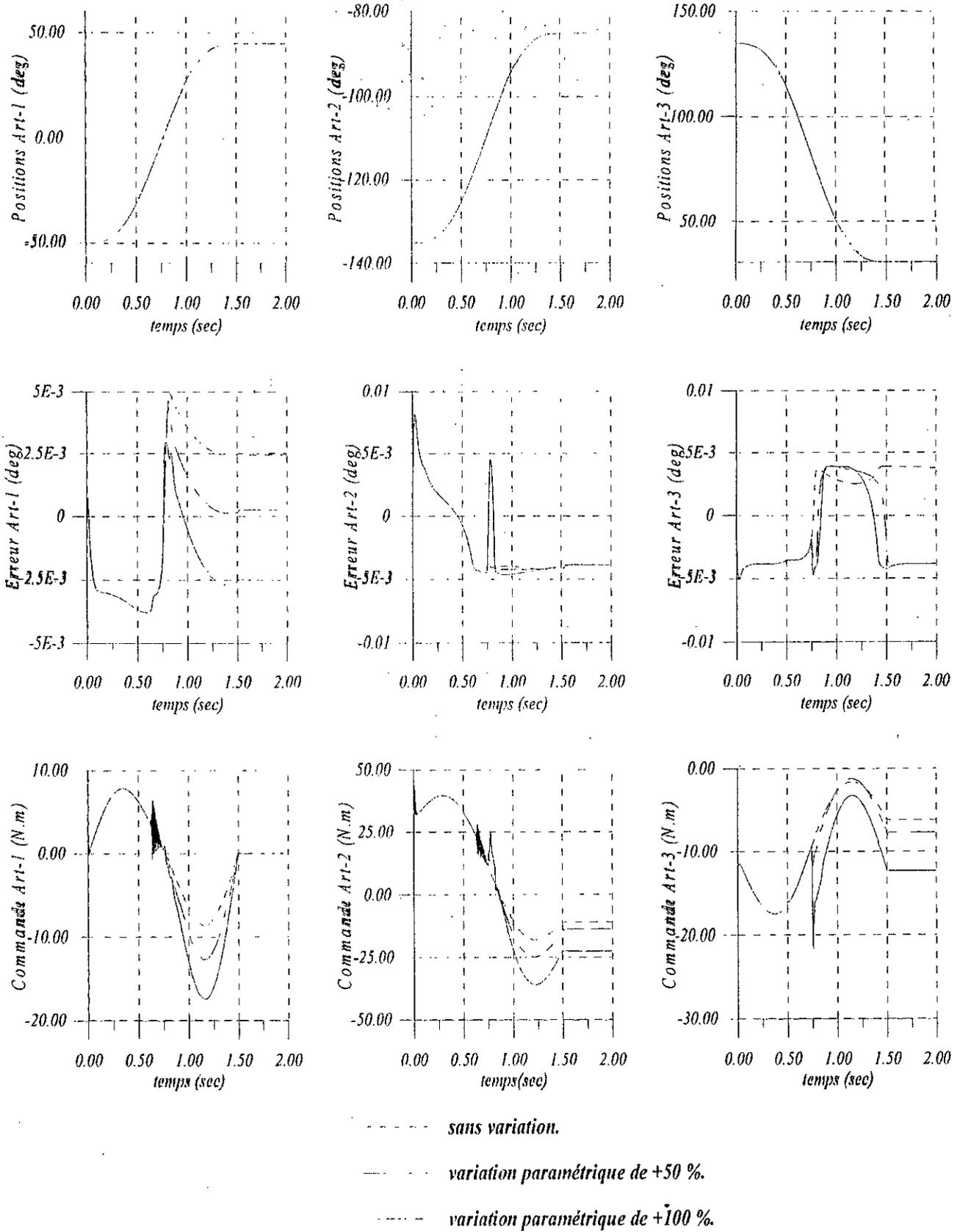


figure (III.35) : Réponses des différentes articulations soumises à la loi CDFAZM :
Test de variation de charge



**figure (III.36) : Réponses des différentes articulations soumises à la loi CDFAZM :
Test de variation paramétrique**

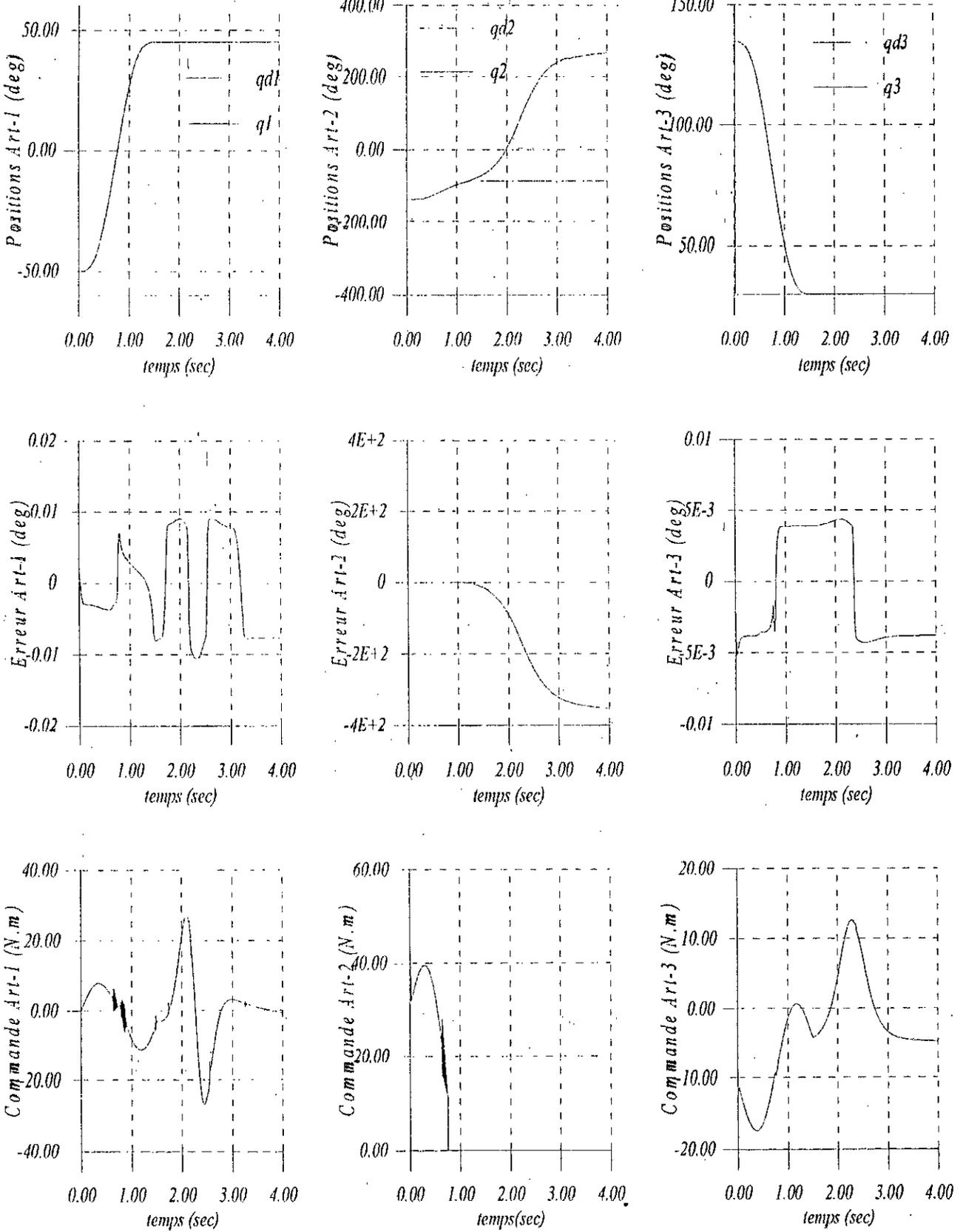


figure (III.37) : Réponses des différentes articulations soumises à la loi CDFAZM :
Test de rupture de commande

Chapitre IV

*Commande décentralisée floue
adaptative discrète*

Chapitre IV :

Commande décentralisée floue adaptative discrète

IV.1. Introduction

Dans plusieurs applications industrielles le problème de commande des robots manipulateurs, en particulier lorsque des connaissances sur la structure, les paramètres et les états du modèle dynamique du robot ne sont pas suffisantes, est d'importance primordiale. Dans les schémas de commande basés sur une connaissance du modèle complet du robot ainsi que sur la disponibilité des mesures de tous les états du système souffrent de sérieuses limitations lors de l'implémentation [SER89][COL95].

Il est souvent irréaliste de supposer que tous les états du robots sont mesurables ou qu'ils sont mesurables avec une bonne précision. En effet, tandis que les mesures de positions peuvent être obtenues avec une grande précision, celles des vitesses sont typiquement entachées par des bruits voir même dans certaines installations ne sont plus disponibles [SLO88][COL95].

Pour palier à ces limitations, une importante motivation a connue naissance pour développer des stratégies de commande permettant d'assurer une bonne poursuite des trajectoires désirées malgré d'éventuelles incertitudes sur le modèle dynamique en utilisant seulement les mesures de positions [NIC90][YUA91][COL95].

Dans le but d'éviter l'utilisation des mesures de vitesses, une première approche proposée consiste à synthétiser un observateur pour estimer les vitesses. Pour obtenir des bonnes performance, cet observateur doit être alimenté par des mesures de positions de haute qualité [CAN92][COL95]. Les variables de vitesses générées par l'observateur sont utilisées dans la loi de commande à la place des vitesses réelles. Dans une autre alternative, on a montré qu'un schéma de commande adéquat peut être obtenu par combinaison de blocs d'intervention directe ("*feedforward*") avec un bloc de dynamique linéaire du premier ordre pour la stabilisation en retour ("*feedback*") [BER93]. Il est à noter que ce type de schéma de commande n'exige pas la mesure de la vitesse cependant une connaissance précise du modèle dynamique du robot lors de l'implémentation où lorsqu'on cherche une meilleure robustesse vis-à-vis les incertitudes sur le modèle est nécessaire.

D'autre part, la stratégie des commandes adaptatives constitue un outil efficace pour la commande des systèmes variables et/ou mal connus [SER89]. En effet, les schémas de commande de cette stratégie sont dotés de la propriété d'adaptation qui permet de palier aux incertitudes et à d'éventuelles variations paramétriques, ce qui est intéressant dans le cas des robots manipulateurs où nous constatons des changements d'inertie et d'effet gravitationnel significatifs. La plupart des travaux de recherche sur la commande adaptative sont consacrés au développement de schémas de commande continue [SLO88][SER89][SIN85]. Cependant dans la pratique les robots manipulateurs sont contrôlés par des calculateurs numériques qui effectuent une variation de la commande à des instants définis par un pas d'échantillonnage. Une implémentation numérique d'un algorithme de commande continue peut conduire à une dégradation des performances du système global voir même à le rendre instable en particulier lorsque le pas d'échantillonnage n'est pas choisit suffisamment petit [TAR89]. Ainsi, le développement d'un schéma adaptatif discret est fort intéressant dès qu'il permet d'aboutir à un

algorithme de commande à implémentation directe sur ordinateur. De plus, une formulation adéquate du problème peut conduire à une loi de commande où seules les mesures successives des positions sont nécessaires.

Dans ce chapitre, nous proposons deux approches de commande décentralisée adaptative discrète en utilisant les systèmes flous applicables aux robots manipulateurs. Dans chaque articulation, la loi de commande est composée d'un compensateur linéaire, d'un modèle flou adaptatif et d'un signal auxiliaire.

Dans la première approche, nous développons une loi d'adaptation à gain constant pour ajuster les paramètres adaptatifs de la commande. La stabilité et la bornitude des différentes variables sont analysées via la théorie de Lyapunov dans le cas discret. Une loi d'adaptation à gain décroissant est proposée dans la deuxième loi de commande. L'analyse du schéma de commande est effectuée en utilisant la version discrète de l'hyperstabilité.

IV.2. Modèle discret décentralisé du robot manipulateur

Dans les schémas de commande discrète des robots manipulateurs, les entrées du régulateur sont la trajectoire désirée et la trajectoire réelle représentées respectivement par les deux vecteurs $q^d(k) \in \mathbb{R}^n$ et $q(k) \in \mathbb{R}^n$ $k = 1, 2, \dots$. En général, les séquences $q(k)$ et $q^d(k)$ sont respectivement, délivrées par des capteurs de position et un planificateur de trajectoire. D'autre part, elles peuvent être considérées comme des valeurs échantillonnées des variables continues $q(t)$ et $q^d(t)$ aux instants équidistants d'intervalle T_0 ($t = kT_0$). La sortie du régulateur est un vecteur $u(k) \in \mathbb{R}^n$ qui sera maintenu constant, par un bloqueur, pour délivrer une commande constante $u(t)$ pendant l'intervalle $kT_0 \leq t \leq (k+1)T_0$.

IV.2.1. Modèle Discret Centralisé :

Dans le but de développer un schéma de commande discrète, il est nécessaire d'établir un modèle discret qui décrit la relation entre le vecteur d'état $q(k)$ et la commande $u(k)$. Deux approches principales peuvent être utilisées :

1° Première approche : Elle consiste à concevoir un système d'équations discrètes qui respecte les principes fondamentaux de la mécanique classique (Neuman et Touris [NEU85]) où les quantités physiques : l'énergie et les moments sont conservées aux instants d'échantillonnage. L'ensemble des équations développées sont mise sous la forme " intégrale de mouvement ". Pour obtenir le modèle discret, il est nécessaire d'appliquer des méthodes d'intégrations. Seules les méthodes numériques itératives sont applicables [TAR89] [NEU85].

2° Deuxième approche : Dans cette approche, le modèle discret est obtenu par discrétisation du modèle continu. Considérons le modèle continu sous la forme :

$$M(q)\ddot{q} + B(q, \dot{q})\dot{q} + K(q, \dot{q})q + G(q) + H(\dot{q}) = u(t) \quad (IV.1)$$

où $q \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$ et $\ddot{q} \in \mathbb{R}^n$, sont respectivement, les vecteur des positions, des vitesses et d'accélération.

$M(q)$ ($\in \mathbb{R}^{n \times n}$) : matrice d'inertie;

$B(q, \dot{q})\dot{q} + K(q, \dot{q})q = N(q, \dot{q})$ ($\in \mathbb{R}^n$) : vecteur de forces et/ou couples centrifuges et de Coriolis ;

$G(q)$ ($\in \mathbb{R}^n$) : vecteur des forces et/ou couples de gravitation;

$H(\dot{q})$ ($\in \mathbb{R}^n$) : vecteur des forces et/ou couples de frottements;

$u(t)$ ($\in \mathbb{R}^n$) : vecteur des forces et/ou couples appliqués au robot .

Pendant le mouvement des articulations, les variables d'états et les paramètres physiques du robot sont généralement des fonctions de temps. Donc le modèle (IV.1) s'écrit :

$$M(t)\ddot{q}(t) + B(t)\dot{q}(t) + K(t)q(t) + GH(t) = u(t) \quad (IV.2)$$

avec $GH(t) = G(t) + H(t)$

Une forme discrète du modèle précédent peut être exprimée sous la forme du système d'équations aux différences suivant [TAR89] :

$$C_0(k, T_0)q(k) + C_1(k, T_0)q(k-1) + C_2(k, T_0)q(k-2) + C_3(k, T_0) = u(k) \quad (IV.3)$$

Pour que le modèle discret donné par (IV.3), reflète le même comportement décrit par le modèle continu (IV.2), les réponses des deux modèles, aux instants d'échantillonnage, doivent être identiques. Cela peut être vérifiée par le biais des deux techniques de discrétisation suivantes [TAR89]:

Discretisation Exacte (Implicite) : Dans cette technique, on établit une représentation d'état du modèle continu à partir de laquelle on détermine la réponse aux instants d'échantillonnage.

Les représentations dans l'espace d'état des deux systèmes continus et discret, donnée par (IV.2) et (IV.3) peut s'écrire comme suit :

$$\dot{X} = D_1(t)X(t) + D_2(t)u(t) + D_3(t) \quad (IV.4a)$$

$$X(k+1) = E_\phi(k, T_0)X(k) + F_\phi(k, T_0)u(k) + N_\phi(k, T_0) \quad (IV.4b)$$

parmi les choix possible, on peut prendre [TAR89]: $X = [q^T(t) \quad \dot{q}^T(t)]^T$. Dans ce cas particulier, on obtient :

$$D_1(t) = \begin{bmatrix} O_{n \times n} & I_n \\ -M^{-1}(t)B(t) & -M^{-1}(t)K(t) \end{bmatrix}; D_2(t) = \begin{bmatrix} O_{n \times n} \\ M^{-1}(t) \end{bmatrix}; D_3(t) = \begin{bmatrix} O_{n \times 1} \\ M^{-1}(t)GH(t) \end{bmatrix} \quad (IV.5)$$

Dans le modèle continu les matrices $D_1(t) \in \mathbb{R}^{2n \times 2n}$, $D_2(t) \in \mathbb{R}^{2n \times n}$ et $D_3(t) \in \mathbb{R}^{2n \times 1}$ sont liées aux matrices $M(t)$, $K(t)$, $B(t) \in \mathbb{R}^{n \times n}$ et $GH(t) \in \mathbb{R}^{n \times 1}$. De même, dans le modèle discret, les matrices $E_\phi(k, T_0) \in \mathbb{R}^{2n \times 2n}$, $F_\phi(k, T_0) \in \mathbb{R}^{2n \times n}$ et $N_\phi(k, T_0) \in \mathbb{R}^{2n \times 1}$ sont liées aux matrices $D_2(k, T_0)$, $D_1(k, T_0)$, $D_3(k, T_0) \in \mathbb{R}^{n \times n}$.

Pour que les deux systèmes (IV.4a) et (IV.4b) possèdent les mêmes réponses aux instants d'échantillonnage, il faut que :

$$E_\phi(k, T_0) = \Phi_T((k-1)T_0, kT_0) \quad (IV.6a)$$

$$F_\phi(k, T_0) = \int_{(k-1)T_0}^{kT_0} \Phi_T((k-1)T_0, \tau) D_2(\tau) d\tau \quad (IV.6b)$$

$$N_\phi(k, T_0) = \int_{(k-1)T_0}^{kT_0} \Phi_T((k-1)T_0, \tau) D_3(\tau) d\tau \quad (IV.6c)$$

avec, $\Phi(\cdot, \cdot)$: matrice de transition du système continu, elle est définie par:

$$\Phi_T(t_1, t_2) = \exp(D_1(t_1 - t_2)) \quad (IV.6d)$$

Dans le modèle continu les matrices $D_1(t) \in \mathbb{R}^{2n \times 2n}$, $D_2(t) \in \mathbb{R}^{2n \times n}$ et $D_3(t) \in \mathbb{R}^{2n \times 1}$ sont liées aux matrices $M(t)$, $K(t)$, $B(t) \in \mathbb{R}^{n \times n}$ et $GH(t) \in \mathbb{R}^{n \times 1}$.

De même, dans le modèle discret, les matrices $E_\phi(k, T_0) \in \mathbb{R}^{2n \times 2n}$, $F_\phi(k, T_0) \in \mathbb{R}^{2n \times n}$ et $N_\phi(k, T_0) \in \mathbb{R}^{2n \times 1}$ sont liées aux matrices $C_2(k, T_0)$, $C_1(k, T_0)$, $C_0(k, T_0) \in \mathbb{R}^{n \times n}$.

Discretisation Approximative (Explicite) : une relation explicite entre le modèle continu et le modèle discret peut être établi en utilisant les approximations suivantes :

$$\dot{q}(t) \cong \frac{q(k) - q(k-1)}{T_0}, \quad \ddot{q}(t) \cong \frac{q(k) - 2q(k-1) + q(k-2)}{T_0^2} \quad (IV.7a)$$

En remplaçant les relations (IV.6) dans le modèle continu donné par (IV.2), on obtient l'équation du modèle discret donnée par (IV.3), avec :

$$\begin{cases} C_0(k, T_0) = K(k) - \frac{B(k)}{T_0} + \frac{M(k)}{T_0^2} \\ C_1(k, T_0) = \frac{B(k)}{T_0} - 2\frac{M(k)}{T_0^2} \\ C_2(k, T_0) = \frac{M(k)}{T_0^2} \\ C_3(k, T_0) = GH(k) \end{cases} \quad (IV.7b)$$

Pour que l'approximation (IV.7a) et par conséquent les relations données par (IV.7b) soient valides, il faut choisir la période d'échantillonnage suffisamment petite.

Il est à noter que la relation entre les matrices du modèle continu et du modèle discret sous la forme implicite exacte (IV.4a) (IV.6a,b,c,d) ou sous la forme explicite approximative (IV.7a,b) n'est pas nécessaire dans le présent travail. Cependant, elle permet d'une part de justifier la représentation de la dynamique du robot sous la forme du modèle discret donné par (IV.3) et d'autre part de déduire des propriétés structurelles à partir des propriétés du modèle continu [TAR89].

Puisque la période d'échantillonnage est constante, il est convenable d'écrire l'expression (IV.3) sous la forme suivante :

$$C_0(k)q(k) + C_1(k)q(k-1) + C_2(k)q(k-2) + C_3(k) = u(k) \quad (IV.8)$$

L'équation (IV.8) décrit le comportement dynamique du robot manipulateur aux instants d'échantillonnage. Comme tous modèles mathématiques, cette équation traduit une description approximative du comportement réel du robot. Cependant, elle constitue une base pour le développement de lois de commande discrètes.

IV.2.2. Décentralisation du modèle discret :

Dans le but d'établir des lois de commande décentralisée pour un robot manipulateur, il est convenable de considérer chaque articulation comme un sous-système indépendant. Pour cela, il est nécessaire de trouver pour chaque articulation un modèle dynamique indépendant où on établit la relation entre les états locaux et la commande correspondante. De ce fait, il apparaîtra des termes de perturbation qui reflètent les interconnexions avec les autres articulations.

Le modèle dynamique discret décentralisé peut être déduit directement à partir de l'équation (IV.8). On aura :

$$c_{0ii}(k)q_i(k) + c_{1ii}(k)q_i(k-1) + c_{2ii}(k)q_i(k-2) + c_{3ii}(k) = u_i(k) - \left[\sum_{\substack{j=1 \\ j \neq i}}^n c_{0ij}(k)q_j(k) + c_{1ij}(k)q_j(k-1) + c_{2ij}(k)q_j(k-2) \right] \quad (IV.9)$$

avec, $c_{ij}(k) = [C_i(k)]_{ij}$: l'élément de la $i^{ème}$ ligne et la $j^{ème}$ colonne de la matrice $C_i(k)$

Le modèle décentralisé donné par (IV.9), décrit la dynamique d'un système linéaire discret du deuxième ordre variant dans le temps. La somme entre crochets du deuxième membre peut être considérée comme une perturbation.

IV.3. Synthèse de lois de commande :

La synthèse d'une loi de commande discrète peut s'effectuer dans deux étapes principales : Le choix de la structure de la loi de commande et la détermination de cette loi de manière à assurer les

performances désirées : stabilité du schéma de commande et bornitude des signaux adaptatifs (aux instants d'échantillonnage).

Après la formulation du problème de poursuite de trajectoires, nous proposons une loi de commande décentralisée floue adaptative discrète. Elle est composée de trois actions principales : une action de stabilisation via un régulateur linéaire et une action de compensation des termes variables dans le temps réalisée par un modèle flou de Sugeno adaptatif, et une troisième action de compensation des erreurs de reconstruction via un signal auxiliaire.

Selon la nature des lois d'adaptation des paramètres adaptatifs, nous distinguons :

- une loi de commande avec matrice et gain d'adaptation constants;
- une loi de commande avec matrice et gain d'adaptation décroissants.

La stabilité du schéma de commande et la bornitude des signaux adaptatifs seront analysées.

IV.3.1. Formulation du problème de poursuite :

Soit le modèle dynamique discret de la $i^{\text{ème}}$ articulation donné par (IV.9), écrit sous la forme :

$$\delta_{oi}q_i(k) = -[\delta_{oi} + c_{oii}(k)]q_i(k) - c_{1ii}(k)q_i(k-1) - c_{2ii}(k)q_i(k-2) - [c_{3ii}(k) + \sum_{\substack{j=1 \\ j \neq i}}^n c_{oij}(k)q_j(k) + c_{1ij}(k)q_j(k-1) + c_{2ij}(k)q_j(k-2)] + u_i(k) \quad i = 1 \dots n \quad (IV.10a)$$

avec δ_{oi} un scalaire constant non nul.

Le problème de poursuite consiste à déterminer une loi de commande $u_i(k)$, qui assure la convergence asymptotique vers zéro de l'erreur de poursuite $e_i(k)$ entre la trajectoire réelle $q_i(k)$ et la trajectoire désirée $q_i^d(k)$. D'après l'équation (IV.10a), on peut écrire :

$$\delta_{oi}e_i(k) = [-\delta_{oi} + c_{oii}(k)]q_i(k) + c_{1ii}(k)q_i(k-1) + c_{2ii}(k)q_i(k-2) + [c_{3ii}(k) + \sum_{\substack{j=1 \\ j \neq i}}^n c_{oij}(k)q_j(k) + c_{1ij}(k)q_j(k-1) + c_{2ij}(k)q_j(k-2)] + \delta_{oi}q_i^d(k) - u_i(k) \quad i = 1 \dots n \quad (IV.10b)$$

avec $e_i(k) = q_i^d(k) - q_i(k)$

Notre hypothèse ici, consiste à considérer que les termes du deuxième membre de l'équation (IV.10b) s'écrivent sous la forme d'une somme de deux termes. Un premier terme $f_i(q_i^d(k), q_i(k), q_i(k-1), q_i(k-2))$ approximable par un modèle flou de type Sugeno (MFS) et un deuxième terme $\varepsilon_i(k)$ non-approximable par le MFS. Dans ce cas l'expression (IV.10b) s'écrit :

$$\delta_{oi}e_i(k) = f_i(q_i^d(k), q_i(k), q_i(k-1), q_i(k-2)) + \varepsilon_i(k) - u_i(k) \quad i = 1 \dots n \quad (IV.11a)$$

avec,

$$f_i(q_i^d(k), q_i(k), q_i(k-1), q_i(k-2)) + \varepsilon_i(k) = [-\delta_{oi} + c_{oii}(k)]q_i(k) + c_{1ii}(k)q_i(k-1) + c_{2ii}(k)q_i(k-1) + [c_{3ii}(k) + \sum_{\substack{j=1 \\ j \neq i}}^n c_{oij}(k)q_j(k) + c_{1ij}(k)q_j(k-1) + c_{2ij}(k)q_j(k-2)] + \delta_{oi}q_i^d(k) \quad (IV.11b)$$

D'autre part, considérons un MFS à quatre variables d'entrées $q_i^d(k), q_i(k), q_i(k-1)$ et $q_i(k-2)$. Soit nf^i le nombre des fonctions d'appartenance associées à la variable $q_i(k)$. Ces fonctions seront notées (μ_l^i) $l = 1 \dots nf^i, i = 1 \dots n$. Donc le nombre des règles floues (m^i) sera donné par $m^i = (nf^i)^4$.

Remarque :

Nous avons choisi le même nombre de fonctions d'appartenance pour les variables $q_i^d(k), q_i(k), q_i(k-1)$ et $q_i(k-2)$, car elle représente la même grandeur physique.

