ECOLE NATIONALE POLYTECHNIQUE
Département de Génie Electrique

Thèse

Présentée par:

GUENFAF LAKHDAR
Ingenieur d'état en Automatique de l'ENP

Pour obtenir le titre de

MAGISTER
en AUTOMATIQUE

Sujet:

ETUDE DE DIFFERENTES STRATEGIES DE COMMANDE ADAPTATIVE

"Application à un robot manipulateur"

Soutenue publiquement le 16/01/1995 devant le jury composé de:

B.DERRAS	Président
R.ILLOUL	Rapporteurs
M.S.BOUCHERIT	
F.BOUDJEMA	Examinateurs
M.S.AITCHEKH	
M.C.SOUAMI	
H.CHEKIREB	Invité

Laboratoire d'AUTOMATIQUE
Ecole Nationale Polytechnique
10, Avenue Pasteur El Harrach-Alger
tel. (231)(2) 52 53 01/03 Fax. (213)(2) 52 29 73

بسم الله الرحمن الرحيم

وما أوتيتم من العلم إلا قليلا

صدق الله العظيم

A ma mère
à qui je dois beaucoup,

A mon père
qu'il trouve l'aboutissement de ses sacrifices,

A mes frères et soeurs,
en particulier Mohamed,

A Bali Noureddine, Bahia sa femme, Amina sa fille,
pour toutes les choses que nous avons partagées,
et qui m'ont soutenu dans ce chemin long et difficile

A tous mes amis (es),
en particulier Med Hadid,
pour leur soutien et leur présence.

AVANT-PROPOS

Ce travail a été effectué au Laboratoire d'Automatique, et au Laboratoire des Systèmes Intelligents Autonomes (LARSIA) de l'Ecole Nationale Polytechnique.

J'exprime ma profonde reconnaissance à Mrs M.S.Boucherit et R.Illoul qui en dirigeant ces travaux, m'ont fait profiter de l'étendue de leurs connaissances, de leurs conseils, de leurs aides et 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'il porte à mes travaux.

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

Je remercie Monsieur M.C. Souami, d'avoir accepté de juger ce travail.

Je remercie également Monsieur M.S. Ait Chikh, pour avoir accepter de se pencher sur ce travail, faire partie du jury et d'examiner la thèse.

J'adresse mes sincères remerciements à Monsieur F. Boudjema, pour avoir accepté de participer à ce jury et d'avoir examiné ce travail.

Je tiens à exprimer ma profonde reconnaissance à Monsieur H. CHEKIREB, d'avoir accepter l'invitation.

Mes remerciements s'adressent également à: Lounici Abd el hakim, Madjouj Nawel, El Hadi (du centre de calcul), Bali: Chava, Tassadit, Tati, Larbi, qui ont contribué à me faciliter l'acquisition du matériel informatique.

Je tiens à adresser mes remerciements à: H.Tayebi, M.Bouri, F. Dari, F. Dehimi, R. Hadjar, A. Laroui, M.Djemai, B.Kacha, M.Taghi, Pour toute la bibliographie qui m'est parvenue pour réaliser ce travail.

A Farida Bali et son marie Fethi, Fouzia Djallali, je voudrai leurs exprimer ma profonde reconnaissance.

Mes sincères remerciements iront à tous ceux qui ont contribué à la réalisation matérielle de ce mémoire notamment ceux du service documentation à savoir: Amara, Ourari, Salah, Karim, Krimo, Abdelkrim, Mourad, Hafida, Chrifa, Ouahiba, Naima, Farida et Ghania sans oublier celles des périodiques.

Je remercie vivement Farid Hallet, Reda Yedou, Samir Ghanem, Azzedine Hasbellaoui, mon oncle Messaoud Tachoua, les frères Mitiche H'cinet et Houcine, Mr Messaoud Hadid et sa femme, Mustapha mon ami et Biga M'hamed pour leur soutient moral et leurs encouragements.

Je n'aurai garde d'oublier dans mes remerciements Noureddine Berrou ainsi que sa femme et mon frère Abdelhakim pour leurs précieuses aides.

Je souhaite remercier Madame Sebaibi pour ses qualités humaines.

Je voudrai terminer en saluant la promotion de l'automatique de Juin 92 de l'ENP d'Alger et celle du Magistère 93.

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

SOMMAIRE

Introduction générale

Chapitre I
Modélisation

Chapitre II
Commande adaptative auto-ajustable implicite et explicite. (STR Self Tuning Regulators)

Chapitre III
Commande non linéaire hybride auto-ajustable.

Chapitre IV
Commande adaptative à perturbation. (Adaptive Perturbation Control)

Chapitre V
Commande linéarisante adaptative.

Chapitre VI
Systèmes adaptatifs à modèle de référence (MRAS)

INTRODUCTION GENERALE

"La pensée ne commence qu'avec le doute".

ROGER MARTIN AUGARD

INTRODUCTION GENERALE

Au cours des vingts dernières années, des contributions fondamentales en commande adaptative, aussi bien théoriques que pratiques, ont apporté des éléments nécessaires à une meilleure compréhension des systèmes adaptatifs [24][26][91][93][98][36]. Le principal objet de la commande adaptative est la synthèse de lois d'adaptation, pour l'ajustement automatique en ligne et en temps réel des régulateurs des boucles de commande, afin de réaliser ou de maintenir un certain niveau de performances quand les paramètres du procédé à commander sont inconnus ou mal connus et/ou susceptibles de varier dans le temps [24].
L'approche consiste à combiner une estimation paramétrique et une méthode de synthèse, en appliquant le principe dit "d'équivalence certaine" [13]. Cet estimateur, qui représente le coeur même de la commande adaptative, nécessite une certaine connaissance de la structure du procédé, en fonction de l'objectif de commande désirée. Cette partie est réalisée en deux étape. Une étape qualitative qui consiste à choisir une structure du modèle de commande entrée-sortie du procédé et une étape quantitative qui consiste à choisir l'estimation des paramètres [27].

La commande des robots manipulateurs constitue à l'heure actuelle l'une des principales préoccupation des recherches en robotique [10][91][93]. 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 modèle cinématique et géométrique direct et inverse du robot [5][6]. La même transformation offre une souplesse dans le calcul du modèle dynamique direct et inverse [1][3][8][11].

Le modèle dynamique des robots manipulateurs est fortement non linéaire. Le choix d'un modèle de représentation aux différences multivarible découplé, permet l'implémentation des algorithmes monovariables auto-ajustables pour la commande des robots [22][76][78][80].
Les régulateurs auto-ajustables (Self Tuning Regulators STR) sont basés sur l'estimation en temps réel des paramètres du système (modélisé sous forme d'équations aux différences) ou ceux du régulateur. 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 indirecte, 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 [14][16][20][24][25].

L'existence des termes non linéaires, dans le modèle du robot, nécessite leurs compensation, pour obtenir des modèles moins complexes facilement commandable. L'utilisation d'un algorithme auto-ajustable, pour la commande du modèle obtenu, aboutit à la commande non linéaire hybride auto-ajustable, constituée d'une boucle de compensation (continu) et d'une commande auto-ajustable (discrète) [1][2][81][82][84].

L'utilisation d'un modèle de représentation aux différences linéaire, pour le calcul de la commande à STR, nécessite un comportement linéaire du modèle du robot. Ce comportement peut être obtenu par linéarisation du modèle nonlinéaire du robot, autour d'une trajectoire nominale (variable dans le temps). On obtient ainsi le modèle à perturbation avec lequel on synthétise la commande [1][44]. 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 [1][22][74].

La connaissance de la structure du modèle décrivant le comportement dynamique du robot, offre la possibilité du choix d'une paramétrisation [91]. 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 réel du robot. La synthèse d'une loi de commande adaptative, basée sur cette paramétrisation, aboutit à la commande linéarisante adaptative [89][93][94]. La stabilité d'un tel algorithme est étudiée en utilisant la théorie de stabilité de Lyapounov et d'hyperstabilité de Popov [88][96].

L'analyse de la convergence et la synthèse des lois d'adaptation peuvent être élaborées en utilisant les techniques adaptatives à modèle de référence [88][98].
Les systèmes adaptatifs à modèle de référence sont utilisés dans de larges domaines, pour résoudre une variétés importantes de problèmes rencontrés en commande, identification et estimation d'état [97][101][106][107].
La synthèse des lois d'adaptation dans la commande adaptative à modèle de référence, est faite en utilisant la théorie d'hyperstabilité de Popov, pour assurer une convergence asymptotique de l'erreur [88][100][101][102]. Plusieurs version de cette commande en découlent (MRAC: Model Reference Adaptive Control, MCS: Minimum Controller Synthesis, DMRAC: Discrete MRAC) [101][106][108][109][113][114].

Dans ce travail on s'intéresse à l'étude de différentes stratégies de commande adaptative. L'application, par simulation, de chaque stratégie à un robot de classe quatre (4), permet une compréhension et une analyse approfondie de la commande utilisée.
Pendant tout notre travail, on a 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 bien valider les différents algorithmes utilisés. On trouve dans chaque chapitre une introduction, une conclusion et des résultats de simulation.

Dans le chapitre un, on s'intéresse à l'élaboration des différents modèles du robot utilisé. L'étude cinématique en utilisant la transformation de Denavit-Hartenberg, permet l'obtention des modèles cinématiques (variationnelles) et géométriques directs et inverses.
Les deux modèles dynamiques, calculés en utilisant les formalismes de Newton-Euler et celui d'Euler-Lagrange, sont exposés.

Le chapitre deux, présente les différentes lois de commande auto-ajustables. Dans un premier lieu, on présente l'algorithme à variance minimale explicite et implicite. L'algorithme à variance minimale généralisée est exposé dans la section (II.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). La contribution de notre travail consiste à l'application de l'algorithme mixte et multirate dans le cas de la robotique.

La commande non linéaire hybride est développée dans le chapitre trois. On présente 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 le géométrie différentielle. L'application du STR (Self Tuning Regulator) au modèle obtenu est notre propre contribution dont on présente les différents résultats à la fin de ce chapitre.

Le chapitre quatre, est consacré à la commande adaptative à perturbation. On développe en premier lieu le modèle à perturbation. En second lieu, les différents algorithmes auto-ajustables sont présentés. Enfin le rôle de la commande à perturbation calculée à partir du modèle à perturbation est bien démontré par les différents résultats de simulation.

Dans le cinquième chapitre, nous présentons la commande linéarisante. Une partie de ce chapitre est consacrée à la commande linéarisante à paramètres fixes. La version prédictive de cette commande est développée dans la seconde partie. L'application de ces différentes lois à notre robot manipulateur (choix de la paramétrisation et des Matrices de Pondération) est l'objectif de ce chapitre.

Dans le dernier chapitre, nous exposons les systèmes adaptatifs à modèle de référence dont on consacre la première partie. La commande linéaire par poursuite d'un modèle est développée dans la seconde partie. La troisième partie concerne la commande adaptative à modèle de référence (MRAC). Le développement de l'algorithme MCS ainsi que sa condition de stabilité asymptotique sont présentés dans la quatrième partie. Dans la cinquième partie, on présente la version discrète du MRAC (DMRAC).

Enfin, on termine notre mémoire par une conclusion générale avec quelques perspectives.

Chapitre I

MODELISATION

"Une loi est un modèle qui n'est plus (ou pas encore!) contesté"
"un modèle devient une loi ... , ou sombre dans l'oubli!".

JEAN LEMAITRE

Chapitre I

MODELISATION

Introduction.

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 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 entre ce système et son environnement extérieur. Lorsque les relations sus-citées 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.

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'Alambert (Algorithme de NE).

I.1 Présentation du robot.

Parmi les différentes définition d'un robot, qu'on trouve dans la littérature, nous avons retenu:

- Celle donnée par le dictionnaire:" *Un robot est un système automatique capable d'accomplir les taches qui lui ont été imposées*". Le mot lui même est originaire du mot Tchèque Robota qui est inclus dans la notion de corvée [1].
- Celle donnée par la JIRA (Japan Industrial Robot Industry Association): "*C'est un système versatile doté d'une mémoire et pouvant effectuer des mouvements comme ceux d'un opérateur humain" [2].*
- Celle donnée par RIA (Robot Institute of America) : "*Un manipulateur à fonctions multiples pouvant être programmé pour réaliser automatiquement des taches variées, éventuellement répétitives" [2].*

La définition descriptive d'un robot est: "*Un robot ou manipulateur est caractérisé par une structure arborescente articulée simple ou multiple dont les segments sont mobiles les un par rapport aux autres. Cet ensemble a pour objectif de mener l'organe terminal vers un lieu géométrique imposé par la tache" [3].*

Parmi les différentes classes de robot établies par D.P.Stoten, et qui sont au nombre de huit [4], on a opté pour un robot de classe quatre. Ce manipulateur est caractérisé par une articulation rotationnelle θ_2 et deux articulations translationnelles, dont les mouvement sont identifiés par les variables d_1 *et* d_2 (Figure I.1).

Afin d'établir les différents modèles, plusieurs hypothèses doivent être prises en considération[4]:

a- Les frottement sont de nature visqueuse, et linéaire par rapport à la vitesse généralisée.

b- Les différentes liaisons sont rigides.
c- Les actionneurs sont idéaux (la force généralisée est directement

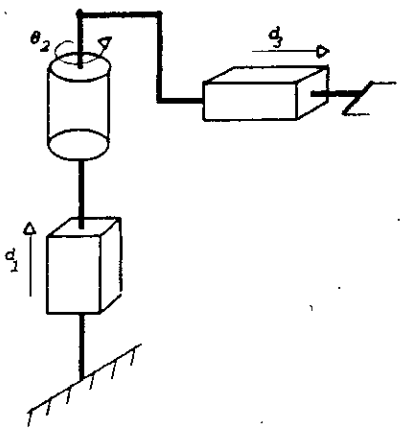


Figure I.1 Schéma du robot

proportionnelle au signal de commande).
d- Les capteurs ont un gain unitaire, et de dynamique négligeable.

I.2 Modélisation cinématique 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 [5][6] 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.

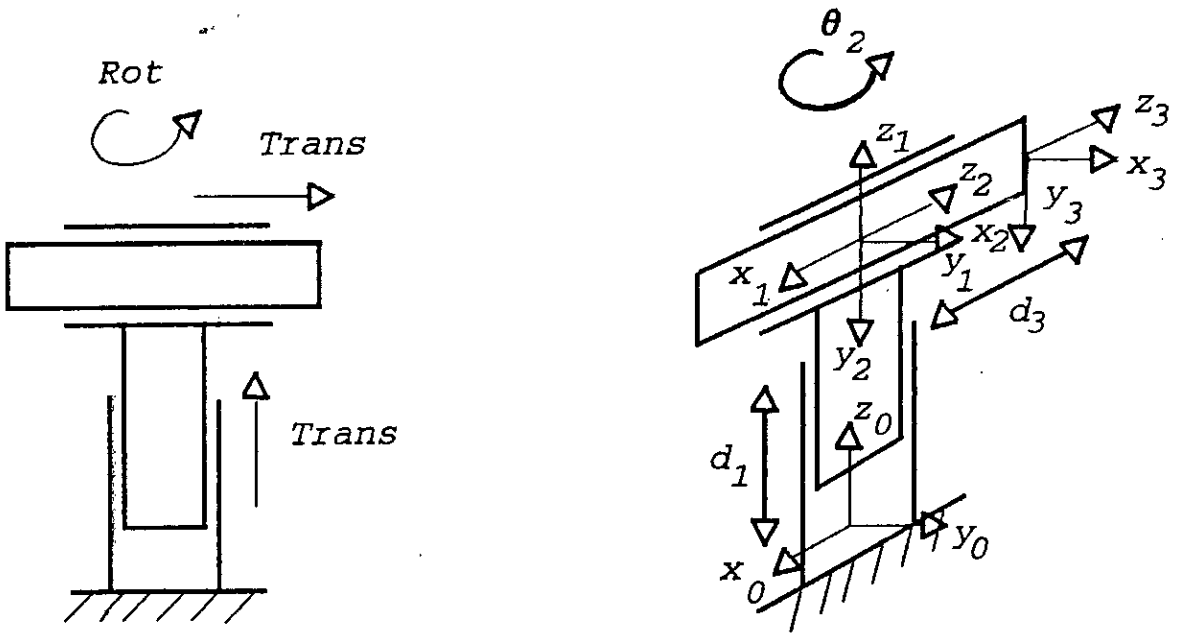


Figure I.2 Implémentation des transformations de DH

En rappellant que d_1, θ_2 et d_3 sont les coordonnées généralisées du robot, on obtient le tableau suivant:

Maillon	Variables	α_i	a_i	d_i	θ_i
1	d_1	0°	0	d_1	**0**
2	θ_2	-90°	0	0	θ_2
3	d_3	0°	0	d_3	0

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

$$ {}_0T^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \quad {}_1T^2 = \begin{bmatrix} C_2 & 0 & -S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \quad {}_2T^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} . $$

${}_iT^{i+1}$: Matrice de passage du repère $R_i(x_i, y_i, z_i)$ au repère $R_{i+1}(x_{i+1}, y_{i+1}, z_{i+1})$

et $C_i = Cos\,\theta_i$; $S_i = Sin\,\theta_i$.

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

$$ A = {}_0T^3 = {}_0T^1 \; {}_1T^2 \; {}_2T^3 \; ; \qquad A = \begin{bmatrix} C_2 & 0 & -S_2 & -d_3S_2 \\ S_2 & 0 & C_2 & d_3C_2 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} . $$

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

$$ {}_0T^2 = {}_0T^1 \; {}_1T^2 \; ; \qquad {}_0T^2 = \begin{bmatrix} C_2 & 0 & -S_2 & 0 \\ S_2 & 0 & C_2 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} . $$

I.2.1 Modélisation géométrique [2][7].

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.

$$ r(R_0) = F(q) \qquad (I.1) $$

où $q^T = [\, q_1 \; q_2 \; \ldots \; q_n \,]$; n: degré de liberté du robot.
$r^T = [x \; y \; z \; 1]$; $F(q) = {}_0T^n \; {}_nr^n$.
${}_nr^n$: coordonnées de l'élément terminal dans $R_n(x_n, y_n, z_n)$.
$F(q)$: fonction vectorielle.

$$F: R^n \rightarrow R^4$$
$$q \rightarrow r = [x\ y\ z\ 1]^T = [f_1(q) \ldots f_4(q)]^T .$$

Dans le cas du robot de classe 4 (n=3) on a:
${}_3r^{3T} = [\ 0\ 0\ 0\ 1\ ]$ et $r = {}_0T^3\ {}_3r^3 = [\ -d_3S_2\ \ d_3C_2\ \ d_1\ 1\ ]^T$.
Donc

$$\begin{cases} f_1(q) = -d_3 \sin\theta_2 \\ f_2(q) = d_3 \cos\theta_2 \\ f_3(q) = d_1 \\ f_4(q) = 1 \end{cases} \qquad (I.2)$$

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 Δq [6]. D'où l'on tire les coordonnées généralisées q_i:

$$\begin{cases} d_1 = z \\ \theta_2 = \arctan(-\frac{x}{y}) \\ d_3 = \frac{y}{\cos\theta_2} \end{cases} \qquad (I.3)$$

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

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 (I.3) peut être utilisée pour la commande cinématique des mécanismes considérés mais il ne s'agit très souvent que de systèmes particulièrement simples.

I.2.2 Modèle cinématique.

Pour résoudre le problème lié aux singularités du modèle géométrique inverse, une procédure systématique de calcul a été introduite et utilisée dans de nombreuses réalisations. Il s'agit du modèle liant les variations des coordonnées généralisées Δq_i et des coordonnées opérationnels $(\Delta x\ \Delta y\ \Delta z\ 0)$, c'est le modèle "*cinématique*" ou "*variationnel*" [7][8].
Pour des déplacements et des vitesses assez faibles, on peut linéariser l'équation (I.1), et utiliser le modèle linéaire pour générer une trajectoire à chaque instant. Pour une position r_0, on a la configuration q_0 du robot. le développement en série de Taylor de la fonction $F(q)$ au point q_0 donne:

$$f_j(q) = \sum_{k=0}^{\infty} \frac{1}{k!} \left\{ \sum_{i=1}^{n} \frac{\partial}{\partial q_i} (q_i - q_{i_0}) \right\}^j f_j(q)\big|_{q_0}; \quad j=1,4 \qquad (I.4)$$

On se limitera au développement jusqu'au premier ordre, d'où:

$$f_j(q) = f_j(q_0) + \sum_{i=1}^{n} \frac{\partial f_j(q)}{\partial q_i}\Big|_{q_0}(q_i - q_{i_0}) ; \quad j=1,4 \tag{I.5}$$

l'écriture de cette équation sous forme matricielle donne:

$$\begin{bmatrix} f_1(q)-f_1(q_0) \\ f_2(q)-f_2(q_0) \\ f_3(q)-f_3(q_0) \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(q)}{\partial q_1} & \cdots & \frac{\partial f_1(q)}{\partial q_n} \\ & & \\ \frac{\partial f_3(q)}{\partial q_1} & \cdots & \frac{\partial f_3(q)}{\partial q_n} \end{bmatrix} \begin{bmatrix} q_1-q_{1_0} \\ . \\ . \\ . \\ q_n-q_{n_0} \end{bmatrix} \tag{I.6}$$

car $f_4(q) = 1$.

Sous forme condensée, on a:

$$r-r_0 = [\,J\,]\,(q-q_0) \tag{I.7}$$

Avec $[J]$: Matrice Jacobienne déterminée à partir de l'équation (I.6).

Le modèle cinématique inverse est obtenu en inversant simplement la matrice $[J]$:

$$q-q_0 = [\,J\,]^{-1}\,(q-q_0) \tag{I.8}$$

L'équation précédente nous permet le calcul des variations des coordonnées généralisées Δq, en utilisant de petites variations Δr. .
Dans le cas du robot de classe quatre (n=3): $q^T = [d_1 \;\; \theta_2 \;\; d_3]$
En utilisant les équations (I.2) et (I.6) la matrice Jacobienne est définie par:

$$[J] = \begin{bmatrix} 0 & -d_3C_2 & -S_2 \\ 0 & -d_3S_2 & C_2 \\ 1 & 0 & 0 \end{bmatrix} \; ; \text{ au point } \; q_0^T = [d_{1_0} \;\; \theta_{2_0} \;\; d_{3_0}] \; .$$

Le calcul analytique de $[\,J\,]^{-1}$ est obtenu facilement à partir de:

$$[J]^{-1} = \begin{bmatrix} 0 & 0 & 1 \\ -\frac{C_2}{d_3} & -\frac{S_2}{d_3} & 0 \\ -S_2 & d_3C_2 & 0 \end{bmatrix} \; ; \text{ avec } \; d_3 \neq 0 \; .$$

Donc à chaque instant t_i, on remplace $q(t_{i-1})$, et on calcule $q(t_i)$, en connaissant $r(t_i)$ et $r(t_{i-1})$. Ainsi on a généré les différentes variables généralisées $q(t_i)$. La forme générale du modèle variationnel inverse devient:

$$\begin{bmatrix} d_1(t_i)-d_1(t_{i-1}) \\ \theta_2(t_i)-\theta_2(t_{i-1}) \\ d_3(t_i)-d_3(t_{i-1}) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \frac{-\cos(\theta_2(t_{i-1}))}{d_3(t_{i-1})} & \frac{-\sin(\theta_2(t_{i-1}))}{d_3(t_{i-1})} & 0 \\ -\sin(\theta_2(t_{i-1})) & d_3(t_{i-1})\cos(\theta_2(t_{i-1})) & 0 \end{bmatrix} \begin{bmatrix} x(t_i)-x(t_{i-1}) \\ y(t_i)-y(t_{i-1}) \\ z(t_i)-z(t_{i-1}) \end{bmatrix} \tag{I.9}$$

L'expression ainsi établi, nous permet d'éviter le calcul répétitif de l'inversion de la matrice Jacobienne.

I.2.3 Génération de trajectoire.

L'équation (I.9) qui caractérise le modèle géométrique variationnel peut être considérée comme une procédure de génération de trajectoire. En effet à chaque point de coordonnées cartésiennes, on détermine les coordonnées généralisées correspondants. Suivant la trajectoire décrite par l'élément terminal, on peut trouver une relation entre les coordonnées généralisées q et les paramètres introduits pour décrire cette même trajectoire.
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 $R(x\ y\ z)$ (voir figure I.3).

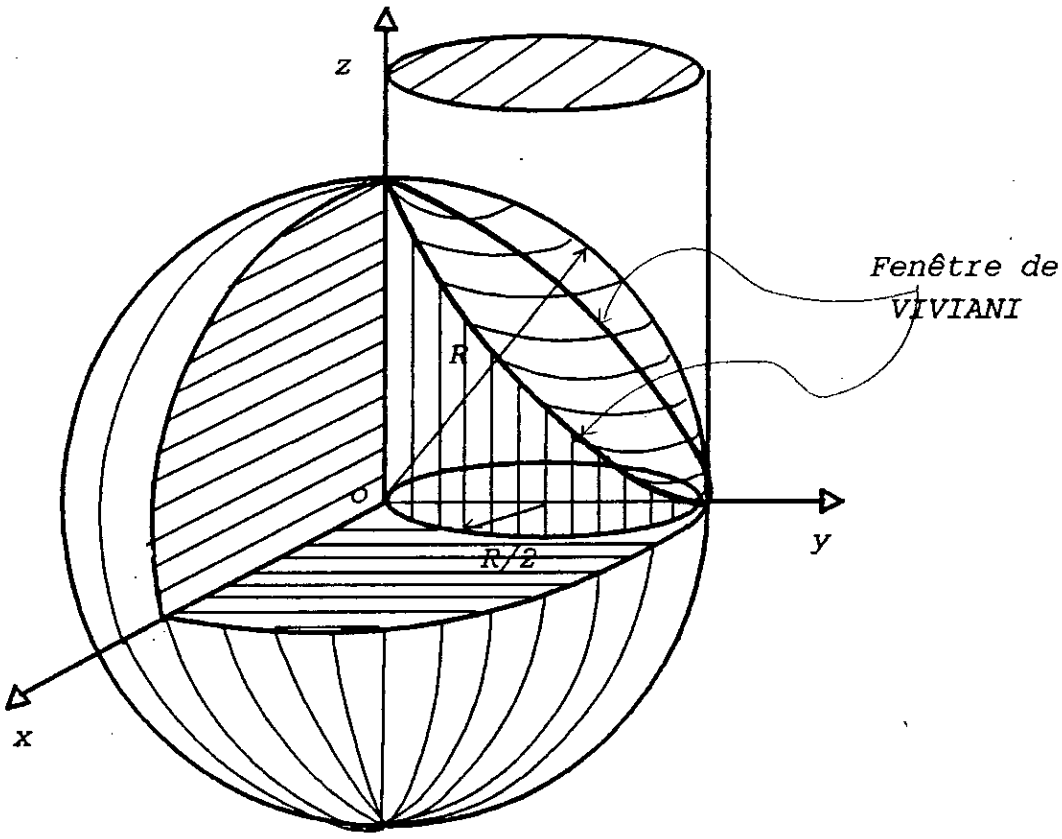


Figure I.3 Fenêtre de VIVIANI.

L'équation du cylindre exprimé dans $R(x\ y\ z)$ est:

$$x^2+(y-\frac{R}{2})^2 = (\frac{R}{2})^2 \qquad (I.10)$$

et de la sphère :

$$x^2+y^2+z^2 = R^2 \qquad (I.11)$$

Soit le changement de variable suivant: $x=\rho\cos\theta$ *et* $y=\rho\sin\theta$. (Figure I.4).
En remplaçant dans (I.10) et (I.11) on obtient $\rho=R\sin\theta$ *et* $z^2=R^2-\rho^2$.
En éliminant la dépendance de z en fonction de ρ , on a:

$$\begin{cases} \rho=R\sin\theta \\ z=|R\cos\theta| \end{cases} \qquad (I.12)$$

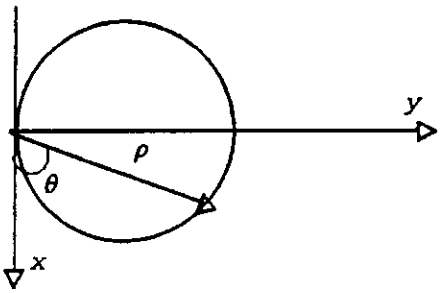
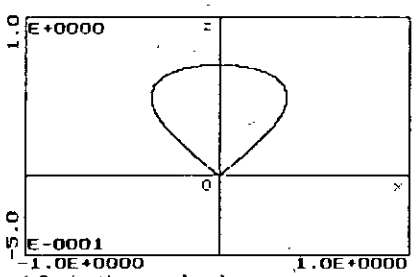


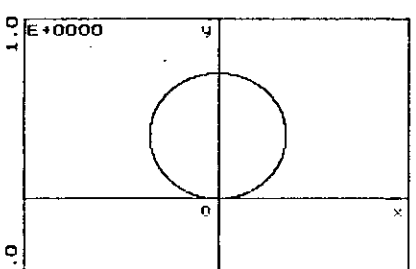
Figure I.4 Schéma de présentation du changement de variable.

Ainsi on détermine la courbe paramètrée dans l'espace, où ρ, z *et* θ sont similaires respectivement à d_3, d_1 *et* θ_2 . Vu que z, ρ dépendent de θ, on a qu'à imposer une seule consigne θ_2 .

Donc, par cette méthode géométrique simple on a pu déterminer l'évolution des variables q en fonction du temps sans utiliser, ni le modèle géométrique, ni le modèle cinématique (cas particulier pour cette trajectoire, la structure du robot est très adaptable aux coordonnées cylindriques).
Pour que l'élément terminal décrive la trajectoire de la fenêtre de Viviani, les consignes pour chaque variable, pour accomplir cette tache sont (en utilisant l'équation (I.12) pour θ=0 ; z=R ρ=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 (voir figures I.5.a et b).

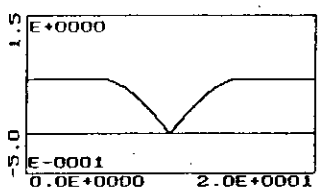


1.Projection sur le plan yoz.

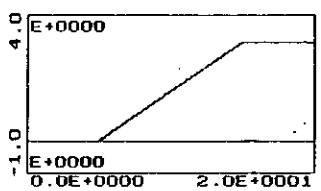


2.Projection sur le plan xoy.

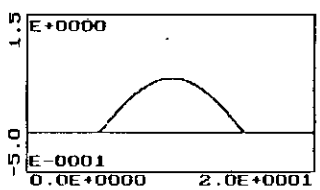
Figure I.5.b Projection des trajectoires désirées de la fenetre de VIVIANI sur les plan principaux.



1.Variable z.(translation)



2.Variable θ.(rotation)



3.Variable (translation)

Figure I.5.a Evolution des différentes variables généralisées dans la temps pour la fenetre de VIVIANI.

Remarque.

L'évolution de la variable θ_2 est choisi linéaire par rapport au temps, de pente α variable par rapport au temps.□

I.3 Modélisation dynamique du robot.[1][2][8]

Après avoir défini la Modélisation cinématique du robot et la spécification de ses taches, on s'intéresse 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 des robots 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 du robot.

I.3.1 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 [9]:

$$\frac{d}{dt}\left(\frac{\delta E_c}{\delta \dot{q}_i}\right)-\frac{\delta E_p}{\delta q_i}+\frac{\delta E_D}{\delta \dot{q}_i}=\tau_i \quad ;i=1,n \tag{I.13}$$

avec E_c :Energie cinétique; E_p :Energie potentielle; E_D :Energie de dissipation.
τ_i :Force généralisée; n: degré de liberté.

De manière simplifiée, l'équation (I.13) devient [1]:

$$\frac{d}{dt}\left(\frac{\delta L}{\delta \dot{q}_i}\right)-\frac{\delta L}{\delta q_i}+\frac{\delta E_D}{\delta \dot{q}_i}=\tau_i \quad ;i=1,n \tag{I.14}$$

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

Les matrices de passage de DH sont définies par:

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

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[1]:

$${}_0V^i=\sum_{j=1}^{i}\left[U_{ij}\dot{q}_j\right]_i r^i \tag{I.15}$$

tel que $U_{ij}=\begin{cases} {}_0T^{j-1}Q_j\ {}_{j-1}T^i & j\le i \\ 0 & J\ge i\end{cases}$ 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; ${}_i r^i$: 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:

$$dK_i = \frac{1}{2} trace(V_i V_i^T)\, dm = \frac{1}{2} trace\left[\sum_{p=1}^{i}\sum_{r=1}^{i} U_{ip}\ {}_i r^i dm\ {}_i r^{i\,T} U_{ir}^T \dot{q}_p \dot{q}_r\right] \qquad (I.16)$$

Donc l'énergie cinétique de la liaison i est

$$K_i = \int dK_i = \frac{1}{2} trace\left[\sum_{p=1}^{i}\sum_{r=1}^{i} U_{ip} J_i U_{ir}^T \dot{q}_p \dot{q}_r\right] \qquad (I.17)$$

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:

$$K = \sum_{i=1}^{n} K_i + E_{c_a} \qquad (I.18)$$

où $E_{c_a} = \frac{1}{2}\sum_{l=1}^{n} I_l \dot{q}_l^2$:Energie cinétique introduite par les actionneurs [1].

I_l :moment d'inertie de l'actionneur 1 dans le cas de la rotation et masse de l'actionneur dans le cas de la translation [1][10].

L'énergie potentielle est [1][7]:

$$P = \sum_{i=1}^{n} -m_i g^T\ {}_0 r^i = \sum_{i=1}^{n} -m_i g^T({}_0 T^i\ {}_i r^i) \qquad (I.19)$$

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 [1][7]:

$$E_D = \frac{1}{2}\sum_{l=1}^{n} f_l \dot{q}_l^2 \qquad (I.20)$$

où f_l: coefficient de frottement visqueux.

Des équations (I.14) à (I.20) on calcule les forces généralisées[1]:

$$\tau_i=\sum_{k=i}^{n}\sum_{p=1}^{k} trace\left[U_{kp}J_kU_{ki}^T\right]\ddot{q}_p-\sum_{k=i}^{n} m_k g^T U_{ki}\ {}^k r^k$$

$$+\sum_{k=i}^{n}\sum_{p=1}^{k}\sum_{r=1}^{k} trace\left[U_{kpr}J_kU_{ki}^T\right]\dot{q}_r\dot{q}_p+I_i\ddot{q}_i+f_i\dot{q}_i;\quad i=1,n \qquad (I.21)$$

où $U_{ijk}=\begin{cases} {}_0T^{k-1}Q_k\ {}_{k-1}T^{j-1}Q_j\ {}_{j-1}T^i & ;\ k\le j\le i. \\ {}_0T^{j-1}Q_j\ {}_{j-1}T^{k-1}Q_k\ {}_{k-1}T^i & ;\ j\le k\le i. \\ 0 & ;\ j\le i\le k. \end{cases}$

Dans le cas du robot de classe quatre on a :

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

$$U_{22}={}_0T^1Q_2\ {}_1T^2=\begin{bmatrix}-S_2&0&-C_2&0\\C_2&0&-S_2&0\\0&0&0&0\\0&0&0&0\end{bmatrix};\quad U_{32}={}_0T^1Q_2\ {}_1T^3=\begin{bmatrix}-S_2&0&-C_2&-d_3C_2\\C_2&0&-S_2&-d_3S_2\\0&0&0&0\\0&0&0&0\end{bmatrix};$$

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

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

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

$$U_{322}={}_0T^1Q_2\ {}_1T^1Q_2\ {}_1T^3=\begin{bmatrix}-C_2&0&S_2&d_3S_2\\-S_2&0&-C_2&-d_3C_2\\0&0&0&0\\0&0&0&0\end{bmatrix};\quad J_i=\begin{bmatrix}\alpha_i&0&0&m_i\overline{x}_i\\0&\beta_i&0&m_i\overline{y}_i\\0&0&\gamma_i&m_i\overline{z}_i\\m_i\overline{x}_i&m_i\overline{y}_i&m_i\overline{z}_i&m_i\end{bmatrix}.$$

Les produits d'inertie sont nuls car $R_i(x_iy_iz_i)$ est parallèle au repère principale de la liaison i.

${}_ir^{i^T}=[\ \overline{x}_i\ \overline{y}_i\ \overline{z}_i\ 1]$: 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.21) on a:

$$\begin{cases} F_1 = (m_1+m_2+m_3)\,(\ddot{q}_1+g) + f_1\dot{q}_1 . \\ \tau_2 = [\alpha_2+\gamma_2+\alpha_3+\gamma_3+d_3m_3\overline{z}_3+m_3(d_3\overline{z}_3+d_3^2)]\,\ddot{q}_2 + \\ \quad m_3\overline{x}_3\ddot{q}_3+2m_3(\overline{z}_3+d_3)\,\dot{q}_3\dot{q}_2+f_2\dot{q}_2 . \\ F_3 = m_3\overline{x}_3\ddot{q}_2+m_3\ddot{q}_3-m_3(\overline{z}_3+d_3)\,\dot{q}_2^2+f_3\dot{q}_3 . \end{cases} \qquad (I.22)$$

Suivant la structure de notre robot on a:

$_2r^2 \approx 0$, $m_2 \approx 0$; $\alpha_2=\gamma_2 \approx 0$; $\overline{x}_2=\overline{y}_2=\overline{z}_2 \approx 0$. $\overline{x}_1=\overline{y}_1=0$; $\overline{z}_1=-\frac{l_1}{2}$; $\overline{x}_3=\overline{y}_3=0$; $\overline{z}_3=-\frac{l_2}{2}$.

Suivant la figure I.6 le calcul de α_3 *et* γ_3 se fait comme suit:

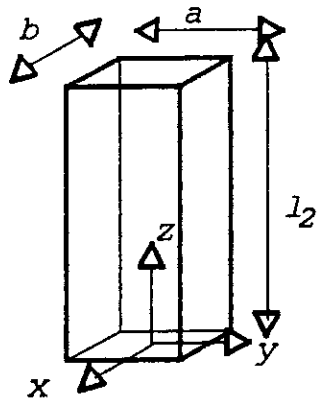


Figure I.6 Schéma de la liaison 3.

$$\alpha_3 = \int x^2 dm_3 = \frac{m_3 b^2}{12} ; \quad \gamma_3 = \int z^2 dm_3 = \frac{m_3 l_2^2}{3}$$

En considérant b négligeable devant l_2 , on a $\alpha_3 < \gamma_3$, on prend alors $\alpha_3 \approx 0$.
En prenant $F_1=k_1U_1$; $\tau_2=k_2U_2$; $F_3=k_3U_3$ L'équation (I.22) devient :

$$\begin{cases} U_1 = \dfrac{(m_1+m_3)}{k_1}(\ddot{q}_1+g) + \dfrac{f_1}{k_1}\dot{q}_1 . \\ U_2 = \dfrac{m_3\dfrac{l_2^2}{3}+2m_3\overline{z}_3q_3+m_3q_3^2}{k_2}\ddot{q}_2 + +2\dfrac{m_3}{k_2}(\overline{z}_3+q_3)\,\dot{q}_3\dot{q}_2+\dfrac{f_2}{k_2}\dot{q}_2 . \\ U_3 = \dfrac{m_3}{k_3}\ddot{q}_3-\dfrac{m_3(\overline{z}_3+q_3)}{k_3}\dot{q}_2^2+\dfrac{f_3}{k_3}\dot{q}_3 . \end{cases} \qquad (I.23)$$

I.3.1.a Résultats de simulation.

La simulation du modèle de connaissance établi par le formalisme de EL nécessite la mise sous forme d'état des équations différentielles non linéaires trouvées (équation (I.23)). La forme d'état établie est:

$$\dot{x}=A\,x+B\,U+d \qquad y=c\,x \tag{I.24}$$

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.

et $A=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{f1}{m_1+m_3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{f_2}{m_3 j^*} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{f_3}{m_3} \end{bmatrix}$; $B=\begin{bmatrix} 0 & 0 & 0 \\ \frac{k1}{m_1+m_3} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{k_2}{m_3 j^*} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{k_3}{m_3} \end{bmatrix}$; $C=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$.

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

La méthode de Range-Kutta (RK) d'ordre quatre nous permet de résoudre numériquement l'équation (I.24), par un choix judicieux du pas de calcul dt.(les conditions initiales sur x sont nulles). $x^T(t=0)=[0\ \ 0\ 0\ 0\ 0\ 0]; dt=0.001s$.
Les caractéristiques du robot, telles que les masses et les dimensions, sont [4]:

$m_1=20kg$; $m_3=10kg$; $l_2=0.75m$; $g=9.81ms^{-2}$; $k_1=100N/V$; $k_2=10Nm/V$; $k_3=10N/V$;
$f_1=30Nsm^{-1}$; $f_2=7.825Nm\ rad^{-1}s$; $f_3=20Nsm^{-1}$.

Les figures I.8.a et b montrent la réponse du robot en position, vitesse et accélération, pour une excitation indicielle, sur un horizon de temps suffisant.

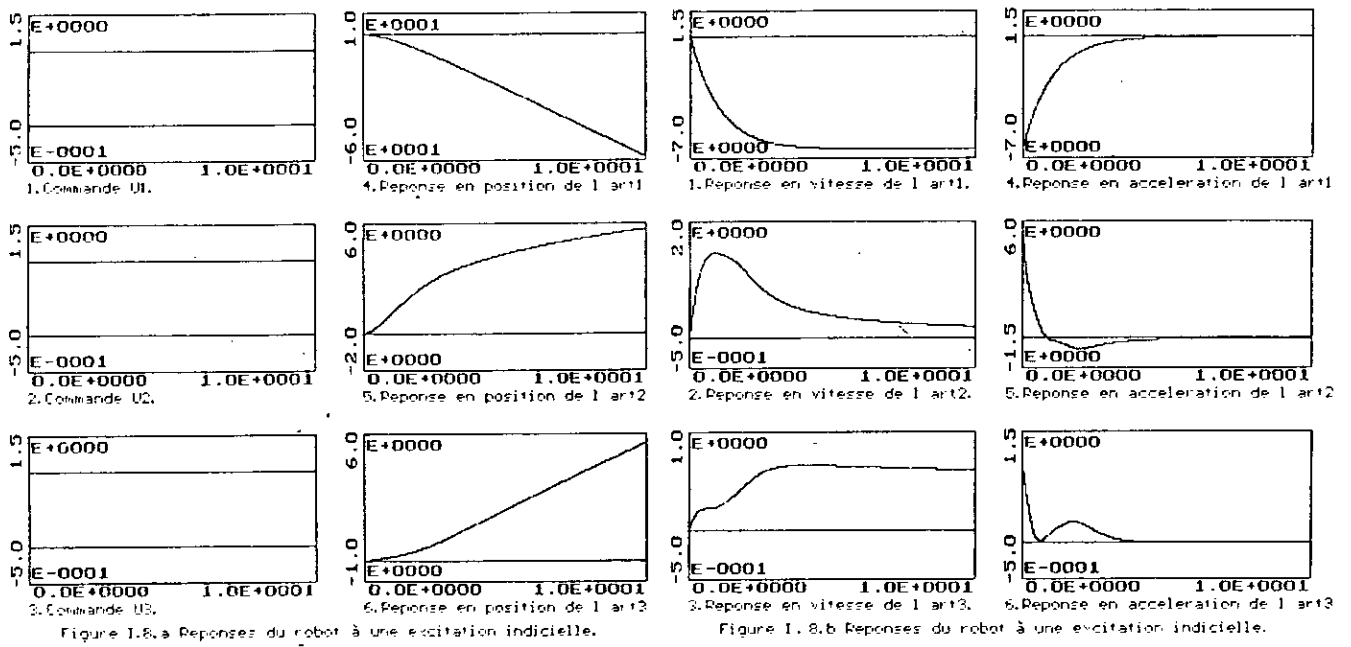


Figure I.8.a Reponses du robot à une excitation indicielle.

Figure I.8.b Reponses du robot à une excitation indicielle.

I.3.2 Formalisme de Newton-Euler.

Dans la partie précédente nous avons élaboré des équations différentielles non linéaires du second ordre obtenues par le formalisme d' **Euler-Lagrange**. 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 récursive 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 [1]:

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

Dans cette étape on calcule 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. On commence de la base jusqu'à l'élément terminal.

$$ {}_iR^0\omega_i = \begin{cases} {}_iR^{i-1}({}_{i-1}R^0\omega_{i-1} + z_0\dot{q}_i) & si\ ①\ est\ verifiée. \\ {}_iR^{i-1}({}_{i-1}R^0\omega_{i-1}) & si\ ②\ est\ verifiée. \end{cases} \tag{I.25} $$

$$ {}_iR^0\dot{\omega}_i = \begin{cases} {}_iR^{i-1}({}_{i-1}R^0\dot{\omega}_{i-1} + z_0\ddot{q}_i + \\ ({}_{i-1}R^0\omega_{i-1}) \wedge z_0\dot{q}_i) & si\ ①\ est\ verifiée. \\ {}_iR^{i-1}({}_{i-1}R^0\dot{\omega}_{i-1}) & si\ ②\ est\ verifiée. \end{cases} \tag{I.26} $$

$$ {}_iR^0\dot{V}_i = \begin{cases} ({}_iR^0\dot{\omega}_i) \wedge ({}_iR^0P_i^*) + ({}_iR^0\omega_i) \wedge [({}_iR^0\omega_i) \wedge ({}_iR^0P_i^*) \\ \quad + {}_iR^{i-1}({}_{i-1}R^0\dot{V}_{i-1}) & si\ ①\ est\ vérifiée. \\ {}_iR^{i-1}(Z_0\ddot{q}_i + {}_{i-1}R^0\dot{V}_{i-1}) + ({}_iR^0\dot{\omega}_i) \wedge ({}_iR^0P_i^*) \\ \quad + 2({}_iR^0\omega_i) \wedge ({}_iR^{i-1}Z_0\dot{q}_i) \\ + ({}_iR^0\dot{\omega}_i) \wedge [({}_iR^0\omega_i) \wedge ({}_iR^0P_i^*)] & si\ ②\ est\ vérifiée. \end{cases} \tag{I.27} $$

$$ {}_iR^0\overline{a}_i = ({}_iR^0\dot{\omega}_i) \wedge ({}_iR^0\overline{S}_i) + ({}_iR^0\omega_i) \wedge [({}_iR^0\omega_i) \wedge ({}_iR^0\overline{S}_i)] + {}_iR^0\dot{V}_i \tag{I.28} $$

2°. Deuxième étape."Backward équations" 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 τ_{n+1} issus du monde extérieur.

$$ {}_iR^0f_i = {}_iR^{i+1}({}_{i+1}R^0f_{i+1}) + m_i\ {}_iR^0\overline{a}_i. \tag{I.29} $$

$$ \begin{aligned} {}_iR^0\eta_i = {}_iR^{i+1}[{}_{i+1}R^0\eta_{i+1} + ({}_{i+1}R^0P_i^*) \wedge ({}_{i+1}R^0f_{i+1})] \\ + ({}_iR^0P_i^* + {}_iR^0\overline{S}_i) \wedge ({}_iR^0F_i) + ({}_iR^0I_i\ {}_0R^i)\ ({}_iR^0\dot{\omega}_i) \\ + ({}_iR^0\omega_i) \wedge [({}_iR^0I_i\ {}_0R^i)\ ({}_iR^0\omega_i)]. \end{aligned} \tag{I.30} $$

f_i *et* η_i sont respectivement la force et le moment exercés sur la liaison i par la liaison i-1 dans $R_{i-1}(x_{i-1}y_{i-1}z_{i-1})$.

$$\tau_i = \begin{cases} ({}_iR^0\eta_i)^T({}_iR^{i-1}Z_0) + c_i\dot{q}_i & si\ ②\ est\ vérifiée. \\ ({}_iR^0 f_i)^T({}_iR^{i-1}Z_0) + c_i\dot{q}_i & si\ ①\ est\ vérifiée. \end{cases} \quad (I.31)$$

où c_i: coefficient de frottements.
La figure I.9 explicite bien les différents paramètres de l'algorithme [11].

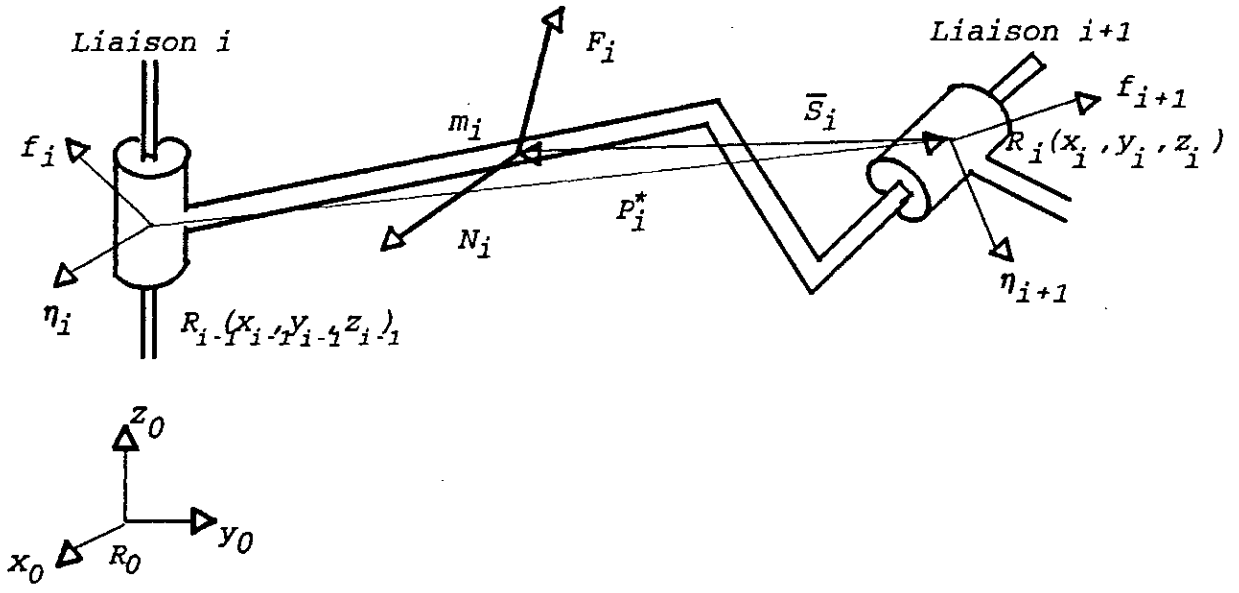


Figure I.9 Présentation des différentes grandeurs.

Λ: produit vectoriel.
①: Liaison translationnelle.
②: Liaison rotationnelle.
Où $F_i = m_i\bar{a}_i$: Force extérieure exercée sur la liaison i en $\bar{S}_i$.
$N_i = I_i\dot{\omega}_i + \omega_i\Lambda(I_i\omega_i)$: Couple extérieur exercé sur la liaison i en $\bar{S}_i$.
${}_iR^0P_i^* = [\ a_i\ \ d_i\sin\alpha_i\ \ d_i\cos\alpha_i\ ]^T$.
${}_iR^0\bar{S}_i$: centre de masse de la liaison i dans $R_i(x_iy_iz_i)$.
${}_iR^0I_i\ {}_0R^i$: Matrice d'inertie de la liaison i autour de son centre de masse.

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

$$\omega_0 = \dot{\omega}_0 = V_0 = 0; \quad \dot{V}_0 = [g_x\ g_y\ g_z]^T; \quad |\dot{V}_0| = g.$$

L'application de cet algorithme au robot de classe 4, exige de définir certains paramètres.

$${}_0R^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad {}_1R^2 = \begin{bmatrix} C_2 & 0 & -S_2 \\ S_2 & 1 & C_2 \\ 0 & -1 & 0 \end{bmatrix}; \quad {}_1R^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad {}_2R^1 = \begin{bmatrix} C_2 & S_2 & 0 \\ 0 & 0 & -1 \\ -S_2 & C_2 & 0 \end{bmatrix}.$$

$${}_2R^3 = {}_0R^1; \quad {}_0R^2 = {}_1R^2; \quad {}_0R^3 = {}_1R^2; \quad {}_1R^3 = {}_1R^2; \quad {}_2R^0 = {}_2R^1; \quad {}_3R^0 = {}_2R^1; \quad {}_3R^1 = {}_2R^1.$$

$${}_1R^0P_1^* = [0\ 0\ d_1]; \quad {}_2R^0P_2^* = [0\ 0\ 0]; \quad {}_3R^0P_3^* = [0\ 0\ d_3]; \quad {}_2R^0\bar{S}_2 = [0\ 0\ 0];$$

$${}_1R^0\overline{s}_1=[0\ 0\ \overline{z}_1];\quad {}_3R^0\overline{s}_3=[0\ 0\ \overline{z}_3];\quad \overline{z}_1=-\frac{l_1}{2};\quad \overline{z}_3=-\frac{l_2}{2}.$$

$${}_iR^0I_i\ {}_0R^i=\begin{bmatrix} I_{xx_i} & 0 & 0\\ 0 & I_{yy_i} & 0\\ 0 & 0 & I_{zz_i}\end{bmatrix};\ i=1,3;\ \text{avec}\ \begin{matrix} I_{xx_i}=\int (y_i^2+z_i^2)\,dm_i\\ I_{yy_i}=\int (x_i^2+z_i^2)\,dm_i.\\ I_{zz_i}=\int (x_i^2+y_i^2)\,dm_i\end{matrix}$$

Les produits d'inertie sont nulles car $R_i(x_iy_iz_i)$ est parallèle aux axes principaux de la liaison considérée.

Pour une barre de longueur l, de section ab, on a d'après la figure I.6 :

$$I_{xx}=\int (y^2+z^2)\,dm=\frac{ml^2}{12}.$$

$$I_{yy}=\int (x^2+z^2)\,dm=\frac{ml^2}{12}.$$

$$I_{zz}=\int (x^2+y^2)\,dm=\frac{m(a^2+b^2)}{12}.$$

Pour i=2 la liaison est ponctuelle donc ${}_2R^0I_2\ {}_0R^2\approx 0$. Le moment d'inertie de la liaison 1 n'a aucun effet, puisqu'elle se déplace seulement en translation. Pour la liaison 3, il n'y a que l'effet de I_{xx_3}, car cette liaison tourne autour de Y_3 , donc on peut prendre:

$${}_3R^0I_3\ {}_0R^3=\begin{bmatrix} 0 & 0 & 0\\ 0 & m_3\frac{l_2^2}{12} & 0\\ 0 & 0 & 0\end{bmatrix}.$$

Remarque.

La transformation R utilisée dans cette partie est une matrice extraite de celle de DH (transformation T).□

Ainsi l'application de l'algorithme devient facile.(Voir annexe 2 pour l'organigramme et le programme). Pour mieux comprendre l'algorithme, dans l'annexe 3, on développe les calculs analytiques.

I.3.2.a. Résultats de simulation.

Avec les mêmes caractéristiques du robot données dans la partie précédente, on simule l'algorithme de NE, pour des consignes sinusoïdales et un couple et une force nulles ($f_4=\eta_4=0$) . La figure I.10 nous montre les forces généralisées (commandes) nécessaire pour accomplir la tâche voulue.

I.3.3 Validation.

Toute phase de modélisation est soumise à la validation. l'obtention d'un modèle dynamique est assujettie à une critique. Pour confirmer un modèle trouvé, on est toujours amené à choisir un autre concept dans lequel on modélise le système en utilisant des lois différente de celles utilisées pour le modèle primal.
Dans notre cas, on s'est intéressé à modéliser le robot, en utilisant deux formalisme, celui du principe de moindre action (LE) et celui du principe de D'Alambert (NE).

La récursivité de l'algorithme de NE lui permet une implémentation directe sur calculateur. L'utilisation de cet algorithme ,pour le calcul manuel des équations différentielles du modèle, est une étape très délicate. On présente dans l'annexe 3 le calcul, fait analytiquement, du modèle du robot. On remarque que le modèle obtenu est le même que celui trouvé par le formalisme de LE. Donc, théoriquement on peut confirmer que notre calcul du modèle est parfait. Le modèle obtenu décrit bien le comportement dynamique du robot.
Dans le cas de la simulation, les deux modèle se valident mutuellement par couplage direct. Les figures I.11 et 12 présentent ce couplage [3].

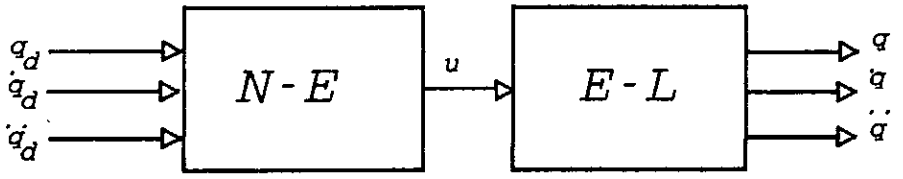


Figure I.11 Couplages des deux modèles NE-EL.

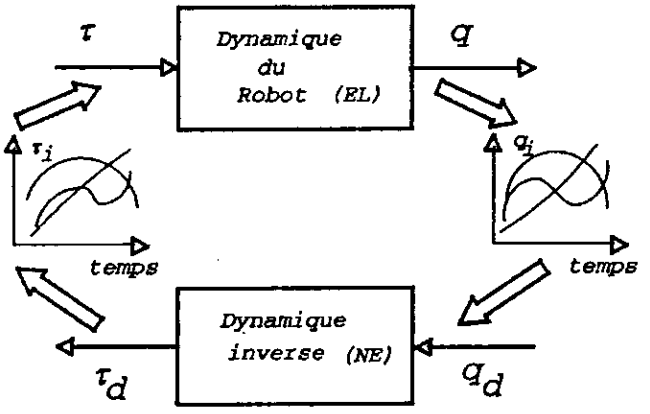


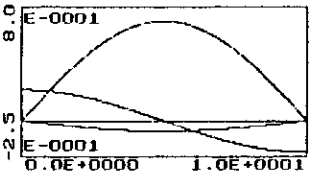
Figure I.12 Schéma explicite des deux modèles NE-EL.

D'après la figure I.12, les deux modèles se valident si les erreurs définies par : $\varepsilon = q_d - q$; $\dot{\varepsilon} = \dot{q}_d - \dot{q}$; $\ddot{\varepsilon} = \ddot{q}_d - \ddot{q}$; seront très faibles. Les figures I.13.a,b et c représentent les positions, vitesses et accélérations désirées et celles calculées par EL. Ces erreurs sont très faible. L'existence de ces erreurs sont dues à la méthode numérique (RK) utilisée pour résoudre les équations différentielles du robot. Si on utilise une autre méthode plus précise que RK (Adams Bashford), l'erreur sera plus faible que celle trouvée.

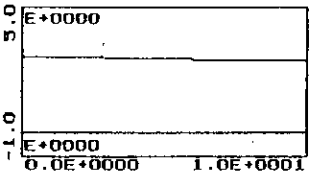
Les caractéristiques du robot sont les mêmes que celles de la partie précédente. Les trajectoires désirées sont définies par:

$$\begin{cases} q_{d_i}(t) = a \sin\omega t \\ \dot{q}_{d_i}(t) = a\omega \cos\omega t \\ \ddot{q}_{d_i}(t) = -a\omega^2 \sin\omega t \end{cases} \quad \text{pour i=1,3 et} \quad x(t=0) = [0 \; a\omega \; 0 \; a\omega \; 0 \; a\omega]^T.$$

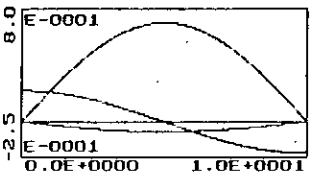
Les forces généralisées sont consignées sur la figure I.13.d



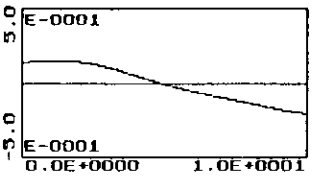
1.Position, vitesse, acceleration désirées de l art1.



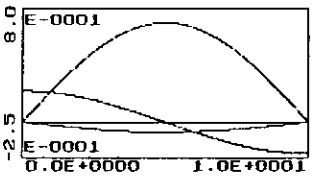
4.Commande U1



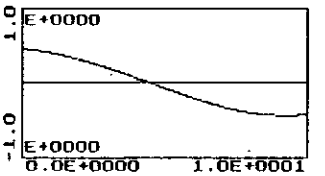
2.Position, vitesse, acceleration désirées de l art2.



5.Commande U2

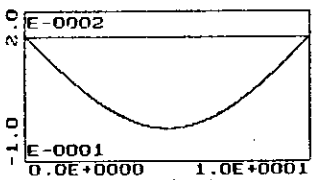


3.Position, vitesse, acceleration désirées de l art3.

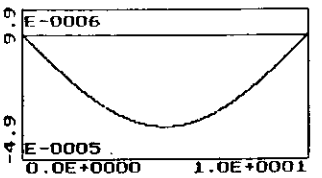


6.Commande U3

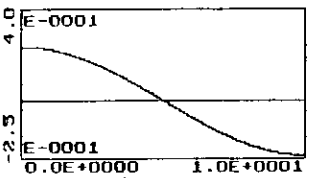
Figure I.10.Representation des trajectoires désirées et des commandes calculées avec l algorithme de Newton-Euler.



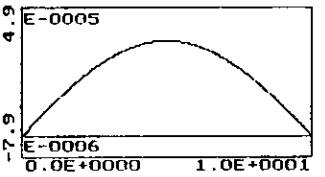
1.Acceleration désirée et reelle de l art1.



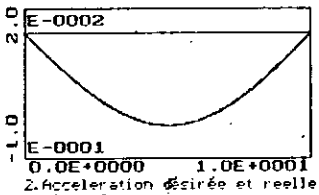
4.Erreur en acceleration de l art1



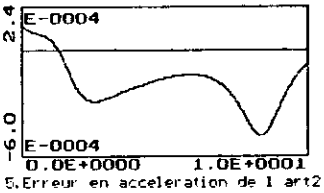
1.Vitesse désirée et reelle de l art1.



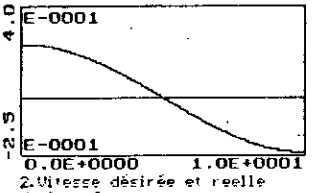
4.Erreur en vitesse de l art1



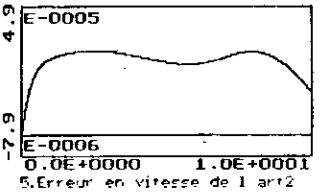
2.Acceleration désirée et reelle de l art2.



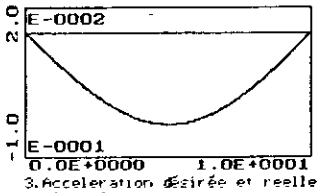
5.Erreur en acceleration de l art2



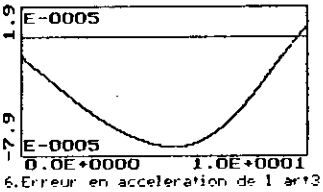
2.Vitesse désirée et reelle de l art2.



5.Erreur en vitesse de l art2

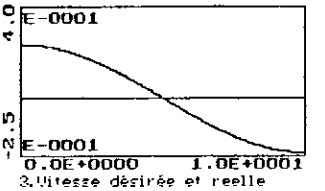


3.Acceleration désirée et reelle de l art3.

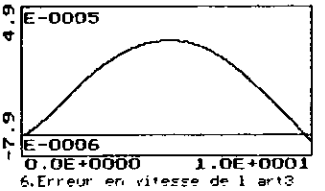


6.Erreur en acceleration de l art3

Figure I.13.c Accelerations et erreurs en acceleration dans le cas du couplage de Newton-Euler et d Euler-Lagrange.

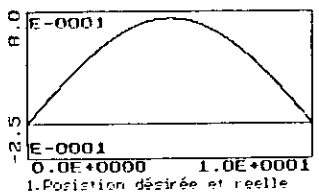


3.Vitesse désirée et reelle de l art3.

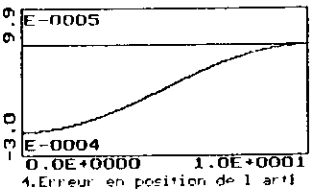


6.Erreur en vitesse de l art3

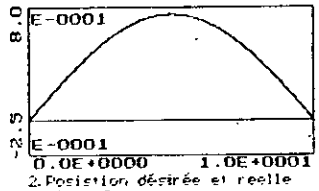
Figure I.13.b vitesses et erreurs en vitesse dans le cas du couplage de Newton-Euler et d Euler-Lagrange.



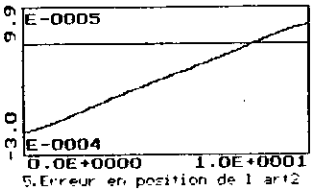
1.Position désirée et reelle de l art1.



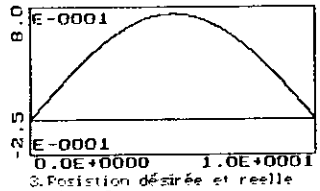
4.Erreur en position de l art1



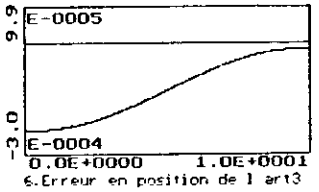
2.Position désirée et reelle de l art2.



5.Erreur en position de l art2

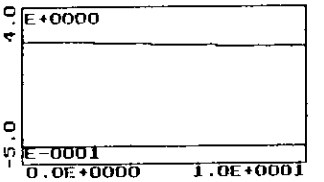


3.Position désirée et reelle de l art3.

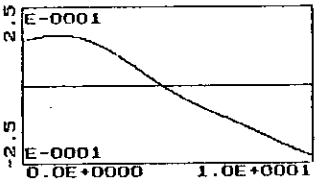


6.Erreur en position de l art3

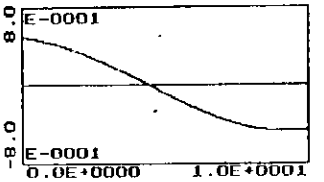
Figure I.13.a Positions et erreurs en position dans le cas du couplage de Newton-Euler et d Euler-Lagrange.



1.Commande u1.



2.Commande u2.



3.Commande u3.

Figure I.13.d Presentation des commandes.

I.3.4 Modèle de représentation.

Après avoir déterminé le modèle de connaissance du robot, on veut maintenant déterminer un modèle de représentation pour faciliter le calcul de la commande en temps réel. On va utiliser des algorithmes adaptatifs pour tenir compte de l'évolution des paramètres du modèle. L'utilisation d'un algorithme d'identification permet la détermination de ces paramètres, aboutissant ainsi au modèle de représentation.
Pour des raisons de simplicité de la commande, on va utiliser un modèle linéaire, où la dynamique est supposée découplée et les interactions seront modélisées par des perturbations. La structure choisie du modèle de représentation est [12]:

$$A(q^{-1})\,y(t) = B(q^{-1})\,U(t) + h \qquad (I.32)$$

avec $h=[h_1 \ldots h_n]^T$: Vecteur absorbant l'effet du couplage.
$U(t)=[U_1(t) \ldots U_n(t)]^T$: Vecteur de commandes.
$y(t)=[y_1(t) \ldots y_n(t)]^T$: Vecteur de sorties.
q^{-1} Opérateur retard; $x(t-1)=q^{-1}x(t)$.
$A(q^{-1})$ et $B(q^{-1})$: Matrices polynomiales en q^{-1} , de dimension nxn, définies par:

$$A(q\text{-}1) = \begin{bmatrix} A_1(q^{-1}) & & 0 \\ & \ddots & \\ 0 & & A_n(q^{-1}) \end{bmatrix}; \quad B(q\text{-}1) = \begin{bmatrix} B_1(q^{-1}) & & 0 \\ & \ddots & \\ 0 & & B_n(q^{-1}) \end{bmatrix}.$$

Où $A_i(q^{-1}) = 1+\sum_{l=1}^{n_i} a_{i_l} q^{-l}$; $i=1,n$; $B_i(q^{-1}) = q^{-d_i}\sum_{l=1}^{m_i} b_{i_l} q^{-l}$; $i=1,n$; $d_i \geq 0$.

La $i^{éme}$ sortie peut s'écrire sous la forme suivante:

$$y_i(t) = -\sum_{l=1}^{n_i} a_{i_l}\, y_i(t-l) + \sum_{l=1}^{m_i} b_{i_l}\, U_i(t-d_i-l) + h_i = \theta_i^T\, \phi_i(t). \qquad (I.33)$$

avec $\theta_i = [a_{i_1} \ldots a_{i_{n_i}}\ b_{i_1} \ldots b_{i_{m_i}}\ h_i]$

et $\phi_i = [-y_i(t-1) \ldots -y_i(t-n_i)\ \ u_i(t-d_i-1) \ldots u_i(t-d_i-m_i)\ \ h_i\ 1]$.

Le système multivariable (le robot) est décomposé en n sous-système SISO (Single Input Single Output). L'algorithme des moindres carrés ordinaires, appliqué au $i^{ème}$ sous-système, est défini par :

$$\theta_i(t+1) = \theta_i(t) + F_i(t)\phi_i(t)\varepsilon_i(t+1). \qquad (I.34.a)$$

$$F_i(t+1) = \frac{1}{\lambda_{i_1}}\left[F_i(t) - \frac{F_i(t)\phi_i(t)\phi_i^T(t)F_i(t)}{c_i+\phi_i^T(t)F_i(t)\phi_i(t)}\right] \qquad (I.34.b)$$

$$\varepsilon_i(t+1) = \frac{y_i(t+1) - \theta_i^T\phi_i(t)}{c_i+\phi_i^T(t)F_i(t)\phi_i(t)}. \qquad (I.34.c)$$

Si $c_i=1$ et $1/\lambda_{i_1}=1$, on obtient l'algorithme des moindres **carrés récursifs** à gain décroissant. Cet algorithme permet une bonne estimation des paramètres variant très lentement dans le temps.
Si $c_i=\lambda_{i_1}/\lambda_{i_2}$ et $trace[F_i(t+1)]=trace[F_i(t)]=trace[F_i(t=0)]$;
avec $F_i(t=0)=diag(GI_i)$; GI_i: Gain initial. On obtient ainsi l'algorithme à trace constante (C_i fixé). En maintenant la trace, donc le gain d'adaptation constant, on réalise une bonne poursuite des paramètres.
La figure I.14 illustre la procédure d'identification connectée au robot manipulateur de classe 4 (C4).

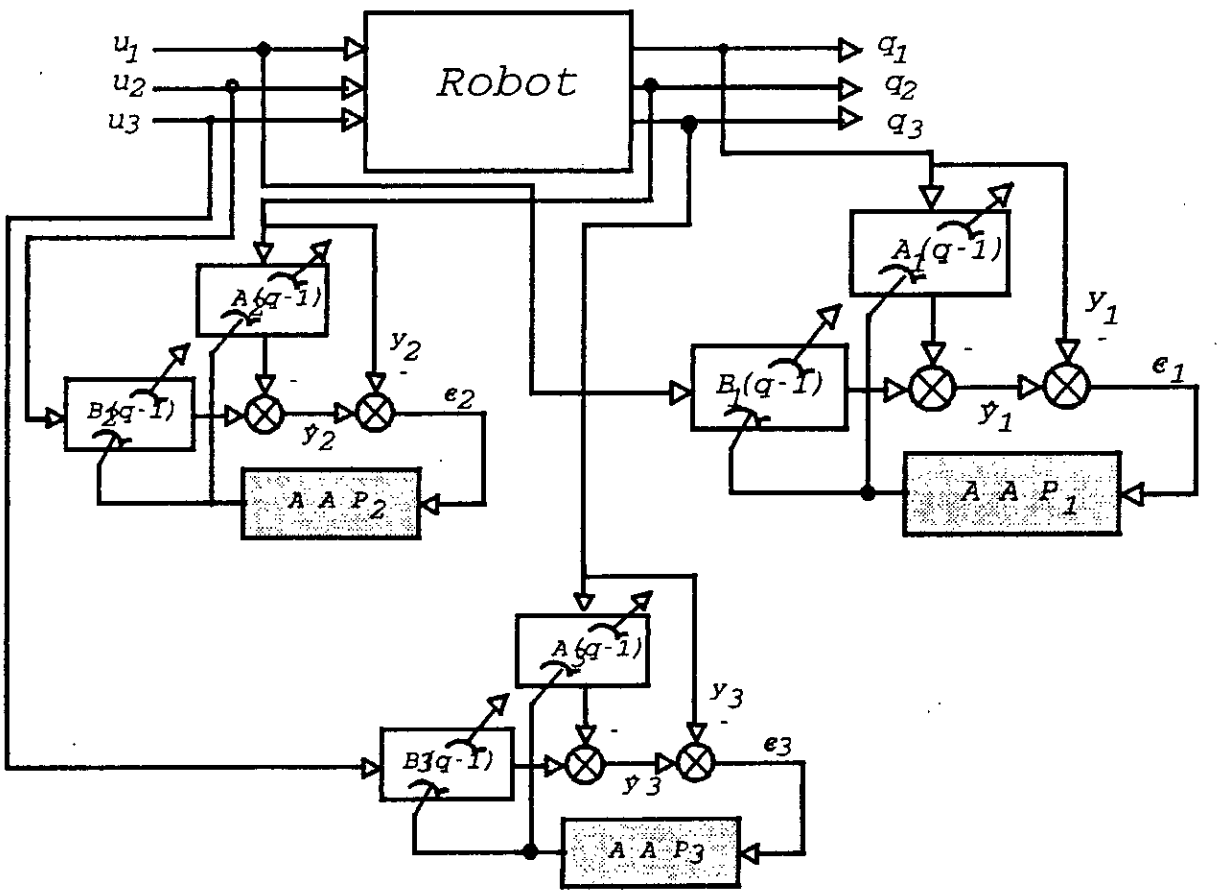


Figure I.14 Structure de l'algorithme d'identification.
(AAP: Algorithme d'Adaptation Paramétrique)

I.3.4.a Résultats de simulation.

Dans cette partie on s'intéresse à déterminer le modèle structurel 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:

$$n_1=2;\quad m_1=2;\quad d_1=0;$$
$$n_2=2;\quad m_2=2;\quad d_2=0.$$

En prenant le modèle de connaissance du robot, 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 Gi=5x10000 et λ_1/λ_2=1. L'excitation du robot est une séquence binaire pseudo aléatoire (SBPA) de longueur 8. La période d'échantillonnage est de 10ms. L'évolution des sorties ainsi que les erreurs sont consignées sur la figure I.15.a. L'évolution, dans le temps, des paramètres est représentée dans la figure I.15.b.

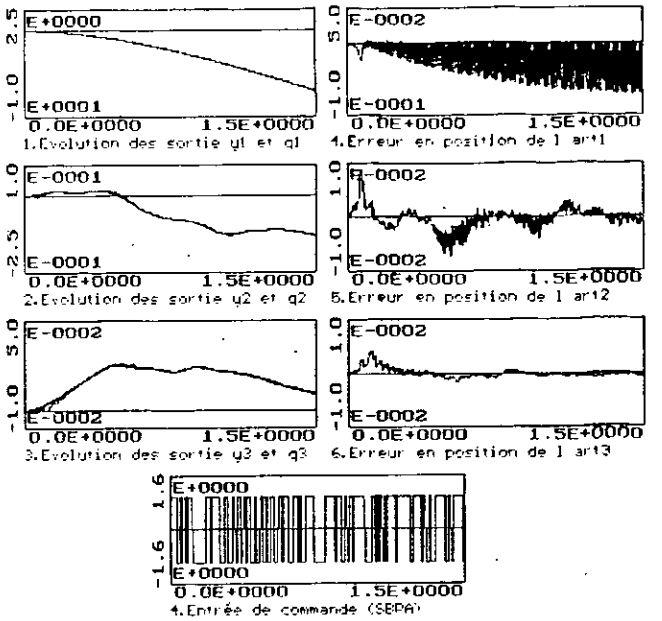


Figure I.15.a Evolution des sorties estimées y et reelles q et des erreurs d'identification (y-q).

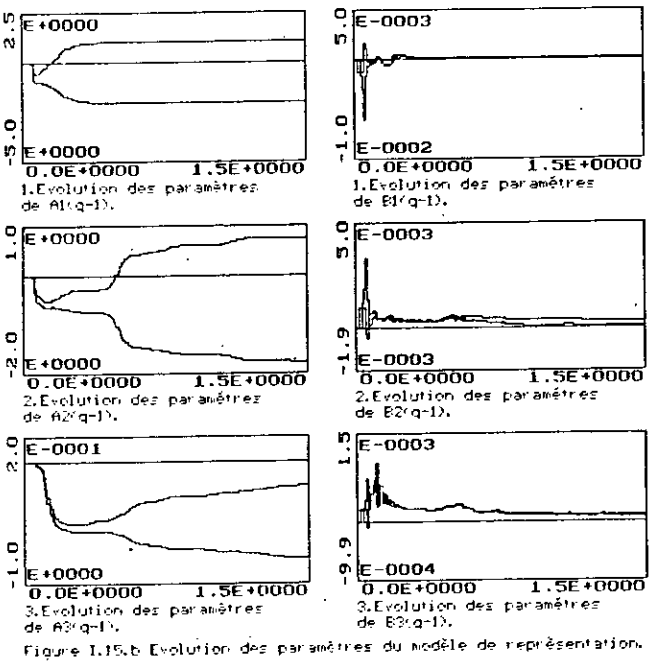


Figure I.15.b Evolution des paramètres du modèle de représentation.

Conclusion.

La modélisation cinématique des robots manipulateurs 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 se présentent dans la vitesse ou l'accélération, cela nécessite une interpolation polynomiale sur la trajectoire généralisée.

Le modèle de connaissance dynamique, établi par le formalisme d' Euler-Lagrange, 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.

La structure du modèle de représentation est choisi comme étant linéaire et découplé. L'ordre des équations linéaires est déterminé par la connaissance du modèle du robot et est égale à deux.
La détermination du modèle aux différences sera effectuée par l'algorithme des moindres carrés recursifs à trace constante.

Chapitre II

LA COMMANDE ADAPTATIVE AUTO-AJUSTABLE

Notre ennemi dans l'étude c'est la suffisance;
quiconque veut réellement apprendre doit commencer par s'en débarrasser.
"S'instruire sans jamais s'estimer satisfait"
et
"Enseigner sans jamais se lasser"
telle doit être notre attitude.

MAO TSE-TOUNG

Chapitre II

LA COMMANDE ADAPTATIVE AUTO-AJUSTABLE

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.
Dans de nombreux domaine (robotique) 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 approximés et réels devient non négligeable.
Les développement récents de la mini et de la micro-informatique rendent possible l'implémentation de lois de commande complexes qui demandent un traitement substentiel. Ce développement a favorisé l'émergence de différentes méthodes d'identification et de commande de processus en temps réel.

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 [13] et Astrom & Wittenmark [14]. Depuis, plusieurs extensions ont apparu, tels que placement de pôles et de zéros[15], variance minimale généralisée[16] et LQG [17](Linear Quadratic Gaussian).

Le régulateur auto-ajustable à STR (Self Tunning Regulator) est basé sur l'estimation en temps réel des paramètres du système ou ceux du régulateur. La commande est calculée en utilisant les paramètres estimés. C'est ce qu' on appelle le principe de l'équivalence certaine[18].
On distingue deux approches de commande auto-ajustable, la commande adaptative directe et indirecte.
Les schémas indirects comportent deux étapes à chaque période d'échantillonnage. Dans une première phase on identifie de manière récursive les paramètres du modèle, puis une deuxième phase, on calcule les paramètres du régulateur à partir des paramètres du procédé .
Les schémas directs qui ne comportent qu'une seule étape à chaque période d'échantillonnage. Les paramètres du régulateur sont directement identifiés de manière récursive. Dans ce cas, on identifie en fait implicitement le procédé mais reparamétrisé en terme de prédicteur.
La commande adaptative directe est plus adaptée au cas des systèmes à minimum de phase[13]. Le temps de calcul d'une telle commande est très réduit, par contre la connaissance à priori sur le procédé pour le mettre en oeuvre est importante[19]. La commande adaptative indirecte permet de traiter le cas des systèmes à non minimum de phase[20].

