

ECOLE NATIONALE POLYTECHNIQUE  
Département de génie électrique

THÈSE

D0006/01B

préparée au

LABORATOIRE DE COMMANDE DES PROCESSUS

en vue de l'obtention du titre de

DOCTEUR D'ETAT

SPÉCIALITÉ : AUTOMATIQUE

par

المدرسة الوطنية المتعددة التخصصات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

Lakhdar GUENFAF

Magister en AUTOMATIQUE

Ecole Nationale Polytechnique d'Alger (ENPA)

---

**SUR L'APPLICATION DE L'AUTOMATIQUE  
ET DES RÉSEAUX DE NEURONES  
EN  
DYNAMIQUE DES STRUCTURES**

---

Soutenue le 25 Juin 2001 Devant le Jury composé de :

Président	<b>D. BERKANI</b>	Professeur /ENP
Directeurs de Thèse	<b>F. BOUDJEMA</b> <b>M.S. BOUCHERIT</b>	Professeur /ENP Maître de Conférence /ENP
Examineurs	<b>A. GUESSOUM</b> <b>E.M. BERKOUK</b> <b>H. SAARI</b>	Professeur /U. de Blida Maître de Conférence /ENP Maître de Conférence /U. Mostaganem
Invités	<b>B. TILIOUINE</b> <b>M. TADJINE</b> <b>H. AFRA</b>	Professeur /ENP Maître de Conférence /ENP Maître de Recherche /CGS

## Avant-Propos

Les travaux présentés dans ce manuscrit sont l'aboutissement de plusieurs années d'étude effectuées au Laboratoire de Commande de Processus (LCP) de l'Ecole Nationale Polytechnique.

J'exprime ma profonde reconnaissance à Ms F. BOUDJEMA et M.S. BOUCHERIT qui en dirigeant ces travaux, m'ont fait profiter de l'étendue de leurs connaissances, de leurs précieux conseils, de leurs aides, leur soutien et de l'intérêt bienveillant qu'ils m'ont témoigné. Qu'ils soient remerciés pour les nombreuses discussions, que nous avons eu et qui ont montré l'intérêt qu'ils portent à mes travaux.

Qu'il me soit permis de remercier Monsieur D. BERKANI, qui nous fait l'honneur d'accepter la présidence du jury de cette thèse.

Je remercie également Monsieur A. GUESSOUM, pour avoir accepté de se pencher sur ce travail, faire partie du jury et d'examiner la thèse.

Je remercie Monsieur E.M. BERKOUK, d'avoir accepté de juger ce travail.

J'adresse mes sincères remerciements à monsieur H. SAARI, pour avoir accepté de participer à ce jury et d'avoir examiné ce travail.

Je tiens à exprimer ma profonde reconnaissance à Monsieur B. TILIOUINE pour avoir accepté de participer au jury et pour la fructueuse discussion qui a montré l'intérêt qu'il porte à mes travaux.

J'adresse tout mes remerciements à Monsieur, M. TADJINE pour avoir accepté d'être membre du jury de cette thèse.

Je tiens à remercier profondément Monsieur, H. AFRA pour l'intérêt permanent qu'il a porté à mes travaux et d'avoir accepté de juger ce travail.

Mes remerciements s'adressent également à toute l'équipe du CGS (Centre du Génie parasismique) particulièrement M. HADID et F. BALI pour la bibliographie et les précieuses discussions scientifiques.

Je n'aurai garde d'oublier dans mes remerciements Nouredine BALI ainsi que sa femme Bahia et ses enfants (Amina et Youcef) pour leurs précieuses aides, encouragements et soutien moral.

A F.F. MESBAH, je voudrais lui exprimer ma profonde reconnaissance.

Je souhaite remercier Madame SEBAIBI pour ses conseils et ses qualités humaines.

A M. TEGUAR, qui n'a cessé de m'encourager pour l'aboutissement de ce travail, je voudrais lui exprimer ma profonde reconnaissance.

Ce travail n'aurai certainement pas vu le jour sans la confiance et le soutien permanents que m'ont témoignés mes parents tout au long de ma vie d'étudiant. Je suis très heureux de leur dédier cette thèse.

Enfin Je voudrai exprimer mes remerciements à tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

## Table des Matières

<b>Introduction Générale et Etat de l'Art</b>	<b>1</b>
<b>Chapitre I Modélisation des Structures Dynamiques</b>	<b>7</b>
Introduction.....	7
I.1 Modélisation géométrique des structures articulées.....	8
I.2 Modélisation dynamique des structures articulées.....	11
I.2.1 Formalisme d'Euler-Lagrange.....	11
I.2.2 Formalisme de Newton-Euler.....	15
I.3 Formulation des équations de mouvement de la structure en génie civil.....	17
I.4 Description du caractère non linéaire des structures en génie civil.....	20
I.4.1 Le modèle viscoplastique parfait.....	20
I.4.2. Le modèle viscoplastique Ecrouissable.....	21
I.4.3. Le modèle d'endommagement.....	21
I.5 Construction du modèle ARMAX de la structure en génie civil.....	23
I.5.1 Modèle sans introduction du seisme.....	23
I.5.2 Modèle avec introduction du seisme de Kanai-Tajimi.....	24
I.5.3 Modèle avec introduction du seisme de Clough-Penzien.....	25
I.6 Résultats de simulation.....	26
I.7 Conclusion.....	31
<b>Chapitre II Étude de synthèse sur les Réseaux de Neurones</b>	<b>32</b>
Introduction.....	32
II.1 Le neurone biologique.....	33
II.2. Le neurone formel.....	34
II.3 Modèle mathématique général du neurone.....	35
II.3.1 fonction discriminante.....	35
II.3.2 Le système dynamique linéaire SISO.....	37
II.3.3 La fonction d'activation non linéaire.....	37

II.4 Connexions neuronales et champs de neurones .....	38
II.5 Architecture des réseaux de neurones.....	38
II.5.1 Réseaux statiques.....	38
II.5.2 Réseaux dynamiques.....	39
II.5.3 Réseaux à architecture évolutive et réseaux Auto-Organisés.....	39
II.6 Stabilité des réseaux de neurones.....	40
II.7 Apprentissage des réseaux de neurones.....	41
II.8 Algorithme d'apprentissage.....	42
II.9 Rappel sur les différents types de commande par réseaux de neurones.....	43
II.9.1 Commande inverse .....	44
II.9.2 Commande supervisée.....	44
II.9.3 Commande par bouclage linéarisant.....	44
II.9.4 Commande adaptative par modèle de référence.....	45
II.9.5 Commande adaptative auto-ajustable.....	45
II.9.6 Commande prédictive.....	45
II.9.7 Commande à apprentissage par renforcement.....	46
II.10 Implémentation des réseaux de neurones.....	46
II.11 Conclusion.....	48
<b>Chapitre III Commande Active Linéaire Déterministe et Stochastique</b>	<b>49</b>
Introduction.....	49
III.1 Commande quadratique linéaire déterministe (LQC : Linear Quadratic Controller).....	50
III.1.1 La Commande 'Optimal Open-Closed Loop Control'.....	52
III.1.2 La Commande 'Optimal Closed Loop Control'.....	53
III.1.3 La Commande 'Optimal Open Loop Control'.....	53
III.2 Commande à variance minimale.....	53
III.2.1 Analyse des performances en boucle fermée.....	55
III.3 Commande à variance minimale généralisée.....	56
III.3.1 Analyse des performances en boucle fermée.....	56
III.4 Poursuite et Régulation à objectifs indépendants.....	57
III.5 Algorithme à placement de pôles.....	60
III.6 Placement de pôles et de zéros.....	62
III.7 Commande linéaire par poursuite d'un modèle. (linear model following control).....	64

III.8 Approche multivariable pour les structures articulées.....	65
III.8.1 Commande à GMV avec découplage du modèle de la structure.....	66
III.8.2 Commande multivariable multirate et mixte.....	68
III.9 Approches multivariées pour les structures en génie civil.....	68
III.9.1 Commande à variance minimale généralisée multivariable.....	69
III.9.2 Modèle ARMAX multivariable de la structure.....	70
III.9.3 Commande à GMV avec modèle empirique.....	71
III.10 Résultats de simulation.....	72
III.10.1 Commande des structures en génie civil.....	72
III.10.2 Commande des structures articulées.....	77
III.11 Conclusion.....	83
<b>Chapitre IV Commande non linéaire déterministe et stochastique</b>	<b>85</b>
Introduction.....	85
IV.1 Compensation de l'effet gravitationnelle.....	86
IV.2 Découplage non linéaire partiel.....	88
IV.3 Découplage non linéaire total.....	90
IV.4 Découplage non linéaire utilisant la géométrie différentielle (Linearisation exacte).....	98
IV.5 Commande Adaptative à Perturbation.....	98
IV.5.1 Modèle à perturbation 'Perturbation Model'.....	98
IV.5.2 Modèle aux différences à perturbation.....	101
IV.5.3 Formulation du problème de la commande "APC".....	102
IV.6 Commande linéarisante. "Computed Torque Control".....	103
IV.7 Commande linéarisante adaptative.....	105
IV.8 Commande adaptative à modèle de référence. (MRAC Model Reference Adaptive Control).....	107
IV.9 Synthèse d'un contrôleur minimal MCS. (Minimum Controller Synthesis).....	109
IV.10 Stratégie de commande par réseaux de neurones.....	110
IV.10.1 Algorithme d'apprentissage.....	111
IV.10.2 Commande d'une structure en génie civil à un seul degré de liberté.....	112
IV.10.3 Commande d'une structure en génie civil à plusieurs degrés de liberté.....	113
IV.11 Commande à gain programmé.....	115
IV.12 Commande à gain programmé des structures en génie civil.....	115

---

IV.13	Commande à variance minimale généralisée avec classification par domaine de Fonctionnement.....	118
IV.14	Commande à variance minimale généralisée avec classification par réseaux de Neurones.....	119
IV.15	Résultats de simulation.....	120
	IV.15.1 Commande des structures articulées.....	120
	IV.15.2 Commande des structures en génie civil.....	132
IV.16	Conclusion.....	141
	<b><i>Conclusion Générale</i></b> .....	<b>143</b>
	<b><i>Bibliographie</i></b> .....	<b>146</b>
	<b><i>Annexes</i></b> .....	<b>151</b>

## *Introduction Générale*

### *Et*

### *Etat de l'Art*

---

Au cours des vingt dernières années, des contributions en commande des structures, aussi bien théoriques que pratiques, ont apporté des éléments nécessaires à une meilleure compréhension des systèmes de contrôle actif [Hou97][Lan86][Lan81][Lan74][Ben91]. Le principal objet de la commande est la synthèse de lois de commande satisfaisant un certain niveau de performance. Afin de réaliser cet objectif, quatre types de commande sont utilisés : la commande passive, la commande active, la commande semi-active et la commande hybride. Dans une commande passive, aucune source d'énergie extérieure n'est utilisée et la force de commande est produite par l'énergie potentielle due à la dynamique de la structure. Par contre, dans la commande active, une énergie extérieure est fournie aux actionneurs pour appliquer la force nécessaire. Cette commande est plus efficace que la commande passive du fait de la présence d'une boucle de retour. Dans ce cas, les forces de commande sont fonction des réponses du système. La combinaison des commandes passive et active conduit à ce qu'on appelle une commande hybride. Enfin, la commande semi-active est un type de commande active qui ne nécessite qu'une faible quantité d'énergie extérieure à fournir. A cet effet, plusieurs algorithmes et stratégies de commande ont été proposés [Hou97][Smi94][Yan87][Ban98][Cal98][Rod87][Lu98][Yam90].

Pour effectuer l'analyse et la synthèse d'un système dynamique, il est nécessaire de connaître les relations entre ses grandeurs d'entrées et de sorties. L'ensemble de ces relations constitue le modèle mathématique du système. L'efficacité de ce modèle repose sur une analogie entre le comportement des objets physiques et celui des êtres mathématiques. Dans le cas des systèmes mécaniques, la représentation interne sera formulée au moyen des lois de la dynamique des structures. L'hypothèse de linéarité dans les problèmes de vibration conduit à représenter l'équation du mouvement par une équation différentielle linéaire à coefficients constants. Lorsque les non linéarités sont importantes, on distingue les méthodes utilisant la représentation par la série de Voltera, par la série de Winner, par les modèles de Winner-Hammerstein et des modèles utilisant les dérivées d'ordre fractionnaire [Arg90]. On trouve dans la littérature plusieurs type de modèles déterministes et stochastiques [Loh96]. Le modèle ARMA (Auto Regressive Moving Average) permet d'établir une relation linéaire entre les échantillons d'entrées et de sorties dans le cas déterministe [Lan93]. Les erreurs de mesure, de modélisation et les perturbations non mesurables nécessitent l'utilisation d'autres types de modèles. Ces dernières introduisent les entrées auxiliaires stochastiques dans le modèle général de la structure. On distingue plusieurs types : Le modèle ARX, le modèle ARMAX, le modèle BOX JENKINS et le modèle d'erreur de sortie [Loh96][Ast70]. Pour les structures en génie civil, nous avons construit un modèle se basant essentiellement sur la caractérisation de l'interaction sol-structure [Gue01]. Nous avons mis en évidence l'aspect aléatoire du processus et la nature de sortie auxiliaire nous permettant de l'introduire dans le modèle développé. L'excitation sismique est considérée comme la principale source de

perturbation extérieure. Le comportement de la structure sous l'effet de cette excitation est modélisé par un processus stochastique [Gue01][Pag99]. En effet, cette excitation est considérée comme un processus aléatoire modélisé par un bruit blanc filtré. Nous avons alors développé des techniques de commande qui prennent en considération l'aspect stochastique des perturbations et l'introduire dans la description du modèle du système ou dans la formulation de la loi de commande [Gue01].

La commande des robots manipulateurs constitue à l'heure actuelle l'une des principales préoccupations des recherches [Rah87][Cra87][Ben91]. Cet intérêt est étroitement lié à la difficulté de commander un robot auquel on demande l'exécution de tâches aussi bien précises que diversifiées. Cette précision est plus difficile à obtenir si la vitesse d'évolution augmente. En effet, la présence des couplages entre les liaisons complique la conception de la commande. L'obtention du modèle du robot s'avère nécessaire pour faciliter la synthèse des lois de commande. L'élaboration du modèle de connaissance nécessite une étude approfondie et détaillée sur la structure du robot. La transformation de Denavit-Hartenberg permet d'aboutir au modèles cinématique et géométrique directes et inverses du robot [Den55][Pau81]. La même transformation offre une souplesse dans le calcul du modèle dynamique direct et inverse [Fu87][Asa85][Kor85].

La commande auto-ajustable des structures articulées nécessite la généralisation des algorithmes monovariables aux cas multivariable. Borison [Bor79] propose un régulateur auto-ajustable multivariable, basé sur la minimisation de la variance du vecteur de sortie, utilisant un estimateur récursif et un contrôleur linéaire obtenu directement des estimés actuels. L'extension de la méthode de Clarke & Gawthrop [Cla79] au cas multivariable est présentée par Koivo [Koi80], dans laquelle l'estimation des paramètres du régulateur se fait avec l'algorithme SRA (Square Root Algorithm). Goodwin [Goo81a] expose l'approche directe de la VM, en utilisant un modèle de représentation Diagonale-Pleine [Gue95b]. Tandis que dans son second article [Goo81b], il présente le même algorithme avec un modèle Plein-Plein [Gue95b]. En utilisant un modèle multivariable CARIMA (Controlled AR Integrated MA) et une connaissance sur la borne supérieure de l'interacteur, Scatoline & Clarke [Sca85] développe la version explicite multivariable de l'algorithme à GMV (Generalized Minimum Variance). La version implicite du GMV est donnée par Mahieddine [Mah89].

Mei-Hua Liu & Wei Lin présentent dans leur premier article [Mei87], une méthode de calcul récursive des polynômes de pondération, pour assurer la spécification des pôles en BF, dans le cas de la GMV. Tandis que dans leur second article [Mei87], ils présentent l'approche directe avec un modèle Diagonale-Diagonale [Gue85b]. Shi-Jun Lang [Shi86] expose l'algorithme à GMV, dans lequel il calcule d'une façon récursive les polynômes de pondération et assurant un découplage total du système en BF. L'extension de la méthode aux systèmes bilinéaires à été développée par De La Sen [Del86]. Toivonen [Toi84] présente l'approche multivariable de LQC. L'algorithme à critère quadratique, sous forme d'état, est facilement extensible au cas multivariable. Le retard d'un système multivariable se traduit par la matrice interacteur, dont Dugard [Dug84] présente son rôle, en développant l'algorithme à VM avec une connaissance seulement sur la borne supérieure de cette matrice. Tandis que Zhang & Lang [Zha89] présente l'algorithme à GMV dans le cas où la matrice interacteur est quelconque et le modèle du système est non linéaire (modèle Hamerstein). Les modèles découplés utilisés par Zi-Li-Deng [Den91], Mei Hua liu [Mei91], Tham [Tha91] permettent l'implémentation directe des algorithmes monovariables aux cas multivariable. Ils permettent en outre l'identification en temps réel de quelques paramètres du procédé et la résolution des équations Diophantines simples (cas monovariable), sans avoir recours à identifier toutes les dynamiques du système qui nécessitent un temps de calcul énorme.

Ainsi nous avons choisi un modèle de représentation aux différences multivariable découplé, afin de permettre l'implémentation des algorithmes monovariables pour la commande des robots [Gue95a].

---

Les régulateurs auto-ajustables sont basés sur l'estimation en temps réel des paramètres du système (modélisé sous forme d'équations aux différences). La commande est calculée en utilisant les paramètres estimés. On distingue deux approches de commande auto-ajustable : la commande adaptative directe et indirecte. Dans le schéma indirect, on calcule les paramètres du régulateur à partir de ceux du système (identifiés de manière récursive). Tandis que dans l'approche directe, on identifie directement les paramètres du régulateur [Ast73][Cla75][M'sa85][Lan86][Wel91].

L'existence des termes non linéaires, dans le modèle du robot, nécessite leur compensation, pour obtenir des modèles moins complexes facilement commandables. L'utilisation d'un algorithme superviseur, pour la commande du modèle obtenu, aboutit à la commande non linéaire hybride, constituée d'une boucle de compensation (continu) et d'une commande discrète [Fu87][Vib87][Gaw81][Isi85][Gra89][Gue95a].

Nous avons en outre utilisé un modèle de représentation aux différences linéaire, pour le calcul de la commande. Le comportement linéaire peut être obtenu par linéarisation, autour d'une trajectoire nominale (variable dans le temps). On obtient ainsi le modèle à perturbation avec lequel on synthétise la commande [Fu87][Ath71]. La commande ainsi réalisée est caractérisée par deux actions. Une action anticipatrice, qui utilise la commande nominale calculée à partir des caractéristiques cinématiques de la trajectoire désirée. La seconde est une action de retour, qui se base principalement sur le modèle à perturbation. La commande ainsi obtenue est la commande adaptative à perturbation [Gue96b].

La connaissance de la structure du modèle décrivant le comportement dynamique du robot, offre la possibilité du choix d'une paramétrisation [Cra88]. Par opposition aux commandes précédentes, qui passent par une phase d'approximation. Cette paramétrisation (linéaire) représente la structure exacte du modèle dynamique sinon la plus proche possible du modèle. La synthèse d'une loi de commande adaptative, basée sur cette paramétrisation, aboutit à la commande linéarisante adaptative [M'si93][Ben91]. La stabilité d'un tel algorithme est étudiée en utilisant la théorie de stabilité de Lyapounov et d'hyperstabilité de Popov [Lan79][Lan89]. L'analyse de la convergence et la synthèse des lois d'adaptation peuvent être élaborées en utilisant les techniques adaptatives à modèle de référence [Lan79].

Les systèmes adaptatifs à modèle de référence sont utilisés dans de larges domaines, pour résoudre une variété importantes de problèmes rencontrés en commande, identification et estimation d'état [Sto83][Sto89]. En effet la synthèse des lois, est faite en utilisant la théorie d'hyperstabilité de Popov, pour assurer une convergence asymptotique de l'erreur. Plusieurs versions de cette commande en découlent (MRAC: Model Reference Adaptive Control, MCS: Minimum Controller Synthesis).

Pour les différents types de compensation non linéaire, nous avons élaboré des algorithmes de commande hybrides permettant le couplage entre une commande continue et une commande discrète [Gue95c]. Par ailleurs, nous avons présenté la commande mixte et multirate permettant de spécifier la loi de commande pour chaque articulation ainsi que la période d'échantillonnage adéquate

La commande active des structures en génie civil est un domaine de recherche interdisciplinaire caractérisé par un type particulier de problèmes [Hou97]. Le plus important de ces problèmes, est la taille ou la dimension du système à commander. En effet, les structures sont des systèmes massifs qui nécessitent des actionneurs puissants pouvant produire des forces allant jusqu'au milliers de kilo-Newtons. Les problèmes de modélisation ne sont pas écartés à cause de la nature intrinsèque de la structure et de ses perturbations. En effet, les structures sont en réalité à degré de liberté infinie et ne peuvent être complètement observées par le biais de capteurs installés à des emplacements spécifiés. Le caractère aléatoire et imprévisible des perturbations (séismes), ainsi que les variations et incertitudes paramétriques rendent la tâche de commande encore plus difficile. Il est intéressant

Dans le cadre de cette thèse, notre travail porte sur à l'application de l'automatique et des réseaux de neurones en dynamique de structure. Les excitations de Kanai Tajimi et de Clough Penzien seront utilisées tout au long de notre travail afin d'évaluer les performances des techniques développées pour les structures en génie civil. Pour les structures articulées, nous avons choisi deux trajectoires de référence. Une trajectoire polynomiale qui assure une continuité en position et en vitesse. Une autre trajectoire spécifiée par la fenêtre de VIVIANI, permet d'activer toutes les articulations et de valider les différents algorithmes utilisés.

Dans le chapitre un, nous nous intéressons à l'élaboration des différents modèles du robot utilisé. L'étude cinématique en utilisant la transformation de Denavit-Hartenberg, permet l'obtention des modèles géométriques directs et inverses. Les deux modèles dynamiques, calculés en utilisant les formalismes de Newton-Euler et celui d'Euler-Lagrange, sont exposés. En second lieu, l'élaboration des modèles dynamiques des structures en génie à plusieurs degrés de liberté sous l'effet d'une excitation sismique est développée. Nous présentons ensuite les différentes non linéarités existantes dans le comportement rhéologique.

Le deuxième chapitre présente les concepts de base des réseaux de neurones. Dans un premier lieu, nous exposons les différentes architectures de réseaux existants ainsi que leurs algorithmes d'apprentissage. En second lieu, une synthèse des différentes lois de commande à base de réseaux de neurones est donnée avec des outils d'implémentation en temps réel.

Le chapitre trois est consacré à une synthèse des différentes lois de commande linéaires déterministes et stochastiques. Dans un premier lieu, on présente l'algorithme à variance minimale. L'algorithme à variance minimale généralisée est exposé dans la section III.2. Les autres sections sont consacrées aux algorithmes suivants: Poursuite et Régulation avec Pondération de l'Entrée (PRPE), Placement de Pôles (PP), Placement de Pôles et de Zéros (PPZ), Linear Quadratic Controller (LQC), Linear Model following Controller (LMFC). Pour chaque technique, on présente le modèle ARMAX utilisé pour l'élaboration de la loi de commande. On met aussi en évidence l'intérêt de la connaissance du modèle de l'excitation et son incorporation dans l'élaboration du modèle discret de la structure en génie civil.

Dans le cadre du chapitre quatre, La commande non linéaire hybride est développée. Nous présentons les différentes méthodes de calcul des retours non linéaires, à savoir, le retour par compensation de gravité, le découplage non linéaire partiel, le découplage non linéaire total et le découplage utilisant la géométrie différentielle. Nous présentons aussi la commande adaptative à perturbation. Dans la section IV.6, nous développons la commande linéarisante à paramètres fixes et adaptative. La commande adaptative à modèle de référence (MRAC) et le développement de l'algorithme MCS font l'objet des sections IV.8 et IV.9. La dernière partie de ce chapitre sera consacrée aux stratégies de commande par réseaux de neurones et la commande à gain programmé. Une version neuronale de cette approche est présentée dans le but d'améliorer les performances du point de vue rapidité et temps de calcul.

Enfin nous terminons notre travail par une conclusion générale et quelques perspectives.

---

## *Chapitre I*

### *Modélisation des Structures Dynamiques*

---

#### **Introduction**

Dans le cadre de l'automatique, modéliser un système consiste à établir un ensemble de relations mathématiques qui permettent de décrire, avec une précision suffisante, les interactions internes entre ce système et son environnement externe. Lorsque les relations suscitées sont issues des équations de la physique, le modèle obtenu est dit modèle de connaissance. Si ces relations découlent des observations disponibles sur le système, on aboutit ainsi au modèle de représentation obtenu par identification. On distingue deux types de représentation : la représentation externe et la représentation interne [Gie95][Arg90].

La représentation externe est une notion de la théorie des systèmes et couramment utilisée en automatique. Elle est fondée sur des théories essentiellement mathématiques et elle a une portée générale ; ainsi elle permet de représenter aussi bien le comportement du système mécanique que celui d'un système chimique ou sociologique. La représentation interne ou une connotation plus concrète, plus 'physique' puisqu'elle s'appuie sur l'écriture des lois régissant le comportement des systèmes étudiés.

Pour des structures à géométrie complexes formées de divers assemblage mécanique, les techniques de calculs telles que la méthode des éléments finis, sont particulièrement utilisées pour traiter les problèmes qui ne peuvent être résolus analytiquement [Zie79].

En effet la détection des non linéarité s'effectue généralement en étudiant la déviation d'une propriété linéaire appliquée au cas non linéaire, autrement dit la violation des hypothèses liées au comportement linéaire. Pour mettre en évidence ces non linéarités, on excite la structure avec des amplitudes différentes et on vérifie l'application du principe de superposition en comparant les réponses obtenues.

La grande diversité des types de non linéarité de structure (non linéarité géométrique, non linéarité de comportement) rend difficile le problème de leur modélisation. Les non linéarités de comportement les plus souvent citées sont celles des raideurs, de dissipation et de frottement sec. Les non linéarités géométriques sont celles qui tiennent compte des termes d'ordre supérieur à un dans la relation déformation-déplacement [Afr95][Arg90].

Dans le domaine de la robotique l'élaboration du modèle nécessite une étude approfondie et détaillée sur la structure du robot. L'utilisation de la transformation de Denavit-Hartenberg (DH) facilite la description géométrique du manipulateur, qui nous permet d'aboutir au modèle cinématique et géométrique direct et inverse du robot. La même transformation offre une souplesse dans le calcul du modèle dynamique direct, en utilisant le formalisme d'Euler-Lagrange (EL), et du modèle dynamique inverse, en utilisant le principe de D'Alembert (Algorithme de NE) [Cra86][Fu87][Pau81].

Le principe des déplacements virtuels s'énonce ainsi : Si un système en équilibre sous l'action d'un ensemble de forces extérieures est soumis à un déplacement virtuel, i.e., une forme de déplacement compatible avec les contraintes du système, la somme des travaux effectués par l'ensemble des forces est égale à zéro [Fu87]. Ce principe est appliqué lorsque le système présente une certaine complexité incluant un nombre de masses ponctuelles ou de corps interconnectés, où l'application de l'équilibre direct des forces peut être difficile.

Par contre l'approche variationnelle, appelée aussi le principe de Hamilton, utilise les variations des énergies potentielles et cinétiques pour formuler les équations de mouvement du système. Le principe s'énonce ainsi : La somme des variations dans le temps de la différence des énergies cinétiques et potentielles et du travail effectué par les forces non conservatives dans un intervalle de temps  $[t_1, t_2]$  est égale à zéro [Kor85].

Le développement qu'a connu l'emploi des calculateurs dans la saisie de mesure et le contrôle de structure nous amène à caractériser la structure que l'on se propose d'étudier par des modèles de type entré-sortie de nature discrète. Ceci rend plus facile la résolution de problème de commande discrète par rapport au problème de commande dans le cas continu.

Ce chapitre présente les étapes et les procédures pour élaborer les modèles dynamiques des structures. Dans le cas du génie civil, l'accélération sismique est considérée comme la principale source d'excitation extérieure. Pour les structures articulées, nous avons développé les différents types de modèles dynamiques et une génération de trajectoire spatiale (fenêtre de Viviani)

## 1.1 Modélisation géométrique des structures articulées.

Parmi les différentes classes de robot établies par D.P.Stoten, et qui sont au nombre de huit [Sto87], nous avons opté pour un robot de classe quatre. Ce manipulateur est caractérisé par une articulation rotationnelle  $\theta_2$  et deux articulations translationnelles, dont les mouvements sont identifiés par les variables  $d_1$  et  $d_2$  (Figure 1.1).

Afin d'établir les différents modèles, plusieurs hypothèses doivent être prises en considération[Sto87]:

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

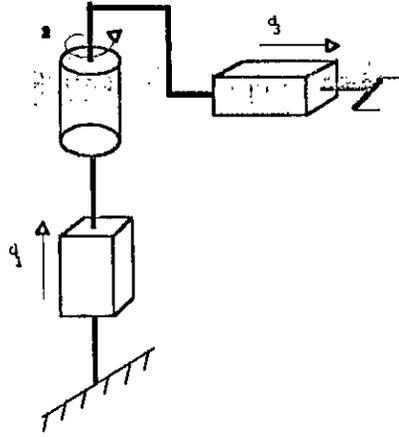


Figure I.1 Schéma du robot

Tout manipulateur peut être considéré comme une chaîne de liaisons connectées par des articulations. Chaque liaison est caractérisé par son propre repère. Utilisant les transformations homogènes, on peut décrire la position et l'orientation de chaque repère par rapport à un autre. Denavit et Hartenberg [Den55][Pau81] ont établi une transformation qui permet le passage d'une liaison à la prochaine, en utilisant quatre paramètres. L'implémentation des repères dans chaque liaison, en utilisant l'algorithme cité en annexe 1, est représenté sur la figure I.2.

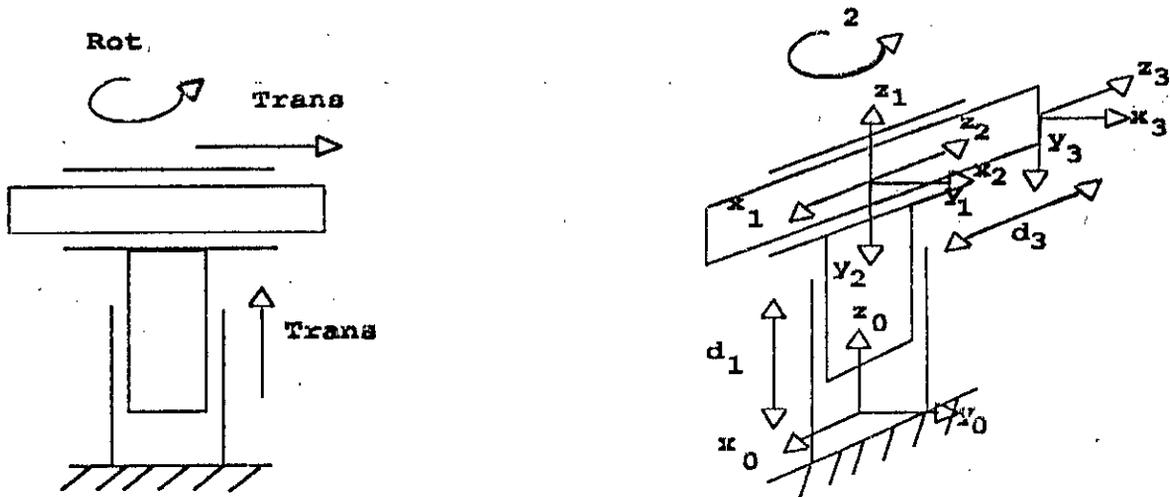


Figure I.2 Implementation des transformations de DH.

En rappelant que  $d_1, \theta_2$  et  $d_3$  sont les coordonnées généralisées du robot, on obtient le tableau suivant:

Maillon	Variables	$\alpha_i$	$a_i$	$d_i$	$\theta_i$
1	$d_1$	$0^\circ$	0	$d_1$	0
2	$\theta_2$	$-90^\circ$	0	0	$\theta_2$
3	$d_3$	$0^\circ$	0	$d_3$	0

Tableau I.1 Determination des parametres de DH.

Ainsi nous tirons les matrices de transformation, représentant les deux translations et la rotation, d'un repère à un autre.

$${}^0T^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad {}^1T^2 = \begin{bmatrix} C_2 & 0 & -S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad {}^2T^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.1)$$

${}^i T^{i+1}$ : Matrice de passage du repère  $R_i(x_i, y_i, z_i)$  au repère  $R_{i+1}(x_{i+1}, y_{i+1}, z_{i+1})$  avec  $C_i = \text{Cos } \theta_i$ ;  $S_i = \text{Sin } \theta_i$ .

La matrice de passage du repère  $R_0(x_0, y_0, z_0)$  à l'élément terminale est:

$$A = {}^0T^3 = {}^0T^1 {}^1T^2 {}^2T^3; \quad A = \begin{bmatrix} C_2 & 0 & -S_2 & -d_3 S_2 \\ S_2 & 0 & C_2 & d_3 C_2 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.2)$$

et la matrice de transformation de  $R_0(x_0, y_0, z_0)$  à  $R_2(x_2, y_2, z_2)$  est:

$${}^0T^2 = {}^0T^1 {}^1T^2; \quad {}^0T^2 = \begin{bmatrix} C_2 & 0 & -S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

Les transformations précédentes permettent d'exprimer, de proche en proche, ou à l'aide des relations de récurrence, la position de l'organe terminal par rapport au repère  $R_0(x_0, y_0, z_0)$ , sous la forme d'une fonction  $r(R_0)$ , à partir des variables articulaires  $q_i$ .

$$r(R_0) = F(q) \quad (1.4)$$

où  $q^T = [q_1, q_2, \dots, q_n]$ : n: degré de liberté du robot.

$r^T = [x \ y \ z \ 1]$ :  $F(q) = {}^0T^n \cdot {}_n r^n$ .

${}_n r^n$ : coordonnées de l'élément terminal dans  $R_n(x_n, y_n, z_n)$ .

$F(q)$ : fonction vectorielle.

$F: R^n \rightarrow R^4$

$q \rightarrow r = [x \ y \ z \ 1]^T = [f_1(q) \dots f_4(q)]^T$ .

Dans le cas du robot de classe 4 (n=3) on a:

${}_3 r^3 = [0 \ 0 \ 0 \ 1]$  et  $r = {}^0T^3 \cdot {}_3 r^3 = [-d_3 S_2 \ d_3 C_2 \ d_1 \ 1]^T$ .

Donc

$$\begin{aligned} f_1(q) &= -d_3 \text{Sin } \theta_2 \\ f_2(q) &= d_3 \text{Cos } \theta_2 \\ f_3(q) &= d_1 \\ f_4(q) &= 1 \end{aligned} \quad (1.5)$$

avec  $q^T = [d_1, \theta_2, d_3]$ .

Cette équation est appelée "*modèle géométrique*" du robot, par opposition au "*modèle cinématique*" qui relie la variation  $\Delta r(R_0)$  correspondant à une variation  $\Delta q$  [Pau81][Gue95a].

D'où l'on tire les coordonnées généralisées  $q_i$  :

$$\begin{aligned} d_1 &= z \\ \theta_2 &= \arctg\left(-\frac{x}{y}\right) \\ d_3 &= \frac{y}{\cos\theta_2} \end{aligned} \quad (1.6)$$

pour  $y \neq 0$  et  $\theta_2 \neq \frac{\pi}{2} + k\pi$ ;  $k \in \mathbb{Z}$ .

Cette équation est le "*modèle géométrique inverse*" du robot.

Dans le cas général, cette équation n'a pas de solutions uniques. Lorsqu'une solution unique est cependant trouvée, l'expression (1.6) peut être utilisée pour la commande cinématique des mécanismes considérés mais il ne s'agit très souvent que de systèmes particulièrement simple

## 1.2 Modélisation dynamique des structures articulées

Après avoir défini la Modélisation géométrique, nous nous intéressons dans cette partie à l'élaboration du modèle dynamique du robot, dans le but de la commande. La complexité structurelle des systèmes mécaniques articulés nous oblige à choisir une approche systématique pour résoudre ce problème. Le modèle dynamique peut être obtenu des lois de la mécanique newtonienne et lagrangienne. Les formalismes d'Euler-Lagrange et de Newton-Euler nous permettent d'aboutir aux équations du mouvement.

### 1.2.1 Formalisme d'Euler-Lagrange.

Le formalisme de Lagrange est utilisé pour modéliser le comportement dynamique d'un robot. Cette approche particulière est assez simple à mettre en oeuvre et elle est bien adaptée aux techniques de calcul manuel ainsi qu'aux méthodes de calcul assistées par ordinateur.

L'application direct du formalisme de Lagrange et de la transformation homogène de DH, aboutit à un algorithme compact pour décrire les équations dynamiques du mouvement. L'équation de Lagrange-Euler est [Set64][Pau81][Asa86][Cra86] :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} + \frac{\partial E_D}{\partial \dot{q}_i} = \tau_i, \quad i = 1, n \quad (1.7)$$

où  $L$  : le lagrangien défini par  $L = E_c - E_p$ .

avec  $E_c$  :Energie cinétique;  $E_p$  :Energie potentielle;  $E_D$  :Energie de dissipation.

$\tau_i$  :Force généralisée;  $n$  : degré de liberté.

Les matrices de passage de DH sont définies par [Fu87][Gue95a]:

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

pour une liaison rotative.

Si la liaison est prismatique, on pose  $a_i = 0$ .

Pour le calcul de l'énergie cinétique, l'expression de la vitesse est[Fu87]:

$${}^0V^i = \sum_{j=1}^i [U_{ij} q_j] r^j \quad (1.9)$$

tel que 
$$U_{ij} = {}^0T^{j-1} Q_{j,j-1} T^i; j \leq i$$
  

$$U_{ij} = 0; j \geq i$$
 et 
$$Q_j = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$
 si la liaison est rotative.

$$Q_j = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$
 si la liaison est tanslatinnelle;  $r^i$ : coordonnée du point i dans le

repère  $R_i(x_i, y_i, z_i)$ .

L'énergie cinétique de l'élément i dans la liaison i est [Pau91]:

$$dK_i = \frac{1}{2} \text{trace}(V_i V_i^T) dm = \frac{1}{2} \text{trace} \left[ \sum_{p=1}^i \sum_{r=1}^i U_{ip} J_i U_{ir}^T \dot{q}_p \dot{q}_r \right] \quad (1.10)$$

Donc l'énergie cinétique de la liaison i est

$$K_i = \int dK_i = \frac{1}{2} \text{trace} \left[ \sum_{p=1}^i \sum_{r=1}^i U_{ip} J_i U_{ir}^T \dot{q}_p \dot{q}_r \right] \quad (1.11)$$

avec 
$$J_i = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix}$$

Et l'énergie cinétique totale du robot est:

$$E_c = \sum_{i=1}^n K_i + E_{ca} \quad (1.12)$$

où  $E_{ca} = \frac{1}{2} \sum_{i=1}^n I_i \dot{q}_i^2$ :Energie cinétique introduite par les actionneurs [Fu87].

$I_i$ :moment d'inertie de l'actionneur i dans le cas la rotation et masse de l'actionneur dans le cas de la translation [Fu87][Rah87].

L'énergie potentielle est [Fu87][Lop84]:

$$P = \sum_{i=1}^n -m_i g^T \cdot {}_0r^i = \sum_{i=1}^n -m_i g^T ({}^0T^i \cdot r^i) \quad (I.13)$$

avec  $g^T = (g_x \ g_y \ g_z \ 1)$ . Dans le repère  $R_0(x_0 y_0 z_0)$ , on a  $g^T = (0 \ 0 \ -g \ 1)$ .

Et l'énergie de dissipation est :

$$E_D = \frac{1}{2} \sum_{l=1}^n f_l \dot{q}_l^2 \quad (I.14)$$

où  $f_l$  : coefficient de frottement visqueux.

Des équations (I.7) à (I.14) on calcule les forces généralisées[Fu87]:

$$\tau_i = \sum_{k=i}^n \sum_{p=1}^k \text{trace}[U_{kp} J_k U_{ki}^T] \dot{q}_p - \sum_{k=i}^n m_k g^T U_{ki} \cdot r^k + \sum_{k=i}^n \sum_{p=1}^k \sum_{r=1}^k \text{trace}[U_{kpr} J_k U_{ki}^T] \dot{q}_r \dot{q}_p + I_i \ddot{q}_i + f_i \dot{q}_i; \quad i = 1, n \quad (I.15)$$

$$U_{ijk} = {}_0T^{k-1} Q_{k \ k-1} T^{j-1} Q_{j \ j-1} T^i; \quad k \leq j \leq i,$$

$$\text{où } U_{ijk} = {}_0T^{j-1} Q_{j \ j-1} T^{k-1} Q_{k \ k-1} T^i; \quad j \leq k \leq i.$$

$$U_{ijk} = 0; \quad j \leq i \leq k$$

Dans le cas du robot de classe quatre on a :

$$Q_1 = Q_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad Q_2 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad U_{11} = {}_0T^0 Q_1 {}_0T^1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

$$U_{22} = {}_0T^1 Q_2 {}_1T^2 = \begin{bmatrix} -S_2 & 0 & -C_2 & 0 \\ C_2 & 0 & -S_2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad U_{32} = {}_0T^1 Q_2 {}_1T^3 = \begin{bmatrix} -S_2 & 0 & -C_2 & -d_3 C_2 \\ C_2 & 0 & -S_2 & -d_3 S_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

$$U_{33} = {}_0T^2 Q_3 {}_2T^3 = \begin{bmatrix} 0 & 0 & 0 & -S_2 \\ 0 & 0 & 0 & C_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \quad U_{222} = {}_0T^1 Q_2 {}_1T^1 Q_2 {}_1T^2 = \begin{bmatrix} -C_2 & 0 & S_2 & 0 \\ -S_2 & 0 & -C_2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

$$U_{21} = {}_0T^0 Q_1 {}_0T^2 = U_{11}; \quad U_{31} = {}_0T^0 Q_1 {}_0T^3 = U_{11};$$

$$U_{111} = U_{211} = U_{212} = U_{221} = U_{311} = U_{312} = U_{313} = U_{321} = [0];$$

$$U_{322} = {}_0T^1 Q_2 {}_1T^2 Q_2 {}_1T^3 = \begin{bmatrix} -C_2 & 0 & S_2 & d_3 S_2 \\ -S_2 & 0 & -C_2 & -d_3 C_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; J_i = \begin{bmatrix} \alpha_i & 0 & 0 & m_i \bar{x}_i \\ 0 & \beta_i & 0 & m_i \bar{y}_i \\ 0 & 0 & \gamma_i & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix}$$

Les produits d'inertie sont nulles car  $R_i(x_i, y_i, z_i)$  sont parallèles aux repères principaux de la liaison  $i$ .

${}_i r^i = [\bar{x}_i \bar{y}_i \bar{z}_i I]$  : centre de masse de la liaison  $i$  relatif au repère  $R_i$ .

$m_i$  : masse de la liaison  $i$ .

$I_i \approx 0$  : dynamique des actionneurs négligeable.

De l'équation (I.15) on a:

$$F_1 = (m_1 + m_2 + m_3)(\ddot{q}_1 + g) + f_1 \dot{q}_1$$

$$\tau_2 = [\alpha_2 + \gamma_2 + \alpha_3 + \gamma_3 + d_3 m_{33} \bar{z}_3 + m_3 (d_3 \bar{z}_3 + d_3^2)] \ddot{q}_2 + m_3 \bar{x}_3 \dot{q}_3 + 2m_3 (\bar{z}_3 + d_3) \dot{q}_3 \dot{q}_2 + f_2 \dot{q}_2 \quad (\text{I.16})$$

$$F_3 = m_3 \bar{x}_3 \dot{q}_3 + m_3 \dot{q}_3 - m_3 (\bar{z}_3 + d_3) \dot{q}_2^2 + f_3 \dot{q}_3$$

Suivant la structure de notre robot on a:

$${}_2 r^2 \approx 0, m_2 \approx 0; \alpha_2 = \gamma_2 \approx 0; \bar{x}_2 = \bar{y}_2 = \bar{z}_2 \approx 0, \bar{x}_1 = \bar{y}_1 = 0; \bar{z}_1 = -\frac{l_1}{2}; \bar{x}_3 = \bar{y}_3 = 0; \bar{z}_3 = -\frac{l_2}{2}$$

$$\alpha_3 = \frac{m_3 b^2}{12}; \gamma_3 \approx 0$$

En prenant  $F_1 = k_1 U_1; \tau_2 = k_2 U_2; F_3 = k_3 U_3$  L'équation (I.16) devient :

$$\begin{aligned} u_1 &= \frac{(m_1 + m_3)}{k_1} (\ddot{q}_1 + g) + \frac{f_1}{k_1} \dot{q}_1 \\ u_2 &= \frac{m_3 l_2^2 / 3 + 2m_3 \bar{z}_3 q_3 + m_3 q_3^2}{k_2} \ddot{q}_2 + \frac{2m_3}{k_2} (\bar{z}_3 + q_3) \dot{q}_3 \dot{q}_2 + \frac{f_2}{k_2} \dot{q}_2 \\ u_3 &= \frac{m_3}{k_3} \dot{q}_3 - \frac{m_3}{k_3} (\bar{z}_3 + q_3) \dot{q}_2^2 + \frac{f_3}{k_3} \dot{q}_3 \end{aligned} \quad (\text{I.17})$$

La forme d'état établie est:

$$\begin{aligned} \dot{x} &= A x + B U + d \\ y &= c x \end{aligned} \quad (\text{I.18})$$

Où  $x^T = (x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6) = (q_1 \ \dot{q}_1 \ q_2 \ \dot{q}_2 \ q_3 \ \dot{q}_3)$  : vecteur d'état.

$y^T = (y_1 \ y_2 \ y_3) = (q_1 \ q_2 \ q_3)$  : vecteur de sorties.

$U^T = (U_1 \ U_2 \ U_3)$  : vecteur de commandes.

$$\text{et } A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{f}{m_1+m_3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{f_2}{m_3 j^*} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{f_3}{m_3} \end{bmatrix}; B = \begin{bmatrix} 0 & 0 & 0 \\ k \frac{l_2}{m_1+m_3} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{k_2}{m_3 j^*} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{k_3}{m_3} \end{bmatrix};$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}; d^T = \begin{bmatrix} 0 & -g & 0 & -\frac{2(x_5 - \frac{l_2}{2})x_4 x_6}{j^*} & 0 & (x_5 - \frac{l_2}{2})x_4^2 \end{bmatrix}; J^* = \frac{l_2^2}{3} - l_2 x_5 + x_5^2.$$

### 1.2.2 Formalisme de Newton-Euler.

Dans la partie précédente nous avons élaboré des équations différentielles non linéaires du second ordre obtenues par le formalisme de Lagrange-Euler. L'utilisation de la matrice de DH 4x4 exige, en temps réel, un temps de calcul énorme pour la détermination des forces généralisées, à partir des données de la position, la vitesse et l'accélération pour une trajectoire donnée. Pour réduire ce temps de calcul, la formulation de Newton-Euler permet d'utiliser un algorithme récursif basé sur le principe de D'Alambert. Cet algorithme offre la possibilité de calculer directement les forces généralisées sans l'intermédiaire d'équations littérales. L'algorithme se compose de deux étapes [Fu87]:

#### 1° Première étape. "Forward équations" pour $i=1, n$ .

Dans cette étape nous calculons les vitesses, accélérations linéaires et angulaires de chaque liaison  $(V_i, \omega_i, a_i, \dot{\omega}_i)$ , exprimées dans le repère lié à la liaison considérée. nous commençons de la base jusqu'à l'élément terminal.

$${}_i R^0 \omega_i = {}_i R^{i-1} ({}_{i-1} R^0 \omega_{i-1} + Z_0 \dot{q}_i) \text{ si c1 est vérifiée.} \quad (1.19.a)$$

$${}_i R^0 \omega_i = {}_i R^{i-1} ({}_{i-1} R^0 \omega_{i-1}) \text{ si c2 est vérifiée.} \quad (1.19.b)$$

$${}_i R^0 \dot{\omega}_i = {}_i R^{i-1} [{}_{i-1} R^0 \dot{\omega}_{i-1} + Z_0 \ddot{q}_i + ({}_{i-1} R^0 \omega_{i-1}) \wedge (Z_0 \dot{q}_i)] \text{ si c1 est vérifiée.} \quad (1.19.c)$$

$${}_i R^0 \dot{\omega}_i = {}_i R^{i-1} ({}_{i-1} R^0 \dot{\omega}_{i-1}) \text{ si c2 est vérifiée} \quad (1.19.d)$$

$${}_i R^0 \dot{V}_i = ({}_i R^0 \dot{\omega}_i) \wedge ({}_i R^0 P_i^*) + ({}_i R^0 \omega_i) \wedge [({}_i R^0 \omega_i) \wedge ({}_i R^0 P_i^*)] + {}_i R^{i-1} ({}_{i-1} R^0 \dot{V}_{i-1}) \text{ si c1 est vérifiée} \quad (1.19.e)$$

$${}_i R^0 \dot{V}_i = ({}_i R^0 \dot{\omega}_i) \wedge ({}_i R^0 P_i^*) + 2({}_i R^0 \omega_i) \wedge ({}_i R^{i-1} Z_0 \dot{q}_i) + ({}_i R^0 \dot{\omega}_i) \wedge ({}_i R^0 P_i^*) + {}_i R^{i-1} ({}_{i-1} R^0 \dot{V}_{i-1} + Z_0 \ddot{q}_i) \text{ si c2 est vérifiée} \quad (1.19.f)$$

$${}_i R^0 \bar{a}_i = ({}_i R^0 \dot{\omega}_i) \wedge ({}_i R^0 \bar{S}_i) + ({}_i R^0 \omega_i) \wedge [({}_i R^0 \omega_i) \wedge ({}_i R^0 \bar{S}_i)] + {}_i R^0 \dot{V}_i \quad (I.19.g)$$

2°. Deuxième étape. "Backward equations" pour  $i=n,1$ .

Dans cette étape on calcule les forces et les couples, allant de l'élément terminal jusqu'à la base. L'élément terminal est soumis à une force  $f_{n+1}$  et un couple  $\tau_{n+1}$  issus du monde extérieur.

$${}_i R^0 f_i = {}_i R^{i+1} ({}_{i+1} R^0 f_{i+1}) + m_i {}_i R^0 \bar{a}_i. \quad (I.20.a)$$

$f_i$  et  $\eta_i$  sont respectivement la force et le moment exercés sur la liaison  $i$  par la liaison  $i-1$

$$\begin{aligned} {}_i R^0 \eta_i = & {}_i R^{i+1} [{}_{i+1} R^0 \eta_{i+1} + ({}_{i+1} R^0 P_i^*) \wedge ({}_{i+1} R^0 f_{i+1})] \\ & + ({}_i R^0 P_i^* + {}_i R^0 \bar{S}_i) \wedge ({}_i R^0 F_i) + ({}_i R^0 I_i \circ R^i) ({}_i R^0 \dot{\omega}_i) \\ & + ({}_i R^0 \omega_i) \wedge [({}_i R^0 I_i \circ R^i) ({}_i R^0 \omega_i)]. \end{aligned} \quad (I.20.b)$$

dans  $R_{i-1}(x_{i-1}, y_{i-1}, z_{i-1})$ .

$$\tau_i = ({}_i R^0 \eta_i)^T ({}_i R^{i-1} Z_0) + c_i \dot{q}_i \text{ si c1 est vérifiée} \quad (I.20.c)$$

$$\tau_i = ({}_i R^0 f_i)^T ({}_i R^{i-1} Z_0) + c_i \dot{q}_i \text{ si c2 est vérifiée} \quad (I.20.d)$$

La figure I.3 explicite les différents paramètres de l'algorithme [Kor85].

c1: Liaison translationnelle.

c2: Liaison rotationnelle.

Où  $F_i = m_i \bar{a}_i$ : Force extérieure exercée sur la liaison  $i$  en  $\bar{S}_i$ .

$N_i = I_i \dot{\omega}_i + \omega \wedge (I_i \omega_i)$ : Couple extérieur exercé sur la liaison  $i$  en  $\bar{S}_i$ .

${}_i R^0 P_i^* = [a_i, d_i \sin \alpha_i, d_i \cos \alpha_i]^T$ .

${}_i R^0 \bar{S}_i$ : centre de masse de la liaison  $i$  dans  $R_i(x_i, y_i, z_i)$ .

${}_i R^0 I_i \circ R^i$ : Matrice d'inertie de la liaison  $i$  autour de son centre de masse.

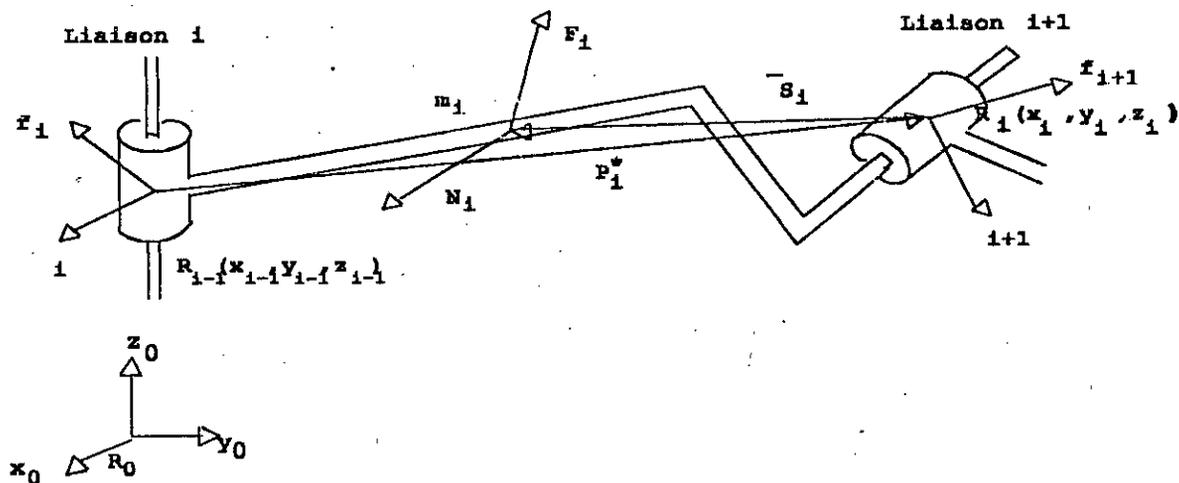


Figure I.3 Présentation des différentes grandeurs.

$${}_{i-1}R^i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i \\ 0 & S\alpha_i & C\alpha_i \end{bmatrix}; {}_iR^{i-1} = [{}_{i-1}R^i]^T; Z_0 = [0 \ 0 \ 1]^T.$$

$$\omega_0 = \dot{\omega}_0 = V_0 = 0; \dot{V}_0 = [g_x \ g_y \ g_z]^T; |\dot{V}_0| = g. \text{ (voir Annexe2)}$$

### I.3 Formulation des équations de mouvement de la structure en génie civil

En dynamique de structure, on parle de modèle dynamique dans le cas où la charge serait variable dans le temps. Plusieurs méthodes permettent l'élaboration des équations différentielles de la structure en tenant compte de certaines considérations aussi bien théoriques que pratiques. Du point de vue théorique, la structure est considérée comme étant un milieu continu à une infinité de degré de liberté. La discrétisation spatiale et la représentation par des modèles à degré de liberté fini est nécessaire afin de faciliter la manipulation des équations différentielles obtenues [Arg90][Clo93]. Des considérations pratiques nécessitent un modèle simplifié afin de pouvoir distinguer l'emplacement des mesures des excitations et des réponses à des endroits préalablement choisis.

Parmi les méthodes de discrétisation on distingue :

- *La procédure à masse concentrée* (Lumped mass procedure) : la structure est fractionnée en un nombre fini de morceau considéré comme des points discrets. Les déplacements et les accélérations de la structure seront ramenés aux déplacements et accélérations de ces points [Clo93].
- *Méthode à déplacement généralisées* (Generalized displacements) : Dans cette approche, on approxime la déformation de la structure par la somme d'une série de déplacements modèles. Ces déplacements modèles sont choisis comme étant des fonctions orthogonales constituant une base d'un espace vectoriel de dimension le nombre de fonctions choisies. On parle alors de déplacement généralisé [Clo93].
- *La méthode des éléments finis* : Cette méthode est une combinaison des deux premières. Elle consiste en une discrétisation du milieu continu en un nombre fini d'éléments. Elle offre un procédé général de discrétisation des problèmes continus décrits par des relations mathématiques. Le milieu continu est divisé en un nombre fini de parties (éléments) dont le comportement est connu à partir d'un nombre fini de paramètre. Ainsi la solution du système complet constitué de l'assemblage de ses éléments suit précisément les mêmes règles que celles applicables aux problèmes discrets standards [Zei79][Clo93].

Afin de formuler les équations dynamiques de la structure, le formalisme de l'équilibre dynamique des forces est utilisée [Clo93]. Formuler l'équilibre dynamique des forces revient à appliquer la seconde loi de Newton : La somme vectorielle des forces agissant sur une masse en mouvement est égale au produit de la valeur de cette masse par son accélération. Les propriétés physiques essentielles d'une structure ou d'un système mécanique soumis à une excitation extérieure ou une charge dynamique sont : sa masse, ses propriétés élastiques ou rigidité et son amortissement.

Pour élaborer le modèle dynamique de cette structure, les hypothèses suivantes sont prises en considération [CLO93][Gue01] :

- La poutre horizontale est supposée rigide et contient toute la masse  $m$  de la structure (procédure de la masse concentrée)
- Les deux axes verticaux sont de masses négligeables, inextensibles verticalement et possédant chacun une constante de rigidité  $k/2$
- La structure possède un amortissement visqueux modéliser par la constante d'amortissement  $c$

L'actionneur est supposé parfait appliquant une force ayant un angle de  $25^\circ$  avec l'horizontale.

Nous nous intéressons dans cette partie à modéliser une structure à plusieurs degrés de liberté (Multi-Degree-Of-Freedom 'MDOF' system), représentée par une construction à plusieurs étages (multi-story building) soumise à l'effet d'une excitation sismique et à des forces de commandes extérieures.

La figure I.4 est une représentation schématique de cette structure.

Chaque étage  $i$  ( $i=1, n$ ) de la structure est caractérisée par :

- ◆ Son déplacement absolu  $x_i$
- ◆ Son déplacement relatif  $x_i$
- ◆ Sa masse  $m_i$

Son amortissement visqueux  $c_i$  et sa constante de rigidité  $k_i$

$u_i$  est la force de commande de l'actionneur installé entre l'étage  $i-1$  et l'étage  $i$ .

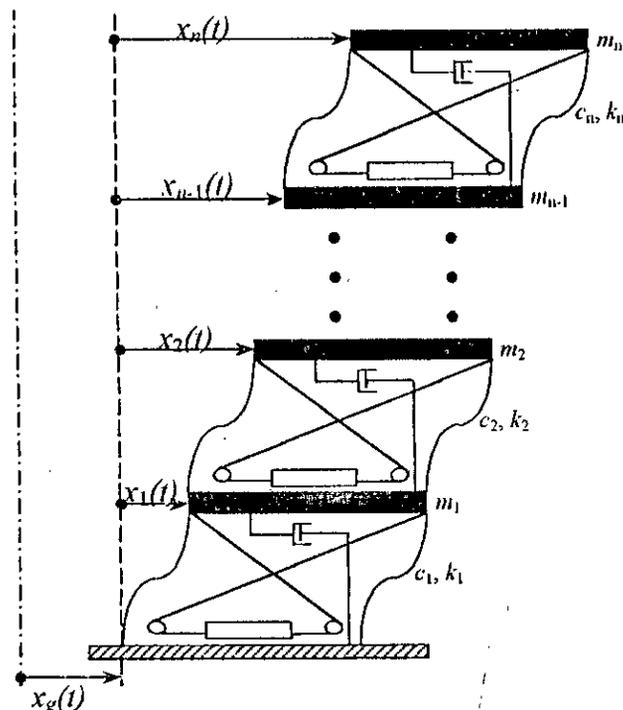


Figure II.4 Schéma d'une construction à plusieurs étages soumise à une excitation

Après quelques manipulations mathématiques, on obtient le système d'équation suivant [Had97] :

$$\begin{cases} m_1 \ddot{x}_1 + (c_1 + c_2) \dot{x}_1 - c_2 \dot{x}_2 + (k_1 + k_2)x_1 - k_2 x_2 = u_1 - u_2 - m_1 \ddot{x}_g \\ m_2 \ddot{x}_2 - c_2 \dot{x}_1 + (c_2 + c_3) \dot{x}_2 - c_3 \dot{x}_3 - k_2 x_1 + (k_2 + k_3)x_2 - k_3 x_3 = u_2 - u_3 - m_2 \ddot{x}_g \\ \vdots \\ m_i \ddot{x}_i - c_i \dot{x}_{i-1} + (c_i + c_{i+1}) \dot{x}_i - c_{i+1} \dot{x}_{i+1} - k_i x_{i-1} + (k_i + k_{i+1})x_i - k_{i+1} x_{i+1} = u_i - u_{i+1} - m_i \ddot{x}_g \\ \vdots \\ m_n \ddot{x}_n - c_n \dot{x}_{n-1} + c_n \dot{x}_n - k_n x_{n-1} + k_n x_n = u_n - m_n \ddot{x}_g \end{cases} \quad (I.21)$$

Sous forme matricielle, l'ensemble des équations (I.21) devient :

$$M\ddot{X}(t) + C\dot{X}(t) + KX(t) = DU(t) - M I_v \ddot{x}_g(t) \quad (I.22)$$

où

$X^T(t) = [x_1(t) \ x_2(t) \ \dots \ x_n(t)]$  vecteur des déplacements,

$U^T(t) = [u_1(t) \ u_2(t) \ \dots \ u_n(t)]$  vecteur des commandes,

$$M = \begin{bmatrix} m_1 & 0 & \dots & 0 \\ 0 & m_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & m_n \end{bmatrix} \text{ la matrice masse de la structure,}$$

$$C = \begin{bmatrix} c_1 + c_2 & -c_2 & 0 & \dots & 0 \\ -c_2 & c_2 + c_3 & -c_3 & \ddots & \vdots \\ 0 & -c_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & c_{n-1} + c_n & -c_n \\ 0 & \dots & 0 & -c_n & c_n \end{bmatrix} \text{ la matrice amortissement de la structure,}$$

$$K = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 & \dots & 0 \\ -k_2 & k_2 + k_3 & -k_3 & \ddots & \vdots \\ 0 & -k_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & k_{n-1} + k_n & -k_n \\ 0 & \dots & 0 & -k_n & k_n \end{bmatrix} \text{ la matrice rigidité de la structure,}$$

$I_v = [1 \ 1 \ \dots \ 1]^T$  vecteur unitaire de dimension n,

$$D = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ 0 & 0 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix} \text{ matrice indiquant l'emplacement des actionneurs dans la}$$

structure,

$\ddot{x}_g(t)$  accélération sismique du sol.

L'équation (I.22) peut être mise sous forme d'état en choisissant le vecteur d'état suivant :  $Z^T(t) = [Z_1(t) \ Z_2(t)] = [X(t) \ \dot{X}(t)]$

L'équation d'état est :

$$\dot{Z}(t) = AZ(t) + BU(t) + E\ddot{x}_g(t) \quad (I.23)$$

avec  $A = \begin{bmatrix} \underline{0} & I \\ -M^{-1}K & -M^{-1}C \end{bmatrix}$  de dimension  $(2n \times 2n)$ ,  $B = \begin{bmatrix} \underline{0} \\ M^{-1}D \end{bmatrix}$  de dimension  $(2n \times n)$ ,

$E = \begin{bmatrix} \underline{0} \\ I_v \end{bmatrix}$  vecteur de dimension  $(2n)$ ,  $\underline{0}$  matrice nulle de dimension  $(n \times n)$ ,

$\underline{0}$  vecteur nul de dimension  $n$ .

Le modèle ainsi obtenu permet de décrire le comportement d'une structure à plusieurs étages permettant ainsi l'analyse et la synthèse de loi de commande dans le cas multivariable.

## I.4 Description du caractère non linéaire des structures en génie civil

L'élaboration des modèles précédents s'est basée sur des modèles rhéologiques simples. Ces modèles permettent de décrire d'une façon linéaire le comportement solide et fluide. Ils représentent une loi de comportement du matériau qui satisfait automatiquement les principes de la thermodynamique.

Nous avons établi le modèle d'une structure dont le comportement reste linéaire durant toute l'amplitude du mouvement, i.e., les caractéristiques force d'élasticité-déplacement et force d'amortissement-vitesse sont des caractéristiques linéaires. En réalité, cette approximation n'est pas toujours valable car les structures présentent un caractère non linéaire [Afr91][Mos99].

Certaines formes des caractéristiques force d'élasticité-déplacement, appelées modèles des non linéarités, sont décrites ci-dessous pour les structures à un seul degré de liberté SDOF.

### I.4.1 Le modèle viscoplastique parfait

Le modèle viscoplastique parfait est le modèle non linéaire le plus fréquemment utilisé en analyse dynamique non linéaire [AFR91]. Ce modèle est illustré dans la figure I.5, et est constitué par les éléments suivants :

- Une masse caractérisée par un facteur de participation  $\rho$
- Un élément élasto-plastique composé d'un ressort élastique de rigidité  $k_{ep}$  en série avec un patin ayant une force de seuil élastique  $f_0 = k_{ep} x_0$ , où  $x_0$  est le déplacement élastique maximal de l'élément élasto-plastique.
- Un amortisseur visqueux  $c$

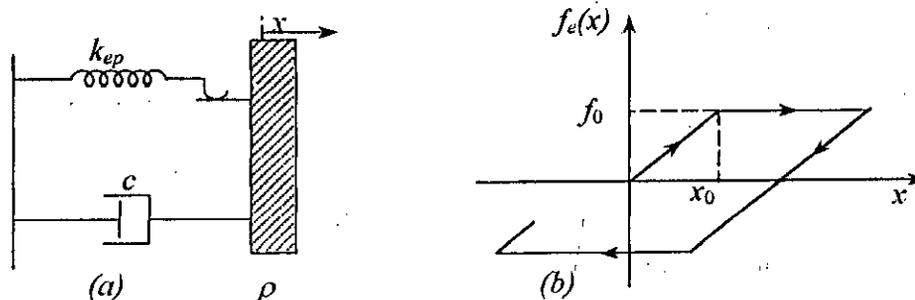


Figure. I.5 Le modèle viscoplastique parfait  
(a) éléments constitutifs (b) diagramme force élastique-déplacement

Ainsi le modèle viscoplastique parfait est caractérisé par un vecteur d'état défini par quatre paramètres ( $k_{ep}$ ,  $x$ ,  $c$ ,  $\rho$ )

### I.4.2. Le modèle viscoplastique Ecroûissable

Ce modèle est une amélioration du modèle précédent par l'introduction d'une variable d'écroûissage. Ce modèle est illustré dans la figure I.6 et est constitué des éléments suivants :

- Une masse caractérisée par un facteur de participation  $\rho$
- Un élément élastique caractérisé par un ressort élastique de raideur  $k_1$
- Un élément élasto-plastique composé d'un ressort élastique de raideur  $k_2$  en série avec un patin ayant une force de seuil élastique  $f_0 = k_2 x_0$ , où  $x_0$  est le déplacement élastique maximal de l'élément élasto-plastique.
- Un amortisseur visqueux  $c$

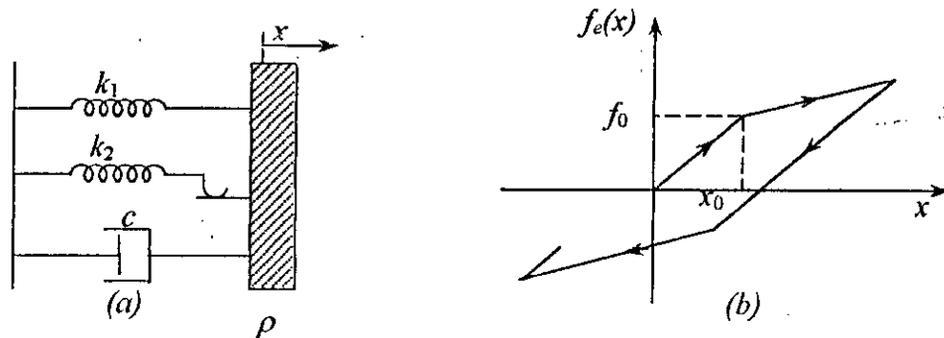


Figure I.6 Le modèle viscoplastique écroûissable  
(a) éléments constitutifs (b) diagramme force élastique-déplacement

Ainsi le modèle viscoplastique écroûissable est caractérisé par un vecteur d'état défini par cinq paramètres ( $k_1$ ,  $k_2$ ,  $x$ ,  $c$ ,  $\rho$ ).

### I.4.3. Le modèle d'endommagement

Ce modèle a été proposé pour donner une représentation plus réaliste du phénomène de dégradation de la rigidité [Hos92][Afr91]. Il est illustré à la figure 1.7 et est constitué des éléments suivants :

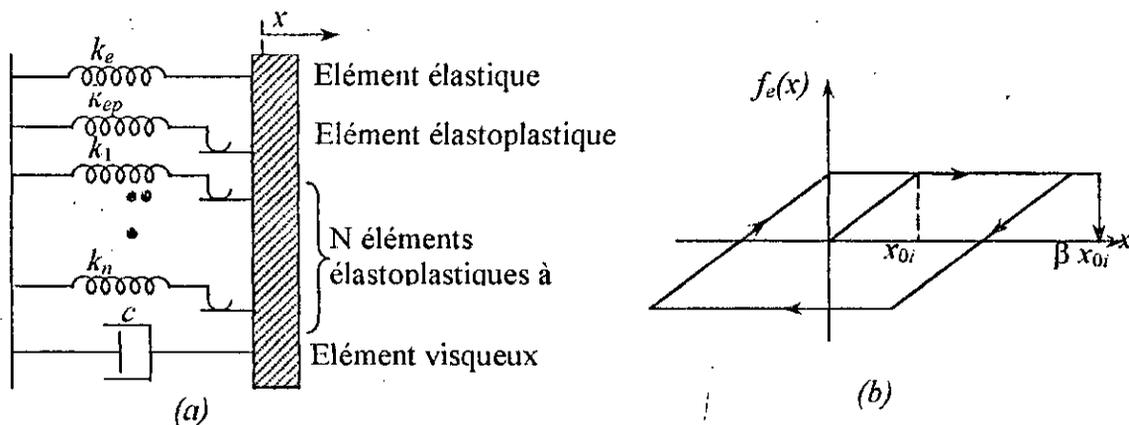


Figure I.7 Le modèle d'endommagement. (a) éléments constitutifs, (b) diagramme force élastique-déplacement d'un élément d'endommagement

- Un élément élastique caractérisé par un ressort élastique de raideur  $k_e$
- Un élément élasto-plastique composé d'un ressort élastique de raideur  $k_{ep}$  en série avec un patin ayant une force de seuil élastique  $f_0 = k_{ep} x_0$ , où  $x_0$  est le déplacement élastique maximal de l'élément élasto-plastique.
- $N$  éléments d'endommagement (éléments élasto-plastiques à rupture brutale). Un élément d'endommagement  $i$  est composé d'un ressort élastique de raideur  $k_i$  en série avec un patin ayant une force de seuil élastique  $f_{0i} = k_i x_{0i}$ , où  $x_{0i}$  est le déplacement élastique maximal de l'élément endommageable  $i$ . On suppose que l'élément  $i$  casse lorsque son déplacement relatif est supérieur à  $\beta x_{0i}$ .
- Un amortisseur visqueux  $c$

D'autres travaux ont été faits dans le but de décrire le comportement non linéaire réel de la structure [Fol96][Ni99][Mei97].

La figure I.8 est une représentation schématique d'une structure non linéaire à un seul degré de liberté, dont l'équation de mouvement est [Fol96]

$$m\ddot{x}(t) + c\dot{x}(t) + \alpha kx(t) + (1 - \alpha)kz(t) = u(t) - m\ddot{x}_g(t) \quad (I.24)$$

où  $m$  masse de la structure,

$c$  constante d'amortissement visqueux,

$k$  constante d'élasticité,

$x(t)$  déplacement relatif de la structure,

$u(t)$  force de commande extérieure,

$\alpha$  taux de rigidité,

$z(t)$  est le déplacement hystérisique défini par l'équation différentielle non linéaire du 1<sup>er</sup> ordre suivante [Fol96]:

$$\dot{z} = \left\{ \dot{x} - \beta |\dot{x}| |z|^{n-1} - \gamma \dot{x} |z|^n \right\} h(z) \quad (I.25)$$

$h(z)$  : fonction de pondération.

où  $\beta$ ,  $\gamma$  et  $n$  sont des paramètres caractéristiques du modèle qui définissent la forme de l'hystérésis.

Le modèle de l'équation (I.25) est connu en littérature sous le nom du modèle original de Bouc-Wen. Grâce à sa souplesse et sa maniabilité mathématique, ce modèle est très utilisé en dynamique des structures non linéaires. Plusieurs extensions à ce modèle ont été proposées et appliquées à divers problèmes d'ingénierie [FOL96][Hun97][Mei97][Mos99].

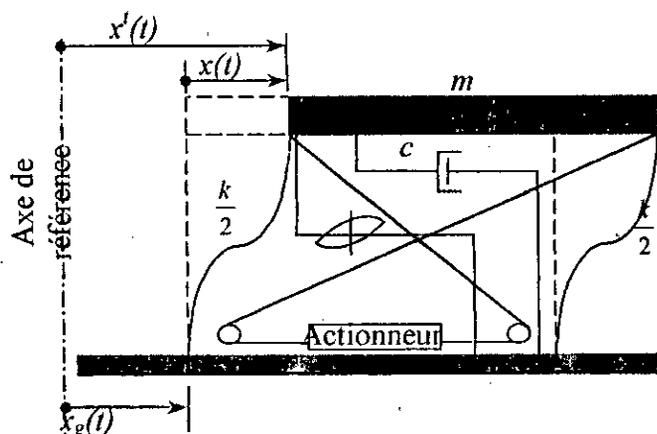


Figure I.8 Schéma représentatif d'une structure non linéaire à un seul degré de liberté

## I.5 Construction du modèle ARMAX de la structure en génie civil

L'influence du séisme sur la structure peut être caractérisée par un processus stochastique. En effet, l'interaction sol-structure est l'une des principales sources de perturbation externe [Lu98]. Comme le modèle de la structure est supposé linéaire, on peut utiliser le principe de superposition et représenter les perturbations par une seule. Le processus et son environnement seront représentés par le modèle ARMAX [Gue01]. Ce modèle représente la forme canonique d'un système discret invariant mono-entrée mono-sortie, avec un retard égal à un multiple de la période d'échantillonnage, assujéti à une perturbation à caractère stationnaire et de densité spectrale rationnelle. Il représente l'ensemble des effets de la commande et des perturbations sur la sortie du système.