En utilisant ce modèle on peut écrire :

$$f_i(q_i^d(k), q_i(k), q_i(k-1), q_i(k-2)) = Z_i^T A_i^* \xi^i + \varepsilon_{f_i} \quad (IV.12a)$$

où, le premier terme $Z_i^T A_i^* \xi^i$ représente la sortie du MFS, avec :

$$Z_i^T = \begin{cases} 1 & \text{si MFS est d'ordre 0} \\ \begin{bmatrix} 1 & q_i^d(k) & q_i(k) & q_i(k-1) & q_i(k-2) \end{bmatrix} & \text{si MFS est d'ordre 1.} \end{cases} \quad (IV.12b)$$

A_i^* : une matrice constante qui regroupe les gains des conséquences des règles floues. Elle est définie par :

$$A_i^* = \arg \min_{A_i \in \Omega_a^i} \left[\sup_{q_i^d(k), q_i(k), q_i(k-1), q_i(k-2) \in \Omega_i^i} \left| Z_i^T A_i \xi^i - f_i(q_i^d(k), q_i(k), q_i(k-1), q_i(k-2)) \right| \right] \quad (IV.12c)$$

avec Ω_a^i, Ω_i^i sont, respectivement, les domaines admissibles des variables A_i et q_i , et :

$$A_i = \begin{cases} \begin{bmatrix} a_{01}^i & \dots & a_{0m^i}^i \end{bmatrix} & \text{si MFS est d'ordre 0} \\ \begin{bmatrix} a_{01}^i & \dots & a_{0m^i}^i \\ a_{11}^i & \dots & a_{1m^i}^i \\ a_{21}^i & \dots & a_{2m^i}^i \\ a_{31}^i & \dots & a_{3m^i}^i \\ a_{41}^i & \dots & a_{4m^i}^i \end{bmatrix} & \text{si MFS est d'ordre 1.} \end{cases} \quad (IV.12d)$$

ξ^i : vecteur des degrés d'activation des différentes règles floues. Il est donné par :

$$\xi^i = \begin{bmatrix} \xi_{s_1}^i & \dots & \xi_{s_{m^i}}^i \end{bmatrix}^T \quad (IV.12e)$$

$$\text{et } \xi_{s_s}^i = \frac{\mu_{s_1}^i(q_i^d(k)) \times \mu_{s_2}^i(q_i(k)) \times \mu_{s_3}^i(q_i(k-1)) \times \mu_{s_4}^i(q_i(k-2))}{\sum_{1 \leq l_1, l_2, l_3, l_4 \leq n_f^i} \mu_{l_1}^i(q_i^d(k)) \times \mu_{l_2}^i(q_i(k)) \times \mu_{l_3}^i(q_i(k-1)) \times \mu_{l_4}^i(q_i(k-2))}; \quad s = \sum_{j=2}^4 \left[(s_j - 1) \prod_{j=1}^{j-1} (n_f^i)^{j'} \right] + s_1$$

avec, $s_1, s_2, s_3, s_4 = 1 \dots n_f^i$.

Le deuxième terme ε_{f_i} est une constante qui représente l'erreur résultante de l'approximation de f_i par le MFS avec un nombre limité de fonctions d'appartenance.

En remplaçant l'équation (IV.12a) dans (IV.11a), on peut déduire l'équation de la dynamique de l'erreur $e_i(k)$ donnée par :

$$\delta_{0i} e_i(k) = Z_i^T A_i^* \xi^i + (\varepsilon_{f_i} + \varepsilon_i(k)) - u_i(k) \quad (IV.13)$$

IV.3.2. Structure de commande

A partir de l'équation (IV.13), nous proposons la loi de commande suivante :

$$u_i(k) = \delta_{1i} e_i(k-1) + \delta_{2i} e_i(k-2) + Z_i^T \hat{A}_i(k) \xi^i + \hat{\varepsilon}_i(k) \quad (IV.14)$$

avec,

δ_{1i}, δ_{2i} : des constantes qui seront déterminées ultérieurement ;

$\hat{A}_i(k)$: une matrice estimée de la matrice A_i^* à l'instant $t = kT_0$;

$\hat{\varepsilon}_i(k)$: signal auxiliaire pour compenser l'erreur de reconstruction ($\varepsilon_{f_i} + \varepsilon_i(k)$).

En remplaçant la loi de commande (IV.14) dans l'équation (IV.13), on obtient l'expression de la dynamique de l'erreur en boucle fermée suivante :

$$e_i(k) = -\frac{\delta_{1i}}{\delta_{0i}} e_i(k-1) - \frac{\delta_{2i}}{\delta_{0i}} e_i(k-2) + \frac{1}{\delta_{0i}} (Z_i^T \tilde{A}_i(k) \xi^i + \tilde{\varepsilon}_i(k)) \quad (IV.15)$$

où, $\tilde{A}_i(k) = A_i^* - \hat{A}_i(k)$: erreur d'estimation de la matrice A_i^* à l'instant $t = kT_0$;

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

On définit le vecteur des erreurs $E_i(k)$ tel que :

$$E_i(k) = [e_i(k) \quad e_i(k-1)]^T \quad (IV.16)$$

dans ce cas la dynamique de l'erreur décrite par (IV.15) s'écrit sous la forme matricielle suivante :

$$E_i(k) = S_i E_i(k-1) + b_i \tilde{\Phi}(k) \quad (IV.17a)$$

où,

$$S_i = \begin{bmatrix} -\frac{\delta_{1i}}{\delta_{0i}} & -\frac{\delta_{2i}}{\delta_{0i}} \\ I & 0 \end{bmatrix}; \quad b_i = \begin{bmatrix} 1 \\ \delta_{0i} \\ 0 \end{bmatrix}; \quad \tilde{\Phi}(k) = Z_i^T \tilde{A}_i(k) \xi^i + \tilde{\varepsilon}_i(k) \quad (IV.17b)$$

Dès que la matrice A_i^* est inconnue et l'erreur de reconstruction $(\varepsilon_{f_i} + \varepsilon_i(k))$ est variable dans le temps, il est nécessaire d'adopter une loi d'adaptation pour $\hat{A}_i(k)$ et $\hat{\varepsilon}_i(k)$. Cette loi doit assurer la convergence asymptotique vers zéro de l'erreur de poursuite $e_i(k)$ ainsi que la bornitude des paramètres adaptatifs et l'état $q_i(k)$. Pour cela, on doit considérer certains hypothèses.

Hypothèses :

H1. Les trajectoires désirées $q_i^d(k) \quad i = 1 \dots n$ sont bornées;

H2. La variation de l'erreur de reconstruction $\varepsilon_{f_i} + \varepsilon_i(k)$ entre deux instants d'échantillonnage est négligeable par rapport à la variation du signal auxiliaire $\hat{\varepsilon}_i(k)$.

IV.3.3. Loi d'adaptation à gain constant :

IV.3.3.1. Structure de la loi d'adaptation

Proposons les lois d'adaptation suivante :

$$\hat{A}_i(k) = \hat{A}_i(k-1) + \Gamma_i b_i^T P_i (E_i(k) + S_i E_i(k-1)) Z_i \xi^{iT} \quad (IV.18a)$$

$$\hat{\varepsilon}_i(k) = \hat{\varepsilon}_i(k-1) + \gamma_i b_i^T P_i (E_i(k) + S_i E_i(k-1)) \quad (IV.18b)$$

avec, Γ_i : une matrice constante symétrique définie positive;

γ_i : un scalaire constant positif;

P_i : matrice symétrique définie positive, solution de l'équation de Riccati discrète suivante :

$$S_i^T P_i S_i - P_i = -Q_i \quad (IV.19)$$

où Q_i est une matrice constante arbitraire symétrique définie positive.

Proposition 1 :

Etant donné un robot manipulateur à n degrés de liberté dont le modèle dynamique discret de chaque articulation peut être mis sous la forme (IV.13).

Si on choisit la loi de commande (IV.14) et en tenant compte des hypothèses H1, H2 et les lois d'adaptation (IV.18a,b) on aura :

R1 : Les états $q_i(k) \quad i = 1 \dots n$ sont bornés;

R2 : Les variables $\hat{A}_i(k)$ et $\hat{e}_i(k)$ et les commandes $u_i(k)$ ($i = 1 \dots n$) sont bornées;

R3 : Les erreurs $e_i(k)$ $i = 1 \dots n$ sont bornées et convergent asymptotiquement vers zéro.

Démonstration

Pour démontrer la proposition 1, nous utilisons la méthode directe de Lyapunov dans le cas discret. Chaque articulation est considérée comme un sous-système indépendant. Ainsi pour simplifier l'écriture, on omit l'indice "i" de l'articulation.

Considérons la fonction (ou la suite) de Lyapunov discrète suivante :

$$V(k) = \frac{1}{2} E^T(k) P E(k) + \frac{1}{2} \text{Tr}(\tilde{A}^T(k) \Gamma_o^{-1} \tilde{A}(k)) + \frac{1}{2} \gamma_o^{-1} \tilde{\varepsilon}^2(k) \quad (IV.20)$$

où, Γ_o^{-1} : une matrice définie positive ;

γ_o^{-1} : une constante positive;

P : une matrice symétrique définie positive qui vérifie la relation (IV.19) (avec $P_i = P$);

La suite $V(k)$ est positive. Donc, pour assurer la convergence asymptotique vers zéro du vecteur d'erreur $E(k)$, de la matrice $\tilde{A}(k)$ et de l'erreur $\tilde{\varepsilon}(k)$, il suffit de montrer que la suite $V(k)$ est décroissante.

La différence entre deux termes consécutifs de $V(k)$ s'écrit :

$$\begin{aligned} \Delta V(k) = V(k) - V(k-1) &= \frac{1}{2} E^T(k) P E(k) + \frac{1}{2} \text{Tr}(\tilde{A}^T(k) \Gamma_o^{-1} \tilde{A}(k)) + \frac{1}{2} \gamma_o^{-1} \tilde{\varepsilon}^2(k) \\ &\quad - \frac{1}{2} E^T(k-1) P E(k-1) - \frac{1}{2} \text{Tr}(\tilde{A}^T(k-1) \Gamma_o^{-1} \tilde{A}(k-1)) - \frac{1}{2} \gamma_o^{-1} \tilde{\varepsilon}^2(k-1) \end{aligned} \quad (IV.21)$$

En utilisant l'expression $E(k)$ donnée par (IV.17a), on obtient

$$\begin{aligned} \bullet \frac{1}{2} E^T(k) P E(k) - \frac{1}{2} E^T(k-1) P E(k-1) &= \frac{1}{2} E^T(k-1) (S^T P S - P) E(k-1) \\ &\quad + E^T(k-1) S^T P b \tilde{\Phi}(k) + \frac{1}{2} \tilde{\Phi}^T(k) b^T P b \tilde{\Phi}(k) \end{aligned} \quad (IV.22)$$

et d'autre part :

$$\begin{aligned} \bullet \frac{1}{2} \text{Tr}(\tilde{A}^T(k) \Gamma_o^{-1} \tilde{A}(k)) - \frac{1}{2} \text{Tr}(\tilde{A}^T(k-1) \Gamma_o^{-1} \tilde{A}(k-1)) &= \\ - \frac{1}{2} \text{Tr} \left[(\tilde{A}(k) - \tilde{A}(k-1))^T \Gamma_o^{-1} (\tilde{A}(k) - \tilde{A}(k-1)) \right] &+ \text{Tr}(\tilde{A}^T(k) \Gamma_o^{-1} \tilde{A}(k)) \\ - \text{Tr}(\tilde{A}^T(k) \Gamma_o^{-1} \tilde{A}(k-1)) & \end{aligned} \quad (IV.23)$$

d'où,

$$\begin{aligned} \frac{1}{2} \text{Tr}(\tilde{A}^T(k) \Gamma_o^{-1} \tilde{A}(k)) - \frac{1}{2} \text{Tr}(\tilde{A}^T(k-1) \Gamma_o^{-1} \tilde{A}(k-1)) &= \\ - \frac{1}{2} \text{Tr}(\Delta \tilde{A}^T(k) \Gamma_o^{-1} \Delta \tilde{A}(k)) + \text{Tr}(\tilde{A}^T(k) \Gamma_o^{-1} \Delta \tilde{A}(k)) & \end{aligned} \quad (IV.24)$$

avec $\Delta \tilde{A}(k) = \tilde{A}(k) - \tilde{A}(k-1)$

$$\bullet \frac{1}{2} \gamma_o^{-1} \tilde{\varepsilon}^2(k) - \frac{1}{2} \gamma_o^{-1} \tilde{\varepsilon}^2(k-1) = \frac{-1}{2} \gamma_o^{-1} (\Delta \tilde{\varepsilon}(k))^2 + \gamma_o^{-1} \Delta \tilde{\varepsilon}(k) \tilde{\varepsilon}(k) \quad (IV.25)$$

avec $\Delta\tilde{\varepsilon}(k) = \tilde{\varepsilon}(k) - \tilde{\varepsilon}(k-1)$.

D'après (IV.22), (IV.24) et (IV.25) l'expression de la différence $\Delta V(k)$ devient :

$$\begin{aligned} \Delta V(k) = & -\frac{1}{2}E^T(k-1)QE(k-1) - \frac{1}{2}\text{Tr}(\Delta\tilde{A}^T(k)\Gamma_o^{-1}\Delta\tilde{A}(k)) - \frac{1}{2}\gamma_o^{-1}\Delta\tilde{\varepsilon}^2(k) \\ & + \frac{1}{2}E^T(k-1)S^T Pb\tilde{\Phi}(k) + \frac{1}{2}\tilde{\Phi}^T(k)b^T Pb\tilde{\Phi}(k) \\ & + \frac{1}{2}E^T(k-1)S^T Pb\tilde{\Phi}(k) + \text{Tr}(\tilde{A}^T(k)\Gamma_o^{-1}\Delta\tilde{A}(k)) + \gamma_o^{-1}\Delta\tilde{\varepsilon}(k)\tilde{\varepsilon}(k) \end{aligned} \quad (\text{IV.26})$$

avec, Q matrice définie par : $S^T P S - P = Q$.

Sachant que $\tilde{\Phi}(k)$ est donnée par (IV.17b), on déduit :

$$\Delta V(k) = \Delta V_1(k) + \Delta V_2(k) + \Delta V_3(k) \quad (\text{IV.27})$$

avec,

$$\Delta V_1(k) = -\frac{1}{2}E^T(k-1)QE(k-1) - \frac{1}{2}\text{Tr}(\Delta\tilde{A}^T(k)\Gamma_o^{-1}\Delta\tilde{A}(k)) - \frac{1}{2}\gamma_o^{-1}\Delta\tilde{\varepsilon}^2(k) \quad (\text{IV.28})$$

$$\Delta V_2(k) = \frac{1}{2}(E^T(k) + E^T(k-1)S^T)PbZ^T\tilde{A}(k)\xi + \text{Tr}(\tilde{A}^T(k)\Gamma_o^{-1}\Delta\tilde{A}(k)) \quad (\text{IV.29})$$

$$\Delta V_3(k) = \frac{1}{2}(E^T(k) + E^T(k-1)S^T)Pb\tilde{\varepsilon}(k) + \gamma_o^{-1}\Delta\tilde{\varepsilon}(k)\tilde{\varepsilon}(k) \quad (\text{IV.30})$$

On a :

1°/ $\Delta V_1(k) \leq 0$ si Q et Γ_o^{-1} sont choisies deux matrices définies positives et $\gamma_o^{-1} > 0$.

2°/ $\Delta V_2(k) = \text{Tr}\left(\frac{1}{2}(E^T(k) + E^T(k-1)S^T)PbZ^T\tilde{A}(k)\xi\right) + \text{Tr}(\tilde{A}^T(k)\Gamma_o^{-1}\Delta\tilde{A}(k))$

En utilisant les deux relations matricielles :

$$\begin{cases} \text{Tr}(L_1 L_2) = \text{Tr}(L_2 L_1) \\ \text{Tr}(L_1 L_2) = \text{Tr}(L_2^T L_1^T) \end{cases}$$

on obtient

$$\Delta V_2(k) = \text{Tr}\left(\frac{1}{2}\tilde{A}^T(k)Zb^T P(E(k) + SE(k-1))\xi^T + \tilde{A}^T(k)\Gamma_o^{-1}\Delta\tilde{A}(k)\right) \quad (\text{IV.31})$$

Choisissons :

$$\Delta\tilde{A}(k) = -\Gamma Z b^T P(E(k) + SE(k-1))\xi^T \quad \text{avec } \Gamma = \frac{\Gamma_o}{2} \quad (\text{IV.32})$$

on aura : $\Delta V_2(k) = 0$.

3°/ $\Delta V_3(k) = \tilde{\varepsilon}(k)\left(\frac{1}{2}(E^T(k) + E^T(k-1)S^T)Pb + \gamma_o^{-1}\Delta\tilde{\varepsilon}(k)\right)$

Choisissons :

$$\Delta\tilde{\varepsilon}(k) = -\gamma b^T P(E(k) + SE(k-1)) \quad \text{avec } \gamma = \frac{\gamma_o}{2} \quad (\text{IV.33})$$

on obtient : $\Delta V_3(k) = 0$.

Donc, les lois d'adaptation (IV.32), (IV.33) permettent d'écrire :

$$\Delta V(k) = -\frac{1}{2}E^T(k-1)QE(k-1) - \frac{1}{2}\text{Tr}(\Delta\tilde{A}^T(k)\Gamma_o^{-1}\Delta\tilde{A}(k)) - \frac{1}{2}\gamma_o^{-1}\Delta\tilde{\varepsilon}^2(k) \leq 0 \quad (\text{IV.34})$$

d'où la décroissance de la suite $V(k)$. On déduit que :

- le vecteur d'erreur $E(\cdot)$ est borné et converge asymptotiquement vers zéro (conséquence R3 de la proposition);
- les variations $\Delta\tilde{A}(\cdot)$ et $\Delta\tilde{\varepsilon}(\cdot)$ convergent asymptotiquement vers zéro;
- les erreurs d'estimation $\tilde{A}(\cdot)$ et $\tilde{\varepsilon}(\cdot)$ sont bornées.

Selon l'hypothèse H1, la bornitude du vecteur $E(\cdot)$ garantit la bornitude des états $q(\cdot)$ (conséquence R1). Pour déduire les lois d'adaptation, on a d'après la définition :

$$\begin{cases} \Delta\tilde{A}(k) = \hat{A}(k-1) - \hat{A}(k) \\ \Delta\tilde{\varepsilon}(k) = \Delta\varepsilon(k) - \Delta\hat{\varepsilon}(k) \end{cases} \quad (IV.35)$$

mais d'après l'hypothèse H2, on a : $\Delta\varepsilon(k) \ll \Delta\hat{\varepsilon}(k)$, d'où :

$$\Delta\tilde{\varepsilon}(k) = \hat{\varepsilon}(k-1) - \hat{\varepsilon}(k) \quad (IV.36)$$

dans ce cas les relations (IV.32), (IV.33) peuvent s'écrire sous la forme :

$$\begin{cases} \hat{A}(k) = \hat{A}(k-1) + \Gamma b^T P (E(k) + SE(k-1)) Z \xi^T \\ \hat{\varepsilon}(k) = \hat{\varepsilon}(k-1) + \gamma b^T P (E(k) + SE(k-1)) \end{cases} \quad (IV.37)$$

avec,

Γ : matrice constante symétrique définie positive ; γ : constante positive.

Dès que le vecteur d'erreur $E(\cdot)$ et les états $q(\cdot)$ sont bornés, on conclut à partir des relations (IV.14), (IV.37) que la commande $u(k)$ et les estimées $\hat{A}(\cdot)$ et $\hat{\varepsilon}(\cdot)$ sont aussi bornées (conséquence R2).

Schéma de principe :

Dès que la matrice Γ_i et le gain γ_i sont fixes, alors la loi d'adaptation des variables \hat{A}_i et du signal $\hat{\varepsilon}_i$ est à gain constant.

Le schéma de principe de la loi de Commande Décentralisée Floue Adaptative Discrète (CDFAD), donnée par (IV.14) et (IV.18), est illustré par la figure (IV.1).

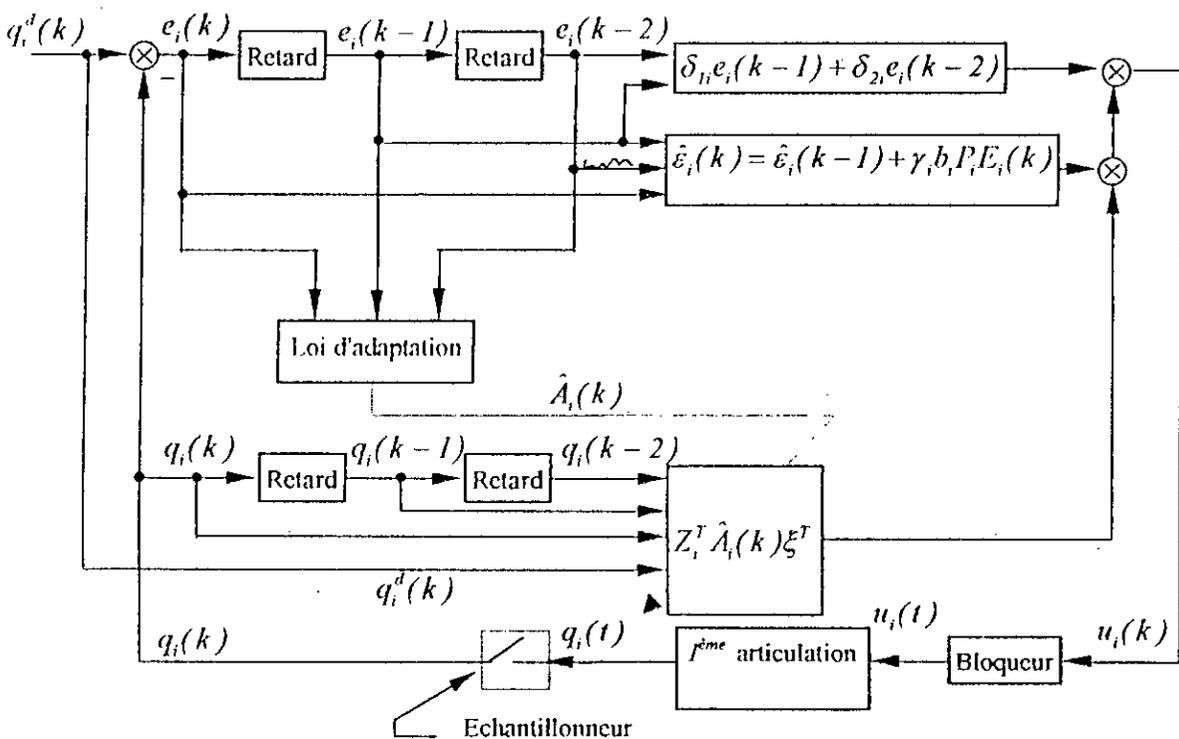


figure (IV.1) : Schéma de principe de la CDFAD à gain constant

La loi de **CDFAD** assure la bornitude et la convergence asymptotique vers zéro des erreurs de poursuite $e_i(k)$ (aux instants d'échantillonnage). La commande $u_i(k)$ de chaque articulation est calculée à partir des mesures locales de positions. Les mesures de la vitesse et de l'accélération ne sont pas nécessaires. Dans l'expression de la commande, on distingue les actions suivantes : une action de stabilisation réalisée via un retour d'état ("Feedback") linéaire avec les paramètres constants δ_{1i} , δ_{2i} . Une deuxième action effectue une estimation de la dynamique liée explicitement aux états de la même articulation. Elle est réalisée par un modèle flou de Sugeno adaptatif. Une troisième action permet de palier aux erreurs de reconstruction. La bornitude des paramètres adaptatifs du modèle flou, du signal auxiliaire et par conséquent de la commande est assurée. Dès que la matrice Γ_i et le gain γ_i sont fixes, alors la loi d'adaptation des paramètres \hat{A}_i et de signal \hat{e}_i est à gain constant. Il faut noter que les paramètres des différentes actions sont liées. En effet, les constantes δ_{1i} , δ_{2i} de l'action de stabilisation définissent la matrice S_i (voir relation (IV.17b)), qui elle-même impose la matrice P_i (voir relation (IV.19)) utilisée dans l'adaptation de la matrice \hat{A}_i et du signal \hat{e}_i .

IV.3.3.2. Application au robot PUMA560

Dans cette section, nous présentons les résultats de l'application de la commande décentralisée floue adaptative discrète à gain constant pour la commande en poursuite du robot PUMA560 à trois degrés de liberté. Le modèle dynamique du robot PUMA560 tel que donné dans l'annexe A est utilisé seulement pour obtenir les réponses. Nous avons choisis la trajectoire de Leahvy[LEA90] de transition cycloïdale pour effectuer les différents tests présentés dans l'annexe A. La simulation est effectuée par la méthode de Runge Kutta d'ordre 4 avec un pas fixe égale à 1 ms.

Paramètres du régulateur :

Chaque articulation est dotée d'un régulateur local indépendant dont le modèle flou de type Sugeno est d'ordre zéro. Deux fonctions d'appartenance triangulaires sont associées à chacune des quatre variables d'entrées ($q_i^d(k)$, $q_i(k)$, $q_i(k-1)$ et $q_i(k-2)$) telles que montrées par la figure (IV.2).

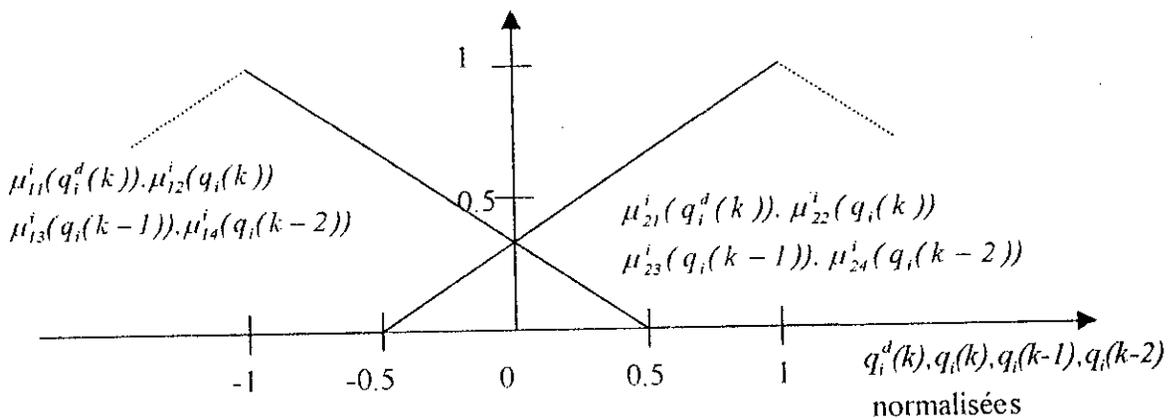


figure (IV.2) : fonctions d'appartenance des variables d'entrées

Ainsi le modèle flou est caractérisé par seize règles.

Les paramètres de la loi de commande sont regroupés dans la table (IV.1)

Avec I_2 la matrice unitaire de dimension 2.

Les différentes variables initiales de la loi de commande sont initialisées par zéro. Le pas d'échantillonnage est fixé à la valeur 5 ms.

Résultats et performances

Test sans charge : pour valider la technique présentée ci-dessus, un test sans charge a été effectué. Nous avons obtenu les résultats montrés par les figures (IV.3a,b). Nous remarquons une bonne poursuite de la trajectoire de Leahvy. Les erreurs maximales sont inférieures à 0.75 deg est convergent asymptotiquement vers zéro. Après un régime transitoire d'adaptation, les trois commandes prennent des allures lisses. Nous constatons que l'intervention des modèles flous est principale dans les commandes (voir leurs sorties notées u_i). Les gains des règles floues sont aussi caractérisée par une variation lisse et convergent vers des valeurs finales constantes.

	Symboles des paramètres	Première articulation i=1	Deuxième articulation i=2	Troisième articulation i=3
	δ_{ni}	137500	137500	137500
	δ_{ii}	27000	27000	27000
	δ_{2i}	-25000	-25000	-25000
	pôles de S_i	-0.5380; 0.3380	-0.5380; 0.3380	-0.5380; 0.3380
	Q_i	$2500 \times I_2$	$2500 \times I_2$	$2500 \times I_2$
	P_i	$\begin{bmatrix} 5499.6 & -244.4 \\ -244.4 & 2681.8 \end{bmatrix}$	$\begin{bmatrix} 5499.6 & -244.4 \\ -244.4 & 2681.8 \end{bmatrix}$	$\begin{bmatrix} 5499.6 & -244.4 \\ -244.4 & 2681.8 \end{bmatrix}$
Facteur de normalisation	$q_i^d(k) : q_i(k) :$	4 : 4	4 : 4	4 : 4
	$q_i(k-1) : q_i(k-2)$	4 : 4	4 : 4	4 : 4
Modèle flou $Z_i^T \hat{A}_i(k) \xi^i$	Γ_i	10000	10000	10000
Signal auxiliaire $\hat{\xi}_i(j)$	γ_i	10	10	10

Table (IV.1) : Paramètres du régulateur sous la loi CDFAD à gain constant

Test de variation de charge : Pour tester les capacités compensatoires de notre régulateur vis-à-vis des variations dynamiques dans le modèle dynamique du robot, nous nous sommes intéressés au test de transport de charge. L'élément terminal porte une masse de 2 Kg puis de 4 Kg j'usqu'à l'instant $t=0.75$ où elle est relâchée. La figure (IV.4) illustre les résultats obtenus. Nous constatons une léger augmentation des erreurs de poursuite en fonction de la masse. Ces erreurs sont maintenues inférieures à 0.76 deg et convergent vers zéro. Ces résultats confirment la robustesse relative du régulateur.