Le modèle dynamique des robots manipulateurs est fortement non linéaire[1,21]. Le choix d'un modèle de représentation adéquat permet l'implémentation des algorithmes auto-ajustables pour la commande des bras manipulateurs[1,22,23].
Dans un premier lieu, on présente l'algorithme auto-ajustable (STR) à Variance Minimale explicite et implicite. L'algorithme à Variance Minimale généralisée est

exposé dans la section II.2. En second lieu, on développe respectivement l'algorithme de **Poursuite et Régulation avec Pondération de l'Entrée** et celui à **Placement de Pôles** dans les sections II.3 et II.4. Le paragraphe II.5 expose l'algorithme STR à **Placement de Pôles et de Zéros**. La version implicite et explicite des différents algorithmes est développée dans chaque section. Le paragraphe II.6 est consacré à la commande **optimale à critère quadratique à un pas (LQC Linear Quadratic Controller)**. Une synthèse des différents algorithmes est traitée à la section II.7. L'extension des algorithmes au cas multivarible est exposé en détail dans le paragraphe II.8. La section II.9 présente l'algorithme à STR **multirate et mixte**. Les résultats de simulation des différents algorithmes appliqués aux robots de classe 4, sont exposés dans la section II.10. Une conclusion générale clôture le chapitre.

II.1 Commande à variance minimale.

La commande à variance minimale, initialement introduite par Aström[13], consiste à minimiser la variance de la sortie. Les objectifs de la commande, dans un environnement stochastique, seront liés à la minimisation de l'effet de la perturbation sur la poursuite d'une trajectoire ou sur les variations de la sortie dans le cas de la Régulation[24]. Dans le cas pratique cette commande nécessite une importante énergie. Une solution à ce problème est d'introduire une dynamique de la sortie prédite [25], cette dynamique est liée au polynôme de perturbation [26].

Le modèle du processus est le modèle ARMAX (Auto-Regressive Mooving average Exogen) défini par [27]:

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t)+C(q^{-1})e(t)+h \qquad (II.1)$$

Avec $A(q^{-1})=1+a_1q^{-1}+\ldots+a_nq^{-n}$ *noté* A.
$B(q^{-1})=b_1q^{-1}+\ldots+b_mq^{-m}$ *noté* B.
$C(q^{-1})=1+c_1q^{-1}+\ldots+c_lq^{-l}$ *noté* C.
d: retard du système (≥ 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.
$e(t)$: bruit blanc de moyenne nulle et de variance σ^2 .
Les polynômes B *et* C sont stables [13][27].

Le critère à minimiser est:

$$J = E\left[Py(t+d+1)-R_w w(t+d+1)\right]^2 \qquad (II.2)$$

Où E: esperance mathématique.
$w(t+d+1)$: représente le signal de référence.

$P(q^{-1})$ *et* $R_w(q^{-1})$: polynômes de pondération avec $p(q^{-1})=\dfrac{P_N(q^{-1})}{P_D(q^{-1})}$.

Les degrés de P *et* R_w peuvent être choisis arbitrairement. On remarque dans le critère que $w(t+d+1)$ est une information disponible, mais $y(t+d+1)$ ne l'est pas. C'est une information future, donc il faut la prédire. Clarke[28] propose la prédiction optimale au sens des moindres carrés dans le cas où P est une fraction polynomiale.
Cherchons donc la prédiction optimale au sens des moindres carrés de $Py(t+d+1)$.
De l'équation (II.1) on a:

$$P\,y(t+d+1) = \frac{P\,B^*}{A}u(t) + \frac{P\,C}{A}e(t+d+1) + q^{d+1}\frac{P}{A}h \qquad (II.3)$$

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

Dans cette équation, on remarque que $Py(t+d+1)$ est fonction de $u(t)$ et de $e(t+d+1)$. Le problème est alors d'exprimer $Py(t+d+1)$ en fonction des données futures de $e(t)$. Ceci revient à séparer les informations futures et les informations actuelles (disponibles). Ce qui s'exprime par une division euclidienne de $P_N C$ par $P_D A$ jusqu'à l'ordre d. D'où on a:

$$\frac{P_N C}{P_D A} = S' + q^{-(d+1)}\frac{R}{P_D A} \qquad (II.4)$$

Où $S'(q^{-1}) = s'_0 + s'_1 q^{-1} + \ldots + s'_d q^{-d}$,
$R(q^{-1}) = r_0 + r_1 q^{-1} + \ldots + r_{n_r-1} q^{-n_r+1}$; $n_r = n + n_d - 1$.

D'où l'équation Diophantine suivante:

$$A\,P_D\,S' + q^{-(d+1)}R = P_N\,C \qquad (II.5)$$

Le calcul de S' et de R, se fait par résolution d'un système triangulaire inférieur [29][30] (Voir annexe 4).
En utilisant (II.4), l'équation (II.3) devient:

$$P\,y(t+d+1) = \frac{P_N\,B^*}{P_D A}u(t) + \frac{R}{P_D A}e(t) + q^{d+1}\frac{P_N}{P_D A}h + S'e(t+d+1) \qquad (II.6)$$

Soit $\psi(t+d+1) = P\,y(t+d+1)$ $(II.7)$.
Cherchons alors la meilleur prédiction de $\psi(t+d+1)$ au sens des moindres carrées. Soit $\psi^*(t+d+1)$ cette prédiction. Il s'agit donc de minimiser le critère J_1 défini par:

$$J_1 = E\left[\psi(t+d+1) - \psi^*(t+d+1)\right]^2 \qquad (II.8)$$

En utilisant l'équation (II.6) on a :

$$J_1 = E\left[\left\{\frac{P_N B^*}{P_D A}u(t) + \frac{R}{P_D A}e(t) + q^{d+1}\frac{P_N}{P_D A}h - \psi^*(t+d+1)\right\}^2\right] + \left(\sum_{i=0}^{d} s'^{\,2}_i\right)\sigma^2 +$$
$$2E\left[\frac{P_N B^* S'}{P_D A}u(t)e(t+d+1) + \frac{R\,s'}{P_D A}e(t)e(t+d+1)\right.$$
$$\left. + q^{d+1}\frac{P_N S'}{P_D A}he(t+d+1) - S'\psi^*(t+d+1)e(t+d+1)\right] \qquad (II.9)$$

Or e(t)est un bruit blanc dont les caractéristiques sont [13]:

$$\phi_{xx}(0) = E[e^2(t)] = \sigma^2.$$
$$\phi_{xx}(\tau) = E[e(t)e(t+\tau)] = 0; \quad pour \quad \tau = 0 \qquad (II.10)$$
(*fonction d'autocorrélation*).
$$\phi_{xy}(\tau) = E[e(t)x(t+\tau)] = 0; \quad \forall\tau \geq 0.$$
(*fonction d'intercorrélation*).

d'où :

$$J_1 = E\left[\left\{\frac{P_N B^*}{P_D A}u(t)+\frac{R}{P_D A}e(t)+q^{d+1}\frac{P_N}{P_D A}h-\psi^*(t+d+1)\right\}^2\right]+\left(\sum_{i=0}^{d} s'^{\,2}_i\right)\sigma^2 \quad (II.11).$$

J_1 est minimale si le premier terme est nulle, d'où :

$$\psi^*(t+d+1)=\frac{P_N B^*}{P_D A}u(t)+\frac{R}{P_D A}e(t)+q^{d+1}\frac{P_N}{P_D A}h \qquad (II.12)$$

C'est la prédiction optimale de $\psi(t+d+1)$.
Le critère défini par (II.2) devient :

$$J = E\left[\psi^*(t+d+1)+\xi(t+d+1)-R_w w(t+d+1)\right]^2 \qquad (II.13)$$

Où $\xi(t+d+1)=S'e(t+d+1)$.

Avec l'hypothèse que $e(t)$ n'est corrélé qu'avec lui même, on a:

$$J=E\left[\psi^*(t+d+1)-R_w w(t+d+1)\right]^2+\sum_{i=0}^{d} s_i'^{\,2}\sigma^2 \qquad (II.14)$$

Calculons $u(t)$ qui minimise J:

$$\frac{\delta J}{\delta u_t}=2E\left[\psi^*(t+d+1)-R_w w(t+d+1)\right]\frac{\delta\psi^*(t+d+1)}{\delta u_t}=0 \qquad (II.15)$$

De l'équation (II.12) on a : $\dfrac{\delta\psi^*(t+d+1)}{\delta u_t}=p_{N_0}b_1$; où p_{N_0}: premier coefficient de $P_N(q^{-1})$.

donc $$\psi^*(t+d+1)-R_w w(t+d+1)=0 \qquad (II.16)$$

Calculons maintenant $\psi^*(t+d+1)$ en fonction de $y(t)$ et $u(t)$. L'équation (II.1) donne:

$$e(t)=\frac{A}{C}y(t)-q^{-(d+1)}\frac{B^*}{C}u(t)-\frac{h}{C} \qquad (II.17)$$

De l'équation (II.17) et (II.12) ainsi que (II.5) on a:

$$\psi^*(t+d+1)=\frac{R}{P_D C}y(t)+\frac{S'B^*}{C}u(t)+q^{d+1}\frac{S'}{C}h \qquad (II.18)$$

On pose $S'B^*=S$. L'équation (II.16) donne la commande à variance minimale suivante:

$$U(t) = \frac{P_D C R_w w(t+d+1)-Ry(t)-P_d(1)S'(1)h}{P_d S} \qquad (II.19)$$

Remarque.

La commande optimale u(t) peut être obtenu en minimisant le critère J' donné par [28]:

$$J' = E\left[(\phi(t+d+1))^2\right] \tag{II.20}$$

Où $\Phi(t+d+1) = Py(t+d+1) - R_w w(t+d+1)$. □

La structure du régulateur est alors:

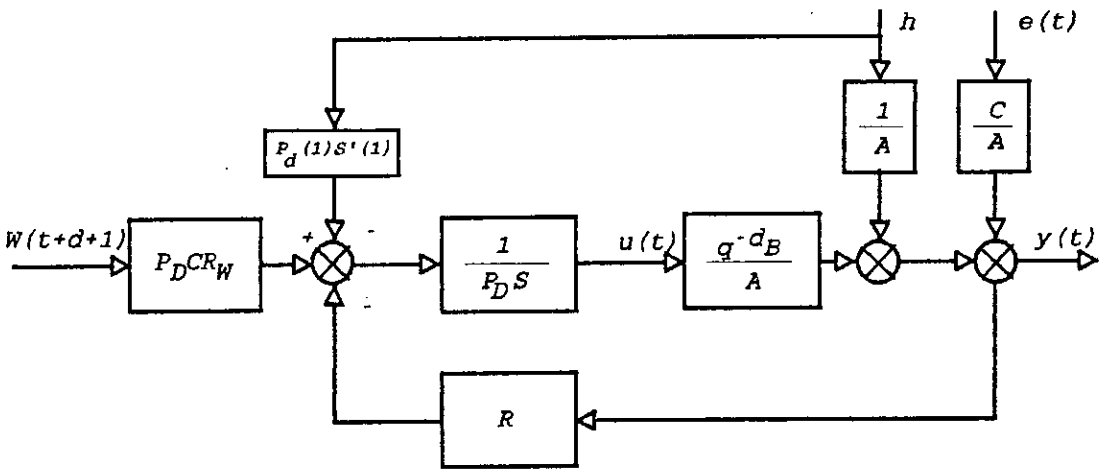


Figure II.1 Architécture du réglage à VM

Cette structure correspond bien à la structure canonique *RST* [24][29] où R est le même, $S=P_D S$ *et* $T=P_D C R_w$.

II.1.1 Calcul de la fonction de transfert en boucle fermée.

Pour évaluer les performances du régulateur utilisé, on est amené à voir le comportement du système en boucle fermé. Des équations (II.1), (II.5) et (II.19) on obtient:

$$y(t) = \frac{P_D R_w}{P_N} w(t) + \frac{P_D S'}{P_N} e(t) \tag{II.21}$$

L'écart entre la sortie et la modèle est un bruit défini par:

$$\xi_1(t) = \frac{P_D S'}{P_N} e(t) \tag{II.22}$$

Soit $P'_D = P_d S' = p'_{D_0} + P'_{D_1} q^{-1} + \ldots + p'_{D_{n_d+d}} q^{-(n_d+d)}$. L'équation (II.22) devient:

$$\xi_1(t) = -\frac{1}{P_{N_0}} \sum_{i=1}^{n_N} P_{N_i} \xi_1(t-i) + \sum_{i=0}^{n_D+d} P'_{D_i} e(t-i) \tag{II.23}$$

D'où :

$$E\left[\xi_1^2(t)\right] \le \frac{1}{P_{N_0}^2} E\left[\left\{\sum_{i=1}^{n_N} P_{N_i} \xi_1(t-i)\right\}^2 + \left\{\sum_{i=0}^{n_D+d} P'_{D_i} e(t-i)\right\}^2\right] \tag{II.24}$$

Alors :

$$E\left[\xi_1^2(t)\right] \leq \left[1-\frac{1}{P_{N_0}^2}\left(\sum_{i=1}^{n_N} P_{N_i}^2\right)\right]^{-1}\left(\sum_{i=0}^{n_D+d} P^2\sigma^2\right) \qquad (II.25)$$

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

Dans ce cas on a évalué la borne maximale de la variance de la sortie, qui peut caractériser les performances en BF. Dans le cas où la condition imposée sur l'inéquation (II.25) n'est pas réalisée, on est amené à calculer la fonction d'autocorrélation à l'origine [31] :

$$r(\tau) = \frac{1}{2\pi}\int S(\omega)\exp(j\omega\tau)\,d\omega \qquad (II.26)$$

Où $S(\omega)$: Densité spectrale d'énergie.

On définit aussi la fonction d'autocorrelation [31] :

$$r(o) = \frac{\sigma^2}{2\pi}\int H(\exp(-j\omega))H(\exp(j\omega)\,d\omega \qquad (II.27)$$

Avec $H(q^{-1}) = \frac{P_D S'}{P_N}$.

D'où

$$r(o) = \frac{\sigma^2}{2\pi}\int \frac{P_D(\exp(-j\omega))\,P_D(\exp(j\omega)\,S'(\exp(-j\omega))\,S(\exp(i\omega))}{P_N(\exp(-j\omega))\,P_N(\exp(j\omega))}\,d\omega \qquad (II.28)$$

C'est la variance du bruit de sortie, dont sa moyenne est nulle, car $H(q^{-1})$ est linéaire et $e(t)$ un bruit blanc de moyenne nulle [31].
On remarque de l'équation (II.21) que la fonction de transfert en BF, entre la sortie et la référence $w(t)$, possède une dynamique définie par:

$$F_{BF}(q^{-1}) = \frac{P_D R_w}{P_N} \qquad (II.29)$$

Alors le choix des polynômes de pondération $P(q-1)$ *et* $R_w(q^{-1})$ definit le modèle de référence à suivre. La référence $w(t+d+1)$ peut être générée à partir d'un modèle choisi, de la forme suivante [24]:

$$w(t) = q^{-(d+1)}\frac{B_m}{A_m}r(t) \qquad (II.30)$$

A_m *et* B_m: polynômes choisis. $r(t)$: entrée de référence.

Dans le cas où $P=R_w=1$, le modèle en boucle fermée est défini par les polynômes A_m *et* B_m.

II.1.2 Structure du régulateur adaptative.

Dans le cas des systèmes à paramètres variables dans le temps, l'estimation récursive du modèle s'impose. L'algorithme à moindres carrées étendu donne une estimation des paramètres des polynômes A, B et C ainsi que h [24][27][29].
Les informations nécessaires sur le système pour implémenter une telle commande sont [13][16]:

i) d: retard du système connu.
ii) La borne maximale de n, m et l connue.
iii) $C(q^{-1})$ et $B(q^{-1})$: polynômes stables.

Le principe de l'équivalence certaine [32][33] permet l'implémentation de l'algorithme en utilisant les estimés des paramètres du modèle. On distingue deux algorithmes de commande adaptative à variance minimale. La commande adaptative explicite ou indirecte, qui utilise les paramètres du système pour calculer ceux du régulateur et enfin calculer la commande [13]. La commande adaptative implicite ou directe, qui estime directement les paramètres du régulateur et calcule la commande [34][36], nécessite la construction d'un modèle de prédiction [27].

a- Algorithme à VM explicite.

Données: Spécifier $P(q^{-1})$, $R_w(q^{-1})$, d, n, m et l.
Etape 1: Estimation de A, B, C et h en utilisant l'algorithme MCR à trace constante (Chap I.3.4) avec:

$$\theta^T = [a_1 \ldots a_n \; b_1 \ldots b_m \; c_1 \ldots c_l \; h].$$
$$\phi^T(t) = [-y(t-1) \ldots -y(t-n) \; u(t-d-1) \ldots u(t-d-m) \; \varepsilon(t-1) \ldots \varepsilon(t-l) \; 1].$$
$$\varepsilon(t) = y(t) - \theta^T \phi(t).$$

Etape 2: Génération de $w(t)$.
Etape 3: Résolution de l'équation Diophantine (II.5).
Etape 4: Calcul de la commande (équation II.19) .
t=t+1 revenir à étape 1.

b- Algorithme à VM implicite.

Pour pouvoir appliquer facilement cet algorithme, on utilise l'équation (II.18) pour construire le prédicteur, avec la transformation suivante [28]:

$$\psi^*(t+d+1) = \frac{R}{C} y_f(t) + \frac{S}{C} u(t) + \frac{\delta}{C} \qquad (II.31)$$

Où $S = S'B^*$; $y_f(t) = \frac{y(t)}{P_D}$ et $\delta = S'(1)h$.
C'est le prédicteur avec lequel on estime C, R, S et δ d'une façon directe, sans résolution de l'équation Diophantine. La commande définie par l'équation (II.19) devient [28]:

$$U(t) = \frac{CR_w w(t+d+1) - Ry_f(t) - \delta}{S} \qquad (II.32)$$

D'où l'algorithme suivant:

Données: Spécifier $P(q^{-1})$, $R_w(q^{-1})$, d, n, m et l.
Etape 1: Estimation des paramètres du régulateur en utilisant l'algorithme MCR à trace constante (Chap I.3.4) avec:

$$\theta^T = [c_1 \ldots c_l \; r_0 \ldots r_{n_r-1} \; s_0 \ldots s_{n_s} \; \delta].$$
$$n_r = n+n_D-1; \; n_s = m+d-1.$$

$$\phi^T(t) = [-\phi^*(t-1) \ldots -\phi^*(t-l) \; y_f(t-d-1) \ldots y_f(t-d-n_r) \; u(t-d-1) \ldots u(t-d-n_s) \; 1].$$

$$\varepsilon(t) = Py(t) - \theta^T \phi(t).$$

Etape 2: Génération de w(t).

Etape 3: Calcul de la commande (équation II.32) .

t=t+1 revenir à étape 1.

Dans cette approche, le polynôme T (forme canonique, voir Chap II.3) est le produit de C et R_w. La sortie y est filtrée à travers P_D. Pour éviter ce produit et ce filtrage, on choisit $P_D=R_w=1$ [28][35].

Remarque 1.

Dans le cas où le système est à phase non minimale, le polynôme S est instable, ce qui engendre une divergence de la commande définie par (II.19 et II.32).

M'Saad [20][36] expose une méthode de stabilisation des zéros d'un système dans le cas où ces derniers sont instables. La structure d'un tel algorithme est la suivante:

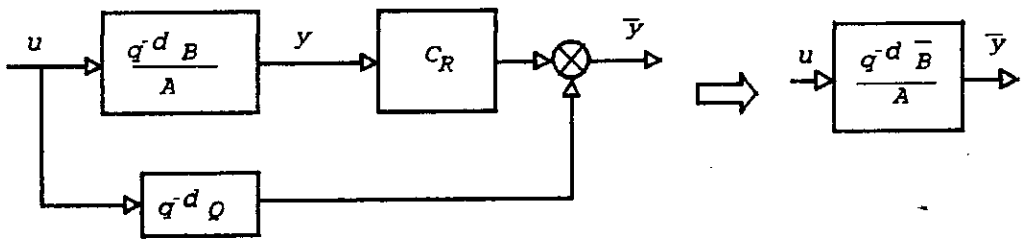


Figure II.2 Structure de la méthode de stabilisation des zéros.

Avec $\bar{B}=C_RB+QA$; C_R et Q polynômes en q^{-1}.

Le polynôme $\bar{B}$ est choisi arbitrairement de telle sorte qu'il soit stable. Cette équation est similaire à l'équation Diophantine (Annexe 5). Ainsi on peut appliquer l'algorithme à VM au nouveau système défini par:

$$A\bar{y}=q^{-d}\bar{B}u \qquad (II.33)$$

Pour un choix de $R_w=C=P=1$, on aboutit ainsi à la structure donnée par M'Saad [20].

De la figure II.2 on a:

$$\bar{y}=C_Ry+q^{-d}Qu \quad d'où \quad u=\frac{\bar{y}-C_Ry}{q^{-d}Q} \qquad (II.34)$$

Cette équation définit la deuxième structure donnée dans la figure II.3. Ainsi l'algorithme à VM peut être utilisé, soit pour les systèmes à phase minimale où à phase non minimale.□

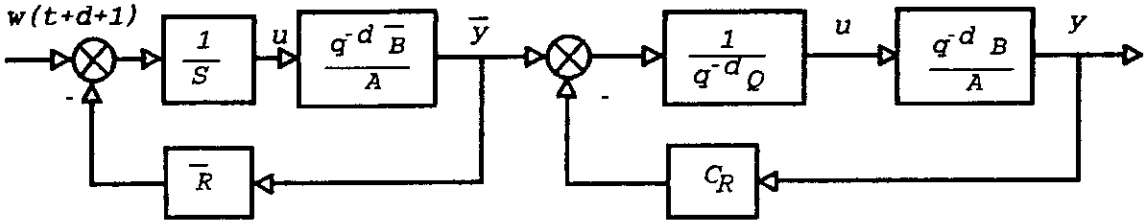


Figure II.3 Structure globale de la commande.(Réglage augmentée).

Remarque 2.

L'inconvénient de la méthode est qu'elle est mal adaptée au cas adaptative (approche directe) et nécessite la résolution de deux équations Diophantines, ce qui exige un temps de calcul plus important.□

II.2 Commande à variance minimale généralisée.

Clarke [37][38] montre l'effet d'un échantillonnage rapide sur les systèmes de degré relatif supérieur ou égal à deux. Ainsi qu'un choix d'une grande période d'échantillonnage et un temps de retard non entier. Toutes ces conditions engendrent un système à phase non minimale, même si celui-ci est à phase minimale dans le cas continu.

La commande à **variance minimale généralisée** a été introduite par Clarke [16] pour la commande des système à phase non minimale. Gawthrop [35] donne quelques interprétations de cette méthode. Enfin Clarke [28] donne un algorithme simple de la méthode GMV (Generalized Minimum Variance).
Le modèle du système est défini par l'équation (II.1). Le critère de commande est [28]:

$$J = E\left[\left(Py(t+d+1) - R_w w(t+d+1)\right)^2 + \left(Q' u(t)\right)^2\right] \qquad \text{(II.35)}$$

Où P *et* R_w sont définis par l'équation (II.2.).

Et $Q'(q^{-1}) = \dfrac{Q'_N(q^{-1})}{Q'_D(q^{-1})}$.

En procédant de la même façon que précédemment, le critère (II.35) devient:

$$J = E\left[\left(\psi^*(t+d+1) - R_w w(t+d+1)\right)^2 + \left(Q' u(t)\right)^2\right] + \sum_{i=0}^{d} s_i'^2 \sigma^2 \qquad \text{(II.36)}$$

Où S' est défini par l'équation (II.4) et $\psi^*(t+d+1)$ par (II.12).

La minimisation de J $\left(\dfrac{\partial J}{\partial u_t} = 0\right)$ donne:

$$\psi^*(t+d+1) - R_w w(t+d+1) + Qu(t) = 0 \qquad \text{(II.37)}$$

Où $Q(q^{-1}) = \dfrac{q'_{N_0}}{q'_{D_0} P_{N_0}} Q'(q^{-1})$.

avec q'_{N_0}, Q'_{D_0} et P_{N_0} sont respectivement les premiers coefficients de Q'_N, Q'_D et P_N.

En utilisant l'équation (II.18) et (II.37), la commande optimale à GMV est:

$$u(t) = \frac{P_D C R_w w(t+d+1) - Ry(t) - P_D(1) S'(1) h}{P_D(S+QC)} \qquad (II.38)$$

II.2.1 Calcul de la fonction de transfert en BF.

En utilisant les équations (II.1), (II.5) et (II.38) on obtient:

$$y(t) = \frac{B^* R_D}{PB^*+QA} w(t) + \frac{S+QC}{PB^*+QA} e(t) + \frac{Q}{PB^*+QA} h \qquad (II.39)$$

$$ou \quad y(t) = \frac{B^* R_D P_D}{P_N B^*+P_D QA} w(t) + \frac{S+QC}{P_N B^*+P_D QA} e(t) + \frac{P_D Q}{P_N B^*+Q P_D A} h \qquad (II.40)$$

La dynamique en boucle fermée est spécifiée par le polynôme :

$$P_N B^*+P_D QA = T_{BF} \qquad (II.41)$$

Ainsi le système est stable en BF, par un choix adéquat de P et Q même si B^* est instable. Mei-Hua & Wei [23] 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 :

$$PB^*+QA = T_{BF} \qquad (II.42)$$

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

II.2.2 Structure du régulateur à GMV.

La structure d'un tel régulateur est défini par la figure II.4.

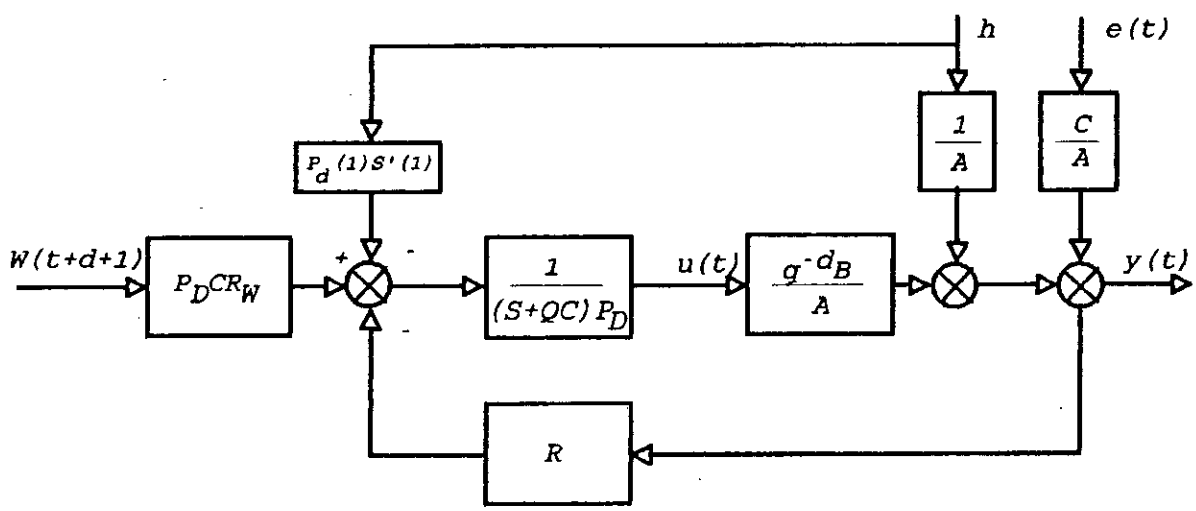


Figure II.4 Structure du régulateur à GMV.

On remarque que la Pondération Q, stabilise le polynôme S. De l'expression de la fonction de transfert en BF (équation II.39), on remarque que l'écart entre la sortie et le modèle est un bruit augmenté d'un filtrage de la perturbation constante h. L'effet de cette dernière peut être éliminé en choisissant :

$$Q(q^{-1}) = (1-q^{-1})\frac{Q_N}{Q_D} \qquad (II.42)$$

Ainsi $P_D(1)Q(1)h$ est nulle. Donc l'effet d'une perturbation constante est éliminé par un choix de $Q(q^{-1})$ multiple de $(1-q^{-1})$.

II.2.3 Algorithme de réglage adaptatif.

Pour les deux cas implicite et explicite, l'algorithme à GMV est défini respectivement dans les sections II.1.2.a et b. Dans le cas de l'approche directe la commande est définie par :

$$u(t) = \frac{CR_w w(t+d+1) - Ry_f - \delta}{(S+QC)} \qquad (II.43)$$

Tandis que pour l'approche indirecte la commande est :

$$u(t) = \frac{R_w w(t+d+1) - \psi^*(t+d+1)}{Q} \qquad (II.44)$$

Remarque.

Pour simplifier les calculs on choisit [29]:

$$Q(q^{-1}) = \frac{\lambda(1-q^{-1})}{(1+\alpha q^{-1})C(q^{-1})}\ ;\ P_D=1;\ R=1.$$

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

L'équation en BF définie par (II.40) devient :

$$y(t) = \frac{B^*}{P_N B^* + QA} w(t) + \frac{S+Q'}{P_N B^* + QA} e(t) \qquad (II.45)$$

Où $Q'=QC$.

Ainsi on évite les multiplications polynomiales et le filtrage de la sortie $y(t)$, lors du calcul de la commande en temps réel.□

II.3 Poursuite et Régulation à objectifs indépendants.

Dans cette approche, on s'intéresse à réaliser deux objectifs: la poursuite et la Régulation [26][24]. 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 suivants est vérifié:

- La présence de perturbations qui affectent le processus, ce qui nécessite le maintient 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** [29].

II.3.1 Calcul de la commande.

L'approche naturelle, dans la génération de la commande est que celle-ci soit une combinaison des composantes de la consigne $(w(k), w(k-1), \ldots)$ et de la sortie $(y(k), y(k-1)\ldots)$ ainsi que des commandes précédentes $(u(k-1), u(k-2)\ldots)$, donc on a [26]:

$$u(k) = \frac{1}{s_0'}\left(\sum_{i=1}^{s} s_i' q^{-i} u(t) + \sum_{i=1}^{t} t_i q^{-i} w(t) + \sum_{i=0}^{r} r_i' q^{-i} y(t)\right) \qquad (II.46)$$

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

$$u(t) = \frac{T(q^{-1})\, w(t) - R(q^{-1})\, y(t)}{s(q^{-1})} \qquad (II.47)$$

Avec $T(q^{-1}) = t_0 + t_1 q^{-1} + \ldots + t_t q^{-t}$ *noté* T.
$R(q^{-1}) = r_0 + r_1 q^{-1} + \ldots + r_r q^{-r}$ *noté* R.
$S(q^{-1}) = s_0 + s_1 q^{-1} + \ldots + s_s q^{-s}$ *noté* S.

Ceci conduit à la structure canonique du régulateur numérique représentée par la figure II.5.

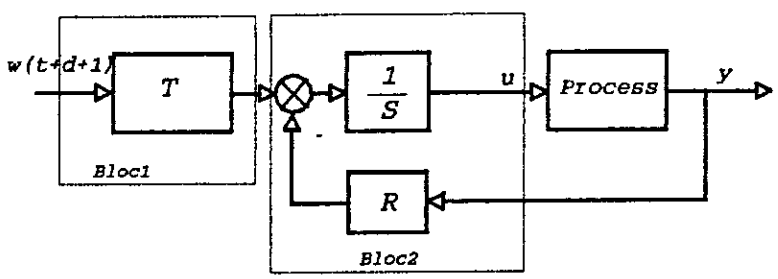


Figure II.5 Structure canonique de réglage RST.

Les polynômes R *et* S caractérisent la régulation, car il 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 dernier. Le modèle du processus est un modèle **DARMA** (Deterministic ARMA), défini par [27]:

$$A(q^{-1})\, y(t) = q^{-d}\, B(q-1)\, u(t) + h \qquad (II.48)$$

Avec $A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n}$. *noté* A
$B(q^{-1}) = b_1 q^{-1} + \ldots + b_m q^{-m}$. *noté* B
d: retard du système (≥ 0).
q^{-1}: Opérateur retard défini par: $q^{-1} y(t+1) = y(t)$.
h: perturbation constante (déterministe).
$y(t)$: sortie du processus.
$u(t)$: u(t) entrée de commande.

Les équations (II.47) et (II.48) donnent la fonction de transfert en boucle fermée (BF) définie par :

$$y(t) = \frac{q^{-d}BT}{AS+q^{-d}BR}w(t)+\frac{S}{AS+q^{-d}BR}h \qquad (II.49)$$

II.3.2 Calcul des polynômes de Régulation R et S.

Le bloc 2 de la figure II.5 montre bien la structure de régulation, car il prend en compte l'évolution de la sortie, pour calculer la commande. La fonction de transfert de ce bloc est :

$$y(t) = \frac{q^{-d}B}{AS+q^{-d}BR}w_1(t)+\frac{S}{AS+q^{-d}BR}h \qquad (II.50)$$

Où $w_1=Tw(t+d+1)$.
Les performances désirées sont un choix des pôles de la fonction de transfert en BF et une simplification des zéros du procédé ($B(q^{-1})$) pour assurer la poursuite parfaite. Elles s'expriment par:

$$H(q^{-1}) = \frac{y(t)}{w_1(t)}=\frac{q^{-(d+1)}}{p(q^{-1})} \qquad (II.51)$$

On choisit alors:

$$S = B^*S' \qquad (II.52)$$

Où $B(q^{-1})=b_1+b_2q^{-2}+\ldots+b_mq^{-m+1}$.

L'équation (II.50) devient alors:

$$y(t) = \frac{q^{-(d+1)}B^*}{B^*(AS'+q^{-(d+1)}R)}w_1(t)+\frac{S'B^*}{B^*(AS'+q^{-(d+1)}R}h \qquad (II.53)$$

Une condition nécessaire surgit: B^* doit être stable (dans le cas contraire on introduit une Pondération). D'où le système en BF:

$$y(t) = \frac{q^{-(d+1)}}{(AS'+q^{-(d+1)}R)}w_1(t)+\frac{S'}{(AS'+q^{-(d+1)}R}h \qquad (II.54)$$

On choisit alors :

$$AS'+q^{-(d+1)}R = PP_o \qquad (II.55)$$

P: spécifie les en boucle fermée.
P_0: polynôme d'observation [27,30] (Voir annexe 5).

Les polynômes S' *et* R sont solutions de l'équation Diophantine (II.55) dont la résolution est en annexe 4 ($deg\ S'=d$, $deg\ R=n-1$) Donc:

Avec $R(q^{-1})=r_0+r_1q^{-1}+\ldots+r_{n_j1}q^{-n+1}$.
$S'(q^{-1})=s'_0+s'_1q^{-1}+\ldots+s'_dq^{-d}$.

II.3.3 Calcul du polynôme de poursuite $T(q^{-1})$.

Pour assurer une bonne poursuite, il faut que la relation qui lie

$w(t)$ et $y(t)$ soit avec la dynamique P, c'est à dire:

$$y(t) = \frac{w(t+d+1)}{P} + bruit \tag{II.56}$$

Il faut donc choisir $w_1 = P_o w(t+d+1)$. d'où:

$$T(q^{-1}) = P_o(q^{-1}) \tag{II.57}$$

Et la fonction de transfert en BF devient:

$$y(t) = \frac{1}{P} w(t+d+1) + \frac{S'}{PP_o} h \tag{II.58}$$

Remarque.

Dans le cas stochastique le polynôme d'observation est identique à $C(q^{-1})$ [27].□

En choisissant :

$$w(t+d+1) = \frac{R_w B_m}{B_m} y_d(t) \tag{II.59}$$

On a alors:

$$y(t) = \frac{R_w B_m}{P A_m} y_d(t) + \frac{S'}{PC} h \tag{II.60}$$

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:

$$A\Delta S_1' + q^{-(d+1)} R = PP_o \tag{II.61}$$

Où $S' = S_1' \Delta$ 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 [38].
Dans ce cas on introduit un retour sur le bloc $1/S$, pour le stabiliser [37]. La valeur de celui-ci est [29]:

$$Q(q^{-1}) = \frac{\lambda(1-q^{-1})}{(1+\alpha q^{-1})} \tag{II.62}$$

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

D'où le schéma de la commande suivant:

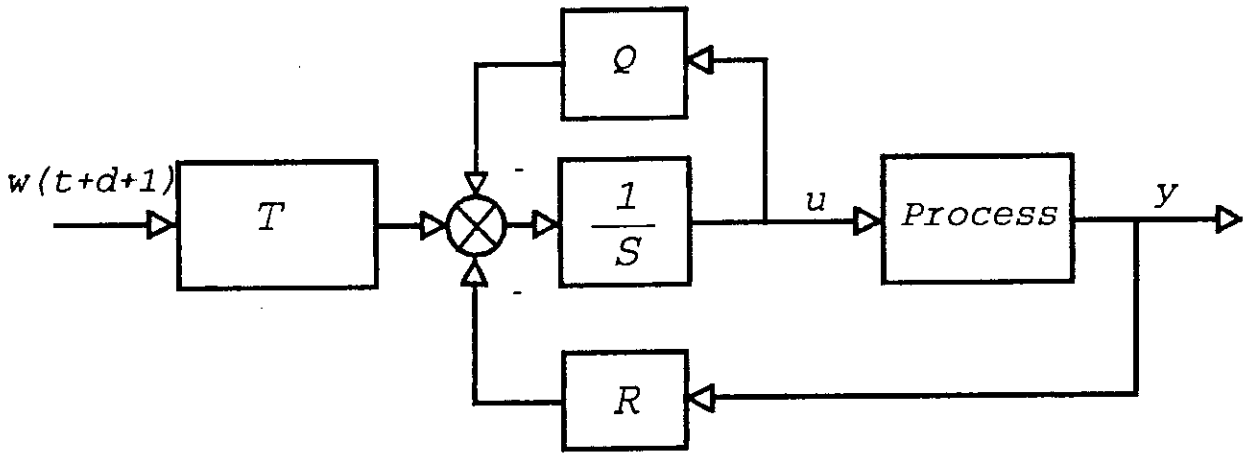


Figure II.6 Structure de réglage par poursuite et Régulation avec Pondération de l'entrée PRPE.

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

$$u(t) = \frac{Tw(t+d+1) - Ry(t)}{S+Q} \qquad \text{(II.63)}$$

Des équations (II.48), (II.61) et (II.63) on trouve:

$$y(t) = \frac{q^{-(d+1)}B^*P_o}{AQ+B^*PP_o}w(t+d+1) + \frac{S+Q}{AQ+B^*PP_o}h \qquad \text{(II.64)}$$

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.

II.3.3 *Structure de régulateur adaptative.*

La stratégie de commande par poursuite et Régulation est applicable au processus déterministe $(e(t)=0)$. Elle est mal adaptée au processus stochastique, à cause de la mauvaise connaissance du polynôme d'observation.

a- Approche explicite de cette stratégie.

Données: Spécifier $P(q^{-1})$, $P_o(q^{-1})$, $Q(q^{-1})$, d, n *et* m.
Etape 1: Estimation de A *et* B en utilisant l'algorithme MCR à trace constante (Chap I.3.4) avec:
$\theta^T = [a_1 \ldots a_n\ b_1 \ldots b_m]$.
$\phi^T(t) = [-y(t-1) \ldots -y(t-n)\ u(t-d-1) \ldots u(t-d-m)\ ]$.
$\varepsilon(t) = y(t) - \theta^T \phi(t)$.

Etape 2: Génération de $w(t)$.
Etape 3: Résolution de l'équation Diophantine (II.61).
Etape 4: Calcul de la commande (équation (II.47) ou (II.63)) .
t=t+1 revenir à étape 1.

b- Approche implicite de cette stratégie.

Dans cette approche on estime directement les paramètres du régulateur. Pour cela on utilise le modèle de prédiction [27]. La méthode de prédiction est la même calculée dans le cas de la VM, donc de l'équation (II.18) on a[27]:

$$\psi^*(t+d+1) = Ry_f(t)+Su_f(t) \qquad \text{(II.65)}$$

Où $\psi^*(t+d+1)=P(q^{-1})y(t+d+1)$; $y_f(t)=\dfrac{y(t)}{P_o(q^{-1})}$ et $u_f(t)=\dfrac{u(t)}{P_o(q^{-1})}$.

Ainsi avec ce prédicteur, on peut estimé directement R et S. L'algorithme de commande adaptative directe à PRPE est:

Données: Spécifier $P(q^{-1})$, $P_o(q^{-1})$, $Q(q^{-1})$ d, n et m.
Etape 1: Estimation des paramètres du régulateur en utilisant l'algorithme MCR à trace constante (Chap I.3.4) avec:

$$\theta^T = [r_0 \ldots r_{n-1}\ s_0 \ldots s_{m+d-1}\ ].$$
$$\phi^T(t) = [y_f(t-d-1) \ldots y_f(t-d-n)\, u_f(t-d-1) \ldots u_f(t-2d-m)].$$
$$\varepsilon(t) = Py(t)-\theta^T\phi(t).$$

Etape 2: Génération de w(t).
Etape 3: Calcul de la commande (équation (II.47) ou (II.63)) .
t=t+1 revenir à étape 1.

II.4 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 [15] se base sur la détermination des paramètres du régulateur de la structure général (canonique) [29], 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 [15]. L'estimation récursive des paramètres du processus engendre deux algorithme de commande adaptative: implicite et explicite.

II.4.1 Formulation du problème.

Le modèle du processus est le modèle DARMA [27] pour un système déterministe:

$$Ay(t) = q^{-d}Bu(t)+h.$$

Où A, B et h définis dans l'équation (II.48).

On désire trouver un controlleur 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 :

$$H_m(q^{-1}) = \frac{Q}{P'} \qquad (II.66)$$

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

La structure canonique du régulateur définie par l'équation (II.47) donne la fonction de transfert en BF suivante :

$$y(t) = \frac{q^{-d}BT}{AS+q^{-d}BR}w(t) + \frac{S}{AS+q^{-d}BR}h \qquad (II.67)$$

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 (II.67) soit identique au modèle désiré défini par (II.66) donc:

$$\frac{q^{-d}BT}{AS+q^{-d}BR} = \frac{Q}{P'} \qquad (II.68)$$

Du point de vue algébrique, le problème revient à trouver trois polynômes R,S *et* T qui vérifient l'équation (II.68). Le problème ainsi posé possède une infinité de solutions dont le choix de la solution adéquate revient à l'utilisateur. De l'équation (II.68), on peut dire qu'il y a une simplification d'un facteur, car le degré de P est normalement inférieur au degré de $AS+q^{-1}BR$.
Dans la théorie de l'espace d'état, on peut montrer que la structure de réglage définie par (II.47) correspond à une combinaison d'un observateur et d'un retour d'état [29,30](voir annexe6). Donc le facteur commun peut être interprété comme le polynôme d'observation P_0.

II.4.2 Calcul des polynômes R et S.

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

$$AS+q^{-(d+1)}B^*R = PP_o \qquad (II.69)$$

Avec $B^*(q^{-1}) = b_1+b_2q^{-1}+\ldots+b_mq^{-m+1}$.
$P(q^{-1}) = \prod_i (1+\alpha_i q^{-1})$; α_i: pôles désirés [30].

Cette équation est la forme canonique générale de l'équation Diophantine présentée dans l'équation (II.55).
En faisant un choix adéquat des degrés de R *et* S [29][94]:

$deg\ S = deg\ R = r_{max}-1$ (*II.70.a*).
$deg(PP_o) \leq 2r_{max}-1$ (*II.70.b*).
où $r_{max} = \max(n, m+d)$.

La résolution de l'équation (II.69) donne les polynômes de régulation R *et* S (Voir annexe 5).

II.4.3 *Calcul des polynômes de poursuite.*

De l'équation (II.68) et avec l'hypothèse de laisser les zéros de la fonction de transfert en boucle ouverte, on impose:

$$Q = Q_1B^* \qquad (II.71)$$

Donc

$$T = Q_1 P_o \tag{II.72}$$

Pour avoir une bonne poursuite, il faut que le comportement en régime permanent soit celui de la fonction définie par (II.66), d'où le choix du polynôme de poursuite:

$$T(q^{-1}) = \begin{cases} \dfrac{Q_1 P_o}{B^*(1)} & si \quad B^*(1) \neq 0. \\ Q_1 P_o & si \quad B^*(1) = 0. \end{cases} \tag{II.73}$$

On obtient ainsi la fonction de transfert en boucle fermée :

$$y(t) = q^{-d} \frac{BQ_1}{B^*(1)P} w(t) + \frac{S}{PP_o} h \tag{II.74}$$

II.4.4 Structure de réglage à PP (Placement de Pôles).

La structure de réglage est représentée par le schéma suivant:

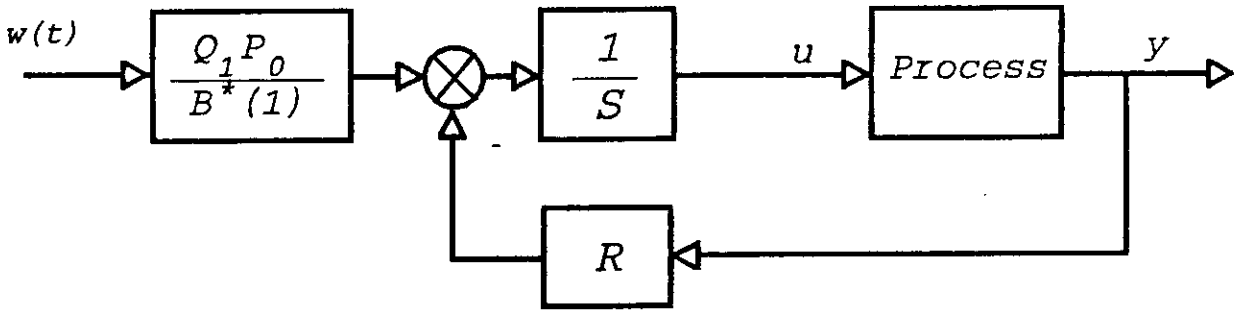


Figure II.7 Structure de réglage à PP.

De l'équation (II.74), 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 (II.69) on résout:

$$A(1-q^{-1})S + q^{-(d+1)} B^* R = PP_o \tag{II.75}$$

II.4.5 *Algorithme de réglage à PP adaptative.*

Lorsque les paramètres du système sont inconnus ou variables dans le temps, l'utilisation de l'estimation récursive s'impose. On distingue deux stratégie de commande adaptative:

- La commande adaptative indirecte.
- La commande adaptative directe basée sur le prédicteur déterministe .

a- Algorithme à PP explicite.[40]

Données: Spécifier $P(q^{-1})$, $P_o(q^{-1})$, $Q_1(q^{-1})$, d, n et m.
Etape 1: Estimation de A et B en utilisant l'algorithme MCR à trace constante (Chap I.3.4) avec:

$$\theta^T = [a_1 \ldots a_n \; b_1 \ldots b_m].$$
$$\phi^T(t) = [-y(t-1) \ldots -y(t-n) \; u(t-d-1) \ldots u(t-d-m)\;].$$
$$\varepsilon(t) = y(t) - \theta^T \phi(t).$$

Etape 2: Génération de $w(t)$.
Etape 3: Résolution de l'équation Diophantine (II.69) et calcul de T à partir de (II.73).
Etape 4: Calcul de la commande de l'équation (II.47).
t=t+1 revenir à étape 1.

b- Algorithme implicite à PP.

L'estimation directe des paramètres du régulateur nécessite la construction d'un prédicteur déterministe à partir de l'équation Diophantine (II.47) [40].

Données: Spécifier $P(q^{-1})$, $P_o(q^{-1})$, $Q_1(q^{-1})$ d, n et m.
Etape 1: Estimation des paramètres du modèle de prédiction défini par:

$$PP_o y(t) = S_1 y(t-d-1) + R_1 u(t-d-1). \quad S_1 = B^* S \text{ et } R_1 = B^* R.$$
$$\theta^T = [s_0 \ldots s_{r'_{max}-1} \; r_0 \ldots r_{r'_{max}-1}\;].$$
$$\phi^T(t) = [y(t-d-1) \ldots y(t-d-r'_{max}) \; u(t-d-1) \ldots u(t-d-r'_{max})].$$
$$\varepsilon(t) = PP_o y(t) - \theta^T \phi(t).$$

En utilisant l'algorithme à MCR à trace constante (Chap I.3.4).
Etape 2: Chercher le facteur commun (le plus grand) entre R_1 et S_1 et factoriser $S_1 = B^* S$ et $R_1 = B^* R$. Calcul de T à partir de (II.73).
Etape 3: Génération de $w(t)$.
Etape 4: Calcul de la commande de l'équation (II.47).
t=t+1 revenir à étape 1.

Remarque.
L'algorithme directe exposé aboutit à un modèle en BF dont les zéros sont variables dans le temps :

$$H(q^{-1}) = \frac{Q_1(q^{-1})\hat{B}(q^{-1})}{P(q^{-1})}.$$

La recherche du plus grand diviseur commin PGDC de R_1 et S_1 est une procédure simple quand les degrés sont faibles.□

II.5 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 $PB+QA$ [16][28][35]. 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 [15]. Tandis que l'algorithme d'Astrom [32] différe de ces derniers, par la simplification des zéros du système qui se situent dans une région de stabilité acceptable [30].
Puthempura [41] propose un algorithme qui simplifie non pas les zéros du système, mais les zéros du signal d'erreur pour une entrée de référence donnée.

II.5.1 Formulation du problème.

Le modèle du processus est le modèle DARMA défini par l'équation (II.48). 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 (II.66). La fonction de transfert en BF est définie par (II.67). Le problème qui se pose est le même que celui à PP défini par l'équation suivante:

$$\frac{q^{-(d+1)}B^*T}{AS+q^{-(d+1)}B^*R} = \frac{Q}{P'} \qquad (II.76)$$

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

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

De l'équation (II.76), 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 [32].

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

$$B^*(q^{-1}) = B^+(q^{-1})B^-(q^{-1}) \qquad (II.77)$$

Où les zéros de B^+ sont dans la région de stabilité $\mathcal{Z}$ du plan complexe et les zéros de B^- à l'extérieur de cette région. La région $\mathcal{Z}$ 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 [32]. La condition nécessaire pour la stabilité du problème posé par (II.76) et (II.77) est :

$$Q = Q_1 B^- \qquad (II.78)$$

En remplaçant (II.77) et (II.78) dans (II.76) on obtient :

$$\frac{q^{-(d+1)}TB^+B^-}{B^+(AS_1+q^{-(d+1)}B^-R)} = \frac{Q_1B^-}{P'} \qquad (II.79)$$

Or B^+ B^+ est stable (Dans la région $\mathcal{Z}$). L'équation (II.79) devient:

$$\frac{q^{-(d+1)}TB^-}{AS_1+q^{-(d+1)}B^-R} = \frac{Q_1B^-}{P'} \qquad (II.80)$$

Où $S=S_1B^+$.

II.5.2 Calcul des polynômes R et S.

Avec un choix des pôles en BF (P) et le polynôme d'observation (P_0) (défini précédemment), l'équation (II.80) donne:

$$AS_1+q^{-(d+1)}B^-R = PP_o \qquad (II.81)$$

Pour la résolution de cette équation voir annexe 5 où:

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

$deg\ S_1 = degS - degB^+$

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

$$S = S_1 B^+ . \tag{II.82}$$

II.5.3 Calcul de T.

Pour le calcul de T, on utilise l'équation (II.80) et le polynôme d'observation P_0, d'où:

$$T = Q_1 P_o \tag{II.83}$$

Pour avoir une bonne poursuite, il faut que le comportement en régime permanent soit celui du modèle défini par (II.66), d'où le choix du polynôme de poursuite :

$$T(q^{-1}) = \begin{cases} \dfrac{Q_1 P_o}{B^-(1)} & si \quad B^-(1) \neq 0. \\ Q_1 P_o & si \quad B^-(1) = 0. \end{cases} \tag{II.84}$$

D'où la fonction de transfert en BF:

$$y(t) = q^{-(d+1)} \frac{B^- Q_1}{B^-(1) P} w(t) + \frac{S_1}{P P_o} h \tag{II.85}$$

II.5.4 Structure du régulateur à PP et de zéros.

La structure de réglage est représentée par la figure II.8, en utilisant (II.82),(II.84),(II.48) et (II.47).

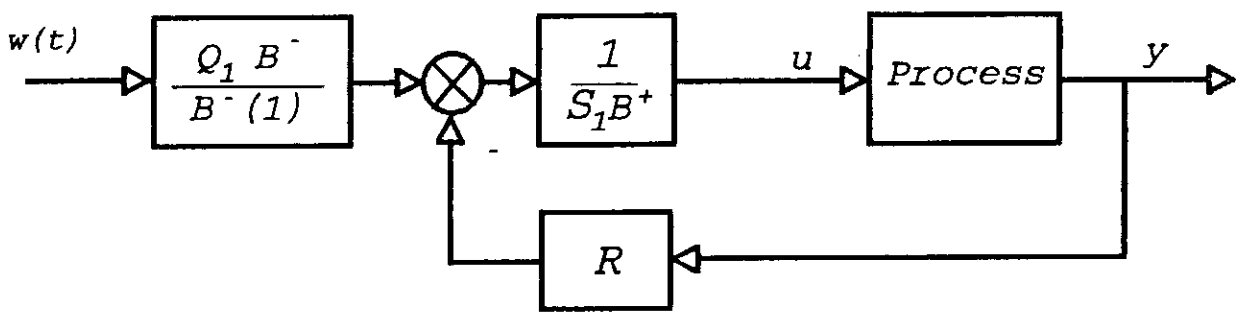


Figure II.8 Structure de réglage à PP et de zéros.

De l'équation (II.84), l'élimination de l'effet d'une perturbation constante se fait par un choix de S_1 multiple de $(1-q^{-1})$.

II.5.5 Algorithme de réglage adaptatif à PP et de zéros.

L'utilisation de l'estimation récursive permet l'implémentation de cet algorithme. L'algorithme d'identification utilisé est celui défini dans (chap I.3.4). On distingue deux approches :

- Clarke [38][42], Astrom [19][32] et Puthempura [14] proposent une approche indirecte de commande adaptative.

- Astrom [40] propose une approche directe basée sur la factorisation de R et S.

a- Algorithme à PP et de zéros explicite.

Données: Spécifier $P(q^{-1})$, $P_0(q^{-1})$, $Q_1(q^{-1})$, d, n et m.
Etape 1: Estimation de A et B en utilisant l'algorithme MCR à trace constante (Chap I.3.4) avec:

$$\theta^T = [a_1 \ldots a_n \; b_1 \ldots b_m].$$
$$\phi^T(t) = [-y(t-1) \ldots -y(t-n) \; u(t-d-1) \ldots u(t-d-m)\;].$$
$$\varepsilon(t) = y(t) - \theta^T \phi(t).$$

Etape 2: Factoriser le polynôme estimé $\hat{B}(q^{-1})$ sous forme de produit de $\hat{B}^+$ et $\hat{B}^-$.
Etape 3: Génération de $w(t)$.
Etape 4: Résolution de l'équation Diophantine (II.81).
Etape 4: Calcul de la commande de l'équation (II.47). Où $S=S_1B^+$ et T défini par (II.83).
t=t+1 revenir à étape 1.

Remarque.
L'algorithme à PP et de zéros aboutit à un modèle en BF défini par :

$$H(q^{-1}) = \frac{Q_1(q^{-1})\hat{B}^-(q^{-1})}{P(q^{-1})}.$$

qui n'est pas fixe, car les zéros de celle-ci sont estimés à chaque instant.

- La difficulté principale de cet algorithme est la factorisation du polynôme B^*, qui demande un temps de calcul important, dans le cas où $deg\ B^*$ est grand. Mais en général $deg(B^*) \leq 3$, donc la factorisation est facilement faisable. Dans le cas contraire, on utilise un algorithme numérique.□

b- Algorithme implicite à PP et de zéros.

L'estimation directe des paramètres du régulateur permet d'aboutir à la commande adaptative directe. L'utilisation d'un prédicteur déterministe s'avère nécessaire.

Données: Spécifier $P_1(q^{-1})$, $P_0(q^{-1})$, $Q_1(q^{-1})$ d, n et m.
Etape 1: Estimation des paramètres du modèle de prédiction défini par [27][40][43]:

$$PP_0y(t) = S_2y(t-d-1) + R_2u(t-d-1). \quad S_2=B^-S \text{ et } R_2=B^-R.$$
$$\theta^T = [s_{2_0} \ldots s_{2_s} \; r_{2_0} \ldots r_{2_r}]. \quad s=deg(B^-)+deg(S) \text{ et } r=deg(B^-)+deg(R).$$
$$\phi^T(t) = [y(t-d-1) \ldots y(t-d-1-s)\, u(t-d-1) \ldots u(t-d-1-r)].$$
$$\varepsilon(t) = PP_0y(t) - \theta^T \phi(t).$$

En utilisant l'algorithme à MCR à trace constante (Chap I.3.4).

Etape 2: Chercher le facteur commun (le plus grand) entre $\hat{R}_2$ et $\hat{S}_2$ et factoriser $\hat{S}_2=\hat{B}^-\hat{S}$ et $\hat{R}_2=\hat{B}^-\hat{R}$. . Calcul de T à partir de (II.83).
Etape 3: Génération de $w(t)$.
Etape 4: Calcul de la commande de l'équation (II.47).
t=t+1 revenir à étape 1.

II.6 Commande quadratique à un pas (LQC: Linear Quadratic Controller).

Dans la commande optimale, le choix du critère de performance spécifie les caractéristiques du système en BF. 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 [31]. 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.

Dans le cas de la régulation, le solution du problème LQC, à horizon fini, consiste à la résolution de l'équation de Ricatti [31][44].Tandis que pour la poursuite, le problème s'étend à la résolution d'une autre équation qui donne la commande nécessaire assurant cet objectif [45].

Thompson [46] Présente une méthode de calcul de la commande avec une mesure, sur un horizon bien fini, de la référence, dans le cas où le critère s'étend sur un horizon infini. Arstein & Leizariwitz [47] étudient le problème de la poursuite pour une référence périodique. Pour une référence non périodique, Leizarowitz [48] démontre que le calcul de la commande, avec une connaissance suffisante sur la référence, pendant un intervalle de temps fini, aboutit à un critère optimal au sens du moins divergent (overtaking optimality criterium). Louam [49] propose une une etude de l'applicabilité des méthodes de la commande optimale dans le cas de la suspension des voitures.

La version discrète du problème de Régulation, sur un horizon fini, a été proposé par Clarke [17], dans laquelle le résolution de l'équation de Ricatti se fait qu'un seul pas chaque période d'échantillonnage. Toivonen [50], M'Saad & Najim [51] présente le même problème, avec un modèle d'état différent de celui de Clarke.
Dans le cas de la poursuite, sur un horizon infini, Astrom [30] et Kwakerneak [52] utilisent le modèle inverse du système pour aboutir à une action anticipatrice sur la commande. Par contre Bühler [53] développe la version discrète du problème, dans le cas où l'horizon du critère est fini.

L'approche polynomiale du problème, qui se base sur la factorisation spectrale, a été proposée par Grimble [54] et bien développée dans Astrom & Wittenmark [30].

Dans ce paragraphe, on présente une méthode similaire à celle de Berger [55], dans laquelle on évite, d'une part, la résolution de l'équation de Ricatti et l'équation de poursuite, d'autre part la résolution de l'équation Diphantine.
Le modèle du système est sous forme d'état suivante [30]:

$$\begin{cases} x(k+1) = \psi x(k) + \Gamma U(k) . \\ y(k) = Cx(k) \quad . \end{cases} \tag{II.86}$$

Où $x^T = [x_1 \ldots x_n]$: état du système.
$U^T = [u_1 \ldots u_m]$: entrées de commande.
$y^T = [y_1 \ldots y_p]$: sorties du système.
ψ : matrice de dimension nxn.
Γ : matrice de dimension nxm.
C : matrice de dimension pxn.

Le critère choisi est sous le forme suivante [1]:

$$J = \frac{1}{2}[x(k+1) - x_d(k+1)]^T Q [x(k+1) - x_d(k+1)] + \frac{1}{2} U^T(k) RU(k) \quad (II.87)$$

Où Q: matrice symétrique semi définie positive de dimension nxn.
R: matrice symétrique définie positive de dimension mxm.
x_d: Trajectoire d'état désirée.

En remplaçant l'équation (II.86) dans (II.87) on obtient :

$$J = \frac{1}{2}[\psi x(k) + \Gamma u(k) - x_d(k+1)]^T Q [\psi x(k) + \Gamma u(k) - x_d(k+1)] + \frac{1}{2} U^T(k) RU(k) \quad (II.88)$$

La commande optimale qui minimise le critère défini par (II.88)est $\left(\frac{\delta J}{\delta u_k} = 0\right)$ [1] [56]:

$$u(k) = -(\Gamma^T Q \Gamma + R)^{-1} [\Gamma^T Q \psi x(k) - \Gamma^T Q x_d(k+1)] \quad (II.89)$$

Remarque.

l'unicité et l'existence de la solution optimale au problème précédent est garantie si les pairs (ψ, Γ) *et* (ψ, Q) sont respectivement commandable et observable [44].□

Le choix des matrices de pondération est une procédure très difficile, qui demande une connaissance approfondie sur le comportement dynamique du système. Athans [44] propose une méthode pour le choix de ces matrices, qui est lié à la formulation mathématique du critère. Il considére que la minimisation du critère défini par (II.87) revient à réduire l'approximation d'un modèle non linéaire par un modèle défini par les équation (II.86).

Dans le cas monovariable, où le système est représenté par le modèle défini comme suit:

$$x(k+1) = \psi x(k) + \Gamma U(k) \quad (II.90)$$

Où $\psi = \begin{bmatrix} \psi_{11} \cdots \psi_{1n} \\ \vdots \quad \vdots \\ \psi_{n1} \cdots \psi_{nn} \end{bmatrix}$ et $\Gamma^T = [\Gamma_1 \ldots \Gamma_n]$

$x^T = [x_1 \ldots x_n]$ état du système de dimension n.
$u(k) = u_1$: commande scalaire.

Dans ce cas, la matrice de Pondération R se réduit à un scalaire, ce qui évite l'inversion de $\Gamma^T Q \Gamma + R$. Le critère devient:

$$J = \frac{1}{2}[x(k+1) - x_d(k+1)]^T Q [x(k+1) - x_d(k+1)] + \frac{1}{2} r U^2(k) \quad (II.91)$$

Où r est une constante réel positive.

II.6.1 Structure du régulateur optimale.

Dans le cas où les paramètres du système sont connus, la commande du processus est définie par le schéma représenté par la figure II.9.

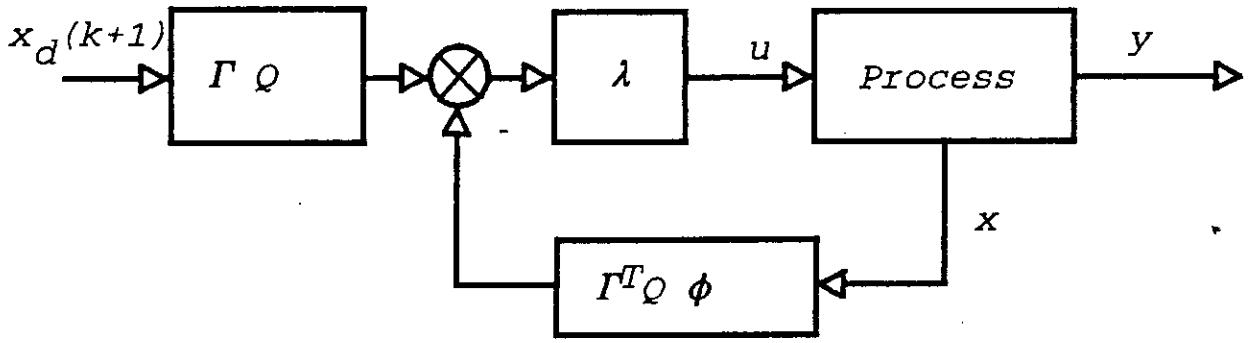


Figure II.9 Structure de réglage à LQC $\lambda=(\Gamma^T Q\Gamma+R)^{-1}$.

L'algorithme ainsi développé assure la poursuite et la régulation et ne nécessite pas la résolution de l'équation de Ricatti, ni l'équation Diophantine.

II.6.2 Algorithme de réglage adaptative à LQC à un pas.

Dans le cas où les paramètres du système sont inconnus ou variables dans le temps, l'utilisation d'un estimateur récursif s'avère nécessaire. Les étapes à suivre dans le cas adaptative sont:

Données: Spécifier
Q, R, n(*ordre du systeme*), m(*nombre d'entrées*=1).
Etape 1: Estimation des paramètres du système (ψ *et* Γ) en utilisant l'algorithme MCR à trace constante (Chap I.3.4) avec:

$$\theta_i^T = [\psi_{i_1} \ldots \psi_{i_n}\ \gamma_i].$$
$$\phi^T(k) = [x_1(k) \ldots x_n(k)\ \ u(k)].$$
$$\varepsilon_i(k) = x_i(k) - \theta_i^T \phi(k). \quad \textit{pour } i=1,n.$$

Etape 2: Génération de $x_d(k+1)$.
Etape 3: Calcul de la commande (équation II.89).
t=t+1 revenir à étape 1.

II.7 Synthèse et discussion.

Toutes les méthodes développées précédemment nécessitent une identification en temps réel. Dans le cas des procédés complexes, l'algorithme à MCR, s'avère lent, on est amené à utiliser d'autres algorithmes "rapide". L'algorithme RMS (Rooth Mean Square algorithm) présenté dans [32][36], procéde par une factorisation du gain d'adaptation et accélère la vitesse de calcul, donc la convergence. Astrom [32] 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 [34] présente un algorithme sous le nom "Multiple récursion algorithme", dans lequel la matrice du gain d'adaptation se réduit à un scalaire, similaire à l'algorithme de projection [27,57].

La commande à Variance Minimale où d'une manière équivalente avec modèle de

référence [35], nécessite la connaissance du retard du système et une condition de stabilité des zéros du processus. Le critère utilise la sortie généralisée tandis que M'Saad [58] utilise un modèle de référence sur l'état partiel.

La commande à **Variance Minimale Généralisée** permet d'échapper au problème des systèmes à phase non minimale. Mais l'introduction d'une Pondération Q, rend difficile la spécification du modèle en BF.

La commande par **Poursuite et Régulation** est similaire à celle à VM. La différence réside dans le choix du polynôme d'observation. Dans le cas stochastique, ce dernier est celui de perturbation. Tandis que dans le cas déterministe ce dernier peut être choisi unitaire, ou des pôles spécifiés par l'utilisateur.

La commande par **Placement de pôles** surmonte le problème de système à phase non minimale, sans introduire une Pondération. Elle permet la spécification d'un modèle de référence en BF. L'algorithme ne permet pas la simplification de la dynamique des zéros instable. La résolution de l'équation Diophantine peut causer des instabilités numérique, par la singularité de la matrice à inverser. Chih-Min Lin [59] présente un algorithme à PP rapide, ne nécessitant pas la résolution de l'équation Diophantine, mais seulement l'inversion d'une matrice de dimension égale au degré du dénominateur du système. Donc le temps de calcul se réduit de moitié. Limo & Bayomi [60] présente une nouvelle approche à placement de pôles, qui nécessite seulement deux identifications. L'une identifiant les paramètres du système et la seconde ceux du régulateur avec les paramètres identifiées (du système) et une construction d'un estimateur bien défini. C.Y.Chan [61] développe un controlleur adaptative directe, applicable seulement aux systèmes stables, car il procéde à la factorisation du polynôme R (multiple de A). La méthode est similaire au PP et de zéros exposé par Astrom [40]. Kinaert [62] souleve le problème de stabilité des polynômes R et S issus de la résolution de l'équation Diophantine, qui peuvent être instable. Il expose une méthode de recherche des polynômes R *et* S stables (dans une région de stabilité spécifiée par l'utilisateur). Astrom [63] analyse le problème de robustesse de l'algorithme à PP et de zéros.

La commande **linéaire quadratique à un pas (LQG)** permet d'éviter la résolution de l'équation Diophantine. Le critère à un pas surmonte le problème de résolution de l'équation de Ricatti et de poursuite. L'extension de l'algorithme aux cas multivariables est évidente. La recherche des pondérations Q *et* r est une procédure délicate, même en faisant le choix proposé par Athans [44], qui nécessite un temps de calcul important dans le cas adaptative. L'approche polynomiale de la commande LQC permet le calcul du polynôme en BF, en utilisant la factorisation spectrale [30] et la résolution de l'équation Diophantine dans le cas explicite [54]. C'est cette propriété qui rend cet algorithme supérieur à celui formulé sous forme d'état. Le système en boucle fermée possède une marge de gain infini et une marge de phase de 60°, dans le cas de la commande à critère infini [64].

II.8 Approche multivariable.

La commande des systèmes multivariables nécessite la généralisation des algorithmes monovariables aux cas multivariables. Borison [65] propose un régulateur auto-ajustable multivariable, basé sur la minimisation de la variance du vecteur de sortie, utilisant un estimateur récursif et un controlleur linéaire obtenu directement des estimés actuels. L'extension de la méthode de Clarke & Gawthrop [28] au cas multivariable est présentée par Koivo [66], dans laquelle l'estimation des paramètres du régulateur se fait avec l'algorithme SRA (Square Rooth Algorithm). Goodwin [57] expose l'approche directe de la VM, en utilisant un modèle de représentation Diagonale-Pleine [9]. Tandis que dans son article [34], il présente le même algorithme avec un modèle Plein-Plein [9].

En utilisant un modèle multivariable CARIMA et une connaissance sur la borne supérieur de l'interacteur, Scatoline & Clarke [67] développe la version explicite multivariable de l'algorithme à GMV. La version implicite du GMV est donnée par Mahieddine [68].

Mei-Hua Liu & Wei Lin présente dans leur premier article [23], une méthode de calcul récursive des polynômes de pondération P et Q, pour assurer la spécification des pôles en BF, dans le cas de la GMV. Tandis que dans leur second article [22], ils présentent l'approche directe avec un modèle Diagonale_Diagonale [9]. Shi-Jun Lang [69] expose l'algorithme à GMV, dans lequel il calcule d'une façon récursive les polynômes P, Q et R; et assurant un découplage totale du système en BF. L'extension de la méthode aux systèmes bilinéaires à été développée par De La Sen [70].

Prager & Wellstead [71] donnent une extension de l'algorithme à PP, initialement développé par Welstead [15]. dans le cas des systèmes stables, H.R.Sirisina [72] développe l'algorithme en spécifiant le modèle en BF. M.O.Tade [73] développe l'algorithme, assurant un découplage totale du système en BF, et ne nécessitant pas la résolution de l'équation Diphantine. Deux algorithmes d'identification sont utilisés, celui des paramètres du système et un autre du régulateur. En utilisant un modèle Diagonale-Plein [9], Zi-li & Xian Ri [74] présente l'algorithme à PP dans lequel la résolution de l'équation Diophantine se fait d'une façon très simple (similaire au cas monovariable).

Toivonen [50] 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 intéracteur, dont Dugard [75] présente son rôle, en développant l'algorithme à VM avec une connaissance seulement sur la borne supérieure de cette matrice. Tandis que Zhang & Lang [76] présente l'algorithme à GMV dans le cas où la matrice intéracteur est quelconque et le modèle du système est nonlineaire (modèle Hamerstein).

Les modèles découplés utilisés par Zi-Li-Deng [74], Mei Hua liu [23], Tham [77] et Bouzouia [78] 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 nécessite un temps de calcul énorme.

Le choix d'un modèle découplé est lié aux nombres 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 on présente les versions multivariables des algorithmes synthétisés dans les parties précédentes. Dans le cas de notre application, le robot posséde trois entrées et trois sorties. On présente seulement l'algorithme à VM. Les autres algorithmes s'en découlent de la même manière. L'algorithme multirate sera présenté en dernier lieu.

II.8.1 Algorithme à variance minimale.

Le modèle du robot est représenté par l'équation (II.1). Dans ce cas on spécifie trois critères de performance régissant chaque entrée-sortie. Le modèle de référence est spécifié par le choix de trois polynômes de pondération. Donc le critère est :

$$J_i = E\left[P_i y_i(t+d_i+1) - R_{w_i} w_i(t+d_i+1)\right]^2; \quad i=1,3 \qquad \text{(II.92)}$$

Où y_i est une composante du vecteur de sortie y.

La minimisation du critère aboutit à la commande :

$$U_i(t) = \frac{P_{D_i} C_i R_{w_i} w_i(t+d_i+1) - R_i y_i(t) - P_{D_i}(1) S_i'(1) h_i}{P_{D_i} S_i}; \quad i=1,3 \qquad \text{(II.93)}$$

Où R_i et S_i: sont solutions de :(voir annexe 4)

$$A_i P_{D_i} S_i' + q^{-(d_i+1)} R_i = P_{N_i} C_i \qquad \text{(II.94)}$$

Les performances en BF sont spécifiés par l'équation (II.21) pour chaque entrée-sortie, avec un choix adéquat des polynômes de pondération indépendamment les uns par rapport aux autres. La structure de régulateur adaptative à VM multivariable, appliquée au robot est illustrée sur la figure II.10.

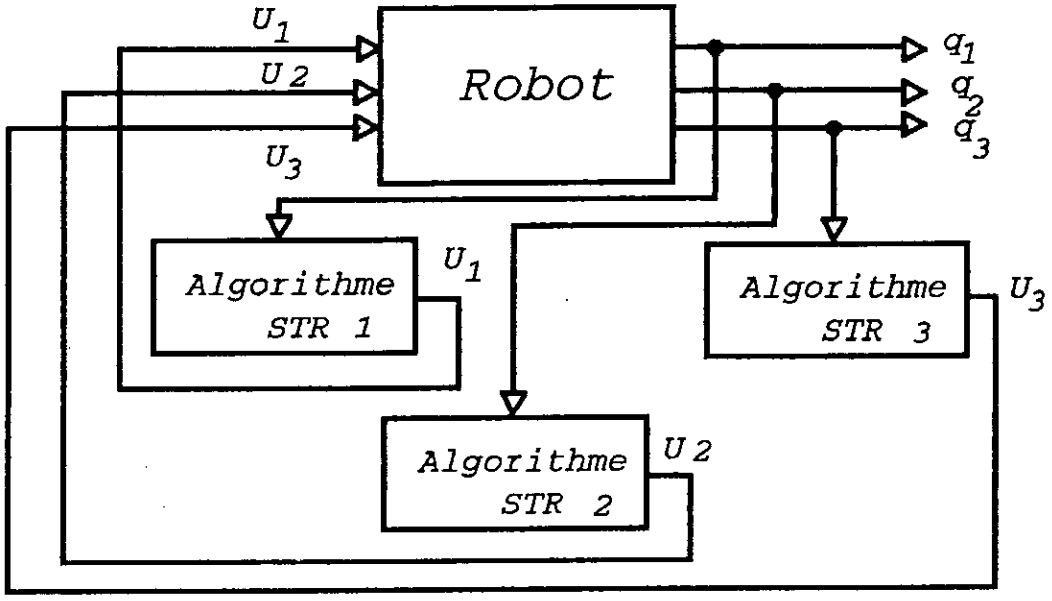


Figure II.10 Structure du régulateur adaptative à VM.

Où chaque bloc définissant $Algo_i$ est représenté sur la figure II.11. Tandis que la figure II.12 illustre la structure de la commande à VM adaptative directe.
Les versions explicites et implicites de la variance minimale adaptative sont développées respectivement dans chap II.1.2.a et b.

Remarque.

Dans le cas de la commande LQC, l'algorithme d'identification est le même que celui du monovariable. La matrice d'état est diagonale par bloc, non pas sous une forme canonique de commandabilité ou d'oservabilité, mais sous une forme quelconque.□

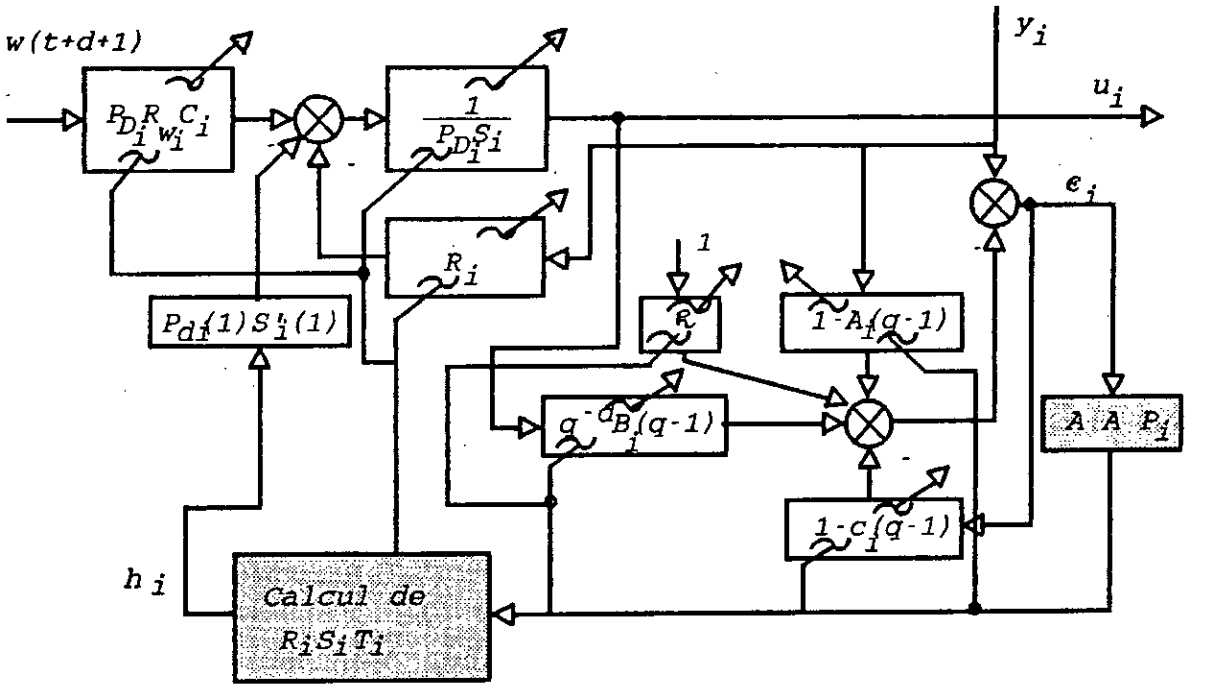


Figure II.11 Structure de chaque bloc directe à VM.

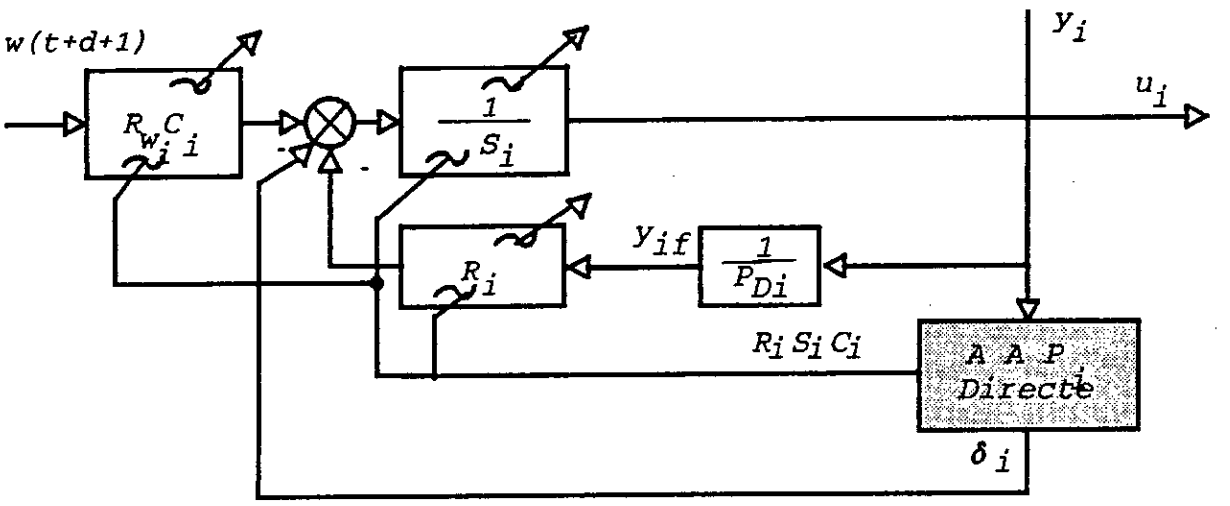


Figure II.12 Structure de chaque bloc directe à VM.

II.8.2 Commande mulivariable 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 controleur 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 [9]. 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 II.10. 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 maniéré importante. Le choix de la période d'échantillonnage s'effectue d'une maniéré indépendante, pour chaque entrée-sortie. Tham [77] et Morris [79] présentent l'algorithme multirate appliqué à une colonne à distiller. Tandis que Montague [80] propose une comparaison entre les algorithmes de commande adaptative multivariable, ainsi que la commande multirate.