Pour les structures à un seul degré de liberté soumises à l'effet d'une excitation sismique, il est facile de déterminer son modèle ARMAX. Mais il sera plus avantageux de tirer parti de la connaissance du modèle de l'excitation sismique, qui est souvent modélisé par un bruit blanc filtré, pour l'inclure dans le calcul du modèle ARMAX. Ce qui peut améliorer les performances du système en boucle fermée. L'idée d'introduire le modèle du séisme sera illustrée dans ce qui suit (annexe3).

Considérons l'équation suivante :

$$\ddot{x}(t) + 2\xi\omega_0\dot{x}(t) + \omega_0^2x(t) = \frac{1}{m}u(t) - \ddot{x}_g(t) \quad (I.26)$$

où  $\omega_0 = \sqrt{\frac{k}{m}}$  fréquence naturelle de la structure.

$\xi = \frac{c}{2m\omega_0}$  taux d'amortissement.

Par application de la transformée de Laplace à cette équation, on obtient :

$$X(s) = \frac{\frac{1}{m}}{s^2 + 2\xi\omega_0s + \omega_0^2}U(s) - \frac{1}{s^2 + 2\xi\omega_0s + \omega_0^2}\ddot{X}_g(s) \quad (I.27)$$

où  $X(s)$ ,  $\ddot{X}_g(s)$  et  $U(s)$  sont respectivement les transformées de Laplace de  $x(t)$ ,  $\ddot{x}_g(t)$  et  $u(t)$ .

La figure I.9 est un schéma bloc du modèle de la structure.

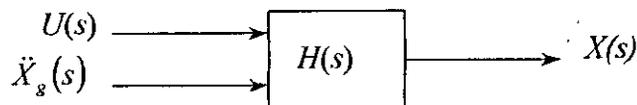


Figure I.9 Schéma bloc du modèle de la structure

Selon le modèle de l'excitation sismique, différents modèles ARMAX peuvent être obtenus.

### I.5.1 Modèle de la structure sans introduction du modèle du séisme

Le modèle de l'excitation sismique n'est pas connu ou n'est pas pris en considération. Le modèle ARMAX de la structure est obtenu par discrétisation de l'équation (I.27). On peut réécrire cette équation sous la forme compacte suivante :

$$X(s) = \frac{H_{1N}(s)}{H_{1D}(s)}U(s) + \frac{H_{2N}(s)}{H_{2D}(s)}\ddot{X}_g(s) \quad (I.28)$$

$$\text{où } \begin{cases} H_{1D}(s) = H_{2D}(s) = s^2 + 2\xi\omega_0 s + \omega_0^2 \\ H_{1N}(s) = \frac{1}{m} \\ H_{2N}(s) = -1 \end{cases}$$

La figure I.10 est une représentation schématique de l'équation (I.28),

$$\text{où } H_1(s) = \frac{H_{1N}(s)}{H_{1D}(s)} \text{ et } H_2(s) = \frac{H_{2N}(s)}{H_{2D}(s)}$$

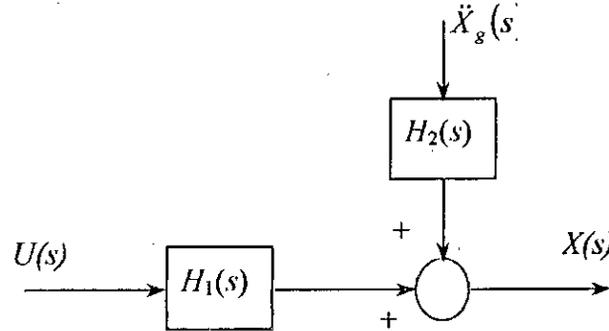


Figure I.10. Modèle de la structure sans prendre en considération le modèle du séisme

La discrétisation de l'équation (I.28) peut être effectuée analytiquement (utilisation de la transformée en Z) ou en utilisant des programmes d'ordinateur prédestinés à accomplir cette tâche (MATLAB 5.2).

Par discrétisation analytique de l'équation (I.28), le modèle ARMAX est donné par [Gue01][Lan93] :

$$x(t) = \frac{B(q^{-1})}{A(q^{-1})}u(t) + \frac{C(q^{-1})}{A(q^{-1})}\ddot{x}_g(t) \quad (I.29)$$

$$\text{où } A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2},$$

$$B(q^{-1}) = b_1q^{-1} + b_2q^{-2},$$

$$C(q^{-1}) = c_1q^{-1} + c_2q^{-2},$$

$$\text{avec } \begin{cases} a_1 = -2\alpha\beta \\ a_2 = \alpha^2 \end{cases}, \begin{cases} b_1 = \frac{1}{m\omega_0} \left[ 1 - \alpha \left( \beta + \frac{\xi\omega_0}{\omega} \gamma \right) \right] \\ b_2 = \frac{1}{m\omega_0} \left[ \alpha^2 + \alpha \left( \frac{\xi\omega_0}{\omega} \gamma - \beta \right) \right] \end{cases}, \begin{cases} c_1 = -\frac{1}{\omega_0} \left[ 1 - \alpha \left( \beta + \frac{\xi\omega_0}{\omega} \gamma \right) \right] \\ c_2 = -\frac{1}{\omega_0} \left[ \alpha^2 + \alpha \left( \frac{\xi\omega_0}{\omega} \gamma - \beta \right) \right] \end{cases}$$

$\omega = \omega_0 \sqrt{1 - \xi^2}$ ,  $\alpha = \exp(-\xi\omega_0 T_e)$ ,  $\beta = \cos(\omega T_e)$ ,  $\gamma = \sin(\omega T_e)$ ,  $T_e$  est la période d'échantillonnage.

### I.5.2 Modèle de la structure avec introduction du modèle de Kanai-Tajimi

L'accélération sismique du sol est décrite par le modèle de Kanai-Tajimi [Lin87] :

$$\ddot{X}_g(s) = G_1(s)E(s) \quad (I.30)$$

$$\text{où } G_1(s) = \frac{G_{1N}(s)}{G_{1D}(s)} = \frac{2\xi_g\omega_g s + \omega_g^2}{s^2 + 2\xi_g\omega_g s + \omega_g^2}$$

$E(s)$  transformée de Laplace du bruit blanc,  $\xi_g$  et  $\omega_g$  fréquence de résonance et amortissement propre du sol.

Par substitution de l'équation (I.30) dans l'équation (I.27), on obtient :

$$X(s) = \frac{1/m}{s^2 + 2\xi\omega_0 s + \omega_0^2} U(s) - \left( \frac{1}{s^2 + 2\xi\omega_0 s + \omega_0^2} \right) \left( \frac{2\xi_g\omega_g s + \omega_g^2}{s^2 + 2\xi_g\omega_g s + \omega_g^2} \right) E(s) \quad (I.31)$$

Ou encore sous la forme compacte suivante :

$$X(s) = \frac{H_{1N}(s)}{H_{1D}(s)} U(s) + \frac{H_{2N}(s)G_{1N}(s)}{H_{1D}(s)G_{1D}(s)} E(s) \quad (I.32)$$

L'équation (I.32) ressemble à un modèle de Box-Jenkins (modèle ARMAX avec pôles différents annexe 2) en continu. Pour obtenir le modèle ARMAX ayant les mêmes pôles, on doit ramener le second membre de l'équation (I.32) au même dénominateur, on obtient :

$$X(s) = \frac{F_{1N}(s)}{F_D(s)} U(s) + \frac{F_{2N}(s)}{F_D(s)} E(s) \quad (I.33)$$

$$\text{où } \begin{cases} F_D(s) = H_{1D}(s)G_{1D}(s) = (s^2 + 2\xi\omega_0 s + \omega_0^2)(s^2 + 2\xi_g\omega_g s + \omega_g^2) \\ F_{1N}(s) = H_{1N}(s)G_{1D}(s) = \frac{1}{m}(s^2 + 2\xi_g\omega_g s + \omega_g^2) \\ F_{2N}(s) = H_{2N}(s)G_{1N}(s) = -(2\xi_g\omega_g s + \omega_g^2) \end{cases}$$

Le modèle ARMAX discret de la structure est obtenu par discrétisation de l'équation (I.33). La figure I.11 est une représentation des étapes permettant d'introduire le modèle du séisme dans le calcul du modèle ARMAX de la structure.

### I.5.3 Modèle de la structure avec introduction du modèle de Clough-Penzien

L'accélération sismique du sol est décrite par le modèle de Clough-Penzien [Lin89] :

$$\ddot{X}_g(s) = G_2(s)E(s) \quad (I.34)$$

$$\text{où } G_2(s) = \frac{G_{2N}(s)}{G_{2D}(s)} = \left( \frac{2\xi_g\omega_g s + \omega_g^2}{s^2 + 2\xi_g\omega_g s + \omega_g^2} \right) \left( \frac{s^2}{s^2 + 2\xi_c\omega_c s + \omega_c^2} \right)$$

En suivant les mêmes étapes que précédemment, on obtient :

$$X(s) = \frac{F_{1N}(s)}{F_D(s)} U(s) + \frac{F_{2N}(s)}{F_D(s)} E(s) \quad (I.35)$$

$$\text{avec } \begin{cases} F_D(s) = H_{1D}(s)G_{2D}(s) = (s^2 + 2\xi\omega_0 s + \omega_0^2)(s^2 + 2\xi_g\omega_g s + \omega_g^2)(s^2 + 2\xi_c\omega_c s + \omega_c^2) \\ F_{1N}(s) = H_{1N}(s)G_{2D}(s) = \frac{1}{m}(s^2 + 2\xi_g\omega_g s + \omega_g^2)(s^2 + 2\xi_c\omega_c s + \omega_c^2) \\ F_{2N}(s) = H_{2N}(s)G_{2N}(s) = -(2\xi_g\omega_g s + \omega_g^2)s^2 \end{cases}$$

Par discrétisation de l'équation (I.35), on obtiendra le modèle ARMAX de la structure avec modèle du séisme de Clough-Penzien.

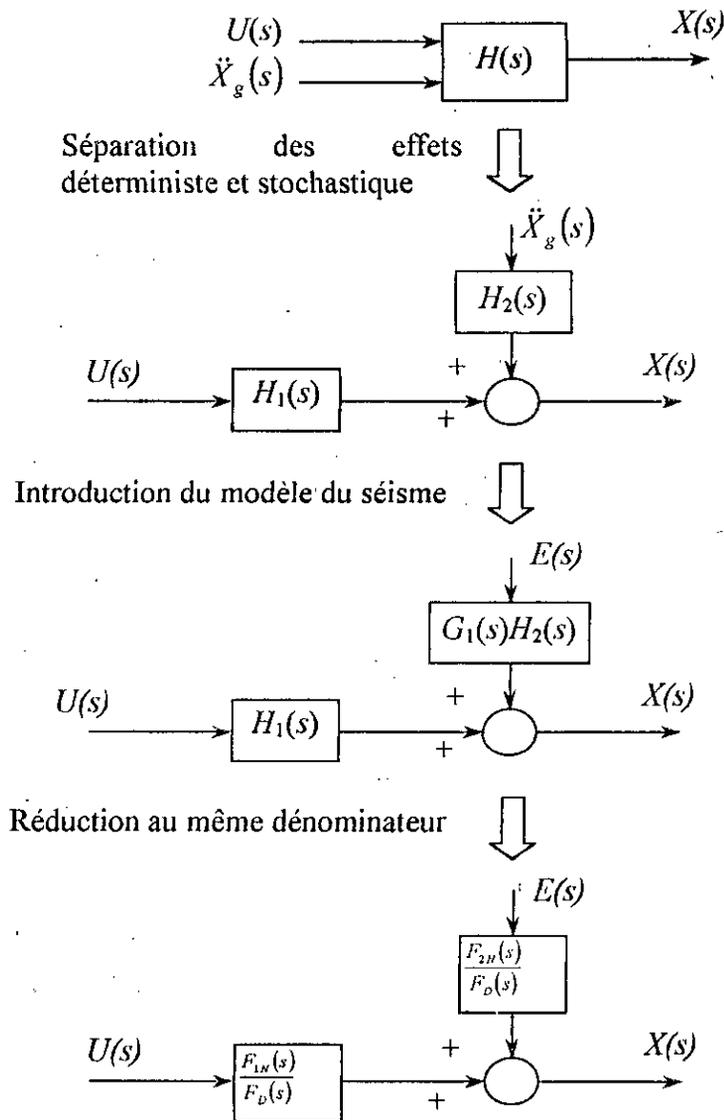


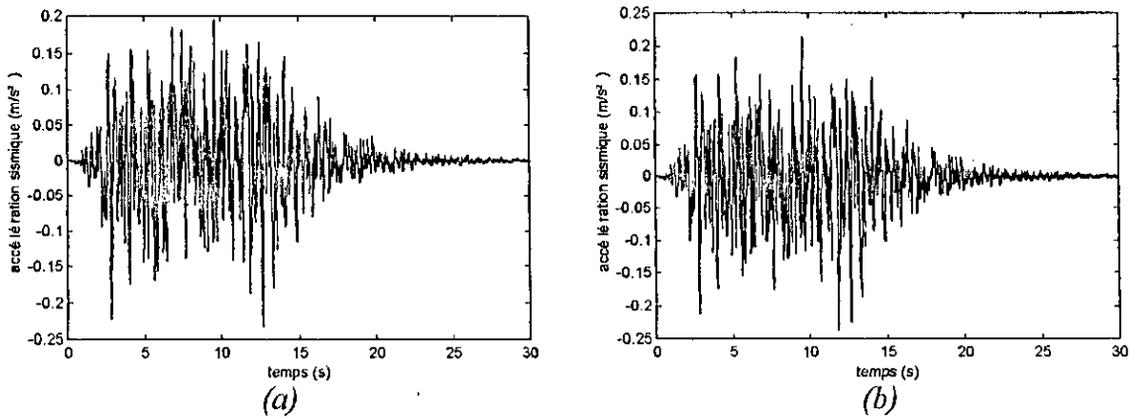
Figure I.11 Introduction du modèle du séisme dans le calcul du modèle ARMAX de la structure

## I.6 Résultats de simulation

Dans cette partie, nous nous intéressons à choisir une structure à un seul degré de liberté pour laquelle on détermine les différents modèles ARMAX pour différentes excitations. On simule le comportement de la structure pour les excitations choisies. En fin une structure à trois degrés de liberté excitée par les bruits de Kanai Tajimi et Clough Penzien, est choisie.

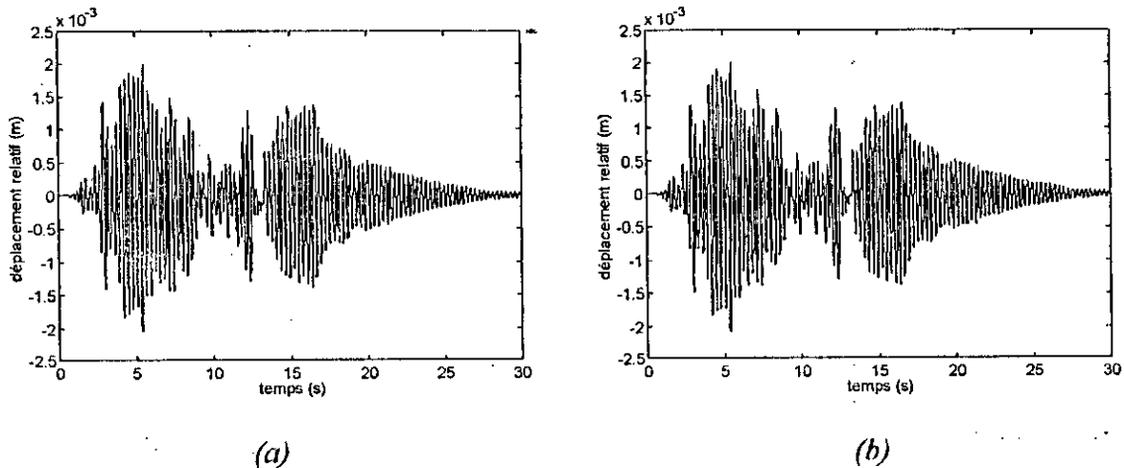
Les paramètres de la structure sont :  $m=2921$  Kg,  $k=1389$  kN/m,  $\xi=0,0124$ . Les paramètres du séisme sont :  $\xi_g=0,65$ ,  $\omega_g=19$  rad/s,  $\xi_c=0,6$ ,  $\omega_c=2$  rad/s [Gue01][Smi94].

Les accélérations sismiques de Kanai-Tajimi et Clough-Penzien ont été simulées et sont présentées à la figure I.12. ( $t_1=3$  s,  $t_2=13$  s,  $a=0.26$ ,  $\xi_g=0.65$ ,  $\omega_g=19$  rad/s,  $\xi_c=0.6$ ,  $\omega_c=2$  rad/s,  $S_0=0.8 \cdot 10^{-2} \text{ m}^2/\text{s}^2$  [YAN87]) (voir Annexe 4).



**Figure I.12** Accélérations sismiques simulées  
(a) modèle de Kanai-Tajimi (b) modèle de Clough-Penzien

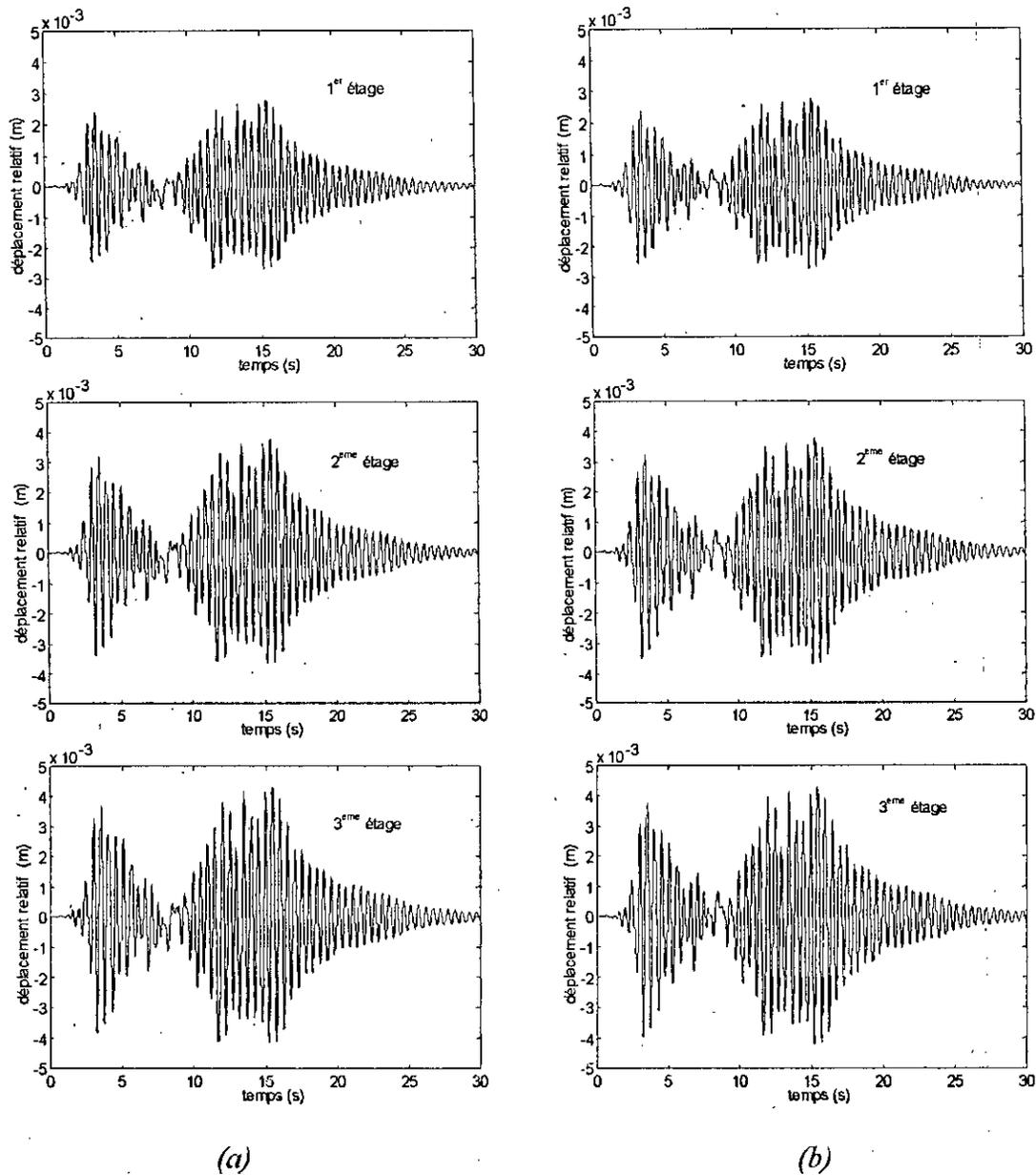
La figure I.13 illustre la réponse d'une structure à un seul degré de liberté au séisme de Kanai-Tajimi et de Clough-Penzien (en boucle ouverte).



**Figure I.13.** Réponse en boucle ouverte d'une structure à un seul degré de liberté à l'excitation sismique de  
(a) Kanai-Tajimi (b) Clough-Penzien

De même, la figure I.14 illustre la réponse en boucle ouverte d'une structure à trois étages (trois degrés de liberté) aux mêmes accélérations sismiques. Les paramètres de cette structure sont [CHA96] :

$$\begin{aligned}
 m_1 &= 2100 \text{ Kg}, & k_1 &= 1262450 \text{ N/m}, & c_1 &= 3675 \text{ N.s/m} \\
 m_2 &= 2100 \text{ Kg}, & k_2 &= 2607500 \text{ N/m}, & c_2 &= 10500 \text{ N.s/m} \\
 m_3 &= 2100 \text{ Kg}, & k_3 &= 2607500 \text{ N/m}, & c_3 &= 10500 \text{ N.s/m}
 \end{aligned}$$



(a) (b)  
**Figure I.14** Réponse en boucle ouverte d'une structure à trois degrés de liberté  
à l'excitation sismique de  
(a) Kanai-Tajimi (b) Clough-Penzien

Dans le cas d'une structure articulée, la trajectoire imposée pour notre robot est spatiale de telle sorte à activer toutes les variables (fenêtre de Viviani). La fenêtre de Viviani est l'intersection d'un cylindre creux d'axe parallèle à  $z$ , de centre  $(0, r/2, 0)$ , et une sphère de centre  $o(0, 0, 0)$  et de rayon  $R$ , dans le repère  $R(x, y, z)$  (figure I.15).

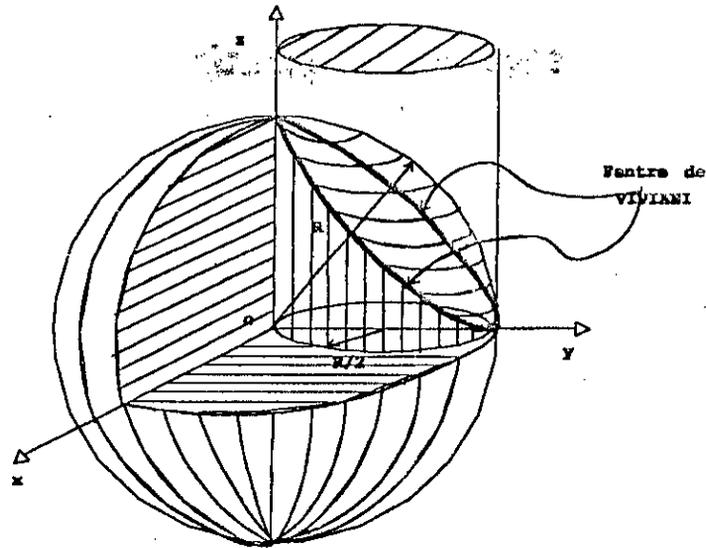


Figure I.15 Fenêtre de VIVIANI

Pour que l'élément terminal décrit la trajectoire de la fenêtre de Viviani, les consignes pour chaque variable, pour accomplir cette tâche sont illustrées dans la figure I.18 (pour  $\theta = 0 ; z = R \rho = 0$ ). Le robot se met d'abord à l'extrémité supérieure de la fenêtre. Il fait le trajet le long de la fenêtre et il revient

Les caractéristiques du robot, telles que les masses et les dimensions, sont [Sto87]:

$$m_1 = 20\text{kg}; m_3 = 10\text{kg}; l_2 = 0.75\text{m}; g = 9.81\text{ms}^{-2}; k_1 = 100\text{N/V}; k_2 = 10\text{Nm/V}; k_3 = 10\text{N/V}; f_1 = 30\text{Ns}$$

On simule l'algorithme de NE, pour des consignes sinusoidales et un couple et une force nulles ( $f_4 = \eta_4 = 0$ ). La figure I.19 nous montre les forces généralisées (commandes) nécessaires pour accomplir la tâche voulue (trajectoire sinusoidale).

Nous avons procédé par simulation à un couplage des deux modèles de la structure articulée (figures I.16 et I.17)

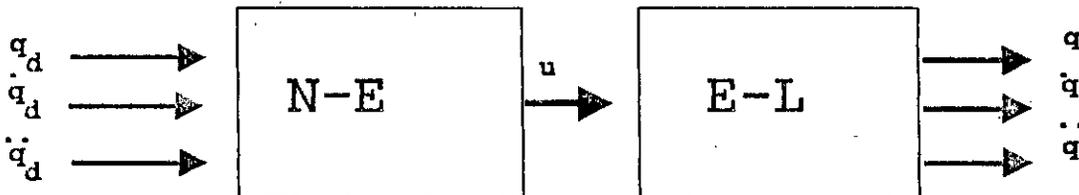


Figure I.16 Couplage des deux modèles NE-EL

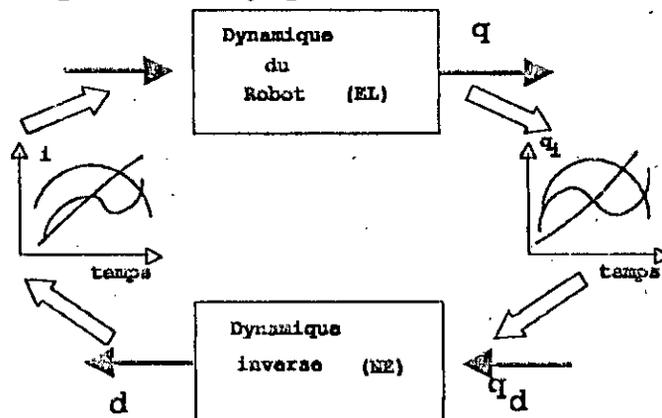
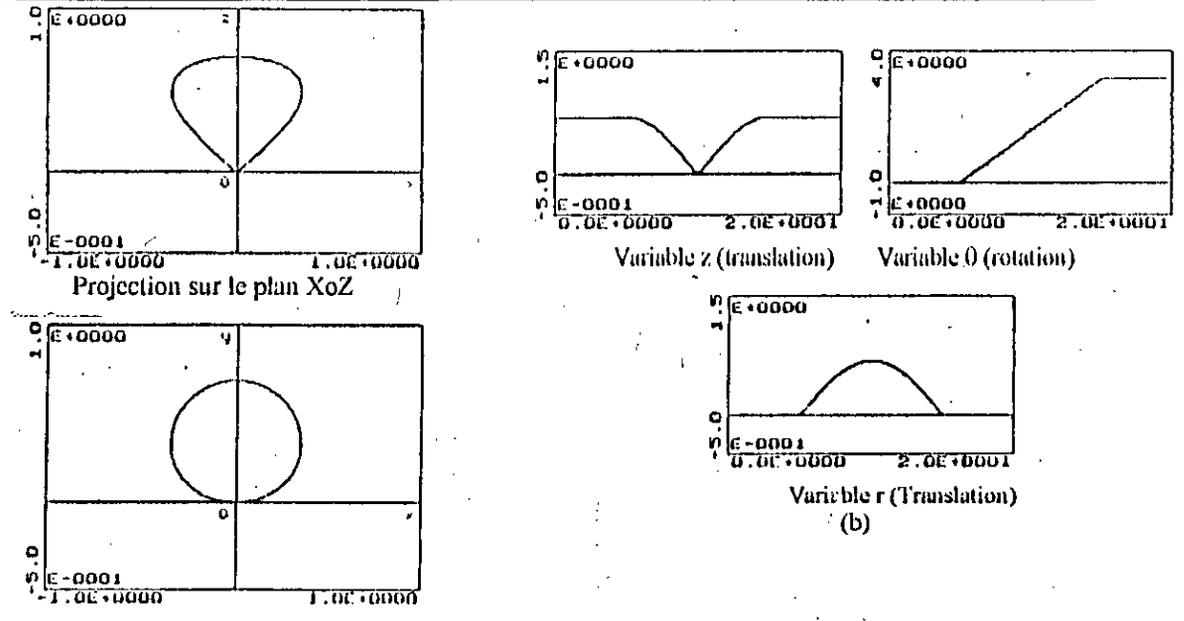


Figure I.17 Schéma explicite du couplage

La figure I.20 illustre les résultats de simulation dans le cas du couplage.



Projection sur le plan XoY  
 Figure 1.18 Représentation temporelle et spatiale de la fenêtre de VIVIANI

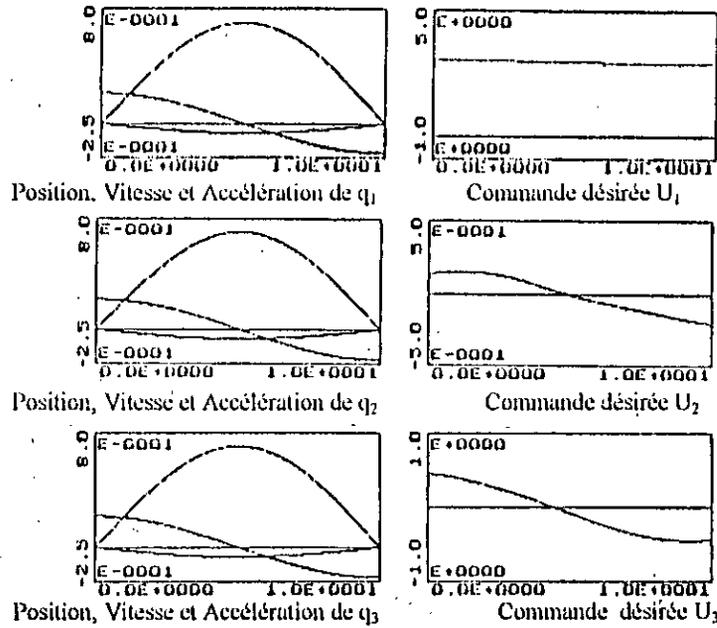


Figure 1.19. Trajectoires désirées et commande calculées par l'algorithme de Newton-Euler

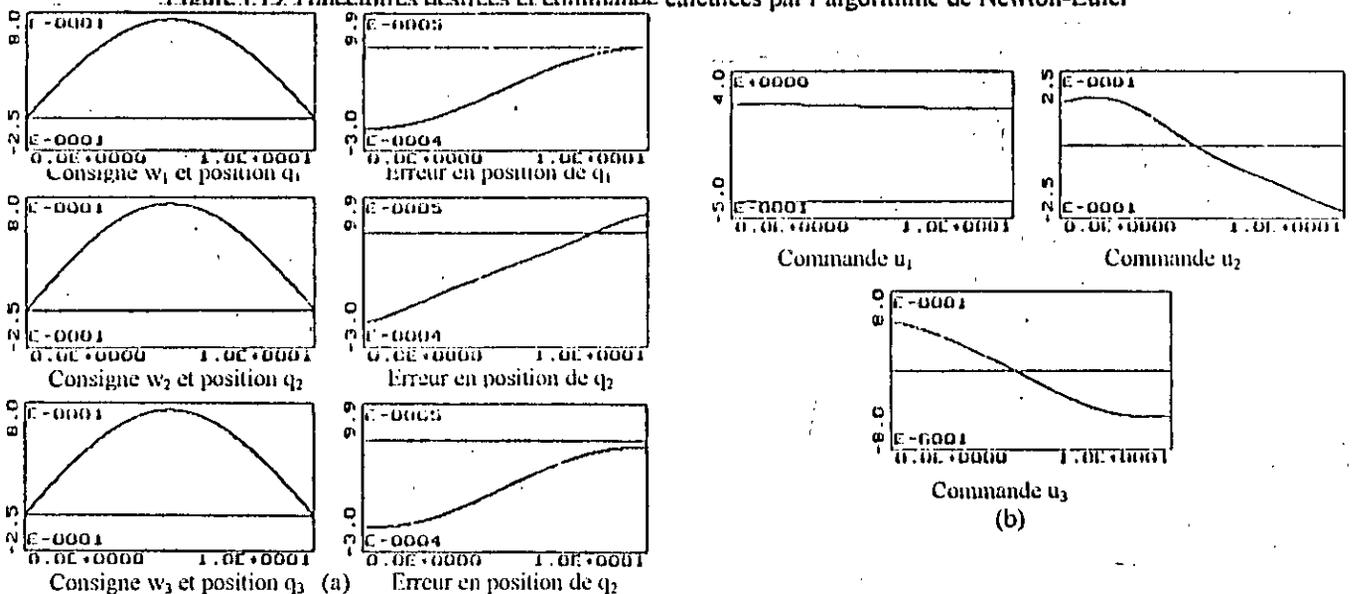


Figure 1.20 Réponses du robot pour le couplage EL et NE (Référence sinusoïdale)

## I.7 Conclusion

Dans ce chapitre, nous avons présenté le modèle dynamique des structures en génie civil. L'élaboration du modèle s'est basée sur la procédure de la masse concentrée en utilisant le formalisme de l'équilibre dynamique des forces. Le modèle de connaissance est établi en se basant sur des modèles rhéologiques simples.

Afin d'élargir la validité des modèles développés, une description du caractère non linéaire a été développée. Ces non linéarités permettent de traduire d'une façon plus réaliste le comportement fluide et solide réel de la structure. Plusieurs modèles ont été établis. Le modèle d'endommagement est le plus complexe des non linéarités existantes lorsque sa description est décrite d'une façon continue. La discrétisation en  $N$  éléments de son comportement non linéaire est une approximation plus ou moins grossière de la réalité. Néanmoins la résolution de l'équation différentielle non linéaire dans le cas continu est compliquée nécessitant des outils mathématiques et théoriques complexes.

Pour décrire le comportement temporel discret de la structure, le modèle ARMAX a été établi. Ce modèle est une discrétisation directe des équations différentielles du modèle de connaissance. Ainsi ce modèle peut être considéré comme un modèle de représentation parfait issu d'une discrétisation exacte dépendant seulement de la période d'échantillonnage.

L'introduction de l'excitation sismique dans le modèle de connaissance linéaire, nous a permis la construction d'un modèle ARMAX personnalisé pour l'excitation considéré. Ce modèle offre la possibilité de développer une loi de commande stochastique prenant en considération l'aspect aléatoire de l'excitation. Par ailleurs, nous avons choisi des excitations sismiques universelles permettant de décrire un comportement sismique parfait. Les excitations de Clough Penzien et Kanai tajimi offrent l'avantage de développer une loi de commande qui permet une généralisation pour d'autres types d'excitations.

Pour les structures articulées, la modélisation cinématique nous permet la détermination des relations directes et inverses entre les coordonnées généralisées et les coordonnées cartésiennes. Le passage des coordonnées cartésiennes de l'élément terminal aux coordonnées généralisées des différentes articulations, présente des singularités. Cependant, ce type de problème peut être évité par l'utilisation de différent modèle cinématique. En effet, le choix de la trajectoire généralisée ou cartésienne, est une étape importante pour la commande des robots. Si des discontinuités dans la vitesse ou l'accélération existent, cela nécessite une interpolation polynomiale sur la trajectoire généralisée.

Le modèle de connaissance dynamique, établi par le formalisme de Lagrange-Euler, nous permet la détermination de la trajectoire du robot connaissant les forces généralisées, par simulation.

L'algorithme de Newton-Euler permet une commande en boucle ouverte, malheureusement très sensible aux perturbations paramétriques et de sorties. Pour compenser ces perturbations nous utilisons une commande additive, élaborée par un algorithme de commande adaptative.

1. The first part of the document is a list of names and addresses of the members of the committee.

2. The second part of the document is a list of names and addresses of the members of the committee.

3. The third part of the document is a list of names and addresses of the members of the committee.

4. The fourth part of the document is a list of names and addresses of the members of the committee.

5. The fifth part of the document is a list of names and addresses of the members of the committee.

6. The sixth part of the document is a list of names and addresses of the members of the committee.

---

## Chapitre II

### *Étude de synthèse sur les Réseaux de Neurones*

---

#### Introduction

Le gain en temps de calcul est obtenu par une simultanéité des opérations alors que l'approche classique consistait à diminuer le temps de traitement des opérations. Ce domaine s'est considérablement développé dans les années quatre-vingt avec un grand effort de recherche de la communauté scientifique [Lac94]. Le chercheur doit alors décomposer son problème en plusieurs tâches concurrentes. La structure de cette décomposition doit permettre une certaine indépendance entre les tâches. L'utilisateur doit cependant gérer les contraintes temporelles (dynamique), la synchronisation et l'accès aux données. Une autre approche consiste à inventer de nouveaux algorithmes. Dans ce cas, l'objectif est de développer des algorithmes mieux adaptés aux nouvelles architectures. Par exemple, il faut un très grand nombre de processeurs et réduire le nombre de dépendances entre les tâches. Dans ce domaine l'état de l'art des techniques connexionnistes fait apparaître des applications potentielles des réseaux de neurones aux problèmes soulevés par l'identification et la théorie du contrôle des systèmes non linéaires [Del91][Hun92].

Aujourd'hui plusieurs types de motivation animent les chercheurs travaillant dans le domaine des réseaux connexionnistes. Un premier groupe, qui rassemble principalement des physiciens et des informaticiens, recherche au moyen de ces réseaux de nouvelles architectures matérielles informatiques, caractérisées par une très grande rapidité dû au parallélisme et une meilleure résistance aux pannes. Un second groupe comprenant notamment des biologistes et des psychologues puise son inspiration dans les systèmes vivants et cherche au moyen des réseaux connexionnistes à construire des modèles capables de produire leurs observations. Enfin, un dernier groupe utilise des réseaux connexionnistes comme des outils de reconnaissance de forme ou plus généralement de traitement de l'information [Lac94][Boc91].

Dans ce chapitre, on se situe dans le dernier groupe et on privilégie la démarche de l'ingénieur face à la résolution d'un problème. On étudie les réseaux connexionnistes comme des techniques de classifications ou d'approximation de fonction.

Plusieurs thèmes de recherches sont abordés à l'aide des réseaux connexionnistes. Parmi ceux-ci, on trouve la mémorisation, représentation et apprentissage [Koh80][Yed98][Mic99]. On s'intéresse surtout aux théories de l'apprentissage mais on évoquera ici les autres problématiques.

En identification, le choix de l'architecture du réseau peuvent traduire le comportement réel du système en prenant en considération l'aspect non linéaire. Plusieurs modèles ont été proposés. Par contre, le choix de la structure de commande est directement liée à la nature du système et la stratégie de commande utilisée [Nar90].

Aussi la détermination et l'adaptation des paramètres de commande sont réalisées en utilisant des algorithmes d'apprentissage approprié permettant une implémentation en temps réel ou en temps différé [Rum86][Sol61][Bob89]. Ces nouvelles architectures permettant de réaliser des performances très élevées. Mais la mise en œuvre de ces nouvelles structures a dû attendre l'évolution de la technologie de réalisation des composants électroniques et leurs méthodes d'intégration « very large scale integration (VLSI)».

Dans les premiers temps de cette évolution, on a surtout exploité les machines à architectures séquentielles (dits de Van Newman) [Mul90]. Ces machines ne peuvent traduire le comportement réel d'un réseau de neurones. Les simulations effectuées dans les ordinateurs traditionnels possèdent un aspect séquentiel. Ceux-ci ne peuvent pas prendre en considération le temps réel nécessaire à l'exécution de la tâche confiée au réseau. Par contre des déductions à aspect mathématiques et théoriques peuvent être faites (stabilité et convergence). Ainsi la détermination des temps de calcul nécessite en premier lieu de connaître l'architecture du réseau utilisé. En second lieu, le matériel nécessaire à son intégration permet de calculer le temps matériel (calcul) et le temps logiciel (gestion) nécessaire aux traitements de l'information [Hir90].

Ce chapitre est consacré à une synthèse sur les réseaux connexionnistes. Nous distinguons plusieurs modes d'apprentissage : non supervisé, supervisé et par renforcement. Nous présentons ensuite les différentes architectures de connexions (statique, dynamique et évolutive). Une classification générale dans différents stratégies de commande utilisant les réseaux de neurones est donnée. Nous terminons ce chapitre par une conclusion.

## II.1 Le neurone biologique

Le neurone est une cellule constituée principalement de trois parties (figure II.1) qui, au regard du transfert d'informations, ont un rôle fonctionnel bien défini. Ce sont les dendrites, le Soma et l'Axone [Kun93][Kos92].

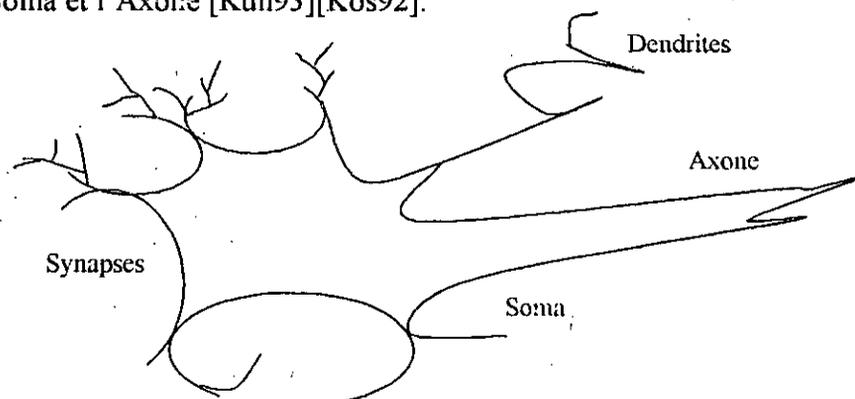


Figure II.1 Le neurone biologique

a. La Dendrite :

Cette partie qui présente une sorte de ramifications appelées *arbre dendritique*, collecte les signaux venant d'autres cellules ou de l'extérieur. La réception des signaux en provenance des autres neurones, se fait par des points de contacts appelés *Synapses*. Certaines cellules peuvent compter jusqu'à 100 000 synapses [Her94]. Ces informations, qui ne sont rien d'autres que des impulsions électriques, sont par la suite acheminées vers le corps cellulaire ou le Soma.

b. Le Soma :

L'arbre dendritique fait converger vers le Soma des influx nerveux venus d'une très large étendue autour du neurone. Le Soma, outre son rôle concernant le métabolisme de la cellule, recueille et concentre les informations reçues et en fait une sommation dite «spatio-temporelle». Si le potentiel somatique dépasse un certain seuil, il y a émission d'un potentiel d'action, appelé « Spike », qui correspond à une oscillation électrique très brève (1ms) [Mil91].

c. L'Axone :

L'information traitée est transmise vers l'extérieur le long de l'axone, et répartie sur les synapses des neurones cibles grâce à l'arborescence terminale que possède l'axone.

## II.2. Le neurone formel

Le premier modèle fut réalisé par McCulloch et Pitts en 1943. Dans ce modèle « un neurone formel fait la somme pondérée des potentiels d'actions qui lui parviennent ( chacun de ces potentiels est une valeur numérique qui représente l'état du neurone qui l'a émit ), puis s'active suivant la valeur de cette sommation pondérée. Si cette somme dépasse un certain seuil, le neurone est activé et transmet une réponse (sous forme de potentiel d'action) dont la valeur est celle de son activation. Si le neurone n'est pas activé, il transmet rien d'autre qu'un produit scalaire entre un vecteur d'entrée et un vecteur poids suivi d'une fonction seuil (figure II.2) [Yed98].

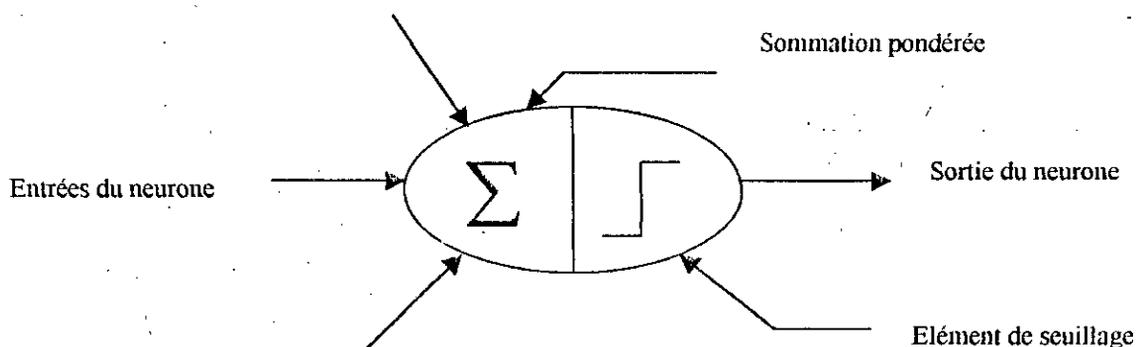


Figure II.2 Le modèle de Mc Culloch et Pitts

## II.3 Modèle mathématique général du neurone

Au début des recherches scientifiques, le souci était de se rapprocher le plus possible du modèle biologique dans l'espoir de pouvoir synthétiser un jour, un réseau qui peut ressembler au cerveau humain dans son fonctionnement. Plus tard, les chercheurs se sont plutôt occupés de l'efficacité, des concepts théoriques et des outils mathématiques nécessaires permettant de résoudre les problèmes rencontrés lors de leurs implémentations. Ceci a nécessité des modifications structurelles sur ce premier modèle. Le réseau classique ne considère qu'une certaine classe de réseaux de neurones. En 1987, Hecht-Nelson introduit le terme de « Mapping Neural Network ». Un MNN est défini comme suit :

Un « mapping neural Network » est un réseau de fonction effectuant une relation  $\phi : I^n \rightarrow R^m$ , en se basant sur l'interconnexion de neurones, considérés comme éléments de base effectuant une relation non linéaire d'une façon parallèle et distribuée, ou  $I^n$  est un hypercube de  $n$  dimensions. Cette définition rassemble une large classe de réseaux de neurones [Kos92][Kun93]. D'après cette définition un réseau de neurones est caractérisé par le neurone comme élément de traitement de base, et de leur architecture d'interconnexion. Ainsi, la figure II.3 présente le modèle mathématique général du neurone :

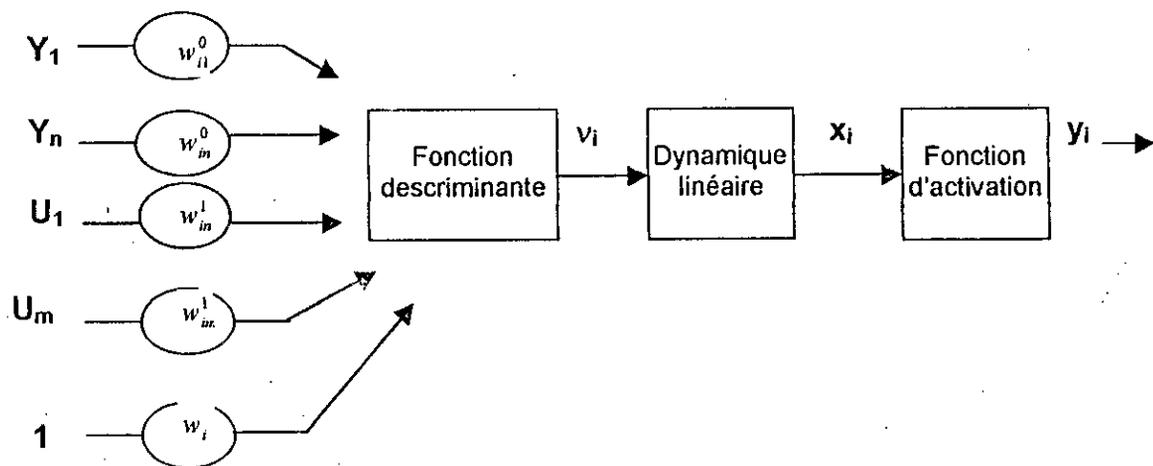


Figure II.3 Modèle mathématique général d'un neurone formel

Ce neurone est l'élément de base d'un réseau de  $n$  neurones à  $n$  sortie  $y_i$  et  $m$  entrée  $U_j$ .  $v_i$  est la sortie de la fonction discriminante.  $w_{ik}^0$  et  $w_{ik}^1$  : poids synaptiques.

Le neurone est considéré alors comme une fonctionnelle caractérisée par :

- Une fonction discriminante.
- Un système dynamique linéaire SISO.
- Une fonction d'activation non linéaire.

### II.3.1 fonction discriminante

Cette fonction, appelée aussi fonction de base (Basis function) définit l'activité du neurone. Afin d'améliorer les performances des réseaux, d'autres formes de fonctions non linéaires ont été élaborées ces dernières années [Yed98], offrant ainsi aux réseaux de meilleures possibilités dans certains domaines et un large champ d'application. Ces techniques nécessitent néanmoins des calculs plus compliqués pour leur élaboration. Nous citons ci-dessous les fonctions de bases existantes.

**a. La fonction de base linéaire LBF (Linear Basis Function) :**

Cette fonction est utilisée dans le cas du perceptron et pour les réseaux multicouches LBF [Rum86]. Elle représente un sommateur pondéré des entrées vers le neurone. Ce sommateur est décrit par :

$$v_i(t) = \sum_{j=1}^n w_{ij}^o y_j(t) + \sum_{k=1}^m w_{ik}^l U_k(t) + w_i \quad (II.1)$$

L'équation II.1, sous forme matricielle s'écrit :

$$\phi_i(Z, w_i) = Z^T w_i \quad i = 1, n \quad (II.2)$$

Avec

$$w_i^T = [w_{i1}^o \dots w_{in}^o \quad w_{i1}^l \dots w_{im}^l \quad w_i]$$

$$Z^T = [y_1 \dots y_n \quad U_1 \dots U_m \quad 1]$$

Où W représente la matrice des poids et Z les entrées qui viennent de l'extérieur ou de la part des autres neurones vers le neurone en question. En plus de son efficacité, cette fonction est la plus simple à implémenter.

**b. La fonction de base radiale (Radial Basis Function) :**

La forme de cette fonction est inspirée des réseaux de neurones utilisés pour la classification [Kun93]. Elle est définie par :

$$\phi_i(Z, w_i) = \frac{\|Z - w_i\|^2}{2} \quad (II.3)$$

Dans les réseaux utilisant cette fonction de base, les poids W sont présentés comme étant les centroides de chaque classe de l'espace d'entrée. En effet cette fonction sert à effectuer un échantillonnage de l'espace des entrées, à chaque groupe de poids synaptiques représente une concentration de données.

**c. La fonction de base Elliptique (Eleptique Basis Function) :**

La forme de ce discriminant est une généralisation de la fonction de base radiale, elle est définie par :

$$\phi_i(Z, w_i) = \sum_{k=1}^{n1} \alpha_k (Z - w_i)^2 + \theta \quad (II.4)$$

avec

$\theta$  : Biais

$\alpha_k$  : Paramètre

Celle-ci a une forme modulée par des paramètres  $\alpha_k$  et dotée d'un biais  $\theta$ .

Ces paramètres permettent de modifier la symétrie de la fonction. Cette fonction conserve cependant les mêmes caractéristiques que la précédente, mais elle est très dépendante des paramètres qui déterminent les formes de zones de regroupement de données de l'espace des entrées.

### I.3.2 Le système dynamique linéaire SISO

Ce bloc détermine la dynamique du réseau. Dans le cas général, cette fonction est régie par l'équation différentielle du premier ordre suivante [Lip87][Del95] :

$$\alpha_0 \dot{x}_i(t) + \alpha_1 x_i(t) = v_i(t) \quad (II.5)$$

Cette équation reflète l'activité électrique réelle du neurone, à savoir la charge et la décharge du potentiel. C'est J. Hopfield qui, pour la première fois, en 1982 a modélisé un réseau de neurones par un circuit électrique régi par cette équation, traduisant son comportement dynamique [Kos92][Kun93].

Un réseau statique, représente un cas particulier de l'équation II.5 où la première dérivé est tout simplement nulle. Celle-ci ne devient donc qu'une simple fonction linéaire.

### I.3.3 La fonction d'activation non linéaire

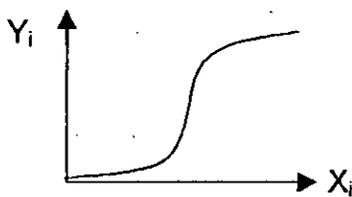
La fonction d'activation transforme le signal  $v_i$  non borné en un signal borné  $y_i$ .

Cette fonction est en général monotone non décroissante, ainsi l'augmentation de l'entrée ne peut que faire augmenter la sortie ou la garder constante. Elles ne peuvent jamais faire décroître le signal.

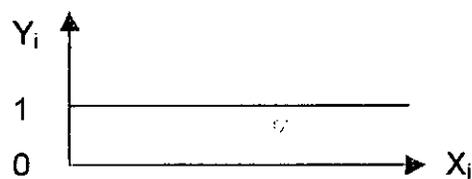
Le choix d'une fonction d'activation linéaire facilite les calculs, mais le neurone perd sa robustesse. Tandis que les neurones dont la fonction d'activation est non linéaire augmente la capacité du réseau à approximer les fonctions complexes [Fre92].

On distingue trois catégories de fonction d'activation :

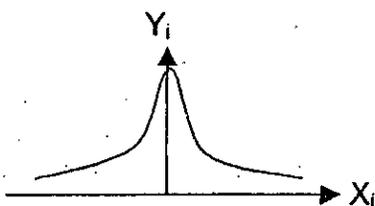
- Une catégorie permet de distinguer entre les fonctions dérivable (sigmoïde, tangente, hyperbolique) et non dérivable (fonction seuil, thresholding) figure II.4.a et b.
- Une deuxième catégorie permet de distinguer les fonctions que possèdent des valeurs importantes autour de zéro (figure II.4.c), et celles qui possèdent des valeurs importantes loin de l'origine (figure II.4.d).
- La troisième catégorie distingue entre les fonctions positives et les fonctions à moyenne nulle (0 et 1 ou -1 et 1) (figure II.4.e).



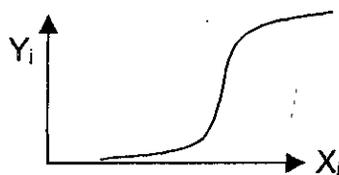
a. fonction sigmoïde (Forme s)



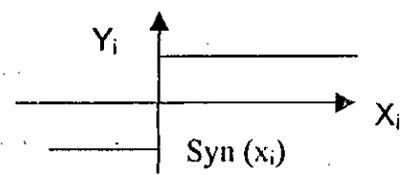
b. fonction thresholding (seuil)



c. fonction gaussienne



d. fonction sigmoïde translaté



e. fonction à moyenne nulle

Figure II.4 Différente forme de fonction d'activation

## II.4 Connexions neuronales et champs de neurones

Dans le domaine des réseaux de neurones, on parle de champs de neurones pour spécifier un regroupement topologique de plusieurs neurones [Kas92]. L'organisation de plusieurs champs entre eux, constitue un réseau de neurones, dont les unités à l'intérieur doivent travailler ensemble pour remplir une tâche spécifique. Cette organisation détermine l'architecture du réseau.

Il existe trois types de couches de neurones : couche d'entrée, couche cachée et la couche de sortie. Deux de ces couches communiquent à travers les poids synaptiques de la connexion pondérée (figure II.5).

- Connexion « Feed back » : c'est le bouclage de l'information de la couche supérieure vers la couche inférieure.
- Connexion latérale : c'est l'interconnexion entre les neurones de la même couche.
- Connexion temporelle (élément retard). Cet élément peut s'insérer entre les connexions pour traduire un comportement dynamique dans le temps (interconnexion et autoconnexion).
- Connexion « feed forward » : c'est la circulation de l'information de la couche inférieure (input layer) et la couche supérieure.

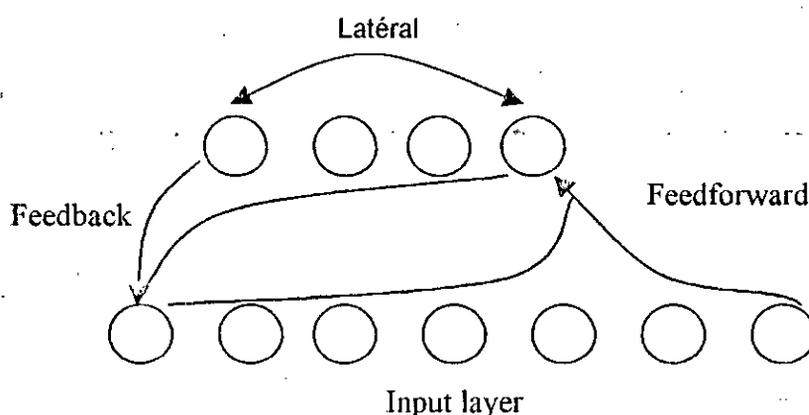


Figure II.5 Représentation d'une connexion neuronale

## II.5 Architecture des réseaux de neurones

L'architecture d'un réseau de neurones définit son fonctionnement. Il existe trois types de réseaux.

### II.5.1 Réseaux statiques

Ce type de réseau est organisé généralement en couches de neurones [Psa88]. Chaque neurone d'une couche reçoit ses entrées à partir des neurones de la couche précédente ou tout simplement de l'entrée du réseau. Dans de tels réseaux il n'existe pas de « feedback » (retour) d'information. Ces réseaux peuvent être utilisés pour les problèmes de classification ou d'approximation de fonctions non linéaires complexes (figure II.6).

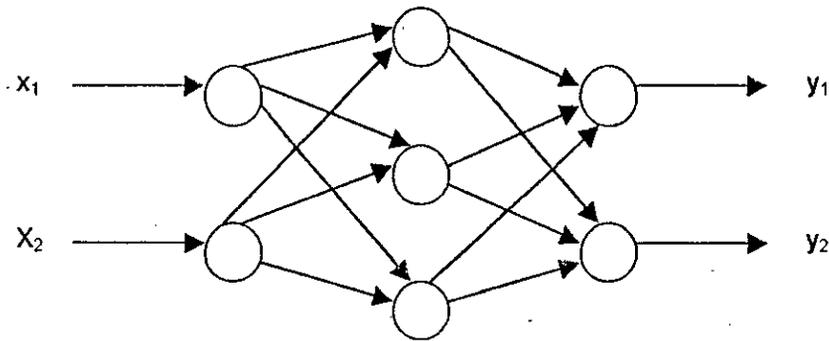


Figure II.6 Architecture du réseau statique

### II.5.2 Réseaux dynamiques

Les réseaux de neurones dynamiques ou récurrents sont différents des réseaux statiques car leur architecture contient un bouclage (la topologie des connexions est bouclée). En général la sortie de chaque neurone est reinjectée en entrée d'autres neurones, grâce à des poids variables. Un réseau de neurone composé de ces éléments, dont tous les poids sont en général non nuls, est dit entièrement connecté. Le réseau étant fondamentalement dynamique, il ne contient qu'une seule couche. L'architecture d'un tel réseau est représentée dans la Figure II.7:

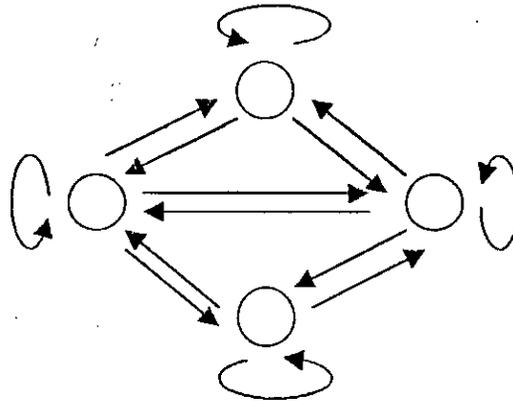


Figure II.7 Architecture d'un réseau récurrent.

L'introduction de bouclage produit un réseau dynamique avec différents points d'équilibre. L'introduction d'une telle structure nécessite l'étude de la stabilité de la dynamique de fondement. Au début, ces réseaux étaient souvent utilisés pour les problèmes de classification, et de mémorisation. Actuellement, dans plusieurs travaux, la structure interne de ces réseaux est exploitée pour l'identification et la commande de systèmes dynamiques [Nar90].

### II.5.3 Réseaux à architecture évolutive et réseaux Auto-Organisés

Les réseaux Auto-organisés sont des réseaux de neurones qui changent leurs structures interne pendant l'utilisation. Ainsi les neurones se regroupent topologiquement suivant la représentation des exemples issus de l'espace d'entrée. Ces réseaux sont généralement des dérivées des modèles de Kohonen, dont le quantificateur de vecteurs LVQ (*Linea Vector Quantiser*) [Koh92][Fre92].

Les réseaux sont dits évolutifs par leur méthode d'apprentissage. C'est la dimension du réseau qui change pendant l'entraînement. Ainsi, le nombre de neurones augmente ou diminue. La structure interne (la nature du réseau) n'est cependant pas bouleversée par ce changement. Parmi ces réseaux, on trouve les GPFN (*Gaussian Potential Function Neural Networks*) entraînés par la méthode d'apprentissage hiérarchique de S. Lee et R.M Kill [Lee91].

## II.6 Stabilité des réseaux de neurones

Le problème majeur dans les architectures de réseaux entièrement inter connecté est l'analyse de leur stabilité. Plusieurs recherches ont été développées afin d'assurer une stabilité asymptotique [Lan79][Kun92][Kos92]. Par ailleurs le formalisme de fonction de Lyapounov s'avère une alternative intéressante. Mais le principal désavantage de cette méthode est dû au fait qu'il n'existe pas une règle générale pour trouver une 'bonne' fonction de Lyapounov. Ce qui empêche de résoudre le problème de stabilité dans beaucoup de cas [Lan 79][Gue 95]. En outre le concept d'hyperstabilité introduit par Popov est une généralisation de la notion de stabilité absolu. Un tel outil permet l'analyse de la stabilité dans le cas d'architecture récursive non linéaire. La mise sous forme d'un bloc SPR (Strict Positive Real) et un bloc non linéaire hyperstable est une étape très importante dans cette analyse [Lan79][Pop73][Gue95].

Cohonen et Groosberg ont rigoureusement démontré qu'il existe une fonction globale d'énergie pour un réseau de neurones dynamique [Kun92]. L'équation décrivant son fonctionnement englobe tous les types de réseaux de neurones.

Le modèle du réseau de neurone utilisé, est décrit par le système d'équations différentielles non linéaires continues entièrement connectées [Kos92] :

$$\frac{dx_i}{dt} = a_i(x_i)[b_i(x_i) - \sum_{j=1}^n c_{ij}d_j(x_j)] \quad (\text{II.6})$$

Où  $x_i$  représente l'activité du  $i^{\text{ème}}$  neurone,  $d(.)$  la sortie du neurone qui par ailleurs doit être monotone,  $c = [c_{ij}]$  la matrice des poids synaptiques et  $a(.)$  Fonction définie positive.

Le modèle du réseau défini par l'équation II.6 présente un cas général d'un réseau entièrement interconnecté. Les autres réseaux peuvent être identifiés comme des cas particuliers de cette équation. Pour démontrer la stabilité de ce modèle, la fonction d'énergie proposée est de la forme [Kun92][Fre92] :

$$V(x) = -\sum_{i=1}^n \int_0^{x_i} b_i(\xi_i) d'_i(\xi_i) d\xi_i + \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n c_{jk} d_j(x_j) d_k(x_k) \quad (\text{II.7})$$

Pour vérifier les conditions de la fonction de Lyapounov, on a :

$$\dot{V}(x) = -\sum_{i=1}^n a_i(x_i) d'_i(x_i) [b_i(x_i) - \sum_{k=1}^n c_{ik} d_k(x_k)]^2 \quad (\text{II.8})$$

Ainsi sous conditions que  $c_{ij} = c_{ji}$  ce qui revient à une matrice C des poids synaptiques symétrique.

Et sachant que  $a_i(x_i) > 0$  (les fonctions  $a(.)$  sont définies positives) et  $d'_i(x_i) > 0$  (la fonction d'activation  $d(.)$  est monotone), on obtient  $\frac{dv}{dt} \leq 0$ .

Ainsi la stabilité de cette couche de neurones entièrement connectée est démontrée.

Les autres modèles de réseaux étant en général de structures relativement plus simples à établir leur stabilité. Les fonctions d'énergie d'autres réseaux comme celui de Hopfield ne sont que des cas particuliers du modèle de Grossberg [Kuw92].

## II.7 Apprentissage des réseaux de neurones

On appelle apprentissage, l'opération par la quel le réseau de neurones acquiert la capacité de réaliser certaines tâches en modifiant ces paramètres internes (poids synoptique). Cette modification se fait en utilisant un algorithme d'adaptation paramétrique appelé «Algorithme d'apprentissage» [Fre92][Kun92]. L'apprentissage consiste alors à améliorer les performances futures du réseau en se basant sur les expériences acquises.

Plusieurs modes d'apprentissage peuvent être distingués [Kos92]. Ceux-ci se différencient principalement par le type d'exemples disponibles (figure II.1).

- L'apprentissage supervisé correspond au cas où l'on dispose d'exemples d'entrées et de sorties simultanément. On cherche alors à construire un modèle capable de prédire la valeur de la sortie connaissant celle de l'entrée. il se fait en présence d'un superviseur (teacher) qui dirige le comportement du réseau en lui présentant les couples d'entrées et leurs sorties désirées. Cet apprentissage se fait toujours par l'intermédiaire d'un critère à optimiser définissant la performance du réseau à chaque étape.
- Lorsque l'on ne connaît que les valeurs d'entrées que l'on cherche à réaliser une classification sur l'ensemble des exemples, on se trouve dans le cas de l'apprentissage non supervisé. Il nécessite la présence des entrées seulement sans l'intervention d'un superviseur. Cet entraînement se fait sur la base d'informations locales existant aux niveaux des neurones et découvre les propriétés collectives qui existent entre les données sur la base desquelles le réseau doit s'organiser.
- Un cas intermédiaire se présente lorsque l'on peut disposer d'une information du type punition/ récompense [Kos92] renseignant sur la qualité de la classification : ce cas correspond à l'apprentissage par renforcement. Ce type d'apprentissage est moins «classique» que les deux premiers qui sont les principaux. Il a d'une part en commun avec l'apprentissage supervisé, la présence d'un critère qui juge l'évolution de l'apprentissage, et d'autre part il ne nécessite que des entrées sans définir les sorties désirées comme dans l'apprentissage non supervisé. Dans ce type d'apprentissage, le réseau ajuste ses poids synaptiques suivant un critère de performance. Celui-ci renforce les poids du réseau si le critère y est favorable et le punit dans le cas contraire.

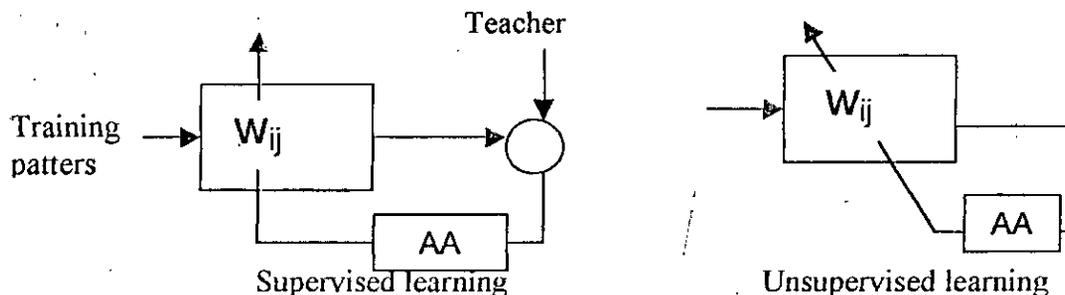


Figure II.8 Schéma des différents apprentissages (AA :Algorithme d'Apprentissage)

## II.8 Algorithme d'apprentissage

Dans le cas de l'apprentissage non supervisé, la règle de Hebb permet la détermination directe des poids synoptiques. Cette règle représente une méthode qualitative d'adaptation des poids. Nous présenterons son contenu tel qu'énoncé par J. Hebb en 1949 [Yed98] :

« Si un neurone A est stimulé à maintes reprises par un autre neurone B pendant qu'il est actif, le neurone A devient plus sensible aux stimulations en provenance de B ; Le lien synaptique de B vers A devient plus significatif. Ainsi B pourra stimuler A plus facilement à l'avenir »

La formulation mathématique la plus utilisée repose sur la méthode de corrélation [Fre92]. Une extension de cette règle aux réseaux compétitifs permet la mise en compétition des neurones du réseau afin de trouver le neurone gagnant représentant le groupe (cluster) dans lequel l'entrée sera classée. Au bout de la compétition, la règle du winner take all où le neurone gagnant, représentant la classe à laquelle le vecteur d'entrée devrait appartenir, seront attribués la valeur maximale «1» comme état et les autres seront inhibés. Par contre dans le cas de l'apprentissage supervisé, la détermination des poids synaptiques se fait à partir d'observation sur des échantillons d'entrées sorties simultanément. Ce type d'apprentissage nécessite la présence d'un superviseur ayant pour rôle de ramener le réseau vers le comportement désiré. Ceci peut se faire par la recherche du vecteur synaptique optimal parmi toutes les combinaisons possible dans l'espace des poids assurant la meilleure approximation possible [Kos90].

Ayant un même objectif, deux approches peuvent être faites pour ce type d'apprentissage. la première est basée sur la validité de la décision du réseau représentée par sa sortie. Ce type de réseau est appelé DBNN «Decision Based Neural Network » ou réseau basé sur l'optimisation d'une fonction coût [Kun92].

Il est nécessaire de déterminer, dans ce genre d'apprentissage, la fonction qui mesure l'écart entre les sorties désirées et celles fournies par le réseau. Cette fonction est appelée fonction objective et constitue le critère à minimiser qui représente en général une fonction de l'erreur.

$$E = \psi(y_k - y_k^*) \quad (\text{II.9})$$

Où  $y_k^*$  : Sortie désirée  
 $y_k$  : Estimation du réseau

La première méthode analytique générale pour la résolution de problèmes d'extremum fut conçue par Fermat en 1629 [Lac94]. Il obtint la condition nécessaire d'extremum (le gradient est nul), un des résultats des plus importants en analyse. Il fut suivi par Leibniz qui formula essentiellement la condition du second ordre ; ce qui permet de distinguer un minimum d'un maximum. C'est dans les années 40 que les exigences de la technique ont donné naissance à une nouvelle branche de l'analyse. la théorie mathématique appropriée fut bâtie au milieu des années 50 sous le nom de commande optimal [Fri97]. Le rôle le plus important y fût tenu par le principe du maximum de pontriguine. Ce principe est essentiellement une généralisation des travaux antérieurs du calcul variationnel. Sous une forme différente, les conditions d'optimalité sont obtenues conjointement par Bellman.

L'un des algorithmes les plus répandus est celui de la «Back propagation» [Els61][Cun88]. Cet algorithme change les poids synaptiques du réseau dont l'architecture est fixée par le superviseur, à chaque fois qu'un couple d'entrée sortie lui est présenté. Ceci est réalisé grâce à la méthode de la descente du gradient. A chaque itération, le signal d'entrée se propage dans le réseau dans le sens inverse (sortie - entrée). Malheureusement, cet algorithme souffre du problème du minimum local.

Une variante de la méthode précédente consiste à choisir un déplacement adéquat pour accélérer la convergence de l'algorithme, on aboutit alors au fast back propagation avec momentum [Kos92]. Une autre version proposée par White en 1989 est appelé Robust Back propagation applicable dans le cas stochastique.

Afin d'améliorer le choix de la direction à entreprendre dans l'espace des poids, on a recours aux méthodes d'optimisation du second ordre de la fonction objective spécifiée par le Hessien.

Il existe beaucoup de variantes possibles dans le choix de la formule de mise à jour de la matrice Hessienne. La méthode de Leavenberg Marquand remplace le calcul du Hessien par une approximation numérique d'une façon itérative [Kol89][Lac94].

Les méthodes précédentes supposaient que la fonction à minimiser est continûment différentiable. Autrement dit, en chaque point la fonction admet un gradient. Lorsque la fonction n'est pas différentiable, ou bien lorsqu'on ne peut pas employer un algorithme du gradient, l'utilisation d'autres méthodes s'avère nécessaire. On procède, alors à des approximation (de type différence finie) de la dérivé de la fonction objective (méthodes de Miñlin). L'algorithme consiste à résoudre par la factorisation de Choleski un système linéaire pour la détermination du gradient. On peut citer encore une méthode de Hooke et Jeeves, méthode de Powell, méthode de Rosenbroock [Lac94][Li95].

Les méthodes de recherche aléatoire ont été largement explorées et utilisées dans le domaine des réseaux de neurones [Lac94][Bab89]. Ceci est dû à leur structure très simple à implementer et facilement modifiable pour les adapter à un problème donné. De plus, l'algorithme est robuste et fonctionne généralement bien pour des fonctions très irrégulières.

Le principe de base est de déterminer, partant d'un point donné un nouveau point à partir d'une fonction de densité de probabilité. Ce nouveau point est en général accepté lorsqu'une décroissance de la fonction est obtenue [Fre92][Kun92][Lac94].

Une autre méthode basée sur ce principe est l'algorithme du recuit simulé. L'algorithme qui possède une certaine analogie avec un système thermodynamique démarre d'une configuration arbitraire et simule le comportement du système à une température donnée. Quand l'équilibre thermique est atteint, on diminue la température et on attend que l'équilibre thermique se stabilise. L'opération est répétée jusqu'à obtention d'une configuration d'énergie minimale.

## II.9 Rappel sur les différents types de commande par réseaux de neurones

En dynamique de structure, la contrainte temporelle pour le contrôle en temps réel peut être draconienne. En effet pour éviter une dégradation des caractéristiques de la structure, il faut réagir en un minimum de temps en cas d'excitation sismique. En outre, la nécessité du traitement parallèle de l'information peut résulter de puissance de calcul élevée. C'est pour cela que l'utilisation des réseaux de neurones en dynamique de structure s'avère intéressante [Jez95][Ban98][Ban99].

Contrairement aux algorithmes de commande classiques où la loi de commande est formulée explicitement, le régulateur neuronal apprend la tâche de commande. Le réseau de neurones acquiert les connaissances sur la commande du système à partir d'un ensemble d'exemples et mémorise ces informations dans les poids d'interconnexion.

Il existe plusieurs stratégies de commande par réseaux de neurones. Nous citerons dans ce qui suit les méthodes les plus connues, où les réseaux de neurones ont trouvé leur application directe [Hun92][Fuk92][Mil92][Nar90][Yed98].

