

Ministère de l'enseignement Supérieur et de la Recherche Scientifique

# Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

DER- Génie Electrique & Informatique

Département d'Automatique

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

THESE de Magister en Automatique

## THEME

Etude et Développement de Méthodes de Commande  
Adaptative Centralisée et Décentralisée en Utilisant  
Différents Types de Réseaux de Neurones Artificiels :  
Application au Robot PUMA 560

Présentée Par: **Omar BOUHALI**

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

*Soutenue le ....octobre 1998 devant le Jury composé de :*

Président	N. LOUAM	Maître de Conférence à L'ENP
Rapporteur	D. BOUKHETALA	Chargé de Cours à L'ENP
Rapporteur	F. BOUDJEMA	Maître de Conférence à L'ENP
Examineur	M. S. BOUCHERIT	Maître de Conférence à L'ENP
Examineur	H. CHEKIREB	Chargé de Cours à L'ENP
Examineur	M. TADJINE	Docteur à L'ENP

Octobre -1998

## Avant Propos

Ce travail a été effectué au sein du Laboratoire de Commande des Processus (LCP) de la DER de Génie Electrique et Informatique de l'Ecole Nationale Polytechnique (ENP) sous la direction conjointe de Monsieur D. Boukbeta, chargé de cours à l'ENP et Monsieur F. Boudjema, maître de conférence à l'ENP, qu'ils trouvent ici l'expression de toute ma gratitude pour les conseils avisés et l'encouragement constant qu'ils n'ont cessé de prodiguer tout au long de cette étude.

Mes remerciements les plus sincères sont adressés à Monsieur N. Louam, maître de conférence à l'ENP, pour l'honneur qu'il m'a fait en acceptant la présidence du jury de cette thèse.

Mes vifs remerciements vont aussi à Monsieur M. S. Boucherit, maître de conférence à l'ENP, je suis très honoré de sa présence dans le jury de cette thèse.

J'exprime ma gratitude à Monsieur H. Chekireb, chargé de cours à l'ENP, pour l'honneur qu'il m'a fait en acceptant de faire partie de ce jury.

Je remercie également Monsieur M. Tadjine, Docteur à l'ENP, pour l'intérêt qu'il a porté à ce travail en me faisant l'honneur de participer à ce jury.

Que mes amis S. Labiod, N. S. Amirouche, H. Derrouiche, C. Boubakir, E. Boubassoun, K. Laidi, Z. Tidjani et S. Boumaza trouvent ici toute ma gratitude pour leur compréhension, leur aide, leur ambiance bien sympathique et leur soutien moral et matériel qui n'ont cessé de m'apporter tout au long de l'élaboration de ce travail.

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

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

يهدف هذا العمل إلى إستعراض كيفية إستعمال الخلايا العصبونية الإصطناعية كوسيلة لحل بعض المشاكل المتعلقة بالتحكم الآلي المركزي و اللامركزي التلاؤمي في الأذرع الآلية. هدف هذا التحكم هو جعل هذه الأخيرة تتبع مسارات مرجعية بدقة. في البداية، قدمنا لمحة عن بعض المفاهيم الأساسية و التطورات الحديثة للخلايا العصبونية الإصطناعية. في ما بعد، قدمنا طرقا للتحكم الآلي المركزي التلاؤمي المباشر و غير المباشر بالخلايا العصبونية، ثم قدمنا إمتداد هذه الطرق إلى التحكم اللامركزي. كل تقنيات التحكم المذكورة طبقت على ذراع آلي من نوع PUMA 560. نتائج المحاكاة العديدة المتحصل عليها سمحت بتقييم مدى فعالة كل طرق التحكم المقترحة.

**كلمات مفتاحية :** الخلايا العصبونية الإصطناعية, تحكم تلاؤمي, تحكم لا مركزي, أذرع آلية.

### Abstract :

This work deals with structures and learning methods for Artificial Neural Networks (ANN) in centralized and decentralized adaptive control for robot manipulators. Fundamentals and advanced developments in neural network for modelling and control are firstly studied. These ANN are used in a centralized direct and indirect adaptive control schemes for robotic manipulators. Extension of these methods to the decentralised case is given. The developed algorithms are applied to the PUMA 560 robot manipulator and simulation results are given to highlight the performances and the robustness of these proposed schemes.

**Key words :** neural networks, adaptive control, centralized control, decentralized control, robotic manipulators.

### Résumé:

L'objectif de ce travail est d'étudier des stratégies de commande adaptatives par différents types de réseaux de neurones artificiels appliqués aux problèmes de poursuite de trajectoire des robots manipulateurs. En premier lieu, nous avons présenté les notions fondamentales et les développements récents des réseaux de neurones utilisés dans la modélisation et la commande des processus. Ensuite, nous avons étudié et proposé des structures de commande neuronale adaptative, à savoir la commande adaptative centralisée directe et indirecte par différents types de Réseaux de Neurones Artificiels (RNA). Une extension de tous ces schémas au cas décentralisé a été également donnée. Les techniques de commande sont ensuite appliquées à la commande en poursuite de trajectoire de référence d'un robot manipulateur de type PUMA560. Les résultats de simulation obtenus ont permis l'évaluation des performances de chaque technique.

## Sommaire

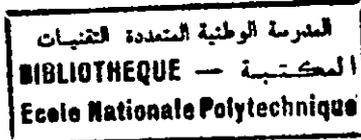
<i>Introduction Générale</i> .....	1
------------------------------------	---

### *Chapitre I Commande Adaptative Centralisée Directe Par RNA*

I.1	Introduction.....	3
I.2	Commande Adaptative Centralisée Directe par RNA.....	4
I.3	Commande Prédictive Adaptative Directe par RNA .....	5
I.4	Commande Adaptative Directe par RBFQ.....	21
I.4.1	RBFQ à Centres Fixes.....	21
I.4.1.a)	Cas SISO : Appliquée à un Pendule .....	21
I.4.1.b)	Cas MIMO .....	29
I.4.2	RBFQ à Centres Adaptatifs.....	33
I.5	Commande Prédictive Adaptative Directe par RBFQ .....	37
I.5.1	RBFQ à Centres Fixes.....	37
I.5.2	RBFQ à Centres Adaptatifs.....	42
I.6	Commande Adaptative Directe par RNA Récurrents .....	45
I.7	Commande Prédictive Adaptative Directe par RNA Récurrents.....	49
I.8	Conclusion.....	52

### *Chapitre II Commande Adaptative Décentralisée Directe Par RNA*

II.1	Introduction .....	53
II.2	Commande Adaptative Décentralisée Directe par RNA Statique.....	53
II.3	Commande Prédictive Adaptative Décentralisée Directe par RNA Statique.....	64
II.4	Commande Adaptative Décentralisée Directe par RBFQ.....	68
II.4.1	RBFQ à Centres Fixes .....	68
II.4.2	RBFQ à Centres Adaptatifs.....	72
II.5	Commande Prédictive Adaptative Décentralisée Directe par RBFQ .....	75
II.5.1	RBFQ à Centres Fixes .....	75
II.5.2	RBFQ à Centres Adaptatifs.....	78
II.6	Commande Adaptative Décentralisée Directe par RNA Récurrents.....	81
II.7	Commande Prédictive Adaptative Décentralisée Directe par RNA Récurrents .....	81
II.8	Conclusion.....	86



### *Chapitre III Commande Adaptative Décentralisée par RNA Basée sur la Méthode du Couple Calculé*

III.1	Introduction.....	87
III-2	Méthode du Couple Calculé.....	87
III.3	Commande Neuronale Adaptative.....	89
III.4	Commande Adaptative Décentralisée par RNA Statique.....	89
III.5	Commande Adaptative Décentralisée par RBFG.....	95
	II.5.1 RBFG à Centres Fixes.....	95
	II.5.2 RBFG à Centres Adaptatifs.....	98
III.6	Commande Adaptative Décentralisée par RNA Récurrents.....	101
III.7	Conclusion.....	104

### *Chapitre IV Commande Adaptative Indirecte Décentralisée Par RNA*

IV.1	Introduction.....	105
IV.2	Commande Adaptative Indirecte Décentralisée par RNA Statique.....	105
IV.3	Commande Adaptative Indirecte Décentralisée par RBFG.....	112
	IV.3.1 RBFG à Centres Fixes.....	112
	IV.3.2 RBFG à Centres Adaptatifs.....	115
IV.4	Commande Adaptative Indirecte Décentralisée par RNA Récurrents.....	118
IV.7	Conclusion.....	121

<i>Conclusion Générale</i> .....	122
----------------------------------	-----

<i>Annexe</i> .....	125
---------------------	-----

<i>Références Bibliographiques</i> .....	128
------------------------------------------	-----

particulier de dynamique, la stocker pour l'utiliser lorsqu'une situation semblable se produira. Cette démarche nécessite un contrôleur intelligent, qui mémorise, s'adapte, mais aussi capable de choisir la loi de commande appropriée. La commande adaptative non linéaire est un champ très important pour l'application des RNA. Les problèmes de commandes non linéaires sont souvent rencontrés dans différents domaines tels que l'hydraulique, *robotique*,...etc. les RNA peuvent être utilisés pour la prédiction ou la modélisation entrée/sorties pour un système ou pour corriger ses sorties.

La plus part des algorithmes de commande sont développés dans un contexte centralisé, dont l'information est traitée par une seule unité de commande nécessitant la mesure de tous les états ou les sorties du système. Cette structure rend généralement le calcul lent et complexe. Afin de surmonter cet inconvénient, les chercheurs se sont intéressés, lors des deux dernières décennies, à l'utilisation de la commande décentralisée [ION 86][SER 89][BOU 96][BOU 98a]. Cette dernière consiste à décomposer le système en plusieurs sous systèmes interconnectés pouvant être géographiquement éloignés. Chaque sous système est commandé par une station de commande locale qui n'a accès qu'aux informations locales du sous système, permettant ainsi la simplification de l'implémentation des algorithmes par rapport à la commande centralisé.

Le travail présenté dans ce mémoire concerne l'étude et le développement de méthodes de commande adaptative centralisée et décentralisée en utilisant différents types de réseaux de neurones artificiels à savoir les réseaux statiques, dynamiques bouclés (récurrents) et les réseaux à fonction de base radiale (à centres fixes et à centres adaptatifs dit aussi réseaux à base radiale généralisés). Ces méthodes sont appliquées par simulation sur un modèle non linéaire d'un robot manipulateur à trois degrés de libertés de type PUMA 560 [AMS 86]. Un robot manipulateur à  $n$  degrés de libertés peut toujours être modélisé par un système interconnecté où chaque liaison est considérée comme étant un sous-système couplé avec les autres sous-systèmes.

Le présent mémoire est organisé en quatre chapitres :

Dans le premier chapitre, plusieurs approches de commande adaptative centralisées directes neuronales ont été étudiées et appliquées par simulation au robot PUMA 560. La structure de commande utilisée est commune à toutes ces approches. Chaque approche se différencie par rapport aux autres par le type du réseau de neurone employé. Ainsi, les réseaux statiques, RBF (Radial Basis Function Gaussien) et récurrents ont été utilisés. L'analyse de la stabilité de la structure de commande utilisée a été effectuée utilisant la théorie de Lyapunov.

Dans le second chapitre, toutes les techniques de commande utilisées dans le premier chapitre ont été étendues au cas décentralisé et des résultats de simulation ont été obtenus pour le même robot.

Dans le troisième chapitre, nous présentons une méthode de commande centralisée adaptative par RNA développée par Ozaki [OZA 91]. Cette méthode est basée sur la technique du couple calculé. Nous proposons, ensuite, une extension de cette méthode au cas décentralisé où tous les réseaux suscités sont utilisés. Des résultats de simulation sont aussi donnés pour le robot PUMA 560.

Dans le dernier chapitre, nous proposons une structure de commande adaptative indirecte décentralisée par différents types de RNA. La technique est fondée sur les travaux introduit par Chen [CHE 91] dans le cas monovarié. Le schéma de commande est ensuite appliqué au même robot que les chapitres précédents.

## Introduction Générale

**E**N 1943, Mc Culloch et Pitts ont effectué la première étude systématique des Réseaux de Neurones Artificiels (RNA). Après quatre ans (1947), ils ont exploré des modèles pour la reconnaissance de formes lors des translations et rotations. En effet, un simple modèle de neurone, appelé perceptron, a été utilisé. Le perceptron n'était qu'un simple produit scalaire d'un vecteur d'entrées et un vecteur poids, suivi d'un élément à seuil. En 1949, D. O. Hebb a donné les bases de l'apprentissage des RNA. Avant cette période, il était connu que l'apprentissage biologique est le résultat des changements au niveau des neurones. En se basant, sur les études physiologiques et psychologiques, Hebb a présenté son hypothèse de l'apprentissage qui reste la base de tous les algorithmes d'apprentissage [BEN 93].

Les réseaux de Neurones Artificiels représente un moyen efficace pour la commande des systèmes, car ils sont construits directement à partir des données multidimensionnelles collectées par les capteurs ou obtenues par simulation. Un RNA a besoin de l'historique, valeurs valables obtenues par le passé, qui est souvent abondante, mais pas trop de théorie. Ainsi, leur adaptivité et non-linéarité font d'eux un moyen adéquat de prédiction, commande et optimisation des processus industriels.

Durant ces dernières années, les RNA ont été appliqués dans beaucoup de systèmes de commande, ils ont souvent donné satisfaction par rapport aux méthodes classiques [KUS 93][KWA 95a][KWA 95b][SHU 96a][SHU 96b][ZIA 96].

En plus de la facilité relative avec laquelle des relations compliquées peuvent être représentées par l'utilisation d'un ensemble de points, le RNA a d'autres avantages. Un exemple est celui de traitement parallèle qui réduit considérablement le temps de calcul, ce qui est très apprécié dans la commande des systèmes. Un autre avantage des RNA est que l'information est distribuée à travers un ensemble de poids, donc, l'erreur ou le non-fonctionnement de certaines connections ne provoque pas une dégradation dramatique du réseau, mais une dégradation graduelle, ce qui est différent des méthodes classiques de commande des systèmes qui échouent pour la moindre erreur. Cette capacité d'éviter les erreurs catastrophiques est une propriété très importante surtout pour un bon régulateur. De la même façon, si des entrées entachées d'erreurs sont présentées au RNA, il se peut qu'elles donnent des réponses correctes. Dans les régulateurs conventionnels, ce problème doit être bien étudié avant d'installer un régulateur.

Jusqu'ici, la commande adaptative, convient mieux que d'autres méthodes aux systèmes présentant des non-linéarités, a réussi à résoudre une partie du problème de perturbations connues [SLO 91][AST 89]. Mais, lorsqu'il s'agit de phénomènes incertains accompagnant les paramètres du modèle du processus contrôlé, la commande adaptative a montré ses limites. Elle ne peut s'adapter à tous les changements paramétriques incertains, car le contrôleur adaptatif est conçu pour être fixé à fonctionner en prévision des phénomènes plus ou moins prédictibles et continue à exécuter les mêmes tâches sans pouvoir réactualiser la loi de commande. Lorsque la dynamique du processus n'est pas correctement modélisée, il est impossible de faire converger les paramètres identifiés aux paramètres réels. C'est pour cette raison qu'il est nécessaire d'apprendre la loi de commande associée à chaque type

---

# *Chapitre 1*

## *Commande Adaptative Centralisée*

### *Directe par RNA*

---

#### **1.1. Introduction**

**L**A théorie des systèmes fournit des outils d'analyse et de synthèse parfaitement adaptée aux systèmes linéaires. Cependant, en pratique, est vu leur caractère non-linéaire des systèmes à commander, les méthodes linéaires sont limitées dans leurs applications, du fait qu'il n'est pas toujours possible de linéariser le système. D'où la nécessité de développer de nouvelles méthodes de commande permettant de commander les systèmes non linéaires avec leurs caractéristiques particulières. Depuis les premiers travaux jusqu'à présent, beaucoup d'études furent développées dans ce sens, notamment celles obtenues dans l'approche adaptative et qui ont donné naissance à des applications impressionnantes dans divers domaines pratiques [AST 89][GOO 84][LAN 79][SLO 88]. Cette approche a pour objectif d'obtenir certaines performances désirées par la réactualisation de certaines représentations déterminant le modèle du processus. La commande adaptative s'adapte, à un certain degré, aux changements au niveau du processus et son environnement. Mais, lorsque la variation des paramètres est continue, et faute de mémorisation les performances se dégradent. Ainsi, la commande adaptative ne convient plus à cette situation. Car, elle traite chaque situation comme une nouvelle situation, du fait qu'il y'a un manque de corrélation entre les situations passées et celles présentes.

L'utilisation des approximateurs universels, tel que les réseaux de neurones et les systèmes flous, dans la commande des systèmes non linéaires peut être interprétée comme une évolution naturelle des techniques de commande. Cette évolution est fondée sur plusieurs points :

- Capacités limitées des régulateurs classiques, nécessitant une étude détaillée de la dynamique du système devant les systèmes complexes.
- Analyse des non-linéarités dures et quelconques (la commande adaptative ne garantit pas la compensation des incertitudes non structurées).
- Absence d'information a priori sur le système à commander, considérant ce dernier comme une boîte noire entrée/sortie.
- La commande adaptative a un objectif temporel.

Les réseaux de neurones avec leurs énormes capacités d'apprentissage et de généralisation sont utilisés efficacement dans la commande des systèmes non linéaires complexes. Parmi les méthodes de synthèse des correcteurs les plus connus, dont les réseaux de neurones artificiels ont trouvé leurs applications directes, nous citons :

- *La commande inverse* [HUN 92][MIL 90] : le réseau est en série avec le système à commander. Il a pour objectif d'identifier le modèle inverse.

- *La commande adaptative par modèle de référence (MRAC)* [NAR 90a] : dans cette technique de commande le réseau fournit la commande nécessaire pour que le système suive un modèle de référence, et l'erreur de poursuite sera directement utilisée pour l'adaptation des poids du réseau.
- *La commande auto-ajustable (STR)* [CHE 90] : c'est une commande adaptative indirecte, dont le réseau identifie le modèle du système ensuite la commande est calculée à partir de ce modèle.
- *La commande supervisée* [HUN 92][MIL 90] : le réseau a pour objectif de recopier un régulateur existant afin d'améliorer leur performance par sa caractéristique de généralisation.
- *La commande par bouçlage linéarisant (Feed-back Linéarisation)* [OSA 91] : en se basant sur le modèle de connaissance du système, on synthétise un retour à base des réseaux de neurones, dans le but de linéariser le système à commander. Ensuite, une commande classique peut être appliquée sur le système linéaire composé du système réel et du retour neuronal.

Cependant, pour toutes les approches neuronales citées ci-dessus, aucune preuve de la stabilité des lois d'adaptation des paramètres du RNA n'a été établie. En effet, jusqu'à présent, on ne parvient toujours pas à faire le constat de la stabilité des schémas de commande proposés, car il est difficile d'étudier la stabilité du système global composé notamment d'un ou de plusieurs réseaux de neurones, vu le caractère non linéaire de ces derniers. Le travail de Lewis [LEW 93] a donné un grand essor pour la commande neuronale, car l'auteur a pu trouver une solution optimale même finale du problème de la stabilité dans les systèmes de commande intelligente. Cependant, un grand nombre de publications a vu le jour dans divers domaines d'application, en se basant sur cette démonstration de la stabilité [KWA 95a][KWA 95b][ZIA 96][SHU 96a][SHU 96b].

A travers ce chapitre, nous présenterons la nouvelle stratégie de commande adaptative directe [LEW 93] [KWA 95b] qui combine entre la théorie de la stabilité de Lyapunov, la théorie d'approximation par les réseaux de neurones artificiels ; statiques, à fonction de base radiale gaussienne (à centres fixes et centres variables), ainsi que les réseaux dynamiques bouclés (récurrents). Et nous présentons également la commande prédictive adaptative directe par RNA DCAL-NN (Desired Compensation Adaptive Law-based Neural Networks) [BOI 88][KWA 95b].

## 1.2. Commande Adaptative Centralisée Directe par RNA Statique

Les RNA créent une corrélation de données par l'utilisation d'un nombre élevé d'éléments de calculs. Ces éléments de calculs se combinent pour former l'équivalent d'une fonction de transfert, qui fait correspondre les données d'entrées à ceux de sorties. Ils n'utilisent pas d'équations explicites à programmer pour créer cette corrélation entrée sortie. Les RNA apprennent à partir de la lecture des échantillons de données entrées/sorties et s'adaptent pour créer cette correspondance paire par paire de données. Le réseau obtenu est capable de répondre à de nouvelles données, en produisant la meilleure estimation de sortie. En effet, ces caractéristiques ont donné de nouvelles issues aux problèmes de commande et d'identification des systèmes non linéaires.

Toutefois, le traitement des RNA se heurte encore aujourd'hui à des difficultés résidant dans le passage obligatoire par un apprentissage hors ligne du bloc neuronal, l'incertitude sur l'initialisation des paramètres du RNA ainsi que la difficulté d'établir la stabilité des schémas de commande. Pour y remédier à cela, il est nécessaire d'approfondir les recherches dans l'étude de la stabilité des réseaux de neurones.

Dans ce sens, Lewis propose une nouvelle approche neuronale [LEW 93] appliquée en robotique et basée sur l'entraînement, en ligne, des paramètres du RNA afin de reconstruire une certaine fonction couple non linéaire ainsi que l'association d'un proportionnelle dérivateur à gain fixe. Cette approche a permis de pallier le problème de la stabilité du système bouclé, ainsi qu'aux problèmes de l'initialisation et l'apprentissage hors ligne.

Dans cette section, nous présentons un nouveau algorithme d'apprentissage en ligne basé sur la théorie de la stabilité de Lyapunov, garantissant la stabilité du système global, la poursuite parfaite ainsi que la bornitude des signaux de commande et de l'erreur de poursuite et même la convergence asymptotique de cette dernière vers zéro.

### 1.2.1. Position du Problème

Considérons un bras manipulateur à  $n$  degrés de libertés, ayant une dynamique décrite par l'équation:

$$M(\theta) \cdot \ddot{\theta} + V_m(\theta, \dot{\theta}) \cdot \dot{\theta} + H(\dot{\theta}) + G(\theta) + \tau_d = \tau \quad (I.1)$$

- $\theta$  : Vecteur des variables articulaires généralisées (position).
- $\dot{\theta}$  : Vecteur des vitesses angulaires.
- $M(\theta)$  : Matrice d'inertie symétrique définie positive ( $n \times n$ ).
- $V_m(\theta, \dot{\theta})$  : Matrice de centrifuge et de coriolis.
- $H(\dot{\theta})$  : Vecteur des forces des frottements visqueux.
- $G(\theta)$  : Vecteur des couples de la gravitation.
- $\tau_d$  : Vecteur regroupant les perturbations et les dynamiques non modélisées.
- $\tau$  : Vecteur des couples appliquer à chaque liaisons.

On définit comme suit :

- L'erreur de poursuite : 
$$e(t) = \theta_d(t) - \theta(t) \quad (I.2)$$

- L'erreur de poursuite filtrée : 
$$r(t) = \dot{e}(t) + \Lambda \cdot e(t) \quad (I.3)$$

Avec

- $\theta_d(t)$  : Vecteur de trajectoire désirée.
- $\Lambda \in \mathcal{R}^{m \times n}$  : Matrice symétrique définie positive.

On dérivant l'erreur filtrée, on obtient

$$\dot{r} = \ddot{e} + \Lambda \cdot \dot{e} = \ddot{\theta}_d - \ddot{\theta} + \Lambda \cdot \dot{e} \quad (I.4)$$

$$M(\theta) \cdot \dot{r} = M(\theta) \cdot \ddot{\theta}_d - M(\theta) \cdot \ddot{\theta} + M(\theta) \cdot \Lambda \cdot \dot{e} \quad (I.5)$$

d'après (I.1), on aura

$$M(\theta) \cdot \dot{r} = -V_m(\theta, \dot{\theta}) \cdot r - \tau(t) + f(x) + \tau_d(t) \quad (I.6)$$

Avec  $f(x)$  la fonction couple non linéaire tel que :

$$f(x) = M(\theta)(\ddot{\theta}_d + \Lambda \cdot \dot{e}) + V_m(\theta, \dot{\theta}) \cdot (\dot{\theta}_d + \Lambda \cdot e) + G(\theta) + H(\dot{\theta}) \quad (I.7)$$

Où

$$x = [e^T \quad \dot{e}^T \quad \theta_d^T \quad \dot{\theta}_d^T \quad \ddot{\theta}_d^T]^T$$

Le problème de la commande adaptative neuronale se pose comme suit :

Pour un bras manipulateur décrit par (I.1), déterminer les lois d'ajustement des poids du RNA qui permettent d'estimer, en ligne, la fonction couple non linéaire définie dans (I.7) ainsi que le couple adéquat  $\tau(t)$  tel que l'erreur de poursuite converge asymptotiquement vers zéro.

$$\lim_{t \rightarrow \infty} |e(t)| = \lim_{t \rightarrow \infty} |\theta(t) - \theta_d(t)| = 0$$

## I.2.2. Structure du Contrôleur Adaptatif Neuronal

A partir de la théorie d'approximation des fonctions, un RNA statique pouvait implémenter n'importe quelle fonction non linéaire pourvu que celle-ci soit bornée et absolument intégrable. Un théorème résume les résultats d'approximation est le suivant [BEL 97] :

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

$$\hat{F}(x_1, \dots, x_n) = \sum_{i=1}^N C_i \phi\left(\sum_{j=1}^n W_{ij} x_j + \theta_j\right)$$

satisfait :

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

A partir de ce théorème, le réseau qui sera utilisé est un réseau de neurones statique à trois couches, une couche d'entrée, une couche cachée et une couche de sortie (Fig. I.1).

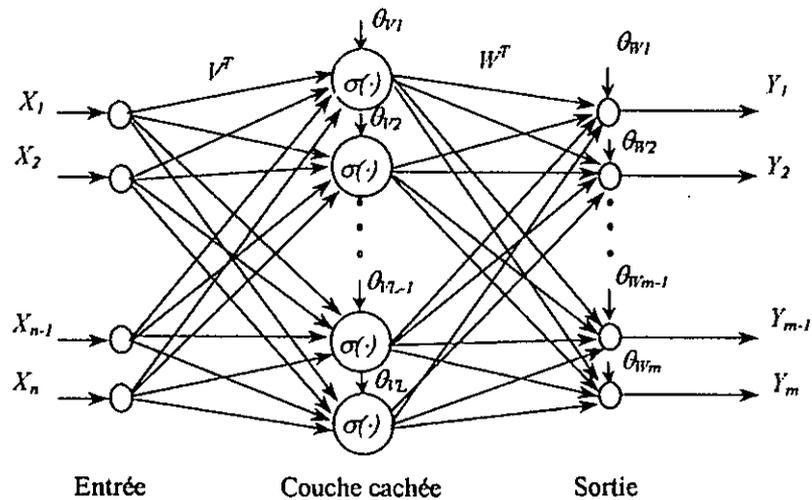


Fig. 1.1 schéma du réseau de neurones artificiel statique.

La sortie du réseau est la suivante:

$$y_i = \sum_{j=1}^L [W_{ij} \cdot \sigma \left[ \sum_{k=1}^n V_{jk} X_k + \theta_{vj} \right] + \theta_{wi}] \quad i=1, \dots, m \quad (1.8)$$

- $X \in \mathbb{R}^n$ : vecteurs d'entrée.
- $Y \in \mathbb{R}^m$ : vecteurs de sortie.
- $L$  : nombre de neurones de la couche cachée.

Avec:

- $\sigma(\cdot)$  : La fonction d'activation (sigmoïde).
- $[V^T] \in \mathbb{R}^{L \times n}$  : Matrice des poids (entrée/couche cachée).
- $[W^T \in \mathbb{R}^{m \times L}]$  : Matrice des poids (couche cachée/sortie).
- $\theta_{vj}, \theta_{wi}$  : Les Biais.

L'expression matricielle de l'équation (1.8) est:

$$Y = W^T \cdot \sigma(V^T \cdot X + \Theta_v) + \Theta_w \quad (1.9)$$

Avec:

$$\sigma(Z) = [\sigma(Z_1), \dots, \sigma(Z_L)] \text{ pour } Z \in \mathbb{R}^L$$

NB : Dans ce paragraphe nous avons donné une description très succincte du RNA. Pour plus de détail nous renvoyons le lecteur aux références suivantes [NAR 90a][YAD 98][HAM 95].

Dans ce schéma de commande, le rôle attribué au réseau de neurones suscité est d'estimer en temps réel la fonction couple non linéaire  $f(x)$  défini par (1.6). Donc, on assumera l'existence des poids idéaux  $W$  et  $V$ , qui permettent la reconstruction parfaite de la fonction  $f(x)$  qui s'exprime par

$$f(x) = W^T \cdot \sigma(V^T \cdot x) + \varepsilon(x) \quad (1.10)$$

$\varepsilon(x)$  est l'erreur de reconstruction de la fonction  $f(x)$ .

$$x = [e^T \quad \dot{e}^T \quad \theta_d^T \quad \dot{\theta}_d^T \quad \ddot{\theta}_d^T]^T.$$

On désigne l'estimée de la fonction  $f(x)$  par  $\hat{f}(x)$ , tel que :

$$\hat{f}(x) = \hat{W}^T \cdot \sigma(\hat{V}^T \cdot x) \quad (I.11)$$

$\hat{W}$  et  $\hat{V}$  les poids estimés de  $W$  et  $V$ .

On définit aussi

– L'erreur de l'estimation de la fonction  $\tilde{f}(x) = f(x) - \hat{f}(x)$  (I.12)

– L'erreur de l'estimation des poids du RNA  $\begin{cases} \tilde{V} = V - \hat{V} \\ \tilde{W} = W - \hat{W} \\ \tilde{Z} = Z - \hat{Z} \end{cases}$  (I.13)

Avec  $Z^T = [W^T \quad V^T]$ .

– L'erreur de sortie de la couche cachée :

$$\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(V^T \cdot x) - \sigma(\hat{V}^T \cdot x) \quad (I.14)$$

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

$$\sigma(V^T \cdot x) = \sigma(\hat{V}^T \cdot x) + \sigma'(\hat{V}^T \cdot x) \cdot \tilde{V}^T \cdot x + O(\tilde{V}^T \cdot x)^2 \quad (I.15)$$

où

$$\sigma'(\hat{Z}) = \left. \frac{d(\sigma(Z))}{dZ} \right|_{Z = \hat{Z}}$$

notons:  $\hat{\sigma}' = \sigma'(\hat{V}^T x)$ , on aura

$$\tilde{\sigma} = \sigma'(\hat{V}^T \cdot x) \cdot \tilde{V}^T \cdot x + O(\tilde{V}^T \cdot x)^2 = \hat{\sigma}' \cdot \tilde{V}^T \cdot x + O(\tilde{V}^T \cdot x)^2 \quad (I.16)$$

avec  $O(\tilde{V}^T \cdot x)^2 = [\sigma(V \cdot x) - \sigma(\hat{V}^T \cdot x)] - \sigma'(\hat{V}^T \cdot x) \cdot \tilde{V}^T \cdot x$

Pour pouvoir établir ce schéma de commande, plusieurs hypothèses et propriétés doivent être prises en considération

### Hypothèse 1:

Les poids idéaux sont bornés, c'est-à-dire qu'il existe des constantes  $V_m$ ,  $W_m$  et  $Z_m$  tel que:

$$\begin{cases} \|V\|_F \leq V_m \\ \|W\|_F \leq W_m \\ \|Z\|_F \leq Z_m \end{cases} \quad (\text{I.17a})$$

avec  $\|\cdot\|_F$  est la norme Frobenienne définit comme suit:

**Définition :**

Soit **A**, **B** et **X** des matrices tel que :

$$A \in \mathfrak{R}^{m \times n}, \quad B \in \mathfrak{R}^{m \times n}, \quad \text{et} \quad X \in \mathfrak{R}^n$$

La norme frobenienne est définit par:

$$\begin{aligned} \|A\|_F^2 &= \text{trace}(A^T \cdot A) = \sum_{i,j} a_{ij}^2 \\ \langle A, B \rangle_F &= \text{trace}(A^T \cdot B) \\ \|A \cdot X\|_F &\leq \|A\|_F \cdot \|X\| \end{aligned}$$

**Hypothèse 2 :**

Le vecteur trajectoire de référence noté  $q_d$  est borné, dans le sens ou  $\|\theta_d\|$ ,  $\|\dot{\theta}_d\|$ ,  $\|\ddot{\theta}_d\|$  sont majorés par une constante positive  $Q_d$ , tel que:

$$\{\|\theta_d\|, \|\dot{\theta}_d\|, \|\ddot{\theta}_d\|\} \leq Q_d \quad (\text{I.17b})$$

**Propriété 1 :**

Le vecteur  $x(t)$  donné par (I.7) vérifie à chaque instant l'équation suivante:

$$\|x\| \leq c_1 \cdot Q_d + c_2 \cdot \|r\| \quad (\text{I.17c})$$

$c_1, c_2$  : constante positive.

**Propriété 2 :**

Pour la fonction sigmoïde, l'erreur du 2<sup>ième</sup> ordre dans le développement en série de Taylor définit dans (I.17) vérifie l'équation suivante :

$$\|O(\tilde{V}^T \cdot x)\|^2 \leq c_3 + c_4 \cdot Q_d \cdot \|\tilde{V}\|_F + c_5 \cdot \|\tilde{V}\|_F \cdot \|r\| \quad (\text{I.17d})$$

$c_3, c_4$  et  $c_5$  des constantes positives calculables.

**Propriété 3 :**

Pour un bras manipulateur défini par (I.1), on a:

1. La matrice d'inertie  $M(\theta)$  est symétrique définie positive avec:

$$m_1 I \leq M(\theta) \leq m_2 \cdot I$$

$m_1, m_2$  : Constante positive.

2.  $V_m(\theta, \dot{\theta}) \leq V_b(\theta) \cdot \|\dot{\theta}\|$ , avec  $V_b(\theta) \in C^1$ .

3. La matrice  $\dot{M}(\theta, \dot{\theta}) - 2 \cdot V_m(\theta, \dot{\theta})$  est antisymétrique.

4. Le couple représentant les perturbations  $\tau_d$  est borné par:

$$\|\tau_d\| < b_d$$

$b_d$  : Constante positive.

**I.2.3. Schéma de Commande**

A partir de la fonction couple non linéaire estimée, nous écrivons l'expression du contrôleur

$$\tau(t) = \hat{f}(x) + K_v \cdot r(t) + v_R(t) \quad (\text{I.18})$$

avec

- $K_v \in \mathfrak{R}^{n \times n}$  : matrice gain symétrique définie positive;
- $v_R(t) \in \mathfrak{R}^n$  : terme robuste.

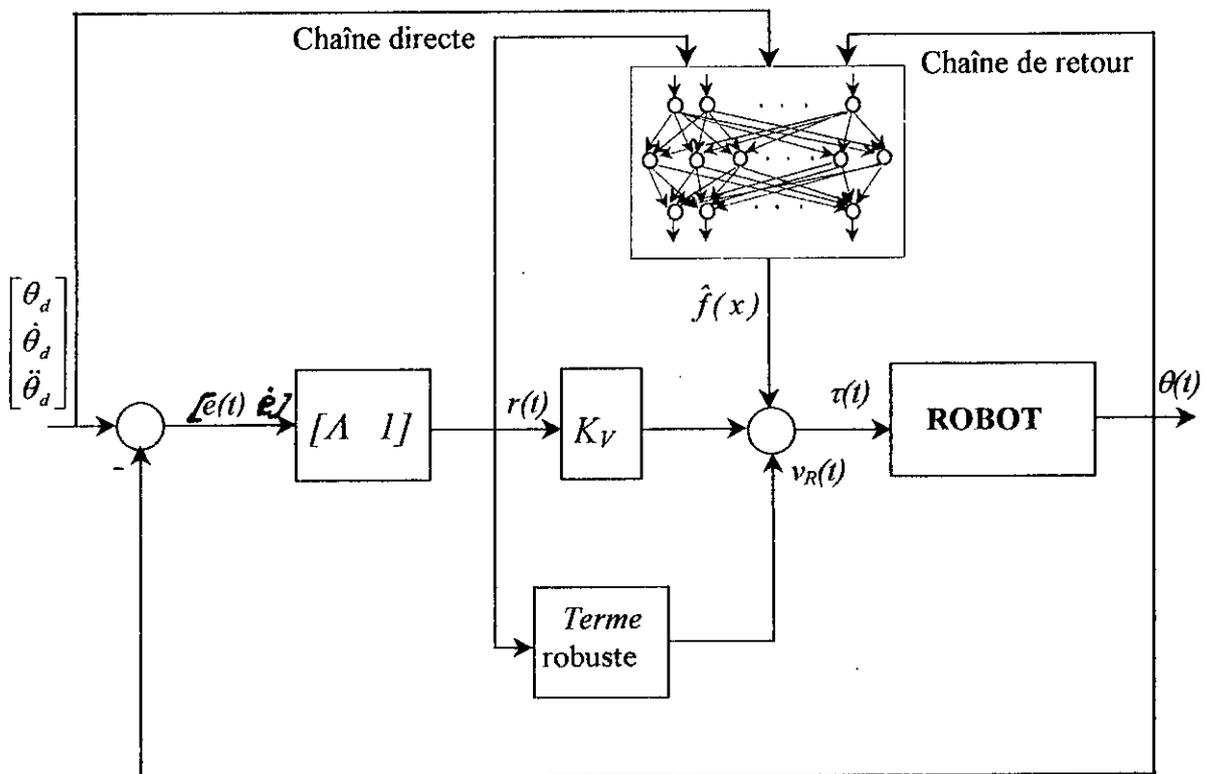


Fig. I.2 structure de base du système de commande adaptatif neuronal.

Le premier terme de l'équation (I.18) correspond à la sortie du RNA (éq.I.11) et considéré comme l'estimation de la fonction couple non linéaire définie dans (I.7). Le second correspond à l'équation classique d'un régulateur proportionnel dérivateur à gain fixe.

Le couple additionnel  $v_R(t)$  est introduit afin d'assurer plus de robustesse en présence d'erreur élevée due à l'estimation.

### Etude de la dynamique de l'erreur filtrée

En introduisant l'équation (I.18) dans l'équation (I.6), la dynamique de l'erreur filtrée sera

$$M(\theta) \cdot \dot{r} = -(K_v + V_m(\theta, \dot{\theta})) \cdot r + f(x) - \hat{f}(x) + \tau_d - v_R(t)$$

et de (I.10) et (I.11), on obtient

$$M(\theta) \cdot \dot{r} = -(K_v + V_m(\theta, \dot{\theta})) \cdot r + W^T \cdot \sigma(V^T \cdot x) - \hat{W}^T \cdot \sigma(\hat{V}^T \cdot x) + (\varepsilon + \tau_d) - v_R(t)$$

d'où

$$M(\theta) \cdot \dot{r} = -(K_v + V_m(\theta, \dot{\theta})) \cdot r + \tilde{W}^T \cdot \hat{\sigma} + \hat{W}^T \cdot \tilde{\sigma} + \tilde{W}^T \cdot \tilde{\sigma} + (\varepsilon + \tau_d) - v_R(t) \quad (I.19)$$

D'après (I.16) :

$$M(\theta) \cdot \dot{r} = -(K_v + V_m(\theta, \dot{\theta})) \cdot r + \tilde{W}^T \cdot \hat{\sigma} + \hat{W}^T \cdot \hat{\sigma}' \cdot \tilde{V}^T \cdot x + w(t) - v_R(t) \quad (I.20)$$

avec:

$$w(t) = \tilde{W}^T \cdot \hat{\sigma}' \cdot \tilde{V}^T \cdot \dot{x} + W^T \cdot O(\tilde{V}^T \cdot x)^2 + (\varepsilon + \tau_d) \quad (I.21)$$

Le terme  $w(t)$  regroupe les perturbations ainsi que les erreurs d'estimation infiniment petites.

### 1.2.4. Algorithme d'Adaptation Paramétrique

Pour pouvoir estimer fidèlement la fonction non linéaire définie dans (I.7), il faut concevoir un algorithme d'adaptation des paramètres du RNA garantissant la stabilité du système en boucle fermée.

L'ajustement se fait en temps réel. Le problème majeur des autres approches développées jusqu'ici étant l'apprentissage au préalable de la fonction à estimer ainsi que le choix de l'initialisation. Dans [LEW 93], on propose un algorithme d'adaptation en ligne qui n'exige pas d'apprentissage hors ligne. De plus, tous les paramètres du RNA sont initialisés à zéro.

Dans ce qui suit, nous donnerons deux algorithmes d'apprentissage en ligne, le premier conçu pour le cas idéal c'est-à-dire lorsque l'erreur de reconstruction  $\varepsilon(x)$  est nulle et les perturbations ainsi que les erreurs de modélisation sont nulles, le second étant plus général.

#### a- Cas Idéal

Si dans l'équation (I.20)  $w(t)=0$  et  $v(t)=0$ , alors les lois d'ajustements des poids du RNA sont données par

$$\dot{\hat{W}}(t) = F_w \cdot \sigma(\hat{V}^T \cdot x) \cdot r^T \quad (I.22a)$$

$$\dot{\hat{V}}(t) = F_v \cdot x \cdot [(\sigma'(\hat{V}^T \cdot x))]^T \cdot \hat{W} \cdot r^T \quad (I.22b)$$

avec:  $F_w$  et  $F_v$  sont des matrices constantes symétriques définies positives.

Ces lois garantissent une stabilité asymptotique du système globale, ainsi qu'une convergence asymptotique de l'erreur filtrée vers zéro. Il s'ensuit que tous les signaux de commande sont uniformément bornés.

### Etude de la Stabilité

Soit la fonction positive candidate de Lyapunov suivante:

$$L(t) = \frac{1}{2} r^T M r + \frac{1}{2} \text{tr}(\tilde{W}^T F_w^{-1} \tilde{W}) + \frac{1}{2} \text{tr}(\tilde{V}^T F_v^{-1} \tilde{V}) \quad (I.23)$$

en dérivant (I.23) par rapport au temps, on obtient

$$\dot{L}(t) = r^T \dot{M} r + \frac{1}{2} r^T \dot{M} r + \text{tr}(\tilde{W}^T F_w^{-1} \dot{\tilde{W}}) + \text{tr}(\tilde{V}^T F_v^{-1} \dot{\tilde{V}}) \quad (I.24a)$$

en remplaçant (I.20) dans (I.24a), on aura

$$\dot{L}(t) = -r^T K_v r + \frac{1}{2} r^T (\dot{M} - 2V_m) r + \text{tr}(\tilde{W}^T (F_w^{-1} \dot{\tilde{W}} + \hat{\sigma} \cdot r^T)) + \text{tr}(\tilde{V}^T (F_v^{-1} \dot{\tilde{V}} + x \cdot r^T \hat{W}^T \hat{\sigma}')) \quad (I.24b)$$

$(\dot{M} - 2V_m)$  étant antisymétrique (propriété 3), donc:

$$r^T (\dot{M} - 2V_m) r = 0.$$

D'après (I.13), on aura:  $\begin{cases} \dot{\tilde{W}} = -\hat{W} \\ \dot{\tilde{V}} = -\hat{V} \end{cases}$ , on remplace (I.22a) et (I.22b) dans (I.24b), on a:

$$\dot{L}(t) = -r^T \cdot K_v \cdot r < 0$$

ce qui constitue une garantie de la stabilité au sens de Lyapunov du schéma de commande proposé, ainsi, l'adaptation définie dans (I.22) assure la bornitude de tous les signaux et la convergence asymptotique des erreurs de poursuite vers zéro.

**b- Cas Général**

En présence des perturbations, des dynamiques non modélisées et d'erreur de reconstruction fonctionnelle, il est nécessaire de modifier les lois d'adaptation et/ou d'ajouter un terme robuste a fin d'assurer les performances souhaitées.

**Propriété 4 :**[KWA 95b]

On suppose que le terme  $w(t)$  de l'équation (I.21) est borné par un polynôme d'état :

$$\begin{cases} \|w(t)\| \leq \rho(z, \hat{W}, \hat{V}) \\ \rho(z, \hat{W}, \hat{V}) \leq c_0 + c_1 \|z\|_F + c_2 \|z\|^2 + c_3 \|\hat{W}\|_F + c_4 \|\hat{V}\|_F + c_5 \|\hat{V}\|_F \|\hat{W}\|_F = S\varphi \end{cases} \quad (I.25)$$

$c_i$  ( $i=1, \dots, 5$ ) sont des constantes positives bornées et dépendent de la trajectoire désirée, des propriétés physiques de robot (i.e., masse et longueur des liaisons, coefficients de frottements, etc.), la borne de la perturbation et de l'erreur et les gains de commande. Les vecteurs  $z$ ,  $S$  et  $\varphi$  utilisés dans (I.25) sont définis par :

$$\begin{aligned} z &= [e^T \quad r^T]^T, \\ S &= [1 \quad \|z\| \quad \|z\|^2 \quad \|\hat{W}\|_F \quad \|\hat{V}\|_F \quad \|\hat{V}\|_F \|\hat{W}\|_F] \\ \varphi &= [c_0 \quad c_1 \quad c_2 \quad c_3 \quad c_4 \quad c_5]^T. \end{aligned}$$

Le terme robuste  $v_R(t)$  est donné par :

$$v_R(t) = \frac{r(t)(S\hat{\varphi})^2}{(S\hat{\varphi})\|r(t)\| + \delta(t)} \quad (I.26)$$

et  $\dot{\delta}(t) = -\gamma\delta(t)$ ,  $\gamma > 0$ ,  $\delta(0)$  : constante positive.

La loi d'adaptation du paramètre  $\varphi$  est

$$\dot{\hat{\varphi}} = \Gamma S^T \|r(t)\| = -\tilde{\varphi} \quad (I.27)$$

avec  $\Gamma$  est une matrice symétrique définie positive et  $\tilde{\varphi} = \varphi - \hat{\varphi}$

**Adaptation des Poids**

$$\dot{\hat{W}} = F_W \hat{\sigma} \cdot r \quad (I.28a)$$

$$\dot{\hat{V}} = F_V x \cdot (\hat{\sigma}' \cdot \hat{W} \cdot r)^T \quad (I.28b)$$

avec:  $F_W = F_W^T > 0$ ,  $F_V = F_V^T > 0$ .

**Etude de la stabilité**

Considérons la fonction candidate de Lyapunov suivante :

$$L(t) = \frac{1}{2} r^T M r + \frac{1}{2} e^T K_p e + \frac{1}{2} \text{tr}(\tilde{W}^T F_W^{-1} \tilde{W}) + \frac{1}{2} \text{tr}(\tilde{V}^T F_V^{-1} \tilde{V}) + L_R \quad (1.29)$$

$$L_R = \frac{1}{2} \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} + \frac{\delta}{\gamma}$$

En dérivant (1.29) et en utilisant (1.20), nous obtenons

$$\dot{L} \leq -z^T Q z + \dot{L}_1 + \dot{L}_2$$

avec

$$z = \begin{bmatrix} e^T & r^T \end{bmatrix}^T,$$

$$Q = \begin{bmatrix} K_p \Lambda & -\frac{K_p}{2} \\ -\frac{K_p}{2} & K_v \end{bmatrix} > 0 \text{ pour un choix approprié des matrices des gains } K_p \text{ et } K_v,$$

$$\dot{L}_1 = \|r\| S \hat{\varphi} - r^T v_R - \delta, \quad (1.30a)$$

$$\dot{L}_2 = \text{tr}[\tilde{W}^T F_W^{-1} \dot{\tilde{W}} + \tilde{W} \hat{\sigma}^T] + \text{tr}[\tilde{V}^T F_V^{-1} \dot{\tilde{V}} + \tilde{V}^T x(\hat{\sigma}^T r)^T] \quad (1.30b)$$

De (1.26) et (1.30a), on obtient

$$\begin{aligned} \dot{L}_1 &= \|r\| S \hat{\varphi} - \frac{r^T r (S \hat{\varphi})^2}{(S \hat{\varphi}) \|r\| + \delta} - \delta \\ &\leq \frac{\delta \|r\| S \hat{\varphi}}{(S \hat{\varphi}) \|r\| + \delta} - \delta \\ &\leq 0. \end{aligned}$$

La substitution des lois d'adaptation des poids (1.28) dans (1.30b) donne

$$\dot{L}_2 = 0 \quad (1.31)$$

La combinaison des résultats ci-dessus, donne

$$\dot{L} \leq -z^T Q z, \quad Q > 0 \quad (1.32)$$

(1.32) implique  $L$ ,  $z$ ,  $e$ ,  $r$ ,  $\tilde{W}$ ,  $\tilde{V}$ , et  $\hat{\varphi}$  sont tous bornés. De la forme de (1.32), il est facile de montrer que le carré de  $z(t)$  est intégrable; d'où les carrés de  $e$  et  $r$  sont intégrables. De l'équation de l'erreur en

boucle fermé (1.19), il est aussi facile de montrer que  $\dot{r}$  est borné, par conséquent,  $r$  est uniformément continu. Par le Lemme de Barbalat [KWA95], nous savons que  $e$  et  $\dot{e}$  tendent asymptotiquement vers zéro.

### 1.2.5. Etude par Simulation :

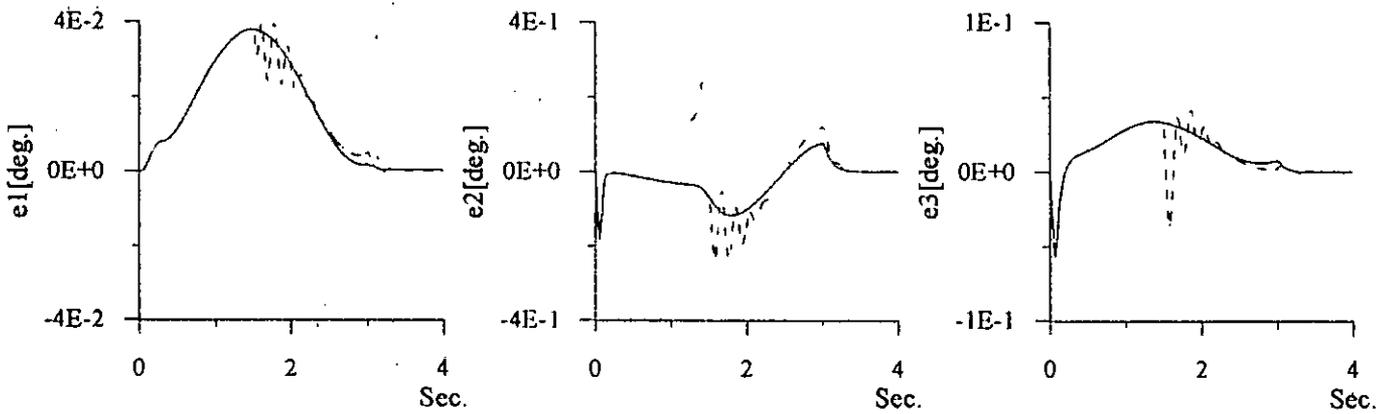
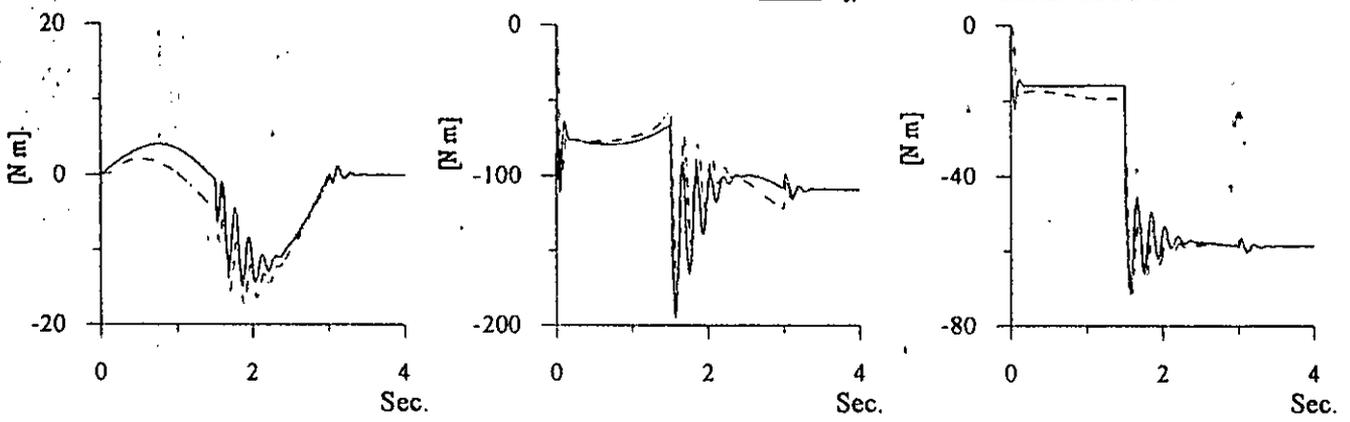
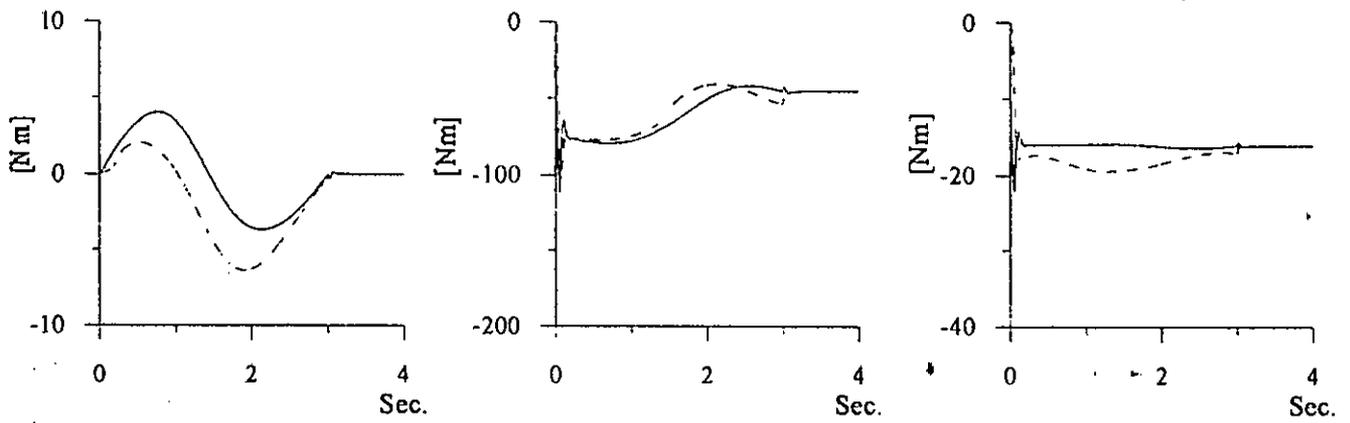
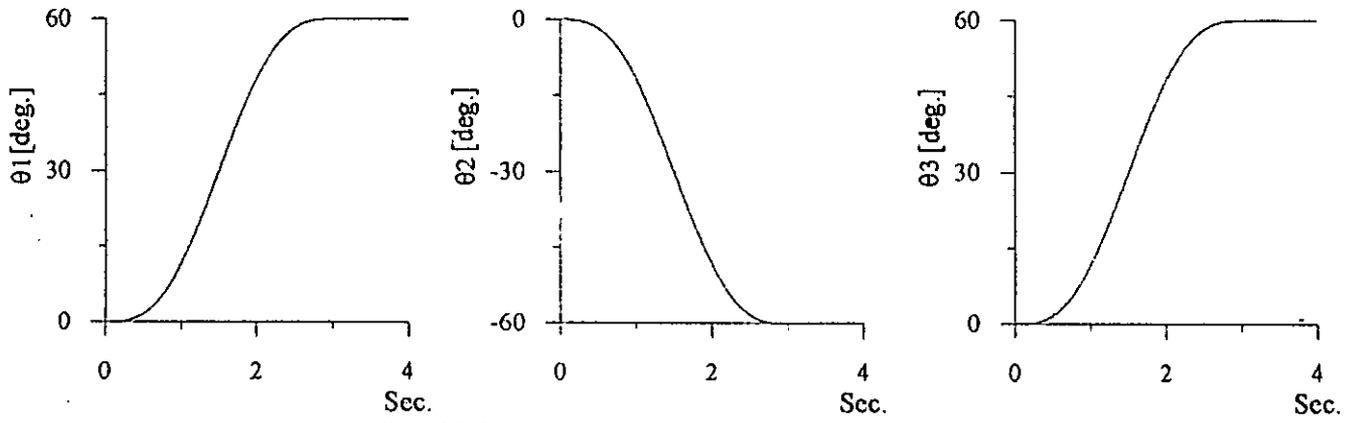
Pour la validation de la stratégie de commande présentée, des simulations ont été effectuées sur les trois premières articulations d'un robot de type PUMA 560. *Pour faciliter la compréhension et pouvoir établir une comparaison entre les résultats des différents schémas de commande présentés dans cette thèse, on procède à des simulations identiques pour chaque schéma.*

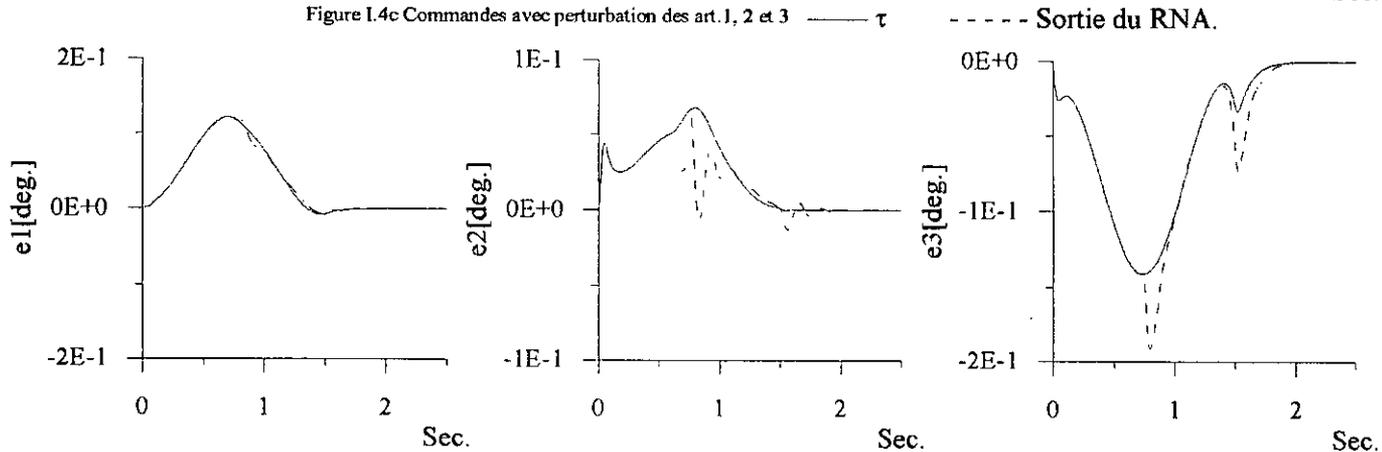
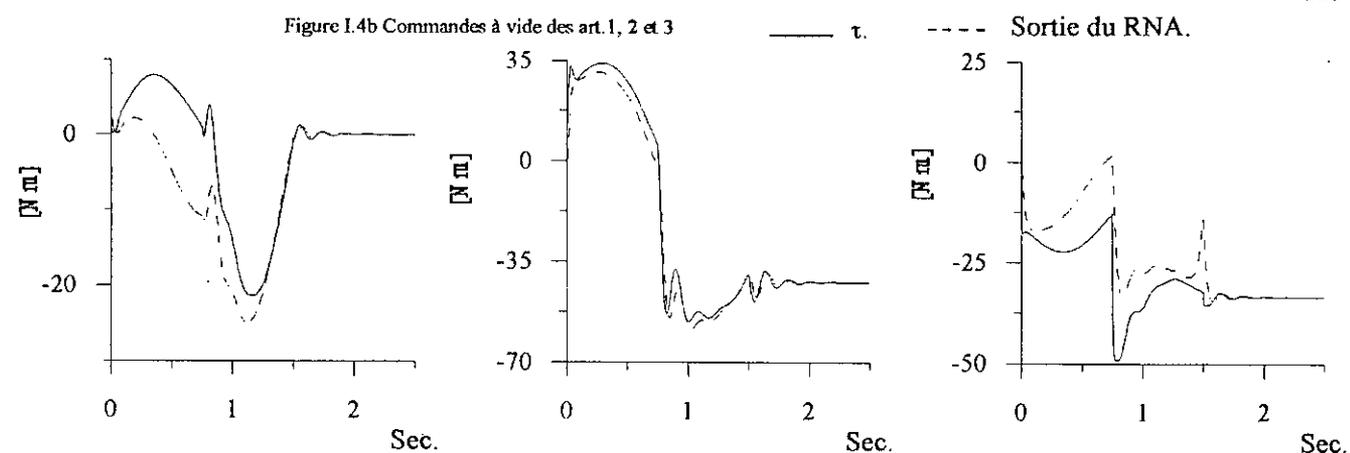
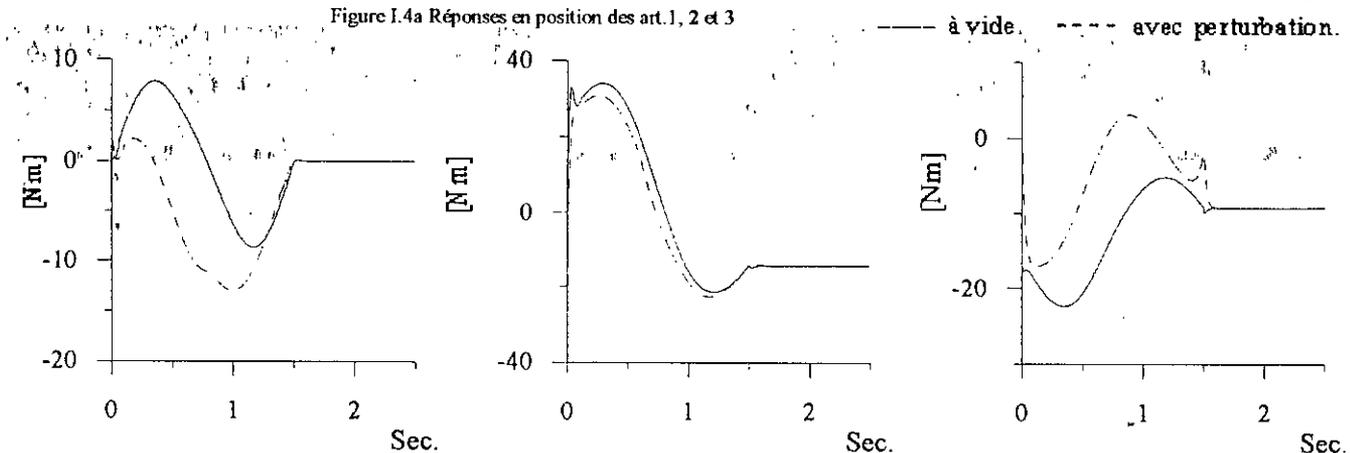
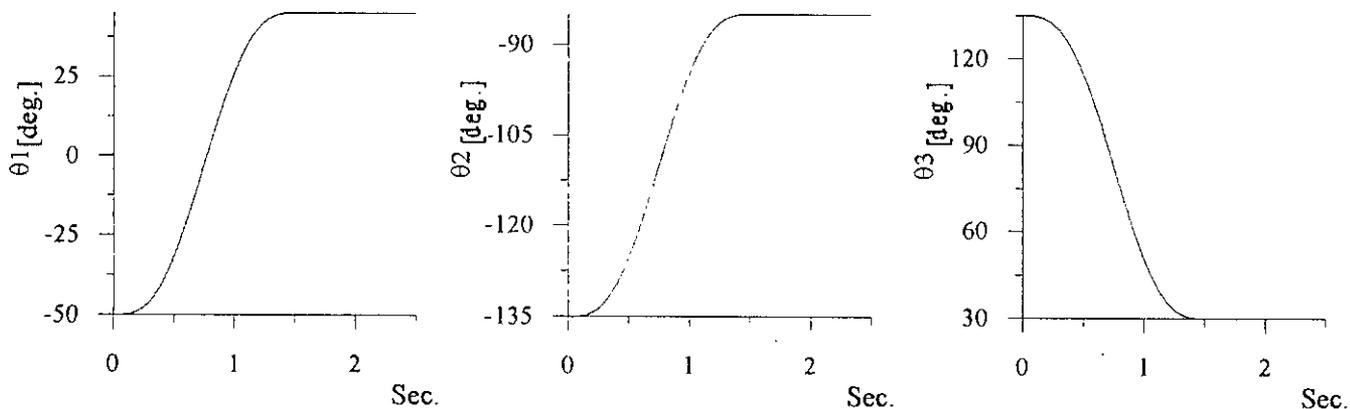
Nous avons opté pour trois réseaux de neurones MISO (Multi Input Simple Output) comportant : 15 entrées, 10 neurones dans la couche cachée, tous les poids des réseaux sont initialisés à zéro.

Le schéma de commande présenté a été testé pour la poursuite d'une trajectoire cycloïdale assurant une continuité en position, vitesse et accélération et allant de  $[0^\circ, 0^\circ, 0^\circ]$  à  $[60^\circ, -60^\circ, 60^\circ]$  en trois seconde. Sur la figure (1.3), on montre l'efficacité et les performances de cette méthode de commande sans et avec perturbation (une variation brusque de 100% de tous les paramètres a été effectuée ainsi qu'une variation de charge de 10kg au milieu du parcours).

Il a été montré dans [LEA 91] que la trajectoire imposée précédemment ne permet pas une véritable évaluation des performances du commande, car elle n'excite pas toutes les dynamiques du bras manipulateur. Ainsi, il a été proposé une trajectoire standard, dite de Leahy pour le test des performances, celle ci consiste au passage de la position  $[-50^\circ, -135^\circ, 135^\circ]$  à la position  $[45^\circ, -85^\circ, 30^\circ]$  en 1.5 seconde. Une manière d'évaluer la robustesse de n'importe quel schéma, serait d'utiliser cette dernière trajectoire en gardant les mêmes paramètres de la structure de commande.

La figure (1.4) montre les résultats obtenus pour la commande en poursuite de la trajectoire de Leahy utilisant le même test de perturbation. Nous constatons que la technique donne des performances de poursuite remarquables.





### 1.3 Commande Prédicative Adaptative Directe par RNA Statique

Dans cette section, nous présentons une technique de commande neuronale basée sur l'approche du DCAL (Desired Compensation Adaptive Law). L'idée de cette dernière est de remplacer la position, la vitesse et l'accélération réelles par celles de la trajectoire désirée, donc elle permet de calculer hors ligne les fonctions dépendantes de la position, vitesse et accélération. De ce fait, elle réduit considérablement le temps de calcul en ligne. On peut noter que l'avantage majeur de cette technique est la non-dépendance du réseau neuronal des sorties réelles du robot. Cependant, le réseau est un régulateur universel qui ne nécessite pas une étude préliminaire sur la dynamique du robot [KWA 95b].

#### **Propriété 5** Approximateurs neuronaux

Utilisant un RNA à trois couches pour approximer la fonction couple non linéaire du robot, en remplaçant les sorties réelles par celles désirées, donné par l'équation (I.7); il vient

$$f_d(x) = M(\theta_d)\ddot{\theta}_d + Vm(\theta_d, \dot{\theta}_d) \cdot \dot{\theta}_d + G(\theta_d) + H(\dot{\theta}_d) \quad (I.33)$$

$$f_d(\theta_d, \dot{\theta}_d, \ddot{\theta}_d) = W^T \cdot \sigma(V^T \cdot x) + \varepsilon(x) \quad (I.34)$$

Où  $f_d(x)$  désigne la fonction couple non linéaire utilisant les sorties désirées et  $x = [\theta_d^T \ \dot{\theta}_d^T \ \ddot{\theta}_d^T]^T$ .

De l'hypothèse 2, la trajectoire désirée est bornée, ce qui garantit la bornitude de la norme de l'erreur de reconstruction de la fonction couple non linéaire

$$\|\varepsilon(x)\| \leq \varepsilon_b$$

Avec  $\varepsilon_b$  est une constante positive.

#### 1.3.1. Schéma de Commande

La structure de base du contrôleur adaptatif neuronal est montrée dans la figure (I.5). L'action du couple commande est la suivante:

$$\tau(t) = \hat{f}_d(x) + K_v \cdot r(t) + v_R(t) \quad (I.35)$$

avec:

- $\hat{f}_d(\cdot)$  : est l'estimation de la fonction  $f_d$  à chaque instant,
- $K_v \in \mathfrak{R}^{n \times n}$  : matrice des gains symétrique définie positive,
- $v_R(t) \in \mathfrak{R}^n$  : terme robuste.

Le premier terme correspond à la sortie du RNA (éq.I.11) et considéré comme l'estimation de la fonction couple non linéaire définie dans (éq.I.34). Le second correspond à l'équation classique d'un régulateur proportionnel dérivateur à gain fixe. Le terme robuste  $v_R(t)$  est introduit afin d'assurer plus de robustesse en présence d'erreur élevée due à l'estimation.

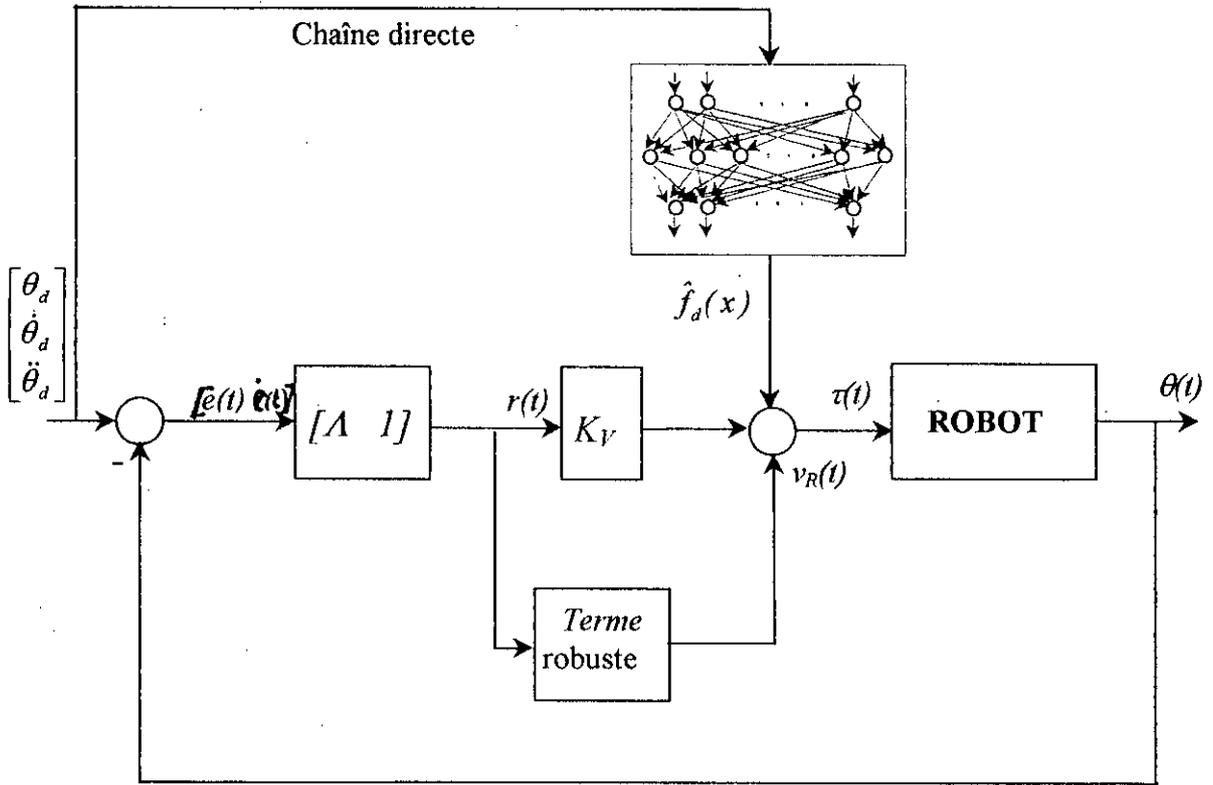


Fig. I.5 structure de base du système de commande prédictif adaptatif neuronal.

**Etude de la dynamique de l'erreur filtrée**

En introduisant ce contrôleur dans l'équation (I.6), la dynamique de l'erreur filtrée devient

$$M(\theta) \cdot \dot{r} = -(K_v + V_m(\theta, \dot{\theta})) \cdot r + f(x) - \hat{f}_d(x) + \tau_d - v_R(t) + f_d(.) - \hat{f}_d(.)$$

En utilisant (I.11) et (I.33), on obtient

$$M(\theta) \cdot \dot{r} = -(K_v + V_m(\theta, \dot{\theta})) \cdot r + W^T \cdot \sigma(V^T \cdot x) - \hat{W}^T \cdot \sigma(\hat{V}^T \cdot x) + (f(.) - \hat{f}_d(.)) + (\varepsilon + \tau_d) - v(t)$$

d'où

$$M(\theta) \cdot \dot{r} = -(K_v + V_m(\theta, \dot{\theta})) \cdot r + \tilde{W}^T \cdot \hat{\sigma} + \hat{W}^T \cdot \tilde{\sigma} + \tilde{W}^T \cdot \tilde{\sigma} + (f(.) - \hat{f}_d(.)) + (\varepsilon + \tau_d) - v_R(t) \quad (I.36)$$

avec  $\tilde{W}, \tilde{V}$  et  $\tilde{\sigma}$  sont définis par (I.13) et (I.14), et d'après (I.16), nous avons

$$M(\theta) \cdot \dot{r} = -(K_v + V_m(\theta, \dot{\theta})) \cdot r + \tilde{W}^T \cdot \hat{\sigma} + \hat{W}^T \cdot \hat{\sigma}' \cdot \tilde{V}^T \cdot x + w(t) - v_R(t) \quad (I.37)$$

avec:

$$w(t) = \tilde{W}^T \cdot \hat{\sigma}' \cdot \tilde{V}^T \cdot x + W^T \cdot O(\tilde{V}^T \cdot x)^2 + (f(\cdot) - f_d(\cdot)) + (\varepsilon + \tau_d) \quad (I.38)$$

Le terme  $w(t)$  regroupe les perturbations ainsi que les erreurs d'estimation infiniment petites.

