

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

ECOLE NATIONALE POLYTECHNIQUE

DER

GENIE ELECTRIQUE ET INFORMATIQUE

FILIERE: "AUTOMATIQUE"

Projet de fin d'études

En vue de l'obtention du diplôme
d'ingénieur d'état en automatique

Thème:

Commande adaptative décentralisée
par réseaux de neurones artificiels:

Application à un bras de robot
manipulateur (PUMA 560)

Proposé par:

D. BOUKHETALA
F. BOUDJEMA

Etudié par:

S. BACHIR
M. LAKEHAL AYAT

Dirigé par:

D. BOUKHETALA
F. BOUDJEMA

Promotion: 1997

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE**

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

ÉCOLE NATIONALE POLYTECHNIQUE

DER

GENIE ELECTRIQUE ET INFORMATIQUE

FILIERE: "AUTOMATIQUE"

Projet de fin d'études

En vue de l'obtention du diplôme
d'ingénieur d'état en automatique

Thème:

Commande adaptative décentralisée
par réseaux de neurones artificiels:

Application à un bras de robot
manipulateur (PUMA 560)

Proposé par:

**D. BOUKHETALA
F. BOUDJEMA**

Etudié par:

**S. BACHIR
M. LAKEHAL AYAT**

Dirigé par:

**D. BOUKHETALA
F. BOUDJEMA**

Promotion: 1997

ENP. 19 Avenue Hassen BADI EL-HARRACH ALGER

DEDICACES

*A ma mère et a mon père
A mes quatre frères
A toute ma famille
A tous mes amis*

BACHIR SMAIL

*A mes très chers parents
A ma tante et son époux
A mes deux frères Djallal et Ahmed
A mes cousins
En particulier Lamine et Fella
A toute ma famille
A tous mes amis*

LAKEHAL-AYAT MOHSEN

REMERCIEMENTS



Il nous est particulièrement agréable de remercier nos deux promoteurs Mr. Boukhetala et Mr. Boudjema pour l'aide précieuse qu'ils nous ont apportés pour réaliser ce mémoire.

Nous tenons à exprimer également nos remerciements à messieurs les membres de jury, qui nous ont fait l'honneur de juger ce travail.

Notre sincère reconnaissance va à nos parents qui ont aidés et encouragés tout au long de nos études.

La réalisation matérielle de ce travail n'aurait pas été possible sans l'aide précieuse de B. Nabil et de l'agence publicitaire Lotus Design que nous tenons à remercier vivement.

Sans oublier d'exprimer nos remerciements à tous nos amis, et tout ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire.

Que les enseignants qui ont contribué à notre formation reçoivent notre gratitude.

SOMMAIRE



INTRODUCTION GENERALE

CHAPITRE I : Modélisation du robot manipulateur

1.1. Introduction	3
1.2. Présentation du robot manipulateur	4
1.3. Modélisation cinématique	4
1.3.1. Représentation de DENAVIT-HARTENBERG	4
1.3.2. Modélisation géométrique directe	7
1.3.3. Modélisation cinématique inverse	7
1.4. Génération de trajectoires	8
1.5. Modélisation dynamique	9
1.5.1. Formalisme d'EULER-LAGRANGE	9
1.5.2. Modèle dynamique du PUMA 560	11
1.5.3. Résultats de simulation	13
1.6. Conclusion	13

CHAPITRE II : Introduction aux réseaux de neurones

2.1. Introduction	15
2.2. Les réseaux de neurones	15
2.2.1. Le neurone	15
2.2.2. Les connexions	18
2.2.3. Les réseaux de neurones statiques	18
2.2.4. Les réseaux de neurones dynamiques	19
2.3. L'apprentissage	20
2.3.1. L'apprentissage supervisé	20
2.3.2. L'apprentissage non supervisé	20
2.3.3. L'apprentissage des réseaux de neurones statiques	20
2.3.4. L'apprentissage des réseaux de neurones dynamiques	22
2.4. Les réseaux de neurones et la commande	23
2.5. Conclusion	24

CHAPITRE III : Commande adaptative par RNA

3.1. Introduction	25
3.2. Schéma de commande adaptative par RNA selon l'approche centralisée de LEWIS	25
3.2.1. Formulation du problème	26
3.2.2. Structure de contrôleur adaptatif neuronal	27

3.2.2.1.	Le réseau de neurones	27
3.2.2.2.	Etude mathématique préliminaire	30
3.2.2.3.	Schéma de commande	31
3.2.2.4.	Les algorithmes d'adaptations	33
3.3.	Simulation	35
3.4.	Conclusion	37
CHAPITRE IV : Commande adaptative décentralisée par RNA		
	Introduction	47
PARTIE 1 : Synthèse d'une commande adaptative décentralisée par RNA		
4.1.1.	Introduction	48
4.1.2.	Description de schéma de commande adaptatif décentralisé par RNA	48
4.1.3.	Analyse de la stabilité	51
4.1.4.	Simulation	53
PARTIE 2 : Etude et conception d'une commande adaptative décentralisée par RNA		
4.2.1.	Introduction	63
4.2.2.	Structure du régulateur adaptatif décentralisée par RNA	63
4.2.2.1.	Position du problème	63
4.2.2.2.	Schéma de commande	64
4.2.3.	Simulation	73
	Conclusion	83
	CONCLUSION GENERALE	84
	ANNEXES ET BIBLIOGRAPHIE	85

المدسة الوطنفة المعمدة الففنفاف
BIBLIOTHEQUE — المكفبفة
Ecole Nationale Polytechnique



INTRODUCTION GENERALE

INTRODUCTION GENERALE

La commande des processus industriels constitue un objectif fondamental dans le domaine technologique, cette commande doit être choisie de façon compatible avec les caractéristiques et les phénomènes imprévisibles qui régissent leurs dynamiques. Il s'avère donc nécessaire d'opter pour une commande à caractère adaptatif.

De point de vue pratique, celle-ci permet l'ajustement automatique et en temps réel des régulateurs, mais reste qu'elle nécessite d'assez puissants calculateurs proportionnelle à la taille et à la complexité du modèle. Ainsi, l'élaboration des algorithmes d'adaptations centralisés et le traitement d'un important volume d'information rend le calcul lent et parfois même impossible pour les systèmes non linéaires fortement couplés.

Pour pallier à ces inconvénients les automaticiens se sont intéressés ces deux dernières décennies à la commande décentralisée.

Pour un système composé de plusieurs sous-système interconnectés, la commande décentralisée a pour objectif de calculer plusieurs contrôleurs locaux, chacun d'eux utilise les informations de son sous-système et commande l'entrée de ce dernier. L'ensemble des contrôleurs coopérant pour stabiliser le système interconnectés global.

Les capacités des réseaux de neurones artificiels (RNA) d'approximer et de généraliser n'importe quelle fonction non linéaire permettent d'envisager leur introduction dans ce type de commande. En effet, les possibilités offertes par les RNA en matière d'identification et de commande des systèmes dynamiques complexes peuvent être exploitées dans la conception de nouvelles techniques de commande adaptative décentralisée.

A fin d'appliquer les différentes techniques de commande adaptative par RNA, nous avons été conduit à mettre au point un modèle pour les système à commander. Ainsi, le premier chapitre fera l'objet d'une modélisation cinématique et dynamique d'un bras de robot à trois degrés de liberté, en l'occurrence le PUMA 560 de *puma unimation inc.* étant donné les fortes interconnexions et les non linéarités qui régissent sa dynamique, ce genre de bras manipulateur permettra d'évaluer les performances et la robustesse des schéma de commande proposés.

Le second chapitre s'intéresse à la théorie générale des réseaux de neurones artificiels, son objectif est de rappeler les principales structures et algorithmes d'apprentissage afin de permettre une compréhension aisée des techniques utilisées.

Le troisième chapitre intitulé « *Commande adaptative par RNA* » est consacré à l'étude détaillée d'un schéma de commande adaptative centralisé par RNA. A travers ce schéma, on verra l'élaboration d'un nouveau algorithme d'adaptation en temps réel ainsi que l'application de la théorie de la stabilité de LYAPUNOV afin d'analyser la stabilité des mécanismes d'adaptations développer.

Le quatrième chapitre intitulé « *Commande adaptative décentralisée par RNA* » est décomposé en deux parties. Dans la première, le schéma de commande adaptative centralisée par RNA développé au chapitre précédent sera étendu à la structure décentralisée. L'étude et la conception d'une nouvelle technique de commande adaptative décentralisée par RNA d'une façon détaillée fera l'objet de la second partie.

Une conclusion générale rassemble des réflexions générales, une comparaison ainsi que des perspectives. En fin de mémoire, les paramètres du PUMA560 et une démonstration de la stabilité d'un mécanisme d'adaptation ainsi qu'un tableau comparative sont regroupés dans trois annexes, ceci dans le but d'alléger la lecture de la thèse.



CHAPITRE I

**MODELISATION DU ROBOT
MANIPULATEUR**

1.1. INTRODUCTION :

Le développement des technologies nouvelles, particulièrement dans les domaines de l'électronique, de l'informatique et de la mécanique de précision, a donné naissance à une nouvelle discipline qu'est la robotique. Il s'ensuit, que la modélisation et la commande des robots manipulateurs constituent actuellement, l'un des plus importants axes de la recherche en matière de robotique.

Du mot tchèque « **ROBOTA** » -1921- qui signifie ouvrier travailleur, plusieurs autres définitions sont proposées dans la littérature. Larousse définit le robot comme étant « un appareil automatique pouvant se substituer à l'homme pour exécuter diverses actions ». Une définition plus précise est proposée par « The Robot Institute Of America », qui définit le robot comme étant un manipulateur multi-fonctionnelle reprogrammable, conçu pour déplacer toutes sortes de matériels tels les outils ou autres dispositifs spéciaux, à travers des trajectoires préprogrammées en vue d'exécuter une variété de tâche (Vibet,1987).

L'application industrielle de robots réclamant que ceux-ci puissent être commandés et programmés à partir de coordonnées rapportées à l'espace des tâches. L'utilisation de la transformation proposée par DENAVIT-HARTENBERG en 1955, facilite la description géométrique du manipulateur, et permet d'aboutir au modèle cinématique et géométrique direct et inverse du robot.

La connaissance du modèle dynamique du robot est nécessaire pour pouvoir synthétiser au mieux une loi de commande. Le formalisme d'EULER-LAGRANGE permet d'obtenir le modèle dynamique direct, la dynamique inverse est obtenue selon le principe de d'ALEMBERT (formalisme de NEWTON-EULER).

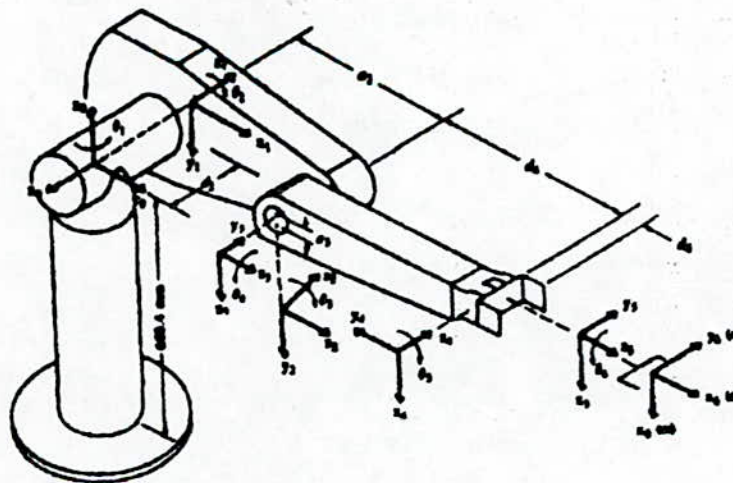


fig (1.1) Présentation du PUMA 560

1.2. PRESENTATION DU ROBOT : [2][5]

Dans le cadre de nos travaux, nous avons utilisé un bras manipulateur le plus courant en industrie, il s'agit notamment du PUMA 560 de Puma Unimation inc., dont on a considérés uniquement les trois articulations rotationnelles θ_1 , θ_2 , et θ_3 , présenté sur la figure (1.1).

Pour pouvoir établir les différents modèles, plusieurs hypothèses doivent être prises en considération [3][24]:

- Les frottement sont de nature visqueuse, et linéaire par rapport à la vitesse généralisée.
- Les différentes liaisons sont rigides.
- Les actionneurs sont idéaux, c'est-à-dire que la force généralisée est proportionnelle au signal de commande.
- Les capteurs ont des gains unitaires, et de dynamique négligée.

1.3. MODELISATION CINEMATIQUE :

Un manipulateur peut être considéré comme chaîne de liaisons connectées par des articulations, chaque liaison est caractérisé par son propre repère, l'orientation d'un repère par rapport à un autre est déduite par une transformation dite homogène.

L'exécution d'une tâche en robotique nécessite au préalable de positionner et d'orienter l'organe terminal. En effet, l'application industrielle des robots réclame que ceux-ci puissent être commandés à partir de coordonnées rapportées dans un référentiel fixe et arbitraire dit *espace de tâche*.

1.3.1. Représentation de DENAVIT-HARTENBERG : [24]

Denavit et Hartenberg propose une méthode systématique, afin de faire correspondre à chaque articulation, un système de coordonnées (référentiel) puis d'en déduire les transformations qui relient chaque référentiel au référentiel précédent, ce qui permet à terme d'exprimer les coordonnées de l'outil terminal par rapport au référentiel de la base du manipulateur.

Pour pouvoir appliquer cette transformation il faut d'abord fixer les différents repères dans chaque articulation d'une façon successive, en procédant comme suit:

ETAPE 1:

_ Numéroter chaque liaison et articulation en commençant de la base, notée liaison « 0 », jusqu'à l'élément terminal, notée liaison « n », la liaison « i » se déplace soit autour (rotation) de l'axe « i-1 », soit le long (prismatique) de cet axe.

ETAPE 2:

_ Etablir le repère de chaque articulation en suivant les règles suivantes:

- L'axe z_{i-1} correspond au déplacement du segment « i » dans le référentiel « i-1 ».

- L'axe x_i est choisi perpendiculaire à z_{i-1} , tel que:
 - Si z_i et z_{i-1} ne se coupent pas, x_i est choisi colinéaire à la perpendiculaire commune à z_i et z_{i-1} .
 - Si z_i et z_{i-1} se coupent, on choisit x_i arbitrairement.
- L'axe y_i est choisi de telle sorte à former un trièdre droit.

ETAPE 3:

Définir les paramètres θ_i , d_i , α_i , et a_i .

- θ_i : l'angle entre x_{i-1} et x_i obtenu par rotation de x_{i-1} vers x_i autour de z_{i-1} .
- d_i : coordonnée de l'origine O_i sur l'axe z_{i-1} , autrement dit, la distance entre O_{i-1} et l'intersection de z_{i-1} avec x_i .
- α_i : l'angle obtenu on visant z_{i-1} vers z_i autour de x_i .
- a_i : distance entre z_{i-1} et z_i le long de x_i .

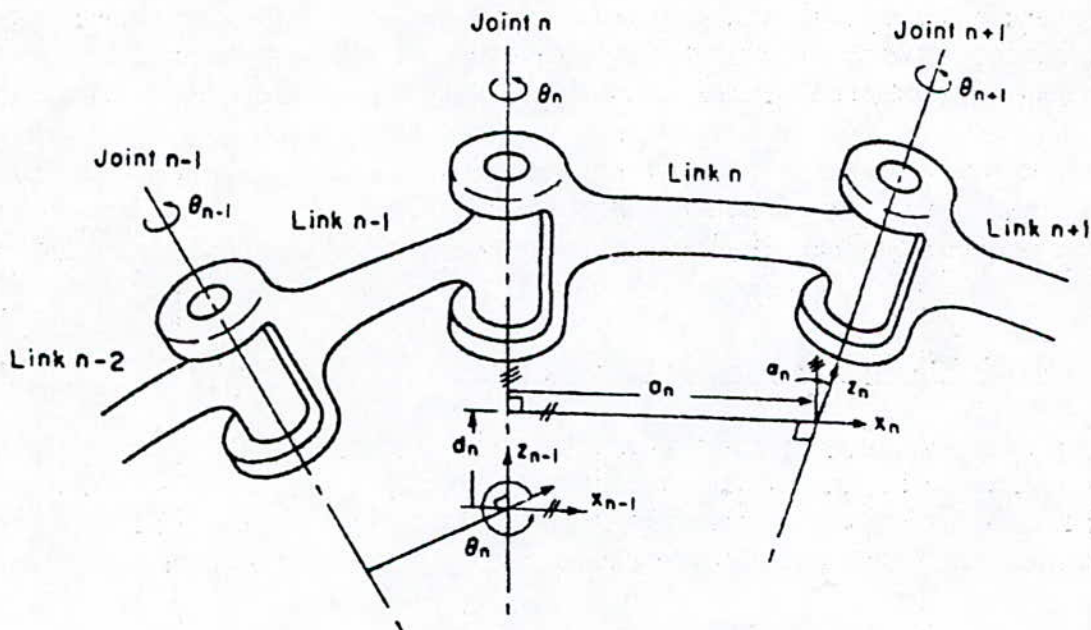


fig (1.2) Présentation des référentiels selon DENAVIT-HARTENBERG

ETAPE 4:

Former la matrice de transition homogène relative à chaque articulation, tout point exprimé dans le référentiel « i » peut être exprimé dans le référentiel « i-1 » en effectuant les opérations suivantes :

1. Translation le long de l'axe z_{i-1} d'une distance d_i .
2. Rotation autour de l'axe z_{i-1} d'un angle θ_i .
3. Translation le long de l'axe x_i d'une distance a_i .
4. Rotation autour de l'axe x_i d'un angle α_i .

Ce qui donne la matrice de transformation homogène:

$${}_{i-1}T^i = \text{Trans}(z,d) \cdot \text{Rot}(z,\theta) \cdot \text{Trans}(x,a) \cdot \text{Rot}(x,\alpha)$$

$${}_{i-1}T^i = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i) \cdot \sin(\theta_i) & \sin(\alpha_i) \cdot \sin(\theta_i) & a_i \cdot \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cdot \cos(\theta_i) & -\sin(\alpha_i) \cdot \cos(\theta_i) & a_i \cdot \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Nous avons appliqué l'algorithme donné ci-dessus sur le Puma 560, le choix des différents repères est illustré sur la figure (1.1), suivant ce choix, on construit le tableau suivant:

i	Vari- able	α_i	a_i	d_i	θ_i
1	θ_1	-90	0	0	θ_1
2	θ_2	0	l_2	d_2	θ_2
3	θ_3	0	l_3	0	θ_3

Ainsi nous aboutissons aux matrices de transformations homogènes, représentant les trois rotations du PUMA 560:

$${}^0T^1 = \begin{bmatrix} C1 & 0 & -S1 & 0 \\ S1 & 0 & C1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad {}^1T^2 = \begin{bmatrix} C2 & -S2 & 0 & a_2 \cdot C2 \\ S2 & C2 & 0 & a_2 \cdot S2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad {}^2T^3 = \begin{bmatrix} C3 & -S3 & 0 & a_3 \cdot C3 \\ S3 & C3 & 0 & a_3 \cdot S3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T^2 = \begin{bmatrix} C1 \cdot C2 & -C1 \cdot S2 & -S1 & a_2 \cdot C1 \cdot C2 - d_2 \cdot S1 \\ S1 \cdot C2 & -S1 \cdot S2 & C1 & a_2 \cdot C2 \cdot S1 + d_2 \cdot C1 \\ -S2 & -C2 & 0 & -a_2 \cdot S2 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$${}^0T^3 = \begin{bmatrix} C1 \cdot C23 & -C1 \cdot S23 & -S1 & C1 \cdot (a_2 \cdot C2 + a_3 \cdot C23) - d_2 \cdot S1 \\ S1 \cdot C23 & -S1 \cdot S23 & C1 & S1 \cdot (a_2 \cdot C2 + a_3 \cdot C23) + d_2 \cdot C1 \\ -S23 & -C23 & 0 & -(a_2 \cdot S2 + a_3 \cdot S23) \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

avec : $C_i = \cos(\theta_i)$, $S_i = \sin(\theta_i)$, $C_{ij} = \cos(\theta_i + \theta_j)$, $S_{ij} = \sin(\theta_i + \theta_j)$.

1.3.2. Modélisation géométrique directe : [3]

La modélisation géométrique directe permet de connaître la situation du robot (position et orientation) en fonction de sa configuration, grâce à cette relation entre les coordonnées opérationnelles et les coordonnées généralisées. Les transformations précédentes permettent d'exprimer la position de l'organe terminal par rapport au repère R_0 sous la forme d'une fonction $\bar{r}(R_0)$ à partir des variable articulaires q_i :

$$\bar{r}(R_0) = F(\bar{q}) \quad \text{avec} \quad \begin{cases} \bar{q} = [q_1 \quad q_2 \quad q_3]^T \\ \bar{r} = [P_x \quad P_y \quad P_z \quad 1]^T \\ F(\bar{q}) = {}_0T^3 \cdot {}_3\bar{r} \end{cases}$$

et ${}_3\bar{r}$: coordonnées de l'outil terminal dans R_3 .

$F(\bar{q})$: fonction vectorielle.

on a : ${}_3\bar{r} = [0 \ 0 \ 0 \ 1]$ d'où
$$\begin{cases} P_x \\ P_y \\ P_z \\ 1 \end{cases} = \begin{cases} \bar{r} = {}_0T^3 \cdot {}_3\bar{r} \\ C1.(L2.C2 + L3.C23) - d_2.S1 \\ S1.(L2.C2 + L3.C23) + d_2.C1 \\ -(L2.S2 + L3.S23) \\ 1 \end{cases}$$

avec $\bar{q}^T = [\theta_1 \quad \theta_2 \quad \theta_3]$.

donc :

$$\begin{cases} P_x = C1.(L2.C2 + L3.C23) - d_2.S1 \\ P_y = S1.(L2.C2 + L3.C23) + d_2.C1 \\ P_z = -(L2.S2 + L3.S23) \end{cases}$$

Ce système d'équations est appelé modèle géométrique du robot, c'est à partir de cette représentation qu'on peut tirer les coordonnées généralisées q_i en fonction des coordonnées cartésiennes, ce qui constitue la cinématique inverse.

1.3.3. Modélisation cinématique inverse:

Les bras manipulateurs commandés par ordinateur, le sont généralement par les variables articulaires, alors que les objets à manipuler sont eux exprimés par leurs coordonnées cartésiennes.

Le problème cinématique direct est toujours soluble. On peut en effet toujours déduire une position géométrique à partir d'un vecteur de coordonnées articulaire. Par contre, le problème inverse ne l'est pas, ce qui signifie que l'on ne peut pas atteindre n'importe quelle position, sous une attitude quelconque.

En générale, le problème cinématique inverse peut être résolu par différentes méthodes, telle que la transformation inverse (Paul et al., 1981), approche géométrique (Lee and Ziegler,

1984), méthode des matrices duales (Denavit, 1956).

Toutefois, et dans le but de ne pas abourdir cette exposé, nous avons préféré donner directement les expressions littérales des solutions tout en renvoyant le lecteur soucieux d'approfondir la question à la référence [8].

- $\theta_1 = \text{Tan}^{-1} \left[\frac{-\text{arm} \cdot P_Y \cdot \sqrt{P_X^2 + P_Y^2 - d_2^2} - P_X \cdot d_2}{-\text{arm} \cdot P_X \cdot \sqrt{P_X^2 - P_Y^2 - d_2^2} - P_Y \cdot d_2} \right] \quad (-\pi \leq \theta_1 \leq \pi).$
- $\theta_2 = \text{Tan}^{-1} \left[\frac{\sin(\alpha) \cdot \cos(\beta) + \text{arm} \cdot \text{elbow} \cdot \cos(\alpha)}{\cos(\alpha) \cdot \sin(\beta) - \text{arm} \cdot \text{elbow} \cdot \sin(\alpha) \cdot \cos(\beta)} \right] \quad (-\pi \leq \theta_2 \leq \pi).$
- $\cos(\alpha) = \frac{-\text{arm} \cdot \sqrt{P_X^2 + P_Y^2 - d_2^2}}{\sqrt{P_X^2 + P_Y^2 + P_Z^2 - d_2^2}} \quad \cos(\beta) = \frac{P_X^2 + P_Y^2 + P_Z^2 - d_2^2 + l_2^2 - l_3^2}{2 \cdot l_2 \cdot \sqrt{P_X^2 + P_Y^2 + P_Z^2 - d_2^2}}$
 - $\sin(\alpha) = \frac{-P_Z}{\sqrt{P_X^2 + P_Y^2 + P_Z^2 - d_2^2}}$
 - $\theta_3 = \phi - \frac{\pi}{2}, \quad (-\pi \leq \theta_3 \leq \pi)$
- $\cos(\phi) = \frac{l_2^2 + l_3^2 - P_X^2 + P_Y^2 + P_Z^2 - d_2^2}{l_3^2} \quad \sin(\phi) = \text{arm} \cdot \text{elbow} \cdot \sqrt{1 - \cos(\phi)^2}$

REMARQUE : *arm* et *elbow* sont des indicateurs de configurations associées aux trois première articulations du PUMA 560 [5].

1.4. GENERATION DE TRAJECTOIRES : [26]

_ Imposer une trajectoire à l'outil terminal reliant un point d'équilibre à un autre, nécessite une continuité en position, vitesse et en accélération . En effet, le comportement dynamique du robot exige d'imposer des trajectoires réalisables, ainsi on offre à ce dernier la possibilité de poursuivre cette trajectoire avec des commandes réalisables.

_ Il existe deux trajectoires standard en robotique, la trajectoire polynomiale et la trajectoire cycloïdale. Toutefois, nous pensons que la seconde est mieux adapté à nos travaux, car elle offre en plus de la continuité en position, vitesse et accélération, au contraire de la trajectoire polynomiale.

_ La trajectoire cycloïdale est donnée par la fonction :

$$\theta(t) = \begin{cases} \theta(0) + \frac{\Delta}{2 \cdot \pi} \cdot [\omega \cdot t - \sin(\omega \cdot t)] & 0 \leq t \leq t_F \\ \theta(t_F) & t \geq t_F \end{cases}$$

avec : $\omega = \frac{2\pi}{t_F}, \quad \Delta = \theta(t_F) - \theta(t_0).$

1.5. MODELISATION DYNAMIQUE : [3][24]

La dynamique, comme son nom l'indique, fait intervenir la notion de force, elle établit la relations entre forces et mouvements. Ainsi, le modèle dynamique d'un bras manipulateur est une description mathématique de son comportement physique. Des approches conventionnelles comme l'approche d'EULER-LAGRANGE (E-L) ou celle de NEWTON-EULER (N-E), peuvent être systématiquement appliquées, en vus de déterminer le modèle dynamique .

1.5.1. Formalisme d'EULER-LAGRANGE :

Le formalisme d'EULER-LAGRANGE permet d'établir un système d'équations différentielles reliant les coordonnées généralisées aux forces généralisées, obtenant ainsi le modèle dynamique du robot manipulateur.

L'équation de (E-L) est donné par: [3][24]

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_i} \right] - \left(\frac{\partial L}{\partial q_i} \right) = \tau_i \quad , i=1,n . \tag{1.1}$$

- L : Fonction de Lagrange.
- E_C : Energie cinétique totale du manipulateur.
- E_P : Energie potentielle totale du manipulateur.
- τ_i : Force généralisée.
- n : Degré de liberté.

L'expression du Lagrangien est définit par:

$$L = E_C - E_P = K - P. \tag{1.2}$$

a. ENERGIE CINETIQUE :

Le calcul de l'énergie cinétique exige la connaissance de la vitesse de chaque liaison. étant donnée un point de la liaison i définit par ces coordonnées _i rⁱ dans le repère R_i , l'expression de la vitesse est :

$${}_0V^i = \frac{d}{dt} ({}_0r^i) = \frac{d}{dt} ({}_0T^i \cdot {}_i r^i) \tag{1.2}$$

d'ou :

$${}_0V^i = \sum_{j=1}^i [U_{ij} \cdot \dot{q}_j] \cdot {}_i r^i \tag{1.3}$$

avec :

$$U_{ij} = \begin{cases} {}_0T^{j1} \cdot Q_{j \cdot j-1} T^i & j \leq i \\ 0 & j \geq i \end{cases} \quad \text{et } Q_j = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

L'énergie cinétique de l'élément (i) de la liaison (i) est:

$$dK_i = \frac{1}{2} \cdot \text{trace}(V_i \cdot V_i^T) \cdot dm = \frac{1}{2} \cdot \text{trace} \left[\sum_{p=1}^i \sum_{r=1}^i U_{ip} \cdot \left[{}_i r^r \cdot dm \cdot {}_i r^i \right] \cdot U_{ir}^T \cdot \dot{q}_p \cdot \dot{q}_r \right] \quad (1.4)$$

avec:

$$J_i = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \quad (1.5)$$

REMARQUE : J_i dépend uniquement de la distribution de la masse de la (i-ème) articulation, et non pas de la position relative de celle-ci, elle n'est donc calculable pour chaque articulation qu'une seule fois. donc, l'expression de l'énergie cinétique :

$$K_{\text{totale}} = \frac{1}{2} \cdot \sum_{i=1}^n \left[\sum_{p=1}^i \sum_{r=1}^i \text{trace}(U_{ip} \cdot [J_i] \cdot U_{ir}^T) \cdot \dot{q}_p \cdot \dot{q}_r \right] \quad (1.6)$$

b. ENERGIE POTENTIELLE :

_ Soit P_i l'énergie potentielle de l'articulation (i), elle sera donnée par:

$$P_i = -m_i \cdot g \cdot {}_0 r^i = -m_i \cdot g \cdot ({}_0 T^i \cdot {}_i r^i) \quad (1.7)$$

_ l'énergie totale du bras manipulateur sera donc:

$$P = \sum -m_i \cdot g^T \cdot ({}_0 T^i \cdot {}_i r^i) \quad (1.8)$$

avec : $g^T = [g_x \quad g_y \quad g_z \quad 1]$, dans le repère de base, d'ou : $g^T = [0 \quad 0 \quad -|g| \quad 1]$.

c. EQUATION DYNAMIQUE :

_ D'après l'expression du lagrangien L sera donnée par :

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{p=1}^i \sum_{r=1}^i \left[\text{trace}(U_{ip} \cdot J_i \cdot U_{ir}^T) \cdot \dot{q}_p \cdot \dot{q}_r \right] + \sum_{i=1}^n m_i \cdot g \cdot ({}_0 T^i \cdot {}_i r^i) \quad (1.9)$$

en appliquant l'équation (1.1) à cette fonction, on aura:

$$T_i = \sum_{j=i}^n \sum_{k=i}^j \left[\text{trace}(U_{jk} \cdot J_j \cdot U_{jk}^T) \dot{q}_k \right] + \sum_{j=1}^n \sum_{k=1}^i \sum_{m=1}^j \left[\text{trace}(U_{jkm} \cdot J_j \cdot U_{ji}^T) \dot{q}_k \dot{q}_m \right] - \sum_{j=1}^n \left[m_j g {}_0 T^i \cdot {}_i r^i \right] \quad (1.10)$$

$i=1, n.$

l'expression matricielle de (1.10) sera la suivante :

$$\mathbf{T} = \mathbf{M}(q) \cdot \ddot{\mathbf{q}} + \mathbf{H}(q, \dot{\mathbf{q}}) + \mathbf{G}(q) \quad (1.11)$$

avec :

- $\mathbf{M} \in \mathcal{R}^{n \times n}$ Matrice d'inertie symétrique et définie positive.
- $\mathbf{H} \in \mathcal{R}^n$ Couple dû à l'accélération de coriolis et centrifuge.
- $\mathbf{G} \in \mathcal{R}^n$ Couple dû à la gravitation.
- $\mathbf{T} \in \mathcal{R}^n$ Vecteur des forces généralisées.

1.5.2. Modèle Dynamique Du PUMA 560 :

Le calcul du modèle dynamique explicite à été effectué à l'aide de l'approche étudiée auparavant, nous nous contenterons donc ici de donner uniquement l'expression littérale correspondante au modèle dynamique du PUMA 560, et on fait référence au [5]. Le modèle dynamique du PUMA 560 sera:

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} I_1^* + I_2^* \cdot c_{23}^2 + I_3^* \cdot c_2^2 + I_4^* \cdot c_2 c_{23} & I_5^* \cdot s_{23} + I_6^* \cdot s_2 & I_5^* \cdot s_{23} \\ I_5^* \cdot s_{23} + I_6^* \cdot s_2 & I_7^* + I_4^* c_3 & I_8^* + 0.5 \cdot I_4^* c_3 \\ I_5^* \cdot s_{23} & I_8^* + 0.5 \cdot I_4^* c_3 & I_8^* \end{bmatrix} \begin{bmatrix} d^2 \theta_1 / dt^2 \\ d^2 \theta_2 / dt^2 \\ d^2 \theta_3 / dt^2 \end{bmatrix} +$$

$$\left[\begin{array}{l} (-2I_3^* s_2 c_2 - 2I_2^* s_{23} c_{23} - I_4^* (c_2 s_{23} + s_2 c_{23})) (d\theta_1 / dt) (d\theta_2 / dt) \\ + (-2I_2^* s_{23} c_{23} - I_4^* c_2 s_{23}) (d\theta_1 / dt) (d\theta_3 / dt) \\ + (I_6^* c_2 + I_5^* c_{23}) (d\theta_2 / dt)^2 + 2I_5^* c_{23} (d\theta_2 / dt) (d\theta_3 / dt) + I_5^* c_{23} (d\theta_3 / dt)^2 \\ \\ (I_3^* c_2 s_2 + I_2^* c_{23} s_{23} + 0.5 I_4^* (s_2 c_{23} + c_2 s_{23})) (d\theta_1 / dt)^2 \\ - I_4^* s_3 (d\theta_2 / dt) (d\theta_3 / dt) - 0.5 I_4^* s_3 (d\theta_3 / dt)^2 \\ \\ (I_2^* s_{23} c_{23} + 0.5 I_4^* c_2 s_{23}) (d\theta_1 / dt)^2 + 0.5 I_4^* s_3 (d\theta_2 / dt)^2 \end{array} \right] +$$

$$\begin{bmatrix} 0 \\ -(m_3 l_2 + 0.5 m_2 \cdot l_2) \cdot g \cdot c_2 - 0.5 m_3 \cdot l_3 \cdot g \cdot c_{23} \\ -0.5 \cdot m_3 \cdot l_3 \cdot g \cdot c_{23} \end{bmatrix}$$

Avec les notations suivantes:

$$I_1^* = I_{YY1} + I_{XX2} + m_2 d_2 (d_2 + e) + m_3 d_2^2 + I_{XX3}$$

$$I_2^* = I_{YY3} - I_{XX3}$$

$$I_3^* = I_{YY2} - I_{XX2} + m_3 l_2^2$$

$$I_4^* = m_3 l_2 l_3$$

$$I_5^* = 0.5 \cdot m_3 l_3 d_2$$

$$I_6^* = 0.5 \cdot m_2 l_2 \cdot (d_2 + e) + m_3 \cdot d_2 \cdot l_2$$

$$I_7^* = I_{ZZ2} + I_{ZZ3} + m_3 \cdot l_2^2$$

$$I_8^* = I_{ZZ3}$$

Les caractéristiques du robot, telles que les masses et les dimensions sont présentées dans l'annexe (A), ces paramètres sont issues d'une modélisation réelle du PUMA 560, disponibles dans la référence [2].

Pour représenter au mieux le PUMA 560 réelle, il faudra inclure l'effet de l'effecteur (poignet), ainsi les termes additifs qui lui sont due, seront :

$$\Delta D_{11} = (I_{ZZ1} + m_1 d_2^2 + I_{m1}) + (I_{YY1} - I_{XX1} + m_1 l_3^2) \cdot c_{23}^2 + m_1 \cdot l_2^2 c_2^2 + 2m_1 l_2 l_3 c_2 c_{23}$$

$$\Delta D_{12} = m_1 l_3 d_2 s_{23} + m_1 d_2 l_2 s_2$$

$$\Delta D_{13} = m_1 \cdot l_3 \cdot d_2 \cdot s_{23}$$

$$\Delta D_{22} = I_{ZZ1} + m_1 \cdot (l_2^2 + l_3^2) + I_{m2} + 2 \cdot m_1 \cdot l_2 \cdot l_3 \cdot c_3$$

$$\Delta D_{23} = I_{ZZ1} + m_1 l_2^2 + m_1 \cdot l_2 \cdot l_3 \cdot c_3$$

$$\Delta D_{33} = I_{ZZ1} + m_1 l_2^2 + I_{m3}$$

Avec I_{m_i} $i=1,3$ moments d'inertie des différents moteurs.

I_{XX1} , I_{YY1} , I_{ZZ1} : moments d'inertie total par rapport aux principaux axes de l'effecteur.

REMARQUE: Les couples de Coriolis totales se déduisent de l'ancienne expression juste en substituant les facteurs I_i^* par leur valeur augmentée des termes additifs dus à l'effet de l'effecteur.

- Effet de la masse de la charge :

La masse de la charge aura, sans nul doute, une importance capitale lors des simulations sur ordinateur, et pour modéliser son effet sur la dynamique du système, l'approche de la matrice jacobienne sera 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 dû à l'effet de la masse sera donné par :

$$T_{\text{masse}} = m \cdot J^T (J \cdot \ddot{q} + J(q, \dot{q}) \cdot \dot{q} + g)$$

avec: m : masse de la charge.

\dot{J} : la matrice jacobienne dérivée par rapport au temps.

1.5.3. Résultats De Simulation :

Le figures (1.4),(1.5) et (1.6) montrent respectivement les réponses en boucle ouverte des différentes articulation en position et en vitesse, pour une excitation indicielle, sans charge , sur un horizon de temps suffisant ($t_f = 10 \text{ sec}$).

CONCLUSION

La modélisation de processus industriel constitue l'une des plus importante phase pour la synthèse d'une loi de commande. ainsi, ce présent chapitre nous à permis de décrire les liens existants entre les différents grandeurs qui régissent le comportement d'un bras manipulateur.

Le modèle dynamique du PUMA 560 ainsi obtenus, met en évidence les fortes non-linearités et le couplage entre les articulations qui le composent. Ceci dit, la simulation en boucle ouverte reflète bien l'aspect instable de ce genre de robot manipulateur, d'ou, la nécessité d'une loi de commande robuste afin de garantir les performances exigées.

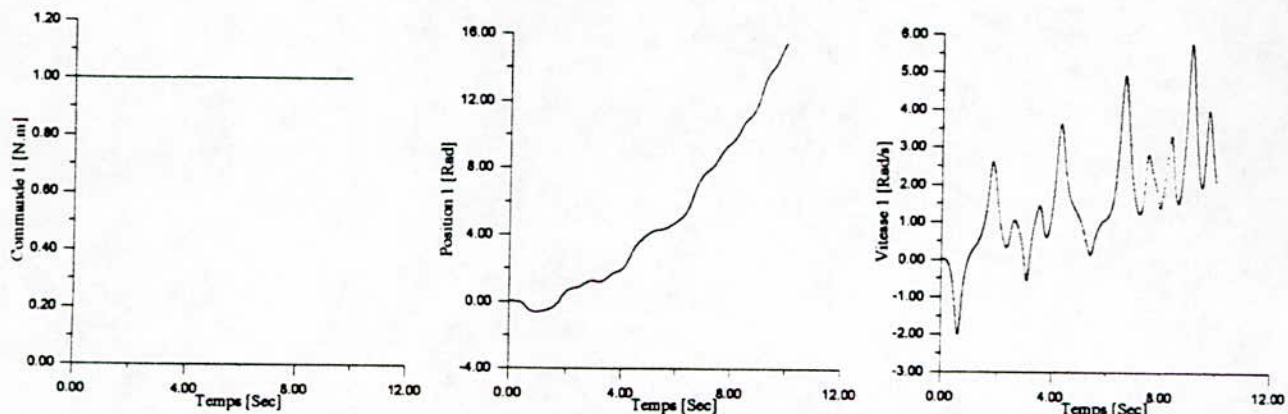


Figure I.4 Réponse indicielle de la première articulation.