### II.9.1 Commande inverse

Dans cette commande, un réseau de neurones apprend le modèle inverse d'un système. Ce réseau est ensuite mis en cascade avec le système pour le commander. Si le réseau a bien appris le modèle inverse, la sortie du système sera très proche de la référence fournie par un opérateur humain ou un programme de calcul à l'entrée du réseau (figure II.9).

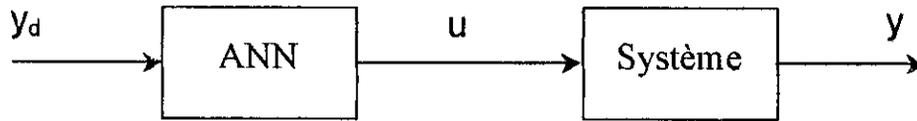


Figure II.9 Schéma de la commande inverse.

### I.9.2 Commande supervisée

Un réseau de neurone apprend à imiter un être humain ou un programme de calcul qui sait déjà accomplir une tâche donnée. Cette technique permet d'améliorer les performances du régulateur existant en utilisant les caractéristiques intrinsèques des réseaux de neurones (figure II.10).

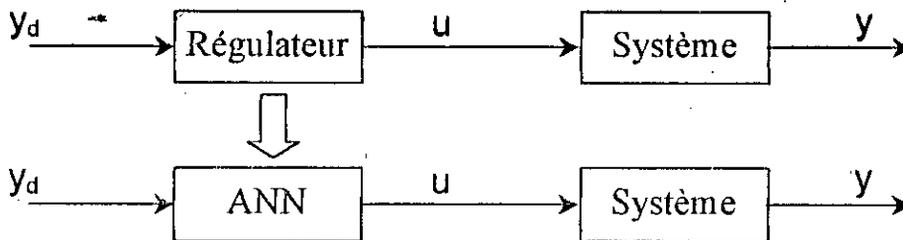


Figure II.10 Schéma de la commande supervisée

### II.9.3 Commande par bouclage linéarisant

En utilisant le modèle de connaissance du système, on synthétise un retour par réseaux de neurones dont le but est de linéariser le système. Puis une commande classique est appliquée au système linéarisé (figure II.11).

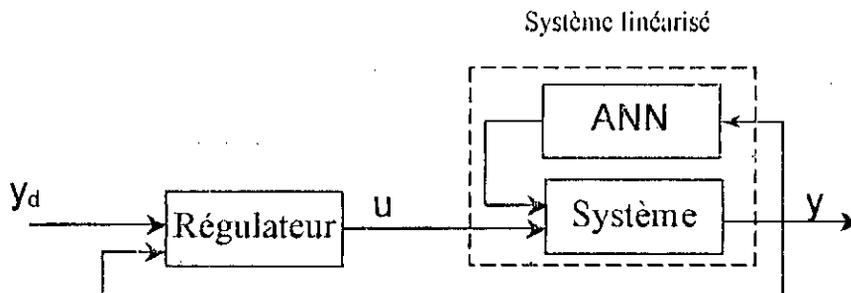


Fig. II.11 Schéma de la commande par bouclage linéarisant

### II.9.4 Commande adaptative par modèle de référence

Les performances du système en boucle fermée sont spécifiées par un modèle de référence. L'erreur entre le système et le modèle de référence est utilisée pour entraîner un réseau de neurones qui fournit la commande nécessaire pour ramener la sortie du système à la sortie désirée. Dans le cas où le modèle de référence serait le système identifié cette approche coïncide avec la commande inverse (figure II.12).

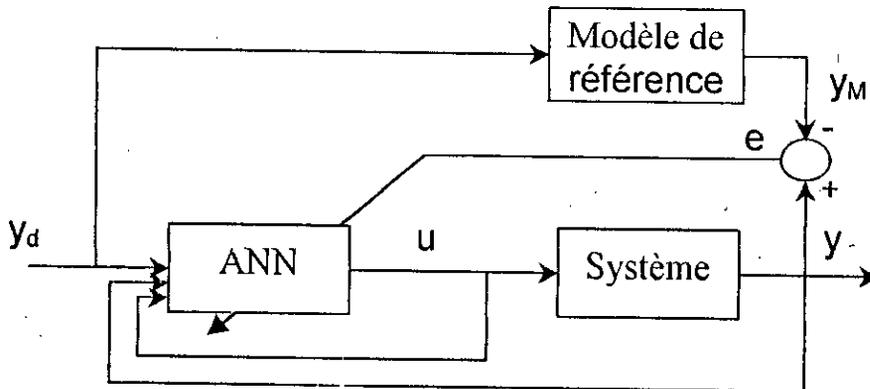


Figure II.12 Schéma de la commande adaptative par modèle de référence en utilisant les réseaux de neurones

### II.9.5 Commande adaptative auto-ajustable

C'est une commande adaptative indirecte, où le réseau est utilisé pour estimer le modèle ou les paramètres du modèle du système à commander. La commande est ensuite calculée en utilisant ces estimations (figure II.13).

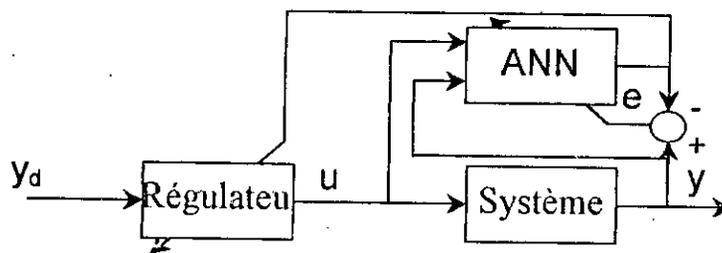


Figure II.13 Schéma de la commande adaptative auto-ajustable par RNA

### II.9.6 Commande prédictive

Dans cette approche un réseau de neurones est entraîné pour prédire la sortie future du système sur un horizon donnée. La prédiction fournie par le réseau est utilisée dans un algorithme d'optimisation afin de minimiser un certain critère de performance et produire le signal de commande nécessaire (figure II.14).

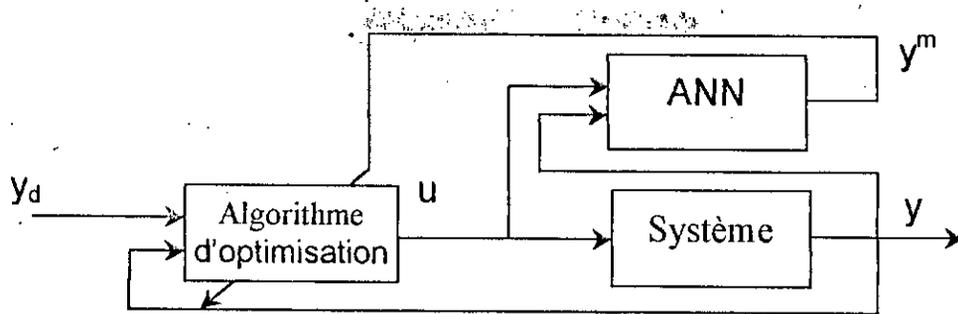


Figure II.14. Schéma de la commande prédictive par RNA

### II.9.7 Commande à apprentissage par renforcement

La stratégie de cette commande consiste à minimiser un critère ou un index de performance appelé *fonction critique* fourni par l'utilisateur. Elle nécessite l'utilisation de deux réseaux, l'un a pour rôle de commander le système, l'autre, appelé réseau critique sert à évaluer la fonction critique et orienter l'adaptation des poids du réseau régulateur en utilisant la technique de l'apprentissage par renforcement. L'avantage de cette stratégie est qu'elle ne nécessite pas la connaissance du modèle du système à commander (figure I.15).

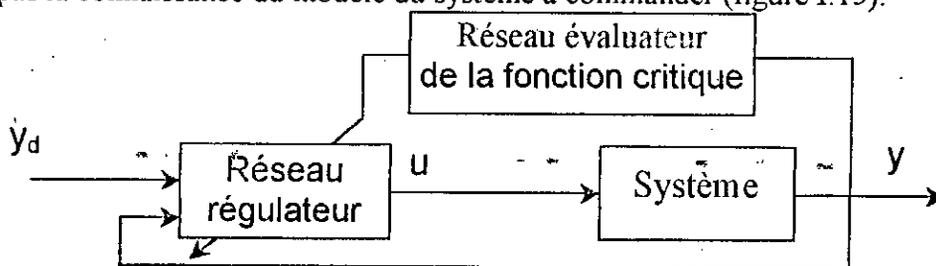


Figure II.15 Schéma de la commande à apprentissage par renforcement

### II.10 Implémentation des réseaux de neurones

Pour l'étude du concept de parallélisme des réseaux de neurones, il est important de s'intéresser à l'aspect implémentation et réalisation matériel de cette architecture. On peut citer les plus importantes alternatives afin de réaliser cette objectif [Kun92][Mul90]:

1. Simulation numérique utilisant les machines séquentielles (type Von Newman).
2. Calcul parallèle utilisant des systèmes multiprocesseurs.
3. Réalisation des réseaux de neurones en utilisant des composants analogiques ou numériques.

La première tentative est la plus utilisée vue la disponibilité de l'ordinateur standard dans la communauté scientifique. Ces ordinateurs ne permettent pas de traduire ni le comportement réel du réseau ni le temps nécessaire de traitement de l'information.

Une des solutions à la conception des réseaux de neurones est l'architecture parallèle. Elle permet la distribution des tâches de traitement aux différents processeurs. De nouveaux concepts d'architecture de machine ont vu le jour, à savoir les systèmes multiprocesseurs, les architectures pipe-line, les matrices processeurs ou les automates circulaires [Hir90].

Les premiers composants implémentant les concepts architecturaux ainsi obtenus apparaissent sur le marché : Les processeurs RISC (Reduced Instruction Set Computer) les processeurs à flots données, les réseaux neurométrique, les matrices SIMD (single introduction stream, multiple data streams) ou les transputeurs (transistor, computer) [Hir90][Kun92]. Par ailleurs,

la disponibilité de ces composants conduit vers un nouvel essor de la recherche appliquée, portant sur les modes d'interconnexion de plusieurs processeurs en réseau (machine hypercube, réseau reconfigurable, structure pyramide et maillage en réseau carré).

Historiquement, la réalisation de systèmes parfait reposait sur le principe que le coût du traitement est plus élevé que celui des mémoires. Ce qui conduit à réaliser des systèmes monoprocesseurs connectés à d'énormes mémoires, aboutissant au goulet d'étranglement intrinsèque à la nature séquentielle des processeurs utilisés. Les progrès de la technologie sont tels qu'à l'heure actuelle cette tendance peut être inversée. Cette potentialité a été exploitée par INMOS et a conduit à la mise sur le marché du transputer [Hir90]. Ce microprocesseur peut directement être utilisé pour concevoir des systèmes hautement concurrents. Il a par ailleurs été conçu de manière à être l'élément de base de systèmes complexes à l'analogie du transistor dans les réalisations électroniques analogiques. Ce qui traduit sa dénomination. C'est un computer (ordinateur) monoboîtier et un composant silicium comme un transistor.

Selon la version, le transputer est un microprocesseur 16 ou 32 bits interconnectable à quatre autres transputers (ou processeurs conventionnels au moyen d'interface adapté) au moyen de canaux de communication série unidirectionnels intégrés appelés liens. Il peut par ailleurs accéder à des ressources propres ou partager à l'aide d'un bus parallèle configurable conventionnel.

La figure II.16 donne une représentation fonctionnelle de ce processeur.

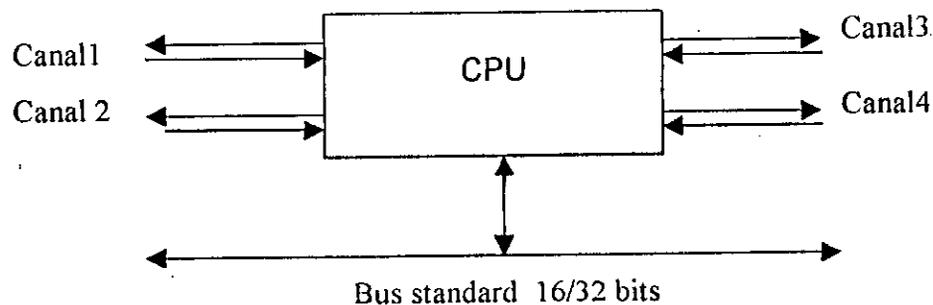


Figure II.16 Représentation fonctionnelle du transputer

Le transputer peut être utilisé comme un microprocesseur conventionnel de haute performance, mais il est particulièrement destiné à la réalisation de réseaux de processeurs de topologies variés. La conception du transputer est le résultat d'une collaboration étroite entre les concepteurs du circuit et les créateurs du langage OCCAM, un langage qui est basé sur les notions de concurrence et de communication [Hir90]. Le transputer intègre ainsi matériellement des caractéristiques importantes d'OCCAM. En particulier, le transputer est une réalisation matérielle de l'objet de base de ce langage, à savoir le processus.

OCCAM est un langage de programmation structuré qui met l'accent sur l'interdépendance entre processus concurrents. Les notions d'entrée-sortie, de priorité entre processus, d'interruption et de temps réel ont été formellement intégrés au langage. Ce qui supprime la nécessité d'un système d'exploitation capable de prendre en charge ces notions. Par ailleurs un programme OCCAM peut directement refléter la structure algorithmique d'une application [Lac94]. Cette structure peut ensuite être simplement transposée sur un réseau de transputers. OCCAM inclut en effet des directives permettant de décrire la configuration du réseau à exécuter l'application. On peut ainsi préciser les processus qui doivent être exécutés par un processeur donné et les liaisons par les liens entre transputers.

L'utilisation des transputers et du langage OCCAM doit donc se faire conjointement si l'on veut tirer une efficacité maximale de ce nouveau composant. Néanmoins, les langages usuels peuvent être employés à condition de lier à ceux-ci les constructeurs particuliers d'OCCAM [Hir90].

L'avantage de ces nouveaux composants ou des réseaux de processeurs est d'autoriser l'obtention de performances notablement accrues et de permettre une réalisation d'application

jusqu'ici difficilement imaginable. Mais ceci est au prix d'un changement de la manière de «penser» une application et des méthodes traditionnelles utilisées pour la conception de systèmes à plusieurs microprocesseurs.

## II.11 Conclusion

La nécessité du traitement parallèle de l'information peut résulter de puissances de calculs élevées pour le traitement d'une grande quantité de données en temps réel. Dans le cas du contrôle en temps réel d'un processeur industriel, la contrainte temporelle peut être draconienne. En effet, pour éviter une dégradation du processus à contrôler, il faut souvent réagir en un minimum de temps.

Dans ce chapitre, nous avons présenté les concepts de base pour la compréhension et l'utilisation des réseaux de neurones. Nous avons procédé par une construction d'une architecture général permettant d'englober tous les types de réseaux. Un modèle mathématique général de l'élément de base du réseau a été donné.

Les différents types de réseaux de neurones peuvent être construits aisément par un choix simultané d'une architecture et de champs de neurones adéquats. Ainsi une généralisation des algorithmes d'apprentissage est donnée par une synthèse des méthodes d'optimisation utilisées en calcul variationnel.

La commande par réseaux de neurones est étroitement liée à l'identification. En effet dans la plus part des stratégies de commandes, on utilise un modèle neuronale direct ou inverse du système à commander. En outre les réseaux de neurones sont fortement sollicités pour la commande des systèmes non linéaires. On a présenté les stratégies de commande les plus utilisées permettant ainsi un choix aisé de la structure de commande neuronale dans le cas d'une structure.

Enfin, nous avons mis à la lumière, les méthodes d'implémentation en temps réel à savoir les transputer.

Le transputer est un composant VLSI programmable conduisant de manière remarquablement simple à des performances élevées et à une adaptabilité extrême aux besoins des utilisateurs. Son architecture interne révolutionnaire autorise l'implémentation concurrence dans un système. Ceci simplifie la conception et la réalisation des applications et ouvre des perspectives nouvelles pour les applications du future. Par ailleurs il est relativement simple à programmer.

---

## Chapitre III

### *Commande Active Linéaire Déterministe et Stochastique*

---

#### **Introduction**

Les algorithmes de commande de type proportionnel, intégral et dérivé commandent de nombreux processus industriels. Leur simplicité et leur robustesse, leurs bonnes performances, dans de nombreux cas, expliquent leur succès. Le choix, souvent difficile, des paramètres de ces régulateurs s'appuie sur une connaissance grossière du processus lui-même [Buh83][Ast90]. Dans de nombreux domaines (dynamique des structures), ces méthodes classiques montrent rapidement leurs limites. Aucun régulateur à paramètres constants ne peut prendre en charge les évolutions temporelles du système à commander. En effet, l'approximation d'un procédé réel par un système linéaire peut se traduire par le fait que les paramètres caractérisant le point de fonctionnement dépendent du temps. Les performances d'une commande classique se dégradent quand l'écart entre les paramètres approximatifs et réels devient non négligeable [Ast89]. Les développements récents de la mini et de la micro-informatique rendent possible l'implémentation de lois de commande complexes qui demandent un traitement substantiel. Ce développement a favorisé l'émergence de différentes méthodes d'identification et de commande de processus en temps réel [Ast89][Lan93][Chal87].

Durant les dix dernières années un important travail a été fait en ce qui concerne la commande adaptative. C'est en 1958 que Kalman proposa pour la première fois un algorithme auto-ajustable de commande. L'extension et l'implémentation de cet algorithme a été étudiée par Astrom [Ast70] et Astrom & Wittenmark [Ast73]. Depuis, plusieurs extensions ont apparues, tels que placement de pôles et de zéros [Wel89], variance minimale généralisée [Cla75] et LQG [Cla85] (Linear Quadratic Gaussian).

La commande des structures en génie civil a fait l'objet, ces dernières décennies, d'un grand nombre d'études théoriques et expérimentales au cours desquelles plusieurs stratégies de commandes ont été considérées [Hou97]. Ce développement a été motivé par le besoin de protéger l'intégrité des structures et sauver les vies humaines des occupants en cas de forts et

imprévisibles séismes. L'objectif de la commande des structures est de réduire l'amplitude de leurs vibrations sous l'effet d'une charge dynamique extérieure.

Dans un premier lieu, nous présentons la commande optimale à critère quadratique (LQC Linear Quadratic Controller). Le paragraphe III.2 est consacré à l'algorithme à Variance Minimale. L'algorithme à Variance Minimale généralisée est exposé dans la section III.3. En second lieu, nous développons respectivement l'algorithme de Poursuite et Régulation avec Pondération de l'Entrée et celui à Placement de Pôles. Nous exposons ensuite l'algorithme à Placement de Pôles et de Zéros. La commande linéaire par poursuite d'un modèle est développée (Linear Model Following Controller) en dernier lieu.

L'extension des algorithmes au cas multivariable est exposé en détail dans le paragraphe III.8. Les résultats de simulation des différents algorithmes appliqués aux robots de classe quatre et aux structures en génie civil sont exposés dans la dernière section. Une conclusion générale clôture le chapitre.

### III.1 Commande quadratique linéaire déterministe (LQC : Linear Quadratic Controller)

Dans la commande optimale, le choix du critère de performance spécifie les caractéristiques du système en boucle fermée. L'objectif est la recherche d'une commande qui minimise cette fonction coût. En général, le critère est choisi sous forme quadratique, pour assurer une commande linéaire en fonctions des états du système [Fri87][Ath71]. La commande ainsi obtenue est appelée **commande quadratique linéaire** (Linear Quadratic Controller) ou bien LQG (Linear Quadratic Gaussian) si le système est soumis à des perturbations stochastiques gaussiennes [Lu98].

Dans le cas de la régulation, le solution du problème LQC, à horizon fini, consiste à la résolution de l'équation de Riccati [Ath71][Had96]. Tandis que pour la poursuite, le problème s'étend à la résolution d'une autre équation qui donne la commande nécessaire assurant cet objectif [Ath66].

L'objectif de ce paragraphe est de formuler de façon plus générale le problème de commande optimale d'une structure. Il s'agit de conduire le structure vers un état désiré avec un signal de commande minimisant une fonction coût. La définition de cette fonction est importante car elle détermine la loi de commande optimale. L'exposé est basé sur le principe d'optimalité de Bellman. Il porte principalement sur les systèmes linéaires soumis à des critères de forme quadratique pour lesquelles la commande s'exprime linéairement en fonction de l'état courant [Ath71][Fri87][Buh83].

On présente l'algorithme de commande optimale sous ces différentes version : La commande optimale classique, la commande en boucle fermé et la commande en boucle ouverte [Yan90] (closed-open loop, closed loop control, open loop control). Le modèle de connaissance de la structure est défini par l'équation 1.23.

Le critère à minimiser est défini par [Had97][Kwa72] :

$$J = \int_0^t \left[ \underline{z}^T(t) \underline{Q} \underline{z}(t) + \underline{u}^T(t) \underline{R} \underline{u}(t) \right] dt \quad (\text{III.1})$$

où  $\underline{Q}$  est  $[2n \times 2n]$  matrice de pondération.

$\underline{R}$   $[n \times n]$  matrice de pondération de la commande.

$t_f$  durée de l'excitation séismique.

La minimisation de ce critère sous la contrainte, permet d'aboutir à la loi de commande optimale. Cette loi est obtenue en utilisant les méthodes de calcul variationnel [Els61] et le principe du maximum de Pontriaguine. Après quelques manipulations mathématiques, on obtient la solution optimale au problème [Kwa72][Had97].

En introduisant le multiplicateur de Lagrange, on obtient :

$$\tilde{J} = \int_0^{t_f} \left\{ \underline{z}^T(t) \underline{Q} \underline{z}(t) + \underline{u}^T(t) \underline{R} \underline{u}(t) \right\} + \underline{\lambda}^T(t) \left[ \underline{A} \underline{z}(t) + \underline{B} \underline{u}(t) + \underline{W} \dot{\underline{x}}_0(t) - \dot{\underline{z}}(t) \right] dt \quad (III.2)$$

L'hamiltonien sera défini alors par :

$$H \left[ \underline{z}(t), \underline{u}(t), \underline{\lambda}(t), \dot{\underline{x}}_0(t), t \right] = \underline{z}^T(t) \underline{Q} \underline{z}(t) + \underline{u}^T(t) \underline{R} \underline{u}(t) + \underline{\lambda}^T(t) \left[ \underline{A} \underline{z}(t) + \underline{B} \underline{u}(t) + \underline{W} \dot{\underline{x}}_0(t) \right] \quad (III.3)$$

Où  $\underline{\lambda}(t)$  est un vecteur de dimension  $2n$  représentant le co-état (the costate variables). En procédant par intégration par partie, on obtient :

$$\tilde{J} = \int_0^{t_f} \left\{ \underline{z}^T(t) \underline{Q} \underline{z}(t) + \underline{u}^T(t) \underline{R} \underline{u}(t) \right\} + \underline{\lambda}^T(t) \left[ \underline{A} \underline{z}(t) + \underline{B} \underline{u}(t) + \underline{W} \dot{\underline{x}}_0(t) \right] dt + \int_0^{t_f} \underline{\lambda}^T(t) \dot{\underline{z}}(t) dt - \underline{\lambda}^T(t) \underline{z}(t) \Big|_0^{t_f} \quad (III.4)$$

et par substitution de l'équation (III.3) dans l'équation (III.4), on obtient :

$$\tilde{J} = \int_0^{t_f} \left\{ H \left[ \underline{z}(t), \underline{u}(t), \underline{\lambda}(t), \dot{\underline{x}}_0(t), t \right] + \underline{\lambda}^T(t) \dot{\underline{z}}(t) \right\} dt - \underline{\lambda}^T(t_f) \underline{z}(t_f) + \underline{\lambda}^T(0) \underline{z}(0) \quad (III.5)$$

En utilisant le calcul variationnel, on aboutit à [Els61][Fri87][Had96] :

$$\delta \tilde{J} = -\underline{\lambda}^T(t_f) \delta \underline{z}(t_f) + \underline{\lambda}^T(0) \delta \underline{z}(0) + \int_0^{t_f} \left\{ \left[ \underline{\lambda}^T(t) + \frac{\partial H}{\partial \underline{z}(t)} \right] \delta \underline{z}(t) + \frac{\partial H}{\partial \underline{u}(t)} \delta \underline{u}(t) \right\} dt \quad (III.6)$$

En prenant en considération les conditions initiales (condition au limite) fixe, la variation de  $J$  définie par l'équation (III.6) est nulle si on satisfait les conditions suivantes :

$$\frac{\partial H}{\partial \underline{u}} = 0, \quad 0 \leq t \leq t_f \quad (III.7)$$

$$\dot{\underline{\lambda}}^T = -\frac{\partial H}{\partial \underline{z}} \quad (III.8)$$

Avec les conditions au limites  $\underline{\lambda}(t_f) = \underline{0}$

En remplaçant les équations du modèle de connaissance dans les équations (III.7) et (III.8) on obtient [Had96b] :

$$\dot{\underline{\lambda}}(t) = -2\underline{Q} \underline{z}(t) - \underline{A}^T \underline{\lambda}(t) \quad (III.9)$$

$$\underline{u}(t) = -\frac{1}{2} \underline{R}^{-1} \underline{B}^T \underline{\lambda}(t) \quad (III.10)$$

où  $\underline{\lambda}$  : multiplicateur de Lagrange

En général, la commande est exprimée par un retour d'état et une excitation extérieure ayant la forme suivante [Yan90][Yan87] :

$$\underline{\lambda}(t) = \underline{P}(t)\underline{z}(t) + \underline{q}(t); \quad \underline{\lambda}(t_f) = \underline{0} \quad (\text{III.11})$$

La détermination de la matrice  $\underline{P}(t)$  et du vecteur  $\underline{q}(t)$  se fait par substitution de l'équation (III.11) dans l'équation de la structure. On obtient alors [Had96][Yan90] :

$$\begin{aligned} & \left[ \underline{\dot{P}}(t) + \underline{P}(t)\underline{A} - \frac{1}{2}\underline{P}(t)\underline{B}\underline{R}^{-1}\underline{B}^T \underline{P}(t) + \underline{A}^T \underline{P}(t) + 2\underline{Q} \right] \underline{z}(t) + \\ & \underline{\dot{q}}(t) - \left[ \frac{1}{2}\underline{P}(t)\underline{B}\underline{R}^{-1}\underline{B}^T - \underline{A}^T \right] \underline{q}(t) + \underline{P}(t)\underline{W}\ddot{x}_o(t) = \underline{0} \end{aligned} \quad (\text{III.12})$$

En exploitant les équations précédentes, on aboutit à trois versions de la loi de commande. Par ailleurs, l'excitation séismique est considérée comme modèle de référence de notre système. Ceci constitue une application directe en dynamique de structure en Génie civil [Yan90].

La résolution en temps inverse de l'équation de Riccati qui nécessite le calcul de la commande optimale sur la totalité de l'intervalle de temps puis sa mise en mémoire avant son exploitation est une difficulté majeure du point de vue pratique [Lou90]. Aussi, dans la majorité des applications, il est préférable de coïncider le temps d'observation comme étant infini. On démontre que lorsque le temps est infini, la matrice de retour  $k(t)$  tend vers une limite unique  $K$  [Lou90][Kwe72][Fri87].

### III.1.1 La Commande 'Optimal Open-Closed Loop Control'

Cette loi de commande proposée par Yang et constitue la solution de base en commande optimale [Yan90].

Dans cette approche, on s'intéresse à réaliser deux objectifs : la poursuite et la régulation [Lan86][Lan93]. Le but de cette commande est d'obtenir un signal d'erreur nul. Cette condition n'est en général pas réalisée du fait qu'au moins un des cas suivant est vérifié [Gue95]:

- La présence de perturbations qui affectent le processus, ce qui nécessite le maintien de la grandeur de sortie à la valeur de consigne : c'est le problème de régulation.
- Les variations de la consigne, il s'agit alors d'un problème de poursuite.

Ces deux problèmes peuvent être résolus simultanément par le choix d'une stratégie de commande appropriée : C'est la commande par poursuite et régulation à objectif indépendant [Gue95][Lan93].

Le problème revient alors à résoudre l'équation suivante :

$$\underline{P}\underline{A} - \frac{1}{2}\underline{P}\underline{B}\underline{R}^{-1}\underline{B}^T \underline{P} + \underline{A}^T \underline{P} + 2\underline{Q} = \underline{0} \quad (\text{III.13})$$

La structure d'une telle commande est définie dans la figure III.1

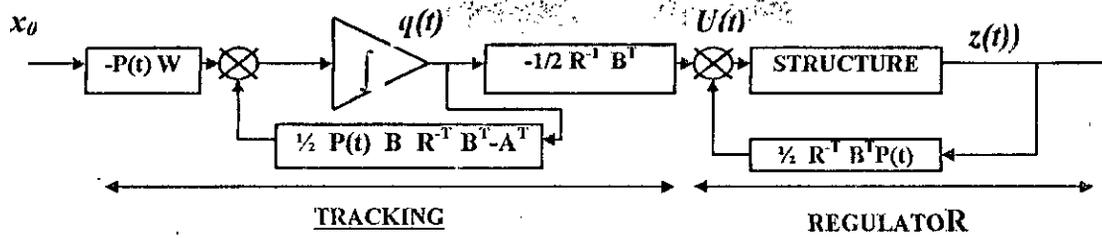


Figure III.1 Schéma bloc de la commande 'Closed-Open Loop Control'

Le bloc régulateur de la figure montre bien la structure de régulation car il prend en compte l'évolution de la sortie pour calculer la commande (structure bouclée). Tandis que le bloc tracking caractérise la poursuite (en relation directe avec la consigne).

### III.1.2 La Commande 'Optimal Closed Loop Control'

Pour une facilité d'implémentation de l'algorithme, on s'intéresse seulement à l'aspect régulation. L'objectif de la commande est alors la régulation de l'état du système. L'excitation sismique est considérée comme une perturbation extérieure qu'on cherche à compensée. On aboutit alors à l'algorithme 'Closed loop control'. Mathématiquement, ceci peut être réalisé pour le choix suivant de la loi de commande [Had96] :

$$\underline{u}(t) = -\frac{1}{2} \underline{R}^{-1} \underline{B}^T P \underline{z}(t) \quad (\text{III.14.a})$$

Où P est la solution du régime permanent de l'équation de Ricatti.

### III.1.3 La Commande 'Optimal Open Loop Control'

Dans cette approche, la loi de commande dépend seulement de l'excitation sismique. Cette loi est définie par [Yan90][Had96] :

$$\underline{u}(t) = -\frac{1}{2} \underline{R}^{-1} \underline{B}^T \underline{q}(t) \quad (\text{III.14.b})$$

Où  $\underline{q}(t)$  est obtenu par résolution inverse de l'équation différentielle (III.12)

## III.2 Commande à variance minimale

L'objectif de la commande à variance minimale [Ast70][Ast77][Ast90][Lan93] est de minimiser la variance de la sortie autour de la valeur de consigne. Cette valeur de consigne peut être constante (régulation à variance minimale) ou variable (poursuite à variance minimale). La commande à variance minimale est implémentée en deux étapes : une étape de prédiction, où on prédit l'effet de la perturbation sur la sortie, et une étape de commande, où on calcul le signal de commande nécessaire pour ramener la sortie prédite à la valeur désirée. En utilisant le résultat de l'étape de prédiction, un certain critère de performance est minimisé pour élaborer la stratégie de commande à variance minimale [Gue95a][Gue95c][Gue96]. Le système à commander est décrit par le modèle ARMAX (Auto Regressive Moving Average eXogen) défini par :

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + C(q^{-1})e(t) \quad (\text{III.15})$$

où  $A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_nq^{-n}$

$$B(q^{-1}) = b_1q^{-1} + \dots + b_mq^{-m}$$

$$C(q^{-1}) = 1 + c_1q^{-1} + \dots + c_lq^{-l}$$

$d \geq 0$  retard du système

$q^{-1}$  opérateur de retard défini par  $q^{-1}y(t+1) = y(t)$

$y(t)$  sortie du système

$u(t)$  entrée de commande

$e(t)$  bruit blanc de moyenne nulle et de variance  $\sigma^2$

Les polynômes  $B$  et  $C$  sont supposés être stables. Aucune restriction n'est imposée sur le polynôme  $A$  [Ast70][Gue95][Cla75].

Le critère à minimiser est donné par [Cla79][Gue01]:

$$J = E \left\{ \left[ P(q^{-1})y(t+d+1) - R_w(q^{-1})w(t+d+1) \right]^2 \right\} \quad (\text{III.16})$$

où  $E$  opérateur espérance mathématique

$w(t+d+1)$  signal de référence

$P$  et  $R_w$  fraction polynomiale de pondération avec  $P(q^{-1}) = \frac{P_N(q^{-1})}{P_D(q^{-1})}$

On remarque dans le critère (III.16) que  $w(t+d+1)$  est une information disponible, mais  $y(t+d+1)$  est une information future qui doit être prédite. La prédiction optimale au sens des moindres carrés de  $\psi(t+d+1) = Py(t+d+1)$  est donnée dans l'annexe 5 [Gue01]:

$$\psi^*(t+d+1) = \frac{P_N B^*}{P_D A} u(t) + \frac{R}{P_D A} e(t) \quad (\text{III.17})$$

où

$$B^*(q^{-1}) = qB(q^{-1})$$

$R$  est la solution de l'équation Diophantine définie par [Gue01]:

$$AP_D S' + q^{-(d+1)} R = P_N C \quad (\text{III.18})$$

avec

$$S'(q^{-1}) = s'_0 + s'_1 q^{-1} + \dots + s'_d q^{-d}$$

$$R(q^{-1}) = r_0 + r_1 q^{-1} + \dots + r_{n_r} q^{-n_r}$$

$n_r = n + n_d - 1$ , où  $n_d$  est le degré du polynôme  $P_D$

Le polynôme  $C$  est un polynôme caractérisant les pôles en boucle fermée. Tandis que le polynôme  $P_N$  est appelé polynôme d'observation [Ast90][Gue95].

En utilisant l'équation (III.17) dans le critère (III.16), la commande à variance minimale est donnée comme suit (Annexe 6) [Gue01]:

$$u(t) = \frac{P_D C R_w w(t+d+1) - R y(t)}{P_D S} \quad (\text{III.19})$$

avec  $S = S' B^*$ .

Le schéma de la commande à variance minimale est illustré à la figure III.2.

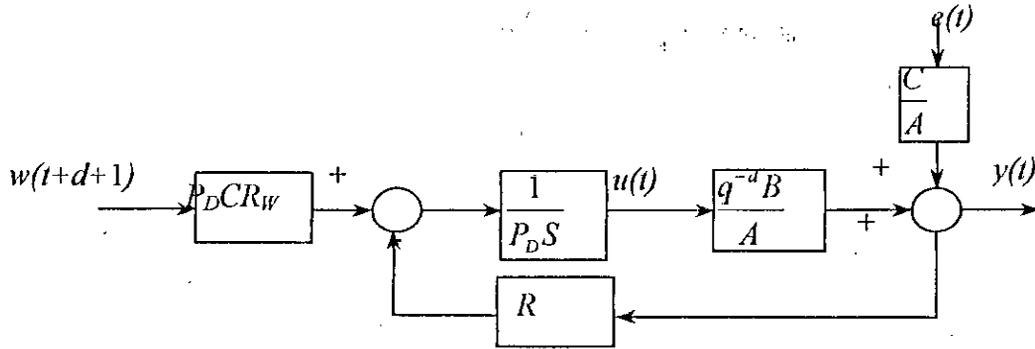


Figure III.2. Structure de la commande à variance minimale

### III.2.1 Analyse des performances en boucle fermée

Pour évaluer les performances de la commande à variance minimale, on doit calculer la fonction de transfert du système en boucle fermée. A partir des équations (III.15), (III.18) et (III.19) on obtient [Gue00][Gue96]:

$$y(t) = \frac{P_D R_w}{P_N} w(t) + \frac{P_D S'}{P_N} e(t) \quad (\text{III.20.a})$$

La différence entre la sortie et le modèle obtenu est un bruit défini par :

$$\xi_1(t) = \frac{P_D S'}{P_N} e(t) \quad (\text{III.20.b})$$

En posant  $P_D'(q^{-1}) = P_D S' = P_{D_0}' + P_{D_1}' q^{-1} + \dots + P_{D_{n_d+d}}' q^{-(n_d+d)}$ , l'équation (III.20.b) devient [Gue95a][Gue01]:

$$\xi_1(t) = -\frac{1}{P_{N_0}} \left[ \sum_{i=1}^{n_N} P_{N_i} \xi_1(t-i) + \sum_{i=0}^{n_d+d} P_{D_i}' e(t-i) \right] \quad (\text{III.21})$$

d'où

$$E[\xi_1^2(t)] \leq \left[ 1 - \frac{1}{P_{N_0}^2} \left( \sum_{i=1}^{n_N} P_{N_i}^2 \right) \right]^{-1} \left( \sum_{i=0}^{n_d+d} P_{D_i}'^2 \sigma^2 \right) \quad (\text{III.22})$$

$$\text{si } 1 - \frac{1}{P_{N_0}^2} \left( \sum_{i=1}^{n_N} P_{N_i}^2 \right) > 0$$

L'équation (III.22) donne une borne maximale de la variance de sortie. Donc les performances en boucle fermée peuvent être améliorées par un choix judicieux des polynômes de pondération  $P$  et  $R_w$ .

#### Remarque

L'inconvénient de cette technique de commande est qu'elle est mal adaptée aux systèmes à phase non minimale. En effet, la loi de commande (III.19) devient

divergente si les racines du polynôme  $B$  sont à l'extérieur du cercle unité [Gue01][Cla84].

### III.3 Commande à variance minimale généralisée

En pratique, la plupart des systèmes physiques présentent un comportement à non minimum de phase (en particulier s'ils sont commandés numériquement). Clarke [CLA81][CLA84] a démontré que l'échantillonnage rapide des systèmes à degré relatif supérieur ou égal à 2, et le choix d'une grande période d'échantillonnage ou d'un temps de retard non entier, résulte en un système à phase non minimale même si celui-ci est à phase minimale dans le cas continu. La stratégie de commande à variance minimale devient non applicable pour ce type de système. De plus, cette dernière peut engendrer des valeurs excessives du signal de commande, car il n'y a pas de contraintes sur la variable de commande dans le critère de performance. Pour palier à ce problème, Clarke [CLA75][CLA79] a introduit la commande à variance minimale généralisée (GMV) pour la commande des systèmes à phase non minimale. C'est une extension de la commande à variance minimale où le critère à minimiser inclut la variable de commande [Gue01].

Le critère à minimiser est :

$$J = E \left[ \left( P(q^{-1})y(t+d+1) - R_w(q^{-1})w(t+d+1) \right)^2 + \left( Q'(q^{-1})u(t) \right)^2 \right] \quad (\text{III.23})$$

où  $P$  et  $R_w$  polynômes de pondération définis précédemment

$$Q'(q^{-1}) = \frac{Q'_N(q^{-1})}{Q'_D(q^{-1})}$$

En suivant les mêmes étapes de la commande à variance minimale, la forme du prédicteur optimal de  $Py(t+d+1)$  est donnée par l'équation (III.17). La loi de commande à GMV est calculée en Annexe 6, elle est donnée par [Gue01].

$$u(t) = \frac{P_D C R_w w(t+d+1) - R y(t)}{P_D (S + Q C)} \quad (\text{III.24})$$

où  $Q(q^{-1}) = \frac{q'_{N_0} P_{D_0}}{q'_{D_0} P_{N_0} b_1} Q'(q^{-1})$ ,  $q'_{N_0}$ ,  $q'_{D_0}$ ,  $P_{N_0}$  et  $P_{D_0}$  sont les premiers coefficients des polynômes  $Q'_N$ ,  $Q'_D$ ,  $P_N$  et  $P_D$ .

Le schéma de la commande à variance minimale généralisée est illustré sur la figure III.3.

#### III.3.1 Analyse des performances en boucle fermée

En calculant la fonction de transfert en boucle fermée, on peut voir que la commande à GMV est bien adaptée aux systèmes à phase non minimale. A partir des équations (III.15) et (III.17), et en tenant compte de l'équation Diophantine, on déduit :

$$y(t) = \frac{B^* P_D R_w}{P_N B^* + A P_D Q} w(t) + \frac{P_D (S + Q C)}{P_N B^* + A P_D Q} e(t) \quad (\text{III.25})$$

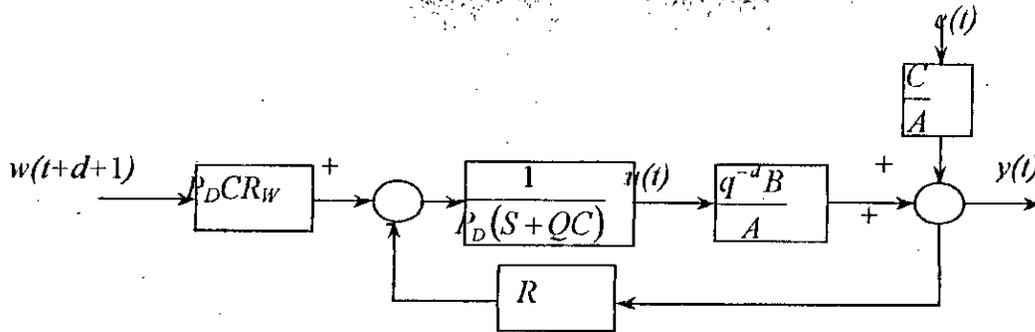


Figure III.3 Structure de la commande à variance minimale généralisée

Les pôles en boucle fermée sont les racines du polynôme  $P_N B^* + A P_D Q$ . Ainsi la dynamique en boucle fermée peut être stable sans pour autant que le polynôme  $B$  soit stable. De plus, la pondération sur la commande dans le critère de performance nous permettra de limiter la commande au domaine des valeurs admissibles.

La dynamique en boucle fermée est spécifiée par le polynôme :

$$P_N B^* + P_D Q A = T_{BF} \quad (\text{III.26})$$

Le système est stable en BF, par un choix adéquat de  $P$  et  $Q$ , même si  $B^*$  est instable. Mei-Hua & Wei [Mei87] expose une méthode de calcul récursif des polynômes  $P$  et  $Q$ , en imposant les pôles en BF ( $T_{BF}$ ) et en résolvant l'équation Diophantine suivante :

$$P B^* + Q A = T_{BF} \quad (\text{III.27})$$

Ainsi le système en boucle fermée devient stable avec une dynamique désirée.

## II.4 Poursuite et Régulation à objectifs indépendants.

Dans cette approche, nous nous intéressons à réaliser deux objectifs: la poursuite et la Régulation [Lan86][Lan93]. Le but de cette commande est d'obtenir un signal d'erreur nul. Cette condition n'est en général pas réalisée du fait d'au moins une des circonstance suivantes :

- La présence de perturbations qui affectent le processus, ce qui nécessite le maintien de la grandeur de sortie à la valeur de consigne: c'est le problème de **régulation**.
- Les variations de la consigne, il s'agit alors d'un problème de **poursuite**.

Ces deux problèmes peuvent être résolus simultanément par le choix d'une stratégie de commande appropriée : c'est la commande par **poursuite et régulation à objectifs indépendants** [Lan93].

L'approche naturelle, dans la génération de la commande est que celle-ci soit une combinaison des composantes de la consigne ( $w(k), w(k-1), \dots$ ) et de la sortie ( $y(k), y(k-1), \dots$ ) ainsi que des commandes précédentes ( $u(k-1), u(k-2), \dots$ ), donc on a :

$$u(k) = \frac{1}{s_0} \left( \sum_{i=1}^s s_i q^{-i} u(t) + \sum_{i=1}^l t_i q^{-i} w(t) + \sum_{i=0}^r r_i q^{-i} y(t) \right) \quad (\text{III.28})$$

La nature du retour impose :  $r_i = -r_i$  et en posant  $s_i = -s_i$ , on a :

$$u(t) = \frac{T(q^{-1})w(t) - R(q^{-1})y(t)}{s(q^{-1})} \quad (\text{III.29})$$

Avec  $T(q^{-1}) = t_0 + t_1 q^{-1} + \dots + t_l q^{-l}$   
 $R(q^{-1}) = r_0 + r_1 q^{-1} + \dots + r_r q^{-r}$   
 $S(q^{-1}) = s_0 + s_1 q^{-1} + \dots + s_s q^{-s}$

Ceci conduit à la structure canonique du régulateur numérique représentée par la figure III.4.

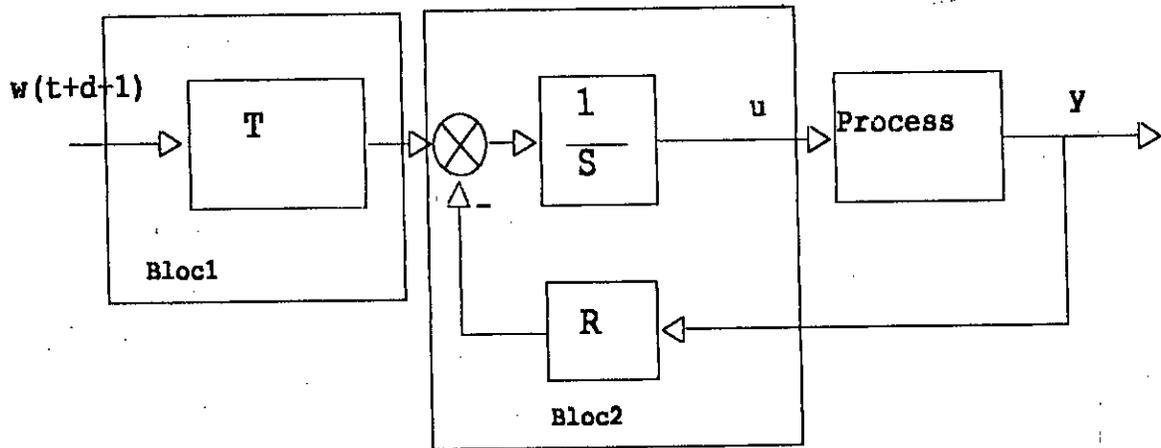


Figure III.4 Structure canonique de réglage RST.

Les polynômes  $R$  et  $S$  caractérisent la régulation, car ils sont dans le bloc de bouclage. Tandis que  $T$  est un polynôme de poursuite (en relation directe avec la consigne).

Pour obtenir de bonnes performances, il suffit de bien choisir  $R$ ,  $S$  et  $T$ , alors le problème qui se pose est la recherche d'une stratégie de commande pour la détermination de ces derniers. Le modèle du processus est un modèle DARMA (Deterministic ARMA), défini par [Goo84]:

$$A(q^{-1})y(t) = q^{-d} B(q^{-1})u(t) + h \quad (\text{III.30})$$

Avec  $A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_n q^{-n}$ .

$B(q^{-1}) = b_1 q^{-1} + \dots + b_m q^{-m}$ .

$d$  : retard du système ( $\geq 0$ ).

$q^{-1}$  : Opérateur retard défini par :  $q^{-1} y(t+1) = y(t)$ .

$h$  : Perturbation constante (déterministe).

$y(t)$  : Sortie du processus.

$u(t)$  : Entrée de commande.

Les équations (III.29) et (III.30) donnent la fonction de transfert en boucle fermée (BF) définie par :

$$y(t) = \frac{q^{-d} B T}{AS + q^{-d} BR} w(t) + \frac{S}{AS + q^{-d} BR} h \quad (\text{III.31})$$

Le bloc 2 de la figure III.4 montre bien la structure de régulation, car il prend en compte l'évolution de la sortie, pour calculer la commande. On choisit alors :

$$A'S + q^{-(d+1)} R = P P_o \quad (\text{III.32})$$

$P$  : spécifie les pôles en boucle fermée.

$P_o$  : polynôme d'observation [Goo84][Ast90].

Les polynômes  $S'$  et  $R$  sont solutions de l'équation Diophantine (III.32) ( $\text{deg } S' = d, \text{deg } R = n - 1$ )

Avec  $R(q^{-1}) = r_0 + r_1 q^{-1} + \dots + r_{n-1} q^{-(n-1)}$ .

$S'(q^{-1}) = s_0 + s_1 q^{-1} + \dots + s_d q^{-d}$ .

Pour assurer une bonne poursuite, il faut que la relation qui lie  $w(t)$  et  $y(t)$  soit avec la dynamique  $P$ , Il faut donc choisir  $w_t = P_o w(t + d + 1)$ , d'où:

$$T(q^{-1}) = P_o(q^{-1}) \quad (\text{III.33})$$

Et la fonction de transfert en BF devient:

$$y(t) = \frac{1}{P} w(t + d + 1) + \frac{S'}{P P_o} h \quad (\text{III.34})$$

Si le système est soumis à une perturbation  $h$  constante, il suffit que  $S'$  soit multiple de  $(1 - q^{-1})$ , pour éliminer son effet. Pour introduire l'effet intégrateur dans le système, on résout:

$$A \Delta S'_r + q^{-(d+1)} R = P P_o \quad (\text{III.35})$$

Où  $S' = S'_r \Delta$  et  $S = B^* S'$ ;  $\Delta = (1 - q^{-1})$ .

Ceci peut se traduire par le modèle **CARIMA**, qui donne le modèle interne des perturbation externe.

Dans ce qui précède, on a supposé que  $B(q^{-1})$  doit être stable. Dans le cas contraire la commande diverge, d'où divergence du système en BF. Notons que des zéros instables peuvent apparaître suite à un échantillonnage trop rapide des systèmes continus ayant une différence entre le numérateur et le dénominateur, de la fonction de transfert, supérieur ou égale à deux [Cla81]. Dans ce cas, on introduit un retour sur le bloc  $1/S$ , pour le stabiliser [Cla84]. La valeur de celui-ci est [Lan88]:

$$Q(q^{-1}) = \frac{\lambda(1 - q^{-1})}{(1 + \alpha q^{-1})}$$

Avec  $\lambda, \alpha$  coefficients réels positifs ( $\alpha < 1$ ).

D'où le schéma de la commande illustré dans la figure III.5

On obtient ainsi l'algorithme de commande par **poursuite et Régulation avec Pondération de l'entrée (PRPE)**. D'où l'expression de la commande :

$$u(t) = \frac{Tw(t + d + 1) - Ry(t)}{S + Q} \quad (\text{III.37})$$

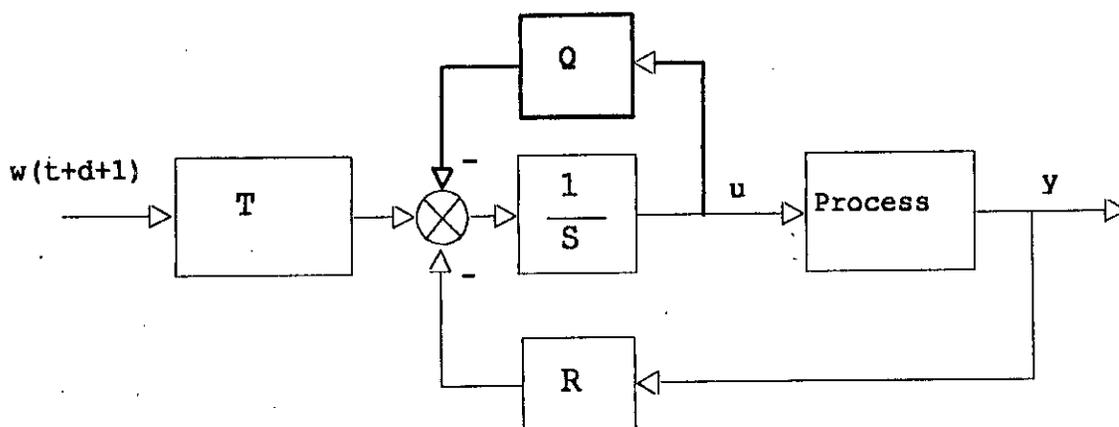


Figure III.5 Structure de réglage par poursuite et Régulation avec Pondération de l'entrée PRPE.

Des équations (III.30), (III.35) et (III.37) on trouve:

$$y(t) = \frac{q^{-(d+1)} B^* P_o}{AQ + B^* PP_o} w(t+d+1) + \frac{S+Q}{AQ + B^* PP_o} h \quad (\text{III.38})$$

Dans ce cas, l'effet de la perturbation constante peut être éliminé en ayant  $S$  et  $Q$  multiple de  $(1-q^{-1})$  (en même temps). Ce qui n'est pas le cas dans l'algorithme à GMV, car l'effet de  $h$  a été introduit dans le calcul de la commande.

### III.5 Algorithme à placement de pôles.

L'idée principale de cet algorithme est concentrée entièrement sur le positionnement des pôles de la fonction de transfert en boucle fermée, en laissant les zéros du processus. Le premier travail de Wellstead [Wel89] se base sur la détermination des paramètres du régulateur de la structure général (canonique) [Lan88], de telle sorte à ce que la fonction de transfert en BF aura les propriétés désirées.

La commande à VM, basée sur un critère quadratique à un pas, est très sensible aux systèmes à phase non minimale, ce qui n'est pas le cas pour cet algorithme. L'insensibilité de ce dernier, par rapport au variation du retard le rend supérieur par rapport aux autres algorithmes de commande adaptative [Wel89]. L'estimation récursive des paramètres du processus engendre deux algorithme de commande adaptative: implicite et explicite.

Le modèle du processus est le modèle DARMA (Deterministic ARMA) [Goo84] pour un système déterministe :

$$Ay(t) = q^{-d} Bu(t) + h. \quad (\text{III.39.a})$$

Où  $A, B$  et  $h$  définis dans l'équation (III.30).

On désire trouver un contrôleur qui assure la stabilité de la fonction de transfert en BF, et un comportement, entre la référence  $w(t)$  et la sortie  $y(t)$ , défini par le modèle :

$$H_m(q^{-1}) = \frac{Q}{P'} \quad (III.39.b)$$

Où  $P'$  et  $Q$  polynômes premier entre eux.

La structure canonique du régulateur définie par l'équation (III.29) donne la fonction de transfert en BF suivante :

$$y(t) = \frac{q^{-d} BT}{AS + q^{-d} BR} w(t) + \frac{S}{AS + q^{-d} BR} h \quad (III.40)$$

Le problème qui se pose alors est la détermination des polynômes  $R, S$  et  $T$  de telle sorte à ce que la fonction de transfert en BF définie par (III.40) soit identique au modèle désiré défini par (III.39) donc :

$$\frac{q^{-d} BT}{AS + q^{-d} BR} = \frac{Q}{P'} \quad (III.41)$$

Du point de vue algébrique, le problème revient à trouver trois polynômes  $R, S$  et  $T$  qui vérifient l'équation (III.41). Le problème ainsi posé possède une infinité de solutions dont le choix de la solution adéquate revient à l'utilisateur.

De l'équation (III.41), on peut dire qu'il y a une simplification d'un facteur, car le degré de  $P$  est normalement inférieur au degré de  $AS + q^{-d} BR$ .

Dans la théorie de l'espace d'état, on peut montrer que la structure de réglage définie par (III.29) correspond à une combinaison d'un observateur et d'un retour d'état [Ast90]. Donc le facteur commun peut être interprété comme le polynôme d'observation  $P_o$ .

Le choix des pôles en BF  $P(q^{-1})$  et le polynôme d'observation donne :

$$AS + q^{-(d+1)} B^* R = P P_o \quad (III.42)$$

Avec  $B^*(q^{-1}) = b_1 + b_2 q^{-1} + \dots + b_m q^{-m+1}$ .

$$P(q^{-1}) = \prod_i (1 + \alpha_i q^{-1}); \alpha_i : \text{pôles désirés [Ast90].}$$

Cette équation est la forme canonique générale de l'équation Diophantine présentée dans l'équation (III.32).

En faisant un choix adéquat des degrés de  $R$  et  $S$  [Lan88]:

$$\text{deg } S = \text{deg } R = r_{\max} - 1$$

$$\text{deg}(P P_o) \leq 2 r_{\max} - 1$$

$$\text{où } r_{\max} = \max(n, m + d).$$

La résolution de l'équation (III.41) donne les polynômes de régulation  $R$  et  $S$ .

De l'équation (III.41) et avec l'hypothèse de laisser les zéros de la fonction de transfert en boucle ouverte, on impose :

$$Q = Q_1 B^* \quad (III.43)$$

Donc

$$T = Q_1 P_o \quad (III.44)$$

Pour avoir une bonne poursuite, il faut que le comportement en régime permanent soit celui de la fonction définie par (III.39.b), d'où le choix du polynôme de poursuite:

$$T(q^{-1}) = \frac{Q_1 P_0}{B^*(1)} \quad \text{si } B^*(1) \neq 0 \quad (\text{III.45.a})$$

$$T(q^{-1}) = Q_1 P_0 \quad \text{si } B^*(1) = 0 \quad (\text{III.45.b})$$

On obtient ainsi la fonction de transfert en boucle fermée :

$$y(t) = q^{-d} \frac{B Q_1}{B^*(1) P} w(t) + \frac{S}{P P_0} h \quad (\text{III.46})$$

La structure de réglage est représentée par le schéma suivant:

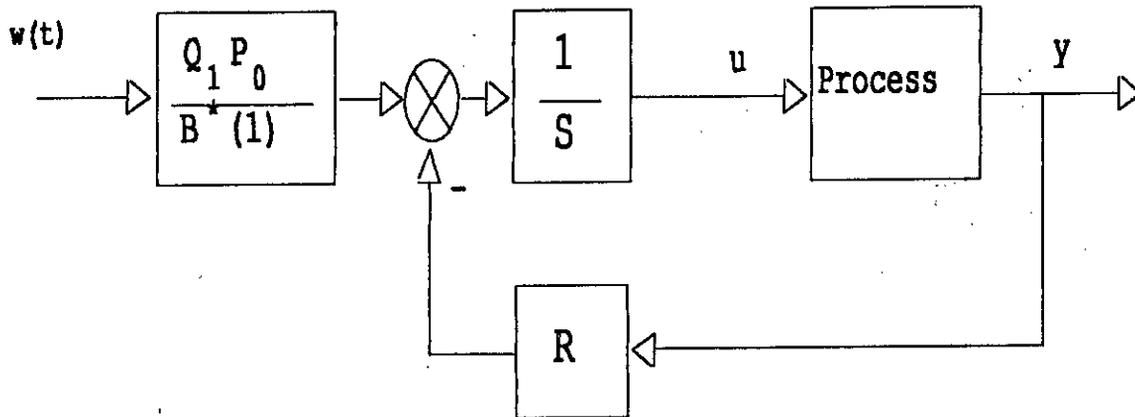


Figure III.6 Structure de réglage à PP.

De l'équation (III.46), on remarque que l'élimination d'une perturbation constante se fait par un choix de  $S$  comme multiple de  $(1 - q^{-1})$  et au lieu de résoudre l'équation (III.42) on résout:

$$A(1 - q^{-1})S + q^{-(d+1)} B^* R = P P_0 \quad (\text{III.47})$$

### III.6 Placement de pôles et de zéros.

L'algorithme à VM développé précédemment donne une dynamique en BF spécifiée par les polynômes  $P$  et  $R$ , quand les zéros du processus sont stables. Lorsque le système est à phase non minimale, l'utilisation de  $Q$ , dans l'algorithme à GMV, stabilise le système en BF, mais aboutit à une dynamique spécifiée par  $PB + QA$  [Cla75][Cla79][Gaw77]. Cette dynamique ne peut être caractérisée par l'utilisateur d'une façon désirée. L'algorithme de Wellstead spécifie seulement les pôles du système en BF [Wel89]. Tandis que l'algorithme d'Astrom [Ast80] diffère de ces derniers, par la simplification des zéros du système qui se situent dans une région de stabilité acceptable [Ast90].

Puthempura [Put87] propose un algorithme qui simplifie non pas les zéros du système, mais les zéros du signal d'erreur pour une entrée de référence donnée.

Le modèle du processus est le modèle DARMA défini par l'équation (III.30). On désire spécifier les pôles de la fonction de transfert en BF et choisir le modèle de référence entre  $w(t)$  et la sortie  $y(t)$ . Le modèle est défini par l'équation (III.39.b). La fonction de transfert en BF est définie par (III.40). Le problème qui se pose est le même que celui à PP défini par l'équation suivante:

$$\frac{q^{-(d+1)} B^* T}{AS + q^{-(d+1)} B^* R} = \frac{Q}{P'} \quad (\text{III.48})$$

Où  $B^*(q^{-1}) = b_1 + b_2 q^{-1} + \dots + b_m q^{-m+1}$ .

L'idée de base de cet algorithme est la suivante:

*De l'équation (III.48), on remarque que tout facteur de  $B^*$ , non diviseur de  $Q$ , doit diviser  $S$ . Or les facteurs de  $B^*$  correspondent aux zéros du système en boucle ouverte. Donc on souhaite simplifier ses facteurs [Ast80].*

On factorise  $B^*(q^{-1})$  de la façon suivante :

$$B^*(q^{-1}) = B^+(q^{-1}) B^-(q^{-1}) \quad (\text{III.49})$$

Où les zéros de  $B^+$  sont dans la région de stabilité  $\underline{Z}$  du plan complexe et les zéros de  $B^-$  à l'extérieur de cette région. La région  $\underline{Z}$  correspond aux modes suffisamment amorties. Donc tous les zéros de  $B^+$  correspondent aux modes bien amorties et les zéros de  $B^-$  correspondent aux modes instables ou mal amorties. La condition nécessaire pour la stabilité du problème posé par (III.48) et (III.49) est :

$$Q = Q_1 B^- \quad (\text{III.50})$$

En remplaçant (III.49) et (III.50) dans (III.48) on obtient :

$$\frac{q^{-(d+1)} T B^+ B^-}{B^+ (AS_1 + q^{-(d+1)} B^- R)} = \frac{Q_1 B^-}{P'} \quad (\text{III.51})$$

Or  $B^+$  est stable (Dans la région  $\underline{Z}$ ). L'équation (III.48) devient :

$$\frac{q^{-(d+1)} T B^-}{AS_1 + q^{-(d+1)} B^- R} = \frac{Q_1 B^-}{P'} \quad (\text{III.52})$$

Où  $S = S_1 B^+$ .

Avec un choix des pôles en BF ( $P$ ) et le polynôme d'observation ( $P_o$ ) (défini précédemment), l'équation (III.52) donne:

$$AS_1 + q^{-(d+1)} B^- R = PP_o \quad (\text{III.53})$$

où:  $\deg S = \deg R = r_{\max} - 1; r_{\max} = \max(n, m + d)$ .

$$\deg S_1 = \deg S - \deg B^+$$

La résolution de cette équation donne  $S_1$  et  $R$ , d'où l'on tire :

$$S = S_1 B^+ \quad (\text{III.54})$$

Pour le calcul de  $T$ , on utilise l'équation (III.52) et le polynôme d'observation  $P_o$ , d'où:

$$T = Q_1 P_o \quad (\text{III.55})$$

Pour avoir une bonne poursuite, il faut que le comportement en régime permanent soit celui du modèle défini par (III.39.b), d'où le choix du polynôme de poursuite :

$$T(q^{-1}) = \frac{Q_1 P_o}{B^-(1)} \quad \text{si } B^-(1) \neq 0 \quad (\text{III.55.a})$$

$$T(q^{-1}) = Q_1 P_o \quad \text{si } B^-(1) = 0 \quad (\text{III.55.b})$$

D'où la fonction de transfert en BF:

$$y(t) = q^{-(d+1)} \frac{B^- Q_1}{B^-(1)P} w(t) + \frac{S_1}{PP_o} h \quad (III.56)$$

La structure de réglage est représentée par la figure III.7.

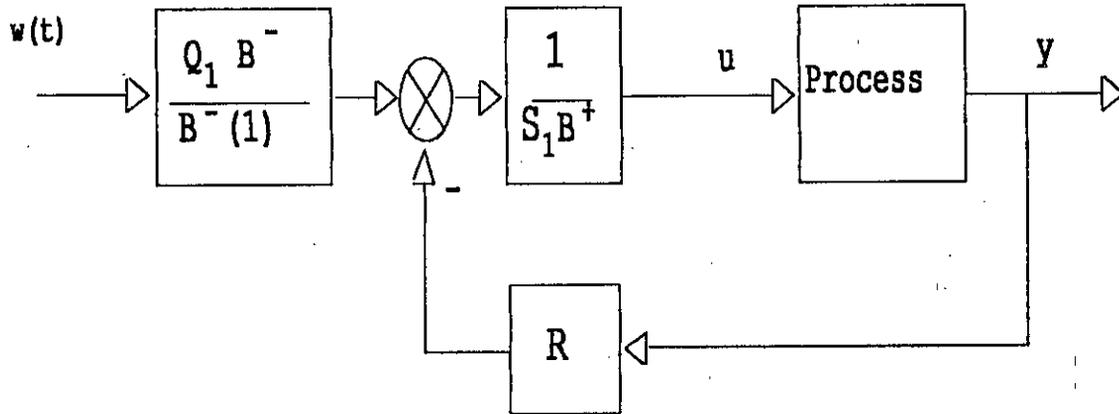


Figure III.7 Structure de réglage à PP et de zéros.

De l'équation (III.56), l'élimination de l'effet d'une perturbation constante se fait par un choix de  $S_1$  multiple de  $(1 - q^{-1})$ .

### III.7 Commande linéaire par poursuite d'un modèle. (Linear Model Following Control).

Dans cette partie, on s'intéresse à développer un contrôleur qui assure une poursuite parfaite d'un modèle de référence. La figure III.8 illustre la structure d'un tel contrôleur.

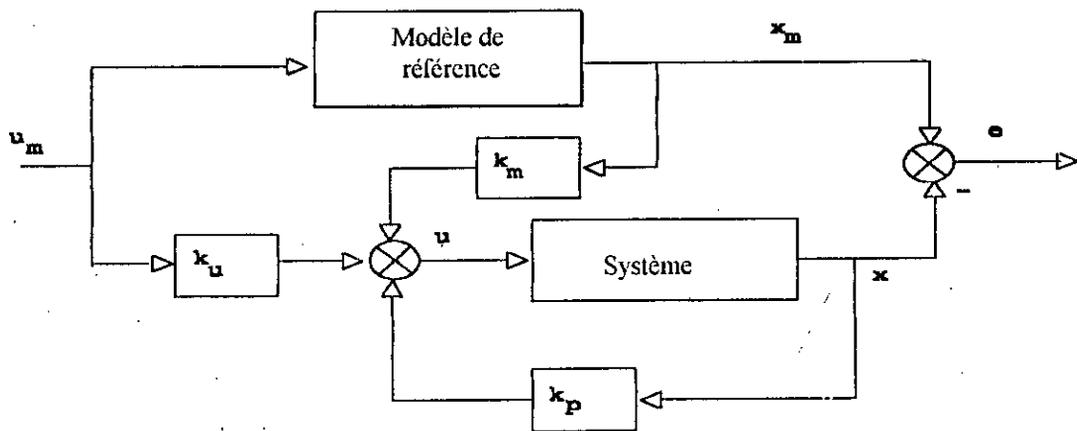


Figure III.8 Commande linéaire par poursuite d'un modèle LMFC.

La structure de ce régulateur est similaire à celle d'un retour d'état. La loi de commande est générée à partir d'un retour (feedback) linéaire et une action directe (feedforward) de  $u_m$  et de la référence  $x_m$ . L'objectif est de minimiser l'erreur d'état  $e$ .

Le modèle de référence est:

$$\dot{x}_m = A_m x_m + B_m u_m \quad (III.57)$$

$A_m$  : matrice d'Hurwitz [Lan79].

Le système est défini par:

$$\dot{x} = Ax + Bu \quad (III.58)$$

Et la commande est:

$$u = -k_p x + k_m \dot{x}_m + k_u u_m \quad (III.59)$$

Où  $K_p, k_m$  et  $k_u$  : matrices de dimensions appropriées.

les pairs  $(A, B)$  et  $(A_m, B_m)$  sont stabilisables.

On définit l'erreur d'état comme suit :

$$e = x_m - x \quad (III.60)$$

Pour assurer une poursuite parfaite, la dynamique de l'erreur doit être imposée. En utilisant les équations (III.57), (III.58) et (III.59), cette dynamique est définie par:

$$\dot{e} = (A_m - B k_m)e + [A_m - A + B(k_p - k_m)]x + (B_m - B k_u)u_m \quad (III.61)$$

Ainsi il suffit d'imposer la condition suivante:

$$[A_m - A + B(k_p - k_m)]x + (B_m - B k_u)u_m = 0 \quad (III.62)$$

Où  $x \in R^n, u_m \in C^m$  ( $C^m$  : espace des commande admissible).

Dans le cas où  $x$  et  $u_m$  sont indépendants, on a:

$$B(k_p - k_m) = A - A_m \quad (III.63)$$

$$k_m - k_p = \frac{B k_u}{B} (A_m - A) \quad (III.64)$$

$$\text{rang}[B] = \text{rang}[B \ A_m - A] = \text{rang}[B \ B_m] \quad (III.65)$$

$$\begin{aligned} (I - BB^+)(A_m - A) &= 0 \\ (I - BB^+)B_m &= 0 \end{aligned} \quad (III.66)$$

De cette façon, la dynamique de l'erreur est définie par la matrice  $A_m - B k_m$ , qui doit être une matrice de Hurwitz. Des équations (III.63) on aboutit à:

Où  $B^{+-} = (B^T B)^{-1} B^T$  : la pseudo inverse de  $B$ .

Cette pseudo inverse existe dans le cas où:

En introduisant les expressions de  $k_m - k_p$  et de  $k_u$  dans (III.63), on aboutit à:

Ces conditions sont connues sous le nom de **conditions d'Erzgerger** [Lan79]. Une fois ces conditions sont satisfaites, la poursuite du modèle défini par (III.57) à la référence  $x_m$  est assurée.

### III.8 Approche multivariable pour les structures articulées.

Le choix d'un modèle découplé est lié aux nombre de sorties et d'entrées du système, qui doivent être égales. Dans le cas de la robotique, cette condition est vérifiée. Le modèle utilisé est celui présenté dans le chapitre I, avec une augmentation du modèle de perturbation dans le cas stochastique.

Dans ce qui suit, nous présentons les versions multivariables des algorithmes synthétisés dans les parties précédentes. Dans le cas de notre application, le robot possède trois entrées et

trois sorties. Nous présentons seulement l'algorithme à VM. Les autres algorithmes s'en découlent de la même manière. L'algorithme multirate sera présenté en dernier lieu.

### III.8.1 Commande à GMV avec découplage du modèle de la structure

Le choix d'un modèle de représentation découplé permet la généralisation des algorithmes monovariables au cas multivariable d'une manière plus simple. Pour les structures, un tel modèle permet de commander chaque étage indépendamment des autres et avec des algorithmes de commande et des périodes d'échantillonnage différentes. L'intérêt de cette approche par rapport à l'approche précédente est la structure décentralisée de la commande ainsi qu'un choix des algorithmes adéquats pour chaque sous-système.

Pour l'approche qu'on présentera dans cette partie, un modèle de représentation découplé de la structure est déterminée par une procédure d'identification récursive. Puis, l'algorithme à GMV monovariable est appliqué à chaque étage. Le schéma de commande est représenté sur la figure III.9. On se limitera dans cette partie à présenter le modèle de représentation et l'algorithme d'identification utilisé. L'algorithme de commande étant déjà présenté dans la section III.3.

Il existe plusieurs structure de représentation pour les systèmes multivariables [GUE95b], à savoir : la représentation Pleine - Pleine (PP), la représentation Diagonale - Pleine (DP) et la représentation Diagonale - Diagonale (DD). Dans notre cas, on s'intéresse à la représentation Diagonale - Diagonale, car elle nous permet de réaliser le découplage et de modéliser les interactions comme étant des perturbations extérieures. Le modèle est le suivant [Lan86].

$$\underline{A}(q^{-1})\underline{y}(t) = q^{-d}\underline{B}(q^{-1})\underline{u}(t) + \underline{C}(q^{-1})\underline{e}(t) \quad (III.67)$$

où

$\underline{y}(t)$  vecteur de sorties de dimension  $n$

$\underline{u}(t)$  vecteur de commandes de dimension  $n$

$\underline{e}(t)$  vecteur de bruit blanc de moyenne nulle et de covariance  $E[\underline{e}(t)\underline{e}^T(t)] = r$ , de dimension  $n$

$\underline{A}(q^{-1})$ ,  $\underline{B}(q^{-1})$  et  $\underline{C}(q^{-1})$  sont des matrices polynomiales de dimension  $n \times n$ , définies par

$$\underline{A}(q^{-1}) = \begin{bmatrix} A_1(q^{-1}) & & 0 \\ & \ddots & \\ 0 & & A_p(q^{-1}) \end{bmatrix}, \underline{B}(q^{-1}) = \begin{bmatrix} B_1(q^{-1}) & & 0 \\ & \ddots & \\ 0 & & B_p(q^{-1}) \end{bmatrix},$$

$$\underline{C}(q^{-1}) = \begin{bmatrix} C_1(q^{-1}) & & 0 \\ & \ddots & \\ 0 & & C_p(q^{-1}) \end{bmatrix}$$

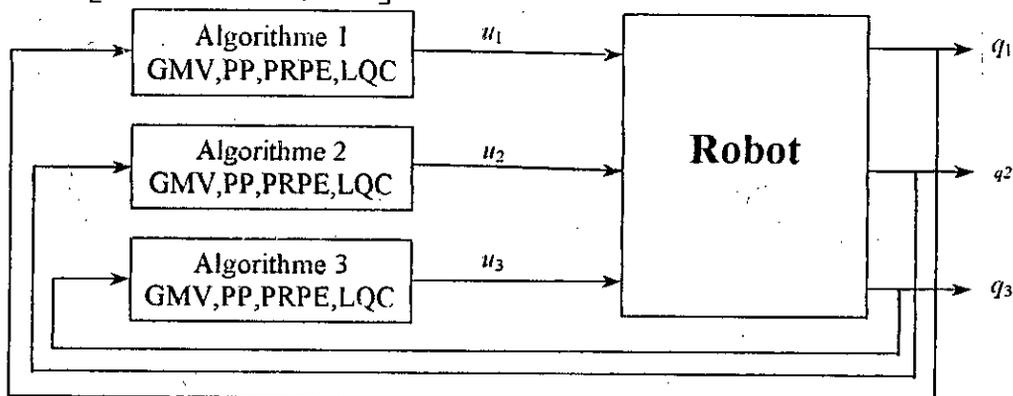


Figure III.9 Structure de la commande avec modèle découplé pour une structure à trois degrés de liberté

où  $A_i(q^{-1}) = 1 + a_{i1}q^{-1} + \dots + a_{im_i}q^{-m_i}$ ,  $i = \overline{1, n}$

$B_i(q^{-1}) = q^{-d_i}(b_{i1}q^{-1} + \dots + b_{im_i}q^{-m_i})$ ,  $i = \overline{1, n}$  et  $d_i \geq 0$

$C_i(q^{-1}) = 1 + c_{i1}q^{-1} + \dots + c_{il_i}q^{-l_i}$ ,  $i = \overline{1, n}$

$n$  est le nombre de degrés de liberté de la structure

La détermination de ce modèle nécessite une identification en temps réel. Dans le cas des structures complexes, l'algorithme à MCR, s'avère lent, on est amené à utiliser d'autres algorithmes "rapide". L'algorithme RMS (Root Mean Square algorithm) présenté dans [Ast80][Cha87], procède par une factorisation du gain d'adaptation et accélère la vitesse de calcul, donc la convergence. Astrom [Ast80] présente un algorithme à facteur d'oubli variable. Ce facteur est exprimé en fonction de l'erreur, pour pouvoir réactualiser les paramètres estimés chaque fois que ceux du système varient. Goodwin [Goo84] présente un algorithme sous le nom "Multiple récursion algorithm", dans lequel la matrice du gain d'adaptation se réduit à un scalaire, similaire à l'algorithme de projection [Goo81][Goo84].

