

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
 MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
 Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

Département : Génie électrique & informatique
 Filière : Automatique

المدرسة الوطنية المتعددة التقنيات
 BIBLIOTHEQUE — المكتبة
 Ecole Nationale Polytechnique

En vue d'obtention du diplôme
 d'ingénieur d'état en Automatique

THEME

**COMMANDE COMBINEE FLOUE, MODE
 GLISSANT ET NEURONALE :
 APPLICATIONS EN ROBOTIQUE**

Proposé & dirigé par :

Mr F. BOUDJEMA
 Mr D. BOUKHETALA

Etudié par :

Riad Ahmed BORSALI
 Mohamed BRACI

Promotion
Juin 2000

E.N.P. 10, Avenue Hassen Badi, 16200 El-Harrach, Alger.

SOMMAIRE

INTRODUCTION GENERALE

Chapitre 1 : Modélisation des bras manipulateurs

1.1 Introduction.....	03
1.2 Modélisation Cinématique.....	03
1.2.1 Cinématique directe.....	03
1.2.2 Convention de Denavit et Hartenberg, 1955.....	04
1.2.3 Représentation du Robot PUMA 560.....	06
1.2.4 Présentation du robot à structure cylindrique de classe 4.....	09
1.3 Modélisation dynamique.....	11
1.3.1 Introduction.....	11
1.3.2 Approche d'Euler Lagrange.....	11
1.3.3 Modèle dynamique du robot puma 560.....	17
1.3.4 Modèle du robot à structure cylindrique.....	21
1.4 Génération de trajectoire.....	23
1.5 Conclusion.....	26

Chapitre 2 : Commande par logique floue

2.1 Introduction.....	27
2.2 Variables linguistiques.....	27
2.3 Ensemble flou.....	28
2.4 Opérateurs de la logique floue.....	28
2.5 Fonction d'appartenance.....	29
2.6 Configuration interne d'un régulateur par logique floue.....	30
2.7 Procédé lors de la conception d'un réglage par logique floue.....	31
2.8 La fuzzification.....	32
2.9 Moteur d'inférences floues.....	33
2.9.1 Méthode d'inférence max-min.....	33
2.9.2 Méthode d'inférence max-prod.....	34
2.9.3 Méthode d'inférence somme-prod.....	35
2.10 Défuzzification.....	35
2.11 Application de la commande floue.....	36
2.11.1 Commande décentralisée.....	36
2.11.2 Structure de la commande floue.....	37
2.11.3 Régulateur flou de type MAMDANI.....	37
2.11.4 Loi de commande.....	38
2.11.5 Les gains de normalisation.....	38
2.11.6 La trajectoire test.....	38
2.11.7 Synthèse des régulateurs.....	38
2.11.8 Application de la commande au robot PUMA 560.....	42
2.11.9 Etude de l'influence des gains de normalisation.....	55

2.13	Application de la commande sur le robot à structure cylindrique.....	59
2.13	Etude comparative.....	60
2.14	Conclusion.....	60

Chapitre 3 : Commande par mode de glissement floue

3.1	Introduction.....	74
3.2	Système de réglage à structure variable.....	74
3.3	Les modes de la trajectoire dans le plan de phase.....	76
3.4	Conception de la commande par mode de glissement.....	76
3.5	Choix de la surface de glissement.....	77
3.6	Calcul de la commande.....	78
3.7	Expression analytique de la commande.....	79
3.8	Elimination du phénomène du "Chattering".....	80
3.9	Application de la commande par mode de glissement flou.....	81
3.9.1	Application de la commande au robot PUMA 560.....	83
3.9.2	Etude de l'influence des coefficients des surfaces.....	96
3.9.3	Application de la commande au robot à structure cylindrique.....	99
3.10	Etude comparative.....	100
3.11	Conclusion.....	100

Chapitre 4 : Commande par réseaux de neurones

4.1	Introduction.....	114
4.2	Structure d'un neurone biologique.....	114
4.3	Le neurone artificiel.....	114
4.4	Architectures des réseaux de neurones adaptatifs.....	116
4.5	Les différents types de réseaux de neurones adaptatifs.....	117
4.5.1	Perceptron.....	117
4.5.2	Back Propagation neural network.....	118
4.5.3	Réseau de neurones à fonction de base Radiale (Radial Basis Network).....	120
4.6	L'algorithme d'apprentissage pour les réseaux Back Propagation.....	121
4.7	Application de la commande au Robot Puma 560.....	124
4.8	Simulations et interprétations.....	126
4.9	Conclusion.....	127

CONCLUSION GENERALE ET PERSPECTIVES

ANNEXES

Annexe 1 - Calcul du modèle dynamique du robot PUMA 560 Avec l'introduction de l'effet de l'effecteur.....	135
Annexe 2 - Calcul du modèle dynamique du robot à structure Cylindrique de classe 4.....	140
Annexe 3 - Gains de normalisation des différents régulateurs.....	142
Annexe 4 - Poids et biais des réseaux de neurones.....	144

BIBLIOGRAPHIE

DEDICACE

A mes très chers parents,

A mon frère Sofiane,

A mes sœurs,

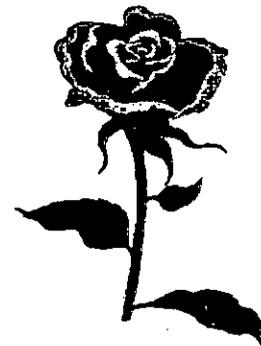
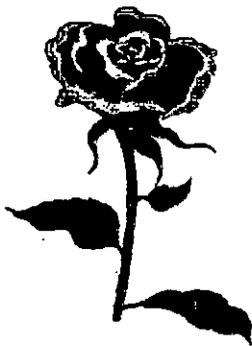
A mes grands parents,

A toute ma famille,

A tous mes amis,

Je dédie ce travail.

Mohamed



تالیفات و رسائل قلمیہ کی فہرست
BIBLIOTHEQUE — فہرست کتب
Ecole Nationale Polytechnique



فہرست کتب و رسائل قلمیہ کی فہرست

AVANT-PROPOS

الجمهورية التونسية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Lorsqu'une thèse s'achève, un désir s'empare de nous afin de remercier chacune des personnes dont la contribution a permis à ce travail de voir le jour.

*Nous tenons d'abord à exprimer nos sincères remerciements à nos deux promoteurs **Mr F. BOUDJEMA** et **Mr D. BOUKHETALA** qui ont accepté de nous encadrer et guider nos travaux jusqu'à la réalisation de ce mémoire. Qu'ils trouvent ici l'expression de notre profonde gratitude.*

Nous tenons également à remercier Messieurs les membres du jury qui nous ont fait l'honneur de juger ce travail.

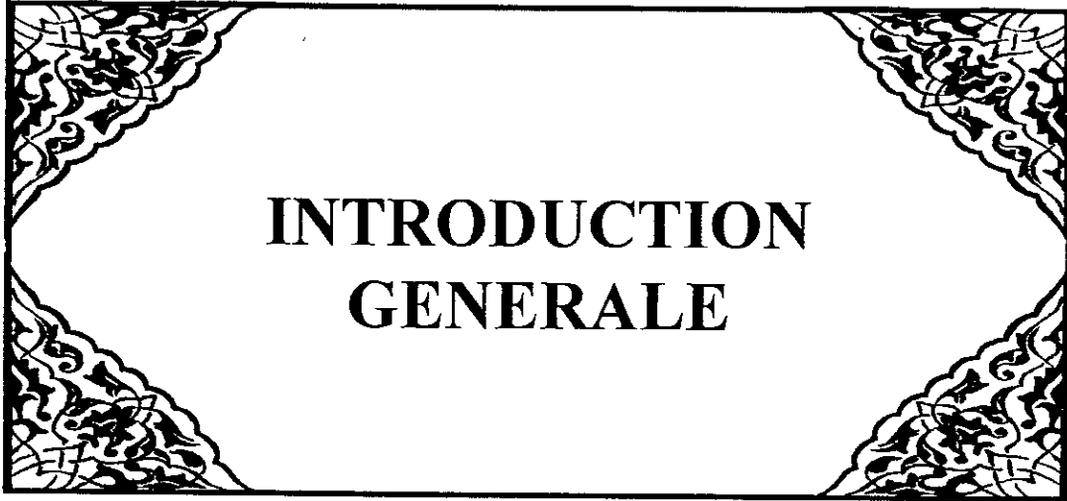
Notre sincère reconnaissance à nos parents qui nous ont aidés, soutenus et encouragés durant toutes nos études.

Que tout les enseignants qui ont contribué à notre formation trouvent ici notre sincère gratitude.

Nous ne manquerons pas de saluer et remercier nos amis présents quand nous avons besoin d'eux et ce, malgré leurs occupations : Islem, nounou, chikou, hatem, Lyes, Fateh, Abdelwaheb, Nabil, Adlan, Kamel, Omar, Yacine.

1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique



**INTRODUCTION
GENERALE**

بیتاؤیہ کتب خانہ
BIBLIOTHEQUE
Ecole Nationale Polytechnique

INTRODUCTION GENERALE

Les robots industriels d'aujourd'hui sont en fait des appareils actionnés mécaniquement sous le contrôle d'un ordinateur. Le mécanisme de préhension, appelé manipulateur, simule les gestes d'un bras humain ; il possède donc des articulations que l'on appelle, par analogie fonctionnelle, épaule, coude ou poignet. Le poignet est doué de mouvements de rotation traxiaux et les articulations sont actionnées par des dispositifs électriques.

Malheureusement, les robots manipulateurs présentent certains inconvénients, comme les non-linéarités des structures dynamiques, ainsi que les variations internes qui sont fortement couplées. Ce qui complique la synthèse des régulateurs utilisant les méthodes classiques, ceci a donné un nouvel axe de recherche aux scientifiques, qui essayent d'établir des méthodes simples, moins encombrante et surtout à structure non linéaire.

Dans ce contexte plusieurs approches de commande robuste ont été élaborées, comme par exemple : la logique floue, les modes glissant et les réseaux de neurones.

Le réglage par logique floue donne une approche plutôt pragmatique, il permet d'inclure les expériences acquises par les opérateurs. Cette nouvelle méthode de réglage a été introduite à large échelle d'abord au Japon et récemment répandue en Europe.

Le réglage par mode de glissement est un mode de fonctionnement particulier des systèmes à structure variable, cette commande a été étudiée d'abord en Union soviétique, puis vulgarisée un peu partout.

La combinaison de la logique floue avec mode de glissement est apparue après et a donné de bons résultats, mais elle reste toujours inexploitée dans le domaine de la robotique. Pour résoudre le problème de difficulté de conception de cette commande, une nouvelle stratégie de commande sera introduite, elle sera basée principalement sur le concept des réseaux de neurones.

Dans le présent travail nous proposons trois types de commandes appliquées à deux robots manipulateurs. D'abord la commande floue classique, puis la commande par mode de glissement flou, basé sur la condition d'existence des modes glissants, enfin en utilisant une approche de réseaux de neurones nous allons effectuer un apprentissage sur les mode glissant flou.

Ce mémoire est réparti en quatre chapitres :

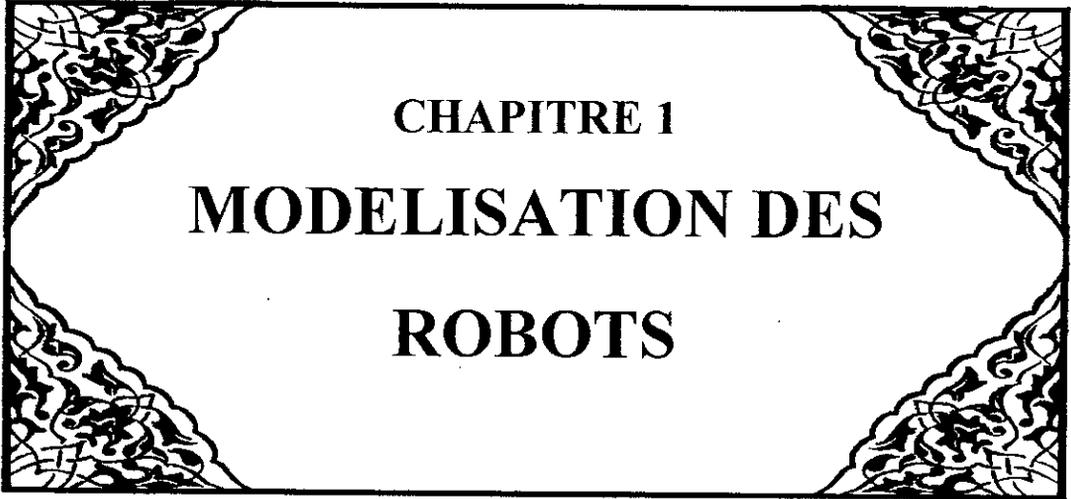
Dans le premier chapitre nous présenterons deux modèles de robots manipulateurs : le robot PUMA 560 caractérisé par trois rotations et le robot à structure cylindrique qui réalise deux translations et une rotation. Leur modélisation est basée sur le formalisme d'Euler-Lagrange augmenté par la technique présentée dans l'article [2]

Le second chapitre est consacré à des généralité sur la logique floue et la méthode de conception du régulateur flou qui sera appliqué sur les deux robots manipulateurs présentés précédemment, en utilisant trois type de classes d'appartenances.

Dans le troisième chapitre nous présenterons des généralités sur les modes glissants, qui seront appliqués en combinaison avec la commande floue sur les robots manipulateurs en utilisant les même classes que celle pour la commande floue.

Pour le dernier chapitre nous avons introduit les réseaux de neurones sur la commande par mode glissant flou, qui sera appliqué au robot PUMA 560, et cela par un concept simple, en utilisant l'algorithme d'apprentissage de Levenberg-Marquart.

Enfin une conclusion générale et quelques perspectives de notre travail clôtureront le présent mémoire.



CHAPITRE 1
MODELISATION DES
ROBOTS

1.1 Introduction :

Le terme "robot" vient du tchèque ; il signifie : travail forcé. Le mot, sous son acceptation moderne, fut introduit par l'auteur tchèque Karel Capek dans son ouvrage R.U.R. qui date de 1921. Bien que les robots de Capek ressemblent fort à des êtres humains, ils sont totalement dépourvus de sentiments et travaillent beaucoup plus dur que ceux-ci.

Pour le sens commun un robot est un dispositif mécanique articulé capable d'imiter certaines fonctions humaines telles que la manipulation d'objets ou la locomotion, dans le but de se substituer à l'homme pour la réalisation de certaines tâches matérielles. Cette réalisation est plus ou moins autonome selon les facultés de perception de l'environnement dont est doté le robot.

Les robots sont en train de révolutionner l'industrie moderne. Ils ne se comportent pas encore comme des êtres humains et sont très loin d'en avoir l'aspect extérieur ; ils n'en accomplissent pas moins bon nombre de leurs tâches. Ils s'avèrent particulièrement précieux dans de nombreuses applications industrielles, en particulier la manutention, la peinture, la soudure, le contrôle et l'assemblage mécanique. Les perspectives ouvertes par les robots sur l'évolution prochaine des usines sont en tout cas impressionnantes. Les recherches actuelles portent en effet sur des robots évolués capable de voir, d'entendre, de toucher et de prendre des décisions.

1.2 Modélisation Cinématique :

1.2.1 Cinématique directe :

On peut considérer qu'un manipulateur est une chaîne de liaisons connectées par des charnières ou glissières d'articulation, en utilisant des matrices de passages d'ordre quatre définies par Denavit et Hartenberg la liaison localise les informations à son propre repère. En ce qui concerne les coordonnées homogènes d'un vecteur quelconque pour ramener toutes les informations dynamiques et cinématiques au repère d'inertie de la base, il est préférable de les localiser à leurs articulations correspondantes avec des informations ramenées à la base.

En temps normal quand on a uniquement des mouvements de rotation on a une matrice de transformation du troisième ordre, mais en ajoutant une translation la matrice de transformation devient une matrice du quatrième ordre et ainsi le référentiel se déplacera vers un autre référentiel en translation. Ainsi le vecteur de position p sera augmenté par une quatrième composante et on aura un vecteur de position p' exprimé par ses coordonnées homogènes :

$$p = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix} \quad (1.1)$$

$$p' = \begin{bmatrix} vp_x & vp_y & vp_z & v \end{bmatrix} \quad (1.2)$$

pour $v=1$ le vecteur p' sera exprimé par les coordonnées physiques.

La matrice augmentée de transformation devient :

$$T = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

et

$$p'_{XYZ} = T p'_{UVW} \quad (1.4)$$

avec OXYZ référentiel fixe au niveau du socle.
OUVW référentiel en mouvement par rapport à OXYZ.

La matrice T lie ces deux référentiels.

1.2.2 Convention de Denavit et Hartenberg, 1955 :

Un manipulateur est une chaîne de liaisons connectées par des articulations. Il y a un mouvement de rotation ou de translation des liaisons par rapport à un repère appelé repère d'inertie, le passage entre le référentiel est garanti par des transformations homogènes basant sur la convention de Denavit et Hartenberg. Ce passage nous permet d'arriver à exprimer rapidement des informations de la base vers l'élément terminal ou l'inverse, et aussi elle facilite le calcul des matrices homogènes.

Cette méthode concerne le passage entre articulations adjacentes d'un système robotique. Elle est valable pour les chaînes cinématiques ouvertes ou l'articulation possède uniquement un degré de liberté. Pour cet aspect on utilise des charnières ou des glissières.

Cette technique comporte les étapes suivantes :

- ① Numérotation de chaque segment qui constitue le bras manipulateurs de la base vers l'élément terminal.
- ② Définir les axes principaux de chaque segment :
 - Z_i est l'axe de rotation du segment (rotation).
 - Z_{i-1} est l'axe de translation du segment dans O_{i-1} (translation).
 - L'axe X_i est perpendiculaire à Z_i et Z_{i-1} .
- ③ Le nombre de paramètres géométriques à fixer pour chaque articulation est de quatre : $d_i, \theta_i, a_i, \alpha_i$.
 - d_i : coordonnée de l'origine O_i sur l'axe Z_{i-1} (pour une glissière, d_i est une variable).
 - a_i : distance entre les axes Z_i et Z_{i-1} mesurée sur l'axe X_i négatif à partir de son origine jusqu'à l'intersection avec l'axe Z_{i-1} .
 - α_i : l'angle entre Z_i et Z_{i-1} obtenu en vissant Z_i vers Z_{i-1} autour de X_i .
- ④ On forme une matrice homogène de déplacement qui lie la rotation et la translation. Le vecteur droit définit la translation et la partie supérieure gauche définit la matrice rotation.

$$T_{i-1}^i = \begin{bmatrix} C_{i-1}^i & S_{i-1}^i & 0 & d_{i-1}^i \\ -S_{i-1}^i & C_{i-1}^i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.5)$$

avec

$$C_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \quad (1.6)$$

et

$$d_{i-1}^i = [a_i \cos\theta_i \quad a_i \sin\theta_i \quad d_i]^T \quad (1.7)$$

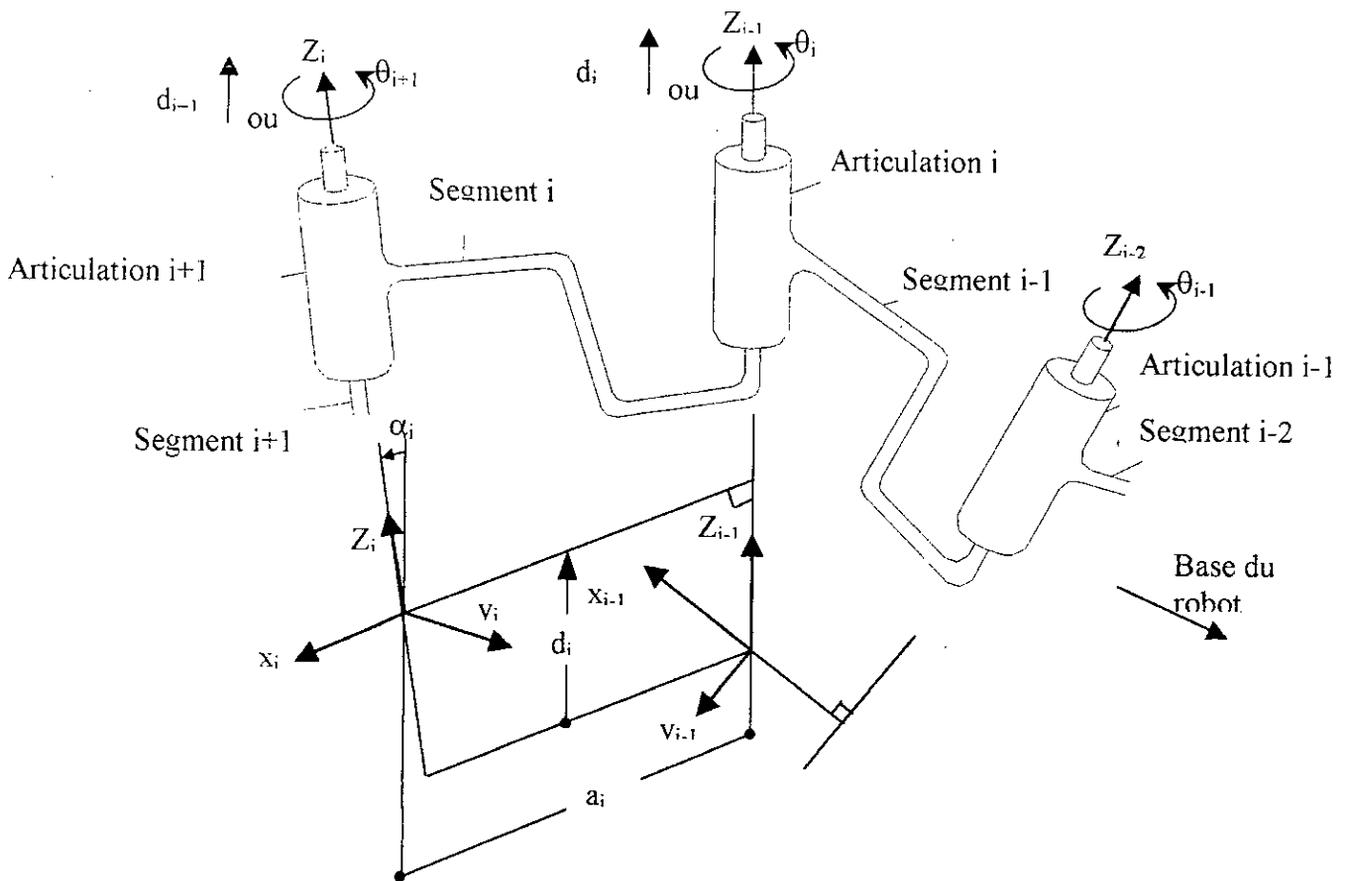


Figure 1.1 : Systèmes de coordonnées associés aux segments d'un robot et leurs paramètres selon la convention de Denavit et Hartenberg

On aura la matrice de transformation suivante :

$$T_{i-1}^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.8)$$

1.2.3 Représentation du Robot PUMA 560 :

Ce robot possède trois degrés de liberté caractérisant trois rotations

N° de la liaison	variable	θ_i	a_i	d_i	α_i
1	q_1	θ_1	0	0	-90°
2	q_2	θ_2	l_2	d_2	0
3	q_3	θ_3	l_3	0	0

Informations géométriques du Robot PUMA 560

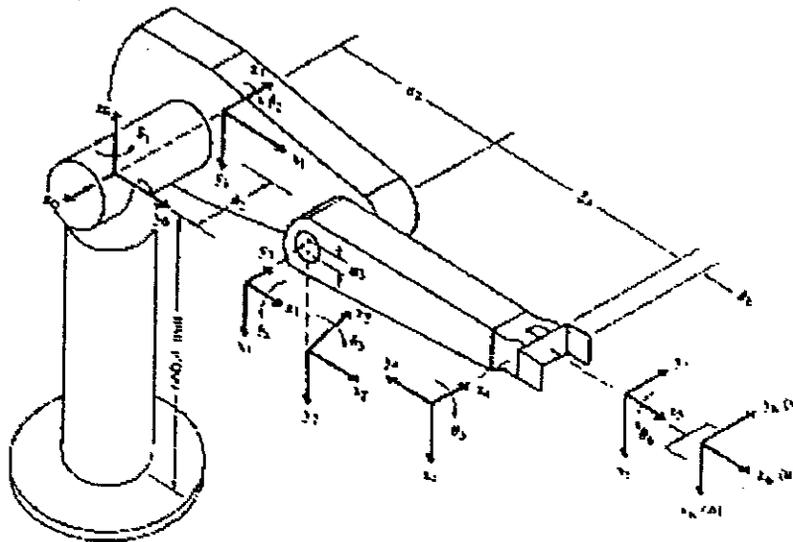


Figure 1.2 : Représentation du robot PUMA 560

Définition des repères :

L'origine du repère d'inertie $R_0(X_0, Y_0, Z_0)$ est prise à l'intersection de l'axe de la première articulation et l'axe de la liaison 1 avec :

- Z_0 est parallèle à la première articulation ;
- X_0 est parallèle à l'axe de la liaison 1.

L'origine du repère $R_1(X_1, Y_1, Z_1)$ coïncide avec celle du repère $R_0(X_0, Y_0, Z_0)$ et Z_1 est parallèle à l'axe de la seconde articulation.

Repère $R_2(X_2, Y_2, Z_2)$ se repose sur l'articulation 2, et X_2 suivant l'axe de la liaison 3.

Repère $R_3(X_3, Y_3, Z_3)$ est sur le centre de masse de l'effecteur.

En utilisant le formalisme de Denavit et Hartenberg on effectue les transformations appropriées suivantes :

$$\begin{aligned}
 T_0^1 &= \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & T_1^2 &= \begin{bmatrix} c_2 & -s_2 & 0 & l_2 c_2 \\ s_2 & c_2 & 0 & l_2 s_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} & T_2^3 &= \begin{bmatrix} c_3 & -s_3 & 0 & l_3 c_3 \\ s_3 & c_3 & 0 & l_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 T_0^2 &= \begin{bmatrix} c_1 c_2 & -c_1 s_2 & -s_1 & l_2 c_1 c_2 - d_2 s_1 \\ s_1 c_2 & -s_1 s_2 & c_1 & l_2 c_2 s_1 + d_2 c_1 \\ -s_2 & -c_2 & 0 & -l_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} & & & (1.9)
 \end{aligned}$$

$$T_0^3 = \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & -s_1 & c_1(l_2 c_2 - l_3 c_{23}) - d_2 s_1 \\ s_1 c_{23} & -s_1 s_{23} & c_1 & s_1(l_2 c_2 + l_3 c_{23}) + d_2 c_1 \\ -s_{23} & -c_{23} & 0 & -(l_2 s_2 + l_3 s_{23}) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

avec $c_i = \cos \theta_i$; $s_i = \sin \theta_i$; $c_{ij} = \cos(\theta_i + \theta_j)$; $s_{ij} = \sin(\theta_i + \theta_j)$.

Modélisation géométrique

La charge est considérée comme un point centrique dans l'espace défini par une position en coordonnées cartésiennes et donne à l'élément terminal la position et l'orientation pour manipuler les objets. Nous pouvons citer quelques solutions :

- Méthode des matrices duelles, par Denavit 1956.
- Quaternions duaux, par Yang et Arenden 1964.
- Transformation inverse, par Paul et Al 1981.
- Approche géométrique, par Lee et Ziegler 1984.

On peut définir les indicateurs de configuration qui constituent le PUMA 560 :

arm (bras), *elbow* (coude), *wrist* (poignet).

Avec:

$$\begin{aligned}
 \text{arm} &= \begin{cases} +1 & \text{right arm} \\ -1 & \text{left arm} \end{cases} & \text{elbow} &= \begin{cases} +1 & \text{above arm} \\ -1 & \text{below arm} \end{cases} & \text{wrist} &= \begin{cases} +1 & \text{Wrist down} \\ -1 & \text{Wrist up} \end{cases}
 \end{aligned}$$

(1.10)

Solution pour les trois premières articulations

Le vecteur position est comme suit :

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} c_1(l_2c_2 + l_3c_{23}) - d_2s_1 \\ s_1(l_2c_2 + l_3c_{23}) + d_2c_1 \\ -(l_2s_2 + l_3s_{23}) \end{bmatrix} \quad (1.11)$$

Solution de la première articulation :

$$\sin\theta_1 = \frac{-arm \ p_y \sqrt{p_x^2 + p_y^2 - d_2^2} - p_x d_2}{p_x^2 + p_y^2} \quad (1.12)$$

$$\cos\theta_1 = \frac{-arm \ p_x \sqrt{p_x^2 + p_y^2 - d_2^2} + p_y d_2}{p_x^2 + p_y^2} \quad (1.13)$$

$$\theta_1 = \tan^{-1} \left[\frac{arm \ p_y \sqrt{p_x^2 + p_y^2 - d_2^2} - p_x d_2}{arm \ p_x \sqrt{p_x^2 + p_y^2 - d_2^2} + p_y d_2} \right] \quad (1.14)$$

Solution de la 2^{ème} articulation :

$$\sin\theta_2 = \sin\alpha \cos\beta + arm \ elbow \ \cos\alpha \ \sin\beta \quad (1.15)$$

$$\cos\theta_2 = \cos\alpha \cos\beta + arm \ elbow \ \sin\alpha \ \sin\beta \quad (1.16)$$

$$\theta_2 = \tan^{-1} \left[\frac{\sin\alpha \cos\beta + arm \ elbow \ \cos\alpha \ \sin\beta}{\sin\alpha \cos\beta + arm \ elbow \ \cos\alpha \ \sin\beta} \right] \quad (1.17)$$

avec

$$\cos\alpha = \frac{-arm \sqrt{p_x^2 + p_y^2 - d_2^2}}{\sqrt{p_x^2 + p_y^2 + p_z^2 - d_2^2}} \quad (1.18)$$

$$\sin\alpha = \frac{p_z}{\sqrt{p_x^2 + p_y^2 + p_z^2 - d_2^2}} \quad (1.19)$$

$$\cos\beta = \frac{p_x^2 + p_y^2 + p_z^2 - d_2^2 - l_2^2 - l_3^2}{2l_2 \sqrt{p_x^2 + p_y^2 + p_z^2 - d_2^2}} \quad (1.20)$$

$$\sin\beta = \sqrt{1 - \cos^2\beta} \quad (1.21)$$

Solution de la 3^{ème} articulation :

$$\cos\varphi = \frac{l_2^2 + l_3^2 - p_x^2 + p_y^2 + p_z^2 - d_2^2}{l_3^2} \quad (1.22)$$

$$\sin\varphi = arm \ elbow \ \sqrt{1 - \cos^2\varphi} \quad (1.23)$$

avec

$$\varphi = \theta_3 + \frac{\pi}{2} \quad (1.24)$$

1.2.4 Présentation du robot à structure cylindrique de classe 4 :

Parmi les robots les plus rencontrés en robotique, le robot à structure cylindrique de classe 4 décrit sur la figure 1.3. Ce manipulateur a été établi par D.P.Stoten [39], il représente près de 45% des systèmes industriels utilisés actuellement. Ce robot réalise une rotation de l'ensemble autour de l'axe vertical suivit d'un mouvement d'évolution vertical et une translation horizontale, dont les mouvements sont identifiés par les variables q_1 , q_2 et q_3 .

L'application de la convention de Denavit et Hartenberg est présentée par la figure 1.3.

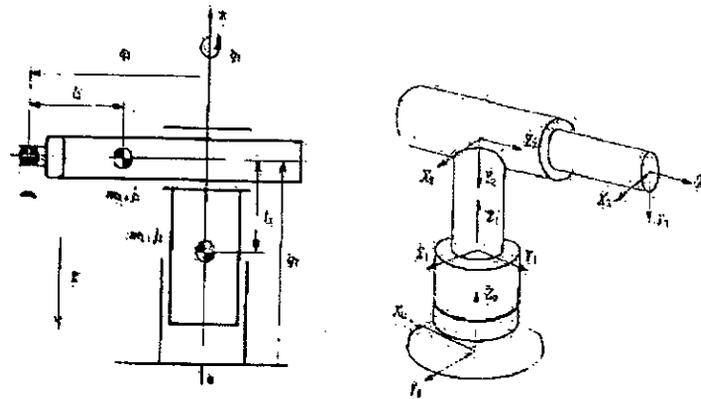


Figure 1.3 : Robot de classe 4 à structure cylindrique

Les coordonnées généralisées de ce robot sont notées par θ_1 , d_1 et d_2 , caractérisant respectivement une rotation et deux translations. Le vecteur des coordonnées généralisées sera :

$$q = [\theta_1, d_2, d_3]^T \quad (1.25)$$

Le tableau suivant donne les informations géométriques de ce robot :

N° de la liaison	Variable	θ_i	a_i	d_i	α_i
1	q_1	θ_1	0	0	0
2	q_2	0	0	d_2	-90°
3	q_3	0	0	d_3	0

Informations géométriques du robot à structure cylindrique

Les matrices de transformation DH appropriées à ce modèle sont

$$\begin{aligned}
 T_0^1 &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & T_1^2 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} & T_2^3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 T_0^2 &= \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} & T_0^3 &= \begin{bmatrix} c_1 & 0 & -s_1 & -d_3 s_1 \\ s_1 & 0 & c_1 & d_3 c_1 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{1.26}$$

avec $c_i = \cos \theta_i$; $s_i = \sin \theta_i$.

et T_i^{i+1} est la matrice de passage du repère $R_i(X_i, Y_i, Z_i)$ au repère $R_{i+1}(X_{i+1}, Y_{i+1}, Z_{i+1})$.

Modélisation géométrique

Cette approche résout le problème de la cinématique inverse, elle positionne l'organe terminal par rapport au repère d'inertie $R_0(X_0, Y_0, Z_0)$ sous forme d'une fonction $r(R_0)$ à partir des variables articulaires q_i .

$$r(R_0) = F(q) \tag{1.27}$$

avec

$$\begin{cases} r = [p_x \ p_y \ p_z \ 1]^T \\ F(p) = T_0^3 r_3^3 \end{cases} \tag{1.28}$$

et r_3^3 coordonnées de l'élément terminal exprimé dans $R_3(X_3, Y_3, Z_3)$ ou :

$$r_3^3 = [0 \ 0 \ 0 \ 1]^T \tag{1.29}$$

et de l'équation (1.28) on trouve :

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} c_1 & 0 & -s_1 & -d_3 s_1 \\ s_1 & 0 & c_1 & d_3 c_1 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{donc} \quad \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} -d_3 s_1 \\ d_3 c_1 \\ d_2 \\ 1 \end{bmatrix} \tag{1.30}$$

de l'équation (1.30) on tire les coordonnées généralisées du robot de classe 4 qui sont la rotation et les deux translations donc :

$$\begin{cases} \theta_1 = \arctan\left(-\frac{p_x}{p_y}\right) \\ d_2 = p_z \\ d_3 = -\frac{p_x}{\sin\theta_1} = \frac{p_y}{\cos\theta_1} \end{cases} \quad (1.31)$$

pour $p_y \neq 0$ et $\theta_1 \neq \frac{\pi}{2} + k\pi$

1.3 Modélisation dynamique :

1.3.1 Introduction :

Jusqu'ici, présentant les modèle géométrique et cinématique, nous avons délibérément fait des hypothèses qui nous facilitaient le travail mathématique d'élaboration d'algorithmes de commande mais dont les fondements ne sont justifiés que si la mécanique du manipulateur est d'une fabrication extrêmement soignée et si on accepte de le faire mouvoir à des vitesses faibles, voire très faibles. C'est dans ce sens que nous allons présenter des modèles dynamiques des bras manipulateurs définis par un ensemble d'équations mathématiques. L'ensemble des équations dynamiques peuvent être déterminés par des lois mécaniques classiques Newtoniennes et Lagrangiennes. Les approches d'Euler-Lagrange et Newton-Euler permettent d'aboutir aux équations du mouvement des robots.

1.3.2 Approche d'Euler-Lagrange :

L'une des approches qui sert à modéliser et présenter la dynamique des robots à travers les équations du mouvement est l'approche d'Euler-Lagrange. Elle s'adapte lors des calculs manuels ainsi que pour des calculs assistés par ordinateur. Le formalisme d'Euler-Lagrange et la transformation homogène de Denavit et Hartenberg amène à un algorithme compact pour présenter les équations dynamiques du mouvement.

L'équation d'Euler-Lagrange est :

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_i} \right] - \frac{\partial L}{\partial q_i} + \frac{\partial E_D}{\partial q_i} = T_i \quad i = 1, \dots, n \quad (1.32)$$

ou L est le Lagrangien qui s'exprime par

$$L = E_c - E_p \quad (1.33)$$

E_c : énergie cinétique totale de toutes les liaisons ;

E_p : énergie potentielle totale de toutes les liaisons ;

E_D : énergie de dissipation en cas de présence de frottement visqueux ;

T_i : force généralisée à la $i^{\text{ème}}$ articulation ;

n : nombre de degré de liberté ;

q_i : coordonnée généralisée d'ordre i ;

\dot{q}_i : dérivée de la coordonnée généralisée.

Energie cinétique.

On peut trouver la valeur de L'énergie cinétique par l'expression de la vitesse :

$$V_o^i = \frac{dr_o^i}{dt} \quad (1.34)$$

$$r_o^i = T_o^i r_i^i \quad (1.35)$$

ou r_i^i est coordonnée homogène du point (i) exprimée dans le repère R_i . Comme la liaison n'est pas flexible, on a $\frac{dr_i^i}{dt} = 0$ donc :

$$V_o^i = \sum_{j=i}^i \frac{\partial T_o^i}{\partial q_j} \frac{dq_j}{dt} r_i^i \quad (1.36)$$

avec T_{i-1}^i est la matrice de transformation homogène. Sous une forme plus compacte, on peut écrire :

$$V_o^i = \sum_{j=i}^i \left[U_{ij} \quad \dot{q}_j \right] r_i^i \quad (1.37)$$

et

$$U_{ij} = \begin{cases} T_o^{j-1} Q_j T_{j-1}^i & j \leq i \\ 0 & j \geq i \end{cases} \quad (1.38)$$

pour une liaison rotative on a :

$$Q_j = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.39)$$

et pour une liaison translationnelle :

$$Q_j = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1.40)$$

l'énergie cinétique de l'élément i dans la liaison i est

$$dE_{Ci} = \frac{1}{2} \text{trace}(V_i V_i^T) dm \quad (1.41)$$

En développant l'expression précédente, on obtient :

$$dE_{Ci} = \frac{1}{2} \text{trace} \left(\sum_{j=1}^i \sum_{k=1}^i U_{ij} \left(r_i^j r_i^{jT} dm \right) U_{ik}^T \dot{q}_j \dot{q}_k \right) \quad (1.42)$$

l'énergie cinétique de la liaison i est :

$$E_{Ci} = \frac{1}{2} \text{trace} \left(\sum_{j=1}^i \sum_{k=1}^i U_{ij} J_i U_{ik}^T \dot{q}_j \dot{q}_k \right) \quad (1.43)$$

avec :

$$J_i = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \quad (1.44)$$

l'énergie cinétique des actionneurs est définie par :

$$E_{Ca} = \frac{1}{2} \sum_{i=1}^n I_i \dot{q}_i^2 \quad (1.45)$$

où I_i caractérise un moment d'inertie dans le cas d'une rotation et une masse dans le cas d'une translation de l'actionneur i . L'énergie cinétique totale sera :

$$E_c = \sum_{i=1}^n E_{Ci} + E_{Ca} \quad (1.46)$$

L'énergie potentielle est décrite par :

$$E_p = \sum_{i=1}^n -m_i g^T r_0^i = \sum_{i=1}^n m_i g^T T_0^i r_i^i \quad (1.47)$$

$$\text{ou } g \text{ est la gravité } g^T = [0 \quad 0 \quad -|g| \quad 1] \quad (1.48)$$

Enfin, l'énergie de dissipation est donnée par :

$$E_D = \frac{1}{2} \sum_{i=1}^n f_{vi} \dot{q}_i^2 \quad (1.49)$$

ou f_{vi} le coefficient de frottement visqueux ($i=1, n$). L'expression finale du Lagrangien sera :

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i \text{trace} (U_{ij} J_i U_{ik}^T) \dot{q}_j \dot{q}_k + \sum_i m_i g^T T_0^i r_i^i \quad (1.50)$$

En appliquant la formule d'Euler-Lagrange à la fonction Lagrangienne donnée ci-dessus, on trouve la force ou couple généralisé

$$T_i = \sum_{j=1}^n \sum_{k=l}^j \text{trace}(U_{ij} J_j U_{ij}^T) \ddot{q}_k + \sum_{j=1}^n \sum_{k=l}^j \sum_{l=1}^j \text{trace}(U_{ijk} J_j U_{ij}^T) \dot{q}_k \dot{q}_l - \sum_{j=l}^n m_j g^T U_{jl} r_j^l + f_{vi} \dot{q}_i \quad (1.51)$$

avec

$$U_{ijk} = \begin{cases} T_0^{k-l} Q_k T_{k-l}^{j-l} Q_j T_{j-l}^i & k \leq j \leq i \\ T_0^{j-l} Q_j T_{j-l}^{k-l} Q_k T_{k-l}^i & j \leq k \leq i \\ 0 & j < i < k \end{cases} \quad (1.52)$$

Méthode directe 1 :

Elle est basée essentiellement sur l'équation (1.51) qui est extraite par la méthode d'Euler-Lagrange, sa forme est :

$$T_i = \sum_{j=1}^n M_{ij}(q) \ddot{q}_j + \sum_{j=1}^n \sum_{k=l}^j N_{ijk}(q) \dot{q}_j \dot{q}_k + G_i(q) + H_i(\dot{q}_i), \quad i=1, \dots, n \quad (1.53)$$

avec

$$M_{ij}(q) = \sum_{k=\max(i,j)}^n \text{trace}(U_{ij} J_k U_{ij}^T) \quad \begin{matrix} i = 1, \dots, n \\ j = 1, \dots, n \end{matrix} \quad (1.54)$$

$$G_i(q) = - \sum_{j=i}^n m_j g^T U_{jl} r_j^l \quad (1.55)$$

$$N_{ijk}(q) = \sum_{l=\max(i,j,k)}^n \text{trace}(U_{ijk} J_l U_{ij}^T) \quad \begin{matrix} i = 1, \dots, n \\ j = 1, \dots, n \\ k = 1, \dots, n \end{matrix} \quad (1.56)$$

$$H_i(\dot{q}_i) = f_{vi} \dot{q}_i \quad (1.57)$$

L'équation dynamique est :

$$T(t) = M(q) \ddot{q} + N(q, \dot{q}) + G(q) + H(\dot{q}) \quad (1.58)$$

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

$M(q) \in \mathcal{R}^{n \times n}$: matrice symétrique définie positive des accélérations inertielles dans l'élément M_{ij} de cette matrice est l'inertie de la $i^{\text{ème}}$ articulation sur la $j^{\text{ème}}$ articulation et vice versa ;

$N(q, \dot{q}) \in \mathcal{R}^n$: vecteur de force et/ou couples dus aux accélérations de coriolis et centrifuge ;

$G(q) \in \mathcal{R}^n$: vecteur de force et/ou couples dus aux forces de gravitation ;

$H(\dot{q}) \in \mathfrak{R}^n$: représente les frottements visqueux ;

$T(t) \in \mathfrak{R}^n$: vecteur de forces et/ou couples moteurs.

Méthode Directe 2 :

Cette méthode améliore le temps de calcul lors de la détermination du vecteur dû aux accélérations de coriolis et centrifuges par rapport à la méthode précédente. Et ici le modèle dynamique sera calculé en trois étapes :

❶ Calcul des éléments de la matrice d'inertie par la méthode directe 1.

❷ Calcul de deux matrices centrifuges et coriolis qui sont multipliées par leurs vecteurs, la somme des vecteurs obtenue nous donnent le vecteur voulu N . L'extrait de ces dernières matrices se fait par la dérivation de la matrice d'inertie (*principe de la conservation d'énergie*).

❸ Enfin, le calcul des vecteurs des gravités et visqueux, par la méthode directe en se basant sur les équations (1.55) et (1.57) respectivement. La forme appropriée du modèle final sera :

$$T(t) = M(q)\ddot{q} + D(q)\left[\dot{q} \quad \dot{q}\right] + C(q)\left[\dot{q}^2\right] + G(q) + H(\dot{q}) \quad (1.59)$$

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

$M(q) \in \mathfrak{R}^{n \times n}$: matrice symétrique définie positive des accélérations inertielles dans l'élément M_{ij} de cette matrice est l'inertie de la $i^{\text{ème}}$ articulation sur la $j^{\text{ème}}$ articulation et vice versa ;

$D(q) \in \mathfrak{R}^{n \times \frac{n(n-1)}{2}}$: matrice des couples de coriolis ;

$C(q) \in \mathfrak{R}^{n \times n}$: matrice des couples de centrifuges ;

$G(q) \in \mathfrak{R}^n$: vecteur des accélérations gravitationnelles ;

$H(\dot{q}) \in \mathfrak{R}^n$: représente les frottements visqueux ;

$T(t) \in \mathfrak{R}^n$: vecteur de forces ou couples généralisés ;

$\left[\dot{q} \quad \dot{q}\right] \in \mathfrak{R}^{n \times \frac{n(n-1)}{2}}$: vecteur de produit de produits des vitesses généralisées ;

$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} \in \mathfrak{R}^n$: vecteur de carré des vitesses généralisées.

Ou:

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} \dot{q}_1 \dot{q}_2 & \dots & \dot{q}_1 \dot{q}_n & \dot{q}_2 \dot{q}_3 & \dots & \dot{q}_2 \dot{q}_n & \dots & \dot{q}_{n-2} \dot{q}_n & \dot{q}_{n-1} \dot{q}_n \end{bmatrix} \quad (1.60)$$

et

$$\begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & \dots & \dot{q}_n \end{bmatrix}^T \quad (1.61)$$

Le calcul de la matrice D de coriolis se base essentiellement sur la formule suivante : [2]

$$D_{ij} = 2 \beta^{i,k} \quad (1.62)$$

d'où le symbole de Christoffel définit par :

$$\beta^{i,jk} = \frac{1}{2} \left[\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_i} \right] \quad (1.63)$$

Puisque la matrice d'inertie est définie positive, on aura les propriétés suivantes :

$$\frac{\partial M_{ij}}{\partial q_k} = \frac{\partial M_{ji}}{\partial q_k} \quad \forall i, j, k \quad (1.64)$$

et

$$\frac{\partial M_{ij}}{\partial q_k} = 0 \quad \text{pour } i \geq k, \quad j \geq k \quad (1.65)$$

Les éléments de la matrice centrifuge sont définis par l'équation suivante [2] :

$$C_{ij} = \beta^{i,j} \quad (1.66)$$

Modèle dynamique général [9] :

Le modèle dynamique général d'un robot manipulateur rigide à n degrés de liberté, avec comme entrée le vecteur de forces ou couples généralisés T et comme sortie le vecteur position q , peut être représenté par un système d'équations différentielles non-linéaires de second ordre comme suit [9] :

$$M(q)\ddot{q} + B(q, \dot{q})\dot{q} + K(q, \dot{q})q + G(q) + H(\dot{q}) = T_p(q) + T(q) \quad (1.67)$$

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

$M(q) \in \mathfrak{R}^{n \times n}$: la matrice d'inertie ;

$B(q, \dot{q})\dot{q} + K(q, \dot{q})q \in \mathfrak{R}^n$: englobe les couples dus aux forces de coriolis et centrifuges ;

$G(q) \in \mathfrak{R}^n$: le vecteur de forces ou couples dus aux forces de gravitation ;

$T_p(t) \in \mathfrak{R}^n$: le vecteur de force ou couples de perturbation externe ;

$T(t) \in \mathfrak{R}^n$: le vecteur de force ou couples moteurs.

M, B, K, G et H sont généralement des fonctions très compliquées et non-linéaires.

Ce modèle possède les propriétés suivantes :

- La matrice $M(q)$ est symétrique définie positive (SDP), par conséquent tous les éléments diagonaux de cette matrice sont positifs $\{M_{ii}(q) > 0 \quad i=1, \dots, n\}$.
- Les matrices M, B, K et les vecteurs G, H sont uniformément bornées.
- L'entrée de commande est indépendante pour chaque articulation du manipulateur.
- Le vecteur des frottements visqueux et secs $H(q)$ est caractérisé par les n éléments $\{H_i(q_i) \quad i=1, \dots, n\}$, tel que :

$$H_i(q_i) = f_{vi} \dot{q}_i + f_{si} \operatorname{sgn}(\dot{q}_i) \quad (1.68)$$

avec f_{vi} et f_{si} sont respectivement les coefficients des frottements visqueux et secs de la $i^{\text{ème}}$ articulation.

Les propriétés 1-4 découlent de la nature physique du robot manipulateur. La propriété 3 est due à la négligence des flexibilités des articulations et des structures.

1.3.3 Modèle dynamique du robot puma 560 [2] :

Le calcul du modèle dynamique est présenté en annexe 1. Ce manipulateur réalise trois mouvements rationnels, le premier suivant le plan horizontal, le second et le troisième suivant deux axes verticaux. Le modèle dynamique est le suivant :

$$T = M(q)\ddot{q} + N(q, \dot{q}) + G(q) + T_{m0} \quad (1.69)$$

avec :

$$M(q) = \begin{bmatrix} I_1 + I_2 c_{23}^2 + I_3 c_2^2 + I_4 c_2 c_{23} & I_5 s_{23} + I_6 s_2 & I_5 s_{23} \\ I_5 s_{23} + I_6 s_2 & I_7 + I_4 c_3 & I_8 + 0.5 I_4 c_3 \\ I_5 s_{23} & I_8 + 0.5 I_4 c_3 & I_9 \end{bmatrix} \quad (1.70)$$

$$N(q, \dot{q}) = \begin{bmatrix} -(2(I_3 s_2 c_2 + I_2 s_{23} c_{23}) + I_4 (c_2 s_{23} + s_2 c_{23})) \dot{q}_1 \dot{q}_2 - (2I_2 s_{23} c_{23} + I_4 c_2 s_{23}) \dot{q}_1 \dot{q}_3 \\ + (I_6 c_2 + I_5 c_{23}) \dot{q}_2^2 + (2I_5 c_{23}) \dot{q}_2 \dot{q}_3 + (I_5 c_{23}) \dot{q}_3^2 \\ (I_3 c_2 s_2 + I_2 c_{23} s_{23} + 0.5 I_4 (s_2 c_{23} + c_2 s_{23})) \dot{q}_1^2 - (I_4 s_3) \dot{q}_2 \dot{q}_3 - (0.5 I_4 s_3) \dot{q}_3^2 \\ (I_2 s_{23} c_{23} + 0.5 I_4 c_2 s_{23}) \dot{q}_1^2 + (0.5 I_4 s_3) \dot{q}_2^2 \end{bmatrix} \quad (1.71)$$

$$G(q) = \begin{bmatrix} 0 \\ -(m_3 l_2 + 0.5 m_2 l_2) g c_2 - 0.5 m_3 l_3 g c_{23} \\ -0.5 m_3 l_3 g c_{23} \end{bmatrix} \quad (1.72)$$

$$T = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \quad (1.73)$$

avec

$$\begin{aligned} c_i &= \cos(q_i) & s_i &= \sin(q_i) \\ c_{ij} &= \cos(q_i + q_j) & s_{ij} &= \sin(q_i + q_j) \end{aligned}$$

et

$$\begin{aligned} I_1 &= I_{yy1} + I_{xx2} + m_2 d_2 (d_2 + e) + m_3 d_2^2 + I_{xx3} + I_{xxt} + m_t d_2^2 + I_{Mt} \\ I_2 &= I_{yy3} - I_{xx3} + I_{yyt} - I_{xxt} + m_t l_3^2 \\ I_3 &= I_{yy2} - I_{xx2} + m_5 l_2^2 + m_t l_2^2 \\ I_4 &= m_3 l_2 l_3 + 2 m_t l_2 l_3 \\ I_5 &= 0.5 m_3 l_3 d_2 + m_t l_3 d_2 \\ I_6 &= 0.5 m_2 l_2 (d_2 + e) + m_3 l_2 d_2 + m_t l_2 d_2 \\ I_7 &= I_{zz2} + I_{zz3} + m_3 l_2^2 + I_{ztt} + m_t (l_2^2 + l_3^2) + I_{Mt} \\ I_8 &= I_{zz3} + I_{ztt} + m_t l_3^2 \\ I_9 &= I_8 + I_{Mt} \end{aligned} \quad (1.74)$$

Les effets des effecteurs sont représentés par la matrice ΔM qui contient les éléments suivant :

$$\begin{aligned}
\Delta M_{11} &= I_{xx1} + m_1 d_2^2 + I_{M1} + (I_{yy1} - I_{xx1} + m_1 l_3^2) c_{23}^2 + m_1 l_2^2 c_2^2 + 2m_1 l_2 l_3 c_2 c_{23} \\
\Delta M_{12} &= m_1 l_3 d_2 s_{23} + m_1 d_2 l_2 s_2 \\
\Delta M_{13} &= m_1 l_3 d_2 s_{23} \\
\Delta M_{22} &= I_{zz1} + m_1 (l_2^2 + l_3^2) + I_{M2} + 2m_1 l_2 l_3 c_3 \\
\Delta M_{23} &= I_{zz1} + m_1 l_3^2 + 2m_1 l_2 l_3 c_3 \\
\Delta M_{33} &= I_{zz1} + m_1 l_3^2 + I_{M3}
\end{aligned} \tag{1.75}$$

I_{Mi} moment d'inertie des différents moteurs.

I_{xx1} , I_{yy1} , I_{zz1} moment d'inertie total par rapport aux principaux axes de l'effecteur.

Le vecteur des couples additifs T_{m0} représente l'effet de la charge, il est calculé par la matrice Jacobienne, cette dernière est la dérivée du vecteur position de l'effecteur.

$$J_j(q) = \frac{\partial P}{\partial q_j} \tag{1.76}$$

avec:

$$p = [c_1(l_2 c_2 + l_3 c_{23}) - d_2 s_2 \quad s_1(l_2 c_2 + l_3 c_{23}) + d_2 c_1 \quad -(l_2 c_2 + l_3 c_{23})]^T \tag{1.77}$$

et

$$J(q) = [J_1(q) \quad J_2(q) \quad J_3(q)]^T \tag{1.78}$$

Donc la matrice Jacobienne :

$$J(q) = \begin{bmatrix} -s_1(l_2 s_2 + l_3 c_{23}) - d_2 c_1 & -c_1(l_2 s_2 + l_3 s_{23}) & -c_1(l_3 s_{23}) \\ c_1(l_2 s_2 + l_3 c_{23}) - d_2 s_1 & -s_1(l_2 s_2 + l_3 s_{23}) & -s_1(l_3 s_{23}) \\ 0 & -(l_2 s_2 + l_3 c_{23}) & -(l_3 c_{23}) \end{bmatrix} \tag{1.79}$$

Le couple dû à l'effet de la charge est :

$$T_{m0} = m_0 J^T(q) [J(q)\ddot{q} + J(q, \dot{q})\dot{q} + g] \tag{1.80}$$

avec $g = [0 \quad 0 \quad 9.81]^T$

Les paramètres du bras de robot PUMA 560

a) masse des différentes liaisons :

$$\begin{aligned}
m_2 &= 17.40 \text{ kg} & m_3 &= 5.04 \text{ kg} & m_4 &= 0.82 \text{ kg} \\
m_5 &= 0.35 \text{ kg} & m_6 &= 0.09 \text{ kg} & m_t &= m_4 + m_5 + m_6 = 1.26 \text{ kg}
\end{aligned}$$

b) paramètres géométriques :

$$d_2 = 1490.09 \text{ mm} \quad l_2 = 431.8 \text{ mm} \quad l_3 = 433.07 \text{ mm}$$

c) paramètre d'inerties :

N° de la liaison	$I_{xxi}(kg\ m^2)$	$I_{yyi}(kg\ m^2)$	$I_{zzj}(kg\ m^2)$	$I_{mi}(kg\ m^2)$
1	-	$350\ 10^{-3}$	-	1.14
2	$130\ 10^{-3}$	$524\ 10^{-3}$	$539\ 10^{-3}$	4.71
3	$192\ 10^{-3}$	$15.4\ 10^{-3}$	$212\ 10^{-3}$	0.83
4	$1.30\ 10^{-3}$	$1.80\ 10^{-3}$	$1.80\ 10^{-3}$	-
5	$0.30\ 10^{-3}$	$0.30\ 10^{-3}$	$0.40\ 10^{-3}$	-
6	$0.04\ 10^{-3}$	$0.15\ 10^{-3}$	$0.15\ 10^{-3}$	-
4+5+6	$1.64\ 10^{-3}$	$2.25\ 10^{-3}$	$2.35\ 10^{-3}$	-

Paramètres d'inertie pour le robot PUMA 560

Les réponses du robot PUMA 560 en position et en vitesse pour une entrée échelon, sur un horizon de temps suffisant, sont consignées sur la figure ci dessous :

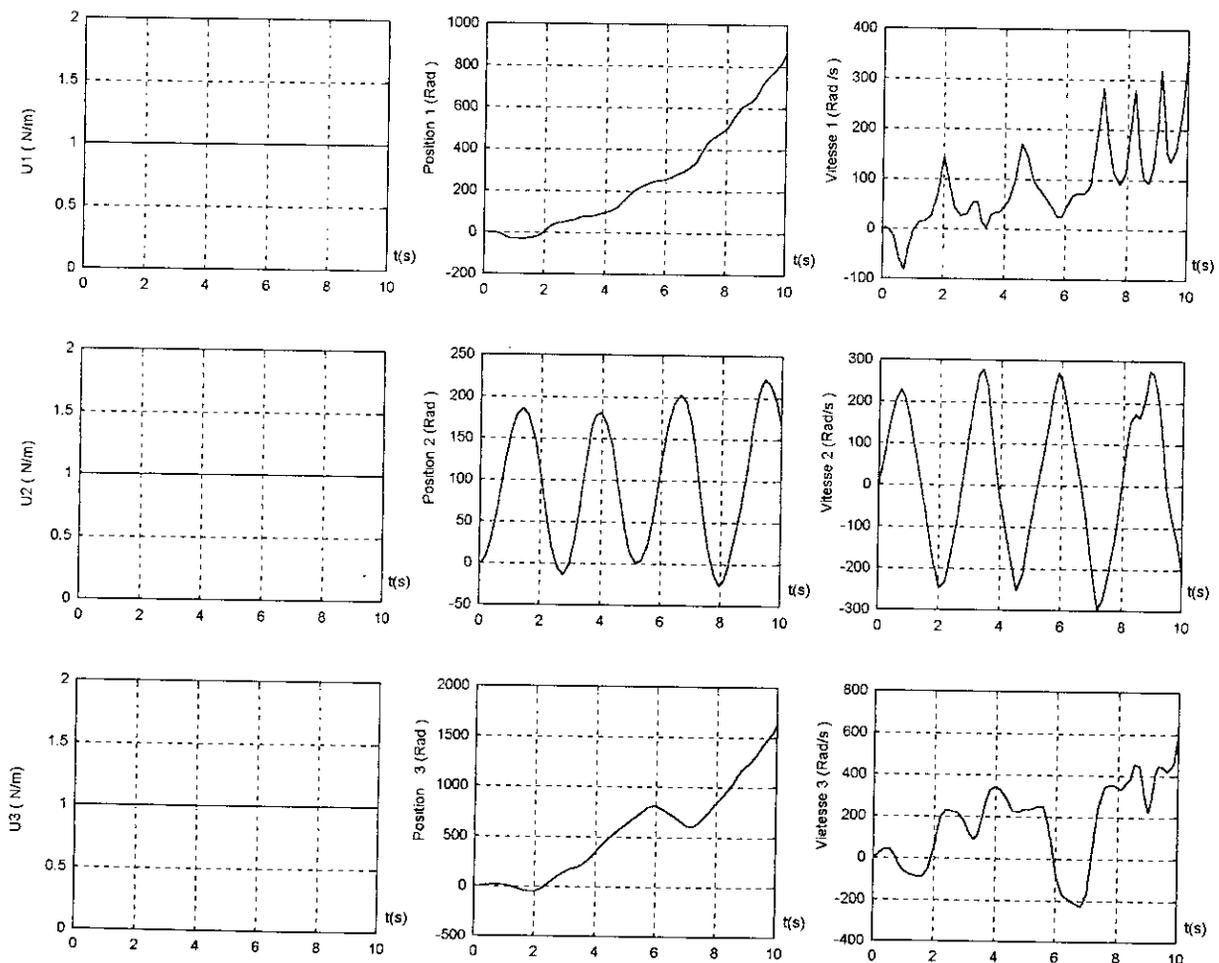


Figure 1.4 Réponse indicielle du robot PUMA 560 en position et en vitesse.

1.3.4 Modèle dynamique du robot à structure cylindrique :

D.P. Stoten [39] a proposé une démarche pour simplifier la modélisation du manipulateur de classe 4, elle consiste à correspondre les masses localisées m_2 et m_3 au segments de longueurs totales l_2 et l_3 et que le moment d'inertie bras l_3 par rapport à un axe parallèle à l'axe oz, mais passant par le centre de masse vaut I_3 . Le moment d'inertie du segment l_2 et de son actionneur est noté par I_2 . La charge à porter est supposée être une masse ponctuelle m_0 rattachée avec la deuxième articulation et localisée à la distance q_3 par rapport à l'axe oz. Afin d'établir le modèle dynamique, plusieurs hypothèses doivent être prises en considération :

- Les frottements sont de nature visqueuse et linéaire par rapport à la vitesse généralisée;
- Les différentes liaisons sont rigides ;
- Les actionneurs sont idéaux (la force généralisée est directement proportionnelle au signal de commande) ;
- Les capteurs ont un gain unitaire et de dynamique négligeable.