L'implémentation de l'algorithme multirate dans le cas adaptative nécessite le choix d'une période d'une période d'échantillonnage de base T_b [79]. 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.

II.9 Résultats de simulation

Le robot utilisé dans la simulation est un robot classe 4 (chapitre I).

La commande à GMV est appliquée au robot, pour une consigne polynomiale du 3^{éme} degré, qui assure une continuité en position et en vitesse aux extrémités [21], définie par:

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

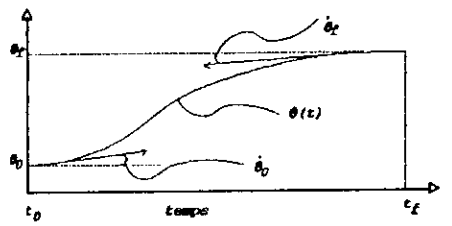
$a_0 = \theta_0$.

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

Où $a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f$.

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

Dont les conditions aux limites: $\theta(t_0) = \theta_0$; $\dot{\theta}(t_0) = \dot{\theta}_0$.
$\theta(t_f) = \theta_f$; $\dot{\theta}_f(t_f) = \dot{\theta}_f$.



Les résultats de simulation sont consignés sur la figure II.18.a. Les valeurs des commandes sont représentées dans la figure II.18.b. Les figures II.19.a et b présentent les sorties et les commandes dans le cas où la pondération est:

$$Q_i = \lambda_1 (1 - q^{-1}); \; i = 1,3.$$

Dans le cas où on introduit une dynamique de sortie définie par le polynôme de pondération P et un modèle de la référence w, dont les valeurs sont:

$$\frac{B_{m_i}}{A_{m_i}} = \frac{0.095}{1 - 0.9q^{-1}}; \; P_i(q^{-1}) = 1 - 0.5q^{-1} + 0.06q^{-2}.$$

les Résultats de simulation sont consignés sur les figures II.20.a et b.

Remarque.
le polynôme de perturbation est $C_i(q^{-1})=1+c_{i_1}q^{-1}+c_{i_2}q^{-2}$ et le polynôme d'observation est unitaire.□

La commande par PRPE est appliquée au robot, pour la même consigne que précédemment. Les figures II.21.a et b représentent les résultats de simulation pour une telle commande. En gardant les même caractéristiques, en prenant $\lambda_1=0.007$, les résultats de simulation sont consignés sur les figures II.22.a et b. Par introduction du polynôme de pondération définissant les pôles en BF, les résultats de simulation sont consignés aux figures II.23.a et b. Pour voir l'effet du polynôme de pondération $Q_i=\lambda_i(1-q^{-1})$, les figures II.24.a et b représentent les résultats de simulation dans un tel cas. En changeant la dynamique des pôles en BF, les résultats sont illustrés dans les figures II.25.a et b. Pour voir l'effet d'introduire un modèle sur la référence w, on choisit $\frac{B_{m_i}}{A_{m_i}}=\frac{0.095}{1-0.9q^{-1}}$. les Résultats de simulation sont représentés sur les figures II.26.a et b.

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 II.27.a et b.
Dans le cas où le modèle en BF est défini par:

$$\frac{Q_i}{P_i}=\frac{1}{1-0.5q^{-1}+0.06q^{-2}}$$

les figures II.28.a et b montrent les Résultats de simulation.
En gardant les mêmes caractéristiques que précédemment pour le modèle en BF, on introduit un modèle de la référence $\frac{B_{m_i}}{A_{m_i}}=\frac{0.095}{1-0.9q^{-1}}$; les résultats dans ces conditions sont consignés sur les figures II.29.a et b.

Les figures II.30.a et b représentent les résultats se simulation dans le cas où le robot est commandé avec la commande LQC à un pas. Les paramètres sont initialisés à 1.

$$où\ Q_i=\begin{bmatrix} q_{i_1} & 0 \\ 0 & q_{i_2} \end{bmatrix}\ et\ R_i=r_i.$$

La commande mixte est appliquée au robot, où la première articulation est commandée avec l'algorithme à PRPE:

$$Q_1=0.007\ ;\ P_1=1\ ;\ \frac{B_{m_1}}{A_{m_1}}=1;$$

et la deuxième articulation est commandée avec GMV:

$$Q_2=0.05\ ;\ P_2=1\ ;\ \frac{B_{m_2}}{A_{m_2}}=\frac{0.095q^{-1}}{1-0.9q^{-1}};$$

tandis que la troisième est commandée avec le PP:

$$\frac{Q_{13}}{P_3}=\frac{1}{1-0.5q^{-1}+0.06q^{-2}} \quad ; \quad \frac{B_{m_2}}{A_{m_2}}=\frac{0.095q^{-1}}{1-0.9q^{-1}};$$

Les résultatsde simulation dans ce cas sont consignés sur les figures II.31.a et b. Dans le cas où les pôles en BF de la troisième articulation se réduisent à l'unité (P_3=1), les figures II.32.a et b montrent les résultats d'une telle commande.

En appliquant la commande mixte, représentée par la figure II.11, à des périodes d'échantillonnage différentes, les résultats de la commande multirate sont illustrés dans la figure II.33.a et b.

Dans le cas où le modèle du robot est mal connu (avec une erreur de 500%) on fait varier tous les paramètres du robot de 500%.

$$(m_1;m_3;l_1;l_2;k_1;k_2;k_3;f_1;f_2;f_3)$$

Avec la commande mixte et les même caractéristiques, les Résultats de simulation sont montrés sur les figures II.34.a et b.

Pour tester la robustesse de l'algorithme à STR, aux variation paramétriques. On varie les paramètres du robot de 500% à t=5s. Les figures II.35.a et b montrent les résultats de simulation.

En dernier lieu pour une consigne définie par la fenêtre de VIVIANI (voir chapitre I) et en utilisant l'algorithme mixte, les figures II.36.a et b montrent les résultats de simulations. La projection de la trajectoire spatiale sur différent plan est consignée sur la figure II.36.c.

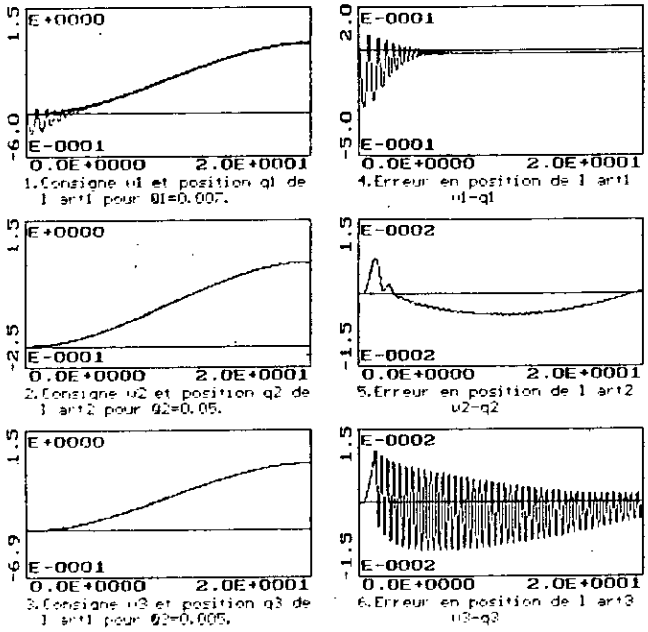


Figure II.18.a Position et erreur en position pour la GMU avec Pi=1;Bmi.Ami=1

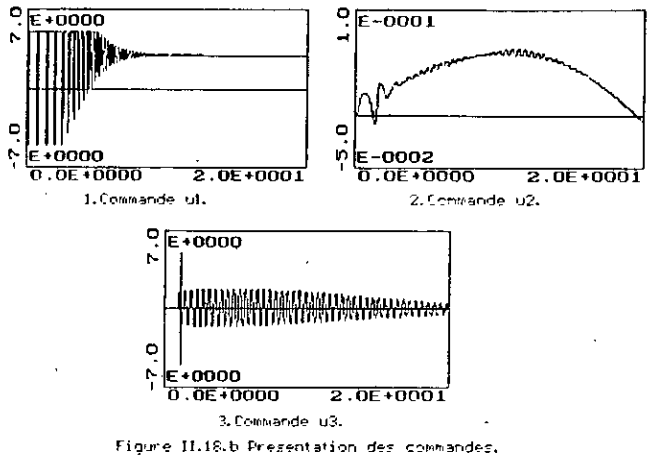
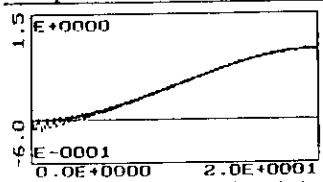
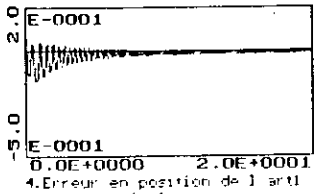


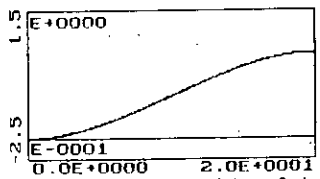
Figure II.18.b Presentation des commandes.



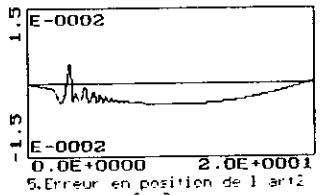
1.Consigne u1 et position q1 de l'art1 pour Q1=0.007(1-q-1).



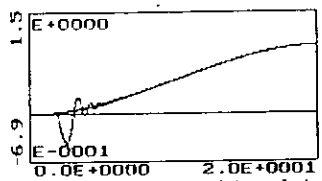
4.Erreur en position de l'art1 u1-q1.



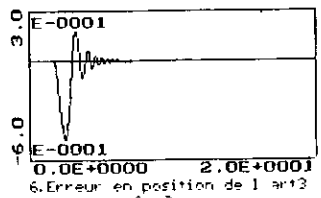
2.Consigne u2 et position q2 de l'art2 pour Q2=0.05(1-q-1).



5.Erreur en position de l'art2 u2-q2.

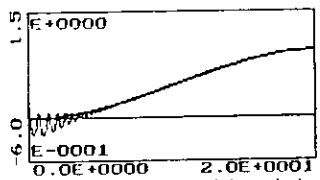


3.Consigne u3 et position q3 de l'art1 pour Q3=0.005(1-q-1).

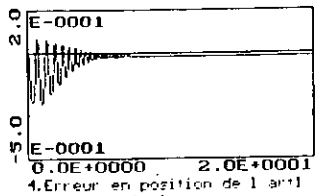


6.Erreur en position de l'art3 u3-q3.

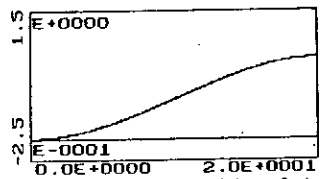
Figure II.19.a Position et erreur en position pour la GMV avec Pi=1;Bmi/Ami=1.



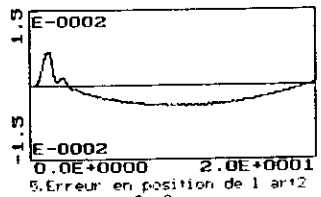
1.Consigne u1 et position q1 de l'art1 pour Q1=0.007.



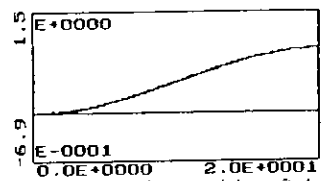
4.Erreur en position de l'art1 u1-q1.



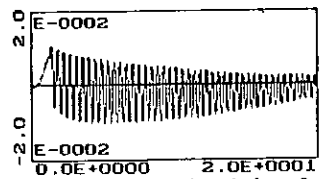
2.Consigne u2 et position q2 de l'art2 pour Q2=0.05.



5.Erreur en position de l'art2 u2-q2.

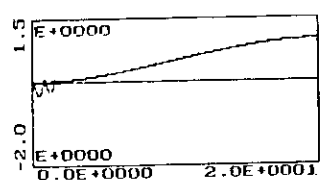


3.Consigne u3 et position q3 de l'art1 pour Q3=0.005.

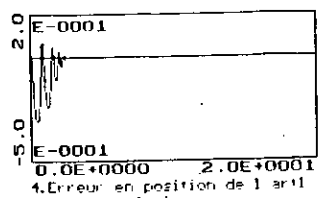


6.Erreur en position de l'art3 u3-q3.

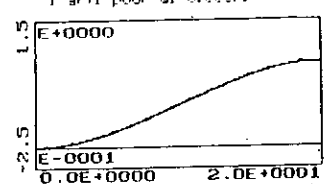
Figure II.20.a Position et erreur en position pour la GMV avec Pi=1-0.5q-1+0.06q-2;Bmi/Ami=0.095q-1/(1-0.9q-1).



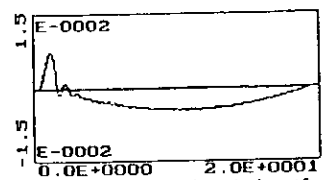
1.Consigne u1 et position q1 de l'art1 pour Q1=0.0007.



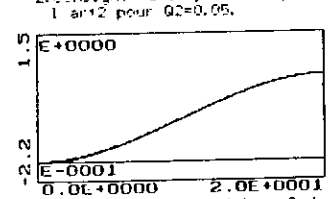
4.Erreur en position de l'art1 u1-q1.



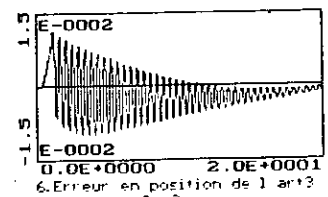
2.Consigne u2 et position q2 de l'art2 pour Q2=0.05.



5.Erreur en position de l'art2 u2-q2.

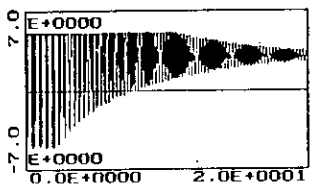


3.Consigne u3 et position q3 de l'art1 pour Q3=0.005.

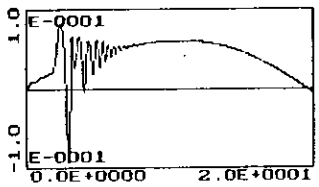


6.Erreur en position de l'art3 u3-q3.

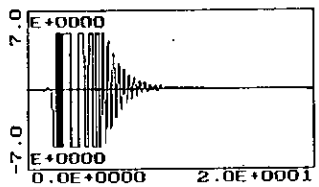
Figure II.21.a Position et erreur en position pour la PPPE avec Pi=1;Bmi/Ami=1.



1.Commande u1.

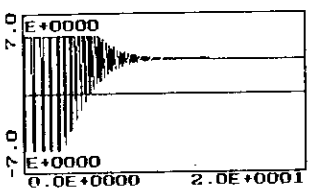


2.Commande u2.

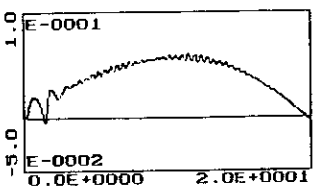


3.Commande u3.

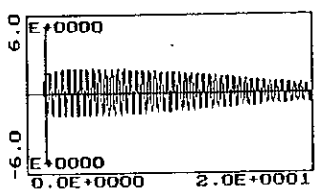
Figure II.19.b Presentation des commandes.



1.Commande u1.

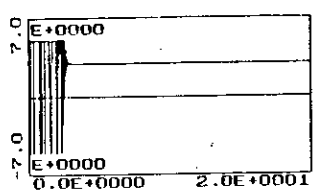


2.Commande u2.

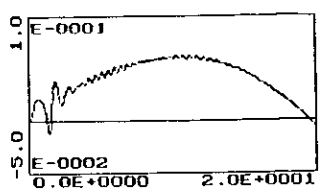


3.Commande u3.

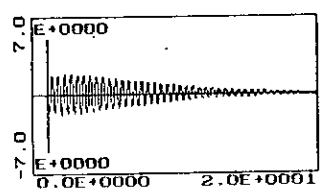
Figure II.20.b Presentation des commandes.



1.Commande u1.

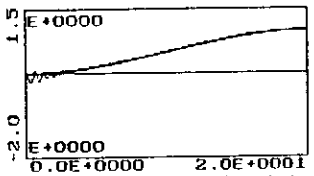


2.Commande u2.

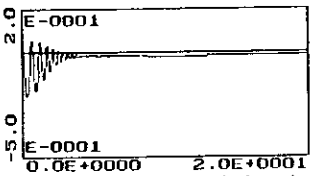


3.Commande u3.

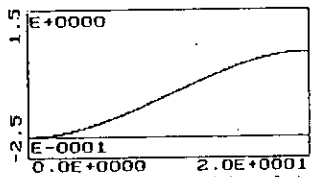
Figure II.21.b Presentation des commandes.



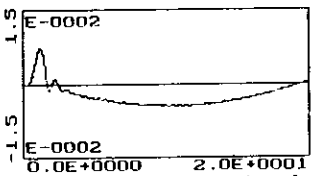
1.Consigne u1 et position q1 de l art1 pour Q1=0.00°.



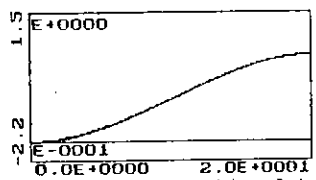
4.Erreur en position de l art1 u1-q1.



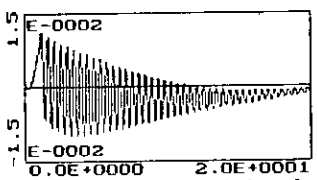
2.Consigne u2 et position q2 de l art2 pour Q2=0.05.



5.Erreur en position de l art2 u2-q2.

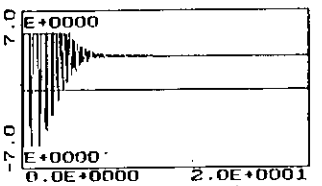


3.Consigne u3 et position q3 de l art1 pour Q3=0.005.

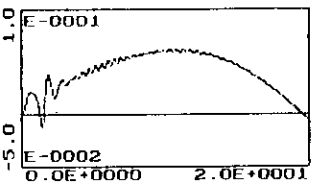


6.Erreur en position de l art3 u3-q3.

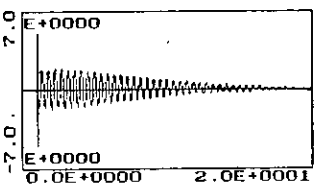
Figure II.22.a Position et erreur en position pour la PRPE avec Pi=1;Emi/Ami=1.



1.Commande u1.

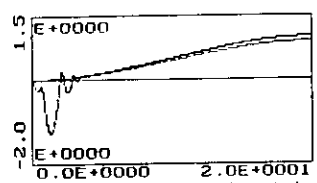


2.Commande u2.

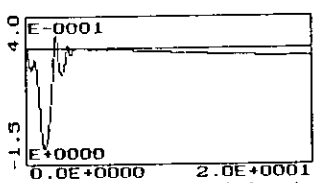


3.Commande u3.

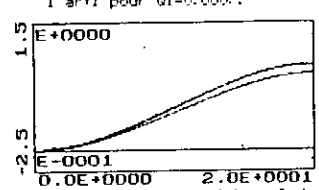
Figure II.22.b Presentation des commandes.



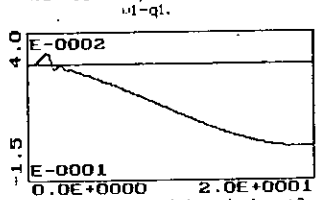
1.Consigne u1 et position q1 de l art1 pour Q1=0.000°.



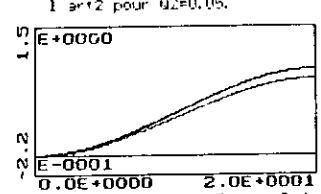
4.Erreur en position de l art1 u1-q1.



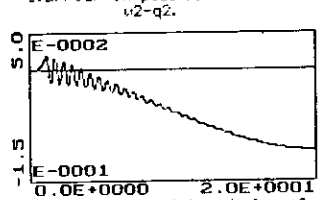
2.Consigne u2 et position q2 de l art2 pour Q2=0.05.



5.Erreur en position de l art2 u2-q2.

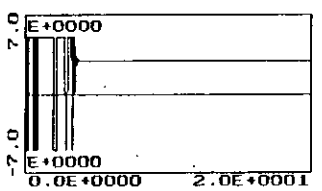


3.Consigne u3 et position q3 de l art1 pour Q3=0.005.

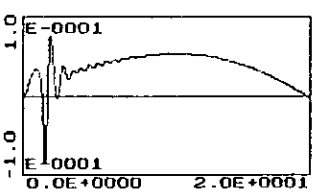


6.Erreur en position de l art3 u3-q3.

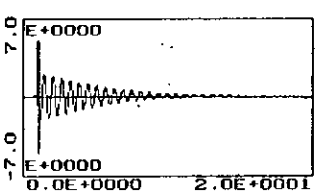
Figure II.23.a Position et erreur en position pour la PRPE avec Pi=1-0.5q-1+0.06q-2;Emi/Ami=1.



1.Commande u1.

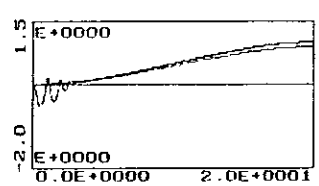


2.Commande u2.

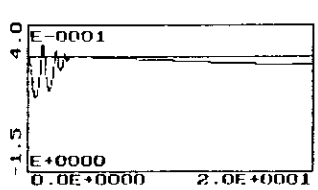


3.Commande u3.

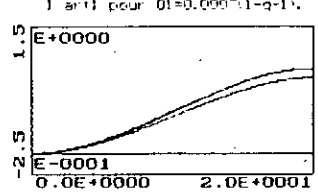
Figure II.23.b Presentation des commandes.



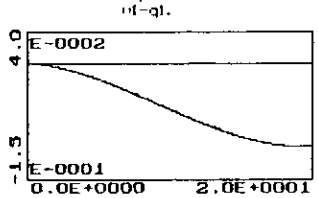
1.Consigne u1 et position q1 de l art1 pour Q1=0.000°(1-q-1).



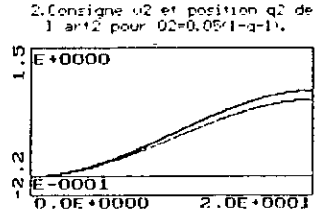
4.Erreur en position de l art1 u1-q1.



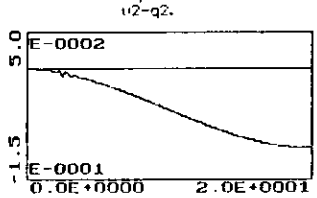
2.Consigne u2 et position q2 de l art2 pour Q2=0.05(1-q-1).



5.Erreur en position de l art2 u2-q2.

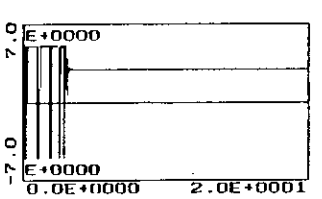


3.Consigne u3 et position q3 de l art1 pour Q3=0.005(1-q-1).

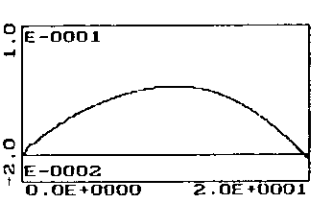


6.Erreur en position de l art3 u3-q3.

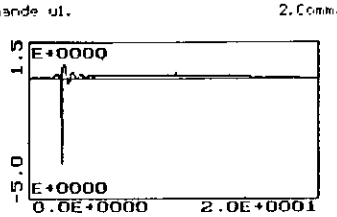
Figure II.24.a Position et erreur en position pour la PRPE avec Pi=1-0.5q-1+0.06q-2;Emi/Ami=1.



1.Commande u1.



2.Commande u2.



3.Commande u3.

Figure II.24.b Presentation des commandes.

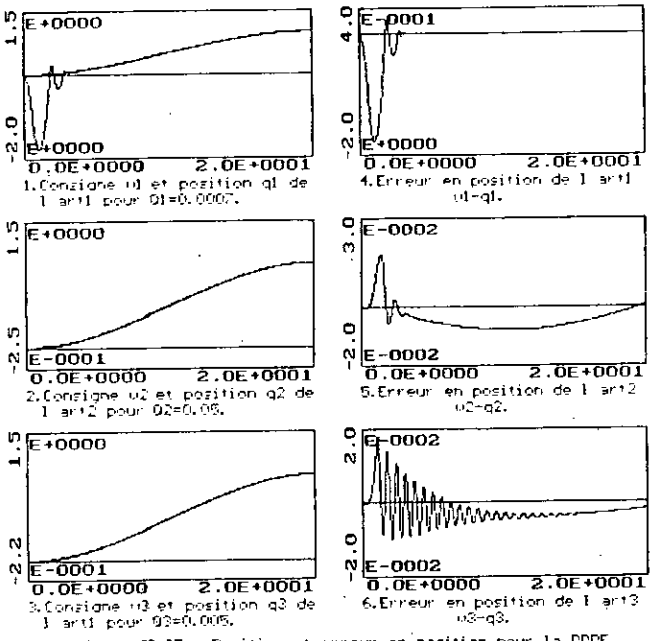


Figure II.25.a Position et erreur en position pour la PRPE avec Pi=1-0.5q-1;Bmi/Ami=1.

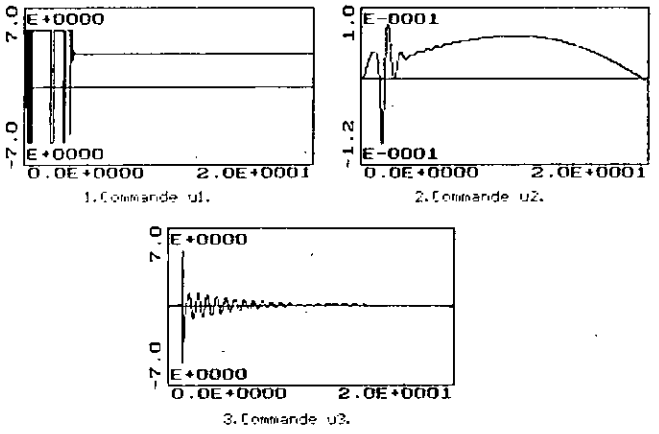


Figure II.25.b Presentation des commandes.

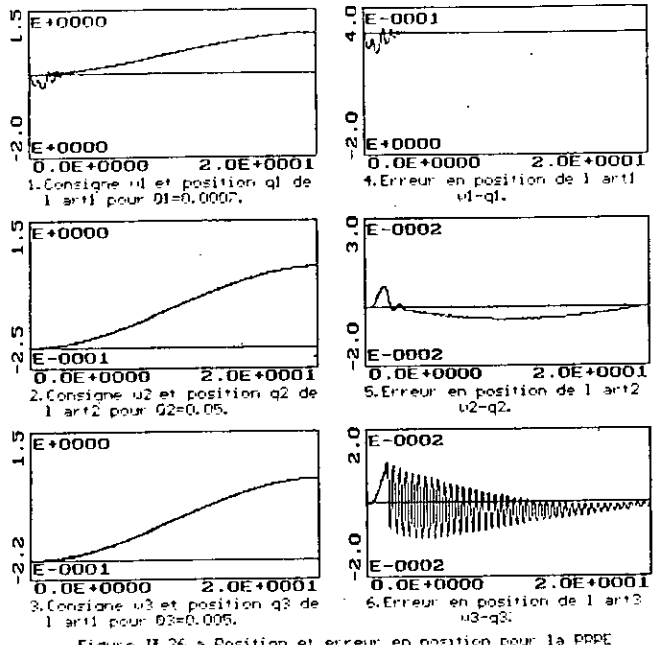


Figure II.26.a Position et erreur en position pour la PRPE avec Pi=1;Bmi/Ami=0.035q-1/(1-0.9q-1).

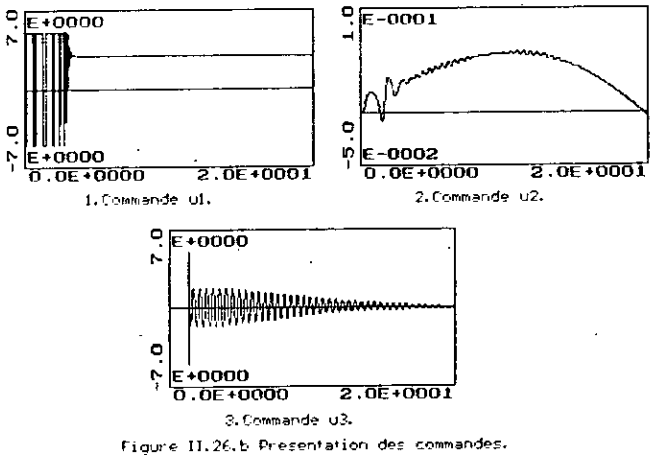


Figure II.26.b Presentation des commandes.

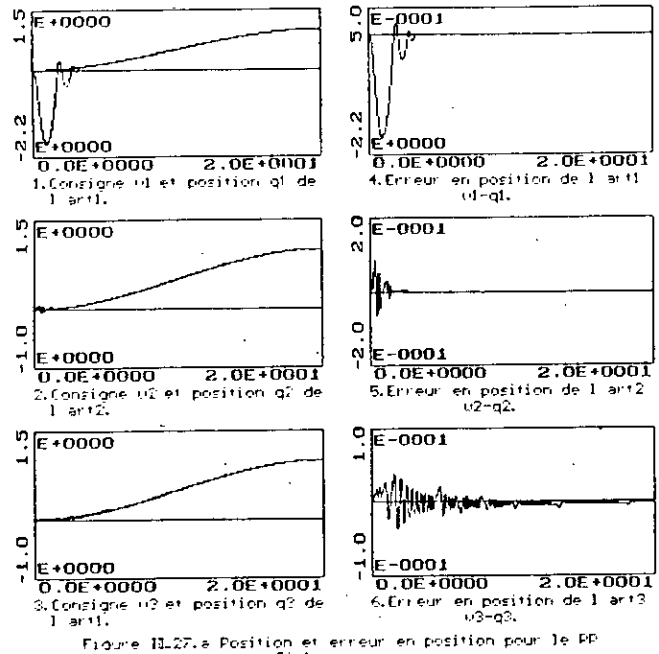


Figure II.27.a Position et erreur en position pour le PP avec Pi=1.

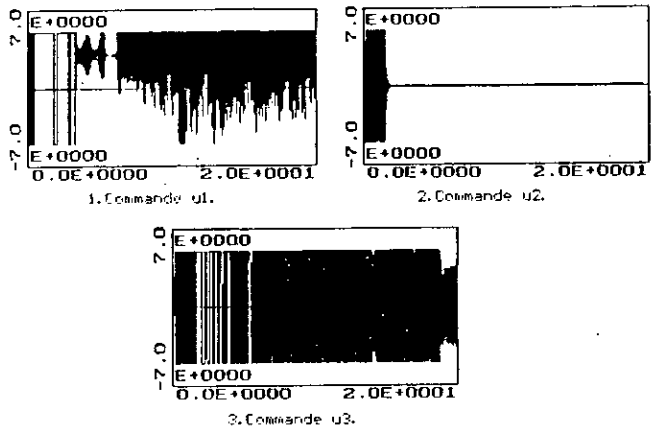
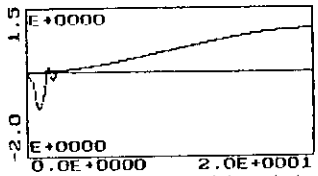
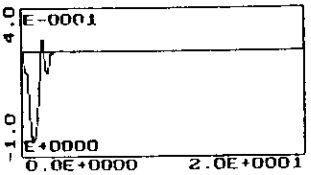


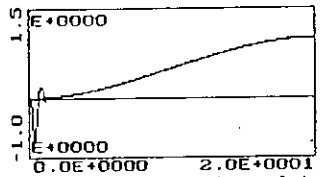
Figure II.27.b Presentation des commandes.



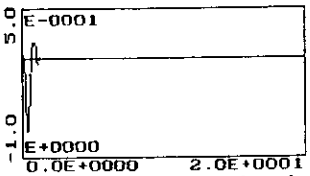
1.Consigne u1 et position q1 de l art1.



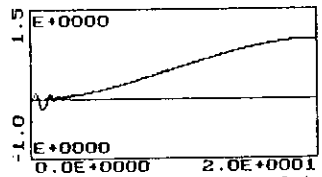
4.Erreur en position de l art1 u1-q1.



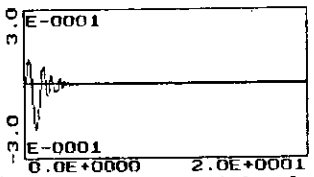
2.Consigne u2 et position q2 de l art2.



5.Erreur en position de l art2 u2-q2.

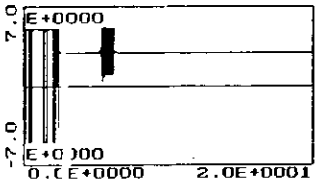


3.Consigne u3 et position q3 de l art1.

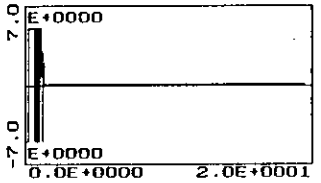


6.Erreur en position de l art3 u3-q3.

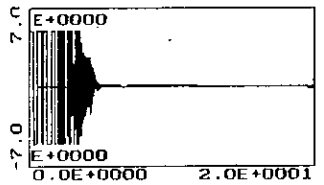
Figure II.28.a Position et erreur en position pour le PP avec Pi=1-0.5q-1+0.06q-2.



1.Commande u1.

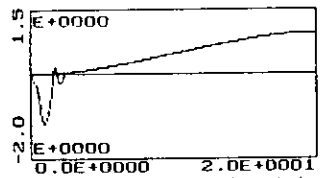


2.Commande u2.

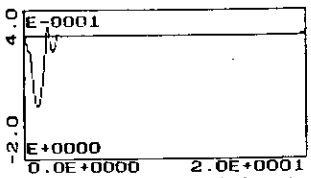


3.Commande u3.

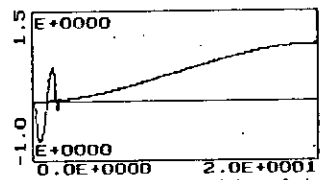
Figure II.28.b Presentation des commandes.



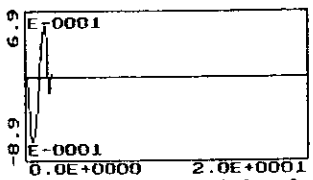
1.Consigne u1 et position q1 de l art1.



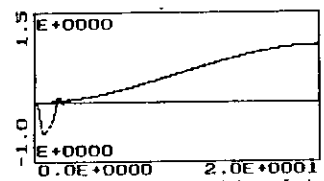
4.Erreur en position de l art1 u1-q1.



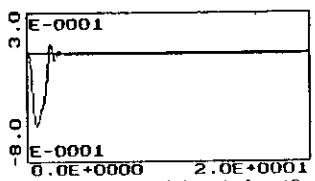
2.Consigne u2 et position q2 de l art2.



5.Erreur en position de l art2 u2-q2.

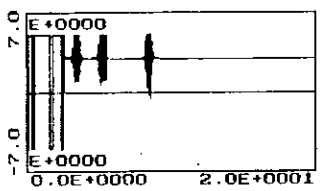


3.Consigne u3 et position q3 de l art1.

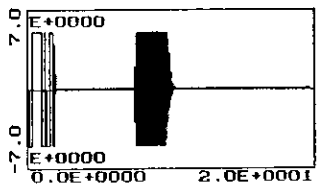


6.Erreur en position de l art3 u3-q3.

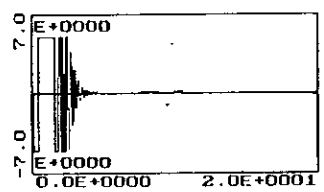
Figure II.29.a Position et erreur en position pour le PP avec Pi=1-0.5q-1+0.06q-2;Emi/Ami=0.035q-1/(1-0.9q-1).



1.Commande u1.

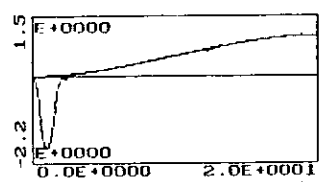


2.Commande u2.

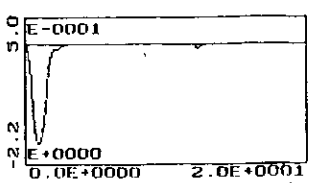


3.Commande u3.

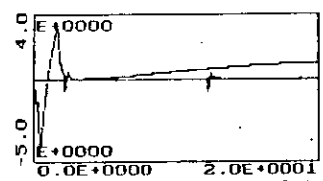
Figure II.29.b Presentation des commandes.



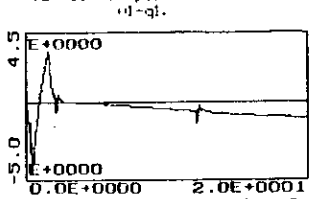
1.Consigne u1 et position q1 de l art1; r1=5e-6;q11=100;q12=0.1.



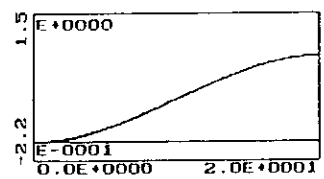
4.Erreur en position de l art1 u1-q1.



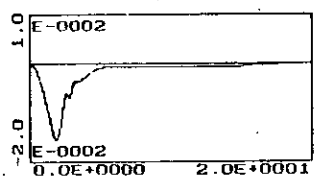
2.Consigne u2 et position q2 de l art2; r2=1e-6;q21=1;q22=0.1.



5.Erreur en position de l art2 u2-q2.

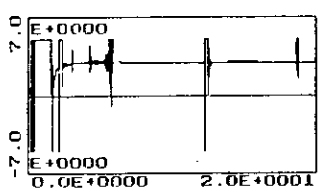


3.Consigne u3 et position q3 de l art1; r3=5e-3;q31=100;q32=1.

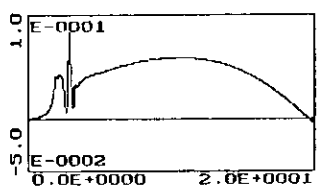


6.Erreur en position de l art2 u3-q3.

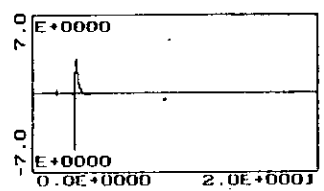
Figure II.30.a Position et erreur en position pour la commande LQG à un pas.



1.Commande u1.

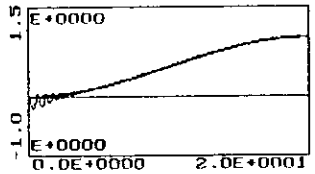


2.Commande u2.

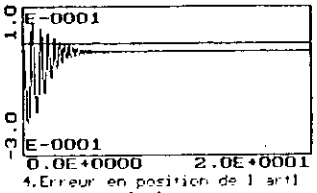


3.Commande u3.

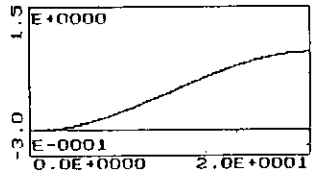
Figure II.30.b Presentation des commandes.



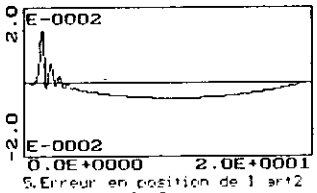
1.Consigne u1 et position q1 de l art1 pour PRPE;Q1=0.007.



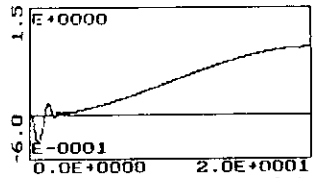
4.Erreur en position de l art1 u1-q1.



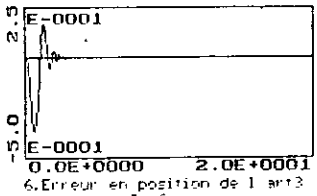
2.Consigne u2 et position q2 de l art2 pour GMV;Q2=0.05.



5.Erreur en position de l art2 u2-q2.

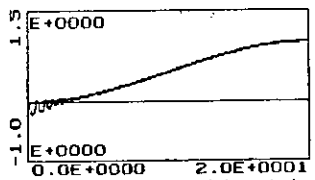


3.Consigne u3 et position q3 de l art1 pour PP;P3=1-0.5q-1+0.06q-2.

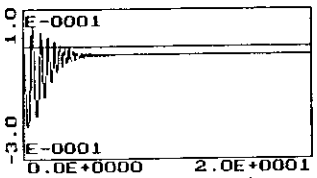


6.Erreur en position de l art3 u3-q3.

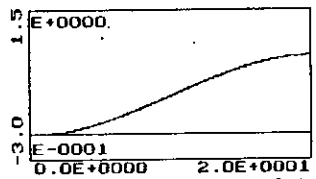
Figure II.31.a Position et erreur en position pour la commande mixte Bm2/Am2=Bm3/Am3=0.095q-1/(1-0.9q-1).



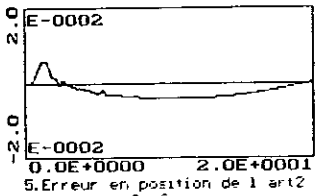
1.Consigne u1 et position q1 de l art1 pour PRPE;Q1=0.007.



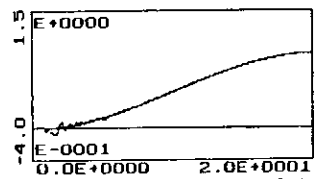
4.Erreur en position de l art1 u1-q1.



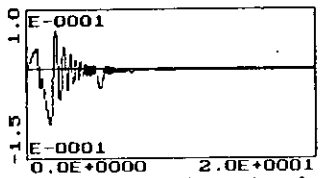
2.Consigne u2 et position q2 de l art2 pour GMV;Q2=0.05.



5.Erreur en position de l art2 u2-q2.

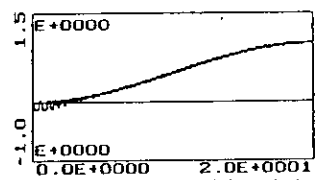


3.Consigne u3 et position q3 de l art1 pour PP;P3=1.

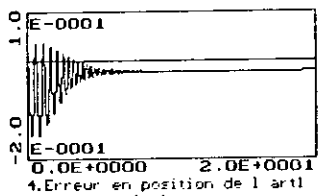


6.Erreur en position de l art3 u3-q3.

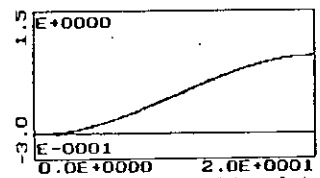
Figure II.32.a Position et erreur en position pour la commande mixte Bm2/Am2=Bm3/Am3=0.095q-1/(1-0.9q-1).



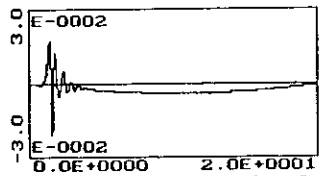
1.Consigne u1 et position q1 de l art1 pour PRPE;Q1=0.007;Te=8ms.



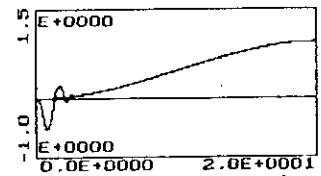
4.Erreur en position de l art1 u1-q1.



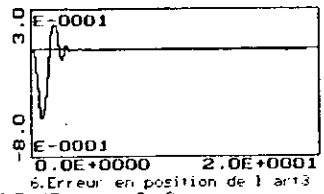
2.Consigne u2 et position q2 de l art2 pour GMV;Q2=0.05;Te=10ms.



5.Erreur en position de l art2 u2-q2.

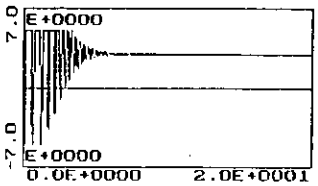


3.Consigne u3 et position q3 de l art1 pour PP;P3=1-0.5q-1+0.06q-2;Te=15ms.

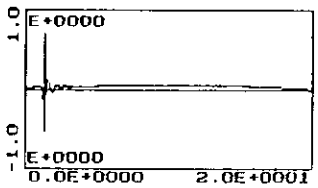


6.Erreur en position de l art3 u3-q3.

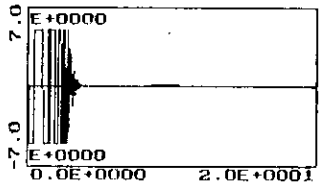
Figure II.33.a Position et erreur en position pour la commande mixte et multirate Bm2/Am2=Bm3/Am3=0.095q-1/(1-0.9q-1).



1.Commande u1.

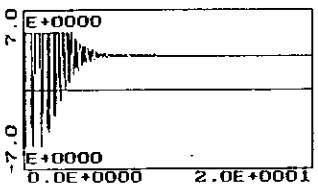


2.Commande u2.

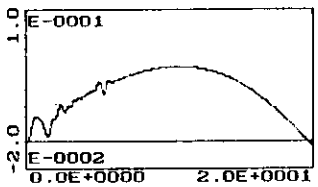


3.Commande u3.

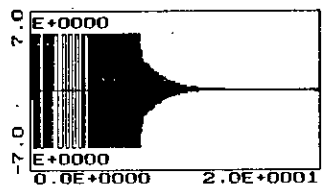
Figure II.31.b Presentation des commandes.



1.Commande u1.

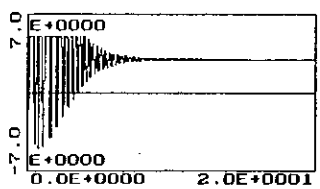


2.Commande u2.

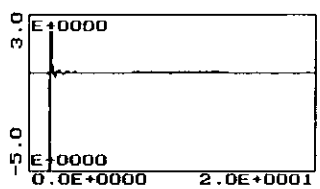


3.Commande u3.

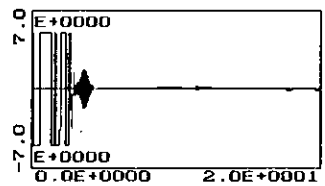
Figure II.32.b Presentation des commandes.



1.Commande u1.

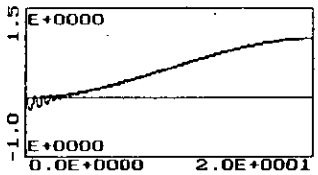


2.Commande u2.

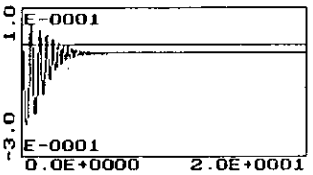


3.Commande u3.

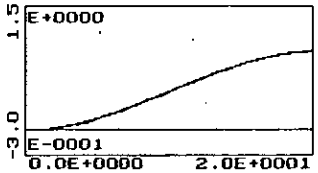
Figure II.33.b Presentation des commandes.



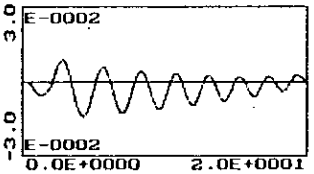
1.Consigne u1 et position q1 de l art1 pour PRPE;Q1=0.007.



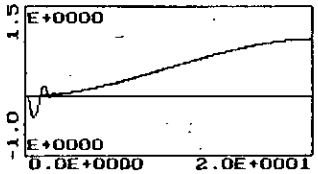
4.Erreur en position de l art1 u1-q1.



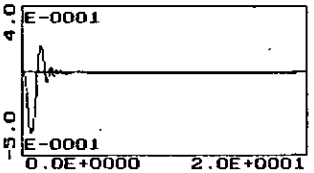
2.Consigne u2 et position q2 de l art2 pour GMV;Q2=0.05.



5.Erreur en position de l art2 u2-q2.

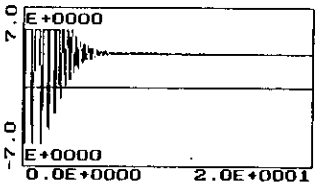


3.Consigne u3 et position q3 de l art1 pour PP;P3=1-0.5q-1+0.06q-2.

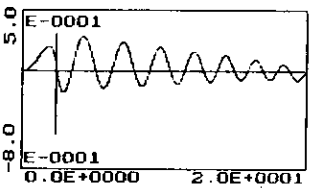


6.Erreur en position de l art3 u3-q3.

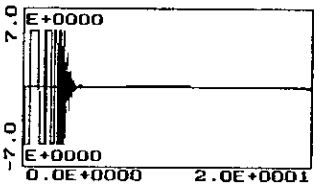
Figure II.34.a Position et erreur en position pour la commande mixte.
Bm2/Am2=Bm3/Am3=0.095q-1/(1-0.9q-1).(variation de 500% du modèle du robot).



1.Commande u1.

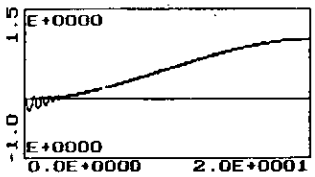


2.Commande u2.

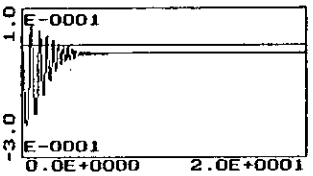


3.Commande u3.

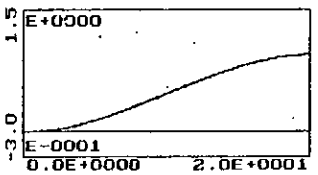
Figure II.34.b Presentation des commandes.



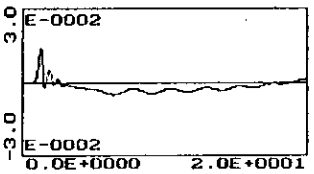
1.Consigne u1 et position q1 de l art1 pour PRPE;Q1=0.007.



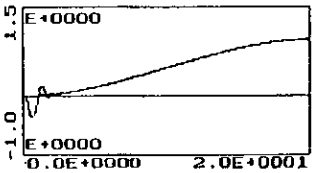
4.Erreur en position de l art1 u1-q1.



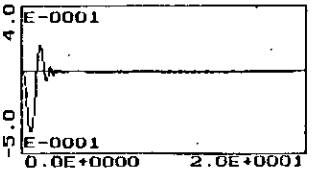
2.Consigne u2 et position q2 de l art2 pour GMV;Q2=0.05.



5.Erreur en position de l art2 u2-q2.

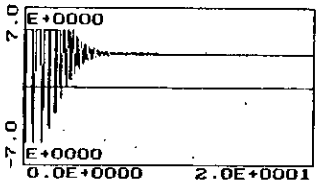


3.Consigne u3 et position q3 de l art1 pour PP;P3=1-0.5q-1+0.06q-2.

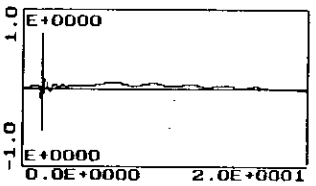


6.Erreur en position de l art3 u3-q3.

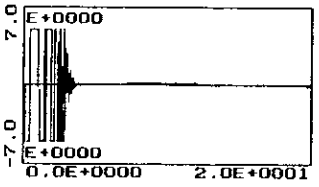
Figure II.35.a Position et erreur en position pour la commande mixte.
Bm2/Am2=Bm3/Am3=0.095q-1/(1-0.9q-1).(variation de 500% à t=5s).



1.Commande u1.

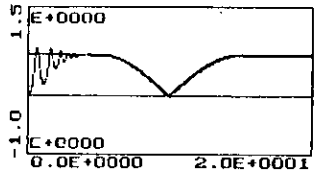


2.Commande u2.

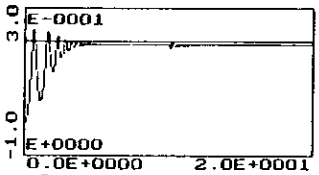


3.Commande u3.

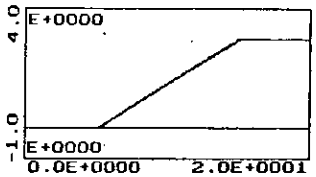
Figure II.35.b Presentation des commandes.



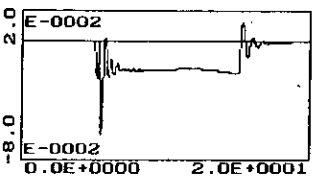
1.Consigne u1 et position q1 de l art1 pour PPPE;Q1=0.007.



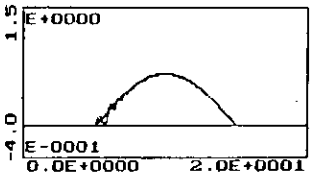
4.Erreur en position de l art1 u1-q1.



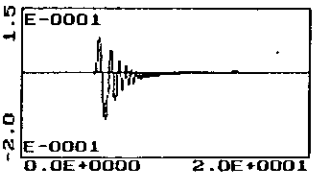
2.Consigne u2 et position q2 de l art2 pour GMW;Q2=0.05.



5.Erreur en position de l art2 u2-q2.

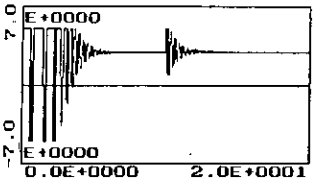


3.Consigne u3 et position q3 de l art1 pour PP;P3=1-0.5q-1+0.06q-2.

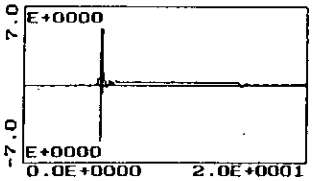


6.Erreur en position de l art3 u3-q3.

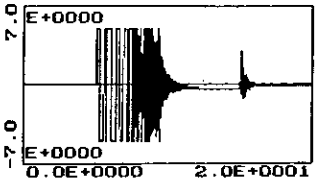
Figure II.36.a Position et erreur en position pour la commande mixte.
Bm2 Am2=Bm3/Am3=0.035q-1/(1-0.9q-1).(consigne fenetre de HUIANI).



1.Commande u1.

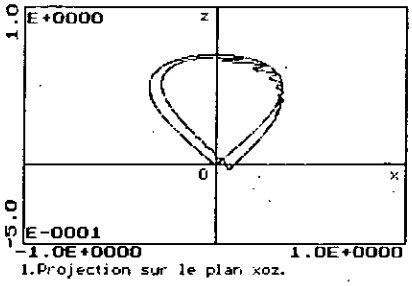


2.Commande u2.

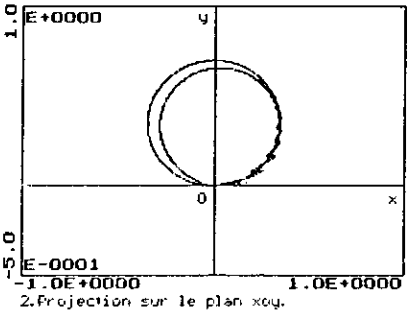


3.Commande u3.

Figure II.36.b Presentation des commandes.



1.Projection sur le plan xoz.



2.Projection sur le plan xoy.

Figure II.36.c Projection des trajectoires désirées et réelles sur les plan principaux.

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 à un pas. Dans la seconde partie, on s'est intéressé à l'application de ces algorithmes pour la commande d'un robot manipulateur et le développement de l'algorithme mixte et multirate.

Le modèle de représentation choisi a permis l'extension des algorithmes monovariables aux cas multivariables. 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. Les techniques d'analyse de convergence pour les algorithmes d'identification n'ont pas été abordées. Elles se trouvent dans la littérature spécialisée.

L'algorithme de commande à GMV adaptative est une solution optimale du problème de commande pour les processus modèlisables 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. Le choix de ces pôles 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 numérique peuvent engendrer des oscillations de la commande, même en imposant une dynamique de référence.

La commande LQC à un pas montre son efficacité même dans le cas des système 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.
La satisfaction de la condition que la paire (Ψ,Γ) soit commandable n'est pas toujours vérifiée, dans le cas adaptatif. Les résultats de simulation montrent que la deuxième articulation ne satisfait pas une telle condition. Dans ce cas, commander cet articulation par LQC, en utilisant un modèle découplé, n'est pas efficace. Il faut alors changer la structure de control de cette dernière et choisir une loi de commande autre que LQC.

Pour tirer profit des performances de ces différents algorithmes, on a 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 tache 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 tache 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 différentes nature (rapidité) pour chaque articulation.

Les résultats de simulation ont montré la robustesse de ces algorithme 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 dans le domaine industriel.

Chapitre III

COMMANDE NONLINEAIRE HYBRIDE AUTO-AJUSTABLE

"Ce que l'on conçoit bien s'annonce clairement.
Et les mots pour le dire arrivent aisément".

BOILEAU

Chapitre III

COMMANDE NON LINEAIRE HYBRIDE AUTO-AJUSTABLE

Introduction.

Dans cette partie, on présente des méthodes de découplage non linéaire pour la réalisation d'asservissements multivariables performants.
Le modèle du robot décrit au chapitre I, est caractérisé par un système d'équation différentielles non linéaires. La connaissance de ce modèle permet l'implèmentation d'une telle commande. Ainsi la détermination du retour non linéaire est essentiellement basé sur le modèle de connaissance. On construit alors un modèle de base (model based) [21] pour la réalisation de cette commande.
L'idée de base de cette commande 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 séveres , 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.

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 sur le modèle du robot. Ceci permet d'utiliser des algorithmes de commande linéaires pour le contrôle du robot, en se basant sur un nouveau modèle moins complexe.
La difficulté majeure dans l'utilisation de cette stratégie est la connaissance de la dynamique du robot. D'une part du fait que certaines composantes du modèle sont extrêmement difficile à déterminer, tels que les coefficients de frottements. D'autre part, l'usure de quelques articulations entraine un changement dans le modèle du robot [21]. L'utilisation d'un algorithme de commande auto-ajustable s'avère nécessaire. On utilise alors dans la boucle externe, un algorithme à STR développé dans le chapitre II. On aboutit alors à la structure défini par la figure III.1.

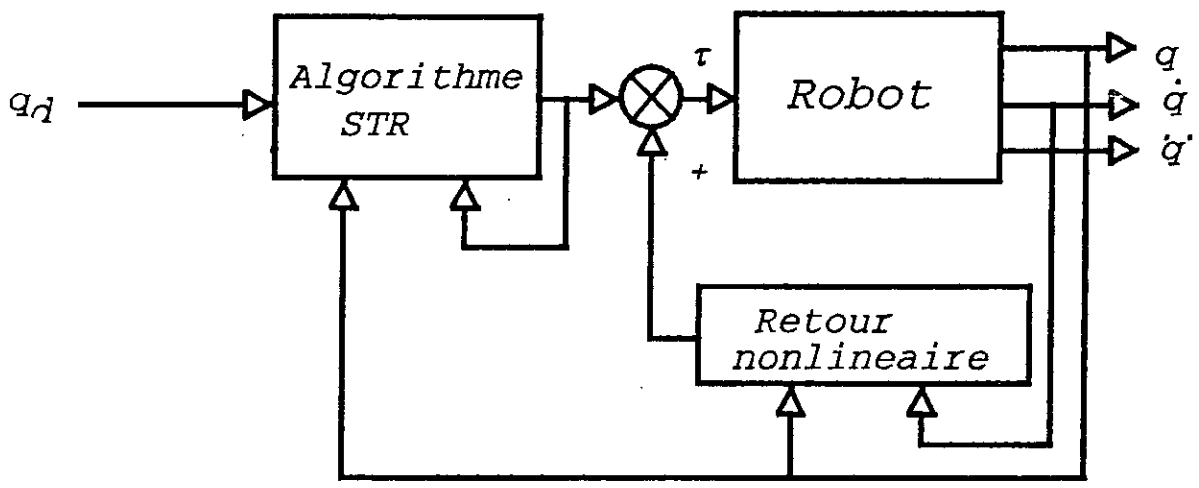


Figure III.1 Commande hybride autoajustable.

On remarque que l'algorithme est composé d'une boucle interne (Modèle Based) et d'une boucle externe (de commande). Dans la boucle interne représentant le modèle de base, le retour non linéaire se calcule à partir du modèle "continu" du robot. On a alors implicitement supposé que le temps de calcul d'un tel retour est très faible. De ce fait, on applique ce retour à des périodes d'échantillonnage adéquatement choisies, de sorte que l'approximation soit raisonnable (On utilise un calculateur rapide). Ainsi la boucle externe considére que la boucle interne est un "autre" modèle continu moins complexe [21]. En considérant cette architecture, la période d'échantillonnage de la dynamique interne doit être plus faible que celle de la commande. De cette façon, on obtient un algorithme hybride, composé d'un retour continu et d'une commande à STR discrète [81]. Le choix de l'algorithme de commande dépend du retour utilisé.

Dans ce qui suit, on présente les différentes méthodes de calcul des retours non linéaires. Le retour par compensation de gravité est exposé dans la section III.1. Dans la section III.2, on présente le découplage non linéaire partiel. Tandis que dans les sections III.3 et III.4, on développe respectivement, le découplage nonolinéaire total et le découplage non linéaire utilisant la géométrie différentielle.

III.1 Compensation de l'effet gravitationnelle.

Dans le pratique, des charges 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 [2], lorsque ce dernier opére dans un environnement spatial . 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 d'Euler Lagrange dans le chapitre I, peut se mettre sous la forme suivante [21]:

$$M(q)\ddot{q}+V(q,\dot{q})+G(q)+f(\dot{q}) = \tau \qquad \text{(III.1)}$$

Où $\tau^T=[\tau_1 \ldots \tau_n]$: couples généralisés.
$q^T=[q_1 \ldots q_n]$: variables articulaires généralisées.
$M(q)$: Matrice d'inertie de dimension nxn.
$V(q,\dot{q})$: 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 par : $g_i(q) = -\sum_{k=1}^{n} m_k g^T U_{ki}\ {}_k r^k$

$f(q)$: vecteur de dimension n exprimant les frottement visqueux.

Pour éliminer l'effet de $G(q)$: dans l'équation (III.1), on choisit le couple généralisé τ, de la manière suivante [2]:

$$\tau = \tau_d+G_0(q) \qquad \text{(III.2)}$$

Où $G_0(q)$: est une approximation de $G(q)$.

Ainsi le modèle du robot défini par (III.1) devient:

$$M(q)\ddot{q}+V(q,\dot{q})+G(q)+f(\dot{q}) = \tau_d+G_0(q) \qquad \text{(III.3)}$$

Dans le cas où l'on connaît "exactement" l'expression analytique de l'effet gravitationnel, on aura :

$$G(q) = G_0(q) \qquad (III.4)$$

Alors l'équation (III.3) devient:

$$M(q)\ddot{q}+V(q,\dot{q})+f(\dot{q}) = \tau_d \qquad (III.5)$$

La structure d'une telle compensation est définie par la figure III.2.

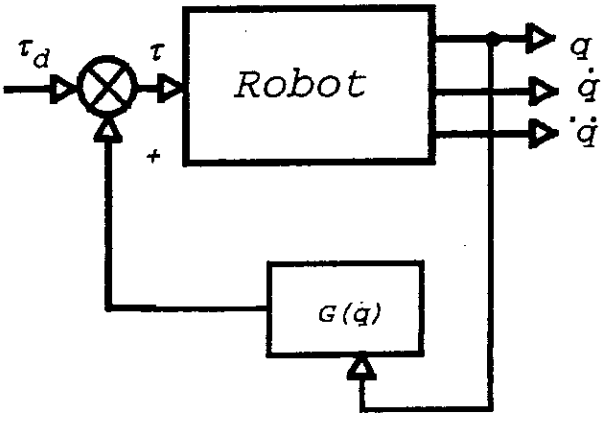


Figure III.2 Compensation des couples de pesanteur.

La compensation des couples dus à la pesanteur est réalisable de deux manières [2]. 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 entraine 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 décrit par la figure III.2, 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 III.1 est assujetti à des erreurs de modélisation. L'utilisation d'un algorithme de commande auto-ajustable s'avère nécessaire. ainsi on aboutit à la structure de commande définie par la figure III.3.

Dans le cas du robot de classe quatre, on a (voir chap I):

$$M(q) = \begin{bmatrix} m_1+m_3 & 0 & 0 \\ 0 & m_3 J^* & 0 \\ 0 & 0 & m_3 \end{bmatrix} \qquad (III.6.1)$$

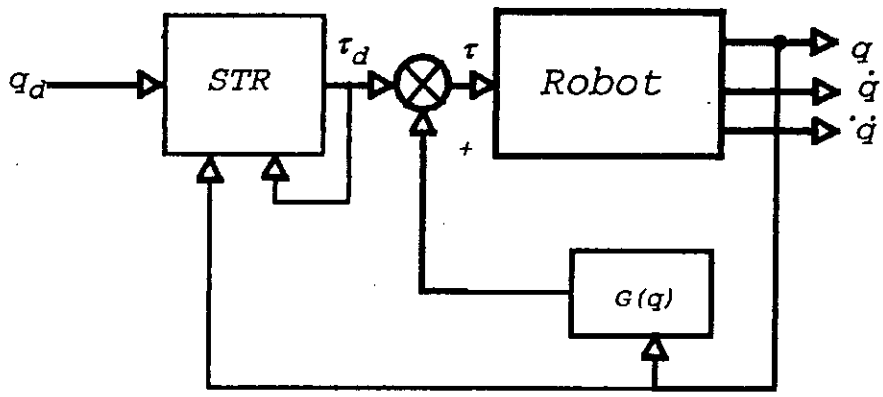


Figure III.3 Commande hybride avec compensation des couples de pesanteur.

$$V^T(q,\dot{q}) = \left[0 \quad 2m_3\left(q_3 - \frac{l_2}{2}\right)\dot{q}_2\dot{q}_3 \quad -m_3\left(q_3 - \frac{l_2}{2}\right)\dot{q}_2^2\right] \qquad (III.6.2)$$

$$G^T(q) = [(m_1+m_3) \quad 0 \quad 0]$$

$$f^T(\dot{q}) = [f_1\dot{q}_1 \quad f_2\dot{q}_2 \quad f_3\dot{q}_3] \qquad (III.6.3)$$

La commande définie par l'équation (III.2) est:

$$\begin{cases} \tau_1 = \tau_{d_1} + (m_1+m_3)\,g \\ \tau_2 = \tau_{d_2} \\ \tau_3 = \tau_{d_3} \end{cases} \qquad (III.7)$$

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 autoajustable pour commander le robot.

III.1.a Résultats de simulation.

Les caractéristiques du robot utilisées dans la simulation sont décrites aux chapitre I.

La figure III.G.1 illustre le comportement du robot lorsque l'effet gravitationnel est compensée. Le retour de compensation est appliqué à une période de 2ms. Le robot ainsi compensé, lui est applique la commande STR mixte, avec une période d'échantillonnage de 10ms, dont les résultats sont illustrés aux figures III.G.2.a, b et c.

Dans le cas où le retour de compensation est calculé à partir d'un modèle erroné de 500%, le comportement du robot en boucle ouverte est présenté dans la figure III.G.3. Avec un tel retour erroné, on applique la commande mixte pour surmonter le problème de cette erreur de modélisation. Les résultats de simulation sont illustrés aux figures III.G.4.a, b et c.

Pour tester la robustesse de la commande hybride à compensation de gravité, on

fait varier soudainement le modèle du robot de 500% à l'instant 5s, dont les résultats sont présentés aux figures III.G.5.a, b et c.

Enfin pour tester la capacité de poursuite de l'algorithme, on applique à ce dernier une consigne calculée à partir de la fenêtre de VIVIANI. Les résultats de simulation pour une telle référence sont consignés aux figures III.G.6.a, b, c et d.

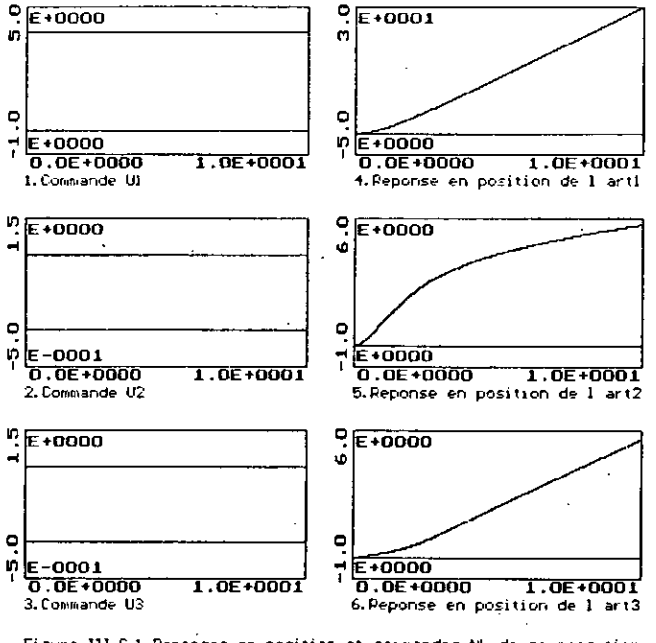


Figure III.6.1 Reponses en position et commandes NL de compensation de gravité pour une entrée unitaire

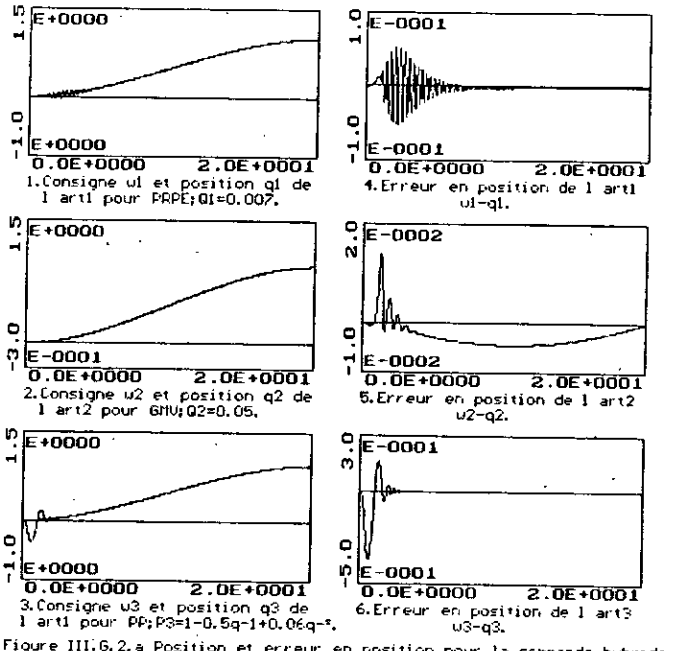


Figure III.6.2.a Position et erreur en position pour la commande hybride à compensation de gravité.(mixte).Bm2/Am2=Bm3/Am3=0.095q-1/(1-0.9q-1).

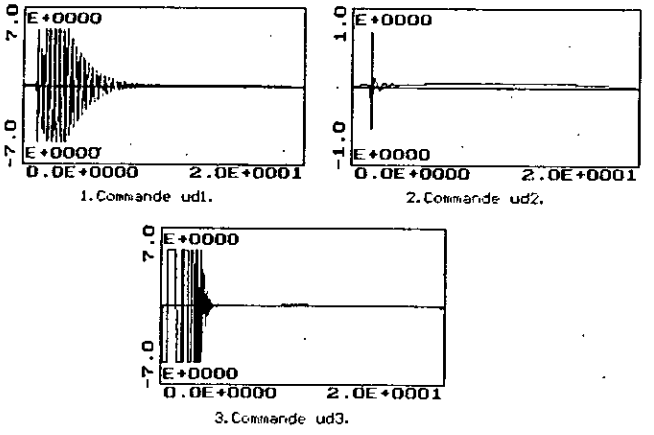


Figure III.6.2.b Presentation des commandes du STR.

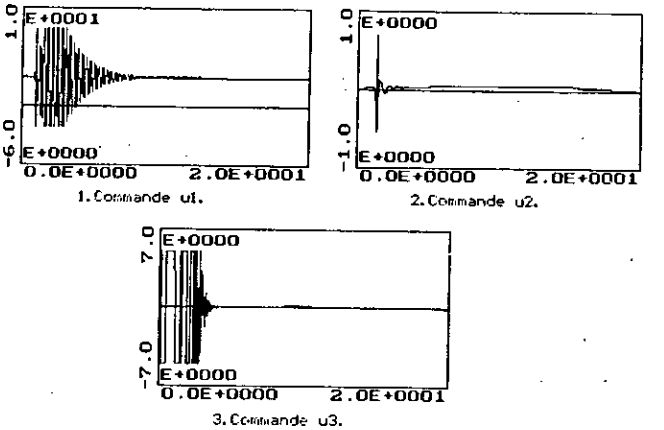
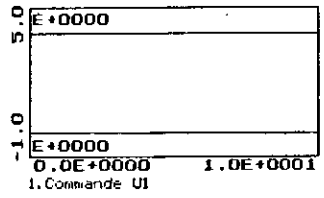
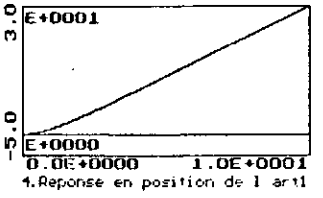


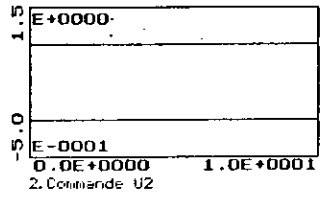
Figure III.6.2.c Presentation des commandes de compensation de gravité.



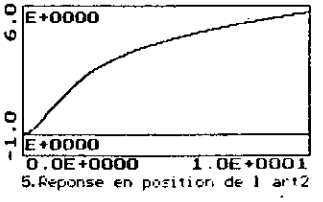
1. Commande U1



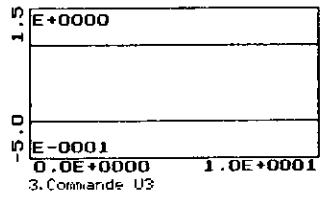
4. Reponse en position de l art1



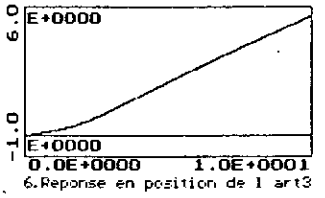
2. Commande U2



5. Reponse en position de l art2

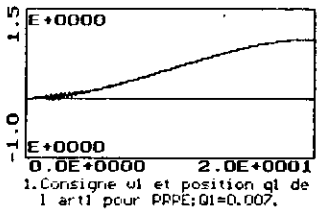


3. Commande U3

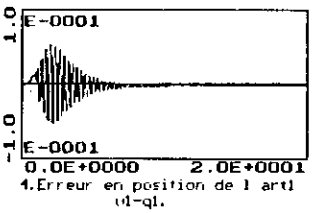


6. Reponse en position de l art3

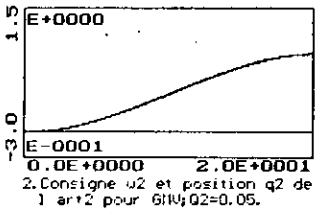
Figure III.6.3 Reponses en position et commandes NL de compensation de gravité pour une entrée unitaire. (variation de 500% du retour NL).



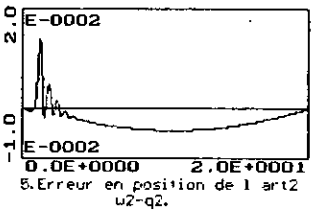
1. Consigne u1 et position q1 de l art1 pour PRPE; Q1=0.007.



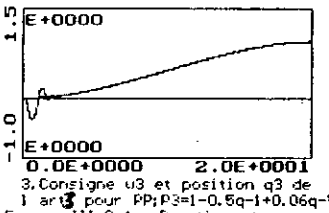
4. Erreur en position de l art1 u1-q1.



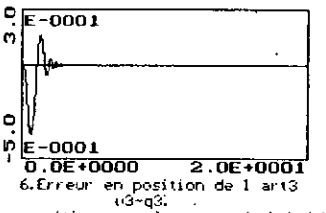
2. Consigne u2 et position q2 de l art2 pour GIW; Q2=0.05.



5. Erreur en position de l art2 u2-q2.

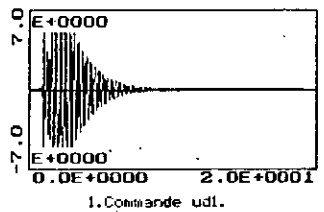


3. Consigne u3 et position q3 de l art3 pour PP; P3=1-0.5q-1+0.06q-2.

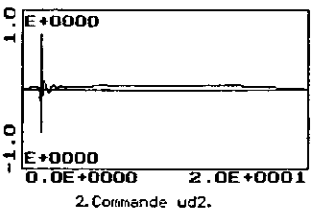


6. Erreur en position de l art3 u3-q3.

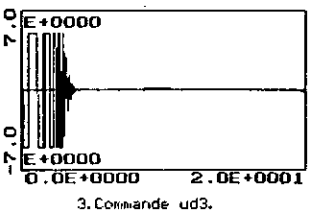
Figure III.6.4.a Position et erreur en position pour la commande hybride à compensation de gravité. (mixte). Bm2/Am2=Bm3/Am3=0.095q-1/(1-0.9q-1). (variation du retour NL de 500%).



1. Commande ud1.

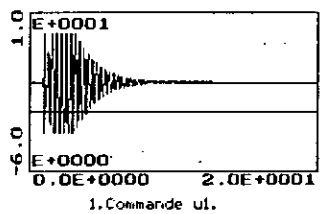


2. Commande ud2.

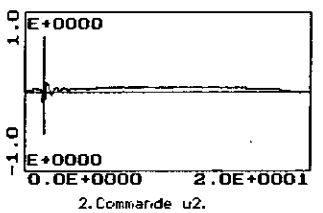


3. Commande ud3.

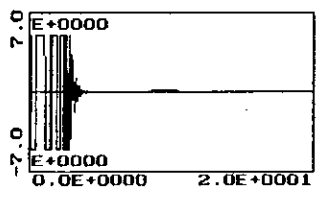
Figure III.6.4.b Presentation des commandes du STR.



1. Commande u1.

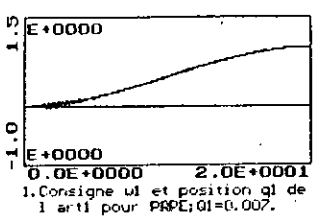


2. Commande u2.

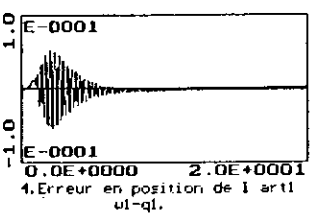


3. Commande u3.

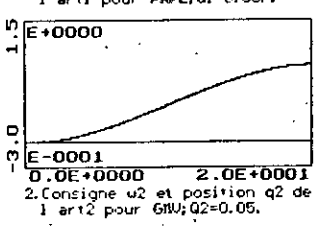
Figure III.6.4.c Presentation des commandes de compensation de gravité.



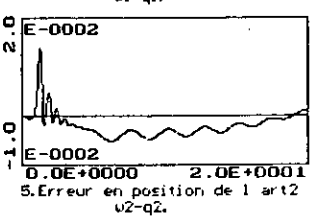
1. Consigne u1 et position q1 de l art1 pour PRPE; Q1=0.007.



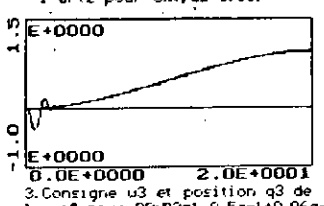
4. Erreur en position de l art1 u1-q1.



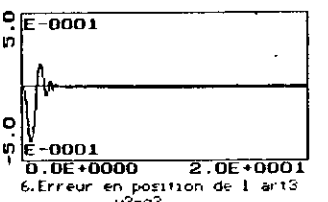
2. Consigne u2 et position q2 de l art2 pour GIW; Q2=0.05.



5. Erreur en position de l art2 u2-q2.

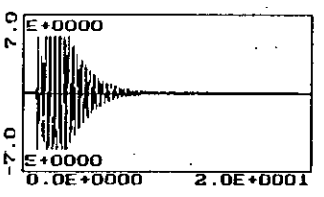


3. Consigne u3 et position q3 de l art3 pour PP; P3=1-0.5q-1+0.06q-2.

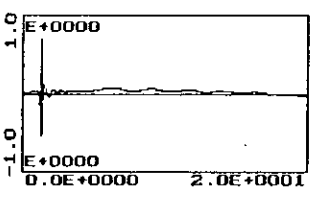


6. Erreur en position de l art3 u3-q3.

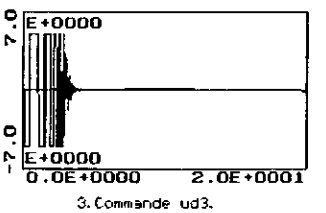
Figure III.6.5.a Position et erreur en position pour la commande hybride à compensation de gravité. (mixte). Bm2/Am2=Bm3/Am3=0.095q-1/(1-0.9q-1). (variation du modele du robot à t=5s de 500%).