L'algorithme des moindres carrés étendus (MCE) à trace constante est utilisé pour identifier la  $i^{\text{ème}}$  articulation de la structure. L'algorithme est donné par les équations suivantes [Lan93][Dio86] [Gue95].

$$\hat{\theta}_i(t+1) = \hat{\theta}_i(t) + D_i(t)\phi_i(t)\varepsilon_i(t+1) \quad (\text{III.68.a})$$

$$\varepsilon_i(t+1) = \frac{y_i(t+1) - \hat{\theta}_i^T(t)\phi_i(t)}{\alpha_i + \phi_i^T(t)D_i(t)\phi_i(t)} \quad (\text{III.68.b})$$

$$D_i(t+1) = \frac{1}{\lambda_{1i}(t)} \left[ D_i(t) - \frac{D_i(t)\phi_i(t)\phi_i^T(t)D_i(t)}{\alpha_i + \phi_i^T(t)D_i(t)\phi_i(t)} \right] \quad (\text{III.68.c})$$

où  $\hat{\theta}_i(t) = [\hat{a}_{i1}(t) \dots \hat{a}_{im_i}(t) \hat{b}_{i1}(t) \dots \hat{b}_{im_i}(t) \hat{c}_{i1}(t) \dots \hat{c}_{il_i}(t)]^T$  vecteur des paramètres estimés

$\phi_i(t) = [-y_i(t) \dots -y_i(t-n_i+1) u_i(t) \dots u_i(t-m_i+1) \varepsilon_i(t) \dots \varepsilon_i(t-l_i+1)]^T$  vecteur des observations

$\alpha_i = \frac{\lambda_{1i}(t)}{\lambda_{2i}(t)} = \text{cte}$ ,  $\lambda_{1i}(t)$  et  $\lambda_{2i}(t)$  sont les facteurs d'oubli.

$\lambda_{1i}(t)$  et  $\lambda_{2i}(t)$  sont calculés à chaque itération de la phase d'identification de façon à maintenir une trace constante du gain d'adaptation  $D_i$  [Gue95a].

$$\text{tr}[D_i(t+1)] = \frac{1}{\lambda_{1i}(t)} \text{tr} \left[ D_i(t) - \frac{D_i(t)\phi_i(t)\phi_i^T(t)D_i(t)}{\alpha_i + \phi_i^T(t)D_i(t)\phi_i(t)} \right] = \text{tr}[D_i(t)] \quad (\text{III.68.d})$$

Sur la base du modèle découplé obtenu, une commande à variance minimale généralisée monovariante est synthétisée puis appliquée à chaque liaison de la structure selon le schéma de commande donné par la figure III.9.

### III.8.2 Commande multivariable multirate et mixte.

Le choix de la structure d'un modèle de représentation découplé permet la commande du robot avec différents algorithmes. En effet chaque liaison peut être commandée avec un contrôleur différent de celui commandant la liaison voisine. Ce qui n'est pas le cas si le choix d'un modèle de représentation est Plein\_Plein [Gue95d]. On aboutit alors à une commande multivariable mixte, dans laquelle on peut choisir pour chaque entrée-sortie une structure de commande adéquate. La structure de réglage est représentée par la figure III.9. Chaque bloc peut être différent de l'autre. Ainsi, suivant le comportement dynamique de chaque liaison, on choisit la méthode de commande adaptative correspondante.

Dans le cas où les temps de réponse des différentes liaisons sont différents d'une manière importante. Le choix de la période d'échantillonnage s'effectue d'une manière indépendante, pour chaque entrée-sortie. Tham [Tha81] et Morris [Mor81] présentent l'algorithme multirate appliqué à une colonne à distiller. Tandis que Montague [Mon85] propose une comparaison entre les algorithmes de commande adaptative multivariables, ainsi que la commande multirate.

L'implémentation de l'algorithme multirate dans le cas adaptative nécessite le choix d'une période d'échantillonnage de base  $T_b$  [Mor81]. En plus de cette condition, toutes les périodes d'échantillonnage de chaque entrée-sortie, sont choisies comme un multiple entier de celle de base.

En faisant le couplage de la commande mixte et multirate, on obtient la commande multivariable mixte et multirate. Cet algorithme donne une flexibilité du choix de la stratégie de commande de chaque entrée-sortie, indépendamment des autres entrées-sorties. Ce qui offre la possibilité de choisir des algorithmes de commande et des périodes d'échantillonnage en tenant compte des capacités de calcul et de la rapidité du processeur numérique utilisé.

## III.9 Approches multivariables pour les structures en génie civil

Les structures à plusieurs degrés de liberté peuvent être considérées comme des systèmes complexes composés de plusieurs sous-systèmes interconnectés. Les sous-systèmes dans ce cas sont les différents étages de la structure ou du bâtiment. Par conséquent, les techniques de commande initialement conçues pour les structures à un seul degré de liberté ne sont pas directement applicables aux structures à plusieurs degrés de liberté. Les effets des interconnexions entre les étages doivent être considérés lors de l'élaboration de la commande. La structure sera considérée comme un système multivariable, ou un ensemble de systèmes monovariants interconnectés. Par conséquent, les modèles sur la base desquels la commande sera développée seront différents [Spa96].

Le choix d'un modèle découplé est lié au nombre de sorties et d'entrées du système, qui doivent être égales. Dans le cas de la structure, cette condition est vérifiée. Le modèle utilisé est celui présenté dans le chapitre I, avec une augmentation du modèle de perturbation dans le cas stochastique.

Dans ce qui suit, nous nous intéressons au développement de deux techniques de commande pour les structures à plusieurs degrés de liberté. On présentera pour chacune, le modèle ARMAX utilisé pour synthétiser la loi de commande.

### III.10.1 Commande à variance minimale généralisée multivariable

Le système à commander est décrit par un vecteur linéaire d'équations au différence (modèle ARMAX multivariable)

$$\underline{A}(q^{-1})\underline{y}(t) = q^{-d}\underline{B}(q^{-1})\underline{u}(t) + \underline{C}(q^{-1})\underline{e}(t) \quad (\text{III.69})$$

où

$\underline{y}(t)$  vecteur de sortie de dimension  $n$

$\underline{u}(t)$  vecteur d'entrée de dimension  $n$

$\underline{e}(t)$  vecteur de bruit blanc de moyenne nulle et de covariance  $E[\underline{e}(t)\underline{e}^T(t)] = r$ , de dimension  $p$

$d \geq 0$  est le retard du système

$\underline{A}(q^{-1}) = I + A_1q^{-1} + \dots + A_nq^{-n}$ , matrice de dimension  $n \times n$ ,  $I$  est la matrice identité

$\underline{B}(q^{-1}) = B_1q^{-1} + \dots + B_nq^{-n}$ , matrice de dimension  $n \times n$ ,  $B_1$  est inversible

$\underline{C}(q^{-1}) = I + C_1q^{-1} + \dots + C_nq^{-n}$ , matrice de dimension  $n \times n$

$A_i$ ,  $B_i$  et  $C_i$ , matrices de dimension  $n \times n$

Le matrice polynomiale  $\underline{C}$  est stable.

On peut résumer les hypothèses sur le système comme suit :

Le nombre d'entrées est égal au nombre de sorties

$B_1$  est inversible

$\underline{C}$  est stable

Le critère à minimiser est [Goo81b] :

$$J = E \left[ \left\| \underline{P}\underline{y}(t+d+1) - \underline{R}_w\underline{w}(t+d+1) \right\|^2 + \left\| \underline{Q}'\underline{u}(t) \right\|^2 \right] \quad (\text{III.70})$$

où  $E$  espérance mathématique

$\underline{w}(t+d+1)$  est le vecteur de dimension  $n \times 1$  des signaux de référence

$\underline{P}$ ,  $\underline{R}_w$  et  $\underline{Q}'$  sont des matrices polynomiales de pondération de dimension  $n \times n$

$\| \cdot \|$  Norme vectorielle définie par  $\|V\|^2 = V^T V$ .

Comme dans le cas monovariante, on doit d'abord trouver une prédiction optimale  $\varphi^*(t+d+1)$  de  $\varphi(t+d+1) = \underline{P}\underline{y}(t+d+1)$  car c'est une information future. Cette prédiction est donnée par la formule suivante (voir Annexe 7)

$$\varphi^*(t+d+1) = \tilde{C}^{-1}(q^{-1}) \left[ \tilde{F}(q^{-1})\underline{B}^*(q^{-1})\underline{u}(t) + \tilde{G}(q^{-1})\underline{y}(t) \right] \quad (\text{III.71})$$

où  $\tilde{C}$ ,  $\tilde{F}$ ,  $\tilde{G}$  sont des matrices polynomiales définies en annexe 7, et  $\underline{B}^*(q^{-1}) = q\underline{B}(q^{-1})$ .

En utilisant les équations (III.28) et (III.30), on démontre que la stratégie de commande est donnée par (voir Annexe 8) [Goo81b]

$$\underline{H}(q^{-1})\underline{u}(t) = \tilde{C}(q^{-1})\underline{R}_w(q^{-1})\underline{w}(t+d+1) - \tilde{G}(q^{-1})\underline{y}(t) \quad (\text{III.72})$$

avec  $\underline{H}(q^{-1}) = \tilde{F}(q^{-1})\underline{B}^*(q^{-1}) + \tilde{C}(q^{-1})\underline{Q}(q^{-1})$  et  $\underline{Q}(q^{-1}) = \left[ (P_0 B_1)^T \right]^{-1} Q'(0)^T Q'(q^{-1})$ .

Afin de pouvoir implanter la GMV multivariable, il est nécessaire de déterminer le modèle ARMAX multivariable du système à commander. Pour une structure à plusieurs degrés de liberté, l'élaboration de ce modèle est décrite dans ce qui suit.

### III.9.2 Modèle ARMAX multivariable de la structure

On s'intéresse dans cette section à déterminer le modèle ARMAX multivariable d'une structure à plusieurs degrés de liberté afin de pouvoir appliquer la GMV multivariable présentée ci-dessus. Ceci peut être effectué en utilisant l'équation dynamique du mouvement de la structure.

Considérant l'équation (I.21) qu'on peut réécrire sous la forme suivante

$$m_i \ddot{x}_i + (c_i + c_{i+1}) \dot{x}_i + (k_i + k_{i+1}) x_i - c_i \dot{x}_{i-1} - k_i x_{i-1} - c_{i+1} \dot{x}_{i+1} - k_{i+1} x_{i+1} = u_i - u_{i+1} - m_i \ddot{x}_g \quad (III.73)$$

$i = \overline{1, n}$

Par application de la transformée de Laplace à cette équation

$$\left( s^2 + \frac{c_i + c_{i+1}}{m_i} s + \frac{k_i + k_{i+1}}{m_i} \right) X_i(s) - \left( \frac{c_i}{m_i} s + \frac{k_i}{m_i} \right) X_{i-1}(s) - \left( \frac{c_{i+1}}{m_i} s + \frac{k_{i+1}}{m_i} \right) X_{i+1}(s) = \frac{1}{m_i} U_i(s) - \frac{1}{m_i} U_{i+1}(s) - \ddot{X}_g(s), \quad i = \overline{1, n} \quad (III.74)$$

Ainsi, on peut déduire le modèle ARMAX continu donné par

$$\overline{A}(s) \underline{Y}(s) = \overline{B}(s) \underline{U}(s) + \overline{C}(s) I_v \ddot{X}_g(s) \quad (III.75)$$

$$\text{où } \overline{A}(s) = \begin{bmatrix} \overline{A}_{11}(s) & \overline{A}_{12}(s) & & 0 \\ \overline{A}_{21}(s) & \overline{A}_{22}(s) & \overline{A}_{23}(s) & \\ & & \ddots & \\ 0 & & \overline{A}_{n(n-1)}(s) & \overline{A}_{nn}(s) \end{bmatrix} = \text{Diag}(A_{i-1}, A_i, A_{i-1}),$$

avec  $A_{10}=0$  et  $A_{nn+1}=0$ .

$$\overline{B}(s) = \begin{bmatrix} \overline{B}_{11}(s) & \overline{B}_{12}(s) & & 0 \\ & \overline{B}_{22}(s) & \overline{B}_{23}(s) & \\ & & \ddots & \\ 0 & & 0 & \overline{B}_{nn}(s) \end{bmatrix} = \text{Diag}(B_{ii}, B_{i+1}), \quad \overline{C}(s) = \begin{bmatrix} \overline{C}_{11}(s) & & 0 \\ & \ddots & \\ 0 & & \overline{C}_{nn}(s) \end{bmatrix},$$

avec  $B_{nn+1}=0$ .

$$\underline{Y}(s) = [X_1(s) \quad \dots \quad X_n(s)]^T, \quad \underline{U}(s) = [U_1(s) \quad \dots \quad U_n(s)]^T$$

$$\text{avec } \overline{A}_{ii}(s) = s^2 + \frac{c_i + c_{i+1}}{m_i} s + \frac{k_i + k_{i+1}}{m_i}, \quad \overline{A}_{i(i-1)}(s) = -\left( \frac{c_i}{m_i} s + \frac{k_i}{m_i} \right), \quad \overline{A}_{i(i+1)}(s) = -\left( \frac{c_{i+1}}{m_i} s + \frac{k_{i+1}}{m_i} \right)$$

$$\overline{B}_{ii}(s) = \frac{1}{m_i}, \quad \overline{B}_{i(i+1)}(s) = -\frac{1}{m_i}, \quad \overline{C}_{ii}(s) = -1, \quad I_v = [1 \quad 1 \quad \dots \quad 1]^T \text{ de dimension } n.$$

Pour obtenir le modèle ARMAX discret, on utilise l'équation (III.74) réécrite sous la forme suivante :

$$X_i(s) + \frac{\overline{A}_{i(i-1)}(s)}{\overline{A}_{ii}(s)} X_{i-1}(s) + \frac{\overline{A}_{i(i+1)}(s)}{\overline{A}_{ii}(s)} X_{i+1}(s) = \frac{\overline{B}_{ii}(s)}{\overline{A}_{ii}(s)} U_i(s) + \frac{\overline{B}_{i(i+1)}(s)}{\overline{A}_{ii}(s)} U_{i+1}(s) + \frac{\overline{C}_{ii}(s)}{\overline{A}_{ii}(s)} \ddot{X}_g(s) \quad (III.76)$$

En utilisant les méthodes de discrétisation, on déduit la forme discrète de l'équation (III.76)

$$x_i(t) + \frac{A_{i(i-1)}(q^{-1})}{A_{ii}(q^{-1})} x_{i-1}(t) + \frac{A_{i(i+1)}(q^{-1})}{A_{ii}(q^{-1})} x_{i+1}(t) = \frac{B_{ii}(q^{-1})}{A_{ii}(q^{-1})} u_i(t) + \frac{B_{i(i+1)}(q^{-1})}{A_{ii}(q^{-1})} u_{i+1}(t) + \frac{C_{ii}(q^{-1})}{A_{ii}(q^{-1})} \ddot{x}_g(t) \quad (\text{III.77})$$

Le modèle ARMAX discret multivariable est ainsi obtenu, il est donné par

$$\underline{A}(q^{-1})\underline{Y}(t) = \underline{B}(q^{-1})\underline{U}(t) + \underline{C}(q^{-1})I_v \ddot{x}_g(t) \quad (\text{III.78})$$

$$\text{où } \underline{A}(q^{-1}) = \begin{bmatrix} A_{11}(q^{-1}) & A_{12}(q^{-1}) & & 0 \\ A_{21}(q^{-1}) & A_{22}(q^{-1}) & A_{23}(q^{-1}) & \\ & & \ddots & \\ 0 & & A_{n(n-1)}(q^{-1}) & A_{nn}(q^{-1}) \end{bmatrix},$$

$$\underline{B}(q^{-1}) = \begin{bmatrix} B_{11}(q^{-1}) & B_{12}(q^{-1}) & & 0 \\ & B_{22}(q^{-1}) & B_{23}(q^{-1}) & \\ & & \ddots & \\ 0 & & & B_{nn}(q^{-1}) \end{bmatrix}, \quad \underline{C}(q^{-1}) = \begin{bmatrix} C_{11}(q^{-1}) & & 0 \\ & \ddots & \\ 0 & & C_{nn}(q^{-1}) \end{bmatrix}$$

### III.9.3 Commande à GMV avec modèle empirique

La technique que nous allons développer ici se base essentiellement sur une reparamétrisation du modèle de la structure à plusieurs degrés de liberté pour obtenir un modèle ayant la forme découplée où les termes de couplage sont considérés comme des perturbations extérieures. Sur la base de ce nouveau modèle, une commande à GMV monovariable est développée puis appliquée à chaque sous-système ou étage.

Pour une structure à plusieurs degrés de liberté, l'équation de mouvement du  $i^{\text{ème}}$  étage est (voir chap. I)

$$m_i \ddot{x}_i + c_i (\dot{x}_i - \dot{x}_{i-1}) + k_i (x_i - x_{i-1}) - c_{i+1} (\dot{x}_{i+1} - \dot{x}_i) - k_{i+1} (x_{i+1} - x_i) = u_i - u_{i+1} - m_i \ddot{x}_g \quad (\text{III.79})$$

Cette équation peut être réécrite sous la forme suivante :

$$m_i \ddot{x}_i + (c_i + c_{i+1}) \dot{x}_i + (k_i + k_{i+1}) x_i = u_i - u_{i+1} - m_i \ddot{x}_g + c_i \dot{x}_{i-1} + c_{i+1} \dot{x}_{i+1} + k_i x_{i-1} + k_{i+1} x_{i+1} \quad (\text{III.80})$$

En introduisant de nouvelles notations, l'équation précédente devient :

$$m_i^* \ddot{x}_i + c_i^* \dot{x}_i + k_i^* x_i = u_i^* - m_i^* \ddot{x}_g + \phi_i^* \quad (\text{III.81})$$

où  $m_i^* = m_i$  ;  $c_i^* = c_i + c_{i+1}$  ;  $k_i^* = k_i + k_{i+1}$

$u_i^* = u_i - u_{i+1}$ , force de commande effective appliquée au  $i^{\text{ème}}$  étage

$\phi_i^* = c_i \dot{x}_{i-1} + c_{i+1} \dot{x}_{i+1} + k_i x_{i-1} + k_{i+1} x_{i+1}$ , contient tous les termes de couplage entre l'étage  $i$  et les autres étages.

L'équation (III.81) peut être interprétée comme étant un nouveau système découplé avec de nouveaux paramètres, une nouvelle variable de commande et un terme de perturbation. On peut ainsi déterminer le modèle ARMAX pour chaque sous-système défini par l'équation (III.81) (voir la section 1.5 pour la détermination du modèle ARMAX), puis appliquer l'algorithme de la GMV pour générer la commande  $u_i^*$  (figure III.10). Les commande  $u_i$  sont ensuite obtenues par une transformation inverse. Pour que cette technique soit efficace, la commande doit compenser les perturbations représentées par le terme  $\phi_i^*$ , dû à l'interaction entre les sous-systèmes.

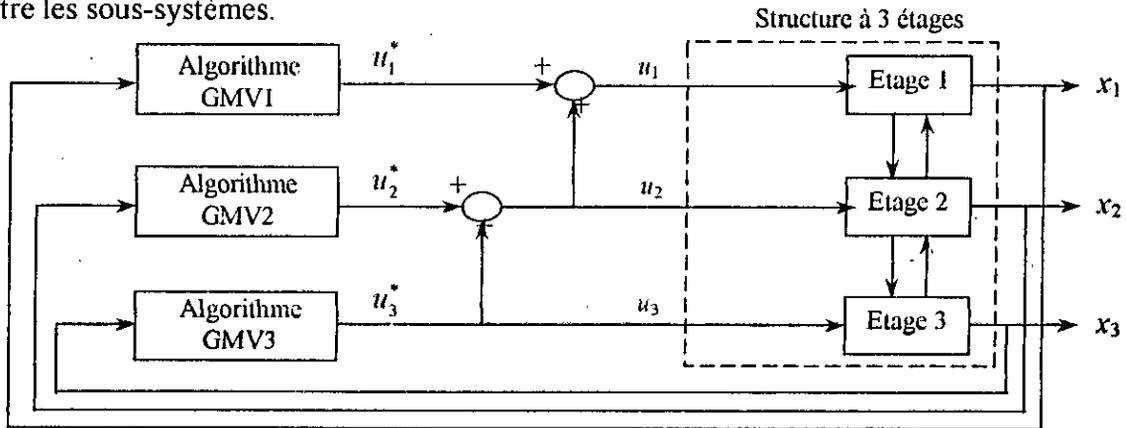


Figure III.10. Structure de la stratégie de commande pour une structure à trois degrés de liberté

### III.10 Résultats de simulation

Nous allons procéder dans cette partie à la simulation des algorithmes développés dans le cas des structures articulées et des structures en génie civil.

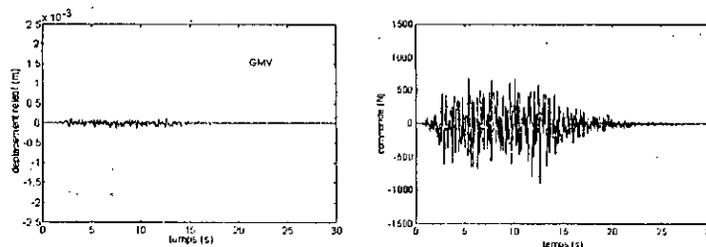
#### III.11.1 Commande des structures en génie civil

Pour démontrer l'efficacité des techniques développées, nous avons effectué des tests de simulation sur une structure à un seul degré de liberté. Un contrôleur à tendon active est installé dans la structure où l'angle d'inclinaison des tendons par rapport au sol est de  $25^\circ$ . La force de commande générée par l'actionneur est  $u/\cos 25^\circ$ . Ainsi, on peut supposer que la commande est appliquée au niveau de l'étage (haut de la structure) et est générée de l'extérieur par une source d'énergie indépendante [Sim94][Gue01].

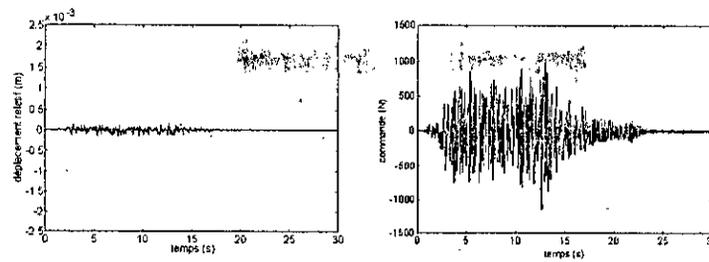
Les polynômes de pondération  $P_D$  et  $P_N$  utilisés dans la commande à GMV sont respectivement : 1 et  $1 - 0.5q^{-1}$ . Les résultats de simulation sont consignés sur les figures III.11 pour une excitation de Kanai-Tajimi.

Les figure III.12 représentent les résultats de simulation dans le cas où la structure est commandée par l'algorithme LQR. Les coefficients de pondération sont :

$$Q = \begin{bmatrix} k & 0 \\ 0 & 0 \end{bmatrix}; \quad R = 0.05$$

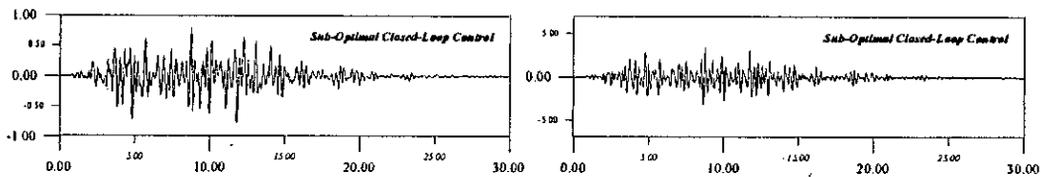


(b) Commande avec introduction du modèle du séisme

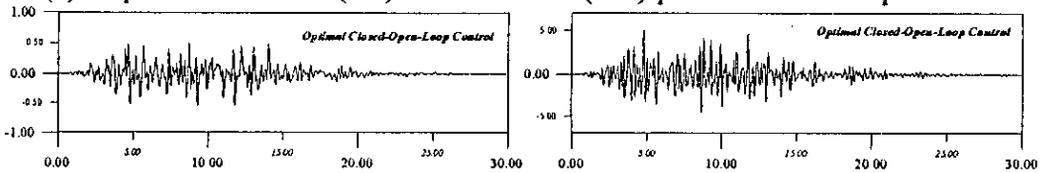


(c) Commande sans introduction du modèle du séisme

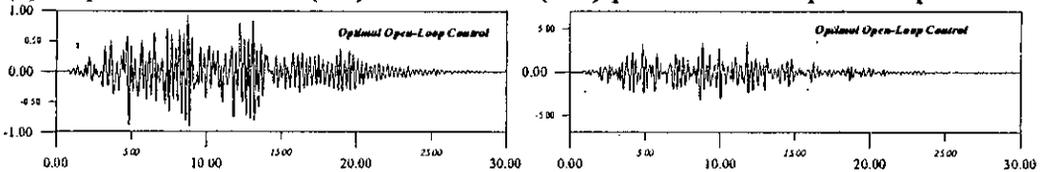
Figure III.11 Réponse de la structure à l'excitation de Kanai-Tajimi pour  $Q=10^{-7}$



(a) Déplacement relatif (cm) et commande (KN) pour 'Closed Loop Control'



(b) Déplacement relatif (cm) et commande (KN) pour 'Closed Open-Loop Control'



(c) Déplacement relatif (cm) et commande (KN) pour 'Open Loop Control'

Figure III.12 Réponse de la structure pour la commande LQR

Les résultats de simulations démontrent l'efficacité de la GMV dans la commande des structures. Le déplacement relatif de la structure a été considérablement réduit et devient presque négligeable par rapport au déplacement en boucle ouverte.

Dans le cas des structures à plusieurs degrés de liberté, des tests de simulation des techniques de commande multivariées développées ont été effectués. Comme dans le cas monovarié, Un contrôleur à tendon active est installé au niveau de chaque étage de la structure où l'angle d'inclinaison des tendons par rapport au sol est de  $25^\circ$ . la force de commande générée par l'actionneur placé à l'étage  $i$  est  $u_i/\cos 25^\circ$ . Ainsi, on peut supposer que la commande est appliquée au niveau de chaque étage et est générée de l'extérieur par une source d'énergie indépendante [Had97].

Les réponses de la structure aux modèles d'excitation de Kanai-Tajimi sont consignées sur les figures III.13.

Une identification en hors-ligne de la structure à trois degrés de liberté a été effectuée en utilisant l'algorithme MCE à trace constante pour déterminer le modèle ARMAX découplé de la structure. La trace de l'algorithme est de  $10^7$  et  $\lambda_1/\lambda_2=1$ . Une séquence binaire pseudo-aléatoire (SBPA) de longueur 8 a été utilisée comme excitation pour chaque étage. L'amplitude de la SBPA est de 120, 100 et 80 pour le 1<sup>er</sup>, 2<sup>ème</sup> et 3<sup>ème</sup> étage respectivement. L'évolution dans le temps des paramètres du modèle ARMAX ainsi que l'erreur de prédiction est représentée sur la figure III.14

Les réponses de la structure aux modèles d'excitation de Kanai-Tajimi sont consignées sur les figures III.15 avec :

$$Q_i(q^{-1}) = 2 \cdot 10^{-7}, P_i(q^{-1}) = 1 - 0.5q^{-1}, \quad i = \overline{1,3}$$

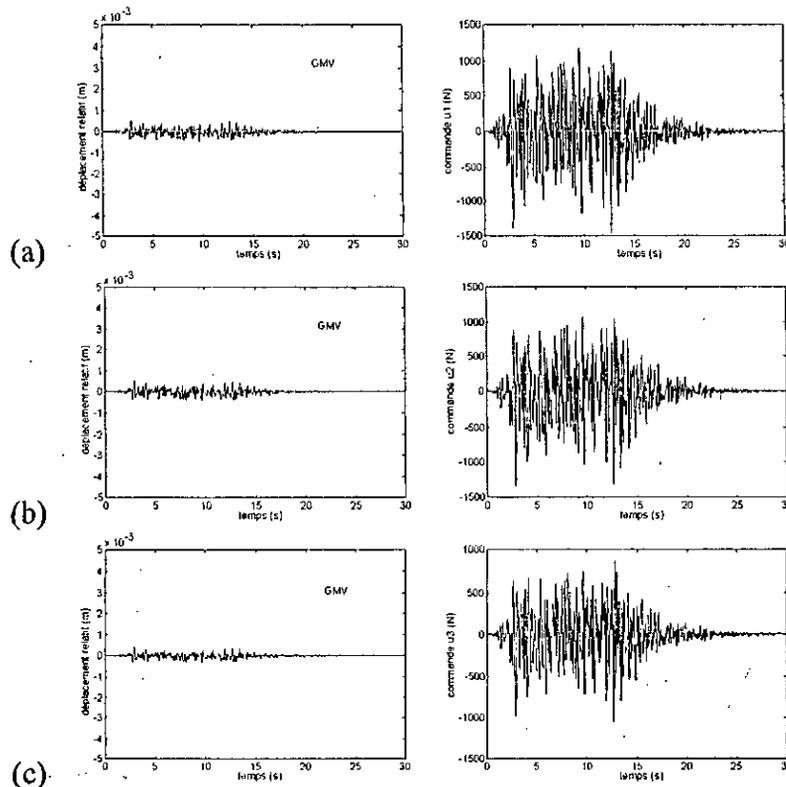
Pour implanter l'approche empirique, la tableau III.1 donne les nouveaux paramètres de la structure (modèle reparamétrisé). Les polynômes de pondération utilisés dans l'algorithme de la GMV sont :

$$Q_i(q^{-1}) = 2 \cdot 10^{-7}, P_i(q^{-1}) = 1 - 0.5q^{-1}, \quad i = \overline{1,3}$$

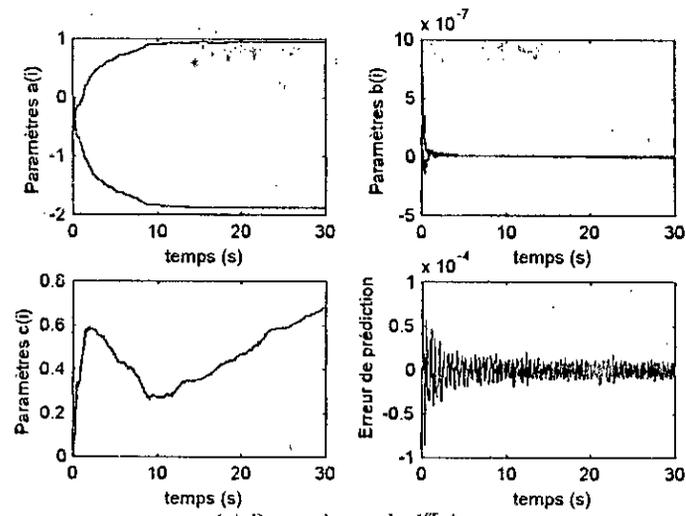
Les réponses de la structure aux modèles d'excitation de Kanai-Tajimi sont consignées sur les figures III.16.

**Tableau III.1** Paramètres de la structure reparamétrisée

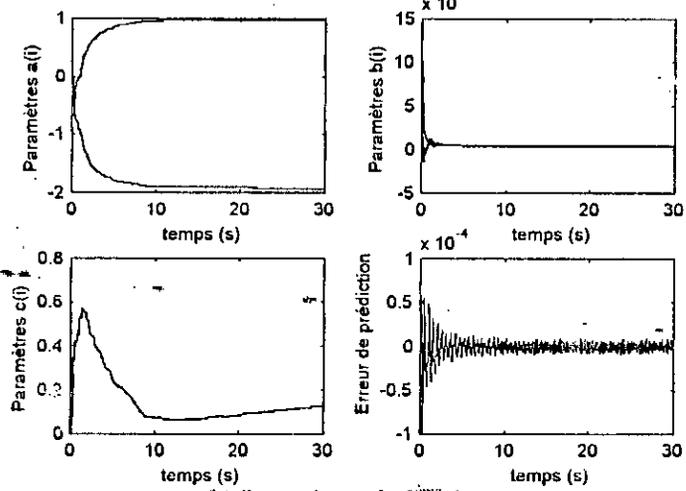
	1 <sup>er</sup> étage	2 <sup>ème</sup> étage	3 <sup>ème</sup> étage
$m_i^*$	2100 Kg	2100 Kg	2100 Kg
$c_i^*$	14175 N.s/m	21000 N.s/m	10500 N.s/m
$k_i^*$	3869950 N/m	5215000 N/m	2607500 N/m



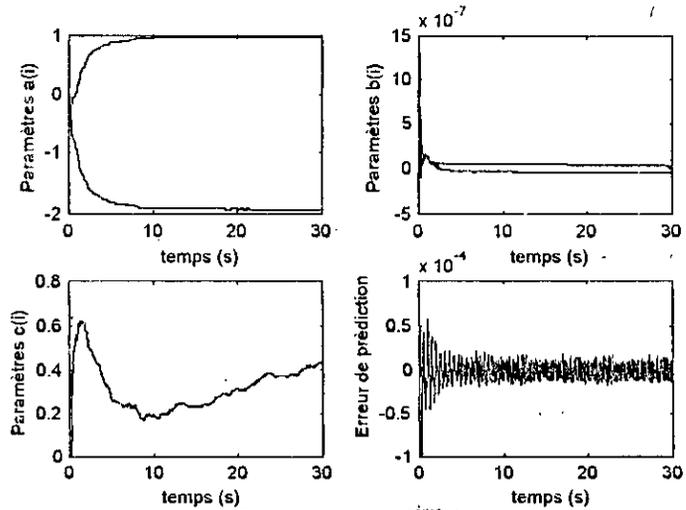
**Figure III.13.** Réponse à l'excitation de Kanai-Tajimi  
 (a) 1<sup>er</sup> étage, (b) 2<sup>ème</sup> étage, (c) 3<sup>ème</sup> étage



(a) Paramètres du 1<sup>er</sup> étage



(b) Paramètres du 2<sup>ème</sup> étage



(c) Paramètres du 3<sup>ème</sup> étage

Figure III.14. Evolution des paramètres du modèle ARMAX identifié

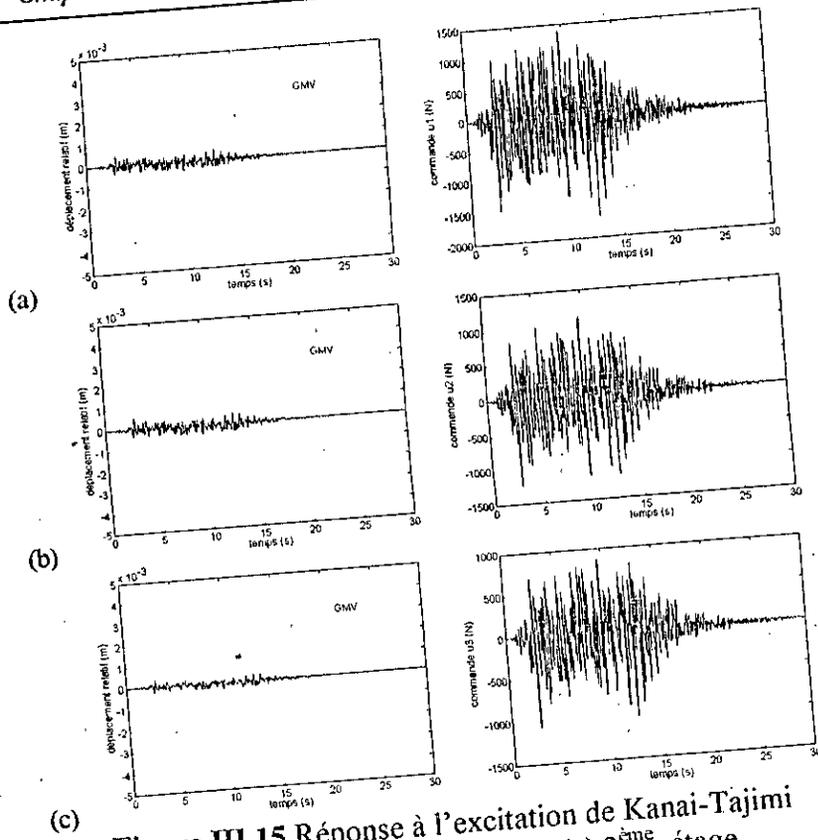


Figure III.15 Réponse à l'excitation de Kanai-Tajimi  
 (a) 1<sup>er</sup> étage, (b) 2<sup>ème</sup> étage, (c) 3<sup>ème</sup> étage

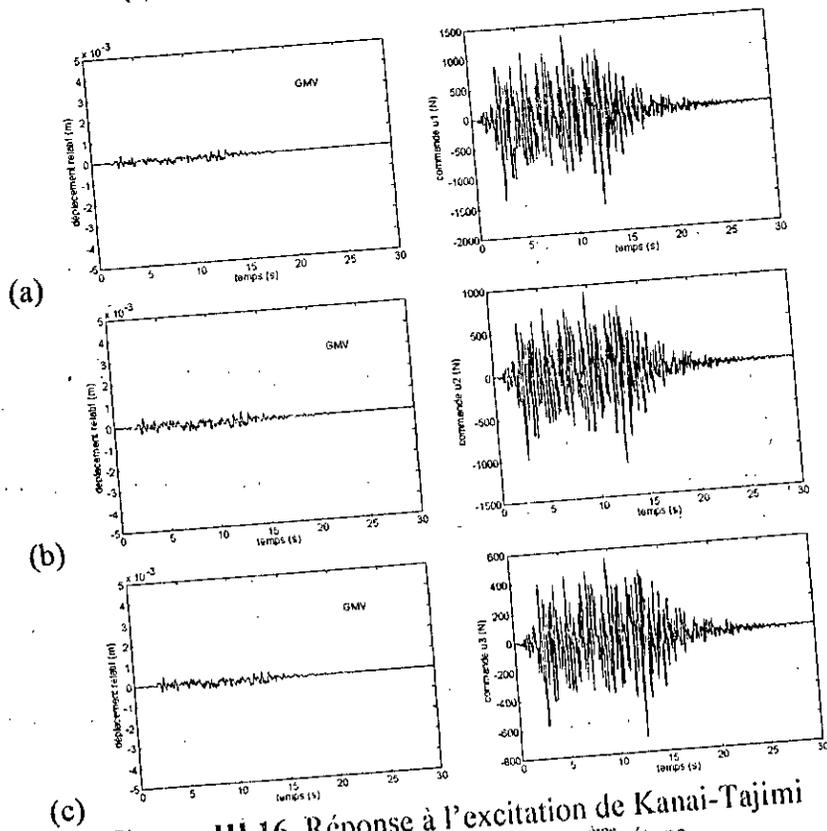


Figure III.16 Réponse à l'excitation de Kanai-Tajimi  
 (a) 1<sup>er</sup> étage, (b) 2<sup>ème</sup> étage, (c) 3<sup>ème</sup> étage

### III.10.2 Commande des structures articulées

Dans cette partie, nous nous intéressons à déterminer le modèle structurelle et paramétrique du robot. La Modélisation par équations aux différences paramétriques, nécessite une connaissance des degrés des différents polynômes. Le comportement dynamique du robot, pouvant être assimilé à un double intégrateur (équation différentielle du second ordre), nous permet de donner la structure suivante:

$$n_1=n_2=n_3=2 : \text{degrés des dénominateurs}$$

$$m_1=m_2=m_3=2 : \text{degrés des numérateurs}$$

On s'attache à déterminer les paramètres du modèle de représentation. L'algorithme du MCR à trace constante, nous permet la détermination du modèle paramétrique. La trace de l'algorithme est de  $5 \text{ Gi} = 5 \cdot 10^4$  et  $\lambda_1 / \lambda_2 = 1$ . L'excitation du robot est une séquence binaire pseudo aléatoire (SBPA) de longueur 8. L'évolution des sorties ainsi que les erreurs sont consignées sur la figure III.17.a. L'évolution, dans le temps, des paramètres est représentée dans la figure III.17.b

La commande à GMV est appliquée au robot, pour une consigne polynomiale du 3<sup>ème</sup> degré, qui assure une continuité en position et en vitesse aux extrémités [Cra86], définie par:

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

$$a_0 = \theta_0.$$

$$a_1 = \dot{\theta}_0.$$

Où  $a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f$ . dont les conditions aux limites:  $\theta(t_0) = \theta_0; \dot{\theta}(t_0) = \dot{\theta}_0.$   
 $\theta(t_f) = \theta_f; \dot{\theta}(t_f) = \dot{\theta}_f.$

$$a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) + \frac{1}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0).$$

Les résultats de simulation sont consignés sur la figure III.18.a pour la commande à GMV. Les valeurs des commandes sont représentées dans la figure III.18.b.

La commande par PRPE est appliquée au robot, pour la même consigne que précédemment. Les figures III.19.a et b représentent les résultats de simulation pour une telle commande.

Dans le cas de la commande par PP, on choisit  $Q_i / P_i = 1$  et le modèle de la référence  $w$  est unitaire. Les résultats de simulation sont consignés sur les figures III.20.a et b.

Les figures III.21.a et b représentent les résultats de simulation dans le cas où le robot est commandé avec la commande LQC avec :

$$Q_i = \begin{bmatrix} q_{i1} & 0 \\ 0 & q_{i2} \end{bmatrix} \text{ et } R_i = r_i.$$

La commande mixte est appliquée au robot, où la première articulation est commandée avec l'algorithme à PRPE:

$$Q_1 = 0.007; P_1 = 1; \frac{B_{m1}}{A_{m1}} = 1;$$

et la deuxième articulation est commandée avec GMV:

$$Q_2 = 0.05; P_2 = 1; \frac{B_{m2}}{A_{m2}} = \frac{0.095q^{-1}}{1-0.9q^{-1}};$$

tandis que la troisième est commandée avec le PP:

$$\frac{Q_{13}}{P_3} = \frac{1}{1-0.5q^{-1}+0.06q^{-2}}; \frac{B_{m2}}{A_{m2}} = \frac{0.095q^{-1}}{1-0.9q^{-1}};$$

Les résultats sont montrés dans la figure III.22.

Dans le cas du LMFC, le modèle du robot est défini dans le chapitre I. En utilisant (III.64), on peut calculer  $k_p$  et  $k_u$ , en imposant  $k_m = 0$ , avec un choix

de:  $A_m = \begin{bmatrix} 0 & 1 \\ -a_1 & -a_2 \end{bmatrix}; B_m = \text{diag} \begin{bmatrix} 0 \\ a_1 \end{bmatrix}$ . Où  $a_1 = \lambda_1 \lambda_2$  et  $a_2 = -(\lambda_1 + \lambda_2)$ . 44

Avec  $\lambda_1$  et  $\lambda_2$  : pôles désirés.

Les conditions d'Erzenbeger (III.66) sont vérifiées pour notre cas. Dans le cas où on applique la commande LMFC au robot, les figures III.23.a et b illustrent les résultats de simulation. ( $\lambda_1 = \lambda_2 = -40$ ).

En dernier lieu pour une consigne définie par la fenêtre de VIVIANI (voir chapitre I) et en utilisant l'algorithme mixte, les figures III.24.a et b montrent les résultats de simulations. La projection de la trajectoire spatiale sur différent plan est consignée sur la figure III.24.c.

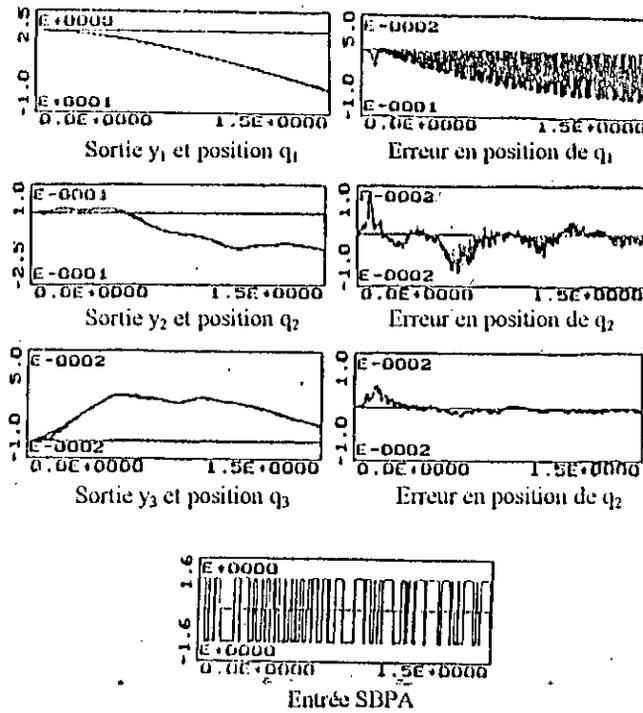


Figure III.17.a Réponses du robot pour SBPA

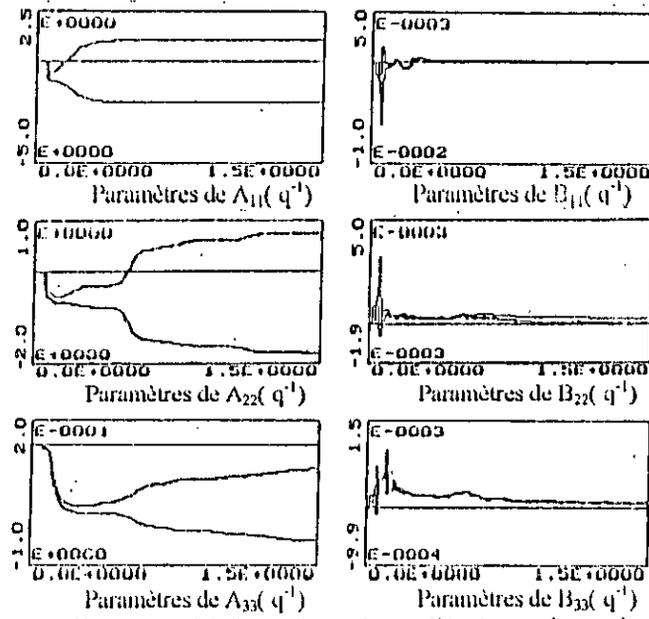


Figure III.17.b Paramètres du modèle de représentation

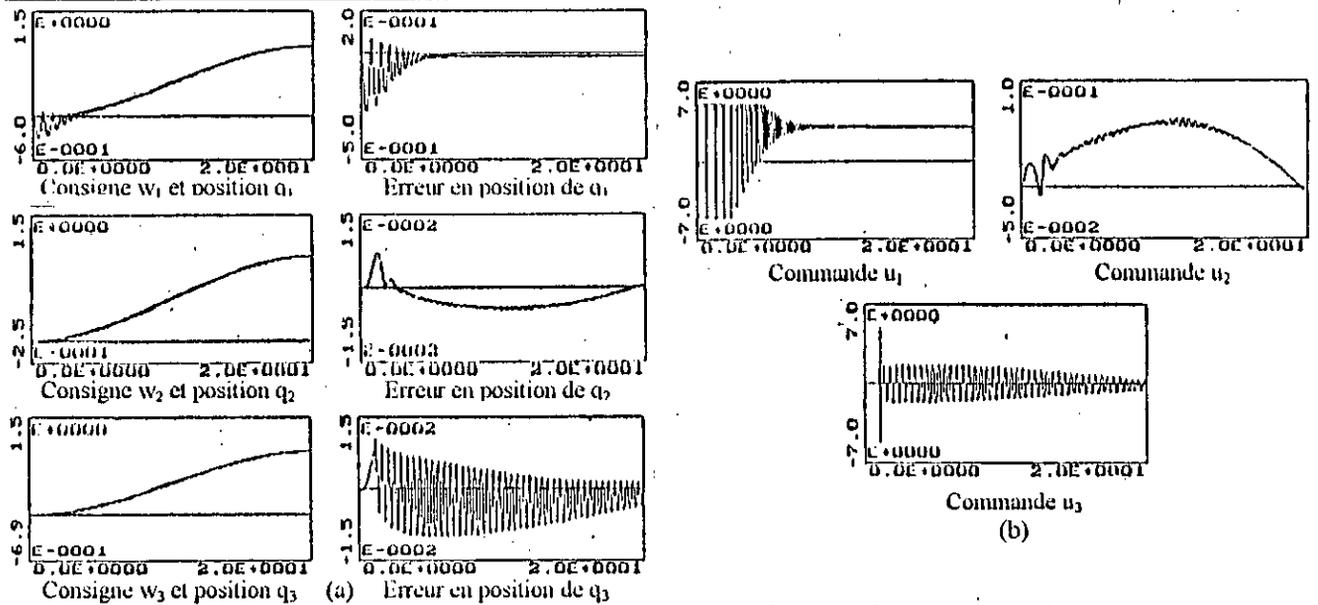


Figure III.18 Réponses du robot pour la commande à GMV (Référence Polynomiale)

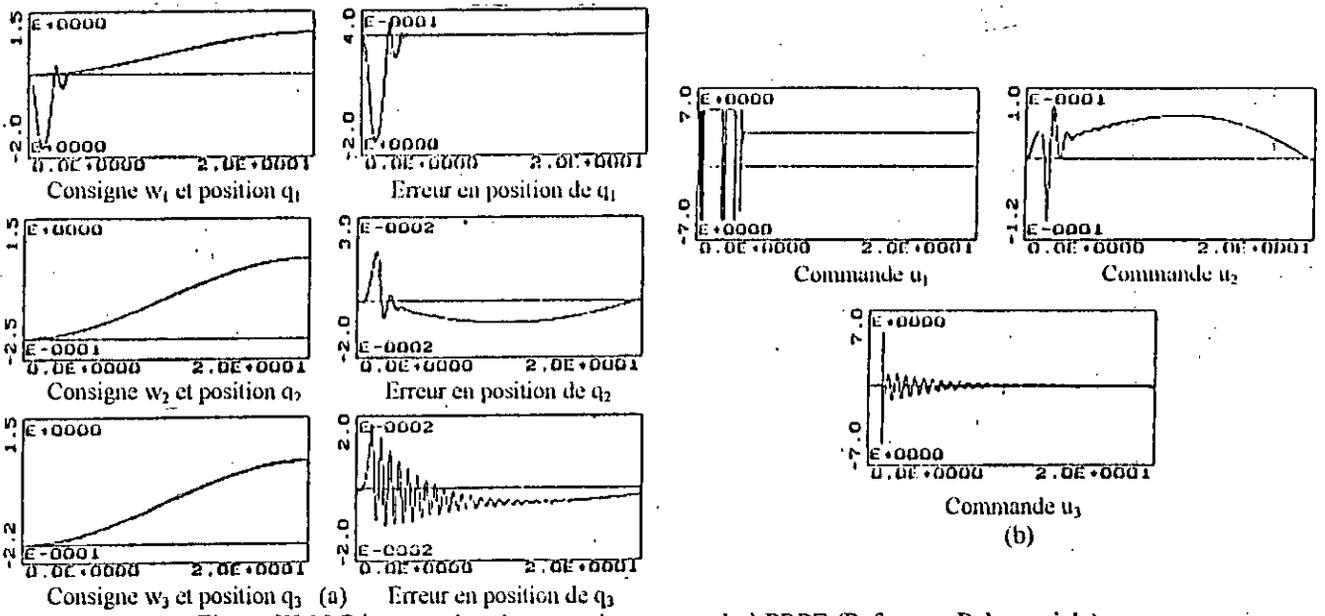


Figure III.19 Réponses du robot pour la commande à PRPE (Référence Polynomiale)

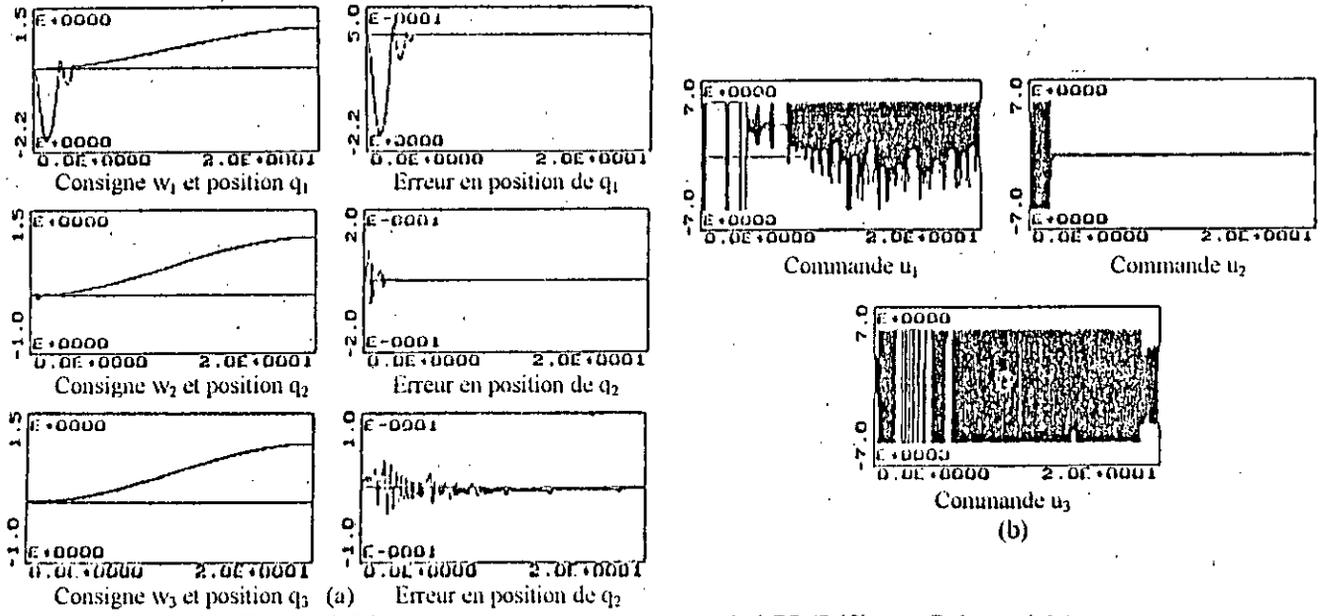


Figure 20 Réponses du robot pour la commande à PP (Référence Polynomiale)

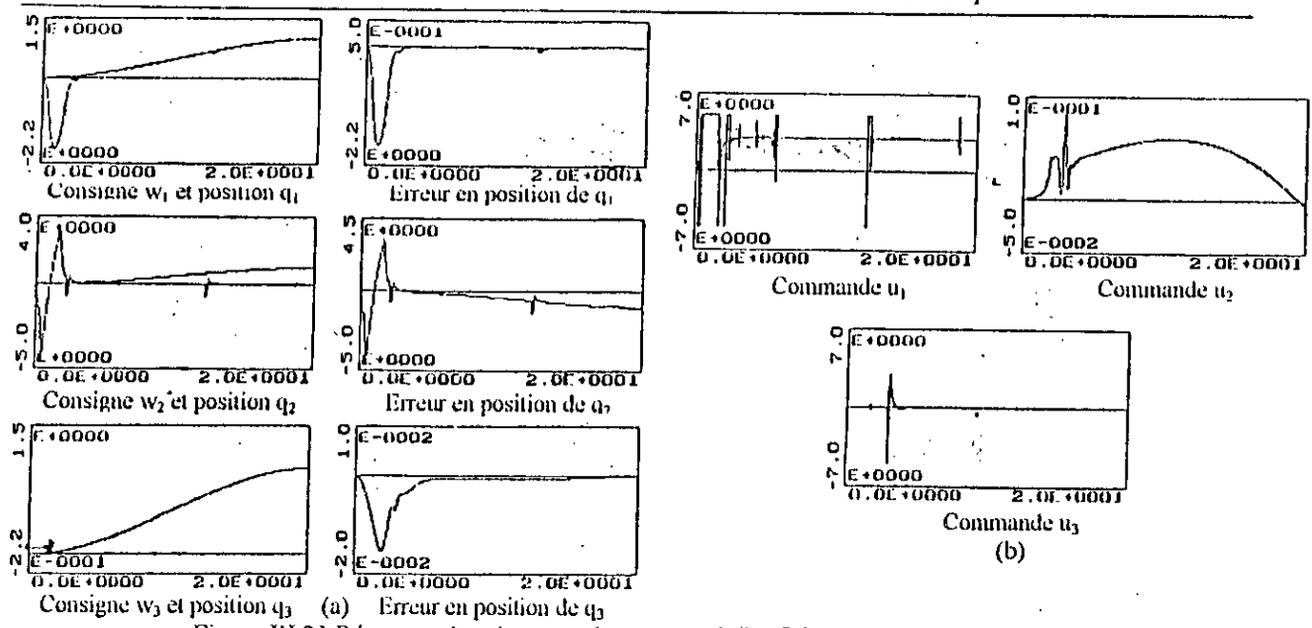


Figure III.21 Réponses du robot pour la commande à LQC (Référence Polynomiale)

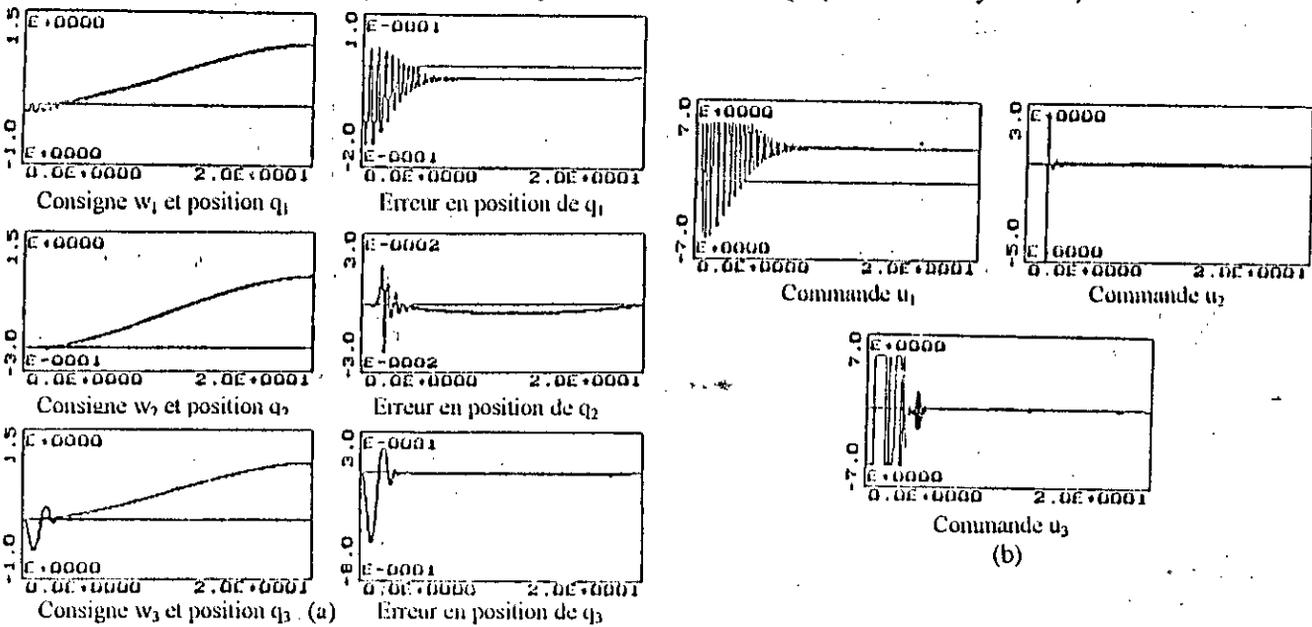


Figure III.22 Réponses du robot pour la commande Mixte (Référence Polynomiale)

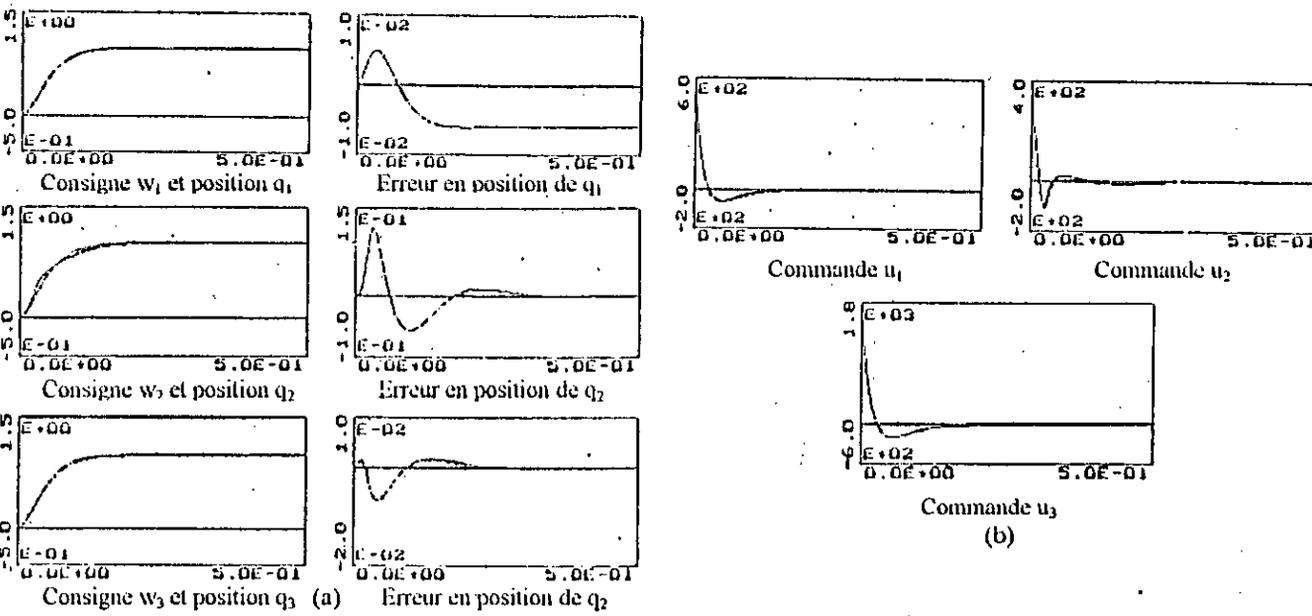


Figure 23 Réponses du robot pour la commande LMFC (Référence Polynomiale)

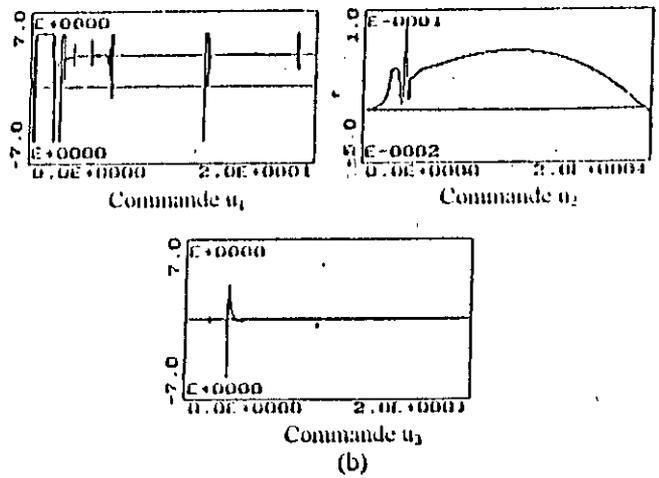
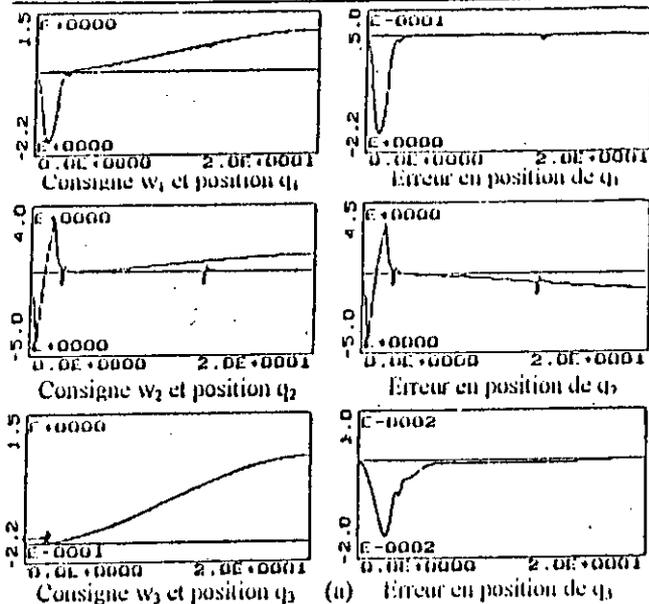


Figure III.21 Réponses du robot pour la commande à LQC (Référence Polynomiale)

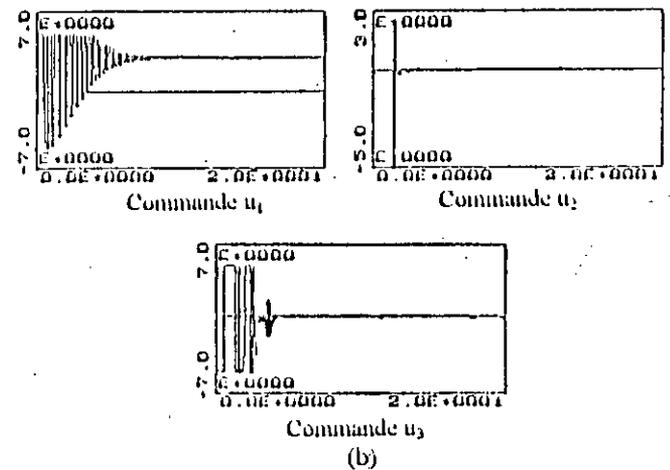
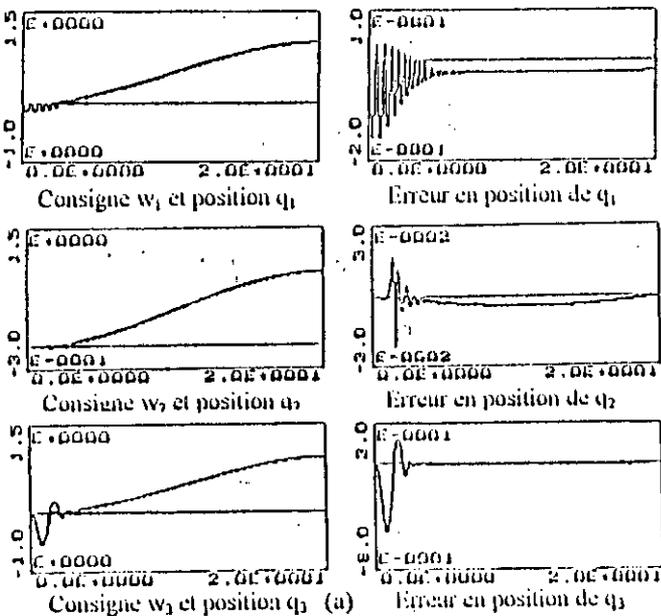


Figure III.22 Réponses du robot pour la commande Mixte (Référence Polynomiale)

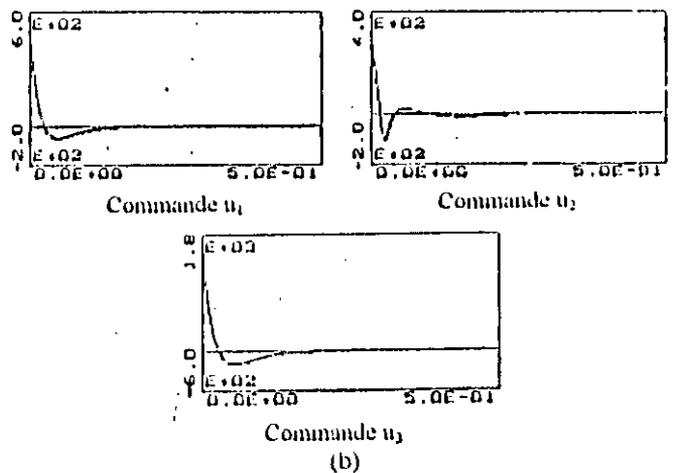
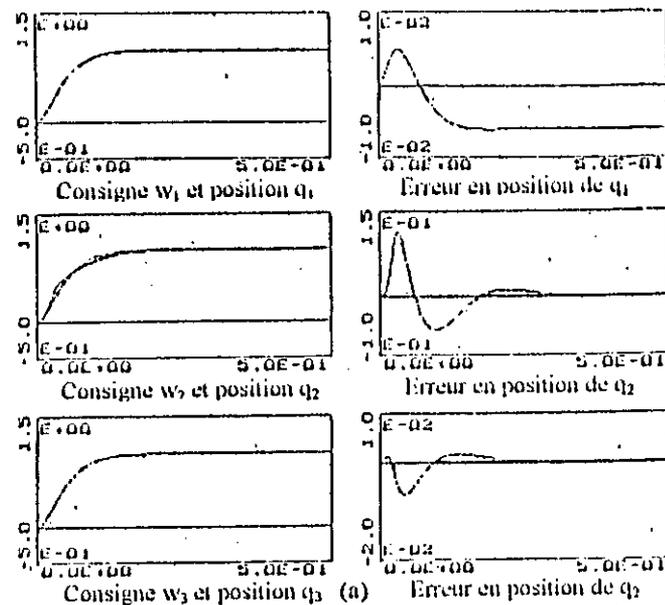


Figure 23 Réponses du robot pour la commande LMFC (Référence Polynomiale)

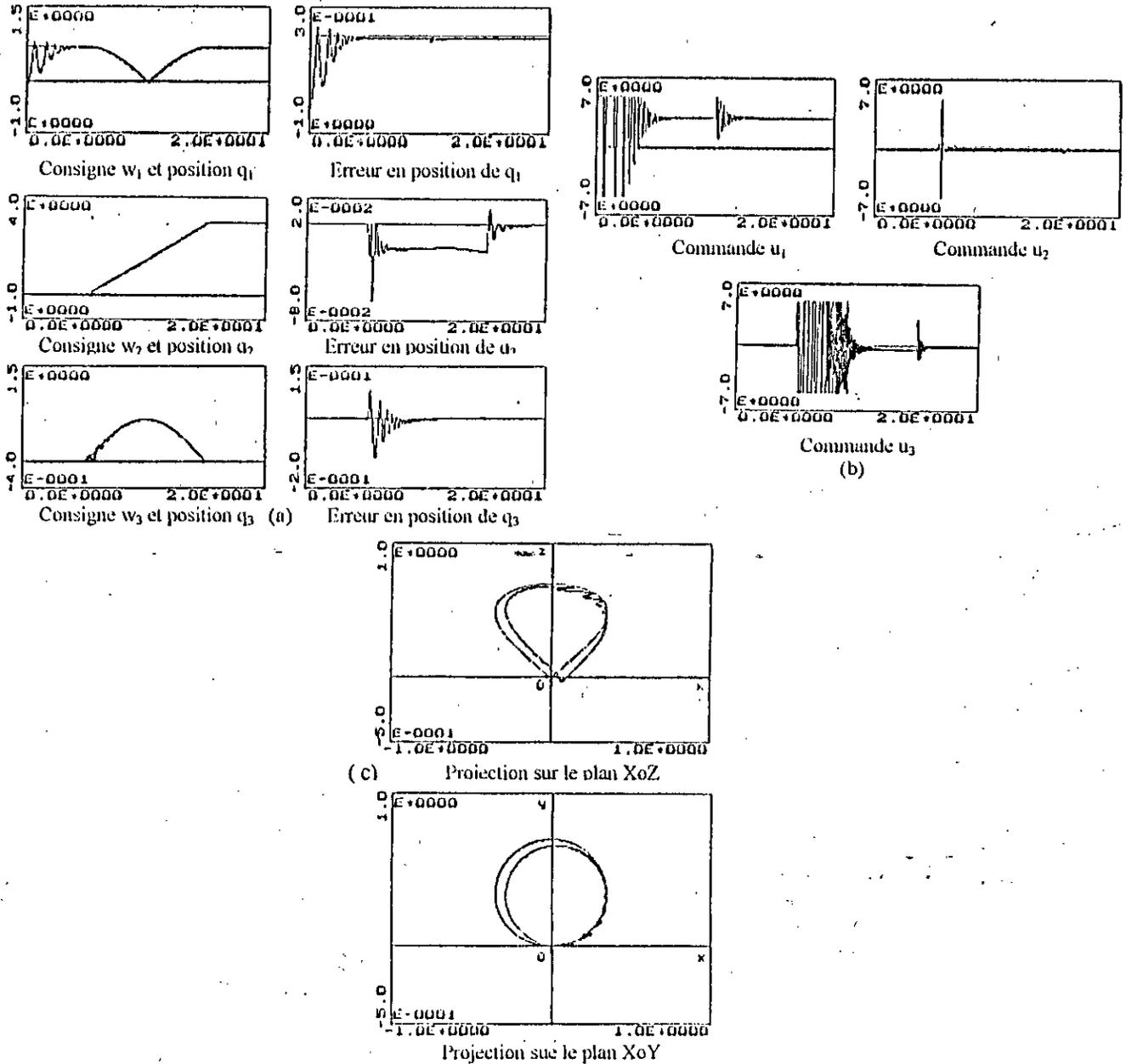


Figure III.24 Réponses du robot pour la commande mixte (Référence VIVIANI)

### III.11 Conclusion

Nous avons présenté dans la première partie de ce chapitre l'algorithme à VM, à GMV, à PRPE, à PP, à PP et de zéros et la commande LQC. Dans la seconde partie, on s'est intéressé à l'application de ces algorithmes pour la commande d'un robot manipulateur et d'une structures en génie civil ainsi que le développement de l'algorithme mixte et multirate.

Le modèle de représentation choisi a permis l'extension des algorithmes monovariabiles aux cas multivariable. Le comportement non linéaire du robot nécessite un algorithme d'identification rapide pour obtenir à chaque période d'échantillonnage un modèle linéaire du manipulateur. Lorsque ce modèle décrit le comportement du robot avec une erreur de sortie acceptable, on peut geler l'algorithme d'identification. Par ailleurs, il est important de souligner qu'il est nécessaire d'élaborer une procédure de supervision pour gérer la logique du gel de l'estimateur.

Pour les structures à un seul degré de liberté en génie civil, la commande à variance minimale généralisée (GMV) permet, avec un bon choix des polynômes de pondération, d'obtenir des performances très satisfaisantes et de compenser l'effet de l'excitation sismique d'une façon considérable. Cependant, on a mis en évidence que la connaissance du modèle de l'excitation sismique et son incorporation dans le développement du modèle ARMAX permet d'améliorer d'avantages les performances du système en boucle fermée. On obtiendra de meilleures performances avec un effort de commande plus réduit. Donc un gain du point de vue performances et du point de vue économie de l'énergie.

