

1/01

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي و البحث العلمي

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

ECOLE NATIONALE POLYTECHNIQUE

Département de Génie Electrique

Filière Automatique

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

Projet de Fin d'Etudes

En vue d'obtenir le Diplôme

d'Ingénieur d'Etat en Automatique

**COMMANDE NON LINEAIRE DECENTRALISEE
AVEC OBSERVATEUR A RESEAU DE NEURONES
ARTIFICIELS DES INTERCONNEXIONS ET DES
PERTURBATIONS : APPLICATION EN ROBOTIQUE**

Proposé et dirigé par :

M. D. BOUKHETALA
M. F. BOUDJEMA

Etudié par :

M. ABBAS-TURKI Mohamed
M. BENALI Mehdi

PROMOTION 2001

REMERCIEMENTS

Nous ne saurions témoigner toute la reconnaissance que nous devons à Messieurs D. BOUKHETALA et F. BOUDJEMA pour les nombreuses discussions et conseils très utiles qu'ils n'ont cessé de nous prodiguer tout au long de ce travail. Qu'ils trouvent ici nos plus respectueux remerciements.

Nous tenons à témoigner notre profonde gratitude à Messieurs les membres du jury pour avoir accepté d'examiner ce travail et jugé du résultat.

Nous sommes également très reconnaissants à tous les enseignants du Département de Génie Electrique, sans l'aide desquels ce mémoire n'aurait pas eu son contenu actuel.

TABLE DES MATIERES

1	INTRODUCTION GENERALE	
4	CHAPITRE I MODELISATION D'UN BRAS DE ROBOT MANIPULATEUR	
5	I.1 Introduction	
5	I.2 Les différentes générations des robots	
6	I.3 Les différents degrés de liberté	
6	I.4 Modélisation cinématique	
6	I.4.1 Modèle géométrique du robot PUMA 560	
10	I.5 Modélisation dynamique	
10	I.5.1 Modèle dynamique global d'un robot	
11	I.5.2 Modèle dynamique décentralisé d'un robot	
12	I.5.3 Modèle dynamique du robot PUMA 560	
14	I.6 Essai à vide du robot PUMA 560	
14	I.7 Conclusion	
15	CHAPITRE II OBSERVATEUR PAR RESEAU DE NEURONES	
16	II.1 Introduction	
16	II.2 Réseaux de neurones	
17	II.2.1 Présentation et historique des réseaux de neurones	
18	II.2.2 Le modèle biologique	
19	II.2.3 Les modèles mathématiques	
22	II.2.4 Structure des réseaux de neurones	
25	II.2.5 Apprentissage	
26	II.2.6 Algorithme de rétropropagation (Back-Propagation)	
27	II.2.7 Conclusion	
28	II.3 Observateur linéaire	
28	II.3.1 Structure de l'observateur linéaire	
30	II.4 Observateur non linéaire	
30	II.4.1 Théorie de base pour l'observation	
30	II.4.2 Différentes approches pour la synthèse d'un observateur non linéaire	
36	CHAPITRE III SYSTEMES A STRUCTURE VARIABLE ET BACKSTEPPING	
37	III.1 Systèmes à structure variable	
37	III.1.1 Introduction	
38	III.1.2 Propriétés des modes glissants	
38	III.1.3 Configuration de base pour les systèmes à structure variable	
39	III.1.4 Solution des systèmes à second membre discontinu	
40	III.1.5 Conception de la commande par mode de glissement	
40	III.1.5.1 Choix de la surface de glissement	
41	III.1.5.2 Condition de convergence et d'existence du mode de glissement	

42	III.1.5.2.1 La fonction directe de commutation
43	III.1.5.2.2 Fonction de Lyapunov
43	III.1.5.3 Calcul de la commande
45	III.1.6 Invariance des modes glissants par rapport aux perturbations
45	III.1.7 Invariance des modes glissants par rapport à une transformation
46	III.1.8 Phénomène de broutement (chattering)
47	III.1.9 Conclusion
47	III.2 Backstepping
47	III.2.1 Introduction
47	III.2.2 Le Backstepping dans le cas monovariable
50	III.2.3 Le Backstepping dans le cas multivariables
51	III.2.4 Robustesse de la commande synthétisée par Backstepping
52	III.2.5 Conclusion
53	CHAPITRE IV COMMANDE DECENTRALISEE PAR MODE DE GLISSEMENT ET BACKSTEPPING AVEC OBSERVATEUR A RESEAU DE NEURONES ARTIFICIELS
54	IV.1 Introduction
54	IV.2 Données de base
58	IV.3 Observateur par réseau de neurones des interconnexions
60	IV.4 Synthèse de la loi de commande décentralisée robuste
64	IV.5 Résultats de simulation
78	IV.6 Conclusion
79	CHAPITRE V ETUDE COMPARATIVE
80	V.1 Régulation du PUMA 560 par un proportionnel dérivé (PD)
84	V.2 Présentation générale de la méthode Y. Tang et coll.
91	V.3 Etude comparative de chaque loi de commande sur le PUMA 560
95	V.4 Etude comparative approfondie
95	V.5 Conclusion
98	CONCLUSION GENERALE
100	ANNEXES
101	ANNEXE A
108	ANNEXE B
110	ANNEXE C
112	ANNEXE D
118	REFERENCES BIBLIOGRAPHIQUES

INTRODUCTION GENERALE

L'utilisation des robots manipulateurs devient de plus en plus fréquente grâce au progrès technologique et au développement des techniques de commandes modernes. Cette évolution fait apparaître l'intérêt technique spécifique de la robotique. En effet, certaines tâches, comportant une part importante de manipulations, ne peuvent être réalisées par un opérateur humain pour des raisons de sécurité ou de contraintes physiques. Dans ces cas, l'utilisation d'un robot reste la seule solution possible. Il en est ainsi pour la manipulation d'objets radioactifs et pour l'exploration spatiale. C'est pourquoi, en plus de son intérêt humain et économique, la recherche en robotique s'intensifie et se diversifie. La robotique s'affirme de jours en jours comme une discipline à part entière avec ses problèmes, ses propriétés et ses lois. Nous proposons d'aborder son aspect modélisation tout au long du premier chapitre.

Le robot manipulateur, étant un dispositif mécanique articulé, est assujéti à des interconnexions dues aux effets qu'exercent les différentes articulations entre elles et à des perturbations dues à son environnement et qui peuvent influencer négativement sur ses performances. Afin de pallier à ces inconvénients, les différentes articulations du robot sont prises comme des sous-systèmes indépendants les uns des autres et à chaque sous-système est appliqué une commande locale. Cette approche, appelée communément commande décentralisée, a fait durant ces dernières années l'objet de plusieurs études et a été appliquée tout particulièrement à la robotique car elle permet de simplifier considérablement les algorithmes de commande et de faciliter l'identification d'éventuels problèmes. L'évolution de la robotique est étroitement liée au développement de techniques de commandes qui ont pour but essentiel l'amélioration de la précision tout en restant robuste vis à vis des perturbations et des changements paramétriques. C'est dans ce contexte que se situe notre travail.

Dans le chapitre II et III, nous présentons les méthodes retenues pour le développement de notre loi de commande car nous avons opté pour une synergie des différentes qualités des méthodes retenues comme c'est le cas de plusieurs études [1] et [2]. Nous avons choisi les réseaux de neurones pour leurs caractéristiques d'approximateur universel [3-5] où ils sont utilisés pour estimer les interconnexions et perturbations, le mode glissant pour sa robustesse [6] et [7] et le Backstepping pour assurer la stabilité du système.

Dans le chapitre IV, nous développons notre loi de commande en utilisant un observateur d'état pour reproduire la valeur des interconnexions et des perturbations qui sont vues comme des variables d'état afin d'atténuer leur effet perturbateur. Une telle représentation est connue sous le nom de système augmenté [8] et [9]. Certains chercheurs utilisent des observateurs pour avoir une variable d'état non bruité [10] et [11] ou pour reproduire des valeurs non mesurables [12]. Ils font appel parfois à des approximateurs universels comme les réseaux de neurones ou la logique floue car certaines valeurs sont non mesurables, voire très difficiles à modéliser.

Nous avons ensuite employé le mode glissant, souvent utilisé pour des modèles incertains non linéaires [13], pour atténuer l'effet des interconnexions et des perturbations qui sont indésirables dans une commande décentralisée [14]. Dans la littérature, nous remarquons souvent que l'inconvénient majeur du mode glissant est l'apparition de ce qu'on appelle l'effet chattering dû à la commutation brutale de la commande. Celle-ci peut exciter des dynamiques négligées lors de la modélisation et peut avoir des conséquences néfastes sur les organes de commandes. L'élimination de cet effet, soit par adoucissement de la commutation [15], soit en utilisant le mode glissant généralisé [9], a fait l'objet de plusieurs travaux. Nous avons opté

pour la deuxième solution, et Enfin, la méthode du Backstepping sera utilisée pour assurer la stabilité de chaque sous-système.

Nous testerons les performances de cette loi de commande par une simulation sur un robot PUMA 560, en procédant à des essais de robustesse et de décentralisation.

Dans le dernier chapitre une étude comparative est faite pour situer notre travail par rapport aux autres lois de commandes classiques et récentes.

CHAPITRE I

MODELISATION D'UN BRAS
DE ROBOT MANIPULATEUR

1.1 Introduction

L'apparition des premiers robots, il y a environ une cinquantaine d'années, a été un réel facteur de progrès technologique dans divers domaines (transport, production industrielle, manutention, levage, assemblage...). Cependant, les premiers robots ont été développés et utilisés par les industries aérospatiale, nucléaire et militaire pour intervenir en milieu hostile (antennes et panneaux solaires, manipulation d'objets radioactifs...).

Les caractéristiques essentielles d'un robot peuvent être résumées par la définition suivante [16] « *un robot est un manipulateur automatique multifonctionnel qui peut s'adapter à l'environnement dans lequel il évolue et qui est destiné à remplacer partiellement ou totalement l'homme dans son action sur l'environnement* ». Aussi, nous pouvons trouver une autre définition du robot et de la robotique [17] « *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 au moins autonome selon les facultés de perception de l'environnement dont est doté le robot. La robotique est l'ensemble des activités de construction et de mise en œuvre de robots* ».

Avant de synthétiser une commande pour un robot manipulateur, il est impératif de connaître avec précision les équations qui régissent le mouvement de ce robot. L'ensemble de ces équations est appelé modèle mathématique, si les équations de ce dernier sont extraites de la physique on a à faire à un modèle de connaissance, et si ces équations découlent des observations sur le système on a alors à faire à un modèle de représentation. Afin d'établir correctement un modèle mathématique, il est nécessaire de transiter par deux étapes essentielles. La première étape est la modélisation cinématique qui est essentiellement une étude de géométrie des déplacements qui décrit les mouvements complexes dans l'espace de l'outil porté par le robot manipulateur. La deuxième étape est la modélisation dynamique qui est un ensemble d'équations différentielles décrivant le comportement dynamique du robot manipulateur.

Le présent chapitre est consacré à expliciter et à détailler les deux étapes citées ci-dessus, afin d'obtenir un modèle mathématique du robot manipulateur. Dans notre travail nous nous intéressons plus particulièrement au modèle du robot PUMA 560.

1.2 Les différentes générations des robots [16]

Une des façons la plus usitée pour classer les robots, est la classification par génération, nous en dénotons trois générations principales.

La première génération comprend des robots dépourvus d'adaptativité mais capable d'exécuter des tâches de façon répétitive. Parmi ces robots nous pouvons citer les modèles « Unimat série 2000 et 2100, Trallsa 3000, Varsatran... », dotés de plusieurs degrés de liberté, ils sont programmables et travaillent en boucle ouverte. Leur programmation se fait suivant un mode d'apprentissage, en faisant exécuter une fois au robot manipulateur le travail demandé, les mouvements seront enregistrés pas à pas et où chaque pas représente une instruction. Le programme ainsi enregistré est exécuté ensuite en mode automatique. L'inconvénient principal de ces robots est leur non adaptativité, car si un quelconque paramètre de leur environnement ou de leur tâche vient à varier, il faudra alors exécuter une nouvelle phase d'apprentissage.

La deuxième génération comprend des robots dotés de fonctions sensorielles (couple, de pression, de proximité...), à la différence de la première génération. Ces robots sont adaptatifs à leur environnement qui peut varier tout en restant prévisible. Le programme initial peut être modifié par le calculateur après analyse des informations récoltées par les capteurs. Parmi ces robots nous trouvons les robots Japonais H-T-Hand et MKI d'Hitachi.

La troisième génération correspond à des robots capables de percevoir leur environnement, de prendre des décisions et d'élaborer des scénarios de commandes. Ces robots font appel à l'intelligence artificielle qui présente plusieurs aspects tels que la reconnaissance des formes, la compréhension du langage naturel, la résolution de problèmes...

1.3 Les différents degrés de liberté

Pour localiser un objet dans l'espace il faut six coordonnées, trois de position pour localiser un point, et trois de rotation pour déterminer l'orientation du corps. Il faut donc utiliser un manipulateur à six degrés de liberté indépendants les uns des autres. De plus si le robot manipulateur doit éviter un obstacle, il doit posséder un degré de liberté supplémentaire.

Plus le degré de liberté d'un manipulateur est élevé, plus il est simple à l'emploi, mais un nombre élevé de degrés de liberté nécessite l'utilisation d'un calculateur puissant ce qui signifie un investissement important et une étude coûteuse, donc lors de la conception d'un robot, l'étude du nombre de degrés de liberté nécessaire est importante. Dans l'industrie les robots manipulateurs utilisés sont dotés de trois à six degrés de liberté.

1.4 Modélisation cinématique

Le modèle cinématique (géométrique) d'un robot manipulateur constitue une représentation mathématique en le considérant comme une structure arborescente comportant p corps rigides, sans masse et articulés entre eux [18]. Chaque corps sera exprimé à travers sa position et son orientation par rapport au déplacement de l'effecteur (organe terminal) d'un point quelconque vers un point désiré caractérisé par un vecteur $x = [x_1, x_2, \dots, x_m]^T$, $m \leq 6$, où x_i représente une coordonnée de position ou d'orientation de l'effecteur dans l'espace \mathcal{R}^m appelé espace de tâche ou espace opérationnel. Il est donc plus naturel d'exprimer le déplacement absolu de l'organe terminal, cependant la commande d'un robot manipulateur est traduite sous forme de couples appliqués à chaque articulation par des actionneurs. Ces derniers confèrent au robot une configuration dans l'espace caractérisée par un vecteur $q = [q_1, q_2, \dots, q_n]^T$, où n est le nombre d'actionneurs ou d'articulations, q_i les variables ou coordonnées articulaires appelées aussi coordonnées généralisées, et \mathcal{R}^n l'espace de configuration.

La modélisation géométrique consiste à établir la relation entre q et x sous la forme $x = f(q)$. Il existe deux méthodes principales, la méthode de Denavit-Hartenberg (annexe A), et la méthode directe.

Pour traiter de la modélisation cinématique, il est nécessaire de maîtriser certains outils géométriques (voir annexe A).

1.4.1 Modèle géométrique du robot PUMA 560

Le robot PUMA 560 est parmi les robots manipulateurs les plus utilisés en industrie. Il réalise six degrés de liberté, tous des rotations (figure I.1).

Les six degrés de liberté sont donc des rotations représentés par les variables q_1, q_2, q_3, q_4, q_5 et q_6 . La première rotation, caractérisée par la variable q_1 , s'effectue autour d'un axe vertical, la deuxième rotation q_2 ainsi que la troisième q_3 s'effectuent autour de deux axes horizontaux. Le poignet englobe les trois dernières rotations identifiées par les variables q_4, q_5 et q_6 .

Afin de pouvoir appliquer les méthodes de modélisation citées en annexe A, et tout particulièrement celle de Denavit-Hartenberg, on admet les hypothèses suivantes :

- Les liaisons du robot manipulateur sont rigides .
- Les jeux dans les articulations sont négligeables .

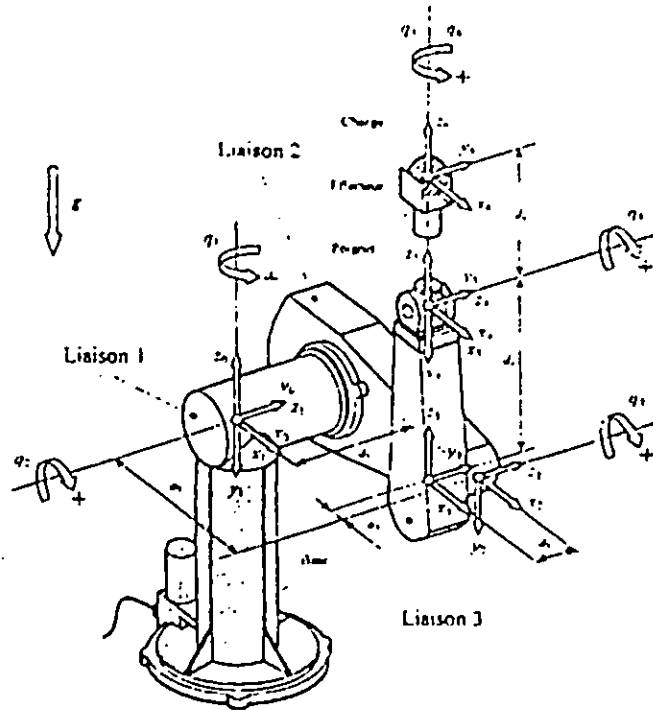


Figure I.1 Position d'origine du PUMA 560 avec les paramètres de Denavit-Hartenberg

Les repères associés à chaque maillon suivant la notation de Denavit-Hartenberg (annexe A) sont donnés comme suit :

- **Liaison 1** : Repère (x_0, y_0, z_0)
 L'origine est prise dans la liaison 1 à l'intersection de l'axe de la base avec l'axe de la liaison 1
 z_0 axe de rotation, $+z_0$ vers le haut
 $+y_0$ coïncide avec l'axe de la liaison 1 et l'axe $+z_1$
 $+x_0$ est en parallèle avec la liaison 2
- **Liaison 2** : Repère (x_1, y_1, z_1)
 L'origine coïncide avec l'origine du repère (x_0, y_0, z_0)
 z_1 axe de rotation, $+z_1$ est perpendiculaire à la liaison 2 et en parallèle à l'axe $+z_2$
 $+y_1$ vers le bas, superposé avec l'axe de la base et en parallèle avec $+y_2$
 $+x_1$ est en parallèle avec la liaison 2

- **Liaison 3** : Repère (x_2, y_2, z_2)
 L'origine est prise dans la liaison 2 à l'intersection de l'axe de la liaison 2 avec l'axe de l'articulation 3
 z_2 axe de rotation, $+z_2$ est perpendiculaire à la liaison 2 et l'axe $+z_3$
 $+y_1$ vers le bas, superposé avec l'axe de la base et en parallèle avec $+y_2$
 $+x_2$ est en parallèle avec la liaison 2
- **Liaison 4** : Repère (x_3, y_3, z_3)
 L'origine est prise dans la liaison 3
 z_3 axe de rotation, $+z_3$ vers le poignet est perpendiculaire à $+z_4$
 $+y_3$ est perpendiculaire à la liaison 2, parallèle avec $+z_4$
 $+x_3$ est en parallèle avec la liaison 2
- **Liaison 5** : Repère (x_4, y_4, z_4)
 L'origine est prise dans le centre du poignet
 z_4 axe de rotation, $+z_4$ est perpendiculaire à la liaison 2 superposé avec $+y_3$
 $+y_4$ opposé à $+z_5$
 $+x_4$ est en parallèle avec la liaison 2
- **Liaison 6** : Repère (x_5, y_5, z_5)
 L'origine coïncide avec l'origine du repère (x_4, y_4, z_4)
 z_5 axe de rotation, $+z_5$ vers le poignet colinéaire avec $+z_6$
 $+y_5$ coïncide avec l'axe de l'articulation 5
 $+x_5$ perpendiculaire à l'axe de l'articulation 5
- **L'effecteur** : Repère (x_6, y_6, z_6)
 L'origine coïncide avec les origines des repères (x_4, y_4, z_4) et (x_5, y_5, z_5)
 $+z_6$ est colinéaire avec $+z_5$
 $+y_6$ est en parallèle avec $+y_5$
 $+x_6$ est en parallèle avec $+x_5$

Les paramètres du robot manipulateur PUMA 560 suivant la convention de Denavit-Hartenberg sont donnés par le tableau suivant:

numéro de la liaison	α_i	θ_i	a_i [m]	d_i [m]
1	-90°	q_1	0	0
2	0°	q_2	0.4318	0.2435
3	+90°	q_3	-0.0203	-0.0934
4	-90°	q_4	0	0.4331
5	+90°	q_5	0	0
6	0°	q_6	0	-

Tableau I.1 : Paramètres de Denavit-Hartenberg du PUMA 560 [19]

L'effecteur (poignet) utilisé diffère d'une application à l'autre, c'est pour cela que la distance d_6 n'est pas représentée, cependant nous la supposons nulle [19].

Nous pouvons simplifier la modélisation du PUMA 560 en négligeant la dynamique des trois dernières rotations identifiées par les variables q_4, q_5 et q_6 ($q_4 = q_5 = q_6 = 0$). Nous nous intéressons donc à l'étude des trois premiers mouvements seulement identifiés par les variables q_1, q_2 et q_3 .

Les mouvements de transformation homogène obtenus à partir du tableau (I.1) sont :

$$T_0^1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.1)$$

$$T_1^2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.2)$$

$$T_0^2 = \begin{bmatrix} c_1 c_2 & -c_1 s_2 & -s_1 & a_2 c_1 c_2 - d_2 s_1 \\ s_1 c_2 & -s_1 s_2 & c_1 & a_2 c_2 s_1 + d_2 c_1 \\ -s_2 & -c_2 & 0 & -a_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.3)$$

$$T_2^3 = \begin{bmatrix} c_3 & 0 & s_3 & a_3 c_3 \\ s_3 & 0 & -c_3 & a_3 s_3 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.4)$$

$$T_0^3 = \begin{bmatrix} c_1 c_{23} & -s_1 & c_1 s_{23} & c_1 (a_2 c_2 + a_3 c_{23}) - (d_2 + d_3) s_1 \\ s_1 c_{23} & c_1 & s_1 s_{23} & s_1 (a_2 c_2 + a_3 c_{23}) + (d_2 + d_3) c_1 \\ -s_{23} & 0 & c_{23} & -(a_2 s_2 + a_3 s_{23}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (I.5)$$

avec : $c_i = \cos \theta_i$, $s_i = \sin \theta_i$, $c_{ij} = \cos (\theta_i + \theta_j)$ et $s_{ij} = \sin (\theta_i + \theta_j)$

D'après les considérations et hypothèses retenues ci-dessus, nous pouvons exprimer les vecteurs des coordonnées de l'effecteur dans le repère R_3 comme suit :

$$r_3^A = \begin{bmatrix} 0 \\ 0 \\ d_4 + d_6 \\ 1 \end{bmatrix} \quad (I.6)$$

Soit le vecteur P des coordonnées de l'effecteur dans le référentiel de la base R_0 :

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \quad (I.7)$$

D'après la matrice de transformation homogène de Denavit-Hartenberg on peut écrire :

$$P = T_0^3 r_3^6 \quad (1.8)$$

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} c_1 c_{23} & -s_1 & c_1 s_{23} & c_1 (a_2 c_2 + a_3 c_{23}) - (d_2 + d_3) s_1 \\ s_1 c_{23} & c_1 & s_1 s_{23} & s_1 (a_2 c_2 + a_3 c_{23}) + (d_2 + d_3) c_1 \\ -s_{23} & 0 & c_{23} & -(a_2 s_2 + a_3 s_{23}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ d_4 + d_6 \\ 1 \end{bmatrix} \quad (1.9)$$

On aura finalement le vecteur des coordonnées de l'organe terminal exprimé dans le référentiel $R_0 (x_0, y_0, z_0)$ lié à la base qui est fixée :

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} c_1 (a_2 c_2 + a_3 c_{23}) - (d_2 + d_3) s_1 + (d_4 + d_6) c_1 s_{23} \\ s_1 ((a_2 c_2 + a_3 c_{23}) + (d_2 + d_3) c_1 + (d_4 + d_6) s_1 s_{23}) \\ (a_2 s_2 + a_3 s_{23}) + (d_4 + d_6) c_{23} \end{bmatrix} \quad (1.10)$$

I.5 Modélisation dynamique

La modélisation dynamique consiste à établir les équations différentielles non linéaires qui relient les efforts actionneurs $\Gamma_i(t)$ aux déplacements articulaires $q_i(t)$, aux vitesses articulaires et aux accélérations articulaires. L'exploitation du modèle dynamique peut se faire de deux manières :

- Résoudre les équations différentielles pour obtenir les mouvements $q_i(t)$ pour des $\Gamma_i(t)$ donnés, et dans le but d'une simulation du comportement dynamique. On parle alors de *modèle dynamique directe*.
- Calculer les efforts $\Gamma_i(t)$ pour des mouvements $q_i(t)$ connus, donnés à l'avance ou mesurés *on-line*. Cette approche constitue le modèle de commande appelé aussi *modèle dynamique inverse*.

L'obtention du modèle dynamique se fait généralement en appliquant l'une des approches de la dynamique : approche de Newton-Euler, approche d'Alembert ou bien les équations d'Euler-Lagrange. Nous nous intéressons à la dernière approche qui est une méthode simple à mettre en œuvre et elle est bien adaptée aux techniques de calcul manuelles, ainsi qu'aux méthodes de calcul assistées par ordinateur (CAO), cette méthode est donnée en annexe A.

I.5.1 Modèle dynamique global d'un robot

Le modèle dynamique d'un robot manipulateur à n degrés de liberté peut être écrit sous forme d'une équation différentielle non linéaire du second ordre, comme suit [20]:

$$M(q)\ddot{q} + B(q, \dot{q})\dot{q} + K(q, \dot{q})q + G(q) + H(\dot{q}) = \Gamma(t) \quad (1.11)$$

ou : $q \in \mathcal{R}^n$, $\dot{q} \in \mathcal{R}^n$, $\ddot{q} \in \mathcal{R}^n$ sont respectivement les vecteurs de position, de vitesse et d'accélération ; $M(q) \in \mathcal{R}^{n \times n}$ est la matrice d'inertie qui est bornée, symétrique et définie positive, un élément m_{ij} de cette matrice représente l'inertie de la $i^{\text{ème}}$ articulation sur la $j^{\text{ème}}$ articulation.

$B(q, \dot{q})\dot{q} + K(q, \dot{q})q$ représente le couple dû aux forces de Coriolis et centrifuges, $G(q)$ est la force et/ou couple dus aux forces de gravitation; $\Gamma(t) \in \mathcal{R}^n$ sont les forces et/ou couples moteurs appliqués au système; $H(\dot{q}) \in \mathcal{R}^n$ correspond aux effets des frottements visqueux.

I.5.2 Modèle dynamique décentralisé d'un robot

Afin de pouvoir synthétiser une commande décentralisée, il est nécessaire d'écrire les équations du robot manipulateur d'une façon décentralisée. Une manière de décentralisation du modèle du robot est de voir chaque degré de liberté comme étant un sous-système. Une autre méthode consiste à prendre les articulations comme étant des sous-systèmes. Nous présentons ci-dessous la première méthode (la deuxième sera donnée au chapitre IV). Ces deux techniques de décentralisation d'un robot sont équivalentes si chaque articulation comporte un seul degré de liberté. Considérons le robot décrit par l'équation matricielle suivante [21]:

$$M(q)\ddot{q} + B(q, \dot{q})\dot{q} + K(q, \dot{q})q + G(q) + H(\dot{q}) = u_m + u_p + u \quad (1.12)$$

avec :

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

$B(q, \dot{q})\dot{q} + K(q, \dot{q})q \in \mathcal{R}^n$ couples dus aux forces de Coriolis et centrifuges

$G(q) \in \mathcal{R}^n$ forces et/ou couples dus aux forces gravitationnelles

$H(\dot{q}) \in \mathcal{R}^n$ vecteur de forces et/ou couples dus aux frottements visqueux et secs

$u \in \mathcal{R}^n$ vecteur force et/ou couple moteur appliqués au système

$u_p \in \mathcal{R}^n$ vecteur des forces et/ou couples des perturbations externes

$u_m \in \mathcal{R}^n$ perturbation due à la portée de la masse

La commande u appliquée au système est un couple appliqué à chaque charge. Le but principal de la loi de commande et de générer u de sorte à ce que la position $q(t)$ suit une trajectoire désirée $q_d(t)$ dans le cas d'une poursuite. Dans le cas d'une régulation c'est la dynamique de $q(t)$ qui est imposée.

A partir des propriétés physiques du robot manipulateur nous pouvons admettre les résultats suivants:

1/ La matrice $M(q)$ est symétrique, bornée et définie positive

2/ Les matrices B et K ainsi que les vecteurs G et H sont bornés.

3/ Le vecteur $H(\dot{q}) = [h_1(\dot{q}_1), \dots, h_n(\dot{q}_n)]$ tel que $i = 1 \dots n$ et

$$h_i(\dot{q}_i) = f_{vi} \dot{q}_i + f_{si} \operatorname{sgn}(\dot{q}_i) \quad (1.13)$$

avec : f_{vi} et f_{si} respectivement les coefficients de frottements visqueux et secs.

4/ La perturbation due à la portée de la charge m correspond à l'effet de la masse portée par le robot. Elle est obtenue à partir de la formule suivante :

$$u_m = m J^T(q) [J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} + g] \quad (1.14)$$

avec :

$$J(q) = \begin{bmatrix} \frac{\partial p_x}{\partial l_1} & \dots & \frac{\partial p_x}{\partial l_n} \\ \frac{\partial p_y}{\partial l_1} & \dots & \frac{\partial p_y}{\partial l_n} \\ \frac{\partial p_z}{\partial l_1} & \dots & \frac{\partial p_z}{\partial l_n} \end{bmatrix} \quad (I.15)$$

L'équation (I.12) peut être écrite d'une manière décentralisée plus développée en l'adaptant à chaque articulation du robot, et en faisant apparaître les interconnexions pour obtenir le modèle dynamique de l'articulation i , on aura donc :

$$u_{pi}(t) + u_i(t) = m_{ii}(q)\ddot{q}_i(t) + \sum_{j=1; j \neq i}^n m_{ij}(q)\ddot{q}_j(t) + b_{ii}(q, \dot{q})\dot{q}_i(t) + \sum_{j=1; j \neq i}^n b_{ij}(q, \dot{q})\dot{q}_j(t) + k_{ii}(q, \dot{q})q_i(t) + \sum_{j=1; j \neq i}^n k_{ij}(q, \dot{q})q_j(t) + g_i(q) + h_i(\dot{q}_i), \quad (I.16)$$

l'indice i réfère au numéro de l'articulation.

Nous pouvons simplifier l'équation précédente en l'écrivant sous la forme finale suivante :

$$m_{ii}(q)\ddot{q}_i(t) + b_{ii}(q, \dot{q})\dot{q}_i(t) + k_{ii}(q, \dot{q})q_i(t) + d_i(q, \dot{q}, \ddot{q}) = u_{pi}(t) + u_i(t) \quad (I.17)$$

Le terme $d_i(\cdot)$ englobe les interconnexions et il est donné par :

$$d_i(q, \dot{q}, \ddot{q}) = g_i(q) + h_i(\dot{q}_i) + \sum_{j=1; j \neq i}^n m_{ij}(q)\ddot{q}_j(t) + b_{ij}(q, \dot{q})\dot{q}_j(t) + k_{ij}(q, \dot{q})q_j(t) \quad (I.18)$$

L'équation (I.17) correspond au modèle dynamique de la i^{me} articulation avec :

- $u_i(t) \in \mathfrak{R}$ commande locale ;
- $q_i(t) \in \mathfrak{R}$, $\dot{q}_i(t) \in \mathfrak{R}$, $\ddot{q}_i(t) \in \mathfrak{R}$ sont respectivement la position, la vitesse et l'accélération linéaire ou angulaire de la i^{me} articulation;
- $m_{ii}(q)$ représente le moment d'inertie local, $m_{ii}(q) > 0$;
- $d_i(q, \dot{q}, \ddot{q}) \in \mathfrak{R}$ représente les interconnexions entre l'articulation i et les autres articulations.

I.5.3 Modèle dynamique du robot PUMA 560 [22]

$M(q)$ (kg.m²):

$$\begin{aligned} m_{11} &\approx 2.57 + 1.38 c_2^2 + 0.30 s_2 s_3 + 7.44 \cdot 0.1 c_2 s_{23} \\ m_{12} = m_{21} &\approx 6.90 \cdot 10^{-1} s_2 - 1.34 \cdot 10^{-1} c_{23} + 2.38 \cdot 10^{-2} c_2 \\ m_{13} = m_{31} &\approx -1.34 \cdot 10^{-1} c_{23} - 3.97 \cdot 10^{-3} s_{23} \\ m_{22} &\approx 6.79 + 7.44 \cdot 10^{-1} s_3 \\ m_{23} = m_{32} &\approx 0.333 + 3.72 \cdot 0.1 s_3 - 1.10 \cdot 10^{-2} c_3 \\ m_{33} &\approx 1.16 \end{aligned}$$

$B(q, \dot{q})$:

$$\begin{aligned} b_{11} &\approx (-2.76 s_2 c_2 + 7.44 \cdot 10^{-1} c_{223} + 0.60 s_2 c_3 - 2.13 \cdot 10^{-2} (1 - 2 s_2 s_3)) \dot{q}_2 \\ b_{12} &\approx (6.90 \cdot 10^{-1} c_2 + 1.34 \cdot 0.1 s_{23} - 2.38 \cdot 10^{-2} s_2) \dot{q}_2 + (2.67 \cdot 10^{-1} s_{23} - 7.58 \cdot 10^{-3} c_{23}) \dot{q}_3 \\ b_{13} &\approx (7.44 \cdot 10^{-1} c_2 c_{23} + 0.60 s_2 c_3 + 2.20 \cdot 10^{-2} c_2 s_{23} - 2.13 \cdot 10^{-2} (1 - 2 s_2 s_3)) \dot{q}_1 \\ b_{21} &\approx -0.5 (-2.76 s_2 c_2 + 7.44 \cdot 10^{-1} c_{223} + 0.60 s_2 c_3 - 2.13 \cdot 10^{-2} (1 - 2 s_2 s_3)) \dot{q}_1 \\ b_{22} &\approx (2.20 \cdot 10^{-2} s_3 + 7.44 \cdot 10^{-1} s_3) \dot{q}_3 \\ b_{23} &\approx 0.5 (2.20 \cdot 10^{-2} s_3 + 7.44 \cdot 10^{-1} s_3) \dot{q}_3 \\ b_{31} &\approx -0.5 (7.44 \cdot 10^{-1} c_2 c_{23} + 0.60 s_2 c_3 + 2.20 \cdot 10^{-2} c_2 s_{23} - 2.13 \cdot 10^{-2} (1 - 2 s_2 s_3)) \dot{q}_1 \\ b_{32} &\approx -0.5 (2.20 \cdot 10^{-2} s_3 + 7.44 \cdot 10^{-1} s_3) \dot{q}_2 \\ b_{33} &\approx 0 \end{aligned}$$

$G(q)$:

$$\begin{aligned} g_1 &= 0 \\ g_2 &\approx -37.2 c_2 - 8.4 s_{23} + 1.02 s_2 \\ g_3 &\approx -8.4 s_{23} + 0.25 c_{23} \end{aligned}$$

$$g^T = [0 \ 0 \ 9.81].$$

$J(q)$:

$$\begin{aligned} j_{11} &= -s_1 (a_2 c_2 + a_3 c_{23}) - (d_2 + d_3) C_1 - d_4 s_1 s_{23} \\ j_{12} &= -c_1 (a_2 s_2 + a_3 s_{23}) + d_4 c_1 c_{23} \\ j_{13} &= -a_3 c_1 s_{23} + d_4 c_1 c_{23} \\ j_{21} &= c_1 (a_2 c_2 + a_3 c_{23}) - (d_2 + d_3) s_1 + d_4 c_1 s_{23} \\ j_{22} &= -s_1 (a_2 s_2 + a_3 s_{23}) + d_4 s_1 c_{23} \\ j_{23} &= -a_3 s_1 s_{23} + d_4 s_1 c_{23} \\ j_{31} &= 0 \\ j_{32} &= -(a_2 c_2 + a_3 c_{23}) - d_4 s_{23} \\ j_{33} &= -a_3 c_{23} - d_4 s_{23} \end{aligned}$$

où $a_2 = 0,4319$ m, $a_3 = -0,0203$ m, $d_2 = 0,2435$ m, $d_3 = -0,0934$ m et $d_4 = 0,4331$.

c_i, s_i, c_{ij}, s_{ij} et c_{ijk}, s_{ijk} dénotent respectivement $\cos(q_i)$, $\sin(q_i)$, $\cos(q_i + q_j)$, $\sin(q_i + q_j)$ et $\cos(q_i + q_j + q_k)$,

1.6 Essai à vide du robot PUMA 560

L'essai à vide représente la réponse en position du robot sans aucune charge, à une entrée échelon ($u_1 = u_2 = u_3 = 1$). Les réponses obtenues sont représentées en figure 1.2 :

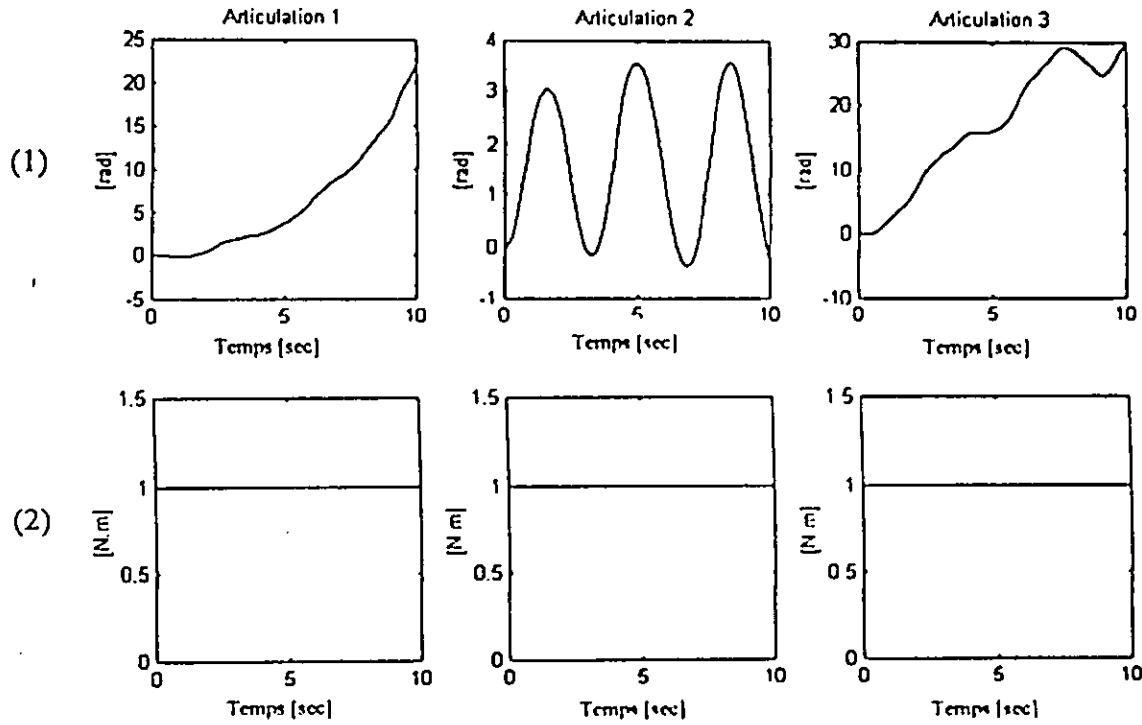


Figure 1.2 Essai à vide

– En ordonnées, pour chaque articulation i : (1): q_i , (2): u_i .

1.7 Conclusion

A partir des équations d'Euler-Lagrange, nous avons pu élaborer le modèle dynamique du robot manipulateur PUMA 560. Notre travail se base essentiellement sur ce robot qui suscite une large utilisation dans le domaine industriel. Nous avons exposé les résultats de simulation du PUMA 560 à vide pour montrer les interconnexions existantes entre ses articulations.

Afin de pouvoir exciter toutes les dynamiques du robot manipulateur PUMA 560, nous avons opté pour une trajectoire cycloïdale test proposée par LEAHVY.

CHAPITRE II

OBSERVATEUR PAR RESEAU DE NEURONES

II.1 Introduction

La commande par retour d'état prend une ampleur considérable par rapport au réglage classique (retour de sortie), car elle permet d'obtenir de très bons résultats vis-à-vis de la grandeur de consigne et de la perturbation. De plus, nous pouvons agir sur un large domaine de comportement dynamique [23] voire même l'influence sur les variables d'état, ce qui permet leur limitation et évite donc la divergence des grandeurs internes du système. Mais le réglage par retour d'état a besoin impérativement de toutes les variables d'état afin de pouvoir les mettre en contre réaction, en vue d'imposer une certaine dynamique au système (commande optimale, découplage, placement de pôles...). Comme dans la plupart des cas, les seules grandeurs accessibles du système sont les variables d'entrées sorties, soit à cause de la non accessibilité à ces variables d'état (non existence d'un capteur qui permet leur mesure), soit parce qu'elles sont purement mathématiques (non existence de grandeurs physiques représentant ces variables d'état). Il est nécessaire, à partir de ces informations, de construire l'état du modèle choisi pour élaborer la commande. Un reconstruteur d'état est un système ayant comme entrée les entrées et les sorties du système réel et dont la sortie est une estimation de l'état de ce système (figure II.1).

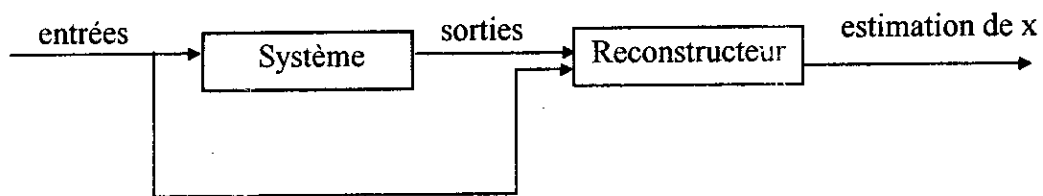


Figure II.1 Principe d'un estimateur

Dans le cas où le modèle d'état est un modèle déterministe, le reconstruteur d'état sera appelé *observateur*, dans le cas où le modèle est stochastique, nous parlerons alors de *filtre*.

L'observateur d'état est un système dynamique qui permet de reproduire les variables d'état non mesurables seulement par connaissance des valeurs d'entrées et de sorties du système, dans le cas discret l'observateur n'est plus vu comme un système dynamique mais comme étant un algorithme. L'observateur d'état ne peut reproduire fidèlement les variables d'état mais seulement leur estimation (figure II.1) qui comporte une erreur devant tendre vers zéro. Quand cette propriété est satisfaite, l'observateur est dit *asymptotique*.

Dans le présent chapitre nous présentons les observateurs linéaires puis quelques types d'observateurs non linéaires. Nous nous intéressons à l'observateur par réseau de neurones en particulier car nous l'utilisons dans la synthèse de la commande comme étant un observateur des interconnexions et des perturbations, pour cette raison nous donnons un rappel sur les réseaux de neurones avant d'aborder les observateurs.

II.2 Réseaux de neurones

Notre objectif est de commander un système non linéaire fortement interconnecté sans la connaissance du modèle exact mais seulement des parties non interconnectées. Nous ferons alors appel, plus loin, à un approximateur des interconnexions, des perturbations et des erreurs de modélisation, dans le but de simplifier le calcul de la commande. En effet, en plus

de la difficulté à modéliser des systèmes pareils, beaucoup d'erreurs subsistent, dues notamment à la non-maîtrise de tous les phénomènes physiques.

Une des plus importantes caractéristiques des réseaux de neurones est d'être des approximateurs universels et ceci nous a mené à les utiliser pour approximer les interconnexions, les perturbations et les erreurs de modélisation.

Dans ce qui suit, nous présentons les réseaux de neurones et les méthodes d'apprentissage largement utilisés.

II.2.1 Présentation et historique des réseaux de neurones [24] [25]

Chez les êtres vivants, chaque mouvement, volontaire ou non, demande une parfaite régulation. De plus, si on les considère comme étant des systèmes, ils sont si complexes par leurs non-linéarités et leurs interconnexions. Faire réaliser de telles opérations par une machine, même moins complexe, est un grand défi. Sachant que toutes ces tâches sont commandées par des neurones explique l'intérêt marqué par l'ingénieur aux neurones biologiques.

Historique

Le champ des réseaux neuronaux a démarré par la présentation en 1943 par W. Mac Culloch et W. Pitts du neurone formel qui est une abstraction du neurone physiologique, dont le retentissement a été énorme. Par cette présentation, ils veulent démontrer que le cerveau est équivalent à une machine de Turing. La pensée devient alors purement des mécanismes matériels et logiques. Ils déclarent en 1955 : "Plus nous apprenons de choses au sujet des organismes, plus nous sommes amenés à conclure qu'ils ne sont pas simplement analogues aux machines, mais qu'ils sont machine." [Mysterium Iniquitatis of Sinful Man Aspiring into the Place of God, repris in Embodiments of mind]. La démonstration de Mac Culloch et Pitts est un des facteurs importants de la création de la cybernétique.

En 1949, D. Hebb présente dans son ouvrage "The Organization of Behavior" une règle d'apprentissage. De nombreux modèles de réseaux aujourd'hui s'inspirent encore de la règle de Hebb. Se basant sur cette règle, en 1958, F. Rosenblatt développe le modèle du Perceptron. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception et une autre liée à la prise de décision. C'est le premier système artificiel capable d'apprendre par expérience.

En 1969, M. Minsky et S. Papert publient une critique des propriétés du Perceptron qui a eu une grande incidence sur la recherche dans ce domaine et qui va fortement diminuer jusqu'à 1980. Néanmoins, quelques études ont été réalisées sur la mémoire associative, qui est une méthode d'apprentissage non supervisée. S. Amari, C. Anderson et T. Kohonen ont proposé des applications à la reconnaissance de formes, mais il y a eu des problèmes d'instabilité et de concepts liés aux systèmes dynamiques. De plus, les travaux de J. Hapfied et M. A. Cohen & S. Grossberg ont définitivement réglé le problème de stabilité des réseaux de neurones dynamiques.