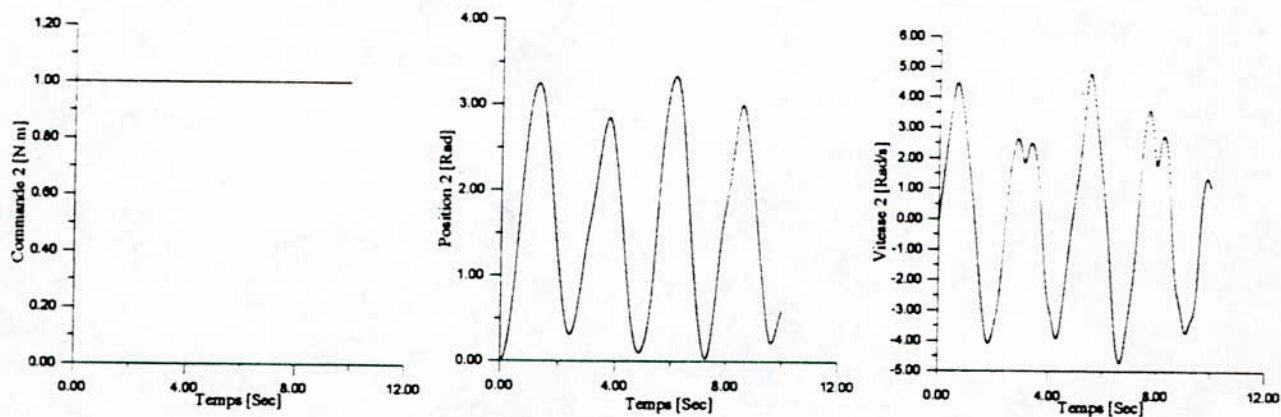


Figure I.5 Réponse indicielle de la deuxième articulation.

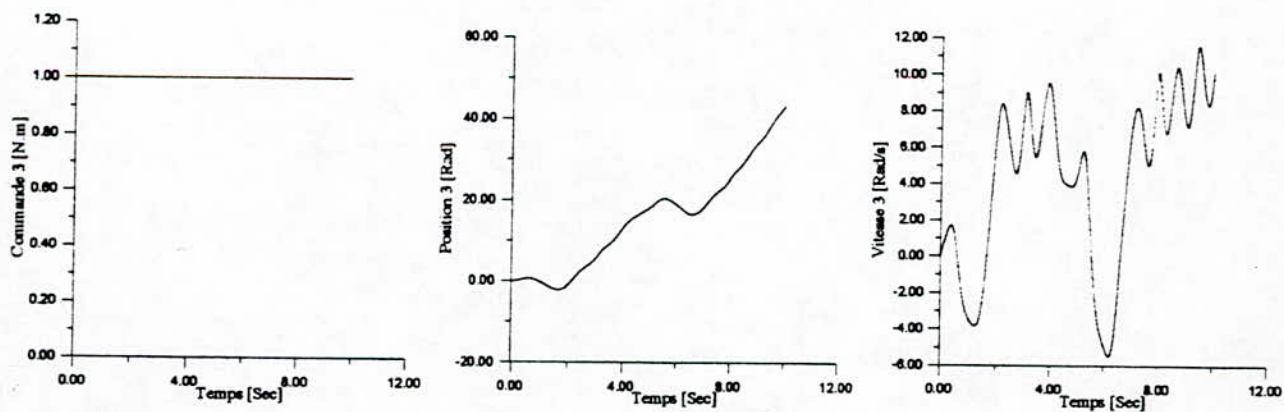


Figure I.6 Réponse indicielle de la troisième articulation

Réposes du PUMA 560 en Boucle ouverte.



CHAPITRE II

**INTRODUCTION AUX RESEAUX
DE NEURONES**

2.1. INTRODUCTION :

Depuis la nuit des temps, et face à une nature plein de mystères, mais parfaitement équilibrée, l'homme, motivé par ces désires, stimulés par la simple curiosité, ne cessent de lancé des défis qui le guide vers de nouvelles conquêtes de la science, afin de copier cette nature.

L'histoire de l'homme à vu l'évolution de sont intelligence. Seulement, la conquête de l'espace ne lui suffit plus, l'homme cette fois-ci essayi de se copier en imaginant, élaborant et perfectionnant des systèmes intelligents.

Mais, devant lui, le système le plus intelligent qu'il connaît, est sans doute son cerveau. Depuis et grâce aux développement que connaît la médecine, les chercheurs ont réussi à décrire le système nerveux humain, par l'introduction d'une notion fondamentale, *le neurone*.

Le cerveau est constitué essentiellement d'un ensembles de c'est petits cellules appeler *neurones*, connectés ensemble d'une manière complexe, non-organisé par les différents *synapses*.

Par analogie, pour nous, un réseau de neurones est composé essentiellement d'une ou plusieurs couches de neurones, *preception* pour Rosenblatt (1958), ou encore *adaline* pour Widrow, interconnectés d'une manière simple suivant une structure bien organisée.

Dans le présent chapitre on va d'abord donner quelques définitions sur les réseaux de neurones (éléments de base, principe de fonctionnement, ... etc.). Puis, les deux structures fondamentales des réseaux, les réseaux statiques et les réseaux dynamiques. On parlera ensuite, de la méthode usuelles de l'apprentissage de chaque réseau cité, et enfin on parlera de la capacité des réseaux de neurones à approximer n'importe quelle fonction.

2.2. LES RESEAUX DE NEURONES : [7][14]

La première modélisation d'un neurone date de 1943, présentée par Mac Culloch et Pitts, s'inspirant de leurs travaux sur les neurones biologiques, ils ont proposés le modèle suivant :

« Un neurone formel fait une somme pondérée des potentiels d'actions qui lui parviennent (chacun de ces potentiels est une valeur numérique qui représente l'état du neurone qui l'a émis). Puis s'active suivant la valeur de cette sommation pondérée. Si cette somme dépasse un certain seuil, le neurone est activé et transmet une réponse (sous forme de potentiel d'action) dont la valeur est celle de son activation. Si le neurone n'est pas activé, il ne transmet rien ».

Ce neurone formel est un automate booléen (sortie binaire), et qui ne considère qu'une certaine classe de RNA, d'ou la nécessité d'un modèle plus détaillés.

2.2.1. LE NEURONE : (Fig. 2.1)

Un neurone est considérée comme une *fonctionnelle* caractérisée par les éléments suivants:

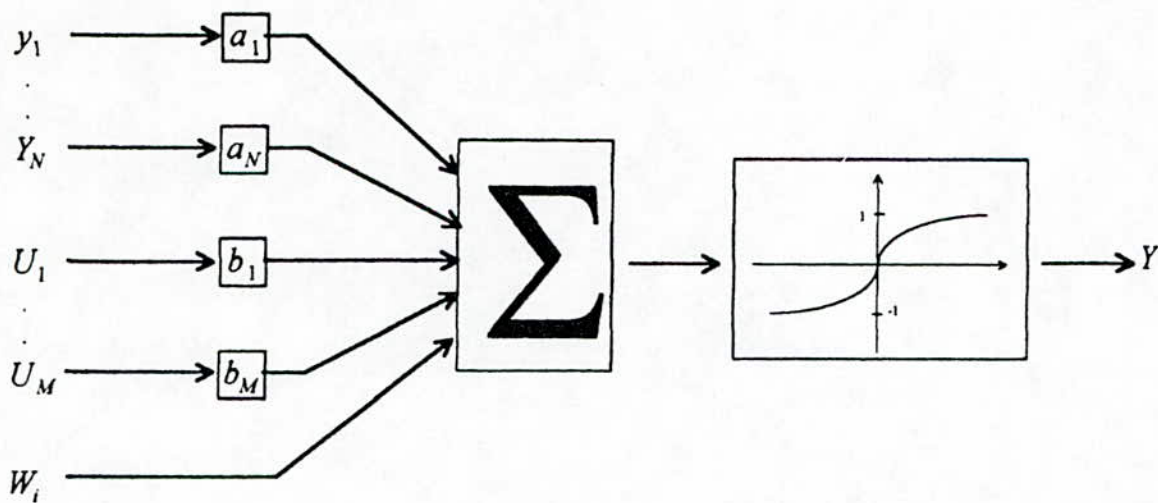


Fig. (2.1) Représentation d'un neurone

1. Un sommateur pondéré.
2. Un système dynamique linéaire SISO.
3. Une fonction non linéaire statique (fonction d'activation).

chacun de ces éléments sera décrit dans les sections suivantes:

a. Le Sommateur Pondéré :

Le sommateur pondéré est décrit par l'équation suivante :

$$v_i(t) = \sum_{j=1}^N a_{ij} \cdot y_j(t) + \sum_{k=1}^M b_{ik} \cdot u_k(t) + w_i$$

cette somme est défini comme un prétraitement sur les entrées.

b. Le Système Dynamique :

Le système linéaire SISO possède v_i comme entrée, x_i comme sortie et $H(s)$ comme fonction de transfert (avec la transformée de Laplace), tel que

$$\bar{x}_i(s) = H(s) \cdot \bar{v}_i(s) \quad (2.1)$$

$H(s)$ peut être une des fonctions suivantes :

$$H(s) = 1$$

$$H(s) = \frac{1}{s}$$

$$H(s) = \frac{1}{1 + s \cdot T}$$

$$H(s) = \frac{1}{\alpha_0 + \alpha_1 \cdot s}$$

$$H(s) = e^{-sT}$$

Cette fonction $H(s)$ donne une dynamique au neurone car elle peut être un filtre ou un retard.

c. Fonction d'Activation :

Cette fonction est élémentaire pour un neurone, car elle transforme le signal x_i non borné à l'instant t en un signal borné y_i .

$$y_i = g(x_i) \quad (2.2)$$

Donc, la fonction $g(\cdot)$ doit être bornée, et en générale monotone non-décroissante, non linéaire, car elle augmente la capacité du réseau à approximer des fonctions complexes, facilitant ainsi la suppression des bruits. Seulement, les calcul et l'analyse sont plus complexes.

Pour nous, on choisit $g(x)$ comme la fonction sigmoïde $\sigma(x)$ possédant la propriété $\sigma(-\infty) = 0$ et $\sigma(\infty) = 1$, avec (Fig 2.2)

$$\sigma(x) = \frac{1}{1 + e^{-\alpha x}}$$

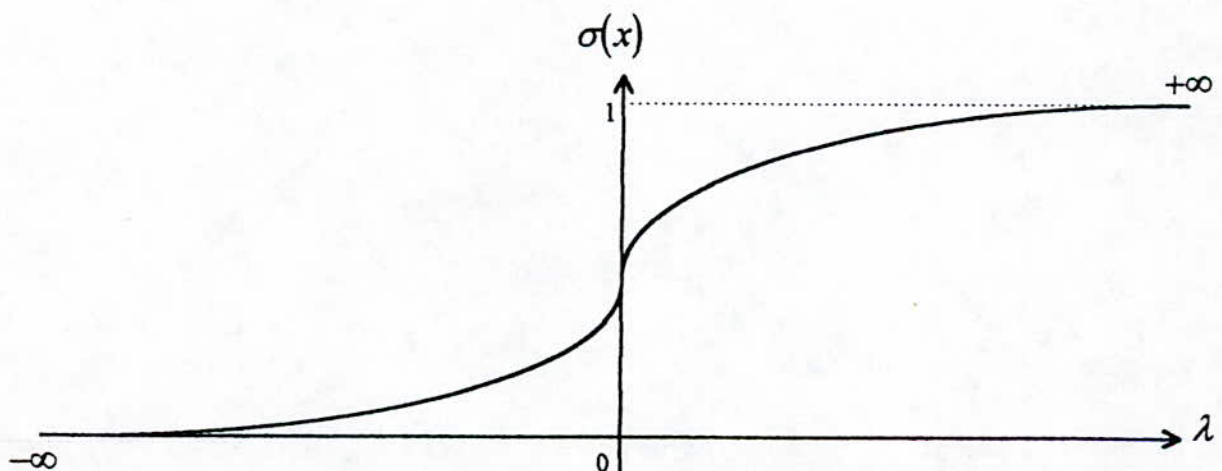


Fig. (2.2) La fonction d'activation $\sigma(x)$

2.2.2. LES CONNEXIONS :

D'une manière générale, l'architecture des réseaux de neurones peut aller d'une connectivité totale (tous les neurones sont reliés les uns aux autres), à une connectivité locale (ou tous les neurones ne sont reliés qu'à leurs plus proches voisins).

Mais, l'essentielle c'est que les réseaux sont partager en deux grandes catégories, selon la dépendance de l'évolution de ceux-ci en fonction explicite du temps.

2.2.3. LES RESEAUX DE NEURONES STATIQUES :

L'interconnexion de plusieurs neurones définit un champ de neurones, ce champ de neurones peut avoir plusieurs architectures, ceci dépend du choix de la dynamique du neurones utilisé comme élément de base.

Ces réseaux sont disposé en plusieurs champs de neurones qui ne possède pas des boucles synaptique fermé. Chaque neurone appartenant a un champ élémentaire appeler *couche* ne dépend que des neurones dans les couches qui se trouve en amont, dans le sens entrée-sortie. (Fig. 2.3)

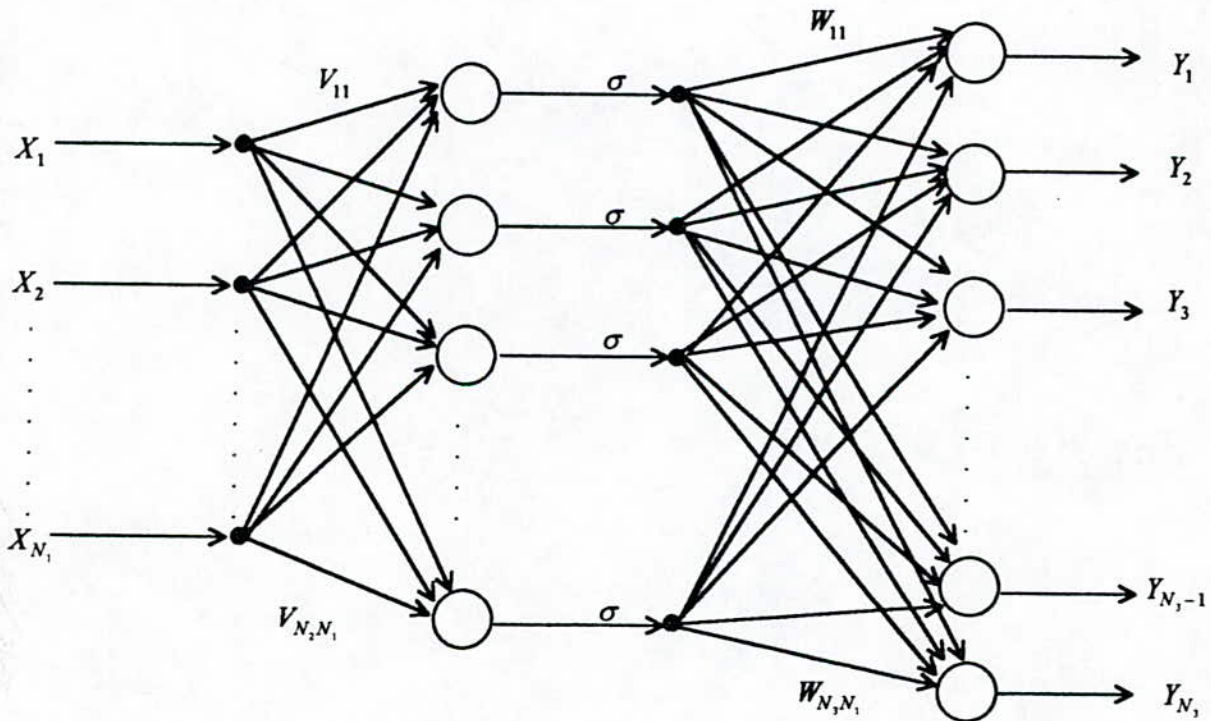


Fig. (2.3) Réseau statique à trois couches

On peut modéliser la sortie d'un réseaux de ce type comme suit:

Pour un réseau à trois couches on a :

$$y_i = \sum_{j=1}^{N_2} \left[w_{ij} \cdot \sigma \left[\sum_{k=1}^{N_1} v_{jk} \cdot x_k + \theta_{vj} \right] + \theta_{wi} \right]$$

pour $i=1, N_3$, avec :

- y_i : la i-ème sortie du réseau.
- $x \in \mathbb{R}^{N_1}$: vecteur d'entrée.
- $y \in \mathbb{R}^{N_3}$: vecteur de sortie.
- $\sigma(.)$: fonction d'activation.
- $[v_{jk}]$: les poids reliant la couche d'entrée avec la couche cachée.
- $[w_{ij}]$: les poids reliant la couche cachée avec la couche de sortie.
- N_i : nombre des neurones de la i-ème couche ($i=1,3$).
- θ_{vj}, θ_{wi} : les biais.

2.2.4. LES RESEAUX DE NEURONES DYNAMIQUE :

Les réseaux de neurones dynamiques ou récurrents sont différents des réseaux statiques, comme leurs nom l'indique, ils contiennent des rebouclage partiels ou totaux entres neurones. En général, la sortie de chaque neurone est réinjectée en entrée de chaque neurone, grâce a des poids variables, et il ne contient qu'une seule couche cachée (Fig. 2.4).

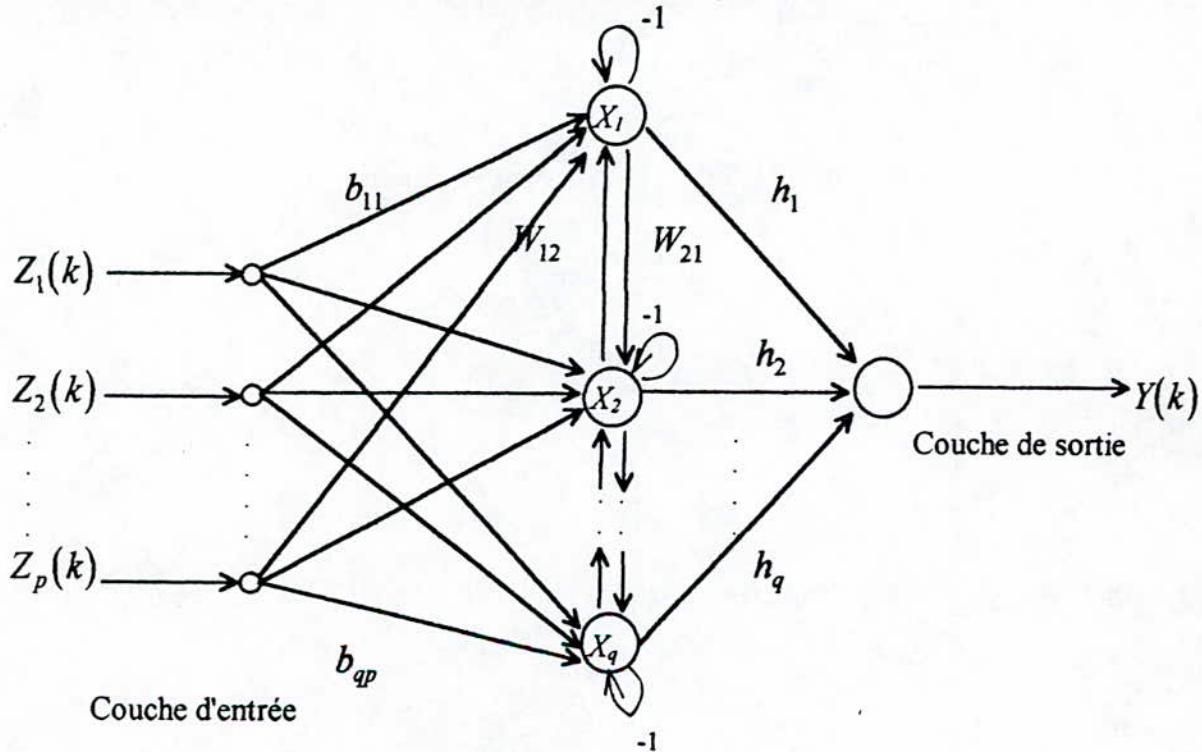


Fig. (2.4) Architecture du réseau dynamique

Ce type de réseau est régi par l'équation suivante:

$$\dot{x} = F(x, u, W)$$

$$y = G(x, W)$$

F : fonction qui représente la structure du réseau.

G : fonction qui représente la relation non linéaire.

remarque : F doit satisfaire les conditions de Lipschitz sur x .

2.3. L'APPRENTISSAGE :

Les procédures d'apprentissage peuvent se subdiviser en deux grandes catégories: apprentissage supervisé ou l'apprentissage non supervisé.

2.3.1. L'APPRENTISSAGE SUPERVISE :

Implique l'existence d'un *professeur* qui a pour rôle d'évaluer le succès (ou l'échec) du réseau quand il lui est présenté un stimulus connu (un stimulus est un exemple appartenant à la base d'apprentissage). Cette supervision consiste à renvoyer au réseau une information lui permettant de faire évaluer ses connexions, afin de faire diminuer son taux d'échec. L'information peut être explicite, sous forme d'une mesure de l'erreur commise par succession d'exemples, ou globalement sur l'ensemble des exemples de la base, et elle peut être implicite (apprentissage par renforcement), sous forme d'une simple appréciation (bon ou mauvaise) sans mesure d'erreur, et qui peut être globale sur l'ensemble des tâches que le réseau doit exécuter.

2.3.2. L'APPRENTISSAGE NON SUPERVISE :

Implique la fourniture à un réseau autonome d'une quantité suffisante d'exemples contenant des corrélations (de la redondance), Ces réseaux sont souvent appelés *auto-organiseurs*, ou à apprentissage compétitif.

Ces systèmes s'adaptent avec les changements qui ne sont pas appris a priori. Les deux architectures de base qui ont été développées sont:

1. feature Map de Kohonen.
2. ART (Adaline Resonance Theory).

2.3.3. L'APPRENTISSAGE DES RESEAUX DE NEURONES STATIQUES :

Un réseau de neurones statique associe un vecteur de sortie à chaque vecteur d'entrées. Pour un vecteur d'entrée x , le réseau lui associe un vecteur de sortie \hat{y} suivant la relation:

$$\hat{y} = \eta(x)$$

ou η est la fonction qui représente le RNA .

Le but de l'apprentissage supervisé est de minimiser l'erreur quadratique $e^2(t) = (y - \hat{y})^2$, la méthode utilisée pour cela est la « *backpropagation* ».

Cette méthode a été mise au point simultanément par deux équipes indépendantes en France (Fogelman-Soulié, Gallinari, Le Cun) et aux Etats-unis (Rumelhart, Hinton, Williams). L'idée de cet algorithme est de disposer d'un ensemble d'exemples sous forme de couples (entrées-sorties désirées).

A chaque étape, un exemple est présenté en entrée du réseau, une sortie réelle est générée. ce calcul est effectué de proche en proche de la couche d'entrée à la couche de sortie. Ensuite l'erreur $e_k^2(t) = \frac{1}{2} \sum_k (y_k - \hat{y}_k)^2$ (somme quadratique des erreurs sur chaque cellule de sortie) est calculée, cette erreur sera rétropropagée dans le réseau, donnant lieu à une modification de chaque poids. Cette étape sera répétée pour chaque exemple jusqu'à la minimisation de l'erreur quadratique commise sur l'ensemble des exemples.

FORMALISATION DE L'ALGORITHME DU BACK-PROPAGATION : [14][28]

La dérivation de l'erreur rétropropagée est liée par l'estimation du gradient de performance $\frac{\partial E}{\partial w_{pq}}$, suivant la définition de :

$$w_{pq}^{(k)} - w_{pq}^{(k-1)} = -\alpha \cdot \frac{\partial E}{\partial w_{pq}}$$

avec :

$$E = \sum_{i=1}^m \phi(y_i - \hat{y}_i) \text{ ou } \phi(y_i - \hat{y}_i) \text{ est le critère de performance.}$$

l'ajustement des poids de neurones de la couche de sortie est comme suit:

$$w_{pq}^{(k)} - w_{pq}^{(k-1)} = -\alpha \frac{\partial E}{\partial w_{pq}}$$

$$w_{pq}^{(k)} - w_{pq}^{(k-1)} = -\alpha \sum_{i=1}^m \frac{\partial \phi(e_i)}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial w_{pq}}$$

avec :

$$\hat{y}_i = \sigma(\bar{y}_i) = \sigma\left(\sum_{j=0}^{n_h} w_{ij} \cdot \hat{h}_{j,k}\right)$$

tel que :

- $[\hat{h}_{j,k}]$: sortie des neurones de la couche qui précède la couche de sortie à l'état k.
- $[y_i]$: la i-ème sortie du réseau de neurones.
- $[w_{pq}]$: vecteur des poids associe avec le p-ième neurones dans la couche de sortie.

Donc, par définition on a :

$$\frac{\partial \bar{y}_{i,k}}{\partial w_p} = \frac{\partial \sigma(\bar{y}_{i,k})}{\partial \bar{y}_{i,k}} \cdot \frac{\partial \bar{y}_{i,k}}{\partial w_p} = \sigma'(\bar{y}_{i,k}) \cdot \hat{h}_k \cdot \delta_{ip}$$

avec δ_{ip} symbole de kronecker.

D'ou l'équation d'ajustement des poids synaptiques w_{pq} sont obtenus comme suit :

$$w_{p,k} - w_{p,k-1} = \alpha \cdot \varepsilon_{p,k}^0 \cdot \hat{h}_k$$

avec

$$\varepsilon_{p,k}^0 = -\sigma'(\bar{y}_{p,k}) \cdot \frac{\partial \phi(e_{p,k})}{\partial \bar{y}_{p,k}}$$

et pour les poids des neurones appartenant à la r-ième couche (à partir de la couche de sortie vers l'entrée) sont ajusté comme suit :

$$v_{p,k}^{(r)} - v_{p,k-1}^{(r)} = \alpha \cdot \varepsilon_{p,k}^{(r)} \cdot \hat{h}_k^{(r+1)}$$

$$\varepsilon_{p,k}^{(r)} = (1 - \hat{h}_{p,k}^{(r)}) \sum_{l=1}^{n_{r-1}} \varepsilon_{l,k}^{(r-1)} \cdot v_{lp}^{(r-1)}$$

$$\forall r = 1, 2, \dots, L$$

avec :

$v_p^{(r)}$: vecteur des poids associes au p-ième neurones de r-ième couche interne.

$\hat{h}_p^{(r)}$: sortie de p-ième neurone de r-ième couche.

2.3.4 APPRENTISSAGE DES RESEAUX DE NEURONES DYNAMIQUES : [13]

L'architecture d'un réseau de neurones dynamique ne suffit pas pour prévoir son comportement dynamique, il faudrait aussi spécifier l'algorithme d'apprentissage [11]. Deux réseaux dynamiques qui sont entraînés avec différents algorithmes évoluant chacun d'une manière différente car un réseau de neurones dynamique est la combinaison de deux systèmes dynamiques, transmission et ajustement.

L'algorithme pour ces types de réseaux est caractérisé par deux techniques, la première, par l'imposition d'un point d'équilibre « *Fixed point learning* », la seconde est d'imposé au réseaux une certaine dynamique à travers l'apprentissage d'une trajectoire « *Trajectory learning* ».

L'algorithme « *Fixed point learning* » se résume comme suit:

Pour un réseau à trois couches régit par :

$$\begin{aligned} \dot{x} &= -x + W \cdot g(x) + B \cdot z(t) \\ y &= h^T \cdot x \end{aligned}$$

$$\begin{aligned} x &\in \mathfrak{R}^q \\ B &\in \mathfrak{R}^{q \times p} \\ W &\in \mathfrak{R}^{q \times q} \\ h &\in \mathfrak{R}^q \end{aligned} \quad \text{les poids associés avec les couches d'entrée, cachée et sortie respectivement.}$$

L'algorithme d'apprentissage peut être facilement déduit, en appliquant l'algorithme de descente du gradient à ce réseau, en effet:

$$\begin{aligned} h_i(k+1) &= h_i(k) - \mu_1 \cdot [y_d(k) - y(k)] \cdot x_i \quad i = \overline{1, q} \\ w_{ij}(k+1) &= w_{ij}(k) - \mu_2 \cdot h_i(k) \cdot [y_d(k) - y(k)] \cdot g_i(x_j) \quad i, j = \overline{1, q} \quad i \neq j \\ b_{ij}(k+1) &= b_{ij}(k) - \mu_3 \cdot h_i(k) \cdot [y_d(k) - y(k)] \cdot z_j(k) \quad i = \overline{1, q} \quad j = \overline{1, p} \end{aligned}$$

tel que:

μ_1, μ_2, μ_3 sont des pas d'adaptation choisis d'une façon appropriée.

x_i point d'équilibre de l'équation dynamique du réseau avec :

$$x_i^* = \sum_{j=1}^q w_{ij} \cdot g(x_j^*) + \sum_{l=1}^p b_{il} \cdot z_l(k)$$

z_i sont les entrées du réseau.

2.4. LES RESEAUX DE NEURONES ET LA COMMANDE : [11][19]

Avec ces propriétés remarquables, les réseaux deviennent des candidats naturels lors de l'identification et la commande des système dynamiques. Vus sous cet angle, les réseaux de neurones sont considérés comme étant des modélisateurs de processus. Ainsi, pour des exemples d'apprentissage suffisamment représentatives de la dynamique du modèle, alors tous ce que nous savons et ce que nous ne savons pas sur la dynamique du système est implicitement stocké dans le réseau. D'ou les propriétés des réseaux de neurones suivantes:

1. *systèmes non linéaires* : les réseaux de neurones peuvent approximer n'importe quelle fonction.
2. *le parallélisme* : les interconnexions de plusieurs neurones créent une relation distribuée et non linéaire entre les différentes couches, ceci augmente leurs robustesse.
3. *implémentation HARDWARE* : avec les développements de la nouvelle technologie des circuits VLSI (Very Large Scale Intégration), ont permis l'implémentation de différents algorithmes d'apprentissage des réseaux de neurones.
4. *apprentissage et généralisation* : un réseau bien entraîné peut généraliser des données non apprises, ceci définit le pouvoir d'extrapolation des RNA.
5. *systèmes multivariables* : les réseaux de neurones sont naturellement multivariables, ils sont directement applicables aux systèmes MIMO.

Avec ces propriétés, les réseaux de neurones sont souhaitables en identification et en commande des systèmes dynamiques.

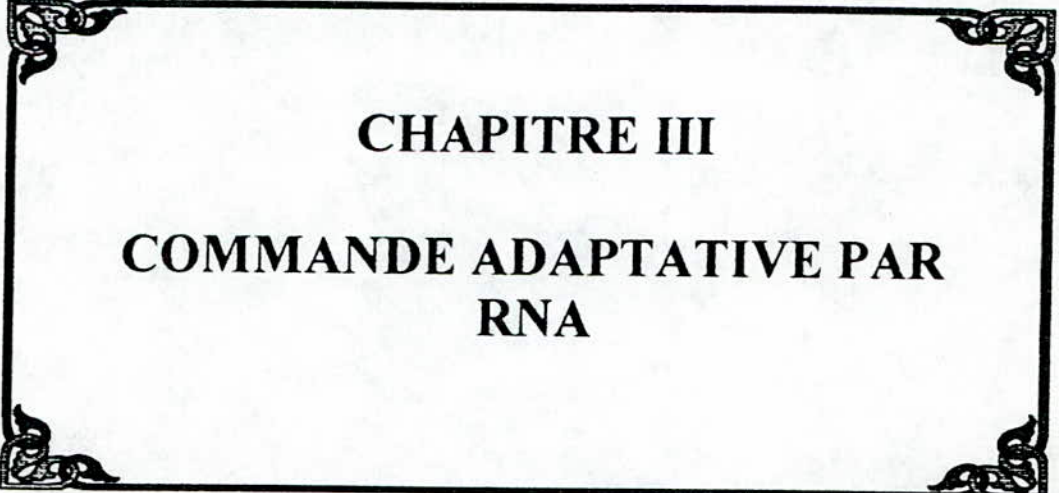
2.5. CONCLUSION :

Au cours de ce chapitre, nous avons présentés l'analogie entre le neurone biologique et le neurone artificiel, afin de permettre au lecteur d'acquérir le fondement philosophique des RNA. Ainsi, l'idée de base des RNA n'est autre que le circuit neuronique humain, d'où leurs capacités et leurs vaste champ d'application, traitement d'image et de la parole, identification des modèles complexes, reconnaissance des formes et de caractères ainsi que la commande des processus industriels.

Jusqu'à présent, les réseaux de neurones multicouches entraînés par rétropropagation sont les modèles les plus étudiés et utilisés. Cependant, il n'existe pas des lois prédéterminées pour un choix optimal de la taille et du nombre de neurones, mais reste que ces problèmes peuvent être résolus en pratique.

L'ensemble des algorithmes de type gradient ne permettent pas l'étude de la stabilité. Toutefois, cette notion est très importante en commande adaptative. En plus, la phase d'apprentissage est très lourde car elle nécessite un temps énorme et des moyens de calcul efficaces.

Ainsi, concevoir des mécanismes d'ajustement des RNA en temps réel issuent de la théorie de la stabilité serait une solution appropriée aux problèmes des méthodes d'entraînement conventionnelles.



CHAPITRE III

**COMMANDE ADAPTATIVE PAR
RNA**

3.1. INTRODUCTION :

La théorie des systèmes fournit des outils d'analyse et de synthèse parfaitement adaptés aux systèmes linéaires. Cependant, en pratique, les méthodes linéaires ne s'avèrent pas toujours applicables, parce qu'il n'est pas toujours possible de linéariser le système à commander. D'où la nécessité de concevoir de nouvelles méthodes de calcul qui doivent prendre en compte les caractéristiques particulières des systèmes non linéaires.

Depuis les premiers travaux jusqu'à présent, beaucoup d'études furent développées dans ce sens, notamment ceux obtenus dans l'approche adaptative et qui ont donné naissance à des applications impressionnantes dans divers domaines pratiques [1][10][16][27]. Cette approche offre la possibilité d'implémenter des algorithmes issus des développements théoriques et motivés par l'évolution technologique des calculateurs numériques.

L'utilisation des réseaux de neurone dans la commande des systèmes non linéaires peut être interpréter comme une évolution naturelle des techniques de commande. Cette évolution fondée sur plusieurs point:

1. Capacités limités des régulateurs classiques.
2. Analyse des non linéarités dures.
3. Absence d'information à priori sur le système à commander.

Les réseaux de neurones avec leurs énormes capacités d'approximation et de généralisation peuvent être utilisé efficacement dans la commande non linéaire. Parmi les méthodes de synthèse des correcteurs les plus connus, dont les réseaux de neurones artificiels ont trouvés leurs applications directe, on cite: *la commande inverse* [11][14][19], *la commande adaptative par modèle de référence (MRAC)* [22], *la commande auto-ajustable (STR)* [6], *la commande supervisée* [11][19] et *la commande par bouclage linéarisant (Feedback Linéarisation)* [23].

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 schéma de commande proposés, car il est difficile d'étudier la stabilité du système global composé notamment d'un ou de plusieurs réseaux de neurones.

A travers, ce chapitre, nous présenterons une nouvelle stratégie de commande qui combine entre la théorie de la stabilité de LYAPUNOV et les réseaux de neurones artificiels.

3.2. SCHEMA DE COMMANDE ADAPTATIVE PAR RNA SELON L'APPROCHE CENTRALISEE DE LEWIS [21]

Les capacités d'approximation et de généralisation des réseaux de neurones ont données de nouvelles issues aux problèmes de commande et d'identification des système non linéaires [14][22].

Toutefois, le traitement des RNA se heurte encore aujourd'hui à des difficultés résidant dans le passage obligatoire par un apprentissage off-line du bloc neuronal, l'incertitude sur

l'initialisation des paramètres du RNA ainsi que la difficulté d'établir la stabilité des schéma de commande [4][13]. 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 [18] appliquée en robotique et basé sur l'entraînement, en temps réel, des paramètres du RNA afin de reconstruire une certaine fonction couple non linéaire ainsi que l'association d'un proportionnelle-dérivateur à gain fixe.

Dans le présent chapitre, on présent un nouveaux algorithmes d'apprentissage on-line issus de la théorie de la stabilité de LYAPUNOV, garantissant ainsi la stabilité du système global et une poursuite parfaite comparable à une commande à caractère adaptative.

3.2.1. FORMULATION DU PROBLEME:

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

$$M(q) \cdot \frac{d^2 q}{dt^2} + V_m \left(q, \frac{dq}{dt} \right) \cdot \frac{dq}{dt} + H \left(\frac{dq}{dt} \right) + G(q) + T_d = T \quad (3.1)$$

q : Vecteur des variables articulaires généralisées (position).

$\frac{dq}{dt}$: Vecteur vitesse angulaire.

$M(q)$: Matrice d'inertie symétrique définit positive ($n \times n$).

$V_m \left(q, \frac{dq}{dt} \right)$: Matrice de centrifuge et de coriolis.

$H \left(\frac{dq}{dt} \right)$: Vecteur des forces des frottements visqueux.

$G(q)$: Vecteur des couples de la gravitation.

T_d : Vecteur regroupant les perturbations et les dynamiques non modélisées.

T : Vecteur des couples appliquer à chaque liaisons.

On définit comme suit:

- L'erreur de poursuite:

$$e(t) = q_d(t) - q(t) \quad (3.2)$$

- L'erreur de poursuite filtrée:

$$r(t) = \dot{e}(t) + \Lambda \cdot e(t) \quad (3.3)$$

Avec:

$q_d(t)$: Vecteur de trajectoire désirée .

$\Lambda \in IR^{n \times n}$: Matrice symétrique définit positive.

On dérivant l'erreur filtrée $r(t)$, on obtient:

$$\dot{r} = \ddot{e} + \Lambda \cdot \dot{e} = (\ddot{q}_d - \ddot{q}) + \Lambda \cdot \dot{e} \quad (3.4)$$

$$M(q) \cdot \dot{r} = M(q) \cdot \ddot{q}_d - M(q) \cdot \ddot{q} + M(q) \cdot \Lambda \cdot \dot{e} \quad (3.5)$$

d'après (3.1), on aura:

$$M(q) \cdot \dot{r} = -V_m(q, \dot{q}) \cdot r - T(t) + f(x) + T_d(t) \quad (3.6)$$

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

$$f(x) = M(q)(\ddot{q}_d + \Lambda \cdot \dot{e}) + V_m(q, \dot{q}) \cdot (\dot{q}_d + \Lambda \cdot e) + G(q) + H(\dot{q}) \quad (3.7)$$

Ou:

$$x = \left[e^T \quad \dot{e}^T \quad q_d^T \quad \dot{q}_d^T \quad \ddot{q}_d^T \right]^T$$

Le problème de la commande adaptative neuronale se pose comme suit. Pour un bras manipulateur décrit par (3.1), déterminer les lois d'ajustement des poids du RNA qui permettent d'estimer, en temps réel, la fonction couple non-linéaire définit dans (3.6) ainsi que le couple adéquat $T(t)$ tel que:

$$\lim_{t \rightarrow \infty} |e(t)| = \lim_{t \rightarrow \infty} |q(t) - q_d(t)| = 0$$

i.e. : L'erreur de poursuite converge asymptotiquement vers zéro.

