

Thèse

M0002/95B

Présentée par:

BALI NOUREDDINE
Ingenieur d'état en Automatique de l'ENP

Pour obtenir le titre de

MAGISTER
en AUTOMATIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Sujet:

ETUDE DES PERFORMANCES
DE LA
COMMANDE PREDICTIVE GENERALISEE
"Application aux robots manipulateurs PUMA et SCARA"

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

B. DERRAS
M. S. BOUCHERIT
N. K. M'SIRDI

Président

Rapporteurs

F. BOUDJEMA
R. ILLOUL
M. C. SOUAMI

Examineurs

D. BOUKHETALA

Invité

المدرسة الوطنية المتعددة التقنيات
المطبخية — BIBLIOTHEQUE
Ecole Nationale Polytechnique

بسم الله الرحمن الرحيم
الحمد لله رب العالمين
والصلاة والسلام على
سيدنا محمد وآله الطيبين
الطاهرين

والله اعلم
بما كنا نعتق
منه
صلى الله عليه وسلم
وآله الطيبين الطاهرين



A ma mère,

A la mémoire de mon père et mon frère Mourad,

A ma femme Bahia et à ma fille Amina,

A mon ami Guenfaf Lakhdar,

A Tati,

A mes frère et soeurs, en particulier Larbi et Saliha,

A mes Beaux Parents,

A tous mes amis en particulier Hallet Farid et sa femme,

A toute ma famille,

Je dédie ce travail.

AVANT-PROPOS



Cette Thèse a été préparée au Laboratoire d'Automatique, et au Laboratoire des Systèmes Intelligents Autonomes (LARSIA) de l'Ecole Nationale Polytechnique.

Tous mes remerciements iront tous d'abord à Mesieurs M.S. BOUCHERIT et N.K. M'SIRDI (LRP-FRANCE) qui ont dirigé mes travaux, et dont l'aide efficace et précieuse m'a permis de mener à bien ce travail, qu'ils trouvent ici le signe de ma reconnaissance et de ma gratitude.

Je remercie vivement Mr B.Derras pour l'honneur qu'il me fait d'être le président du jury.

Je tiens à remercier également:

- Mr F.Boudjema pour ses conseils et pour l'intérêt qu'il a porté à ce travail.
- Mr R.Illoul qui a montré un grand intérêt à ce travail et pour les discussions intéressantes.
- Mr M.C.Souami d'avoir accepté de juger ce travail.

Je remercie aussi D.Boukhetala d'avoir examiné ce travail et d'avoir accepté l'invitation.

J'adresse mes sincères remerciements à: Lounici Abd el hakim, Madjouj Nawel et El Hadi (du centre de calcul) qui ont contribué à nous faciliter l'acquisition du matériel informatique.

J'exprime toute ma reconnaissance à: H.Tayebi, M.Bouri, F.Dari, F.Dehimi, R.Hdjar, A.Laroui, M.Djemai, B.Kacha, M.Taghi, Pour toute la bibliographie qui m'est parvenue pour réaliser ce travail.

Que tous les membres de l'équipe de la BIBLIOTHEQUE De l'ENP à savoir: Amara, Ourari, Salah, Karim, Krimo, Abdelkrim, Mourad, Hafida, Chrifa, Ouahiba, Naima, Farida et Ghania sans oublier celles des périodiques, reçoivent le témoignage de ma reconnaissance et de mon respect.

Je remercie vivement mon cousin Bali Abderrahim, Reda Yedou, Arezki Ouhachi, mon beau frère Zeboudji Lyes, Hadjout Larbi, Ouadou Nacer et Mouzali Djamel pour leur soutien moral et leurs encouragements.

Je ne saurai oublier dans mes remerciements Mr Noureddine Berrou pour m'avoir fait le tirage de cette thèse.

J'aimerais remercier très particulièrement Madame Sebaibi pour son respect et 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 près ou de loin à l'aboutissement de ce travail.

SOMMAIRE

Introduction générale



Chapitre 1

Analyse des performances de la commande prédictive généralisée.

Introduction	3
1.1 Algorithmes de commande prédictive	3
1.2 Modélisation et commande	5
1.2.1 Modèle linéaire discret	6
1.2.2 Type de modèle de prédiction utilisé	7
1.3 Commande prédictive généralisée GPC	7
1.4 Synthèse de la GPC	9
1.4.1 Modélisation	9
1.4.2 Prédiction de la sortie	10
1.4.3 Résolution itérative de l'équation Diophantine	11
1.4.4 Lois de commande prédictive	12
1.4.5 Critère de commande	13
1.4.5 Détermination du prédicteur à J-pas	13
1.4.7 Introduction des horizons N_1, N_2, N_U	16
1.5 Structure de réglage RST	16
1.6 Résultats de simulation	17
1.6.1 Etude d'un système stable en boucle ouverte	18
1.6.2 Etude d'un système à phase non-minimale	24
1.6.3 Etude d'un système instable en boucle ouverte	27
1.6.4 Etude d'un système instable et à phase non-minimale	29
1.6.5 Influence de la sortie à une perturbation en régime établi..	32
1.7 Synthèse et discussion	35
Conclusion	37

Chapitre 2

Extension de la commande prédictive généralisée.

Introduction	38
2.1 Critère de commande	39
2.2 Sortie auxiliaire	39
2.3 Calcul du prédicteur	39
2.4 Filtrage de la sortie et de la commande	40
2.5 Mise sous forme matricielle du prédicteur	41
2.6 Détermination du régulateur RST	42
2.7 Equivalence entre la GPC et la GMV	43
2.8 Résultats de simulation	44
2.8.1 Modèle de perturbation	51
2.8.2 Influence du retard et de la surparamétrisation sur la GPC ..	53
conclusion	56

Chapitre 3

Commande prédictive généralisée adaptative

Introduction	57
3.1 Différentes techniques de commande adaptative	58
3.1.1 Commande adaptative à gain programmé	58
3.1.2 Commande à modèle de référence	58
3.1.3 Réglage auto-ajustable	59
3.1.4 Commande duale	60
3.2 Identification	60
3.3 type de modèle utilisé	61
3.4 Identification récursive	61
3.4.1 Formulation du problème	62
3.4.2 Expression du critère	63
3.4.3 Algorithme des moindres carrés récursif	63
3.4.4 Initialisation des paramètres	64
3.4.5 Cas stochastique	64
3.4.6 Variantes de l'algorithme des MCR	65
3.5 Principe d'équivalence certaine	66
3.6 Commande prédictive généralisée adaptative: <u>Approche indirecte</u>	67
3.7 Commande prédictive généralisée adaptative: <u>Approche directe</u>	68
3.8 Résultats de simulation	71
3.8.1 Influence de la variation paramétrique	71
3.8.2 Influence de la surparamétrisation	76
3.8.3 Influence de la sousparamétrisation	77
3.8.4 Influence du retard	82
3.8.5 Simulation de la GPC directe	84
Conclusion	87

Chapitre 4

Modélisation de deux robots manipulateurs

Introduction	89
4.1 Présentation des robots	89
4.2 Modélisation cinématique des deux robots	90
4.2.1 Le robot PUMA	91
4.2.2 Le robot SCARA	93
4.3 Modélisation géométrique des deux robots	93
4.4 Génération de trajectoire	95
4.5 Modélisation dynamique des robots	98
4.5.1 Formalisme d'Euler-Lagrange	98
4.5.2 Résultats de simulation	102
4.5.3 Formalisme de Newton-Euler	106
4.5.4 Résultats de simulation	108
4.6 Validation des deux modèles	109
Conclusion	112

Chapitre 5



Application de la GPC aux bras manipulateurs PUMA et SCARA

Introduction	113
5.1 Modèle paramétrique des systèmes multivariable	114
5.2 Modèle de représentation	115
5.3 Commande multivariable découplée	117
5.4 Algorithme à GPC multivariable	117
5.5 Avantage de la commande multivariable découplée	119
5.6 Résultats de simulation	120
PARTIE I: Résultats de simulation du robots PUMA	120
PARTIE II: Résultats de simulation du robots SCARA	137
Conclusion	144

Conclusion générale.

Annexes.

Références Bibliographiques.

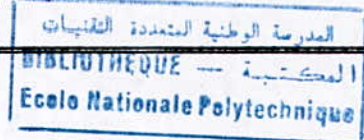
INTRODUCTION GENERALE

*"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

Etude des performances de la commande prédictive généralisée

Application aux robots (PUMA et SCARA)



INTRODUCTION GENERALE.

Depuis 1959, la commande par ordinateur a fait de grands progrès, tant sur le plan matériel que sur le plan méthode [34][35]. Le développement extraordinaire des microprocesseurs a provoqué des changements importants dans la conception des systèmes de commande. Leurs performances et leur faible coût les rendent aptes à prendre intégralement en charge les aspects commande avec des performances nettement supérieures à celles des régulateurs analogiques [10][11].

Pour tirer réellement profit des capacités des microprocesseurs, il ne suffit pas de reproduire le comportement des régulateurs classique analogique, mais il faut mettre en oeuvre des techniques d'automatique spécifiques et plus performantes, développées pour la commande par ordinateur.

Vers la fin des années 80, l'un des derniers algorithmes de commande prédictive est né, à savoir, l'algorithme de commande prédictive généralisée (GPC: Generalized Predictive Control) [1][2][9][16][69]. Cette stratégie de commande appartient à la famille des algorithmes à horizon étendu [10][17]. En effet, la sortie d'un système n'est plus prédite à un instant proche du retard mais à des instants futurs, relatifs à un certain intervalle choisi [1].

L'objectif de cette loi de commande prédictive consiste à prendre en compte, à l'instant présent le comportement futur, en utilisant explicitement un modèle numérique du système afin de prédire dans le futur sur un horizon fini. Cependant, il n'existe pas une stratégie unique, mais plutôt tout un ensemble de méthodes de commande prédictive, assez similaire, bâties autour de principes connus, mais présentant néanmoins quelques différences dans l'interprétation des concepts clés [12].

Une des richesses de ces méthodes provient du fait que, pour une consigne connue ou précalculée (au moins sur un certain horizon), il est ainsi possible d'exploiter pleinement les informations de trajectoires prédéfinies situées dans le futur. En effet, le but de la stratégie prédictive sera de faire coïncider la sortie du processus avec cette consigne dans le futur, sur un horizon fini [2]. C'est pourquoi la GPC apparaît toute indiquée dans les problèmes de poursuite et plus spécialement de suivi de trajectoire. C'est le cas dans la commande des robots manipulateurs où les trajectoires à suivre sont parfaitement connues [64].

La détermination de la loi de commande est basée sur la minimisation d'un critère quadratique multipas au sens des moindres carrés [2][9].

Deux versions de la GPC ont été étudiées, à savoir, l'algorithme de base et son extension [2]. La première version consiste à étudier les performances du processus en faisant varier les paramètres de synthèse de l'algorithme. Par le choix de ces paramètres on peut réaliser un 'placement de pôles' aléatoire, ce qui nous entraîne à étudier la deuxième version [16][25]. Cette dernière réalise une extension de l'algorithme en faisant augmenter le nombre de paramètre de synthèse. Un de ces paramètres permet de spécifier le comportement en boucle fermée selon le choix de l'utilisateur. Donc la GPC présente les avantages des méthodes à placement de pôles

tout en étant beaucoup plus robuste et ceci grâce à sa possibilité de maîtriser la dynamique du système en boucle fermée.

Cette technique de commande prédictive est bien adaptée à résoudre les problèmes surgissants lors de la commande des processus. En effet, la variation de l'ordre, du retard et des paramètres peuvent être résolus en introduisant la version adaptative de la GPC [2][35].

Lorsqu'une estimation en ligne des paramètres du système à régler est nécessaire, la GPC est une méthode qui vérifie l'un des principes de la commande adaptative en l'occurrence celui de compatibilité. En effet, la GPC minimise un critère quadratique multiple compatible avec les méthodes d'identification basées sur les moindres carrés. Le principe de la compatibilité explique que l'amélioration d'une identification minimisant un critère quadratique sur la sortie et d'une commande minimisant ce même type de critère sur la sortie, en écart par rapport à une consigne, donne de bons résultats dans le cadre de la commande adaptative. En effet, aussitôt que le premier type de critère diminue le second s'améliore, ceci signifie que l'effet de la commande est très vite satisfait.

La synthèse des régulateurs adaptatifs peut être faite de deux manières. La première consiste à identifier en permanence les paramètres du système et de les utiliser pour le calcul des régulateurs, c'est l'approche Indirecte [2]. La deuxième se traduit par l'identification directe des paramètres du régulateur, c'est l'approche Directe [39].

Une des parties intéressantes dans ce travail est l'application de la commande prédictive généralisée à deux robots manipulateurs (PUMA et SCARA) pour la poursuite d'une trajectoire [44][45].

Un modèle de connaissance basé sur les lois physiques régissant l'ensemble de la structure des deux robots est élaboré. Ce modèle est obtenu par l'utilisation des formalismes de Lagrange-Euler et de Newton-Euler aboutissant à un modèle dynamique du manipulateur [43][48]. Ce modèle est fortement non linéaire [43].

Des hypothèses simplificatrices sont introduites pour obtenir un modèle réduit capable de reproduire le comportement physique du système considéré [45].

Pour pouvoir appliquer la loi de commande GPC, qui a été développée sous la base d'un modèle linéaire, à un modèle de connaissance non linéaire, l'introduction d'un modèle intermédiaire est nécessaire [64][66]. L'identification en temps réel du modèle de connaissance à partir des entrées et des sorties aboutit au modèle intermédiaire dit modèle de représentation.

L'application de la GPC aux bras manipulateurs est facilitée par l'utilisation d'un modèle de représentation Diagonal-Diagonal [66].

Ce travail est réparti sur cinq chapitres consacrés à l'étude des performances de la commande prédictive généralisée et de son application aux bras manipulateurs. On trouve dans chaque chapitre une introduction, une conclusion et des résultats de simulation validant le développement théorique.

Le chapitre un, traitera le développement de l'algorithme de base de la GPC. L'étape de modélisation est consacrée à l'étude des systèmes discrets en particulier CARIMA (Controlled Auto Regressive integrated Moving Average). La commande est

élaborée à partir de la minimisation d'un critère quadratique multipas basé sur l'écart entre la sortie prédite et la référence. L'efficacité de l'algorithme est testée sur quatre types de systèmes.

Le chapitre Deux, réalise une extension de l'algorithme de base de la GPC. Elle se traduit par l'introduction de nouvelles pondération dans le critère de commande, permettant ainsi d'avoir accès aux choix des pôles en boucle fermée. Des résultats de simulation montreront l'efficacité de l'algorithme sur les problèmes du retard et de la surparamétrisation.

Le chapitre trois, traite la méthode de commande prédictive généralisée adaptative. L'identification paramétrique est réalisée à partir de l'algorithme des moindres carrés récursifs étendu à trace constante. Les paramètres estimés sont utilisés pour la synthèse de la commande comme s'ils étaient les vrais paramètres du processus. Une partie de ce chapitre est consacrée à l'étude de l'approche directe de la GPCA (A:Adaptative). Cette approche permet une identification directe des paramètres du régulateur évitant ainsi la résolution d'une équation algébrique (équation Diophantine) contrairement à l'approche indirecte.

Le chapitre quatre, est consacré à la modélisation de deux robots manipulateurs PUMA et SCARA. La méthode d'Euler-Lagrange nous permet d'établir le modèle dynamique des deux robots. Dans le but de valider ces modèles, la méthode de Newton-Euler est étudiée. Les réponses en boucle ouverte des deux modèles sont obtenues pour deux trajectoires, à savoir, une trajectoire polynomiale et une trajectoire circulaire.

Le chapitre cinq, traite l'application de la commande prédictive généralisée, étudiée dans les trois chapitres précédant, aux deux robots manipulateurs PUMA et SCARA du chapitre quatre. Vu que les modèles des robots sont à deux degrés de libertés, une version multivariable de la GPC s'impose. Elle est basée sur un modèle de représentation découplé. Ce type de modèle facilite l'extension de l'algorithme de commande GPC monovariable, au cas multivariable.

Enfin une conclusion générale termine ce mémoire.

Chapitre 1

ANALYSE DES PERFORMANCES DE LA COMMANDE PREDICTIVE GENERALISEE

*"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 1

ANALYSE DES PERFORMANCES DE LA COMMANDE PREDICTIVE GENERALISEE (GPC)

Introduction.

La sensibilité des algorithmes de commande classique (PID) aux variations de retards du procédé à contrôler nécessite l'utilisation d'une nouvelle loi de commande. Cette loi prend en charge les caractéristiques structurelle du processus. Le changement de ces caractéristiques est dû à la constante de temps des éléments constituant le processus et aux blocs d'acquisition et conversion de signaux [1]. C'est pour résoudre ce type de problème qu'a été introduite la notion de Commande Prédictive. L'idée principale de cette stratégie de commande est le calcul d'un retour issu d'une minimisation d'un critère incluant la sortie à des instants futurs, appelée "sortie prédite". Cette notion de sortie prédite permet de prendre en compte l'effet du retard par un choix judicieux d'un indice de performance adéquat [2].

1.1 Algorithmes de commande prédictive.

Astrom [3] propose un algorithme de commande à variance minimale (VM) basé sur une prédiction à d -pas où d est le retard du système (un nombre multiple de la période d'échantillonnage du processus). La commande est alors synthétisée par minimisation d'un critère quadratique. Ce critère minimise la variance de l'erreur de prédiction entre la sortie du système et la référence désirée.

L'inconvénient majeur de la commande à variance minimale est son domaine d'application relativement restreint [4]. Cette stratégie nécessite une connaissance exacte du retard du processus à commander et un modèle de prédiction à phase minimale.

La détermination du retard, en pratique, est une tâche très difficile dans le cas des systèmes variants dans le temps LTVS (Linear Time Varying Systems) et nonlinéaires. Le comportement à phase non-minimale (PNM) est généralement dû, d'une part au choix de la période d'échantillonnage, d'autre part au comportement du système continu. Clarke [5][6] montre qu'un échantillonnage rapide, d'un système continu à phase minimale, aboutit à un système discret à zéro instable dans le cas où la différence des degrés entre le numérateur et le dénominateur est égale à un.

Une solution à ce problème a été apportée par la minimisation d'un même critère à variance minimale augmenté d'un terme de pondération qui pénalise la commande [7][8][9]. C'est la commande à variance minimale généralisée (GMV) dont le critère J est défini par: [7][8]

$$J = E \{ P [W(t+d) - Y(t+d)]^2 + Q [u(t) - u(t-1)]^2 / t \}$$

avec:

$E \{.\}$: espérance mathématique.

$u(t)$, $y(t)$ et $w(t)$: sont respectivement l'entrée, la sortie et la référence.

P et Q : sont des polynômes de pondération.

$\{./t\}$: informations disponibles.

L'introduction de la pondération dans la GMV, aboutit à un modèle en boucle fermée, dont la dynamique dépend essentiellement du polynôme $PB+QA$ où A et B sont des polynômes caractérisant le système. Le choix du polynôme Q est arbitraire, ce qui nécessite d'utiliser la méthode de lieu des racines [7][8] pour fixe les pôles en boucle fermée, dans le cas où ce polynôme se réduit à une constante.

La commande à placement de pôles (PP) ne nécessite ni une connaissance précise du retard, ni un modèle à phase non-minimale. En effet, il suffit d'augmenter l'ordre du modèle de représentation pour prendre en compte l'incertitude sur la valeur du retard. Le problème inhérent à cette méthode est le manque de robustesse numérique dans la résolution de l'équation Diophantine qui nécessite l'inversion d'une matrice. La singularité de cette dernière est dû, d'une part, à la présence de facteur commun des polynômes du modèle, d'autre part, aux choix de ces degrés

Pour résoudre le problème dû aux variations du retard et à l'instabilité des zéros du processus, la notion de commande prédictive étendu LRP (Long-Range Prediction) est apparue comme solution possible. L'idée de base de cette stratégie de commande, est la minimisation de l'erreur de prédiction sur un horizon de temps. Ainsi, on aboutit à un algorithme de commande insensible au variation du retard et n'exigeant pas la stabilité des zéros du système.

Le fait de prédire "plus loin" le comportement du système sur un horizon de temps, supérieur ou égal au retard, permet d'obtenir des commandes plus douces et donc d'améliorer la régularité de fonctionnement du procédé considéré. Le choix de l'horizon de prédiction de l'erreur de sortie, aboutit à plusieurs algorithmes de commande.

Le concept de prédiction étendu (LRP), en tant qu'outil de synthèse efficace de commande, est dû à Richalet [10]. Celui-ci a proposé une méthode IDCOM (IDentification-COMmande) basée sur une modélisation du procédé par réponse impulsionnelle (Weighting- sequence models), couplée à la synthèse d'une trajectoire de référence, représentant le comportement désiré en boucle fermée et ne considérant aucune pondération sur les actions de commande. Mais cette méthode ne permet pas de stabiliser des systèmes instables en boucle ouverte, ou à phase non-minimale.

Au cours des années 80 Cutler et Ramaker [11] ont proposés l' algorithme DMC (Dynamic Matrix Control), dans lequel la prédiction se fait à l'aide d'un modèle de réponse indicielle, sur un horizon égal à l'ordre du procédé. Le critère quadratique comporte alors les erreurs de prédictions.

Clarke et Zhang [12] ont fait le développement et la comparaison de quatre types d'algorithmes de commande à savoir, IDCOM, DMC, PCA (Predictive Control Algorithme) et la commande prédictive généralisée (GPC:Generalized Predictive Control).

Carlos [13] reprend le travail de Clarke en présentant un survey sur les modèles et les algorithmes de commande ainsi que leur application dans le domaine industriel. Il compare la DMC, MAC (Model Algorithm Control), IMC (Internal Model Control) et LQ

(Linear Quadratic).