1. Commande ud1.

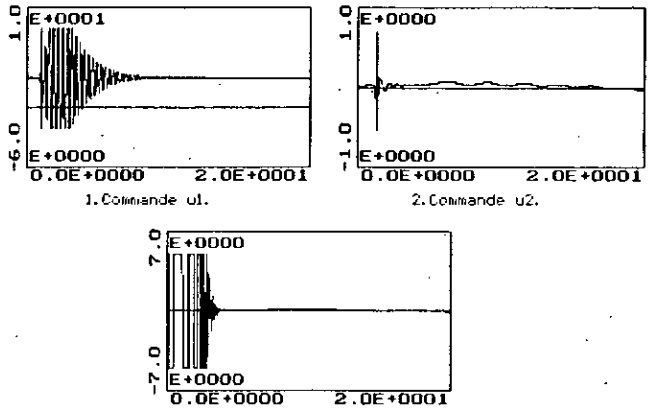


2. Commande ud2.



3. Commande ud3.

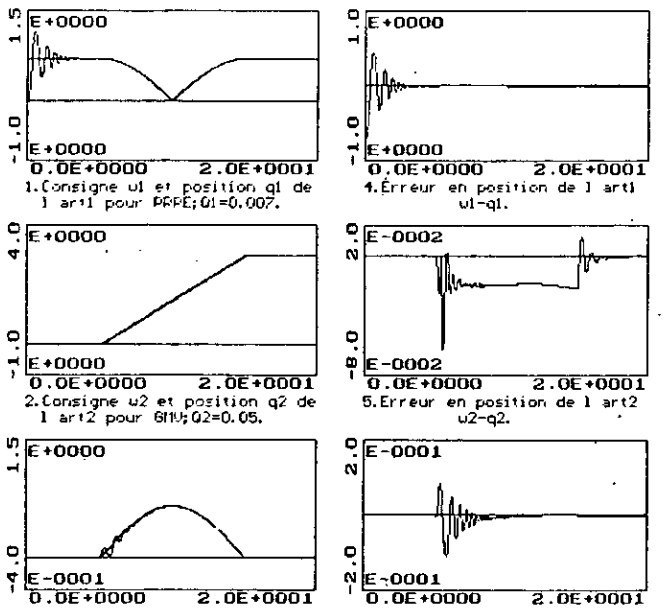
Figure III.6.5.b Presentation des commandes STR.



1. Commande u1. 2. Commande u2.

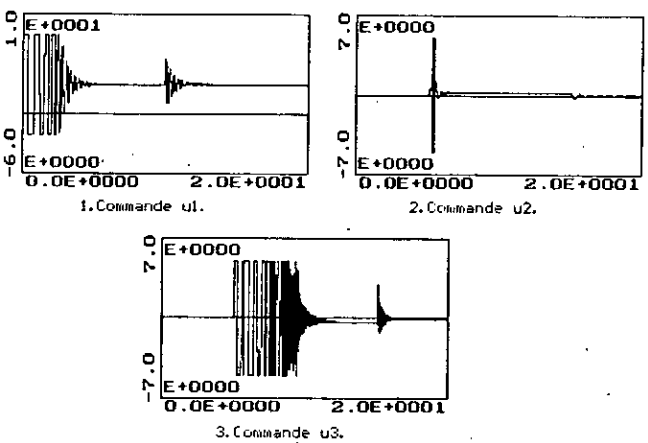
3. Commande u3.

Figure III.6.5.c Presentation des commandes de compensation de gravité.



1. Consigne u1 et position q1 de l'art1 pour PRBF; Q1=0.007.
4. Erreur en position de l'art1 u1-q1.
2. Consigne u2 et position q2 de l'art2 pour GMU; Q2=0.05.
5. Erreur en position de l'art2 u2-q2.
3. Consigne u3 et position q3 de l'art3 pour PP; P3=1-0.5q-1+0.06q-2.
6. Erreur en position de l'art3 u3-q3.

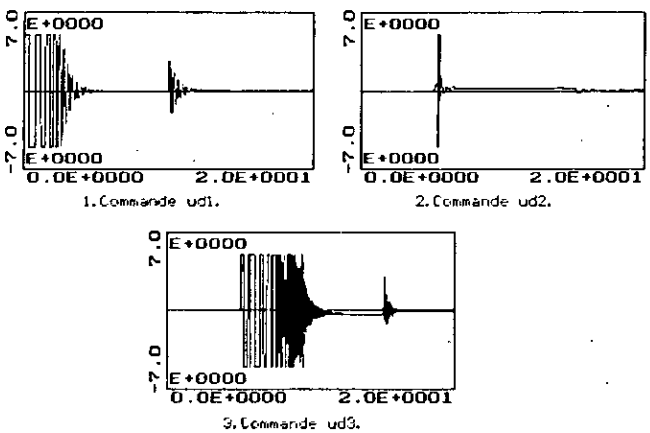
Figure III.6.6.a Position et erreur en position pour la commande hybride à compensation de gravité. (mixte), Bm2/Am2=Bm3/Am3=0.095q-1/(1-0.9q-1). (Consigne fenetre de WUIANI).



1. Commande u1. 2. Commande u2.

3. Commande u3.

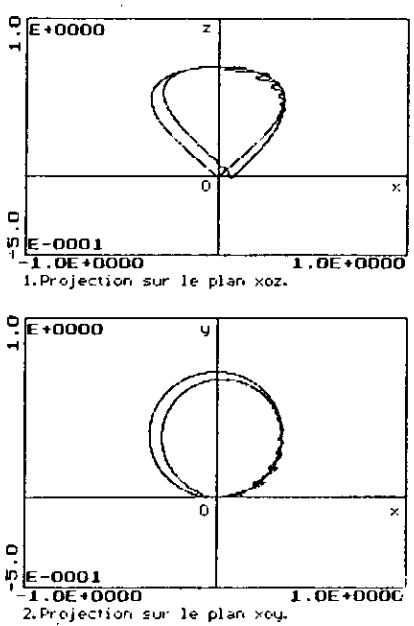
Figure III.6.6.c Presentation des commandes de compensation de gravité.



1. Commande ud1. 2. Commande ud2.

3. Commande ud3.

Figure III.6.6.b Presentation des commandes STR.



1. Projection sur le plan xoz.

2. Projection sur le plan xoy.

Figure II.6.6.d Projection des trajectoires désirées et réelles sur les plan principaux.

III.2 Découplage non linéaire partiel.

L'idée de base d'une telle commande est de compenser tous les effets des non linéarités existantes 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 (III.1), pour la quasi linéarisation de ce modèle, le couple de commande τ peut être pris sous la forme suivante [2]:

$$\tau = \tau_d + V_0(q,\dot{q}) + G_0(q) + f_0(\dot{q}) \qquad \text{(III.8)}$$

Où V_0, G_0 *et* f_0 sont des approximations de V, G *et* f. Ces derniers sont calculés, à partir de q *et* $\dot{q}$.
τ_d: couple de commande.

En commandant le robot à partir du couple défini par (III.8) et en admettant que:

$$V_0(q,\dot{q}) \simeq V(q,\dot{q}) ;\ G_0(q) \simeq G(q) ;\ f_0(\dot{q}) \simeq f(\dot{q}) \qquad \text{(III.9)}$$

L'équation du robot définit par (III.1) devient:

$$M(q)\ddot{q} = \tau_d \qquad \text{(III.10)}$$

Ces opérations de compensation non linéaire partielle sont décrites par la figure III.4.

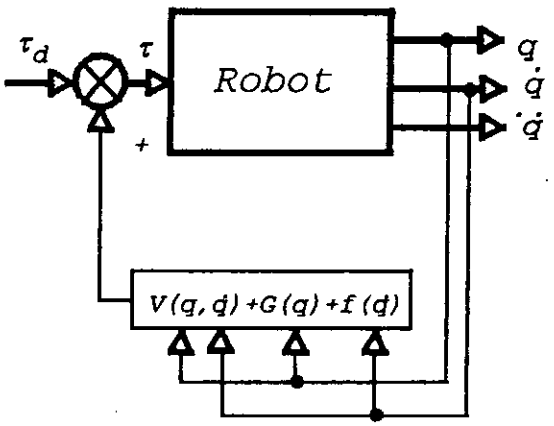


Figure III.4 Découplage nonlinéaire partiel.

Le modèle ainsi compensé est quasiment découplé, car la matrice $M(q)$ est définie par [1]:

$$M_{ij} = \sum_{k=\max(i,j)}^{n} trace(U_{kj} J_k U_{ki}^T) ;\quad i,j=1,3 \qquad \text{(III.11)}$$

La matrice ainsi définie n'est pas forcement 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 par compensation non linéaire partielle, les termes V_0, G_0 *et* f_0 doivent être calculés en temps réel. Ceci exige 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 (III.10), 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 (STR), est d'une importance primordiale, du fait que le calcul du couple défini par (III.8), 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 entraine un modèle complexe mais non découplé [21].
La structure de l'algorithme de commande auto-adaptative hybride est défini par la figure III.5.

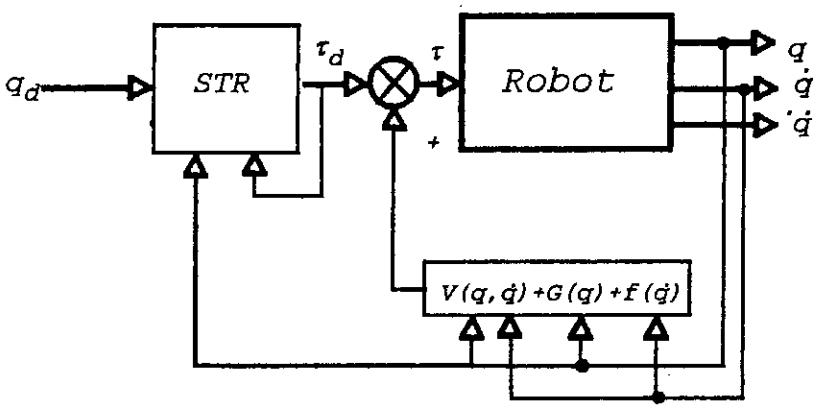


Figure III.5 Commande hybride avec découplage nonlinéaire partiel.

Dans le cas du robot de classe quatre, la matrice M(q) est une matrice diagonale à coefficients variables (équation (III.6.1)). Donc le découplage dans ce cas s'avère d'un avantage remarquable. La commande définie par l'équation (III.8) est:

$$\begin{cases} \tau_1 = \tau_{d_1} + (m_1+m_3)\,g + f_1\dot{q}_1 \\ \tau_2 = \tau_{d_2} + 2m_3\left(q_3 - \frac{l_2}{2}\right)\dot{q}_2\dot{q}_3 + f_2\dot{q}_2 \\ \tau_3 = \tau_{d_3} - m_3\left(q_3 - \frac{l_2}{2}\right) + f_3\dot{q}_3 \end{cases} \qquad \text{(III.12)}$$

avec $\tau_i = k_i u_i \quad i=1,3$.
Où k_i *et* u_i définis par (III.7.1).

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.

III.2.a Résultats de simulation.

Le robot utilisé dans la simulation est de classe quatre, défini au chapitre I. La figure III.DP.1 montre le comportement du robot en BO, à une entrée unitaire, lorsque ce dernier est partiellement découplé.

Le retour non linéaire est appliqué au robot chaque 2ms. Le modèle du robot ainsi découplé est commandé avec la commande STR. En premier lieu, la commande par PRPE est appliquée au modèle découplé, dont les résultats sont illustrés aux figures III.DP.2.a, b et c. Les figures III.DP.3.a, b et c montrent les résultats de simulation lorsque les pondérations sont changés à $\lambda_1=0.0007$ *et* $\lambda_2=0.005$. Pour améliorer le comportement de la troisième articulation on introduit un effet intégrateur défini par $Q_3=0.005(1-q^{-1})$, dont les résultats sont consignés aux figures III.DP.4.a, b et c.

La figure III.DP.5 montre le comportement du robot en BO, lorsque Le retour non linéaire est calculé à partir d'un modèle erroné de 500% du modèle originale. Avec l'algorithme à PRPE, on commande le robot dont les résultats sont consignés dans les figures III.DP.6.a, b et c.

Pour tester la robustesse de la commande hybride à découplage non linéaire, on introduit une variation paramétrique soudaine à 10ms, d'une valeur de 500%. Les résultats d'une telle simulation sont présentés aux figures III.DP.7.a, b et c.

La capacité de poursuite de l'algorithme hybride à découplage partiel est testé, en appliquant le consigne de la fenêtre de VIVIANI, dont les résultats sont illustrés aux figures III.DP.8.a, b, c et d.

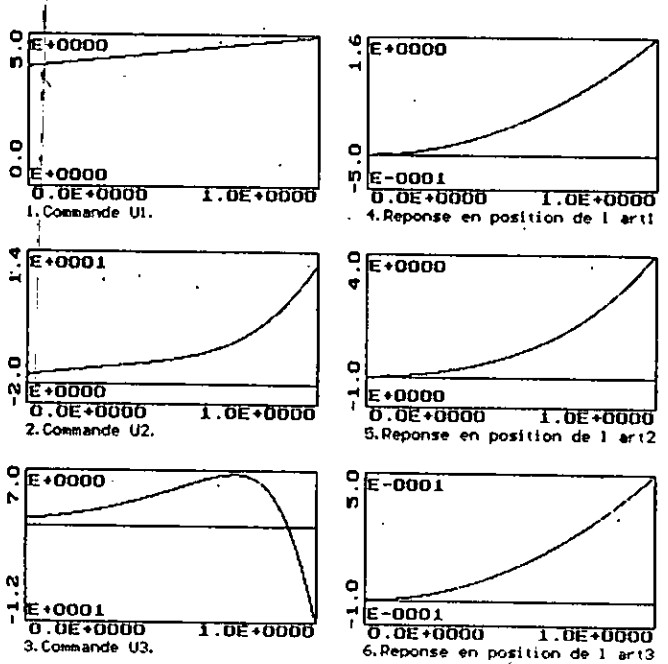


Figure III.DP.1 Reponses en position et commandes du decouplage partiel pour une entrée unitaire.

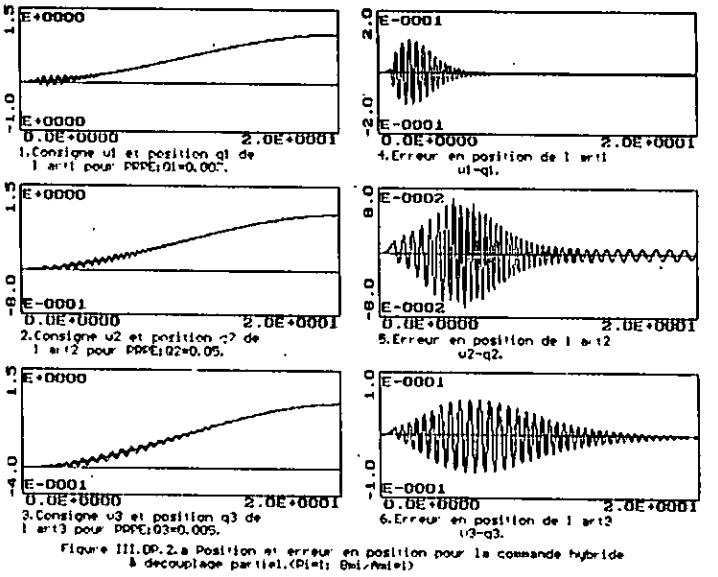


Figure III.DP.2.a Position et erreur en position pour la commande hybride à decouplage partiel.(Pi=1; Bmi/Amie1)

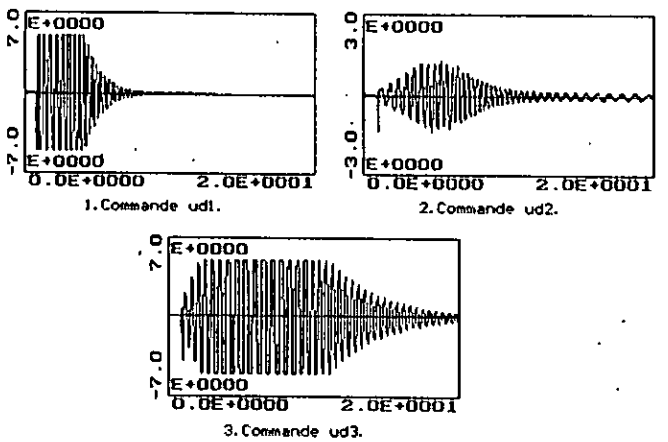


Figure III.DP.2.b Presentation des commandes STR.

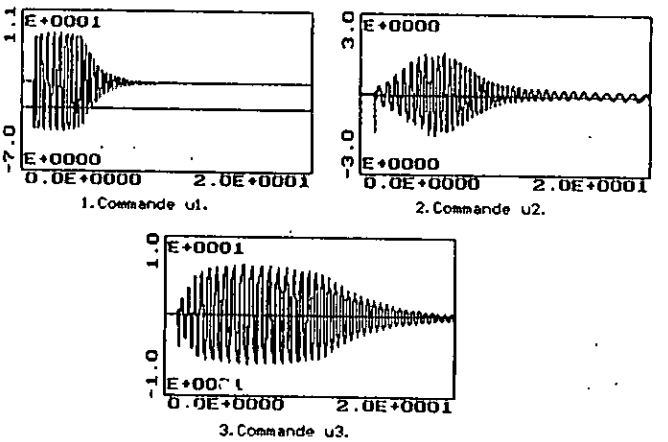


Figure III.DP.2.c Presentation des commandes du decouplage NL partiel.

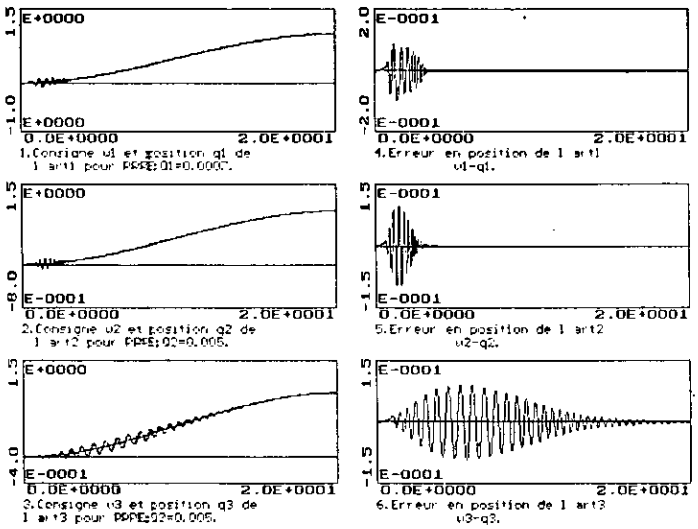


Figure III.DP.3.a Position et erreur en position pour la commande hybride à découplage partiel.(Pi=1; Bmi/Ami=1)

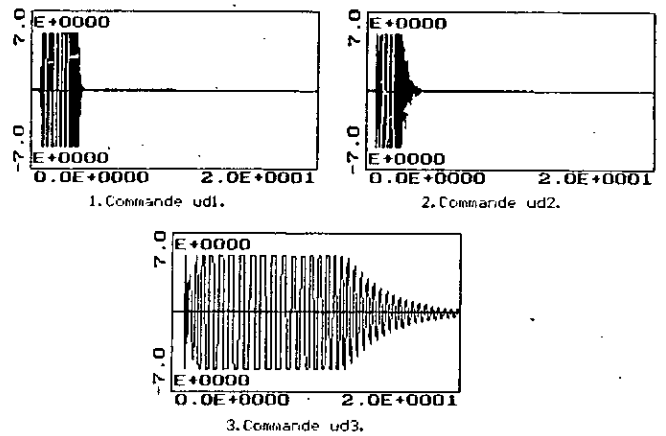


Figure III.DP.3.b Presentation des commandes STR.

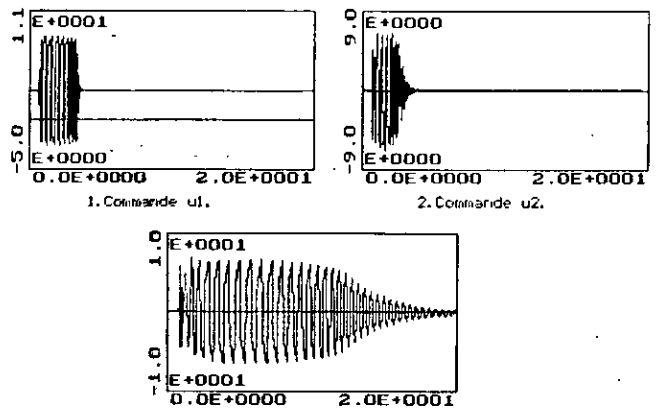


Figure III.DP.3.c Presentation des commandes du decouplage NL partiel.

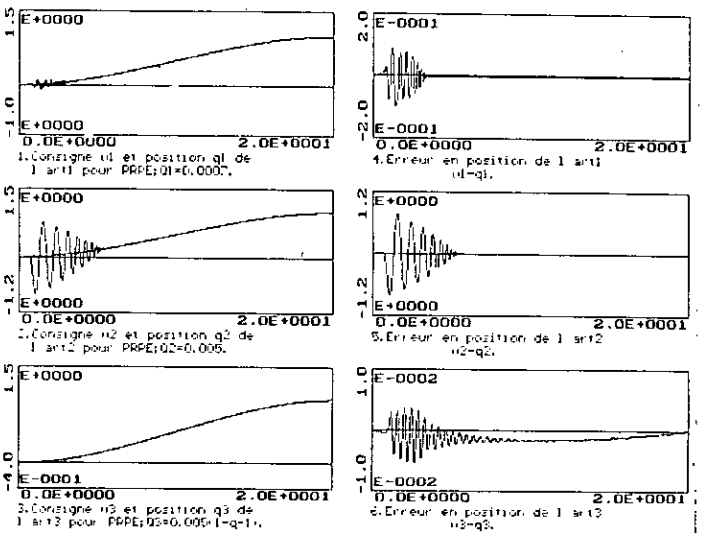


Figure III.DP.4.a Position et erreur en position pour la commande hybride à découplage partiel.(Pi=1; Bmi/Ami=1)

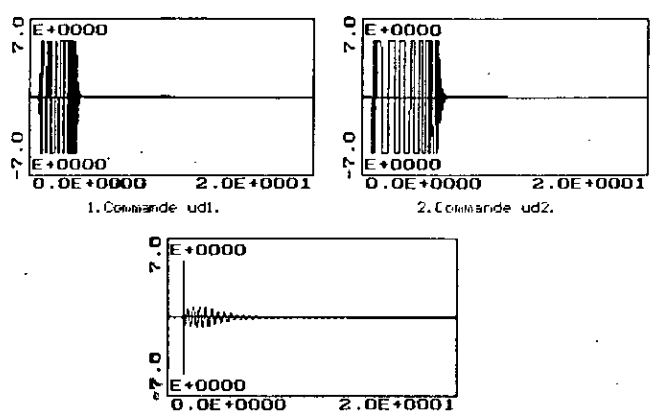


Figure III.DP.4.b Presentation des commandes STR.

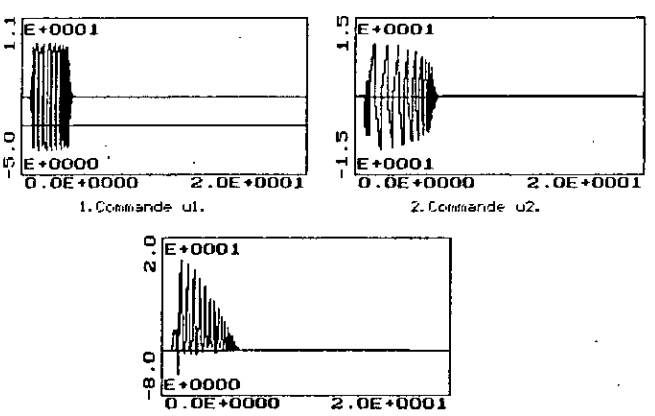


Figure III.DP.4.c Presentation des commandes du decouplage NL partiel.

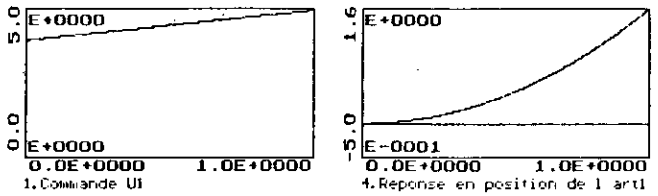


Figure III.DP.5.a Reponses en position et commandes NL du decouplage partiel pour une entrée unitaire.(variation de 500% du modele du robot)

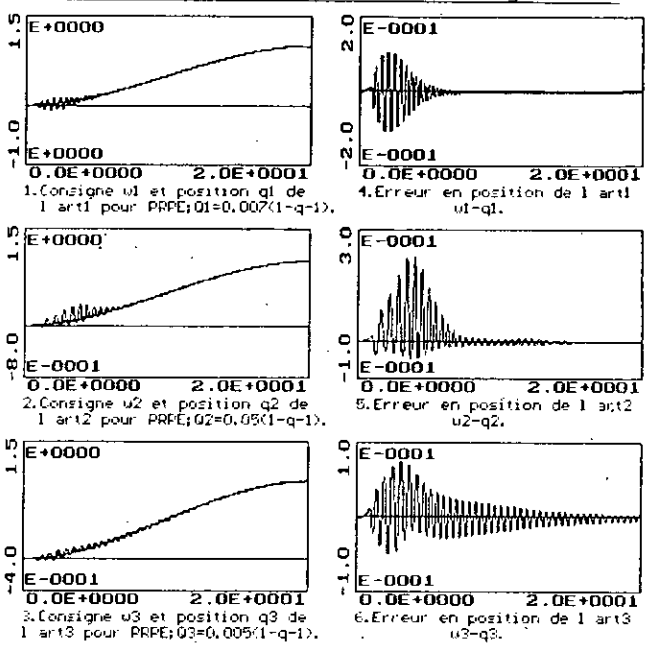


Figure III.DP.6.a Position et erreur en position pour la commande hybride à decouplage partiel.(Pi=1; Bmi/Ami=1; variation du retour NL de 500%)

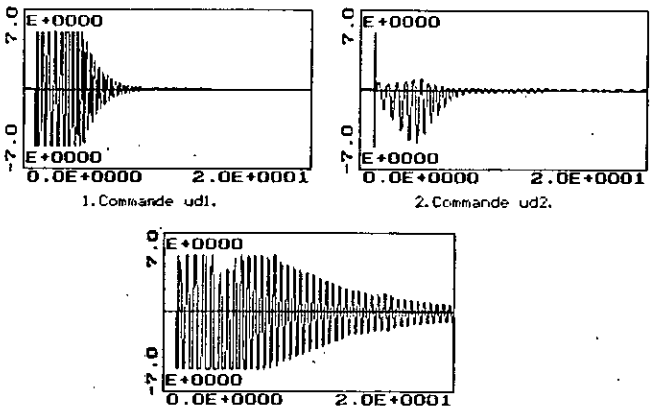


Figure III.DP.6.b Presentation des commandes STR.

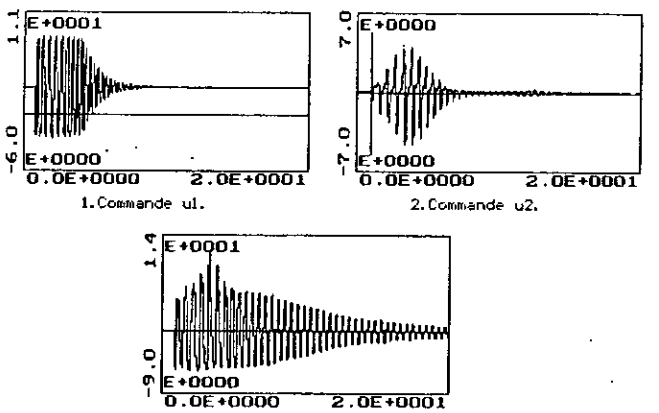


Figure III.DP.6.c Presentation des commandes du decouplage NL partiel.

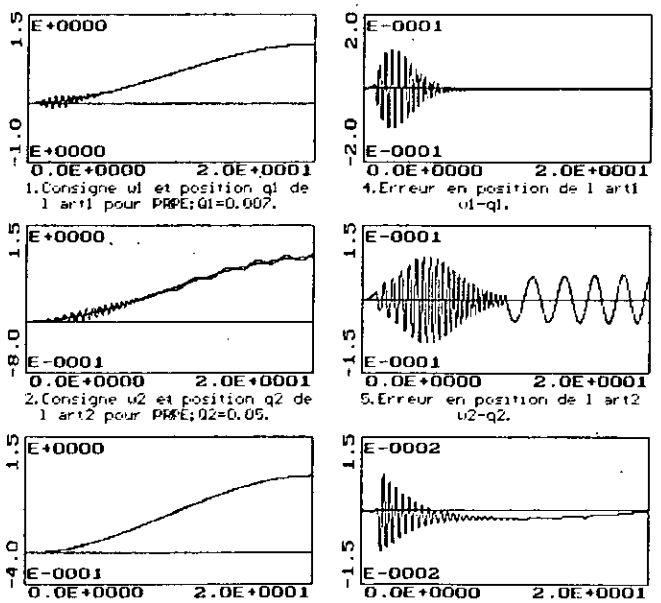


Figure III.DP.7.a Position et erreur en position pour la commande hybride à decouplage partiel.
(Pi=1; Bmi/Ami=1; variation du modele du robot de 500% à t=10s.)

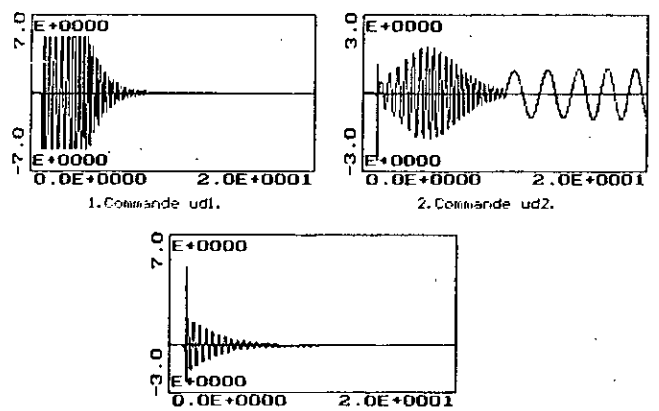
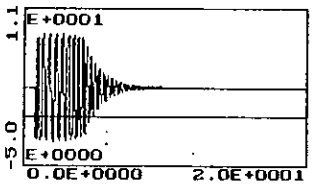
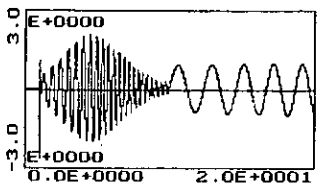


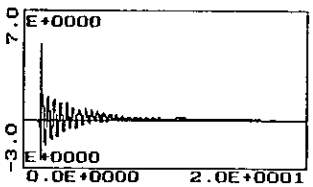
Figure III.DP.7.b Presentation des commandes STR.



1.Commande u1.

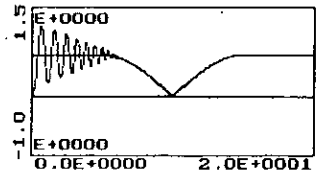


2.Commande u2.

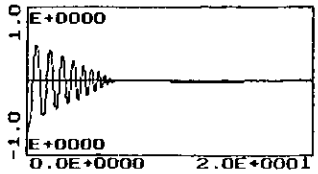


3.Commande u3.

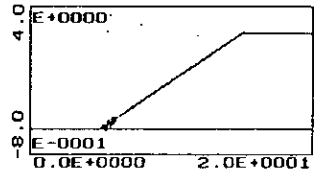
Figure III.DP.7.c Presentation des commandes du decouplage NL partiel.



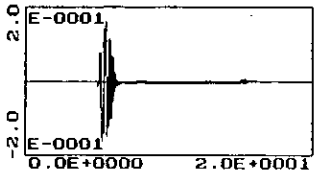
1.Consigne u1 et position q1 de l'art1 pour PRPE; Q1=0.0007.



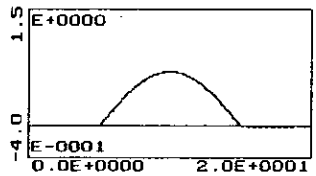
4.Erreur en position de l'art1 u1-q1.



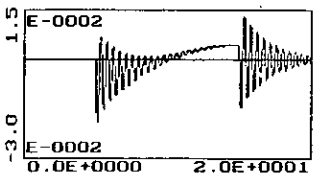
2.Consigne u2 et position q2 de l'art2 pour PRPE; Q2=0.005.



5.Erreur en position de l'art2 u2-q2.

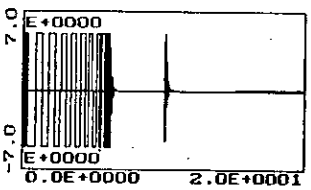


3.Consigne u3 et position q3 de l'art3 pour PRPE; Q3=0.005.(1-q-1).

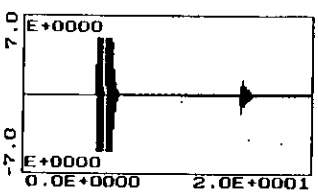


6.Erreur en position de l'art3 u3-q3.

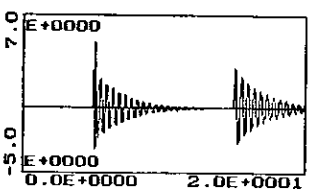
Figure III.DP.8.a Position et erreur en position pour la commande hybride à decouplage partiel.(Pi=1; Bmi/Ami=1; consigne fenetre de UTIfANIL.)



1.Commande ud1.

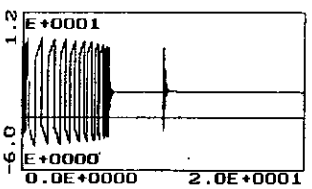


2.Commande ud2.

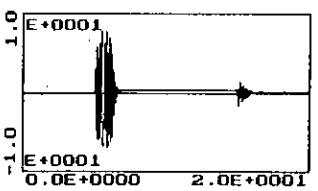


3.Commande ud3.

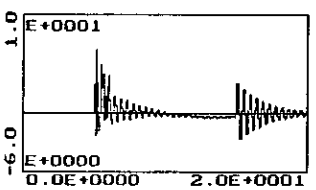
Figure III.DP.8.b Presentation des commandes STR.



1.Commande u1.

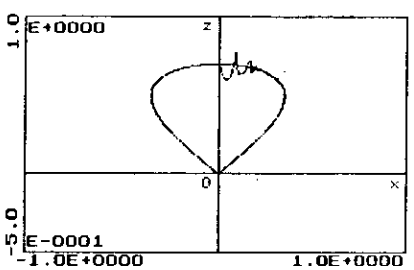


2.Commande u2.

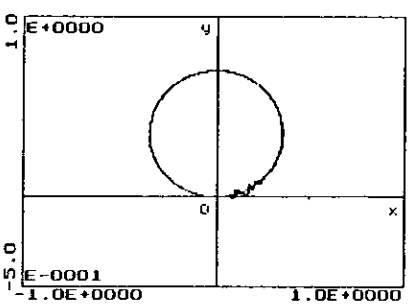


3.Commande u3.

Figure III.DP.8.c Presentation des commandes du decouplage NL partiel.



1.Projection sur le plan xoz.



2.Projection sur le plan xoy.

Figure III.DP.8.d Projection des trajectoires désirées et réelles sur les plan principaux.

III.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 [21]:

$$\tau = \Omega\tau_d + \beta \qquad \text{(III.13)}$$

Où τ *et* τ_d sont définis précédemment.
β : vecteur de dimension n.
Ω : matrice de dimension nxn.

Si on choisit correctement Ω *et* β, 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 (III.1). En utilisant (III.13) et (III.1), on peut choisir Ω *et* β de la manière suivante [21]:

$$\begin{cases} \Omega = M_0(q) \\ \beta = V_o(q,\dot{q}) + G_0(q) + f_0(\dot{q}) \end{cases} \qquad \text{(III.14)}$$

Où M_0, V_0, G_0 *et* f_0: des approximations de M, V, G *et* f. Ils sont calculés en temps réel.

Le schéma d'une telle commande est représenté sur la figure III.6.

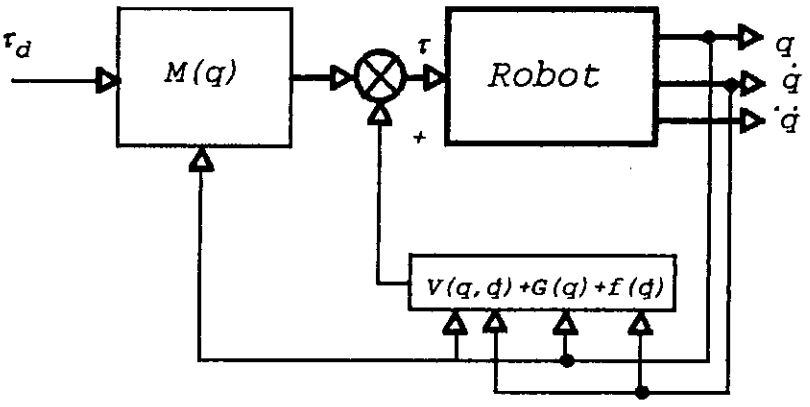


Figure III.6 Découplage et linéarisation.

La loi de commande est définie par:

$$\tau = M_0(q)\tau_d + V_0(q,\dot{q}) + G_0(\dot{q}) + f_0(\dot{q}) \qquad \text{(III.15)}$$

Dans le cas où M_0=M; V_0=V ; G_0=G *et* f_0=f; les équations (III.1) et (III.15) donnent:

$$\tau_d = \ddot{q} \qquad \text{(III.16)}$$

On remarque à partir de (III.16), que le robot se comportera comme un double intégrateur. Ainsi le découplage est total , ce qui permet l'utilisation des algorithmes monovariables continus, non adaptatifs du type PID (voir chapitre V). Dans le cas où la condition sur l'équation (III.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 continue 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 DARMA). La structure de réglage hybride est illustrée dans la figure III.7.

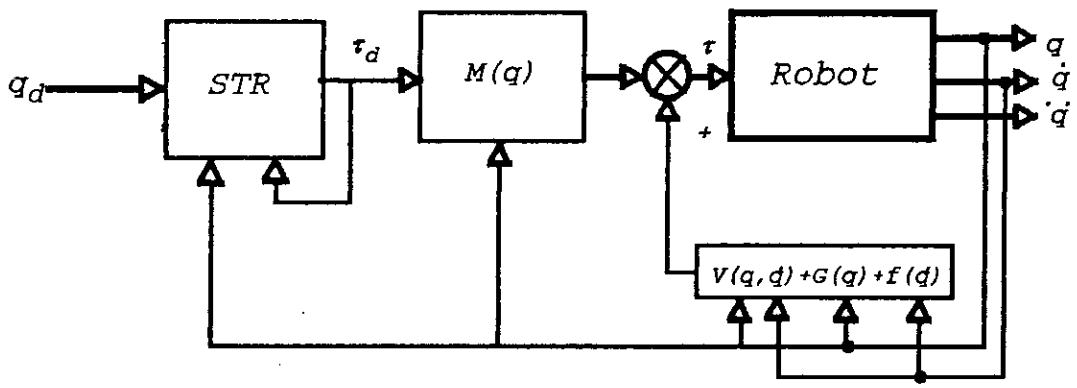


Figure III.7 Commande hybride par découplage et linéarisation.

Dans le cas du robot classe 4, la commande définie par (III.15) devient:

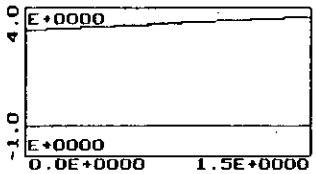
$$\begin{cases} \tau_1 = (m_1+m_3)\,\tau_{d_1} + (m_1+m_3)\,g + f_1\dot{q}_1 \\ \tau_2 = m_3 J^* \tau_{d_2} + 2m_3\left(q_3 - \frac{l_2}{2}\right)\dot{q}_2\dot{q}_3 + f_2\dot{q}_2 \\ \tau_3 = m_3\tau_{d_3} - m_3\left(q_3 - \frac{l_2}{2}\right) + f_3\dot{q}_3 \end{cases} \qquad \text{(III.17)}$$

avec $\tau_i = k_i u_i \quad i=1,3.$
Où k_i *et* u_i définis par (III.7.1).

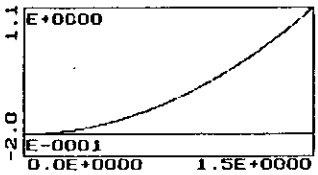
III.3.a Résultats de simulation.

Le robot utilisé dans la simulation est présenté au chapitre I. Le comportement du robot en BO, lorsqu'on lui applique le découplage non linéaire total à une période de 2ms, pour une entrée constante, est illustré par simulation à la figure III.DT.1.

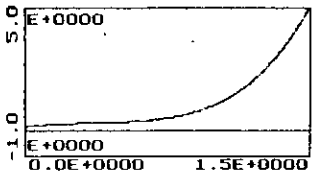
L'application par simulation de la commande mixte au modèle découplé, à une période de 10ms, montre son effet dont les résultats sont consignés aux figures III.DT.2.a, b et c. Par introduction d'un effet intégrateur, par le biais de la pondération $Q_2 = 0.07\,(1-q^{-1})$. Les résultats de simulation, dans un tel cas, sont illustrés aux figures III.3.a, b et c.



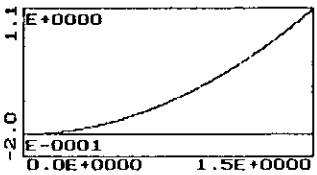
1.Commande U1.



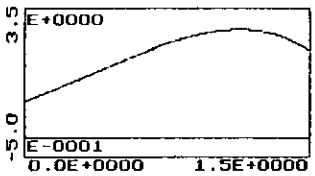
4.Reponse en position de l art1



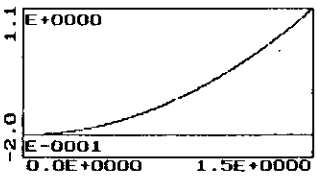
2.Commande U2.



5.Reponse en position de l art2

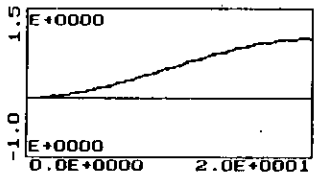


3.Commande U3.

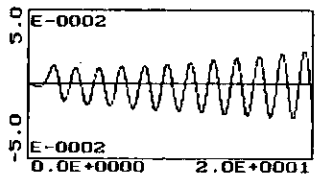


6.Reponse en position de l art3

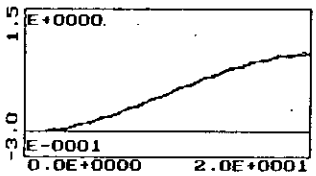
Figure III.DT.1 Reponses en position et commandes du decouplage total pour une entrée unitaire.



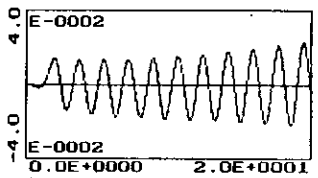
1.Consigne u1 et position q1 de l art1 pour GMV;Q1=0.07.



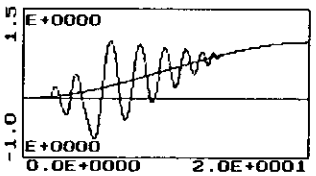
4.Erreur en position de l art1 u1-q1.



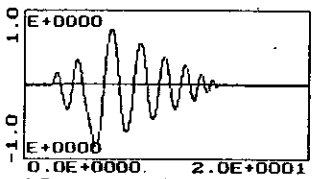
2.Consigne u2 et position q2 de l art2 pour PRPE;Q2=0.07;P2=1-0.1q-1.



5.Erreur en position de l art2 u2-q2.

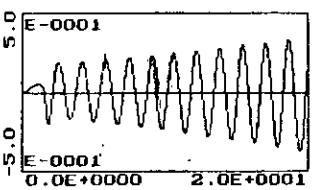


3.Consigne u3 et position q3 de l art3 pour PP;P3=1-0.5q-1+0.06q-2.

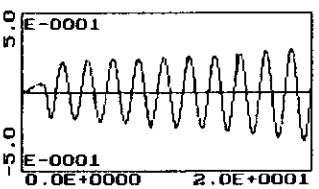


6.Erreur en position de l art3 u3-q3.

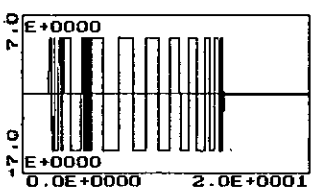
Figure III.DT.2.a Position et erreur en position pour la commande hybride à decouplage total.($A_{m1}/B_{m1}=B_{m2}/A_{m2}=0.095q-1/(1-0.9q-1)$).



1.Commande ud1.

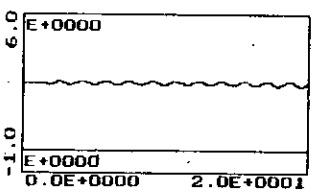


2.Commande ud2.

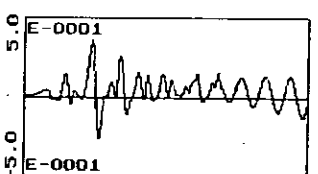


3.Commande ud3.

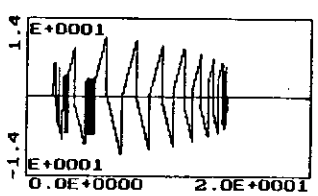
Figure III.DT.2.b Presentation des commandes STR.



1.Commande u1.

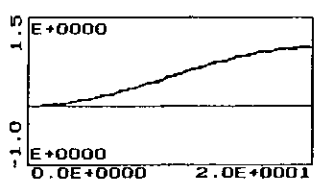


2.Commande u2.

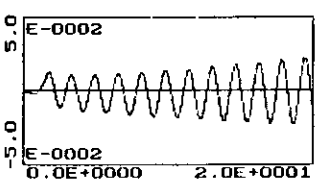


3.Commande u3.

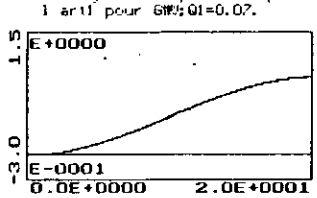
Figure III.DT.2.c Presentation des commandes du decouplage NL total.



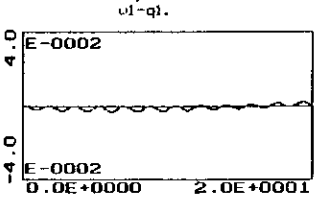
1.Consigne u1 et position q1 de l art1 pour GMV;Q1=0.07.



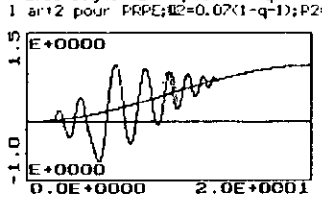
4.Erreur en position de l art1 u1-q1.



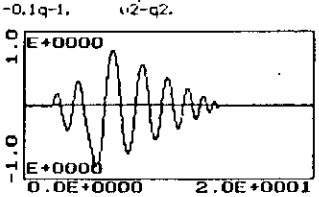
2.Consigne u2 et position q2 de l art2 pour PRPE;Q2=0.07(1-q-1);P2=1-0.1q-1.



5.Erreur en position de l art2 u2-q2.

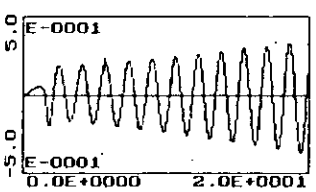


3.Consigne u3 et position q3 de l art3 pour PP;P3=1-0.5q-1+0.06q-2.

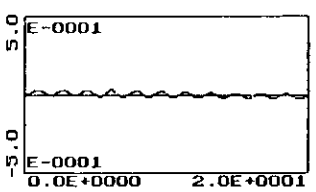


6.Erreur en position de l art3 u3-q3.

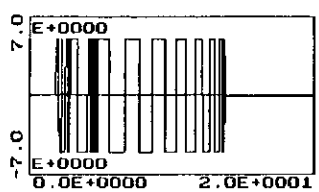
Figure III.DT3.a Position et erreur en position pour la commande hybride à decouplage total.($A_{m2}/B_{m2}=B_{m3}/A_{m3}=0.095q-1/(1-0.9q-1)$).



1.Commande ud1.



2.Commande ud2.



3.Commande ud3.

Figure III.DT.3.b Presentation des commandes STR.

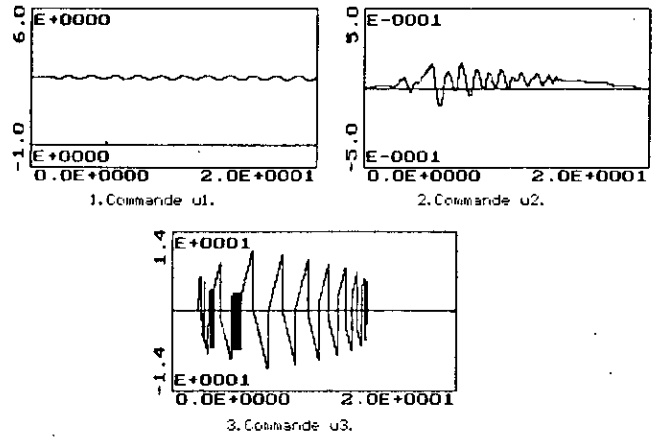


Figure III.DT.3.c Presentation des commandes du decouplage NL total.

III.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 [82]. Avec cette technique, un système non linéaire peut être linéarisé exactement (exact linéarization). 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 [83] 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 [84]. Sastry & Isidori [85] 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:

$$\begin{cases} \dot{x} = f(x) + g(x)\,u \\ y = h(x) \end{cases} \qquad \text{(III.18)}$$

Où $f^T(x) = [f_1(x) \ldots f_n(x)]$.
$g(x) = [g_1(x) \ldots g_m(x)]$ et $g_i^T = [g_{i_1}(x) \ldots g_{i_n}(x)]$ $i=1,m$.
$h^T(x) = [h_1(x) \ldots h_p(x)]$.

f_i, h_i, g_{i_j}: fonctions réelles de $R^n \rightarrow R$ infiniment différentiables.
$x^T = [x_1 \ldots x_n]$: vecteur d'état du système.
$u^T = [u_1 \ldots u_m]$: entrées de commande.
$y^T = [y_1 \ldots y_p]$: sorties du système.

D'une manière plus simplifiée on a:

$$\begin{cases} \dot{x}_i = f_i(x) + \sum_{k=1}^{m} g_{i_k}(x)\, u_k; \quad i=1,n. \\ y_i = h_i(x); \quad i=1,p. \end{cases} \tag{III.19}$$

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 [82][84].
On définit la dérivée directionnelle (dérivée de Lie) d'une fonction λ comme suit [82]:

$$L_f\lambda(x) = \sum_{i=1}^{n} \frac{\delta\lambda(x)}{\delta x_i} f_i \tag{III.20}$$

Où $\lambda(x)$: fonctions réelles de $R^n \to R$.
f: champ vecteur.

En utilisant l'équation (III.19) on a [84]:

$$y_i^{(d_i+1)} = L_f^{d_i+1} h_i(x) + L_g L_f^{d_i} h_i(x)\, u \tag{III.21}$$

Avec $y_i^k = L_f^k h_i(x)$; $k=1,d_i$ avec $L_f^0 h_i(x) = h_i(x)$.
et $L_g L_f^{d_i} h_i(x) = \left[L_{g_1} L_f^{d_i} h_i \ldots L_{g_m} L_f^{d_i} h_i \right]$.
où $d_i+1 = \min \left\{ k: L_g L_f^{k-1} h_i(x) \neq 0 \right\}$: appelé degré relatif.

Si le système est controllable on a alors [85][82]: $d = \sum_{i=1}^{m} (d_i+1) \leq n$.
En faisant le changement de variable suivant [86]:

$$z = \phi(x) \tag{III.22}$$

Où ϕ: matrice de transformation définis par:

$$\begin{cases} z_1 = y_1 = \phi_1(x) = h_1(x). \\ z_2 = y_1^{(1)} = \phi_2(x) = L_f h_1(x). \\ \cdots \\ z_{d_1+1} = y_1^{(d_1)} = \phi_{d_1+1}(x) = L_f^{d_1} h_1(x). \\ \cdots \\ z_{d-(d_m+1)} = y_m = \phi_{d-d_m}(x) = h_m(x). \\ z_{d-d_m} = y_m^{(1)} = \phi_{d-d_m+1}(x) = L_f h_m(x). \\ \cdots \\ z_d = y_m^{(d_m)} = \phi_d(x) = L_f^{d_m} h_m(x). \end{cases} \tag{III.23}$$

Si $d<n$, il est toujours possible de choisir $\phi_{d+1}(x), \ldots, \phi_n(x)$ de telle manière à avoir [82]:

$$L_g \phi_i(x) = 0 \qquad \text{(III.24)}$$

Le choix de $\phi_i(x)$ *pour* $d+1 \le i \le n$ peut se faire de telle sorte que la matrice de transformation $\phi(x)$ soit non singulière. On obtient alors [86]:

$$\begin{cases} \dot{z}_1 = z_2 . \\ \dot{z}_2 = z_3 . \\ \quad . \quad . \\ \dot{z}_{d_1} = z_{d_1+1} . \\ \dot{z}_{d_1+1} = b_1(z) + \sum_{i=1}^{m} a_{1i}(z)\, u_i . \end{cases}$$

$$\begin{cases} \quad . \quad . \\ \dot{z}_{d-(d_m+1)} = z_{d-d_m} . \\ \dot{z}_{d-d_m} = z_{d-d_m+1} . \\ \quad . \quad . \\ \dot{z}_d = b_m(z) + \sum_{i=1}^{m} a_{mi}(z)\, u_i . \end{cases} \qquad (III.25)$$

et

$$\begin{cases} \dot{z}_{d+1} = \phi_{d+1}(x) . \\ \quad . \quad . \\ \dot{z}_n = \phi_n(x) . \end{cases} \qquad (III.25.1)$$

avec $\begin{cases} y_1 = z_1 . \\ \quad . \quad . \\ y_m = z_{d-(d_m+1)} . \end{cases}$

et $b_i(z) = L_f^{d_i+1} h_i[\phi^{-1}(z)]; \quad i=1,m.$

$a_{ij}(z) = L_{g_i} L_f^{d_i} h_i[\phi^{-1}(z)]; \quad i,j=1,m.$

Le système défini par (III.25.1) caractérise la dynamique des zéros, car il est indépendant de u [82].

Après avoir mis le système sous la forme de Brunowski définie par (III.25)[86]. On cherche une commande μ telle que le système soit découplé. L'écriture matricielle de la relation (III.21) est [83][84]:

$$\begin{bmatrix} y_1^{(d_1+1)} \\ \cdot \\ \cdot \\ \cdot \\ y_m^{(d_m+1)} \end{bmatrix} = \begin{bmatrix} L_f^{d_1+1} h_1(x) \\ \cdot \\ \cdot \\ \cdot \\ L_f^{d_m+1} h_m(x) \end{bmatrix} + \begin{bmatrix} L_g L_f^{d_1} h_1(x) \\ \cdot \\ \cdot \\ \cdot \\ L_g L_f^{d_m} h_m(x) \end{bmatrix} u \qquad \text{(III.26)}$$

$$= \psi_1(x) + \psi_2(x)\, u.$$

Le problème est alors de calculer une commande μ qui découple le système, tel que:

$$u = \alpha(x) + \beta(x)\,\mu \qquad \text{(III.27)}$$

Où $\alpha^T(x) = [\alpha_1(x) \ldots \alpha_m(x)]$

$\beta_i^T(x) = [\beta_{i_1}(x) \ldots \beta_{i_m}(x)]$; $i = 1, m.$

$\beta^T(x) = [\beta_1(x) \ldots \beta_m(x)]$.

Si le système est découplable on a:

$$\begin{bmatrix} y_1^{(d_1+1)} \\ \cdot \\ \cdot \\ \cdot \\ y_m^{(d_m+1)} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \cdot \\ \cdot \\ \cdot \\ \mu_m \end{bmatrix} \qquad \text{(III.28)}$$

En utilisant (III.26) et (III.28), si $\psi_2(x)$ est nonsingulière:

$$u = [\psi_2(x)]^{-1}(\mu - \psi_1(x)) \qquad \text{(III.29)}$$

donc:

$$\begin{aligned} \alpha(x) &= -[\psi_2(x)]^{-1}\psi_1(x) \\ \beta(x) &= [\psi_2(x)]^{-1} \end{aligned} \qquad \text{(III.30)}$$

Ainsi le système est découplable par ce retour nonlinéaire. Chaque sortie y_i se comporte vis à vis de l'entrée μ_i comme le montre la figure III.8.
Le retour nonlinéaire est représenté dans la figure III.9.

Le découplage nonlinéaire, en utilisant la géométrie différentielle, a permi l'obtention d'un système linéaire totalement découplé. Par opposition à la linéarisation locale par approximation, cette méthode donne une linéarisation exacte, indépendamment du point de fonctionnement. On peut alors utiliser des contrôleurs linéaires de n'importe quel type.

Après avoir présenter la méthode de découplage, on considère maintenant le

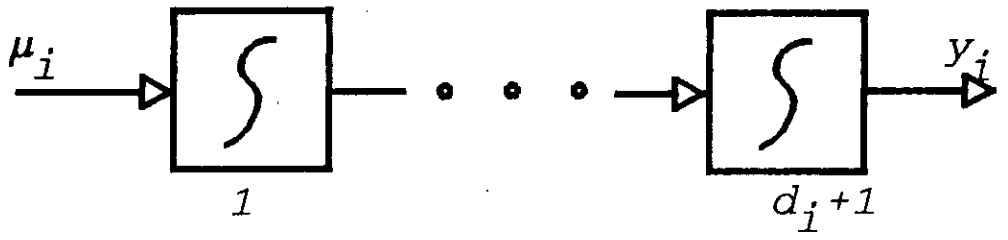


Figure III.8 Comportement de la sortie y_i envers μ_i.

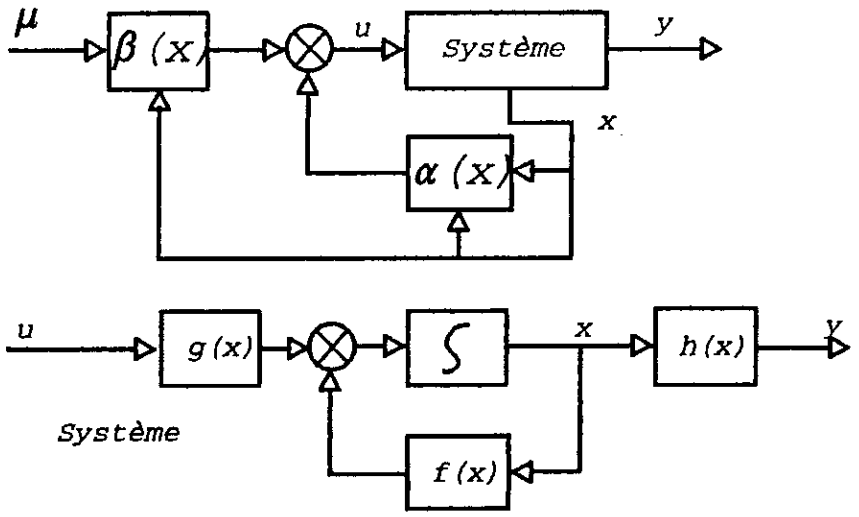


Figure III.9 Structure du découplage et linéarisation.

problème du découplage du robot de classe quatre. Le modèle dynamique du robot mis sous forme d'état est :

$$\begin{cases} \dot{x} = f(x) + g(x)\,u \\ y = h(x) \end{cases} \qquad \text{(III.31)}$$

Avec $f^T(x) = [f_1(x) \ldots f_n(x)]$.

où $f_1(x) = x_2$; $f_2(x) = -\dfrac{f_1}{m_1+m_3}x_2 - g$; $f_3(x) = x_4$;

$$f_4(x) = -\frac{f_2}{m_3 J^*}x_4 - 2\left(x_5 - \frac{l_2}{2}\right)\frac{x_4 x_6}{j^*}; \quad f_5(x) = x_5; \quad f_6(x) = -\frac{f_3}{m_3}x_6 - \left(x_5 - \frac{l_2}{2}\right)x_4^2.$$

et $x^T = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6]$:

$h_1(x) = x_1$; $h_2(x) = x_3$; $h_3(x) = x_5$.

$g^T(x) = [g_1\ g_2\ g_3]$;

avec $g_1^T(x)=\left[0 \quad \frac{k_1}{m_1+m_3} \quad 0 \; 0 \; 0 \; 0\right]$; $g_2^T(x)=\left[0 \; 0 \; 0 \quad \frac{k_2}{m_3 J^*} \quad 0 \; 0\right]$;

$g_3^T(x)=\left[0 \; 0 \; 0 \; 0 \; 0 \quad \frac{k_3}{m_3}\right]$;

et $u^T=[u_1 \; u_2 \; u_3]$; $y^T=[y_1 \; y_2 \; y_3]$.

On procéde d'abord par le calcul des degrés relatifs d_i.

Pour i=1 on a:

$$L_f h_1(x) = \sum_{i=1}^{6} \frac{\delta h_1(x)}{\delta x_i} f_i = f_1 = x_2; \qquad (III.32.1)$$

$$L_g h_1(x) = \left[L_{g_1} h_1(x) \; L_{g_2} h_1(x) \; L_{g_3} h_1(x)\right]$$

$$= \left[\sum_{i=1}^{6} \frac{\delta h_1(x)}{\delta x_i} g_{1i} \quad \sum_{i=1}^{6} \frac{\delta h_1(x)}{\delta x_i} g_{2i} \quad \sum_{i=1}^{6} \frac{\delta h_1(x)}{\delta x_i} g_{3i}\right] = [0 \; 0 \; 0] \qquad (III.32.2)$$

$$L_f^2 h_1(x) = \sum_{i=1}^{6} \frac{\delta L_f h_1(x)}{\delta x_i} f_i = f_2 = -\frac{f_1}{m_1+m_3} x_2 - g; \qquad (III.32.3)$$

$$L_g L_f h_1(x) = \left[\sum_{i=1}^{6} \frac{\delta L_f h_1(x)}{\delta x_i} g_{1i} \quad \sum_{i=1}^{6} \frac{\delta L_f h_1(x)}{\delta x_i} g_{2i} \quad \sum_{i=1}^{6} \frac{\delta L_f h_1(x)}{\delta x_i} g_{3i}\right] = [g_{12} \; g_{22} \; g_{32}]$$

$$= \left[\frac{k_1}{m_1+m_3} \quad 0 \quad 0\right] \qquad (III.32.4)$$

Ce dernier terme défini par (III.32.4) n'est pas nul donc $d_1=1$. On procédé de la même manière pour les autres entrées sorties. on a:

Pour i=2 on a:

$L_f h_2(x) = x_4$; (III.32.5)

$L_g h_2(x) = [0 \; 0 \; 0]$ (III.32.6)

$$L_f^2 h_1(x) = -\frac{f_2}{m_2 J^*} x_5 - 2\left(x_5 - \frac{l_2}{2}\right)\frac{x_4 x_6}{J^*}; \qquad (III.32.8)$$

$$L_g L_f h_1(x) = \left[0 \quad \frac{k_2}{m_3 J^*} \quad 0\right] \neq 0 \qquad (III.32.8)$$

D'où $d_2=1$.

Pour i=3 on a:

$L_f h_3(x) = x_6$; (III.32.9)

$L_g h_3(x) = [0 \; 0 \; 0]$ (III.32.10)

$$L_f^2 h_3(x) = -\frac{f_3}{m_3} x_6 - \left(x_5 - \frac{l_2}{2}\right) x_4^2 \qquad (III.32.11)$$

$$L_g L_f h_1(x) = \left[0 \quad 0 \quad \frac{k_3}{m_3}\right] \neq 0$$

D'où $d_3=1$.

D'où $d_3=1$.
Donc de (III.26) et en utilisant (III.32) on a:

$$\psi_1(x) = \begin{bmatrix} -\frac{f_1}{m_1+m_3}x_2 - g \\ -\frac{f_2}{m_2 J^*}x_4 - 2\left(x_5 - \frac{l_2}{2}\right)\frac{x_4 x_6}{J^*} \\ -\frac{f_3}{m_3}x_6 + \left(x_5 - \frac{l_2}{2}\right)x_4^2 \end{bmatrix} \qquad \text{(III.33)}$$

et

$$\psi_2(x) = \begin{bmatrix} \frac{k_1}{m_1+m_3} & 0 & 0 \\ 0 & \frac{k_2}{m_3 J^*} & 0 \\ 0 & 0 & \frac{k_3}{m_3} \end{bmatrix} \qquad \text{(III.34)}$$

La matrice $\psi_2(x)$ est inversible, d'où:

$$\psi_2^{-1}(x) = \begin{bmatrix} \frac{m_1+m_3}{k_1} & 0 & 0 \\ 0 & \frac{m_3 J^*}{k_2} & 0 \\ 0 & 0 & \frac{m_3}{k_3} \end{bmatrix} \qquad \text{(III.35)}$$

Donc de (III.33) et (III.35), le calcul de $\alpha(x)$ *et* $\beta(x)$, à partir de (III.30) est facile. On aboutit à la loi de commande en utilisant (III.29):

$$\begin{cases} u_1 = \frac{1}{k_1}\left[(m_1+m_3)(\mu_1+g) + f_1 x_2\right] \\ u_2 = \frac{m_3}{k_2}\left[J^*\mu_2 + f_2\frac{x_4}{m_2} + 2\left(x_5 - \frac{l_2}{2}\right)x_4 x_6\right] \\ u_3 = \frac{m_3}{k_3}\left[\mu_3 - \left(x_5 - \frac{l_2}{2}\right)x_4^2 + \frac{l_3}{m_3}x_6\right] \end{cases} \qquad \text{(III.36)}$$

Avec ce retour nonlinéaire, 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.
La structure d'une telle commande est définie par la figure III.10.

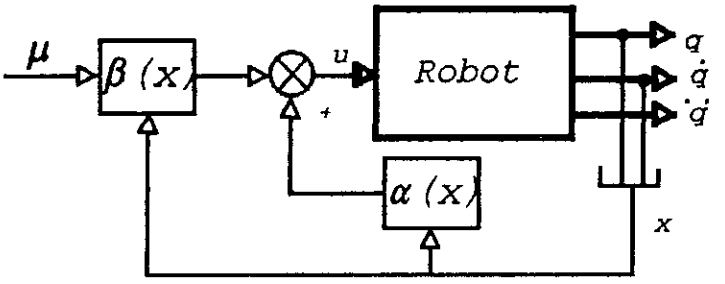


Figure III.10 Découplage et linéarisation exacte du robot. (utilisant la géométrie différentielle).

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 III.11.

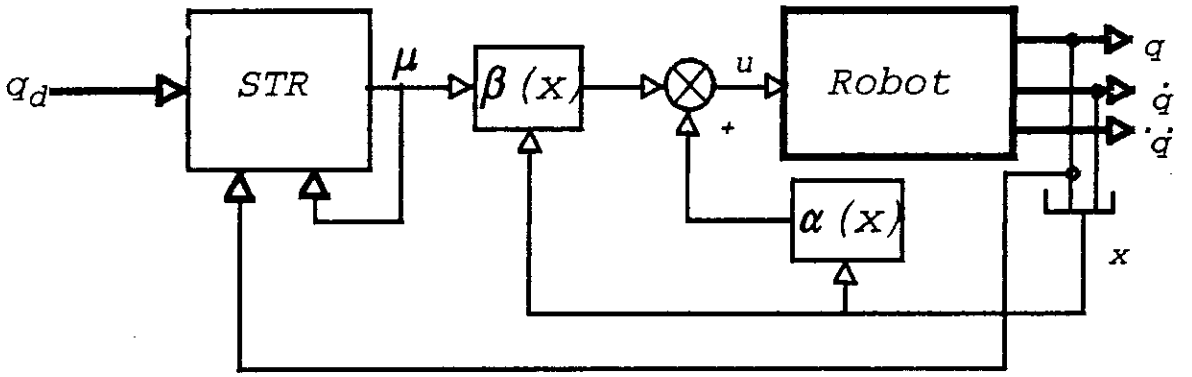


Figure III.11 Commande autoajustable hybride par découplage et linéarisation. (utilisant la géométrie différentielle).

III.4.a Résultats de simulation.

Le robot est le même que celui utilisé dans les parties précédentes. Par application du découplage non linéaire(NL) utilisant la géométrie différentielle, le comportement du robot en BO, à une entrée unitaire est présenté à la figureIII.ISO.1 (ISO: ISODORI). Le retour NL est appliqué à une période de 2ms.

Le robot ainsi découplé,lui est applique la commande à PP (Placement de Pôles), dont les résultats de simulation sont montrés aux figures III.ISO.2.a, b et c. Pour améliorer le comportement en BF, la commande à PRPE est appliqué au robot. Les figures III.ISO.3.a, b et c montrent les résultats de simulation dans un tel cas.

Dans le cas où le retour NL est calculé à partir d'un modèle erroné de 500%, le comportement du robot en BO, pour une entrée unitaire est montré à la figure III.ISO.4. Le modèle ainsi obtenu, on lui applique la commande par PRPE, dont les résultats de simulation sont illustrés aux figures III.ISI.5.a, b et c.

Pour tester la robustesse de la commande hybride à découplage NL utilisant la géométrie différentielle, on introduit une perturbation soudaine de 500% dans tous les paramètres du robot, à l'instant 5s. Les résultats de simulation dans un tel cas sont consignés aux figures III.ISO.6.a, b et c.

Enfin, pour tester la capacité de poursuite de l'algorithme, on applique à ce

dernier une consigne calculée à partir de la fenêtre de VIVIANI. La consigne de l'articulation 1 a été soumise à un lissage avec un polynôme du troisième degré, qui assure la continuité en position et en vitesse aux deux extrémités [criag].Les résultats de simulation dans une telle situation sont illustrés aux figures III.ISO.7.a, b et c ainsi que la projection de la fenêtre sur différents plans, est consignée sur la figure III.ISO.7.d.

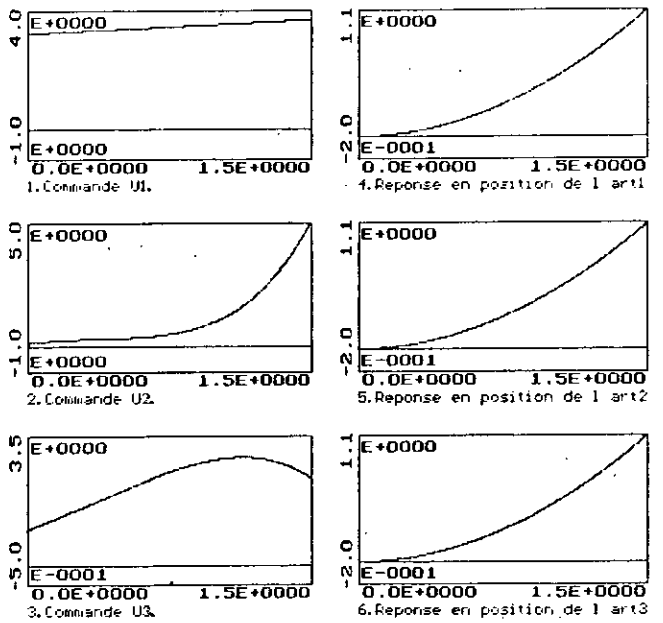


Figure III.ISO.1 Reponses en position et commandes du decouplage NL utilisant la géometrie différentielle pour une entrée unitaire.

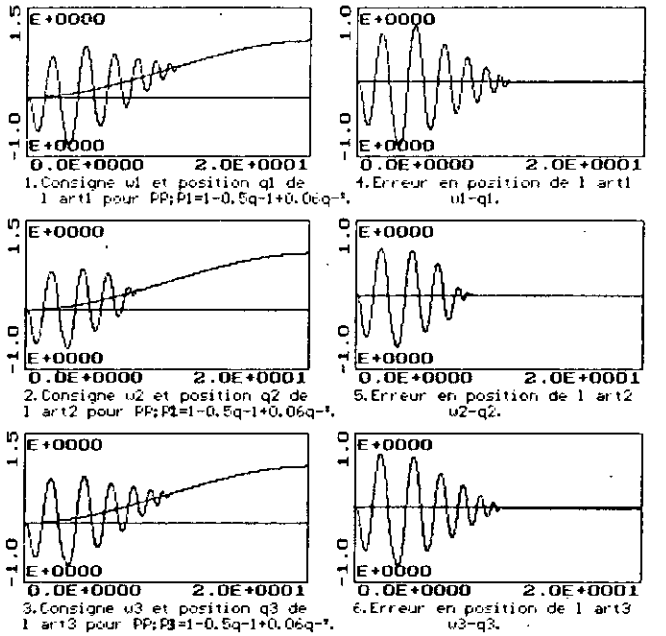


Figure III.ISO.2.a Position et erreur en position pour la commande hybride en utilisant la géometrie différentielle.(Ami/Bmi=0.095q-1/(1-0.9q-1)).

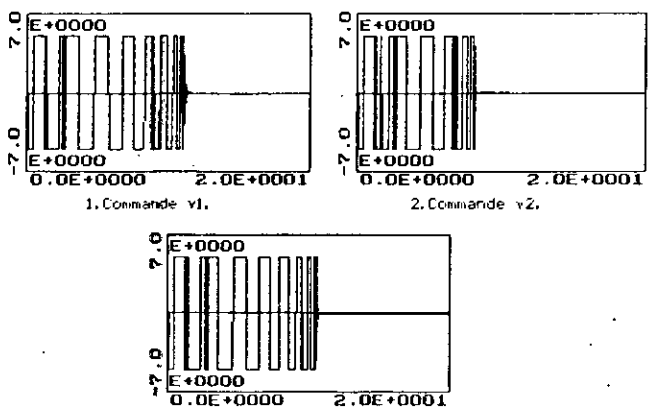


Figure III.ISO.2.b Presentation des commandes STR.

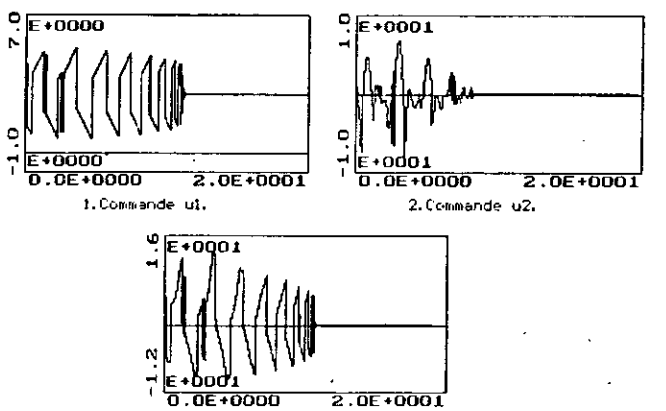
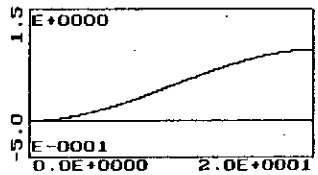
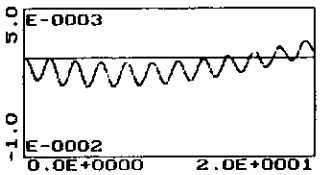


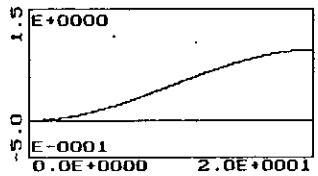
Figure III.ISO.2.c Presentation des commandes du decouplage NL total utilisant la géometrie différentielle.



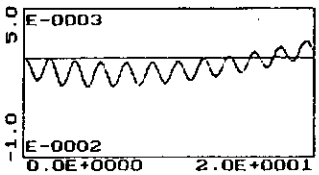
1.Consigne u1 et position q1 de l art1 pour PRPE.



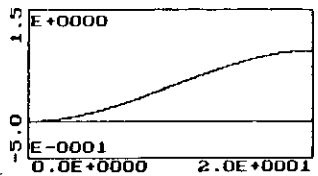
4.Erreur en position de l art1 u1-q1.



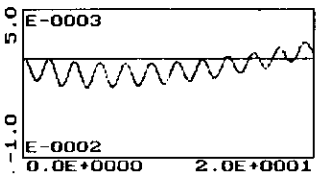
2.Consigne u2 et position q2 de l art2 pour PRPE.



5.Erreur en position de l art2 u2-q2.

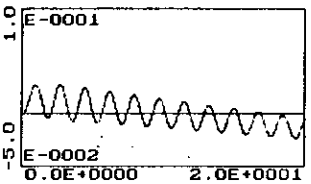


3.Consigne u3 et position q3 de l art3 pour PRPE.

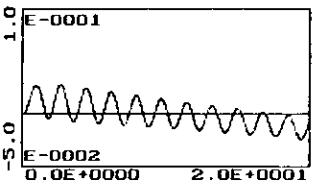


6.Erreur en position de l art3 u3-q3.

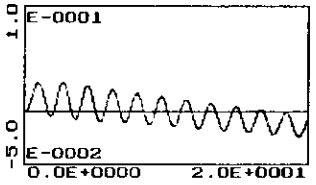
Figure III.ISO.3.a Position et erreur en position pour la commande hybride en utilisant la géometrie différentielle.
(Ami/Bmi=0.095q-1/(1-0.9q-1); Qi=0.07(1-q-1); Pi=1-0.1q-1).



1.Commande v1.

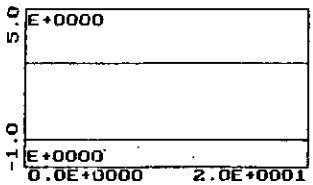


2.Commande v2.

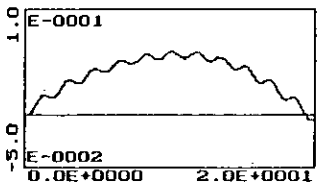


3.Commande v3.

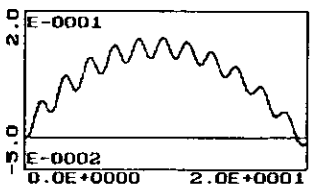
Figure III.ISO.3.b Presentation des commandes STR.



1.Commande u1.

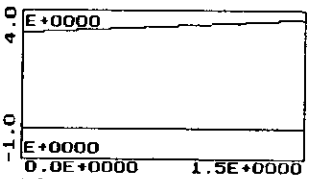


2.Commande u2.

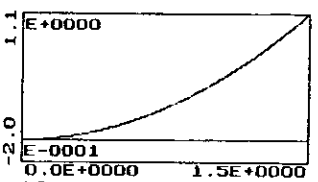


3.Commande u3.

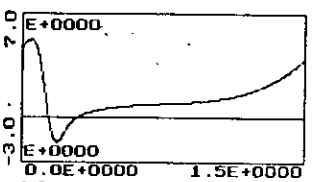
Figure III.ISO.3.c Presentation des commandes du decouplage NL total utilisant la géometrie différentielle.



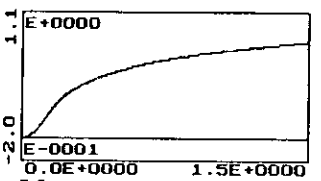
1.Commande U1



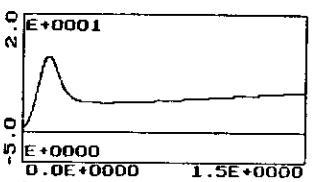
4.Reponse en position de l art1



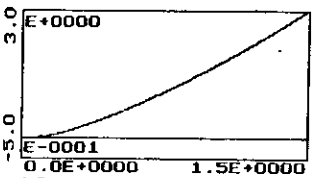
2.Commande U2



5.Reponse en position de l art2

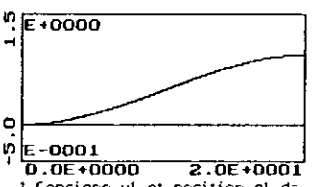


3.Commande U3

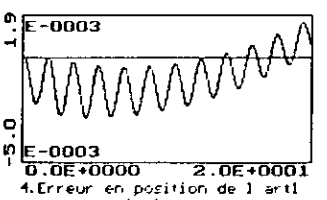


6.Reponse en position de l art3

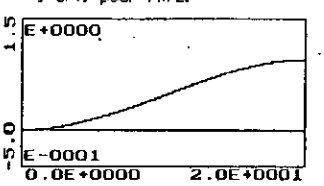
Figure III.ISO.4. Reponses en position et commandes NL du decouplage NL utilisant la géometrie différentielle pour une entrée v unitaire. (variation de 500% du modele du robot).