L'énergie potentielle totale du système est :

$$E_p = (m_2 + m_3 + m_0)g(q_2 - q_{20}) \quad (1.81)$$

avec q_{20} correspond à la position particulière q_2 du manipulateur pour laquelle l'énergie potentielle est nulle. L'énergie cinétique s'exprime par :

$$E_c = \frac{1}{2} \left[(I_2 + I_3 + m_3(q_3 - l_3)^2 + m_0 q_3^2) \dot{q}_1^2 + (m_2 + m_3 + m_0) \dot{q}_2^2 + (m_3 + m_0) \dot{q}_3^2 \right] \quad (1.82)$$

avec l'introduction des frottements visqueux f_{v1}, f_{v2}, f_{v3} aux différentes articulations l'application du formalisme d'Euler-Lagrange donne :

$$\begin{aligned} (I_2 + I_3 + m_3(q_3 - l_3)^2 + m_0 q_3^2) \ddot{q}_1 + 2(m_3(q_3 - l_3) + m_0 q_3) \dot{q}_1 \dot{q}_3 + f_{v1} \dot{q}_1 &= \Gamma_1 \\ (m_2 + m_3 + m_0) \ddot{q}_2 + (m_2 + m_3 + m_0)g + f_{v2} \dot{q}_2 &= F_2 \\ (m_3 + m_0) \ddot{q}_3 - (m_3(q_3 - l_3) + m_0 q_3) \dot{q}_1^2 + f_{v3} \dot{q}_3 &= F_3 \end{aligned} \quad (1.83)$$

Avec Γ_1, F_2, F_3 couples et forces généralisées. Les équations dynamiques du manipulateur de classe 4 font apparaître des effets de coriolis. Comme les actionneurs sont supposés être idéaux, les forces et les couples généralisés sont directement proportionnels aux signaux de commande ($\Gamma_1 = k_{c1}u_1$, $F_2 = k_{c2}u_2$, $F_3 = k_{c3}u_3$).

Les paramètres du manipulateur sont :

$$\begin{array}{llll} l_3 = 0.75m & m_2 = 20 \text{ kg} & m_3 = 10 \text{ kg} & I_2 = 2 \text{ kg m}^2 \\ I_3 = 0.2 \text{ kg m}^2 & f_{v1} = 7.825 \text{ N rad}^{-1} \text{ s} & f_{v2} = 30 \text{ N m}^{-1} \text{ s} & f_{v3} = 20 \text{ N m}^{-1} \text{ s} \\ k_{c1} = 10 \text{ N m V}^{-1} & k_{c2} = 100 \text{ N V}^{-1} & k_{c3} = 10 \text{ N V}^{-1} & g = 9.81 \text{ m s}^{-2} \end{array}$$

Les réponses du robot à structure cylindrique en position et en vitesse pour une entrée échelon, sur un horizon de temps suffisant, sont consignées sur la figure ci dessous :

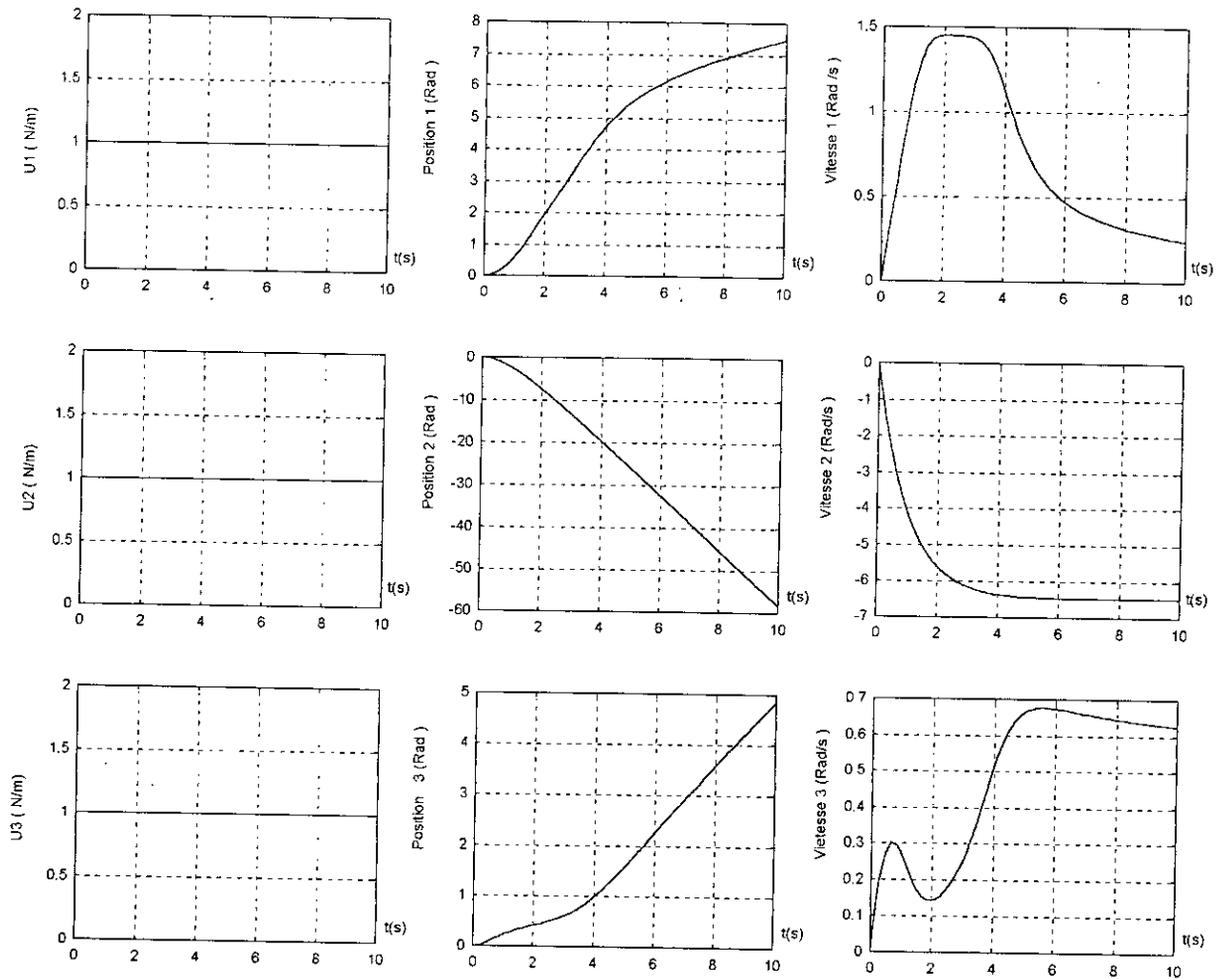


Figure 1.5 Réponse indicielle du robot à structure cylindrique en position et en vitesse.

1.4 Génération de trajectoire :

La dynamique des robots exige l'imposition d'une trajectoire de référence réalisable afin d'assurer le déplacement de l'élément terminal du robot d'un point initial vers le point final avec des commandes admissibles sur les articulations.

Pour avoir une valeur finale constante par exemple, On doit utiliser un polynôme du troisième degré, et on aura donc une continuité en position et en vitesse. Le passage d'un état d'équilibre à un autre état d'équilibre définit les conditions aux limites du polynôme d'interpolation :

$$P(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

$$\begin{cases} p(\theta) = \theta_0 & p(\theta) = \theta_0 \\ p(t_f) = \theta_f & p(t_f) = \theta_f \end{cases} \quad (1.84)$$

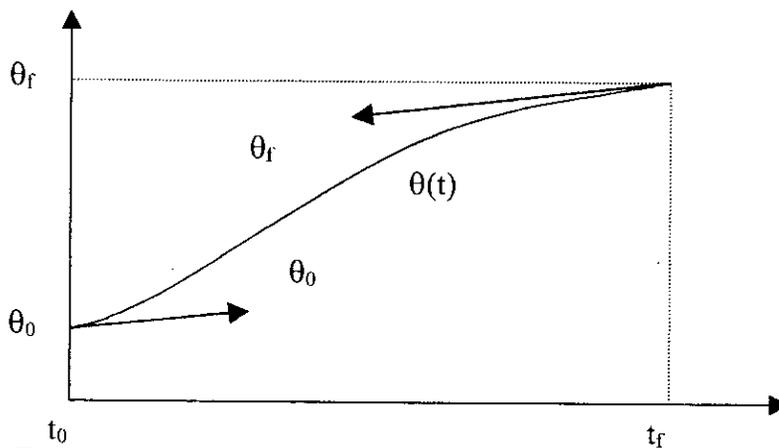


Figure 1.6 : Génération d'un polynôme d'interpolation

A partir des conditions aux limites (1.84), on calcule les différents coefficients du polynôme $p(t)$, on aura alors :

$$\begin{cases} a_0 = \theta_0 \\ a_1 = \theta \\ a_2 = \frac{3}{t_f^3}(\theta_f - \theta_0) - \frac{2}{t_f}\theta_0 - \frac{1}{t_f}\theta_f \\ a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) + \frac{1}{t_f^2}(\theta_f + \theta_0) \end{cases} \quad (1.85)$$

Trajectoire cycloïdale :

Parmi les trajectoires les plus répandues en robotique, on choisit pour la conduite normale du robot une trajectoire cycloïdale qui assure une continuité en position, en vitesse et en accélération. Cet avantage offre au robot la possibilité de poursuivre cette trajectoire avec

des commandes réalisables. Le test de la capacité de poursuite peut être réalisé par des trajectoires particulières qui excitent toute la dynamique du robot :

$$q_d(t) = \begin{cases} q_d(0) + \frac{D}{2\pi} \left[2\pi \frac{t}{t_f} - \sin\left(2\pi \frac{t}{t_f}\right) \right] & \text{pour } 0 \leq t \leq t_f \\ q_d(t_f) & \text{pour } t > t_f \end{cases} \quad (1.86)$$

avec $D = q_d(t_f) - q_d(0)$ est le déplacement, et t_f est l'instant final du mouvement, et q_d est la trajectoire de référence.

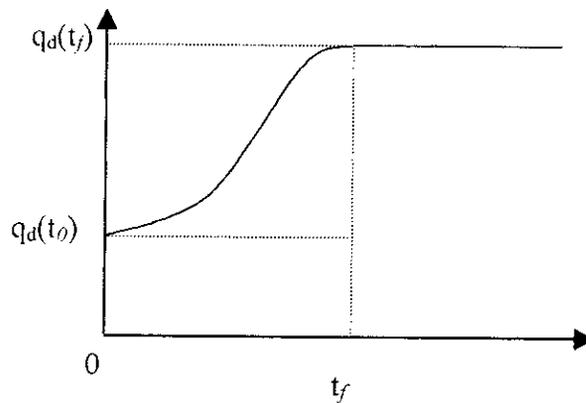


Figure 1.7 : trajectoire cycloïdale

Pour le PUMA 560, il existe une trajectoire cycloïdale test proposé par LEAHVY. Les différentes articulations (art1, art2, art3) se déplacent de la position $(-50^\circ, -135^\circ, 135^\circ)$ à la position $(45^\circ, -85^\circ, 30^\circ)$ en un temps de mouvement égal à 1.5 secondes. Cette trajectoire est choisie car elle excite toute la dynamique de ce bras de robot.

Pour une trajectoire donnée (trajectoire de l'effecteur du bras de robot), avec une consigne variable, on peut facilement déterminer les déplacements et vitesses dus à chaque articulation $(\theta_i, \dot{\theta}_i)$, ceci en utilisant les équations du modèle géométrique et cinématique inverse des robots.

Trajectoire de VIVIANI. [26]

La trajectoire de VIVIANI sert à exciter toutes les dynamiques des robots dont la structure est cylindrique, elle est définie par l'intersection d'un cylindre d'axe verticale de rayon $R/2$ et de centre $(0, R/2, 0)$ et une sphère de rayon R et de centre $(0,0,0)$:

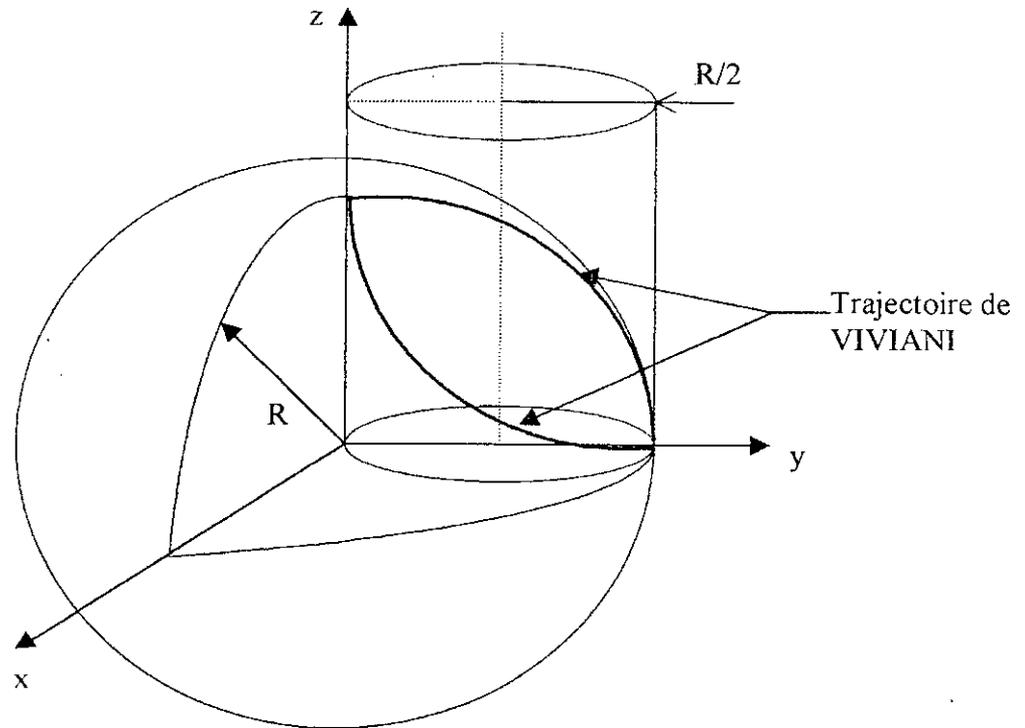


Figure 1.8 : Détermination de la trajectoire de VIVIANI

L'équation de la sphère est :

$$x^2 + y^2 + z^2 = R^2 \quad (1.87)$$

L'équation du cylindre est :

$$x^2 + \left(y - \frac{R}{2}\right)^2 = \left(\frac{R}{2}\right)^2 \quad (1.88)$$

On fait le changement de variable suivant :

$$\begin{cases} x = \rho \cos(\theta) \\ y = \rho \sin(\theta) \end{cases} \quad (1.89)$$

En appliquant le changement de variable aux équations (1.87) et (1.88), on obtient :

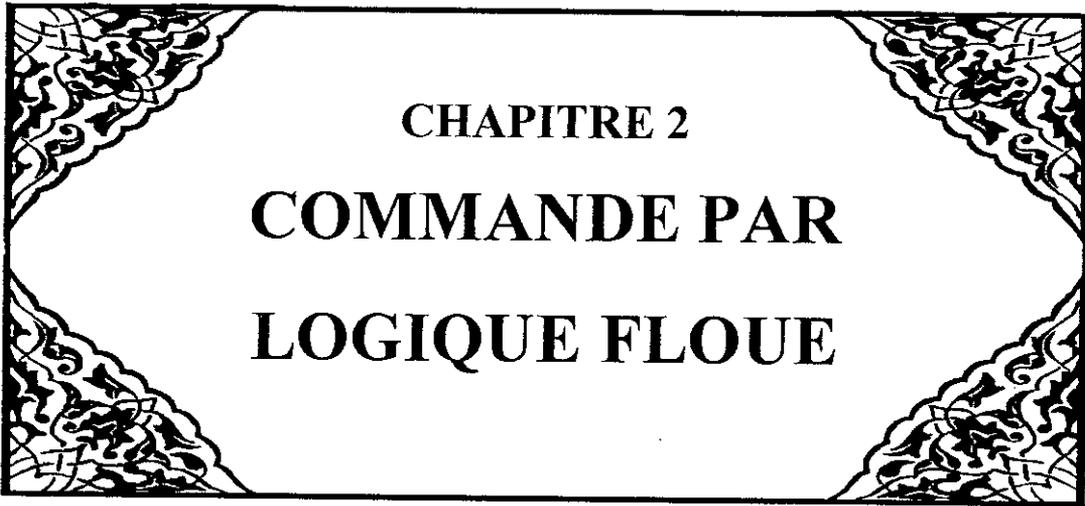
$$\begin{cases} \rho = R \sin(\theta) \\ z = |R \cos(\theta)| \end{cases} \quad (1.90)$$

La détermination de la courbe paramétrée par θ , ρ et z est similaire à θ_1 , d_1 et d_2 pour le robot cylindrique. En imposant θ on obtient directement les valeurs de z et ρ , on prend par exemple $\theta = k t$ une droite passant par l'origine.

1.5 Conclusion :

Dans ce chapitre, nous avons appliqué les formalismes d'Euler-Lagrange à deux types de robots manipulateur à trois degrés de liberté : le robot PUMA 560 et le robot à structure cylindrique, qui nous ont permis d'établir les systèmes d'équations différentielles reliant les coordonnées généralisées aux forces ou aux couples généralisés. Les modèles obtenus nous ont permis de mettre en évidence les fortes non-linearités qui régissent le fonctionnement des robots, ce qui met en évidence la difficulté de commander ce type de bras manipulateurs.

Les robots manipulateurs ne supportent pas des discontinuités des références, c'est la raison pour laquelle on impose des références douces et continues en position et en vitesse, permettant ainsi le contrôle des robots avec des commandes acceptables.



CHAPITRE 2
COMMANDE PAR
LOGIQUE FLOUE

2.1 Introduction :

De nos jours, la logique floue (en anglais fuzzy logic) est de grande actualité. L'opinion courante est qu'il s'agit d'une nouvelle méthode de traitement pour les problèmes de réglage et de prise de décision provenant du Japon. Cependant, cette argumentation n'est pas du tout correcte. En effet, les bases théoriques de la logique floue ont été établies en 1965 par le professeur Lotfi A.Zadeh de l'université de Californie de Berkeley [43][44]. A cette époque, la théorie de la logique floue n'a pas été prise au sérieux. En effet, les ordinateurs avec leur fonctionnement exact par tout ou rien (1 ou 0) ont commencé à se répandre sur une large échelle, contrairement à la logique floue qui permet de traiter des variables non exactes dont la valeur peut varier entre 0 et 1. Initialement, cette théorie a été appliquée dans des domaines non techniques, comme le commerce, la jurisprudence ou la médecine, dans le but de compléter les systèmes experts, et afin de leur donner l'aptitude de prise de décision.

A partir de 1985 environ, ce sont les Japonais qui commencent à utiliser la logique floue dans les produits industriels pour résoudre des problèmes de réglage et de commande. Tout récemment, aussi en Europe, une intense activité de recherche a débuté afin d'introduire le principe du réglage par logique floue.

Les avantages de la logique floue sont :

- Comportement non linéaire sans dynamique.
- Aptitude à régler convenablement des systèmes avec des comportements dynamiques et dont la modélisation est difficile.
- Non nécessité d'une modélisation pour faire la régulation (sauf pour la simulation)
- Connaissance approximative des paramètres du système à régler.

Les désavantages de la logique floue sont :

- Inexistence de règles pour le choix des méthodes de défuzzification et inférence.
- Impossible de démontrer la stabilité du système à régler sans modèle adéquat.
- Inexistence de lois mathématique pour la détermination des gains de normalisations.

Les éléments de base de la logique floue :

- Les variables linguistiques.
- La définition des variables linguistiques.
- Les déductions floues (inférences).
- Les opérateurs.

2.2 Variables linguistiques :

La description d'une certaine situation, d'un phénomène ou d'un procédé contient en générale des expressions floue comme : quelque, beaucoup, souvent ; chaud, froid, rapide, lent ; grand, petit, etc. Les expressions de ce genre forment les valeurs des *variables linguistiques* de la logique floue. Afin de permettre un traitement numérique, il est indispensable de les soumettre à une définition à l'aide de fonctions d'appartenance.

Une variable linguistique est représentée généralement par un triplet : $(x, T(x), U)$, dans lequel est le nom de variable linguistique (vitesse, position, erreur, ...), exemple, si l'erreur est considérée comme variable linguistique, définie sur le référentiel $U=[-10, +10]$, ses valeurs linguistiques peuvent être définies comme suit :

$T(\text{erreur}) = \{ \text{Négatif grand (NG)}, \text{Négatif petit (NP)}, \text{Environ zéro (EZ)}, \text{Positif petit (PP)}, \text{Positif grand (PG)} \}$.

Ces symboles linguistiques peuvent être considérés comme des ensembles flous, d'où la fonction d'appartenance représentée sur la figure 2.1.

2.3 Ensemble flou :

Mathématiquement, un ensemble flou A est défini sur un référentiel U , par une fonction d'appartenance, notée μ_A , qui appliquée à un élément u de U , retourne un degré ensemble de paires ordonnées : $\{(u, \mu_A(u)) / u \in U\}$.

Si U est décrit, A est représenté par :

$$A = \sum_{u_j \in U} \frac{\mu(u_j)}{u_j} \quad (2.1)$$

Si U est continu, A est représentée par :

$$A = \int \frac{\mu(u)}{u} \quad (2.2)$$

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

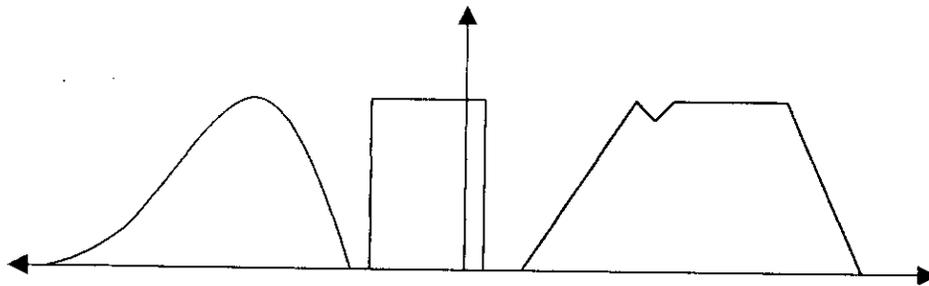


Figure 2.1 : Exemple d'ensemble flou

2.4 Opérateurs de la logique floue :

Les variables linguistiques sont liées entre elles au niveau des inférences par des opérateurs ET ou OU. IL s'agit d'opérateurs de la logique floue qui interviennent sur les fonctions d'appartenance représentant les variables linguistiques.

Le plus souvent, les opérateurs ET ou OU sont réalisés respectivement par les règles «min» et «max». Il y a alors une certaine affinité avec les règles de la logique classique. Cependant, il existe un grand nombre de règles pour la réalisation des opérateurs ET et OU qui tiennent compte du caractère particulier de la logique floue.

Soit A et B deux ensembles flous définis dans le référentiel U avec les fonctions d'appartenances μ_A et μ_B respectivement.

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

$$\mu_{A \cup B}(u) = \mu_A(u) + \mu_B(u) \quad (2.3)$$

Dans le domaine de la commande, le + représente souvent le MAX :

$$\mu_A(u) + \mu_B(u) = \text{Max}(\mu_A(u), \mu_B(u)) \quad (2.4)$$

Définition 2 : (intersection) La fonction d'appartenance $\mu_{A \cap B}$ de l'intersection $A \cap B$, est définie pour tout $u \in U$ par : $\mu_A(u) * \mu_B(u) = \mu_{A \cap B}(u)$ (2.5)

Définition 3 : (Complémentation) La fonction d'appartenance $\mu_{\bar{A}}$ du complément de l'ensemble A, notée \bar{A} , est définie pour tout $u \in U$ par :

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u) \quad (2.6)$$

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

$$\mu_{A_1 \times \dots \times A_n}(u) = \mu_{A_1}(U_1) \times \dots \times \mu_{A_n}(U_n) \quad (2.7)$$

Définition 5 : (Relation floue) Une relation floue représente le degré de présence, ou d'absence d'une association entre les éléments de deux ou plusieurs ensembles flous.

Une relation floue d'ordre n est un ensemble flou défini sur $U_1 \times \dots \times U_n$ par l'expression suivante :

$$R_{U_1 \times \dots \times U_n} = \{((U_1, \dots, U_n), \mu_R(u_1, \dots, u_n)) / (u_1, \dots, u_n) \in U_1 \times \dots \times U_n\} \quad (2.8)$$

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

$$\mu_{R \circ S}(u, w) = \left\{ (u, w), \text{SUP}_{v \in V} (\mu_R(u, v) \times \mu(v, w)) \right\} \quad (2.9)$$

2.5 Fonction d'appartenance :

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

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

$$\mu(x) = \text{Max} \left(\min \left(\frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (2.10)$$

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

$$\mu(x) = \text{Max} \left(\min \left(\frac{x-a}{b-a}, 1, \frac{c-x}{c-d}, 0 \right) \right) \quad (2.11)$$

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

$$\mu(x) = \exp \left(-\frac{(x-m)^2}{2\sigma^2} \right) \quad (2.12)$$

Fonction sigmoïdale : Elle est définie par deux paramètres $\{a, c\}$:

$$\mu(x) = \frac{1}{1 + \exp(-a(x-c))} \quad (2.13)$$

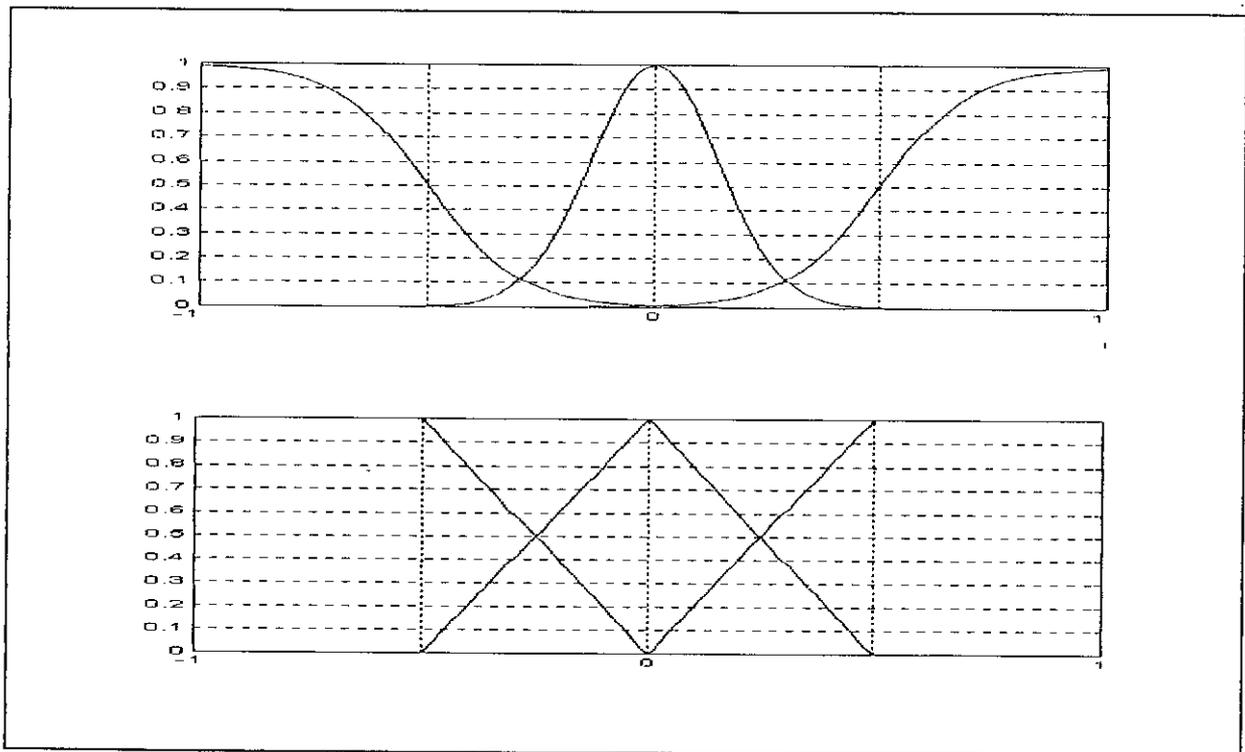


Figure 2.2 : Formes de fonctions d'appartenances

2.6 Configuration interne d'un régulateur par logique floue :

Par opposition à un régulateur standard ou à un régulateur par contre-réaction d'état, le *régulateur par logique floue* ne traite pas une relation mathématique bien définie

(algorithme de réglage), mais utilise des inférences avec plusieurs règles, se basant sur des variables linguistiques. Ces inférences sont alors traitées par des opérateurs de la logique floue.

La figure 2.3 montre la configuration interne d'un régulateur par logique floue. On peut distinguer trois parties :

- *La fuzzification* : contient en générale un traitement de données préliminaire, par exemple la formation de l'écart de réglage $e = w - y$ ou la détermination de la variation d'une grandeur (différence entre deux instants d'échantillonnage). Ces grandeurs sont alors traitées par des variables linguistiques, ce qui nécessite leur définition par des fonctions d'appartenances.
- *Le moteur d'inférence floue* : Les valeurs des variables linguistiques sont liées par plusieurs règles qui doivent tenir compte du comportement statique et dynamique du système à régler ainsi que des buts de réglages envisagés. En particulier, le circuit de réglage doit être stable et bien amorti. On obtient ainsi une information floue pour la variable de sortie x_R du régulateur.
- *La défuzzification* : Puisque l'organe de commande doit être attaqué avec une valeur bien précise pour le signal de commande u_{cm} , il faut transformer la valeur floue en une valeur déterminée. Cela se fait dans le bloc de défuzzification.

La fuzzification, les inférences et la défuzzification provoquent une caractéristique non linéaire du régulateur par logique floue. Il s'agit souvent d'un avantage par rapport à un régulateur linéaire.

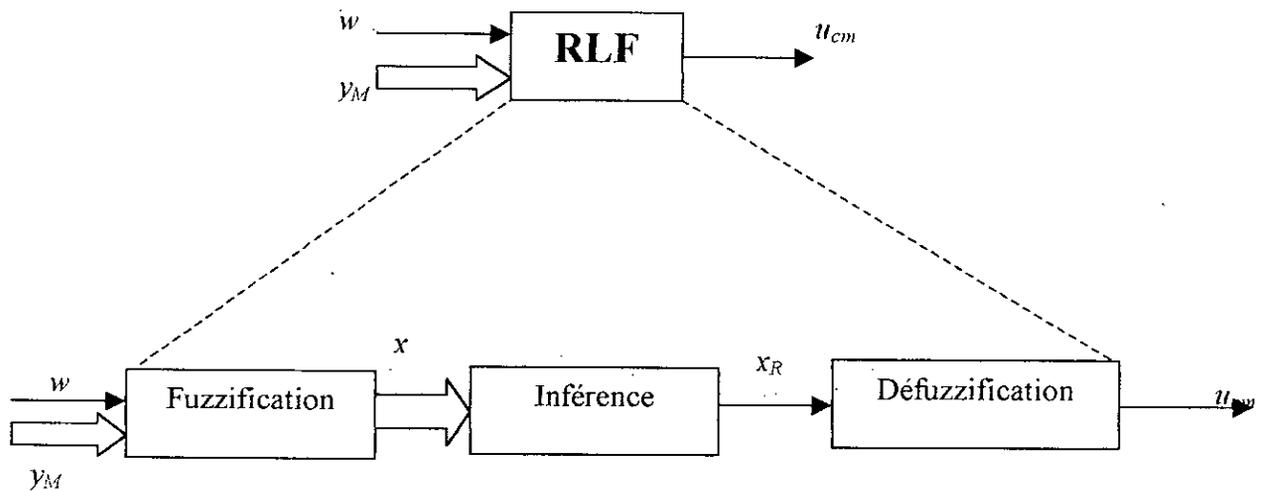


Figure 2.3 : Configuration interne d'un régulateur par logique floue.

2.7 Procédé lors de la conception d'un réglage par logique floue :

Le procédé à suivre lors de la conception d'un réglage par logique floue est assez différent de celui d'un réglage conventionnel. La figure 2.4 en montre les étapes principales.

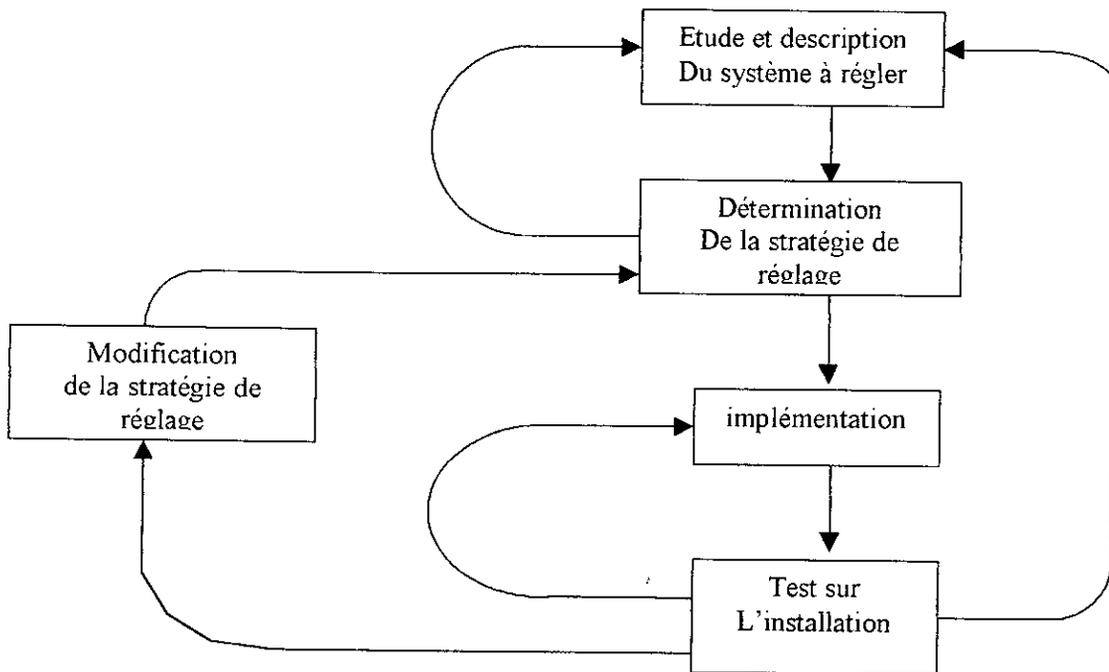


Figure 2.4 : Procédé lors de la conception d'un réglage par logique floue.

2.8 La fuzzification :

Le régulateur par logique floue se compose de trois parties, dont la première est la fuzzification. Il s'agit de la conversion analogique/digitale, ainsi que du traitement des grandeurs mesurées y_M et de leurs transformations en variables linguistiques avec la définition des fonctions d'appartenance. Lors de l'implémentation de la conversion analogique/digitale, il faut introduire des facteurs d'adaptation qui tiennent compte du format choisi pour les grandeurs converties. Cette conversion est liée d'une part à une quantification des signaux à cause de la longueur finie des grandeurs digitales et d'autre part à un échantillonnage à cause du fonctionnement discontinu de la conversion A/D et du traitement des inférences.

La fuzzification proprement dite consiste à définir les fonctions d'appartenances pour les différentes variables, en particulier pour les variables d'entrée, on réalise ainsi le passage des grandeurs physiques (grandeurs déterminées) en variables linguistiques (variables floues) qui peuvent alors être traitées par les inférences.

On prend comme exemple une variables $x = 0.57$ et une fonction d'appartenance avec cinq ensemble (Figure 2.5), on aura :

$$\mu_{ng}(x) = \mu_{np}(x) = \mu_{ez}(x) = 0$$

$$\mu_{pp}(x) = 0.29$$

$$\mu_{pg}(x) = 0.71$$

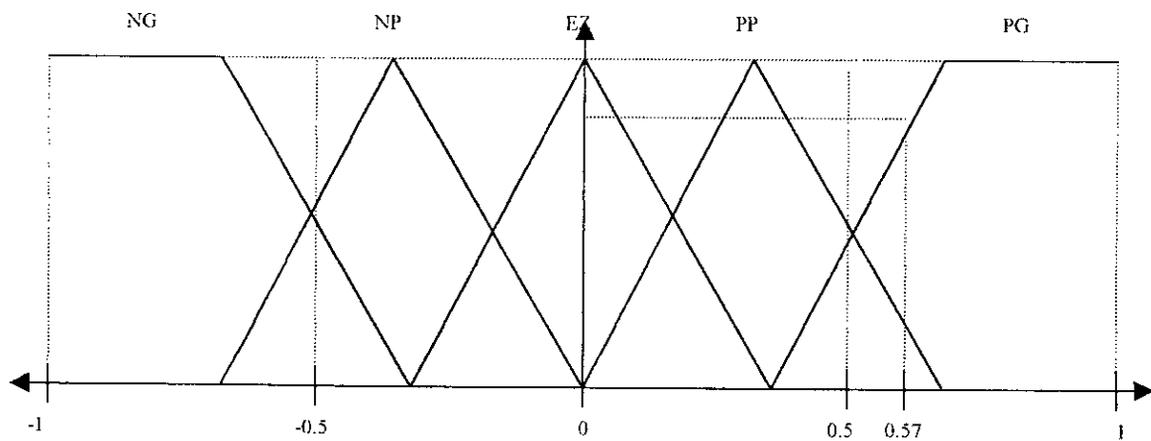


Figure 2.5 : Exemple de fuzzification

2.9 Moteur d'inférence floues :

La stratégie de réglage dépend essentiellement des inférences adoptées. Elles lient les grandeurs mesurées, qui sont les variables d'entrées x (transformées en variables linguistiques à l'aide de la fuzzification), à la variable de sortie x_R , cette dernière est également exprimée comme variable linguistique. Dans le cas du réglage par logique floue, il s'agit d'inférences avec plusieurs règles.

On peut décrire l'inférence de manière explicite par la description linguistique à l'aide d'un certain nombre de règles. Chaque règle possède une condition, précédée du symbole SI et une conclusion, action ou opération, précédée du symbole ALORS.

Une simplification ultérieure de la description des inférences s'obtient à l'aide d'une représentation graphique, appelée matrice d'inférence, cette dernière fait correspondre à chaque doublé d'ensembles de variables d'entrées un ensemble de variable de sortie, défini par une règle d'inférence, les variables d'entrées sont liées par l'opérateur ET, tandis que les variables de sorties sont à combiner par l'opérateur OU, afin d'obtenir la variable de sortie de l'inférence.

Il existe plusieurs possibilités pour réaliser ces opérateurs qui s'appliquent aux fonctions d'appartenance. On introduit alors la notion de méthode d'inférence. Elle détermine la réalisation des différents opérateurs dans une inférence, permettant ainsi un traitement numérique de cette dernière. Pour le réglage par logique floue, on utilise en général une des méthodes suivantes :

- Méthode d'inférence max-min.
- Méthode d'inférence max-prod.
- Méthode d'inférence somme-prod.

2.9.1 Méthode d'inférence max-min :

Méthode d'inférence max-min, réalise, au niveau de la condition, l'opérateur OU par la formation du maximum et l'opérateur ET par la formation du minimum.

La figure 2.6 représente graphiquement le principe de la Méthode d'inférence max-min.

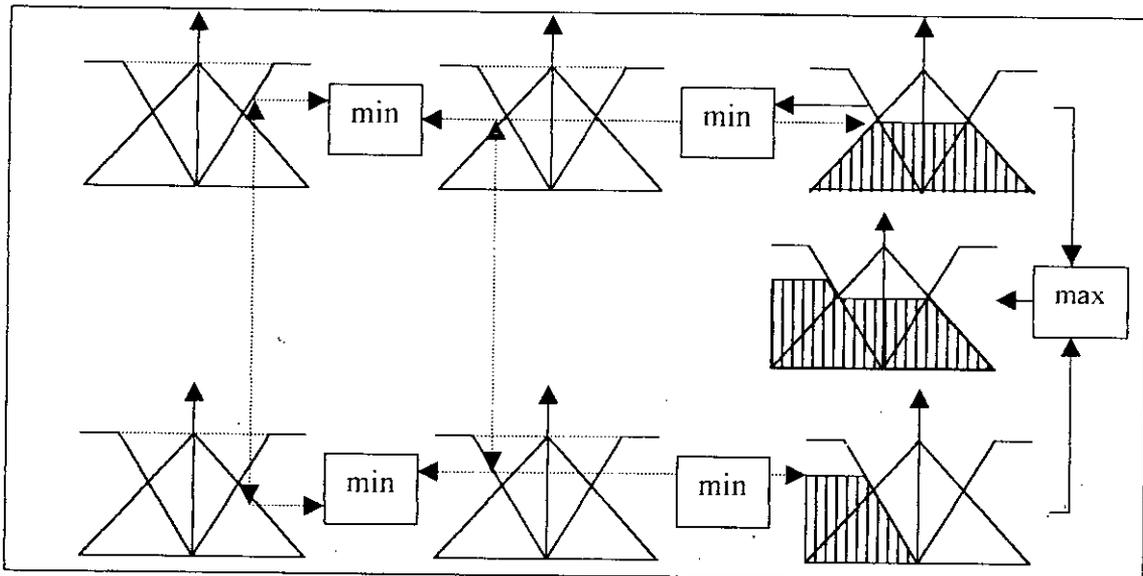


Figure 2.6 : Méthode d'inférence max-min pour deux variables d'entrée et deux règles .

2.9.2 Méthode d'inférence max-prod :

La méthode d'inférence max-prod réalise en général, au niveau de la condition, l'opérateur OU par la formation du maximum et l'opérateur Et par la formation du minimum, par contre, la conclusion dans chaque règle, introduite par ALORS, qui lie le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie x_R par l'opérateur ET, est réalisée cette fois ci par la formation du produit (minimum pour la méthode d'inférence max-min).

La représentation graphique du principe de la méthode d'inférence max-prod se trouve sur la figure 2.7.

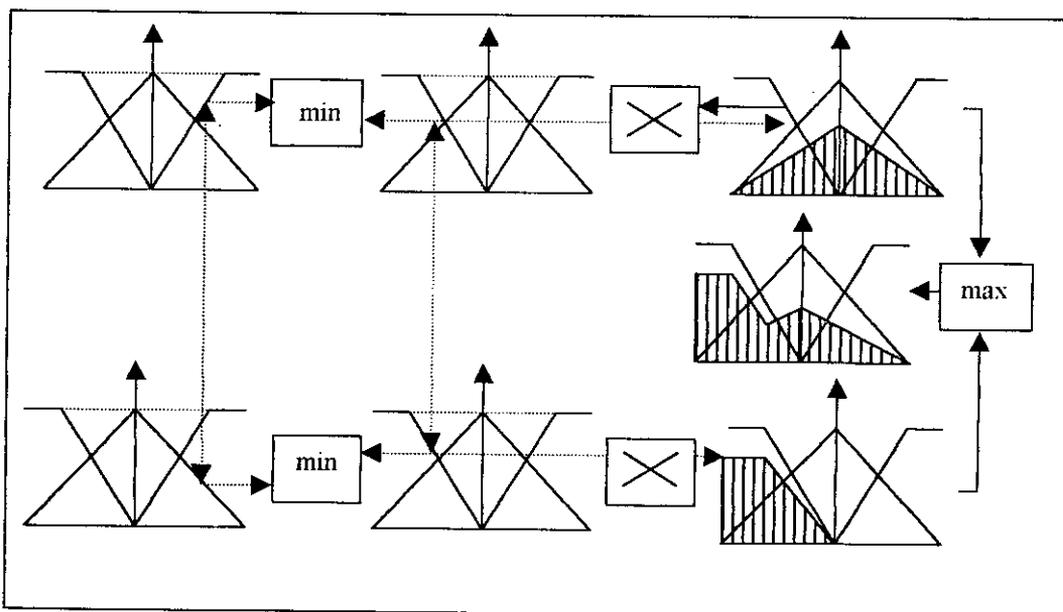


Figure 2.7 : Méthode d'inférence max-prod pour deux variables d'entrée et deux règles

2.9.3 Méthode d'inférence somme-prod :

Par opposition aux méthodes d'inférence précédentes, la Méthode d'inférence somme-prod réalise, au niveau de la condition, l'opération OU par la formation de la somme, plus précisément par la valeur moyenne, tandis que l'opération ET est réalisée par la formation du produit. La méthode d'inférence somme-prod est représentée graphiquement à la figure 2.8.

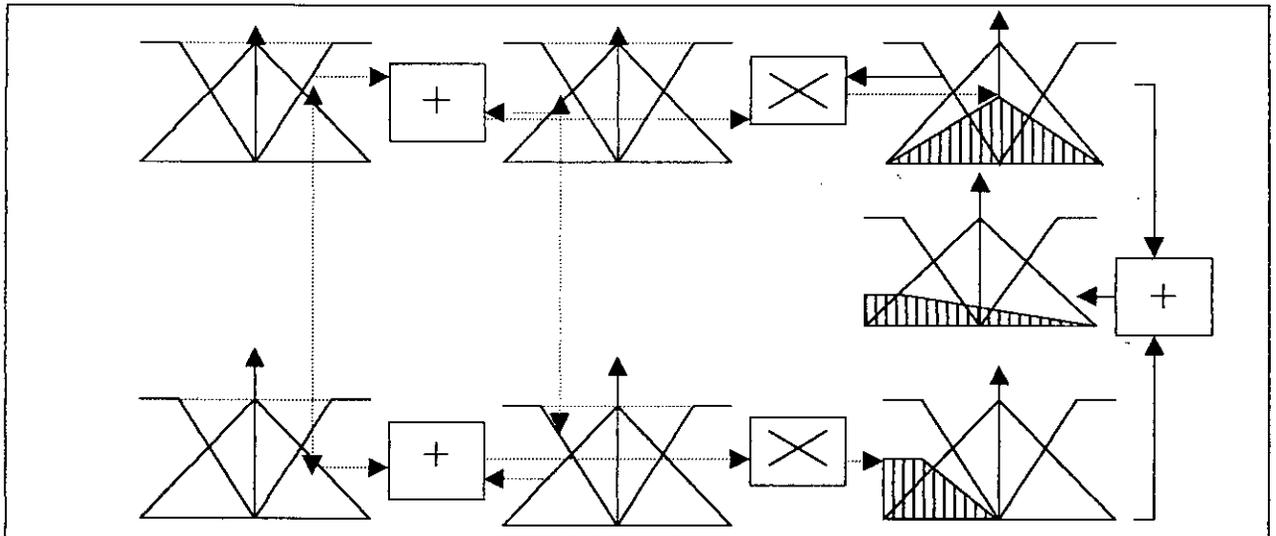


Figure 2.8 : Méthode d'inférence somme-prod pour deux variables d'entrée et deux règles

2.10 Défuzzification :

Les méthodes d'inférence fournissent une fonction d'appartenance résultante $\mu_{RES}(x_R)$ pour la variable de sortie x_R . Il s'agit donc d'une information floue, étant donné que l'organe de commande nécessite un signal de commande u_{cm} précis à son entrée, il faut prévoir une transformation de cette information floue en une information déterminée, cette transformation est appelée défuzzification.

Il existe plusieurs méthodes de défuzzification, mais la méthode la plus utilisée est la défuzzification par centre de gravité, elle consiste à calculer le centre de gravité x_G de la fonction d'appartenance $\mu_{RES}(x_R)$ résultante du moteur d'inférence floue (voir figure 2.9)

La valeur de x_G est déduite comme suit :

$$x_G = \frac{\int_{-1}^1 x \mu_{RES}(x) dx}{\int_{-1}^1 \mu_{RES}(x) dx} \quad (2.14)$$

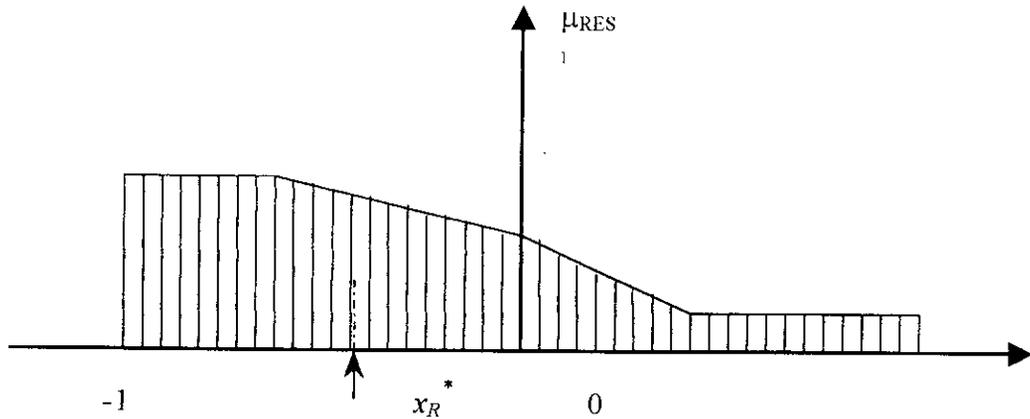


Figure 2.9 : Défuzzification par centre de gravité

IL existe une autre méthode de défuzzification, qui est la défuzzification par valeur maximum, dans cette méthode on attribue à x_G la valeur maximum de la fonction d'appartenance résultante du moteur d'inférence floue (voir figure 2.10).

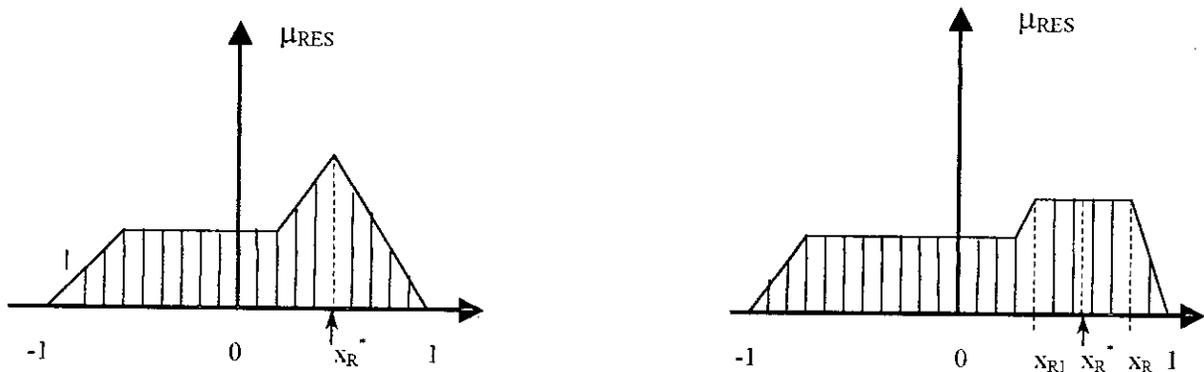


Figure 2.10 : Défuzzification par valeur maximum

2.11 Application de la commande floue :

2.11.1 Commande décentralisée :

Avant l'application de la commande floue à un système complexe il faut choisir entre deux stratégies de commande, d'abord la commande centralisée, puis la commande décentralisée en s'appuyant sur les critères suivant :

- La réalisabilité de la commande.
- Temps de calcul.

L'utilisation de la commande centralisée nécessite un nombre de règles très élevé, et aussi une bonne connaissance du système, donc un temps de calcul très grand. Par contre l'utilisation de la commande décentralisée facilite le travail, ceci en réduisant les règles à utiliser, ainsi que le temps de calcul de la commande. Le système global sera décomposé en trois sous-systèmes, chacun d'eux est commander indépendamment Les interconnexions seront considérées comme étant des perturbations.

La commande utilisée est la commande décentralisée, avec trois systèmes indépendants à réguler chacun représentant une articulation. Les interconnexions ainsi que l'effet des commandes des autres articulations seront considérées par le régulateur comme étant des perturbations externes.

2.11.2 Structure de la commande floue :

Sur la figure (2.11) sont représentés les trois régulateurs flous des trois articulations d'un robot à 3 degrés de liberté. Ces derniers sont indépendants les uns des autres, chacun à deux entrées qui sont l'erreur et la variation de l'erreur et une sortie qui est la commande à appliquer à l'articulation.

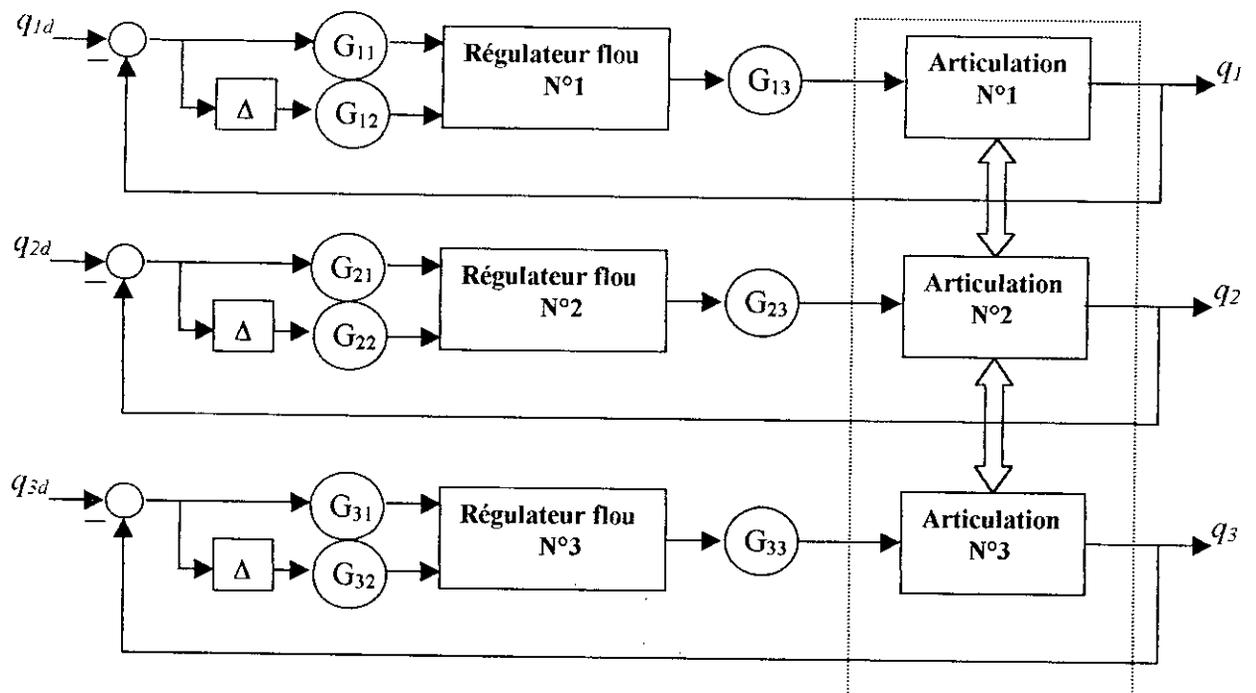


Figure 2.11: structure de la commande floue du robot

2.11.3 Régulateur flou de MAMDANI [27] :

Le schéma du régulateur de type MAMDANI est composé de :

- Un bloc de calcul de la variation de l'erreur (Δe) au cours du temps.
- Les facteurs d'échelle ou gains de normalisation associés à l'erreur (e), à sa variation (Δe), et à la commande (U).
- Un bloc de fuzzification de l'erreur et de sa variation.
- Une base de règles de contrôle flou.
- Une logique pour l'évaluation des règles de contrôle flou (inférence), pour notre cas c'est la méthode max-min qui est utilisée.
- Un bloc de défuzzification servant à convertir la commande floue en une valeur numérique.

2.11.4 Loi de commande :

La loi utilisée est fonction des entrées choisies pour notre contrôleur, en l'occurrence l'erreur et sa variation sur une période d'échantillonnage T_e :

$$U(kT_e) = f(e(kT_e), \Delta e(kT_e)) \quad (2.15)$$

Par conséquent, l'utilisation de l'ensemble des règles de décision associées donne la commande U nécessaire et qui constitue donc la sortie de notre contrôleur.

Cette commande est obtenue par lecture d'une table de décision prédéfini hors ligne. Notre choix de la loi de commande s'est porté sur la forme suivante:

$$U(kT_e) = G_u \cdot u_n(kT_e) \quad (2.16)$$

G_u : le gain associé à la commande.

2.11.5 Les gains de normalisation :

L'une des choses les plus importantes dans la commande floue est de trouver les gains de l'erreur, variation de l'erreur et de la commande ou encore gains de normalisation. En effet, ce sont ces derniers qui fixeront les performances de la commande. Il n'y a aucune méthode systématique qui donne ces paramètres d'emblée ; en effet, il faut procéder par un tâtonnement en utilisant des règles empiriques et l'expérience acquise au fil du temps.

2.11.6 La trajectoire test :

Pour tester les performances des commandes appliquées au Robot PUMA 560, on va lui imposer une trajectoire de référence, c'est la trajectoire de LEAHVY qui excite toute sa dynamique et qui consiste à faire varier dans un intervalle de temps de 1.5 secondes la position de :

- La première articulation entre -50° et 45° .
- La deuxième articulation entre -135° et -85° .
- La troisième articulation entre 135° et 30° .

Pour le robot à structure cylindrique la trajectoire de LEAHVY sera appliquée dans un intervalle de temps de 2s avec les positions suivantes :

- La première articulation entre 0 et 1.6rad
- La deuxième et la troisième articulation entre 0 et 1m.

2.11.7 Synthèse des régulateurs :

La base des règles est obtenue à partir de la matrice de Macvicar-whelan, les régulateurs des trois sous-systèmes se différencient seulement par les gains de normalisation et les gains associés aux commandes, les fonctions d'appartenance sont Gaussiennes pour les entrées et les sorties.