Test de variation paramétrique : Dans le but de tester les capacités d'adaptation du régulateur sujet à des variations paramétriques, nous introduisons une augmentations de tous les paramètres du modèle avec une quantité de 50% puis de 100%. Les réponses enregistrées, montrées sur la figure (IV.5), permettent d'affirmer que notre régulateur fait preuve d'une assez bonne robustesse.

Test de rupture de commande : Pour montrer l'efficacité de l'indépendance du schéma proposé, nous introduisons une rupture de la commande de la deuxième articulation à partir de l'instant $t=0.75$ sec (chute libre). Les résultats obtenus sont illustrés par la figure (IV.6). Nous constatons la robustesse des régulateurs locales des deux autres articulations. En effet, malgré le couplage important entre articulation, la première et troisième articulations continuent remarquablement la poursuite des trajectoires désirées avec des erreurs maximales inférieures à 0.75 sec. Cependant des commandes considérables sont générées afin de compenser la grande perturbation introduite par la rupture.

Test de variation du pas d'échantillonnage: Dans ce test nous avons augmenté le pas d'échantillonnage à la valeur 10 ms. Pour maintenir les performances du système de commande, nous avons diminué les paramètres du régulateur comme indique la table (IV.2) :

Les réponses obtenues illustrées par la figure (IV.7), montrent une poursuite satisfaisante des trajectoires désirées. Une léger augmentation des erreurs de poursuite qui restent inférieures à 0.8 deg et convergent vers zéro. Après un régime transitoire d'adaptation les commandes prennent des allures lisses.

Symboles des paramètres	Première articulation $i=1$	Deuxième articulation $i=2$	Troisième articulation $i=3$
δ_{0i}	50000	83333	41667
δ_{1i}	12000	20000	8000
δ_{2i}	-10700	-17833	-7133
pôles de S_i	-0.5979; 0.3579	-0.5979; 0.3579	-0.5207; 0.3287
Q_i	$2500 \times I_2$	$2500 \times I_2$	$2500 \times I_2$
P_i	$\begin{bmatrix} 5778.7 & -377.6 \\ -377.6 & 2764.6 \end{bmatrix}$	$\begin{bmatrix} 5778.7 & -377.6 \\ -377.6 & 2764.6 \end{bmatrix}$	$\begin{bmatrix} 5443.1 & -215.9 \\ -215.9 & 2659.5 \end{bmatrix}$
Γ_i	10000	10000	10000
γ_i	10	10	10

Table (IV.2) : Paramètres du régulateur sous la loi CDFAD à gain constant avec un pas d'échantillonnage $T_0=10$ ms.

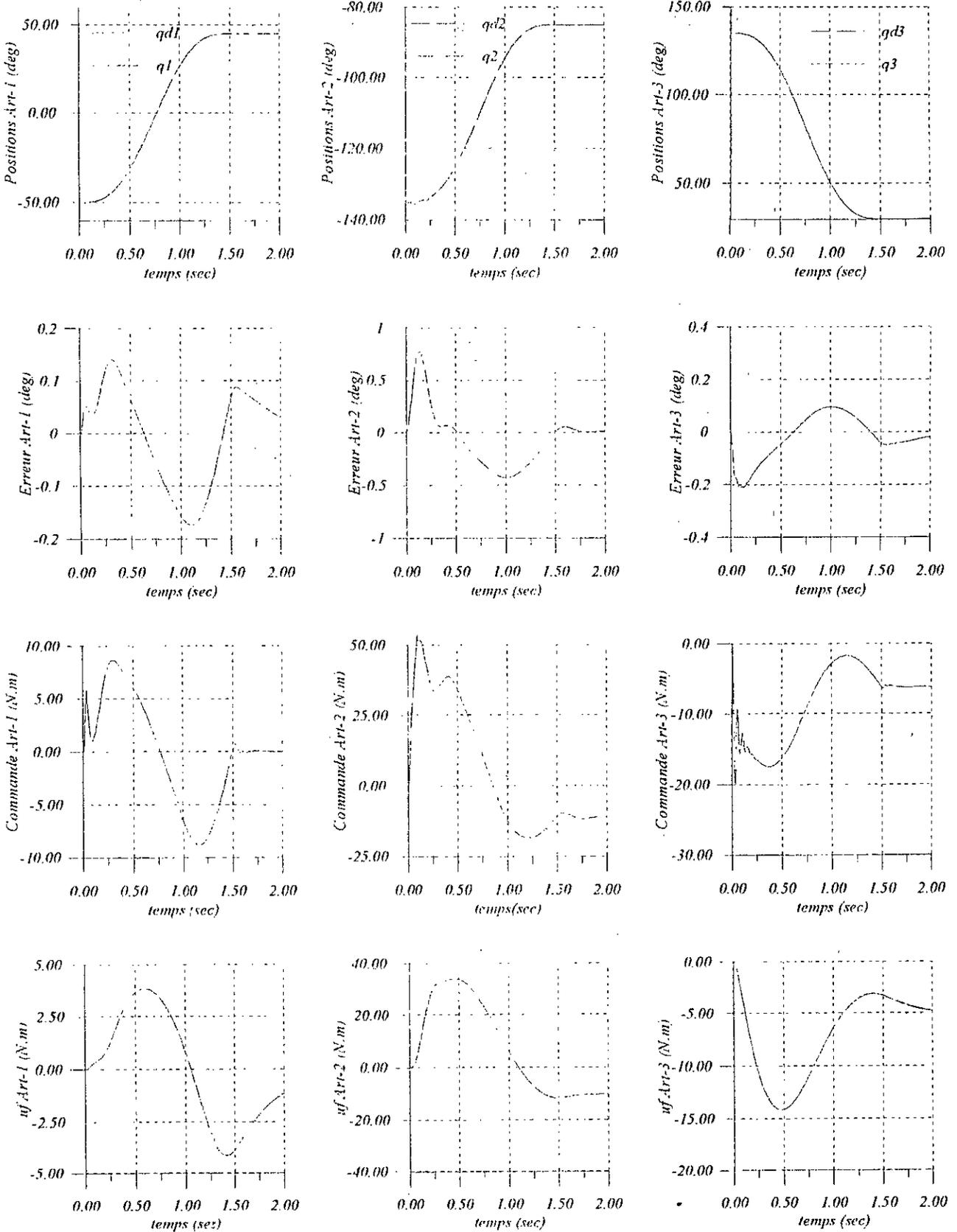


figure (IV.3a) : Réponses des différentes articulations soumises à la loi CDFAD à gain constant:
Test sans charge

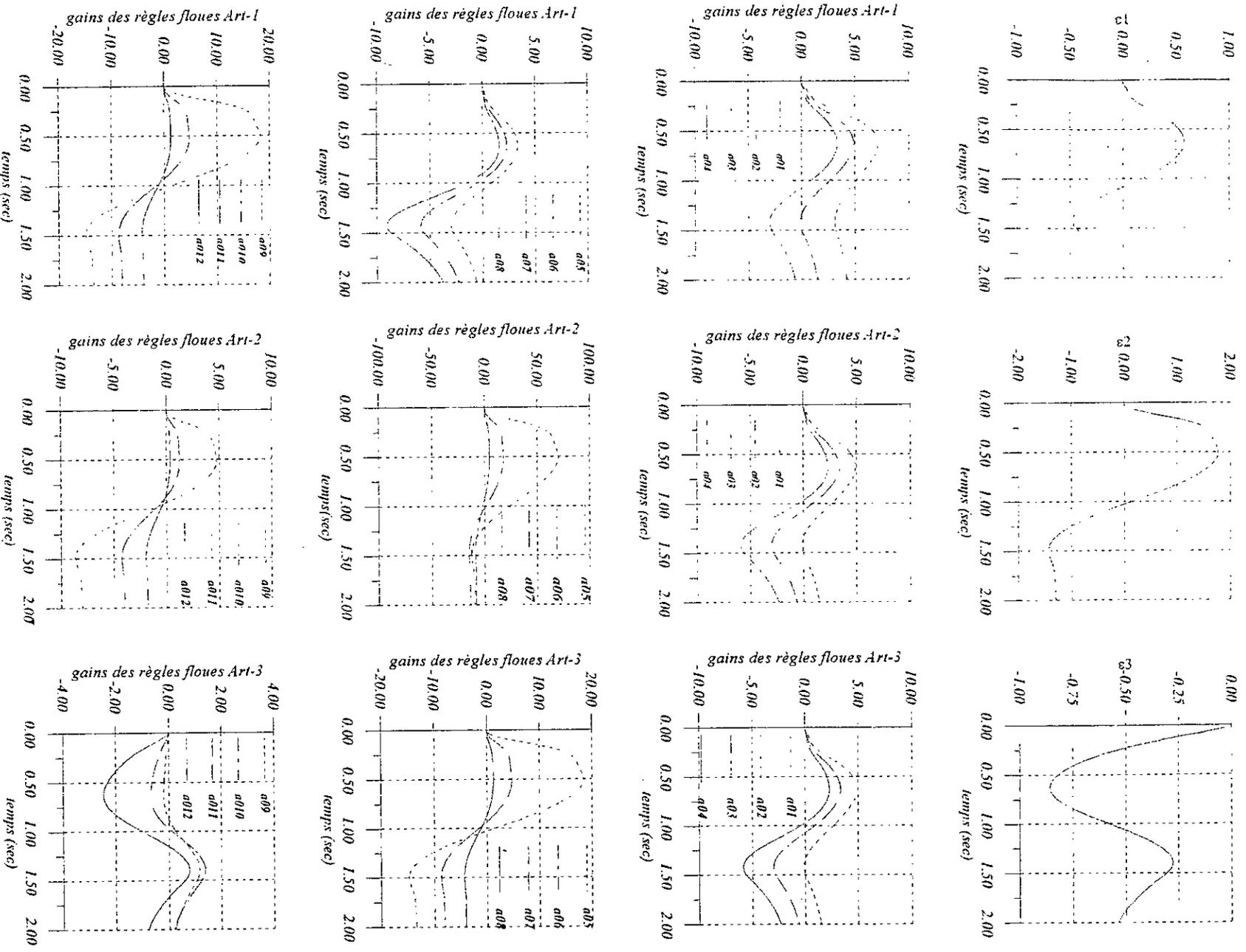


Figure (IV.3b) : Réponses des différentes articulations soumises à la loi CDFAD à gain constant.

Test sans charge

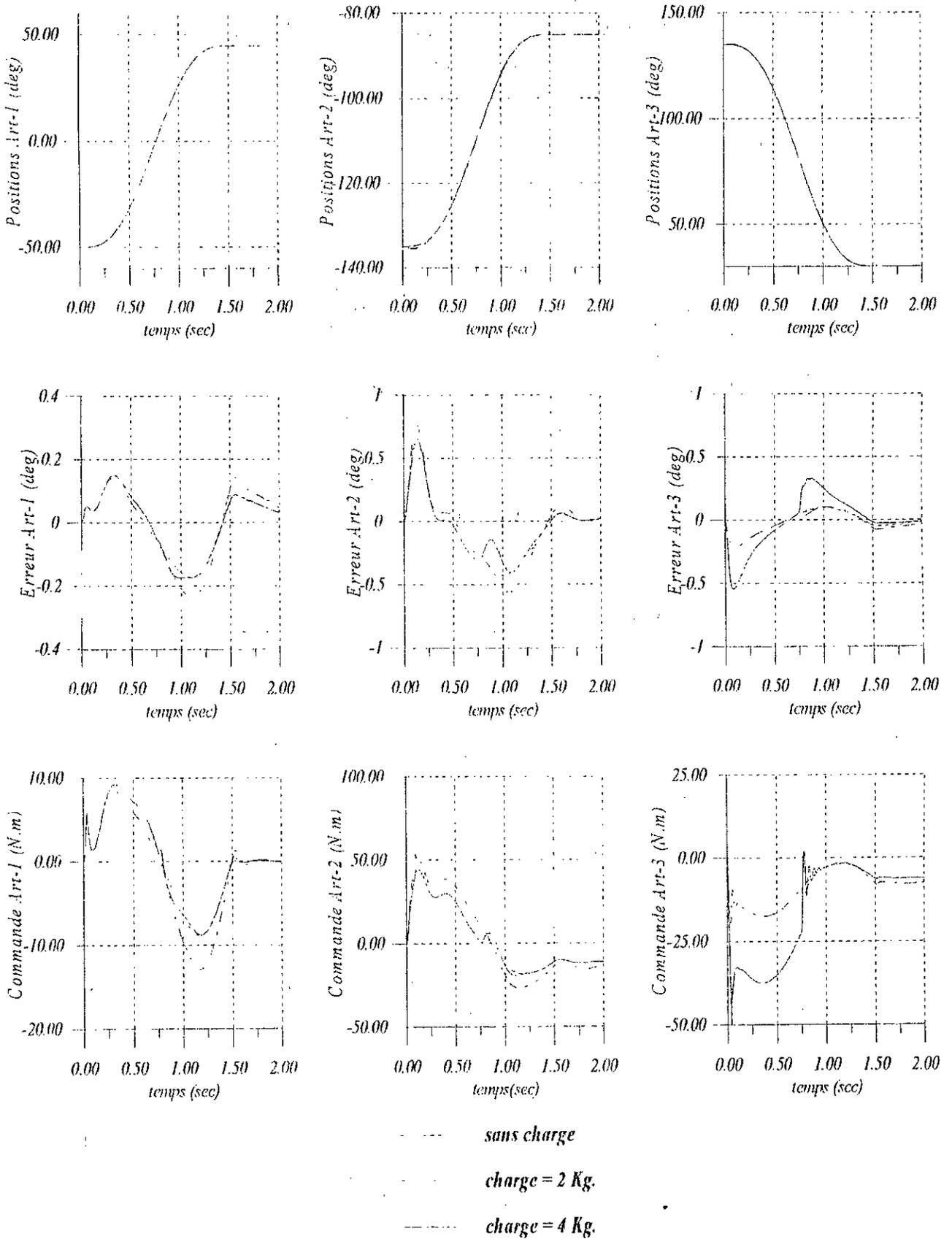


figure (IV.4) : Réponses des différentes articulations soumises à la loi CDFAD à gain constant:
 Test de variation de charge

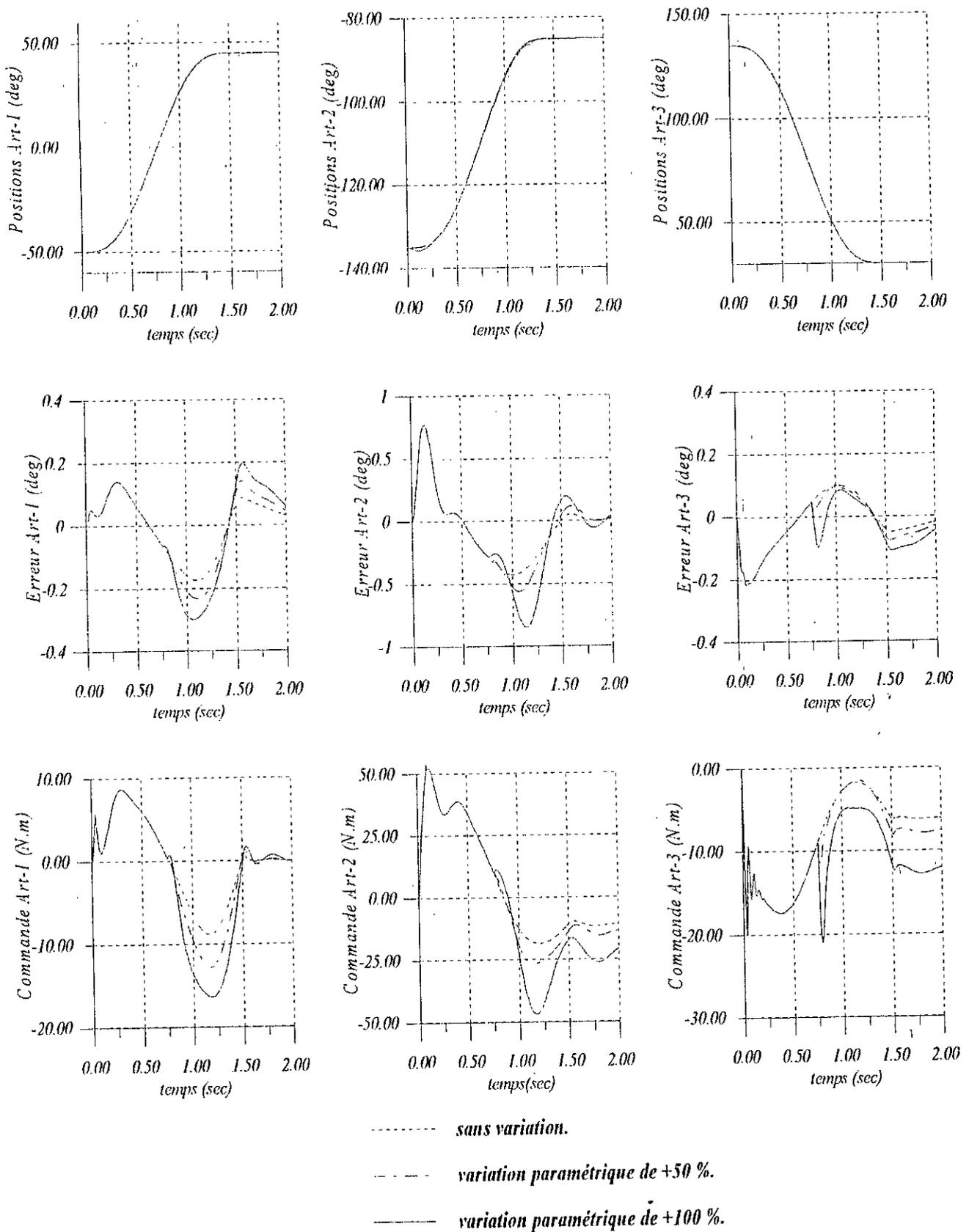


figure (IV.5) : Réponses des différentes articulations soumises à la loi CDFAD à gain constant:
Test de variation paramétrique

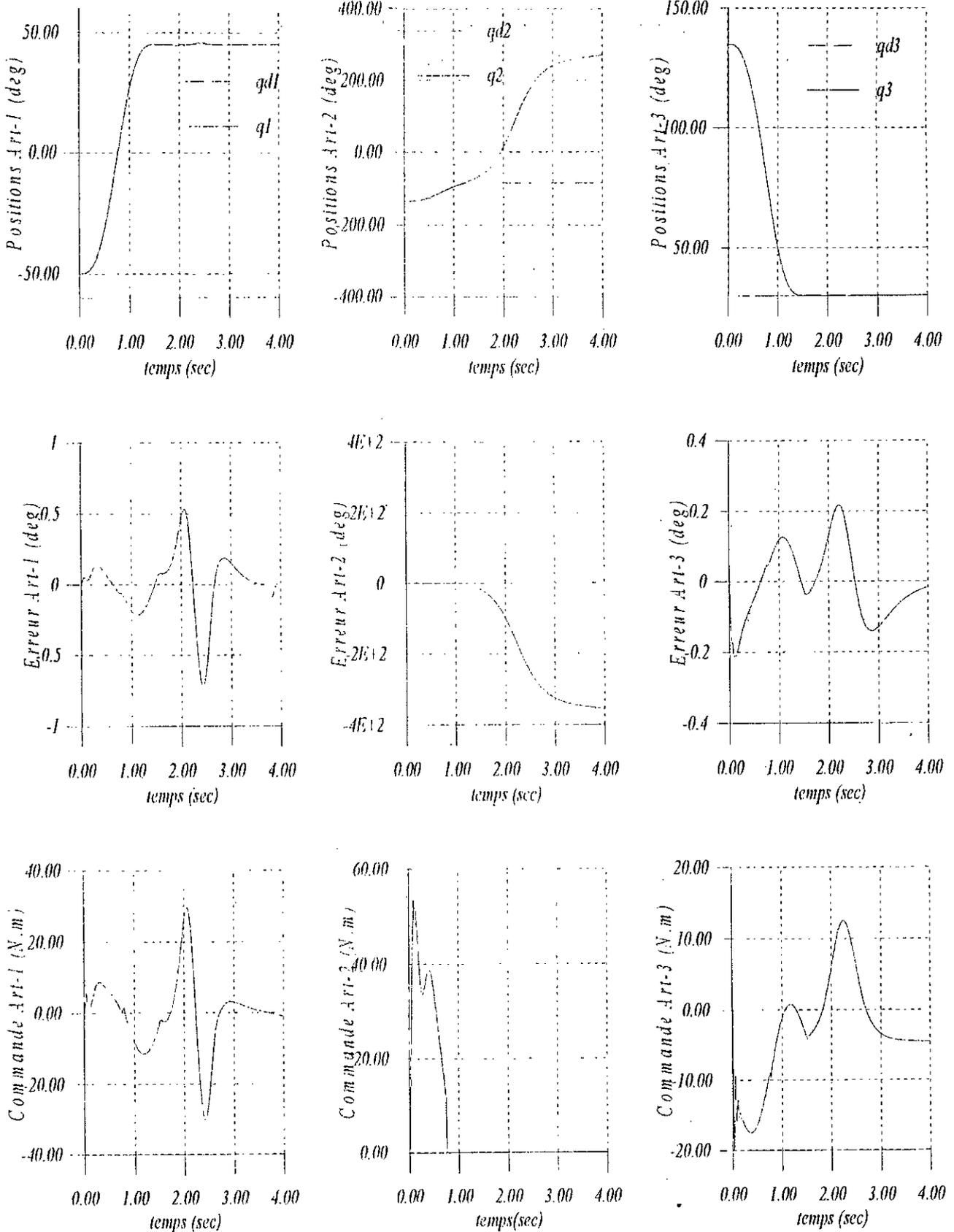


figure (IV.6) : Réponses des différentes articulations soumises à la loi CDFAD à gain constant:
Test de rupture de commande

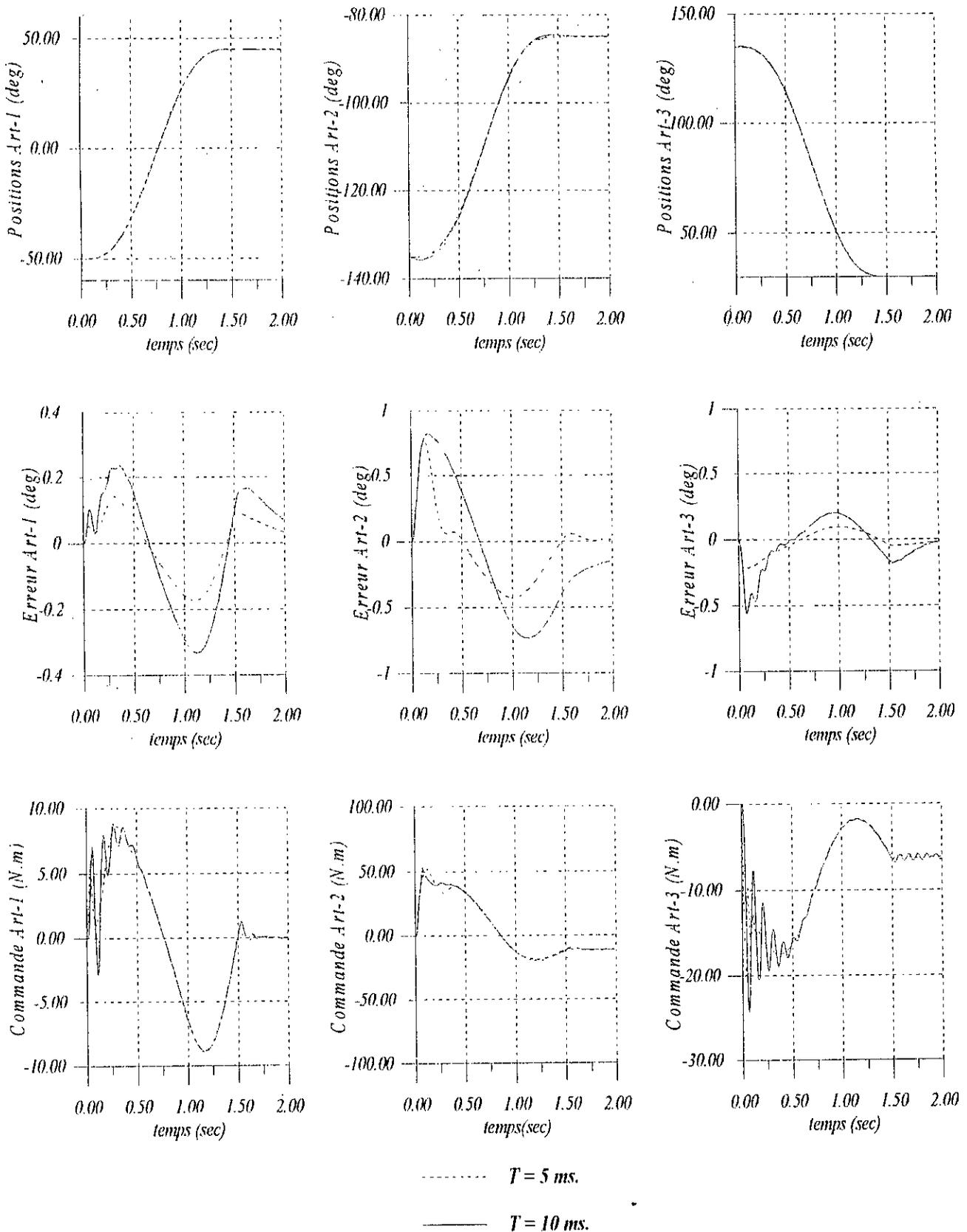


figure (IV.7) : Réponses des différentes articulations soumises à la loi CDFAD à gain constant:
 Test de variation de pas d'échantillonnage

IV.3.4. Loi d'adaptation à gain décroissant :

IV.3.4.1. Structure de la loi d'adaptation

Dans la section précédente, nous avons proposé pour l'adaptation de l'estime $\hat{A}_i(\cdot)$ et du signal auxiliaire $\hat{e}_i(\cdot)$ une loi d'adaptation à gain constant. En identification, ce type d'adaptation est souhaitable pour l'estimation des paramètres constants ou lentement variables. Cependant, lorsque les paramètres à estimer sont variables, il est préférable d'adopter une autre variété de loi d'adaptation appelée à gain décroissant [LAN89].

Dans le cas général, le modèle discret décentralisé du robot (donné par (IV.10a)), est caractérisé par des paramètres variables dans le temps. Pour cette raison nous proposons dans cette section une loi d'adaptation à gain décroissant pour estimer la matrice $\hat{A}_i(\cdot)$ et pour l'adaptation du signal $\hat{e}_i(\cdot)$.