La commande linéaire quadratique est synthétisée par minimisation d'un critère quadratique sur un horizon de temps. La minimisation de ce critère à temps infini d'observation est un moyen très efficace d'élaboration d'une structure de commande par retour d'état pour les systèmes linéaire. Toute fois le choix des matrices de pondération  $Q$  et  $R$  n'est pas évident. Lorsque l'horizon est fini, le système à commande optimale n'est pas stationnaire, de plus pour suivre son évolution, il faut connaître la matrice  $P(t)$  c'est à dire résoudre l'équation de Ricatti.

Pour les structures articulées, l'algorithme de commande à GMV est une solution optimale du problème de commande pour les processus modélisable avec un modèle ARMAX, sous réserve du choix d'une bonne pondération dans le cas où le système est à phase non minimale (ce qui est le cas pour le robot manipulateur). L'introduction de la pondération modifie le comportement en BF du robot. Le choix du polynôme  $P$  permet de donner au robot une dynamique spécifique en BF et le polynôme de pondération de la commande, permet de faire le lien entre la GMV et PP.

Dans le cas où l'on modélise le robot avec un modèle déterministe (DARMA), l'algorithme précédent est un cas particulier de celui du PRPE ( $P_0 = C$ ), qui donne une plus grande liberté de choix des pôles en BF. Cette solution n'est néanmoins pas optimale dans le cas stochastique, le choix des pôles pouvant être inadéquat. Ce choix est une étape importante pour réduire l'effort de commande et assurer une meilleur poursuite et régulation. L'instabilité des zéros du modèle de représentation, nécessite l'introduction de la pondération de la commande. L'introduction d'une pondération multiple de  $(1-q^{-1})$  n'est pas nécessaire et détériore les performances de poursuite.

Le problème du comportement à phase non minimale du robot est résolu d'une manière très simple (sans introduire de pondération) par l'algorithme à PP, qui en faisant attention aux choix des pôles dans le cas stochastique, donne de très bons résultats.

L'imposition d'une dynamique unité, nécessite un effort de commande important. Pour surmonter ce problème, on introduit une dynamique en BF par la spécification des pôles. Mais des instabilités numériques peuvent engendrer des oscillations de la commande, même en imposant une dynamique de référence.

La commande LQC montre son efficacité même dans le cas des systèmes à phase non minimale. La complexité de calcul devient importante lors de l'utilisation d'un estimateur d'état, dans le cas où l'on utilise le modèle entrée-sortie. Ceci s'avère inutile en identifiant directement le modèle d'état et non pas le modèle entrée-sortie. Le choix des coefficients de pondération est une étape difficile.

Pour tirer profit des performances de ces différents algorithmes, nous avons proposé l'algorithme mixte. On peut avec cet algorithme commander chaque entrée-sortie avec une commande adéquate. Le choix de la stratégie adéquate est fonction du comportement dynamique de l'entrée-sortie choisie. Ainsi on peut sélectionner la loi de commande, en fonction de la tâche à accomplir par l'articulation considérée.

L'implémentation pratique d'une telle commande nécessite un processeur multitache, pour accélérer le temps de calcul de la commande. L'application industrielle d'une telle commande n'a pas été encore réalisée dans le domaine pratique.

Le temps de réponse de chaque articulation du robot, suivant la tâche prédéfinie, permet de choisir des périodes d'échantillonnage différentes. On aboutit alors à l'algorithme multirate. Cet algorithme offre, en premier lieu, la possibilité de générer chaque trajectoire d'une façon indépendante des autres. En second lieu, elle permet de favoriser les articulations les plus actives par rapport aux autres. Une articulation active nécessite une période d'échantillonnage rapide, tandis que la plus lente ne le nécessite pas. L'implémentation d'une telle commande nécessite l'utilisation des processeurs de différente nature (rapidité) pour chaque articulation.

Les résultats de simulation ont montré la robustesse de ces algorithmes aux variations paramétriques et aux erreurs de modélisation. Les objectifs de poursuite et régulation sont atteints d'une façon satisfaisante. Ce qui offre à ces algorithmes une large application en dynamique de structure.



---

## Chapitre IV

### *Commande Non Linéaire Déterministe et Stochastique*

---

#### **Introduction**

Le modèle des structures articulées, est caractérisé par un système d'équations différentielles non linéaires. La connaissance de ce modèle permet l'implémentation d'asservissements non linéaires multivariables performants. Ainsi la détermination du retour non linéaire est essentiellement basée sur le modèle de connaissance. Nous construisons alors un modèle de base (model based) [Cra86] pour la réalisation de ces commandes. L'idée de base est la compensation des non linéarités existantes dans le modèle du robot. Ceci a une incidence marquante sur leurs caractéristiques intrinsèques, du fait que l'erreur de modélisation est une étape inévitable dans l'élaboration de tout modèle de connaissance. Le but du contrôle est de réaliser des modèles dont les non linéarités les plus complexes sont compensées. Dans le cas où les non linéarités ne sont pas sévères, une linéarisation locale peut être utilisée pour l'obtention d'un modèle linéaire. Ce modèle est une approximation des équations non linéaires autour du point de fonctionnement. L'utilisation d'un algorithme de commande linéaire peut réaliser les performances voulues [Gue95a]. On aboutit alors à la commande non linéaire hybride.

Il existe plusieurs techniques de commande non linéaire développées dans la littérature [Fu87][Vib87][Lop84][Cra86]. L'inconvénient majeur de ces techniques réside dans la connaissance exacte du modèle du robot. En effet, l'obtention de l'inertie des différentes liaisons et le coefficient de frottement de chaque articulation, est une tâche très difficile.

La commande adaptative à perturbation ( APC ) développée dans ce chapitre est basée sur un modèle à perturbation obtenu par linéarisation du modèle non linéaire autour d'une trajectoire nominale. Cette trajectoire est spécifiée par une interpolation de la position, de la vitesse et de l'accélération des articulations, entre différents points de l'espace opérationnel [Gue96b].

La commande ainsi réalisée est caractérisée par deux actions. Une action anticipatrice (feedforward), qui utilise la commande nominale calculée à partir des caractéristiques cinématiques de la trajectoire. La seconde est une action de retour (feedback) dans laquelle, on calcule la commande séparément de celle calculée par l'action "feedforward". Cette dernière se base principalement sur le modèle à perturbation.

Dans le cas du découplage non linéaire, on s'intéresse à changer la structure du modèle de connaissance, par des retours non linéaires, sans introduction d'approximation. Ceci permet d'utiliser des algorithmes de commande linéaires, en se basant sur un nouveau modèle moins complexe [Gue00a][Gue00b].

On introduira ensuite une loi de commande plus performante qui ne nécessite aucune approximation. Cette loi se base sur la linéarisation et le découplage. La connaissance exacte du modèle de connaissance permet l'implémentation directe de la commande linéarisante à paramètres connus [M'si93][Cra88][Ben91]. Cette stratégie exige l'obtention des expressions analytiques des fonctions non linéaires dans le modèle du robot. Le comportement du robot, bouclé avec cette loi de commande, est identique à celui d'un système du second ordre. La réalisation pratique d'une telle loi nécessite un outil de calcul puissant. Pour surmonter ce problème, la commande linéarisante prédictive a été introduite [Ben91].

Dans le cas où le modèle dynamique du robot est mal connu, l'implémentation de cette loi s'avère inutile, du fait qu'elle a besoin d'un modèle précis. L'utilisation d'un algorithme d'adaptation paramétrique s'impose. Celui-ci estime les paramètres du robot pour que la commande assure un suivi uniforme de la trajectoire désirée et garantit la stabilité globale du système en boucle fermée. L'étude de cette stabilité est basée sur la méthode directe de Lyapounov et sur la passivité et la théorie d'hyperstabilité de Popov [Lan79][Pop73][Lan89]. Craig [Cra87] développe l'algorithme de commande linéarisante adaptative. L'application d'une telle loi est implémentée sur le robot Adept-One, dont les résultats sont intéressants.

Les systèmes adaptatifs à modèle de référence (MRAS Model Reference Adaptive Systems) sont très utilisés dans de larges domaines, pour résoudre une variété importante de problèmes rencontrés en commande, identification et estimation d'état [Lan74][Lan79]. Le caractère dual de ces méthodes permet leur utilisation suivant la structure spécifiée, dans différentes applications [Lan79].

On distingue deux façons d'adapter le système ajustable, à fin d'assurer une minimisation de l'erreur généralisée ou de sortie [Cha87][Ast89][Lan79], entre le système ajustable et le modèle de référence. L'adaptation paramétrique se base sur l'ajustement des paramètres de la structure choisie; tandis que le signal de synthèse (signal synthesis) est une manière de transformer l'ajustement en un signal d'entrée qui attaque le bloc ajustable.

Dans ce chapitre, nous présentons en premier lieu les différentes stratégies de commande par retour non linéaire. En second lieu, des techniques basées sur la théorie d'hyperstabilité de Popov sont développées. Nous examinons les performances de la commande à gain programmé appliquée en génie civil. Nous présentons ensuite la version neuronale de certaines stratégies.

## IV.1 Compensation de l'effet gravitationnelle.

Dans la pratique, des charges variées de masses très différentes sont transportées par les robots. Ceci a une incidence marquante sur leurs caractéristiques intrinsèques. L'effet gravitationnel, introduit par les masses des différentes liaisons, introduit des non linéarités importantes dans le modèle dynamique du robot [Vib87], lorsque ce dernier opère dans un environnement spatiale. La connaissance de l'expression analytique de cet effet permet de le compenser et d'aboutir à un modèle simplifié.

Le modèle dynamique du robot, établi par le formalisme de Lagrange Euler dans le chapitre I, peut se mettre sous la forme suivante [Cra86]:

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) + f(\dot{q}) = \tau \quad (IV.1)$$

Où  $\tau^T = [\tau_1 \dots \tau_n]$ : couples généralisés.

$q^T = [q_1 \dots q_n]$ : variables articulaires généralisées.

$M(q)$ : Matrice d'inertie de dimension  $n \times n$ .

$V(q, \dot{q})$ : vecteur de dimension  $n$  spécifiant l'effet centrifuge et de coriolis.

$G(q)$ : vecteur de dimension  $n$  exprimant l'effet gravitationnel, dont la composante  $i$  est définie

$$\text{par } g_i(q) = - \sum_{k=i}^n m_k g^T U_{ki} r^k$$

$f(\dot{q})$ : vecteur de dimension  $n$  exprimant les frottements visqueux.

Pour éliminer l'effet de  $G(q)$  dans l'équation (IV.1), on choisit le couple généralisé  $\tau$ , de la manière suivante [Vib87]:

$$\tau = \tau_d + G_o(q) \quad (IV.2)$$

Où  $G_o(q)$ : est une approximation de  $G(q)$ .

Ainsi le modèle du robot défini par (IV.1) devient:

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) + f(\dot{q}) = \tau_d + G_o(q) \quad (IV.3)$$

Dans le cas où on connaît "exactement" l'expression analytique de l'effet gravitationnel, on a:

$$G(q) = G_o(q) \quad (IV.4)$$

Alors l'équation (IV.3) devient

$$M(q)\ddot{q} + V(q, \dot{q}) + f(\dot{q}) = \tau_d \quad (IV.5)$$

La compensation des couples dus à la pesanteur est réalisable de deux manières [Vib87]. Une possibilité consiste à équilibrer la structure articulée avec des masses de façon que celle-ci soit en équilibre indifférent quelque soit sa configuration. Ceci entraîne un alourdissement des structures. La seconde possibilité consiste à utiliser les moteurs des articulations pour créer des couples de compensation. Dans ce cas les moteurs sont sollicités en permanence même lorsque la structure n'est pas en mouvement. Il faut noter que dans le cas d'un robot réel, l'opération de compensation des couples de pesanteur nécessite la connaissance des différentes masses des articulations.

D'une façon générale, la compensation des couples de pesanteur d'un robot est d'un apport déterministe pour la création d'asservissement performant. Dans le système compensé, les erreurs statiques de position, de l'asservissement, seront nulles (effet intégrateur du modèle du robot). Ce point constitue l'apport essentiel qui résulte de l'application de la méthode de compensation des couples dus à la pesanteur.

Cette technique nécessite la connaissance des différentes masses des articulations et l'expression analytique de  $G(q)$ . En pratique le modèle défini par l'équation IV.1 est assujéti à des erreurs de modélisation. L'utilisation d'un algorithme de commande superviseur s'avère nécessaire. ainsi on aboutit à la structure de commande définie par la figure IV.1.

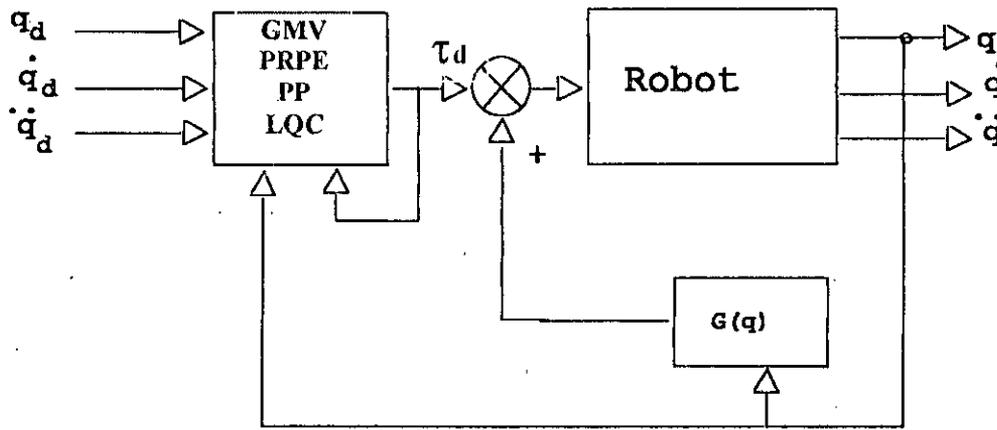


Figure IV.1 Commande hybride avec compensation des couples de pesanteur.

Dans le cas du robot de classe quatre, on a ( voir chapitre I ):

$$M(q) = \begin{bmatrix} m_1 + m_3 & 0 & 0 \\ 0 & m_3 J^* & 0 \\ 0 & 0 & m_3 \end{bmatrix}$$

$$V^T(q, \dot{q}) = \begin{bmatrix} 0 & 2m_3(q_3 - \frac{l_2}{2})\dot{q}_2\dot{q}_3 & -m_3(q_3 - \frac{l_2}{2})\dot{q}_2^2 \end{bmatrix} \quad G^T(q) = [(m_1 + m_3) \quad 0 \quad 0]$$

$$f^T(\dot{q}) = [f_1\dot{q}_1 \quad f_2\dot{q}_2 \quad f_3\dot{q}_3]$$

La commande définie par l'équation (IV.2) est:

$$\tau_1 = \tau_{d1} + (m_1 + m_3)g$$

$$\tau_2 = \tau_{d2}$$

$$\tau_3 = \tau_{d3}$$

(IV.6)

avec  $\tau_i = k_i u_i \quad i = 1, 3$ .

Où  $k_i$ : coefficient de proportionnalité de l'actionneur.

$u_i$ : commande de l'actionneur

Ainsi on obtient un modèle dynamique "continu" du robot, sans effet gravitationnel. On choisit ainsi différents algorithmes pour commander le robot.

## IV.2 Découplage non linéaire partiel.

L'idée de base d'une telle commande est de compenser tous les effets des non linéarités existants dans le modèle dynamique du robot. Ainsi le problème de commande des articulations d'un robot à  $n$  degré de liberté est ramené, par découplage non linéaire à celui du contrôle de  $n$  asservissements quasi-linéaires du second ordre.

Partant du modèle de robot défini par (IV.1), pour la quasi linéarisation de ce modèle, le couple de commande  $\tau$  peut être pris sous la forme suivante [Vib87]:

$$\tau = \tau_d + V_o(q, \dot{q}) + G_o(q) + f_o(\dot{q}) \quad (IV.7)$$

Où  $V_o, G_o$  et  $f_o$  sont des approximations de  $V, G$  et  $f$ . Ces derniers sont calculés, à partir de  $q$  et  $\dot{q}$ .

$\tau_d$  : couple de commande

En commandant le robot à partir du couple défini par (IV.7) et en admettant que:

$$V_o(q, \dot{q}) = V(q, \dot{q}); G_o(q) = G(q); f_o(\dot{q}) = f(\dot{q}) \quad (IV.8)$$

L'équation du robot défini par (IV.1) devient:

$$M(q)\ddot{q} = \tau_d \quad (IV.9)$$

Le modèle ainsi compensé est quasiment découplé, car la matrice  $M(q)$  est définie par [Fu87]:

$$M_{ij} = \sum_{k=\max(i,j)}^n \text{trace}(U_{kj} J_k U_{ki}^T); i, j = 1, 3 \quad (IV.10)$$

La matrice ainsi définie n'est pas forcément diagonale. Suivant la structure de chaque robot, cette matrice peut avoir plusieurs formes. C'est pour cette raison que le découplage est partiel.

Dans cette méthode de linéarisation par compensation non linéaire partielle, les termes  $V_o, G_o$  et  $f_o$  doivent être calculés en temps réel. Ceci réclame non seulement une connaissance correcte du modèle du robot, mais aussi une masse importante de calcul et un temps d'échantillonnage pour boucler un calcul conséquent.

De l'équation (IV.9), on remarque que le modèle du robot ainsi compensé est moins complexe, facilement commandable par des algorithmes adaptatives. L'introduction d'une commande auto-ajustable, est d'une importance primordiale, du fait que le calcul du couple défini par (IV.7), nécessite la connaissance exacte de plusieurs termes. Dans le cas où il y a variation dans le modèle du robot, ceci se répercutera d'une façon importante dans le modèle de connaissance, ce qui entraîne un modèle complexe mais non découplé [Cra86].

La structure de l'algorithme de commande hybride est défini par la figure IV.2.

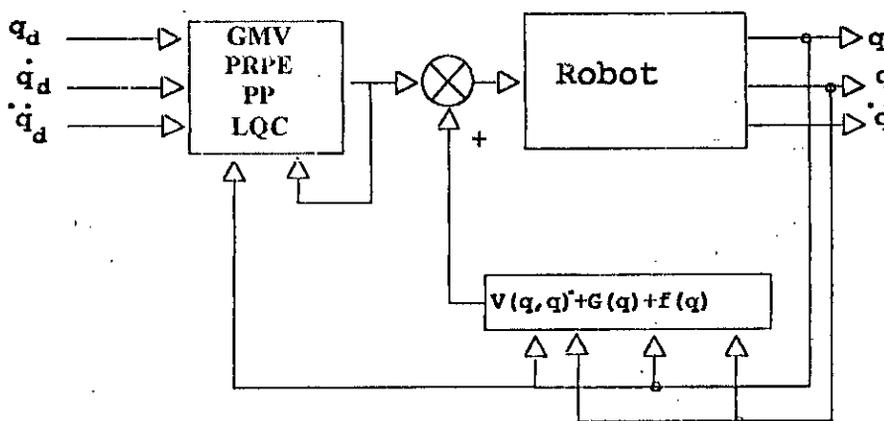


Figure IV.2 Commande hybride avec découplage non linéaire partiel.

Dans le cas du robot de classe quatre, la matrice  $M(q)$  est une matrice diagonale à coefficients variables. Donc le découplage dans ce cas s'avère d'un avantage remarquable. La commande définie par l'équation (IV.7) est:

$$\begin{aligned}
 \tau_1 &= \tau_{d1} + (m_1 + m_3)g + f_1 \dot{q}_1 \\
 \tau_2 &= \tau_{d2} + 2m_3 \left( q_3 - \frac{l_2}{2} \right) \dot{q}_2 \dot{q}_3 + f_2 \dot{q}_2 \\
 \tau_3 &= \tau_{d3} - m_3 \left( q_3 - \frac{l_2}{2} \right) + f_3 \dot{q}_3
 \end{aligned}
 \tag{IV.11}$$

avec  $\tau_i = k_i u_i$   $i = 1, 3$ .

Ainsi le modèle dynamique du robot est partiellement découplé. Son comportement est similaire à celui d'un double intégrateur, dont la constante de temps est variable.

### IV.3 Découplage non linéaire total.

Dans les parties précédentes, on s'est intéressé à la compensation des non linéarités, dans le modèle dynamique du robot pour aboutir à un modèle moins complexe. Le modèle obtenu par compensation de la pesanteur est non linéaire. Tandis que le modèle obtenu par le découplage non linéaire partiel est quasi-découplé, car en général la matrice  $M(q)$  n'est pas diagonale. Donc le modèle obtenu est non linéaire (même dans le cas de notre robot, la constante de temps de la troisième articulation est fonction de  $q$ ).

Dans cette approche, on s'intéresse à calculer un retour non linéaire aboutissant à un modèle linéaire totalement découplé. On cherche alors à trouver une commande utilisant le modèle inverse du robot. Le couple de commande peut être choisi sous la forme suivante [Cra86]:

$$\tau = \Omega \tau_d + \beta \tag{IV.12}$$

Où  $\tau$  et  $\tau_d$  sont définis précédemment.

$\beta$  : vecteur de dimension  $n$ .

$\Omega$  : matrice de dimension  $n \times n$ .

Si on choisit correctement  $\Omega$  et  $\beta$ , le robot peut être vu comme un système totalement découplé. c'est pour cette raison que la méthode est appelée découplage et linéarisation. Le modèle du robot est défini par l'équation (IV.1). En utilisant (IV.12), on peut choisir  $\Omega$  et  $\beta$  de la manière suivante :

$$\begin{aligned}
 \Omega &= M_o(q) \\
 \beta &= V_o(q, \dot{q}) + G_o(q) + f_o(\dot{q})
 \end{aligned}
 \tag{IV.13}$$

Où  $M_o, V_o, G_o$  et  $f_o$  : des approximations de  $M, V, G$  et  $f$ . Ils sont calculés en temps réel.

La loi de commande est définie par:

$$\tau = M_o(q) \tau_d + V_o(q, \dot{q}) + G_o(q) + f_o(\dot{q}) \tag{IV.14}$$

Dans le cas où  $M_o = M; V_o = V; G_o = G$  et  $f_o = f$ ; les équations (IV.1) et (IV.14) donnent:

$$\tau_d = \ddot{q} \tag{IV.15}$$

On remarque à partir de (IV.15), que le robot se comportera comme un double intégrateur. Ainsi le découplage est totale, ce qui permet l'utilisation des algorithmes monovariables continus, non adaptatifs du type PID. Dans le cas où la condition sur l'équation (IV.15) n'est pas vérifiée, le découplage n'est pas réalisé et on aboutit à un modèle complexe. La non validité de ces

conditions est due aux erreurs de modélisation et à l'implémentation numérique de la commande qui nécessite une commande continu idéale. On est amené alors à utiliser des algorithmes auto-adaptatives pour palier ce problème. Le mélange entre la méthode de "découplage et linéarisation" et les régulateurs à STR aboutit à une commande non linéaire hybride. Le comportement du système découplé est similaire à celui d'un double intégrateur, ce qui nous permet le choix de la structure du modèle de représentation aux différences (ARMAX ou

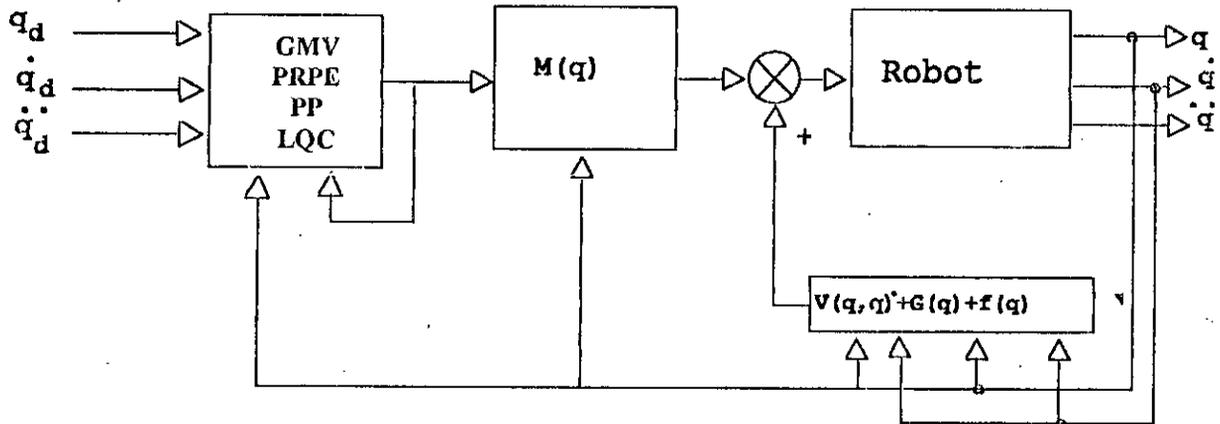


Figure IV.3 Commande hybride par découplage et linéarisation.

DARMA ). La structure de réglage hybride est illustrée dans la figure IV.3. Dans le cas du robot classe 4, la commande définie par (IV.14) devient:

$$\begin{aligned} \tau_1 &= (m_1 + m_3)\tau_{d1} + (m_1 + m_3)g + f_1\dot{q}_1 \\ \tau_2 &= m_3\tau_{d2} + 2m_3\left(q_3 - \frac{l_2}{2}\right)\dot{q}_2\dot{q}_3 + f_2\dot{q}_2 \\ \tau_3 &= m_3\tau_{d3} - m_3\left(q_3 - \frac{l_2}{2}\right) + f_3\dot{q}_3 \end{aligned} \quad (IV.16)$$

#### IV.4 Découplage non linéaire utilisant la géométrie différentielle. (Linearisation exacte)

Durant ces dernières années, des outils théoriques de commande non linéaire ont été développés, en utilisant le concept de géométrie différentielle [Isi85]. Avec cette technique, un système non linéaire peut être linéarisé exactement (exact linéarisation). Par contre la linéarisation locale utilise le développement en série de Taylor, autour d'un point de fonctionnement.

On s'intéresse dans cette partie à utiliser l'outil de géométrie différentielle, pour réaliser le découplage du modèle dynamique du robot manipulateur.

M.A.Henson & Seborg [Hen88] présente un "survey" des différentes techniques de linéarisation exacte, basées sur la géométrie différentielle. La liaison existante entre le découplage linéaire et le découplage non linéaire a été étudiée par Gras & Nijmeijer [Gra89]. Sastry & Isidori [Sas89] développe la version adaptative de la méthode de découplage et linéarisation exacte pour les systèmes à phase non minimale.

Le modèle du système est défini par le modèle bilinéaire suivant:

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}\tag{IV.17}$$

Où  $f^T(x) = [f_1(x) \dots f_n(x)]$ ,  $g(x) = [g_1(x) \dots g_m(x)]$ ,  $g_i^T = [g_{i1}(x) \dots g_{in}(x)]$   $i = 1, m$ .  
 $h^T(x) = [h_1(x) \dots h_p(x)]$ ,  $f_i, h_i, g_{ij}$ : fonctions réelles de  $R^n \rightarrow R$  infiniment différentiables.

$x^T = [x_1 \dots x_n]$ : vecteur d'état du système.

$u^T = [u_1 \dots u_m]$ : entrées de commande.

$y^T = [y_1 \dots y_m]$ : sorties du système.

D'une manière plus simplifiée on a:

$$\begin{aligned}\dot{x}_i &= f_i(x) + \sum_{k=1}^m g_{ik}(x)u_k \\ y_i &= h_i(x)\end{aligned}\tag{IV.18}$$

On s'intéresse au découplage de ce système, c'est à dire que la sortie  $y_i$  n'est influencée que par la commande  $u_i$ , donc  $m = p$ , le nombre d'entrées est égale au nombre de sorties.

En considérant le vecteur de sorties  $y$ , on cherche à trouver une expression mathématique liant directement  $y$  à  $u$ , en utilisant les dérivées successives de  $y$ , jusqu'à ce que la commande apparait pour la première fois [Isi85]:

$$L_f \lambda(x) = \sum_{i=1}^n \frac{\delta \lambda(x)}{\delta x_i} f_i\tag{IV.19}$$

Où  $\lambda(x)$ : fonctions réelles de  $R^n \rightarrow R$ .

$f$ : champ vecteur.

En utilisant l'équation (IV.19) on a [Gra89]:

$$y_i^{(d_i+1)} = L_f^{d_i+1} h_i(x) + L_g L_f^{d_i} h_i(x)u\tag{IV.20}$$

Avec  $y_i^k = L_f^k h_i(x)$ ;  $k = 1, d_i$ ,  $L_f^0 h_i(x) = h_i(x)$ .

et  $L_g L_f^{d_i} h_i(x) = [L_{g_1} L_f^{d_i} h_i \dots L_{g_m} L_f^{d_i} h_i]$

où  $d_i + 1 = \min \{k : L_g L_f^{k-1} h_i(x) \neq 0\}$ : appelé degré relatif.

Si le système est contrôlable, on a alors [Isi85][Sas89]:

$$d = \sum_{i=1}^m (d_i + 1) \leq n\tag{IV.20.a}$$

En faisant le changement de variable suivant [Isi85]:

$$z = \phi(x)\tag{IV.21}$$

Où  $\phi$ : matrice de transformation définis par:

$$z_1 = y_i = \phi_1(x) = h_i(x).$$

$$z_2 = y_i^{(1)} = \phi_2(x) = L_f h_i(x).$$

$$z_{d+1} = y_1^{(d_1)} = \phi_{d+1}(x) = L_f^{d_1} h_1(x). \quad (IV.22)$$

.....

$$z_{d-(d_m+1)} = y_m = \phi_{d-d_m}(x) = h_m(x).$$

$$z_{d-d_m} = y_m^{(1)} = \phi_{d-d_m+1}(x) = L_f h_m(x).$$

.....

$$z_d = y_m^{(d_m)} = \phi_d(x) = L_f^{d_m} h_m(x).$$

Si  $d < n$ , il est toujours possible de choisir  $\phi_{d+1}(x), \dots, \phi_n(x)$  de telle manière à avoir :

$$L_g \phi_i(x) = 0 \quad (IV.23)$$

Le choix de  $\phi_i(x)$  pour  $d+1 \leq i \leq n$  peut se faire de telle sorte que la matrice de transformation  $\phi(x)$  soit non singulière. On obtient alors [Isi85]:

$$\dot{z}_1 = z_2.$$

$$\dot{z}_2 = z_3.$$

.....

$$\dot{z}_{d_1} = z_{d_1+1}.$$

$$\dot{z}_{d_1+1} = b_1(z) + \sum_{i=1}^m a_{1i}(z) u_i. \quad (IV.24)$$

.....

$$\dot{z}_{d-(d_m+1)} = z_{d-d_m}.$$

$$\dot{z}_{d-d_m} = z_{d-d_m+1}.$$

.....

$$\dot{z}_d = b_m(z) + \sum_{i=1}^m a_{mi}(z) u_i.$$

et  $\dot{z}_{d+1} = \phi_{d+1}(x).$

.....

$$(IV.24.a)$$

$$\dot{z}_n = \phi_n(x).$$

avec  $y_1 = z_1.$

.....

$$y_m = z_{d-(d_m+1)}.$$

et  $b_i(z) = L_f^{d_i+1} h_i[\phi^{-1}(z)]$ ,  $i = 1, m.$

$$a_{ij}(z) = L_{g_i} L_f^{d_j} h_i[\phi^{-1}(z)]$$
,  $i, j = 1, m.$

Le système défini par (IV.24.a) caractérise la dynamique des zéros, car il est indépendant de  $u$  [Isi85].

Ainsi le système mis sous la forme de Brunovski définie par (IV.24)[Isi85]. On cherche une commande  $\mu$  telle que le système soit découplé. L'écriture matricielle de la relation (IV.20) est:

$$\begin{bmatrix} y_1^{(d_1+1)} \\ \vdots \\ y_m^{(d_m+1)} \end{bmatrix} = \begin{bmatrix} L_f^{d_1+1} h_1(x) \\ \vdots \\ L_f^{d_m+1} h_m(x) \end{bmatrix} + \begin{bmatrix} L_g L_f^{d_1} h_1(x) \\ \vdots \\ L_g L_f^{d_m} h_m(x) \end{bmatrix} u = \psi_1(x) + \psi_2(x)u. \quad (IV.25)$$

Le problème est alors de calculer une commande  $\mu$  qui découple le système, tel que:

$$u = \alpha(x) + \beta(x)\mu \quad (IV.26)$$

Où  $\alpha^T(x) = [\alpha_1(x) \dots \alpha_m(x)]$   
 $\beta_i^T(x) = [\beta_{i1}(x) \dots \beta_{im}(x)]$ ;  $i = 1, m$ .  
 $\beta^T(x) = [\beta_1(x) \dots \beta_m(x)]$

Si le système est découplable, on a:

$$\begin{bmatrix} y_1^{(d_1+1)} \\ \vdots \\ y_m^{(d_m+1)} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_m \end{bmatrix} \quad (IV.27)$$

En utilisant (IV.25) et (IV.27), si  $\psi_2(x)$  est non singulière:

$$u = [\psi_2(x)]^{-1} (\mu - \psi_1(x)) \quad (IV.28)$$

donc:

$$\begin{aligned} \alpha(x) &= -[\psi_2(x)]^{-1} \psi_1(x) \\ \beta(x) &= [\psi_2(x)]^{-1} \end{aligned} \quad (IV.29)$$

Ainsi le système est découplable par ce retour non linéaire. Chaque sortie  $y_i$  se comporte vis à vis de l'entrée  $\mu_i$  comme  $d_i$  integrateur..

Le retour non linéaire est représenté dans la figure IV.4.

Le découplage non linéaire, en utilisant la géométrie différentielle, a permis l'obtention d'un système linéaire totalement découplé. Par opposition de la linéarisation locale par approximation, cette méthode donne un linéarisation exacte, indépendamment du point de fonctionnement. On peut alors utiliser des contrôleurs linéaires de n'importe quel type.

Après avoir présenté la méthode de découplage, on considère maintenant le problème du découplage du robot de classe quatre. On définit les différentes fonctions du modèle du robot :

$$f^T(x) = [f_1(x) \dots f_n(x)]$$

$$\text{où } f_1(x) = x_2; f_2(x) = -\frac{f_1}{m_1 + m_3} x_2 - g; f_3(x) = x_4;$$

$$f_4(x) = -\frac{f_2}{m_3 J} x_4 - 2(x_5 - \frac{l_2}{2}) \frac{x_4 x_6}{J}; f_5(x) = x_5; f_6(x) = -\frac{f_3}{m_3} x_6 - (x_5 - \frac{l_2}{2}) x_4^2.$$

$$\text{et } x^T = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6];$$

$$h_1(x) = x_1; h_2(x) = x_3; h_3(x) = x_5.$$

$$g^T(x) = [g_1 \ g_2 \ g_3];$$

$$\text{avec } g_1^T(x) = \left[ 0 \ \frac{k_1}{m_1 + m_3} \ 0 \ 0 \ 0 \ 0 \right]; g_2^T(x) = \left[ 0 \ 0 \ 0 \ \frac{k_2}{m_3 J} \ 0 \ 0 \right]; g_3^T(x) = \left[ 0 \ 0 \ 0 \ 0 \ 0 \ \frac{k_3}{m_3} \right];$$

$$\text{et } u^T = [u_1 \ u_2 \ u_3]; y^T = [y_1 \ y_2 \ y_3]$$

On procède d'abord par le calcul des degrés relatifs  $d_i$ .

Pour  $i=1$  on a:

$$L_f h_1(x) = \sum_{i=1}^6 \frac{\delta h_1(x)}{\delta x_i} f_i = f_1 = x_2;$$

$$L_g h_1(x) = [L_{g_1} h_1(x) \ L_{g_2} h_1(x) \ L_{g_3} h_1(x)]$$

$$= \left[ \sum_{i=1}^6 \frac{\delta h_1(x)}{\delta x_i} g_{1i} \ \sum_{i=1}^6 \frac{\delta h_1(x)}{\delta x_i} g_{2i} \ \sum_{i=1}^6 \frac{\delta h_1(x)}{\delta x_i} g_{3i} \right] = [0 \ 0 \ 0]$$

$$L_f^2 h_1(x) = \sum_{i=1}^6 \frac{\delta L_f h_1(x)}{\delta x_i} f_i = f_2 = -\frac{f_1}{m_1 + m_3} x_2 - g;$$

$$L_g L_f h_1(x) = \left[ \sum_{i=1}^6 \frac{\delta L_f h_1(x)}{\delta x_i} g_{1i} \ \sum_{i=1}^6 \frac{\delta L_f h_1(x)}{\delta x_i} g_{2i} \ \sum_{i=1}^6 \frac{\delta L_f h_1(x)}{\delta x_i} g_{3i} \right] = [g_{12} \ g_{22} \ g_{32}]$$

$$= \left[ \frac{k_1}{m_1 + m_3} \ 0 \ 0 \right]$$

Ce dernier terme défini par (IV.20) n'est pas nul donc  $d_1 = 1$ . On procède de la même manière pour les autres entrées sorties. on a:

Pour  $i=2$  on a:

$$L_f h_2(x) = x_4; \quad L_g h_2(x) = [0 \ 0 \ 0] \cdot L_f^2 h_1(x) = -\frac{f_2}{m_2 J^*} x_5 - 2 \left( x_5 - \frac{l_2}{2} \right) \frac{x_4 x_6}{J^*};$$

$$L_g L_f h_1(x) = \left[ 0 \quad \frac{k_2}{m_3 J^*} \quad 0 \right] \neq 0$$

D'où  $d_2 = 1$ .

Pour  $i=3$  on a:

$$L_f h_3(x) = x_6; \quad L_g h_3(x) = [0 \ 0 \ 0], \quad L_f^2 h_3(x) = -\frac{f_3}{m_3} x_6 - \left( x_5 - \frac{l_2}{2} \right) x_4^2$$

$$L_g L_f h_1(x) = \left[ 0 \quad 0 \quad \frac{k_3}{m_3} \right] \neq 0$$

D'où  $d_3 = 1$ .

Donc de (IV.25), on a:

$$\psi_1(x) = \begin{bmatrix} -\frac{f_1}{m_1 + m_3} x_2 - g \\ -\frac{f_2}{m_2 J^*} x_4 - 2 \left( x_5 - \frac{l_2}{2} \right) \frac{x_4 x_6}{J^*} \\ -\frac{f_3}{m_3} x_6 + \left( x_5 - \frac{l_2}{2} \right) x_4^2 \end{bmatrix} \quad (IV.30)$$

et

$$\psi_2(x) = \begin{bmatrix} \frac{k_1}{m_1 + m_3} & 0 & 0 \\ 0 & \frac{k_2}{m_3 J^*} & 0 \\ 0 & 0 & \frac{k_3}{m_3} \end{bmatrix} \quad (IV.31)$$

La matrice  $\psi_2(x)$  est inversible, d'où:

$$\psi_2^{-1}(x) = \begin{bmatrix} \frac{m_1 + m_3}{k_1} & 0 & 0 \\ 0 & \frac{m_3 J^*}{k_2} & 0 \\ 0 & 0 & \frac{m_3}{k_3} \end{bmatrix} \quad (IV.32)$$

Donc de (IV.30) et (IV.32), le calcul de  $\alpha(x)$  et  $\beta(x)$ , à partir de (IV.29) est facile. On aboutit à la loi de commande en utilisant (IV.28):

$$\begin{aligned} u_1 &= \frac{1}{k_1} (m_1 + m_3)(\mu_1 + g) + f_1 x_2 \\ u_2 &= \frac{m_3}{k_2} (j^* \mu_2 + f_2 \frac{x_4}{m_2} + 2(x_5 - \frac{l_2}{2})x_4 x_6) \\ u_3 &= \frac{m_3}{k_3} (\mu_3 - (x_5 - \frac{l_2}{2})x_4^2 + \frac{l_3}{m_3} x_5) \end{aligned} \quad (IV.33)$$

Avec ce retour non linéaire, chaque articulation du robot se comporte comme un double intégrateur. Les degrés relatifs  $d_i = 1; i = 1, 3$ , nous permettent de confirmer l'absence de la dynamique zéro, ce qui facilite la commande du robot.

Dans le cas où le modèle d'état n'est pas connu avec une précision suffisante, les retours non linéaires  $\alpha(x)$  et  $\beta(x)$  seront erronés et le modèle en boucle ne sera pas découplé. En utilisant une commande autoajustable, on peut donner au robot n'importe quelle consigne à suivre même si les paramètres de ce dernier varient. L'architecture d'une telle commande est illustrée dans la

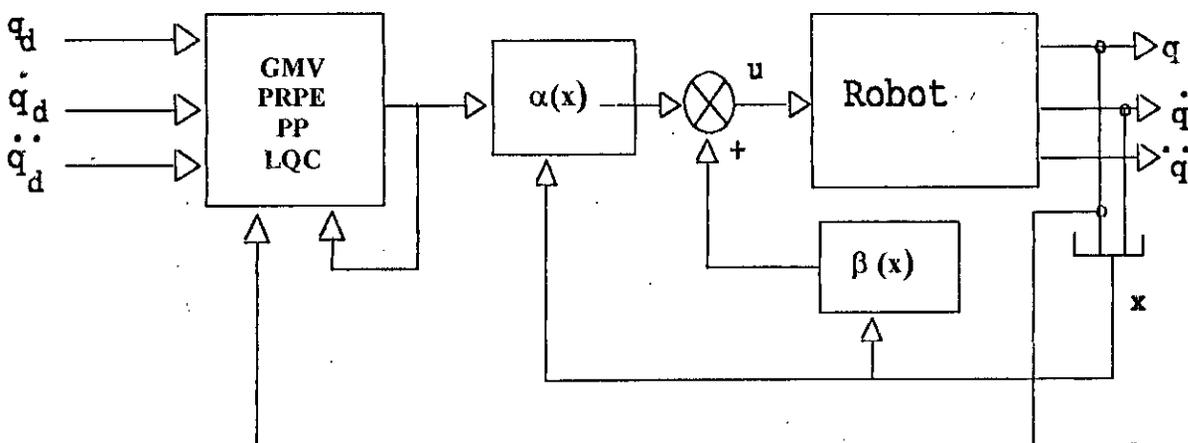


Figure IV.4 Commande hybride par découplage et linéarisation. (utilisant la géométrie différentielle).

figure IV.4.

## IV.5 Commande Adaptative à Perturbation

Cette technique est basée sur un modèle à perturbation obtenu par linearisation du modèle non linéaire du robot autour d'une trajectoire nominale.

### IV.5.1 Modèle à perturbation 'Perturbation Model'

On s'intéresse dans cette section à développer un modèle variationnel autour d'une région nominale. Considérant le modèle de la structure comme suit :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ x(t_0) = x_0 \end{cases} \quad (IV.34)$$

Où  $x^T(t) = [x_1(t) \dots x_n(t)]$ : vecteur d'état du système de dimension n.

$u^T(t) = [u_1(t) \dots u_m(t)]$ : vecteur de commande de dimension m.

$x_0$ : vecteur initial de l'état du système à l'instant  $t_0$ .

$f^T = [f_1 \dots f_n]$ : fonction vectorielle  $R^{n+m} \rightarrow R^n$ , continue et deux fois dérivable (pour des raisons techniques [Ath71]).

#### Remarque

Le modèle considéré dans ce cas est invariant dans le temps (c'est à dire  $f(\dots)$  ne dépend pas explicitement du temps). Dans le cas contraire le modèle considéré sera:  $\dot{x}(t) = f(x(t), u(t), t)$ .

La sortie du système peut être une combinaison non linéaire, invariante dans le temps des états et définie par :

$$y(t) = g(x(t)) \quad (IV.35)$$

Où  $g^T = [g_1 \dots g_r]$ : fonction vectorielle  $R^n \rightarrow R^r$ , continue et deux fois dérivable.

La détermination de l'état  $x(\tau)$ , et la sortie  $y(\tau)$ , pour  $\tau > t$ , peut être effectuée directement à partir du modèle définie par (IV.34) et (IV.35). Ainsi la connaissance des conditions initiales sur l'état ( $x_0$ ), nous permet le calcul d'un état idéal et d'une sortie idéale pendant un intervalle de temps  $[t_0, T]$ . On obtient alors les résultats suivants:

- 1- Une fonction "déterministe idéale" de l'entrée:  $u_0(t), t \in [t_0, T]$ .
- 2- Une trajectoire d'état "déterministe idéale":  $x_0(t), t \in [t_0, T]$ .
- 3- Une sortie "déterministe idéale":  $y_0(t), t \in [t_0, T]$ .

Toutes ces quantités sont reliées par:

$$\begin{cases} \dot{x}_0(t) = f(x_0(t), u(t)) = x_0 \\ y_0(t) = g(x_0(t)) \end{cases} \quad (IV.36)$$

L'état  $x_0(t)$  représente l'évolution de l'état désiré du système, à partir d'un état initial  $x_0$ . L'utilisation d'un calculateur numérique et des résultats pratiques, nous permettent la

détermination de  $u_0(t)$  et  $x_0(t)$  par simulation. Cependant, il existe une autre approche systématique pour la détermination de  $u_0(t)$  et  $x_0(t)$ . Cette dernière consiste à la résolution d'un problème non linéaire à critère optimal. Cela nécessite le choix d'une fonction coût scalaire (non obligatoirement quadratique) définie par [Ath71] :

$$I = \Phi(x(T)) + \int_{t_0}^T L(x(t), u(t)) dt \quad (IV.37)$$

Où  $x(T)$  est l'état terminal désiré.

$\Phi(\cdot), L(\cdot, \cdot)$  : fonctions scalaires non linéaires.

Le problème peut être formulé de la façon suivante [Ath71] :

*'Ayant le système défini par (IV.34) et l'état initial  $x_0$ . Trouver  $u_0(t)$  et  $x_0(t)$  pour  $t \in [t_0, T]$ , qui minimise la fonctionnelle définie par (IV.37).'*

La résolution de ce problème, utilisant la théorie du calcul variationnel [Els61], ou des techniques numériques itératives, permet la détermination de  $u_0(t)$  et  $x_0(t)$ .

Une fois  $x_0(t)$  est calculé, la réponse idéale  $y_0(t)$  est trouvée en utilisant (IV.35). Ainsi le triplet  $[x_0(t), u_0(t), y_0(t)]$  pour  $t \in [t_0, T]$  peut être calculé en temps réel ou mémorisé dans des mémoires de masse pour être utilisé ultérieurement.

Dans tout ce qui précède, on a résolu le problème (IV.37), en utilisant le modèle de connaissance (IV.34) et (IV.35), obtenu par modélisation et non pas du système réel. Considérant le triplet  $[x(t), u(t), y(t)]$  qui définit la commande réelle, l'état réel et la sortie réelle du processus physique. Ce triplet peut être obtenu par des mesures exactes pendant l'évolution du processus. Le problème qui se pose est alors :

*'Pour une entrée excitant le système réel et son modèle, obtient-t-on les mêmes états et les mêmes sorties ?'*

D'une autre manière on a [Ath71] :

Pour  $u(t) = u_0(t); t \in [t_0, T]$ .

A-t-on :

$$\begin{cases} x(t) = x_0(t) & \text{pour tout } t \in [t_0, T]? \\ y(t) = y_0(t) & \text{pour tout } t \in [t_0, T]? \end{cases} \quad (IV.38)$$

En général la réponse est « Non ». D'une part, le calcul de  $x_0(t)$  est réalisé en utilisant un modèle mathématique. Ce modèle est obtenu en considérant des lois physiques régissant les phénomènes les plus connus. Dans le cas de la dynamique des structures, les phénomènes mal connus sont les non linéarités géométriques et les non linéarités de comportement. D'autre part, la valeur des paramètres utilisés dans le modèle mathématique est nominale, déterminée par expérimentation, telles que les masse et les amortissements visqueux.

Dans le cas où le modèle de connaissance est déterminé d'une façon précise (en considérant les phénomènes agissant sur le système), cette différence peut être causée par le choix de la condition initiale  $x_0(t_0)$  qui peut être différente de  $x_0(t)$ . Une erreur de  $x_0(t) - x_0(t_0)$  peut engendrer des erreurs importantes accumulées au cours du temps.

L'analyse précédente permet de définir un objectif de poursuite. L'état du système  $x(t)$  et la commande  $u(t)$  doivent suivre respectivement l'état idéal  $x_0(t)$  et la commande idéale  $u_0(t)$ . Ainsi on définit les équations variationnelles suivantes :

- Vecteur d'état de perturbation :  
 $\delta x(t) = x(t) - x_0(t)$
- Vecteur de sortie de perturbation :  
 $\delta y(t) = y(t) - y_0(t)$
- Vecteur de correction de commande :  
 $\delta u(t) = u(t) - u_0(t)$

On suppose que la commande réelle  $u(t)$ , état réel  $x(t)$  et la sortie réelle sont reliés par le modèle défini par (IV.34) et (IV.35) (puisque'on a pas d'autre modèle). On a alors :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = g(x(t)) \end{cases} \quad (IV.39)$$

La commande idéale  $u_0(t)$ , l'état idéal  $x_0(t)$  et la sortie idéale  $y_0(t)$  sont reliés par le système d'équation défini par (IV.36). Le développement en série de Taylor de  $f(\cdot, \cdot)$  et  $g(\cdot)$  autour de  $x_0(t), u_0(t)$  donne [Gue95][Ath75]:

$$\begin{cases} f(x(t), u(t)) = f(x_0(t), u_0(t)) + \left. \frac{\partial f}{\partial x} \right|_0 \delta x(t) + \left. \frac{\partial f}{\partial u} \right|_0 \delta u(t) + \alpha_0(\delta x(t), \delta u(t)) \\ g(x(t)) = g(x_0(t)) + \left. \frac{\partial g}{\partial x} \right|_0 \delta x(t) + \beta_0(\delta x(t)) \end{cases} \quad (IV.40)$$

Où  $\alpha_0(\delta x(t), \delta u(t))$  et  $\beta_0(\delta x(t))$  sont les termes d'ordre supérieur dans le développement en série de Taylor.

Le système d'équation (IV.40) peut se mettre sous la forme suivante ( en utilisant (IV.36) et (IV.39)):

$$\begin{cases} \dot{\delta x}(t) = A_0(t)\delta x(t) + B_0(t)\delta u(t) + \alpha_0(\delta x(t), \delta u(t)) \\ \delta y(t) = C_0(t)\delta x(t) + \beta_0(\delta x(t)) \end{cases} \quad (IV.41)$$

Où  $A_0(t) = \left. \frac{\partial f}{\partial x} \right|_{x_0(t), u_0(t)}$  est une matrice variable dans le temps de dimension  $n \times n$ , obtenue par calcul de la matrice Jacobienne  $\frac{\partial f}{\partial x}$  au point variable  $x_0(t)$  et  $u_0(t)$ .

$B_0(t) = \left. \frac{\partial f}{\partial u} \right|_{x_0(t), u_0(t)}$  est une matrice variable dans le temps de dimension  $n \times m$ , obtenue par calcul de la matrice Jacobienne  $\frac{\partial f}{\partial u}$  au point variable  $x_0(t)$  et  $u_0(t)$ .

$C_0(t) = \left. \frac{\partial g}{\partial x} \right|_{x_0(t)}$  est une matrice variable dans le temps de dimension  $r \times n$ , obtenue par calcul de la matrice Jacobienne  $\frac{\partial g}{\partial x}$  au point variable  $x_0(t)$ .

Les équations définies par (IV.41), incluant les termes d'ordre supérieur, représentent la relation exacte entre  $\delta x(t), \delta u(t)$  et  $\delta y(t)$ . Le modèle à perturbation continu (continuous perturbation model) est obtenu en négligeant les termes  $\alpha_0(\delta x(t), \delta u(t))$  et  $\beta_0(\delta x(t))$ . On obtient ainsi [Ath71]:

$$\begin{cases} \delta \dot{x}(t) = A_0(t)\delta x(t) + B_0(t)\delta u(t) \\ \delta \dot{y}(t) = C_0(t)\delta x(t) \end{cases} \quad (IV.42)$$

Ce modèle est la forme d'état standard d'un système linéaire variable dans le temps (LTVS Linear Time Varying System).

### IV.5.2 Modèle aux différences à perturbation.

La solution du système différentiel défini par (IV.42) est [Fri87]:

$$\delta x(t) = \phi(t, t_0)\delta x(t_0) + \int_{t_0}^t \phi(t - \tau, t_0) B_0(\tau)\delta u(\tau) d\tau \quad (IV.43)$$

Où  $\phi(t, t_0)$  : matrice de transition du système.

Le processus définissant le robot manipulateur est commandé par ordinateur. Le modèle discret de (IV.43) est nécessaire. En remplaçant  $t$  par  $(k+1)T$  et  $t_0$  par  $kT$ , où  $T$  est la période d'échantillonnage. Le modèle à perturbation discret est exprimé par [Buh83]:

$$\delta x(k+1) = \bar{A}(k)\delta x(k) + \bar{B}(k)\delta u(k) \quad (IV.44)$$

Où  $k+1$  et  $k$  représentent respectivement  $(k+1)T$  et  $kT$ .

$$\text{Et } \bar{A}(k) = \phi((k+1)T, kT); \bar{B}(k) = \int_{kT}^{(k+1)T} \phi((k+1)T - \tau, kT) B_0(\tau) d\tau.$$

L'équation de la sortie (IV.42) sous sa forme discrète est défini par:

$$\delta y(k) = \bar{C}(k)\delta x(k) \quad (IV.45)$$

Où  $\bar{C}(k) = C_0(kT)$ .

Le modèle d'état discret, défini par (IV.44) et (IV.45), exprime une relation de récurrence entre les états du système. La combinaison de L'équation d'état et de l'équation de sortie, permet d'obtenir la forme canonique du modèle au différence [Toi84][Koi80][Goo81][Dio86]:

$$\begin{bmatrix} A_{11}(q^{-1}) & \dots & A_{1r}(q^{-1}) \\ \dots & \dots & \dots \\ A_{r1}(q^{-1}) & \dots & A_{rr}(q^{-1}) \end{bmatrix} \delta y(k) = \begin{bmatrix} B_{11}(q^{-1}) & \dots & B_{1m}(q^{-1}) \\ \dots & \dots & \dots \\ B_{r1}(q^{-1}) & \dots & B_{rm}(q^{-1}) \end{bmatrix} \delta u(k) + e(k) \quad (IV.46)$$

$$\text{Où } A_{ij}(q^{-1}) = \delta_{ij} + \sum_{l=1}^{n_{ij}} a_{ijl} q^{-l}; i, j = 1, r;$$

$$B_{ij}(q^{-1}) = q^{-d_{ij}} \sum_{l=1}^{m_{ij}} b_{ijl} q^{-l}; i = 1, r; j = 1, m; d_{ij} \geq 0.$$

$q^{-1}$  Opérateur retard;  $\delta y(t-1) = q^{-1} \delta y(t)$ .

$\delta_{ij}$  : symbole de Kroneker.

$e(k)$  : vecteur d'erreur d'échantillonnage ou de perturbations extérieures de dimension  $r$ .

$d_{ij}$  : retard (entier) du système entre  $\delta y_i$  et  $\delta u_j$ .

Le modèle ainsi obtenu est le modèle aux différences à perturbation (perturbation difference model).

### IV.5.3 Formulation du problème de la commande "APC".

L'équation (IV.46) donne un modèle à perturbation aux différences entrées- sorties multivariable. L'objectif est de trouver une commande  $\delta u(k)$  qui fait tendre  $\delta y(k)$  vers zéro. Le calcul direct d'une telle commande en se basant sur ce modèle nécessite beaucoup de calcul. Le modèle (IV.46) est obtenu par linéarisation des équations dynamiques du système (robot) au point nominal. La commande nominale est calculée à partir de l'algorithme récursif de Newton-Euler (voir chapitre I). Ainsi l'effet du couplage est moindre, ce qui nous permet de définir le modèle de représentation suivant [Mei87]:

$$A_i(q^{-1})\delta y_i(k) = q^{-d_i} B_i(q^{-1})\delta u_i(k) + h_i(k) \quad i = 1, m \quad (IV.47)$$

Où  $A_i(q^{-1}) = 1 + \sum_{l=1}^{m_i} a_{il} q^{-l}$ ;  $B_i(q^{-1}) = \sum_{l=1}^{m_i} b_{il} q^{-l}$ .  $d_i$ : retard (entier) du système entre  $\delta y_i$  et  $\delta u_i$ .  
 $h_i(k) = e_i(k) + \sum_{j=1}^{m} B_{ij}(q^{-1})\delta u_j(k) - \sum_{j=1}^{m} A_{ij}(q^{-1})\delta y_j(k); \quad i = 1, m.$

Le modèle (IV.47) est une approximation acceptable, du fait que le couplage est compensé par le calcul d'une commande anticipatrice.

Dans le cas des robots manipulateurs, l'utilisation de l'algorithme de Newton-Euler (NE) comme dynamique inverse du robot, compense l'interaction entre les différentes liaisons. Cette dernière est réalisée par une action anticipatrice (feedforward) sur la commande. La réduction de l'erreur de position et de vitesse autour de la trajectoire nominale est réalisée par l'action de retour (feedback). La commande de correction  $\delta u(k)$ , calculée à partir du modèle à perturbation permet d'accomplir cette tâche. L'identification du modèle défini par (IV.47) se fait en temps réel, pour détecter d'éventuelles variations paramétriques du modèle du robot.

Différentes stratégies de commande peuvent être utilisées pour la régulation du modèle à perturbation autour de la trajectoire nominale. Les paramètres du régulateur synthétisés sont ajustés à chaque période d'échantillonnage, pour le calcul de la commande nécessaire. La commande totale agissant sur le manipulateur est la somme d'une commande calculée à partir de l'algorithme de NE est d'une commande à perturbation calculée à partir du modèle au différence à perturbation [Fu87][Mei87]. Cette stratégie de commande (la commande APC) transforme le problème de commande non linéaire en un problème linéaire autour d'une trajectoire nominale. La structure de la commande APC est illustrée par le schéma de la figure IV.5.

On s'intéresse maintenant à développer quelques algorithmes de commande, exposés dans le chapitre III, en utilisant le modèle à perturbation.

Athans [44] utilise le modèle à perturbation pour développer la commande LQG. Le modèle aux différences à perturbation est utilisé par Liu [22][23], pour la commande des robots manipulateurs.

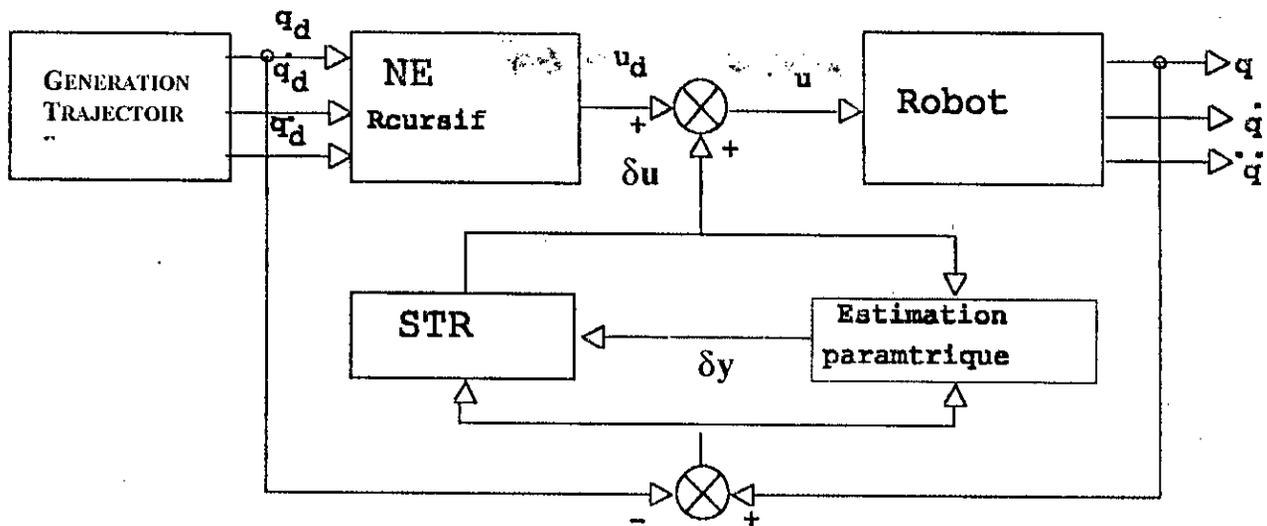


Figure IV.5 Structure de la commande APC.

Dans ce qui suit, on présente le développement de la commande APC dans le cas de la GMV. Tous les autres algorithmes peuvent être étendus de la même manière.

Le modèle du système est un modèle ARMAX défini par (III.15 et IV.47). Le critère de commande est défini par [Mei87]:

$$J_i = E \left[ (P_i \delta y_i(t+d+1))^2 + (Q_i \delta u_i(t))^2 \right] \quad i = 1, 3 \quad (IV.48)$$

Où  $P_i$  et  $Q_i$  définis par (III.23).

Le critère (IV.48) exprime un problème de régulation, qui est bien adapté à notre problème. Dans le cas du modèle à perturbation, on s'intéresse à faire tendre  $\delta y_i$  vers zéro avec une dynamique  $P_i(q^{-1})$ .

La loi de commande qui minimise ce critère est :

$$\delta u_i(t) = \frac{-R_i \delta y_i(t) - P_{D_i}(1) S_i(1) h_i}{P_{D_i}(S_i + Q_i C_i)} \quad (IV.49)$$

Où  $P_{D_i}$ ,  $Q_i$  et  $C_i$  : définis dans chapitre III.

$R_i$ ,  $S_i$  et  $S_i$  : solution de L'équation Diophantine.

Ainsi tous les autres algorithmes se développent de la même manière.

## IV.6 Commande linéarisante. "Computed Torque Control".

Dans le cas où les paramètres dynamiques sont connus. Le modèle du robot est régi par le système différentiel suivant:

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) + f(\dot{q}) = \tau \quad (IV.50)$$

Où  $M, V, G, f$  et  $\tau$  sont définis dans (IV.1).

La boucle de la linérisation est réalisée par le choix d'un couple  $\tau$  à appliquer au robot, de la forme suivante [Cra87][Ben91]:

$$\tau = M_o(q)\tau' + V_o(q, \dot{q}) + G_o(q) + f_o(\dot{q}) \quad (\text{IV.51})$$

Où  $M_o, V_o, G_o$ , et  $f_o$  sont les estimés de  $M, V, G$ , et  $f$ .

La combinaison des équations (IV.50) et (IV.51) donne :

$$M(q)\ddot{q} + V(q, \dot{q}) + G(q) + f(\dot{q}) = M_o(q)\tau' + V_o(q, \dot{q}) + G_o(q) + f_o(\dot{q}) \quad (\text{IV.52})$$

Dans le cas où le modèle du robot est connu avec une précision suffisante, on a :

$$\begin{aligned} M(q) &\approx M_o(q); G(q) \approx G_o(q) \\ V(q, \dot{q}) &\approx V_o(q, \dot{q}); f(q) \approx f_o(q) \end{aligned} \quad (\text{IV.53})$$

L'équation (IV.51) devient alors :

$$\ddot{q} = \tau' \quad (\text{IV.54})$$

dans le cas où  $M^{-1}(q)$  est nonsingulière.

Ce qui transforme le problème de commande du robot en celui de la commande de  $n$  doubles intégrateurs découplés. L'entrée  $\tau'$  peut être interprétée comme une commande de la boucle externe qui devra asservir le système. Cette commande est spécifiée en terme d'accélération. Elle peut être obtenue par un compensateur linéaire proportionnel dérivé (PD) ou un retour d'état pour le système linéarisé :

$$\tau' = \ddot{q}_d - k_v \dot{e}(t) - k_p e(t) \quad (\text{IV.55})$$

$$\begin{aligned} e(t) &= q(t) - q_d(t) \\ \dot{e}(t) &= \dot{q}(t) - \dot{q}_d(t) \end{aligned} \quad (\text{IV.56})$$

Où  $k_p$  et  $k_v$  : matrices de dimensions appropriées.

Le vecteur  $e(t)$  représente l'erreur de suivi des trajectoires et  $\dot{e}(t)$  sa dérivée par rapport au temps

$$k_{v_i} = 2\xi_i \omega_{0_i}; k_{p_i} = \omega_{0_i}^2 \quad (\text{IV.57})$$

ou erreur en vitesse du système

Les matrices des gains de retour  $k_p$  et  $k_v$  sont diagonales définies positives et peuvent être choisies de manière à imposer une dynamique de rejection de l'erreur de suivi des trajectoires désirées. On peut imposer au système de se comporter en boucle fermée comme un système multivariable du second ordre avec pour amortissement  $\xi$  et pulsation  $\omega_0$ . Ceci peut être réalisé par [Cra88]:

En remplaçant l'expression de  $\tau'$  (IV.55) dans L'équation (IV.54), on obtient L'équation de l'erreur en boucle fermée:

$$M(q)(\ddot{e}(t) + k_v \dot{e}(t) + k_p e(t)) = 0 \quad (IV.58)$$

L'équation (IV.58) a pour solution un signal  $e(t)$  qui tend exponentiellement vers zéro, car la

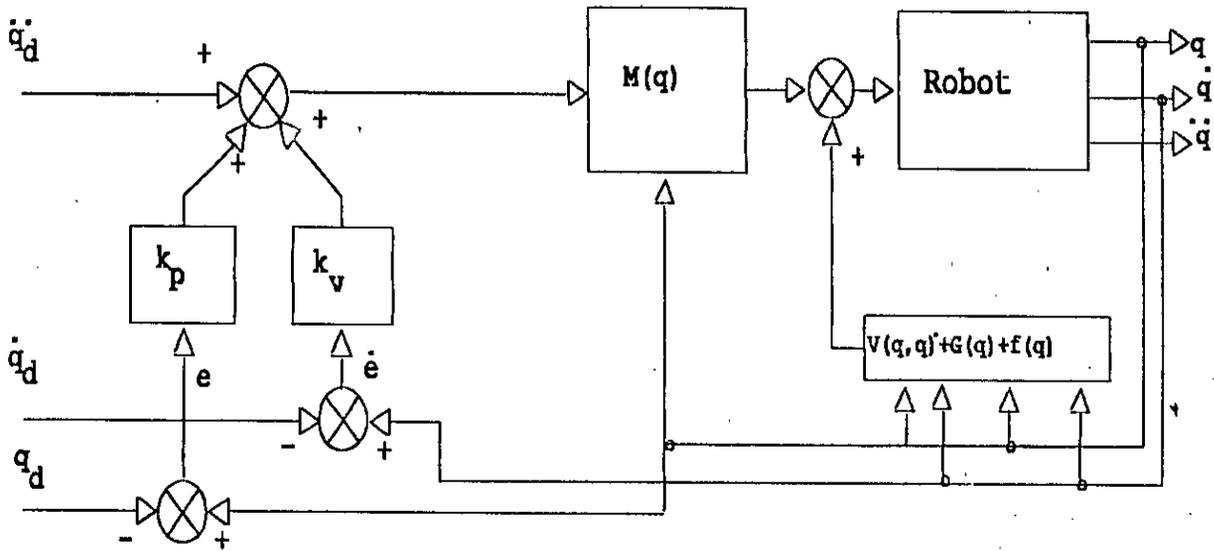


Figure IV.6 Architecture de la commande linéarisante à paramètres connus.

matrice d'inertie est nonsingulière [Ben91]. On peut donc conclure que le système en BF, avec cette commande, dans le cas où les paramètres dynamiques sont bien connus, est exponentiellement stable.

L'architecture d'une telle loi de commande est représentée sur la figure IV.6.

## IV.7 Commande linéarisante adaptative.

Pour faciliter la manipulation des termes non linéaires dans le modèle de connaissance du robot, le manipulateur est modélisé sous forme compacte suivante [Cra87][Cra88]:

$$\tau = M(q)\ddot{q} + Q(q, \dot{q}) \quad (IV.59)$$

Où  $Q(q, \dot{q}) = V(q, \dot{q}) + G(q) + f(\dot{q})$ .

Dans le cas où on utilise les estimés des paramètres dynamiques du robot, la commande (IV.59) devient:

$$\tau = \hat{M}(q)\tau' + \hat{Q}(q, \dot{q}) \quad (IV.60)$$

Où  $\hat{M}(q)$  et  $\hat{Q}(q, \dot{q})$  sont les estimés de  $M(q)$  et  $Q(q, \dot{q})$ .

Le terme  $\tau'$  est choisi de la même manière que (IV.55):

$$\tau' = \ddot{q}_d - k_v \dot{e}(t) - k_p e(t) \quad (IV.61)$$

Des équations (IV.59), (IV.60) et (IV.61) on obtient:

$$M(q)\ddot{q} + Q(q, \dot{q}) = \hat{M}(q)[\ddot{q}_d - k_v \dot{e}(t) - k_p e(t)] + \hat{Q}(q, \dot{q}) \quad (IV.62)$$

En ajoutant et en retranchant le terme  $\hat{M}(q, \dot{q})\ddot{q}$ , du terme de droite de L'équation (IV.62), on obtient:

$$\ddot{e}(t) + k_v \dot{e}(t) + k_p e(t) = -M^{-1}(q) [\tilde{Q} + \tilde{M}\ddot{q}] \quad (\text{IV.63})$$

où  $\tilde{Q} = \hat{Q}(q, \dot{q}) - Q(q, \dot{q})$ ;  $\tilde{M} = \hat{M}(q) - M(q)$ .

Pour l'élaboration de l'algorithme d'identification, l'écriture paramétrique du modèle du robot s'avère nécessaire. On réécrit le modèle du robot défini par (IV.59), sous la forme suivante [Cra88]:

$$\tau_i = \sum_{j=1}^{a_i} m_{ij} f_{ij}(q, \dot{q}) + \sum_{j=1}^{b_i} q_{ij} g_{ij}(q, \dot{q}); \quad i = 1, n \quad (\text{IV.64})$$

Où  $m_{ij}$  et  $q_{ij}$  sont des paramètres traduisant les masses, les éléments du tenseur d'inertie, longueurs et coefficients de frottement.

$f_{ij}(q, \dot{q})$  et  $g_{ij}(q, \dot{q})$  sont des fonctions non linéaires issues du modèle du robot après quelques arrangements.

$a_i$  : nombre des paramètres  $m_{ij}$ .

$b_i$  : nombre des paramètres  $q_{ij}$ .

L'écriture de L'équation (IV.64) sous forme matricielle donne :

$$\tau = W(q, \dot{q}, \ddot{q})\theta \quad (\text{IV.65})$$

Où  $W(q, \dot{q}, \ddot{q})$  : matrice englobant toutes les fonctions non linéaires  $f_{ij}$  et  $g_{ij}$ .

$\theta$  : vecteur de paramètres englobant  $m_{ij}$  et  $q_{ij}$  de dimension  $\sum_{i=1}^n (a_i + b_i)$ .

On suppose que le modèle estimé est construit avec une estimation de  $r_i$  coefficients  $m_{ij}$  et  $s_i$  coefficients  $q_{ij}$  pour l'articulation  $i$  considérée. Le modèle estimé s'écrit alors sous la forme suivante [Cra88]:

$$\hat{\tau}_i = \sum_{j=1}^{r_i} \hat{m}_{ij} f_{ij}(q, \dot{q}) + \sum_{j=1}^{s_i} \hat{q}_{ij} g_{ij}(q, \dot{q}); \quad i = 1, n \quad (\text{IV.66})$$

Sous forme matricielle, L'équation (IV.66) devient:

$$\hat{\tau} = W(q, \dot{q}, \ddot{q})\hat{\theta} \quad (\text{IV.67})$$

Où  $\hat{\theta}^T = [\theta_1 \dots \theta_r]$  avec  $r \leq \sum_{i=1}^n (s_i + r_i)$ .

L'algorithme d'adaptation calcule l'évolution des estimés en utilisant le signal d'erreur filtré. Ce signal d'erreur filtré est défini par [Cra88]:

$$e_i(t) = \dot{e}(t) + \psi e(t) \quad (\text{IV.68})$$

$$\hat{\theta} = -\Gamma W^T(q, \dot{q}, \ddot{q}) \hat{M}^{-1}(q) e_i(t) \quad (\text{IV.69})$$

Où  $\psi = \text{diag}(\psi_1 \dots \psi_n)$ . Donc l'algorithme d'adaptation est :

Où  $\Gamma = \text{diag}(\gamma_1 \dots \gamma_r)$  et  $\gamma_i > 0$ .

C'est un algorithme du type Intégral [Gue95a].

## IV .8 Commande adaptative à modèle de référence. (MRAC Model Reference Adaptive Control).

Dans le cas où les paramètres du système sont inconnus ou variables dans le temps, la commande linéaire par poursuite d'un modèle n'est plus applicable. On utilise alors la commande adaptative (adaptive model following control). L'implémentation d'une telle loi, peut se faire de

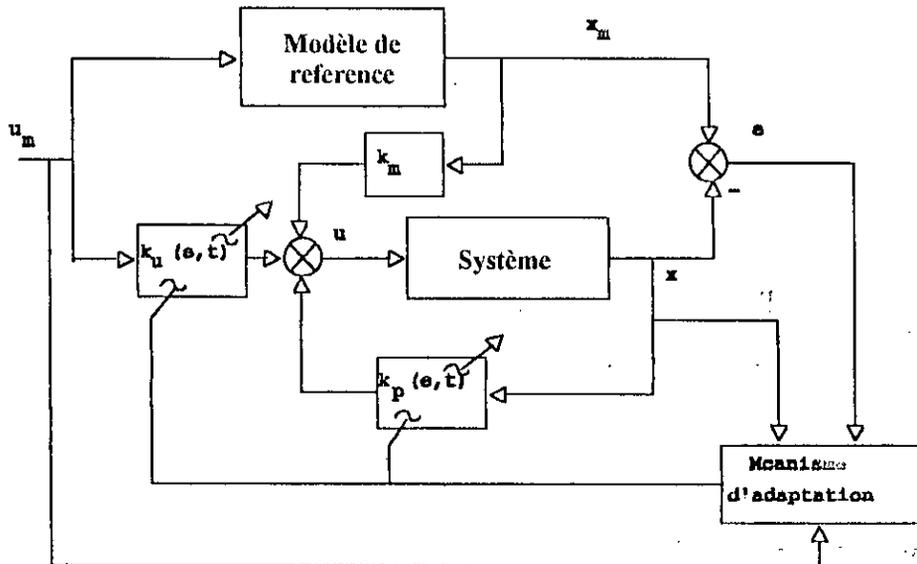


Figure IV.7 AMFC parallèle avec l'adaptation paramétrique.

deux manières [Lan79]:

- 1- Adaptation paramétrique ( figure IV.7 ).