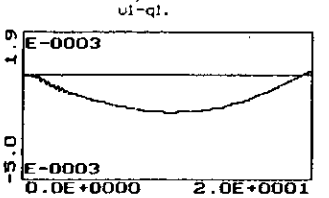
1.Consigne u1 et position q1 de l art1 pour PRPE.



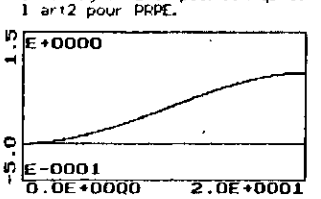
4.Erreur en position de l art1 u1-q1.



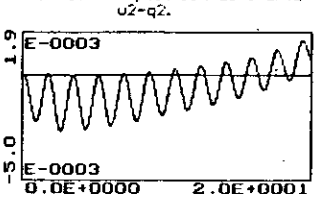
2.Consigne u2 et position q2 de l art2 pour PRPE.



5.Erreur en position de l art2 u2-q2.

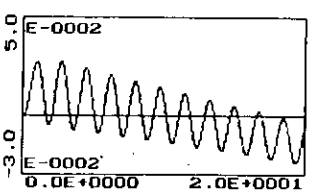


3.Consigne u3 et position q3 de l art3 pour PRPE.

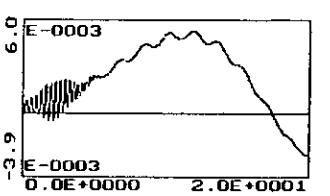


6.Erreur en position de l art3 u3-q3.

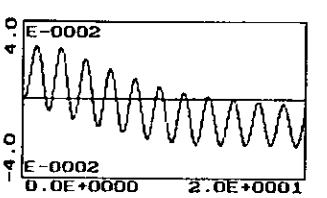
Figure III.ISO.5.a Position et erreur en position pour la commande hybride en utilisant la géometrie différentielle.(variation de 500% du retour NL)
(Ami/Bmi=0.095q-1/(1-0.9q-1); Qi=0.07(1-q-1); Pi=1-0.1q-1).



1.Commande v1.

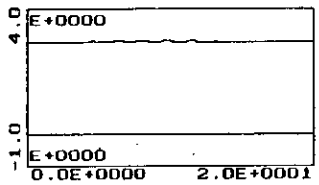


2.Commande v2.

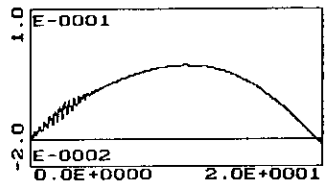


3.Commande v3.

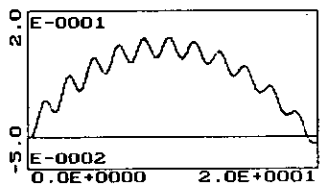
Figure III.ISO.5.b Presentation des commandes STR.



1.Commande u1.

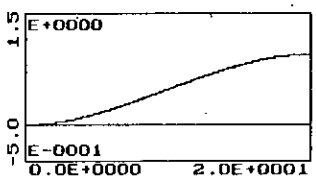


2.Commande u2.

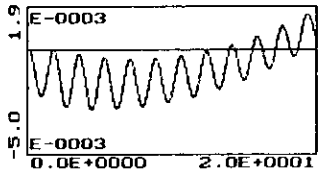


3.Commande u3.

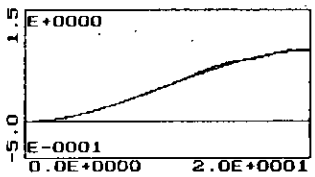
Figure III.ISO.5.c Presentation des commandes du decouplage NL utilisant la géometrie différentielle.



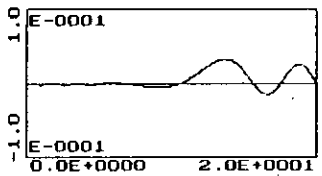
1.Consigne u1 et position q1 de l art1 pour PRPE.



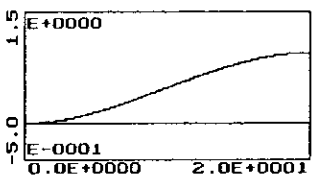
4.Erreur en position de l art1 u1-q1.



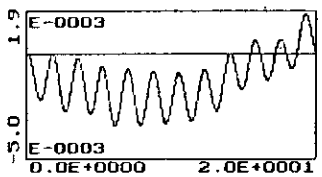
2.Consigne u2 et position q2 de l art2 pour PRPE.



5.Erreur en position de l art2 u2-q2.

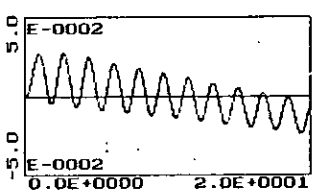


3.Consigne u3 et position q3 de l art3 pour PRPE.

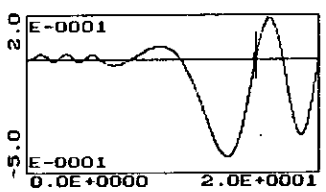


6.Erreur en position de l art3 u3-q3.

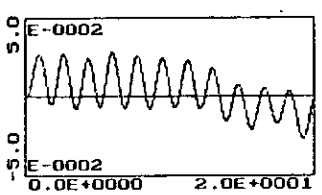
Figure III.ISO.6.a Position et erreur en position pour la commande hybride en utilisant la géometrie différentielle.(variation de 500% du modéle du robot à t=5s: Ami/Bmi=0.035q-1/(1-0.9q-1);Qi=0.07(1-q-1);Pi=1-0.1q-1).



1.Commande v1.

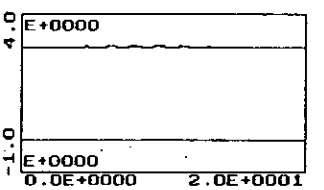


2.Commande v2.

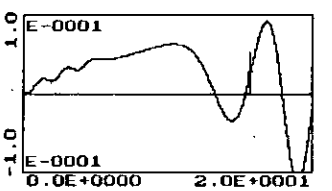


3.Commande v3.

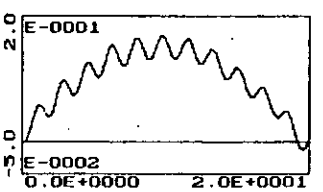
Figure III.ISO.6.b Presentation des commandes STR.



1.Commande u1.

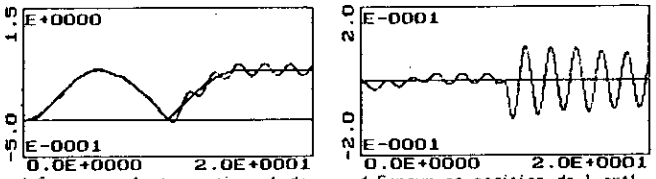


2.Commande u2.



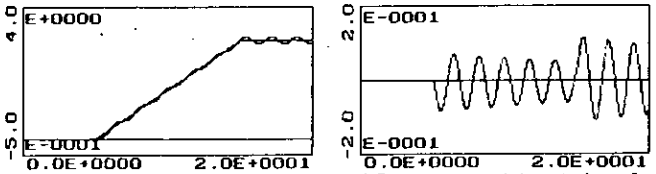
3.Commande u3.

Figure III.ISO.6.c Presentation des commandes du decouplage NL utilisant la géometrie différentielle.



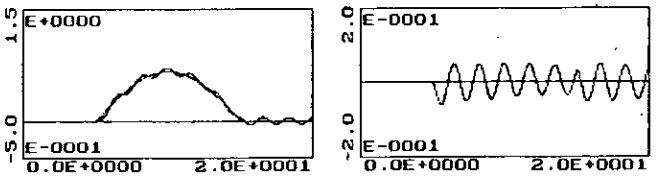
1.Consigne u1 et position q1 de l art1 pour PRPE.

4.Erreur en position de l art1 u1-q1.



2.Consigne u2 et position q2 de l art2 pour PRPE.

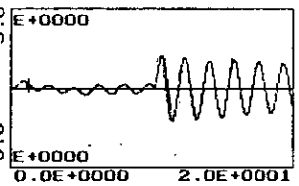
5.Erreur en position de l art2 u2-q2.



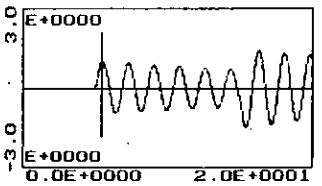
3.Consigne u3 et position q3 de l art3 pour PRPE.

6.Erreur en position de l art3 u3-q3.

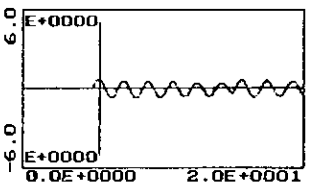
Figure III.ISO.7.a Position et erreur en position pour la commande hybride en utilisant la géometrie différentielle.(Consigne fenetre de VIVIANI) (Ami/Bmi=0.095q-1/(1-0.9q-1); Qi=0.02(1-q-1); Pi=1-0.1q-1).



1.Commande v1.

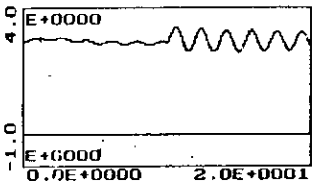


2.Commande v2.

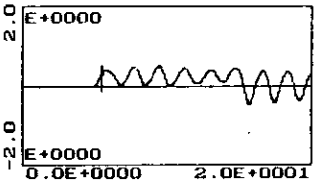


3.Commande v3.

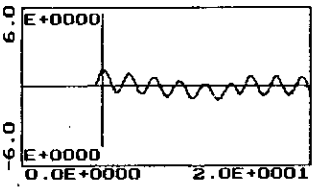
Figure III.ISO.7.b Presentation des commandes STR.



1.Commande u1.

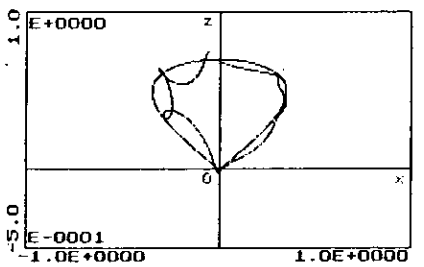


2.Commande u2.

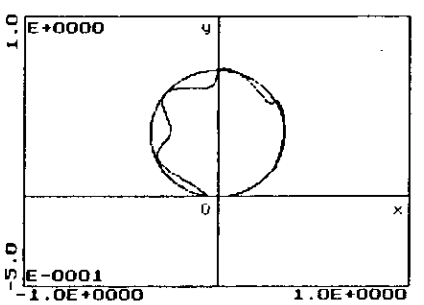


3.Commande u3.

Figure III.ISO.7.c Presentation des commandes du decouplage NL utilisant la géometrie différentielle.



1.Projection sur le plan xoz.



2.Projection sur le plan xoy.

Figure III.ISO.7.d Projection des trajectoires désirées et réelles sur les plan principaux.

Conclusion

Dans ce chapitre, on a présenté les différents algorithmes de commande hybride utilisant un retour non linéaire et une commande à STR mixte développée dans le chapitre II. On distingue la commande hybride par compensation de gravité, le découplage non linéaire partiel, le découplage non linéaire total et le découplage utilisant la géométrie différentielle ou linéarisation exacte.

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 au erreur 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. Le modèle obtenu, utilisant cet outil, est le même que celui utilisant le découplage non linéaire total. Par ailleurs le commande de ce modèle est la même. On utilise alors l'algorithme à STR mixte. L'algorithme à PRPE donne de très bonnes performances de poursuite, avec un effort de commande moins important. L'insensibilité de ce modèle, obtenu par commande hybride, à des erreurs de modélisation, montre sa supériorité par rapport aux autres algorithmes.
L'introduction des variation paramétriques n'affecte pas les performances de commande.
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'implémentation industrielle de telles commandes nécessite, des processeurs rapides, de sorte que le retour NL calculé, soit avec une approximation acceptable, identique à celui du modèle du robot, donc une période d'échantillonnage très faible.

Quand à la commande à STR, un processeur suffisamment rapide, peut assurer la tache. Donc l'utilisation de deux processeurs s'avère une idée intéressante. L'un programmé en langage machine de bas niveau, avec une optimisation des cycles machines, tandis que l'autre, avec seulement un langage évolué tel que le "C" ou le "Pascal".

Chapitre IV

COMMANDE ADAPTATIVE A PERTURBATION

"APC: ADAPTIVE PERTURBATION CONTROL"

"La pensée n'est qu'un éclaire au milieu d'une longue nuit, mais c'est cet éclaire qui est tout".

HENRI POINCARRE

Chapitre IV

COMMANDE ADAPTATIVE A PERTURBATION "APC:Adaptive Perturbation Control"

Introduction.

Dans les chapitres précédents, on s'est intéressé à la commande adaptative des robots manipulateurs en se basant sur un modèle aux différences entrées-sorties linéaire. En premier lieu, on a considéré le robot comme un modèle aux différences multivariable totalement découplé et variable dans le temps. L'utilisation de l'identification récursive à trace constante, permet l'obtention de ce modèle à chaque période d'échantillonnage. En second lieu, la compensation des nonlinéarités gravitationnelles et de structure, permet de mieux approcher le modèle dynamique du robot compensé, au modèle de représentation considéré. L'effet des couplages devient important lorsque le robot se déplace à des vitesses importantes. La considération de ses couplages s'avère nécessaire.
L'utilisation d'une commande complémentaire, basée sur la connaissance de la dynamique et de la structure du robot, permet de surmonter ce problème. C'est la commande adaptative à perturbation (adaptive perturbation control) [1].

Il existe plusieurs technique de commande non adaptative développées dans la littérature [1][2][7][21]. L'inconvénient majeur de ces techniques réside dans la connaissance exacte du modèle de connaissance du robot. En effet, l'obtention de l'inertie des différentes liaisons et le coefficient de frottement de chaque articulation, est une tache très difficile.
La commande adaptative à perturbation (APC) développée dans cette section est basée sur un modèle à perturbation obtenu par linéarisation du modèle nonlinéaire du robot autour d'une trajectoire nominale. Cette tajectoire est spécifiée par une interpolation de la position, de la vitesse et de l'accélération des articulations, entre différents points de l'espace opérationnel.
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 ce chapitre, on développe en premier lieu, le modèle à perturbation dans la section IV.1. En second lieu, les différents algorithmes auto-ajustables seront présentés. Enfin les résultats de simulation seront illustrés dans la dernière partie.

IV.1 Modèle à perturbation (perturbation model).

Le point essentiel dans une phase de modélisation d'un processus physique est la quantification du modèle par des équations mathématiques. Ceci nécessite une connaissance approfondie des lois physiques et des caractéristiques structurelles nominales du processus.
Considérant le modèle d'état suivant:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ x(t_0) = x_0 \end{cases} \qquad \text{(IV.1)}$$

Où $x^T(t) = [x_1(t) \ldots x_n(t)]$: vecteur d'état du système de dimension n.

$u^T(t)=[u_1(t)\ldots u_n(t)]$: vecteur de commande de dimension n.
x_0: vecteur initial de l'état du système à l'instant t_0.
$f^T=[f_1\ldots f_n]$: fonction vectorielle $R^{n+m}\to R^n$, continue et deux fois dérivable (pour des raison technique [44]).

Remarque.

Le modèle considéré dans ce cas est invariant dans le temps (c'est à dire $f(.,.)$ 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 :

$$y(t) = g(x(t)) \tag{IV.2}$$

Où $g^T=[g_1\ldots g_r]$: fonction vectorielle $R^n\to R^r$, 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.1) et (IV.2). 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:

$$\begin{cases} \dot{x}_0(t) = f(x_0(t),u_0(t)); \ x(t_0) = x_0. \\ y_0(t) = g(x_0(t)) \end{cases} \tag{IV.3}$$

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 le 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 d'optimisation non linéaire. Cela nécessite le choix d'une fonction coût scalaire (non obligatoirement quadratique) définie par:

$$I = \Phi(x(T)) + \int_{t_0}^{T} L(x(t),u(t))dt \tag{IV.4}$$

Où $x(T)$ est l'état terminal désiré.
$\Phi(.)$, $L(.,.)$: fonctions scalaires nonlinéaires.

le problème peut être formulé de la façon suivante:

"Ayant le système défini par (IV.1) 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.4)."

La résolution de ce problème, utilisant la théorie du calcul variationnel [86], ou des techniques numériques itératives, permettent la détermination de

$u_0(t)$ et $x_0(t)$.
Une fois $x_0(t)$ calculé, la réponse idéale $y_0(t)$ est trouvée en utilisant(IV.2).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.4), en utilisant le modèle de connaissance (IV.1) et (IV.2), 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 [44]:

Pour $u(t)=u_0(t)$; $t\in[t_0, T]$.
A-t-on:

$$\begin{cases} x(t) = x_0(t) \text{ pour tout } t\in[t_0, T] ?. \\ y(t) = y_0(t) \text{ pour tout } t\in[t_0, T] ?. \end{cases} \qquad (IV.5)$$

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 robotique, les phénomènes mal connus sont les frottements visqueux et secs. D'autre part, la valeur des paramètres utilisés dans le modèle mathématique est nominale, déterminée par expérimentation, telles que l'inertie, les masse et les dimensions des différentes liaisons.
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 $x(t_0)$. Une erreur de $x(t_0)-x_0(t_0)$ peut engendrer des erreurs importantes accumulées au cours du temps.

IV.1.1 Modèle à perturbation continu.

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 :
$$\delta x(t)=x(t)-x_0(t) \qquad (IV.6)$$
- Vecteur de sortie de perturbation :
$$\delta y(t)=y(t)-y_0(t) \qquad (IV.7)$$
- Vecteur de correction de commande :
$$\delta u(t)=u(t)-u_0(t) \qquad (IV.8)$$

On suppose que la commande réelle $u(t)$, l'état réel $x(t)$ et la sortie réelle sont reliés par le modèle défini par (IV.1) et (IV.2) (puisque on a pas d'autre modèle). On a alors :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = g(x(t)) \end{cases} \qquad (IV.9)$$

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.3). Le développement en série de Taylor de $f(.,.)$ et $g(.)$ autour de $x_0(t), u_0(t)$ donne [44]:

$$\begin{cases} f(x(t),u(t)) = f(x_0(t),u_0(t)) + \dfrac{\delta f}{\delta x}\Big|_0 \delta x(t) + \dfrac{\delta f}{\delta u}\Big|_0 \delta u(t) + \alpha_0(\delta x(t),\delta u(t)) \\ g(x(t)) = g(x_0(t)) + \dfrac{\delta g}{\delta x}\Big|_0 \delta x + \beta_0(\delta x(t)) \end{cases} \tag{IV.10}$$

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.10) peut se mettre sous la forme suivante (en utilisant (IV.3) et (IV.9)):

$$\begin{cases} \delta\dot{x}(t) = A_0(t)\delta x(t) + B_0(t)\delta u(t) + \alpha_0(\delta x(t),\delta u(t)) \\ \delta y(t) = C_0(t)\delta x(t) + \beta_0(\delta x(t)) \end{cases} \tag{IV.11}$$

Où $A_0(t) = \delta f/\delta x|_{x_0(t),u_0(t)}$ est une matrice variable dans le temps de dimension nxn, obtenue par calcul de la matrice Jacobienne $\delta f/\delta x$ au point variable $x_0(t)$ et $u_0(t)$.

$B_0(t) = \delta f/\delta u|_{x_0(t),u_0(t)}$ est une matrice variable dans le temps de dimension nxm, obtenue par calcul de la matrice Jacobienne $\delta f/\delta u$ au point variable $x_0(t)$ et $u_0(t)$.

$C_0(t) = \delta g/\delta x|_{x_0(t)}$ est une matrice variable dans le temps de dimension rxn, obtenue par calcul de la matrice Jacobienne $\delta g/\delta x$ au point variable $x_0(t)$.

Les équations définies par (IV.11), 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 (continious perturbation model) est obtenu en négligent les termes $\alpha_0(\delta x(t),\delta u(t))$ et $\beta_0(\delta x(t))$. On obtient ainsi [44]:

$$\begin{cases} \delta\dot{x}(t) = A_0(t)\delta x(t) + B_0(t)\delta u(t) \\ \delta y(t) = C_0(t)\delta x(t) \end{cases} \tag{IV.12}$$

Ce modèle est la forme d'état standard d'un système linéaire variable dans le temps (LTVS Linear Time Varying System).

IV.1.2 Modèle aux différences à perturbation.

La solution du système différentiel défini par (IV.12) est [23][31]:

$$\delta x(t) = \phi(t,t_0)\delta x(t_0) + \int_{t_0}^{t} \phi(t-\tau,t_0) B_0(\tau)\delta u(\tau)\, d\tau \tag{IV.13}$$

Où $\phi(t,t_0)$: matrice de transition d'état du système défini par (IV.12).

Le processus définissant le robot manipulateur est commandé par calculateur. Le modèle discret de (IV.13) est nécessaire. En remplaçant dans (IV.13) 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 [23][53]:

$$\delta x(k+1) = \overline{A}(k)\,\delta x(k) + \overline{B}(k)\,\delta u(k) \qquad \text{(IV.14)}$$

Où $k+1$ *et* k représentent respectivement $(k+1)T$ *et* kT.

et $\overline{A}(k) = \phi((k+1)T, kT)$; $\overline{B}(k) = \int_{kT}^{(k+1)T} \phi((k+1)T-\tau, kT)\,B_0(\tau)\,d\tau$.

L'équation de la sortie (IV.12) sous sa forme discrète est défini par:

$$\delta y(k) = \overline{C}(k)\,\delta x(k) \qquad \text{(IV.15)}$$

Où $\overline{C}(k) = C_0(kT)$.

Le modèle d'état discret, défini par (IV.14) et (IV.15), exprime une relation de récurrence entre les états du système. La combinaison de L'équation d'état (IV.14) et de L'équation de sortie (IV.15), permet d'obtenir la forme canonique du modèle au différence [50] [57] [66] [87]:

$$\begin{bmatrix} A_{11}(q^{-1}) & \cdots & A_{1r}(q^{-1}) \\ \vdots & \cdots & \vdots \\ A_{r1}(q^{-1}) & \cdots & A_{rr}(q^{-1}) \end{bmatrix} \delta y(k) = \begin{bmatrix} B_{11}(q^{-1}) & \cdots & B_{1m}(q^{-1}) \\ \vdots & \cdots & \vdots \\ B_{r1}(q^{-1}) & \cdots & B_{rm}(q^{-1}) \end{bmatrix} \delta u(k) + e(k)$$

(IV.16)

Où $A_{ij}(q^{-1}) = \delta_{ij} + \sum_{l=1}^{n_{ij}} a_{ij_l} q^{-l}$; $i,j=1,r$;

$$B_{ij}(q^{-1}) = q^{-d_{ij}} \sum_{l=1}^{m_{ij}} b_{ij_l} q^{-l};\ i=1,r\ ;j=1,m;\ \ d_{ij} \geq 0.$$

q^{-1} Opérateur retard; $\delta y(t-1) = q^{-1}\delta y(t)$.

δ_{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 δy_i *et* δu_j.

Le modèle ainsi obtenu est le **modèle aux différences à perturbation** (perturbation difference model).

IV.2 Formulation du problème de la commande "APC".

L'équation (IV.16) 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 le modèle (IV.16) nécessite beaucoup de calcul.

Le modèle (IV.16) 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 [23]:

$$A_i(q^{-1})\,\delta y_i(k) = q^{-d_i} B_i(q^{-1})\,\delta u_i(k) + h_i(k) \quad i=1,m \qquad \text{(IV.17)}$$

Où $A_i(q^{-1}) = 1+\sum_{l=1}^{n_i} a_{i_l} q^{-l}$; $B_i(q^{-1}) = \sum_{l=1}^{m_i} b_{i_l} q^{-l}$.

d_i: retard (entier) du système entre δy_i et δu_i.

$$h_i(k) = e_i(k) + \sum_{j \neq i} B_{ij}(q^{-1})\,\delta u_i(k) - \sum_{j \neq i} A_{ij}(q^{-1})\,\delta y_i(k); \quad i=1,m.$$

Le modèle (IV.17) est une approximation acceptable, du fait que le couplage est compensé par le calcul d'une commande anticipatrice. Tandis que dans le chapitre II, cette dernière est valable à des faibles vitesses.

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 réalisé par l'action de retour (feedback). La commande de correction $\delta u(k)$, calculée à partir du modèle à perturbation permet d'accomplir cette tache. L'identification du modèle défini par (IV.17) 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 [1][23]. 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.1) [22][23].

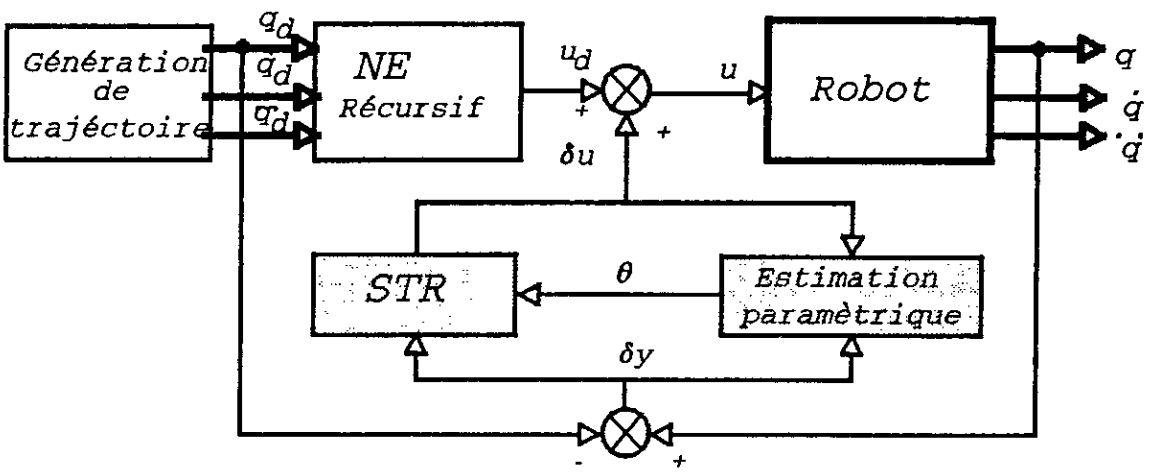


Figure IV.1 Structure de la commande APC.

Le vecteur de paramètres θ est celui du modèle à perturbation dans le cas où la commande est indirecte, celui du régulateur dans le cas où la commande est directe.

L'algorithme de commande indirecte est [23]:

Etape 1: Génération de trajectoire désirées $q_d, \dot{q}_d$, et $\ddot{q}_d$.
Etape 2: Calcul de la commande nominale $u_0(t)$ en utilisant l'algorithme récursif de Newton-Euler (voir annexe 2).

Etape 3: Identification des paramètres du modèle à perturbation.
Etape 4: Calcul des paramètres du régulateur.
Etape 5: Calcul de la commande $\delta u(k)$ qui réalise la régulation.
Etape 6: Calcul de la commande globale définie par $u(k)=u_0(k)+\delta u(k)$.
k=k+1 revenir à étape 1.

L'algorithme de commande directe est [22][23]:

Etape 1: Génération de trajectoire désirées $q_d, \dot{q}_d$, et $\ddot{q}_d$.
Etape 2: Calcul de la commande nominale $u_0(t)$ en utilisant l'algorithme récursif de Newton-Euler (voir annexe 2).
Etape 3: Identification des paramètres du régulateur en construisant un prédicteur à partir du modèle à perturbation (voir chapitre I).
Etape 4: Calcul de la commande $\delta u(k)$ qui réalise la régulation.
Etape 5: Calcul de la commande globale définie par $u(k)=u_0(k)+\delta u(k)$.
k=k+1 revenir à étape 1.

Remarque.
Pour réduire le temps de calcul de l'algorithme, la trajectoire nominale peut être calculée hors ligne (off line) et enregistrée dans des masses mémoires. □

IV.3 Différentes stratégies de commande.

On s'intéresse dans cette section à développer quelques algorithmes de commande à STR, exposés dans le chapitre II, 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.

Dans ce qui suit, on présente le développement de la commande APC dans le cas de la GMV. Tous les autres algorithmes peuvent être étendu de la même manière.

IV.3.1 Commande à variance minimale généralisée (GMV).

Le modèle du système est un modèle ARMAX défini par (II.1). Le critère de commande est défini par [23]:

$$J_i = E\left[(P_i\delta y_i(t+d+1))^2+(Q_i'\delta u_i(t))^2\right]; \quad i=1,3 \qquad \text{(IV.18)}$$

Où P_i et Q_i' définis par (II.35).

Le critère (IV.18) 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 δy_i vers zéro avec une dynamique $P_i(q^{-1})$.

La loi de commande qui minimise (IV.22) est :

$$\delta u_i(t) = \frac{-R_i\delta y_i(t)-P_{D_i}(1)S_i'(1)h_i}{P_{D_i}(S_i+Q_iC_i)} \qquad \text{(IV.19)}$$

Où P_{D_i}, Q_i et C_i: définis dans chapitre II.
R_i, S_i' et S_i: solution de L'équation Diophantine (II.5).

L'algorithme de commande adaptative indirecte est défini par:

Données: Spécifier $P_i(q^{-1})$, $Q_i(q^{-1})$, d_i, n_i, m_i et l_i; $i=1,3$.
Etape 1 et Etape 2 : comme précédemment.

Etape 3: Estimation de A_i, B_i, C_i et h_i en utilisant l'algorithme MCR à trace constante (Chap I.3.4) avec:

$$\theta_i^T = [a_{i_1} \ldots a_{i_{n_i}}\ b_{i_1} \ldots b_{i_{m_i}}\ c_{i_1} \ldots c_{i_{l_i}}\ h_i].$$

$$\phi_i^T(t) = [-\delta y_i(t-1) \ldots -\delta y_i(t-n_i)\ \delta u_i(t-d_i-1) \ldots \delta u_i(t-d_i-m_i)\ \varepsilon_i(t-1) \ldots \varepsilon_i(t-l_i)\ 1].$$

$$\varepsilon_i(t) = \delta y_i(t) - \theta_i^T \phi_i(t).$$

Etape 4: Calcul de S_i' et R_i par résolution de l'équation Diophantine (II.5) pour i=1,3.

Etape 5: Calcul de la commande $\delta u(k)$ qui réalise la régulation (équation IV.19).

Etape 6: Calcul de la commande globale définie par $u(k) = u_0(k) + \delta u(k)$.

k=k+1 revenir à étape 1.

L'algorithme de commande adaptative directe est défini par:

Données: Spécifier $P_i(q^{-1})$, $Q_i(q^{-1})$, d_i, n_i, m_i et l_i; $i=1,3$.

Etape 1 et Etape 2 : comme précédemment.

Etape 3: Estimation de C_i, R_i, S_i et δ_i en utilisant l'algorithme MCR à trace constante (Chap I.3.4) avec:

$$\theta_i^T = [c_{i_1} \ldots c_{i_{l_i}}\ r_{i_0} \ldots r_{i_{n_r-1}}\ s_{i_0} \ldots s_{n_s}\ \delta_i].$$

$$n_r = n_i - 1;\ n_s = m_i + d_i - 1.$$

$$\phi_i^T(t) = [-\phi_i^*(t-1) \ldots -\phi_i^*(t-l_i)\ \delta y_{i_f}(t-d_i-1) \ldots \delta y_f(t-d_i-n_r)\ \delta u_i(t-d_i-1) \ldots \delta u_i(t-d_i-n_s)\ 1].$$

avec $\delta y_{i_f} = \delta y_i / P_d$

$$\varepsilon_i(t) = P_i \delta y_i(t) - \theta_i^T \phi_i(t).$$ (voir chapitre II).

Etape 4: Calcul de la commande $\delta u(k)$ qui réalise la régulation (équation IV.19 modifiée):

$$\delta u_i(t) = \frac{-R_i \delta y_{if}(t) - \delta_i}{S_i + Q_i C_i}$$

Etape 5: Calcul de la commande globale définie par $u(k) = u_0(k) + \delta u(k)$.

k=k+1 revenir à étape 1.

L'algorithme explicite et implicite de la GMV, ainsi développé peut être utilisé pour la commande du robot manipulateur, avec un choix de la trajectoire nominale et de la structure du modèle à perturbation.

Remarque.

Les différentes stratégies de commande autre que GMV (voir chapitre II)peuvent être directement implémentées dans le bloc STR de la figure IV.1.

La combinaison de ces différentes stratégies, pour la commande du robot de classe quatre, à des périodes d'échantillonnage différentes aboutit à l'algorithme de commande mixte et "multirate" développé dans le chapitre II.□

IV.4 Résultats de simulation.

Le robot utilisé dans ce chapitre est celui développé au chapitre I. Les résultats de simulation du couplage de l'algorithme de NE et de Euler-Lagrange (voir figure I.1), pour une consigne polynomiale, sont illustrées aux figures IV.2.a, b, c et d. La commande désirée calculée à partir de l'algorithme de NE est appliqué chaque 2ms (les conditions initiales sont nulles).

Dans le cas où on applique la commande APC (voir figure IV.1), l'algorithme à STR appliqué est mixte, à une période de 10ms. Les résultats de simulation, dans de telles situations, sont illustrés aux figures IV.3.a, b et c.

Les figures IV.4.a et b montrent les résultats de simulation de l'algorithme de NE couplé à EL, dans le cas où le modèle de NE est erroné, par rapport au modèle du robot, de 500%. L'application de l'architecture APC, utilisant l'algorithme à STR mixte, est illustrée par simulation aux figures IV.5.a, b et c.

Pour tester la robustesse de la commande APC à des variations paramétriques, on introduit une perturbation, de tous les paramètres du robot, à l'instant 5s, d'une valeur de 500%. Les résultats de simulation, dans une telle situation, sont consignés aux figures IV.6.a, b et c.

Dans le cas de l'architecture de la figure I.11, les résultats de simulation, pour la consigne de fenêtre de VIVIANI, sont illustrés aux figures IV.7.a, b et c. Les conditions initiales du robot lors de la simulation sont nulles sauf $x_1(t=0)=0.7m$.
Pour compenser les erreurs de poursuite la commande APC est introduite, dont les résultats de simulation sont illustrés aux figures IV.8.a, b, c et d.

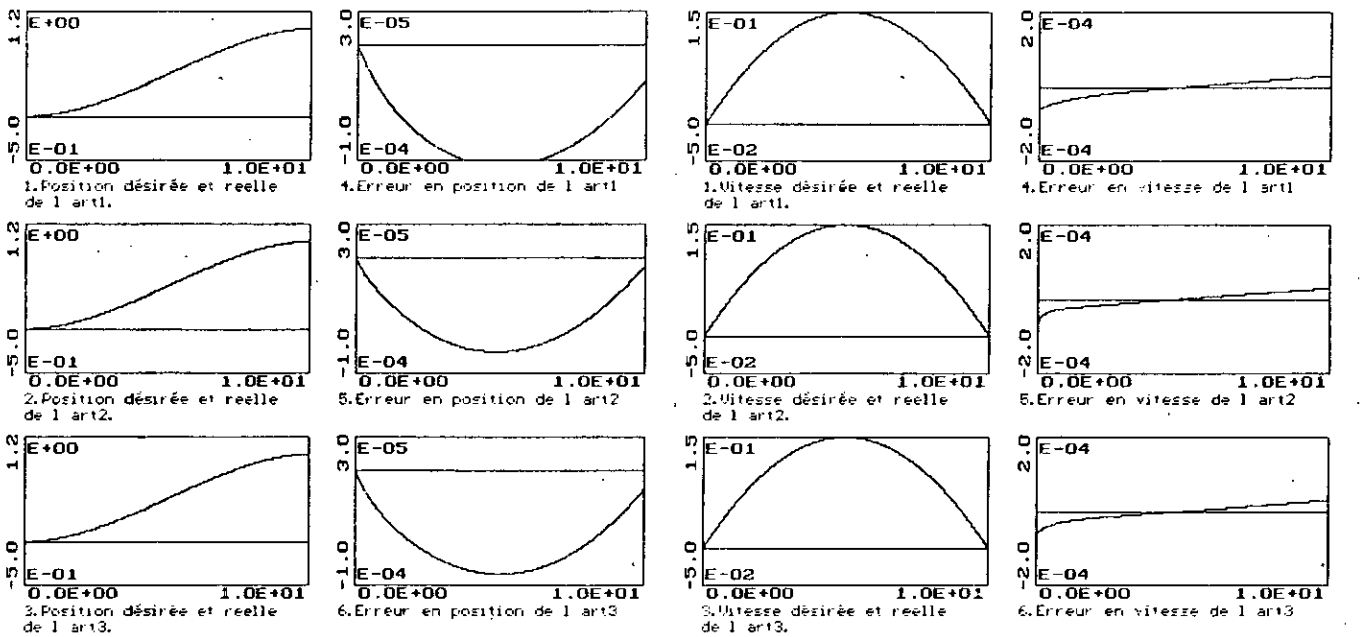


Figure IV.2.a Positions et erreurs en position dans le cas du couplage de Newton-Euler et d Euler-Lagrange. (feedforward)

Figure IV.2.b Vitesses et erreurs en vitesse dans le cas du couplage de Newton-Euler et d Euler-Lagrange. (feedforward)

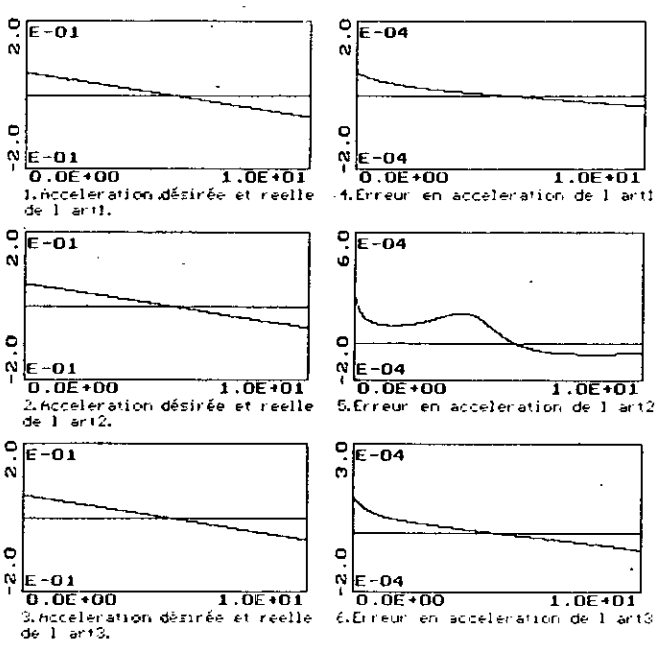


Figure IV.2.c Accelerations et erreurs en acceleration dans le cas du couplage de Newton-Euler et d Euler-Lagrange. (feedforward)

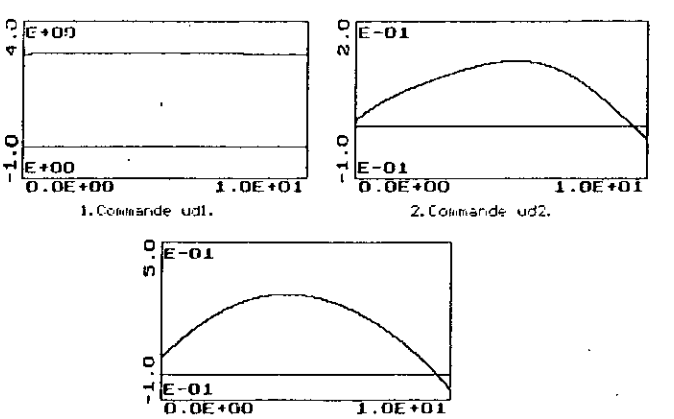
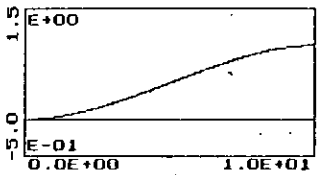
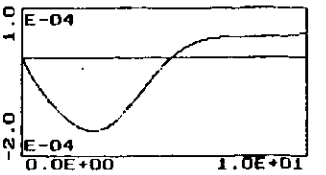


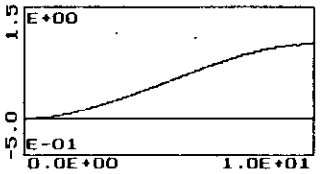
Figure IV.2.d Presentation des commandes.



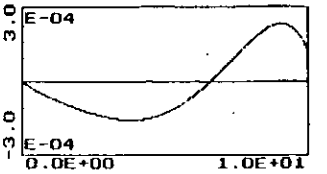
1.Position désirée et reelle de l art1. GMV Q1=0.007.



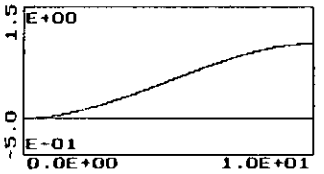
4.Erreur en position de l art1



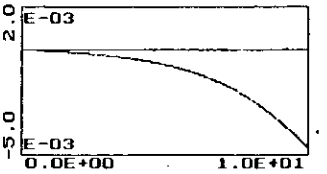
2.Position désirée et reelle de l art2. PRPE Q2=0.05;P2=1.



5.Erreur en position de l art2

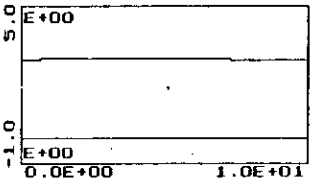


3.Position désirée et reelle de l art3. PRPE Q3=0.005;P3=1.

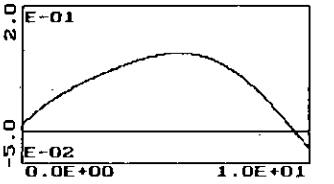


6.Erreur en position de l art3

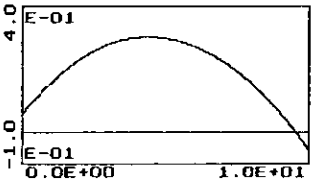
Figure IV.3.a Positions et erreurs en position pour APC.
(feedforward+feedback_STR)



1.Commande u1.

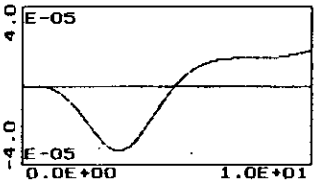


2.Commande u2.

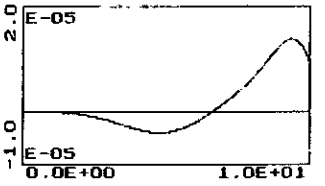


3.Commande u3.

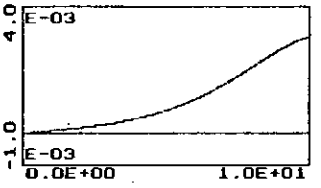
Figure IV.3.b Presentation des commandes.



1.Commande δu1.

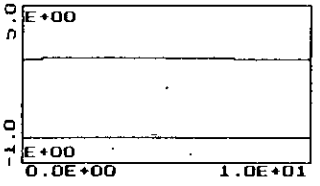


2.Commande δu2.

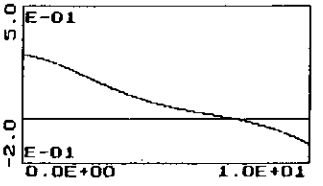


3.Commande δu3.

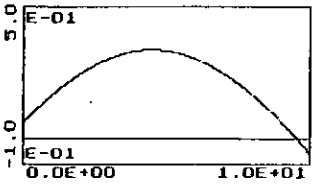
Figure IV.3.c Presentation des commandes δu.



1.Commande ud1.

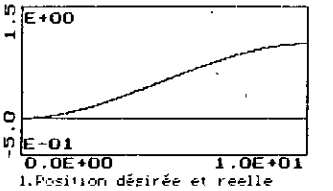


2.Commande ud2.

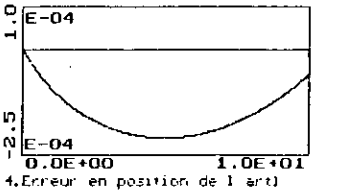


3.Commande ud3.

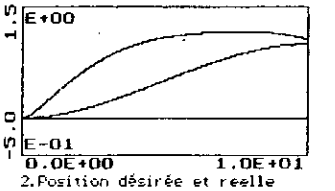
Figure IV.4.b Presentation des commandes.



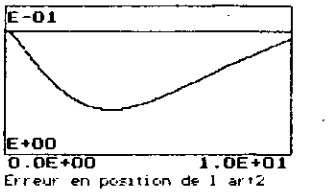
1.Position désirée et reelle de l art1.



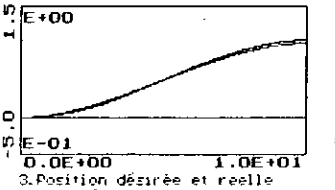
4.Erreur en position de l art1



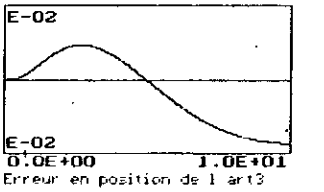
2.Position désirée et reelle de l art2.



5.Erreur en position de l art2

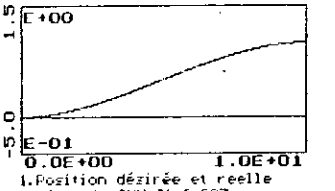


3.Position désirée et reelle de l art3.

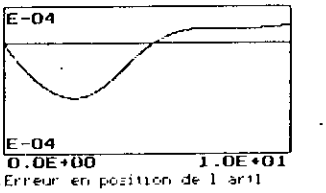


6.Erreur en position de l art3

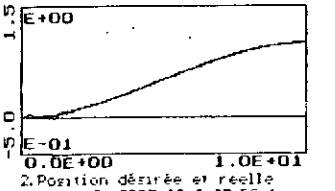
Figure IV.4.a Positions et erreurs en position dans le cas du couplage de Newton-Euler et d Euler-Lagrange.(feedforward)
(variation du modèle de NE de 500% du modèle réel)



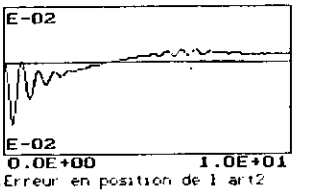
1.Position désirée et reelle de l art1. GMV Q1=0.007.



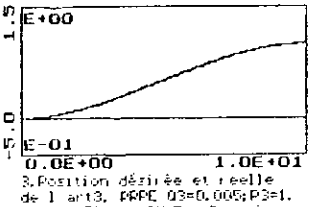
4.Erreur en position de l art1



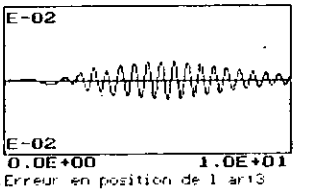
2.Position désirée et reelle de l art2. PRPE Q2=0.05;P2=1.



5.Erreur en position de l art2

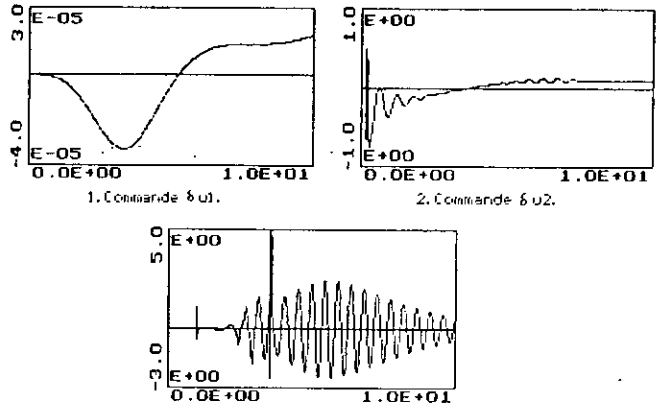


3.Position désirée et reelle de l art3. PRPE Q3=0.005;P3=1.



6.Erreur en position de l art3

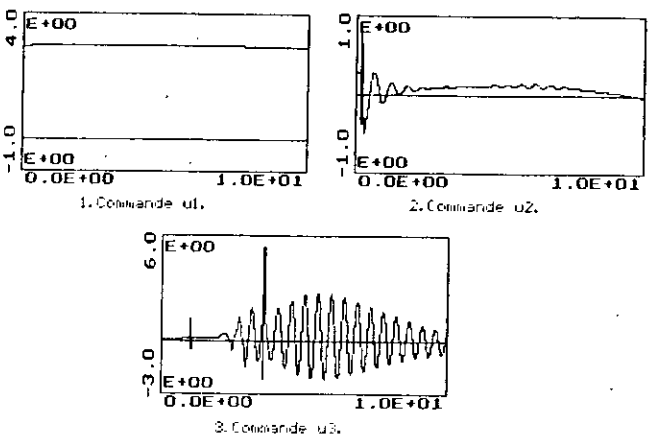
Figure IV.5.a Positions et erreurs en position pour APC.
(feedforward+feedbact_STR; variation du modèle de NE de 500%)



1. Commande δu1. 2. Commande δu2.

3. Commande δu3.

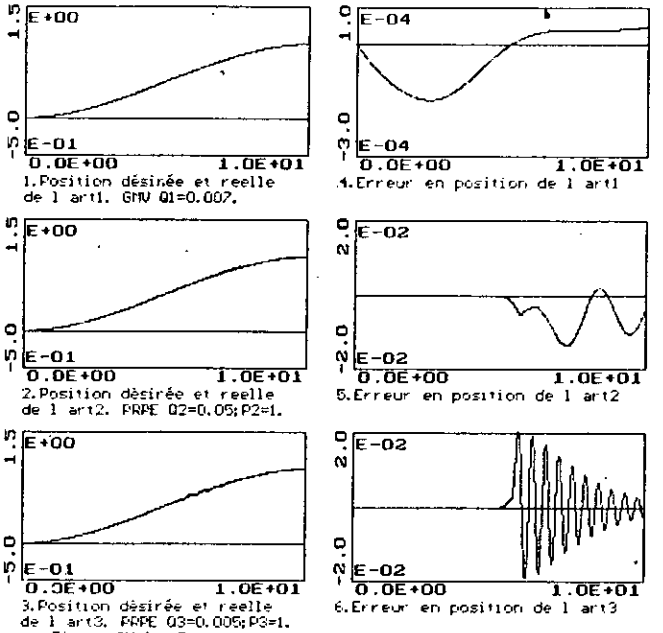
Figure IV.5.b Presentation des commandes δu.



1. Commande u1. 2. Commande u2.

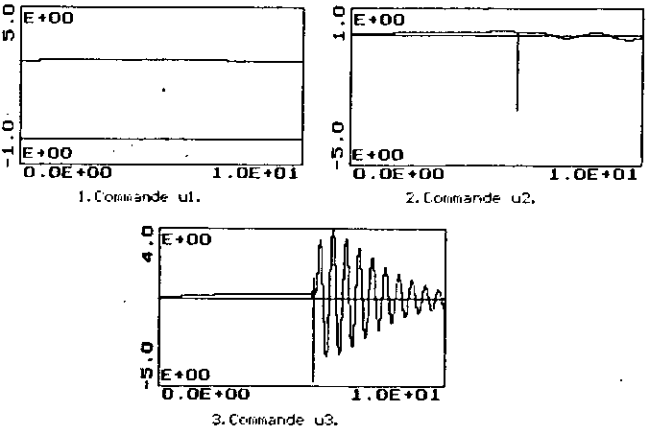
3. Commande u3.

Figure IV.5.c Presentation des commandes.



1. Position désirée et reelle de l art1. GMV Q1=0.007.
4. Erreur en position de l art1
2. Position désirée et reelle de l art2. PRPE Q2=0.05; P2=1.
5. Erreur en position de l art2
3. Position désirée et reelle de l art3. PRPE Q3=0.005; P3=1.
6. Erreur en position de l art3

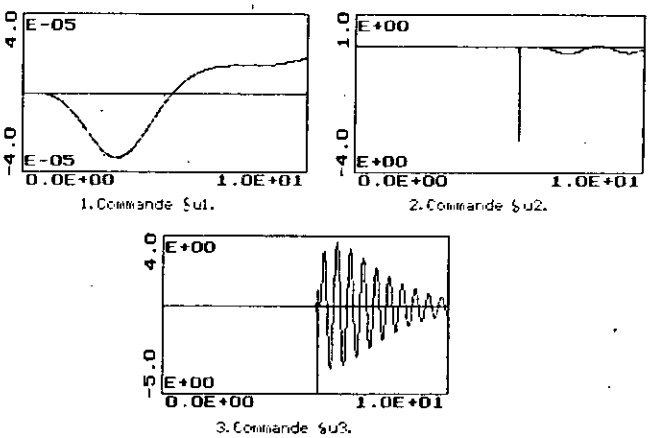
Figure IV.6.a Positions et erreurs en position pour APC.
(feedforward+feedback_STR; x1(t=0)=0.7)
(variation de 500% du modèle du robot à t=5s)



1. Commande u1. 2. Commande u2.

3. Commande u3.

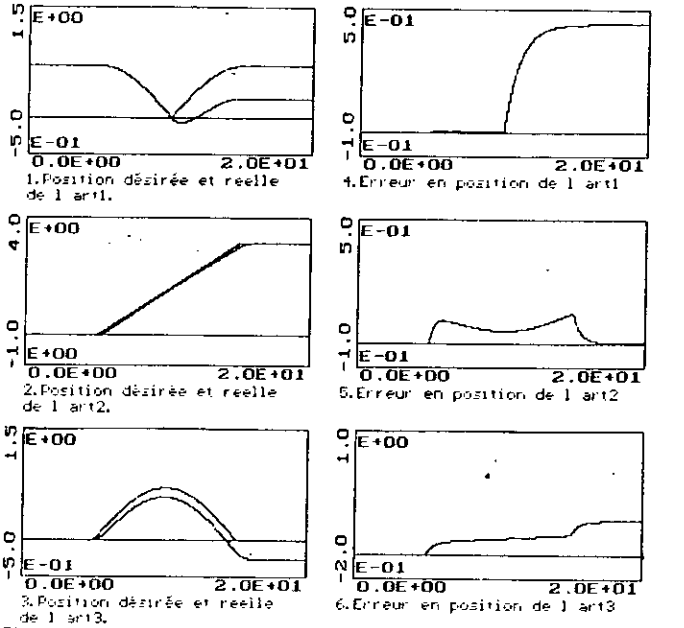
Figure IV.6.c Presentation des commandes.



1. Commande δu1. 2. Commande δu2.

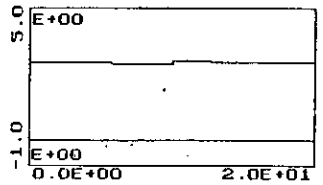
3. Commande δu3.

Figure IV.6.b Presentation des commandes δu.

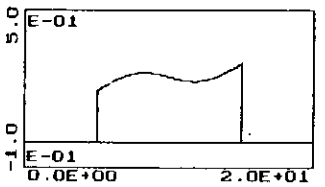


1. Position désirée et reelle de l art1.
4. Erreur en position de l art1
2. Position désirée et reelle de l art2.
5. Erreur en position de l art2
3. Position désirée et reelle de l art3.
6. Erreur en position de l art3

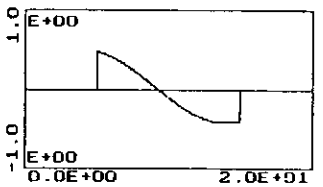
Figure IV.7.a Positions et erreurs en position dans le cas du couplage de Newton-Euler et d Euler-Lagrange.(feedforward; x1(t=0)=0.7)
(consigne fenetre de UDWANI)



1.Commande ud1.

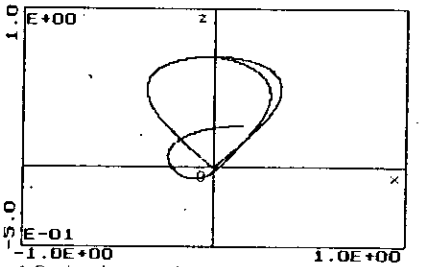


2.Commande ud2.

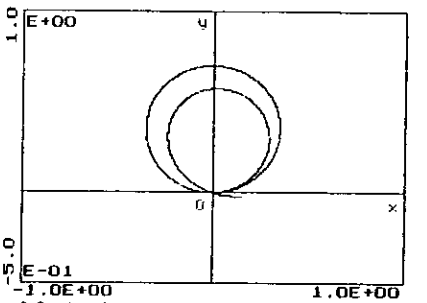


3.Commande ud3.

Figure IV.7.b Presentation des commandes.

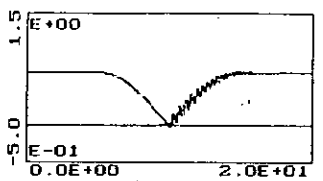


1.Projection sur le plan xoz.

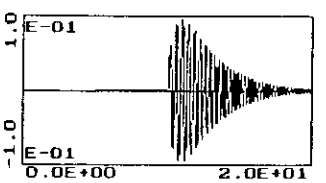


2.Projection sur le plan xoy.

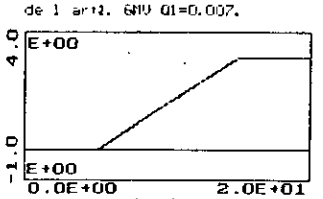
Figure IV.7.c Projection des trajectoires désirées et réelles sur les plan principaux.



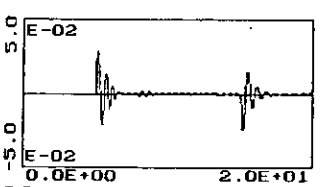
1.Position désirée et reelle de l art1. GNU Q1=0.007.



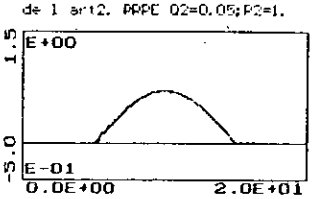
4.Erreur en position de l art1



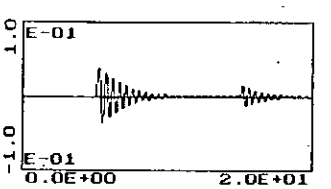
2.Position désirée et reelle de l art2. PRPE Q2=0.05;P2=1.



5.Erreur en position de l art2

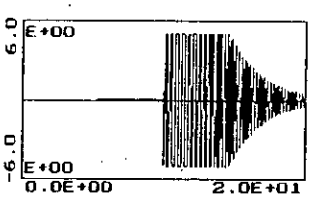


3.Position désirée et reelle de l art3. PRPE Q3=0.005;P3=1.

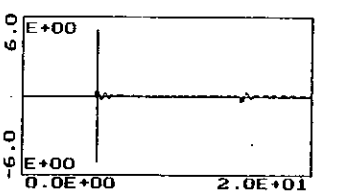


6.Erreur en position de l art3

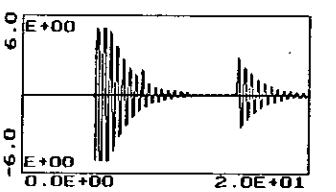
Figure IV.8.a Positions et erreurs en position pour APC.
(feedforward+feedback_STR; γ1=1=0.7 ;
' consigne fenetre de HOVIANI)



1.Commande $u1.

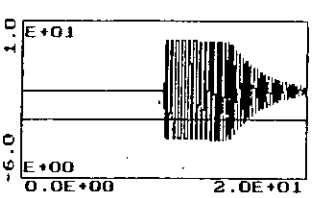


2.Commande $u2.

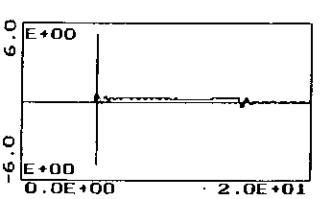


3.Commande $u3.

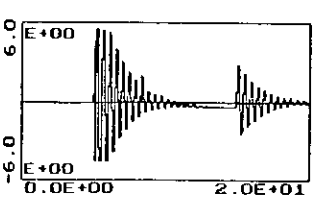
Figure IV.8.b Presentation des commandes $u.



1.Commande u1.

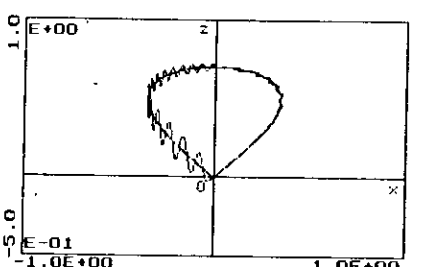


2.Commande u2.

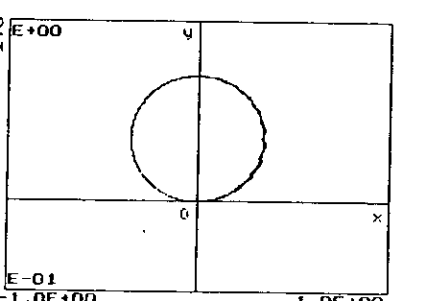


3.Commande u3.

Figure IV.8.c Presentation des commandes.



1.Projection sur le plan xoz.



2.Projection sur le plan xoy.

Figure IV.8.d Projection des trajectoires désirées et réelles sur les plan principaux.

Conclusion

Dans ce chapitre, on s'est intéressé à développer la commande "APC". En premier lieu, on a présenté le modèle à perturbation continu et discret. En second lieu, les algorithmes autoajustables à STR ont été développés dans un objectif de régulation seulement.

L'algorithme récursif de Newton-Euler, permet le calcul de la commande désirée, a 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.
Dans le cas où le modèle du robot n'est pas assujetti à des erreurs de modélisation, une telle commande est idéale. 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).Dans la pratique, de telles exigences ne sont pas satisfaites. L'implémentation directe d'une telle commande détériore le robot, surtout dans le cas où celui-ci est soumis à des perturbations, car la commande élaborée est une commande en boucle ouverte.
Pour assurer de bonne capacité de régulation, l'utilisation d'un retour linéaire s'avère nécessaire. Le rôle de ce retour est de compenser les perturbations de sorties et les erreurs de modélisation.
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.
La poursuite de trajectoire est assurée par une telle commande. L'algorithme récursif de Newton-Euler est très sensible aux erreurs de modélisation, qui engendrent des écarts très important dans le cas de la poursuite d'une trajectoire. L'utilisation du retour adaptatif linéaire surmonte ce problème et réduit d'une façon considérable cet écart.

L'insensibilité de la commande APC, au variation paramétrique est une propriété très importante de cette stratégie.
La poursuite n'est pas assurée, dans le cas de trajectoires discontinues en vitesse, pour l'algorithme de Newton-Euler, sans retour linéaire. Pour assurer cette poursuite, l'introduction de ce retour s'avère nécessaire. On utilise alors la commande APC, mais de légères oscillations apparaissent dans la première articulation.

L'importance primordiale dans l'implémentation d'une telle commande est l'effort de commande, qui ne présente pas de fortes oscillations, par contre des commandes lisses.

D'un autre côté, l'implémentation industrielle de cette stratégie, nécessite un processeur rapide, opérant à une période d'échantillonnage très faible, pour l'algorithme de NE et un autre processeur pour le calcul de la commande à perturbation.

Chapitre V

COMMANDE LINEARISANTE ADAPTATIVE

"Lorsqu'un théoricien trouve un résultat nouveau,
personne n'y croit sauf lui!
lorsqu'un expérimentateur trouve un résultat nouveau,
tout le monde y croit sauf lui!"

JEAN LEMAITRE

Chapitre V

COMMANDE LINEARISANTE ADAPTATIVE

Introduction.

Les stratégies de commande développées dans les chapitres précédents se basent sur l'approximation du modèle de connaissance du robot par un modèle moins complexe. Dans la commande auto-ajustable (STR) , le robot est représenté par un modèle totalement découplé. Tandis que dans les autres algorithmes de commande, on compense quelques nonlinéarités pour que le modèle de représentation choisi soit valable.
Le modèle obtenu par la compensation de la pesanteur est **non linéaire**, tandis que le modèle obtenu par le découplage **non linéaire** partiel est quasi-découplé. La construction du modèle de base (model based), dans le découplage total, aboutit à un modèle totalement découplé, mais dont le comportement est un double intégrateur, ce qui nécessite une autre loi de commande externe.

Dans ce chapitre, on introduira 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** [89][91][93][94].
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 [93].

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 paramétriques 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** [88][95][96].

Benallegue [93] expose une étude des différentes lois de commande dynamique adaptatives. Dans la commande dynamique robuste, l'analyse de la stabilité est faite en utilisant la passivité et l'hyperstabilité [97]. L'implémentation d'une telle loi nécessite quelques conditions sur le modèle du robot et sur la vitesse de la trajectoire désirée. L'algorithme d'adaptation paramétrique est celui du type PI (Proportionnel Intégral).
Hamy Blighuis & Al [92] proposent une nouvelle loi de commande dynamique robuste, sans restriction sur le modèle du robot, ni sur la trajectoire désirée (restrictions moins exigeantes). Tandis que l'algorithme d'identification utilisé est du type I (Intégral).
Craig [90][91] 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.

Dans la première partie de ce chapitre, on présente la commande linéarisante à paramètres fixes. La version prédictive de cette commande est développée dans la seconde partie. La troisième partie concerne la commande linéarisante adaptative. Le développement de l'algorithme d'adaptation paramétrique est présenté dans la quatrième partie. Des résultats de simulation seront présentés dans la dernière partie. On termine par une conclusion.

V.1 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:

$$M(q)\ddot{q}+V(q,\dot{q})+G(q)+f(\dot{q}) = \tau \qquad (V.1)$$

Où M, V, G, f et τ sont définis dans (III.1).

La boucle de la linérisation est réalisée par le choix d'un couple τ à appliquer au robot, de la forme suivante [90][93]:

$$\tau = M_0(q)\tau'+V_0(q,\dot{q})+G_0(q)+f_0(\dot{q}) \qquad (V.2)$$

Où M_0, V_0, G_0, et f_0 sont les estimés de M, V, G, et f.

La combinaison des équations (V.1) et (V.2) donne :

$$M(q)\ddot{q}+V(q,\dot{q})+G(q)+f(\dot{q}) = M_0(q)\tau'+V_0(q,\dot{q})+G_0(q)+f_0(\dot{q}) \qquad (V.3)$$

Dans le cas où le modèle du robot est connu avec une précision suffisante, on a:

$$\begin{aligned} M(q) &\approx M_0(q)\ ;\ G(q) \approx G_0(q) \\ V(q,\dot{q}) &\approx V_0(q,\dot{q})\ ;\ f(q) \approx f_0(q) \end{aligned} \qquad (V.4)$$

L'équation (V.2) devient alors :

$$\ddot{q} = \tau' \qquad (V.5)$$

dans le cas où $M^{-1}(q)$ est nonsinguliére [93].

Ce qui transforme le problème de commande du robot en celui de la commande de n doubles intégrateurs découplés. L'entrée τ' 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 [91][93]:

$$\tau' = \ddot{q}_d - k_v\dot{e}(t) - k_p e(t) \qquad (V.6)$$

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 ou erreur en vitesse du système:

$$\begin{aligned} e(t) &= q(t) - q_d(t) \\ \dot{e}(t) &= \dot{q}(t) - \dot{q}_d(t) \end{aligned} \qquad (V.7)$$

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 ξ et pulsation ω_0. Ceci peut être réalisé par [91]:

$$k_{v_i} = 2\xi_i\omega_{0_i};\ k_{p_i} = \omega_{0_i}^2 \qquad (V.8)$$

En remplaçant l'expression de τ' (V.6) dans L'équation (V.5), on obtient L'équation de l'erreur en boucle fermée:

$$M(q)(\ddot{e}(t)+k_v\dot{e}(t)+k_p e(t)) = 0 \qquad (V.9)$$

L'équation (V.9) a pour solution un signal e(t) qui tend exponontiellement vers zéro, car la matrice d'inertie est non singulière [93].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 V.1 [21].

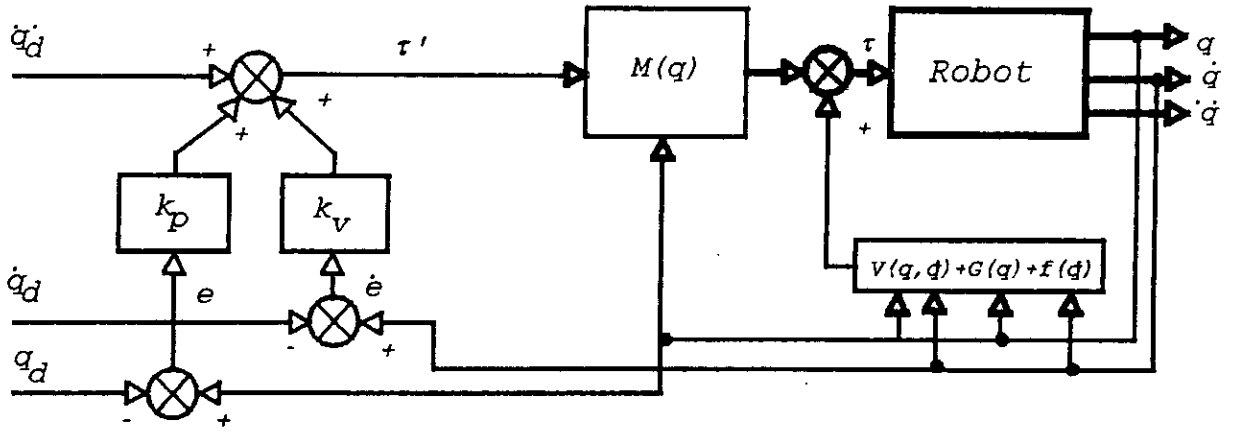


Figure V.1 Architecture de la commande linéarisante à paramètres connus.

Algorithme de commande:

Dans le cas où les paramètres dynamiques du robot sont connus, l'algorithme de la commande linéarisante est :

Données: modèle du robot M, V, G, f.
Choix de k_p et k_v.
Etape 1: Génération de trajectoire.
Etape 2: Calcul de l'erreur en position et en vitesse (équation (V.7)).
Etape 3: Calcul de τ' de L'équation (V.6).
Etape 4: Calcul de τ de L'équation (V.2).
t=t+1 revenir à étape 1.

V.2 Commande linéarisante prédictive.

L'implémentation pratique de la commande par dynamique inverse (commande linéarisante à paramètres connus), nécessite beaucoup de temps de calcul, pour les termes nonlinéaires, qui sont fonction des positions et des vitesses mesurées à chaque instant. Pour réduire cette complexité de calcul, un autre schéma de commande peut être utilisé.

La commande prédictive peut être obtenue à partir de la commande linéarisante en remplaçant les variables $q(t)$ et $\dot{q}(t)$ par celles des trajectoires désirées $q_d(t)$ et $\dot{q}_d(t)$. Ces dernières peuvent être connus à l'avance, par conséquent tous les calculs leurs correspondant peuvent étre fait "hors-ligne", ce qui réduit la complexité des calculs nécessaires en ligne ou en temps réel. Les termes calculés hors-ligne sont $M(q_d)$ et $V(q_d,\dot{q}_d)+G(q_d)+f(\dot{q}_d)$, L'expression de la commande deviendra [93]:

$$\tau = M(q_d)\tau' + V(q_d, \dot{q}_d) + G(q_d) + f(\dot{q}_d) \qquad (V.10)$$

et

$$\tau' = \ddot{q}_d - k_v\dot{e}(t) - k_p e(t) \qquad (V.11)$$

Ces calculs sont stockés en mémoire pour être utilisés lors de la commande. Le schéma de la commande ainsi obtenu est donné par la figure V.2.

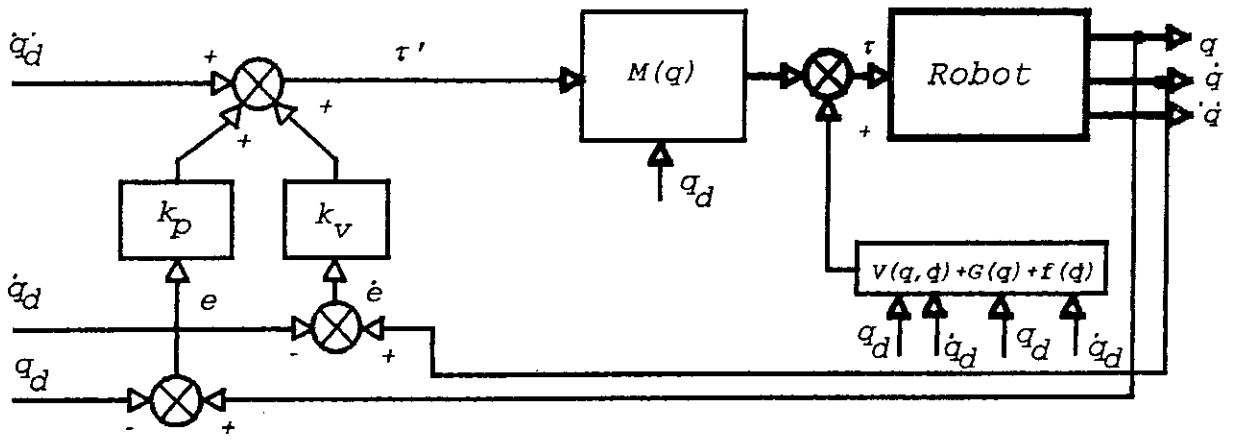


Figure V.2 Commande linéarisante prédictive.

Algorithme de commande:

Données: modèle du robot M, V, G, f.
Choix de k_p et k_v.
Génération et stockage des trajectoires q_d, $\dot{q}_d$ et $\ddot{q}_d$.
Calcul et stockage de $M(q_d)$ et $V(q_d, \dot{q}_d) + G(q_d) + f(\dot{q}_d)$.
Etape 1: Calcul de l'erreur en position et en vitesse. équation (V.7).
Etape 2: Calcul de τ' de L'équation (V.11).
Etape 3: Calcul de τ de L'équation (V.10).
t=t+1 revenir à étape 1.

V.3 Commande linéarisante adaptative.

Dans le cas où il existe une perturbation (τ_p) sur les couples appliqués au manipulateurs, L'équation du système s'écrit [89]:

$$\tau + \tau_p = M(q)\ddot{q} + V(q, \dot{q}) + G(q) + f(\dot{q}) \qquad (V.12)$$

En pratique, la perturbation τ_p peut matérialiser des dynamiques négligées dans la modélisation du système, telle que l'imprécision de la modélisation des frottements nonlinéaire ou des imperfections des actionneurs.
Avec la commande définie par (V.2) et (V.6), L'équation de l'erreur devient :

$$\ddot{e}(t) + k_v\dot{e}(t) + k_p e(t) = M^{-1}(q)\tau_p \qquad (V.13)$$

La perturbation est bien rejetée avec la dynamique imposée au système en BF. Par contre, elle est modulée par l'inverse de la matrice d'inertie et on remarque que le membre de droite n'est pas découplé, car $M(q)$ n'est pas forcement diagonale. Donc,

une perturbation sur une seule articulation, sera répercutée sur les autres articulations et produire une erreur de suivi. Ceci est du fait que le système est non linéaire est que le découplage ne concerne que la loi de commande.

En pratique, on ne dispose que d'une estimation plus ou moins précise des paramètres du robot, tel que les termes d'inertie, de coriolis, de centrifuge et de frottement. On utilise dans ce cas le modèle estimé et la commande est [89]:

$$\tau = \hat{M}(q)\tau' + \hat{V}(q,\dot{q}) + \hat{G}(q) + \hat{f}(\dot{q}) \qquad (V.14)$$

L'application de cette commande, au robot défini par (V.1) donne:

$$\hat{M}(q)\tau' + \hat{V}(q,\dot{q}) + \hat{G}(q) + \hat{f}(\dot{q}) = M(q)\ddot{q} + V(q,\dot{q}) + G(q) + f(\dot{q}) + \hat{M}(q)\ddot{q} - \hat{M}(q)\ddot{q} \qquad (V.15)$$

D'où en utilisant (V.6) on obtient:

$$\ddot{e}(t) + k_v\dot{e}(t) + k_p e(t) = -M^{-1}(q)\,(\Delta M\,\ddot{q} + \Delta V + \Delta G + \Delta f) \qquad (V.16)$$

Avec

$$\begin{aligned} \Delta M &= M(q) - \hat{M}(q)\ ;\ \delta G = G(q) - \hat{G}(q) \\ \Delta V &= V(q,\dot{q}) - \hat{V}(q,\dot{q})\ ;\ \Delta f = f(\dot{q}) - \hat{f}(\dot{q}) \end{aligned} \qquad (V.17)$$

Si les estimations sont précises alors l'erreur de suivi s'annulera. Par contre, dans le cas d'une erreur de modélisation, une excitation interne nonlinéaire (dynamique générée par l'erreur paramétrique), se propage dans le système en BF en provoquant une erreur de suivi de trajectoire.

Dans cette partie, on présente le schéma adaptative de la commande linéarisante, qui prend en considération l'effet des perturbations sur le couple et sur les paramètres dynamiques. La commande ainsi obtenue maintient les performances en BF, mais nécessite l'utilisation d'un algorithme d'adaptation paramétrique.

Pour faciliter la manipulation des termes non linéaires dans le modèle de connaissance du robot, le manipulateur est modélisé sous la forme compacte suivante [90][91]:

$$\tau = M(q)\ddot{q} + Q(q,\dot{q}) \qquad (V.18)$$

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 (V.2) devient:

$$\tau = \hat{M}(q)\tau' + \hat{Q}(q,\dot{q}) \qquad (V.19)$$

Où $\hat{M}(q)$ et $\hat{Q}(q,\dot{q})$ sont les estimés de $M(q)$ et $Q(q,\dot{q})$.

Le terme τ' est choisi de la même manière que (V.6):

$$\tau' = \ddot{q}_d - k_v\dot{e}(t) - k_p e(t) \qquad (V.20)$$

Des équations (V.18), (V.19) et (V.20) on obtient:

$$M(q)\ddot{q}+Q(q,\dot{q}) = \hat{M}(q)[\ddot{q}_d-k_v\dot{e}(t)-k_pe(t)]+\hat{Q}(q,\dot{q}) \qquad (V.21)$$

En ajoutant et en retranchant le terme $\hat{M}(q,\dot{q})\ddot{q}$, du terme de droite de L'équation (V.21), on obtient:

$$\ddot{e}(t)+k_v\dot{e}(t)+k_pe(t) = -M^{-1}(q)[\tilde{Q}+\tilde{M}\ddot{q}] \qquad (V.22)$$

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 (V.18), sous la forme suivante [91]:

$$\tau_i = \sum_{j=1}^{a_i} m_{ij}f_{ij}(q,\ddot{q})+\sum_{j=1}^{b_i} q_{ij}g_{ij}(q,\dot{q}) ; \quad i=1,n \qquad (V.23)$$

Où m_{ij} *et* q_{ij} sont des paramètres traduisant les masses, les éléments du tenseur d'inertie, longueurs et coefficients de frottement.
$f_{ij}(q,\ddot{q})$ *et* $g_{ij}(q,\dot{q})$ sont des fonctions nonlinéaires issues du modèle du robot après quelques arrangement.
a_i: nombre des paramètres m_{ij}.
b_i: nombre des paramètres q_{ij}.

L'écriture de L'équation (V.23) sous forme matricielle donne [91][93]:

$$\tau = W(q,\dot{q},\ddot{q})\theta \qquad (V.24)$$

Où $W(q,\dot{q},\ddot{q})$: matrice englobant toutes les fonctions nonlinéaires f_{ij} *et* g_{ij}.
θ: 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 [91][93]:

$$\hat{\tau}_i = \sum_{j=1}^{r_i} \hat{m}_{ij}f_{ij}(q,\ddot{q})+\sum_{j=1}^{s_i} \hat{q}_{ij}g_{ij}(q,\dot{q}) ; \quad i=1,n \qquad (V.25)$$

Sous forme matricielle, L'équation (V.25) devient:

$$\hat{\tau} = W(q,\dot{q},\ddot{q})\hat{\theta} \qquad (V.26)$$

Où $\hat{\theta}^T=[\hat{\theta}_1 \ldots \hat{\theta}_r]$ avec $r\leq\sum_{i=1}^{n}(s_i+r_i)$.