Proposons la loi de commande suivante :

$$u_i(k) = \delta_{1i} e_i(k-1) + \delta_{2i} e_i(k-2) + Z_i^T \hat{A}_i(k) \xi^i + \hat{e}_i(k) \tag{IV.38}$$

et soit les lois d'adaptation :

$$\hat{A}_i'(k) = \hat{A}_i'(k-1) + P_{1i}'(k) w_i^{jT}(k) E_{\beta}(k) \tag{IV.39a}$$

$$\hat{e}_i(k) = \hat{e}_i(k-1) + P_{2i}(k) b_i^T E_{\beta}(k) \tag{IV.39b}$$

$$E_{\beta}(k) = \left[S_{\beta}^{-1} + b_i b_i^T P_{2i}(k) + \sum_{j=1}^{m_j} w_i^j(k) P_{1i}^j(k) w_i^{jT}(k) \right]^{-1} \left[E_i(k) + S_i \tilde{E}_i(k-1) \right] \tag{IV.39c}$$

où,

$$P_{1i}^{j-1}(k) = P_{1i}^{j-1}(k-1) + w_i^{jT}(k-1) R_i^{-1}(k-1) w_i^j(k-1) \tag{IV.39d}$$

$$P_{2i}^{-1}(k) = P_{2i}^{-1}(k-1) + b_i^T R_i^{-1}(k-1) b_i \tag{IV.39e}$$

$$w_i^{jT}(k) = \xi_j^i(k) Z_i(k) b_i^T \tag{IV.39f}$$

$$\tilde{E}_i(k) = S_{\beta} \tilde{E}_i(k-1) + Z_i^T(k) \Delta \hat{A}_i(k) \xi^i(k) \quad \tilde{E}_i(0) = [0 \quad 0]^T \tag{IV.39g}$$

$$E_i(k) = [e_i(k) \quad e_i(k-1)]^T \quad S_i = \begin{bmatrix} \delta_{1i} & -\delta_{2i} \\ \delta_{0i} & \delta_{0i} \\ 1 & 0 \end{bmatrix}; \quad b_i = \begin{bmatrix} 1 \\ \delta_{0i} \\ 0 \end{bmatrix} \tag{IV.39h}$$

- $\hat{A}_i^j(k)$: dénote la $j^{\text{ème}}$ colonne de la matrice $\hat{A}_i(k)$ (vecteur des paramètres de la $j^{\text{ème}}$ règle);
- $\Delta \hat{A}_i(k) = \hat{A}_i(k) - \hat{A}_i(k-1)$;
- $\xi_j^i(k)$: désigne le $j^{\text{ème}}$ élément du vecteur d'activation $\xi^i(k)$;
- $P_{1i}^{j-1}(0) \in \mathbb{R}^{np \times np}$, $R_i^{-1}(k-1) \in \mathbb{R}^{2 \times 2}$ ($P_{2i}^{-1}(0) \in \mathbb{R}$) : matrices arbitraires définies positives (scalaires arbitraires positifs) avec np est le nombre des gains dans la conséquence de chaque règle floue. Il égal à 1 (respectivement 5) pour un modèle flou de Sugeno d'ordre zéro (d'ordre un).

La matrice S_{β} est la solution de l'équation de Recatti discrète donnée par : $S_i S_{\beta} S_i - S_{\beta} = -L_i^T L_i - Q_i$ où L_i est une matrice arbitraire et Q_i une matrice arbitraire définie positive. Les constantes δ_{0i} , δ_{1i} , δ_{2i} sont choisies telles que les pôles de l'équation caractéristique de la matrice S_i soient situées dans le cercle unitaire.

Proposition 2 :

Etant donné un robot manipulateur à n degrés de liberté dont le modèle dynamique discret de chaque articulation peut être mis sous la forme (IV.13).

Si on choisit la loi de commande (IV.38) et en tenant compte des hypothèses H1, H2 et les lois d'adaptation (IV.19a,b,c,d,e,f,g,h), on aura :

R1 : Les états $q_i(k)$ $i = 1 \dots n$ sont bornés;

R2 : Les paramètres $\hat{A}_i(k)$ et $\hat{e}_i(k)$, $i = 1 \dots n$ et les commandes $u_i(k)$ sont bornées;

R3 : Les erreurs $e_i(k)$ $i = 1 \dots n$ sont bornées et convergent asymptotiquement vers zéro.

Démonstration

Pour démontrer cette proposition, nous utilisons la théorie d'hyperstabilité de Popov. Une étape primordiale pour appliquer cette théorie, consiste à représenter le système en contre-réaction de deux sous-systèmes :

- un sous-système linéaire invariant dans le temps dans la chaîne directe;
- un sous-système non linéaire variant dans le temps dans la chaîne de retour.

Cette représentation particulière est appelée *représentation équivalente par bouclage* [LAN79]. Elle est à base d'un arrangement mathématique, et ne doit pas nécessairement correspondre à une configuration physique.

Rappel sur la théorie d'hyperstabilité :

Considérons un système en boucle fermée tel que représenté par la figure (IV.8), constitué d'un sous-système linéaire invariant dans le temps, décrit par le système d'équations discrètes suivantes:

$$\begin{cases} x(k+1) = A_d x(k) + B_d v(k) \\ y(k) = C_d x(k) + D_d v(k) \end{cases} \quad (IV.40)$$

avec le paire (A_d, B_d) commandable, et le paire (A_d, C_d) observable et d'un bouclage non linéaire sur l'état : $w_f(k) = f_d(y(k), k, l)$, $l \leq k$ variant dans le temps.

On suppose que le bouclage sur l'état satisfait l'inégalité de Popov suivante [LAN79]:

$$\eta(k_1, k_2) \triangleq \sum_{k=k_1}^{k_2} w_f^T(k) y(k) \geq -\rho_0^2 \quad \forall k_2 \geq k_1 \quad (IV.41)$$

avec ρ_0^2 une constante positive et les vecteurs $w(k)$, $y(k)$ sont supposés de même dimension.

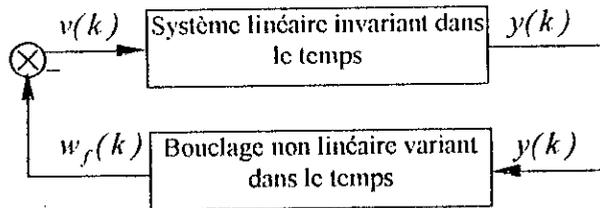


figure (IV.8): Représentation équivalente par bouclage

Définitions [LAN79]

Le système de la figure (IV.8), est dit hyperstable s'il existe des constantes positives $\alpha_0 \geq 0$ et $\beta > 0$, telles que pour toute solution $x(k)$ de (IV.40) initialisée à $x(0)$, l'inégalité :

$$\|x(k)\| \leq \beta_0 [\|x(0)\| + \alpha_0]$$

est satisfaite pour tous bouclage non linéaire $w_f(k)$ satisfaisant l'inégalité (IV.41). Il est dit asymptotiquement hyperstable, s'il est hyperstable et de plus $\lim_{k \rightarrow \infty} x(k) = 0$ pour tous bloc satisfaisant l'inégalité (IV.41).

Théorème (d'hyperstabilité asymptotique) : [LAN79]

La condition nécessaire et suffisante pour que le système de la figure (IV.8), soit asymptotiquement hyperstable, est que la fonction de transfert discrète associée au bloc linéaire :

$$H(z) = D_d + C_d(zI - A_d)^{-1} B_d \tag{IV.42}$$

soit strictement réelle positive (SRP).

Application de l'approche d'hyperstabilité :

Pour appliquer la théorie d'hyperstabilité à notre problème on suit les étapes suivantes :

1. On établit la représentation équivalente par bouclage de la dynamique du système en boucle fermée. Pour cela, il est nécessaire de prendre en considération les équations de la dynamique de l'erreur, de la loi de commande et des lois d'adaptation. Afin d'assurer la convergence asymptotique de l'erreur en boucle fermée, l'état du sous-système linéaire est choisis dépendant de l'erreur de poursuite;
2. On vérifie la condition de stricte réelle positive de la fonction de transfert du sous-système linéaire;
3. On vérifie l'inégalité de Popov dans le sous-système non linéaire.

Dans la démonstration, chaque articulation est considérée comme un sous-système indépendant. Ainsi, pour simplifier l'écriture, on omit l'indice "i" de l'articulation.

1. Représentation équivalente par bouclage :

Considérons l'équation dynamique de l'erreur E suivante :

$$E(k) = SE(k-1) + b\tilde{\Phi}(k-1) \tag{IV.43a}$$

où,

$$\tilde{\Phi}(k-1) = Z^T(k)\tilde{A}(k-1)\xi(k) + \tilde{\varepsilon}(k) \tag{IV.43b}$$

Cette équation est similaire à l'équation donnée par (IV.17a,b), où dans la loi de commande, on utilise les estimées \hat{A} et $\hat{\varepsilon}$ de l'instant $k-1$ au lieu des estimées à l'instant k .

Soit le vecteur E_p dont l'évolution est donnée par :

$$E_p(k) = SE_p(k-1) + b\tilde{\Phi}(k) \text{ avec } E_p(0) = E(0) \tag{IV.44a}$$

où,

$$\tilde{\Phi}(k) = Z^T(k)\tilde{A}(k)\xi(k) + \tilde{\varepsilon}(k) \tag{IV.44b}$$

La variable E_p représente le vecteur des erreurs de poursuites lorsqu'on utilise la loi de commande (IV.38). D'autre part, on définit l'erreur filtrée E_f telle que :

$$E_f(k) = S_f E_p(k) \tag{IV.45}$$

Si on considère l'hypothèse H2 on peut écrire :

En boucle fermée, les erreurs $E_p(k)$, $E_f(k)$ sont décrites par :

$$\begin{cases} E_p(k) = SE_p(k-1) + \Pi(k) \\ E_f(k) = S_f SE_p(k-1) + S_f \Pi(k) \end{cases} \tag{IV.46}$$

avec, $\Pi(k) = -bZ^T(k)\hat{A}(k)\xi(k) - b\hat{\varepsilon}(k)$

les lois d'adaptation de $\hat{A}(k)$, $\hat{\varepsilon}(k)$ sont données par (IV.39a,b) :

$$\hat{A}^j(k) = \hat{A}^j(k-1) + P_1^j(k)w^{jT}(k)E_{r_j}(k) \quad (IV.47a)$$

$$\hat{e}(k) = \hat{e}(k-1) + P_2(k)b^T E_{r_j}(k) \quad (IV.47b)$$

avec, $\hat{A}^j(k)$: la $j^{ème}$ colonne de la matrice $\hat{A}(k)$;

$$w^j(k) = bZ^T(k)\xi_j(k) ;$$

$\xi_j(k)$: le $j^{ème}$ élément du vecteur $\xi(k)$;

alors on peut déduire :

$$\Pi(k) = - \left[b\hat{e}(k) + \sum_{j=1}^{nr} w^j(k)\hat{A}^j(k) \right] \quad (IV.48)$$

Une représentation équivalente par bouclage est illustrée par le schéma de la figure (IV.9). Dans ce schéma, on a introduit les deux matrices supplémentaires Λ et $\Gamma(k)$ de manière que la description du comportement du système en boucle fermée ne soit pas modifiée. Dans cette représentation, on distingue :

- un système linéaire invariant dans le temps (S_A), d'entrée $\Pi(k)$ et de sortie $\zeta(k)$ et de vecteur d'état $E_p(k)$;
- un système non linéaire variant dans le temps (S_B), d'entrée $\bar{\Pi}(k) = \zeta(k)$ et de sortie $\bar{\zeta}(k) = -\Pi(k)$. Il englobe les lois d'adaptation non linéaires des paramètres $\hat{A}^j(k)$ et $\hat{e}(k)$.

Pour assurer la convergence asymptotique vers zéro de $E_p(k)$, il suffit de montrer que :

- Le système S_A possède une fonction de transfert discrète strictement réelle positive ;
- Le produit entrée-sortie du système S_B vérifie l'inégalité de Popov.

2. Vérification de la condition de stricte réelle positive :

Le sous-système linéaire S_A est composé de deux sous-systèmes en parallèle S_{A1} et S_{A2} : le premier sous système (S_{A1}) est d'entrée $\Pi(k)$ et de sortie $\zeta(k)$. Il est décrit par le système d'équations :

$$\begin{cases} E_p(k) = SE_p(k-1) + \Pi(k) \\ \zeta_1(k) = S_f SE_p(k-1) + \frac{1}{2} S_f \Pi(k) \end{cases} \quad (IV.49)$$

Le deuxième sous-système (S_{A2}) est d'entrée $\Pi(k)$ et de sortie $\zeta_2(k)$. Il est décrit par la relation :

$$\zeta_2(k) = \frac{1}{2}(S_f - \Lambda)\Pi(k) \quad (IV.50)$$

la sortie global du sous-système linéaire est donnée par :

$$\zeta(k) = \zeta_1(k) + \zeta_2(k) \quad (IV.51)$$

Montrons que les sous-systèmes S_{A1} et S_{A2} sont strictement réels positifs.

- Sous-système S_{A1} :

A partir de l'équations (IV.49), les matrices du système d'équations d'état sont données par le quadruplé $\left(S, I_2, S_f S, \frac{1}{2} S_f \right)$. Pour montrer que les équations (IV.49) correspond à une fonction de transfert, strictement réelle positive, on considère le lemme suivant :

Lemme 1 : (Lemme positif réel en temps discret) [GEO92]

La fonction de transfert d'un système linéaire discret de quadruplé (A_d, B_d, C_d, D_d) est strictement réelle positive si et seulement s'il existe des matrices K_d , L_d et des matrices symétriques définies positives T_d et Q_d telles que :

$$\begin{cases} A_d^T T_d A_d - T_d = -L_d L_d^T - Q_d \\ B_d^T T_d A_d - K_d^T L_d^T = C_d \\ K_d^T K_d = D_d + D_d^T - B_d^T T_d B_d \end{cases} \quad (\text{IV.52})$$

Posons $T_d = S_f$, Q_d matrice arbitraire définie positive, L_d matrice arbitraire et K_d matrice d'éléments nuls. On montre que le quadruplé $(S, I_3, S_f S, \frac{1}{2} S_f)$ vérifie les conditions (IV.52), si la matrice S_f est choisie symétrique et vérifie la relation :

$$S^T S_f S - S_f = -L_d L_d^T - Q_d \quad (\text{IV.53})$$

En appliquant le lemme de Kalman-Yagobovitch dans le cas discret, la solution S_f de l'équation (IV.53) existe si la matrice S est choisie avec des valeurs propres dont les modules sont inférieures à l'unité. Dans ce cas, le sous-système S_{A1} est de fonction de transfert strictement réelle positive.

- Sous-système S_{A2} :

Le sous-système S_{A2} est d'entrée $\Pi(k)$ et de sortie $\zeta_2(k)$. Il est décrit par la relation :

$$\zeta_2(k) = \frac{1}{2} (S_f - A) \Pi(k) \quad (\text{IV.54})$$

Si on choisit la matrice $(S_f - A)$ symétrique définie positive, le produit entrée-sortie du sous-système S_{A2} vérifie :

$$\sum_{k=0}^{K_1} \zeta_2^T(k) \Pi(k) = \frac{1}{2} \sum_{k=1}^{K_1} \Pi^T(k) (S_f - A) \Pi(k) \geq 0 \quad (\text{IV.55})$$

D'après la propriété des systèmes linéaires positifs, le système S_{A2} possède une fonction de transfert strictement réelle positive.

Puisque la mise en parallèle de deux sous-systèmes ayant des fonctions de transfert strictement réelles positives produit un système de fonction de transfert strictement réelle positive, on conclut que le système S_B est SRP.

3. Vérification de la condition d'inégalité de Popov :

Le système S_B est constitué de deux sous-systèmes principales en contre réaction :

- un sous-système dans la chaîne directe, lui même composé de $m+1$ sous-systèmes en parallèles $(S_{B1} \cdots S_{B_{m+1}})$ qui définissent les dynamiques des variables $\hat{A}^j(k)$ $j = 1 \cdots m$ et $\hat{e}(k)$. Chacun de ces sous-système (d'entrée $E_j(k)$ et de sortie $w^j(k) \tilde{A}^j(k)$) est en contre réaction avec le système de matrice de gain $\frac{1}{2} \Gamma(k)$.
- un sous-système $S_{B_{m+2}}$ en contre réaction avec les sous-systèmes $S_{B1} \cdots S_{B_{m+1}}$. Il est caractérisé par une matrice de gain $\frac{1}{2} (A - \Gamma(k))$.

D'autre part la mise en parallèle ou en contre réaction de deux ou de plusieurs sous-systèmes, chacun vérifie l'inégalité de Popov, produit un système qui vérifie cette inégalité. Donc, pour que le système

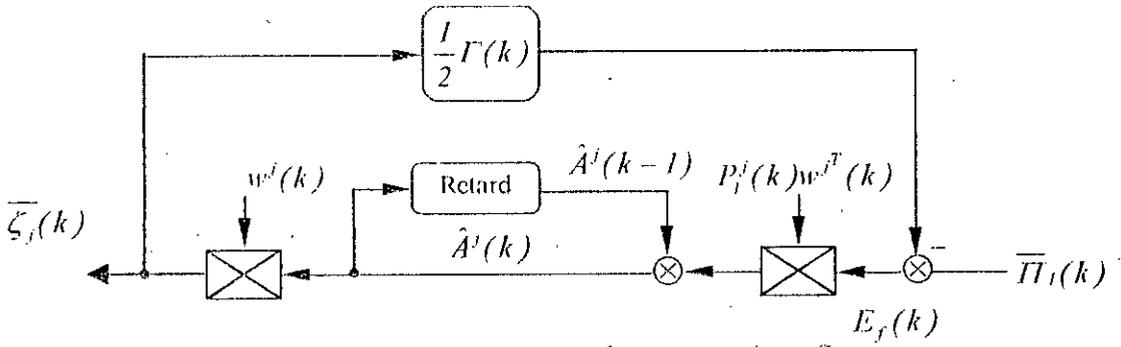


figure (IV.11) : Représentation du sous-système S_{Bj}

Le sous-système linéaire est décrit par :

$$\begin{cases} \hat{A}^j(k) = \hat{A}^j(k-1) + P_i^j(k)w^{jT}(k)E_f(k) \\ \bar{\zeta}_j(k) = w^j(k)\hat{A}^j(k-1) + w^j(k)P_i^j(k)w^{jT}(k)E_f(k) \end{cases} \quad (IV.60)$$

avec $w^j(k) = bZ^T(k)\xi_j(k)$.

Pour que le produit entrée-sortie du système de la figure (IV.11) vérifie l'inégalité de Popov, on peut utiliser le lemme 2. Pour cela, il suffit de vérifier les relations (IV.57) et la condition définie positive de la matrice (IV.58).

Les équations d'état du système (IV.60) sont décrites par le quadruplé :

$$(I_{np}, P_i^j(k)w^{jT}(k), w^j(k), w^j(k)P_i^j(k)w^{jT}(k))$$

Donc, on doit trouver des séquences de matrices $U_a(k)$, $S(k)$, $S_a(k)$ et $N_a(k)$ telles que :

$$U_a(k) - U_a(k-1) = -M_a(k-1) + w^{jT}(k-1)\Gamma(k-1)w^j(k-1) \quad (IV.61a)$$

$$w^j(k-1)P_i^{jT}(k-1)U_a(k) + S_a^T(k-1) = w^j(k-1) + w^j(k-1)P_i^{jT}(k-1)w_j^T(k-1)\Gamma(k-1)w^j(k-1) \quad (IV.61b)$$

$$\begin{aligned} N_a(k-1) - w^j(k-1)P_i^{jT}(k-1)w^{jT}(k-1)\Gamma(k-1)w^j(k-1)P_i^j(k-1)w^{jT}(k-1) = \\ w^j(k-1)P_i^j(k-1)w^{jT}(k-1) + w^j(k-1)P_i^{jT}(k-1)w^{jT}(k-1) + \\ w^j(k-1)P_i^{jT}(k-1)U_a(k)P_i^j(k-1)w^{jT}(k-1) \end{aligned} \quad (IV.61c)$$

De manière que la matrice suivante:

$$\begin{bmatrix} M_a(k-1) & S_a(k-1) \\ S_a^T(k-1) & N_a(k-1) \end{bmatrix}$$

soit semi-définie positive.

Si on choisit :

$$U_a(k) = P_{ij}^j(k) : \text{matrice symétrique définie positive ;}$$

$$M_a(k) = O_{np \times np} \quad \forall k, S_a(k) = O_{np \times 2} \quad \forall k ;$$

$$\Gamma(k-1) = R^{-1}(k-1) : \text{matrice symétrique définie positive;}$$

$$N_a(k-1) = w^j(k-1)P_i^j(k-1)w^{jT}(k-1).$$

Dans ce cas, les relations (IV.61a,b,c) deviennent :

$$P_i^{j-1}(k) - P_i^{j-1}(k-1) = w^{jT}(k-1)R^{-1}(k-1)w^j(k-1) \quad (IV.62a)$$

$$\begin{cases} \hat{\varepsilon}(k) = \hat{\varepsilon}(k-1) + P_2(k)b^T E_f(k) \\ \bar{\zeta}_{m+1}(k-1) = b\hat{\varepsilon}(k-1) + P_2(k)bb^T E_f(k) \end{cases} \quad (IV.66)$$

Ce sous-système apparaît comme un cas particulier des sous-systèmes S_{B_j} $j = 1 \dots m$ (donnée par (IV.60)) où $w^{jT}(k) = b^T$. En utilisant le lemme 2, et en suivant les mêmes étapes effectuées pour les sous-systèmes S_{B_j} , on montre que le choix :

$$P_2^{-1}(k) = P_2^{-1}(k-1) + b^T R^{-1}(k-1)b \quad (IV.67)$$

avec, $R^{-1}(k-1) \in \mathbb{R}^{m \times m}$ matrice arbitraire symétrique définie positive. $P_2^{-1}(0) > 0$ assure l'existence des matrices $U_d(k)$, $M_d(k)$, $S_d(k)$ et $N_d(k)$ qui vérifié les conditions du lemme. Ainsi, on obtient :

$$\sum_{k=0}^{k_I+1-T} \bar{\zeta}_{m+1}(k) \bar{\Pi}_1(k) \geq -\frac{1}{2} P_2^{-1}(0) \hat{\varepsilon}(0)^2 \quad (IV.68)$$

Le sous-système $S_{B_{m+2}}$ (en contre réaction) d'entrée $\bar{\zeta}(k) = \sum_{j=1}^{m+1} \bar{\zeta}_j(k)$ et de sortie $\bar{\Pi}_2(k)$ est caractérisé par une matrice de gain $\frac{1}{2}(\Lambda - \Gamma(k))$. Le sous-système $S_{B_{m+2}}$ vérifie l'inégalité de Popov si on choisit la matrice $\frac{1}{2}(\Lambda - \Gamma(k))$ définie positive. Dans ce cas on a :

$$\sum_{k=0}^{k_I+1} \bar{\zeta}^T(k) \bar{\Pi}_2(k) = \sum_{k=0}^{k_I+1} \zeta^T(k) \left(\frac{1}{2}(\Lambda - \Gamma(k)) \zeta(k) \right) \quad (IV.69)$$

$$\sum_{k=0}^{k_I+1} \bar{\zeta}^T(k) \bar{\Pi}_2(k) = \frac{1}{2} \sum_{k=0}^{k_I+1} \zeta^T(k) (\Lambda - \Gamma(k)) \zeta(k) \geq 0 \quad (IV.70)$$

Le produit entrée-sortie du système global S_R vérifie :

$$\sum_{k=0}^{k_I+1} \bar{\zeta}^T(k) \bar{\Pi}(k) = \sum_{k=1}^{k_I+1} \zeta^T(k) \bar{\Pi}_1(k) - \sum_{k=1}^{k_I+1} \zeta^T(k) \bar{\Pi}_2(k) \quad (IV.71)$$

$$\sum_{k=0}^{k_I+1} \bar{\zeta}^T(k) \bar{\Pi}(k) = \sum_{j=1}^{m+1} \left(\sum_{k=0}^{k_I+1} \zeta_j^T(k) \bar{\Pi}_1(k) \right) - \sum_{k=0}^{k_I+1} \zeta^T(k) \bar{\Pi}_2(k) \quad (IV.72)$$

en utilisant les relations (IV.65), (IV.68) et (IV.70), l'expression (IV.72) devient :

$$\sum_{k=0}^{k_I+1} \bar{\zeta}^T(k) \bar{\Pi}(k) \geq -\gamma_0^2 \quad (IV.73a)$$

avec,

$$\gamma_0^2 = \frac{1}{2} \sum_{j=1}^m \lambda_{\max} \left(P_j^{j-1}(0) \right) \left\| \hat{A}^j(0) \right\|^2 + \frac{1}{2} P_2^{-1}(0) \hat{\varepsilon}^2(0) \quad (IV.73b)$$

Les relations (IV.73a,b) montrent que le système non linéaire S_R vérifie l'inégalité de Popov.

Puisque le système linéaire de la chaîne directe de la représentation équivalente par bouclage possède une fonction de transfert strictement réelle positive, et le système non linéaire vérifie l'inégalité de Popov, on conclut d'après le théorème d'hyperstabilité asymptotique que le vecteur des erreurs de poursuite ($E_p(k)$) converge asymptotiquement vers zéro (conséquence R3 de la proposition 2).

D'autre part, selon l'hypothèse H1, les variables de position des trajectoires désirées sont bornées. Alors, on déduit que les variables articulaires des positions sont aussi bornées aux instants d'échantillonnage (conséquence R1).

L'utilisation des lois d'adaptation des estimés \hat{A} et \hat{e} (données par (IV.47a,b)), nécessite le calcul de la grandeur $E_f(k)$. Cette dernière ne peut pas être calculée directement à partir des relations (IV.44a) et (IV.45), d'où la nécessité de trouver une expression convenable pour $E_f(k)$.

On définit le vecteur d'erreur $\tilde{E}(k)$ tel que :

$$\tilde{E}(k) = E_p(k) - E(k) \tag{IV.74}$$

Sa variation en fonction du temps peut être calculée à partir de (IV.43a,b) et (IV.44a,b), on obtient :

$$\tilde{E}(k) = S\tilde{E}(k-1) + b\Delta\tilde{\Phi}(k) \quad \tilde{E}(0) = [0 \ 0]^T \tag{IV.75a}$$

où,

$$\Delta\tilde{\Phi}(k) = \tilde{\Phi}(k) - \tilde{\Phi}(k-1) = b(Z(k)\Delta\tilde{A}(k)\xi(k) + \Delta\tilde{e}(k)) \tag{IV.75b}$$

$$\Delta\tilde{A}(k) = \tilde{A}(k) - \tilde{A}(k-1) \tag{IV.75c}$$

$$\Delta\tilde{e}(k) = \tilde{e}(k) - \tilde{e}(k-1) \tag{IV.75c}$$

D'après la définition de \tilde{A} et \tilde{e} , et en considérant l'hypothèse H2, on peut écrire :

$$\begin{cases} \Delta\tilde{A}(k) = -\Delta\hat{A}(k) = -(\hat{A}(k) - \hat{A}(k-1)) \\ \Delta\tilde{e}(k) = -\Delta\hat{e}(k) = -(\hat{e}(k) - \hat{e}(k-1)) \end{cases} \tag{IV.76}$$

d'autre part les expressions de $\Delta\hat{A}(k)$ et $\Delta\hat{e}(k)$ données par (IV.47a,b) peuvent être utilisées pour déduire :

$$\tilde{E}(k) = S\tilde{E}(k-1) - \left(\sum_{i=1}^m w^i(k) P_i^j(k) w^{jT}(k) + bP_2(k)b^T \right) E_f(k) \tag{IV.77}$$

En utilisant la définition de $E_p(k)$ et $\tilde{E}(k)$ données par (IV.45) et (IV.74), on peut déduire l'expression de $E_f(k)$. On trouve :

$$E_f(k) = \left[S_i^{-1} + \left(\sum_{j=1}^m w^j(k) P_i^j(k) w^{jT}(k) + bP_2(k)b^T \right) \right]^{-1} [S\tilde{E}(k-1) + E(k)] \tag{IV.78}$$

l'expression précédente permet le calcul directe de $E_f(k)$ à l'instant k .

D'après l'équation (IV.45) on déduit que l'erreur $E_f(k)$ est bornée. Si de plus on considère les équations (IV.47a,b) (IV.62a) et (IV.67), on conclut la bornitude des estimées $\hat{A}^j(k)$, $\hat{e}(k)$, des gains $P_i^j(k)$, $P_2(k)$ (conséquence R2 de la proposition 2) et par conséquent des commandes. Selon les équations (IV.62a) et (IV.67), les gains d'adaptation $P_i^j(k)$, $P_2(k)$ sont décroissants.

Schéma de principe :

Dès que la matrice $P_i^j(k)$ et le gain $P_2(k)$ sont décroissants, alors la loi d'adaptation des paramètres \hat{A}_i et du signal \hat{e}_i sont à gain décroissant.