3.2.2. STRUCTURE DU CONTROLEUR ADAPTATIF NEURONALE:

3.2.2.1. Le Réseau du neurone :

Le réseau utilisé ici est un réseau de neurones statique à une seule couche cachée (Fig 3.1) soit:

$X \in IR^n$: vecteurs d'entrée.

$Y \in IR^m$: vecteurs de sortie.

L : nombre du neurone de couche cachée.

La sortie du réseau est la suivante:

$$y_i = \sum_{j=1}^L \left[W_{ij} \cdot \sigma \left[\sum_{K=1}^n V_{jK} X_K + \theta_{Vj} \right] + \theta_{Wi} \right] \quad i=1, \dots, m \quad (3.8)$$

Avec:

- $\sigma(\cdot)$: la fonction d'activation (sigmoïde).
- $[V]^T \in IR^{L \times n}$: Matrice des poids (entrée couche cachée).
- $[W]^T \in IR^{m \times L}$: Matrice des poids (couche cachée sortie).
- $\theta_{Vj} \quad \theta_{Wj}$: les Biais .

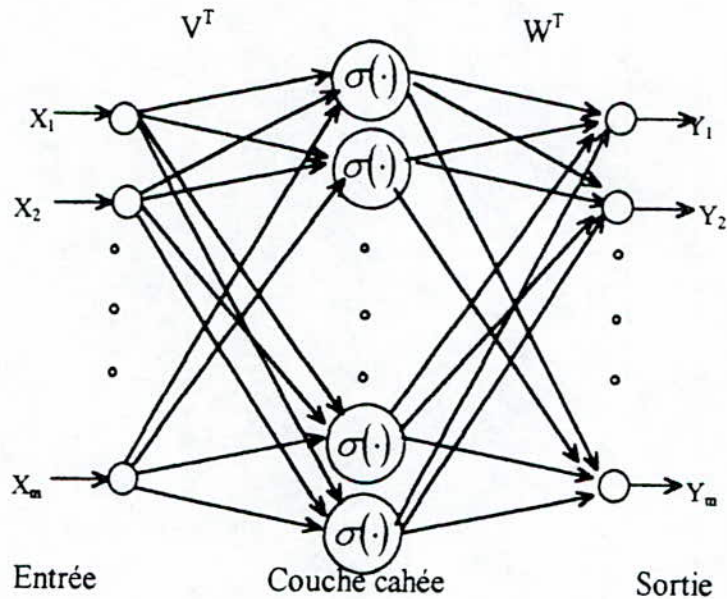


Fig. (3.1) schéma du réseau de neurone

L'expression matricielle (3.8) est:

$$Y = W^T \cdot \sigma(V^T \cdot X) \quad (3.9)$$

Avec:

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

Dans ce schéma de commande, le rôle attribué au réseau de neurones est d'estimer en temps réel la fonction couple non-linéaire $f(x)$ défini par (3.6). Donc, on assumera l'existence des poids idéaux W et V , qui permettent la reconstruction parfaite de la fonction $f(x)$. Soit:

$$f(x) = W^T \cdot \sigma(V^T \cdot x) + \varepsilon(x) \quad (3.10)$$

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

$$x = \begin{bmatrix} e^T & \dot{e}^T & q_d^T & \dot{q}_d^T & \ddot{q}_d^T \end{bmatrix}.$$

On définit l'estimée fonctionnelle de $f(x)$ notée $\hat{f}(x)$, tel que :

$$\hat{f}(x) = \hat{W}^T \cdot \sigma(\hat{V}^T \cdot x) \tag{3.11}$$

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

On définit comme suit :

- l'erreur de l'estimation fonctionnelle:

$$\tilde{f}(x) = f(x) - \hat{f}(x) \tag{3.12}$$

- l'erreur d'estimation des poids du RNA:

$$\begin{cases} \tilde{V} = V - \hat{V} \\ \tilde{W} = W - \hat{W} \\ \tilde{Z} = Z - \hat{Z} \end{cases} \tag{3.13}$$

avec $Z^T = \begin{bmatrix} W^T & V^T \end{bmatrix}$.

- l'erreur de sortie de la couche cachée :

$$\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(V^T \cdot x) - \sigma(\hat{V}^T \cdot x) \tag{3.14}$$

Le développement en série de Taylor de la fonction d'activation (la sigmoïde) donne:

$$\sigma(V^T \cdot x) = \sigma(\hat{V}^T \cdot x) + \sigma'(\hat{V}^T \cdot x) \cdot \tilde{V}^T \cdot x + O(\tilde{V}^T \cdot x)^2 \tag{3.15}$$

Ou

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

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

$$\tilde{\sigma} = \sigma'(\hat{V}^T \cdot x) \cdot \tilde{V}^T \cdot x + O(\tilde{V}^T \cdot x)^2 = \hat{\sigma}' \cdot \tilde{V}^T \cdot x + O(\tilde{V}^T \cdot x)^2 \tag{3.16}$$

avec $O(\tilde{V}^T \cdot x)^2 = [\sigma(V \cdot x) - \sigma(\tilde{V}^T \cdot x)] - \sigma'(\hat{V}^T \cdot x) \cdot \tilde{V}^T \cdot x \tag{3.17}$

3.2.2.2. Etude Mathématique Préliminaire: [18]

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:

$$\begin{cases} \|V\|_F \leq V_m \\ \|W\|_F \leq W_m \\ \|Z\|_F \leq Z_m \end{cases} \quad (3.18)$$

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

Définition :

Soit A , B et X des matrices tel que :

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

La norme frobenienne est définit par:

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

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

$$\|A \cdot X\|_F \leq \|A\|_F \cdot \|X\|$$

Hypothèse 2 :

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

$$\{\|q_d\|, \|\dot{q}_d\|, \|\ddot{q}_d\|\} \leq Q_d \quad (3.19)$$

Propriété 1 :

Le vecteur $x(t)$ donné par (3.7) vérifie à chaque instant l'équation suivante:

$$\|X\| \leq c_1 \cdot Q_d + c_2 \cdot \|r\| \quad (3.20)$$

c_1, c_2 : constante positive.

Propriété 2 :

Pour la fonction sigmoïde, l'erreur du 2^{ième} ordre dans le développement en série de Taylor défini dans (3.17) vérifie l'équation suivante :

$$\|O(\tilde{V}^T \cdot x)\|^2 \leq c_3 + c_4 \cdot Q_d \cdot \|\tilde{V}\|_F + c_5 \cdot \|\tilde{V}\|_F \cdot \|r\| \quad (3.21)$$

c_3, c_4 et c_5 des constantes positives calculables.

Propriété 3 :

Pour un bras manipulateur défini par (1.1), on a:

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

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

m_1, m_2 : Constante positive.

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

3. La matrice $\dot{M}(q, \dot{q}) - 2 \cdot V_m(q, \dot{q})$ est skew symétrique.

4. Le couple représentant les perturbations T_d est borné par:

$$\|r_d\| < b_d$$

b_d : Constante positive.

3.2.2.3. Schéma de Commande : [18]

La structure de base du contrôleur adaptatif neuronal est montrée dans la figure (3.2). l'action du couple commande est la suivante:

$$T(t) = \hat{f}(x) + K_v \cdot r(t) - v(t) \quad (3.22)$$

avec:

- $K_v \in \mathcal{R}^{n \times n}$: matrice gain symétrique définie positive;
- $v(t) \in \mathcal{R}^n$: couple additionnel robust.

Le premier terme correspond à la sortie du RNA (3.11) et considérée comme l'estimation de la fonction défini dans (3.7). Le second correspond à l'équation classique d'un proportionnelle - dérivateur à gain fixe.

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

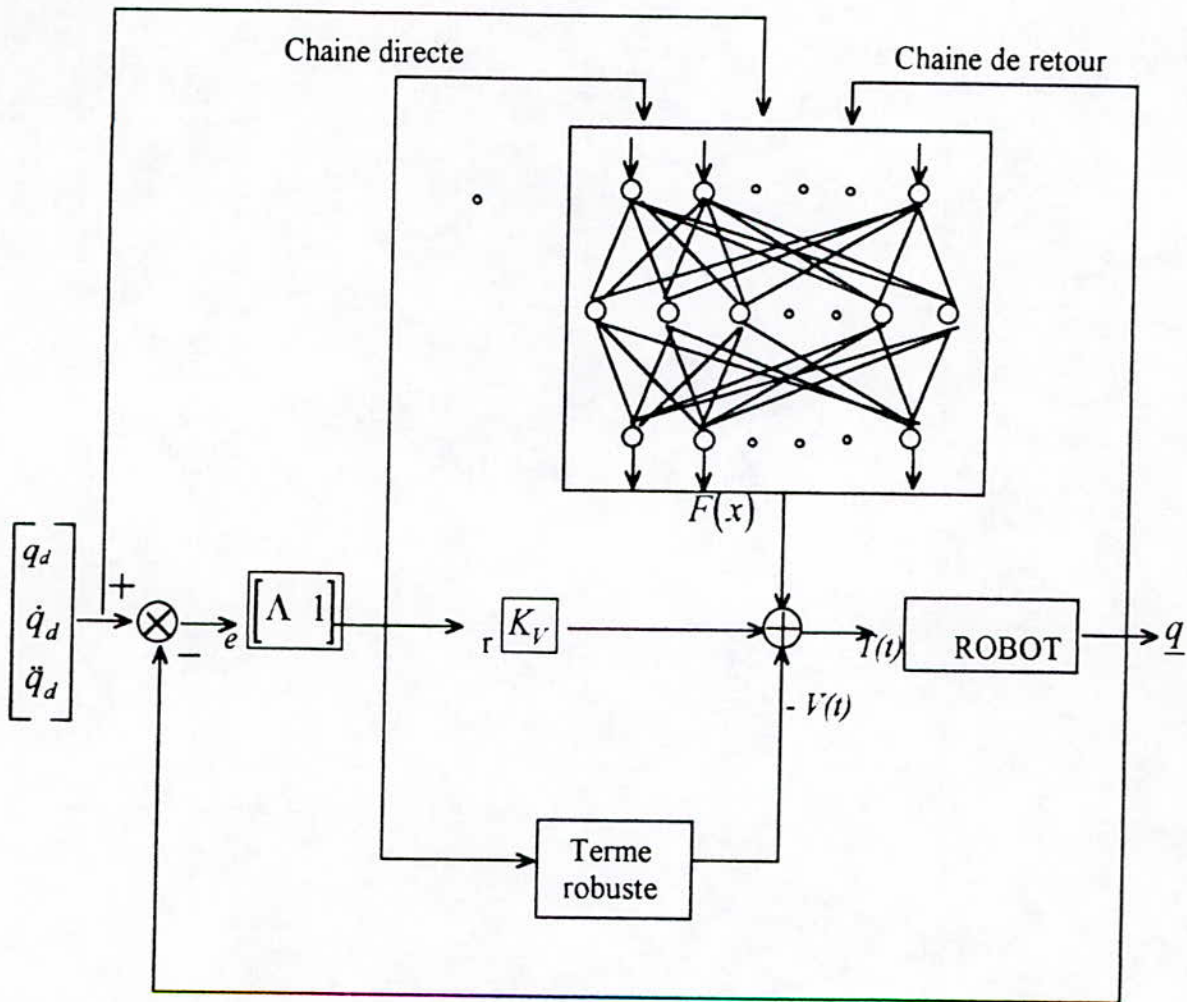


Fig. (3.2) structure de base du système adaptatif neuronal proposé par LEWIS

• Etude de la dynamique de l'erreur filtrée :

On introduisons ce contrôleur dans l'équation (3.7), on obtient la dynamique de l'erreur filtrée tel que:

$$M(q) \cdot \dot{r} = -(K_v + V_m(q, \dot{q})) \cdot r + f(x) - \hat{f}(x) + T_d + v(t)$$

En remplaçons (3.10), (3.11) on obtient:

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

d'ou

$$M(q) \cdot \dot{r} = -(K_v + V_m(q, \dot{q})) \cdot r + \tilde{W}^T \cdot \hat{\sigma} + \hat{W}^T \cdot \tilde{\sigma} + \tilde{W}^T \cdot \tilde{\sigma} + (\varepsilon + T_d) + v(t) \quad (3.23)$$

avec \tilde{W}, \tilde{V} et $\tilde{\sigma}$ définis par (3.13) et (3.14). et d'après (3.16):

$$M(q) \cdot \dot{r} = -(K_v + V_m(q, \dot{q})) \cdot r + \tilde{W}^T \cdot \hat{\sigma} + \hat{W}^T \cdot \hat{\sigma}' \cdot \tilde{V}^T \cdot x + w(t) + v(t) \quad (3.24)$$

avec:

$$w(t) = \tilde{W}^T \cdot \hat{\sigma}' \cdot \tilde{V}^T \cdot x + W^T \cdot O(\tilde{V}^T \cdot x)^2 + (\varepsilon + T_d) \quad (3.25)$$

Le terme (3.25) regroupe les perturbations ainsi que les erreurs d'estimation infiniment petits.

3.2.2.4. Les algorithmes d'adaptation : [18]

Pour pouvoir estimer fidèlement la fonction non-linéaire définie dans (3.5), il faut concevoir des algorithmes d'adaptation des paramètres du RNA qui garantissent la stabilité du système en boucle fermée.

L'ajustement se faisant en temps réel, le problème majeur des autres approches développées jusqu'ici (CHAPITRE 2) étant l'apprentissage au préalable de la fonction à estimer ainsi que le choix de l'initialisation. F.L. LEWIS [18] propose des algorithmes d'adaptation on-line qui exigent pas d'apprentissage de la fonction ainsi qu'une initialisation à zéro de tous les paramètres du RNA.

Par la suite, nous développerons deux algorithmes d'apprentissage on-line, le premier conçu pour le cas idéal, le second étant plus général.

a. Cas Idéal :

Nous appellerons idéal, le cas où:

1. L'erreur de reconstruction fonctionnelle $\varepsilon(x)$ est nulle.
2. Les perturbations ainsi que les erreurs de modélisation sont nulles.

Théorème 3-1 : [18]

si dans l'équation (3.24) $w(t) = 0$ $v(t) = 0$, alors les lois d'ajustements des poids du RNA sont:

$$\dot{\hat{W}}(t) = F \cdot \sigma(\hat{V}^T \cdot x) \cdot r^T \quad (3.36a)$$

$$\dot{\hat{V}}(t) = G \cdot x \cdot [(\sigma'(\hat{V}^T \cdot x))^T \cdot \hat{W} \cdot r]^T \quad (3.26b)$$

avec: F, G matrice constante symétrique définie positive.

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.

• **Preuve De La Stabilité :**

Soit la fonction positive candidate de LYAPUNOV suivante:

$$L(t) = \frac{1}{2} r^T M r + \frac{1}{2} \text{tr}(\tilde{W}^T F^{-1} \tilde{W}) + \frac{1}{2} \text{tr}(\tilde{V}^T G^{-1} \tilde{V}) \quad (3.27)$$

en dérivons (3.27), on obtient:

$$\dot{L}(t) = r^T \dot{M} r + \frac{1}{2} r^T \dot{M} r + \text{tr}(\tilde{W}^T F^{-1} \dot{\tilde{W}}) + \text{tr}(\tilde{V}^T G^{-1} \dot{\tilde{V}})$$

en remplaçons (3.24), on aura:

$$\dot{L}(t) = -r^T K_v r + \frac{1}{2} r^T (\dot{M} - 2V_m) r + \text{tr}(\tilde{W}^T (F^{-1} \dot{\tilde{W}} + \hat{\sigma} \cdot r^T)) + \text{tr}(\tilde{V}^T (G^{-1} \dot{\tilde{V}} + x \cdot r^T \hat{W}^T \hat{\sigma}')) \quad (3.28)$$

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

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

d'après (3.13), on aura: $\begin{cases} \dot{\tilde{W}} = -\hat{W} \\ \dot{\tilde{V}} = -\hat{V} \end{cases}$, en remplaçons (3.26a) et (3.26b) dans (3.28), on a:

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

ce qui constitue une garantie de la stabilité au sens de LAYPUNOV du schéma de commande proposé, ainsi, l'adaptation définit dans (3.26) assure la convergence de tout les signaux vers un équilibre asymptotiquement stable.

b. Cas Général :

On présence de perturbation, des dynamiques non modélisées et d'erreur de reconstruction fonctionnelle, il est nécessaire de modifié les lois d'adaptation et d'additionné un couple robuste a fin d'assurer les performances souhaités.

Théorème 3-2 : [18]

si dans la loi de commande définit dans (3.22) et :

$$v(t) = -K_z \cdot (\|\hat{Z}\|_F + Z_m) \cdot r \quad (3.29)$$

avec: $K_z > c_2$ (c_2 définit dans (3.20)).

$$\dot{\hat{W}} = F \cdot \hat{\sigma} \cdot r - F \cdot \hat{\sigma}' \cdot \hat{V}^T \cdot x \cdot r^T - k \cdot F \cdot \|r\| \cdot \hat{W} \quad (3.30a)$$

$$\dot{\hat{V}} = G \cdot x \cdot (\hat{\sigma}' \cdot \hat{W} \cdot r)^T - k \cdot G \cdot \|r\| \cdot \hat{V} \quad (3.30b)$$

avec: $F=F^T > 0$, $G=G^T > 0$, $k > 0$.

Alors, l'erreur filtrée $r(t)$ ainsi que les poids estimés \hat{W} et \hat{V} seront asymptotiquement stable, et :

$$\lim_{t \rightarrow \infty} |r(t)| = 0$$

• Preuve de la stabilité :

La démonstration de ce théorème ne diffère pas beaucoup de la précédente, puisque c'est la même fonction candidate de LYAPUNOV. Ainsi, nous renvoyons tout lecteurs veulent consulter cette démonstration vers la référence [18].

Remarque :

_ L'algorithme d'apprentissage on-line proposé par LEWIS [18] dans (3.26a) et (3.26b) nous est familier, ce n'est autre que le backpropagation standard développé par Werbos [28], et illustré dans le *chapitre 3*. La différence majeure réside dans le signal de propagation utilisé ici, à savoir l'introduction de l'erreur filtrée (pondérée) dans l'ajustement des poids du RNA, ce qui apporte à notre avis un supplément d'information sur la poursuite de la trajectoire désirée (position et vitesse).

_ L'algorithme d'adaptation des paramètres du RNA donné par (3.30a) et (3.30b) comprend:

1. Un premier terme identique à l'algorithme obtenu dans le cas idéal.
2. Un second terme introduit uniquement dans (3.30a), qui correspond à une correction apportée sur le backpropagation standard due au second facteur dans le développement en série de Taylor de la fonction d'activation.
3. Le dernier terme est bien connu en commande adaptative robuste, il correspond à la modification apportée sur les paramètres de commande introduite par Narendra [21], appelle *e-modification*, afin d'améliorer la robustesse d'un schéma de commande adaptative et d'éviter une éventuelle divergence en présence de perturbation et des dynamiques non-modélisées (*drift parameters*).

_ En pratique, il est à noter qu'un assez grand choix du gain K_v , à pour effet de diminuer l'erreur de poursuite. Cependant, en augmentant ce gain un comportement oscillatoire, voir instable, est à craindre.

3.3. SIMULATION :

Nous avons procédé à l'application des schéma de commande proposés dans ce mémoire sur le bras de robot PUMA 560 par simulation sur un ordinateur de type PC munis d'un processeur INTEL-PENTIUM 100 Mhz, et en utilisant les deux langages de programmation FORTRAN et

MATLAB 4.0 sous l'environnement graphique WINDOWS 3.11, le logiciel GRAFFER sous windows est utilisé pour la réalisation des graphes. La méthode de Rung-Kutta du 4ème ordre avec un pas de 0.001 seconde à été utilisé pour résoudre les équation différentielles .

Pour faciliter la compréhension et pouvoir établir une comparaison entre les résultats des différents schémas de commande, on procède, pour chaque schéma, à des simulations identiques.

Afin d'évaluer les performances de chaque schéma de commande, on effectue les quatre tests suivants:

TEST 1 : Test de validation

On impose une trajectoire allant de l'état initial $[0^\circ, 0^\circ, 0^\circ]$ vers l'état final $[60^\circ, -60^\circ, 60^\circ]$ en 3 secondes, sans charge (à vide). On procède selon cette trajectoire avec et sans le signal émit par le réseau de neurones pour mettre en évidence son rôle.

TEST 2 : Variation massique subite

On impose une trajectoire similaire à celle du premier test, en déplaçant une charge d'une masse ($m= 2 \text{ Kg}$, $m= 4 \text{ Kg}$) qui sera lâchée en milieu de parcours à l'instant $t = 1.5$ seconde.

TEST 3 : Rupture de la commande

Sans charge et selon la même trajectoire que le premier test, on procède à l'interruption d'une seule commande parmi les trois successivement afin d'établir les limites de la commande centralisée.

TEST 4 : Test de LEAHVY [17]

Proposé par LEAHVY, ce test permet d'exciter toute la dynamique du PUMA 560 afin d'évaluer la robustesse de n'importe quel schéma de commande. Sans charge, il impose une trajectoires allant de $[-50^\circ, -135^\circ, 135^\circ]$ à $[45^\circ, -85^\circ, 30^\circ]$ en 1.5 seconde.

REMARQUE :

Les trajectoires imposées sont d'une nature cycloïdale définit dans le chapitre I. Nous recommandons de ce référer à ce diagramme car il sera respecté pour toutes les simulations présenter par la suite.

PARAMETRES DE COMMANDE :

Le réseau de neurones comporte: 15 entrées, 10 neurones dans la couche cachée et 3 sorties.

$$K_v = \text{Diag}(20, 20, 20), \quad \Lambda = \text{Diag}(15, 15.5, 15), \quad F = 0.8, \quad G = 0.8, \quad k = 0.001, \quad K_z = 1, \quad Z_m = 10$$

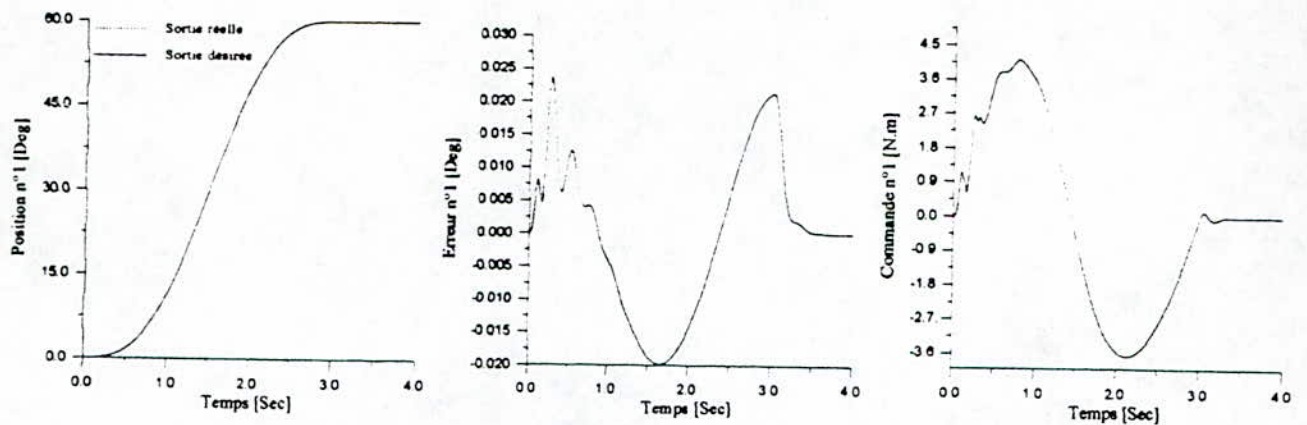
INTERPRETATION DES RESULTATS :

- **Test de validation :** La figure (3.3) montre que l'erreur de poursuite s'annule rapidement après des légers pics d'erreurs. La figure(3.4) montre que le signal de sortie du RNA est prépondérant dans le schéma de commande, ce qui confirme son rôle de compensation des couples de coriolis et de gravitation.
- **Variation massique subite :** Les figures (3.5) et (3.6) nous permettent de constater la stabilité du système vis-à-vis des variations subite des paramètres du modèle introduites par l'effet de la charge.
- **Rupture de la commande :** Les figures(3.7), (3.8) et (3.9) illustrent la nettes détérioration des performances suite à la rupture d'une commande. Ceci est du, sans doute, au caractère centralise de la commande. Ainsi, une instabilité dans une articulation affecte l'ensemble du système. Nous pensons que l'introduction de *e-modification* à permit d'éviter la dérive des paramètres du RNA, mais induit en contre partie à une commande qui n'évoluent plus dans les normes.
- **Test de LEAHVY :** La figure (3.10) permet de constater la robustesse du schéma de commande face à des trajectoires très rapide. Toutefois, on remarque des pics d'erreurs relativement grand par rapport à la trajectoire normale.

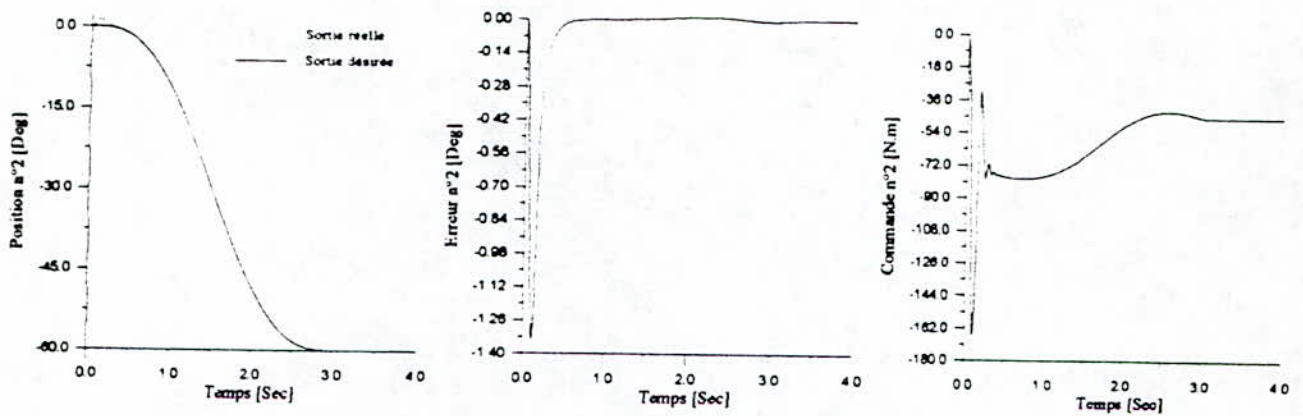
3.4. CONCLUSION :

Dans le présent chapitre, nous avons présenté une nouvelle stratégie de commande adaptative par RNA, basée sur l'ajustement en temps réel des paramètres du réseau suivant des algorithmes d'adaptation issuent de la théorie de la stabilité de LYAPUNOV.

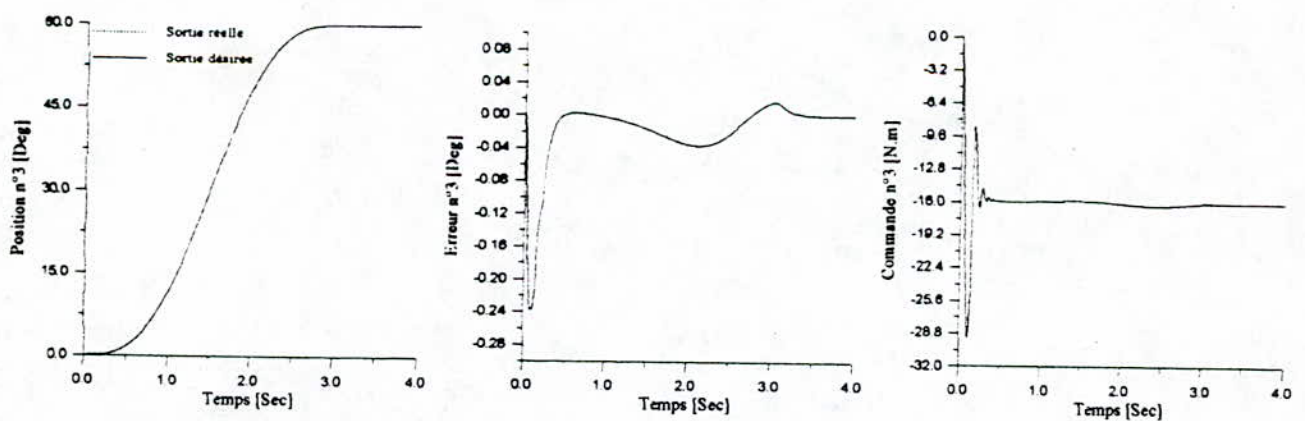
Ce schéma de commande centralisée à été appliqué sur le PUMA 560 pour mettre en évidence les performances et la robustesse de cette technique. Ainsi, en simulation, nous constatons une grande différence par rapport aux méthodes de commande conventionnelles par RNA, de point de vue souplesse, simplicité et rapidité de convergence. Cependant, nous avons pu soulever à travers les différents tests effectués, les limites de la commande centralisée.



Position, Erreur et Commande de la première articulation.

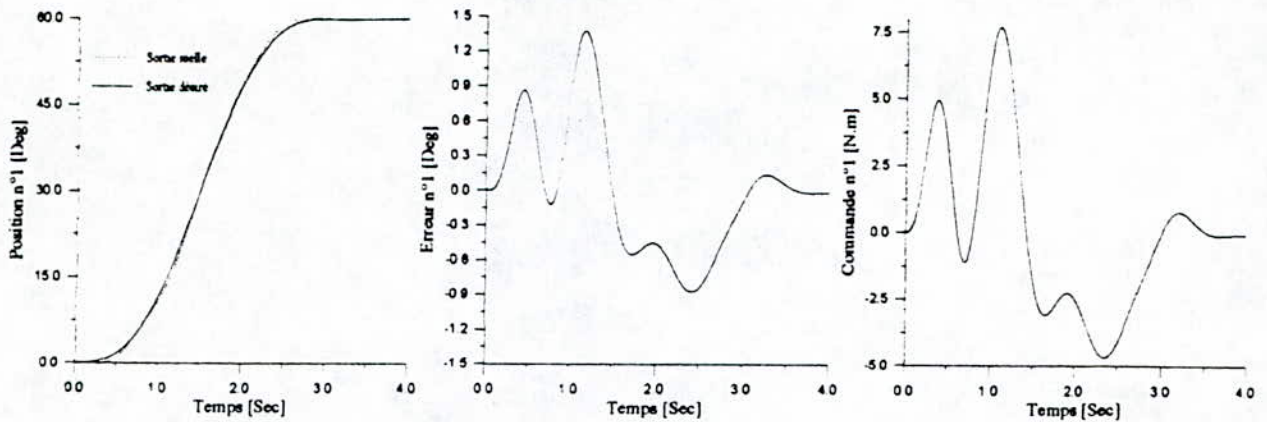


Position, Erreur et Commande de la deuxième articulation.

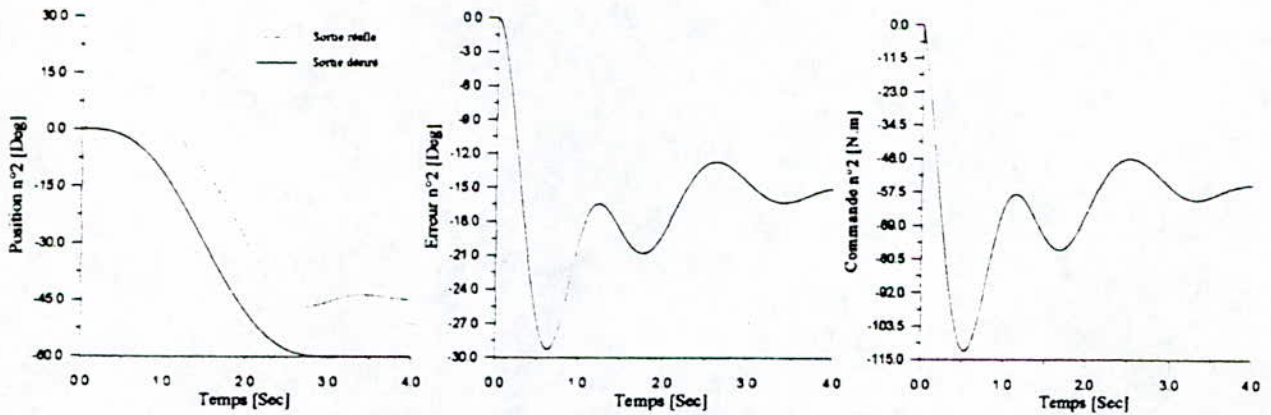


Position, Erreur et Commande de la troisième articulation.

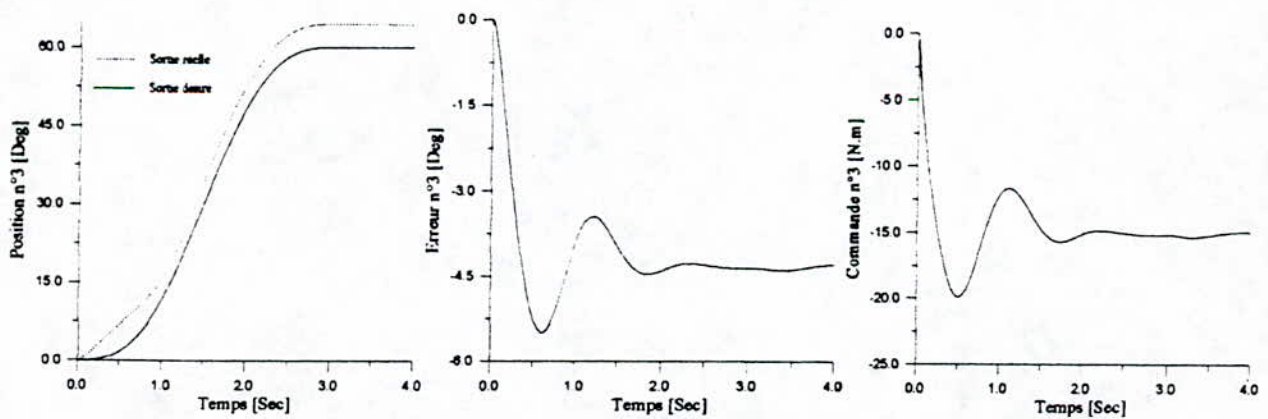
Figure 3.3
Réponses en boucle fermée sans charge (à vide).



Position, Erreur et Commande de la première articulation.

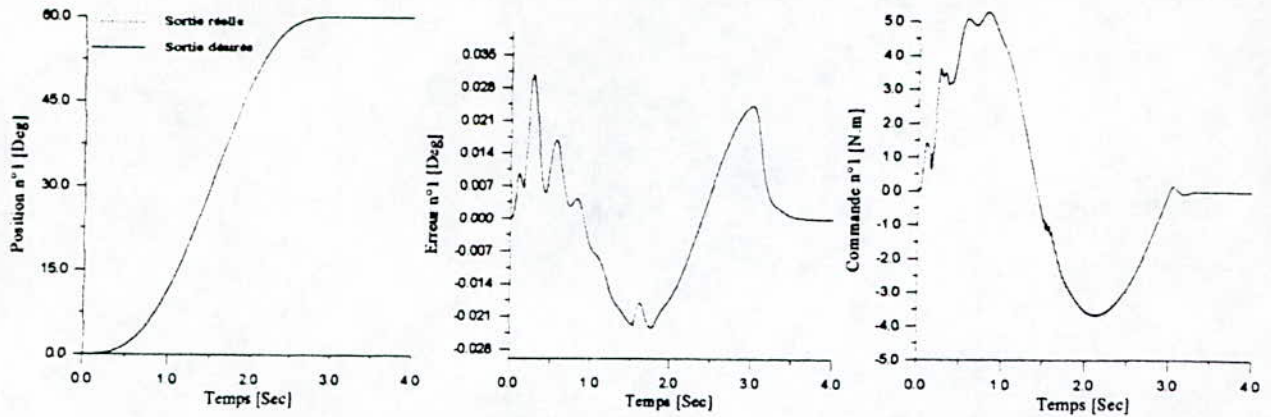


Position, Erreur et Commande de la deuxième articulation.

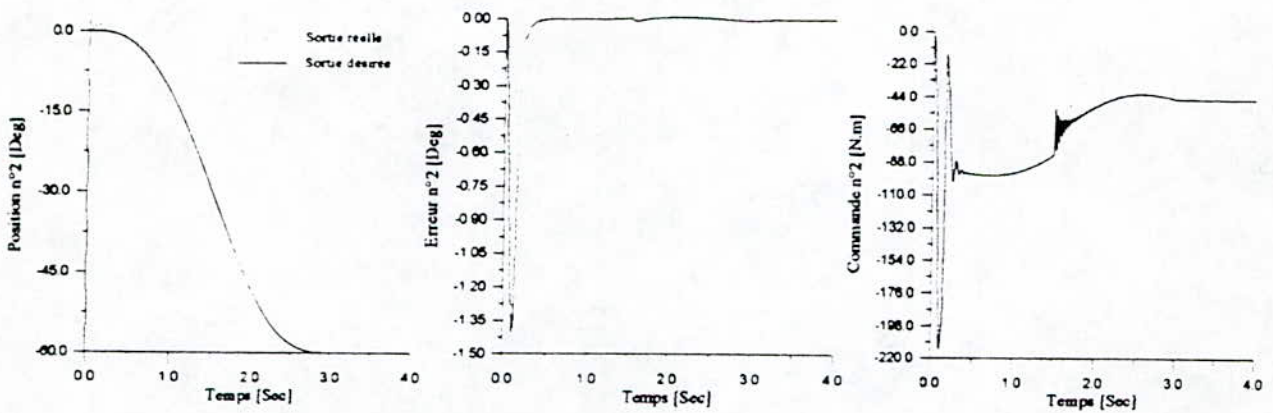


Position, Erreur et Commande de la troisième articulation.

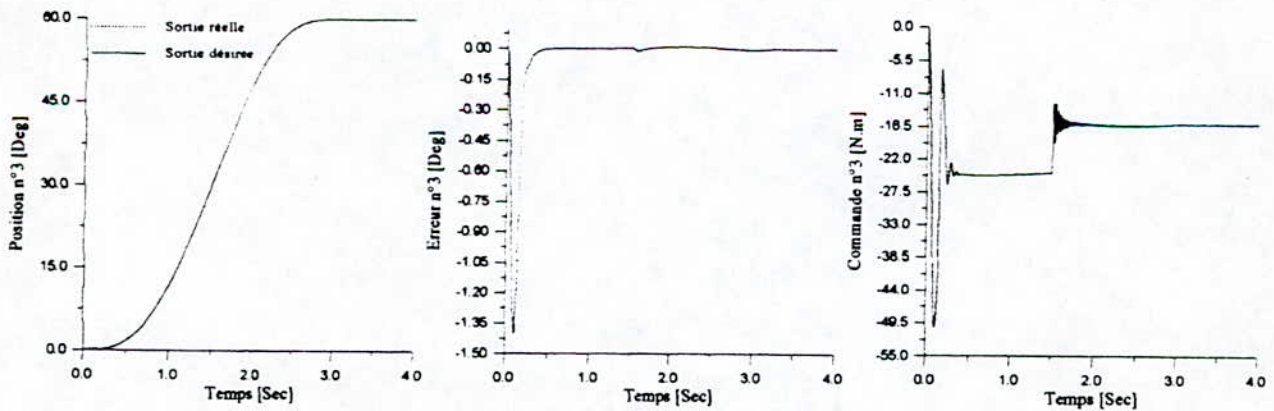
Figure 3.4
Réponses en boucle fermée avec le P.D. fixe (à vide).