La GPC est un des derniers membres de la famille des commandes à horizon étendu LRHPC (Long-Range Horizon Predictive Control), et se veut une généralisation des algorithmes cités ci-dessus. Elle se base sur cinq concepts [2]:

- Le modèle de prédiction utilisé CARIMA (Controlled Autoregressive Integral Moving Average) est une extension du modèle CARMA, dans la mesure où il incorpore un effet intégrale dans le but d'éliminer l'écart permanent et l'effet des perturbations constantes.
- L'utilisation de la prédiction à horizon fini supérieur au retard et à la limite égale à l'ordre du système.
- La résolution récursive de l'équation Diophantine.
- L'introduction de la pondération sur les incréments de commande dans le critère.
- Le choix de l'horizon de commande à partir duquel tous les incréments sont pris égales à zéro.

La combinaison de ces cinq concepts, issus de plusieurs algorithmes, a donné une méthode de commande puissante et performante, très réussie en pratique [1][4][14][15]

La plus part des algorithmes de commande à prédiction étendue sont des cas particuliers de la GPC, et peuvent être retrouvés par certain choix des paramètres de synthèse, à savoir, horizons de prédiction, horizon de commande et la pondération: Ce qui fait la puissance de la GPC [16].

La GMV développée par Clarke [7][8] pour des systèmes à retard connu peut être vu comme un cas particulier de la GPC [2] dont les deux horizons de prédiction N_1 et N_2 sont égaux au retard du système, et seulement un seul signal de commande est pondéré. (N_1, N_2 horizon minimal et maximal de prédiction).

Par un choix judicieux des horizons de prédiction et de commande, avec une pondération nulle, la GPC se comporte comme l'approche de la commande à horizon étendu de Ydstie [17] et l'algorithme EPSAC de Dekeyser [18].

Dans ce chapitre, on s'intéresse à présenter le modèle de représentation utilisé pour l'élaboration de la commande. La commande prédictive généralisée, sera ensuite développée, dont on donne la version de base. L'interprétation du comportement en boucle fermée (BF), illustrera l'influence des horizons de prédiction et de commande, sur les pôles en boucle fermée. Les résultats de simulation permettent, d'analyser les performances de la GPC. Une conclusion clôture ce chapitre.

1.2 Modélisation et commande.

Un modèle de procédé est un ensemble de relations mathématiques permettant de prédire certains aspects de son fonctionnement. Ce modèle doit décrire, avec une précision suffisante, les interactions du procédé avec son milieu extérieur. Le type

de modèle qui nous intéresse est le modèle qui décrit l'évolution d'un procédé dans le temps, par rapport aux contraintes du milieu extérieur: commande et perturbations [19].

1.2.1 Modèle linéaire discret.

Soit un signal analogique $y(t)$ que nous considérons représenté par ses échantillons aux instants $t, t+1, \dots, t+p$ et que nous noterons: $\{y(t), y(t+1), \dots, y(t+p)\}$ avec p entier naturel.

Nous supposons que ce signal est généré à partir d'une excitation $u(t)$, représentée elle aussi par ses échantillons $\{u(t), u(t+1), \dots, u(t+q)\}$ avec q entier naturel.

Un modèle linéaire discret du signal sera une relation linéaire entre les échantillons $\{y(t)\}$ et $\{u(t)\}$, que nous noterons:

$a_0y(t) + \dots + a_p y(t-p) = b_0u(t) + \dots + b_q u(t-q)$. Une telle représentation constitue un modèle ARMA dit d'ordre (p, q) .

- Le processus décrit par: $y(t) + a_1y(t-1) + \dots + a_n y(t-na) = e(t)$ lorsque $\{e(t), t = \dots, -1, 0, 1, \dots\}$ est une séquence de bruit noncorrélée, est appelé processus Auto-Regressive (AR) qui est illustré sur la figure (GPCB.1).

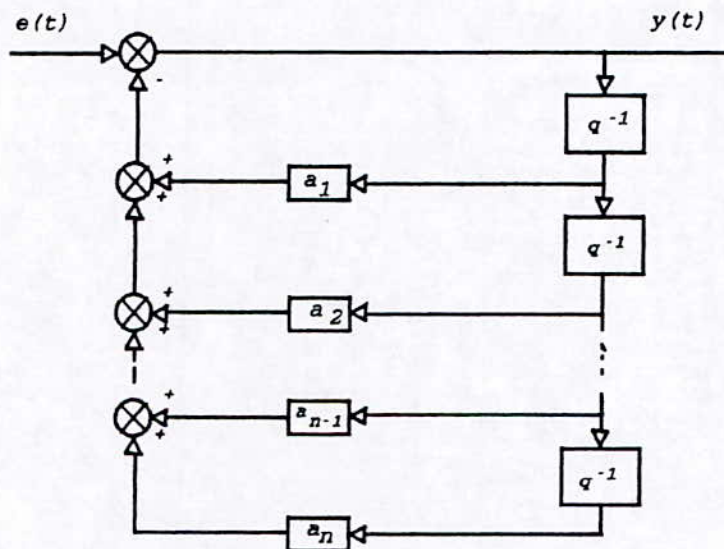


Figure (GPCB.1) Structure d'un modèle AR.

- le processus décrit par: $y(t) = c_1e(t-1) + \dots + c_{nc}e(t-nc)$ est appelé processus à moyenne mobile MA (Moving-Average) dont la structure est présentée sur la figure (GPCB.2).

La combinaison des trois modèles, s'exprimant comme suit:

$y(t) + \dots + a_n y(t-na) = b_0u(t-1) + \dots + b_{nb}u(t-nb) + e(t) + \dots + c_{nc}e(t-nc)$ est appelé processus CARMA.

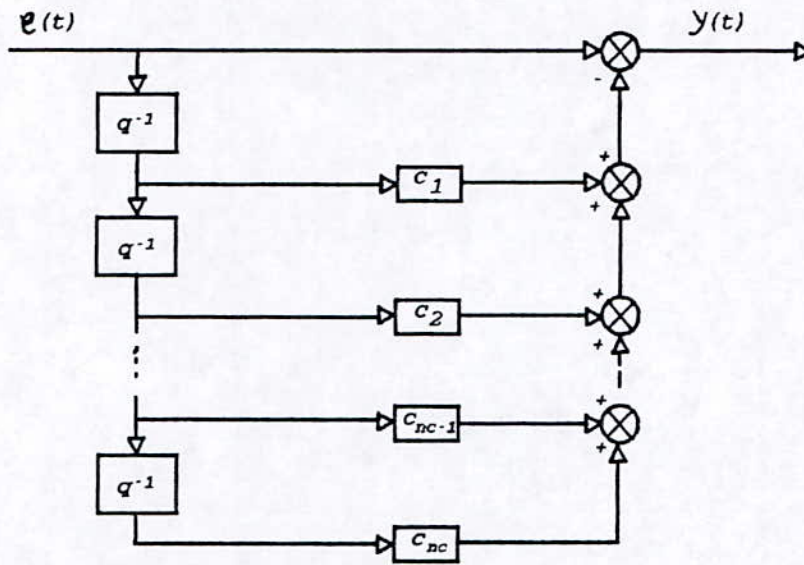


Figure (GPCB.2) Structure d'un modèle MA.

1.2.2 Type de modèle de prédiction utilisé.

L'utilisation de modèles de type fonction de transfert, la plus part des temps discrètes, donne une bonne représentation du processus à commander. De plus qu'il ne nécessite qu'un nombre réduit de paramètres, il permet de situer exactement l'instabilité du système. Soit le modèle CARMA suivant:

$$A(q^{-1})y(t) = B(q^{-1})u(t-d) + C(q^{-1})\xi(t) \quad (1.1)$$

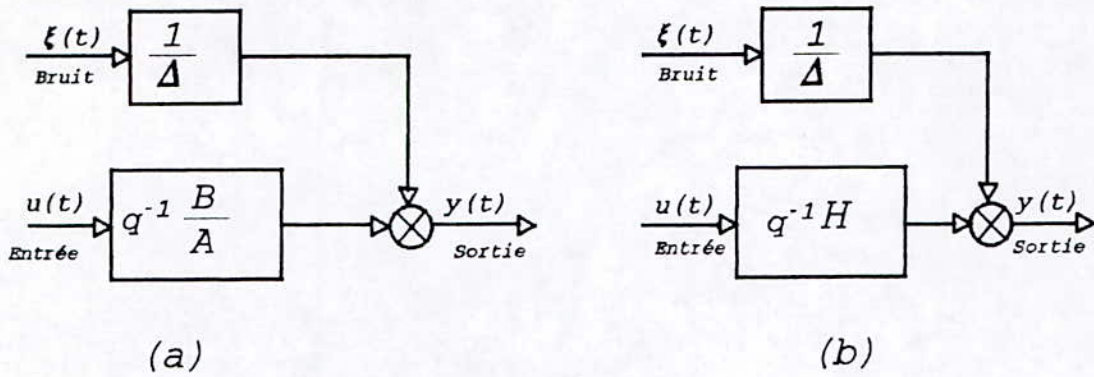
où: q^{-1} : est l'opérateur retard défini par $q^{-1}y(t) = y(t-1)$.
 A, B, C : polynômes en q^{-1} de degrés appropriés.
 $\xi(t)$: séquence de variable aléatoire non corrélée.

Remarque.

Le modèle à réponse impulsionnel [16][22] facilite le développement théorique des algorithmes de commande, mais il possède un domaine d'application très restreint dans la mesure où il ne peut pas représenter un système instable en boucle ouverte. L'intervalle d'échantillonnage doit être bien choisi et suffisamment grand afin de représenter réellement le système à commander. Il est évident que de bonne performance sont obtenues en utilisant les algorithmes se basant sur ce type de modèle mais avec le modèle CARMA ou CARIMA on pourra limiter le nombre de paramètres représentatifs au système et pouvoir commander des processus instables en boucle ouverte. La figure (GPCB.3) illustre la structure d'un modèle impulsionnel (H) et celui du modèle CARIMA.

1.3 Commande prédictive généralisée.

La nature du degré de complexité, met en défaut la plupart des algorithmes de commande développés jusqu'à présent. Pour remplir ce rôle, une méthode ou algorithme de commande doit être appliqué à [2]:



$$\Delta = 1 - q^{-1}$$

figure (GPCB.3) Structure des modèles:
 a) CARIMA. b) à séquence pondérée.

- Un système à phase non-minimale: la plus part des fonctions de transfert temps continus tendent, lors de la discrétisation à exhiber des zéros à l'extérieur du cercle unité.
- Un système instable en boucle ouverte.
- Un système à retard variable ou inconnu: certaines méthodes (VM) sont très sensibles aux hypothèses faites sur le retard et des approches qui essaient de l'estimer.
- Un système à ordre inconnu/variable la méthode à placement de pôles et LQC (Linear Quadratic Gaussian) ont de mauvaises performances si l'ordre du système est surestimé [7].

La méthode que nous allons utiliser GPC surmonte ces problèmes en un seul algorithme. Elle est capable de commander des processus dont l'ordre, le retard et les paramètres varient.

L'objectif de l'étude présentée dans ce travail, consiste en la mise en oeuvre d'une méthode moderne issues de récents développements de l'automatique pour la commande des processus réels [4] [14] [15] [25] [27] [28].

La commande à horizon étendu est calculée pour satisfaire l'identité: $E\{w(t+k) - y(t+k)\}$ où $k > d$. Le calcul de la commande peut se faire par minimisation du critère et seule la commande $u(t)$ est appliquée au procédé, c'est le principe de la commande à horizon glissant (Receding horizon). La robustesse de cet algorithme est garantie pour les systèmes à phase non-minimale si l'horizon de prédiction (j) est suffisamment grand, et la connaissance du retard n'est évidemment pas nécessaire. Par contre le fait qu'une seule prédiction soit utilisée ne permet pas de stabiliser des systèmes naturellement instables. On peut donc étendre cet algorithme de la manière suivante: La sortie n'est plus prédite à un instant futur, mais à un ensemble d'instant depuis $t+d$, jusqu'à $t+j$, le critère multipas a la forme suivant [23]:

$$J = \sum_{i=d}^N [y(t+i) - w(t+i)]^2 \tag{1.2}$$

La première étape de la synthèse d'une loi de commande est la modélisation du

procédé à contrôler. Le modèle doit être capable de représenter les différentes dynamiques présentes, tout en ayant une structure compatible avec le calcul de la loi de commande. La représentation choisie est basée sur un modèle paramétrique, linéaire et perturbé [3] [21].

Les procédés réels étant par nature nonlinéaire, ce modèle décrit le comportement externe du processus dans une zone plus ou moins large autour de son point de fonctionnement.

1.4 Synthèse de la commande prédictive généralisée.

Nous allons présenter dans cette partie une approche de commande qui résout la plupart des problèmes rencontrés lors de l'utilisation d'autres types de commande

1.4.1 Modélisation.

Le modèle adopté par la GPC est un modèle paramétrique linéaire qui s'écrit comme suit:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \omega(t) \quad (1.3)$$

où A et B sont des polynômes fonctions de l'opérateur retard q^{-1} , tel que:

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_{na} q^{-na} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \dots + b_{nb} q^{-nb} \\ \text{où } na &= d^{\circ}(A) \text{ et } nb = d^{\circ}(B) \end{aligned}$$

Avec $u(t)$ la commande ou l'entrée du procédé, $y(t)$ est la sortie ou la variable mesurée. Le retard un est dû à la discrétisation du modèle continu. Ce modèle ne fait pas apparaître le retard d car si la système possède un retard différent de zéros, les premiers éléments du polynôme B sont nuls. $\omega(t)$ est le terme de perturbation qui se présente sous forme d'un modèle MA (moyenne glissante) qui s'exprime comme suit.

$$\omega(t) = C(q^{-1})\xi(t) \quad (1.4)$$

$$\text{avec } C(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc} \text{ où } nc = d^{\circ}(C)$$

Tous processus peut être influencé par son environnement, il est caractérisé par un processus stochastique. Dans cette équation, $\xi(t)$ est une séquence de variable aléatoires noncorrélées à moyenne nulle et de variance finie (bruit blanc). En combinant avec l'équation (1.3) on obtient le modèle CARMA [2].

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\xi(t) \quad (1.5)$$

Dans le but d'éliminer les perturbations constantes et de donner un modèle général des perturbations, aléatoires et déterministes, on introduit un effet intégral, en écrivant le terme de perturbation $\omega(t)$ comme suit: [16] [24]

$$\omega(t) = C(q^{-1}) \frac{\xi(t)}{\Delta} \quad (1.6)$$

$\omega(t)$ représente alors la sortie d'un système excité par une séquence de variables aléatoires indépendantes $\{\xi(t)\}$. $\Delta = (1 - q^{-1})$ est l'opérateur de différence, en introduisant $\omega(t)$ dans l'équation (1.1) nous obtenons le modèle CARIMA (I: Integrated) suivant.

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1}) \frac{\xi(t)}{\Delta} \quad (1.7)$$

Pour faciliter le développement de l'algorithme, posons $C(q^{-1}) = 1$, le cas où $C(q^{-1}) \neq 1$ sera traité dans le chapitre 2.

1.4.2 Prédiction de la sortie.

Pour commander un système il faut disposer de la prédiction sur un horizon donné. Cet horizon a été pris égal au retard dans le développement de la GMV [3][4][5]. La commande prédictive généralisée consiste en la prédiction d'un ensemble de sorties sur un horizon j supérieur au retard. Considérons l'équation (1.7) écrite à l'instant $t+j$:

$$y(t+j) = \frac{B(q^{-1})}{A(q^{-1})} u(t+j-1) + \frac{\xi(t+j)}{A(q^{-1})\Delta(q^{-1})} \quad (1.8)$$

avec: $q^{-1} \frac{B(q^{-1})}{A(q^{-1})}$: fonction de transfert relative à la sortie et à la commande.

$\frac{1}{A(q^{-1})\Delta(q^{-1})}$: fonction de transfert relative à la sortie et à la perturbation aléatoire $\xi(t)$.

Le terme $\{\xi(t+j)/A(q^{-1})\Delta(q^{-1})\}$ dépend de $\{\xi(t+j), \dots, \xi(t+1), \xi(t), \xi(t-1), \dots\}$ et $\{\xi(t), \xi(t-1), \dots\}$ peuvent être calculés à partir de l'information disponible à l'instant t , c'est à dire à partir des mesures $\{y(t), y(t-2), \dots\}$ et des commandes $\{u(t-1), u(t-2), \dots\}$. Les termes $\{\xi(t+j), \dots, \xi(t+1)\}$ constituent la partie non prédictible.

Pour expliciter analytiquement cette décomposition, introduisons l'identité polynomiale suivante:

$$1 = E_j(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-j}F_j(q^{-1}) \quad (1.9)$$

Appelée équation Diophantine, obtenue à l'aide de l'algorithme d'Euclide qui permet d'effectuer la division de 1 par le polynôme $A(q^{-1})\Delta(q^{-1})$ jusqu'à l'ordre j . Les polynômes E_j et F_j sont déterminés à partir d'un algorithme récursif développé par Clarke [2]. Nous pouvons ainsi remarquer qu'ils dépendent uniquement de $A(q^{-1})$ et de l'intervalle de prédiction j tel que, $j-1$ et na sont les degrés respectifs.

En multipliant l'équation (1.8) par $E_j A \Delta$, on obtient la décomposition de l'information connue de celle qui ne l'est pas, et en substituant $E_j A \Delta$ de l'équation (1.9), on obtient la sortie à l'instant $t+j$.

$$y(t+j) = E_j(q^{-1}) B(q^{-1}) \Delta(q^{-1}) u(t+j-1) + F_j(q^{-1}) y(t) + E_j(q^{-1}) \xi(t+j) \quad (1.10)$$

Le polynôme E_j est de degré $j-1$, les composantes du bruit sont toutes dans le futur.

La minimisation de l'erreur de prédiction au sens des moindres carrés, aboutit à la détermination du prédicteur optimal compte tenu des informations disponibles à l'instant t . ce prédicteur s'écrit

$$\hat{y}(t+j/t) = G_j(q^{-1}) \Delta u(t+j-1) + F_j(q^{-1}) y(t) \quad (1.11)$$

avec $G_j(q^{-1}) = E_j(q^{-1}) B(q^{-1})$

La structure de ce prédicteur optimal est présentée sur la figure (GPCB.4).

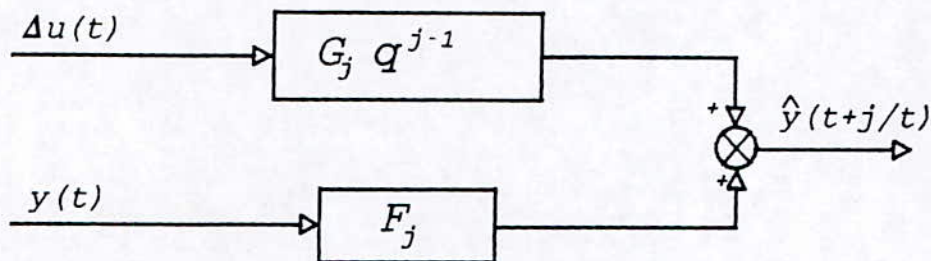


Figure (GPCB.4) Structure du prédicteur optimale.

dans Le développement de certains algorithmes de commande prédictive une seule prédiction, $\hat{y}(t+k/t)$ où k est le retard du système, est considérée. Dans la commande prédictive généralisée, nous considérons un ensemble de prédiction sur j -pas, où j varie dans un intervalle appelé horizon de prédiction. Pour $j < k$ le prédicteur $\hat{y}(t+j/t)$ dépend des données disponibles jusqu'à l'instant t , mais pour $j > k$ des hypothèses doivent être faites sur les actions de commande future.

1.4.3 Résolution itérative de l'équation Diophantine.

L'implémentation de la GPC nécessite la résolution de l'équation Diophantine. Le développement itérative de cette équation ne présente aucune difficulté du point de vu mathématique [2][25].

Considérons l'équation suivante:

$$1 = E_j(q^{-1}) A(q^{-1}) \Delta(q^{-1}) + q^{-j} F_j(q^{-1}) \quad (1.12)$$

Supposons que E_j et F_j sont connus et cherchons à calculer E_{j+1} et F_{j+1} . Ces deux polynômes doivent vérifier l'équation (1.12) pour $j+1$.

$$1 = E_{j+1}(q^{-1}) A(q^{-1}) \Delta(q^{-1}) + q^{-(j+1)} F_{j+1}(q^{-1}) \quad (1.13)$$

Les polynômes E_j et E_{j+1} étant respectivement de degrés $j-1$ et j , on obtient alors

par soustraction des polynômes E_{j+1} et E_j :

$$F_{j+1}(q^{-1}) = q \{F_j(q^{-1}) - A(q^{-1}) \Delta(q^{-1}) r_j\} \quad (1.14)$$

et posons $F_j(q^{-1}) = f_{j,0} + f_{j,1}q^{-1} + \dots + f_{j,na}q^{-na}$

$$A(q^{-1}) \Delta(q^{-1}) = a_0^* + a_1^*q^{-1} + \dots + a_{na+1}^*q^{-(na+1)}$$

La récursion qui permet de calculer les coefficients des polynômes E_{j+1} et F_{j+1} connaissant les polynômes E_j et F_j est:

$$\begin{cases} r_j = f_{j,0} \\ f_{j+1,i} = f_{j,i+1} - a_{i+1}^* r_j \quad i=0, \dots, na \end{cases} \quad (1.15)$$

Avec la condition initiale pour $j=1$;

$$1 = E_1 A \Delta + q^{-1} F_1 \quad (1.16)$$

avec:
$$\begin{cases} E_1 = 1 \\ F_1 = q(1 - A\Delta) \end{cases} \quad \text{et} \quad \begin{cases} E_{j+1} = E_j + q^{-j} r_j \\ G_{j+1} = B E_{j+1} \end{cases}$$

1.4.4 Lois de commande prédictive.

L'objectif de la loi de commande prédictive est de conduire la sortie future du système $y(t+j)$ à la référence, comme c'est illustré dans la figure (GPCB.5)

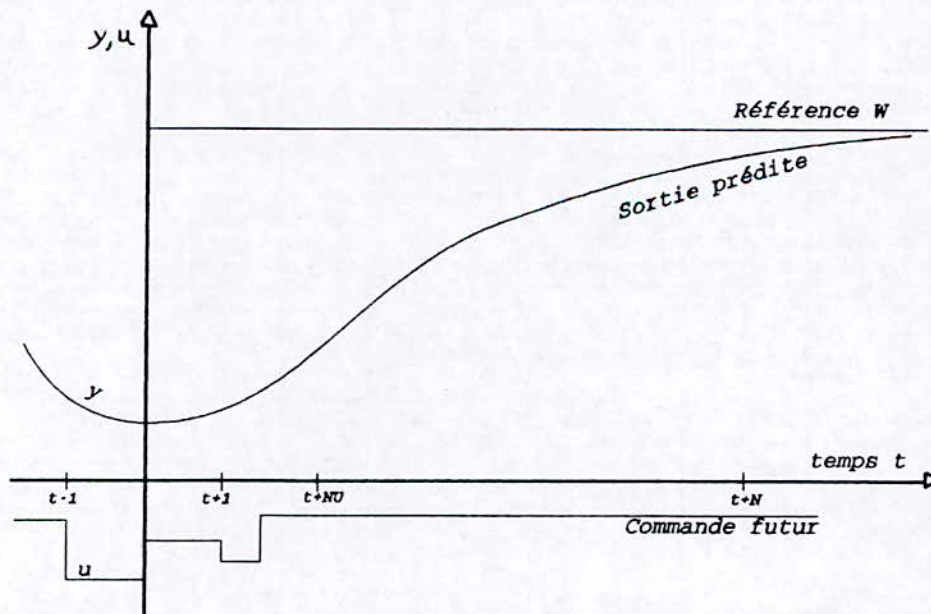


Figure (GPCB.5) Référence, commande et sortie dans la GPC.