Le schéma de principe de la loi de Commande Décentralisée Flou Adaptative Discrète (CDFAD) à gain décroissant, donnée par (IV.38) et (IV.39a,b,c,d,e,f,g,h), est illustré par la figure (IV.13).

La proposition 2 révèle que l'utilisation de la loi de CDFAD à gain décroissant assure la bornitude et la convergence asymptotique vers zéro des erreurs de poursuite $e_i(k)$ (aux instants d'échantillonnage). Lors de calcul de la commande $u_i(k)$, seules les mesures successives de position de l'articulation considérée sont nécessaire. Dès que chaque colonne de la matrice $\hat{A}_i(k)$ correspond à une règle floue, on constate que pendant l'adaptation chaque règle utilise des gains d'adaptation décroissants indépendants. La décroissance des gains est maintenue même après l'établissement du régime permanent.

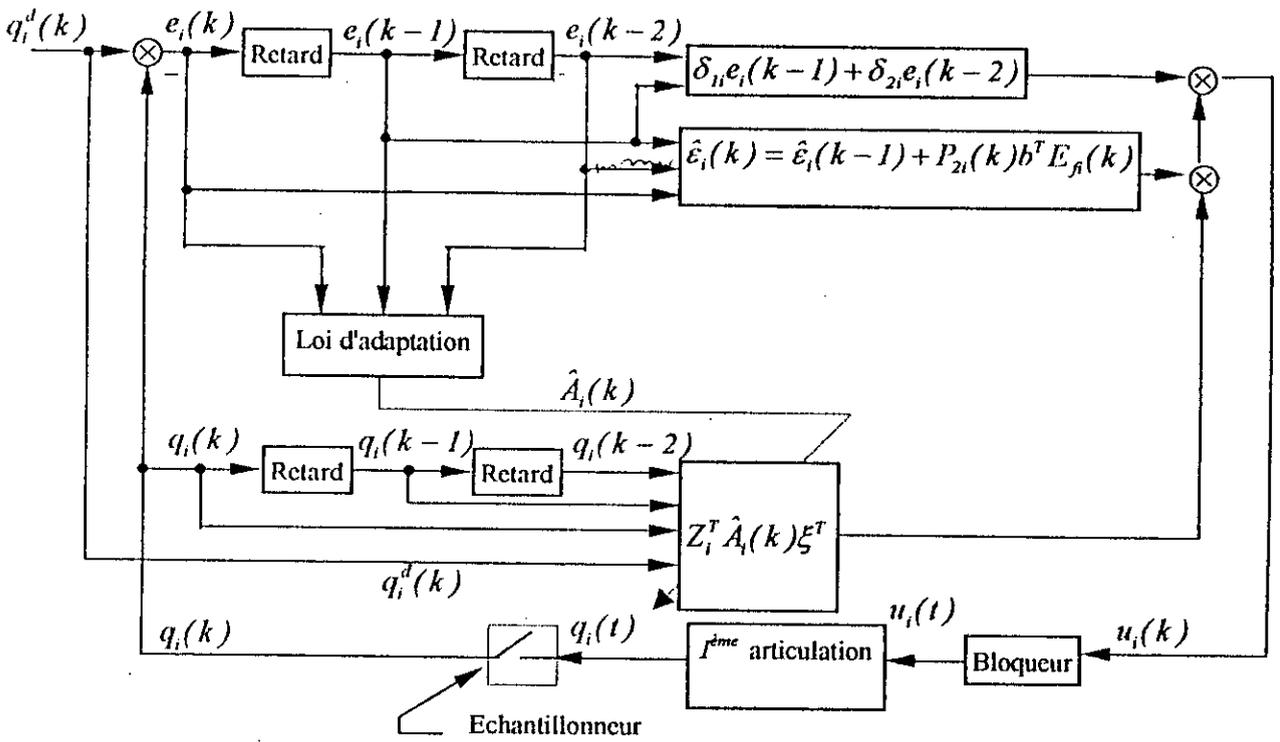


figure (IV.13) : Schéma de principe de la CDFAD à gain décroissant.

IV.3.4.2. Application au robot PUMA560

Nous avons appliqué la technique de la CDFAD à gain décroissant pour la commande en poursuite du Robot PUMA560 à trois degrés de liberté. Les résultats obtenus sont présentés dans cette section

Paramètres du régulateur :

Chaque articulation est dotée d'un régulateur local indépendant dont le modèle flou de type Sugeno est d'ordre zéro ayant huit règles identiquement à ceux choisis lors de l'application de la CDFAD à gain décroissant. Les paramètres de la loi de commande sont regroupés dans la table (IV.3)

	Symboles des paramètres	Première articulation i=1	Deuxième articulation i=2	Troisième articulation i=3
	δ_{0i}	200000	200000	137500
	δ_{1i}	35000	35000	27000
	δ_{2i}	-27000	-30000	-25000
	pôles de S_i	-0.4652; 0.2902	-0.4846; 0.3096	-0.4500; 0.3000
	Q_i	$2500 \times I_2$	$2500 \times I_2$	$2500 \times I_2$
	S_{fi}	$\begin{bmatrix} 5310.2 & -145.0 \\ -145.0 & 2596.8 \end{bmatrix}$	$\begin{bmatrix} 5341.5 & -165.0 \\ -165.0 & 2620.2 \end{bmatrix}$	$\begin{bmatrix} 5250.7 & -122.9 \\ -122.9 & 2595.7 \end{bmatrix}$
Facteur de normalisation	$q_i^d(k) ; q_i(k) ;$	4 ; 4	4 ; 4	4 ; 4
	$q_i(k-1) ; q_i(k-2)$	4 ; 4	4 ; 4	4 ; 4
	$R_i^{-1}(k) \forall k$	$1000 \times I_2$	$1000 \times I_2$	$1000 \times I_2$
	$P_{1i}^{j-1}(0) \quad j=1..8$	10000	10000	10000
	$P_2^{-1}(0)$	1000	1000	1000

Table (IV.3) : Paramètres du régulateur sous la loi CDFAD à gain décroissant

Avec I_2 la matrice unitaire de dimension 2.

Les différentes variables adaptatives de la loi de commande sont initialisées par zéro. Le pas d'échantillonnage est fixé à la valeur 5 ms.

Résultats et performances

Test sans charge : Suite à un test sans charge nous avons obtenu les résultats illustrés par les figures (IV.14a,b). Nous remarquons une bonne poursuite de la trajectoire de Leahvy. Les erreurs des trois articulations sont, dans l'ordre, inférieures à 0.06 deg, 0.5 deg et 0.2 deg et convergentes asymptotiquement vers zéro. Après un régime transitoire d'adaptation les commandes deviennent lisses. Les sorties des modèles flous, notées uf , ainsi que les signaux auxiliaires, ont une intervention principale dans les commandes générées. Nous constatons la décroissance des gains d'adaptation des matrices estimées de modèles flous. Cette décroissance continue même en régime permanent.

Test de variation de charge : Pour tester la capacité d'adaptation du régulateur, nous avons chargé l'élément terminal avec une masse de 2 Kg puis de 4 Kg jusqu'à l'instant $t=0.75$ sec. Les résultats obtenus sont montrés par la figure (IV.15). Nous remarquons que notre régulateurs prend en charge cette variation dynamique dès l'instant initiale. De même après la chute de cette masse, le robot continu remarquablement la poursuite des trajectoires. Les erreurs maximales sont inférieures à 0.5 deg.

Test de variation paramétrique: Pour tester la robustesse du régulateur lors d'incertitudes paramétriques, nous avons augmenté tous les paramètres du modèle dynamique du robot par une quantité de 50% puis de 100%. La figure (IV.16) montre les réponses obtenues. Nous constatons une poursuite satisfaisante des trajectoires désirées avec une léger augmentation des erreurs de poursuite qui sont maintenues inférieures à 0.5 deg. Ce qui confirme la robustesse du régulateur.

Test de rupture de commande : Pour montrer l'efficacité de la décentralisation dans le schéma proposé, nous avons appliqué une rupture de la commande de la deuxième articulation à partir de l'instant $t=0.75$ sec (commande nulle). Les réponses enregistrées, illustrées par la figure (IV.17), montre l'efficacité de l'indépendance des régulateurs associés aux différentes articulations. En effet, les erreurs de poursuites sont inférieures à 0.2 deg malgré le couplage important existant entre articulations.

Test de variation du pas d'échantillonnage : Dans ce test nous avons augmenté le pas d'échantillonnage à 10 ms. Pour maintenir des bonnes performances (y compris la stabilité), il était nécessaire de diminuer les valeurs de certains paramètres du régulateur comme indique la table (IV.4).

Symboles des paramètres	Première articulation $i=1$	Deuxième articulation $i=2$	Troisième articulation $i=3$
δ_{0i}	25000	10000	50000
δ_{1i}	4375	17500	5000
δ_{2i}	-3375	-15000	-4583
pôles de S_i	-0.4652; 0.2902	-0.4846; 0.3096	-0.3569; 0.2569
Q_i	$2500 \times I_2$	$2500 \times I_2$	$2500 \times I_2$
S_{ji}	$\begin{bmatrix} 885.02 & -24.17 \\ -24.17 & 432.79 \end{bmatrix}$	$\begin{bmatrix} 1335.00 & -41.20 \\ -41.20 & 655.00 \end{bmatrix}$	$\begin{bmatrix} 2552.10 & -25.80 \\ -25.80 & 1271.4 \end{bmatrix}$

Table (IV.4) : Paramètres du régulateur sous la loi CDFAD à gain décroissant

Les réponses obtenues sont montrées par la figure (IV.18). Nous remarquons une poursuite satisfaisante des trajectoires désirées. Les erreurs maximales sont inférieures à 0.8 deg qui convergent asymptotiquement vers zéro.

IV.4. Conclusion

Dans ce chapitre nous avons proposé et appliqué deux techniques de commande décentralisée floue adaptative discrète (CDFAD) pour la commande en poursuite des robots manipulateurs : à gain constant et à gain décroissant.

Ayant des structures facile à implémenter, les deux lois de commande n'exigent aucune information structurelle ou paramétrique sur le modèle dynamique du robot. La décentralisation dans le schéma de commande permet un dimensionnement plus pratique, en considérant chaque articulation comme un sous-système indépendant. De plus, seules les mesures successives des positions sont nécessaires dans le calcul de la commande.

A la lumière des réponses enregistrées, nous constatons une très bonne poursuite de la trajectoire de Leahvy. L'adaptation du modèle flou ainsi que du signal auxiliaire est caractérisée par un apprentissage lent.

Les différents tests démontrés à travers les résultats obtenus nous ont permis de juger positivement la stabilité et l'efficacité de ces algorithmes lors de conditions opératoires variables.

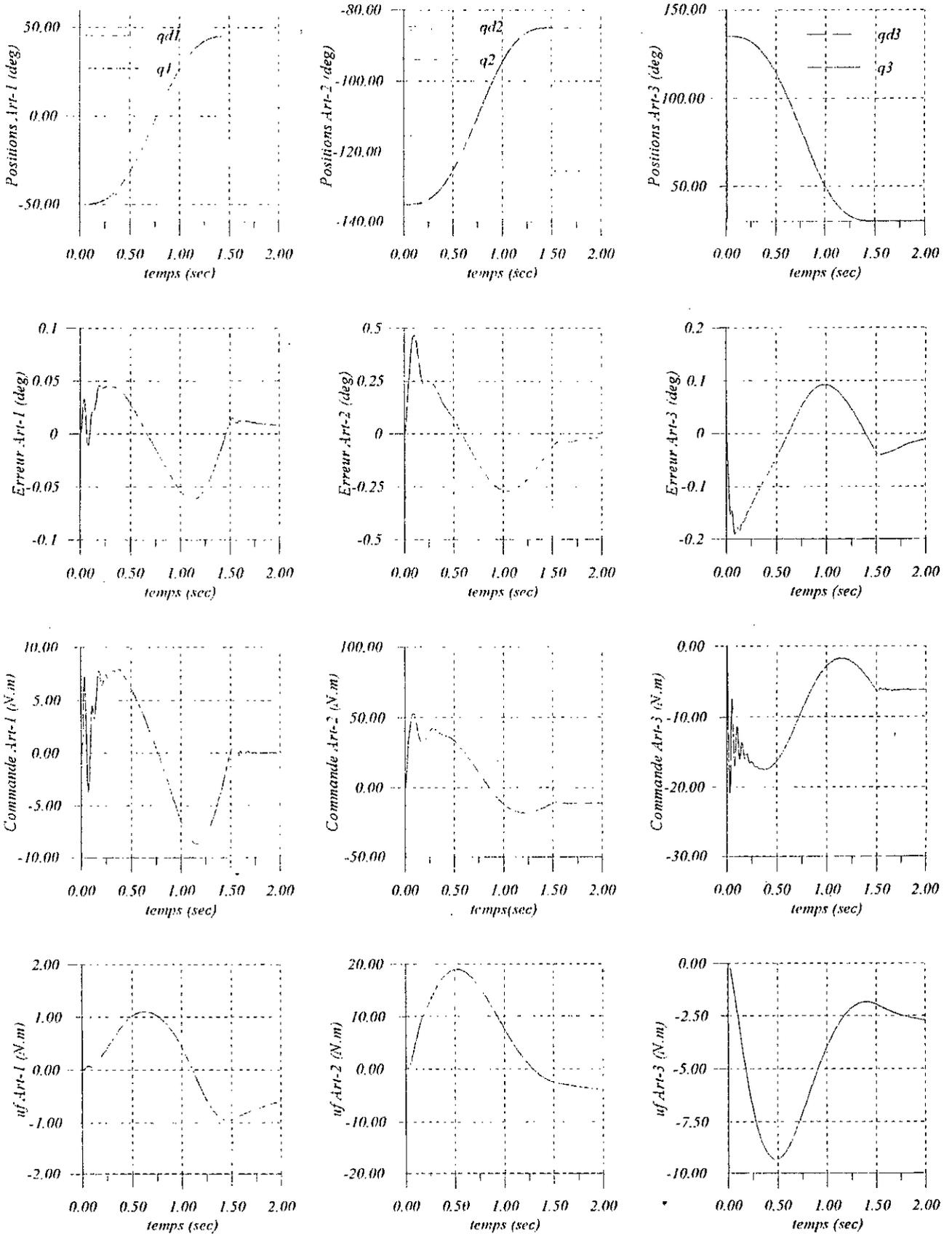


figure (IV.14a) : Réponses des différentes articulations soumises à la loi CDFAD à gain décroissant :
Test sans charge

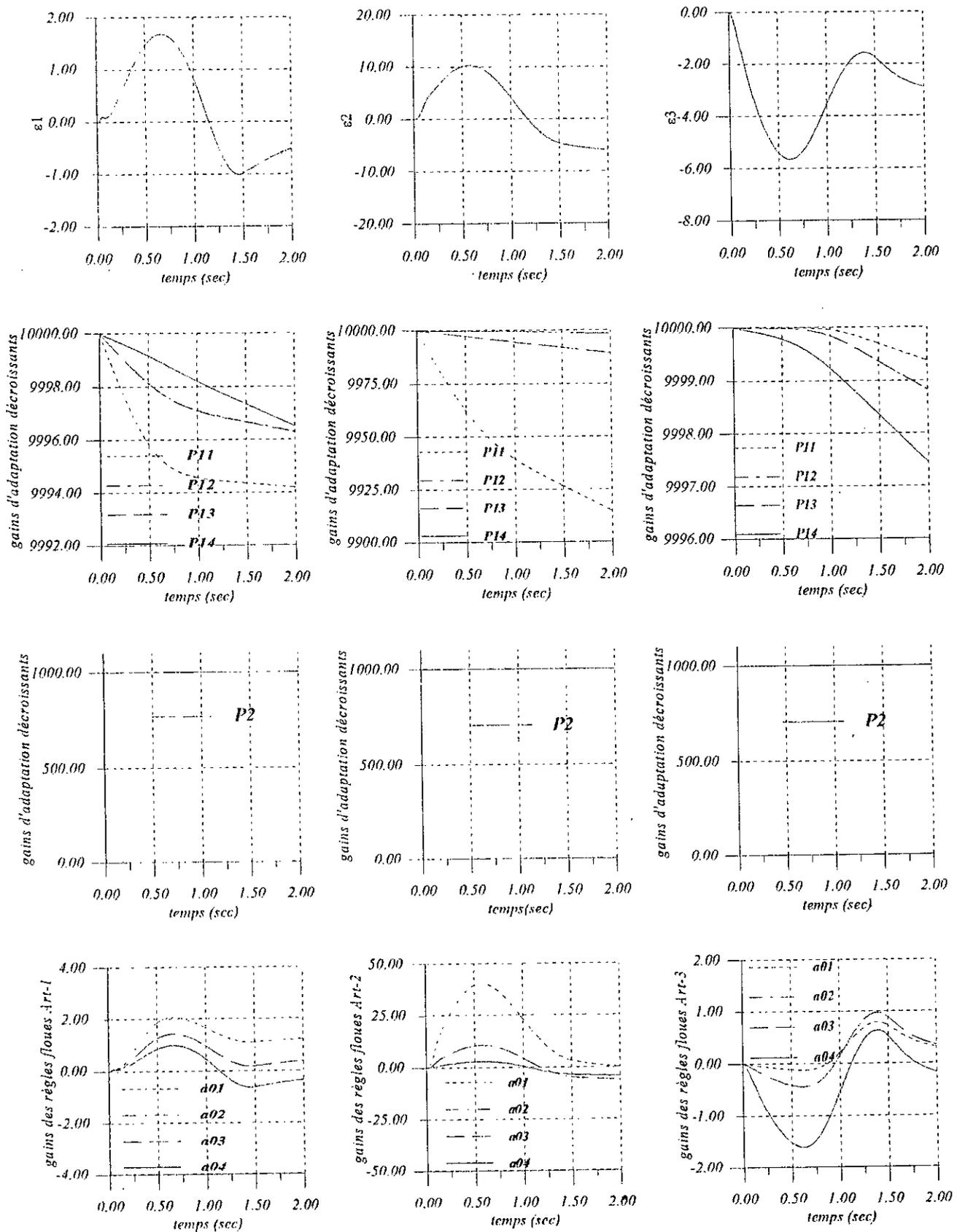


figure (IV.14b) : Réponses des différentes articulations soumises à la loi CDFAD à gain décroissant :
Test sans charge

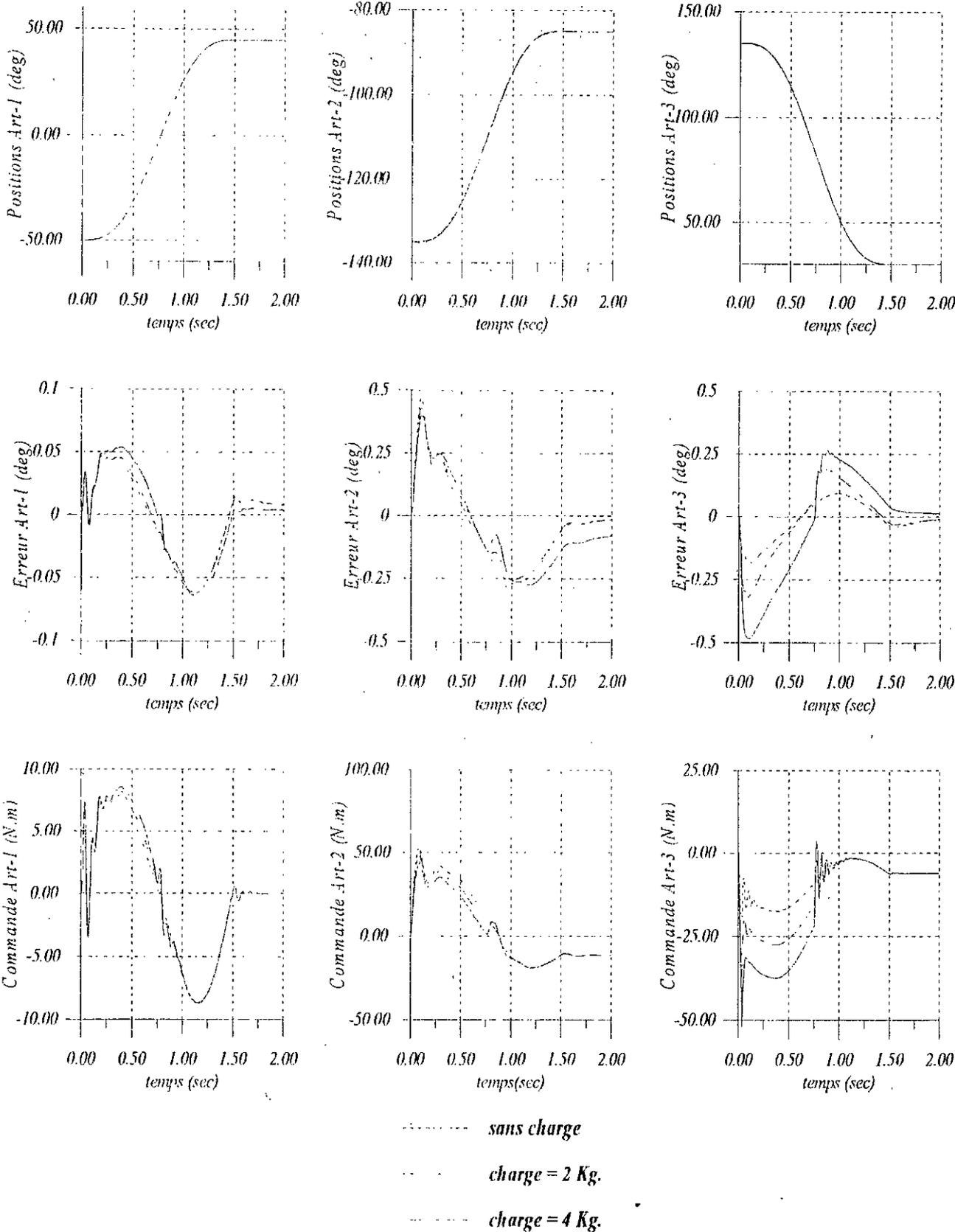


figure (IV.15) : Réponses des différentes articulations soumises à la loi à gain décroissant :
Test de variation de charge

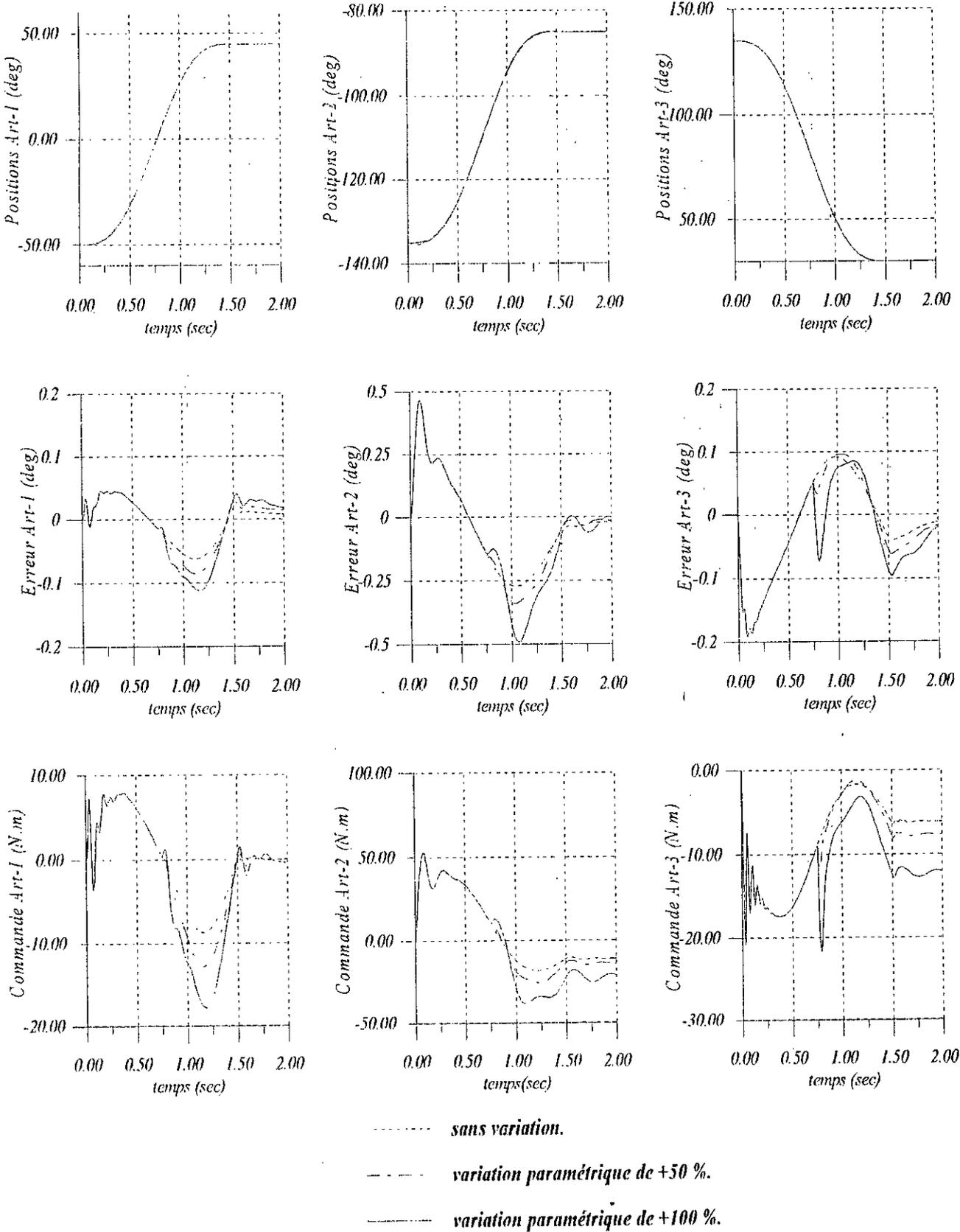


figure (IV.16) : Réponses des différentes articulations soumises à la loi à gain décroissant : Test de variation paramétrique

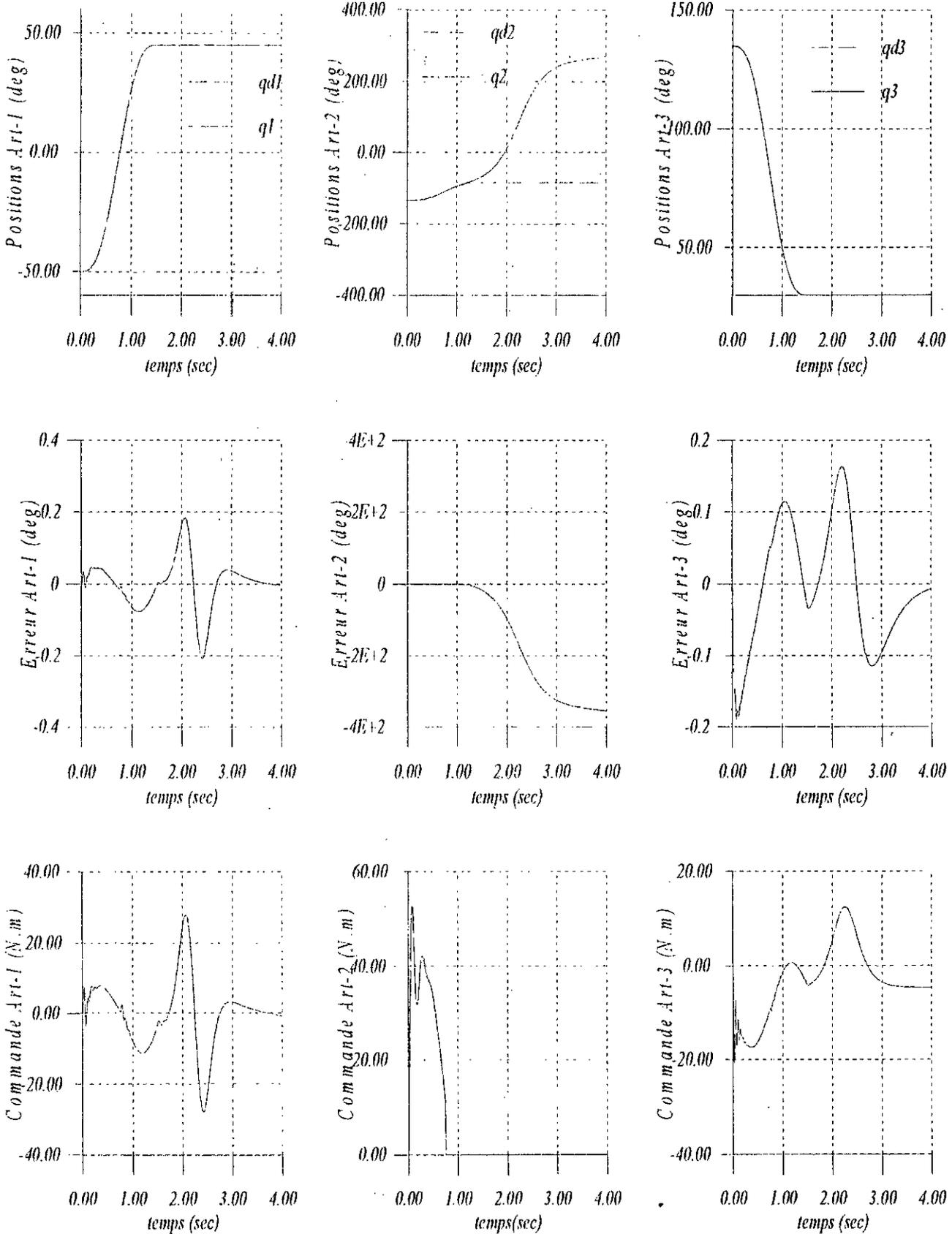


figure (IV.17) : Réponses des différentes articulations soumises à la loi CDI'AD à gain décroissant :
Test de rupture de commande