Position, Erreur et Commande de la première articulation.

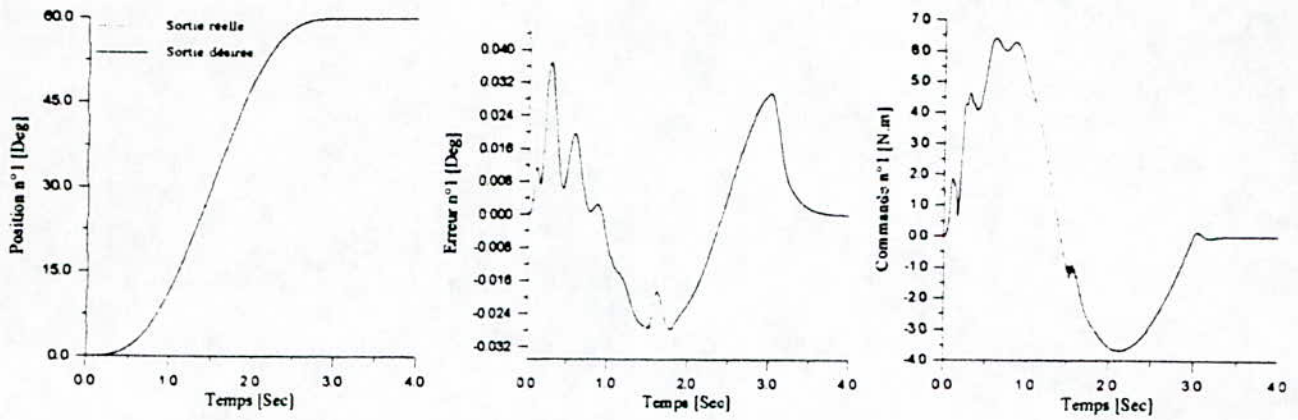


Position, Erreur et Commande de la deuxième articulation.

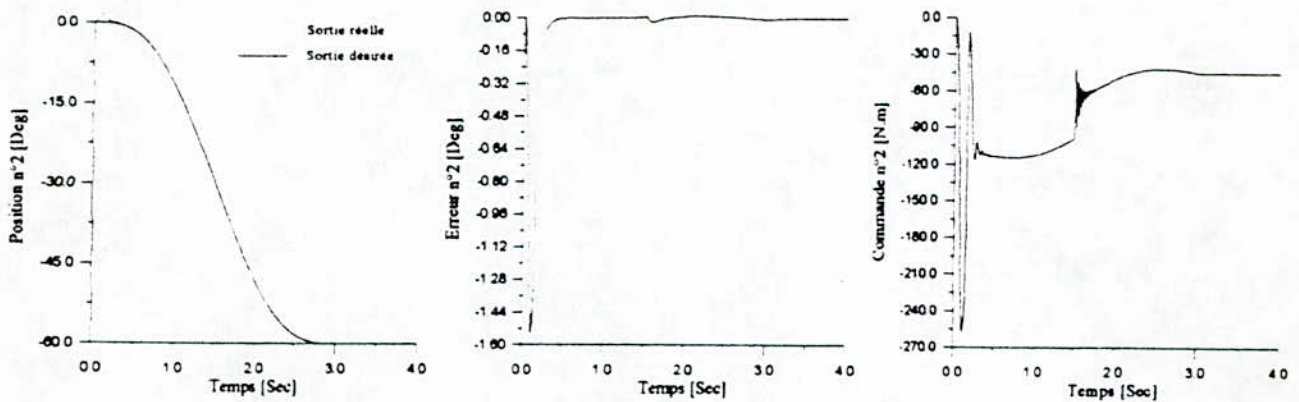


Position, Erreur et Commande de la troisième articulation.

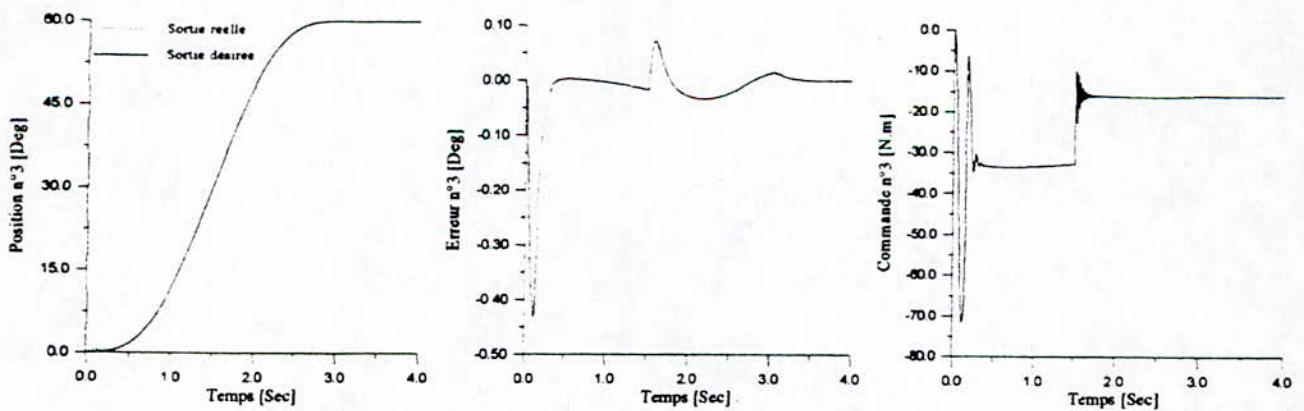
Figure 3.5
Réponses en boucle fermée avec une charge de 2Kg (lâchée à l'instant $t=1.5s$).



Position, Erreur et Commande de la première articulation.

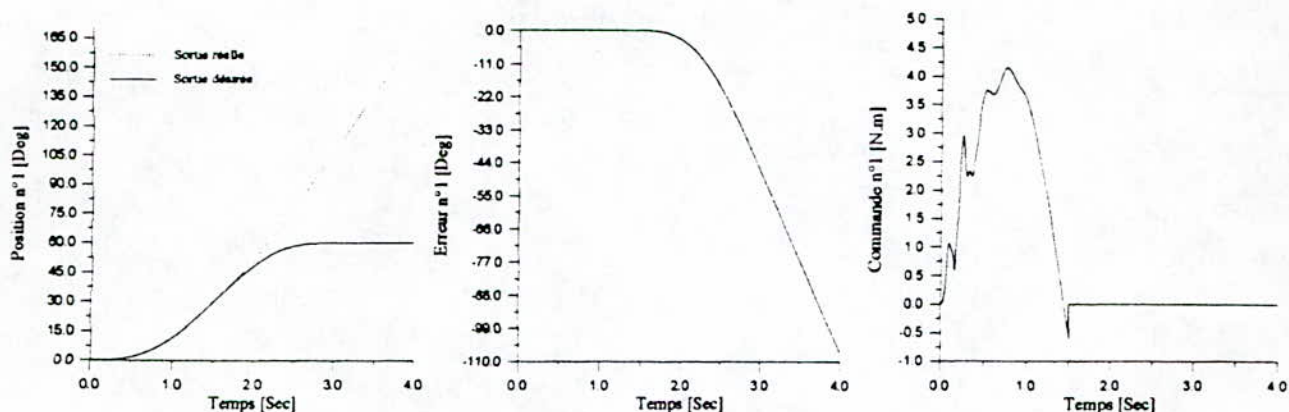


Position, Erreur et Commande de la deuxième articulation.

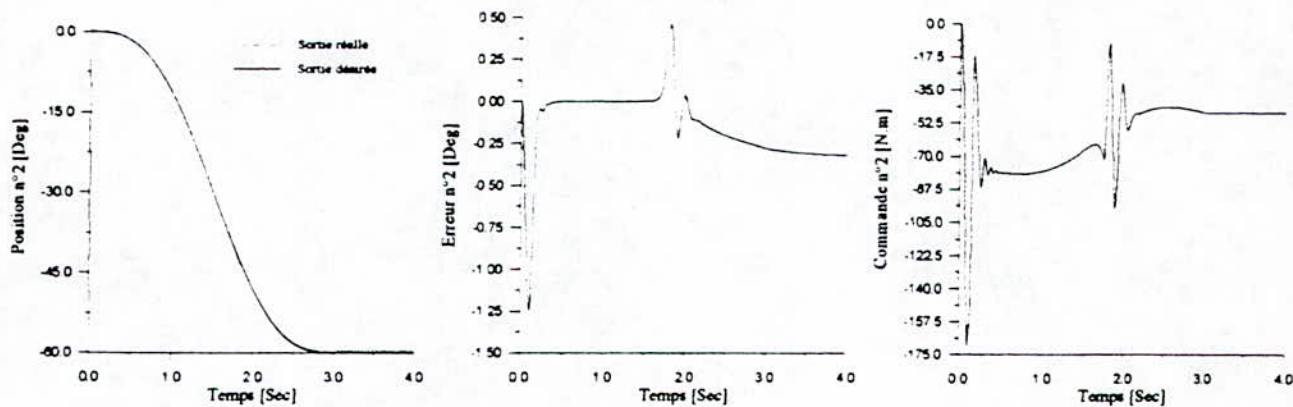


Position, Erreur et Commande de la troisième articulation.

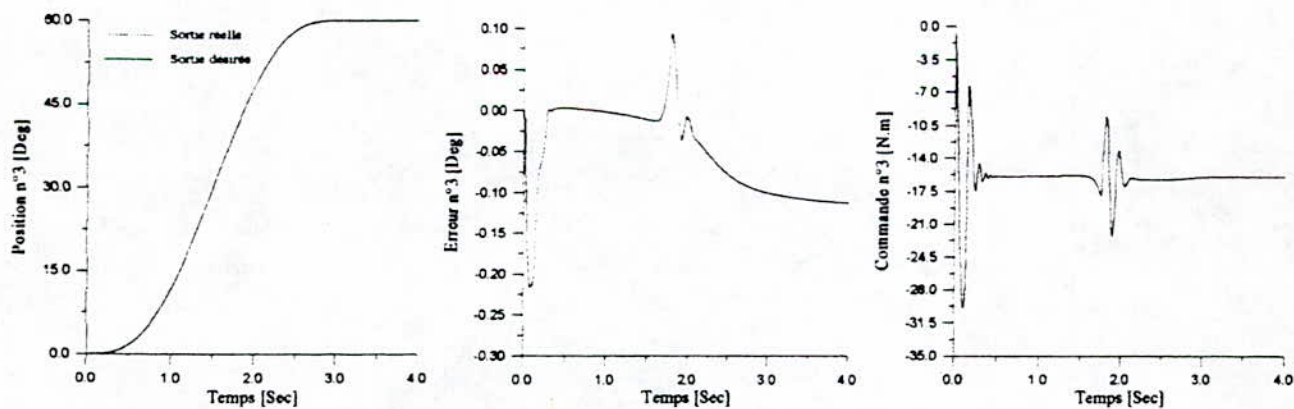
Figure 3.6
Réponses en boucle fermée avec une charge de 4Kg (lâchée à l'instant $t=1.5s$).



Position, Erreur et Commande de la première articulation.

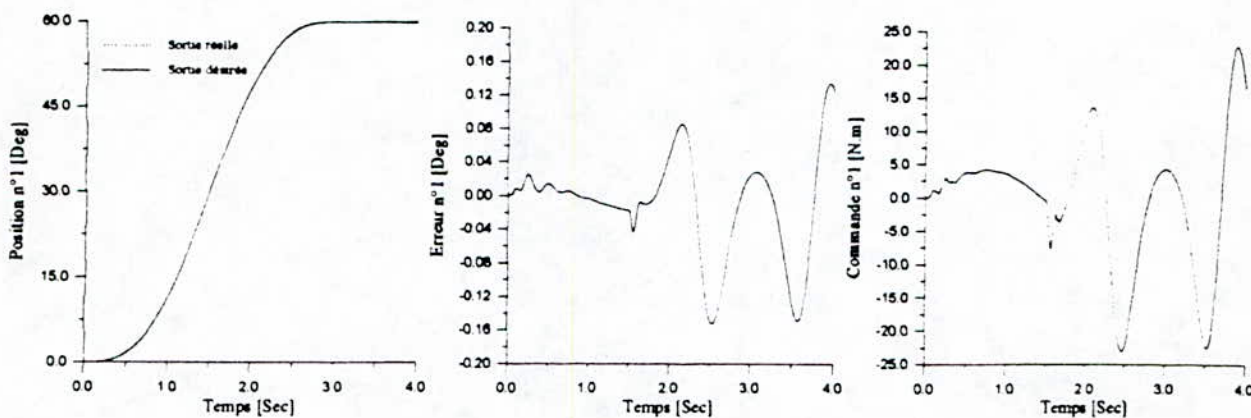


Position, Erreur et Commande de la deuxième articulation.

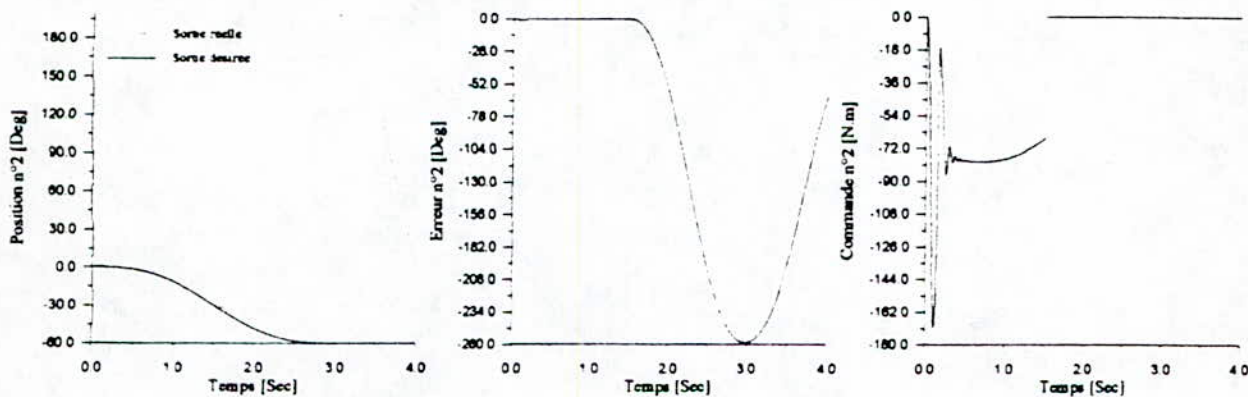


Position, Erreur et Commande de la troisième articulation.

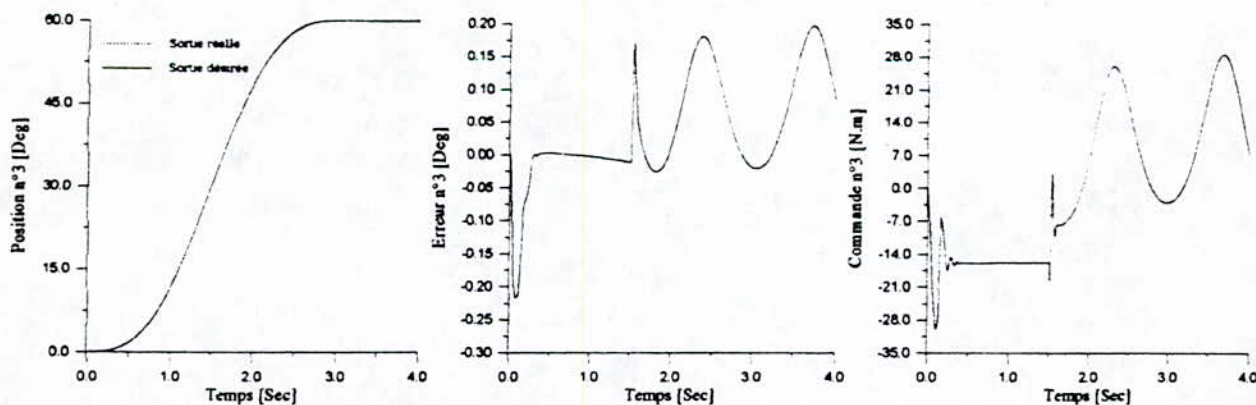
Figure 3.7
Réponses en boucle fermée avec rupture de la commande, $u(1)=0$ à $t=1.5s$.



Position, Erreur et Commande de la première articulation.



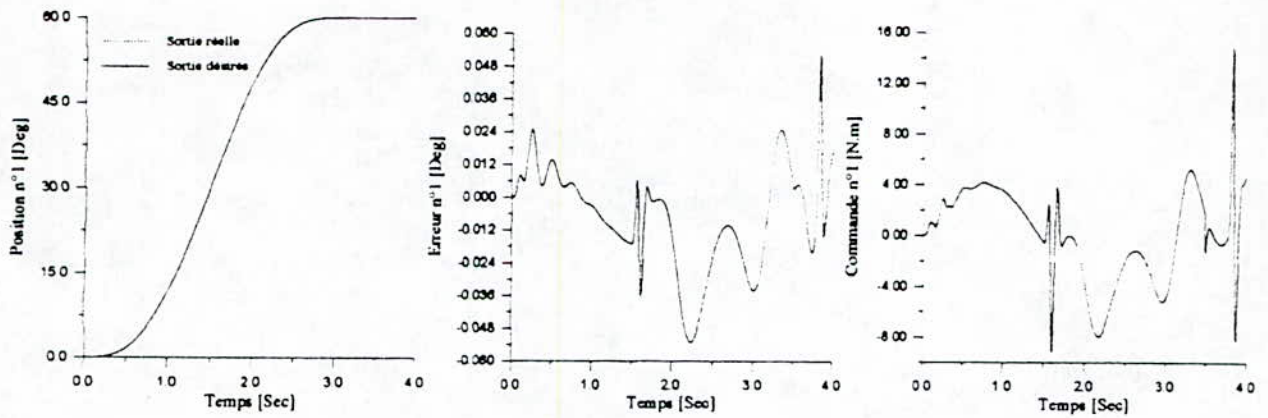
Position, Erreur et Commande de la deuxième articulation.



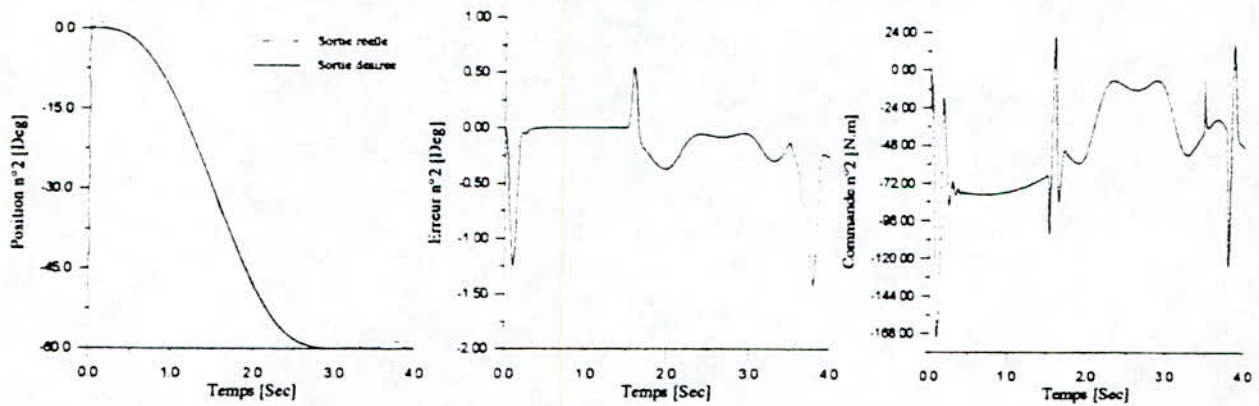
Position, Erreur et Commande de la troisième articulation.

Figure 3.8

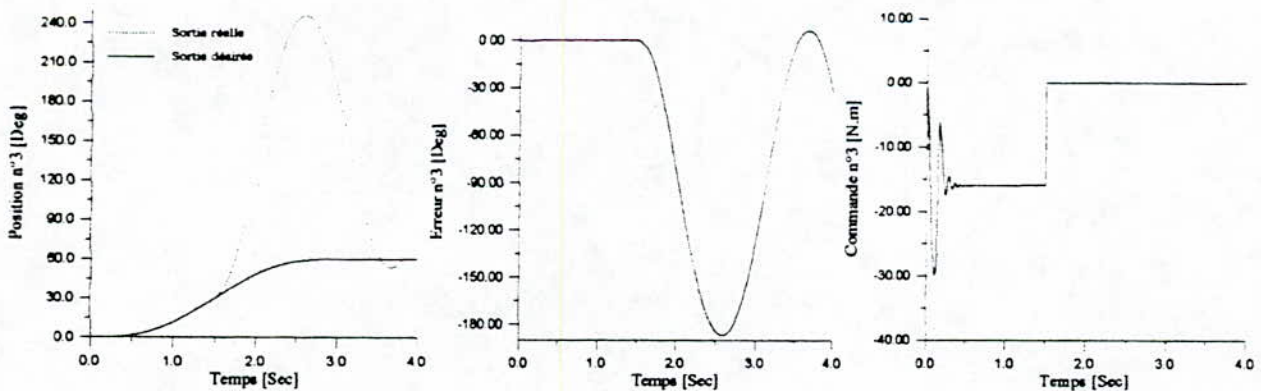
Réponses en boucle fermée avec rupture de la commande, $u(2)=0$ à $t=1.5s$.



Position, Erreur et Commande de la première articulation.

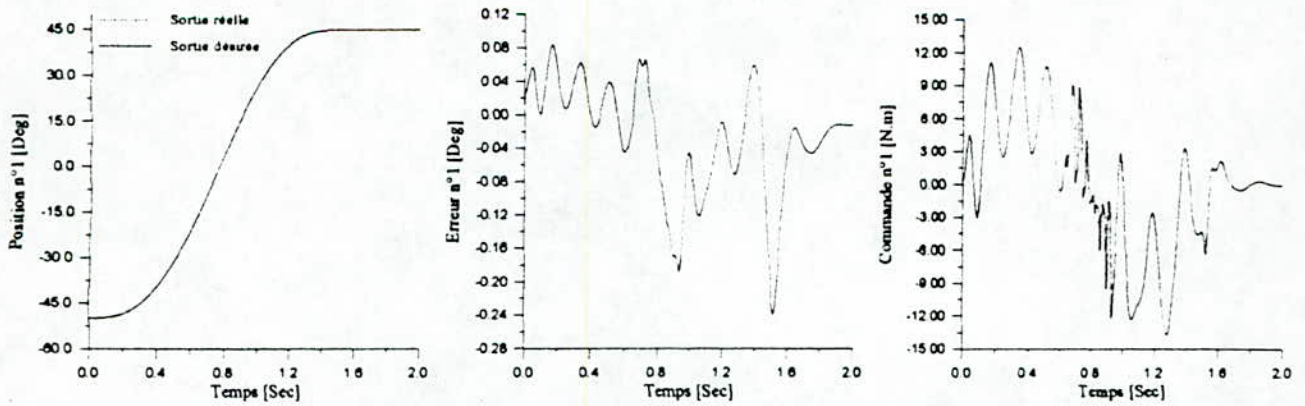


Position, Erreur et Commande de la deuxième articulation.

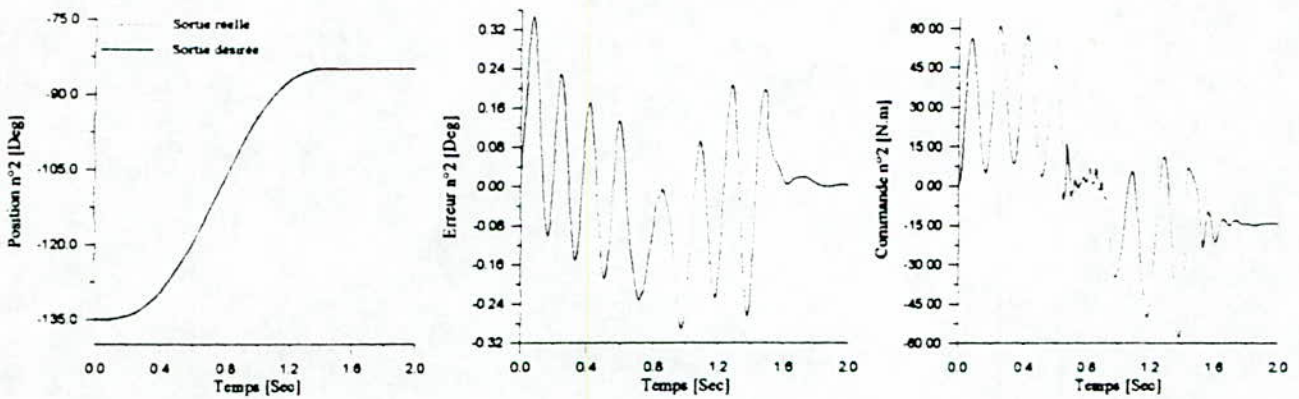


Position, Erreur et Commande de la troisième articulation.

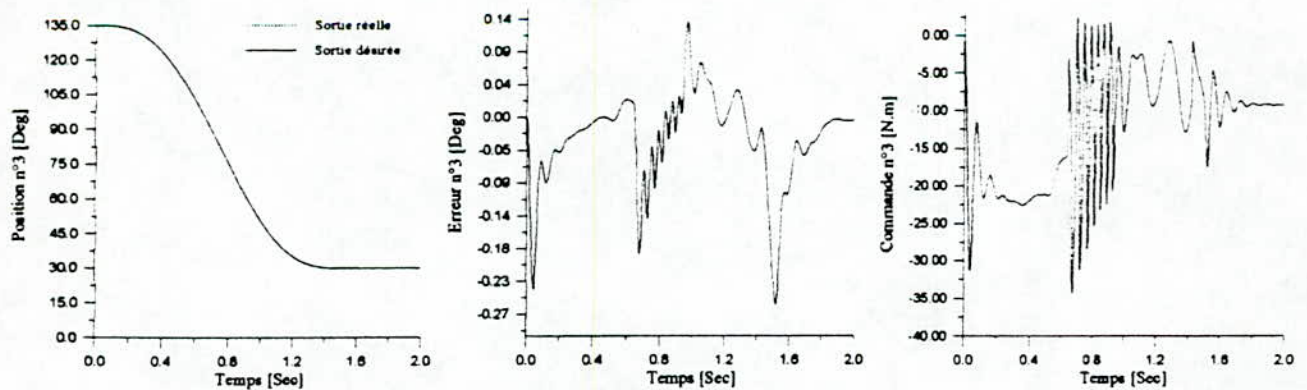
Figure 3.9
Réponses en boucle fermée avec rupture de la commande, $u(3)=0$ à $t=1.5s$.



Position, Erreur et Commande de la première articulation.

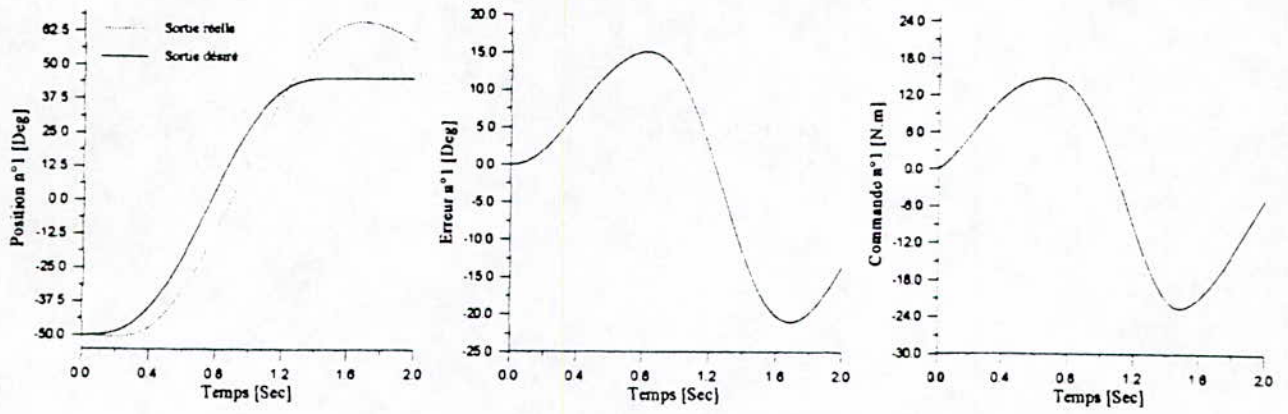


Position, Erreur et Commande de la deuxième articulation.

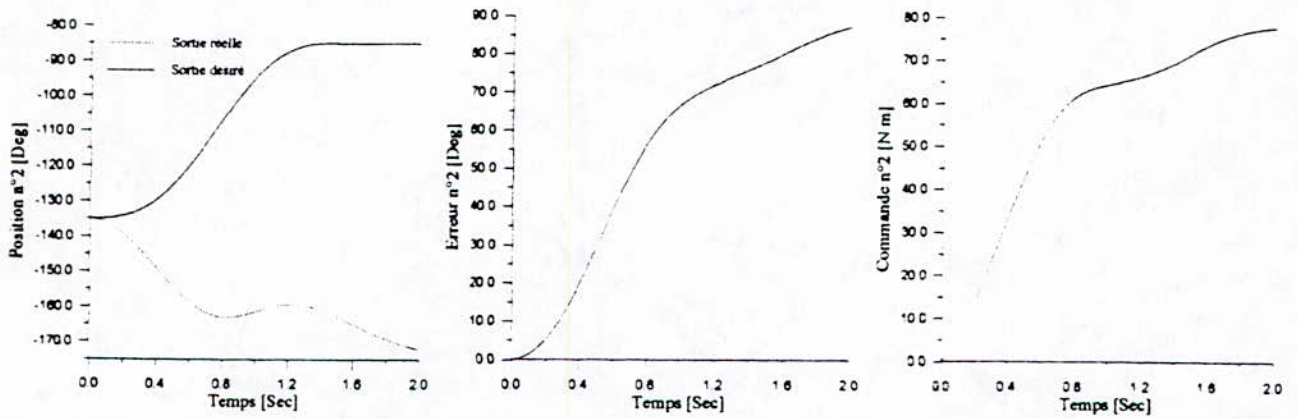


Position, Erreur et Commande de la troisième articulation.

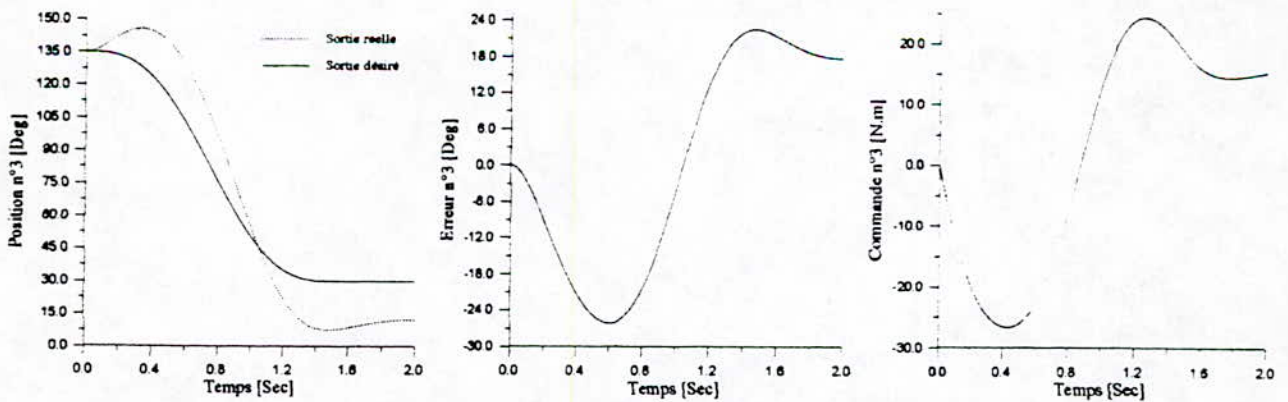
Figure 3.10
Réponses en boucle fermée sous le test de Leavhy.



Position, Erreur et Commande de la première articulation.



Position, Erreur et Commande de la deuxième articulation.



Position, Erreur et Commande de la troisième articulation.

Figure 3.11
Réponses en boucle fermées sous le test de LEAHVY avec un PD fixe.



CHAPITRE IV

**COMMANDE ADAPTATIVE
DECENTRALISEE PAR RNA**

INTRODUCTION

Dans un milieu industriel, la majorité des tâches présent en charge par des robots sont délicates et exigent une très grande précision sous des trajectoires rapides. Les méthodes de synthèse des régulateurs conventionnelles s'avèrent donc inadéquates faces à la complexité et à l'ignorance de certaines dynamiques du robot.

En vue d'améliorer les performances des manipulateurs, des recherches avancées ont permis d'émerger des nouvelles techniques de commande notamment ceux obtenus dans l'approche adaptative [1][10][16], et qui ont conduit à des résultats intéressants en robotique [3][25][27], mais qui ont soulevés divers problèmes qui n'avaient pas été envisagés dans les premiers travaux tel que la nécessité, en générale, d'assez puissants moyens de calcul du à la complexité et à la taille du modèle et aux liaisons entre les différentes grandeurs du processus, ce qui rend ce traitement assez lent et parfois même impossible pour les robots, étant donnés les fortes non-linéarités et les interconnexions qui régissent leurs dynamiques.

Pour y remédier à ces inconvénients, la commande décentralisée ou ce qu'on appelle l'*indépendant joint control* est de plus en plus envisageable. Pour celle-ci, un système est aperçu comme étant composé de plusieurs sous-systèmes interconnectés, pour chacun d'entre eux, une loi de commande est générée en utilisant uniquement les informations disponibles localement. Ce qui facilite énormément la commande en temps réel, car il n'y aura pas d'échange d'informations entre les différents sous-systèmes, ainsi l'implémentation, en pratique, sur des calculateurs à multiprocesseurs est possible.

L'ensemble des travaux effectués dans ce sens sont principalement focalisés sur la commande décentralisée à caractère adaptative [5][9][12][26]. Cependant, avec le développement de la théorie des réseaux de neurones artificiels (RNA) et les possibilités offertes en matière d'identification et de commande des systèmes dynamiques [7][14][15][19][22][23] permettent d'envisager l'introduction des RNA dans la commande décentralisée. Les travaux consacrés à la synthèse des schémas de commande adaptative décentralisée par RNA [4][13], quoique pas nombreux, ont montrés les performances de cette approche.

Dans ce chapitre, on présentera deux méthodes de commande adaptative décentralisée par RNA, illustrées à travers les deux parties suivantes:

- La première, concerne la synthèse d'une commande adaptative neuronale semblable à la commande proposé par LEWIS [18] et étendue à la structure décentralisée.
- La seconde, consacrée à l'étude et la conception d'une nouvelle technique de commande adaptative décentralisée par RNA.

PARTIE 1 : SYNTHÈSE D'UNE COMMANDE ADAPTATIVE DECENTRALISEE PAR RNA

4.1.1. INTRODUCTION

La commande décentralisée part du principe qu'un système dynamique fortement couplé est mieux étudié, lorsque les sous systèmes qui le composent sont traités localement. [12]

Autrement dit, pour un système composé de plusieurs sous systèmes interconnectés, concevoir une commande à structure décentralisée revient à désigner pour chaque sous système, un contrôleur local qui n'a accès qu'aux informations disponibles au niveau de sous système, en considérant les perturbations ainsi que les interconnexions avec les autres sous systèmes comme un vecteur de perturbation pour le sous système en question.

Dans cette partie, nous proposons d'étendre le schéma adaptatif par RNA de LEWIS [18] au cas décentralisé, pour cela, on attribut à chaque articulation du bras manipulateur un régulateur adaptatif par RNA semblable à celui proposé dans [18]. Ce qui revient à terme, de constituer plusieurs stations de commande neuronale, chaque station s'occupera de la commande d'une seule articulation en utilisant les informations disponibles localement.

4.1.2. DESCRIPTION DU SCHEMA DE COMMANDE ADAPTATIF DECENTRALISEE PAR RNA.

Le schéma adaptatif neuronal qu'on utilisera ici, et toutes les considérations et propriétés sont identiques à ceux énoncés au chapitre III [18].

Considérons un bras manipulateur à n degrés de liberté décrit par l'équation (Chap. 3 Eq. (3.1)):

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + H(\dot{q}) + G(q) + T_d(t) = T(t) \quad (4.1.1)$$

La dynamique de la $i^{\text{ème}}$ articulation peut être représentée par:

$$m_{ii}\ddot{q}_i + V_{m_i}(q, \dot{q})\dot{q}_i + \sum_{\substack{j=1 \\ j \neq i}}^n m_{ij}\ddot{q}_j + \sum_{\substack{j=1 \\ j \neq i}}^n V_{m_j}(q, \dot{q})\dot{q}_j + h_i(\dot{q}) + g_i(q) + T_{d_i}(t) = T_i(t) \quad (4.1.2)$$

Soit:

$$m_{ii}\ddot{q}_i + V_{m_i}(q, \dot{q})\dot{q}_i + d_i(q, \dot{q}, \ddot{q}) = T_i(t) \quad (4.1.3)$$

Avec:

$$d_i(q, \dot{q}, \ddot{q}) = \sum_{\substack{j=1 \\ j \neq i}}^n m_{ij} \ddot{q}_j + \sum_{\substack{j=1 \\ j \neq i}}^n V_{m_j}(q, \dot{q}) \dot{q}_j + h_i(\dot{q}) + g_i(q) + T_{d_i}(t) \quad (4.1.4)$$

On définit pour cette articulation:

L'erreur de poursuite:

$$e_i(t) = q_{d_i}(t) - q_i(t) \quad (4.1.5)$$

L'erreur pondérée:

$$r_i(t) = \dot{e}_i(t) + \lambda_i e_i(t) \quad (4.1.6)$$

La fonction couple non linéaire $f_i(x_i)$ tel que:

$$f_i(x_i) = \sum_{j=1}^n m_{ij} (\ddot{q}_{d_j} + \lambda_j \dot{e}_j) + \sum_{j=1}^n V_{m_j}(q, \dot{q}) (\dot{q}_{d_j} + \lambda_j e_j) + g_i(q) + h_i(q) \quad (4.1.7)$$

Avec:

$q_{d_i}(t)$: $i^{\text{ème}}$ trajectoire désirée.

$$\underline{x}_i = [e_i \quad \dot{e}_i \quad q_{d_i} \quad \dot{q}_{d_i} \quad \ddot{q}_{d_i}]^T \quad (4.1.8)$$

Structure du contrôleur: [18]

La structure du régulateur neuronale est la même que celle donnée par (3.22) ainsi pour chaque articulation, on définit la loi de commande suivante:

$$T_i(t) = \hat{f}_i(x_i) + K_{V_i} r_i(t) - V_i(t) \quad (4.1.9)$$

Ce régulateur est représenté sur la figure (4.1.1), il est constitué:

- D'un réseau de neurones identique à celui représenté à la figure (3.1), chargé d'estimer la fonction couple non linéaire défini dans (4.1.7) tel que:

$$\hat{f}_i(x_i) = \hat{W}_i^T \sigma(\hat{V}_i^T x_i) \tag{4.1.10}$$

Avec:

\hat{W}_i et \hat{V}_i : Paramètres du RNA i .