Parmi les germes qui ont contribué d'une manière frappante à la renaissance des travaux dans les années 80, il y a ceux de Widrow et Hoff sur les algorithmes adaptatifs ou modèle de l'Adaline (ADaptive LINar Element).

Le plus grand pas dans les réseaux de neurones, on le doit aux résultats d'un groupe de chercheurs de l'Université de Stanford : Rumelhart, Hinton et Williams en 1986. Il s'agit de l'algorithme dit rétropropagation ou "Backpropagation".

Aujourd'hui, les réseaux neuronaux sont utilisés dans de nombreux domaines tels que l'aide à la décision, la planification, la compression de données, la prévision météorologique et la reconnaissance de formes. Ils sont également employés par les cognitivistes et les neurobiologistes pour simuler le comportement humain dans les domaines des vie et de l'intelligence artificielle, à cause de leurs propriétés, dont leur capacité d'apprentissage, et parce qu'il sont des systèmes dynamiques. L'avantage des réseaux de neurones est qu'ils acceptent des données incomplètes, incertaines ou bruitées. Ils présentent, de plus, une grande robustesse face aux défaillances techniques et s'enrichissent de leurs expériences. Par contre, leurs architectures parallèles nécessitent l'emploi de processeurs spécialisés pour accélérer les calculs. Il est nécessaire de passer par la phase d'apprentissage avant d'utiliser le réseau. Enfin, il ne faut pas penser qu'un réseau de neurone est capable d'égaliser un cerveau humain, dont la structure est encore trop peu connue pour qu'on puisse l'imiter.

II.2.2 Le modèle biologique [26] [27]

Le neurone, au sens biologique, est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. Celles-ci sont parfois si nombreuses que l'on parle alors de chevelure dendritique ou d'arborisation dendritique. Les neurones communiquent entre eux en émettant des signaux électriques. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone. Chaque neurone effectue une sommation des signaux qui lui parviennent par les dendrites. L'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. En fait, il existe un espace intercellulaire de quelques nanomètres entre l'axone du neurone afférent et les dendrites du neurone efférent. La jonction entre deux neurones est appelée la synapse, comme indiqué à la figure II.2. Il existe deux types de synapses : les synapses excitatrices qui renvoient un signal de sortie proportionnel au signal d'entrée, et les synapses inhibitrices qui retournent un signal inversement proportionnel au signal d'entrée.

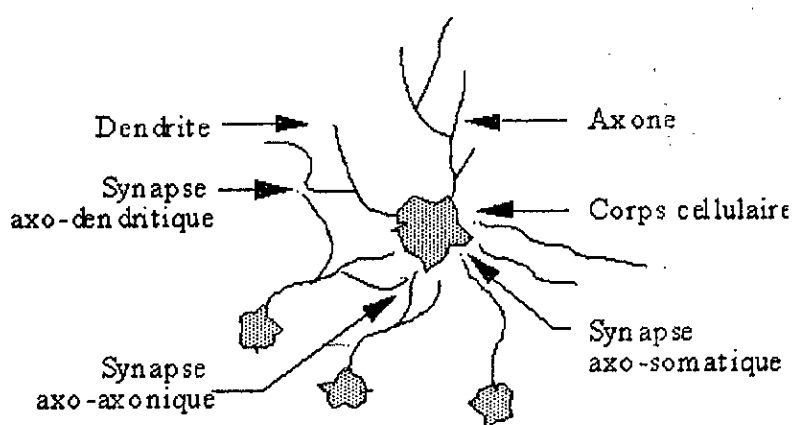


Figure II.2 Neurone avec son arborisation dendritique

Selon le type du neurone (figure II.3), la longueur de l'axone peut varier de quelques microns à 1,50 mètres pour un moto-neurone. De même, les dendrites mesurent de quelques microns à 1,50 mètres pour un neurone sensoriel de la moelle épinière. Le nombre de synapses par neurone varie aussi considérablement de plusieurs centaines à une dizaine de milliers.

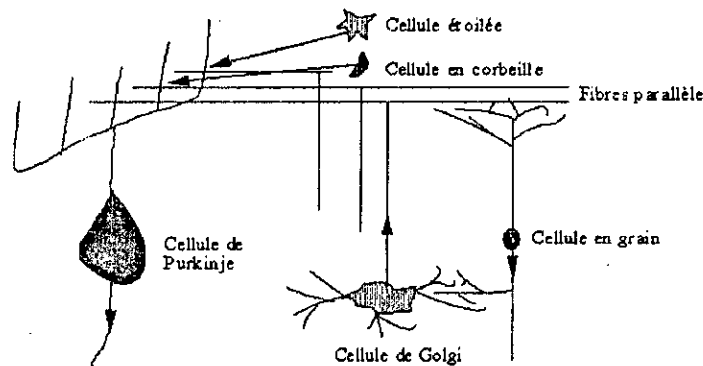


Figure II.3 Description schématique des divers types structuraux de neurones présents dans le cortex cérébelleux (les axones sont repérés par une flèche)

II.2.3 Les modèles mathématiques [24] [28] [29]

Les réseaux de neurones biologiques réalisent facilement un certain nombre d'applications telles que la reconnaissance de formes, le traitement du signal, l'apprentissage par l'exemple, la mémorisation, la généralisation. Ces applications sont pourtant, malgré tous les efforts déployés en algorithmique et en intelligence artificielle, à la limite des possibilités actuelles. C'est à partir de l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base du cerveau que les réseaux de neurones artificiels se sont développés. Ils sont des modèles et, à ce titre, peuvent être décrits par leurs composants, leurs variables descriptives et les interactions des composants.

Le neurone formel

Le modèle du neurone formel, conçu par Mac Culloch et Pitts, est un modèle mathématique très simple dérivé d'une analyse de la réalité biologique (figure II.4). C'est dans l'article "A logical Calculus of the ideas immanent in nervous activity" paru dans le Bulletin of mathematical, en 1943, qu'est apparue pour la première fois la notion de neurone formel.

A l'époque, Mac Culloch étudie la strychnisation du cerveau des chimpanzés. Il se convainc que les circuits d'influx nerveux sont analogues à la propagation logique d'un raisonnement.

Il en déduit la notion de psychon, événement psychique, qui implique son antécédent temporel et intervient dans la détermination des psychons subséquents.

Avec l'aide de Pitts, il présente le neurone formel (figure II.5):

- Il est binaire : soit actif, soit inactif.
- Chaque neurone possède un seuil de déclenchement fixé.
- Le neurone reçoit des entrées de synapse excitatrices, ayant des poids identiques.
- Le neurone peut aussi recevoir des entrées de synapses inhibitrices, dont l'action est absolue ; c'est à dire que, si la synapse est active, le neurone ne peut pas être actif.

- Un quantum de temps est nécessaire à l'intégration des entrées synaptiques, basé sur le délai observé physiologiquement.

On constate que le modèle biologique fait intervenir une notion temporelle qui est difficile à intégrer dans un modèle simple. On remplace donc l'intégration temporelle par une sommation des signaux arrivant au neurone. On effectue en fait une somme pondérée. Le fonctionnement est simple si aucune synapse inhibitrice n'est active. Le neurone ajoute ses entrées synaptiques et vérifie que la somme atteint ou excède son seuil. On dit alors que le neurone devient actif sinon, il est inactif.

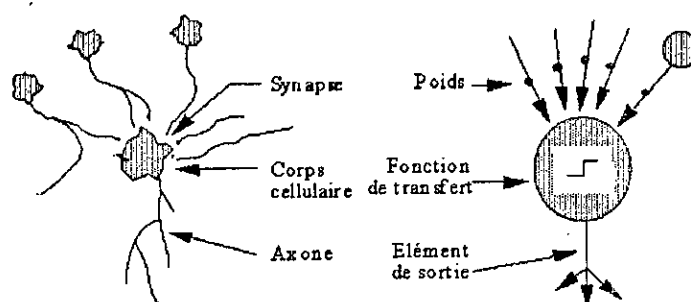


Figure II.4 Mise en correspondance neurone biologique / neurone artificiel

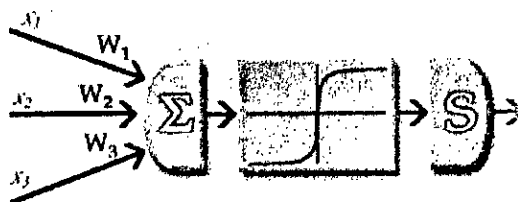


Figure II.5 le neurone formel

On distingue deux phases. La première est habituellement le calcul de la somme pondérée (p) des entrées. Le neurone reçoit les entrées, ou excitations, $x_1, x_2, x_3 \dots x_n$, dont les coefficients synaptiques ou poids synaptiques sont $W_1, W_2, W_3 \dots W_n$. De nos jours, nous utilisons des poids différents pour $W_1, W_2, W_3 \dots W_n$. Les poids peuvent même être corrigés dans le cas de l'apprentissage. Le potentiel p est donné par $p = x_1 \times W_1 + x_2 \times W_2 + x_3 \times W_3 + \dots + x_n \times W_n$.

Dans la deuxième phase, une fonction de transfert calcule, à partir de cette donnée, la valeur de l'état du neurone où $s = f(p)$, $f(.)$ étant la fonction de transfert. Il existe de nombreuses formes possibles pour la fonction de transfert. Les plus courantes sont reportées à la figure II.6. Contrairement aux neurones biologiques et celui créé par McCulloch et Pitts qui n'utilisent qu'une fonction binaire (figure II.6(a)), la plupart des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprises dans l'intervalle $[0, 1]$ ou $[-1, 1]$.

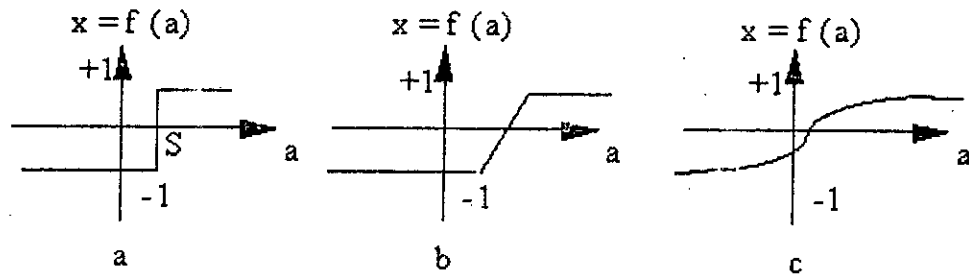


Figure II.6 Différents types de fonctions de transfert pour le neurone artificiel, a : fonction à seuil (S : valeur du seuil), b : linéaire par morceaux, c : sigmoïde.

Cependant il existe une autre fonction d'activation qui n'est pas monotone, c'est la fonction gaussienne (figure II.7) qui se distingue par une non-linéarité et un comportement impulsionnel. Son efficacité réside dans son comportement local, ce qui rend ses réponses différentes dans chaque zone et cela dépend de la distance entre l'entrée et le centre de la fonction.

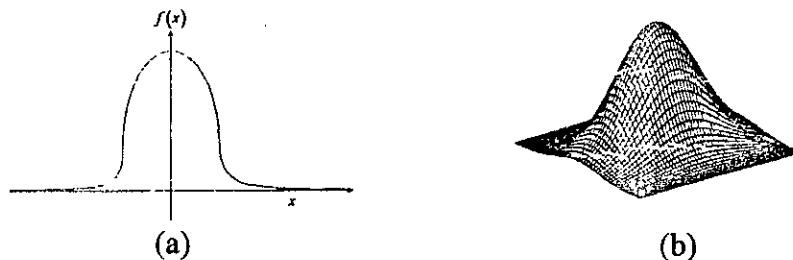


Figure II.7 Fonction gaussienne
Une seule entrée (a). Deux entrées (b).

Nous constatons que les équations, décrivant le comportement des neurones artificiels, n'introduisent pas la notion de temps. En effet, et c'est le cas pour la plupart des modèles actuels de réseaux de neurones, nous avons à faire à des modèles à temps discret et synchrone, dont le comportement des composants ne varie pas dans le temps.

Exemple (multiplication)

Voici un petit exemple simple d'un réseau neuronal. Il permet de calculer le produit de deux nombres.

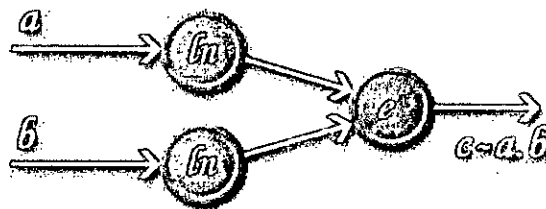


Figure II.8 Multiplication neuronale

Les deux premiers neurones ont comme fonction de transfert la fonction logarithme. Ensuite $\ln(a)$ et $\ln(b)$ sont sommés en entrées du troisième neurone. La fonction de transfert est alors l'exponentiel. La sortie donne le produit $a \times b$.

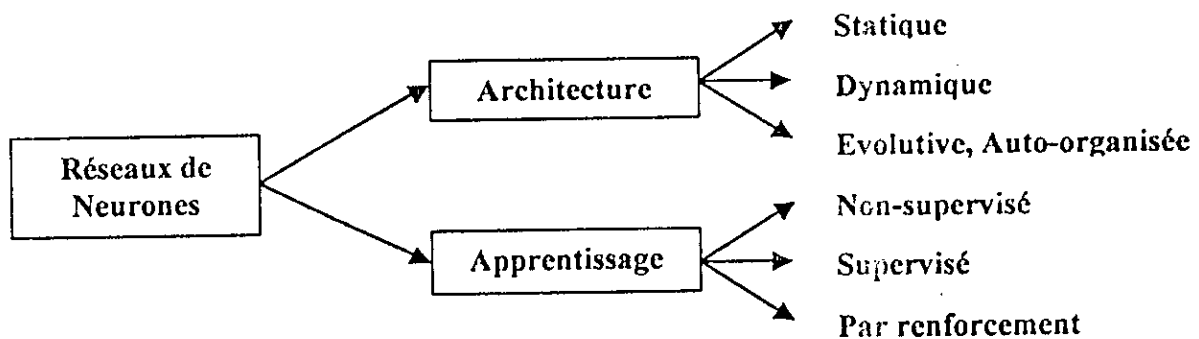
On peut remarquer que les calculs sont parallèles et ne sont plus linéaires (entre les deux premiers neurones).

II.2.4 Structure des réseaux de neurones [5] [24] [30]

Nous avons présenté le neurone qui n'est qu'une unité élémentaire de tout un réseau, car l'efficacité du cerveau réside dans le regroupement des neurones. Chaque neurone reçoit et émet de l'information, pour que le réseau puisse remplir la tâche qui lui est définie. Dans un réseau, les neurones sont regroupés en plusieurs couches. Les données d'entrée traversent d'abord la première couche, puis les réponses de cette première couche sont renvoyées à la seconde, et ainsi de suite. Les réseaux les plus simples sont constitués d'une ou de deux couches. Ils sont très rapides mais ne peuvent résoudre que des problèmes linéaires. Les réseaux à plus de deux couches peuvent théoriquement résoudre tous les problèmes, mais leur manipulation est difficile. Leur phase d'apprentissage s'effectue différemment : il est nécessaire d'utiliser un algorithme dit de rétropropagation qui corrige les poids des connexions des différentes couches en fonction des erreurs commises.

Classification

En sachant que les réseaux de neurones n'ont fait qu'essayer de copier le concept déjà présent chez les êtres vivants, d'où les multitudes d'approches faites au niveau de la réalisation des réseaux de neurones. Mais, cette diversité de réseaux peut être classée comme suit :



La fonction réalisée par le réseau de neurones est l'élément déterminant du choix du type de réseau à utiliser.

Architecture

L'architecture décrit le comportement du réseau de neurones. Cette dernière est définie par les connexions entre les neurones et le sens du transfert de l'information entre les couches. On note qu'on peut fonctionnellement partager un réseau complexe en plusieurs couches, et qu'il est souvent possible de distinguer une certaine régularité dans la topologie des connexions entre les neurones.

Réseau multicouche (au singulier) : les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches avales (figure II.9). Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et à celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc de définir les concepts de neurone d'entrée et neurone de sortie. Par extension, on appelle couche d'entrée, l'ensemble des neurones d'entrée, et couche de sortie, l'ensemble des

neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées.

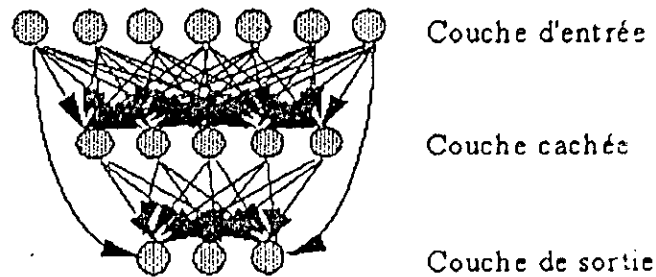


Figure II.9 Définition des couches d'un réseau multicouche.

Réseau à connexions locales : Il s'agit d'une structure multicouche mais qui, à l'image de la rétine, conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avale (figure II.10). Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique.

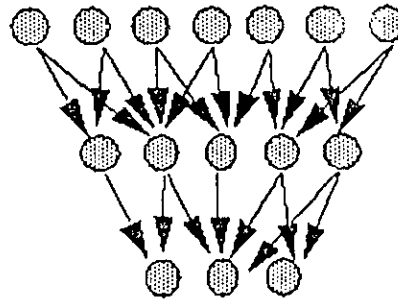


Figure II.10 Réseau à connexions locales

Réseau à connexions récurrentes : Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche. Ces connexions sont le plus souvent locales (figure II.11).

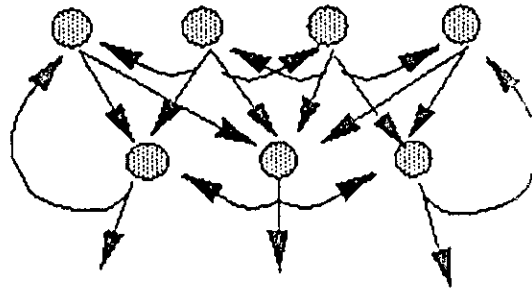


Figure II.11 Réseau à connexions récurrentes

Réseau à connexions complètes : C'est la structure d'interconnexion la plus générale (figure II.12). Chaque neurone est connecté à tous les neurones du réseau et à lui-même.

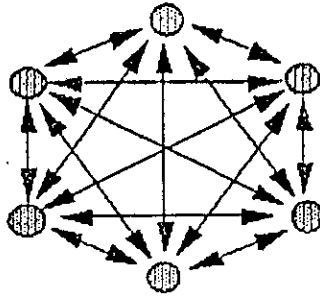


Figure II.12 Réseau à connexions complètes

Dans l'identification des systèmes, dans le but de les commander, nous utilisons comme base le multicouche classique (Feed Forward) (figure II.13) pour les systèmes statiques où nous avons pris comme exemple un réseau de neurones à trois couches. Par contre, pour les systèmes dynamiques, nous utilisons les réseaux à connexions récurrentes. Ce retour d'information est présenté par le retard z^{-1} , injecté au système (figure II.14). En général, les systèmes identifiés sont composés des opérations élémentaires suivantes :

- i) Retard (système Dynamique),
- ii) Sommation,
- iii) Fonction non-linéaire.

L'opérateur retard z^{-1} est remplacé par un intégrateur dans le cas continu. Dans certains cas, la multiplication par une constante est envisagée. Certains réseaux ont un retour non linéaire.

On peut générer les fonctions des systèmes qui soient dynamiques ou statiques par les deux fonctions N_i {Fonction non-linéaire, définie comme suit : $N_i = \Gamma_i[W_i]$ avec Γ_i : fonction d'activation des neurones et W_i : pondération} et $V_i(z)$ fonction linéaire de z qui représente le retard. La figure II.15 présente les différentes configurations de base existantes où le système n'est qu'une mise en série d'une ou de toutes ces configurations.

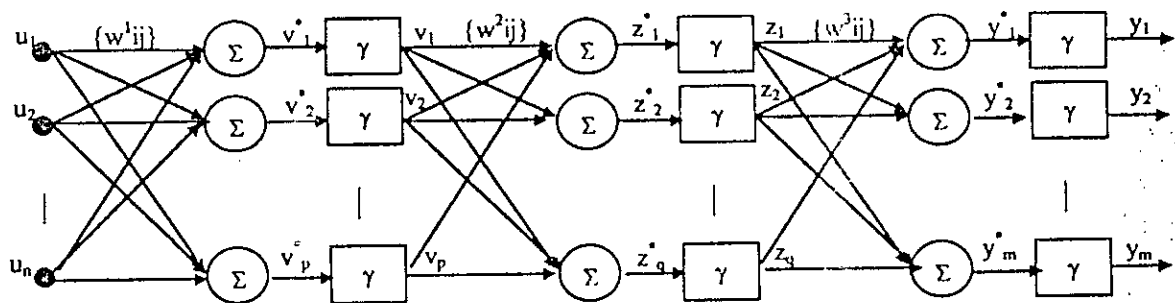


Figure II.13 Réseau de neurones statiques à trois couches

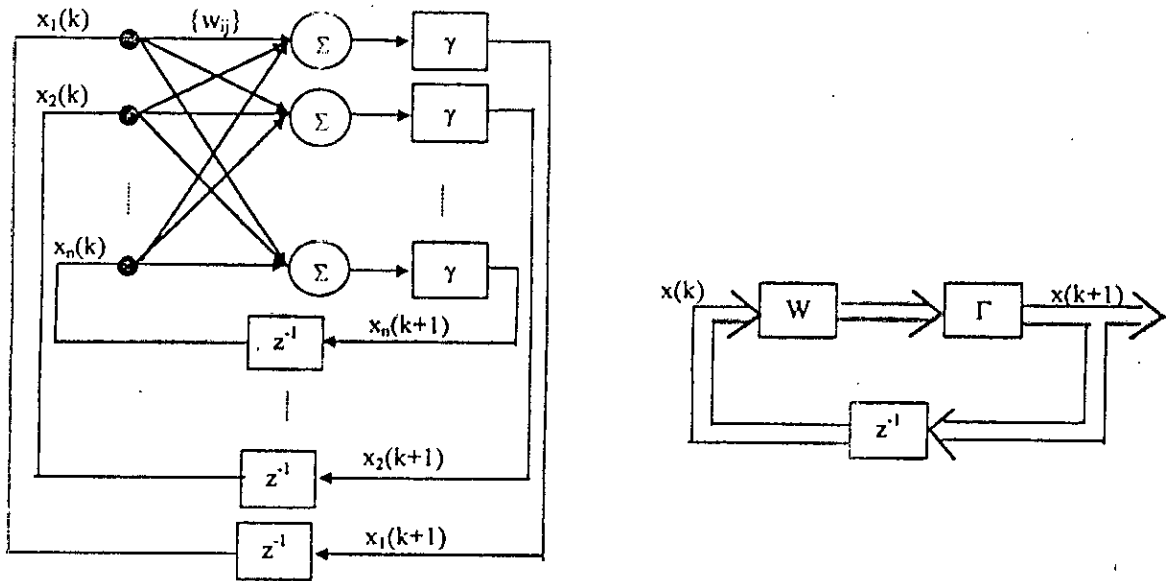


Figure II.14 Réseau de neurones récurrents

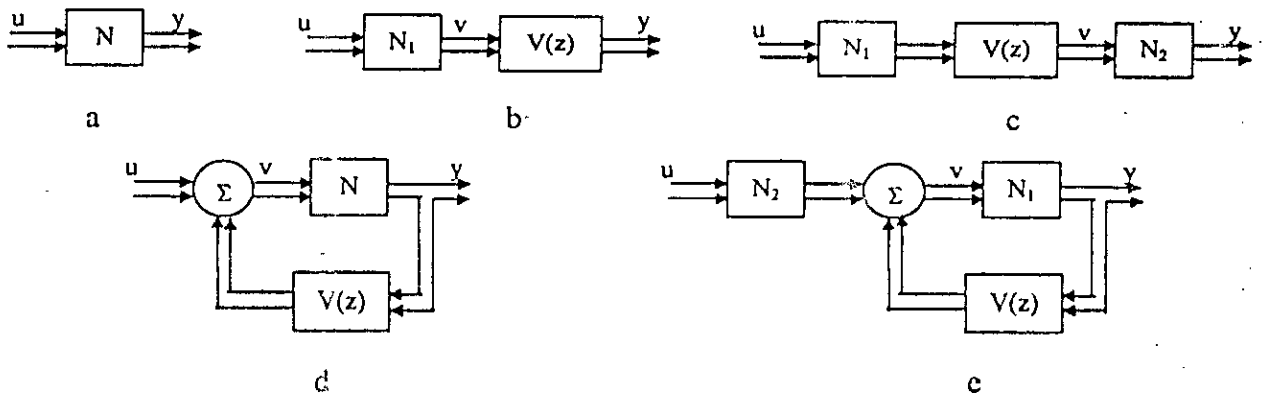


Figure II.15 Différentes configurations de base en bloc diagramme

II.2.5 Apprentissage [24] [30]

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux de neurones. Cependant, elle ne concerne pas tous les modèles, mais les plus utilisés. La phase d'apprentissage dépend beaucoup de la structure du réseau.

Définition :

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention d'un comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement.

Dans le cas des réseaux de neurones artificiels, on ajoute souvent à la description du modèle l'algorithme d'apprentissage. Le modèle sans apprentissage présente en effet peu d'intérêt. Dans la majorité des algorithmes actuels, les variables modifiées pendant l'apprentissage sont les poids des connexions et le seuil des neurones. L'apprentissage est la modification des poids du réseau et du seuil des neurones dans l'optique d'accorder la réponse du réseau aux exemples et à l'expérience. Il est souvent impossible de décider a priori des

valeurs des poids des connexions d'un réseau pour une application donnée. A l'issue de l'apprentissage, les poids sont fixés : c'est alors la phase d'utilisation. Certains modèles de réseaux sont improprement dénommés réseaux à apprentissage permanent. Tel est le cas dans la commande adaptative où il est vrai que l'apprentissage ne s'arrête jamais. Cependant, on peut toujours distinguer une phase d'apprentissage, en fait de remise à jour du comportement, et une phase d'utilisation. Cette technique permet de conserver au réseau un comportement adapté malgré les fluctuations dans les données d'entrées.

Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé. Cette distinction repose sur la forme des exemples d'apprentissage. Dans le cas de l'apprentissage supervisé, les exemples sont des couples (Entrée, Sortie associée). On effectue alors la modification des poids pour retrouver cette sortie imposée alors que l'on ne dispose que des valeurs (Entrée) pour l'apprentissage non supervisé. Les réseaux à apprentissage non-supervisé sont capables de "mémorisation", c'est à dire qu'ils raisonnent par analogie avec ce qu'ils ont déjà effectué. On distingue deux types de mémoires :

- Les mémoires hétéro-associatives
- Les mémoires auto-associatives

Nous remarquons cependant que les modèles à apprentissage non supervisé nécessitent, avant la phase d'utilisation, une étape de « labelisation » effectuée par l'opérateur, qui n'est autre chose qu'une part de supervision. Enfin, certains réseaux associent les deux types d'apprentissage.

II.2.6 Algorithme de rétropropagation (Back-Propagation) [5]

En identifiant un système, qu'il soit statique ou dynamique, et en utilisant les réseaux de neurones comme modèle, notre objectif est de trouver un algorithme adaptatif où son rôle est d'ajuster les paramètres du réseau à partir de l'ensemble formé des paires entrée-sortie. Si les poids du réseau sont les seuls paramètres à ajuster, on construit alors le vecteur θ tel que le processus d'apprentissage entraîne la détermination du vecteur θ^* qui optimise la fonction coût J basée sur l'erreur de sortie. Rétropropagation est une des plus importantes méthodes d'apprentissage supervisé. Dans le cas d'un système statique comme celui de la figure II.12, on calcule le gradient de la fonction coût par rapport aux vecteurs θ , notés $\nabla_{\theta} J$ (vecteurs des dérivés partielles de J par rapport à θ), puis nous ajustons le vecteur θ par l'algorithme suivant :

$$\theta_{i+1} = \theta_i - \eta \nabla_{\theta} J \quad (\text{II.1})$$

Où i représente le résultat de la $i^{\text{ème}}$ itération. η est le gain d'adaptation. Ce dernier a une très grande importance dans la stabilité de l'algorithme car il détermine sa vitesse de convergence. Pour cela, des méthodes ont été développées pour le calcul du gain. L'algorithme résultant est dit algorithme du gradient amélioré. Un des problèmes du gradient est qu'il a la possibilité de tomber sur un minima local. Pour pallier à ce problème, des modifications ont été apportées [24].

Pour les systèmes dynamiques la différence apparaît au niveau du calcul du gradient. Dans la plus part des cas $J = \sum \|e\|^2$ avec s l'ensemble des paires entrées-sorties et $e = y - y_d$ (y : vecteur de sortie du réseau, y_d : vecteur de sortie désiré), alors le calcul du gradient revient à

calculer $\frac{\partial e}{\partial \theta_j}$ (j représente le $j^{\text{ème}}$ élément du vecteur θ) qui n'est que le calcul de $\frac{\partial y}{\partial \theta_j}$ car y_d est constant.

Nous donnons ci dessous $\frac{\partial y}{\partial \theta_j}$ pour les différentes représentations des systèmes dynamiques en figure II.15 :

$$b: \frac{\partial y}{\partial \theta_j} = V(z) \frac{\partial v}{\partial \theta_j} \quad \frac{\partial v}{\partial \theta_j} \text{ se calcule comme dans le cas d'un système statique}$$

l : nombre de sorties du bloc $V(z)$

$$c: \frac{\partial y_i}{\partial \theta_j} = \sum_l V(z) \frac{\partial y_i}{\partial v_l} \frac{\partial v_l}{\partial \theta_j} \quad \frac{\partial y_i}{\partial v_l} \text{ se calcule comme dans le cas d'un système statique}$$

$\frac{\partial v_l}{\partial \theta_j}$ s'obtient comme dans b

(II.2)

$$d: \frac{\partial y}{\partial \theta_j} = \frac{\partial N(v)}{\partial v} V(z) \frac{\partial y}{\partial \theta_j} + \frac{\partial N(v)}{\partial \theta_j}$$

$$e: \frac{\partial y}{\partial \theta_j} = \frac{\partial N^1(v)}{\partial v} \left[\frac{\partial N^2(u)}{\partial \theta_j} + V(z) \frac{\partial y}{\partial \theta_j} \right]$$

En annexe B, nous donnons le diagramme de rétropropagation qui n'est qu'une autre interprétation de l'algorithme.

Remarque : Nous n'avons pas présenté les réseaux de neurones évolutifs car ils ne nous intéressent pas dans ce travail.

II.2.7 Conclusion

Nous avons essayé de présenter, de manière générale, les différents types d'architectures de réseaux de neurones avec les méthodes d'apprentissage appropriées, tout en mettant en évidence la propriété d'approximateur universel qui fait tout l'intérêt des réseaux de neurones. Cette caractéristique n'est pas attribuée à tous les réseaux pour n'importe quel système. Mais, à chaque système, on lui associe un type de réseaux de neurones qui est le plus adéquat à sa représentation. Comme nous l'avons déjà vu, un système dynamique doit être représenté par un réseau dynamique.

Une étape primordiale, avant de dire que le réseau de neurone reproduit le fonctionnement d'un système, est la phase d'apprentissage, par analogie avec les neurones biologiques, par laquelle chaque être vivant passe, qui n'est que le développement des connexions entre les neurones. L'apprentissage est équivalent à l'adaptation des poids dans les réseaux de neurones artificiels. Pour adapter ces poids, nous avons utilisé l'algorithme de rétropropagation qui est le plus utilisé pour l'apprentissage supervisé. Cet algorithme corrige les poids des interconnexions en minimisant une fonction coût et en utilisant l'algorithme du gradient. Cependant, l'apprentissage n'est pas toujours optimal. Pour cette raison, des modifications sont apportées à l'algorithme du gradient, en vue d'améliorer les résultats de l'apprentissage, et peuvent être faites au début, donc avant l'exploitation du réseau ou au cours

de son utilisation. Ceci est connu respectivement sous le nom d'apprentissage off-line ou on-line.

Nous n'avons pas présenté les réseaux de neurones évolutifs ou auto-organisés, car ils ne sont pas tellement utilisés pour identifier les systèmes dans le but de les commander.

II.3 Observateur linéaire

Le principe de l'observateur d'état est de voir le modèle comme étant un système à régler où la sortie de ce modèle est comparée à la sortie du système à régler, puis l'erreur de sortie est injectée au modèle comme étant une contre réaction permettant d'annuler l'erreur d'observation ou estimation ($\tilde{x} = x - \hat{x}$), et de lui imposer une dynamique.

II.3.1 Structure de l'observateur linéaire

Soit un système défini par la réalisation linéaire déterministe suivante :

$$\begin{aligned} \dot{x}(t) &= A x(t) + B u(t) & x(0) &= x_0 \\ y(t) &= C x(t) \end{aligned} \quad (II.3)$$

où : $t \in \mathcal{R}$ est le temps; $u(t) \in \mathcal{R}^l$ est l'entrée du système; $y(t) \in \mathcal{R}^m$ est la sortie mesurée du système ; $x(t) \in \mathcal{R}^n$ est l'état du modèle.

A, B, C sont des matrices constantes tel que: $A \in \mathcal{R}^{n \times n}$; $B \in \mathcal{R}^{n \times l}$; $C \in \mathcal{R}^{m \times n}$

D'après l'équation (II.3), comme première solution, on peut écrire:

$$\dot{\hat{x}}(t) = A \hat{x}(t) + B u(t) \quad (II.4)$$

avec : \hat{x} estimation de x

Cette solution est inacceptable car le modèle mathématique (II.3) ne peut reproduire fidèlement le fonctionnement du système, ce qui est du à la négligence volontaire (simplification des équations) ou non volontaire (non maîtrise de toutes les lois qui influencent le fonctionnement du système), de plus les valeurs initiales des états ne sont pas connues. D'autre part l'erreur d'estimation augmente exponentiellement dans le cas d'un modèle instable, la commande $u = f(x)$, calculée en boucle fermée, et implantée sous la forme $u = f(\hat{x})$, devient en réalité une commande en boucle ouverte. Pour remédier à tous ces problèmes qui excluent la forme (II.4), nous corrigeons la dynamique de l'estimation dans (II.4) en tenant compte de l'écart entre la grandeur de sortie réelle et la sortie reconstruite. Cela conduit à l'observateur :

$$\begin{aligned} \dot{\hat{x}}(t) &= A \hat{x}(t) + B u(t) + K [y(t) - C \hat{x}(t)], & \hat{x}(0) &= \hat{x}_0 \\ \hat{y}(t) &= C \hat{x}(t) \end{aligned} \quad (II.5)$$

on a donc:

$$y(t) - C \hat{x}(t) = y(t) - \hat{y}(t) = \tilde{y}(t) \quad (II.6)$$

avec K le gain d'observation.

L'équation (II.5) représente la mise en contre réaction de la différence entre $y(t)$ et son estimation. On obtient ainsi la structure de l'observateur représentée à la figure II.16.

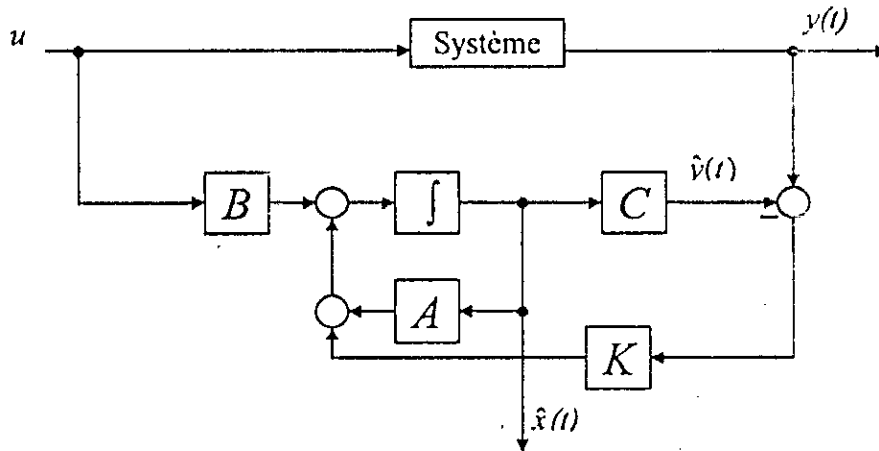


Figure II.16 Structure d'un observateur continu

Notons que : $\hat{A} = A - K C$, donc on peut écrire l'équation (II.5) sous la forme :

$$\dot{\hat{x}} = \hat{A} \hat{x}(t) + B u(t) + K y(t) \quad (\text{II.7})$$

soit : $\tilde{x} = x - \hat{x}$, l'erreur d'observation ou d'estimation. D'après (II.3) et (II.4) nous avons :

$$\dot{\tilde{x}}(t) = \hat{A} \tilde{x}(t); \tilde{x}(0) = x_0 - \hat{x}_0 \quad (\text{II.8})$$

\hat{x}_0 étant une estimation a priori de l'état initial de (II.3), et de façon générale on ne peut avoir $\hat{x}(0) = 0$. En choisissant K tel que \hat{A} soit une matrice de Hurwitz, ce qui veut dire que toutes les valeurs propres de la matrice \hat{A} appartiennent au demi plan gauche (partie réelle négative), pour que l'observateur (II.5) soit asymptotique, donc $\lim_{t \rightarrow \infty} \hat{x}(t) = 0$, et ce d'après (II.8).

Théorème : [31]

Les valeurs propres de $A - K C$ peuvent être fixées arbitrairement si et seulement si la paire (A, C) est observable.

On dit de la paire (A, C) est observable si l'on a :

$$\text{rang} [C^T, A^T C^T, A^{T^2} C^T, \dots, A^{T^{(n-1)}} C^T] = n \quad (\text{II.9})$$

La fixation de la matrice K est de façon générale faite de sorte que les valeurs propres de \hat{A} aient des parties réelles négatives plus grandes que ceux de A , donc la dynamique de convergence de l'erreur d'observation est plus rapide que la dynamique du système. Par conséquent, nous pouvons utiliser les valeurs de l'observateur dans la contre réaction de régulation du système. Mais le problème est que nous ne pouvons pas prendre des valeurs de K infiniment grandes car l'augmentation de la bande passante, résultant d'un tel choix, ne permettrait plus de négliger les bruits.

Les méthodes de calcul de la matrice K sont explicitées par [31] et [32].

II.4 Observateur non linéaire

II.4.1 Théorie de base pour l'observation

L'observabilité est la caractéristique élémentaire pour que l'on puisse parler d'un observateur. La notion d'observabilité devient plus complexe si le système est non linéaire, car pour un système linéaire l'observabilité dépend seulement de la matrice d'évolution et de la matrice de sortie [31]. Par contre, pour les systèmes non linéaires, la notion d'observabilité dépend des entrées appliquées au système et des états initiaux.

II.4.2 Différentes approches pour la synthèse d'un observateur non linéaire :

Approche 1 [12]

Soit le système décrit comme suit :

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= h(x) \end{aligned} \quad (\text{II.10})$$

avec : $x \in \mathcal{R}^n$, $u \in \mathcal{R}^q$ et $y \in \mathcal{R}^p$

On fait, si possible, une transformation sur l'espace d'état et l'espace de sortie:

$$\begin{aligned} Z &= Z(x) \\ w &= w(y) \end{aligned} \quad (\text{II.11})$$

telle que les nouvelles coordonnées du système peuvent s'écrire sous la forme :

$$\begin{aligned} \dot{Z} &= A_c Z + \gamma(w, u) \\ w &= C_c Z \end{aligned} \quad (\text{II.12})$$

où A_c et C_c sont la forme canonique duale de Brunovsky, tel que:

$$A_c = \text{bloc diag} [A_1, \dots, A_p] \quad (\text{II.13})$$

$$C_c = \text{bloc diag} [C_1, \dots, C_p]$$

avec :

$$A_i = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \end{bmatrix}_{K_i \times K_i} ; \quad C_i = [1 \ 0 \ \dots \ 0]_{1 \times K_i}$$

K_i étant l'indice d'observabilité.

D'après (II.12) l'observateur peut être écrit sous la forme suivante :

$$\dot{\hat{Z}} = (A_c + K C_c) \hat{Z} - K w + \gamma(w, u) \quad (\text{II.14})$$

avec \hat{Z} l'approximé de Z .

En faisant la différence entre (II.14) et (II.12), nous aurons l'équation de la dynamique de l'erreur :

$$\dot{\tilde{Z}} = (A_c + K C_c) \tilde{Z} \quad (\text{II.15})$$

Avec: $\tilde{Z} = \hat{Z} - Z$

Avec un bon choix de la matrice de gain, on aura une erreur qui tend exponentiellement vers zéro.

Avec la transformée inverse de (II.11), l'estimation des variables d'état initiales peut être obtenue comme suit:

$$\hat{x} = Z^{-1}(\hat{Z}) \quad (\text{II.16})$$

D'après (II.10), l'obtention de l'observateur peut se faire en deux étapes. Dans la première étape, l'entrée est considérée comme étant soit une constante u_0 , soit une fonction de la sortie $u_0(y)$. L'équation (II.10) peut s'écrire comme suit :

$$\dot{x} = f_0(x) + f_1(x, u) \quad (\text{II.17})$$

tel que : $f_0(x) = f(x, u_0(y(x)))$
 $f_1(x, u) = f(x, u) - f_0(x)$

En utilisant (II.11), nous avons:

$$\begin{aligned} \dot{x} &= f_0(x) \\ y &= h(x) \end{aligned} \quad (\text{II.18})$$

Sous une forme observable:

$$\begin{aligned} \dot{Z} &= A_c Z + y_0(w) \\ w &= C_c Z \end{aligned} \quad (\text{II.19})$$

La deuxième étape consiste à appliquer la même transformation à (II.17) tout en vérifiant que $f_1(x, u)$, avec les nouvelles coordonnées, dépend seulement des vecteurs entrée et sortie :

$$\gamma_1(w, u) = \frac{\partial Z}{\partial x} f_1(x, u) \Big|_{x=Z^{-1}(Z)} \quad (\text{II.20})$$

Si cette dernière condition est vérifiée, alors le système (II.10) peut s'écrire sous la forme observable (II.12) avec :

$$\dot{\gamma}(w, u) = \gamma_0(w) + \gamma_1(w, u) \quad (\text{II.21})$$

Nous pouvons donc construire un observateur asymptotique. Dans le cas contraire, si la condition précédente n'est pas satisfaite, le système ne peut pas se mettre sous la forme (II.12), et l'observateur (II.14) ne peut être réalisé.

Approche 2 [33]

Nous présentons une méthode systématique pour la synthèse d'un observateur pour des systèmes pouvant se mettre sous une forme bien définie (II.22). Les résultats obtenus par

Thau ont été généralisés par Kou et Al et Banks. Cette méthode donne une condition suffisante de stabilité exponentielle de l'erreur d'observation.

Soit un système pouvant se mettre sous la forme:

$$\begin{cases} \dot{x} = Ax + Bu + f(x) \\ y = Cx \end{cases} \quad (II.22)$$

avec $x(t) \in \mathcal{R}^n$; $f: \mathcal{R}^n \rightarrow \mathcal{R}^n$ est dérivable; $u(t) \in \mathcal{R}^m$; $y(t) \in \mathcal{R}^p$.

Théorèmes [33]

- Definition*
- 1- Une fonction f est dite Lipschitzienne si $\|f(x_1) - f(x_2)\| \leq K \|x_1 - x_2\|$ pour tout $x_1, x_2 \in \mathcal{R}^n$, avec $K > 0$.
 - 2- Une matrice A est dite définie négative ou a un spectre dans le demi plan gauche, si et seulement si, pour toute matrice Q symétrique définie positive, il existe une matrice P symétrique définie positive solution de l'équation algébrique de Lyapunov : $A_0^T P + P A_0 = -2Q$
 - 3- La paire (A, C) est observable, donc il existe une matrice $L \in \mathcal{R}^{n \times p}$ telle que la matrice $A_0 = A - L C$ est définie négative.
 - 4- L'inégalité suivante est vérifiée: $(\lambda_{\min}(a) / \lambda_{\max}(p)) > 0$, tel que $\lambda_{\min}(\cdot)$ et $\lambda_{\max}(\cdot)$ désignent respectivement la plus petite et plus grande valeur propre.

Si 1, 2 et 4 sont vérifiés, alors l'observateur suivant : $\dot{\hat{x}} = A\hat{x} + B u + f(\hat{x}) + L(y - C\hat{x})$ assure une convergence exponentielle de l'erreur d'observation à zéro.

Approche 3 (Observateur glissant) [34] [35]

L'observateur glissant «*sliding observer*» dérive de la théorie des systèmes à structure variable. Ce type d'observateur s'adapte très bien avec les systèmes non linéaires et incertains.

Soit le système non linéaire suivant:

$$\begin{cases} \dot{x} = f(x, u, t) \\ y = Cx \end{cases} \quad (II.23)$$

avec : $x \in \mathcal{R}^n$, $y \in \mathcal{R}^m$, $u \in \mathcal{R}^p$, et, $C \in \mathcal{R}^{m \times n}$ une matrice constante.

L'observateur glissant est défini comme suit :

$$\dot{\hat{x}} = f(\hat{x}, y, u, t) + A u_s \quad (II.24)$$

avec :

- \hat{x} : l'estimé de x
- \hat{f} : le modèle d'estimation
- A : matrice de gain de dimension $n \times p$
- u_s : vecteur défini comme $u_s = [\text{sgn}(s_1) \text{sgn}(s_2) \dots \text{sgn}(s_p)]$,
avec $[s_1 \ s_2 \ \dots \ s_p]^T = S = \Gamma [y - C \hat{x}]$
- Γ : matrice carrée ($p \times p$) à fixer.

En soustrayant les équations (II.24) de (II.23), nous obtenons:

$$\begin{cases} \dot{\tilde{x}} = \Delta f - \Lambda u, \\ \Delta f = f(x, u, t) - f(\hat{x}, y, u, t) \\ \tilde{x} = x - \hat{x} \end{cases} \quad (\text{II.25})$$

pour que le vecteur surface de glissement S soit attractif il faut que [slotine] :

$$s_i \dot{s}_i < 0 \quad i = 1, \dots, p \quad (\text{II.26})$$

La grandeur équivalente du terme de commutation, durant le régime de glissement, est donnée comme suit :

$$\Gamma C (\Delta f - \Lambda \tilde{u}_s) = 0 \quad (\text{II.27})$$

d'où :

$$\tilde{u}_s = (\Gamma C \Lambda)^{-1} \Gamma C \Delta f \quad (\text{II.28})$$

La condition primordiale dans le choix de Λ et Γ , est tel que le produit $\Gamma C \Lambda$ soit inversible:

$$\dot{\tilde{x}} = (I - \Lambda (\Gamma C \Lambda)^{-1} \Gamma C) \Delta f \quad (\text{II.29})$$

d'après l'équation (II.29), le choix des matrices Γ et Λ et du modèle \hat{f} est déterminant pour assurer la convergence de l'erreur vers zéro.