2.11.7.1 Régulateur flou à cinq classes :

$e \backslash \Delta e$	NB	NS	EZ	PS	PB
NB	NB	NB	NB	NS	EZ
NS	NB	NB	NS	EZ	PS
EZ	NB	NS	EZ	PS	PB
PS	NS	EZ	PS	PB	PB
PB	EZ	PS	PB	PB	PB

Tableau 2.1 : Base de règles, RLF (5×5)

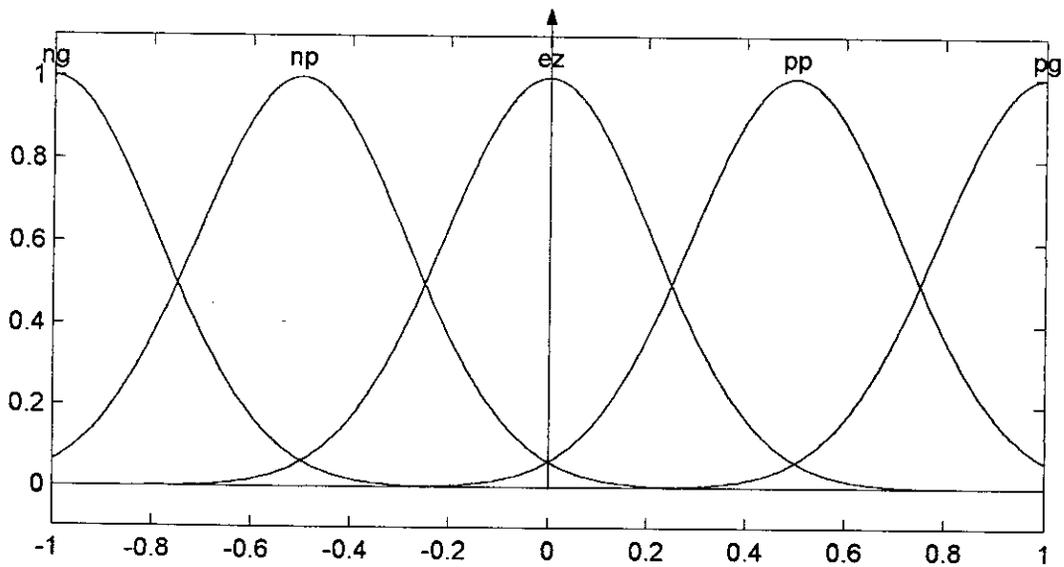


Figure 2.12 : Fonctions d'appartenances du RLF à 5 classes

2.11.7.1 Régulateur flou à sept classes :

$e \backslash \Delta e$	NB	NM	NS	EZ	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	EZ
NM	NB	NB	NB	NM	NS	EZ	PS
NS	NB	NB	NM	NS	EZ	PS	PM
EZ	NB	NM	NS	EZ	PS	PM	PB
PS	NM	NS	EZ	PS	PM	PB	PB
PM	NS	EZ	PS	PM	PB	PB	PB
PB	EZ	PS	PM	PB	PB	PB	PB

Tableau 2.2 : Base de règles, RLF (7x7)

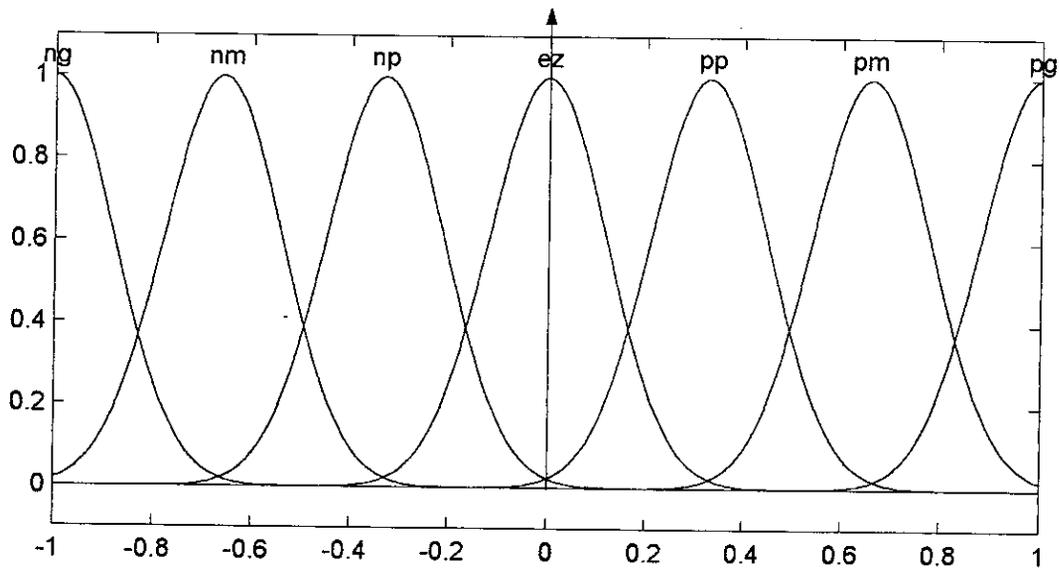


Figure 2.13 : Fonctions d'appartenances du RLF à 7 classes

2.11.7.1 Régulateur flou à neuf classes :

$e \backslash \Delta e$	NVB	NB	NS	NVS	EZ	PVS	PS	PB	PVB
NVB	NVB	NVB	NVB	NVB	NVB	NB	NS	NVS	EZ
NB	NVB	NVB	NVB	NVB	NB	NS	NVS	EZ	PVS
NS	NVB	NVB	NVB	NB	NS	NVS	EZ	PVS	PS
NVS	NVB	NVB	NB	NS	NVS	EZ	PVS	PS	PB
EZ	NVB	NB	NS	NVS	EZ	PVS	PS	PB	PVB
PVS	NB	NS	NVS	EZ	PVS	PS	PB	PVB	PVB
PS	NS	NVS	EZ	PVS	PS	PB	PVB	PVB	PVB
PB	NVS	EZ	PVS	PS	PB	PVB	PVB	PVB	PVB
PVB	EZ	PVS	PS	PB	PVB	PVB	PVB	PVB	PVB

Tableau 2.3 : Base de règles, RLF (9x9)

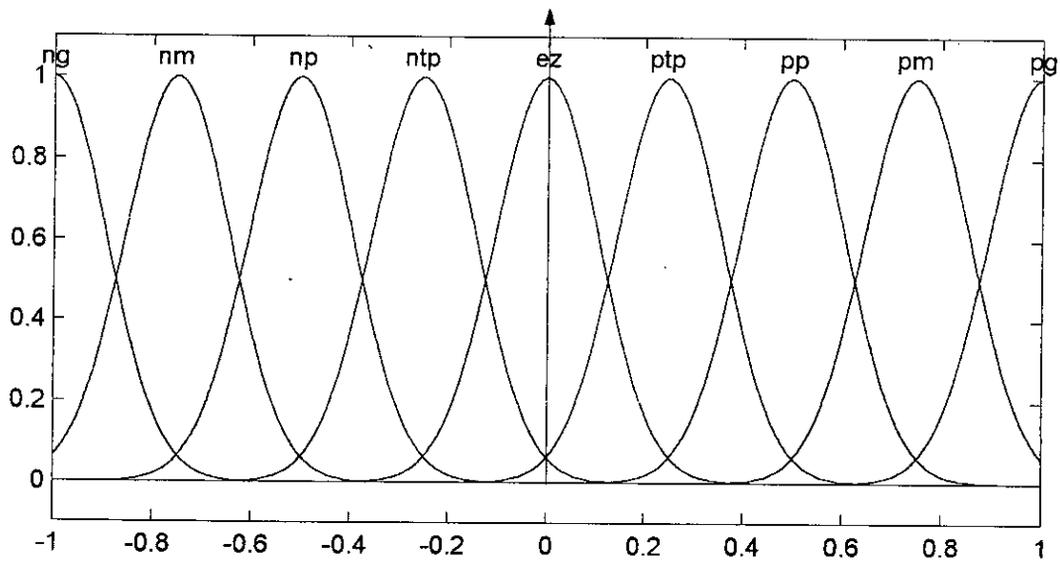


Figure 2.14 : Fonctions d'appartenances du RLF à 9 classes

2.11.8 Application de la commande floue au robot PUMA 560 :

Simulations et interprétations :

Les simulations des lois de commandes ont été réalisées par le logiciel MATLAB 5.3, par l'intermédiaire du SIMULINK. L'algorithme de résolution utilisée est celui de Runge-Kutta d'ordre 4 avec un pas de simulation de $5 \cdot 10^{-3}$ et un intervalle de simulation de $[0 \ 2.5s]$.

Les figures (2.15), (2.18) et (2.21) représentent les résultats des simulations à vide des trois articulations du PUMA 560, avec les trois régulateurs flous (5, 7 et 9 classes d'appartenance), avec comme trajectoire de référence la trajectoire de LEAHVY. De l'observation des figures nous remarquons que chaque articulation suit convenablement sa trajectoire de référence dans un intervalle d'erreur acceptable de l'ordre de $5 \cdot 10^{-3}$ degrés en régime dynamique, mais persistante en régime statique. Les commandes sont lisses et donc physiquement réalisables.

Les figures (2.16), (2.17), (2.19), (2.20), (2.22) et (2.23) représentent les résultats des simulations des trois articulations obtenus avec l'introduction de charges (5 et 10kg respectivement). Le système reste toujours performant malgré un léger accroissement des erreurs et notamment l'apparition d'un pic de démarrage de la 3^{ème} commande.

Pour les tests de robustesse, qui consiste en un lâché de charge et une rupture de la 2^{ème} commande, nous avons pris le régulateur ayant donné les meilleurs résultats du point de vue énergie d'erreur conformément au tableau ci dessous :

L'essai effectué		5 classes d'appartenances	7 classes d'appartenances	9 classes d'appartenances
A vide	e_1	$6.905 \cdot 10^{-7}$	$2.075 \cdot 10^{-6}$	$4.138 \cdot 10^{-6}$
	e_2	$2.164 \cdot 10^{-6}$	$2.396 \cdot 10^{-6}$	$3.349 \cdot 10^{-6}$
	e_3	$8.449 \cdot 10^{-6}$	$1.322 \cdot 10^{-5}$	$1.537 \cdot 10^{-5}$
	Total	$1.13 \cdot 10^{-5}$	$1.76 \cdot 10^{-5}$	$2.28 \cdot 10^{-5}$
A charge (10kg)	e_1	$1.569 \cdot 10^{-6}$	$4.612 \cdot 10^{-6}$	$1.085 \cdot 10^{-5}$
	e_2	$5.489 \cdot 10^{-6}$	$7.157 \cdot 10^{-6}$	$1.085 \cdot 10^{-5}$
	e_3	$1.406 \cdot 10^{-4}$	$2.566 \cdot 10^{-4}$	$3.150 \cdot 10^{-4}$
	Total	$1.47 \cdot 10^{-4}$	$2.68 \cdot 10^{-4}$	$3.36 \cdot 10^{-4}$

Avec e_i : énergie de l'erreur de la $i^{\text{ème}}$ articulation.

En classant les résultats par ordre décroissant de l'énergie, nous aurons respectivement le régulateur à 9 classes suivi de celui à 7 classes et enfin le régulateur à 5 classes d'appartenance. Il apparaît clairement que le régulateur à 5 classes d'appartenance est plus performant du point de vue énergie de l'erreur.

- **Test avec lâché de charge :**

Pour ce test nous avons muni le robot d'une charge de 10kg qui sera lâchée après un temps de 0.75s.

La figure (2.24) représente le résultat de simulations obtenues par ce test, on voit qu'après le lâché de la charge (10kg), de grandes variations apparaissent, mais grâce à la rapidité du système les commandes reviennent à leurs états de stabilité et on peut dire que le test est réussi sur le régulateur.

- **Test avec rupture de commande :**

La figure (2.25) représente le résultat de la simulation des trois articulations pour le régulateur à 5 classes d'appartenance, avec défaillance de l'une des trois commandes pendant son mouvement, en l'occurrence la commande de la deuxième articulation, à l'instant 0.75s, on a ainsi obtenu une stabilité de la 1^{ère} et la 3^{ème} articulation malgré la divergence de la 2^{ème} articulation.

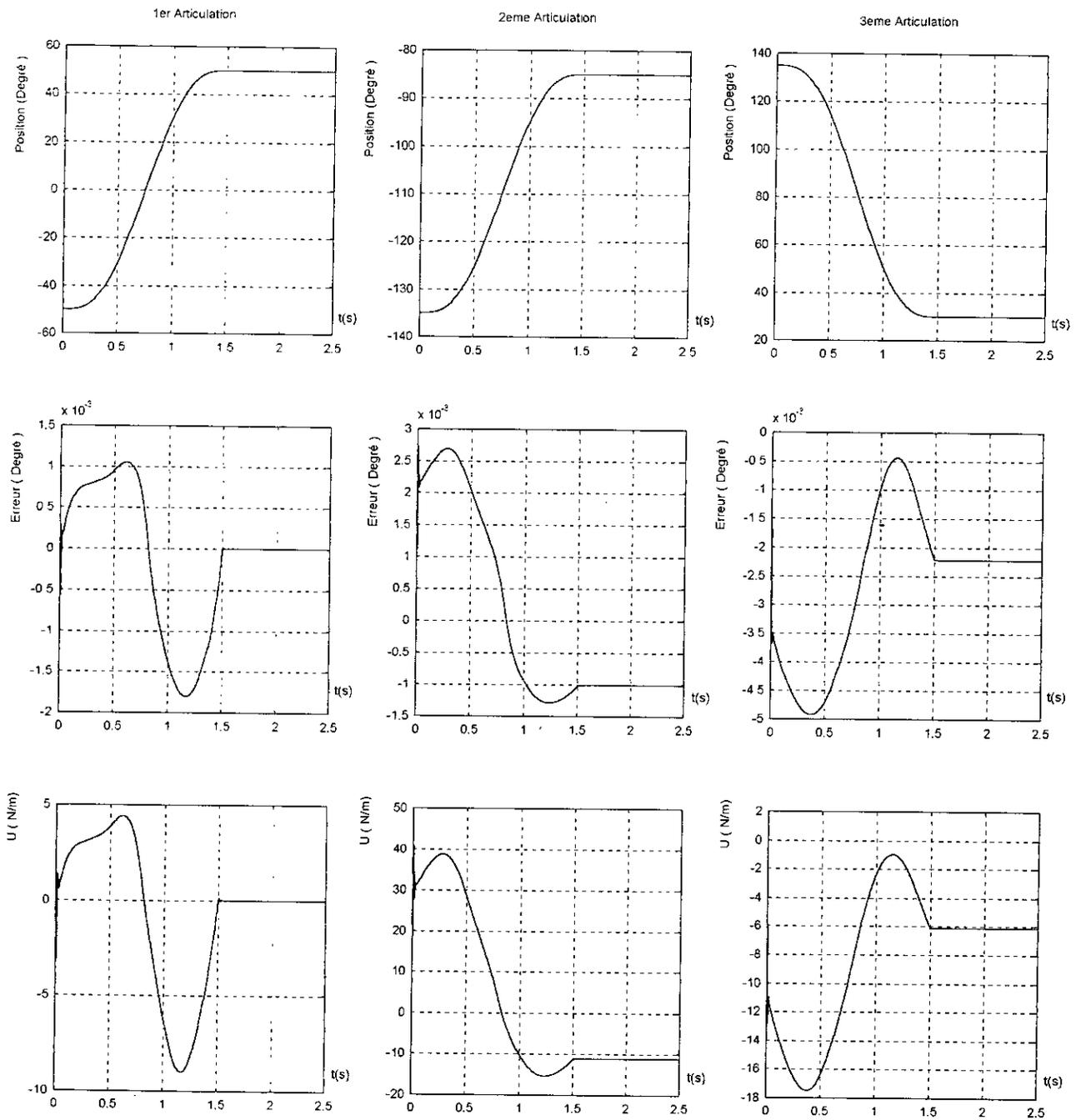


Figure (2.15) : Application de la commande floue (5 classes) au PUMA 560
Poursuite à vide

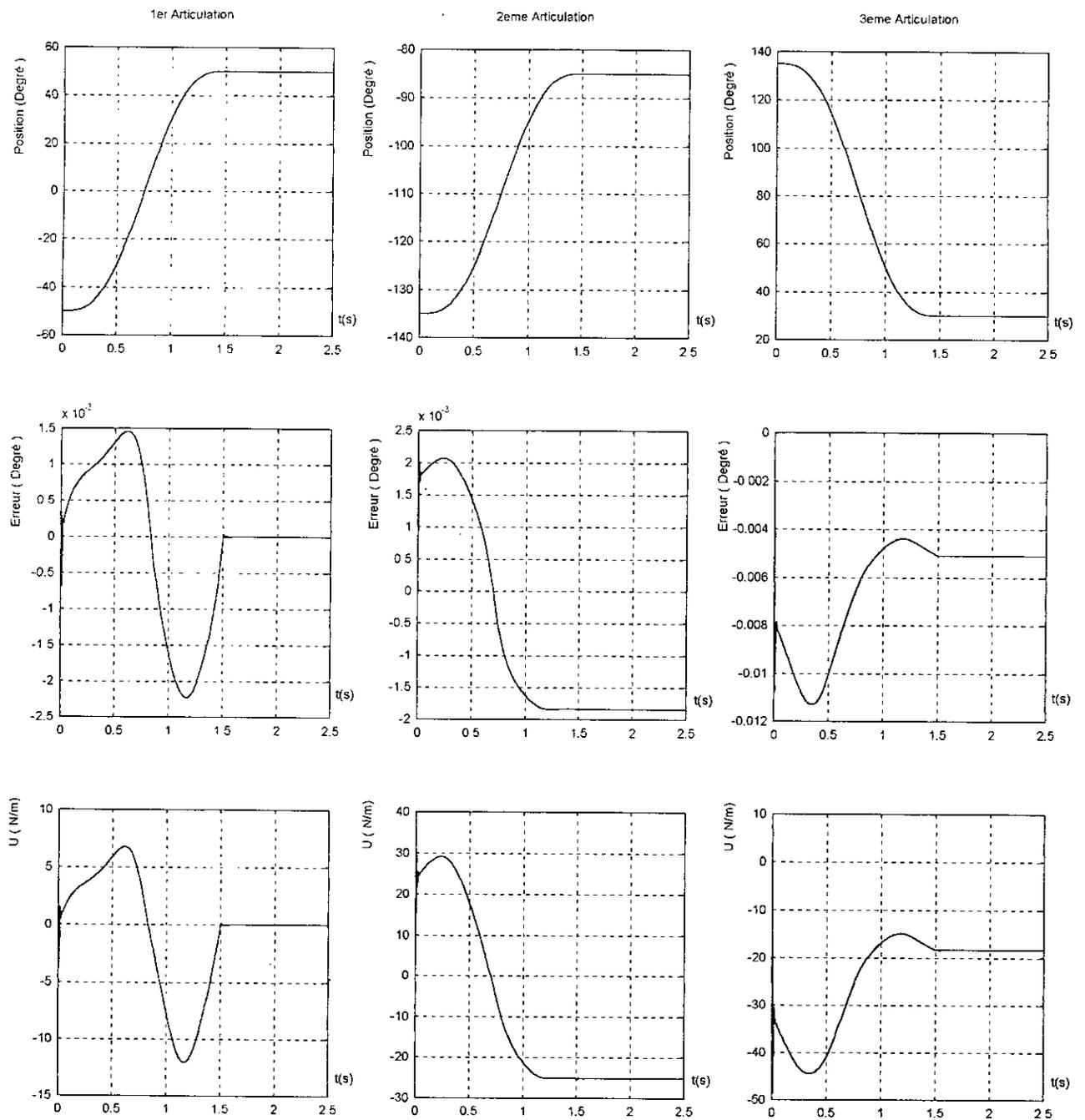


Figure (2.16) : Application de la commande floue (5 classes) au PUMA 560
Poursuite avec effet de la charge $m_0 = 5\text{kg}$

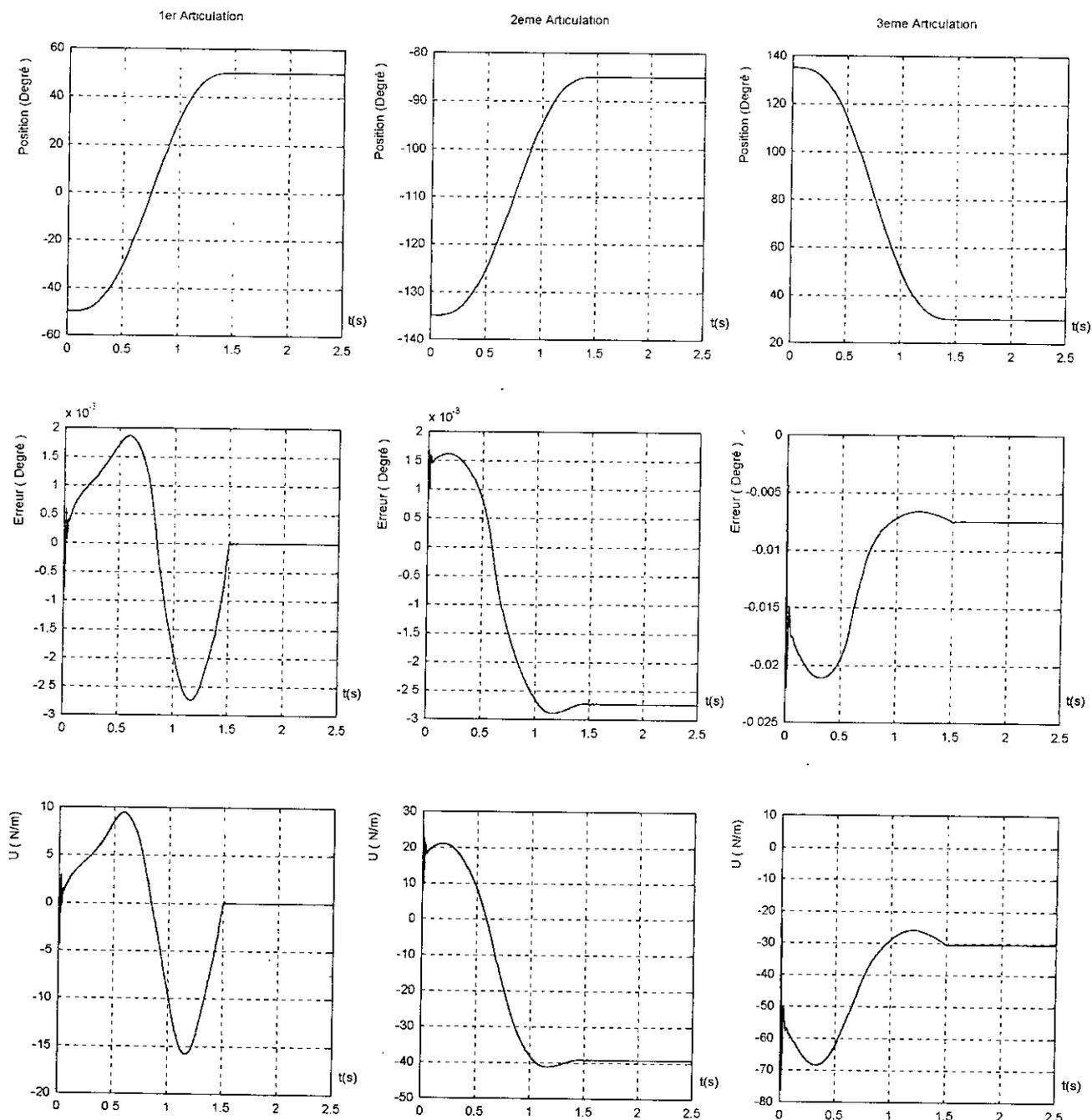


Figure (2.17) : Application de la commande floue (5 classes) au PUMA 560
Poursuite avec effet de la charge $m_0 = 10\text{kg}$

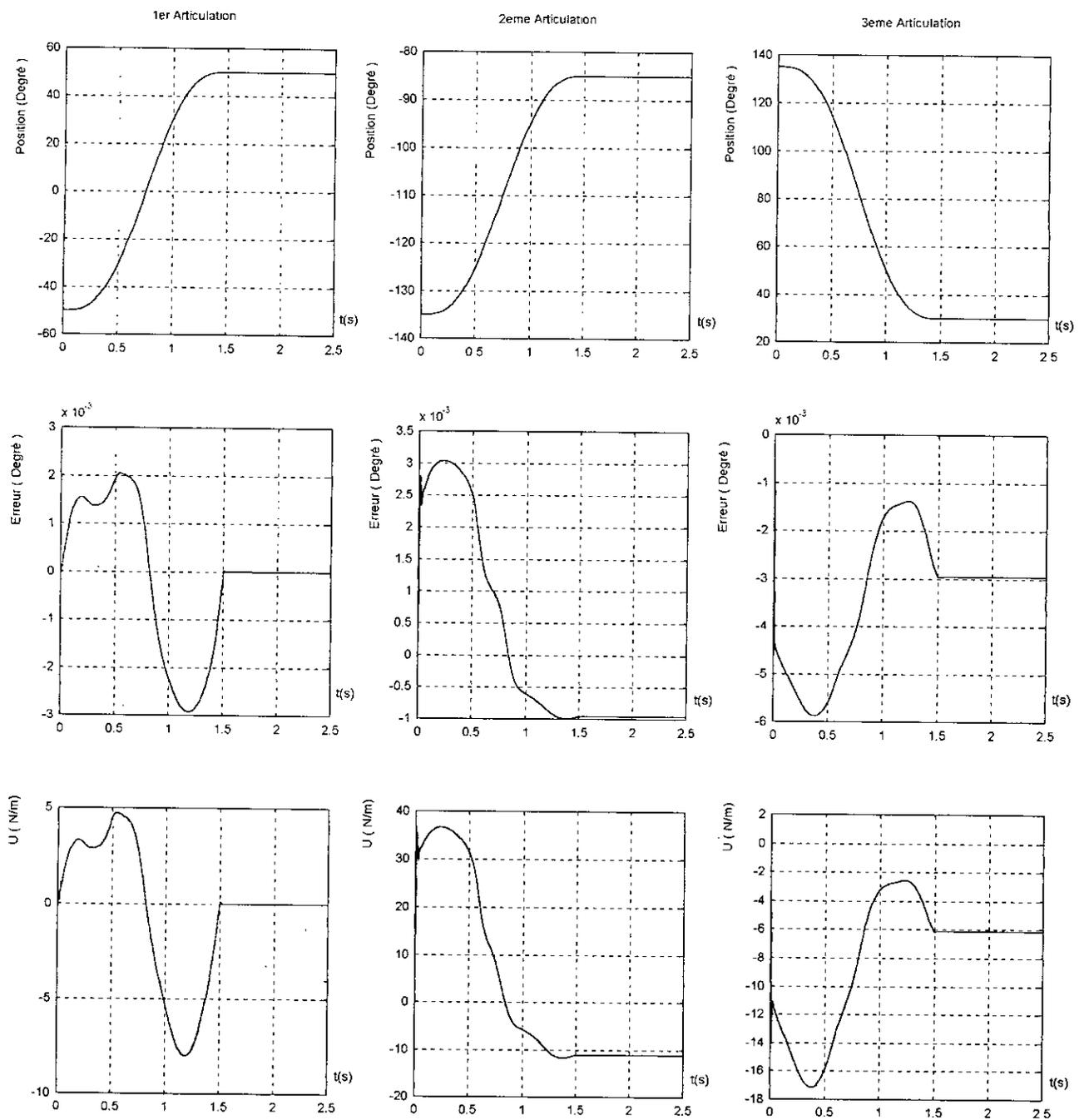


Figure (2.18) : Application de la commande floue (7 classes) au PUMA 560
Poursuite à vide

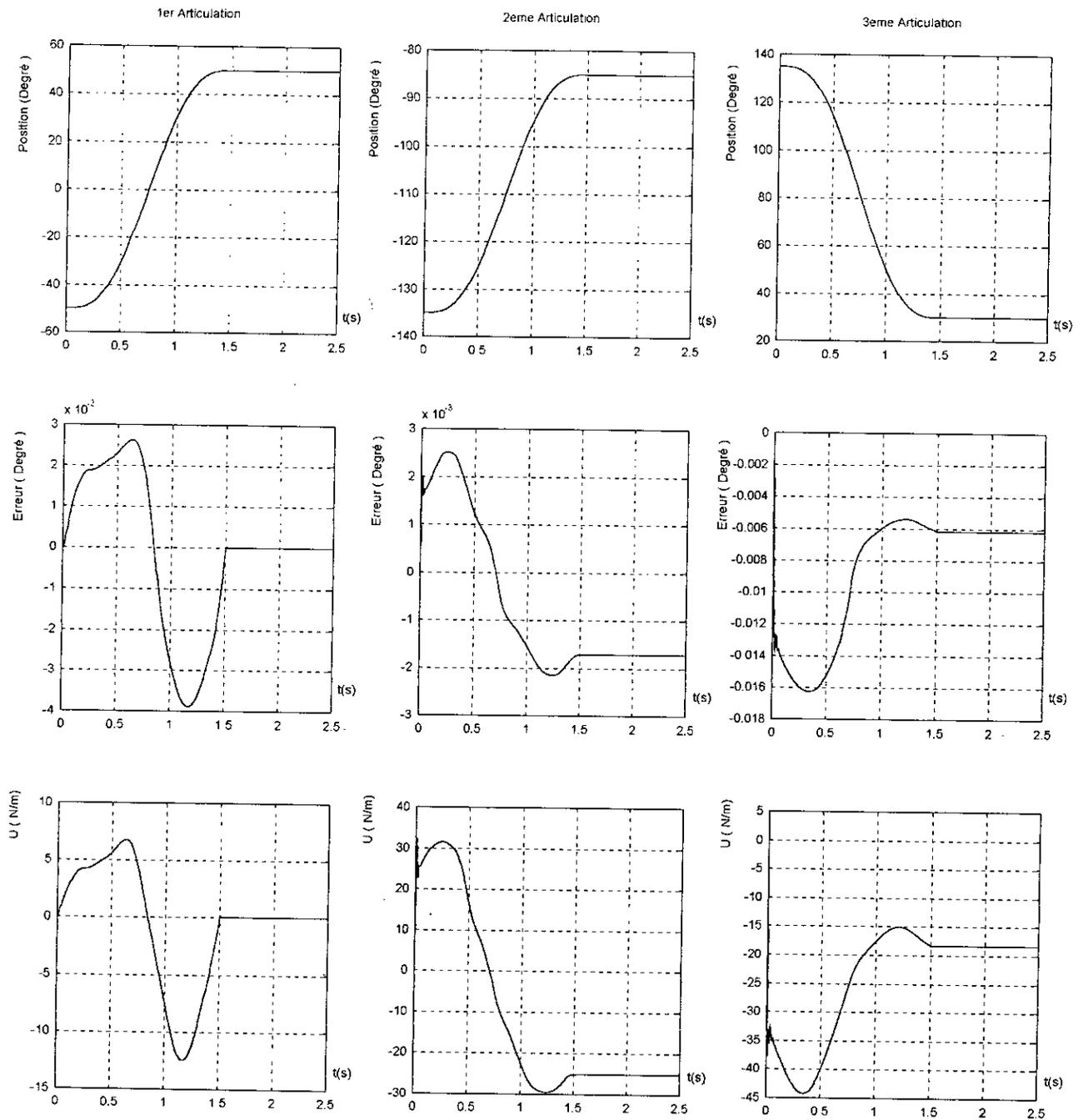


Figure (2.19) : Application de la commande floue (7 classes) au PUMA 560
Poursuite avec effet de la charge $m_0 = 5\text{kg}$

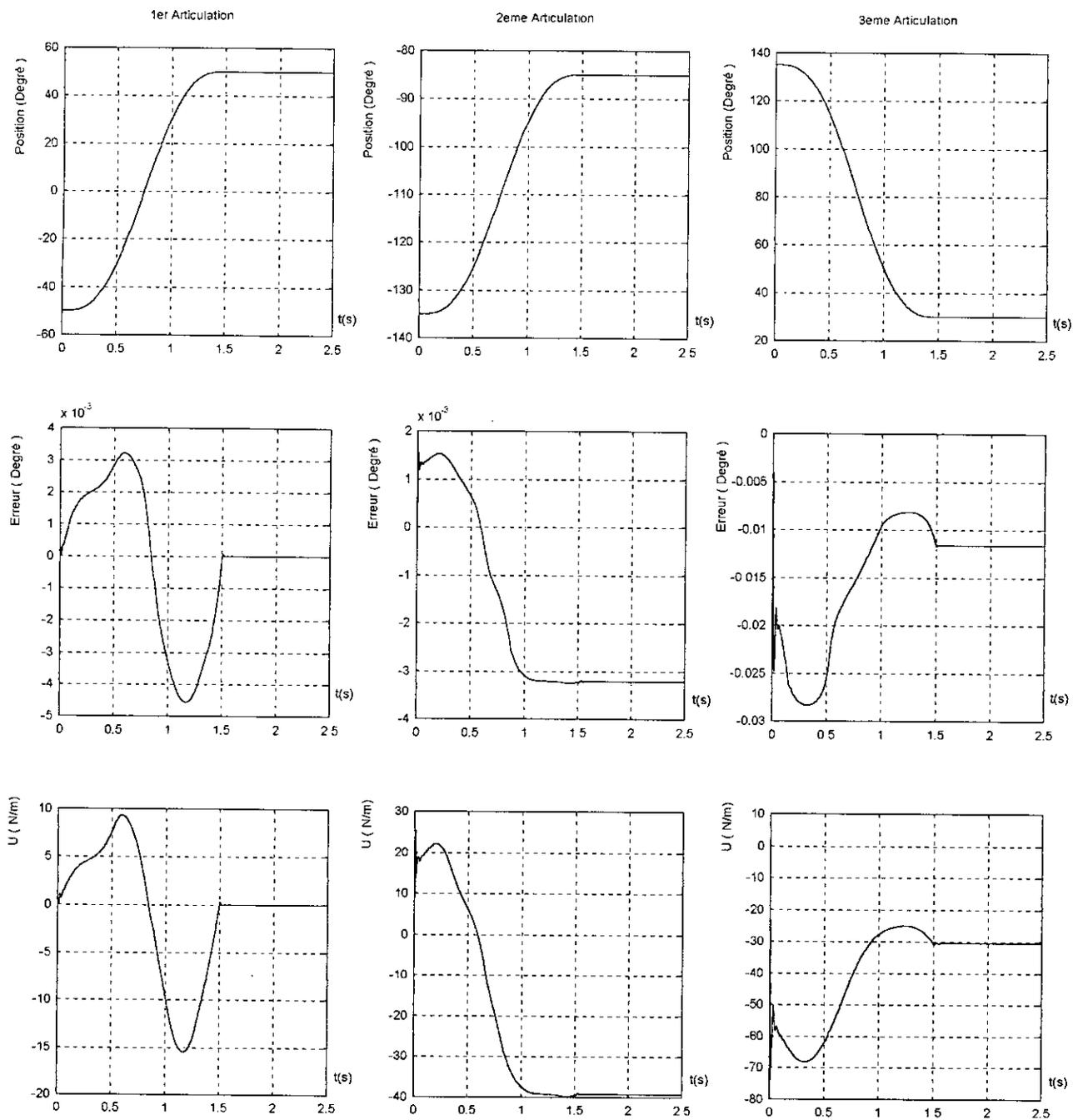


Figure (2.20) : Application de la commande floue (7 classes) au PUMA 560
Poursuite avec effet de la charge $m_0 = 10\text{kg}$

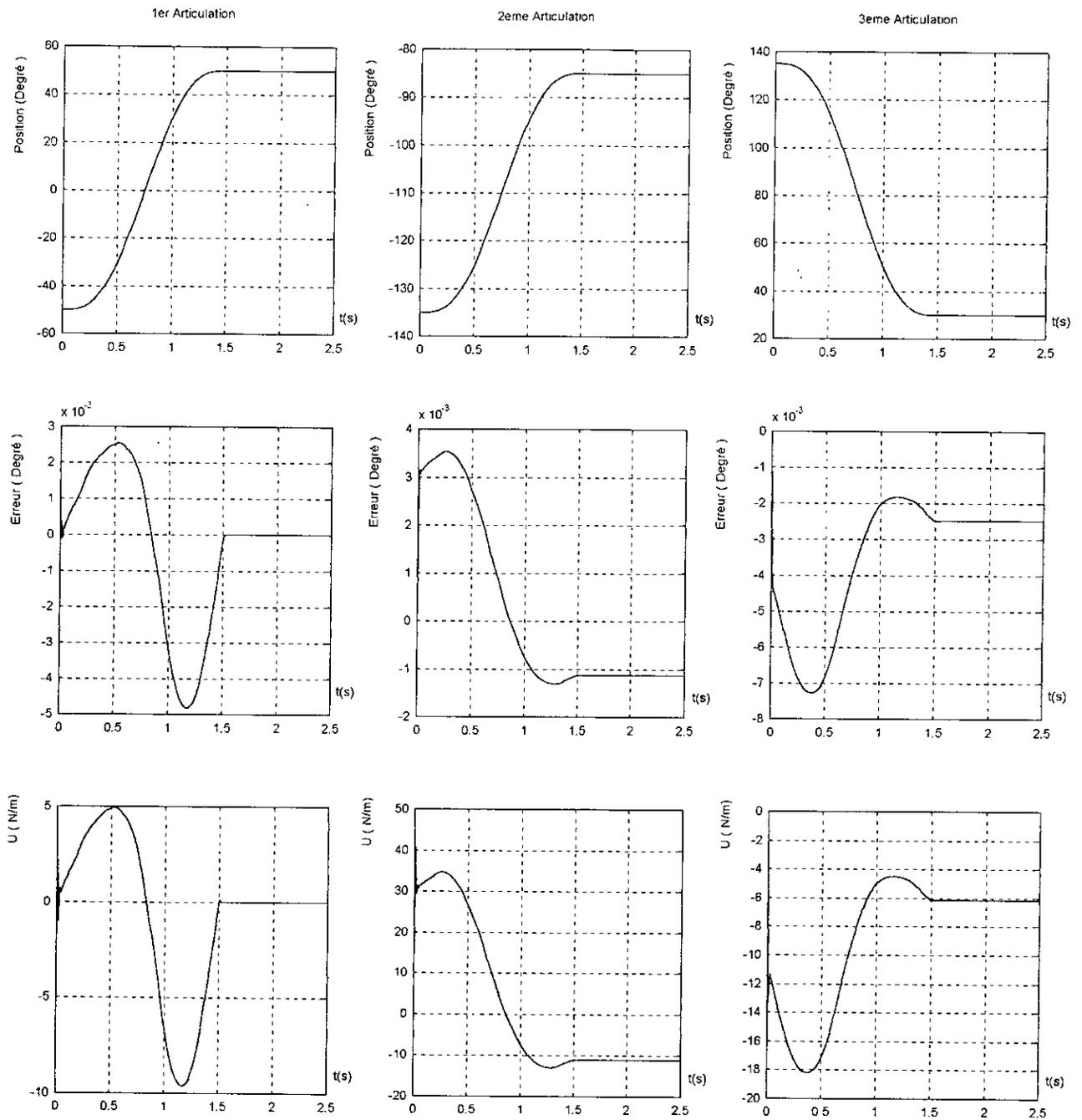


Figure (2.21) : Application de la commande floue (9 classes) au PUMA 560
Poursuite à vide

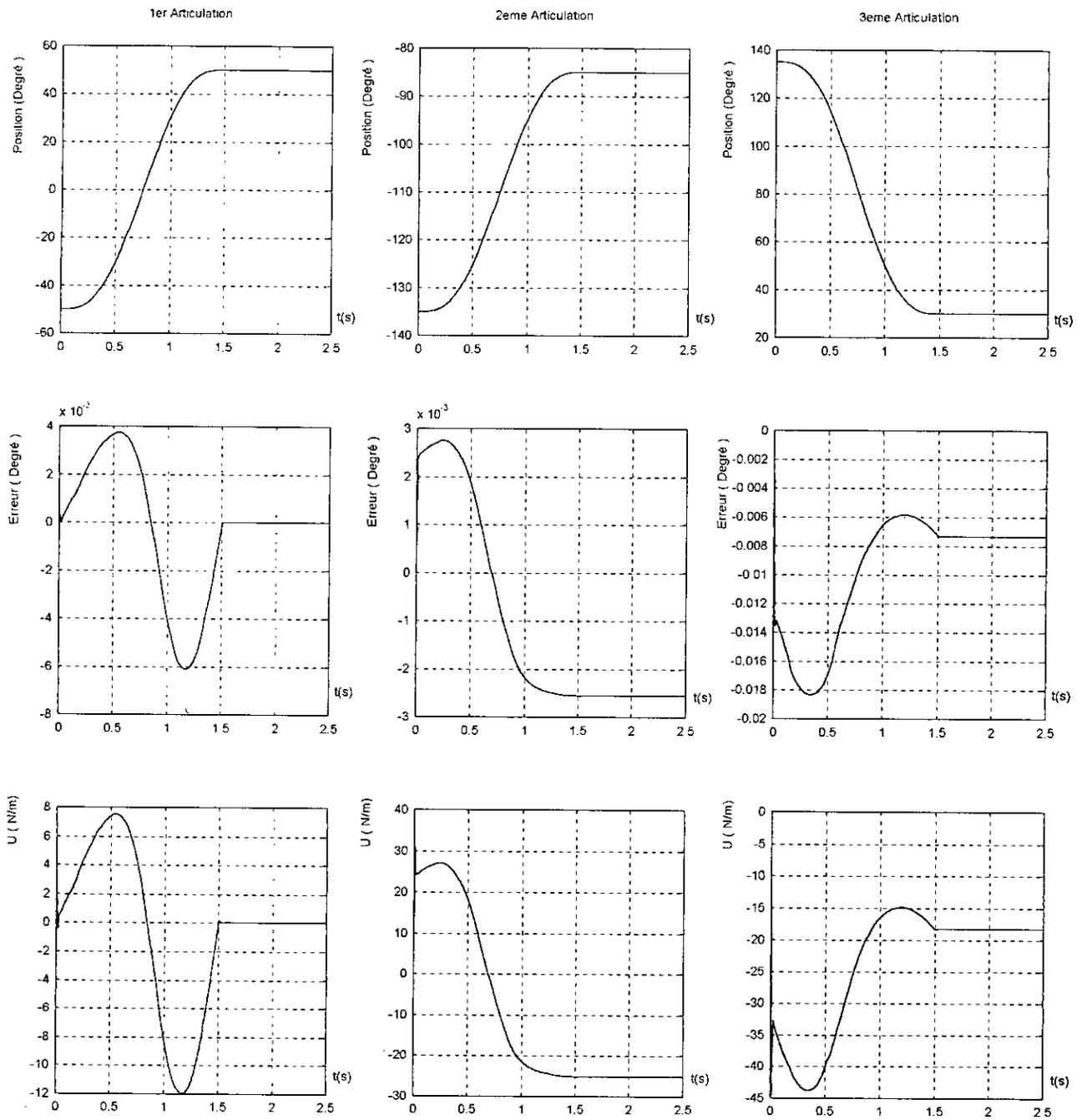


Figure (2.22) : Application de la commande floue (9 classes) au PUMA 560
Poursuite avec effet de la charge $m_0 = 5\text{kg}$

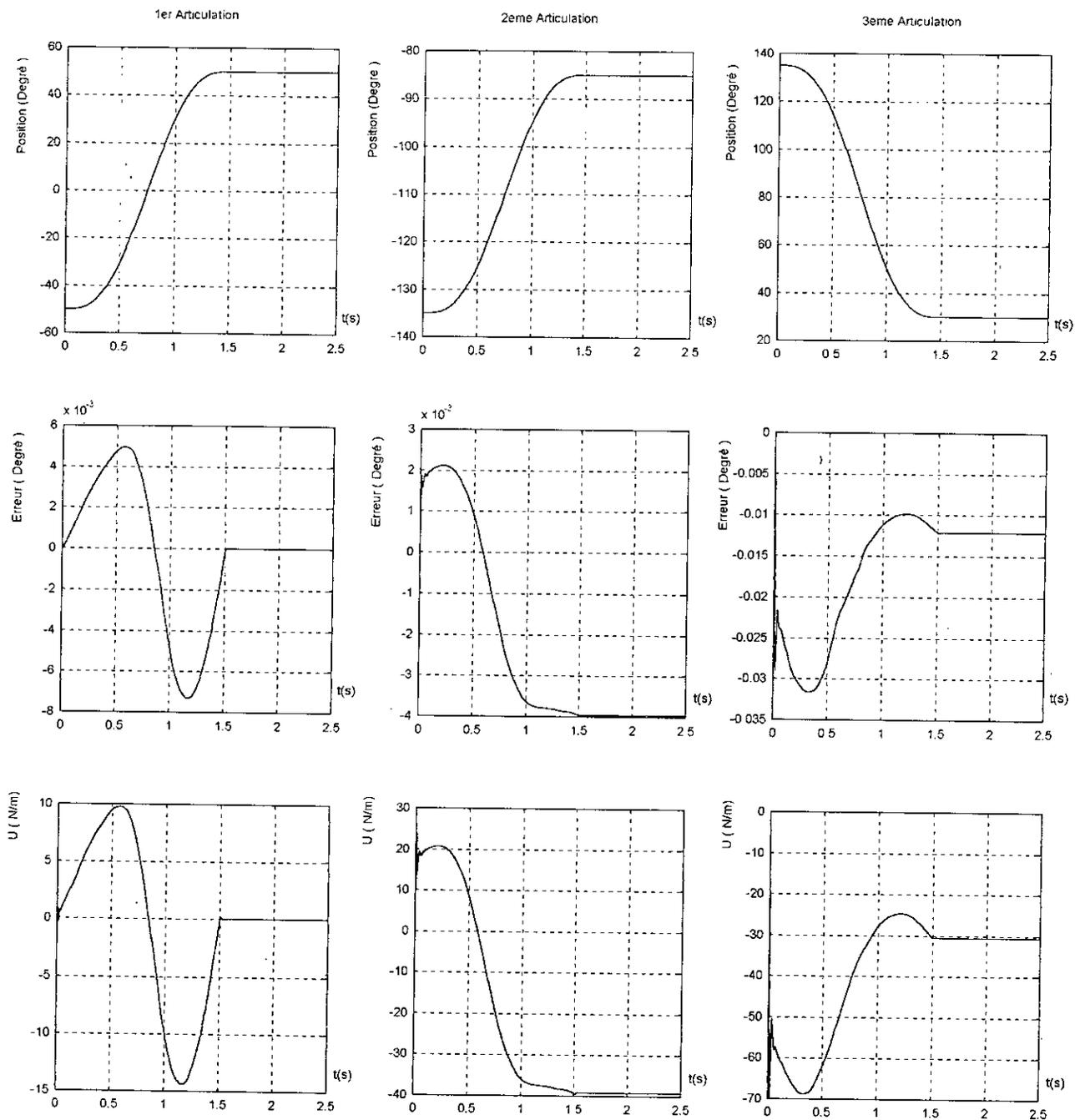


Figure (2.23) : Application de la commande floue (9 classes) au PUMA 560
Poursuite avec effet de la charge $m_0 = 10\text{kg}$

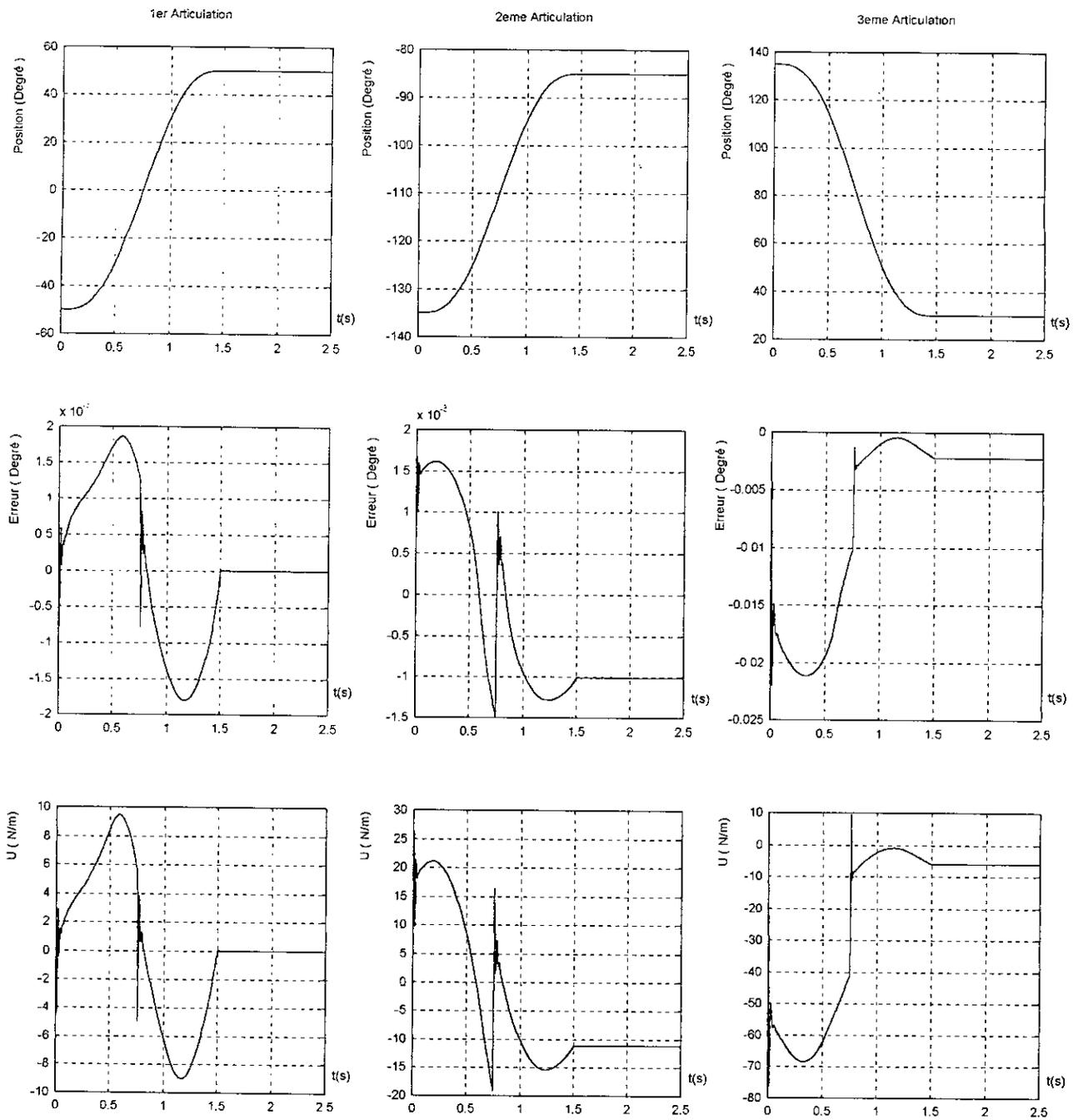


Figure (2.24) : Application de la commande floue (5 classes) au PUMA 560
Poursuite avec effet de la charge $m_0 = 10\text{kg}$ et la décharge à l'instant $t=0.75\text{s}$

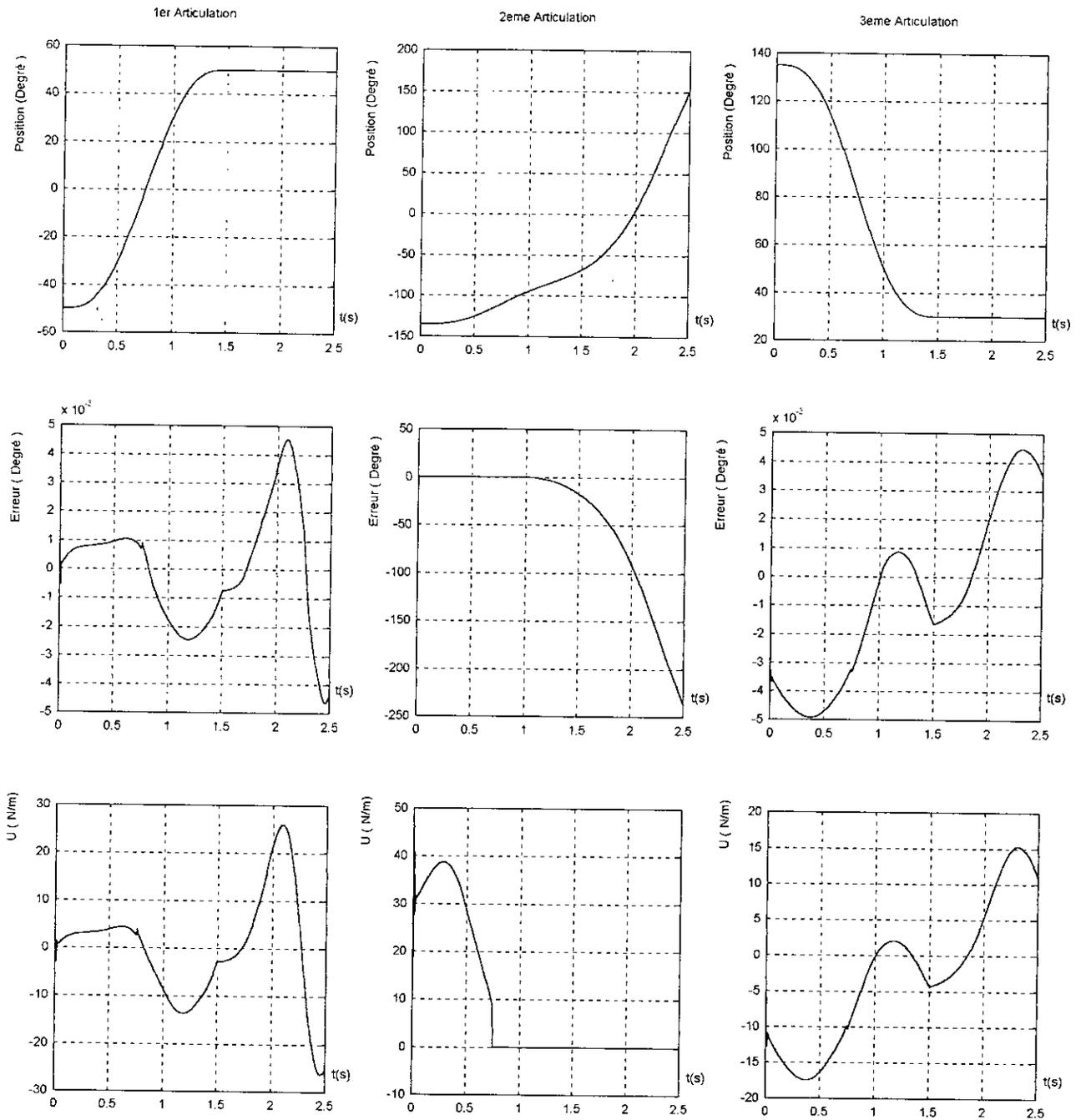


Figure (2.25) : Application de la commande floue (5 classes) au PUMA 560
Poursuite avec rupture de la deuxième commande à l'instant $t=0.75s$

2.11.9 Etude de l'influence des gains de normalisation sur le système :

Pour trouver les gains de normalisation de notre régulateur flou, nous avons effectué un tâtonnement, en partant des combinaisons de gains donnant de faux résultats, c'est à dire des commandes instables et des erreurs considérables, vers le meilleur résultat possible et jusqu'à trouver une combinaison de gains compatible avec le système.

Pour mieux saisir l'influence de ces gains sur le système, nous avons procédé par la démarche inverse, c'est à dire à partir des résultats obtenus, faire varier les gains de normalisation l'un après l'autre et un seul à la fois, puis observer le comportement des commandes et des variations des énergies des erreurs, tant pour l'essai à vide qu'en charge.

Gains des erreurs:

Première articulation :

En augmentant ce gain, on aura l'apparition d'oscillation en régime dynamique des trois commandes, qui grandissent au fur et à mesure. Par contre en diminuant ce gain on aura toujours un bon système, avec des commandes stables. Selon le graphe ci-dessus l'énergie de l'erreur diminue, jusqu'à la valeur $G_e=150$, elle augmente au-delà.

Deuxième articulation :

Quand on augmente ce gain, le système reste stable jusqu'à une certaine limite qui est aux environs de 180, au-delà on aura d'abord apparition de petites oscillations en régime dynamique, puis croissantes tout au long des commandes. A l'inverse quand on diminue le gain on aperçoit la naissance de pics en début des commandes qui grandissent au fur et à mesure.

Ainsi et d'après le graphe, on peut dire que l'énergie est une fonction décroissante.

Troisième articulation :

Lors de l'essai en charge, quand on augmente le gain on remarque l'apparition d'oscillations en régime dynamique alors que celles-ci n'apparaissent qu'en régime statique lors de l'essai à vide. Et plus on augmente le gain, les oscillations apparaissent pour les deux régimes, tant à vide qu'en charge.

A l'inverse et pour l'essai à vide, lorsqu'on diminue le gain le système reste stable. La plus petite énergie de l'erreur est obtenue pour un gain de 50, au-delà elle augmente. Contrairement, à l'essai en charge, ou la courbe de l'énergie est une fonction décroissante.

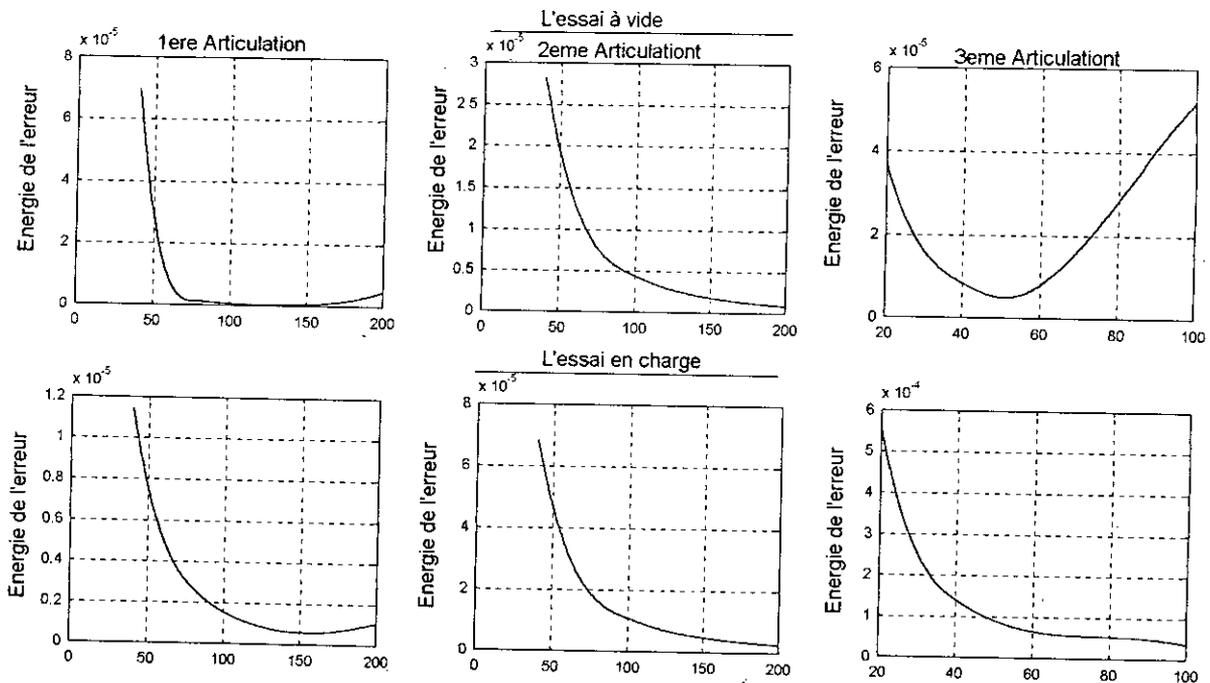


Figure 2.26 : Variation des gains des erreurs

Gains des variations des erreurs :

Première articulation :

Quand on augmente le gain, on observe d'abord de grandes oscillations en régimes dynamiques pour la 1^{ère} et la 2^{ème} commande et de petites oscillations en régime transitoire de la 3^{ème} commande et si on continue l'augmentation de ce gain on observe de fortes oscillations dans les deux régimes (dynamique et statique) et ceci pour les 3 commandes. En diminuant ce gain on aura apparition de petites oscillations en régime dynamique pour les deux premières commandes.

D'après le graphe ci-dessous on peut dire que l'énergie de l'erreur est une fonction décroissante du gain.

Deuxième articulation :

On observe une apparition de pics sur les commandes quand on augmente ou diminue ce gain. Mais si on augmente trop le gain les commandes deviennent instables avec apparition d'oscillation tout au long de l'intervalle de simulation.

La figure ci-dessous montre que que l'énergie de l'erreur est fonction décroissante du gain.

Troisième articulation :

Quand on diminue ce gain on a une bonne réponse pour l'essai à vide, alors que pour l'essai en charge on observe de petites oscillations pour la 1^{ère} commande et des grandes oscillations pour les deux autres. Quand on diminue le gain, toutes les commandes deviennent complètement instables.

Le courbe de l'énergie de l'erreur est une fonction croissante du gain.

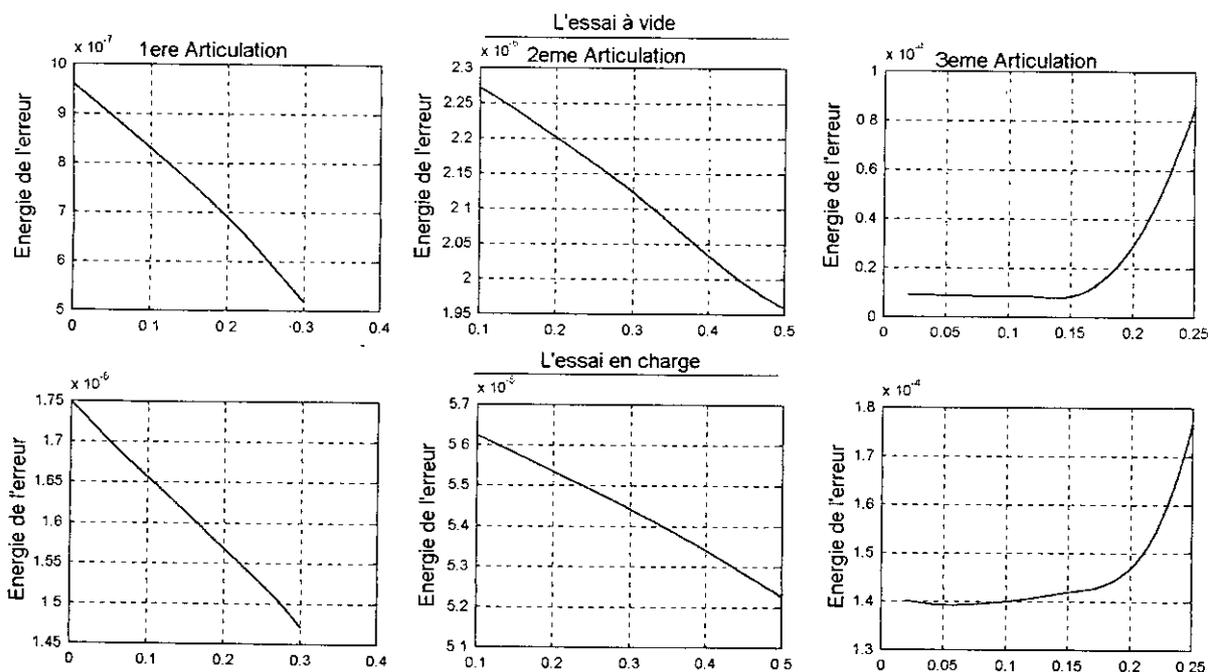


Figure 2.27 : Courbe des gains de variation des erreurs

Gains des commandes :

Première articulation :

L'augmentation de ce gain provoque de grandes oscillations en régime dynamique des deux premières commandes et de petites oscillations en régime dynamique de la dernière commande. En diminuant le gain les commandes restent stables jusqu'à une certaine valeur, au delà il y a apparition d'oscillations. pour l'essai à vide et des saturations des commandes pour l'essai en charge.

Le graphe ci-dessous montre que l'énergie de l'erreur diminue jusqu'à une certaine valeur du gain (environ 90), au delà elle augmente.

Deuxième articulation :

En faisant augmenter progressivement ce gain, on observe une apparition de pics pour les commandes 2 et 3, puis apparition d'oscillations en régime dynamique lors de l'essai à vide et instabilité des commandes pour l'essai en charge, puis les commandes deviennent instables pour les deux essais. En diminuant ce gain on observe d'abord des pics jusqu'aux environs de 50, au delà on observe des saturations des commandes.

Du graphe ci-dessous on voit que l'énergie de l'erreur diminue jusqu'à une valeur du gain (environ 160) et augmente au delà.

Troisième articulation :

Lorsqu'on augmente progressivement le gain on observe d'abord l'apparition d'oscillation en régime statique, puis elles se propageront en régime dynamique, puis les commandes deviennent complètement instable. En diminuant le gain on abouti à une saturation des commandes.

On observe du graphe ci-dessous que l'énergie de l'erreur est une fonction décroissante du gain.