$\sigma(\cdot)$: Fonction d'activation (sigmoïde).

x_i : Entrée du RNA défini dans (4.1.8)

- D'un proportionnel-derivateur à gain fixe.
- D'un terme correspondant au couple robuste défini dans (3.29) son expression dans le cas décentralisé est:

$$V_i(t) = -K_z (\|\hat{Z}_i\|_f + Z_{M_i}) r_i \tag{4.1.11}$$

Avec:

K_z, Z_{m_i} : Constantes positives.

$$Z_i = \begin{bmatrix} W_i \\ V_i \end{bmatrix}$$

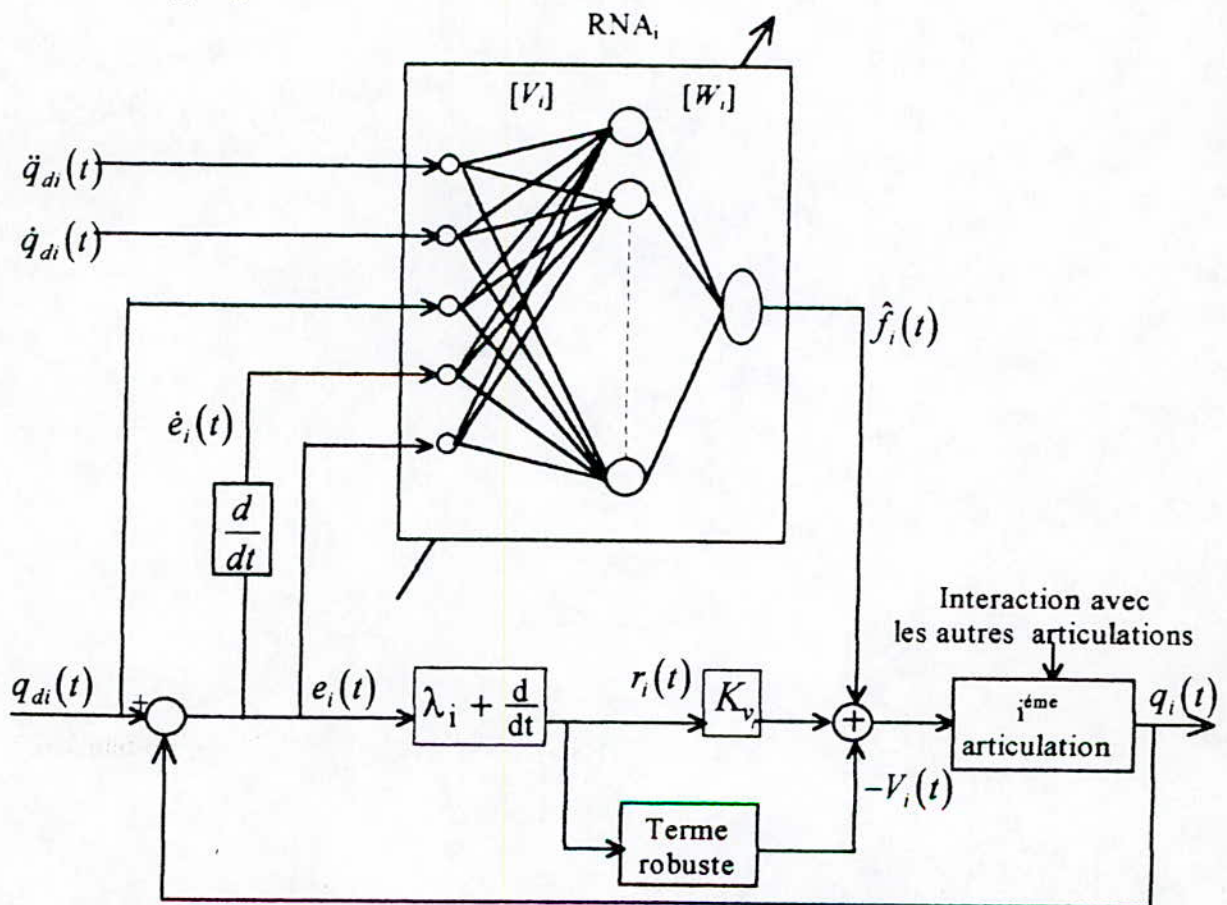


Fig. (4.1.1) Structure du $i^{\text{ème}}$ régulateur adaptatif par RNA

Algorithmes d'adaptation des paramètres du RNA_i :

Les lois d'ajustement en temps réel des paramètres du i^{ème} RNA associe à la i^{ème} articulation sont identiques à ceux obtenus dans l'approche centralisée de LEWIS [18] et définit dans (3.30a) et (3.30b) soit:

$$\begin{cases} \dot{\hat{W}}_i = F_i \hat{\sigma}_i r_i - F_i \hat{\sigma}' \hat{V}_i^T x_i r_i - K_i F_i |r_i| \hat{W}_i \\ \dot{\hat{V}}_i = G_i x_i r_i (\hat{\sigma}' \hat{W}_i)^T - K_i G_i |r_i| \hat{V}_i \end{cases} \quad (4.1.12)$$

Avec:

- F_i, G_i : Constantes positives.
- K_i : Constante positive ou nulles.

4.1.3. ANALYSE DE LA STABILITE

Pour valider un mécanisme d'adaptation quelconque, il faut étudier la stabilité du schéma adaptatif proposé.

Dans notre cas, on étudiera la stabilité du système global en reconstitutions une fonction candidate de LYAPUNOV semblable à celle définit au chapitre précédent mais qui rassemblera un ensemble de fonctions associes à chaque articulation. Ainsi, on parviendra à prouver la stabilité des mécanismes d'adaptations au niveau des sous systèmes en remontons au système global.

Pour cela, soit:

$$\begin{aligned} \hat{V}^T &= \text{Diag}(\hat{V}_1^T, \hat{V}_2^T, \dots, \hat{V}_n^T) = \text{Diag}(\hat{V}_i^T) \\ \hat{W}^T &= \text{Diag}(\hat{W}_1^T, \hat{W}_2^T, \dots, \hat{W}_n^T) = \text{Diag}(\hat{W}_i^T) \\ F &= \text{Diag}(F_1, F_2, \dots, F_n) = \text{Diag}(F_i) \\ G &= \text{Diag}(G_1, G_2, \dots, G_n) = \text{Diag}(G_i) \\ \Lambda &= \text{Diag}(\lambda_1, \lambda_2, \dots, \lambda_n) = \text{Diag}(\lambda_i) \\ K_v &= \text{Diag}(K_{v1}, K_{v2}, \dots, K_{vm}) = \text{Diag}(K_{vi}) \end{aligned}$$

Donc:

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x) = [\hat{f}_1(x_1) \hat{f}_2(x_2) \cdots \hat{f}_n(x_n)]^T$$

$$r = [r_1 r_2 \cdots r_n]^T$$

Avec:

$$x = [x_1^T x_2^T \cdots x_n^T]^T$$

Soit la fonction candidate de LYAPUNOV suivante:

$$L(r, \tilde{W}, \tilde{V}) = \frac{1}{2} r^T M r + \frac{1}{2} \sum_{i=1}^n \frac{1}{F_i} \tilde{W}_i^T \tilde{W}_i + \frac{1}{2} \sum_{i=1}^n \frac{1}{G_i} \tilde{V}_i^T \tilde{V}_i \quad (4.1.13)$$

Tel que:

$$\tilde{W}_i = W_i - \hat{W}_i, \quad \tilde{V}_i = V_i - \hat{V}_i$$

En prenant en compte les considérations données ci-dessus, l'équation (4.1.13) peut être réécrite sous la représentation matricielle suivante:

$$L(r, \tilde{W}, \tilde{V}) = \frac{1}{2} r^T M r + \frac{1}{2} \text{tr}(\tilde{W}^T F^{-1} \tilde{W}) + \frac{1}{2} \text{tr}(\tilde{V}^T G^{-1} \tilde{V}) \quad (4.1.14)$$

Donc, on aboutit à la même fonction de LYAPUNOV donnée dans (3.27). [18]

On procède de la même manière, on obtient la dérivée suivante:

$$\dot{L}(t) = -r^T K_V r = -\sum_{i=1}^n K_{V_i} (r_i)^2 < 0 \quad (4.1.15)$$

Cette dérivée est la somme de plusieurs fonctions définies négatives associées à chaque articulation.

Donc, le schéma de commande adaptatif décentralisé donné dans (4.1.9) ainsi que les lois d'adaptations définies dans (4.1.12) garantissent la convergence de tous les signaux vers un équilibre asymptotiquement stable.

En plus, les erreurs paramétriques \tilde{W}_i , \tilde{V}_i et l'erreur de poursuite $e_i(t)$ convergent asymptotiquement vers zéro donc:

$$\lim_{t \rightarrow \infty} e_i(t) = 0$$

4.1.4. SIMULATION :

PARAMETRES DE COMMANDE :

Comme il existe trois articulations, donc on a trois stations de commande, et chaque stations est caractérisé comme suite:

Station 1: Un réseau de neurones comporte 5 entrées, 10 neurones dans la couche cachée et une sortie.

$$K_v=20, \quad \lambda=10, \quad F=0.35, \quad G=1.8, \quad k=0.007, \quad K_z=1, \quad Z_m=20$$

Station 2: Un réseau de neurones similaire à celui de la station 1 avec (5,10,1) entrées, couche cachée et sortie respectivement.

$$K_v=20, \quad \lambda=10, \quad F=3.05, \quad G=0.8, \quad k=0.01, \quad K_z=1, \quad Z_m=20$$

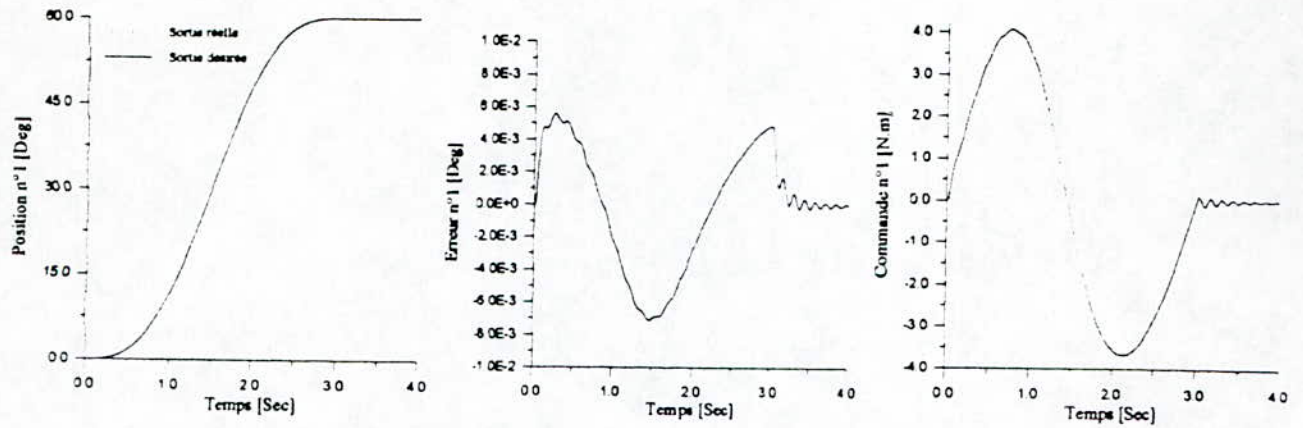
Station 3: Un réseau de neurones (5,10,1).

$$K_v=20, \quad \lambda=10, \quad F=0.55, \quad G=0.8, \quad k=0.01, \quad K_z=1, \quad Z_m=20$$

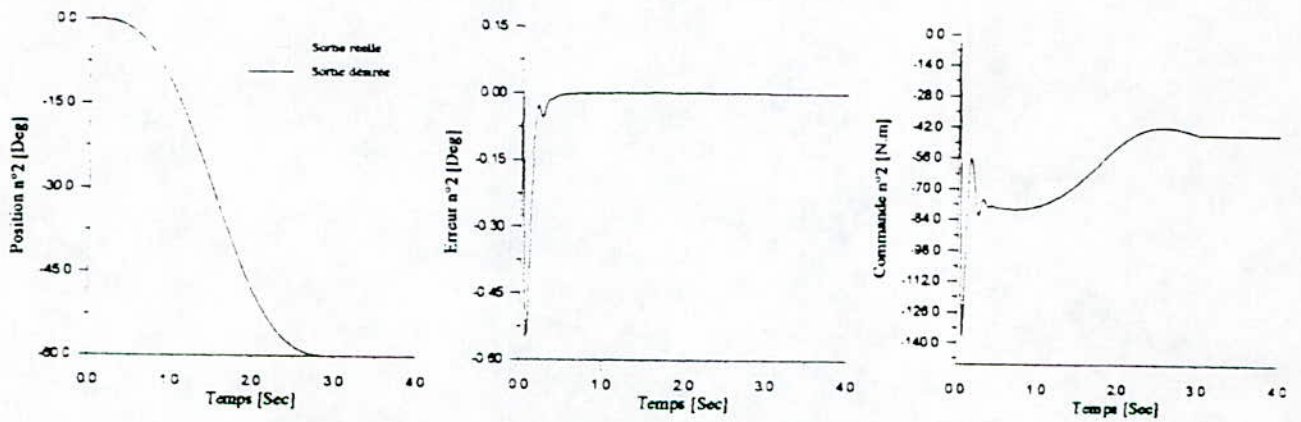
INTERPRETATION DES RESULTATS :

- **Test de validation :** Nous remarquons d'après la figure (4.1.2) que les erreurs de poursuite de toutes les articulations s'annulent (négligeant les petits valeurs) après des pics d'erreurs en régime transitoire, Le pic le plus important est constater sur la deuxième articulation due au faite qu'elle supporte plus de charge que les autres articulations. sur la figure (4.1.3), une détérioration des performance est constater due, sans doute, à l'enlèvement des signaux émis par les RNA, ce qui montre l'importance capitale de ces signaux dans le schéma de commande.
- **Variation massique subite :** Les figures (4.1.4) et (4.1.5) montre la robustesse de la commande face à l'effet de masse de la charge. Ainsi, même pour des charges importantes la commande répond très rapidement à ces variations massiques.
- **Rupture de la commande :** Les figures (4.1.6), (4.1.7) et (4.1.8) montre l'insensibilité du système face à la rupture d'une commande car, sur ces figures, on remarque que même si la commande d'une articulation est interrompu induisant ainsi à la détérioration de la réponse de l'articulations en question, les autres articulations ne sont pas affectées et leurs erreurs de poursuite reste borné suffisamment petit pour être négligé, malgré l'effet de cette l'articulation sur la dynamique des autres. Ceci est prévue, car contrairement à la commande centralisée, la commande décentralisée permet de localiser les différentes commandes et donne un aspect d'indépendance pour chaque articulation.

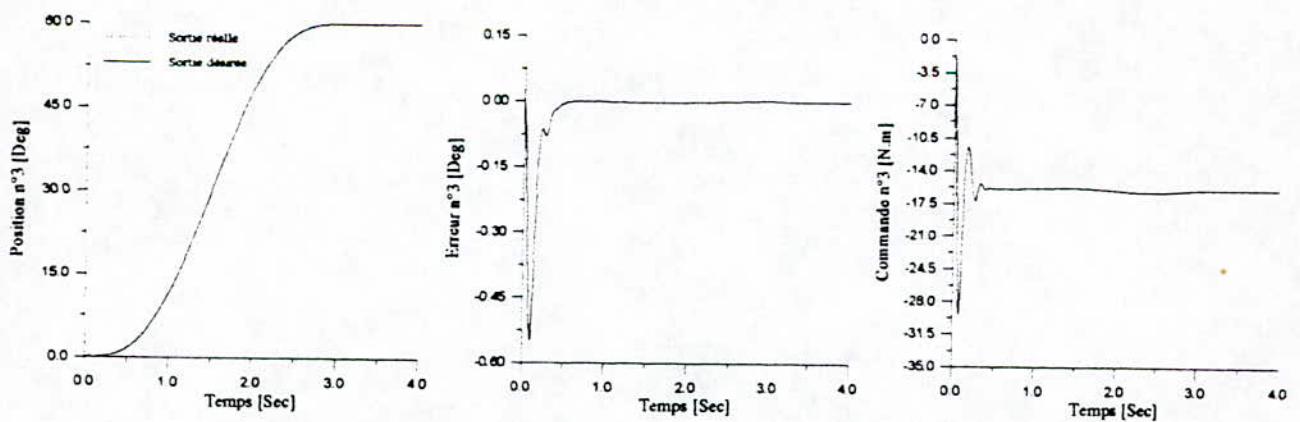
- **Test de LEAHVY** : La figure (4.1.9) permet d'illustrer la robustesse de la commande décentralisée par rapport à des variations rapides des termes d'interconnexions due à la rapidité de la trajectoire imposée par Leahvy. Comparant avec la figure (4.1.10) on remarque rapidement l'effet des RNA dans la commande.



Position, Erreur et Commande de la première articulation.

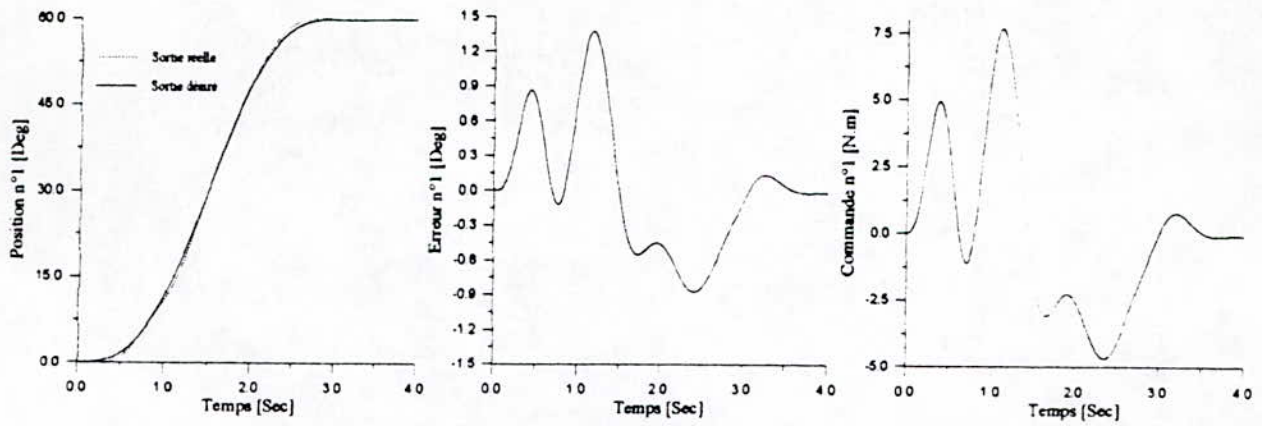


Position, Erreur et Commande de la deuxième articulation.

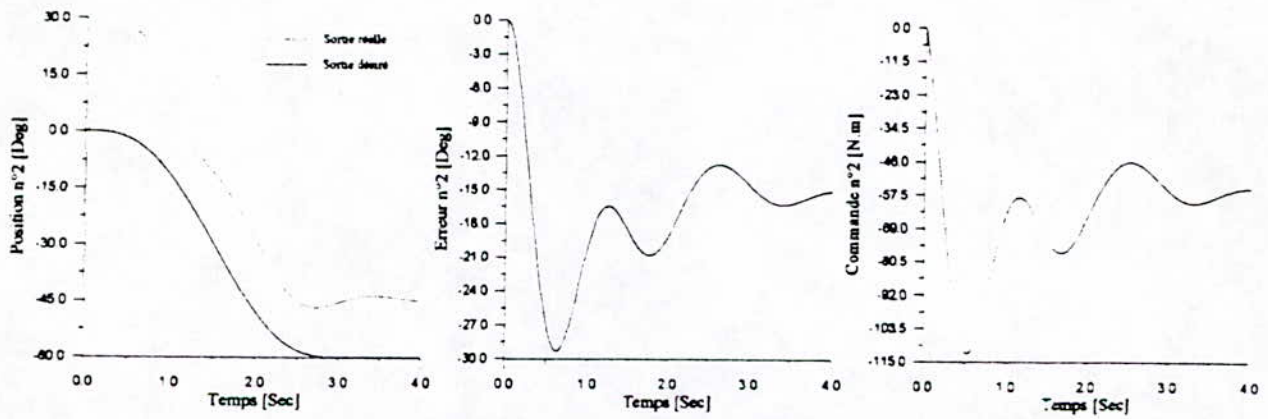


Position, Erreur et Commande de la troisième articulation.

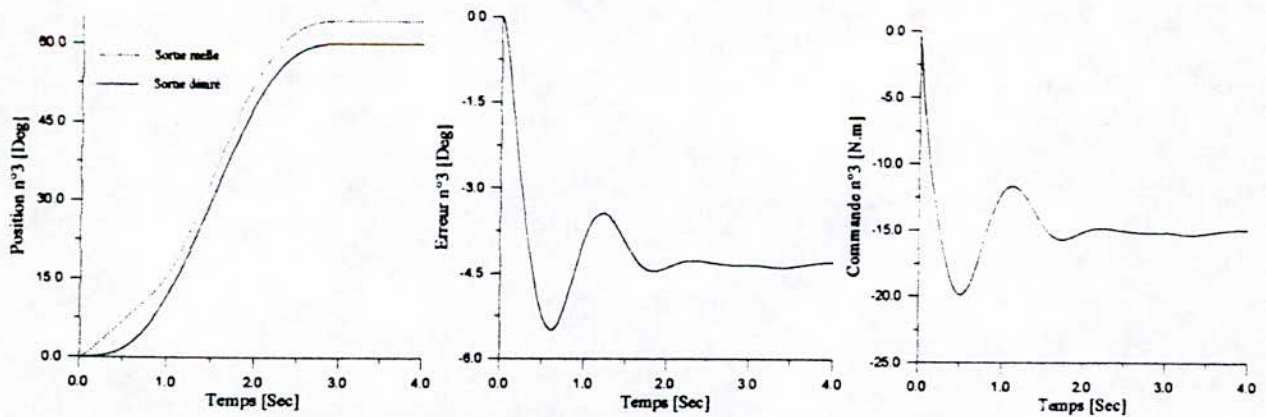
Figure 4.1.2
Réponses en boucle fermée sans charge (à vide).



Position, Erreur et Commande de la première articulation.

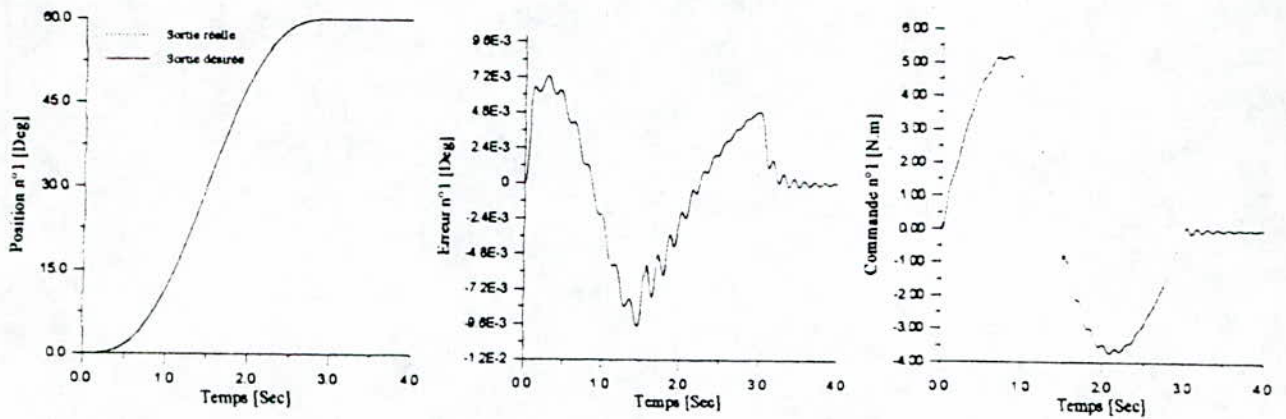


Position, Erreur et Commande de la deuxième articulation.

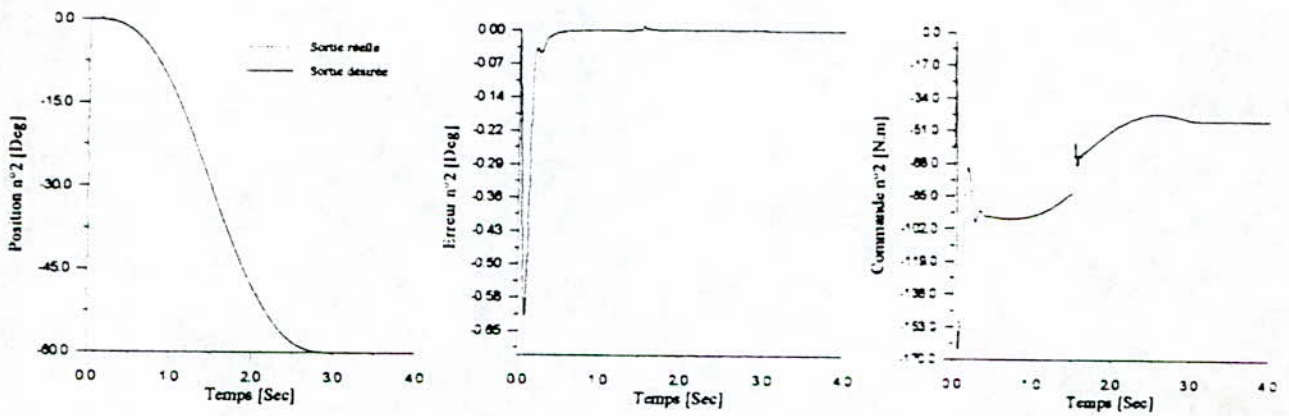


Position, Erreur et Commande de la troisième articulation.

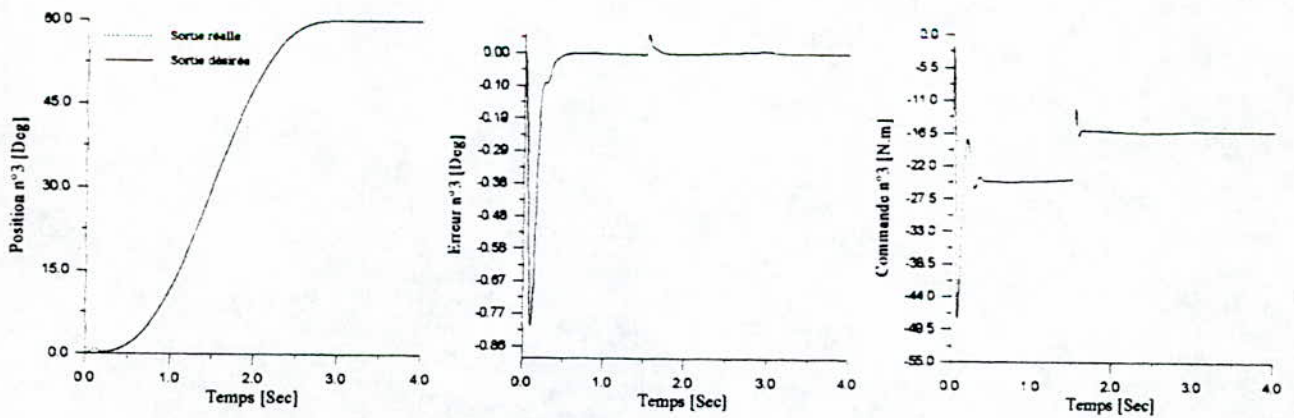
Figure 4.1.3
Réponses en boucle fermée avec le P.D. fixe (à vide).



Position, Erreur et Commande de la première articulation.

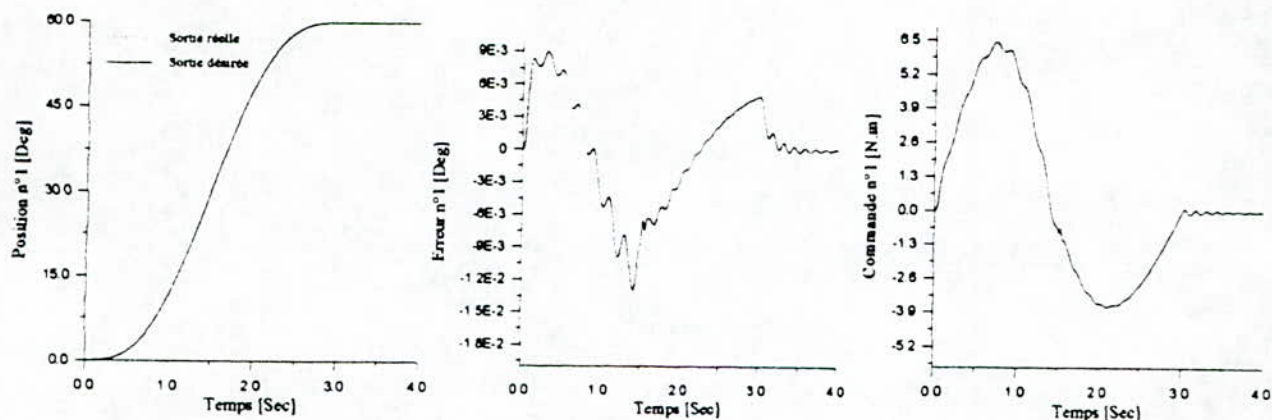


Position, Erreur et Commande de la deuxième articulation.

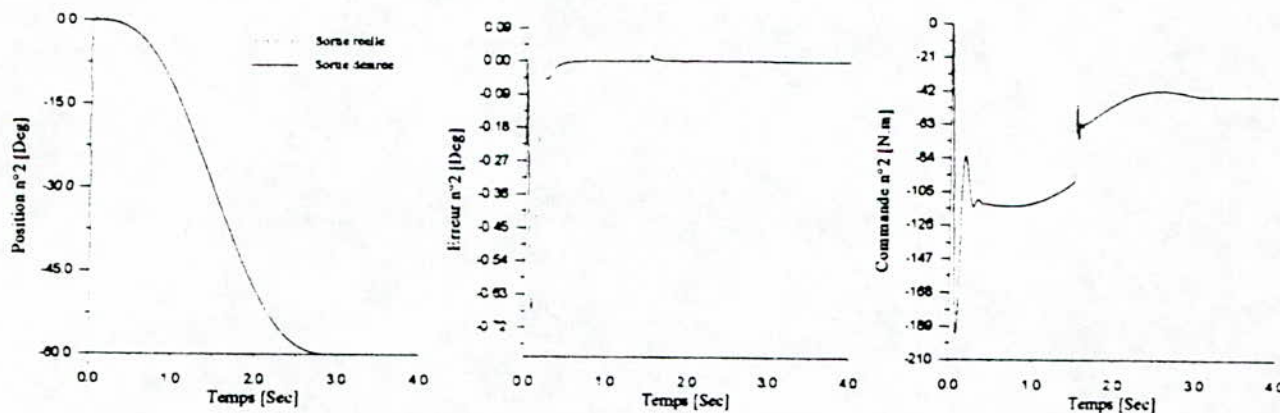


Position, Erreur et Commande de la troisième articulation.

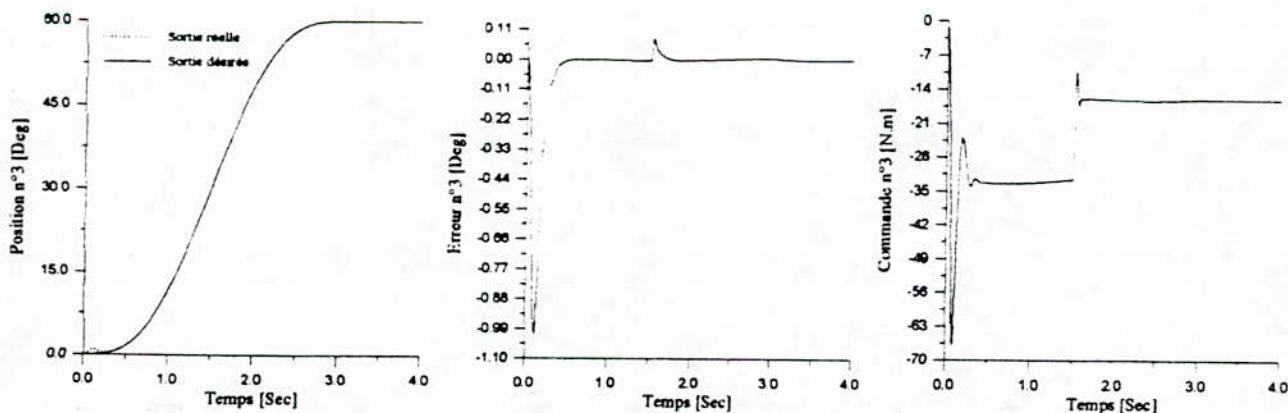
Figure 4.1.4
Réponses en boucle fermée avec une charge de 2 Kg (lâchée à l'instant $t=1.5s$).



Position, Erreur et Commande de la première articulation.

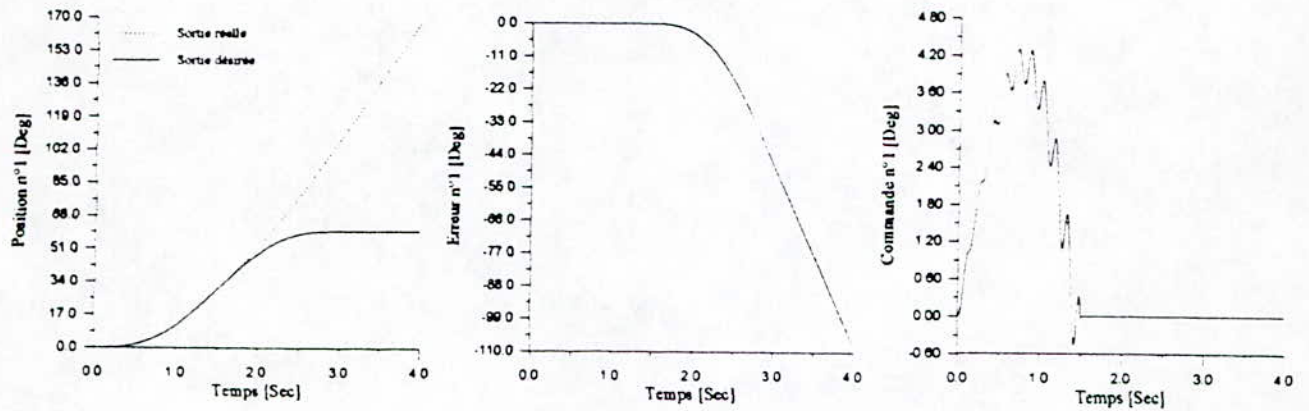


Position, Erreur et Commande de la deuxième articulation.

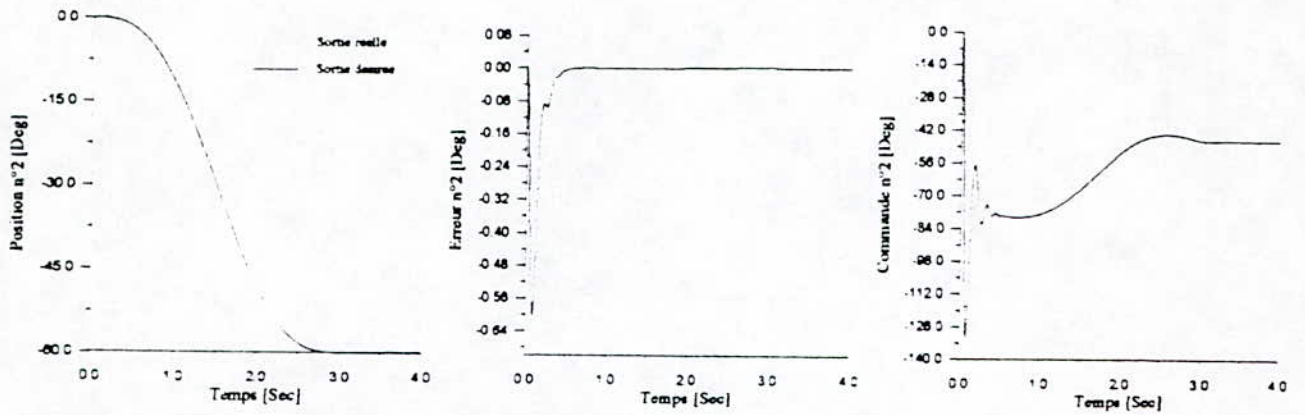


Position, Erreur et Commande de la troisième articulation.

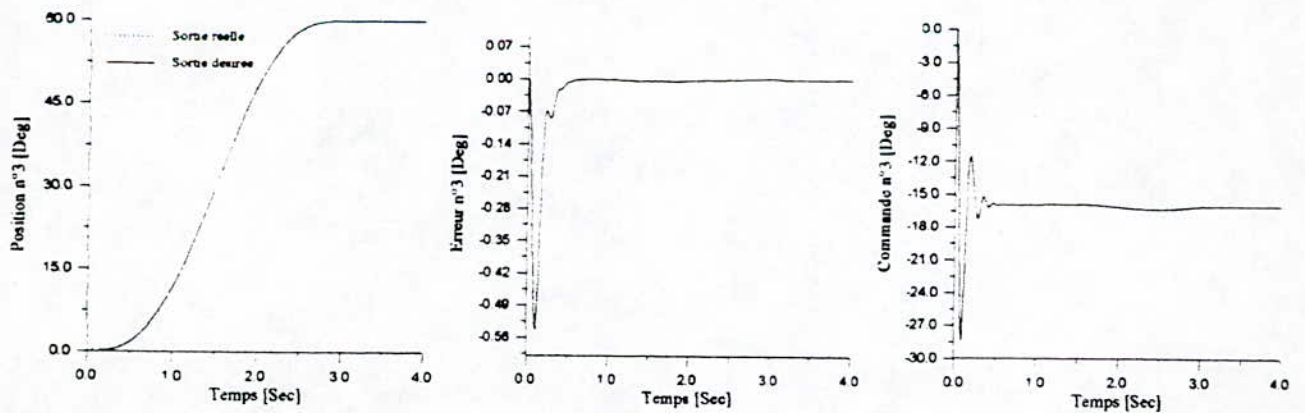
Figure 4.1.5
Réponses en boucle fermée avec une charge de 4 Kg (lâchée à l'instant t=1.5s).



Position, Erreur et Commande de la première articulation.

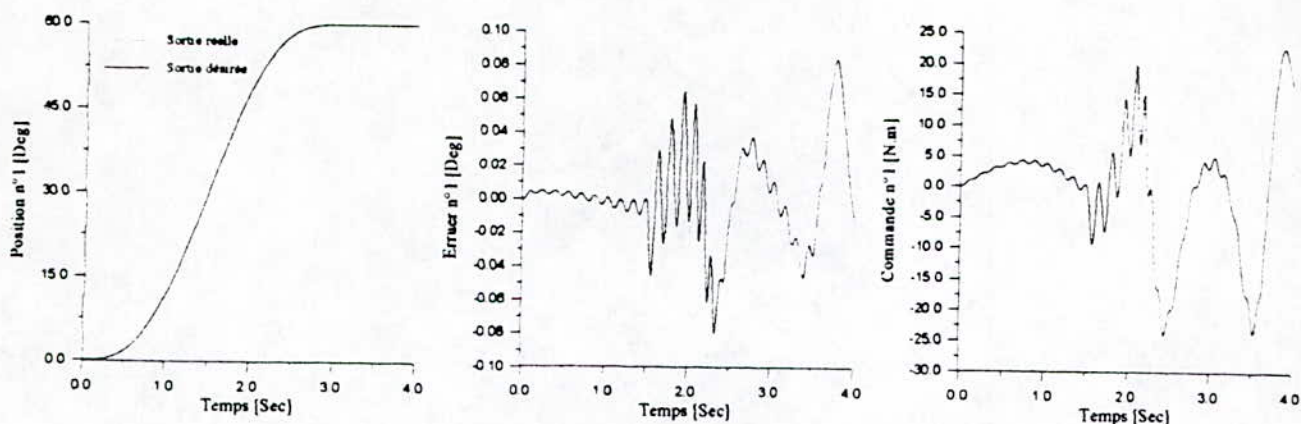


Position, Erreur et Commande de la deuxième articulation.