La stratégie d'horizon fuyant nous permet d'atteindre cet objectif. A chaque instant t

Les étapes de l'algorithme sont:

- étape 1. La séquence de référence futur $w(t+j)$ est calculée.
- étape 2. Le modèle de prédiction de l'équation (1.11) est utilisé pour générer un ensemble de sortie $\hat{y}(t+j/t)$ avec l'erreur de prédiction correspondant au système $e(t+j) = w(t+j) - \hat{y}(t+j)$. notons que pour $j > k$, $\hat{y}(t+j/t)$ dépend des signaux de commande future $u(t+i)$ qui doivent être déterminées.
- étape 3. une fonction quadratique incluant des erreurs futures et des commandes, est minimisée, en tenant compte qu'après un certain "horizon de commande" tous les incréments de commandes futures sont égales à zéros.
- étape 4. Seul le premier élément $u(t)$ de la séquence est appliqué au système.
- étape 5. Tous les vecteurs de données sont décalés de telle sorte que les calculs peuvent être répétés à chaque instant.

1.4.5 Critère quadratique.

La loi de commande prédictive considérée par la GPC est associée à la minimisation du critère quadratique suivant [2][23]:

$$J(N1, N2, NU, t) = E \left[\sum_{j=N1}^{N2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{NU} \lambda_j [\Delta u(t+j-1)]^2 \right] \quad (1.17)$$

- où: $w(t+j)$: est une séquence de référence future initialement connue.
- $N1$: est l'horizon minimum de prédiction.
- $N2$: est l'horizon maximum de prédiction.
- NU : est l'horizon de commande.
- λ_j : scalaire positif agissant sur les incréments de commande future

1.4.6 Détermination du prédicteur à j -pas.

L'expression de la sortie à l'instant $t+j$, s'écrit:

$$\left\{ \begin{array}{l} y(t+1) = G_1 \Delta u(t) + F_1 y(t) + E_1 \xi(t+1) \\ y(t+2) = G_2 \Delta u(t+1) + F_2 y(t) + E_2 \xi(t+2) \\ \vdots \\ y(t+N2) = G_{N2} \Delta u(t+N2-1) + F_{N2} y(t) + E_{N2} \xi(t+N2) \end{array} \right. \quad (1.18)$$

Qui consiste en trois termes, le premier dépend des actions de commande future (terme à déterminer), un autre dépend des commandes connues et le troisième des signaux de bruit futurs (terme imprédictible).

Le prédicteur j -pas de l'équation (1.11) peut être écrit sous forme matricielle, et cela dans le but de:

- faire une prédiction en terme d'incrément de commande.
- décomposer la sortie prédite en deux termes, un dépendant des commandes futures et l'autre ne comportant que des signaux connus à l'instant t .

Soit:

$$\hat{y}(t+j/t) = \hat{y}_1(t+j) + \hat{y}_2(t+j/t) \quad (1.19)$$

Nous avons $\hat{y}(t+j/t) = G_j \Delta u(t+j-1) + F_j y(t)$ et le polynôme G_j peut être décomposé en deux parties tel que: $G_j = \tilde{G}_j + q^{-j} \bar{G}_j$

$$\text{avec: } \begin{cases} \tilde{G}_j = g_0 + g_1 q^{-1} + \dots + g_{j-1} q^{-(j-1)} \\ \bar{G}_j = g_{j0} + g_{j1} q^{-1} + \dots + g_{j(nb-1)} q^{-(nb-1)} \end{cases}$$

Donc l'équation du prédicteur optimal devient:

$$\hat{y}(t+j/t) = \tilde{G}_j \Delta u(t+j-1) + \bar{G}_j \Delta u(t-1) + F_j y(t) \quad (1.20)$$

Posons $f(t+i) = \bar{G}_i \Delta u(t-1) + F_i y(t)$ où $i=1, \dots, j$

En identifiant l'équation(1.20) à (1.19) nous aurons:

$\hat{y}_1(t+j) = \tilde{G}_j \Delta u(t+j-1)$: fonction des valeurs présentes et futurs des incrément de commande ($\Delta u(t+i)$ pour $i \geq 0$)

$\hat{y}_2(t+j/t) = \bar{G}_j \Delta u(t-1) + F_j y(t)$: fonction des données disponibles à l'instant t .

Considérons maintenant la séquence $\{f(t+1), f(t+2), \dots\}$ qui est fonction des données disponibles à l'instant t , mais seulement, il faut réécrire $\tilde{G}_j = q^j (G_j - \bar{G}_j)$ ce qui nous permet d'écrire le prédicteur à j -pas sous forme matricielle.

$$\begin{bmatrix} \hat{y}(t+1/t) \\ \hat{y}(t+2/t) \\ \vdots \\ \hat{y}(t+j/t) \end{bmatrix} = \begin{bmatrix} g_{10} & \cdot & \cdot & \cdot & 0 \\ g_{21} & g_{20} & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{jj-1} & g_{jj-2} & \cdot & \cdot & g_{j0} \end{bmatrix} \times \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \cdot \\ \cdot \\ \Delta u(t+j-1) \end{bmatrix} \quad (1.20)$$

$$+ \begin{bmatrix} (G_1 - g_{10}) \Delta u(t) + F_1 y(t) \\ q(G_2 - g_{20} - g_{21} q^{-1}) \Delta u(t) + F_2 y(t) \\ \cdot \\ \cdot \\ q^{j-1} (G_j - g_{j0} - g_{j1} q^{-1} - \dots - g_{jj-1} q^{-(j-1)}) \Delta u(t) + F_j y(t) \end{bmatrix}$$

Sous une autre forme:

$$\hat{Y} = G \tilde{U} + f \tag{1.22}$$

Avec tous les vecteurs sont de dimension $(N2 \times 1)$ sous l'hypothèse que $g_{ij} = g_j$ pour $j=0, 1, 2, \dots, i$.

$$\text{Avec: } \begin{cases} \hat{Y} = [\hat{y}(t+1), \dots, \hat{y}(t+N2)] \\ \tilde{U} = [\Delta u(t), \dots, \Delta u(t+N2-1)] \\ f = [f(t+1), \dots, f(t+N2)] \end{cases}$$

En tenant compte de l'hypothèse précédente, l'équation (1.21) devient:

$$\begin{bmatrix} \hat{y}(t+1/t) \\ \hat{y}(t+2/t) \\ \vdots \\ \hat{y}(t+N2/t) \end{bmatrix} = \begin{bmatrix} g_0 & \cdot & \cdot & \cdot & \cdot & 0 \\ g_1 & g_0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{N2-1} & g_{N2-2} & \cdot & \cdot & \cdot & g_0 \end{bmatrix} \times \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N2-1) \end{bmatrix} \tag{1.23}$$

$$+ \begin{bmatrix} (G_1 - g_0) \Delta u(t) + F_1 y(t) \\ q(G_2 - g_0 - g_1 q^{-1}) \Delta u(t) + F_2 y(t) \\ \vdots \\ q^{N2-1} (G_{N2} - g_0 - g_1 q^{-1} - \dots - g_{N2-1} q^{-(N2-1)}) \Delta u(t) + F_{N2} y(t) \end{bmatrix}$$

La matrice G se présente sous la forme triangulaire de dimension $(N2 \times N2)$.

$$G = \begin{bmatrix} g_0 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ g_1 & g_0 & 0 & \cdot & \cdot & \cdot & 0 \\ g_2 & g_1 & g_0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{N2-1} & g_{N2-2} & \cdot & \cdot & \cdot & \cdot & g_0 \end{bmatrix} \tag{1.24}$$

Il est à noter que si le retard du système est supérieur à un ($k > 1$) les premiers $k-1$ lignes de la matrice G seront égales à zéros, ceci est explicité en annexe 1. Le critère (1.17) peut alors s'écrire sous la forme vectorielle suivante:

$$J = E [(Y-W) + \lambda \tilde{U}^T \tilde{U}] \tag{1.25}$$

La minimisation de ce critère par rapport à \tilde{U} s'écrit [2]:

$$\tilde{U} = \Delta u = (G^T G + \lambda I)^{-1} G^T (W - f) \tag{1.26}$$

avec I est la matrice identité.

En se basant sur le principe d'horizon fuyant, la première commande $u(t)$ est appliquée au système [2] [16] [24].

$$u(t) = u(t-1) + \bar{g}^T (W - f) \tag{1.27}$$

où \bar{g}^T est la première ligne de $(G^T G + \lambda I)^{-1} G^T$.

1.4.7 Introduction des horizons $N1$, $N2$ et NU .

La matrice G de l'équation (1.24) est de dimension $(N2 \times N2)$. Basé sur l'hypothèse faite à propos des actions de commande futurs, c'est à dire qu'après l'horizon de commande NU tous les incréments de commande sont pris égaux à zéros, soit:

$$\Delta u(t+j-1) = 0 \quad \text{pour } j > NU \tag{1.28}$$

la matrice G devient de dimension $(N2 \times NU)$, et donc la matrice à inverser sera de dimension $(NU \times NU)$. Si dans le cas où NU est égal à un ($NU=1$) la matrice $(G^T G + \lambda I)^{-1}$ se réduit à un scalaire dont l'inversion ne pose aucun problème. Par l'introduction de l'horizon initial de prédiction $N1$ la matrice G se réduit au dimension $(N2 + N1 - 1) \times (NU)$. Ce qui se traduit par une troncature de la matrice G du haut (ligne) par l'introduction de l'horizon initial de prédiction et à gauche (colonne) par l'horizon de commande NU . La matrice G se présente sous la forme suivante.

$$G = \begin{bmatrix} g_{N1-1} & g_{N1-2} & \dots & g_0 & 0 & \dots & 0 \\ g_{N1} & g_{N1-1} & \cdot & \cdot & g_0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & g_c \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{N2-1} & g_{N2-2} & \cdot & \cdot & \cdot & \cdot & \cdot g_{N2-NU} \end{bmatrix} \tag{1.29}$$

1.5 Structure de réglage RST.

La formulation de la loi de commande GPC sous la forme canonique RST, dont la structure est illustrée sur la figure (GPCB.6), et l'expression des équations du système en boucle fermée, permettent d'étudier ses caractéristiques et ses spécifications de commande [1] [2] [23] [26] [29].

En utilisant les résultats obtenus lors de la synthèse de l'algorithme de base, concernant la commande à appliquer au système, on aboutit à la somme suivante après quelques manipulation mathématiques [1] [2]:

$$\Delta u(t) = \sum_{j=N1}^{N2} \gamma_{ij} [w(t+j) - f(t+j)] \tag{1.30}$$

tel que γ_{ij} sont les éléments de la matrice définie par $\Gamma = (G^T G + \lambda I)^{-1} G^T$

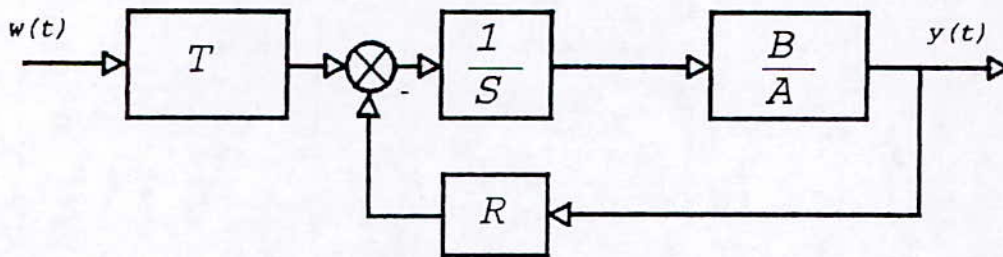


Figure (GPCB.6) Structure du régulateur RST.

avec: $[i, j] = [(1, NU), (1, N2 - N1 + 1)]$.

En remplaçant l'expression de $f(t+j)$ déjà établie dans l'équation (1.30), on aboutit à l'égalité suivante:

$$\Delta u(t) \left[1 + q^{-1} \sum_{i=N1}^{N2} \gamma_{ij} g_i \right] = w(t) \sum_{i=N1}^{N2} \gamma_{1i} q^i - y(t) \sum_{i=N1}^{N2} \gamma_{1i} F_i \quad (1.31)$$

Sachant que $S(q^{-1})u(t) = T(q^{-1})w(t) - R(q^{-1})y(t)$, obtenue à partir du schéma bloc de la figure (1.6), et en l'identifiant membre à membre à l'équation (1.31) on trouve les polynômes constituant le régulateur RST, qui s'écrivent [9]:

$$\begin{cases} S(q^{-1}) = 1 + q^{-1} \sum_{i=N1}^{N2} \gamma_{1i} g_i \\ R(q^{-1}) = \sum_{i=N1}^{N2} \gamma_{1i} F_i \\ T(q^{-1}) = \sum_{i=N1}^{N2} \gamma_{1i} q^i \end{cases} \quad (1.32)$$

La fonction de transfert du système en boucle fermée, exprimée en fonction de R, S et T , s'écrit:

$$y(t) = \frac{BT}{SA\Delta + BR} w(t) + \frac{S}{SA\Delta + BR} \xi(t) \quad (1.33)$$

L'expression analytique des polynômes de la structure canonique RST ne permet pas une analyse facile du comportement en boucle fermée du système à commander. L'expression en BF d'un système pour différentes valeurs des paramètres de synthèse de la GPC est présentés en annexe 1.

1.6 Résultats de simulation.

L'objectif de cette partie est d'étudier l'influence des paramètres de synthèse de la GPC sur plusieurs systèmes, à savoir, coefficient de pondération de la commande λ , horizon initial de prédiction $N1$, horizon maximal de prédiction $N2$ et l'horizon de commande NU . En effet le choix de ces paramètres influent considérablement sur les performances du système. L'algorithme de commande GPC a été appliqué à quatre type de système:

- Système stable en boucle ouverte.
- Système à phase non-minimale.
- Système instable en boucle ouverte.
- Système instable et à phase non-minimale.

Le but est de tester l'efficacité de la GPC (de base) et d'obtenir des résultats suffisamment généraux concernant la choix des paramètres λ , $N1$, $N2$, et NU .

La référence imposée aux systèmes est un signal carré qui se répète sur quatre périodes. Chaque période contient 100 échantillons.

Pour chaque type de système et pour chaque paramètre de commande choisi, on présente les résultats de simulation concernant l'évolution de la sortie et de la commande correspondante.

1.6.1 Etude d'un système stable en boucle ouverte.

Soit la fonction de transfert discrète du deuxième ordre suivante:

$$y(t) = \frac{0.8q^{-1} + 0.4q^{-2}}{1 - 1.7q^{-1} + 0.72q^{-2}} u(t)$$

dont les pôles et les zéros sont stables et ayant pour valeur: $p_1=0.9$, $p_2=0.8$ et $z=0.5$.

La simulation de ce système en boucle ouverte, pour une entrée échelon, présentée sur la figure (1.1) montre que le régime statique est établi à 60. Par le choix de $N1=1$ $N2=1$ $NU=1$ et $\lambda=0$, le système répond très rapidement en suivant exactement la référence imposée avec des pics de commande important ceci est illustré sur la figure (1.2).

Les figures (1.3)(1.4)(1.5) montrent l'évolution du système simulé pour plusieurs valeurs de l'horizon maximal de prédiction respectivement pour $N2=2$ $N2=5$ $N2=10$ en gardant $N1=1$ $NU=1$ $\lambda=0$. Cette augmentation joue sur la dynamique de sortie et le temps de réponse. On constate aussi que les pics et l'effort de commande diminuent. En effet, ces pics peuvent être dangereux pour le système lors d'un changement de consigne.

La diminution de l'effort et des pics de commande peut être aussi obtenue par l'introduction d'une pondération sur la commande. Soit $\lambda=10^{-3}$ cette pondération et pour $N1=1$ $N2=1$ on constate les résultats consignés sur la figure (1.6) qui peuvent être comparés ceux de la figure (1.2) où $\lambda=0$. Ce paramètre ne doit pas être changé si les commandes sont douces et réalisables, autrement la réponse du système sera affectée du point de vu dynamique. On dit alors qu'il n'y a pas lieu de pondérer la commande.

Les figures (1.5)(1.7) montrent l'influence de l'augmentation de NU sur l'effort de commande. En effet, on constate que si $NU=2$ figure (1.7) et $NU=8$ figure (1.8) les performance de sortie sont affectés. Néanmoins une solution à cette effort peut être donnée par l'augmentation de $N2=13$ illustrée sur la figure (1.9) où de $\lambda=1$ figure(1.10).

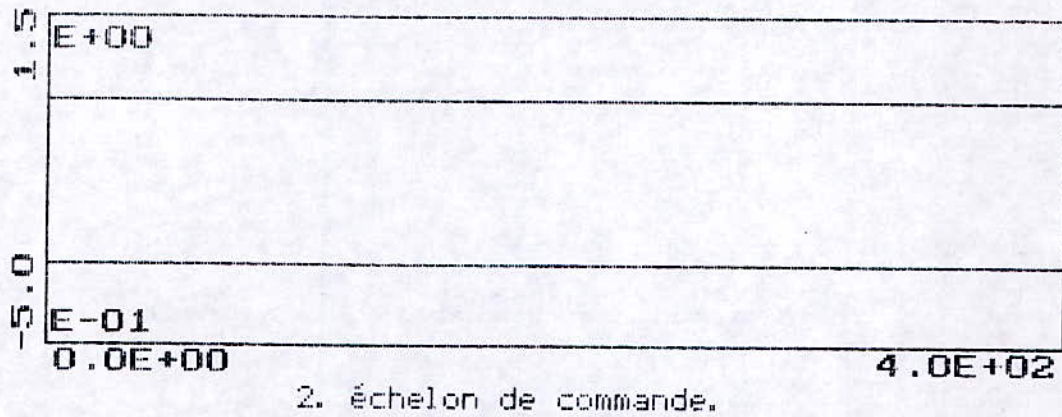
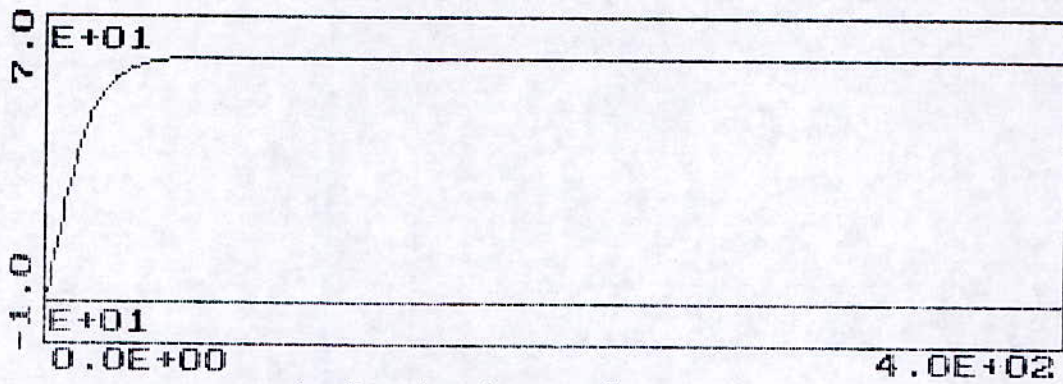
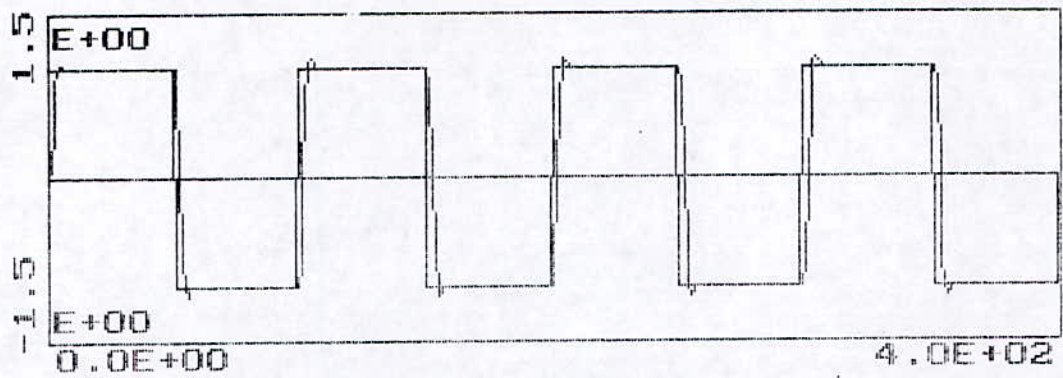
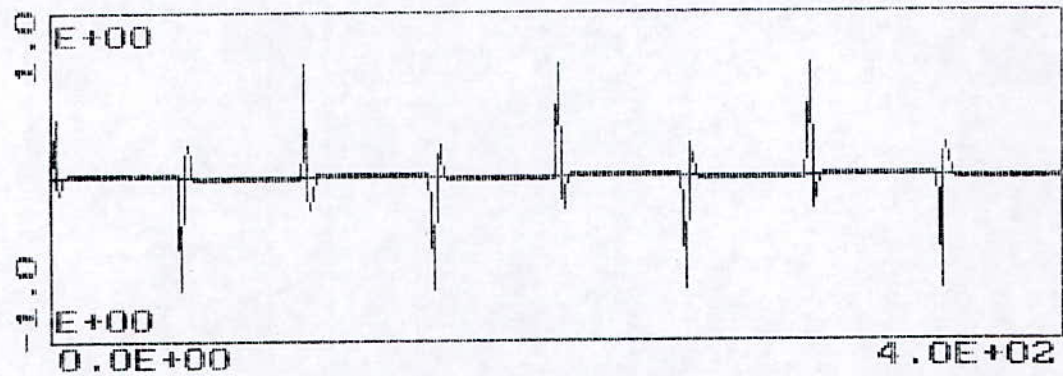


Figure 1.1 Réponse d'un système en boucle ouverte à une entrée échelon.