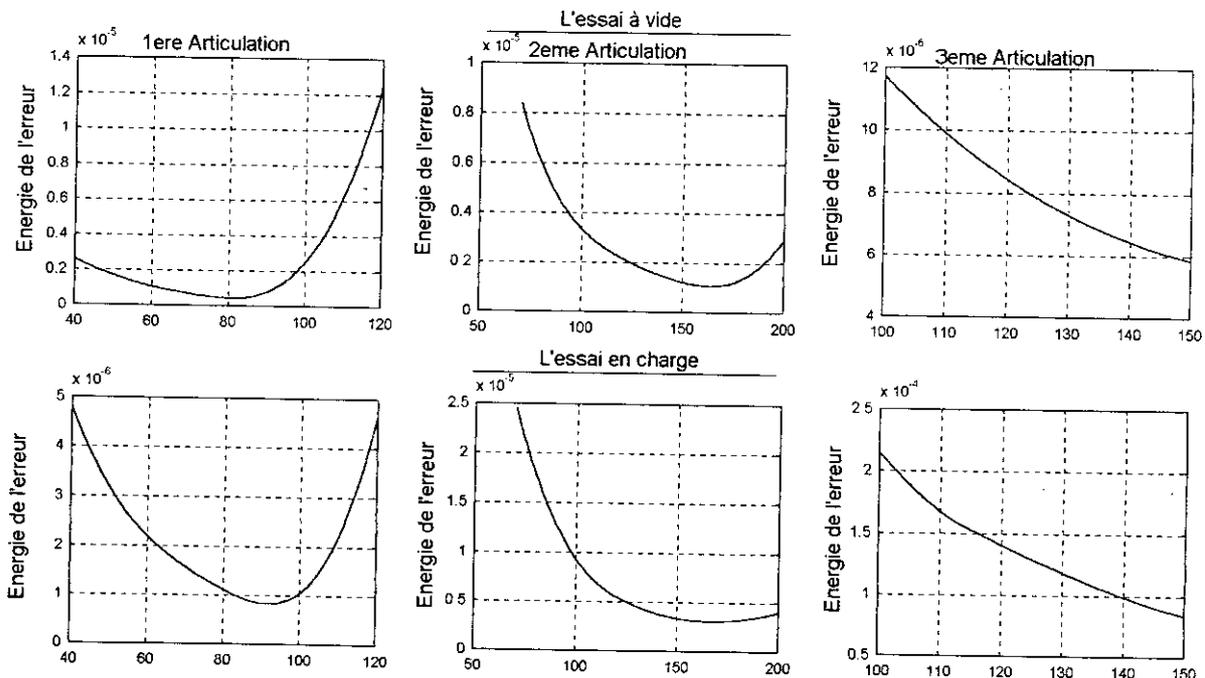


Figure 2.28 : Variation des gains des commandes

2.11.10 Application de la commande au robot à structure cylindrique :

Pour l'application de la commande floue sur le robot cylindrique, nous reprendrons les mêmes régulateurs classes pour le robot PUMA 560, on l'occurrence 5, 7 et 9.

- **Interprétations et simulations :**

Nous avons réalisé les mêmes simulations des lois de commande et utilisé le même l'algorithme de résolution qu'avec PUMA 560.

Les figures (2.26), (2.29) et (2.32) représentent les résultats des simulations des trois articulations pour les trois régulateurs et en utilisant toujours la trajectoire de LEAHVY comme trajectoire de référence à vide. On obtient des commandes admissible et réalisable et des erreurs de poursuite acceptable mais avec persistance des erreurs statique.

Les figures (2.27), (2.28), (2.30), (2.31), (2.33) et (2.34) représentent les résultats des simulations des trois articulations en introduisant des charges de 5 et 10kg. Les commandes restent acceptables et réalisables, malgré la variation de la charge.

Comme pour le robot PUMA 560, on a effectué une comparaison entre les trois régulateurs flous (5, 7 et 9 classes d'appartenance), en se basant sur le même critère, c'est à dire l'énergie de l'erreur. Nous avons obtenu le tableau comparatif suivant :

L'essai effectué		5 classes d'appartenance	7 classes d'appartenance	9 classes d'appartenance
A vide	e_1	$1.736 \cdot 10^{-7}$	$1.98 \cdot 10^{-7}$	$1.673 \cdot 10^{-7}$
	e_2	$3.223 \cdot 10^{-6}$	$9.225 \cdot 10^{-7}$	$2.386 \cdot 10^{-6}$
	e_3	$6.075 \cdot 10^{-7}$	$3.064 \cdot 10^{-7}$	$8.562 \cdot 10^{-7}$
	Total	$4.00 \cdot 10^{-6}$	$1.73 \cdot 10^{-6}$	$3.14 \cdot 10^{-6}$
A charge (10kg)	e_1	$4.492 \cdot 10^{-7}$	$4.841 \cdot 10^{-7}$	$4.931 \cdot 10^{-7}$
	e_2	$7.087 \cdot 10^{-6}$	$1.693 \cdot 10^{-6}$	$4.271 \cdot 10^{-6}$
	e_3	$1.079 \cdot 10^{-6}$	$6.996 \cdot 10^{-7}$	$1.667 \cdot 10^{-6}$
	Total	$8.61 \cdot 10^{-6}$	$2.87 \cdot 10^{-6}$	$6.43 \cdot 10^{-6}$

Avec e_i : l'erreur de l'énergie de la $i^{ème}$ articulation.

En classant les résultats par ordre de performance décroissant, nous aurons le régulateur à 7 classes suivi de celui à 9 classes et enfin le régulateur à 5 classes d'appartenance. Donc nous appliquerons les tests de robustesse sur le régulateur flou à 7 classes d'appartenance.

- **Test de lâché de charge :**

La figure (2.35) représente le résultat des simulations des trois articulations avec introduction d'une charge de 10kg qui sera relâcher après un temps de 1s, on obtient ainsi une bonne poursuite malgré le pic des commandes, qui reviennent rapidement a leur état de stabilité.

- **Test de capacité de poursuite :**

La figure (2.38) représente le résultat des simulations des trois articulations en appliquant la trajectoire de VIVIANI (voir chapitre 1), ce test excite toute la dynamique du système, avec un lissage cycloïdal pour la deuxième articulation, malgré la discontinuité de cette trajectoire le système reste stable après de forts pics apparus aux commandes, cela montre la capacité de poursuite du système.

- **Test avec introduction de perturbations externe :**

La figure (2.36) représente le résultat des simulations des trois articulations avec introduction de perturbation sous forme de forces électromotrices au niveau de chaque moteur, c'est à dire augmenté l'amplitude de chaque commande de 1V à $t=1s$, on constate que les commandes retournent rapidement à leurs états de stabilité, ce qui veut dire que le système rejette parfaitement les perturbations.

- **Test avec variation paramétrique :**

La figure (2.37) représente le résultat des simulations des trois articulations, avec variation des paramètres mécaniques du robot de 100% après l'instant $t=1s$, On remarque la nette capacité de commander malgré cette variation.

2.13 Etude comparative :

En utilisant le critère de l'énergie de l'erreur, nous avons trouvé que pour le robot PUMA 560 le régulateur à 5 classes donnait les meilleurs résultats suivi par celui à 7 classes puis celui à 9 classes. Pour le robot à structure cylindrique c'est le régulateur à 7 classes qui est meilleur puis celui à 9 classes puis celui à 5 classes.

Ainsi nous ne pouvons pas garantir l'amélioration du régulateur en augmentant ou en diminuant le nombre de classes d'appartenance.

2.14 Conclusion :

Nous avons présenté dans ce chapitre des généralités sur la logique floue et les différentes approches de commande. L'application de cette commande sur le robot PUMA 560 et le robot à structure cylindrique a permis de donner de très bons résultats, notamment la robustesse vis à vis des perturbations et des variations paramétriques.

Pour étudier le meilleur régulateur flou appliqué au robot PUMA 560, nous avons varié les gains de normalisations et observé leur influence sur le système.

Nous proposons dans le prochain chapitre de combiner la commande floue à la commande par mode de glissement pour essayer d'améliorer encore plus les réponses du système et d'éliminer les erreurs statiques.

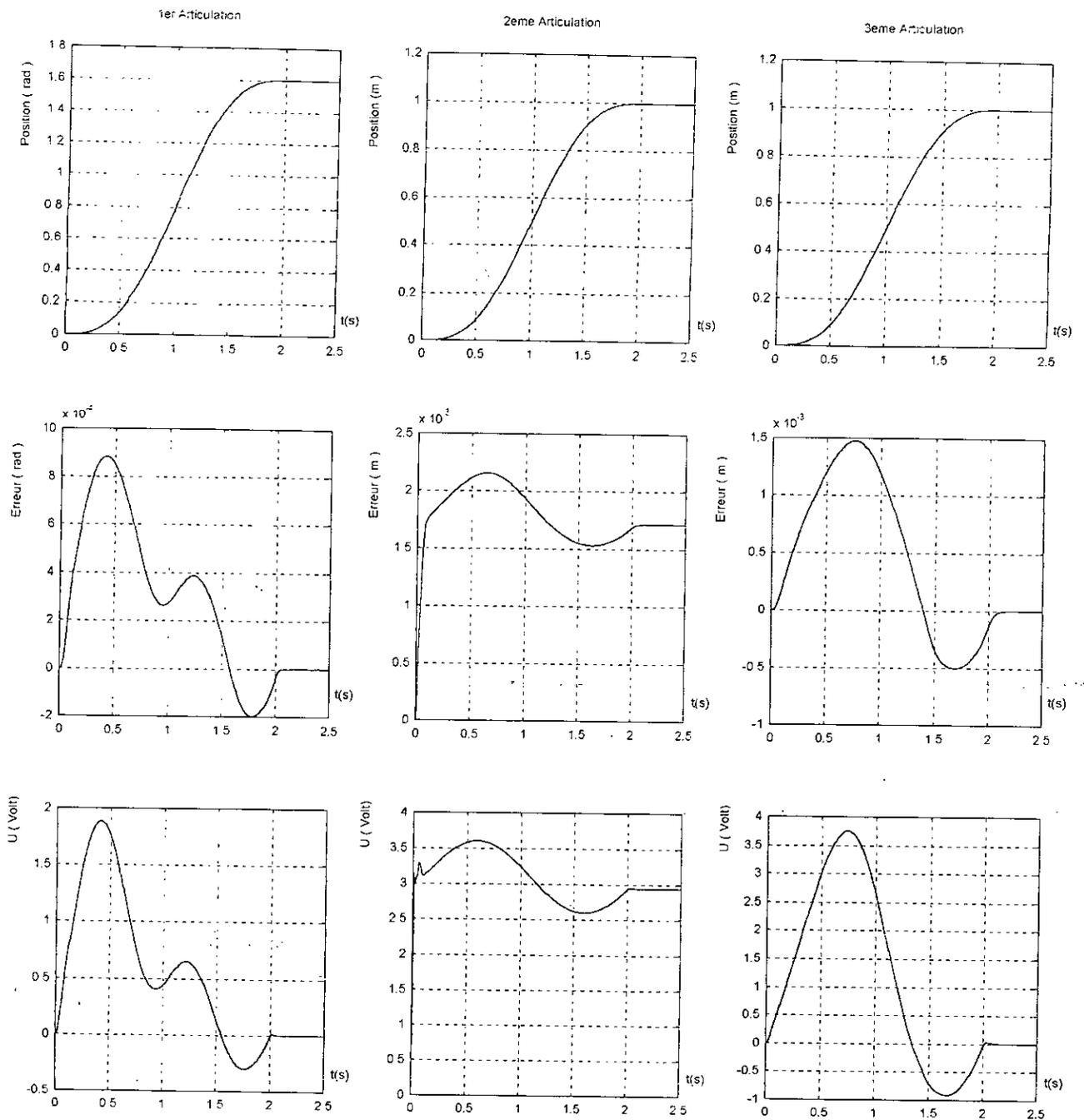


Figure (2.29) : Application de la commande floue (5 classes) au robot à structure cylindrique. Poursuite à vide

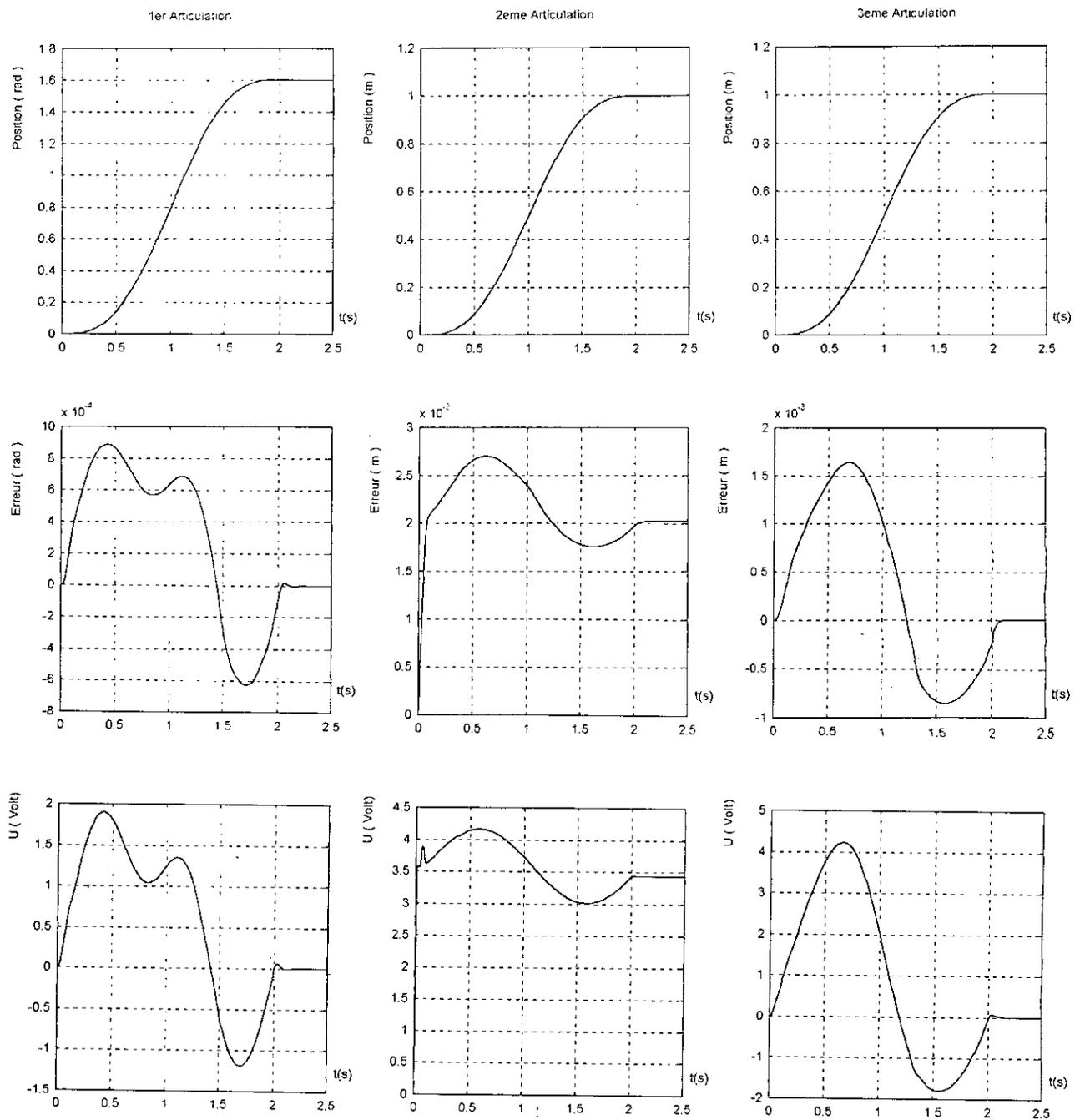


Figure (2.30) : Application de la commande floue (5 classes) au robot à structure cylindrique. Poursuite avec effet de la charge $m_0 = 5\text{kg}$

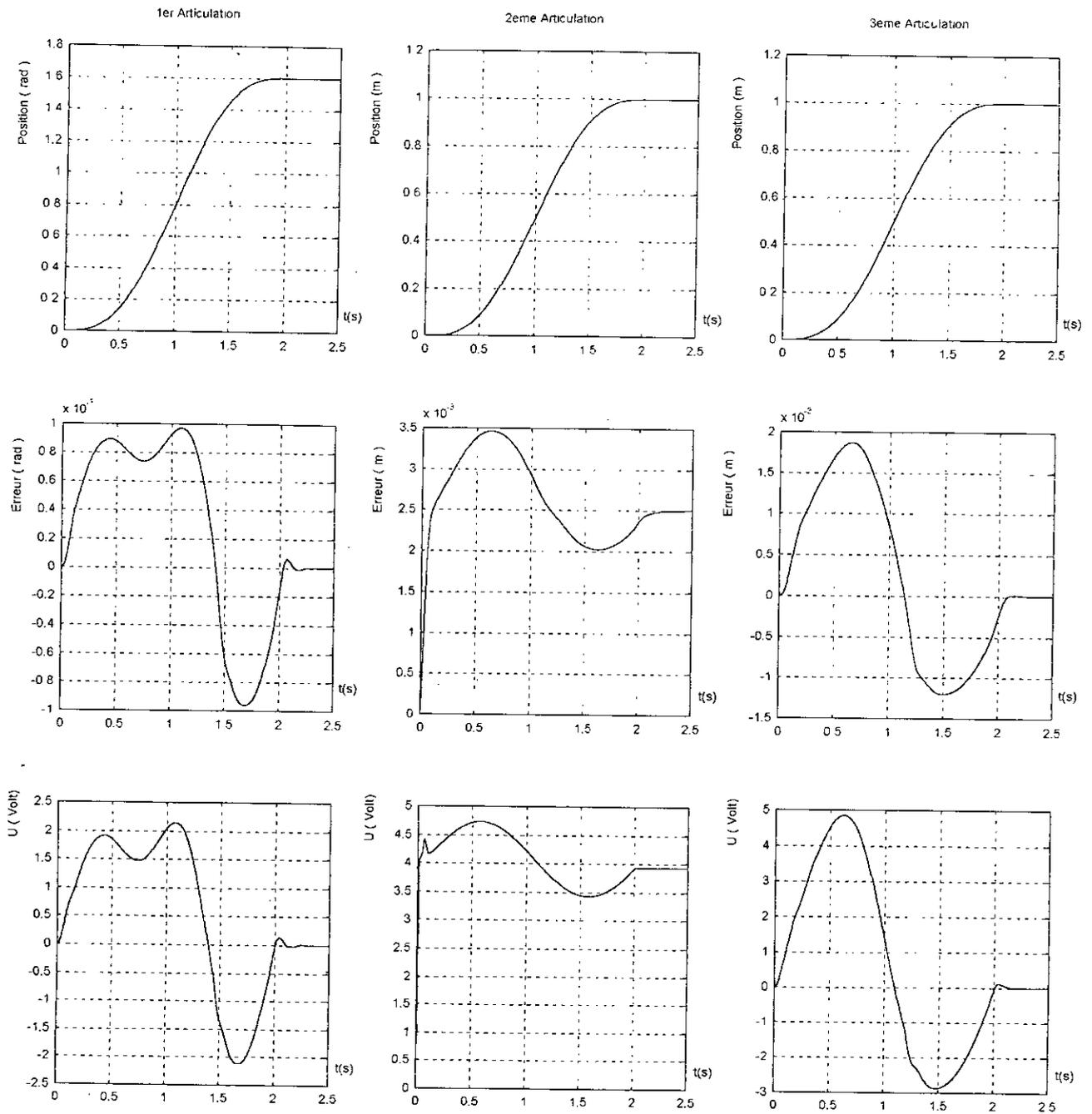


Figure (2.31) : Application de la commande floue (5 classes) au robot à structure cylindrique. Poursuite avec effet de la charge $m_0 = 10\text{kg}$

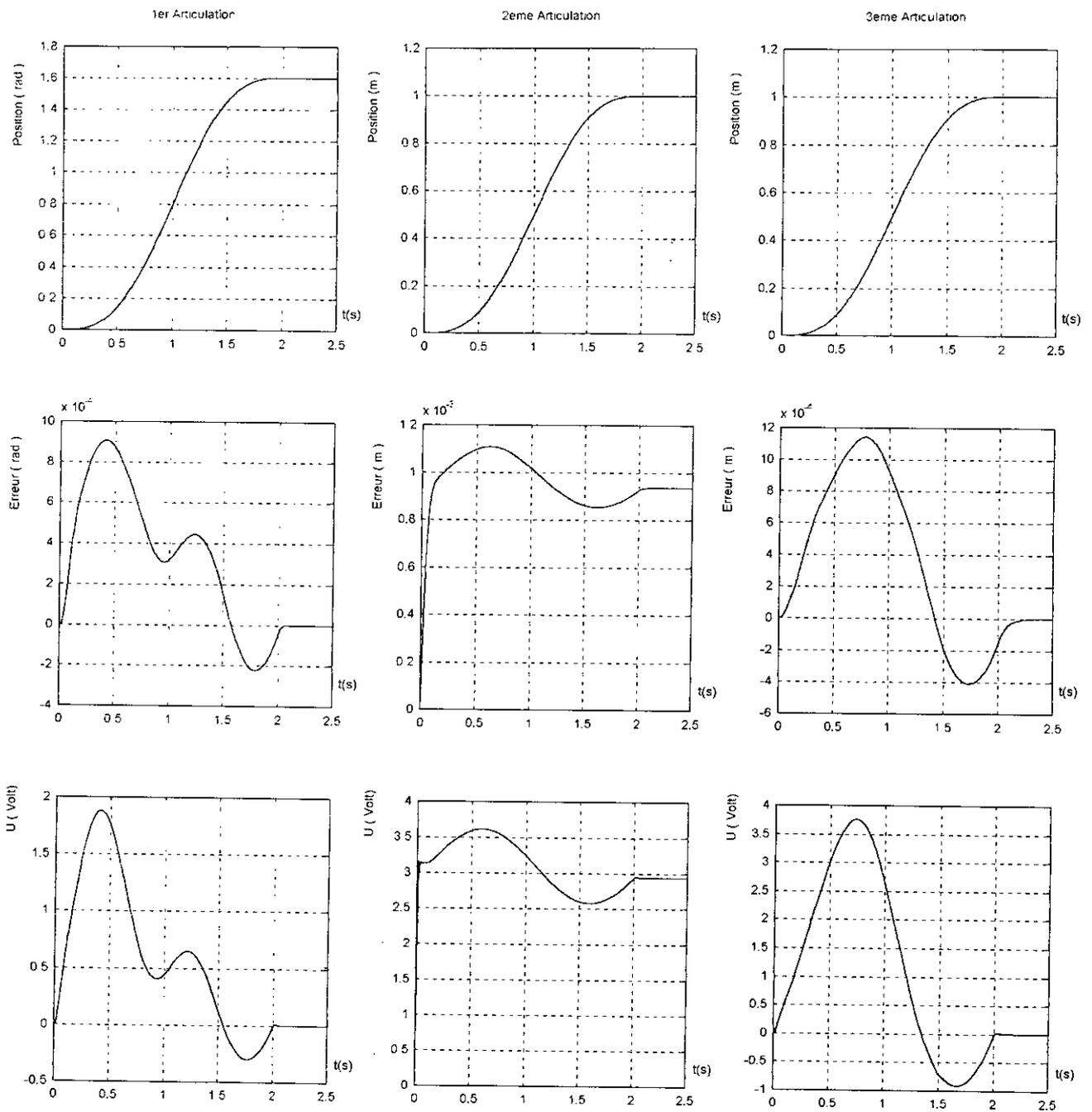


Figure (2.32) : Application de la commande floue (7 classes) au robot à structure cylindrique. Poursuite à vide

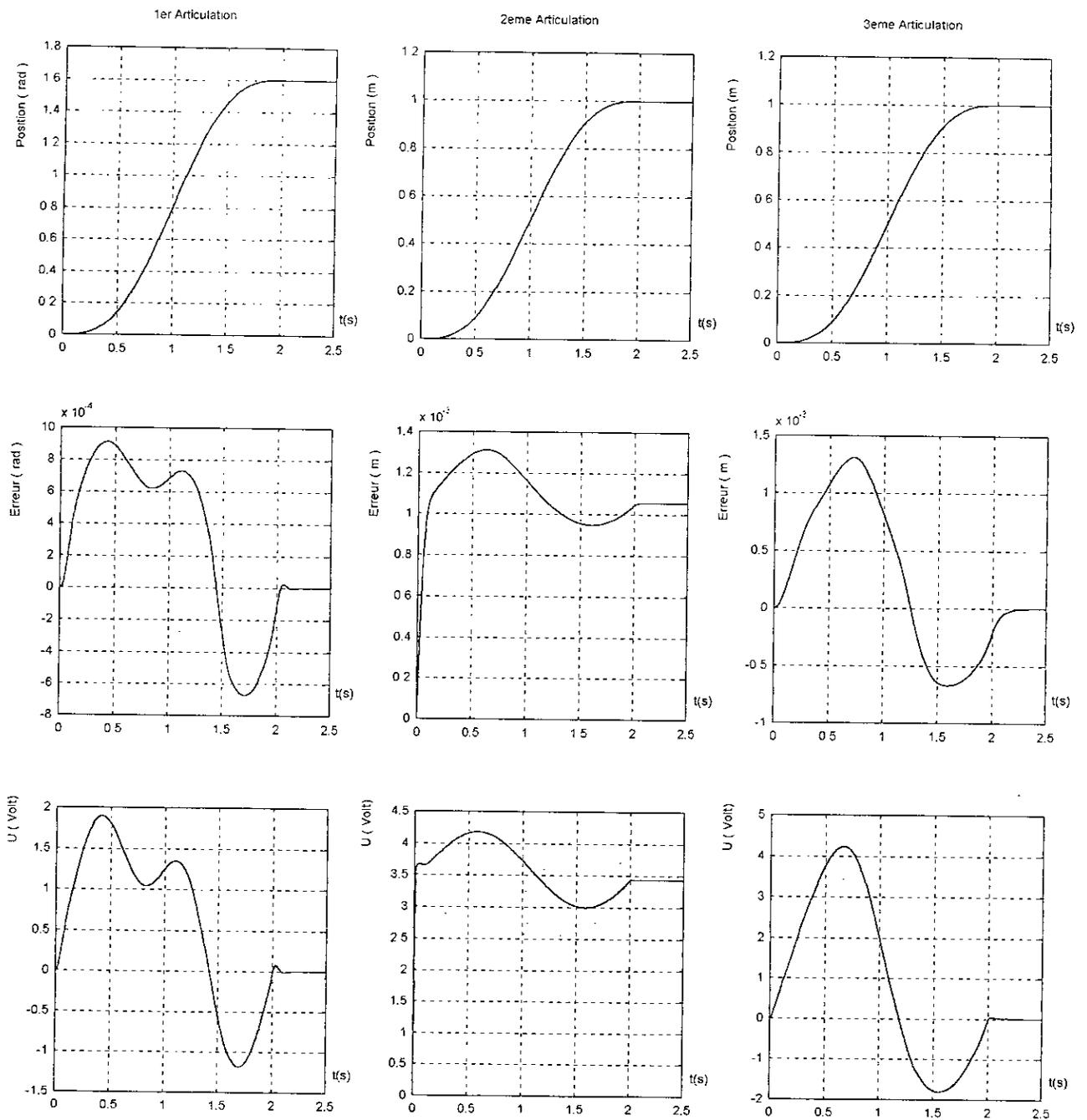


Figure (2.33) : Application de la commande floue (7 classes) au robot à structure cylindrique. Poursuite avec effet de la charge $m_0 = 5\text{ kg}$

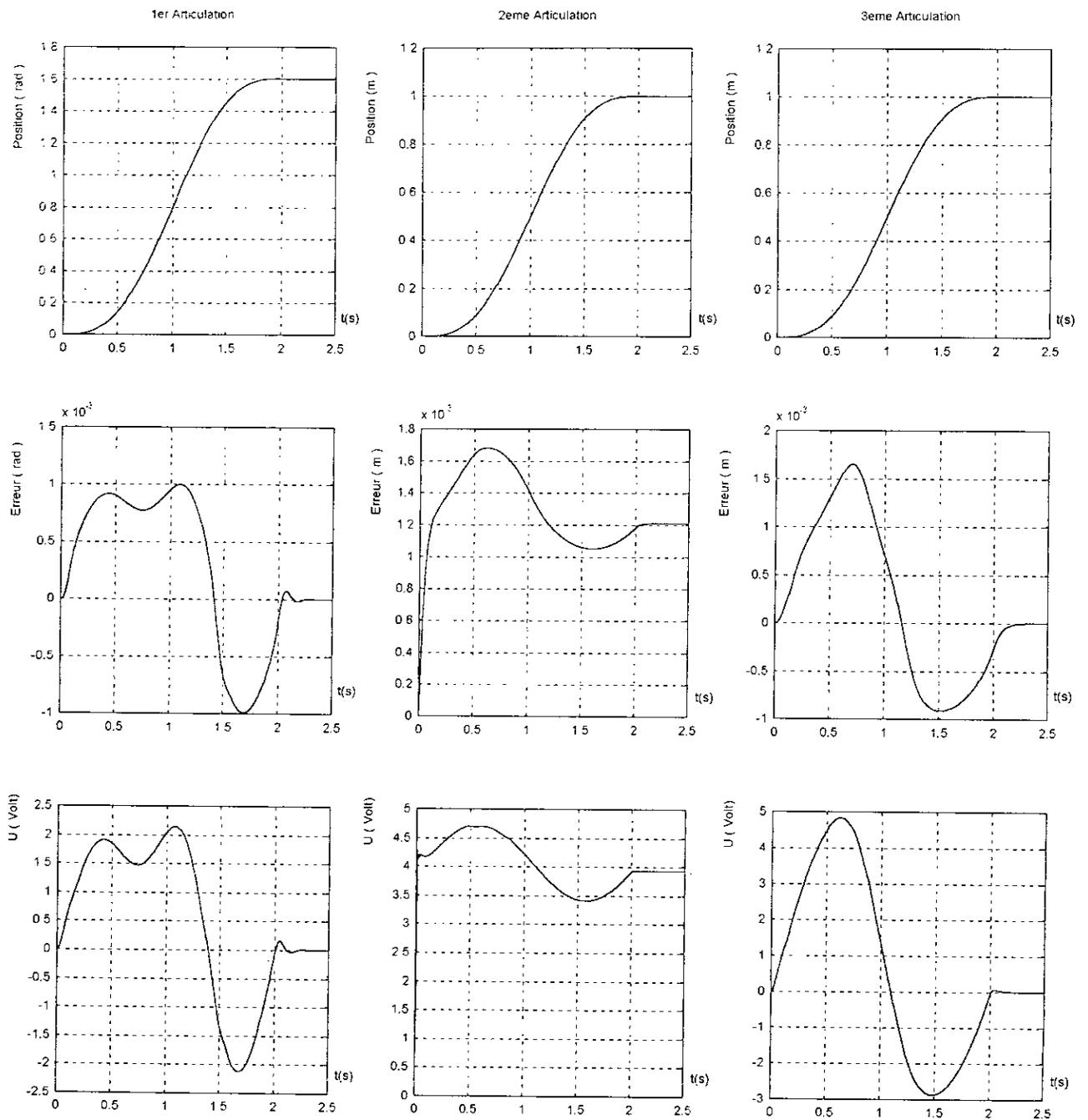


Figure (2.34) : Application de la commande floue (7 classes) au robot à structure cylindrique. Poursuite avec effet de la charge $m_0 = 10\text{kg}$

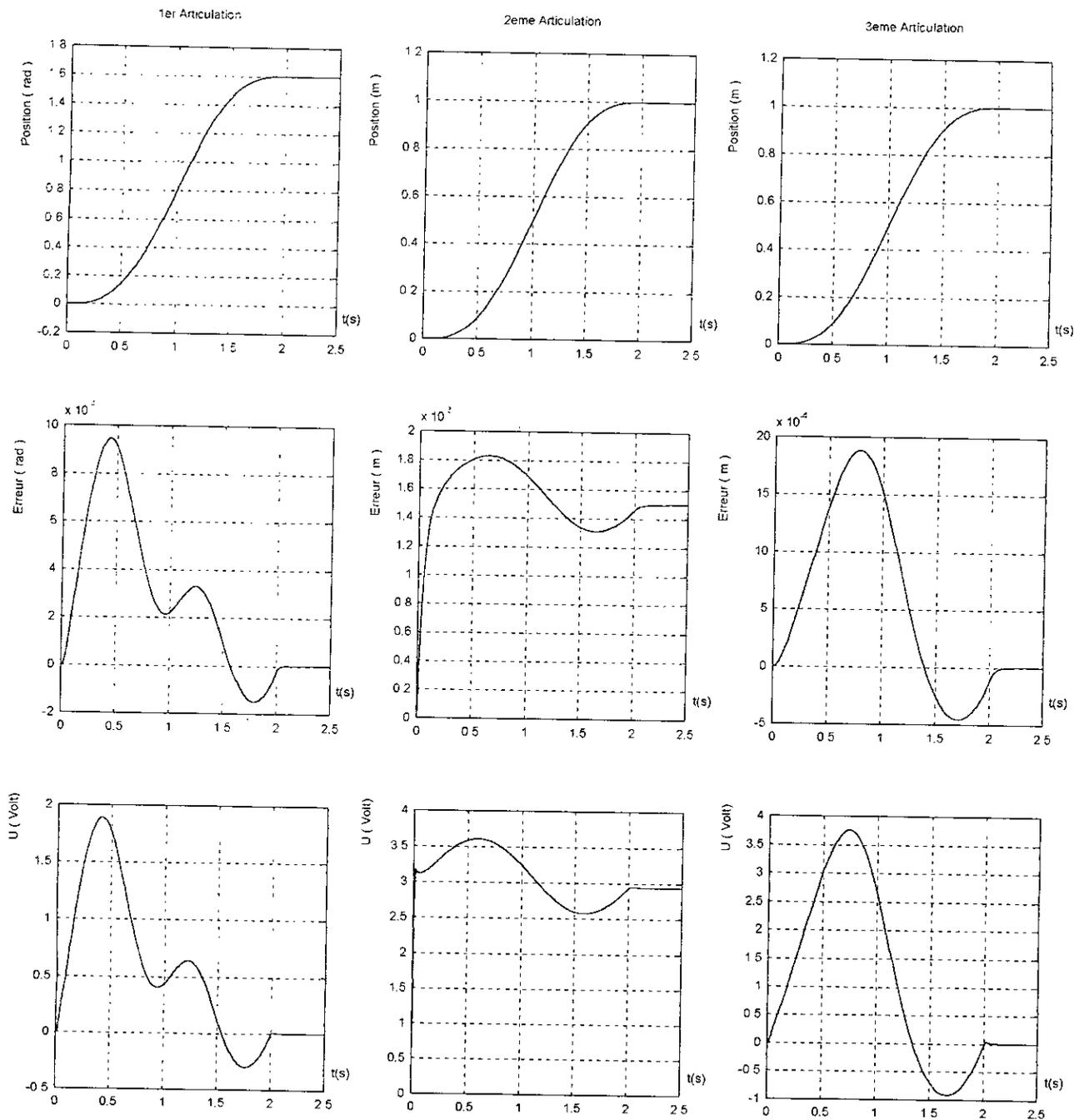


Figure (2.35) : Application de la commande floue (9 classes) au robot à structure cylindrique. Poursuite à vide

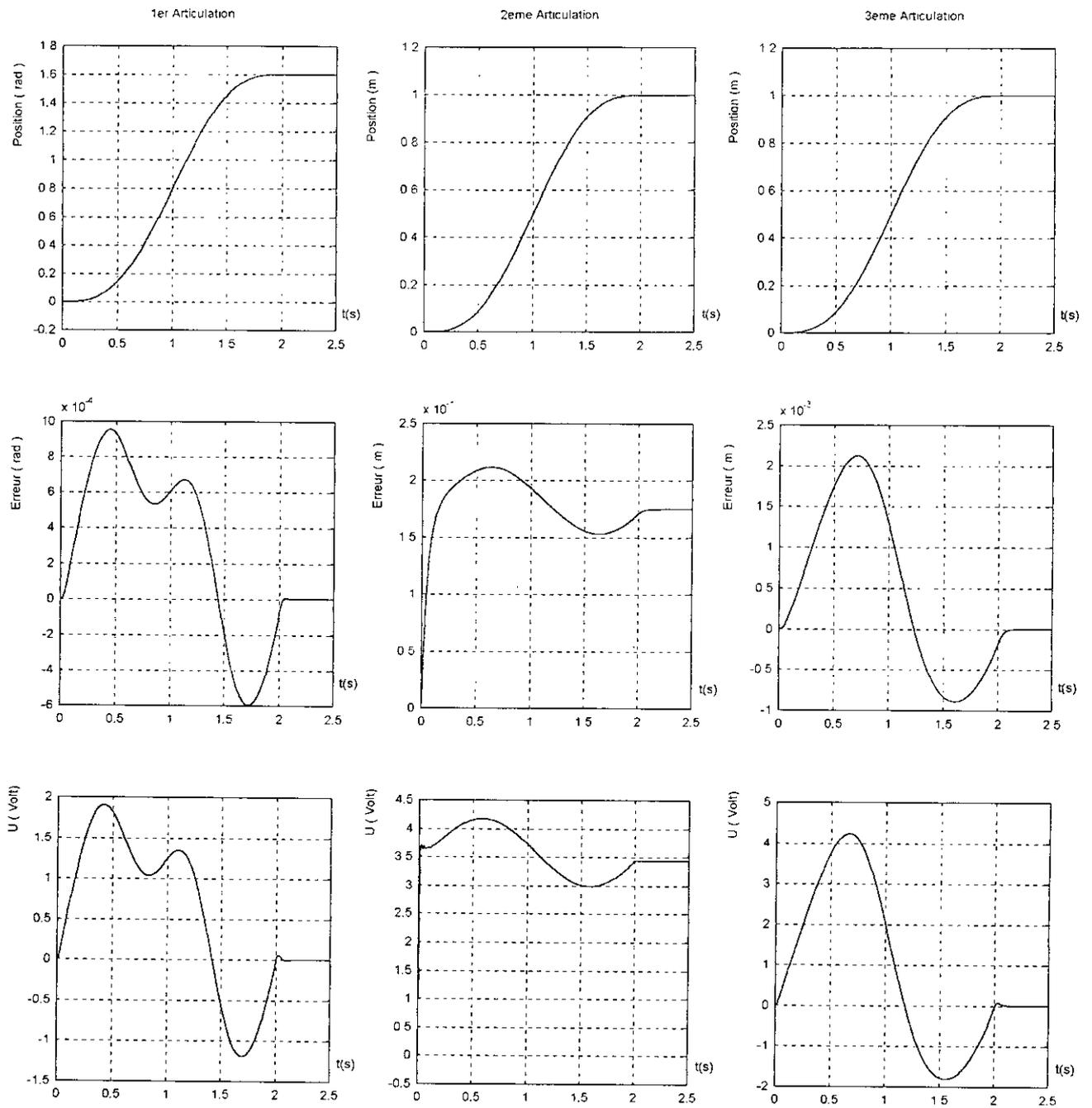


Figure (2.36) : Application de la commande floue (9 classes) au robot à structure cylindrique. Poursuite avec effet de la charge $m_0 = 5\text{kg}$

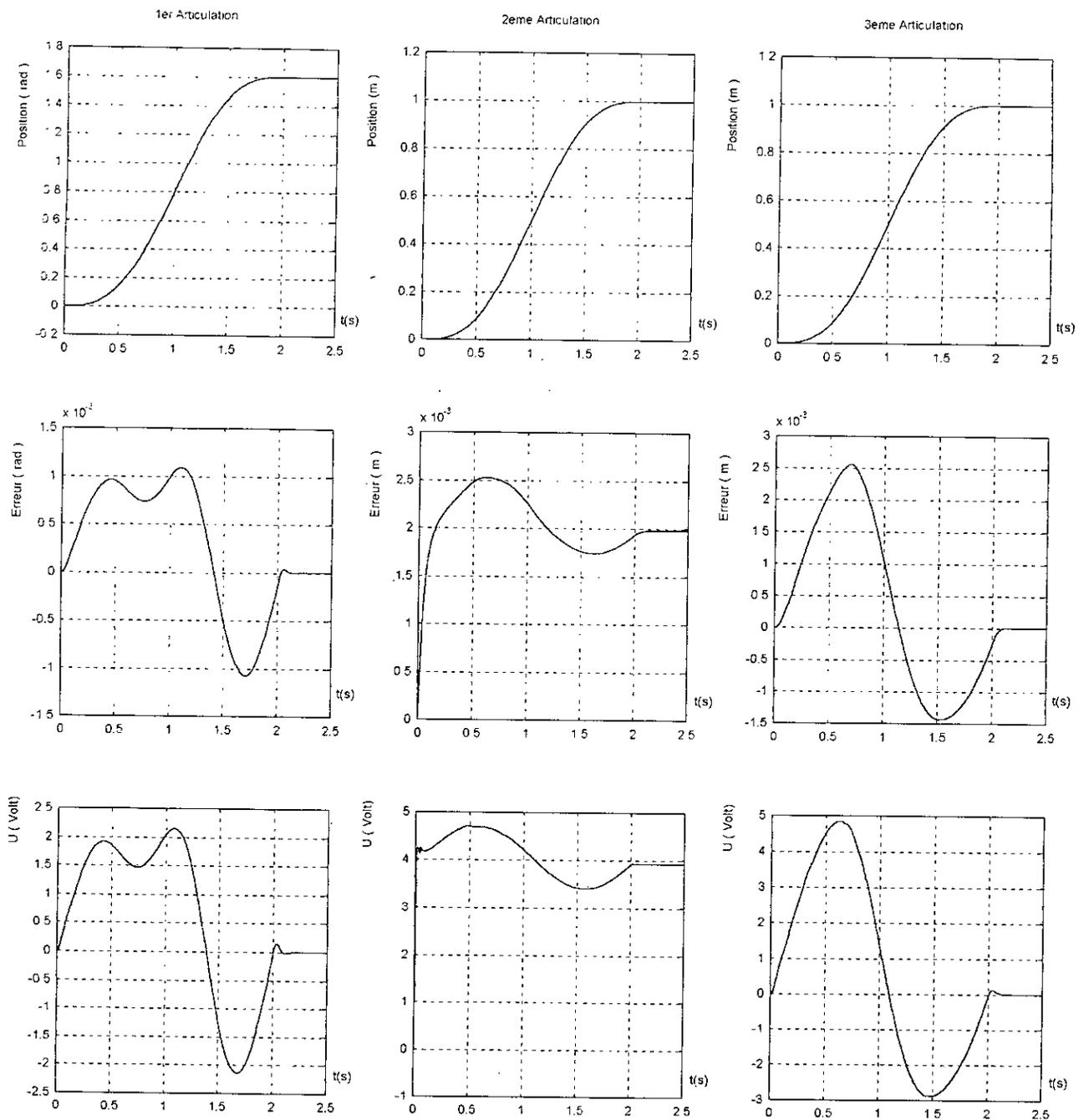


Figure (2.37) : Application de la commande floue (9 classes) au robot à structure cylindrique. Poursuite avec effet de la charge $m_0 = 10\text{kg}$

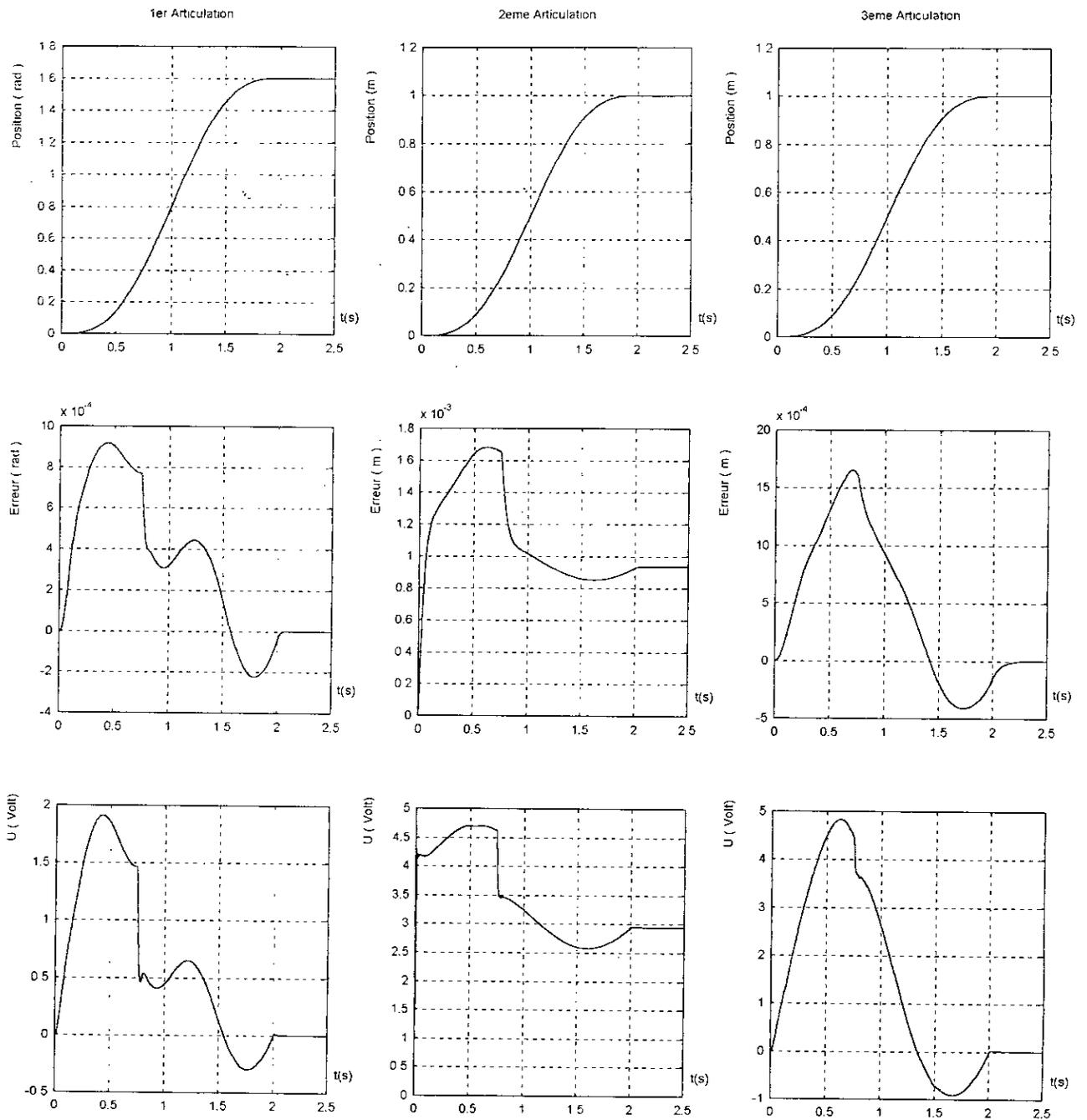


Figure (2.39) : Application de la commande floue (7 classes) au robot à structure cylindrique
 Poursuite avec effet de la charge $m_0 = 10\text{kg}$ et la décharge à l'instant $t=1\text{s}$.

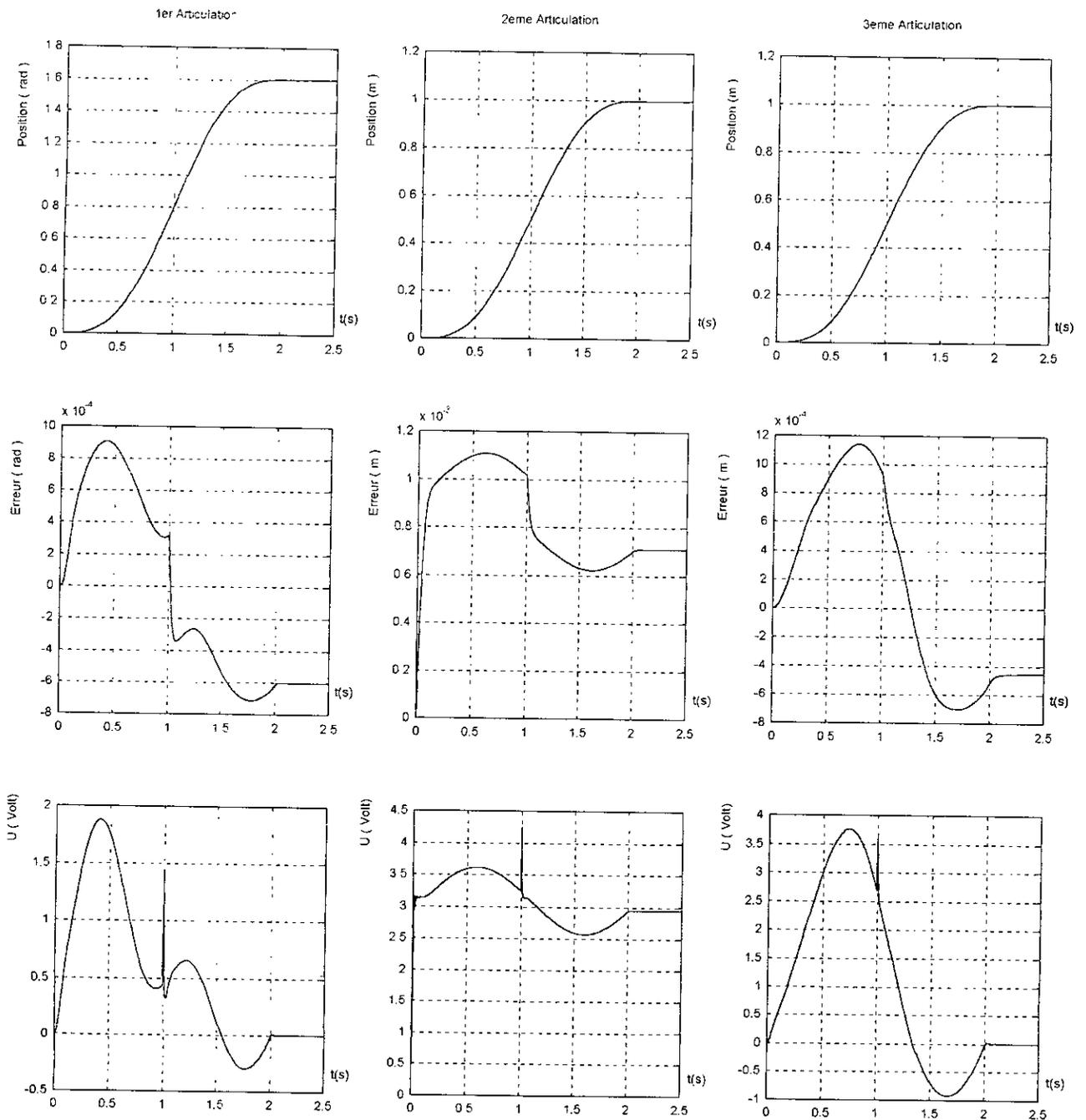


Figure (2.39) : Application de la commande floue (7 classes) au robot à structure cylindrique
 Poursuite avec des perturbations externes $u_{pi} = -1$ Volt à partir de l'instant $t=1$ s.

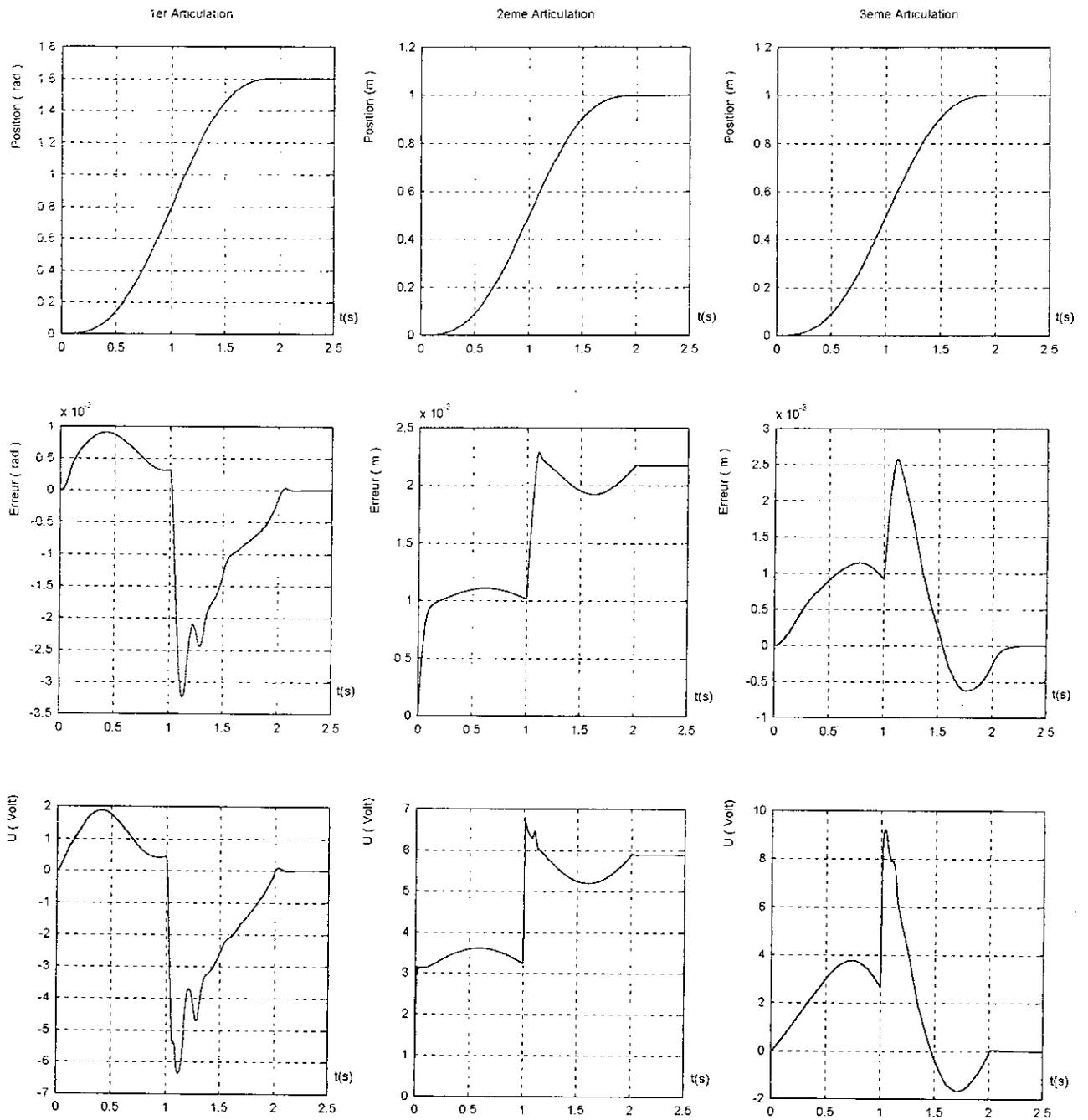


Figure (2.40) : Application de la commande floue (7 classes) au robot à structure cylindrique
Poursuite avec une variation paramétrique de 100% à partir de l'instant $t = 1$ s.

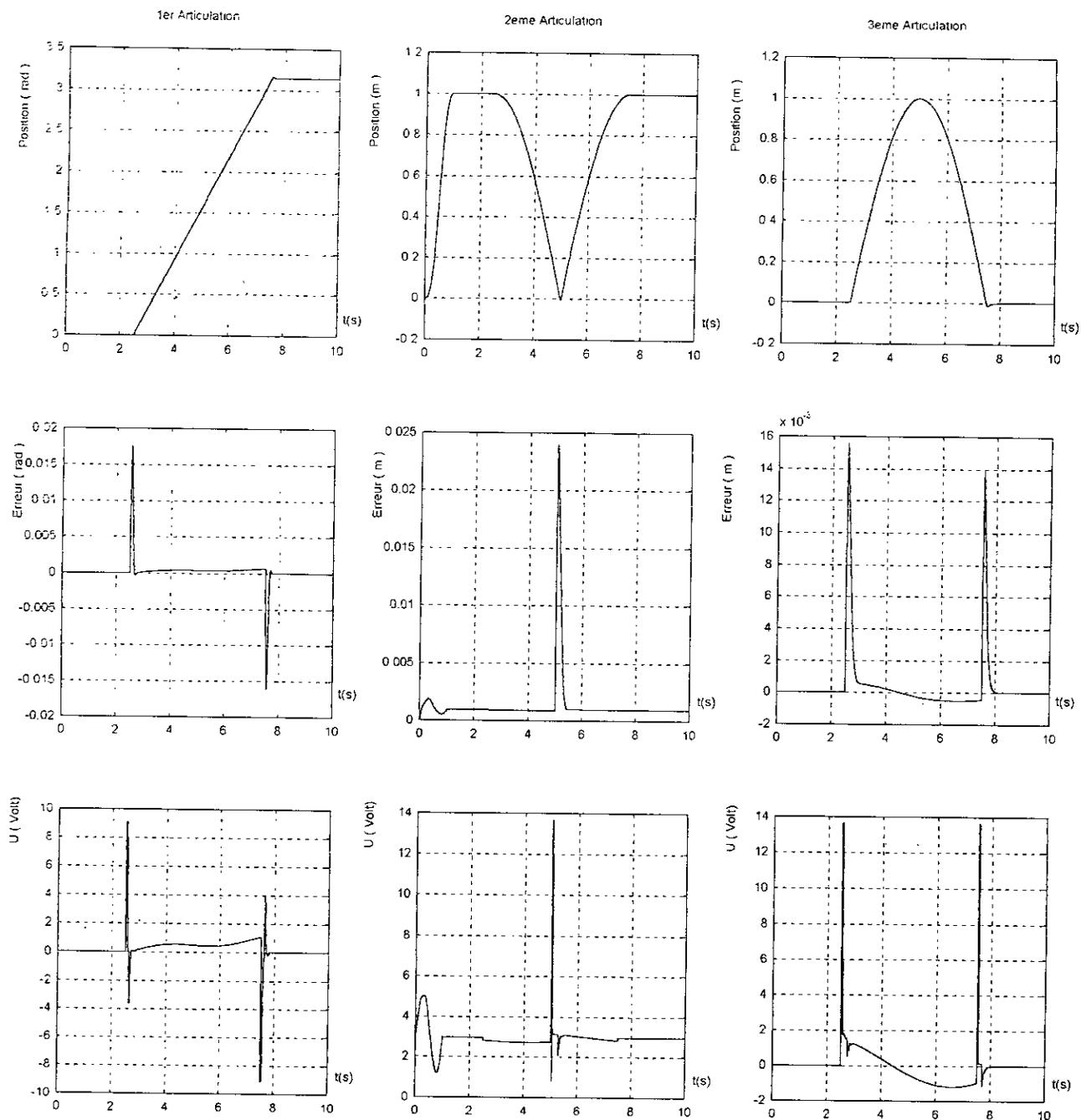
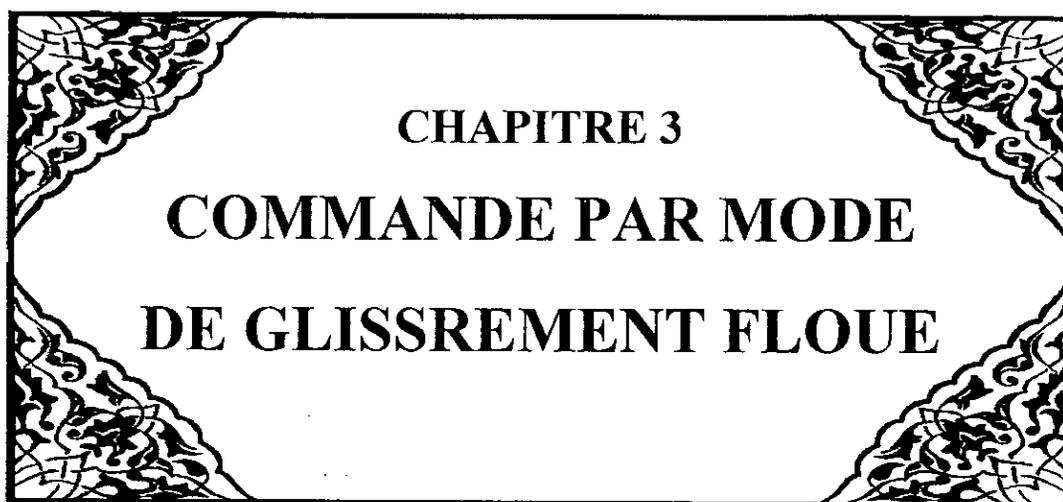


Figure (2.41) : Application de la commande floue (7 classes) au robot à structure cylindrique
 Test de capacité de poursuite (fenêtre de VIVIANI)



CHAPITRE 3
COMMANDE PAR MODE
DE GLISSREMENT FLOUE

3.1 Introduction :

Les régulateurs classiques permettent de commander avec précision des systèmes linéaires sans perturbations et à paramètres constants. Mais lorsqu'on a des perturbations ou des variations de paramètres relativement faibles, on ne peut pas utiliser ces régulateurs, surtout si les exigences sur la précision et sur les caractéristiques dynamiques du système de commande ne sont pas strictes. Dans le cas contraire on peut utiliser une solution simple, appelée "**Systèmes à Structure Variable**".

Le cas particulier de cette théorie n'est rien d'autre que la technique des modes glissant qui amène la trajectoire d'état du système à régler vers une surface de glissement (surface de commutation). La théorie des systèmes à structure variable (SVV) et des modes glissants associés (MG) est une technique de commande non linéaire, caractérisée par des commandes discontinues. La trajectoire d'état du système en mode de glissement appartient à une surface de commutation inférieure à celle de l'espace d'état, le choix des coefficients de la surface de glissement donne la dynamique du système en mode de glissement.

La théorie des systèmes à structure variable a connu un développement durant les 40 années précédentes, en ex-Union Soviétique par V.Utkin [40][41], aux Etats-Unis par J.J.Slotine [36] [37] [38], et ceci à partir des travaux théoriques du mathématicien Russe A.F.Filipov.