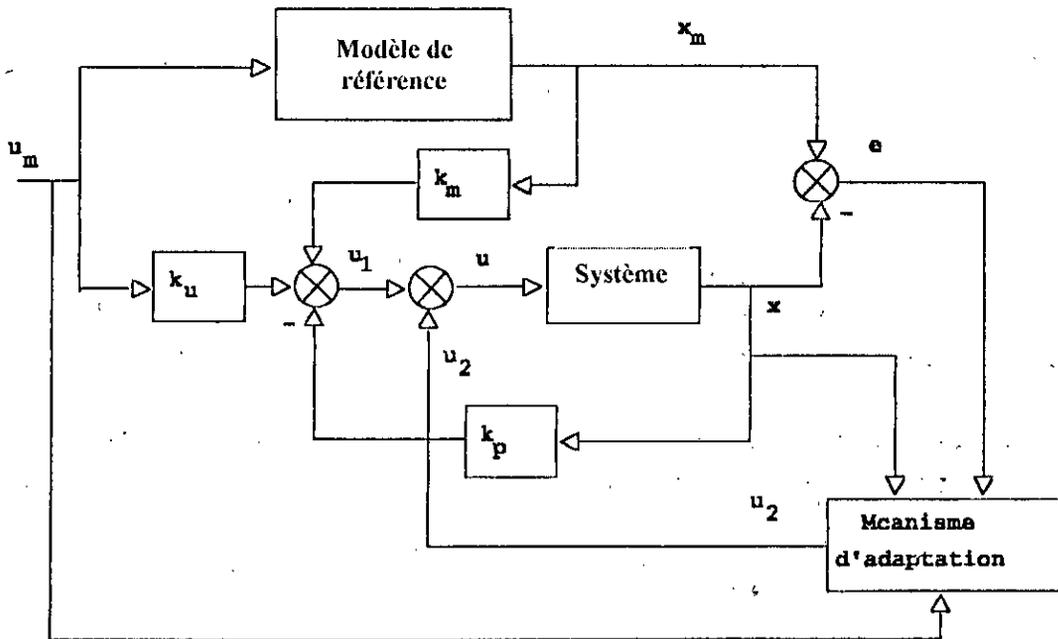


Figure VI.8 AMFC avec l'adaptation par signal de synthèse.

- 2- Adaptation par signal de synthèse ( figure IV.8 ).

Dans le cas de l'adaptation paramétrique, l'entrée du système est exprimée par [Lan78]:

$$u = -k_p(e, t)x + k_m x_m + k_u(e, t)u_m \quad (IV.70)$$

Où  $k_p(\cdot)$  et  $k_u(\cdot)$  : matrices variables dans le temps dépendant de  $e$ .

$k_m$  : matrice constante.

$k_p(\cdot)$  et  $k_u(\cdot)$  : peuvent être exprimées par:

$$\begin{aligned} k_p(e,t) &= k_p - \delta k_p(e,t) \\ k_u(e,t) &= k_u + \delta k_u(e,t) \end{aligned} \quad (IV.71)$$

Où  $k_p$  et  $k_u$  sont des matrices constantes, déterminées à partir du LMFC (chapitre III).

Avec cette décomposition, on peut écrire:

$$u = u_1 + u_2 \quad (IV.72)$$

Où  $u_1 = -k_p x + k_m x_m + k_u u_m$  et  $u_2 = \delta k_p(e,t)x + \delta k_u(e,t)u_m$ .

L'entrée  $u_1$  du système représente la commande linéaire, l'entrée  $u_2$  est la contribution de la commande adaptative.

Cette technique est connue sous le nom de l'adaptation par signal de synthèse. On voit bien que ces deux techniques sont équivalentes.

L'algorithme d'adaptation paramétrique est [gue95a][Lan79]:

$$\begin{aligned} \delta k_p(e,t) &= \delta k_p(V,t) = \int_0^t \phi_1(V,t,\tau) d\tau + \phi_2(V,t) + \delta k_p(0) \\ \delta k_u(e,t) &= \delta k_u(V,t) = \int_0^t \psi_1(V,t,\tau) d\tau + \psi_2(V,t) + \delta k_u(0) \end{aligned} \quad (IV.73)$$

Où  $V=De$ .

La détermination de  $D$  est déduite du fait que la chaîne directe est SPR et ceci par résolution de l'équation de Lyapounov [Lan79]

Landau [Lan79] propose plusieurs choix des matrices  $\phi_1, \phi_2, \psi_1$  et  $\psi_2$  qui satisfont l'inégalité de Popov. Pour que l'inégalité soit satisfaite, il suffit que les deux termes de gauche satisfaisant le même type d'inégalité. En faisant le choix du premier lemme (annexe D de la référence [Lan79]), on a:

$$\begin{aligned} \phi_1(V,t,\tau) &= F_A(t-\tau)V(\tau)[G_A x(\tau)]^T, \phi_2(V,t) = F_{A'}(t)V(t)[G_{A'} x(t)]^T \\ \psi_1(V,t,\tau) &= F_B(t-\tau)V(\tau)[G_B u(\tau)]^T, \psi_2(V,t) = F_{B'}(t)V(t)[G_{B'} u(t)]^T, \text{ Pour } \tau \leq t. \end{aligned}$$

Où  $F_A(t-\tau)$  et  $F_B(t-\tau)$  sont des matrices définies positives, dont les transformées de Laplace, sont des matrices de transfert positives réelles possédant un pôle à l'origine ( $s=0$ ).

$G_A$  et  $G_B$  sont des matrices constantes définies positives.

$F_{A'}(t), F_{B'}(t), G_{A'}(t)$  et  $G_{B'}(t)$  sont des matrices définies positives variables dans le temps pour  $t \geq 0$ .

◆ *Quelques choix particuliers.*

Les matrices  $\phi_1(\cdot), \phi_2(\cdot), \psi_1(\cdot)$  et  $\psi_2(\cdot)$  peuvent être choisies sous la forme suivante:

$$\begin{aligned} F_A(t-\tau) &= F_A > 0 \\ F_B(t-\tau) &= F_B > 0 \\ F_{A'}(t) &= F_{A'}; F_{B'}(t) = F_{B'} \\ G_{A'}(t) &= G_{A'}; G_{B'}(t) = G_{B'} \end{aligned}$$

On obtient alors une loi d'adaptation du type PI. D'autres lois existent, pour un choix particulier de ces matrices et aboutit à une loi du type "Integral+Relay adaptation" [M'si93]. La convergence globale asymptotique de l'erreur est assurée [Lan79].

#### VI.4 Synthèse d'un contrôleur minimal MCS. (Minimum controller synthesis).

La synthèse de la commande adaptative à modèle de référence MRAC, nécessite la

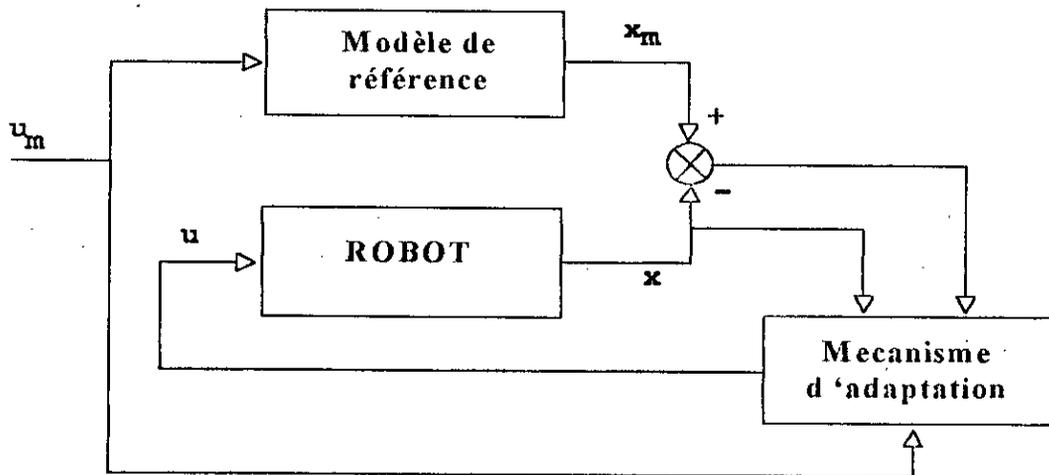


Figure IV.9 Structure de base du contrôleur MCS.

disponibilité d'un modèle de connaissance linéaire, vérifiant les conditions d'Erzenberger, pour le calcul de  $k_u$  et  $k_p$ . On cherche dans cette partie à développer un contrôleur avec le minimum d'information sur le système (tel que le degré). C'est un contrôleur dans lequel, la commande provient totalement du mécanisme d'adaptation et ne suppose aucune implémentation préalable d'un quelconque contrôleur non adaptatif. C'est le contrôleur MCS [Sto89]. La structure d'un tel contrôleur est illustrée dans la figure IV.9.

Ce contrôleur peut être déduit directement du MRAC, par imposition de l'hypothèse suivante :

$$k_m = k_p = k_u = 0 \quad (IV.74)$$

La synthèse d'un tel contrôleur est similaire à la précédente, en utilisant l'outil des systèmes adaptatifs à modèle de référence.

Le modèle de référence est décrit par:

$$\dot{x}_m = A_m x_m + B_m u_m \quad (IVI.75)$$

$A_m$  : matrice d'Hurwitz [Lan79].

Le système est défini par:

$$\dot{x} = A x + B u \quad (IV.76)$$

La commande définie par (IV.70) et (IV.71) devient :

$$u = \delta k_p(e, t)x + \delta k_u(e, \tau) u_m \quad (IV.77)$$

Stoten & Benchoubane [Sto89] proposent une loi d'adaptation sous la forme suivante:

$$\delta k_p(V', t) = \int_0^t \alpha V'(\tau) x^T(\tau) d\tau + \beta V'(t) x^T(t) \quad (\text{IV.78})$$

$$\delta k_u(V', t) = \int_0^t \alpha V'(\tau) U_m^T(\tau) d\tau + \beta V'(t) U_m^T(t)$$

Où  $V' = B^T P e$  et  $\delta k_p(0) = \delta k_u(0) = 0$ .

$\alpha, \beta$  : des scalaires tel que  $\alpha > 0; \beta \geq 0$ .

Ainsi nous remarquons une simplification remarquable dans le calcul de la loi d'adaptation.

## IV.10 Stratégie de commande par réseaux de neurones

Le principe de la stratégie de commande adoptée dans ce chapitre consiste en une commande supervisée où le réseau de neurones est utilisé pour remplacer le régulateur à GMV (figure IV.10). Le réseau de neurones apprendra à commander la structure à partir de données générées de la loi de commande à GMV. En utilisant leurs capacités d'apprentissage et de généralisation, les réseaux de neurones vont pouvoir commander la structure tout en surmontant les inconvénients de la GMV dus à sa dépendance du modèle de la structure.

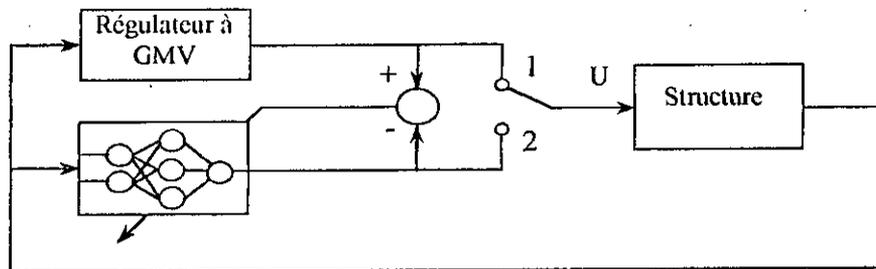


Figure IV.10 Schéma de la stratégie de commande par réseaux de neurones

La commande est implantée en deux phases :

- Phase d'apprentissage (1) : dans cette phase, un réseau de neurones apprend la fonction entrées/sorties du régulateur existant. Pour cela, un fichier d'exemples est généré en laissant ce régulateur commander le système. Puis un algorithme d'apprentissage supervisé est utilisé pour entraîner le réseau.
- Phase de commande (2) : une fois l'apprentissage effectué, le régulateur est enlevé et le réseau de neurones est mis à sa place pour commander le système.

L'un des premiers exemples de la commande supervisée est le pendule inversé de Widrow [Fuk92] en 1960. Widrow a référé à cette commande comme une méthode de construction d'un système expert neuronal par l'acquisition du savoir-faire d'un expert existant. Guez et Selinsky [Gue88] ont fourni une version plus améliorée et plus sophistiquée de la commande supervisée. Ils ont mis en évidence la capacité d'un réseau de neurone à apprendre à commander un système dynamique à partir d'exemples générés par des lois de commande linéaire, non linéaire et par un opérateur humain. Jorgenson [Mil92] a lui aussi fourni un excellent exemple de la commande supervisée dans le domaine de l'aéronautique.

### IV.10.1 Algorithme d'apprentissage

L'algorithme utilisé pour entraîner les réseaux de neurones est le fameux algorithme de la rétropropagation (Backpropagation) [Rum86][Kun92]. Cet algorithme est basé sur la minimisation d'un critère quadratique de l'erreur entre la sortie désirée et la sortie du réseau. Ceci est fait en changeant continuellement les poids du réseau dans la direction de la plus grande descente de l'erreur (procédure de descente du gradient). Le changement de chaque paramètre est proportionnel à sa relative contribution dans le critère d'erreur.

L'apprentissage par cette méthode s'effectue en 2 étapes :

- 1<sup>ère</sup> étape : un signal d'entrée appliqué au réseau se propage à travers les couches supérieures pour enfin générer le signal de sortie.
- 2<sup>ème</sup> étape : la sortie du réseau est comparée avec la sortie désirée et un signal d'erreur est généré. Ce dernier se rétropropage à travers les couches du réseau, provoquant ainsi une modification des poids dans le sens de la minimisation de l'erreur.

Pour un réseau statique à couches, la sortie  $O_j^l$  d'un neurone de la couche  $l$  est donnée par l'équation suivante :

$$\begin{cases} O_j^l = f(\text{net}_j^l) \\ \text{net}_j^l = \sum_{i=1}^{N_{l-1}} w_{ji}^l O_i^{l-1} \end{cases} \quad (\text{IV.79})$$

où  $\text{net}_j^l$  est la somme pondérée des sorties de la couche  $(l-1)$

$w_{ji}^l$  est le poids d'interconnexion entre le  $i^{\text{ème}}$  neurone de la couche  $(l-1)$  et le  $j^{\text{ème}}$  neurone de la couche  $l$

$N_{l-1}$  est le nombre de neurones de la couche  $(l-1)$

$f(.)$  est la fonction d'activation des neurones. La fonction tangente hyperbolique est utilisée. Elle est exprimée par :

$$f(x) = \frac{1 - \exp(-x)}{1 + \exp(-x)} \quad (\text{IV.80})$$

la fonction d'erreur à minimiser à chaque itération de l'algorithme de Backpropagation est définie par :

$$E_p = \frac{1}{2} \sum_{k=1}^{N_l} (y_{pk} - O_{pk}^L)^2 \quad (\text{IV.81})$$

où  $O_{pk}^L$  est la sortie du réseau

$y_{pk}$  est la sortie désirée

$N_l$  est le nombre de neurones de la couche de sortie

L'indice  $p$  indique l'élément  $p$  de l'ensemble des exemples d'apprentissage.

Le poids  $w_{ji}^l$  est ajuster de telle sorte à minimiser  $E_p$  :

$$w_{ji}^l(t+1) = w_{ji}^l(t) - \eta \frac{\partial E_p}{\partial w_{ji}^l(t)} \quad (\text{IV.82})$$

où  $\eta$  est le pas d'adaptation.

On peut résumer l'algorithme de Backpropagation dans les points suivants :

1. Initialiser les poids du réseau à de petites valeurs aléatoires, généralement comprises entre -1 et 1
2. Présenter un nouvel exemple  $p$  au réseau. L'exemple  $p$  est constitué des coordonnées  $(X_p, Y_p)$ , où  $X_p$  est le vecteur d'entrée au réseau, et  $Y_p$  et le vecteur des sorties désirées.
3. Calculer la sortie du réseau suivant l'équation (IV.79) ainsi que l'erreur de sortie suivant l'équation (IV.81)

4. Ajuster les poids du réseau selon la formule

$$w_{ji}^l(t+1) = w_{ji}^l(t) + \Delta w_{ji}^l(t) \quad (IV.83)$$

avec  $\Delta w_{ji}^l(t) = -\eta \frac{\partial E_p}{\partial w_{ji}^l(t)} = \eta \delta_j^l O_i^{l-1}$

$$\delta_j^l = \begin{cases} (y_{pj} - O_{pj}^L) f'(net_j^L) & \text{pour les neurones de la couche de sortie} \\ f'(net_j^l) \sum_{k=1}^{N_{l+1}} \delta_k^{l+1} w_{kj}^{l+1} & \text{pour les neurones des couches cachées} \end{cases}$$

5. Répéter les étapes 2 à 4 jusqu'à ce que l'erreur totale sur l'ensemble des exemples soit inférieure à un certain seuil de performance.

*Remarque*

*Comme la sortie du réseau, du fait de la fonction tangente hyperbolique, est comprise entre -1 et 1, les exemples d'apprentissage doivent être normaliser à ces valeurs. Pour éviter la saturation du réseau lors de la phase d'apprentissage ou lors de la phase de commande, la fonction d'activation des neurones de la couche de sortie est souvent choisie comme étant une fonction linéaire.*

### IV.10.2 Commande d'une structure en génie civil à un seul degré de liberté

Pour une structure à un seul degré de liberté, la figure IV.11 donne le schéma de commande une fois la phase d'apprentissage achevée. Le réseau de neurone remplace l'algorithme à GMV après avoir acquis les informations nécessaires sur la commande de la structure.

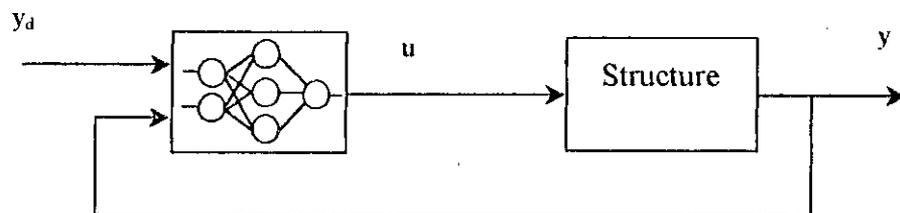


Figure IV.11 Schéma de la commande pour une structure à un seul degré de liberté

Pour pouvoir implementer la stratégie de commande décrite ci-dessus, on utilise un réseau statique multicouche ayant 8 entrées, 10 neurones dans la couche cachée et un neurone dans la couche de sortie. Les réseaux statiques à couches se prêtent bien à l'approximation des fonctions. Un réseau statique à une seule couche cachée ayant des fonctions d'activation de type sigmoïde peut approximer arbitrairement les fonctions continues [Hun92]. L'architecture du réseau est illustrée dans la Figure IV.12.

Les entrées du réseau pour générer la commande à l'instant  $t$  sont : l'accélération sismique à 2 instants précédents, le déplacement relative à 3 instants précédents et la commande à 3 instants précédents. Il n'existe pas de méthodes systématiques pour le choix de ces variables d'entrées ainsi que le nombre d'instants précédents pour chaque variable [MIL92][BAN98]. Néanmoins, il faut fournir au réseau un nombre suffisant d'informations pour qu'il puisse approximer la loi de commande à GMV.

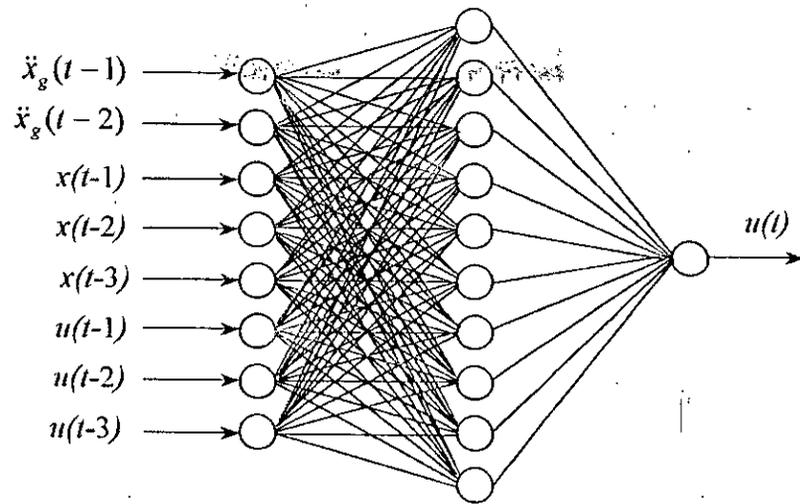


Figure IV.12 Architecture du réseau utilisé

### IV.10.3 Commande d'une structure en génie civil à plusieurs degrés de liberté

Il existe deux approches différentes pour la commande des structures à plusieurs degrés de liberté par la stratégie de commande précédemment décrite. La 1<sup>ère</sup> approche envisagée consiste à utiliser un seul réseau de neurone pour approximer le régulateur existant. Néanmoins, cette approche présente plusieurs inconvénients en particulier la dimension que prend le réseau utilisé. En effet, ce dernier nécessitera des informations sur tous les étages de la structure et aura un nombre de sorties égal au nombre d'étage à commander et la dimension de ce réseau augmentera proportionnellement avec le nombre de degrés de liberté. On voit bien que cette dimension pose des problèmes lors de l'apprentissage ainsi que la capacité mémoire nécessaire pour implanter ce réseau et son apprentissage dans un ordinateur. Le nombre élevé de paramètres à ajuster lors de la phase d'apprentissage nécessitera un temps énorme pour que le réseau atteigne les performances voulues.

Pour ces raisons, nous adopterons la 2<sup>ème</sup> approche qui consiste à utiliser plusieurs réseaux de neurones où chaque réseau sera dédié à la commande d'un seul étage de la structure. On fournit au réseau des informations locales et ce dernier génère l'effort de commande nécessaire pour maintenir l'équilibre d'un étage donné. Cette approche présente plusieurs avantages par rapport à la 1<sup>ère</sup> approche, parmi lesquels : (1) la décentralisation donc la possibilité d'implanter chaque réseau dans un ordinateur indépendant, (2) dimension réduite des réseaux utilisés, en effet chaque réseau aura seulement une sortie et ne nécessitera que des informations locales sur l'étage qu'il est entrain de commander, (3) diminution considérable du temps d'apprentissage et de la capacité mémoire nécessaire pour implanter les réseaux.

Le schéma de cette commande est illustré dans la figure IV.13 une fois la phase d'apprentissage achevée.

La figure IV.14 donne une représentation du réseau dédié à la commande de l'étage  $i$ . Il possède 8 entrées, 10 neurones dans la couche cachée et une sortie de commande. Comme indiqué précédemment, le choix des entrées pour le réseau n'est pas basé sur des méthodes systématiques.

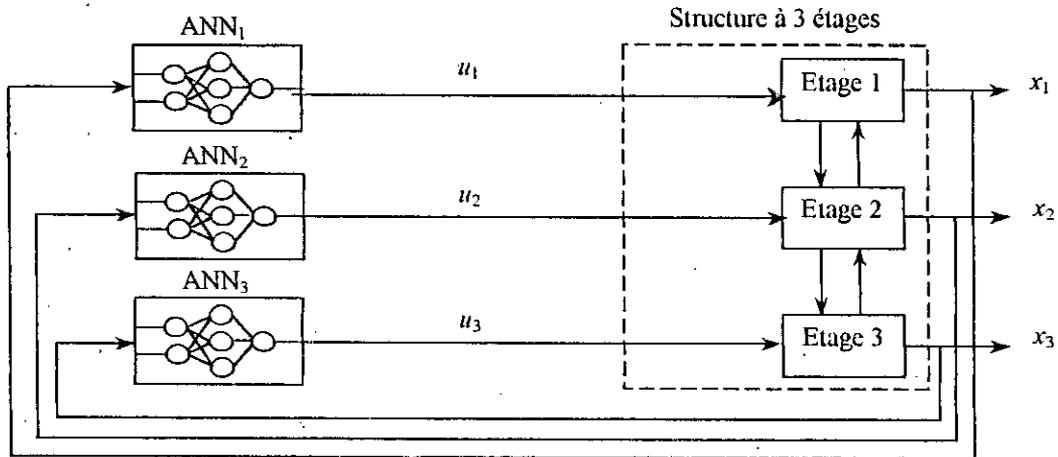


Figure IV.13 schéma de commande pour une structure à trois degrés de liberté

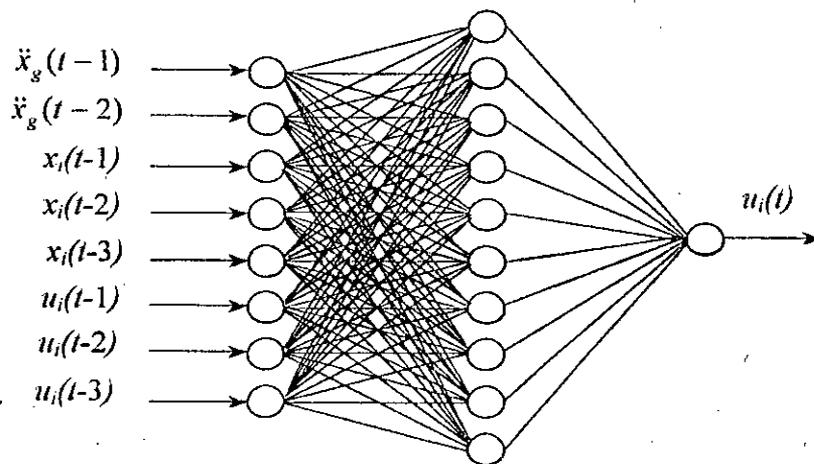


Figure IV.14. Architecture du réseau utilisé pour la commande de l'étage  $i$

## IV.11 Commande à gain programmé

Il est quelques fois possible de trouver des variables auxiliaires qui reflètent le changement dans la dynamique du système en fonction du changement des conditions opératoires. Il est alors possible de commander le système et réduire l'effet de ces changements en adaptant les paramètres du régulateur comme une fonction des variables auxiliaires. C'est là l'idée de base de la commande à gain programmé dont le schéma est illustré sur la figure IV.15 [Ast89][Cha89].

Ainsi cette dernière peut être vue comme un système de commande en boucle fermée, où les paramètres du régulateur sont ajustés par une action anticipatrice.

Pour implementer une commande à gain programmé, il faudrait d'abord trouver les variables auxiliaires ou les variables de programmation mentionnées ci-dessus. Puis un régulateur linéaire est synthétisé et dont les paramètres sont calculés à un certain nombre de conditions opératoires. Ce régulateur est ainsi ajusté ou calibré, i.e., ses paramètres sont modifiés de tel sorte à accommoder le changement des conditions opératoires du système.

Du fait que les paramètres du régulateur sont compensés ou modifiés en boucle ouverte, la commande à gain programmé ne doit pas être vue comme un régulateur adaptatif mais plutôt comme un type particulier de régulateur non linéaire. L'un des inconvénients de cette technique de commande est qu'elle peut nécessiter beaucoup de temps pour son implementation car les paramètres du régulateur doivent être déterminés pour plusieurs conditions opératoires. Par contre avec ce schéma de commande le régulateur peut réagir rapidement face aux changements des conditions opératoires du système.

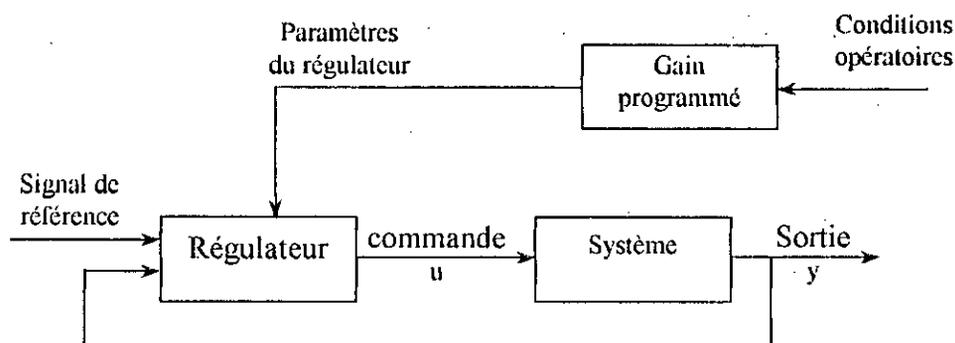


Figure IV.15 Schéma bloc de la commande à gain programmé

Afin d'atteindre cet objectif, une linéarisation des équations dynamiques de la structure au point nominal est nécessaire. On obtient alors un modèle à perturbation (modèle variationnel) [Gue95]. La commande est calculée à partir de ce modèle de fonctionnement choisi [Gue96].

## IV.12 Commande à gain programmé des structures en génie civil

Pour une structure non linéaire, les conditions opératoires qui engendrent le changement de la dynamique de la structure sont les non linéarités dont on connaît le modèle. Ainsi le modèle non linéaire de la structure est linéarisé autour d'un certain nombre de points de fonctionnement qui représentent la plage d'évolution de la réponse de la structure. Pour chaque modèle linéaire obtenu, un régulateur est synthétisé. Puis en fonction de l'état actuel du système, le régulateur adéquat est activé.

Le schéma de cette commande est représenté sur la figure IV.16

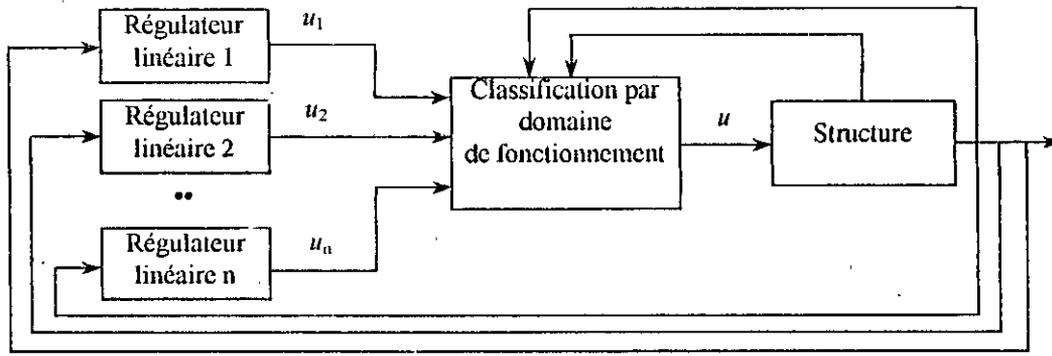


Figure IV.16 Schéma de la commande à gain programmé pour une structure non linéaire

La figure I.8 est une représentation schématique d'une structure non linéaire à un seul degré de liberté, dont l'équation de mouvement est définie par (I.24) [Fol96].

La simulation en boucle ouverte du modèle de la structure permet de tracer la caractéristique d'hystérésis de cette structure. Pour une excitation sismique de type bruit blanc de densité spectrale  $S_0=10$ , et les paramètres  $m=2000$  kg,  $\omega_0 = \sqrt{k/m} = 2$  rad/s,  $\xi = c/2m\omega_0 = 0.05$ ,  $\alpha = 0.1$ ,  $\beta = 1.5$ ,  $\gamma = -0.5$ ,  $h(z)=1$ ,  $n=1$ ; cette caractéristique est illustré sur la figure IV.17.

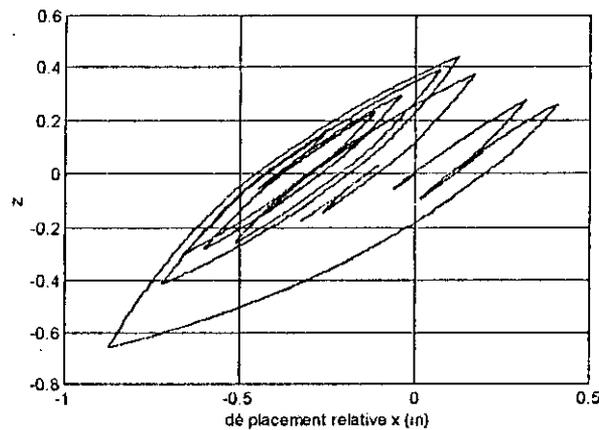


Figure IV.17 Caractéristique d'hystérésis de la structure excitée par un bruit blanc

On remarque à travers cette figure, la très forte non linéarité de la structure. En conséquence un tel comportement nécessite une linéarisation autour d'un certain nombre de points de fonctionnement qu'on choisira ultérieurement. On procède alors dans ce qui suit à développer le modèle à perturbation continu.

Pour un vecteur d'état  $X^T = [x_1 \ x_2 \ x_3] = [x \ \dot{x} \ z]$ , les équations (I.24) et (I.25) peuvent être mises sous la forme d'état non linéaire suivante :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -\frac{c}{m}x_2 - \alpha\frac{k}{m}x_1 - (1-\alpha)\frac{k}{m}x_3 + \frac{1}{m}u - \ddot{x}_g \\ \dot{x}_3 = x_2 - \beta|x_2|x_3 - \gamma x_2|x_3| \end{cases} \quad (\text{IV.84})$$

Afin de pouvoir implementer la commande à gain programmé, il est d'abord nécessaire d'effectuer une linéarisation du modèle de la structure décrit par la forme d'état (IV.84) [Slo91] Ainsi, les techniques de linéarisation permettent l'extension des méthodes de commande linéaires aux systèmes non linéaires.

La linéarisation de l'équation d'état (IV.84) revient en faite à linéariser la 3<sup>ème</sup> équation de ce système qui contient la non linéarité

$$\dot{x}_3 = x_2 - \beta|x_2|x_3 - \gamma x_2|x_3| \quad (IV.85)$$

Cette équation est de la forme :

$$\dot{x}_3 = f(x_2, x_3) \quad (IV.86)$$

On obtient la forme linéarisée suivante :

$$\delta\dot{x}_3 = a\delta x_2 + b\delta x_3 \quad (IV.87)$$

avec  $\delta x_2 = x_2 - x_{20}$

$\delta x_3 = x_3 - x_{30}$

$$a = \left. \frac{\partial \dot{x}_3}{\partial x_2} \right|_{x_{20}, x_{30}} = \left\{ 1 - \beta x_3 \operatorname{sgn}(x_2) - \gamma |x_3| \right\}_{x_{20}, x_{30}}, \quad b = \left. \frac{\partial \dot{x}_3}{\partial x_3} \right|_{x_{20}, x_{30}} = \left\{ -\beta |x_2| - \gamma x_2 \operatorname{sgn}(x_3) \right\}_{x_{20}, x_{30}}$$

Notons que la 1<sup>ère</sup> composante du vecteur d'état  $x_1$  et la variable de commande  $u$  ne figurent pas explicitement dans le modèle de la non linéarité (i.e. l'état de linéarisation ne dépend pas de  $x_1$  et de  $u$ ), on peut alors introduire les équations variationnelles suivantes

$$\delta x_1 = x_1 - x_{10} \quad \text{avec } x_{10} = 0 \text{ et } u_0 = 0$$

$$\delta u = u - u_0$$

Ceci nous permet de définir le modèle d'état linéarisé (perturbation model [ATH71]) de la structure :

$$\begin{cases} \delta\dot{x}_1 = \delta x_2 \\ \delta\dot{x}_2 = -\frac{c}{m}\delta x_2 - \alpha \frac{k}{m}\delta x_1 - (1-\alpha)\frac{k}{m}\delta x_3 + \frac{1}{m}\delta u - \ddot{x}_g \\ \delta\dot{x}_3 = a\delta x_2 + b\delta x_3 \end{cases} \quad (IV.88)$$

La synthèse d'une commande à GMV nécessite l'élaboration du modèle ARMAX du système. Pour le modèle linearisé de la structure, l'élimination des variables  $\delta x_2$  et  $\delta x_3$  conduit au modèle ARMAX équivalent en  $s$  suivant :

$$\delta X(s) = \frac{B(s)}{A(s)} \delta U(s) + \frac{C(s)}{A(s)} \ddot{X}_g(s) \quad (IV.89)$$

avec  $\delta X(s)$ ,  $\delta U(s)$  et  $\ddot{X}_g(s)$  transformées de Laplace des variables  $\delta x_i(t)$ ,  $\delta u(t)$  et  $\ddot{x}_g(t)$  respectivement.

$$\text{et } \begin{cases} A(s) = s^3 + (2\xi\omega_0 - b)s^2 + (\alpha\omega_0^2 - 2b\xi\omega_0^2 + (1-\alpha)\omega_0^2 a)s - b\alpha\omega_0^2 \\ B(s) = \frac{1}{m}(s - b) \\ C(s) = (s - b) \end{cases} \quad (IV.90)$$

avec  $\omega_0^2 = k/m$  et  $\xi = c/2m\omega_0$ .

Par discrétisation de l'équation (IV.89) on obtient le modèle ARMAX discret qui nous permet de synthétiser la commande à GMV.

La simulation en boucle ouverte de la réponse de la structure soumise à 300% de l'excitation de Kanai-Tajimi (avec  $S_0=1$ ) permet de tracer l'évolution de l'état de la structure dans le plan des variables  $(x_2, x_3) = (\dot{x}, z)$ . Comme indiqué sur la figure IV.18 et le tableau

IV.1, quatre points de linéarisation sont choisis correspondant à quatre domaines de fonctionnement.

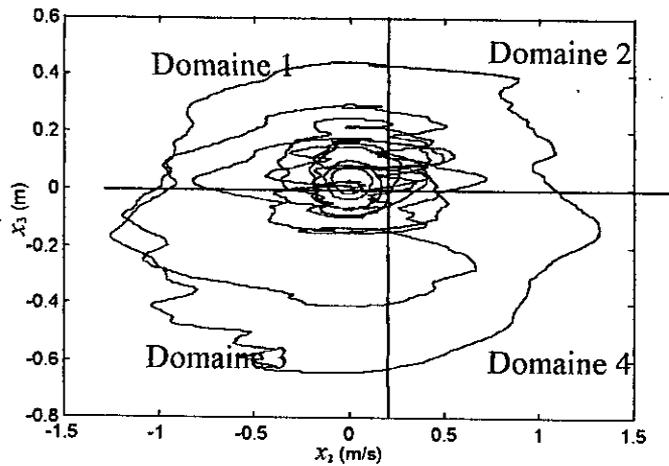


Figure IV.18. Evolution de l'état de la structure

Tableau IV.1 Points de linéarisation et domaines de fonctionnement

Domaine de fonctionnement	limites du domaine	Point de linéarisation	$a$	$b$
1	$x_2 < 0, x_3 > 0$	$x_2 = -0.5, x_3 = 0.1$	1.2	-1
2	$x_2 > 0, x_3 > 0$	$x_2 = 0.5, x_3 = 0.1$	0.9	-0.5
3	$x_2 < 0, x_3 < 0$	$x_2 = -0.5, x_3 = -0.1$	0.9	-0.5
4	$x_2 > 0, x_3 < 0$	$x_2 = 0.5, x_3 = -0.1$	1.2	-1

### IV.13 Commande à variance minimale généralisée avec classification par domaine de fonctionnement

En utilisant le modèle linéarisé obtenu précédemment, un régulateur à variance minimale généralisée est synthétisé. Ceci est effectué pour un certain nombre de points de linéarisation  $(x_{20}, x_{30})$ . Chacun de ces points représente une plage de fonctionnement dans le plan des variables  $(x_2, x_3)$ . Lorsque l'état actuel  $(x_2, x_3)$  de la structure est à l'intérieur de cette plage, le régulateur à GMV correspondant est activé.

Cette commande peut être considérée comme une classification des régulateurs préalablement synthétisés, en fonction de l'état actuel du système représenté par les deux variables de programmation (*scheduling variables*)  $x_2, x_3$ . Ceci est schématiquement représenté à la figure IV.19.

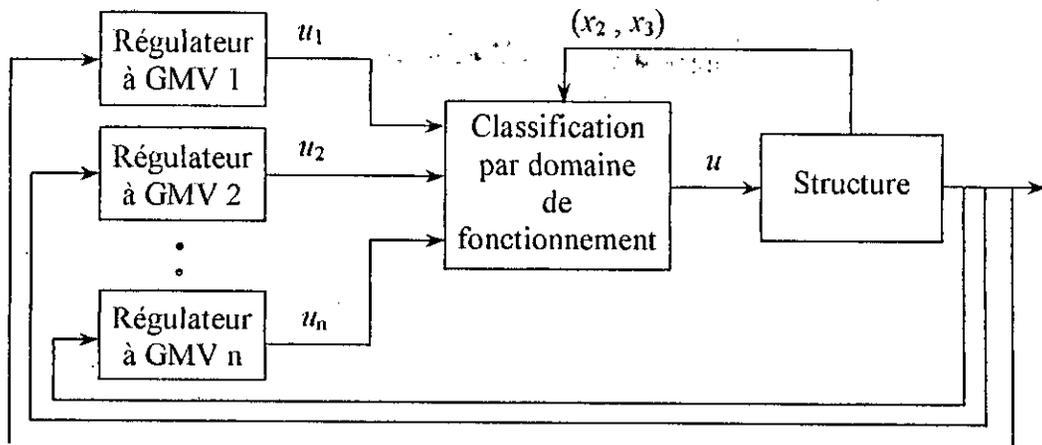


Figure IV.19. Commande à variance minimale généralisée avec classification par domaines de fonctionnement

Pour le modèle ARMAX défini par (IV.89), le critère de commande à minimiser est défini par [Gue95] :

$$J = E \left[ \left( P(q^{-1}) \delta x(t+d+1) \right)^2 + \left( Q'(q^{-1}) \delta u(t) \right)^2 \right] \quad (\text{IV.91})$$

$P, Q'$  sont des polynômes de pondération.

On constate ici que la détermination de la loi de commande  $\delta u(t)$  qui minimise le critère (IV.91) conduit directement à la détermination de la commande originale  $u(t)$  car  $\delta u(t) = u(t) - u_0 = u(t)$ . De même, la minimisation de  $\delta x(t)$  autour de 0, conduit à la minimisation du déplacement relative de la structure  $x$ , car  $\delta x(t) = x(t) - x_{i0} = x(t)$ .

La loi de commande est [Gue95]

$$\delta u(t) = \frac{-R \delta x(t)}{P_D(S + QC)} \quad (\text{IV.92})$$

où les différents polynômes sont définis dans le chapitre III.

#### IV.14 Commande à variance minimale généralisée avec classification par réseaux de neurones

Dans le but d'améliorer les performances de la stratégie de commande développée précédemment du point de vue temps de calcul, on se propose dans cette approche d'utiliser un réseau de neurones pour faire la classification ou la programmation des régulateurs. En fonction de l'état actuel du système, c'est le réseau de neurones qui active le régulateur adéquat pour commander le système.

La loi de commande est donnée par :

$$u(t) = \sum_{i=1}^n O_i(t) \times u_i(t) \quad (\text{IV.93})$$

où  $O_i$  sont les sorties du réseau

$u_i$  sorties des régulateurs à GMV.

Ceci est schématiquement représenté sur la figure IV.20

Un réseau de neurones à couche est utilisé pour accomplir cette tâche. Les entrées du réseau sont les variables de programmation (*scheduling variables*)  $x_2, x_3$ . Le nombre de sorties du réseau est égal au nombre de régulateurs linéaires à GMV synthétisés et qui correspond aussi au nombre de points de fonctionnement choisis pour faire la linéarisation.

Le réseau est entraîné par l'algorithme de Backpropagation de tel sorte que l'une de ses sorties soit égale à 1 et les autres sorties soient égales à 0. La sortie activée correspond au régulateur qui doit être mis en fonctionnement.

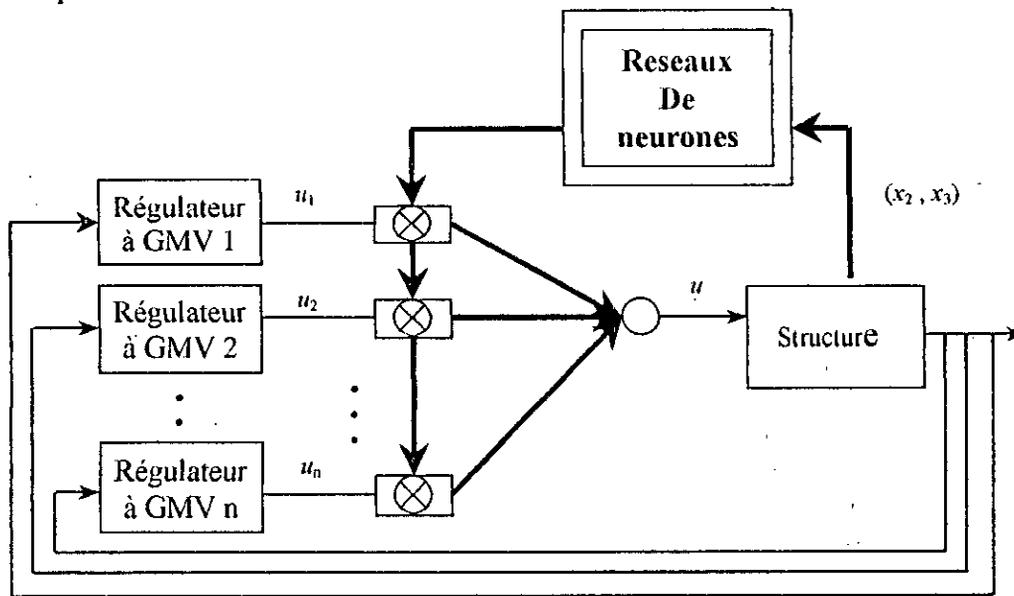


Figure IV.20. Commande à variance minimale généralisée avec classification par réseaux de neurones

### III.15 Résultats de simulation

Nous allons procéder dans cette partie à la simulation des algorithmes développés dans le cas des structures articulées en robotique et des structures en génie civil.

#### III.15.1 Commande des structures articulées

Le modèle du robot utilisé dans cette partie est le même que celui développé dans le chapitre I. Les différentes stratégies de commande non linéaire ont été testées dont nous présentons les résultats de simulation.

Dans le cas de l'algorithme à compensation de gravité, la trajectoire utilisée est polynomiale. L'algorithme de commande est mixte dont les paramètres respectifs sont : PRPE,  $Q_1=0.007$  et GMV,  $Q=0.05$  et PP,  $P_3=1-0.5q^{-1}+0.06q^{-2}$ . Les modèles de référence sont  $Bm_2/Am_2=0.095/(1-0.9q^{-1})$ . La figure IV.21 illustre les résultats de simulation dans le cas d'une variation paramétriques de 500% à  $t=5s$ . En appliquant une consigne de fenêtre de VIVIANI, On remarque une erreur statique dans la figure IV.22.

Pour le découplage non linéaire partiel, l'algorithme de commande pour toute les articulations est PRPE avec :  $Q_1=0.007$ ,  $Q_2=0.05$ ,  $Q_3=0.005/(1-q^{-1})$  et  $P_i=1$ ,  $Bm_i/Am_i=1$ . Dans le cas de la variation paramétrique précédente (à  $t=10s$ ), on constate dans la IV.23, des oscillations de la deuxième articulation. Pour la fenêtre de VIVIANI, l'erreur de poursuite est faible comme le montre la figure IV.24.

En ce qui concerne le découplage utilisant la géométrie différentielle, l'algorithme de commande est le PRPE pour toutes les articulations avec  $Bm_i/Am_i=0.095/(1-0.9q^{-1})$ ,  $Q_i=0.005/(1-q^{-1})$ ,  $P_i=1-0.1q^{-1}$ . En gardant les même variations paramétriques, on remarque

dans la figure IV.25 un suivi de trajectoire et une diminution des oscillations. Pour la fenêtre de VIVIANI, les discontinuités de la référence imposée engendrent des oscillations remarquable (figure IV.26).

Dans le cas de l'architecture de la commande APC utilisant un algorithme mixte : articulation 1 GMV  $Q_1=0.007$ , articulation 2 PRPE  $Q_2=0.05$ , articulation 3 PRPE  $Q_3=0$   $P_3=1$ . Les résultats sont montrés dans la figure IV.27 avec variation paramétrique. Pour une consigne de fenêtre de VIVIANI, cette commande arrive à maintenir le suivi de trajectoire. La première articulation présente une erreur remarquable après passage par le point de discontinuité comme le montre la figure IV.28.

L'application de la commande linéarisante à paramètres connus au robot est illustrée dans la figure IV.29 dont les paramètres sont :  $K_{pi}=100$ ,  $K_{vi}=20$  (variation paramétrique).

La simulation de la commande linéarisante adaptative, où la paramétrisation est:

$$\tau = W(q, \dot{q}, \ddot{q})\hat{\theta}$$

$$\text{avec } W(q, \dot{q}, \ddot{q}) = \begin{bmatrix} w_{11} & w_{12} & w_{13} & 0 & 0 \\ 0 & w_{22} & 0 & w_{24} & 0 \\ 0 & w_{32} & 0 & 0 & w_{35} \end{bmatrix}; \text{ et } \hat{\theta}^T = [m_1 \quad m_3 \quad f_1 \quad f_2 \quad f_3]$$

$$w_{11} = w_{12} = \ddot{q}_1 + g; w_{13} = \dot{q}_1; w_{22} = (q_3^2 + \frac{l_2^2}{3} - q_3 l_2)\ddot{q}_2 + 2(q_3 - \frac{l_2}{2})\dot{q}_2 \dot{q}_3;$$

$$w_{24} = \dot{q}_2; w_{35} = \dot{q}_3; w_{32} = \ddot{q}_3 - (q_3 - \frac{l_2}{2})\dot{q}_2^2,$$

nécessite le choix des pondérations suivantes:

$$\psi_i = 5; i = 1, 5 \text{ et } \Gamma = \text{diag}(100, 100, 50, 50, 50).$$

et les paramètres initiaux  $\tilde{\theta}_0 = [10 \quad 10 \quad 10 \quad 10 \quad 2]$ .

Dans de telles conditions, on remarque une amélioration de l'effort de commande illustré par la figure IV.30. Pour une consigne de fenêtre de VIVIANI avec  $\Gamma = \text{diag}(1000 \quad 1000 \quad 500 \quad 500 \quad 500)$ , les performances de cette commande sont remarquables comme le montre la figure IV.31.

Pour la commande adaptative à modèle de référence, on choisit:

$$A_m = \begin{bmatrix} 0 & I \\ -a_1 & -a_2 \end{bmatrix}; B_m = \text{diag} \begin{bmatrix} 0 \\ a_1 \end{bmatrix}. \text{ Où } a_1 = \lambda_1 \lambda_2 \text{ et } a_2 = -(\lambda_1 + \lambda_2).$$

Avec  $\lambda_1 = \lambda_2 = -4$  : pôles désirés.

La résolution de l'équation de Lyapounov permet la détermination du compensateur.

$$Q = \text{diag} \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}.$$

Les matrices de la loi d'adaptation sont :

$$G_A = G_A' = I_{6 \times 6}; G_B = G_B' = I_{3 \times 3}; F_A = F_B' = \alpha I_{3 \times 3}; F_A' = F_A' = \beta I_{3 \times 3}; \\ \alpha = 100 \text{ et } \beta = 1, \text{ avec les conditions initiales nulles } (\delta k_p(0) = \delta k_v(0) = 0).$$

Les résultats de simulation sont consignés dans la figure IV.32 pour une variation paramétrique. Pour tester la capacité de poursuite, l'imposition d'une consigne de fenêtre de VIVIANI est appliquée, dont les résultats sont dans la figure IV.33. On remarque de légères oscillations de l'articulation 3.

Pour le test de l'algorithme du MCS ( $V = Pe$ ), les pondérations sont:

$$G_A = G_{A'} = I_{6 \times 6}; G_B = G_{B'} = I_{3 \times 3};$$

$$F_A = F_B = \text{diag}[0 \ \alpha_i] = \text{diag}[0 \ 100]; \dim(F_A) = 3 \times 6;$$

$$F_{A'} = F_{B'} = \text{diag}[0 \ \beta_i] = \text{diag}[0 \ 1]; \dim(F_{A'}) = 3 \times 6;$$

$$Q = \text{diag} \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix} \text{ et } \lambda_1 = \lambda_2 = -4.$$

Les résultats sont représentés dans la figure IV.34. Dans le cas de la consigne de fenêtre de VIVIANI, et à fin de mettre en évidence l'importance des coefficients de pondération, les figures IV.35 et IV.36 montrent les résultats de simulation pour respectivement  $\beta_i=60$  et  $\beta_i=1$ .

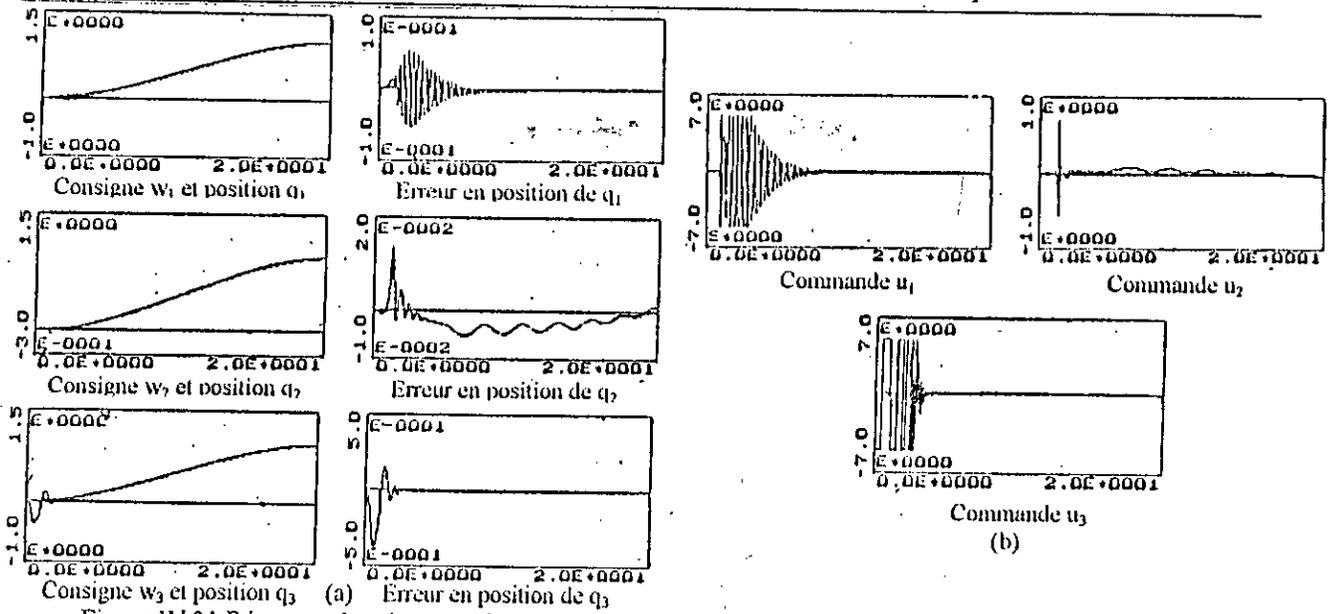


Figure IV.21 Réponses du robot pour la commande par compensation de gravité (Référence Polynomiale)

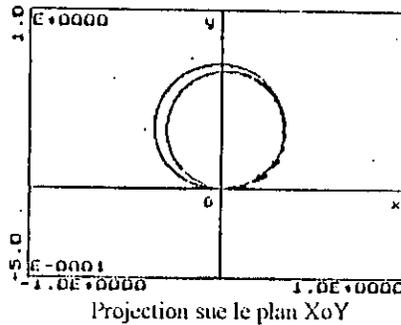
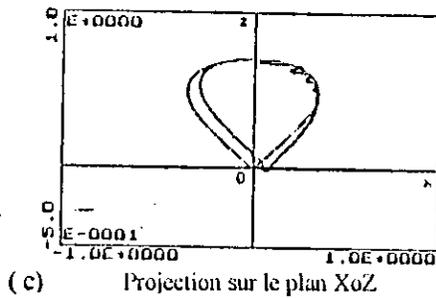
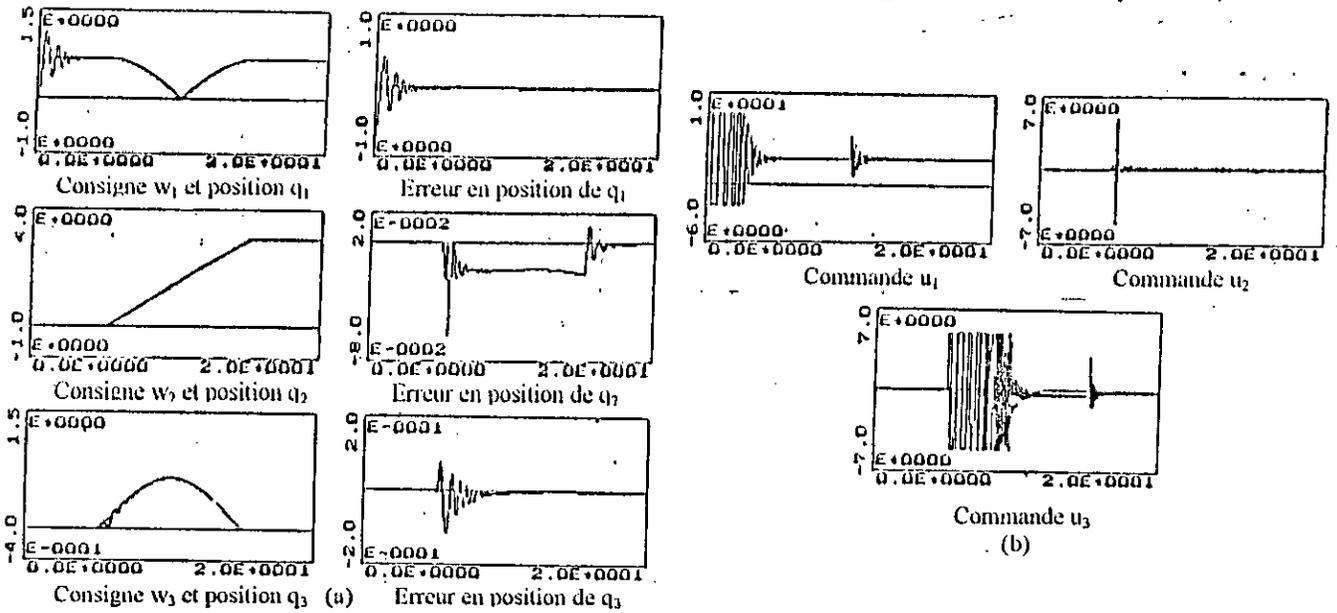


Figure IV.22 Réponses du robot pour la commande par compensation de gravité (Référence VIVIANI)

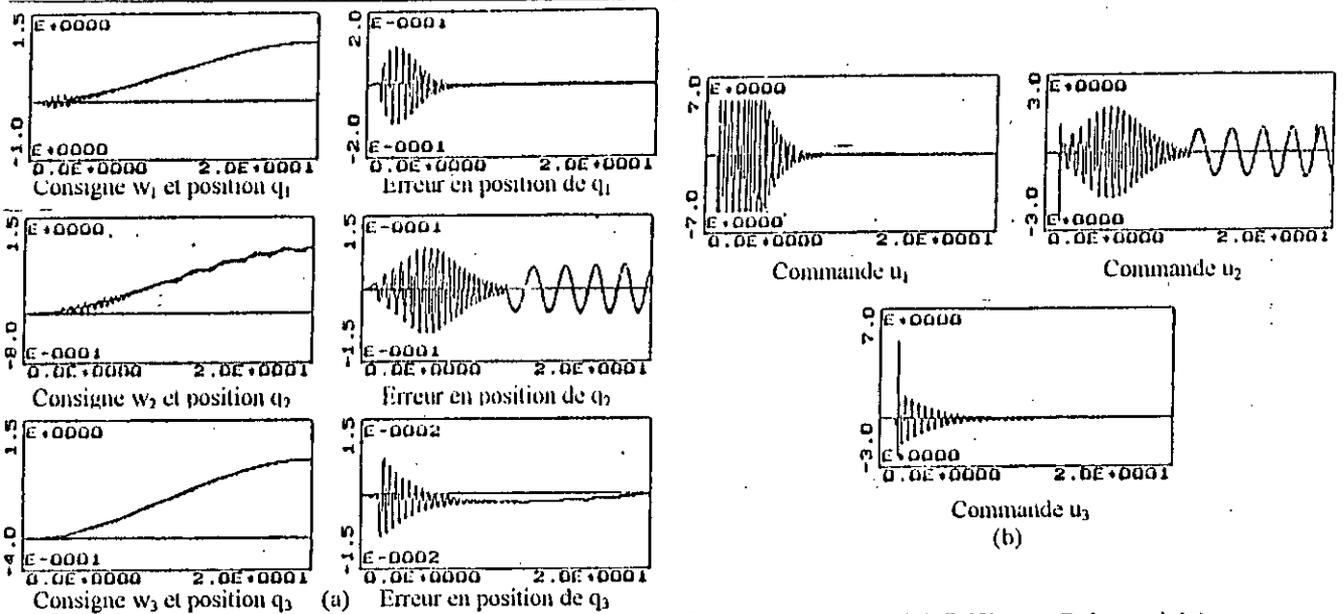


Figure IV.23 Réponses du robot pour la commande par Découplage NL partiel (Référence Polynomiale)

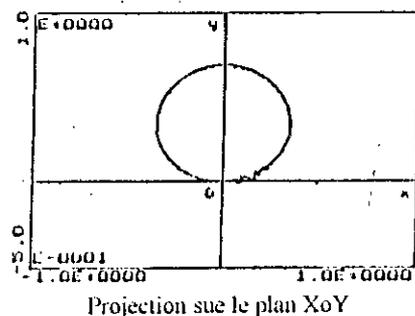
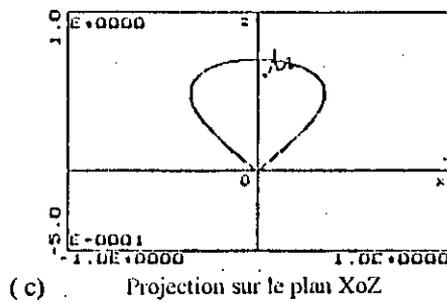
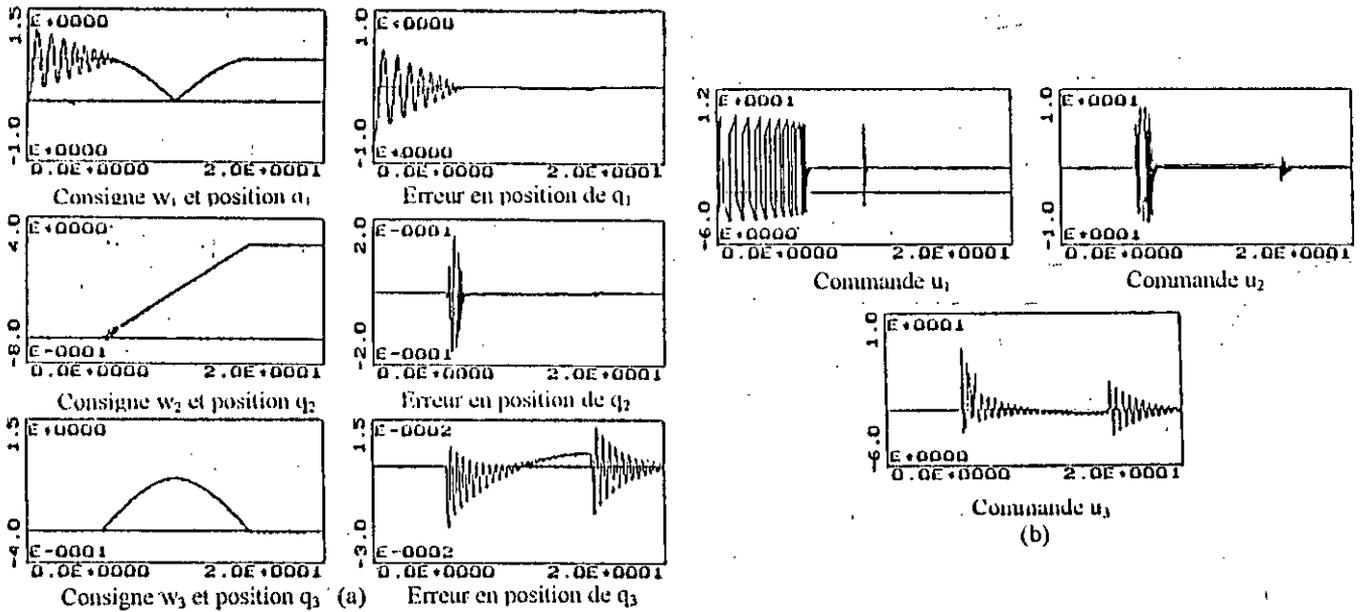


Figure IV.24 Réponses du robot pour la commande par Découplage NL partiel (Référence VIVIANI)

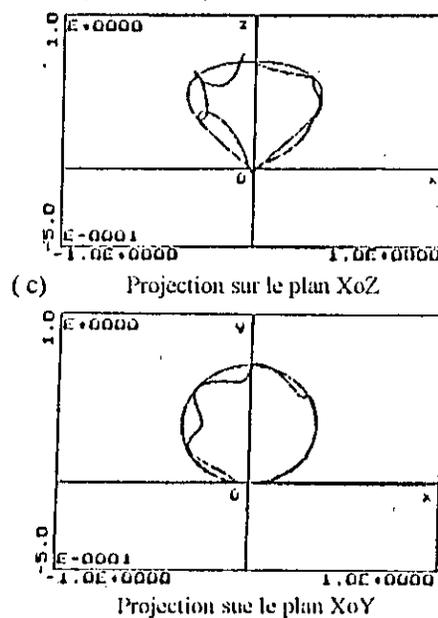
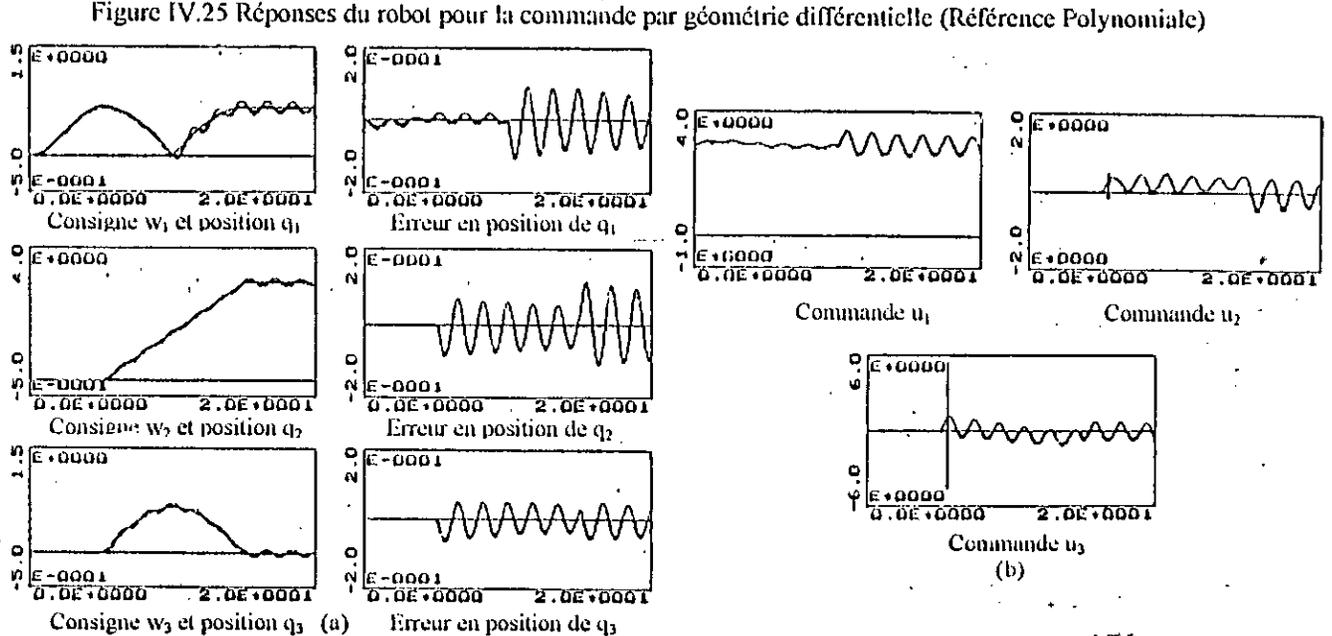
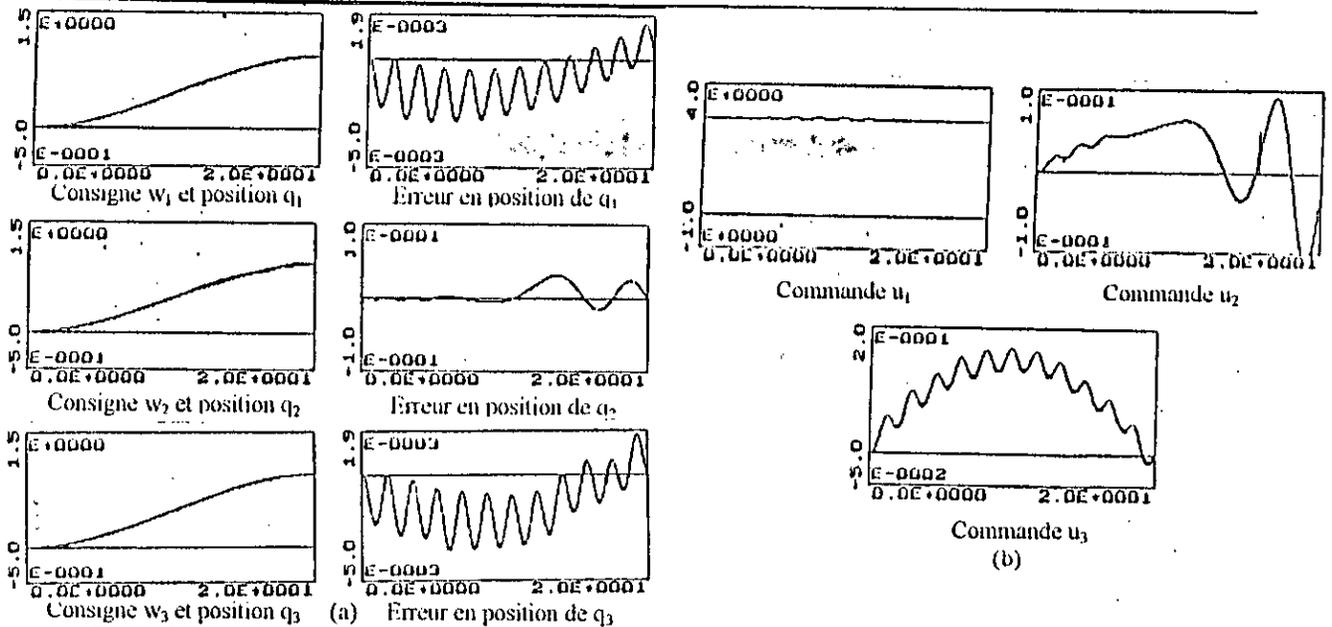


Figure IV.26 Réponses du robot pour la commande par géométrie différentielle (Référence VIVIANI)

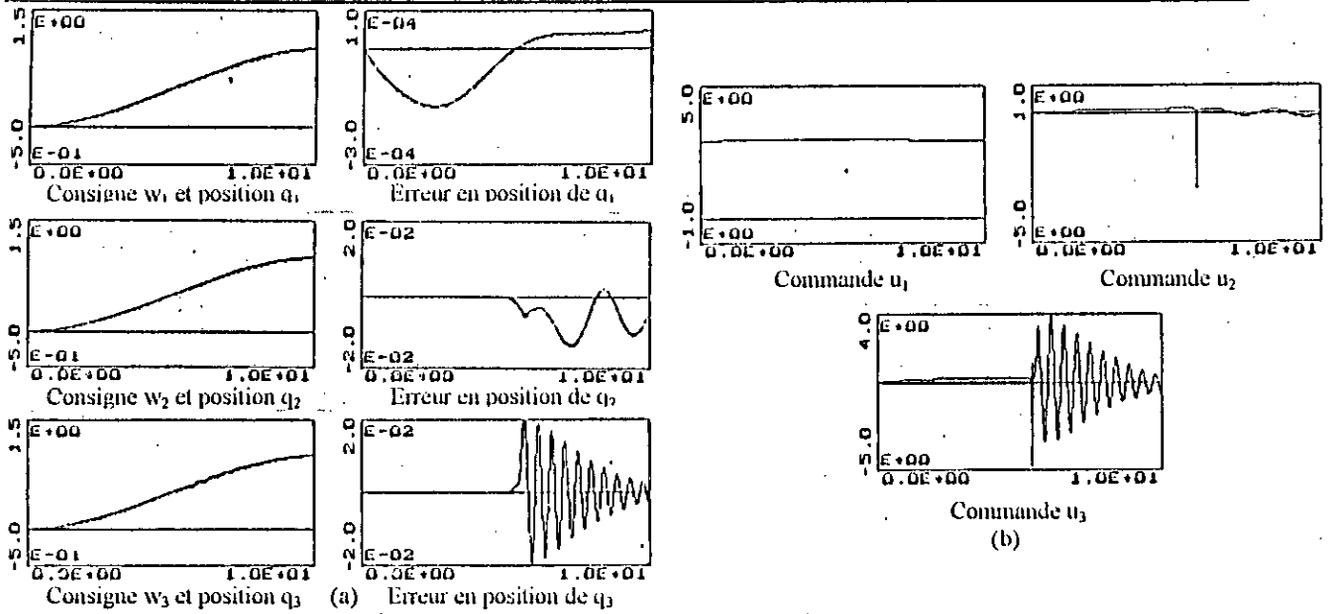


Figure IV.27 Réponses du robot pour la commande APC (Référence Polynomiale)

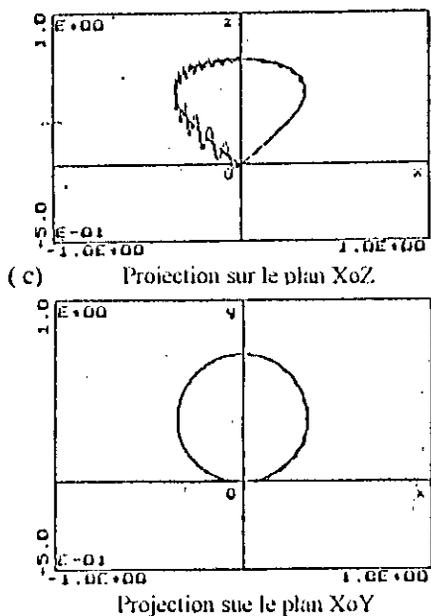
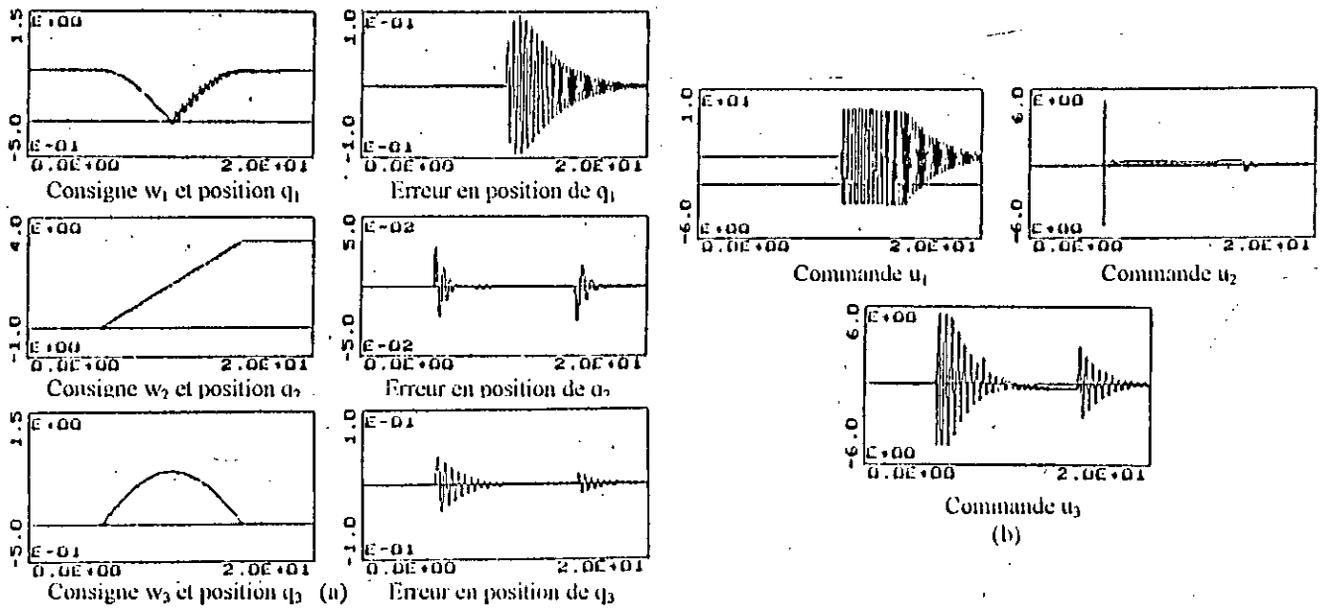