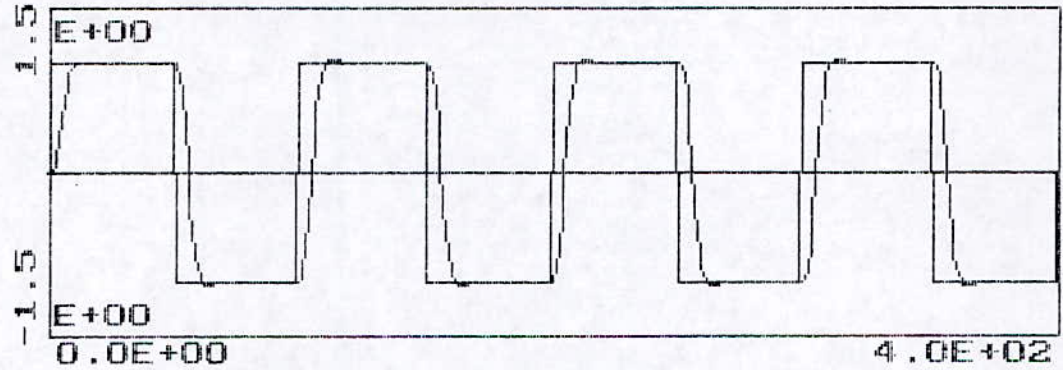


1. évolution de la sortie du système.

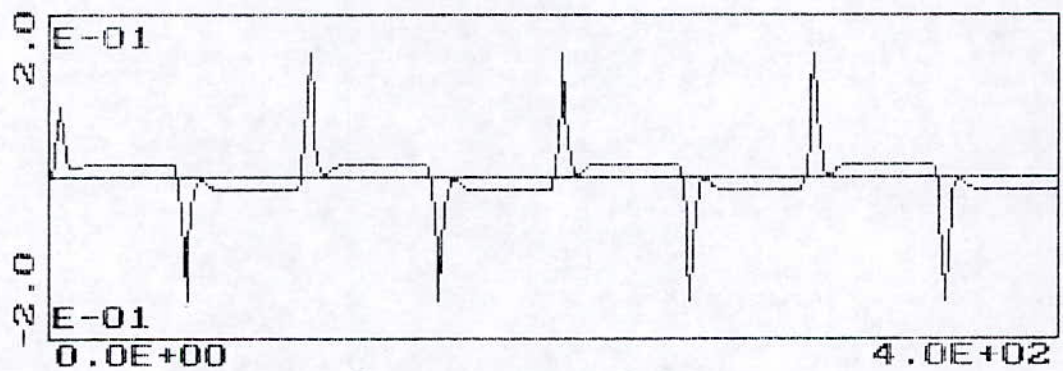


2. évolution de la commande.

Figure 1.3 Présentation sortie/ commande d'un système stable en B0 pour $N1=1$ $N2=2$ $NU=1$ $\lambda=0$.

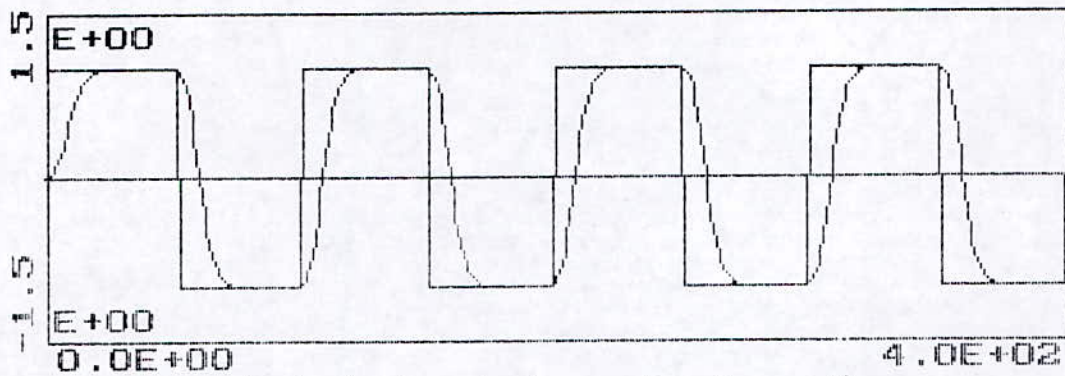


1. évolution de la sortie du système.

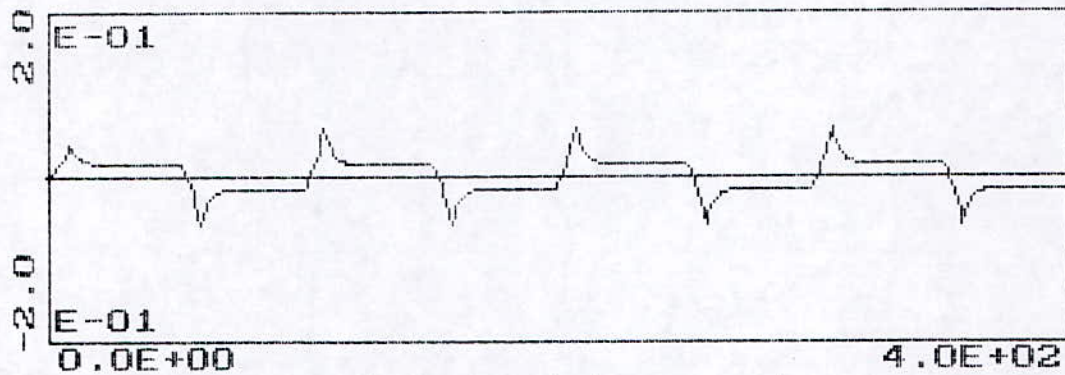


2. évolution de la commande.

Figure 1.4 Présentation sortie/ commande d'un système stable en B0 pour $N1=1$ $N2=5$ $NU=1$ $\lambda=0$.

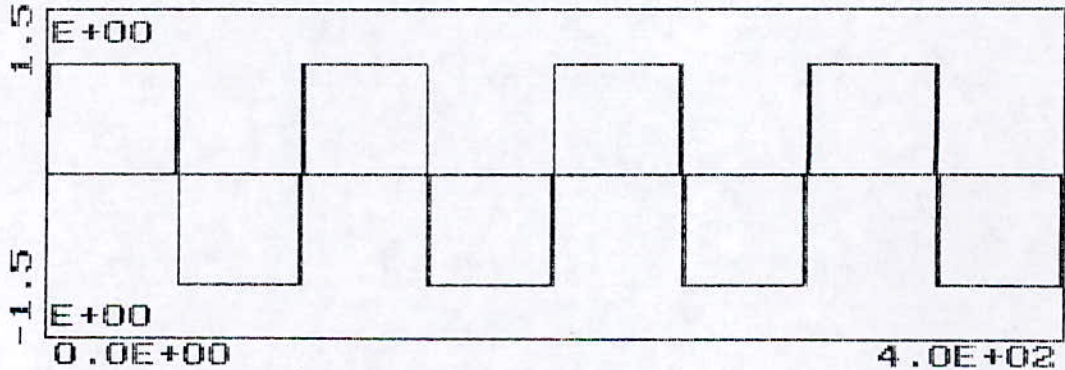


1. évolution de la sortie du système.

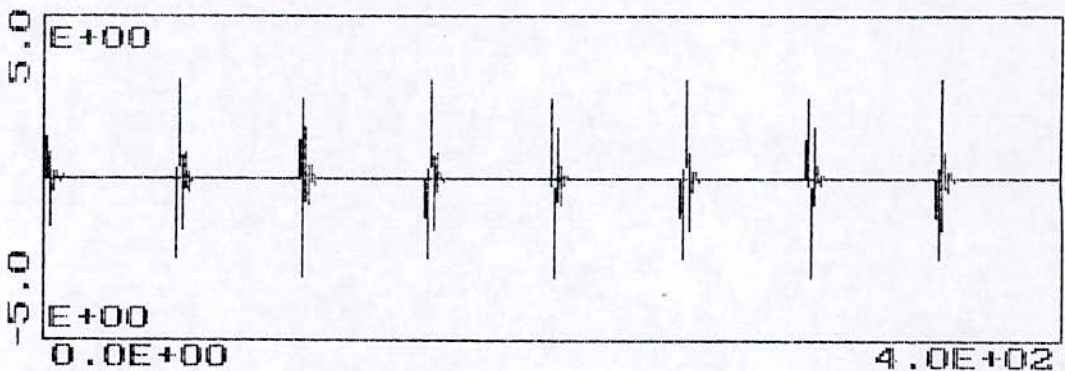


2. évolution de la commande.

Figure 1.5 Présentation sortie/ commande d'un système stable en B0 pour $N1=1$ $N2=10$ $NU=1$ $\lambda=0$.

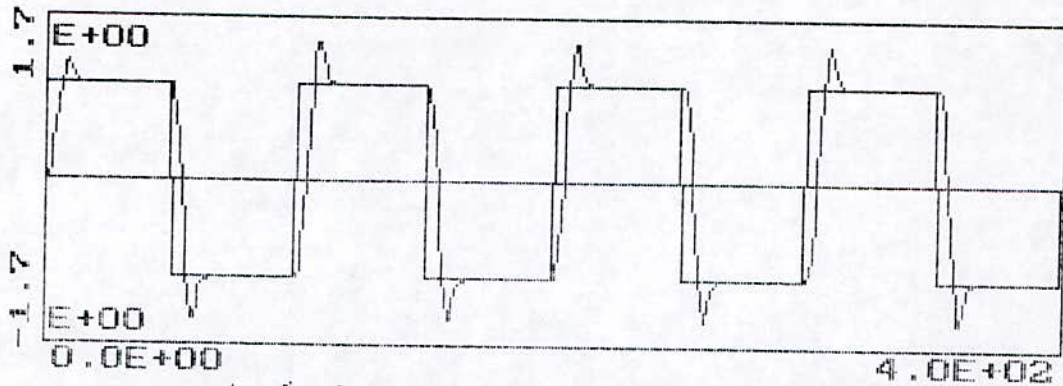


1. évolution de la sortie du système.

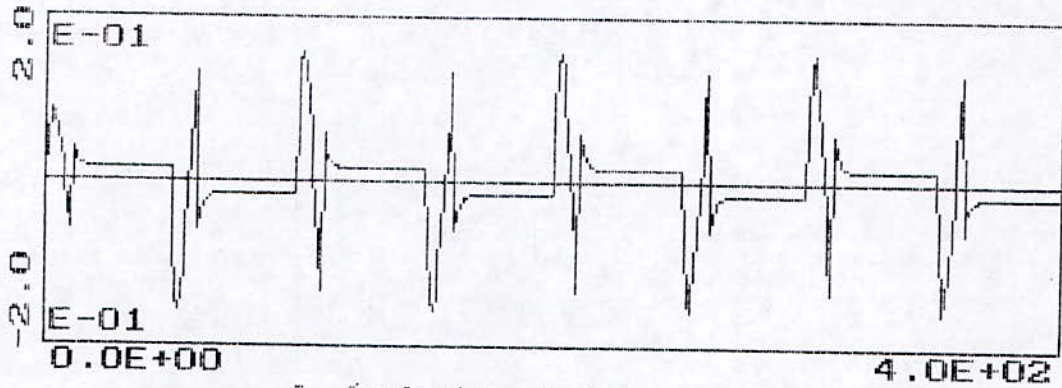


2. évolution de la commande.

Figure 1.6 Présentation sortie/ commande d'un système stable en B0 pour $N1=1$ $N2=1$ $NU=1$ $\lambda=0.001$.

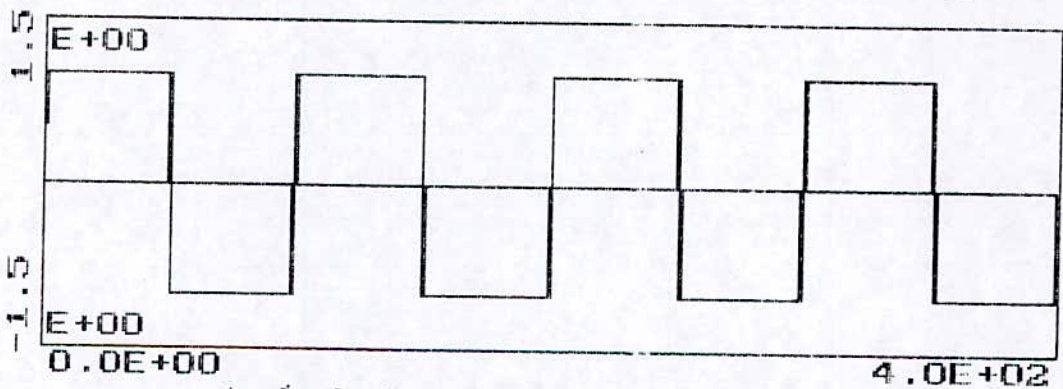


1. évolution de la sortie du système.

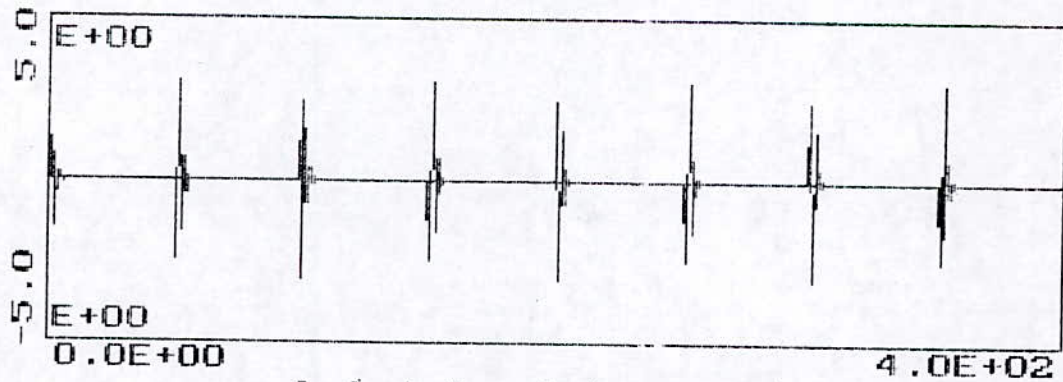


2. évolution de la commande.

Figure 1.7 Présentation sortie/ commande d'un système stable en B0 pour $N1=1$ $N2=10$ $NU=2$ $\lambda=0$.

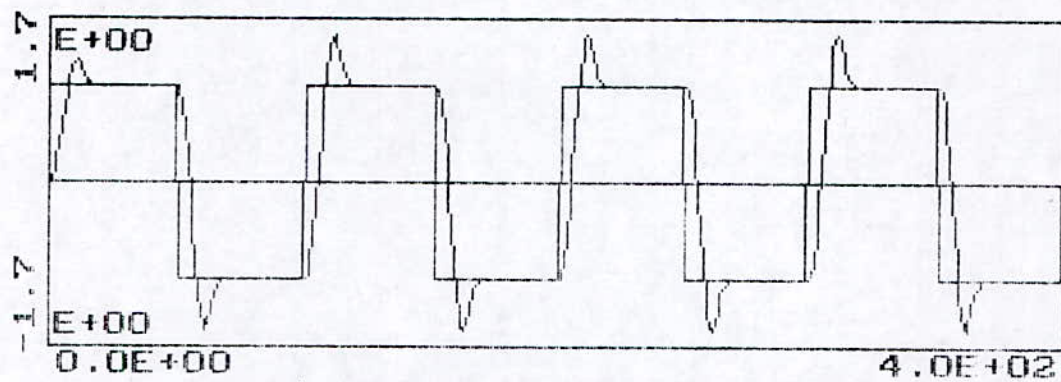


1. évolution de la sortie du système.

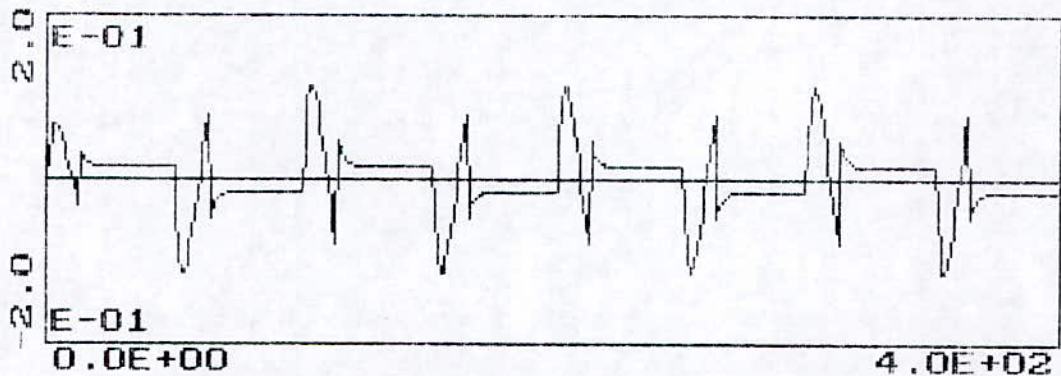


2. évolution de la commande.

Figure 1.8 Présentation sortie/ commande d'un système stable en B0 pour $N1=1$ $N2=10$ $NU=8$ $\lambda=0$.

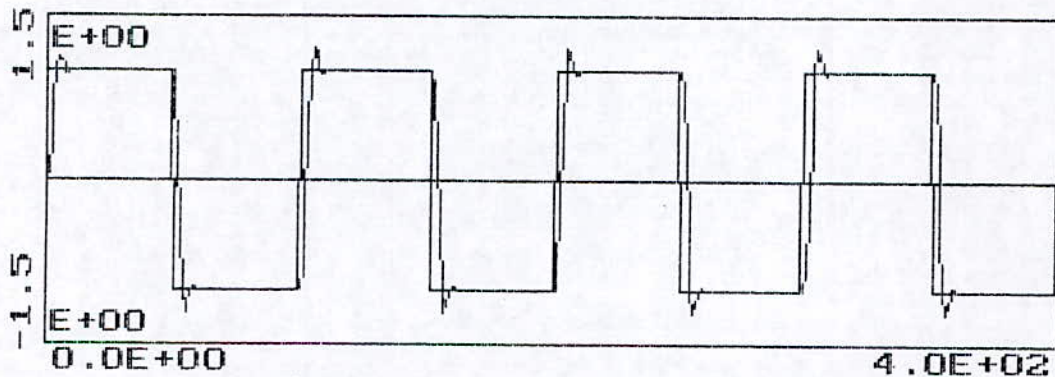


1. évolution de la sortie du système.

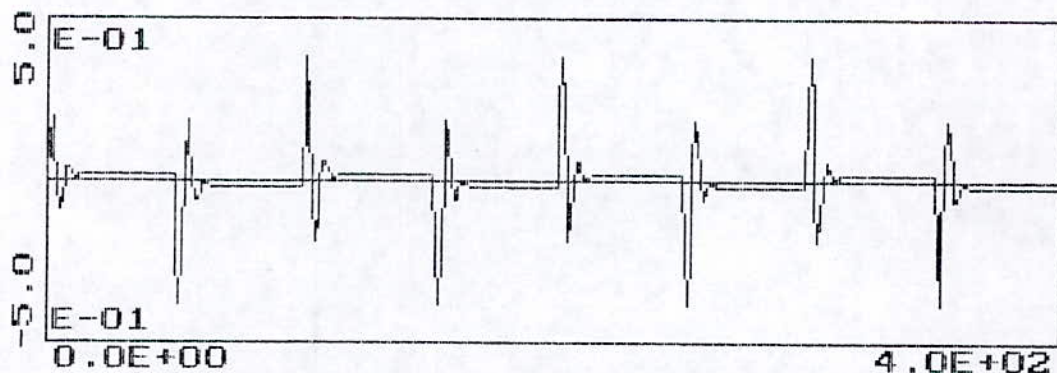


2. évolution de la commande.

Figure 1.9 Présentation sortie/ commande d'un système stable en B0 pour $N1=1$ $N2=13$ $NU=2$ $\lambda=0$.



1. évolution de la sortie du système.



2. évolution de la commande.

Figure 1.10 Présentation sortie/ commande d'un système stable en B0 pour $N1=1$ $N2=10$ $NU=8$ $\lambda=1$.