L'intérêt récent à cette technique de commande est dû essentiellement à la disponibilité d'interrupteurs rapides et de microprocesseurs plus en plus performants permettant la commande en temps réel des systèmes dynamiques[5].

3.2 Système de réglage à structure variable :

La configuration de base des systèmes à structure variable peut être représentée par un schéma général, ce dernier est illustré par la figure 3.1.

On peut remarquer d'après la figure 3.1 que la commande globale se compose de 3 blocs élémentaires à savoir :

- Le premier bloc (1) constitue deux vecteurs de commandes auxiliaires, un des deux formes la première partie u_1 de la commande globale suivant le signe de la loi de commutation $\sigma(x,t)$.
- Le deuxième bloc (2) fournit la commande u_2 . Cette commande est formée en multipliant le vecteur désiré par une matrice d'anticipations $\psi^+(x_d)$ ou $\psi^-(x_d)$ par le biais du signe de la loi de commutation.
- Le troisième bloc (3) fournit la commande u_3 commutant entre deux contres réactions d'état, cette commande est donnée par $\phi^+(x) x$ ou $\phi^-(x) x$ suivant toujours le signe de loi de commutation.

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

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

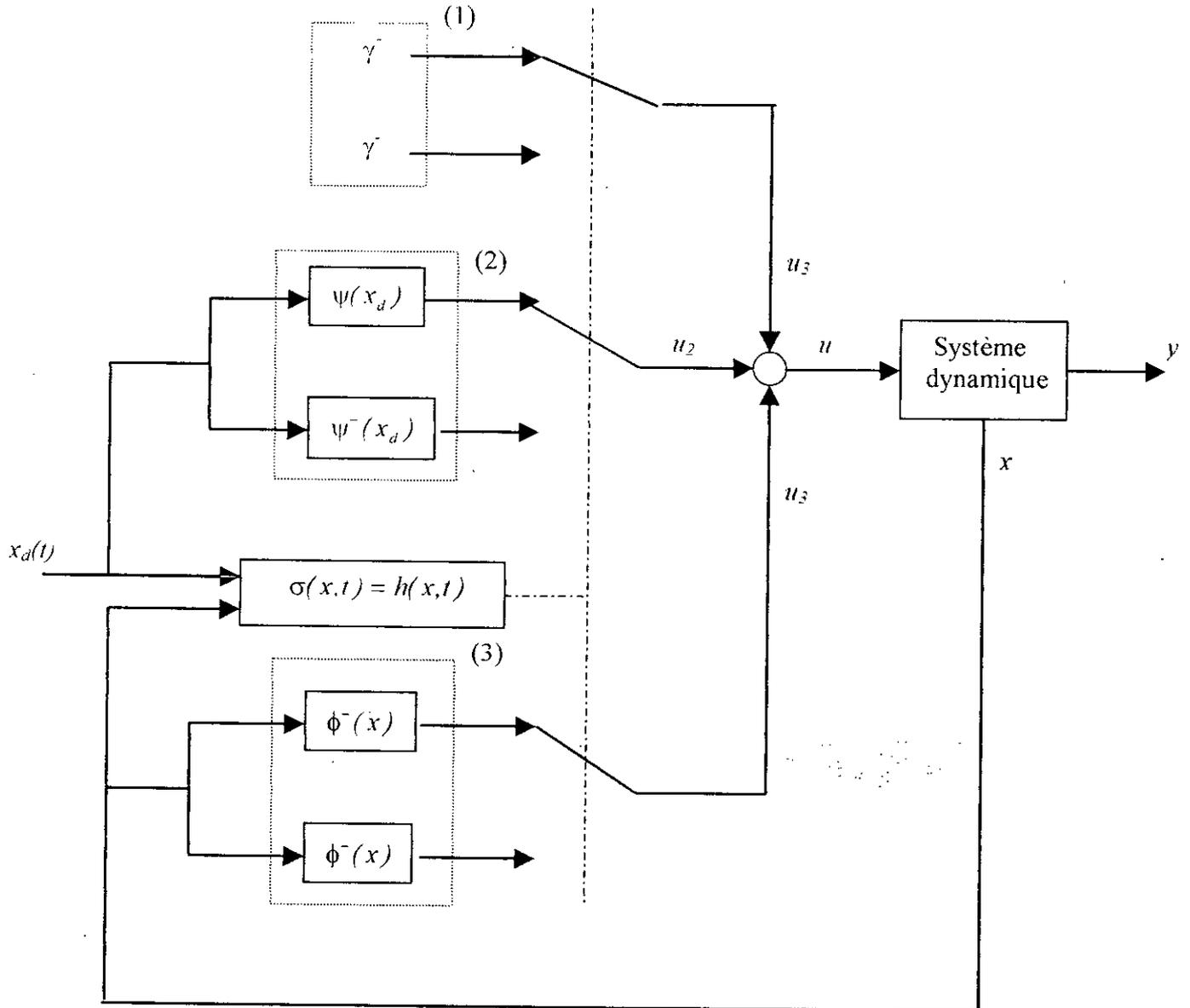


Figure 3.1 : Configuration de base pour les systèmes à structure variable.

3.3 Les modes de la trajectoire dans le plan de phase :

La technique de la commande par mode glissant consiste à ramener la trajectoire d'état d'un système vers la surface de glissement et de faire commuter à l'aide d'une logique de commutation approprié jusqu'au point d'équilibre.

Cette trajectoire est constituée de trois parties distinctes (fig 3.2) :

- ◆ **Le mode de convergence (MC)** : dont la variable à réguler se déplace à partir du point d'équilibre initial et tend vers la surface de glissement $S(x)=0$.
- ◆ **Le mode de glissement (MG)** : durant lequel la variable d'état a atteint la surface de glissement et tend vers l'origine du plan de phase.
- ◆ **Le mode de régime permanent (MRP)** : il est nécessaire pour l'étude du comportement du système autour du point d'équilibre (l'origine du plan de phase).

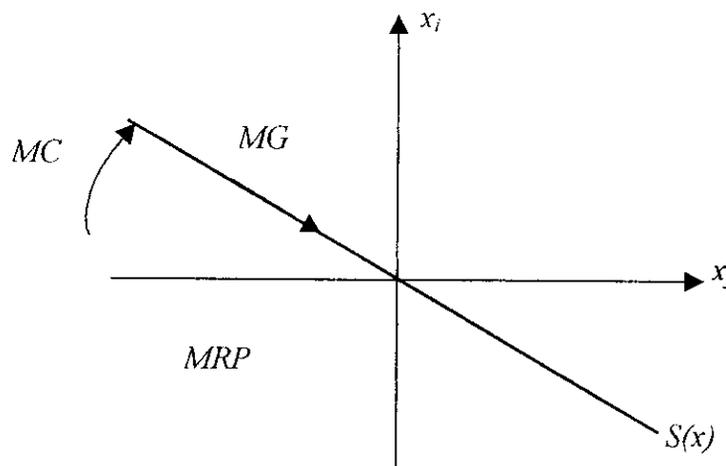


Figure 3.2 : Les modes de trajectoire dans le plan de phase

3.4 Conception de la commande par mode de glissement :

La commande par mode de glissement possède plusieurs avantages comme par exemple :

- Robustesse vis-à-vis des variations paramétriques.
- Simplicité de la mise en œuvre et d'adaptation, aussi bien aux processus linéaires qu'aux processus non linéaires.

Quant au phénomène de "Chattering" qui lui est conceptuellement associé et qui, à l'origine, a limité son utilisation, il est actuellement bien maîtrisé et peut être éliminé par des procédés simples comme la linéarisation dans un voisinage des surfaces de commutation. Ce type de commande peut être également utilisé d'une manière tout à fait analogue, non seulement en régulation, mais dans les problèmes de poursuite de trajectoires ou de poursuite de modèle. La conception des contrôleurs par mode de glissement prend en compte les problèmes de stabilité et des bonnes performances.

La conception de cette commande peut être divisée en trois étapes principales

1. Le choix des surfaces.
2. L'établissement des conditions d'existence et de convergence.
3. La détermination de la loi de commande.

3.5 Choix de la surface de glissement :

Deux paramètres entrent en jeu pour le choix de la surface de glissement c'est le nombre nécessaire ainsi que la forme. Ces deux facteurs sont en fonction de l'application et l'objectif visé. En général, concernant le nombre de surfaces nous pouvons trouver dans la littérature pour un système défini par l'équation d'état suivante :

$$\begin{cases} \dot{X}(t) = f(X,t) + g(X,t)U(t) \\ Y = C^T X, \quad Y \in R^m \end{cases} \quad (3.1)$$

Il faut choisir m surfaces de glissement pour un vecteur V de dimension m . En ce qui concerne la forme de la surface, deux possibilités se présentent, soit dans le plan de phase ou dans l'espace d'état. Dans ce dernier cas, on trouve la méthode connue par "**loi de commutation par contre réaction d'état**" [8]. Celle-ci utilise les concepts du réglage par contre réaction d'état pour synthétiser la loi de commutation.

Dans le cas du traitement dans l'espace de phase, la fonction de commutation est une fonction scalaire, telle que la variable à régler glisse sur cette surface pour atteindre l'origine du plan de phase [36] [37] [38]. Ainsi la surface $S(x)$ représente le comportement dynamique désiré du système. J.J.Slotine nous propose une forme d'équation générale pour déterminer la surface de glissement qui assure la convergence d'une variable x à sa valeur de consigne.

$$S(x) = \left(\frac{\partial}{\partial t} + \lambda_x \right)^{r-1} e(x) \quad (3.2)$$

x : variable à réguler.

$e(x)$: l'écart de la variable à réguler $e(x) = x^* - x$.

λ_x : une constante positive qui interprétera la bande passante du contrôle désiré.

r : degré relatif (égal au nombre de fois qu'il faut dériver la sortie pour faire apparaître la commande).

$$\text{pour } r=1 \quad S(x) = e(x)$$

$$\text{pour } r=2 \quad S(x) = \lambda_x e(x) + \dot{e}(x)$$

$$\text{pour } r=3 \quad S(x) = \lambda_x^2 e(x) + 2\lambda_x \dot{e}(x) + \ddot{e}(x)$$

$S(x)=0$ est une équation différentielle linéaire dont l'unique solution est $e(x)=0$ pour un choix convenable du gain λ_x . En d'autre terme, la difficulté revient à un problème de poursuite de trajectoire, ce problème est équivalent à celui du problème qui consiste à amener les points représentatifs du système à rester sur la surface de glissement défini par l'équation $S(x)=0$,

ceci est équivalent à une linéarisation exacte de l'écart tout en respectant une condition de convergence. La linéarisation exacte de l'écart figure (3.3). Consiste à forcer la dynamique de l'écart d'être une dynamique d'un système linéaire autonome d'ordre r .

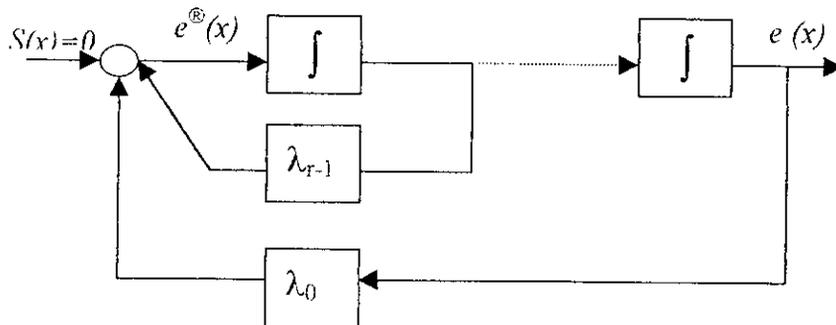


Figure 3.3 : Linéarisation exacte de l'écart

3.6 Calcul de la commande :

Une fois la surface de glissement est choisie, ainsi que le critère de convergence, il reste à déterminer la commande nécessaire pour attirer la variable à régler vers la surface, ensuite vers son point d'équilibre (origine du plan de phase) en maintenant la condition d'existence du mode glissant. Lorsqu'il y a un régime glissant, la dynamique du système est indépendante de la loi de commande, c'est pourquoi la surface de glissement a pu être déterminée indépendamment de la commande, sur la base du système et des performances désirées.

De ce fait, on peut introduire une partie continue pour diminuer l'amplitude de la discontinuité, on aura donc :

$$U = U_{eq} + U_s \quad (3.3)$$

U_{eq} correspond à la commande équivalente proposée par Filipov et Utkin. On peut la considérer comme la valeur moyenne continue que prend la commande lors d'une commutation rapide entre deux valeurs U_{max} et U_{min} (figure 3.4).

U_s correspond à la commande qui garanti l'attractivité de la variable à contrôler vers la surface et satisfaire la condition :

$$S(x)\dot{S}(x) < 0 \quad (3.4)$$

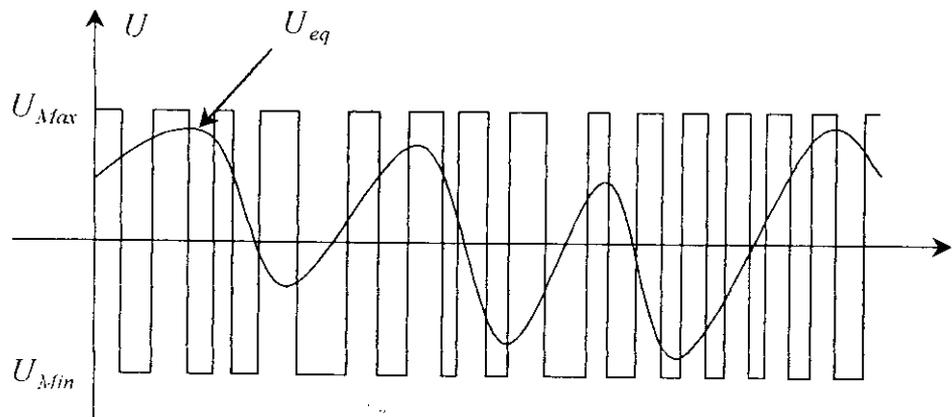


Figure 3.4 : la valeur continue U_{eq} prise par la commande lors de la commutation entre U_{Max} et U_{Min} .

3.7 Expression analytique de la commande :

Nous nous intéressons au calcul de la commande équivalente du système défini dans l'espace d'état par l'équation (3.3).

$$\dot{x} = f(x) + g(x)U \quad (3.5)$$

avec: $x \in \mathcal{R}^n$,
 $u \in \mathcal{R}^1$.

En régime de glissement idéal, l'expression des surfaces et de leurs dérivées sont nulles. Ceci se traduit par:

$$U_s = 0 \quad (3.6)$$

Donc

$$\dot{S}(x) = \left(\frac{\partial S}{\partial x} \right)^T (f(x) + g(x)U) = 0 \quad \text{avec } U = U_{eq} \quad (3.7)$$

Ainsi, la commande équivalente est donnée :

$$U_{eq} = - \left[\left(\frac{\partial S}{\partial x} \right)^T g(x) \right]^{-1} \left\{ \left(\frac{\partial S}{\partial x} \right)^T f(x) \right\} \quad (3.8)$$

Avec la condition de transversalité : $\det \left[\left(\frac{\partial S}{\partial x} \right)^T g(x) \right] \neq 0$

Mais, le régime idéal n'est pratiquement jamais réalisable. On doit faire usage du deuxième terme de la commande pour ramener l'état du système vers la surface à chaque fois qu'il s'en écarte. Ainsi, il convient de prendre :

$$U_x \equiv \dot{S}(x) = -K \text{sign}(S(x)) \quad (3.9)$$

ou $K = \text{diag}(k_1, \dots, k_n)$

qui est la forme d'un relais :

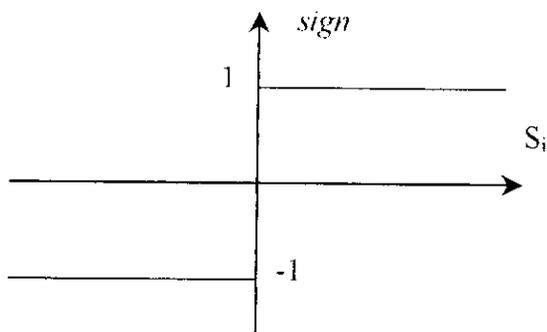
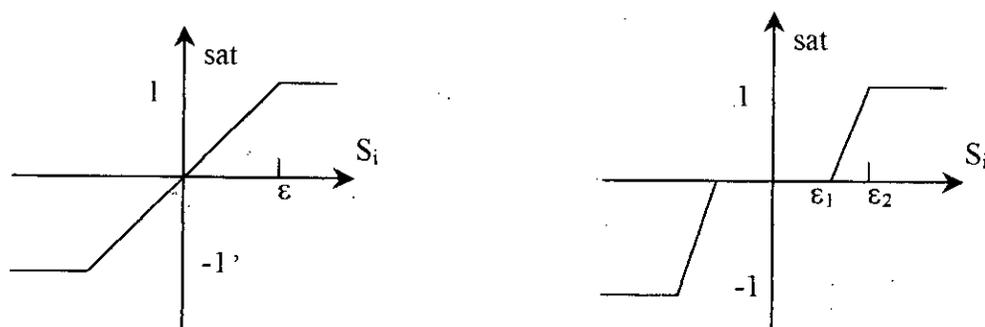


Figure 3.5 : Représentation de la fonction sign

3.8 Elimination du phénomène du "Chattering" :

L'un des principaux inconvénients du réglage par mode de glissement est le phénomène du "chattering" (phénomène de broutement), car il peut endommager les actionneurs par des sollicitations trop fréquentes et nuire au fonctionnement et aux performances du système.

Dans le but de réduire ces oscillations plusieurs solutions ont été apportées, comme par exemple remplacer la fonction sign par une fonction de saturation caractérisée par un ou deux seuils (atténuation des amplitudes des ondulations) :



a) Fonction SAT avec un seuil

b) Fonction SAT avec deux seuils (zone morte)

Figure 3.6 : Fonction « SAT »

Ces deux fonctions son définies par :

a) :

$$sat(S) = \begin{cases} -1 & \text{si } S < -\varepsilon \\ S & \text{si } |S| < \varepsilon \\ \varepsilon & \text{si } S > \varepsilon \\ 1 & \text{si } S > \varepsilon \end{cases}$$

b):

$$sat(S) = \begin{cases} 0 & \text{si } |S| < \varepsilon_1 \\ \frac{S - \varepsilon_1}{\varepsilon_2 - \varepsilon_1} & \text{si } \varepsilon_1 < |S| < \varepsilon_2 \\ sign(S) & \text{si } |S| > \varepsilon_2 \end{cases}$$

On peut aussi donner la fonction de Smooth :

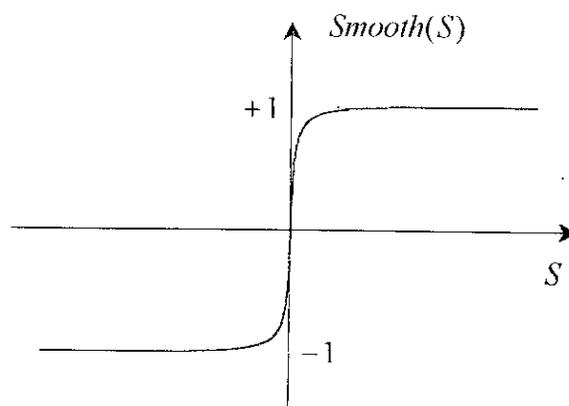


Figure 3.7 : Fonction « Smooth ».

$$Smooth(S) = \frac{S}{|S| + \mu}$$

3.9 Application de la commande par mode de glissement flou [30] :

Les bases de règles de la commande flou ont été construit en utilisant la condition :

$$e \dot{e} < 0 \quad (3.10)$$

Sachant que la condition d'existence des modes glissants est :

$$S \dot{S} < 0 \quad (3.11)$$

On peut dire qu'il y a une similitude entre les deux lois, on peut donc remplacer l'erreur e par la surface de glissement S pour avoir le régulateur flou glissant.

Le schéma de la figure (3.8) représente le régulateur flou glissant d'un robot à trois articulations. Les entrées de ce contrôleur sont la surface de glissement S et la variation de cette surface ΔS .

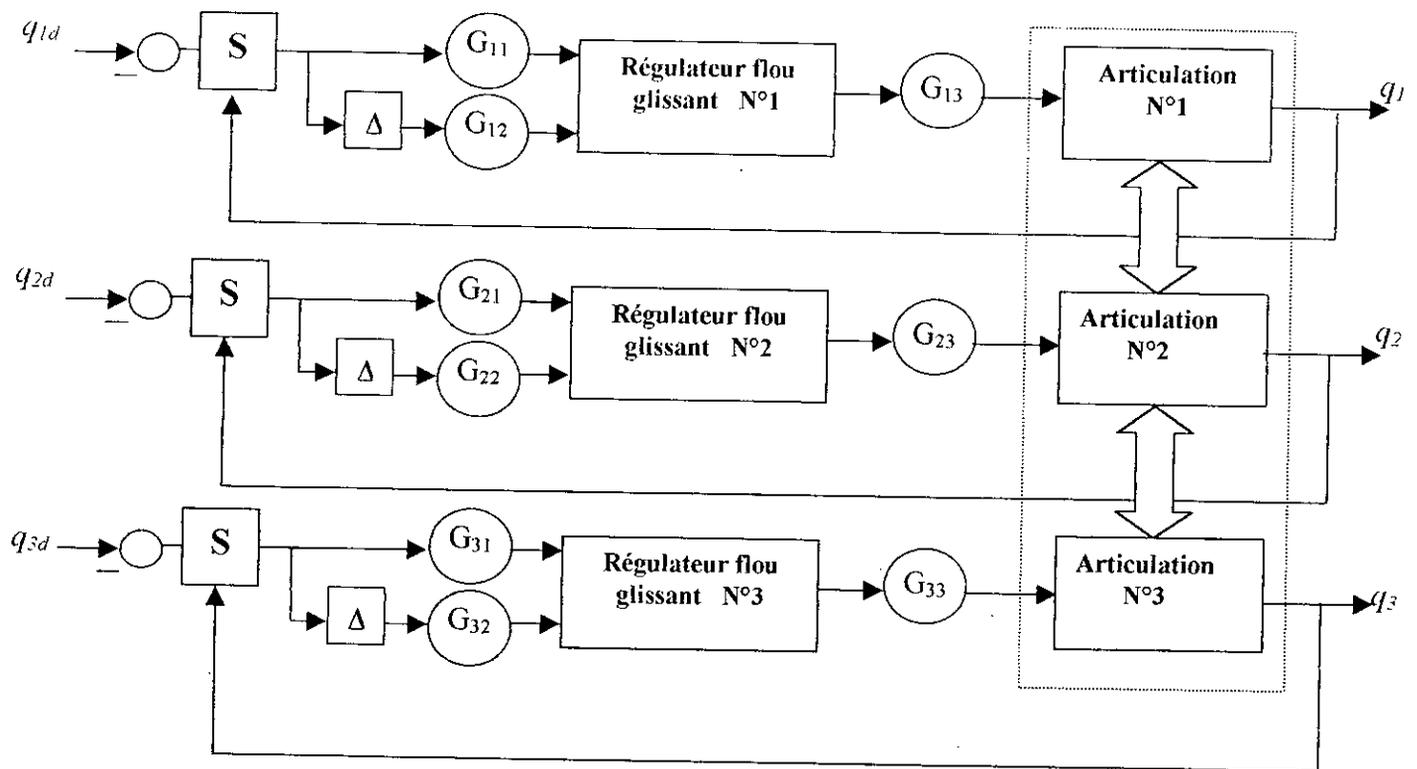


Figure 3.8 : régulateur flou glissant d'un robot à trois articulations

En partant de la forme générale de la surface de Slotine (3.2), nous avons choisi la surface de glissement pour le PUMA 560 :

$$S = e + 0.005\dot{e} + 20\int e dt$$

Et pour le robot à structure cylindrique :

$$S = e + 0.005\dot{e} + 10\int e dt$$

3.9.1 Application de la commande au robot PUMA 560 :

Pour l'application de la commande par mode de glissement flou, nous reprendrons les mêmes régulateurs utilisés pour la commande floue, on l'occurrence 5, 7 et 9 classes d'appartenance.

Les figures (3.9), (3.12) et (3.15) représentent les résultats des simulations des trois articulations pour l'essai à vide et pour les trois régulateurs flous glissants à 5, 7 et 9 classes d'appartenance. Les résultats obtenus montre l'amélioration des énergies des erreurs misent en évidence dans le tableau ci-dessous qui montre que le régulateur flou glissant est nettement meilleur que le régulateur flou simple. Toutes les énergies des erreurs ont été carrément divisées par 10 avec notamment l'élimination totale des erreurs statique. Cette amélioration reste valable pour les essais en charge 5 et 10kg respectivement (Voir figures (3.10), (3.11), (3.13), (3.14), (3.16) et (3.17)), avec toujours un léger accroissement des erreurs des 3 articulations.

Nous avons comparé les résultats obtenus pour la commande floue simple avec ceux de la commande par mode glissant flou et ceci pour les essais à vide et en charge. Ces résultats ont été repris dans le tableau qui suit :

Essai effectué		5 classes d'appartenance	7 classes d'appartenance	9 classes d'appartenance
Commande floue à vide	e ₁	6.905 10 ⁻⁷	2.075 10 ⁻⁶	4.138 10 ⁻⁶
	e ₂	2.164 10 ⁻⁶	2.396 10 ⁻⁶	3.349 10 ⁻⁶
	e ₃	8.449 10 ⁻⁶	1.322 10 ⁻⁵	1.537 10 ⁻⁵
	total	1.13 10 ⁻⁵	1.76 10 ⁻⁵	2.28 10 ⁻⁵
Commande floue en charge (10kg)	e ₁	1.569 10 ⁻⁶	4.612 10 ⁻⁶	1.085 10 ⁻⁵
	e ₂	5.489 10 ⁻⁶	7.157 10 ⁻⁶	1.085 10 ⁻⁵
	e ₃	1.406 10 ⁻⁴	2.566 10 ⁻⁴	3.150 10 ⁻⁴
	total	1.47 10 ⁻⁴	2.68 10 ⁻⁴	3.36 10 ⁻⁴
Commande par mode glissant floue à vide	e ₁	1.430 10 ⁻⁷	1.505 10 ⁻⁷	1.173 10 ⁻⁷
	e ₂	2.691 10 ⁻⁷	2.754 10 ⁻⁷	2.180 10 ⁻⁷
	e ₃	1.015 10 ⁻⁶	1.805 10 ⁻⁶	7.050 10 ⁻⁷
	total	1.42 10 ⁻⁶	2.23 10 ⁻⁶	1.04 10 ⁻⁶
Commande par mode glissant floue En charge (10kg)	e ₁	3.272 10 ⁻⁷	3.510 10 ⁻⁷	3.312 10 ⁻⁷
	e ₂	1.760 10 ⁻⁷	1.698 10 ⁻⁷	1.313 10 ⁻⁷
	e ₃	1.781 10 ⁻⁵	2.561 10 ⁻⁵	1.555 10 ⁻⁵
	total	1.831 10 ⁻⁵	2.61 10 ⁻⁵	1.601 10 ⁻⁵

Avec e_i : l'erreur de l'énergie de la i^{ème} articulation.

Nous avons effectué par la suite les mêmes tests de robustesses qu'utilisés auparavant, en choisissant le meilleur régulateur flou glissant. Ici, contrairement au réglage par la logique floue classique, l'ordre de performance est 9, 5 et 7 classes d'appartenance.

- **Test avec lâché de charge :**

La figure (3.18) représente le résultat des simulations des trois articulations avec introduction d'une charge de 10kg qui sera relâchée après un temps de 0.75s, en utilisant le régulateur à 9 classes d'appartenance, on a une très bonne réponse des commandes avec apparitions de pics dès le lâché de charge, mais elles reviennent à leurs états de stabilité

- **Test de rupture de commande :**

La figure (3.19) représente le résultat des simulations des trois articulations, avec rupture de la deuxième commande après un temps de 0.75s. Les erreurs de poursuite restent dans des plages admissibles et les commandes sont réalisables et acceptables.

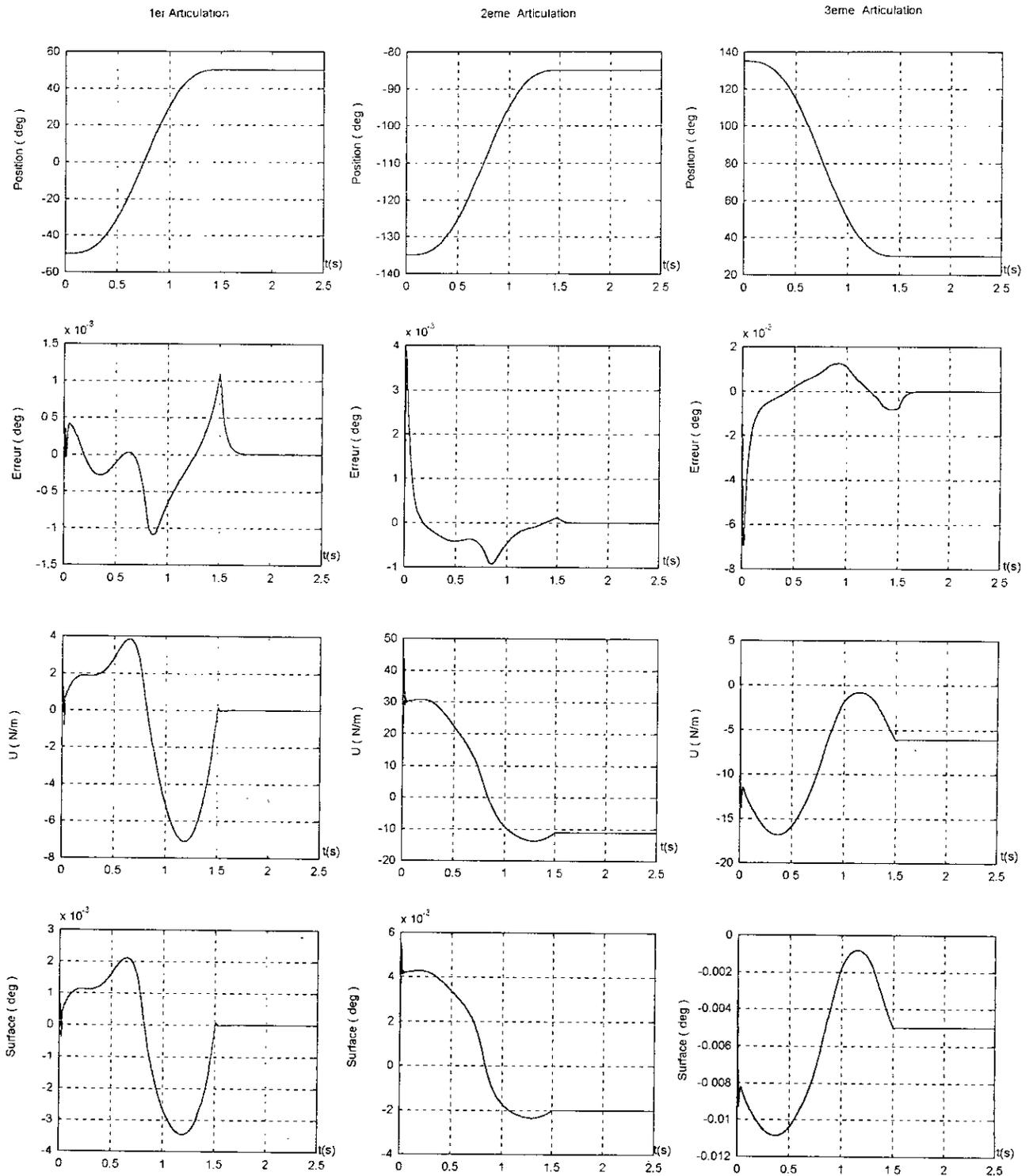


Figure (3.9) : Application de la commande par mode de glissement flou (5 classes) au PUMA 560. Poursuite à vide

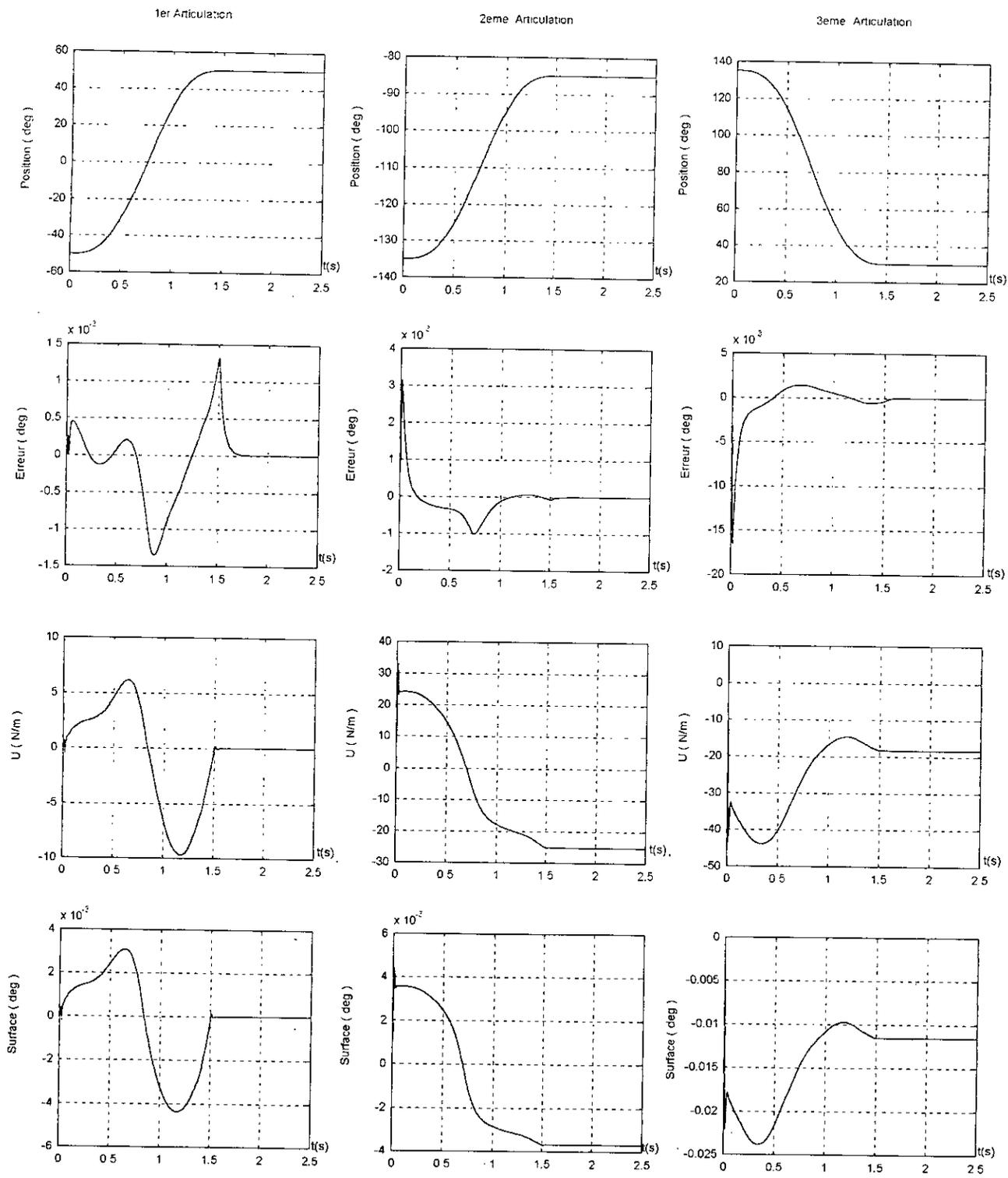


Figure (3.10) : Application de la commande par mode de glissement floue (5 classes) au PUMA 560. Poursuite avec effet de la charge $m_0 = 5\text{kg}$

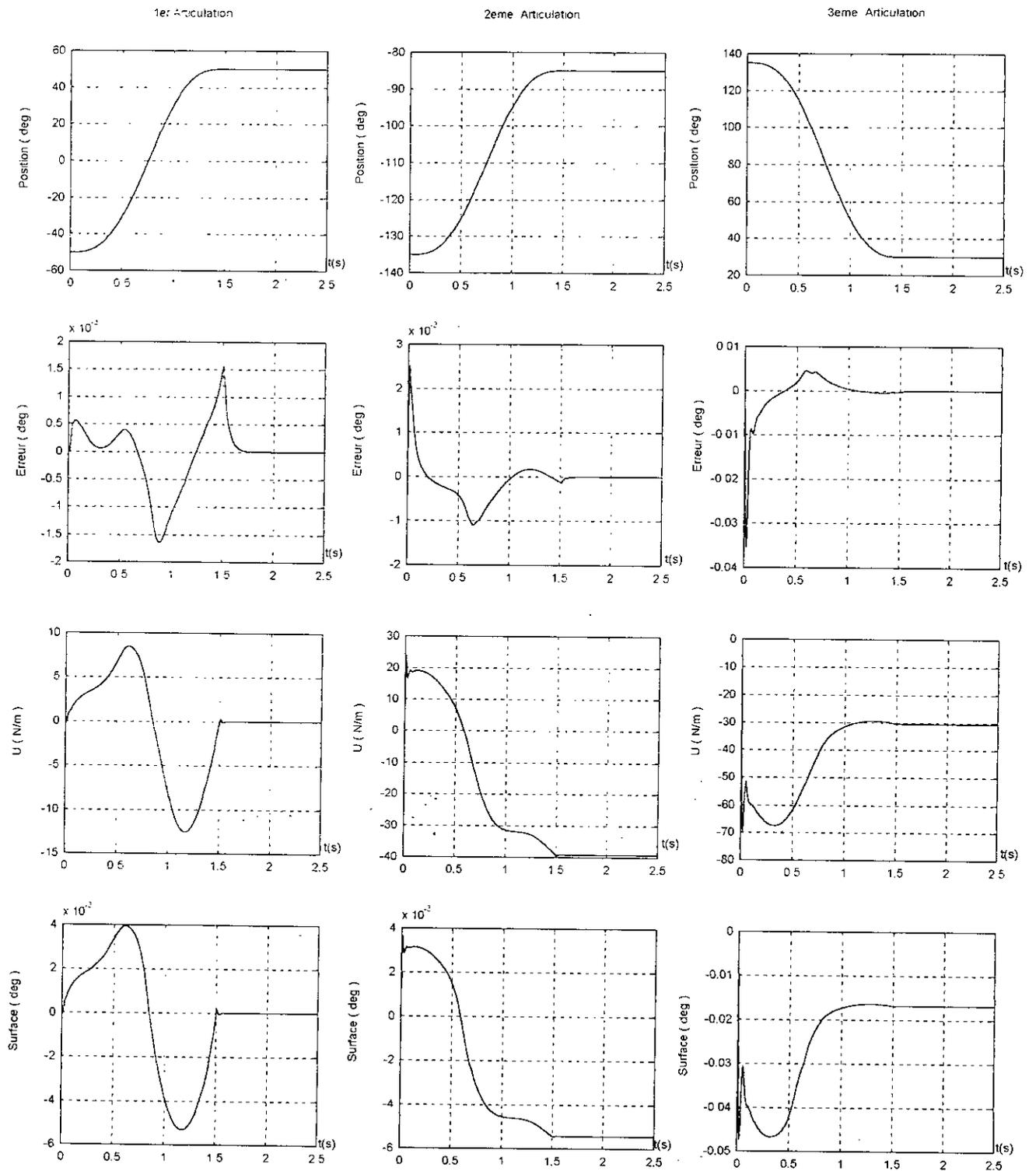


Figure (3.11) : Application de la commande par mode de glissement floue (5 classes) au PUMA 560. Poursuite avec effet de la charge $m_0 = 10\text{kg}$

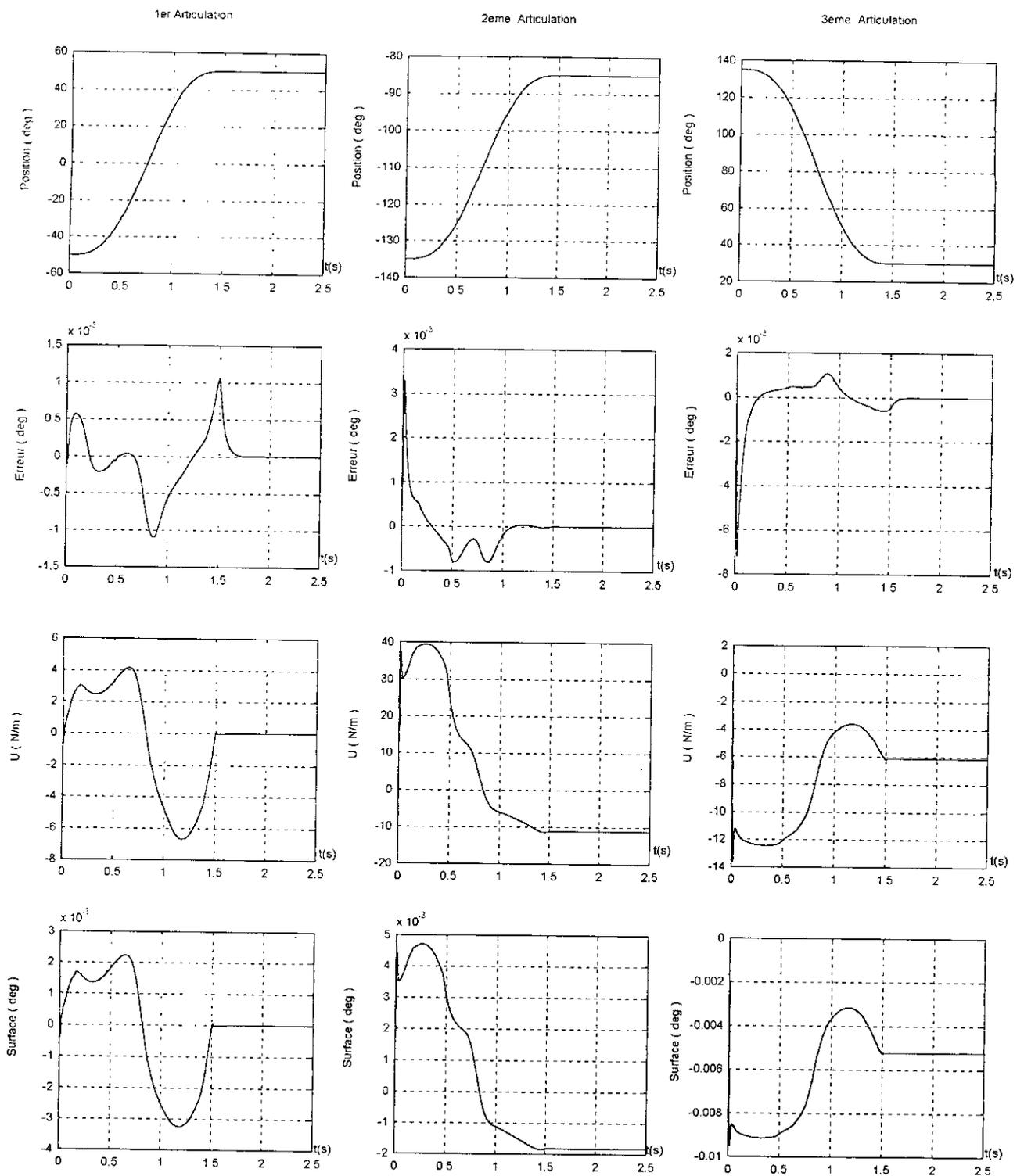


Figure (3.12) : Application de la commande par mode de glissement fluide (7 classes) au PUMA 560. Poursuite à vide

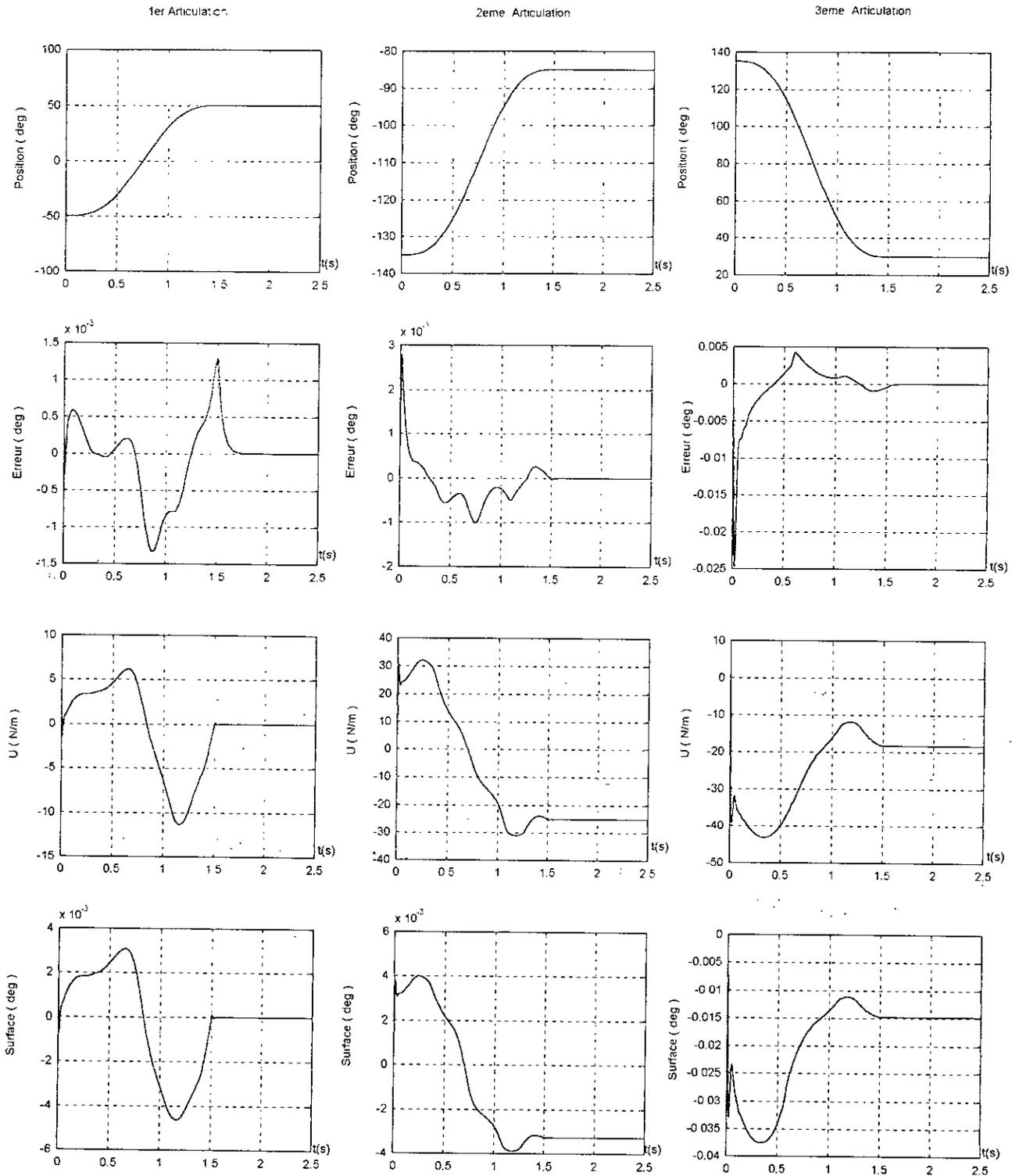


Figure (3.13) : Application de la commande par mode de glissement flou (7 classes) au PUMA 560. Poursuite avec effet de la charge $m_0 = 5\text{kg}$

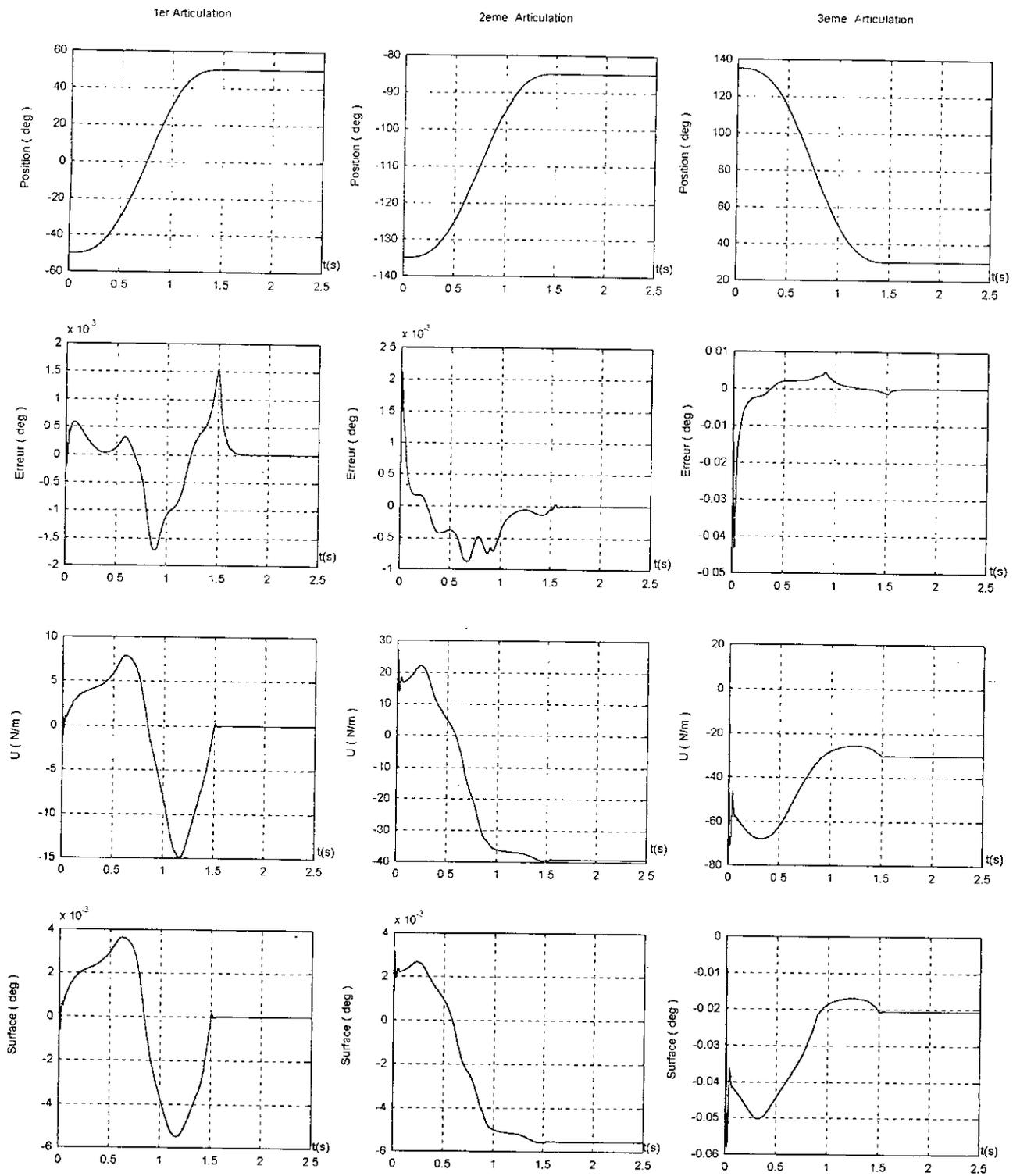


Figure (3.14) : Application de la commande par mode de glissement flou (7 classes) au PUMA 560. Poursuite avec effet de la charge $m_0 = 10\text{kg}$

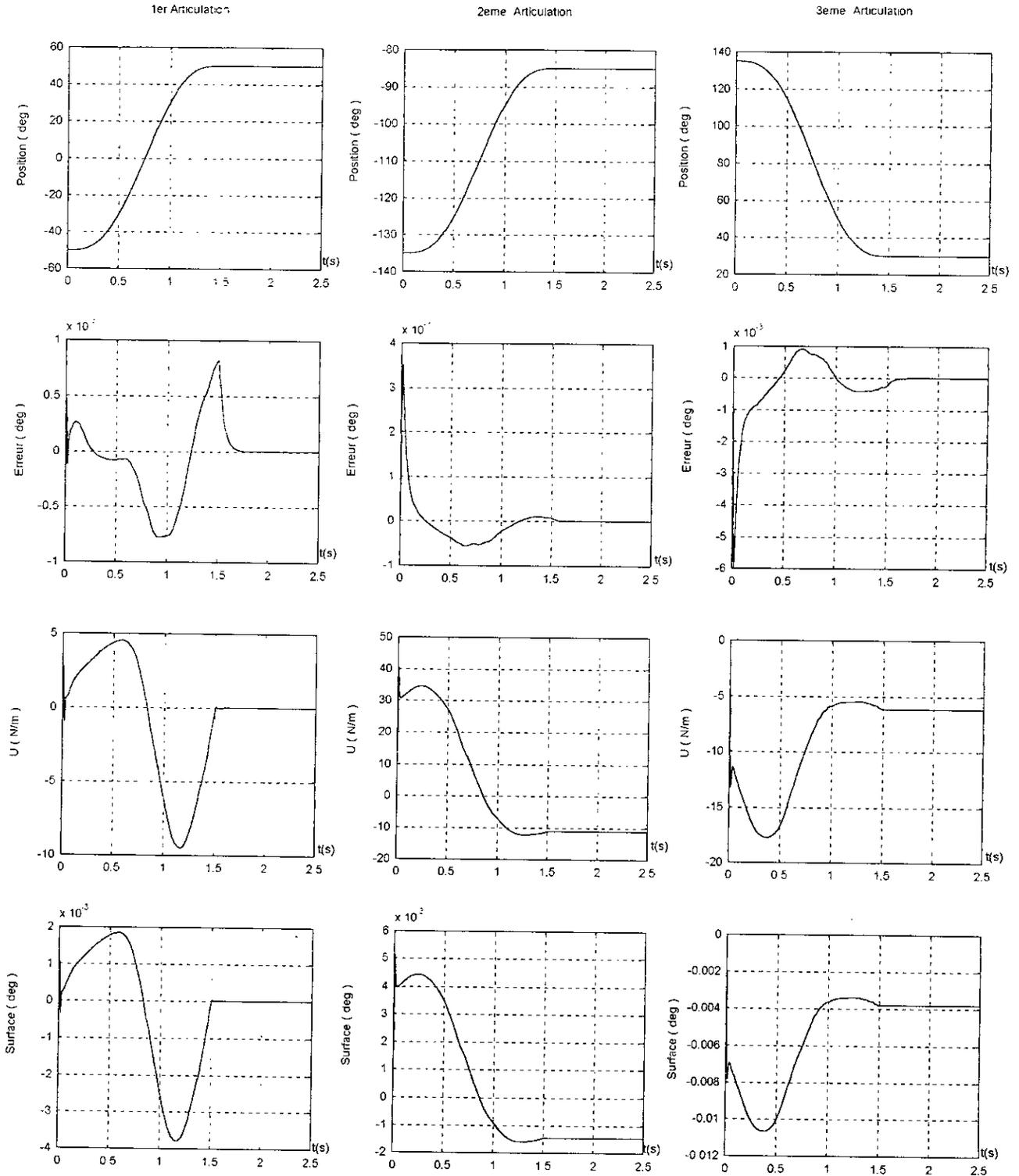


Figure (3.15) : Application de la commande par mode de glissement flou (9 classes) au PUMA 560. Poursuite à vide

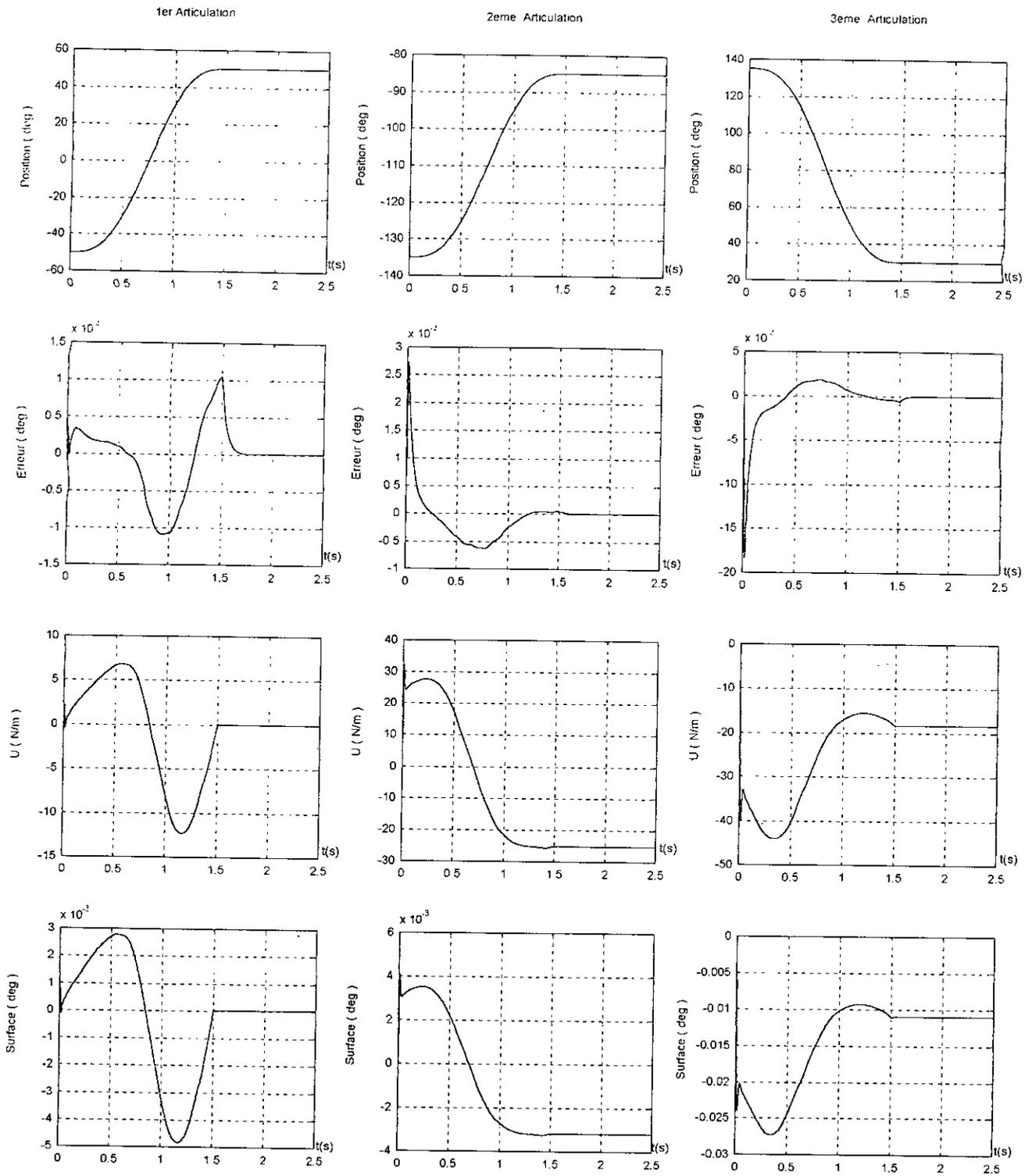


Figure (3.16) : Application de la commande par mode de glissement floue (9 classes) au PUMA 560. Poursuite avec effet de la charge $m_0 = 5\text{kg}$

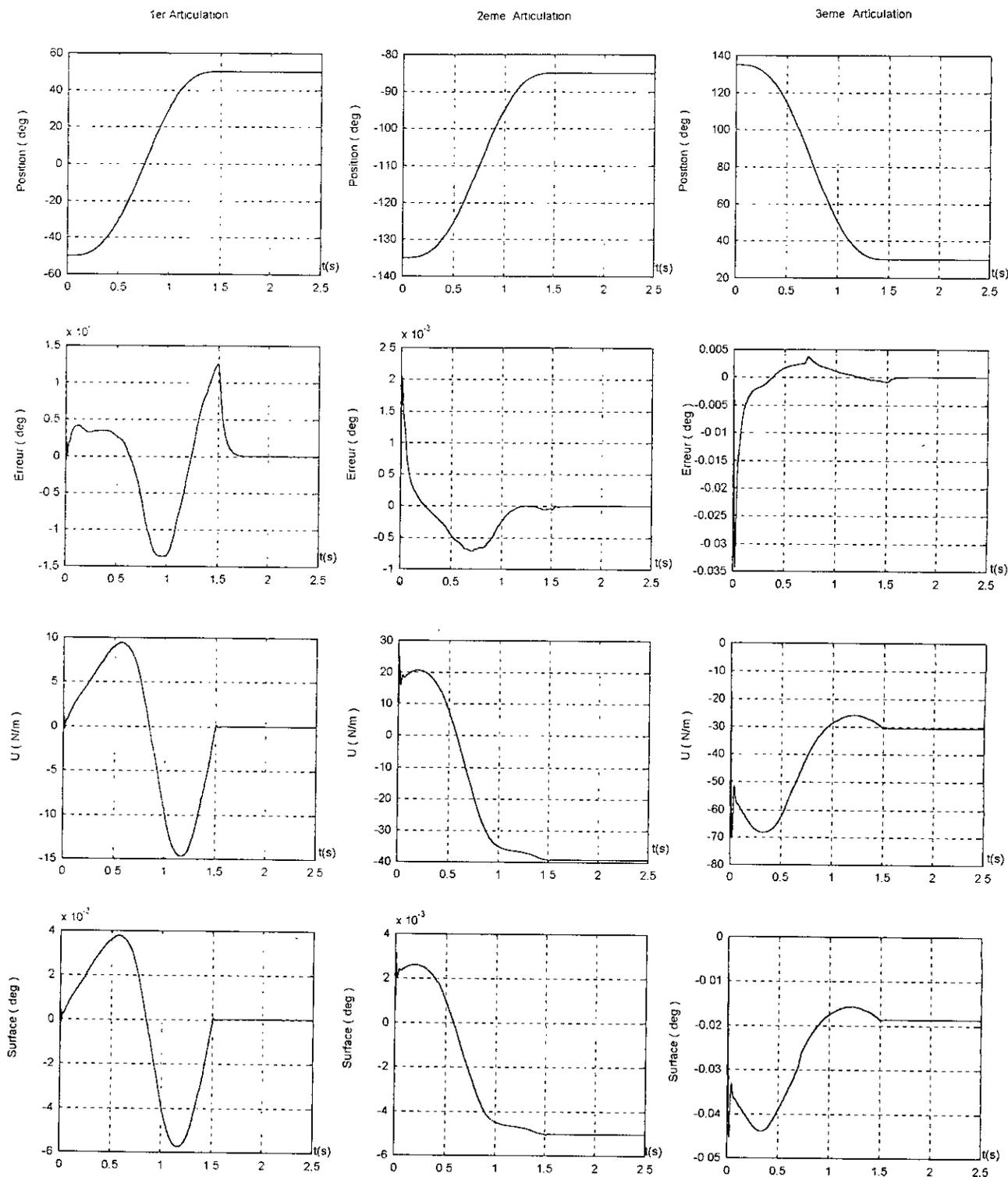


Figure (3.17) : Application de la commande par mode de glissement floue (9 classes) au PUMA 560. Poursuite avec effet de la charge $m_0 = 10\text{kg}$