**REMARQUE :** Les lois d'adaptation, ainsi que l'étude de la stabilité sont les mêmes que celles de la méthode précédente dans le cas générale.

### **Etude par Simulation**

Nous avons opté pour trois réseaux de neurones MISO (Multi Input Simple Output) comportant chacun : 9 entrées, 5 neurones dans la couche cachée, tous les poids des réseaux sont initialisés à zéro.

Les résultats de simulations, pour la première trajectoire de la commande sans et avec variation paramétrique sont montrés à la figure (I.6). Afin de tester le schéma de commande pour des trajectoires très rapides, le test de Leahy a été effectué et les résultats obtenus sans et avec perturbation sont présentés sur la figure (I.7).

Notons, que les résultats sont presque identiques à ceux de la méthode précédente, cela confirme l'efficacité de la commande prédictive par RNA. Par ailleurs, nous avons réduit la dimension du vecteur d'entrée de 15 à 9.

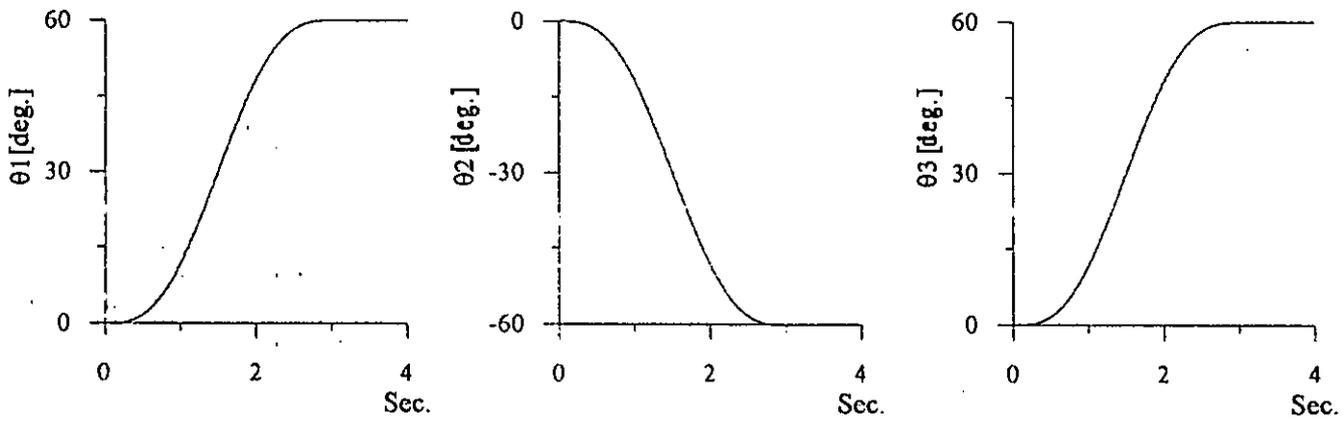


Figure 1.6a Réponses en position des art. 1, 2 et 3

— à vide. - - - avec perturbation.

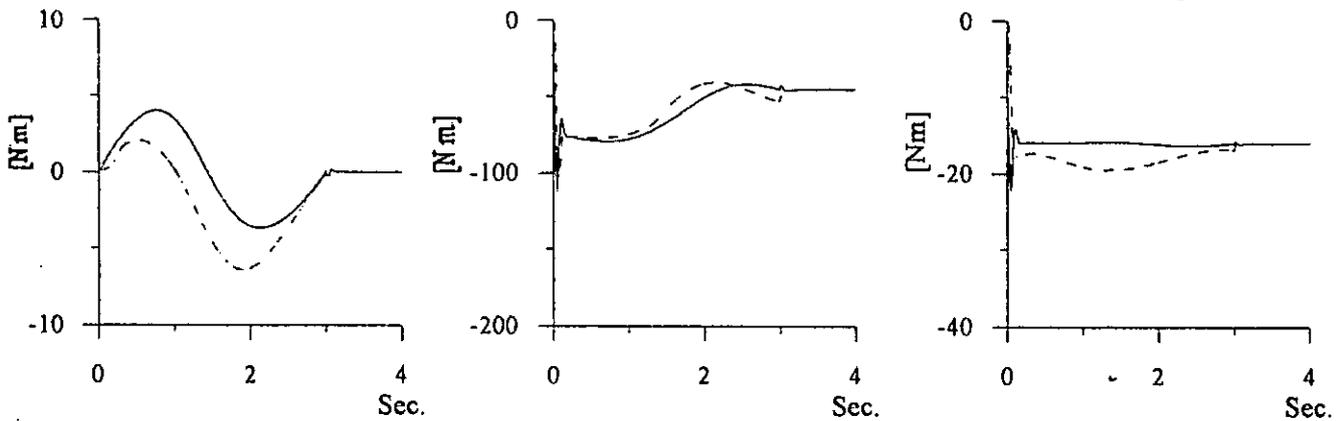


Figure 1.6b Commandes à vide des art. 1, 2 et 3

—  $\tau$ .

- - - Sortie du RNA.

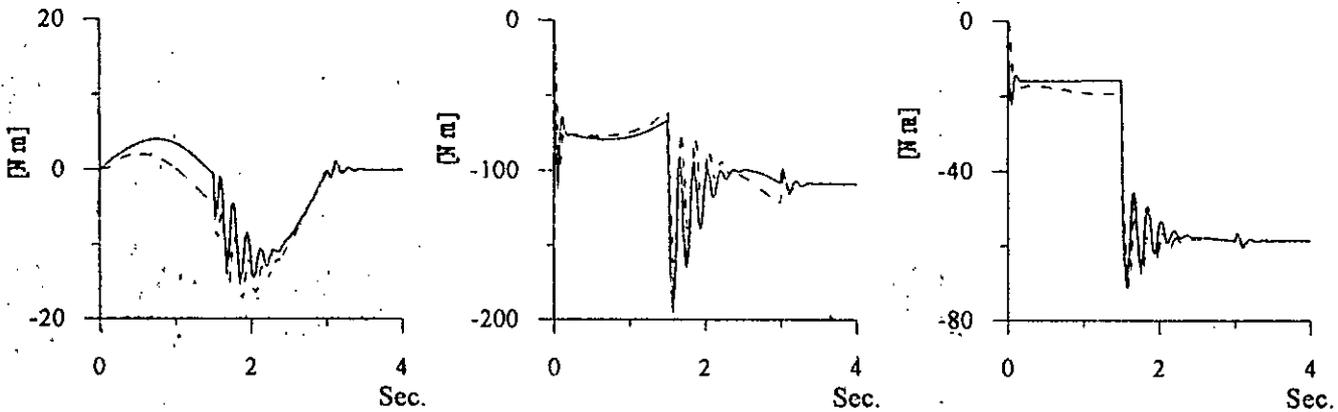


Figure 1.6c Commandes avec perturbation des art. 1, 2 et 3

—  $\tau$

- - - Sortie du RNA.

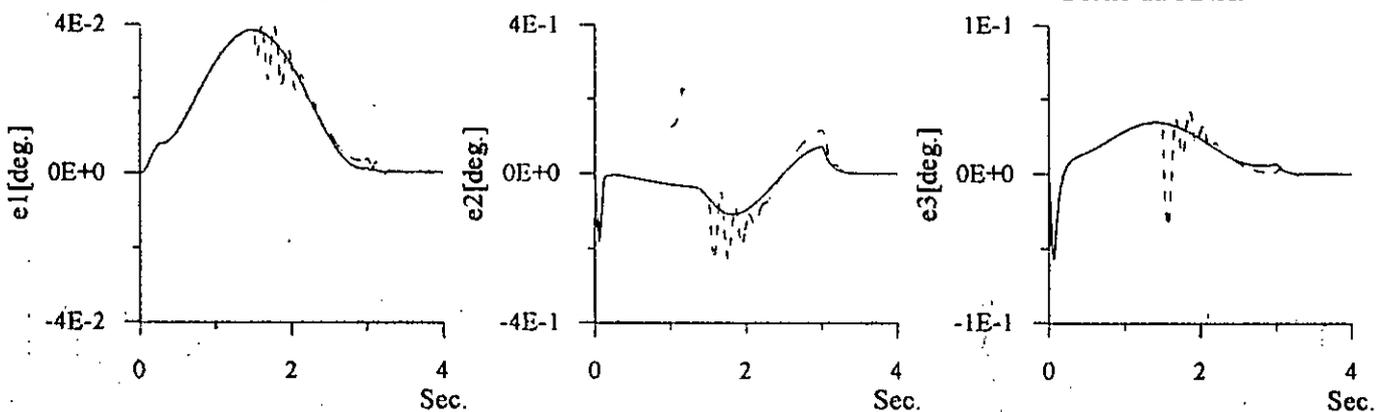
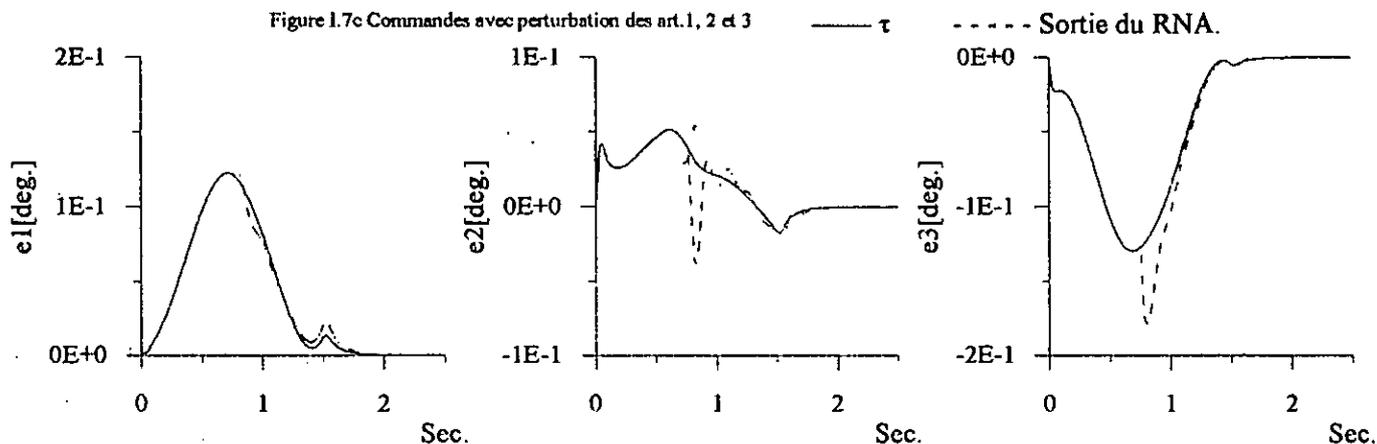
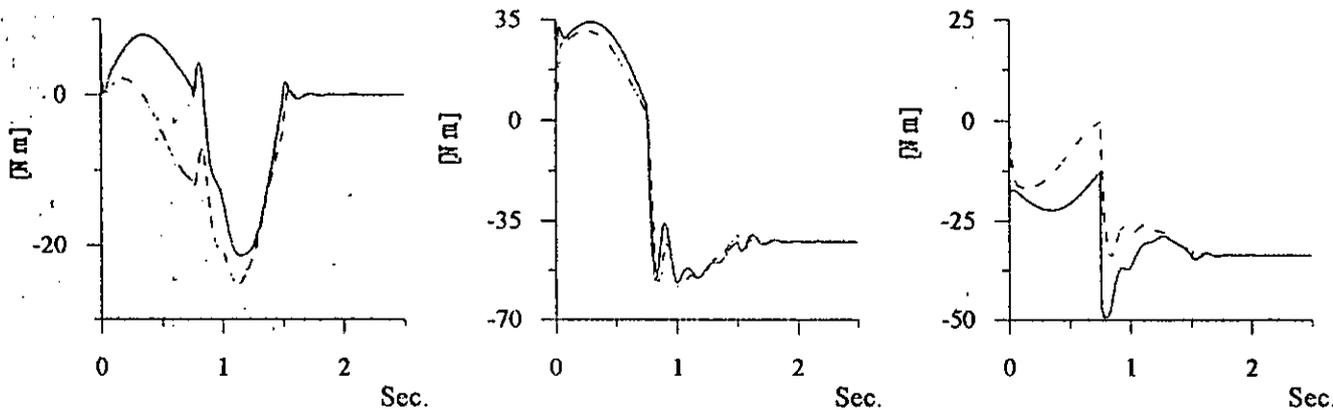
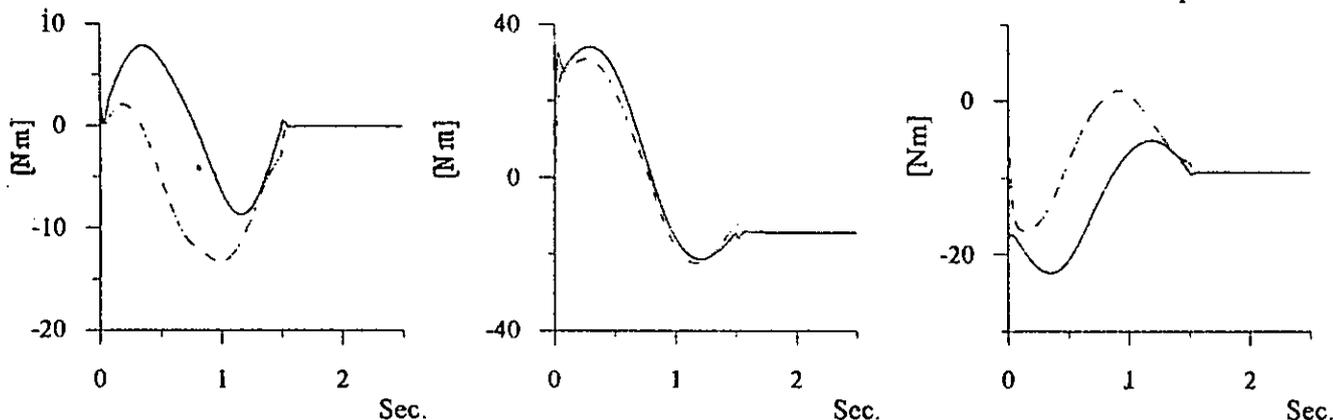
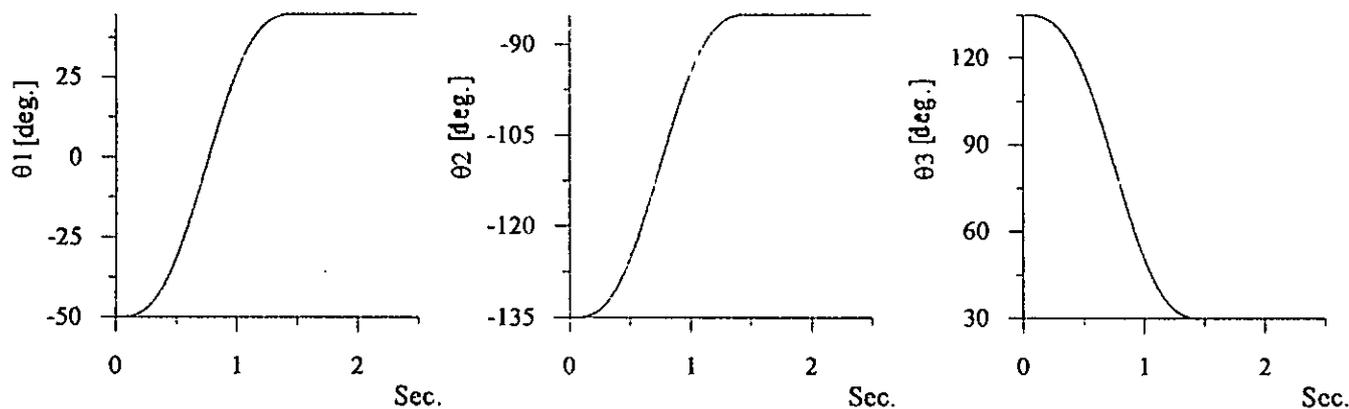


Figure 1.6d Erreurs des art. 1, 2 et 3

— à vide.

- - - avec perturbation.



## 1.4 Commande Adaptative Centralisée Directe par RNA à Fonction de Base Radiale Gaussienne (RFBG)

### 1.4.1 Fonction de Base Radiale à Centres Fixes

Les fonctions de base radiales ont leurs origines dans la théorie de l'approximation où elles sont utilisées pour produire des approximations aux fonctions inconnues. Basée sur un ensemble de données entrées/sorties représentant la fonction inconnue, les fonctions de base radiales sont capables de produire des approximations à une fonction inconnue pour un ensemble de données. L'approximation est produite en traversant chaque point de l'entrée par un ensemble de fonctions de base, chacune contient un des centres RFBG, multiplier le résultat de chaque fonction gaussienne par un coefficient, et enfin les additionnant linéairement pour calculer la sortie du réseau. Dans la littérature deux types de réseaux gaussiens sont développés, ceux à centres fixes, qui fera l'objet de cette section, et ceux à centres adaptatifs qui seront présentés par la suite. Pour illustrer l'approche une application à un pendule a été effectuée.

#### 1.4.1. a) Cas SISO (Single Input Single Output) : Appliquée à un Pendule

##### Position du Problème

Considérons la forme générale suivante caractérisant un système non linéaire monovarié.

$$x^{(n)}(t) + f(x(t), \dot{x}(t), \dots, x^{(n-1)}(t)) = b(x)u(t) \quad (1.39)$$

$u(t)$  désigne le signal de commande.

$f(\cdot)$ ,  $b(\cdot)$  sont deux fonctions pouvant être linéaires ou non linéaires de  $x(t)$ .

Dans ce qui suit, nous allons présenter le schéma de commande dans le cas particulier où  $b(x)=1$ .

L'objectif de la commande est de forcer le vecteur d'état du système  $X = [x, \dot{x}, \dots, x^{(n-1)}]^T$  à suivre une trajectoire désirée  $X_d = [x_d, \dot{x}_d, \dots, x_d^{(n-1)}]^T$ .

On définit le vecteur de l'erreur de poursuite  $\tilde{X}(t) = X(t) - X_d(t)$ .

Le problème est de synthétiser une loi de commande  $u(t)$  telle que  $\lim_{t \rightarrow \infty} \tilde{X}(t) \rightarrow 0$

Cette loi aura la forme

$$u(t) = x_d^{(n)}(t) + u_{pd}(t) + u_{ad}(t) \quad (1.40)$$

$u_{pd}$  est la sortie d'un régulateur linéaire de type PD et  $u_{ad}$  est une loi de commande adaptative.

En substituant (1.40) dans (1.39), on obtient

$$\tilde{x}^{(n)} = -k^T \tilde{x}(t) + (u_{ad}(t) - f(\cdot)) \quad (1.41)$$

$k^T$  est un vecteur des gains.

La fonction inconnue  $f(x)$  est liée aux fonctions de base "Basis Functions"  $Y_i(x)$  [OPP 89] par la relation:

$$f(x) = \sum_{i=1}^N a_i Y_i(x) \quad (I.42)$$

de l'équation (I.41), il sera donc judicieux de choisir un contrôleur adaptatif possédant la même structure de l'équation (I.42). Ceci peut être fait par le choix suivant :

$$u_{ad}(t) = \sum_{i=1}^N \hat{a}_i(t) Y_i(x) \quad (I.43)$$

avec  $\hat{a}_i(t)$  est une estimation du  $i^{\text{ème}}$  coefficient de l'expression de  $f(\cdot)$  d'où:

$$u_{ad}(t) - f(x) = \sum_{i=1}^N \tilde{a}_i(t) Y_i(x) \quad (I.44)$$

$\tilde{a}_i(t) = \hat{a}_i(t) - a_i$  désigne l'erreur paramétrique.

La poursuite est assurée si la loi d'ajustement des coefficients  $\hat{a}_i(t)$  garantit la bornitude des erreurs paramétriques et de poursuite.

En commande adaptative la condition de corrélation des erreurs de poursuites est donnée par [SLO 91]

$$r(t) = \lambda^T \tilde{x}(t) \quad (I.45)$$

Où  $\lambda$  est tel que la fonction de transfert  $\lambda^T (PI - A)^{-1} b$  soit strictement réelle positive.

Pour satisfaire cette condition les vecteurs constants  $k$  et  $\lambda$  doivent être choisis tel que l'évolution de  $r(t)$  soit régie par l'équation différentielle de la forme

$$\dot{r}(t) = -k_D r(t) + \sum_{i=1}^N \tilde{a}_i(t) Y_i(x(t)) \quad (I.46)$$

$k_D$  est une constante positive non nulle.

### Approximation par RBFN

Etant donnée une fonction  $f(x)$  à approximer par un réseau de neurones à couche. Cette approximation est représentée mathématiquement par la relation

$$\hat{f}(x) = \sum_{i=1}^N c_i g_i(x, \xi_i) \quad (I.47)$$

où  $g_i$  représente la fonction élémentaire de nœud  $i$ ,  $\xi_i$  représente le poids entrant au nœud  $i$  et  $c_i$  représente le poids sortant du nœud  $i$ .

En utilisant la théorie de l'échantillonnage et l'analyse du Fourier, dans [OPP 89] on a montré l'existence d'une large classe de représentation pouvant approximer uniformément des fonctions lisses dans des ensembles compacts.

Les fonctions RBFN sont des fonctions bornées, strictement positives et absolument intégrables dans  $\mathbb{R}^n$  et possèdent leurs propres transformées de Fourier suivantes

$$\exp\left(\frac{-\pi v^T v}{\sigma_v^2}\right) \Leftrightarrow \sigma_v^n \exp(-\pi \sigma_v^2 x^T x) \quad (I.48)$$

Grâce à leurs propriétés, ces fonctions sont utilisées dans la théorie de l'échantillonnage.

Soit  $f_A$  la reconstruction de la fonction  $f$  par un nombre fini  $I_0$  de RFBG dans un ensemble compact  $A$

$$f_A(x) = \sum_{I \in I_0} c_I g_\sigma(x - \xi_I) \quad (I.49)$$

En utilisant le filtre de reconstruction gaussien (I.48)  $g_\sigma$  sera donnée par :

$$g_\sigma(x - \xi) = \exp(-\pi \sigma_v^2 \|x - \xi\|^2) \quad (I.50)$$

ou  $\xi$  désigne le centre de la fonction RFBG et  $\sigma_v^2$  sa variance.

Dans le réseau de neurones utilisé, le nombre de nœuds est défini à partir de l'ensemble  $A$  et le rang d'une maille  $\Delta$  de cet ensemble. Les paramètres des RFBG sont choisis afin d'assurer la bornitude de l'erreur de poursuite [SLO 91]

### Commande Adaptative par Réseaux Gaussiens

Comme il a déjà été mentionné, le réseau Gaussien est utilisé comme composante adaptative de la loi de commande (I.40).

L'équation d'erreur (I.46) devient donc

$$\begin{aligned} \dot{r}(t) &= -K_D r(t) - f(x) + \sum_{I \in I_0} \hat{c}_I g_\sigma(x(t) - \xi_I) \\ &= -K_D r(t) + \sum_{I \in I_0} \tilde{c}_I g_\sigma(x(t) - \xi_I) + d(t) \end{aligned} \quad (I.51)$$

Avec  $\hat{c}_I(t)$  est une estimation de  $c_I$  à l'instant  $t$ . Le terme  $d(t)$  représente l'erreur entre la fonction réelle et l'approximation donnée par le réseau.

La présence de ce terme de perturbation change fondamentalement les propriétés de la stabilité de l'algorithme de commande adaptative classique. Également, si on peut s'assurer a priori que l'état  $x$  évolue à l'intérieur de l'hyperplan  $A_d$  contenant la trajectoire désirée durant l'opération de poursuite, la perturbation sera uniformément bornée par une petite constante  $\varepsilon_f$ .

Pour surmonter le problème d'instabilité on utilise une zone morte de largeur  $\Phi$  dans l'algorithme d'adaptation [SLO 91][SAN 92].

Afin de forcer le système à évoluer dans  $A_d$ , nous avons introduit dans la loi adaptative une composante non adaptative et non linéaire (mode de glissement). Cette composante est donnée par la forme générale

$$u_{sl}(t) = -k_{sl} \operatorname{sat}\left(\frac{r(t)}{\Psi}\right) \quad (I.52)$$

$\Psi$  est la largeur de la zone bornée de transition à la commande discontinue et  $k_{sl}$  désigne le gain de la commande par mode de glissement. La commande par mode de glissement opère à l'extérieur de  $A$  et la commande adaptative opère à l'intérieur de  $A_d$ . Entre les deux régions  $A_d$  et  $A$  les deux modes sont actives, ceci est réalisé en utilisant une fonction de modulation continue  $m(t)$  afin de déterminer le niveau de contribution de chaque composante dans la loi de commande complète.

### Architecture Générale du Contrôleur

La loi de commande utilisée (voir figure 1.7)

$$u(t) = u_{pd}(t) + [1 - m(t)]u_{ad}(t) + m(t)u_{sl}(t) \quad (1.53)$$

Les ensembles  $A_d$  et  $A$  sont définis par :

$$A_d = \{ x / \|x - x_0\|_{p,w} \leq 1 \} \quad (1.54)$$

$$A = \{ x / \|x - x_0\|_{p,w} = 1 + \Psi \} \quad (1.55)$$

$\Psi$  est une constante positive représentant la largeur de la zone de transition entre les deux modes de commande,  $x_0$  fixe la location absolue de l'ensemble dans l'espace d'état du système et  $\|x\|_{p,w}$  est la norme p des poids

$$\|x\|_{p,w} = \left( \sum_{i=1}^n \left( \frac{|x_i|}{w_i} \right)^p \right)^{1/p} \quad (1.56)$$

Pour un ensemble de poids positifs  $\{ w_i, i=1, \dots, n \}$ . Pour  $p=\infty$  l'équation (1.56) devient

$$\|x\|_{\infty,w} = \max \left( \frac{|x_1|}{w_1}, \frac{|x_2|}{w_2}, \dots, \frac{|x_n|}{w_n} \right) \quad (1.57)$$

La fonction de modulation est donnée par

$$m(t) = \max \left[ 0, \text{sat} \left( \frac{r_0(t) - 1}{\Psi} \right) \right] \quad (1.58)$$

$r_0(t)$  est calculé par  $r_0(t) = \|x(t) - x_0\|_{p,w}$ .

La composante adaptative  $u_{ad}(t)$  estimant  $f_A(t, x(t))$  est réalisée par la sortie du RBFG est donc

$$u_{ad} = \hat{f}_A(t, x(t)) = \sum_{i \in I_0} \hat{c}_i g(x(t) - \xi_i) \quad (1.59)$$

Les poids  $\xi_i$  sont fixes et ceux de sortie  $c_i(t)$  sont ajustés afin d'atteindre les performances désirées.

L'erreur de poursuite sera définie par [SLO 91][SAN 92]

$$r(t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x}(t) \tag{1.60}$$

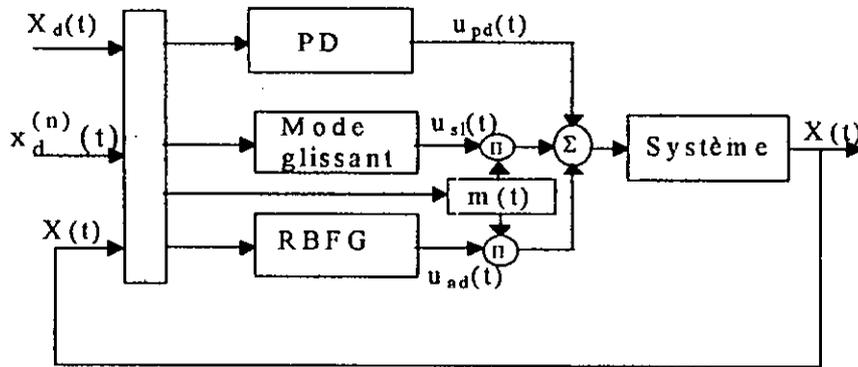


Fig. 1.8 Schéma de Commande.

avec  $\lambda > 0$ .  
d'où on obtient :

$$r(t) = \lambda^T \tilde{x}(t) \text{ avec } \lambda^T = [\lambda^{n-1}, (n-1)\lambda^{n-2}, \dots, 1] \tag{1.61}$$

$r(t)=0$  définit une surface variable dans le temps, dans laquelle le vecteur de l'erreur de poursuite tend exponentiellement vers zéro. Si  $r(t)$  est bornée par la constante  $\Psi$ , l'erreur de poursuite sera asymptotiquement bornée suivant l'équation :

$$|\tilde{x}^{(i)}(t)| \leq 2^i \lambda^{i-n+1} \Psi, \quad i = 0, \dots, n-1 \tag{1.62}$$

La zone morte de largeur  $\Phi$  introduite dans la mesure de l'erreur donne l'équation

$$r_\Delta(t) = r(t) - \Phi \text{ sat}(r(t)/\Phi) \tag{1.63}$$

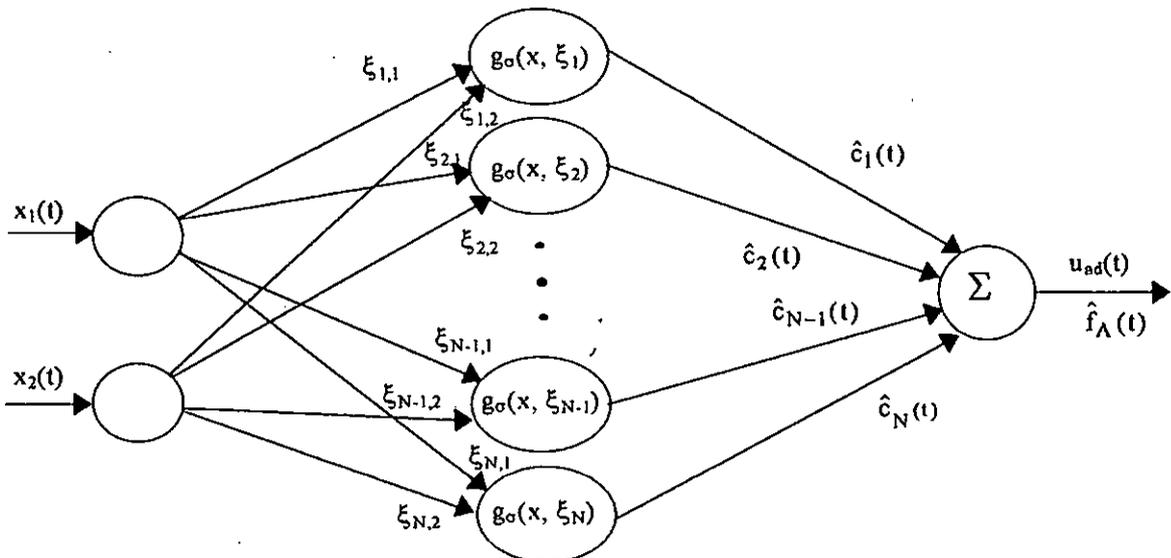


Fig. 1.9 Le Réseau Gaussien estimant la fonction non linéaire  $f(x)$ .

### Synthèse du contrôleur et Analyse de la Stabilité

Dans cette section, nous allons présenter la loi de commande pour les systèmes ayant la forme (I.39). En supposant connaître à priori la borne supérieure de  $f$  pour les points situés à l'extérieur de l'ensemble  $A_d$ , i.e

$$|f(x)| \leq M_0(x) \quad \text{quand } x \in A_d^c \quad (I.64)$$

nous avons choisi  $A \in A_d$ , dans lequel  $f_A(x)$  est l'approximation gaussienne de  $f(x)$ . La dérivée de  $r(t)$  (éq. (I.60)) par rapport au temps

$$\dot{r}(t) = a_r(t) - f_A(x(t)) + u(t) + d(t) \quad (I.65)$$

avec:

$$a_r(t) = \lambda_v^T \tilde{x}(t) - x_d^{(n)}(t), \quad \lambda_v^T = [0, \lambda^{n-1}, 2\lambda^{n-2}, \dots, (n-1)\lambda]$$

et la perturbation  $d(t) = f_A(x(t)) - f(x(t))$  est bornée  $\forall x(t) \in A$ .

En utilisant cette expression et les développements des sections précédentes, nous obtenons la loi de commande suivante:

$$u(t) = -K_D r(t) - a_r(t) + (1 - m(t)) \hat{f}_A(t, x(t)) + m(t) u_{st}(t) \quad (I.66)$$

$K_D > 0$  et  $k_{st} = M_0(x(t)) - \varepsilon_f$ . La figure (I.8) illustre le réseau gaussien implémentant la composante adaptative de la loi de commande.

En utilisant cette loi de commande et l'équation de  $\dot{r}(t)$ , on peut écrire

$$\begin{aligned} \dot{r}(t) &= -K_D r(t) + (1 - m(t)) (\tilde{f}_A + d(t)) + m(t) (u_{st}(t) - f(x(t))) \\ \text{et} \quad \tilde{f}_A(t) &= \hat{f}_A(t, x(t)) - f_A(x(t)) \end{aligned} \quad (I.67)$$

L'ajustement des poids de sorties du réseau est effectué, en utilisant

$$\frac{d\hat{c}(t)}{dt} = -k_a (1 - m(t)) r_\Delta(t) g_\sigma(x(t) - \xi_1) \quad (I.68)$$

$k_a > 0$ , désigne le gain d'adaptation.

si  $\Phi$  est choisie tel que  $\phi \geq \varepsilon_f / K_D$ , l'erreur de poursuite converge asymptotiquement vers une zone bornée proche de zéro [SAN 92]. Pour cela, on considère la fonction de Lyapunov

$$v(t) = \frac{1}{2} (r_\Delta^2(t) + \frac{1}{k_a} \sum_{i \in I_\sigma} \tilde{c}_i(t)^2) \quad (I.69)$$

$$\frac{dr_\Delta^2}{dt} = 2r_\Delta \dot{r} \quad \text{si } |r| = \Phi.$$

En utilisant (I.69)  $\dot{v} = 0$  si  $|r| \leq \Phi$ .

D'une manière général puisque  $r_A \text{sat}(\frac{r}{\Phi}) = |r_A|$ , en utilisant (I.51) et (I.55), on obtient:

$$\dot{v} = -(K_D r_A^2 + |r_A| K_D \Phi) + r_A (1-m) (\tilde{f}_A + d + k_a^{-1} \sum_{j \in I_0} \tilde{c}_j \frac{d\tilde{c}_j}{dt} - m(r_A f + k_{st} |r_A|)) \quad (I.70)$$

en substituant (I.69) et utilisant les bornes des fonctions discutées, l'équation (I.70) devient

$$\dot{v} \leq -K_D r_A^2 + (1-m) |r_A| (|d| - K_D \Phi) + m |r_A| (|f| - K_D \Phi - k_{st}) \quad (I.71)$$

Le deuxième terme de la partie droite est non positif par construction du réseau [SAN 92], le troisième est aussi non positif par le choix de gain  $K_{st}$ . Ainsi, on obtient

$$\dot{v} \leq -K_D r_A^2 \quad \forall t \geq 0 \quad (I.72)$$

D'où si  $r_A$  et  $c_j$  sont bornés à l'instant  $t = 0$ , ils le seront  $\forall t > 0$ . On démontre que  $r_A \rightarrow 0$  quand  $t \rightarrow \infty$  [SLO 91] [SAN 92], donc  $\dot{v} \rightarrow 0$  quand  $t \rightarrow \infty$ , cependant  $|r(t)| \leq \phi$  est obtenue asymptotiquement.

A partir de l'équation (I.62), on déduit que l'erreur de poursuite est asymptotiquement bornée.

### Etude par Simulation

Le modèle non linéaire du pendule est donné [CHA 95] sous la forme (I.39) avec  $n=2$  et  $f$  donnée par

$$f(x, \dot{x}) = \dot{x} + \sin(x)$$

On se basant sur la trajectoire désirée, l'ensemble  $A_d$  est choisi en utilisant la norme infinie :

$$\|x\|_{\infty, w} = \max\left(\frac{|x|}{1.13}, \frac{|\dot{x}|}{0.11}\right)$$

centré à  $x_0 = [0, 0]^T$ , ceci correspond à un sous-ensemble rectangulaire de l'espace d'état  $A_d$  ( $A_d = [-1.13, 1.13] * [-0.11, 0.11]$ ).  $\Psi = 0.005$  et  $K_D = \lambda = 0.3$ , en utilisant (I.62), on obtient  $\varepsilon_f \leq 0.05$ .

La fonction  $f_A$  est supposée dans le cube  $[-2, 2]^2$  d'où  $\beta = 2$ , avec  $\theta = 2$  et  $l = 4$ . La variance des nœuds  $\sigma_v^2 = 4\pi$  et le rang de la grille  $\Delta = 0.015$ , donc on obtient un réseau constitué de 451 nœuds Gaussiens.  $k_{st} = 0.2$ ,  $k_a = 50$ ,  $\Phi = 0.0005$  et  $T = 0.03s$ .

Les résultats de simulation (Fig. I.9) montrent une bonne poursuite dès le premier passage, ce qui montre l'efficacité de la technique de commande présentée par rapport à celle basée sur la rétropropagation de l'erreur.

La méthode de commande présentée dans la présente section est basée sur les réseaux de neurones Gaussiens. D'après les résultats obtenus, l'utilisation de ce type de réseaux donne de meilleures performances par rapport aux réseaux de neurones classiques, basés sur la technique de la rétropropagation de l'erreur.

Les résultats de l'application au pendule, montrent l'efficacité de la stratégie de commande adaptative directe par RBFN en termes de poursuite de trajectoires de références pour les systèmes non linéaires.

Nous avons choisi  $A \in A_d$ , dans lequel  $f_A(x)$  est l'approximation gaussienne de  $f(x)$ . De l'équation (I.6) la dynamique de l'erreur filtrée  $r(t)$ , définie par (I.61) avec  $n=2$ , est donnée par

$$M(\theta) \cdot \dot{r} = -V_m(\theta, \dot{\theta}) \cdot r - \tau(t) + f_A(x) + d(t) + \tau_d(t) \quad (I.75)$$

la perturbation  $d(t) = f_A(x(t)) - f(x(t))$  est bornée  $\forall x(t) \in A$ .

En utilisant les développements des sections précédentes, nous obtenons la loi de commande suivante :

$$u(t) = K_D r(t) + (1 - m(t)) (\hat{f}_A(t, x(t)) + v_R(t)) + m(t) u_{st}(t) \quad (I.76)$$

$K_D > 0$  et  $k_{st} = M_o(x(t)) - \varepsilon_f$ . La figure (I.8) illustre le réseau gaussien implémentant la composante adaptative de la loi de commande.

En utilisant cette loi de commande et l'équation (I.75), on peut écrire

$$M(\theta) \cdot \dot{r} = -(K_D + V_m(\theta, \dot{\theta})) \cdot r + (1 - m(t)) (\tilde{f}_A + d(t) + \tau_d + v_R(t)) + m(t) (u_{st}(t) - f(x(t)) + \tau_d)$$

et  $\tilde{f}_A(t) = \hat{f}_A(t, x(t)) - f_A(x(t))$  (I.77)

L'ajustement des poids de sortie du réseau est effectué, en utilisant

$$\dot{\hat{c}}(t) = -k_a (1 - m(t)) r(t) g_\sigma(x(t) - \xi_1) \quad (I.78)$$

$k_a > 0$ , désigne le gain d'adaptation.

### Analyse de la stabilité

Soit la fonction positive de Lyapunov suivante :

$$\dot{L}(t) = \frac{1}{2} r^T M \dot{r} + \frac{1}{2} e^T K_p e + \frac{1}{2} \text{tr}(\tilde{C}^T K_a^{-1} \tilde{C}) + L_R \quad (I.79)$$

$$L_R = \frac{1}{2} \tilde{\varphi} \Gamma^{-1} \tilde{\varphi} + \frac{\delta}{\gamma}$$

Dérivons (I.79) et utilisons (I.77), nous obtenons

$$\dot{L} \leq -z^T Q z + \dot{L}_1 + \dot{L}_2 + \dot{L}_3$$

avec

$$z = \begin{bmatrix} e^T & r^T \end{bmatrix}^T,$$

$$Q = \begin{bmatrix} K_p \Lambda & -\frac{K_p}{2} \\ -\frac{K_p}{2} & K_v \end{bmatrix} > 0 \text{ pour un choix approprié des matrices des gains } K_p \text{ et } K_v,$$

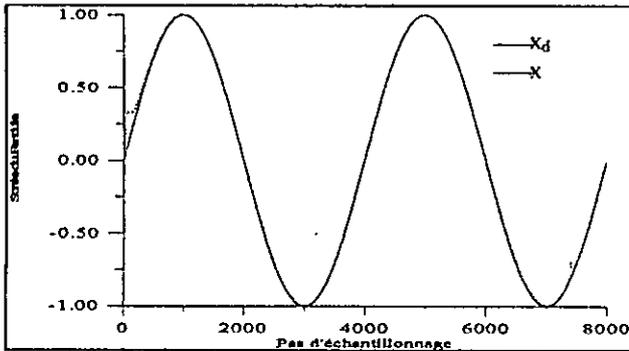


Fig. 1.10a Sortie  $x$  et Sortie Désirée  $x_d$  du Pendule.

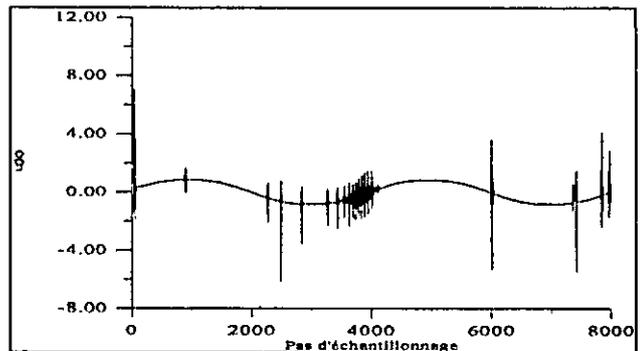


Fig. 1.10c La commande  $u(k)$ .

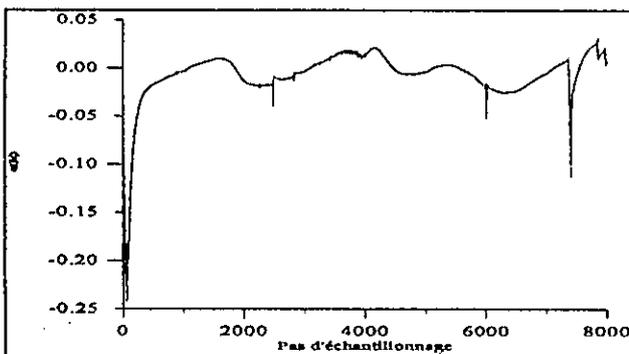


Fig. 1.10b Erreur de poursuite  $e(k)$ .

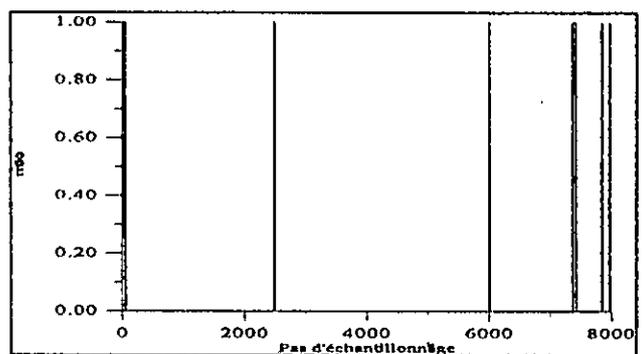


Fig. 1.10d La fonction de modulation  $m(k)$ .

### 1.4.1. b) cas MIMO (Multi Input Multi Output)

En se basant sur la technique de commande présentée dans la section (I.2), la commande par réseaux gaussiens à centres fixes SISO sera étendue, au cas MIMO, pour la commande en poursuite des robots manipulateurs.

La fonction couple non linéaire du robot  $f(x)$  donnée par l'équation (I.7) est liée aux fonctions de base radiale par la relation (I.51). Donc, cette fonction sera estimée par une fonction possédant la même structure donnée par l'équation (I.52). d'où :

$$u_{ad}(t) - f(x) = \sum_{i=1}^N \tilde{a}_i(t) Y_i(x), \quad x = \begin{bmatrix} \theta^T & \dot{\theta}^T & \ddot{\theta}_d^T \end{bmatrix}^T \quad (I.73)$$

$\tilde{a}_i(t) = \hat{a}_i(t) - a_i$  désigne l'erreur paramétrique.

### Synthèse du contrôleur

Dans cette section, nous présentons la loi de commande pour les robots manipulateurs. En supposant connaître a priori la borne supérieure de  $f(x)$  pour les points situés à l'extérieur de l'ensemble  $A_d$ , i.e

$$\|f(x)\| \leq M_o(x) \quad \text{quand } x \in A_d^c \quad (I.74)$$

$$\dot{L}_1 = \|r\| S\hat{\varphi} - r^T v_R - \delta, \quad (I.80a)$$

$$\dot{L}_2 = \text{tr}[\tilde{C}^T K_a^{-1} \tilde{C} + \tilde{C}(1-m(t))r(t)g_\sigma(x(t)-\xi_1)] \quad (I.80b)$$

$$\dot{L}_3 = m(t)r(t)[(M_o(x(t)) + \tau_d) + u_{st}(t)] \quad (I.80c)$$

utilisons (I.26) dans (I.80a), on obtient

$$\begin{aligned} \dot{L}_1 &= \|r\| S\hat{\varphi} - \frac{r^T r (S\hat{\varphi})^2}{(S\hat{\varphi})\|r\| + \delta} - \delta \\ &\leq \frac{\delta\|r\| S\hat{\varphi}}{(S\hat{\varphi})\|r\| + \delta} - \delta \\ &\leq 0. \end{aligned} \quad (I.81a)$$

substituant la loi d'adaptation des poids (I.78) dans (I.80b), il vient :

$$\dot{L}_2 = 0 \quad (I.81b)$$

en utilisant la commande  $u_{st}$  donnée par l'équation (I.52), on obtient :

$$\dot{L}_3 \leq 0 \quad (I.81c)$$

combinons les résultats ci dessus, on obtient

$$\dot{L} \leq -z^T Qz, \quad Q > 0 \quad (I.82)$$

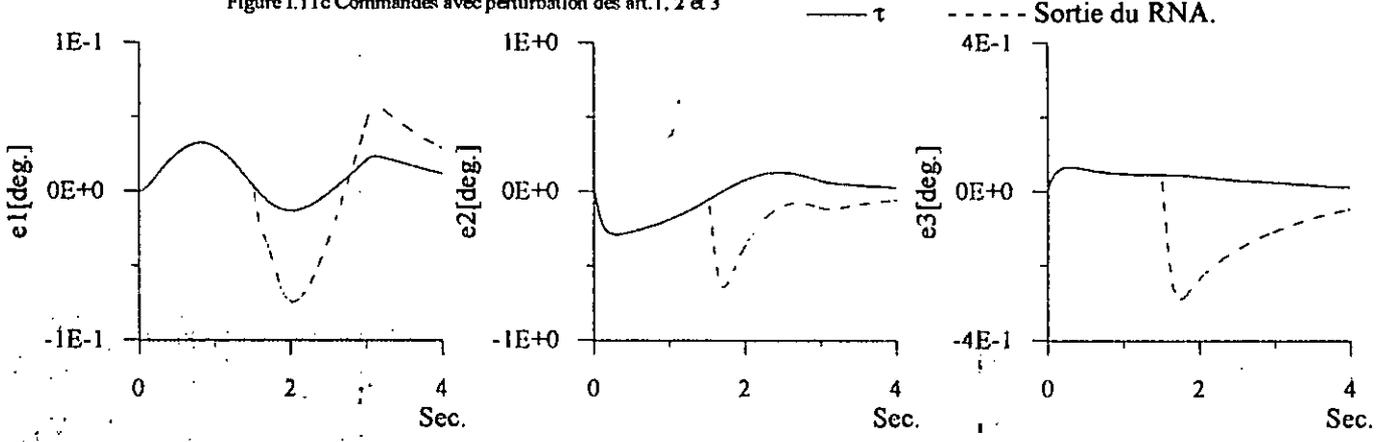
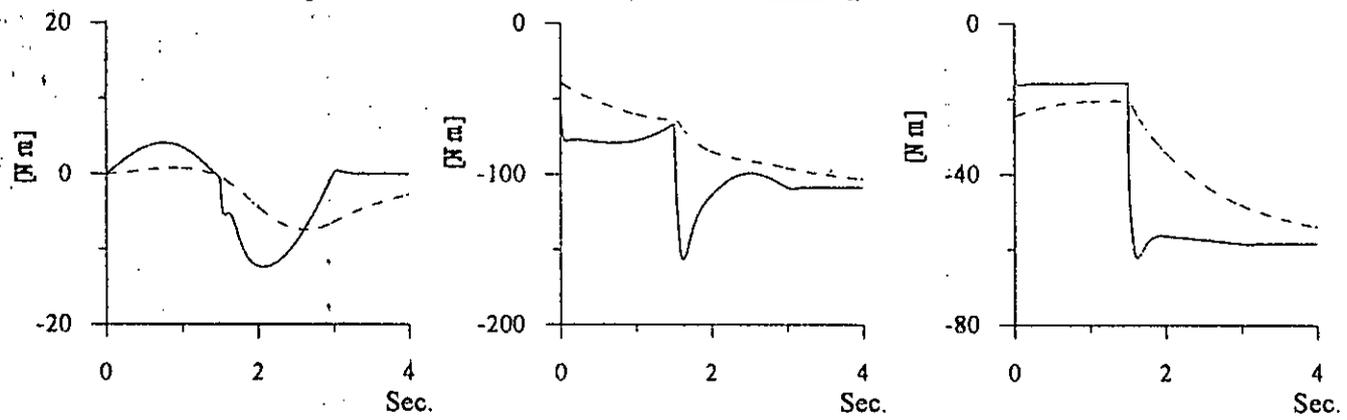
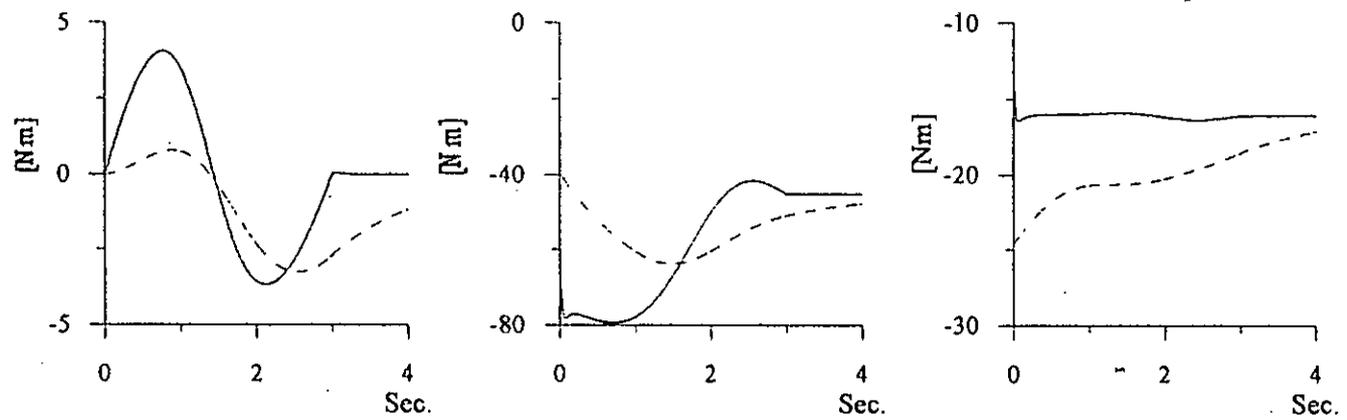
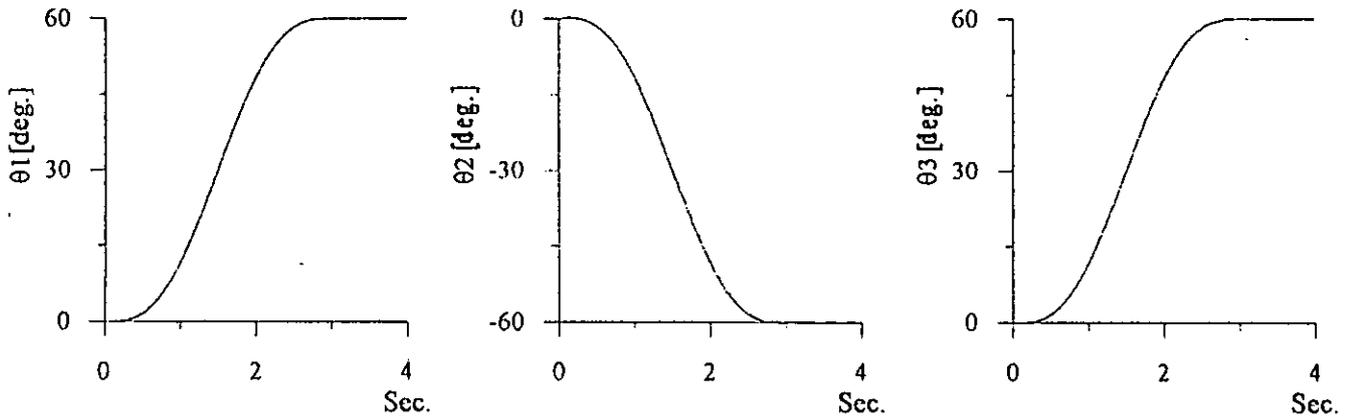
(I.82) implique  $L$ ,  $z$ ,  $e$ ,  $s$ ,  $\tilde{C}$ , et  $\hat{\varphi}$  sont tous bornés. A partir de l'équation (I.32), il est facile de montrer que le carré de  $z(t)$  est intégrable ; d'où les carrés de  $e$  et  $r$  sont intégrables. De l'expression de l'erreur en boucle fermée (I.77), il est aussi facile de montrer que  $\dot{r}$  est borné. Par conséquent,  $r$  est uniformément continu. Par la Lemme de Barbalat [KWA 95b], on démontre que  $e$  et  $\dot{e}$  tendent asymptotiquement vers zéro.

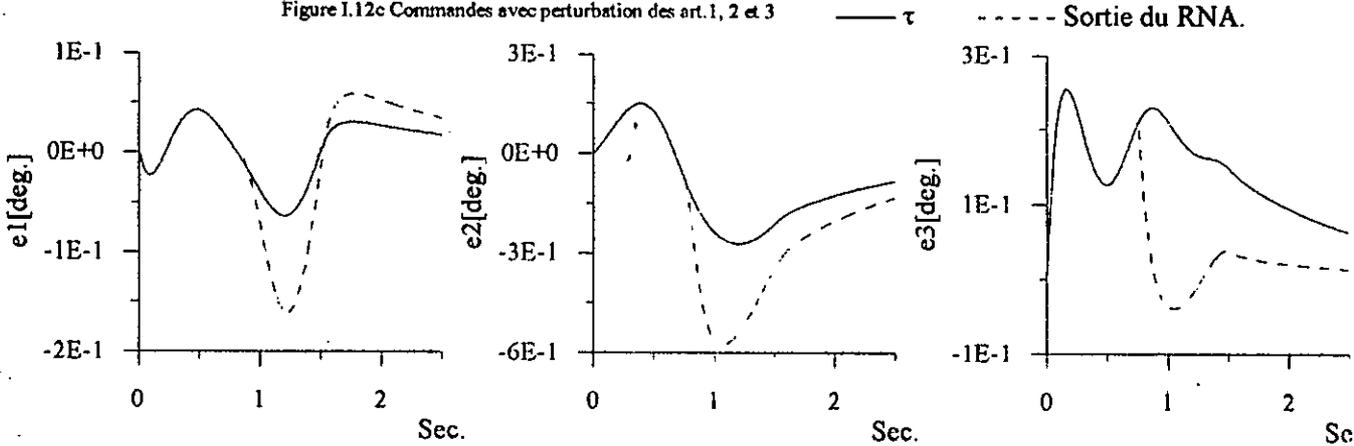
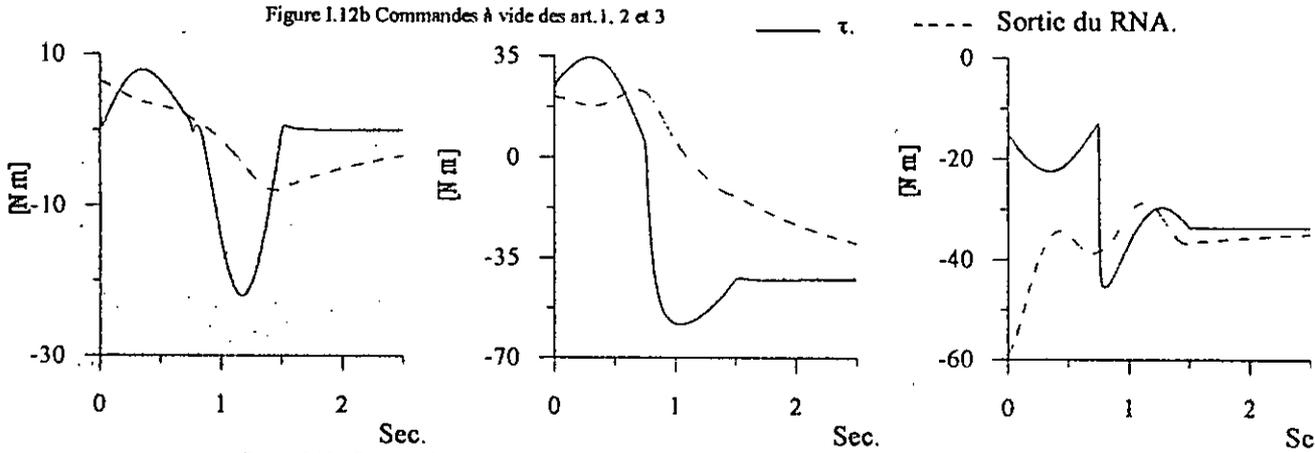
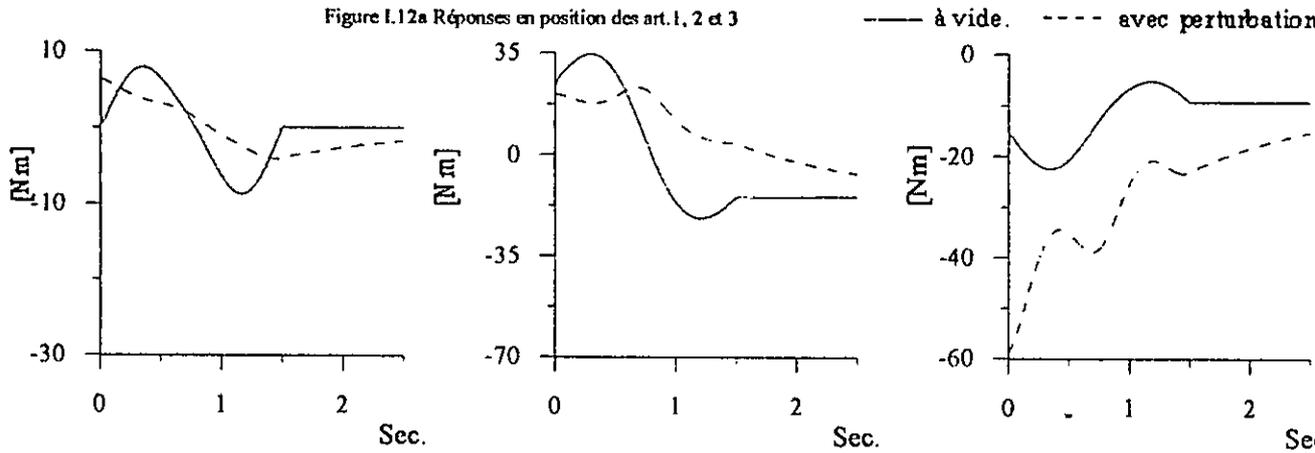
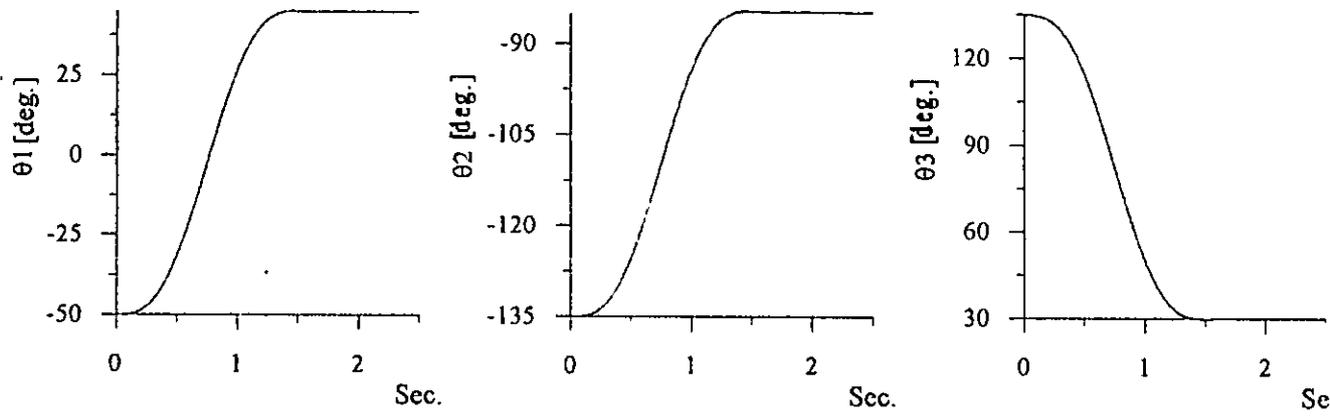
### Etude par Simulation

En se basant sur la trajectoire désirée, l'ensemble  $A_d$  est choisi en utilisant les bornes min et max de  $\theta_d$ ,  $\dot{\theta}_d$  et  $\ddot{\theta}_d$ .

Les ensembles  $A_{di}$  sont subdivisés à  $3^9 = 19683$  neurones (centres).  $\Psi=0.005$  et  $K_D=\lambda=600$  et  $K_p=6000$ . La variance des nœuds  $\sigma_v^2 = 1/20$  et le rang de la grille.  $k_{st} = 0.2$ ,  $k_a = 50$ .

Les résultats (figure I.11 et I.12) de l'application au robot PUMA 560 montrent l'efficacité de la stratégie de commande adaptative directe par RBFN en termes de poursuite de trajectoire de référence pour les robots manipulateurs, ainsi que la robustesse vis-à-vis des perturbations. Malgré ces résultats, l'utilisation de ce type de réseaux neuronaux reste limité car le nombre de neurone dans la couche cachée est très élevé ce qui rend le calcul lent et l'apprentissage difficile.





### 1.4.2 Fonction de Base Radiale à Centres Adaptatifs (Réseaux à Fonction de Base Radiale Généralisés)

Dans la section précédente nous avons utilisé des RBFG à centres fixes. Ces derniers sont disposés en treillis régulier pour couvrir uniformément la partie utile de l'espace d'entrée et respecter ainsi les contraintes de couverture et de généralisation localisée. Connaissant a priori certaines caractéristiques spatiales de la fonction à approcher (entre autres, sa 'fréquence spatiale de coupure'), il est possible de trouver une disposition des centres (en nombre suffisant, dépendant de la 'fréquence de coupure') garantissant que l'erreur d'approximation est inférieure à un seuil donné [SAN 92][BOU 97b].

En outre, la disposition des centres a priori s'avère quelque fois impraticable. En effet, le nombre de centres requis est parfois trop élevé. Donc, il est judicieux de laisser libre la disposition de ces centres ; dans ce cas, l'algorithme d'apprentissage devra ajuster, outre les poids de sortie ainsi que les positions des centres. Idéalement, le réseau s'auto-organisera pour placer un nombre élevé de centre là où la fonction est accidentée et espacer les centre dans les régions où la fonction est plus lisse. Ces réseaux sont appelés 'réseaux à fonctions de base radiales généralisés' [REN 95].

L'ajustement des centres et des poids des réseaux à base radiale généralisés peut être aisément effectué au moyen d'un algorithme de type 'décence de gradient'.

Pour chaque fonction inconnue  $f$ , l'approximation de cette fonction est entreposée essentiellement dans les coefficients et les centres du RBFG. Cependant, ces paramètres n'ont pas un chemin unique, depuis pour chaque fonction approximer  $f$ , beaucoup de combinaison des valeurs des paramètres existes. Les fonctions de base radiales RBFG possèdent la représentation mathématique suivante :

$$f(x) = c_0 + \sum_{i=1}^N c_i g(\|x - R_i\|) \quad (1.83)$$

Les vecteurs  $R_i$  ( $i=1, \dots, N$ ) sont appelés 'centres des Gaussiennes' (ils ont la même dimension que le vecteur d'entrée  $x$ ).  $c_i$  sont les poids de sortie du réseau,  $N$  le nombre de neurones dans la couche cachée. La norme utilisée, norme euclidienne, est définie dans la section précédente ainsi que la fonction gaussienne  $g(x)$ .

Le réseau Gaussien a pour rôle d'estimer la fonction couple non linéaire donnée par l'équation (I.7). Nous utilisons les mêmes étapes dans la synthèse du régulateur ainsi que dans l'étude de la stabilité de la section (I.2), remplaçons les poids d'entrée du RNA statique par les centres, du réseau RBFG.

La commande est donnée par l'équation suivante :

$$\tau(t) = \hat{f}(x) + K_v \cdot r(t) + v_R(t) \quad (1.84)$$

avec:

- $\hat{f}(.)$  est l'estimation de la fonction  $f$  à chaque instant  $t$  par le réseau RBFG généralisé,
- $K_v \in \mathfrak{R}^{n \times n}$  : matrice gain symétrique définie positive,
- $v_R(t) \in \mathfrak{R}^n$  : couple additionnel robuste donné par l'équation (I.26).

Les lois d'adaptation des réseaux à base radiale généralisés sont :

$$\dot{\hat{C}}_i(t) = F_C \cdot g(\|x - R_i\|) \cdot r^T \quad (I.85a)$$

$$\dot{\hat{R}}_i(t) = F_R \frac{2}{\sigma^2} \cdot (x - \hat{R}_i) \cdot g(\|x - R_i\|) \cdot \hat{C}_i \cdot r \quad (I.85b)$$

avec:  $F_C$  et  $F_R$  sont des matrices constantes symétriques définies positives.