1.6.2 Etude d'un système à phase non-minimale.

Soit le système suivant:

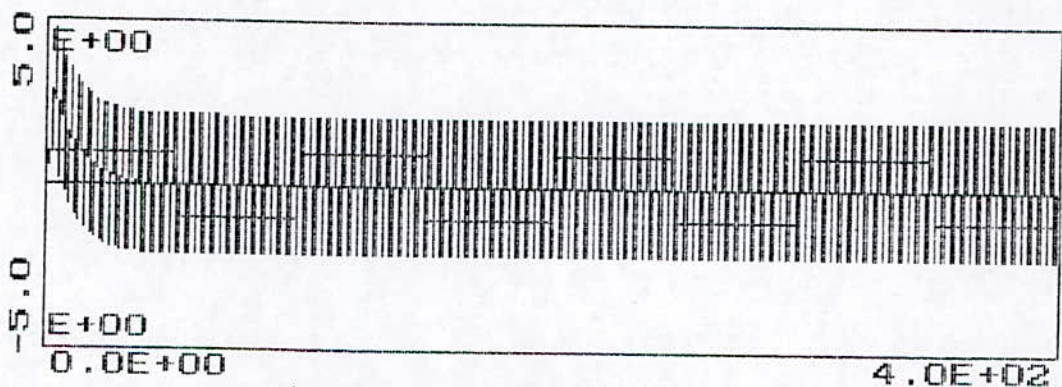
$$y(t) = \frac{(1+2q^{-1})q^{-1}}{1-0.9q^{-1}} u(t)$$

avec le zéro $z=-2$ (instable) et de pôle $p=0.9$ (stable).

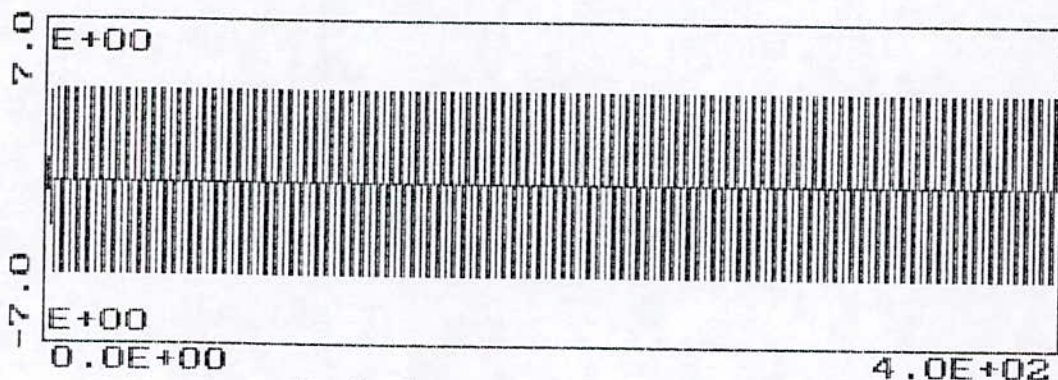
L'équation caractéristique de la fonction de transfert en boucle fermée est multiple du polynôme B commandée, soit par la GPC pour $N1=N2=NU=1$ $\lambda=0$ soit par la GMV. Ce qui exige donc pour avoir des commandes et des réponses stables que les zéros de $B(q^{-1})$ soient à l'intérieur du cercle unité. Pour pallier à ce problème dans la cas de la GMV l'introduction d'une pondération est nécessaire, par contre pour la GPC il suffit d'augmenter $N2$ ou tous simplement d'introduire la pondération.

La figure (1.11) montre la réponse du système pour $N1=N2=NU=1$, avec une limitation de la commande à quatre. L'augmentation de $N2$ illustrée sur les figures (1.12)(1.13) rend la réponse du système appréciable grâce à la stabilité du régulateur engendré par la commande à GPC.

L'introduction du paramètre λ peut être interprété comme un coefficient qui limite les fluctuations de commande. la sortie présente un dépassement relativement négligeable qui peut être éliminé en faisant une recherche approfondie sur le paramètre λ le plus adéquat, figure (1.14). Cette figure peut être comparée à la figure (1.12).

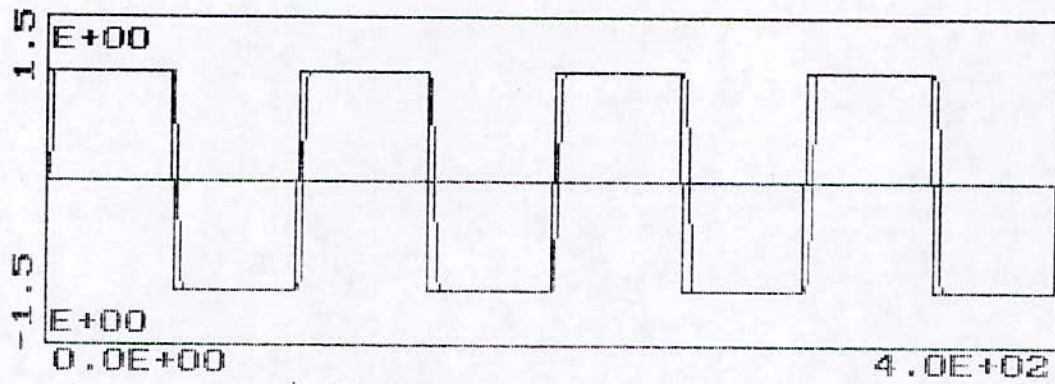


1. évolution de la sortie du système.

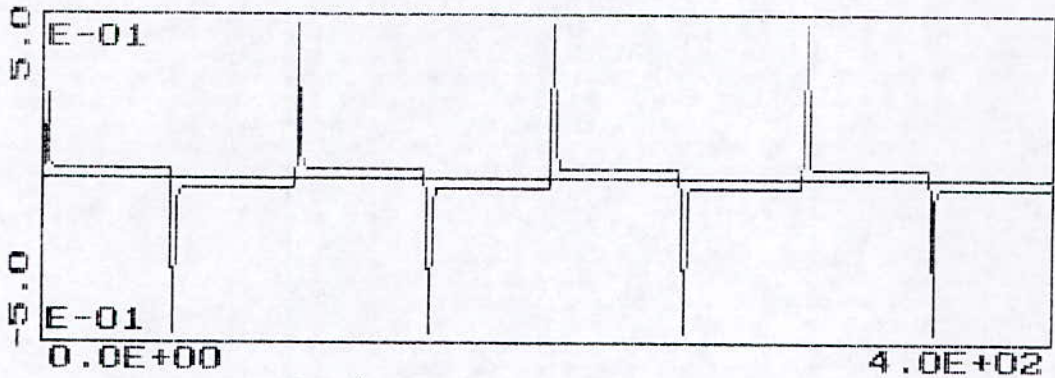


2. évolution de la commande.

Figure 1.11 Présentation sortie/ commande d'un système à PNM pour $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

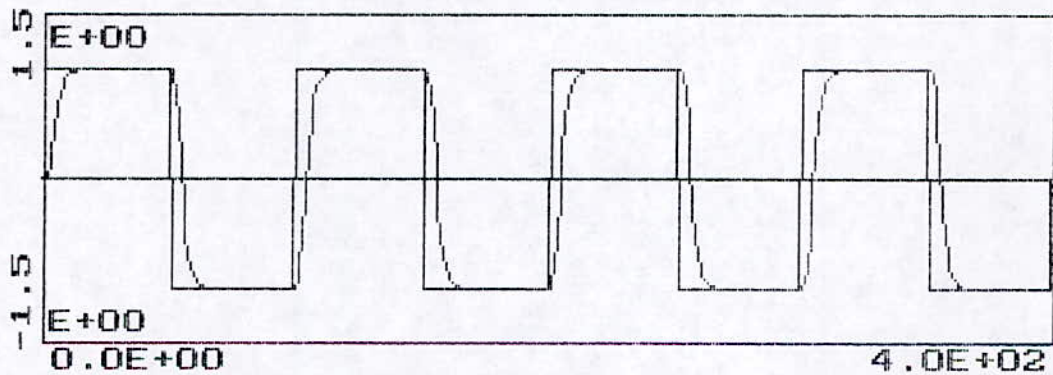


1. évolution de la sortie du système.

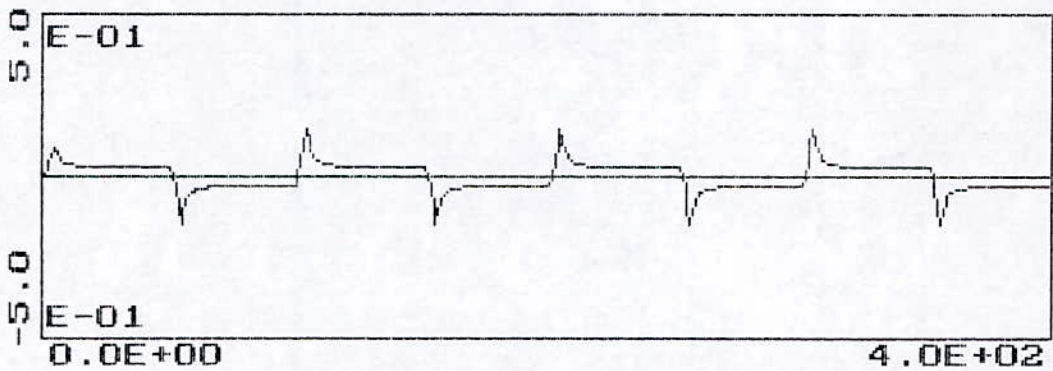


2. évolution de la commande.

Figure 1.12 Présentation sortie/ commande d'un système à PNM pour $N1=1$ $N2=2$ $NU=1$ $\lambda=0$.

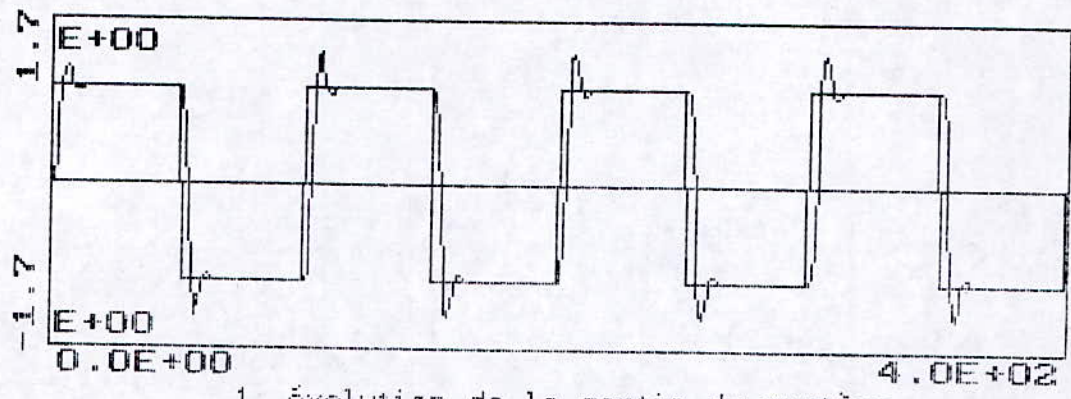


1. évolution de la sortie du système.

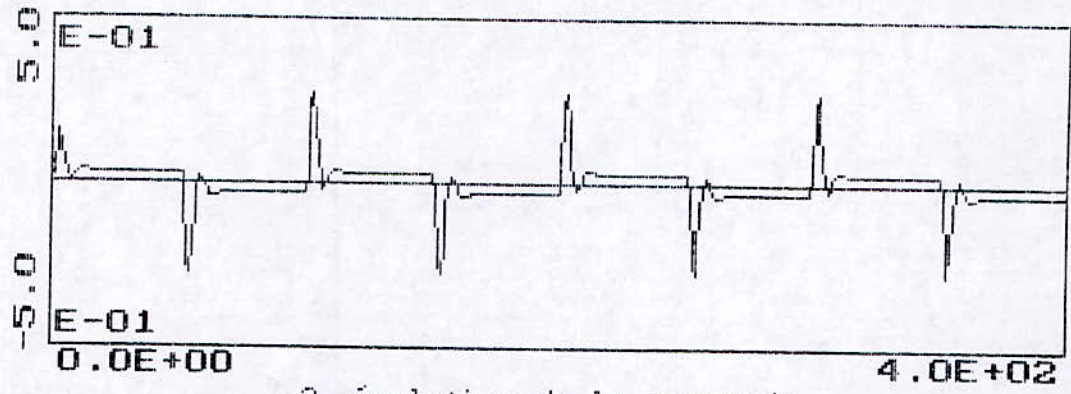


2. évolution de la commande.

Figure 1.13 Présentation sortie/ commande d'un système à PNM pour $N1=1$ $N2=5$ $NU=1$ $\lambda=0$.



1. évolution de la sortie du système.



2. évolution de la commande.

Figure 1.14 Présentation sortie/ commande d'un système à PNM pour $N1=1$ $N2=2$ $NU=1$ $\lambda=20$.

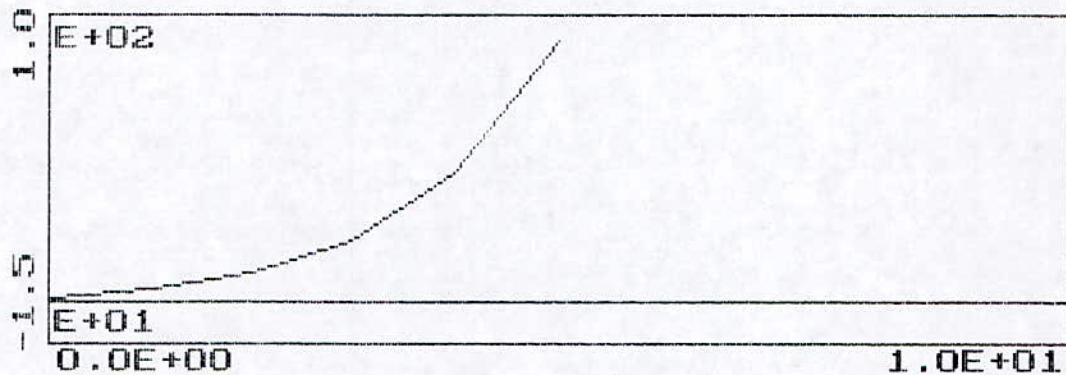
1.6.3 Etude d'un système instable en boucle ouverte.

Soit à étudier le système instable suivant:

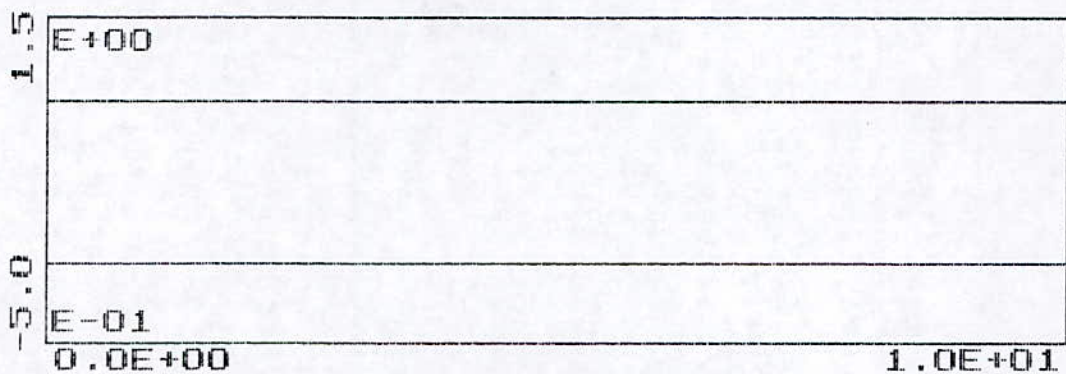
$$y(t) = \frac{(1-0.9)q^{-1}}{1+2q^{-1}} u(t-1)$$

présentant un pôle et un zéro respectivement de 0.9 et -2.

La réponse de ce système en BO excité pour une commande unitaire, est présenté sur la figure (1.15). Les résultats de simulation concernant ce système présentés sur les figures (1.16)(1.17)(1.18), montrent l'influence des paramètres $N2$ et λ sur la sortie et la commande. L'instabilité des pôles en BO n'ont aucun effet sur la dynamique en BF évitant ainsi toute instabilité du régulateur.

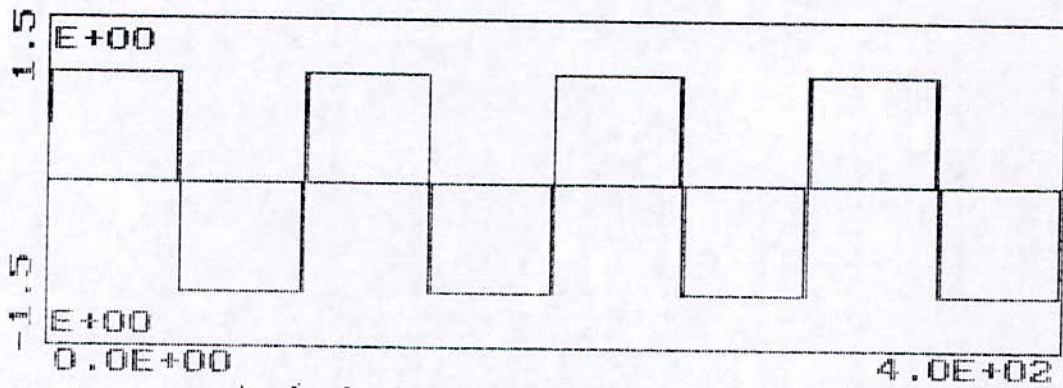


1. Sortie du système.

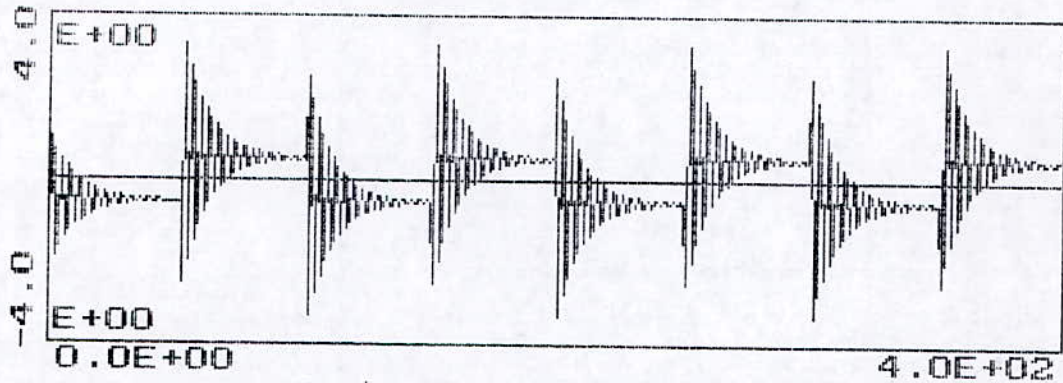


2. échelon de commande.

Figure 1.15 Réponse d'un système instable en boucle ouverte à une entrée échelon.

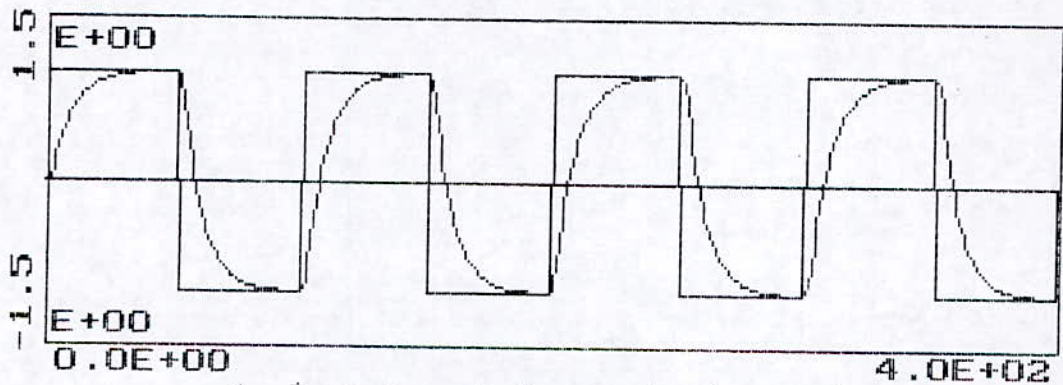


1. évolution de la sortie du système.

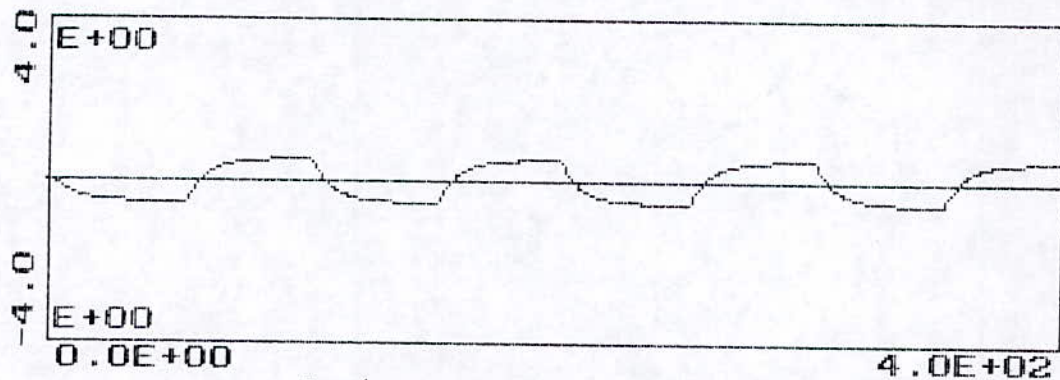


2. évolution de la commande.

Figure 1.16 Présentation sortie/ commande d'un système instable pour $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

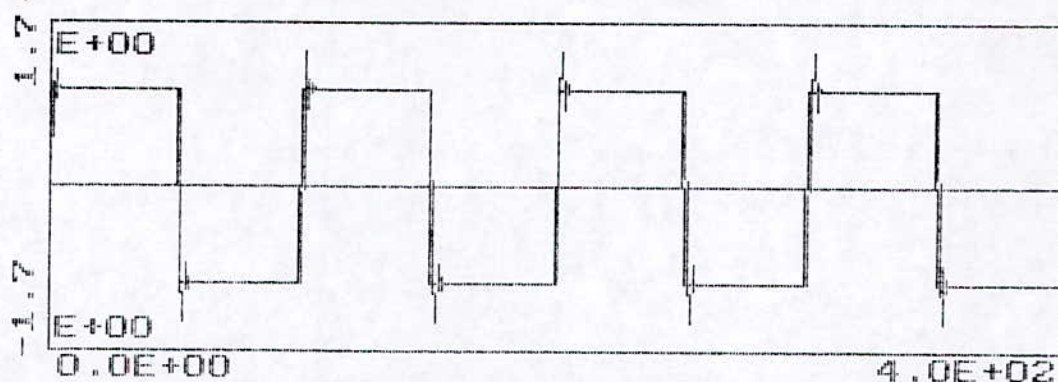


1. évolution de la sortie du système.