Figure IV.28 Réponses du robot pour la commande APC (Référence VIVIANI)

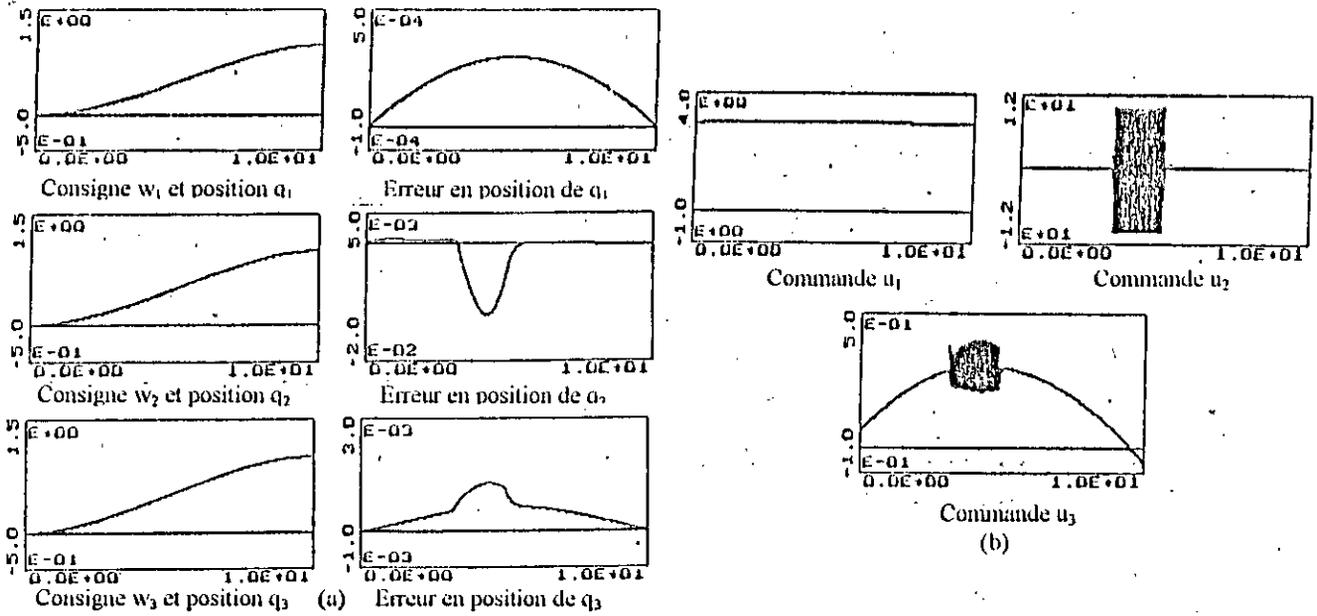
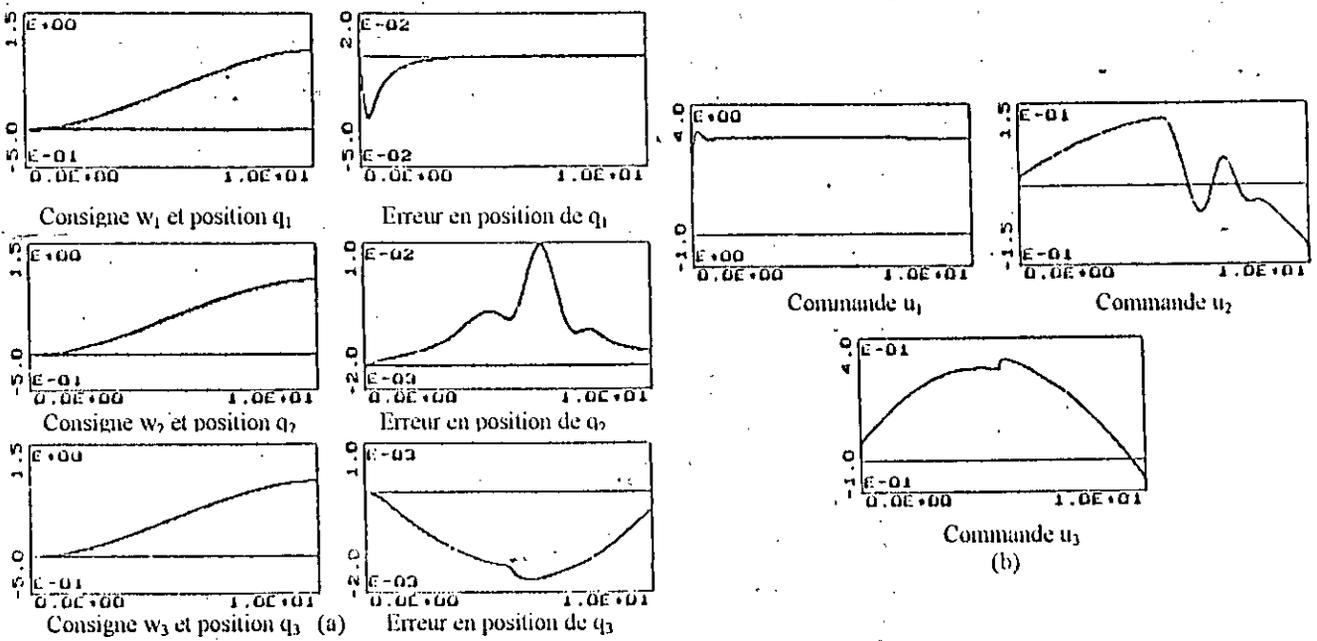


Figure IV.29 Réponses du robot pour la commande linéarisante (Référence Polynomiale)



Projection sur le plan XoY

Figure IV.30 Réponses du robot pour la commande Linéarisante Adaptative (Référence Polynomiale)

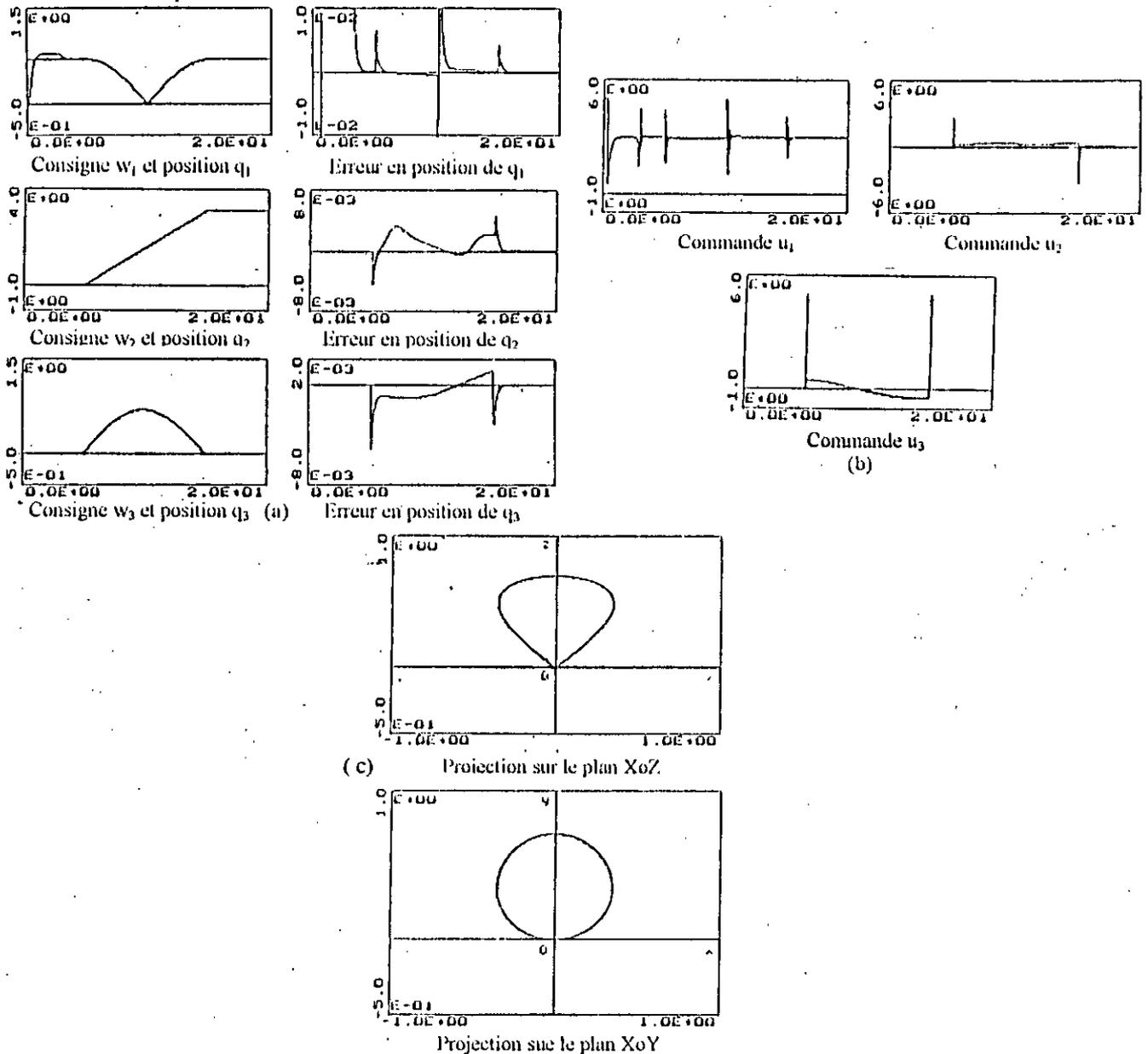


Figure IV.31 Réponses du robot pour la commande Linéarisante Adaptative (Référence VIVIANI)

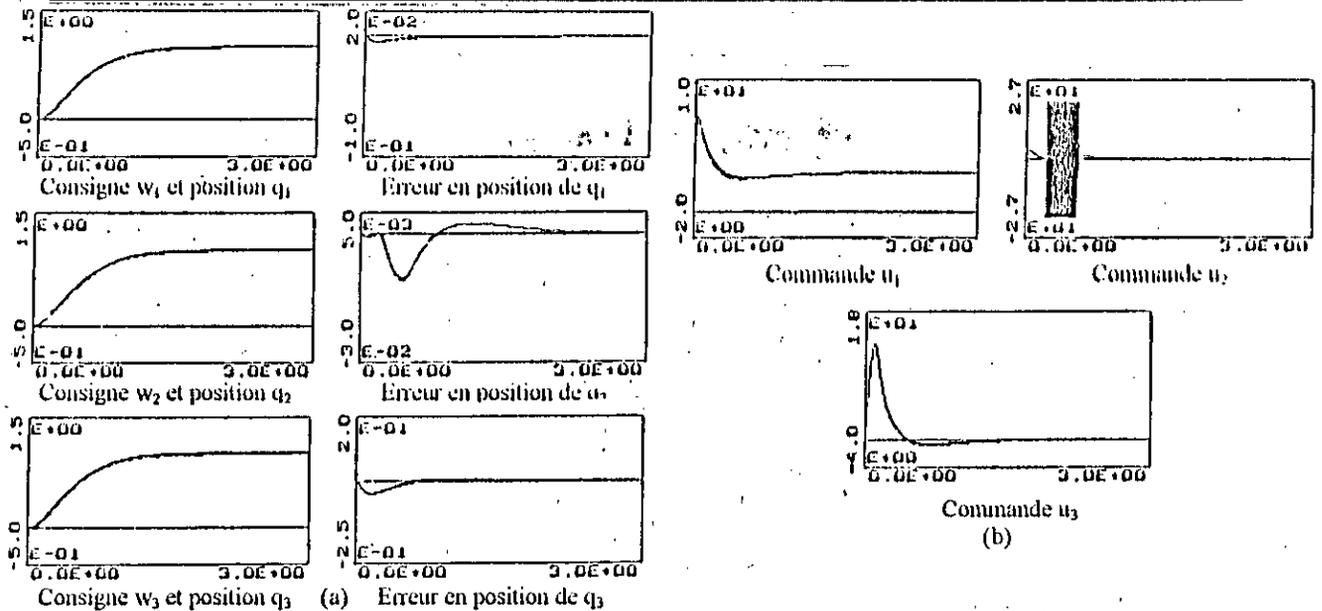


Figure IV.32 Réponses du robot pour la commande AMFC (Référence Polynomiale)

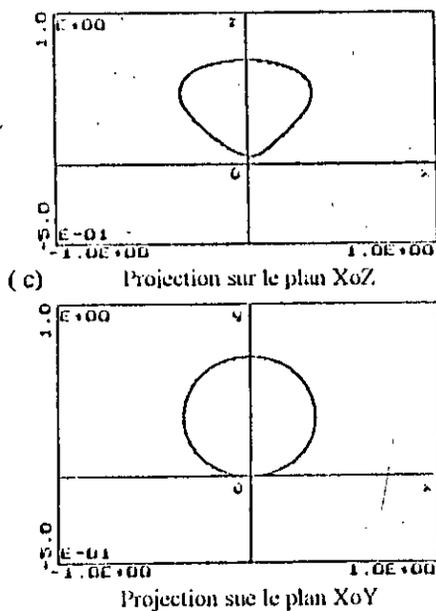
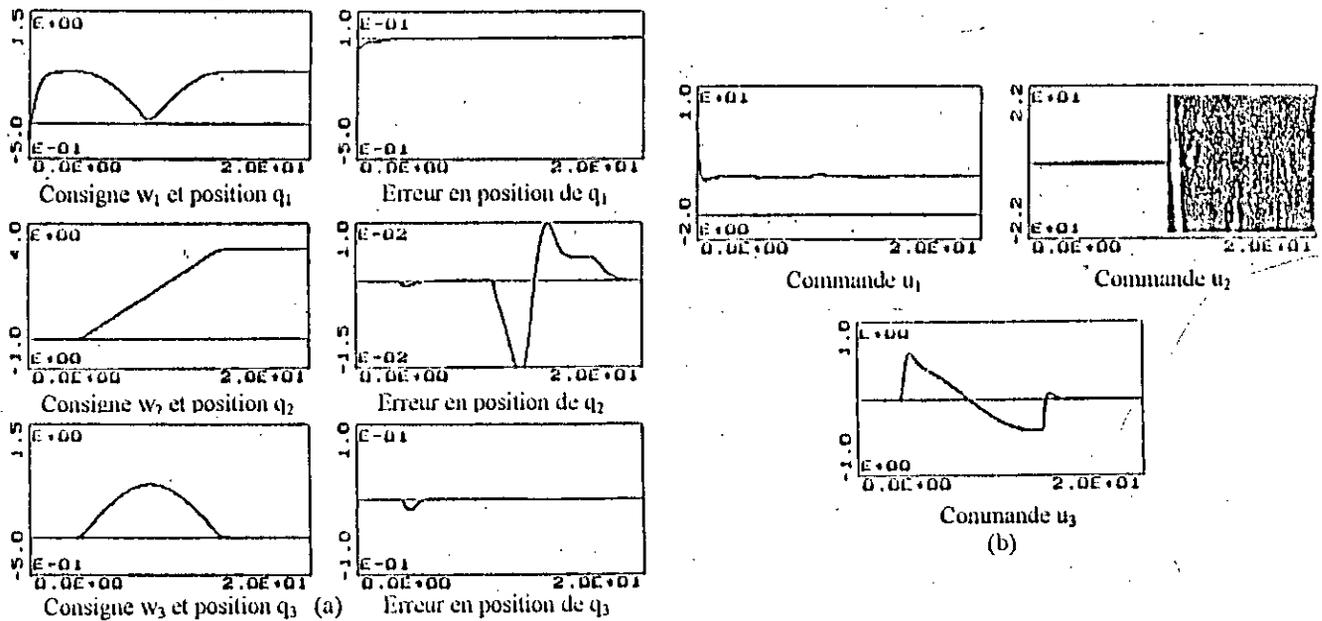


Figure IV.33 Réponses du robot pour la commande AMFC ( $\beta_i=60$ ) (Référence VIVIANI)

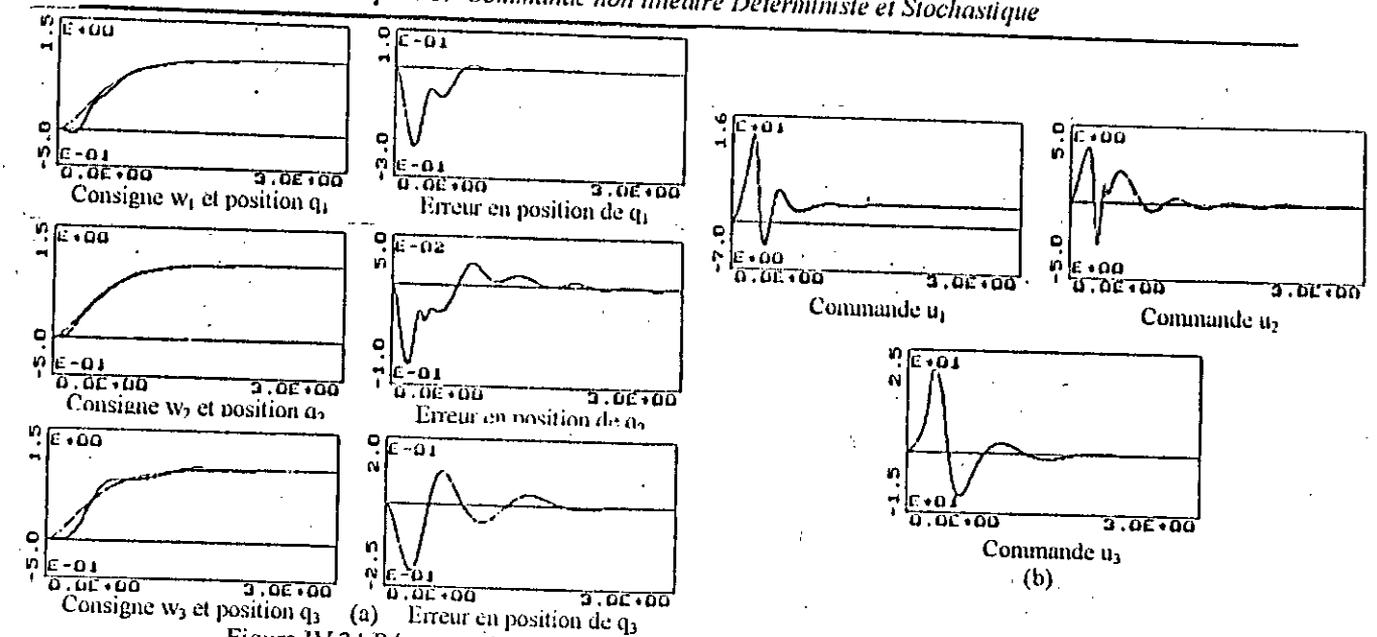


Figure IV.34 Réponses du robot pour la commande MCS (Référence Polynomiale)

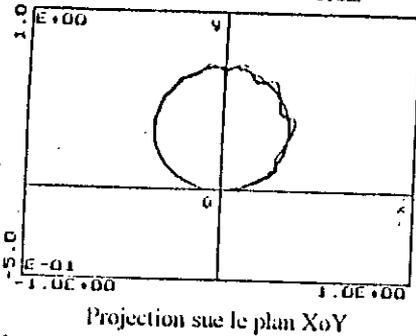
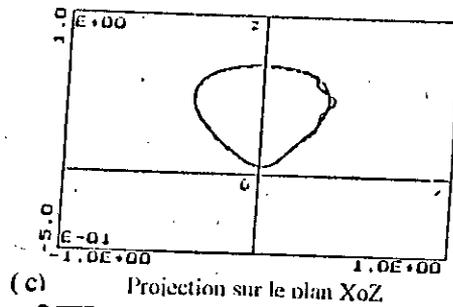
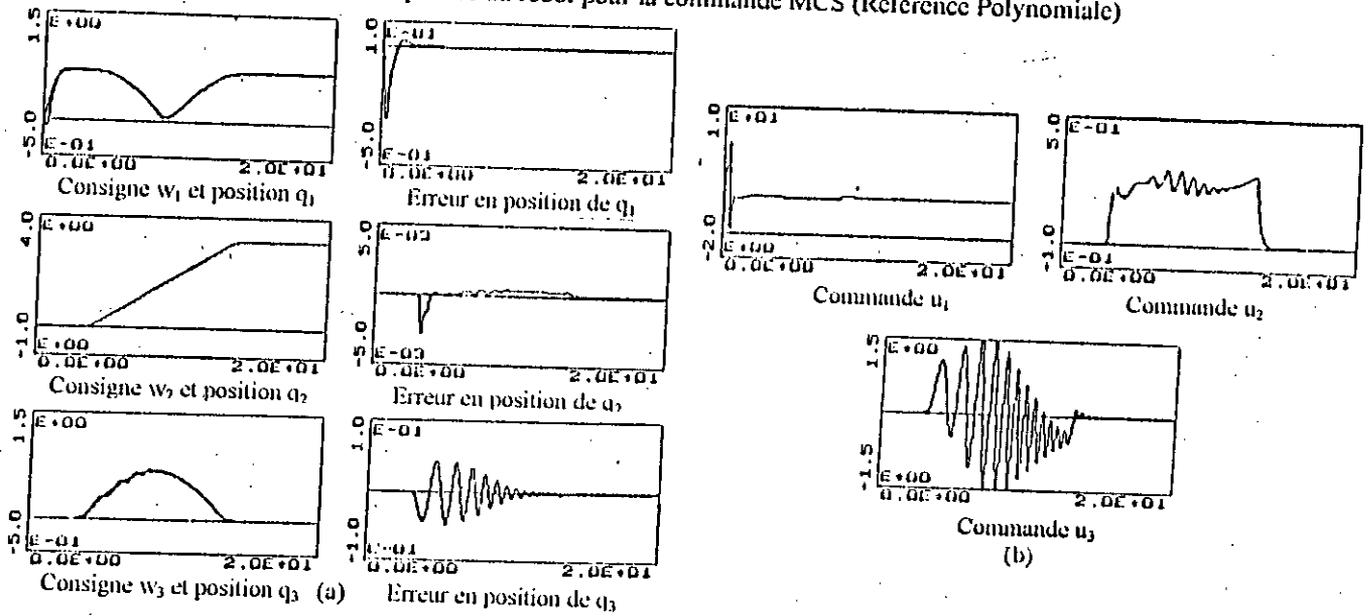


Figure IV.35 Réponses du robot pour la commande AMFC ( $\beta_1=1$ ) (Référence VIVIANI)

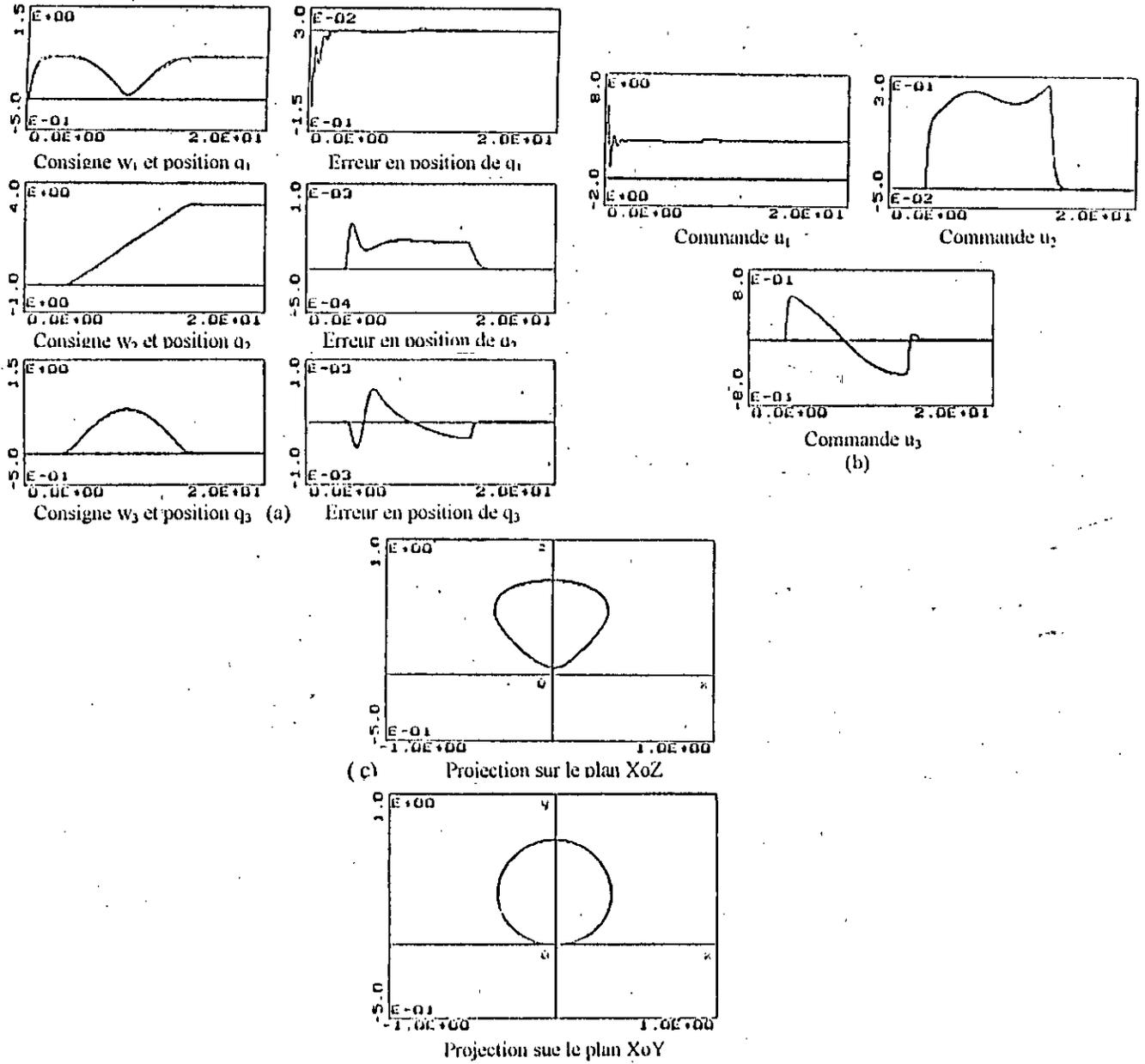


Figure IV.36 Réponses du robot pour la commande MCS (Référence VIVIANI)

## IV.15.2 Commande des structures en génie civil

Pour démontrer l'efficacité de la stratégie de commande présentée dans ce chapitre, nous avons effectué des tests de simulation sur la structure à un seul degré de liberté.

Les exemples d'apprentissage sont générés à partir de la commande à GMV monovariante avec un modèle d'excitation de Kanai-Tajimi qui est inclus dans le calcul du modèle ARMAX. Les valeurs des pondérations utilisées dans l'algorithme de la GMV sont :

$P_D(q^{-1}) = 1$ ,  $P_N(q^{-1}) = 1 - 0.5q^{-1}$ ,  $Q(q^{-1}) = 10^{-8}$ . L'accélération sismique de Kanai-Tajimi est utilisée comme excitation à la base de la structure. 150 exemples sont choisis pour entraîner le réseau avec l'algorithme de Backpropagation avec un pas d'adaptation de 0,2 pour la 1<sup>ère</sup> couche et 0,1 pour la 2<sup>ème</sup> couche. L'apprentissage a nécessité environ 2500 cycles pour atteindre une erreur acceptable.

Après la phase d'apprentissage, la structure a été soumise à différents types d'excitations (Kanai-Tajimi, Clough-Penzien, 200% Kanai-Tajimi) ainsi que des variations paramétriques (diminution de la masse de la structure de 20% à  $t=10s$ ) afin de tester la capacité de généralisation du réseau de neurones et comparer ses performances avec celles de la GMV.

Les réponses de la structure sont consignées sur les figures IV.37, IV.38, IV.39, IV.40 pour les différents cas.

Les résultats de simulation démontrent l'efficacité du régulateur à base de réseaux de neurones. Le réseau de neurones a pu compenser les réponses de la structure avec des performances très proches de celles obtenues par la GMV. Mais le résultat le plus important ici est que le réseau de neurones a pu compenser la variation paramétrique, due à la diminution de la masse, et a maintenu ses performances pendant tout l'intervalle de commande. Alors que les performances de la GMV se sont vite dégradées et l'on constate l'instabilité de la commande. La variation paramétrique a entraîné une grande modification du modèle ARMAX de la structure et par conséquent la GMV dérivée sur la base d'un modèle ARMAX initial est devenue inefficace.

Des tests de simulation de la stratégie de commande par réseaux de neurones développée ont été effectués en utilisant une structure à trois degrés de liberté. Le régulateur que nous avons approximé est celui de la GMV avec modèle empirique présentée dans le chapitre III. Pour une excitation de Kanai-Tajimi, 150 exemples sont générés par cette approche et utilisés pour entraîner trois réseaux de neurones par la méthode de Backpropagation avec un pas d'adaptation de 0,2 pour la 1<sup>ère</sup> couche et 0,1 pour la 2<sup>ème</sup> couche pour les trois réseaux. L'apprentissage a nécessité 2000 cycles pour les trois réseaux dédiés à la commande de l'étage 1, 2 et 3 respectivement, pour atteindre une erreur acceptable. Après l'accomplissement de la phase d'apprentissage, la structure a été soumise à plusieurs types d'excitations (Kanai-Tajimi, Clough-Penzien, 200% de l'excitation de Kanai-Tajimi) et à des variations paramétriques (diminution de la masse  $m_3$  de 40% à  $t=10s$ ) afin de tester l'efficacité des réseaux de neurones et leur pouvoir de généralisation et comparer leurs performances avec celles de l'approche empirique. Les réponses de la structure sont consignées sur les figures IV.41, IV.42, IV.43, IV.44 pour les différents cas.

Les résultats de simulation ont démontré l'efficacité des réseaux de neurones à compenser les réponses de la structure et présentent des performances très proches de celles de l'approche empirique. Bien que les réseaux de neurones ont été entraînés sur un ensemble d'exemples incluant seulement une excitation de Kanai-Tajimi, ils ont pu généraliser pour d'autres types d'excitations non incluses dans l'apprentissage et donnent de très bons résultats.

Dans le cas de la commande à gain programmé, une période d'échantillonnage  $T_e=0.02s$  est choisie pour la détermination des modèles ARMAX. Les paramètres du modèle sont calculés pour chaque point de fonctionnement.

Les réponses de la structure sans et avec commande à 300% de l'excitation de Kanai-Tajimi sont consignées sur la figure IV.45

Pour implémenter la version neuronale de la commande à gain programmé, on a choisi 25 points dans chaque domaine du plan  $(x_2, x_3)$  sur la base desquels l'apprentissage a été effectué. Le réseau utilisé possède 2 entrées, 10 neurones dans la couche cachée et 4 sorties. Les résultats de simulation de cette technique sont consignés sur la figure IV.46.

Les résultats de simulation montrent l'efficacité des techniques développées dans ce chapitre. On constate que le déplacement permanent de la structure dû à la non linéarité a été compensé et le déplacement relatif a été réduit d'une façon considérable. Le choix des polynômes de pondération permet de faire un compromis entre les performances voulues et la force de commande disponible.

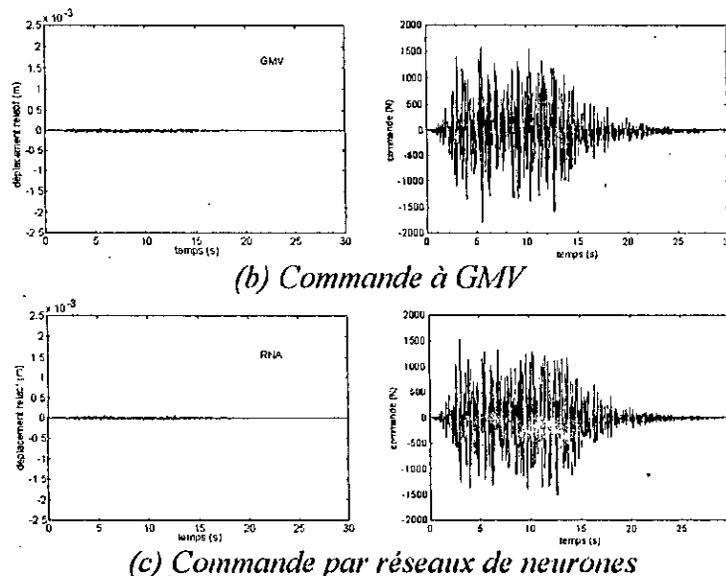
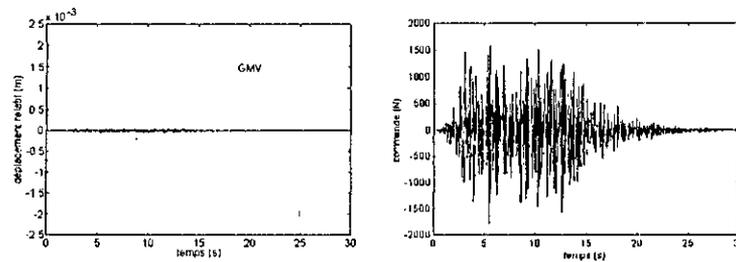
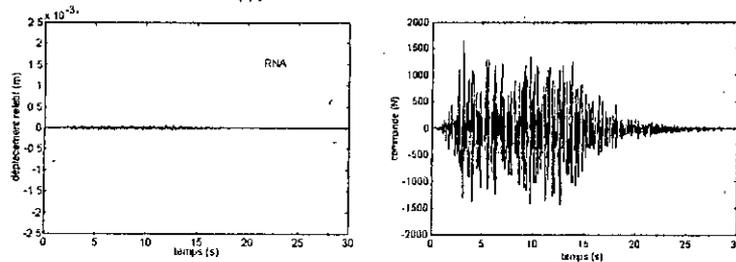


Figure IV.37. Réponse de la structure pour une excitation de Kanai-Tajimi

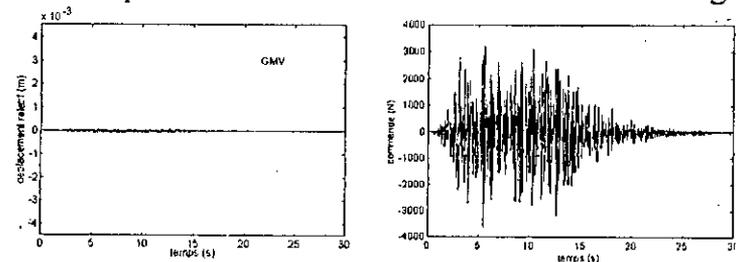


(b) Commande à GMV

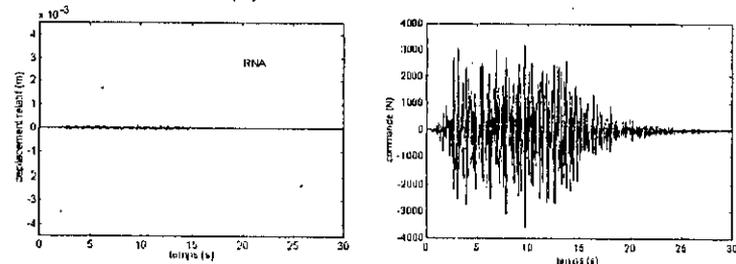


(c) Commande par réseaux de neurones

Figure IV.38 Réponse de la structure à une excitation de Clough-Penzien

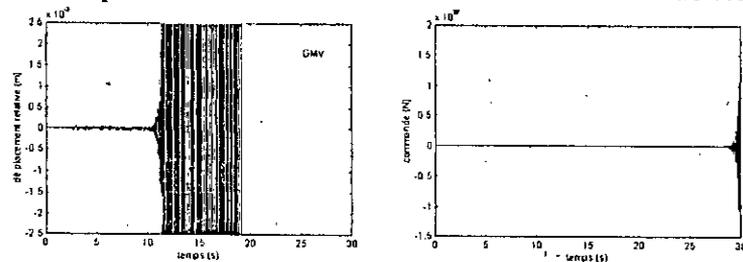


(b) Commande à GMV

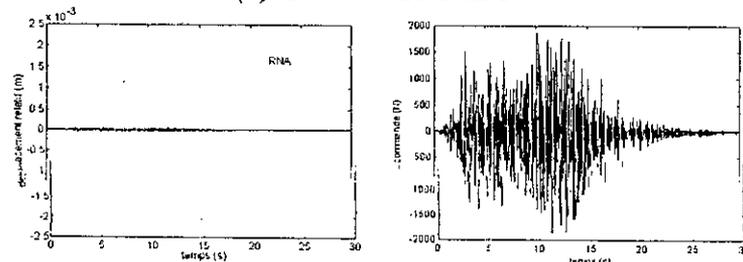


(c) Commande par réseaux de neurones

Figure IV.39 Réponse de la structure à 200% de l'excitation de Kanai-Tajimi

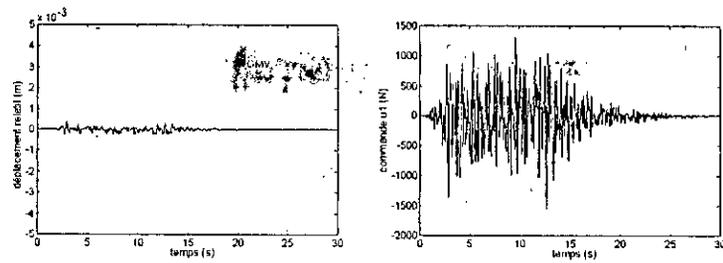


(b) Commande à GMV

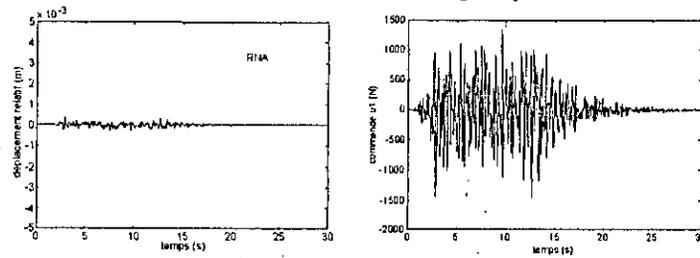


(c) Commande par réseaux de neurones

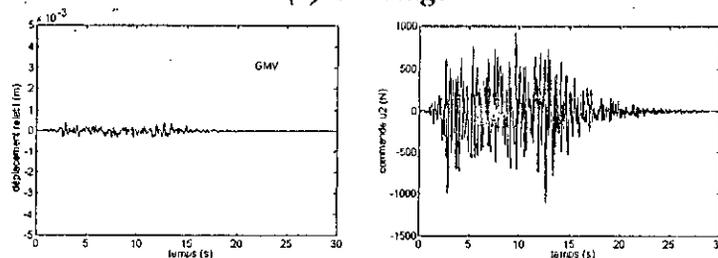
Figure IV.40. Réponse de la structure à une excitation de Kanai-Tajimi avec variation paramétrique (diminution de la masse de la structure  $m$  de 20% à  $t=10s$ )



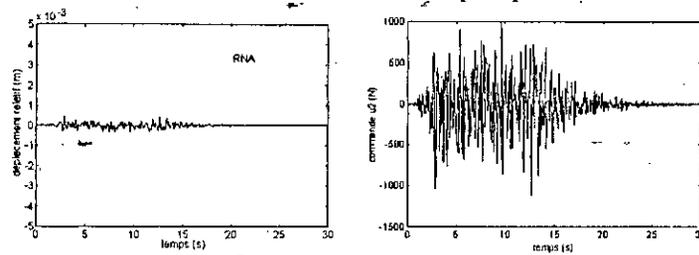
GMV avec modèle empirique



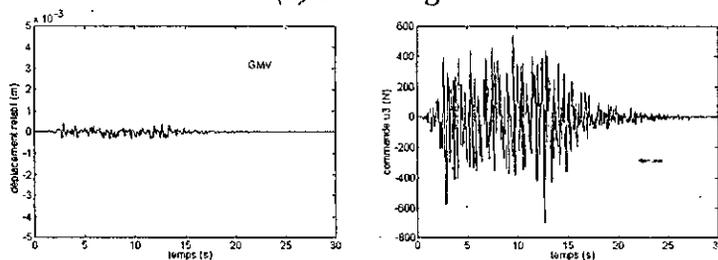
Commande par RNA  
(a) 1<sup>er</sup> étage



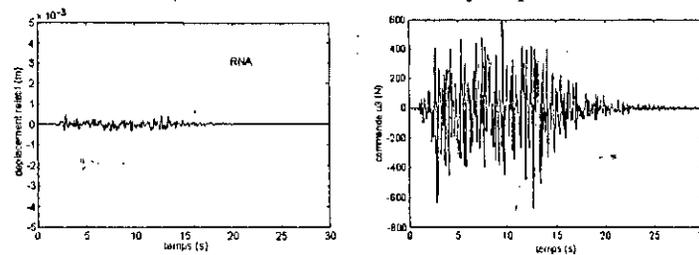
GMV avec modèle empirique



Commande par RNA  
(b) 2<sup>ème</sup> étage

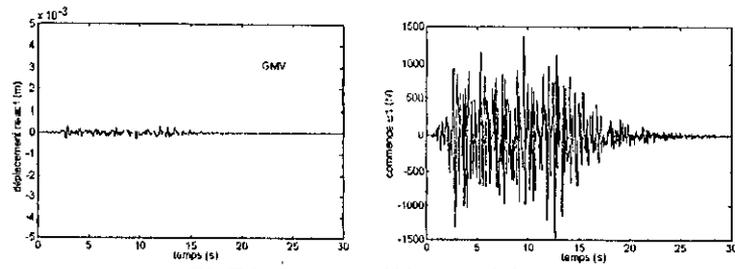


(GMV avec modèle empirique

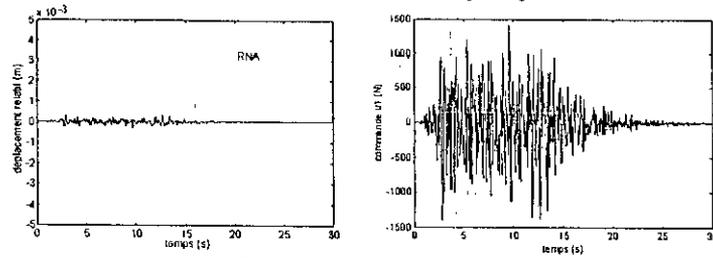


Commande par RNA  
(c) 3<sup>ème</sup> étage

Figure IV.41: Réponse de la structure à l'excitation de Kanai-Tajimi

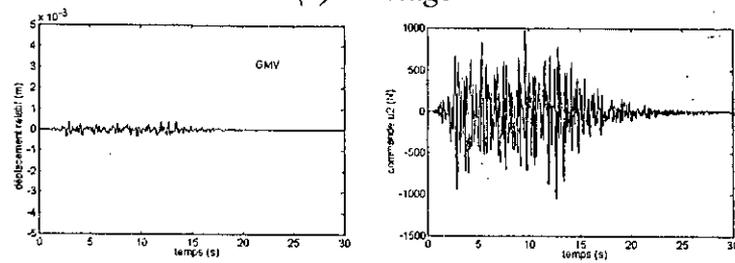


GMV avec modèle empirique

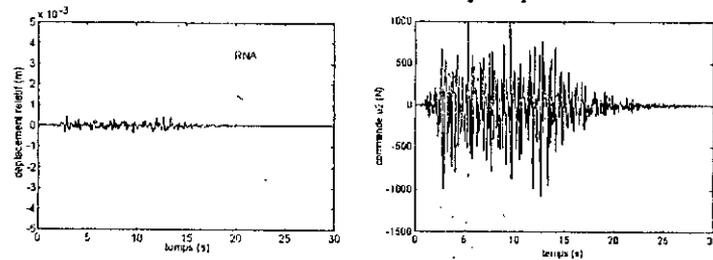


Commande par RNA

(a) 1<sup>er</sup> étage

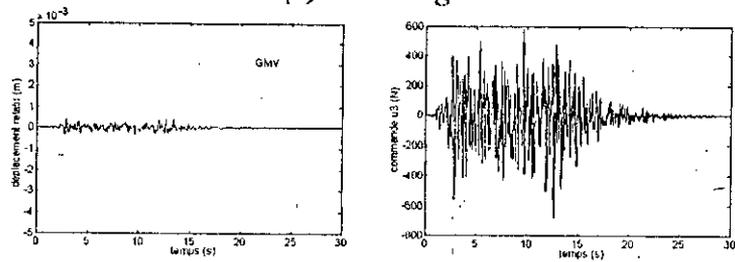


GMV avec modèle empirique

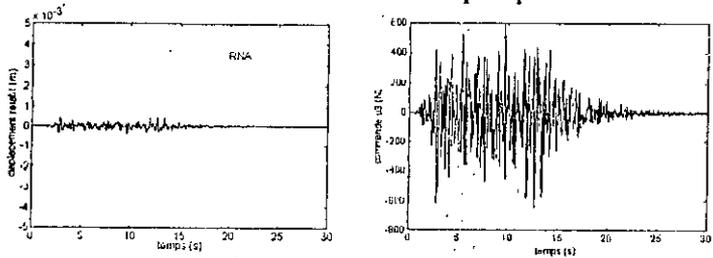


Commande par RNA

(b) 2<sup>ème</sup> étage



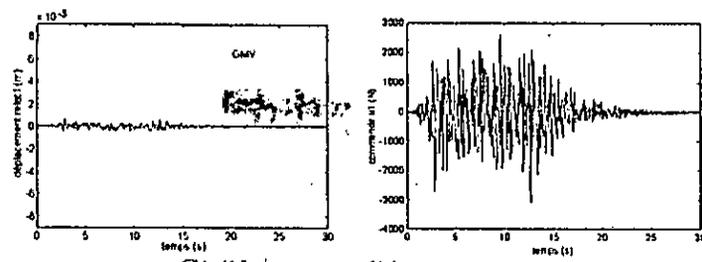
GMV avec modèle empirique



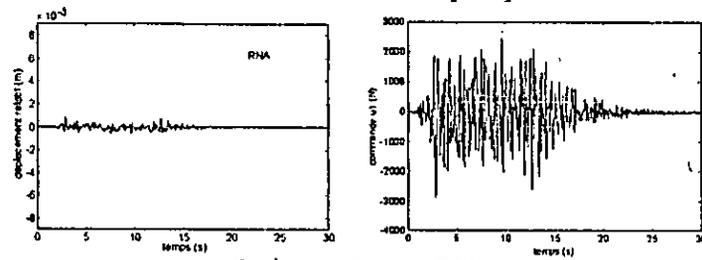
Commande par RNA

(c) 3<sup>ème</sup> étage

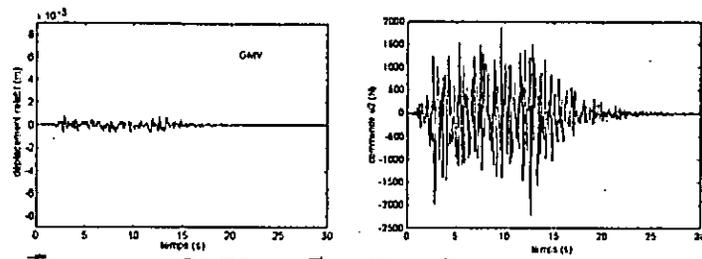
Figure IV.42 Réponse de la structure à l'excitation de Clough-Penzien



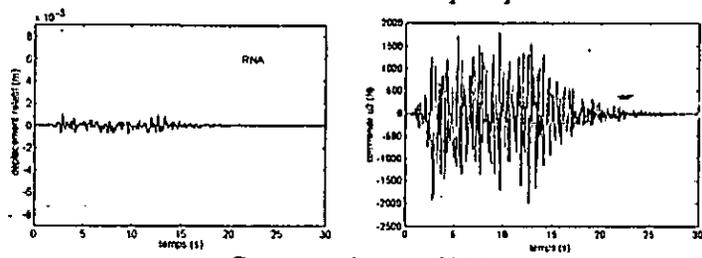
GMV avec modèle empirique



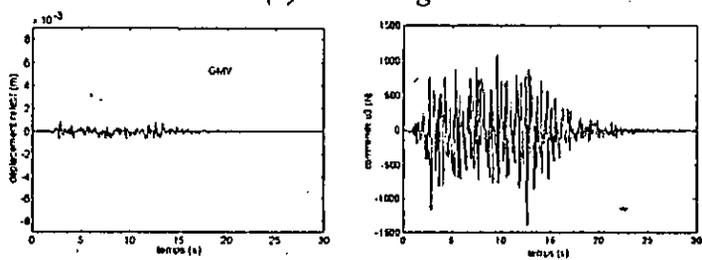
Commande par RNA  
(a) 1<sup>er</sup> étage



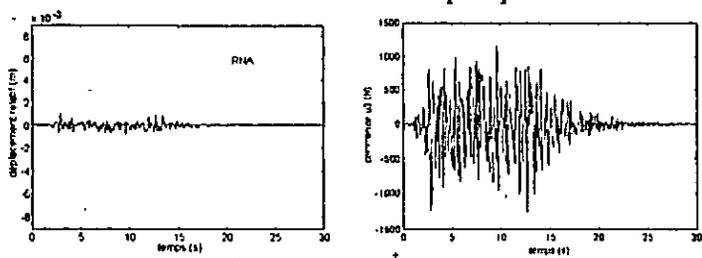
GMV avec modèle empirique



Commande par RNA  
(b) 2<sup>ème</sup> étage

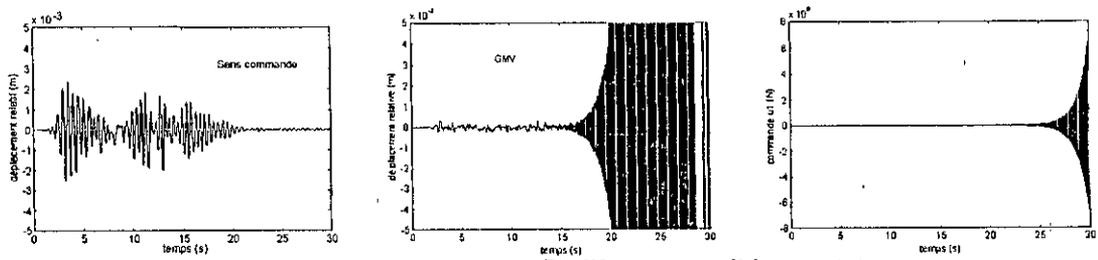


GMV avec modèle empirique



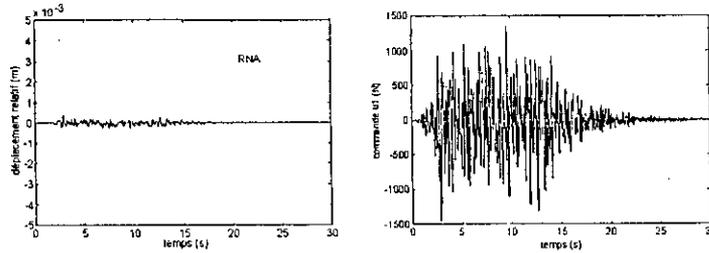
Commande par RNA  
(c) 3<sup>ème</sup> étage

Figure IV.43 Réponse de la structure à 200% de l'excitation de Kanai-Taji

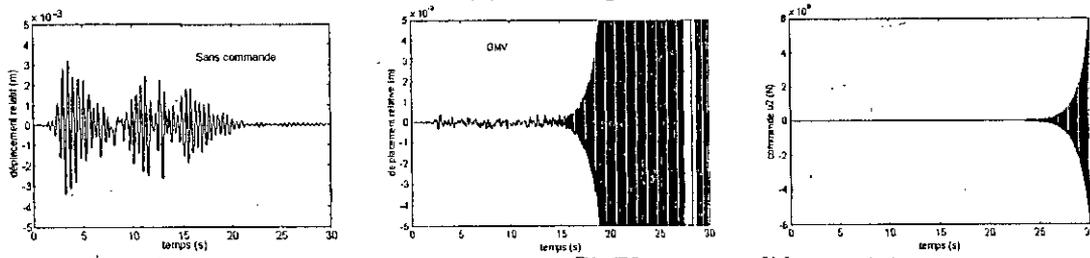


Sans commande

GMV avec modèle empirique

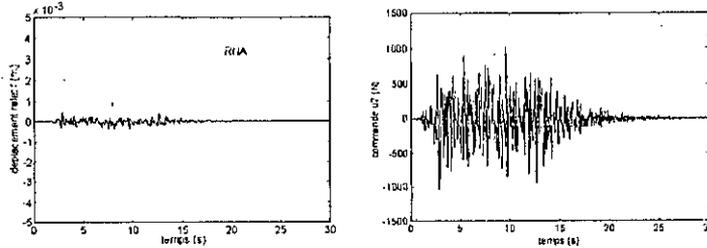


Commande par RNA  
(a) 1<sup>er</sup> étage

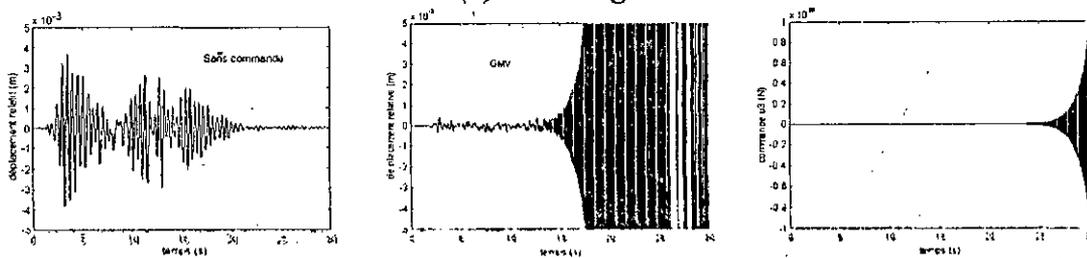


Sans commande

GMV avec modèle empirique

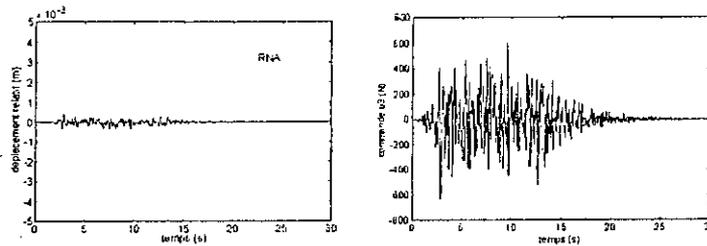


Commande par RNA  
(b) 2<sup>ème</sup> étage



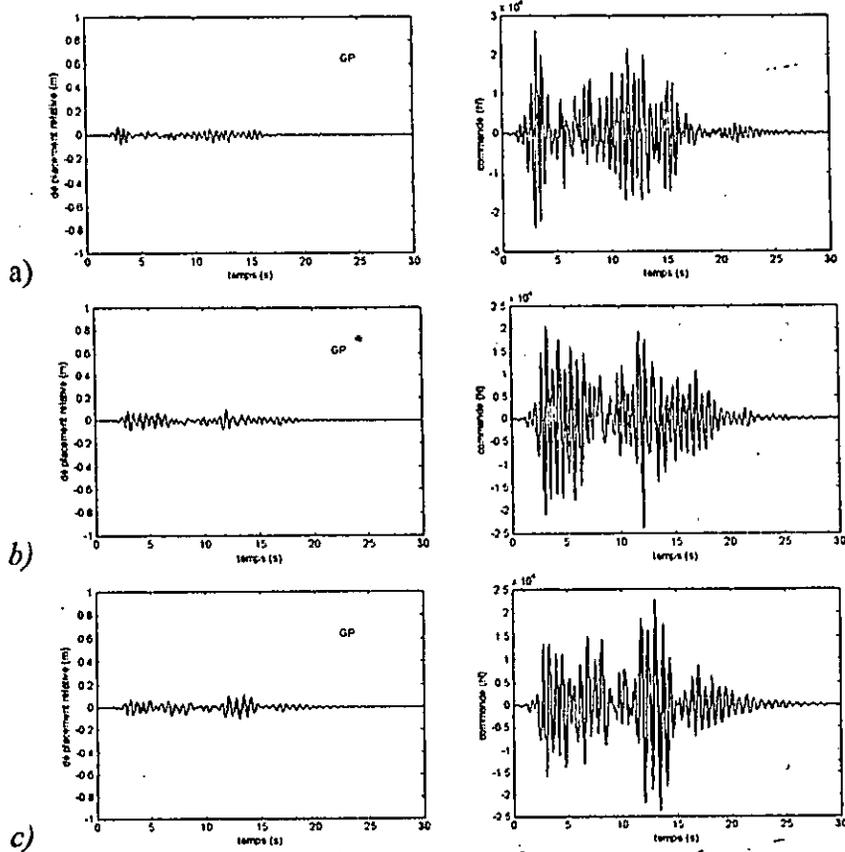
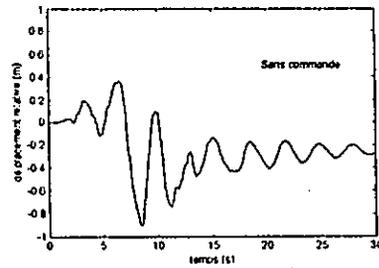
Sans commande

GMV avec modèle empirique



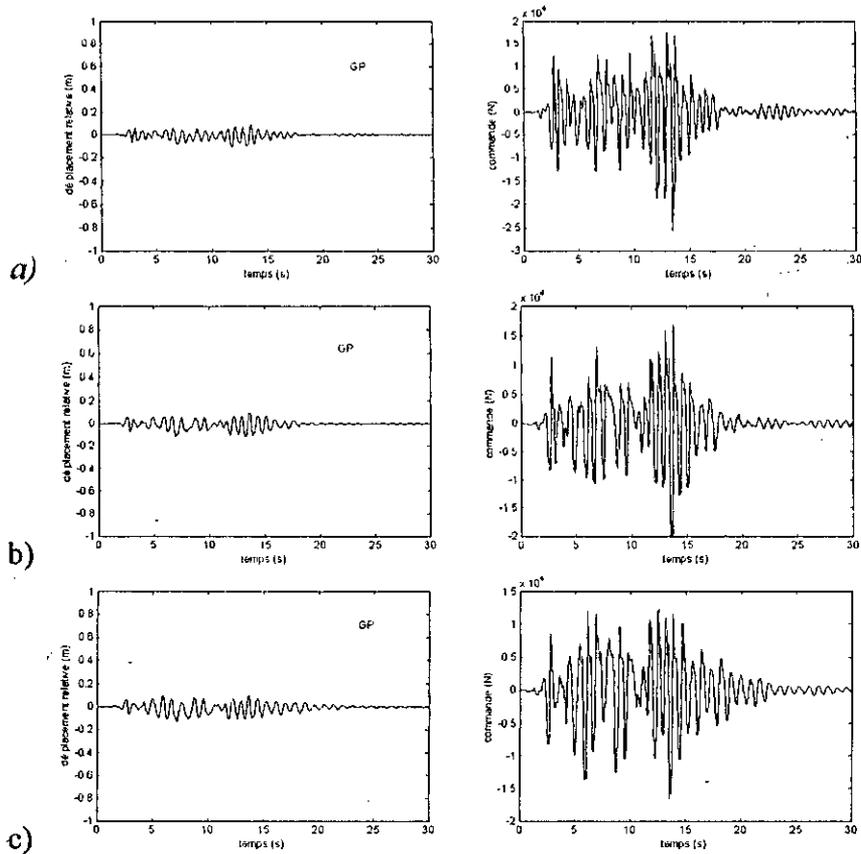
Commande par RNA  
(c) 3<sup>ème</sup> étage

(d) Figure IV.44 Réponse de la structure à l'excitation de Kanai-Tajimi avec variation paramétrique (diminution de la masse  $m_3$  de 40% à  $t=10s$ )



a)  $Q = 3 \cdot 10^{-6}$ , b)  $Q = 4 \cdot 10^{-6}$ , c)  $Q = 5 \cdot 10^{-6}$

Figure. IV.45 Réponses de la structure (commande à G.P.)



a)  $Q = 3 \cdot 10^{-6}$ , b)  $Q = 4 \cdot 10^{-6}$ , c)  $Q = 5 \cdot 10^{-6}$

Figure IV.46 Réponses de la structure (commande à G.P. avec classification par RNA)

## IV.15 Conclusion

Nous avons présenté dans ce chapitre les stratégies de commande non linéaire utilisant le modèle interne de la structure. La non linéarité est intrinsèque dans le cas des structures articulées. Par contre, pour les structures en génie civil, on distingue une non linéarité de comportement que nous avons surmonté par une linéarisation autour des points de fonctionnement. En robotique, c'est cette non linéarité en elle même qui est exploitée pour élaborer les lois de commande

La commande par compensation de gravité est une projection du robot de l'espace opérationnel à 3D sur l'espace bidirectionnel fictif, n'incluant pas l'effet de la gravitation. L'implémentation d'une telle commande nécessite la connaissance exacte du terme de gravitation dans l'équation du modèle du robot. L'application d'un algorithme à STR mixte permet la commande du nouveau modèle.

Dans le cas où le retour de compensation est issu d'un modèle erroné, l'algorithme de commande utilisé arrive à surmonter ce problème. La stratégie de commande obtenue est robuste envers des variations paramétriques et aux erreurs de modélisation. Dans notre cas, c'est la première articulation qui est sensible aux erreurs de modélisation du retour non linéaire.

L'obtention d'un modèle quasi-linéaire est assurée par le découplage non linéaire partiel. Le nouveau modèle obtenu nécessite le choix de la stratégie de commande nécessaire pour atteindre les performances voulues. La connaissance de l'expression analytique des termes non linéaires est d'une importance primordiale, pour assurer une compensation parfaite. Le modèle découplé est très sensible aux erreurs de modélisation. ces dernières peuvent être surmontées par l'application de la commande à STR mixte. La commande hybride obtenue est robuste envers des variations paramétriques, en utilisant l'algorithme d'identification à trace constante comme dans le cas précédent. La capacité de poursuite, d'une telle commande est d'une importance marquante, surtout lorsque la trajectoire est discontinue. Ce qui peut nuire d'une façon remarquable le retour non linéaire.

Un modèle linéaire découplé est obtenu par découplage non linéaire total. Le comportement du modèle obtenu est similaire à un double intégrateur. La commande d'un tel modèle nécessite une connaissance parfaite des paramètres du modèle du robot, ce qui exige une identification avant l'application de la commande.

L'utilisation de la géométrie différentielle est un outil puissant pour assurer une linéarisation exacte du robot. Le modèle linéaire obtenu est un double intégrateur totalement découplé. L'utilisation du modèle d'état nécessite une connaissance parfaite de l'expression analytique de tous les termes du modèle de connaissance du robot. Des discontinuités dans la vitesse des trajectoires désirées engendrent des oscillations autour de la consigne. Ce qui nécessite un lissage de la trajectoire (avec un polynôme du troisième degré) lors de l'application d'une telle commande

L'algorithme récursif de Newton-Euler, permet le calcul de la commande désirée, à partir des caractéristiques cinématiques de la trajectoire désirée (position, vitesse et accélération). La connaissance des caractéristiques paramétriques du robot est une étape importante pour l'élaboration d'un tel algorithme. Donc implicitement la connaissance du modèle du robot est nécessaire. La condition nécessaire pour l'implémentation d'une telle stratégie est la continuité de la trajectoire désirée en position, vitesse et accélération, et une connaissance exacte des conditions initiales (position). L'architecture du retour considéré est composée d'un modèle à perturbation et d'une stratégie de commande autoajustable à STR (Régulation). L'estimation utilisée donne le modèle à perturbation.

L'implémentation de la commande linéarisante à paramètres connu, nécessite la connaissance du modèle du robot et une continuité des trajectoires désirées en position et en vitesse. Dans de telles situations, cette commande est idéale. Le choix du modèle de référence  $(k_p, k_v)$  en BF est indépendant de la structure du robot, dans le cas où la matrice d'inertie est non singulière.

La connaissance du modèle linéaire du robot est nécessaire, pour le calcul des paramètres du régulateur linéaire (LMFC), dans le cas de AMFC. En annulant les paramètres de ce dernier régulateur (LMFC), on aboutit à un algorithme de commande adaptative, dont l'effort de commande est totalement synthétisé par la loi d'adaptation, c'est le MCS. La loi d'adaptation est "exactement" la même que dans l'AMFC.

En dynamique des structures génie civil nous avons remédié aux inconvénients de la commande à GMV par ses différentes approches présentées dans le chapitre III. Les performances de la GMV dépendent de la connaissance du modèle ARMAX du système à commander. Ainsi, nous avons utilisé une stratégie de commande par réseaux de neurones qui permet de palier à ce type de problème. Cette stratégie consiste à apprendre à un réseau de neurones à commander le système à partir d'exemples générés par le régulateur à GMV, puis laisser ce réseau commander le système en le soumettant à plusieurs tests afin d'analyser ses performances.

La stratégie de commande par réseaux de neurones possède un autre avantage du point de vue pratique qui est celui de l'architecture distribuée (parallélisme). Donc on aura une diminution du temps de calcul et la rapidité d'action par rapport à la GMV qui sera implementée dans un calculateur séquentiel.

La technique de commande à gain programmé est conceptuellement simple, mais les résultats de simulation ont démontré son efficacité à compenser les réponses des structures non linéaires. Elle a permis de compenser le déplacement permanent dû au comportement non linéaire de la structure et aussi de réduire considérablement le déplacement relatif.

La deuxième approche de commutation possède des performances proches de celle de la 1<sup>ère</sup> approche, avec un intérêt du point de vue implémentation. En effet, les réseaux de neurones avec leur caractère distribué permettent la réduction du temps de calcul (qui est dans notre cas le temps de choisir le régulateur adéquat à activer) et aussi une robustesse vis-à-vis des perturbations (pannes, dysfonctionnement). D'autre part l'emploi d'un réseau de neurones comme classificateur permet d'éviter l'ambiguïté qui se pose pour la commande à gain programmé lorsque l'état du système est à la limite entre deux plages de fonctionnement.

## Conclusion Générale

---

Dans ce travail, nous nous sommes intéressés à l'application de l'automatique et des réseaux de neurones en dynamique des structures. Nous avons présenté les différentes procédures de calcul des modèles de connaissance et de représentation des structures articulées et des structures en génie civil. Ensuite, une étude de synthèse sur les différents algorithmes de commande linéaires et non linéaires est donnée. L'intérêt est porté sur le comportement stochastique des structures en génie civil soumises à des excitations sismiques. Nous avons élaboré une nouvelle loi de commande stochastique permettant de minimiser d'une façon importante la variance de sortie. Dans le cas non linéaire, nous avons appliqué des stratégies de commande performantes pour la commande d'un robot de classe quatre utilisant différents types de références.

Pour les structures articulées, la modélisation cinématique nous permet la détermination des relations directes et inverses entre les coordonnées généralisées et les coordonnées cartésiennes. Le passage des coordonnées cartésiennes de l'élément terminal aux coordonnées généralisées des différentes articulations, présente des singularités. Cependant, ce type de problème peut être évité par l'utilisation de différent modèle cinématique. En effet, le choix de la trajectoire généralisée ou cartésienne, est une étape importante pour la commande des robots. Si des discontinuités dans la vitesse ou l'accélération existent, cela nécessite une interpolation polynomiale sur la trajectoire généralisée. Le modèle de connaissance dynamique, établi par le formalisme de Lagrange-Euler, nous permet la détermination de la trajectoire du robot connaissant les forces généralisées. L'algorithme de Newton-Euler permet une commande en boucle ouverte, malheureusement très sensible aux perturbations paramétriques et de sorties. Pour compenser ces perturbations nous utilisons une commande additive, élaborée par un algorithme de commande adaptative.

En dynamique des structures en génie civil, l'élaboration du modèle de connaissance s'est basée sur la procédure de la masse concentrée en utilisant le formalisme de l'équilibre dynamique des forces. Afin d'élargir la validité des modèles développés, une description du caractère non linéaire a été développée. Le modèle d'endommagement est le plus complexe des non linéarités lorsque sa description est décrite d'une façon continue. Par ailleurs la résolution analytique de son modèle dynamique est compliquée nécessitant des outils mathématiques théoriques complexes. L'introduction de l'excitation sismique dans le modèle de connaissance linéaire a permis la construction d'un modèle ARMAX spécifique pour l'excitation considérée. Ce modèle offre la possibilité de développer une loi de commande stochastique prenant en considération l'aspect aléatoire de l'excitation.

Nous avons présenté les concepts de base pour la compréhension et l'utilisation des réseaux de neurones. Nous avons procédé par une construction d'une architecture générale permettant d'englober tous les types de réseaux. Un modèle mathématique générale de l'élément de base du réseau a été donné. En effet les réseaux de neurones sont fortement sollicités pour la commande des systèmes non linéaires. Nous avons présenté alors les stratégies de commande les plus utilisées permettant ainsi un choix aisé de la structure de

commande neuronales. Dans la dernière section, Nous avons mis à la lumière les méthodes d'implémentation en temps réel des réseaux de neurones à savoir les transputers.

Nous nous sommes intéressés à la commande linéaire déterministe et stochastique des structures. La commande linéaire quadratique est synthétisée par minimisation d'un critère sur un horizon de temps infini afin d'éviter la résolution inverse de l'équation de Ricatti. Par contre la commande à variance minimale généralisée permet de surmonter le problème par résolution d'une autre équation appelée 'équation Diophantine'. En premier lieu, cet algorithme assure la poursuite et la régulation avec des dynamiques spécifiées. En second lieu, il prend en considération l'aspect aléatoire du séisme par une construction du modèle ARMAX et d'un critère de performances stochastique. Nous avons mis à l'évidence que la connaissance du modèle de l'excitation sismique et son introduction dans le modèle ARMAX permet d'améliorer d'avantages les performances du système en boucle fermée. On a obtenu une plus grande réduction des amplitudes de vibration avec un effort de commande plus réduit.

Dans le cas non linéaire, plusieurs stratégies de commande ont été développées. La commande hybride permet de minimiser l'effet des termes non linéaire pour donner des asservissements plus performants. Le choix de la compensation adéquate dépend du coût de l'implémentation pratique et des performances des régulateurs auto-ajustables utilisés.

La commande APC nécessite la disposition du modèle de connaissance du robot. L'algorithme de Newton-Euler, peut être utilisé, en temps réel ou en temps différé, pour le calcul de la commande désirée. Le retour auto-ajustable peut être soumis à une politique de gel, qui met en action ce retour dès que l'erreur de suivi est importante. L'utilisation de la vitesse et de l'accélération, dans l'algorithme de Newton-Euler, nécessite un continuité du second ordre (première et seconde dérivés) de la trajectoire désirée, pour assurer une poursuite parfaite.

La possibilité de paramétrisation linéaire du modèle de connaissance du robot, a permis la synthèse de la loi d'adaptation, dans le cas de la commande linéarisante adaptative. Le choix de la fonction de Lyapounov candidate est d'une importance primordiale pour l'étude de la stabilité d'un tel algorithme. Ainsi que le choix du compensateur linéaire affecte la convergence de l'erreur. La loi d'adaptation permet d'ajuster les paramètres du modèle de connaissance utilisé, pour le calcul des retours nécessaires. Les performances d'une telle commande sont remarquables.

Les systèmes adaptatifs à modèle de référence, permettent l'analyse de la convergence des algorithmes adaptatifs et la synthèse des lois d'adaptation. La mise sous forme d'un bloc SPR et d'un bloc non linéaire hyperstable, est une étape très importante dans cette analyse. Elle permet le choix de la loi d'adaptation, en fonction de l'algorithme de commande utilisé. Elle est applicable dans les deux cas: discret et continu. La construction de la structure parallèle-parallèle, à partir de l'algorithme de commande choisi, permet de trouver facilement la loi d'adaptation des coefficients du régulateur et d'assurer une convergence asymptotique de l'erreur.

La stratégie de commande par réseaux de neurones consiste à apprendre aux réseaux la commande du système à partir d'exemples générés par le régulateur à variance minimale. Nous avons constaté que les réseaux de neurones, par leur capacité d'apprentissage et de généralisation, ont montré de très bonnes performances même si la structure est sujette à des variations paramétriques. Nous avons proposé aussi une stratégie de commande pour les structures non linéaires. Il s'agit de commande à gain programmé. Elle consiste à synthétiser des régulateurs programmés de façon à accommoder le changement de la dynamique ou des paramètres de la structure dû à l'évolution de la non linéarité dans le temps. Cette stratégie permet d'avoir une action rapide de la commande face aux changements des conditions opératoires car les régulateurs sont synthétisés à l'avance. La deuxième approche de

commutation possède des performances proches de celle de la première, avec un intérêt du point de vue implémentation. En effet les réseaux de neurones avec leur avantage distribué permettent la réduction du temps de calcul (qui est dans notre cas le temps pour choisir le régulateur adéquat à activer), ainsi qu'une robustesse vis-à-vis des perturbations. En outre, l'emploi d'un réseau de neurones comme classificateur permet d'éviter l'ambiguïté posée par la commande à gain programmé lorsque l'état du système est à la limite entre deux plages de fonctionnement.

A l'issue de cette thèse, l'application de l'automatique et des réseaux de neurones en dynamique de structure reste toujours un domaine vierge pour lequel on peut apporter de nouveaux outils.

Dans le cas des structures fortement non linéaires, l'élaboration d'une loi de commande neuronale à base d'un critère de performance établi à partir d'un modèle neuronale s'avère intéressante. On aura recours alors à la théorie d'hyperstabilité de Popov afin de permettre l'étude de la stabilité de tels algorithmes. Il suffit alors de structurer l'algorithme sous forme d'un bloc directe strictement réel positif et d'un bloc de retour hyperstable. Ceci aboutit à un nouvel axe de recherche dans la théorie de la commande.