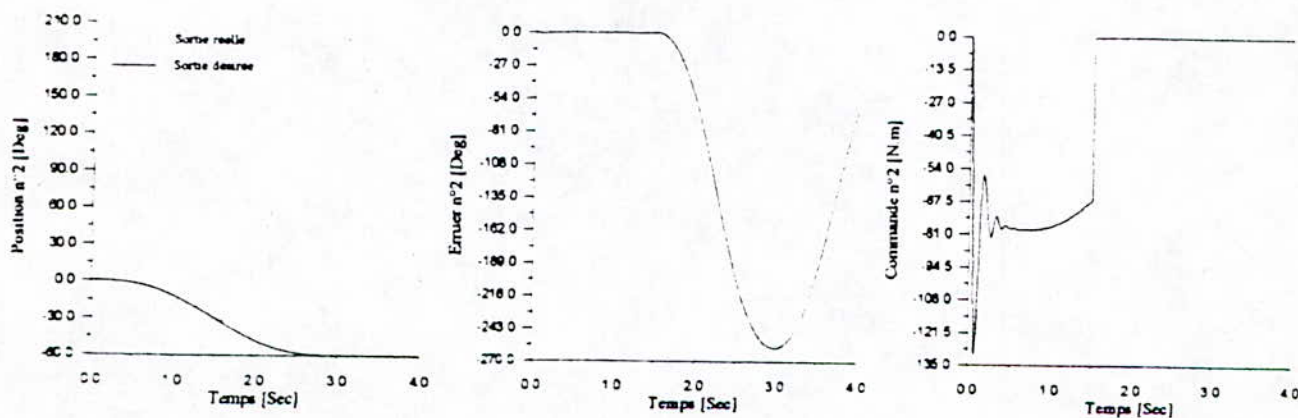


Position, Erreur et Commande de la troisième articulation.

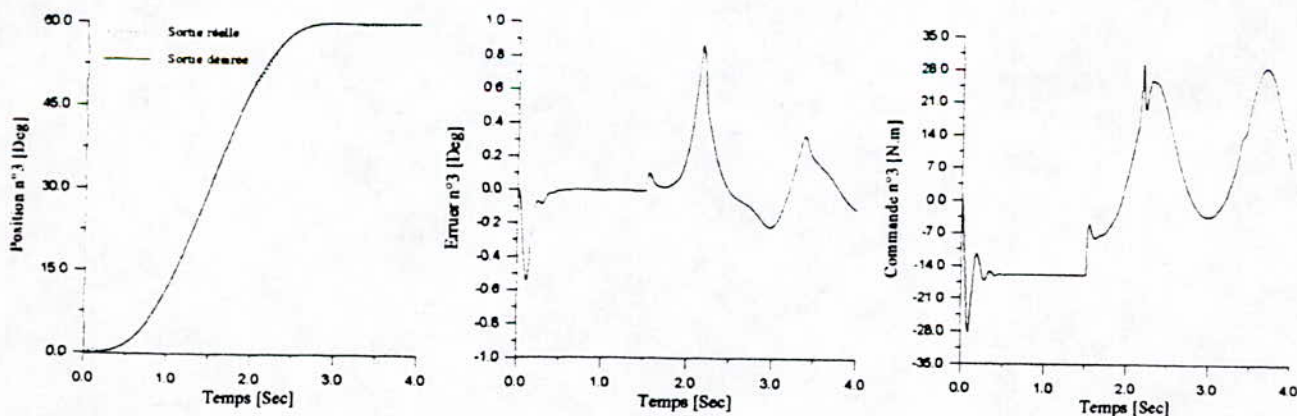
Figure 4.1.6
Réponses en boucle fermée avec rupture de la commande, $u(1)=0$ à $t=1.5s$.



Position, Erreur et Commande de la première articulation.

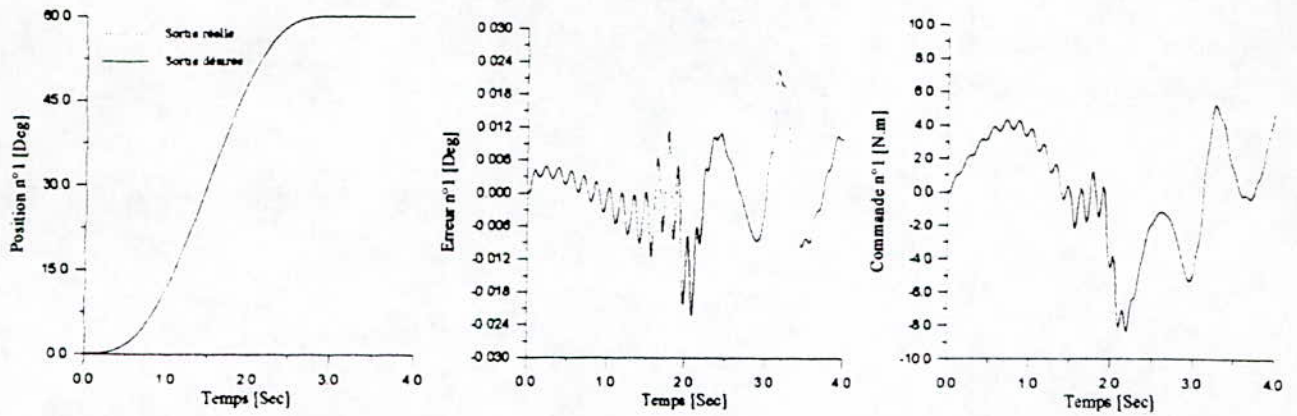


Position, Erreur et Commande de la deuxième articulation.

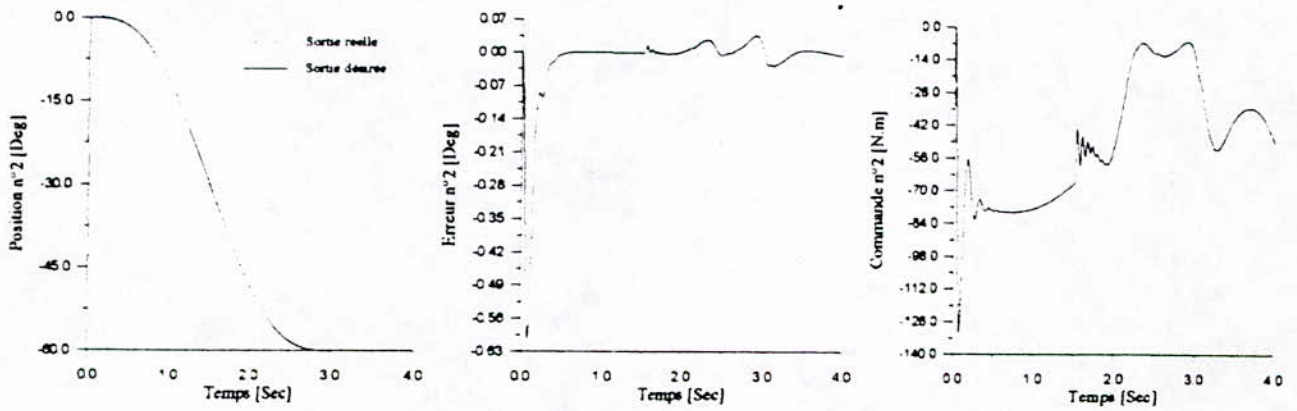


Position, Erreur et Commande de la troisième articulation.

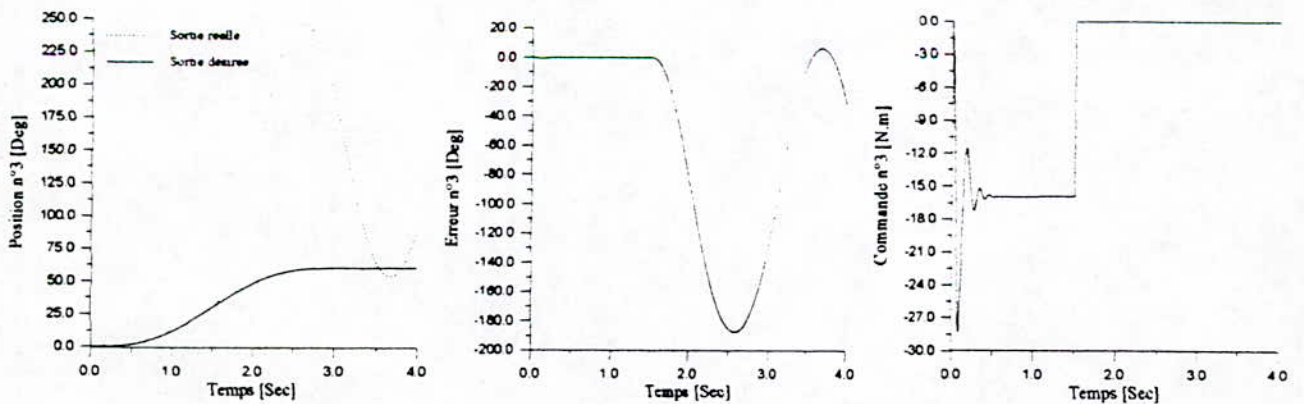
Figure 4.1.7
Réponses en boucle fermée avec rupture de la commande, $u(2)=0$ à $t=1.5s$.



Position, Erreur et Commande de la première articulation.

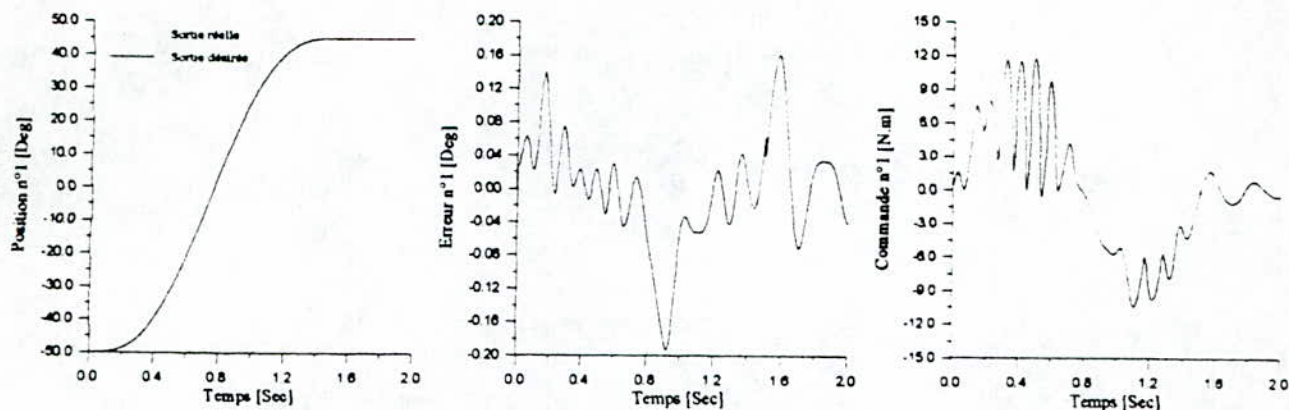


Position, Erreur et Commande de la deuxième articulation.

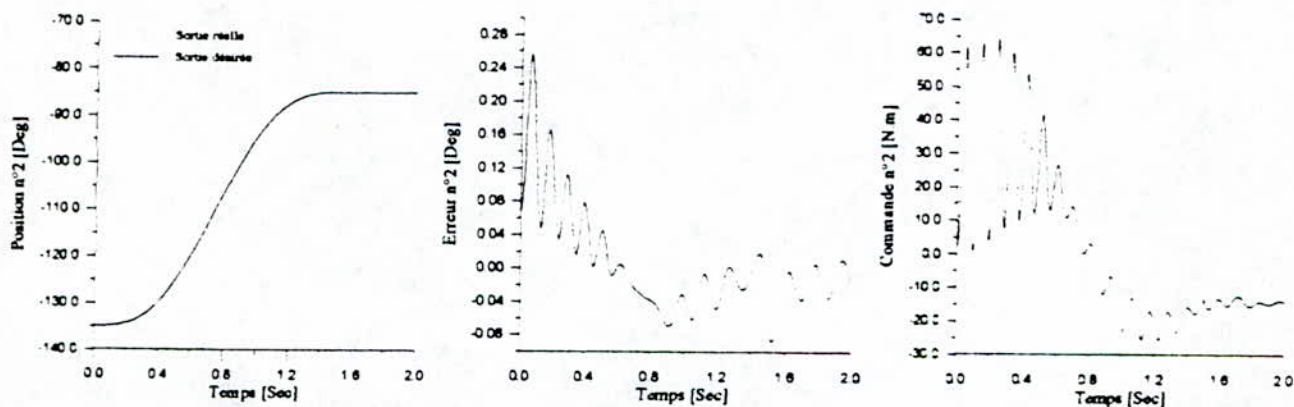


Position, Erreur et Commande de la troisième articulation.

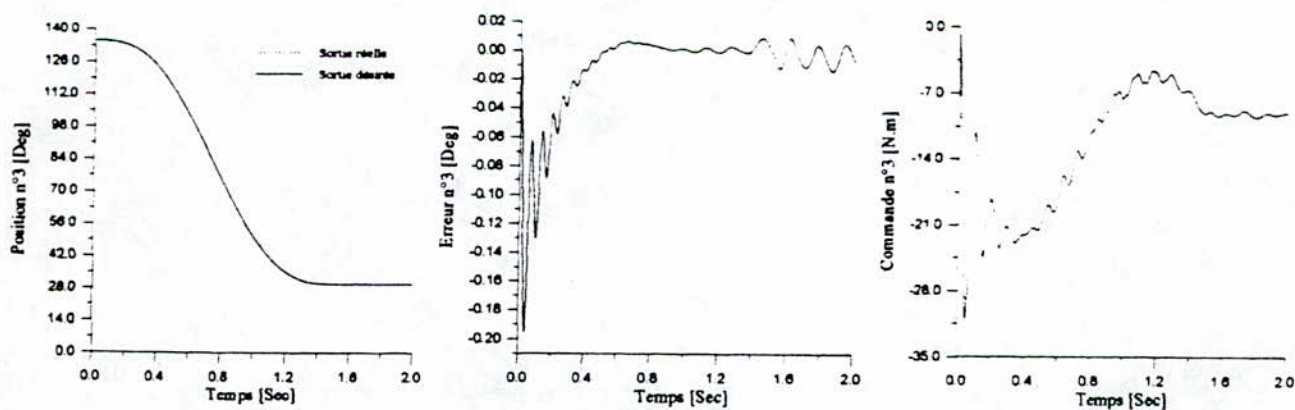
Figure 4.1.8
Réponses en boucle fermée avec rupture de la commande, $u(3)=0$ à $t=1.5s$.



Position, Erreur et Commande de la première articulation.

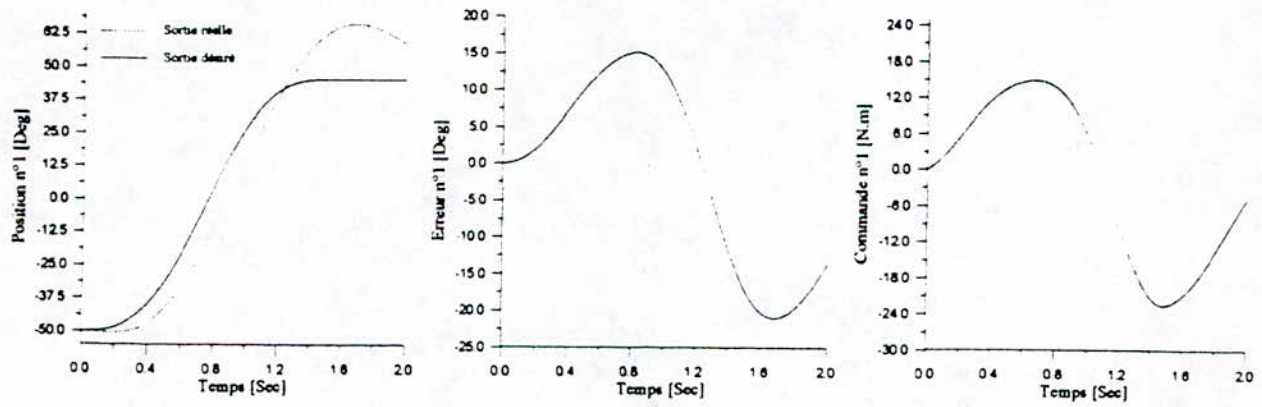


Position, Erreur et Commande de la deuxième articulation.

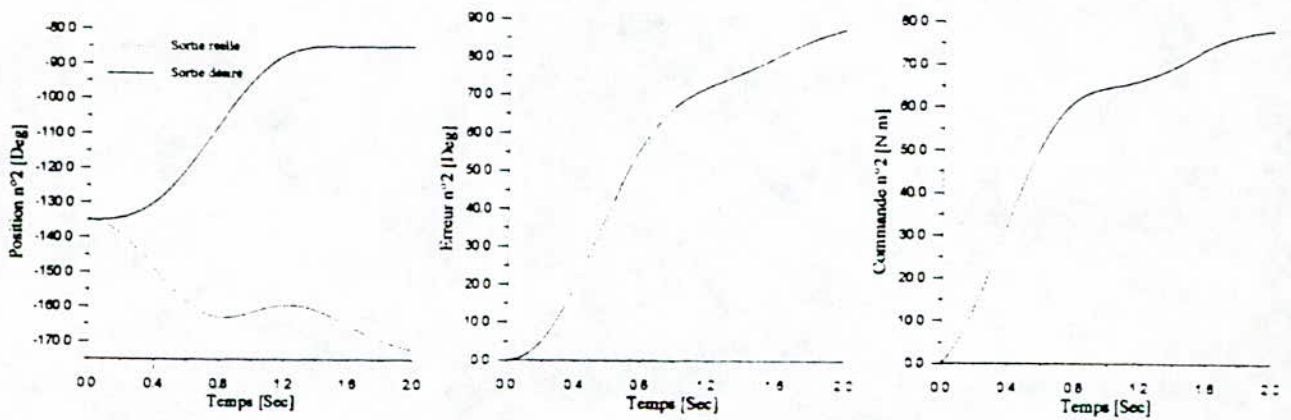


Position, Erreur et Commande de la troisième articulation.

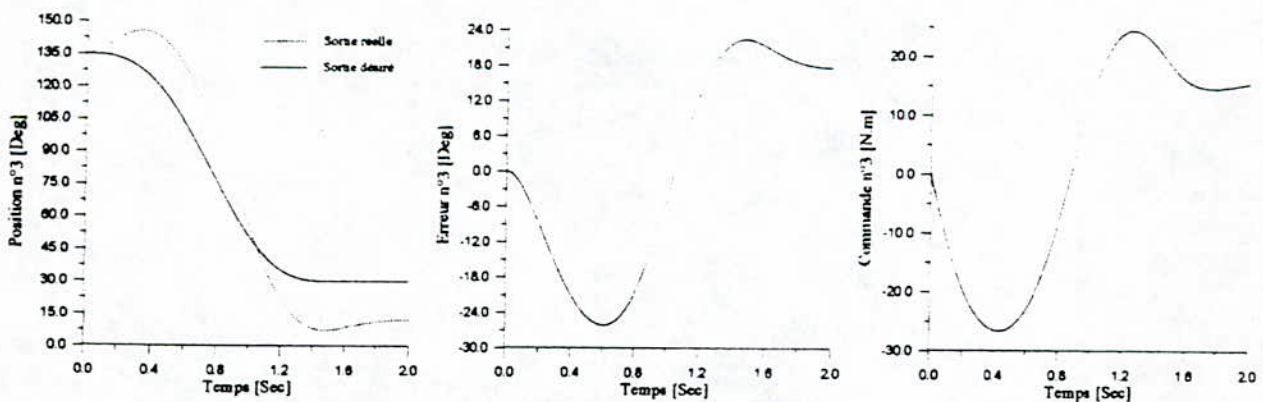
Figure 4.1.9
Réponses en boucle fermée sous le test de Leavhy.



Position, Erreur et Commande de la première articulation.



Position, Erreur et Commande de la deuxième articulation.



Position, Erreur et Commande de la troisième articulation.

Figure 4.1.10
Réponses en boucle fermées sous le test de LEAHVY avec un PD fixe.

PARTIE 2: ETUDE ET CONCEPTION D'UNE COMMANDE ADAPTATIVE DECENTRALISEE PAR RNA

4.2.1. INTRODUCTION

A travers le chapitre précédent, nous avons présentés une nouvelle technique de commande neuronale, basé sur l'ajustement en temps réel des paramètres du réseau de neurone selon des algorithmes d'adaptation qui assurent la stabilité du système global.

Toutefois, cette nouvelle stratégie de commande quoiqu'efficace, présente néanmoins quelques inconvénients rencontrés en simulation tel que la difficulté de choisir un gain adéquat du PD ainsi que le nombre élevé des paramètres (gain et pondération) à fixés judicieusement pour garantir les performances souhaitées.

Dans ce présent chapitre, nous proposons une nouvelle technique de commande adaptative décentralisée par RNA permettant plus de souplesse lors de l'implémentation.

Motiver par les travaux de LEWIS [18] dans le domaine de la commande par RNA et de ceux de SERAJI [26] en commande adaptative décentralisée, nous proposons d'associer au réseau de neurone un proportionnelle dérivateur à gain ajustable et de changer la fonction non linéaire estimé par le RNA d'une manière intelligente afin de pouvoir dérivé les lois d'adaptation. L'imposition d'une dynamique à l'erreur de poursuite permettra de prouver la stabilité de ce schéma de commande par l'approche de LYAPUNOV au niveau de chaque articulation.

4.2.2. STRUCTURE DU REGULATEUR ADAPTATIF DECENTRALISE PAR RNA :

Le schéma adaptatif neuronal qu'on étudiera ici est différent de celui exposé auparavant. Cependant, toutes les considérations sur le système interconnecté ainsi que le bloc neuronal sont identiques à ceux donner dans le chapitre 3.

4.2.2.1. POSITION DU PROBLEME

Considérons un bras manipulateur décrit par l'équation:[26]

$$M(q)\ddot{q} + N(q, \dot{q}) + G(q) + H(\dot{q}) = T \quad (4.2.1)$$

$q \in \mathbb{R}^n$: Vecteur des variables articulaires généralisées.

$M(q) \in \mathbb{R}^{n \times n}$: Matrice d'inertie symétrique définie positive.

$N(q, \dot{q}) \in \mathbb{R}^n$: Vecteur des forces de coriolis et centrifuge.

$G(q) \in \mathbb{R}^n$: Vecteur des forces de gravité.

$H(\dot{q}) \in \mathbb{R}^n$: Vecteur des forces de frottement.

$T \in \mathbb{R}^n$: Vecteur des forces et / ou des couples.

Chaque articulation peut être représentée par l'équation:

$$m_{ii} \ddot{q}_i(t) + \left[\sum_{\substack{j=1 \\ j \neq i}}^n m_{ij} \ddot{q}_j(t) \right] + n_i(q, \dot{q}) + g_i(q) + h_i(q) = T_i(t) \quad (4.2.2)$$

$$m_{ii} \ddot{q}_i(t) + d_i(q, \dot{q}, \ddot{q}) = T_i(t) \quad , i=1, n \quad (4.2.3)$$

Avec:

$$d_i(q, \dot{q}, \ddot{q}) = \left[\sum_{\substack{j=1 \\ j \neq i}}^n m_{ij} \ddot{q}_j(t) \right] + n_i(q, \dot{q}) + g_i(q) + h_i(q) \quad (4.2.4)$$

Le problème de la commande décentralisée adaptative par RNA se présente comme suit:

Pour la $i^{\text{ème}}$ articulation décrite par (4.2.2), détermine la loi de commande locale $T_i(t)$ qui garantira une poursuite asymptotique de la trajectoire désirée $q_{di}(t)$, en utilisant uniquement les informations disponibles localement.

4.2.2.2. SCHEMA DE COMMANDE :

Pour chaque articulation, on définit le régulateur régi par la loi de commande suivante:

$$T_i(t) = \hat{f}_i(t) + [K_{p_i}(t)e_i(t) + K_{v_i}(t)\dot{e}_i(t)] \quad (4.2.5)$$

Avec:

$$e_i(t) = q_{di}(t) - q_i(t) : \text{erreur de poursuite en position.}$$

Ce régulateur est composé:

- * D'un terme $\hat{f}_i(t)$ représentant l'estimation de la fonction couple non linéaire suivante:

$$f_i(t) = m_{ii} \ddot{q}_{di}(t) + d_i(q, \dot{q}, \ddot{q}) \tag{4.2.6}$$

Et correspondant à la sortie du RNA.

- * D'un terme $[K_{P_i}(t)e_i(t) + K_{V_i}(t)\dot{e}_i(t)]$ correspondant à un régulateur proportionnel dérivateur adaptatif.

On définit l'erreur pondérée:

$$r_i(t) = \omega_{P_i} e_i(t) + \omega_{V_i} \dot{e}_i(t) \tag{4.2.7}$$

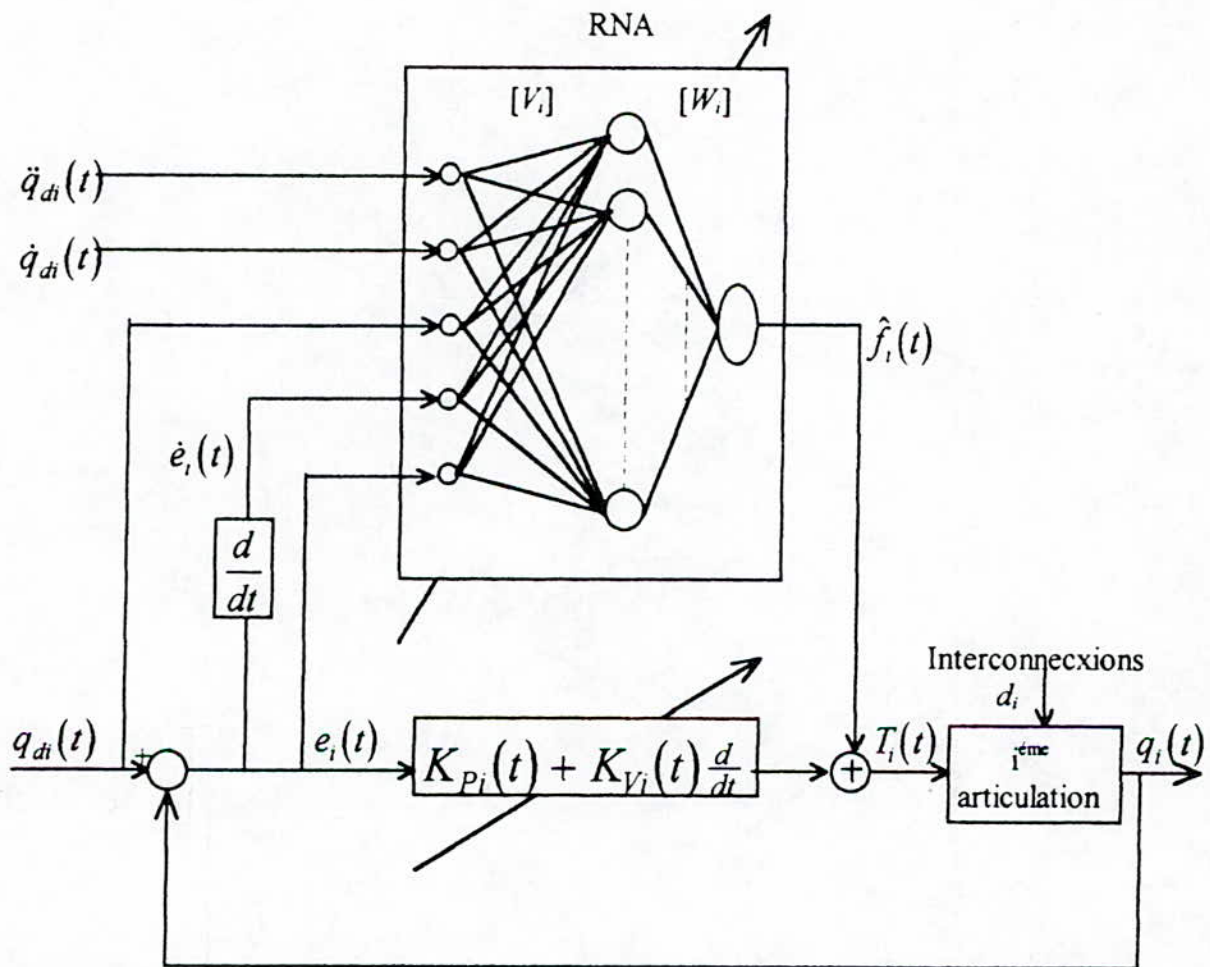


Fig.(4.2.1) Structure du contrôleur adaptatif par RNA associé à la i^{ème} articulation

A°. LE RESEAU DE NEURONE

Le réseau de neurone utilisé ici est un réseau à trois couches, identique à celui représenté auparavant sur la figure (3.1) est décrit par:

$$\hat{f}(x_i) = \hat{W}_i^T \sigma(\hat{V}_i^T x_i) \quad : \text{Sortie de RNA.}$$

Avec:

$$x_i = [e_i(t) \dot{e}_i(t) q_{di}(t) \dot{q}_{di}(t) \ddot{q}_{di}(t)] \quad : \text{Entrée du RNA.}$$

$\sigma(.)$: Fonction d'activation (sigmoïde).

On définit les erreurs d'estimation suivantes:

$$\begin{aligned} \tilde{W}_i &= W_i - \hat{W}_i, \quad \tilde{V}_i = V_i - \hat{V}_i \text{ et } \tilde{Z}_i = Z_i - \hat{Z}_i \\ \tilde{\sigma}_i &= \sigma(V_i^T x_i) - \sigma(\hat{V}_i^T x_i) = \sigma_i - \hat{\sigma}_i \end{aligned}$$

Avec:

$$Z_i = \begin{bmatrix} W_i \\ V_i \end{bmatrix}$$

D'après l'équation (Chapitre 3 équation (3.16)), on a:

$$\tilde{\sigma}_i = \sigma'(\hat{V}_i^T x_i) \tilde{V}_i^T x_i + o(\tilde{V}_i^T x_i) \quad (4.2.8)$$

W_i et V_i correspondent aux poids idéaux permettant de reconstruire la fonction décrite dans (4.2.6).

Soit:

$$f(x) = W_i^T \sigma(V_i x_i) + \xi_i$$

L'erreur d'estimation fonctionnelle est:

$$\begin{aligned} \tilde{f}(x_i) &= f(x_i) - \hat{f}(x_i) = W_i^T \sigma(V_i^T x_i) + \xi_i - \hat{W}_i^T \sigma(\hat{V}_i^T x_i) \\ \tilde{f}(x_i) &= W_i^T \sigma_i - \hat{W}_i^T \hat{\sigma}_i + \xi_i \end{aligned}$$

Parés calcul, on aboutit à l'équation suivante:

$$\tilde{f}(x_i) = \tilde{W}_i^T \hat{\sigma}_i + \hat{W}_i^T \tilde{\sigma}_i + \xi_i$$

en remplaçant (4.2.8) on aura:

$$\tilde{f}(x_i) = \tilde{W}_i^T (\hat{\sigma}_i - \hat{\sigma}_i' \hat{V}_i^T x_i) + \hat{W}_i^T \hat{\sigma}_i' \tilde{V}_i^T x_i + w_i(t) \quad (4.2.9)$$

Avec:

$$w_i(t) = \tilde{W}_i^T \hat{\sigma}_i' V_i^T x_i + W_i^T O(\tilde{V}_i^T x_i)^2 + \xi_i \quad (4.2.10)$$

Terme regroupant toutes les erreurs d'estimation infiniment petit et considéré comme étant une perturbation.

B°. DERIVATION DES LOIS D'ADAPTATION

Considérons la $i^{ème}$ articulation décrit par (4.2.3) et (4.2.4), en remplaçant l'expression de la loi de commande donnée par (4.2.5), on obtient la dynamique de l'erreur suivante:

$$m_{ii} \ddot{e}_i + K_{p_i} \dot{e}_i + K_{v_i} e_i = m_{ii} \ddot{q}_{di} + d_i(q, \dot{q}, \ddot{q}) - \hat{f}(x_i) \quad (4.2.11)$$

En posant:

$$X_i(t) = [e_i(t) \quad \dot{e}_i(t)]^T$$

La dynamique de l'erreur peut être représenté sous la forme d'état suivante:

$$\dot{X}_i(t) = \begin{bmatrix} 0 & 0 \\ -K_{p_i} & -K_{v_i} \\ m_{ii} & m_{ii} \end{bmatrix} X_i(t) + \frac{1}{m_{ii}} \begin{bmatrix} 0 \\ f(x_i) - \hat{f}(x_i) \end{bmatrix} \quad (4.2.12)$$

On définit un modèle de référence $e_{mi}(t)$, décrit par l'équation différentielle du 2^{ème} ordre suivant:

$$\ddot{e}_{mi}(t) + 2\xi_i \omega_i \dot{e}_{mi}(t) + \omega_i^2 e_{mi}(t) = 0 \quad (4.2.13)$$

ξ_i : Coefficient d'amortissement.

ω_i : pulsation propre.

(4.2.13) peut être écrite sous la représentation d'état suivante:

$$\dot{X}_{mi}(t) = \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\xi_i \omega_i \end{bmatrix} \cdot X_{mi}(t) = D \cdot X_{mi}(t) \quad (4.2.14)$$

Avec:

$$X_{mi}(t) = [e_{mi}(t) \quad \dot{e}_{mi}(t)]^T$$

Puisque le modèle de référence est stable, il existe d'après le théorème de KALMAN - YACUBOVICH [16], une matrice P symétrique définie positive:

$$P = \begin{bmatrix} P_1 & P_2 \\ P_2 & P_3 \end{bmatrix}$$

Est Q matrice définie positive, tel que:

$$PD + D^T P = -Q \tag{4.2.15}$$

Le modèle de référence étant stable, si on choisit:

$$X_{mi}(0) = \begin{bmatrix} e_{mi}(0) \\ \dot{e}_{mi}(0) \end{bmatrix} \equiv 0 \Rightarrow X_{mi}(t) \equiv 0 \quad \text{pour tout } t.$$

On définit l'erreur:

$$\begin{aligned} E_i(t) &= X_{mi}(t) - X_i(t) \\ \dot{E}_i(t) &= \dot{X}_{mi}(t) - \dot{X}_i(t) \end{aligned} \tag{4.2.16}$$

D'après (4.2.12) et (4.2.14) on obtient:

$$\begin{aligned} \dot{E}_i(t) &= \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & 2\xi_i\omega_i \end{bmatrix} E_i(t) + \begin{bmatrix} 0 & 0 \\ \frac{K_{Pi}}{m_{ii}} - \omega_i^2 & \frac{K_{Vi}}{m_{ii}} - 2\xi_i\omega_i \end{bmatrix} X_i(t) \\ &\quad + \frac{1}{m_{ii}} \begin{bmatrix} 0 \\ f_i - \hat{f}_i \end{bmatrix} \end{aligned} \tag{4.2.17}$$

$$\tilde{f}_i = f_i - \hat{f}_i$$

En remplaçant \tilde{f}_i par sa valeur (4.2.9), on obtient:

$$\begin{aligned} \dot{E}_i(t) &= \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\xi_i\omega_i \end{bmatrix} E_i(t) + \begin{bmatrix} 0 & 0 \\ \frac{K_{Pi}}{m_{ii}} - \omega_i^2 & \frac{K_{Vi}}{m_{ii}} - 2\xi_i\omega_i \end{bmatrix} X_i(t) \\ &\quad + \frac{1}{m_{ii}} \begin{bmatrix} 0 \\ \tilde{W}_i^T (\hat{\sigma}_i - \hat{\sigma}_i' \hat{V}_i^T x_i) + \tilde{W}_i^T \hat{\sigma}' \tilde{V}_i^T x_i + w_i(t) \end{bmatrix} \end{aligned} \tag{4.2.18}$$

Soit la fonction positive candidate de LYAPUNOV:

$$V_i = \frac{1}{2} E^T P E + \frac{1}{2} Q_1 \left(\frac{K_{P_i}}{m_{ii}} - \omega_i^2 - K_{P_i}^* \right)^2 + \frac{1}{2} Q_2 \left(\frac{K_{V_i}}{m_{ii}} - 2\xi_i \omega_i - K_{V_i}^* \right)^2 + \frac{\Gamma_1}{2} \text{tr}(\tilde{W}_i^T \tilde{W}_i) + \frac{\Gamma_2}{2} \text{tr}(\tilde{V}_i^T \tilde{V}_i) \quad (4.2.19)$$

Avec:

$K_{P_i}^*$ et $K_{V_i}^*$: Fonction variantes dans le temps qu'on spécifiera l'expression ultérieurement.

Q_1 et Q_2 : Constantes positives.

Γ_1 et Γ_2 : Constantes positives.

En dérivant $V_i(t)$, on obtient:

$$\dot{V}_i(t) = \frac{1}{2} (\dot{E}_i^T P E_i + E_i^T P \dot{E}_i) + Q_1 \left(\frac{K_{P_i}}{m_{ii}} - \omega_i^2 - K_{P_i}^* \right) \left(\frac{\dot{K}_{P_i}}{m_{ii}} - \dot{K}_{P_i}^* \right) + Q_2 \left(\frac{K_{V_i}}{m_{ii}} - 2\xi_i \omega_i - K_{P_i}^* \right) \left(\frac{\dot{K}_{V_i}}{m_{ii}} - \dot{K}_{V_i}^* \right) + \Gamma_1 \text{tr}(\tilde{W}_i^T \dot{\tilde{W}}_i) + \Gamma_2 \text{tr}(\tilde{V}_i^T \dot{\tilde{V}}_i) \quad (4.2.20)$$

Après calcul et simplification, on aboutit à l'équation suivante:

$$\dot{V}_i(t) = \dot{V}_l(t) + \frac{1}{m_{ii}} \left(\tilde{W}_i^T (\hat{\sigma}_i - \hat{\sigma}_i' \hat{V}_i^T x_i) + \hat{W}_i^T \hat{\sigma}_i' \tilde{V}_i^T x_i + w_i(t) \right) r_i + \Gamma_1 \text{tr}(\tilde{W}_i^T \dot{\tilde{W}}_i) + \Gamma_2 \text{tr}(\tilde{V}_i^T \dot{\tilde{V}}_i) \quad (4.2.21)$$

Avec:

$$\dot{V}_l(t) = \frac{-1}{2} E_i^T Q E_i + \left(\frac{K_{P_i}}{m_{ii}} - \omega_i^2 \right) \left[Q_1 \left(\frac{\dot{K}_{P_i}}{m_{ii}} - \dot{K}_{P_i}^* \right) - r_i e_i \right] - Q_1 K_{P_i}^* \left(\frac{\dot{K}_{P_i}}{m_{ii}} - \dot{K}_{P_i}^* \right) + \left(\frac{K_{V_i}}{m_{ii}} - 2\xi_i \omega_i \right) \left[Q_2 \left(\frac{\dot{K}_{V_i}}{m_{ii}} - \dot{K}_{V_i}^* \right) - r_i \dot{e}_i \right] - Q_2 K_{V_i}^* \left(\frac{\dot{K}_{V_i}}{m_{ii}} - \dot{K}_{V_i}^* \right) \quad (4.2.22)$$

Et

$$r_i = P_2 e_i + P_3 \dot{e}_i$$

Pour la suite de cette démonstration, deux cas possible se présentent:

- Le premier correspondant au cas idéal, c'est à dire le terme regroupant les erreurs d'estimation infiniment petit est négligé.
- Le deuxième est plus général et prend en considérations ces erreurs.