Approche 4 (Observateur basé sur la méthode du Backstepping) [33]

D'après ce qui précède, l'erreur de sortie entre le système et l'observateur est injectée dans l'observateur comme étant une commande, où son rôle est d'annuler l'erreur d'observation et de lui imposer une dynamique. Cette intervention de l'erreur peut être linéaire comme c'est le cas dans l'approche 1 et 2, sous une forme non linéaire pour l'approche 3. Dans les méthodes vues jusqu'à présent, la forme dans laquelle intervient l'erreur est connue d'avance, ce qui n'est pas le cas dans l'observateur par la méthode du Backstepping, où ce dernier est construit en plusieurs étapes. A chaque étapes les erreurs d'observation non mesurables sont considérées comme perturbations, nous les remplaçons par leur majoration, ce qui est une manière de contourner le problème. Cette procédure est connue sous le nom de *compensation non linéaire «nonlinear damping»*.

La méthode du Backstepping peut être utilisée comme une méthode de construction systématique d'un observateur (pour les systèmes non linéaires) qui converge exponentiellement. Nous donnons un exemple en annexe C.

Approche 5 (Observateur par réseau de neurones)

Les approches, déjà vues et permettant de construire un observateur, se basent sur le modèle de connaissance du système. Ce type de modèle est parfois très difficile à obtenir. Pour cette raison nous remplaçons la partie modèle de connaissance, entièrement ou partiellement, par un réseau de neurones. Dans ce cas, le retour de l'erreur de sortie ne sert pas seulement dans l'algorithme de l'observateur pour éliminer l'erreur d'observation, mais aussi pour adapter les paramètres du réseau de neurones qui sont souvent les poids du réseau.

Modèle mathématique

le choix du type de réseau de neurones se base sur les caractéristiques essentielles du système

Soit le système non linéaire suivant:

$$\begin{aligned}\dot{x} &= f(x, u) \\ y &= g(x)\end{aligned}\quad (\text{II.30})$$

$x \in \mathfrak{R}^{n \times m}$: vecteur d'état

$y \in \mathfrak{R}^m$: vecteur de sortie

$u \in \mathfrak{R}^p$: vecteur de commande

On suppose que $f(., .)$ est la fonction qu'on désire approximer par un réseau de neurones.

Donc le modèle neuronal s'écrit:

$$\dot{x} = W_r^T \delta_r [W_{r-1}^T [\dots W_2^T \delta_2 [W_1^T \delta_1 [x, u]] \dots]] + \varepsilon(x) \quad (\text{II.31})$$

$\varepsilon(x)$: erreur de construction

W_i : matrice des poids entre la i^{me} couche cachée et $i^{me} + 1$ couche cachée, avec $i=1, \dots, r$

δ_i : vecteur des fonctions d'activation de la i^{me} couche cachée

Même si les poids du réseau de neurones sont parfaitement choisis, on aura toujours une erreur $\varepsilon(x)$ due à la non fidélité du réseau de neurones, car la fonction de ce dernier n'est pas exactement celle du système.

Dans le modèle donné par l'équation (II.31), nous retrouvons deux paramètres dont nous ne pouvons pas avoir les valeurs: W_i et $\varepsilon(x)$. Pour W_i , nous utilisons l'erreur de sortie pour l'adaptation, comme par exemple «l'algorithme de rétropropagation».

En ce qui concerne l'erreur de construction, nous ne la retrouvons pas dans l'algorithme de l'observateur. Mais cette erreur ainsi que l'erreur due à l'estimation des poids, vont être compensées par la contre réaction de l'erreur de sortie dans l'algorithme de l'observateur. Ce dernier a la forme suivante:

$$\begin{aligned}\hat{\dot{x}} &= \hat{W}_r \delta_r [\hat{W}_{r-1} \delta_{r-1} [\dots \hat{W}_2 \delta_2 [\hat{W}_1 \delta_1 [\hat{x}, u]] \dots]] + K \tilde{y} \\ \hat{y} &= g(\hat{x})\end{aligned}\quad (\text{II.32})$$

\hat{x} : vecteur d'état observé ;

\hat{y} : sortie de l'observateur ;

$\tilde{y} = y - \hat{y}$: vecteur erreur de sortie ;

$K \in \mathfrak{R}^{n \times m}$: matrice de gain .

Pour l'adaptation des poids, nous pouvons employer un apprentissage off-line où les poids du réseau sont ajustés avant de les utiliser dans l'observateur. L'ajustement se fait par le biais des données entrée-sortie. Cette manière de procéder nous facilite la tâche car notre seul souci est l'annulation de l'erreur d'observation. Cependant, on risque d'avoir des instabilités si l'estimation des poids n'est pas satisfaisante, à cause de la non suffisance d'exemples d'apprentissage ou du changement dans le fonctionnement du système causé par des

perturbations ou par des défaillances dans le système. Pour remédier à ce problème nous utilisons un apprentissage on-line, soit après un apprentissage off-line pour réduire les erreurs d'estimation des poids au début de l'algorithme, soit en se donnant des valeurs initiales dans le cas où nous avons une bonne connaissance du système.

L'apprentissage on-line adapte les poids tout au long de l'utilisation de l'observateur, en se basant sur l'erreur de sortie présente à chaque instant d'échantillonnage, dans le cas où nous sommes obligés ou non à choisir de travailler en échantillonné, comme c'est le cas pour l'algorithme de rétropropagation où l'échantillonnage est impératif.

Mais le problème de l'apprentissage on-line, est de pouvoir assurer la convergence des deux erreurs (\tilde{x} et $\tilde{W} = W - \hat{W}$) en même temps. Cela nous oblige, des fois, de personnaliser l'algorithme d'apprentissage au système et de démontrer la convergence des deux erreurs en utilisant la fonction de Lyapunov.

En conclusion, cette approche nous permet d'éviter des étapes de modélisation qui peuvent être pénibles, voire non fiables et cela en se donnant un modèle qui est basé sur la structure du réseau de neurones et en adaptant ce réseau au fonctionnement du système. Ce type d'observateur est utilisé pour l'observation de interconnexions et des perturbations.

CHAPITRE III

SYSTEMES A STRUCTURE VARIABLE ET BACKSTEPPING

Lors de notre étude, nous avons utilisé deux approches pour la synthèse de la commande. La première, le mode glissant pour atténuer l'effet des interconnexions et des perturbations et la deuxième, la méthode du Backstepping pour éliminer l'erreur de poursuite. Nous présentons, dans ce chapitre, ces deux approches.

III.1 Systèmes à structure variable

III.1.1 Introduction

Un système à structure variable est un système pouvant changer de structure en faisant commuter sa commande entre deux valeurs variables, suivant une logique de commutation bien spécifique $\sigma(x)$ (figure III.1). Pour une meilleure compréhension des systèmes à structure variable, des exemples sont illustrés en [36] et [37].

Pour les systèmes à structure variable, le choix adéquat des paramètres de chaque structure et une bonne définition de la logique de commutation sont essentiels.

Cette méthode a connu un développement théorique depuis 30 ans grâce à la résolution de l'équation différentielle à second membre discontinu en 1960 par le mathématicien Russe FILLIPOV, suivie de recherches du professeur EMILYANOV en 1967 et du professeur UTKIN en 1978.

Sous certaines conditions, la commutation peut se faire à une très grande fréquence (théoriquement infiniment élevée). Le système travaille alors en *mode de glissement (sliding mode)*. Le comportement dynamique du système est alors déterminé par la condition $\sigma(x)=0$, $\sigma(x)$ étant la surface de glissement, le but essentiel est d'amener la trajectoire d'état du système vers cette surface jusqu'à atteindre le point d'équilibre (un exemple est traité en [36]).

La commande à structure variable par mode de glissement n'a connu de développement, notamment dans le domaine industriel, qu'à partir des années 80, grâce, d'une part à l'évolution vertigineuse de l'électronique de puissance qui a mis à la disposition de cette théorie des organes de commande qui répondent parfaitement aux exigences posées (variateur de courant continu, onduleur monophasé ou triphasé à pulsation...), et d'autre part, aux microprocesseurs qui sont de plus en plus performants.

La commande par mode de glissement est bien adaptée aux systèmes non linéaires ayant un modèle imprécis, c'est le cas de notre étude.

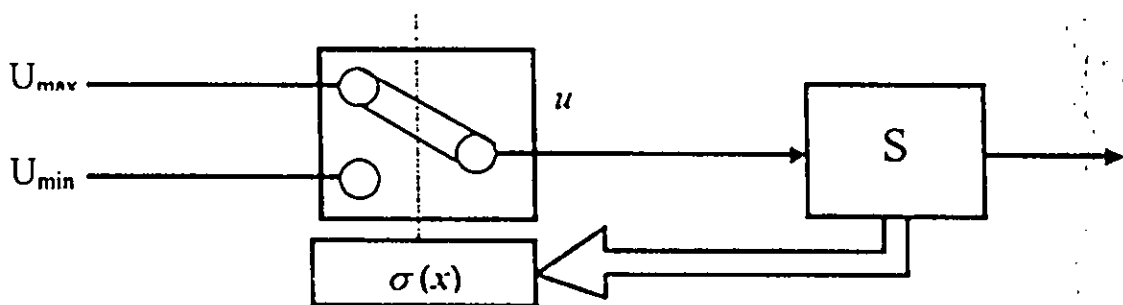


Figure III.1

III.1.2 Propriétés des modes glissants [38]

- Réduction d'ordre, car la trajectoire d'état du système en mode glissant appartient à une surface de commutation inférieure à celle de l'espace d'état. Par conséquent l'ordre des équations différentielles décrivant le fonctionnement du système en mode glissant est réduit.
- Le choix des coefficients de la surface de glissement détermine la dynamique du système en mode glissant.
- Robustesse vis à vis des perturbations extérieures et des variations paramétriques.
- La commande par mode de glissement est une commande discontinue.

III.1.3 Configuration de base pour les systèmes à structure variable [22]

La figure III.2 illustre une configuration qui englobe toutes les configurations de base des systèmes à structure variable. Dans cette configuration la commande globale est la somme de trois commandes (u_1 , u_2 , u_3):

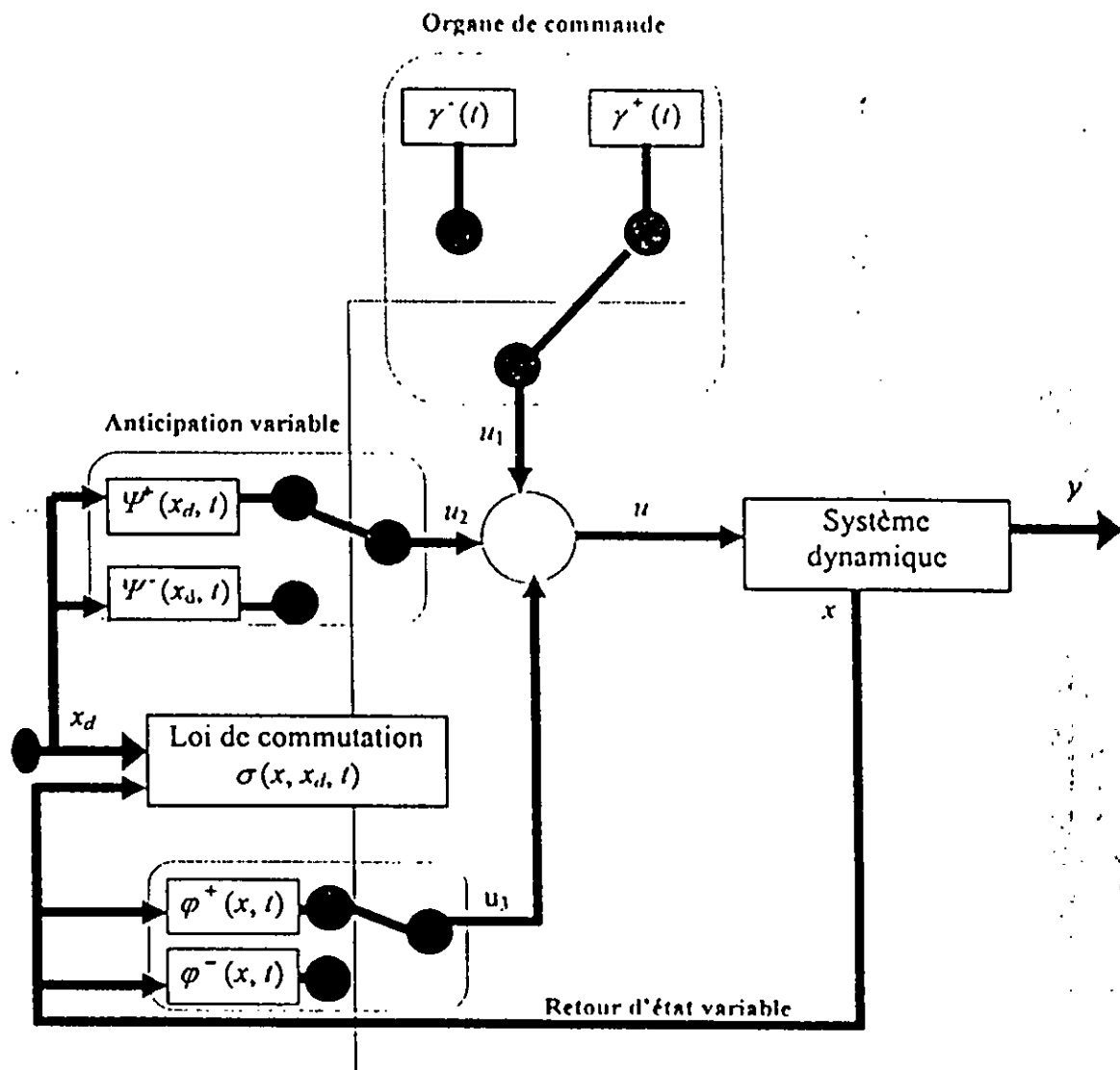


Figure III.2 Configuration de base pour les systèmes à structure variable.

- La commande u_1 représente un des deux vecteurs de commande auxiliaire suivant le signe de la loi de commutation $\sigma(x, x_d, t)$.
- La commande u_2 est formée en multipliant le vecteur désiré par une matrice d'anticipation $\Psi^+(x_d, t)$ ou $\Psi^-(x_d, t)$, suivant le signe de la loi de commutation $\sigma(x, x_d, t)$.
- La commande u_3 est formée en multipliant le vecteur d'état x par une matrice de contre réaction $\varphi^+(x_d, t)$ ou $\varphi^-(x_d, t)$, suivant le signe de la loi de commutation $\sigma(x, x_d, t)$.

on aura :

$$u = u_1 + u_2 + u_3 = \begin{cases} u^+ = \gamma^+(t) + \psi^+(x_d, t)x_d + \varphi^+(x, t)x & \text{si } \sigma(x, x_d, t) > 0 \\ u^- = \gamma^-(t) + \psi^-(x_d, t)x_d + \varphi^-(x, t)x & \text{si } \sigma(x, x_d, t) < 0 \end{cases} \quad (\text{III.1})$$

III.1.4 Solution des systèmes à second membre discontinu

La résolution du problème de l'équation différentielle à second membre discontinu, établie par le mathématicien soviétique F.FILIPOV suivie par les recherches du professeur EMILYANOV en 1967 et V.I.UTKIN en 1978, a permis à la théorie des systèmes à structure variable de se développer d'une manière très rapide durant les 40 dernières années.

Dans cette section nous développons la méthode d'UTKIN qui est la plus simple et aussi la plus usitée .

Méthode d'UTKIN [39]

Cette méthode est basée sur la commande équivalente. Soit le système non linéaire suivant :

$$\dot{x} = f(x, t) + g(x, t)u(t) \quad (\text{III.2})$$

en régime glissant idéal on a :

$$\dot{x}^* = f(x^*, t) + g(x^*, t)u_{eq} \quad (\text{III.3})$$

dans ce cas la loi de commutation s'annule, donc sa dérivée est nulle aussi:

$$\dot{\sigma}(x^*, \dot{x}^*, t) = \left(\frac{\partial \sigma}{\partial x^*} \right)^T \dot{x}^* + \frac{\partial \sigma}{\partial t} \quad (\text{III.4})$$

en remplaçant l'expression de la dérivée de x^* (III.3) dans l'équation (III.4) on aura:

$$\dot{\sigma}(x^*, \dot{x}^*, t) = \left(\frac{\partial \sigma}{\partial x^*} \right)^T (f(x^*, t) + g(x^*, t)u_{eq}) + \frac{\partial \sigma}{\partial t} \quad (\text{III.5})$$

de l'équation (III.5) on déduit l'expression de la commande équivalente :

$$u_{eq} = - \left[\left(\frac{\partial \sigma}{\partial x^*} \right)^T g(x^*, t) \right]^{-1} \left[\left(\frac{\partial \sigma}{\partial x^*} \right)^T f(x^*, t) + \frac{\partial \sigma}{\partial t} \right] \quad (\text{III.6})$$

la commande équivalente existe si et seulement si la condition de transversalité est vérifiée :

$$\det \left[\left(\frac{\partial \sigma}{\partial x^*} \right)^T g(x^*, t) \right] \neq 0 \quad (\text{III.7})$$

enfin, en mode de glissement idéal, l'équation d'état du système (III.2) devient :

$$\dot{x}^* = \left\{ 1 - g(x^*, t) \left[\left(\frac{\partial \sigma}{\partial x^*} \right)^T g(x^*, t) \right]^{-1} \left(\frac{\partial \sigma}{\partial x^*} \right)^T \right\} f(x^*, t) - g(x^*, t) \left[\left(\frac{\partial \sigma}{\partial x^*} \right)^T g(x^*, t) \right]^{-1} \frac{\partial \sigma}{\partial t} \quad (\text{III.8})$$

III.1.5 Conception de la commande par mode de glissement [34] [38]

Synthétiser une loi de commande par mode de glissement revient, en premier lieu, à choisir la surface de glissement qui permettra la convergence de la trajectoire d'état du système vers le point d'équilibre désiré, en deuxième lieu, à établir la condition d'existence du mode de glissement qui est reliée à la convergence de la trajectoire d'état vers le point d'équilibre, et en troisième lieu à déterminer la loi de commande qui aura pour rôle de maintenir les conditions de glissement (attractivité). La conception de la loi de commande par mode de glissement est ainsi décomposée en trois étapes essentielles dépendantes les unes des autres :

- 1- Choix de la surface de glissement
- 2- Etablissement de la condition de convergence
- 3- Détermination de la loi de commande

III.1.5.1 Choix de la surface de glissement. [34] [37]

La surface de glissement est un hyperplan dans l'espace d'état global, et représente le comportement dynamique désiré, d'où la trajectoire d'état du système doit suivre parfaitement cette surface. Il n'existe pas de critères spécifiques pour le choix d'une surface de glissement appropriée, c'est au concepteur de la commande de concevoir une surface de glissement stable. Cependant, nous pouvons trouver dans la littérature différentes formes de surface de glissement donnant chacune les meilleures performances pour des utilisations spécifiques.

En général, la surface de glissement est choisie linéaire par rapport aux états du système. Soit un système à m entrées de commande, la surface est alors décrite par la forme suivante :

$$\sigma = \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_m \end{bmatrix} = Sx + S_d x_d \quad (\text{III.9})$$

avec :

$x \in \mathfrak{R}^n$, vecteur d'état du système

$x_d \in \mathfrak{R}^n$, vecteur d'état désiré

$S \in \mathfrak{R}^{m \times n}$, matrice des gains définissant la dynamique de la surface de glissement

$S_d \in \mathfrak{R}^{m \times n}$, matrice des gains du vecteur consigne

On remarque que la surface de glissement globale est décomposée en m surfaces de glissement, chacune travaillant avec une seule entrée de commande :

$$\sigma_i = \sum_{j=1}^n S_{ij} x_j + \sum_{j=1}^n S_{dij} x_{dj} \quad i = 1, \dots, m \quad (\text{III.10})$$

Nous pouvons trouver dans certains ouvrages [15], une forme non linéaire de la surface de glissement qui est fonction de l'écart sur la variable à régler "x", elle est donnée par :

$$\sigma(x) = \left(\frac{\partial}{\partial t} + \lambda_x \right)^{r-1} e(x) \quad (\text{III.11})$$

avec :

$e(x)$: écart sur la variable à régler ($e(x) = x - x_d$)

λ_x : constante positive

r : degré relatif, qui représente le nombre de fois qu'il faut dériver la surface de glissement pour faire apparaître la commande.

En conclusion, nous pouvons dire que le choix de la surface de glissement est fonction de l'application et de l'objectif visé.

III.1.5.2 Condition de convergence et d'existence du mode de glissement

Définition du domaine du mode glissant [36]

Le domaine S , de divers $\sigma = 0$, est un domaine du mode glissant, si pour chaque $\varepsilon > 0$, il existe un $\delta > 0$ tel que tout mouvement, débutant au voisinage δ de S_n (domaine de dimension n), peut quitter le voisinage ε de S_n , seulement à travers le voisinage ε des bornes de S_n (figure III.3).

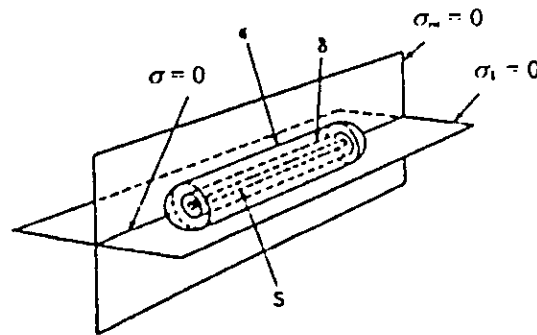


Figure III.3 Définition du domaine de glissement

Le domaine du mode glissant, appelé aussi région d'attraction, est une partie de l'espace d'état global où pour toutes conditions initiales appartenant à ce domaine, le vecteur vitesse de la trajectoire d'état converge vers la surface de glissement.

Exemple [37]

Pour mieux illustrer ce qui a été dit, nous prenons l'exemple d'un système à régler du deuxième ordre avec les deux grandeurs d'état x_1 et x_2 . La figure III.4 nous illustre l'existence du mode de glissement lorsque les commutations ont lieu continûment entre u_{\max} et u_{\min} :

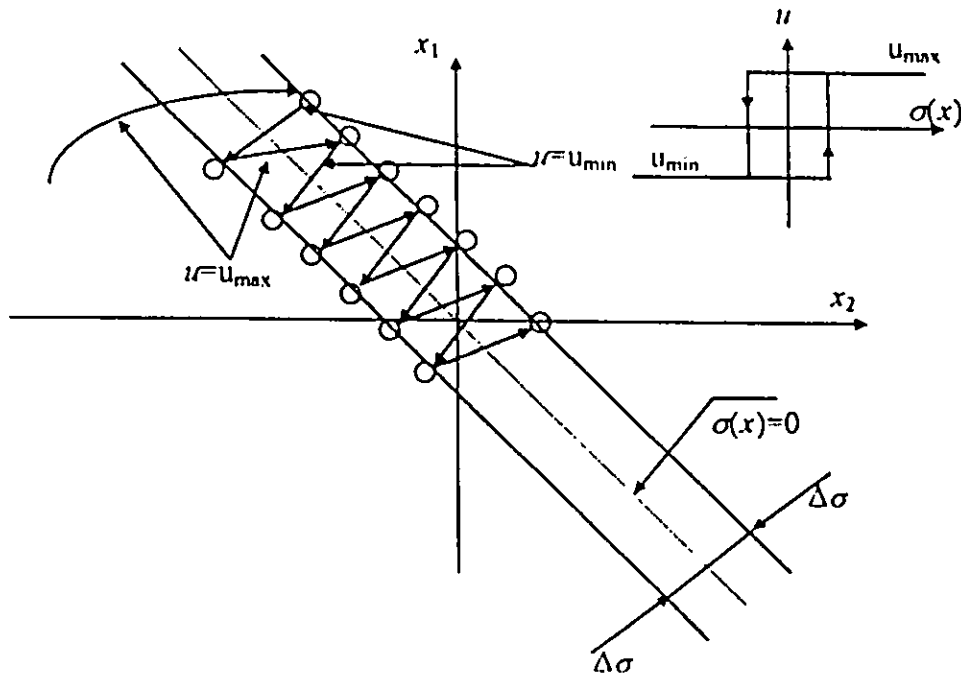


Figure III.4 Exemple d'illustration de l'existence du mode de glissement

Les commutations ont lieu sur les droites décollées parallèlement $\pm \Delta\sigma$, dans le cas d'une hystérésis infiniment petite, la loi de commutation fait un mouvement infiniment petit autour de $\sigma(x) = 0$. Par conséquent, le vecteur d'état x suit une trajectoire qui respecte cette condition. La fréquence de commutation est alors infiniment élevée.

Les conditions d'existence du mode de glissement sont des critères qui permettent aux dynamiques du système de converger vers la surface de glissement et d'y rester indépendamment des perturbations. Ces conditions d'existence du mode de glissement peuvent être définies soit par la fonction directe de commutation, soit par la théorie de stabilité de Lyapunov.

III.1.5.2.1 La fonction directe de commutation

C'est la première condition de convergence qui a été proposée par les professeurs EMILYANOV et UTKIN, et qui est basée sur le principe de donner à la surface de glissement une dynamique convergente vers zéro, elle est formulée de la manière suivante :

$$\lim_{\sigma \rightarrow 0^-} \dot{\sigma} > 0 \quad \text{et} \quad \lim_{\sigma \rightarrow 0^+} \dot{\sigma} < 0 \quad (\text{III.12})$$

Nous pouvons aussi la formuler de la manière suivante :

$$\dot{\sigma}(x)\sigma(x) < 0 \quad (\text{III.13})$$

III.1.5.2.2 Fonction de Lyapunov

Théorème [40]

Soit Ω l'espace d'état global de dimension n , et σ la surface de glissement d'ordre m associée au système. Soit D un sous espace de Ω de dimension $(n-m)$. La condition suffisante pour que D soit un domaine du mode de glissement, est l'existence d'une fonction $V(x, \sigma, t)$, dite fonction de Lyapunov, continue, dérivable et qui satisfait les conditions suivantes :

$$1) V(x, \sigma, t) \text{ est définie positive : } \begin{cases} V(x, \sigma, t) > 0 & \text{si } \sigma \neq 0 \\ V(x, \sigma, t) = 0 & \text{si } \sigma = 0 \end{cases} \quad \forall x \text{ et } t \quad (III.14)$$

2) Pour $\|\sigma\| \leq \rho$, la fonction $V(x, \sigma, t)$ est une fonction bornée :

$$h_\rho \leq V(x, \sigma, t) \leq H_\rho \quad h_\rho \geq 0 \text{ et } H_\rho > 0 \quad (III.15)$$

$$3) \quad \dot{V}(x, \sigma, t) < 0 \quad \forall x \in D \quad (III.16)$$

si x appartient à la surface de glissement alors $\dot{V}(x, \sigma, t)$ n'existe pas.

La fonction de Lyapunov doit être choisie positive et décroissante, de manière à contraindre la trajectoire du système à se diriger vers la surface de glissement des deux côtés. Par conséquent, l'idée est de choisir une fonction scalaire $\sigma(x)$ qui garantit l'attraction de la variable x vers une valeur de référence et ceci par une commande u tel que le carré de $\sigma(x)$ correspond à la fonction de Lyapunov :

$$V(x) = \frac{1}{2} \sigma^2(x) \quad (III.17)$$

$$\dot{V}(x) = \sigma(x)\dot{\sigma}(x) \quad (III.18)$$

il suffit donc de choisir u de manière à vérifier la condition (III.13).

III.1.5.3 Calcul de la commande [37] [38] [41] [42]

Après avoir sélectionné la surface de glissement σ stable de sorte que la sortie $y(x)$ du système converge vers une sortie désirée $y_d(x)$, et après s'être assuré de l'existence et de la convergence du mode de glissement, on construit une commande qui forcera les états du système à suivre la surface de glissement jusqu'au point d'équilibre, tout en maintenant la condition d'existence du mode glissant.

Le calcul de la commande par mode glissant peut se faire de plusieurs manières en fonction du système et des objectifs désirés. Nous pouvons trouver dans la littérature différentes méthodes pour la construction de la commande par mode de glissement, parmi elles les méthodes basées sur la décomposition de $u(x)$ en une somme de deux commandes, l'une continue et l'autre discontinue, ou celles basées sur la transformation de la commande originale ou de la surface de glissement. Nous présentons la première méthode que nous avons utilisée dans notre étude. Les autres sont données en annexe D.

Méthodes basées sur la décomposition $u(t) = u_N + u_{eq}$

La commande u peut être décomposée en une somme de deux commandes, l'une discontinue qui a pour rôle principal de vérifier la condition de convergence en dépit de l'imprécision sur les paramètres et le modèle, l'autre continue qui peut réduire autant qu'on veut, l'amplitude de la partie discontinue, ainsi :

$$u = u_N + u_{eq} \quad (\text{III.19})$$

u_{eq} étant la commande équivalente (continue) qui sert à maintenir l'état sur la surface de glissement $\sigma(x) = 0$, et u_N étant la commande stabilisante (discontinue et discrète).

En régime idéal, la dérivée de la surface est nulle :

$$\dot{\sigma}(x) = \frac{\partial \sigma}{\partial t} = \frac{\partial \sigma}{\partial x} \frac{\partial x}{\partial t} = \frac{\partial \sigma}{\partial x} \dot{x} \quad (\text{III.20})$$

de (III.19) et (III.2) on obtient :

$$\dot{x} = f(x, t) + g(x, t) u_{eq} + g(x, t) u_N \quad (\text{III.21})$$

on remplace (III.21) dans (III.20) :

$$\dot{\sigma}(x) = \frac{\partial \sigma}{\partial x} [f(x, t) + g(x, t) u_{eq}] + \frac{\partial \sigma}{\partial x} g(x, t) u_N \quad (\text{III.22})$$

En régime idéal, la commande appliquée au système est la commande équivalente u_{eq} , la surface de glissement est nulle, par conséquent sa dérivée ainsi que la partie discontinue sont aussi nulles. On obtient donc :

$$\begin{cases} u_N = 0 \\ \dot{\sigma}(x) = 0 \end{cases} \quad (\text{III.23})$$

$$u_{eq} = - \left(\frac{\partial \sigma}{\partial x} g(x, t) \right)^{-1} \frac{\partial \sigma}{\partial x} f(x, t) \quad (\text{III.24})$$

$$\frac{\partial \sigma}{\partial x} g(x, t) \neq 0 \quad (\text{III.25})$$

Donc en remplaçant la commande équivalente par son expression (III.24) dans (III.22) nous trouvons :

$$\dot{\sigma}(x) = \frac{\partial \sigma}{\partial x} g(x, t) u_N \quad (\text{III.26})$$

Nous choisissons pour des raisons de simplification, une surface de glissement qui vérifie la relation suivante :

$$\frac{\partial \sigma}{\partial x} g(x, t) = I \quad (\text{III.27})$$

avec I la matrice identité de dimension $m \times m$.

Donc de (III.27) et (III.26) nous avons :

$$\dot{\sigma}(x) = u_N \quad (\text{III.28})$$

La commande discontinue est choisie de manière à satisfaire la condition suffisante d'existence des modes glissants, ainsi il existe plusieurs formes pour cette commande, nous présentons quelques unes d'entre elles :

a/

$$u_N = \begin{cases} \delta(x) \operatorname{sgn}(\sigma) & \text{si } \sigma \neq 0 \\ 0 & \text{si } \sigma = 0 \end{cases} \quad (\text{III.29})$$

avec $\delta(x) < 0$.

La condition d'existence du mode de glissement est vérifiée :

$$\dot{\sigma} = \delta \sigma \operatorname{sgn}(\sigma) < 0 \quad (\text{III.30})$$

b/

$$u_N = L\sigma \quad (\text{III.31})$$

avec L matrice constante de dimension $m \times m$ définie positive.

La condition d'existence du mode de glissement est vérifiée :

$$\sigma^T \dot{\sigma} = -\sigma^T L \sigma < 0 \quad (\text{III.32})$$

c/ [8]

$$u_N = -\sum_{i=1}^l \Psi_i x_i - \delta_0 \operatorname{sgn} \sigma \quad (\text{III.33})$$

avec : $1 \leq k \leq n-1$

$$\Psi_i = \begin{cases} \alpha_i & \text{si } x_i \sigma > 0 \\ \beta_i & \text{si } x_i \sigma < 0 \end{cases} \quad (\text{III.34})$$

avec $\alpha_i > 0$ et $\beta_i < 0$ et δ_0 scalaire positif infiniment petit

La condition d'existence du mode de glissement est vérifiée :

$$\sigma^T \dot{\sigma} = -\sigma(\Psi_1 x_1 + \Psi_2 x_2 + \dots + \Psi_l x_l + \delta_0 \operatorname{sgn}(\sigma)) < 0 \quad (\text{III.35})$$

d/

$$u_N = (\sigma / \|\sigma\|) \rho \quad (\text{III.36})$$

avec $\rho < 0$ La condition d'existence du mode de glissement est vérifiée : $\sigma^T \dot{\sigma} = \frac{\rho}{\|\sigma\|} < 0$

III.1.6 Invariance des modes glissants par rapport aux perturbations [38] [39]

Soit le système suivant :

$$\dot{x} = f(x) + g(x)u + d \quad (\text{III.37})$$

avec d perturbation externe.Le régime glissant sur σ du système (III.37) satisfait les propriétés d'invariance vis à vis de d si et seulement le vecteur perturbation satisfait la condition suivante :

$$d \in \operatorname{span} \{g(x)\} \quad (\text{III.38})$$

avec $\operatorname{span}\{.\}$, espace engendré par le vecteur $\{.\}$.

III.1.7 Invariance des modes glissants par rapport à une transformation [40]

Théorème

La trajectoire du système équivalent en mode glissant est invariante par rapport à une transformation de la surface $\sigma(x, t) = \Omega(x, t) \alpha(x, t)$ si $\|\dot{\Omega}\|$ et $\|\Omega^{-1}\|$ sont bornées.

Démonstration

En dérivant l'expression (D.7), voir annexe D, on obtient en mode glissant idéal :

$$\dot{\sigma}^* = \Omega(x, t) \frac{\partial \sigma}{\partial x} (f(x, t) + g(x, t) u_{eq}^*) + \Omega(x, t) \frac{\partial \sigma}{\partial t} + \dot{\Omega} \sigma = 0 \quad (\text{III.39})$$

$$u_{eq}^* = - \left[\left(\frac{\partial \sigma}{\partial x} \right)^T g(x, t) \right]^{-1} \frac{\partial \sigma}{\partial x} f(x, t) - \left[\left(\frac{\partial \sigma}{\partial x} \right)^T g(x, t) \right]^{-1} \Omega^{-1} \dot{\Omega} \sigma - \left[\left(\frac{\partial \sigma}{\partial x} \right)^T g(x, t) \right]^{-1} \frac{\partial \sigma}{\partial t} \quad (\text{III.40})$$

or en mode glissant idéal on a $\sigma = 0$, donc :

$$u_{eq}^* = - \left[\left(\frac{\partial \sigma}{\partial x} \right)^T g(x, t) \right]^{-1} \frac{\partial \sigma}{\partial x} f(x, t) - \left[\left(\frac{\partial \sigma}{\partial x} \right)^T g(x, t) \right]^{-1} \frac{\partial \sigma}{\partial t} = u_{eq} \quad (\text{III.41})$$

finalemt, malgré la transformation, les deux systèmes admettent le même régime glissant .

III.1.8 Phénomène de broutement (chattering) [41] [43]

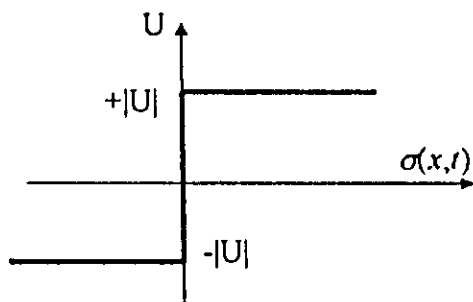
La commande par mode de glissement est discontinue et commute entre deux valeurs $\pm|U|$. En théorie la fréquence de commutation est infinie, ce qui est impossible à réaliser en pratique car la fréquence de commutation des interrupteurs est limitée, mais aussi à cause du temps de retard pour le calcul de la commande.

La commande commute entre $+|U|$ et $-|U|$ à haute fréquence, de là apparaît ce qu'on appelle phénomène de broutement (chattering), ce phénomène est déconseillé pour le bon fonctionnement du système, car il pourrait exciter des dynamiques négligées durant la modélisation.

Pour remédier à ce problème, plusieurs solutions ont été étudiées. L'adoucissement de la commande est l'une de ces méthodes, elle est simple et efficace. Cette méthode utilise des commandes à seuils en délimitant la surface de glissement par des bandes, afin d'atténuer le changement brusque de la commande de $+|U|$ à $-|U|$ (figure III.5), et dans ce cas le changement se fera à travers une ou plusieurs pentes (Annexe D) suivant le nombre de seuils choisis.

A cause du choix du nombre de seuils qui est étroitement lié à la précision du système en boucle fermée, on se trouve face à un dilemme: plus le nombre de seuils est élevé, plus le phénomène chattering est réduit, plus la précision diminue, et plus l'écart statique augmente. Nous remarquons aussi que l'invariance par rapport aux perturbations extérieures et aux imprécisions du modèle se détériore, et la robustesse est à étudier .

Fonction signe : Elle représente le changement brusque du signe de la commande de $+|U|$ à $-|U|$ qui est à l'origine du phénomène chattering .



$$U_n = \begin{cases} +|U| & \text{si } \sigma(x, t) > 0 \\ -|U| & \text{si } \sigma(x, t) < 0 \end{cases} \quad (\text{III.42})$$

Figure III.5 Fonction signe

III.1.9 Conclusion

Dans ce chapitre, la théorie des systèmes à structure variable a été présentée. La commande par mode de glissement constitue un cas particulier de ces systèmes. C'est une commande discontinue qui force les états du système à suivre une surface de glissement représentant la dynamique du système désiré. Les deux principaux avantages de cette commande est la robustesse vis-à-vis des variations paramétriques et le rejet des perturbations extérieures.

La commutation de la commande à haute fréquence cause le phénomène de broutement (chattering) qui peut sérieusement endommager l'organe de commande. En effet, des solutions à ce problème ont été proposées.

Pour notre étude, le mode de glissement sera utilisé pour atténuer l'effet des interconnexions et des perturbations dans les systèmes interconnectés tout en choisissant une surface simple et linéaire.

III.2 Backstepping

III.2.1 Introduction

Le Backstepping est une méthode récursive qui transforme un problème de synthèse d'une loi de commande pour un système global en une synthèse de séquence de commandes pour des systèmes réduits (même pour des systèmes scalaire). Cette méthode se base essentiellement sur un bon choix de la fonction de Lyapunov, qui nous assure la stabilité à chaque étape ainsi que le calcul de la commande [44].

Dans ce chapitre nous exposons succinctement le Backstepping dans le cas des systèmes monovariabiles et multivariabiles.

III.2.2 Le Backstepping dans le cas monovariable [44] [45]

1- Considérons en premier lieu un système simple écrit sous la forme suivante :

$$\begin{cases} \dot{\eta} = f(\eta) + g(\eta)\zeta \\ \dot{\zeta} = u \end{cases} \quad (\text{III.43})$$

$$[\eta^T \zeta]^T \in \mathcal{R}^{n+1}, u \in \mathcal{R}, f(0) = 0$$

On décompose le système en deux sous systèmes dont le premier est : $\dot{\eta} = f(\eta) + g(\eta)\zeta$ avec ζ considéré comme étant une commande de ce premier sous système, le second sous système est un intégrateur : $\dot{\zeta} = u$.

Soit $\phi(\eta)$ la commande stabilisante du premier sous système, tel que :

$$\phi(0) = 0, \quad \dot{\eta} = f(\eta) + g(\eta)\phi(\eta)$$

en prenant $V(\eta)$ la fonction de Lyapunov vérifiant :

$$\frac{\partial V}{\partial \eta} [f(\eta) + g(\eta)\phi(\eta)] \leq -w(\eta), \quad \forall \eta \in D \quad (\text{III.44})$$

$$w(\eta) \geq 0, \quad \forall \eta \in D$$

En ajoutant et soustrayant le terme $g(\eta)\phi(\eta)$ dans le système (V.1), et en posant $z = \zeta - \phi(\eta)$ on obtient :

$$\begin{cases} \dot{\eta} = [f(\eta) + g(\eta)\phi(\eta) + g(\eta)z] \\ \dot{z} = u - \dot{\phi} \\ \dot{\phi} = \frac{\partial \phi}{\partial \eta} [f(\eta) + g(\eta)\zeta] \end{cases} \quad (\text{III.45})$$

en posant $v = u - \dot{\phi}(\eta)$, le système (III.45) devient :

$$\begin{cases} \dot{\eta} = f(\eta) + g(\eta)\phi(\eta) + g(\eta)z \\ \dot{z} = v \end{cases} \quad (\text{III.46})$$

A la différence du système (III.43), on aboutit à un système (III.46) dont le premier sous système est asymptotiquement stable quand $z = 0$, ce résultat sera exploité pour la synthèse de la commande v .

Soit la fonction de Lyapunov candidate suivante :

$$V_a(\eta, z) = V(\eta) + \frac{1}{2} z^2 \quad (\text{III.47})$$

On a donc :

$$\dot{V}_a = \frac{\partial V}{\partial \eta} [f(\eta) + g(\eta)\phi(\eta)] + \frac{\partial V}{\partial \eta} g(\eta)z + z v \quad (\text{III.48})$$

En utilisant (III.44) dans l'équation précédente, on aura :

$$\dot{V}_a \leq -w(\eta) + \frac{\partial V}{\partial \eta} g(\eta)z + z v \quad (\text{III.49})$$

En choisissant $v = -\frac{\partial V}{\partial \eta} g(\eta) - Kz$, on a alors :

$$\dot{V}_a(\eta, z) \leq -w(\eta) - Kz^2 \quad (\text{III.50})$$

D'après le théorème de Lyapunov, le point $z = 0$ est asymptotiquement stable, puisque $\phi(0) = 0$, alors le point $(\eta = 0, \zeta = 0)$ est aussi asymptotiquement stable, d'où :

$$u = \frac{\partial \phi}{\partial \eta} [f(\eta) + g(\eta)\zeta] - \frac{\partial V}{\partial \eta} g(\eta) - K(\zeta - \phi(\eta)) \quad (\text{III.51})$$

Et si $V(\eta)$ est radialement bornée, on peut donc conclure que l'origine est globalement stable.

2- Soit le cas d'un système plus général que le système précédent (III.43) :

$$\begin{cases} \dot{\eta} = f(\eta) + g(\eta)\zeta \\ \dot{\zeta} = f_a(\eta, \zeta) + g_a(\eta, \zeta)u \end{cases} \quad (\text{III.52})$$

avec : $g_a(\eta, \zeta) \neq 0 \quad \forall \eta, \zeta$

En posant $u = [1/g_a(\eta, \zeta)] [u_a - f_a(\eta, \zeta)]$ dans le système (III.52) on obtient :

$$\begin{cases} \dot{\eta} = f(\eta) + g(\eta)\zeta \\ \dot{\zeta} = u_a \end{cases} \quad (\text{III.53})$$

S'il existe un retour $\phi(\eta)$ et une fonction de Lyapunov $V(\eta)$ qui vérifient (III.44), alors :

$$u = [1/g_a(\eta, \zeta)] \left[\frac{\partial \phi}{\partial \eta} (f(\eta) + g(\eta)\zeta) - \frac{\partial V}{\partial \eta} g(\eta) - K(\zeta - \phi) - f_a(\eta, \zeta) \right] \quad (\text{III.54})$$

avec :

$$K > 0 \quad \text{et} \quad V_a(\eta, \zeta) = V(\eta, \zeta) = V(\eta) + \frac{1}{2} [\zeta - \phi(\eta)]^2$$

désignent respectivement, la commande stabilisante et la fonction de Lyapunov pour le système global (III.52)

Avec une application récursive du Backstepping, on peut stabiliser des systèmes dits « *strict feedback systems* » qui ont la forme suivante :

$$\begin{cases} \dot{x} = f_0(x) + g_0(x)z_1 \\ \dot{z}_1 = f_1(x, z_1) + g_1(x, z_1)z_2 \\ \dot{z}_2 = f_2(x, z_1, z_2) + g_2(x, z_1, z_2)z_3 \\ \vdots \\ \dot{z}_k = f_k(x, z_1, z_2, \dots, z_k) + g_k(x, z_1, z_2, \dots, z_k)u \end{cases} \quad (\text{III.55})$$

avec $x \in \mathcal{R}^n$ et z_1, \dots, z_k sont des scalaires et, $g_i(x, z_1, \dots, z_i) \neq 0 \quad \forall 1 \leq i \leq k$

La procédure récursive débute par le sous système :

$$\dot{x} = f_0(x) + g_0(x)z_1$$

avec z_1 la commande virtuelle de ce sous système, posons $z_1 = \phi_0(x)$ et $\phi_0(0) = 0$, on considère aussi la fonction de Lyapunov $V_0(x)$ qui vérifie :

$$\frac{\partial V_0}{\partial x} [f_0(x) + g_0(x)\phi_0(x)] \leq -w(x) \quad \text{avec} \quad w(x) \geq 0 \quad (\text{III.56})$$

Il est à noter que : x est souvent choisi comme scalaire pour simplifier la procédure du Backstepping.

Dans le cas où $\phi_0(x)$ et $V_0(x)$ sont connus, on procède à l'application systématique du Backstepping, pour illustrer cette approche prenons l'exemple suivant :

Exemple : Prenons un cas simple du système (III.52) :

$$\begin{cases} \dot{x} = f_0(x) + g_0(x)z_1 \\ \dot{z}_1 = f_1(x, z_1) + g_1(x, z_1)z_2 \end{cases} \quad (\text{III.57})$$

avec : $x = \eta, z_1 = \zeta, z_2 = u, f_0 = f, g_0 = g, f_1 = f_a, g_1 = g_a$.