Afin d'implémenter ces algorithmes dans le cas réel, il serait souhaitable de refaire ces travaux pour les structures en génie civil, sur la base d'enregistrement accélérométriques de séismes réels et d'introduire la dynamique des actionneurs et des capteurs dans le modèle de connaissance.

Enfin, l'introduction de l'interaction sol-structure dans le modèle ARMAX reste toujours incomplète. Il est intéressant alors de développer un modèle ARMAX non linéaire incluant un modèle d'interaction plus réaliste. On aura alors à développer une loi de commande à variance minimale non linéaire dont on construira un prédicteur non linéaire à base de réseaux de neurones.

## Bibliographie

- [Afr91] H. Afra, '*Identification du comportement sismique de bâtiments à partir de leurs réponses accélérométriques*', Thèse de doctorat, Ecole Nationale des Ponts et Chaussées, France, 1991.
- [Arg90] P. Argoul, '*Identification des structures vibrantes*', Thèse de Doctorat, Spécialité : Structures et Matériaux, Ecole Nationale des Ponts et Chaussées, France, 1990.
- [Ars85] Z.VI.Arstein & A.Leizarowitz, '*Tracking periodic signals with the overtaking criterion*', IEEE Trans. Aut. Contr., vol AC 30, N°11, pp 1123-1126, Nov 1985.
- [Asa86] H.Asada & J.E.Slotinc, '*Robot analysis and control*', MIT, John Wiley and Sons, 1986.
- [Åst70] K. J. Åström, '*Introduction to stochastic control theory*', Academic Press, 1970.
- [Ast73] K.J.Åström & B.Wittenmark, '*On self tuning regulators*', Automatica, vol 9, pp. 195-199, 1973.
- [Åst77] K. J. Åström, '*theory and applications of self tuning regulators*', Automatica, vol. 13, No.99, pp. 457-476, 1977.
- [Ast78] K.J.Åström & Al, '*Self tuning controllers based on poles placement design*', Report of Lund Institute of Technology, dept. Auto. Contr., May, 1978,
- [Ast80] K.J.Åström, '*Robustness of a design method based on assignment of poles and zeros*', IEEE trans. auto. contr., Vol. 25, N°3, June 1980.
- [Ast80] K.J.Åström, '*Self tuning regulators: Design principles and application.*' in '*Application of adaptive control*', K.S.Narendra & R.V.Monopoli, Academic Press, 1980.
- [Åst89] K. J. Åström and B. Wittenmark, '*Adaptive control*', Addison-Wesley Publishing company, 1989.
- [Åst90] K. J. Åström and B. Wittenmark, '*Computer-controlled systems*', Prentice-Hall Inc., Englewood Cliffs, 1990.
- [Ath66] M.Athans & P.L.Falb, '*Optimal control: an introduction to the theory and its applications*', Macc Graw Hill, 1966.
- [Ath71] M. Athans, '*The role and the use of the stochastic linear-quadratic-gaussian problem in control system design*', IEEE Trans. on Automatic Control, vol. AC-16, No. 6, 1971.
- [Bab89] N. Baba, '*A new approach for finding the global minimum of error function of neural networks*' Neural Networks, vol. 2, pp. 367-373, 1989.
- [Ban98] K. Bani-Hani and J. Ghaboussi, '*Nonlinear structural control using neural networks*', J.eng. mech., ASCE, vol. 124, No. 3, pp.319-327, 1998.
- [Ban99] K. Bani-Hani, J. Ghaboussi, S. P. Schneider, '*Experimental study of identification and control of structures using neural network Part 1: Identification. Part2: Control*', Earthquake Eng. Struct. Dyn., vol. 28, pp.995-1018, 1999.
- [Ben91] M.Benallegue, '*Contribution à la commande dynamique adaptative des robots manipulateurs rapides*', Thèse de Doctorat de l'université de Paris 6, UPMC, 1991.
- [Boc91] L. Bochereau, '*Conception et interpretation de reseaux connexionnistes multichouches*', Thèse de doctorat de l'université de Paris 6, Spécialité informatique, France, 1991.
- [Bor79] U. Borison, '*Self-tuning regulators for a class of multivariable systems*', Automatica, vol.15, pp. 209-215, 1979.
- [Buh83] H.Büler, '*Réglages échantillonnés, volume 2:traitement dans l'espace d'état*', Press Polytechniques Romandes, 1983.
- [Cal98] A. J. Calise and G. D. Sweriduk, '*Active attenuation of building structural response using robust control*', J. eng. mech. ASCE, vol. 124, No. 5, pp. 520-528, 1998.
- [Cap95] D. Capecchi and F. Vestroni, '*Steady states dynamic Analysis of hysteretic systems*', Journal of Engineering Mechanics, vol.111, No.12, pp.1515-1531, 1995.
- [Cha87] V. V. Chalam '*Adaptive control systems: Techniques and applications*', Marcel Dekker, 1989.
- [Chan89] C.Y.Chan & H.R.Sirisena , '*Convergence of adaptive pole zero placement controller for stable non-minimum phase systems*', Int. J. Cntrol, Vol. 50, N°3, pp. 743-754, 1989.
- [Cha96] A. G. Chassiakos and S. F. Masri, '*Modelling unknown structural systems through the use of neural networks*', Earthquake Eng. Struct. Dyn., vol. 25, pp. 117-128, 1996.

- [Chi86] Chih-Min Lin & Al, '*Adaptive controller with desired pole/zero assignement*', IEE Proc., vol. 133, pt. D, N°6, pp. 301-306, Nov. 1986.
- [Chu89] L. L. Chung, R. C. Lin, T. T. Soog and A. M. Reinhorn, '*Experimental study of active control for MDOF seismic structures*', Journal of Engeneering Mechanics, vol. 115, No. 8, pp. 1609-1627, August 1989.
- [Cla75] D. W. Clarke and P. J. Gawthrop, '*self-tuning controller*', Proc. IEE, vol. 22, No. 9, pp.929-934, 1975.
- [Cla79] D. W. Clarke and P. J. Gawthrop, '*self-tuning control*', Proc. IEE, vol. 126, No. 6, pp.633-640, 1979.
- [Cla81] D. W. Clarke, '*Introduction to self tuning controller*', IEE, Control Eng. Series 15, H.Nichelson and B. H. Swanick, *self tuning and adaptive control : theory and application*, Peter pregrinus, 1981.
- [Cla84] D. W. Clarke, '*Self tuning control of non-minimum phase systems*', Automatica, vol. 20, No. 5, pp. 501-517, 1984.
- [Cla85] D.W.Clarke & Al, '*A generalised LQG approach to self tuning control. Part I, Aspects of design*', Int. J. Control, Vol 41, N°6, pp 1509-1529, 1985.
- [Clo93] R. W. Clough and J. Penzien, '*Dynamics of structures*', McGraw-Hill, New York, 1993.
- [Cra86] J.J.Craig, '*Introduction to robotics: Mechanics and control*', Addison-wesly, 1986.
- [Cra87] J.J.Craig & Al, '*Adaptive control of mechanical manipulators*', International Journal of robotics research, Vol. 6, N°2, pp. 16-29, Summer 1987.
- [Cra88] J.J.Craig, '*Adaptive control of mechanical manipulators*', Adisson Wesley, 1988.
- [Cun88] Y. Le Cun, '*A theoretical framework for Bach propagation*', Proceeding of this connectionnist Models, Sumer school, page 21-29, 1988.
- [Dar91] E. Daraload P. Naim, '*Neural Network*', Macmill Education 1991 Edition cyrolles 1990.
- [Del95] A. Delgas, C. Kambrampati and K. Warwick. '*Dynamic recurent neural network for system identification and control*', IEE proc-control theory appl, vol 142, 104, July 1995 pp. 307-314.
- [Den55] J.Denavit & Al, '*A kinematic notation for lower pair mechanisms based on matrices*', Journal of applied mechanics, pp. 215-221, June 1955.
- [Dio86] J. M. Dion and L. Dugard '*Commande adaptative multivariable: quelques resultats théoriques et pratiques*', APII 20, pp. 337-358, 1986.
- [Dje00] M. Djebiri, '*Developpement de commandes à variance minimale et à reseaux de neurones: application en dynamiques des structures*', Thèse de Magister à l'ENP, Département de G. Electricue, Automatique, 2000.
- [Dug84] L.Dugard & Al, '*The role of the interactor matrix in multivariable stochastic adaptive control*', Automatica, Vol. 20, N°5, pp 701-709, 1984.
- [Els61] LE. Elsgolc, '*Calculs of variation*', Pergamon Press, 1961.
- [Fol96] G. C. Foliente, M. P. Singh and M. N. Noori, '*Equivalent linearization of generally pinching hysteretic, degrading systems*', *Earthquake Eng. Struct. Dyn.*, vol. 25, pp. 611-629, 1996.
- [Fre92] J. A. Freeman and D. M. Skapura, '*Neural network : Algoritms, Applications and programing techniques*', Addison-Wesley Publishing company, 1992.
- [Fri87] B.Friedland, '*Control system design: An introduction to state space methods*', Macc Graw Hill, 1987.
- [Fu87] K.S.Fu & Al, '*Robotics: Control, Sensing, Vision and intelligence*', Mcc Graw Hill, 1987.
- [Fuj88] A. T. Fujita, Q. Feng, E. Takena, T.Takano and Y.Suizu, '*Active isolation of sensitive equipement for weak earthquakes*', Proceeding of Ninth work conference on earth quake engineering August 2-9, Tokyo-Kyoto, Japan (vol VIII), 1988.
- [Fuk92] T. Fukuda and T. Shibata, '*Theory and applications of neural networks for industrial control systems*', IEEE Trans. Industrial Electronics, vol. 39, No. 6, pp. 472-489, 1992.
- [Gaw77] P.J.Gawthrop, '*Some interpretations of self tuning controller*', proc. IEE, vol. 124, N°10, pp. 889-894, Oct. 1977.
- [Goo81] G. C. Goodwin and Al '*Discrete time multivaivable adaptive control*', SIAM J. cont. And optimization, vol 19, N°6, pp.829-853, Nov 1981.
- [Goo81] G.C.Goodwin & Al, '*Discrete time stochastic adaptive control*', SIAM J.Control and Optimization, vol 19, N°6, pp 828-853, Nov 1981.
- [Goo84] G.C.Goodwin & Kwai Sang Sin, '*Adaptive filtering prediction and control*', Prentice Hall, 1984.
- [Gra89] L. C. J. M. Gras and H. Nijmeijer, '*Decoupling in nonlinear systems : from linearity to nonlinearity*', IEEE pro. Vol.136, pt. D, No2, pp 53-62, March 1989.
- [Gri84] M.J.Grimble, '*Implicit and explicit LQG self tuning contollers*', Automatica vol. 20, N°5, pp 661-669, 1984.
- [Gue95a] L. Guenfaf, '*Etude de Différentes stratégies de commande adaptative : Application à un robot manipulateur*', Thèse de Magister à l'ENP, Département de G. Electricue, Automatique, 1995.
- [Gue95b] L. Guenfaf, N. Bali et M.S. Boucherit, '*Application des techniques auto-ajustables dans le cas de la robotique*', 5<sup>ème</sup> colloque Maghrebien sur les modèles numeriques de l'ingenieur, Rabat, 1995.
- [Gue95c] L. Guenfaf, N. Bali et M.S. Boucherit, '*Commande auto-ajustable multirate à variance minimale généralisée du robot Portos*', ROB'95, Journée d'étude sur la robotique et son environnement, Ecole militaire Polytechnique, 1995, Algeria.

- [Gue95d] L. Guenfaf, N. Bali et M.S. Boucherit, '*Identification récursive des systèmes multivariables : Etude de différentes structures*', *Special issue of AJOT, Proc. Of 2<sup>nd</sup> CEA '94*, pp. 150-153, 1995.
- [Gue96a] L. Guenfaf, N. Bali et M.S. Boucherit, '*Performances Study of Multirate generalized Minimum Variance Algorithm applied to Robotic Manipulators*', ITHURS'95, International 1995 AMSE conferences ENG'95 with IEEE.
- [Gue96b] L. Guenfaf, M.S. Boucherit, R. I. Iloul and, N. Bali, '*Commande adaptative à perturbation : application aux robots manipulateurs*', COMAIE'96, Conference Maghrebien sur l'automatique, l'Electrotechnique Industrielle, 1996.
- [Gue00a] L. Guenfaf, N. Bali et M.S. Boucherit, '*Commande non lineaire Hybride par decouplage utilisant la géométre différentielle*', ICEL'2000, Oran, pp.342-347, International conference on Electrotechnics, Nov 2000.
- [Gue00b] L. Guenfaf, N. Bali et M.S. Boucherit, '*Commande nonlineaire autoajustable d'un bras de robot Manipulateur*', ICEE, Boumerdes 2000, First International Conference on Electrical Engineering, Nov 2000.
- [Gue01] L. Guenfaf, M. Djebiri, M. S. Boucherit and F. Boudjema, '*Generalized minimum variance control for buildings under seismic ground motion*', *Earthquake Eng. Struct. Dyn.*, vol. 30, Issue 7, pp945-960, 2001.
- [Gue88] A. Guez and J. Selinsky, '*A trainable neuromorphic controller*', *Journal of Robotic Systems*, vol. 5, pp. 363-388, 1988.
- [Had96a] M. Hadid, H. Afra and L. Guenfaf, '*Active control techniques for building inder seismic ground Motion*', 1<sup>er</sup> colloque National de genie parasismique, ENSAG, Bouzereah, vol.1, pp.157-163, Juin 1996.
- [Had96b] M. Hadid, L. Guenfaf and H. Afra, '*Linear Optimal Control Theory applied to earthquake Engineering Problems*', Acte de la 2<sup>eme</sup> conférence Maghrebine sur l'automatique, l'électrotechnique et l'électronique industrielle, COMAEI 96, Tlemcen, Algiers, vol.1, pp. 43-47, Dec 1996.
- [Had97] M. Hadid, H. Afra and L. Guenfaf, '*Active control Application for MDOF structures inder earthquake excitation*', First Arab Mechanics congress, Damascus, Syria, 1-3 June, vol.3, pp.48-53, 1997.
- [Hen88] M. A. Hensen and D. E. Seborg, '*A critique of differential geometric control strategies for process control*', 4<sup>th</sup> IFAC congress, pp.1-8, 1988.
- [Hir90] E. Hirsh, '*Les transputers : Application à la programmation concurrentes*', Eyrolles 1990.
- [Hou97] G. W. Housner, L. A. Bergman, T. K. Caughey, A. G. Chassiakos, R. O. Claus, S. F. Masri, R. E. Skelton, T. T. Soong, B. F. Spencer and J. T. P. Yao, '*Structural control: past, present and future*', *J. Eng. Mech.*, Special issue ASCE, vol. 123, No. 9, pp. 897-970, 1997.
- [Hun92] K. J. Hunt, D. Sbarbaro, R. Zbikowski and P. J. Gawthrop, '*Neural networks for control systems - a survey*', *Automatica*, vol. 28, No. 6, pp. 1083-1112, 1992.
- [Hun99] K. Hung and R. Betti, '*Predictive optimal control for seismic analysis of non linear and hysteretic structure*', *Earthquake Eng. Struct. Dyn.*, Vol. 28, 585-607, 1999.
- [Isi85] A. Isidori, '*Nonlinear control systems: A introduction*', Springer Verlag, 1985.
- [Iwa86] W. D. Iwan and A. O. Cifuentes, '*A model for system identification of Degrading structures*', *Earthquake Eng. Struct. Dyn.*, vol.14, 877-890, 1986.
- [Jez95] L. Jezequel, V. Roberti, '*Adaptative semi active isolation of seismic buildings*', 18<sup>eme</sup> Seminaire regional European de genie parasismique', Ecole centrale de Lyon, pp385-400, Sep. 1995.
- [Kin92] M. Kinnaert & V. Blondel, '*Discret-time pole placement with stable controller*', *Automatica*, Vol. 28, N°5, pp. 935-943, 1992.
- [Koh80] K. kohonen, '*A introduction to Neural computing*', *Neural Network*, vol. 1, pp. 3-16, peganon Press, 1980.
- [Kor85] Y. Koren, '*Robotics for engineers*', Macc Graw Hill, 1985.
- [Kos92] B. Kosko, '*Neural Network and Fuzzy systems*', Pentice Hall Inc., 1992.
- [Koi80] H. N. Koivo, '*A multivariable self-tuning controller*', *Automatica*, vol. 16, pp. 351-366, 1980.
- [Kun93] S. Y. Kung, '*Digital Neural Network*', Prentice Hall Inc. 1993.
- [Kwa72] H. Kwakernaak & R. Sivan, '*Linear optimal control systems*', John Willery and sons, 1972.
- [Lac94] F. Lacassagne, '*Etude et Parrerllelisation de methodes d'optimisation directe: Application à la programmation dynamique et au systeme non lineaire*', these de Doctorat d'état, laboratoire LAAS, université Paul Sabatier de Toulouse, Spécialité: Automatique, 1994 France.
- [Lan74] I. D. Landau, '*A survey of model reference adaptive techniques: theory and applications*', *Automatica*, pp. 353-379, 1974.
- [Lan79] I. D. Landau, '*Adaptive Control : the model reference approach*', Marcel Dekker, 1979.
- [Lan86] I. D. Landau and L. Dugard, '*Commande adaptative: Aspects pratiques et théorique*', Masson, 1986.
- [Lan93] I. D. Landau, '*Identification et commande des systèmes*', Hermès, Paris, 1993.
- [Lan86] S. J. Lang, X.-Y. Gu and T.-Y. Chai, '*A multivariable generalized self-tuning controller with decoupling design*', *IEEE Trans. Automatic Control*, vol. AC-31, No. 5, 1986.

- [Lan89] I.D.Landau & R.Horowitz , '*Application of the passive systems approach to the stability analysis of adaptive controllers for robot manipulators*', International Journal of adaptive control and signal processing, vol. 3, pp. 23-38, 1989.
- [Lee91] S. Lee and R. H. Kil, '*A gaussian Potential Function Network whith Hieirchically self organizing learning*', Neural Network, vol.4, pp. 207-224, 1991.
- [Lei86] A.Leizarowitz, '*Tracking non periodic trajectories with the overtaking criterion*', Applied mathematics and optimization, Springer Verlag, New York, 14, pp 155-161, 1986.
- [Li95] C. J. Li and L. Yan and N. W. Chbat, '*Powell's method applied to learning neural control of three inkhoun dynamic systems*', Optimal control applications and method, vol.16, pp. 251-262 1995.
- [Loh96] C. H. Loh and H.-M. Lin, '*Application of off-line and on-line identification techniques to buildings seismic response data*', Earthquake eng. struct. Dyn., vol. 25, pp. 269-290, 1996.
- [Li86] M.Li De La Sen, '*A model reference adaptive control system for discret multivariable bilinear systems interconected sudsystem*', IEE proc., Vol. 133, pt D, N°4, pp 165-171, July 1986.
- [Li89] Li.Mo & M.M.Bayoumi, '*A novel approach to the explicit pole assignement self tuning contoller design*', IEEE. Trans. automatic control, Vol. 34, N°3, pp. 359-363, March 1989.
- [Lin87] Y. K. Lin and Y. Yong, '*Evolutionary Kanai-Tajimi earthquake models*', J. eng. Mech. ASCE, vol. 113, No. 8, pp. 1119-1137, 1987.
- [Lin89] B. C. Lin, I. G. Tadjbakhsh and A. S. Papageorgiou, '*Response of base-isolated buildings to random excitations described by th Clough-Penzien model*', Earthquake eng. struct. Dyn., vol. 18, pp. 49-62, 1989.
- [Lip87] R P Lipmann, '*A Introduction computing whith Neural Nets*', IEEE Acoustics, Speech and signal processing Magazine, pp 4-22, April 1987.
- [Liu99] D. A. Liut, E. E. Matheu, M. P. Singh and D. T. Mook. '*Neural network control of building structures by a force matching training scheme*', Earthquake eng. struct. Dyn eng 28, 1601-1620, 1999.
- [Lop84] P.Lopez & J.N.Foulc, '*Introduction a la robotique*', 2 tomes, Editest, Paris 1984.
- [Lou90] N.Louam, '*The application of linear optimal control theory to the design active automotive suspension*', PhD Thesis, Leeds universsity, UK, 1990.
- [Lu98] L.T. Lu, W.-L. Chaing and J.P. Tang. '*LQG/LTR control methodology in active structural control*', J. eng. mech. ASCE, vol. 124, No. 4, pp. 446-454, 1998.
- [Luc98] J. E. Luco '*A simple model for structural control including soil-structure interaction effects*', Earthquake eng. struct. Dyn, vol.27, pp.225-242, 1998.
- [Mah89] F.Mahieddine & A.S.Morris, '*Decoupling multivariable Self tuning controller for varying time delays*', IEE, proc., Vol. 136, pt. D, N°5, pp 209-214, Sept 1989.
- [Mei87] Mei-Hua Liu & Wei lin, '*Pole assignemnet self tuning controller for robotic*' Int. J. Control, Vol 46, N°4, pp 1307-1317, 1987
- [Mei97] L. Meirovitch and T. J. Stemple, '*Nonlinear control of structures in Earthquakes*', Journal of Engineering Mechanics, pp.1090-1095. October 1997.
- [Mic99] A. N. Michel and J. A. Farrcl, '*Associative Memories via Artificial Neural Networks*', IEEE control systems Magazine, pp.6-17. Avril 1999.
- [Mil92] W. T. Miller, R. S. Sutton and P. J. Werbos, '*Neural Networks for Contro*', The MIT Press, Massachusetts, 1992.
- [Mon86] G.A.Montague & Al, '*Performance evaluation of three multivariable self tuning controller design techniques*', IEEE, proc. of 25<sup>th</sup> conference on decision and control, pp. 1564-1569, 1986.
- [Mor81] A.Morris & Al, '*Single and multivariable applicationof self tuning contollers*', IEE, Control Engineering, serie 15, H.Nicholson & B.H.Swanick 'Self tuning and adaptive control theory and application' Peter Pregrinus, 1981.
- [Mos99] N. Mostaghel, '*Analytical description of pinching ,Degrading Hysteretic systems*', Journal of Eng. Mech., vol.125, 102, pp.216-224, February 1999.
- [M'sa] M.M'Saad & G.Sanchez, '*Partial state reference model adaptive contol of multivariable systems*', Automatica, vol. 28, N°6, pp. 1189-1197, 1992.
- [M'si93] N.K.M'Sirdi, '*Modélisation, analyse et commande de processus: Application à la commande de systèmes robotisés*', Genie robotique et productive, LRP, Université de Versailles St Quentin (UVSQ), 1993.
- [Mul90] B. Muller, '*Neural Networks*', Springer Verlg, 1990.
- [Nar90] K. S. Narendra and K. Parthasarathy, '*Identification and control of dynamical systems using neural networks*', IEEE Trans. Neural networks, vol. 1, No. 1, pp. 4-27, 1990.
- [Ni99] Y. Q. Ni, J. M. Ko and C. W. Wong. '*Nonparametric Identification of nonlinear Hysteretic Systems*', Journal of Engineering Mechanics, pp.206-215, February1999.
- [Pag99] L. C. Pagnini and G. Solar, '*Stochastic analysis of the linear devices*', Earthquake eng. struct. Dyn., 28, 543-560, 1999.
- [Pan97] G. F. Panariello, R. Betti and R. W. Longman, '*Optimal structural control via training on ensemble of Earthqurakes*', Journal of Engineering mechanics, pp. 1170-1179. november 1997.

- [Pau81] P.P.Paul & Al, '*Kinematic control equations for simple manipulators*' IEE. Trans. on System, Man and cybernetics, Vol. SMC 11, N°6, pp. 449-455, June 1981.
- [Psa88] S. Psaltis, A. Sideris and A. A. Yamamura, '*A multilayered neural network*', IEEE, pp17-21, April 1988.
- [Pop73] V.V.M. Popov, '*L'hyperstabilité des systèmes automatiques*', Dunod 1973.
- [Put87] S.C.Puthempura & J.F.Macgregor, '*Pole zero placement controllers and self tuning regulators with better set point tracking*', IEE proc, vol 134, Pt D, N°1, pp. 26-30, January 1987.
- [Rah87] N.A.Rahim, '*Adaptive control of robot manipulators*' thèse doctorat ' University of Leeds departement of mechanical Engineering', Aout 1987.
- [Rak97] Z. Rakicevic and D. Jurukovski, '*Passive and active control systems : IZIIS experience*', 18<sup>th</sup> European Regional Earthquake Eng. Seminar, 1997
- [Ris93] D. Ristic, '*Control of structural behavior, part II : Passive structural control*', lectures for the international post graduate studies, University St. Cyril and Methodius, Skopje, 1993.
- [Rod87] J. Rodellar, A. H. Barbat and M. Martín-Sánchez, '*Predictive control of structures*', Journal of .eng. mech. ASCE, vol. 113, No. 6, pp. 797-812, 1987.
- [Rum86] D. E. Rumelhart, G. E. Hinton and R. J. Williams, '*Learning Internal Representations by error propagation*', In parrallel distributed processing, vol. 1, pp. 318-363, Cambridg MA. MIT Press, 1986.
- [Sag68] A.P.Sage, '*Optimum systems control*', Prentice Hall, 1968.
- [Sas89] S. H. Sastry and A. Isodori, '*Adaptive control of linearizable systems*', IEEE trans. On auto. Contr, vol. 34, No; 11 pp. 1123-1131, 1989.
- [Sca85] R.Scattolini & D.W.Clark, '*Multivariable model-following self tuning control with offset rejection*', INT. J. Control, Vol. 42,N°6, 1309-1322, 1985.
- [Set64] W.W.Seto, '*Theory and problem of mechanical vibrations*', Shaun's outline series. Macc Graw Hill, 1964.
- [Shi86] Shi-Jun Lang & Al, '*A multivariable generalized self tuning controller with daccoupling design*'. IEEE. Trans. Automatic Control, Vol. 31, N°5, pp 474-477, May 1986.
- [Slo91] J.J. E. Slotine and W. Li, '*Applied nonlinear control*', Prentice-Hall, Inc., Englewood Cliffs 1991.
- [Smi94] H. A. Smith, W. H. Wu and R. I. Borja, '*Structural control considering soil-structure interaction effects*', *Earthquake eng. struct. Dyn.*, vol. 23, pp. 609-626, 1994.
- [Sol61] V. F. J. Solis and R. J. P. Wets, '*Minimization by Random search techniques*', Mathematics of operations research, vol.6, No.1., pp. 19-29, february 1961.
- [Spa96] P. D. Spanos and B. A. Zeldin, '*Efficient iterative ARMA approximation of multivariate random processes for structural dynamics applications*', *Earthquake eng. struct. Dyn.*, vol.25, pp. 497-507, 1996.
- [Sto87] D.Stoten, '*Modelling and control of elastic joint robots*', ASME, Journal of dynamics system Measurement and control, vol 109, pp.310-319, 1987.
- [Sto89] D.P.Stoten & H.Benchoubane, '*Empirical studies of an MRAC algorithm with minimal controller*', Int. J. Cont. vol. 50, N°6, pp. 2249-2268, 1989.
- [Sto89] D.P.Stoten, '*Generalized manipulator dynamics with regard to model reference adaptive control*', Int. J. Cont., vol. 50, N°6, pp. 2249-2268, 1989.
- [Tha91] M.T.Tham & Al, '*Multivariable and multirate self tuning control:a distillation column case study*', IEE proc., pt. D, Vol. 138, N°1, January 1991.
- [Tho80] A.G.Thompson & Al, '*An optimal linear active suspension with finite road preview*', Society of Auto motive Engineer Inc, paper N°800520, 1980.
- [Toi84] H. T. Toivonen, '*Multivariable adaptive control*', Modeling, identification and Control, vol. 5, No. 1, pp. 19-45, 1984.
- [Vib87] C.Vibet, '*Robots: principes et controle*', Ed Ellipses, 1987.
- [Wel89] P.E.Wellstead & Al, '*Pole assignement self tuning regulator*', Proc IEE, Vol 126, N°8, pp. 781-789, August 1989.
- [Yan87] J. N. Yang, A. Akbapour and P. Ghaemmaghani, '*New optimal control algorithms for structural control*', J. eng. mech. ASCE, vol. 113, No. 9, pp. 1369-1386, 1987.
- [Yan90] J. N. Yang and A. Akbarpour, '*Effect of system uncertainty on control of sesmic excited buildings*', Journal of Engineering Mechanics, vol. 116, 102, pp.462-478, February 1990
- [Yed98] Y. M. Yeddou, '*Etude de synthèse sur les réseaux de neurones et leurs applications*', thèse de magister à l'ENP, DER de G. Electricité et Informatique, Automatique, 1998.
- [Zei79] O.C. Zienkiewicz, '*La methode des elements finis*', Mc Graw Hill Inc., 1979.
- [Zha89] J.E.Zhang & S.Lang, '*Adaptive control of a class of multivariable non-linear systems and convergence analysis*', IEEE, trans. auto. cont., Vol. 34, N°7, July 1989.
- [Zi91] Zi-Li Deng & Xian-Ri Huang, '*Multivariable decoupling pole assignement self tuning feed forward controller*', IEE, proc., pt. D, Vol 138, N°1, pp 85-88, january 1991.

---

## Annexe I

### Transformation de Denavit et Hartenberg

---

La complexité structurelle des systèmes mécaniques articulés, rend difficile leur analyse cinématique et dynamique. L'introduction d'un outil mathématique facilitant cet analyse s'impose. Cette difficulté réside dans la représentation analytique de la position spatiale de l'organe considéré, avec un minimum de paramètres.

Tout système mécanique (robot) est constitué d'une succession de liaisons et d'articulations rotatives, prismatiques ou à vis. La position de cette articulation est déterminée par la connaissance de l'orientation de son axe. Donc on cherche à représenter une droite dans l'espace avec un minimum de paramètres; ainsi le problème est posé. La droite  $(D)$  représentée par la figure A1.1, qui passe par le point  $P(x, y, z)$ , nécessite les paramètres  $\beta$  et  $\gamma$  pour compléter sa description [Den55].

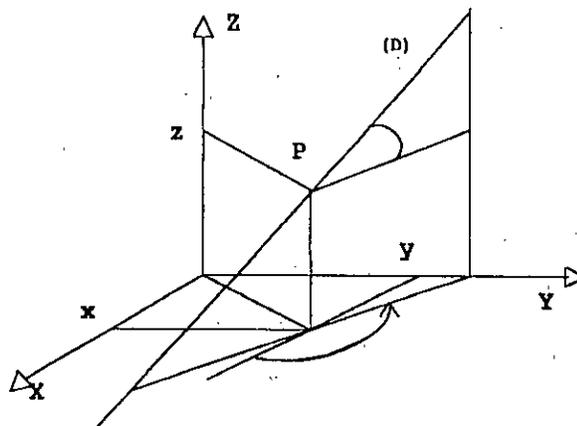


Figure A.1.1 Représentation de la droite dans l'espace.

Donc les cinq paramètres  $(x, y, z, \beta, \gamma)$  sont un nombre suffisant pour déterminer l'orientation de  $(D)$ . Mais la droite  $(D)$  peut être définie par ses équations:

$$x = az + b \text{ et } y = cz + d.$$

On remarque qu'on a besoin seulement de quatre paramètres  $(a, b, c, d)$ . La figure A1.2.a donne une description basée sur la détermination de l'unique perpendiculaire  $(O'H)$  entre  $(D)$  et  $(Oz)$ . Cette droite  $(D)$  est tangente à l'unique cylindre de rayon  $a$  et d'axe  $(Oz)$ .

Donc les quatre paramètres sont  $\alpha, \theta, a$  et  $d$ . Ayant déterminé ainsi les paramètres nécessaires, pour la détermination de l'orientation d'une droite (D) dans le repère  $R(x, y, z)$ , on peut alors élaborer la transformation  $T$ . Cette transformation permet le passage d'un repère  $R(O, x, y, z)$  à un autre repère  $R'(O', x', y', z')$ . Celle-ci peut être décomposée en quatre transformations (figure A1.2.b).

$${}_{R_i-1}T^{R_i} = Rot(z, \theta) Trans(0, 0, d) Trans(a, 0, 0) Rot(x, \alpha).$$

Où  $Rot(z, \theta)$  : Rotation autour de  $z$  d'un angle  $\theta$

$Rot(x, \alpha)$  : Rotation autour de  $x$  d'un angle  $\alpha$ .

$Trans(0, 0, d)$  : Translation d'une distance  $d$  le long de  $z$ .

$Trans(a, 0, 0)$  : Translation d'une distance  $a$  le long de  $x$ .

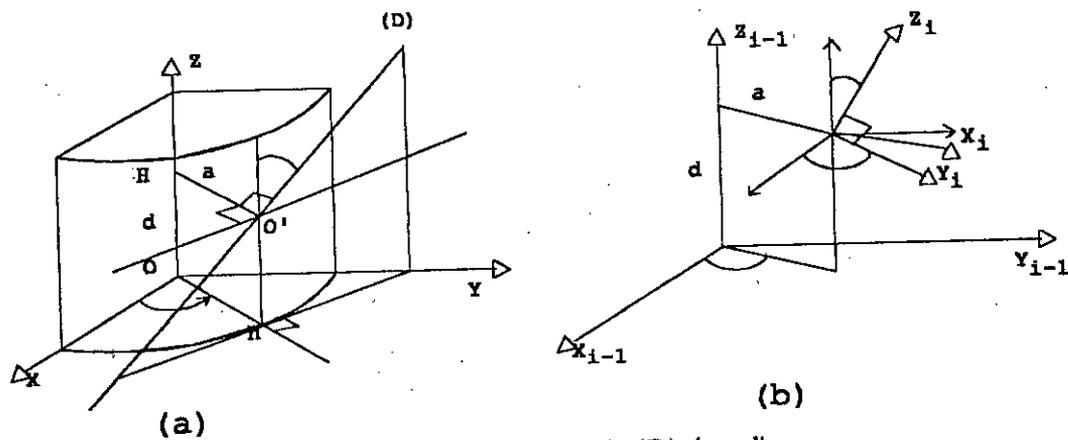


Figure A 1.2 Représentation de (D) dans l'espace.

Donc

$${}_{R}T^{R'} = \begin{bmatrix} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha & -S\alpha & 0 \\ 0 & S\alpha & C\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C\theta & -S\theta C\alpha & S\theta S\alpha & aC\theta \\ S\theta & C\theta C\alpha & -C\theta S\alpha & aS\theta \\ 0 & S\alpha & C\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

C'est la transformation de Denavit-Hartenberg. C'est cette transformation que nous utilisons comme matrice de passage du repère lié à l'articulation  $i$ , au repère lié à l'articulation  $i+1$ , dans un robot manipulateur.

### Implémentation de la transformation D-H.

Pour l'application de cette transformation, il faut d'abord fixer les différents repères dans chaque articulations d'une façon successive, en procédant de la façon suivante:

**Etape 1:** Numéroté chaque liaison et articulation, en commençant de la base, notée liaison 0 et l'effecteur (élément terminal) la liaison  $n$  (figure A 1.3). La liaison  $i$  se déplace par rapport à  $i-1$  autour (rotative) ou le long (prismatique) de celle-ci.

**Etape 2:** Etablir les repères de chaque articulation, en suivant les règles suivantes:

- L'axe  $Z_{i-1}$  est choisi le long de l'axe de l'articulation  $i$ .
- L'axe  $X_i$  est choisi perpendiculaire à  $Z_{i-1}$  dont le sens peut être choisi arbitrairement. ( $X_i = Z_{i-1} \wedge Z_i$ )
- L'axe  $Y_i$  est choisi de telle sorte à former un trièdre droit.

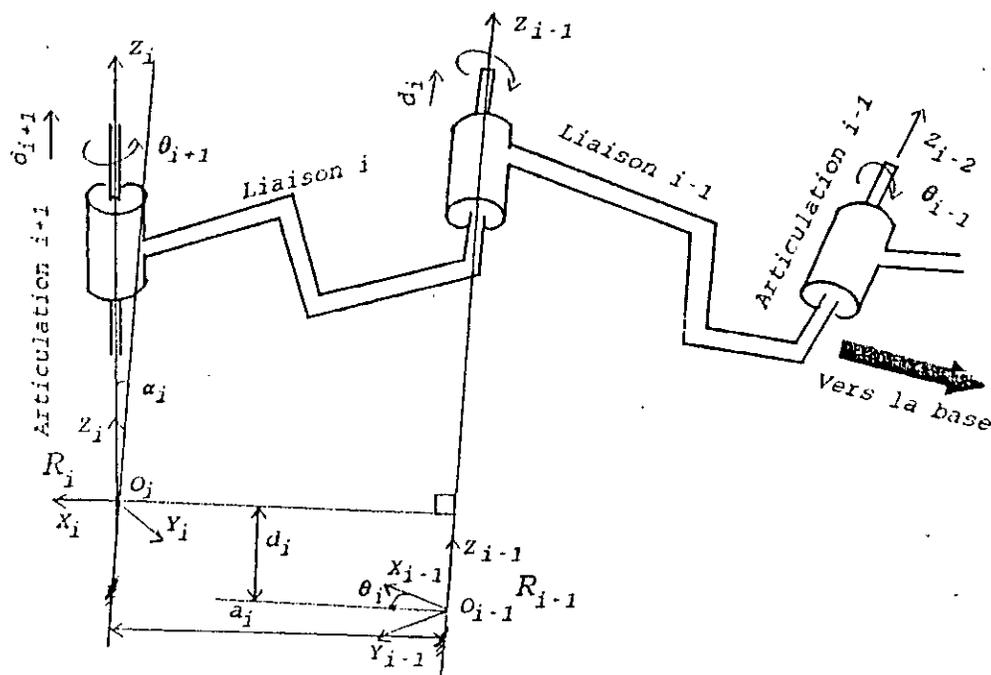
**Etape 3:** Définir les paramètres  $\theta_i, d_i, a_i$  et  $\alpha_i$ .

$\theta_i$ : angle entre  $X_{i-1}$  et  $X_i$ , obtenu par rotation de  $X_{i-1}$  vers  $X_i$  autour de  $Z_{i-1}$ .

$d_i$ : coordonnées de  $O_i$  dans  $R_{i-1}$ . (le long de  $Z_{i-1}$ ).

$a_i$ : distance entre  $Z_{i-1}$  et  $Z_i$ . (le long de  $X_i$ ).

$\alpha_i$ : angle entre  $Z_{i-1}$  et  $Z_i$ .



**Figure A1.3** Représentation des repères dans les articulations

## Annexe 2

### Calcul du modèle du Robot en utilisant L'Algorithme de Newton Euler

L'application de l'algorithme de NE au robot de classe 4, exige de définir certains paramètres.

$$\begin{aligned}
 {}_0R^1 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad {}_1R^2 = \begin{bmatrix} C_2 & 0 & -S_2 \\ S_2 & 1 & C_2 \\ 0 & -1 & 0 \end{bmatrix}; \quad {}_1R^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad {}_2R^1 = \begin{bmatrix} C_2 & S_2 & 0 \\ 0 & 0 & -1 \\ -S_2 & C_2 & 0 \end{bmatrix}. \\
 {}_2R^3 &= {}_0R^1; \quad {}_0R^2 = {}_1R^2; \quad {}_0R^3 = {}_1R^2; \quad {}_1R^3 = {}_1R^2; \quad {}_2R^0 = {}_2R^1; \quad {}_3R^0 = {}_2R^1; \quad {}_3R^1 = {}_2R^1. \\
 {}_1R^0 P_1^* &= [0 \ 0 \ d_1]; \quad {}_2R^0 P_2^* = [0 \ 0 \ 0]; \quad {}_3R^0 P_3^* = [0 \ 0 \ d_3]; \quad {}_2R^0 \bar{S}_2 = [0 \ 0 \ 0]; \\
 {}_1R^0 \bar{S}_1 &= [0 \ 0 \ \bar{z}_1]; \quad {}_3R^0 \bar{S}_3 = [0 \ 0 \ \bar{z}_3]; \quad \bar{z}_1 = -\frac{l_1}{2}; \quad \bar{z}_3 = -\frac{l_2}{2}.
 \end{aligned}$$

$${}_iR^0 I_i {}_0R^i = \begin{bmatrix} I_{xx_i} & 0 & 0 \\ 0 & I_{yy_i} & 0 \\ 0 & 0 & I_{zz_i} \end{bmatrix}; \quad i = 1, 3; \quad \text{avec } I_{xx_i} = \int (y_i^2 + z_i^2) dm_i. \\
 I_{yy_i} = \int (x_i^2 + z_i^2) dm_i. \\
 I_{zz_i} = \int (x_i^2 + y_i^2) dm_i.$$

Les produits d'inertie sont nulles car  $R_i(x_i, y_i, z_i)$  est parallèle aux axes principaux de la liaison considérée. Le calcul analytique du modèle dynamique du robot en utilisant ce formalisme est :

**1<sup>ère</sup> étape.** "forward equations".

pour  $i=1$  on a:

$${}_1R^0 \omega_1 = {}_1R^0 ({}_0R^0 \omega_0) = 0.$$

$${}_1R^0 \dot{\omega}_1 = {}_1R^0 ({}_0R^0 \dot{\omega}_0) = 0.$$

$$\begin{aligned}
 {}_1R^0 \dot{V}_1 &= {}_1R^0 (Z_0 \ddot{q}_1 + {}_0R^0 \dot{V}_0) + ({}_1R^0 \dot{\omega}_1) \wedge ({}_1R^0 P_1^*) + 2({}_1R^0 \omega_1) \wedge ({}_0R^0 Z_0 \dot{q}_1) \\
 &+ ({}_1R^0 \omega_1) \wedge [({}_1R^0 \omega_1) \wedge ({}_1R^0 P_1^*)] = [0 \ 0 \ \ddot{q}_1 + g]^T; \quad \text{car } \dot{V}_0 = [0 \ 0 \ g]^T.
 \end{aligned}$$

$${}_1R^0 \bar{a}_1 = ({}_1R^0 \dot{\omega}_1) \wedge ({}_1R^0 \bar{S}_1) + ({}_1R^0 \omega_1) \wedge [({}_1R^0 \omega_1) \wedge ({}_1R^0 \bar{S}_1)] + {}_1R^0 \dot{V}_1 = [0 \ 0 \ \ddot{q}_1 + g]^T.$$

pour  $i=2$  on a:

$${}_1R^0 \omega_2 = {}_2R^1 ({}_1R^0 \omega_1 + Z_0 \dot{q}_2) = [0 - \ddot{q}_2 \ 0].$$

$${}_2R^0 \dot{\omega}_2 = {}_2R^1 [{}_1R^0 \dot{\omega}_1 + Z_0 \ddot{q}_2 + ({}_1R^0 \omega_1) \wedge (Z_0 \dot{q}_2)] = [0 - \ddot{q}_2 \ 0]^T.$$

$${}_2R^0 \dot{V}_2 = ({}_2R^0 \dot{\omega}_2) \wedge ({}_2R^0 P_2^*) + ({}_2R^0 \omega_2) \wedge [({}_2R^0 \omega_2) \wedge ({}_2R^0 P_2^*)] + {}_2R^1 ({}_1R^0 \dot{V}_1) = [0 - \ddot{q}_1 - g \ 0]^T.$$

$${}_2R^0 \bar{a}_2 = ({}_2R^0 \dot{\omega}_2) \wedge ({}_2R^0 \bar{S}_2) + ({}_2R^0 \omega_2) \wedge [({}_2R^0 \omega_2) \wedge ({}_2R^0 \bar{S}_2)] + {}_2R^0 \dot{V}_2 = [0 - \ddot{q}_1 - g \ 0]^T.$$

pour  $i=3$  on a:

$${}_3R^0 \omega_3 = {}_3R^2 ({}_2R^0 \omega_2) = [0 - \dot{q}_2 \ 0]^T.$$

$${}_3R^0 \dot{\omega}_3 = {}_3R^2 ({}_2R^0 \dot{\omega}_2) = [0 - \ddot{q}_2 \ 0]^T.$$

$$\begin{aligned} & {}_3R^0 \dot{V}_3 = ({}_3R^0 \dot{\omega}_3) \wedge ({}_3R^0 P_3^*) + 2({}_3R^0 \omega_3) \wedge ({}_3R^0 Z_0 \dot{q}_3) \\ & + ({}_3R^0 \omega_3) \wedge [({}_3R^0 \omega_3) \wedge ({}_3R^0 P_3^*)] + {}_3R^2 ({}_2R^0 \dot{V}_2 + Z_0 \ddot{q}_3) \\ & = [-d_3 \ddot{q}_2 - 2 \dot{q}_2 \dot{q}_3 \quad -\ddot{q}_1 - g \quad \ddot{q}_3 - \dot{q}_2^2 d_3]^T. \end{aligned}$$

$$\begin{aligned} & {}_3R^0 \bar{a}_3 = ({}_3R^0 \dot{\omega}_3) \wedge ({}_3R^0 \bar{S}_3) + ({}_3R^0 \omega_3) \wedge [({}_3R^0 \omega_3) \wedge ({}_3R^0 \bar{S}_3)] + {}_3R^0 \dot{V}_3 \\ & = [-(\bar{z}_3 + d_3) \ddot{q}_2 - 2 \dot{q}_2 \dot{q}_3 \quad -\ddot{q}_1 - g \quad \ddot{q}_3 - (d_3 + \bar{z}_3) \dot{q}_2^2]^T. \end{aligned}$$

### 2<sup>ème</sup> Etape: "backward equations"

Dans cette étape, on a  $f_4 = 0$ ;  $\eta_4 = 0$  et  ${}_3R^4 = I$ .

Pour  $i=3$ :

$$\begin{aligned} & {}_3R^0 f_3 = {}_3R^4 ({}_4R^0 f_4) + m_3 {}_3R^0 \bar{a}_3 = m_3 [-(\bar{z}_3 + d_3) \ddot{q}_2 - 2 \dot{q}_2 \dot{q}_3 \quad -\ddot{q}_1 - g \quad \ddot{q}_3 - (d_3 + \bar{z}_3) \dot{q}_2^2]^T. \\ & -{}_3R^0 \eta_3 = {}_3R^4 [{}_4R^0 \eta_4 + ({}_4R^0 P_1^*) \wedge ({}_4R^0 f_4)] + ({}_3R^0 P_3^* + {}_3R^0 \bar{S}_3) \wedge (m_3 {}_3R^0 \bar{a}_3) \\ & + ({}_3R^0 I_3 \ 0 R^3) ({}_3R^0 \dot{\omega}_3) + ({}_3R^0 \omega_3) \wedge [({}_3R^0 I_3 \ 0 R^3) ({}_3R^0 \omega_3)] \\ & = m_3 [(g + \ddot{q}_1) (\bar{z}_3 + d_3) \quad -(d_3 + \bar{z}_3) \ddot{q}_2 - 2 \dot{q}_2 \dot{q}_3 \quad (d_3 + \bar{z}_3) - \beta_3 \ddot{q}_2 \ 0]^T. \end{aligned}$$

Pour  $i=2$ :

$$\begin{aligned} & {}_2R^0 f_2 = {}_2R^3 ({}_3R^0 f_3) + m_2 {}_2R^0 \bar{a}_2 \\ & = [m_3 [-(\bar{z}_3 + d_3) \ddot{q}_2 - 2 \dot{q}_2 \dot{q}_3] \quad -(m_2 + m_3) (\ddot{q}_1 + g) \quad m_3 [\ddot{q}_3 - (d_3 + \bar{z}_3) \dot{q}_2^2]]^T. \\ & -{}_2R^0 \eta_2 = {}_2R^3 ({}_3R^0 \eta_3 + ({}_3R^0 P_2^*) \wedge ({}_3R^0 f_3)) + ({}_2R^0 P_2^* + {}_2R^0 \bar{S}_2) \wedge ({}_2R^0 F_3) \\ & + ({}_2R^0 I_2 \ 0 R^2) ({}_2R^0 \dot{\omega}_2) + ({}_2R^0 \omega_2) \wedge (({}_2R^0 I_2 \ 0 R^2) ({}_2R^0 \omega_2)) \\ & = m_3 [(g + \ddot{q}_1) (\bar{z}_3 + d_3) \quad -(d_3 + \bar{z}_3) \ddot{q}_2 - 2 \dot{q}_2 \dot{q}_3 \quad (d_3 + \bar{z}_3) - (\beta_2 + \beta_3) \ddot{q}_2 \ 0]^T. \end{aligned}$$

Pour  $i=1$ :

$$\begin{aligned} & {}_1R^0 f_1 = {}_1R^2 ({}_2R^0 f_2) + m_1 {}_1R^0 \bar{a}_1 = [m_3 C_2 (-(\bar{z}_3 + d_3) \ddot{q}_2 - 2 \dot{q}_2 \dot{q}_3) - m_3 S_2 (\ddot{q}_3 - (d_3 + \bar{z}_3) \dot{q}_2^2) \\ & \quad m_3 S_2 (-(d_3 + \bar{z}_3) \dot{q}_2 \dot{q}_3) + m_3 C_2 (\ddot{q}_3 - (d_3 + \bar{z}_3) \dot{q}_2^2) (m_1 + m_2 + m_3) (\ddot{q}_1 + g)]^T \end{aligned}$$

${}_1R^0 \eta_1$ : à ne pas calculer car la liaison est translationnelle.

Donc

$$F_1 = ({}_1R^0 f_1)^T ({}_1R^0 Z_0) + f_1 \dot{q}_1 = (m_1 + m_2 + m_3)(\ddot{q}_1 + g) + f_1 \dot{q}_1.$$

$$\begin{aligned} \tau_2 &= ({}_2R^0 \eta_2)^T ({}_2R^1 Z_0) + f_2 \dot{q}_2 \\ &= (m_3 \bar{z}_3 d_3 + m_3 d_3 (d_3 + \bar{z}_3) + m_3 \bar{z}_3^2 + \beta_2 + \beta_3) \ddot{q}_2 + 2 m_3 (d_3 + \bar{z}_3) \dot{q}_2 \dot{q}_3 + f_2 \dot{q}_2. \end{aligned}$$

$$F_3 = ({}_3R^0 f_3)^T ({}_3R^2 Z_0) + f_3 \dot{q}_3 = -m_3 (\bar{z}_3 + d_3) \ddot{q}_2 + m_3 \dot{q}_2^2 + f_3 \dot{q}_3.$$

Ainsi en remplaçant les paramètres nécessaires:

$$F_1 = k_1 U_1; \tau_2 = k_2 U_2; F_3 = k_3 U_3;$$

$$m_2 \approx 0; \bar{z}_3 = \frac{l_2}{2}; \beta_3 = m_3 \frac{l_2^2}{12}; \beta_2 = 0;$$

On aboutit au modèle dynamique du robot trouvé dans le chapitre I en utilisant le formalisme d'Euler Lagrange.

## Annexe 3

### Schémas blocs des Modèles Linéaires

---

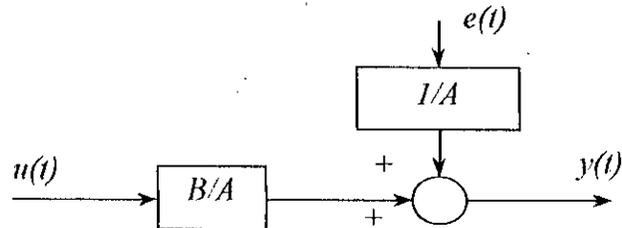


Figure A3.1 ARX Model (AutoRegressive with eXtra input)

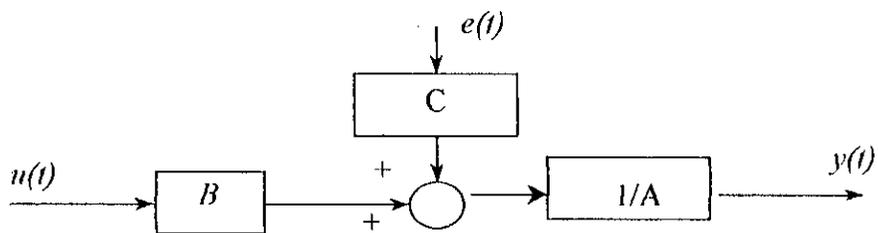


Figure A3.2 ARMAX Model (AutoRegressive with Moving Average and eXtra input)

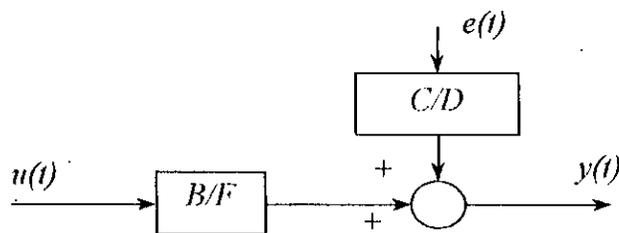


Figure A3.3 Box Jenkins Model input

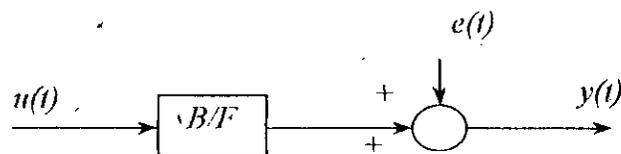


Figure A3.4 Output Error Model

## Annexe 4

### Différents Modèles de Séismes

---

L'accélération sismique du sol est alors modélisée comme étant un processus aléatoire non stationnaire uniformément modulé [CLO93][YAN87][LIN89] :

$$\ddot{x}_g(t) = \psi(t)\ddot{x}_s(t) \quad (\text{A4.1})$$

où

$\ddot{x}_s(t)$  est un processus stationnaire de moyenne nulle qui consiste en un bruit blanc filtré  
 $\psi(t)$  fonction enveloppe déterministe non négative pour convertir le processus stationnaire  $\ddot{x}_s(t)$  en un processus non stationnaire  $\ddot{x}_g(t)$ . Cette fonction reflète la variation de l'intensité du séisme en fonction de la distance entre la source du séisme et le site ou l'endroit considéré. Une fonction très particulièrement utilisée est donnée par la formule suivante :

$$\psi(t) = \begin{cases} 0 & \text{si } t < 0 \\ \left(\frac{t}{t_1}\right)^2 & \text{si } 0 \leq t \leq t_1 \\ 1 & \text{si } t_1 \leq t \leq t_2 \\ \exp[-\alpha(t-t_2)] & \text{si } t \geq t_2 \end{cases} \quad (\text{A4.2})$$

où  $t_1$ ,  $t_2$  et  $\alpha$  sont des paramètres choisis convenablement pour refléter la forme et la durée de l'accélération sismique du sol.

Deux modèles de séismes sont utilisés dans cette thèse :

#### *Modèle de Kanai-Tajimi*

Pour le modèle de Kanai-Tajimi, le processus stationnaire  $\ddot{x}_s(t)$  est défini par la densité spectrale dite de Kanai-Tajimi donnée [YAN87][LIN87] par :

$$\phi_g(\omega) = \left[ \frac{1 + 4\xi_g^2 \left(\frac{\omega}{\omega_g}\right)^2}{\left(1 - \left(\frac{\omega}{\omega_g}\right)^2\right)^2 + 4\xi_g^2 \left(\frac{\omega}{\omega_g}\right)^2} \right] S_0^2 \quad (\text{A4.3})$$

où  $\xi_g$  et  $\omega_g$  sont les paramètres du filtre,  $S_0$  est la densité spectrale constante du bruit blanc.

#### Modèle de Clough-Penzien

On peut facilement constater que les spectres du déplacement et de la vitesse, obtenus à partir du spectre d'accélération de Kanai-Tajimi décrit par l'équation (A3.3), ont de fortes singularités à la fréquence 0. Ces singularités résulteront en des valeurs excessives de la vitesse et du déplacement autour de cette fréquence. Pour palier à ce problème, Clough et Penzien [CLO93][LIN89] ont proposé d'utiliser un filtre passe-haut tel que la densité spectrale du signal  $\ddot{x}_s(t)$  devient :

$$\phi_c(\omega) = \left[ \frac{1 + 4\xi_g^2 \left(\frac{\omega}{\omega_g}\right)^2}{\left(1 - \left(\frac{\omega}{\omega_g}\right)^2\right)^2 + 4\xi_g^2 \left(\frac{\omega}{\omega_g}\right)^2} \right] \left[ \frac{\left(\frac{\omega}{\omega_c}\right)^2}{\left(1 - \left(\frac{\omega}{\omega_c}\right)^2\right)^2 + 4\xi_c^2 \left(\frac{\omega}{\omega_c}\right)^2} \right] S_0^2 \quad (\text{A4.4})$$

où  $\xi_g, \omega_g, \xi_c$  et  $\omega_c$  sont les paramètres du filtre.

## Annexe 5

### Calcul du prédicteur optimal monovarié

On cherche le prédicteur optimal au sens des moindres carrés de  $\psi(t+d+1) = Py(t+d+1)$ . Ce prédicteur, noté  $\psi^*(t+d+1)$ , minimise le critère suivant

$$J_1 = E[\psi(t+d+1) - \psi^*(t+d+1)]^2 \quad (\text{A5.1})$$

En utilisant l'équation (III.15) on a

$$\psi(t+d+1) = \frac{PB^*}{A}u(t) + \frac{PC}{A}e(t+d+1) \quad (\text{A5.2})$$

où  $B^*(q^{-1}) = qB(q^{-1})$ .

Dans l'expression de  $\psi(t+d+1)$  on doit séparer les informations futures des informations disponibles. La séparation de ces informations revient à faire la division euclidienne de  $P_N C$  par  $P_D A$

$$\frac{P_N C}{P_D A} = S' + q^{-(d+1)} \frac{R}{P_D A} \quad (\text{A5.3})$$

où

$$S'(q^{-1}) = s'_0 + s'_1 q^{-1} + \dots + s'_d q^{-d}$$

$$R(q^{-1}) = r_0 + r_1 q^{-1} + \dots + r_{n_r} q^{-n_r}, \quad n_r = n + n_d - 1, \quad \text{où } n_d \text{ est le degré du polynôme } P_D$$

Ceci conduit à l'équation Diophantine

$$AP_D S' + q^{-(d+1)} R = P_N C \quad (\text{A5.4})$$

En tenant compte de l'équation Diophantine, l'équation (A5.2) devient

$$\psi(t+d+1) = \frac{P_N B^*}{P_D A} u(t) + \frac{R}{P_D A} e(t) + S' e(t+d+1) \quad (\text{A5.5})$$

En substituant l'équation (A5.5) dans l'équation (A5.1)

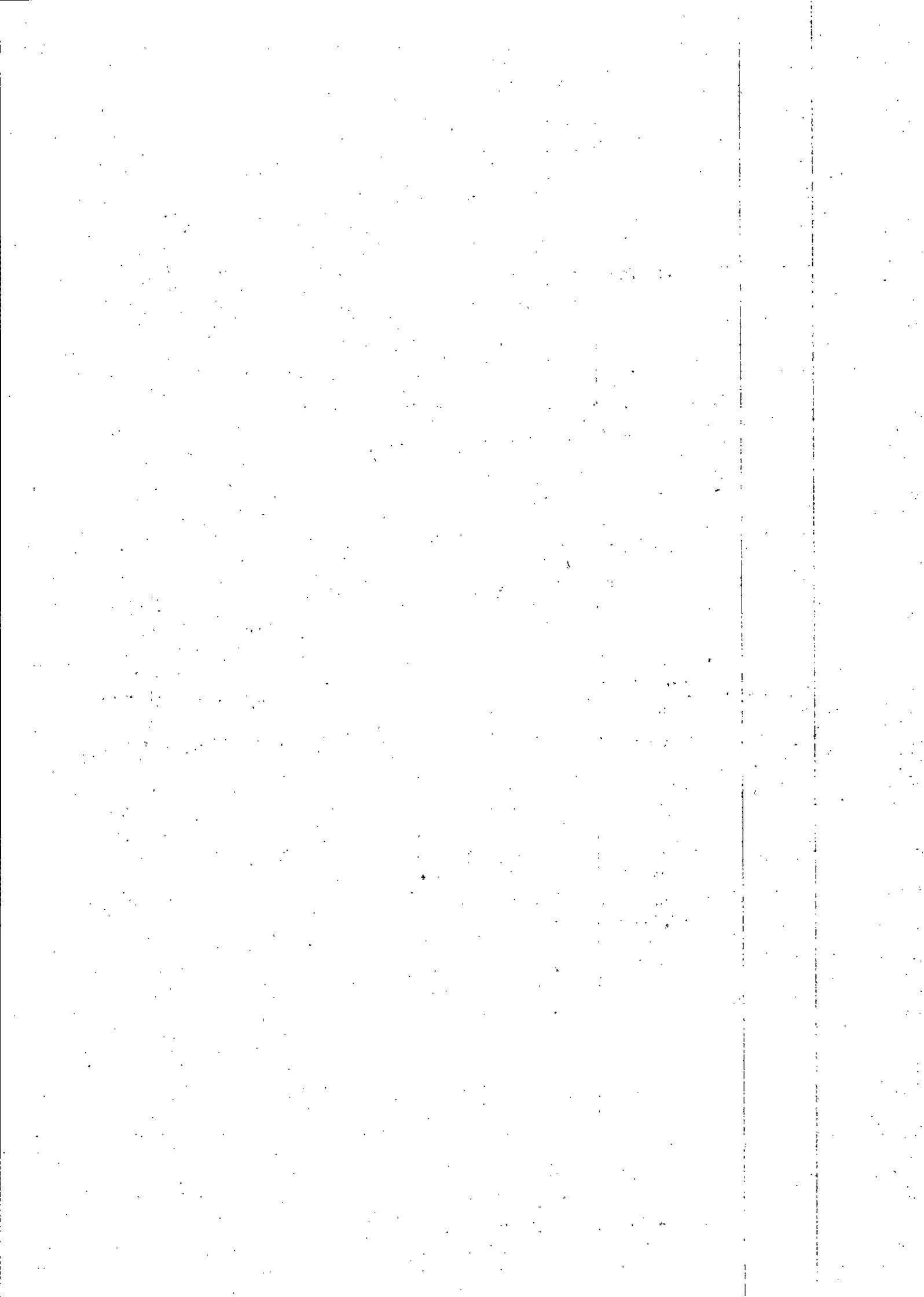
$$J_1 = E \left[ \frac{P_N B^*}{P_D A} u(t) + \frac{R}{P_D A} e(t) - \psi^*(t+d+1) \right]^2 + \left( \sum_{i=0}^d s_i'^2 \right) \sigma^2 + 2E \left[ \frac{P_N B^* S'}{P_D A} u(t) e(t+d+1) + \frac{RS'}{P_D A} e(t) e(t+d+1) - S'' \psi^*(t+d+1) e(t+d+1) \right] \quad (\text{A5.6})$$

En sachant que  $e(t)$  est un bruit blanc, qui n'est corrélé qu'avec lui-même, l'équation (A5.6) devient

$$J_1 = E \left[ \frac{P_N B^*}{P_D A} u(t) + \frac{R}{P_D A} e(t) - \psi^*(t+d+1) \right]^2 + \left( \sum_{i=0}^d s_i'^2 \right) \sigma^2 \quad (\text{A5.7})$$

$J_1$  est minimal si le premier terme du second membre de l'équation (A5.7) est nul, ainsi

$$\psi^*(t+d+1) = \frac{P_N B^*}{P_D A} u(t) + \frac{R}{P_D A} e(t) \quad (\text{A5.8})$$



## Annexe 6

### Calcul de la loi de commande à variance minimale Généralisée monovariante

En utilisant l'équation (A5.1) le critère de l'équation (III.23) devient

$$J = E \left[ \left( \psi^*(t+d+1) + S'e(t+d+1) - R_w w(t+d+1) \right)^2 + (Q'u(t))^2 \right] \quad (\text{A6.1})$$

ou encore

$$J = E \left[ \left( \psi^*(t+d+1) - R_w w(t+d+1) \right)^2 + (Q'u(t))^2 \right] + \left( \sum_{i=0}^d s_i^2 \right) \sigma^2 \quad (\text{A6.2})$$

la minimisation de  $J$  par rapport à  $u(t)$  revient à écrire

$$\frac{\partial J}{\partial u(t)} = 2E \left[ \left( \psi^*(t+d+1) - R_w w(t+d+1) \right) \frac{\partial \psi^*(t+d+1)}{\partial u(t)} + (Q'u(t)) \frac{\partial Q'u(t)}{\partial u(t)} \right] = 0 \quad (\text{A6.3})$$

mais on a  $\frac{\partial \psi^*(t+d+1)}{\partial u(t)} = \frac{p_{N_0} b_1}{p_{D_0}}$  et  $\frac{\partial Q'u(t)}{\partial u(t)} = \frac{q'_{N_0}}{q'_{D_0}}$  où  $q'_{N_0}$  et  $q'_{D_0}$  sont les premiers coefficients du numérateur  $Q'_N(q^{-1})$  et du dénominateur  $Q'_D(q^{-1})$  du polynôme  $Q(q^{-1})$  respectivement.

Donc l'équation (A5.3) devient

$$\psi^*(t+d+1) - R_w w(t+d+1) + Qu(t) = 0 \quad (\text{A6.4})$$

où  $Q(q^{-1}) = \frac{q'_{N_0} p_{D_0}}{q'_{D_0} p_{N_0} b_1} Q'(q^{-1})$ .

en substituant l'équation (A4.6) dans l'équation (A5.4), on déduit la loi de commande à variance minimale généralisée suivante

$$u(t) = \frac{P_D C R_w w(t+d+1) - R y(t)}{P_D (S + QC)} \quad (\text{A6.5})$$

où  $S = S'B^*$ .

## Annexe 7

### Calcul du prédicteur optimal multivariable

Introduisant d'abord l'équation polynomiale matricielle suivante

$$\bar{C}(q^{-1}) = \underline{A}(q^{-1})\underline{F}(q^{-1}) + q^{-(d+1)}\underline{G}(q^{-1}) \quad (\text{A7.1})$$

où

$$\underline{F}(q^{-1}) = F_0 + F_1 q^{-1} + \dots + F_d q^{-d}$$

$$\underline{G}(q^{-1}) = G_0 + G_1 q^{-1} + \dots + G_{ng} q^{-ng}, \quad ng = \max[l + np - d - 1, n - 1] \text{ et } np = \deg[\underline{P}(q^{-1})]$$

et

$$\bar{C}(q^{-1}) = \bar{P}(q^{-1})\underline{C}(q^{-1}) \quad (\text{A7.2})$$

$$\bar{A}(q^{-1})\underline{P}(q^{-1}) = \bar{P}(q^{-1})\underline{A}(q^{-1}) \quad (\text{A7.3})$$

avec  $\det \bar{A}(q^{-1}) = \det \underline{A}(q^{-1})$  et  $\bar{A}(0) = I$ .

il existe des matrices  $\bar{F}$  and  $\bar{G}$ , non uniques, vérifiant

$$\bar{F}(q^{-1})\underline{G}(q^{-1}) = \bar{G}(q^{-1})\underline{F}(q^{-1}) \quad (\text{A7.4})$$

où  $\det \bar{F}(q^{-1}) = \det \underline{F}(q^{-1})$

Introduisant la matrice polynomiale  $\tilde{C}$

$$\tilde{C}(q^{-1}) = \bar{F}(q^{-1})\underline{A}(q^{-1}) + q^{-(d+1)}\tilde{G}(q^{-1}) \quad (\text{A7.5})$$

A partir des équations (A7.1) à (A7.5), on démontre que

$$\tilde{C}(q^{-1})\underline{P}(q^{-1}) = \tilde{F}(q^{-1})\underline{A}(q^{-1}) + q^{-(d+1)}\tilde{G}(q^{-1}) \quad (\text{A7.6})$$

$$\tilde{C}(q^{-1})\underline{F}(q^{-1}) = \tilde{F}(q^{-1})\underline{C}(q^{-1}) \quad (\text{A7.7})$$

avec  $\tilde{F}(q^{-1}) = \bar{F}(q^{-1})\bar{P}(q^{-1})$  et  $\tilde{G}(q^{-1}) = \bar{G}(q^{-1})\underline{P}(q^{-1})$

Equation (III.23) peut être écrite sous la forme

$$\underline{A}(q^{-1})\underline{y}(t+d+1) = \underline{B}^*(q^{-1})\underline{u}(t) + \underline{C}(q^{-1})\underline{e}(t+d+1) \quad (\text{A7.8})$$

En multipliant l'équation (A7.8) par  $\tilde{F}$  et en utilisant les équations (A7.6) et (A7.7), on obtient

$$\varphi(t+d+1) = \underline{P}(q^{-1})\underline{y}(t+d+1) = \tilde{C}^{-1}(q^{-1})\left[\tilde{G}(q^{-1})\underline{y}(t) + \tilde{F}(q^{-1})\underline{B}^*(q^{-1})\underline{u}(t)\right] + \underline{F}(q^{-1})\underline{e}(t+d+1) \quad (\text{A7.9})$$

On remarque que  $\underline{F}(q^{-1})\underline{e}(t+d+1) = F_0 \underline{e}(t+d+1) + F_1 \underline{e}(t+d) + \dots + F_d \underline{e}(t+1)$  est une information future. Ainsi, la prédicteur optimal de  $\varphi(t+d+1)$  est donné par :

$$\varphi^*(t+d+1) = \tilde{C}^{-1}(q^{-1})\left[\tilde{G}(q^{-1})\underline{y}(t) + \tilde{F}(q^{-1})\underline{B}^*(q^{-1})\underline{u}(t)\right] \quad (\text{A7.10})$$

## Annexe 8

### Calcul de la loi de commande à variance minimale Généralisée multivariable

En substituant l'équation (A7.9) dans le critère (III.29) on obtient

$$J = E \left[ \left\| \varphi^*(t+d+1) - \underline{R}_w(q^{-1})\underline{w}(t+d+1) + \underline{F}(q^{-1})\underline{e}(t+d+1) \right\|^2 + \left\| \underline{Q}'(q^{-1})\underline{u}(t) \right\|^2 \right] \quad (\text{A8.1})$$

Le terme  $\underline{F}(q^{-1})\underline{e}(t+d+1)$  n'est pas corrélé avec  $\underline{u}(t-i)$ ,  $\underline{w}(t-i)$  et  $\underline{y}(t-i)$  pour  $0 \leq i \leq t$ .

Ainsi, le critère  $J$  devient

$$J = E \left[ \left\| \varphi^*(t+d+1) - \underline{R}_w(q^{-1})\underline{w}(t+d+1) \right\|^2 + \left\| \underline{Q}'(q^{-1})\underline{u}(t) \right\|^2 \right] + E \left[ \left\| \underline{F}(q^{-1})\underline{e}(t+d+1) \right\|^2 \right] \quad (\text{A8.2})$$

Le minimum de  $J$  est obtenu en annulant le gradient de  $J$  par rapport à  $\underline{u}(t)$

$$\frac{\partial J}{\partial \underline{u}(t)} = 2E \left[ \left( \frac{\partial \varphi^*(t+d+1)}{\partial \underline{u}(t)} \right)^T \left( \varphi^*(t+d+1) - \underline{R}_w(q^{-1})\underline{w}(t+d+1) \right) + \underline{Q}'(0)^T \underline{Q}'(q^{-1})\underline{u}(t) \right] = 0 \quad (\text{A8.3})$$

ou

$$\varphi^*(t+d+1) - \underline{R}_w(q^{-1})\underline{w}(t+d+1) + \underline{Q}(q^{-1})\underline{u}(t) = 0 \quad (\text{A8.4})$$

avec  $\underline{Q}(q^{-1}) = \left[ (P_0 B_1)^T \right]^{-1} \underline{Q}'(0)^T \underline{Q}'(q^{-1})$

Par substitution de l'équation (A7.10) dans l'équation (A8.4), on déduit la loi de commande à GMV multivariable

$$\underline{H}(q^{-1})\underline{u}(t) = \tilde{\underline{C}}(q^{-1})\underline{R}_w(q^{-1})\underline{w}(t+d+1) - \tilde{\underline{G}}(q^{-1})\underline{y}(t) \quad (\text{A8.5})$$

avec  $\underline{H}(q^{-1}) = \tilde{\underline{F}}(q^{-1})\underline{B}^*(q^{-1}) + \tilde{\underline{C}}(q^{-1})\underline{Q}(q^{-1})$

## ملخص

يهدف هذا العمل إلى تطبيق التحكم الآلي والشبكات العصبية في ميدان البناءات الديناميكية. قمنا بعرض طريقة إيجاد النموذج الديناميكي لكلا بنائات الهندسة المدنية والبناءات المفصلية. قدمنا فيما بعد طريقة إيجاد نموذج جديد لبناءات الهندسة المدنية تحت الهزات الأرضية (نموذج ARMAX) بالأحد بعين الاعتبار الضفة الاحتمالية للحملة. ثم عرضنا طريقة التحكم بالانحراف الأدنى المعم لكلا الحالتين. مقارنة بين مختلف التحكمات الخطية قد أعطيت. وبالتالي قدمنا استراتيجيات غير خطية. في حالة البناءات الهندسة المدنية، قدمنا استراتيجية المنضم المرمج باستعمال الشبكات العصبية. نتائج المحاكات قد أثبتت نجاعة طرق التحكم المطورة.

كلمات مفتاحية : الانحراف الأدنى المعم، بناءات مفصلية، شبكات عصبية، التحكم الأعظمي، التحكم الغير الخطي، نموذج ARMAX، الهندسة المدنية.

## Abstract

In this work, the role and the use of automatic control theory and artificial neural networks in structural dynamic is presented. Firstly, the dynamical models of the structures are derived in both cases of structures in civil engineering and articulated structures. Secondly, a new discrete model is developed for describing the stochastic behavior of structures in civil engineering under seismic ground motion (Auto Regressive Moving Average eXogen (ARMAX) model). Then we develop the Generalized Minimum Variance (GMV) Algorithm for the two cases. A comparative study between different linear control algorithms is shown. Therefore, we develop an hybrid and multirate control laws in the nonlinear case. Finally the gain-scheduling algorithm is established for civil engineering using neural networks. To demonstrate the effectiveness of these control techniques, simulation tests are performed.

**Key words:** Generalized Minimum Variance, Articulated structures, neural networks, optimal control, nonlinear control, ARMAX model, Civil engineering.

## Resumé

Dans ce travail, nous nous intéressons à l'application de l'automatique et des réseaux de neurones en dynamique des structures. Nous avons procédé à l'élaboration des modèles de connaissance des structures articulés ainsi que les structures en génie civil. Ensuite nous avons contribué à élaborer un nouveau modèle discret pour les structures en génie civil soumises à des excitations sismiques, en prenant en considération l'aspect stochastique du problème (modèle ARMAX). Nous avons développé la commande à variance minimale généralisée (GMV) pour les deux types de structures (robot et bâtiment). Une analyse comparative entre les différents types de commande linéaire est donnée. Par ailleurs nous avons synthétisé des commandes non linéaires hybrides mixtes et multirates. Dans le cas des structures en génie civil, nous avons contribué à la synthèse des stratégies de commande à gain programmé à base de réseaux de neurones. Les résultats de simulation ont montré la robustesse de ces algorithmes à maintenir les performances désirées.

**Mots clés:** Variance minimale généralisée, structures, articulées, réseaux de neurones, commande optimale, commande non linéaire, modèle ARMAX, génie civil.