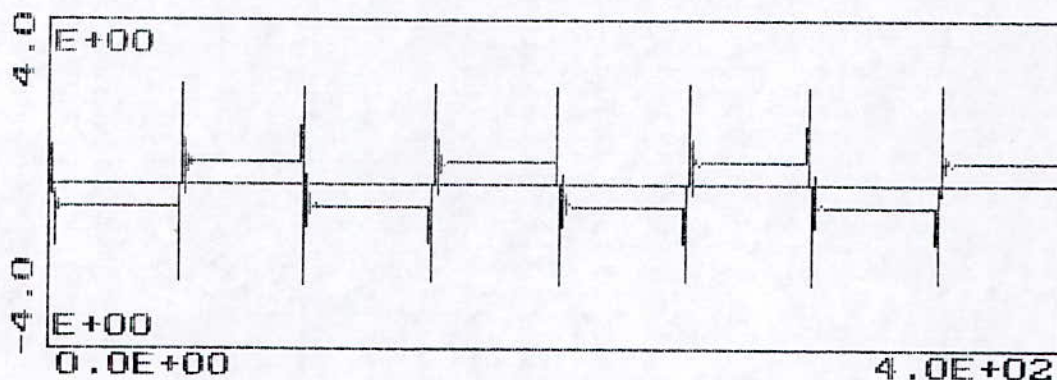


2. évolution de la commande.

Figure 1.17 Présentation sortie/ commande d'un système instable pour $N1=1$ $N2=4$ $NU=1$ $\lambda=0$.



1. évolution de la sortie du système.



2. évolution de la commande.

Figure 1.18 Présentation sortie/ commande d'un système instable pour $N1=1$ $N2=1$ $NU=1$ $\lambda=0.08$.

1.6.4 Etude d'un système instable et à phase non-minimale.

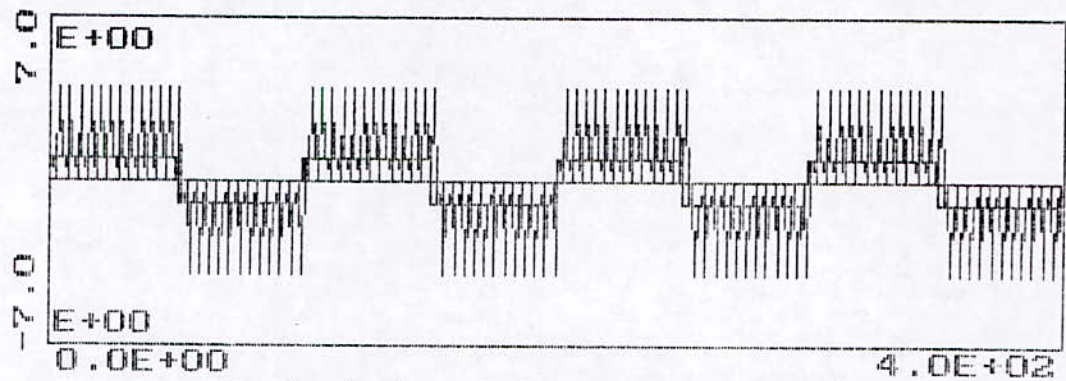
En combinant les deux systèmes étudiés précédemment on aboutit à un système instable et à phase non-minimale suivant:

$$y(t) = \frac{(1+2)q^{-1}}{1-2q^{-1}} u(t-1)$$

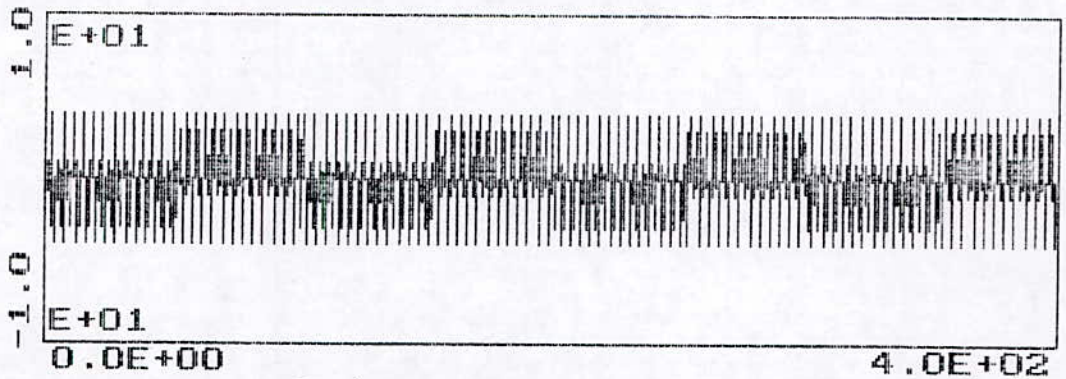
L'effet du paramètre $N2$ sur la commande et la sortie de ce système est présenté sur les figures (1.19)(1.20)(1.21). On remarque l'influence de l'augmentation de $N2$ qui est identique aux autres systèmes étudiés précédemment.

La recherche et le choix du paramètre λ est très délicat. En effet, ils varient sur un intervalle qu'il faut déterminer.

Pour le cas de ce système, λ varie entre 0.1 et 0.9, autre valeur du coefficient de pondération dégrade les performances en boucle fermée. Les figures (1.22)(1.23) présentent les réponses respectivement pour $\lambda=0.8$ et $\lambda=0.4$.

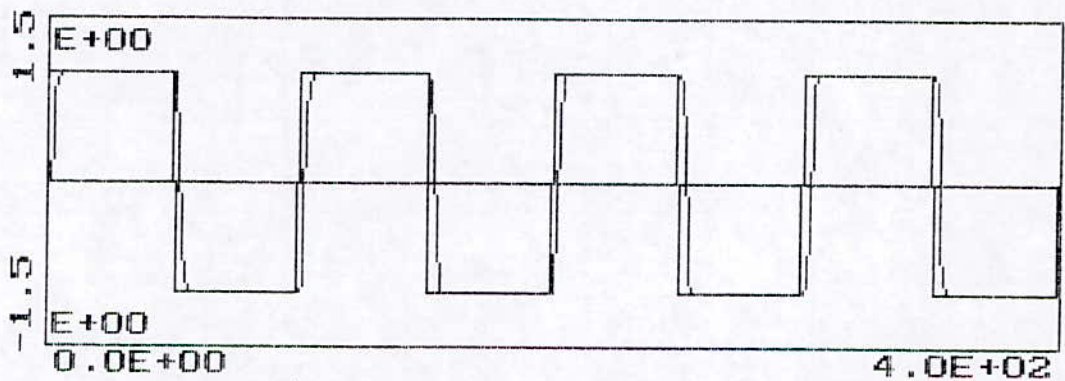


1. évolution de la sortie du système.

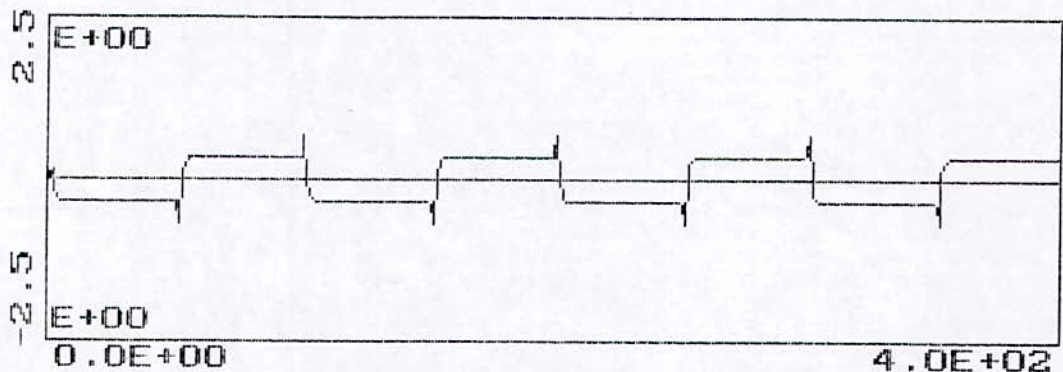


2. évolution de la commande.

Figure 1.19 Présentation sortie/ commande d'un système instable et à PNM pour $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

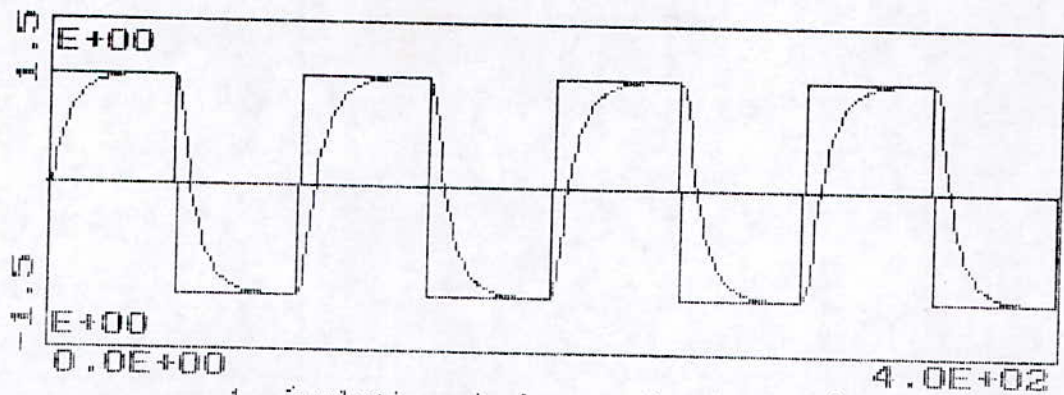


1. évolution de la sortie du système.

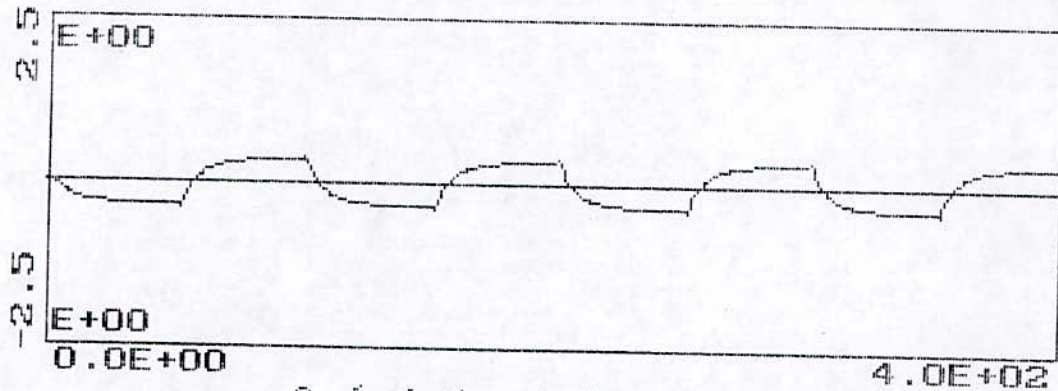


2. évolution de la commande.

Figure 1.20 Présentation sortie/ commande d'un système instable et à PNM pour $N1=1$ $N2=2$ $NU=1$ $\lambda=0$.

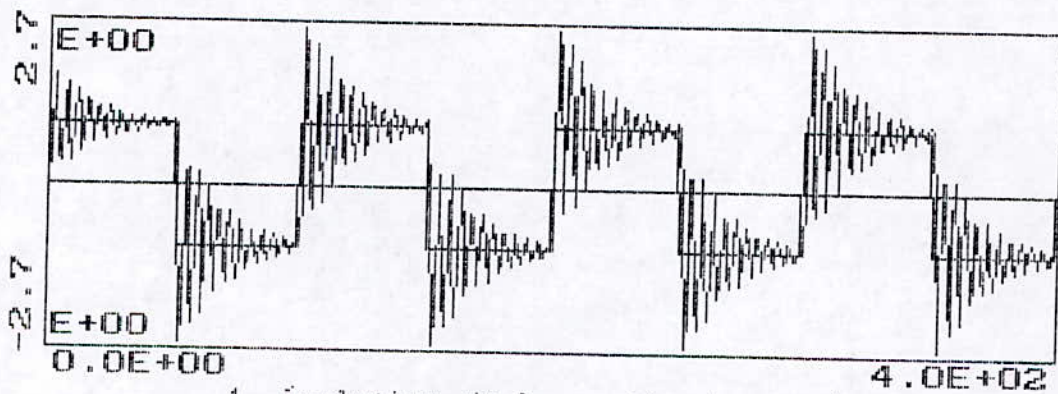


1. évolution de la sortie du système.

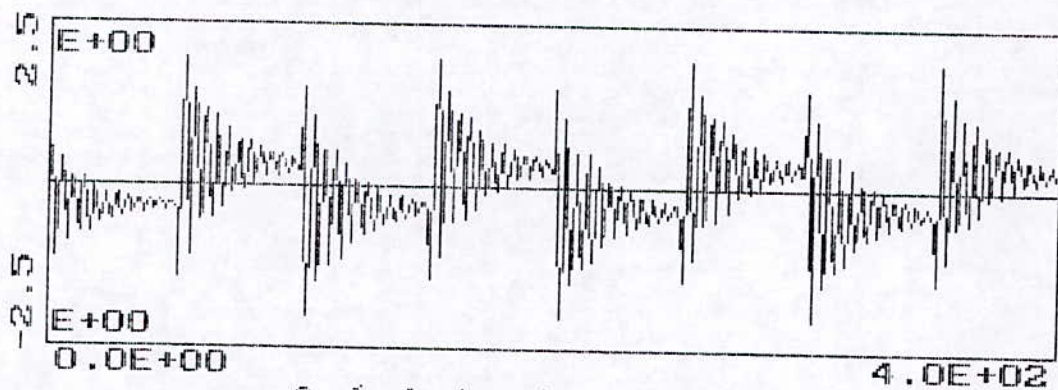


2. évolution de la commande.

Figure 1.21 Présentation sortie/ commande d'un système instable et à PNM pour $N1=1$ $N2=4$ $NU=1$ $\lambda=0$.

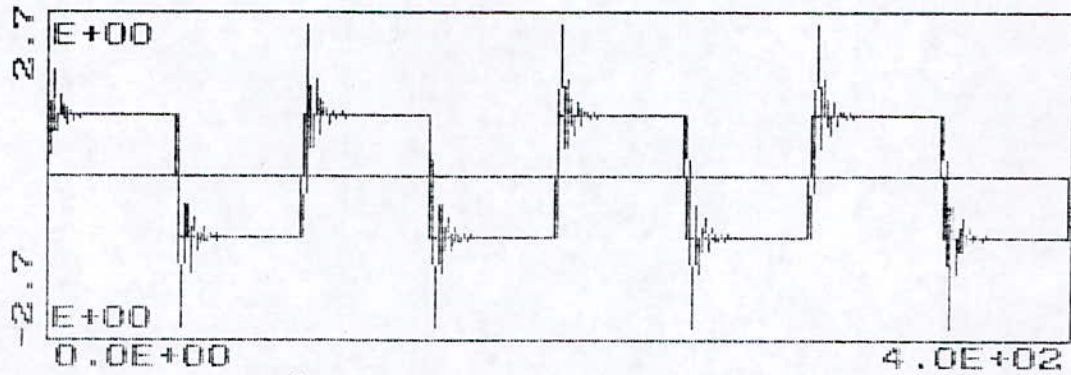


1. évolution de la sortie du système.

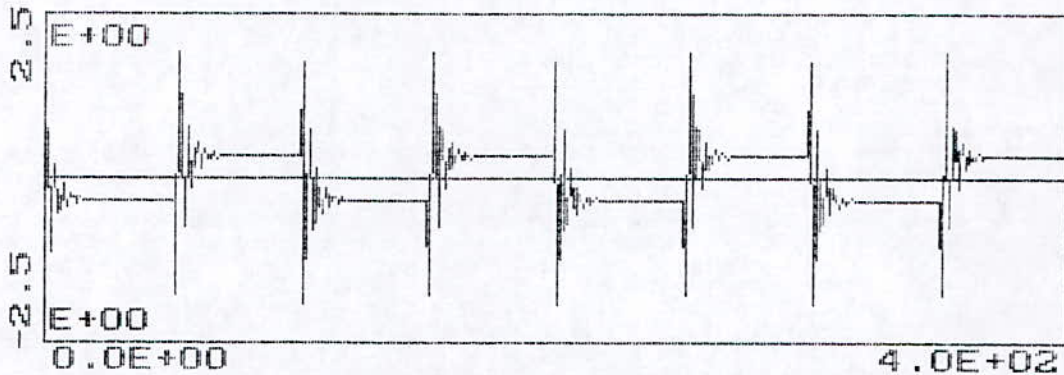


2. évolution de la commande.

Figure 1.22 Présentation sortie/ commande d'un système instable et à PNM pour $N1=1$ $N2=1$ $NU=1$ $\lambda=0.8$.



1. évolution de la sortie du système.



2. évolution de la commande.

Figure 1.23 Présentation sortie/ commande d'un système instable et à PNI pour $N1=1$ $N2=1$ $NU=1$ $\lambda=0.4$.

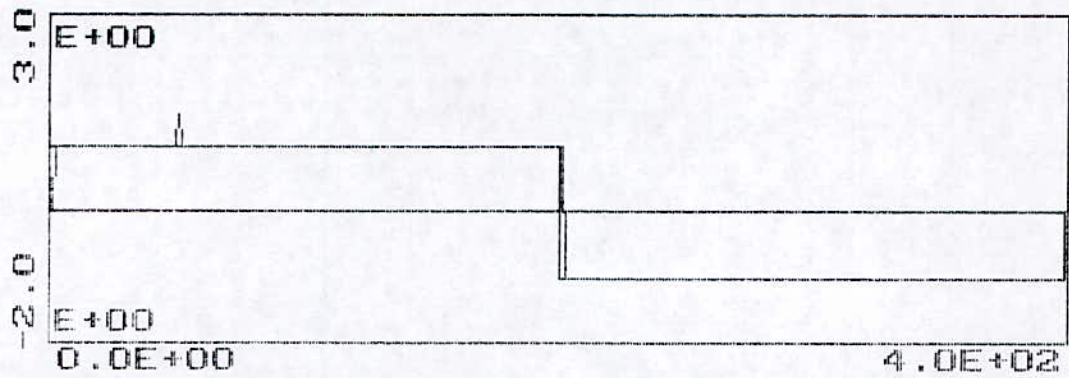
1.6.5 Influence de la sortie à une perturbation, en régime établi.

Pour tester l'efficacité de la GPC à des perturbations en échelon, on applique au cours de l'évolution de la sortie en régime établi un échelon de 0.5. Les résultats de simulation de la figure (1.24), réalisés sur le système stable, montrent que cette perturbation n'a aucun effet sur la sortie puisque le système poursuit son régime établi. Cette compensation est due à la présence d'un intégrateur dans le modèle CARIMA imposée par la GPC. Nous avons déjà noté que cet intégrateur joue un double rôle à savoir, réduire l'écart permanent et éliminer les perturbations constantes.

Les figures (1.25)(1.26) montrent l'effet de $N2$ et λ sur la sortie et la commande. La dynamique introduite par la perturbation dans la figure (1.26) peut être légèrement réduite en imposant une autre dynamique ce qui est illustré dans les résultats de simulation du chapitre 2.

La figure (1.27) montre que la GPC a bien compensée la perturbation impulsionnelle introduite au cours du régime permanent.

En ce qui concerne les perturbations non constante, de type intégrale, on remarque sur la figure (1.28) que la GPC n'est pas capable de s'imposer, Néanmoins une solution à ce problème peut être résolue en introduisant un double intégrateur dans le développement théorique de l'algorithme de commande GPC.

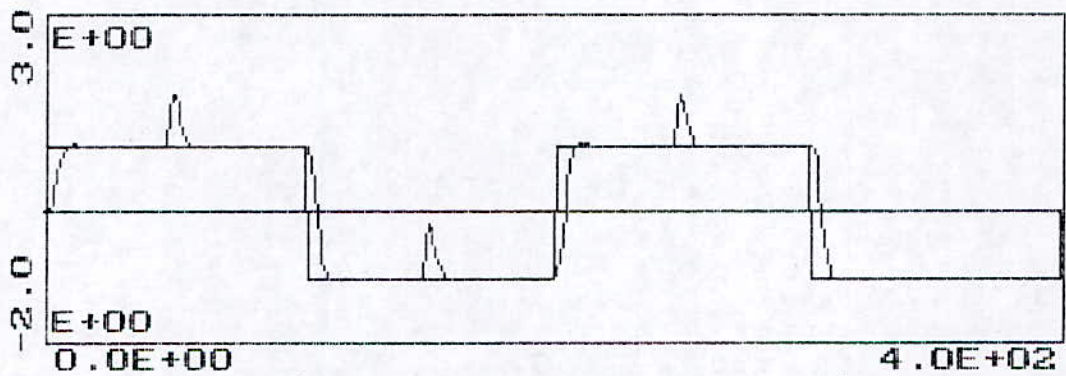


1. évolution de la sortie du système.

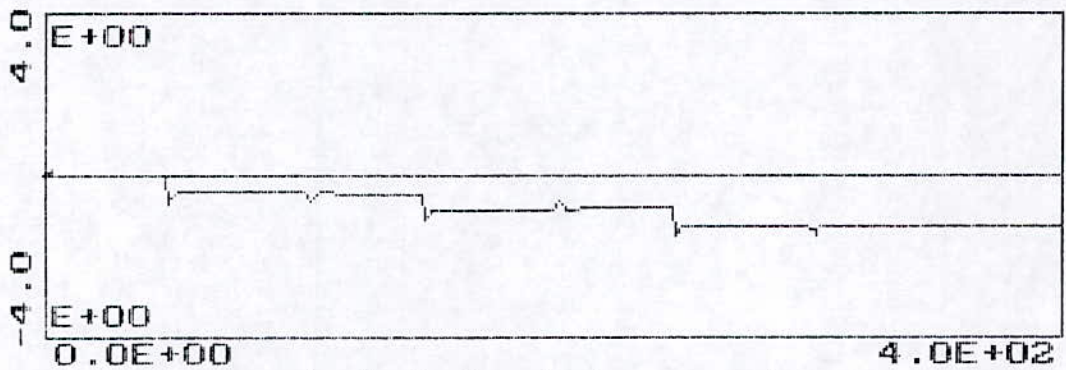


2. évolution de la commande.