L'équation de l'erreur (V.22) relie les paramètres estimés au erreur de commande. La paramètrisation (V.26) donne une partition des termes connus et des termes mal connus. Cette partition nous permettera la construction du schéma adaptative.
Donc L'équation de l'erreur deviendra:

$$\ddot{e}(t)+k_v\dot{e}(t)+k_pe(t) = M^{-1}(q)W(q,\dot{q},\ddot{q})\tilde{\theta} \qquad (V.27)$$

Où $\tilde{\theta}=\hat{\theta}-\theta$: vecteur de dimension $r=\sum_{i=1}^{n}(s_i+r_i)$.

et $[\ W(q,\dot{q},\ddot{q})\tilde{\theta}\ ]_i = \sum_{j=1}^{r_i} \tilde{m}_{ij} f_{ij}(q,\ddot{q}) + \sum_{j=1}^{s_i} \tilde{q}_{ij} g_{ij}(q,\dot{q})\ ;\quad i=1,n.$ avec $\begin{matrix}\tilde{m}_{ij}=\hat{m}_{ij}-m_{ij}\\ \tilde{q}_{ij}=\hat{q}_{ij}-q_{ij}\end{matrix}$.

Dans l'analyse qui suit, il est important de noter que le produit $\hat{M}^{-1}W(q,\dot{q},\ddot{q})$ doit être borné [91]. D'une part la matrice W est composée de fonctions bornées $(f_{ij}\ ,\ g_{ij})$ dans le cas où la trajectoire du robot est bornée. D'autre part la matrice $\hat{M}(q)$ est SDP (Semi Définie Positive) dans le cas où les paramètres estimés évoluent au voisinage des paramètres réels. Avec cette motivation, les estimés des paramètres doivent être bornés de la façon suivante [91]:

$$Inf_i-\delta<\hat{\theta}_i<Sup_i+\delta \qquad (V.28)$$

Où Inf_i, Sup_i et δ sont choisis de sorte que $\hat{M}^{-1}(q)$ reste bornée [91].

V.3.1 Algorithme d'adaptation paramétrique.

L'algorithme d'adaptation calcule l'évolution des estimés en utilisant le signal d'erreur filtré. Ce signal d'erreur filtré est défini par [91][93]:

$$e_1(t) = \dot{e}(t)+\psi e(t) \qquad (V.29)$$

Où $\psi=diag(\psi_1 \ldots \psi_n)$.

L'erreur filtrée de chaque articulation est alors :

$$e_{1_i}(t) = \dot{e}_i(t)+\psi_i e_i(t) \qquad (V.30)$$

D'où en passant dans le plan de Laplace:

$$e_{1_i}(s) = (s+\psi_i)\, e_1(s) \qquad (V.31)$$

Où s est l'opérateur de Laplace.

Les coefficients ψ_i sont choisis de sorte que la fonction de transfert:

$$F_i(s) = \frac{s+\psi_i}{s^2+k_{v_i}s+k_{p_i}} \qquad (V.32)$$

soit SPR (Stricte Réelle Positive) [91][93].

Remarque.
Une fonction rationnel $T(s)$ est SPR, si $T(s)$ est analytique et $Re[T(j\omega)]>0,\ \forall\omega$. □

Avec cette positivité réelle stricte de $F_i(s)$, on assure l'existence des matrices définies positives P_i et Q_i tel que [91][93]:

$$\begin{cases} A_i^T P_i + P_i A_i = -Q_i \\ P_i B_i = C_i^T;\quad i=1,n \end{cases} \qquad (V.33)$$

Où les matrices A_i, B_i et C_i sont les matrices de la réalisation minimale dans l'espace d'état de L'équation d'erreur filtrée (V.27) avec:

$$x_i^T=[e_i^T\ \dot{e}_i^T]$$

$$A_i=\begin{bmatrix}0 & 1\\ -k_{p_i} & -k_{v_i}\end{bmatrix};\ B_i=\begin{bmatrix}0\\1\end{bmatrix};\ C=[\psi_i\ 1].$$

L'écriture matricielle de L'équation (V.27) sous forme d'état est:

$$\begin{cases}\dot{x} = A\,x+B\,\hat{M}^{-1}(q)\,W(q,\dot{q},\ddot{q})\,\tilde{\theta}\\ e_1 = C\,x\end{cases} \qquad (V.34)$$

Où A, B et C: des matrices diagonales par bloc (avec A_i, B_i et C_i: sur leurs diagonales respectivement).
$x^T=[x_1\ldots x_n]$.

L'équation (V.33) devient alors :

$$\begin{cases}A^TP+PA = -Q\\ PB = C^T\end{cases} \qquad (V.35)$$

Où $P=diag(P_1\ldots P_n)$ et $Q=diag(Q_1\ldots Q_n)$.

En utilisant la théorie de stabilité de Lyapounov [88][89], on définit la fonction de Lyapounov candidate [91][93]:

$$V(x,\tilde{\theta}) = x^TPx + \tilde{\theta}^T\Gamma^{-1}\tilde{\theta} \qquad (V.36)$$

Où $\Gamma=diag(\gamma_1\ldots\gamma_r)$ et $\gamma_i>0$.

En différentiant la fonction $V(.,.)$ par rapport au temps, on aboutit à (utiliser (V.34) et (V.35)):

$$\dot{V}(x,\tilde{\theta}) = -x^TQx + 2\tilde{\theta}^T\left(W^T\hat{M}^{-1}e_1+\Gamma^{-1}\dot{\tilde{\theta}}\right) \qquad (V.37)$$

Si on choisit:

$$\dot{\tilde{\theta}} = -\Gamma W^T(q,\dot{q},\ddot{q})\,\hat{M}^{-1}(q)\,e_1(t) \qquad (V.38)$$

On obtient:

$$\dot{V}(x,\tilde{\theta}) = -x^TQx \qquad (V.39)$$

Etant donné que Q est SDP, donc x et $\tilde{\theta}$ sont bornées [91]. Donc l'algorithme d'adaptation est :

$$\dot{\hat{\theta}} = -\Gamma W^T(q,\dot{q},\ddot{q})\,\hat{M}^{-1}(q)\,e_1(t) \qquad (V.40)$$

C'est un algorithme du type Intégral [91][93].
Dans le cas où les estimés des paramètres ne vérifient pas L'équation (V.28), il suffit d'actualiser les paramètres $\hat{\theta}_i$ avec la condition suivante [91]:

$$\begin{aligned}\hat{\theta}_i(t^+) &= Inf_i \quad si\ \ \hat{\theta}_i(t)\le Inf_i-\delta\\ \hat{\theta}_i(t^+) &= Sup_i \quad si\ \ \hat{\theta}_i(t)\ge Sup_i+\delta\end{aligned} \qquad (V.41)$$

Ainsi les paramètres restent toujours bornés, sans aucune influence sur la stabilité

du système. La dérivé de la fonction de Lyapounov définie par (V.36) restera toujours négative [91].

V.3.2 Convergence de l'algorithme d'adaptation paramétrique.

L'équation de l'erreur est défini par:

$$\begin{cases} \dot{x} = A\,x + B\,\hat{M}^{-1}(q)\,W(q,\dot{q},\ddot{q})\,\tilde{\theta} \\ e_1 = c\,x \end{cases} \qquad \text{(V.42.a)}$$

et

$$\dot{\tilde{\theta}} = -\Gamma W^T(q,\dot{q},\ddot{q})\,\hat{M}^{-1}(q)\,e_1(t) \qquad \text{(V.42.b)}$$

La théorie de stabilité de Lyapounov et d'hyperstabilité de Popov [88][89] permet l'analyse de la stabilité d'un tel algorithme.
L'architecture de l'algorithme est schématisée sur la figure V.3.

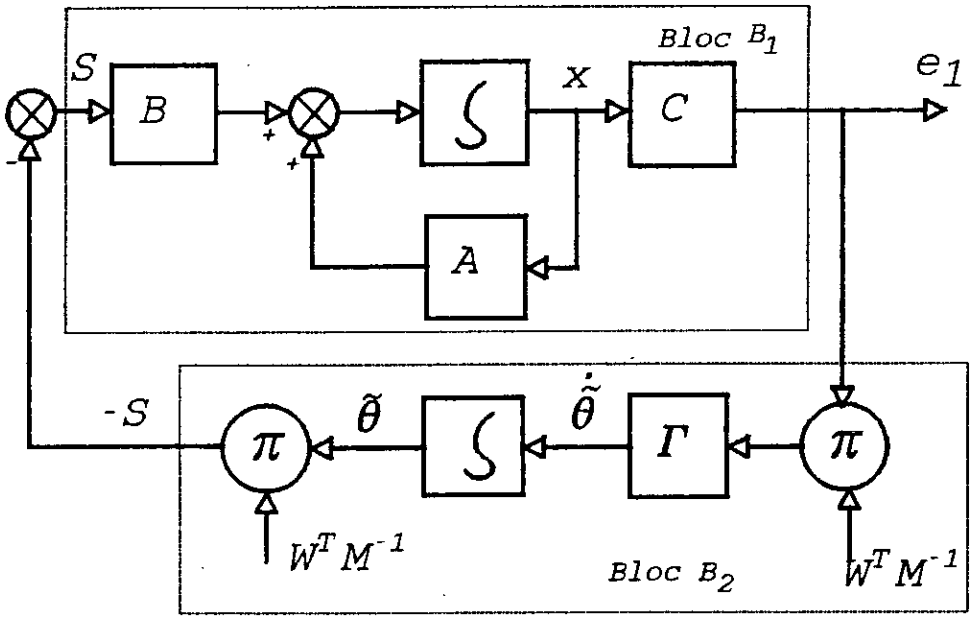


Figure V.3 Schéma de l'algorithme d'adaptation paramétrique ($S=M^{-1}W\,\Theta$).

La structure de la figure V.3 est constituée, d'un bloc linéaire dans la chaîne directe (bloc B_1) et d'un retour **non linéaire** (bloc B_2). Pour démontrer la stabilité asymptotique d'une telle structure, il suffit d'utiliser le théorème d'hyperstabilité asymptotique [88][93] (voir annexe 7).
La chaîne directe est SPR par construction, suivant le choix de ψ.
L'inégalité de Popov doit être vérifiée pour le retour non linéaire. Il suffit de montrer que:

$$I = \int_0^t -S^T e_1\,dt \geq -\gamma_0^2 \qquad \text{(V.43)}$$

Suivant l'architecture de la figure V.3, on a:

$$I = \int_0^t -\tilde{\theta}^T W^T(q,\dot{q},\ddot{q})\,\hat{M}^{-1}(q)\,e_1\,dt \qquad \text{(V.44)}$$

De L'équation (V.42.b) on a:

$$W^T(q,\dot{q},\ddot{q})\hat{M}^{-1}(q)\,e_1(t) = -\Gamma^{-1}\dot{\tilde{\theta}} \tag{V.45}$$

En remplaçant cette expression dans (V.44), on obtient:

$$I = \int_0^t -\tilde{\theta}^T\Gamma^{-1}\dot{\tilde{\theta}}\,dt \tag{V.46}$$

qui peut se mettre sous la forme suivante:

$$I = \frac{1}{2}\int_0^t \frac{d}{dt}(\tilde{\theta}^T\Gamma^{-1}\tilde{\theta})dt \tag{V.47}$$

De L'équation (V.47) on peut dire que [93]:

$$I \geq -\frac{1}{2}\tilde{\theta}^T(0)\Gamma^{-1}\tilde{\theta}(0) \tag{V.48}$$

La matrice Γ^{-1} est définie positive alors:

$$\gamma_0^2 \geq -\frac{1}{2}\tilde{\theta}^T(0)\Gamma^{-1}\tilde{\theta}(0) \tag{V.49}$$

Donc le retour non linéaire vérifie l'inégalité de Popov. On conclue alors que l'algorithme d'adaptation est asymptotiquement hyperstable.

V.3.3 Structure de l'algorithme de la commande linéarisante adaptative.

La structure de l'algorithme de commande linéarisante adaptative est illustrée sur le schéma de la figure V.4.

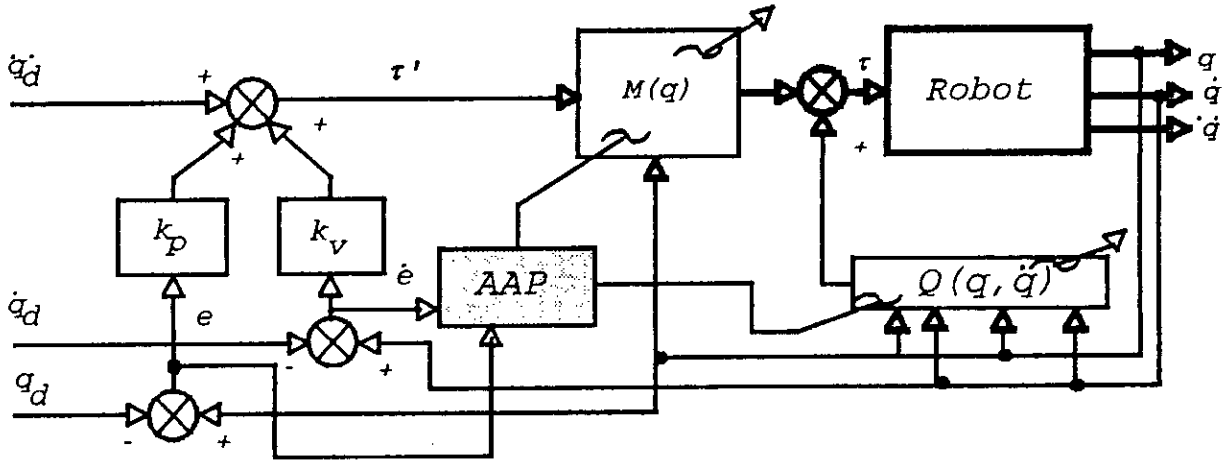


Figure V.4 Schéma de la commande linéarisante adaptative.

Algorithme de commande:

Données: modèle du robot $\hat{M}$, $\hat{Q}$.
Choix de k_p, k_v, Γ et ψ.
Etape 1: Génération de trajectoire.
Etape 2: Calcul de l'erreur filtrée définie par (V.29).
Etape 3: Estimation des paramètres à partir de (V.40).
Etape 4: Calcul de τ' de L'équation (V.20).
Etape 5: Calcul de τ de L'équation (V.19).
t=t+1 revenir à l'étape 1.

V.4 Résultats de simulation.

Le robot est le même que celui utilisé dans les parties précédentes.

Dans le cas où on suppose connaître le modèle du robot d'une façon acceptable. L'application de la commande linéarisante à paramètres connus s'avère très utile. Les résultats de simulation d'une telle commande, à une période d'échantillonnage de 2ms, sont consignés sur les figures V.5.a et b. La référence est un polynôme du troisième degré qui assure la continuité en position et en vitesse [21].
La simulation de la commande linéarisante prédictive a donné les mêmes résultats que la commande précédente.

Par contre une erreur de modélisation nécessite l'identification des paramètres mal connus. La simulation de la commande linéarisante adaptative, où la paramètrisation est: $\tau = W(q,\dot{q},\ddot{q})\hat{\theta}$.

avec $W(q,\dot{q},\ddot{q}) = \begin{bmatrix} w_{11} & w_{12} & w_{13} & 0 & 0 \\ 0 & w_{22} & 0 & w_{24} & 0 \\ 0 & w_{32} & 0 & 0 & w_{35} \end{bmatrix}$; et $\hat{\theta}^T = [m_1 \; m_3 \; f_1 \; f_2 \; f_3]$.

$w_{11} = w_{12} = \ddot{q}_1 + g$; $w_{13} = \dot{q}_1$; $w_{22} = (q_3^2 + \frac{l_2^2}{3} - q_3 l_2)\ddot{q}_2 + 2(q_3 - \frac{l_2}{2})\dot{q}_2\dot{q}_3$;

$w_{24} = \dot{q}_2$; $w_{35} = \dot{q}_3$; $w_{32} = \ddot{q}_3 - (q_3 - \frac{l_2}{2})\dot{q}_2^2$,

nécessite le choix des pondérations suivantes [93]:
$\psi_i = 5$; $i = 1,5$ *et* $\Gamma = diag(10,10,5,5,5)$.
et les paramètres initiaux $\hat{\theta}_0 = [10\ 10\ 10\ 10\ 2]$.

Dans de telles conditions et avec les mêmes k_p *et* k_v que précédemment, les figures V.6.a et b illustrent les résultats de simulation. Pour accélérer la convergence de l'erreur en position, on augmente la valeur du vecteur $\Gamma^T = [100\ 100\ 50\ 50\ 50]$.
La convergence rapide, de la commande linéarisante adaptative est illustrée dans les résultats consignés sur les figures V.7.a et b.

Pour tester la robustesse de la commande linéarisante à paramètres connus, on a calculé le retour non linéaire(model based) à partir d'un modèle erroné de 500% du modèle réel. Avec les mêmes caractéristiques que dans la figure V.5, les résultats de simulation, dans un tel cas, sont consignés sur les figures V.8.a et b, avec une limitation de la commande.

Pour diminuer les oscillations de la commande, on simule la commande linéarisante adaptative en faisant varier le modèle de 500% à 5s, avec les caractéristique suivantes:
$\hat{\theta}_0 = [10\ 10\ 10\ 10\ 2]$; $k_{p_i} = 100$; $k_{v_i} = 10$;
$\psi = [5\ 5\ 5]$; $\Gamma = diag(100,100,50,50,50)$.

Les résultats de simulation dans un tel cas se trouvent sur les figures V.9.a et b.

Enfin, pour tester la capacité de poursuite de la commande linéarisante à paramètres connus, on applique au robot une consigne calculée à partir de la fenêtre de VIVIANI. La simulation d'une telle commande avec k_{p_i}=100 *et* k_{v_i}=100, donne des résultats acceptable, consignés sur les figures V.10.a, b et c.
Pour diminuer l'erreur de poursuite, l'application de la commande linéarisante adaptative s'avère nécessaire. La simulation d'un tel algorithme dans les conditions suivantes:

$\hat{\theta}_0$=[10 10 10 10 2]; k_{p_i}=100; k_{v_i}=10;

ψ=[5 5 5]; Γ=*diag*(1000,1000,500,500,500).

donne les résultats consignés sur les figure V.11.a, b et c.

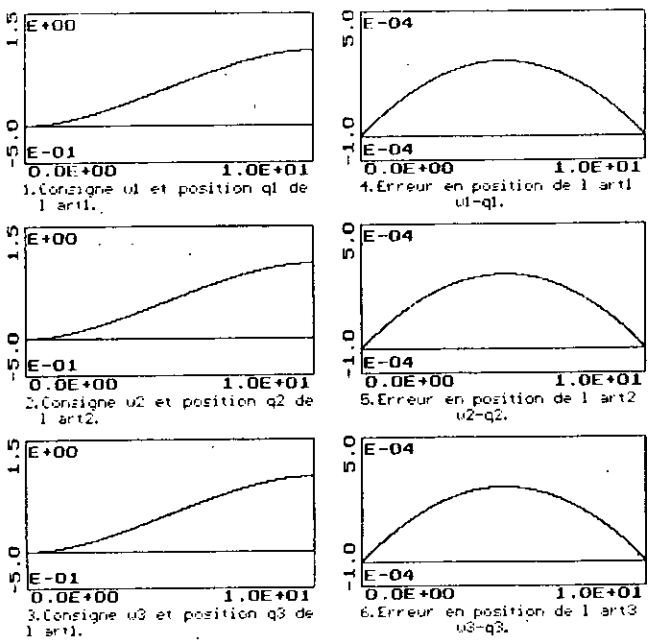


Figure V.9.a Position et erreur en position pour la commande linéarisante à paramètres connus.(kpi=100; kvi=20)

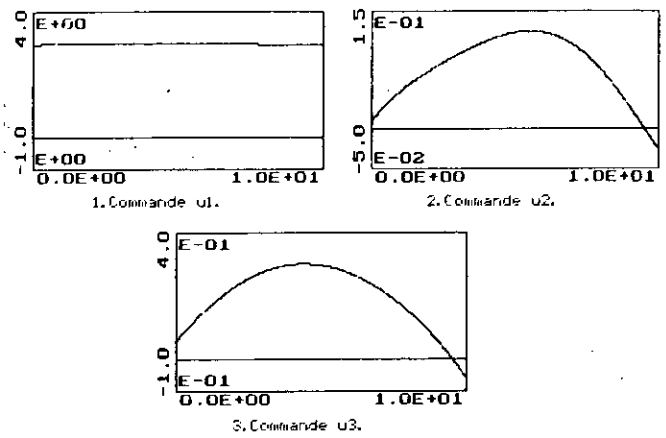


Figure V.9.b Présentation des commandes.

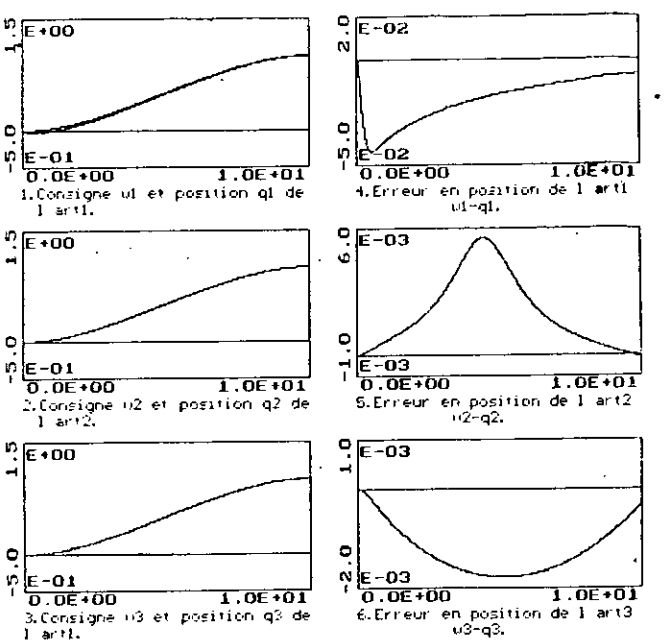


Figure V.10.a Position et erreur en position pour la commande linéarisante adaptative.(kpi=100;kvi=20;$\hat{\theta}$=(10 10 10 10 2);Γ=(10 10 5 5 5);Ψ=(5 5 5))

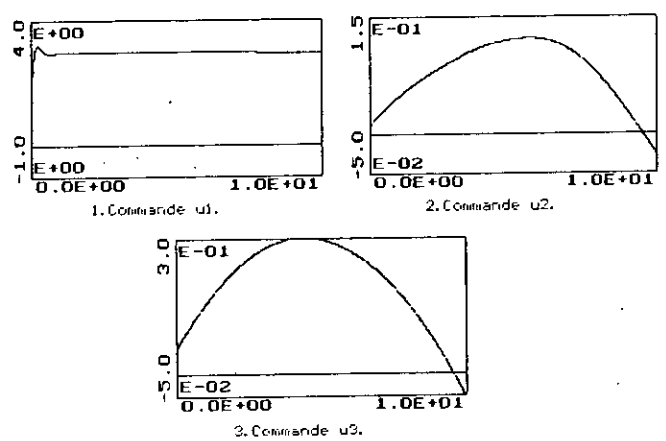
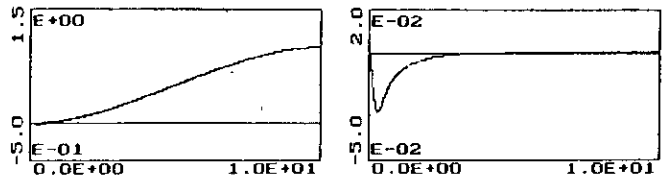
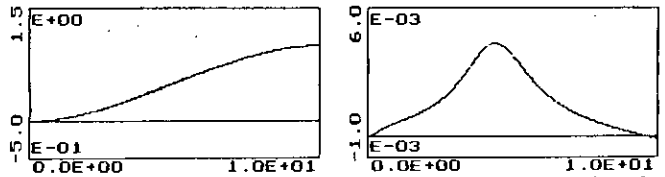


Figure V.10.b Présentation des commandes.



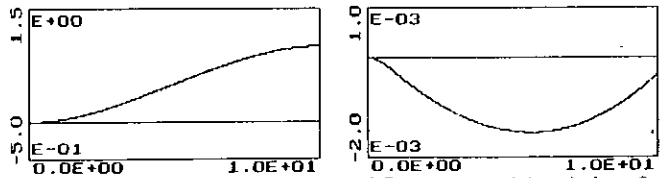
1.Consigne u1 et position q1 de l art1.

4.Erreur en position de l art1 u1-q1.



2.Consigne u2 et position q2 de l art2.

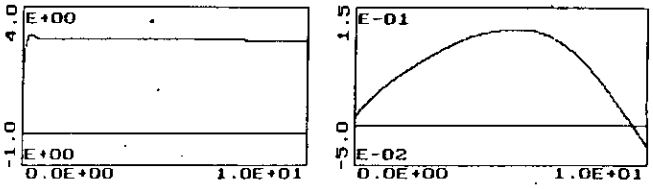
5.Erreur en position de l art2 u2-q2.



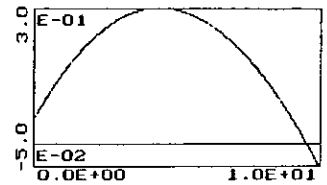
3.Consigne u3 et position q3 de l art1.

6.Erreur en position de l art3 u3-q3.

Figure V.7.a Position et erreur en position pour la commande linearisante adaptative.(kpi=100;kvi=20;θ_c=(10 10 10 10 2);F=(100 100 50 50 50);γ=(5 5 5)

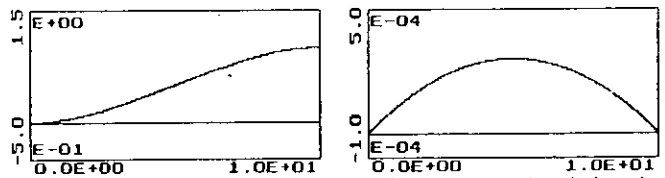


1.Commande u1. 2.Commande u2.



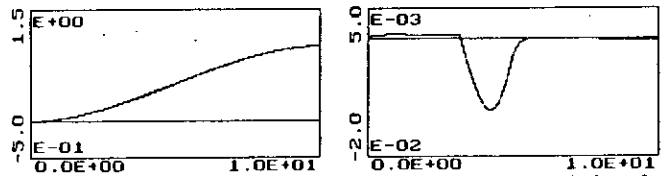
3.Commande u3.

Figure V.7.b Presentation des commandes.



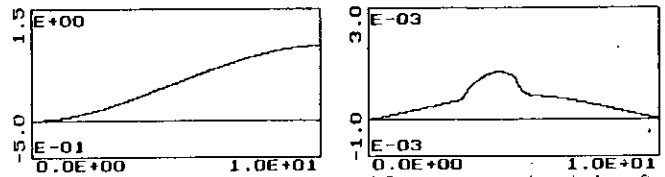
1.Consigne u1 et position q1 de l art1.

4.Erreur en position de l art1 u1-q1.



2.Consigne u2 et position q2 de l art2.

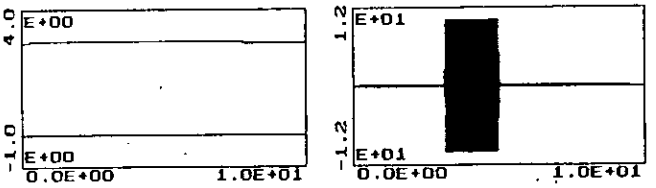
5.Erreur en position de l art2 u2-q2.



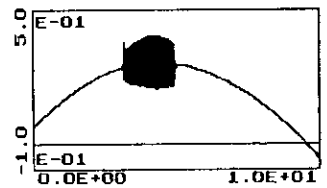
3.Consigne u3 et position q3 de l art1.

6.Erreur en position de l art3 u3-q3.

Figure V.8.a Position et erreur en position pour la commande linearisante à paramètres connus.(kpi=100;kvi=20;variation du modele du robot de 500%)

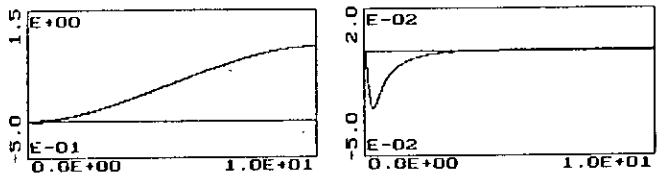


1.Commande u1. 2.Commande u2.



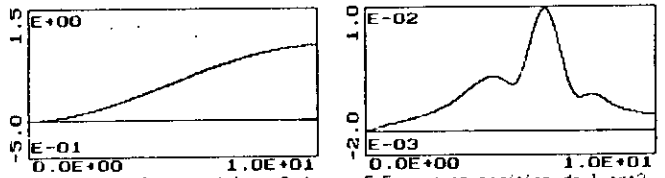
3.Commande u3.

Figure V.8.b Presentation des commandes.



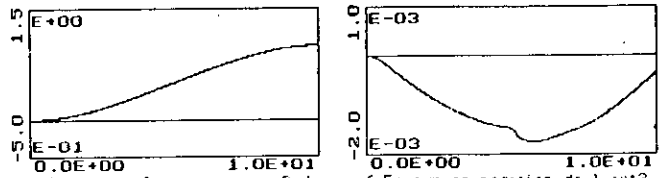
1.Consigne u1 et position q1 de l art1.

4.Erreur en position de l art1 u1-q1.



2.Consigne u2 et position q2 de l art2.

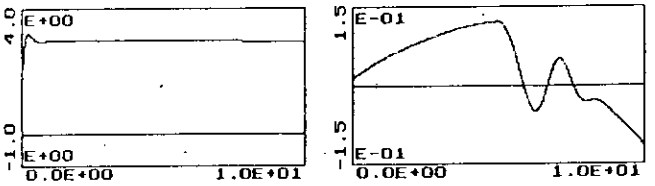
5.Erreur en position de l art2 u2-q2.



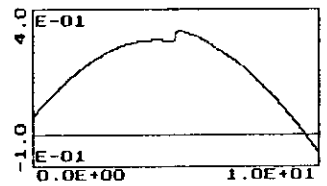
3.Consigne u3 et position q3 de l art1.

6.Erreur en position de l art3 u3-q3.

Figure V.9.a Position et erreur en position pour la commande linearisante adaptative.(kpi=100;kvi=20;θ=(10 10 10 10 2);F=(100 100 50 50 50);γ=(5 5 5); (variation du modèle du robot de 500% à t=5s)

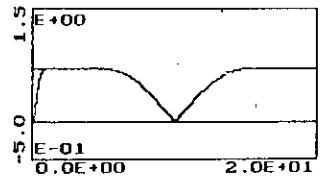


1.Commande u1. 2.Commande u2.

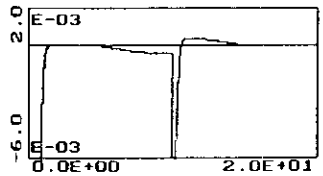


3.Commande u3.

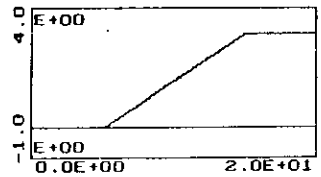
Figure V.9.b Presentation des commandes.



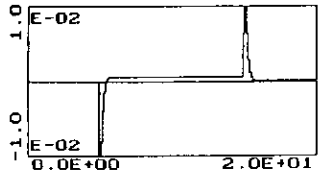
1.Consigne u1 et position q1 de l art1.



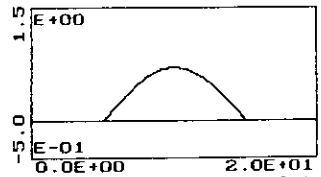
4.Erreur en position de l art1 u1-q1.



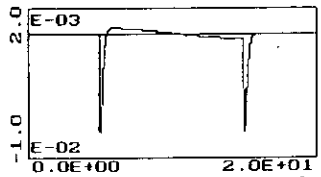
2.Consigne u2 et position q2 de l art2.



5.Erreur en position de l art2 u2-q2.

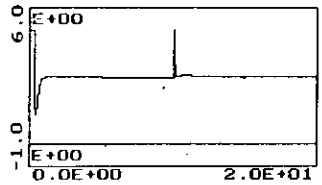


3.Consigne u3 et position q3 de l art1.

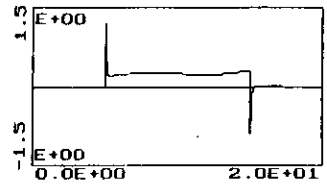


6.Erreur en position de l art3 u3-q3.

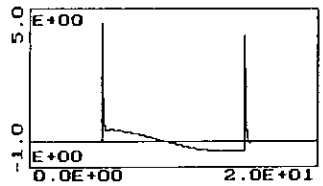
Figure V.10.a Position et erreur en position pour la commande linéarisante à paramètres connus.(kpi=100; kvi=20;)
(consigne fenetre de UIUIANI)



1.Commande u1.

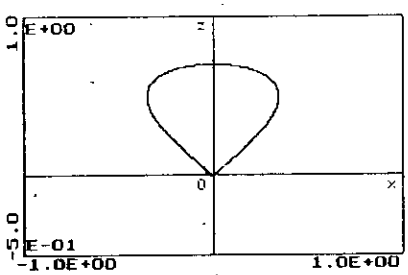


2.Commande u2.

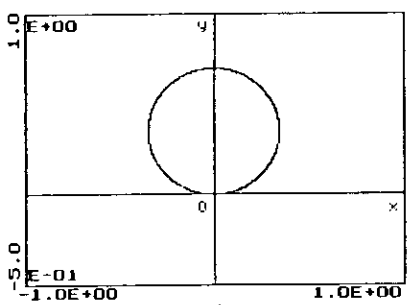


3.Commande u3.

Figure V.10.b Presentation des commandes.

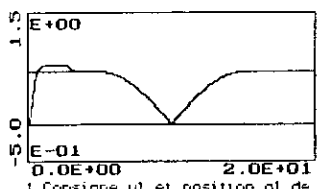


1.Projection sur le plan xoz.

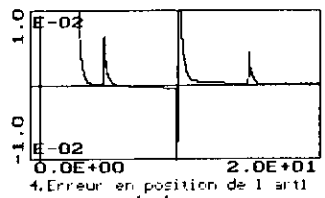


2.Projection sur le plan xoy.

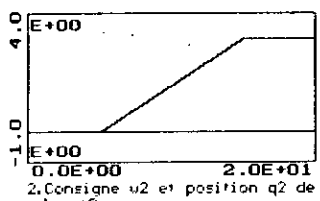
Figure V.10.c Projection des trajectoires désirées et réelles sur les plan principaux.



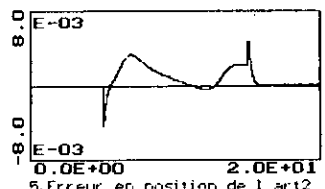
1.Consigne u1 et position q1 de l art1.



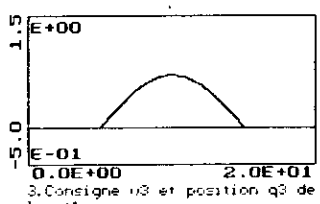
4.Erreur en position de l art1 u1-q1.



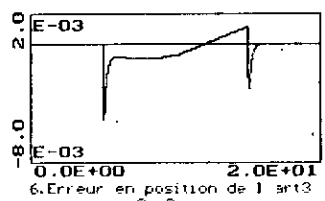
2.Consigne u2 et position q2 de l art2.



5.Erreur en position de l art2 u2-q2.

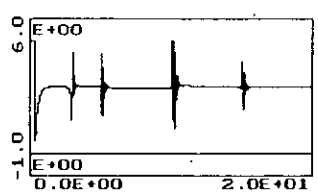


3.Consigne u3 et position q3 de l art1.

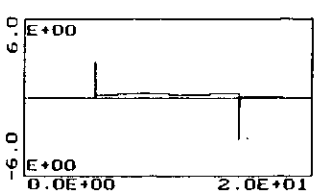


6.Erreur en position de l art3 u3-q3.

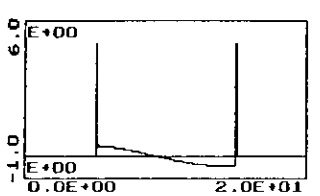
Figure V.11.a Position et erreur en position pour la commande linéarisante adaptative.(kpi=100;kvi=20;θ=<10 10 10 10 2>;Γ=<10000 10000 500 500 500>; ψ=<5 5 5>; consigne fenetre de UIUIANI.)



1.Commande u1.

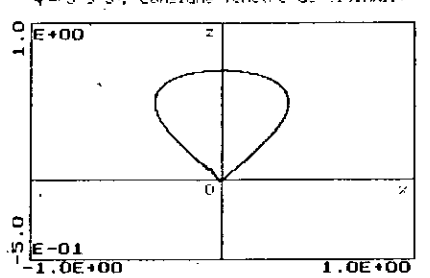


2.Commande u2.

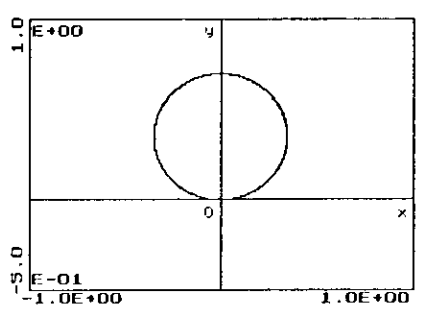


3.Commande u3.

Figure V.11.b Presentation des commandes.



1.Projection sur le plan xoz.



2.Projection sur le plan xoy.

Figure V.11.c Projection des trajectoires désirées et réelles sur les plan principaux.

Conclusion

Dans ce chapitre, on s'est intéressé à développer la commande linéarisante à paramètres connus, prédictive et adaptative.

L'implémentation de la commande linéarisante à paramètres connus, 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. Dans le cas de notre robot, une méconnaissance de 500% des paramètres du robot n'affecte pas les performances en BF, mais nécessite un effort de commande important et trop oscillatoire.
La capacité de poursuite est assurée avec un tel algorithme, même avec des discontinuités en vitesse, cependant l'erreur de poursuite est importante.
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.

L'implémentation pratique de la commande précédente, nécessite un temps de calcul, en temps réel, important. Pour surmonter ce problème, on a recours à la commande linéarisante prédictive, qui donne les mêmes résultats. Par contre elle nécessite un espace mémoire pour emmagasiner les valeurs des différents paramètres $M(q_d)$ et $V(q_d, \dot{q}_d) + G(q_d) + f(\dot{q}_d)$ et un processeur non nécessairement rapide.

Dans le cas de changement ou de mauvaise connaissance des paramètres dynamique du robot, la commande linéarisante adaptative s'avère nécessaire. La notion de passivité, de stabilité de Lyapounov et d'hyperstabilité permet l'étude de la stabilité d'un tel algorithme. Le choix des matrices ψ et Γ sont d'une importance primordiale.

En premier lieu, la matrice Γ, améliore la vitesse de convergence vers les paramètres réels. En second lieu, la matrice ψ, assure la positivité de la chaîne directe du schéma d'identification. L'algorithme d'identification est du type Intégral, ce qui permet l'ajustement des paramètres dés que l'erreur (en position et en vitesse) de poursuite est importante. La sensibilité de l'algorithme d'identification aux discontinuités en vitesse des trajectoires désirées, affecte l'ajustement des paramètres estimés d'une façon importante, même si ces derniers convergent vers les paramètres réels.

L'implémentation d'un tel algorithme d'identification, nécessite une mesure de l'accélération. Cette dernière peut être calculée numériquement, mais ceci n'est pas souhaitable, dans le cas des mesures bruitées.

L'adaptation permet de réduire l'effort de commande, dans le cas d'une erreur de modélisation. Elle permet aussi de minimiser l'erreur de poursuite, dans le cas des trajectoires désirées discontinues en vitesse, mais avec un effort de commande plus important aux instants de discontinuité.

Chapitre VI

SYSTEMES ADAPTATIFS A MODELE DE REFERENCE

"Car il est plus facile,
une fois qu'on a acquis une certaine connaissance des questions,
d'en imaginer ensuite la démonstration,
que si l'on recherchait celle-ci sans aucune notion préalable".

ARCHIMEDE

Chapitre VI

SYSTEMES ADAPTATIFS A MODELE DE REFERENCE

Introduction.

Dans cette partie, on s'intéresse à développer les techniques adaptatives à modèle de référence. Ces techniques sont des outils puissants pour l'analyse de la convergence et la synthèse des lois d'adaptation de ces structures ajustables [88].

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 [88][98]. Le caractère dual de ces méthodes permet leur utilisation suivant la structure spécifiée, dans différentes applications [36][88].
On distingue deux façons d'adapter le système ajustable, afin d'assurer une minimisation de l'erreur généralisée ou de sortie [36][39][88], 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 [36][88].

Le problème majeur dans la synthèse des systèmes "MRAS" est le "design" du mécanisme d'adaptation. Plusieurs recherches ont été développées pour synthétiser de tels mécanismes, à fin d'assurer une stabilité asymptotique [98].
La méthode de **MIT** (Massachusset Institute of Technology) [39][88] se base sur la minimisation du carré de l'erreur de sortie, dont la loi d'adaptation nécessite le calcul de la fonction sensibilité. Tandis que d'autres algorithmes [1][11] minimisent le carré de l'erreur ainsi que ses dérivés et aboutit à une loi d'adaptation nécessitant la disponibilité des dérivés successives de l'erreur.
L'analyse de la stabilité de tels algorithmes est très difficile, et n'assure pas en général une convergence de l'erreur [39][88]. Parks [99] montre l'incapacité de la méthode de MIT à assurer une convergence asymptotique de l'erreur. Il suggère une méthode de synthèse basée sur la fonction de Lyapounov. Le principal désavantage de la méthode est dû au fait qu'il n'existe pas une régle générale pour trouver une "bonne" fonction de Lyapounov. Ce qui empêche de résoudre le problème de la stabilité dans beaucoup de cas [39][88].
Le concept d'hyperstabilité introduit par Popov est une généralisation de la notion de stabilité absolue [95]. Landau [88][100] introduit ce concept pour l'analyse et la synthèse des lois d'adaptation. Il présente un théorème qui permet d'établir un critère simple de synthèse pour les systèmes hyperstables. Pour un choix, de loi d'adaptation bien spécifiée, Landau [101] démontre la convergence asymptotique de l'erreur. un tel outil permet l'analyse des systèmes adaptatifs dans le cas de procédé à paramètres, fortement ou faiblement variables dans le temps [102]. La version discrète du MRAS est présentée dans [103]. Landau [104] propose une comparaison entre différentes méthodes d'identification, par élimination de la condition de positivité réelle de la chaîne directe dans le schéma hyperstable équivalent.

Dans le domaine de la commande, la structure MRAS, se réduit à la détermination de la loi d'adaptation des paramètres du régulateur.
La commande par poursuite d'un modèle linéaire (Linear Model **Following** Control) est très adaptée à la structure **parallèle-parallèle** [36][88]. La satisfaction des conditions d'Erzenberger est une condition nécessaire pour l'existence du contrôleur linéaire assurant la poursuite parfaite [36][88] du modèle de référence. Dans le cas

des systèmes à paramètres inconnus ou variable dans le temps, l'utilisation d'une loi d'adaptation s'avère nécessaire. On aboutit à la commande adaptative à modèle de référence "**MRAC**" (**M**odel **R**eference **A**daptive **C**ontrol) [36][88]. L'hyperstabilité d'un tel algorithme est assurée, dans le cas où le bloc de contre réaction nonlinéaire, dans le schéma équivalent, assure l'inégalité de Popov et le bloc linéaire de la chaîne directe assurant la condition de positivité réelle [88]. Cette dernière condition est assurée par introduction d'un compensateur linéaire dans la chaîne directe et la résolution de l'équation de Lyapounov ou bien vérifiant le critère **MKY** (Meyer Kalman Yokotovic) [36][39].

Johnstone [105] présente une comparaison entre un contrôleur PI (Proportionnel Intégral) et l'MRAC, en utilisant l'erreur augmentée, dans la loi d'adaptation. Le critère de la commande est similaire à celui du GMV.
L'application du "AMFC" aux robots manipulateurs est présentée par Stoten [106][107], en comparant ce dernier au contrôleur **SFCIE** (State Feedback Control with Integrated Error).

Stoten & Benchoubane [108] proposent l'algorithme "**MCS**" (**M**inimum **C**ontroller **S**unthésis) similaire à L'AMFC, ne nécessitant pas la connaissance du modele lineaire du système, par contre le degré de ce dernier est nécessaire. Pour démontrer l'efficacité d'un tel algorithme, l'implémentation pratique, en langage C, a été faite, ainsi que des résultats de simulation. La supériorité de cet algorithme par rapport à SFCIE, MRAC et PI a été démontrée. L'étude de sa robustesse a été faite [109]. Pour améliorer la vitesse de convergence du MCS, Benchoubane [110] donne l'influence du choix des coefficients de pondération sur le gain d'adaptation afin d'accélérer la convergence.
Narendra [111] expose la version continue et discrète monovariable du MRAC, en utilisant différents "prototypes" d'erreurs et une condition sur la stabilité des zéros du système.

L'implémentation pratique du MRAC est exposée par Neuman [112] dans le cas monovariable. L'ajustement se fait par la méthode du gradient en utilisant, soit l'erreur de sortie (output error method), soit l'erreur d'entrée (input error method). L'application de la version discrète du MRAC, dans le domaine de la robotique, a été effectuée par Stoten [113] et Tarokh [114].

Pour utiliser, un modèle de référence réduit, l'algorithme **CGT** (Command Generator Tracker) a été introduit. Ritonja [115] applique un tel algorithme pour la commande d'un moteur à courant continu, dont la loi d'adaptation est non linéaire discontinue nommée **UVAL** (**U**nit **V**ector **A**daptation **L**aw).
La condition ASPR (Almost SPR) a été utilisée dans [116] pour assurer une convergence asymptotique de l'erreur.
Sobel [117] analyse la convergence du CGT implicite, en utilisant la fonction de Lyapounov, sans satisfaire les conditions du PMFC (Perfect Model Folowing Control).
Bartolini [118] calcule la commande en mettant en parallèle avec le système, un bloc du premier ordre et un filtrage de la sortie, de la commande et de la référence. L'algorithme d'adaptation est du type PI.
Liaw [119] montre la validité du MRAC, dans le cas où on calcule la commande en se basant sur un modèle réduit avec une technique de réduction "Power decomposition".

L'étude du MRAC, dans le cas stochastique, a été introduite par Goodwin [120]. Sean [121] présente un tel algorithme avec une estimation des paramètres du régulateur en utilisant le filtre de Kalman.
Tsakalis [122][123][124] montre l'applicabilité du MRAC aux systèmes linéaire variant dans le temps (Time Varying Systems). Balestrino [125] présente une comparaison entre VSS (Variable Structure System), AMFC et l'algorithme mixte regroupant VSS et AMFC. L'analyse de la stabilité a été faite en utilisant la fonction de Lyapounov.
Rahim [11] présente une comparaison entre MRAC et le control Floue(fuzzy logic), dont

la synthèse de la loi d'adaptation, se fait en utilisant la méthode du gradient et une extension de l'algorithme MRAC sous forme de MRIAC (Model Référence Input Adaptive Control) pour assurer la robustesse.

Dans la première partie de ce chapitre, on présente les systèmes adaptative à modèle de référence (MRAS). La commande linéaire par poursuite d'un modèle (LMFC) est développée dans la seconde partie. La troisième partie concerne la commande adaptative à modèle de référence. Le développement de l'algorithme MCS (Minimum Controller Synthesis) est présenté dans la quatrième partie. La version discrète du MRAC (DMRAC Discrète MRAC) est exposée dans la cinquième partie. Des résultats de simulation sont présentés dans la dernière partie. On terminera par une conclusion générale.

VI.1 Description mathématique du MRAS.

Il existe plusieurs types de systèmes adaptatifs à modèle de référence. On peut les classer suivant les critères suivants [98]:

VI.1.a Structure:

On distingue trois principales structures :
- MRAS parallèle.
- MRAS série.
- MRAS série-parallèle.

La structure parallèle MRAS (figure VI.1) est la structure la plus connue, nommée la méthode de l'erreur de sortie dans le cas de l'identification.

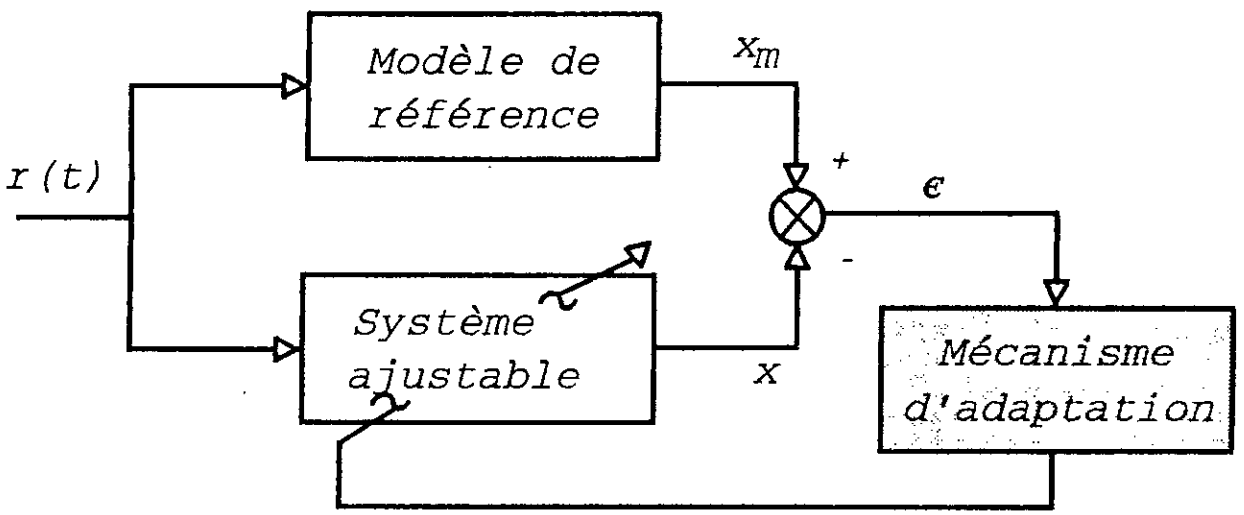


Figure V.1 Structure parallèle du MRAS.

VI.1.b Principe d'adaptation :

1- Par ajustement des paramètres du système ajustable (Parameter adaptation).
2- Par un signal de synthèse d'adaptation (Signal synthesis adaptation), une entrée auxiliaire est ajoutée aux système ajustable.
3- En combinant l'adaptation paramétrique et le signal de synthèse.

VI.1.c Conditions opératoires :

1- Avec un signal de test : ce signal est introduit à l'entrée du système ou bien à l'entrée du mécanisme d'adaptation.

2- Sans signal de test: l'adaptation se fait avec le signal existant déjà à l'entrée du système.

VI.1.d Indice de performance.

1- Minimisation de la norme de l'erreur généralisée et de ses dérivées.
2- Minimisation de la distance d'état.
3- Minimisation de la distance de structure.

VI.1.e Domaine d'application.

1- Système de commande adaptative par poursuite d'un modèle (AMFC System).
2- Identification avec un modèle ajustable.
3- Observation d'état (estimation).
4- Régulateur autoajustable (STR).

On présentera une Description du MRAS, qui va être utilisée tout le long de ce chapitre. Le caractère dual de l'algorithme lui permet l'extension directe dans le cas de la commande, observation et identification [88].

Avant de synthétiser les lois d'adaptation, certaines hypothèses de base doivent être posées [88]:

1- Le modèle de référence doit être un système linéaire invariant dans le temps.
2- Le modèle de référence et le système ajustable ont la même dimension.
3- Tous les paramètres du système ajustable sont accessibles pour l'adaptation (dans le cas de l'adaptation paramétriques).
4- Durant le processus d'adaptation, les paramètres du systèmes ajustable dépendent seulement du mécanisme d'adaptation.
5- Aucun signal, autre que le vecteur d'entrée, n'agit sur le système.
6- La différence initiale entre les paramètres du modèle et ceux du système est connue.
7- Le vecteur erreur d'état et de sortie sont mesurables.

Cet ensemble d'hypothèses constitue le cas idéal et permet un traitement analytique directe du MRAS. Mais dans les situations réelles, certaines de ces conditions ne sont pas toujours satisfaites. Ces conditions qui violent les hypothèses de base, se résument comme suit [88]:

1- Le modèle de référence est un système nonliéaire variable dans le temps.
2- Le système ajustable est un système nonlinéaire variable dans le temps.
3- Le modèle de référence et le système ajustable n'ont pas la même dimension.
4- Les paramètres du système ajustable ne sont pas tous accessibles pour l'adaptation.
5- Durant le processus d'adaptation, les paramètres du système ajustable ne dépendent pas seulement du mécanisme d'adaptation,mais ils sont aussi l'objet d'autres perturbations paramétriques.
6- Les perturbations sont appliquées à différentes parties du système.
7- La mesure du vecteur erreur est toujours affectée par un bruit.

Cet ensemble d'hypothèse est appelé le cas réel ou le cas général.

Dans ce qui suit, on développera l'MRAS dans le cas idéal pour la structure parallèle-parallèle, en utilisant la représentation d'état (figure VI.2).

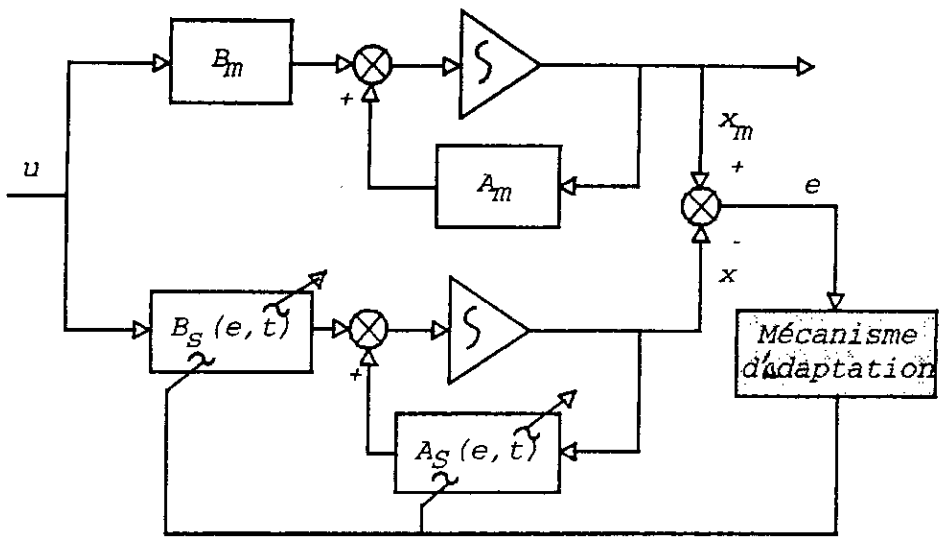


Figure VI.2 Représentation paralléle-parralléle du MRAS dans l'espace d'état.

Le modèle de référence choisi est sous la forme suivante [98]:

$$\dot{x}_m = A_m x_m + B_m u \qquad x_m(0) = x_{m_0} \tag{VI.1}$$

Où x_m: vecteur d'état de dimension n.
u: vecteur d'entrée de dimension m.
A_m et B_m: matrices de dimensions respectives nxn et nxm.

Le modèle de référence est choisi stable et complètement contrôlable [88].

Le système ajustable est choisi sous la forme d'un modèle linéaire variable dans le temps:

$$\dot{x} = A_s(e,t)\,x + B_s(e,t)\,u \qquad x(0) = x_0;\ A_s(0) = A_{s_0};\ B_s(0) = B_{s_0}. \tag{VI.2}$$

Où x: vecteur d'état de dimension n.
u: vecteur d'entrée de dimension m.
A_s et B_s: matrices variables dans le temps de dimensions respectives nxn et nxm.

$$e(t) = x_m(t) - x(t) \tag{VI.3}$$

e: vecteur d'état de l'erreur généralisée.
t: temps.

L'objectif de la synthèse du MRAS est de trouver une loi d'adaptation paramétrique qui permette d'ajuster les matrices A_s et B_s, de sorte que l'erreur $e(t)$ tende vers zéro quand t tend vers l'infini et pour toute entrée de référence et que:

$$\lim_{t\to\infty}(x_m - x) = 0 \tag{VI.4.a}$$

$$\lim_{t\to\infty} A_s(e,t) = A_m \tag{VI.4.b}$$

$$\lim_{t\to\infty} B_s(e,t) = B_m \qquad \text{(VI.4.c)}$$

Ceci doit être assuré quelque soit la différence initiale entre les paramètres du modèle de référence et ceux du système ajustable [88][98].
De plus, on veut que le mécanisme d'adaptation ait de la mémoire (c'est à dire qu'il mémorise les bonnes valeurs des paramètres une fois trouvés), ce qui conduit à l'introduction d'un intégrateur, dans le mécanisme d'adaptation. Ce dernier aura pour effet de rendre les paramètres du système ajustable, à l'instant t, dépendant non seulement de $e(t)$ mais aussi de l'histoire de $e(t)$ (c'est à dire $e(\tau)$, $\tau>t$).
La loi d'adaptation paramétrique sera alors [88][98]:

$$\begin{aligned} A_s(e,t) &= F(e,\tau,t)+A_s(0) \\ B_s(e,t) &= G(e,\tau,t)+B_s(0) \end{aligned} \qquad \text{(VI.5)}$$

Pour $0\le\tau\le t$.

F et G déterminent la relation fonctionnelle entre $A_s(e,t)$ et $B_s(e,t)$ et les valeurs de l'erreur e, dans l'intervalle $0\le\tau\le t$.

Remarque.
Dans le cas du signal d'adaptation de synthèse, la loi d'adaptation est [88]:

$$u_a(e,t) = u(e,\tau,t)+u_a(0) \qquad \text{(VI.6)}$$

Pour $0\le\tau\le t$.
Où $u(.)$ détermine la relation fonctionnelle entre $u_a(.)$ et les valeurs du vecteur de l'erreur dans l'intervalle $0\le\tau\le t$. □

En utilisant les équations (VI.1),(VI.2),(VI.3) et (VI.5), on peut établir l'équation différentielle qui caractérise la dynamique de l'erreur :

$$\begin{aligned} \dot{e} = \dot{x}_m-\dot{x} = A_m e+[A_m-A_s(0)-F(e,\tau,t)]x \\ +[B_m-B_s(0)-G(e,\tau,t)]u \end{aligned} \qquad \text{(VI.7)}$$

On obtient ainsi une équation décrivant le système adaptatif à modèle de référence. La représentation équivalente comme étant un système non linéaire variable dans le temps, est illustrée dans la figure VI.3.

L'analyse de la stabilité d'une telle structure, peut se faire en utilisant le théorie d'hyperstabilité de Popov [88][96]. Pour assurer la convergence asymptotique de l'erreur, d'une telle structure, la chaîne directe doit vérifier la condition de positivité réelle stricte, tandis que la chaîne de retour doit satisfaire l'inégalité de Popov [36][88].
Puisque la matrice A_m est prédéterminée par le choix du modèle de référence, la condition de positivité réelle ne peut être toujours vérifiée. Pour satisfaire une telle condition, on introduit dans la chaîne directe un compensateur linéaire D. Ce compensateur est choisi de telle sorte que le bloc linéaire soit SPR. On utilise alors, pour l'adaptation, non pas le vecteur erreur e, mais un vecteur V, défini par [36][88]:

$$V = De \qquad \text{(VI.8)}$$

Où D matrice constante de dimension nxn.

Avec cette nouvelle formulation, les matrices ajustables $A_s(e,t)$ et $B_s(e,t)$ peuvent être réécrites comme suit [88]:

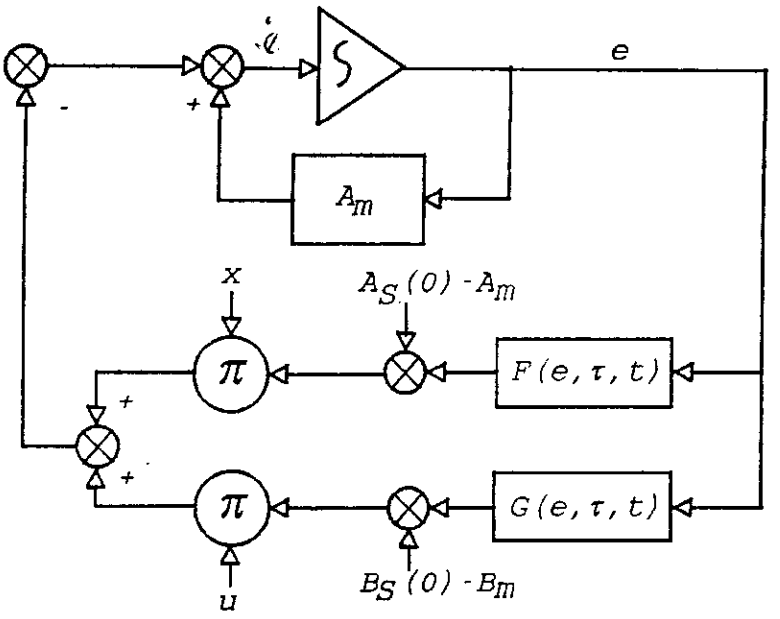


Figure VI.3 Représentation équivalente de l'erreur généralisée d'état.

$$A_s(e,t) = A_s(V,t) = \int_0^t \phi_1(V,t,\tau)\,d\tau + \phi_2(V,t) + A_s(0)$$
$$B_s(e,t) = B_s(V,t) = \int_0^t \psi_1(V,t,\tau)\,d\tau + \psi_2(V,t) + B_s(0) \qquad \text{(VI.9)}$$

Où les premiers termes assurent la mémorisation du mécanisme d'adaptation (action intégrale) et les seconds termes deviennent nulles quand $V=0$ ($e=0$, action proportionnelle).

$\phi_1(.)$, $\psi_1(.)$: matrices de dimensions respectives nxn et nxm, qui déterminent une relation non linéaire variable dans le temps entre $A_s(.), B_s(.)$ et les valeurs de v pour $0 \le \tau \le t$.

$\phi_2(.)$, $\psi_2(.)$: matrices de mêmes dimensions que ϕ_1 et ψ_1, , qui déterminent une relation non linéaire variable dans le temps entre $A_s(.), B_s(.)$ et les valeurs de v (elles sont nulles pour $V=0$).

Les relations (VI.8) et (VI.9) définissent la loi d'adaptation. La combinaison des équations (VI.7),(VI.8) et (VI.9), aboutit au système suivant :

$$\dot{e} = A_m e + w_1$$
$$V = De$$
$$w = -w_1 = \left[\int_0^t \phi_1(V,t,\tau)\,d\tau + \phi_2(V,t) + A_s(0) - A_m\right]x + \left[\int_0^t \psi_1(V,t,\tau)\,d\tau + \psi_2(V,t) + B_s(0) - B_m\right]u \qquad \text{(VI.10)}$$

A partir de ces équations, la figure VI.4 montre la représentation équivalente d'un

système adaptatif à modèle de référence paralléle-parralléle.

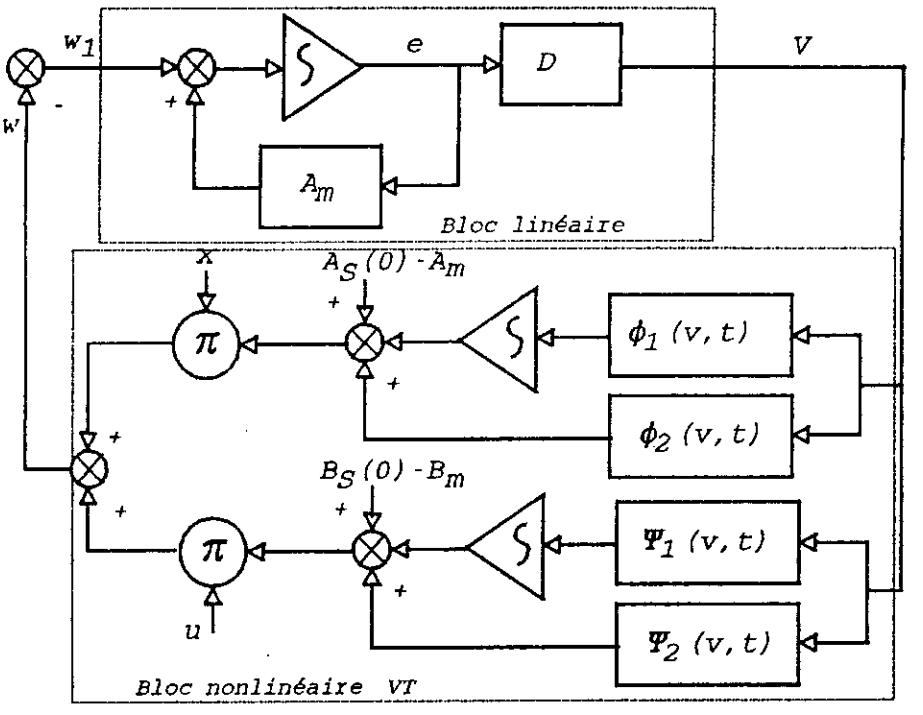


Figure VI.4 Représentation équivalente du MRAS parallèle parallèle.

La convergence asymptotique, de l'erreur d'état, est assurée dans le cas où le bloc linéaire est SPR c'est à dire:

Qu'il existe une matrice définie positive P et une matrice semi défini positive Q, tel que:

$$\begin{cases} PA_m + A_m^T P = -Q \\ P = D \end{cases} \qquad \text{(VI.11)}$$

Ainsi le compensateur D est déterminé, par résolution de l'équation de Lyapounov. Cette résolution peut se faire en utilisant l'algorithme de Jammeson [126], dans le cas où A_m est pleine. Par contre lorsqu'on choisit A_m comme matrice diagonale par bloc (de dimension deux), la résolution est facile. Ceci d'une part, d'autre part le bloc non linéaire doit satisfaire l'inégalité de Popov:

$$\eta(0, t_1) \geq -\gamma_0^2 \qquad \text{(VI.12)}$$

$$\eta(0, t_1) = \int_0^{t_1} V^T \left[\int_0^t \phi_1(V, t, \tau)\, d\tau + \phi_2(V, t) + A_0 \right] x \; dt$$

Où

$$+ \int_0^{t_1} V^T \left[\int_0^t \psi_1(V, t, \tau)\, d\tau + \psi_2(V, t) + B_0 \right] u \; dt$$

Et $A_0 = A_s(0) - A_m$; $B_0 = B_s(0) - B_m$.

Landau [88] propose plusieurs choix des matrices ϕ_1, ϕ_2, ψ_1 et ψ_2 qui satisfaient cette inégalité. Pour que l'inégalité soit satisfaite, il suffit que les deux termes de gauche satisfassent le même type d'inégalité. En faisant le choix du premier lemme (annexe D de la référence [88]), on a:

$$\phi_1(V,t,\tau) = F_A(t-\tau)\,V(\tau)\left[G_A x(\tau)\right]^T \qquad (VI.13.a)$$

$$\phi_2(V,t) = F_A'(t)\,V(t)\left[G_A' x(t)\right]^T \qquad (VI.13.b)$$

$$\psi_1(V,t,\tau) = F_B(t-\tau)\,V(\tau)\left[G_B u(\tau)\right]^T \qquad (VI.13.c)$$

$$\psi_2(V,t) = F_B'(t)\,V(t)\left[G_B' u(t)\right]^T \qquad (VI.13.d)$$

Pour $\tau \le t$.

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 $t \ge 0$.

Quelques choix particuliers.

Les matrices $\phi_1(.)$, $\phi_2(.)$, $\psi_1(.)$ *et* $\psi_2(.)$ peuvent être choisies sous la forme suivante:

$$F_A(t-\tau) = F_A > 0 \qquad (VI.14.a)$$

$$F_B(t-\tau) = F_B > 0 \qquad (VI.14.b)$$

$$F_A'(t) = F_A';\ F_B'(t) = F_B' \qquad (VI.14.c)$$

$$G_A'(t) = G_A';\ G_B'(t) = G_B' \qquad (VI.14.d)$$

On obtient alors une loi d'adaptation du type PI. D'autres lois existe, pour un choix particulier de ces matrices et aboutit à une loi du type "Integral+Relay adaptation" [88][89].
Ainsi on a assuré la convergence globale asymptotique de l'erreur décrite par les équations (VI.10) (théorème de landau [88]), et la représentation équivalente est illustré dans la figure VI.5.

Remarque.

La connexion qui existe entre l'approche utilisant la fonction de Lyapounov et celle utilisant la théorie d'hyperstabilité est montrée par Landau [98]. Puisque le système est asymptotiquement stable, donc il existe une fonction de Lyapounov qui nous permet d'aboutir à la même loi d'adaptation. Cette fonction est [98]:

$$V = e^T P e + 2\int_0^t V^T w\,dt + 2\gamma_0^2 \qquad (VI.15)$$

Où
$$\gamma_0^2 = \frac{1}{2}\,trace\left[A_m - A_s(0)\right]^T F_A^{-1}\left[A_m - A_s(0)\right] + \frac{1}{2}\,trace\left[B_m - B_s(0)\right]^T F_B^{-1}\left[B_m - B_s(0)\right]$$

Et P, F_A et F_B matrices définies positives.

La seule différence, c'est que ici, on aboutit à une loi d'adaptation, non pas en fonction de V mais en fonction de e [98].□

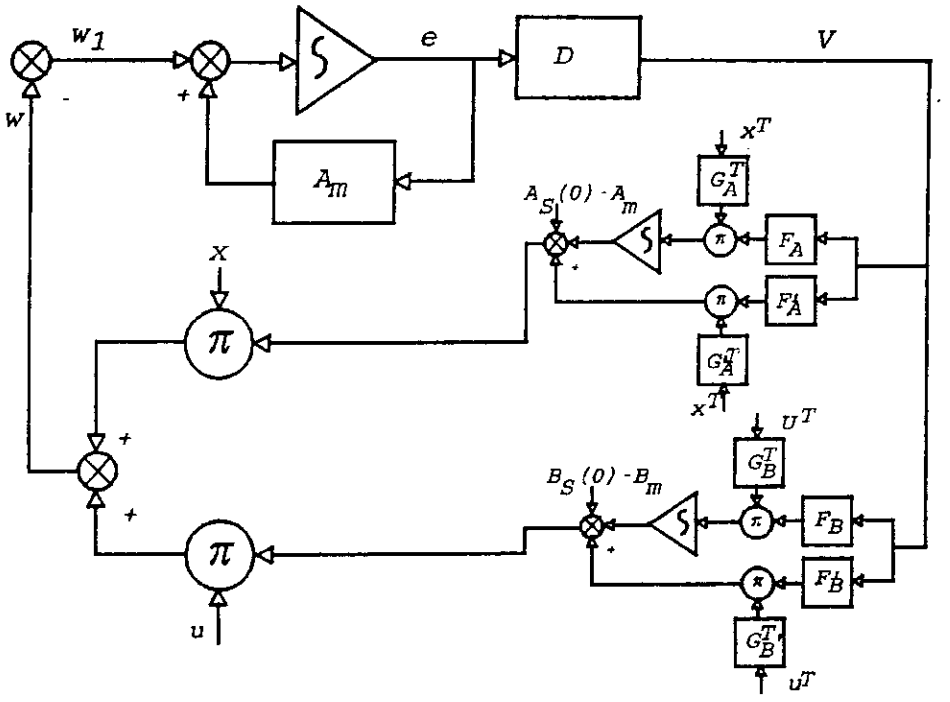


Figure VI.5 Représentation équivalente du MRAS avec adaptation PI.

VI.2 Commande linéaire par poursuite d'un modèle. (linear model following control).

Dans cette partie, on s'intéresse à développer un controlleur qui assure une poursuite parfaite d'un modèle de référence. La figure VI.6 illustre la structure d'un tel controlleur.

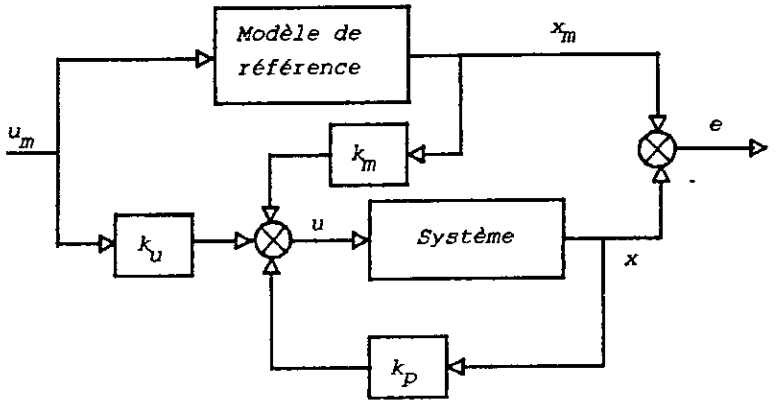


Figure VI.6 Commande linéaire par poursuite d'un modèle LMFC.

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 d'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:

$$\dot{x}_m = A_m x_m + B_m u_m \qquad \text{(VI.16)}$$

A_m: matrice d'Hurwitz [88].

Le système est défini par:

$$\dot{x} = A\,x + B\,u \tag{VI.17}$$

Et la commande est:

$$u = -k_p x + k_m x_m + k_u u_m \tag{VI.18}$$

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 :

$$e = x_m - x \tag{VI.19}$$

Pour assurer une poursuite parfaite, la dynamique de l'erreur doit être imposée. En utilisant les équations (VI.16), (VI. 17) et (VI.19), cette dynamique est définie par:

$$\dot{e} = (A_m - Bk_m)\,e + [A_m - A + B(k_p - k_m)]\,x + (B_m - Bk_u)\,u_m \tag{VI.20}$$

Ainsi il suffit d'imposer la condition suivante:

$$[A_m - A + B(k_p - k_m)]\,x + (B_m - Bk_u)\,u_m = 0 \tag{VI.21}$$

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:

$$\begin{aligned} B(k_p - k_m) &= A - A_m \\ Bk_u &= B_m \end{aligned} \tag{VI.22}$$

De cette façon, la dynamique de l'erreur est définie par la matrice $A_m - Bk_m$, qui doit être une matrice de Hurwitz. Des équations (VI.22) on aboutit à:

$$\begin{aligned} k_m - k_p &= B^+(A_m - A) \\ k_u &= B^+ B_m \end{aligned} \tag{VI.23}$$

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

Cette pseudo inverse existe dans le cas où:

$$rang[B] = rang[B\ A_m - A] = rang[B\ B_m] \tag{VI.24}$$

En introduisant les expressions de $k_m - k_p$ et de k_u dans (VI.22), on aboutit à:

$$\begin{aligned} (I - BB^+)(A_m - A) &= 0 \\ (I - BB^+)B_m &= 0 \end{aligned} \tag{VI.25}$$

Ces conditions sont connues sous le nom de conditions d' Erzberger [88]. Une fois ces

conditions sont satisfaites, la poursuite du modèle défini par (VI.17) à la référence x_m est assurée.

Algorithme de commande.

Données : - Mettre le système linéaire ou non linéaire sous la forme (VI.17) (dans le cas **non linéaire**, rejeter les **non linéarités aux** perturbations, sous la forme $\dot{x}=Ax+Bu+d$, où d: le terme englobant les nonlinéarités)
- Choisir le modèle de référence (VI.16).
- Calcul de k_p, k_u *et* k_m à partir de (VI.23).
- Vérifier les conditions (VI.25).
Si les conditions sont vérifiées alors LMFC est applicable.
si non l'algorithme n'est pas applicable ou bien changer le modèle de référence.

Etape 1: Générer la référence x_m à partir de (VI.16).

Etape 2: Calcul de la commande de (VI.18).

t=t+1 revenir à étape 1.

VI.3 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'implémantation d'une telle loi, peut se faire de deux manière [88]:

1- Adaptation paramétrique (figure VI.7).
2- Adaptation par signal de synthèse (figure VI.8).

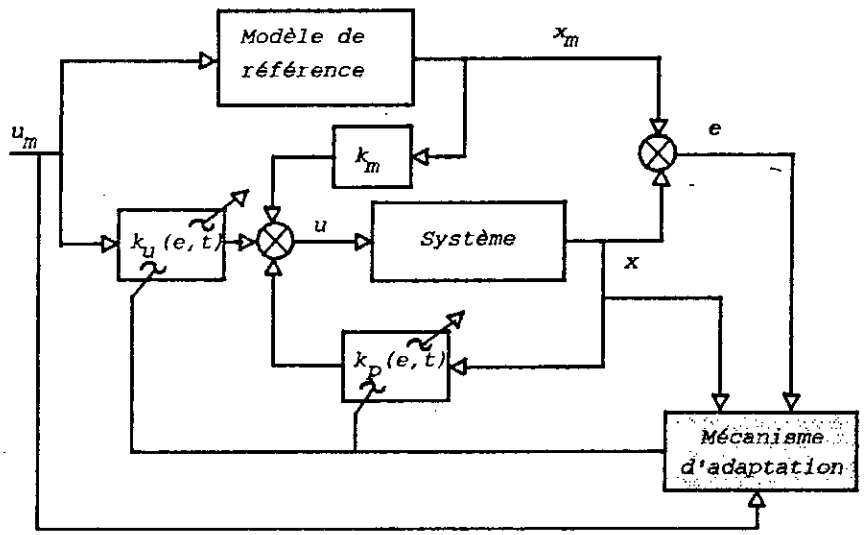


Figure VI.7 AMFC parallèle avec l'adaptation paramétrique.

Dans le cas de l'adaptation paramétrique, l'entrée du système est exprimée par [88]:

$$u = -k_p(e,t)\,x+k_m x_m+k_u(e,\tau)\,u_m \qquad \text{(VI.26)}$$

Où $k_p(.)$ *et* $k_u(.)$: matrices variables dans le temps dépendant de e.
k_m: matrice constante.

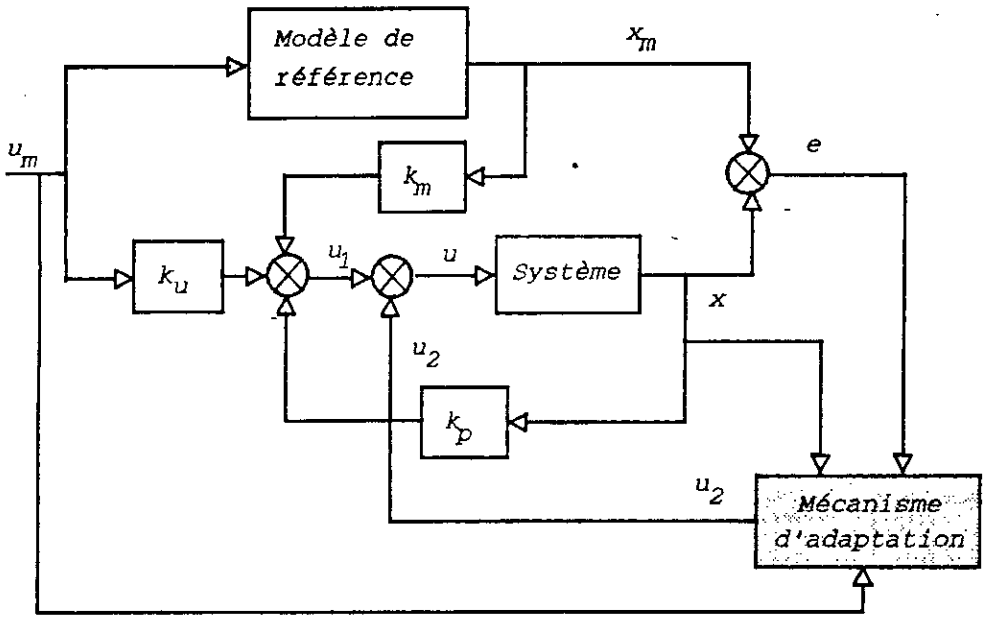


Figure VI.8 AMFC avec l'adaptation par signal de synthèse.

$k_p(.)$ *et* $k_u(.)$: peuvent être exprimées par:

$$\begin{aligned} k_p(e,t) &= k_p - \delta k_p(e,t) \\ k_u(e,t) &= k_u + \delta k_u(e,t) \end{aligned} \qquad \text{(VI.27)}$$

Où k_p *et* k_u sont des matrices constantes, déterminées à partir du LMFC.

Avec cette décomposition, on peut écrire:

$$u = u_1 + u_2 \qquad \text{(VI.28)}$$

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.