1°/Cas idéal :

$$w_i(t) = 0$$

Donc (4.2.21) devient:

$$\begin{aligned} \dot{V}_i(t) = \dot{V}_1(t) + \frac{1}{m_{ii}} \text{tr} \left[\tilde{W}_i^T \left(\Gamma_1 m_{ii} \dot{\tilde{W}}_i + (\hat{\sigma}_i - \hat{\sigma}'_i \hat{V}_i^T x_i) r_i \right) \right] \\ + \frac{1}{m_{ii}} \text{tr} \left[\tilde{V}_i^T \left(\Gamma_2 m_{ii} \dot{\tilde{V}}_i + x_i \hat{W}_i^T \hat{\sigma}'_i r_i \right) \right] \end{aligned} \quad (4.2.23)$$

Pour que le système régit par la représentation d'état (4.2.18) soit asymptotiquement stable, il suffit que la fonction $\dot{V}_i(t)$ soit définie négative.

Pour cela on propose:

$$\begin{aligned} Q_1 \left(\frac{\dot{K}_{P_i}}{m_{ii}} - \dot{K}_{P_i}^* \right) - r_i e_i = 0 \\ Q_2 \left(\frac{\dot{K}_{V_i}}{m_{ii}} - \dot{K}_{V_i}^* \right) - r_i \dot{e}_i = 0 \end{aligned} \quad (4.2.24)$$

$$\begin{aligned} m_{ii} \Gamma_1 \dot{\tilde{W}}_i + (\hat{\sigma}_i - \hat{\sigma}'_i \hat{V}_i^T x_i) r_i = 0 \Rightarrow \dot{\tilde{W}}_i = F_i (\hat{\sigma}_i - \hat{\sigma}'_i \hat{V}_i^T x_i) r_i \\ m_{ii} \Gamma_2 \dot{\tilde{V}}_i + x_i \hat{W}_i^T \hat{\sigma}'_i r_i = 0 \Rightarrow \dot{\tilde{V}}_i = G_i (x_i \hat{W}_i^T \hat{\sigma}'_i r_i) \end{aligned} \quad (4.2.25)$$

Avec:

$$\begin{aligned} \tilde{W}_i &= W_i - \hat{W}_i \Rightarrow \dot{\tilde{W}}_i = -\dot{\hat{W}}_i \\ \tilde{V}_i &= V_i - \hat{V}_i \Rightarrow \dot{\tilde{V}}_i = -\dot{\hat{V}}_i \\ F_i &= \frac{1}{\Gamma_1 m_{ii}} \quad \text{et} \quad G_i = \frac{1}{\Gamma_2 m_{ii}} \end{aligned}$$

D'où l'expression de $\dot{V}_i(t)$:

$$\dot{V}_i(t) = \frac{-1}{2} E_i^T Q E_i - K_{P_i}^* r_i e_i - K_{V_i}^* r_i \dot{e}_i$$

Pour avoir $\dot{V}_i(t) < 0$, on choisit:

$$\begin{aligned} K_{P_i}^*(t) &= Q_1^* r_i e_i \\ K_{V_i}^*(t) &= Q_2^* r_i \dot{e}_i \end{aligned} \tag{4.2.26}$$

Avec:

Q_1^* et Q_2^* constantes positives arbitraires.

D'où l'expression de $\dot{V}_i(t)$ devient:

$$\dot{V}_i(t) = \frac{-1}{2} E_i^T Q E_i - Q_1^* (r_i e_i)^2 - Q_2^* (r_i \dot{e}_i)^2 < 0$$

$\dot{V}_i(t)$ est négative, ainsi:

Tous les signaux $E_i(t), K_{P_i}(t), K_{V_i}(t), \hat{W}_i(t)$ et $\hat{V}_i(t)$ de l'équation (4.2.18), sont uniformément bornées. En plus, les erreurs paramétriques \tilde{W}_i et \tilde{V}_i ainsi que l'erreur de poursuite globale $E_i(t)$ convergent asymptotiquement vers zéro. Donc:

$$E_i(t) \rightarrow 0 \quad \text{quand} \quad t \rightarrow \infty$$

D'où:

$$X_i(t) \rightarrow X_{mi}(t) \quad \text{quand} \quad t \rightarrow \infty$$

Donc:

$$\begin{aligned} \lim_{t \rightarrow \infty} e_i(t) &= 0 \end{aligned}$$

En posant:

$$Q_1^* = \frac{B_1}{m_{ii}}, Q_2^* = \frac{B_2}{m_{ii}}$$

$$Q_1 = \frac{m_{ii}}{\alpha_1}, Q_2 = \frac{m_{ii}}{\alpha_2}$$

Et en remplaçant (4.2.26) dans (4.2.24) on obtient les lois d'adaptation du PD suivant:

$$\begin{cases} \dot{K}_{P_i}(t) = \alpha_1 (r_i e_i) + B_1 \frac{d}{dt} (r_i e_i) \\ \dot{K}_{V_i}(t) = \alpha_2 (r_i \dot{e}_i) + B_2 \frac{d}{dt} (r_i \dot{e}_i) \end{cases} \quad (4.2.27)$$

Et les lois d'ajustement des poids du RNA suivant:

$$\begin{cases} \dot{\hat{W}}_i(t) = F_i (\hat{\sigma}_i - \hat{\sigma}_i' \hat{V}_i^T x_i) r_i \\ \dot{\hat{V}}_i(t) = G_i (x_i \hat{W}_i^T \hat{\sigma}_i' r_i) \end{cases} \quad (4.2.28)$$

2°/Cas général:

La dérivation des lois d'adaptation pour le cas général, c'est à dire que:

$$w_i(t) \neq 0$$

Est illustré dans l'annexe (voir ANNEXE B).

Remarques importantes:

- Les lois d'adaptation des paramètres du proportionnelle dérivateur sont fort simples, l'implémentation de ceux-ci en commande digitale peut se faire par simple intégration trapézoïdale [26] ou par discrétisation directe.
- Dans le schéma de commande proposé, on aboutit aux mêmes lois d'ajustement des poids du RNA donnés par LEWIS [18]. Ceci est justifiable dans la mesure où on a pris les mêmes considérations, on ce qui concerne le bloc neuronal.
- L'idée de base d'une telle stratégie de commande est d'imposer une dynamique à l'erreur de poursuite, ce qui diffère de l'approche développée auparavant. Cependant, pour pouvoir

dérivé les lois d'adaptation on émit la supposition que les éléments de la matrice d'inertie "*varient lentement*" par rapport à la variation des signaux de commande.

- Pour pallier aux problèmes des interconnexions et des dynamiques non modélisée, plusieurs approches sont utilisées en pratique. IOANNOU [12] propose l'introduction du " σ modification" dans les lois d'adaptation afin de garantir plus de robustesse en présence de perturbation.

4.2.3. SIMULATION :

PARAMETRES DE COMMANDE :

Station 1: La station comporte un réseau de neurones de 5 entrées, 10 neurones dans la couche cachée et 1 sorties , et avec les paramètres suivants:

$$\lambda=0.1, F=3, G=0.18, k=0.3, Z_m=20, \alpha_1=100, \beta_1=30, \alpha_2=100, \beta_2=30$$

Station 2: Comporte un réseau de neurones de 5 entrées, 10 neurones dans la couche cachée et 1 sorties , et avec les paramètres suivants:

$$\lambda=1.5, F=5, G=1.05, k=0.015, Z_m=20, \alpha_1=100, \beta_1=30, \alpha_2=100, \beta_2=30$$

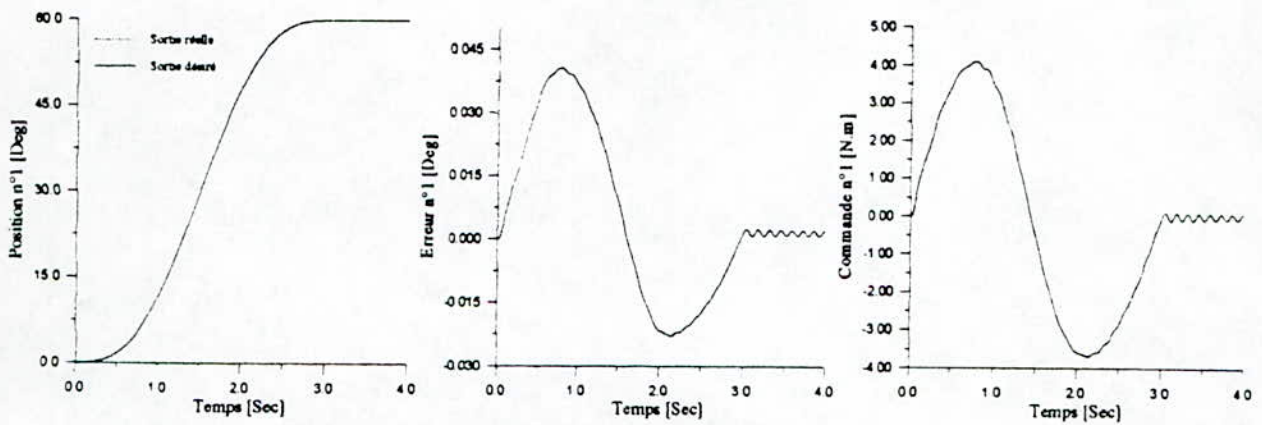
Station 3: Comporte un réseau de neurones de 5 entrées, 10 neurones dans la couche cachée et 1 sorties , et avec les paramètres suivants:

$$\lambda=5, F=1, G=1.05, k=0.015, Z_m=20, \alpha_1=100, \beta_1=30, \alpha_2=100, \beta_2=30$$

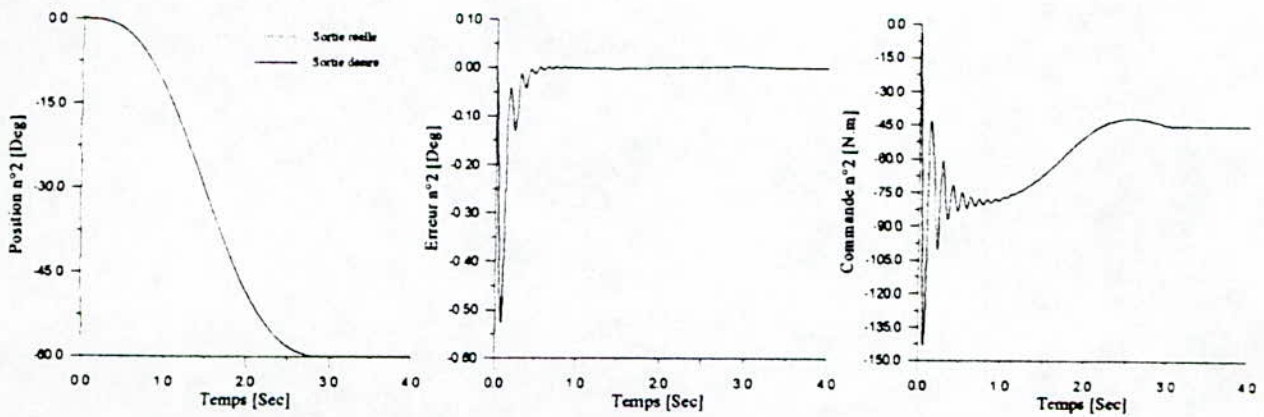
INTERPRETATION DES RESULTATS :

- **Test de validation :** La figure (4.2.2) nous permet de constater une similitude avec les réponses obtenus dans l'approche précédente. ainsi, les erreurs s'annulent (négligeant les petits valeurs) après des pics d'erreurs négligeables. La figure (4.2.3) montre la détérioration des résultats en absence des RNA. Ce qui permet de mettre en évidence leurs rôles de compensation des interconnexions due au couple de coriolis et de gravitation. Ainsi, avec ces test on peut valider l'algorithme de la commande proposée dans ce présent chapitre.
- **Variation massique subite :** Les figures (4.2.3) et (4.2.4) montrent la stabilité du système malgré le fait que la condition imposée sur la variation des éléments de la matrice d'inertie, étant donné la variation brusque de la masse de charge. Cela pourrait s'expliquer par le fait que la théorie de la stabilité de LYPUNOV nous permet d'avoir des conditions suffisantes et non nécessaires de la stabilité. Une observation attentionnée permet de remarquer que la commande répond d'une manière instantanée au variation de la charge. ceci est simple à expliquer, car la commande est de type adaptative.

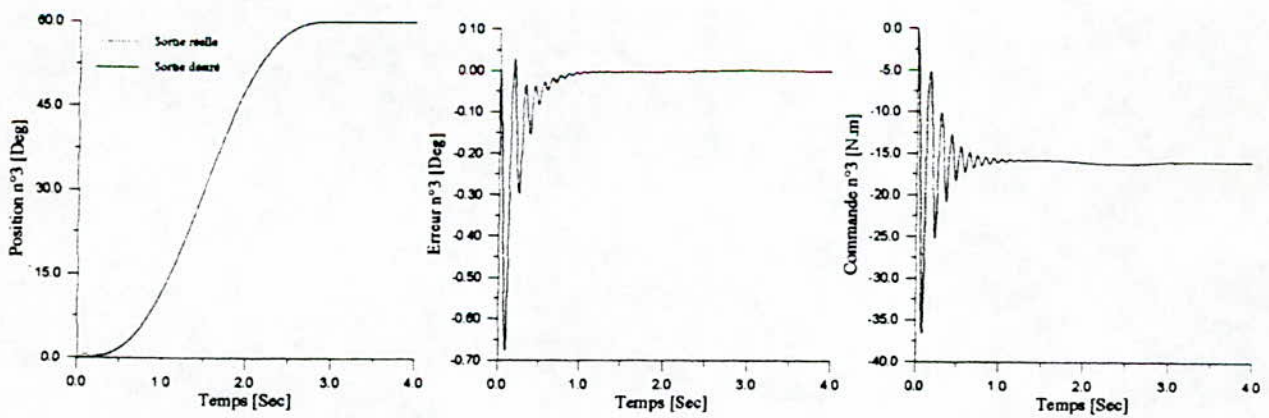
- **Rupture de la commande :** Comme pour les résultats de la première approche, les figures (4.2.6), (4.2.7) et (4.2.8) nous permettent de constater que la décentralisation de la commande a d'énormes avantages, car une panne quelconque sur une articulation n'affecte pas beaucoup les réponses des autres articulations.
- **Test de LEAHVY:** Malgré l'imposition de trajectoires très rapides qui excitent toute la dynamique du PUMA 560, nous constatons d'après la figure (4.2.9) que la commande répond efficacement et assure une poursuite parfaite des trajectoires de référence. La figure (4.2.10) montre la détérioration des résultats en absence des RNA sous le même test, comparant les résultats on remarque rapidement l'effet des RNA dans la commande.



Position, Erreur et Commande de la première articulation.

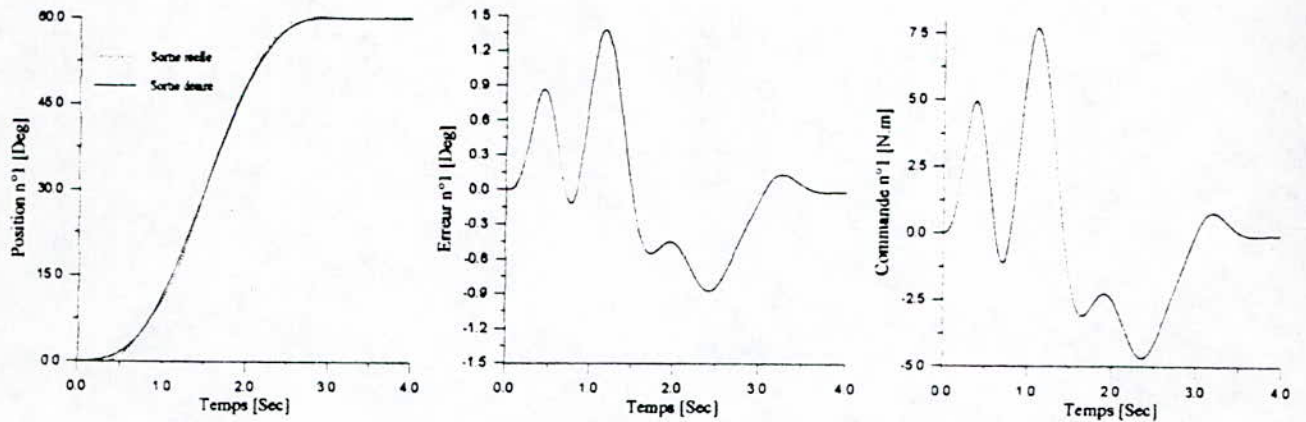


Position, Erreur et Commande de la deuxième articulation.

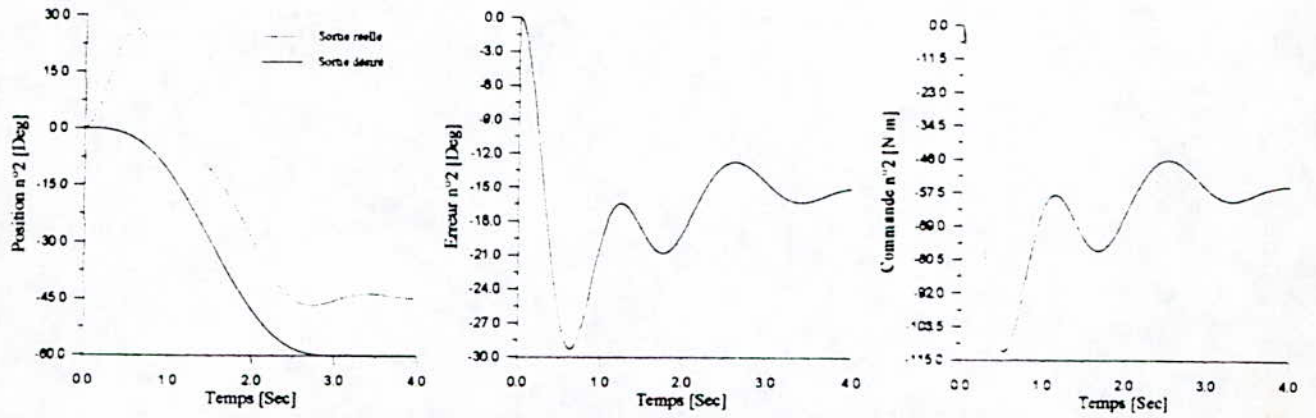


Position, Erreur et Commande de la troisième articulation.

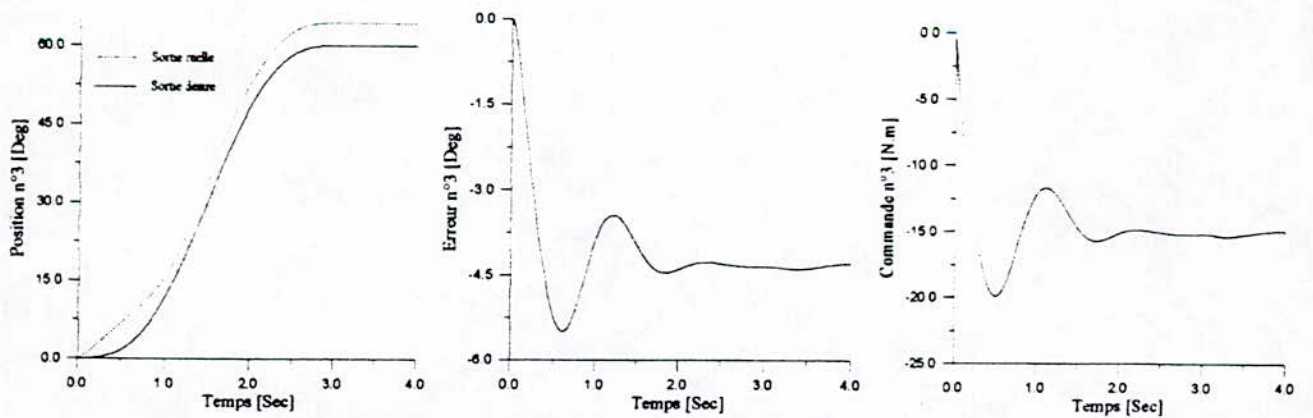
Figure 4.2.2
Réponses en boucle fermée sans charge (à vide).



Position, Erreur et Commande de la première articulation.

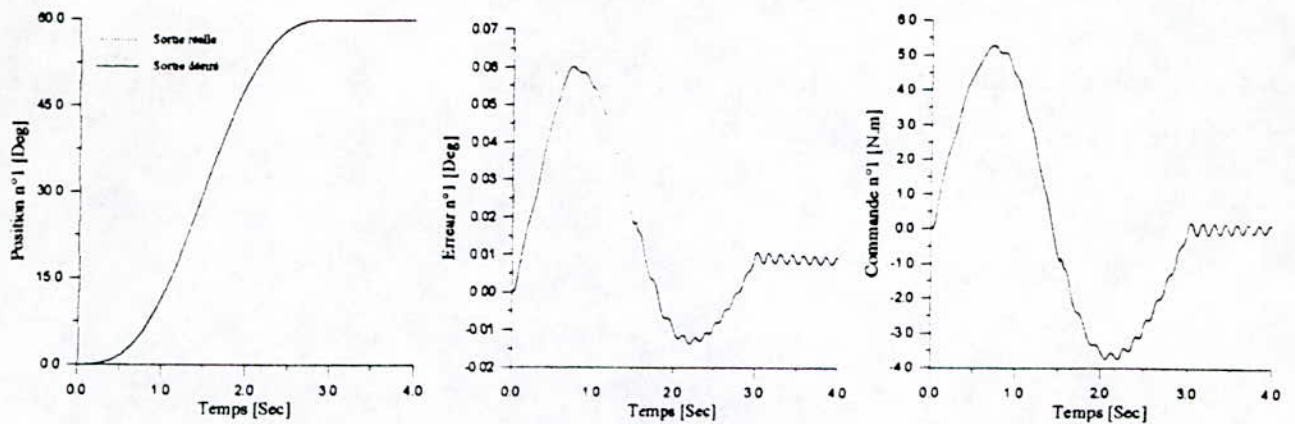


Position, Erreur et Commande de la deuxième articulation.

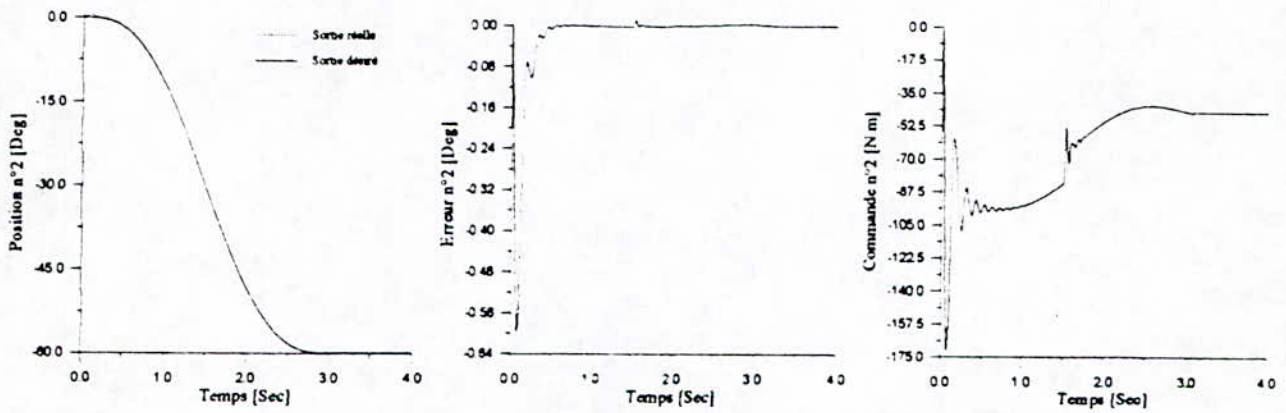


Position, Erreur et Commande de la troisième articulation.

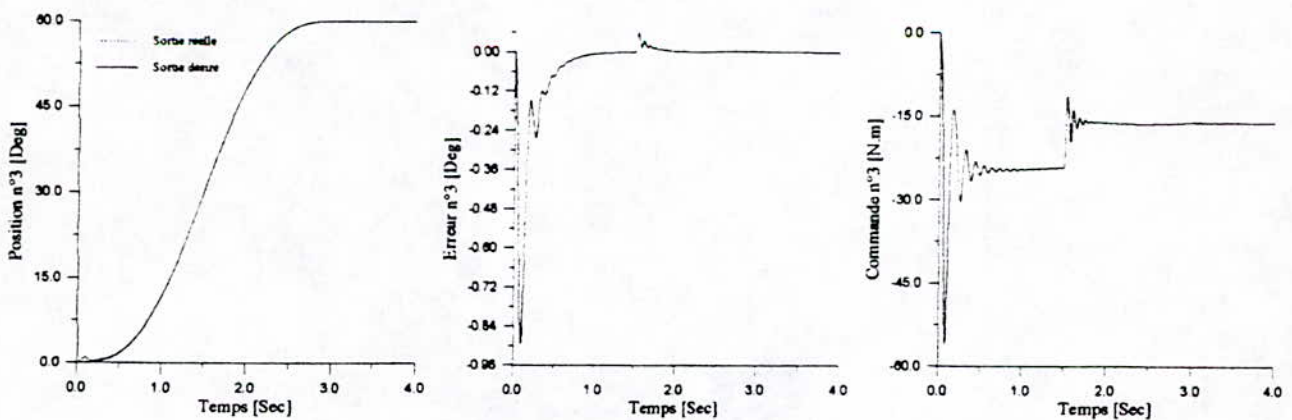
Figure 4.2.3
Réponses en boucle fermée avec le P.D. fixe (à vide).



Position, Erreur et Commande de la première articulation.

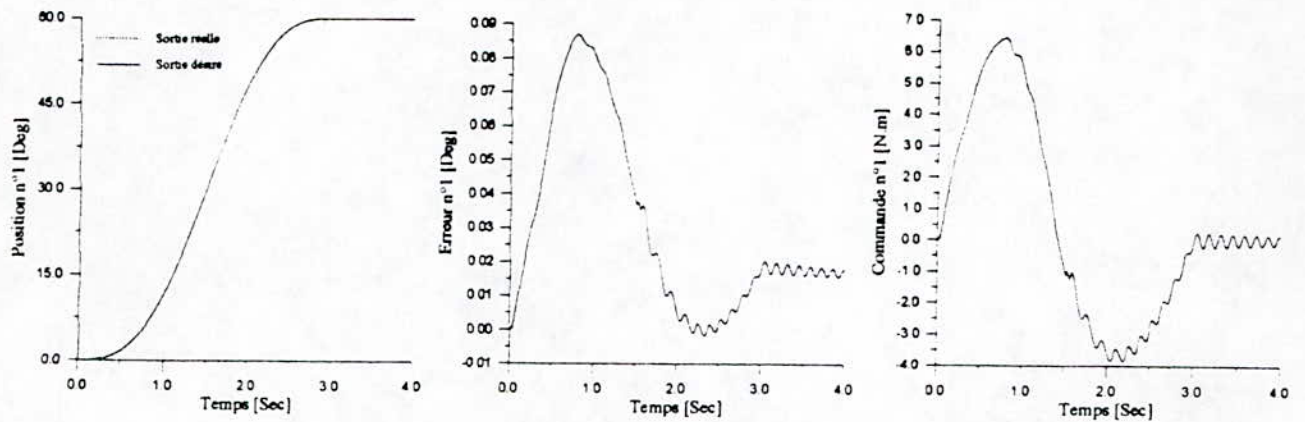


Position, Erreur et Commande de la deuxième articulation.

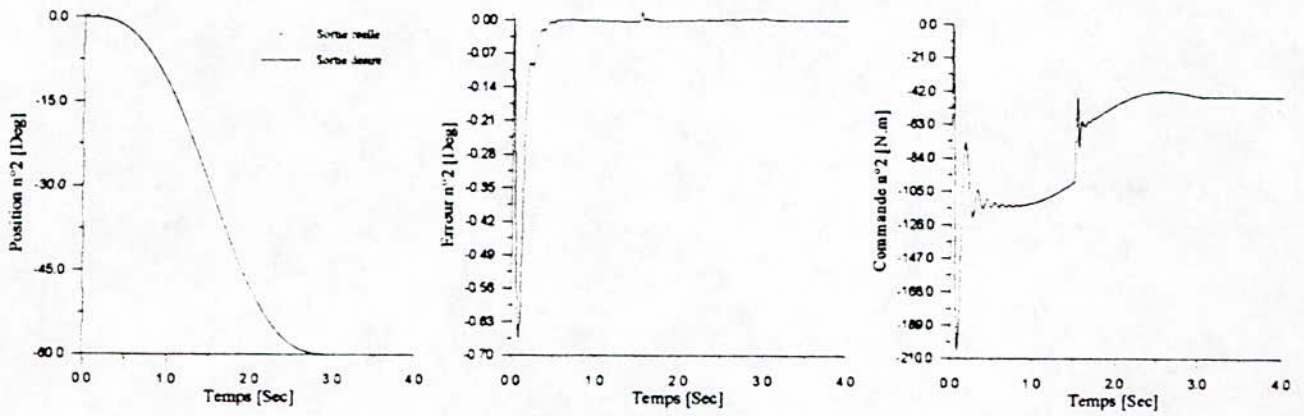


Position, Erreur et Commande de la troisième articulation.

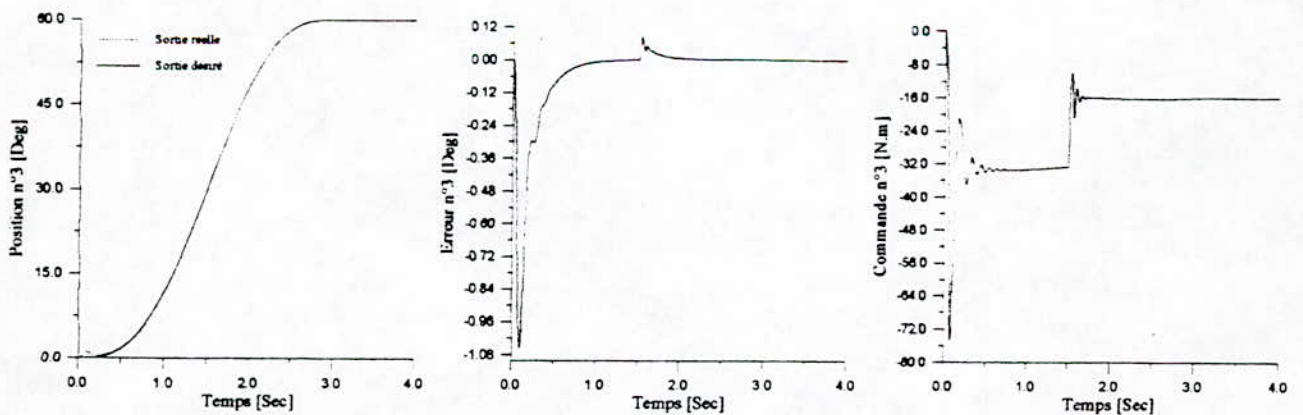
Figure 4.2.4
Réponses en boucle fermée avec une masse de 2 Kg (Lâchée à l'instant $t=1.5s$).



Position, Erreur et Commande de la première articulation.

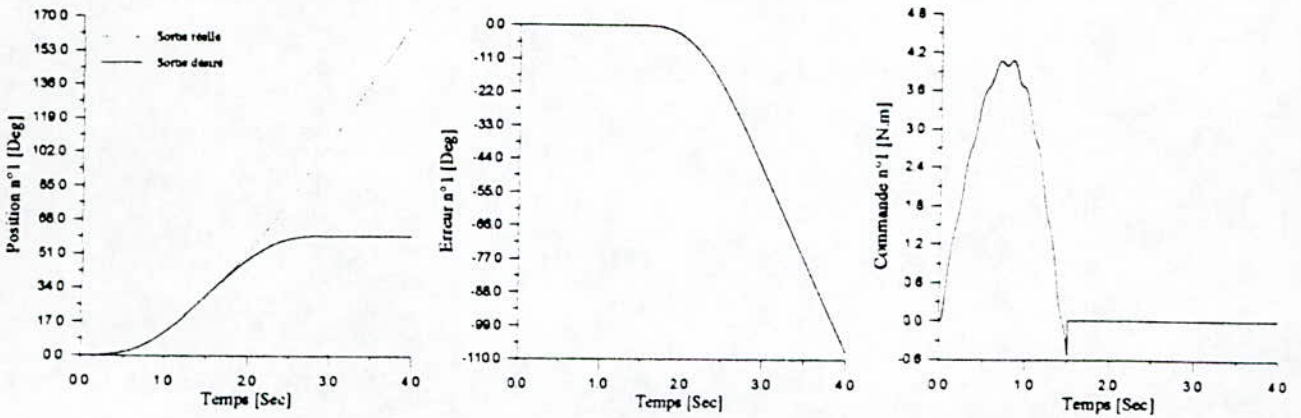


Position, Erreur et Commande de la deuxième articulation.

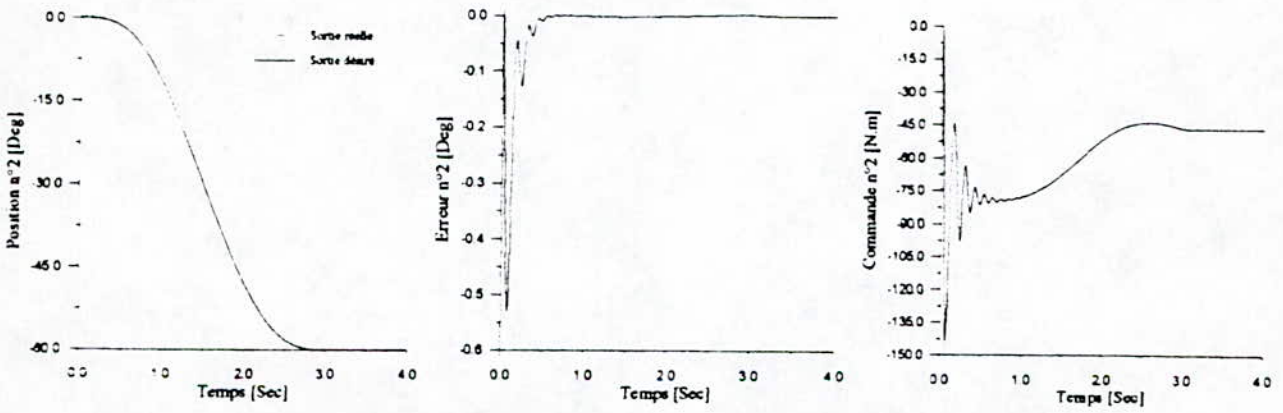


Position, Erreur et Commande de la troisième articulation.

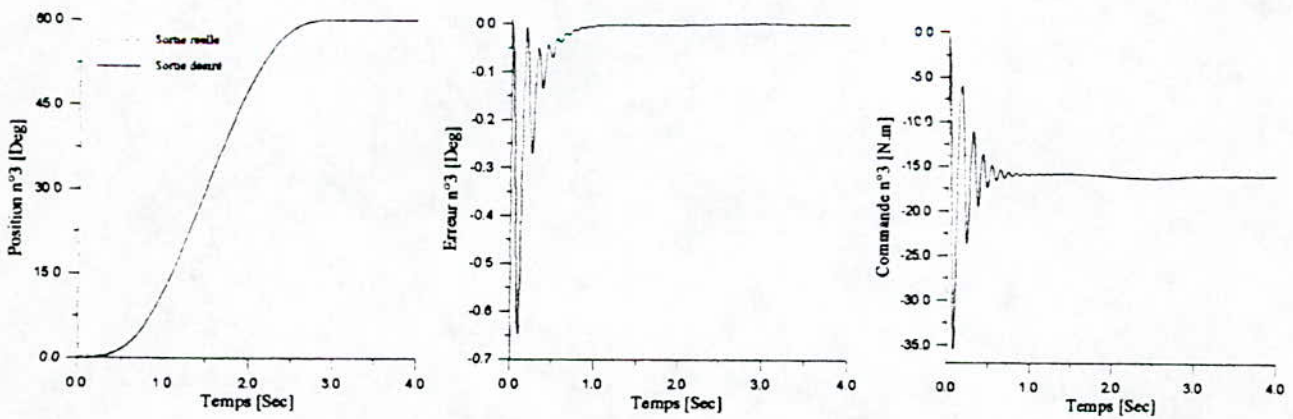
Figure 4.2.5
Réponses en boucle fermée avec une masse de 4 Kg (lâchée à l'instant $t=1.5s$).



Position, Erreur et Commande de la première articulation.

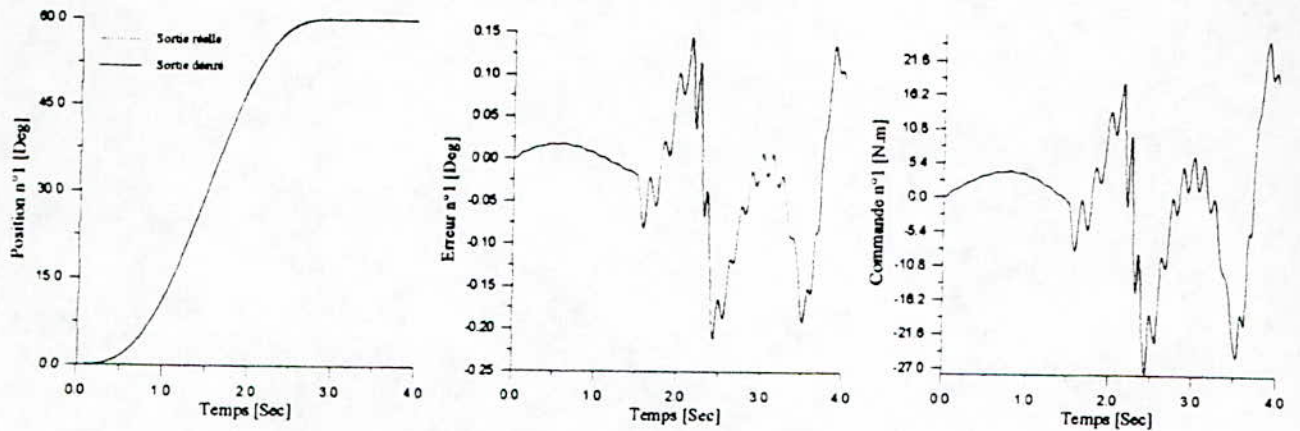


Position, Erreur et Commande de la deuxième articulation.