Figure 1.24 Présentation sortie/ commande d'un système stable à une perturbation constante (0.5) pour $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

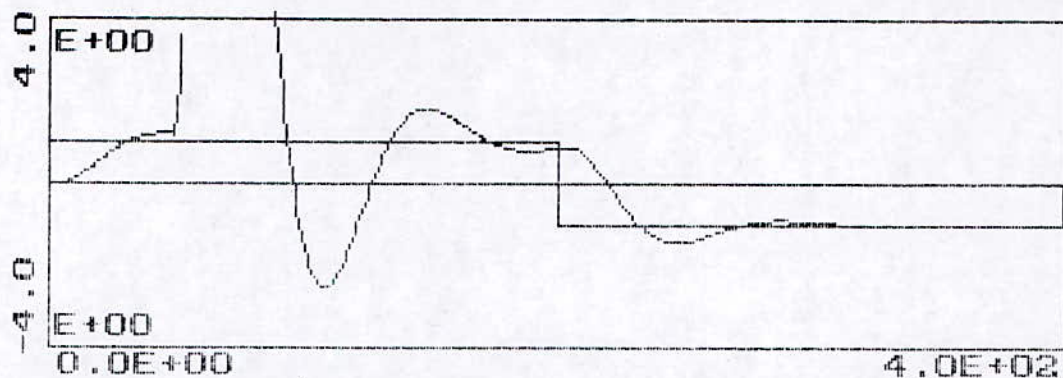


1. évolution de la sortie du système.

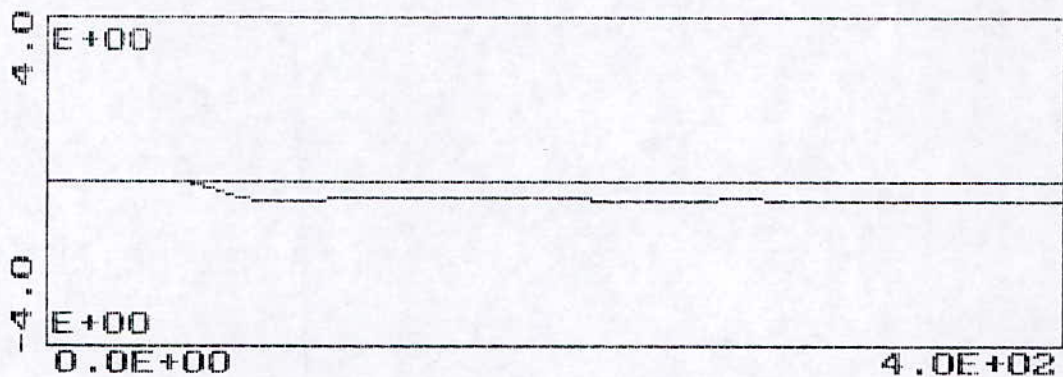


2. évolution de la commande.

Figure 1.25 Présentation sortie/ commande d'un système stable à une perturbation constante (0.5) pour $N1=1$ $N2=4$ $NU=1$ $\lambda=0$.

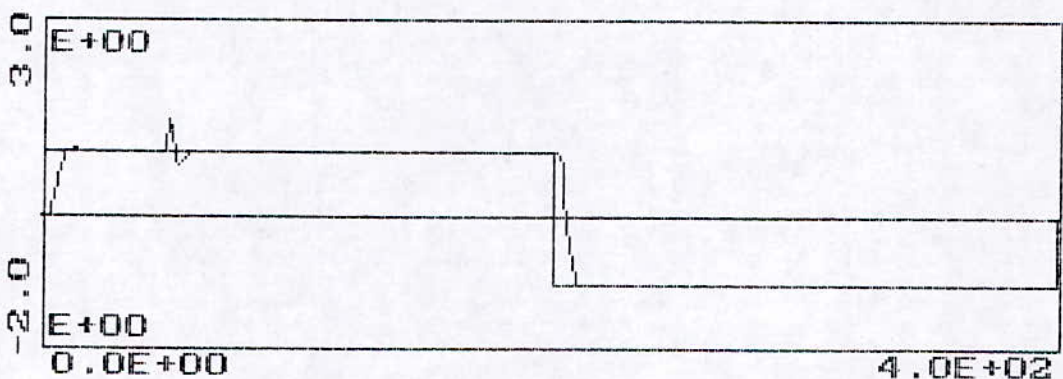


1. évolution de la sortie du système.

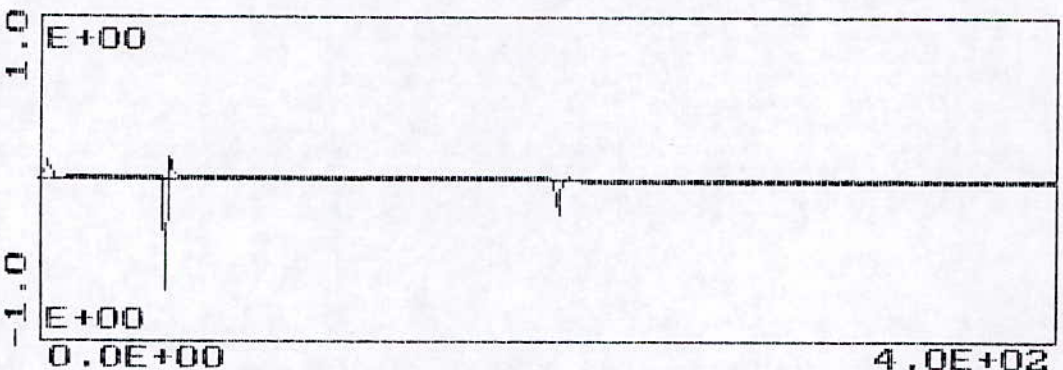


2. évolution de la commande.

Figure 1.26 Présentation sortie/ commande d'un système stable à une perturbation constante (0.5) pour $N1=1$ $N2=1$ $NU=1$ $\lambda=600$.

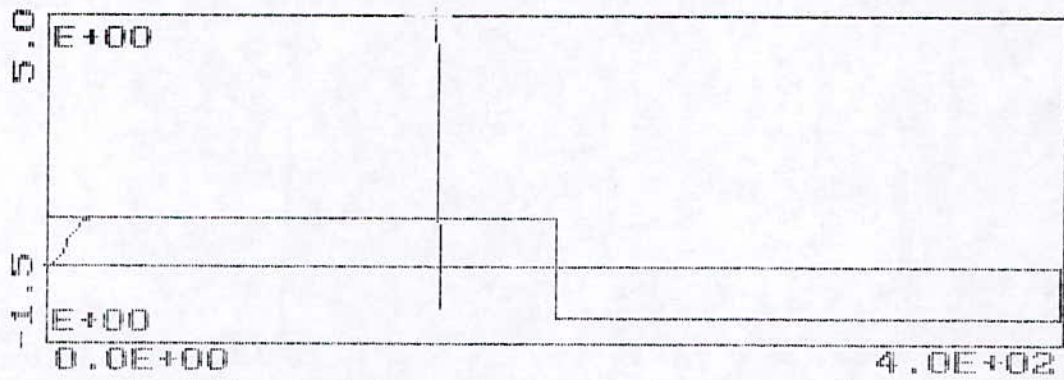


1. évolution de la sortie du système.

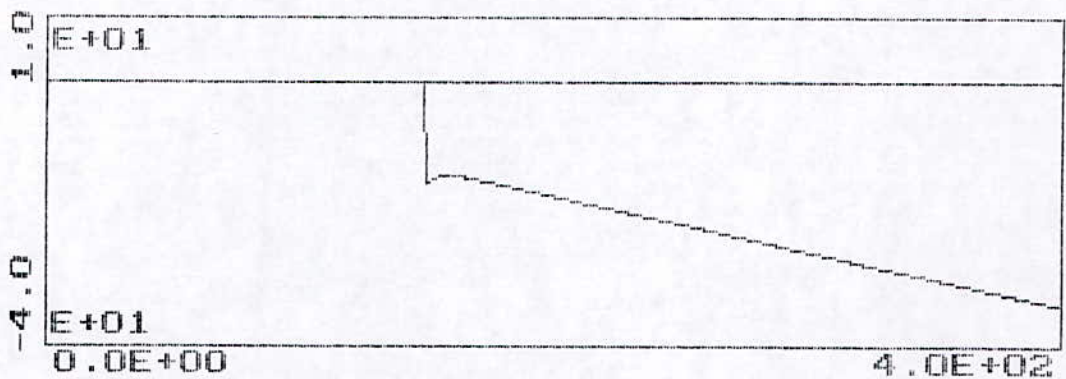


2. évolution de la commande.

Figure 1.27 Présentation sortie/ commande d'un système stable à une impulsion pour $N1=1$ $N2=4$ $NU=1$ $\lambda=0$.



1. évolution de la sortie du système.



2. évolution de la commande.

Figure 1.28 Présentation sortie/ commande d'un système stable à une perturbation de type intégral (pente=0.05) pour $N1=1$ $N2=6$ $NU=1$ $\lambda=0$.

1.7 Synthèse et discussion.

L'étude des quatre systèmes précédents nous a montré l'efficacité de la GPC envers les problèmes de pôles et de zéros instables. Son efficacité réside dans le choix des paramètres de synthèse $\lambda, N1, N2, NU$ qui demeure délicat.

Le choix de ces horizons et du coefficient de pondération a un aspect fondamental dans le succès de la commande prédictive généralisée (GPC). Plusieurs auteurs ont traité ce problème [2][14][27][28]. Ce qui nous amène à faire une discussion sur chaque paramètre.

1.7.1 Horizon initial de prédiction $N1$.

Le paramètre $N1$ est toujours déterminé par $N1.T_e \geq \text{retard}$ (T_e : période d'échantillonnage du système). En effet, si le retard du système est connu il n'y a pas de raisons à imposer à $N1$ une valeur inférieure au retard autrement des calculs inutile seront effectués. Dans le cas où le retard k est inconnu (cas adaptatif), $N1$ doit être mit à un et le degré de $B(q^{-1})$ augmente pour renfermer toutes valeurs possible de k .

donc:
$$N1 = \frac{\text{Retard du système}}{\text{période d'échantillonnage}}$$

1.7.2 Horizon maximal de prédiction $N2$.

Généralement $N2$ doit être supérieur au degré de B , ce qui est évident relativement au choix de $N1$. En effet, si $N1$ est supérieur au retard du système qui lui même en rapport avec le degré de B , ce qui veut dire que $N2$ est choisi supérieur ou égale à $N1$. (voir annexe 1.1 l'influence des paramètres de synthèse de la GPC). Dans la commande des systèmes, on privilégie le régime établi. En effet, ce régime peut être atteint par augmentation de $N2$ avec des commandes douces, mais au détriment du temps de réponse. En règle générale $N2$ est choisie de façon à ce qu'il corresponde au temps de montée du système.

donc:
$$N2 = \frac{\text{Temps de montée}}{\text{période d'échantillonnage}}$$

Remarques

1. Pour tous procédés stables, la dynamique de régulation devient plus rapide lorsque l'horizon de prédiction diminue.
2. Pour certains systèmes, une valeur de $N2$ assez petite, ne fournira pas un réglage stable.

1.7.3 Horizon de commande NU .

Pour les différents systèmes étudiés, on a constaté que la valeur de $NU=1$ donnait de bon résultats, l'augmentation de NU ne fait que rendre la commande plus active et importante, une autre augmentation ne fera presque aucune différence. Cette augmentation introduit une énorme complexité de calcul d'où un temps de calcul très important. Dans le cas de la commande numérique à paramètres connus, ce temps de calcul ne nous intéresse pas puisqu'ils se feront hors ligne. Donc on a intérêt dans le cas adaptatif (chapitre 3) à fixer NU le plus petit possible ($NU=1$) donnant des résultats satisfaisants et évitant ainsi l'inversion de la matrice $(G^T G + \lambda I)^{-1}$.

1.7.4 Coefficient de pondération de la commande λ .

L'atténuation des variations brusques de commande que l'on peut constater avec cette algorithm (GPC) est possible par l'introduction du paramètre λ . Il permet de donner plus ou moins de poids à la commande de façon à assurer de bonnes réponses lorsque le système de départ présente un risque d'instabilité.

Le coefficient λ ne peut être lié qu'au gain du système à régler, parceque plus le gain d'un système est grand plus la commande doit être pondérée, c'est à dire plus λ est important et inversement.

Donc λ doit être choisi d'une manière à amplifier ou à réduire l'effort de commande.

Conclusion.

Le chapitre est consacré au développement théorique de l'algorithme de commande prédictive généralisée (GPC) de base, suivi de la simulation de quatre type de système présentant chacun des particularités spécifiques. Les constatations et les remarques ont été élaborées à partir des fonctions de transfert discrètes à paramètres connus.

La commande numérique des processus exige que le calcul de la commande doit se faire pendant une partie courte de la période d'échantillonnage. Dans le cas de la GPC, la relation fournissant la commande se traduit par un algorithme simple. Une partie du calcul pouvant être faite "hors ligne", il ne reste à traiter en temps réel que l'acquisition de la sortie et le calcul de la commande. Cette simplicité permet à la commande d'avoir des applications dans de très nombreux domaines en particulier lorsque les constantes de temps des processus à commander sont très petites, nécessitant ainsi, l'emploi d'algorithme rapide. Le choix de la valeur de l'horizon de commande, permet de réduire d'une façon importante ce temps de calcul.

L'étude précédente nous a affirmé que l'algorithme de la GPC est capable de commander des systèmes simples et complexes et ceci grâce à un choix très particulier des paramètres de synthèse permettant d'élaborer des commandes douces et des sorties suivants exactement la référence (erreur de poursuite faible). Les performances des systèmes commandés par la GPC sont grandement influencés par le choix de la pondération de la commande et de l'horizon de prédiction. Une grande attention doit donc être accordée à leur choix, pour spécifier le comportement en boucle fermée (dynamique).

En fin, on a aussi montré que la GPC peut faire face à une variation de consigne et de perturbation en échelon sur la sortie réglée. Dans le cas de perturbation de type particulier et mesurable, la spécification du polynôme Δ (dans le modèle CARIMA) autre que $1-q^{-1}$, permet une compensation totale de cette perturbation et un écart statique nulle.

Chapitre 2

EXTENSION DE LA COMMANDE PREDICTIVE GENERALISEE

*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 2

EXTENSION DE LA COMMANDE PREDICTIVE GENERALISEE

Introduction.

Le développement de l'algorithme de base de la commande prédictive généralisée donne la possibilité aux paramètres de synthèse $N1, N2, NU$ et λ d'améliorer la qualité du réglage du processus à commander. Les performances en boucle fermée du système dépendent essentiellement des zéros du polynôme caractéristique. Les zéros de ce polynôme sont en fonction du choix des paramètres de synthèse.

Pour un choix spécifique des paramètres structurelles de la GPC ($N1, N2, NU$), la dynamique en boucle fermée dépendra seulement du coefficient de pondération de la commande λ . La variation de ce coefficient permet un changement des zéros du polynôme caractéristique. Ceci conduit à un placement de pôles empirique, car on ne peut avoir une relation explicite entre le coefficient λ et la position des pôles de la fonction de transfert en boucle fermée. Donc pour une structure fixe ($N1, N2, NU$), constante du régulateur, on ne dispose que d'un seul degré de liberté caractérisé par λ .

Dans le but d'offrir plus de degré de liberté pour une structure fixe, une extension de l'algorithme GPC s'impose. Elle se traduit par l'introduction de nouvelles pondérations permettant ainsi d'atteindre cet objectif. Ceci nous permet d'avoir accès au choix de l'équation caractéristique. Les polynômes ont été introduits dans le but d'éliminer l'erreur de poursuite (steady-state) et d'assigner les pôles en boucle fermée.

La minimisation de la variance de la sortie est assurée par un choix adéquat du polynôme caractéristique. Ce dernier doit être celui des perturbations ($c(q^{-1})$) [3]. La GPC de base donne une dynamique unité en boucle fermée, ce qui ne permet pas une flexibilité dans le choix de ce polynôme dit polynôme "d'observation". Il permet, d'une part, la minimisation du bruit de sortie, du système à commander, en fonction de la structure du modèle caractérisant le bruit. D'autre part, la spécification de ce polynôme (choisi par l'utilisateur) offre une possibilité de filtrage des quantités mesurables.

L'introduction de nouveaux termes de pondération permet de réduire l'effort de commande et de donner des commandes douces avec une dynamique réalisable.

Dans ce chapitre nous présenterons les différentes étapes de l'extension de la GPC, à savoir, présentation du critère, calcul du prédicteur, filtrage de commande et de sortie, résolution d'un autre type d'équation Diophantine, mise sous forme matricielle du prédicteur, calcul des régulateur pour différents types de pondération et nous terminerons par des résultats de simulation.

2.1 Critère de commande.

Le critère que la commande doit minimiser s'écrit sous la forme générale suivante [2] [23]:

$$J = E \left[\sum_{j=N1}^{N2} [P_j(q^{-1})y(t+j) - Q_j(q^{-1})w(t+j)]^2 + \sum_{j=1}^{NU} \lambda_j [\Delta u(t+j-1)]^2 \right] \quad (2.1)$$

P_j et R_j peuvent être interprétés comme un ensemble de polynômes étendus sur l'intervalle de prédiction, tel que:

$$\begin{aligned} P_j(q^{-1}) &= 1 + P_{1j}q^{-1} + \dots + P_{npj}q^{-np} \\ Q_j(q^{-1}) &= 1 + q_{1j}q^{-1} + \dots + q_{nrj}q^{-nr} \end{aligned} \quad \text{où} \quad P_j = \frac{P_{jN}}{P_{jD}}$$

2.2 Sortie auxiliaire.

L'idée de base dans l'introduction d'un polynôme de pondération de la sortie dans l'algorithme à variance minimale (VM) a été exploitée pour la GPC [16]. Cet algorithme utilise une sortie auxiliaire définie par:

$$\Psi(t) = P(q^{-1})y(t) \quad (2.2)$$

L'objectif à atteindre se résume à la spécification du comportement de cette sortie auxiliaire par rapport à la référence $w(t)$, ceci se traduit par l'équation suivante:

$$\Psi_1(t) = Q(q^{-1})w(t) \quad (2.3)$$

où R est un polynôme de pondération tel que $Q(1)=1$. Le comportement du système en boucle fermée est défini par les deux polynômes Q et P . Le polynôme P introduit la dynamique du système en boucle fermée tel que [16]:

$$y(t) = \frac{Q}{P}w(t) \quad (2.4)$$

Dans le cas de la GPC chaque prédiction de la sortie comporte sa propre dynamique. L'équation du modèle du prédicteur est obtenue par multiplication des deux membres de l'équation du modèle du système par le polynôme $P(q^{-1})$.

2.3 Calcul du prédicteur.

En se basant sur le même développement établi dans l'algorithme de base, la sortie du système à l'instant $t+j$ s'écrit [2]:

$$P_j y(t+j) = \frac{P_j B}{A} u(t+j-1) + \frac{P_j C}{A \Delta} \xi(t+j) \quad (2.5)$$

Pour séparer l'information connue de celle qui ne l'est pas, considérons l'équation Diophantine suivante:

$$P_{jN}C = E_j P_{jD} A \Delta + q^{-j} F_j \tag{2.6}$$

Obtenue à partir de la division de PC par $A\Delta$ jusqu'à l'ordre j .
Le critère devient:

$$J = E \left[\sum_{j=N_1}^{N_2} \frac{P_{jB}}{A} u(t+j-1) + E_j \xi(t+j) + \frac{F_j}{A \Delta P_{jD}} \xi(t) - R_j w(t+j) - \sum_{j=1}^{N_2} \lambda [\Delta u(t+j-1)]^2 \right] \tag{2.7}$$

En faisant le même raisonnement que dans la cas de l'algorithme de base, on aboutit au prédicteur optimal suivant:

$$\Psi(t+j/t) = G_j \Delta u^f(t+j-1) + \frac{F_j}{P_{jD}} y^f(t) \tag{2.8}$$

avec: $G_j = E_j B$; Δu^f ; $y^f = \frac{Y}{C}$.

2.4 Filtrage de la commande et de la sortie.

Les termes Δu^f et y^f correspondent aux valeurs filtrées des variations des signaux de commande et de sortie à travers le polynôme $C(q^{-1})$. Le but est de déterminer la commande qui minimise le critère J . Ce critère s'exprime en fonction des grandeurs non filtrées. Donc, pour la minimisation de ce critère par rapport aux variations du signal de commande, il faut faire apparaître de façon explicite ces variations [2]. Pour ce faire, nous allons diviser le polynôme $G_j(q^{-1})$ par $C(q^{-1})$. La partie non filtrées correspondra au quotient de cette division et la partie filtrée au reste.

D'où

$$\frac{G_j}{C} = G'_j + q^{-j} \frac{\Gamma_j}{C} \tag{2.9}$$

avec $\begin{cases} \deg[\Gamma_j] = \max(m, nc) \\ \deg[G'_j] = j-1 \end{cases}$

Nous aboutissant donc à un autre type d'équation Diophantine dont la résolution se présente comme suit [2][25]:

posons $C(q^{-1}) = T(q^{-1})$ pour spécifier son choix par l'utilisateur.