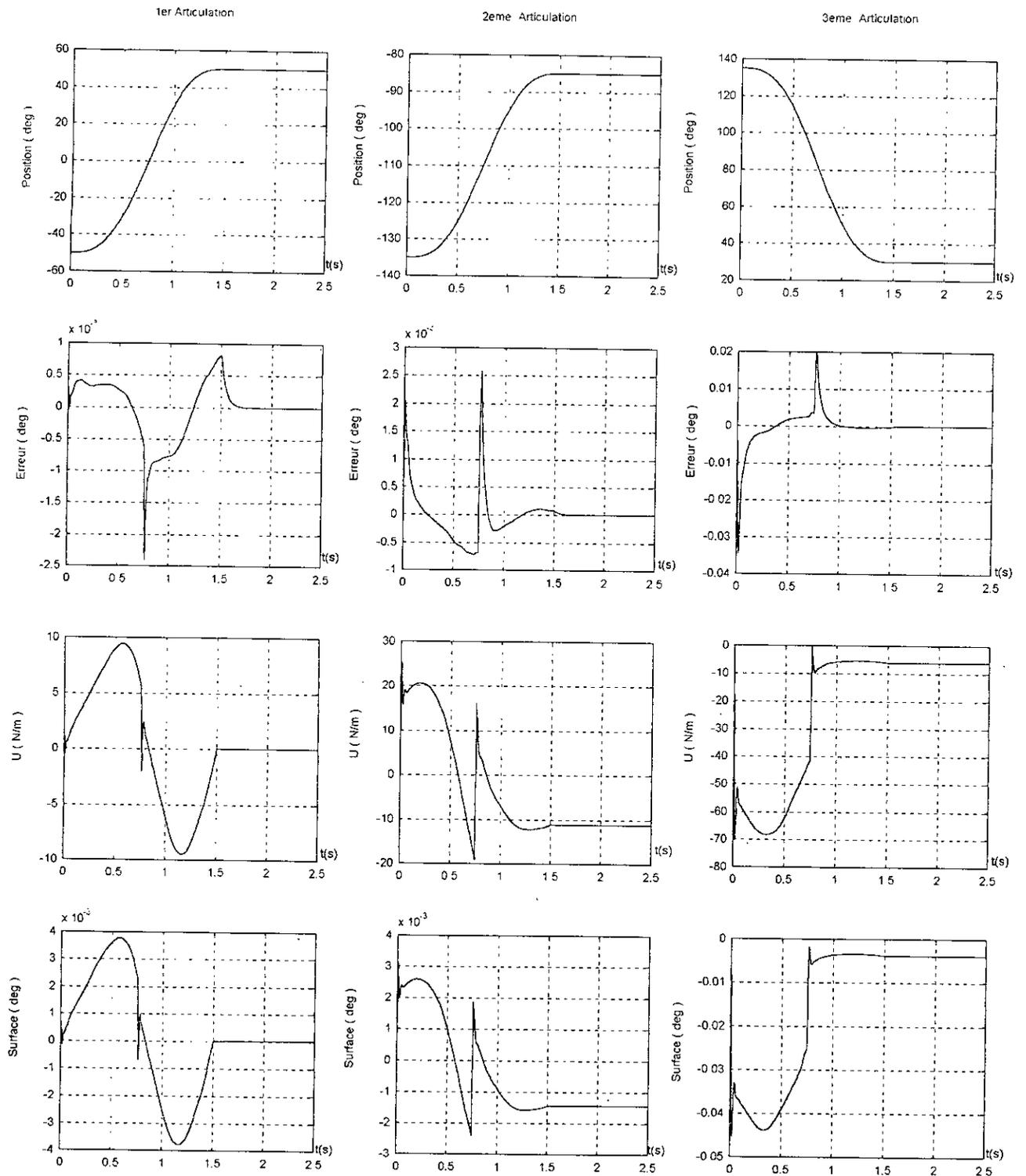


Figure (3.18) : Application de la commande par mode de glissement floue (9 classes) au PUMA 560
 Poursuite avec effet de la charge $m_0 = 10\text{kg}$ et la décharge à l'instant $t=0.75\text{s}$

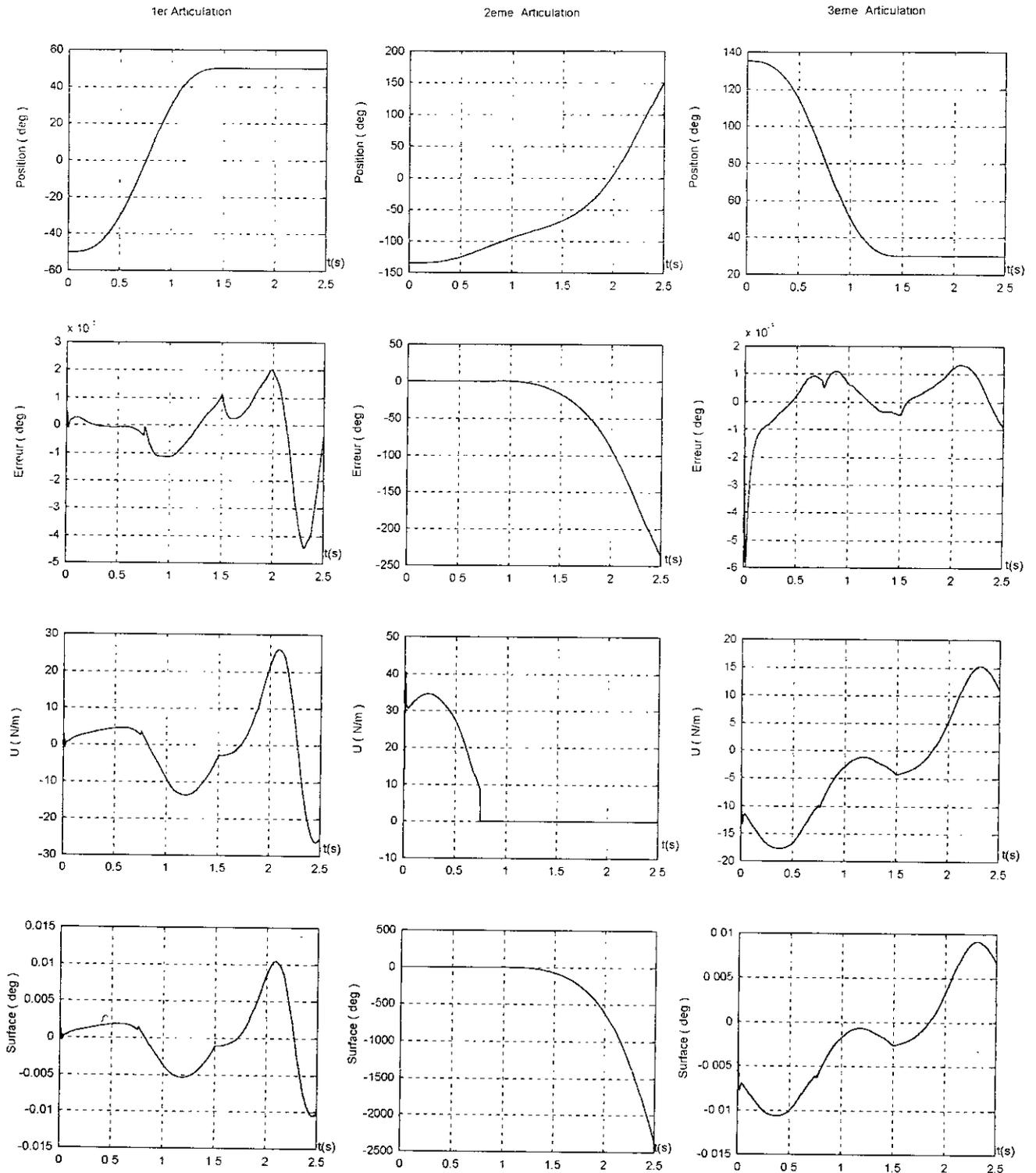


Figure (3.19) : Application de la commande par mode de glissement flou (9 classes) au PUMA 560. Poursuite avec rupture de la deuxième commande à l'instant $t=0.75s$

3.9.2 Etude de L'influence des coefficients des surfaces :

Nous avons effectué dans le chapitre précédent une étude approfondie sur l'influence de la variation des gains sur le robot PUMA 560. Dans ce qui suit, nous nous proposons d'étudier l'influence des coefficients des surfaces de glissement sur le même robot.

Nous allons varier les coefficients de chaque surface de glissement (9 coefficients) et observer les réactions du système, tant pour l'essai à vide qu'en charge.

Coefficients de l'intégrale de l'erreur des trois surfaces :

En variant ces coefficients nous observons les mêmes réactions du système et ceci pour les trois articulations.

On peut dire que ces coefficients n'influent pas sur la stabilité des commandes, sauf quand on dépasse une certaine limite (environ 40), au-delà de laquelle il y a apparition d'oscillations pour les deux régimes (dynamique et statique), les commandes restent stables même pour une valeur nulle de ces coefficients (surface sous forme PD).

D'après les graphes ci dessous on peut dire que chaque énergie de l'erreur diminue quand on augmente le coefficient correspondant.

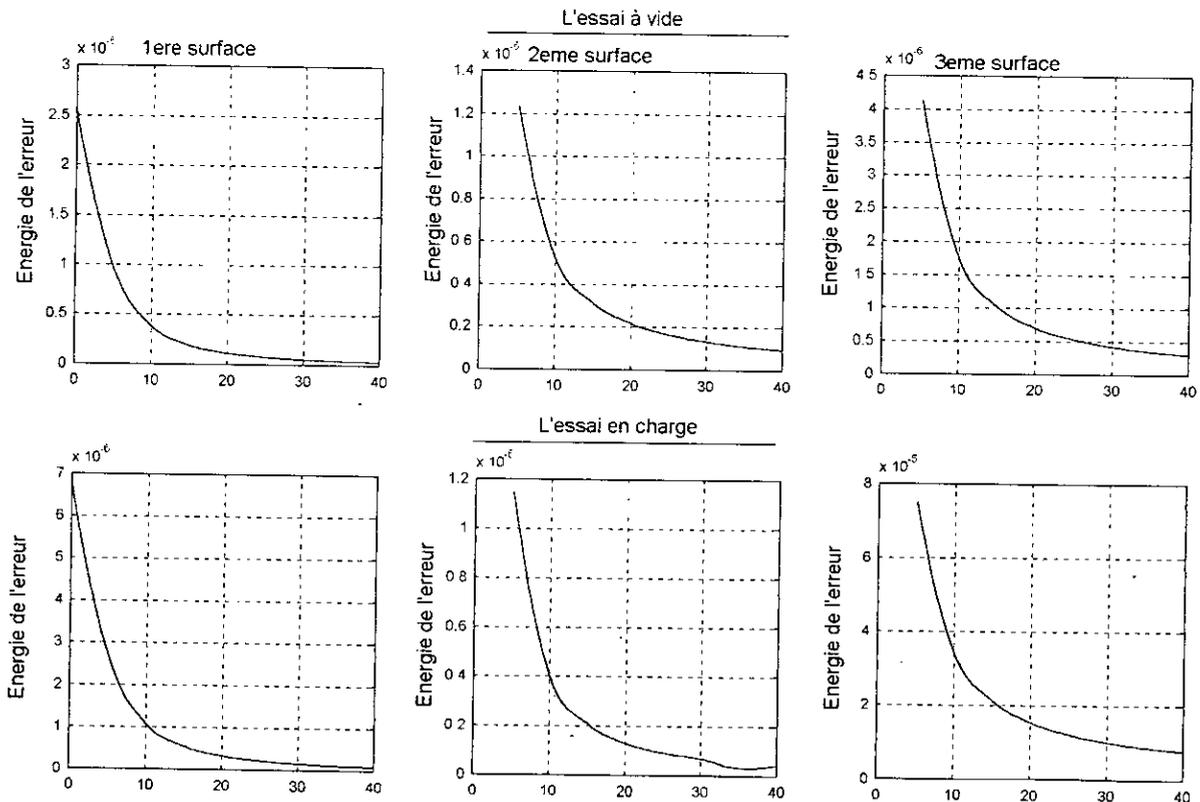


Figure 3.20 : Variation du coefficient de l'intégrale de l'erreur

Coefficients des erreurs :

Première surface :

En augmentant ce coefficient on aura une apparition de grandes oscillations sur les deux premières commandes et de petites oscillations sur la dernière, par contre sa diminution n'aura aucun effet, car les commandes restent stables.

Les deux graphes ci dessous, qui ont la même allure, montrent une diminution de l'énergie de l'erreur jusqu'à une certaine limite, au-delà de laquelle elle reprend sa croissance.

Deuxième et troisième surface :

On a une même réaction du système pour la variation des deux coefficients. La diminution n'influe pas sur le système, par contre l'augmentation progressive introduit des pics au début des commandes 1 et 2, qui deviennent de grandes oscillations pour l'essai en charge, et pour l'essai à vide l'apparition de pics puis de grandes oscillations sur tout l'intervalle de simulation pour les trois articulations.

Concernant l'énergie de l'erreur on observe :

- Pour l'essai à vide : une courbe décroissante jusqu'à une certaine limite et croissante au-delà.
- Pour l'essai en charge : une courbe décroissante.

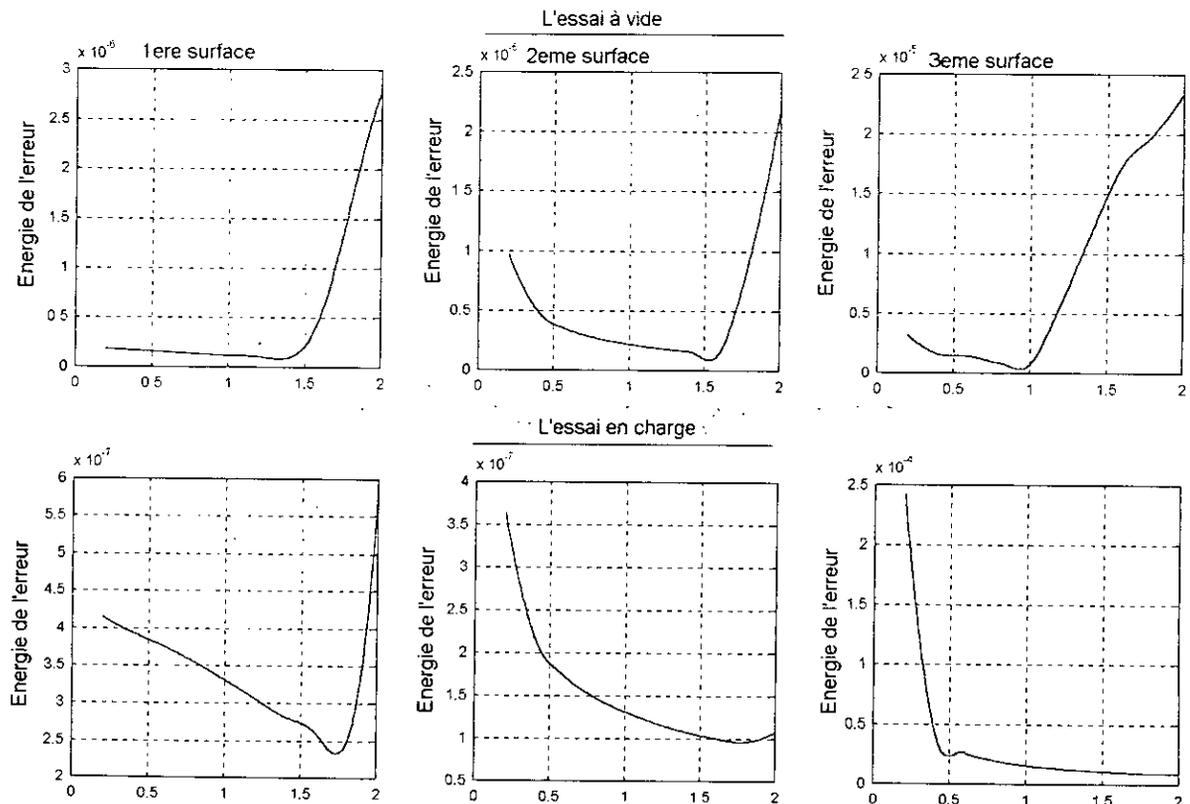


Figure 3.21 : Variation du coefficient de l'erreur

Coefficients des dérivées des erreurs :

Première et deuxième surface :

On remarque une similitude des réactions des commandes et les allures des courbes d'énergie pour les deux coefficients. En augmentant l'un des deux coefficients, on aura des oscillations en régime dynamique et statique pour l'essai à vide, mais pour l'essai en charge on aura de grandes oscillations en régime dynamique des commandes 1 et 2 et de petites oscillations pour la dernière commande. En diminuant chacun des deux coefficients au aura des commandes admissibles, sauf pour 0 (surface de la forme PI) où on aura de petites oscillations au régime dynamique des deux premières commandes. En ce qui concerne les courbes de l'énergie, on remarque la même allure c'est à dire légèrement décroissante en premier, arriver à 0.007 elle devient croissante.

Troisième surface :

En diminuant ce coefficient on garde de bons résultats, même si on observe quelques oscillations au régime dynamique des trois commandes lors de l'essai à charge.

L'augmentation de ce coefficient donne des commandes complètement instables. Concernant l'allure des courbes de l'énergie, qui ne sont pas identiques pour les deux essais :

- En charge : on observe une forte décroissance jusqu'au environ de 0.005 puis une croissance lente.
- A vide : on observe d'abord une légère décroissance jusqu'à la valeur de 0.005 puis une forte croissance de l'énergie de l'erreur.

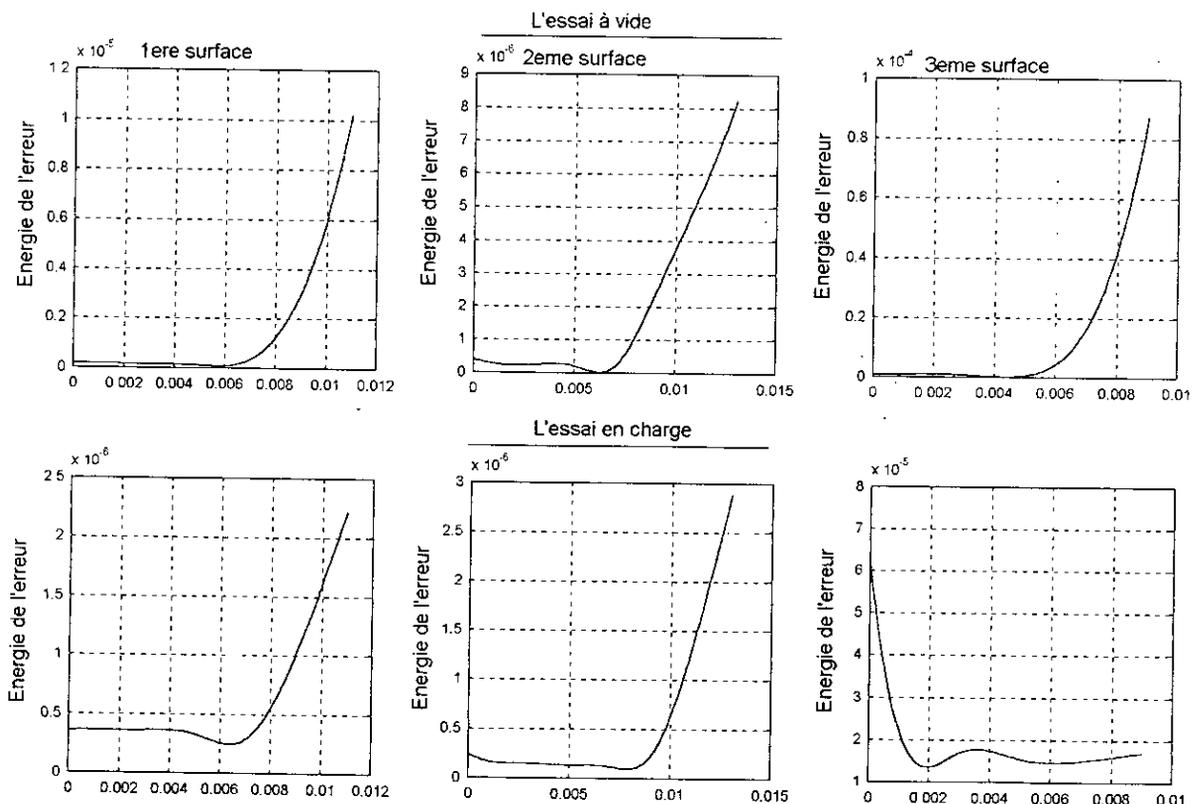


Figure 3.22 : Variation du coefficient de la dérivée de l'erreur

3.9.3 Application de la commande au robot à structure cylindrique :

Les commandes utilisées pour ce robot sont les mêmes que pour le robot puma 560, notamment ce qui concerne les fonctions d'appartenance (5, 7 et 9 classes d'appartenance).

La réalisation des simulations des lois de commande et l'algorithme de résolution utilisée sont similaire a ceux utilisée avec le robot puma 560.

Les figures (3.23), (3.26) et (3.29) représentent les résultats des simulations à vide pour les trois articulations, en utilisant les trois régulateurs flous (5, 7 et 9 classes d'appartenance), on obtient ainsi une bonne poursuite avec des commandes admissibles.

Les figures (3.24), (3.25), (3.27), (3.28), (3.30) et (3.31) représentent les résultats des simulations a charge (5kg respectivement 10kg) pour les trois articulations, en utilisant les trois régulateurs flous, on a de bon résultat avec des commandes réalisables, et cela pour les deux charges.

Pour comparer entre les trois régulateurs (5, 7 et 9 classes), on a effectuer un tableau comparatif, ce tableau nous compare aussi la régulation par commande floue et la commande par mode glissant-floue.

Essai effectué		5 classes d'appartenance	7 classes d'appartenance	9 classes d'appartenance
Commande floue A vide	e_1	$1.736 \cdot 10^{-7}$	$1.98 \cdot 10^{-7}$	$1.673 \cdot 10^{-7}$
	e_2	$3.223 \cdot 10^{-6}$	$9.225 \cdot 10^{-7}$	$2.386 \cdot 10^{-6}$
	e_3	$6.075 \cdot 10^{-7}$	$3.064 \cdot 10^{-7}$	$8.562 \cdot 10^{-7}$
	Total	$4.00 \cdot 10^{-6}$	$1.73 \cdot 10^{-6}$	$3.14 \cdot 10^{-6}$
Commande floue en charge (10kg)	e_1	$4.492 \cdot 10^{-7}$	$4.841 \cdot 10^{-7}$	$4.931 \cdot 10^{-7}$
	e_2	$7.087 \cdot 10^{-6}$	$1.693 \cdot 10^{-6}$	$4.271 \cdot 10^{-6}$
	e_3	$1.079 \cdot 10^{-6}$	$6.996 \cdot 10^{-7}$	$1.667 \cdot 10^{-6}$
	total	$8.61 \cdot 10^{-6}$	$2.87 \cdot 10^{-6}$	$6.43 \cdot 10^{-6}$
Commande par mode glissant-floue à vide	e_1	$6.902 \cdot 10^{-8}$	$5.29 \cdot 10^{-8}$	$1.045 \cdot 10^{-7}$
	e_2	$1.573 \cdot 10^{-7}$	$1.8 \cdot 10^{-7}$	$2.700 \cdot 10^{-7}$
	e_3	$2.07 \cdot 10^{-7}$	$1.993 \cdot 10^{-7}$	$3081 \cdot 10^{-7}$
	total	$4.33 \cdot 10^{-7}$	$4.32 \cdot 10^{-7}$	$7.56 \cdot 10^{-7}$
Commande par mode glissant-floue à charge (10kg)	e_1	$2.335 \cdot 10^{-7}$	$1.741 \cdot 10^{-7}$	$3.802 \cdot 10^{-7}$
	e_2	$3.530 \cdot 10^{-7}$	$3.234 \cdot 10^{-7}$	$4.976 \cdot 10^{-7}$
	e_3	$6.782 \cdot 10^{-7}$	$4.881 \cdot 10^{-7}$	$1.036 \cdot 10^{-6}$
	total	$1.265 \cdot 10^{-6}$	$9.857 \cdot 10^{-7}$	$1.914 \cdot 10^{-6}$

Avec e_i : l'erreur de l'énergie de la $i^{ème}$ articulation.

En effectuant une comparaison entre les différents régulateurs flous glissant, on trouve que le meilleur régulateur est celui à 7 classes (comme pour la commande floue), suivi de celui à 5 classes puis celui à 9 classes. En comparant entre la régulation par logique floue et par mode de glissement flou, on peut remarquer l'amélioration des énergies des erreurs de chaque articulation et cela que ce soit pour le 9 classes ou le 7 classes ou le 5 classes.

- **Test avec lâché de charge :**

La figure (3.31) représente le résultat des simulations des trois articulations avec introduction d'une charge de 10kg qui sera relâché après un temps de 1s, on peut dire que ce test est réussi malgré les pics qui sont vite atténués grâce à la rapidité du système.

- **Test de capacité de poursuite :**

La figure (3.32) représente le résultat des simulations des trois articulations en appliquant la trajectoire de VIVIANI, malgré l'apparition de forts pics les commandes retournent à leur état de stabilité, cela montre la capacité de poursuite du système.

- **Test avec introduction de perturbations externe :**

La figure (3.33) représente le résultat des simulations des trois articulations avec introduction de perturbation avec augmentation de l'amplitude de chaque commande de 1V à $t=1s$, ce test a été réussi avec un rejet total de la perturbation.

- **Test avec variation paramétrique :**

La figure (3.34) représente le résultat des simulations des trois articulations, avec variation des paramètres mécaniques du robot de 100% après l'instant $t=1s$, On la bonne réaction du système avec des commandes admissibles.

3.10 Etude comparative :

Le critère utilisé précédemment démontre que pour le PUMA 560 le régulateur à 9 classes était le meilleur suivi par celui à 5 classes puis celui à 7 classes. Par contre, et pour le robot à structure cylindrique c'est le régulateur à 7 classes qui est meilleur puis celui à 9 classes puis celui à 5 classes, donc comme pour la commande floue, on ne peut pas formuler de critère sur le choix du nombre de classes.

3.11 Conclusion :

Dans ce chapitre, nous avons d'abord présenté des généralités sur la commande à structure variable et plus précisément sur la régulation par mode de glissement, qui sera utilisée par la suite en combinaison avec la commande floue pour constituer un nouveau régulateur. La nouvelle commande a été appliquée sur les deux robots (PUMA 560 et à structure cylindrique), qui a donné des résultats meilleurs que ceux trouvés précédemment (régulateur flou simple). Nous avons utilisé, comme dans le précédent chapitre, trois classes d'appartenance (5, 7 et 9).

Pour étudier le meilleur régulateur flou glissant appliqué au robot PUMA 560, nous avons varié les coefficients des surfaces et observé leurs influences sur le système. Néanmoins, la réalisation d'une telle commande est complexe et prend un temps de calcul considérable à cause des nombreuses étapes de son élaboration.

C'est la raison pour laquelle nous nous proposons d'appliquer, dans le prochain chapitre, les réseaux de neurones pour simplifier la structure de la commande par mode glissant floue.

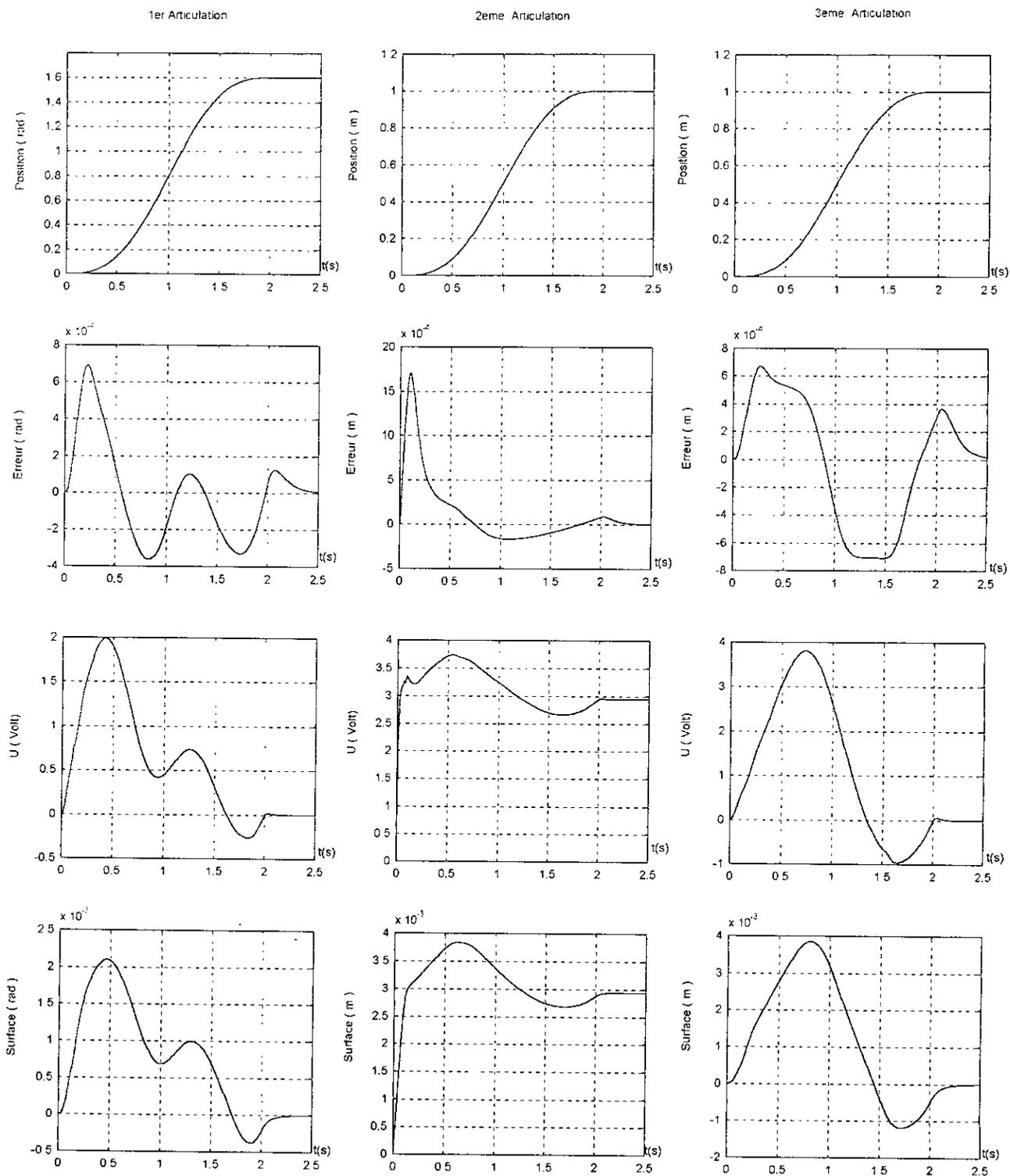


Figure (3.23) : Application de la commande par mode de glissement flou (5 classes) au robot à structure cylindrique. Poursuite à vide

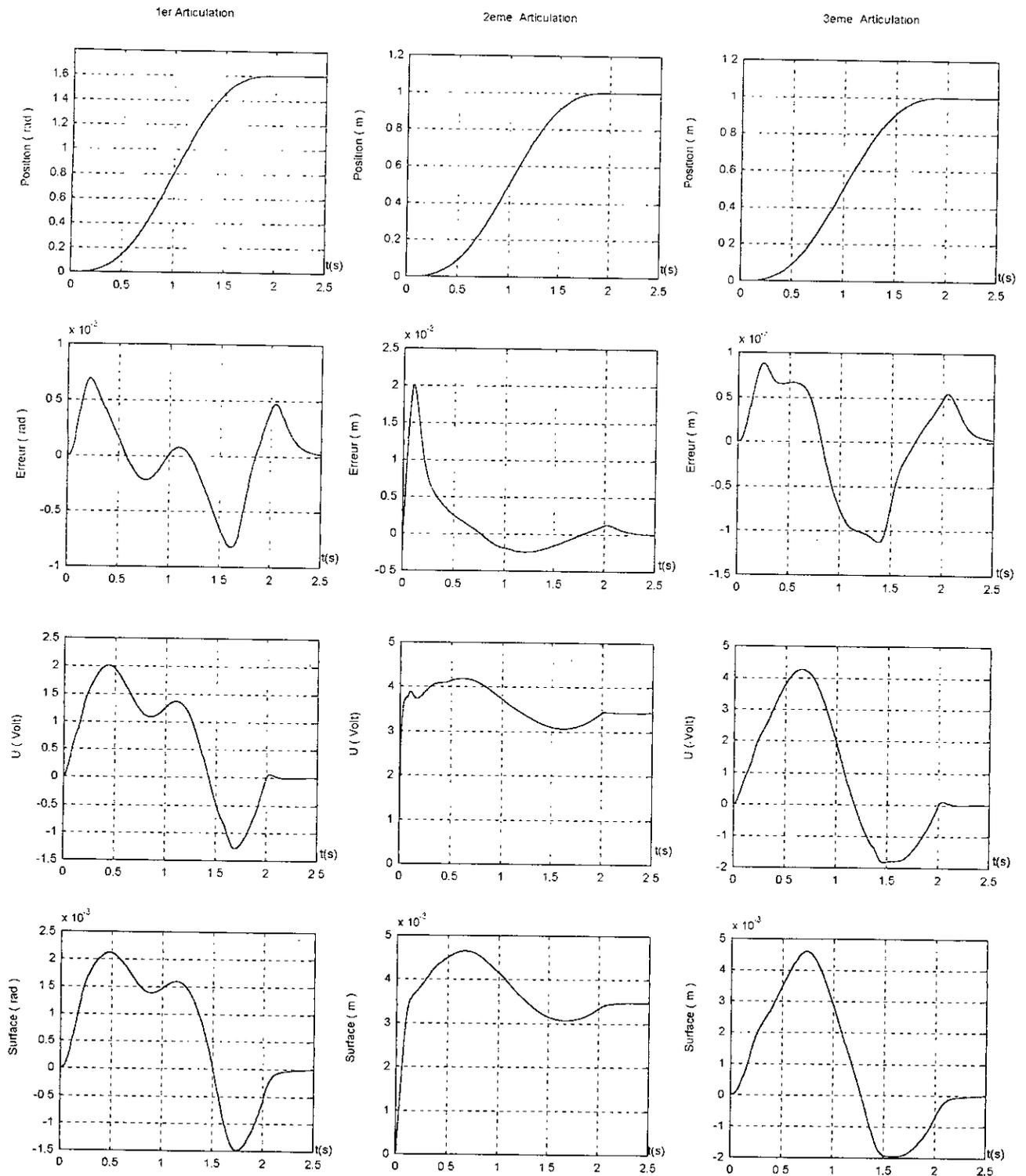


Figure (3.24) : Application de la commande par mode de glissement flou (5 classes) au robot à structure cylindrique. Poursuite avec effet de la charge $m_0 = 5\text{kg}$

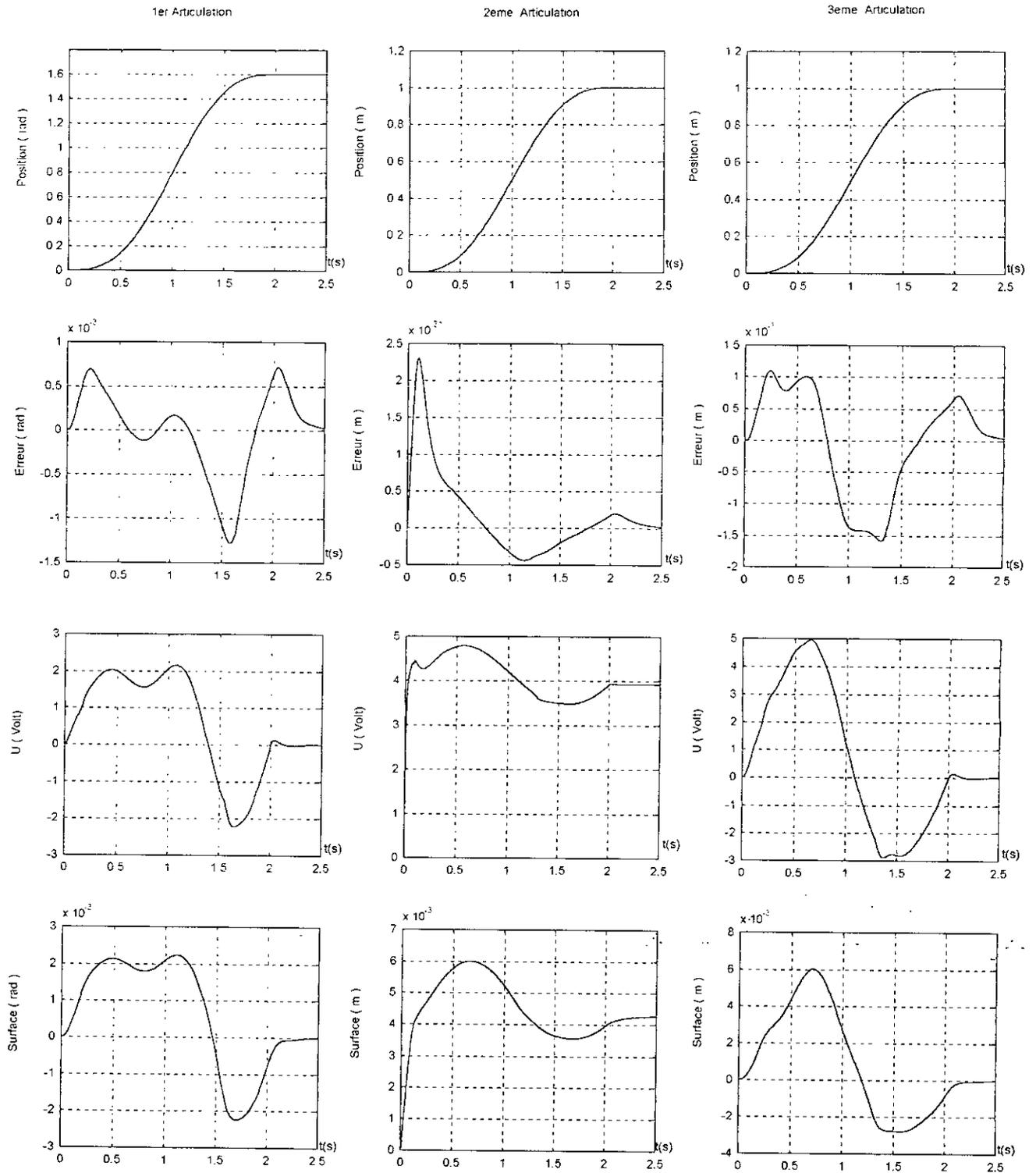


Figure (3.25) : Application de la commande par mode de glissement flou (5 classes) au robot à structure cylindrique
Poursuite avec effet de la charge $m_0 = 10\text{kg}$

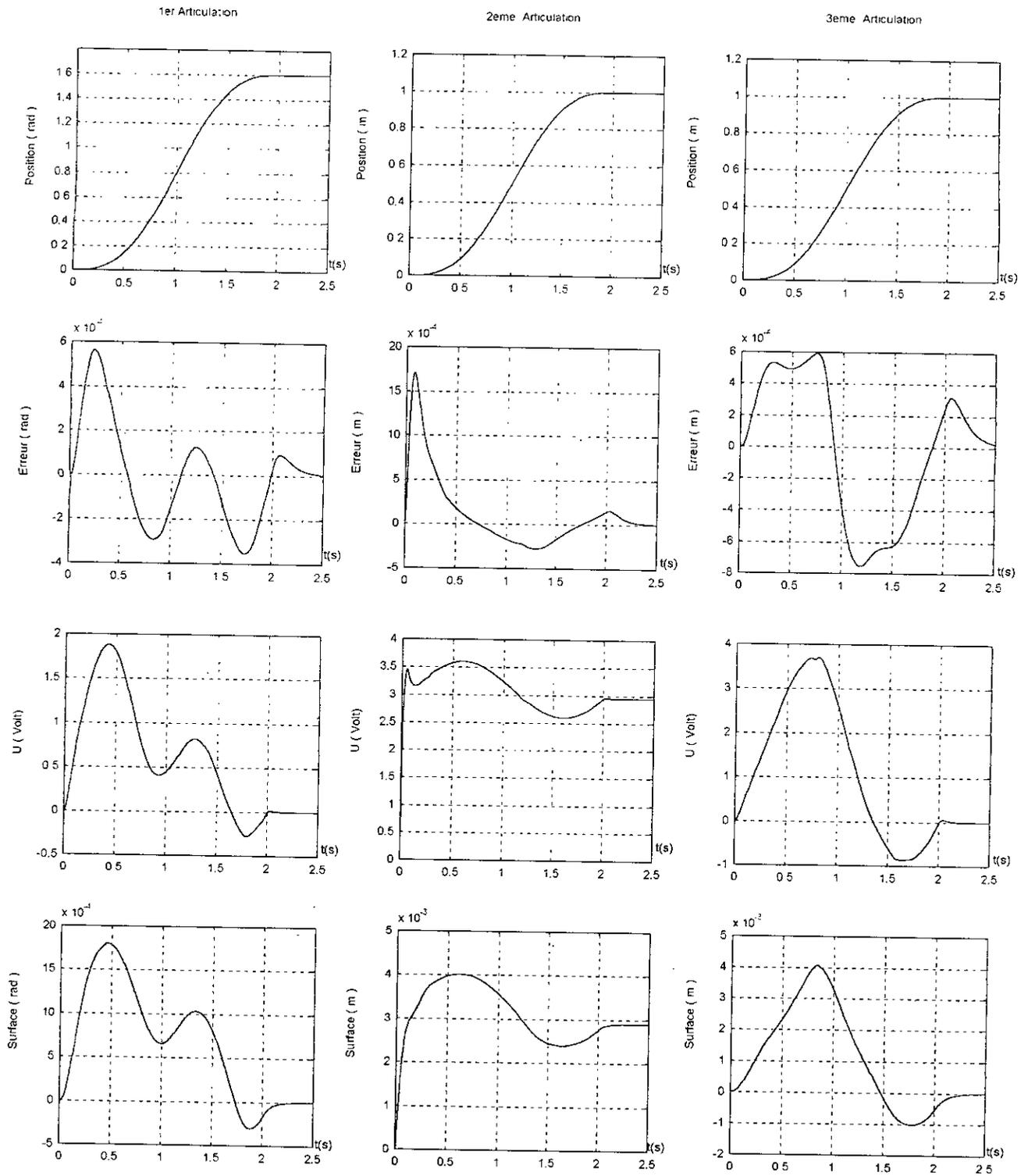


Figure (3.26) : Application de la commande par mode de glissement flou (7 classes) au robot à structure cylindrique. Poursuite à vide

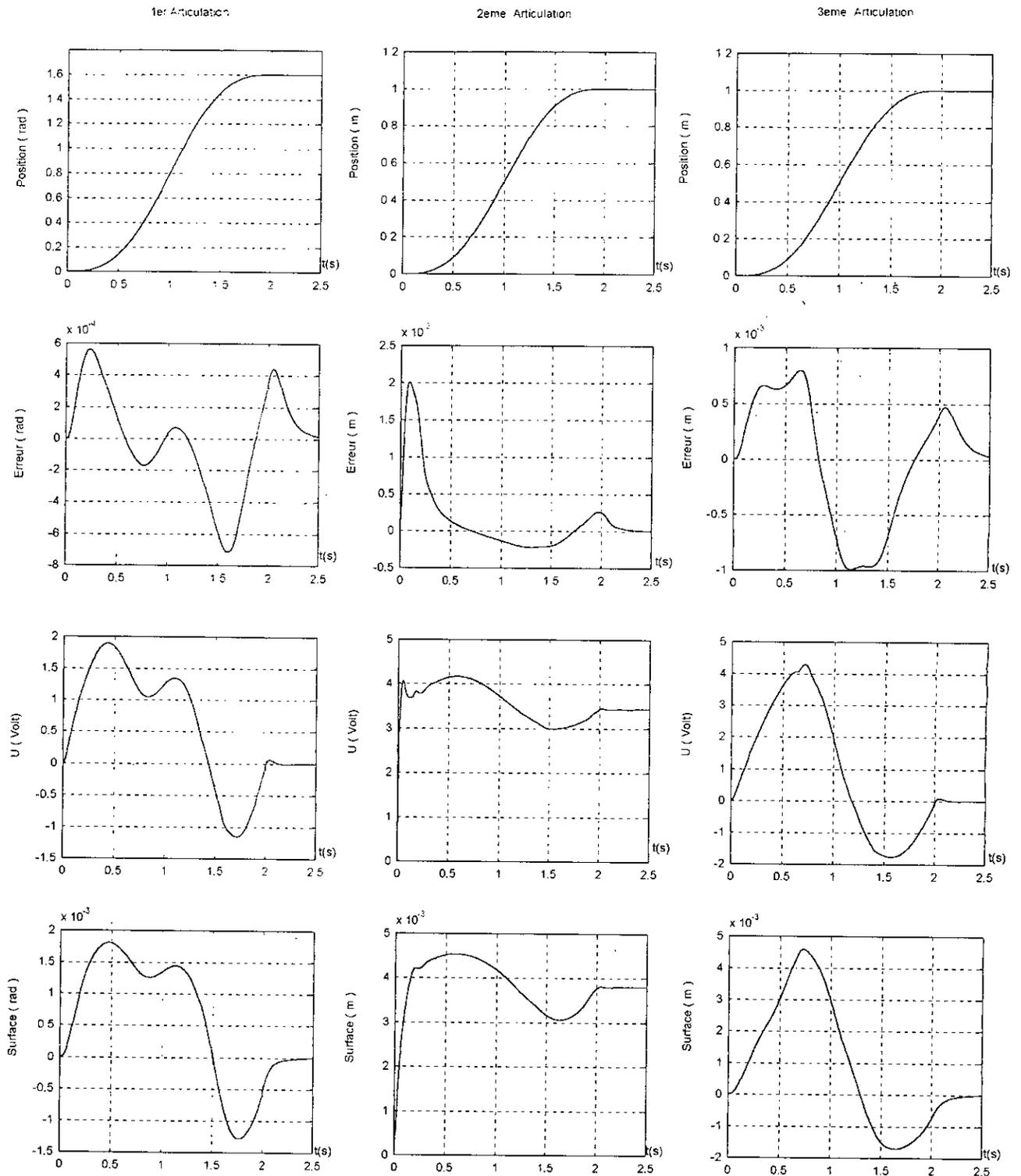


Figure (3.27) : Application de la commande par mode de glissement flou (7 classes) au robot à structure cylindrique
Poursuite avec effet de la charge $m_0 = 5\text{kg}$

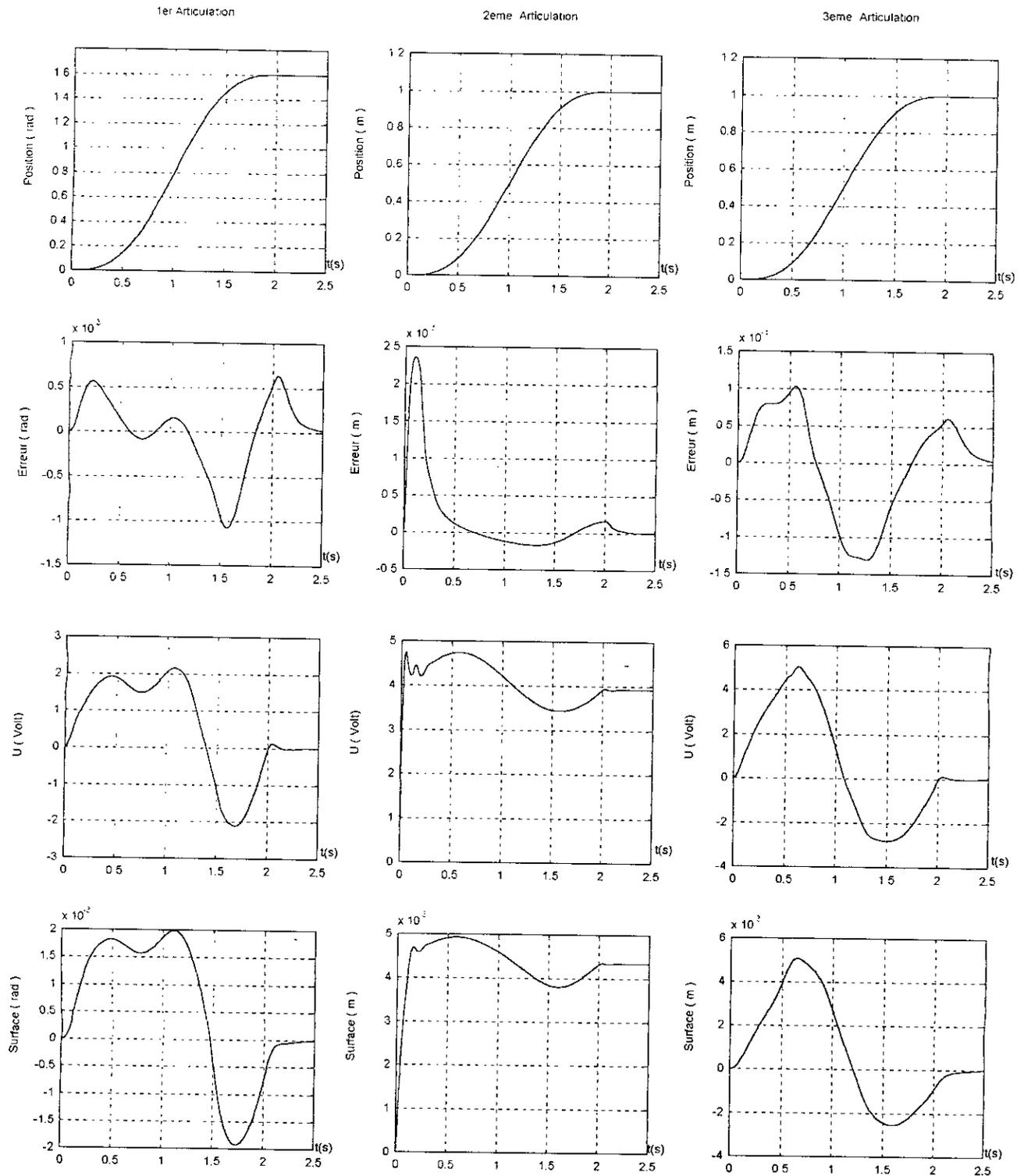


Figure (3.28) : Application de la commande par mode de glissement flou (7 classes) au robot à structure cylindrique
Poursuite avec effet de la charge $m_0 = 10\text{kg}$

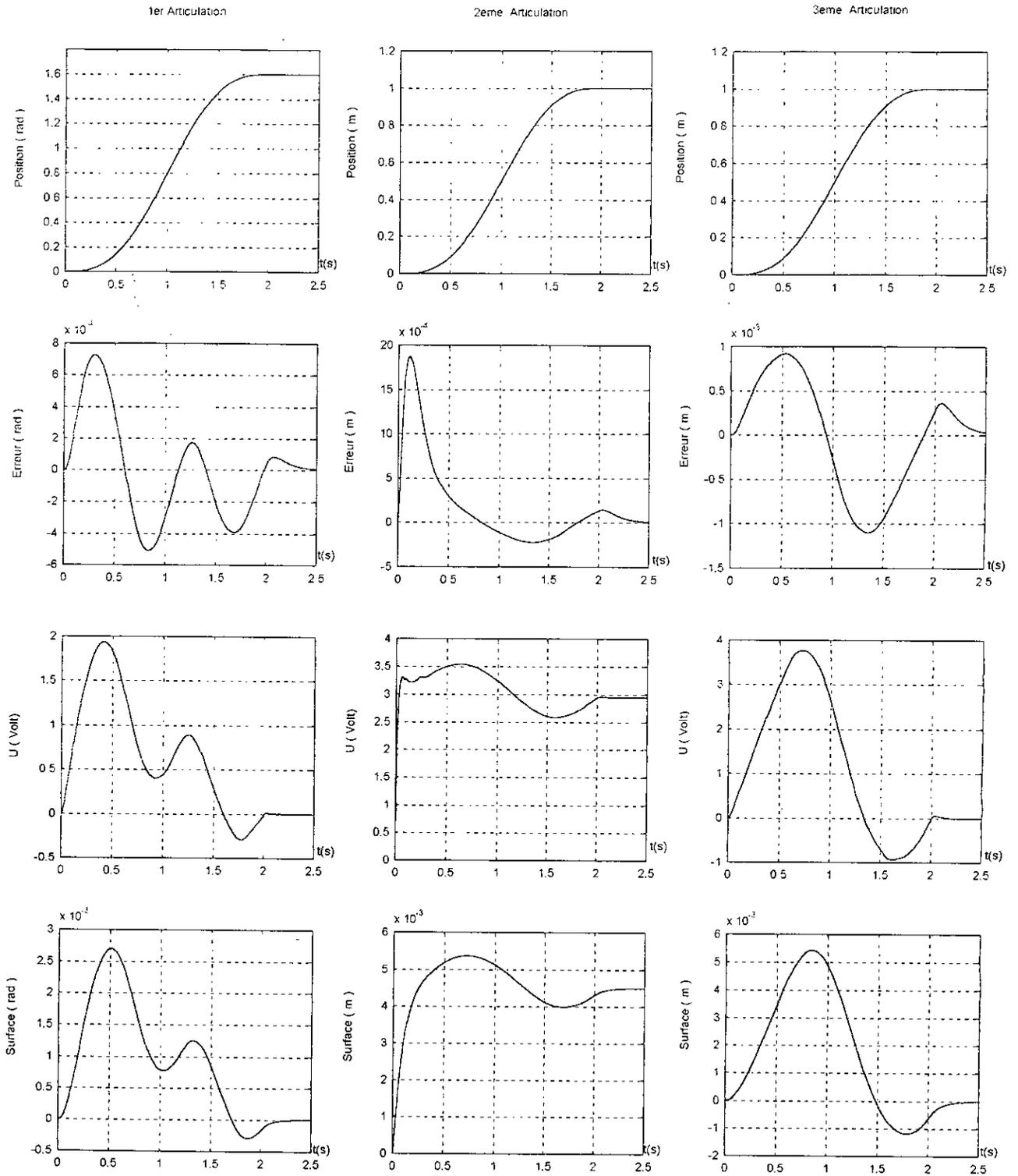


Figure (3.29) : Application de la commande par mode de glissement flou (9 classes) au robot à structure cylindrique. Poursuite à vide

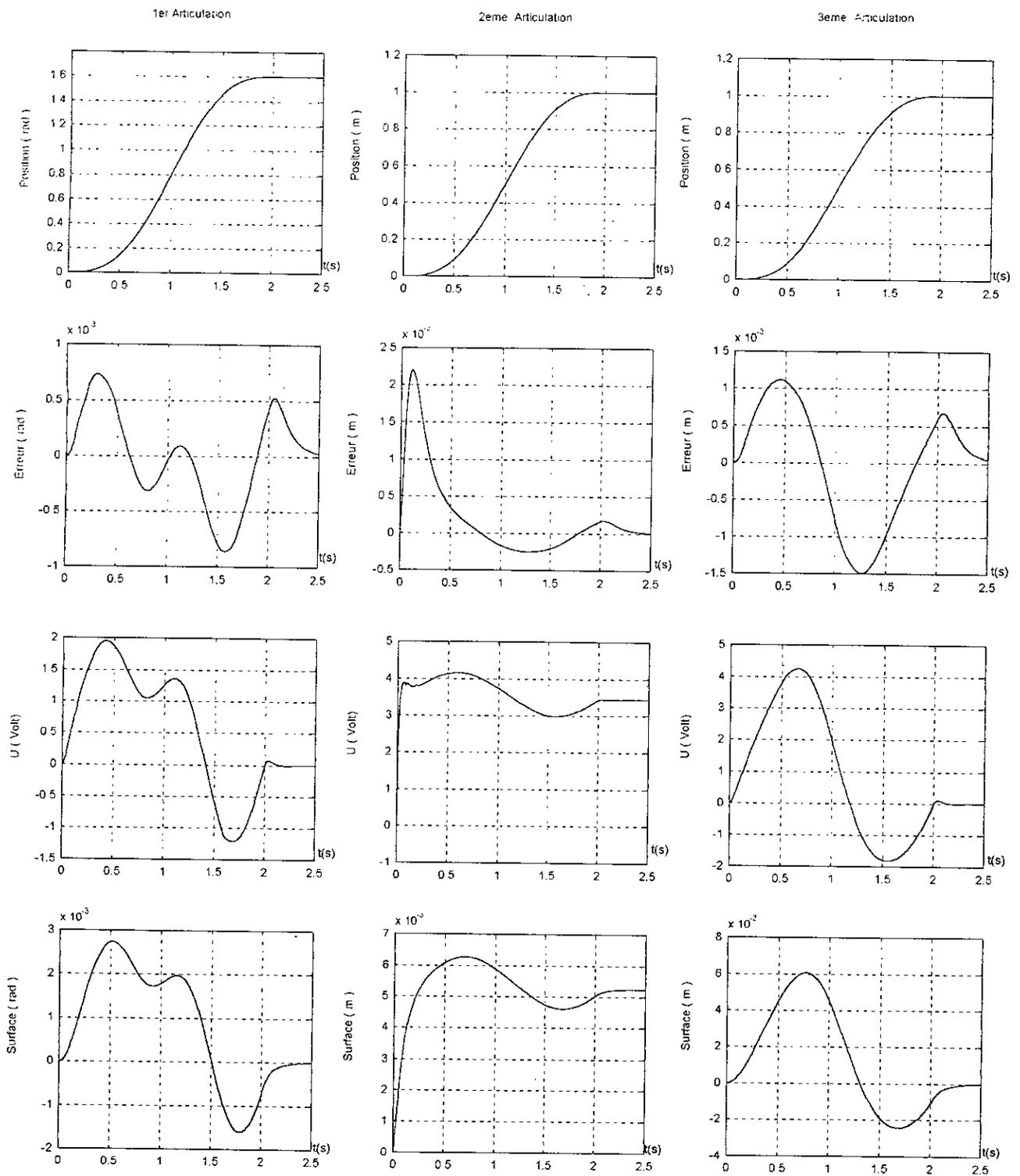


Figure (3.30) : Application de la commande par mode de glissement flou (9 classes) au robot à structure cylindrique
Poursuite avec effet de la charge $m_0 = 5\text{kg}$

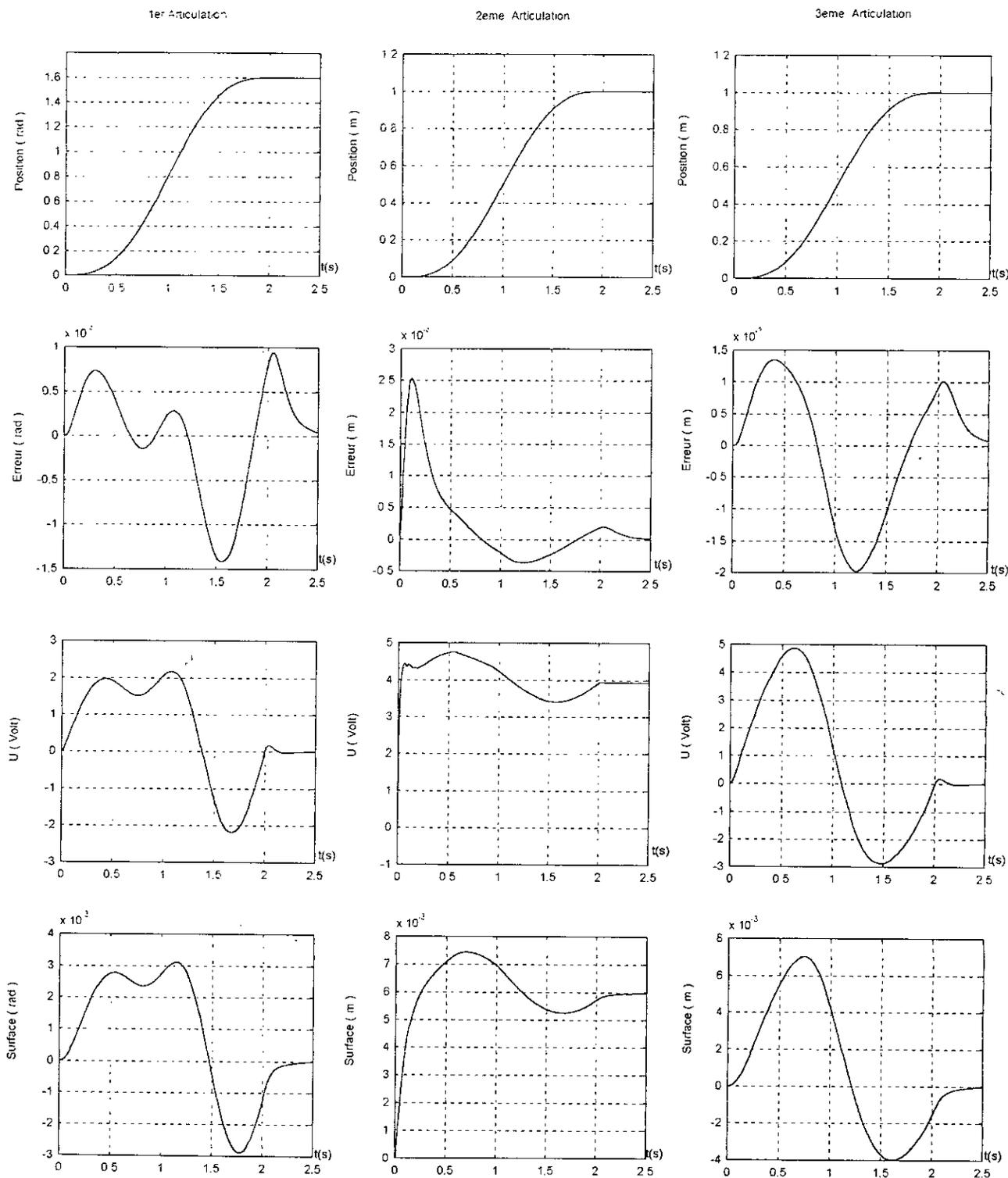


Figure (3.31) : Application de la commande par mode de glissement flou (9 classes) au robot à structure cylindrique
Poursuite avec effet de la charge $m_0 = 10\text{kg}$

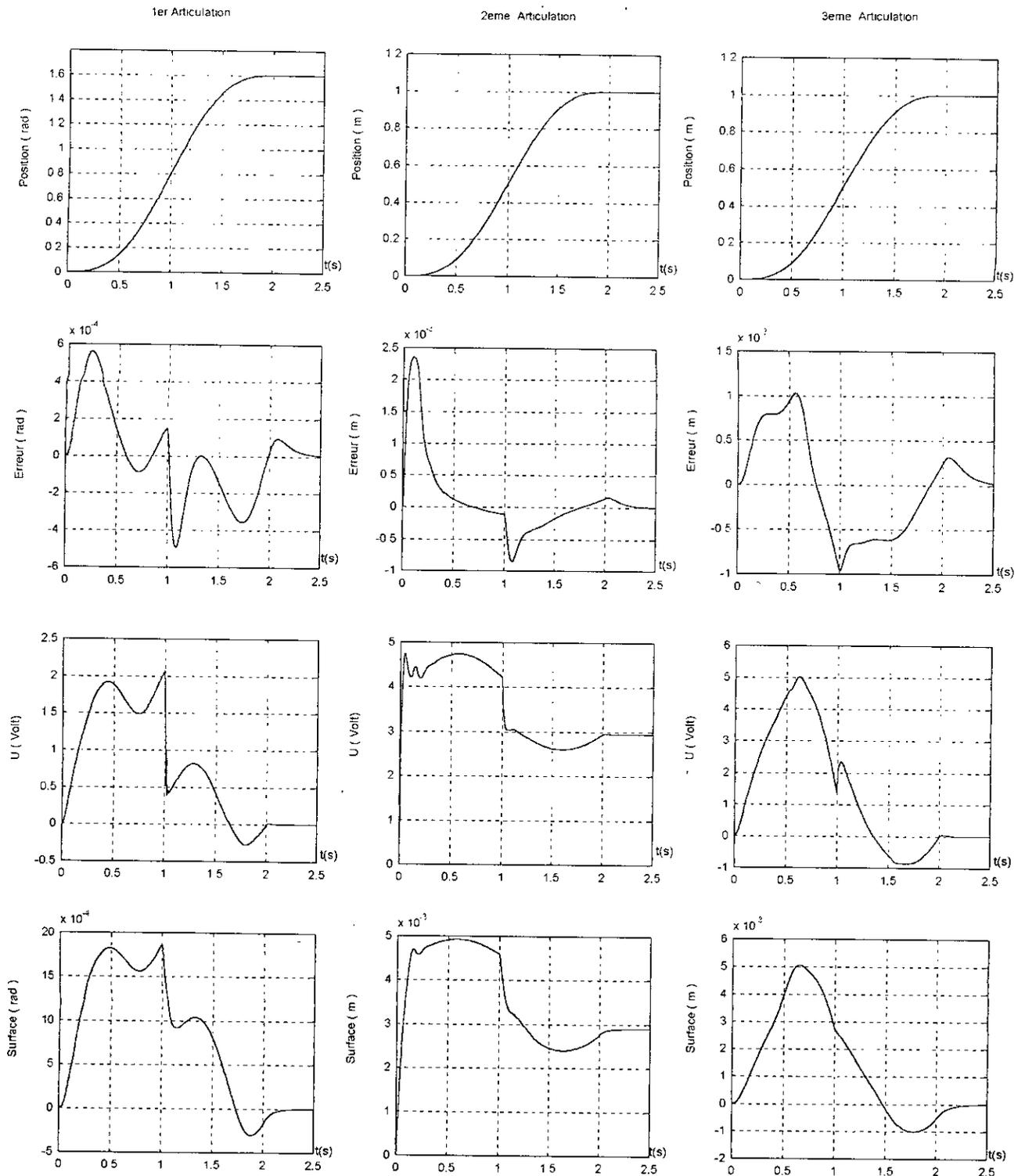


Figure (3.32) : Application de la commande par mode de glissement flou (7 classes) au robot à structure cylindrique
 Poursuite avec effet de la charge $m_0 = 10\text{kg}$ et la décharge à l'instant $t=1\text{s}$.