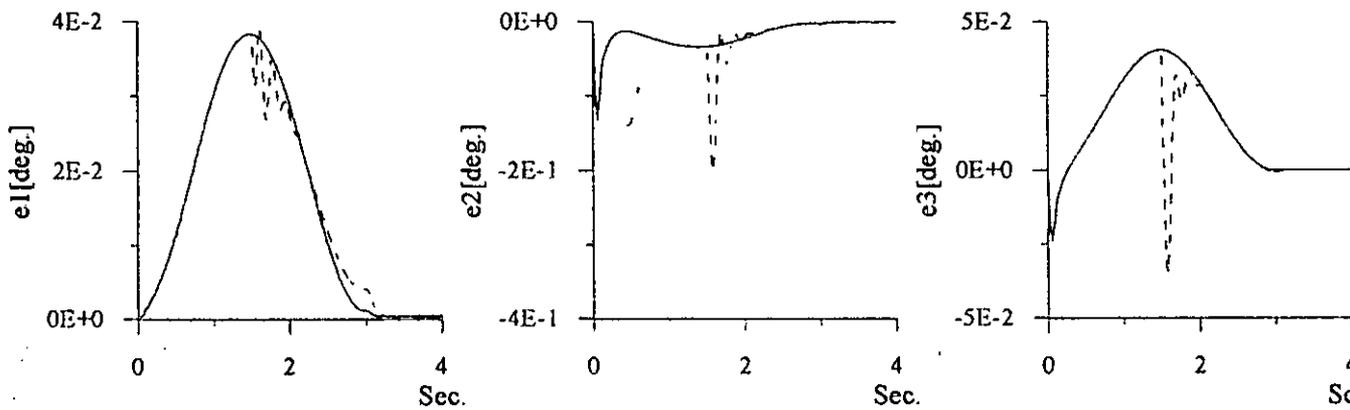
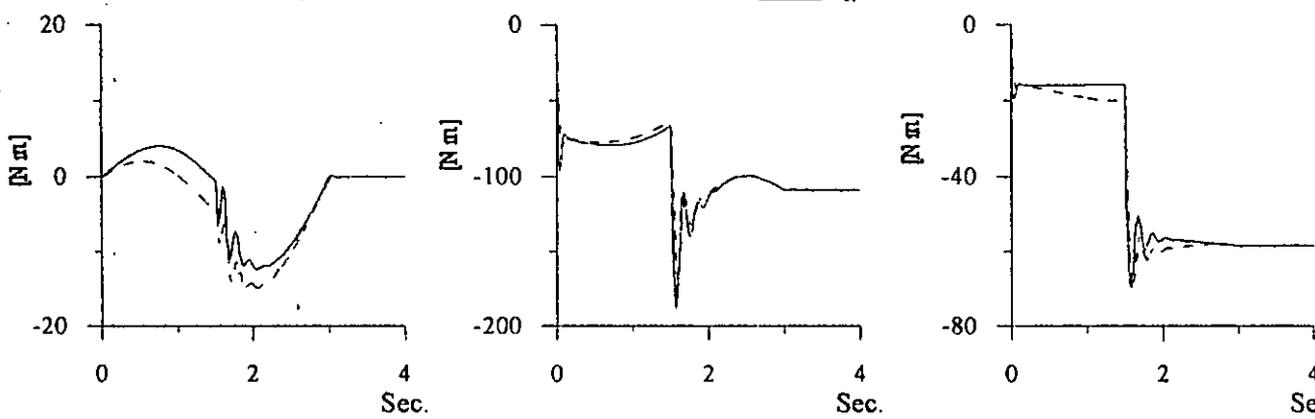
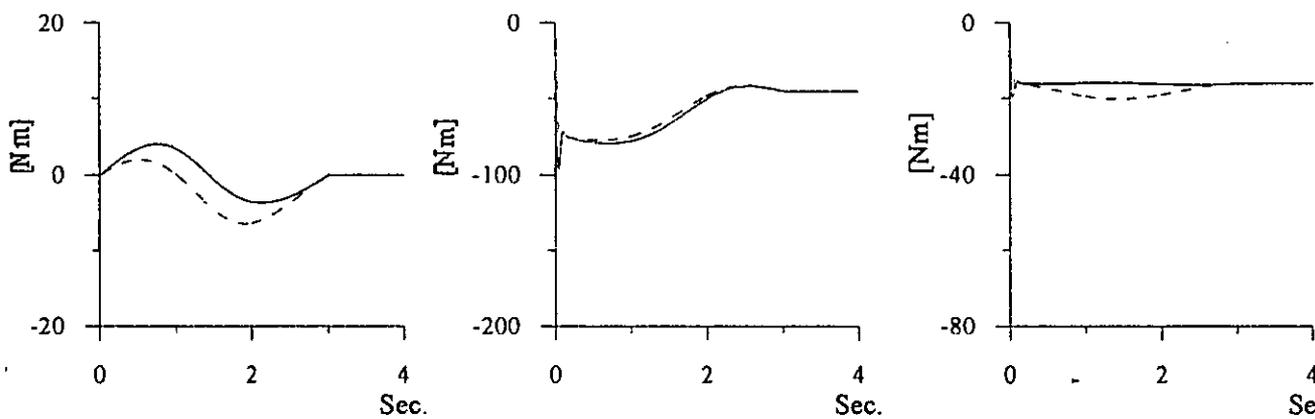
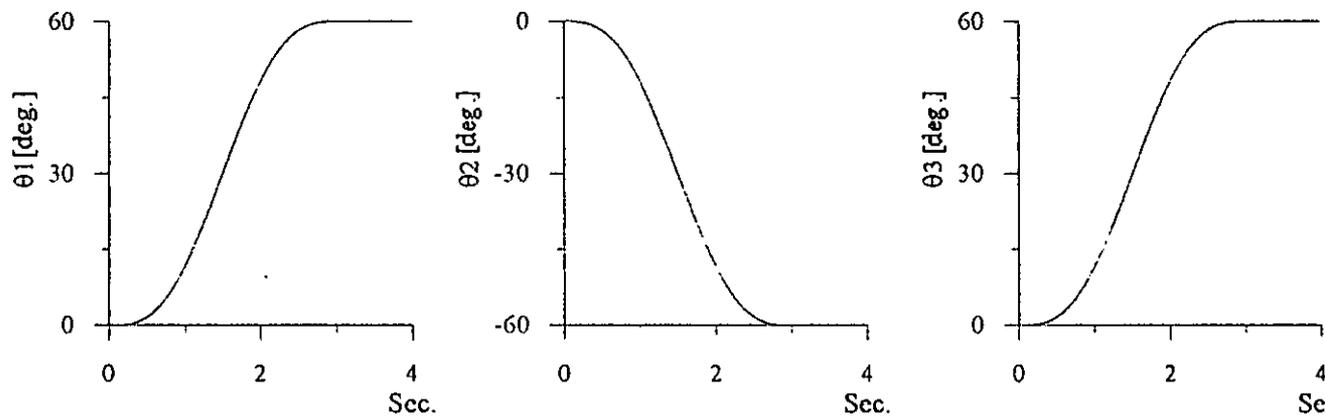
*Remarque : l'étude de la stabilité est la même que celle développée dans la section I.2, en changeant la fonction d'activation sigmoïde par la fonction gaussienne.*

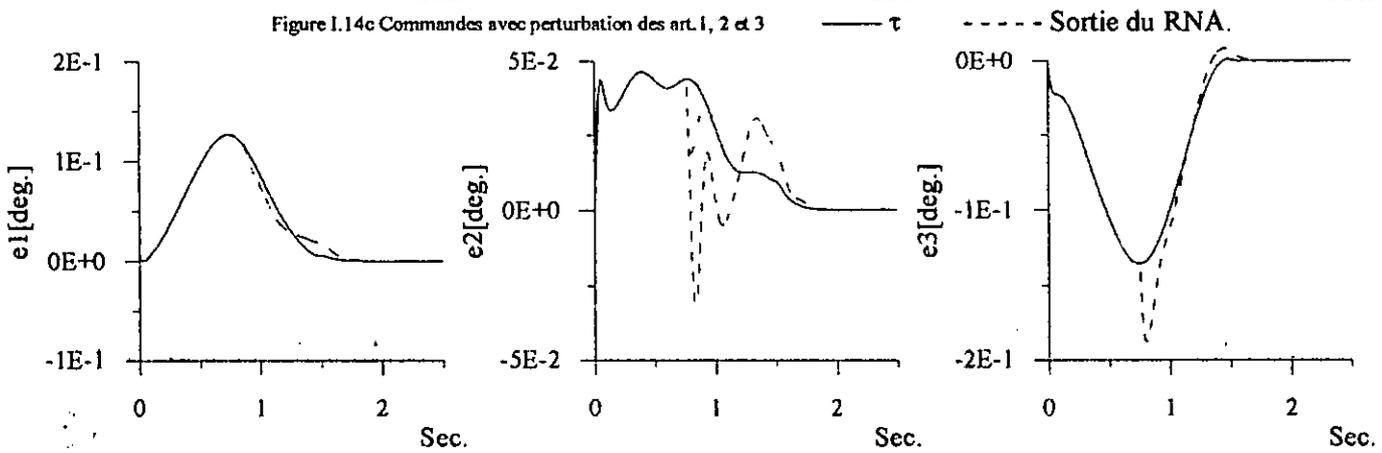
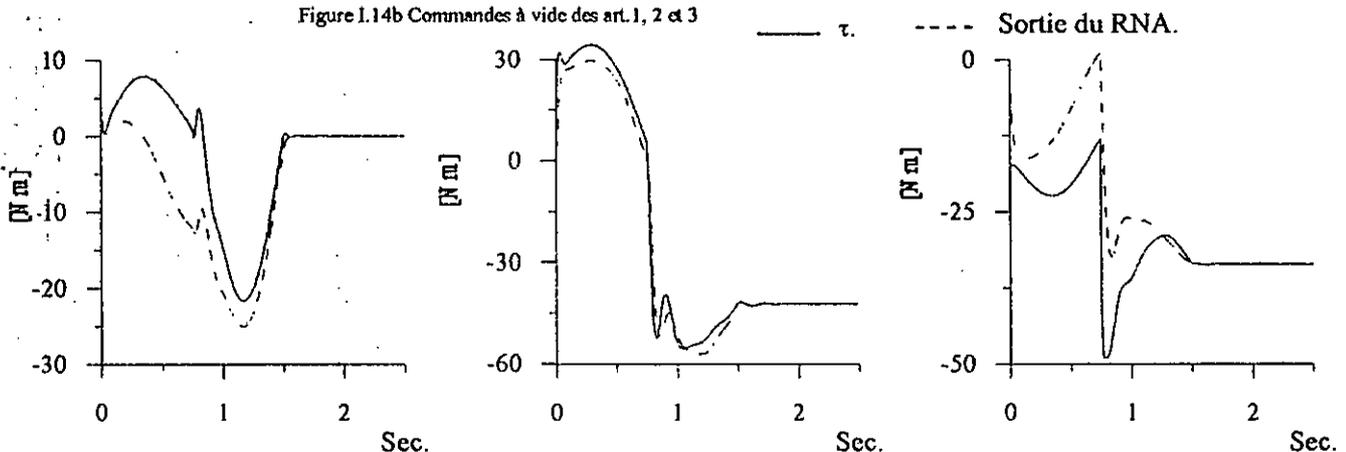
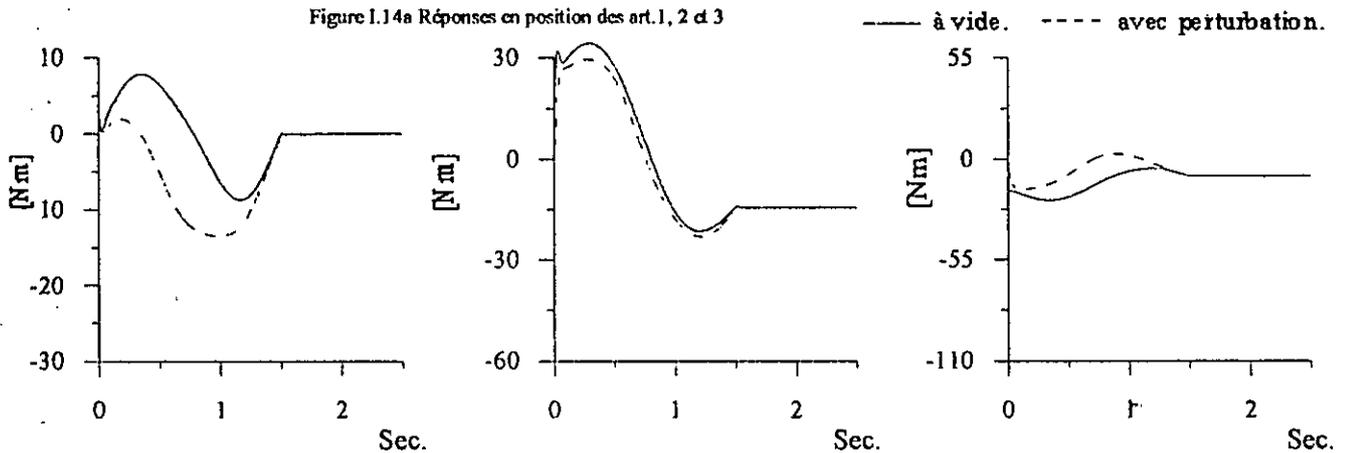
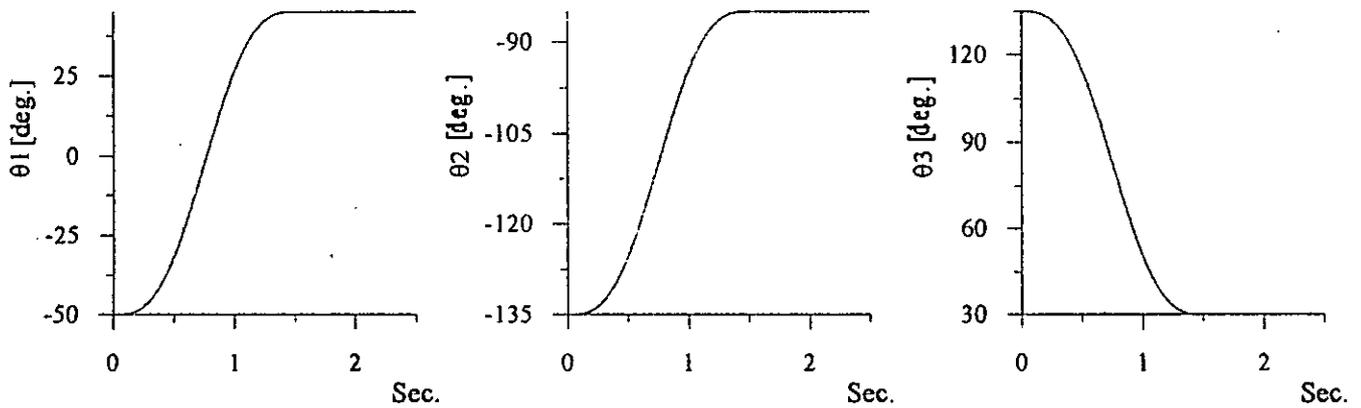
### **Etude par Simulation :**

Pour la validation de la stratégie de commande présentée, les mêmes simulations utilisées précédemment ont été effectuées sur les trois premières articulations d'un robot de type PUMA 560.

Nous avons opté pour trois réseaux de neurones MISO (Multi Input Single Output) comportant : 9 entrées, 10 neurones (centres) dans la couche cachée, tous les poids des réseaux sont initialisés à zéro.

Sur la figure (I.13), on montre l'efficacité et les performances de cette méthode de commande sans et avec le test de perturbation. Les résultats du deuxième test effectué, test de Leahy, sont présentés sur la figure (I.14). Ces résultats de l'application au robot PUMA 560, montrent l'efficacité de la stratégie de commande adaptative directe par RBFG généralisés en termes de poursuite de trajectoire de référence pour les robots manipulateurs, ainsi que la robustesse vis-à-vis des perturbations. Par ailleurs, nous avons diminué le nombre de neurone dans la couche cachée du 19683 à 10. Donc, cette technique permet une grande réduction du nombre de neurone par rapport à la technique de commande basée sur les RBFG à centres fixes.





## 1.5 Commande Prédicative Adaptative Directe par RNA à Fonction de Base Radiale Gaussienne

### 1.5.1 Fonction de Base Radiale à Centres Fixes

En se basant sur la technique de commande présentée dans la section (I.4) et utilisant les mêmes étapes de la commande prédictive présentée dans la section (I.3), la commande prédictive par réseaux gaussiens à centres fixes fera l'objet de cette section.

La fonction couple non linéaire du robot  $f_d(x)$  donnée par l'équation (I.34) est liée aux fonctions de base radiales par la relation (I.51). Donc, cette fonction sera estimée par une fonction possédant la même structure donnée par l'équation (I.34). d'où :

$$u_{ad}(t) - f_d(x) = \sum_{i=1}^N \tilde{\alpha}_i(t) Y_i(x), \quad x = \begin{bmatrix} \theta_d^T & \dot{\theta}_d^T & \ddot{\theta}_d^T \end{bmatrix}^T \quad (I.86)$$

$\tilde{\alpha}_i(t) = \hat{\alpha}_i(t) - \alpha_i$  désigne l'erreur paramétrique.

### Synthèse du contrôleur

Dans cette section, nous allons présenter la loi de commande pour les robots manipulateurs. En supposant connaître a priori la borne supérieure de  $f(x)$  pour les points situés à l'extérieur de l'ensemble  $A_d$ , i.e

$$|f(x)| \leq M_o(x) \quad \text{quand } x \in A_d^c \quad (I.87)$$

nous avons choisi  $A \in A_d$ , dans lequel  $f_{dA}(x)$  est l'approximation gaussienne de  $f_d(x)$ . De l'équation (I.6), la dynamique de l'erreur filtrée  $r(t)$ , définie par (I.61) avec  $n=2$ , est donnée par

$$M(\theta) \cdot \dot{r} = -V_m(\theta, \dot{\theta}) \cdot r - \tau(t) + f_{dA}(x) + d(t) + f(\cdot) - f_d(x) + \tau_d(t) \quad (I.88)$$

la perturbation  $d(t) = f_{dA}(x(t)) - f_d(x(t))$  est bornée  $\forall x(t) \in A$ .

En utilisant les développements des sections précédentes, nous obtenons la loi de commande suivante :

$$u(t) = K_D r(t) + (1 - m(t)) (\hat{f}_{dA}(t, x(t)) + v_R(t)) + m(t) u_{sl}(t) \quad (I.89)$$

$K_D > 0$  et  $k_{sl} = M_o(x(t)) - \varepsilon_f$ . La figure (I.8) illustre le réseau gaussien implémentant la composante adaptative de la loi de commande.

En utilisant cette loi de commande et l'équation (I.88), on peut écrire

$$M(\theta) \cdot \dot{r} = -(K_D + V_m(\theta, \dot{\theta})) \cdot r + (1 - m(t)) (\tilde{f}_{dA} + d(t) + (f(\cdot) - f_d(x)) + \tau_d + v_R(t)) + m(t) (u_{sl}(t) - f(\cdot) + \tau_d) \quad (I.90)$$

et  $\tilde{f}_{dA}(t) = \hat{f}_{dA}(t, x(t)) - f_{dA}(x(t))$

L'ajustement des poids de sorties du réseau est effectué, en utilisant

$$\hat{c}(t) = -k_a (1-m(t)) r(t) g_\sigma(x(t) - \xi_1) \quad (I.91)$$

$k_a > 0$ , désigne le gain d'adaptation.

### Analyse de la stabilité

Soit la fonction positive de Lyapunov suivante :

$$L(t) = \frac{1}{2} r^T M r + \frac{1}{2} e^T K_p e + \frac{1}{2} \text{tr}(\tilde{C}^T K_a^{-1} \tilde{C}) + L_R \quad (I.92)$$

$$L_R = \frac{1}{2} \tilde{\varphi}^T \Gamma^{-1} \tilde{\varphi} + \frac{\delta}{\gamma}$$

Dérivons (I.92) et utilisons (I.90), nous obtenons

$$\dot{L} \leq -z^T Q z + \dot{L}_1 + \dot{L}_2 + \dot{L}_3$$

avec

$$z = \begin{bmatrix} e^T & r^T \end{bmatrix}^T,$$

$$Q = \begin{bmatrix} K_p \Lambda & -\frac{K_p}{2} \\ -\frac{K_p}{2} & K_v \end{bmatrix} > 0 \text{ pour un choix approprié des matrices des gains } K_p \text{ et } K_v,$$

$$\dot{L}_1 = \|r\| S \hat{\varphi} - r^T v_R - \delta, \quad (I.93a)$$

$$\dot{L}_2 = \text{tr}[\tilde{C}^T K_a^{-1} \tilde{C} + \tilde{C}(1-m(t)) r(t) g_\sigma(x(t) - \xi_1)] \quad (I.93b)$$

$$\dot{L}_3 = m(t) r(t) [(M_o(x(t)) + \tau_d) + u_n(t)] \quad (I.93c)$$

utilisons (I.26) dans (I.93a), on obtient

$$\begin{aligned} \dot{L}_1 &= \|r\| S \hat{\varphi} - \frac{r^T r (S \hat{\varphi})^2}{(S \hat{\varphi}) \|r\| + \delta} - \delta \\ &\leq \frac{\delta \|r\| S \hat{\varphi}}{(S \hat{\varphi}) \|r\| + \delta} - \delta \\ &\leq 0. \end{aligned}$$

substituons les lois d'adaptation des poids (I.91) dans (I.93b), on trouve

$$\dot{L}_2 = 0 \quad (I.94)$$

en utilisant la commande  $u_{si}$  donnée par l'équation (I.52), il vient

$$\dot{L}_3 \leq 0 \quad (I.95)$$

en combinant les résultats ci dessus, on obtient

$$\dot{L} \leq -z^T Q z, \quad Q > 0 \quad (I.96)$$

(I.96) implique  $L$ ,  $z$ ,  $e$ ,  $s$ ,  $\tilde{C}$ , et  $\hat{\phi}$  sont tous bornés. A partir de l'équation (I.32), il est facile de montrer que le carré de  $z(t)$  est intégrable ; d'où les carrés de  $e$  et  $r$  sont intégrables. De l'équation de l'erreur en boucle fermée (I.77), il est aussi facile de montrer que  $\dot{r}$  est borné. Par conséquent,  $r$  est uniformément continu. Par la Lemme de Barbalat [KWA 95b], nous montrons que  $e$  et  $\dot{e}$  les tendent asymptotiquement vers zéro.

### Etude par Simulation

En se basant sur la trajectoire désirée, l'ensemble  $A_d$  est choisi en utilisant les bornes min et max de  $\theta_d$ ,  $\dot{\theta}_d$  et  $\ddot{\theta}_d$ . Les ensembles  $A_{dt}$  sont subdivisés à  $3^9 = 19683$  neurones (centres).  $\Psi=0.005$  et  $K_D=\lambda=600$  et  $K_p=6000$ .  $k_{s1} = 0.2$ ,  $k_a = 50$ .

La stratégie de commande prédictive adaptative par RBFG à centres fixes présentée a été testée pour la commande en poursuite de trajectoires de références utilisées précédemment utilisant le même robot. Sur les figures (I.15) et (I.16), on montre l'efficacité et les performances de cette stratégie de commande en termes de poursuite de trajectoire de référence pour les robots manipulateurs, ainsi que la robustesse vis-à-vis des perturbations.

L'apport de cette technique de commande par rapport à celle développée dans la section (I.4.1.b) (commande adaptative par RBFG à centres fixes) réside dans le calcul hors ligne des sorties de la couche cachée qui ne dépends que de la trajectoire désirée et les centres fixés à priori. Cela, nous permet de réduire le temps de calcul en ligne.

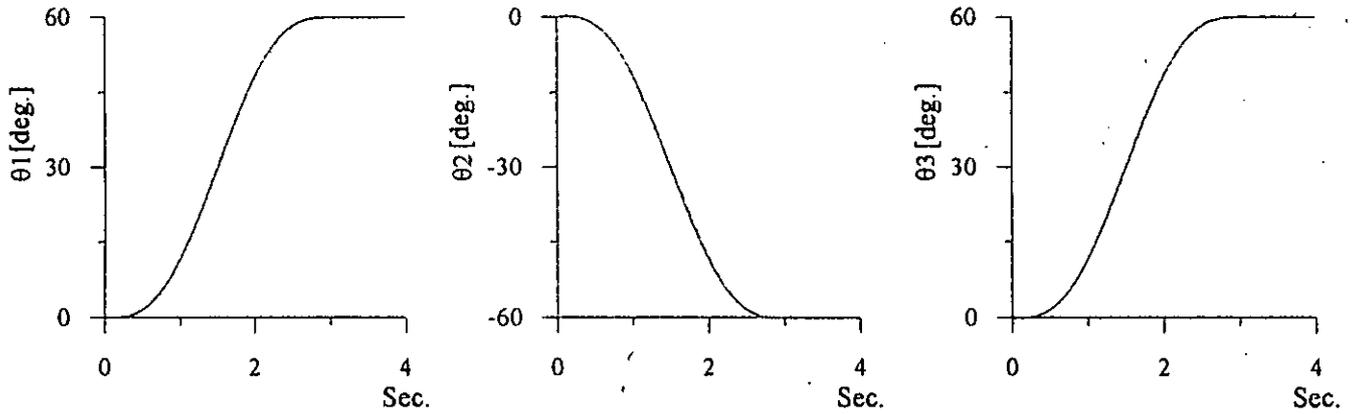


Figure 1.15a Réponses en position des art. 1, 2 et 3

----- à vide. - - - - avec perturbation.

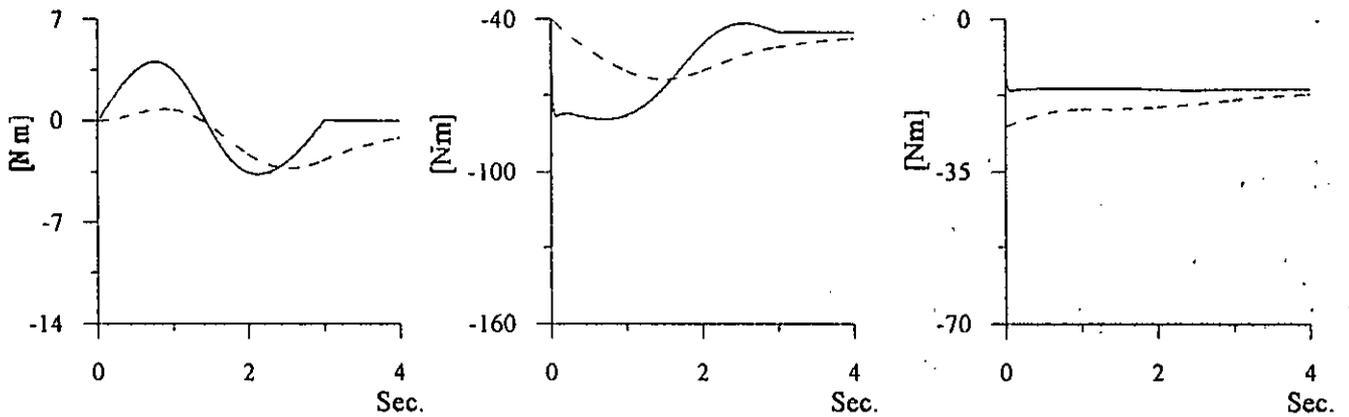


Figure 1.15b Commandes à vide des art. 1, 2 et 3

——  $\tau$ .

----- Sortie du RNA.

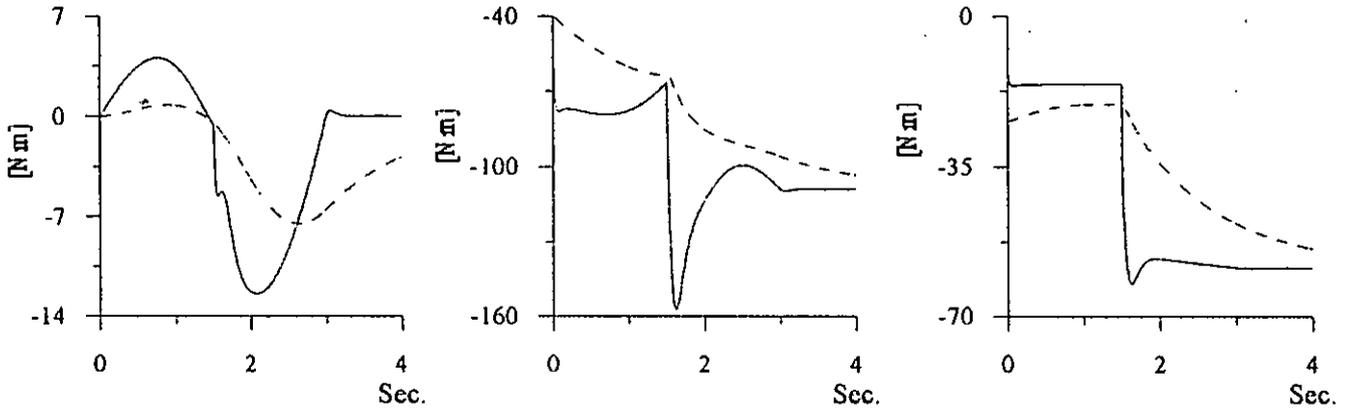


Figure 1.15c Commandes avec perturbation des art. 1, 2 et 3

——  $\tau$

----- Sortie du RNA.

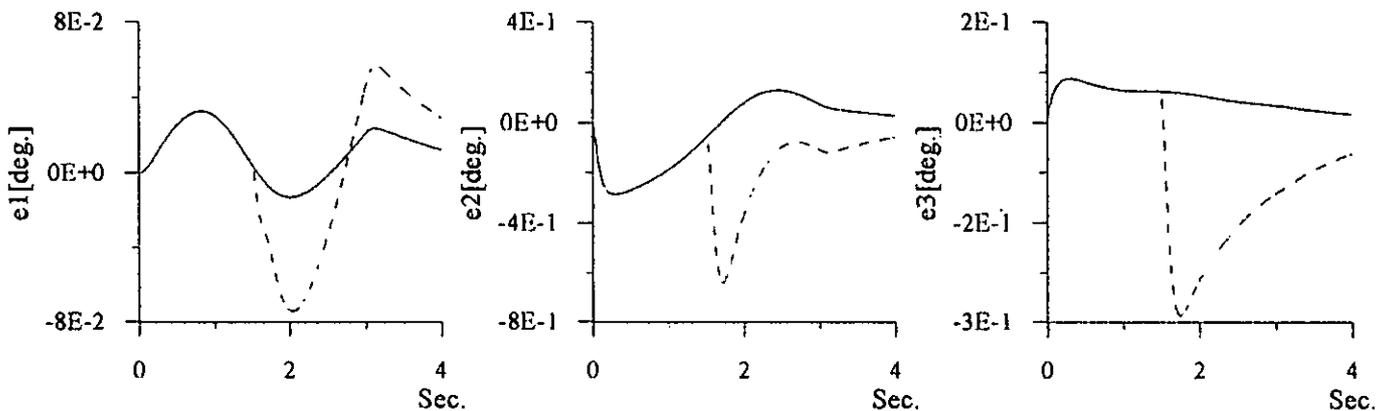
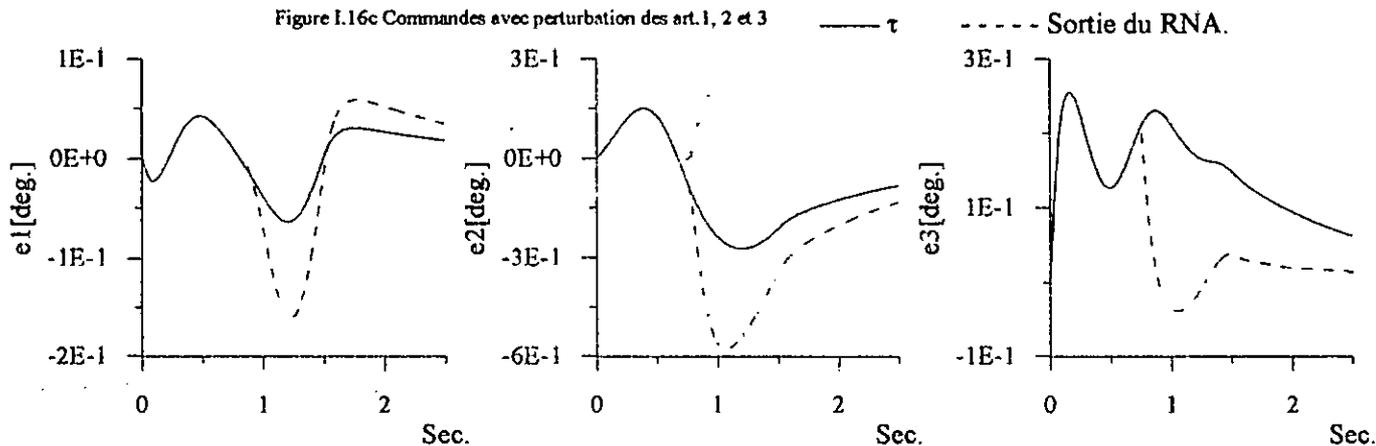
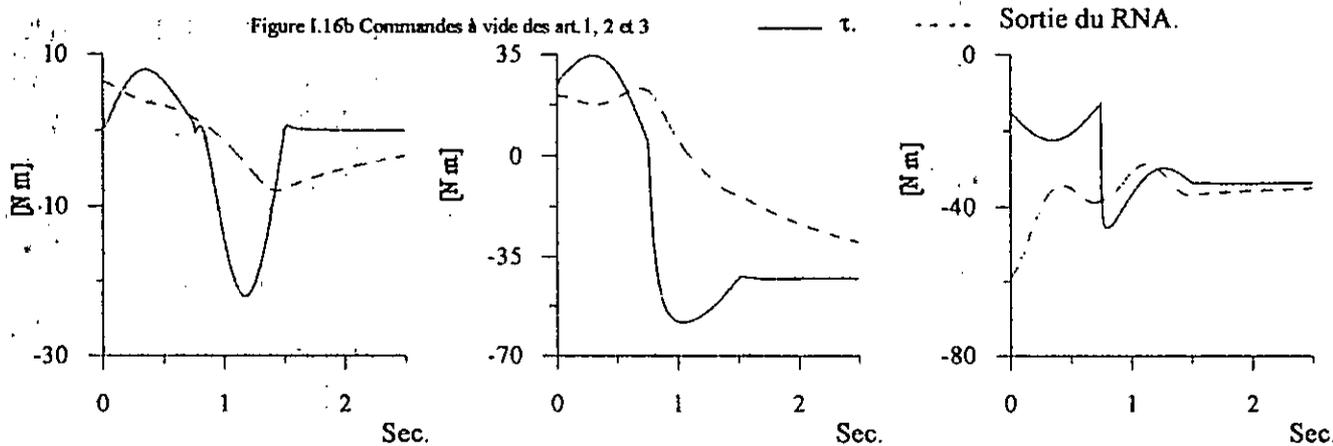
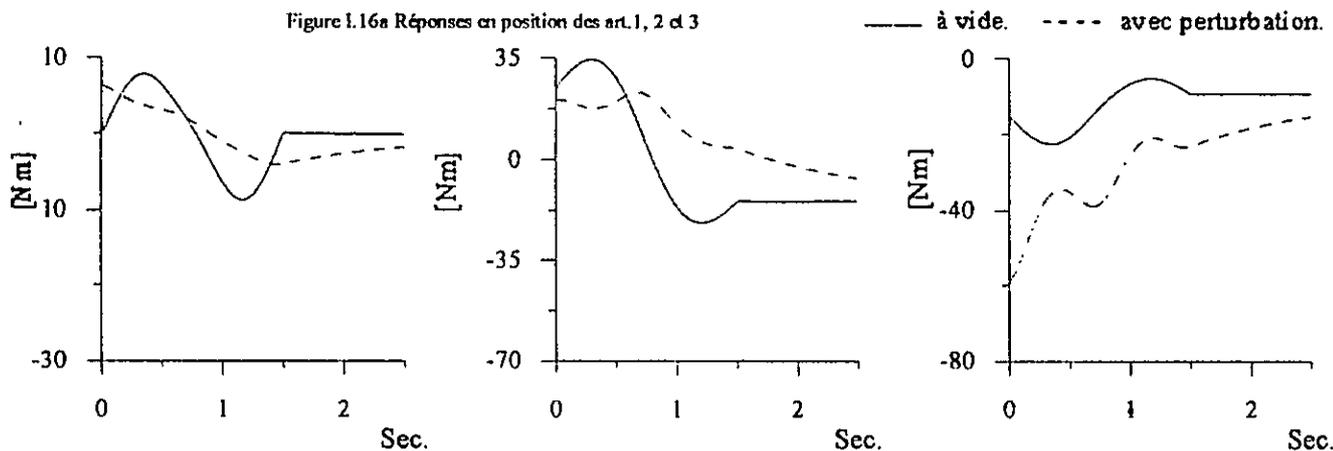
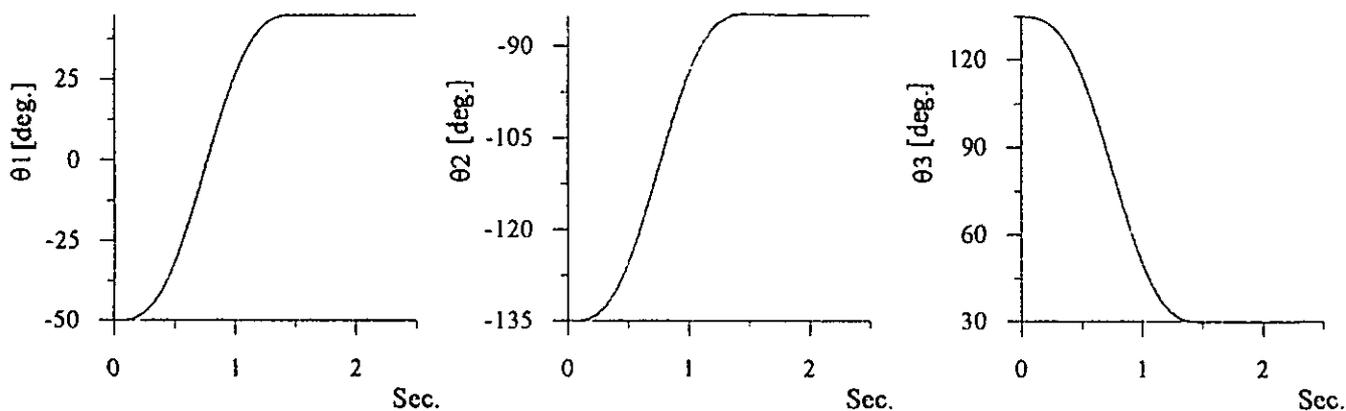


Figure 1.15d Erreurs des art. 1, 2 et 3

—— à vide. - - - - avec perturbation.



### I.5.2 Fonction de Base Radiale à Centres Adaptatifs (Réseaux à Fonction de Base Radiale Généralisés)

Dans cette section, la commande prédictive adaptative par réseaux à fonction de base radiale gaussiens généralisés est développée utilisant la même structure de commande présentée dans la section (I.4.2). Le réseau Gaussien à pour rôle d'estimer la fonction couple non linéaire donnée par l'équation (I.34). Nous utilisons les mêmes étapes dans la synthèse du régulateur ainsi que dans l'étude de la stabilité de la section (I.4.2), remplaçons les poids d'entrée du RNA statique par les centres du réseau RBFG.

La commande est donnée par l'équation suivante :

$$\tau(t) = \hat{f}_d(x) + K_v \cdot r(t) + v_R(t) \quad (\text{I.97})$$

$$x = \left[ \theta_d^T \quad \dot{\theta}_d^T \quad \ddot{\theta}_d^T \right]^T$$

avec:

- $\hat{f}_d(.)$  est l'estimation de la fonction  $f_d$  à chaque instant  $t$  par le réseau RBFG généralisé,
- $K_v \in \mathfrak{R}^{m \times n}$  : matrice gain symétrique définie positive,
- $v_R(t) \in \mathfrak{R}^n$  : couple additionnel robuste donné par l'équation (I.26).

Les lois d'adaptation des réseaux à base radiale généralisés sont les mêmes que celles données par (I.85a) et (I.85b).

#### **Etude par Simulation**

Pour la validation de la stratégie de commande présentée, les mêmes simulations utilisées précédemment ont été effectuées sur les trois premières articulations d'un robot de type PUMA 560.

Nous avons opté pour trois réseaux de neurones MISO (Multi Input Simple Output) comportant : 9 entrées, 10 neurones (centres) dans la couche cachée, tous les poids des réseaux sont initialisés à zéro.

La figure (I.17) montre l'efficacité et les performances de cette méthode de commande sans et avec perturbation.

Les résultats du deuxième test effectué, test de Leahy, sont présentés sur la figure (I.18). Nous constatons que les résultats obtenus pour les deux tests effectués sont satisfaisants.

Les résultats obtenus, de l'application au robot PUMA 560, montrent l'efficacité de la stratégie de commande adaptative directe par RBFG généralisés en termes de poursuite de trajectoire de références pour les robots manipulateurs, ainsi que la robustesse vis-à-vis des perturbations.

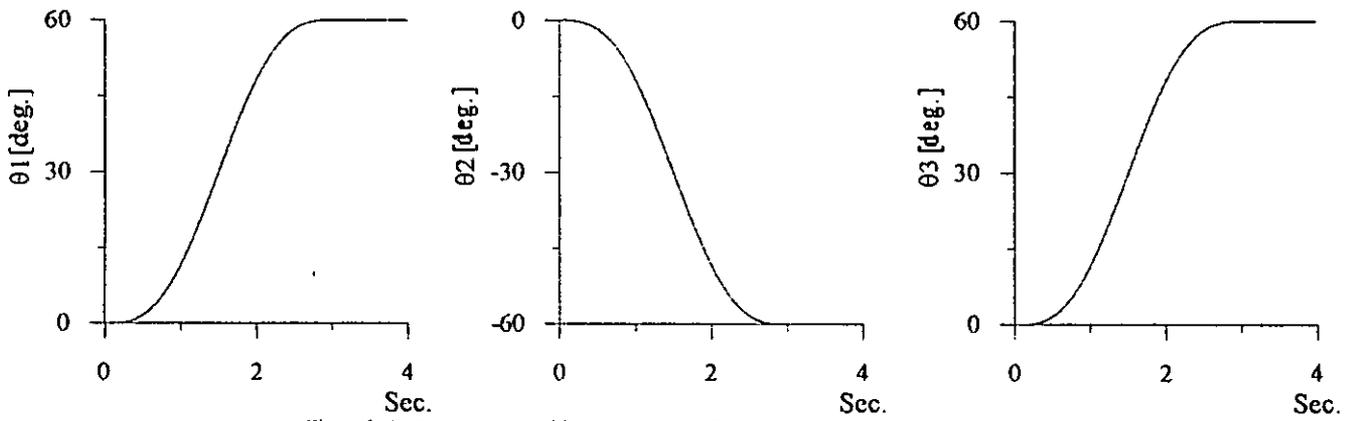


Figure 1.17a Réponses en position des art. 1, 2 et 3

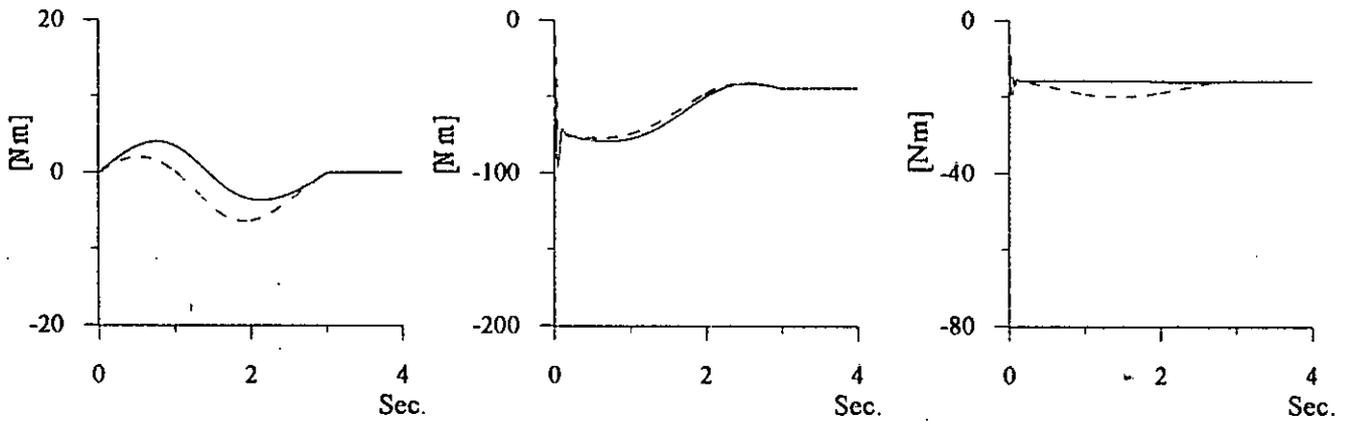


Figure 1.17b Commandes à vide des art. 1, 2 et 3

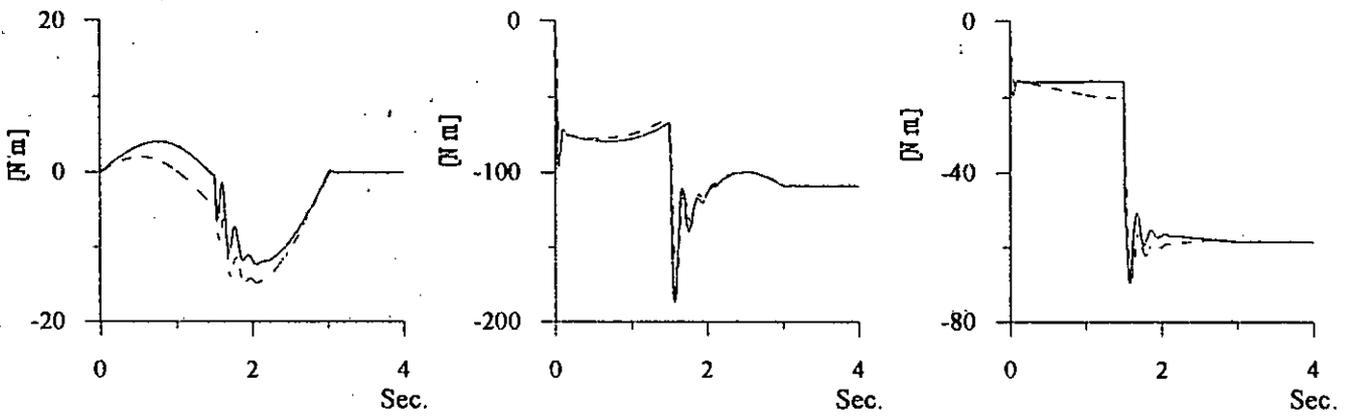


Figure 1.17c Commandes avec perturbation des art. 1, 2 et 3

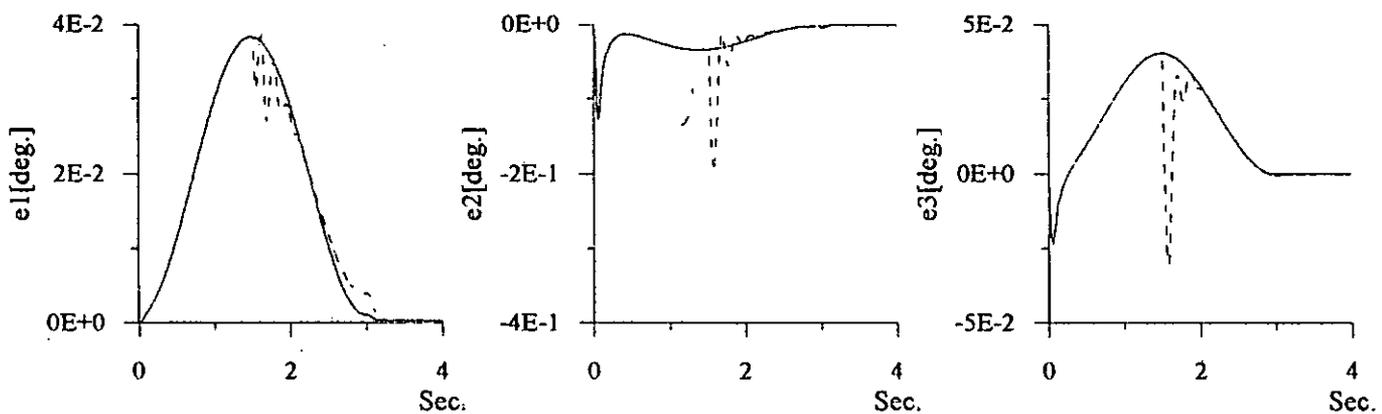


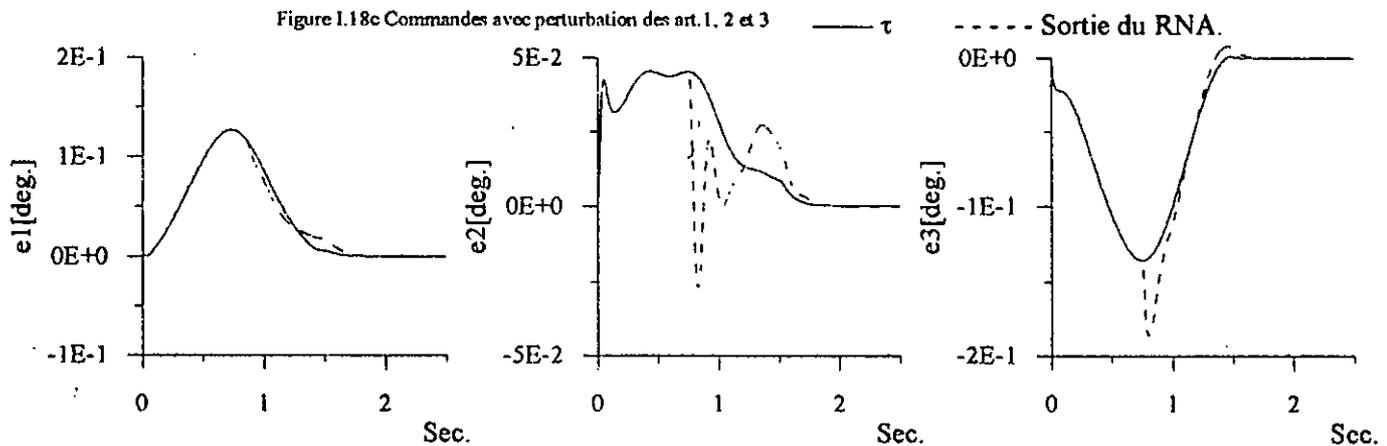
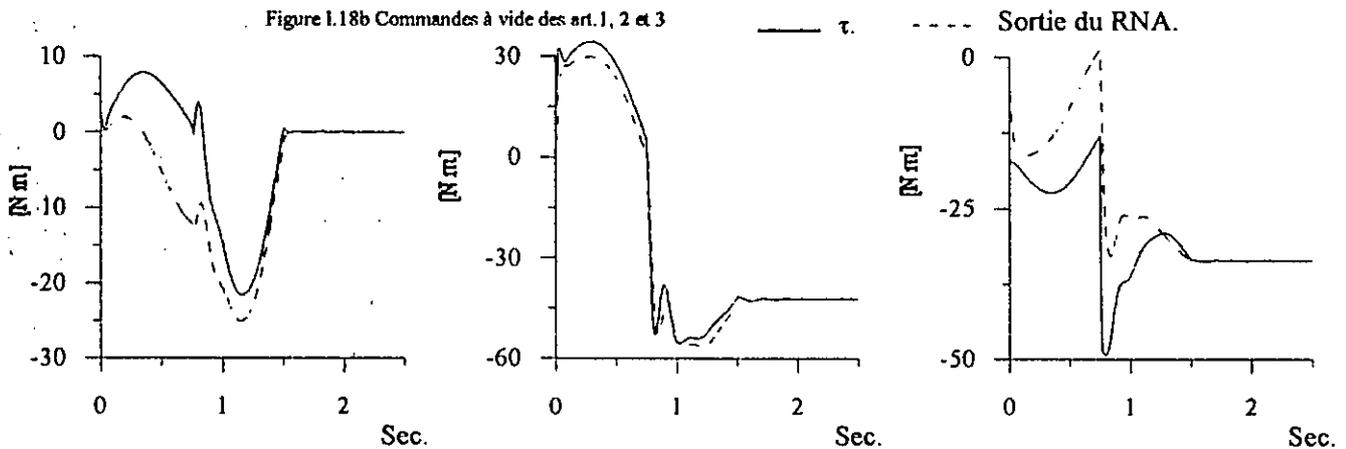
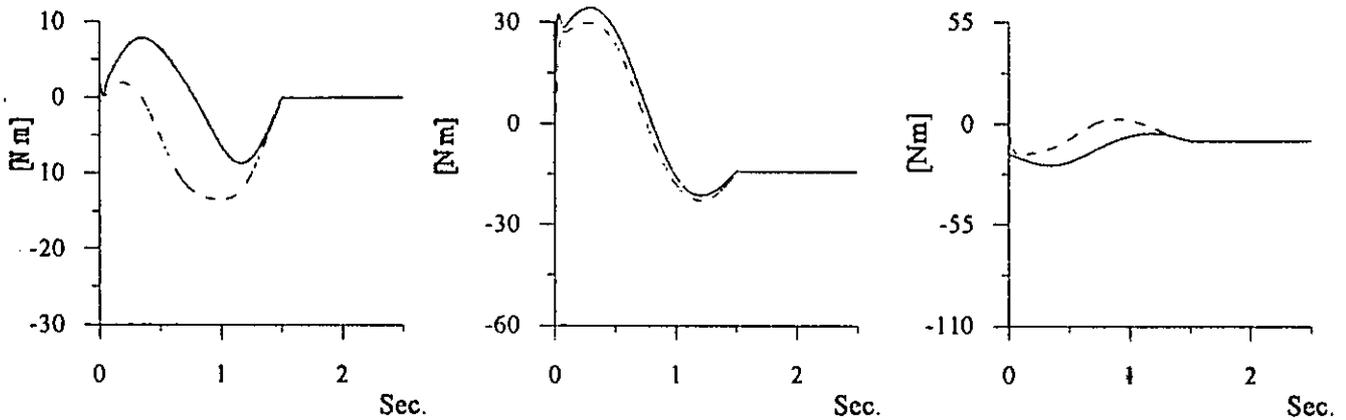
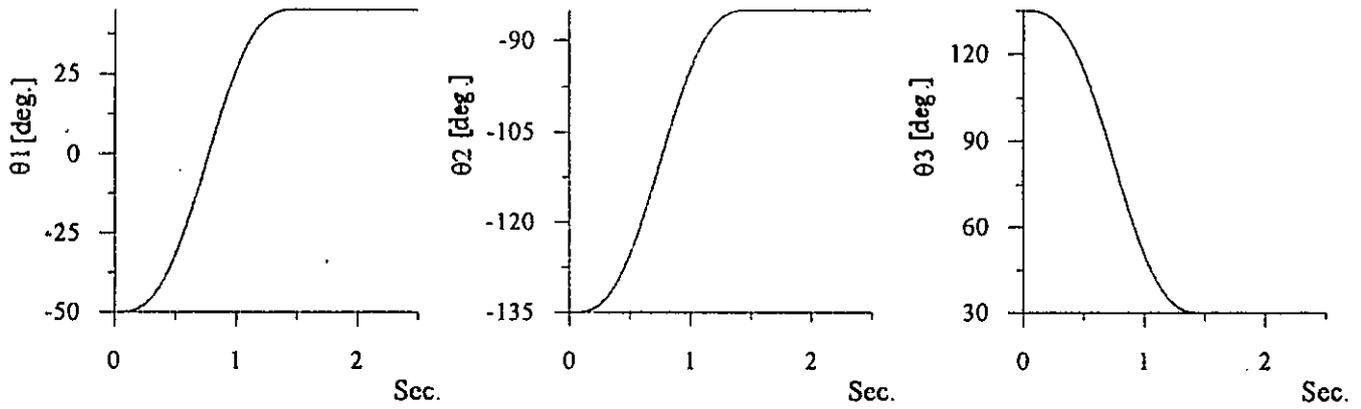
Figure 1.17d Erreurs des art. 1, 2 et 3

----- à vide.    - - - - avec perturbation.

——  $\tau$ .    - - - - Sortie du RNA.

——  $\tau$ .    - - - - Sortie du RNA.

—— à vide.    - - - - avec perturbation.



## 1.6 Commande Adaptative Centralisée Directe par RNA Récurrents (RNAR)

Les techniques récentes développées dans le domaine de la commande adaptative possèdent un énorme potentiel d'applications, surtout lorsque les systèmes à commander sont en présence d'incertitude. Malgré que les systèmes adaptatifs soient de nature non linéaire, la plus part des théories traitant ces systèmes sont basés sur la théorie linéaire [SLO 91][KRS 95]. Par ailleurs, Les RNA multicouches et récurrents, ayant eu un grand succès dans les problèmes de reconnaissance de formes et les problèmes d'optimisation, peuvent être considérés comme éléments importants à utiliser dans la commande des systèmes non linéaires [HUN 92],[BOU 97b]. Du point de vue théorique, les RNA multicouches représentent une correspondance non linéaire statique, par contre, les réseaux récurrents sont des systèmes dynamiques [NAR 90a] [NAR 90b]. A l'inverse des RNA multicouches à propagation avant (feed forward ou layered networks), sans boucle de retour, où une large bibliographie est disponible, les RNA bouclés ou récurrents (Recurrent Networks) n'ont pas eu le même privilège, ils ont été peu étudiés. Malgré leurs difficultés de conception. Ce type de réseau est d'une importance particulière pour l'avenir, d'après certains spécialistes du domaine [HAM 97].

Leurs grandes difficultés se manifestent dans l'ajustement des paramètres du réseau car il n'existe pas d'algorithme d'entraînement standard. Chatry [CHA 97] a proposé une méthode, pour l'identification hors ligne, basée sur l'approche de Lagrange. Cette méthode est vue comme une généralisation de l'algorithme de la rétropropagation (backpropagation) classique.

Dans cette section, l'algorithme d'adaptation des poids des réseaux récurrents, proposé par Chatry est combiné avec l'algorithme d'entraînement des réseaux de neurones généralisés [BOU-98a], afin de l'utiliser en ligne dans la commande adaptative par réseaux dynamiques bouclés.

### 1.6.1 Structure des RNAR Utilisées

La structure des RNAR utilisés est présentée dans la figure (I.18). Les réseaux utilisés sont à trois couches ; une couche d'entrée, une couche cachée et une couche de sortie. La couche d'entrée est composée de deux parties, l'une des entrées externes  $u = [e \quad \theta_a \quad \dot{\theta}_a \quad \ddot{\theta}_a]$  et l'autre des entrées bouclées  $y_{k-1}, \dots, y_{k-ny}$ .  $A_k^c$  et  $V_k^c$  sont respectivement l'activité et la sortie de la couche c. Le réseau de neurone récurrent est défini par l'équation (I.98).

$$\begin{cases} A_k^1 = \sum_{i=0}^{m_1} W_i^{E_1} u_{k-i} + \sum_{j=1}^{ny} W_j^{w_1} y_{k-j} \\ V_k^1 = f_1(A_k^1) \\ A_k^2 = W^2 V_k^1 \\ y_k = V_k^2 = f_2(A_k^2) \end{cases} \quad (I.98)$$

### 1.6.1 Algorithme d'Adaptation des RNAR

L'entraînement des poids des RNAR est effectué par la rétropropagation de l'erreur filtrée  $r_i(t)$ .

Les entrées externes du réseau RNAR<sup>i</sup> sont  $\theta_i$  et  $\dot{\theta}_i$  et l'erreur rétropropagée est [BOU 98a] [OZA 91]:

$$V_d^2 - V_k^2 = r_i(t) \quad \text{pour RNAR}^i$$

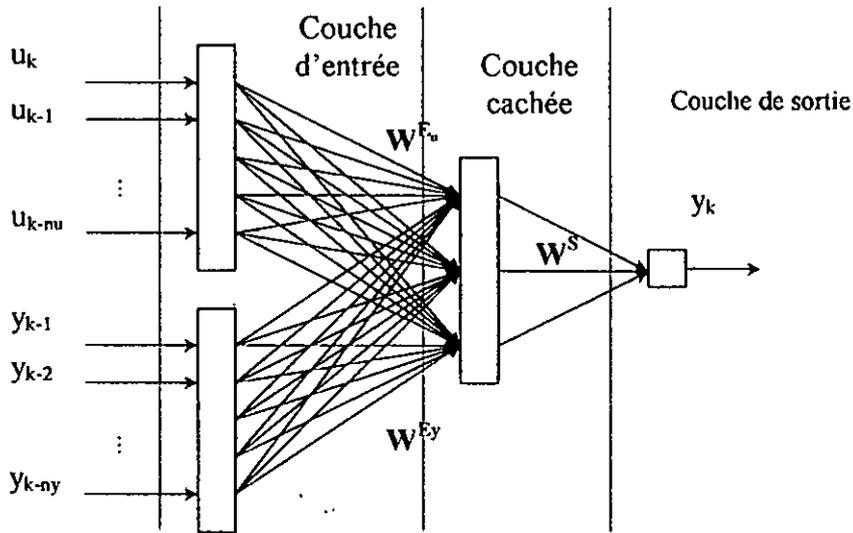


Figure 1.19. structure des RNAR utilisés

Le Lagrangien associé à ce problème d'optimisation est donné par :

$$L = \frac{1}{2} \sum_{k=ny}^k (V_k^2 - V_k^d)^2 + \sum_{k=ny}^k \Psi_k^{1T} (V_k^1 - f^1(A_k^1)) + \sum_{k=ny}^k \Psi_k^{2T} (V_k^2 - f^2(A_k^2)) \quad (I.99)$$

En utilisant les conditions de stationnarité  $\frac{\partial L}{\partial V^2} = 0$  et introduisant la variable  $Y_k^c = F_k^c \psi_k^c$ ,  $c = 1, 2$ , on obtient l'algorithme d'adaptation suivant

$$\begin{aligned} Y_{k-i}^2 &= F_{k-i}^2 (V_{k-i}^2 - V_{k-i}^d) + \sum_{j=k-i+1}^k F_j^2 W_j^{E_y} F_j^1 W_j^{S^T} Y_j^2 \\ Y_{k-i}^1 &= F_{k-i}^1 W^{S^T} Y_{k-i}^2, \quad i = 0, \dots, ny \\ \nabla J_i^{E_u} &= \sum_{j=0}^{ny} Y_{k-j}^1 u_{k-i-j}, \quad i = 0, \dots, nu \\ \nabla J_i^{E_y} &= \sum_{j=0}^{ny} Y_{k-j}^1 y_{k-i-j}, \quad i = 1, \dots, ny \\ \nabla J_{W^S} &= \sum_{j=0}^{ny} Y_{k-j}^2 V_{k-j}^1 \end{aligned} \quad (I.100)$$

à noter que l'ajustement des paramètres des réseaux se réalise à chaque  $(ny+1)$  pas d'échantillonnage [NAR 90b].

### Résultats de Simulation

Nous avons opté pour des RNAR chacun a trois couches, la couche cachée possède 5 neurones et  $ny=2$  (deux retours de la sortie).

Les résultats de simulation obtenus pour les deux tests sont présentés sur les figures (I.20) et (I.21). nous constatons que les performances de poursuite sont satisfaisantes.

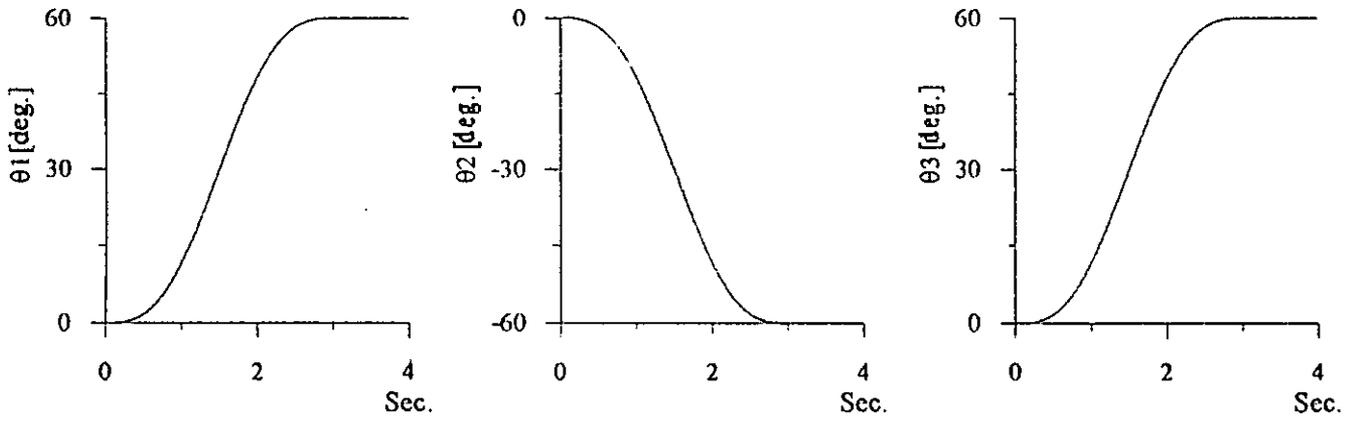


Figure 1.20a Réponses en position des art. 1, 2 et 3

----- à vide.    - - - - avec perturbation.

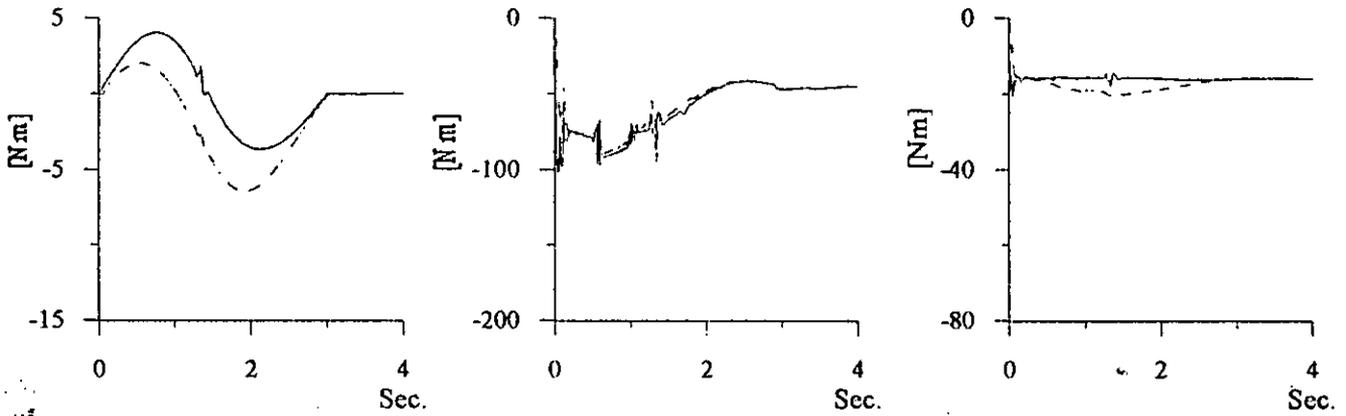


Figure 1.20b Commandes à vide des art. 1, 2 et 3

——  $\tau$ .

- - - - Sortie du RNA.

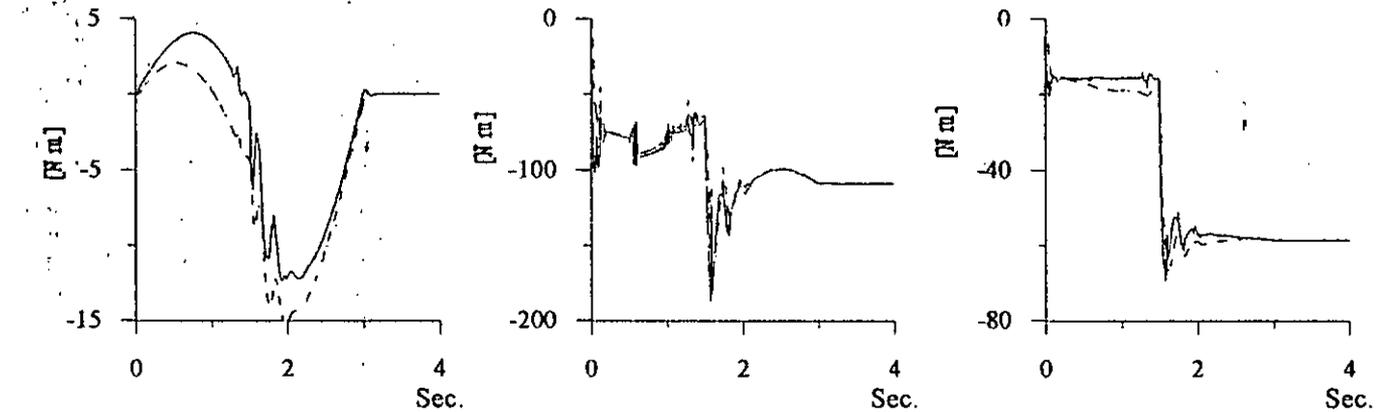


Figure 1.20c Commandes avec perturbation des art. 1, 2 et 3

——  $\tau$

- - - - Sortie du RNA.

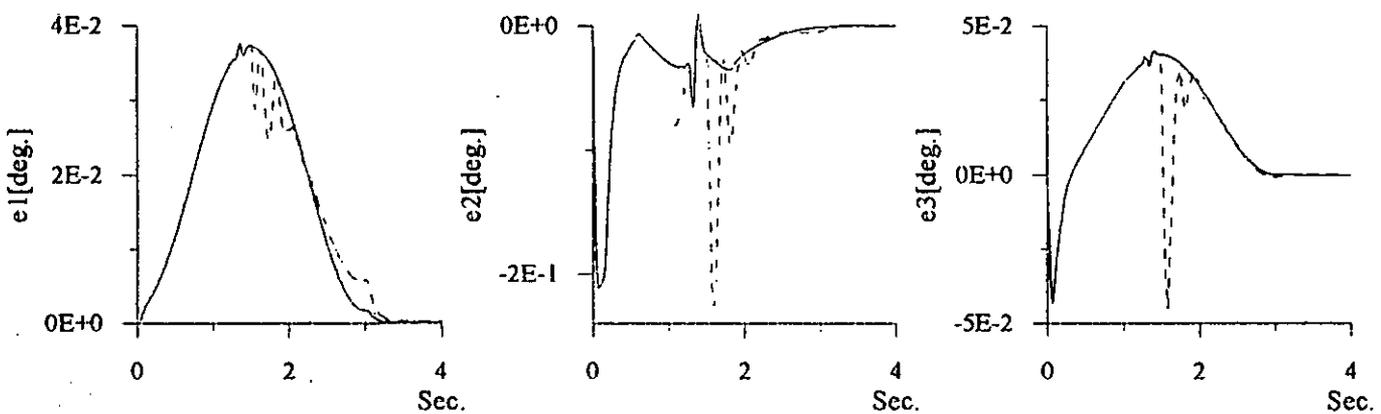
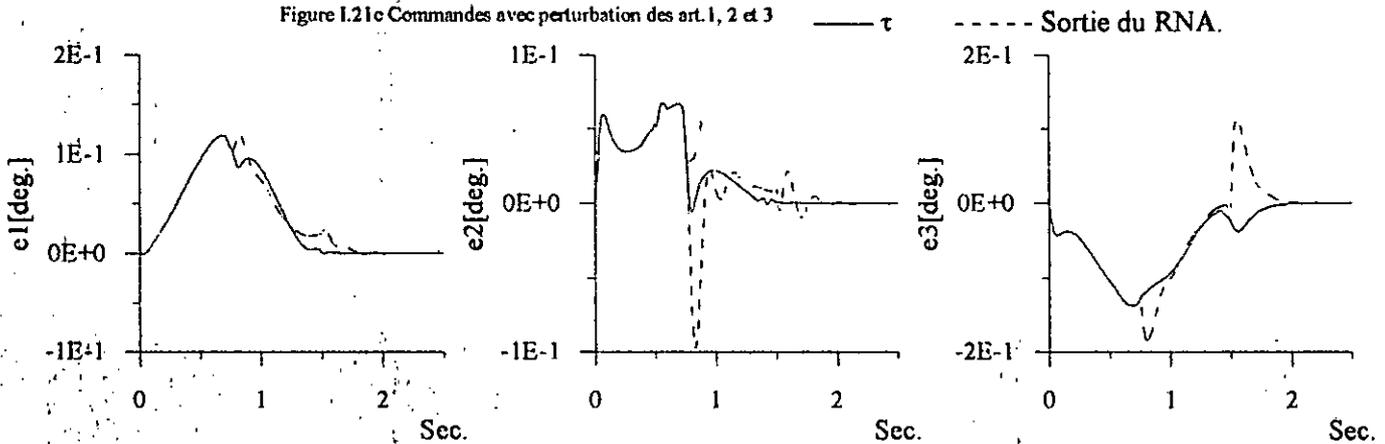
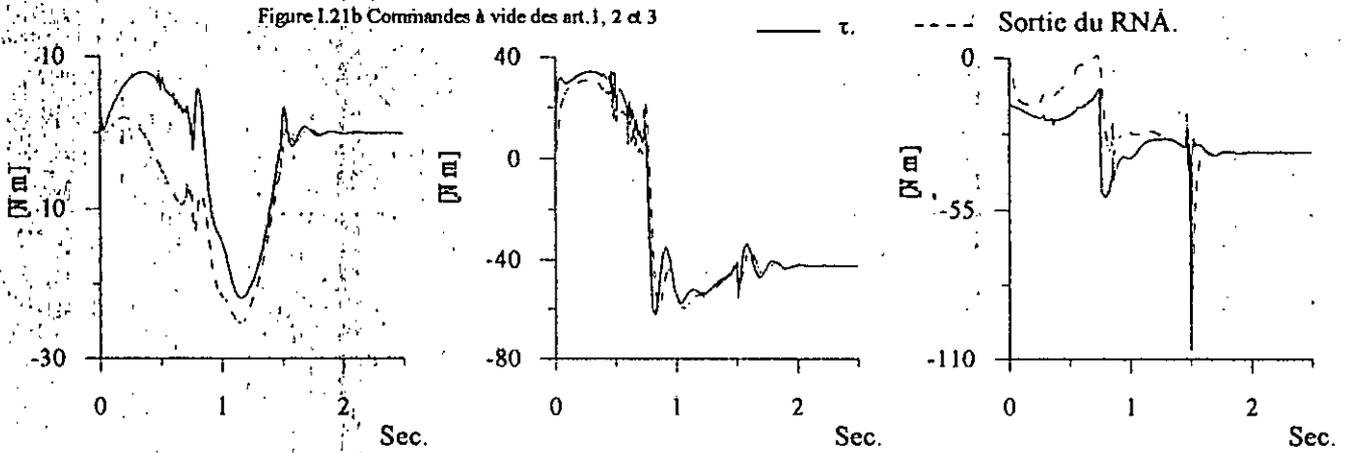
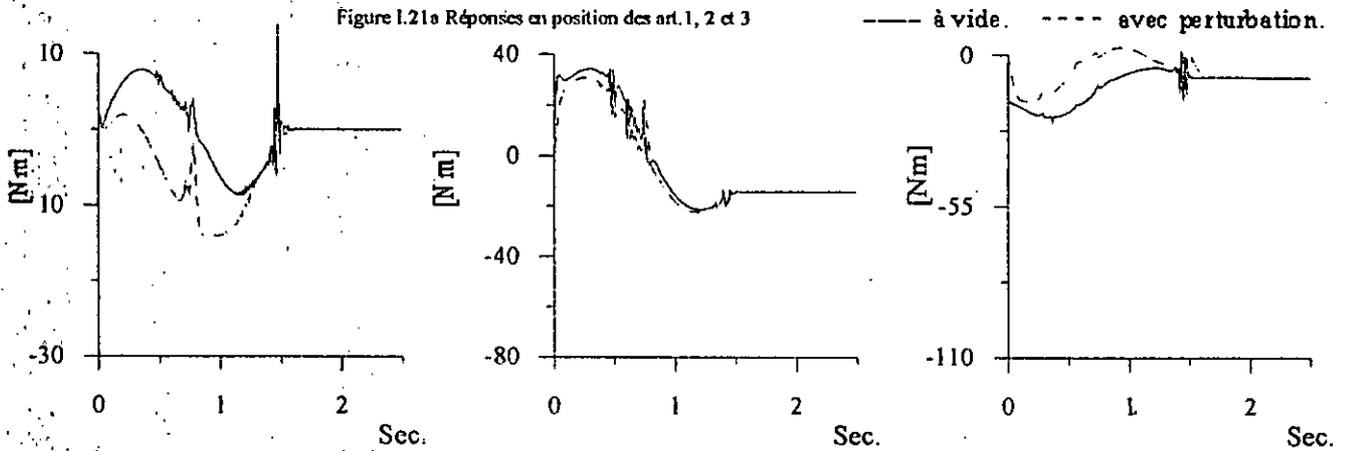
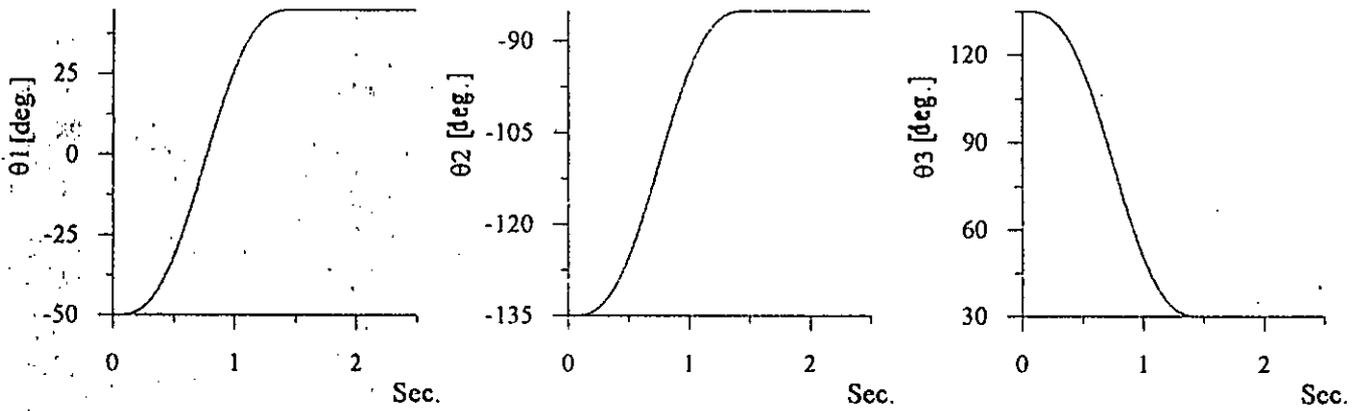


Figure 1.20d Erreurs des art. 1, 2 et 3

—— à vide.    - - - - avec perturbation.