Etape 1 :

Le système précédent peut être stabilisé par :

$$\phi_1(x, z_1) = [1 / g_1] \left[\frac{\partial \phi_0}{\partial x} (f_0 + g_0 z_1) - \frac{\partial V_0}{\partial x} g_0 - K_1 (z_1 - \phi_0) - f_1 \right] \quad (\text{III.58})$$

La fonction de Lyapunov candidate est :

$$V_1(x, z_1) = V_0(x) + \frac{1}{2} (z_1 - \phi_0(x))^2 \quad (\text{III.59})$$

Etape 2 :

Considérons maintenant le système augmenté suivant :

$$\begin{cases} \dot{x} = f_0(x) + g_0(x)z_1 \\ \dot{z}_1 = f_1(x, z_1) + g_1(x, z_1)z_2 \\ \dot{z}_2 = f_2(x, z_1, z_2) + g_2(x, z_1, z_2)z_3 \end{cases} \quad (\text{III.60})$$

avec : $\eta = \begin{bmatrix} x \\ z_1 \end{bmatrix}$, $\zeta = z_2$, $u = z_3$, $f = \begin{bmatrix} f_0 + g_0 z_1 \\ f_1 \end{bmatrix}$, $g = \begin{bmatrix} 0 \\ g_1 \end{bmatrix}$, $f_a = f_2$, $g_a = g_2$

Le système précédent peut être stabilisé par :

$$\phi_2(x, z_1, z_2) = [1 / g_2] \left[\frac{\partial \phi_0}{\partial x} (f_0 + g_0 z_1) + \frac{\partial \phi_1}{\partial z_1} (f_1 + g_1 z_2) - \frac{\partial V_1}{\partial z_1} g_1 - K_2 (z_2 - \phi_1) - f_2 \right] \quad (\text{III.61})$$

et sa fonction de Lyapunov est :

$$V_2(x, z_1, z_2) = V_1(x, z_1) + \frac{1}{2} (z_2 - \phi_1)^2 \quad (\text{III.62})$$

En continuant de la même manière jusqu'à l'étape k, on obtiendra la loi de commande globale et la fonction de Lyapunov associée : $u = \phi_k(x, z_1, z_2, \dots, z_k)$ et $V_k(x, z_1, z_2, \dots, z_k)$

III.2.3 Le Backstepping dans le cas multivariables [44] [45]

Soit le système suivant :

$$\begin{cases} \dot{\eta} = f(\eta) + G(\eta)\zeta \\ \dot{\zeta} = f_a(\eta, \zeta) + G_a(\eta, \zeta)u \end{cases} \quad (\text{III.63})$$

Soit $\zeta = \phi(\eta)$, $\phi(0) = 0$ et $V(\eta)$ respectivement la commande stabilisante et la fonction de Lyapunov du premier sous système de (III.63) qui vérifie :

$$\frac{\partial V}{\partial \eta} [f(\eta) + G(\eta) \phi(\eta)] \leq -w(\eta) \quad (\text{III.64})$$

$$w(\eta) > 0 \quad \forall \eta$$

soit :
$$V_a(\eta, \zeta) = V(\eta) + \frac{1}{2} [\zeta - \phi(\eta)]^T [\zeta - \phi(\eta)] \quad (\text{III.65})$$

en dérivant la fonction de Lyapunov candidate par rapport au temps, on obtient :

$$\dot{V}_a = \frac{\partial V}{\partial \eta} [f + g\phi] + \frac{\partial V}{\partial \eta} G [\zeta - \phi]^T [f_a + G_a u - \frac{\partial \phi}{\partial \eta} [f + G \zeta]] \quad (\text{III.66})$$

on choisit :

$$u = G_a^{-1} \left[\frac{\partial \phi}{\partial \eta} [f + G \zeta] - \left(\frac{\partial V}{\partial \eta} G \right)^T - f_a - K [\zeta - \phi] \right] \quad (\text{III.67})$$

en remplaçant u dans (III.66) on obtient :

$$\dot{V}_a = \frac{\partial V}{\partial \eta} [f + G \phi] - K [\zeta - \phi]^T [\zeta - \phi] \leq -w(\eta) - K [\zeta - \phi]^T [\zeta - \phi] \quad (\text{III.68})$$

Donc le point d'équilibre ($\eta = 0, \zeta = 0$) est asymptotiquement stabilisable par la commande u .

III.2.4 Robustesse de la commande synthétisée par Backstepping [44]

Soit le système suivant :

$$\begin{cases} \dot{\eta} = f(\eta) + G(\eta)\zeta + \delta_\eta(\eta) \\ \dot{\zeta} = f_a(\eta, \zeta) + G_a(\eta, \zeta)u + \delta_\zeta(\eta, \zeta) \end{cases} \quad (\text{III.69})$$

avec : δ_η et δ_ζ sont des termes inconnus, on suppose que :

$$\|\delta_\eta(\eta)\|_2 \leq \alpha_1 \|\eta\|_2 \quad (\text{III.70})$$

$$\|\delta_\zeta(\eta, \zeta)\|_2 \leq \alpha_2 \|\eta\|_2 + \alpha_3 \|\zeta\|_2 \quad (\text{III.71})$$

Supposons que $\zeta = \phi(\eta)$, avec $\phi(0)=0$, est une commande stabilisante pour le premier sous système, et que $V(\eta)$ est la fonction de Lyapunov vérifiant la relation suivante :

$$\frac{\partial V}{\partial \eta} [f(\eta) + g(\eta)\phi(\eta) + \delta(\eta)] \leq -c \|\eta\|_2, \quad c > 0 \quad (\text{III.72})$$

donc le point $\eta = 0$ est un point d'équilibre asymptotiquement stable du premier sous système si et seulement si :

$$\|\phi(\eta)\| \leq \alpha_4 \|\eta\| \quad \text{et} \quad \left\| \frac{\partial \phi}{\partial \eta} \right\| \leq \alpha_5 \|\eta\| \quad (\text{III.73})$$

$$\text{soit la fonction de Lyapunov globale : } V_a(\eta, \zeta) = V(\eta) + \frac{1}{2} [\zeta - \phi(\eta)]^2 \quad (\text{III.74})$$

$$\dot{V}_a = \frac{\partial V}{\partial \eta} [f + g\phi + \delta_\eta] + \frac{\partial V}{\partial \eta} g(\zeta - \phi) + (\zeta - \phi) [f_a + G_a u + \delta_\zeta - \frac{\partial \phi}{\partial \eta} [f + g\zeta + \delta_\eta]] \quad (\text{III.75})$$

$$\text{avec : } u = [1/g_a] \left[\frac{\partial \phi}{\partial \eta} [f + g\zeta] - \frac{\partial V}{\partial \eta} g - f_a - K(\zeta - \phi) \right] \quad (\text{III.76})$$

$$\text{on aura donc : } \dot{V}_a \leq -c \|\eta\|_2 + 2\alpha_6 \|\eta\|_2 \|\zeta - \phi\| - (K - \alpha_6) (\zeta - \phi)^2 \quad (\text{III.77})$$

$$\dot{V}_\sigma \leq - \begin{bmatrix} \|\eta\|_2 \\ |\zeta - \phi| \end{bmatrix}^T \begin{bmatrix} c & -\alpha_6 \\ -\alpha_6 & K - \alpha_3 \end{bmatrix} \begin{bmatrix} \|\eta\|_2 \\ |\zeta - \phi| \end{bmatrix} \quad (\text{III.78})$$

donc finalement $\alpha_6 > 0$ si et seulement si $K > \alpha_3 + (\alpha_6^2 / c)$

on aura donc :

$$\dot{V}_\sigma \leq \sigma [\|\eta\|_2^2 + (\zeta - \phi)^2] \quad \sigma > 0 \quad (\text{III.79})$$

Lemme [44]

Considérons le système (III.69), dont les incertitudes vérifient les conditions (III.70), (III.71) et (III.73), s'il existe une commande $\phi(\eta)$ qui stabilise le premier sous système de (III.69), ainsi qu'une fonction de Lyapunov qui vérifie la condition (III.72), alors la loi de commande de (III.76) stabilise l'origine du système global et si $V(\eta)$ est radialement bornée, alors l'origine est asymptotiquement et globalement stable.

III.2.5 Conclusion

La méthode du Backstepping est simple et efficace pour la synthèse de loi de commande pour des systèmes non linéaires. Cependant elle n'est pas applicable à tous les systèmes, mais seulement à une classe de systèmes non linéaires dits « *strict feedback* ». Pour pallier à ce problème, il faut trouver un difféomorphisme afin de transformer le système de sa forme initiale vers la forme appropriée.

L'application de la méthode du Backstepping est utilisée dans le domaine de l'aéronautique [46], sur les systèmes à grande échelle tels que les réseaux électriques [47] et les machines électriques [48] [49].

CHAPITRE IV

**COMMANDE DECENTRALISEE PAR MODE DE GLISSEMENT
ET BACKSTEPPING AVEC OBSERVATEUR
A RESEAU DE NEURONES ARTIFICIELS**

IV.1 Introduction

Après avoir introduit les outils théoriques nécessaires au développement de la loi de commande, nous la synthétisons pour un système mécanique défini par les équations d'Euler-Lagrange. Nous appliquons ensuite cette commande en simulation sur un robot manipulateur PUMA 560, décrit dans le chapitre I.

Le travail est organisé de la manière suivante : En premier lieu, nous exposons les notions de base nécessaires au développement de la commande. En deuxième lieu, nous nous intéressons à la synthèse de la commande comme première étape puis à sa simulation.

La synthèse de la commande se résume à un observateur par réseau de neurones artificiels des interconnexions et des perturbations, une commande glissante pour atténuer leurs effets et une commande par la méthode du Backstepping pour la consigne. La simulation est faite sur un robot manipulateur PUMA 560 avec des tests de robustesse et de décentralisation puis, nous l'interprétons. La comparaison avec d'autres types de commande (Proportionnel Dérivé et commande de Y.Tang et coll. [14]) est donnée au chapitre V.

IV.2 Données de base

A/ Rappels mathématiques

- On note \mathfrak{R} l'ensemble des réels, \mathfrak{R}^n l'ensemble des vecteurs de dimension n et $\mathfrak{R}^{n \times m}$ l'ensemble des matrices de dimension $n \times m$.
- On définit la norme 2 d'un vecteur $x \in \mathfrak{R}^n$ comme suit : $\|x\|_2 = \sqrt{x_1^2 + \dots + x_n^2}$
- On définit la norme de la matrice $A \in \mathfrak{R}^{n \times m}$ comme suit : $\|A\| = \sqrt{\lambda_{\max}[A^T A]}$
où $\lambda_{\max}[\cdot]$ et $\lambda_{\min}[\cdot]$ sont respectivement la plus grande et la plus petite valeur propre de A .
- On note par $\text{tr}(\cdot)$ la trace de la matrice, c'est à dire la somme de ses éléments diagonaux, et par $\|\cdot\|_F$ comme étant la norme Frobenienne de la matrice qui est définie comme suit :

$$\|A\|_F^2 = \text{tr}(A^T A) \quad \text{avec : } A \in \mathfrak{R}^{n \times m}$$

A.1/ Les normes 2 et Frobenienne sont compatibles :

$$\|Ax\|_2 \leq \|A\|_F \|x\|_2 \quad \text{avec : } A \in \mathfrak{R}^{n \times m} \text{ et } x \in \mathfrak{R}^m$$

A.2/ La trace d'une matrice vérifie la propriété suivante :

$$\text{tr}(Q^T v W^T) = v^T Q W \quad \text{avec : } v \in \mathfrak{R}^n, W \in \mathfrak{R}^l \text{ et } Q \in \mathfrak{R}^{n \times l}$$

A.3/ Soient a et b deux nombres réels, la propriété suivante est toujours vérifiée :

$$ab \leq (a^2 + b^2) / 2$$

B/ Modèle du robot

Le modèle dynamique d'un robot manipulateur à n degrés de liberté s'écrit sous la forme d'une équation différentielle non linéaire du second ordre :

$$M(q)\ddot{q} + B(q, \dot{q})\dot{q} + K(q, \dot{q})q + G(q) = \Gamma + u_p \quad (IV.1)$$

$q \in \mathcal{R}^n$, $\dot{q} \in \mathcal{R}^n$, $\ddot{q} \in \mathcal{R}^n$ sont respectivement les vecteurs de position, de vitesse et d'accélération. $M(q) \in \mathcal{R}^{n \times n}$ est la matrice d'inertie. Elle est bornée symétrique et est définie positive. $B(q, \dot{q})\dot{q} + K(q, \dot{q})q$ représentent le couple dû aux forces de Coriolis et centrifuges. $G(q)$ est la force due à la gravitation. Γ sont les forces généralisées appliquées au système et u_p est la force appliquée au système par l'effet de la charge.

Les équations du robot manipulateur vues d'une manière décentralisée sont [14] :

$$\begin{aligned}
 m_i(q_i)\ddot{q}_i + c_i(q_i, \dot{q}_i)\dot{q}_i + g_i(q_i) + z_i(q, \dot{q}, \ddot{q}) + d_i(q, \dot{q}) &= u_i \\
 z_i(q, \dot{q}, \ddot{q}) &\stackrel{\Delta}{=} \left\{ \sum_{j=1, j \neq i}^n m_{ij}(q)\ddot{q}_j + [m_{ii}(q) - m_i(q_i)]\ddot{q}_i \right\} + \\
 &\left\{ \sum_{j=1, j \neq i}^n c_{ij}(q, \dot{q})\dot{q}_j + [c_{ii}(q, \dot{q}) - c_i(q_i, \dot{q}_i)]\dot{q}_i \right\} + [\bar{g}_i(q) - g_i(q_i)] \quad (IV.2) \\
 d_i(q_i, \dot{q}_i) &\stackrel{\Delta}{=} f_i(q_i, \dot{q}_i) + T_i(q_i)
 \end{aligned}$$

avec :

- $m_i(q_i) \in \mathcal{R}^{n_i \times n_i}$, matrice d'inertie de l' $i^{ème}$ articulation,
- $c_i(q_i, \dot{q}_i)\dot{q}_i \in \mathcal{R}^{n_i}$, couples dus aux forces de Coriolis et centrifuges appliqués à l' $i^{ème}$ articulation,
- $g_i(q_i) \in \mathcal{R}^{n_i}$, forces et/ou couples dus à la gravitation, appliqués à l' $i^{ème}$ articulation,
- $z_i(q, \dot{q}, \ddot{q}) \in \mathcal{R}^{n_i}$, vecteur des interconnexions :
- $q^T = [q_1^T \dots q_N^T] \in \mathcal{R}^n$, $n = \sum_{i=1}^N n_i$, N étant le nombre d'articulations
- $d_i(q, \dot{q}) \in \mathcal{R}^{n_i}$, vecteur de perturbation appliqué à l' $i^{ème}$ articulation.

Propriétés : [3] [14]

B.1/ La matrice $m_i(q_i)$ est symétrique et bornée avec :

$$\bar{m}_i > \underline{m}_i > 0 \quad \underline{m}_i I < m_i(q_i) = m_i^T(q_i) < \bar{m}_i I, \forall q_i \in \mathcal{R}^{n_i}$$

B.2/ Les interconnexions sont bornées :

$$\|z_i(q, \dot{q}, \ddot{q})\|_2 \leq z_i \quad z_i > 0 \quad \forall i = 1 \dots N$$

B.3/ Les perturbations sont bornées :

$$\|d_i(q, \dot{q})\|_2 \leq d_i \quad d_i > 0 \quad \forall i = 1 \dots N$$

B.4/ Les dérivées des interconnexions sont bornées :

$$\|\dot{z}_i(q, \dot{q}, \ddot{q})\|_2 \leq z_i' \quad z_i' > 0 \quad \forall i = 1 \dots N$$

B.5/ Les dérivées des perturbations sont bornées :

$$\|\dot{d}_i(q, \dot{q})\|_2 \leq d'_i \quad d'_i > 0 \quad \forall i = 1 \dots N$$

C/ Réseau de neurones [11]

Dans cette partie, nous donnons certaines notions de base concernant le réseau de neurones utilisé par l'observateur.

Nous représentons à la figure IV.1 une manière fonctionnelle de liaison dans la structure du réseau de neurones considéré comme étant un réseau de neurones à une seule couche cachée avec une entrée de pré-traitement. L'aptitude de ce genre de réseau de neurones, à approximer des fonctions continues, a été généralement étudiée [50] et [51]. Ce type de réseau est connu sous le nom de *functional link neural network* (FLNN), il est utilisé pour approximer des fonctions non linéaires.

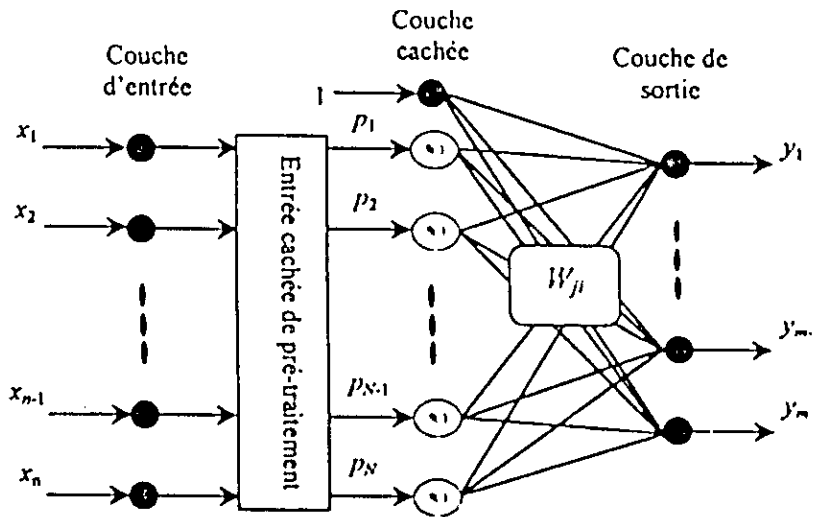


Figure IV.1 Structure d'un réseau de neurones de type FLNN

C.1/ Eléments constitutants du FLNN

a- Couche d'entrée : C'est une couche où les neurones ont un seul rôle : celui de transmettre l'information sans la modifier à l'entrée cachée de pré-traitement.

b- Entrée cachée de pré-traitement : Cette couche a comme entrées, les entrées de la (ou des) fonction(s) à reproduire et comme sorties des valeurs de fonctions pré-définies par des connaissances qu'on a sur cette (ou ces) fonction(s). Ces sorties peuvent être des normes vectorielles, des fonctions dépendantes seulement de certaines entrées, ou voire valeurs de certaines entrées.

c- Couche cachée du réseau de neurones : Elle est constituée de neurones où chacun a une fonction d'activation $\delta(.)$ qui est continue de $\mathcal{R} \rightarrow \mathcal{R}$ et qui doit être bornée sur \mathcal{R} . Les fonctions d'activation généralement utilisées sont :

- La fonction sigmoïde : $\delta(k) = 1 / (1 + e^{-k})$
- La fonction tangente hyperbolique : $\delta(k) = (e^k - e^{-k}) / (e^k + e^{-k})$
- La fonction gaussienne : $\delta(k) = e^{-k \times k}$

Les fonctions d'activation des neurones ne doivent pas être nécessairement les mêmes. Nous pouvons avoir dans une couche cachée plusieurs types de fonctions d'activation. Dans notre cas nous avons choisi une fonction sigmoïde où :

$$\delta(k) = \begin{cases} 1 & \text{si } k \rightarrow +\infty \\ 0 & \text{si } k \rightarrow -\infty \end{cases} \quad (IV.3)$$

Le nombre de neurones est égal au nombre de sorties de l'entrée cachée de pré-traitement. De plus, nous avons une entrée unitaire qui sera directement transmise à la sortie de la couche cachée.

d- Matrice de pondération : C'est la partie dynamique du réseau, car c'est à ce niveau qu'intervient l'apprentissage pour arriver à reproduire la fonction voulue. Cette matrice a le rôle de décision sur l'influence de chaque sortie de la couche cachée sur la sortie ou les sorties, si nous avons plusieurs fonctions à reproduire.

Hypothèse : Les valeurs qui constituent la matrice poids idéal du réseau de neurones sont bornées comme suit :

$$\|W\|_F \leq W_m \quad \text{avec } W_m \in \mathcal{Y} \quad (IV.4)$$

e- Couche de sortie : Les neurones de cette couche auront à sommer les valeurs venant de la couche cachée après avoir été pondérées.

Ces étapes, que l'information de l'entrée va subir pour nous reproduire la fonction voulue, restent insuffisantes pour que nous puissions dire que nous avons une reproduction parfaitement fidèle, même si nous avons la matrice poids adéquate. Par conséquent, nous devons toujours tenir compte de cette erreur dans l'équation mathématique du réseau de neurones par un terme $\varepsilon(x)$ qui est borné comme suit :

$$\|\varepsilon(x)\|_2 < \varepsilon_m \quad \text{où } \varepsilon_m \in \mathcal{Y} \quad (IV.5)$$

L'expression mathématique, qui donne le fonctionnement du réseau de neurones en utilisant la forme matricielle, est la suivante :

$$y(x) = W^T \delta(p) + \varepsilon(x) \quad \text{avec } W \in \mathcal{Y}^{N \times m}, \delta(p) \in \mathcal{Y}^N, y(x) \in \mathcal{Y}^m \text{ et } \varepsilon(x) \in \mathcal{Y}^m \quad (IV.6)$$

- W est la matrice poids idéal.
- $\delta(p)$ est le vecteur constitué de fonctions d'activation des neurones de la couche cachée.
- $\varepsilon(x)$ est le vecteur d'erreur de construction.
- $y(x)$ est le vecteur de sortie du réseau de neurones.

En réalité, les valeurs de la matrice poids idéal ne sont pas connues. Par ailleurs, nous ne pouvons pas connaître l'erreur de construction. Par conséquent, la sortie du réseau de neurones qui approxime la (les) fonction(s) à reproduire est la suivante :

$$\hat{y}(x) = \hat{W} \delta(p) \quad (IV.7)$$

où $\hat{y}(x)$ et \hat{W} sont respectivement la sortie estimée de $y(x)$ et l'estimation de la matrice poids idéal W , que nous essayons d'approcher le plus possible en utilisant un algorithme d'apprentissage on-line présenté plus loin.

IV.3 Observateur par réseau de neurones des interconnexions

En commande décentralisée la présence physique des interconnexions constitue une contrainte supplémentaire dans la synthèse et la réalisation de la commande. En effet, lors de la commande, chaque station de commande locale doit compenser l'effet des interconnexions et des perturbations provenant des autres sous-systèmes. La connaissance des signaux relatifs à ces interconnexions et perturbations au niveau local devient nécessaire et la conception d'un estimateur de ces grandeurs s'impose.

Dans notre travail, nous proposons un algorithme d'estimation à base de réseaux de neurones artificiels.

On note par $y_i \in \mathcal{R}^{m_i}$, la partie des interconnexions plus perturbations, appliquée à l' $i^{\text{ème}}$ articulation. y_i s'écrit donc de la manière suivante :

$$y_i = z_i(q, \dot{q}, \ddot{q}) + d_i(q_i, \dot{q}_i) + \omega_i \quad (\text{IV.8})$$

Pour synthétiser l'algorithme de l'observateur, on doit dériver y_i qui est prise comme étant une variable d'état remplaçant les interconnexions et les perturbations et un terme ω_i qui sera vu ultérieurement. Le modèle du système résultant est augmenté et, par conséquent, l'algorithme s'écrit comme suit :

$$\dot{y}_i = \dot{z}_i(q, \dot{q}, \ddot{q}) + \dot{d}_i(q_i, \dot{q}_i) + \dot{\omega}_i \quad (\text{IV.9})$$

Le terme $z_i(q, \dot{q}, \ddot{q}) + d_i(q_i, \dot{q}_i)$, caractérise l'interconnexion et la perturbation dont on ne connaît que sa borne.

Pour reproduire cette fonction, on fait appel au réseau de neurones de type FLNN cité précédemment. L'algorithme prendra donc la forme suivante :

$$\dot{y}_i = W_i^T \delta_i(p_i) + \dot{\omega}_i + \varepsilon_i(x) \quad (\text{IV.10})$$

avec : $p_i \in \mathcal{R}^{l_i}$, $W_i^T \in \mathcal{R}^{m_i \times l_i}$, $\dot{\omega}_i \in \mathcal{R}^{m_i}$, $\varepsilon_i(x) \in \mathcal{R}^{m_i}$ et $\dot{y}_i \in \mathcal{R}^{m_i}$.

p_i est la sortie de la couche d'entrée cachée de pré-traitement. Cette dernière a seulement comme entrée les coordonnées généralisées de l' $i^{\text{ème}}$ articulation à cause de la décentralisation. Si W_i et $\varepsilon_i(x)$ sont connus, l'équation (IV.10) est l'observateur des interconnexions et perturbations. Dans le cas contraire, on approxime W_i par \hat{W}_i qui sera calculé par un algorithme. On ajoute alors un terme correctif à l'observateur.

La forme finale de l'observateur est :

$$\dot{\hat{y}}_i = \hat{W}_i^T \delta_i(p_i) + \dot{\omega}_i + K_i \tilde{y}_i \quad (\text{IV.11})$$

\hat{y}_i est l'estimation de y_i , $K_i \in \mathcal{R}^{m_i \times m_i}$ est une matrice diagonale définie positive et $\tilde{y}_i = \dot{y}_i - \dot{\hat{y}}_i$ est l'erreur d'observation.

Des équations (IV.10) et (IV.11), on obtient :

$$\begin{aligned}\dot{\tilde{y}}_i &= \dot{y}_i - \hat{y}_i = W_i^T \delta_i(p_i) + \dot{\omega}_i + \varepsilon_i(x) - \hat{W}_i^T \delta_i(p_i) - \dot{\omega}_i + K_i \tilde{y}_i \\ \dot{\tilde{y}} &= \tilde{W}_i^T \delta_i(p_i) - K_i \tilde{y}_i + \varepsilon_i(x) \\ \tilde{W}_i^T &= W_i^T - \hat{W}_i^T\end{aligned}\tag{IV.12}$$

L'algorithme d'adaptation des poids du réseau de neurones est donné par l'expression suivante :

$$\dot{\hat{W}}_i^T = \delta_i(p_i) \tilde{y}_i^T - K_{ei} \|\tilde{y}_i\|_2 \hat{W}_i^T\tag{IV.13}$$

où K_{ei} représente une constante réelle strictement positive.

La stabilité des deux algorithmes précédents (observateur, poids W_i) doit être démontrée. Cela revient à vérifier que les erreurs d'observation \tilde{y}_i et d'approximation des poids tendent vers zéro. L'utilisation de la fonction de Lyapunov est nécessaire car le système est non linéaire et la convergence des deux erreurs doit être simultanée.

La fonction de Lyapunov choisie est la suivante :

$$\begin{aligned}V(\tilde{y}, \tilde{W}) &= \frac{1}{2} \sum_{i=1}^N [\tilde{y}_i^T \tilde{y}_i + \text{tr}(\tilde{W}_i^T \tilde{W}_i)] \\ \tilde{y}^T &= [\tilde{y}_1^T \dots \tilde{y}_N^T] \\ \tilde{W} &= \text{blocdiag}(\tilde{W}_1 \tilde{W}_2 \dots \tilde{W}_N)\end{aligned}\tag{IV.14}$$

D'après l'équation (IV.14), nous avons $V(\tilde{y}, \tilde{W})$ toujours positive.

Les erreurs \tilde{y}, \tilde{W} doivent converger vers zéro :

$$\begin{aligned}\dot{V}(\tilde{y}, \tilde{W}) &= \sum_{i=1}^N [\tilde{y}_i^T \dot{\tilde{y}}_i + \text{tr}(\tilde{W}_i^T \dot{\tilde{W}}_i^T)] = \sum_{i=1}^N [\tilde{y}_i^T (\tilde{W}_i^T \delta_i(p_i) - K_i \tilde{y}_i + \varepsilon_i(x)) + \text{tr}(\tilde{W}_i^T \dot{\tilde{W}}_i^T)] \\ \dot{V}(\tilde{y}, \tilde{W}) &= \sum_{i=1}^N [\tilde{y}_i^T \tilde{W}_i^T \delta_i(p_i) - \tilde{y}_i^T K_i \tilde{y}_i + \tilde{y}_i^T \varepsilon_i(x) + \text{tr}(-\tilde{W}_i^T \delta_i(p_i) \tilde{y}_i^T + K_{ei} \|\tilde{y}_i\|_2 \tilde{W}_i^T)] \\ \text{car : } \dot{\tilde{W}}_i &= \dot{W}_i - \dot{\hat{W}}_i = 0 - \dot{\hat{W}}_i\end{aligned}\tag{IV.15}$$

En tenant compte de A.2 :

$$\dot{V}(\tilde{y}, \tilde{W}) = \sum_{i=1}^N [-\tilde{y}_i^T K_i \tilde{y}_i + \tilde{y}_i^T \varepsilon_i(x) + \text{tr}(-\tilde{W}_i^T K_{ei} \|\tilde{y}_i\|_2 \tilde{W}_i + K_{ei} \|\tilde{y}_i\|_2 \tilde{W}_i^T W_i)]$$

D'après A.1 :

$$\dot{V}(\tilde{y}, \tilde{W}) \leq \sum_{i=1}^N [-\lambda_{\min}(K_i) \|\tilde{y}_i\|_2^2 + \|\tilde{y}_i\|_2 \varepsilon_m - K_{ei} \|\tilde{y}_i\|_2 \|\tilde{W}_i\|_F^2 + K_{ei} \|\tilde{y}_i\|_2 W_m \|\tilde{W}_i\|_F]$$

D'après (IV.5) : $\|W_i\|_F \leq W_m, \forall i = 1 \dots N$

D'après A.3:

$$\dot{V}(\tilde{y}, \tilde{W}) \leq \sum_{i=1}^N -\|\tilde{y}_i\| \left[\lambda_{\min}(K_i) \|\tilde{y}_i\| - \varepsilon_m + K_{c_i} \|\tilde{V}_i\|_F^2 - K_{c_i} \frac{W_m^2}{2} - K_{c_i} \frac{\|\tilde{V}_i\|_F^2}{2} \right]$$

$$\dot{V}(\tilde{y}, \tilde{W}) \leq \sum_{i=1}^N -\|\tilde{y}_i\| \left[\lambda_{\min}(K_i) \|\tilde{y}_i\| - \varepsilon_m + \frac{K_{c_i}}{2} \|\tilde{V}_i\|_F^2 - K_{c_i} \frac{W_m^2}{2} \right]$$

La fonction de Lyapunov a une dérivée négative si :

$$\|\tilde{y}_i\| \geq \frac{\varepsilon_m + \frac{K_{c_i} W_m^2}{2}}{\lambda_{\min}(K_i)} = \beta_{\tilde{y}} \quad \text{domaine de convergence.}$$

Pour diminuer ce domaine, nous prenons $\lambda_{\min}(K_i)$ grand et K_{c_i} petit.

IV.4 Synthèse de la loi de commande décentralisée robuste

Considérons le modèle décentralisé du système mécanique:

$$m_i(q_i) \ddot{q}_i + c_i(q_i, \dot{q}_i) \dot{q}_i + g_i(q_i) + z_i(q, \dot{q}, \ddot{q}) + d_i(q, \dot{q}) = u_i$$

Nous décomposons la commande u_i en deux commandes telles que : $u_i = v_i - \omega_i$:

- v_i désigne la commande du système sans interconnexions.
- ω_i désigne la commande utilisée pour atténuer l'effet des interconnexions et des perturbations.

D'après (IV.8), l'équation de l' i^{me} articulation s'écrit comme suit :

$$m_i(q_i) \ddot{q}_i + c_i(q_i, \dot{q}_i) \dot{q}_i + g_i(q_i) + y_i = v_i \quad (\text{IV.16})$$

A/ Atténuation de l'effet des interconnexions et perturbations

L'objectif de la commande ω_i est de faire tendre y_i vers zéro. La commande ω_i sera alors synthétisée à l'aide de la méthode du mode glissant. Ce choix est justifié par la robustesse de cette loi vis-à-vis des erreurs de construction.

En se basant sur des travaux déjà faits dans ce domaine [8] et [9], la surface de glissement sera choisie de la manière suivante :

$$S_i = y_i \quad (\text{IV.17})$$

avec S_i : surface de glissement.

D'après (IV.10), on a :

$$\dot{S}_i = \dot{y}_i = W_i^T \delta_i(p_i) + \dot{\omega}_i + \varepsilon_i(x) \quad (\text{IV.18})$$

On ne dispose pas de valeurs de y_i , mais seulement de son approximation \hat{y}_i . Par conséquent :

$$\dot{S}_i = \hat{y}_i \quad (\text{IV.19})$$

D'après (IV.11), on obtient :

$$\dot{\hat{S}}_i = \hat{W}_i^T \delta_i(p_i) + \dot{\omega}_i + K_i \tilde{y}_i \quad (IV.20)$$

La commande ω_i va éliminer \hat{y}_i . Pour que y_i soit atténué, la convergence de \tilde{y}_i vers $B_{\tilde{y}}$ doit être plus rapide que celle de \hat{y}_i vers zéro. Cette condition est vérifiée par un choix judicieux des constantes de temps qui déterminent la dynamique de la convergence.

La convergence de \hat{y}_i vers zéro coïncide avec la convergence de \hat{S}_i vers zéro qui est obtenue d'après la condition d'existence du mode glissant [36]:

$$\hat{S}_i^T \dot{\hat{S}}_i < 0 \quad (IV.21)$$

On opte alors pour le choix :

$$\dot{\hat{S}}_i = -\hat{S}_i \operatorname{sgn}(\hat{S}_i) - K_{si} \hat{S}_i \quad (IV.22)$$

Dans l'équation (IV.22), le premier terme assure la commutation et le deuxième impose la dynamique de convergence de \hat{S}_i vers zéro.

Calcul de la commande ω_i

Des équations (IV.20) et (IV.22), on obtient :

$$\hat{W}_i^T \delta_i(p_i) + \dot{\omega}_i + K_i \tilde{y}_i = -\hat{S}_i \operatorname{sgn}(\hat{S}_i) - K_{si} \hat{S}_i \quad (IV.23)$$

Or, d'après la relation (IV.19), on a :

$$\hat{W}_i^T \delta_i(p_i) + \dot{\omega}_i + K_i \tilde{y}_i = -\hat{y}_i \operatorname{sgn}(\hat{y}_i) - K_{si} \hat{y}_i \quad (IV.24)$$

$$\dot{\omega}_i = -\hat{y}_i \operatorname{sgn}(\hat{y}_i) - K_{si} \hat{y}_i - \hat{W}_i^T \delta_i(p_i) - K_i \tilde{y}_i \quad (IV.25)$$

Le calcul de ω_i revient à intégrer l'équation (IV.25). Cette méthode est connue sous le nom de mode glissant généralisé. Elle élimine l'effet chattering, en évitant la discontinuité de la commande.

B/ Loi de commande pour l' i^{me} articulation sans interconnexions

Après avoir atténué l'effet des interconnexions et des perturbations appliquées à chaque articulation, on utilise la commande v_i pour imposer une trajectoire avec une certaine dynamique à chacune d'elles. On synthétise la commande v_i en utilisant la méthode du Backstepping. L'obtention de la commande v_i se fait en deux étapes :

- A la première étape, nous obtenons une commande virtuelle qui n'est qu'une variable d'état.
- A la deuxième étape apparaît la commande réelle v_i .

Forme des équations d'état de l' i^{me} articulation :

$$\begin{cases} \dot{q}_{1i} = q_{2i} \\ \dot{q}_{2i} = m_i(q_{1i})^{-1} [-c_i(q_{1i}, q_{2i}) q_{2i} - g_i(q_{1i}) + v_i] \end{cases} \quad (IV.26)$$

La trajectoire de référence est notée q_{1d} et sa dynamique (vitesse) $q_{2d} = \dot{q}_{1d}$.
 L'erreur de poursuite: $e_{1l} = q_{1d} - q_{1l}$ et l'erreur de vitesse: $\dot{e}_{1l} = \dot{q}_{1d} - \dot{q}_{1l}$.

Synthèse de la commande :

Etape 1 :

On prend la fonction de Lyapunov :

$$V_1(e_{1l}) = 1/2 e_{1l}^T e_{1l} \quad (IV.27)$$

$$\dot{V}_1(e_{1l}) = e_{1l}^T \dot{e}_{1l} = -\frac{\alpha}{2} V_1(e_{1l}), \text{ avec } \alpha > 0$$

$$\dot{V}_1(e_{1l}) = -\alpha e_{1l}^T e_{1l}$$

$$\dot{e}_{1l} = -\alpha e_{1l} \quad (IV.28)$$

$$\dot{q}_{1d} - q_{2l} = -\alpha(q_{1d} - q_{1l})$$

$$q_{2l} = \alpha(q_{1d} - q_{1l}) + \dot{q}_{1d} = \varphi(\cdot)$$

Etape 2:

On pose $e_{2l} = \varphi(\cdot) - q_{2l}$

La fonction de Lyapunov augmentée s'écrit :

$$V_2(e_{1l}, e_{2l}) = \frac{1}{2} e_{1l}^T e_{1l} + \frac{1}{2} e_{2l}^T e_{2l} \quad (IV.29)$$

$$e_{2l} = \alpha(q_{1d} - q_{1l}) + \dot{q}_{1d} - q_{2l}$$

$$\dot{e}_{2l} = \alpha(\dot{q}_{1d} - \dot{q}_{2l}) + \ddot{q}_{1d} - \dot{q}_{2l}$$

$$\dot{V}_2(e_{1l}, e_{2l}) = -\alpha e_{1l}^T e_{1l} + e_{2l}^T e_{2l}$$

$$\dot{V}_2(e_{1l}, e_{2l}) = -\frac{\alpha}{2} V_1(e_{1l}) + e_{2l}^T \dot{e}_{2l}$$

$$\text{On a } -\frac{\alpha}{2} V_1(e_{1l}) < 0 \text{ donc } e_{2l}^T \dot{e}_{2l} \text{ doit être inférieur à zéro} \quad (IV.30)$$

$$e_{2l}^T \dot{e}_{2l} = -b e_{2l}^T e_{2l}, \text{ avec } b > 0$$

$$\dot{e}_{2l} = -b e_{2l}$$

$$\alpha[\dot{q}_{1d} - q_{2l}] + \ddot{q}_{1d} - \dot{q}_{2l} = -b[\alpha[q_{1d} - q_{1l}] + \dot{q}_{1d} - q_{2l}]$$

$$\dot{q}_{2l} = b[\alpha[q_{1d} - q_{1l}] + \dot{q}_{1d} - q_{2l}] + \alpha[\dot{q}_{1d} - q_{2l}] + \ddot{q}_{1d}$$

$$v_l = m_l(q_{1l}) \{ b[\alpha[q_{1d} - q_{1l}] + \dot{q}_{1d} - q_{2l}] + \alpha[\dot{q}_{1d} - q_{2l}] + \ddot{q}_{1d} \} + c_l(q_{1l}, q_{2l}) v_{2l} + g(q_{1l})$$

Nous donnons en figure IV.2 le schéma fonctionnel de la loi de commande.

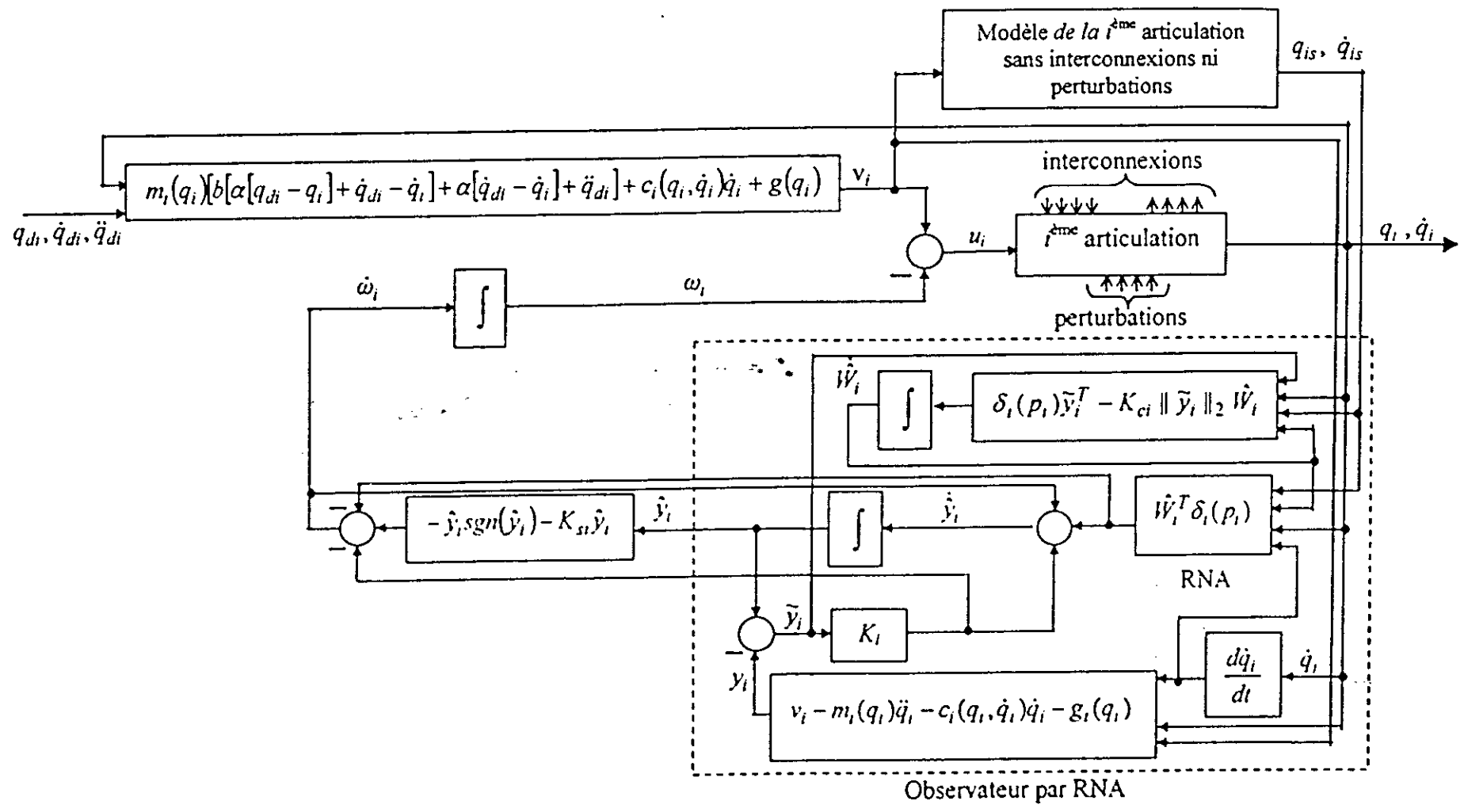


Figure IV.2 Schéma fonctionnel de la loi de commande

IV.5 Résultats de simulation

Nous appliquons cette loi de commande sur un robot manipulateur PUMA 560, en lui imposant une trajectoire cycloïdale normalisée (LEAHVY): Les différentes articulations se déplacent respectivement des positions $\{-50^\circ, -135^\circ, +135^\circ\}$ aux positions $\{+45^\circ, -85^\circ, +30^\circ\}$ en un temps de mouvement égal à $t_f = 2$ sec. Ces conditions excitent toute la dynamique de ce robot.

Nous avons utilisé pour la résolution numérique des systèmes d'équations différentielles non-linéaires, la méthode de Range Kutta du quatrième ordre à pas constant $dt = 0,00035$ sec.

Le modèle mécanique du robot manipulateur PUMA 560 sans interconnexions et sans perturbations est le suivant :

$$m_i(q_{1s})\ddot{q}_{1s} + c_i(q_{1s}, \dot{q}_{1s})\dot{q}_{1s} + g_i(q_{1s}) = v_i \quad (\text{IV.31})$$

articulation 1:
 $m_1(q_{1s}) = 2,57$
 $c_1(q_{1s}, \dot{q}_{1s}) = 0$
 $g_1 = 0$

articulation 2:
 $m_2(q_{2s}) = 6,79$
 $c_2(q_{2s}, \dot{q}_{2s}) = 0$
 $g_2 = -37,2 C_{2s} + 1,02 S_{2s}$

articulation 3:
 $m_3(q_{3s}) = 1,16$
 $c_3(q_{3s}, \dot{q}_{3s}) = 0$
 $g_3 = 0$

Le choix du vecteur de sortie de l'entrée cachée de pré-traitement est, pour chaque articulation, comme suit:

articulation 1:
 $p_1^T = [1 \quad q_1 - q_{1s} \quad \dot{q}_1 - \dot{q}_{1s} \quad q_1 \quad \dot{q}_1 \quad \ddot{q}_1]$

articulation 2:
 $p_2^T = [1 \quad q_2 - q_{2s} \quad \dot{q}_2 - \dot{q}_{2s} \quad q_2 \quad \dot{q}_2 \quad \ddot{q}_2]$

articulation 3:
 $p_3^T = [1 \quad q_3 - q_{3s} \quad \dot{q}_3 - \dot{q}_{3s} \quad q_3 \quad \dot{q}_3 \quad \ddot{q}_3]$

Les valeurs des gains pour chaque articulation i :

$$i = 1, 2, 3:$$

$$K_i = 200, K_{ei} = 0.01, K_{si} = 100, \alpha = 50, h = 10.$$

Plusieurs tests, permettant de juger la qualité de la commande, sont effectués, parmi lesquels figurent ceux de robustesses et de décentralisations. Les résultats de chaque test sont présentés sur deux figures. La première représente les valeurs des interconnexions et des perturbations observées et réelles de chaque articulation ainsi que l'erreur d'observation. La figure est organisée de la manière suivante pour chaque articulation i suivant une colonne :

Premier graphe: valeurs des interconnexions et des perturbations réelles (·) et observées (—)

Deuxième graphe: courbe de l'erreur d'observation.

La deuxième figure présente les positions, les erreurs commises sur celles-ci, les deux commandes v_i et ω_i et la commande globale u_i . Elle comporte pour chaque articulation i suivant une colonne :

Premier graphe: courbe de la trajectoire désirée (-) et de la trajectoire de l'articulation(-.-).
 Deuxième graphe: courbe de l'erreur de consigne.
 Troisième graphe: les courbes des commandes \dot{v}_i (-) et ω_i (-.-).
 Quatrième graphe: courbe de la commande équivalente u_i .