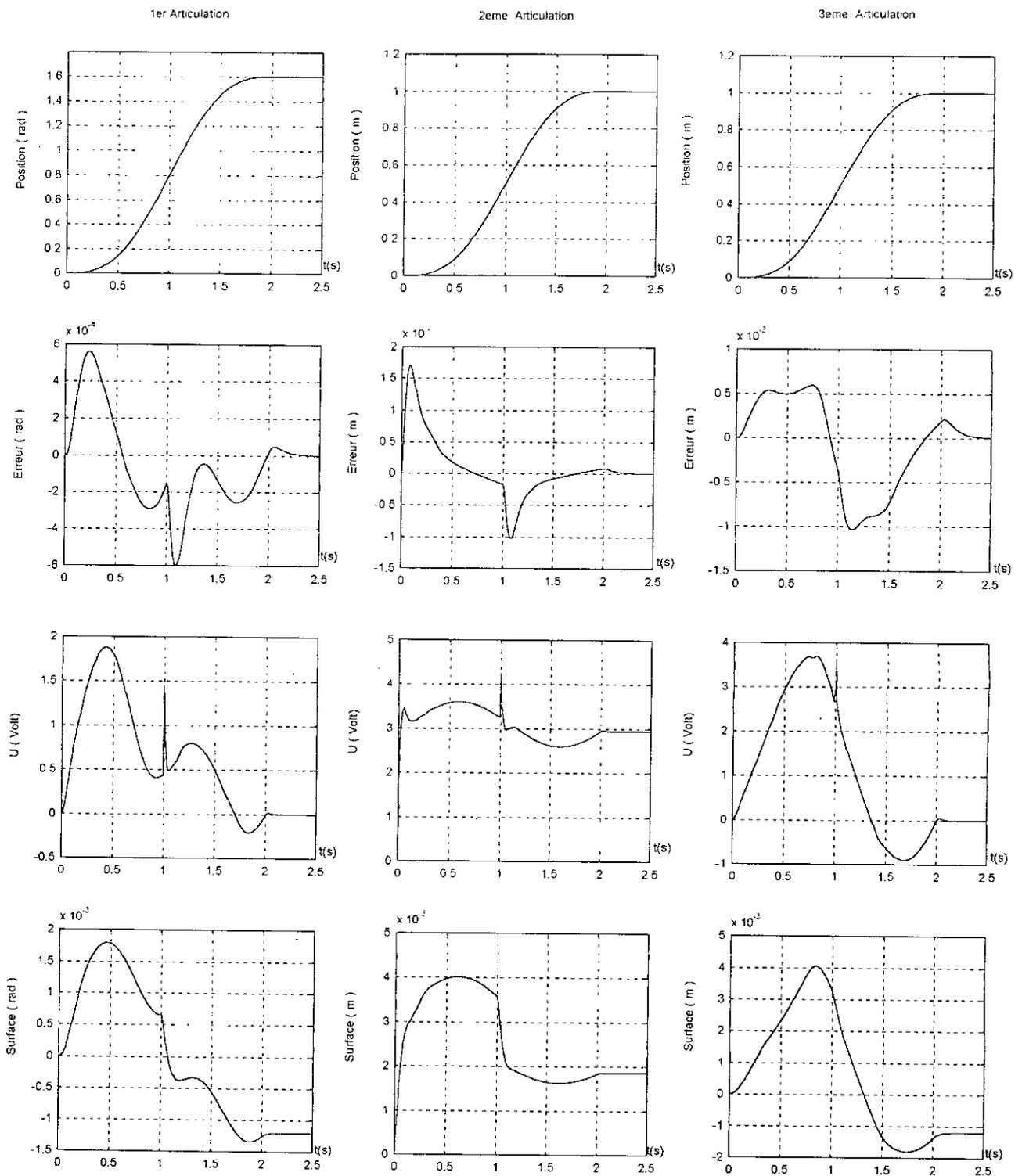


Figure (3.33) : Application de la commande par mode de glissement flou (7 classes) au robot à structure cylindrique
Poursuite avec des perturbations externes $u_{pi} = -1$ Volt à partir de l'instant $t = 1$ s.

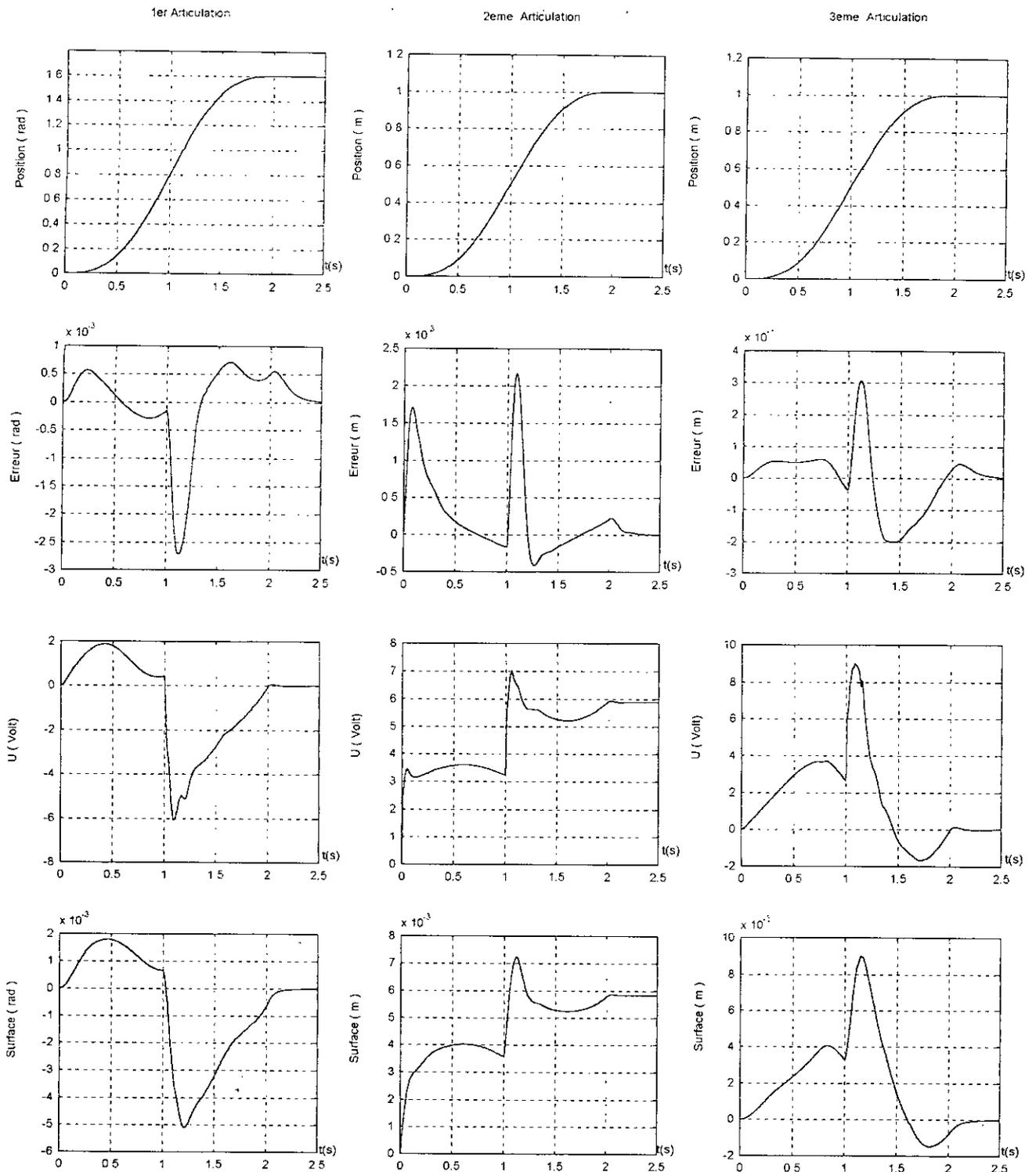


Figure (3.34) : Application de la commande par mode de glissement flou (7 classes) au robot à structure cylindrique
 Poursuite avec une variation paramétrique de 100% à partir de l'instant $t = 1s$.

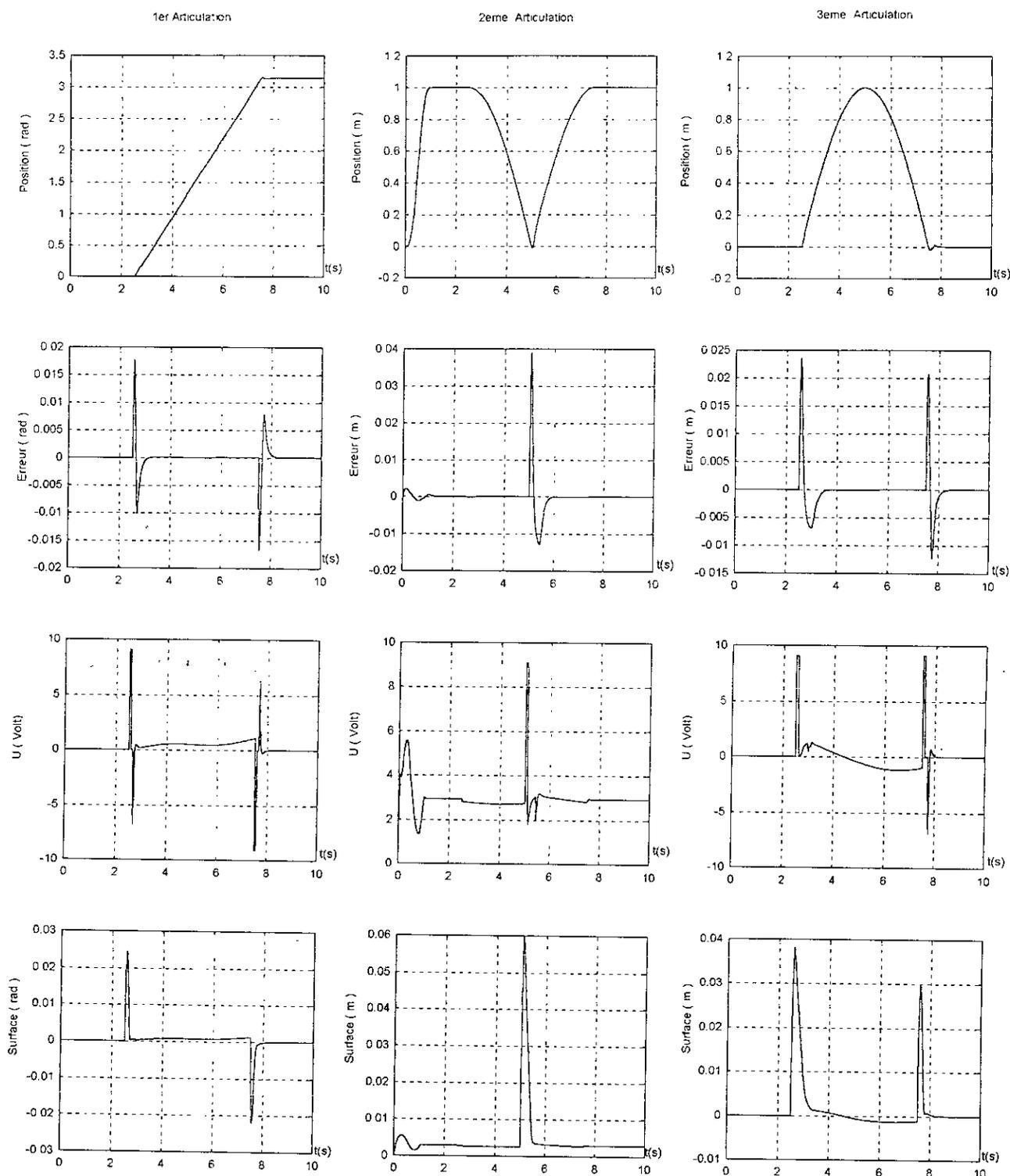
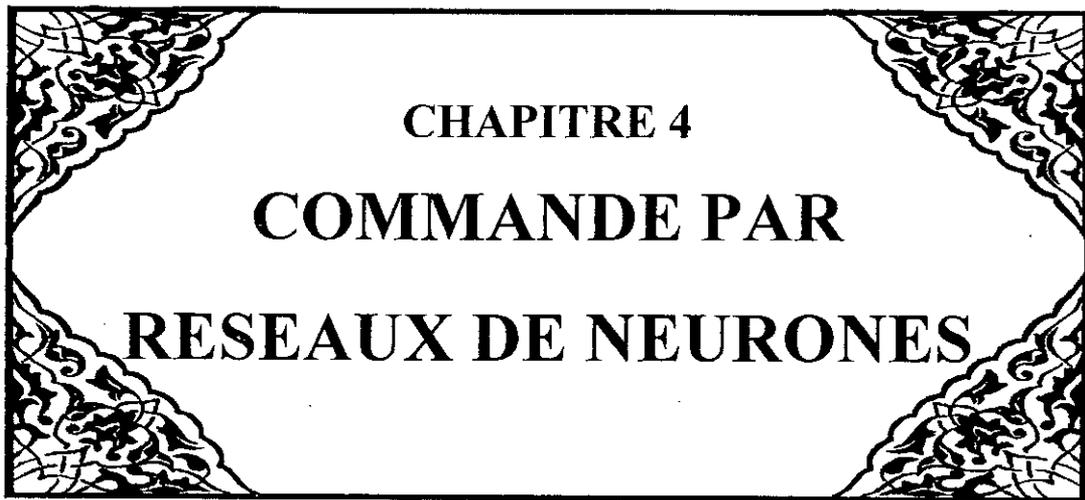


Figure (3.35) : Application de la commande par mode de glissement floue (7 classes) au robot à structure cylindrique
 Test de capacité de poursuite (fenêtre de VIVIANI)



CHAPITRE 4
COMMANDE PAR
RESEAUX DE NEURONES

4.1 Introduction :

Parmi les métaphores biologiques employées de nos jours pour la résolution de problème, les réseaux de neurones artificiels constituent certainement l'approche la plus utilisée. Ils consistent en des modèles plus ou moins inspirés du fonctionnement cérébral de l'être humain, en se basant principalement sur le concept de neurone.

4.2 Structure d'un neurone biologique :

Un neurone est une cellule particulière, comme celle représentée à la figure (1.1) possédant des extensions par lesquelles il peut distribuer des signaux (axones) ou en recevoir (dendrites) ; les échanges de signaux se font au niveau des synapses, qui relient les dendrites d'un neurone aux axones des autres cellules. L'information est transmise au moyen de signaux électriques provoqués par des échanges d'ions potassium-sodium au niveau des membranes synaptiques. La plupart des modèles ne retiennent du fonctionnement réel du neurone que les principes suivant :

- Les réseaux de neurones sont caractérisés par des interconnexions denses entre des unités de traitement simples agissant en parallèles.
- A chaque connexion est associé un poids qui détermine l'influence réciproque des 2 unités connectées.
- Les poids des connexions sont modifiables et c'est cette plasticité qui donne lieu aux facultés d'adaptation et d'apprentissage.

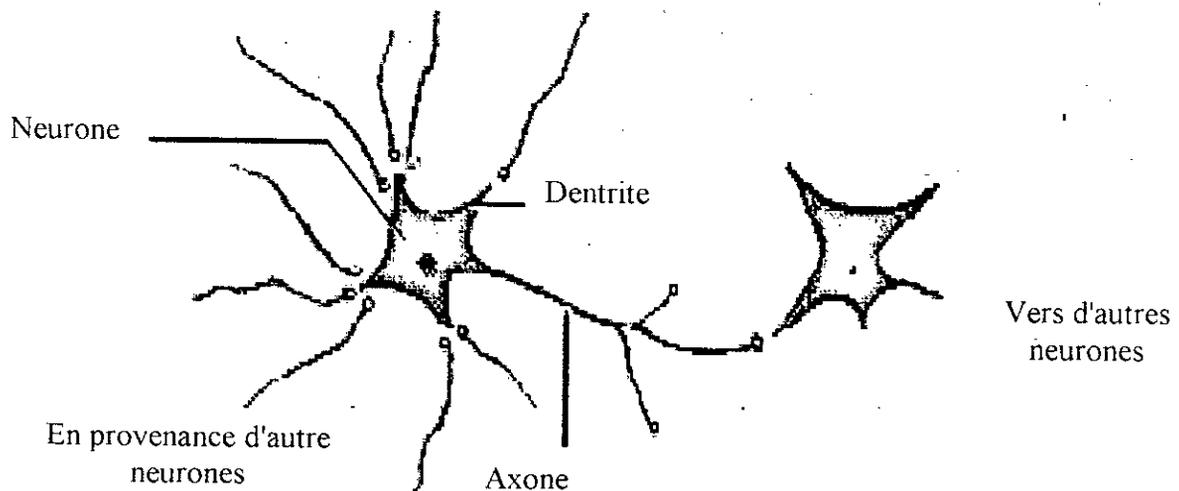


Figure 4.1 : Schéma d'un neurone biologique

4.3 Le neurone artificiel :

Comme le montre la figure 4.2, le neurone artificiel est caractérisé par :

- Ses entrées,
- Sa fonction d'entrée,
- Sa fonction de transfert,
- Sa sortie.

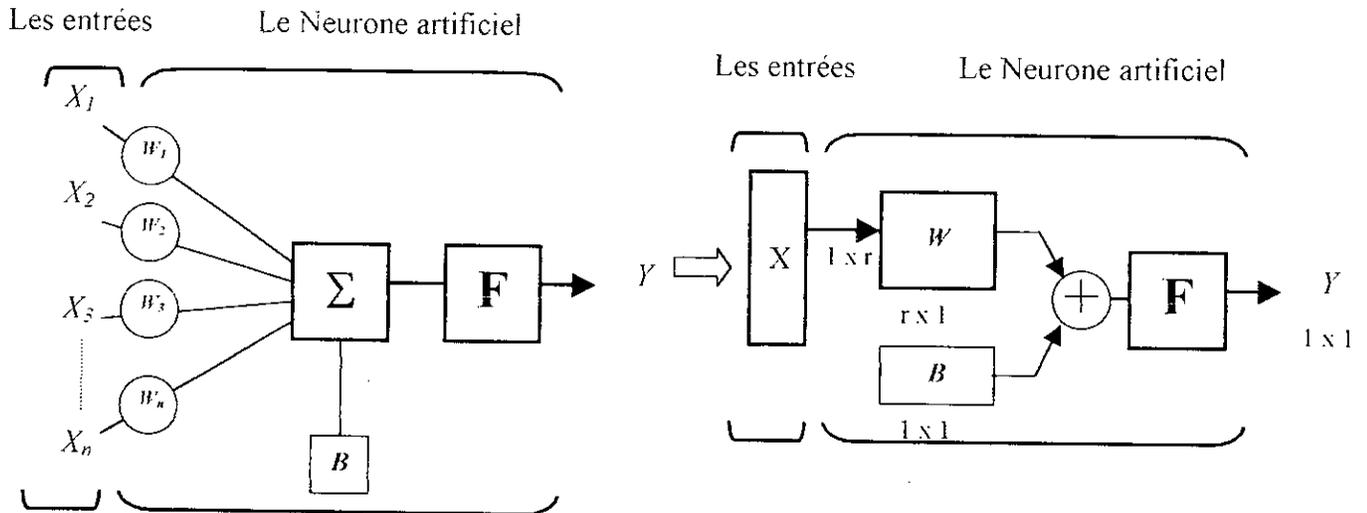


Figure 4.2 : Schéma synoptique du neurone artificiel

Le neurone effectue la somme des potentiels pondérés (matrice poids) reçus avec le biais à travers la fonction de transfert F :

$$Y = F \left(\sum_{k=1}^N W_k X_k + B \right) = F(W X + B) \quad (4.1)$$

Il existe plusieurs types de fonction de transfert, parmi les plus courantes on cite :

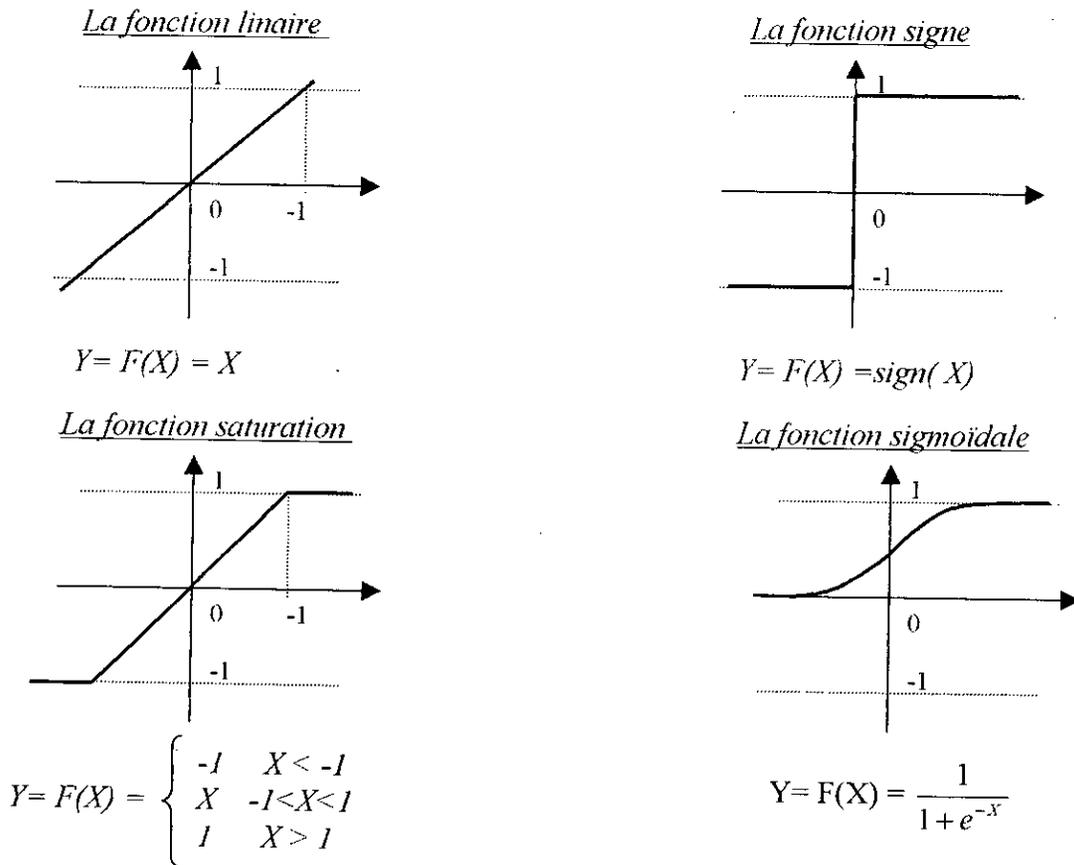


Figure 4.3 : Les différents types de fonctions

4.4 Architectures des réseaux de neurones adaptatifs :

Le réseau de neurones adaptatif est une structure dont le comportement dépend de la valeur de certains paramètres variables. Plus expressément un réseau de neurones adaptatif est composé d'un nombre de nœuds connecté par des liens, chaque nœud réalise une fonction et chaque lien spécifie la direction de l'information d'un nœud vers un autre.

Généralement la fonction du nœud est une fonction paramétrique dont les paramètres sont ajustables. En changeant ces paramètres on change la fonction du nœud ainsi que le comportement général du réseau adaptatif.

La figure 4.4 est une représentation en couches d'une la structure typique d'un réseau de neurones adaptatif.

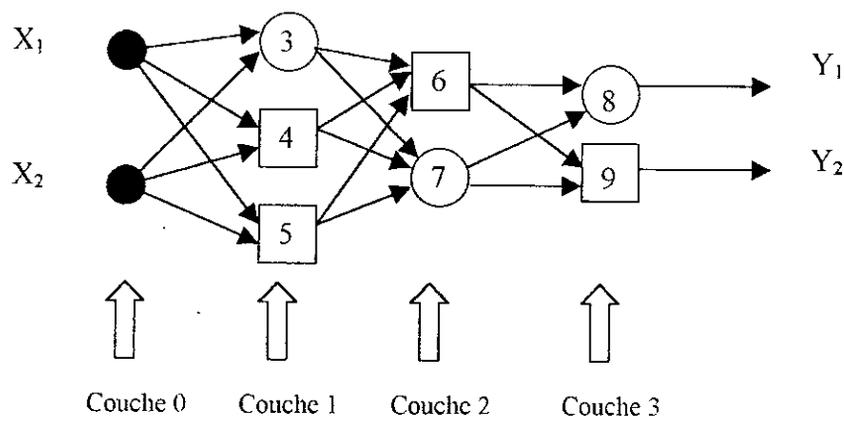


Figure 4.4 : Structure d'un réseau de neurones adaptatif

Chaque nœud a ses propres paramètres locaux, leur union définit les paramètres du réseau global. On représente par un carré le nœud dont les paramètres sont variables et par un cercle celui dont les paramètres sont fixes. On classe généralement les réseaux adaptatifs en deux catégories, selon le sens des liens entre les différents nœuds : *feedforward* et *récurrent*. La figure 4.4 est un réseau *feedforward*, la propagation de l'information se fait dans un seul sens (de gauche vers la droite), s'il existe un lien qui transmet l'information dans le sens inverse en formant un chemin circulaire dans le réseau, alors le réseau est appelé *récurrent*. La figure 4.5 en représente un exemple.

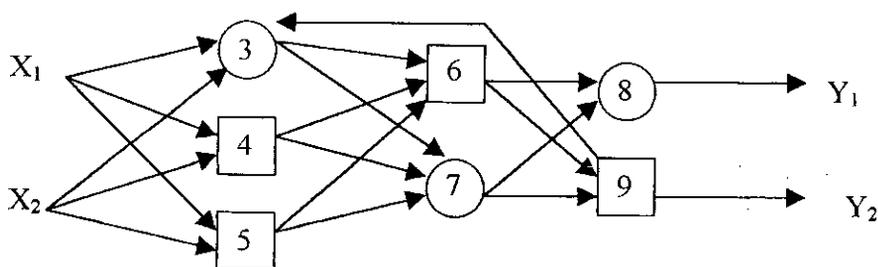


Figure 4.5 : réseau adaptatif récurrent

Une autre représentation du réseau feedforward existe c'est la représentation par ordre topologique, elle ordonne en séquences 1,2,..., de telle sorte qu'il n'y ait pas de liens du nœud i vers le nœud j pour $i \geq j$, (figure 4.6). Cette représentation facilite la formulation de l'algorithme d'apprentissage.

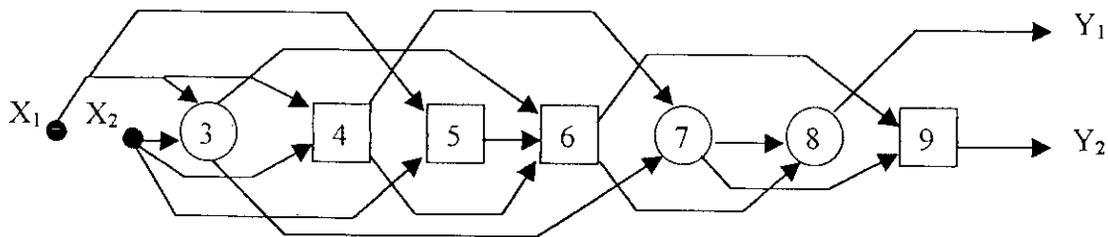


Figure 4.6 : représentation par ordre topologique

4.5 Les différents types de réseaux de neurones adaptatifs :

Dans ce qui suit nous décrivons les architectures des réseaux adaptatifs qui englobent tous les types des réseaux de neurones ayant une capacité d'apprentissage supervisé. En particulier, nous présenterons trois types de réseaux permis les plus utilisés dans la littérature des réseaux de neurones : le Perceptron, Back-propagation neural network (BPNN) et Radial basis function network (RBFN).

4.5.1 Perceptron :

ROSENBLATT [34] a créé plusieurs types de Perceptron, le plus simple est composé d'une seule couche dont les paramètres (poids et biais) doivent être ajustés pour approcher la fonction voulue. La technique d'apprentissage utilisée dans ce cas est appelée Règle d'apprentissage du Perceptron (Perceptron learning rule).

Le Perceptron est spécialement appliqué aux problèmes simples de classification de modèles et grâce à sa simplicité, il sert de base d'étude pour d'autres réseaux complexes.

Le neurone utilisé dans le Perceptron se schématise comme suit :

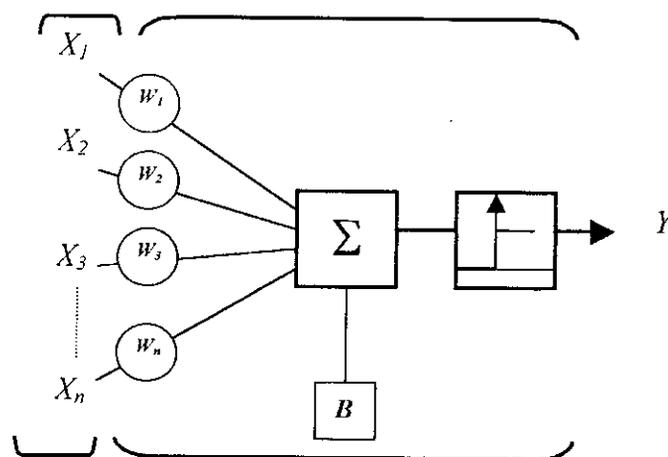


Figure 4.7 Schéma du neurone utilisé en Perceptron

$$Y = Ech \left(\sum_{k=1}^N W_k X_k + B \right) \approx Ech(WX + B) \quad (4.2)$$

Avec Y est la sortie du neurone qui est l'échelon de la somme des entrées pondérées avec le biais B , qui sera égale à 1 si $WX+B > 0$, et nul dans le cas contraire.

Un réseau Perceptron est composé d'une seule couche, à n neurones reliés aux r entrées, comme le montre la figure ci-dessous.

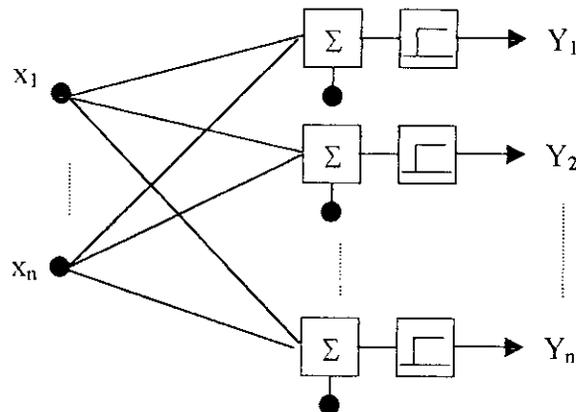


Figure 4.8 : Structure d'un réseau Perceptron

Malgré sa simplicité, le réseau Perceptron présente certains désavantages, les sorties du réseau ne peuvent prendre que 2 valeurs (0 ou 1) en raison de la fonction de transfert utilisée (échelon), l'algorithme d'apprentissage ne peut être appliqué qu'aux réseaux à une seule couche.

4.5.2 Back Propagation neural network :

Le Back Propagation a été élaboré en généralisant l'algorithme d'apprentissage de WINDROW-HOFF aux réseaux multicouches avec des fonctions de transfert différentiables et non linéaires, l'apprentissage se réalise par la création d'un fichier E/S puis par l'application de l'algorithme d'apprentissage qui aboutira à la détermination des poids et des biais nécessaires à l'approche d'une fonction ou la classification des vecteurs d'entrées. Un tel réseau et avec les poids et les biais appropriés, une couche de sigmoïdes et une autre couche linéaire, peut approximer n'importe quelle fonction avec un nombre fini de discontinuités.

Un neurone élémentaire du Back propagation est représenté par la figure ci-dessous

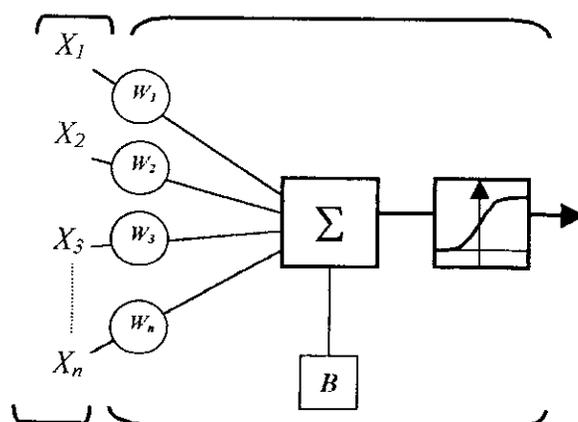


Figure 4.9 : Neurone utilisé en Back Propagation

La somme des entrées pondérées avec le biais, est l'entrée d'une fonction de transfert différentiable.

Les fonctions de transfert les plus utilisées sont :

$$\text{Fonction sigmoïdale} \quad f(x) = \frac{1}{1 + e^{-x}} \quad (4.3)$$

$$\text{Fonction tangente hyperbolique} \quad f(x) = \tanh(x/2) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (4.4)$$

$$\text{Fonction linéaire} \quad f(x) = x \quad (4.5)$$

A la différence des réseaux Perceptron, les réseaux Back propagation peuvent être composés de plusieurs couches. Si la fonction de transfert de la dernière couche est une sigmoïde, les sorties du réseau seront toujours limitées entre 0 et 1. C'est la raison pour laquelle on impose à la dernière couche la fonction linéaire. Le réseau décrit ci dessous, représente un réseau du type Back Propagation à 3 couches, tel que, la fonction de transfert de la couche de sortie soit une fonction linéaire et celle de la couche cachée une fonction sigmoïdale.

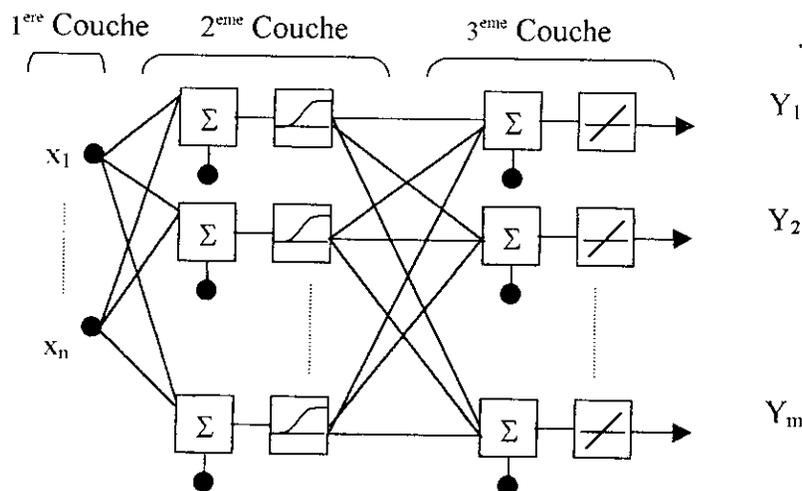


Figure 4.10 : Structure d'un réseau Back Propagation

La sortie des neurones de la 2^{ème} couche est déterminée par les relations suivantes :

$$\begin{cases} \bar{x}_{i+1,j} = \sum_{k=1}^n w_{i,k} \cdot x_{i,k} + I_{i+1,j} \\ x_{i+1,j} = f(\bar{x}_{i+1,j}) = \frac{1}{1 + e^{-\bar{x}_{i+1,j}}} \end{cases} \quad (4.6)$$

L'algorithme d'apprentissage courant est l'algorithme du gradient, il utilise une formule récurrente basée sur la valeur du gradient de l'erreur d'apprentissage. Quoique cette méthode soit très répandue, elle ne converge pas rapidement (minimas locaux), à cause du gain d'apprentissage qui doit être pris petit pour assurer la stabilité de l'apprentissage. Pour remédier à ce problème, on fait recours aux algorithmes conjugués, tels que : les algorithmes quasi Newtoniens, Levenberg-Marquardt, ...

Le choix du nombre de neurones de la couche cachée, a une grande influence sur les performances du réseau, pour assurer des performances appréciables on fixe en générale le nombre de neurones entre 3 et 5.

4.5.3 Réseau de neurones à fonction de base Radiale (Radial Basis Network) :

Les RBN exigent plus de neurones que les réseaux Back Propagation standards, mais leur construction est plus rapide. Ils sont encore plus performants quand on dispose de plusieurs fichiers E/S.

Un neurone élémentaire du RBN se présente comme suit :

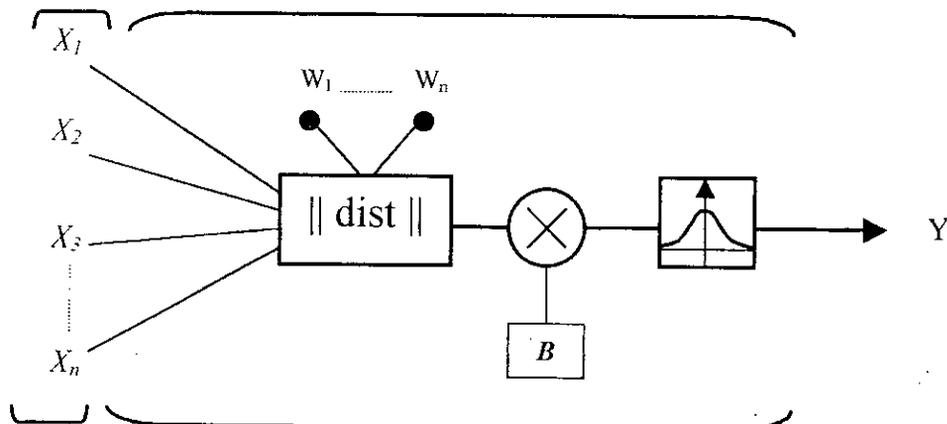


Figure 4.11 : Neurone utilisé en RBN

On peut noter que le neurone utilisé en RBN est différent de ceux utilisés auparavant, sa fonction de transfert est gaussienne, dont l'entrée est la distance entre les poids et les entrées X multipliées par le biais B . L'expression analytique de cette fonction est :

$$f(x) = e^{-x^2} \quad (4.7)$$

L'architecture du RBN est identique à celle des réseaux Back Propagation, sauf que les neurones de la couche cachée du réseau sont remplacés par ceux du RBN.

La sortie d'un réseau RBN est calculée par la relation suivante :

$$f(\vec{u}) = \frac{\sum_{j=1}^L w_j \cdot \rho_j(\vec{u})}{\sum_{j=1}^L \rho_j(\vec{u})} \quad (4.8)$$

Ou :

- w_j : sont les poids associés à chaque sortie d'unité de la couche cachée,
- $\sum_{j=1}^L \rho_j(\bar{u})$: est le terme de normalisation.

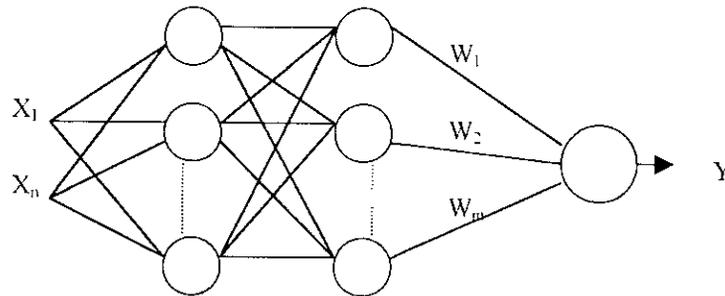


Figure 4.12 : Architecture d'un réseau de neurones à fonction de base Radiale

4.6 Les algorithmes d'apprentissage pour les réseaux Back Propagation :

Il existe plusieurs algorithmes de Back propagation, mais le plus simple consiste à faire varier le poids et les biais dans le sens où la fonction de performance diminue le plus rapidement, ou bien dans le sens négatif du gradient de la fonction de performance. Une itération de cet algorithme peut être écrite comme suit :

$$x_{k+1} = x_k - \eta_k g_k \tag{4.9}$$

avec :

- x_k : valeur courante des poids ou des biais.
- g_k : gradient de l'erreur (fonction de performance).
- η_k : gain d'apprentissage.

Supposant qu'on dispose d'un réseau *feed-forward* adaptatif comportant L couches et chaque couche k a $N(k)$ nœuds. Donc les sorties et les fonctions du nœud $i(i=1,2,\dots,N(l))$ de la couche k peuvent être représentés par $x_{l,i}$ et $f_{l,i}$ respectivement, comme le montre la figure 4.12, l'expression générale de la fonction $f_{l,i}$ est :

$$x_{l,i} = f_{l,i}(x_{l-1,1}, \dots, x_{l-1,N(l-1)}, \alpha, \beta, \gamma, \dots) \tag{4.10}$$

Avec α, β, γ des paramètres internes du nœud.

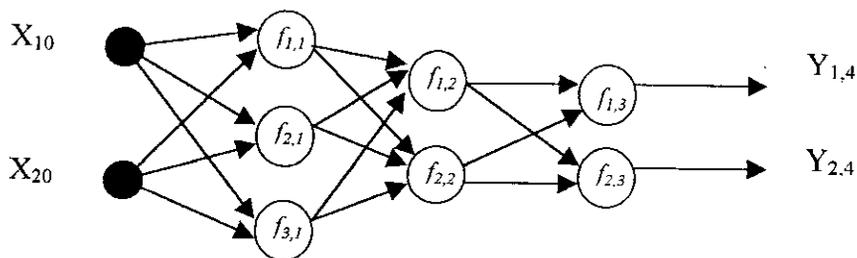


Figure 4.13 : Représentation d'un réseau *feed-forward*

Pour un vecteur à P entrées de données, on définit une erreur de mesure (Fonction de performance) pour la $p^{\text{ème}}$ entrée ($1 \leq p \leq P$) comme étant la somme de toutes les erreurs au carré :

$$E_p = \sum_{k=1}^{N(l)} (d_k - x_{l,k})^2 \quad (4.11)$$

Avec :

- d_k : la $k^{\text{ème}}$ composante de la $p^{\text{ème}}$ sortie désirée.
- $x_{l,k}$: la $k^{\text{ème}}$ composante du vecteur actuel de sortie du réseau.

Quant E_p est nulle, le réseau est capable de reproduire exactement la $p^{\text{ème}}$ sortie désirée. Notre but est de minimiser une erreur globale définie comme suit :

$$E = \sum_{p=1}^P E_p \quad (4.12)$$

Remarque : La définition de E_p n'est pas unique, d'autres définitions sont possibles pour des cas spécifiquement ou des applications.

• L'algorithme du gradient :

On définit le signal d'erreur $\varepsilon_{l,i}$ comme étant la dérivée ordonnée de l'erreur mesurée E_p du nœud i de la couche l :

$$\varepsilon_{l,i} = \frac{\partial^+ E_p}{\partial x_{l,i}} \quad (4.13)$$

Pour les nœuds de la dernière couche le signal d'erreur est :

$$\varepsilon_{L,i} = \frac{\partial^+ E_p}{\partial x_{L,i}} = \frac{\partial E_p}{\partial x_{L,i}} = 2(d_i - x_{L,i}) \quad (4.14)$$

Pour les nœuds internes on a :

$$\varepsilon_{l,i} = \frac{\partial^+ E_p}{\partial x_{l,i}} = \sum_{m=1}^{N(l+1)} \frac{\partial^+ E_p}{\partial x_{l+1,m}} \times \frac{\partial f_{l+1,m}}{\partial x_{l,i}} = \sum_{m=1}^{N(l+1)} \varepsilon_{l+1,m} \frac{\partial f_{l+1,m}}{\partial x_{l,i}} \quad (4.15)$$

Si α est un paramètre dans le $i^{\text{ème}}$ nœud de la couche l , on a :

$$\frac{\partial^+ E_p}{\partial \alpha} = \frac{\partial^+ E_p}{\partial x} \frac{\partial f_{l,i}}{\partial \alpha} = \varepsilon_{l,i} \frac{\partial f_{l,i}}{\partial \alpha} \quad (4.16)$$

S'il intervient dans plusieurs nœud l'équation devient :

$$\frac{\partial^+ E_p}{\partial \alpha} = \sum_{x^* \in S} \frac{\partial^+ E_p}{\partial x^*} \frac{\partial f^*}{\partial \alpha} \quad (4.17)$$

Avec :

- S est l'ensemble des nœuds ou α influe directement,
- f^* est la fonction du nœud pour une valeur x^* calculée.

La dérivée ordonnée de l'erreur de mesure globale par rapport à α est :

$$\frac{\partial^+ E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial^+ E_p}{\partial \alpha} \quad (4.18)$$

L'ajustement du paramètre α se fait par la relation :

$$\alpha(t+1) = \alpha(t) - \eta \frac{\partial^+ E}{\partial \alpha} \quad (4.19)$$

Avec $\frac{\partial^+ E}{\partial \alpha} = g_k$: gradient de l'erreur ce qui nous renvoie à l'équation (4.9)

Tel que η est le pas d'apprentissage qui représente le défaut de l'algorithme du gradient, lorsque η est assez grand, il déstabilise la convergence de l'algorithme, alors que la convergence de l'algorithme devient lente quand η est petit.

• Méthode de Newton :

La méthode de Newton est une alternative qui peut être conjugué avec l'algorithme du gradient pour une optimisation plus rapide, l'étape de base de cette méthode est :

$$x_{k+1} = x_k - A^{-1} g_k \quad (4.20)$$

Tel que A est la matrice hessienne qui est la deuxième dérivée de l'index de performance des valeurs courantes des poids et des biais. L'algorithme de Newton converge plus rapidement que la méthode du gradient, malheureusement, il est difficile de calculer la matrice hessienne qui prend, à chaque étape, un temps de calcul considérable.

Pour remédier à ce problème, plusieurs méthodes ont été développées basé sur l'algorithme de newton en approximant la matrice hessienne. Parmi ces méthodes il y a celle de Levenberg Marquardt.

• Algorithme de Levenberg-Marquardt

L'algorithme de Levenberg Marquardt a été conçu pour approcher la vitesse du second ordre, sans avoir à calculer la matrice hessienne, puisqu'on l'approxime par :

$$H = J^T J \quad (4.21)$$

Et le gradient peut être calculé comme suit :

$$g = J^T e \quad (4.22)$$

ou J est la matrice jacobienne contenant les premières dérivées des erreurs du réseau, respectivement des poids et des biais et e le vecteur des erreurs.

La matrice J peut être calculée par une technique standard de Back propagation [15] et qui est bien moins complexe à déterminer que la matrice hessienne.

L'algorithme de newton approximé devient alors :

$$x_{k+1} = x_k - [J^T J]^{-1} J^T e \quad (4.23)$$

Pour conjuguer l'algorithme du gradient avec celui de newton approximés, l'algorithme de Levenberg Marquardt se présente comme suit :

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (4.24)$$

Dans le cas où μ est pris très petit devant $J^T J$ on aura l'algorithme de newton approximé, si μ est assez grand ça sera l'algorithme du gradient avec un petit pas d'apprentissage. Comme la méthode de newton est plus rapide et plus précise pour des petites erreurs, l'utilisation de l'algorithme de Levenberg-Marquardt a pour but de décaler vers la méthode de newton aussi rapidement que possible. Alors μ sera diminué après chaque étape réussie et sera augmenté seulement quand la fonction de performance augmente. De cette façon la fonction de performance sera toujours réduite à chaque étape de l'algorithme.

4.7 Application de la commande au Robot PUMA 560 :

Dans ce qui suit, nous appliquerons la commande par réseaux de neurones au Robot PUMA 560. Nous constituerons d'abord un fichier d'E/S pour chaque articulation et ce à partir des résultats obtenus lors des simulations du système de commande à mode glissant flou et spécialement la simulation avec une charge de 10 kg, pour couvrir le maximum de variation de la commande. Le fichier d'E/S est constitué de 3 entrées et 1 sortie, puisque la surface de chaque articulation est une combinaison linéaire entre l'erreur, sa dérivée et son intégrale.

Le réseau de neurones utilisés, est un réseau du type Back Propagation, dont la couche cachée est composée de 4 neurones. La fonction de transfert de la couche cachée et celle de la couche de sortie sont respectivement : sigmoïdale et linéaire.

L'algorithme d'apprentissage choisi est celui de Levenberg-Marquardt pour sa rapidité de convergence. L'initialisation des poids et des biais est aléatoire et l'ordre de variation de μ est de 10, c'est à dire que pour diminuer ou augmenter μ , on le multiplie ou on le divise par 10.

Le réseau de neurone appliqué est de la forme :

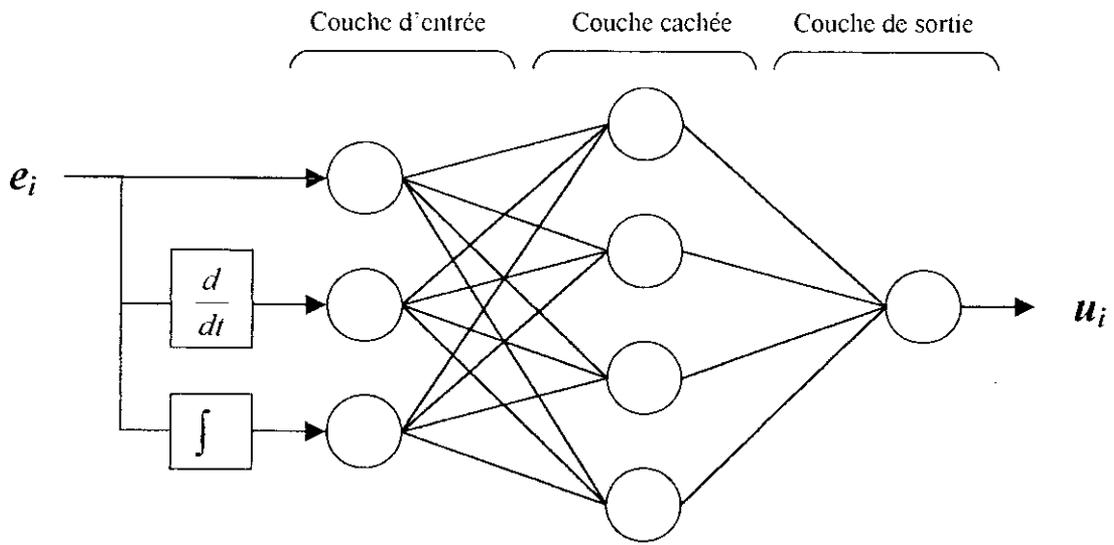


Figure 4.14 : Le réseau de neurones appliqué au Robot PUMA560

La structure de commande globale est de la forme :

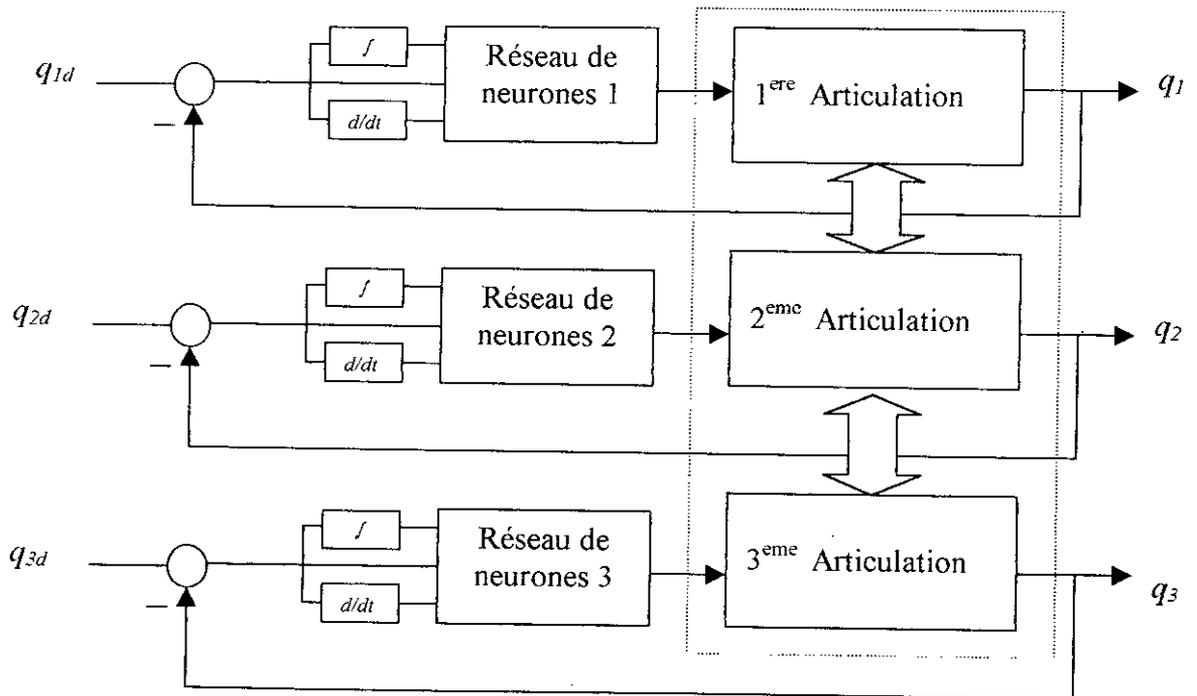


Figure 4.15 : structure de la commande par réseaux de neurones

4.8 Simulation et interprétation :

Les comportements du robot sont représentés par les figures (4.16, 4.17 et 4.18), qui correspondent respectivement à l'essai à vide et aux essais en charge (5 kg puis 10kg).

Avec l'essai à vide le système a réagi avec une très bonne poursuite puisque l'erreur ne dépasse pas 10^{-2} degrés en régime dynamique et une erreur nulle en régime statique. Les commandes sont lisses et parfaitement stables. Il en est de même pour les essais en charge ou le système garde des performances très satisfaisantes, avec notamment un pic de démarrage qui apparaît à la 3^{ème} commande.

Nous comparons dans le tableau ci dessous, les résultats obtenus avec ceux trouvés par la commande floue classique et la commande par mode glissant flou, en nous basant sur le critère employé auparavant et ceux pour les essais à vide et en charge :

Essais effectués		Flou	Mode glissant flou	Réseaux de neurones
A vide	e ₁	$6.905 \cdot 10^{-7}$	$1.17 \cdot 10^{-7}$	$1.25 \cdot 10^{-7}$
	e ₂	$2.164 \cdot 10^{-6}$	$2.18 \cdot 10^{-7}$	$2.27 \cdot 10^{-7}$
	e ₃	$8.449 \cdot 10^{-6}$	$7.05 \cdot 10^{-7}$	$9.94 \cdot 10^{-7}$
	Total	$1.13 \cdot 10^{-5}$	$1.04 \cdot 10^{-6}$	$1.346 \cdot 10^{-6}$
En charge (10kg)	e ₁	$1.569 \cdot 10^{-6}$	$3.312 \cdot 10^{-7}$	$3.46 \cdot 10^{-7}$
	e ₂	$5.489 \cdot 10^{-6}$	$1.313 \cdot 10^{-7}$	$1.409 \cdot 10^{-7}$
	e ₃	$1.406 \cdot 10^{-4}$	$1.555 \cdot 10^{-5}$	$1.581 \cdot 10^{-5}$
	Total	$1.475 \cdot 10^{-4}$	$1.601 \cdot 10^{-5}$	$1.62 \cdot 10^{-5}$

Avec e_i : l'erreur de l'énergie de la $i^{\text{ème}}$ articulation

On voit que la commande par réseaux de neurones donne de meilleurs résultats que la commande floue et approche remarquablement la commande par mode glissant floue.

- **Test de robustesse :**

On reprend les mêmes tests de robustesse utilisés auparavant et la aussi la décharge provoque de fortes variations des commandes mais qui restent stable (figure 4.19).

La figure (4.20) correspond au test de rupture de la 2^{ème} commande à l'instant 0.75 s et qui montre que les articulations 1 et 3 restent toujours stables, ce qui traduit que le système de commande assure une indépendance entre les articulations.