## 1.7. Commande Prédicative Adaptative Directe par RNAR

En se basant sur la technique présentée dans la section 1.6, la commande prédictive adaptative par réseaux récurrents fera l'objet de cette section. L'entrée externe du réseau dans cette technique de commande est donnée par :

$$u = [\theta_d \quad \dot{\theta}_d \quad \ddot{\theta}_d].$$

La loi de commande est donnée par :

$$\tau(t) = \hat{f}_d(x) + K_v r(t) \quad (1.101)$$

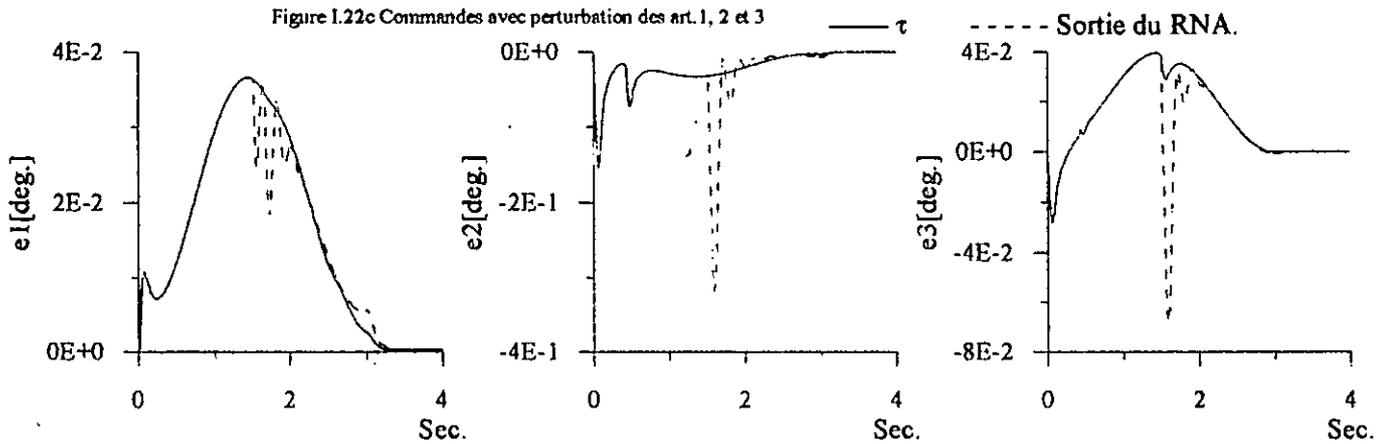
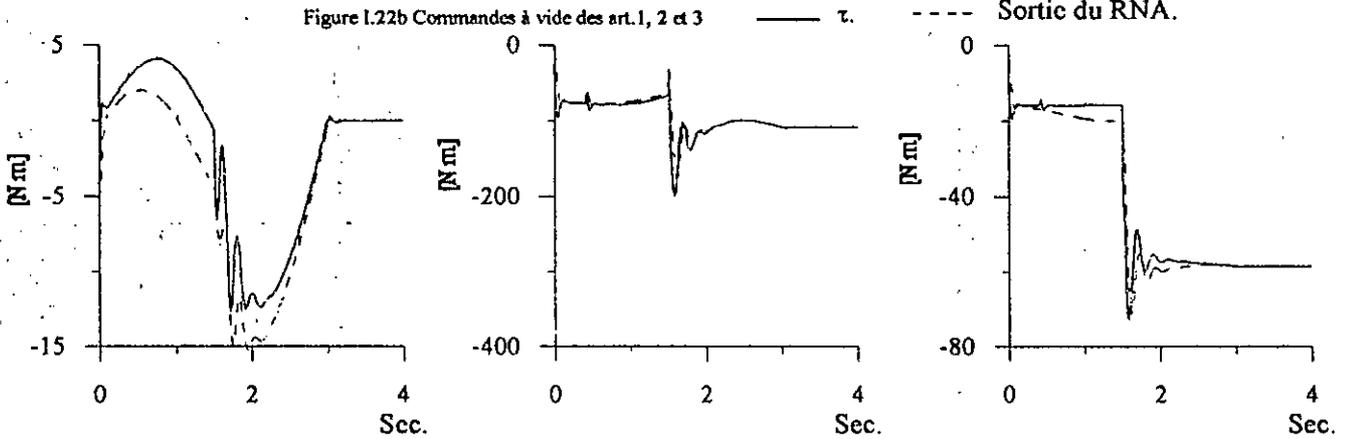
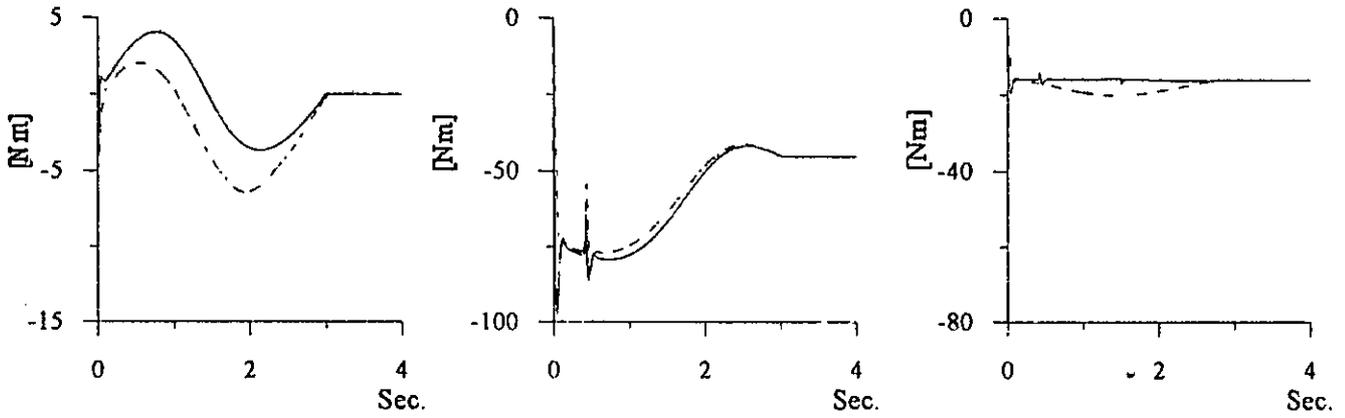
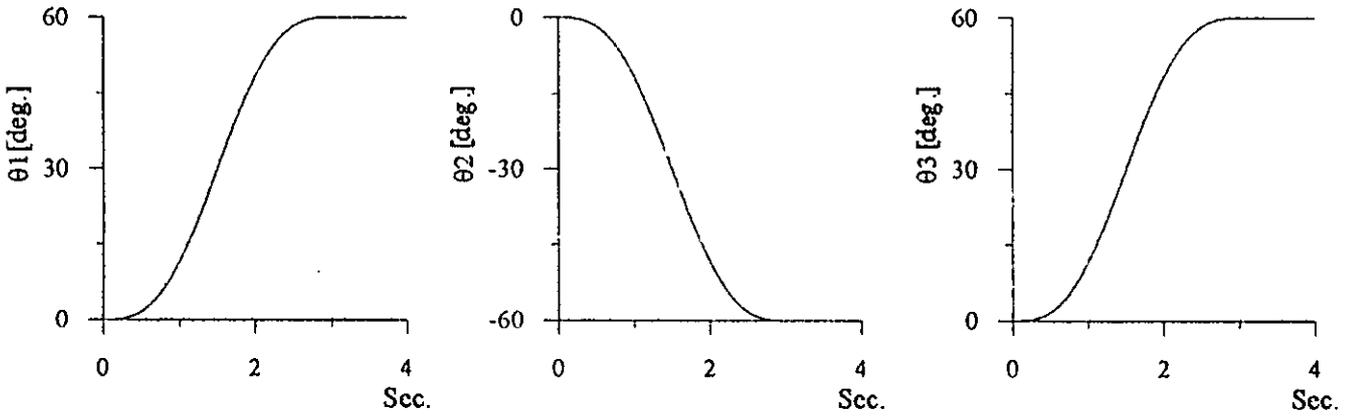
**REMARQUE :** les lois d'adaptation sont les mêmes que celles de la commande précédente (section 1.6).

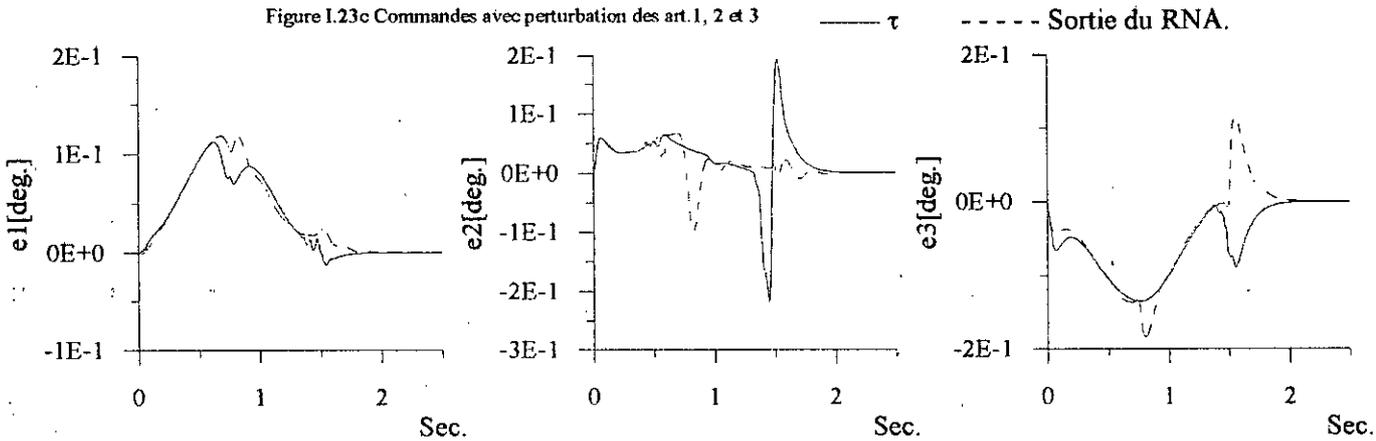
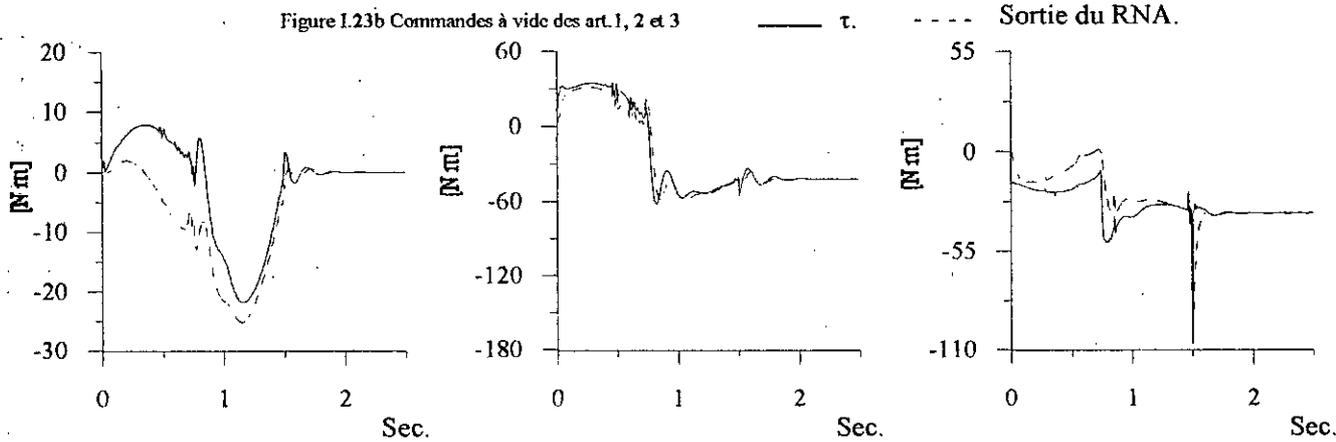
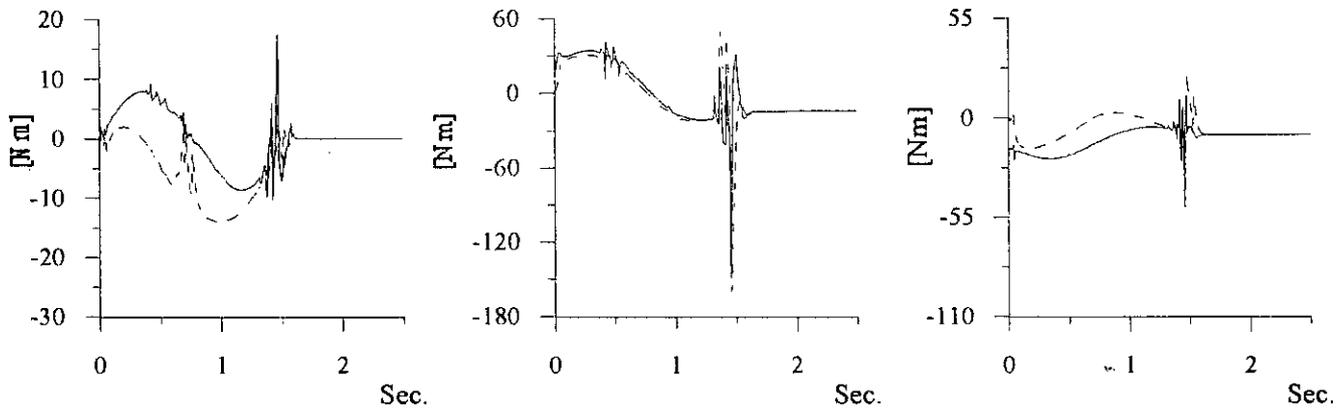
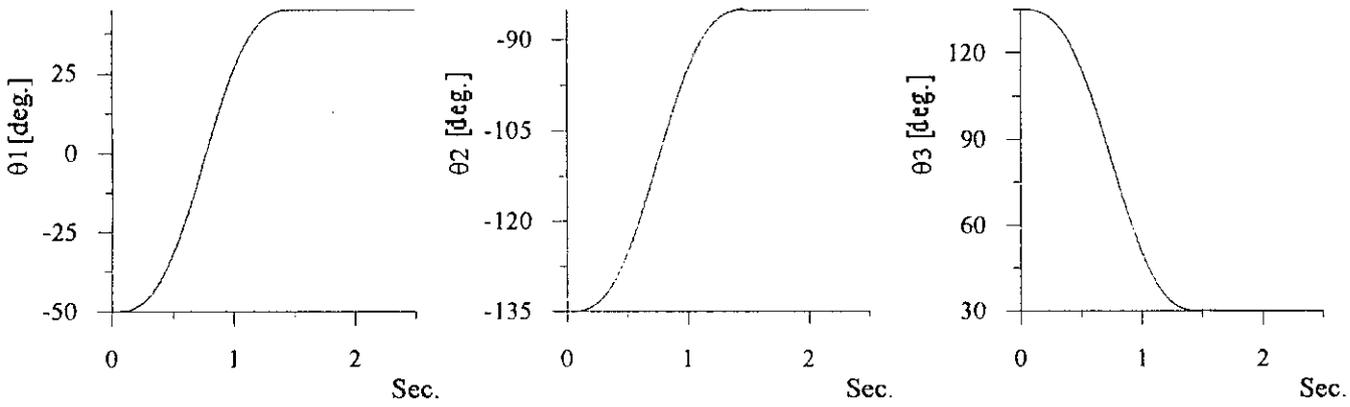
### Résultats de Simulation

Nous avons opté pour des RNAR chacun a trois couches, la couche cachée possède 5 neurones et  $n_y=2$  (deux retours de la sortie).

L'efficacité du schéma de commande proposé à été démontré par l'application de la commande en poursuite de trajectoire d'un robot manipulateur de type PUMA 560. Les figures (1.22) et (1.23) illustrent les résultats obtenus pour les tests utilisés précédemment.

Nous constatons que les erreurs de poursuite sont très faibles et les perturbations n'affectent pas les performances de poursuite ce qui confirme l'efficacité de l'algorithme d'apprentissage utilisé, double rétropropagation, pour l'adaptation en ligne des poids des réseaux récurrents.





## **I.8 CONCLUSION**

Dans ce chapitre, nous avons donné une description générale sur différent type de RNA que nous avons utilisé par la suite dans la commande adaptative directe, en se basant sur les développements théoriques récents permettant l'utilisation des lois d'adaptation qui assurent la stabilité du système global. A travers les résultats obtenus par application au robot PUMA 560, nous avons constaté que l'utilisation des réseaux RBFG à centres fixes exige un nombre très élevé de neurones dans la couche cachée et pour remédier à cet handicap, nous avons utilisé un autre type de réseau RBFG dont les centres sont adaptatifs. De plus, un autre type de RNA faisant partie de la famille des réseaux dynamiques a été utilisé, c'est un réseau dynamique bouclé appelé réseau récurrent. Les poids de ce réseau sont adaptés par la rétropropagation double (double-backpropagation)

D'une part, nous constatons que l'utilisation des RNA à trois couches ; statique, RBFG à centres fixes et RBFG à centres adaptatifs garantit la stabilité globale du système et donne des performances de poursuite ainsi qu'une robustesse remarquables. Par ailleurs, des résultats similaires ont été obtenus par les RNA récurrents. D'autre part, l'usage de la commande neuronale prédictive adaptative a donné naissance à une réduction notable du temps de calcul en ligne.

---

# *Chapitre 2*

## *Commande Adaptative Décentralisée*

### *Directe par RNA*

---

### II.1 Introduction

La plupart des résultats de la commande adaptative décentralisée des systèmes non linéaires complexes sont obtenus pour les systèmes avec interconnexions du premier ordre bornées [JAI 95a]. Ces résultats ne peuvent pas garantir la stabilité quand les interconnexions entre les sous-systèmes sont d'ordre élevés. Dans [SER 89], les contrôleurs adaptatifs décentralisés pour les robots manipulateurs sont conçus sous l'hypothèse que l'interconnexion due aux effets de coriolis et centrifuges est lentement variable dans le temps. Cependant, pour la poursuite des trajectoires rapides cette hypothèse n'est pas vérifiée ce qui peut ramener le système au régime d'instabilité. Parmi les résultats obtenus sur la stabilisation décentralisée globale des systèmes à grande échelle avec des interconnexions d'ordre élevé, on peut citer les travaux de Shi et Singh [SHI 92] dont les interconnexions sont supposées bornées par un polynôme d'état d'ordre quelconque.

Dans le présent chapitre nous proposons une extension au cas décentralisé des méthodes présentées au chapitre précédent. Les schémas proposés présentent deux principaux avantages. D'une part, l'utilisation de la commande à caractère adaptatif ainsi que l'intelligence artificielle (RNA) rend la connaissance du modèle dynamique du robot, de ces paramètres et de la charge n'est pas obligatoire dans le cas d'une implémentation en temps réel. D'autre part, la structure décentralisée rend l'implémentation sur une architecture parallèle possible, et le temps de calcul sera considérablement réduit par rapport aux mêmes algorithmes à structure centralisée à cause de la simplification des algorithmes.

### II.2 Commande Adaptative Directe Décentralisée par RNA Statique

#### II.2.1 Position du Problème

Considérons un bras manipulateur à  $n$  degrés de libertés, ayant une dynamique décrite par l'équation (I.1).

L'expression matricielle de la fonction couple non linéaire, donnée par l'équation (I.3), est

$$f(x) = [f_1(x), \dots, f_n(x)] \quad (\text{II.1})$$

où  $f_i(x)$  est la fonction couple non linéaire de la  $i^{\text{ème}}$  articulation, et qui dépend de tous les états du système.

Dans le chapitre précédent, nous avons présenté la commande adaptative directe dans laquelle, nous estimons en ligne la fonction couple non linéaire du robot. Afin d'étendre cette technique de commande pour le cas décentralisé, nous allons compenser d'une façon décentralisée cette fonction par les termes non linéaires de la loi de commande sans modifier sa structure. C'est dans cette voie, que Shi & Singh [Shi 92] ont porté leurs contributions. Le premier problème qui se pose chronologiquement, est d'obtenir une loi de commande partiellement centralisée, puis essayer de décentraliser tout l'algorithme de calcul de la commande. Dans un premier lieu, Shi & Singh ont proposé une commande qui comporte une composante décentralisée et un compensateur (signal auxiliaire) centralisé, puis une décentralisation totale de ce dernier dans le cas général (interconnexion d'ordre quelconque).

### II.2.2 Commande avec Terme Robuste partiellement Centralisé

L'idée d'utiliser cette stratégie de progression dans la décentralisation des commandes n'est pas nouvelle. En effet, Ossman [OSS 89] a proposé une commande adaptative indirecte décentralisée, mais qui utilise, pour l'estimation des paramètres, un algorithme d'adaptation paramétrique centralisé. On trouve aussi dans [BEN 88] la possibilité d'utiliser des commandes locales avec des estimateurs adaptatifs globaux.

On suppose la fonction couple non linéaire de la  $i^{\text{ème}}$  articulation donnée par l'équation (II.1), soit :

$$f_i(x) = f_i^*(x_i) + \alpha_i(t, x) \quad (\text{II.2})$$

où  $f_i^*(x_i)$  est une fonction couple non linéaire de la  $i^{\text{ème}}$  articulation et ne dépend que des états de la  $i^{\text{ème}}$  articulation, et le terme  $\alpha_i(t, x)$  est considéré comme une perturbation extérieure ou une interaction entre le sous système  $i$  et les autres sous systèmes.

On suppose l'existence des poids idéaux  $W_i$  et  $V_i$  tel que :

$$f_i^*(x_i) = W_i^T \sigma(V_i^T x_i) + \varepsilon_i(x_i) \quad (\text{II.3})$$

son estimé par un RNA est donné par :

$$\hat{f}_i^*(x_i) = \hat{W}_i^T \sigma(\hat{V}_i^T x_i) \quad (\text{II.4})$$

ou  $\varepsilon_i$  est l'erreur de reconstruction.

Le terme  $\alpha_i(t, x)$  est supposé borné par le produit des éléments d'un vecteur  $\varphi(x)$  connu et des paramètres  $\xi_{ij}$  inconnus [VES 97], ie :

$$\|\alpha_i(t, x)\| \leq \sum_{j=1}^n \varphi_j^T(x_j) \xi_{ij} \quad (\text{II.5})$$

avec  $\varphi_j(x_j) = \sum_{k=1}^p \|x_j\|^k$

En développant l'équation (II.5), on obtient

$$\begin{aligned} \|\alpha_i(t, x)\| &\leq \sum_{k=0}^p \sum_{j=1}^n \xi_{ij}^k \|x_j\|^k \leq \zeta_{\max} \sum_{k=0}^p \sum_{j=1}^n (\|e_j\| + \|x_{mj}\|)^k \\ &\leq \zeta_{\max} \sum_{k=0}^p \sum_{j=1}^n s_{ij}^k \|e_j\|^k \end{aligned}$$

$$\text{avec } \zeta_{\max} = \max_{i,j,k} (\xi_{ij}^k), s_{ij}^k = \sum_{l=k}^p C_j^{j-l} \|x_{mj}\|^{j-l} \text{ et } C_j^{j-l} = \frac{j!}{l!(j-l)!}.$$

La loi de commande appliquée au  $i^{\text{ème}}$  sous système est dite décentralisée dans la mesure où chaque contrôleur local opère uniquement sur son sous système associé, sans échange d'information. Cela implique que pour chaque contrôleur adaptatif, l'algorithme d'adaptation n'utilise que les informations locales. En d'autres termes, le problème de la commande décentralisée est la synthèse d'un ensemble de commande locale qui génère à chaque instant le couple  $\tau_i(t)$  en utilisant seulement  $\theta_i$  et  $\theta_{di}$  et ses dérivées successives.

La loi de commande de la  $i^{\text{ème}}$  articulation est donnée par :

$$\tau_i(t) = \hat{W}_i^T \sigma(\hat{V}_i^T x_i) + k_{vi} r_i + v_{Ri}(x) \quad (\text{II.6})$$

- $v_{Ri}(x)$  est un compensateur à structure centralisée.
- $k_{vi} r_i$  est la sortie d'un PD classique.
- $\hat{f}_i^*(x_i)$  est l'estimation de la fonction  $f_i^*(x_i)$  par un réseau de neurone à trois couches.

En utilisant cette loi de commande dans l'équation (I.6), la dynamique de l'erreur filtrée sera :

$$M\dot{r} = -(k_v + V_m)r + [\phi_1 \dots \phi_n]^T \quad (\text{II.7})$$

avec

$$\phi_i = f_i^*(x_i) - \hat{f}_i^*(x_i) + \alpha_i(t, x) - v_{Ri}, \quad i = 1, \dots, n \quad (\text{II.8})$$

utilisons respectivement les équations (II,3), (II,4), (I,15) et (I,16) dans (II,8), on obtient :

$$\phi_i = \tilde{W}_i^T \hat{\sigma}_i + \hat{W}_i^T \hat{\sigma}_i' \tilde{V}_i^T x_i + w_i - v_{Ri}, \quad i = 1, \dots, n \quad (\text{II.9})$$

avec

$$w_i = \tilde{W}_i^T \hat{\sigma}_i' \tilde{V}_i^T x_i + W_i^T O(\tilde{V}_i^T x_i)^2 + \tau_{di} + \alpha_i(t, x) + \varepsilon_i(x_i) \quad (\text{II.10})$$

### Adaptation des poids

Les lois d'ajustement en ligne des paramètres du  $i^{\text{ème}}$  réseau associé à la  $i^{\text{ème}}$  articulation sont identiques à celles obtenues dans l'approche centralisée définie par les équations (I.22a) et (I.22b), donc :

$$\dot{\hat{W}}_i = F_{w_i} \cdot \hat{\sigma}_i \cdot r_i \quad (II.11)$$

$$\dot{\hat{V}}_i = F_{v_i} \cdot x_i \cdot (\hat{\sigma}_i^T \cdot \hat{W}_i \cdot r_i)^T$$

avec:  $F_{w_i} = F_{w_i}^T > 0$ ,  $F_{v_i} = F_{v_i}^T > 0$ .

### Propriété II.1

On suppose que le terme  $w_i(t)$  de l'équation (II.10) est borné par un polynôme d'état :

$$\begin{cases} \|w_i(t)\| \leq \rho(z, \hat{W}_i, \hat{V}_i) \\ \rho(z, \hat{W}_i, \hat{V}_i) \leq c_{0i} + c_{1i} \|z\|_F + c_{2i} \|z\|^2 + c_{3i} \|\hat{W}_i\|_F + c_{4i} \|\hat{V}_i\|_F + c_{5i} \|\hat{V}_i\|_F \|\hat{W}_i\|_F = S_i \varphi_i \end{cases} \quad (II.12)$$

$c_i$  ( $i=1, \dots, 5$ ) sont des constantes positives bornées dépendant de la trajectoire désirée, des propriétés physiques du robot (i.e., masse et longueur des liaisons, coefficients de frottements, etc.), la borne de perturbation et de l'erreur et les gains de commande. Les vecteurs  $z$ ,  $S_i$  et  $\varphi_i$  utilisés dans (II.12) sont définis par :

$$\begin{aligned} z &= \begin{bmatrix} e^T & r^T \end{bmatrix} \\ S_i &= \begin{bmatrix} 1 & \|z\| & \|z\|^2 & \|\hat{W}_i\|_F & \|\hat{V}_i\|_F & \|\hat{V}_i\|_F & \|\hat{W}_i\|_F \end{bmatrix} \\ \varphi_i &= \begin{bmatrix} c_{0i} & c_{1i} & c_{2i} & c_{3i} & c_{4i} & c_{5i} \end{bmatrix}^T. \end{aligned}$$

Le terme robuste  $v_{Ri}(t)$  est donné par :

$$v_{Ri}(t) = \frac{r_i(t) (S_i \hat{\varphi}_i)^2}{(S_i \hat{\varphi}_i) \|r_i(t)\| + \delta_i(t)} \quad (II.13)$$

et

$$\dot{\delta}_i(t) = -\gamma_i \delta_i(t), \quad \gamma_i > 0, \delta_i(0) : \text{constante positive.} \quad (II.14)$$

La loi d'adaptation du paramètre  $\varphi_i$  est choisie tel que :

$$\dot{\hat{\varphi}}_i = \Gamma_i S_i^T \|r_i(t)\| = -\tilde{\varphi}_i \quad (II.15)$$

avec  $\Gamma_i$  est une matrice symétrique définie positive et  $\tilde{\varphi}_i = \varphi_i - \hat{\varphi}_i$

### Etude de la stabilité

Nous considérons la fonction candidate de Lyapunov suivante :

$$L(t) = \frac{1}{2} r^T M r + \frac{1}{2} e^T K_p e + \frac{1}{2} \sum_{i=1}^n \text{tr}(\tilde{W}_i^T F_{w_i}^{-1} \tilde{W}_i) + \frac{1}{2} \sum_{i=1}^n \text{tr}(\tilde{V}_i^T F_{v_i}^{-1} \tilde{V}_i) + \sum_{i=1}^n L_{Ri} \quad (II.16)$$

avec  $L_{Ri} = \frac{1}{2} \tilde{\varphi}_i^T \Gamma_i^{-1} \tilde{\varphi}_i + \frac{\delta_i}{\gamma_i}$ .

dérivons cette fonction définie positive, on obtient :

$$\dot{L}(t) = r^T M \dot{r} + e^T K_p \dot{e} + \sum_{i=1}^n \text{tr}(\tilde{W}_i^T F_{W_i}^{-1} \dot{\tilde{W}}_i) + \frac{1}{2} \sum_{i=1}^n \text{tr}(\tilde{V}_i^T F_{V_i}^{-1} \dot{\tilde{V}}_i) + \sum_{i=1}^n \dot{L}_{Ri} \quad (\text{II.17})$$

En utilisant respectivement les équations (II.7), (II.9), (II.10) et (II.11), cette équation devient :

$$\dot{L} \leq -z^T Q z + \sum_{i=1}^n (\dot{L}_{1i} + \dot{L}_{2i}) \quad (\text{II.18})$$

avec

$$z = \begin{bmatrix} e^T & r^T \end{bmatrix}^T,$$

$$Q = \begin{bmatrix} K_p \Lambda & -\frac{K_p}{2} \\ -\frac{K_p}{2} & K_v \end{bmatrix} > 0 \text{ pour un choix approprié des matrices diagonales des gains } K_p \text{ et } K_v,$$

$$\dot{L}_{1i} = \|r_i\| S_i \dot{\varphi}_i - r_i v_{Ri} - \delta_i, \quad (\text{II.19a})$$

$$\dot{L}_{2i} = \text{tr}[\tilde{W}_i^T F_{W_i}^{-1} \dot{\tilde{W}}_i + \tilde{W}_i \dot{\sigma}_i r_i] + \text{tr}[\tilde{V}_i^T F_{V_i}^{-1} \dot{\tilde{V}}_i + \tilde{V}_i^T x_i (\dot{\sigma}_i^T r_i)^T] \quad (\text{II.19b})$$

utilisons (II.13) dans (II.19a), on obtient

$$\begin{aligned} \dot{L}_{1i} &= \|r_i\| S_i \dot{\varphi}_i - \frac{r_i r_i^T (S_i \dot{\varphi}_i)^2}{(S_i \dot{\varphi}_i) \|r_i\| + \delta_i} - \delta_i \\ &\leq \frac{\delta_i \|r_i\| S_i \dot{\varphi}_i}{(S_i \dot{\varphi}_i) \|r_i\| + \delta_i} - \delta_i \\ &\leq 0. \end{aligned} \quad (\text{II.20})$$

La substitution des lois d'adaptation des poids (II.11) dans (II.19b) donne

$$\dot{L}_{2i} = 0 \quad (\text{II.21})$$

combinons les résultats ci-dessus, on obtient

$$\dot{L} \leq -z^T Q z, \quad Q > 0 \quad (\text{II.22})$$

(II.22) implique  $L_i$ ,  $z_i$ ,  $e_i$ ,  $r_i$ ,  $\tilde{W}_i$ ,  $\tilde{V}_i$ , et  $\dot{\varphi}_i$ ,  $i$  allant de 1 à  $n$ , sont tous bornés. De la forme de (II.22), il est facile de montrer que le carré de  $z(t)$  est intégrable; d'où les carrés de  $e$  et  $r$  sont intégrables. De l'équation de l'erreur en boucle fermée (II.7), il est aussi facile de montrer que  $\dot{r}$  est borné. Par conséquent,  $r$  est uniformément continu. Par la lemme de Barbalat [KWA 95b], nous savons que  $e$  et  $\dot{e}$  tendent vers zéro asymptotiquement.

### II.2.3 Commande avec Terme Robuste Décentralisé (Interconnexion d'ordre quelconque)

L'hypothèse (éq. II.5) utilisée pour caractériser la nature des interconnexions à permet la synthèse d'une commande décentralisée avec un terme robuste centralisé garantissant la bornitude globale du système. Cependant, la question qui se pose à ce niveau est : peut on garantir la stabilité en utilisant un terme robuste décentralisé afin d'atteindre notre objectif ; commande complètement décentralisée (fig.II.1).

Dans cette section, cette possibilité est illustrée pour des interconnexions d'ordre quelconque ( $p > 1$ ), en se basant sur les travaux de [JAI 97a], [JAI 97b].

**Remarque :** Les lois d'adaptation des poids des RNA sont les mêmes que celles de la section précédente.

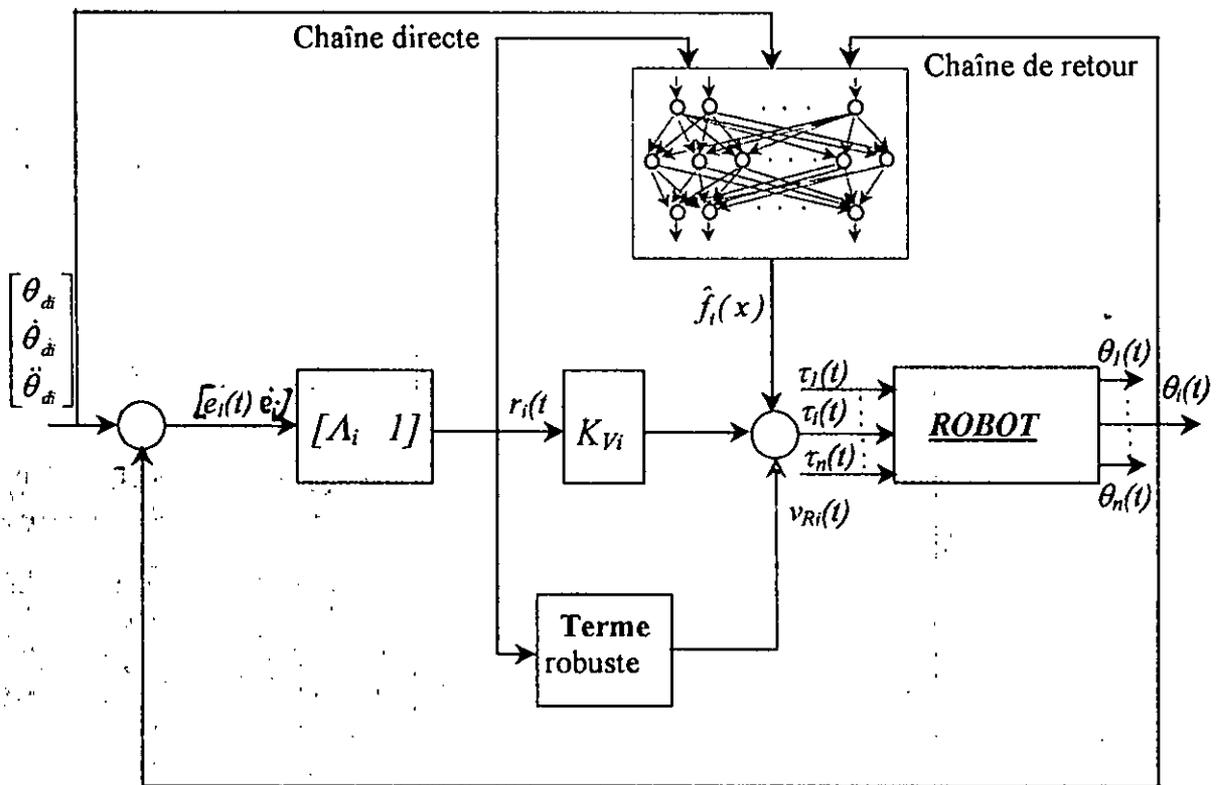


Figure II.1 structure de base du système de commande adaptatif décentralisé par RNA.

#### Propriété II.2

D'après la propriété II.1 et l'équation II.5 (interconnexions d'ordre quelconque), le terme  $w_i(t)$  de l'équation (II.10) est borné par :

$$\begin{cases} \|w_i(t)\| \leq \rho(z, \hat{W}_i, \hat{V}_i) \\ \rho(z, \hat{W}_i, \hat{V}_i) \leq \left\{ \sum_{k=0}^p \sum_{j=1}^n (c_{ij}^k \|z_j\|^k) \right\} + c_{i_{n+1}}^{p+1} \|\hat{W}_i\|_F + c_{i_{n+1}}^{p+2} \|\hat{V}_i\|_F + c_{i_{n+1}}^{p+3} \|\hat{V}_i\|_F \|\hat{W}_i\|_F = S_i \varphi_i \end{cases} \quad (II.23)$$

$c_{ij}^k$  ( $j=1, \dots, n+1, k=1, \dots, p+3$ ) sont des constantes positives bornées et dépendent de la trajectoire désirée, des propriétés physiques du robot (c-à-d., masse et longueur des liaisons, coefficients de frottements,

etc.), la borne de perturbation et de l'erreur et les gains de commande. Les vecteurs  $z_i$ ,  $S_i$  et  $\varphi_i$  utilisés dans (II.23) sont définis par :

$$\begin{aligned} z_j &= [e_j \quad r_j] \\ S_i &= [I \quad \|z_1\| \dots \|z_n\| \dots \|z_1\|^p \dots \|z_n\|^p \quad \|\hat{W}_i\|_F \quad \|\hat{V}_i\|_F \quad \|\hat{V}_i\|_F \quad \|\hat{W}_i\|_F] \\ \varphi_i &= [(c_{i1}^0 + \dots + c_{in}^0) \quad c_{i1}^1 \dots c_{in}^1 \dots c_{i1}^p \dots c_{in}^p \quad c_{i_{n+1}}^{p+1} \quad c_{i_{n+1}}^{p+2} \quad c_{i_{n+1}}^{p+3}]^T \end{aligned} \quad (II.24)$$

Le terme robuste  $v_{Ri}(t)$  est donné par :

$$v_{Ri}(t) = \frac{r_i(t)(S_i^* \hat{\varphi}_i^*)^2}{(S_i^* \hat{\varphi}_i^*) \|r_i(t)\| + \delta_i(t)} \quad (II.25)$$

Les vecteurs  $z_i$ ,  $S_i^*$  et  $\varphi_i^*$  utilisés dans (II.25) sont définis par :

$$\begin{aligned} S_i^* &= [I \quad \|z_1\| \dots \|z_1\|^p \quad \|\hat{W}_i\|_F \quad \|\hat{V}_i\|_F \quad \|\hat{V}_i\|_F \quad \|\hat{W}_i\|_F] \\ \varphi_i^* &= [c_i^0 \quad c_i^1 \dots c_i^p \quad c_i^{p+1} \quad c_i^{p+2} \quad c_i^{p+3}]^T \end{aligned} \quad (II.26)$$

et

$$\dot{\delta}_i(t) = -\gamma_i \delta_i(t), \quad \gamma_i > 0, \delta_i(0) : \text{constante positive.} \quad (II.27)$$

La loi d'adaptation du paramètre  $\varphi_i$  est choisie tel que :

$$\dot{\hat{\varphi}}_i^* = \Gamma_i S_i^{*T} |r_i(t)| = -\tilde{\varphi}_i^* \quad (II.28)$$

avec  $\Gamma_i$  est une matrice symétrique définie positive et  $\tilde{\varphi}_i^* = \varphi_i^* - \hat{\varphi}_i^*$ .

**Etude de la stabilité**

Nous considérons la fonction candidate de Lyapunov donnée par l'équation (II.16). En utilisant les mêmes étapes de la section précédente, on obtient :

$$\dot{L} \leq -z^T Qz + \sum_{i=1}^n (\dot{L}_{1i} + \dot{L}_{2i}) \quad (II.29)$$

$$\dot{L}_{1i} = |r_i| S_i \varphi_i - r_i v_{Ri} - \delta_i, \quad (II.30)$$

$$\dot{L}_{2i} = tr[\tilde{W}_i^T F_{W_i}^{-1} \dot{\tilde{W}}_i + \tilde{W}_i \hat{\sigma}_i r_i] + tr[\tilde{V}_i^T F_{V_i}^{-1} \dot{\tilde{V}}_i + \tilde{V}_i^T x_i (\hat{\sigma}_i^T r_i)^T] \quad (II.31)$$

utilisons (II.23) dans (II.30), on obtient

$$\dot{L}_i = \sum_{j=1}^n \dot{L}_{1j} = \sum_{j=1}^n |r_j| (S_j \hat{\varphi}_j) - r_j v_{Rj} - \delta_j \quad (II.32)$$

$$\dot{L}_1 = \sum_{i=1}^n |r_i| \left\{ \sum_{k=0}^p \sum_{j=1}^n (c_{ij}^k \|z_j\|^k) \right\} + c_{i_{n+1}}^{p+1} \|\hat{W}_i\|_F + c_{i_{n+1}}^{p+2} \|\hat{V}_i\|_F + c_{i_{n+1}}^{p+3} \|\hat{V}_i\|_F \|\hat{W}_i\|_F - r_i v_{Ri} - \delta_i \quad (\text{II.33})$$

$$\dot{L}_1 = \sum_{i=1}^n \left\{ \frac{1}{2} |r_i| \sum_{j=1}^n \sum_{k=1}^p 2 \zeta_j^{-j} \zeta_j^{-j} (c_{ij}^k \|z_j\|^k) + |r_i| (c_{i1}^0 + \dots + c_{in}^0) + c_{i_{n+1}}^{p+1} \|\hat{W}_i\|_F \right. \\ \left. + c_{i_{n+1}}^{p+2} \|\hat{V}_i\|_F + c_{i_{n+1}}^{p+3} \|\hat{V}_i\|_F \|\hat{W}_i\|_F - r_i v_{Ri} - \delta_i \right\} \quad (\text{II.34})$$

L'utilisation de l'inégalité Cauchy-Schwartz suivante :

$$2ab \leq a^2 + b^2 \quad (\text{II.35})$$

donne

$$\dot{L}_1 = \sum_{i=1}^n \left\{ \frac{1}{2} |r_i| \sum_{j=1}^n \sum_{k=1}^p \left\{ \zeta_j^{-j} + \zeta_j (c_{ij}^k \|z_j\|^k)^2 \right\} + |r_i| (c_{i1}^0 + \dots + c_{in}^0) + c_{i_{n+1}}^{p+1} \|\hat{W}_i\|_F \right. \\ \left. + c_{i_{n+1}}^{p+2} \|\hat{V}_i\|_F + c_{i_{n+1}}^{p+3} \|\hat{V}_i\|_F \|\hat{W}_i\|_F - r_i v_{Ri} - \delta_i \right\} \quad (\text{II.36})$$

et nous avons :

$$\dot{L}_1 = \sum_{i=1}^n \left\{ \frac{1}{2} |r_i| \sum_{j=1}^n \sum_{k=1}^p \left\{ \zeta_j^{-j} + \zeta_j (c_{ij}^k)^2 (\|z_j\|^{2k}) \right\} + |r_i| (c_{i1}^0 + \dots + c_{in}^0) + c_{i_{n+1}}^{p+1} \|\hat{W}_i\|_F \right. \\ \left. + c_{i_{n+1}}^{p+2} \|\hat{V}_i\|_F + c_{i_{n+1}}^{p+3} \|\hat{V}_i\|_F \|\hat{W}_i\|_F - r_i v_{Ri} - \delta_i \right\} \quad (\text{II.37})$$

$$\dot{L}_1 = \sum_{i=1}^n \left\{ \frac{1}{2} |r_i| \sum_{k=1}^p \left\{ \left( \sum_{j=1}^n \zeta_j^{-j} \right) + \zeta_i \left( \sum_{j=1}^n (c_{ij}^k)^2 \right) (\|z_i\|^{2k}) \right\} + |r_i| (c_{i1}^0 + \dots + c_{in}^0) + c_{i_{n+1}}^{p+1} \|\hat{W}_i\|_F \right. \\ \left. + c_{i_{n+1}}^{p+2} \|\hat{V}_i\|_F + c_{i_{n+1}}^{p+3} \|\hat{V}_i\|_F \|\hat{W}_i\|_F - r_i v_{Ri} - \delta_i \right\} \quad (\text{II.38})$$

posons

$$d_{ik} = \sum_{j=1}^n (c_{ij}^k)^2 \quad (\text{II.39})$$

$$\zeta_i^* = \sum_{j=1}^n \zeta_j^{-j}$$

on obtient,

$$\dot{L}_1 \leq \sum_{i=1}^n \left\{ \frac{1}{2} |r_i| \sum_{k=1}^p \left\{ \zeta_i^* + \zeta_i d_{ik} (\|z_i\|^{2k}) \right\} + |r_i| (c_{i1}^0 + \dots + c_{in}^0) + c_{i_{n+1}}^{p+1} \|\hat{W}_i\|_F \right. \\ \left. + c_{i_{n+1}}^{p+2} \|\hat{V}_i\|_F + c_{i_{n+1}}^{p+3} \|\hat{V}_i\|_F \|\hat{W}_i\|_F - r_i v_{Ri} - \delta_i \right\} \quad (\text{II.40})$$

Maintenant il suffit de regrouper les constantes pour avoir la même forme de la section précédente. D'où,

$$\begin{aligned} \dot{L}_{11} &\leq \|r_i\| S_i^* \hat{\phi}_i^* - \frac{r_i r_i (S_i^* \hat{\phi}_i^*)^2}{(S_i^* \hat{\phi}_i^*) \|r_i\| + \delta_i} - \delta_i \\ &\leq \frac{\delta_i \|r_i\| S_i^* \hat{\phi}_i^*}{(S_i^* \hat{\phi}_i^*) \|r_i\| + \delta_i} - \delta_i \\ &\leq 0. \end{aligned} \quad (\text{II.41})$$

$$\Rightarrow \dot{L}_1 = \sum_{i=1}^n \dot{L}_{11} \leq 0 \quad (\text{II.42})$$

La substitution des lois d'adaptation des poids (II.11) dans (II.31) donne

$$\dot{L}_{2i} = 0 \quad (\text{II.43})$$

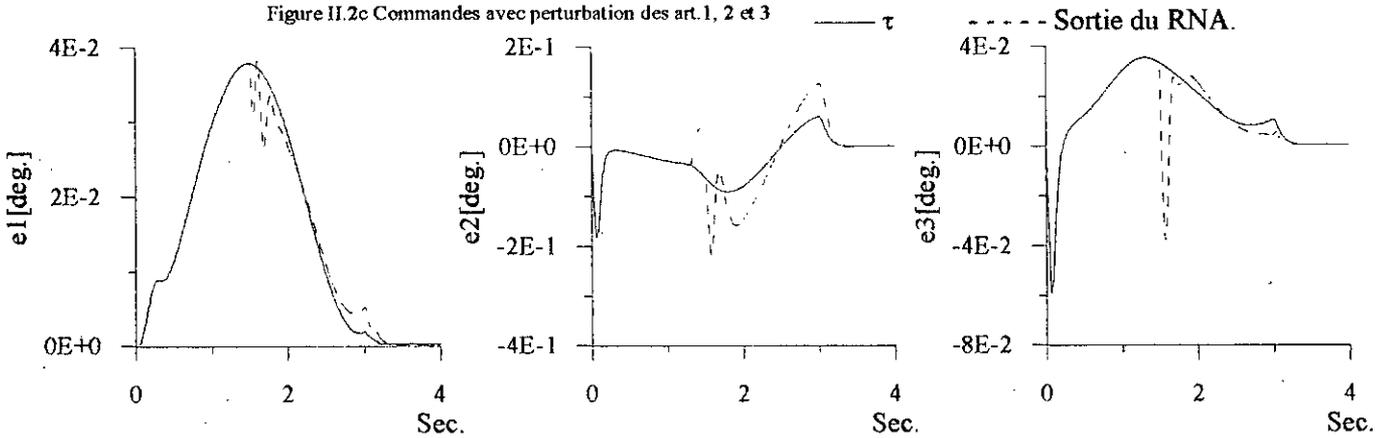
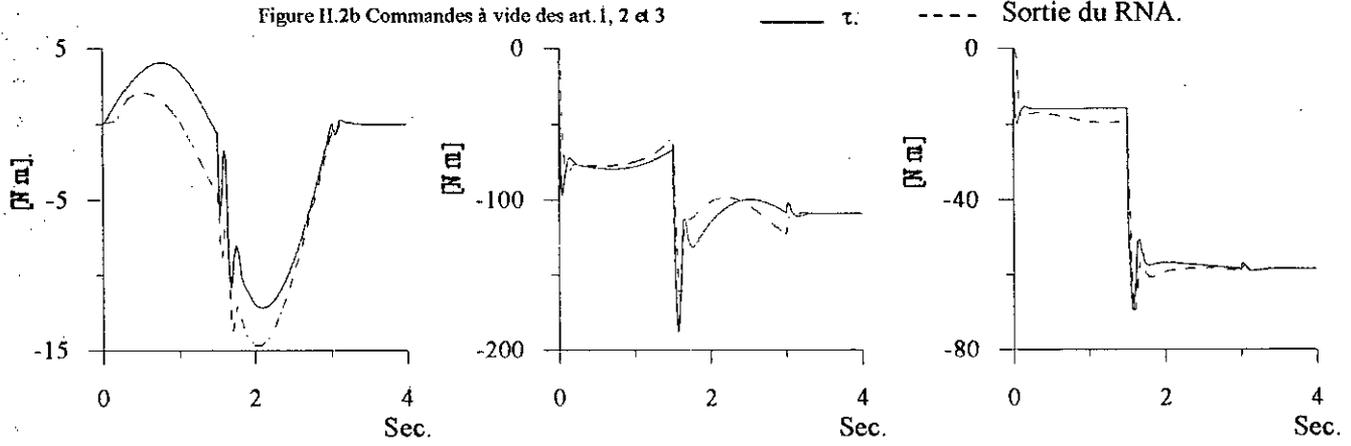
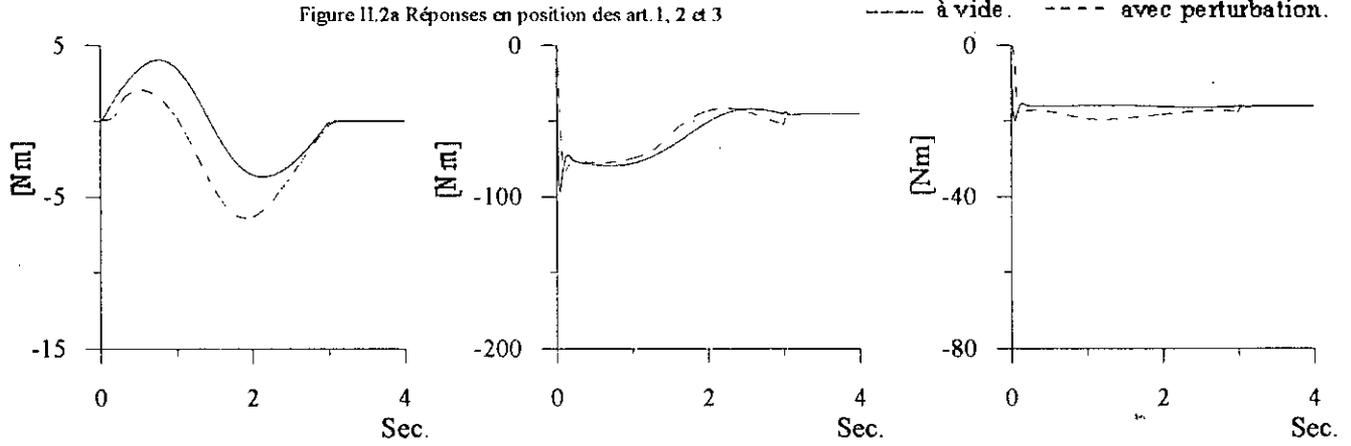
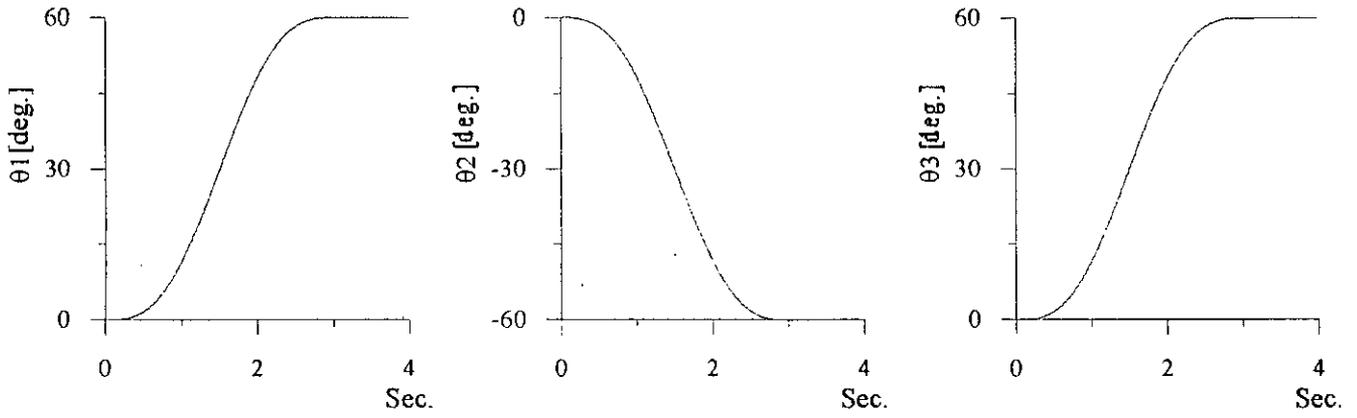
Et combinant les résultats ci-dessus, on obtient

$$\dot{L} \leq -z^T Q z, \quad Q > 0 \quad (\text{II.44})$$

(II.44) implique  $L_i$ ,  $z_i$ ,  $e_i$ ,  $r_i$ ,  $\hat{W}_i$ ,  $\hat{V}_i$ , et  $\hat{\phi}_i$ ,  $i$  allant de 1 à  $n$ , sont tous bornés. De la forme de (II.44), il est facile de montrer que le carré de  $z(t)$  est intégrable; d'où les carrés de  $e$  et  $r$  sont intégrables. De l'équation de l'erreur en boucle fermée (II.7), il est aussi facile de montrer que  $\dot{r}$  est borné. Par conséquent,  $r$  est uniformément continu. Par la lemme de Barbalat [KWA 95b], nous savons que  $e$  et  $\dot{e}$  tendent asymptotiquement vers zéro.

### Etude par simulation

Les résultats de simulation pour la trajectoire normale et la trajectoire de Leahy sans et avec perturbation sont respectivement présentés sur les figures (II.2) et (II.3). Nous constatons que la poursuite est satisfaisante et les erreurs de poursuite sont très faibles.



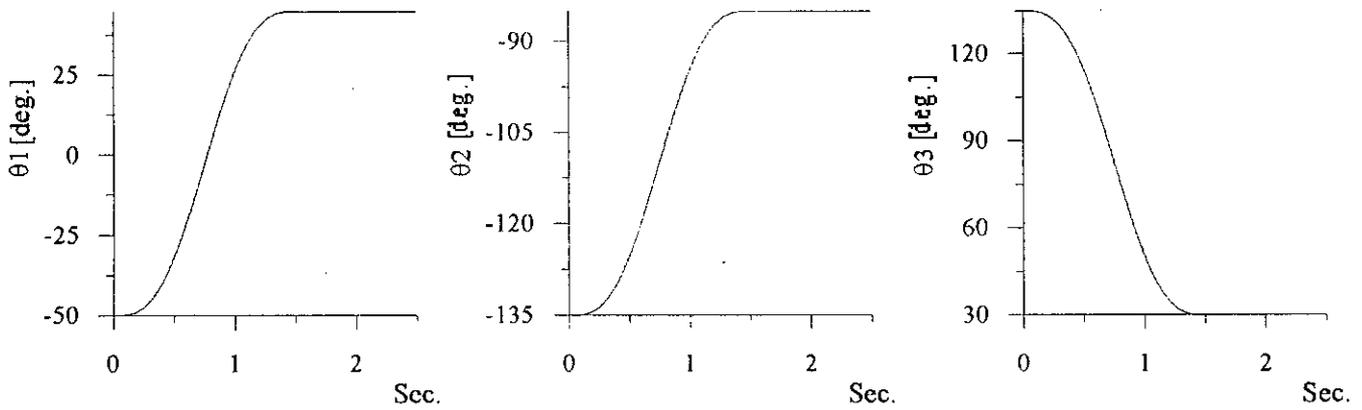


Figure II.3a Réponses en position de l'art. 1, 2 et 3

----- à vide. - - - - avec perturbation.

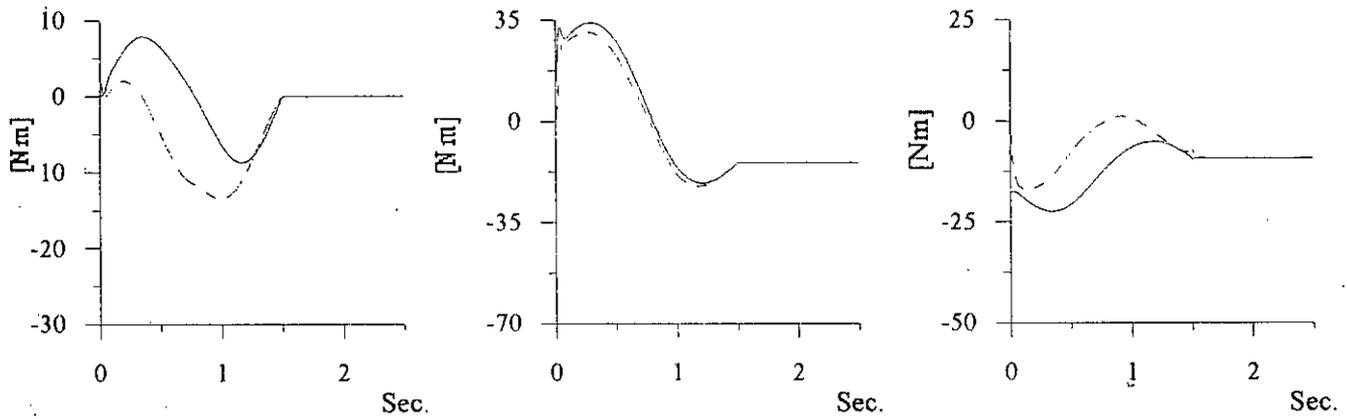


Figure II.3b Commandes à vide de l'art. 1, 2 et 3

-----  $\tau$ . - - - - Sortie du RNA.

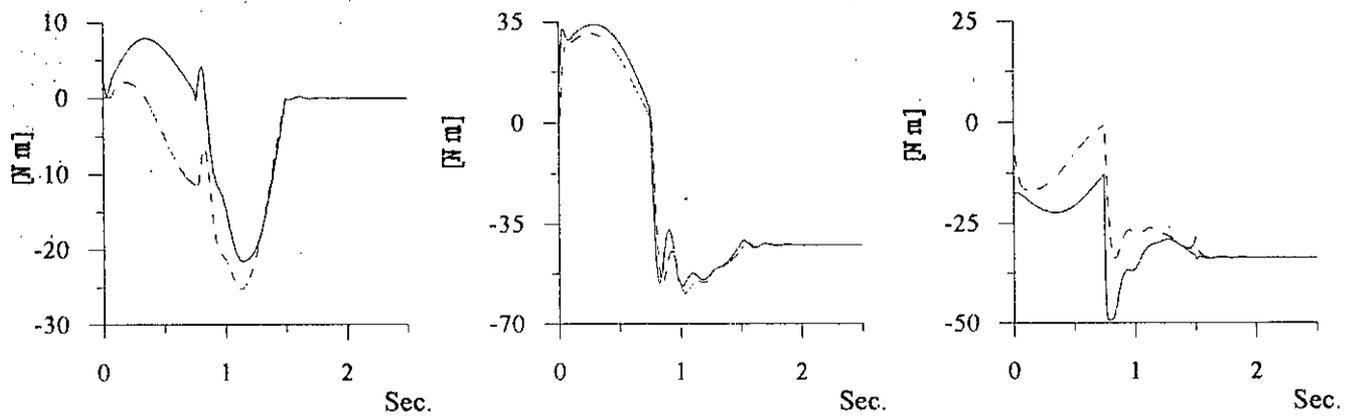


Figure II.3c Commandes avec perturbation de l'art. 1, 2 et 3

-----  $\tau$ . - - - - Sortie du RNA.

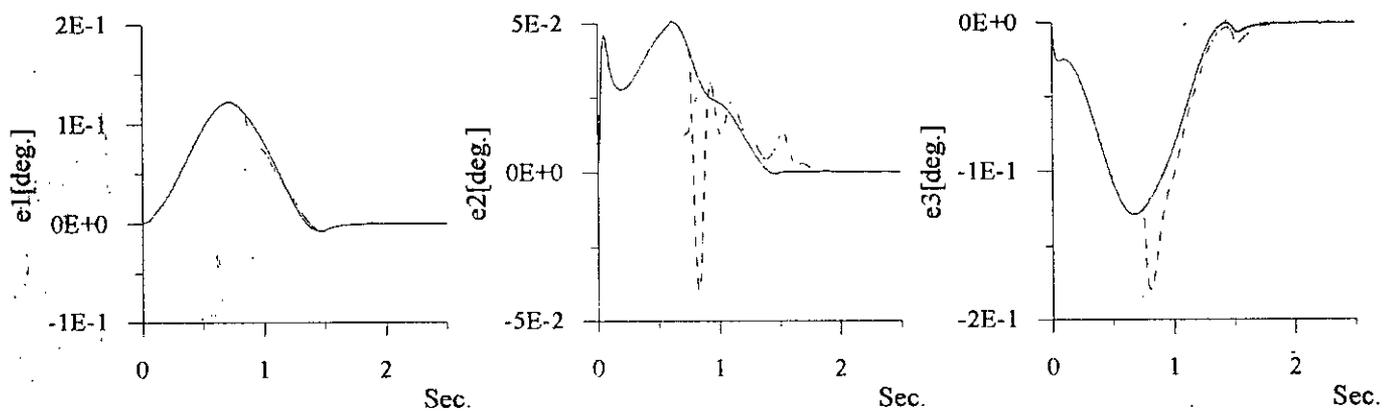


Figure II.3d Erreurs de l'art. 1, 2 et 3

----- à vide. - - - - avec perturbation.

### II.3 Commande Prédictive Adaptative Décentralisée Directe par RNA Statique

Motiver par les résultats obtenus par la technique présentée dans la section (I.3), l'extension au cas décentralisé de cette dernière fera l'objet de cette section, en utilisant les mêmes étapes de la section (II.2) précédente.

A partir de la fonction couple non linéaire désirée donnée par l'équation (I.34), on peut écrire :

$$f_{\dot{a}_i}(x) = f_{\dot{a}_i}^*(x_i) + \alpha_{\dot{a}_i}(t, x) \tag{II.45}$$

où  $x = [\theta_d \quad \dot{\theta}_d^T \quad \ddot{\theta}_d^T]^T$ , et  $x_i = [\theta_{\dot{a}_i} \quad \dot{\theta}_{\dot{a}_i}^T \quad \ddot{\theta}_{\dot{a}_i}^T]^T$ .

où  $f_{\dot{a}_i}^*(x_i)$  est la fonction couple non linéaire désirée de la  $i^{\text{ème}}$  articulation et ne dépend que des états de la  $i^{\text{ème}}$  articulation, et le terme  $\alpha_{\dot{a}_i}(t, x)$  est considéré comme une perturbation extérieure ou une interaction entre le sous système  $i$  et les autres sous systèmes qui est supposée bornée suivant l'équation (II.5).

Nous supposons l'existence des poids idéaux  $W_i$  et  $V_i$  tel que :

$$f_{\dot{a}_i}^*(x_i) = W_i^T \sigma(V_i^T x_i) + \varepsilon_i(x_i) \tag{II.46}$$

l'estime de cette dernière par un RNA statique est donné par :

$$\hat{f}_{\dot{a}_i}^*(x_i) = \hat{W}_i^T \sigma(\hat{V}_i^T x_i) \tag{II.47}$$

ou  $\varepsilon_i$  est l'erreur de reconstruction.

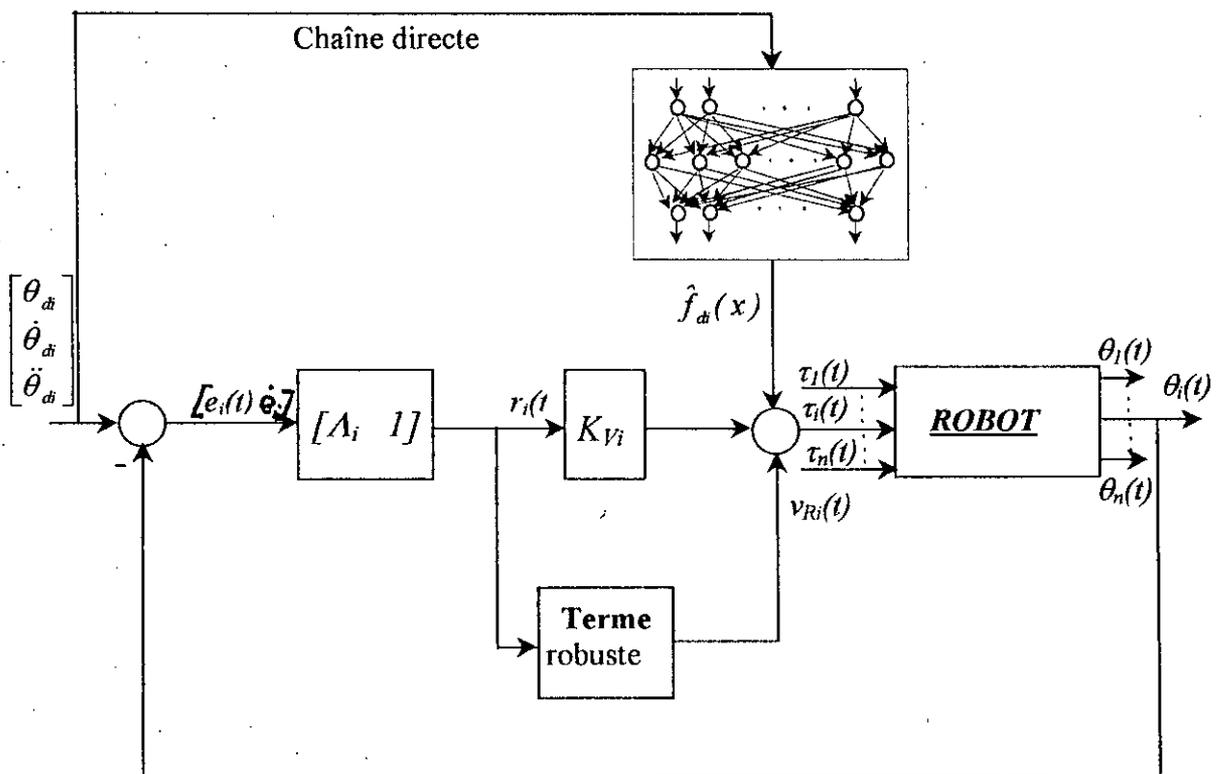


Figure II.4 structure de base du système de commande prédictif adaptatif décentralisé par RNA.

La loi de commande de la  $i^{\text{ème}}$  articulation est donnée par (fig.II.4) :

$$\tau_i(t) = \hat{f}_{\dot{a}_i}^*(x_i) + k_{v_i} r_i + v_{Ri}(x) \quad (\text{II.48})$$

- $v_{Ri}(x)$  un compensateur à structure centralisée.
- $k_{v_i} r_i$  est la sortie d'un PD classique.
- $\hat{f}_{\dot{a}_i}^*(x_i)$  l'estimation de la fonction  $f_{\dot{a}_i}^*(x_i)$  par un réseau de neurones à trois couches (éq. II.47).

En utilisant cette loi de commande dans l'équation (I.6), la dynamique de l'erreur filtrée sera :

$$M\dot{r} = -(k_v + V_m)r + [\phi_1 \dots \phi_n]^T \quad (\text{II.49})$$

avec

$$\phi_i = f_{\dot{a}_i}^*(x_i) - \hat{f}_{\dot{a}_i}^*(x_i) + (f_i^*(\cdot) - f_{\dot{a}_i}^*(x_i)) + \alpha_{\dot{a}_i}(t, x) - v_{Ri}, \quad i = 1, \dots, n \quad (\text{II.50})$$

utilisons respectivement les équations (II,45), (II,46), (II,47) et (I,16) dans (II,50), on obtient :

$$\phi_i = \tilde{W}_i^T \hat{\sigma}_i + \hat{W}_i^T \hat{\sigma}_i' \tilde{V}_i^T x_i + w_i - v_{Ri}, \quad i = 1, \dots, n \quad (\text{II.51})$$

avec

$$w_i = (f_i^*(\cdot) - f_{\dot{a}_i}^*(x_i)) + \tilde{W}_i^T \hat{\sigma}_i' \tilde{V}_i^T x_i + W_i^T O(\tilde{V}_i^T x_i)^2 + \tau_{\dot{a}_i} + \alpha_{\dot{a}_i}(t, x) + \varepsilon_i(x_i) \quad (\text{II.52})$$

ce terme regroupe les perturbations, les interconnexions ainsi que les erreurs d'estimation infiniment petites.

**Remarque :** les lois d'adaptation ainsi que l'étude de la stabilité sont les mêmes que celles de la méthode développée dans la section précédente, en utilisant le terme de perturbation définie par l'équation (II.52).

### Etude par simulation

Pour la validation de la stratégie de commande proposée, des simulations ont été effectuées sur les trois premières articulations du même robot PUMA 560. Les figures (II.5) et (II.6) montrent les performances et l'efficacité de la stratégie pour la commande en poursuite de trajectoires de références, ainsi que la robustesse vis-à-vis des perturbations.

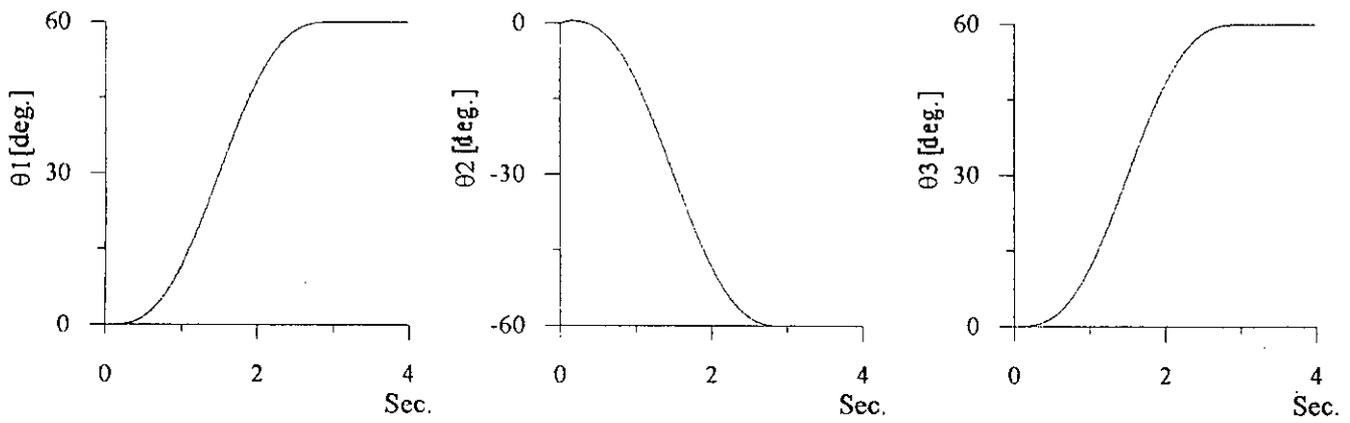


Figure II.5a Réponses en position des art. 1, 2 et 3

----- à vide. - - - - avec perturbation.

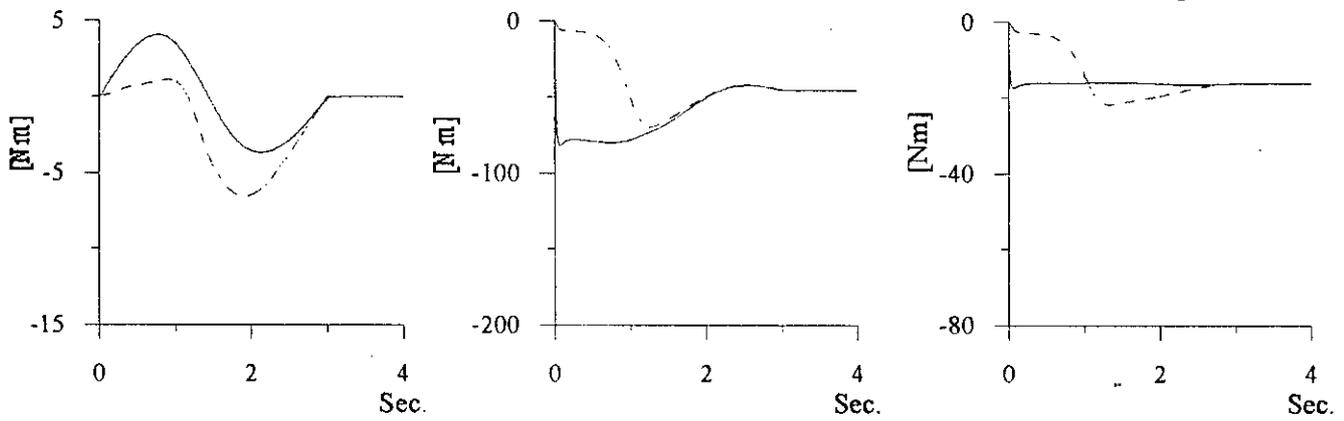


Figure II.5b Commandes à vide des art. 1, 2 et 3

———  $\tau$ . - - - - Sortie du RNA.

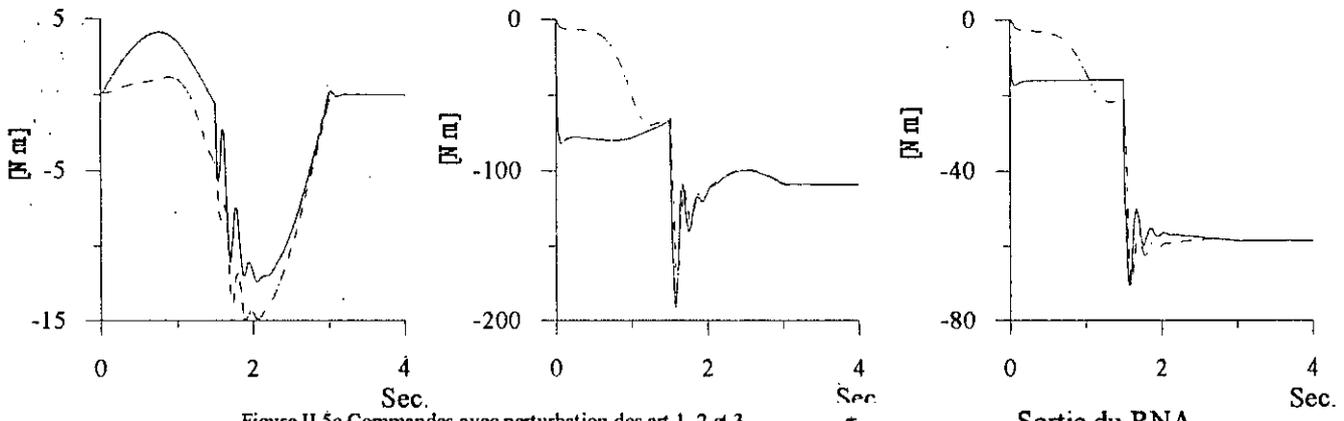


Figure II.5c Commandes avec perturbation des art. 1, 2 et 3

———  $\tau$ . - - - - Sortie du RNA.