Essai à vide

En premier lieu nous appliquons la loi de commande proposée au robot sans aucune masse, voir figures IV.3 et IV.4.

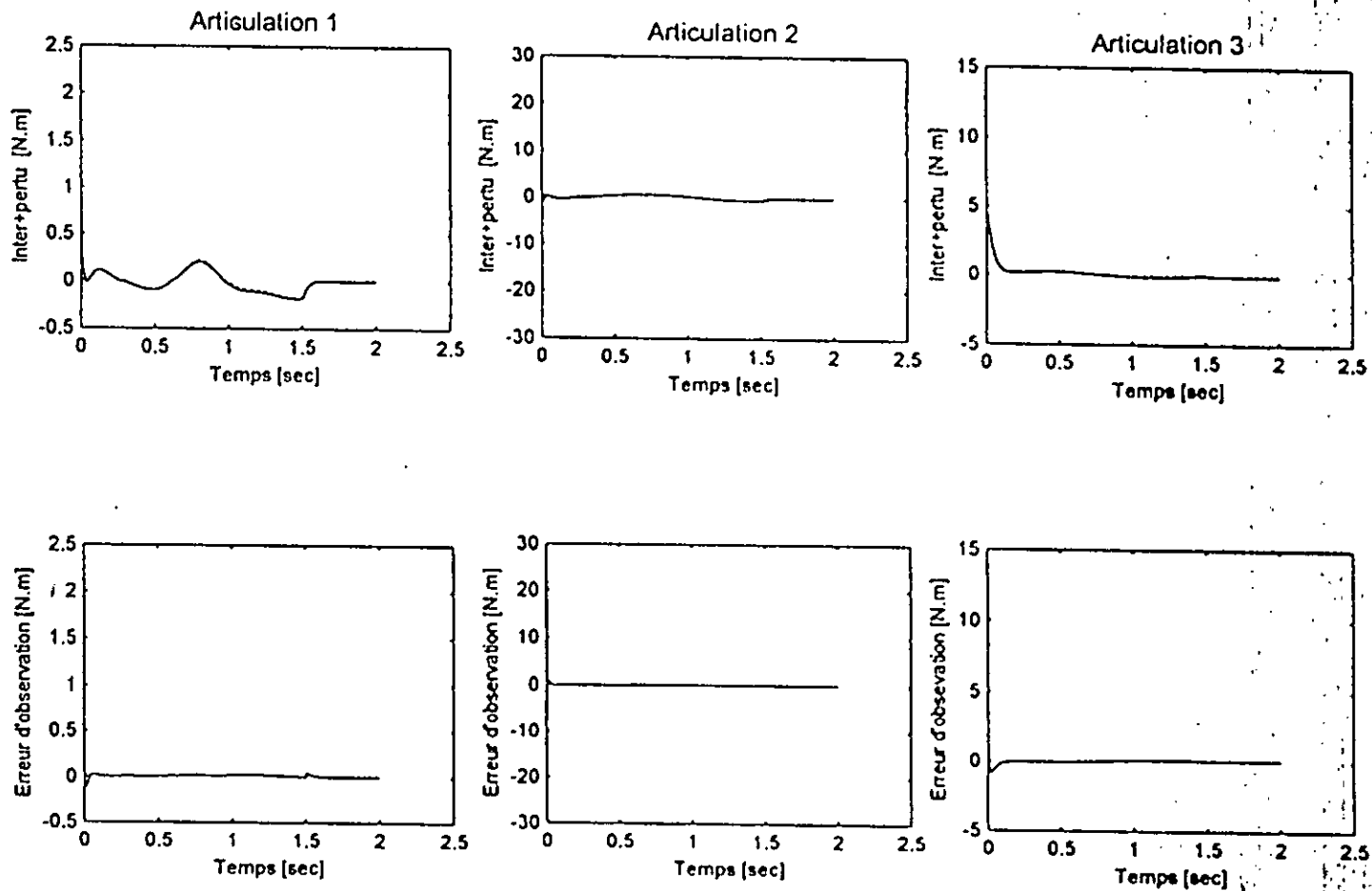


Figure IV.3 Essai à vide, observateur des interconnexions et des perturbations: y_i (-) et \hat{y}_i (-.-)

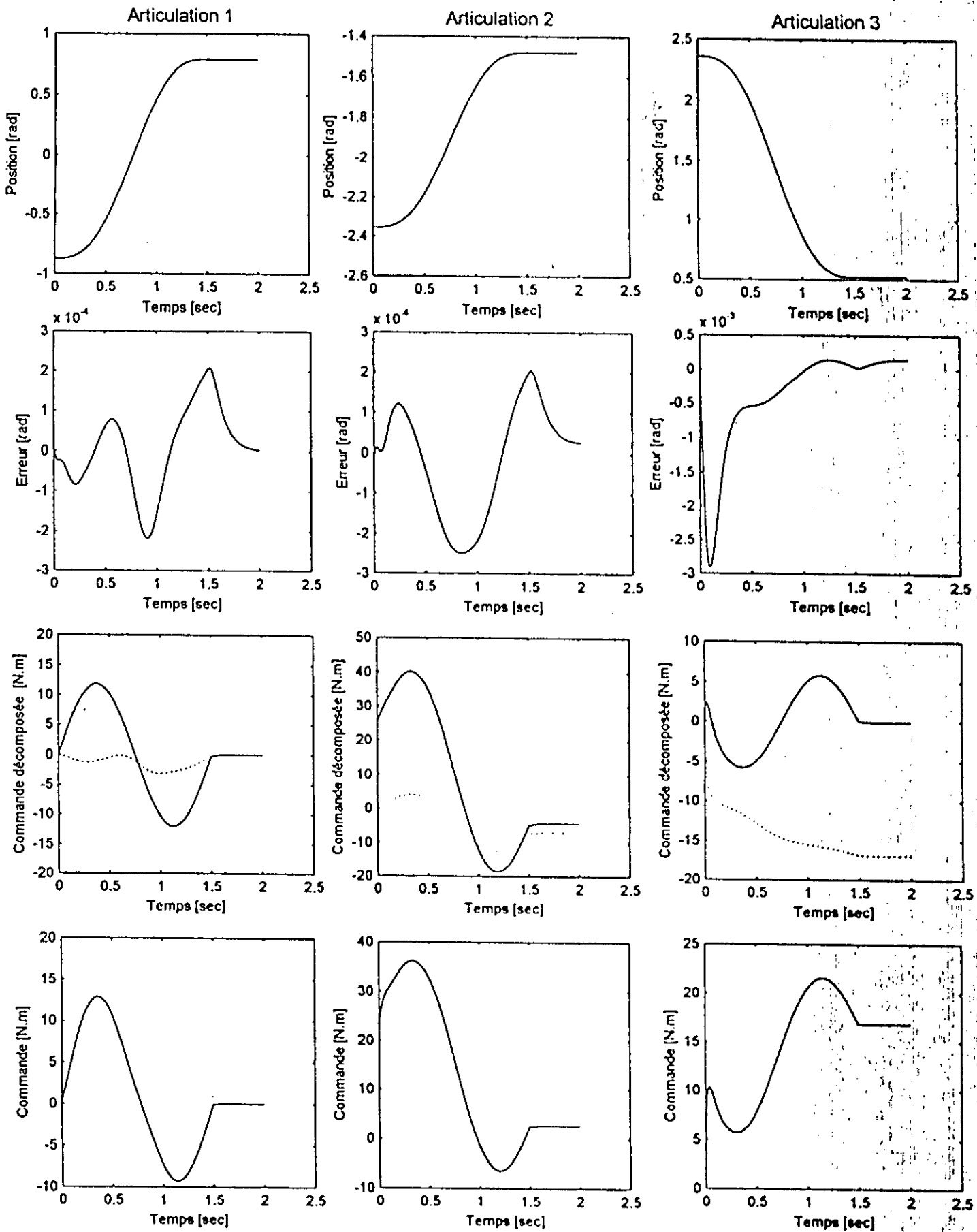


Figure IV.4 Essai à vide
 Position: q_{id} (-), q_i (---). Commande décomposée: v_i (-), α_1 (···).

Interprétation

La Loi de commande donne un très bon résultat. Nous pouvons constater la réduction de l'effet des interconnexions et des perturbations. La commande ω_i est lisse et elle ne s'annule pas malgré que la commande v_i s'annule lorsque les articulations atteignent $\{+45^\circ, -85^\circ, +30^\circ\}$, car elle doit toujours atténuer l'effet des interconnexions et des perturbations.

Essai en charge

Nous aurons deux valeurs de charge à déplacer, une de 5 kg et l'autre de 10 kg. Nous avons choisi deux valeurs au lieu d'une, pour montrer que la qualité de la commande n'est pas influencée par la valeur de la masse, voir figures IV.5 et IV.7 pour $m = 5$ kg et figures IV.6 et IV.8 pour $m = 10$ kg.

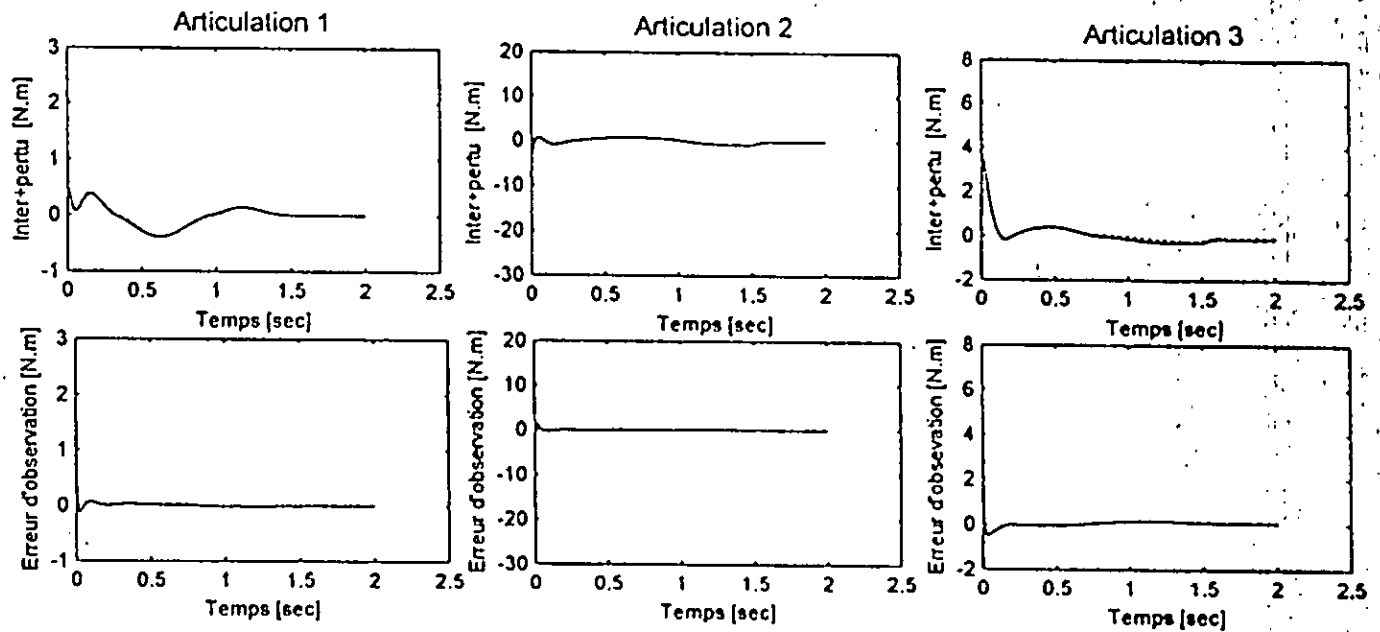


Figure IV.5 Essai en charge avec $m = 5$ kg, observateur des interconnexions et des perturbations: y_i , (\cdot) et \hat{y}_i , $(-)$

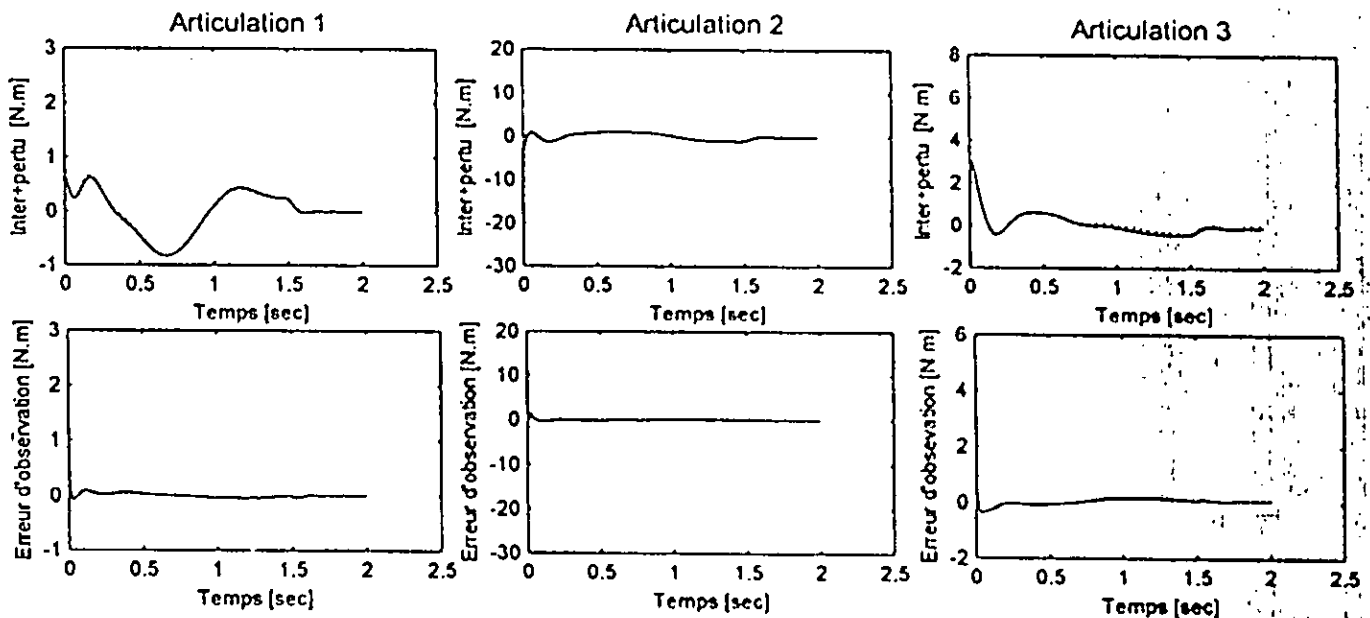


Figure IV.6 Essai en charge avec $m = 10$ kg, observateur des interconnexions et des perturbations: y_i , (\cdot) et \hat{y}_i , $(-)$

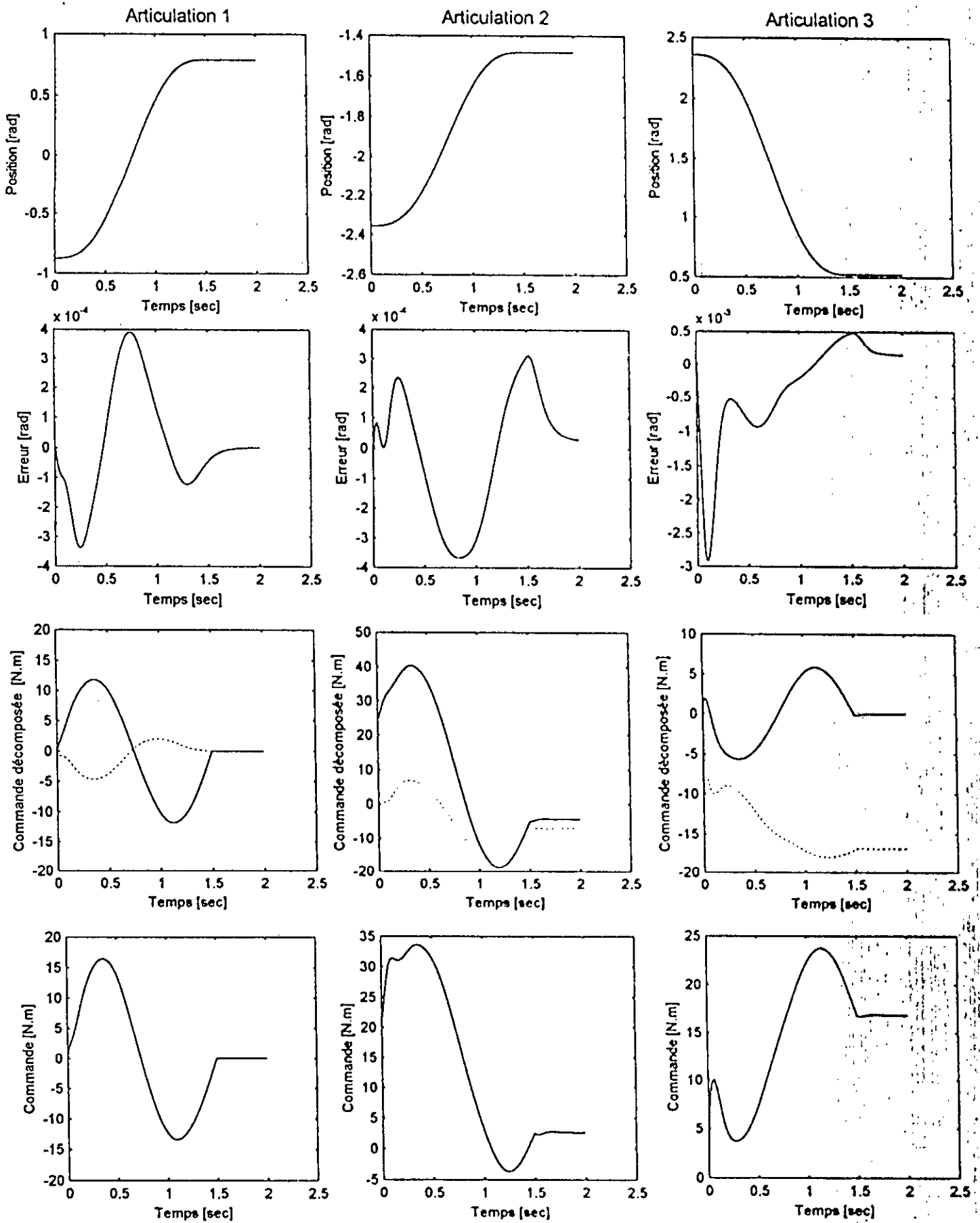


Figure IV.7 Essai en charge avec $m = 5$ kg.
 Position: q_{1d} (-), q_1 (---). Commande décomposée: v_1 (-), α_1 (··).

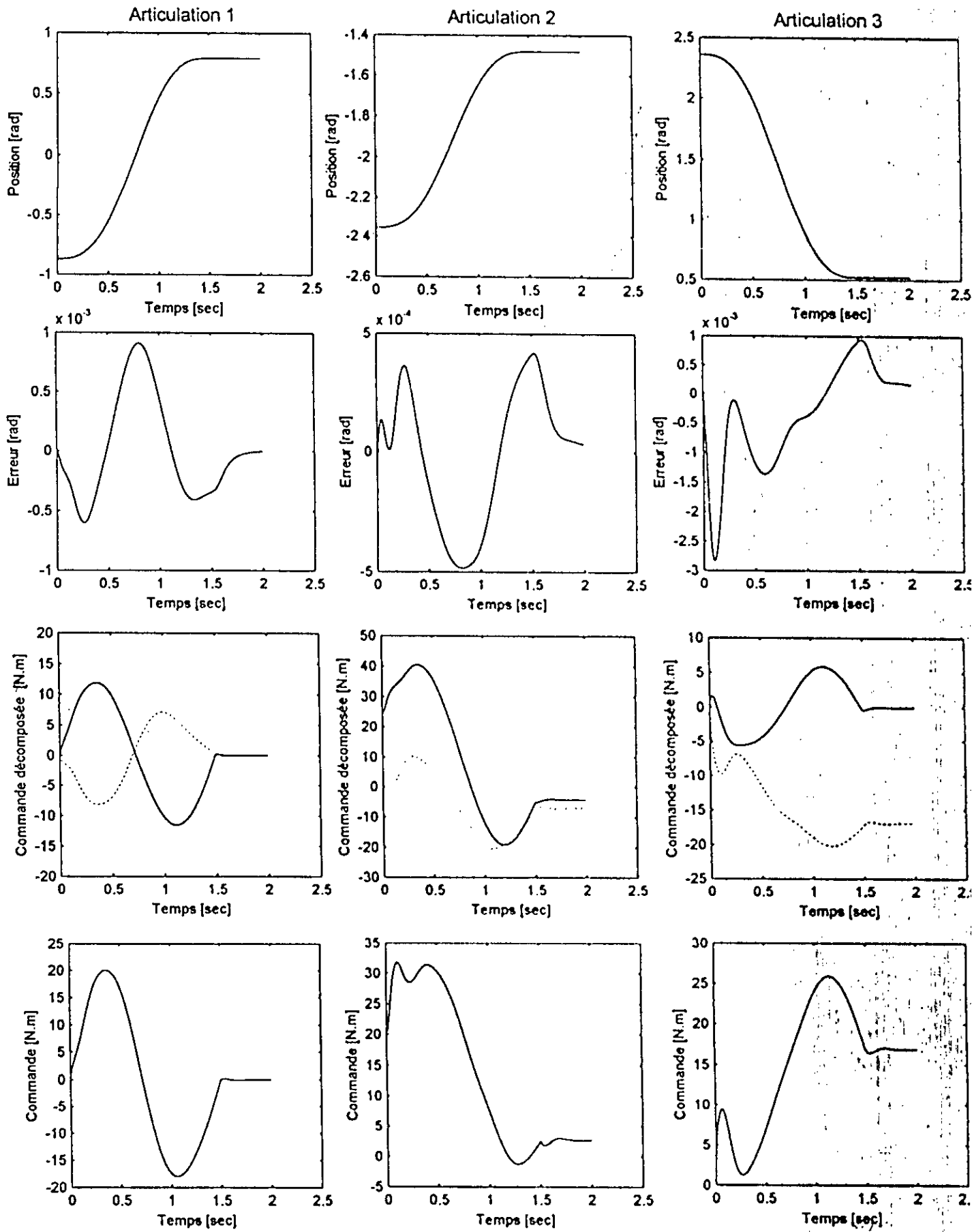


Figure IV.8 Essai en charge avec $m = 10$ kg.
 Position: q_{1d} (-), q_1 (---). Commande décomposée: v_1 (-), ω_1 (---).

Interprétation

D'après les figures IV.7 et IV.8, la commande s'applique bien pour des valeurs différentes de la masse. L'effet de la charge qui est vu comme étant une perturbation, en figures IV.5 et IV.6, va être atténué par la commande ω_i .

Essai de robustesse

Nous effectuons deux types d'essais de robustesse. Le premier correspond à la chute de la masse (charge $m = 10$ kg) à l'instant $t = 0,75$ sec, figures IV.9 et IV.11, et le deuxième essai consiste en une variation des valeurs des paramètres du robot de 50% à l'instant $t = 0,75$ sec, figures IV.10 et IV.12.

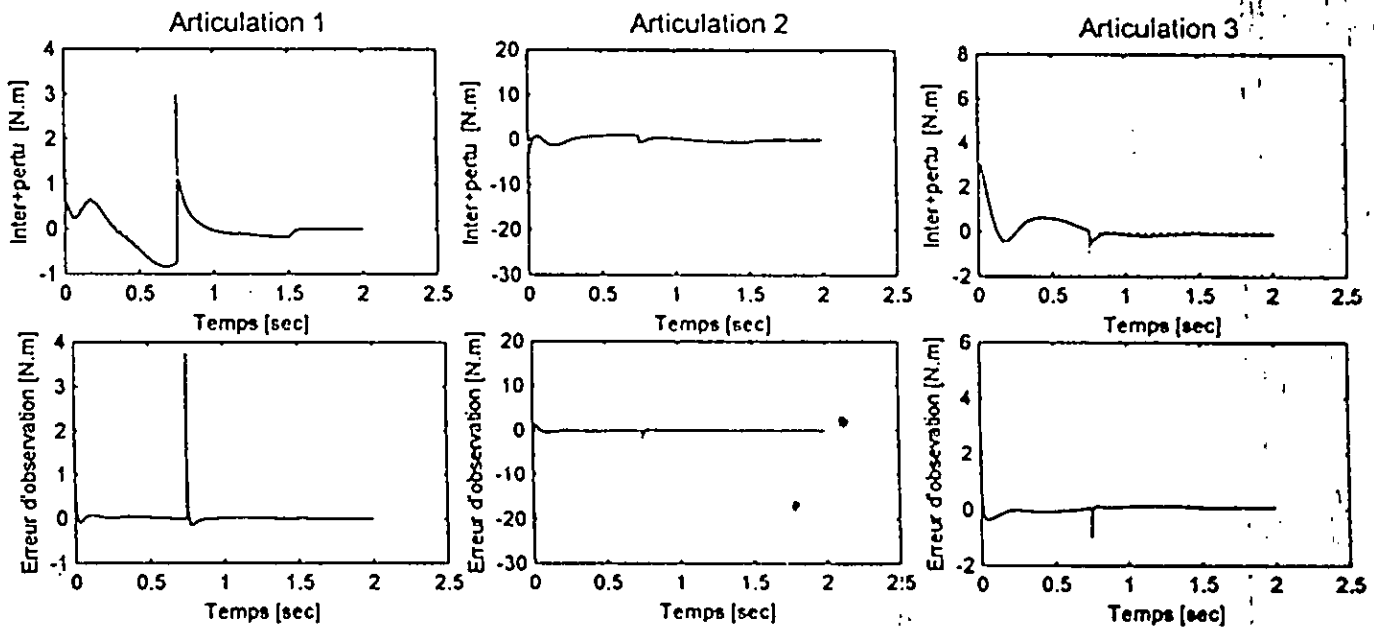


Figure IV.9 Essai de robustesse, chute de masse ($m = 10$ kg) à $t = 0,75$ sec, observateur des interconnexions et des perturbations: $y_i(\cdot)$ et $j_i(-)$

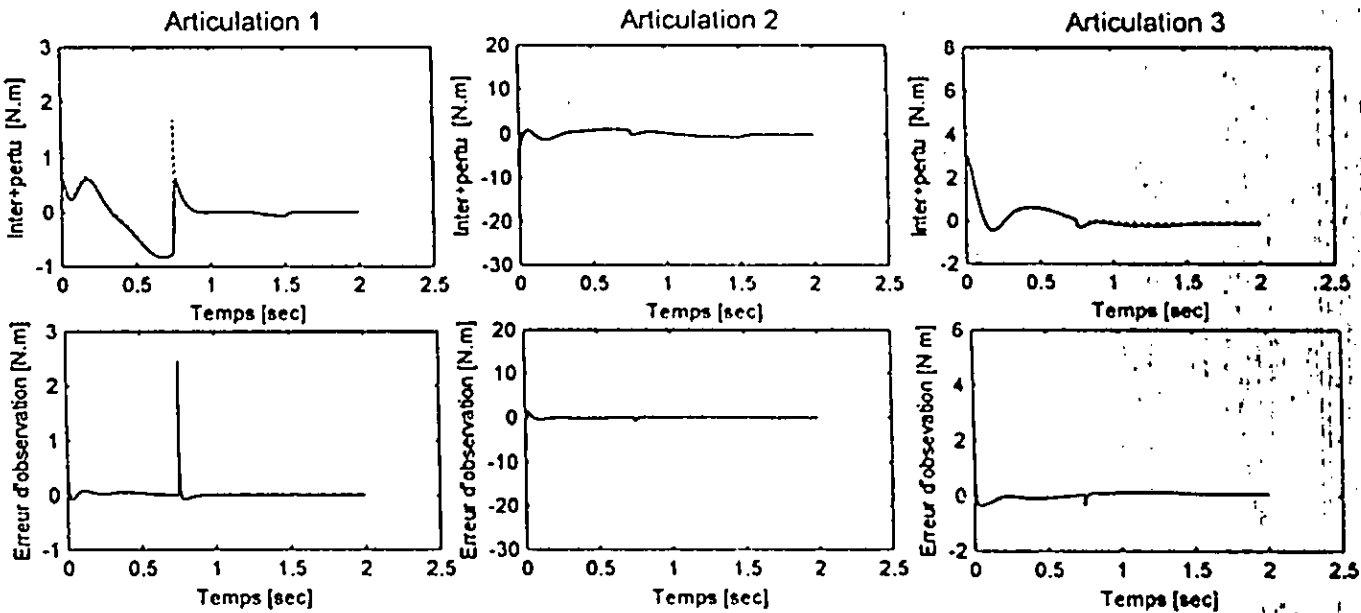


Figure IV.10 Essai de robustesse, changement paramétrique de 50% à $t = 0,75$ sec, observateur des interconnexions et des perturbations: $y_i(\cdot)$ et $j_i(-)$

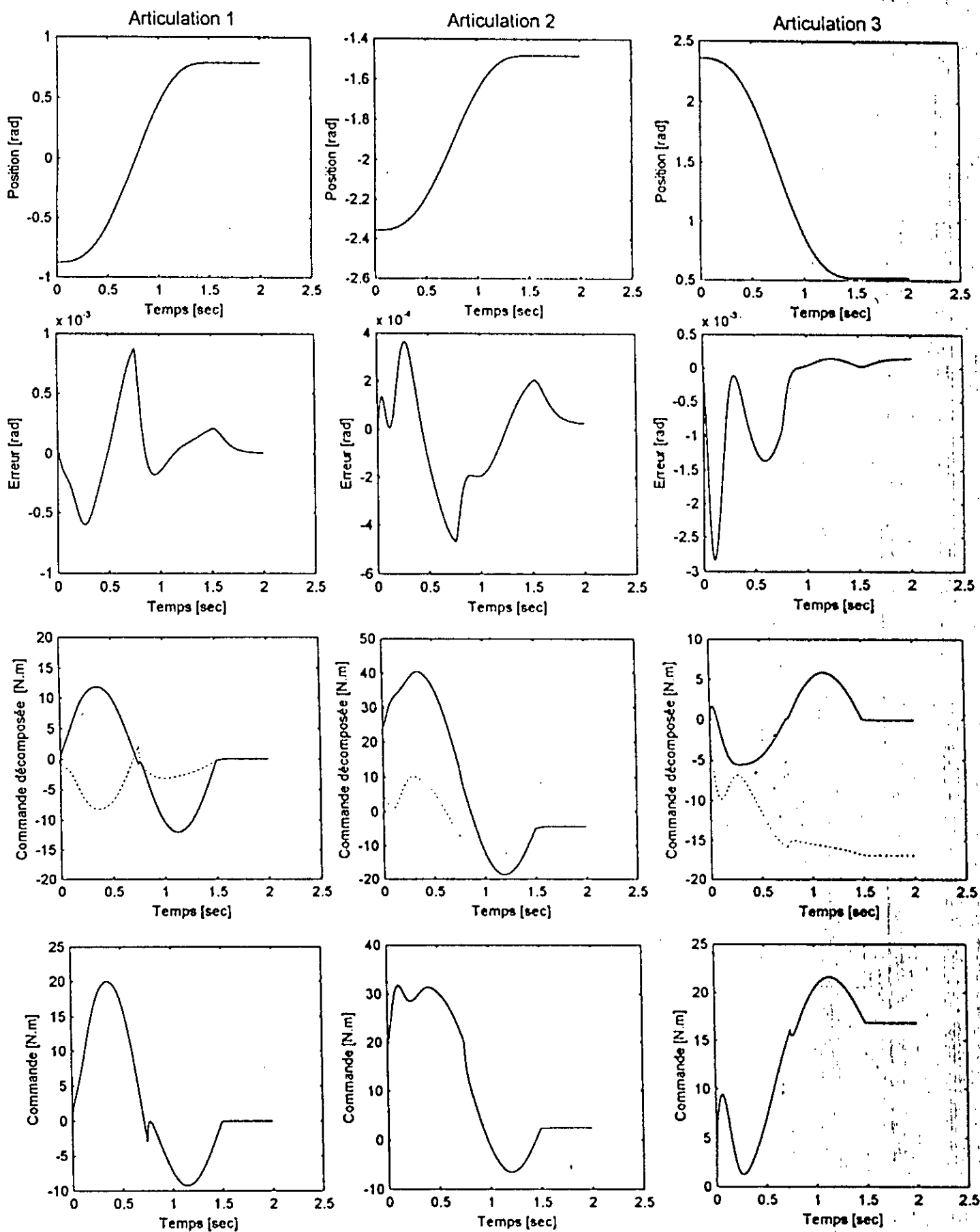


Figure IV.11 Essai de robustesse, chute de masse ($m = 10$ kg) à $t = 0,75$ sec.
 Position: q_{id} (-), q_i (---). Commande décomposée: v_i (-), α_i (···).

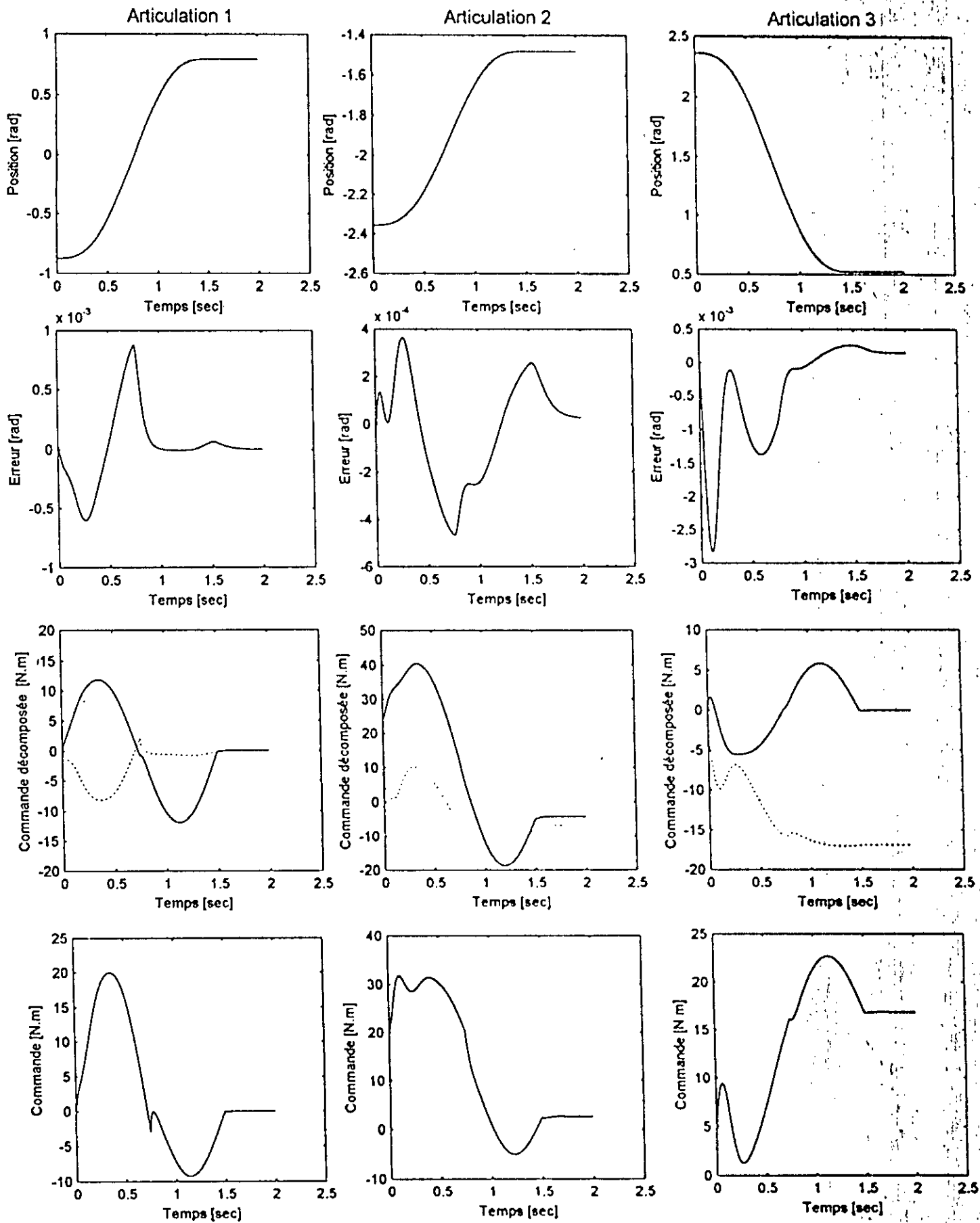


Figure IV.12 Essai de robustesse, changement paramétrique de 50% à $t = 0,75$ sec.
 Position: q_{id} (-), q_i (---). Commande décomposée: v_i (-), ω_i (---).

Interprétation

Les deux types de test ont prouvé la robustesse de la loi de commande vis-à-vis des perturbations. A l'instant $t = 0,75$ sec, nous remarquons le changement de la valeur de y_i et de la réaction de la commande ω_i comme réponse à ce changement. L'observateur suit parfaitement y_i , ce qui garantit l'efficacité de la commande ω_i .

Essai de décentralisation

Nous avons réalisé deux types d'essais de décentralisation, le premier à vide et le second avec une masse de 10 kg. A l'instant $t = 0,75$ sec, nous avons une interruption de la commande associée à la deuxième articulation (u_2), voir figures IV.13 et IV.15 pour l'essai à vide et figures IV.14 et IV.16 pour $m = 10$ kg.

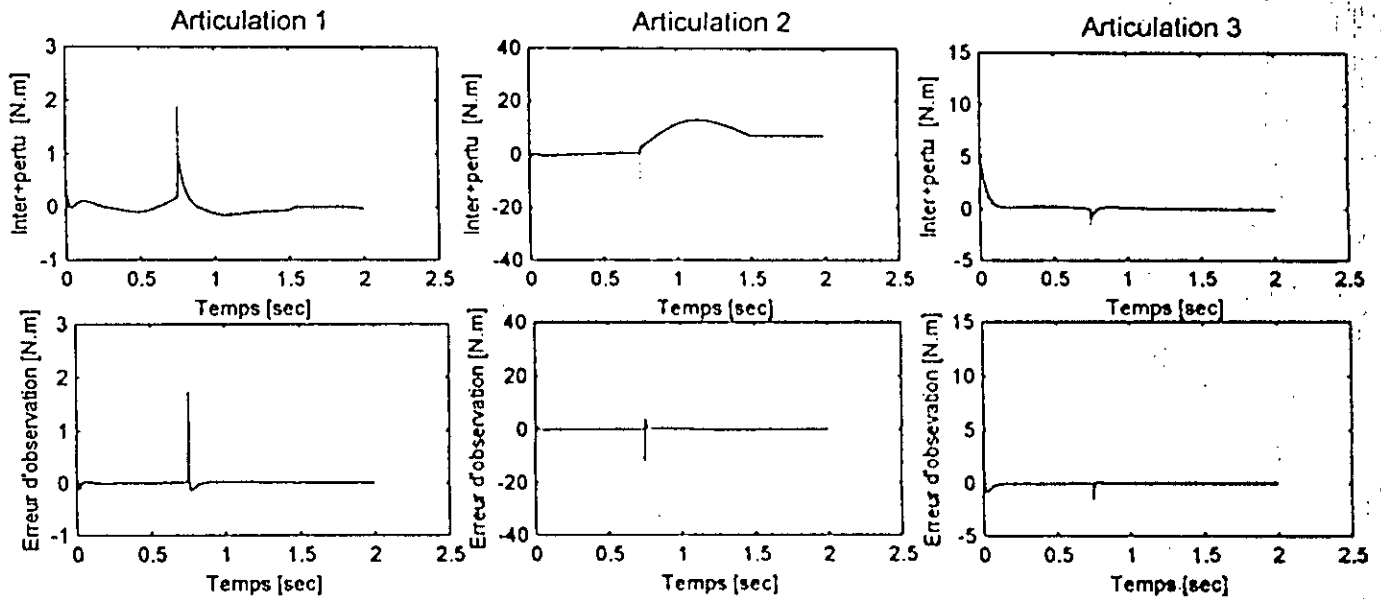


Figure IV.13 Essai de décentralisation, à vide où $u_2 = 0$ à $t = 0,75$ sec, observateur des interconnexions et des perturbations: y_i (·) et \hat{y}_i (-)

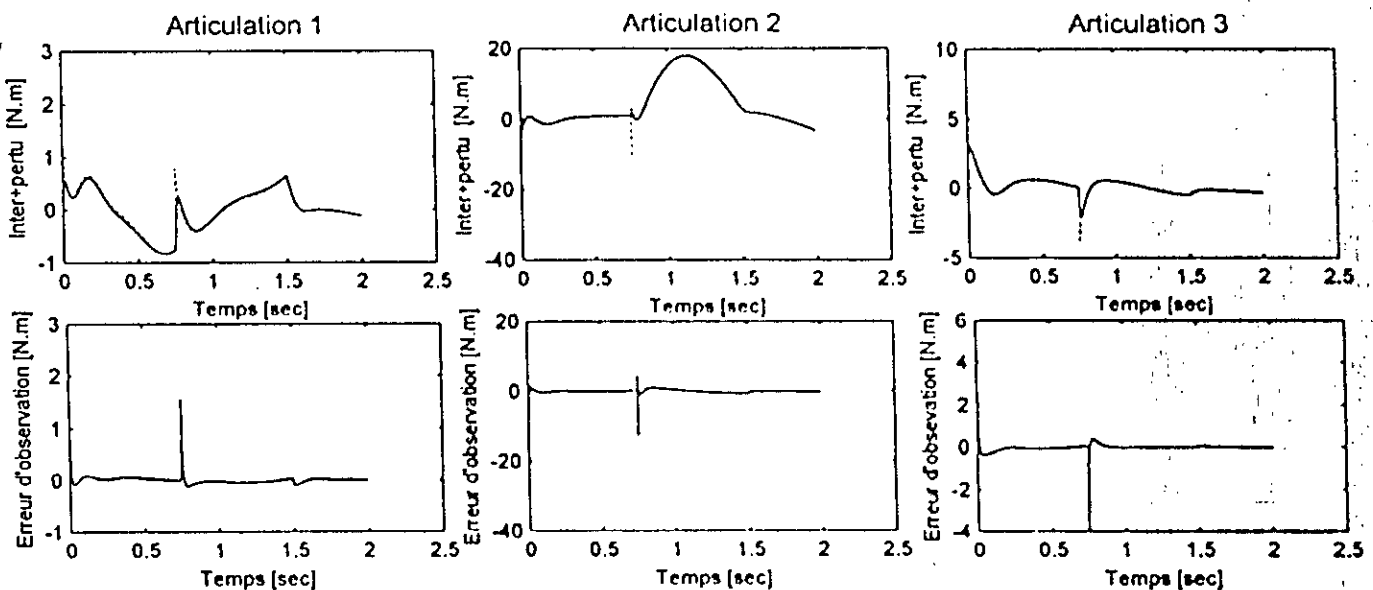


Figure IV.14 Essai de décentralisation, en charge ($m = 10$ kg) où $u_2 = 0$ à $t = 0,75$ sec, observateur des interconnexions et des perturbations: y_i (·) et \hat{y}_i (-)

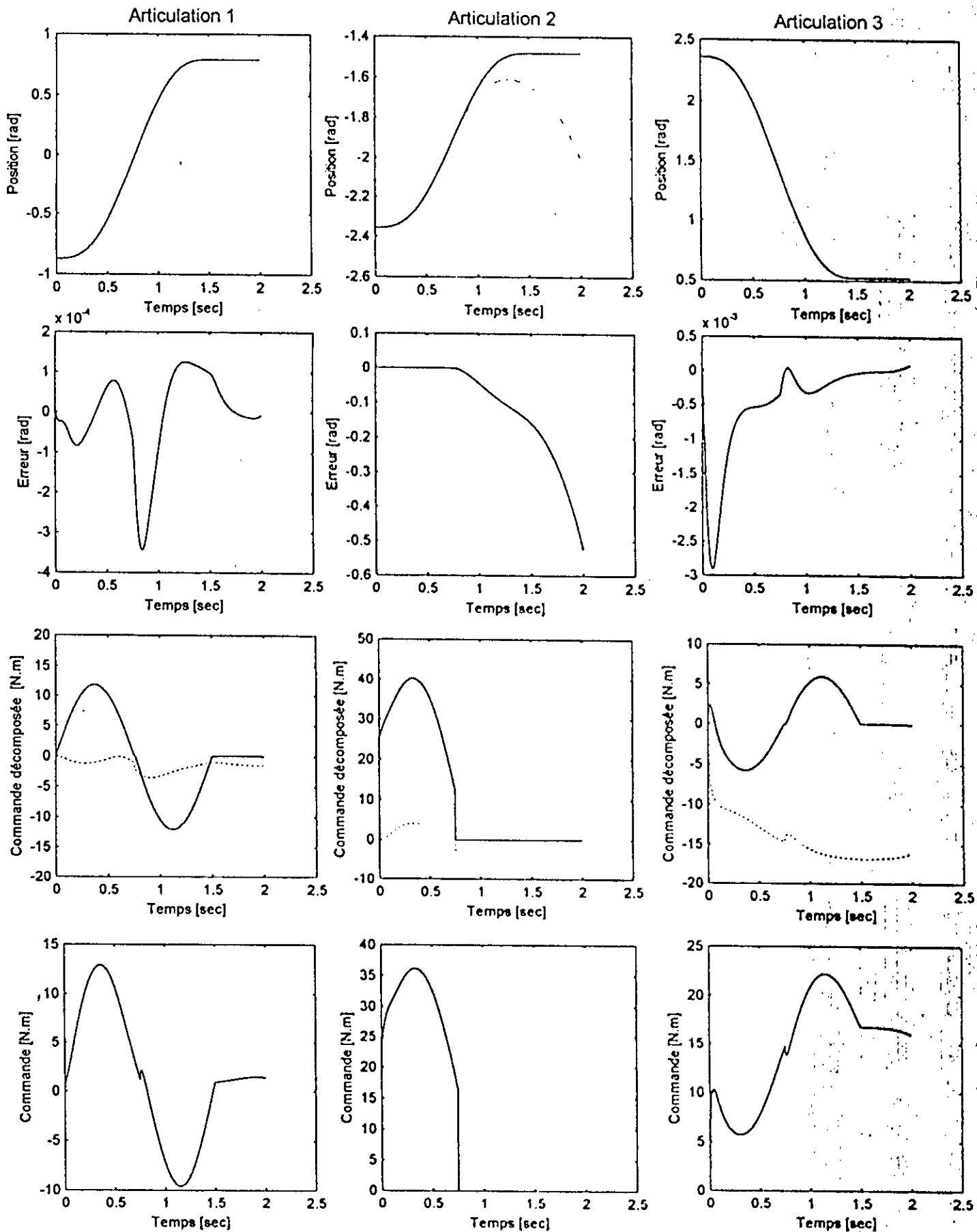


Figure IV.15 Essai de décentralisation, à vide où $u_2 = 0$ à $t = 0,75$ sec.
 Position: q_{1d} (-), q_1 (---). Commande décomposée: v_1 (-), q_{1d} (---).