Pour synthétiser la loi d'adaptation, il suffit de transformer la structure de la figure VI.8, sous forme de celle de la figure VI.2. En utilisant les équations (VI.27) et (VI.26) dans (VI.17), on trouve:

$$\begin{aligned} \dot{x} = {} & [A - Bk_p + Bk_m + B\,\delta k_p(e,t)]\,x \\ & + B[k_u + \delta k_u(e,t)]\,u_m + Bk_m e \end{aligned} \qquad \text{(VI.29)}$$

Donc [88]:

$$\begin{aligned} A_s(V,t) &= A - Bk_p + Bk_m + B\,\delta k_p(e,t) \\ B_s(e,t) &= B[k_u + \delta k_u(e,t)] \end{aligned} \qquad \text{(VI.30)}$$

En posant $Bk_m = k$, l'équation (VI.29) devient:

$$\dot{x} = A_s(V,t)x + B_s(V,t)\,u_m + ke \qquad \text{(VI.31)}$$

Où

$$V = De \tag{VI.32}$$

l'équation de l'erreur devient alors [88]:

$$\dot{e} = (A_m - k)\, e + I w_1 \tag{VI.33}$$

Où w_1 est définie par (VI.10).

La stabilité asymptotique du système défini par (VI.31),(VI.32),(VI.33) et (VI.10) est assurée si la fonction de transfert [88]:

$$H(s) = D(sI - A_m + k)^{-1} \tag{VI.34}$$

est SPR, et le bloc non linéaire assurant l'inégalité de Popov.

En faisant le choix suivant [88]:

$$\begin{aligned} A_s(V,t) &= B\left[\int_0^t \phi_1(V,t,\tau)\, d\tau + \phi_2(V,t)\right] + A_s(0) \\ B_s(V,t) &= B\left[\int_0^t \psi_1(V,t,\tau)\, d\tau + \psi_2(V,t)\right] + B_s(0) \end{aligned} \tag{VI.35}$$

Où $A_s(0) = A + B[k_m - k_p + \delta k_p(0)]$; $B_s = B[k_u + \delta k_u(0)]$.

En comparant les équations (VI.35) et (VI.30), on trouve:

$$\begin{aligned} \delta k_p(e,t) &= \delta k_p(V,t) = \int_0^t \phi_1(V,t,\tau)\, d\tau + \phi_2(V,t) + \delta k_p(0) \\ \delta k_u(e,t) &= \delta k_u(V,t) = \int_0^t \psi_1(V,t,\tau)\, d\tau + \psi_2(V,t) + \delta k_u(0) \end{aligned} \tag{VI.36}$$

Où ϕ_1, ϕ_2, ψ_1, ψ_2 sont définies par (VI.13).

Avec cette loi d'adaptation, qu'elle est la valeur du compensateur D, assurant la positivité réelle stricte? Pour ce faire, nous allons établir les équations dynamiques de l'erreur. Des équations (VI.29) et (VI.16), on obtient :

$$\begin{aligned} \dot{e} = (A_m - Bk_m)\, e + [A_m - A + B(k_p - k_m) - B\,\delta k_p(V,t)]\, x \\ + [B_m - Bk_u - B\,\delta k_u(V,t)]\, u_m \end{aligned} \tag{VI.37}$$

En supposant que la solution, de la poursuite parfaite existe, les équations (VI.22) sont donc vérifiées c'est à dire:

$$\begin{aligned} B(k_p^0 - k_m) &= A - A_m \\ Bk_u^0 &= B_m \end{aligned} \tag{VI.38}$$

Avec k_p^0 et k_u^0 sont inconnues.

Les équations (VI.38),(VI.37) et (VI.32) définissent la dynamique de l'erreur décrite par les équations suivantes:

$$\begin{aligned} \dot{e} &= (A_m - Bk_m)\, e + B\, w_1 \\ V &= De \\ w = -w_1 &= [\delta k_p(V,t) + k_p^0 - k_p]\, x + [\delta k_u(V,t) - k_u^0 + k_u]\, u_m \\ &= \left[\int_0^t \phi_1(V,t,\tau)\, d\tau + \phi_2(V,t) + \delta k_p^0\right] x \\ &+ \left[\int_0^t \psi_1(V,t,\tau)\, d\tau + \psi_2(V,t) + \delta k_u^0\right] u_m \end{aligned} \qquad \text{(VI.39)}$$

Où $\delta k_p^0 = \delta k_p(0) - k_p + k_p^0$; $\quad \delta k_u^0 = \delta k_u(0) + k_u - k_u^0$.

Le schéma équivalent d'une telle structure est illustrée dans la figure VI.9.

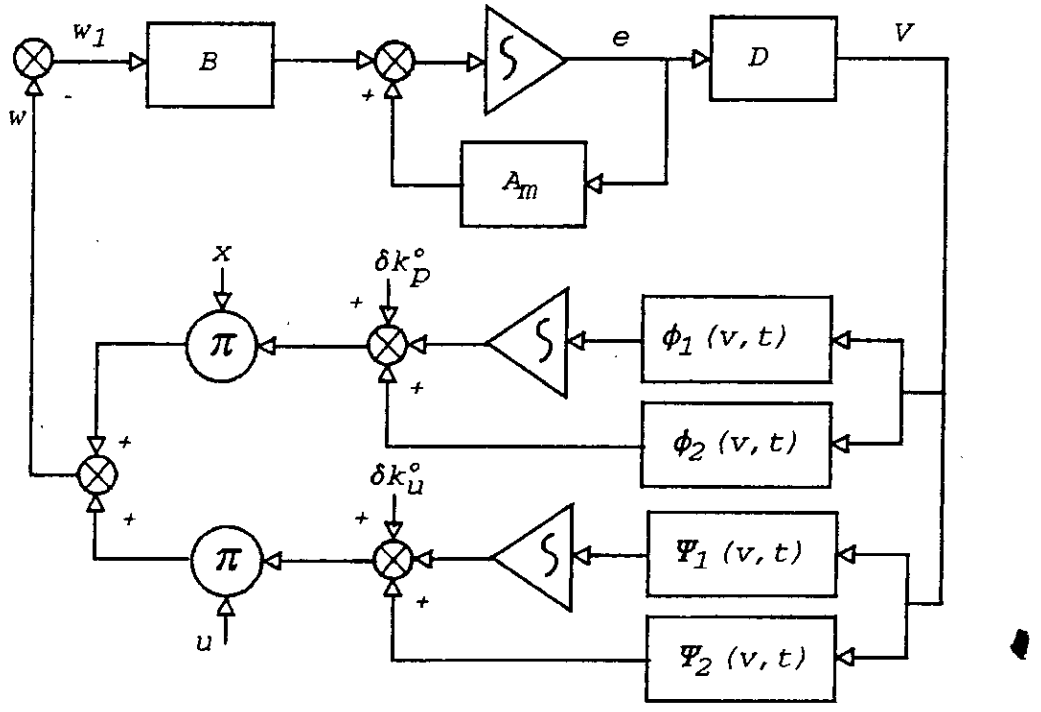


Figure VI.9 Schéma équivalent de l'évolution de l'erreur dans le cas du AMFC parallèle parallèle.

Ainsi la matrice peut être déterminée, en utilisant l'équation (VI.38) et le théorème de Landau[88], on obtient:

$$D = B^T P \qquad \text{(VI.39.a)}$$

En conclusion, la stabilité asymptotique de l'erreur est assurée. La loi d'adaptation est définie par (VI.36) et l'ajustement se fait par le signal de synthèse.

Algorithme de commande.

Données : - Mettre le système linéaire ou non linéaire sous la forme (VI.17) (dans le cas nonlinéaire, rejeter les nonlinéarité aux perturbations, sous la forme $\dot{x}=Ax+Bu+d$, où d: le terme englobant les non linéarités).
- Choisir le modèle de référence (VI.16).
- Calcul de k_p, k_u et k_m à partir de (VI.23).
- Vérifier les conditions (VI.25).
Si les conditions sont vérifiées alors LMFC est applicable. si non l'algorithme n'est pas applicable ou bien changer le modèle de référence.
- Choisir P et Q et calculer D de (VI.39.a).
- Choisir les matrices définies par (VI.14).
- Choisir $\delta k_p(0)$ et $\delta k_u(0)$.

Etape 1: Générer la référence x_m à partir de (VI.16).
Etape 2: Calcul de la commande linéaire U_1 dans (VI.28).
Etape 3: Adaptation des paramètres δk_p et δk_u en utilisant (VI.36). Calcul du signal de synthèse u_2 dans (VI.28).
Etape 4: Calcul de la commande globale u dans (VI.28).
t=t+1 revenir à étape 1.

VI.3.1 Résultats de simulation.

Dans le cas du LMFC, le modèle du robot est :

$$\dot{X}=A\,X+B\,U+d.$$

Où $A=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{f1}{m_1+m_3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{f_2}{j'} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{f_3}{m_3} \end{bmatrix}$; $B=\begin{bmatrix} 0 & 0 & 0 \\ \frac{k1}{m_1+m_3} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{k_2}{j'} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{k_3}{m_3} \end{bmatrix}$;

$d^T=\left[0 \quad -g \quad 0 \quad -2m_3(x_5-\frac{l_2}{2})x_4\frac{x_6}{j'}-\frac{m_3}{J'}(x_5^2-l_2x_5)\dot{x}_4 \quad 0 \quad (x_5-\frac{l_2}{2})x_4^2\right]$; $J'=m_3\frac{l_2^2}{3}$.

En utilisant (VI.23), on peut calculer k_p et k_u, en imposant $k_m=0$, avec un choix de: $A_m=\begin{bmatrix} 0 & 1 \\ -a_1 & -a_2 \end{bmatrix}$; $B_m=diag\begin{bmatrix} 0 \\ a_1 \end{bmatrix}$. Où $a_1=\lambda_1\lambda_2$ et $a_2=-(\lambda_1+\lambda_2)$.
Avec λ_1 et λ_2: pôles désirés.

Les conditions d'Erzberger (VI.25) sont vérifiées pour notre cas. Pour tester cette validité par simulation, on a imposé d=0, les résultats de simulation dans de tels conditions sont consignés sur les figures VI.LMFC.1.a et b, pour une entrée unitaire, avec $\lambda_1=\lambda_2=-4$. On remarque que la poursuite est parfaite.
Dans le cas où on applique la commande LMFC au robot ($d\neq0$), les figures VI.LMFC.2.a et b illustrent les résultats de simulation. Dans le cas où on impose un modèle de référence rapide (dix fois le précédent $\lambda_1=\lambda_2=-40$).

Le système nécessite un effort de commande important, ce qui est illustré dans les figures VI.LMFC.3.a et b. Pour rester dans le cas pratique, on a limité la commande à 20. Les performances se détériorent et les figures VI.LMFC.4.a et b montrent bien ce résultats.

Dans le cas où on impose une dynamique réalisable $(\lambda_1=\lambda_2=-4)$ et une limitation sur la commande, l'utilisation de la commande adaptative s'avère nécessaire. La résolution de l'équation de Lyapounov permet la détermination du compensateur D. Le modèle de référence est défini comme précédemment [40]:

$$(\lambda_1=\lambda_2=-4) \text{ et } Q = diag\begin{bmatrix}9 & 0\\ 0 & 1\end{bmatrix}.$$

La résolution de l'équation de Lyapounov est simple, car A_m et Q sont choisies diagonales par bloc.
Les matrices de la loi d'adaptation sont [4]:

$$G_A=G_A'=I_{6x6};\ G_B=G_B'=I_{3x3}.\quad F_A=F_B'=\alpha I_{3x3};\ F_A'=F_B'=\beta I_{3x3}.$$

Où $\alpha>0$ et $\beta>0$: constantes réelles.
En faisant un choix de $\alpha=100$ et $\beta=1$, avec les conditions initiales nulles $(\delta k_p(0)=\delta k_u(0)=0)$. les résultats de simulation de AMFC sont consignés sur les figures VI.AMFC.1.a et b, pour une entrée unitaire. Avec les même pondération, en imposant une entrée polynomiale (polynômes du troisième degré), les figures VI.AMFC.2.a et b montrent les résultats de simulation.

Pour tester la robustesse de l'algorithme, on lui impose une variations paramétriques de 500% à $t=1.5s$. La simulation d'une telle variation est consignée sur les figures VI.AMFC.3.a et b.

Enfin, pour tester la capacité de poursuite, l'imposition d'une consigne la fenêtre de VIVIANI, s'avère adéquate, dont les résultats sont sur les figures VI.AMFC.4.a,b,c.

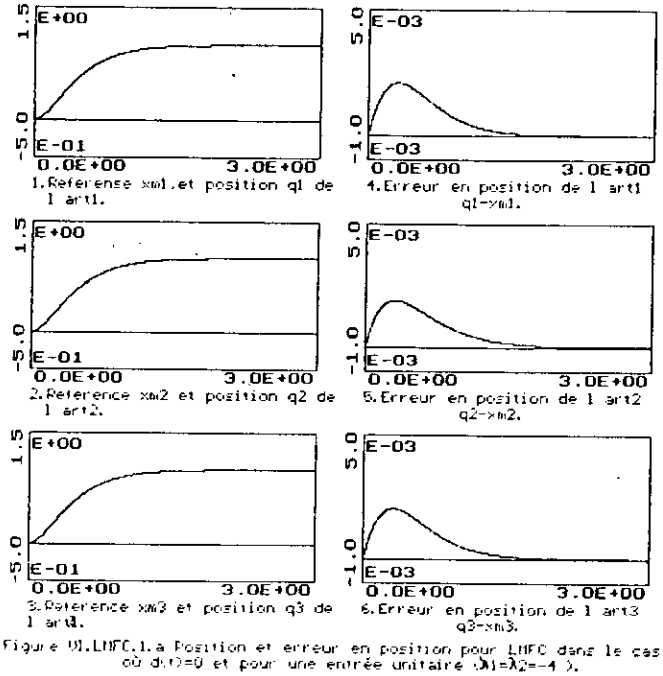


Figure VI.LMFC.1.a Position et erreur en position pour LMFC dans le cas où d(t)=0 et pour une entrée unitaire ($\lambda_1=\lambda_2=-4$).

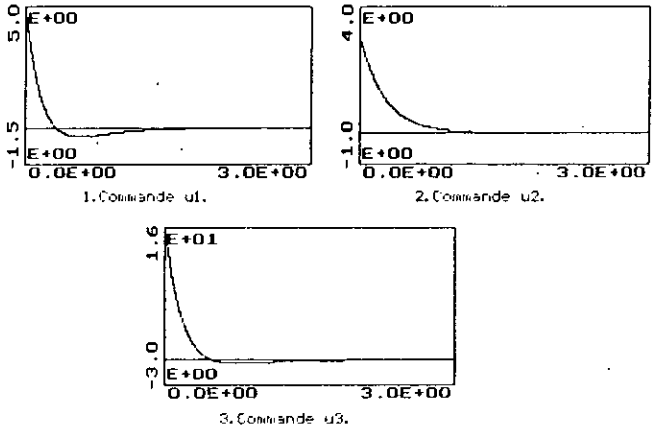
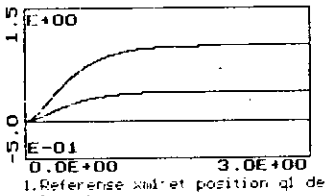
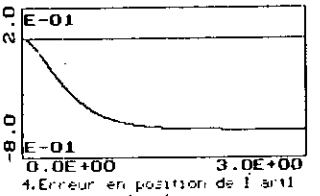


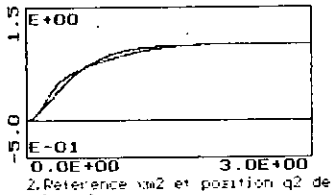
Figure VI.LMFC.1.b Presentation des commandes.



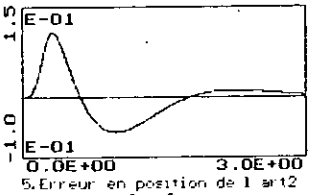
1.Reference xm1 et position q1 de l art1.



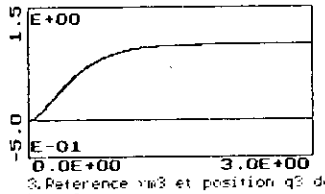
4.Erreur en position de l art1 q1-xm1.



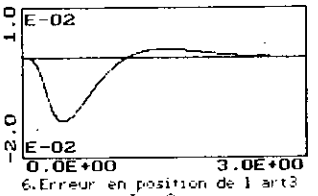
2.Reference xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

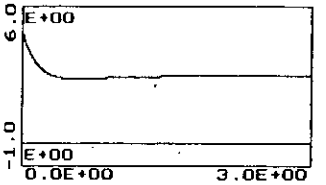


3.Reference xm3 et position q3 de l art3.

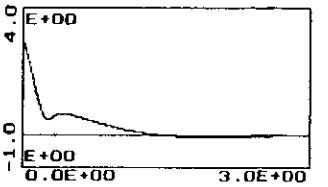


6.Erreur en position de l art3 q3-xm3.

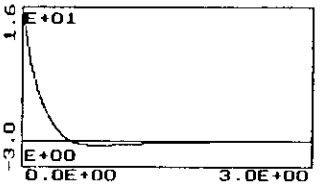
Figure VI.LMFC.2.a Position et erreur en position pour LMFC. (entrée unitaire; d(t)≠0; λ1=λ2=-4).



1.Commande u1.

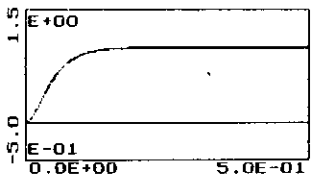


2.Commande u2.

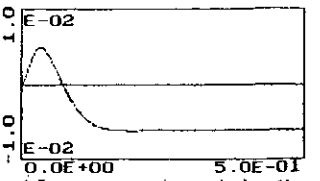


3.Commande u3.

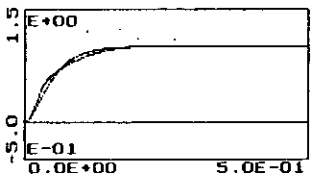
Figure VI.LMFC.2.b Presentation des commandes.



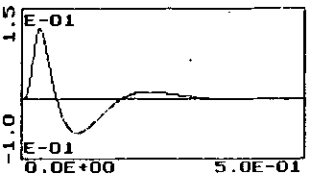
1.Reference xm1 et position q1 de l art1.



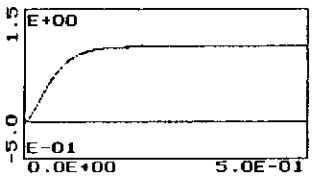
4.Erreur en position de l art1 q1-xm1.



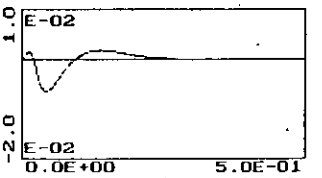
2.Reference xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

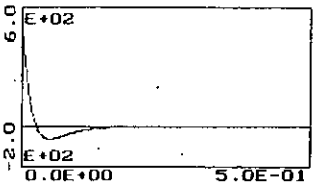


3.Reference xm3 et position q3 de l art3.

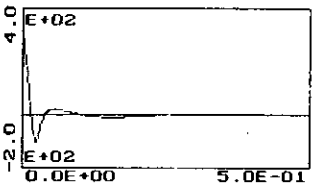


6.Erreur en position de l art3 q3-xm3.

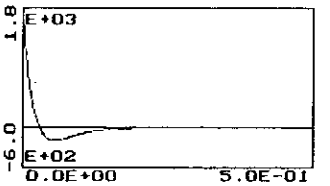
Figure VI.LMFC.3.a Position et erreur en position pour LMFC. (entrée unitaire; d(t)=0; λ1= λ2=-40).



1.Commande u1.

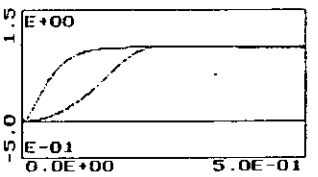


2.Commande u2.

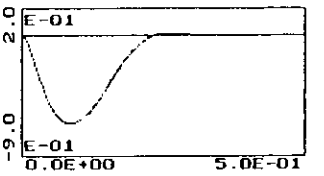


3.Commande u3.

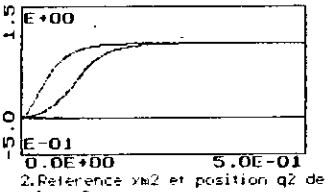
Figure VI.LMFC.3.b Presentation des commandes.



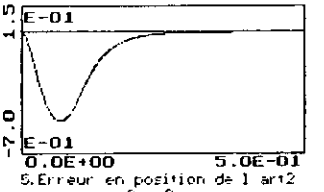
1.Reference xm1 et position q1 de l art1.



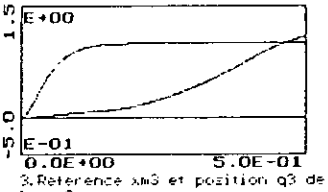
4.Erreur en position de l art1 q1-xm1.



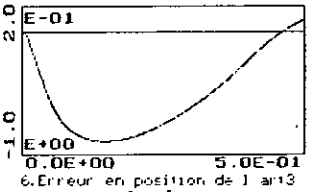
2.Reference xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

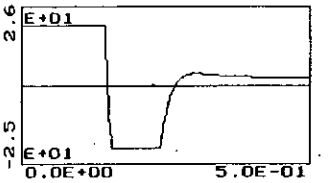


3.Reference xm3 et position q3 de l art3.

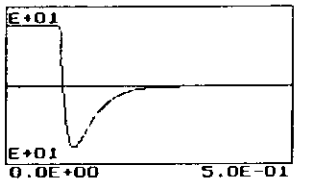


6.Erreur en position de l art3 q3-xm3.

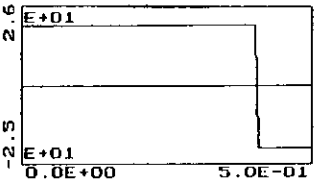
Figure VI.LMFC.4.a Position et erreur en position pour LMFC. (entrée unitaire; limitation de la commande; d(t)≠0; λ1=λ2=-40).



1.Commande u1.



2.Commande u2.



3.Commande u3.

Figure VI.LMFC.4.b Presentation des commandes.

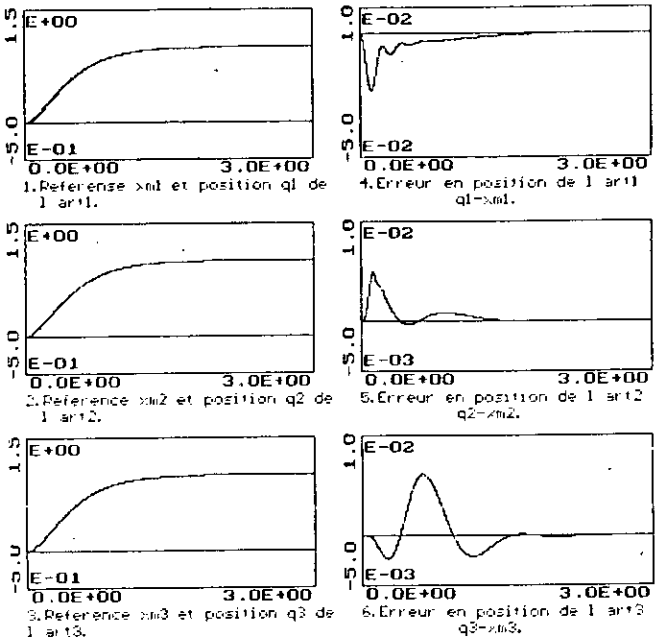


Figure VI.AMFC.1.a Position et erreur en position pour AMFC. (entrée unitaire)

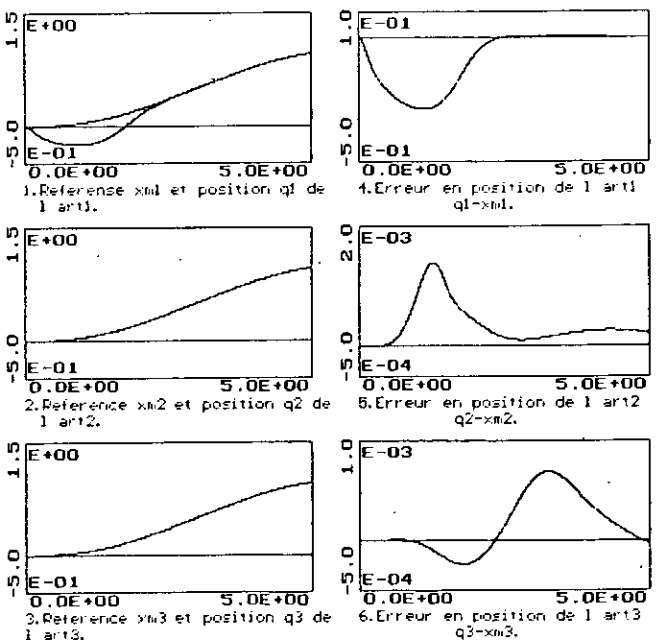


Figure VI.AMFC.2.a Position et erreur en position pour AMFC. (entrée polynomiale)

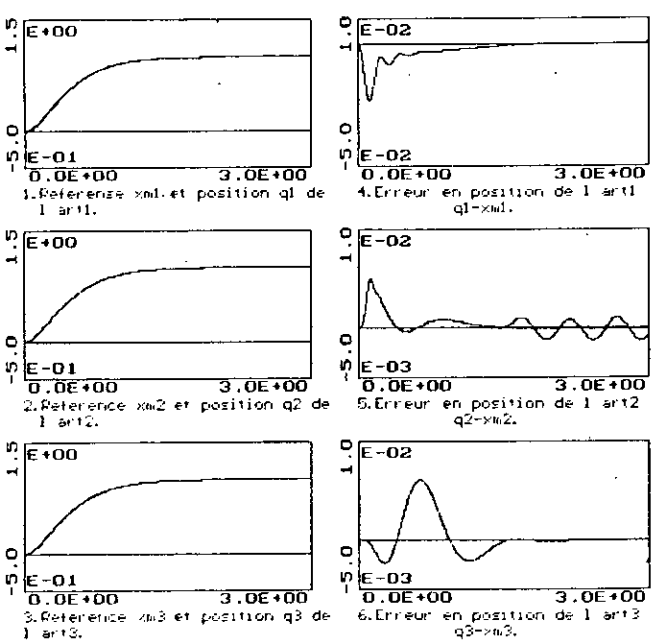


Figure VI.AMFC.3.a Position et erreur en position pour AMFC. (entrée unitaire; variation de 500% du modèle du robot à t=1.5s.)

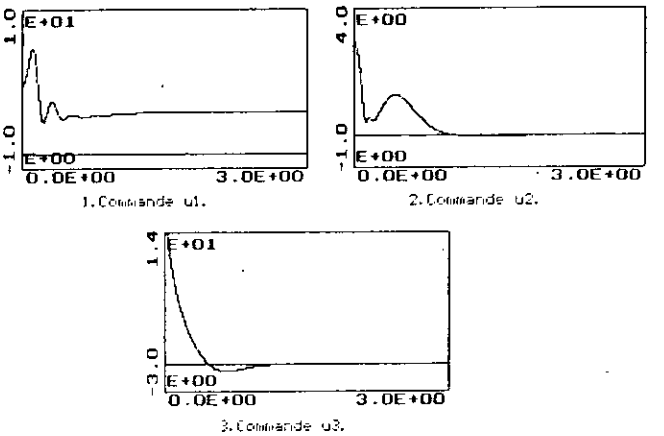


Figure VI.AMFC.1.b Presentation des commandes.

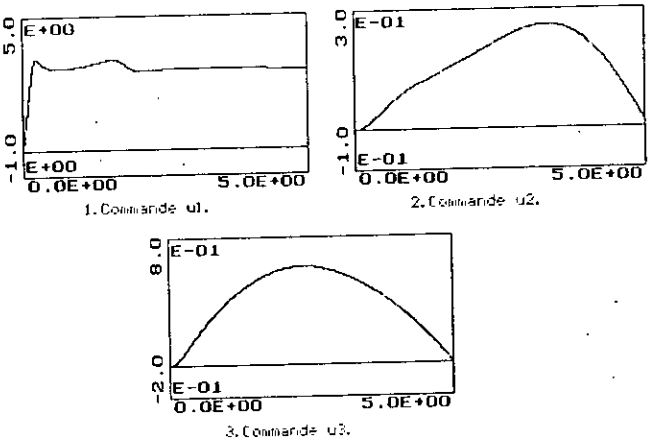


Figure VI.AMFC.2.b Presentation des commandes.

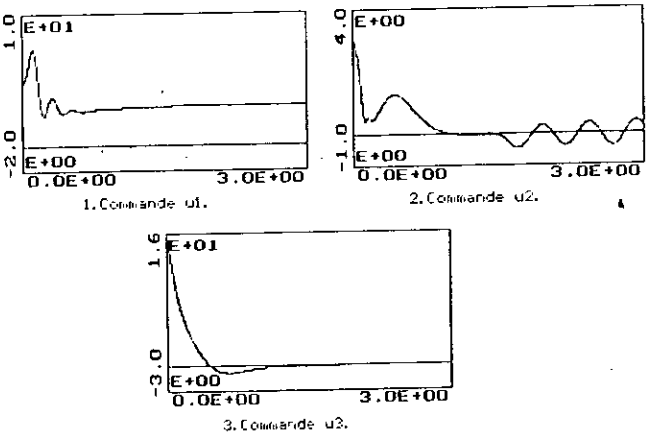
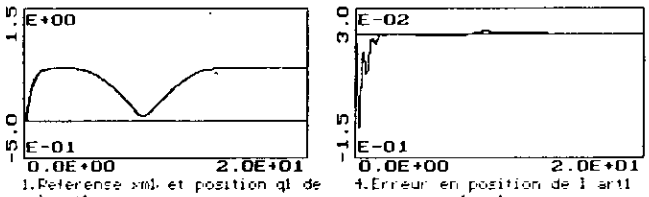
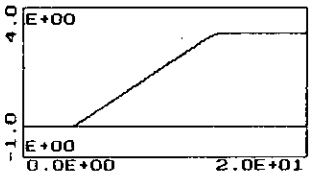


Figure VI.AMFC.3.b Presentation des commandes.



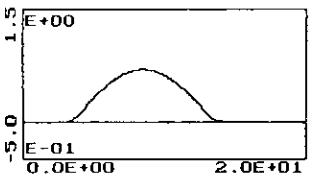
1.Reference xm1 et position q1 de l art1.

4.Erreur en position de l art1 q1-xm1.

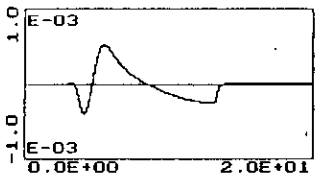


2.Reference xm2 et position q2 de l art2.

5.Erreur en position de l art2 q2-xm2.

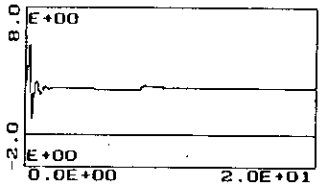


3.Reference xm3 et position q3 de l art3.

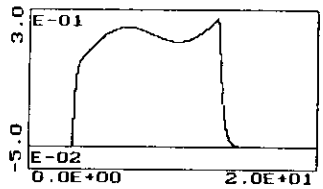


6.Erreur en position de l art3 q3-xm3.

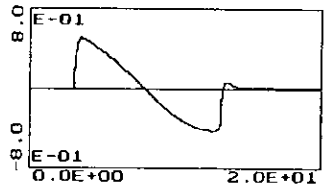
Figure VI.AMFC.4.a Position et erreur en position pour AMFC. consigne fenetre de HAMMING



1.Commande u1.

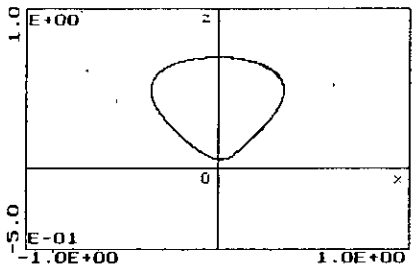


2.Commande u2.

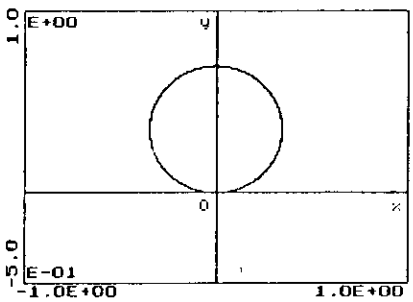


3.Commande u3.

Figure VI.AMFC.4.b Presentation des commandes.



1.Projection sur le plan xoz.



2.Projection sur le plan xoy.

Figure VI.AMFC.4.c Projection des trajectoires désirées et réelles sur les plan principaux.

VI.4 Synthèse d'un contrôleur minimal MCS. (Minimum controller synthesis).

La synthèse de la commande adaptative à modèle de référence MRAC, nécessite la 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 [108]. La structure d'un tel contrôleur est illustrée dans la figure VI.10.

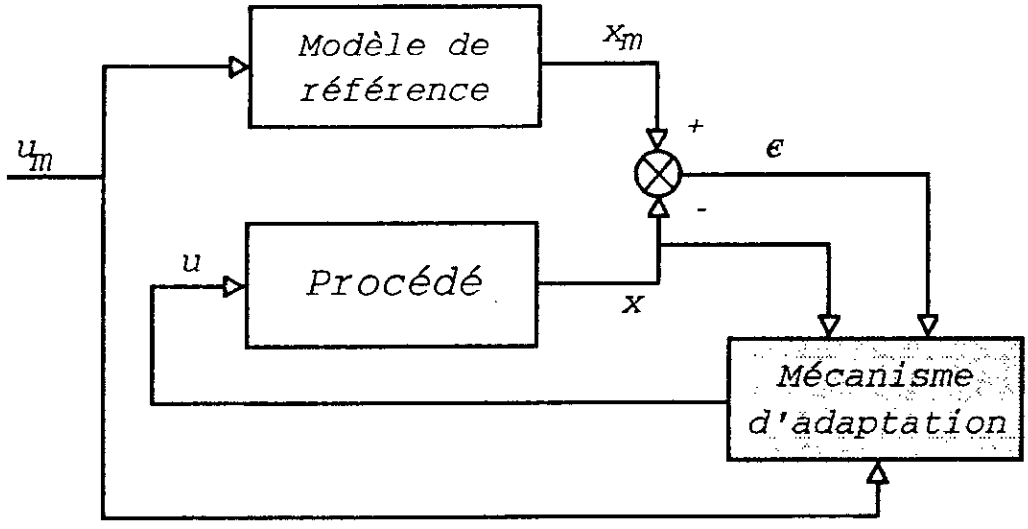


Figure VI.10 Structure de base du controlleur MCS.

Ce contrôleur peut être déduit directement du MRAC, par imposition de l'hypothèse suivante [108]

$$k_m = k_p = k_u = 0 \qquad \text{(VI.40)}$$

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:

$$\dot{x}_m = A_m x_m + B_m u_m \qquad \text{(VI.41)}$$

A_m: matrice d'Hurwitz [88].

Le système est défini par:

$$\dot{x} = A\,x + B\,u \qquad \text{(VI.42)}$$

La commande définie par (VI.26) et (VI.27) devient [108]:

$$u = \delta k_p(e,t)\,x + \delta k_u(e,\tau)\,u_m \qquad \text{(VI.43)}$$

La dynamique de l'erreur est alors définie par :

$$\dot{e} = A_m e + [A_m - A - B\,\delta k_p(V,t)]\,x + [B_m - B\,\delta k_u(V,t)]\,u_m \qquad \text{(VI.44)}$$

Avec $e = x_m - x$.

Le système ajustable est:

$$\dot{x} = [A+B\,\delta k_p(e,t)]\,x+B\,\delta k_u(e,t)\,u_m \qquad \text{(VI.45)}$$

donc $\begin{cases} A_s(V,t) = A+B\,\delta k_p(e,t) \\ B_s(e,t) = B\,\delta k_u(e,t) \end{cases}$

La loi d'adaptation est la même que celle définie par (VI.35) avec:

$$\begin{matrix} A_s(0) = A+B\,\delta k_p(0) \\ B_s(0) = B\,\delta k_u(0) \end{matrix} \qquad \text{(VI.46)}$$

On déduit alors:

$$\begin{aligned} \delta k_p(e,t) = \delta k_p(V,t) &= \int_0^t \phi_1(V,t,\tau)\,d\tau + \phi_2(V,t)+\delta k_p(0) \\ \delta k_u(e,t) = \delta k_u(V,t) &= \int_0^t \psi_1(V,t,\tau)\,d\tau + \psi_2(V,t)+\delta k_u(0) \end{aligned} \qquad \text{(VI.47)}$$

Où $V=De$.

Remarque.

Stoten & Benchoubane [108] proposent une loi d'adaptation sous la forme suivante:

$$\begin{aligned} \delta k_p(V',t) &= \int_0^t \alpha V'(\tau)\,x^T(\tau)\,d\tau + \beta V'(t)\,x^T(t) \\ \delta k_u(V',t) &= \int_0^t \alpha V'(\tau)\,U_m^T(\tau)\,d\tau + \beta V'(t)\,U_m^T(t) \end{aligned} \qquad \text{(VI.48)}$$

Où $V'=B^TPe$ et $\delta k_p(0)=\delta k_u(0)=0$.

α, β: des scalaires tel que $\alpha>0$; $\beta\geq 0$.

Le schéma équivalent de l'évolution de l'erreur est défini sur la figure VI.11. La détermination de D est déduite du fait que la chaîne directe est SPR et ceci par résolution de l'équation de Lyapounov:

$$\begin{cases} PA_m+A_m^TP = -Q \\ D = P \end{cases} \qquad \text{(VI.50)}$$

Le bloc de retour non linéaire doit satisfaire l'inégalité de Popov, c'est à dire:

$$\int_0^{t_1} V^Tw_1\,dt \geq -\gamma_0^2 \qquad \text{(VI.51)}$$

Avec $w_1=[B\,\delta k_p(e,t)-A_m+A]\,x+B[\delta k_u(e,t)-B_m]\,U_m$.

où $\delta k_p(.)$ et $\delta k_u(.)$ sont définies par les équations (VI.47) et (VI.13).

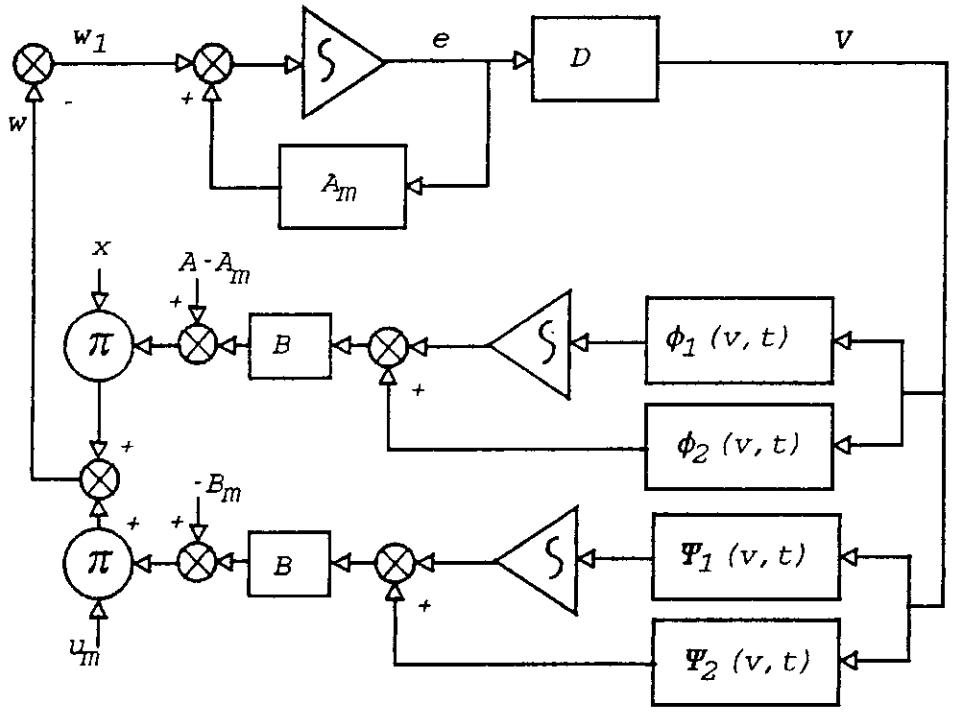


Figure VI.11 Schéma équivalent de l'évolution de l'erreur dans le cas du MCS.

En introduisant l'expression de w_1 dans (VI.51), on obtient:

$$\int_0^{t_1} V^T w_1 dt = \int_0^{t_1} V^T \left[B \int_0^t F_A V(\tau) \left[G_A x(\tau) \right]^T d\tau + F_a' V(t) \left[G_A' x(t) \right]^T \right] x \, dt + \int_0^{t_1} V^T \left[B \int_0^t F_B V(\tau) \left[G_B u_m(\tau) \right]^T d\tau + F_B' V(t) \left[G_B' u_m(t) \right]^T - B_m \right] u_m \, dt \qquad \text{(VI.52)}$$

Ainsi on voit l'effet de B sur l'inégalité de Popov. Pour que l'inégalité soit satisfaite, il suffit que le signe de B soit connu, pour assuré la positivité des matrices suivante (condition suffisante) [88]:

$$BF_A > 0;\ BF_A' \geq 0 \ et \ BF_B > 0;\ BF_B' \geq 0 \qquad \text{(VI.52.a)}$$

Ainsi le MCS nécessite la connaissance du degré du système ainsi que le signe des éléments de la matrice B, ou bien vérifier la positivité de cette dernière.

Algorithme de commande.

Données : - Degré du système et signe de B.
- Choisir le modèle de référence (VI.16).
- Choisir P et Q et calculer D de (VI.50).
- Choisir les matrices définies par (VI.14).
- Choisir $\delta k_p(0)$ et $\delta k_u(0)$.

Etape 1: Générer la référence x_m à partir de (VI.41).

Etape 2: Adaptation des paramètres δk_p et δk_u en utilisant (VI.49). Calcul du signal de synthèse u_2 dans (VI.28).

Etape 3: Calcul de la commande à partir (VI.43).

t=t+1 revenir à étape 1.

VI.4.1 Résultats de simulation.

Dans cette partie, la mise sous forme d'état linéaire du modèle du robot n'est pas nécessaire.

Avant d'utiliser l'algorithme MCS développé, on a considéré l'algorithme AMFC, dans lequel, on impose $k_p=k_u=0$, avec $V=B^T Pe$, ce qui aboutit à un algorithme similaire à celui de Stoten [108]. Avec les mêmes caractéristiques que dans la partie précédente, on a relevé les différents caractéristiques de l'algorithme pour différentes valeurs de β ($\beta=1$ *et* $\beta=60$), dans le cas de l'entrée unitaire, figures VI.MCS.1 et VI.MCS.2. Dans le cas d'une entrée, la fenêtre de VIVIANI, les figures VI.MCS.3 et VI.MCS.4 montrent les résultats de simulation.

Pour le test de l'algorithme du MCS ($V=Pe$), les pondérations sont [109]:

$$G_A=G_A'=I_{6x6};\quad G_B=G_B'=I_{3x3};$$

$$F_A=F_B=diag[0\ \alpha_i]=diag[0\ 100];\quad dim(F_A)=3x6;$$

$$F_A'=F_B'=diag[0\ \beta_i]=diag[0\ 1];\quad dim(F_A')=3x6;$$

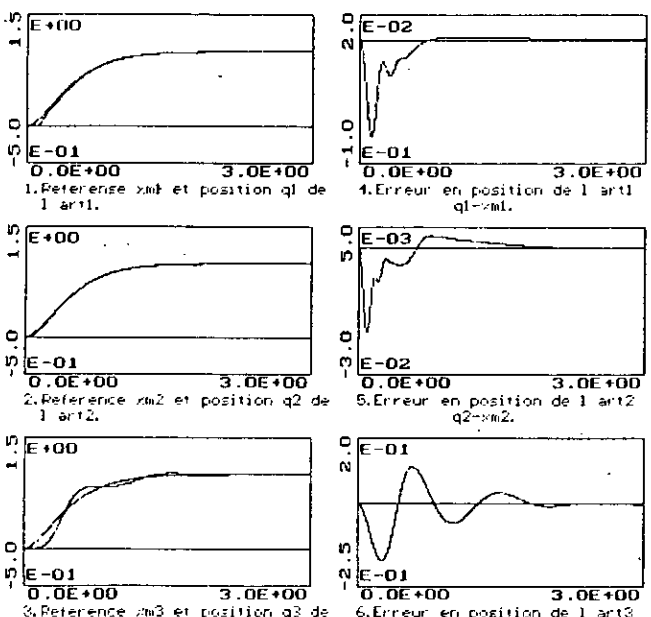
$$Q=diag\begin{bmatrix}9 & 0\\ 0 & 1\end{bmatrix}\ et\ \lambda_1=\lambda_2=-4.$$

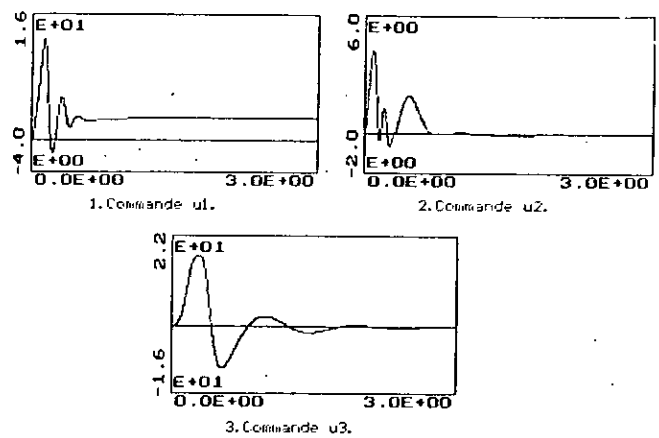
Les résultats de simulation avec de telles caractéristiques, pour une entrée unitaire, sont consignés sur les figures VI.MCS.5.a et b.
Dans le cas où on choisit:

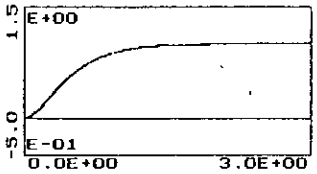
$$\alpha_1=333;\ \alpha_2=533;\ \alpha_3=100.$$

$$\beta_1=3.33;\ \beta_2=5.33;\ \beta_3=1.$$

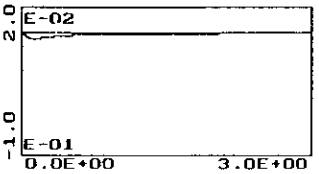
Les résultats de simulation sont identiques à ceux consignés aux figures VI.1.a,b.



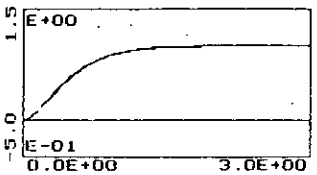




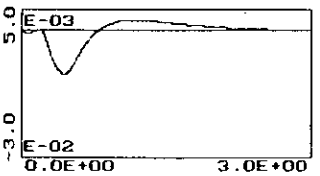
1.Reference xm1 et position q1 de l art1.



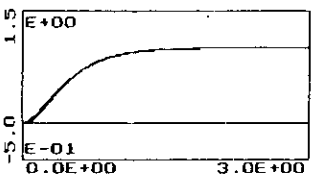
4.Erreur en position de l art1 q1- m1.



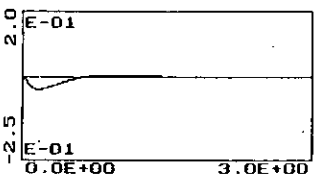
2.Reference xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

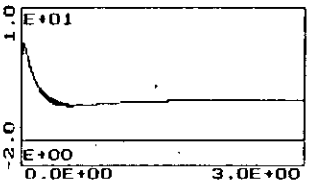


3.Reference xm3 et position q3 de l art3.

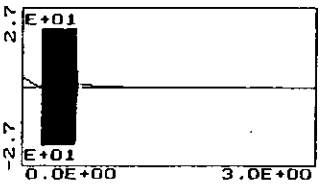


6.Erreur en position de l art3 q3-xm3.

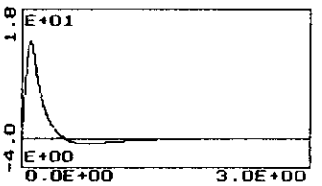
Figure VI.MCS.2.a Position et erreur en position du AMFC.
(Kp=Kv=0; U=B^T Pe; β_2=60)



1.Commande u1.

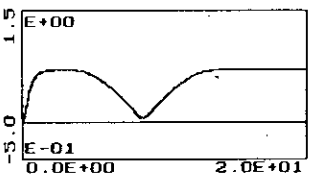


2.Commande u2.

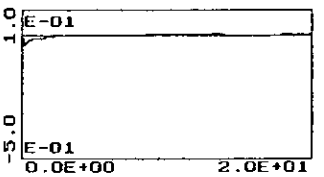


3.Commande u3.

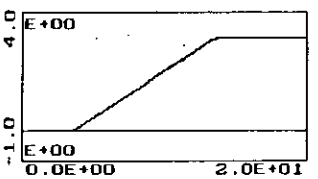
Figure VI.MCS.2.b Presentation des commandes.



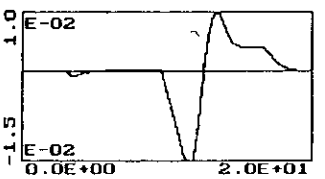
1.Reference xm1 et position q1 de l art1.



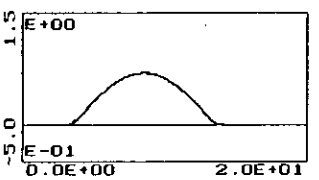
4.Erreur en position de l art1 q1-xm1.



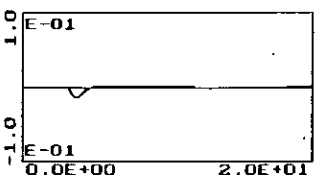
2.Reference xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

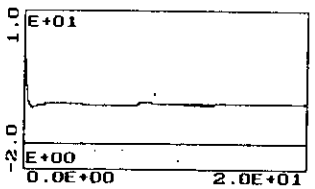


3.Reference xm3 et position q3 de l art3.

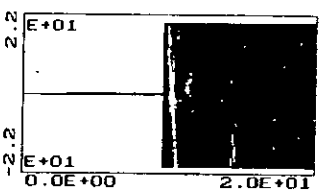


6.Erreur en position de l art3 q3-xm3.

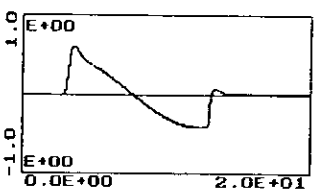
Figure VI.MCS.3.a Position et erreur en position pour AMFC.
(Kp=Kv=0; U=B^T Pe; β_2=60; consigne tenetre de U1U1nN1)



1.Commande u1.

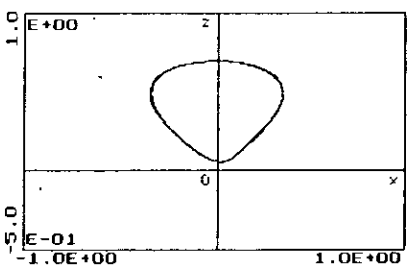


2.Commande u2.

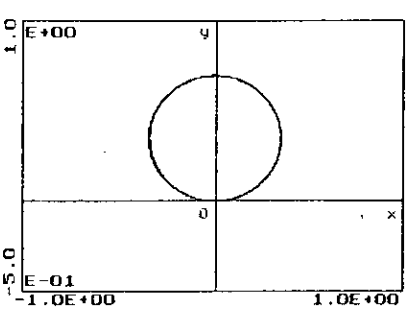


3.Commande u3.

Figure VI.MCS.3.b Presentation des commandes.

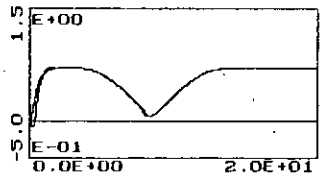


1.Projection sur le plan xoz.

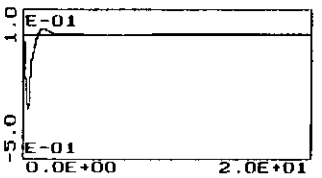


2.Projection sur le plan xoy.

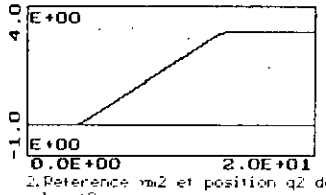
Figure VI.MCS.3.c Projection des trajectoires désirées et réelles sur les plan principaux.



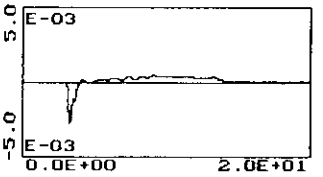
1.Référence xm1 et position q1 de l art1.



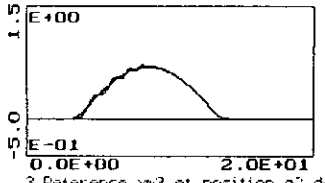
4.Erreur en position de l art1 q1-xm1.



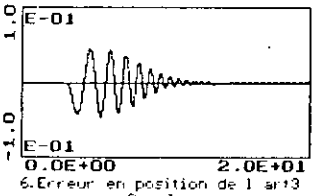
2.Référence xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

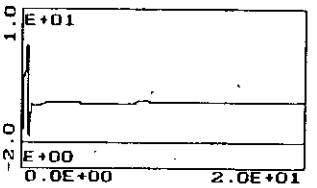


3.Référence xm3 et position q3 de l art3.

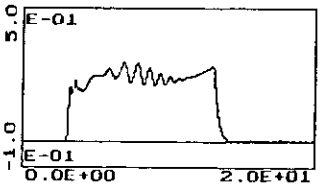


6.Erreur en position de l art3 q3-xm3.

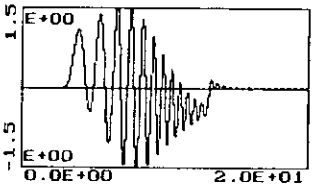
Figure VI.NCS.4.a Position et erreur en position pour AMFC.
(kp=1 v=0; U=B⁺Pe; β=1; consigne fenetre de WUHANI)



1.Commande u1.

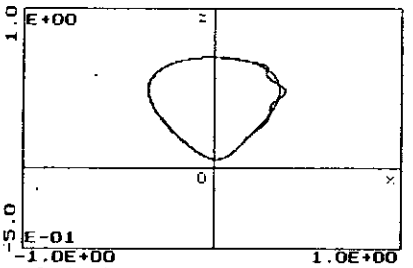


2.Commande u2.

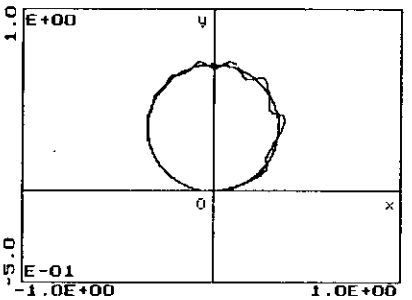


3.Commande u3.

Figure VI.NCS.4.b Presentation des commandes.

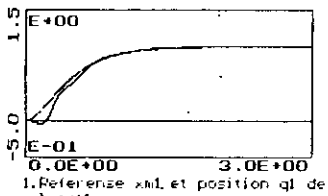


1.Projection sur le plan xoz.

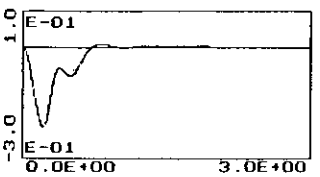


2.Projection sur le plan xoy.

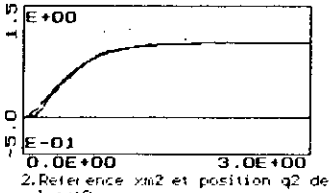
Figure VI.NCS.4.c Projection des trajectoires désirées et réelles sur les plan principaux.



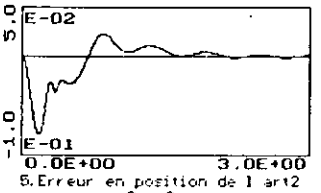
1.Référence xm1 et position q1 de l art1.



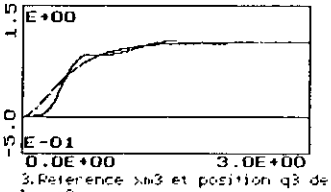
4.Erreur en position de l art1 q1-xm1.



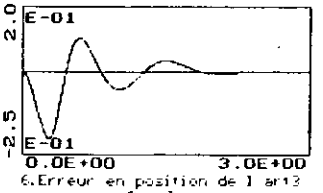
2.Référence xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

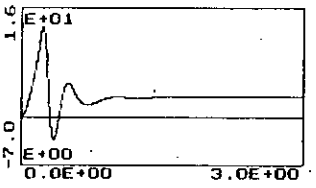


3.Référence xm3 et position q3 de l art3.

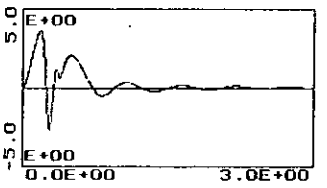


6.Erreur en position de l art3 q3-xm3.

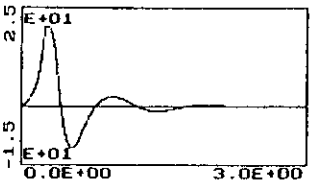
Figure VI.NCS.5.a Position et erreur en position du NCS.
(U=Pe; Fmdiag(0 1000; Fm²=diag(0 1))



1.Commande u1.



2.Commande u2.



3.Commande u3.

Figure VI.NCS.5.b Presentation des commandes.

VI.5 Commande adaptative discrète à modèle de référence. (Discrète MRAC).

Dans les parties précedentes de ce chapitre, on a présenté la commande adaptative continue à modèle de référence. L'implémentation d'une telle commande nécessite un calculateur rapide, qui permet, pendant un temps très faible, de calculer la commande et appliquer cette dernière au robot.

La version discrète de cet algorithme est nécessaire. Elle permet de voir l'influence de la période d'échantillonnage sur les performances de la commande. Dans un tel cas, la synthèse de la loi de commande se base sur un modèle discret.
Stoten [113] propose une version discrète du MRAC, en approximant le modèle d'état continu du robot par un modèle d'état discret (avec une approximation du premier ordre). La loi de commande synthétisée est similaire à la commande adaptative par poursuite d'un modèle. L'adaptation est du type PI, synthétisée en utilisant une erreur généralisée particulière.

Tarokh [114] présente la commande adaptative discrète à modèle de référence, en approximant le modèle de connaissance du robot (équations différentielle du second ordre) par des équations aux différences. La synthèse de la loi d'adaptation se base sur l'imposition de la dynamique de l'erreur. En utilisant la théorie de l'hyperstabilité, l'ajustement des paramètres, est du type PI, choisi d'une manière particulière. Le choix a été fait de telle sorte à satisfaire l'hyperstabilité asymptotique de l'erreur.

On présente dans ce qui suit, le développement de l'algorithme de Tarokh avec une application de ce dernier à notre robot.

VI.5.a Modèle discret du robot.

Considérons le modèle de connaissance du robot défini par [114]:

$$B_2(q)\ddot{q}(t)+B_1(q,\dot{q})\dot{q}(t)+B_0(q,\dot{q})q(t) = u(t) \qquad \text{(VI.53)}$$

Où $B_2(.)$, $B_1(.)$ *et* $B_0(.)$ sont des matrices de fonctions non linéaires de la position, de la vitesse et de la masse.
q: variable généralisé exprimant la position.

Pendant le mouvement du robot (dynamique), ces matrices sont variables dans le temps et l'équation (VI.53) peut s'écrire sous la forme:

$$B_2(t)\ddot{q}(t)+B_1(t)\dot{q}(t)+B_0(t)q(t) = u(t) \qquad \text{(VI.54)}$$

L'équation différentielle du second ordre, définie par (VI.54), peut s'écrire sous forme d'équation aux différences suivante [114]:

$$A_2(k)q(k-2)+A_1(k)q(k-1)+A_0(k)q(k) = u(k) \qquad \text{(VI.55)}$$

Où $A_2(.)$, $A_1(.)$ *et* $A_0(.)$: matrices de dimension nxn.

Pour que le modèle discret, défini par (VI.55), décrit d'une façon acceptable, le comportement dynamique du robot, une relation explicite entre les matrices du modèle discret et celles du modèle continu, est nécéssaire. Cette relation peut être obtenue en utilisant les approximations de premières ordres suivantes [114]:

$$\dot{q}(t) \approx \frac{1}{T}[q(k) - q(k-1)]$$
$$\ddot{q}(t) \approx \frac{1}{T^2}[q(k) - 2q(k-1) + q(k-2)] \qquad \text{(VI.56)}$$

Où T: période d'échantillonnage; $k \in N$.

En remplaçant les expressions de $\dot{q}(t)$ et $\ddot{q}(t)$ (équation (IV.56)) dans (VI.54) on trouve:

$$\begin{aligned} A_2(k) &= \frac{B_2(k)}{T^2} \\ A_1(k) &= -\frac{B_1(k)}{T} - 2\frac{B_2(k)}{T^2} \\ A_0(k) &= B_0(k) + \frac{B_1(k)}{T} + \frac{B_2(k)}{T^2} \end{aligned} \qquad \text{(VI.57)}$$

Ainsi le modèle discret est établi. C'est une équation aux différence similaire à celle définie par (IV.19).

VI.5.b Schéma de commande adaptative.

On s'intéresse à développer un algorithme de commande adaptative discret à base du modèle (VI.55), qui assure un suivi de trajectoire désirée, avec une dynamique imposée, sans aucune connaissance explicite sur le modèle du robot.
On définit le vecteur erreur suivant:

$$\varepsilon_q(k) = q_m(k) - q(k) \qquad \text{(VI.58)}$$

Où q_m: trajectoire désirée.

La dynamique de l'erreur peut être obtenu, en utilisant les équations (VI.58) et (VI.55). On a alors l'équation aux différences suivante:

$$\begin{aligned} A_0(k)\varepsilon_q(k) = &-u(k) - A_1(k)\varepsilon_q(k-1) - A_2(k)\varepsilon_q(k-2) \\ &+A_0(k)q_m(k) + A_1(k)q_m(k-1) + A_2(k)q_m(k-2) \end{aligned} \qquad \text{(VI.59)}$$

De l'équation (VI.59), on peut suggérer une commande qui influencera la dynamique de l'erreur. Cette commande est [114]:

$$\begin{aligned} u(k) = &P_1(k)\varepsilon_q(k-1) + P_2(k)\varepsilon_q(k-2) + Q_0(k)q_m(k) \\ &+Q_1(k)q_m(k-1) + Q_2(k)q_m(k-2) + \eta(k) \end{aligned} \qquad \text{(VI.60)}$$

Où $P_1(k), P_2(k)$: matrices variables dans le temps, du retour (feedback).
$Q_0(k), Q_1(k), Q_2(k)$: matrices variables dans le temps, de la chaîne directe (feedforward).
Et $\eta(k)$: un signal auxiliaire pour améliorer l'adaptation.

Le schéma de commande est défini par la figure VI.12.
En substituant (VI.60) dans (VI.59), l'équation aux différences de l'erreur devient:

$$\begin{aligned} &A_0(k)\varepsilon_q(k) + [P_1(k) + A_1(k)]\varepsilon_q(k-1) + [P_2(k) + A_2(k)]\varepsilon_q(k-2) = \\ &[A_0(k) - Q_0(k)]q_m(k) + [A_1(k) - Q_1(k)]q_m(k-1) + [A_2(k) - Q_2(k)]q_m(k-2) - \eta(k) \end{aligned} \qquad \text{(VI.61)}$$

On suppose que les performances désirées du robot, sont définies par l'imposition de

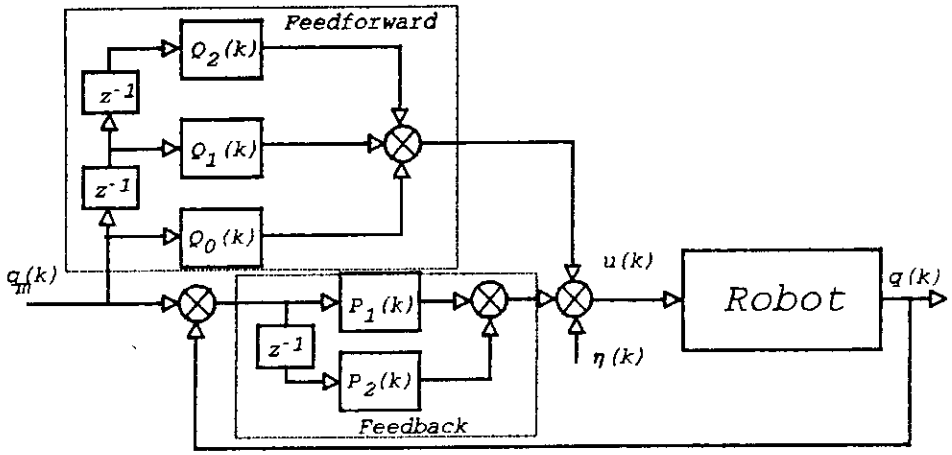


Figure VI.12 Schéma de commande discrète sans adaptation.

la dynamique de l'erreur [114]:

$$\varepsilon_m(k) + C_1\varepsilon_m(k-1) + C_2\varepsilon_m(k-2) = 0 \qquad \text{(VI.62)}$$

Où $\varepsilon_m(k)$: vecteur de dimension n, du modèle de référence de l'erreur.
C_1 *et* C_2 : matrices de dimension nxn, choisies de telle sorte à définir la dynamique de $\varepsilon_m(k)$.

Pour avoir un modèle de référence découplé, on choisit [114]:

$$C_1 = diag(c_{1_i}) \text{ et } C_2 = diag(c_{2_i}) \qquad \text{(VI.63)}$$

Le polynôme caractéristique de cette dynamique est alors:

$$\Delta(z) = \det(I_n z^2 + C_1 z + C_2) = \prod_{i=1}^{n} \delta_i(z) ; \qquad \text{(VI.63.a)}$$

Où $\delta_i(z) = z^2 + C_{1_i} z + C_{2_i} = (z+\lambda_{1_i})(z+\lambda_{2_i})$ *et* $C_{1_i} = \lambda_{1_i} + \lambda_{2_i}$; $C_{2_i} = \lambda_{1_i}\lambda_{2_i}$.

avec $|\lambda_{1_i}| < 1$; $|\lambda_{2_i}| < 1$.

On définit la déviation, qui existe entre l'erreur idéal $\varepsilon_m(k)$ et l'erreur actuelle $\varepsilon_q(k)$ comme [114]:

$$\varepsilon(k) = \varepsilon_m(k) - \varepsilon_q(k) \qquad \text{(VI.64)}$$

En combinant (VI.61), (VI.62) et (VI.64), on trouve:

$$\varepsilon(k) + C_1\varepsilon(k-1) + C_2\varepsilon(k-2) + w(k) = 0 \qquad \text{(VI.65)}$$

Où
$$w(k) = \psi_1(k)\varepsilon_q(k-1) + \psi_2(k)\varepsilon_q(k-2) + \psi_3(k)q_m(k) + \psi_4(k)q_m(k-1) + \psi_5(k)q_m(k-2) + \rho(k) .$$

Avec
$$\psi_1(k) = C_1 - A_0^{-1}(k)[A_1(k) + P_1(k)] ; \quad \psi_2(k) = C_2 - A_0^{-1}(k)[A_2(k) + P_2(k)] ;$$
$$\psi_3(k) = A_0^{-1}(k)[A_0(k) + Q_0(k)] ; \quad \psi_4(k) = A_0^{-1}(k)[A_1(k) + Q_1(k)] ;$$
$$\psi_5(k) = A_0^{-1}(k)[A_2(k) + Q_2(k)] ; \text{ et } \rho(k) = -A_0^{-1}(k)\eta(k) .$$

Le problème, dans cette synthèse, c'est de trouver une loi d'adaptation, de la chaine de retour $(P_1(k), P_2(k))$, de la chaine directe $(Q_1(k), Q_2(k), Q_0(k))$ et du signal

auxiliaire $\eta(k)$, de telle sorte à assurer la convergence asymptotique vers zéro de $\varepsilon(k)$. Cette convergence assure la poursuite parfaite de la trajectoire désirée, avec la dynamique désirée.

On met sous la forme standard, pour L'étude de la stabilité et déduire une loi d'adaptation.
L'équation (VI.65) peut se mettre sous la forme suivante:

$$\begin{bmatrix} \varepsilon(k-1) \\ \varepsilon(k) \end{bmatrix} = \begin{bmatrix} 0 & I_n \\ -C_2 & -C_1 \end{bmatrix} \begin{bmatrix} \varepsilon(k-2) \\ \varepsilon(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ I_n \end{bmatrix} w(k) \qquad \text{(VI.66)}$$

Où I_n: matrice identité de dimension nxn.

En considérant le vecteur erreur suivant:

$$V(k) = D \begin{bmatrix} \varepsilon(k-1) \\ \varepsilon(k) \end{bmatrix} \qquad \text{(VI.67)}$$

La représentation équivalente des équations (VI.66) et (VI.67) est définie par la figure VI.13.

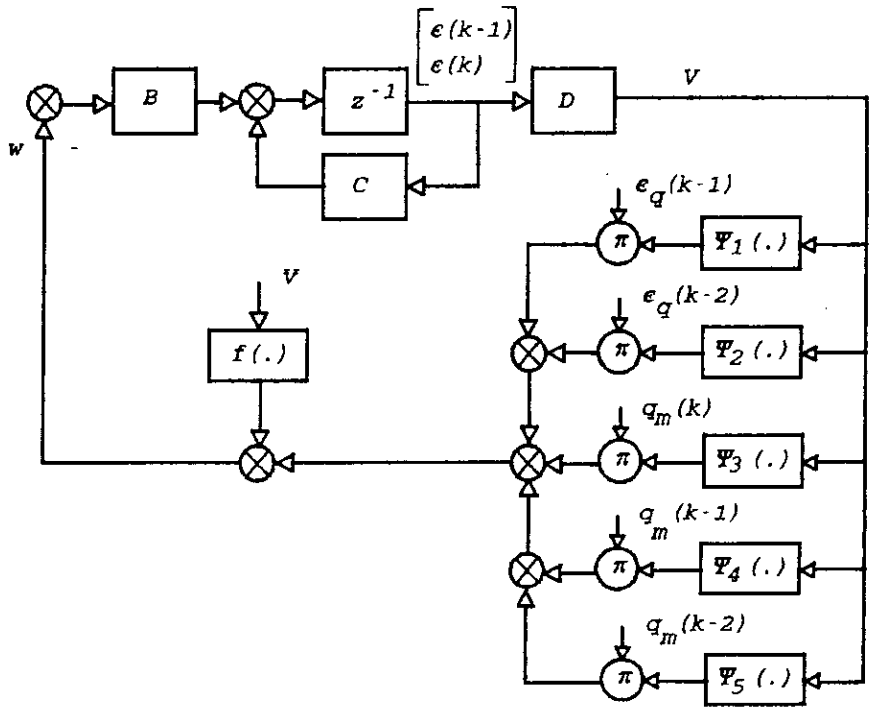


Figure VI.13 Représentation équivalente de la dynamique de l'erreur.

Où $B=\begin{bmatrix} 0 \\ I_n \end{bmatrix}$ et $C=\begin{bmatrix} 0 & I_n \\ -C_2 & -C_1 \end{bmatrix}$.

En utilisant la théorie de l'hyperstabilité de Popov [95] et le théorème de Landau [88], on déduit que la structure de la figure VI.13 est assymptotiquement hyperstable si:

1- La matrice de transfert discrète:

$$H(z) = D(zI-C)^{-1}B \qquad (VI.68)$$

est strictement réelle positive.

2- Le retour non linéaire et variant dans le temps, doit satisfaire l'inégalité de Popov, définie par [88][114] :

$$\sum_{k=0}^{k_1} V^T(k)\, w(k) \geq -\gamma_0^2; \quad \forall k_1 > 0 \qquad (VI.69)$$

Pour satisfaire les conditions précédentes, Tarokh propose une loi d'adaptation du type PI définie par [114] (voir annexe 8):

$$P_1(k) = P_1(k-1) + \hat{\varepsilon}_q(k)\,\varepsilon_q^T(k-1)\,E_{1_p} + \hat{\varepsilon}_q(k-1)\,\varepsilon_q^T(k-2)\left[E_{1_I} - E_{1_p}\right] \qquad (VI.70.a)$$

$$P_2(k) = P_2(k-1) + \hat{\varepsilon}_q(k)\,\varepsilon_q^T(k-2)\,E_{2_p} + \hat{\varepsilon}_q(k-1)\,\varepsilon_q^T(k-3)\left[E_{2_I} - E_{2_p}\right] \qquad (VI.70.b)$$

$$Q_0(k) = Q_0(k-1) + \hat{\varepsilon}_q(k)\,q_m^T(k)\,F_{0_p} + \hat{\varepsilon}_q(k-1)\,q_m^T(k-1)\left[F_{0_I} - F_{0_p}\right] \qquad (VI.70.c)$$

$$Q_1(k) = Q_1(k-1) + \hat{\varepsilon}_q(k)\,q_m^T(k-1)\,F_{1_p} + \hat{\varepsilon}_q(k-1)\,q_m^T(k-2)\left[F_{1_I} - F_{1_p}\right] \qquad (VI.70.d)$$

$$Q_2(k) = Q_2(k-1) + \hat{\varepsilon}_q(k)\,q_m^T(k-2)\,F_{2_p} + \hat{\varepsilon}_q(k-1)\,q_m^T(k-3)\left[F_{2_I} - F_{2_p}\right] \qquad (VI.70.e)$$

$$\eta(k) = \eta(k-1) + \beta_p \hat{\varepsilon}_q(k) + (\beta_I - \beta_p)\,\hat{\varepsilon}_q(k-1) \qquad (VI.70.f)$$

$$\hat{\varepsilon}_q(k) = R_2 \varepsilon_q(k-1) + R_3 \varepsilon_q(k) \qquad (VI.70.g)$$

Où $E_{1_p}, E_{1_I}, \ldots, F_{2_p}, F_{2_I}$: matrices constantes symétriques définies positives.
β_p, β_I: scalaires positives.

Les indices p et I dénotent respectivement les parties proportionnelles et intégrales.
R_2, R_3: matrices diagonales dont les éléments sont [114]:

$$\begin{aligned} r_{2_i} &= \alpha_i \lambda_{1_i} \lambda_{2_i} (\lambda_{1_i} + \lambda_{2_i}) \\ r_{3_i} &= \alpha_i (1 + \lambda_{1_I} \lambda_{2_I}); \quad i = 1, n \end{aligned} \qquad (VI.71)$$

La conditions de positivité réelle de la chaîne directe, du schéma de la figure VI.13, est assurée par le choix (VI.70.g) et (VI.71), en utilisant le critère MKY (Meyer Kalman Yokotovic)[88][114], sans résolution de l'équation de Lyapounov.

Remarque 1.
La relation qui relie $V(k)$ et $\varepsilon(k)$ est définie par [114]:
$V(k) = R_2 \varepsilon(k-1) + R_3 \varepsilon(k)$
d'où, en utilisant (VI.67), on a: $D = [R_2 \; R_3]$. Or $\varepsilon_m(q)$ tend vers zéro, donc:
$V(k) = -[R_2 \varepsilon_q(k-1) + R_3 \varepsilon_q(k)] = -\hat{\varepsilon}_q(k)$. □

Remarque 2.
Les conditions initiales de l'algorithme, peuvent être nulles ou arbitraires. □

Le schéma de la commande adaptative est défini par la figure VI.14.

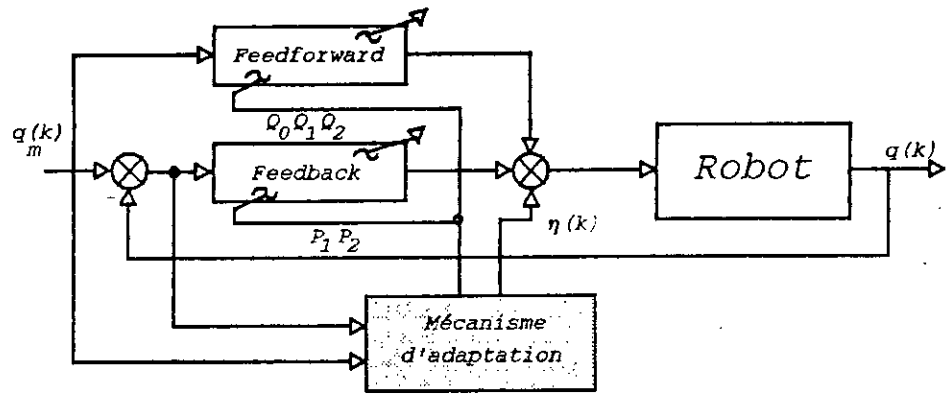


Figure VI.14 Schéma de commande adaptative discrète.

Algorithme de commande.

Données: - Degré du système et signe de B.
- Choisir le modèle de référence (VI.62).
- Choisir R_2 *et* R_3 en utilisant (VI.71).
- Choisir les matrices $E_{1_P}, E_{1_I}, \ldots, F_{2_P}, F_{2_I}$, dans (VI.70).
- Choisir β_P *et* β_I.
- Choisir les valeurs initiales de P_1, P_2, Q_0, Q_1, Q_2 *et* η.

Etape 1: Générer la trajectoire référence $q_m(k)$.

Etape 2: Adaptation des paramètres P_1, P_2, Q_0, Q_1, Q_2 *et* η, en utilisant (VI.50.a-e). Synthèse de η de (VI.70.f).

Etape 3: Calcul de la commande à partir de (VI.60).

t=t+1 revenir à étape 1.

VI.5.c Résultats de simulation.

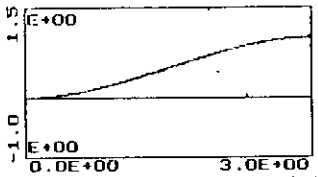
Pour la simulation du DMRAC (Discrète MRAC), la période d'échantillonnage est choisie de valeurs 10ms et le pas de calcul de simulation du robot est de 1ms [114]. L'algorithme d'adaptation choisi est du type intégrale [114] avec:

$Q_0=Q_1=Q_2=0.$
$E_{1_P}=E_{2_P}=0.$
$E_{1_I}=E_{2_I}=3I.$
$\alpha_i=100;\ \lambda_{1_i}=\lambda_{2_i}=-0.7;\ \beta_i=0.02;\ i=1,3.$

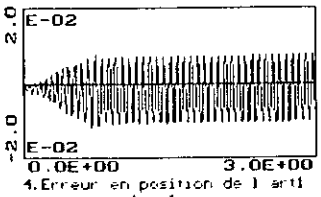
En faisant varier la valeur de β_P respectivement de 30, 10 et 5, pour une entrée polynomiale, les résultats de simulation sont respectivement consignés sur les figures VI.DMRAC.1, VI.DMRAC.2, VI.DMRAC.3.

Pour le test de la robustesse, on maintient β_P=5 et on fait une variation paramétrique de 500% à t=1,5s. Les résultats sont consignés sur les figures VI.DMRAC.4.a et b.

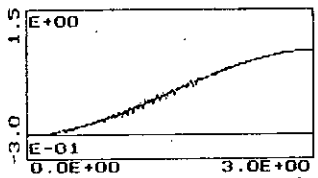
En ce qui concerne la poursuite, la fenêtre de VIVIANI(avec un lissage au démarrage avec un polynôme du troisième degré), a eté imposée à l'entrée du robot, dont les résultats de simulation sont illustrés aux figures VI.DMRAC.5.a, b et c.



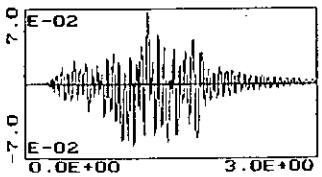
1.Reference xm1 et position q1 de l art1.



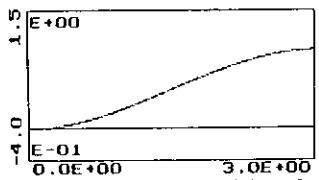
4.Erreur en position de l art1 q1-xm1.



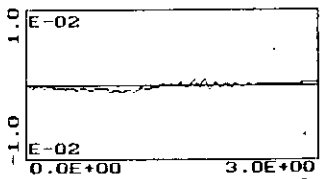
2.Reference xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

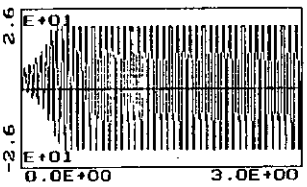


3.Reference xm3 et position q3 de l art3.

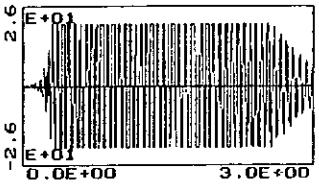


6.Erreur en position de l art3 q3-xm3.