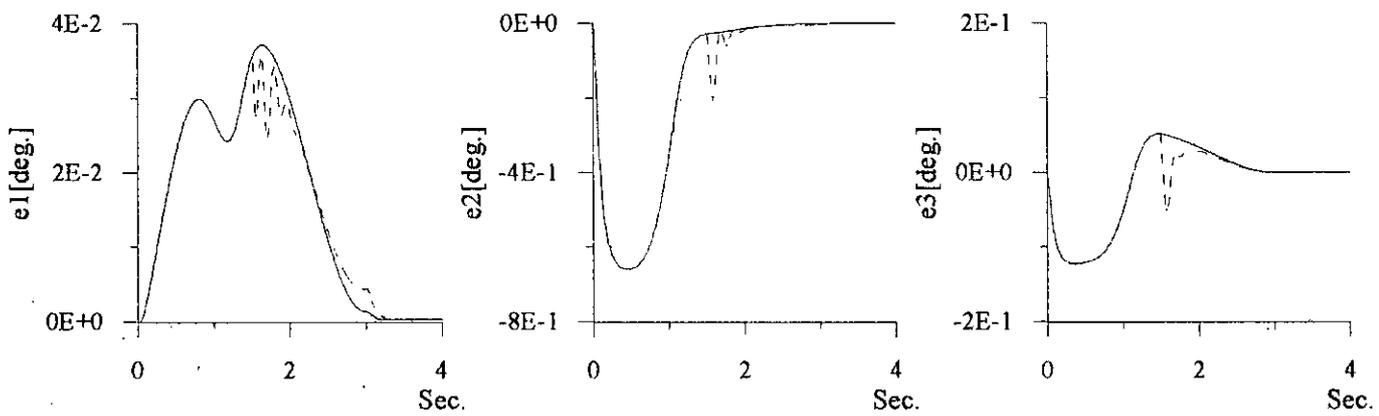
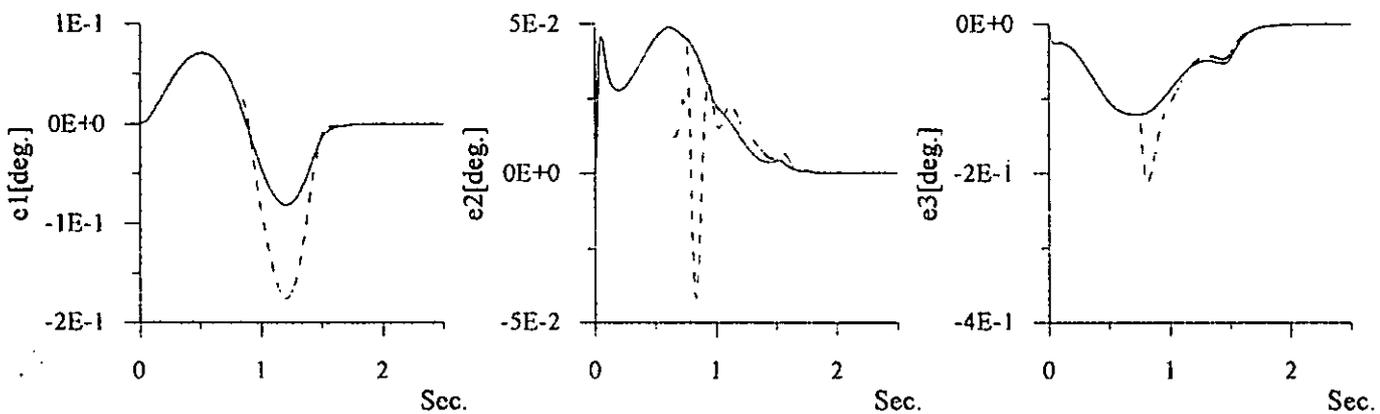
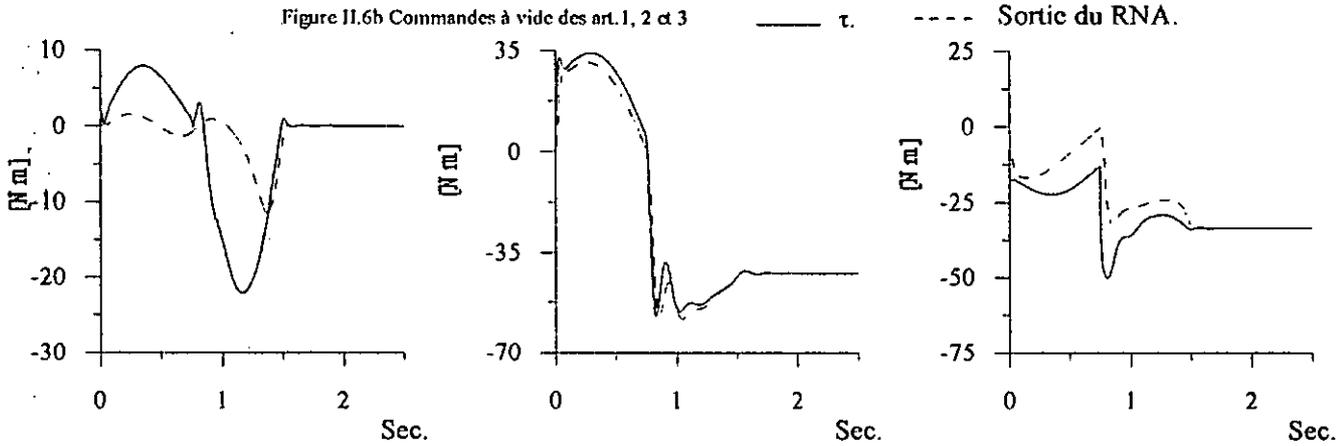
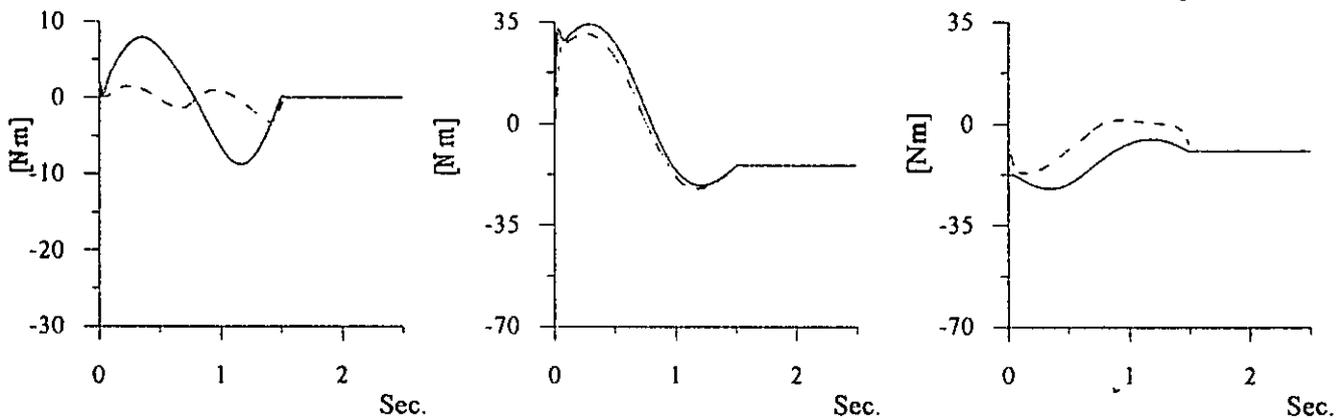
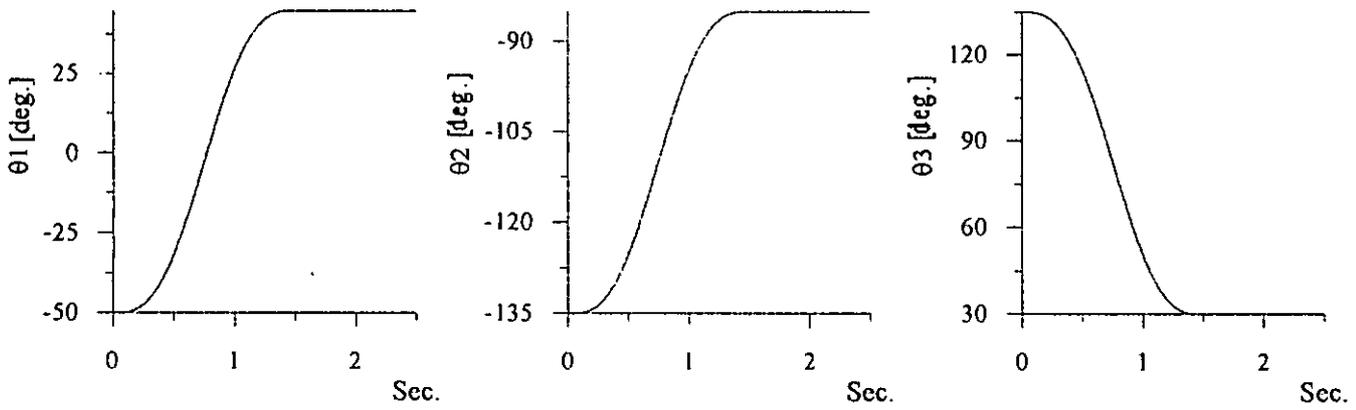


Figure II.5d Erreurs des art. 1, 2 et 3

——— à vide. - - - - avec perturbation.



## II.4 Commande Adaptative Décentralisée Directe par RBF

### II.4.1 RBF à Centres Fixes

Dans la section I.4.1b, la commande des systèmes SISO par RBF à centres fixes a été étendue au cas MIMO, pour la commande en poursuite des robots manipulateurs. En effet, cette technique de commande opère dans l'espace d'entrée ce qui impose un nombre de neurones dans la couche cachée très élevé (nombre de neurones = (nombre de subdivisions)<sup>nombre d'entrée</sup>). Afin de surmonter cet inconvénient, il faut minimiser le nombre d'entrée dont ceci peut être facilement obtenu par l'utilisation de la structure décentralisée.

Dans cette section, une commande adaptative décentralisée par RBF à centres fixes est développée par le biais de l'extension de la version centralisée présentée dans le chapitre I.

L'expression matricielle de la fonction  $f_A$  (éq. I.49) est :

$$f_A(x) = [f_{A1} \quad \dots \quad f_{An}]^T; \quad x = [\theta \quad \dot{\theta} \quad \ddot{\theta}_d]^T \quad (\text{II.53})$$

et  $f_{Ai}(x_i)$  est supposée donner sous la forme suivante :

$$f_{Ai}(x) = f_{Ai}^*(x_i) + \alpha_{Ai}(t, x) \quad (\text{II.54})$$

où  $f_{Ai}^*(x_i)$  est la fonction couple non linéaire de la  $i^{\text{ème}}$  articulation et ne dépend que des états de celle-ci,  $\alpha_{Ai}(t, x)$  est un terme regroupant les interconnexions entre le sous système  $i$  et les autres sous systèmes.

L'estimée de la fonction  $f_{Ai}^*(x_i)$  par un réseau de neurone gaussien à centres fixes est donnée par :

$$\hat{f}_{Ai}^*(x_i) = \sum_{l \in I_0} c_{il} g_{\sigma_l}(x_i - \xi_{l,i}) \quad (\text{II.55})$$

où  $\xi_{li}$  est le poids de la couche cachée reliant le neurone  $l$  par le vecteur d'entrée  $x_i$ , et  $c_{il}$  est le poids de la couche de sortie reliant le neurone  $l$  par la sortie.

Nous supposons connaître a priori la borne supérieure de  $f_i(x)$  pour les points situés à l'extérieur de l'ensemble  $A_{di}$ , c-à-d :

$$|f_i(x)| \leq M_{oi}(x) \quad \text{quand } x \in A_i^c \quad (\text{II.56})$$

En se basant sur la loi de commande donnée par l'équation (I.76), la loi de commande de la  $i^{\text{ème}}$  articulation est :

$$\tau_i(t) = K_{Di} \dot{r}_i(t) + (1 - m_i(t)) (\hat{f}_{Ai}(x_i(t)) + v_{Ri}(t)) + m_i(t) u_{si}(t) \quad (\text{II.57})$$

$K_{Di} > 0$  et  $k_{si} = M_{oi}(x(t)) - \varepsilon_{fi}$ .

En utilisant cette loi de commande et l'équation (I.75), la dynamique de l'erreur filtrée est donnée par l'équation (II.7) avec  $\phi_i$  est définie comme suit :

$$\phi_i = (1 - m_i(t)) (\tilde{f}_{Ai}^* + d_i(t) + \alpha_i(t, x) + \tau_{di} - v_{Ri}) + m_i(t) (u_{si}(t) - f_i(x) - \tau_{di}) \quad (\text{II.58})$$

avec  $\tilde{f}_{Ai}^* = \hat{f}_{Ai}^* - f_{Ai}^*$ ,  $d_i(t) = f_{Ai}(x) - f_i(x)$ .

L'ajustement des poids de sorties du  $i^{\text{ème}}$  réseau est effectué, en utilisant l'expression suivante :

$$\dot{\hat{c}}_i(t) = -k_{oi} (1 - m_i(t)) r_i(t) g_{oi}(x_i(t) - \xi_{ii}) \quad (\text{II.59})$$

$k_{oi} > 0$ , désigne le gain d'adaptation.

### Etude de la Stabilité

L'analyse de la stabilité est effectuée, en utilisant la fonction positive de Lyapunov suivante :

$$L(t) = \frac{1}{2} r^T M r + \frac{1}{2} e^T K_p e + \frac{1}{2} \sum_{i=1}^n \text{tr}(\tilde{C}_i^T K_{oi}^{-1} \tilde{C}_i) + \sum_{i=1}^n L_{Ri} \quad (\text{II.60})$$

$$\text{avec } L_{Ri} = \frac{1}{2} \tilde{\varphi}_i^T \Gamma_i^{-1} \tilde{\varphi}_i + \frac{\delta_i}{\gamma_i}.$$

La dérivée de cette fonction positive est bornée suivant l'inégalité donnée par l'équation (II.18), avec :

$$\dot{L}_{1i} = |r_i| |S_i \varphi_i - r_i v_{Ri} - \delta_i| \quad (\text{II.61})$$

$$\begin{aligned} \dot{L}_{2i} = \text{tr} [ & \tilde{C}_i^T k_{oi}^{-1} \dot{\tilde{C}}_i + \tilde{C}_i (1 - m_i(t)) r_i g_{oi}(\|x_i - \xi_{ii}\|) ] \\ & + m_i(t) r_i [ M_{oi}(x) + \tau_{oi} + u_{si}(t) ] \end{aligned} \quad (\text{II.62})$$

utilisons (II.13) dans (II.61), on obtient

$$\dot{L}_{1i} \leq 0 \quad (\text{II.63})$$

substituons la loi d'adaptation des poids (II.59) et utilisons la commande  $u_{si}(t)$  donnée par l'équation (I.52), il vient :

$$\dot{L}_{2i} \leq 0 \quad (\text{II.64})$$

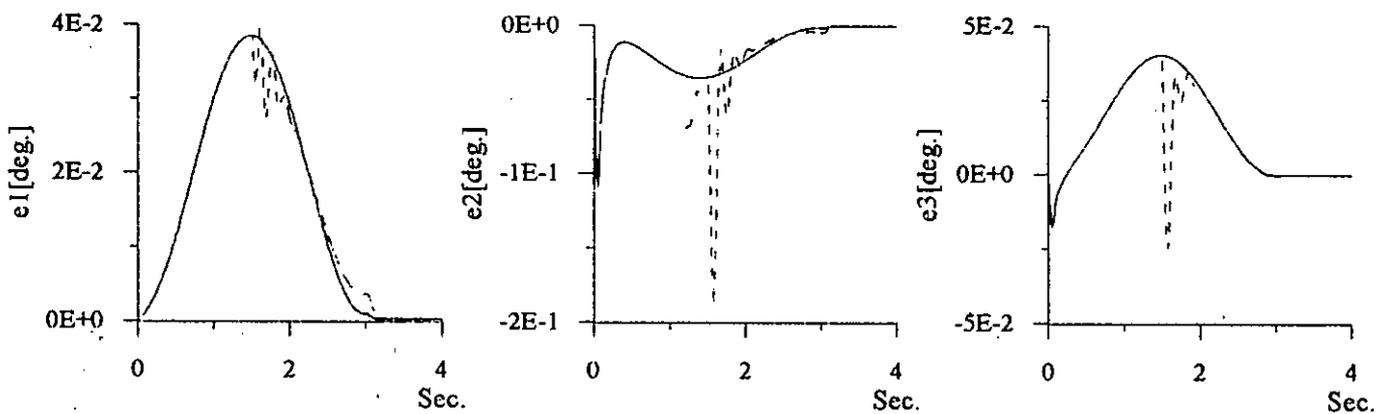
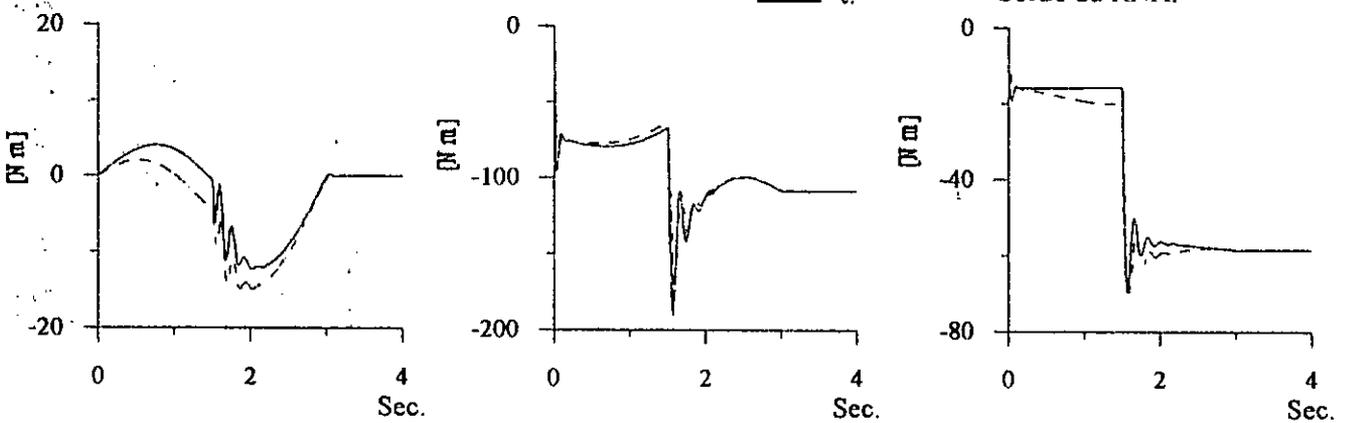
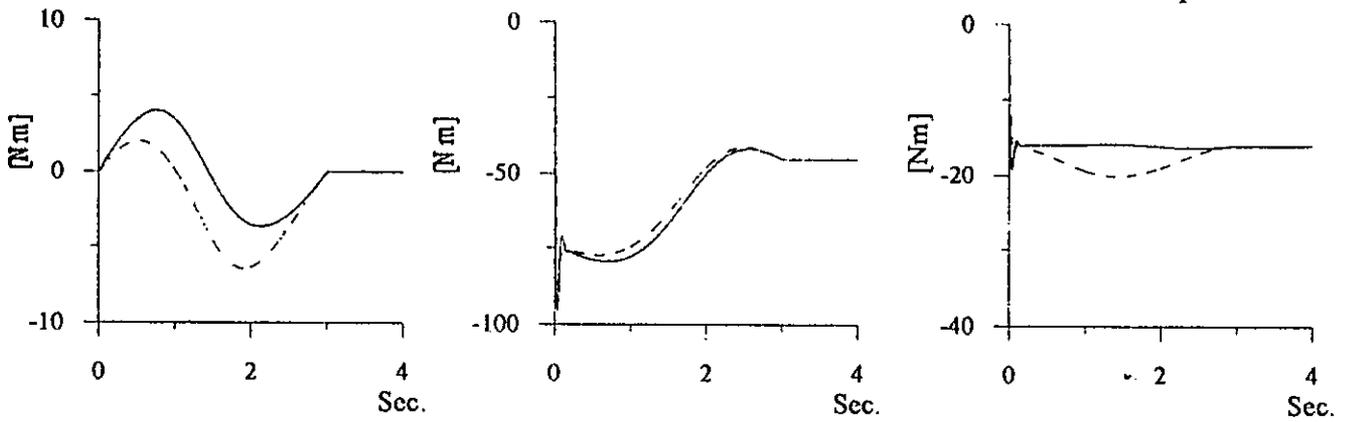
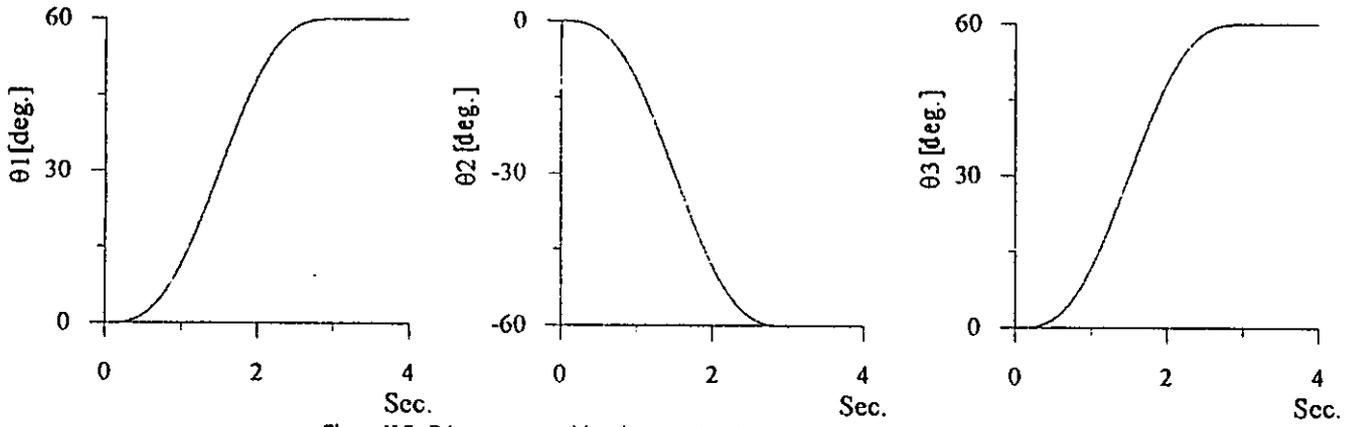
combinons ces résultats, on obtient

$$\dot{L} \leq -z^T Q z, \quad Q > 0 \quad (\text{II.65})$$

(II.65) implique  $L_i$ ,  $z_i$ ,  $e_i$ ,  $r_i$ ,  $\hat{W}_i$ ,  $\hat{V}_i$ , et  $\hat{\varphi}_i$ ,  $i$  allant de 1 à  $n$ , sont tous bornés. De la forme de (II.65), il est facile de montrer que le carré de  $z(t)$  est intégrable; d'où les carrés de  $e$  et  $r$  sont intégrables. De l'équation de l'erreur en boucle fermée (II.7), il est aussi facile de montrer que  $\dot{r}$  est borné; par conséquent,  $r$  est uniformément continu. Par la lemme de Barbalat [KWA 95b], nous savons que  $e$  et  $\dot{e}$  tendent asymptotiquement vers zéro.

### Etude par simulation

Les figures (II.7) et (II.8) montrent les performances satisfaisantes obtenues par cette structure de commande pour les mêmes tests utilisés dans les sections précédentes.



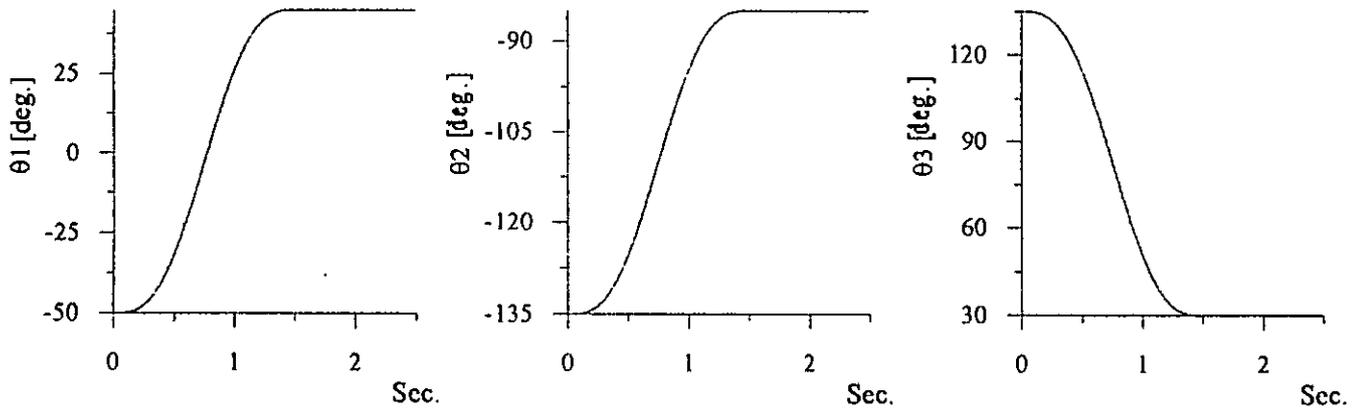


Figure II.8a Réponses en position des art. 1, 2 et 3

----- à vide. - - - - avec perturbation.

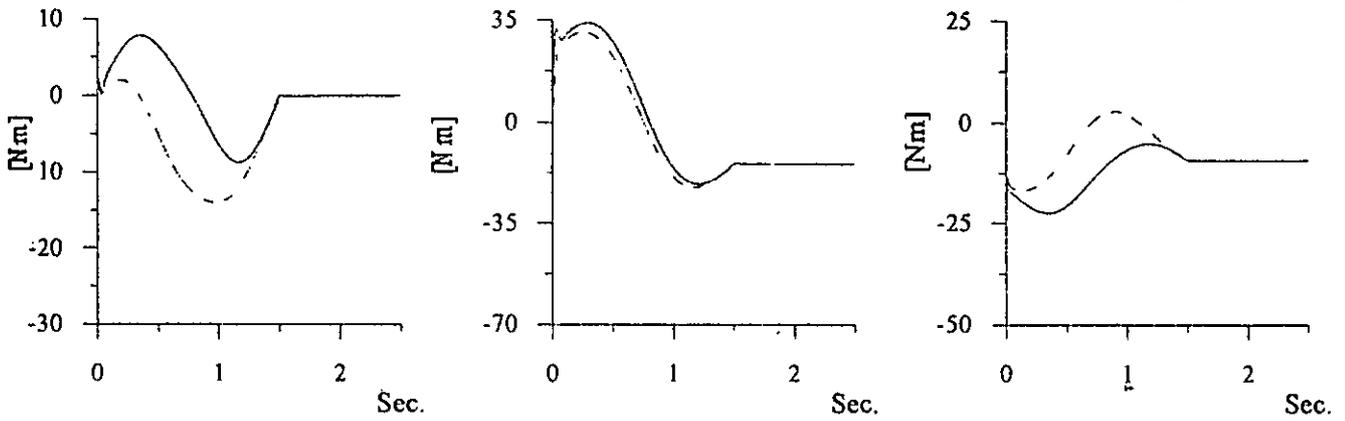


Figure II.8b Commandes à vide des art. 1, 2 et 3

—  $\tau$ .

----- Sortie du RNA.

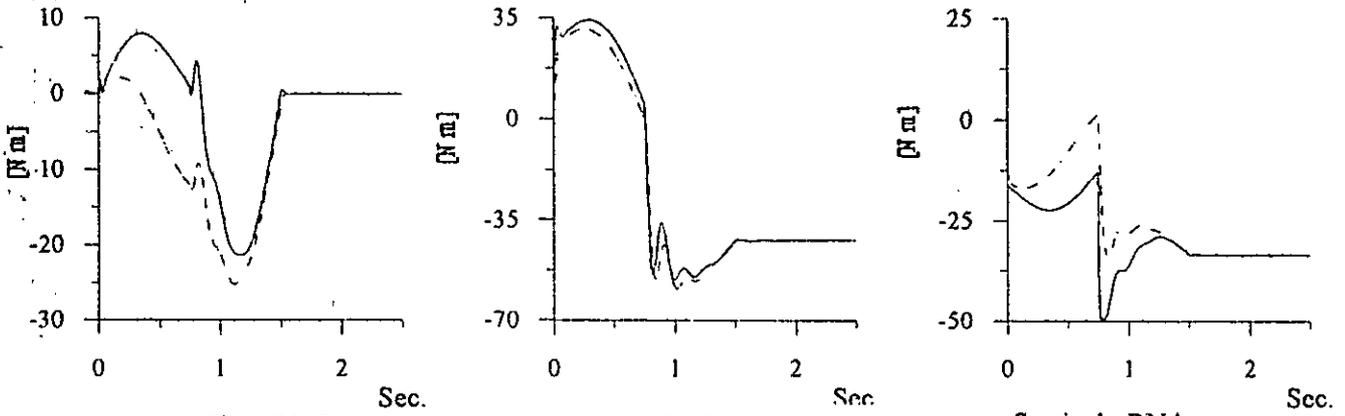


Figure II.8c Commandes avec perturbation des art. 1, 2 et 3

—  $\tau$ .

----- Sortie du RNA.

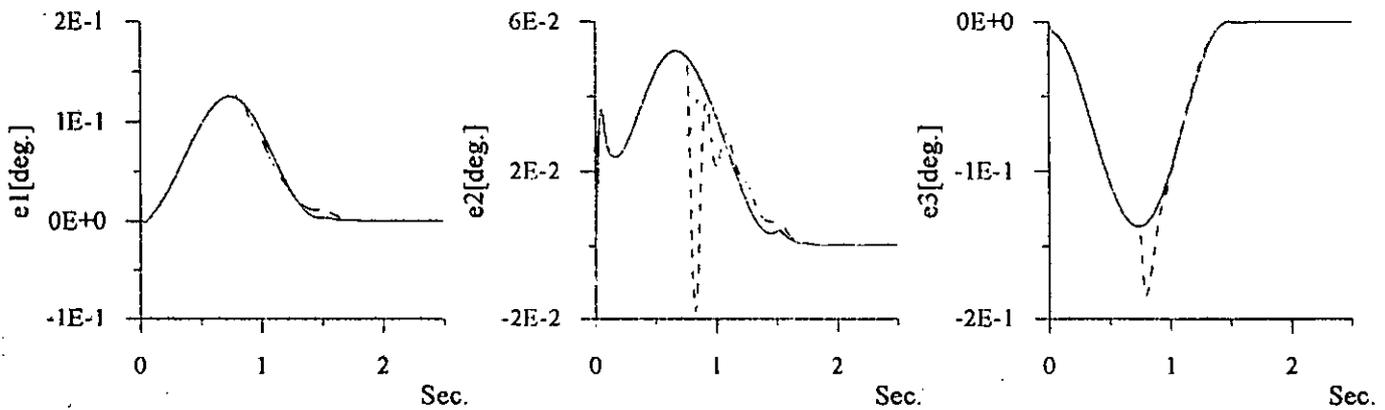


Figure II.8d Erreurs des art. 1, 2 et 3

— à vide. - - - - avec perturbation.

## II.4.2 Fonction de Base radiale à Centres Adaptatifs (Réseaux à Fonction de Base Radiale Gaussienne Généralisés) (RBFGG)

Malgré le grand succès dans la réduction notable du nombre de neurones par la décentralisation effectuée dans la section précédente, le nombre de neurone reste lié au nombre de subdivision des entrées, donc lié par les caractéristiques de reconstruction, de cela il risque d'être élevé. Pour surmonter cet handicap, l'utilisation des réseaux gaussiens généralisés a été l'objet de cette section.

Soit la fonction couple non linéaire de la  $i^{\text{ème}}$  articulation définie par l'équation (II.2), son représentation par un réseau RBFG (éq. I.83) est donnée par l'équation suivante :

$$f_i^*(x_i) = c_{0i} + \sum_{j=1}^N c_{ij} g(\|x_i - \xi_{ij}\|) \quad (\text{II.66})$$

La commande de la  $i^{\text{ème}}$  articulation est donnée par :

$$\tau_i(t) = \hat{f}_i^*(x_i) + K_{vi} \cdot r_i(t) + v_{Ri}(t) \quad (\text{II.67})$$

avec:

- $\hat{f}_i^*(.)$  : l'estimation de la fonction  $f_i^*(.)$  à chaque instant  $t$  par un réseau de base radiale généralisé,
- $K_{vi} \in \mathfrak{R}^{n \times n}$  : matrice symétrique définie positive,
- $v_{Ri}(t) \in \mathfrak{R}^n$  : terme robuste donné par l'équation (II.13).

Les lois d'adaptation des réseaux à base radiales généralisés sont :

$$\dot{\hat{C}}_j(t) = F_{Cj} \cdot g(\|x_j - \hat{\xi}_j\|) \cdot r_j^T, \quad j = 1, \dots, n \quad (\text{II.68})$$

$$\dot{\hat{R}}_j(t) = F_{Rj} \frac{2}{\sigma^2} \cdot (x_j - \hat{\xi}_j) \cdot g(\|x_j - \hat{\xi}_j\|) \cdot \hat{C}_j \cdot r_j, \quad j = 1, \dots, n \quad (\text{II.69})$$

avec:  $F_{Cj}$  et  $F_{Rj}$  ( $j=1, \dots, n$ ) sont des matrices constantes symétriques définies positives.

### Etude par Simulation

Pour mettre en évidence la capacité d'adaptation du schéma de commande présenté, les mêmes simulations utilisées précédemment ont été effectuées sur les trois premières articulations d'un robot de type PUMA 560.

Les résultats sont présentés sur les figures (II.9) et (II.10). Ils montrent l'efficacité de la stratégie de commande adaptative directe par RBFG généralisés en termes de poursuite de trajectoire de référence pour les robots manipulateurs, ainsi que la robustesse vis-à-vis des perturbations.

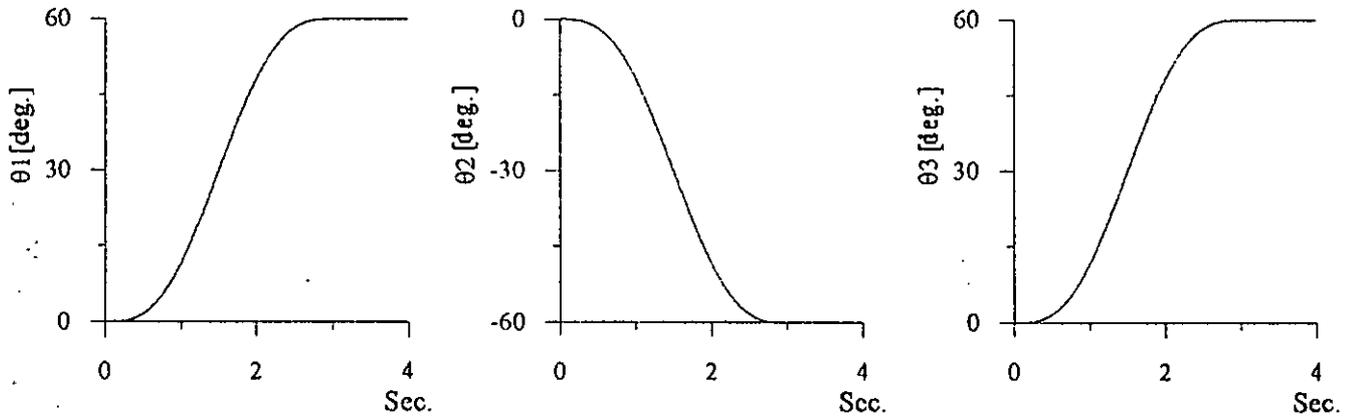


Figure 11.9a Réponses en position des art. 1, 2 et 3

----- à vide.    - - - - avec perturbation.

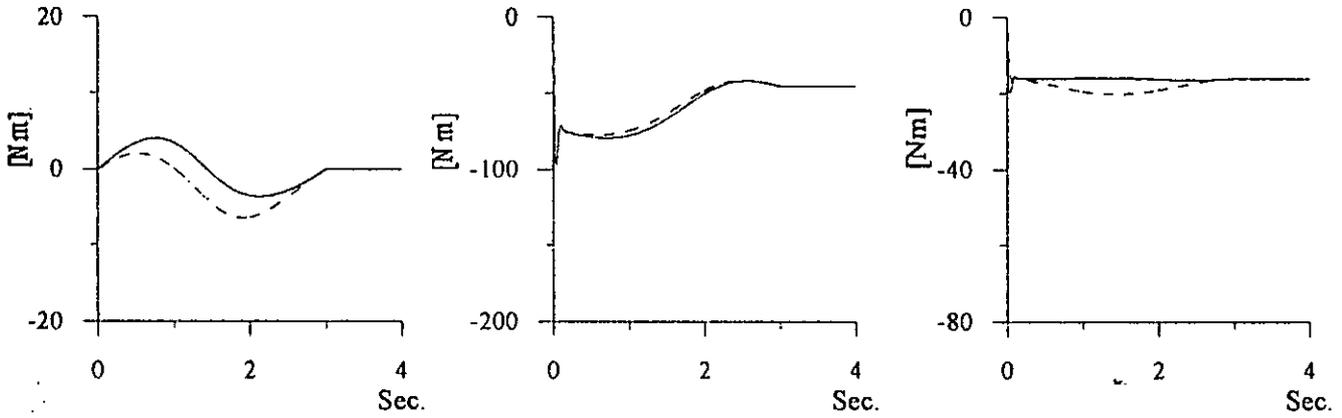


Figure 11.9b Commandes à vide des art. 1, 2 et 3

-----  $\tau$ .    - - - - Sortie du RNA.

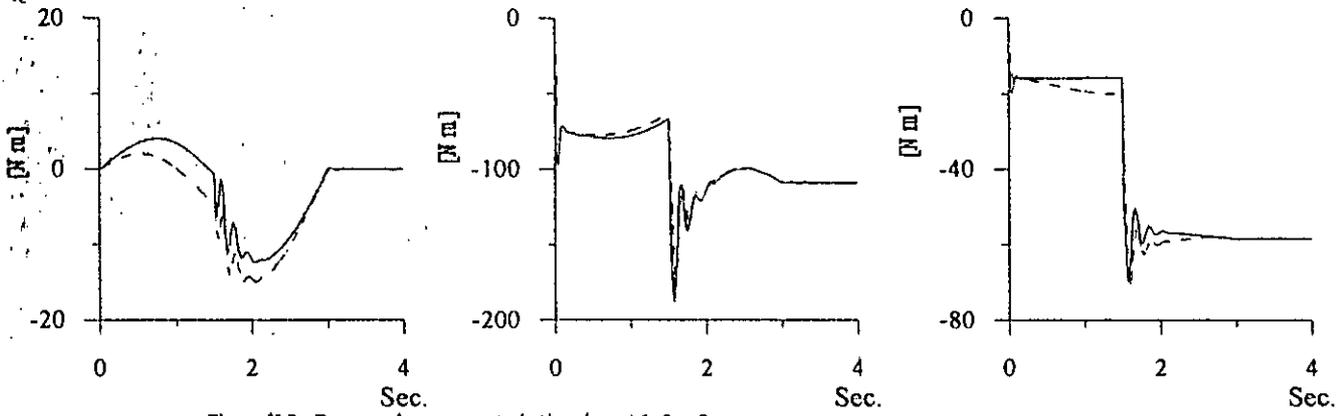


Figure 11.9c Commandes avec perturbation des art. 1, 2 et 3

-----  $\tau$ .    - - - - Sortie du RNA.

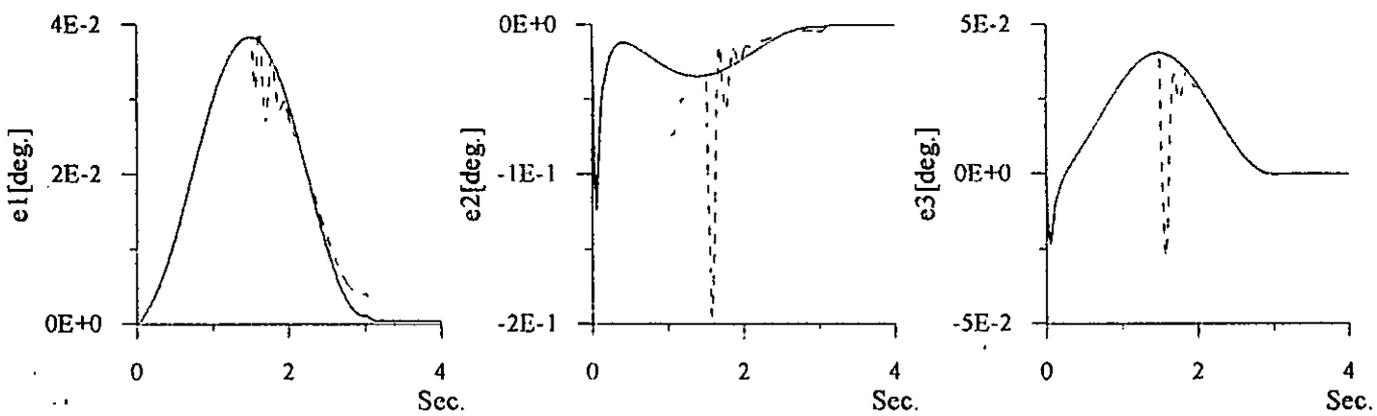
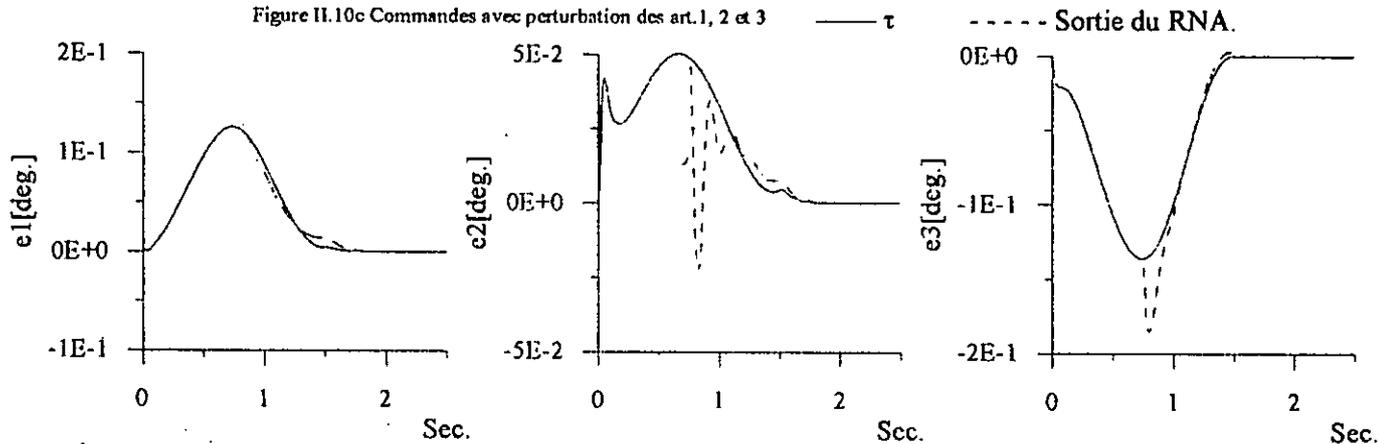
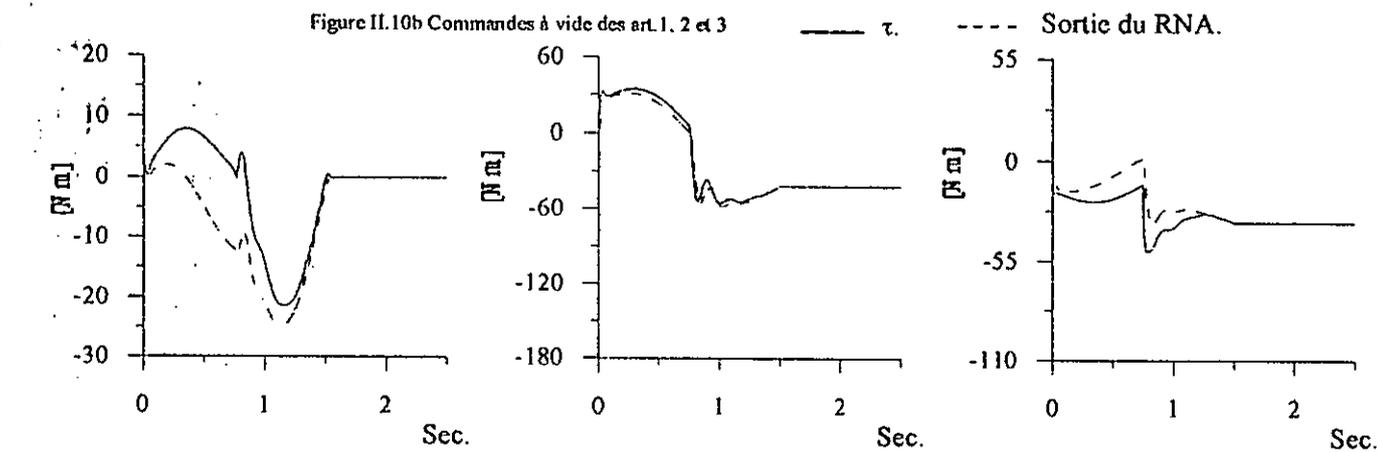
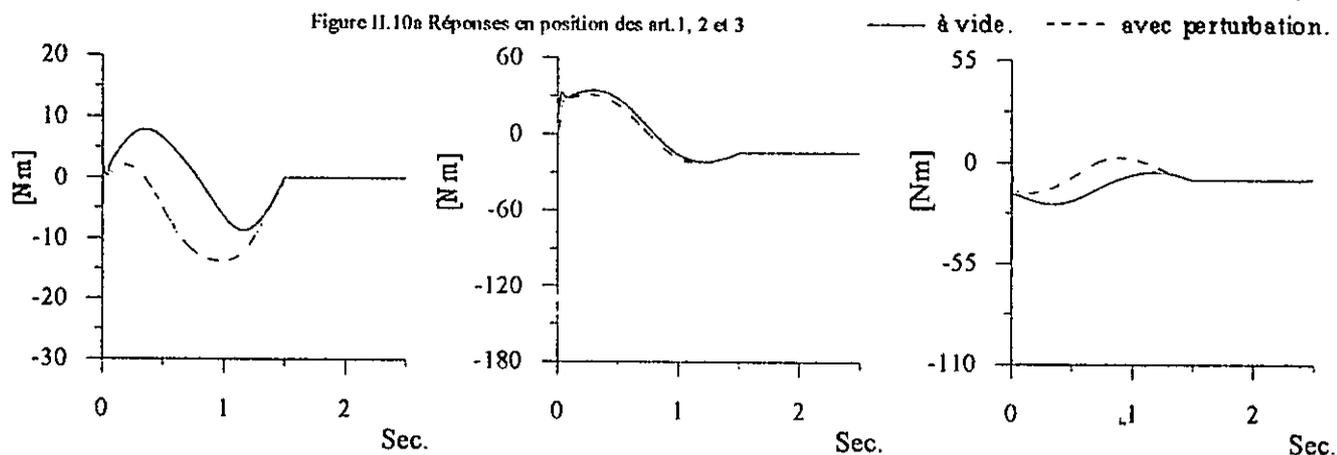
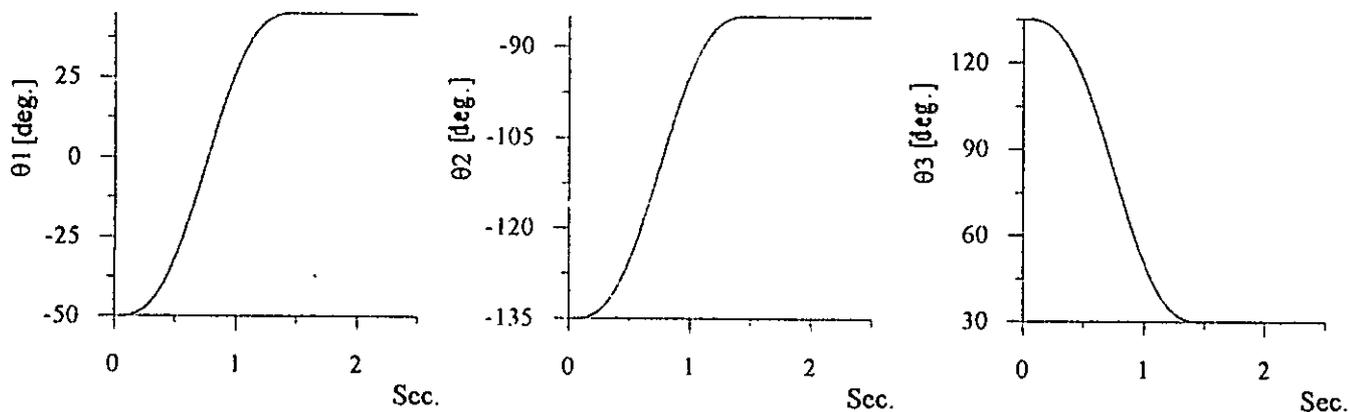


Figure 11.9d Erreurs des art. 1, 2 et 3

----- à vide.    - - - - avec perturbation.



## II.5. Commande Prédictive Adaptative Décentralisée Directe par RNA à Fonction De Base Radiale Gaussienne

### II.5.1 Fonction de Base Radiale à Centres Fixes

Vu d'une part, les résultats obtenus par la commande adaptative décentralisée et d'autre part, les résultats obtenus par la commande prédictive (DCAL), la combinaison des deux a été établie.

En utilisant les états désirés, l'équation (II.66) devient

$$f_{di}^*(x_i) = c_{0i} + \sum_{j=1}^N c_{ij} g(\|x_i - \xi_{ij}\|), \quad x_i = [\theta_{di} \quad \dot{\theta}_{di} \quad \ddot{\theta}_{di}]^T \quad (\text{II.70})$$

L'expression matricielle de la fonction  $f_{dA}^*$  (équation II.54) est

$$f_{dA}^*(x) = [f_{dA1}^*(x_1) \quad \dots \quad f_{dAn}^*(x_n)]^T \quad x_i = [\theta_{di} \quad \dot{\theta}_{di} \quad \ddot{\theta}_{di}]^T, \quad i = 1, \dots, n \quad (\text{II.71})$$

l'estimée de cette fonction par les RNA à structure décentralisée est donnée par

$$\hat{f}_{dA}^*(x) = [\hat{f}_{dA1}^*(x_1) \quad \dots \quad \hat{f}_{dAn}^*(x_n)]^T \quad x_i = [\theta_{di} \quad \dot{\theta}_{di} \quad \ddot{\theta}_{di}]^T, \quad i = 1, \dots, n \quad (\text{II.72})$$

La loi de commande de la  $i^{\text{ème}}$  articulation est donnée par

$$\tau_i(t) = K_{Di} r_i(t) + (1 - m_i(t)) (\hat{f}_{dAi}^*(x_i(t)) + v_{Ri}(t)) + m_i(t) u_{si}(t) \quad (\text{II.73})$$

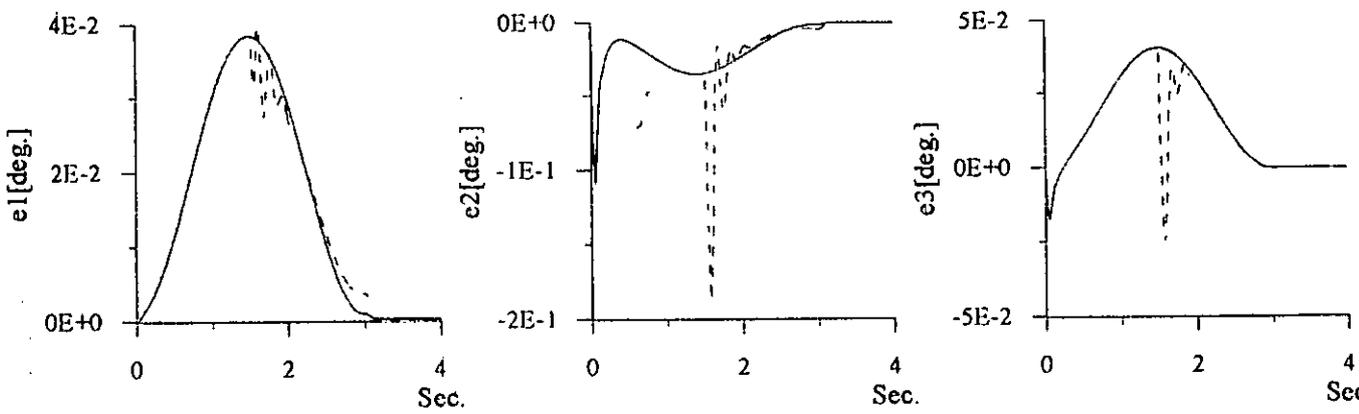
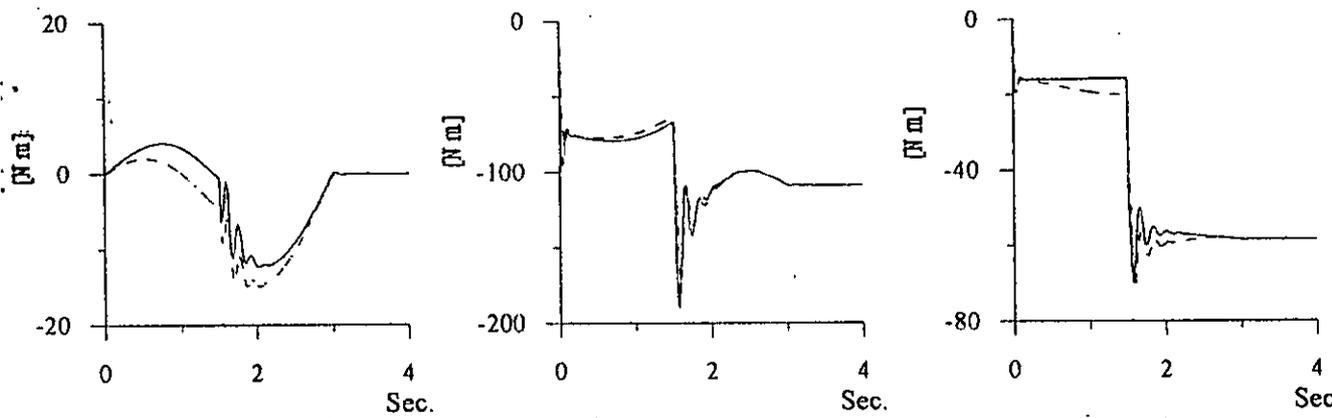
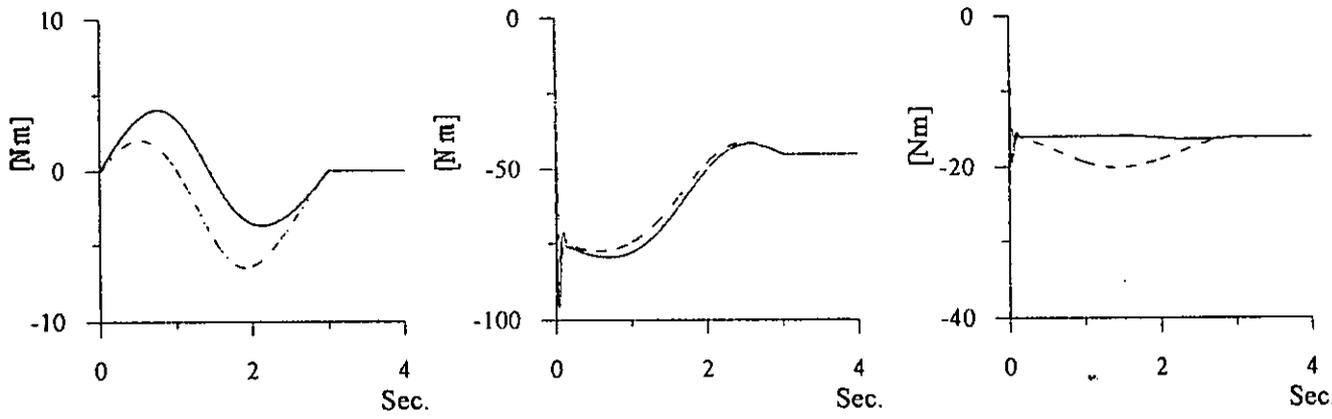
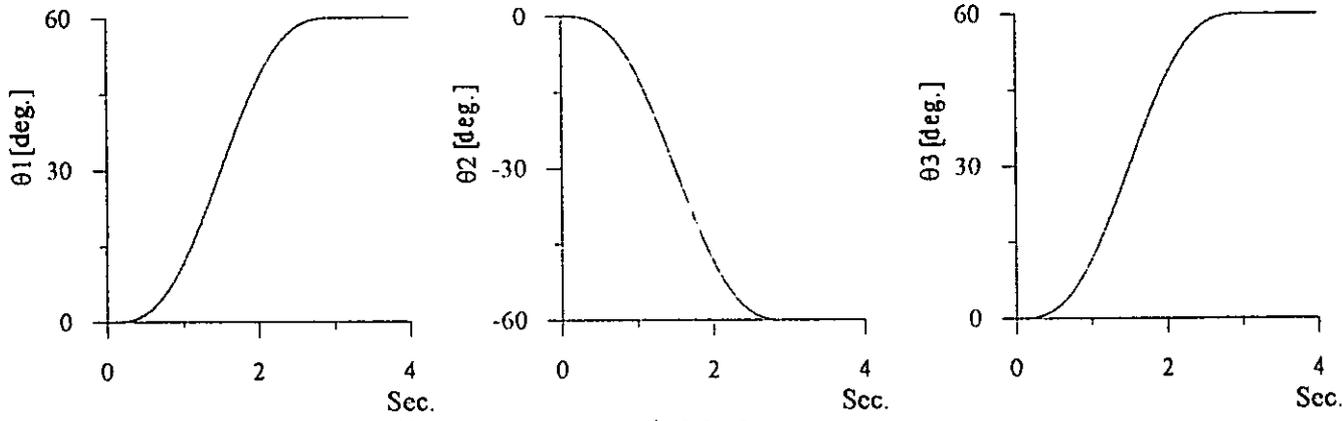
$K_{Di} > 0$  et  $k_{si} = M_{0i}(x(t)) - \varepsilon_{fi}$ .

L'ajustement des poids et l'analyse de la stabilité sont les mêmes que celles de l'approche développée dans la section précédente (section II.4.1)

### Etude par simulation

Pour la validation de la stratégie de commande présentée, des simulations ont été effectuées sur le même bras de robot manipulateur de type PUMA 560.

Les figures (II.11) et (II.12) montrent les performances satisfaisantes obtenues par cette structure de commande. Nous constatons que les commandes sont lisses et les erreurs de poursuite sont très faibles.



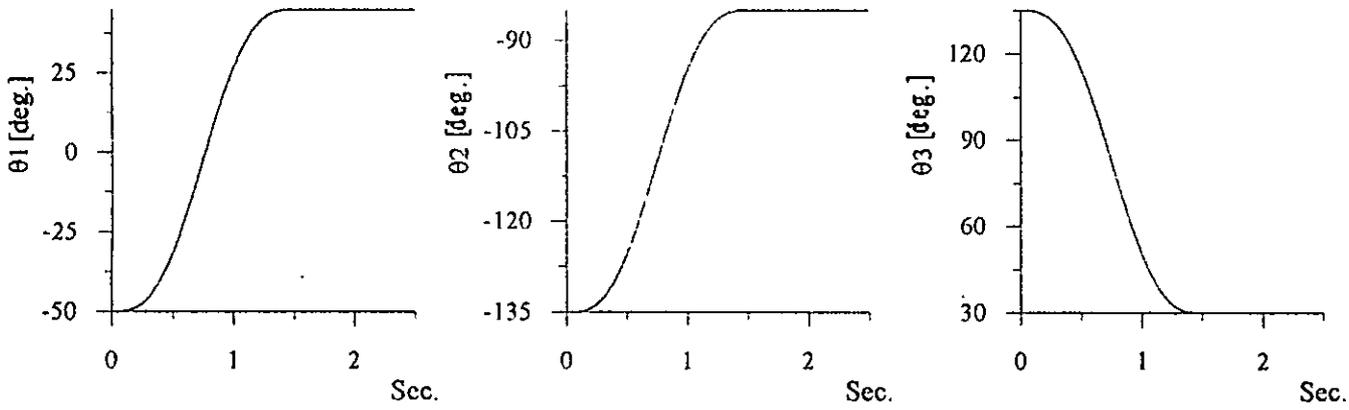


Figure II.12a Réponses en position des art. 1, 2 et 3

— à vide. - - - avec perturbation.

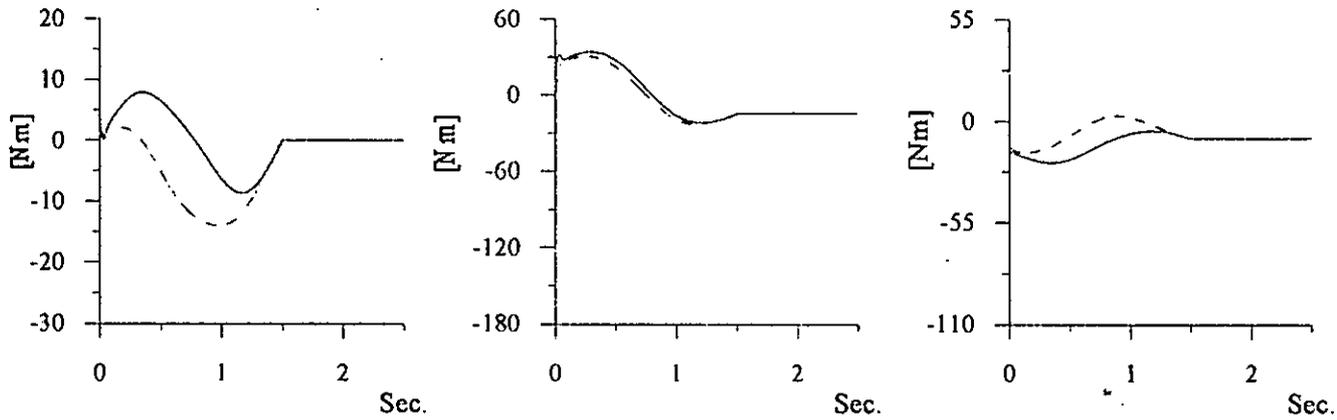


Figure II.12b Commandes à vide des art. 1, 2 et 3

—  $\tau$ . - - - Sortie du RNA.

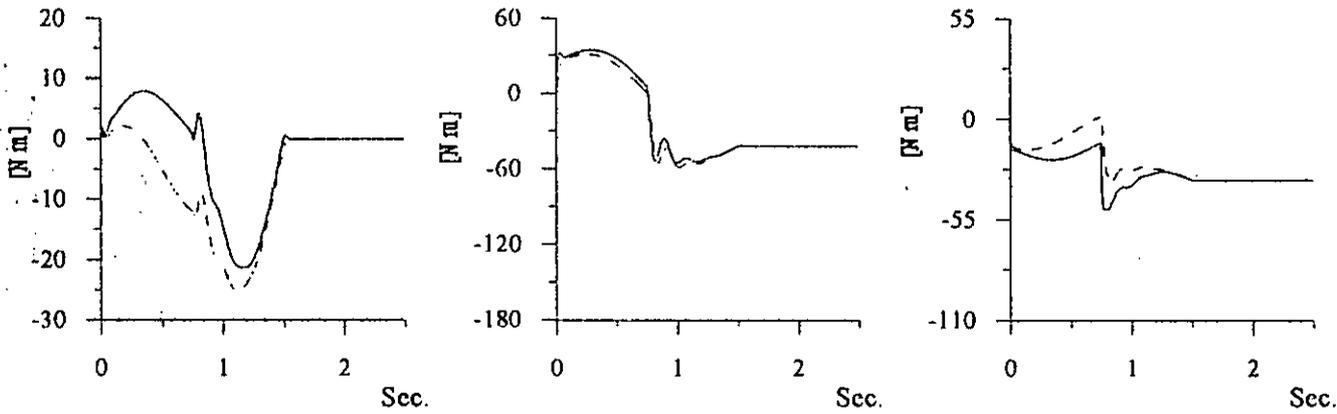


Figure II.12c Commandes avec perturbation des art. 1, 2 et 3

—  $\tau$ . - - - Sortie du RNA.

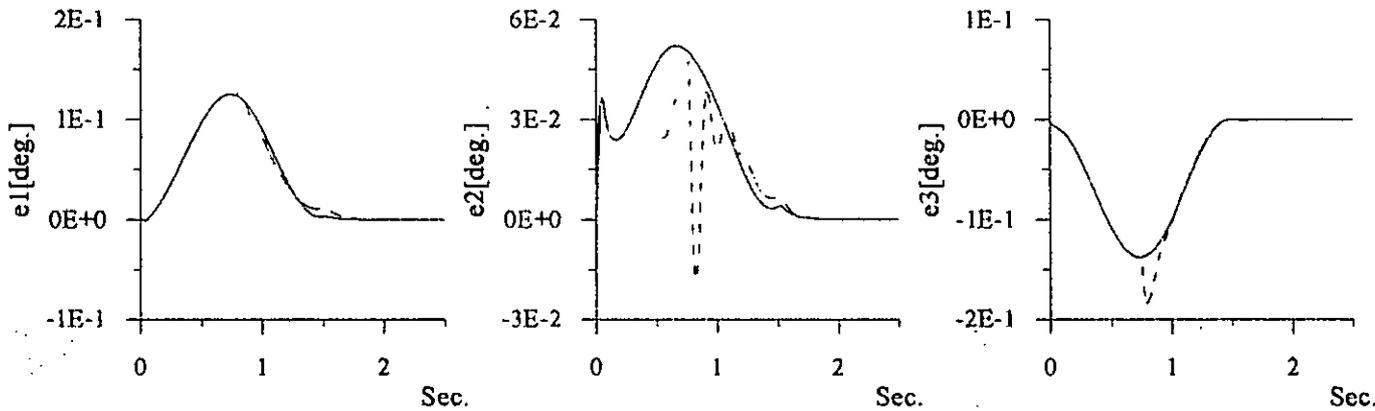


Figure II.12d Erreurs des art. 1, 2 et 3

— à vide. - - - avec perturbation.

### II.5.2 Fonction de Base Radiale à Centres Adaptatifs (Réseaux à Fonction de Base Radiale Généralisé)

Cette technique de commande est basée sur l'extension de l'approche développée dans la section (I.5.2) au cas décentralisé.

La commande de la  $i^{\text{ème}}$  articulation est donnée par :

$$\tau_i(t) = \hat{f}_{\hat{d}_i}(x_i) + K_{v_i} \cdot r_i(t) + v_{Ri}(t), \quad x_i = [\theta_{\hat{d}_i} \quad \dot{\theta}_{\hat{d}_i} \quad \ddot{\theta}_{\hat{d}_i}]^T \quad (\text{II.74})$$

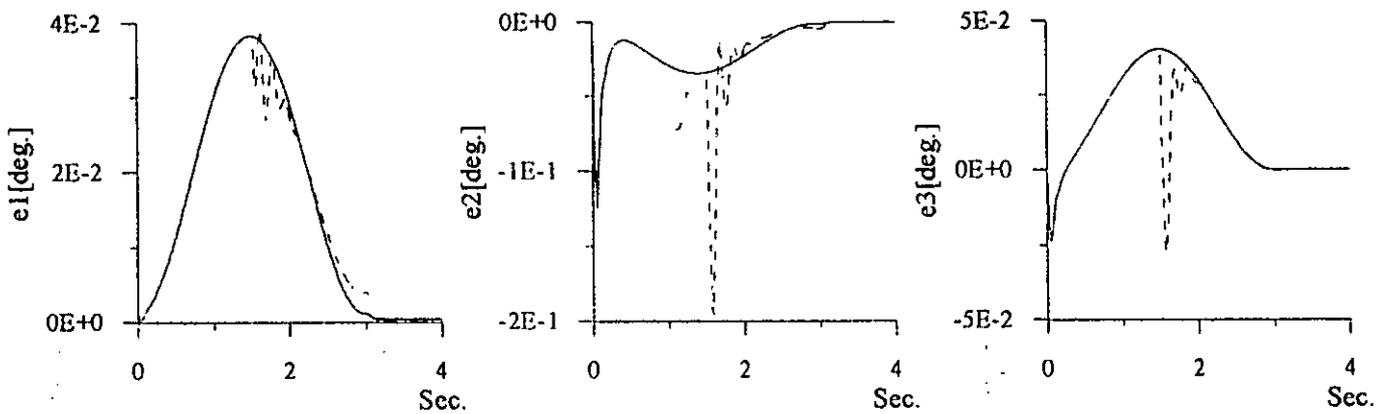
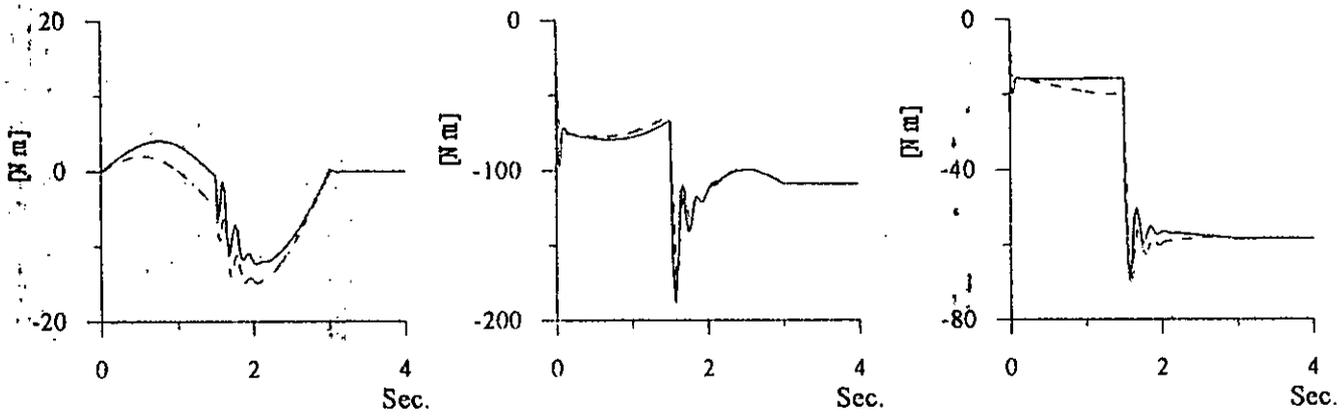
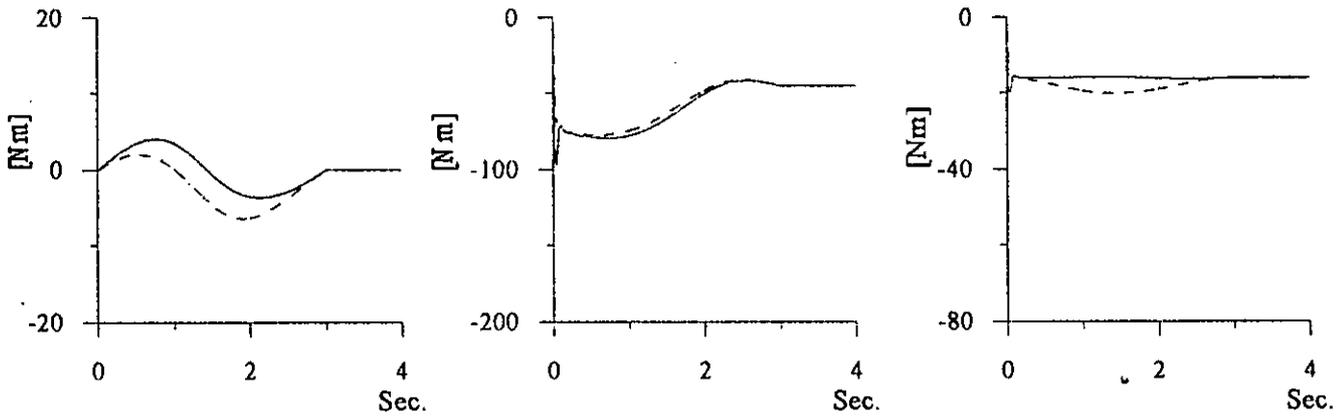
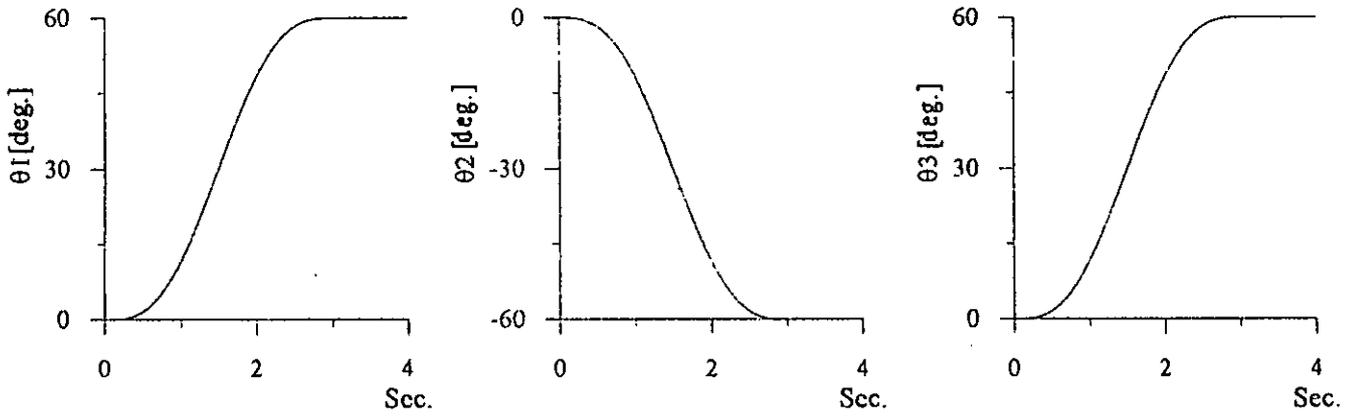
avec:  $\hat{f}_i(.)$  est l'estimation de la fonction  $f_i$  à chaque instant  $t$  par le réseau RBFG généralisé,  $K_{v_i} \in \mathfrak{R}^{n \times n}$  : matrice gain symétrique défini positive,  $v_{Ri}(t) \in \mathfrak{R}^n$  : couple additionnel robuste donné par l'équation (II.13).