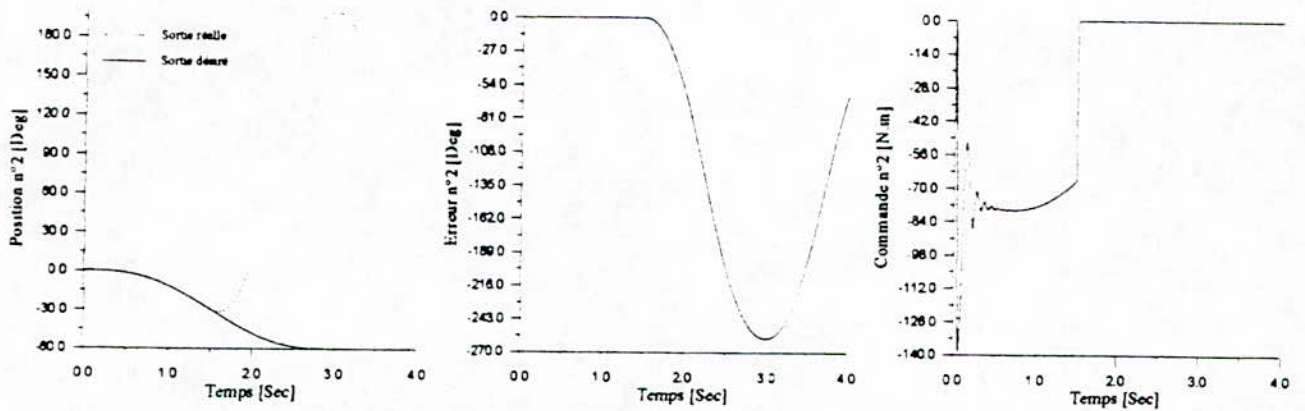


Position, Erreur et Commande de la troisième articulation.

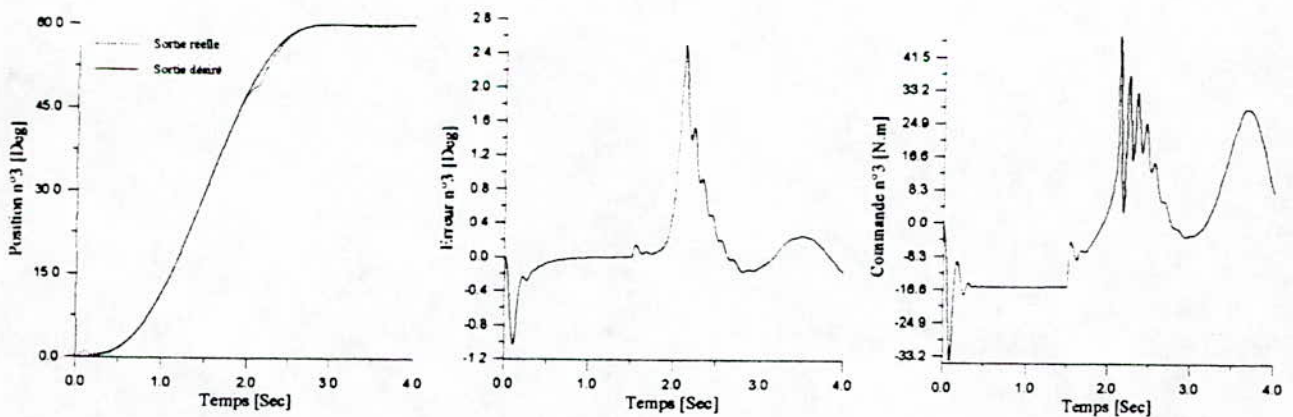
Figure 4.1.6
Réponses en boucle fermée avec rupture de la commande, $u(1)=0$ à $t=1.5s$.



Position, Erreur et Commande de la première articulation.

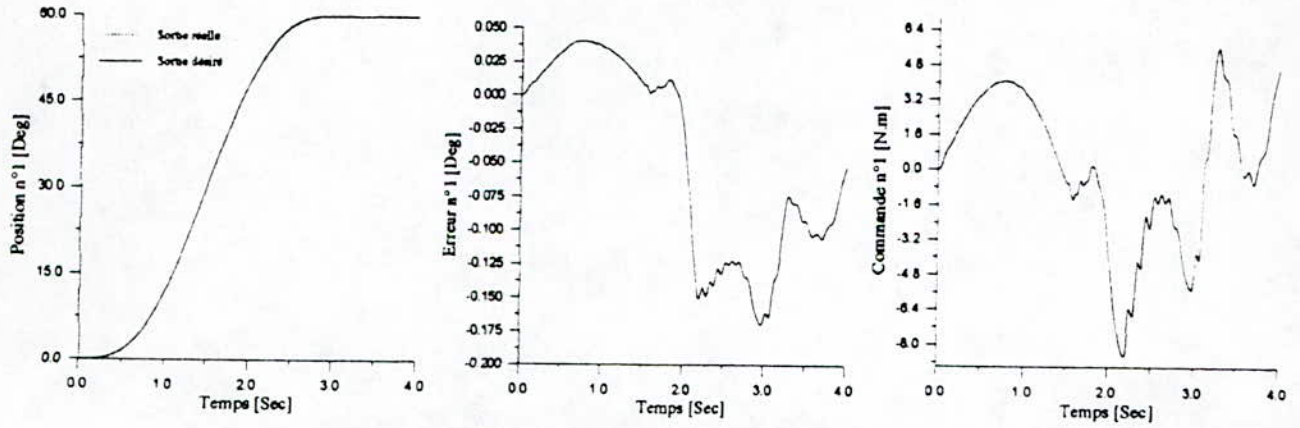


Position, Erreur et Commande de la deuxième articulation.

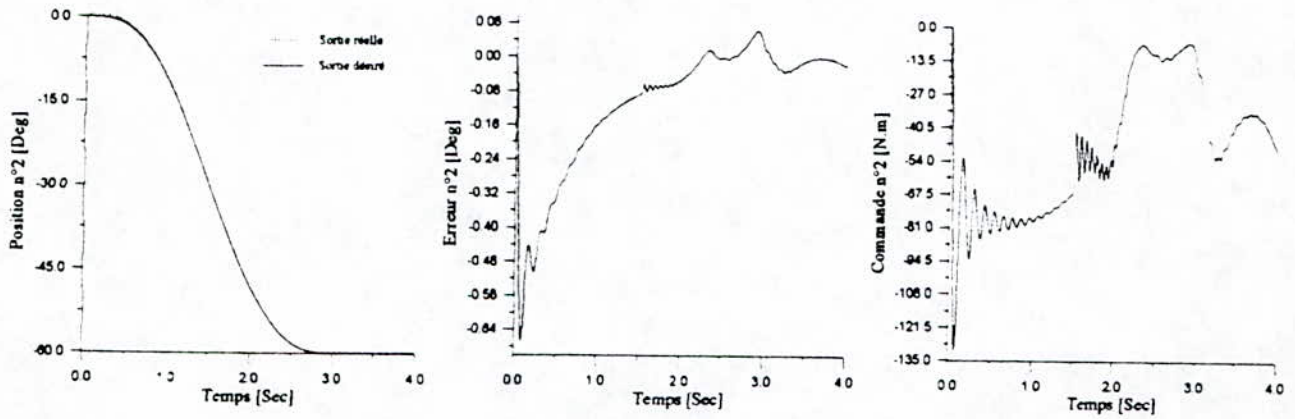


Position, Erreur et Commande de la troisième articulation.

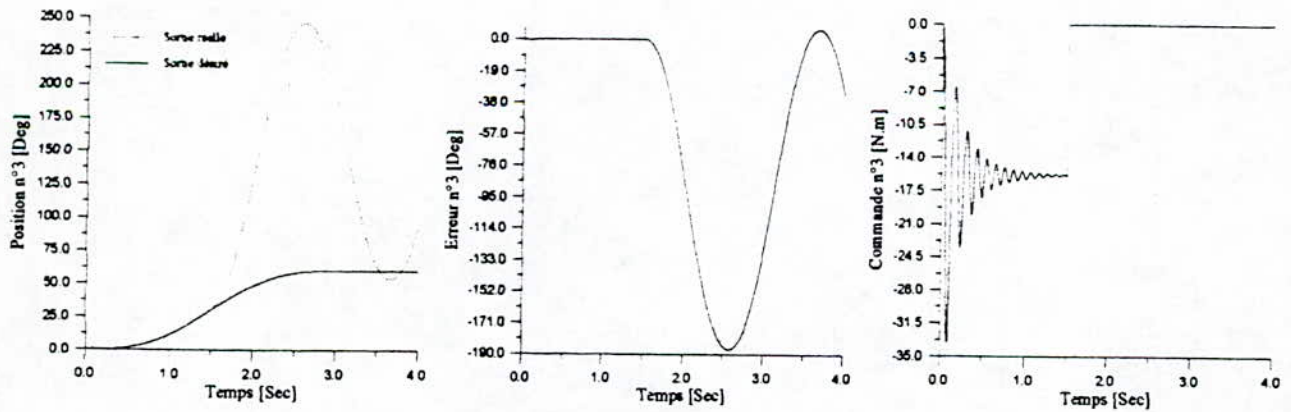
Figure 4.2.8
Réponses en boucle fermée avec rupture de la commande, $u(2)=0$ à $t=1.5s$.



Position, Erreur et Commande de la première articulation.

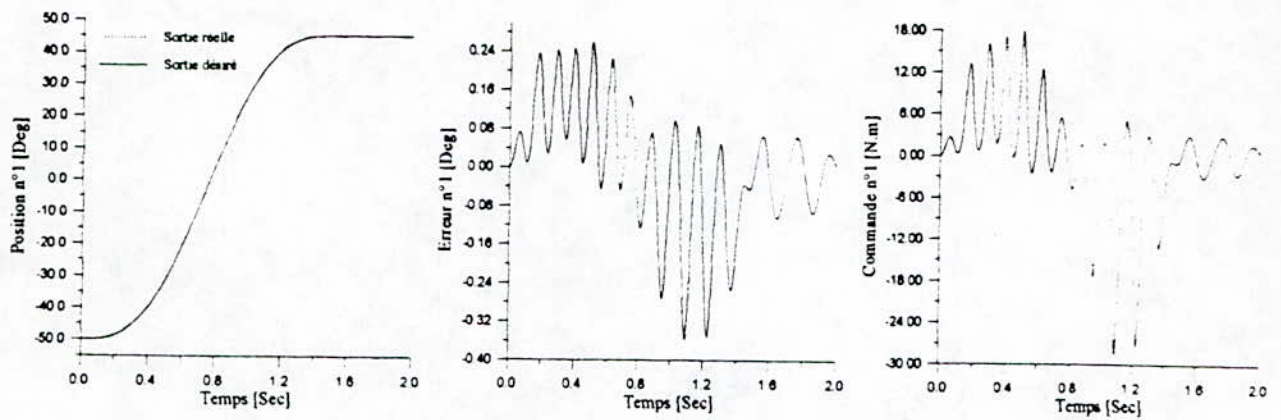


Position, Erreur et Commande de la deuxième articulation.

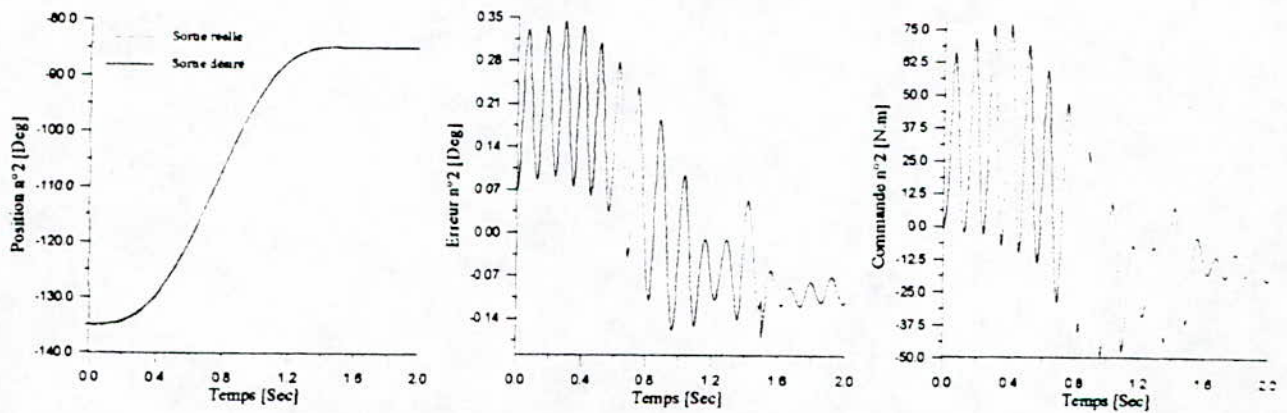


Position, Erreur et Commande de la troisième articulation.

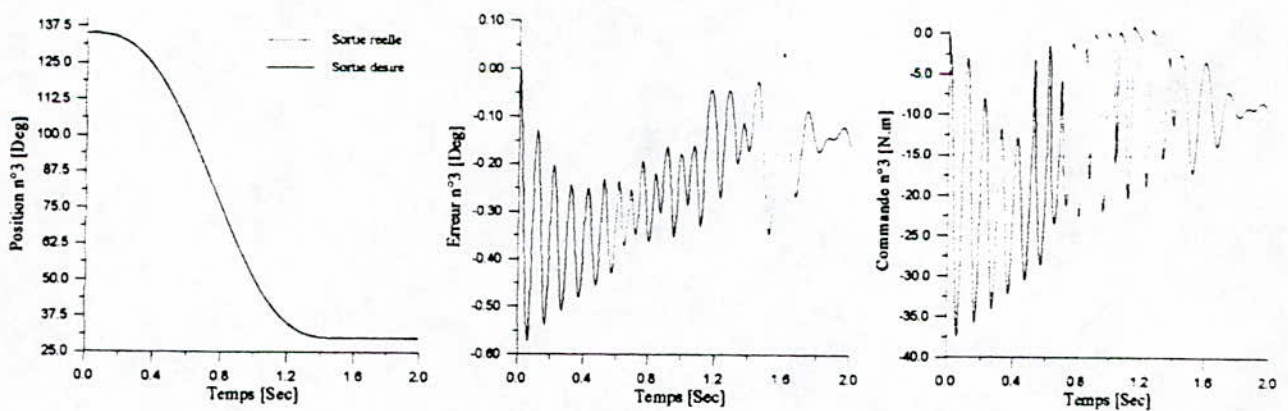
Figure 4.2.8
Réponses en boucle fermée avec rupture de la comande, $u(3)=0$ à $t=1.5s$.



Position, Erreur et Commande de la première articulation.

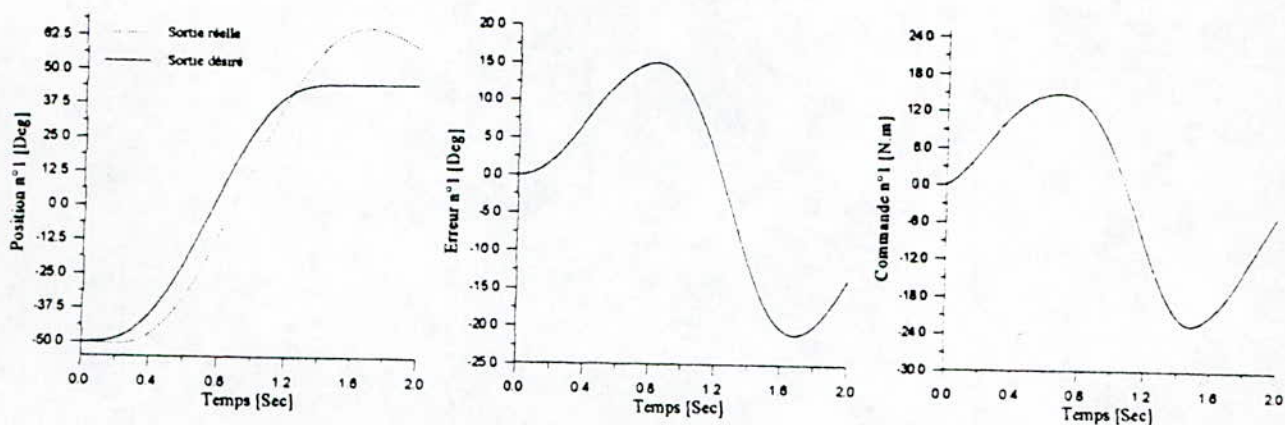


Position, Erreur et Commande de la deuxième articulation.

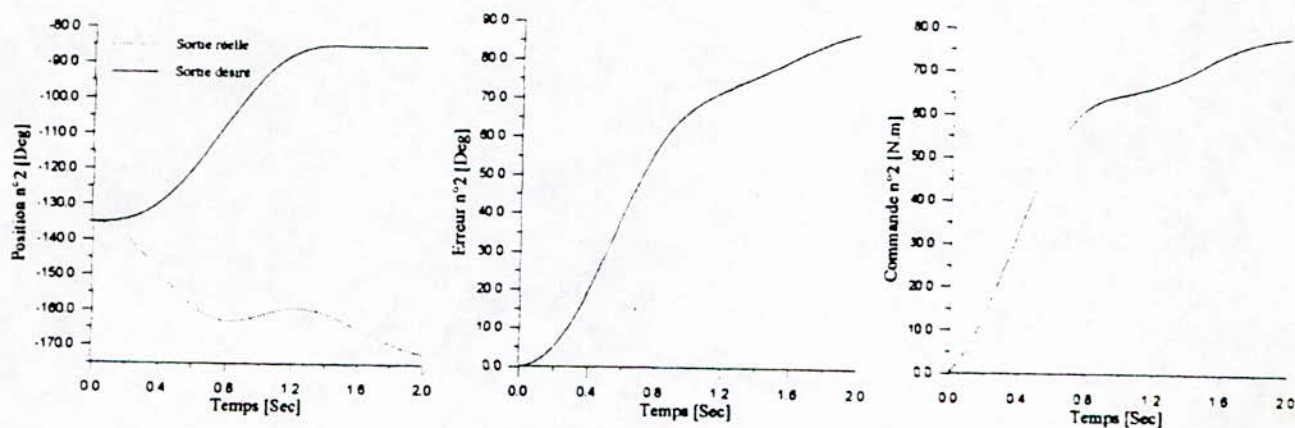


Position, Erreur et Commande de la troisième articulation.

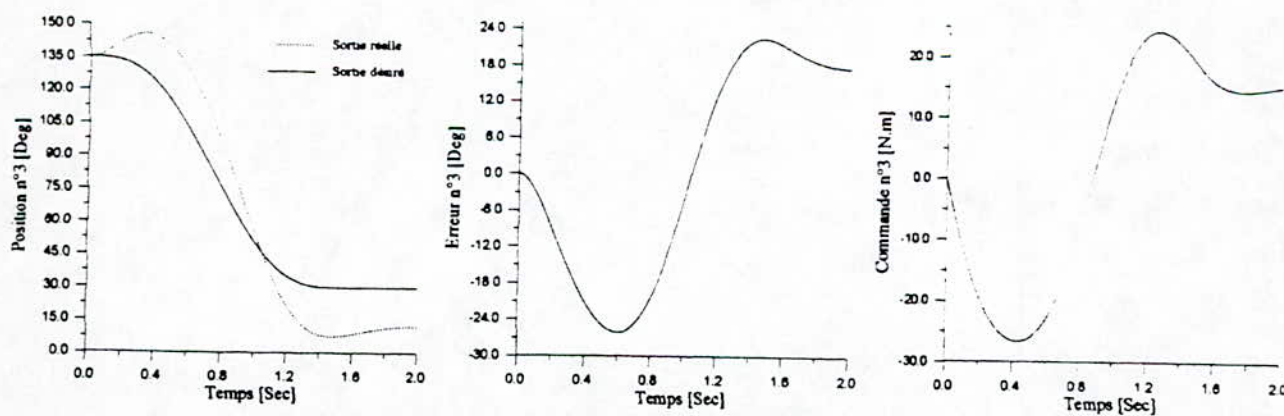
Figure 4.2.9
Réponses en boucle fermée sous le test de LEAHVY.



Position, Erreur et Commande de la première articulation.



Position, Erreur et Commande de la deuxième articulation.



Position, Erreur et Commande de la troisième articulation.

Figure 4.2.10
Réponses en boucle fermées sous le test de LEAHVY avec un PD fixe.

CONCLUSION

Dans ce chapitre, nous avons développés deux nouvelles stratégies de commande adaptative décentralisée par réseaux de neurones artificiels appliquées en robotique.

Dans un premier temps, nous avons étendu l'approche adaptative centralisée par RNA présenter au chapitre précédent à la structure décentralisée. Dans la seconde partie, nous avons conçus une nouvelle technique de commande adaptative décentralisée par RNA en empruntant un chemin différent afin de dériver les lois d'ajustement des paramètres du régulateur adaptatif neuronal.

A travers les différentes simulations effectuées sur le PUMA 560, nous avons pu mettre en évidence les performances et l'efficacité de ces deux techniques de commande adaptative décentralisée par RNA en termes de poursuites des trajectoires de références et de robustesse paramétrique. il est à noter également, que dans le cas de première approche, nous constatons de meilleurs performances. Cependant, la deuxième approche offre plus de souplesse et de simplicité lors de l'implémentation. Toutefois, et dans l'ensemble, le rapport performance / simplicité est toujours vérifié.



CONCLUSION GENERALE

CONCLUSION GENERALE

Dans ce mémoire, nous avons été amenés à étudier une nouvelle approche adaptative par réseaux de neurones appliquée en robotique, basée sur l'ajustement en temps réel des paramètres de commande, selon des algorithmes d'adaptation issus de la théorie de la stabilité non-linéaire. Cette théorie développée dans le cas centralisé appliquée sur un bras de robot manipulateur en l'occurrence le *PUMA 560*, montre des performances satisfaisantes. Par contre, elle nécessite d'assez puissants moyens de calcul, due principalement :

- A la taille et aux liaisons entre les différentes grandeurs du modèle.
- Dimensionnement du réseau de neurones.
- Difficultés d'un choix adéquat des paramètres de commande (gains et pondérations)

Vu les inconvénients que présente cette approche, une commande à caractère décentralisée présente une solution appropriée. Ainsi, cette technique a été étendue à la structure décentralisée en concevant plusieurs régulateurs neuronaux dédiés à chaque articulation.

Et dans le même contexte, mais suivant une analyse différente, nous avons conçu une nouvelle commande adaptative décentralisée par réseaux de neurones dont la structure diffère par l'introduction d'un régulateur de type proportionnelle-dérivateur à gain ajustable, sans autant utilisé le signal robuste. Ces deux méthodes ont été appliquées sur le même bras de robot manipulateur.

Afin d'évaluer les performances de ces deux commandes, nous avons établis leurs robustesses par rapport à la variation de la masse de la charge, à la rupture de la commande d'une articulation et aux trajectoires rapides (test de Leahvy). Les résultats obtenus sont très satisfaisants. En effet, l'ensemble de ces commandes montre des performances remarquables et assure une poursuite parfaite sous des conditions de travail hostiles (une charge d'une masse allant jusqu'à 80% de la troisième articulation).

Jusqu'à présent, les réseaux de neurones avec leurs énormes capacités d'estimation et de généralisation ont été utilisés efficacement dans les autres approches neuronales. Cependant, aucune preuve de stabilité n'a été avancée. Par contre, les algorithmes d'adaptation développés dans notre cas assurent une stabilité du système globale dont la preuve a été établie selon la théorie de la stabilité de LYAPUNOV.

Toutefois, il faut signaler l'importance de la structure décentralisée apportée dans ce mémoire, car la commande centralisée, contrairement aux commandes décentralisées, soumise aux mêmes tests présente une dégradation des performances lorsqu'on applique une rupture d'une commande. Ceci est prévu, car la détérioration d'une articulation induit immédiatement à des commandes qui n'évoluent plus dans les normes (forte oscillation). Par contre, en commande décentralisée, le traitement local des informations permet de rendre la synthèse de la commande de chaque articulation indépendante des autres, malgré l'effet important de ces derniers sur la dynamique de l'articulation commandée, d'où l'intérêt d'une telle stratégie.

Cependant, il est à noter, que dans le cas de la première approche décentralisée, les performances sont meilleures, toutefois, la deuxième approche offre plus de souplesse et de simplicité lors de l'implémentation. Alors le rapport performances simplicité est largement optimisé.

Vu le caractère particulier des ces méthodes présenter qui se restreint à la robotique, il s'avérait judicieux de généraliser ces méthodes, permettant ainsi, une application sur l'ensemble des systèmes dynamiques, il nous semble intéressant d'investiguer:

- Les réseaux électriques.
- Les processus industriels.
- Les services des télécommunications, etc. .

Les méthodes proposées sont basées sur la compensation de certaines dynamiques non-linéaires qui caractérisent les manipulateurs. Cependant, nous pensons qu'il serait intéressant de tester, dans ce contexte et suivant la même technique de dérivation des algorithmes d'adaptations, d'autres approches afin d'effectuer une étude comparative telle que:

- La commande adaptative par le modèle de référence (*MRAC*).
- Le bouclage linéarisant (*FEEDBACK LINEARIZATION*).
- La commande adaptative indirecte.
- La commande inverse.



ANNEXES ET BIBLIOGRAPHIE

ANNEXE A

PARAMETRES DU PUMA 560 [2][5]• PARAMETRES GEOMETRIQUE ET MASSES DES LIAISONS :

$$d_2 = 146.09 \text{ mm}$$

$$L_2 = 431.09 \text{ mm}$$

$$L_3 = 433.07 \text{ mm}$$

$$m_2 = 17.40 \text{ Kg}$$

$$m_3 = 5.04 \text{ Kg}$$

$$m_4 = 0.82 \text{ Kg}$$

$$m_5 = 0.35 \text{ Kg}$$

$$m_6 = 0.09 \text{ Kg}$$

$$m_l = 1.26 \text{ Kg}$$

• INERTIE DES DIFFERENTES LIAISONS :

N° de la liaison	I_{xxi} (Kg.m ²)	I_{yyi} (Kg.m ²)	I_{zz_i} (Kg.m ²)	Inertie des moteurs I_{mi} (Kg.m ²)
1	–	0.35	–	1.14
2	0.130	0.524	0.539	4.71
3	0.192	0.0154	0.212	0.83
4	1.3 E-3	1.8 E-3	1.8 E-3	–
5	0.3 E-3	0.3 E-3	0.4 E-3	–
6	0.04 E-3	0.15 E-3	0.15 E-3	–
Effecteur	1.64 E-3	2.24 E-3	2.35 E-3	–

ANNEXE B

PREUVE DE LA STABILITE DANS LE CAS GENERALE

Dans le cas ou $w_i(t) \neq 0$, l'équation (4.2.23) devient :

$$\begin{aligned} \dot{V}_i(t) = \dot{V}_1(t) + \frac{1}{m_{ii}} \text{tr} \left[\tilde{W}_i^T \left(\Gamma_1 m_{ii} \dot{\tilde{W}}_i + (\hat{\sigma}_i - \hat{\sigma}_i' \hat{V}_i^T x_i) r_i \right) \right] \\ + \frac{1}{m_{ii}} \text{tr} \left[\tilde{V}_i^T \left(\Gamma_2 m_{ii} \dot{\tilde{V}}_i + x_i \hat{W}_i^T \hat{\sigma}_i' r_i \right) \right] + r_i \cdot w_i \end{aligned} \quad (\text{B.1})$$

Et si on choisit les lois d'adaptation des poids \hat{W}_i et \hat{V}_i posé comme suite :

$$\begin{aligned} \dot{\hat{W}}_i(t) = F_i \left(\hat{\sigma}_i - \hat{\sigma}_i' \hat{V}_i^T x_i \right) r_i - k_i \|r_i\| \cdot \hat{W}_i \\ \dot{\hat{V}}_i(t) = G_i \left(x_i \hat{W}_i^T \hat{\sigma}_i' r_i \right) - k_i \|r_i\| \cdot \hat{V}_i \end{aligned} \quad (\text{B.2})$$

Avec

$$Z_i = [W_i \quad V_i]$$

L'équation (B.1) devient :

$$\dot{V}_i(t) = \dot{V}_1(t) + \frac{k_i \|r_i\|}{m_{ii}} \left[\text{tr} \left[\tilde{W}_i^T (W_i - \tilde{W}_i) \right] + \text{tr} \left[\tilde{V}_i^T (V_i - \tilde{V}_i) \right] \right] + r_i w_i \quad (\text{B.3})$$

Et pour que $\dot{V}_i(t) < 0$ soit défini négativement, il faut que :

$$\begin{aligned} Q_0 \left(\frac{\dot{K}_{Pi}}{m_{ii}} - \dot{K}_{Pi}^* \right) - r_i e_i = 0 \\ Q_1 \left(\frac{\dot{K}_{Vi}}{m_{ii}} - \dot{K}_{Vi}^* \right) - r_i \dot{e}_i = 0 \end{aligned} \quad (\text{B.4})$$

Et si on choisit:

$$\begin{aligned} K_{Pi}^*(t) = Q_0^* r_i e_i (\|\hat{Z}_i\|_F + Z_{Mi}) \\ K_{Vi}^*(t) = Q_1^* r_i \dot{e}_i \end{aligned} \quad (\text{B.5})$$

Q_0^* et Q_1^* constantes positives arbitraires.

Si on pose : $Kz_i = 2Q_0^* e_i^2$ $k_i^* = k_i/m_{ii}$, alors :

$$\dot{V}_i(t) = -\frac{1}{2} E_i^T Q E_i - Kz_i (\|\tilde{Z}_i\|_F + Z_{Mi}) - 2Q_l^* (r\dot{e})^2 + k^* \|r_i\| \text{tr}(\tilde{Z}_i^T (Z_i - \tilde{Z}_i)) + r_i w_i \quad (\text{B.6})$$

et après les propriété 1 et 2, on a :

$$\text{tr}(\tilde{Z}_i^* (Z_i - \tilde{Z}_i)) \leq \|\tilde{Z}_i\|_F \|Z_i\|_F - \|\tilde{Z}_i\|_F^2$$

$$r_i w_i \leq \|r_i\| \cdot \|w_i\|$$

$$\text{et } \|r_i\| \cdot \|w_i\| \leq \|r_i\| (c_{0i} + c_{li} \|\tilde{Z}_i\|_F + \|\tilde{Z}_i\|_F \|r_i\|)$$

Donc, combinant le tout dans l'équation (4.2.34), on a :

$$\begin{aligned} \dot{V}_i(t) \leq & -\frac{1}{2} E_i^T Q E_i - 2Q_l^* (r\dot{e})^2 + k^* \|r_i\| \cdot \|\tilde{Z}_i\|_F (Z_{Mi} - \|\tilde{Z}_i\|_F) \\ & - Kz_i (\|\hat{Z}_i\|_F + Z_{Mi}) \cdot \|r_i\|^2 + \|r_i\| \cdot \|w_i\| \quad (\text{B.7}) \end{aligned}$$

$$\Leftrightarrow \dot{V}_i(t) \leq -\frac{1}{2} E_i^T Q E_i - 2Q_l^* (r\dot{e})^2 - \|r_i\| \cdot \left[k_i^* \|\tilde{Z}_i\|_F (\|\tilde{Z}_i\|_F - Z_{Mi}) - c_{0i} - c_{li} \|\tilde{Z}_i\|_F \right] \quad (\text{B.8})$$

Si

$$\|\tilde{Z}_i\|_F > \frac{c_{3i}}{2} + \sqrt{\frac{c_{3i}^2}{4} + \frac{c_{0i}}{k_0^*}} \quad \text{avec } c_{3i} = Z_{Mi} + \frac{c_{li}}{k_i^*}$$

La fonction de LYAPUNOV reste négative d'ou, tout les signaux de l'équation (4.2.18) sont uniformément bornées. En plus, l'erreur de poursuite $e_i(t)$ converge asymptotiquement vers zéro.

ANNEXE C

Pour un bras de robot à n -degrés de liberté. On établie ce tableau comparatif, afin de montrer les avantages de la commande décentralisée par RNA :

Paramètres de commande		Commande adaptative centralisée par RNA	Commande adaptative décentralisée par RNA	
RNA	Nombre de RNA	1	n	n
	Nombre d'entrée	5.n	5	5
	Nombre de sortie	10	1	1
	Nombre de neurone *	10	10	10
	Nombre de paramètre	n^2+n^2+1	3	3
Gain du P.D.		n^2	1	—
Pondération		n^2	1	2
Paramètres du terme robuste		2	2	—

N.B. : * Nomd्रे de neurones dans la couche cahées du réseau statique utilisé.

REFERENCES BIBLIOGRAPHIQUES

- [1] **K.J. ASTROM, B. WITTENMARK;** *Adaptive control*, Addison-Wesley Publishing Company, New-York, 1989.
- [2] **B. AMSRONG, O. KHATIB, and J. BURDICK,** « *The explicit dynamic model and inertial parameters of the PUMA 560 arm* », Proc. IEEE International Conference on Robotics and Automation, pp 510-518, San Francisco 1986.
- [3] **N. BALI,** *Etude des performances de la commande prédictive généralisée: Appliquée aux robots manipulateurs PUMA et SCARA*, Thèse de Magister, ENP 1995.
- [4] **O. BOUHALI,** *Commande adaptative à structure décentralisée par RNA: Appliquée à un bras de robot manipulateur*, PFE, ENP 1996.
- [5] **H. BOUKARI, M.S. NEDJARI,** *Commande adaptative décentralisée: Application en robotique*, PFE, ENP 1996.
- [6] **F.C CHEN,** « *Backpropagation neural networks for nonlinear self-tuning adaptive control* », IEEE Control Systems Magazine, vol. 10, pp 44-48, 1990.
- [7] **J.A. FREEMAN, D.M. SKAPURA,** *Neural network: Algorithms, Applications and Programming*, Addison-Wesley Publishing Company, New York, 1992.
- [8] **K.S. FU, A.C. GONZALEZ, C.S.G. LEE,** *Robotics control, sensing, vision and intelligence*, Mc Graw-Hill Book Company.
- [9] **M. GAREH,** *Sur la commande adaptative décentralisée des systèmes interconnectés*, PFE, ENP 1995.
- [10] **G.C. GOODWIN, K.S. SIN,** *Adaptive filtering prediction and control*, Prentice-Hall, Englewood Cliffs, New Jersey, 1984.
- [11] **K.J HUNT, D. SBARBARO, R. ZBIKWOSKI, P.J. GAWTHROP,** « *Neural networks for control systems: A survey* », Automatica, vol. 28, pp 1083-1112, 1992.
- [12] **P.A. IOANNOU,** « *Decentralized adaptive control of interconnected systems* », IEEE Trans. Automatic Control, vol. AC-31, no. 4, April 1986.
- [13] **A. KARAKASOGLU, S. SUDHARSANAN, M.K. SUNDARESHAN,** « *Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators* », IEEE Trans. on Neural Networks, vol. 4, no. 6, Nov. 1993.
- [14] **N.B. KARAYIANNIS, A.N. VENETSANTANOPOULOS,** *Artificial neural networks: Learning algorithms, Performance, Evaluation and Application*, Kluwer Academic Press, London 1993.

-
- [15] **S.Y. KUNG, J.N. HWANG**, « *Neural networks architectures for robotics applications* », IEEE Trans. on Robotics and Automation, vol. 5, pp 641-657, 1989.
- [16] **I.D. LANDAU**, *Adaptive control: The model reference approach*, Marcel Dekker, New York 1979.
- [17] **M.B. LEAHVY Jr.**, « *Model based auxiliary input control: Development and experimental analysis* », Proceedings of the 29th Conf. on Decision and Control Honolulu, Hawaii Dec. 1990.
- [18] **F.L. LEWIS, A. YESILDIREK, K. LIU**, « *Multilayer neural net robot controller with guaranteed tracking performance* », Int. Proc. 32nd IEEE Conference on Decision and Control, San Antonio TX, pp 2785-2791, 1993.
- [19] **W.T. MILLER, R.S. SUTTON, P.J. WERBOS**, *Neural networks for control*, MIT Press, Cambridge, MA, 1990.
- [20] **K.S. NARENDRA, L.S. VALAVANI**, « *Stable adaptive controller design - Direct control -* », IEEE Trans. on Automatic control, vol. AC-23, August 1978.
- [21] **K.S. NARENDRA, A.M. ANNASWAMY**, « *A new adaptive law for robust adaptation without persistent excitation* », IEEE Trans. Automatic Control, vol. AC-32, no. 2, Feb. 1987.
- [22] **K.S. NARENDRA, K. PARTHASARTHY**, « *Identification and control of dynamical systems using neural networks* », IEEE Trans. on Neural Networks, vol. 1, pp 4-27, March 1990.
- [23] **T. OZAKI, T. SUZUKI, T. FURUHASHI, S. OKUMU, Y. UCHIKAWA**, « *Trajectory control of robotics manipulators using neural networks* », IEEE Trans. on Industrial Electronics, vol. 38, no. 3, June 1991.
- [24] **R.P. PAUL**, *Robot manipulators: Mathematics, Programming and control*, MIT Press, Cambridge, MA, 1981.
- [25] **M. SAAD, L.A. DESSAINT, P. BIGRAS, K. AL-HADDAD**, « *Adaptive versus neural adaptive control: Application to robotics* », International Journal of Adaptive Control and Signal Processing, vol. 8, pp 223-236, 1994.
- [26] **H. SERAJI**, « *Decentralized adaptive control of manipulators: Theory, simulation and experimentation* », IEEE Trans. on Robotics and Automation, vol. 5, no. 2, April 1989.
- [27] **J.J. SLOTINE, W. LI**, « *Adaptive manipulator control: A case study* », IEEE Trans. on Automatic Control, vol. 11, pp 995-1003, Nov. 1988.
- [28] **P.J. WERBOS**, « *Backpropagation through time: What it does and how to do it ?* », Proc. of IEEE, vol. 78, pp 1550-1560, 1990.
-

"Celui qui défile joyeusement au pas cadencé a déjà gagné mon mépris. C'est par erreur qu'on lui a donné un grand cerveau puisque la moelle épinière lui suffirait amplement.

On devrait éliminer sans délai cette honte de la civilisation. L'héroïsme sur commande, la brutalité stupide, cette lamentable attitude de patriotisme, quelle haine j'ai pour tout cela. Combien méprisable et vile est la guerre. Je préférerais être déchiré en lambeaux plutôt que de participer à quelque chose d'aussi méprisable. Je suis convaincu que tuer sous prétexte de guerre n'est rien d'autre qu'un assassinat pur et simple."

ALBERT EINSTEIN

ملخص :

في هذا العمل، عرضنا طريقة جديدة في التحكم الآلي التآلفي بواسطة الشبكات العصبية الاصطناعية من جهة. من جهة أخرى، لقد درسنا تقنيتين في التحكم اللامركزي. الأولى هي تعميم هذه الطريقة للهيكلة اللامركزية، أما الثانية فهي تقنية جديدة في التحكم التآلفي اللامركزي بالشبكات العصبية الاصطناعية. مجموع هذه الطرق قد طبقت على يد آلية ذات ثلاث درجات من الحرية لكي نقيم قدرات التحكم. كلمات مفتاحية : التحكم التآلفي، اللامركزية، الشبكات العصبية الاصطناعية، التقدير، الآلي.

RESUME :

Dans ce travail, nous avons présenté une nouvelle méthode de commande adaptative par réseaux de neurones (RNA) appliquées en robotique d'une part. D'autre part, nous avons étudiés deux techniques de commande décentralisée. La première est une généralisation de cette méthode à la structure décentralisée, la seconde est une nouvelle approche adaptative décentralisée par RNA. L'ensemble de ces méthodes ont été appliquées sur un bras de robot manipulateur à trois degrés de liberté afin d'évaluer leurs performances.

Mot Clés : *Commande Adaptative, Décentralisée, Réseaux de neurones, Estimation, Robotique.*

ABSTRACT :

In this work, a new decentralized adaptive robot controllers via neural network is developed. The method which synthesis the controller was presented with two techniques, the first, is the generalization of the centralized Lewis approach to a decentralized structure. The second is a new decentralized adaptive neural approach. Those techniques are applied to a robot manipulator with three degrees of freedom known as PUMA 560, in order to evaluate their performances.

Key Words : *Adaptive Control, Decentralized, Neurons Networks, Estimation, Robotics.*