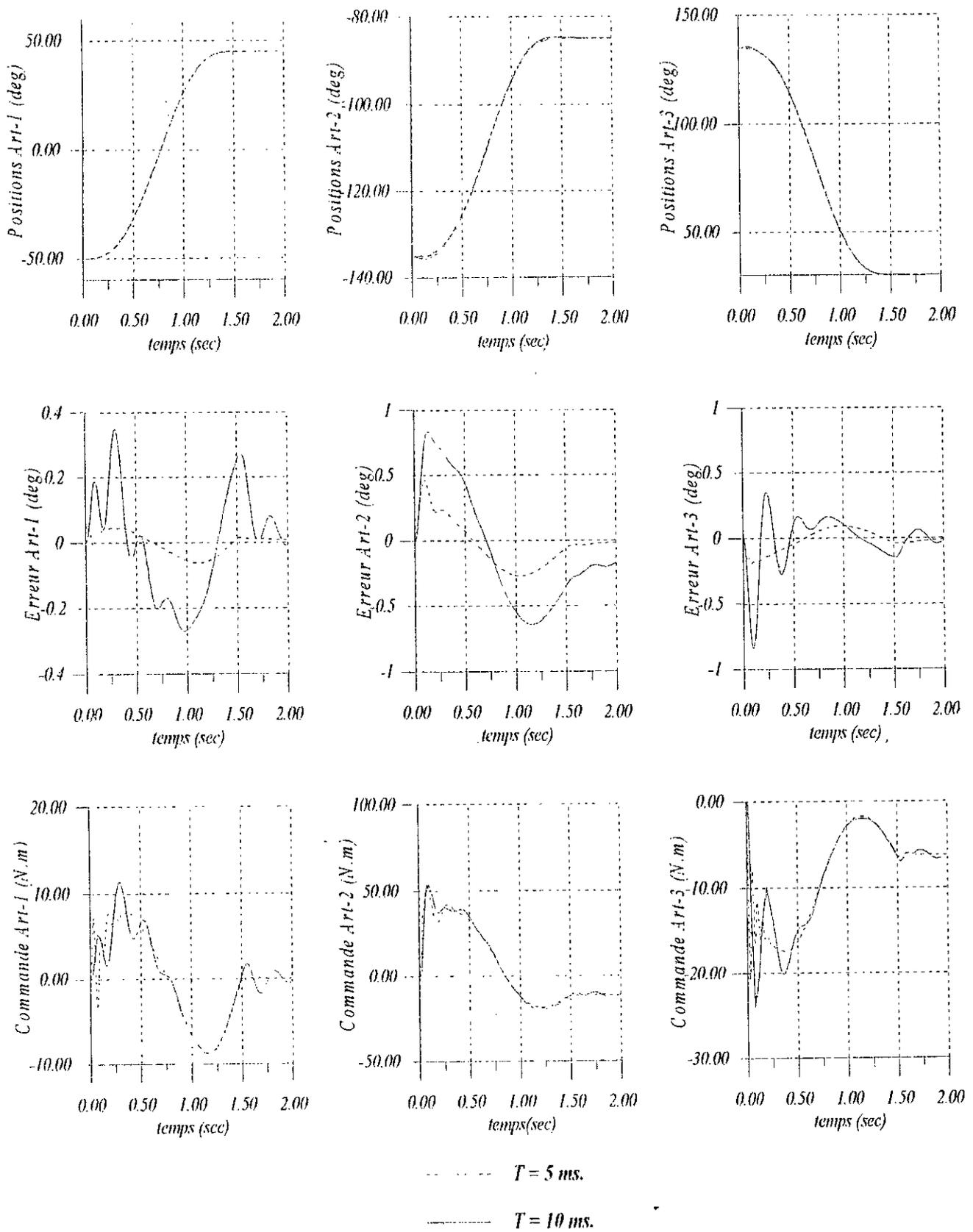


figure (IV.18) : Réponses des différentes articulations soumises à la loi CDFAD à gain décroissant : Test de variation du pas d'échantillonnage

*Conclusion générale
et
perspectives*

Conclusion générale et perspectives

Dans ce travail, nous nous sommes intéressés à la commande décentralisée des robots manipulateurs en utilisant des systèmes flous appliquée au problème de poursuite de trajectoires de référence. Ainsi, nous avons développé une commande floue associée à un réglage à modes glissants et plusieurs techniques de commande décentralisée floue adaptative continue et discrète.

L'application par simulation est effectuée sur le robot PUMA560 en considérant les trois premières articulations. Ce célèbre robot est souvent considéré comme un robot de test vu la complexité et le couplage important qui caractérisent sa dynamique. Pour valider et évaluer les performances des régulateurs associés au robot PUMA560, nous avons utilisé la trajectoire rapide de Leahvy. Cette trajectoire permet une excitation de toutes les dynamiques de couplages entre les articulations pendant une courte durée comparée à la dynamique du robot.

Nous avons tous d'abord commencer par l'élaboration de la formulation mathématique des systèmes flous utilisant les deux modèles : modèles de Mamdani et de Sugeno. Le premier modèle est mieux adapté à modéliser une description linguistique de comportement que doit réaliser le régulateur. Tandis que le modèle de Sugeno est souhaitable pour une identification, compensation et approximation de fonctions inconnues et/ou variables ce qui est intéressant dès qu'on sache que la sortie due à ce type de modèle possède une expression linéaire en terme des gains des conséquences des règles floues. En effet, cela permet de faciliter l'exploitation de la propriété d'approximation universelle dont sont dotés certaines classes de ces systèmes.

Ayant une structure décentralisée, la commande floue associée à un réglage à modes glissants, que nous avons proposé, se base sur un modèle flou de type Mamdani. Dans la table de décision interviennent deux variables de distances comme variables d'entrée et la commande comme variable de sortie. Les deux variables de distance permettent de quantifier l'éloignement du point d'état par rapport à une surface de glissement présélectionnée dans deux directions opposées et non dans une seule direction comme dans le cas des régulateurs à modes glissants avec couche bornée ("*boundary layer*"). Par rapport au régulateur flou basé sur la table très connue de Mac Vichar, le régulateur étudié permet d'insérer l'erreur de poursuite, ses dérivées d'ordres supérieures et son intégration dans le calcul de la commande sans que la table de décision ne soit augmentée ou même modifiée et d'imposer directement la dynamique de convergence de l'erreur via le choix des paramètres de la surface de glissement. Avec ce même choix, des régulateurs flous dont la fonction est similaire à celle des régulateurs classiques de type PI, PD et PID^(m) peuvent être obtenus. De même notre régulateur permet l'adoption des surfaces de glissement non linéaires. Nous avons proposé des directives de dimensionnement pour choisir la structure et pour déterminer tous les paramètres du régulateur convenablement. Par ailleurs, grâce au caractère décentralisé de la commande et aux vues des résultats obtenus, des très bonnes performances sont obtenues (très faibles erreurs qui s'annulent en régime permanent et commandes réalisables) en adoptant dans chacune des articulations la structure la plus appropriée. Avec un nombre de règle relativement réduit (huit règles), le régulateur fait preuve d'une assez bonne robustesse vis-à-vis des variations de charge et paramétrique.

Les techniques de commande décentralisée floue adaptative continue sont basées sur la compensation des termes inconnus à l'aide des systèmes flous en utilisant les mesures de la position, la vitesse et l'accélération. Ces techniques supposent une adaptation rapide au niveau du régulateur ainsi qu'une variation lente des éléments diagonaux de la matrice d'inertie. Pour chaque articulation, le schéma de commande fait intervenir trois actions principales : une action de stabilisation réalisée par un régulateur de type PD, une action de compensation effectuée par un modèle flou adaptatif de type Sugeno et une troisième action à l'aide d'un signal auxiliaire pour remédier à l'erreur de reconstruction. Selon que les gains du régulateur PD sont adaptatifs ou non, nous distinguons deux approches.

Utilisant un régulateur PD adaptatif, la première approche nécessite les mesures de la position et de la vitesse. La première technique appelée commande décentralisée floue adaptative (CDFA) utilise un modèle flou dont le nombre de gains adaptatifs est égale à $nf^3 \times np$. Dans la deuxième technique, commande décentralisée floue adaptative robuste (CDFAR), ce nombre est réduit à $nf^2 \times np$. Cependant cette technique suppose l'existence et la connaissance de bornes supérieures aux éléments diagonaux de la matrice d'inertie. Cette supposition restreint le nombre des robots considérés par cette technique. Un lissage de la fonction discontinue dans la loi de commande est nécessaire pour obtenir des commandes réalisables. Dans ce cas, la stabilité du schéma de commande n'est plus assurée. Nous avons également proposée la technique de commande décentralisée floue adaptative modifiée (CDFAM) pour palier à la discontinuité de la loi de commande issue de la CDFAR tout en assurant la stabilité du schéma de commande. Ayant un modèle flou à $nf^2 \times np$ gains adaptatifs, cette technique est applicable aux différentes classes des robots manipulateurs. L'analyse effectuée via des fonctions de Lyapunov candidates démontre la stabilité des schémas de commande, la convergence des erreurs de poursuite vers zéro et la bornitude des différentes variables adaptatives y compris la commande et les variables d'état.

Dans la deuxième approche, l'action de stabilisation consiste en un régulateur PD à gains constants. Le calcul de la commande nécessite la disponibilité de la mesure de l'accélération. Les lois d'ajustements des gains adaptatifs du modèle flou proposées respectent le principe de perturbation minimale. Cela permet une identification et une intervention selon des modèles locaux définis par la distribution des fonctions d'appartenance adoptées tout en offrant une interpolation entre les différents modèles locaux via le vecteur d'activation. Cela conduit à un soulagement des calculs dans le régulateur global. La loi particulière d'évolution des matrices et gains d'adaptations assure une meilleure robustesse vis-à-vis des variations paramétriques ou structurelles dans le modèle du robot. La technique de commande décentralisée floue adaptative simple (CDFAS) utilise un modèle flou qui comporte $nf^3 \times np$ gains adaptatifs. La technique de commande décentralisée floue adaptative robuste (CDFAR) permet de réduire ce nombre à $nf^2 \times np$ dans le cas des robot dont les éléments diagonaux de la matrice d'inerties peuvent être bornées par des constantes connues. Afin d'éviter des oscillations intempestives et/ou une activation maximale de la commande, on introduit un lissage sur la fonction discontinue dans la loi de commande. Également, la technique de commande décentralisée floue adaptative à zone morte (CDFAZM) permet de maintenir le nombre réduit des gains adaptatifs $nf^2 \times np$ tout en assurant la convergence asymptotique de l'erreur de poursuite vers une zone voisine du zéro. La taille de cette zone est indépendante des paramètres du robot et peut être réduite considérablement par un choix convenable des gains du régulateur.

Grâce au caractère décentralisé adopté dans la démonstration, les différentes techniques de commande décentralisée floue adaptative peuvent être combinées pour obtenir des meilleurs performances tout en assurant la stabilité du système global. Dans ce cas, nous adoptons pour chaque articulation la technique de commande la plus appropriée. Particulièrement les techniques de CDFAR sont mieux adaptées aux articulations soumises à un effet gravitationnel considérable.

A la lumière des réponses enregistrées, les différentes techniques de commande décentralisée floue adaptative présentent de bonnes performances malgré que les différentes variables adaptatives y comprise la commande sont initialisées par zéro et le modèle flou n'utilise que le nombre minimale de règles (huit ou quatre règles). Les techniques de la deuxième approche offrent des performances nettement supérieures (erreur quasiment nulle qui s'annulent en régime permanent) au prix de la nécessité de disponibilité de la mesure de l'accélération.

Les techniques de commande décentralisée floue adaptative discrète proposées sont universelles dans le sens où elles sont applicables à toutes les classes des robots rigides sans que des informations aux préalables sur le modèle dynamique ne soient exigées. Un modèle flou adaptatif est

adopté pour faire face aux incertitudes dans le modèle dynamique discret. Seules les mesures successives de la position sont utilisées. Ainsi, dans la loi de commande interviennent trois actions principales : une action de stabilisation via un filtre linéaire, une action de compensation des termes inconnus réalisée par un modèle flou de Sugeno adaptatif et une troisième action pour remédier à l'erreur de reconstruction. Le modèle flou utilise quatre variables d'entrée. Ainsi le nombre des gains adaptatifs est égale à $nf \times np$. Selon la nature des gains d'adaptation, nous avons proposées deux lois de commande.

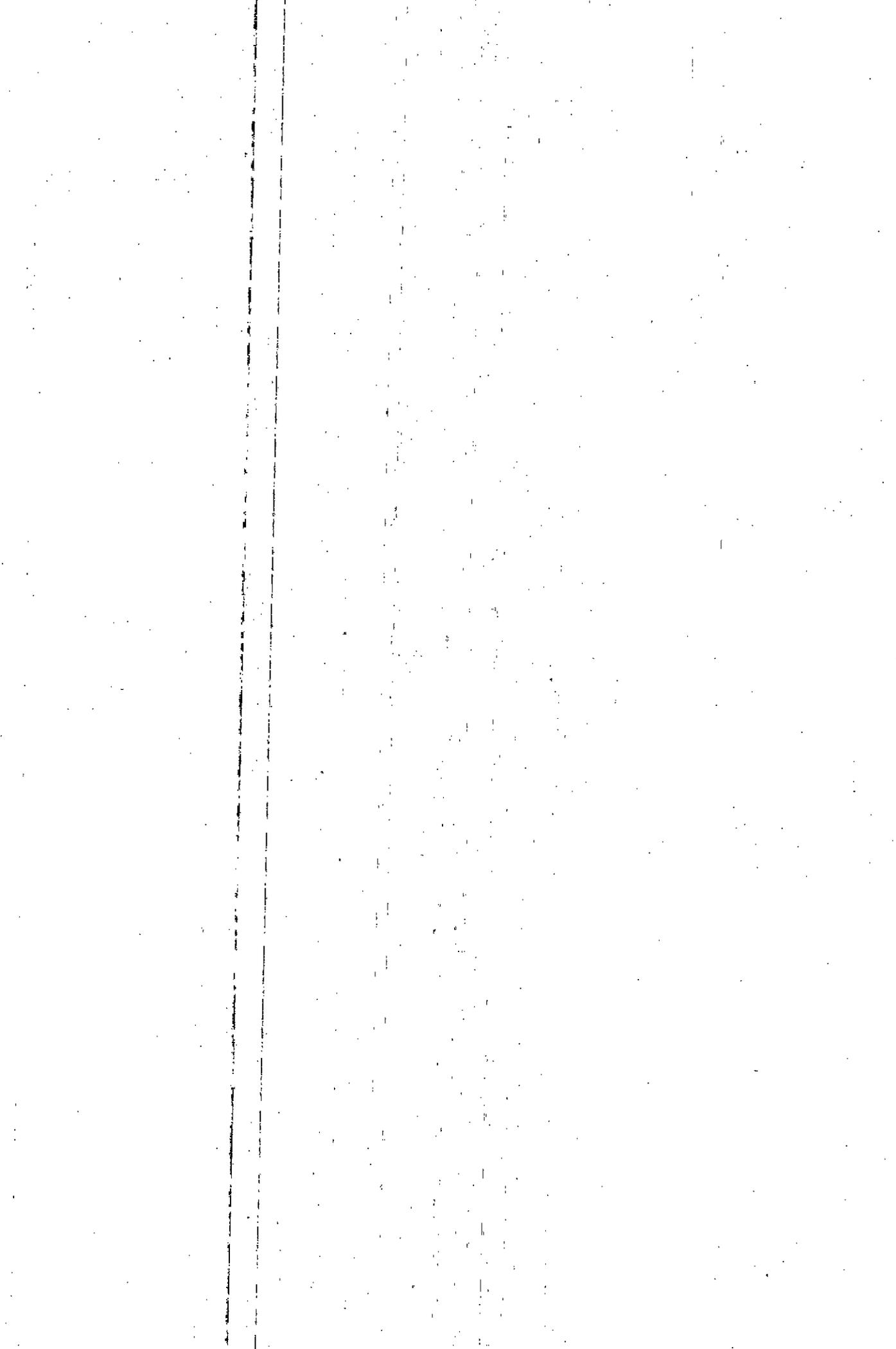
Dans la première loi de commande, les variables adaptatives (gains adaptatifs du modèle flou et signal auxiliaire) sont ajustés via une loi à gain d'adaptation constant. Dans la deuxième technique, la loi d'ajustement des variables adaptatives utilisent des gains d'adaptations décroissants. Ce type de gains, particulièrement intéressant lors de l'identification, de la compensation et de l'approximation de dynamiques à paramètres variables dans le temps est obtenu en appliquant l'approche d'hyperstabilité dans sa version discrète. Ayant la même structure, les gains adaptatifs de chaque règle floue possèdent leur propre gain d'adaptation décroissant ce qui reflète le caractère de modèles locaux. Une interpolation et une continuité entre les différentes règles floues est assurée via le vecteur d'activation à deux niveaux différents : au niveau d'identification, lors de l'ajustement des gains décroissants et des gains des règles floues, et au niveau d'intervention dans la commande, via la normalisation dans l'expression de la sortie du modèle flou. Comparé au cas continu, l'apprentissage du modèle flou dans le cas discret est caractérisé par une dynamique lente. Cela est dû principalement à la non consistance des informations rapportées par les variables d'entrées. En effet, les différentes variables sont l'image de la même grandeur physique (positions articulaires désirée et réelles).

En général, avec un choix approprié des différentes pondérations et des paramètres arbitraires, on modifie la contribution des actions dans la commande ce qui permet d'obtenir les performances satisfaisantes. De même la décentralisation dans le schéma de commande présente l'avantage d'un dimensionnement indépendant des différents régulateurs locaux.

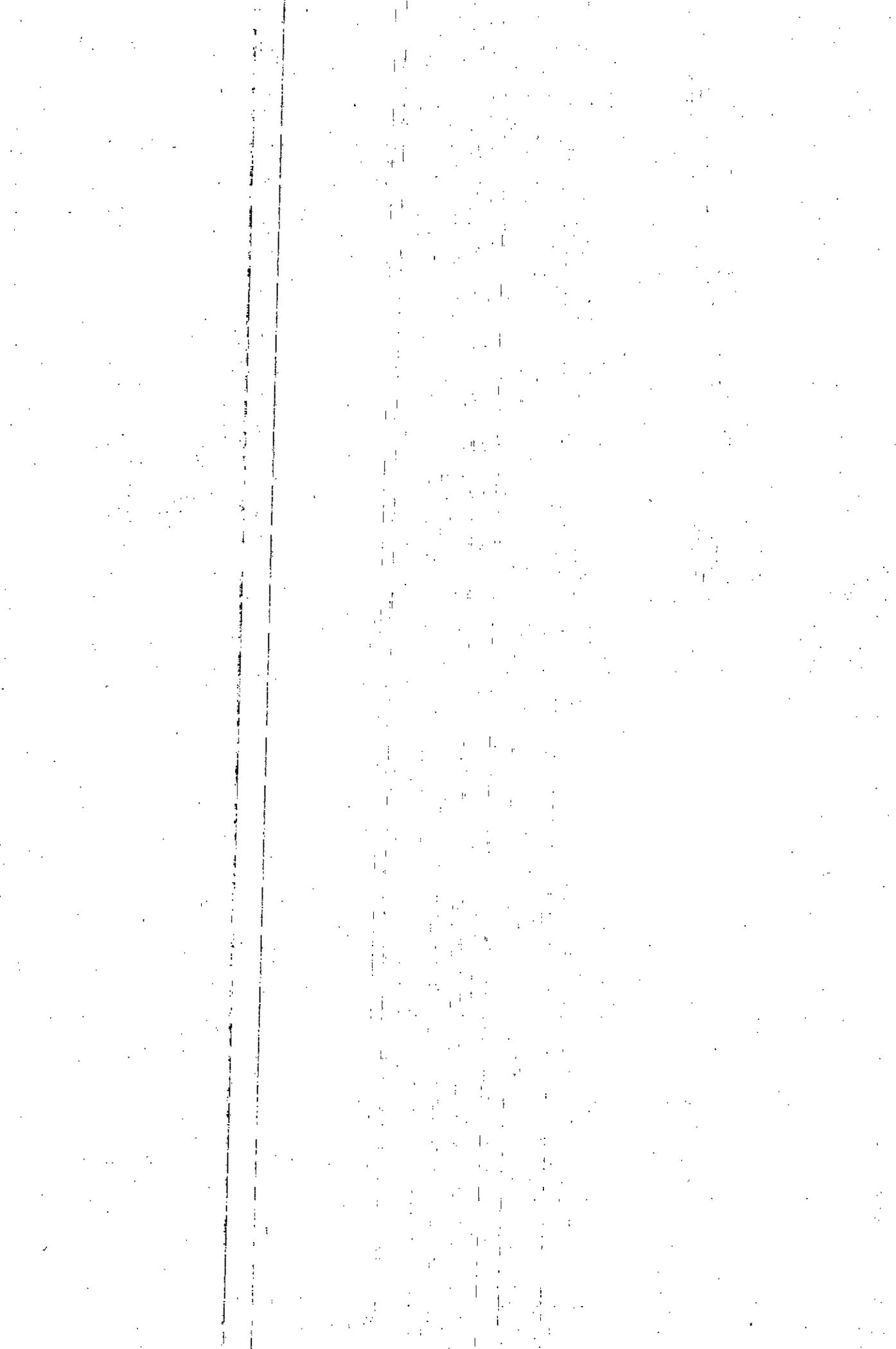
Les techniques de commande adaptative continue et discrète sont basées sur l'hypothèse d'une adaptation rapide des variables adaptatives du régulateur et une variation lente des paramètres du modèle du robot. Au vue des résultats de simulation obtenus, nous constatons de très bonnes performances et une assez bonne robustesse des schémas de commande vis-à-vis d'une variation paramétrique et non paramétrique et cela malgré la violation des hypothèses considérés. Cependant, l'augmentation du pas d'échantillonnage exige la diminution des gains d'adaptation si des bonnes performances sont recherchées. Le test de rupture de commande a permis de mettre en valeur l'aptitude des schémas de commande à préserver des bonnes performances en présence de défaillance d'un actionneur.

A l'issue de l'étude effectuée et des résultats obtenus présentées dans ce mémoire, nous envisageons les perspectives de continuité suivantes :

- ❖ L'application de la loi de commande floue associée à un réglage à modes glissants à des robots flexibles, à la machine électrique et aux systèmes d'ordre supérieure à deux;
- ❖ L'extension de la loi de commande décentralisée floue adaptative à zone morte à toutes les classes des robots et l'utilisation de zone morte adaptative pour remédier à l'erreur permanente;
- ❖ L'association d'observateurs décentralisés aux techniques de commande de la deuxième approche (CDFAS, CDFAR et CDFAZM). Ces observateurs seront utilisés pour estimer les accélérations réelles;
- ❖ L'application des techniques proposées pour la commande en poursuite dans l'espace opérationnel (cartésien). En effet, les modèles du robot dans l'espace généralisé et dans l'espace cartésien possèdent la même structure et des propriétés similaires. Cependant, le caractère de décentralisation sera perdu au niveau de calcul des commandes généralisées à partir des forces cartésiennes virtuelles.



Annexes



Annexe A :**Présentation du robot PUMA560 à trois degrés de liberté, modèle dynamique et trajectoires de référence**

En général, la dynamique de mouvement des différentes articulations d'un robot manipulateur est liée au commande issues des différentes actionneurs par un ensemble d'équations différentielles du second ordre qui constituent le modèle dynamique du robot. Les approches d'Euler-Lagrange et de Newton-Euler permettent d'aboutir au modèle dynamique. Ce dernier peut être exprimé sous la forme contractée suivante [PAU86][FU87]:

$$M(q)\ddot{q} + N(q, \dot{q}) + G(q) + H(\dot{q}) = u(t) \quad (\text{A.1})$$

avec,

$q \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$ et $\ddot{q} \in \mathbb{R}^n$: représentent, respectivement, les positions, les vitesses et les accélérations articulaires ;

$M(q) \in \mathbb{R}^{n \times n}$: matrice d'inertie.

$N(q, \dot{q}) \in \mathbb{R}^n$: couples et/ou forces dus aux termes de Coriolis et centrifuges ;

$G(q) \in \mathbb{R}^n$: vecteur des forces et/ou couples visqueux et secs ;

Pour l'application nous avons choisit le robot manipulateur PUMA560 à trois degrés de liberté avec des mouvements tous rotationnels. Le robot original possède six degrés de liberté. Dans notre cas, les trois dernières articulation sont considérées à l'arrêt. Ainsi, notre robot réalise trois mouvements de rotations, le premier suivant l'axe horizontal, le second et le troisième selon des axes verticaux. Son schéma de principe est illustré par la figure (A.1).

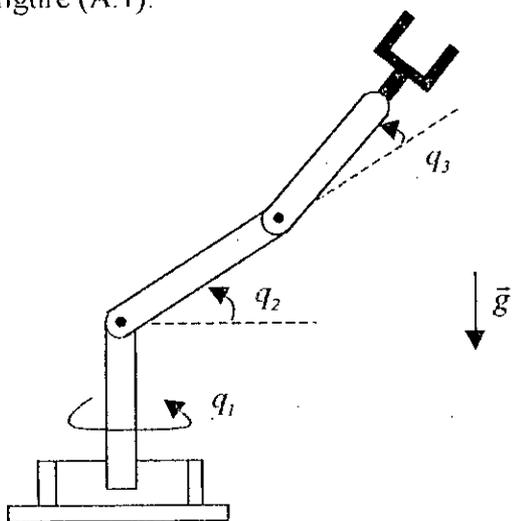


figure (A.1) : Schéma simplifié du robot PUMA560 à trois degrés de liberté.