Les lois d'adaptation sont les mêmes que celles données par les équations (I.68) et (I.69).

#### Etude par Simulation

Pour mettre en évidence les performances et l'efficacité de la stratégie de commande présentée, les mêmes simulations utilisées précédemment ont été effectuées.

Les résultats, sont présentés sur les figures (II.13) et (II.14). Ils montrent l'efficacité de la stratégie de commande prédictive adaptative directe par RBFG généralisés en termes de poursuite de trajectoire de références pour les robots manipulateurs, ainsi que la robustesse vis-à-vis des perturbations.



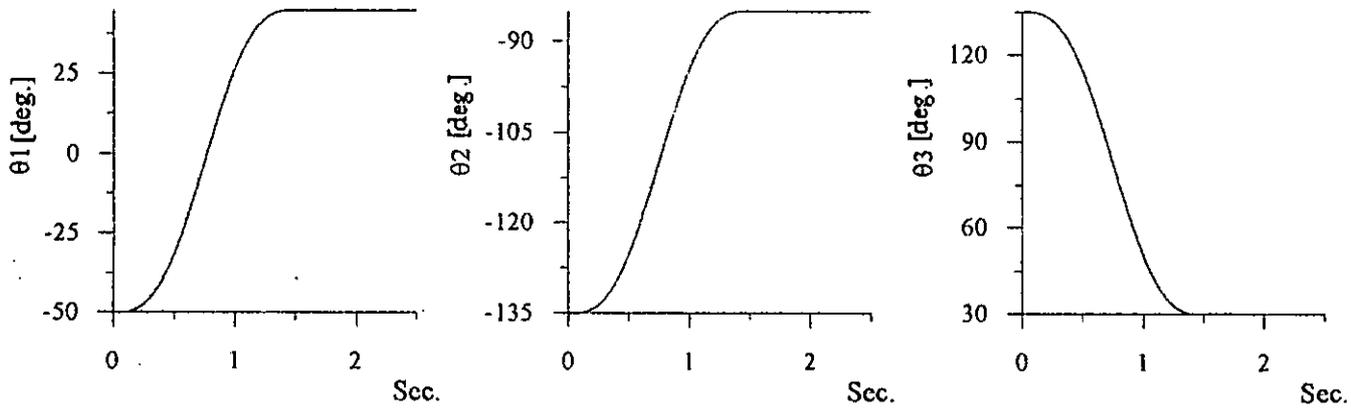


Figure II.14a Réponses en position des art. 1, 2 et 3

— à vide. - - - avec perturbation

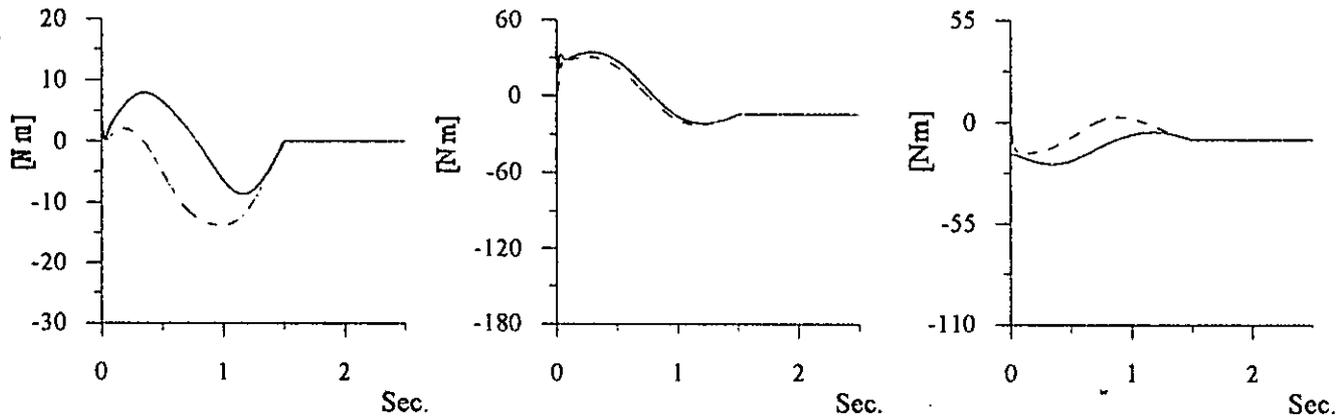


Figure II.14b Commandes à vide des art. 1, 2 et 3

—  $\tau$ .

Sortie du RNA.

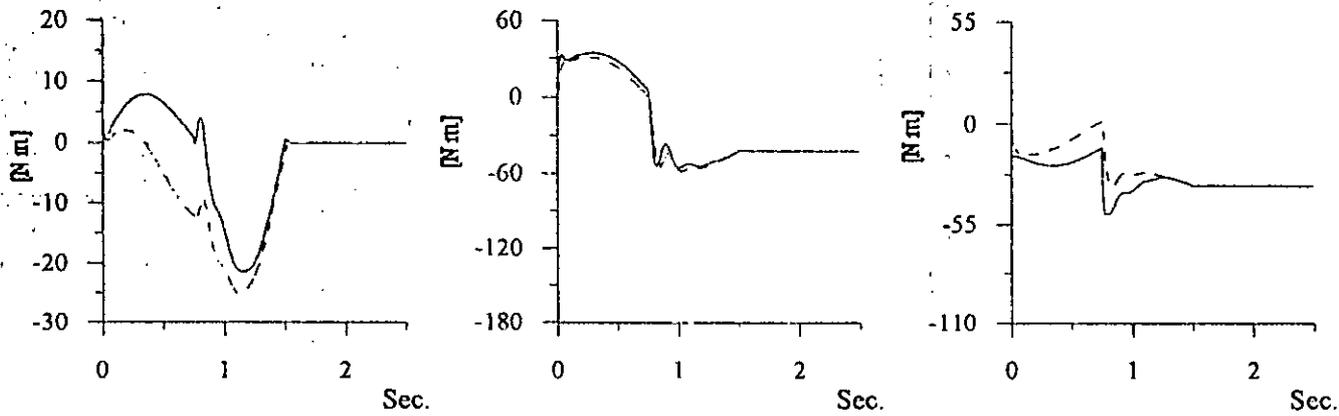


Figure II.14c Commandes avec perturbation des art. 1, 2 et 3

—  $\tau$

- - - Sortie du RNA.

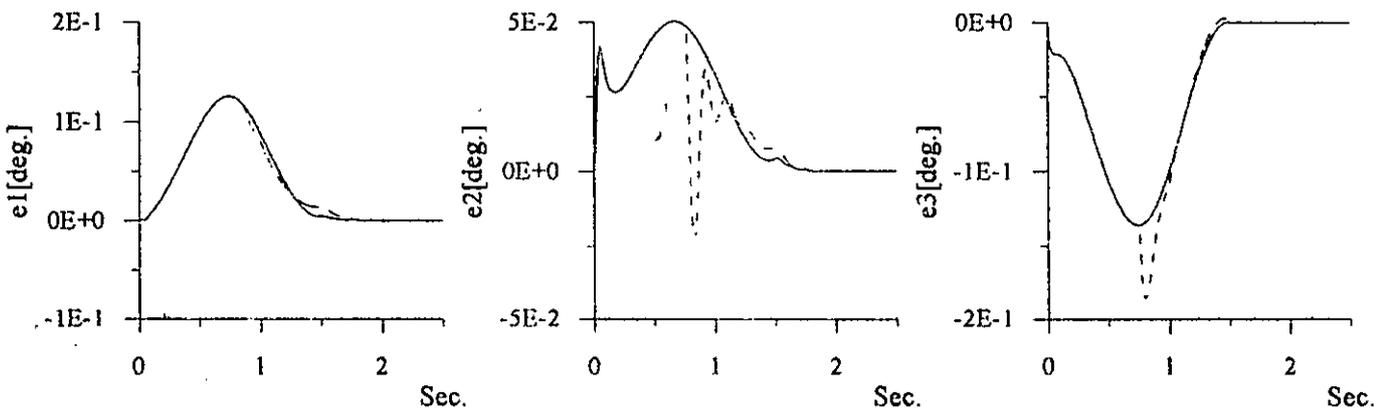


Figure II.14d Erreurs des art. 1, 2 et 3

— à vide. - - - avec perturbation.

## II.6 Commande Adaptative Décentralisée Directe par RNA Récurrents (RNAR)

Ce n'est que l'extension, au cas décentralisé, de l'approche développée dans la section (I.6). L'entrée externe du réseau, qui approxime  $f_i(x_i)$ , dans ce schéma de commande est donnée par :

$$u_i = [\theta_i \quad \dot{\theta}_i \quad \ddot{\theta}_i]^T.$$

La loi de commande est donnée par :

$$\tau_i(t) = \hat{f}_i(u_i) + K_v \cdot r_i(t) \quad (\text{II.75})$$

Les lois d'adaptation de chacun des réseaux sont les mêmes que celles de la section (I.6).

### Etude par simulation

L'efficacité du schéma de commande proposé a été vérifiée par l'application à la commande en poursuite de trajectoire d'un robot de type PUMA 560. Les figures (II.15) et (II.16) illustrent les résultats obtenus pour le test utilisé auparavant.

## II.7 Commande Prédictive Adaptative Décentralisée Directe par RNA Récurrents

L'extension, au cas décentralisé, de la commande prédictive adaptative directe par réseaux récurrents (section I.7) a été effectuée. L'entrée externe du réseau estimant  $f_i(x_i)$  (fonction couple non linéaire de la  $i^{\text{ème}}$  articulation) est donnée par :

$$u_i = [\theta_{\alpha} \quad \dot{\theta}_{\alpha} \quad \ddot{\theta}_{\alpha}]^T.$$

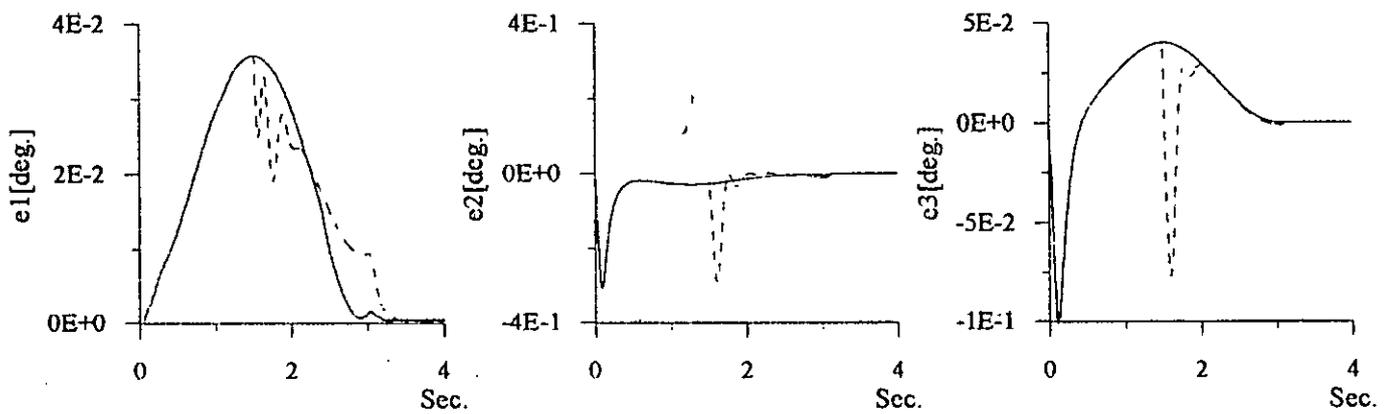
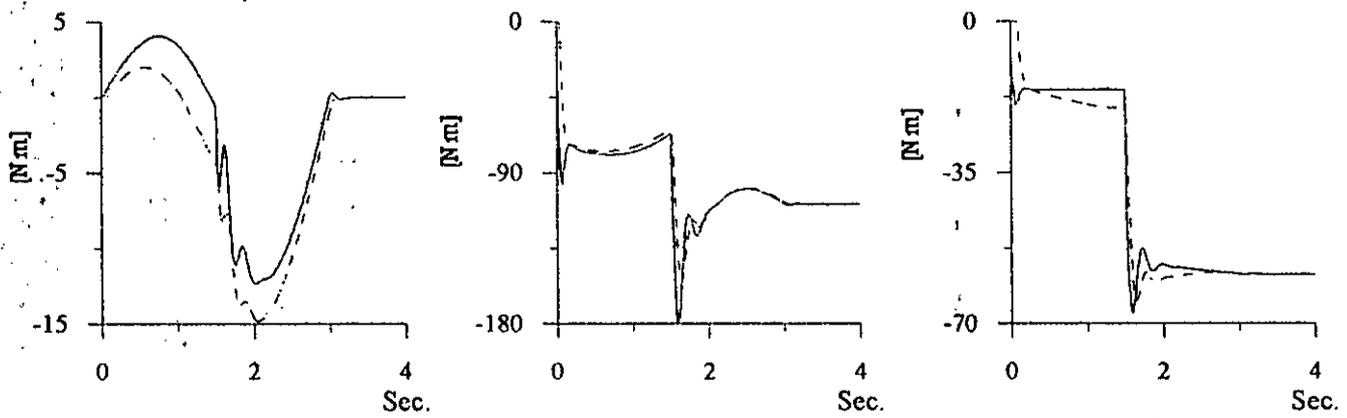
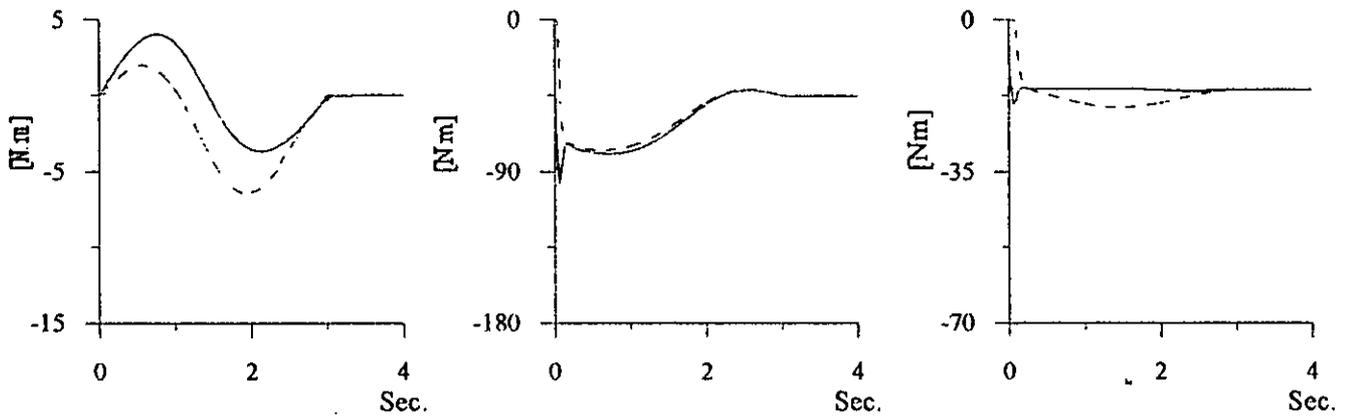
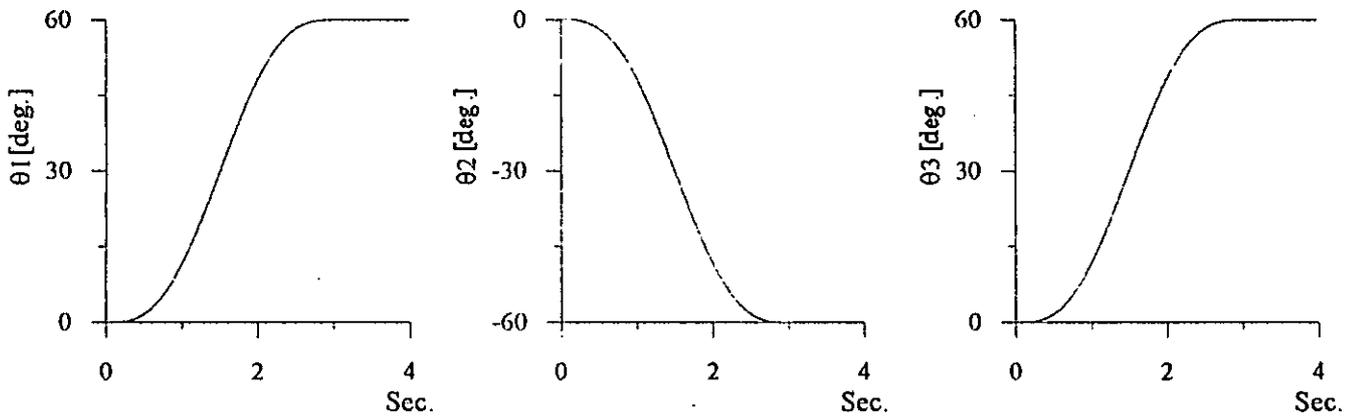
La loi de commande est donnée par :

$$\tau_i(t) = \hat{f}_{\alpha}(u_i) + K_v \cdot r_i(t) \quad (\text{II.76})$$

Les lois d'adaptation de chacun des réseaux sont les mêmes que celles de la section (I.6).

### Etude par simulation

Pour mettre en évidence les performances du schéma de commande proposé, nous avons effectué les mêmes simulations utilisées précédemment et les résultats sont présentés sur les figures (II.17) et (II.18). Nous constatons que les performances de poursuite sont satisfaisantes.



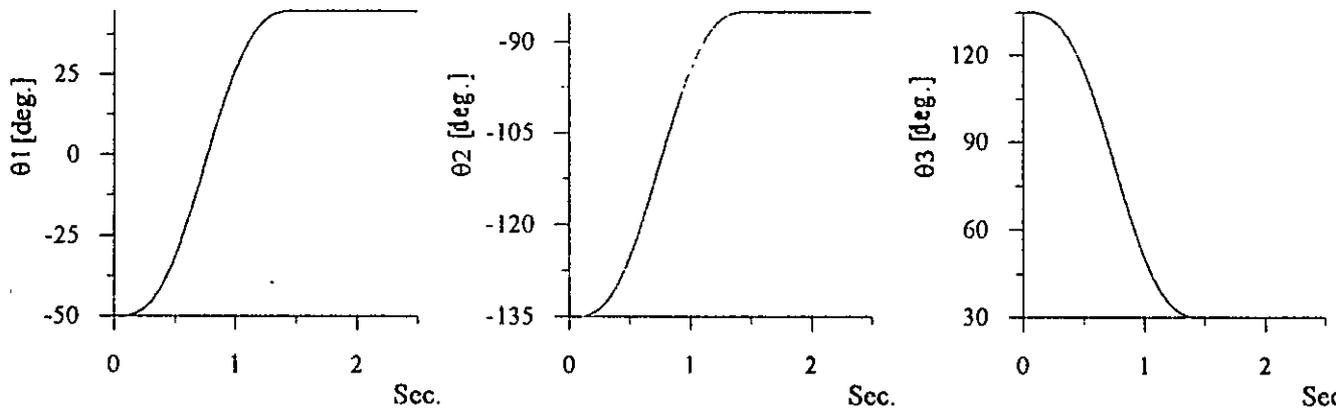


Figure II.16a Réponses en position des art. 1, 2 et 3

— à vide. - - - avec perturbation.

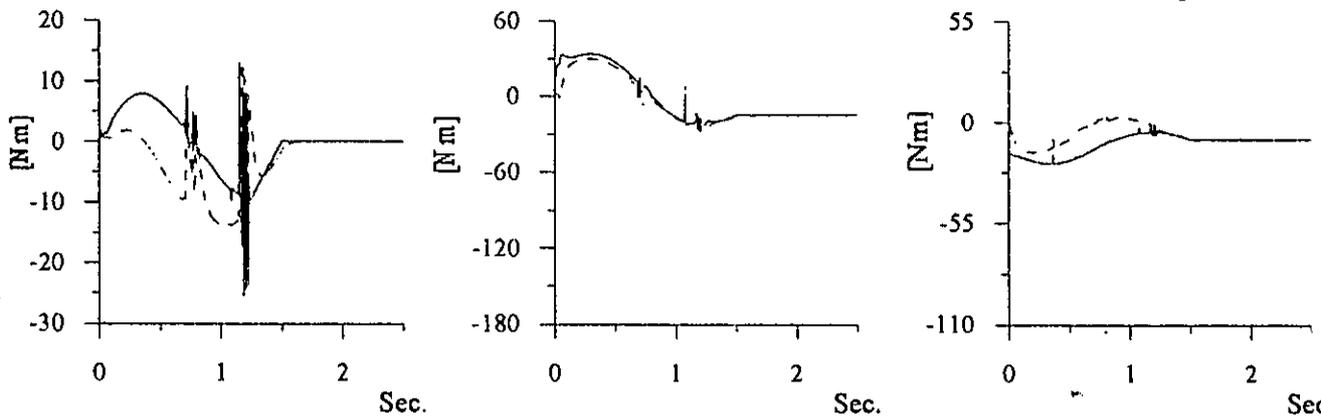


Figure II.16b Commandes à vide des art. 1, 2 et 3

—  $\tau$ . - - - Sortie du RNA.

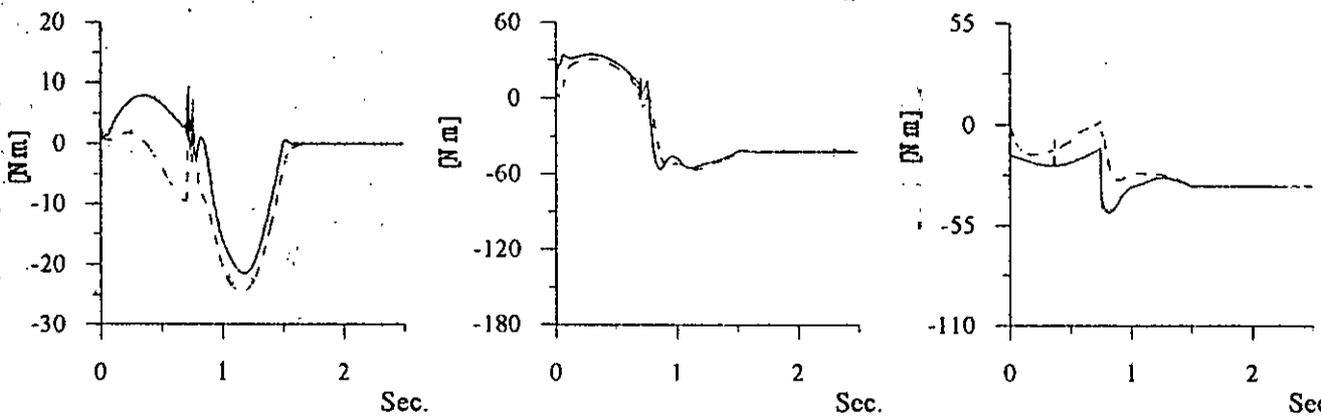


Figure II.16c Commandes avec perturbation des art. 1, 2 et 3

—  $\tau$ . - - - Sortie du RNA.

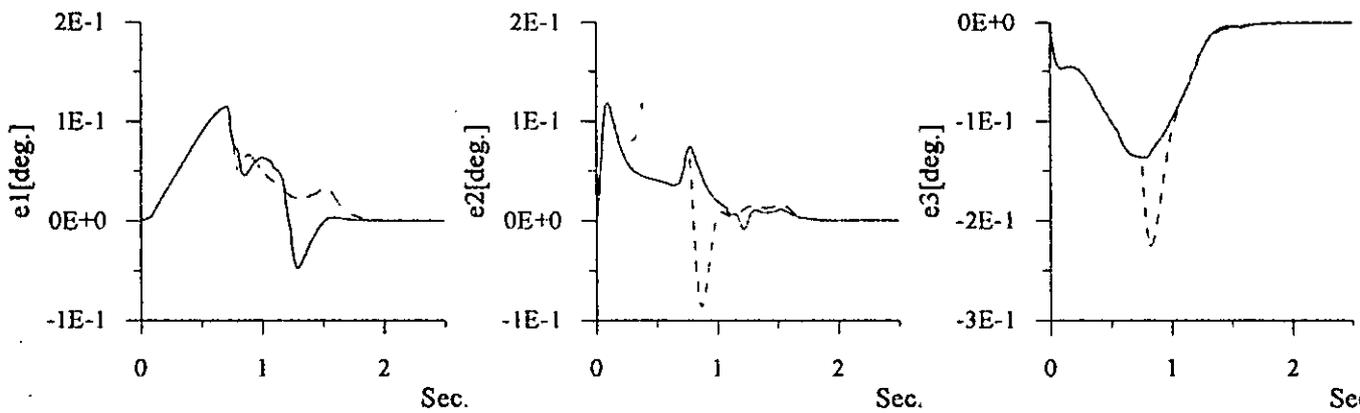
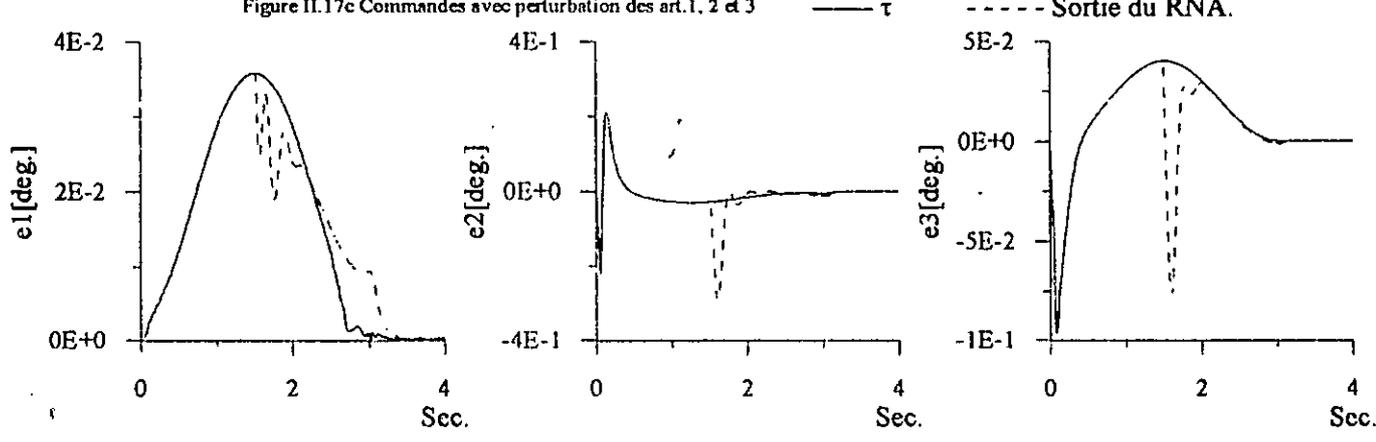
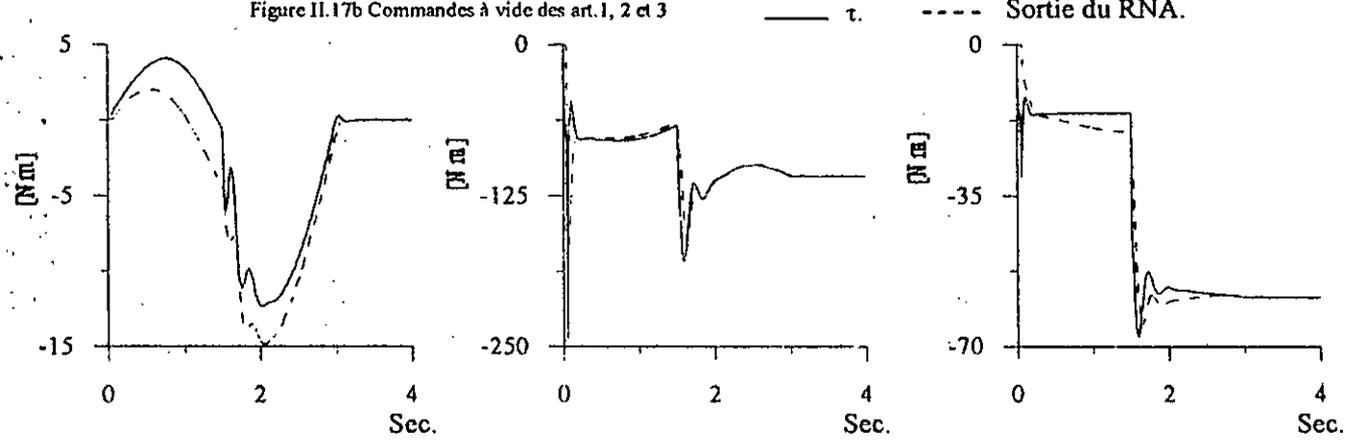
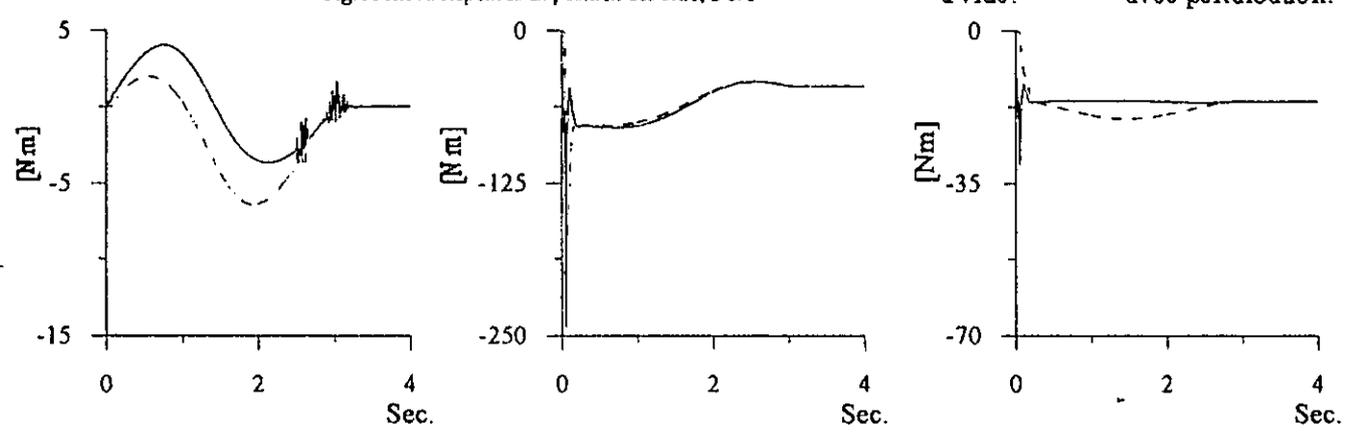
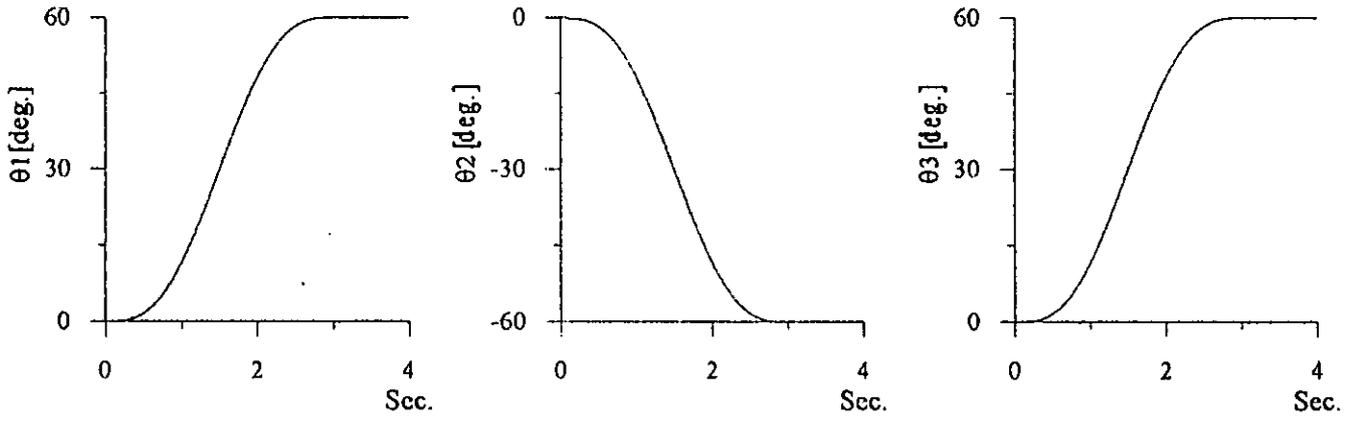
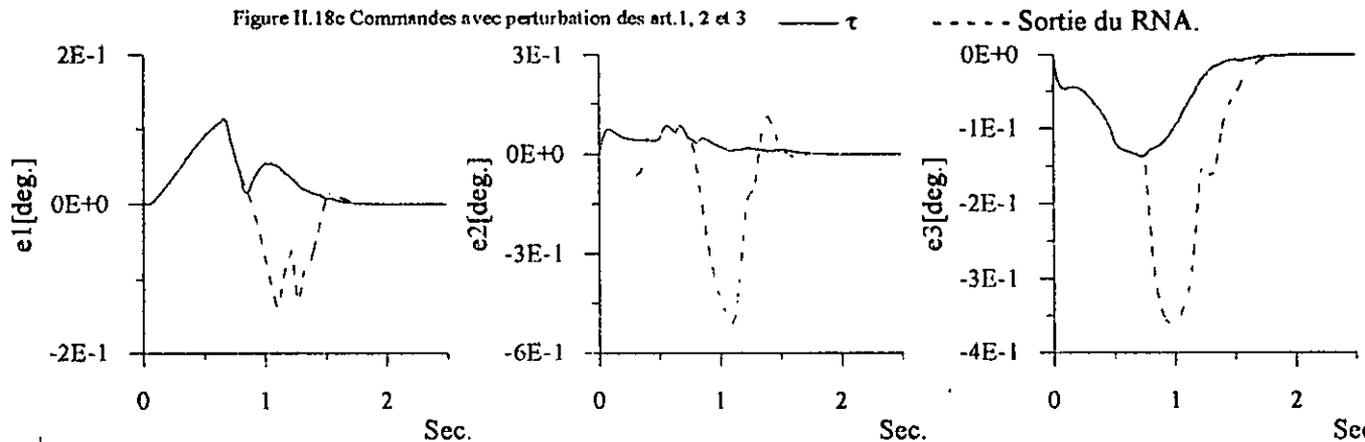
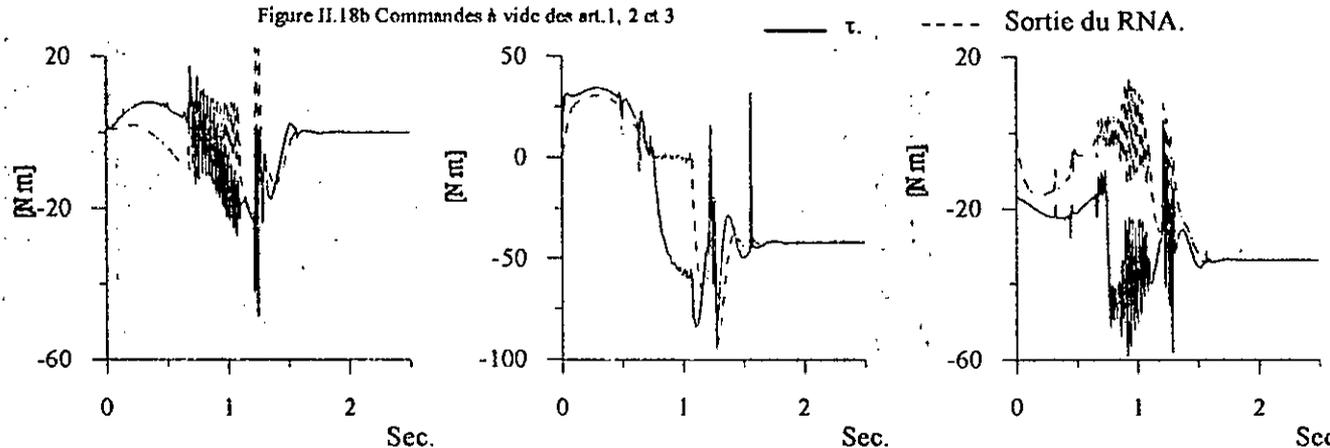
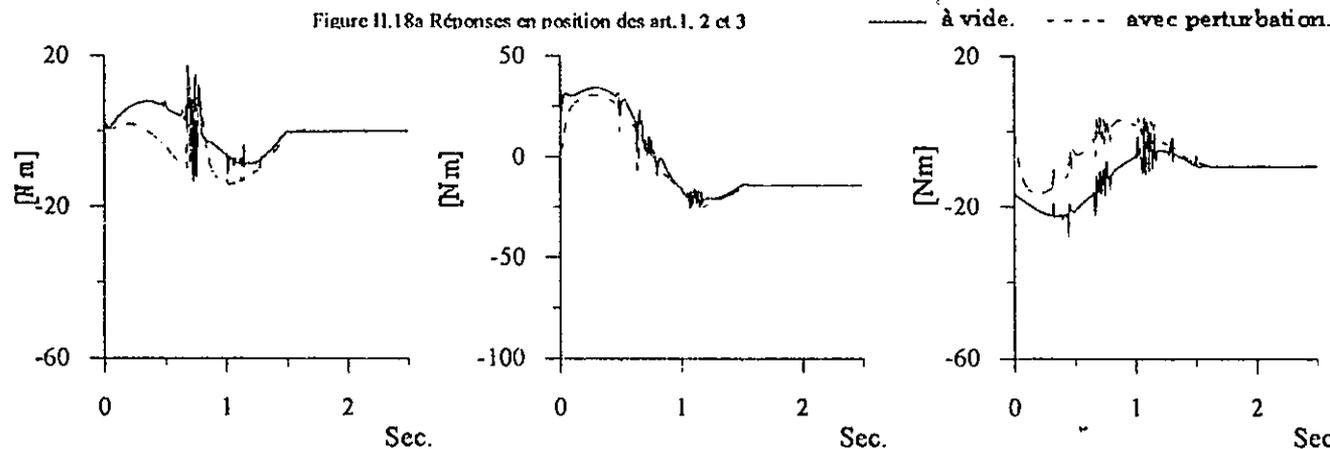
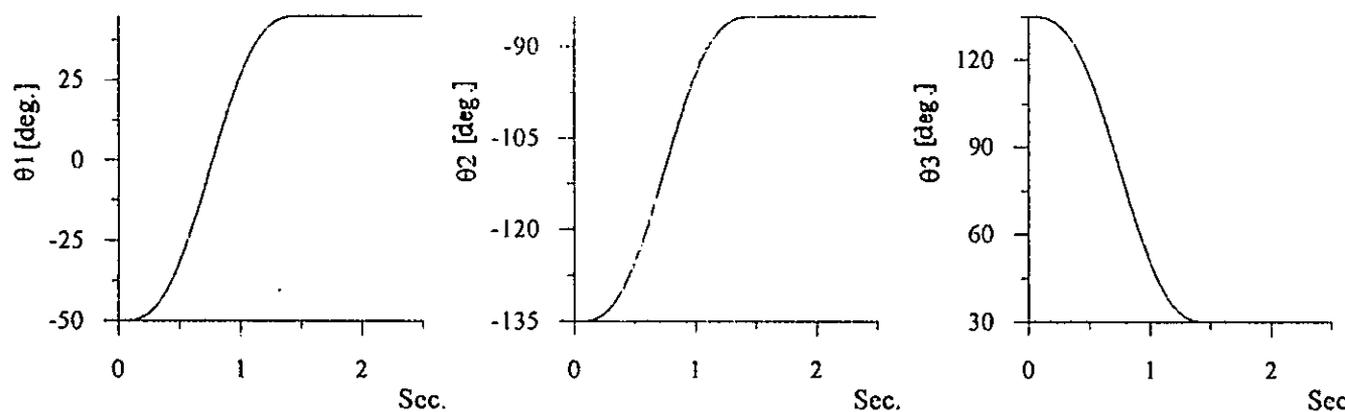


Figure II.16d Erreurs des art. 1, 2 et 3

— à vide. - - - avec perturbation.





## II.8 Conclusion

Ce chapitre a été consacré au développement des commandes décentralisées à partir de l'extension, au cas décentralisé, des méthodes présentées au chapitre I. Ces derniers ont permis l'amélioration considérable de la robustesse vis-à-vis des perturbations par l'utilisation des RNA. A noter, que la décentralisation de la commande a permis de réduire considérablement le calcul de celle-ci. En effet, nous avons développé la commande adaptative décentralisée par RNA statique en deux étapes. Dans la première étape est développée une loi de commande partiellement centralisée comportant une composante décentralisée et un compensateur centralisé. Dans la deuxième étape est effectuée une décentralisation totale de ce dernier dans le cas général, c.-à-d. : des interconnexions d'ordre quelconque. En plus, nous avons développé la commande prédictive adaptative décentralisée directe par RNA statique. Dans cette technique de commande, les réseaux ne dépendent pas des états du robot, ils dépendent que de la trajectoire désirée ce qui minimise la taille du vecteur de l'entrée et permet donc, le calcul hors ligne les fonctions dépendantes de ce dernier. De ce fait, elle réduit considérablement le temps de calcul en ligne. En outre, le développement de la commande adaptative décentralisée directe par RBFG à centres fixes et à centres adaptatifs ainsi que sa version prédictive a été effectuée. En fin, la commande adaptative décentralisée par les réseaux récurrents ainsi que sa version prédictive ont été proposées.

L'étude par simulation a été effectuée en utilisant les quatre types de réseaux de neurones (les réseaux statiques, les réseaux RBFG à centres fixes, les réseaux RBFG généralisés et les réseaux récurrents) en vue d'une comparaison de leurs performances dans la poursuite de la trajectoire du robot PUMA 560. Les résultats obtenus sont très satisfaisants et il est important de noter que les réseaux RBFG généralisés donnent une meilleur poursuite de la référence et les commandes sont plus lisses et le test de perturbation n'affecte pas les performances de poursuite.

# *Commande Adaptative Décentralisée par RNA basée sur la Méthode du Couple Calculé*

---

### **III.1 Introduction**

La commande des robots manipulateurs constitue à l'heure actuelle l'une des principales préoccupations des chercheurs en robotique. Ces derniers se trouvent généralement, face à des difficultés d'élaborer des algorithmes de commande des robots afin qu'ils puissent exécuter des tâches précises et diversifiées.

En effet, la présence des couplages entre les liaisons complique la conception de la commande. Cette commande doit être choisie de façon compatible avec les caractéristiques et les phénomènes imprévisibles qui régissent leurs dynamiques. L'une des méthodes utilisées est celle du couple calculé 'computed torque'. Cette méthode est basée sur la connaissance exacte du modèle dynamique du robot. Elle présente des faiblesses à cause des incertitudes du modèle. A cet effet, plusieurs approches adaptatives ont été proposées pour maintenir les performances de poursuite en présence des incertitudes structurelles [SLO 88][KOI 83][SER 89]. Malgré l'efficacité de la commande adaptative pour la compensation des incertitudes structurelles, la compensation des incertitudes non structurelles n'est pas garantie [OZA 91]. Or, ceci peut être assuré par les possibilités offertes par les RNA pour l'identification et la commande des systèmes dynamiques complexes [NAR 90a][HUN 92][BOU 97b]. Dans [OZA 91] l'idée de la méthode du couple calculé a été introduite pour compenser des non-linéarité en utilisant les RNA. Cette méthode est différente de celles utilisant le modèle inverse.

La majorité des algorithmes de commande sont développés dans un contexte centralisé, dont l'information est traitée dans une seule unité de commande nécessitant la mesure de tous les états (sorties) du système. Cette structure rend généralement le calcul lent [SER 89]. Pour pallier cet inconvénient, les chercheurs se sont intéressés lors des deux dernières décennies à la commande décentralisée [SER 89][KAR 93][BOU 96]. Cette dernière consiste à décomposer le système en plusieurs sous systèmes interconnectés pouvant être géographiquement éloigné, chaque sous système est commandé localement par une station de commande qui n'a accès qu'aux informations locales du sous système considéré et permettant ainsi la simplification de l'implémentation des algorithmes par rapport à la commande centralisée.

Dans ce chapitre, l'approche développée dans [OZA 91] a été étendue au cas décentralisé en synthétisant une commande adaptative décentralisée par RNA avec poursuite de trajectoire du robot manipulateur PUMA 560 [AMS 86].

### III.2 Méthode du Couple Calculé

Dans le domaine de la robotique, l'élaboration des modèles nécessite une étude approfondie et détaillée de la structure. Des efforts considérables ont été consacrés à l'élaboration des modèles dynamiques pour différents robots [FU 87][VIB 87][PAU 81]. En se basant sur ces modèles, plusieurs types de commande ont été appliqués aux différents robots [PAU 81][STO 87]. Vu le caractère fortement non linéaire et le couplage existant, l'évaluation des paramètres s'est avéré difficile.

Le modèle dynamique d'un bras de robot manipulateur à  $n$  degrés de liberté est donné par l'équation suivante :

$$M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) + F = \tau \quad (\text{III.1})$$

l'expression de la loi de commande par la méthode du couple calculé (figure III.1) est de la forme [SLO 88][OZA 91]:

$$\tau = \hat{M}(\theta)u + \hat{h}(\theta, \dot{\theta}) \quad (\text{III.2})$$

$$u = \ddot{\theta}_d + k_p(\theta_d - \theta) + k_v(\dot{\theta}_d - \dot{\theta}) \quad (\text{III.3})$$

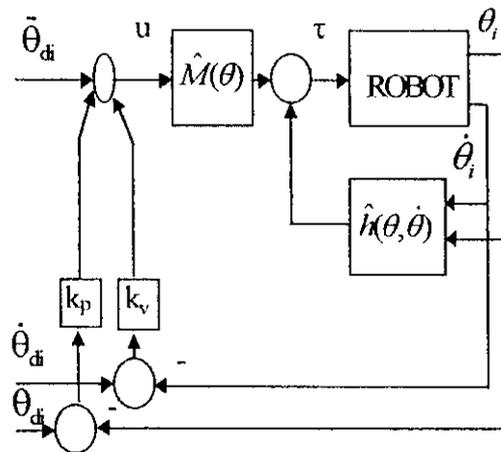


Figure III.1. Commande par la méthode du couple calculé

où  $\hat{M}(\theta)$  et  $\hat{h}(\theta)$  sont les estimées de  $M(\theta)$  et  $h(\theta)$  respectivement. Ces estimations sont utilisées pour la compensation des non linéarités du robot manipulateur.

$\theta_d$  désigne la position désirée, les matrices des gains  $k_p$  et  $k_v$  sont diagonales définies positives et choisies de manière à imposer une dynamique à l'erreur de poursuite des trajectoires désirées. On peut imposer au système un comportement en boucle fermée semblable à celui d'un système du second ordre ayant un amortissement  $\xi$  et une pulsation  $\omega_0$ . Ceci peut être réalisé en choisissant

$$k_v = 2\xi\omega_0 \text{ et } k_p = \omega_0^2$$

donc

$$\ddot{e} + k_v\dot{e} + k_p e = \hat{M}^{-1}((M - \hat{M})\ddot{\theta} + h - \hat{h} + F) \tag{III.4}$$

F désigne les incertitudes non structurales.

Pour F=0, plusieurs schémas de commande adaptative ont été proposés [SLO 88][KOI 83] et l'algorithme de commande est asymptotiquement stable. Par contre, si les incertitudes non structurées F ne sont pas négligeables, la stabilité de l'algorithme n'est pas garantie. Pour surmonter ce problème, Ozaki [OZA 91] a proposé un schéma de commande basé sur la méthode du couple calculé utilisant les RNA pour estimer la matrice d'inertie M et le terme h+F par deux réseaux RNA<sub>1</sub> et RNA<sub>2</sub> respectivement.

### III.3. Commande Neuronale Adaptative

La commande neuronale proposée par Ozaki [OZA 91] est présentée par la figure (III.2). Les deux réseaux RNA<sub>1</sub> et RNA<sub>2</sub> sont respectivement l'approximation de la matrice d'inertie et le terme h+F du robot manipulateur. En effet, la structure de commande est simple et ces performances sont comparables avec ceux des contrôleurs adaptatifs [OZA 91]. Malgré ces résultats obtenus, cette technique présente un inconvénient majeur dans le cas où le degré de liberté est très élevé car les entrées du RNA<sub>1</sub> et RNA<sub>2</sub> sont respectivement n et 2n et les sorties sont respectivement n×n et n. Par exemple, si le robot est à trois articulations (n=3) cela implique que le réseau RNA<sub>1</sub> possède trois entrées et neuf sorties et RNA<sub>2</sub> possède six entrées et trois sorties, ce qui rend l'apprentissage difficile à cause de l'utilisation des réseaux MIMO (Multi Input Multi Output). Or, les réseaux MISO (Multi Output Simple Output) sont les plus utilisés dans la commande [NAR 90a] [HUN 91] [BOU 97b] vu leurs avantages tels que : la taille du réseau, caractéristiques d'entraînement, etc...

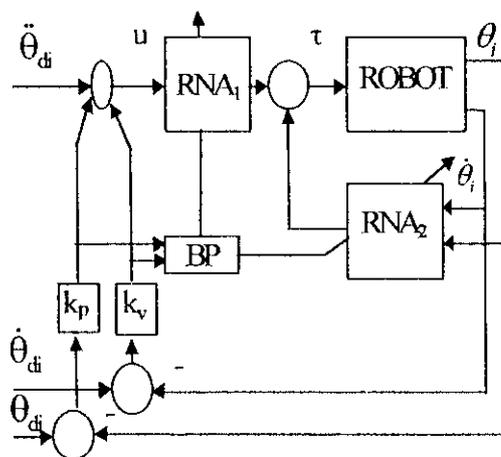


Figure III.2. Commande Neuronale Adaptative.

### III.4. Commande Adaptative Décentralisée par RNA Statique

Dans le but de surmonter le problème de la taille des entrées/sorties présenté dans la commande neuronale adaptative, une stratégie de commande adaptative décentralisée a été développée [BOU 98a]. Cette méthode constitue une extension, au cas décentralisé, de la méthode du couple calculé en utilisant les RNA.

Le modèle du robot est considéré comme un système interconnecté composé de  $n$  sous-systèmes dont chacun représente une articulation. De l'équation (III.1), on peut donc écrire:

$$m_{ii}(\theta)\ddot{\theta}_i + d_i(\theta, \dot{\theta}, \ddot{\theta}) = \tau_i(t) \tag{III.5}$$

$$d_i(\theta, \dot{\theta}, \ddot{\theta}) = \sum_{\substack{j=1 \\ j \neq i}}^n m_{ij}(\theta)\ddot{\theta}_j + h_i(\theta, \dot{\theta}) + F_i$$

La structure du régulateur est la même que celle donnée par (III.2), ainsi pour chaque articulation on définit la loi de commande suivante:

$$\tau_i(t) = \hat{m}_{ii} u_i + \hat{d}_i \quad i=1, \dots, n \tag{III.6}$$

Ce régulateur est représenté sur la figure (III.3). On obtient aussi l'équation de l'erreur en boucle fermée de la  $i^{\text{ème}}$  articulation

$$\ddot{e}_i + k_v \dot{e}_i + k_p e_i = \hat{m}_{ii}^{-1} ((m_{ii} - \hat{m}_{ii})\ddot{\theta}_i + d_i - \hat{d}_i), \quad i=1, \dots, n \tag{III.7}$$

Deux RNA sont synthétisés pour chaque articulation. Les poids des réseaux sont ajustés à chaque instant par la rétropropagation de l'erreur filtrée [Oza 91][SER 89].

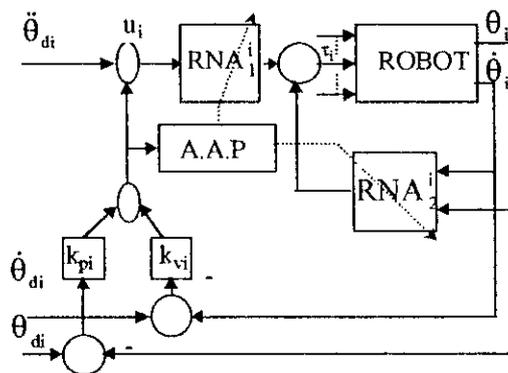


Figure III.3. Commande adaptative décentralisée par RNA de la  $i^{\text{ème}}$  articulation.

### III.4.1 Structure des RNA Utilisé

Les structure des RNA<sub>1</sub><sup>i</sup> et RNA<sub>2</sub><sup>i</sup> sont présentées dans la figure (III.4). Les réseaux utilisés sont à trois couches ; une couche d'entrée, une couche cachée et une couche de sortie. Les entrées des RNA<sub>1</sub><sup>i</sup> et RNA<sub>2</sub><sup>i</sup> sont  $\theta_i$ , et  $\theta_i, \dot{\theta}_i$  respectivement, et les sorties sont données par les équations suivantes :

$$\hat{m}_{ii} = RNA_1^i = W_1^{iT} \sigma(V_1^i \theta_i) \tag{III.8}$$

$$\hat{h}_i = RNA_2^i = W_2^{iT} \sigma(V_2^{iT} [\theta_i, \dot{\theta}_i]^T) \tag{III.9}$$

ou  $\sigma(\cdot)$  est la fonction d'activation des neurones de la couche cachée ' $\sigma(x)=\tanh(x)$ '. Notons que les fonctions d'activation de la couche d'entrée et celles de sortie sont des fonctions linéaires.

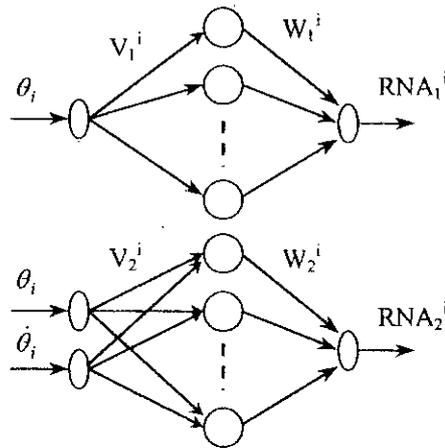


Figure III.4 Structure des RNA

### III.4.2 Algorithme d'Apprentissage des RNA

L'entraînement des poids des RNA est effectué par la rétropropagation de l'erreur filtrée  $r_i(t)$  [OZA 91][SER 89] définie par :

$$r_i(t) = \omega_{pi} (\theta_{di} - \theta_i) + \omega_{vi} (\dot{\theta}_{di} - \dot{\theta}_i) \tag{III.10}$$

$\omega_{pi}$  et  $\omega_{vi}$  sont respectivement les pondérations de l'erreur et la variation de l'erreur. Les poids de la couche de sortie du réseau RNA<sub>1</sub><sup>i</sup> ( $i=1,2$  et  $i=1, \dots, n$ ) sont adaptés par :

$$\Delta W_1^i = \mu_i \delta_i^i \sigma(V_1^{iT} X_1^i) \tag{III.11}$$

où  $X_1^i = \theta_i$ ,  $X_2^i = [\theta_i, \dot{\theta}_i]^T$  sont les vecteurs d'entrée de RNA<sub>1</sub><sup>i</sup> et RNA<sub>2</sub><sup>i</sup> respectivement.

$$\begin{aligned} \delta_1^i &= u_i r_i(t) \\ \delta_2^i &= r_i(t) \end{aligned} \tag{III.12}$$

Les poids de la couche cachée sont adaptés par :

$$\Delta V_i^i = \mu_i^i X_i^i \left( \sigma' (V_i^{i^T} X_i^i) W_i^i \delta_i^i \right) \quad (\text{III.13})$$

$$\mu_i^i = \frac{\alpha_i^i}{1 + \|J_i^i\|^2} \quad (\text{III.14})$$

avec  $J_1^i = u_i \frac{\partial \hat{m}_{ii}}{\partial Z_1^i}$ ,  $J_2^i = \frac{\partial \hat{d}_i}{\partial Z_1^i}$  et  $Z_1^i = [V_i^i, W_i^i]^T$ .

et  $\alpha_i^i$  un gain positif inférieur à un.

### Résultats de Simulation

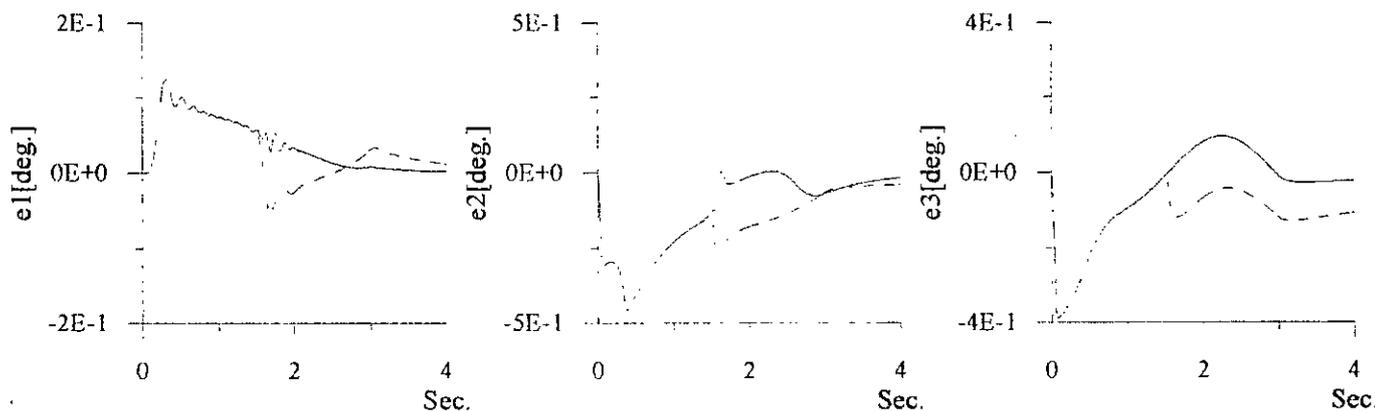
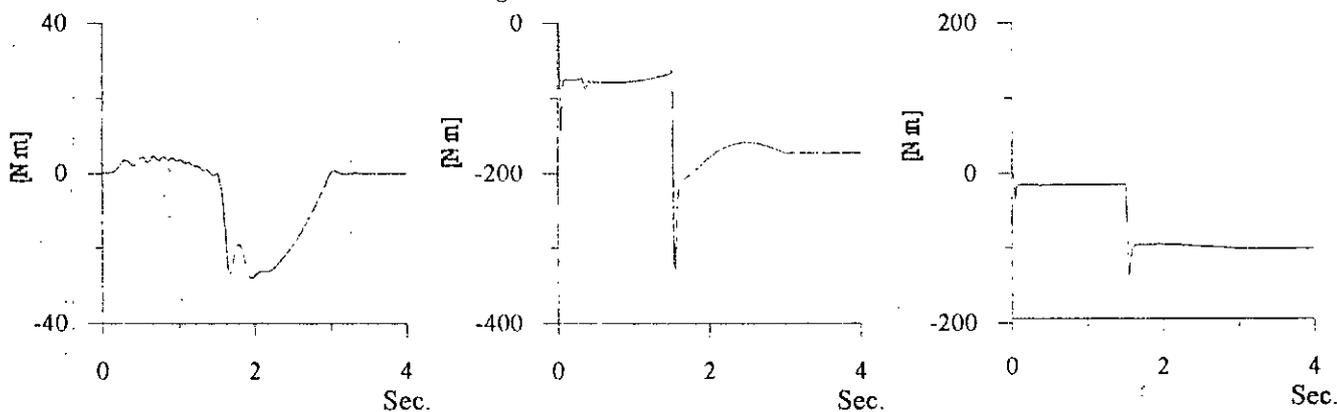
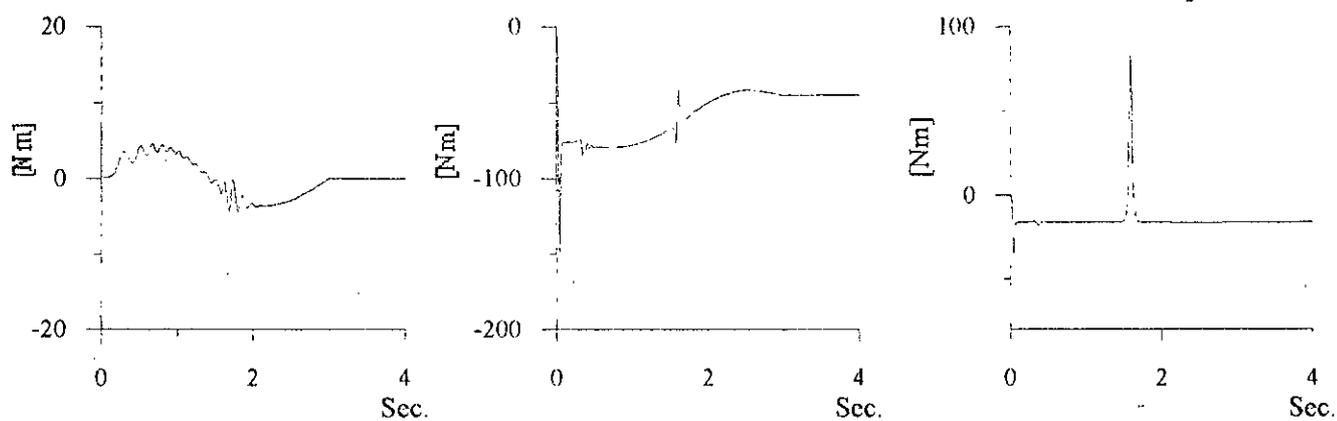
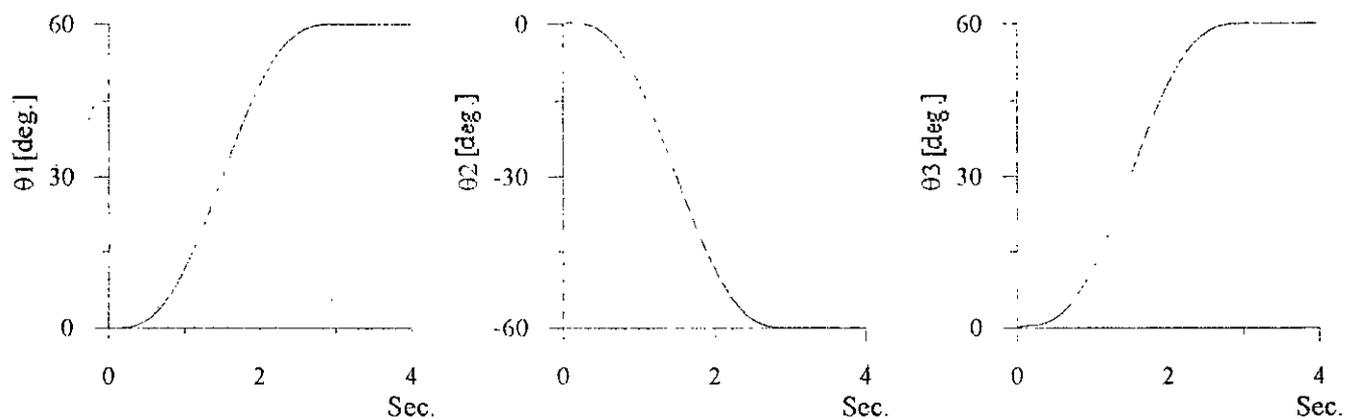
Nous avons opté pour des RNA chacun a trois couches, la couche cachée possède 5 neurones. Les pondérations des erreurs filtrées sont données par :

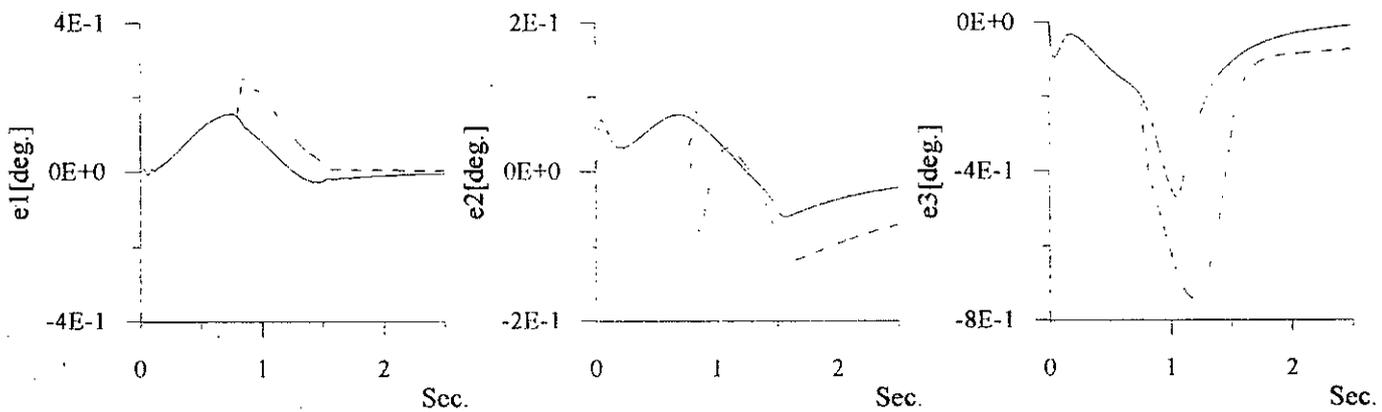
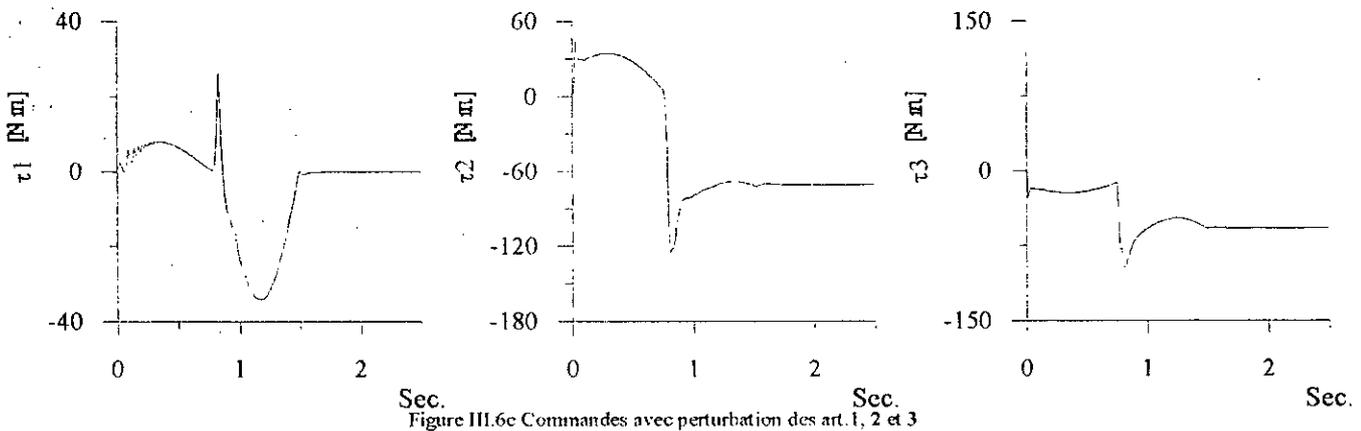
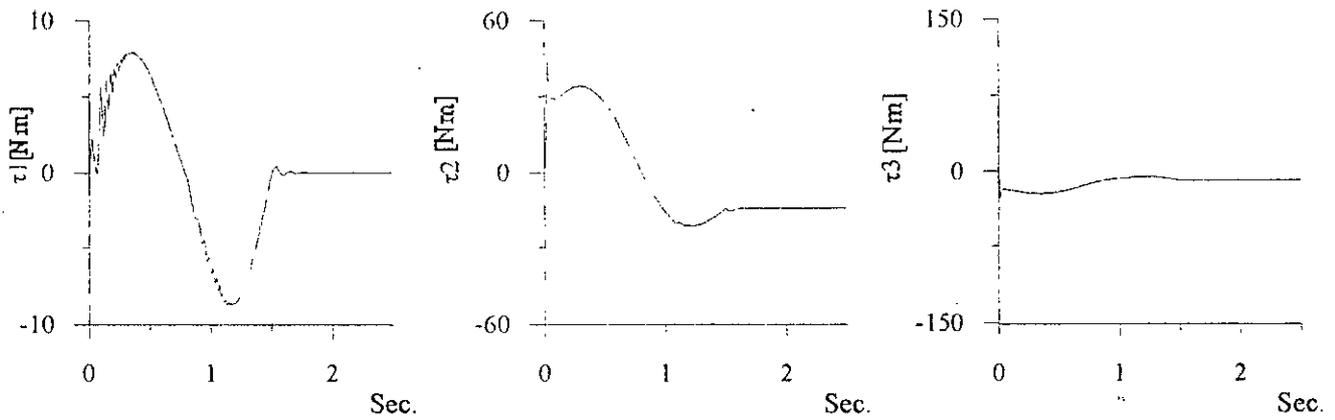
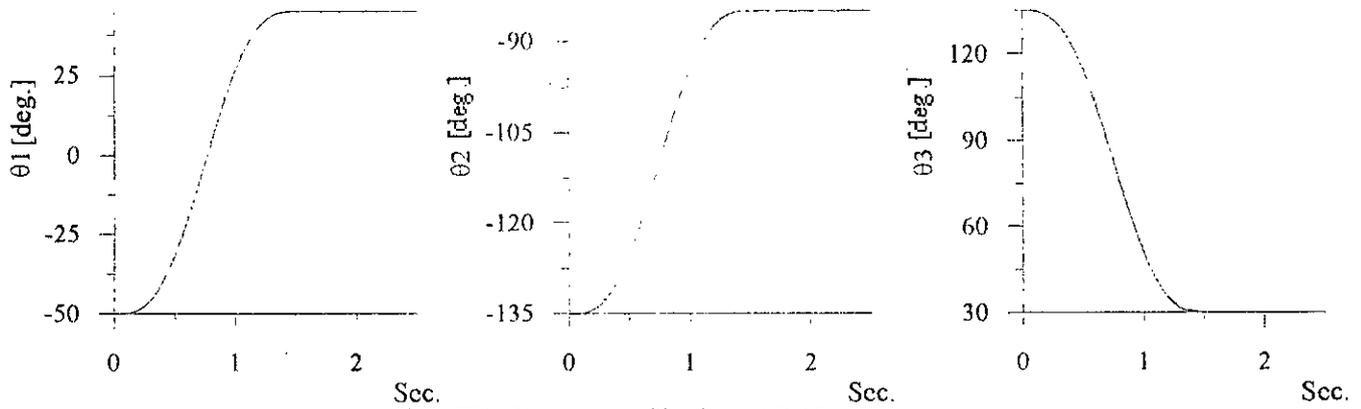
$\omega_{p1}=30$ ,  $\omega_{p2}=40$ ,  $\omega_{p3}=12$ ,  $\omega_{v1}=20$ ,  $\omega_{v2}=20$ ,  $\omega_{v3}=4$ .

Les paramètres de PD sont  $\xi=1$  et  $\omega_0=20$ .

Pour les simulations, nous avons utilisé une trajectoire cycloïdale assurant une continuité en position, vitesse et accélération. La figure (III.5) montre les performances et l'efficacité du schéma de commande proposé, pour la commande en poursuite de la trajectoire allant de  $[0^\circ, 0^\circ, 0^\circ]$  jusqu'à  $[60^\circ, -60^\circ, 60^\circ]$  en 3 seconde sans et avec perturbation du robot PUMA 560.

Pour tester la capacité de poursuite de la commande développée lors des trajectoires rapides du bras de robot nous avons utilisé la trajectoire de Leahy sans et avec perturbation utilisée précédemment. Les réponses obtenues par cette architecture de commande sont présentées sur la figure (III.6). Nous constatons que cette variation n'affecte pas les performances de poursuite.





## III.5 Commande Adaptative Décentralisée par RBF

### III.5.1 RBF à Centres Fixes

Dans la section précédente (III.4), nous avons développé une commande adaptative décentralisée en utilisant les RNA statiques. Or, il est judicieux d'utiliser les différents types de réseaux présentés dans le premier chapitre, afin de mettre en évidence les caractéristiques ainsi que le champ d'application de chacun d'eux.

Soit le modèle dynamique du robot donné par l'équation (III.5). La commande est donnée par l'équation (III.6).  $\hat{m}_i$  et  $\hat{d}_i$  sont respectivement les sorties des deux réseaux gaussiens à centres fixes  $RNA_1^i$  et  $RNA_2^i$ .

$$\hat{m}_i = RNA_1^i = \sum_{l \in I_0} \hat{c}_{1,l}^i g_\sigma(x_1^i - \xi_{1l}^i) \quad (III.15)$$

$$\hat{d}_i = RNA_2^i = \sum_{l \in I_0} \hat{c}_{2,l}^i g_\sigma(x_2^i - \xi_{2l}^i) \quad (III.16)$$

où  $x_1^i = \theta_i, x_2^i = [\theta_i, \dot{\theta}_i]^T$  sont respectivement les vecteurs d'entrées du  $RNA_1^i$  et  $RNA_2^i$ .