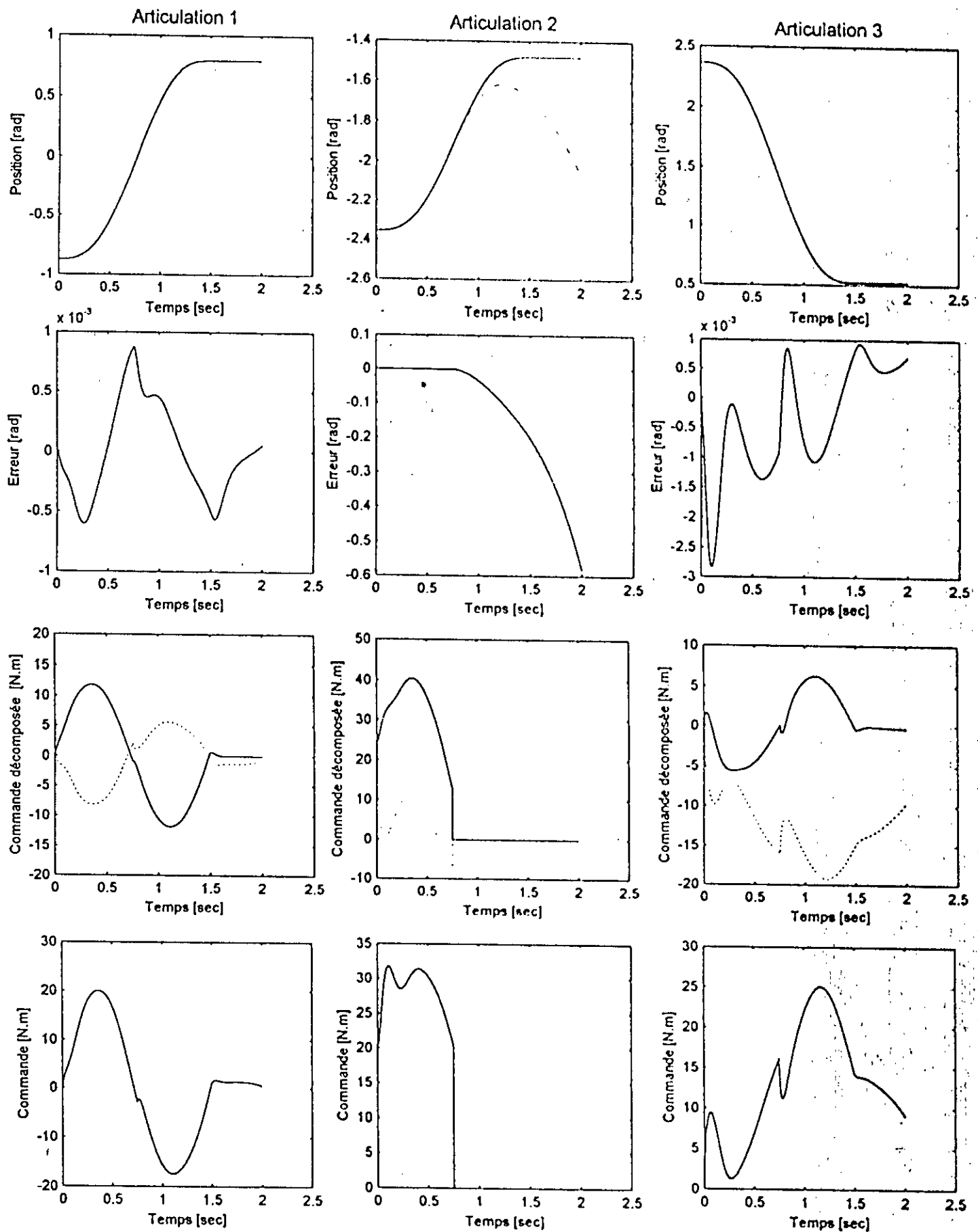


Figure IV.16 Essai de décentralisation, en charge ($m = 10$) où $u_2 = 0$ à $t = 0,75$ sec.
 Position: q_{1d} (-), q_1 (---). Commande décomposée: v_1 (-), α_1 (---).

Interprétation

Puisque les interconnexions sont prises comme étant des perturbations et leurs valeurs nous sont restituées grâce à l'observateur, donc les réactions des commandes de la première et de la deuxième articulation se font sentir de la même manière que dans le cas de la chute de la masse ou du changement paramétrique.

Essai du cas extrême

Cet essai réunit les différents essais précédents en un seul test, pour vérifier la fiabilité de la commande dans le cas extrême, figures IV.17 et IV.18. Il comprend l'interruption de la commande u_1 et u_3 avec la chute de la masse (10 kg) à l'instant $t = 0,75$ sec.

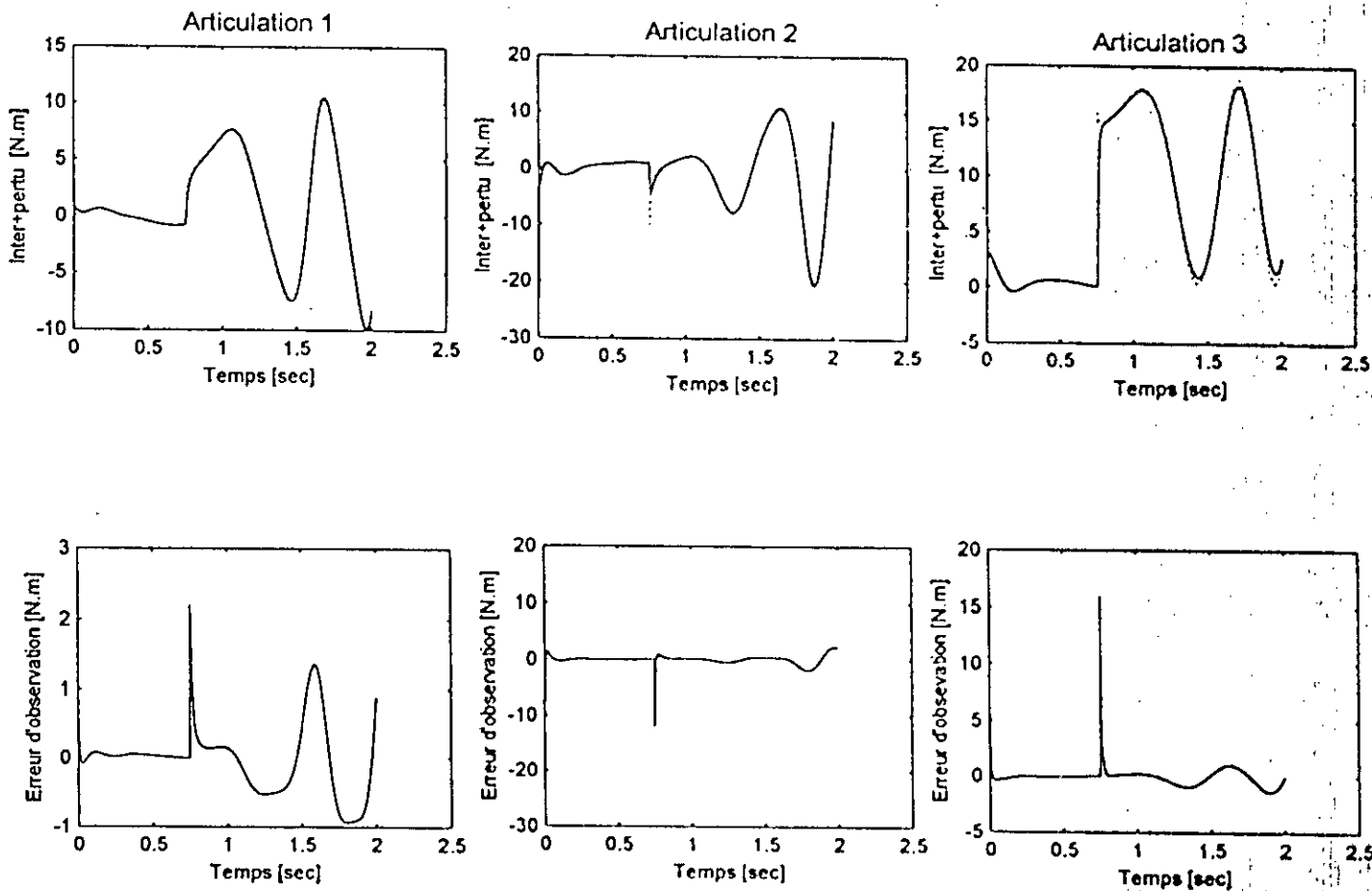


Figure IV.17 Essai du cas extrême, chute de masse ($m = 10$ kg) et $u_1 = u_3 = 0$ à $t = 0,75$ sec, observateur des interconnexions et des perturbations: y_i (—) et \hat{y}_i (- -)

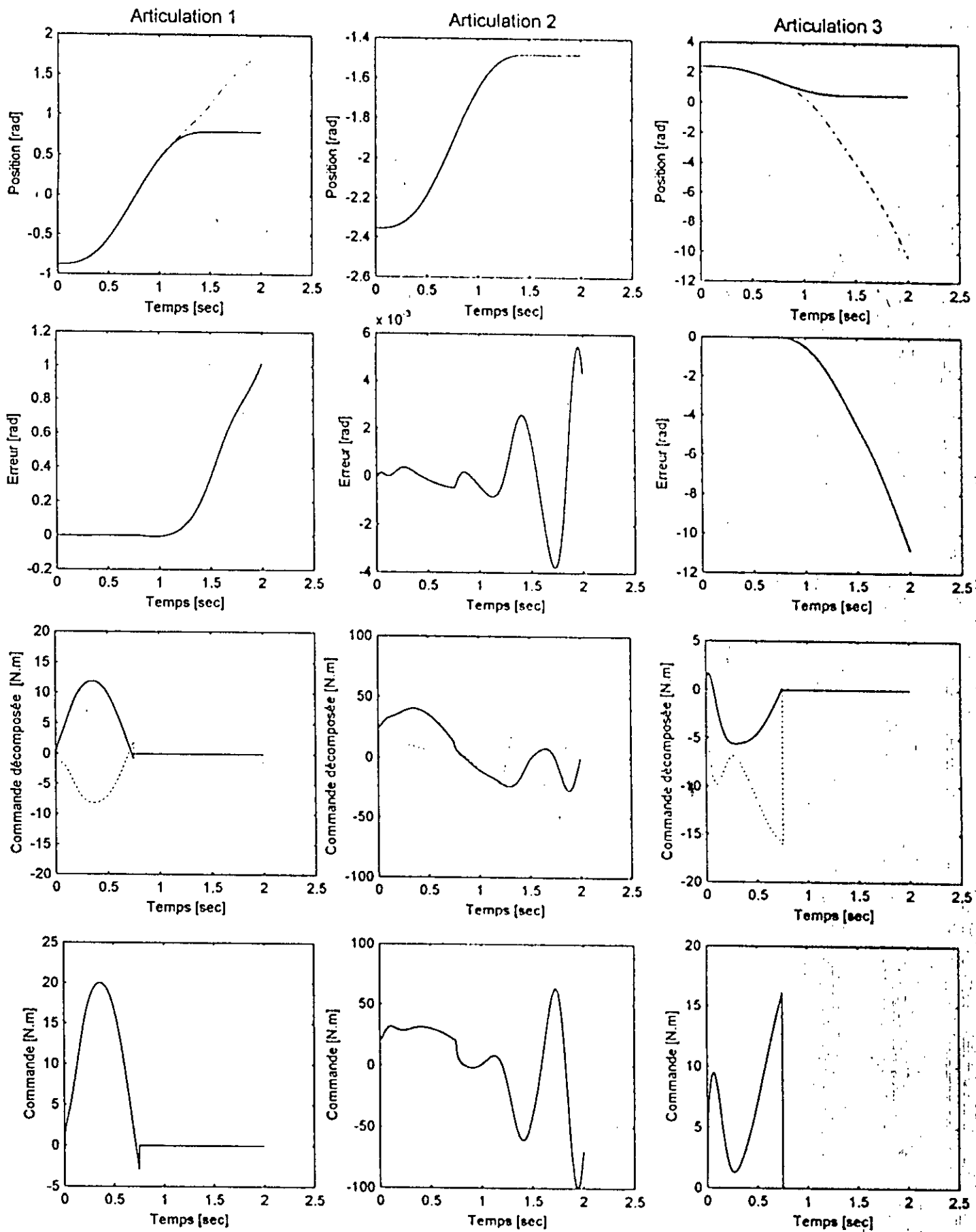


Figure IV.18 Essai du cas extrême, chute de masse ($m = 10$) et $u_1 = u_2 = 0$ à $t = 0,75$ sec.
 Position: q_{id} (-), q_i (---). Commande décomposée: v_i (-), α_i (---).

Interprétation

Ce test ne fait que confirmer la qualité de la commande déjà prouvée par les essais précédents.

IV.6 Conclusion

Dans ce travail, nous proposons une commande robuste décentralisée pour un système mécanique fortement interconnecté modélisé par les équations d'Euler-Lagrange. L'utilisation d'un observateur par réseau de neurones a permis d'estimer les valeurs des interconnexions et des perturbations afin d'atténuer leur effet par une commande par mode glissant généralisé. Ainsi, la synthèse de la commande pour chaque articulation devient simple. La méthode de Backstepping assure une convergence rapide de l'erreur de poursuite vers zéro. La qualité de la commande a été démontrée par simulation sur le robot PUMA 560, comportant plusieurs tests de robustesse.

CHAPITRE V
ETUDE COMPARATIVE

Dans cette partie de notre travail, nous comparons notre loi de commande à deux méthodes : le régulateur Proportionnel Dérivé, très utilisé en industrie [52] et une loi de commande récente, celle de Y. Tang et coll. [14]

Avant d'entamer cette étude, nous synthétisons un régulateur Proportionnel Dérivé, en utilisant le placement de pôles, qui est une méthode connue et introduite dans plusieurs ouvrages [23].

La méthode de Y. Tang et coll. [14] nécessite une présentation de la loi de commande que nous donnons brièvement ci-dessous, les détails de démonstration de la stabilité et de la convergence sont données dans [14].

V.1 Régulation du PUMA 560 par un Proportionnel Dérivé (PD)

La régulation par le Proportionnel Dérivé est la plus utilisée en industrie au vue de son coût économique et de sa facilité d'implémentation, mais quand l'application nécessite une erreur très faible et/ou un large domaine de fonctionnement, nous serons alors obligés de recourir à d'autres formes de commande plus complexes et plus coûteuses.

Pour le robot PUMA 560, qui comporte trois articulations, nous utilisons donc trois régulateurs Proportionnels Dérivés, un pour chaque articulation. Afin de pouvoir appliquer le régulateur Proportionnel Dérivé, nous devons linéariser le modèle du robot PUMA 560 autour d'un point d'équilibre.

a- Linéarisation du modèle du PUMA 560

Soit le modèle dynamique du robot PUMA 560 :

$$M(q)\ddot{q} + B(q, \dot{q})\dot{q} + G(q) = u - \Gamma \quad (\text{V.1})$$

avec Γ l'effet de la masse, et u la commande .

Au point d'équilibre nous avons $\dot{q} = 0$ et $\ddot{q} = 0$, l'équation (V.1) devient :

$$G(q_0) = u_0 - \Gamma_0 \quad (\text{V.2})$$

On prend $\Gamma_0 = 0$ (pas de masse), donc on aura :

$$u_0 = G(q_0) \quad (\text{V.3})$$

Calculons le point d'équilibre pour chaque articulation :

$$\begin{aligned} q_1 \in [-50^\circ, 45^\circ] &= [-5\pi/18, \pi/4] \Rightarrow q_1^* = -\pi/72 \\ q_2 \in [-135^\circ, -85^\circ] &= [-3\pi/4, -17\pi/36] \Rightarrow q_2^* = -11\pi/18 \\ q_3 \in [135^\circ, 30^\circ] &= [3\pi/4, \pi/6] \Rightarrow q_3^* = 11\pi/24 \end{aligned} \quad (\text{V.4})$$

En appliquant les points d'équilibre à l'équation (V.3) nous obtenons :

$$u_0 = \begin{bmatrix} 0 \\ 15.64 \\ 4.1 \end{bmatrix} \quad (\text{V.5})$$

La linéarisation du modèle (V.1) autour d'un point d'équilibre nous donne :

$$\delta u_i = \sum_{j=1}^3 M_{ij}^* \delta j_j + \sum_{j=1}^3 \frac{\partial \dot{x}_i}{\partial j_j} \delta j_j = M_i^* \delta j_i + \frac{\partial \dot{x}_i}{\partial j_i} \delta j_i + F \quad (\text{V.6})$$

$$\delta u_i = u_i - u_i^*$$

$$\text{avec : } \delta \ddot{q}_i = \ddot{q}_i$$

$$\delta q_i = q_i - q_i^*$$

L'équation (V.6) nous donne:

$$\delta u_1 = m_{11}^* \delta \ddot{q}_1 + (\partial \dot{x}_1 / \partial j_1) \delta j_1 \quad (\text{V.7})$$

$$\delta u_2 = m_{22}^* \delta \ddot{q}_2 + (\partial \dot{x}_2 / \partial j_2) \delta j_2 \quad (\text{V.8})$$

$$\delta u_3 = m_{33}^* \delta \ddot{q}_3 + (\partial \dot{x}_3 / \partial j_3) \delta j_3 \quad (\text{V.9})$$

Etablissons maintenant la représentation d'état du système précédent :

$$x_{11} = \delta j_1 \text{ et } x_{21} = \delta \dot{q}_1 \Rightarrow X_1^T = [x_{11}, x_{21}] \quad (\text{V.10})$$

$$x_{12} = \delta j_2 \text{ et } x_{22} = \delta \dot{q}_2 \Rightarrow X_2^T = [x_{12}, x_{22}] \quad (\text{V.11})$$

$$x_{13} = \delta j_3 \text{ et } x_{23} = \delta \dot{q}_3 \Rightarrow X_3^T = [x_{13}, x_{23}] \quad (\text{V.12})$$

Le système linearisé est mis sous la forme :

$$\begin{cases} \dot{X} = AX + BU \\ Y = CX \end{cases} \quad (\text{V.13})$$

avec :

$$A_i = \begin{bmatrix} 0 & \vdots & 1 \\ \dots & \dots & \dots \\ -m_i^{*-1} \left(\frac{\partial \dot{x}_i}{\partial j_i} \right) & \vdots & 0 \end{bmatrix} \quad (\text{a}) \quad (\text{V.14})$$

$$B_i = \begin{bmatrix} 0 \\ \dots \\ m_i^{*-1} \end{bmatrix} \quad (\text{b})$$

$$C_i = [1 \quad \vdots \quad 0] \quad (\text{c})$$

Les fonctions de transfert de chaque articulation seront calculées par l'équation suivante :

$$G_i(s) = C (sI - A_i)^{-1} B_i \quad (\text{V.15})$$

On aura donc:

$$\dot{X}_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} X_1 + \begin{bmatrix} 0 \\ 0.39 \end{bmatrix} U_1$$

$$\dot{X}_2 = \begin{bmatrix} 0 & 1 \\ 5.65 & 0 \end{bmatrix} X_2 + \begin{bmatrix} 0 \\ 0.13 \end{bmatrix} U_2 \quad (\text{V.16})$$

$$\dot{X}_3 = \begin{bmatrix} 0 & 1 \\ 6.32 & 0 \end{bmatrix} X_3 + \begin{bmatrix} 0 \\ 0.86 \end{bmatrix} U_3$$

Les fonctions de transferts seront comme suit:

$$G_1 = \frac{39}{100s^2} \quad (a)$$

$$G_2 = \frac{13}{100s^2 - 565} \quad (b) \quad (V.17)$$

$$G_3 = \frac{43}{50s^2 - 316} \quad (c)$$

b- Calcul des paramètres du régulateur Proportionnel Dérivé (PD)

Pour le calcul des paramètres du régulateur, nous avons utilisé la méthode d'imposition des pôles [23], les pôles choisis sont :

$$p_1 = -20 - 20j \text{ et } p_2 = -20 + 20j \quad (V.18)$$

La fonction de transfert d'un régulateur proportionnel dérivé est :

$$p(s) = K(1 + T_D s) \quad (V.19)$$

Pour la première articulation on a :

$$p_1(s) = K_1(1 + T_1 s) \quad (V.20)$$

Posons :

$$(39/100) K_1 = K \quad (V.21)$$

Appliquons le PD à la première articulation, à partir de l'équation (V.17a), (V.20) et (V.21) on aura :

$$F_1(s) = \frac{K(1 + T_1 s)}{s^2 + K(1 + T_1 s)} \quad (V.22)$$

avec $F_1(s)$ fonction de transfert de la première articulation avec régulateur, en boucle fermée :

$$F_1(s) = \frac{P_1}{1 + G_1} \quad (V.23)$$

Nous avons donc :

$$s^2 + K T_1 s + K = (s + 20 + 20j)(s + 20 - 20j) = s^2 + 40s + 800 \quad (V.24)$$

$$\Rightarrow K = 800 \text{ et } K_1 T_1 = 40 \Rightarrow \boxed{T_1 = 0.05}$$

de l'équation (V.21) nous avons : $\boxed{K_1 = 2051.28}$

De la même manière, nous obtenons les paramètres du régulateur pour les deux articulations restantes :

$$\boxed{K_2 = 6197.3 \text{ et } T_2 = 0.05}$$

$$\boxed{K_3 = 937.58 \text{ et } T_3 = 0.05}$$

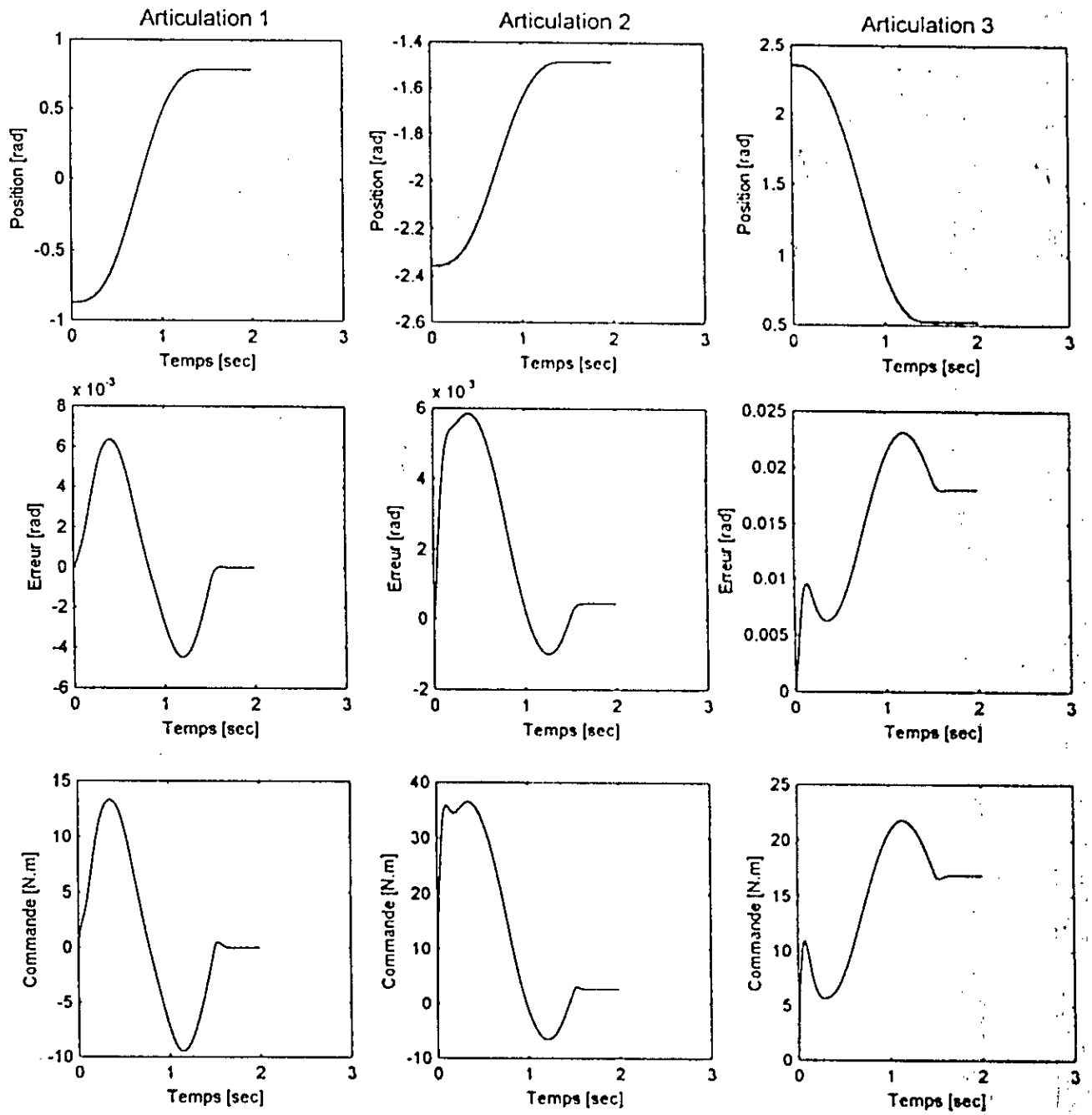


Figure V.1 Essai à vide, en régulation PD

Position: q_{id} (-), q_i (-.-).

Les réponses obtenues à partir des résultats précédents pour un essai à vide sont représentés en figure V.1.

V.2 Présentation générale de la méthode Y. Tang et coll. [14]

Cette méthode est appliquée à un système mécanique décrit par les équations d'Euler-Lagrange et fortement interconnecté. Elle propose une loi de commande décentralisée qui garantit une stabilité uniforme dont l'objectif est d'imposer une trajectoire lisse désirée. La synthèse de la commande est basée sur les propriétés physiques du système et la théorie de stabilité de Lyapunov [53]. Nous présentons d'une façon générale la loi de commande puis un exemple d'application sur deux pendules couplés inversés, avant de la comparer à notre méthode sur le robot PUMA 560.

Remarque: Nous appliquons notre loi de commande sur deux pendules couplés inversés à titre de vérification.

Loi de commande Y. Tang et coll.

Soit le système mécanique vu d'une manière décentralisée et défini par:

$$M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + g(q_i) = \Gamma_i \quad (V.25)$$

(V.25) est l'équation du $i^{ème}$ sous système, $i \in [1, N]$

Avec :

$M_i(q_i) \in \mathfrak{R}^{n_i \times n_i}$ (Matrice d'inertie) ; $q_i(t) \in \mathfrak{R}^{n_i}$ (coordonnées généralisées) ;
 $C_i(q_i, \dot{q}_i)\dot{q}_i$ (Les couples dus aux forces de coriolis et centrifuges) ; $g(q_i)$ (forces et/ou couples dus aux forces de gravitation) ; $\Gamma_i = u_i - Z_i(q_i, \dot{q}_i) - D_i(q_i, \dot{q}_i)$; avec $u_i \in \mathfrak{R}^{n_i}$:
 vecteur de commande {force/couple} $q^r = [q_1^r \ q_2^r \ \dots \ q_N^r]^T \in \mathfrak{R}^n$ avec $n = \sum_{i=1}^N n_i$; $Z_i(q, \dot{q})$:
 vecteur interconnexion ; $D_i(q_i, \dot{q}_i)$: vecteur perturbation.

On note $(.)_i^o$ la valeur nominale de $(.)_i$. La commande décentralisée appliquée à chaque sous système (V.25):

$$u_i = M_i^o(q_i)\ddot{q}_i^r + C_i^o(q_i, \dot{q}_i)\dot{q}_i^r + g_i^o(q_i) - K_i s_i + \omega_i \quad (V.26)$$

Où \dot{q}_i^r est la vitesse de référence et s_i est une variable glissante. Elles sont définies comme suit:

$$\dot{q}_i^r = \dot{q}_i^d - A_i \tilde{q}_i, \quad s_i = \tilde{\dot{q}}_i + A_i \tilde{q}_i = \dot{q}_i - \dot{q}_i^r \quad (V.27)$$

$q_i^d(t) \in \mathfrak{R}^{n_i}$ est la trajectoire désirée, q_i^d , \dot{q}_i^d et \ddot{q}_i^d sont des trajectoires lisses et bornées. On définit l'erreur de poursuite comme étant $\tilde{q}_i(t) = q_i(t) - q_i^d(t)$.

$K_i, A_i \in \mathfrak{R}^{n_i \times n_i}$ sont définies positives et $\omega_i \in \mathfrak{R}^{n_i}$ est synthétisé pour compenser les incertitudes, les perturbations et les interconnexions.

$$\omega_i = -(\hat{\delta}_i S_i)^2 \frac{S_i}{\hat{\delta}_i S_i \|s_i\| + \varepsilon_i} \quad (\text{V.28})$$

$$\dot{\hat{\delta}}_i = -\sigma_i \hat{\delta}_i + \gamma_i S_i \|s_i\|, \quad \hat{\delta}_i(t_0) > 0 \quad (\text{V.29})$$

$$S_i = 1 + \|s_i\| + \dots + \|s_i\|^p \quad (\text{V.30})$$

avec $\varepsilon_i, \sigma_i, \gamma_i > 0$ sont des paramètres à choisir. $p, \hat{\delta}_i > 0$ dépendent des valeurs des bornes des incertitudes sur les paramètres, les perturbations et les interconnexions.

Application sur deux pendules couplés inversés

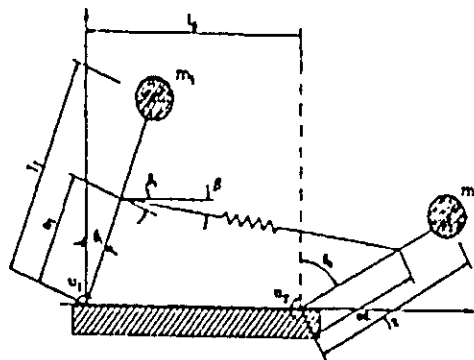


Figure V.2 Deux pendules couplés inversés

Considérons deux pendules inversés identiques, couplés par un ressort, figure V.2. L'entrée pour chacun des pendules est le couple $u_i, i = 1, 2$. Le paramètre a_i , qui est la position du ressort par rapport à l'axe de rotation de chaque pendule, et les paramètres k et A du ressort sont inconnus et varient dans le temps. En outre, la charge m_i est aussi incertaine, avec $m_i = m_i^\circ + \Delta m_i$, m_i° et Δm_i désigne respectivement la valeur nominale de la charge et l'incertitude sur sa valeur. Les équations décrivant le mouvement des deux pendules sont les suivantes:

$$\begin{aligned} m_1 l_1^2 \ddot{\theta}_1 - m_1 g l_1 \sin(\theta_1) &= \Gamma_1 = u_1 - b_1 \dot{\theta}_1 + F a_1 \cos(\theta_1 - \beta) \\ m_2 l_2^2 \ddot{\theta}_2 - m_2 g l_2 \sin(\theta_2) &= \Gamma_2 = u_2 - b_2 \dot{\theta}_2 - F a_2 \cos(\theta_2 - \beta) \end{aligned} \quad (\text{V.31})$$

b_1, b_2 sont les coefficients d'amortissement.

$$F = k \left[1 + A^2 (l_k - l_0)^2 \right] l_k - l_0, \quad |A(l_k - l_0)| < 1$$

$$\beta = \arctan \left[\frac{a_1 \cos(\theta_1) - a_2 \cos(\theta_2)}{l_0 - a_1 \sin(\theta_1) + a_2 \sin(\theta_2)} \right] \quad (\text{V.32})$$

$$l_k = \left[(l_0 - a_1 \sin(\theta_1) + a_2 \sin(\theta_2))^2 + (a_1 \cos(\theta_1) - a_2 \cos(\theta_2))^2 \right]^{1/2}$$

Ce système peut être représenté sous la forme (V.25) avec $q_i = \theta_i$, $M_i = m_i l_i^2$, $C_i = 0$, $g_i = -m_i g l_i \sin(q_i)$, $D_i = b_i \dot{q}_i$, $Z_i = (-1)^i F a_i \cos(q_i - \beta) \cdot p - 3$.

Les valeurs numériques utilisées dans la simulation sont: $m_1^\circ = 0,40$, $m_2^\circ = 0,50$, $\Delta m_1 = 0,04$, $\Delta m_2 = 0,04$, $l_1 = 0,30$, $l_2 = 0,35$, $b_1 = b_2 = 0,009$, $k = 30,00$, $A = 0,10$ et la position du ressort $a_1 = 0,1$ pour $t \in [0,1[$ et $0,3$ si $t \in [1,6]$, $a_2 = 0,35$ pour $t \in [0,3[\cup [5,6]$ et $0,05$ si $t \in [3,5[$.

Les paramètres de la loi de commande sont: $K_1 = 30$, $K_2 = 30$, $A_1 = 20$, $A_2 = 30$, $\varepsilon_1 = \varepsilon_2 = 0,008$, $\sigma_1 = \sigma_2 = 5$, $\gamma_1 = \gamma_2 = 1$ et les valeurs initiales de $\hat{\delta}_1(0) = \hat{\delta}_2(0) = 0,10$.

Remarque: La valeur de l_0 n'a pas été donnée dans l'article, nous prenons alors une valeur, suite à plusieurs simulations, de $l_0 = 0,50$.

Les trajectoires désirées sont $q_1^d(t) = 0,55 \cos(6,28t)$, $q_2^d(t) = 0,35 \cos(9,42t)$ et les valeurs initiales de position, vitesse et accélération des deux pendules sont zéro. La figure V.4 nous donne les graphes des positions angulaires $\{q_i(-), q_i^d(-)\}$, l'erreur de consigne, la commande appliquée à chaque pendule et la variation de la valeur de $\hat{\delta}_i$ avec les changements brutaux des interconnexions dues aux changements a_i de la position du ressort ($i = 1$: pendule gauche, $i = 2$: pendule droit).

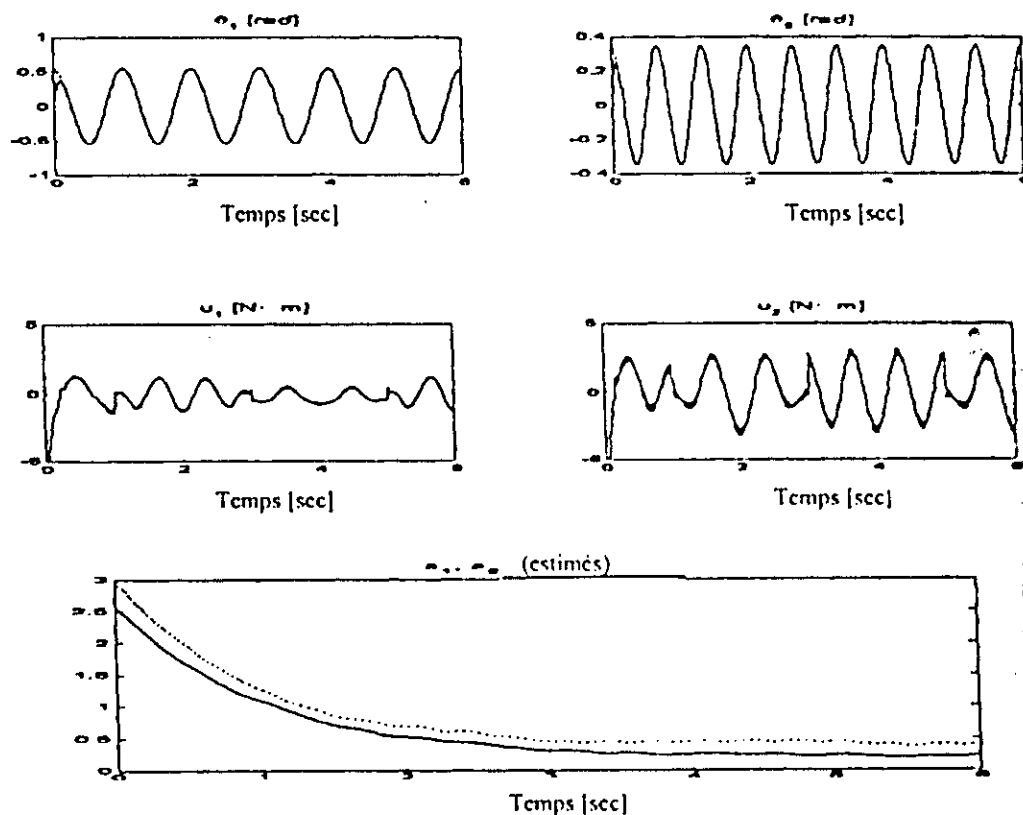


Figure V.3 Application de la loi de commande Y.Tang et coll. (article) sur deux pendules couplés inversés.

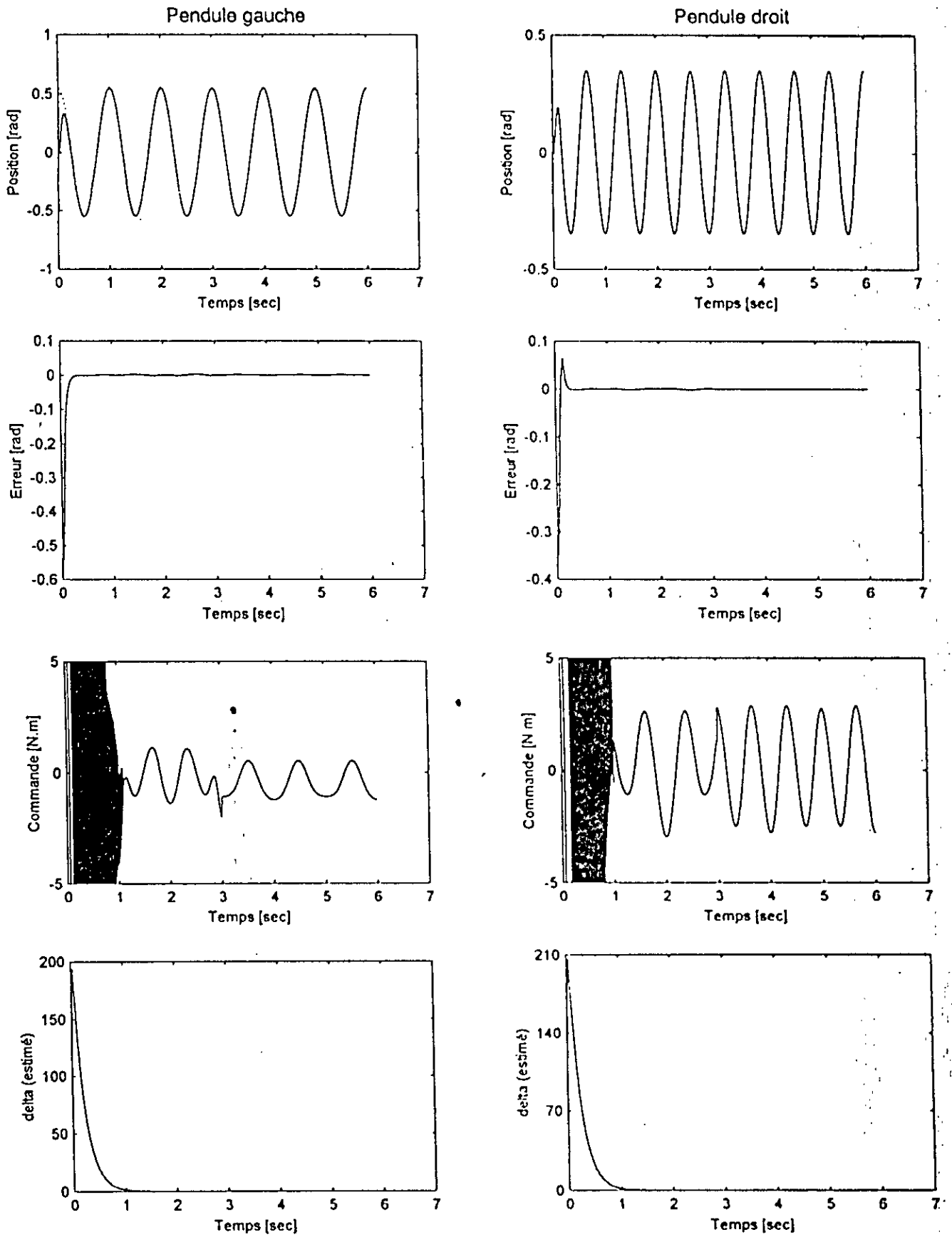


Figure V.4 Application de la loi de commande Y.Tang et coll. (simulation) sur deux pendules couplés inversés, avec $\rho = 3$.

Position: $\theta_{id}(\cdot)$, $\theta_i(-)$.

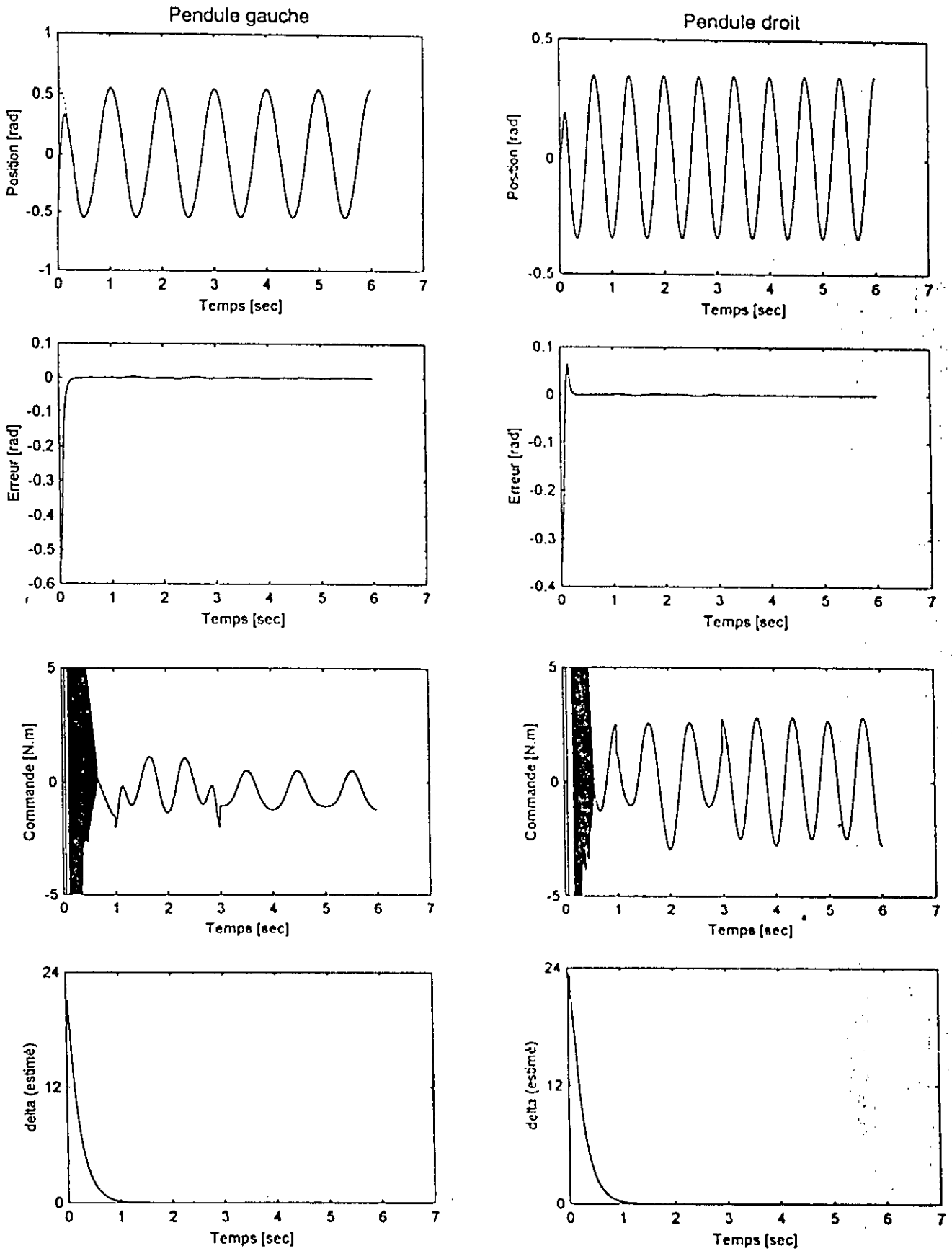


Figure V.5 Application de la loi de commande Y.Tang et coll. (simulation) sur deux pendules couplés inversés, avec $p = 2$.