Modèle dynamique du robot PUMA560 à trois degrés de liberté :

Le modèle dynamique de du robot considéré est donné par [ARM86][MAD97]:

$$(M(q) + \Delta M(q))\ddot{q} + N(q, \dot{q}) + G(q) + u_{m_0} = u(t) \quad (\text{A.2})$$

avec,

$$M(q) = \begin{bmatrix} I_1 + I_2 c_{23}^2 + I_3 c_2^2 + I_4 c_2 c_{23} & I_5 s_{23} + I_6 s_2 & I_5 s_{23} \\ I_5 s_{23} + I_6 s_2 & I_7 + I_4 c_3 & I_8 + 0.5 I_4 c_3 \\ I_5 s_{23} & I_8 + 0.5 I_4 c_3 & I_9 \end{bmatrix} \quad (\text{A.3})$$

$$N(q, \dot{q}) = \begin{bmatrix} -(2(I_3 s_2 c_2 + I_2 s_{23} c_{23}) + I_4 (c_2 s_{23} + s_2 c_{23})) \dot{q}_1 \dot{q}_2 - (2I_2 s_{23} c_{23} + I_4 c_2 s_{23}) \dot{q}_1 \dot{q}_3 \\ + (I_6 c_2 + I_5 c_{23}) \dot{q}_2^2 + (2I_5 c_{23}) \dot{q}_2 \dot{q}_3 + (I_5 c_{23}) \dot{q}_3^2 \\ (I_3 c_2 s_2 + I_2 c_{23} s_{23} + 0.5 I_4 (s_2 c_{23} + c_2 s_{23})) \dot{q}_1^2 - (I_4 s_3) \dot{q}_2 \dot{q}_3 - (0.5 I_4 s_3) \dot{q}_3^2 \\ (I_3 s_{23} c_{23} + 0.5 I_4 c_2 s_{23}) \dot{q}_1^2 + (0.5 I_4 s_3) \dot{q}_2^2 \end{bmatrix} \quad (\text{A.4})$$

$$G(q) = \begin{bmatrix} 0 \\ -(m_3 l_2 + 0.5 m_2 l_2) g c_2 - 0.5 m_3 l_3 g c_{23} \\ -0.5 m_3 l_3 g c_{23} \end{bmatrix} \quad (\text{A.5})$$

$$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} \quad (\text{A.6})$$

où

$$c_i = \cos(q_i); s_i = \sin(q_i); c_{ij} = \cos(q_i + q_j); s_{ij} = \sin(q_i + q_j)$$

et

$$\begin{cases} I_1 = I_{yy1} + I_{xx2} + m_2 d_2 (d_2 + e) + m_3 d_2^2 + I_{xx3} + I_{xx1} + m_1 d_2^2 + I_{M1} \\ I_2 = I_{yy3} - I_{xx3} + I_{yy1} - I_{xx1} + m_1 l_3^2 \\ I_3 = I_{yy2} - I_{xx2} + m_3 l_2^2 + m_1 l_2^2 \\ I_4 = m_3 l_2 l_3 + 2 m_1 l_2 l_3 \\ I_5 = 0.5 m_3 l_3 d_2 + m_1 l_3 d_2 \\ I_6 = 0.5 m_2 l_2 (d_2 + e) + m_3 d_2 l_2 + m_1 d_2 l_2 \\ I_7 = I_{zz2} + I_{zz3} + m_3 l_2^2 + I_{zz1} + m_1 (l_2^2 + l_3^2) + I_{M2} \\ I_8 = I_{zz3} + I_{zz1} + m_1 l_3^2 \quad I_9 = I_8 + I_{M3} \end{cases} \quad (\text{A.7})$$

$$\begin{cases} \Delta M_{11} = I_{xx1} + m_1 d_2^2 + I_{M1} + (I_{yy1} - I_{xx1} + m_1 l_3^2) c_{23}^2 + m_1 l_2^2 c_2^2 + 2 m_1 l_2 l_3 c_2 c_{23} \\ \Delta M_{12} = m_1 l_3 d_2 s_{23} + m_1 d_2 l_2 s_2 \\ \Delta M_{13} = m_1 l_3 d_2 s_{23} \end{cases} \quad (\text{A.8})$$

$$\begin{cases} \Delta M_{22} = I_{zz1} + m_1 (l_2^2 + l_3^2) + I_{M2} + 2 m_1 l_2 l_3 c_3 \\ \Delta M_{23} = I_{zz1} + m_1 l_3^2 + m_1 l_2 l_3 c_3 \\ \Delta M_{33} = I_{zz1} + m_1 l_3^2 + I_{M3} \end{cases} \quad (\text{A.9})$$

I_{Mi} moments d'inertie des différents moteurs.

$I_{xxi}, I_{yyi}, I_{zzi}$. moments d'inertie totale par rapport aux principaux axes de l'effecteur.

u_{m0} : représente le couple supplémentaire dû à la charge portée de masse m_0 . Il est donnée par :

$$u_{m0} = m_0 J^T(q) [J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} + g] \quad (\text{A.10})$$

où, $J(q), \dot{J}(q, \dot{q})$ désignent la matrice jacobienne et sa dérivée avec :

$$J(q) = \begin{bmatrix} -s_1(l_2 c_2 + l_3 c_{23}) - d_2 c_1 & -c_1(l_2 s_2 + l_3 s_{23}) & -c_1(l_3 s_{23}) \\ c_1(l_2 c_2 + l_3 c_{23}) - d_2 s_1 & -s_1(l_2 s_2 + l_3 s_{23}) & -s_1(l_3 s_{23}) \\ 0 & -(l_2 c_2 + l_3 c_{23}) & -(l_3 c_{23}) \end{bmatrix} \quad (\text{A.11})$$

avec $g = [0 \ 0 \ 9.81]^T$.

Les paramètres réels de ce robot sont comme suivant :

masses des différentes liaisons :

$$\begin{aligned} m_2 &= 17.40 \text{ kg} & m_3 &= 5.04 \text{ kg} & m_4 &= 0.82 \text{ kg} \\ m_5 &= 0.35 \text{ kg} & m_6 &= 0.09 \text{ kg} & m_1 &= m_4 + m_5 + m_6 = 1.26 \text{ kg}. \end{aligned}$$

Paramètres géométriques

$$d_2 = 149.09 \text{ mm} \quad l_2 = 431.8 \text{ mm} \quad l_3 = 433.07 \text{ mm}.$$

Les paramètres d'inertie sont donnée par la table (A.1):

N° de la liaison	I_{xxi} [kg m^2]	I_{yyi} [kg m^2]	I_{zzi} [kg m^2]	I_{Mi} [kg m^2]
1	-	$350 \cdot 10^{-3}$	-	1.14
2	$130 \cdot 10^{-3}$	$524 \cdot 10^{-3}$	$539 \cdot 10^{-3}$	4.71
3	$192 \cdot 10^{-3}$	$15.4 \cdot 10^{-3}$	$212 \cdot 10^{-3}$	0.83
4	$1.30 \cdot 10^{-3}$	$1.80 \cdot 10^{-3}$	$1.80 \cdot 10^{-3}$	-
5	$0.30 \cdot 10^{-3}$	$0.30 \cdot 10^{-3}$	$0.40 \cdot 10^{-3}$	-
6	$0.04 \cdot 10^{-3}$	$0.15 \cdot 10^{-3}$	$0.15 \cdot 10^{-3}$	-
4+5+6	$1.64 \cdot 10^{-3}$	$2.25 \cdot 10^{-3}$	$2.35 \cdot 10^{-3}$	-

table (A.1) : Paramètres d'inertie pour le robot PUMA 560.

Les réponses indicielles des différentes articulations sont montrées par la figure (A.2)

Le robot choisit constitue un modèle typique pour l'évaluation d'algorithme de commande, particulièrement, ceux ayant une structure décentralisée. En effet, le modèle dynamique est caractérisé par des fonctions non linéaire très complexes avec des forts couplages entre articulations. De plus la dynamique des différentes articulations est instable.

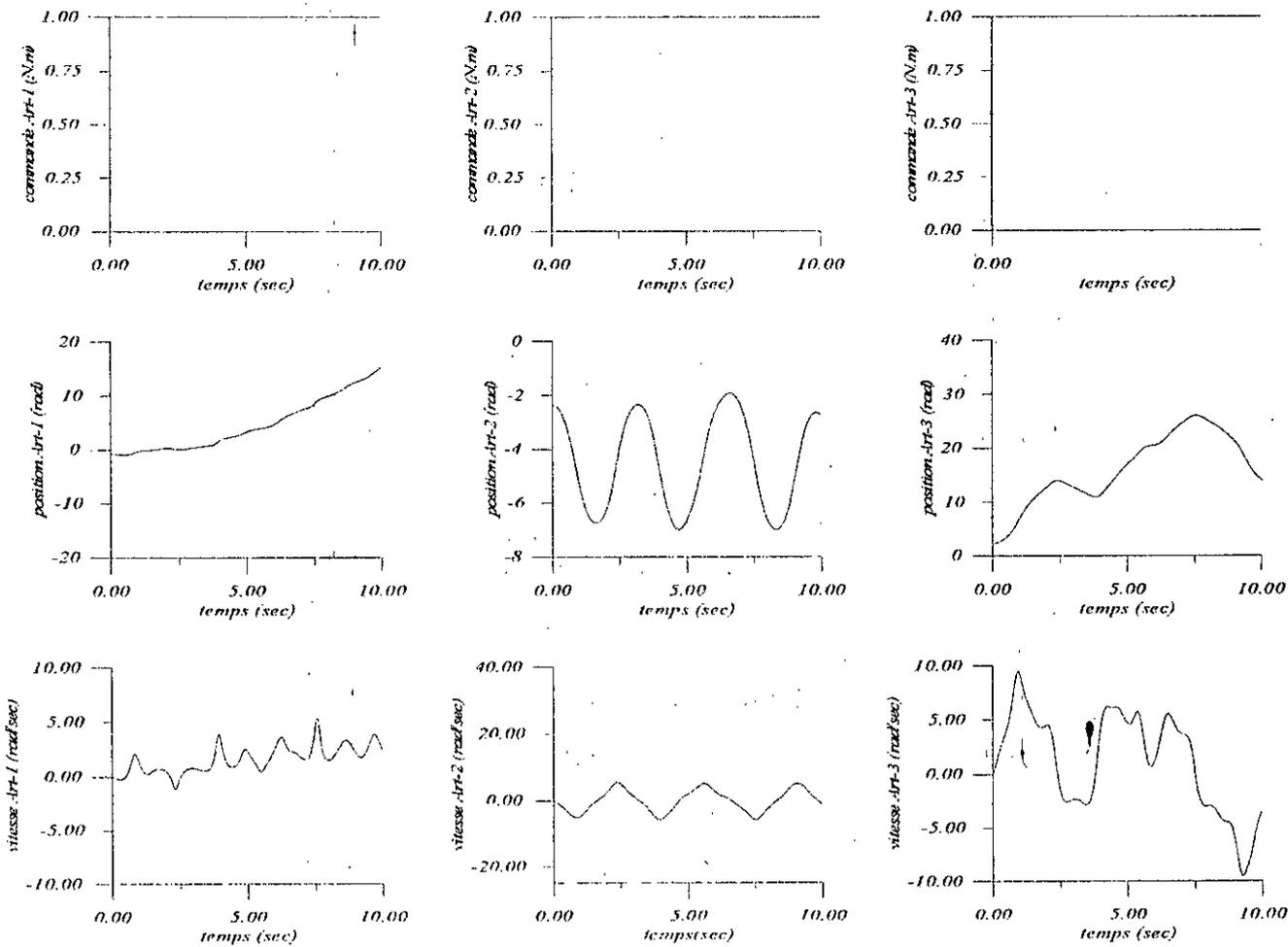


figure (A.2) Réponses indicielles des trois premières articulations du robot PUMA560

Trajectoires de référence

La dynamique des robot exige l'imposition de trajectoires de référence réalisable afin d'assurer le déplacement de l'élément terminal du robot d'un point initial q^{di} vers un point final q^{df} avec des commandes admissibles.

Dans notre travail, nous avons adopté le test standard de Leahy [LEA90] avec une transition cycloïdale [SER89][TAR89]. Dans ce test, les différentes articulations sont déplacer, respectivement, de la position initiale [-50 deg, -135 deg, 135 deg] à la position finale [45 deg, -85 deg, 30 deg] en un temps de mouvement égale à $t_f = 1.5 \text{ sec}$.

La rapidité d'exécution ainsi que la configuration initiale et finale permettent d'exciter toute la dynamique du robot PUMA560 [LEA90]. Tandis que la transition cycloïdale assure une continuité de la position, la vitesse et l'accélération et même de la dérivée de l'accélération ("Jerk"), ce qui reflète l'imposition de couples sans discontinuité.

La trajectoire cycloïdale est définie comme suivante :

$$q^i(t) = \begin{cases} q^{di} + (q^{df} - q^{di}) \cdot \left(\frac{t}{t_f} - \sin(2\pi t/t_f) \right) & \text{si } t \leq t_f \\ q^{df} & \text{si } t > t_f \end{cases} \quad (\text{A.12})$$

Annexe B :**Démonstration des propositions de la première approche de la commande décentralisée floue adaptative continue****Démo 1****Démonstration de la proposition 1 (CDFA)**

En appliquant l'équation de la loi de commande (III.9) dans l'équation du modèle dynamique de l'erreur (III.8), et on omit l'indice de l'articulation, on obtient :

$$m\ddot{e} = -k_v \dot{e} - k_p e + Z^T \tilde{A} \xi + \tilde{\varepsilon} \quad (\text{B.1})$$

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

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

posons $X = [e \ \dot{e}]^T$. Donc l'équation (B.1) devient :

$$\dot{X} = \begin{bmatrix} 0 & I \\ -\frac{k_p}{m} & -\frac{k_v}{m} \end{bmatrix} X + \frac{1}{m} \begin{bmatrix} 0 \\ Z^T \tilde{A} \xi + \tilde{\varepsilon} \end{bmatrix} \quad (\text{B.2})$$

considérons un modèle de référence stable de l'erreur sous la forme :

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

où, η et ω sont respectivement, le coefficient d'amortissement et la pulsation propre du modèle de référence. Sous forme matricielle on a :

$$\dot{X}_m = DX_m \quad (\text{B.3a})$$

avec,

$$X_m = [e_m \ \dot{e}_m]^T; D = \begin{bmatrix} 0 & I \\ -\omega^2 & -2\eta\omega \end{bmatrix} \quad (\text{B.3b})$$

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

$$PD + D^T P = -Q \quad (\text{B.4})$$

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

$$E_v = X_m - X \quad (\text{B.5})$$

Si les conditions initiales sont tels que : $e_m(0) = \dot{e}_m(0) = 0$, alors $\forall t \geq 0, X_m(t) = 0$, d'où :

$$E_v(t) = -X(t) \quad (\text{B.6})$$

D'après les équations (B.2), (B.3) et (B.5), on aura :

$$\dot{E}_v = DE_v + \begin{bmatrix} 0 & 0 \\ \frac{k_p}{m} - \omega^2 & \frac{k_v}{m} - 2\eta\omega \end{bmatrix} X - \frac{1}{m} \begin{bmatrix} 0 \\ Z^T \tilde{A} \xi + \tilde{\varepsilon} \end{bmatrix} \quad (\text{B.7})$$

avec, $m > 0$.

Considérons, d'autre part la fonction de Lyapunov donnée par :

$$V_t = \frac{1}{2} E_v^T P E_v + \frac{1}{2} Q_1 \left(\frac{k_p}{m} - \omega^2 - k_p^* \right)^2 + \frac{1}{2} Q_2 \left(\frac{k_v}{m} - 2\eta\omega - k_v^* \right)^2 + \frac{1}{2} \text{Tr}((\tilde{A} + A_0)^T \Gamma_0^{-1} (\tilde{A} + A_0)) + \frac{1}{2} \gamma_0^{-1} (\tilde{\varepsilon} + \varepsilon_0)^2 \quad (\text{B.8})$$

où,

- $\text{Tr}(\cdot)$ désigne la trace de la matrice (\cdot);

- $(\cdot)^T$ dénote le transpose de la matrice (\cdot) ;
- $Q_1, Q_2, \gamma_0 > 0$ des scalaires strictement positifs ;
- $\Gamma_0 > 0$ matrice définie positive ;
- k_p^*, k_v^*, A_0 et ε_0 des fonctions introduites pour obtenir des lois d'adaptation de type proportionnelle-intégrale qui seront déterminées ultérieurement.

On note que la fonction V_l est positive de forme quadratique. C'est une fonction proposée en se basant sur la fonction de Lyapunov utilisée par Seraji [SER89] où on a introduit un terme supplémentaire lié aux paramètres du modèle flou.

Si on considère que l'inertie (ou la masse) m varie lentement (hypothèse H2), on peut la considérer comme une constante. Dans ce cas, après dérivation de la fonction V_l , on obtient :

$$\begin{aligned} \dot{V}_l = & \frac{1}{2} [\dot{E}_v^T P E_v + E_v^T P \dot{E}_v] + Q_1 \left(\frac{k_p}{m} - \omega^2 - k_p^* \right) \left(\frac{k_p}{m} - k_p^* \right) \\ & + Q_2 \left(\frac{k_v}{m} - 2\eta\omega - k_v^* \right) \left(\frac{k_v}{m} - k_v^* \right) + \text{Tr}((\tilde{A}^T + A_0) \Gamma_0^{-1} (\tilde{A} + \dot{A}_0)) + \gamma_0^{-1} (\tilde{\varepsilon} + \varepsilon_0) (\tilde{\varepsilon} + \dot{\varepsilon}_0) \end{aligned} \quad (\text{B.9})$$

Si on considère les relations (B.4), (B.6) et (B.7), le premier terme de l'expression (B.9) devient :

$$\frac{1}{2} [\dot{E}_v^T P E_v + E_v^T P \dot{E}_v] = -\frac{1}{2} E_v^T Q E_v - r \left(\frac{k_p}{m} - \omega^2 \right) e - r \left(\frac{k_v}{m} - 2\eta\omega \right) \dot{e} + \frac{\gamma_0^{-1}}{m} \tilde{\varepsilon} r + \frac{1}{m} Z^T \tilde{A} \xi r \quad (\text{B.10})$$

avec, $r = p_2 e + p_3 \dot{e}$, où, p_2 et p_3 sont des éléments de la matrice P tels que : $P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix}$.

en remplaçant l'équation (B.10) dans (B.9) et après arrangement on obtient :

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

avec,

$$\begin{aligned} \dot{V}_{11} = & -\frac{1}{2} E_v^T Q E_v + \left(\frac{k_p}{m} - \omega^2 \right) \left(Q_1 \left(\frac{k_p}{m} - k_p^* \right) - r e \right) \\ & + \left(\frac{k_v}{m} - 2\eta\omega \right) \left(Q_2 \left(\frac{k_v}{m} - k_v^* \right) - r \dot{e} \right) - Q_1 \left(\frac{k_p}{m} - k_p^* \right) k_p^* - Q_2 \left(\frac{k_v}{m} - k_v^* \right) k_v^* \end{aligned} \quad (\text{B.11b})$$

$$\dot{V}_{12} = \tilde{\varepsilon} \left(\frac{1}{m} r + \gamma_0^{-1} (\tilde{\varepsilon} + \dot{\varepsilon}_0) \right) + \gamma_0^{-1} \varepsilon_0 (\tilde{\varepsilon} + \dot{\varepsilon}_0) \quad (\text{B.11c})$$

$$\dot{V}_{13} = \frac{1}{m} r Z^T \tilde{A} \xi + \text{Tr}(\tilde{A}^T \Gamma_0^{-1} (\tilde{A} + \dot{A}_0)) + \text{Tr}(A_0^T \Gamma_0^{-1} (\tilde{A} + \dot{A}_0)) \quad (\text{B.11d})$$

Si l'erreur de reconstruction ε varie lentement par rapport au signal auxiliaire $\hat{\varepsilon}$ (hypothèse H3), on a : $\tilde{\varepsilon} = \hat{\varepsilon} - \varepsilon = -\dot{\hat{\varepsilon}}$, donc :

$$\dot{V}_{12} = \tilde{\varepsilon} \left(\frac{1}{m} r - \gamma_0^{-1} (\dot{\hat{\varepsilon}} - \dot{\varepsilon}_0) \right) - \gamma_0^{-1} \varepsilon_0 (\dot{\hat{\varepsilon}} - \dot{\varepsilon}_0) \quad (\text{B.12})$$

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

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

pour avoir $\dot{V}_l \leq 0$, on choisit :

$$\begin{cases} Q_1 \left(\frac{\dot{k}_p}{m} - \dot{k}_p^* \right) = re \\ Q_2 \left(\frac{\dot{k}_v}{m} - \dot{k}_v^* \right) = r\dot{e} \\ k_p^* = Q_1^* Q_1 \left(\frac{\dot{k}_p}{m} - \dot{k}_p^* \right) = Q_1^* re & Q_1^* \geq 0. \\ k_v^* = Q_2^* Q_2 \left(\frac{\dot{k}_v}{m} - \dot{k}_v^* \right) = Q_2^* r\dot{e} & Q_2^* \geq 0. \end{cases} \quad (\text{B.14a})$$

$$\begin{cases} \gamma_0^{-1} (\dot{\varepsilon} - \dot{\varepsilon}_0) = \frac{r}{m} \\ \varepsilon_0 = \gamma_0^* \gamma_0^{-1} (\dot{\varepsilon} - \dot{\varepsilon}_0) = \gamma_0^* \frac{r}{m} & \gamma_0^* \geq 0 \\ \Gamma_0^{-1} (\dot{A} - \dot{A}_0) = \frac{r}{m} Z \xi^T \\ A_0 = \Gamma_0^* \Gamma_0^{-1} (\dot{A} - \dot{A}_0) = \Gamma_0^* \frac{r}{m} Z \xi^T & \Gamma_0^* \geq 0 \text{ (SDP)} \end{cases} \quad (\text{B.14b})$$

on obtient :

$$\begin{aligned} \dot{V}_{11} &= -\frac{1}{2} E_v^T Q E_v - Q_1^* (re)^2 - Q_2^* (r\dot{e})^2 \leq 0 \\ \dot{V}_{12} &= -\gamma_0^{-2} \gamma_0^* (\dot{\varepsilon} - \dot{\varepsilon}_0)^2 \leq 0 \quad \text{car } \gamma_0^* \geq 0 \\ \dot{V}_{13} &= -\text{Tr} \left((\Gamma_0^{-1} (\dot{A} - \dot{A}_0))^T \Gamma_0^{*T} (\Gamma_0^{-1} (\dot{A} - \dot{A}_0)) \right) \leq 0, \quad \text{car } \Gamma_0^* \geq 0 \end{aligned}$$

donc,

$$\begin{aligned} \dot{V}_1 &= -\frac{1}{2} E_v^T Q E_v - Q_1^* (re)^2 - Q_2^* (r\dot{e})^2 - \gamma_0^{-2} \gamma_0^* (\dot{\varepsilon} - \dot{\varepsilon}_0)^2 \\ &\quad - \text{Tr} \left((\Gamma_0^{-1} (\dot{A} - \dot{A}_0))^T \Gamma_0^{*T} (\Gamma_0^{-1} (\dot{A} - \dot{A}_0)) \right) \leq 0 \end{aligned} \quad (\text{B.15})$$

Donc, les relations (B.14a,b) assure la négativité de la fonction dérivée ($\dot{V}_1 \leq 0$).

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

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

$$L_\infty = \{s(t) : |s(t)| < \infty\} : \text{ensemble des fonctions bornées ;}$$

$$L_2 = \left\{ s(t) : \int_0^{+\infty} s(t)^2 dt < \infty \right\} : \text{ensembles des fonctions quadratiquement intégrables.}$$

D'après la relation (B.6), on a : $E = -[e \quad \dot{e}]^T$, et sachant que les trajectoires désirées $q^d, \dot{q}^d \in L_\infty$ (hypothèse H1), alors la bornitude de E implique que : $q, \dot{q} \in L_\infty$ (première conséquence de la proposition 1(R1)).

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

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

On considère une fonction réelle r qui vérifie les conditions suivantes :

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

- $r \in L_2$

alors cette fonction converge asymptotiquement vers zéro.

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

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

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

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

La première condition C1 résulte directement de la bornitude de $V_v(t)$. Pour vérifier la condition C2, on choisit la matrice Q de forme diagonale : $Q = \text{diag}\{2\rho_1, 2\rho_2\}$ avec, $\rho_1, \rho_2 > 0$. De l'équation (B.13), on peut déduire :

$$\dot{V}_1 \leq -\frac{1}{2}\rho_1 e^2 - \frac{1}{2}\rho_2 \dot{e}^2 \quad (B.16)$$

d'où,

$$\begin{cases} \dot{V}_1 \leq -\frac{1}{2}\rho_1 e^2 \\ \dot{V}_1 \leq -\frac{1}{2}\rho_2 \dot{e}^2 \end{cases} \quad \text{alors} \quad \begin{cases} \int_0^t e(\tau)^2 d\tau \leq \frac{2}{\rho_1}(V(0) - V(\infty)) \\ \int_0^t \dot{e}(\tau)^2 d\tau \leq \frac{2}{\rho_2}(V(0) - V(\infty)) \end{cases}$$

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

D'après l'équation (B.14a), on peut déduire :

$$k_p(t) = Q_1^{-1} m \int_0^t r e d\tau + Q_1^* m r e + k_p(0) \quad (B.17a)$$

$$k_v(t) = Q_2^{-1} m \int_0^t r \dot{e} d\tau + Q_2^* m r \dot{e} + k_v(0) \quad (B.17b)$$

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

$$\left| \int_0^t e \dot{e} d\tau \right| \leq \int_0^t e^2 d\tau + \int_0^t \dot{e}^2 d\tau \quad (B.18)$$

on déduit que : k_p et $k_v \in L_\infty$ (conséquence R2). D'autre part, on sait que $E(t), X(t), \tilde{A}$ et $\tilde{\varepsilon}$ sont bornées ($\in L_\infty$), alors, on conclut d'après l'équation (B.7) que $\ddot{e} \in L_\infty$. Donc la troisième condition (C3) est vérifiée. D'après le corollaire, les erreurs $e(t)$ et $\dot{e}(t)$ convergent asymptotiquement vers zéro (conséquence R3). D'après l'hypothèse H1, $\ddot{q}^d \in L_\infty$ donc la bornitude de \ddot{e} ($\ddot{e} \in L_\infty$) implique que $\ddot{q} \in L_\infty$ (conséquence R1). Pour montrer la bornitude des variables \hat{A} et $\hat{\varepsilon}$, on a :

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

d'où la conséquence R3.

Pour déterminer p_2 et p_3 on utilise la relation (B.4), on obtient :

$$p_2 = \frac{\rho_1}{\omega^2}, p_3 = \frac{1}{2\eta\omega} \left(\frac{\rho_1}{\omega} + \rho_2 \right) \quad (B.19)$$

Puisque η, ω des arbitraires positifs, il suffit de prendre $p_2, p_3 > 0$. Si de plus dans l'expression (B.14a,b) et (B.17a,b), on choisit :

$$\left\{ \begin{array}{l} Q_1^{-1} = \frac{\alpha_1}{m}, \quad Q_1^* = \frac{\alpha_2}{m} \quad \alpha_1 > 0, \alpha_2 \geq 0 \\ Q_2^{-1} = \frac{\beta_1}{m}, \quad Q_2^* = \frac{\beta_2}{m}, \quad \beta_1 > 0, \beta_2 \geq 0 \\ \gamma_0 = \gamma_1 \quad m, \quad \gamma_0^* = \gamma_2 \quad m \quad \gamma_1 > 0, \gamma_2 \geq 0 \\ \Gamma_0 = \Gamma_1 \quad m, \quad \Gamma_0^* = \Gamma_2 \quad m, \quad \Gamma_1 > 0, \Gamma_2 \geq 0. \end{array} \right. \quad (\text{B.20})$$

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

$$\left\{ \begin{array}{l} \dot{k}_p(t) = \alpha_1 \int_0^t r e \, d\tau + \alpha_2 r e + k_p(0) \\ \dot{k}_v(t) = \beta_1 \int_0^t r \dot{e} \, d\tau + \beta_2 r e + k_v(0) \\ \dot{\lambda}(t) = \Gamma_1 \int_0^t Z \xi^T r \, d\tau + \Gamma_2 Z \xi^T r + \hat{\lambda}(0) \\ \dot{\hat{e}}(t) = \gamma_1 \int_0^t r \, d\tau + \gamma_2 r + \hat{e}(0). \end{array} \right. \quad (\text{B.21})$$

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

Démo 2

Démonstration de la proposition 2 (CDFAR)

Pour démontrer les résultats énoncés dans la proposition 2, on suit les mêmes étapes utilisées dans la démonstration de la proposition 1. Les variables introduites dans la section précédente gardent leurs définitions sauf s'il est indiqué autrement.