### Algorithme d'Apprentissage

Les poids de la couche d'entrée (les centres) sont fixés au préalable et les poids de la couche de sortie du réseau  $RNA_1^i$  ( $l=1,2$  et  $i=1,\dots,n$ ) sont adaptés par :

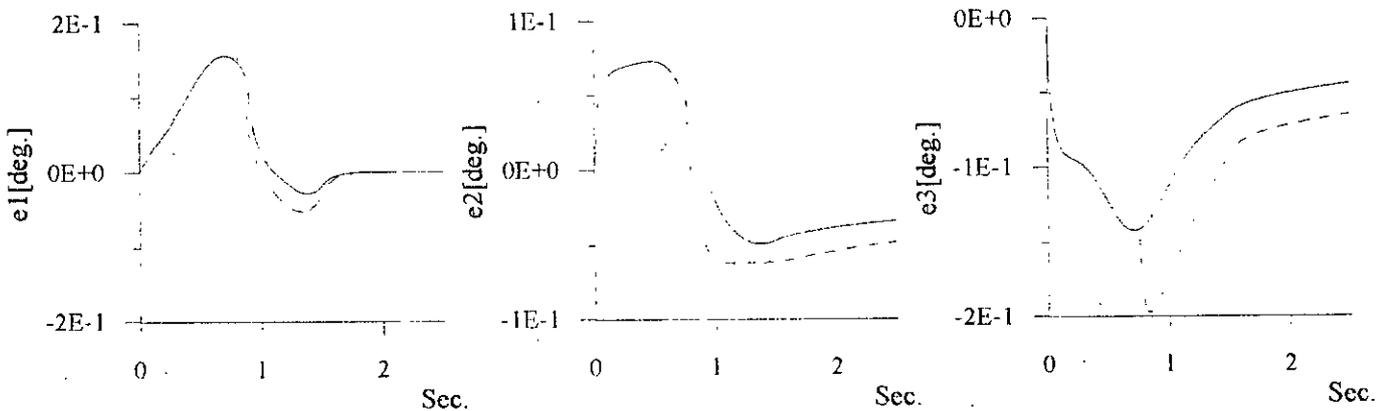
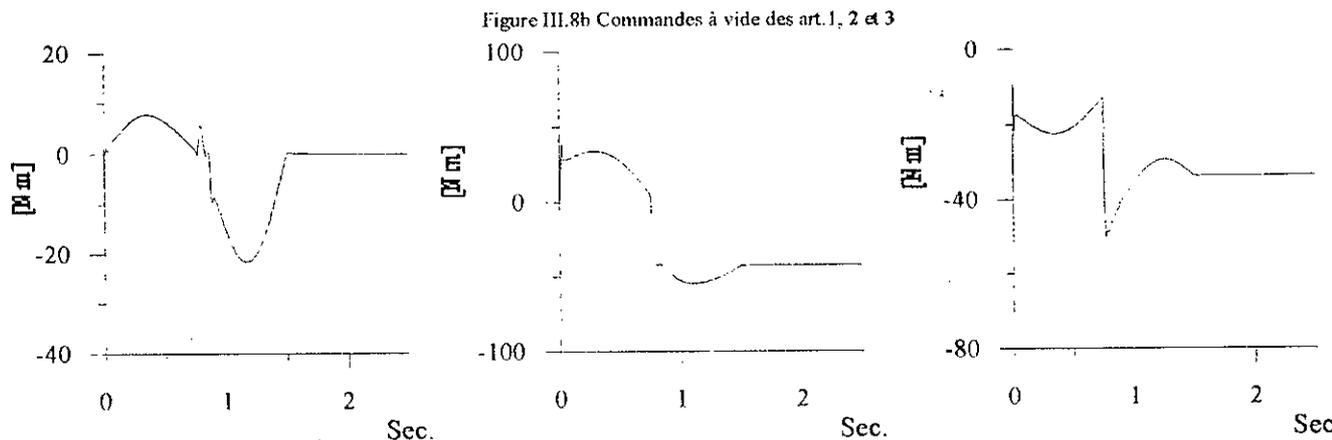
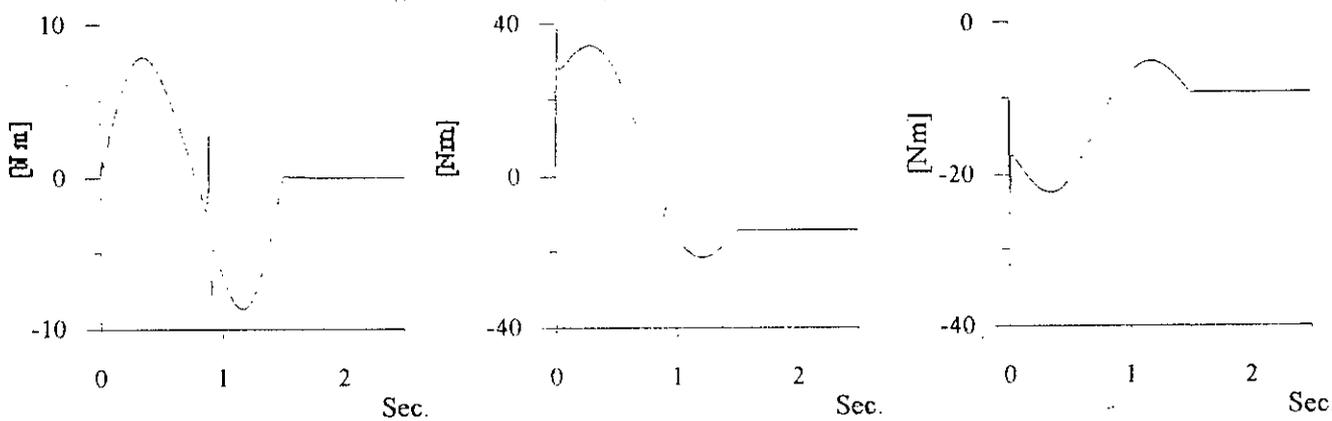
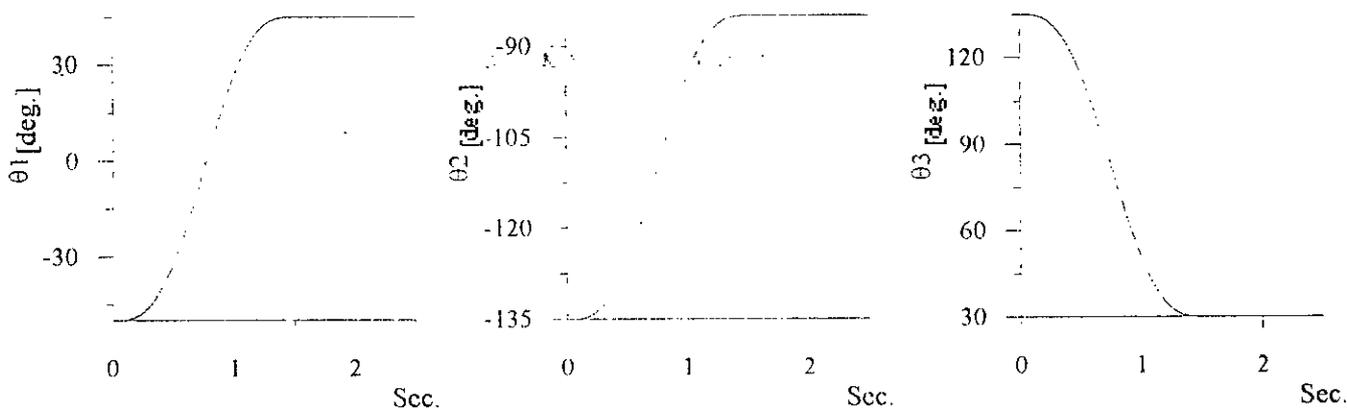
$$\Delta \hat{c}_{ll}^i = \mu_{ll}^i \delta_l^i g_\sigma(x_1^i - \xi_{ll}^i) \quad (III.17)$$

avec  $\delta_l^i$  est définie par (III.12) et  $\mu_{ll}^i$  par (III.14).

### Etude par simulation

En se basant sur la trajectoire désirée, l'ensemble  $A_{di}$  est choisi en utilisant les bornes min et max de  $\theta_{di}$  et  $\dot{\theta}_{di}$  ( $i = 1, \dots, n$ ).

Les résultats (figures III.7 et III.8) obtenus par simulation sur le robot PUMA 560 montrent l'efficacité de cette stratégie de commande par RBF à centres fixes en terme de poursuite de trajectoires de références pour le robot manipulateur ainsi que la robustesse vis-à-vis des perturbations.



### III.5.2 RBFN à Centres Adaptatifs

Nous avons déjà mentionné dans le chapitre I que l'approximation par les réseaux RBFN à centres fixes est basée sur la reconstruction exacte de la fonction non linéaire à partir de certaines données (l'espace d'entrée, la borne supérieure de la fonction, ...etc). En effet, la taille du réseau est généralement très élevée. Afin de surmonter cet handicap, il est judicieux de laisser libre la disposition des centres et à l'aide d'un algorithme d'adaptation, les centres sont ajustés pour donner la meilleure combinaison possible des paramètres dans le but de produire la meilleure approximation.

Les réseaux gaussiens ont pour rôle d'estimer  $m_{ii}$  et  $d_i$  ( $i=1, \dots, n$ ).

La commande est donnée par l'équation (III.6) et la structure de ce régulateur est représentée sur la figure (III.3).

Les lois d'adaptation des réseaux à base radiale généralisés sont :

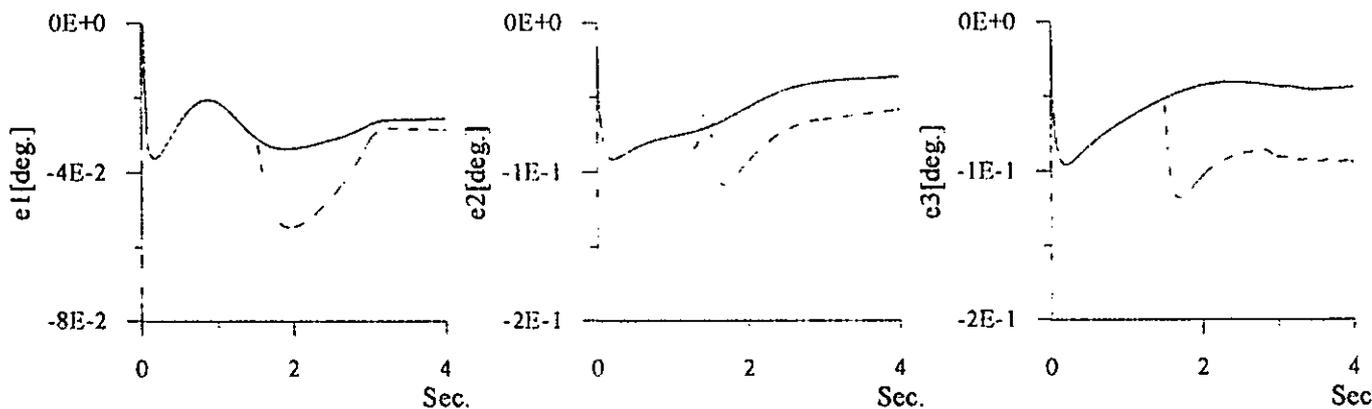
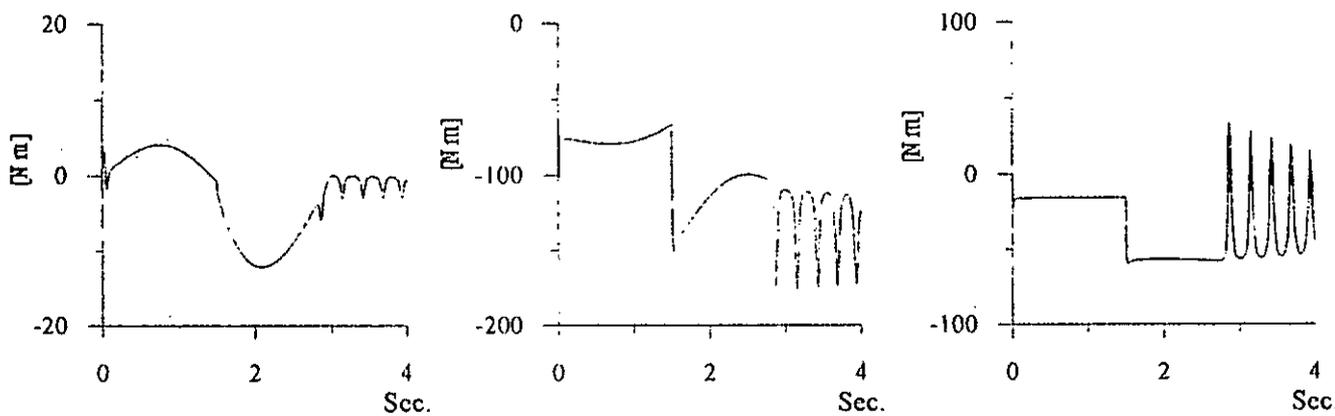
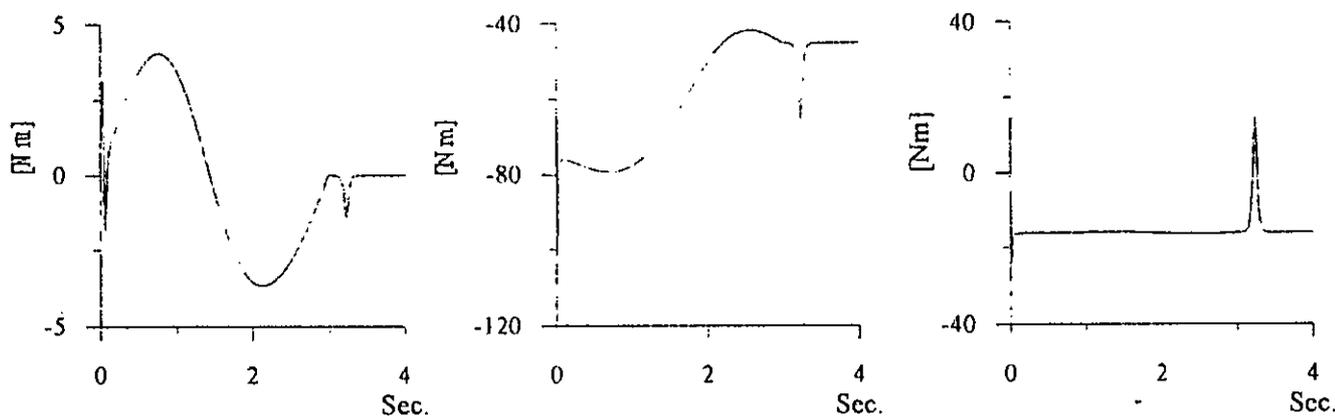
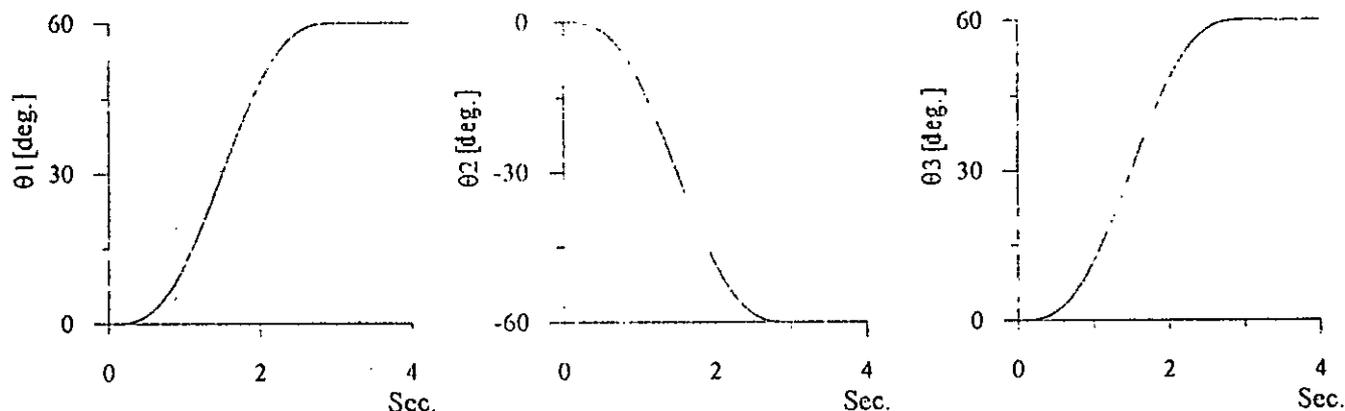
$$\Delta \hat{c}_i^j = \mu_i^j \delta_i^j g_\sigma(x_i^j - \xi_{ii}^j) \quad (III.18)$$

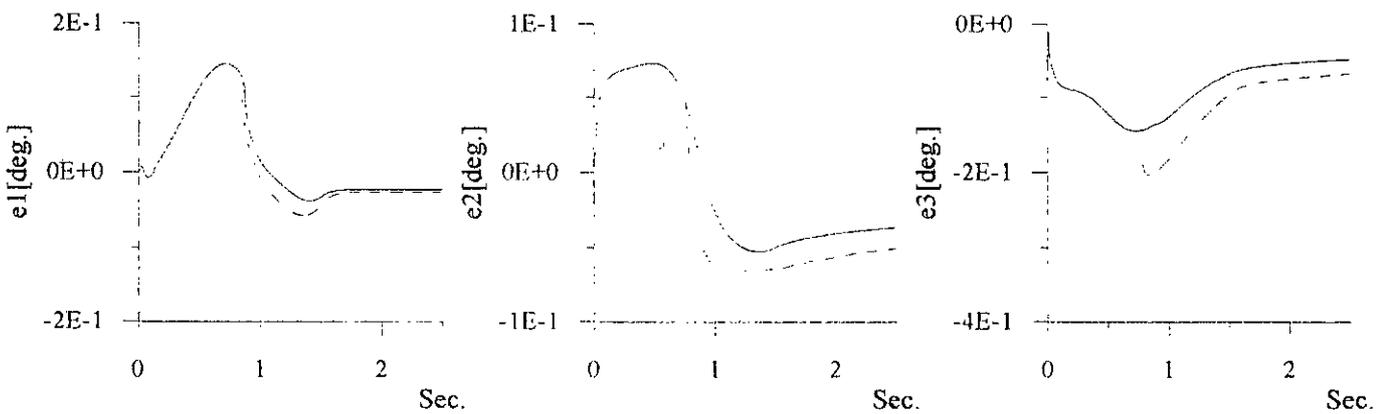
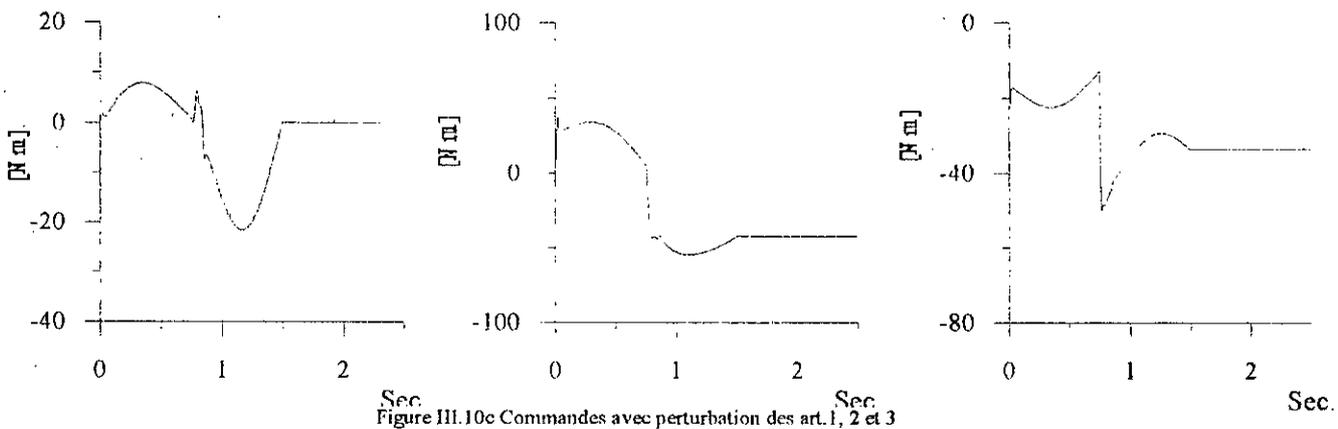
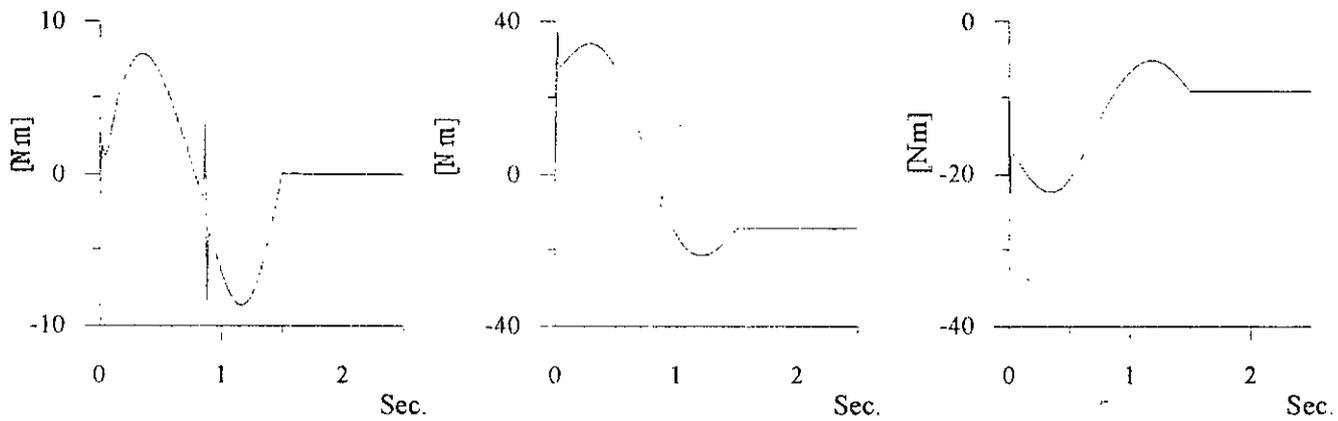
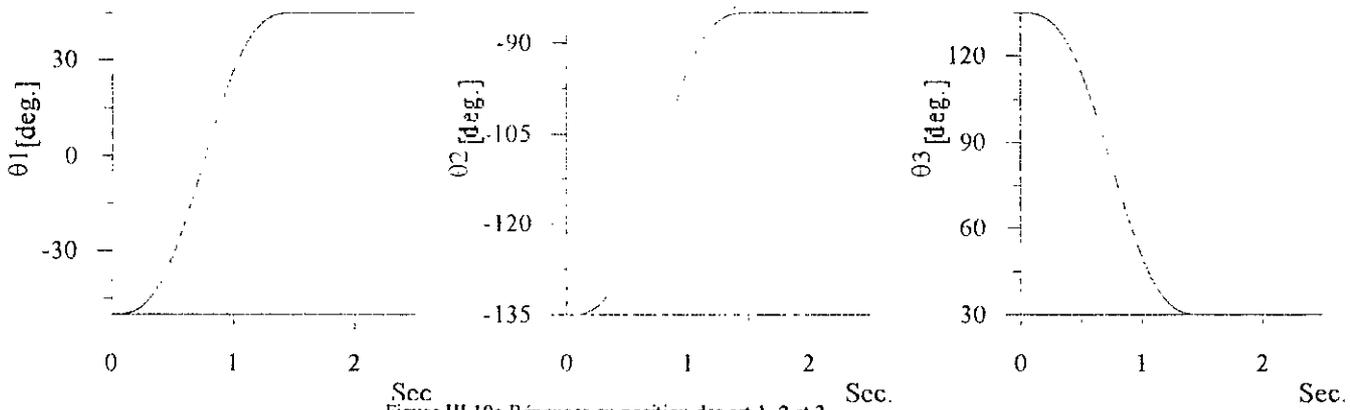
$$\Delta \hat{\xi}_{ii}^j = \mu_i^j \frac{2}{\sigma^2} \delta_i^j (x_i^j - \xi_{ii}^j) g_\sigma(x_i^j - \xi_{ii}^j) \hat{c}_{ii}^j \quad (III.19)$$

avec  $\delta_i^j$  est définie par (III.12) et  $\mu_i^j$  est définie par (III.14).

#### **Etude par Simulation**

Les résultats (figures III.9 et III.10) obtenus par simulation sur le robot PUMA 560 mettent en évidence les performances et l'efficacité de cette stratégie de commande par les réseaux à base radiale généralisés en terme de poursuite des trajectoires de références ainsi que la robustesse vis-à-vis des perturbations.





### III.6 Commande Adaptative Décentralisée par RNAR

Dans cette section, nous présentons l'utilisation des réseaux récurrents dans la commande neuronale adaptative décentralisée.

$m_{ii}$  et  $d_i$  sont estimés par des RNAR définis par l'équation (I.98). Les entrées externes sont respectivement  $\theta_i$  et  $[\theta_i, \dot{\theta}_i]$  ( $i=1, \dots, n$ ).

Les lois d'adaptations des réseaux récurrents sont données par l'équation (I.100) avec :

$$V_d^2 - V_k^2 = \delta_i' \quad (\text{III.20})$$

avec  $\delta_i'$  est définie par l'équation (III.12)

#### **Etude par Simulation**

Nous avons opté pour des RNAR chacun à trois couches, la couche cachée possède 5 neurones et  $n_y = 2$  (deux retours de la sortie).

Les figures (III.11) et (III.12) montrent les performances du schéma de commande proposé par les réseaux récurrents sans et avec perturbation, pour la commande en poursuite de trajectoire des trois premières articulations du robot PUMA 560. Nous constatons que l'utilisation des réseaux récurrents donne des résultats satisfaisant pour la commande en poursuite des robots manipulateurs.

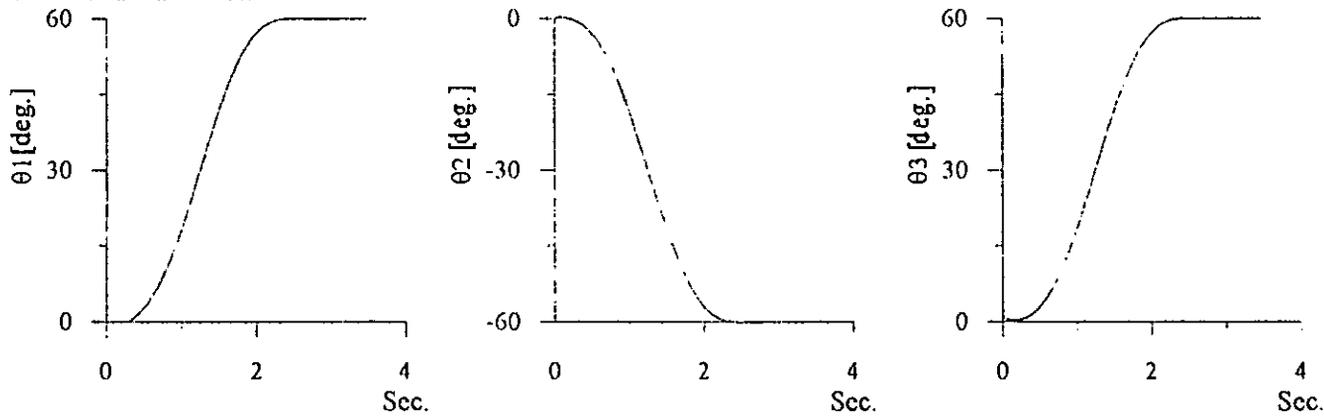


Figure III.11a Réponses en position des art.1, 2 et 3

--- à vide. - - - avec perturbation.

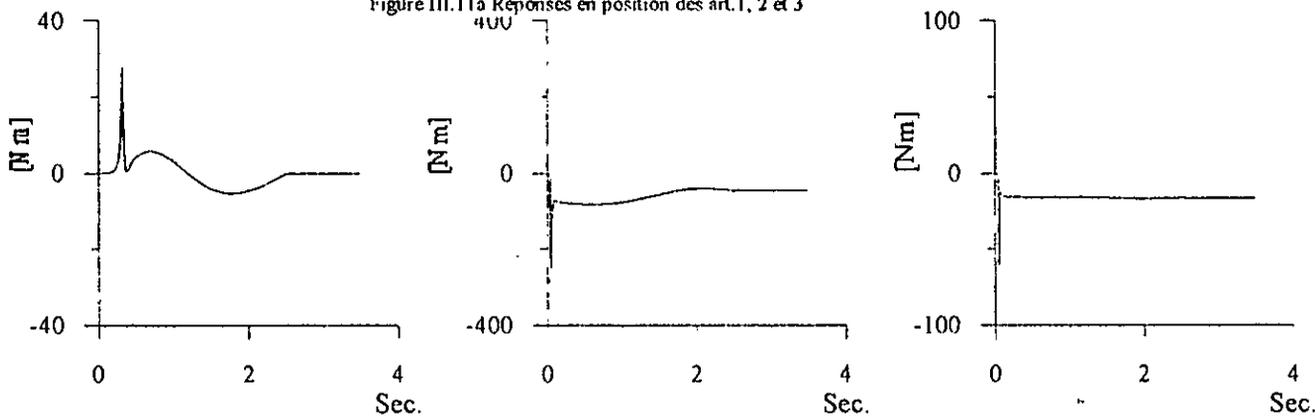


Figure III.11b Commandes à vide des art.1, 2 et 3

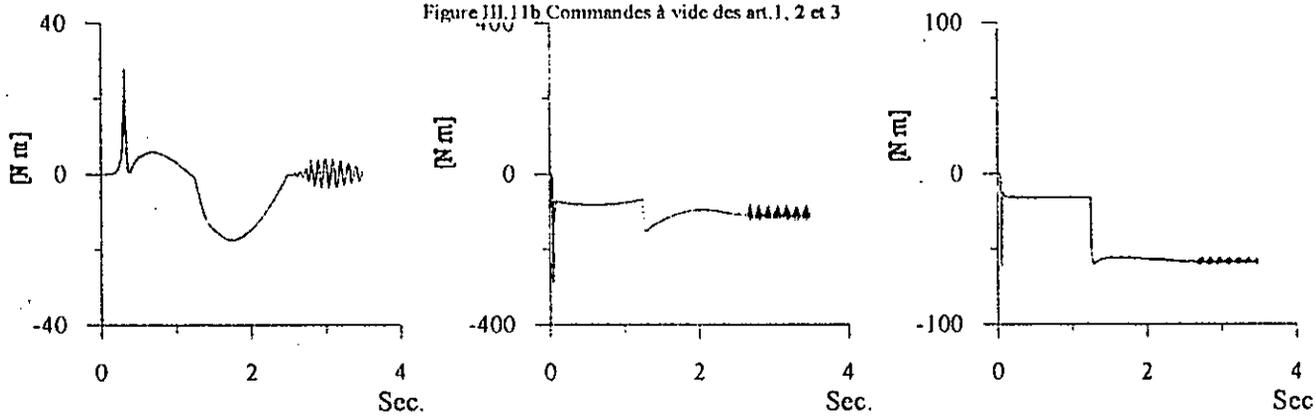


Figure III.11c Commandes avec perturbation des art.1, 2 et 3

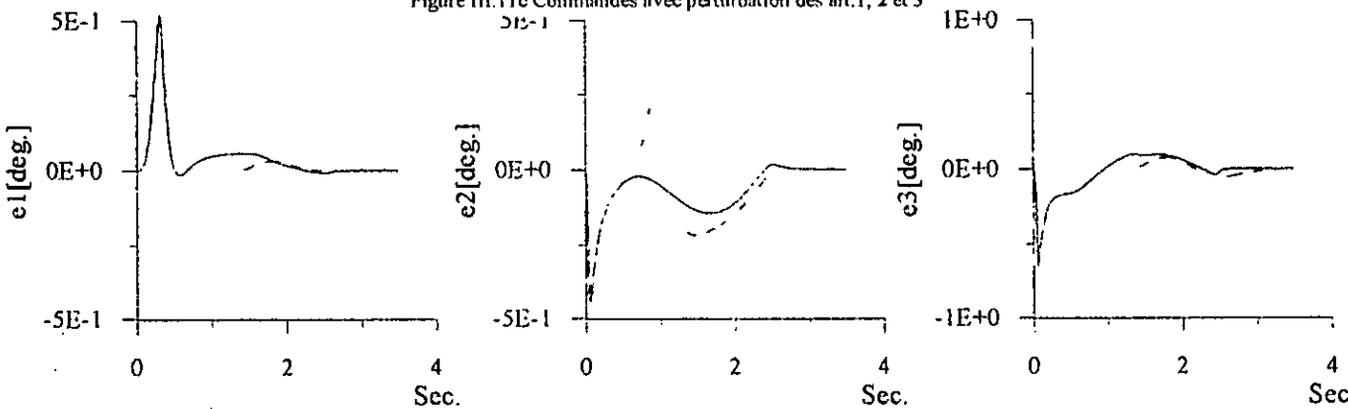
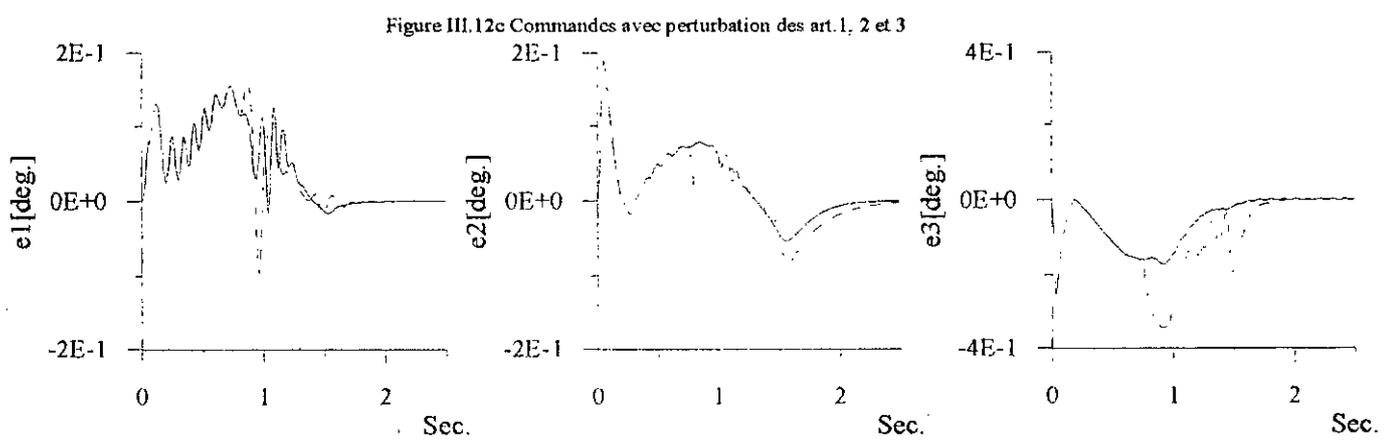
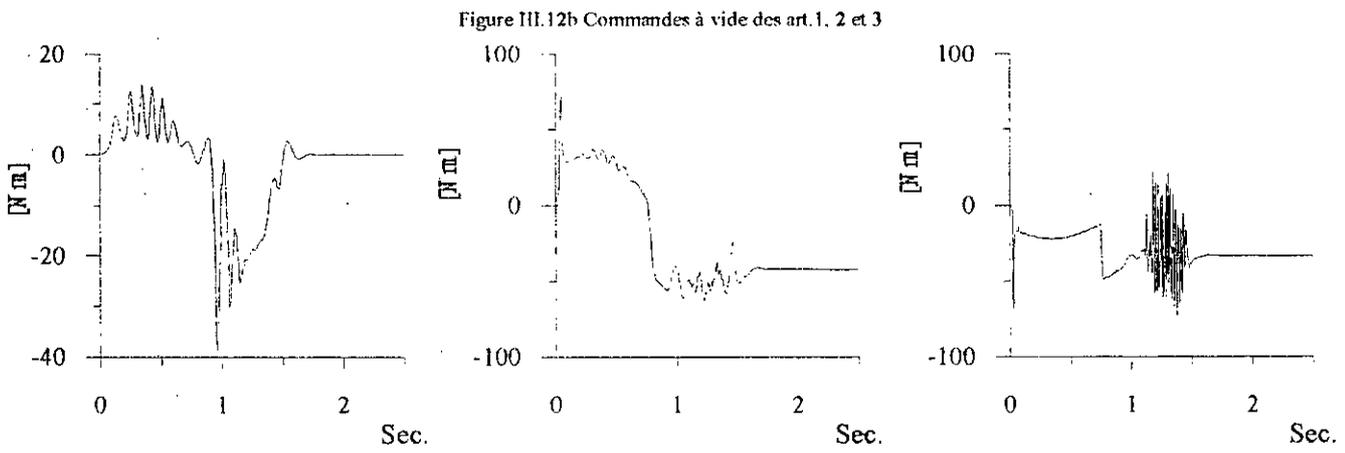
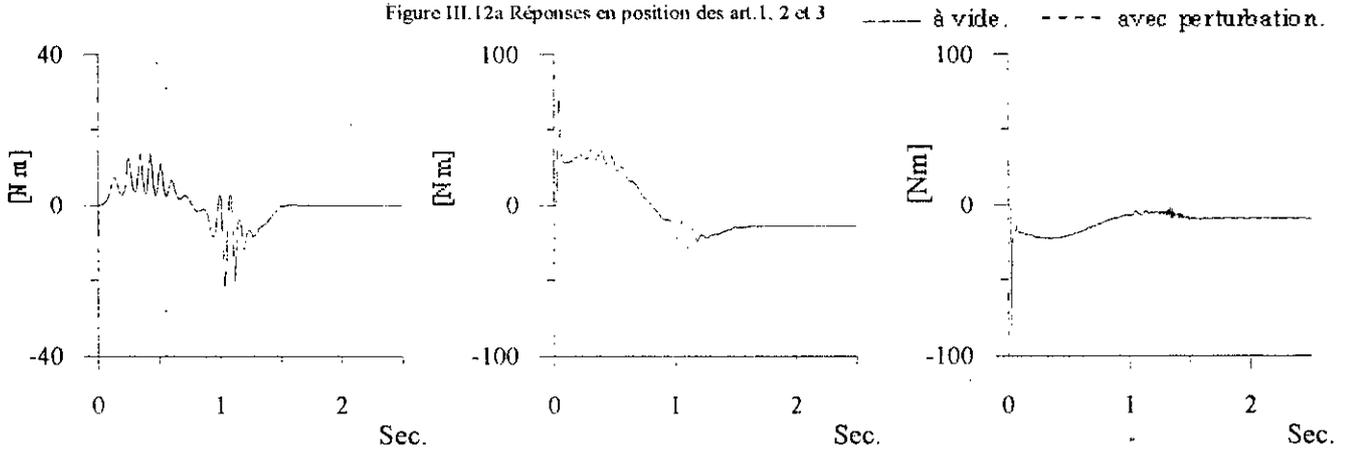
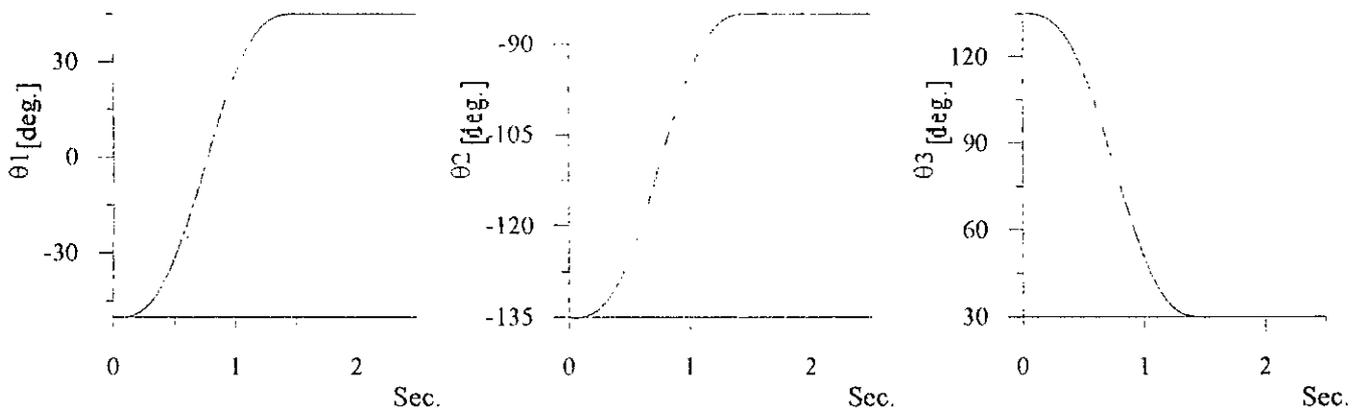


Figure III.11d Erreurs des art.1, 2 et 3

— à vide. - - - avec perturbation.



### **III.7. Conclusion**

Dans ce chapitre est étudiée la commande linéarisante des robots manipulateurs utilisant les RNA mis au point par Ozaki en 1991. En effet, cette technique de commande est impraticable et même impossible dans le cas, où le degré de liberté est très élevé. Pour remédier à cet handicap, nous avons étudié et développé la commande adaptative décentralisée par RNA statique par le biais de l'extension, au cas décentralisé, de la commande linéarisante adaptative par RNA. Ensuite, les réseaux RBFG à centres fixes et à centres adaptatifs ainsi que les réseaux récurrents ont été utilisés dans cette technique de commande.

La stratégie de commande développée dans le présent chapitre présente un double intérêt. La décentralisation de la commande permet de réduire considérablement le calcul. L'utilisation des réseaux de neurones a permis l'amélioration de la robustesse de la commande en éliminant le défaut de la méthode du couple calculé classique.

L'étude envisagée ici montre l'efficacité de la stratégie de commande neuronale adaptative décentralisée par différents types de RNA en termes de poursuite de trajectoires de références et de robustesse paramétrique, pour les systèmes dynamiques complexes tel que le robot manipulateur multi articulations.

---

# *Chapitre 4*

## *Commande Adaptative*

### *Indirecte Décentralisée par RNA*

---

#### **IV.1 Introduction**

Les robots manipulateurs sont devenus de plus en plus répandus dans le domaine de l'automatisation industrielle. La rapidité et la précision dans la commande en poursuite de trajectoires sont indispensables pour l'exécution des tâches multiples effectuées par ces robots [MSA 94],[SLO 88].

La commande adaptative a été utilisée pour compenser les effets des incertitudes structurées [AST 89][LAN 86][GOO 84]. En générale, l'élimination de l'effet des incertitudes non structurées n'est pas garantie. Vu les possibilités offertes par les RNA et leurs capacités prouvées dans le domaine de la commande et l'identification des systèmes dynamiques complexes, leur utilisation peut constituer une alternative pour pallier les problèmes des incertitudes non structurées [FRE 92][KAR 93][MIL 92][NAR 90a]. L'adaptivité et la non-linéarité des RNA font d'eux un moyen adéquat de prédiction, commande et optimisation des processus industriels. Dans les dernières années, les RNA ont été appliqués à beaucoup de systèmes de commande et ils ont souvent, donnés satisfaction.

La plus part des algorithmes de commande sont développés dans un contexte centralisé. Dans cette structure de commande, l'information est traitée dans une seule unité de commande nécessitant la mesure de tous les états (Sorties) du système. Cette structure rend généralement le calcul lent et/ou complexe. Pour pallier cet inconvénient, les chercheurs se sont intéressés lors des deux dernières décennies à l'utilisation de la commande décentralisée [BOU 97b][IOU 86][SER 89]. Cette dernière consiste à décomposer le système en plusieurs sous systèmes interconnectés pouvant être géographiquement éloigné, chaque sous système est commandé localement par une station de commande qui n'a accès qu'aux informations locales du sous système et permettant ainsi la simplification de l'implémentation des algorithmes par rapport aux lois de commandes centralisées [BOU 96][BOU 98a].

Dans le présent chapitre, une nouvelle approche de la commande adaptative indirecte basée sur la combinaison des RNA et la méthode classique dite du couple calculé sera présentée. Les RNA sont utilisés pour donner un modèle de représentation décentralisé de la dynamique directe du robot manipulateur. La loi de commande est ensuite générée à partir de ce modèle en utilisant la méthode du couple calculé.

## IV.2 Commande Adaptative Indirecte Décentralisée par RNA Statique

Dans le domaine de la robotique, l'élaboration des modèles nécessite une étude approfondie et détaillée de la structure. Des efforts considérables ont été fournis pour l'obtention des modèles dynamiques des différents robots [SLO 88][LAB 97]. Vu le caractère fortement non linéaire et le couplage existant, l'évaluation des paramètres est souvent difficile.

Le modèle dynamique d'un bras de robot manipulateur à  $n$  degrés de liberté est donné par :

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) + G(\theta) \quad (\text{IV.1})$$

avec

- $\tau$  ( $n \times 1$ ) : Vecteur des forces et/ou des couples.
- $M$  ( $n \times n$ ) : Matrice d'inertie symétrique et définie positive.
- $h$  ( $n \times 1$ ) : Vecteur des forces de coriolis et centripètes.
- $\theta$  ( $n \times 1$ ) : Vecteur des coordonnées généralisées des articulations.
- $\dot{\theta}$  ( $n \times 1$ ) : Vecteur des vitesses.
- $\ddot{\theta}$  ( $n \times 1$ ) : Vecteur des accélérations.
- $G$  ( $n \times 1$ ) : Vecteur des forces de gravité.

La synthèse de commande adaptative est basée sur un modèle du processus à commander. Ce modèle peut être élaboré analytiquement en utilisant des lois physiques régissant le fonctionnement du système. Cependant, dans certains cas, il peut être très complexe pour la conception des contrôleurs.

Le problème de la commande adaptative indirecte décentralisée, en utilisant les réseaux de neurones artificiels, est formulé de la manière suivante :

Etant donné le modèle dynamique du robot (éq.IV.1) ; en multipliant les deux membres par  $M^{-1}(\theta)$ , on obtient :

$$\ddot{\theta} = -M(\theta)^{-1} [h(\theta, \dot{\theta}) + G(\theta)] + M(\theta)^{-1} \tau \quad (\text{IV.2})$$

d'où

$$\ddot{\theta} = f(\theta, \dot{\theta}) + g(\theta) \tau \quad (\text{IV.3})$$

$$f(\theta, \dot{\theta}) = -M(\theta)^{-1} [h(\theta, \dot{\theta}) + G(\theta)]$$

$$g(\theta) = M(\theta)^{-1}$$

Dans un contexte décentralisé, et pour le sous système  $i$ , l'équation (IV.3) devient :

$$\ddot{\theta}_i = f_i(\theta_i, \dot{\theta}_i) + g_i(\theta_i) \tau_i + \varepsilon_i(t) \quad , j=1, \dots, n \quad (\text{IV.4})$$

où  $n$  est le nombre d'articulation.

$\varepsilon_i(t)$  un terme caractérisant les perturbations, les erreurs de modélisations et l'effet des interconnexions.

L'objectif est de générer une loi de commande locale basée sur l'équation (IV.4) en utilisant les estimées des fonctions  $f_i$  et  $g_i$ . La synthèse de la commande s'effectue, donc, en deux étapes : Identification et commande.

**A. Etape d'identification**

De l'équation (IV.4) on peut écrire

$$\ddot{\theta}_i = \hat{f}_i(\theta_i, \dot{\theta}_i) + \hat{g}_i(\theta_i) \tau_i \tag{IV.5}$$

$\hat{f}_i(.)$  et  $\hat{g}_i(.)$  sont respectivement les estimées de  $f_i(.)$  et  $g_i(.)$ , générées respectivement par deux réseaux RNA<sub>1</sub><sup>i</sup> et RNA<sub>2</sub><sup>i</sup>, dont les poids sont ajustés à chaque instant de temps par un algorithme d'adaptation paramétrique (AAP) [BOU 98][LAB 97] utilisant la rétropropagation de l'erreur pondérée [SER 89].

La relation donnée par l'équation (IV.5) n'est qu'une représentation pour la commande du robot ; c'est un prédicteur d'accélération qui ne tient pas compte des couplages existants entre les articulations du robot.

**B. Etape de commande**

Notre but est de concevoir une commande de telle sorte que le robot suit une trajectoire de référence. Pour cela, la loi de commande appliquée à chaque articulation  $i$ , calculée suivant la méthode du couple calculé [MSA 94][SLO 88], est donnée par la relation suivante

$$\tau_i = \frac{u_i - \hat{f}_i(.)}{\hat{g}_i(.)} \tag{IV.6}$$

$$u_i = \ddot{\theta}_{d_i} + 2\lambda_1 \dot{e}_i + \lambda_1^2 e_i$$

où  $\hat{f}_i(.)$  et  $\hat{g}_i(.)$  sont générées par les RNA suscités suivant les expressions (IV.7) et (IV.8).

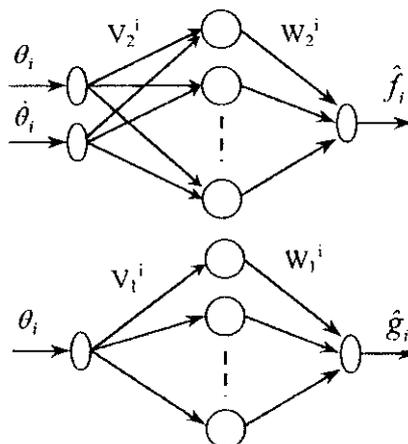


Figure IV.1. Structure des RNA

$$\hat{f}_i = W_2^{iT} \sigma(V_2^{iT} [\theta_i, \dot{\theta}_i]^T) \tag{IV.7}$$

$$\hat{g}_i = W_1^{iT} \sigma(V_1^i \theta_i) \tag{IV.8}$$

Les topologies des RNA utilisés sont illustrées par la figure 1 chaque réseau est constitué de trois couches, une couche d'entrée, une couche cachée et une couche de sortie. Les entrées de RNA<sub>1</sub><sup>i</sup> et RNA<sub>2</sub><sup>i</sup> sont respectivement  $X_1^i = [\theta_i, \dot{\theta}_i]$  et  $X_2^i = \theta_i$ .  $\sigma(\cdot)$  est la fonction d'activation des neurones de la couche cachée ( $\sigma(x) = \tanh(x)$ ), notons que, la fonction d'activation de la couche d'entrée et celle de la couche de sortie sont choisies linéaires.

Le schéma de la figure (IV.2) illustre l'algorithme de commande utilisé.

**Algorithme d'adaptation des réseaux de neurones**

L'entraînement des poids des RNA est effectué par la rétropropagation de l'erreur pondérée  $r_i(t)$  définie par :

$$r_i(t) = \omega_{p_i} (\theta_{d_i} - \theta_i) + \omega_{v_i} (\dot{\theta}_{d_i} - \dot{\theta}_i) \tag{IV.9}$$

$\omega_{p_i}$  et  $\omega_{v_i}$  sont respectivement les pondérations de l'erreur et la variation de l'erreur.

Les poids de la couche de sortie du réseau RNA<sub>1</sub><sup>i</sup> ( $l=1,2$  et  $i=1, \dots, n$ ) sont adaptés par :

$$\Delta W_1^i = -\mu_i \delta_i' \sigma(V_1^{iT} X_1^i) \tag{IV.10}$$

où  $X_1^i = [\theta_i, \dot{\theta}_i]^T$ ,  $X_2^i = \theta_i$  sont respectivement les vecteurs d'entrée de RNA<sub>1</sub><sup>i</sup> et RNA<sub>2</sub><sup>i</sup>.

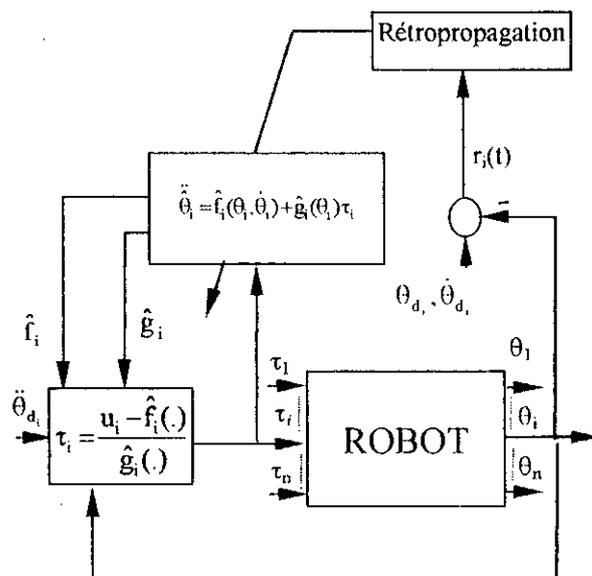


Figure IV.2. Principe de la Commande Adaptative Indirecte Décentralisée par RNA.

$$\begin{aligned}\delta_1^i &= r_i(t) \\ \delta_2^i &= \tau_i r_i(t)\end{aligned}\quad (IV.11)$$

Les poids de la couche cachée sont adaptés par :

$$\Delta V_i^i = -\mu_i^i X_i^i \left( \sigma'(V_i^{i^T} X_i^i) W_i^i \delta_i^i \right)^T \quad (IV.12)$$

$$\mu_i^i = \frac{\alpha_i^i}{1 + \|J_i^i\|^2} \quad (IV.13)$$

avec  $J_1^i = \frac{\partial \hat{f}_i}{\partial Z_1^i}$ ,  $J_2^i = \tau_i \frac{\partial \hat{g}_i}{\partial Z_2^i}$  et  $Z_i^i = [V_i^i, W_i^i]^T$ .

$\alpha_i^i$  est un gain positif inférieur à un.

### Etude par Simulation

Les simulations ont été effectuées sur les trois premières articulations d'un robot de type PUMA 560. Le schéma de commande développé a été testé pour la poursuite d'une trajectoire cycloïdale assurant une continuité en position, vitesse et accélération et allant de  $[0^\circ, 0^\circ, 0^\circ]$  à  $[60^\circ, -60^\circ, 60^\circ]$  en trois seconde.

Sur la figure (IV.3), on montre l'efficacité et les performances de la méthode proposée. Ainsi que le test de la robustesse et de la capacité d'adaptation du schéma, face aux variations paramétriques, une variation de 100% de tous les paramètres a été effectuée ainsi qu'une variation de charge de 10 kg au milieu du parcours.

La figure IV.4 montre les résultats obtenus pour la commande en poursuite de la trajectoire de Leahy. Ainsi que le même test de perturbation utilisé ci dessus.

#### Paramètres du contrôleur :

Les poids des RNA sont initialisés entre  $-0.1$  et  $0.1$ . Les valeurs des pondérations des erreurs sont :

$\omega_{p1}=3, \omega_{p2}=6, \omega_{p3}=6, \omega_{v1}=2, \omega_{v2}=5, \omega_{v3}=2.$

$K_{pi}=400, k_{vi}=40$  ( $i=1, 2$  et  $3$ ), et  $\alpha_1^i=0.6, 0.6, 0.6$  et  $\alpha_2^i=0.01, 0.0005, 0.01$  ( $i=1, 2$  et  $3$ ).

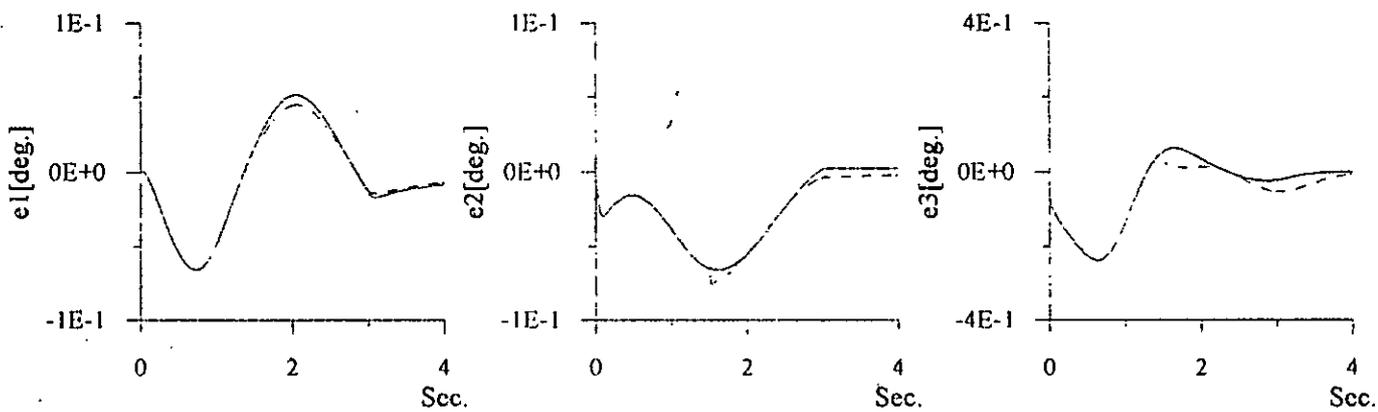
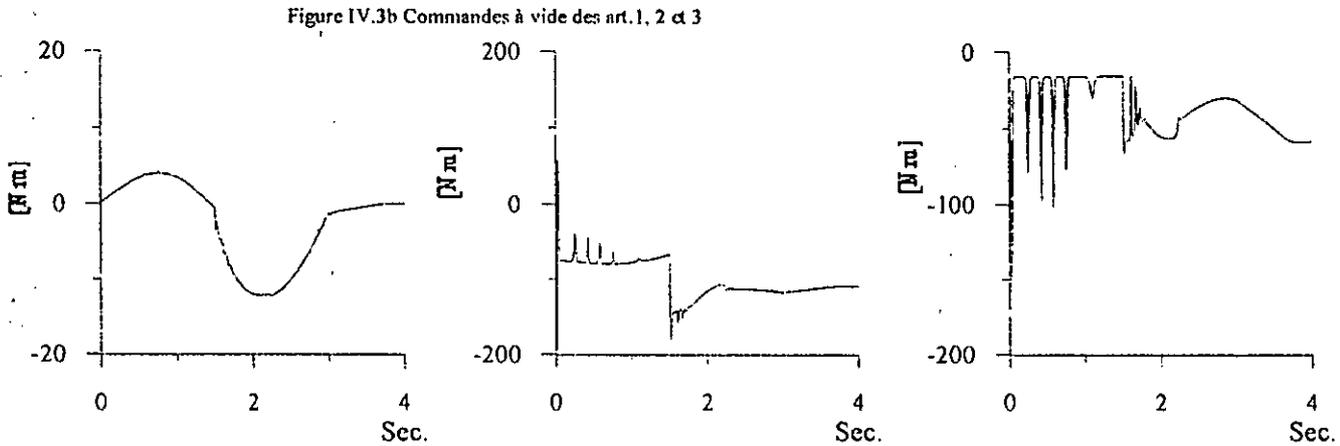
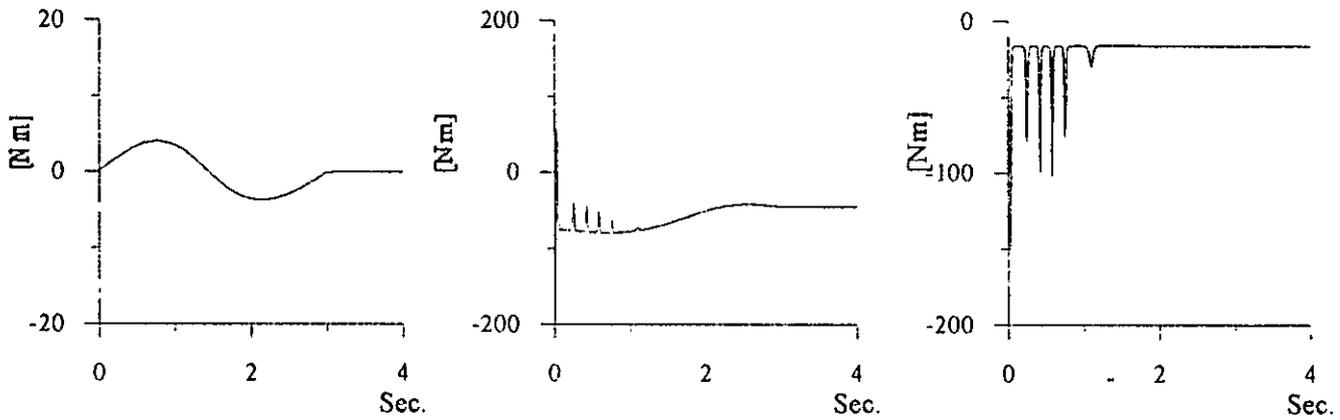
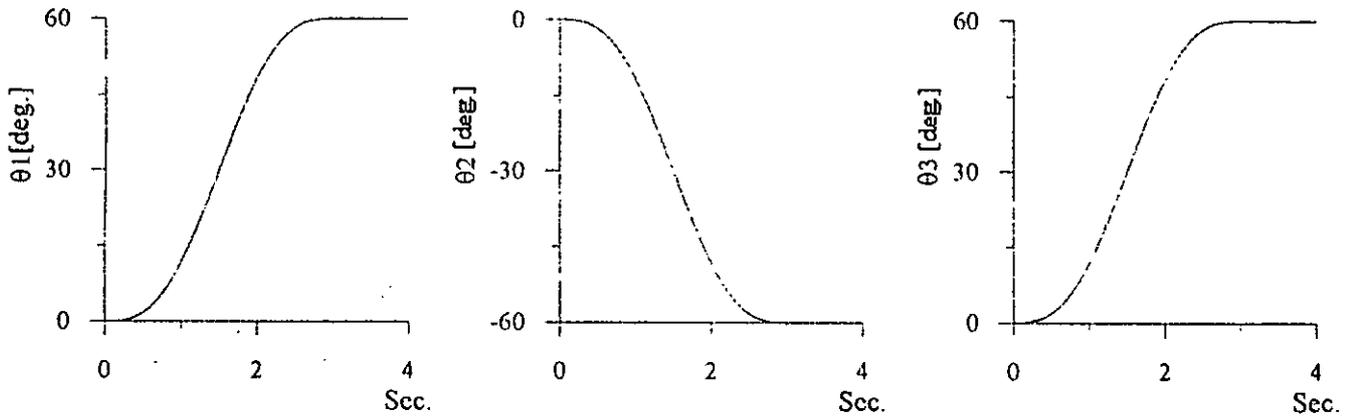
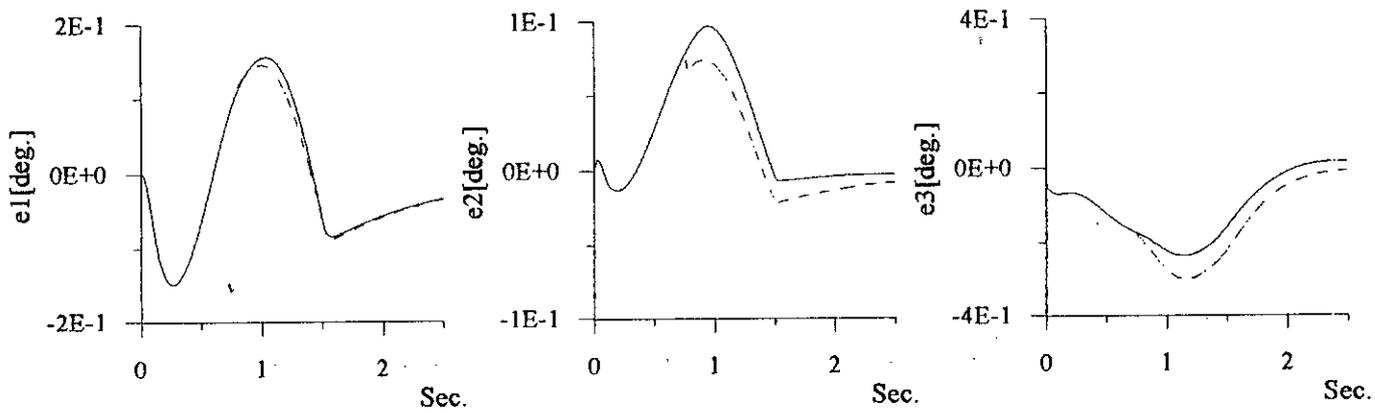
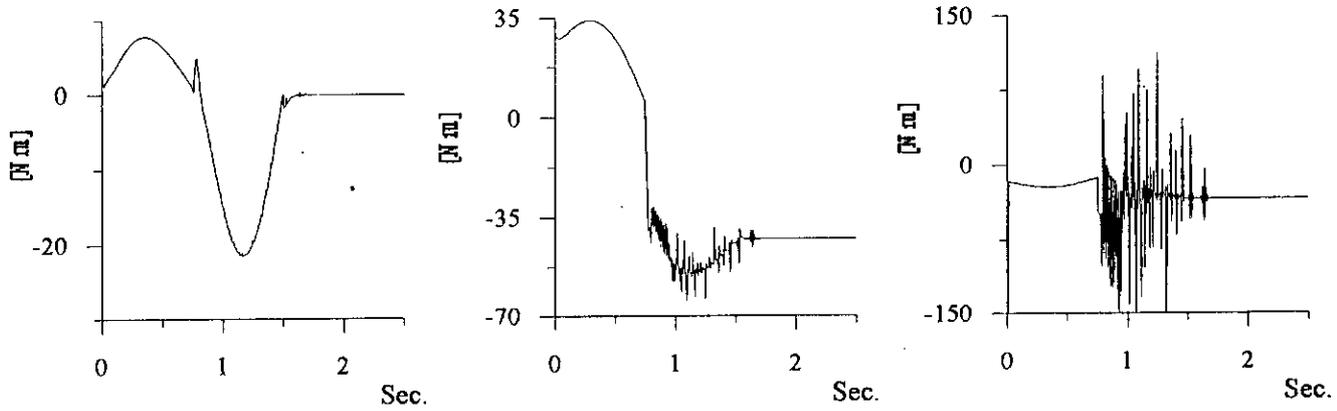
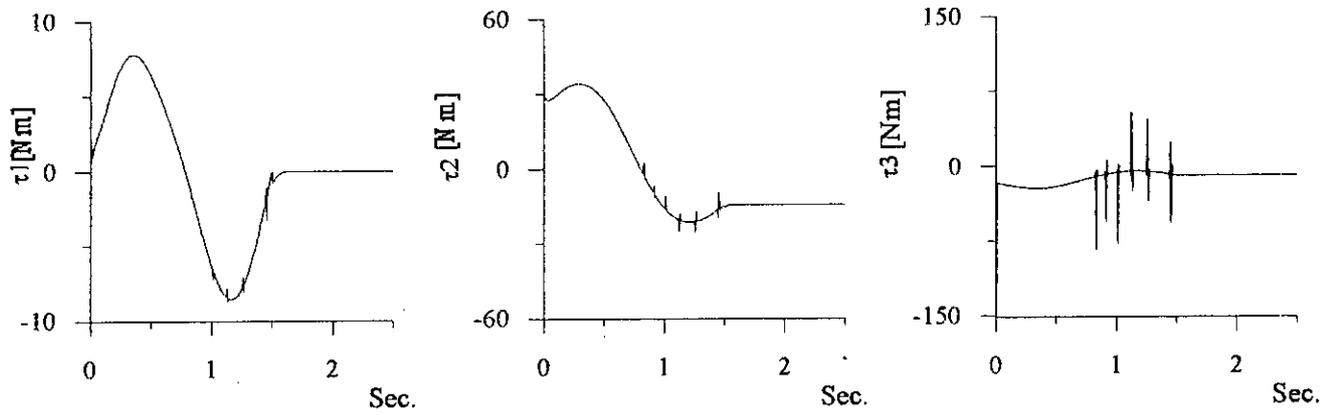
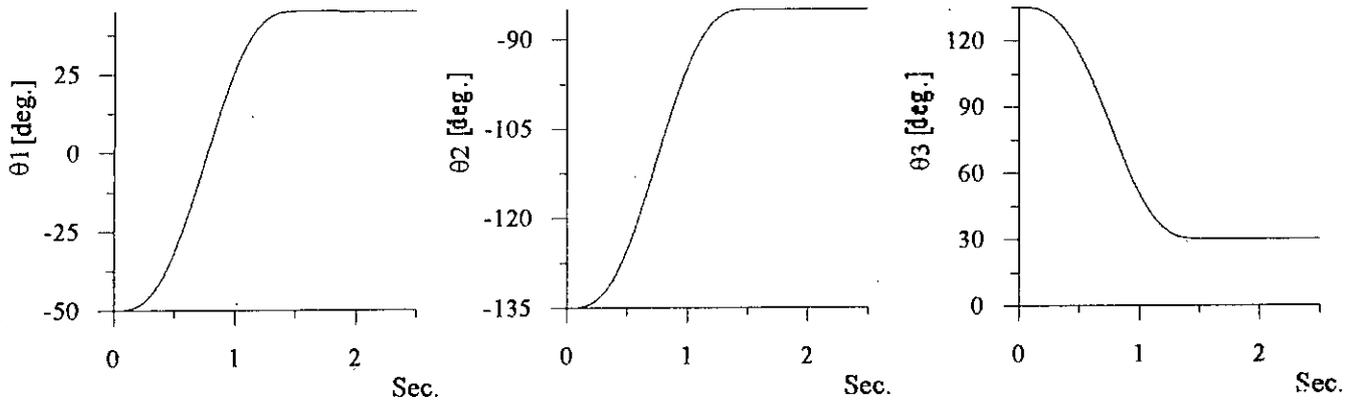


Figure IV.3d Erreurs des art.1, 2 et 3

— à vide.      - - - avec perturbation.



### IV.3 Commande Adaptative Indirecte Décentralisée par RBFG

#### IV.3.1 RBFG à Centres Fixes

Soit le modèle dynamique du robot donné par l'équation (IV.4). Les fonctions  $f_i(\cdot)$  et  $g_i(\cdot)$  ( $i=1, \dots, n$ ) sont liées aux fonctions de base radiales par la relation (I.51). En effet, ces fonctions seront estimées par des fonctions possédant la même structure donnée par l'équation (I.52).

#### A. Etape d'identification :

Le modèle neuronale est donné par l'équation (IV.5).  $\hat{f}_i(\cdot)$  et  $\hat{g}_i(\cdot)$  sont respectivement les estimées de  $f_i(\cdot)$  et  $g_i(\cdot)$  générées respectivement par deux réseaux RBFG à centre fixes  $\text{RNA}_1^i$  et  $\text{RNA}_2^i$ , dont les centres sont fixés a priori et les poids de sortie sont ajustés à chaque instant par un algorithme d'adaptation paramétrique (AAP) utilisant la rétropropagation de l'erreur filtrée.

#### B. Etape de Commande :

Utilisons les estimées des fonctions  $f_i(\cdot)$  et  $g_i(\cdot)$ , la loi de commande de la  $i^{\text{ème}}$  articulation est donnée par la relation (IV.6) et le principe de commande utilisée est illustré par la figure (IV.2).

#### Algorithme d'Adaptation

L'adaptation des poids de sortie des réseaux est effectuée par la rétropropagation de l'erreur filtrée  $r_i(t)$  définie par l'équation (IV.9).