Position: θ_{id} (··), θ , (-)

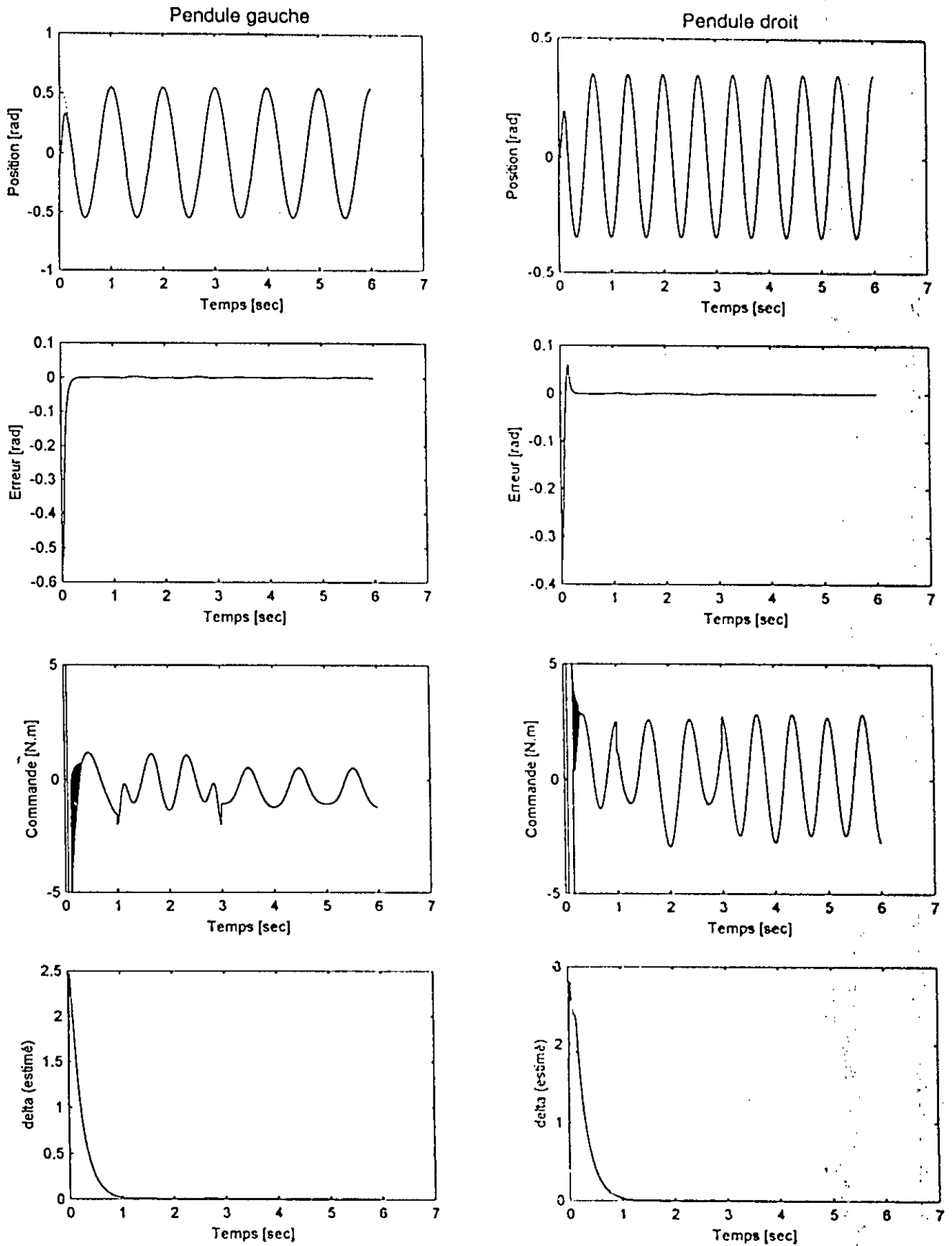


Figure V.6 Application de la loi de commande Y.Tang et coll. (simulation) sur deux pendules couplés inversés, avec $p = 1$.

Position: $\theta_{i,d}(\cdot)$, $\theta_i(\cdot)$

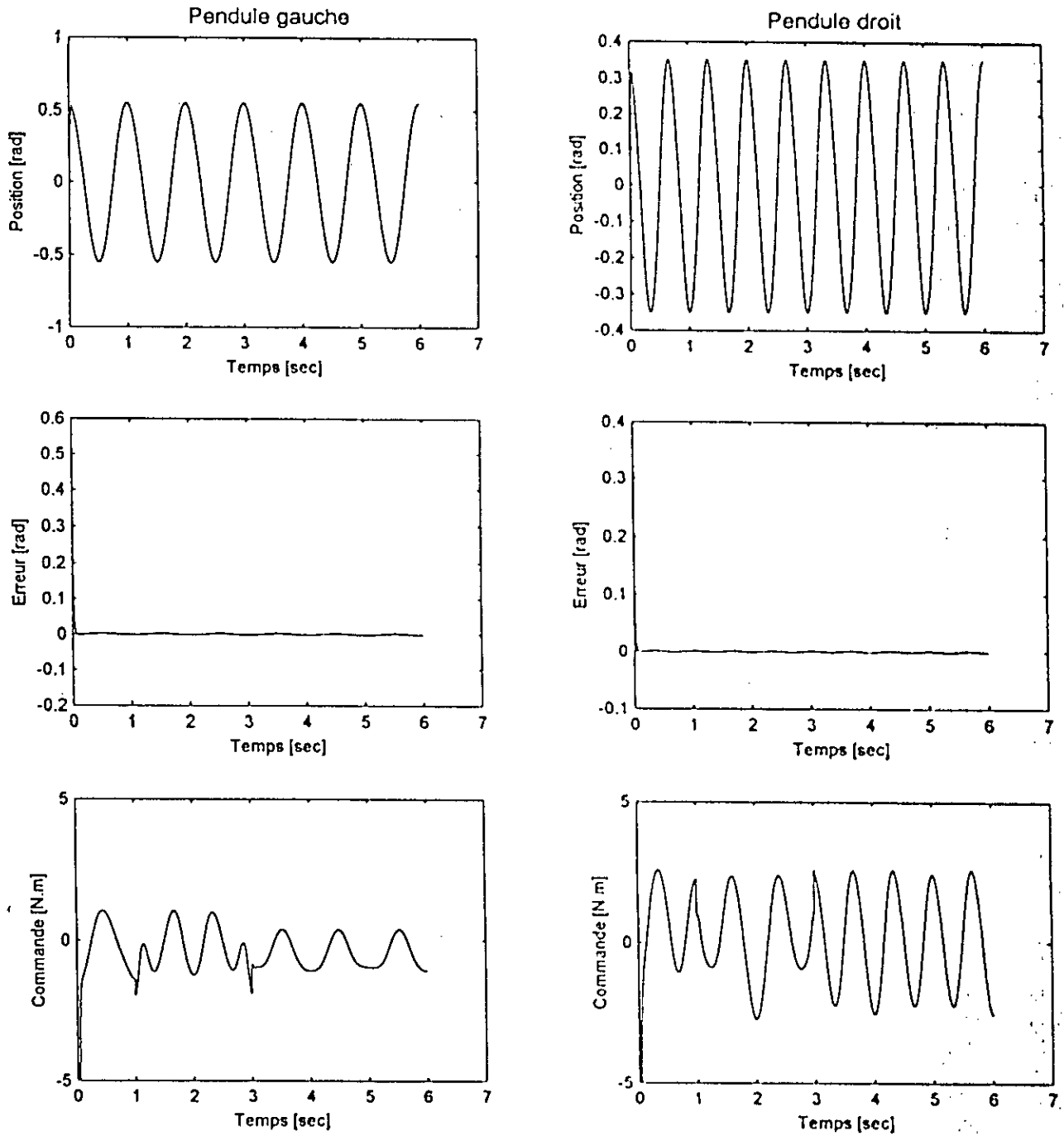


Figure V.7 Simulation de notre loi de commande sur deux pendules couplés inversés.

Position: $\theta_{id}(\cdot)$, $\theta_r(-)$

Discussion des résultats:

La figure V.3 donne les graphes de l'article et la figure V.4 donne les graphes obtenus par la simulation. A partir des graphes de la figure V.4, nous remarquons que les deux courbes de commandes u_1 et u_2 passent par un régime transitoire, où les commandes sont trop excitées. Les résultats de la commande et de $\hat{\delta}_j$ ne s'accordent pas avec ceux donnés dans l'article (figure V.3) $\{\hat{\delta}_1 (-), \hat{\delta}_2 (\cdot\cdot)\}$. Les courbes de $\hat{\delta}_1$ et $\hat{\delta}_2$ (figure V.4) montrent que la borne des valeurs des interconnexions, perturbations et incertitudes est surestimée. Cette conclusion nous amène à diminuer la valeur de p . la figure V.5 et la figure V.6 donnent respectivement les résultats de simulation pour $p = 2$ et $p = 1$. Nous constatons une amélioration considérable pour $p = 1$ où les courbes de $\hat{\delta}_1$ et $\hat{\delta}_2$ s'accordent bien avec celles données dans l'article de même pour u_1 et u_2 .

La figure V.7 donne les graphes de simulation de notre loi de commande sur deux pendules couplés inversés. Les résultats obtenus sont très encourageants pour deux raisons : les commandes sont moins bruitées et la convergence des erreurs est plus rapide.

Remarque: La variation dans la valeur de l_0 n'a pas changé dans les résultats pour $p = 3$.

Application de la méthode Y. Tang et coll. sur le PUMA 560

Nous donnons seulement les résultats de la simulation à vide sur le PUMA 560, figure V.8. Les autres essais (robustesses et décentralisations) seront réalisés ultérieurement

Les paramètres de la loi de commande sont: $K_1 = 100$, $K_2 = 100$, $A_1 = 50$, $A_2 = 50$, $\varepsilon_1 = \varepsilon_2 = 0.01$, $\sigma_1 = \sigma_2 = 10$, $\gamma_1 = \gamma_2 = 5$ et les initiales de $\hat{\delta}_1(0) = \hat{\delta}_2(0) = 0,10$. $p = 3$.

Les graphes des commandes et des erreurs de consigne sont appréciables. A vue d'œil, si nous les comparons avec ceux donnés par notre méthode, nous pouvons dire qu'ils sont peu différents. Une comparaison plus approfondie est donnée à la fin de ce chapitre.

V.3 Etude comparative de chaque loi de commande sur le PUMA 560

Nous réalisons trois types de tests. La comparaison se fait sur le maximum de la valeur absolue de la commande et de l'erreur. Pour distinguer entre les valeurs données par chaque loi de commande, la notation de la commande et de l'erreur de la i^{me} articulation se fait de la manière suivante:

- Notre loi de commande: u_i, e_i .
- Y.Tang: $u_{i(Tang)}, e_{i(Tang)}$.
- Proportionnel Dérivé: $u_{i(PI)}, e_{i(PI)}$.

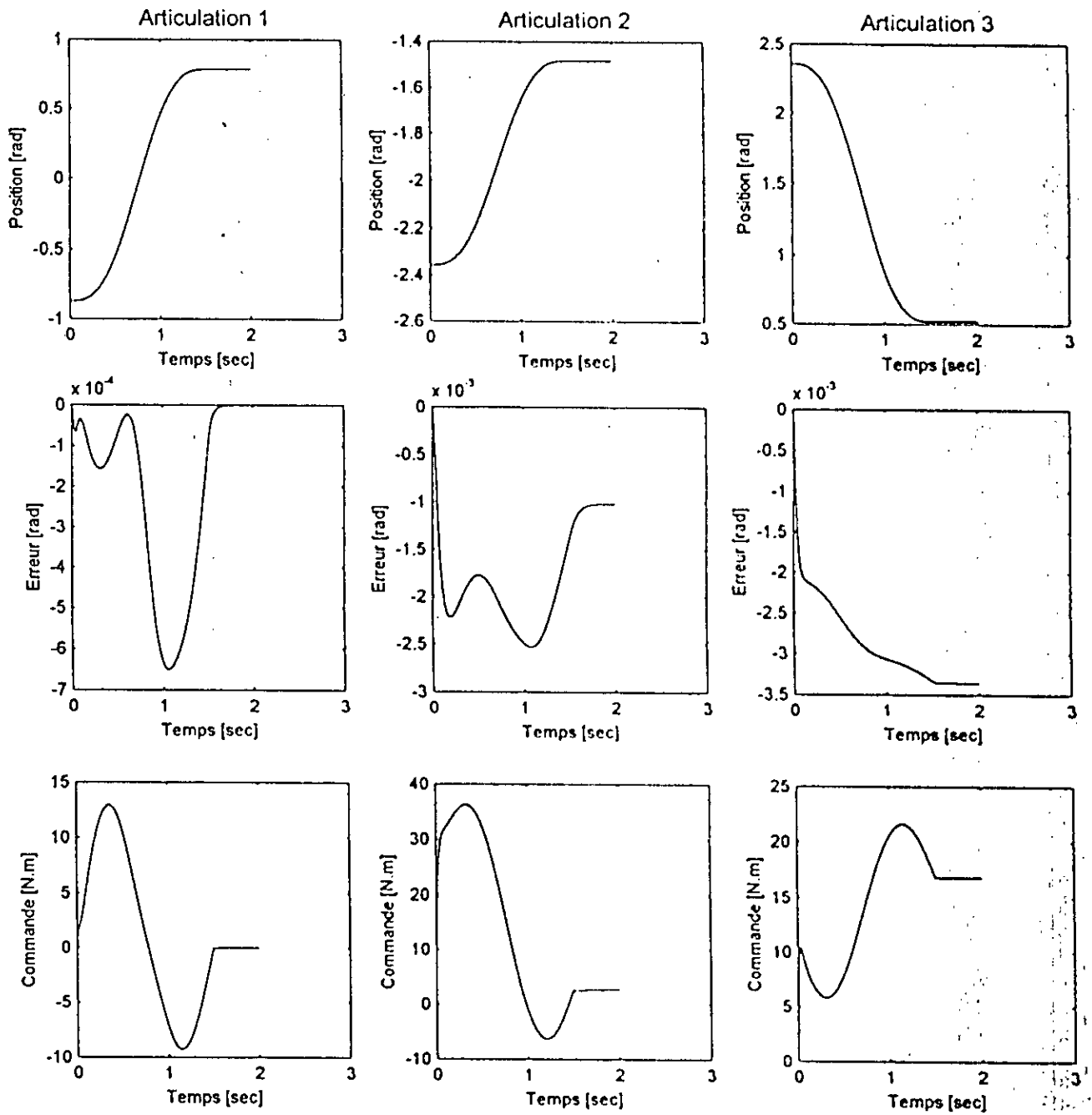


Figure V.8 Essai à vide de la loi de commande Y. Tang et coll.

Position: q_{id} (-), q_t (—).

Test en charge

Ce test se résume en un changement de masse. L'intérêt d'un tel test est de comparer l'effet de la variation de la masse sur la commande et l'erreur, entre les trois types de loi de commande. Les résultats obtenus sont donnés ci dessous:

Articulation 1:

		0 kg	2 kg	4 kg	6 kg	8 kg	10 kg
Commande	$\max u_1 $	12,9145	14,3183	15,7263	17,1384	18,5613	20,0081
	$\max u_1 (Tang) $	12,9512	14,3869	15,8294	17,2789	18,7364	20,2027
	$\max u_1 (PD) $	13,3477	15,1253	16,9906	18,8800	20,7643	22,6369
Erreur [rad]	$\max e_1 $	$2,1788 \times 10^{-4}$	$1,8067 \times 10^{-4}$	$3,0263 \times 10^{-4}$	$4,8933 \times 10^{-4}$	$6,9687 \times 10^{-4}$	$9,1315 \times 10^{-4}$
	$\max e_1 (Tang) $	$6,5049 \times 10^{-4}$	$4,1795 \times 10^{-4}$	$6,9588 \times 10^{-4}$	$6,7866 \times 10^{-4}$	0,0013	0,0016
	$\max e_1 (PD) $	0,0064	0,0072	0,0081	0,0090	0,0099	0,0108

Articulation 2:

		0 kg	2 kg	4 kg	6 kg	8 kg	10 kg
Commande	$\max u_2 $	36,1948	35,1543	34,1250	33,1322	32,2041	31,7897
	$\max u_2 (Tang) $	36,2856	35,2006	34,0833	32,9282	32,7539	32,7386
	$\max u_2 (PD) $	36,5655	37,0133	37,6002	37,8457	37,8372	37,6358
Erreur [rad]	$\max e_2 $	$2,4844 \times 10^{-4}$	$2,9601 \times 10^{-4}$	$3,4344 \times 10^{-4}$	$3,9036 \times 10^{-4}$	$4,3649 \times 10^{-4}$	$4,8358 \times 10^{-4}$
	$\max e_2 (Tang) $	0,0025	0,0029	0,0032	0,0035	0,0039	0,0042
	$\max e_2 (PD) $	0,0059	0,0056	0,0056	0,0057	0,0057	0,0057

Articulation 3:

		0 kg	2 kg	4 kg	6 kg	8 kg	10 kg
Commande	$\max u_3 $	21,5871	22,4516	23,3181	24,1842	25,0505	25,9180
	$\max u_3 (Tang) $	21,6291	22,5118	23,4030	24,3031	25,2127	26,1324
	$\max u_3 (PD) $	21,8237	22,3536	22,9913	23,7193	24,5402	25,4614
Erreur [rad]	$\max e_3 $	0,0029	0,0029	0,0029	0,0029	0,0029	0,0028
	$\max e_3 (Tang) $	0,0034	0,0034	0,0035	0,0037	0,0038	0,0040
	$\max e_3 (PD) $	0,0231	0,0237	0,0243	0,0250	0,0258	0,0267

Test de décentralisation

Nous allons refaire le même test pour notre loi de commande mais en prenant plusieurs valeurs de masse. Donc nous incluons deux résultats, celui de l'effet de l'interruption de la commande u_2 et la variation de son influence avec le changement de masse. Les résultats obtenus sont donnés ci dessous:

Articulation 1:

		0 kg	2 kg	4 kg	6 kg	8 kg	10 kg
Commande [N.m]	max $ u_1 $	12,9145	14,3183	15,7263	17,1384	18,5613	20,0081
	max $ u_1 (Tang) $	12,9512	14,3869	15,8294	17,2789	18,7364	20,2027
	max $ u_1 (PD) $	13,3477	15,1253	16,9906	18,8800	20,7643	22,6369
Erreur [rad]	max $ e_1 $	$3,4424 \times 10^{-4}$	$2,0416 \times 10^{-4}$	$3,0263 \times 10^{-4}$	$4,8804 \times 10^{-4}$	$6,8176 \times 10^{-4}$	$8,7690 \times 10^{-4}$
	max $ e_1 (Tang) $	$7,0645 \times 10^{-4}$	$4,6288 \times 10^{-4}$	$6,9588 \times 10^{-4}$	$9,7866 \times 10^{-4}$	0,0013	0,0016
	max $ e_1 (PD) $	0,0064	0,0072	0,0081	0,0090	0,0099	0,0108

Articulation 3:

		0 kg	2 kg	4 kg	6 kg	8 kg	10 kg
Commande [N.m]	max $ u_3 $	22,2076	23,0157	23,6686	24,2144	24,6839	25,0978
	max $ u_3 (Tang) $	22,2283	23,0177	23,6515	24,1763	24,6225	25,0105
	max $ u_3 (PD) $	22,4583	23,0617	23,6537	24,2184	24,7543	25,2627
Erreur [rad]	max $ e_3 $	0,0029	0,0029	0,0029	0,0029	0,0029	0,0028
	max $ e_3 (Tang) $	0,0033	0,0034	0,0035	0,0036	0,0037	0,0038
	max $ e_3 (PD) $	0,0238	0,0244	0,0250	0,0255	0,0261	0,0266

Test de robustesse

La particularité de ce test est de montrer l'efficacité de la loi de commande vis-à-vis des variations pouvant se produire tout au long du fonctionnement du système. Dans notre cas, nous considérons seulement la chute de la masse à l'instant $t = 0,75$ sec. Ce test est réalisé pour plusieurs valeurs de la charge. Les résultats obtenus sont donnés ci dessous:

Articulation 1:

		0 kg	2 kg	4 kg	6 kg	8 kg	10 kg
Commande [N.m]	max $ u_1 $	12,9145	14,3183	15,7263	17,1384	18,5613	20,0081
	max $ u_1 (Tang) $	12,9512	14,3869	15,8294	17,2789	18,7364	20,2027
	max $ u_1 (PD) $	13,3477	15,1253	16,9906	18,8800	20,7643	22,6369
Erreur [rad]	max $ e_1 $	$2,1788 \times 10^{-4}$	$2,1634 \times 10^{-4}$	$3,0263 \times 10^{-4}$	$4,8802 \times 10^{-4}$	$6,8150 \times 10^{-4}$	$6,7585 \times 10^{-4}$
	max $ e_1 (Tang) $	$6,5049 \times 10^{-4}$	$6,5058 \times 10^{-4}$	$6,9588 \times 10^{-4}$	$6,7866 \times 10^{-4}$	0,0013	0,0016
	max $ e_1 (PD) $	0,0064	0,0072	0,0081	0,0090	0,0099	0,0108

Articulation 2:

		0 kg	2 kg	4 kg	6 kg	8 kg	10 kg
Commande [N.m]	max $ u_2 $	36,1948	35,1543	34,1250	33,1322	32,2041	31,7897
	max $ u_{2(Tang)} $	36,2856	35,2006	34,0833	32,9282	32,7539	32,7386
	max $ u_{2(PD)} $	36,5655	37,0133	37,6002	37,8457	37,8372	37,6358
Erreur [rad]	max $ e_2 $	$2,4844 \times 10^{-4}$	$2,8131 \times 10^{-4}$	$3,2797 \times 10^{-4}$	$3,7503 \times 10^{-4}$	$4,2238 \times 10^{-4}$	$4,6972 \times 10^{-4}$
	max $ e_{2(Tang)} $	0,0025	0,0025	0,0025	0,0025	0,0025	0,0025
	max $ e_{2(PD)} $	0,0059	0,0056	0,0056	0,0057	0,0057	0,0057

Articulation 3:

		0 kg	2 kg	4 kg	6 kg	8 kg	10 kg
Commande [N.m]	max $ u_3 $	21,5871	22,4516	23,3181	24,1842	25,0505	25,9180
	max $ u_{3(Tang)} $	21,6291	22,5118	23,4030	24,3031	25,2127	26,1324
	max $ u_{3(PD)} $	21,8237	22,3536	22,9913	23,7193	24,5402	25,4614
Erreur [rad]	max $ e_3 $	0,0029	0,0029	0,0029	0,0029	0,0029	0,0028
	max $ e_{3(Tang)} $	0,0034	0,0034	0,0035	0,0037	0,0038	0,0040
	max $ e_{3(PD)} $	0,0231	0,0237	0,0243	0,0250	0,0258	0,0267

Interprétation générale

Lors de l'étude comparative, nous constatons que notre proposition donne de meilleurs résultats du point de vue commande et erreur de consigne que ceux obtenus par les deux autres lois de commande. Nous remarquons aussi la non variation de la valeur maximale absolue de la commande de la première et de la deuxième articulation et un léger changement de la troisième commande au niveau du test de décentralisation. Cette constatation révèle que dans un tel système, le choix adéquat de la commande est très important.

La loi de commande donnée par Y. Tang et coll. [14] suscite un intérêt particulier vu qu'elle a été récemment publiée (2000). Nous avons donc jugé nécessaire de faire une étude comparative plus approfondie afin de pouvoir mieux la comparer avec celle que nous proposons.

V.4 Etude comparative approfondie

Nous étudions les deux lois de commande, Y. Tang et coll. [14] et la notre, au niveau des valeurs des commandes et des erreurs de consignes sur tout l'intervalle de simulation en comparant leurs courbes. Cette étude comporte quatre tests: de charge (10 kg) en figure V.9, de chute de masse (10 kg) en figure V.10, de variation paramétrique (50%) en figure V.11 et de décentralisation ($u_2 = 0$) en figure V.12. Chaque test est illustré par deux graphes et ce pour chaque articulation:

Premier graphe: e_i (—) et $e_{i(Tang)}$ (··).

Deuxième graphe: u_i (—) et $u_{i(Tang)}$ (··).

V.5 Conclusion

D'après les figures V.9, V.10, V.11 et V.12, les résultats de cette comparaison ne font que confirmer la performance de la loi de commande que nous proposons.

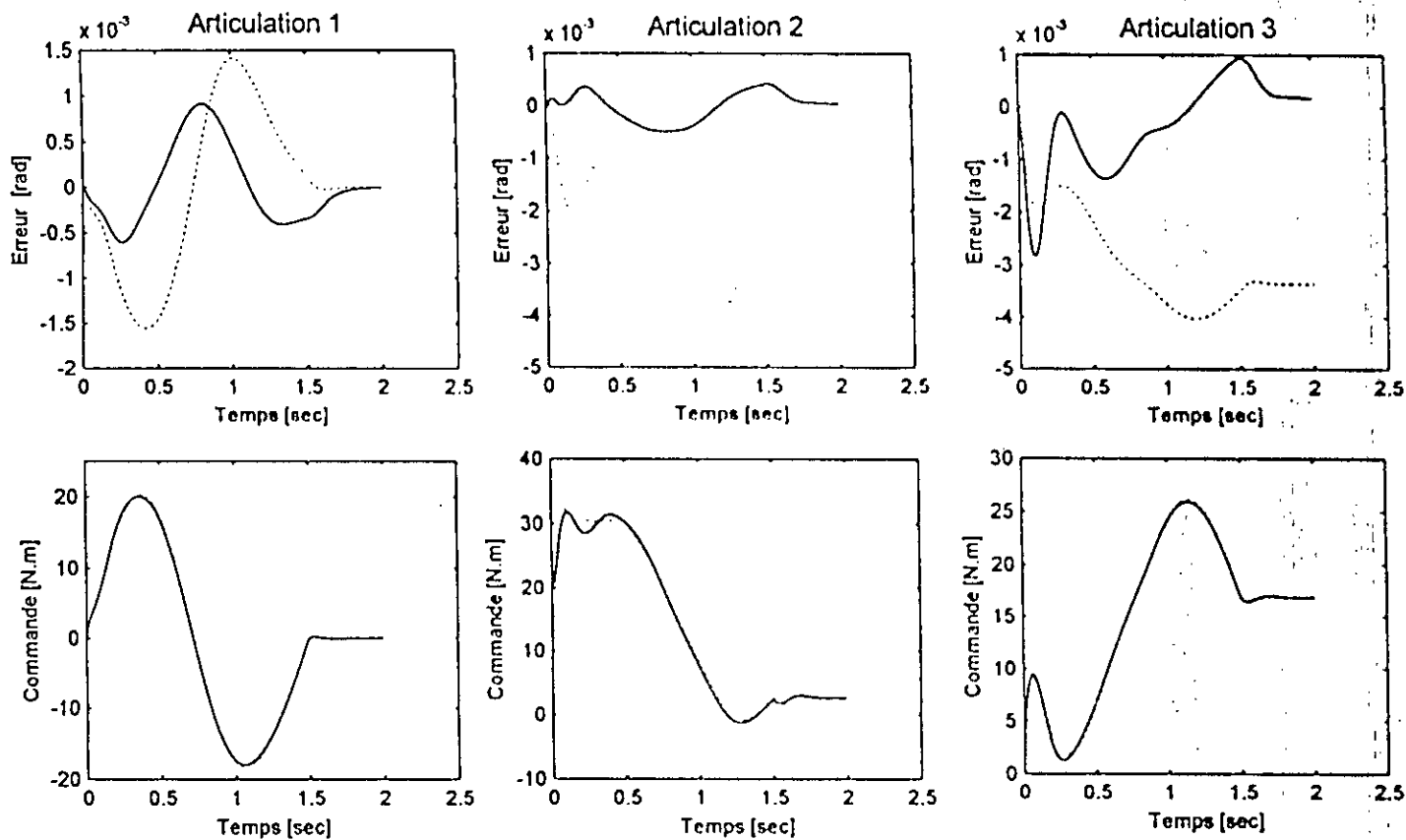


Figure V.9 Comparaison entre Y. Tang et coll. (· ·) et notre proposition (—), pour un essai en charge ($m = 10$ kg).

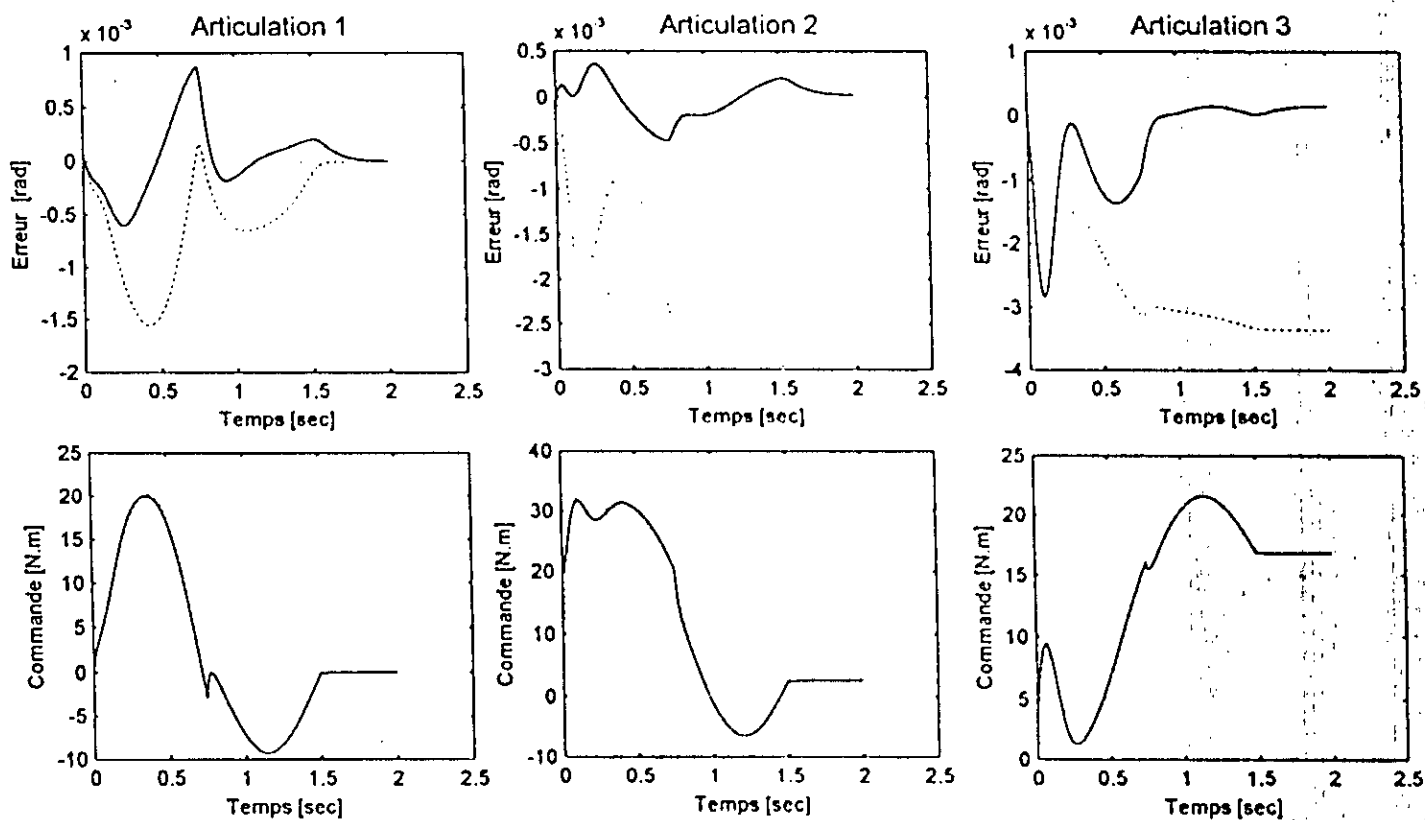


Figure V.10 Comparaison entre Y. Tang et coll. (· ·) et notre proposition (—), pour un essai de robustesse, chute de masse ($m = 10$ kg) à $t = 0,75$ sec.

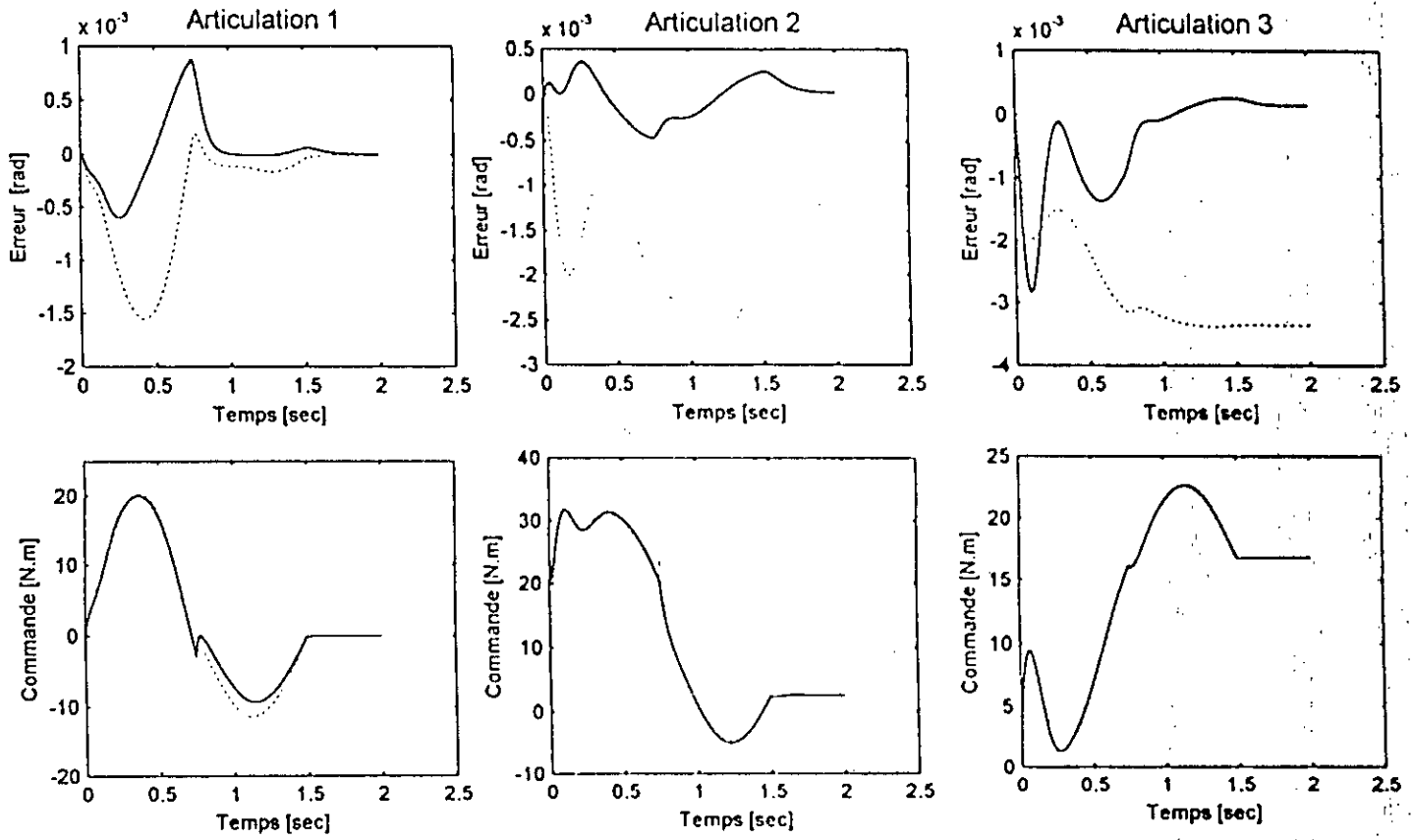


Figure V.11 Comparaison entre Y. Tang et coll. (· ·) et notre proposition (—), pour un essai de robustesse, changement paramétrique de 50% à $t = 0,75$ sec.

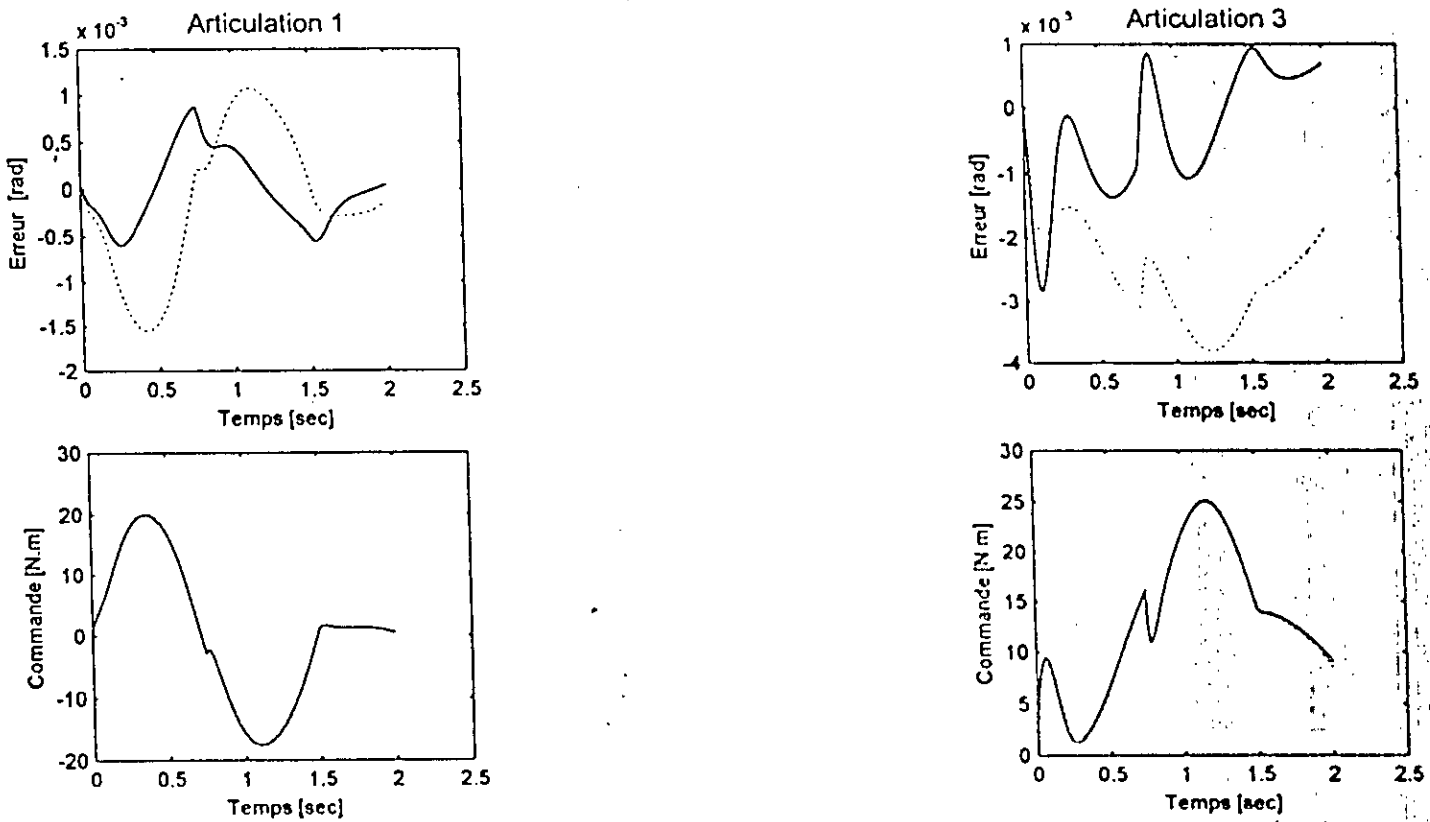


Figure V.12 Comparaison entre Y. Tang et coll. (· ·) et notre proposition (—), pour un essai de décentralisation, à vide où $\nu_2 = 0$ à $t = 0,75$ sec.

CONCLUSION GENERALE

La commande décentralisée des systèmes mécaniques articulés a fait l'objet de plusieurs travaux de recherche. Lors de la synthèse de ce type de commande, l'objectif principal est l'atténuation de l'effet des interconnexions et des perturbations souvent difficile à mesurer, entraînant des difficultés majeures lors de la commande de tels systèmes. D'autre part, des incertitudes sur les modèles sont généralement inévitables. Dans le but de pallier à ces problèmes, nous avons mis au point une commande décentralisée robuste en atténuant l'effet des interconnexions et des perturbations. Nous avons d'abord synthétisé un observateur par réseau de neurones artificiels pour reconstituer les interconnexions et les perturbations. Une commande glissante a été synthétisée pour atténuer leur effet. Enfin, une commande par la méthode du Backstepping a été proposée pour imposer la dynamique du système.

Une importante recherche bibliographique nous a permis de situer notre travail parmi ceux qui associent les qualités et avantages des différentes techniques de synthèse de lois de commande.

L'atténuation de l'effet des interconnexions et des perturbations est le but essentiel des différentes techniques de commande décentralisée. La solution apportée par ces techniques est de regrouper l'effet des interconnexions et des perturbations comme étant une perturbation, puis la majorer par une valeur connue, et synthétiser ainsi la commande en se basant sur cette valeur.

Cette majoration peut entraîner une utilisation excessive de l'énergie malgré que le besoin ne se fait pas sentir. L'utilisation d'un observateur par réseau de neurones artificiels nous a permis de remédier à cet inconvénient, en remplaçant la majoration par la reconstitution apportée par l'observateur.

Dans la littérature, les tests de performance des méthodes de commande se font sur des systèmes mécaniques articulés. Nous avons effectué des tests sur le robot manipulateur PUMA 560, vu sa large utilisation dans le secteur industriel. Les tests utilisés comprennent des essais de robustesse et de décentralisation. Les résultats obtenus sont très intéressants du point de vue de la commande et de l'erreur de poursuite.

La simplicité et l'efficacité de cette méthode lui permettent une large utilisation.

Notre modeste contribution peut donner lieu à d'autres études, comme l'application à des systèmes interconnectés tels que les réseaux électriques, ou encore remplacer l'observateur par réseau de neurones par un observateur flou.

ANNEXES

Titre

ANNEXE A

Outils nécessaires à la modélisation d'un robot

A.1 Éléments géométriques de la modélisation cinématique

Par soucis de clarté nous ne détaillerons pas cette partie qui a été amplement étudiée par [17] et [18].

A.1.1 Matrice de changement de base

Soit deux repères orthonormés directs R_0 et R_1 de vecteurs unitaires respectifs X, Y, Z et x, y, z . On définit la matrice de passage de R_0 à R_1 la matrice A dont les colonnes sont les composantes des vecteurs de R_1 dans R_0 . A s'écrit comme suit

$$A = \begin{bmatrix} x.X & y.X & z.X \\ x.Y & y.Y & z.Y \\ x.Z & y.Z & z.Z \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

La matrice A est appelée matrice des cosinus directeurs ou matrice de rotation.

A.1.2 Coordonnées d'un même point dans deux repères

Soit deux repères R_i et R_j d'origine O_i et O_j . Soit un point P de coordonnées $P_i = (x_i, y_i, z_i)$ dans R_i et de coordonnées $P_j = (x_j, y_j, z_j)$ dans R_j . Soit aussi $L_{ij} = (L_{ij1}, L_{ij2}, L_{ij3})$ les coordonnées de O_j dans R_i . On a la relation vectorielle :

$$O_i P = O_j P + O_i O_j \quad (\text{A.1})$$

Ou d'une autre manière on écrit :

$$P_i^T \{e_i\} = P_j^T \{e_j\} + L_{ij}^T \{e_i\} \quad (\text{A.2})$$

avec e vecteur unitaire .

$$\text{On peut donc écrire :} \quad P_i^T \{e_i\} = P_j^T A \{e_j\} + L_{ij}^T \{e_i\} \quad (\text{A.3})$$

$$\text{On aura finalement :} \quad P_i = A P_j + L_{ij} \quad (\text{A.4})$$

A.1.3 Matrice de transformation homogène

Prenons cette fois ci les coordonnées homogènes de P dans R_i et R_j , on écrit alors :

$$P_i = (x_i, y_i, z_i, 1) \text{ et } P_j = (x_j, y_j, z_j, 1).$$

La relation (A.4) peut être écrite alors de la manière suivante :

$$P_i = T_j^i P_j \quad (\text{A.5})$$

avec T_j^i matrice de transformation homogène définie comme suit :

$$T_i^j = \begin{bmatrix} A & I_{ij} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} x.X & y.X & z.X & I_{ij1} \\ x.Y & y.Y & z.Y & I_{ij2} \\ x.Z & y.Z & z.Z & I_{ij3} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.6})$$

A.1.4 Paramétrage de Denavit-Hartenberg

Ce paramétrage a été introduit par Denavit-Hartenberg dans les années 50 pour l'étude systématique des systèmes mécaniques articulés. Cette méthode concerne les chaînes cinématiques ouvertes où l'articulation possède uniquement un degré de liberté.

Pour faciliter le calcul des matrices de transformation homogène, qui permettent d'exprimer rapidement les informations de l'élément terminal vers la base, le choix des repères doit être judicieux.

Les étapes à suivre afin d'appliquer cette méthode sont :

- 1- Numérotation des segments (maillons) constitutifs du bras manipulateur de la base vers l'organe terminal. La base étant associée au référentiel zéro et l'effecteur associé au référentiel d'ordre n (voir figure A.1).
- 2- Définir les axes principaux de chaque segment:
 - Si z_i et z_{i-1} ne se coupent pas on choisit x_i parallèle à l'axe perpendiculaire à z_i et z_{i-1}
 - Si z_i et z_{i-1} sont colinéaires, on choisit x_i dans le plan perpendiculaire à z_{i-1} .
- 3- Fixer les paramètres : d_i , θ_i , a_i , et α_i pour chaque articulation (figure A.1):
 - d_i et α_i sont deux paramètres de localisation de l'axe z_i dans le référentiel R_{i-1} , on les définit comme suit:
 - d_i : coordonnée de l'origine O_i sur l'axe z_{i-1} , pour une glissière d_i est une variable, et elle est constante pour une charnière
 - α_i : angle entre z_i et z_{i-1} obtenu en vissant z_{i-1} vers z_i autour de x_i
 - a_i et θ_i sont deux paramètres de mouvement (rotation et/ou translation) de R_i / R_{i-1} , on les définit comme suit:
 - θ_i : angle que l'on obtient par vissage de x_{i-1} vers x_i autour de l'axe z_{i-1} . θ_i est variable pour une charnière et constante pour une glissière
 - 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}
- 4- On construit alors la matrice de transformation homogène de Denavit-Hartenberg qui lie la rotation et la translation:

$$T_{i-1}^i = \begin{bmatrix} R_{i-1}^i & I_{i-1}^i \\ 000 & 1 \end{bmatrix} \quad (\text{A.7})$$

avec R_i^{i-1} matrice de rotation :

$$R'_{i-1} = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & \sin \alpha_i \cos \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix} \quad (\text{A.8})$$

et L'_{i-1} vecteur de translation:

$$L'_{i-1} = \begin{bmatrix} a_i \cos \theta_i \\ a_i \sin \theta_i \\ d_i \end{bmatrix} \quad (\text{A.9})$$

on obtient donc:

$$T'_{i-1} = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & \sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.10})$$

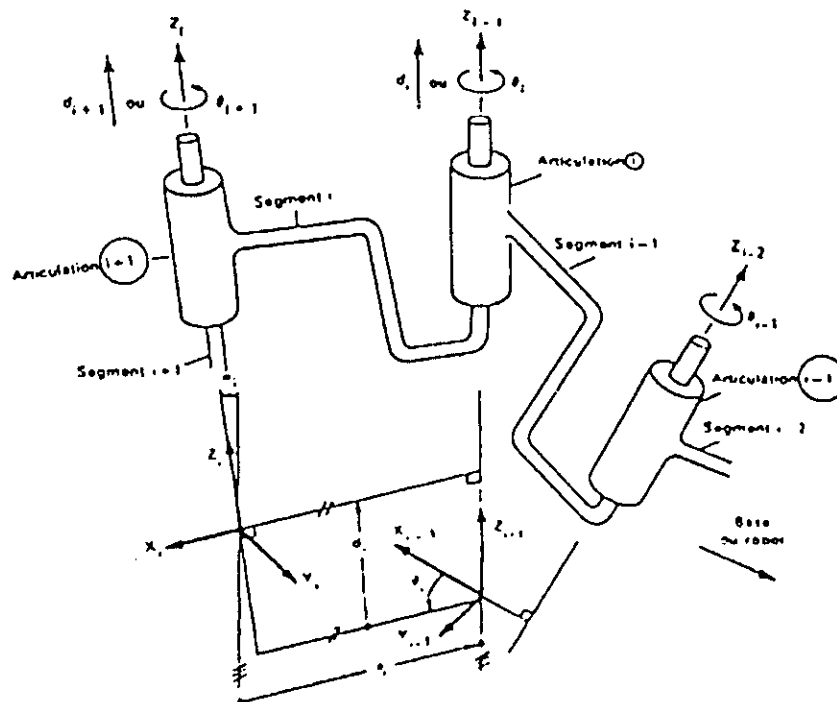


Figure A.1 Organisation des référentiels associés à un système articulé d'après Denavit-Hartenberg

A.1.5 Méthode directe de calcul des coordonnées [54]

Cette méthode, appelée aussi *Général-Electrique*, ne fait appel qu'à un seul référentiel pour décrire la cinématique d'un robot articulé. Généralement ce référentiel est celui de la base R_0 . Le mouvement le long du bras manipulateur est décrit par des rotations et translations par rapport à R_0 . Ainsi la matrice de transformation homogène est obtenue à partir d'un produit de matrices de translations et de rotations écrites à partir de la matrice (A.6). Cette méthode s'appuie plus sur des aspects géométriques, et elle devient compliquée pour un degré de liberté élevé.