Le modèle dynamique (III.16), s'écrit sous la forme matricielle suivante :

$$\dot{E}_v = DE_v + \begin{bmatrix} 0 & 0 \\ \frac{k_p}{m} - \omega^2 & \frac{k_v}{m} - 2\eta\omega \end{bmatrix} X - \frac{1}{m} \begin{bmatrix} 0 \\ Z^T \tilde{A} \xi + \tilde{\varepsilon} \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0 \\ m\ddot{q}^d - u_s(t) \end{bmatrix} \quad (\text{B.22})$$

Par rapport à l'équation (B.7), il apparaît le terme lié à l'accélération désirée.

Proposons la fonction de Lyapunov V_2 de la même forme que V_1 (donnée par (B.8)) :

$$\begin{aligned} V_2 = & \frac{1}{2} E_v^T P E_v + \frac{1}{2} Q_1 \left(\frac{k_p}{m} - \omega^2 - k_p^* \right)^2 + \frac{1}{2} Q_2 \left(\frac{k_v}{m} - 2\eta\omega - k_v^* \right)^2 \\ & + \frac{1}{2} \text{Tr}((\tilde{A} + A_0)^T \Gamma_0^{-1} (\tilde{A} + A_0)) + \frac{1}{2} \gamma_0^{-1} (\tilde{\varepsilon} + \varepsilon_0)^2 \end{aligned} \quad (\text{B.23})$$

Après dérivation, on obtient :

$$\begin{aligned} \dot{V}_2 = & \frac{1}{2} [\dot{E}_v^T P E_v + E_v^T P \dot{E}_v] + Q_1 \left(\frac{k_p}{m} - \omega^2 - k_p^* \right) \left(\frac{\dot{k}_p}{m} - \dot{k}_p^* \right) \\ & + Q_2 \left(\frac{k_v}{m} - 2\eta\omega - k_v^* \right) \left(\frac{\dot{k}_v}{m} - \dot{k}_v^* \right) + \text{Tr}((\tilde{A}^T + A_0) \Gamma_0^{-1} (\dot{\tilde{A}} + \dot{A}_0)) + \gamma_0^{-1} (\tilde{\varepsilon} + \varepsilon_0) (\dot{\tilde{\varepsilon}} + \dot{\varepsilon}_0) \end{aligned} \quad (\text{B.24})$$

En utilisant l'équation (B.22), le premier terme de (B.24) s'écrit :

$$\begin{aligned} \frac{1}{2} [\dot{E}_v^T P E_v + E_v^T P \dot{E}_v] = & -\frac{1}{2} E_v^T Q E_v - r \left(\frac{k_p}{m} - \omega^2 \right) e - r \left(\frac{k_v}{m} - 2\eta\omega \right) \dot{e} \\ & + \frac{1}{m} \tilde{\varepsilon}^T r + Z^T \tilde{A} \xi \quad r + \frac{1}{m} (m\ddot{q}^d - u_s(t)) r \end{aligned} \quad (\text{B.25})$$

donc \dot{V}_2 s'écrit :

$$\dot{V}_2 = \dot{V}_1 + \dot{V}_{21} \quad (\text{B.26})$$

où,

- \dot{V}_1 est donnée par l'expressions (B.11a,b,c,d) ;

$$- \dot{V}_{21} = \frac{1}{m} (m\ddot{q}^d - u_s(t)) r$$

On a montré dans *Démo 1* que les lois d'adaptation (B.21) permettent d'avoir :

$$\begin{aligned} \dot{V}_1 = & -\frac{1}{2} E_v^T Q E_v - Q_1^*(re)^2 - Q_2^*(r\dot{e})^2 - \gamma_0^{-2} \gamma_0^* (\hat{\varepsilon} - \hat{\varepsilon}_0)^2 \\ & - \text{Tr} \left((\Gamma_0^{-1} (\hat{A} - \dot{A}_0))^T \Gamma_0^{*T} (\Gamma_0^{-1} (\hat{A} - \dot{A}_0)) \right) \leq 0 \end{aligned} \quad (\text{B.27})$$

Si en plus, on choisit :

$$u_s(t) = k_{s1} |\ddot{q}^d| \text{sign}(r) \quad \frac{k_{s1}}{m} \geq 1 \quad (\text{B.28})$$

On obtient :

$$\dot{V}_{21} = \ddot{q}^d r - \frac{k_{s1}}{m} |\ddot{q}^d| |r| \leq 0 \quad (\text{B.29})$$

D'après (B.26), (B.27) et (B.29), on déduit :

$$\dot{V}_2 \leq -\frac{1}{2} E_v^T Q E_v - Q_1^*(re)^2 - Q_2^*(r\dot{e})^2 \quad (\text{B.30})$$

donc, $\dot{V}_2 \leq 0$.

Puisque la fonction de Lyapunov V_2 est positive et \dot{V}_2 est négative, on conclut que la fonction V_2 et les variables $E(t)$, $\tilde{\varepsilon}(t)$ et $\tilde{A}(t)$ sont bornées.

En suivant les mêmes étapes effectués dans *Démo 1*, on montre que les conditions du corollaire sont vérifiées. D'où on peut déduire que les conséquences énoncés dans la proposition 2 sont valides si on choisit les lois d'adaptation (B.22) et $u_s(t)$ selon la relation (B.28).

Démo 3

Démonstration de la proposition 3 (CDFAM)

Pour démontrer la proposition 3, nous suivons les mêmes étapes utilisées dans la démonstration de la proposition 1 (*Démo 1*).

En remplaçant (III.18b) dans (III.18a), le modèle dynamique de l'erreur en boucle fermée donnée par (III.14) s'écrit sous la forme matricielle suivante :

$$\dot{E}_v = D E_v + \begin{bmatrix} 0 & 0 \\ \frac{k_p}{m} - \omega^2 & \frac{k_v}{m} - 2\eta\omega \end{bmatrix} X + \frac{1}{m} \begin{bmatrix} 0 \\ Z^T \tilde{A} \xi + \tilde{\varepsilon} \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0 \\ \tilde{k}_m \ddot{q}^d \end{bmatrix} \quad (\text{B.31})$$

où, $\tilde{k}_m = m - k_m$.

D'autre part, considérons une fonction de Lyapunov V_3 donnée par :

$$V_3 = \frac{1}{2} E_v^T P E_v + \frac{1}{2} Q_1 \left(\frac{k_p}{m} - \omega^2 - k_p^* \right)^2 + \frac{1}{2} Q_2 \left(\frac{k_v}{m} - 2\eta\omega - k_v^* \right)^2 + \frac{1}{2} \text{Tr}((\tilde{A} + A_0)^T \Gamma_0^{-1} (\tilde{A} + A_0)) + \frac{1}{2} \gamma_0^{-1} (\tilde{\varepsilon} + \varepsilon_0)^2 + \frac{1}{2} \gamma_{m0}^{-1} (\tilde{k}_m + k_m^*)^2 \quad (\text{B.32})$$

avec, $\gamma_{m0}^{-1} > 0$

k_m^* : Une fonction introduite pour obtenir une loi d'adaptation de type proportionnelle-intégrale.

De la même façon que dans la section *Démo 1*, on obtient après dérivation :

$$\dot{V}_3 = \frac{1}{2} [\dot{E}_v^T P E_v + E_v^T P \dot{E}_v] + Q_1 \left(\frac{k_p}{m} - \omega^2 - k_p^* \right) \left(\frac{\dot{k}_p}{m} - \dot{k}_p^* \right) + Q_2 \left(\frac{k_v}{m} - 2\eta\omega - k_v^* \right) \left(\frac{\dot{k}_v}{m} - \dot{k}_v^* \right) + \text{Tr}((\tilde{A}^T + A_0) \Gamma_0^{-1} (\tilde{A} + A_0)) + \gamma_0^{-1} (\tilde{\varepsilon} + \varepsilon_0) (\dot{\tilde{\varepsilon}} + \dot{\varepsilon}_0) + \gamma_{m0}^{-1} (\tilde{k}_m + k_m^*) (\dot{\tilde{k}_m} + \dot{k}_m^*) \quad (\text{B.33})$$

En utilisant l'équation (B.31), le premier terme de \dot{V}_3 s'écrit :

$$\frac{1}{2} [\dot{E}_v^T P E_v + E_v^T P \dot{E}_v] = -\frac{1}{2} E_v^T Q E_v - r \left(\frac{k_p}{m} - \omega^2 \right) \dot{e} - r \left(\frac{k_v}{m} - 2\eta\omega \right) \dot{e} + \frac{1}{m} \tilde{\varepsilon}^T r + \frac{1}{m} Z^T \tilde{A} \xi^T r + \frac{1}{m} \tilde{k}_m \ddot{q}^d \quad (\text{B.34})$$

donc \dot{V}_3 devient :

$$\dot{V}_3 = \dot{V}_1 + \dot{V}_{31} \quad (\text{B.35})$$

où, \dot{V}_1 est donnée par l'expression (B.11a, b, c et d), et :

$$\dot{V}_{31} = \frac{1}{m} \tilde{k}_m (\ddot{q}^d r - m \gamma_{m0}^{-1} (k_m - k_m^*)) - \gamma_{m0}^{-1} k_m^* (k_m - k_m^*) \quad (\text{B.36})$$

car $\dot{\tilde{k}_m} = -\dot{k}_m$ (hypothèse H2).

On a montré dans *Démo 1* que les lois d'adaptation (B.21) permettent d'avoir :

$$\dot{V}_1 \leq -\frac{1}{2} E_v^T Q E_v - Q_1^* (re)^2 - Q_2^* (r\dot{e})^2 \quad (\text{B.37})$$

Si de plus, on choisit :

$$\begin{cases} m \gamma_{m0}^{-1} (k_m - k_m^*) = \ddot{q}^d r \\ k_m^* = \gamma_m^* \gamma_{m0}^{-1} (k_m - k_m^*) = \frac{1}{m} \gamma_m^* \ddot{q}^d r \end{cases} \quad (\text{B.38})$$

On aura :

$$\dot{V}_{31} = -\gamma_m^* \left(\frac{\ddot{q}^d r}{m} \right)^2 \leq 0 \quad \text{car } \gamma_m^* \geq 0 \quad (\text{B.39})$$

D'après (B.35), (B.37) et (B.39), on déduit que :

$$\dot{V}_3 \leq -\frac{1}{2} E_v^T Q E_v - Q_1^* (re)^2 - Q_2^* (r\dot{e})^2 \leq 0 \quad (\text{B.40})$$

Puisque la fonction de Lyapunov V_3 est positive et \dot{V}_3 est négative, alors la fonction $V_3(t)$ et les variables $E(t), \tilde{E}(t), \tilde{A}(t)$ et $\tilde{k}_m(t)$ sont bornées. $\tilde{k}_m(t)$ est bornée implique que $k_m(t)$ est aussi bornée.

De la même manière que dans *Demo 1*, on montre aussi que les conditions du corollaire sont vérifiées. D'où on peut énoncer que les conséquences de la proposition 2 sont valides, sous les hypothèses H1, H2 et H3, si on choisit les lois d'adaptation (B.21). L'expression du gain $k_m(t)$ peut être déduite à partir des relations (B.38). Pour cela, on pose :

$$\begin{cases} \gamma_{m0} = m\gamma_{m1} & \gamma_{m1} > 0 \\ \gamma_m^* = m\gamma_{m2} & \gamma_{m2} \geq 0 \end{cases} \quad (\text{B.41})$$

d'où on trouve :

$$k_m(t) = \gamma_{m1} \int_0^t \ddot{q}^d r d\tau + \gamma_{m2} \dot{q}^d r + k_m(0) \quad (\text{B.42})$$

Annexe C :*Cas particulier de la classe \mathfrak{I}*

Considérons un ensemble d'équations différentielles donnée sous la forme matricielle suivante :

$$\dot{P}^{-1} = -\lambda P^{-1} + \lambda W_p \quad (\text{C.1})$$

avec, λ une constante positive ou nulle ($\lambda \in \mathbb{R}_+$)

W_p une matrice d'éléments constants positifs ($W_p \in \mathbb{R}_+^{l \times l}$)

P^{-1} une matrice inverse de P d'éléments variables ($P \in \mathbb{R}_+^{l \times l}$)

L'équation (C.1) est l'écriture matricielle de $l \times l$ équations différentielles du premier ordre à coefficients constants identiques λ , et à deuxièmes membres constants. Donc la solution est donnée par :

$$P^{-1}(t) = P^{-1}(0) \exp(-\lambda t) + \lambda \int_0^t \exp(-\lambda(t-\tau)) W_p d\tau \quad (\text{C.2})$$

avec, $P^{-1}(0) = P^{-1}(t=0)$

d'où :

$$P^{-1}(t) = (P^{-1}(0) - W_p) \exp(-\lambda t) + W_p \quad (\text{C.3})$$

La première et la deuxième dérivées de P^{-1} s'écrivent :

$$\dot{P}^{-1}(t) = -\lambda (P^{-1}(0) - W_p) \exp(-\lambda t) \quad (\text{C.4})$$

$$\ddot{P}^{-1}(t) = \lambda^2 (P^{-1}(0) - W_p) \exp(-\lambda t) \quad (\text{C.5})$$

donc, pour que la matrice $P^{-1}(t)$, $P^{-1}(t)$ soient respectivement, matrice définie positive (propriété D1 de la classe \mathfrak{I}), matrice semi-définie négative (propriété D2), il suffit de choisir :

$$\begin{cases} (P^{-1}(0) - W_p), W_p & \text{matrices définies positive si } \lambda \neq 0 \\ P^{-1}(0) & \text{matrice définie positive si } \lambda = 0 \end{cases} \quad (\text{C.6})$$

En examinant les expressions (C.4) et (C.5), on constate que les matrices $\dot{P}^{-1}(t)$ et $\ddot{P}^{-1}(t)$ sont bornées d'où la propriété D3 est vérifiée.

On conclut que la matrice P appartient à la classe \mathfrak{I} . Pour déterminer l'équation différentielle en P , on utilise la relation :

$$P \dot{P}^{-1} = -\dot{P} P^{-1} \quad (\text{C.7})$$

sachant que P^{-1} est donnée par (C.1), on déduit que :

$$\begin{cases} \dot{P}(t) = \lambda P - \lambda P W_p P \\ P(t=0) = (P^{-1}(0))^{-1} \end{cases} \quad (\text{C.8})$$

Il faut noter, que la valeur finale de la matrice $P(t)$ est :

$$\lim_{t \rightarrow \infty} P(t) = W_p^{-1} \quad (\text{C.9})$$

Bibliographie

Bibliographie

- [ARM86] B. Armstrong, O. Khatib and J. Burdick, "*The explicit dynamic model and inertial parameters of the PUMA560 arm*", In Proceedings Of IEE, pp. 510-519, 1986.
- [BER93] H. Berghuis and H. Nijmeijer, "*Global regulation of robots using only position measurements*", Systems and Control Letters, Vol. 21, pp. 289-293, 1993.
- [BUH86] H. Bühler, "*Réglage par mode de glissement*", Presse polytechnique et universitaire Romande, Losane (Suisse), 1986.
- [BUH94] H. Bühler, "*Réglage par logique floue*", Presse polytechnique et universitaire Romande, Losane (Suisse), 1994.
- [CAN92] C. Canudas de Wit and N. Fixot, "*Adaptive control of robot manipulators via velocity estimated feedback*", IEEE Transactions on Automatic Control, Vol. 37, no. 8, pp. 1234-1237, 1992.
- [CHA96] W.-D. Chang, L.-C. Fu and J.-H. yang, "*Adaptive robust neural-network based control for SISO systems*", 13th. Triennial world Congress, San-Fransisco, USA, pp. 163-168, 1996.
- [CHO88] Y.K. Choi and Z. Bien, "*Decentralized control scheme for control of a multi-arm-type robot*", International Journal of Control, Vol. 48, no. 4, pp. 1715-1722, 1988.
- [COL95] R. Colbaugh and K. Glass, "*Adaptive tracking control of manipulators using only position measurements*", Proceedings of the 34th Conference On Decision And Control, New-Orleans, pp. 51-54, LA-Dec. 1995.
- [COM96] S. Commuri and F. L. Lewis, "*Design and stability analysis of adaptive-fuzzy controllers for a class of nonlinear systems*", IEEE proceedings of the 35th Conference on Decision and Control, Kobe (Japan), Dec. 1996.
- [CRA86] J. J. Craig, P. Hsu and S. Sastry, "*Adaptive control of mechanical manipulators*", International Conference on Robotics Automation, San-Fransisco, CA, 1986.
- [DAZ81] J. D'azzo and C. H. Houpis, "*Linear control ssystem: analysis and design*", Mc Graw-Hill (ed.), 1981.
- [DRI93] D. Driankov, H. Hellendoor and M. Reinfrank, "*An introduction to fuzzy control*", Springer-Verlag, Berlin Heidelberg, 1993
- [DUB79] S. Dubowsky and D. T.Desforges, "*The application of model referenced adaptive control to robotic manipulators*", ASME Journal of Dynamic systems, Measurements and Control, Vol. 101, no. 3, pp. 193-200, 1979.
- [FU87] K. S. Fu, R. C. Gonzalez and C. S. G. Lee, "*Robotics: control, sensing, vision and intelligence*", Edition Mc Graw-Hill, 1997.
- [FU92] L.-C. Fu, "*Robot adaptive decentralized control of robot manipulators*", IEE Transaction On Automatic and Control, Vol. 37, pp. 106-110, Jan. 1992.

- [GEO92] G. George, *"Sur des méthodes de commande non linéaires adaptatives : Aspects échantillonné et applications"*, Doctorat es science, Université de Paris XI Orsay, France, 1992.
- [GUZ86] C. Guzzella and H. P. Geering, *"Model following variable structure control for a class of uncertain mechanical systems"*, IEEE Conference on Decision and Control, Athens, Greece, 1986.
- [HA87] I. J. Ha and E. G. Gilbert, *"Robust tracking in nonlinear systems"*, IEEE Transaction on Automatic and Control, Vol. AC-32, pp. 763-771, Sep. 1987.
- [HOR82] R. Horwitz and M. Tomizuka, *"Discrete time model reference adaptive control of mechanical manipulator"*, Computer in Engineering Robot and Robotics, ASME, 2, pp. 107-112, 1982.
- [HOR92] S.-I. Horikawa, T. Furuhashi and Y. Uchikawa, *"On fuzzy modelling using guzzy neural networks with the back-propagation algorithm"*, IEEE Transactions on Neural Networks, Vol. 3, no. 5, pp. 801-806, Sept. 1992.
- [JAN92] J.-S. R. Jang, *"Self learning fuzzy controllers based on temporal back propagation"*, IEEE Transactions on Neural Networks, Vol. 3, no. 5, pp. 714-723, Sept. 1992.
- [JAN95] J.-S. R. Jang, *"Neuro fuzzy modelling and control"*, Proceedings of the IEEE, Vol. 83, no.3, pp. 378-404, Mar. 1995.
- [KET92] R. Ketata, *"Méthodologies des régulations numériques incluant la logique floue"*, Thèse de doctorat de l'institut des sciences appliquées de Toulouse, L.A.A.S, France, 1992.
- [KIM94] K.-C. Kim and J.-H. Kim, *"Multicriteria fuzzy control"*, Journal of Intelligent and Fuzzy Systems, Vol. 2, pp. 279-288, 1994.
- [KRS95] M. Krstic, I. Kanellakopoulos and P. Kokotovic, *"Nonlinear and adaptive control design"*, John Wiley & soons, Inc., 1995.
- [KWA97] C. Kwan and all., *"Robust robot control using fuzzy CMAC"*, Proceedings of IFAC/IFIP/IMACS Conference, Belfort, France, 1997.
- [LAN79] Y. D. Landau, *"Adaptive control : Model reference approach"*, Marcel Dekker, New York, 1979.
- [LEA90] M. B. Leahy, *"Model based auxiliary input control : developement and experimental analysis"*, Proceedings Of the 20th Conference On Decision and Control, Honolulu, Hawai, Dec. 1990.
- [LEW93] F. L. Lewis, A. Yesildirek and k. Liu, *"Neural net robot controller with guaranteed stability"*, IEEE Transaction on Robotics and Automation, Vol. 5, no. 2, pp. 103-108, 1993.
- [LOU97] M. Loudini, *"Modélisation, analyse et méthodologies de commande linguistique floue d'un bras manipulateur de robot flexible"*, Mémoire de magister, Ecole nationale polytechnique (ENP), Alger, 1997.
- [MAD97] T. Madani et S. Amarati, *"Commande décentralisée à structure variable: application en robotique "*, Mémoire de PFE de l'école nationale polytechnique (ENP), Alger, 1997.

- [MEN95] T. M. Mendel, *"Fuzzy logic systems for engineering : A tutorial"*, Proceedings of the IEEE, Vol. 83, no. 3, pp. 345-377, Mar. 1989.
- [NEU85] C.-P. Neuman and V. D. Tourassis, *"Discrete dynamic robot models"*, IEEE Transaction on Systems, Man and Cybernetics, Vol. SMC-15, no. 2, Mar./Apr. 1985.
- [NIC90] S. Nicosia and P. Tomei, *"Robust control by using only joint positions measurements"*, IEEE Transactions on Automatic Control, Vol. 37, no. 8, pp. 1234-1237.
- [PAU86] R. P. Paul, *"Robots manipulators : mathematics, programming and control"*, The Mit press, 1986.
- [PIC97] J. Pico and P. Moya, *"Localized BF-type networks for identification and adaptive control of discrete-time nonlinear systems"*, 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications-SICICA'97, Annecy, France, pp. 653-658, Jun. 9-11, 1997.
- [PRO79] J. J. Procyk and E. H. Mamdani, *"A linguistic self-organizing process controller"*, Automatica, Vol. 15, pp. 15-30, 1979.
- [SCH85] E. M. Scharf and N. J. Mandic, *"The applicatio, of a fuzzy controller to the control of a multi degree-of-freedom robot arm"*, Industrial Applications of fuzzy control, M. Sugeno (ed.), Elsevier, Science publishers B.V. (North-Holland), 1985.
- [SER87] H. Seraji, *"A new approach to adaptive control of manipulators"*, ASME Journal of Dynamic Systems, Measurement and Control, Vol. 109, no. 3, pp. 193-202, Nov., 1987.
- [SER89] H. Seraji, *"Decentralized adaptive control of manipulators : theory, simulation and experimentation"*, Transaction on robotics and Automation, Vol. 5, no. 2, Apr. 1989.
- [SIN85] S. N. Singh, *"Adaptive model following control of nonlinear robotic systems"*, IEEE Transactions on Automatic Control, 30, 1099-1100, 1985.
- [SLO83] J.-J. E. Slotine and S. Sastry, *"Tracking control of nonlinear systems using sliding surfaces with application to robot manipulators"*, International Journal of Control, Vol. 38, pp. 465-492, 1983.
- [SLO88] J.-J. E. Slotine, *"Adaptive manipulator control : A case study"*, IEEE Transaction on Automatic Control, Vol. 33, no. 11, pp. 995-1002, 1987.
- [SLO91] J. J.-E. Slotine and W. Li, *"Applied nonlinear control"*, Prentice Hall, 1991.
- [SPO95a] J. T. Spooner and K. M. Passino, *"Stable indirect adaptive control using fuzzy systems and neural networks"*, IEEE Proceedings of the 34th Conference On Decision and Control, New-Orleans, pp. 243-248, LA-Dec. 1995.
- [SPO95b] J. T. Spooner and K. M. Passino, *"Stable direct adaptive control using fuzzy systems and neural networks"*, IEEE Proceedings of the 34th Conference On Decision and Control, New-Orleans, pp. 249-254, LA-Dec. 1995.
- [SPO95c] J. T. Spooner and K. M. Passino, *"Adaptive control of a class of decentralized nonlinear systems"*, Proceedings of the 34th Conference On Decision and Control, New-Orleans, pp. 1135-1139, LA-Dec. 1995.

- [SUG85] M. Sugeno and T. Takagi, "*Derivation of fuzzy control rules from human operator's control actions*", IFAC Fuzzy Information, France, 1985.
- [TAN88] R. Tansheit and E. M. Sharf, "*Experiments with the use of rule-based self-organising controller for robotics application*", Fuzzy Sets and Systems 26, pp. 195-214, North-Holland, 1988.
- [TAR89] M. Tarokh, "*A discrete-time adaptive control scheme for robot manipulators*", Journal Of Robotic Systems, 7(2), pp. 145-166, 1990.
- [TRE97] A. Trebi-Ollennu and B. A. White, "*Robust output tracking for MIMO nonlinear systems : An adaptive fuzzy systems approach*", IEE Proceedings of Control Theory Application, Vol. 144, no. 6, pp. 537-544, Nov. 1997.
- [TZA90] S. Tzafestas, N.P. Papanikolopoulos, "*Incremental fuzzy expert PID control*", IEEE transactions On Industrial Electronics, Vol. 37, no. 5, Oct. 1990.
- [WAT97] M. Watheq El Kharachi and M. Sheirah, "*Fuzzy logic controllers : A comparative study*", Proceedings IFAC/IFIP/IMACS Conference, pp. 209-215, Belfort, France, May 1997.
- [YUA91] J. Yuan and Y. Stepanenko, "*Robust control of robotic manipulators without velocity feedback*", International Journal of Robust and Nonlinear Control, Vol. 1, pp. 203-213, 1991.
- [ZAD65] L. Zadeh, "*Fuzzy sets*", Information and Control, Vol. 8, 1965.
- [ZHA93] Z.-Y. Zhao, M. Tomizuka and J. Isaka, "*Fuzzy gain scheduling of PID controllers*", IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, no. 5, Sep./Oct. 1993.

الملخص:

الهدف من هذا العمل هو استعراض كيفية استخدام الأنظمة الغامضة في حل مشاكل التحكم الآلي اللامركزي لأذرع آلية. الغرض من هذا التحكم هو جعل الأذرع الآلية تتبع مسارات مرجعية بحيث أن القوى أو العزوم المطبقة على كل ساعد من سواعد الذراع الآلي تحتكم فقط إلى القياسات المحلية من موضع و سرعة و تسارع. في البداية قمنا بتقديم لمحة عن المفاهيم الأساسية للأنظمة الغامضة و المتعلقة بالتحكم الآلي، ثم عرجنا على طريق التحكم اللامركزي بواسطة الأنظمة الغامضة. قمنا بتقديم طريقة التحكم الغامض المستمدة من التحكم ذي الأنماط المنزلة، ثم اقترحنا ضرباً من التحكم اللامركزي التلازمي المستمر، كما اقترحنا أيضاً طرق تحكم لامركزي تلازمي عددي. جميع هذه التقنيات طبقت على ذراع آلي من نوع PUMA560.

نتائج المحاكاة العددية سمحت بتقييم مدى نجاعة كل طريقة.

كلمات مفاتيح: أنظمة غامضة، تحكم لامركزي، ذراع آلي، تحكم تلازمي، إتباع مسار.

Abstract:

The purpose of this work is to present and synthesize decentralized control approaches by using fuzzy systems applied to robot trajectory tracking control problem.

First, main notions of fuzzy systems related to control problem are presented. Then we study and design decentralized fuzzy control schemes. Thus a fuzzy control based on sliding mode technique, continuous and discrete decentralized fuzzy adaptive control schemes are developed. All of the developed technics are applied to trajectory tracking control problem of three first links of the PUMA560.

Simulation results are used to evaluate the performance and the efficiency of the proposed technics.

Key words: Fuzzy systems, Decentralized control, Robot manipulator, Adaptive control, Trajectory tracking.

Résumé:

Le travail présenté dans ce mémoire concerne des techniques de commande décentralisée, en utilisant des systèmes flous appliquées à la commande en poursuite de trajectoires de référence des robots manipulateurs.

En premier lieu, nous avons abordé des notions fondamentales des systèmes flous liées aux problèmes de commande. par la suite, nous avons étudié et proposé des structures de commande décentralisée floue. Ainsi, nous avons présenté la commande floue à base de réglage à mode glissant. La commande décentralisée floue adaptative continue et la commande décentralisée floue adaptative discrète. Ces techniques sont appliquées à la commande en poursuite de trajectoires de référence des trois premières liaisons d'un bras de robot manipulateur de type PUMA560.

Les résultats de simulation obtenus ont permis l'évaluation des performances de ces techniques.

Mots clés: Système flou, Commande décentralisée, Robot manipulateur, Commande adaptative, Poursuite de trajectoire.