Les poids de la couche de sortie du réseau  $\text{RNA}_l^i$  ( $l=1,2$  et  $i=1, \dots, n$ ) sont adaptés par :

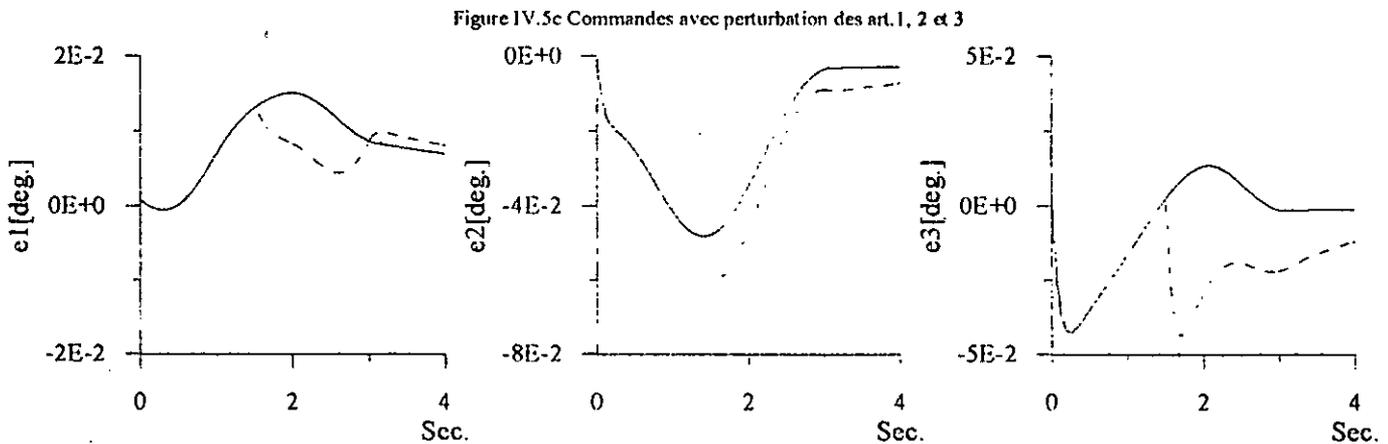
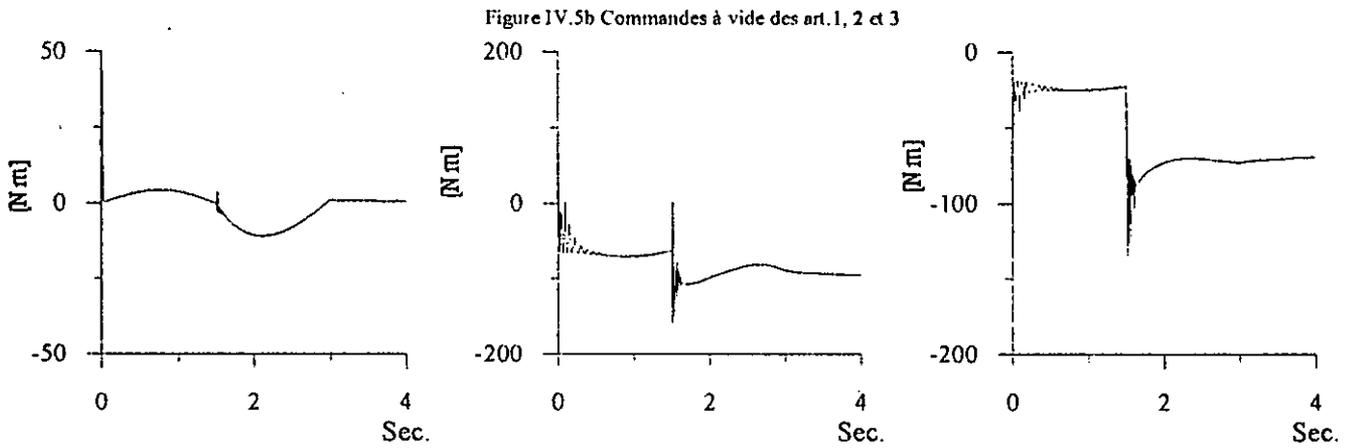
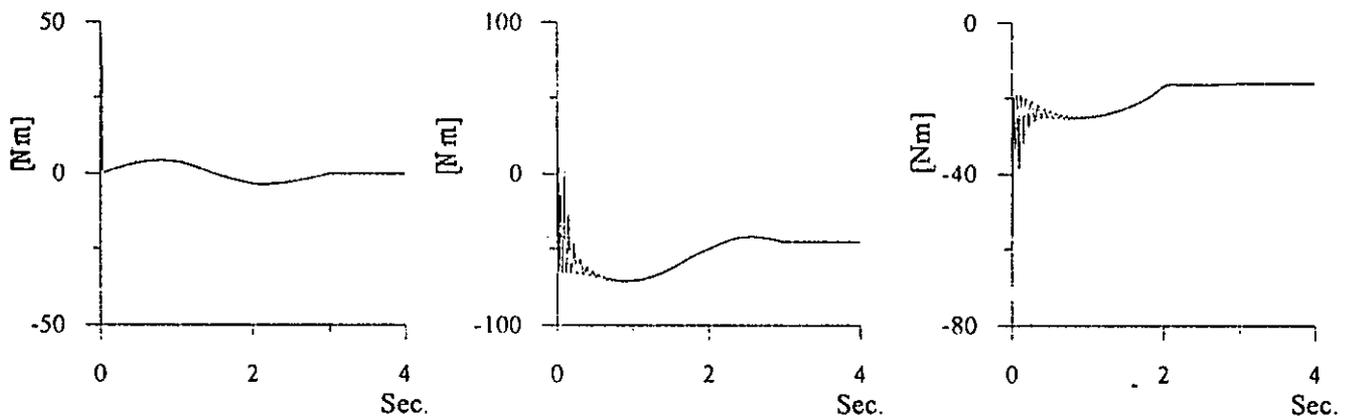
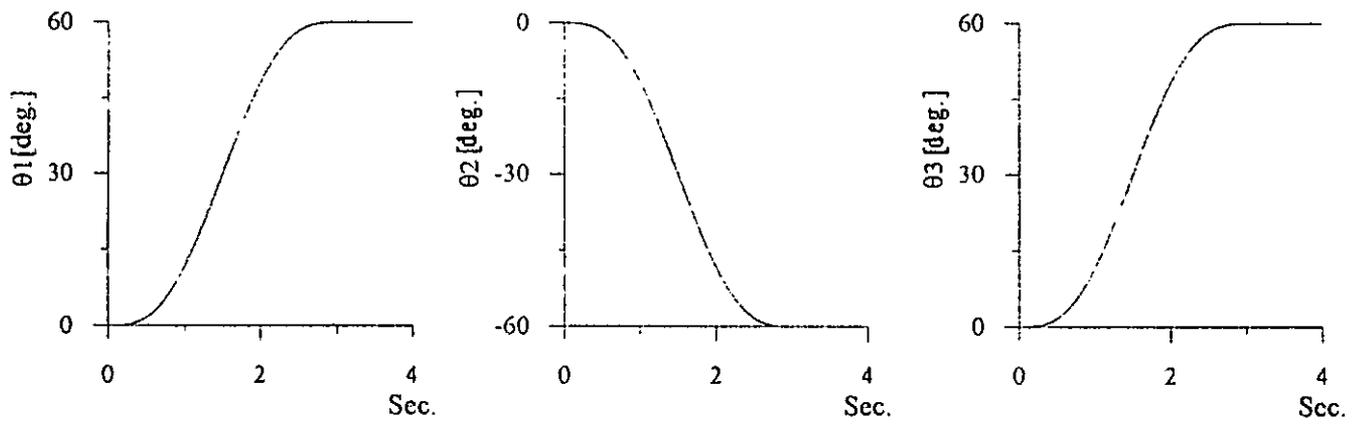
$$\Delta \hat{c}_l^i = \mu_l^i \delta_l^i g_\sigma(x_l^i - \xi_{ll}^i) \quad (\text{IV.14})$$

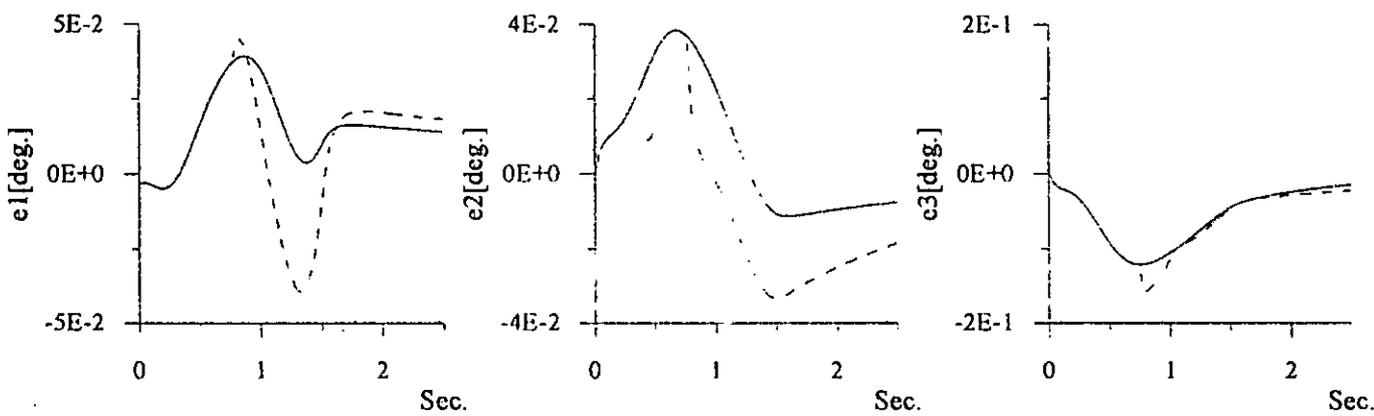
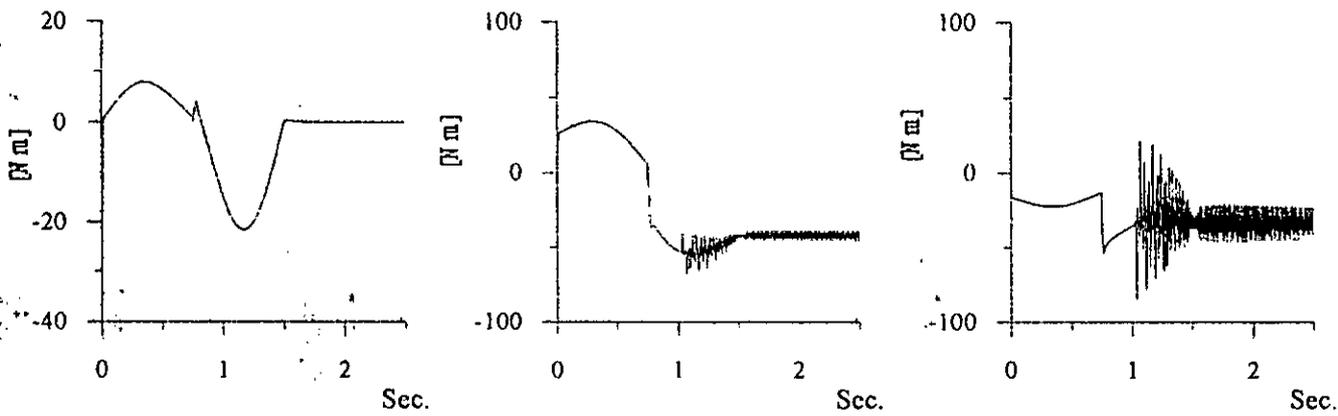
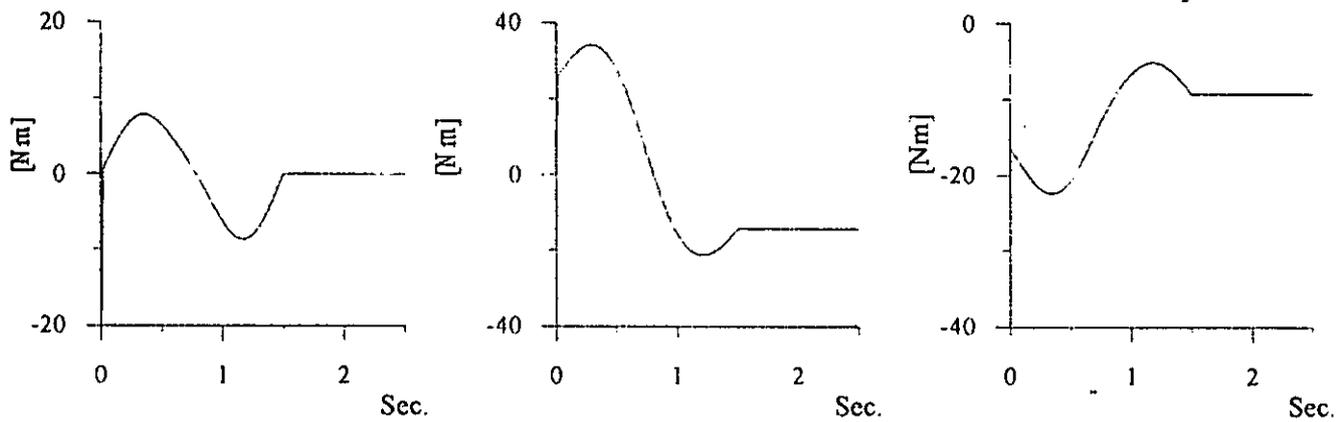
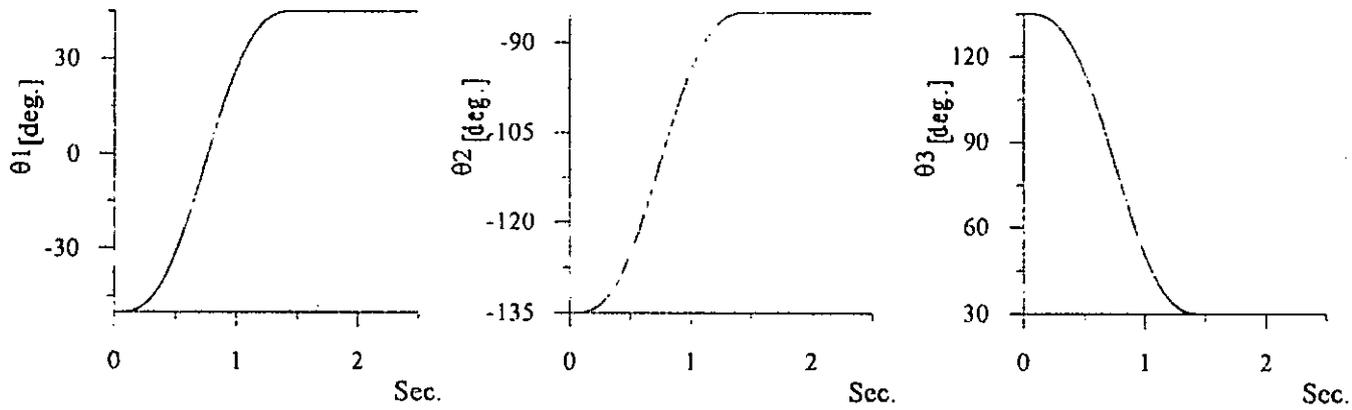
avec  $\delta_l^i$  est définie par (IV.11) et  $\mu_l^i$  est définie par (IV.13).

#### Etude par Simulation

En se basant sur la trajectoire désirée, les ensembles  $A_{di}$  sont choisis en utilisant les bornes min et max des  $\theta_{di}, \dot{\theta}_{di}$  ( $i=1, \dots, n$ ).

Les figures (IV.5) et (IV.6) montrent les performances et la robustesse de cette stratégie de commande vis-à-vis des perturbations pour différentes trajectoires.





### IV.3.2 RBFG à Centres Adaptatifs

Vu la taille des réseaux dans le cas des fonctions de base radiales à centre fixes nous développons la commande adaptative indirecte décentralisée par réseaux à fonction de base radiale généralisés (à centres adaptatifs).

Les étapes d'identification et de commande sont les mêmes que celles de la section (IV.3.1).

#### Algorithme d'Adaptation

L'adaptation des réseaux à base radiale généralisés est effectuée par la rétropropagation de l'erreur filtrée (IV.9) suivant les équations :

$$\Delta \hat{c}_i^i = \mu_i^i \delta_i^i g_\sigma(x_i^i - \xi_{ii}^i) \quad (\text{IV.15})$$

$$\Delta \hat{\xi}_{ii}^i = \mu_i^i \frac{2}{\sigma^2} \delta_i^i (x_i^i - \xi_{ii}^i) g_\sigma(x_i^i - \xi_{ii}^i) \hat{c}_{ii}^i \quad (\text{IV.16})$$

avec  $\delta_i^i$  est définie par (IV.11) et  $\mu_i^i$  est définie par (IV.13).

#### Etude par Simulation

Les résultats (figures IV.7 et IV.8) obtenus montrent les performances et l'efficacité de cette stratégie de commande par les réseaux RBFG à centres adaptatifs en terme de poursuite des trajectoires de références pour le robot manipulateur ainsi que la robustesse vis-à-vis des brusques perturbations.

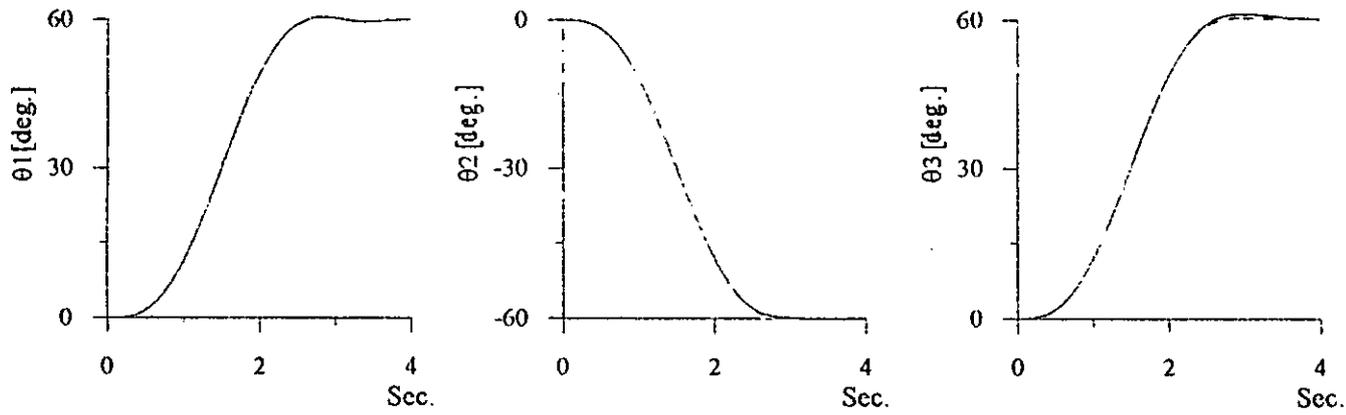


Figure IV.7a Réponses en position des art.1, 2 et 3 ——— à vide. - - - avec perturbation.

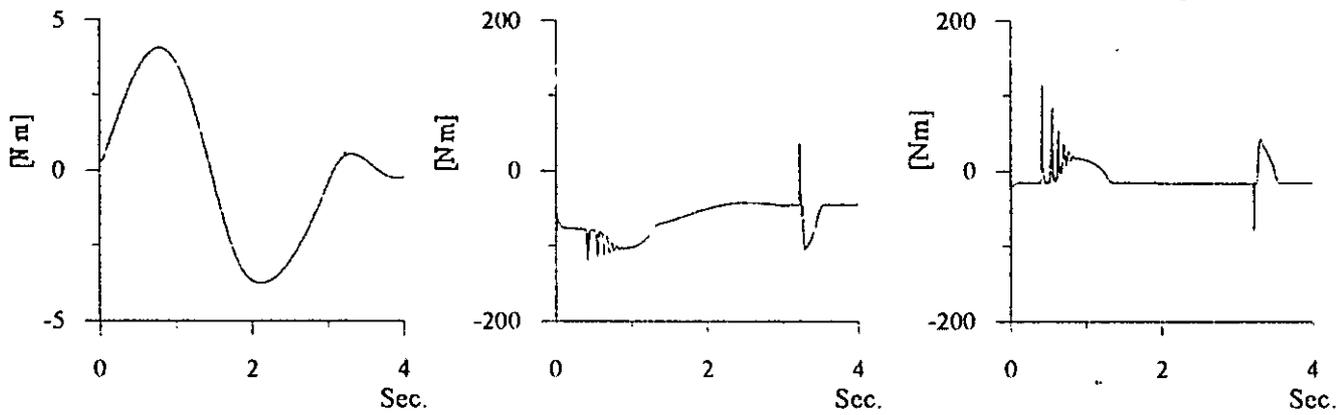


Figure IV.7b Commandes à vide des art.1, 2 et 3

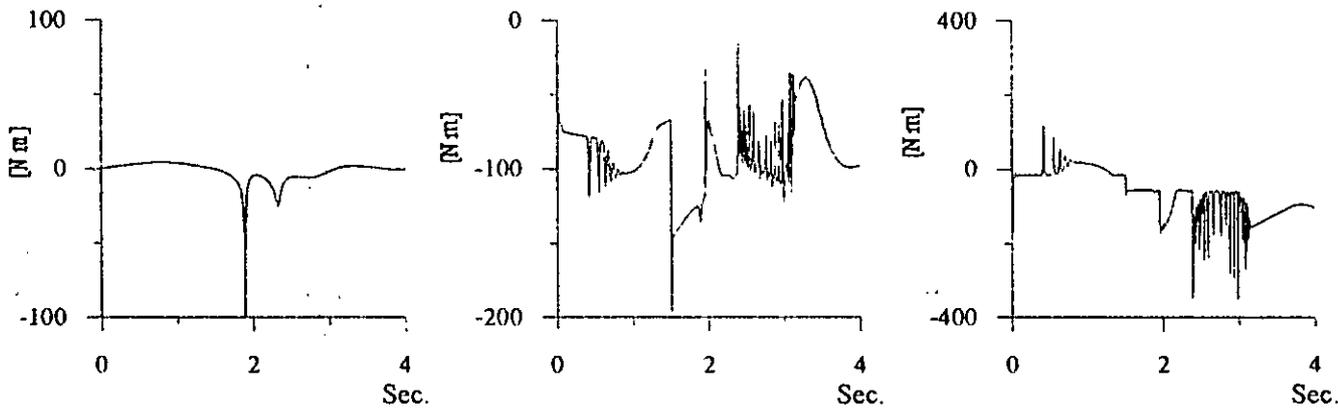


Figure IV.7c Commandes avec perturbation des art.1, 2 et 3

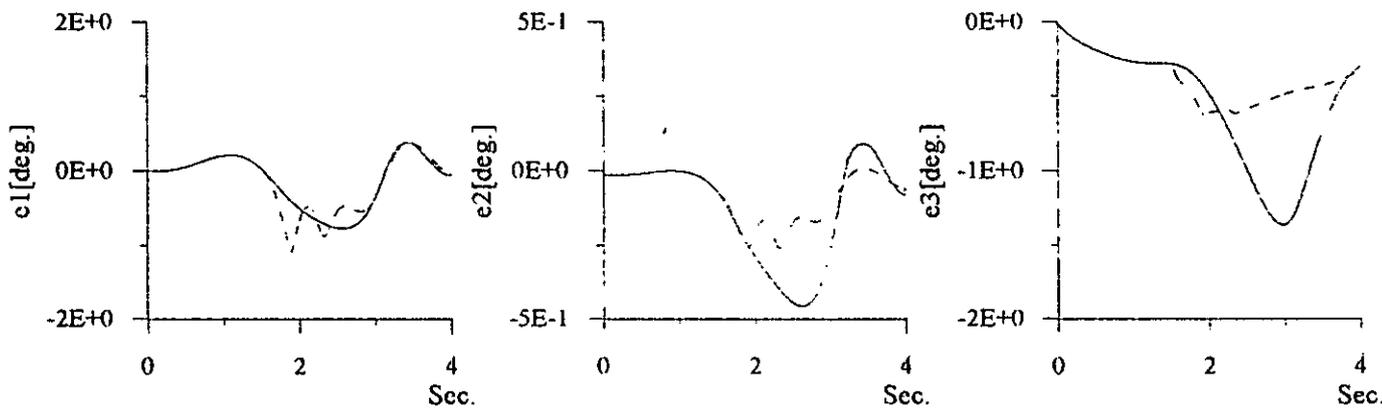


Figure IV.7d Erreurs des art.1, 2 et 3 ——— à vide. - - - avec perturbation.

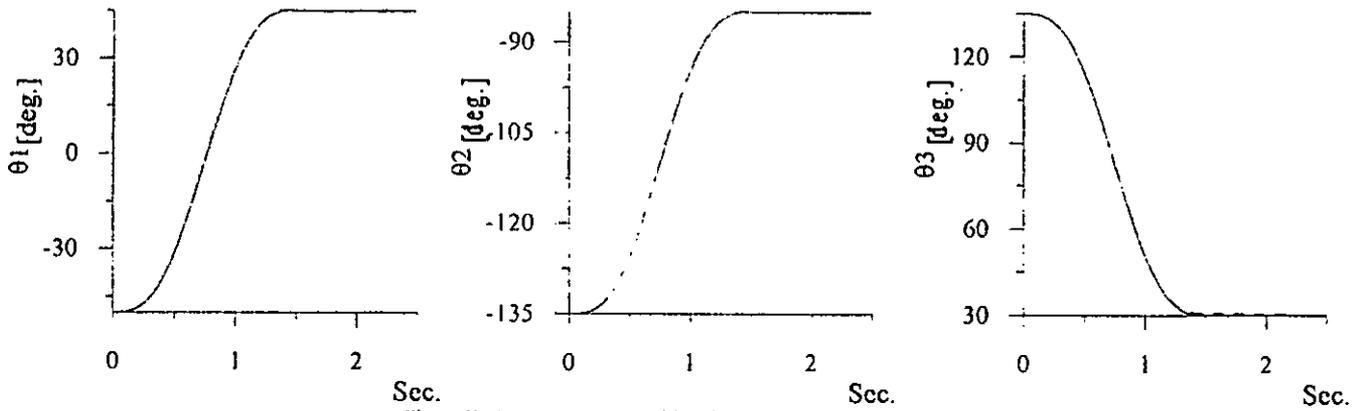


Figure IV.8a Réponses en position des art. 1, 2 et 3

----- à vide, ----- avec perturbation.

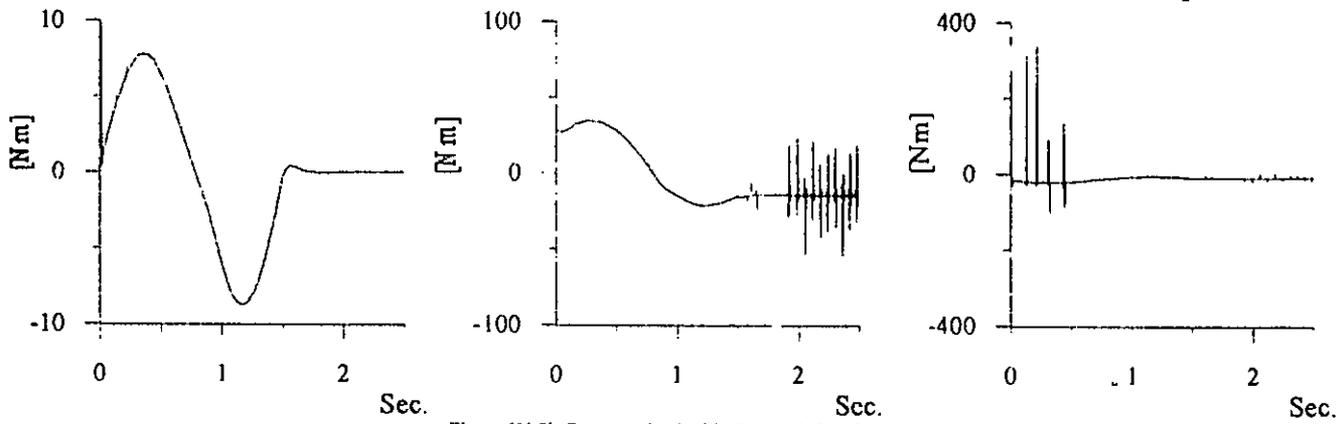


Figure IV.8b Commandes à vide des art. 1, 2 et 3

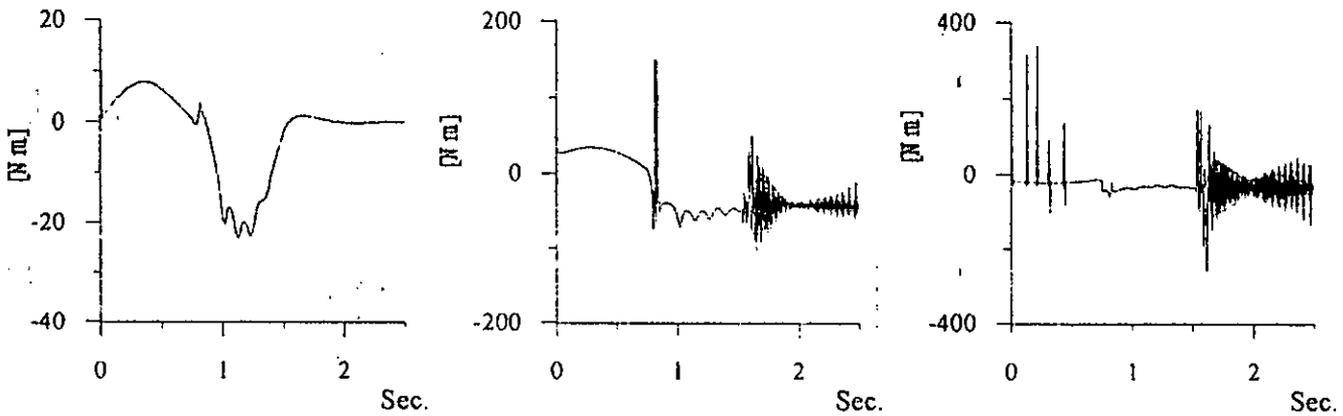


Figure IV.8c Commandes avec perturbation des art. 1, 2 et 3

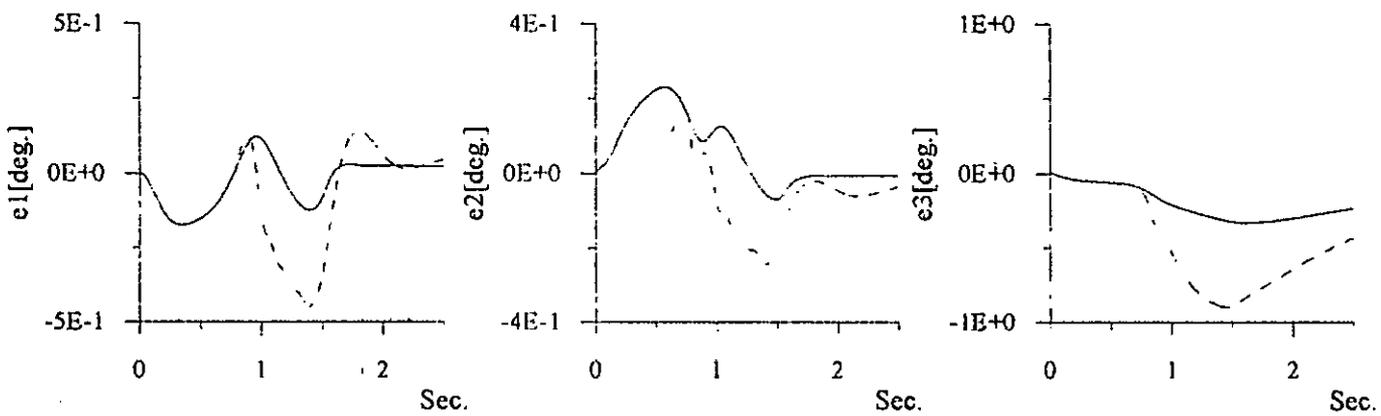


Figure IV.8d Erreurs des art. 1, 2 et 3

----- à vide, ----- avec perturbation.

#### IV.4 Commande Adaptative Indirecte par RNAR

Dans cette section, nous présentons l'utilisation des réseaux récurrents dans la commande adaptative indirecte décentralisée.

Les étapes d'identification et de commande sont les mêmes que celles de la section (IV.3.1).

Les lois d'adaptations des réseaux récurrents sont données par l'équations (I.100) avec :

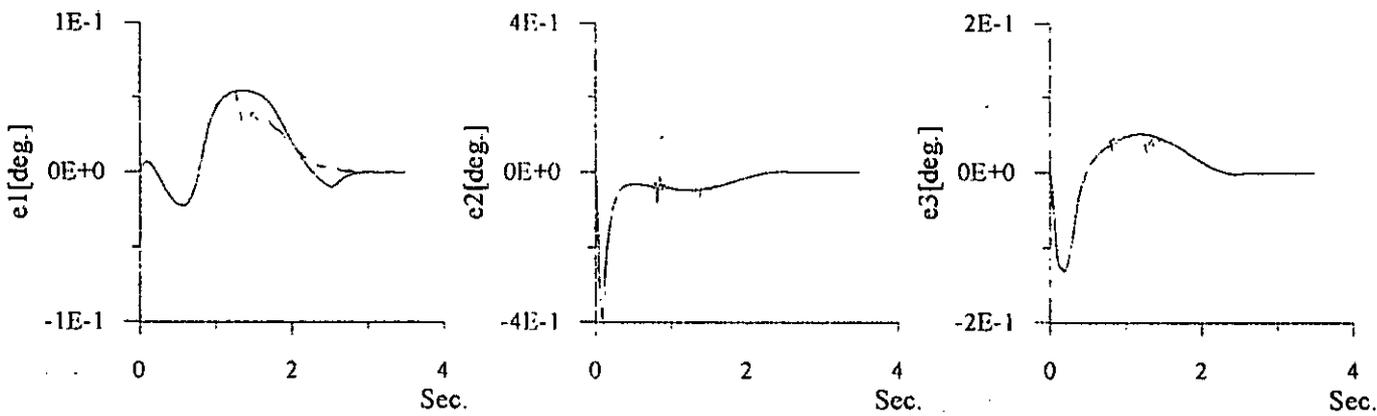
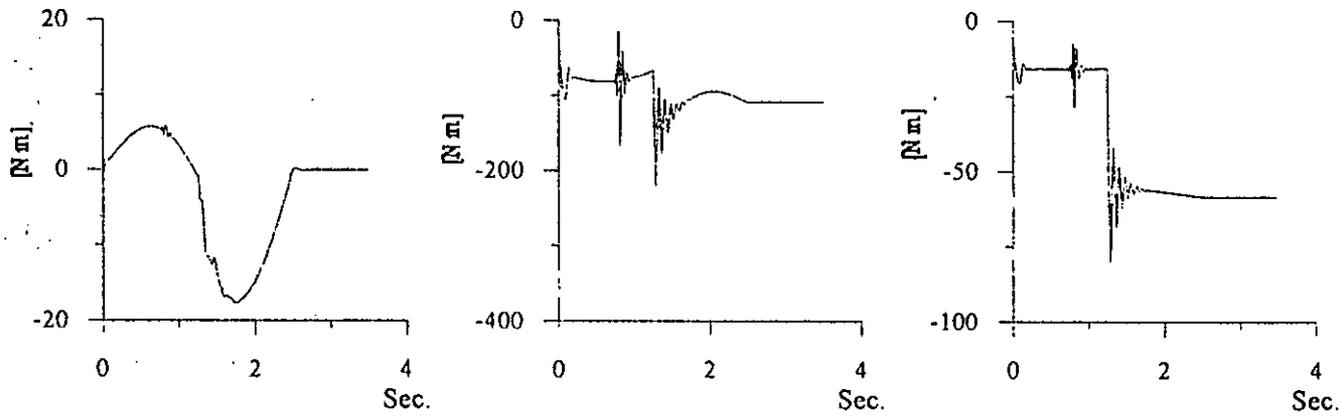
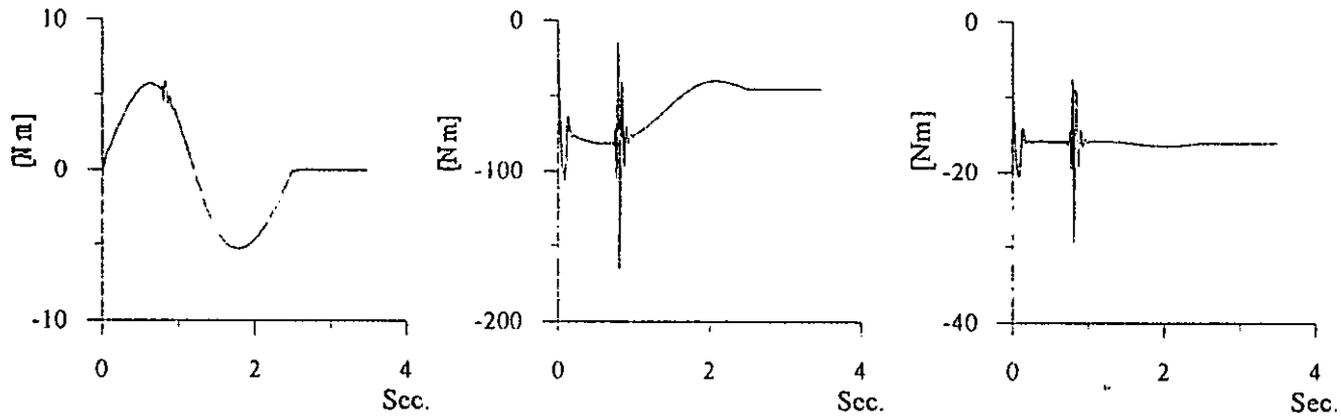
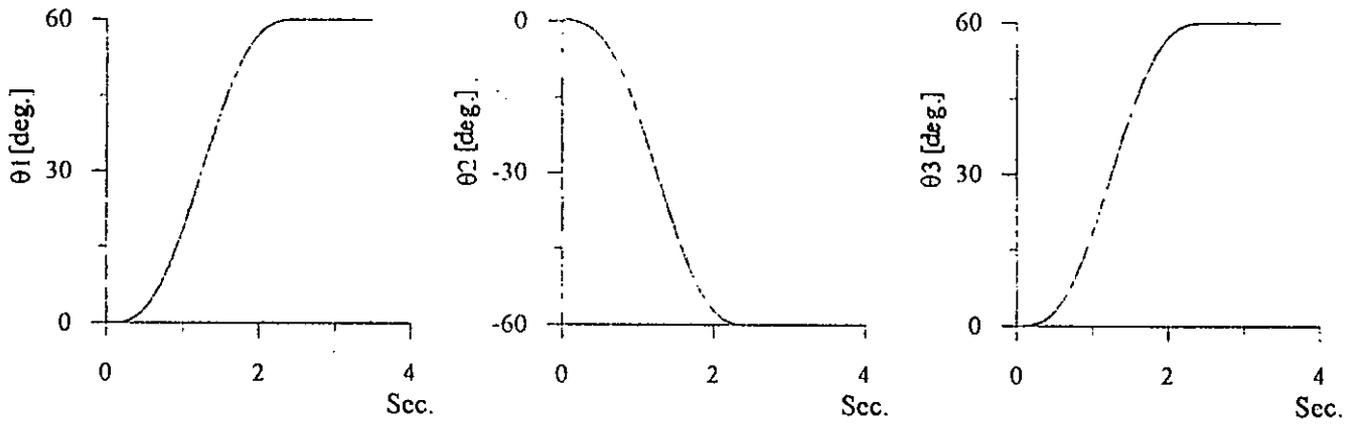
$$V_d^2 - V_k^2 = \delta_i^j \quad (\text{IV.17})$$

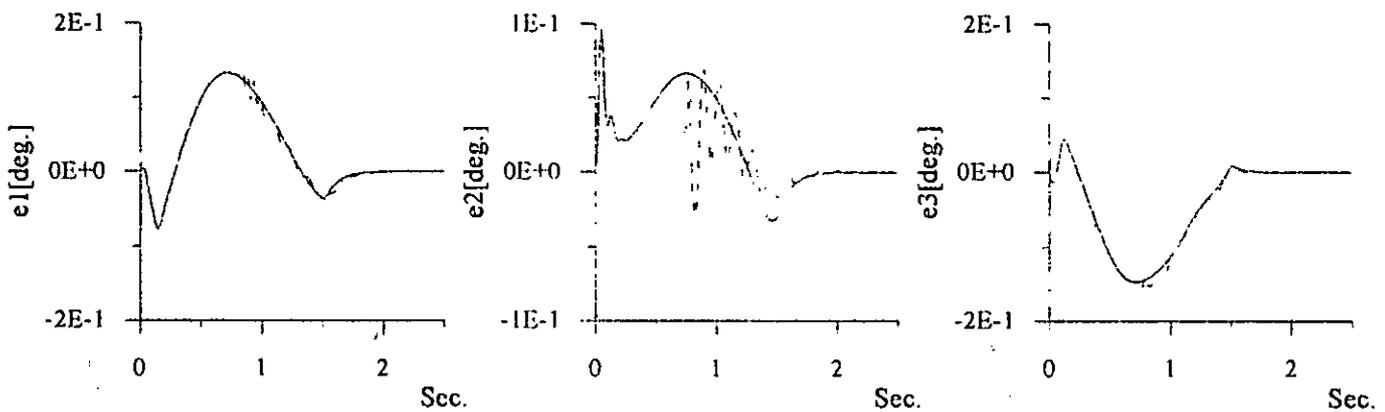
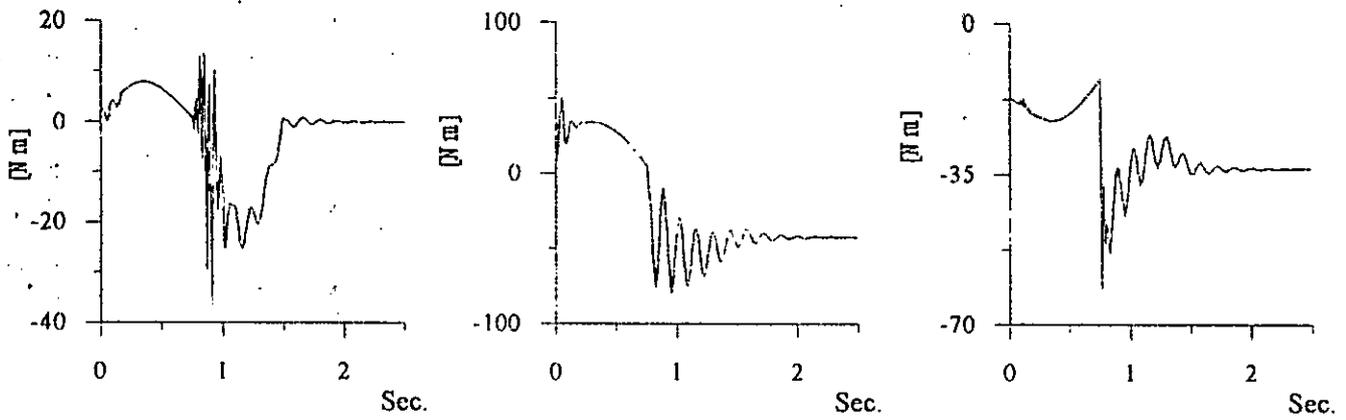
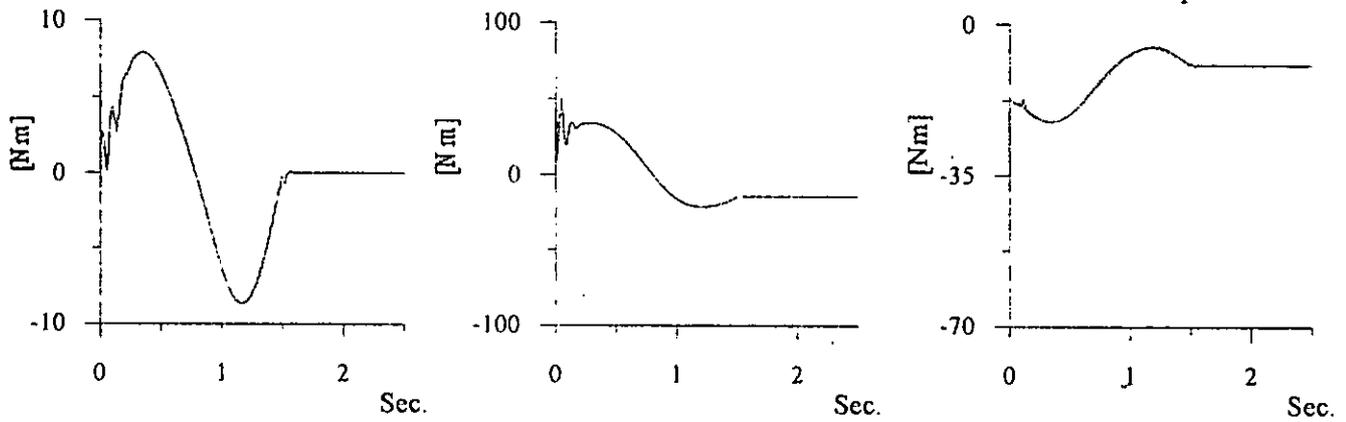
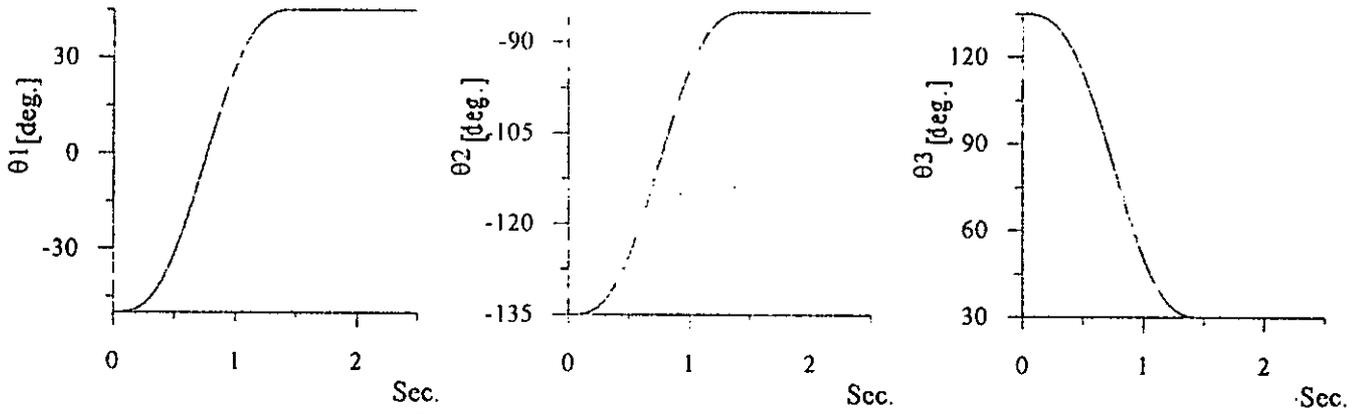
avec  $\delta_i^j$  est définie par l'équation (IV.11)

##### **Etude par Simulation**

Nous avons opté pour des RNAR chacun à trois couches, la couche cachée possède 5 neurones et  $n_y = 2$  (deux retours de la sortie).

Les figures (III.9) et (III.10) montrent les performances du schéma de commande proposé par les réseaux récurrents sans et avec perturbation, pour la commande en poursuite de trajectoires de références des trois premières articulations du robot PUMA 560. Nous constatons que l'utilisation des réseaux récurrents donne des résultats satisfaisants pour la commande en poursuite des robots manipulateurs.





## IV.5 CONCLUSION

Dans le présent chapitre, nous avons présenté une approche pour la synthèse d'une loi de commande adaptative basée sur l'identification du système à commander par un modèle neuronal décentralisé. Les résultats de simulations obtenus en utilisant quatre types différents de réseaux de neurones (réseaux statiques, réseaux RBF à centres fixes, réseaux RBF généralisés et réseaux récurrents), pour deux trajectoires différentes, montrent la robustesse du schéma de commande vis à vis des brusques variations de la charge et à tous les paramètres.

Cette technique de commande présente un double intérêt. Ainsi, la décentralisation de la commande permet de réduire considérablement le temps de calcul de celle-ci. Par ailleurs, nous avons constaté que, l'utilisation des RNA a permis l'augmentation considérable de la robustesse de la commande en éliminant les défauts des méthodes classiques.

A noter que le calcul de la loi de commande contient une division par un terme. Ceci peut provoquer une division par zéro lors de l'adaptation des paramètres de ces termes. Pour remédier à ce problème, les algorithmes de projection pour l'adaptation des paramètres [KHA 96] pourront être utilisés.

---

## *Conclusion Générale*

---

**D**ANS ces travaux, nous avons étudié et développé des stratégies de commande adaptative centralisées et décentralisées utilisant les réseaux de neurones artificiels, appliquées à la commande en poursuite de trajectoires de références des robots manipulateurs. Ainsi, nous nous sommes intéressés à la commande adaptative directe centralisée, à la commande prédictive adaptative directe centralisée, à la commande adaptative directe décentralisée, à la commande prédictive adaptative directe décentralisée, à la commande neuronale adaptative décentralisée et à la commande adaptative indirecte décentralisée.

Dans le premier chapitre, une large recherche bibliographique a été faite dont le but d'explorer les différents types de réseaux de neurones mis en oeuvre efficacement dans la commande des processus. Ainsi, nous avons étudié deux nouvelles approches adaptatives directes centralisées par RNA appliquées en robotique, basées sur l'ajustement en temps réel des paramètres de commande, selon des algorithmes issus de la théorie de la stabilité non-linéaire. A l'issue de cette recherche, nous avons constaté, d'une part, que depuis l'apparition du premier neurone artificiel mis au point par Mc Culloch et Pitts en 1943, les réseaux de neurones restent marqués par l'algorithme d'apprentissage de la rétropropagation (Backpropagation) permettant d'exploiter la structure fort puissante des réseaux multicouches. D'autre part, nous avons constaté que les RNA sont parfois difficiles à construire. En effet, la détermination d'une architecture optimale par des méthodes systématiques précises n'est pas encore bien établie. En automatique, l'utilisation des RNA comme éléments de base pour la modélisation et la commande des processus nous épargne, dans la majorité des cas, de la connaissance détaillée du modèle du système, et on se contente de son comportement entrée/sortie. Malgré les résultats satisfaisants obtenus par les différents types de RNA utilisés (statiques, RBFG à centres fixes, RBFG généralisés et récurrents) nous constatons que la taille du RNA complique la conception de la commande et rend l'apprentissage lent ce qui nécessite d'assez puissants moyens de calcul (exp. : lorsque la dimension d'entrée augmente, la couche cachée des réseaux RBFG à centres fixes risque d'être surchargée en neurones). Une solution immédiate à cet inconvénient est d'utiliser la structure décentralisée. Dans cette structure, chaque sous système (liaison) est commandé séparément, en utilisant que les entrées et les sorties de ce dernier, donc elle nous permet une réduction notable de la taille du réseau et simplifie la conception de la commande.

Le deuxième chapitre a été consacré au développement des commandes décentralisées en se basant sur l'extension, au cas décentralisé, des méthodes de commande vues dans le premier chapitre. Ces méthodes ont été appliquées sur le même bras manipulateur et ont donné des résultats remarquables de point de vue : la taille des réseaux, temps de calcul ainsi que les performances de poursuite.

Dans le troisième chapitre, nous avons développé une technique de commande neuronale adaptative décentralisée basée sur la méthode du couple calculé. Une question peut être posée la déçu : Pour quoi une commande décentralisée ? . Notre réponse est comme suit : parce que dans cette technique de commande centralisée on identifie les éléments de la matrice d'inertie et les éléments du vecteur de coriolis et des forces de centrifuge par deux RNA MIMO. Les entrées sont la position et la vitesse de toutes les articulations et les sorties sont respectivement les éléments de la matrice d'inertie, les

éléments du vecteur de coriolis et les forces de centrifuge. Puis à partir de ces derniers et en utilisant la méthode du couple calculé, la commande est calculée. Donc, l'apprentissage dans cette technique est difficile et même impossible si le nombre d'articulation est très élevé. La solution appropriée à ce problème est donnée par les caractéristiques de la structure décentralisée. Dans laquelle, on identifie que deux fonctions inconnues pour chaque articulation. L'utilisation des réseaux MISO simplifie la conception de la commande et la rendre universelle.

Le dernier chapitre est consacré au développement de la commande adaptative indirecte décentralisée par RNA. Dans cette technique de commande, nous utilisons deux RNA pour identifier chaque articulation. Ensuite, à partir de ce modèle neuronal, on calcule la commande correspondante. Donc, ce régulateur neuronal peut être universel, car il est conçu en absence de la moindre information sur les paramètres et la complexité de la dynamique du bras de robot manipulateur.

La simulation effectuée sur les trois premières articulations d'un robot de type PUMA 560 montre l'efficacité des RNA utilisés dans la modélisation et la commande des robots manipulateurs. En effet, nous constatons que les différentes commandes conduisent à des faibles erreurs de poursuite. De plus, les techniques de commande adoptées sont robustes par rapport aux variations paramétriques du système. Par ailleurs, l'utilisation des RNA comme élément de base pour l'approximation du modèle dynamique fortement non linéaire du bras manipulateur a conduit à une amélioration notable des performances de la commande adaptative classique.

La puissance particulière de ces techniques de commandes adaptatives proposées, dans cette thèse, par rapport aux approches classiques est la non nécessité du modèle de connaissance du système commandé. Donc, elles peuvent être vues comme des contrôleurs universels des robots manipulateurs rigides.

Jusqu'à présent, les réseaux de neurones avec leurs énormes capacités d'estimation et de généralisation ont été utilisés efficacement dans plusieurs approches de commande. Cependant, aucune preuve de stabilité n'a été avancée. Par contre, les algorithmes d'adaptation développés dans le premier et le deuxième chapitre assurent une stabilité du système global dont la preuve a été établie selon la théorie de la stabilité de Lyapunov.

Toutefois, il faut signaler d'une part, l'importance de la structure décentralisée qui nous a permis la réduction du temps de calcul. D'autre part, l'utilisation des RNA a permis l'amélioration des performances de poursuite et l'augmentation de la robustesse vis-à-vis des perturbations.

A partir de l'étude de la contribution des différentes structures de réseaux de neurones dans la commande des robots manipulateurs, on peut constater que :

- Les RNA récurrents sont des réseaux bouclés, capables de traiter des ensembles temporels, acceptent des séquences d'entrée et les produisent comme sortie. Ces RNA peuvent être entraînés par le paradigme de la rétropropagation. Cet entraînement de ces réseaux consomme du temps. Dans le RNA à propagation directe, chaque événement peut être séparé des autres et les sorties dépendent uniquement des entrées courantes et les poids sans prendre en considération ce que les entrées précédentes ont effectué.
- Les réseaux RBFG constituent des modèles de réseaux très efficaces notamment pour l'approximation des fonctions. Néanmoins, ces réseaux présentent quelques inconvénients. En effet, lorsque la dimension de l'entrée augmente, le nombre de neurones dans la couche cachée risque d'être très élevé compliquant ainsi les calculs et ralentissant l'apprentissage. En outre, afin de

rendre ces réseaux praticables, il est possible d'adapter les centres des neurones obtenant ainsi les réseaux RBFG généralisés. Ces réseaux ont pris l'avance par rapport aux réseaux multicouches et ils sont devenus les plus utilisés dans les applications récentes.

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

- Application de ces techniques de commande à d'autres types de systèmes non linéaires, tels que les robots à jonctions flexibles, les robots flexibles, les machines électriques et les systèmes chimiques,... etc.
- Commande adaptative décentralisée stable par les systèmes floue.
- L'utilisation des réseaux DRNN (Diagonal Recurent Neural Networks) et LBF (Localised Based Function),... etc.
- La synthèse de lois de commande adaptative généralisées.
- L'utilisation des nouvelles techniques dans la synthèse des lois de commande tel que 'Backstepping'.

### A.1 Présentation du robot PUMA 560

Dans cette étude, nous avons utilisé un robot manipulateur de type PUMA 560. On a considéré uniquement les trois premières articulations rotationnelles  $\theta_1$ ,  $\theta_2$  et  $\theta_3$  présenté sur la figure (A.1).

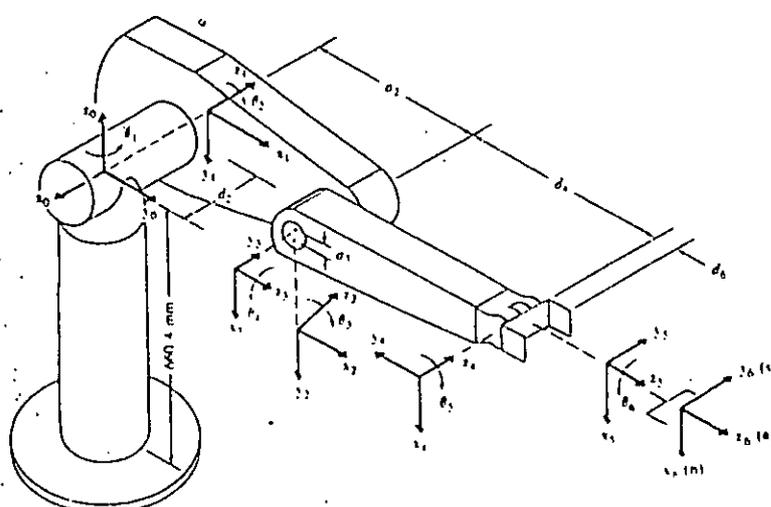


Figure A.1 Présentation du robot PUMA 560.

### A.2 Modèle dynamique du robot manipulateur PUMA 560

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

$$M(\theta)\ddot{\theta} + V_m(\theta, \dot{\theta})\dot{\theta} + G(\theta) + \tau_d = \tau \quad (\text{A.1})$$

avec

$$M(\theta) = \begin{bmatrix} I_1 + I_2 c_{23}^2 + I_3 c_2^2 + I_4 c_2 c_{23} & I_5 s_{23} + I_6 s_2 & I_5 s_{23} \\ I_5 s_{23} + I_6 s_2 & I_7 + I_4 c_3 & I_8 + 0.5I_4 c_3 \\ I_5 s_{23} & I_8 + 0.5I_4 c_3 & I_9 \end{bmatrix}$$

$$V_m(\theta, \dot{\theta}) = \begin{bmatrix} -(2(I_3 s_2 c_2 + I_2 s_{23} c_{23}) + I_4 (c_2 s_{23} + s_2 c_{23})) \dot{\theta}_1 \dot{\theta}_2 - (2I_2 s_{23} c_{23} + I_4 c_2 s_{23}) \dot{\theta}_1 \dot{\theta}_3 \\ + (I_6 c_2 + I_3 c_{23}) \dot{\theta}_2^2 + 2I_3 c_{23} \dot{\theta}_2 \dot{\theta}_3 + I_5 c_{23} \dot{\theta}_3^2 \\ (I_3 c_2 s_2 + I_2 c_{23} s_{23} + 0.5 I_4 (s_2 c_{23} + c_2 s_{23})) \dot{\theta}_1^2 - I_4 s_3 \dot{\theta}_2 \dot{\theta}_3 - 0.5 I_4 s_3 \dot{\theta}_3^2 \\ (I_2 s_{23} c_{23} + 0.5 I_4 c_2 s_{23}) \dot{\theta}_1^2 + 0.5 I_4 s_3 \dot{\theta}_2^2 \end{bmatrix}$$

$$G(\theta) = \begin{bmatrix} 0 \\ -(m_3 l_2 + 0.5 m_2 l_2) g c_2 - 0.5 m_3 l_3 g c_{23} \\ -0.5 m_3 l_3 g c_{23} \end{bmatrix}$$

Avec les notations suivantes:

$$\begin{aligned} I_1 &= I_{YY1} + I_{XX2} + m_2 d_2 (d_2 + e) + m_3 d_2^2 + I_{XX3} + I_{xx1} + m_1 d_2^2 + I_{M1} \\ I_2 &= I_{YY3} - I_{XX3} + I_{yy1} - I_{xx1} + m_1 l_3^2 \\ I_3 &= I_{YY2} - I_{XX2} + m_3 l_2^2 + m_1 l_2^2 \\ I_4 &= m_3 l_2 l_3 + 2 m_1 l_2 l_3 \\ I_5 &= 0.5 m_3 l_3 d_2 + m_1 l_3 d_2 \\ I_6 &= 0.5 m_2 l_2 (d_2 + e) + m_3 d_2 l_2 + m_1 d_2 l_2 \\ I_7 &= I_{ZZ2} + I_{ZZ3} + m_3 l_2^2 + I_{zz1} + m_1 (l_2^2 + l_3^2) + I_{M2} \\ I_8 &= I_{ZZ3} + I_{zz1} + m_1 l_3^2 \\ I_9 &= I_8 + I_{M3} \end{aligned}$$

La matrice  $\Delta M$  représente les effets de l'effecteur car ce dernier possède une masse non négligeable.

$$\begin{aligned} \Delta M_{11} &= I_{xx1} + m_1 d_2^2 + I_{m1} + (I_{YY1} - I_{XX1} + m_1 l_3^2) c_{23}^2 + m_1 l_2^2 c_2^2 + 2 m_1 l_2 l_3 c_2 c_{23} \\ \Delta M_{12} &= m_1 l_3 d_2 s_{23} + m_1 d_2 l_2 s_2 \\ \Delta M_{13} &= m_1 l_3 d_2 s_{23} \\ \Delta M_{22} &= I_{zz1} + m_1 (l_2^2 + l_3^2) + I_{M2} + 2 m_1 l_2 l_3 c_3 \\ \Delta M_{23} &= I_{zz1} + m_1 l_3^2 + m_1 l_2 l_3 c_3 \\ \Delta M_{33} &= I_{zz1} + m_1 l_3^2 + I_{m3} \end{aligned}$$

Avec  $I_{Mi}$   $i=1,3$  moments d'inertie des différents moteurs.

$I_{xx1}$ ,  $I_{yy1}$ ,  $I_{zz1}$  moments d'inertie total par rapport aux principaux axes de l'effecteur.

### Effet de la masse de la charge

La masse de la charge a une importance capitale lors des simulations sur ordinateur, et pour modéliser son effet sur la dynamique du système, l'approche de la matrice jacobienne est utilisée.

L'expression de la matrice jacobienne est la suivante:

$$J(\theta) = \begin{bmatrix} -s_1(l_2c_2 + l_3c_{23}) - d_2c_1 & -c_1(l_2s_2 + l_3s_{23}) & -c_1(l_3s_{23}) \\ c_1(l_2c_2 + l_3c_{23}) - d_2s_1 & -s_1(l_2s_2 + l_3s_{23}) & -s_1(l_3s_{23}) \\ 0 & -(l_2c_2 + l_3c_{23}) & -(l_3c_{23}) \end{bmatrix}$$

Il s'ensuit que le couple additif du à l'effet de la masse sera donné par :

$$\tau_{masse} = m \cdot J^T (J \cdot \ddot{q} + \dot{J}(q, \dot{q}) \cdot \dot{q} + g)$$

avec:  $m$  : masse de la charge.  $J$  : la matrice jacobienne dérivée par rapport au temps,  $g = [0 \ 0 \ 9.81]^T$ .

Les paramètres réels de ce robot sont :

- $m_2 = 17.40\text{kg}$        $m_3 = 5.04\text{kg}$        $m_4 = 0.82\text{kg}$
- $m_5 = 0.35\text{kg}$        $m_6 = 0.09\text{kg}$        $m_1 = m_4 + m_5 + m_6 = 1.26\text{kg}$

Paramètres géométriques :

- $d_2 = 149.09\text{mm}$        $l_2 = 431.8\text{mm}$        $l_3 = 433.07\text{mm}$

Paramètres d'inertie :

N° de la liaison	$I_{xxi} [\text{kg m}^2]$	$I_{yyi} [\text{kg m}^2]$	$I_{zzi} [\text{kg m}^2]$	$I_{Mi} [\text{kg m}^2]$
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	$1.64 \cdot 10^{-3}$	$2.25 \cdot 10^{-3}$	$2.35 \cdot 10^{-3}$	-

Tableau A.1 paramètres d'inertie du robot PUMA 560.

---

## Références Bibliographiques

---

- [ARM 86] B. Armstrong and al, "The explicit Dynamic Model and Inertial Parameters of the PUMA 560 arm," *Proceedings IEEE*, pp. 510-519, 1986.
- [AST 89] K. J. Aström, B. Wittenmark *Adaptive Control*, Addison-Wesley, New York, 1989.
- [BEL 97] A. Belmehdi et al, "Commande Linéarisante par Feedback Utilisant les Réseaux de Neurones Multicouches. Application à un Bras de Robot à Trois Degrés de Liberté," *Proceeding of the IEEE*, Vol. 2, pp 88-92. Dec. 1997.
- [BEN 88] A. Benhammou, Contribution à l'Etude de la Commande Adaptative Décentralisée des Systèmes Interconnectés, Thèse de Doctorat d'état, université Paul Sabatier de Toulouse, 1988.
- [BEN 93] H. Bennassar, Les Réseaux de Neurones Artificiels dans les Commandes de Systèmes, Wichita State University, Kansa, USA. Octobre 1993.
- [BOI 88] J. D. Boissonnat et al, *Techniques de la Robotique (Tome 1)*, Editions Hermès, Paris, 1988.
- [BOU 96] D. Boukhetala, O. Bouhali et F. Boudjema, "Commande Adaptative à Structure Décentralisée par Réseaux de Neurones d'un Bras de Robot," *COMAEI'96*, pp. 165-170.
- [BOU 97a] O. Bouhali, D. Boukhetala, F. Boudjema et M. S. Boucherit, "Commande Adaptative directe par Réseaux de Neurones Gaussiens Appliquée à un Pendule," *Proceeding of the IEEE*, Vol. 1, pp 76-80. Dec. 1997.
- [BOU 97b] D. Boukhetala, O. Bouhali, F. Boudjema et N. K. M'Sirdi, "Commande Adaptative Décentralisée des Robots Manipulateurs en Utilisant les Réseaux de Neurones Artificiels : Etude Comparative," Soumis pour AJOT.
- [BOU 98a] O. Bouhali, D. Boukhetala, F. Boudjema et N. K. M'Sirdi, "Commande Neuronale Adaptative Décentralisée du Robot PUMA560," *CNP'98*, 30-31 mai 1998 Tizi-Ouzou, Algérie, pp 200-204.
- [BOU 98b] O. Bouhali, D. Boukhetala, F. Boudjema et N. K. M'Sirdi, "Commande Adaptative Indirecte Décentralisée par RNA du Rbot Manipulateur PUMA 560," *ICEL'98*, accepté.
- [BOU 98c] O. Bouhali, D. Boukhetala, F. Boudjema et N. K. M'Sirdi, "Une Approche de Commande Adaptative Décentralisée par Réseaux de Neurones Récurrents des Robots Manipulateurs," Soumis au *COMAEI'98*.
- [BOU 99] O. Bouhali, D. Boukhetala, F. Boudjema et N. K. M'Sirdi, "Commande Neuronale Adaptative Directe des Robots Manipulateurs: Etude Comparative," Soumis au *SENIA'99*.
- [CHA 95a] F. C. Chang and H. K. Khalil, "Adaptive Control of a Class of Nonlinear Discrete-Time Systems Using Neural Networks," *IEEE Trans. on Automatic Control*, Vol. 40, N° 5, MAY 1995.
- [CHA 95b] N. Chatry and al, "Double Backpropagation for the Gradient Calculation in Dynamic Feedback Networks," *IFAC-IFIP-IMACS Conference, Control of Industrial systems*, pp 535-538, 20-22 May 1997, Belfort France.
- [CHE 90] F. C. Chen, "Backpropagation Neural Networks fo Nonlinear Self-Tuning Adaptive Control," *IEEE Control Syst. Mag.*, vol. 10, pp. 44-48, 1990.
- [FRE 92] J. A. Freeman, D. M. Skapura, *Neural Networks* Addison-Wisley Publishing Company, Inc, 1992.
- [FU 87] Fu, S. et al, (1987) *Robotics: Control, Sensing, Vision and Intelligence*, Mcc Graw Hill.

- [GOO 84] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*, Englewood Cliffs, NJ :Prentice Hall, 1984.
- [HAM 97] Hamzi, B., Labiod, S., Boudjema and F Boukhetala, D. "Supervised Control of the Inverted Pendulum with Dynamical Neural Networks," IFAC-IFIP-IMACS Conference, Control of Industrial Systems, vol. 3, pp 125-129, 20-22 May 1997, Belfort, France.
- [HUN 92] K.J. Hunt and al, "Neural Networks for Control Systems-A Survey," *Automatica*, vol. 28, NO. 6, pp. 1083-1112, 1992.
- [IOU 86] P. A. Iouannou, "Decentralized Adaptive Control of Interconnected Systems," *IEEE Trans. Aut. Contr.*, Vol. AC-31, No. 04, Apr. 1986.
- [JAI 97a] D.S. Jain *et al* ,"Decentralized Adaptive Control of a Class of Large-Scale Nonlinear Systems," *IEEE Trans. Aut. Contr.*, Vol. 42 pp 729-735. No. 05, May 1997.
- [JAI 97b] D.S. Jain *et al* ," Decentralized Adaptive Output Feedback Design for Large-Scale Interconnected Nonlinear Systems," *IEEE Trans. Aut. Contr.*, Vol. 42 pp 136-154. No. 02, 1997.
- [KHA 96] H. K. Khalil,"Adaptive Output Feedback Control of Nonlinear Systems Represented by input-Output Models", *IEEE Trans. on Automatic Control*, Vol. 41, N° 2, pp 177-188, Feb. 1996.
- [KHE 94] S. Khemaissia and A. S. Morris, "Review of Artificial Neural Networks and Choice of Radial Basis Function Networks for System Identification," *TECHNOLOGIES ADVANCES*, N° 6, September 1994.
- [KOI 83] Koivo, A. J. and Guo, T. H. (1983) "Adaptive linear Controller for Robotic Manipulators", *IEEE Transactions on Automatic control*, Vol. AG-28, No. 2.
- [KRA 93] A. Krakasoglu *et al*, "Identification and Decentralized Adaptive Control using dynamical Neural Networks with Application to Robotic Manipulators, " *IEEE, Trans. Neural Networks*, vol. 4, no.6, Nov. 1993.
- [KRS 95] M. Krstic, I. Kanellakopoulos and P. Kokotovic, *Nonlinear and Adaptive Control Design*, John Wiley & Sons, Inc., 1995.
- [KUS 93] J. G. Kuschewski, and al ,"Application of Feedforward Neural Networks to Dynamique System Identification and Control", *IEEE Transactions on Control Systems Technology*. Vol. 1, No. 1, March 1993.
- [KWA 95a] C. M. Kwan and al, "Robust Neural Network Control of Flexible-Joint Robots," *Proceeding of the 34<sup>th</sup> Conference in decision & Control*, pp. 1296-1301, December 1995.
- [KWA 95b] C. M. Kwan and al, "Robust Adaptive Control of Robots Using Neural Network : Global Traking Stabilité," *Proceeding of the 34<sup>th</sup> Conference in decision & Control*, pp. 1846-1850, December 1995.
- [LAB 97] S. Labiod, H. Chekireb et M. S. Boucherit, "Commande Floue Adaptative d'un Robot Manipulateur," *Proceeding des 17èmes journées Tunisiennes d'Electrotechnique et d'Automatique*, 5 et 6 Nov. 1997, Nabeul-Tunisie.
- [LAN 79] I. D. Landau, *Commande Adaptative Aspects Pratiques et Théoriques*, Masson, 1979.
- [LEA 90] M. B. Leahy, "Model Based Auxiliary Input Control : Development and experimental analysis," *proceeding of the 29<sup>th</sup> conf. on Dec. and Control Honolulu, Hawaii, dec. 1990*
- [LEW 93] F. L. Lewis and al, "Neural Net Robot Controller With Guaranteed Traking Performances," *Proc. IEEE Int. Symp. Intellegent Control* , 1993.
- [MIL 92] W. T. Miller, R. Sutton , P.J. Werbos, *Neural Networks for Control*, The MIT Press, Cambridge, Massachusetts, London, 1992.
- [NAR 90a] K.S. Narendra , K. Parthasarathy " Identification and Control of Dynamical Systems using Neural Networks," *IEEE Trans, Neural Networks*, vol. 1, pp. 4-27, 1990.

- [NAR 90b] K.S. Narendra , K. Parthasarathy “ Gradient Methods for the Optimisation of Dynamical Systems Containing Neural Networks,” *IEEE Trans, Neural Networks*, vol. 2, NO. 2, pp. 252-262, 1990.
- [OPP 89] A. V. Oppenheim and R. W. Shafer, *Digital Signal Processing*, Prentice-Hall , 1989.
- [OSS 89] K. A. Ossman, “Indirect Adaptive Control for Interconnected Systems,” *IEEE Trans. on Automatic Control*, Vol. 34, N° 8, pp 908-911, August 1989.
- [OZA 91] T. Ozaki et al, “Trajectory Control of Robotic Manipulators,” *IEEE Trans. On Industrial Electronics*, vol. 38, NO. 3, 1991.
- [PAU 81] R. P. Pau et al, “Kinematic Control Equations for Simple Manipulators,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, no. 6, pp. 445-449, June 1981.
- [REN 95] J. M. Renders, *Algorithmes génétiques et réseaux de neurones*, Edition HERMES, Mars 1995.
- [RIV 95] I. Rivals, *Modélisation et commande de processus par réseaux de neurones ; Application au pilotage d'un véhicule autonome*, Thèse de doctorat de l'université Paris 6.
- [SAA 94] M. Saad, L. A. Dessait, P. Bigras and K. El-haddad, “Adaptive versus neural network Control : Application to Robotics, ” *Int. J. of Adaptive Cont. And Sign. Processing*, Vol. 8, pp 223-236, 1994.
- [SAN 92] R. M. Sanner and J-J E. Slotine, “Gaussian Networks for Direct Adaptive Control,” *IEEE Trans, Neural Networks*, vol. 3, N°6, Nov 1992.
- [SER 89] H. Seraji, “Decentralized Adaptive Control of Manipulators: Theory, Simulation, and Experimentation”, *IEEE Transactions in Robotics and Automation*, Vol. 5, No. 2, April 1989.
- [SHI 92] L. Shi and S. K. Singh, “Decentralised Adaptive Controller Design for Large-Scale Systems with Higher Order Interconnections,” *IEEE Trans. on Automatic Control*, Vol. 37, N° 8, pp 1106-1118, August. 1992.
- [SHU 96a] D. Shukla and F. W. Paul, “Computationally Efficient Control of Nonlinear Systems using Orthonormal Activation Function Based Neural Networks”, *IFAC 1996, 13<sup>th</sup> Triennial World Congress, San Francisco, USA*, pp 55-60.
- [SHU 96b] S. Ge. Shuzhi, “Robust Adaptive Control of Robots Based on Static Neural Networks”, *IFAC 1996, 13<sup>th</sup> Triennial World Congress, San Francisco, USA*, pp 139-144.
- [SLO 88] J. J. Slotine and W. Li, “Adaptive Manipulator Control : A case study,” *IEEE Trans. Aut. Contr.*, Vol.33, No. 11, pp. 995-1002, Nov. 1988.
- [SLO 88] J. J. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ :Prentice Hall, 1991.
- [STO 87] D. Stoten, “Modeling and Control of Elastic Joint Robots,” *ASME, Journal of Dynamics System Measurement and Control*, vol. 109, pp.310-319, 1987.
- [SUN 94] Sunid Elanayar V. T. and Yung C. Shin, “Radial Basis function Neural Network for Approximation and Estimation of Nonlinear Stochastic Dynamic systems,” *IEEE Tran. on Neural Networks*, Vol.5, N° 4, july 1994.
- [VES 97] V. Vesely , “Robust Adaptive Decentralised Control for a Class of Complex Systems,” *IFAC-IFIP-IMACS Conference, Control of Industrial Systems*, vol. 3, pp 69-74, 20-22 May 1997, Belfort, France
- [VIB 87] C. Vibet, *Robots : Principe et Contrôle*, Ellipse, Paris, 1987.
- [WAR 92] K. Warwick, G. W. Irwin and K. G. Hunt, *Neural Networks for Control and Systems*, Peter Pergrinus LTD, on behalf of the Institution of Electrical Engineers. Februry 1992.
- [ZIA 96] S. M. Ziauddin and A. M. S. Zalzala, “Model-Based Neural Networks Compensation for Uncertainties of Robotic Arms”, *IFAC 1996, 13<sup>th</sup> Triennial World Congress, San Francisco, USA*, pp 157-162.