A.2 Formalisme d'Euler-Lagrange

Cette méthode consiste à exprimer l'énergie cinétique ainsi que l'énergie potentielle de l'ensemble des solides, ce qui nous permettra d'établir le modèle dynamique à partir de l'équation d'Euler-Lagrange qui est donnée par [55] :

$$\frac{d}{dt} \left[\frac{\partial L}{\partial \dot{q}_i} \right] - \frac{\partial L}{\partial q_i} + \frac{\partial E_D}{\partial \dot{q}_i} = T_i \quad i = 1, \dots, n \quad (\text{A.11})$$

avec :

$$L = E_c - E_p \quad (\text{A.12})$$

où :

- E_p : énergie potentielle totale de toutes les liaisons
- E_c : énergie cinétique totale de toutes les liaisons
- E_D : énergie de dissipation en cas de présence de frottement visqueux
- T_i : force généralisée de la i^{me} articulation
- n : nombre de degrés de liberté
- q_i : coordonnée généralisée d'ordre i
- \dot{q}_i : dérivée de la coordonnée généralisée d'ordre i
- L : le Lagrangien

a- Energie cinétique

Soit un point i , les coordonnées de ce point dans le référentiel lié à la base R_0 sont données par le vecteur r_0^i , la vitesse du point i dans ce même référentiel est donnée par l'expression suivante:

$$V_0^i = \frac{dr_0^i}{dt} \quad (\text{A.13})$$

Soit r_i^i le vecteur des coordonnées homogènes du point i dans le référentiel R_i , et T_0^i la matrice de transformation homogène, on peut donc écrire :

$$r_0^i = T_0^i r_i^i \quad (\text{A.14})$$

donc la relation (A.13) devient:

$$V_0^i = \frac{d(T_0^i r_i^i)}{dt} \quad (\text{A.15})$$

comme la liaison n'est pas flexible on a donc:

$$dr_i^i / dt = 0 \quad (\text{A.16})$$

et la relation (A.15) devient:

$$V_0^i = \sum_{j=1}^i \frac{dT_0^i}{dq_j} \frac{dq_j}{dt} r_i^i \quad (\text{A.17})$$

$$V_0^i = \sum_{j=1}^i [U_{ij} \dot{q}_j] r_i^i \quad (\text{A.18})$$

avec:

$$U_{ij} = \begin{cases} T_0^{j-1} Q_j T_{j-1}^i & \text{pour } j \leq 0 \\ 0 & \text{pour } j \geq 0 \end{cases} \quad (\text{A.19})$$

pour une liaison rotoïde on a:

$$Q_j = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.20})$$

pour une liaison prismatique:

$$Q_j = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{A.21})$$

L'énergie cinétique d'un élément i de la liaison i est donnée par

$$dE_{ci} = \frac{1}{2} \text{trace} (V_i V_i^T) dm \quad (\text{A.22})$$

les équations (A.17) et (A.22) donnent:

$$dE_{ci} = \frac{1}{2} \text{trace} \left(\sum_{j=1}^i \sum_{k=1}^i U_{ij} (r_i^i r_i^T dm) U_{ik}^T \dot{q}_j \dot{q}_k \right) \quad (\text{A.23})$$

$$\dot{E}_{ca} = \frac{1}{2} \text{trace} \left(\sum_{j=1}^l \sum_{k=1}^l U_{ij} J_i U_{ik}^T \dot{q}_j \dot{q}_k \right) \quad (\text{A.24})$$

avec :

$$J_i = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \quad (\text{A.25})$$

L'énergie cinétique des actionneurs est donnée par:

$$\dot{E}_{ca} = \frac{1}{2} \sum_{i=1}^n I_i \dot{q}_i^2 \quad (\text{A.26})$$

où I_i correspond à un moment cinétique dans le cas d'une rotation et une masse dans le cas d'une translation.

Enfin, nous obtenons:

$$\dot{E}_c = \sum_{i=1}^n \dot{E}_{ci} + \dot{E}_{ca} \quad (\text{A.27})$$

b- Énergie potentielle [56]

L'énergie potentielle est donnée par:

$$E_p = \sum_{i=1}^n -m_i g^T r'_i = - \sum_{i=1}^n m_i g^T T'_0 r'_i \quad (\text{A.28})$$

avec $g^T = [g_x \ g_y \ g_z \ 1]$ exprimée dans le repère R_0 .

Si l'orientation de l'axe z_0 est vers le haut, nous aurons alors: $g = [0 \ 0 \ -|g| \ 1]$ (A.29)

c- Énergie de dissipation :

Elle est donnée par l'expression suivante : $E_D = \frac{1}{2} \sum_{i=1}^n f_{vi} \dot{q}_i^2$ (A.30)

où f_{vi} représente le coefficient de frottement visqueux ($i=1, \dots, n$).

Finalement nous obtenons l'expression du Lagrangien :

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^l \sum_{k=1}^l \text{trace} \left(U_{ij} J_i U_{ik}^T \right) \dot{q}_j \dot{q}_k + \sum_i^n m_i g^T T'_0 r'_i \quad (\text{A.31})$$

En appliquant la formule d'Euler-Lagrange à l'équation (A.31), on obtient la force ou couple généralisé suivant:

$$T_i = \sum_{j=1}^n \sum_{k=1}^j \text{trace}(U_{jk} J_j U_{ji}^T) \ddot{q}_k + \sum_{j=1}^n \sum_{k=1}^j \sum_{l=1}^j \text{trace}(U_{jkl} J_j U_{ji}^T) \dot{q}_k \dot{q}_l - \sum_{j=1}^n m_j g^T U_{ji} r_j^j + f_{Vi} \dot{q}_i \quad (\text{A.32})$$

$$U_{ijk} = \begin{cases} T_0^{k-1} Q_k T_{k-1}^{j-1} Q_j T_{j-1}^i & k \leq j \leq i \\ T_0^{j-1} Q_j T_{j-1}^{k-1} Q_k T_{k-1}^i & j \leq k \leq i \\ 0 & j < i < k \end{cases} \quad (\text{A.33})$$

nous pouvons formuler autrement l'équation (A.32) :

$$T_i = \sum_{j=1}^n M_{ij}(q) \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n N_{ijk}(q) \dot{q}_j \dot{q}_k + G_i(q) + H_i(\dot{q}_i) \quad i = 1, \dots, n \quad (\text{A.34})$$

avec :

$$M_{ij}(q) = \sum_{k=\max(i,j)}^n \text{trace}(U_{jk} J_k U_{ki}^T) \quad i = 1, \dots, n \text{ et } j = 1, \dots, n \quad (\text{A.35})$$

$$G_i(q) = - \sum_{j=1}^n m_j g^T U_{ji} r_j^j \quad (\text{A.36})$$

$$N_{ijk}(q) = \sum_{l=\max(i,j,k)}^n \text{trace}(U_{jkl} J_l U_{li}^T) \quad i = 1, \dots, n, j = 1, \dots, n \text{ et } k = 1, \dots, n \quad (\text{A.37})$$

$$H_i(\dot{q}_i) = f_{Vi} \dot{q}_i \quad (\text{A.38})$$

ANNEXE B

Diagramme de rétropropagation

Diagramme de rétropropagation (Diagrammatic Representation of Back-Propagation)

Dans cette annexe nous donnons une représentation du diagramme rétropropagation qui est équivalent à la représentation Algorithmique d'un point de vue information alors que la précision sur le calcul est différente car les deux méthodes considèrent différemment l'information de la forme architecturale du réseau. En particulier la représentation du diagramme donne une meilleure compréhension visuelle du processus entier de rétropropagation, tout en calculant les modifications de façon plus efficace et en apportant des nouvelles modifications à la structure existante par d'autres fonctions d'extension.

Le réseau de neurones à trois couches, représenté à la figure B.1, a comme vecteur d'entrée $u^T = [u_1, u_2, \dots, u_n]$ alors que le vecteur de sortie est $y^T = [y_1, y_2, \dots, y_m]$. $v^T = [v_1, v_2, \dots, v_p]$ et $z^T = [z_1, z_2, \dots, z_q]$ sont respectivement les sorties de la première et de la deuxième couche cachées. $\{w_{ij}^1\}_{p \times n}$, $\{w_{ki}^2\}_{q \times p}$ et $\{w_{lk}^3\}_{m \times q}$ sont des matrices de poids associées aux trois couches indiquées à la figure B.1. Les vecteurs $\bar{v} \in \mathfrak{R}^p$, $\bar{z} \in \mathfrak{R}^q$ et $\bar{y} \in \mathfrak{R}^m$ sont aussi représentés à la figure B.1 avec $\gamma(\bar{v}_i) = v_i$, $\gamma(\bar{z}_k) = z_k$ et $\gamma(\bar{y}_l) = y_l$ où \bar{v}_i , \bar{z}_k et \bar{y}_l sont respectivement les éléments de \bar{v} , \bar{z} et \bar{y} . Si $y_d^T = [y_{d1}, y_{d2}, \dots, y_{dm}]$ est le vecteur de sortie désiré, alors l'erreur du vecteur de sortie pour un vecteur d'entrée u est définie comme : $e = y - y_d$, le vecteur coût J est donc défini comme : $J = \sum_s \|e\|^2$, où cette somme est calculée pour tous les vecteurs d'une série s

donnée {vecteurs de sortie associés aux vecteurs d'entrée}. Si les vecteurs d'entrée sont considérés présents à tout instant, le paramètre J peut être interprété comme la somme carré de l'erreur sur tout l'intervalle de temps. C'est l'interprétation appropriée aux systèmes dynamiques. Dans ce dernier cas, les entrées et les sorties sont séquentielles. Le facteur J a alors la forme :

$$\frac{1}{T} \sum_{i=k-T+1}^k e^2(i), \text{ où } T \text{ est un nombre naturel.}$$

L'ajustement des paramètres doit se faire en déterminant le gradient de J dans l'espace des paramètres. La procédure habituelle suivie est de les ajuster à tout instant en se basant sur l'erreur à cet instant et pour une petite valeur du gain d'adaptation η . Si θ_j représente un des paramètres,

$$\frac{\partial e}{\partial \theta_j} \text{ doit être déterminé pour calculer le gradient comme : } e^T \left(\frac{\partial e}{\partial \theta_j} \right).$$

La méthode rétropropagation est celle qui est recommandée pour obtenir ce gradient.

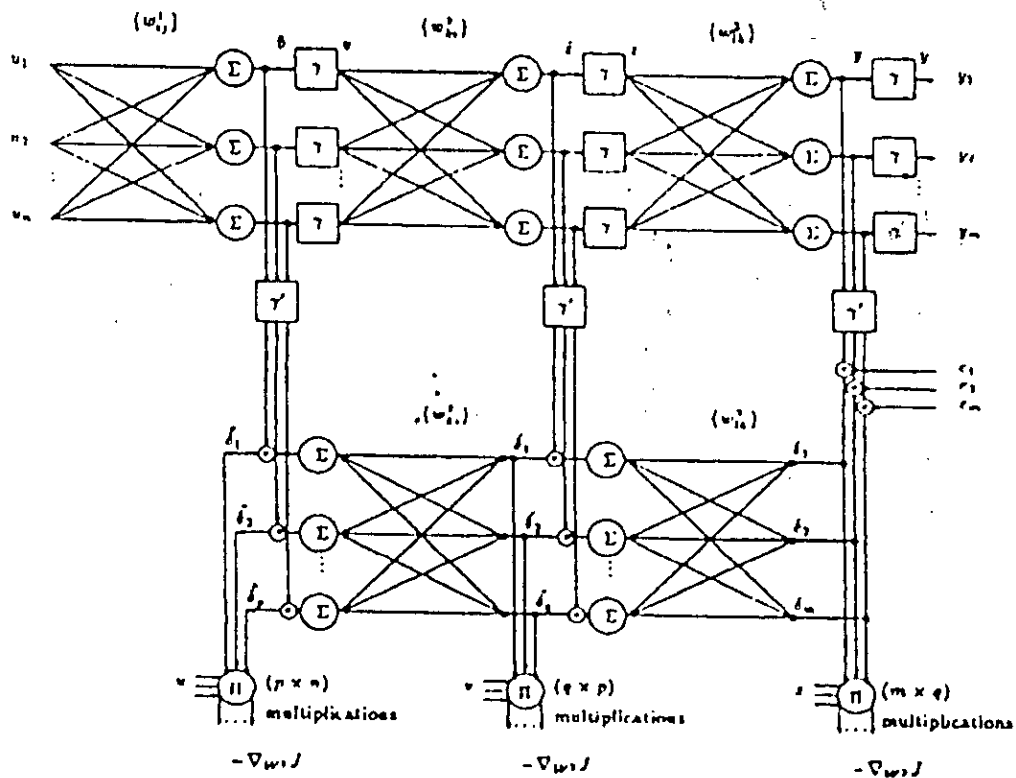


Figure B.1 Architecture du diagramme de rétropropagation

La figure B.1 représente le diagramme de rétropropagation du réseau de neurones à trois couches. La méthode analytique de calcul de dérivé du gradient est bien connue dans la littérature. Nous ne la décrivons pas ici. La figure B.1 montre, d'une façon simple, comment sont réalisées les différentes étapes du gradient. Pour notre exemple, nous voyons que les signaux u , v et z et $\gamma'(\bar{v})$, $\gamma'(\bar{z})$ et $\gamma'(\bar{y})$, ainsi que le vecteur erreur, sont utilisés dans le calcul du gradient (où $\gamma'(x)$ est la dérivée de $\gamma(x)$ par rapport à x). q_m , p_q et n_q sont des multiplications nécessaires au calcul des dérivées partielles par rapport respectivement aux éléments de W^3 , W^2 et W^1 . La structure des matrices poids dans le réseau de neurones utilisée pour calculer les dérivées est identique à celle du réseau de neurones original tant que le mouvement est dans le sens opposé, justifiant ainsi l'usage du terme rétropropagation (Back-Propagation).

ANNEXE C

Observateur par Backstepping

Exemple [35]: Soit le système suivant :

$$\begin{aligned}\dot{\xi} &= x \\ \dot{x} &= -x + \sin \xi + u\end{aligned}\quad (\text{C.1})$$

notre observateur a la forme suivante :

$$\begin{aligned}\dot{\hat{\xi}} &= \hat{x} + \Lambda_1 \\ \dot{\hat{x}} &= -\hat{x} + \sin \hat{\xi} + u + \Lambda_2\end{aligned}\quad (\text{C.2})$$

la dynamique de l'erreur :

$$\begin{aligned}\dot{e}_1 &= e_2 - \Lambda_1 \\ \dot{e}_2 &= -e_2 + \sin \xi - \sin \hat{\xi} - \Lambda_2\end{aligned}\quad (\text{C.3})$$

on suppose que :

$$v = \sin \xi - \sin \hat{\xi}\quad (\text{C.4})$$

avec :

$$\begin{aligned}e_1 &= \xi - \hat{\xi} \\ e_2 &= x - \hat{x}\end{aligned}\quad (\text{C.5})$$

on a donc :

$$\begin{aligned}\dot{e}_1 &= e_2 - \Lambda_1 \\ \dot{e}_2 &= -e_2 + v - \Lambda_2 \quad \text{avec } v < 2\end{aligned}\quad (\text{C.6})$$

Etape 1

$$V(e_1) = \frac{1}{2} e_1^2\quad (\text{C.7})$$

la dérivée de cette fonction est :

$$\begin{aligned}\dot{V}(e_1) &= \dot{e}_1 e_1 \\ &= e_1 (e_2 - \Lambda_1) \\ &< a |e_1| - e_1 \Lambda_1\end{aligned}\quad (\text{C.8})$$

avec :

$$|e_1| = e_1 \operatorname{sgn}(e_1) \quad \text{et} \quad |e_2| < a\quad (\text{C.9})$$

donc :

$$\dot{V}(e_1) \leq e_1 (a \operatorname{sgn} e_1 - \Lambda_1) \quad (\text{C.10})$$

si on prend :

$$\Lambda_1 = k_1 e_1 + a \operatorname{sgn} e_1 \quad \text{avec : } k_1 > 0 \quad (\text{C.11})$$

enfin :

$$\dot{V}(e_1) \leq -k_1 e_1^2 \quad (\text{C.12})$$

Étape 2

$$V(e_1) = \frac{1}{2} e_1^2 + \frac{1}{2} e_2^2 \quad (\text{C.13})$$

la dérivée de cette fonction:

$$\begin{aligned} \dot{V}(e_1, e_2) &= \dot{e}_1 e_1 + \dot{e}_2 e_2 \\ &\leq -k_1 e_1^2 + e_2 (-e_2 + v - \Lambda_2) \\ &\leq -k_1 e_1^2 - e_2^2 + 2|e_2| + e_2 \Lambda_2 \end{aligned} \quad (\text{C.14})$$

donc :

$$\dot{V}(e_1, e_2) \leq -k_1 e_1^2 - e_2^2 + 2a + a \Lambda_2 \quad \text{et} \quad \Lambda_2 > 0 \quad (\text{C.15})$$

si on prend :

$$\Lambda_2 = (k_2 e_1^2 + 2a) / a \quad \text{avec : } k_2 > 0 \quad (\text{C.16})$$

enfin :

$$\dot{V}(e_1, e_2) \leq -(k_1 - k_2) e_1^2 - e_2^2 \quad \text{et} \quad k_1 > k_2 \quad (\text{C.17})$$

on peut agir sur la dynamique de l'observateur en changeant les gains k_1 et k_2 .

ANNEXE D
Méthodes de synthèse d'une commande
par mode glissant et fonctions d'adoucissement

D-1 Calcul de la commande [37][38][41][42]

A) Méthodes basées sur la transformation de $u(t)$ ou de $\sigma(t)$

A-1) Commande par diagonalisation

Soit le système non linéaire décrit par l'équation d'état (III.2). La commande peut se construire par transformation de la commande originale, ou bien par transformation de la surface de glissement.

Méthode 1

On fait subir à $u(x)$ une transformation non singulière pour obtenir un nouveau vecteur de commande. Cette transformation est donnée par :

$$u^*(t) = Q^{-1}(x,t) \left(\frac{\partial \sigma}{\partial x} \right)^T g(x,t) u(t) \quad (D.1)$$

avec : $Q(x,t) = \text{diag} [q_1(x,t), \dots, q_n(x,t)]$ matrice diagonale arbitraire ;

$$\inf |q_i(x,t)| > 0 \quad \forall x, \forall t > 0 \quad (D.2)$$

L'équation (D.1) nous donne :

$$u(t) = \left[\left(\frac{\partial \sigma}{\partial x} \right)^T g(x,t) \right]^{-1} Q(x,t) u^*(t) \quad (D.3)$$

On remplace $u(t)$ dans l'équation d'état (III.2), nous obtenons :

$$\dot{x}(t) = f(x,t) + g(x,t) \left[\left(\frac{\partial \sigma}{\partial x} \right)^T g(x,t) \right]^{-1} Q(x,t) u^*(t) \quad (D.4)$$

Substituant l'équation (D.4) dans l'équation (III.20) :

$$\dot{\sigma}(x) = \frac{\partial \sigma}{\partial x} f(x,t) + Q(x,t) u^*(t) \quad (D.5)$$

Le choix de $u^*(t)$ doit satisfaire la condition $\sigma^T(x) \dot{\sigma}(x) < 0$, il suffit de poser :

$$\begin{cases} q_i(x,t)u_i^+ < -\sum_{j=1}^n S_{ij}f_j(x,t) & \text{si } \sigma_i(x) > 0 \\ q_i(x,t)u_i^- < -\sum_{j=1}^n S_{ij}f_j(x,t) & \text{si } \sigma_i(x) < 0 \end{cases} \quad (\text{D.6})$$

la commande $u(t)$ est calculée à partir de la transformation inverse de $u^*(t)$.

Méthode 2

Nous faisons subir une transformation non singulière $\Omega(x, t)$ à la surface de glissement :

$$\Omega(x, t) = Q(x, t) \left[\left(\frac{\partial \sigma}{\partial x} \right)^T g(x, t) \right]^{-1} \quad (\text{D.7})$$

$$\dot{\sigma}^*(x, t) = \Omega(x, t) \dot{\sigma}(x, t) \quad (\text{D.8})$$

$$\dot{\sigma}^*(x, t) = \Omega(x, t) \dot{\sigma}(x, t) + \dot{\Omega}(x, t) \sigma(x, t) \quad (\text{D.9})$$

$$\dot{\sigma}^*(x, t) = Q(x, t) \left[\left(\frac{\partial \sigma}{\partial x} \right)^T g(x, t) \right]^{-1} \left(\frac{\partial \sigma}{\partial x} \right) \dot{x} + \dot{\Omega}(x, t) \sigma(x, t) \quad (\text{D.10})$$

$$\dot{\sigma}^*(x, t) = Q(x, t) \left[\left(\frac{\partial \sigma}{\partial x} \right)^T g(x, t) \right]^{-1} \left(\frac{\partial \sigma}{\partial x} \right) [f(x, t) + g(x, t)u(t)] + \dot{\Omega}(x, t) \sigma(x, t) \quad (\text{D.11})$$

$$\dot{\sigma}^*(x, t) = Q(x, t) \left[\left(\frac{\partial \sigma}{\partial x} \right)^T g(x, t) \right]^{-1} \left(\frac{\partial \sigma}{\partial x} \right) f(x, t) + Q(x, t)u(t) + \dot{\Omega}(x, t) \sigma(x, t) \quad (\text{D.12})$$

$$\dot{\sigma}^*(x, t) = \sigma^0(x, t) + Q(x, t)u(t) + \sigma_\Omega(x, t) \quad (\text{D.13})$$

La condition d'existence du mode de glissement est vérifiée si :

$$\begin{cases} q_i(x,t)u_i^+ < -\sigma_i^0(x,t) - \sigma_{\Omega i}(x,t) & \text{si } \sigma_i^*(x) > 0 \\ q_i(x,t)u_i^- > -\sigma_i^0(x,t) - \sigma_{\Omega i}(x,t) & \text{si } \sigma_i^*(x) < 0 \end{cases} \quad (\text{D.14})$$

A-2) Commande par la forme quadratique [39]

La transformation non singulière simplifie énormément le choix de l'équation de Lyapunov afin d'obtenir une forme quadratique définie positive, la commande est choisie de manière à ce que la fonction de Lyapunov soit définie négative :

Méthode 1

Elle est basée sur la transformation non singulière de la commande $u(t)$. Soit le système (III.2) avec le vecteur de commande discontinue dont le i^{me} élément est :

$$u_i(t) = \begin{cases} u_i^+(t) & \text{si } \sigma_i(x) > 0 \\ u_i^-(t) & \text{si } \sigma_i(x) < 0 \end{cases} \quad (\text{D.15})$$

Soit la loi de commande définie par l'équation (III.9), de l'équation (D.15) nous pouvons écrire :

$$u = u^0 + U \operatorname{sgn}(\sigma) \quad (\text{D.16})$$

avec :

$$u^0 = 1/2 (u^+ + u^-) \quad (\text{D.17})$$

$$U = \operatorname{diag} [u_1^0 \dots u_m^0] \quad (\text{D.18})$$

$$\operatorname{sgn}(\sigma) = [\operatorname{sgn}(\sigma_1) \dots \operatorname{sgn}(\sigma_m)]^T \quad (\text{D.19})$$

En supposant que x_i est un signal constant, nous obtenons :

$$\dot{\sigma} = S\dot{x} \quad (\text{D.20})$$

$$\dot{\sigma} = Sf(x, t) + Sg(x, t)u \quad (\text{D.21})$$

Injectons de l'équation (D.16) dans l'équation (D.21) :

$$\dot{\sigma} = Sf(x, t) + Sg(x, t)u^0 + Sg(x, t)U \operatorname{sgn}(\sigma) \quad (\text{D.22})$$

En posant :

$$D(x, t) = Sg(x, t)U \quad (\text{D.23})$$

$$d(x, t) = Sf(x, t) + Sg(x, t)u^0 \quad (\text{D.24})$$

L'équation (D.22) devient donc :

$$\dot{\sigma} = D(x, t) \operatorname{sgn}(\sigma) + d(x, t) \quad (\text{D.25})$$

avec $D(x, t)$ matrice symétrique arbitraire dont les mineurs principaux de $-D(x, t)^{-1}$ sont positifs et bornés.

La transformation non singulière choisie est :

$$\Omega(x, t) = U^T U^{-1} \quad (\text{D.26})$$

donc :

$$u^*(t) = \Omega(x, t) u(t) = U^T U^{-1} u(t) \quad (\text{D.27})$$

$$u(t) = U(U^*)^{-1} u^*(t) \quad (\text{D.28})$$

de (D.23) on obtient :

$$u(t) = [Sg(x,t)]^{-1} D(x,t) (U^*)^{-1} u^*(t) \quad (\text{D.29})$$

avec :

$$U^* = \text{diag} [\frac{1}{2}(u_1^{*+} - u_1^{*-}) \dots \frac{1}{2}(u_m^{*+} - u_m^{*-})] \quad (\text{D.30})$$

et :

$$u_i^*(t) = \begin{cases} u_i^{*+}(t) & \text{si } \sigma_i(x) > 0 \\ u_i^{*-}(t) & \text{si } \sigma_i(x) < 0 \end{cases} \quad (\text{D.31})$$

de (D.29) on a :

$$u^0 = [Sg(x,t)]^{-1} D(x,t) (U^*)^{-1} u^{*0} \quad (\text{D.32})$$

on remplace dans (D.24) on obtient :

$$d(x,t) = Sf(x,t) + D(x,t) (U^*)^{-1} u^{*0} \quad (\text{D.33})$$

avec :

$$u^{*0} = [\frac{1}{2}(u_1^{*+} - u_1^{*-}) \dots \frac{1}{2}(u_m^{*+} - u_m^{*-})]^T \quad (\text{D.34})$$

le domaine D est un domaine du mode de glissement si :

$$\sup_{x \in D} |\lambda_i| < 1 \quad i=1, \dots, m \quad (\text{D.35})$$

$$\lambda = -D(x,t)^{-1} d(x,t) \quad (\text{D.36})$$

la condition est vérifiée si :

$$u^{*0} = -U^* D(x,t)^{-1} Sf(x,t) \quad (\text{D.37})$$

Méthode 2

Cette deuxième méthode est basée sur une transformation toujours non singulière, la surface de glissement est :

$$\dot{\sigma}^* = \Omega(x,t) D(x,t) \text{sgn}(\sigma^*) + \Omega(x,t) d(x,t) + \dot{\Omega}(x,t) \Omega^{-1}(x,t) \sigma^* \quad (\text{D.38})$$

avec $D(x,t)$ et $d(x,t)$ préalablement définis.

Posons $\Omega = D^* D^{-1}$, avec $D^*(x,t)$ choisi de manière à ce que les mineurs principaux de $-D^*(x,t)^{-1}$ soient positifs et bornés, ainsi l'équation (D.38) devient :

$$\dot{\sigma}^* = D^*(x,t) \text{sgn}(\sigma^*) + D^*(x,t) D^{-1}(x,t) d(x,t) + \frac{d[D^* D^{-1}]}{dt} D^* D^{-1} \sigma^* \quad (\text{D.39})$$

En supposant $\|\dot{\Omega}\|$ et $\|\Omega^{-1}\|$ bornées, et en prenant σ suffisamment petite, D est alors un domaine de glissement si (D.35) est vérifiée et (D.35) est vérifiée si :

$$u^0 = -(Sg(x, t))^{-1} Sf(x, t) \quad (\text{D.40})$$

alors :

$$\lambda = -D(x, t)^{-1} d(x, t) = 0 \quad (\text{D.41})$$

NB : Ces deux dernières méthodes sont bien explicitées par [39].

D.2 Fonctions d'adoucissement de la commande.

a) Commande douce à un seul seuil : La commande U_n varie entre $+|U|$ et $-|U|$ suivant une pente passant par l'origine (figure D.1) :

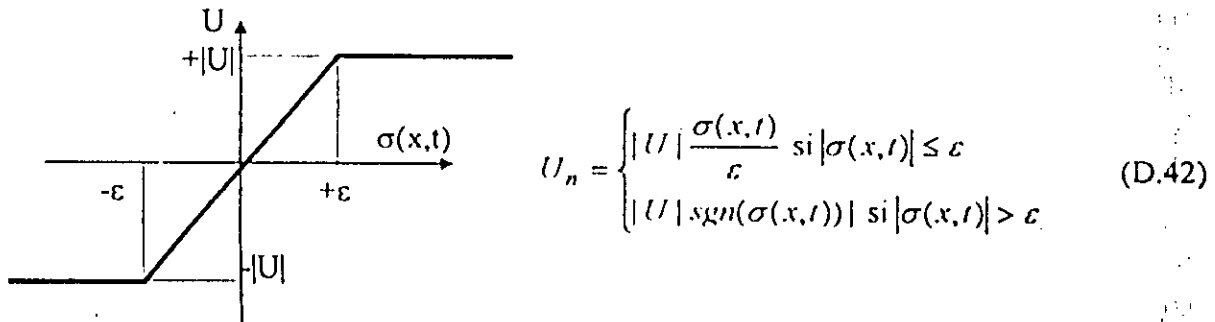


Figure D.1 Commande douce à un seuil

b) Fonction Smooth : C'est une commande douce à un seul seuil, mais dans ce cas la commande varie de $-|U|$ à $+|U|$ suivant une fonction régulière (Smooth : figure D.2).

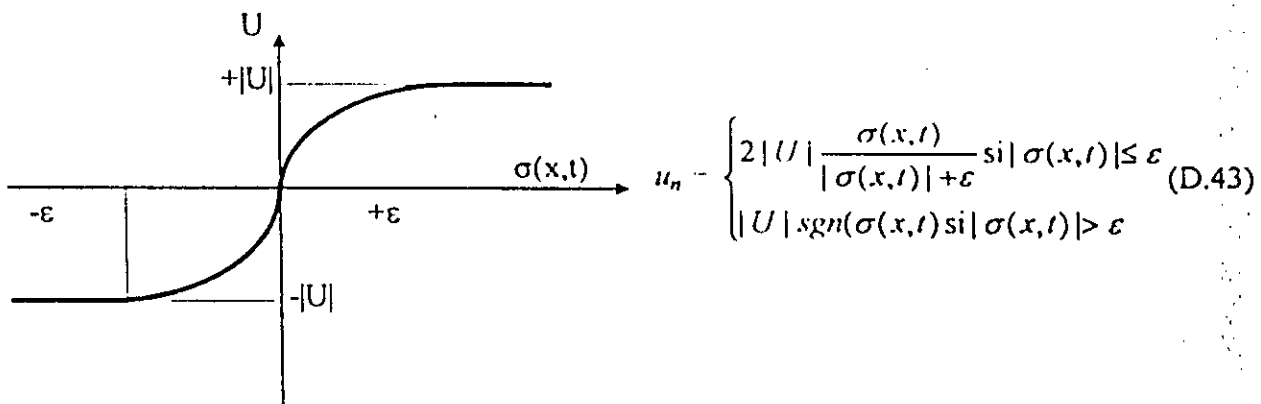


Figure D.2 Fonction Smooth

d) Commande douce à deux seuils : La commande u_n varie de $-|U|$ à $+|U|$ suivant une pente entre les deux seuils (figure D.3.) :

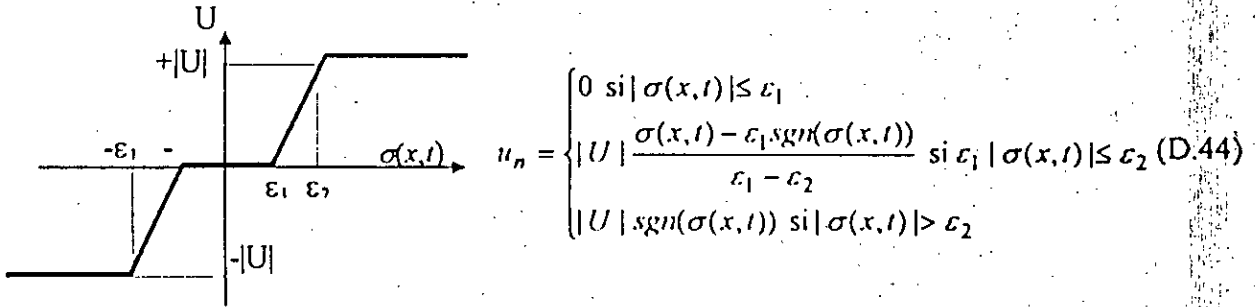


Figure D.3 Commande douce à 2 seuils

REFERNCES BIBLIOGRAPHIQUES

- [1] CHUEN-TSAI SUN, "Rule-base structure identification in an adaptative-network based Fuzzy inference system," *IEEE Trans. Fuzzy syst.*, vol.2, no.1, pp.61-73, February 1994.
- [2] J. C. WU and T. S. LIU, "A sliding-mode approche to Fuzzy control design," *IEEE Trans. Contr. Syst. Techno*, vol.4, no.2, pp.141-151, March 1996.
- [3] J. T. SPOONER and K. M. PASSINO, "Decentralized adaptative control of nonlinear systems using radial basis neural networks," *IEEE Trans. Automat. Contr*, vol.44, no.11, pp.2050-2057, November 1999.
- [4] M. S. AHMED, "Neural-Net based direct adaptative control for a class of nonlinear plants," *IEEE Trans. Automat. Contr*, vol.45, no.1, pp.119-124, January 2000.
- [5] K. S. NARENDRA and K. PARTHASARATHY, "Identification and control of dynamical systems using neural netwoks," *IEEE Trans. Neural Networks*, vol.1, no.1, pp.4-27, March 1990.
- [6] B. DRAZENOVIC, "The invariance conditions in variable systems," *Automatica*, vol.5, pp.287-295, 1969.
- [7] J. J. E. SLOINE and S. S. SASTRY, "Tracking control of nonlinear systems using sliding surface, with application to robot manipulators," *Int. J. Contr.*, vol.48, pp.465-492, 1983.
- [8] G. BARTOLINI and P. PYDYNOWSKI, "An improved, chattering free, V.S.C scheme for uncertain dynamical systems," *IEEE Trans. Automat. Contr.*, vol.41, no.8, pp.1220-1226, August 1996.
- [9] G. BARTOLINI, A. FERRARA and E. USAI, "Chattering avoidance by second-order sliding mode control," *IEEE Trans. Automat. Contr*, vol.43, no.2, pp.241-246, February 1998.
- [10] S. NICOSIA and P. TOMEI, "Robot control by using only joint position measurments," *IEEE Trans. Automat. Control.*, vol.35, no.9, pp.1058-1061, September 1990.
- [11] Y. H. KIM and F. L. LEWIS, "Neural network output feedback control of robot manipulators," *IEEE Trans. Robot. Automat.*, vol.15, no.2, pp.301-309, April 1999.
- [12] S. NICOSIA, P. TOMEI and A. TORNAMBE, "A nonlinear observer for elastic robot," *IEEE J. Robot. Automat.*, vol.4, no.1, pp.45-52, February 1988.
- [13] G. BARTOLINI and A. FERRARA, "Multi-input sliding mode control of a class of uncertain nonlinear systems," *IEEE Trans. Automat. Contr.*, vol.41, no.11, pp.1662-1666, November 1996.
- [14] Y. TANG, M. TOMIZUKA, G. GUERRERO and G. MONTEMAYOR, "Decentralized robust control of mechanical systems," *IEEE Trans. Automat. Contr.*, vol.45, no.4, pp.771-776, April 2000.
- [15] J. E. SLOINE, "The robust control of robot manipulators," *Int. J. Robotics Research*, vol.4, pp.49-63, 1985.

- [16] J. N. FULO et P. LOPEZ, *Initiation à la Robotique*, Le nouvel Automatismes, no.15, tome XXV, septembre 1980.
- [17] J. P. LALLEMANDS et S. ZEGHLOUL, *Robotique aspects fondamentaux*, Edition Masson, 1994.
- [18] B. GORLA et M. RENAUD, *Modèles des robots manipulateurs application à leur commande*, Edition Cepadues, Toulouse, 1984.
- [19] B. ARMSTRONG, O. KHATIB and J. BURDICK, "The explicit dynamic model and inertial parameters of the PUMA 560 arm." *In proc. IEEE Int. Conf on Robotics and automation*, San Francisco, CA, pp 510-518, 1986.
- [20] H. SERADJI, "Decentralized adaptative control of manipulator : theory, simulation and experimentation," *IEEE Trans. Robot. Automat*, vol 5, no.2, April 1989.
- [21] K. YEUNG and Y. P. CHEN, "A new controller design for manipulators using the theory of VSS," *IEEE Trans. Automat. Contr*, vol.33, no.2, Feb 1988.
- [22] T. MADANI, *Différentes approches de commande décentralisée à structure variable appliquées en Robotique*, Thèse de Magister, Ecole Nationale Polytechnique, 2000.
- [23] H. BUHLER, *Conception des systèmes automatiques*, Presses Polytechniques Romandes, 1988.
- [24] M. YEDDOU, *Etude de synthèse sur les réseaux de neurones et leurs applications*, Thèse de Magister, Ecole Nationale Polytechnique, 1998.
- [25] M. WEINFELD, *Réseaux de neurones*, Techniques de l'ingénieur, Traité informatique, H 1 990, pp. 1-16.
- [26] M. BOUKALLEL et S. MAOUCHE, *Application d'une approche neuro-flou à la commande d'un moteur à courant continu*, Projet de Fin d'Etudes, USTHB, 1998.
- [27] J. HERAULT, *Réseaux de neurones et traitement du signal*, Hermès, 1994.
- [28] F. CHITOUR et A. KAHLOUCHE, *La commande inverse par réseaux de neurones de la machine asynchrone*, Projet de Fin d'Etudes, Ecole Nationale Polytechnique, 1997.
- [29] J. A. FREEMAN and D. M.SKAPURA, *Neural networks algorithms, application and programming techniques*, Addison-Wesley Publishing Company 1991.
- [30] J. M. RENDERS, *Algorithmes génétiques et réseaux de neurones*, Hermès, 1995.
- [31] P. BRONE, *Commande et optimisation des processus*, Edition Technip, 1990.
- [32] H. BUHLER, *Réglages échantillonnés*, Volume 2, Presses Polytechniques Romandes, 1983.
- [33] B. DJERIDANE et M.DJEDDOU, *Sur les différents types de commandes et observateurs d'un simulateur de vol d'hélicoptère*, Projet de Fin d'Etudes, Ecole Nationale Polytechnique, 1999.
- [34] A. KARBOUA, *Commandes et observateurs par mode de glissement : application à une machine asynchrone alimentée en tension*, Thèse de Magister, Ecole Nationale Polytechnique, 1999.

- [35] J. F. SLOTINE, J. K. HERDICK and E. A. MIZAWA, "On sliding observer for nonlinear systems," *Jour. Dynam. Syst. Measur. Contr.*, vol.109/245, September 1987.
- [36] V. I. UTKIN, "Variable structure systems with sliding modes," *IEEE, Trans. Automat. Contr.*, vol.AC-22, no.2, pp.212-222, April 1977.
- [37] H. BUHLER, *Réglage par mode de glissement*, Presses Polytechniques Romandes, 1986.
- [38] F. BOUDJEMA, *Commande par mode de glissement application aux convertisseurs électriques*, Thèse de Doctorat, Université de Paul Sabatier, Toulouse, 1991.
- [39] V. I. UTKIN, *Sliding mode and their application in variable structure systems*, Moscow, Soviet Unio, MIR Publishers 1978.
- [40] G. P. MATHEWS, *Variable structure control of interconnected nonlinear systems*, Phd thesis, University Microfilms International, 1985.
- [41] J. J. E. SLOTINE, *Applied nonlinear control*, Edition Printence Hall, 1991.
- [42] T. MADANI et S. AMRATI, *Commande décentralisée à structure variable application en Robotique*, Projet de Fin d'Etudes, Ecole Nationale Polytechnique, 1997.
- [43] W. GAO, "Discrete-time variable structure control systems," *IEEE, Trans. Ind. Elect.*, vol.42, no.2, pp.117-122, Canada, 1995.
- [44] H. KHALIL, *Non linear systems*, Edition Printence Hall, 1996.
- [45] M. KRSTIC, I. KANELLAKOPOULOS and P.KOKOTOVIC, *Non linear and adaptative control design*, A Wiley-Interscience Publication, 1995.
- [46] M. KRSTIC and T. M. PORTZ, "Backstepping design for jet engin stall and surge control," *Proceeding of 34th conference on decision and control*, pp.3049-3055, December 1995.
- [47] S. JAIN and F. KHORRAMI, "Application of a decentralized adaptative out put feedback based on backstepping to power systems," *Proceeding of 34th conference on decision and control*, pp.1585-1590, December 1995.
- [48] J. XU, Q. JIA and T. LEE, "Analysis of adaptative robust backstepping control schemes for PM synchronous motors," *Proceeding of 34th conference on decision and control*, pp.3516-3521, December 1995.
- [49] M. F. HADJ MILOUD, *Commande adaptative par la technique de Backstepping appliquée en Robotique*, Projet de Fin d'Etudes, Ecole Nationale Polytechnique, 1998.
- [50] Y. H. PAO and B.IGELINK, "Mathematical concepts underlying the functional link approach," *In Proc. World Cong. Neural Net*, vol.1, pp.236-245, 1994.
- [51] N. SADEGH, "A perceptron network for functional identification and control of nonlinear systems," *IEEE, Trans. Neural. Networks*, vol.4, no.6, pp.982-988, 1993.

- [52] T. C. S. HSIA, T. A. LASKY and Z. GUO, "Robust independent joint controller design for industrial robot manipulators," *IEEE, Trans. Ind. Electron.*, vol.38, no.1, pp.21-25, February 1991.
- [53] M. CORLESS and G. LEITMANN, "Continuous state feedback guaranteeing uniform ultimate boundedness for uncertain dynamic systems," *IEEE, Trans. Automat. Contr.*, vol.26, pp.1139-1144, 1981.
- [54] C. VIBET, *Robots principe et contrôle*, Edition Ellipses, 1987.
- [55] R. P. PAUL, *Robot manipulators mathematics programming and control*, Library of congress cataloguing in publication data, 1981.
- [56] K. S. FU, R. C. GONZALEZ and C. S. G. LEE, *Robotics : control, sensing, vision and intelligence*, Edition Mc Graw Hill, 1987.

ملخص

قمنا في هذا العمل باقتراح نظام تحكم لا مركزي لضبط أنظمة ميكانيكية متشابكة، ممدجة بمعادلات Euler-Lagrange. في هذا النظام للتحكم نستعمل مرصد شبكات العصبونية من نوع FLNN، لرصد اشارات للتخللات و الاضطرابات. ثم نستعمل التحكم الانزلاقي المعمم للتخفيف من حدتهم. نظام التحكم المقترح يضمن استقرار النظام و يجعل أخطاء المتابعة لمسار أملس مرغوب تزول الى الصفر باستعمال الرجوع المرحلي. تطبيق التقنية المقترحة بالمحاكاة العددية لذراع آلي من نوع PUMA 560 و للدراسة يبينان امتيازات هذا التحكم مقارنة مع تقنية PD و تقنية جديدة نشرت سنة 2000. الكلمات المفتاحية: تحكم لا مركزي، مرصد، شبكات العصبونية، للتحكم الانزلاقي المعمم، الرجوع المرحلي.

Abstract

In this work, we propose a decentralized robust control for mechanical systems described by Euler-Lagrange equations and involving interconnections. This control uses a neural network's observer known as FLNN for values of interconnections and disturbances in order to decrease their effect by using generalized sliding mode. This control guarantees the stability of the system and tracking error convergence to zero, by using a Backstepping procedure, for a smooth desired trajectory. The quality of this control is related to the difference dynamics at each step of design procedure. The performances of the proposed control algorithm with comparative study are showed toward a simulation applied to PUMA 560 arm.

Key words

Decentralized control, observer, neural network, generalized sliding mode, Backstepping.

Résumé

Dans ce travail, nous proposons une commande décentralisée robuste pour des systèmes mécaniques fortement interconnectés, décrits par les équations d'Euler-Lagrange. Elle fait appel à un observateur par réseau de neurones de type FLNN pour les valeurs des interconnexions et perturbations afin d'atténuer leur effet indésirable par une commande synthétisée par mode de glissement généralisé. Cette commande garantit la stabilité du système, et une convergence rapide de l'erreur de poursuite vers zéro, en utilisant une commande par Backstepping, pour une trajectoire lisse désirée. La qualité de cette régulation réside dans l'ordre dynamique reliant les étapes qui la constituent. La simulation appliquée au robot manipulateur PUMA 560 et l'étude comparative illustrent les performances de l'algorithme de commande proposé.

Mots clés

Commande décentralisée, observateur, réseau de neurones, mode de glissement généralisé, Backstepping.