4.9 Conclusion :

Dans ce chapitre nous avons présenté la commande par réseaux de neurones avec les différentes architectures et algorithmes d'apprentissage. Cette commande présente des avantages comparée à d'autres types de commande par sa haute aptitude à identifier des systèmes très complexe et sa simplicité de conception.

Les paramètres du réseau de neurone que nous avons appliqués au robot PUMA 560 ont été identifiés par l'algorithme de Levenberg-Marquardt que nous avons choisi pour sa rapidité de convergence. Les simulations montre que la commande par réseaux de neurones donne des résultats très satisfaisants en approchant remarquablement les performances de la commande par mode glissant flou, tout en ayant une structure plus simple.

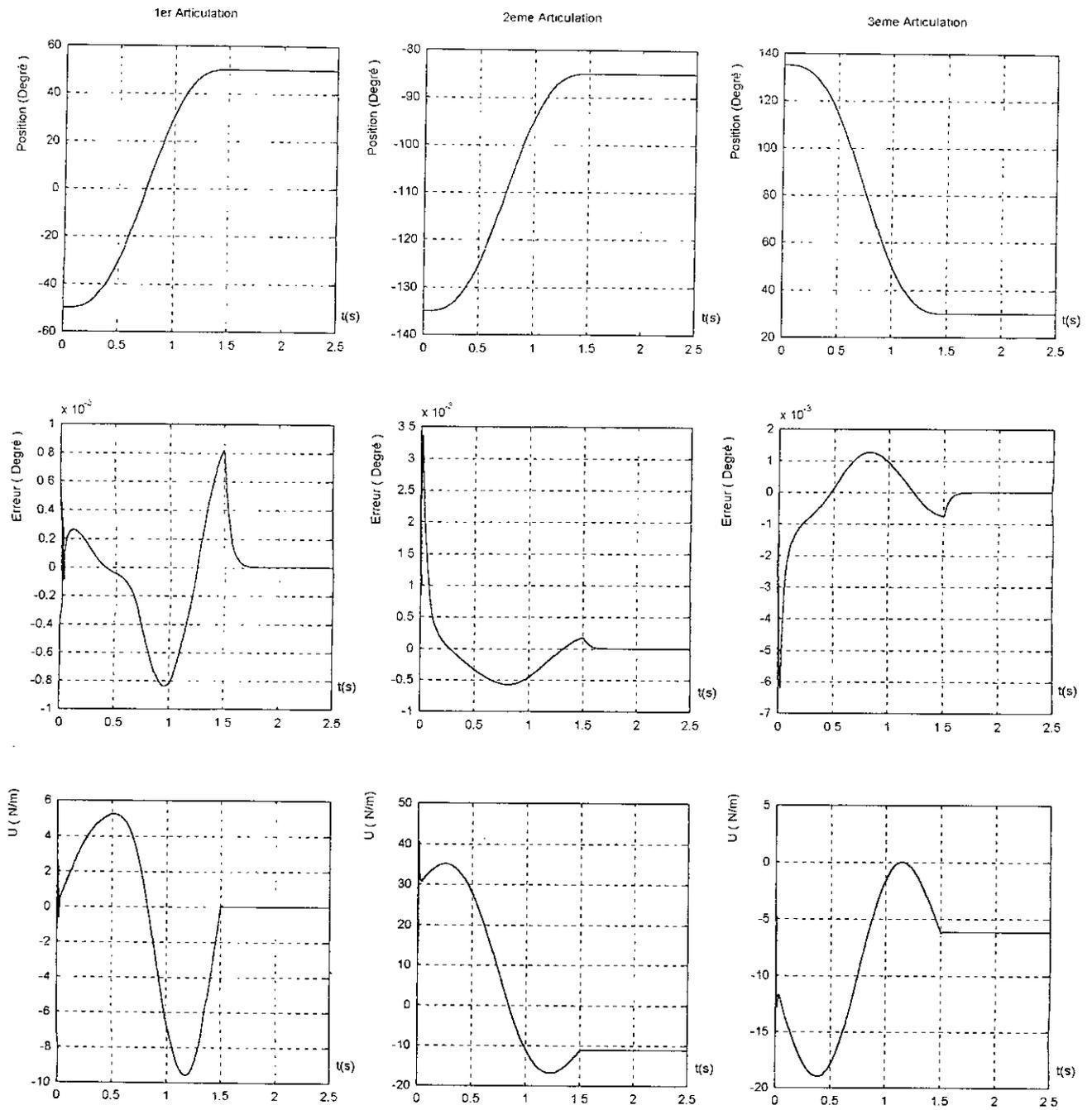


Figure (4.16) : Application de la commande par réseau de neurones au PUMA 560. Poursuite à vide

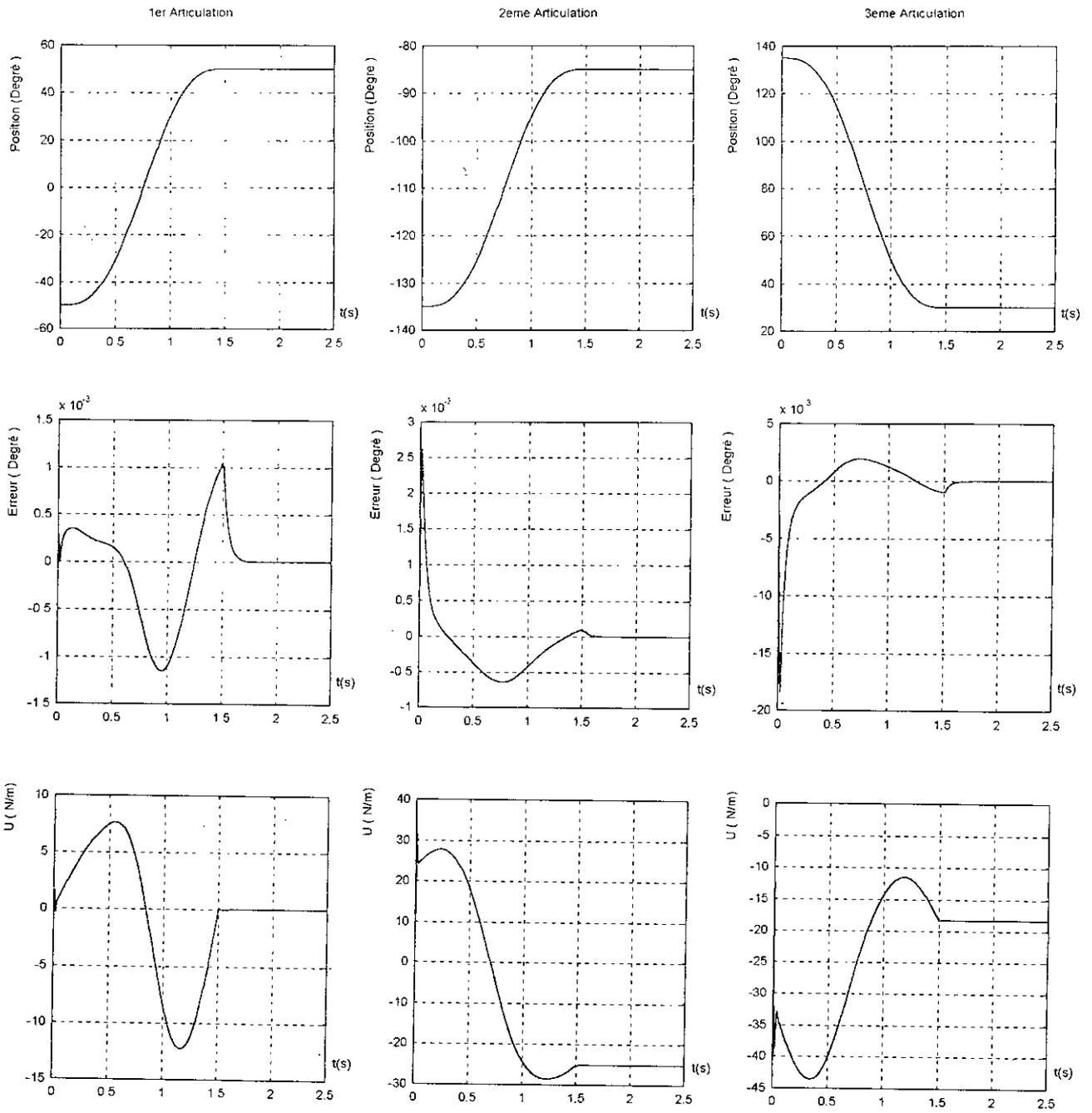


Figure (4.17) : Application de la commande par réseau de neurones au PUMA 560. Poursuite avec effet de la charge $m_0 = 5\text{kg}$

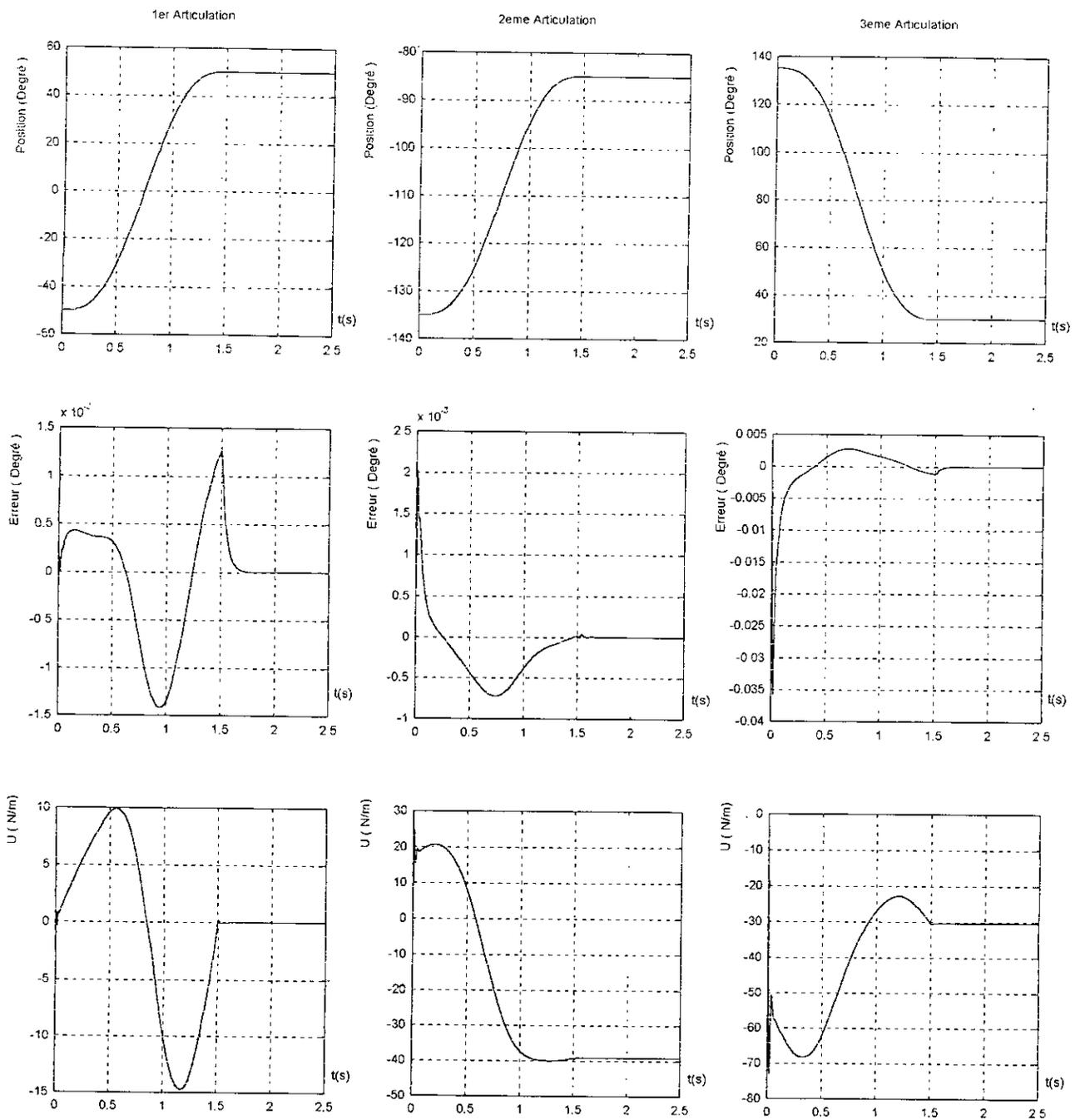


Figure (4.18) : Application de la commande par réseau de neurones au PUMA 560. Poursuite avec effet de la charge $m_0 = 10\text{kg}$

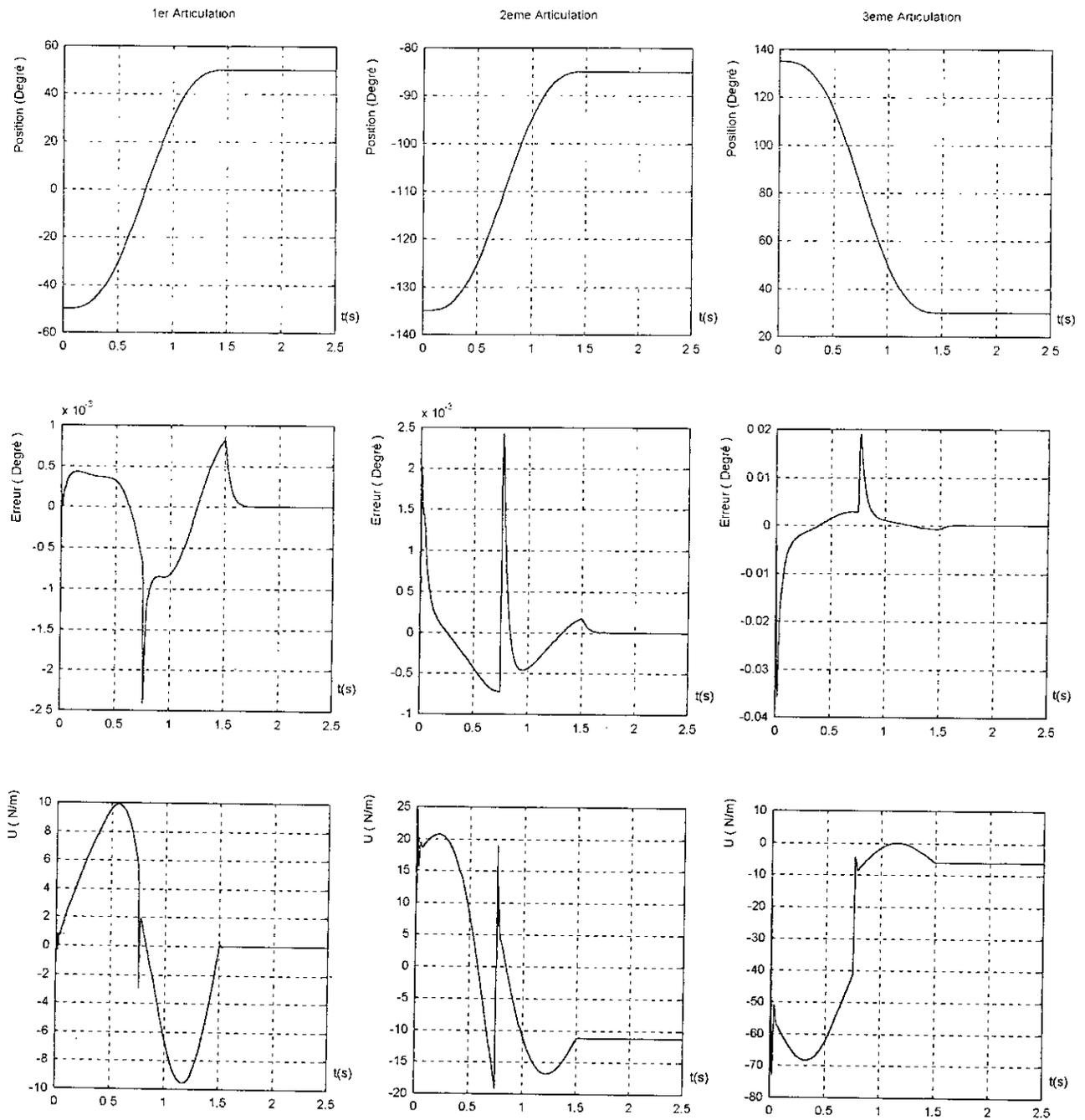


Figure (4.19) : Application de la commande par réseau de neurones au PUMA 560
Poursuite avec effet de la charge $m_0 = 10\text{kg}$ et la décharge à l'instant $t=0.75\text{s}$

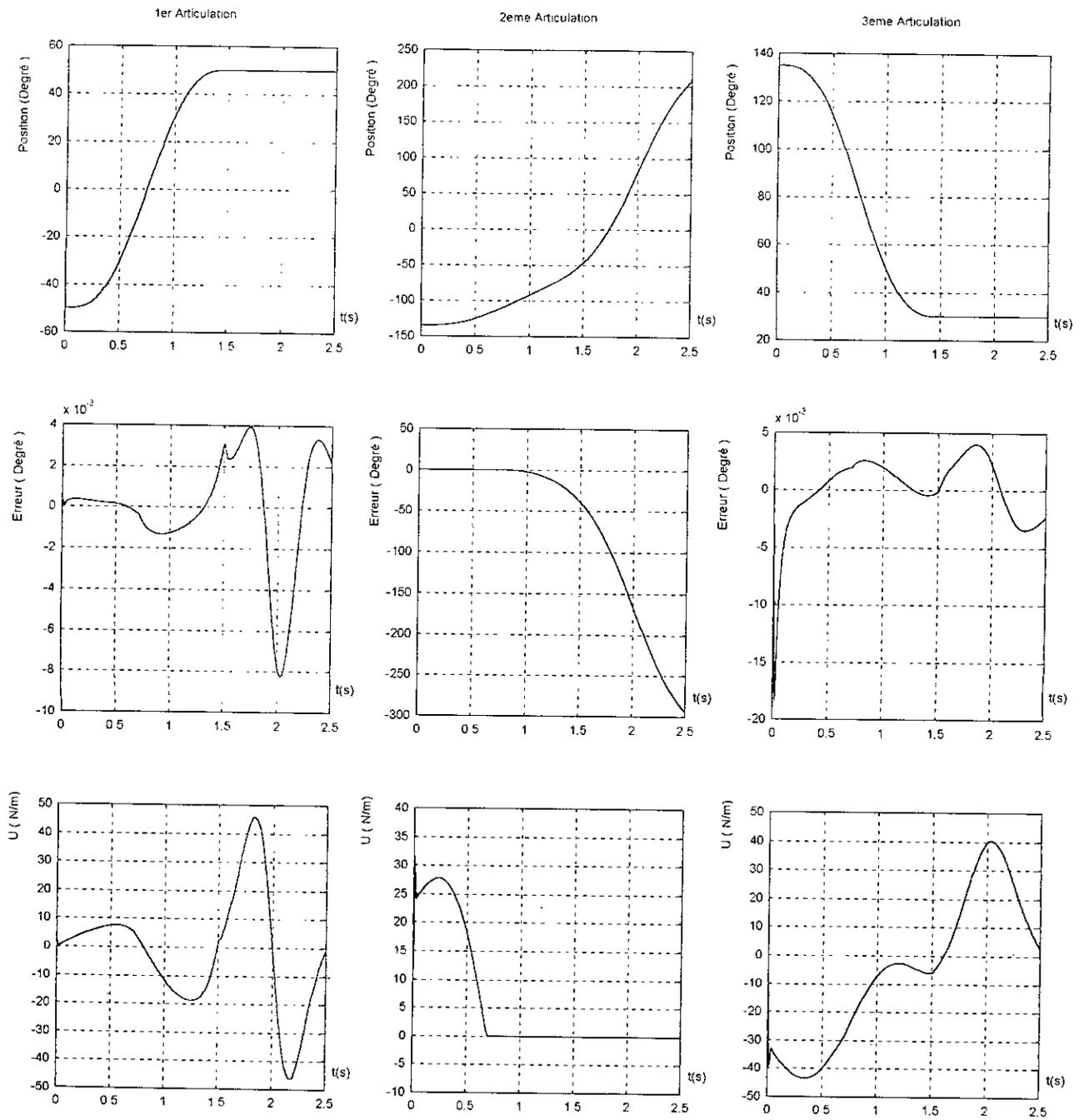
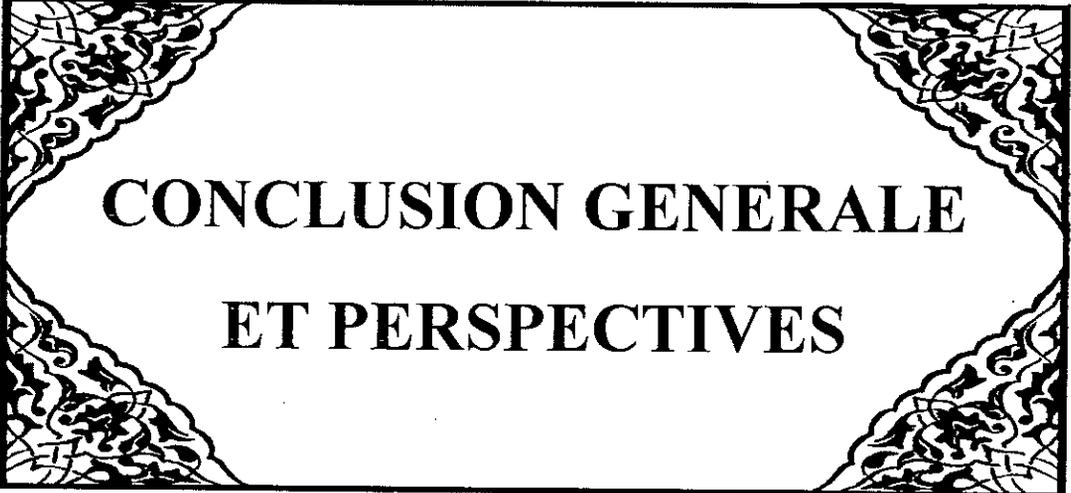


Figure (4.20) : Application de la commande par réseau de neurones au PUMA 560
Poursuite avec rupture de la deuxième commande à l'instant $t = 0.75$ s



CONCLUSION GENERALE
ET PERSPECTIVES

CONCLUSION GENERALE ET PERSPECTIVES

Les travaux présentés dans cette thèse ont porté sur l'étude des lois de commandes non linéaires avec applications aux robots manipulateurs : PUMA 560 et le robot à structure cylindrique de classe 4.

Dans le premier chapitre, nous avons appliqué le formalisme d'Euler Lagrange, qui nous a aidé à constituer un système d'équations différentielles reliant les coordonnées généralisées aux forces et /ou couples généralisé. Ce qui nous a permis d'obtenir les modèles dynamiques de deux bras manipulateurs mettant en évidence la difficulté à commander ce genre de bras manipulateurs.

Le second chapitre a été consacré à des généralités sur la commande par logique floue qui a été appliquée aux deux bras manipulateurs en utilisant trois classes d'appartenance différente.

Sur la base du critère de l'énergie de l'erreur, nous avons pu comparer les trois régulateurs et déterminer un ordre de performance pour les deux modèles ainsi que l'influence des variations de chaque gain de normalisation du meilleur régulateur que nous avons appliqué au robot PUMA 560.

Au troisième chapitre, nous avons présenté les principes de la commande par mode de glissement qui sera appliquée en combinaison avec la commande floue pour commander les deux bras manipulateurs en utilisant les mêmes classes que celle utilisées pour la commande floue classique. Sur la base du critère de l'énergie de l'erreur, nous avons pu tirer les remarques suivantes :

- Le choix de la meilleure forme de la fonction d'appartenance dépend du système c'est à dire du passage de la commande par logique floue à la commande par mode de glissement flou.
- Le régulateur flou glissant donne de meilleurs résultats que le régulateur flou classique avec en plus élimination de l'erreur statique.

De plus nous avons pu, dans cette partie, déterminer l'influence des coefficients de chaque surface de glissement sur le système.

La dernière partie de nos travaux a été consacrée à l'investigation sur l'étude de la commande des bras manipulateurs en introduisant les réseaux de neurones sur la commande par mode de glissement flou et en réalisant un algorithme d'apprentissage selon Levenberg-Marquart. Nous avons constaté que cette méthode simplifie la conception de la commande par mode de glissement flou tout en préservant les mêmes performances.

En perspectives, Notre travail, loin d'être une finalité, pourrait être prolonger dans plusieurs directions :

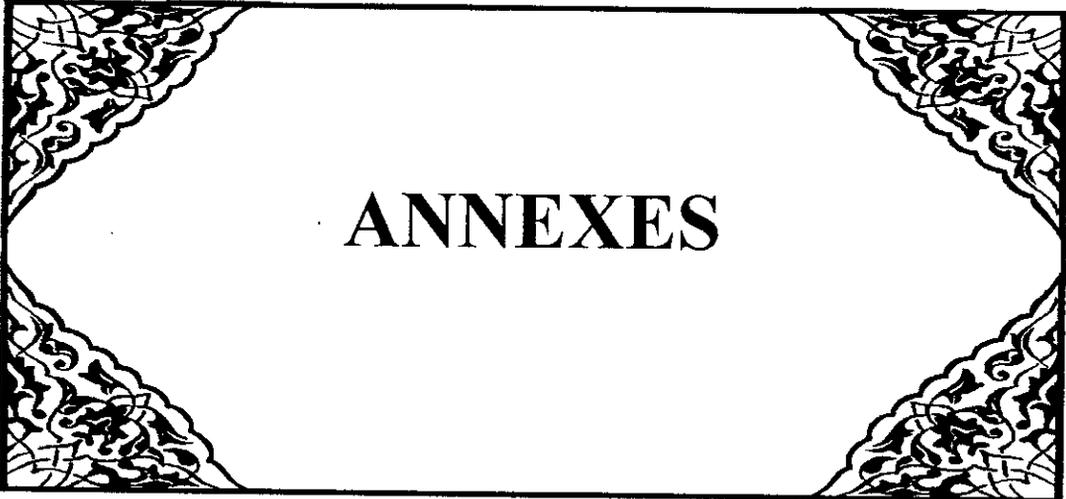
- ✦ L'application de l'algorithme de Levenberg-Marquart sur le robot à structure cylindrique de classe 4.

- ✦ L'application des commandes non linéaires présentées dans ce travail sur d'autres types de bras manipulateurs.

- ✦ Travailler sur d'autres critères de comparaison entre différentes classes d'appartenance.

- ✦ Essayer de trouver une méthode directe pour le calcul des gains de normalisation.

- ✦ Développé d'autre méthodes pour combiner la logique floue avec les modes glissants.



ANNEXES

ANNEXE 1

**Calcul du modèle dynamique du robot PUMA 560
Avec l'introduction de l'effet de l'effecteur**

La forme générale du modèle est la suivante :

$$(M(q) + \Delta M(q))\ddot{q} + D(q)\dot{q} + C(q)\dot{q}^2 + G(q) = T \quad (\text{Ann1.1})$$

Les matrices U_{ij} sont :

$$\begin{aligned}
 U_{11} &= \begin{bmatrix} -s_1 & 0 & -c_1 & 0 \\ c_1 & 0 & -s_1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} & U_{22} &= \begin{bmatrix} -s_1c_2 & s_1s_2 & -c_1 & -(l_2c_2s_1 + d_2c_1) \\ c_1c_2 & -c_1s_2 & -s_1 & (l_2c_2c_1 - d_2s_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 U_{22} &= \begin{bmatrix} -c_1s_2 & -c_1c_2 & 0 & -l_2c_1s_2 \\ -s_1s_2 & -s_1c_2 & 0 & -l_2s_1s_2 \\ -c_2 & s_2 & 0 & -l_2c_2 \\ 0 & 0 & 0 & 0 \end{bmatrix} & U_{31} &= \begin{bmatrix} -s_1c_{23} & s_1s_{23} & c_1 & -s_1(l_2c_2 + l_3c_{23}) - d_2c_1 \\ -s_1s_{23} & -s_1c_{23} & -s_1 & c_1(l_2c_2 + l_3c_{23}) - d_2s_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 U_{32} &= \begin{bmatrix} -c_1s_{23} & -c_1c_{23} & 0 & -c_1(l_2s_2 + l_3s_{23}) \\ -s_1s_{23} & -s_1c_{23} & 0 & -s_1(l_2s_2 + l_3s_{23}) \\ c_{23} & s_{23} & 0 & -(l_2c_2 + l_3c_{23}) \\ 0 & 0 & 0 & 0 \end{bmatrix} & U_{33} &= \begin{bmatrix} -c_1s_{23} & -c_1c_{23} & 0 & -l_3c_1s_{23} \\ -s_1s_{23} & -s_1c_{23} & 0 & -l_3s_1s_{23} \\ -c_{23} & s_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
 \end{aligned}$$

avec: $U_{12} = U_{13} = U_{23} = 0$ (Ann1.2)

et

$$\begin{aligned}
 c_i &= \cos(q_i) & c_{ij} &= \cos(q_i + q_j) \\
 s_i &= \sin(q_i) & s_{ij} &= \sin(q_i + q_j)
 \end{aligned}$$

La matrice J_i est donnée par :

$$J_i = \begin{bmatrix} \alpha_i & 0 & 0 & x_i \\ 0 & \beta_i & 0 & 0 \\ 0 & 0 & \gamma_i & z_i \\ x_i & 0 & z_i & m_i \end{bmatrix} \quad (\text{Ann1.3})$$

Les éléments de la matrice sont donnés par :

$$\begin{aligned}\alpha_i &= \frac{-I_{xvi} + I_{yyi} + I_{zzi}}{2} \\ \beta_i &= \frac{I_{xvi} - I_{yyi} + I_{zzi}}{2} \quad i = 1,2,3 \\ \gamma_i &= \frac{I_{xvi} + I_{yyi} - I_{zzi}}{2}\end{aligned}\quad (\text{Ann1.4})$$

et

$$x_2 = -\frac{l_2}{2}m_2 \quad ; \quad x_3 = -\frac{l_3}{2}m_3 \quad ; \quad z_1 = -\frac{l_1}{2}m_1 \quad (\text{Ann1.5})$$

Calcul de la matrice d'inertie :

$$\begin{aligned}M_{11}(q) &= \text{trace}(U_{11}J_1U_{11}^T) + \text{trace}(U_{21}J_2U_{21}^T) + \text{trace}(U_{31}J_3U_{31}^T) \\ &= \alpha_1 + \gamma_1 + \alpha_2c_2^2 + \beta_2s_2^2 + \gamma_2 + 2x_2(l_2c_2^2) + m_2(l_2^2c_2^2 + d_2(d_2 + e)) + \alpha_3c_{23}^2 \\ &\quad + \beta_3s_{23}^2 + \gamma_3 + 2x_3(c_{23}(l_2c_2 + l_3c_{23})) + m_3((l_2c_2 + l_3c_{23})^2 + d_2^2) \\ &= I_1 + I_2c_{23}^2 + I_3c_2^2 + I_4c_2c_{23}.\end{aligned}$$

$$\begin{aligned}M_{22}(q) &= \text{trace}(U_{22}J_2U_{22}^T) + \text{trace}(U_{32}J_3U_{32}^T) \\ &= \alpha_2 + \beta_2 + \alpha_3 + \beta_3 + m_3l_3c_3 + m_3l_2^2 \\ &= I_7 + I_4c_3.\end{aligned}$$

$$\begin{aligned}M_{33}(q) &= \text{trace}(U_{33}J_3U_{33}^T) \\ &= \alpha_3 + \beta_3 + 2x_3l_3 + m_3l_3^2 \\ &= I_8.\end{aligned}$$

$$\begin{aligned}M_{12}(q) &= \text{trace}(U_{22}J_2U_{21}^T) + \text{trace}(U_{32}J_3U_{31}^T) \\ &= x_2(d_2s_2) + m_2(l_2d_2s_2) + x_3(d_2s_{23}) + m_3(l_2s_2 + l_3s_{23})d_2 \\ &= I_5s_{23} + I_6s_2.\end{aligned}$$

$$\begin{aligned}M_{13}(q) &= \text{trace}(U_{33}J_3U_{31}^T) \\ &= x_3(d_2s_{23}) + m_3(l_3d_2s_{23}) \\ &= I_5s_{23}.\end{aligned}$$

$$\begin{aligned}M_{23}(q) &= \text{trace}(U_{33}J_3U_{32}^T) \\ &= \alpha_3 + \beta_3 + x_3(2l_3 + l_2c_3) + m_3(l_3^2 + l_2l_3c_3) \\ &= I_8 + 0.5I_4c_3.\end{aligned}$$

$$M_{12}(q)=M_{12} \quad ; \quad M_{31}(q)=M_{13}(q) \quad ; \quad M_{32}(q)=M_{23}(q) \quad (\text{Ann1.6})$$

avec:

$$\begin{aligned}
 I_1 &= \alpha_1 + \gamma_1 + \beta_2 + \gamma_2 + \beta_3 + \gamma_3 m_3 d_2^2 + m_2 d_2 (d_2 + e) \\
 &= I_{yy1} + I_{xx2} + m_2 d_2 (d_2 + e) + m_3 d_2^2 + I_{xx3} \\
 I_2 &= \alpha_1 - \beta_3 = I_{yy3} - I_{xx3} \\
 I_3 &= \alpha_3 - \beta_3 + m_3 l_2^2 = I_{yy2} - I_{xx2} + m_3 l_2^2 \\
 I_4 &= m_3 l_2 l_3 \\
 I_5 &= 0.5 m_3 l_3 d_2 \\
 I_6 &= 0.5 m_2 l_2 (d_2 + e) + m_3 d_2 l_2 \\
 I_7 &= \alpha_2 + \beta_2 + \alpha_3 + \beta_3 + m_3 l_2^2 = I_{zz2} + I_{zz3} + m_3 l_2^2 \\
 I_8 &= I_{zz3}
 \end{aligned} \tag{Ann1.7}$$

Calcul de la matrice des couples de Coriolis :

$$\begin{aligned}
 D_{11}(q) &= \left(\frac{\partial M_{11}}{\partial q_2} + \frac{\partial M_{12}}{\partial q_1} - \frac{\partial M_{12}}{\partial q_1} \right) = -2I_2 c_{23} s_{23} - 2I_3 c_2 s_2 - I_2 (s_2 c_{23} + c_2 s_{23}) \\
 D_{12}(q) &= \left(\frac{\partial M_{11}}{\partial q_3} + \frac{\partial M_{13}}{\partial q_1} - \frac{\partial M_{13}}{\partial q_1} \right) = -2I_2 c_{23} s_{23} - I_4 c_2 s_{23} \\
 D_{13}(q) &= \left(\frac{\partial M_{12}}{\partial q_3} + \frac{\partial M_{13}}{\partial q_2} - \frac{\partial M_{23}}{\partial q_1} \right) = 2I_5 c_{23} \\
 D_{21}(q) &= \left(\frac{\partial M_{21}}{\partial q_2} + \frac{\partial M_{12}}{\partial q_1} - \frac{\partial M_{12}}{\partial q_2} \right) = 0 \\
 D_{23}(q) &= \left(\frac{\partial M_{22}}{\partial q_3} + \frac{\partial M_{23}}{\partial q_2} - \frac{\partial M_{23}}{\partial q_2} \right) = -I_4 s_3 \\
 D_{31}(q) &= \left(\frac{\partial M_{131}}{\partial q_3} + \frac{\partial M_{32}}{\partial q_1} - \frac{\partial M_{12}}{\partial q_3} \right) = 0 \\
 D_{32}(q) &= \left(\frac{\partial M_{31}}{\partial q_3} + \frac{\partial M_{33}}{\partial q_1} - \frac{\partial M_{13}}{\partial q_3} \right) = 0 \\
 D_{33}(q) &= \left(\frac{\partial M_{32}}{\partial q_3} + \frac{\partial M_{33}}{\partial q_2} - \frac{\partial M_{23}}{\partial q_3} \right) = 0
 \end{aligned} \tag{Ann1.8}$$

Calcul de la matrice des couples de centrifuges :

$$C_{11}(q) = \frac{1}{2} \left(\frac{\partial M_{11}}{\partial q_1} + \frac{\partial M_{11}}{\partial q_1} - \frac{\partial M_{11}}{\partial q_1} \right) = 0.$$

$$C_{12}(q) = \frac{1}{2} \left(\frac{\partial M_{12}}{\partial q_2} + \frac{\partial M_{12}}{\partial q_2} - \frac{\partial M_{22}}{\partial q_1} \right) = I_5 c_{23} + I_6 c_2.$$

$$C_{12}(q) = \frac{1}{2} \left(\frac{\partial M_{13}}{\partial q_2} + \frac{\partial M_{13}}{\partial q_3} - \frac{\partial M_{33}}{\partial q_1} \right) = I_5 c_{23}.$$

$$C_{21}(q) = \frac{1}{2} \left(\frac{\partial M_{21}}{\partial q_1} + \frac{\partial M_{21}}{\partial q_1} - \frac{\partial M_{11}}{\partial q_2} \right) = I_2 c_{23} s_{23} + I_3 c_2 s_2 + 0.5 I_4 (s_{23} c_2 + s_2 c_{23}).$$

$$C_{22}(q) = \frac{1}{2} \left(\frac{\partial M_{22}}{\partial q_2} + \frac{\partial M_{22}}{\partial q_2} - \frac{\partial M_{22}}{\partial q_2} \right) = 0.$$

$$C_{23}(q) = \frac{1}{2} \left(\frac{\partial M_{23}}{\partial q_3} + \frac{\partial M_{23}}{\partial q_3} - \frac{\partial M_{33}}{\partial q_2} \right) = -0.5 I_4 s_3.$$

$$C_{31}(q) = \frac{1}{2} \left(\frac{\partial M_{31}}{\partial q_1} + \frac{\partial M_{31}}{\partial q_1} - \frac{\partial M_{11}}{\partial q_3} \right) = I_2 s_{23} c_{23} + 0.5 I_4 s_{23} c_2.$$

$$C_{32}(q) = \frac{1}{2} \left(\frac{\partial M_{32}}{\partial q_2} + \frac{\partial M_{32}}{\partial q_2} - \frac{\partial M_{22}}{\partial q_3} \right) = 0.5 I_4 s_3.$$

$$C_{33}(q) = \frac{1}{2} \left(\frac{\partial M_{33}}{\partial q_3} + \frac{\partial M_{33}}{\partial q_3} - \frac{\partial M_{33}}{\partial q_3} \right) = 0.$$

(Ann1.9)

Les couples de gravitation sont donnés par :

$$r_1^1 = \begin{bmatrix} 0 & 0 & -\frac{l_1}{2} & 1 \end{bmatrix}^T ; \quad r_2^2 = \begin{bmatrix} -\frac{l_2}{2} & 0 & d_2 & 1 \end{bmatrix}^T ; \quad r_3^3 = \begin{bmatrix} -\frac{l_3}{2} & 0 & 0 & 1 \end{bmatrix}^T \quad (\text{Ann1.10})$$

et

$$g = \begin{bmatrix} 0 & 0 & -|g| & 1 \end{bmatrix}^T \quad (\text{Ann1.11})$$

On aura donc :

$$G_1(q) = -m_1 g^T U_{11} r_1^1 - m_2 g^T U_{21} r_2^2 - m_3 g^T U_{31} r_3^3 = 0.$$

$$G_2(q) = -m_2 g^T U_{22} r_2^2 - m_3 g^T U_{32} r_3^3 = -0.5 m_2 g l_2 c_2 - 0.5 m_3 g l_3 c_{23} - m_3 g l_2 c_2. \quad (\text{Ann1.12})$$

$$G_3(q) = -m_3 g^T U_{33} r_3^3 = 0.5 m_3 g l_3 c_{23}.$$

On introduit l'effet de l'effecteur :

$$J_4 = \begin{bmatrix} \alpha_4 & 0 & 0 & 0 \\ 0 & \beta_4 & 0 & 0 \\ 0 & 0 & \gamma_4 & 0 \\ 0 & 0 & 0 & m_1 \end{bmatrix} \quad (\text{Ann1.13})$$

avec

$$\begin{aligned} \alpha_4 &= \frac{-I_{xvi} + I_{yvi} + I_{zzi}}{2} \\ \beta_4 &= \frac{I_{xv} - I_{yvi} + I_{zzi}}{2} \\ \gamma_4 &= \frac{I_{xvi} + I_{yvi} - I_{zzi}}{2} \end{aligned} \quad (\text{Ann1.14})$$

Calcul de la matrice ΔM :

$$\begin{aligned} \Delta M_{11} &= \text{trace}(U_{31} J_4 U_{31}^T) + I_{M1} \\ &= \alpha_4 (s_1^2 c_{23}^2 + c_1^2 c_{23}^2) + \beta_4 (s_1^2 s_{23}^2 + c_1^2 s_{23}^2) + \gamma_4 \\ &\quad + m_1 [(s_1 (l_2 c_2 + l_3 c_{23}) + d_2 c_1)^2 + (c_1 (l_2 c_2 + l_3 c_{23}) - d_2 s_1)^2] + I_{M1} \end{aligned}$$

$$\begin{aligned} \Delta M_{22}(q) &= \text{trace}(U_{32} J_4 U_{32}^T) + I_{M2} \\ &= \alpha_4 (c_1^2 s_{23}^2 + s_1^2 s_{23}^2 + c_{23}^2) + \beta_4 (c_1^2 c_{23}^2 + s_1^2 c_{23}^2 + s_{23}^2) \\ &\quad + m_1 [c_1^2 (l_2 s_2 + l_3 s_{23})^2 + s_1^2 (l_2 s_2 + l_3 s_{23})^2] + I_{M2} \end{aligned}$$

$$\begin{aligned} \Delta M_{33}(q) &= \text{trace}(U_{33} J_4 U_{33}^T) + I_{M3} \\ &= \alpha_4 (c_1^2 s_{23}^2 + s_1^2 s_{23}^2 + c_{23}^2) + \beta_4 (c_1^2 c_{23}^2 + s_1^2 c_{23}^2 + s_{23}^2) \\ &\quad + m_1 (l_3^2 c_1^2 s_{23}^2 + l_3^2 s_1^2 s_{23}^2 + l_3^2 c_{23}^2) + I_{M3} \end{aligned}$$

$$\begin{aligned} \Delta M_{12}(q) &= \text{trace}(U_{32} J_4 U_{31}^T) \\ &= m_1 d_2 l_2 s_2 + m_1 d_2 l_3 s_{23} \end{aligned}$$

$$\begin{aligned} \Delta M_{13}(q) &= \text{trace}(U_{33} J_4 U_{31}^T) \\ &= \alpha_4 s_1 c_1 s_{23} c_{23} - \beta_4 c_1 s_1 c_{23} s_{23} + m_1 l_3 c_1 s_{23} [s_1 (l_2 c_2 + l_3 c_{23}) + d_2 c_1] \\ &\quad - \alpha_4 s_1 c_1 s_{23} c_{23} + \beta_4 c_1 s_1 c_{23} s_{23} + m_1 l_3 s_1 s_{23} [c_1 (l_2 c_2 + l_3 c_{23}) + d_2 s_1] \end{aligned}$$

$$\begin{aligned} \Delta M_{23}(q) &= \text{trace}(U_{33} J_4 U_{32}^T) \\ &= \alpha_4 + \beta_4 + m_1 l_2 l_3 c_3 + m_1 l_3^2 \end{aligned}$$

$$\Delta M_{21}(q) = \Delta M_{12}(q) \quad ; \quad \Delta M_{31}(q) = \Delta M_{13}(q) \quad ; \quad \Delta M_{32}(q) = \Delta M_{23}(q) \quad (\text{Ann1.15})$$

ANNEXE 2

Calcul du modèle dynamique du robot à structure cylindrique de classe 4

L'énergie potentielle :

$$E_p = (m_2 + m_3 + m_0)g(q_2 - q_{20}) \quad (\text{Ann2.1})$$

d'où q_{20} correspond à la position particulière q_2 du manipulateur pour laquelle l'énergie potentielle est nulle.

L'énergie cinétique s'exprime par :

$$E_c = \frac{1}{2} \left[(I_2 + I_3 + m_3(q_3 - l_3)^2 + m_0 q_3^2) \dot{q}_1^2 + (m_2 + m_3 + m_0) \dot{q}_2^2 + (m_3 + m_0) \dot{q}_3^2 \right] \quad (\text{Ann2.2})$$

Le lagrangien est :

$$\begin{aligned} L &= E_c - E_p \\ &= \frac{1}{2} \left[(I_2 + I_3 + m_3(q_3 - l_3)^2 + m_0 q_3^2) \dot{q}_1^2 + (m_2 + m_3 + m_0) \dot{q}_2^2 + (m_3 + m_0) \dot{q}_3^2 \right] \\ &\quad - (m_2 + m_3 + m_0)g(q_2 - q_{20}) \end{aligned} \quad (\text{Ann2.3})$$

L'énergie de dissipation est donnée par :

$$E_D = \frac{1}{2} \left(f_{v1} \dot{q}_1^2 + f_{v2} \dot{q}_2^2 + f_{v3} \dot{q}_3^2 \right) \quad (\text{Ann2.4})$$

L'équation d'Euler-Lagrange pour la première articulation est :

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_1} \right] - \frac{\partial L}{\partial q_1} + \frac{\partial E_D}{\partial \dot{q}_1} = \Gamma_1 \quad (\text{Ann2.5})$$

donc :

$$(I_2 + I_3 + m_3(q_3 - l_3)^2 + m_0 q_3^2) \ddot{q}_1 + 2(m_3(q_3 - l_3) + m_0 q_3) \dot{q}_1 \dot{q}_3 + f_{v1} \dot{q}_1 = \Gamma_1 \quad (\text{Ann2.6})$$

L'équation d'Euler-Lagrange pour la deuxième articulation est :

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_2} \right] - \frac{\partial L}{\partial q_2} + \frac{\partial E_D}{\partial \dot{q}_2} = F_2 \quad (\text{Ann2.7})$$

donc :

$$(m_2 + m_3 + m_0) \ddot{q}_2 + (m_2 + m_3 + m_0)g + f_{v2} \dot{q}_2 = F_2 \quad (\text{Ann2.8})$$

L'équation d'Euler-Lagrange pour la troisième articulation est :

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_3} \right] - \frac{\partial L}{\partial q_3} + \frac{\partial E_D}{\partial \dot{q}_3} = F_3 \quad (\text{Ann2.9})$$

donc :

$$(m_3 + m_0) \ddot{q}_3 - (m_3(q_3 - l_3) + m_0 q_3) \dot{q}_1^2 + f_{v3} \dot{q}_3 = F_3 \quad (\text{Ann2.10})$$

ANNEXE 3

Gains de normalisation des différents régulateurs

Gains de normalisations des régulateurs flous appliqués au robot PUMA 560 :

	5 classes d'appartenance			7 classes d'appartenance			9 classes d'appartenance		
	K_e	K_{de}	K_u	K_e	K_{de}	K_u	K_e	K_{de}	K_u
1 ^{ère} articulation	100	0.2	70	45	0.2	70	40	0.3	50
2 ^{ème} articulation	140	0.25	120	135	0.35	100	100	0.4	100
3 ^{ème} articulation	40	0.12	120	30	0.135	95	30	0.135	85

Gains de normalisations des régulateurs flous appliqués au robot à structure cylindrique:

	5 classes d'appartenance			7 classes d'appartenance			9 classes d'appartenance		
	K_e	K_{de}	K_u	K_e	K_{de}	K_u	K_e	K_{de}	K_u
1 ^{ère} articulation	200	5	15	200	5	10	200	5	20
2 ^{ème} articulation	200	10	10	200	10	15	200	10	10
3 ^{ème} articulation	200	10	15	200	15	15	200	10	10

Gains de normalisations des régulateurs flous glissants appliqués au robot PUMA 560 :

	5 classes d'appartenance			7 classes d'appartenance			9 classes d'appartenance		
	K_e	K_{de}	K_u	K_e	K_{de}	K_u	K_e	K_{de}	K_u
1 ^{ère} articulation	33	0.03	100	30	0.003	91.5	28	0.006	90
2 ^{ème} articulation	56.5	0.03	160	56.5	0.013	130	56.5	0.013	140
3 ^{ème} articulation	16.5	0.02	130	10.5	0.02	120	16.5	0.023	100

Gains de normalisations des régulateurs flous glissants appliqués au robot à structure cylindrique :

	5 classes d'appartenance			7 classes d'appartenance			9 classes d'appartenance		
	K_e	K_{de}	K_u	K_e	K_{de}	K_u	K_e	K_{de}	K_u
1 ^{ère} articulation	156.5	10	10	100	8	10	66	6.6	10
2 ^{ème} articulation	156.5	10	10	100	5	10	66	6.6	10
3 ^{ème} articulation	156.5	11.5	10	100	11.5	10	66	8.5	10

K_e : gain de normalisation de l'erreur

K_{de} : gain de normalisation de la variation de l'erreur

K_u : gain de normalisation de la commande.

ANNEXE 4

Poids et biais des réseaux de neurones

$$w_{11} = \begin{bmatrix} 3.366 & -11.678 & 1.865 \\ -7.345 & 11.459 & -3.722 \\ -3.208 & 9.505 & -1.775 \\ -3.208 & 0.0577 & 208.584 \end{bmatrix}$$

$$w_{21} = \begin{bmatrix} 3.366 & -11.677 & 1.865 \\ -7.345 & 11.458 & -3.722 \\ -3.208 & 9.503 & -3.77 \\ 18.87 & 0.102 & 376.908 \end{bmatrix}$$

$$w_{31} = \begin{bmatrix} 6.902 & -2.263 & 1.778 \\ 10.15 & 0.695 & -246.775 \\ 24.83 & 0.738 & -66.662 \\ 10.41 & 0.3035 & 25.4 \end{bmatrix}$$

$$B_{11} = \begin{bmatrix} -18.295 \\ -18.763 \\ 20.344 \\ 1.443 \end{bmatrix}$$

$$B_{21} = \begin{bmatrix} -18.297 \\ -18.765 \\ 20.351 \\ -0.434 \end{bmatrix}$$

$$B_{31} = \begin{bmatrix} -16.663 \\ -0.1546 \\ 3.129 \\ 1.163 \end{bmatrix}$$

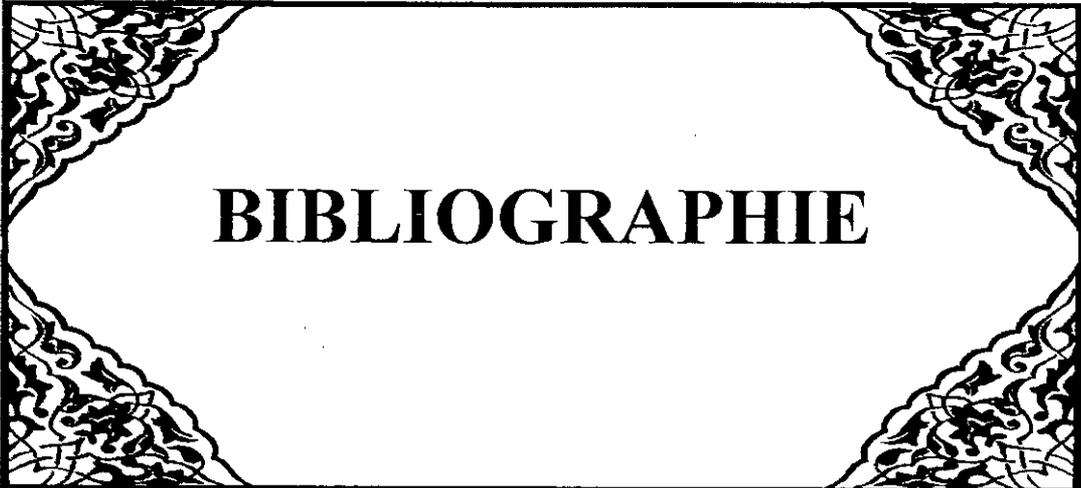
Avec w_{11} , w_{21} , w_{31} , B_{11} , B_{21} et B_{31} sont les poids et les biais de la couche cachée.

$$w_{12} = [9.381 \quad -16.093 \quad -623.47 \quad 1536.9] \quad w_{22} = [9.388 \quad -16.093 \quad -342.89 \quad 1735.3]$$

$$w_{32} = [-28.608 \quad -367.1 \quad -547.58 \quad 1573.3]$$

$$B_{12} = -619.87 \quad B_{22} = -339.28 \quad B_{32} = -505$$

Avec w_{12} , w_{22} , w_{32} , B_{12} , B_{22} et B_{32} sont les poids et les biais de la couche de sortie.



BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [1] **A. Akhnaq et Cherifi**, « *Commande floue et neuro-floue d'un filtre actif parallèle a deux niveaux* », PFE, ENP, 1998.
- [2] **B.Amstrong et O.Khatib et J.Burdick**, « *The explicit dynamic model and inertial parameters of the PUMA 560 arm* », Proceeding IEEE, pp. 510-519, 1986.
- [3] **Arago**, « *Logique floue* », O.F.T.A. Masson, Paris 1994.
- [4] **N.Bali**, « *Etude des performances de la commande prédictive généralisée* », Thèse de Magistère, ENP, 1995.
- [5] **F. Boudjema**, « *Commande par mode de glissement : application aux convertisseur électrique* », Thèse de Doctorat d'Etat, ENP, Avril, 1995.
- [6] **Bouhssoun**, « *Contribution a la commande par mode de glissement de la machine asynchrone avec pilotage vectoriel* », Thèse de magistère, ENP, 1998.
- [7] **H. Bühler**, « *Réglage par logique floue* », Presses polytechnique et université romandes, 1994.
- [8] **H. Bühler**, « *Réglage par mode de glissement* », Presses polytechnique et université romandes, 1986.
- [9] **Y.P.Chen et Kais Yeung**, « *A new controller design for manipulators using the theory of variable structure systems* », IEEE Transaction on Automatic control, Vol 33, N°2, 1988.
- [10] **S.L.Chiu et F.J.LIN**, « *Adaptative fuzzy sliding-mode control for PM synchronous serco motor drives* », IEE Proc Control Theory Appli., Vol 145, N°1, 1998.
- [11] **P.Coifet**, « *interaction avec l'environnement* », Edition Hermes, 1981.
- [12] **R.A.Decarlo et S.H.Zac et G.P.Matthews**, « *Variable structure control of nonlinear multivariable system :Atutorial* », Proceedings of IEEE, Vol 76, N°3, pp 212-232, 1988.
- [13] **H. Demuth et M. Beale**, « *The neural network toolbox for use with MATLAB* », Natick, MA : The mathWork, Inc, Copyright 1997.
- [14] **J.A. Freeman et D.M. Skapura**, « *Neural Networks : Algorithms, applications, and programming techniques* », Ed Addison-Wesley, 1992.

- [15] **M.T.Hagan et M.Menhaj**, « *Training feed-forward networks with the Marquardt algorithm* », IEEE Transaction on Neuro Network, Vol.5, N°6, pp 989 993, 1994.
- [16] **F.Hadj Miloud**, « *Commande adaptative decentralisée par la technique du Backstepping : application en robotique* », PFE, ENP, 1998.
- [17] **Hadj Sadok**, « *Modélisation et commande floue par mode de glissement et neuro linguistique : application en robotique a degré de liberté* », PFE, ENP, 1994.
- [18] **M. Jimshidi**, « *Fuzzy logic and control* », Ed Printice Hall, 1993.
- [19] **J.F.Jodouin**, « *Les réseaux de neurones principes et définitions* », ED. Hermes, 1994.
- [20] **Y.Koren**, « *La robotique pour ingenieurs* », Edition Mc Graw-Hill, 1986.
- [21] **B. Kosko**, « *Neural networks and Fuzzy systems* », Ed Prentice-Hall, 1992.
- [22] **G.Langari et M.Tomizuka**, « *Stability of fuzzy linguistic control system* », IEEE Proc of the 29th Conference on decision and control, December 1990.
- [23] **M.Larabi**, « *Différentes approches de commandes à structure variable : Application à une machine asynchrone* », PFE, ENP, 1997.
- [24] **M.B.Leahvy Jr**, « *Model based auxiliary input control : developement and experimental analysis* », Proceedings of the 29th conference of decision and control Honolulu, Hawai, December 1990.
- [25] **T.P.Leung et Q.J.Zhou et Chun Yi Su**, « *An adaptative variable structure model following control design for robot manipulators* », IEEE Transaction on Automatic Control, Vol 36, N°36, 1991.
- [26] **Y.F. Li et C.C. Lau**, « *Developement of fuzzy algorithmes for servo-systems* », IEEE Control Systems Magazine, pp 65-79, April 1989.
- [27] **M.Loudini**, « *Modélisation, Analyse et Méthodologie de commande linguistique Floue d'un brs manipulateur de robot flexible* », Thèse de Magistère, ENP, 1997.
- [28] **M. Madani et S.Amrati**, « *Commande décentralisée à structure variable : application en Robotique* », PFE, ENP, 1997.
- [29] **E.H. Mamdani**, « *An experiment in linguistic synthesis with a fuzzy logic controller* », int. J. Man Mach. Studies, Vol. 7, N°.1,pp 1-73, 1975.
- [30] **M. Mehdaoui et M.N Dahia**, « *Commande et observateur à mode glissant flou d'une machine asynchrone* », PFE, ENP, 1999.
- [31] **Melahi**, « *Commande floue et neuro-floue decentralisée : application en robotique* », PFE, ENP, 1997.

- [32] S. Netfi et Y. Baba-Hadji, « *Commande floue et neuro-linguistique d'une machine asynchrone triphasée* », PFE, ENP, 1995.
- [33] C.P. Paris et A.T. Alouani, « *On the implementation of a fuzzy variable structure control* », IEEE Control systems, pp 287-291, 1998.
- [34] F.Rosenblatt, « *Principales of Neuro dynamics* », Washington D.C, Spartanpress 1961.
- [35] Patterson Dan W, « *Artificial neural networks theory and applications* », Printice Hall, 1996.
- [36] Seba et Souami, « *Comparaison entre la commande floue et classique : Application a un bras de robot* », PFE, ENP, 1999.
- [37] J.J.Slotine, « *Sliding controller design for non-linear systems* », I.J.C., Vol 40, pp 421-434, 1984.
- [38] J.J.Slotine, « *Adaptative sliding controller synthesis for no-linear systems* », I.J.C, Vol 43, N° 6, pp 1631-1651, 1986
- [39] J.J.Slotine, « *Applied non-linear control* », Englewood Cliffs. NJ: Printice Hall, 1991.
- [40] V.I.Utkin, « *Variable structure systems with sliding mode* », IEEE Transactions on Automatic control, Vol. Ac, N° 2, April 1977.
- [41] V.i. Utkin, « *Sliding mode control design principals and applications to electric drives* », IEEE Transactions on Industrial Electronics, Vol. 40, N° 1, February 1993.
- [42] R.R. Yager et L.A.Zadeh, « *Fuzzy sets, Neural Network, and Soft computing* », Ed Van Vostrand Reinhold, 1994.
- [43] L.A. Zadeh, « *Fuzzy logic* », IEEE Computer, pp. 83-93, April 1988.
- [44] L.A. Zadeh, « *Fuzzy sets* », Information and Control, Vol. 8, pp. 338-353, 1965.
- [45] L.Zeroual, « *Application de la commande floue et neuro-floue a structure décentralisée au robot PUMA 560* », PFE, ENP, 1998.

المخلص

في هذه المذكرة اقترحنا ثلاث طرق لتحكم اللامركزي مطبقة على ذراعين آليين : ذراع PUMA 560 وذراع نو بينية أسطوانية. بنية التحكم المستعملة تعتبر الذراع الآلي كنظام مترابط بحيث كل مفصل يكون نظام جزئي ويتحكم في هذا الأخير من مقر التحكم المحلي . الطريقة الأولى للتحكم هي المنطق الغامض من نوع ممداني لأقسام مختلفة . الطريقة الثانية المطبقة هي نمط الانزلاق الغامض اعتمادا على مبادئ وجود النماذج المنزقة. وفي الأخير طريقة التحكم بالشبكة العصبية التي تسهل بنية التحكم . قدمنا نتائج التحليل وطبقنا اختبار عرض العضالة على الآليين ، التي أعطت نتائج مرضية

كلمات المفتاح : المنطق الغامض ، نمط الانزلاق ، الشبكة العصبية ، الذراع الآلي

Abstract

In this work we propose three types of decentralized controls which are applied to two manipulator robots at three degree of freedom: PUMA 560 and the robot at the cylindrical structure. The structure of the control developed considers the robot as an interconnected system in which each articulation constitutes a subsystem controlled by a local control system. The first applied control is the classical fuzzy, type MAMDANI for different classes. The second applied control is the control by the fuzzy sliding mode, based on the existing condition of the sliding modes with the same classes used in the classical fuzzy control. Finally, the last control is the approach by the neural network, which has the advantage of simplifying the structure of the control. For the two robots, we have presented the simulation results and performed the robustness tests, which gives good results.

Key words: fuzzy logic, sliding mode, neural network, manipulator robot.

Résumé

Dans ce travail nous proposons trois types de commande décentralisée appliquée a deux robots manipulateurs a trois degrés de liberté : PUMA 560 et le robot à structure cylindrique. La structure de la commande développé considère le robot comme un système interconnecté dans lequel chaque articulation constitue un sous système commandé par une station de commande locale. La première commande appliquée est la commande floue classique du type MAMDANI pour différent choix de classes. La deuxième commande appliqué est la commande par mode de glissement flou, basée sur la condition d'existence des modes glissant avec les même classes prises dans la commande floue classique. Enfin, la dernière commande est l'approche par réseaux de neurones, qui a l'avantage de simplifier la structure de la commande. Pour les deux robots nous avons présenté les résultats de simulations et effectuer des tests de robustesses, qui s'avèrent réussis.

Mots clés : logique floue, mode glissant flou, réseaux de neurones, robot manipulateur