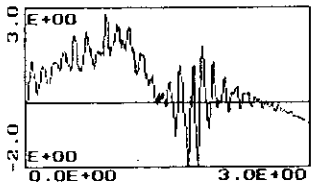
Figure VI.DMRAC.1.a Position et erreur en position pour DMRAC. (trajectoire polynomiale;β_r=30)



1.Commande u1.

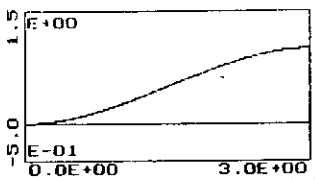


2.Commande u2.

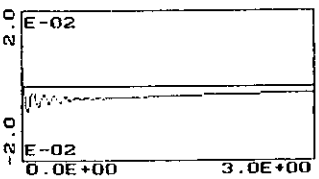


3.Commande u3.

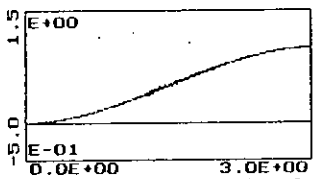
Figure VI.DMRAC.1.b Presentation des commandes.



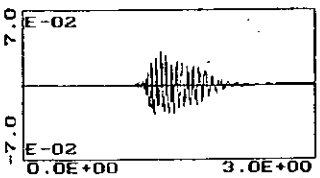
1.Reference xm1 et position q1 de l art1.



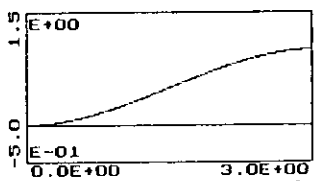
4.Erreur en position de l art1 q1-xm1.



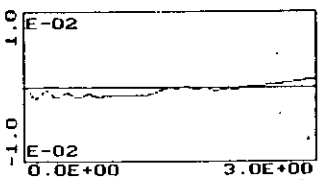
2.Reference xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

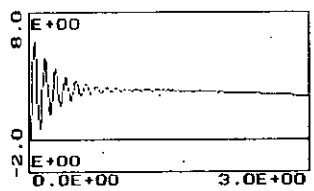


3.Reference xm3 et position q3 de l art3.

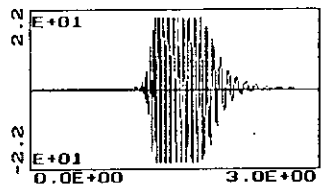


6.Erreur en position de l art3 q3-xm3.

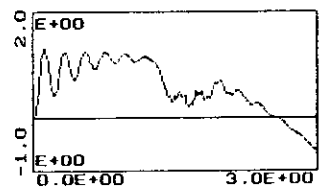
Figure VI.DMRAC.2.a Position et erreur en position pour DMRAC. (trajectoire polynomiale;β_r=10)



1.Commande u1.

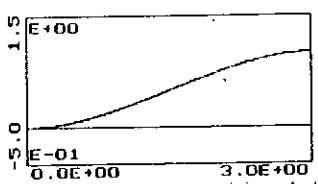


2.Commande u2.

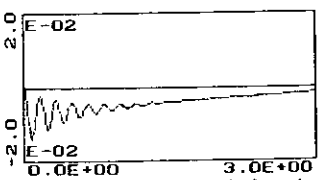


3.Commande u3.

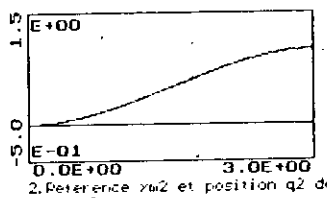
Figure VI.DMRAC.2.b Presentation des commandes.



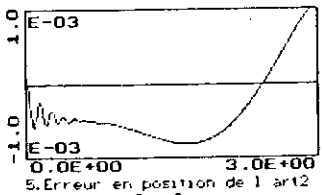
1.Reference xm1 et position q1 de l art1.



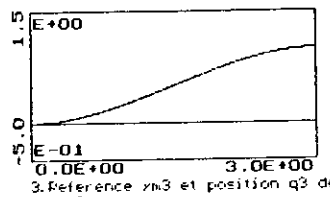
4.Erreur en position de l art1 q1-xm1.



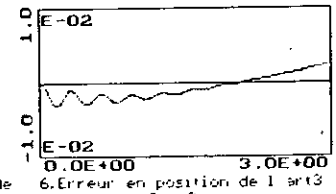
2.Reference xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

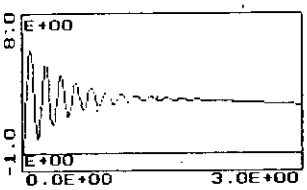


3.Reference xm3 et position q3 de l art3.

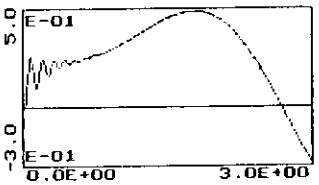


6.Erreur en position de l art3 q3-xm3.

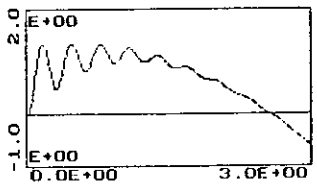
Figure VI.DMRAC.3.a Position et erreur en position pour DMRAC. (trajectoire polynomiale;β_r=5)



1.Commande u1.

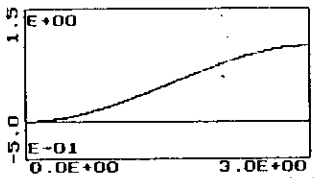


2.Commande u2.

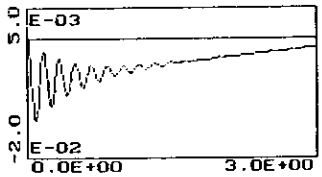


3.Commande u3.

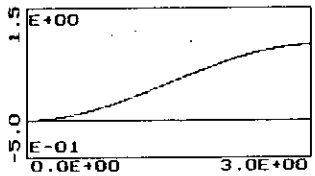
Figure VI.DMRAC.3.b Presentation des commandes.



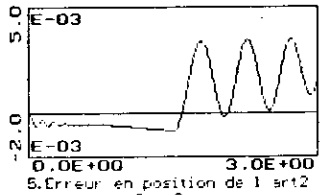
1.Reference xm1 et position q1 de l art1.



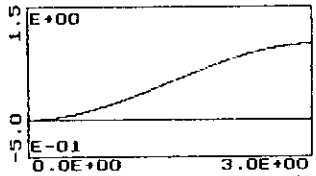
4.Erreur en position de l art1 q1-xm1.



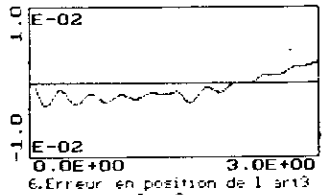
2.Reference xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

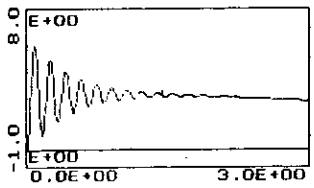


3.Reference xm3 et position q3 de l art3.

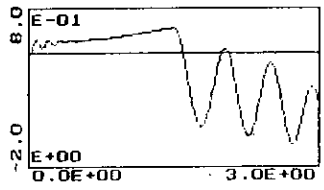


6.Erreur en position de l art3 q3-xm3.

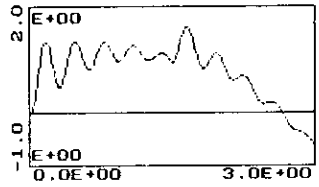
Figure VI.DMRAC.4.a Position et erreur en position pour DMRAC.(trajectoir (polynomiale; variation du modèle du robot de 500% à t=1.5s; =5)



1.Commande u1.

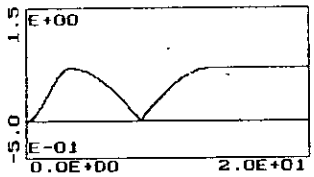


2.Commande u2.

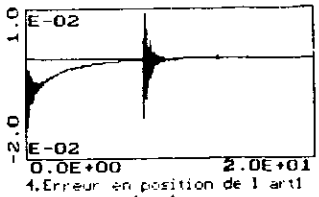


3.Commande u3.

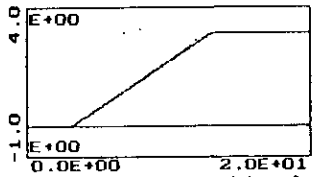
Figure VI.DMRAC.4.b Presentation des commandes.



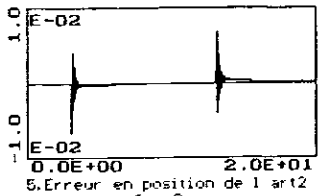
1.Reference xm1 et position q1 de l art1.



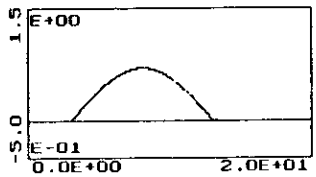
4.Erreur en position de l art1 q1-xm1.



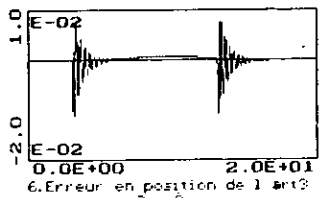
2.Reference xm2 et position q2 de l art2.



5.Erreur en position de l art2 q2-xm2.

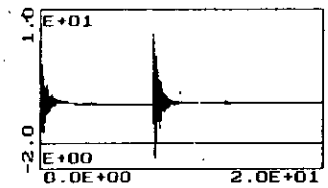


3.Reference xm3 et position q3 de l art3.

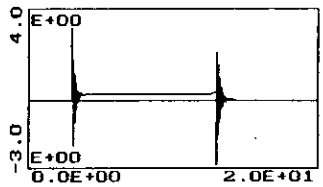


6.Erreur en position de l art3 q3-xm3.

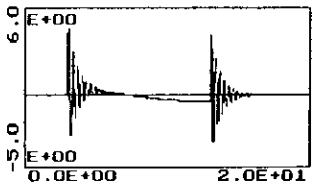
Figure VI.DMRAC.5.a Position et erreur en position pour DMRAC. (consigne fenetre de VI(THHI; =5)



1.Commande u1.

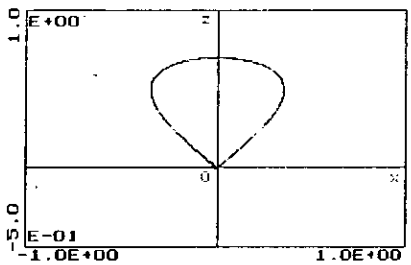


2.Commande u2.

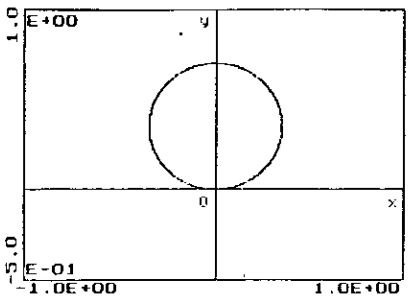


3.Commande u3.

Figure VI.DMRAC.5.b Presentation des commandes.



1.Projection sur le plan xoz.



2.Projection sur le plan xoy.

Figure VI.DMRAC.5.c Projection des trajectoires désirées et réelles sur les plan principaux.

Conclusion

Nous avons présenté, dans la première partie de ce chapitre, une méthode d'analyse et de synthèse de loi d'adaptation à modèle de référence. La commande linéaire par poursuite d'un modèle, ainsi que sa version adaptative sont présentées dans la seconde partie. La troisième partie a été consacrée à l'algorithme MCS. En dernier lieu, on a développé l'algorithme discret du MRAC.

La commande linéaire par poursuite d'un modèle (LMFC) montre son efficacité dans le cas où le terme de perturbation est nul. Les conditions d'Erzerberger sont toujours satisfaites pour les robot manipulateurs de classe une et quatre [4][107]. Lorsque le terme nonlinéaire d, est non nul , cette dernière n'est plus adéquate. En imposant un modèle de référence rapide, l'amélioration dans la poursuite est remarquable, en dépit d'un effort de commande important (irréalisable). En introduisant une limitation à ce dernier, les performances se détériorent. L'utilisation d'une loi adaptative s'avère nécessaire.
La commande adaptative par poursuite d'un modèle (AMFC) donne de bons résultats, avec une loi d'adaptation du type PI, l'effort de commande est acceptable.
Une telle loi de commande est robuste envers des variations paramétrique et assure une bonne poursuite, même dans le cas de trajectoires discontinues en vitesse.

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. Théoriquement, l'hyperstabilité d'un tel algorithme n'est pas démontrée, car cette loi n'est pas la même que celle développée dans ce chapitre. Par contre, cet algorithme montre son efficacité, pour un bon choix des paramètres de pondération de la loi d'adaptation. Une augmentation dans l'adaptation proportionnelle, améliore les performances de poursuite avec un effort de commande moindre mais oscillant.
La loi de commande MCS, développée dans ce chapitre, présente les mêmes caractéristiques que la précédente. Pour un choix particulier des matrices de pondération de la loi d'adaptation, on aboutit aux mêmes résultats. La différence, dans ces deux algorithme, réside dans l'utilisation de l'erreur V. Pour l'MCS, issus de AMFC, on a $V=B^TPe$ et pour l'MCS développé, on a $V=Pe$. Le second algorithme nécessite un choix particulier des matrices de pondération, surtout dans le cas où ces dernières sont rectangulaires (cas de notre robot).
La connaissance du signe des coefficients de la matrice B, ou la positivité de cette dernière, est une condition nécessaire pour assure l'hyperstabilité asymptotique de l'erreur. Les coefficients de la matrice B sont toujours positifs , dans le cas des robots manipulateurs [107]. L'utilisation d'un algorithme, ne nécessitant pas la connaissance du signe de B, s'avère nécessaire.

La version discrète du MRAC, développée dans ce chapitre, ne nécessite ni la connaissance du modèle du robot, ni le signe de B. Une loi d'adaptation du type intégrale est suffisante pour ajuster le feed-back, sans que le feedforward existe. Implicitement l'adaptation proportionnelle est incluse dans le signal auxiliaire $\eta\,(coef\ \beta_p)$.
Les performances de poursuite, ainsi que l'effort de commande, sont très sensibles aux variations de l'action proportionnelle, dans le loi d'adaptation de η. Un bon choix de ce paramètre, assure une très bonne poursuite, avec un effort de commande réalisable et lisse.
La robustesse d'un tel algorithme, est vérifié dans le cas de variation paramétrique. Une discontinuité en position de la trajectoire désirée, n'est pas tolérable surtout au démarrage. Par contre une discontinuité en vitesse de cette dernière, n'affecte pas les performances de poursuite, mais des oscillations importantes dans l'effort

de commande en résulte. Donc cet algorithme nécessite des trajectoires désirées, assurant un continuité en position et en vitesse, pour aboutir à des efforts de commande lisses.

Les systèmes **adaptatifs** à modèle de référence, permettent l'analyse de la convergence des algorithmes adaptatifs et la synthèse des lois d'adaptation, dans les cas discret et continu. 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 développé.

CONCLUSION GENERALE

"Tout est dit ... tout reste à faire".

LUDOWING VAN BEETHOVEN

CONCLUSION GENERALE

La modélisation des procédés est une étape très importante pour la synthèse de n'importe quelle loi de commande adaptative. Le modèle de connaissance permet, en premier lieu, une simulation numérique ou analogique du procédé. En second lieu, ce modèle est nécessaire pour l'implémentation des lois de commande telles que la commande non linéaire hybride, la commande APC et la commande linéarisante.
Le modèle de représentation aux différence, est exigé par les commandes utilisant les régulateurs auto-ajustables (STR, APC).

La commande auto-ajustable nécessite la connaissance du degré du système, ainsi que son retard. Le choix des différentes pondération est lié aux performances désirées. Dans le cas de la GMV, l'introduction de la pondération est nécessaire pour la commande du robot. Cette pondération modifie les performances en boucle fermée. La spécification du modèle en boucle fermée, dans le cas du placement de pôles, permet d'échapper au comportement à phase non minimale du robot, mais les performances de poursuite se dégradent et nécessite un temps de calcul important.

La commande hybride permet de minimiser l'effet des termes non linéaires pour donner des asservissement 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 dés que l'erreur de suivi est importante.
L'utilisation de la vitesse et de l'accélération, dans l'algorithme de Newton-Euler, nécessite une 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 condidate 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 parralléle-parrelléle, 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.

Dans ce mémoire, on a présenté une analyse de différentes lois de commande adaptative, qui existent dans la littérature. On a utilisé des outils théoriques puissants, telles que la géométrie différentielle et l'hyperstabilite, pour la compréhension et la réalisation par simulation de tels algorithmes.
Malgré toute cette analyse, le domaine de la commande reste toujours vaste.
L'étude de la robustesse de tous ces algorithmes étudiés s'avère très utile ainsi que

le choix de l'algorithme de commande adéquat.
L'intégration d'une loi de supervision de ces différentes commandes, en fonction du comportement du système, exige l'utilisation de la théorie de la logique floue (fuzzy logic), pour alléger la charge de l'opérateur dans le choix de la commande adéquate et l'automatisation d'une partie concernant la recherche des coefficients de pondération.
Le temps de calcul des différentes lois d'adaptation élaborées, devient important suivant la complexité structurelle du système. L'utilisation d'une loi d'adaptation très rapide s'avère très intéressante. Cette rapidité peut être assurée en utilisant les réseaux de neurones. Ceci aboutit a un nouveau axe de recherche, dans lequel on combine la théorie de la commande adaptative et les réseaux de neurones. Le problème crucial qui se pose est l'étude de la stabilité de tels algorithmes. Cette étude peut être faite en utilisant la théorie d'hyperstabilité de Popov, à une condition de bien structurer l'algorithme sous forme d'un bloc directe SPR et d'un bloc de retour hyperstable.
La réalisation de réseaux de neurones adaptatifs, dont on doit synthétiser la loi d'adaptation, peut conduire à d'autre voies de recherche et l'utilisation de nouvelles théories, et nécessitant des outils de calcul très puissants.

ANNEXES

"Autant j'apprends, autant je deviens bête".

VICTOR HUGO

Annexe 1

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 Ann.1.1, qui passe par le point $P(x,y,z)$, nécessite les paramètres β *et* γ pour compléter sa description[5].

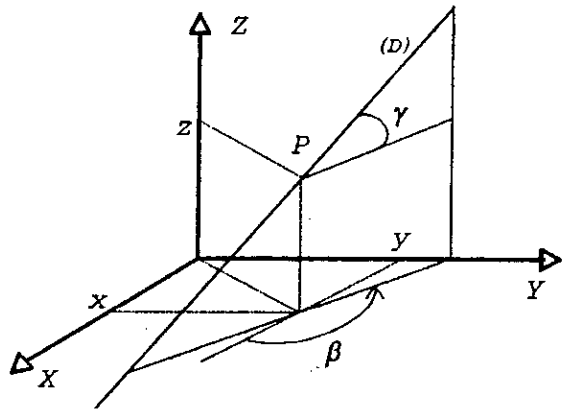


Figure Ann.1.1 Représentation de la droite dans l'espace.

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 Ann1.2.a donne une description basée sur la détection de l'unique perpendiculaire (OH') entre (D) *et* (Oz). Cette droite (D) est tangente à l'unique cylindre de rayon a et d'axe (Oz).
Donc les quatres paramètres sont α, θ ,*a et d* . Ayant déterminer 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 la 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 transformation (figure Ann1.2.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 θ .
$Rot(x,\theta)$: Rotation autour de x d'un angle α .
$Trans(0,0,d)$: Translation d'une distance d le long de z.
$Trans(a,0,0)$: Translation d'une distance a le long de x.

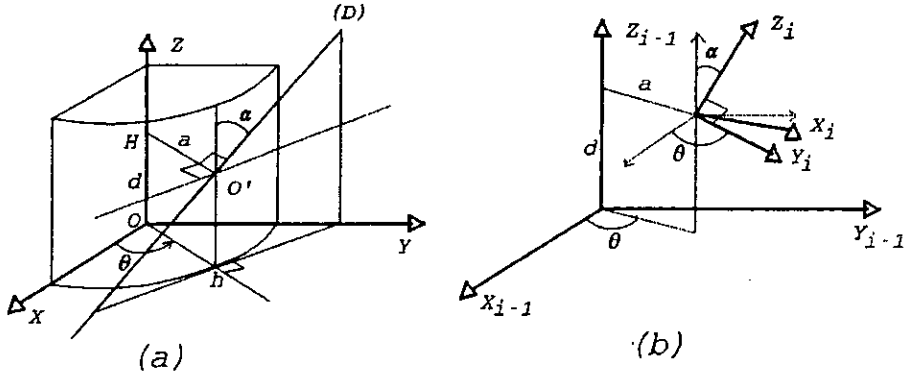


Figure Ann 1.2 Représentation de (D) dans l'espace.

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 qui va être utilisée 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: Numéroter chaque liaison et articulation, en commençant de la base, notée liaison 0 et l'effecteur (élément terminal) la liaison n (figure Ann 1.4). 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:
- a. L'axe Z_{i-1} est choisi le long de l'axe de l'articulation i.
- b. L'axe X_i est choisi perpendiculaire à Z_{i-1} dont le sens peut être choisi arbitrairement. ($X_i = Z_{i-1} \wedge Z_i$)
- c. L'axe Y_i est choisi de telle sorte à former un trièdre droit.

Etape 3: Définir les paramètres θ_i, d_i, a_i *et* α_i.
- θ_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).
- α_i: angle entre Z_{i-1} et Z_i.

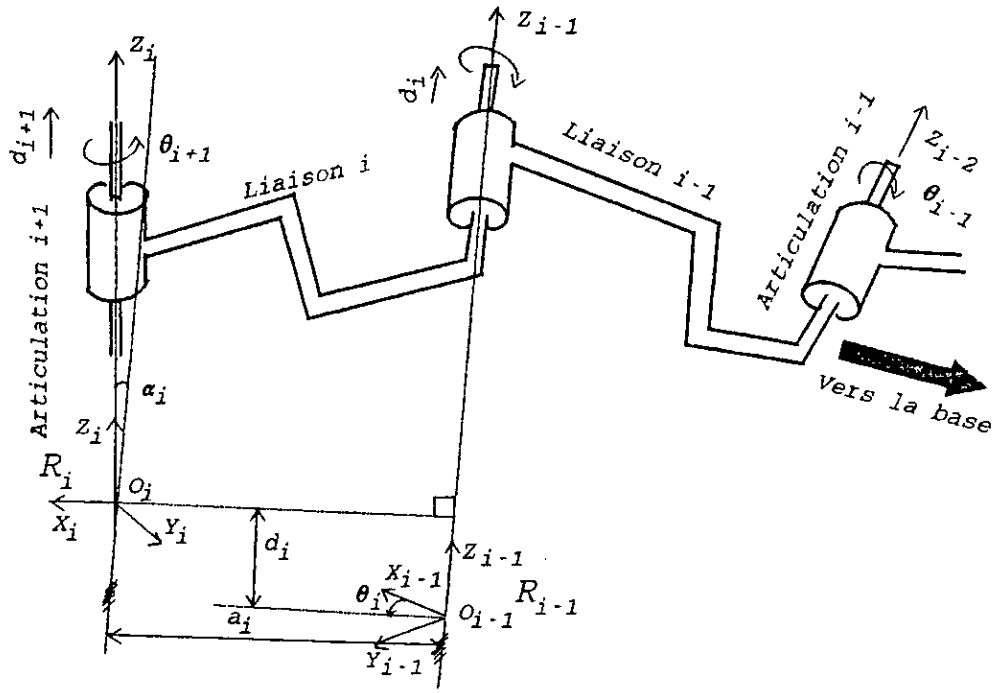


Figure Ann1.3 représentation des repères dans les articulations.

Annexe 2

ORGANIGRAMME DE L'ALGORITHME DE NE POUR LE ROBOT C4

Debut

Lecture des données

$m_1, m_3, l_2, k_1, k_2, k_3, g$

${}_2R^1, {}_1R^2, c_1, c_2, c_3, {}_3R^0P_3^*, {}_3R^0\bar{S}_3, {}_3R^0I_3\ {}_0R^3$

$\eta_4=0\ ,\ f_4=0.$

Forward equations.

1_{ere} liaison $\begin{cases} {}_1R^0\omega_1=0\ ;\ {}_1R^0\dot{V}_1=[0\ 0\ \ddot{q}_1+g]^T. \\ {}_1R^0\dot{\omega}_1=0\ ;\ {}_1R^0\bar{a}_1={}_1R^0\dot{V}_1. \end{cases}$

2_{eme} liaison $\begin{cases} {}_2R^0\omega_2=[0\ -\dot{q}_2\ 0]^T\ ;\ {}_1R^0\dot{\omega}_2=[0\ -\ddot{q}_2\ 0]^T. \\ {}_2R^0\dot{V}_2={}_2R^1\ {}_1R^0\dot{V}_1\ ;\ {}_2R^0\bar{a}_2={}_2R^0\dot{V}_2. \end{cases}$

3_{eme} liaison $\begin{cases} {}_3R^0\omega_3={}_2R^0\omega_2. \\ {}_3R^0\dot{\omega}_3={}_2R^0\dot{\omega}_2. \\ {}_3R^0\dot{V}_3=[0\ 0\ 1\ ]^T\ddot{q}_3\ {}_2R^0\dot{V}_2+({}_3R^0\dot{\omega}_3)\wedge({}_3R^0P_3^*)+2\,({}_3R^0\omega_3)\wedge([0\ 0\ \dot{q}_3]^T \\ +({}_3R^0\omega_3)\wedge\left[({}_3R^0\omega_3)\wedge({}_3R^0P_3^*)\right] \\ {}_3R^0\bar{a}_3=({}_3R^0\dot{\omega}_3)\wedge({}_3R^0\bar{S}_3)+{}_3R^0\dot{V}_3+({}_3R^0\omega_3)\wedge\left[({}_3R^0\omega_3)\wedge({}_3R^0\bar{S}_3)\right] \end{cases}$

Backward equations

3_{eme} liaison $\begin{cases} {}_3R^0f_3={}_4R^0f_4+m_3\,({}_3R^0\bar{a}_3)\,. \\ {}_3R^0\eta_3={}_4R^0\eta_4+({}_4R^0P_1^*)\wedge({}_4R^0f_4)+({}_3R^0\omega_3)\wedge\left[({}_3R^0I_3\ {}_0R^3)\ ({}_3R^0\omega_3)\right] \\ +({}_3R^0P_3^*+{}_3R^0\bar{S}_3)\wedge m_3\,({}_3R^0\bar{a}_3)+({}_3R^0I_3\ {}_0R^3)\ ({}_3R^0\dot{\omega}_3)\,. \end{cases}$

2_{eme} liaison $\begin{cases} {}_2R^0f_2={}_3R^0f_3. \\ {}_2R^0\eta_2={}_3R^0\eta_3. \end{cases}$

1_{ere} liaison $\{\ {}_1R^0f_1={}_1R^2\,({}_2R^0f_2)+m_1\ {}_1R^0a_1.$

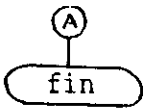
Calcul de u.

$$U(1)=\frac{[{}_1R^0f_1(3)+c_1\dot{q}_1]}{k_1}.$$

$$U(2)=\frac{[-{}_2R^0f_2(2)+c_2\dot{q}_2]}{k_2}.$$

$$U(3)=\frac{[{}_3R^0f_3(3)+c_3\dot{q}_3]}{k_3}.$$

(A)



Remarque.

c_1, c_2 *et* c_3: coefficients de frotement visqueux.□

Annexe 3

DEVELOPPEMENT DU MODELE DYNAMIQUE EN UTILISANT L'ALGORITHME DE NE

Dans cette annexe on présente le calcul analytique du modèle dynamique du robot en utilisant le formalisme de Newton-Euler. Le développement a été fait dans le but de valider le modèle calculé par le formalisme de Lagrange-Euler. Les considérations prises sont:

$$ {}_iR^0 I_i \; {}_0R^i = \begin{bmatrix} \alpha_i & 0 & 0 \\ 0 & \beta_i & 0 \\ 0 & 0 & \gamma_i \end{bmatrix}; \; i=1,3. $$

Les autres paramètres sont indiqués dans le chapitre I.3.2.
Pour i=1,2 et 3 la liaison est respectivement translationnelle, rotationnelle et translationnelle.

1ère étape. "forward équations".
pour i=1 on a:

$$ {}_1R^0\omega_1 = {}_1R^0({}_0R^0\omega_0) = 0. $$

$$ {}_1R^0\dot{\omega}_1 = {}_1R^0({}_0R^0\dot{\omega}_0) = 0. $$

$$ {}_1R^0\dot{V}_1 = {}_1R^0(Z_0\ddot{q}_1 + {}_0R^0\dot{V}_0) + ({}_1R^0\dot{\omega}_1)\wedge({}_1R^0P_1^*) + 2({}_1R^0\omega_1)\wedge({}_0R^0Z_0\dot{q}_1) $$
$$ + ({}_1R^0\omega_1)\wedge[({}_1R^0\omega_1)\wedge({}_1R^0P_1^*)] = [0 \; 0 \; \ddot{q}_1+g]^T; \quad car \; \dot{V}_0 = [0 \; 0 \; g]^T. $$

$$ {}_1R^0\overline{a}_1 = ({}_1R^0\dot{\omega}_1)\wedge({}_1R^0\overline{S}_1) + ({}_1R^0\omega_1)\wedge[({}_1R^0\omega_1)\wedge({}_1R^0\overline{S}_1)] + {}_1R^0\dot{V}_1 = [0 \; 0 \; \ddot{q}_1+g]^T. $$

pour i=2 on a:

$$ {}_1R^0\omega_2 = {}_2R^1({}_1R^0\omega_1 + Z_0\dot{q}_2) = [0 \; -\dot{q}_2 \; 0]. $$

$$ {}_2R^0\dot{\omega}_2 = {}_2R^1[{}_1R^0\dot{\omega}_1 + Z_0\ddot{q}_2 + ({}_1R^0\omega_1)\wedge(Z_0\dot{q}_2)] = [0 \; -\ddot{q}_2 \; 0]^T. $$

$$ {}_2R^0\dot{V}_2 = ({}_2R^0\dot{\omega}_2)\wedge({}_2R^0P_2^*) + ({}_2R^0\omega_2)\wedge[({}_2R^0\omega_2)\wedge({}_2R^0P_2^*)] + {}_2R^1({}_1R^0\dot{V}_1) = [0 \; -\ddot{q}_1-g \; 0]^T. $$

$$ {}_2R^0\overline{a}_2 = ({}_2R^0\dot{\omega}_2)\wedge({}_2R^0\overline{S}_2) + ({}_2R^0\omega_2)\wedge[({}_2R^0\omega_2)\wedge({}_2R^0\overline{S}_2)] + {}_2R^0\dot{V}_2 = [0 \; -\ddot{q}_1-g \; 0]^T. $$

pour i=3 on a:

$$ {}_3R^0\omega_3 = {}_3R^2({}_2R^0\omega_2) = [0 \; -\dot{q}_2 \; 0]^T. $$

$$ {}_3R^0\dot{\omega}_3 = {}_3R^0({}_2R^0\dot{\omega}_2) = [0 \; -\ddot{q}_2 \; 0]^T. $$

$$ {}_3R^0\dot{V}_3 = ({}_3R^0\dot{\omega}_3)\wedge({}_3R^0P_3^*) + 2({}_3R^0\omega_3)\wedge({}_3R^0Z_0\dot{q}_3) $$
$$ + ({}_3R^0\omega_3)\wedge[({}_3R^0\omega_3)\wedge({}_3R^0P_3^*)] + {}_3R^2({}_2R^0\dot{V}_2 + Z_0\ddot{q}_3) $$
$$ = [-d_3\ddot{q}_2 - 2\dot{q}_2\dot{q}_3 \quad -\ddot{q}_1-g \quad \ddot{q}_3 - \dot{q}_2^2 d_3]^T. $$

$$ {}_3R^0\overline{a}_3 = ({}_3R^0\dot{\omega}_3)\wedge({}_3R^0\overline{S}_3) + ({}_3R^0\omega_3)\wedge[({}_3R^0\omega_3)\wedge({}_3R^0\overline{S}_3)] + {}_3R^0\dot{V}_3 $$
$$ = [-(\overline{z}_3+d_3)\ddot{q}_2 - 2\dot{q}_2\dot{q}_3 \quad -\ddot{q}_1-g \quad \ddot{q}_3 - (d_3+\overline{z}_3)\dot{q}_2^2]^T. $$

2éme Etape:"backward équations"

Dans cette étape on a $f_4=0$; $\eta_4=0$ *et* ${}_3R^4=I$.

Pour i=3:

$${}_3R^0 f_3 = {}_3R^4 ({}_4R^0 f_4) + m_3 \; {}_3R^0 \overline{a}_3 = m_3 [-(\overline{z}_3 + d_3)\ddot{q}_2 - 2\dot{q}_2\dot{q}_3 \quad -\ddot{q}_1 - g \quad \ddot{q}_3 - (d_3 + \overline{Z}_3)\dot{q}_2^2]^T.$$

$$\begin{aligned} {}_3R^0 \eta_3 &= {}_3R^4 [{}_4R^0 \eta_4 + ({}_4R^0 P_1^*) \wedge ({}_4R^0 f_4)] + ({}_3R^0 P_3^* + {}_3R^0 \overline{S}_3) \wedge (m_3 \; {}_3R^0 \overline{a}_3) \\ &\quad + ({}_3R^0 I_3 \; {}_0R^3)({}_3R^0 \dot{\omega}_3) + ({}_3R^0 \omega_3) \wedge [({}_3R^0 I_3 \; {}_0R^3)({}_3R^0 \omega_3)] \\ &= m_3 [(g + \ddot{q}_1)(\overline{z}_3 + d_3) \quad [-(d_3 + \overline{Z}_3)\ddot{q}_2 - 2\dot{q}_2\dot{q}_3](d_3 + \overline{Z}_3) - \beta_3 \ddot{q}_2 \quad 0]^T. \end{aligned}$$

Pour i=2:

$$\begin{aligned} {}_2R^0 f_2 &= {}_2R^3 ({}_3R^0 f_3) + m_2 \; {}_2R^0 \overline{a}_2 \\ &= [m_3 [-(\overline{z}_3 + d_3)\ddot{q}_2 - 2\dot{q}_2\dot{q}_3] \quad -(m_2 + m_3)(\ddot{q}_1 + g) \quad m_3 [\ddot{q}_3 - (d_3 + \overline{Z}_3)\dot{q}_2^2]]^T. \end{aligned}$$

$$\begin{aligned} {}_2R^0 \eta_2 &= {}_2R^3 ({}_3R^0 \eta_3 + ({}_3R^0 P_2^*) \wedge ({}_3R^0 f_3)) + ({}_2R^0 P_2^* + {}_2R^0 \overline{S}_2) \wedge ({}_2R^0 F_3) \\ &\quad + ({}_2R^0 I_2 \; {}_0R^2)({}_2R^0 \dot{\omega}_2) + ({}_2R^0 \omega_2) \wedge (({}_2R^0 I_2 \; {}_0R^2)({}_2R^0 \omega_2)) \\ &= m_3 [(g + \ddot{q}_1)(\overline{z}_3 + d_3) \quad (-(d_3 + \overline{Z}_3)\ddot{q}_2 - 2\dot{q}_2\dot{q}_3)(d_3 + \overline{Z}_3) - (\beta_2 + \beta_3)\ddot{q}_2 \quad 0]^T. \end{aligned}$$

Pour i=1:

$$\begin{aligned} {}_1R^0 f_1 &= {}_1R^2 ({}_2R^0 f_2) + m_1 \; {}_1R^0 \overline{a}_1 = [m_3 C_2 (-(\overline{z}_3 + d_3)\ddot{q}_2 - 2\dot{q}_2\dot{q}_3) - m_3 S_2 (\ddot{q}_3 - (d_3 + \overline{Z}_3)\dot{q}_2^2) \\ &\quad m_3 S_2 (-(d_3 + \overline{Z}_3)\dot{q}_2\dot{q}_3) + m_3 C_2 (\ddot{q}_3 - (d_3 + \overline{Z}_3)\dot{q}_2^2 (m_1 + m_2 + m_3)(\ddot{q}_1 + g)]^T \end{aligned}$$

${}_1R^0 \eta_1$: à ne pas calculer car la liaison est translationnelle.

Donc

$$F_1 = ({}_1R^0 f_1)^T ({}_1R^0 Z_0) + f_1 \dot{q}_1 = (m_1 + m_2 + m_3)(\ddot{q}_1 + g) + f_1 \dot{q}_1.$$

$$\begin{aligned} \tau_2 &= ({}_2R^0 \eta_2)^T ({}_2R^1 Z_0) + f_2 \dot{q}_2 \\ &= (m_3 \overline{z}_3 d_3 + m_3 d_3 (d_3 + \overline{Z}_3) + m_3 \overline{Z}_3^2 + \beta_2 + \beta_3)\ddot{q}_2 + 2m_3 (d_3 + \overline{Z}_3)\dot{q}_2\dot{q}_3 + f_2 \dot{q}_2. \end{aligned}$$

$$F_3 = ({}_3R^0 f_3)^T ({}_3R^2 Z_0) + f_3 \dot{q}_3 = -m_3 (\overline{z}_3 + d_3)\ddot{q}_2 + m_3 \dot{q}_2^2 + f_3 \dot{q}_3.$$

Ainsi en remplaçant les paramètres nécessaires:

$F_1 = k_1 U_1$; $\tau_2 = k_2 U_2$; $F_3 = k_3 U_3$;

$m_2 \approx 0$; $\overline{z}_3 = \dfrac{l_2}{2}$; $\beta_3 = m_3 \dfrac{l_2^2}{12}$; $\beta_2 = 0$; d'où :

$$\begin{cases} U_1 = \dfrac{(m_1 + m_3)}{k_1}(\ddot{q}_1 + g) + \dfrac{f_1}{k_1}\dot{q}_1. \\ U_2 = \dfrac{m_3 \dfrac{l_2^2}{3} + 2m_3 \overline{z}_3 q_3 + m_3 q_3^2}{k_2}\ddot{q}_2 + +2\dfrac{m_3}{k_2}(\overline{z}_3 + q_3)\dot{q}_3\dot{q}_2 + \dfrac{f_2}{k_2}\dot{q}_2. \\ U_3 = \dfrac{m_3}{k_3}\ddot{q}_3 - \dfrac{m_3 (\overline{z}_3 + q_3)}{k_3}\dot{q}_2^2 + \dfrac{f_3}{k_3}\dot{q}_3. \end{cases}$$

C'est le modèle dynamique du robot.

Annexe 4

RESOLUTION DE L'EQUATION DIOPHANTINE REDUITE

$$A(q^{-1})\,S'(q^{-1}) + q^{-(d+1)}R(q^{-1}) = P_R(q^{-1}) \qquad \text{(Ann4.1)}$$

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

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

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

$P_R(q^{-1}) = 1 + p_1 q^{-1} + \ldots + p_{n+d} q^{-(n+d)}$.

- Le polynôme $P_R(q^{-1})$ est le produit de deux polynômes. Ce produit est calculé avant d'utiliser cet algorithme.
- Le polynôme $A(q^{-1})$ n'est pas forcement celui du système, il peut être le produit du polynôme du système et un autre polynôme introduit par la pondération ou l'intégration.

L'équation (Ann4.1) peut se mettre sous la forme matricielle suivante:

$$M\,x = b \qquad \text{(Ann4.2)}$$

Où $x^T = (1\ s'_1 \ldots s'_d\ r_0 \ldots r_{n-1})$.

$b^T = (1\ p'_1 \ldots p'_{n+d})$.

et $M =$

$$\begin{bmatrix}
1 & 0_1 & & 0_d & & & & 0_{n+d} \\
a_1 & 1 & & & & & & \\
a_2 & a_1 & & & & & & \\
 & a_2 & & 0 & & & & \\
 & & & 1 & 0 & & & \\
 & & & a_1 & 1 & & & \\
a_n & & & a_2 & 0 & & & \\
0_1 & a_n & & & 0 & & & \\
 & & & & & 0 & & \\
 & & & & & 1 & 0 & \\
0_d & & & a_n & 0 & 0 & 1 &
\end{bmatrix}$$

; avec $0_i = 0\ \forall i$.

Après réduction le système devient:

$$\begin{bmatrix} 1 & 0_d & & & 0_{n+d-1} \\ a_1 & & & & \\ & 0 & & & \\ & 1 & 0 & & \\ & a_1 & 1 & & \\ a_n & & 0 & & \\ 0_1 & & & & \\ & & & 0 & \\ & & & 1 & 0 \\ 0_d & a_n & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s'_1 \\ s'_2 \\ \\ \\ s'_d \\ r_0 \\ r_1 \\ \\ \\ r_{n-1} \end{bmatrix} = \begin{bmatrix} p_1-a_1 \\ p_2-a_2 \\ \\ \\ p_n-a_n \\ p_{n+1} \\ p_{n+2} \\ \\ \\ p_{n+d} \end{bmatrix} \qquad \text{(Ann4.3)}$$

La procédure en PASCAL de chargement de la matrice est la suivante:

```
for i:=1 to n+d do
 begin
  for j:=1 to n+d do
   if i=j then M[i,j]:=1 else M[i,j]:=0;
 end;
for j:=1 to d do
 for i:=1 to j+n do
  M[i,j]:=a[i];
Pour le vecteur b on a:
for i:=1 to n do
 b[i]:=p[i]-a[i];
for i:=n+1 to n+d do
 b[i]:=p[i];
```

La Résolution du système est simple, M étant triangulaire inférieure.

Annexe 5

RESOLUTION DE L'EQUATION DIOPHANTINE GENERALE

$$A(q^{-1})S(q^{-1})+B(q^{-1})R(q^{-1}) = P_R(q^{-1}) \qquad \text{(Ann5.1)}$$

avec $A(q^{-1})=a_0+a_1q^{-1}+\ldots+a_{n_a}q^{-n_a}$.
$B(q^{-1})=b_0+b_1q^{-1}+\ldots+b_{n_b}q^{-n_b}$.
$R(q^{-1})=r_0+r_1q^{-1}+\ldots+r_rq^{-r}$.
$S(q^{-1})=s_0+s_1q^{-1}+\ldots+s_sq^{-s}$.
$P_R(q^{-1})=p_{R_0}+p_{R_1}q^{-1}+\ldots+p_{R_p}q^{p}$.

-Les polynômes A et B ne sont pas forcement ceux du système.
- $P_R(q^{-1})=P_o(q^{-1})P(q^{-1})$.

L'écriture matricielle de cette équation donne:

$$\begin{bmatrix} a_0 & 0_s & b_0 & 0_1 & & 0_r \\ a_1 & & b_1 & b_0 & & \\ a_2 & & & b_1 & & \\ & 0 & & & & \\ & a_0 & b_{n_b} & & 0 & \\ & a_1 & 0 & b_{n_b} & b_0 & 0 \\ a_{n_a} & a_2 & 0 & & b_1 & b_0 \\ 0_1 & & 0 & & & b_1 \\ & & & & b_{n_b} & \\ 0_s & a_{n_a} & 0 & & 0 & b_{n_b} \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ \\ \\ s_s \\ r_0 \\ r_1 \\ \\ r_r \end{bmatrix} = \begin{bmatrix} p_{R_0} \\ p_{R_1} \\ \\ \\ p_{R_s} \\ p_{R_{s+1}} \\ p_{R_{s+2}} \\ \\ p_{R_p} \end{bmatrix} \qquad \text{(Ann5.2)}$$

avec $p \le \max(n_a+s+1, n_b+r+1)$.

l'équation (An5.1) admet des solutions R et S si et seulement si le nombre d'inconnus r+s+2 est égale au nombre d'équation $\max(n_a+s+1, n_b+r+1)$. D'où la condition:

$$\max(n_a+s+1, n_b+r+1) = s+r+1 \quad (12) \qquad \text{(Ann5.3)}$$

Cherchons les polynômes réalisant cette condition.

1- Si $n_a+s+1 \ge n_b+r+1$. alors $n_a+s+1=s+r+2 \Rightarrow r=n_a-1$ (*Ann5.4*).
Or $n_a+s+1 \ge n_b+r+1 \Rightarrow n_a+s+1 \ge n_b+n_a-1+1$.
d'où $s \ge n_b-1$ (*Ann5.5*).

2- Si $n_a+s+1 \le n_b+r+1$. alors $n_b+s+1=s+r+2 \Rightarrow s=n_b-1$ (*Ann5.7*).
Or $n_b+r+1 \ge n_a+s+1 \Rightarrow n_b+r+1 \ge n_b+n_a-1+1$.
d'où $r \ge n_a-1$ (*Ann5.7*).

Une condition suffisante pour que le système (Ann5.3) admet des solutions R et S est alors:

$$\begin{matrix} r \geq n_a - 1 \\ s \geq n_b - 1 \end{matrix} \qquad \text{(Ann5.8)}$$

Une réalisation minimale des polynômes R et S est obtenu pour $r=n_a-1$ et $s=n_b-1$.

Remarque.

1- Le choix des degrés de R et s est arbitraire pourvu que la condition (Ann5.8) soit vérifiée.

2- La réalisation choisie pour la commande à PP est [29]:

$$\begin{matrix} g=\max(n_a, n_b)\,. \\ s=r=g-1 \end{matrix} \qquad \text{(Ann5.9)}$$

3- Si le système possède un retard d alors les d premiers coefficients de B seront pris nuls.□

Remplissage de la matrice.

Dans notre programme on a choisi $s_0=1$ et $p_{R_0}=1$, ainsi l'ordre de la matrice est réduit de un, d'où:

$$\begin{bmatrix} a_0 & 0 & 0 & b_1 & b_0 & 0 & & & 0 \\ a_1 & & & & b_1 & & & & \\ a_2 & & 0 & & & & & & \\ & & a_0 & b_{n_b} & & & 0 & & \\ & & a_1 & 0 & b_{n_b} & & b_0 & 0 & \\ a_{n_a} & & a_2 & 0 & & & b_1 & b_0 & \\ 0_1 & & & 0 & & & & b_1 & \\ & & & & & & b_{n_b} & & \\ 0_s & & a_{n_a} & 0 & & & 0 & b_{n_b} & \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ \\ \\ s_s \\ r_0 \\ r_1 \\ \\ r_r \end{bmatrix} = \begin{bmatrix} p_{R_0} \\ p_{R_1} \\ \\ \\ p_{R_s} \\ p_{R_{s+1}} \\ p_{R_{s+2}} \\ \\ p_{R_p} \end{bmatrix} \qquad \text{(Ann5.10)}$$

La procédure en PASCAL de chargement de la matrice est la suivante:

```
 for j:=1 to g-1 do
  begin
   for i:=j to na+j do
    if i=j then M[i,j]:=1 else M[i,j]:=a[i-j];
  end;
 for j:=g to 2g-1 do
  for i:=j-g+d+1 to j-g+d+m do
   M[i,j]:=a[i-j+g-d];
```

pour le vecteur b on a:

```
 for i:=1 to na do
  b[i]:=pR[i]-a[i];
 for i:=n+1 to p do
  b[i]:=p[i];
```

La Résolution du système est faite par la méthode de Jordan avec $a_0=1$; $b_0=0$; $n_b=m+d$ et $g=\max(n_a, n_b)$

Annexe 6

INTERPRETATION DU POLYNOME D'OBSERVATION

Dans cette annexe, On s'intéresse à montrer l'interprétation du polynôme d'observation P_0, introduit dans l'équation Diophantine obtenue par PP et par PRPE. En particulier, on montre que la commande par placement de pôles ou PRPE sont des combinaison d'un observateur et d'un retour d'état, dans l'espace d'état.
Pour se faire, considérant le système défini par l'équation aux différence suivante:

$$A(q^{-1})y(t) = B(q\text{-}1)u(t) \qquad \text{(Ann6.1)}$$

Avec $A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n}$.
$B(q^{-1}) = b_1 q^{-1} + \ldots + b_m q^{-m}$.

mise sous forme d'état de contrôlabilité on a:

$$x(t+1) = \begin{bmatrix} -a_1 & \cdot & \cdot & \cdot & -a_n \\ 1 & 0 & \cdot & \cdot & 0 \\ 0 & 1 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} u(t) \qquad \text{(Ann6.2)}$$

$$y(t) = [b_1 \ldots b_m \; 0 \ldots 0]$$

Où $x^T = [x_1 \ldots x_n]$.

Si les états $x_i(t)$ sont mesurables, on peut considérer le retour d'état suivant [27,30]:

$$u(t) = -k^T x(t) + w(t) \qquad \text{(ann6.3)}$$

Où $k^T = [k_1 \ldots k_n]$ *et* $w(t)$: la référence.

En remplaçant l'équation (Ann6.3) dans (Ann6.2) on trouve:

$$x(t+1) = \begin{bmatrix} -a_1 - k_1 & \cdot & \cdot & \cdot & -a_n - k_n \\ 1 & 0 & \cdot & \cdot & 0 \\ 0 & 1 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} w(t) \qquad \text{(Ann6.4)}$$

ou

$$x(t+1) = \bar{A}\, x(t) + \bar{B} w(t) \qquad \text{(Ann6.5)}$$

ont le polynôme caractéristique est défini par:

$$\det(\bar{A}-zI) = z^n+(a_1+k_1)z^{n-1}+\ldots+(a_n+k_n) \qquad \text{(Ann6.6)}$$

Donc le choix du retour k^T est facile, en imposant seulement le polynôme caractéristique.
Dans le cas où les états ne sont pas mesurables, la commande est définie par:

$$u(t) = -k^T\hat{x}(t)+w(t) \qquad \text{(ann6.7)}$$

où $\hat{x}$ est l'estimé de x.

L'observateur est défini par l'équation au différence suivante [27][30][31]:

$$\hat{x}(t+1) = (\psi-lc)\,\hat{x}(t)+l\,y(t)+\Gamma u(t) \qquad \text{(Ann6.8)}$$

où $l^T=[l_1\ldots l_n]$ et $c=[b_1\ldots b_m\ 0\ldots 0]$;

$$\psi=\begin{bmatrix} -a_1 & \cdot & \cdot & \cdot & -a_n \\ 1 & 0 & \cdot & \cdot & 0 \\ 0 & 1 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad et\ \Gamma=[1\ 0\ldots 0].$$

Le choix du vecteur l, se fait de la même façon, que pour le vecteur k (on passe à la forme canonique d'observabilité). On cherche alors a trouver une équation au différence qui décrit l'évolution de l'état x et de son estimé. Les équation (Ann6.7) et (Ann6.8) donnent:

$$\begin{cases} x(t+1) = \psi x(t)-\Gamma k^T\hat{x}(t)+\Gamma w(t) \\ \hat{x}(t+1) = (\psi-lc-\Gamma k^T)\hat{x}(t)+ly(t)+\Gamma w(t) \end{cases} \qquad \text{(Ann6.9)}$$

sous forme matricielle on a (avec $y=cx$):

$$\begin{bmatrix} x(t+1) \\ \hat{x}(t+1) \end{bmatrix} = \begin{bmatrix} \psi & -\Gamma k^T \\ lc & \psi-lc-\Gamma k^T \end{bmatrix}\begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix}+\begin{bmatrix} \Gamma \\ \Gamma \end{bmatrix} w(t) \qquad \text{(Ann6.10)}$$

De l'expression suivante:

$$\begin{bmatrix} x(t+1) \\ \tilde{x}(t+1) \end{bmatrix} = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix}\begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ x(t)-\hat{x}(t) \end{bmatrix} \qquad \text{(Ann6.11)}$$

et de l'équation (Ann6.10) on trouve:

$$\begin{bmatrix} x(t+1) \\ \tilde{x}(t+1) \end{bmatrix} = \begin{bmatrix} \psi-\Gamma k^T & -\Gamma k^T \\ 0 & \psi-lc \end{bmatrix}\begin{bmatrix} x(t) \\ \tilde{x}(t) \end{bmatrix}+\begin{bmatrix} b \\ 0 \end{bmatrix} w(t) \qquad \text{(Ann6.12)}$$

L'équation caractéristique du système défini par (Ann6.12) est:

$$\Delta(z) = \det(\psi - \Gamma k^T - zI)\det(\psi - lc - zI) \qquad \text{(Ann6.13)}$$

avec $\det(\psi - \Gamma k^T - zI) = z^n P(z^{-1})$.

et $\det(\psi - lc - zI) = z^n P_o(z^{-1})$.

P et P_0 les polynômes définissant respectivement les pôles désirées en BF et le polynôme d'observation désiré. D'où

$$\Delta(z^{-1}) = P(z^{-1})P_o(z^{-1}) \qquad \text{(Ann6.14)}$$

Donc on voit bien que les pôles en boucle en fermée sont bien le produit de deux polynômes (d'observation et de dynamique désirée).

Annexe 7

THEOREMES D'HYPERSTABILITE

On considère le système en boucle fermée de la figure ci-dessous, où le système de la chaîne directe est linéaire et invariant dans le temps. Ce dernier est décrit par l'équation d'état suivante:

$$\begin{cases} \dot{x} = A\,x + B\,u_1 \\ y_1 = c\,x + D u_1 \end{cases} \qquad \text{(Ann7.1)}$$

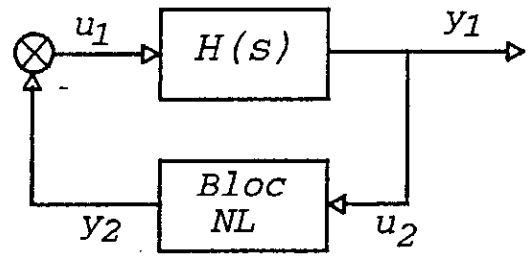


Figure Ann7.1 Schéma d'un retour NL sur une chaîne directe linéaire.

Les pairs (A,B) *et* (A,C) sont respectivement contrôlable et observable. Le système décrit par (Ann7.1) est caractérisé par sa fonction de transfert donnée par :

$$H(s) = D + C(sI - A)^{-1}B \qquad \text{(Ann7.2)}$$

Le bloc nonlinéaire traduisant le retour, est défini par :

$$y_2 = f(u_2, t, \tau) \quad avec\ \tau \le t \qquad \text{(Ann7.3)}$$

Ce dernier vérifie l'inégalité de Popov donnée par :

$$\int_{t_0}^{t_1} y_2(\tau)\,u_2(\tau)\,d\tau \ge -\gamma_0^2 \qquad \text{(Ann7.4)}$$

Avec $\gamma_0^2 < \infty$; $\forall t_1 \ge t_0$.

Théorème 1.

Toutes solutions $x(x(0),t)$, du système décrit par les équations ci dessus, en boucle fermée, sur le bloc nonlinéaire qui vérifie l'inégalité de Popov, satisfait la propriété suivante:

$$\|x\| < \delta[\|x_0\| + \gamma_0] \qquad \text{(Ann7.5)}$$

Où $\delta > 0$, $\gamma_0 > 0$ $\forall t \ge 0$.
si et seulement si la matrice de transfert $H(s)$ est positive réelle.

Définition 1 (Positivité réelle).

Une fonction rationnelle $H(s)$, de la variable complexe $s = \sigma + j\omega$, est positive réelle si:

1- $H(s)$ est réelle pour s réel.
2- $Re[H(s)] \ge 0$, pour tout $Re[s] > 0$.

Théorème 2 (hyperstabilité asymptotique).

Toutes solutions $x(x_0, t)$, du système décrit ci dessus en BF, sur un bloc nonlinéaire, qui vérifie l'inégalité de Popov, satisfait (Ann7.5), de plus nous avons:

$$\lim_{t\to\infty} x(t) = 0 \qquad \text{(Ann7.6)}$$

pour toutes entrées $u_1(t)$ bornée, si et seulement si la matrice de transfert $H(s)$ est strictement positive réelle.

définition 2 (Positivité réelle stricte).

Une fonction rationnelle $H(s)$ de la variable complexe $s=\sigma+j\omega$, est strictement positive réelle si:

1- $H(s)$ est réelle pour s réel.
2- $Re[H(s)]>0$, $\forall\omega$.

Annexe 8

CALCUL DE LA LOI D'ADAPTATION POUR "DMRAC"

L'inegalité à satisfaire est:

$$\sum_{k=0}^{k_1} V^T(k)\, w(k) \geq -\gamma_0^2; \quad \forall k_1 > 0 \tag{Ann8.1}$$

De l'expression de $w(k)$ dans l'equation (VI.65), on tire à partir de l'equation (Ann8.1):

$$\begin{aligned}\sum_{k=0}^{k_1} V^T(k) \big(&\psi_1(k)\,\varepsilon_q(k-1) + \psi_2(k)\,\varepsilon_q(k-2) + \psi_3(k)\,q_m(k) \\ &+ \psi_4(k)\,q_m(k-1) + \psi_5(k)\,q_m(k-2) + \rho(k)\big) \geq -\gamma_0^2\end{aligned} \tag{Ann8.2}$$

En distribuant le signe somme à l'interieur du terme de gauche, on obtient:

$$\sum_{k=0}^{k_1} V^T(k)\,\psi_1(k)\,\varepsilon_q(k-1) + \ \ldots\ + \sum_{k=0}^{k_1} V^T(k)\,\rho(k) \ \geq -\gamma_0^2 \tag{Ann8.3}$$

Il suffit que chaque somme verifie l'inegalité de Popov, c'est à dire:

$$\begin{aligned}&\sum_{k=0}^{k_1} V^T(k)\,\psi_1(k)\,\varepsilon_q(k-1) \geq -\gamma_{0_1}^2 \\ &\qquad\vdots \\ &\sum_{k=0}^{k_1} V^T(k)\,\rho(k) \ \geq -\gamma_{0_6}^2\end{aligned} \tag{Ann8.4}$$

Cherchons la loi d'adaptation pour $\psi_1(k)$ et les autres termes se deduisent directement.
On pose une loi de type PI [88]:

$$\psi_1(k) \ = G\, V(k)\, \varepsilon_q^T(k-1)\, E_{1_p} + G \sum_{l=0}^{k-1} V(l)\, \varepsilon_q^T(l-1)\, E_{1_I} \tag{Ann8.5}$$

avec G, E_{1_p} *et* E_{1_I}: des matrices definies positives.

En remplaçant l'expression de $\psi_1(k)$ definie par l'equation (Ann8.5), dans la première inequation de (Ann8.4), on trouve:

$$\begin{aligned}I_1 = &\sum_{k=0}^{k_1} \Big(V^T(k)\, G\, V(k)\, \varepsilon_q^T(k-1)\, E_{1_p}\, \varepsilon_q(k-1)\Big) \\ &\sum_{k=0}^{k_1} \left(V^T(k)\, G \left(\sum_{l=0}^{k-1} V(l)\, \varepsilon_q^T(l-1)\, E_{1_I}\right) \varepsilon_q(k-1)\right)\end{aligned} \tag{Ann8.6}$$

Où G *et* E_{1_p}: les matrices definies positives, donc le premier terme est positif. En

ce qui concerne le second terme, G et E_{1_I} sont des matrices definies positives et de la même manière, ce terme est positif [88]. Donc l'inegalité de Popov est verifiée.

Ainsi on a determiné la loi d'adaptation de $\psi_1(k)$. Le problème alors n'est pas de determiner cette loi, par contre celle de $P_1(k)$.
De l'equation (VI.65), on a:

$$\psi_1(k) = C_1 - A_0^{-1}(k)\left[A_1(k) + P_1(k)\right] \qquad \text{(Ann8.7)}$$

Des equations (Ann8.7) et (Ann8.5), en posant $G = A_0^{-1}(k)$, on trouve:

$$V(k)\,\varepsilon_q^T(k-1)\,E_{1_p} + \sum_{l=0}^{k-1} V(l)\,\varepsilon_q^T(l-1)\,E_{1_I} = A_0(k)\,C_1 - A_1(k) + P_1(k) \qquad \text{(Ann8.8)}$$

En imposant $-V(k) = \varepsilon_q^T(k)$, on trouve:

$$\begin{aligned} P_1(k) &= A_0(k)\,C_1 - A_1(k) + \hat{\varepsilon}_q(k)\,\varepsilon_q^T(k-1) \\ &\quad + \sum_{l=0}^{k-1} \hat{\varepsilon}_q(l)\,\varepsilon_q^T(l-1)\,E_{1_I} \end{aligned} \qquad \text{(Ann8.9)}$$

On pose $P_1(0) = A_0(k)\,C_1 + A_1(k)$.
On peut ecrire (en ajoutant et en retranchant un terme):

$$\begin{aligned} P_1(k) &= P_1(0) + \hat{\varepsilon}_q^T(k-1)\,\varepsilon_q^T(k-2)\,E_{1_p} + \sum_{l=0}^{k-2} \hat{\varepsilon}_q(l)\,\varepsilon_q^T(l-1)\,E_{1_I} \\ &\quad - \hat{\varepsilon}_q(k-1)\,\varepsilon_q^T(k-2)\,E_{1_p} + \hat{\varepsilon}_q(k-1)\,\varepsilon_q^T(k-2) + \hat{\varepsilon}_q(k)\,\varepsilon_q^T(k-1)\,E_{1_p} \end{aligned} \qquad \text{(Ann8.10)}$$

De l'equation (Ann8.9), on a:

$$P_1(k) = P_1(k-1) + \hat{\varepsilon}_q(k)\,\varepsilon_q^T(k-1)\,E_{1_p} + \hat{\varepsilon}_q(k-1)\,\varepsilon_q^T(k-2)\left[E_{1_I} - E_{1_p}\right] \qquad \text{(Ann8.11)}$$

De la même manière on peut deduire toute les lois d'adaptation des autres paramétres.

REFERENCES BIBLIOGRAPHIQUES

"Savoir ce que l'on sait ce que l'on sait
et
ce que l'on ne sait pas ce que l'on ne sait pas,
voilà la véritable science".

CONFUCIUS

REFERENCES

BIBLIOGRAPHIQUES

[1] K.S.Fu & Al 'Robotics: Control, Sensing, Vision and intelligence', Mcc Graw Hill, 1987.

[2] C.Vibet 'Robots: principes et controle' Ellipses, 1987.

[3] H.Asada & J.E.Slotine 'Robot analysis and control', MIT, John Wiley and Sons, 1986.

[4] D.Stoten 'Modelling and control of elastic joint robots' ASME, Journal of dynamics system Measurement and control, vol 109, pp310-319, 1987.

[5] J.Denavit & Al 'A kinematic notation for lower pair mechanisms based on matrices' Journal of applied mechanics, pp215-221, June 1955.

[6] P.P.Paul & Al 'Kinematic control equations for simple manipulators' IEE trans on System, Man and cybernetics, Vol SMC 11, N°6, pp449-455, June 1981.

[7] P.Lopez & J.N.Foulc 'Introduction a la robotique' 2 tomes, Editest, Paris 1984.

[8] R.P.Paul 'Robot manipulators: Mathematics, Programming and control', MIT press 1981.

[9] W.W.Seto 'Theory and problem of mechanical vibrations' Shaum's outline series. Macc Graw Hill, 1964.

[10] N.A.Rahim 'Adaptive control of robot manipulators' thése doctorat ' University of Leeds departement of mechanical Engineering', Aout 1987.

[11] Y.Koren 'Robotics for enginneers' Macc Graw Hill, 1985.

[12] L.Guenfaf & N.Bali & M.S.Boucherit 'Identification récursive des systèmes multivariables: Etude de différentes structures' 2nd international meeting on components and Electronic systems, Sidi Bel Abbes, IMCES 2, 1994.

[13] K.J.Aström 'Introduction to stochastic control theory' Academic Press, 1970.

[14] K.J.Aström & B.Wittenmark 'On self tunnig regulators' Automatica, vol 9, pp 195-199, 1973.

[15] P.E.Wellstead & Al 'Pole assignement self tuning regulator' Proc IEE, Vol 126, N°8, pp 781-789, August 1989.

[16] D.W.Clarke & P.J.Gawthrop 'Self tuning controller' Proc IEE, Vol 22, N°9, pp 929-934, Sept 1975.

[17] D.W.Clarke & Al 'A generalised LQG approach to self tuning contol. Part I, Aspects of design' Int. J. Control, Vol 41, N°6, pp 1509-1529, 1985.

[18] H.Nichelson & B.H.Swanick 'Self tuning and adaptive control: theory and application' Peter Peregrinus Ltd, 1981.

[19] K.J.Aström & Al 'Theory and application of self tuning regulators' Automatica, Vol 13, pp 457-476, 1977.

[20] M.M'Saad & Al 'Adaptive controllers for discrete time systems with arbitrary zeros: An overview' Automatica vol 21,N°4, pp 413-423, 1985.

[21] J.J.Craig 'Introduction to robotics: Mechanics and control' Addison-wesly, 1986.

[22] Mei-Hua Liu & Wei lin 'Pole assignemnet self tuning controller for robotic' Int. J. Control, Vol 46, N°4, pp 1307-1317, 1987.

[24] I.D.Landau & L.Dugard 'Commande adaptative: Aspects pratiques et théorique' Masson, 1986.

[25] Wellstead and Zarrop 'Self tuning systems. Control and signal processing' John Wily and Sons, 1991.

[26] I.D.Landau 'Model reference adaptive controllers and stochastic sel tunning regulators- A unified approach.' Trans of ASME, JMC dynamics systems, Vol 103, pp 404-417, Dec 1981.

[27] G.C.Goodwin & Kwai Sang Sin 'Adaptive filtering prediction and control' Prentice Hall, 1984.

[28] D.W.Clarke & P.J.Gawthrop 'Self tuning control' Proc IEE, vol 126, N°6, pp 633-640, June 1979.

[29] I.D.Landau 'Identification et commande des systemes' Masson, 1988.

[30] K.J.Aström and B.Wittenmark 'Computer controlled systems:theory and design' Prentice Hall, 1990.

[31] **B.Friedland** 'Control system design: An introduction to state space methods' Macc Graw Hill, 1987.
[32] **K.J.Åström** 'Self tunig regulators: Design principles and application.' in 'Application of adaptive control' K.S.Narendra & R.V.Monopoli, Academic Press, 1980.
[33] **O.L.R.Jacobs** 'Introduction to adaptive control' IEE control Eng, serie 15, H.Nichelson & B.H.Swanick 'Self tuning and adaptive control: Theory and application' Academic Press, 1981.
[34] **G.C.Goodwin & Al** 'Discrete time stochastic adaptive control' SIAM J.Control and Optimization, vol 19, N°6, pp 828-853, Nov 1981.
[35] **P.J.Gawthrop** 'Some interpretations of self tuning controller' proc IEE, vol 124, N°10, pp 889-894, Oct 1977.
[36] **V.V.Chalam** 'Adaptive control systems: Techniques and applications' Marcel Dekker, 1987.
[37] **D.W.Clarke** 'Self tuning control of non-minimum phase systems' Automatica, vol 20, N°5, pp 501-517, 1984.
[38] **D.W.Clarke** 'Introduction to self tuning controller' IEE, Control Eng Serie 15, H.Nichelson & B.H.Swanick 'Self tuning and adaptive control: theory and application' Peter pregrinus, 1981.
[39] **K.J. Åström and B.Wittenmark** 'Adaptive control' Addison Wisley, 1989.
[40] **K.J.Åström & Al** 'Self tuning controllers based on poles placement design' Report of Lund Institute of Technology, dept. Auto. Contr., May 1978,
[41] **S.C.Puthempura & J.F.Macgregor** 'Pole zero placement controllers and self tuning reguletors withs better set point tracking' IEE proc, vol 134, Pt D, N°1, pp 26-30, Junuary 1987.
[42] **D.W.Clarke** 'Model following and poles placement self tuners' Optimal Control, Applications and methods, vol 3, pp 323-335, 1982.
[43] **J.J.Åström** 'Theory and methods of adaptive control: A survey' Automatica, Vol 19, N°5, pp 471-485, 1983.
[44] **M.Athans** 'The role and the use of stochastic linear quadratic Gaussian problem in control system design' IEEE Trans on Automatic Control, vol AC 16, N°6, pp 529-552, Dec 1971.
[45] **M.Athans & P.L.Falb** 'Optimal control: an introduction to the theory and its applications' Macc Graw Hill, 1966.
[46] **A.G.Thompson & Al** 'An optimal linear active suspension with finite road preview' Society of Auto motive Engineer Inc, paper N°800520, 1980.
[47] **Z.VI.Arstein & A.Leizarowitz** 'Tracking periodic signals with the overtaking criterion' IEEE Trans. Aut. Cont., vol AC 30, N°11, pp 1123-1126, Nov 1985.
[48] **A.Leizarowitz** 'Tracking non periodic trajectories with the overtaking criterion' Applied mathematics and optimization, Springer Verlag, New York, 14, pp 155-161, 1986.
[49] **N.Louam** 'The application of linear optimal control theory to the design active automotive suspension' PhD Thesis, Leeds universsity, UK, 1990.
[50] **H.T.Toivonen** 'Multivariable adaptive control' Modeling Identification and Control, vol. 5, N°5, pp 19-45, 1984.
[51] **K.Najim & M.M'Saad** 'Adaptive control:theory and practical aspects' IEE. Proc. contr., vol. 1, pp 84-95, March 1991.
[52] **H.Kwakernaak & R.Siran** 'Linear optimal control systems' John Willery and sons, 1972.
[53] **H.Büler** 'Réglages échantillonnés, volume 2:traitement dans l'espace d'état' Press Polytechniques Romandes, 1983.
[54] **M.J.Grimble** 'Implicit and explicit LQG self tuning contollers' Automatica vol. 20, N°5, pp 661-669, 1984.
[55] **C.Berger** 'New Pole-placement design method for adaptive contollers' IEE Proc. vol. 129, pt D, N°1, pp 13-14, January 1982.
[56] **K.Najim & M.Muratet** 'optimisation et commande en génie des procédés' Masson, 1987.
[57] **G.C.Goodwin & Al** 'Discrete time multivariable adaptive control' SIAM J. cont. and optimization, vol 19, N°6, pp 829-853, Nov 1981.

[58] **M.M'Saad & G.Sanchez** 'Partial state reference model adaptive contol of multivariable systems' Automatica, vol. 28, N°6, pp 1189-1197, 1992.
[59] **Chih-Min Lin & Al** 'Adaptive controller with desired pole/zero assignement' IEE Proc. vol. 133, pt. D, N°6, pp 301-306, Nov 1986.
[60] **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.
[61] **C.Y.Chan & H.R.Sirisena** 'Convergence of adaptive pole zero placement contoller for stable non-minimum phase systems' Int. J. Cntrol, Vol. 50, N°3, pp 743-754, 1989.
[62] **M.Kinaert & V.Blondel** 'Discret-time pole placement with stable controller' Automatica, Vol. 28, N°5, pp 935-943, 1992.
[63] **K.J.Aström** 'Robustness of a design method based on assignment of poles and zeros' IEEE trans. auto. contr., Vol. 25, N°3, June 1980.
[64] **A.P.Sage** 'Optimum systems control' Prentice Hall, 1968.
[65] **Ulf Borison** 'Self tuning regulators for class of multivariable systems' Atomatoca, Vol. 15, pp 209-215, 1979.
[66] **H.N.Koivo** 'A multivariable self tuning contoller' Automatica, Vol. 16, pp 351-366, 1980.
[67] **R.Scatollini & D.W.Clarke** 'Multivarible model-following self tuning control with offset rejection' INT. J. Control, 1985, Vol. 42,N°6, 1309-1322.
[68] **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.
[69] **Shi-Jun Lang & Al** 'A multivariable generalized self tuning controller with dacoupling design' IEEE, Trans. Automatic Control, Vol. 31, N°5, pp 474-477, May 1986.
[70] **M.Li De La Sen** 'A model reference adaptive control system for discret multivariable bilinear systems interconnected sudsystem' IEE proc., Vol. 133, pt D, N°4, pp 165-171, July 1986.
[71] **D.L.Prager & P.E.Wellstead** 'Multivariable pole-assignement self tuning regulators' IEE, proc., Vol. 128, pt. D, N°1, pp 9-18, Junuary 1980.
[72] **H.R.Sirisina & F.C.Teng** 'Multivariable pole-zero placement self tuning controllers' Int. J.System. SCI., Vol. 17, N°2, pp 345-352, 1986.
[73] **M.O.Tade & Al** 'Adaptive decoupling of a class of Multivariable dynamic systems out put feedback' IEE proc., Vol. 133, pt. D, N°6, pp 265-275, Nov 1986.
[74] **Zi-Li Deng & Xian-Ri Huang** 'Multivariable decoupling pole assignament self tuning feed forward controller' IEE, proc., pt. D, Vol 138, N°1, pp 85-88, junuary 1991.
[75] **L.Dugard & Al** 'The role of the interactor matrix in multivariable stochastic adaptive control' Automatica, Vol. 20, N°5, pp 701-709, 1984.
[76] **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.
[77] **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.
[78] **B.Bouzouia** 'Adaptive control for robot manipulator: the self tuning approach' Rappot interne, LAAS, Toulouse, 1991.
[79] **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.
[80] **G.A.Montague & Al** 'Performance evaluation of three multivariable self tuning controller design techniques' IEEE, proc. of 25th conference on decision and control, pp 1564-1569, 1986.
[81] **P.J.Gawthrop** 'Hybrid self tuning control' IEE proc., Vol. 125, pt D, N°5, Sept 1980.
[82] **A.Isidori** 'Nonlinear control systems: A introduction' Springer Verlag, 1985.
[83] **M.A.Hensen & D.E.Seborg** 'A critique of differential geometric control strategies for process control' 4th IFAC congress, pp 1-8.
[84] **L.C.J.M.Gras & H.Nijmeijer** 'Decoupling in nonlinear systems: from linearity to nonlinearity' IEE proc., vol. 136, pt. D, N°2, pp 53-62, March 1989.

[85] **S.H.Sastry & A.Isodori** 'Adaptive control of linearizable systems' IEEE trans. on auto. contr., vol. 34, N°11, pp 1123-1131, 1989.
[86] **LE.Elsgolc** 'Calculus of variation' Pergamon Press, 1961.
[87] **J.M.Dion & L.Dugard** 'Commande adaptative multivariable: quelques resultats théoriques et pratiques' APII 20, pp 337-358, 1986.
[88] **I.D.Landau** 'Adaptive control: the model reference approch' Marcel Dekker 1979.
[89] **N.K.M'Sirdi** 'Modélisation, analyse et commande de processus: Application à la commande de systèmes robotisés' Genie robotique et productique, LRP, Université de Versailles St Quentin (UVSQ), 1993.
[90] **J.J.Craig & Al** 'Adaptive control of mechanical manipulators' International Journal of robotics research, Vol. 6, N°2, pp 16-29, Summer 1987.
[91] **J.J.Craig** 'Adaptive control of mechanical manipulators' Adisson Wesley, 1988.
[92] **H.Blighuis & Al** 'A robust adaptive controller for robot manipulators' Proceding of IEEE, International conference of robotics and automation, pp 1876-1881, May 1992.
[93] **M.Benallegue** 'Contribution à la commande dynamique adaptative des robots manipulateurs rapides' Thése de Doctorat de l'université de Paris 6, UPMC, 1991.
[94] **H.Tayebi** 'Commande linearisante adaptative, commande dynamique robuste, commande adaptative basée sur la passivité' Rapport DEA de robotique, LRP/CEMAGREF, 1993.
[95] **V.V.M.Popov** 'L'hyperstabilité des systèmes automatiques' Dunod 1973.
[96] **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.
[97] **B.Brogliato & I.D.Landau & R.Lozano-Leal** 'Adaptive montion control of robot manipulators: A unified approach based on passivity" International Journal of robust and nonlinear control, vol. 1, pp 197-202, 1991.
[98] **I.D.Landau** 'A survey of model reference adaptive techniques: theory and applications' Automatica, pp 353-379, 1974.
[99] **P.C.Parcks** 'Liapunov redesign of model reference adaptive control systems' IEEE trans. Aut. Contr., vol. 11, N°3, pp 362-367, July 1966.
[100] **I.D.Landau** 'Analyse et synthése des commandes adaptatives à l'aide d'un modèle par des méthodes d'hyperstabilité' Automatisme, Tome XIV, N°7-8, pp 301-309, juillet-aout 1969.
[101] **I.D.Landau** 'A hyperstability criterion for model reference adaptive control systems' IEEE, Trans. Aut. Cont., pp 552-555, October 1969.
[102] **B.D.Riedle & P.V.Kokotovic** 'Stability analysis of an adaptive system with unmodelled dynamics' Int. J. Cont., vol. 41, N°2, pp 389-402, 1985.
[103] **I.D.Landau & H.M.Silveira** 'A stability theorem with application to adaptive control' IEEE, trans. on Aut. cont., vol. 24, N°2, pp 305-312, April 1979.
[104] **I.D.Landau** 'Elimination of real positivity condition in the design of parrrallel MRAS' IEEE Trans. Aut. Cont., vol. 23, N°6, pp 1015-1020, December 1978.
[105] **R.M.Johnstone & Al** 'Experimental evaluation of hyperstable model reference adaptive control' in 'Application of adaptive control' K.S.Narendra & R.V.Monopoli, Academic Press, 1980.
[106] **D.P.Stoten** 'The adaptive control of manipulator arms' Mechanism and machine theory, vol. 18, N°4, pp 283-288, 1983.
[107] **D.P.Stoten** 'Generalized manipulator dynamics with regard to model reference adaptive control' Int. J. Cont., vol. 50, N°6, pp 2249-2268, 1989.
[108] **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.
[109] **D.P.Stoten & H.Benchoubane** 'Robustness of minimal controller synthesis algorithm' Int. J. Cont., vol. 51, N°4, pp 861-881, 1990.
[110] **H.Benchoubane & D.P.Stoten** 'Convergence rates of an adaptive control algorithm with application to the speed control of DC machine' IEEE, IECON 1990.
[111] **K.S.Narendra & Y.H.Lin** 'Design of stable model reference adaptive controllers' in 'Application of adaptive control' K.S.Narendra & R.V.Monopoli, Academic Press 1980.
[112] **C.P.Neuman & R.L.Morris** 'Classical control interpretation and design of microcomputer adaptive controllers' in 'Application of adaptive control' K.S.Narendra

& R.V.Monopoli, Academic Press 1980.
[113] **D.P.Stoten** 'Discrete adaptive control of a manipulator arm' Optimal Control Application and Methods, vol. 3, pp 423-433, 1982.
[114] **M.Tarokh** 'A discrete time adaptive control scheme for robot manipulators' Journal of robotics systems, 782, pp 145-166, 1990.
[115] **J.Ritonja & Al** 'Adaptive control structures of dynamic system', Yogoslavia, University of Maribor, 1989.
[116] **I.Bar-Kana & H.Kaufman** 'Robust simplified adaptive control for class of multivariable continious time systems' Proceeding of 24th conference on Decision and Control, IEEE, pp 141-146, 1985.
[117] **K.Sobel & Al** 'Implicit adaptive control for a class of MIMO systems' IEEE trans on aerospace and electronics systems, vol AES 18, N°5, Sep 1982.
[118] **G.Bartolini & A.Ferrara** 'On reduced order model reference adaptive control' pp 312-317, Italy 1990.
[119] **C.M.Liaw** 'Modified linear model following controller for current source inverter-fed induction motor drives' IEE proc, vol 137, pt D, N°1, pp 49-56, Junuary 1990.
[120] **G.C.Goodwin & D.Q.Mayne** 'Continious time stochastic model reference adaptive control' IEEE Tran. Aut. Cont., vol. 36, N°11, pp 1254-1263, Nov 1991.
[121] **S.P.Meyn & O.Brown** 'Model reference adaptive control of time varying and stochatic systems' IEEE trans. Aut. Cont., Vol. 38, N°12, pp 1738-1753, Dec 1993.
[122] **K.S.Tsakalis & P.A.Ioannou** 'Adaptive control of linear time varying plants' Automatica, vol. 23, N°4, pp 459-468, 1987.
[123] **K.S.Tsakalis & P.A.Ioannou** 'Adaptive control of linear time varying plants: A new controller structure' American control conference 87, WP6.3:30, pp 513-588, 1987.
[123] **K.S.Tsakalis & P.A.Ioannou** 'Adaptive control of linear time varying plants: A new model reference controller structure' IEEE Trans. Auto. Contr., vol. 34, N°10, pp 1038-1046, Oct 1989.
[125] **A.Balestrino & al** 'Nonlinear adaptive model-following control' Automatica, vol 20, N°5, pp 559-568, 1984.
[126] **D.Rothchild & A.Jamson** 'Comparaison of four numerical algorithms for solving the Liapunov matrix equation' Int. J. Control. vol. 11, N°2, pp 181-198, 1970.