Soit la récursion suivante:

$$\begin{cases} G_j = G'_j T + q^{-j} \Gamma_j \\ G_{j+1} = G'_{j+1} T + q^{-(j+1)} \Gamma_{j+1} \end{cases} \tag{2.10}$$

Rappelons que $G_j = E_j B$, où:

$$E_j(q^{-1}) = e_0 + e_1 q^{-1} + \dots + e_{j-1} q^{-(j-1)} \tag{2.11}$$

En faisant la soustraction de G_{j+1} et G_j de l'équation (2.10), on obtient:

$$q^{-j}e_jB = q^{-j}g'_{j+1}T + q^{-j} [q^{-1}\Gamma_{j+1} - \Gamma_j] \tag{2.12}$$

Soit $\gamma_{j,i}$ le $i^{\text{ème}}$ coefficient du polynôme Γ_j ; en identifiant terme à terme les coefficients des éléments q^{-i} , on trouve l'algorithme suivant [2]:

$$\left\{ \begin{array}{l} g'_{j+1} = \frac{1}{t_0(e_j b_0 + \gamma_{(j)0})} \\ \gamma_{(j+1)i} = \gamma_{(j)i} + e_j b_i + g'_{j+1} t_i \\ \text{pour } i = 1, \dots, \max(\delta B, \delta T) \end{array} \right. \tag{2.13}$$

où t_i représente les coefficients du polynôme $T(q^{-1})$.

L'équation concernant le prédicteur optimal pourra donc être écrite sous la forme suivante.

$$\Psi(t+j/t) = G'_j \Delta u(t+j-1) + \Gamma_j \Delta u^f(t-1) + \frac{F_j}{P_{jD}} y^f(t) \tag{2.14}$$

Les polynômes G'_j et Γ_j sont obtenus par la résolution de la récursion précédente.

2.5 Mise sous forme matricielle du prédicteur.

Le développement de la sortie du système écrite à chaque instant dans le futur se présente sous la forme suivante [2]:

$$\left\{ \begin{array}{l} \Psi(t+1) = G'_1 \Delta u(t) + \Gamma_1 \Delta u^f(t-1) + \frac{F_1}{P_{1D}} y^f(t) + E_1 \xi(t+1) \\ \Psi(t+2) = G'_2 \Delta u(t+1) + \Gamma_2 \Delta u^f(t-1) + \frac{F_2}{P_{2D}} y^f(t) + E_2 \xi(t+2) \\ \vdots \\ \Psi(t+N2) = G'_{N2} \Delta u(t+N2-1) + \Gamma_{N2} \Delta u^f(t-1) + \frac{F_{N2}}{P_{N2D}} y^f(t) + E_{N2} \xi(t+N2) \end{array} \right. \tag{2.15}$$

et sous la forme matricielle en faisant apparaître la matrice G' .

$$\begin{bmatrix} \Psi(t+1) \\ \Psi(t+2) \\ \vdots \\ \Psi(t+N2) \end{bmatrix} = \begin{bmatrix} g'_0 & \cdot & \cdot & \cdot & \cdot & 0 \\ g'_1 & g'_0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g'_{N2-1} & g'_{N2-2} & \cdot & \cdot & g'_0 & \cdot \end{bmatrix} \times \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \cdot \\ \cdot \\ \cdot \\ \Delta u(t+NU-1) \end{bmatrix} \quad (2.16)$$

$$+ \begin{bmatrix} \Gamma_1 \Delta u^f(t-1) + \frac{F_1}{P_{1D}} y^f(t) + E_1 \xi(t+1) \\ \Gamma_2 \Delta u^f(t-1) + \frac{F_2}{P_{2D}} y^f(t) + E_2 \xi(t+2) \\ \cdot \\ \cdot \\ \cdot \\ \Gamma_{N2} \Delta u^f(t-1) + \frac{F_{N2}}{P_{N2D}} y^f(t) + E_{N2} \xi(t+N2) \end{bmatrix}$$

la forme condensée relatif au prédicteur optimal s'écrit:

$$\Psi = G' \tilde{U} + f \quad (2.17)$$

avec: $\Psi = [\Psi(t+1), \Psi(t+2), \dots, \Psi(t+N2)]^T$.
 G' : Matrice comportant les coefficients des polynômes G'_j .
 $\tilde{U} = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+NU-1)]^T$.
 $f = [f(t+1), f(t+2), \dots, f(t+N2)]^T$.

$$et \quad f(t+j) = \Gamma_j \Delta u^f(t-1) + \frac{F_j}{P_{jD}} y^f(t) + E_j \xi(t+j) \quad (2.18)$$

La minimisation du critère aboutit à un vecteur de commande qui s'écrit [2]:

$$\tilde{U} = \Delta u = (G'^T G' + \lambda I)^{-1} G'^T (W - f) \quad (2.19)$$

avec: $\begin{cases} \lambda = \lambda_{0j} I & (I: \text{matrice identité de dimension } NU \times NU) \\ W = [Q_1 w(t+1), Q_2 w(t+2), \dots, Q_{N2} w(t+N2)] \end{cases}$

2.6 Détermination du régulateur RST .

En écrivant l'équation (2.19) sous la forme canonique $Su = Tw - Ry$, on obtient les expressions donnant les polynômes R, S et T .
 Les différents polynômes Q_j et P_{jN} se traduit par le fait que chaque sortie prédite est pondérée par son propre polynôme. C'est à dire que chaque sortie prédite possède sa propre dynamique.

Dans la suite du raisonnement, on considérait que $P_{jD}=1$ pour la simplicité des calculs. Suivant la nature de la perturbation qui affecte le processus, on peut choisir le polynôme C égal soit à un ou à une valeur polynomiale quelconque. Plusieurs structures du régulateur RST sont obtenues pour différentes valeurs des polynômes de pondération et celui de perturbations. Le tableau (N°1) suivant présente l'expression analytique des polynômes de la structure canonique pour différent cas.

C	P_j	Q_j	P_{jL}	$R(q^{-1})$	$S(q^{-1})$	$T(q^{-1})$
1	1	1	1	$\sum_{i=N1}^{N2} \gamma_{1i} F_i$	$1+q^{-1} \sum_{i=N1}^{N2} \gamma_{1i} g_{fin,i}$	$\sum_{i=N1}^{N2} \gamma_{1i} q^i$
1	P	1	1	$\sum_{i=N1}^{N2} \gamma'_{1i} F'_i$	$1+q^{-1} \sum_{i=N1}^{N2} \gamma'_{1i} g'_{fin,i}$	$\sum_{i=N1}^{N2} \gamma'_{1i} q^i$
C	1	1	1	$\sum_{i=N1}^{N2} \gamma_{1i} F_i$	$C+q^{-1} \sum_{i=N1}^{N2} \gamma_{1i} \Gamma_i$	$C \sum_{i=N1}^{N2} \gamma_{1i} q^i$
C	P	1	1	$\sum_{i=N1}^{N2} \gamma'_{1i} F'_i$	$C+q^{-1} \sum_{i=N1}^{N2} \gamma'_{1i} \Gamma'_i$	$C \sum_{i=N1}^{N2} \gamma'_{1i} q^i$
C	P_{jN}	P_{jN}	P_{jD}	$\sum_{i=N1}^{N2} \gamma_{1i} \frac{F_i}{P_{iD}}$	$C+q^{-1} \sum_{i=N1}^{N2} \gamma_{1i} \Gamma_i$	$C \sum_{i=N1}^{N2} P_{iN} \gamma_{1i} q^i$
C	P_N	P_N	P_D	$\frac{1}{P_D} \sum_{i=N1}^{N2} \gamma_{1i} F_i$	$C+q^{-1} \sum_{i=N1}^{N2} \gamma_{1i} \Gamma_i$	$C P_N \sum_{i=N1}^{N2} \gamma_{1i} q^i$

Tableau (N°1): Tableau récapitulatif des régulateurs RST .

2.7 Equivalence entre la GPC et la GMV.

Après avoir étudié l'extension de l'algorithme GPC on a pu à travers un exemple montrer que la loi de commande GPC peut être équivalente à la GMV et ceci par un choix bien spécifique des paramètres de synthèse.

Soit un système stable et à phase non-minimale défini par les polynômes A et B suivant [25]:

$$A(q^{-1}) = 1+0.8q^{-1}+0.15q^{-2}$$

$$B(q^{-1}) = 1+0.7q^{-1}$$

La prédiction à un pas, c'est à dire $N1=N2=NU=1$, de la sortie auxiliaire du système s'exprime en fonction des polynômes E_1, F_1 et $G_1=E_1B_1$. en utilisant le développement

de l'algorithme récursif de la résolution de l'équation Diophantine, on aboutit aux expressions de E_1, F_1 et G_1 .

$$\begin{cases} E_1=1 \\ F_1=0.2+0.65q^{-1}+0.15q^{-2} \\ G_1=1+0.7q^{-1} \end{cases}$$

le prédicteur optimal concernant la sortie auxiliaire $\Psi(t)$, ($\Psi(t)=P(q^{-1})y(t)$) est:

$$\hat{\Psi}(t+1)=(1+0.7)\Delta u(t)+F_1y(t)$$

L'expression de la commande $\Delta u(t)$ obtenue à partir de l'équation (2.19) devient:

$$\Delta u(t) = \frac{w - F_1y(t)}{1+0.7q^{-1}}$$

En identifiant cette équation à celle de la forme canonique du régulateur RST, on obtient:

$$\begin{cases} S(q^{-1})=1+0.7q^{-1} \\ R(q^{-1})=F(q^{-1}) \\ T(q^{-1})=1 \end{cases}$$

Sachant que la fonction de transfert en boucle fermée s'écrit: $y(t) = \frac{BT}{SA\Delta + BR} w(t)$ et que $q^{-1}F_1 = P - E_1A\Delta$ on a:

$$y_{BF}(t) = \frac{1}{P} w(t)$$

La dynamique du système en boucle fermée est assurée par la stabilité du polynôme $P(q^{-1})$, (résultat semblable obtenu dans le cas de la GMV) [7]

2.8 Résultats de simulation.

l'exemple traité précédemment montre que, la dynamique en BF est identique à celle de la GMV, par le choix des paramètres de synthèse de la GPC. Pour illustrer l'effet de cette dynamique nous présentons dans cette partie, les résultats de simulation relatifs à l'extension de la GPC.

Cette extension nous permet de stabiliser le système en se basant sur l'analyse polynomiale du système en BF et en traitant le problème comme celui d'un placement de pôles. En effet, le polynôme de pondération de la sortie, choisi par l'utilisateur, apparaît dans l'équation caractéristique du système en BF. Il convient donc, de faire un choix judicieux de ce polynôme, pour éviter toute instabilité du système, sachant qu'il traduit la dynamique de sortie.

Généralement le polynôme de pondération $P(q^{-1})$ doit être choisi de telle sorte que ces zéros soient à l'intérieur d'une zone de stabilité adéquatement choisie, autrement des résultats très mauvais seront observés (voir résultats de simulation).

Nous présentons dans ce qui suit l'influence des paramètres de la GPC, à savoir $N1, N2, NU, \lambda$ et $P(q^{-1})$ sur la dynamique de sortie et de commande.

Pour $N1=N2=NU=1, \lambda=0$ et $A=0.8q^{-1}+0.4q^{-2}$; $B=1-1.7q^{-1}+1.72q^{-2}$
on a:

La figure 2.1 présente la sortie et la commande pour un polynôme de pondération
 $P(q^{-1})=1-0.5q^{-1}+0.06q^{-1}$ ayant pour zéros $z_1=0.3$ et $z_2=0.2$.

La figure 2.2 illustre les résultats obtenus concernant un
polynôme $P(q^{-1})=1-0.5q^{-1}+0.5q^{-1}$ et ayant pour zéros complexe
 $z_{1,2}=0.25\pm 0.66i$.

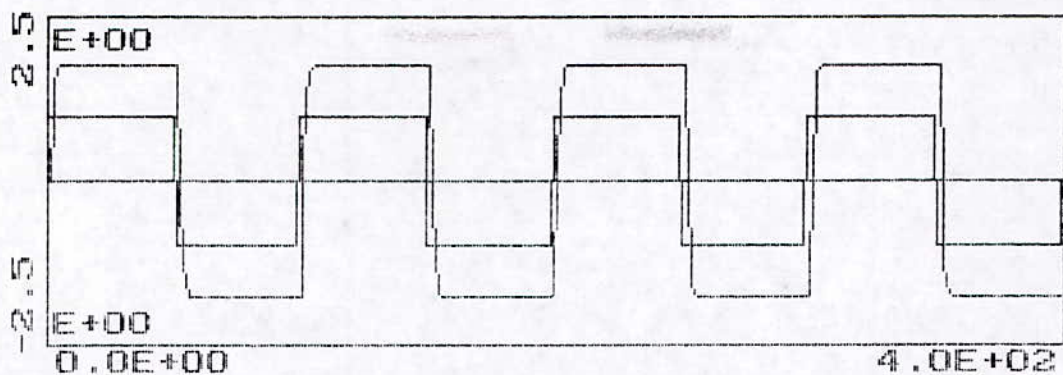
La figure 2.3 montre les résultats dans le cas où le polynôme choisi est
 $P(q^{-1})=1-0.05q^{-1}+0.06q^{-1}$ contenant les zéros suivants:
 $z_{1,2}=0.025\pm 0.24i$.

La figure 2.4 consigne des résultats obtenus pour un polynôme $P(q^{-1})$ stable
 $P(q^{-1})=1-0.001q^{-1}$ avec un zéro $z_0=0.001$.

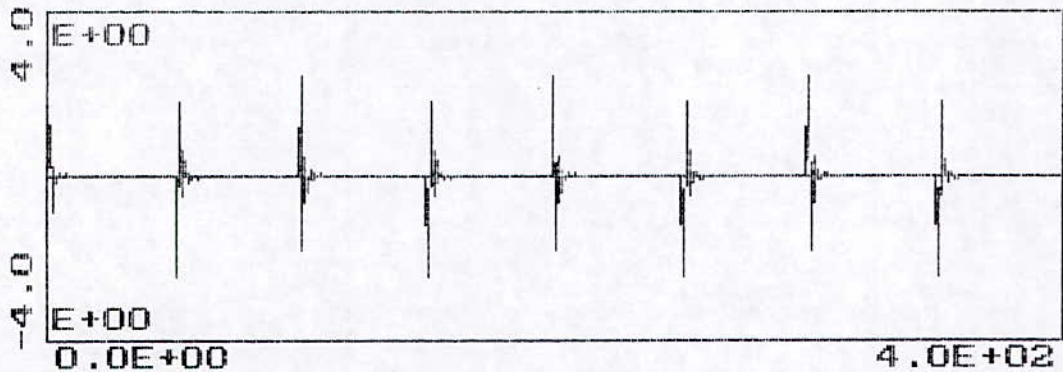
Les résultats de simulation sur un procédé stable mais pour des zéros du polynôme de pondération instables sont présentés sur les figures (2.5)(2.6) ayant respectivement pour zéros $z_1=0.03, z_2=1.96$ et $z_1=0.5, z_2=-1$. La limitation de la commande a été imposée pour le premier à 0.02 et 2 pour le deuxième, s'il n'y avait pas de limitation le système aura divergé.

L'influence des autres paramètres de la GPC sur la dynamique de sortie et la commande est présentée sur les figures (2.7)(2.8)(2.9), en gardant la même échelle pour la commande, on remarque que l'effort de commande a considérablement diminué.

La dynamique imposée par l'effet de la perturbation constante illustrée sur la figure (1.26) du chapitre 1 peut être atténuée en imposant une autre dynamique à savoir $P(q^{-1})=1+0.05q^{-1}-0.2q^{-1}$ présenté sur la figure (2.10).

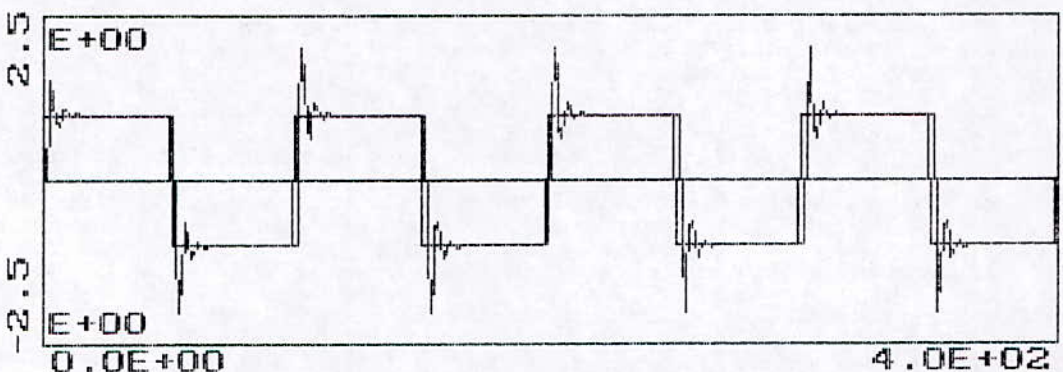


1. évolution de la sortie du système.

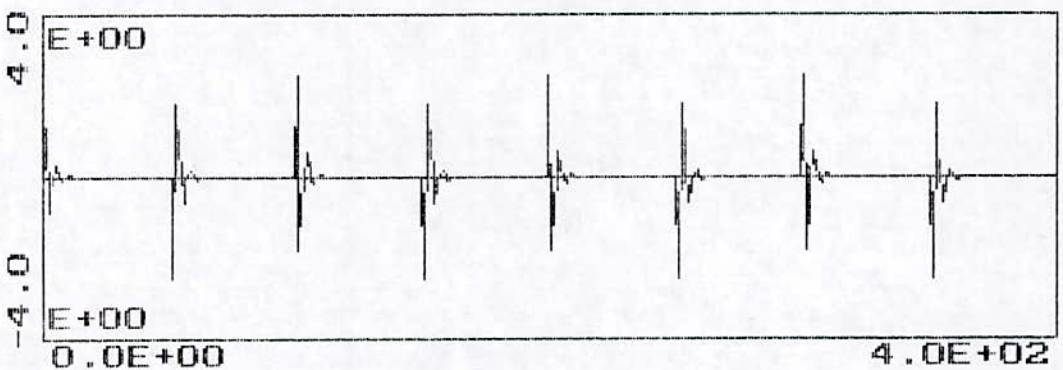


2. évolution de la commande.

Figure 2.1 Présentation sortie/ commande d'un système stable pour $P(q) = 1 - 0.5q^{-1} + 0.06q^{-2}$, $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

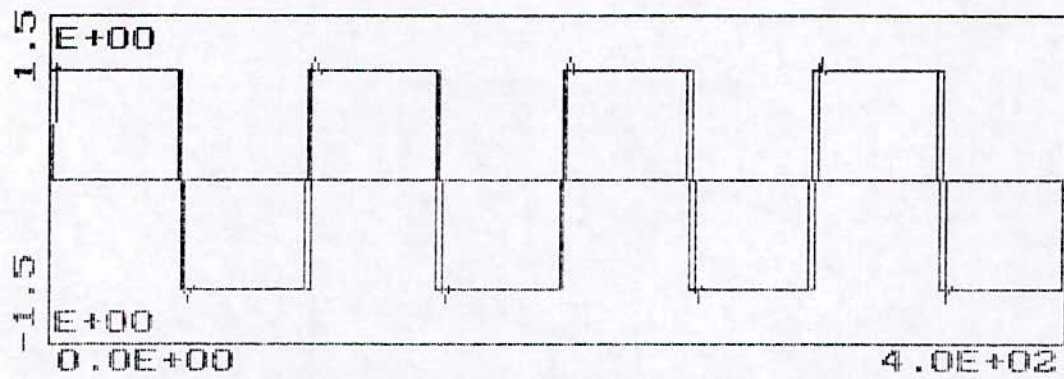


1. évolution de la sortie du système.

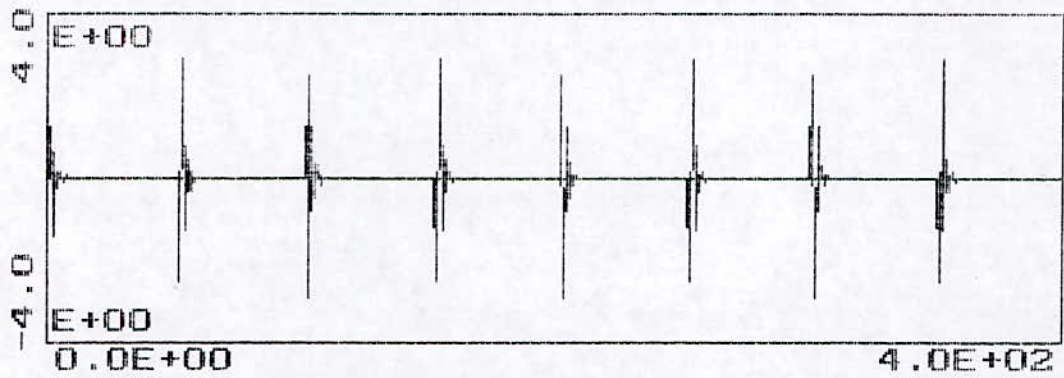


2. évolution de la commande.

Figure 2.2 Présentation sortie/ commande d'un système stable pour $P(q) = 1 - 0.5q^{-1} + 0.5q^{-2}$, $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

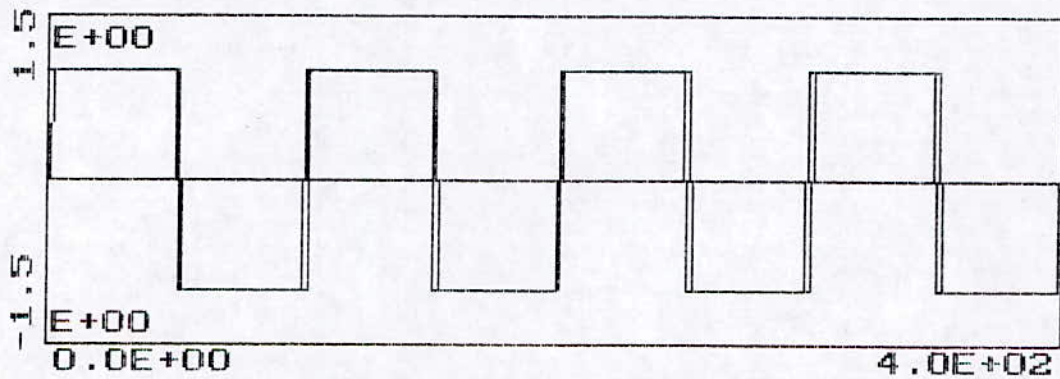


1. évolution de la sortie du système.

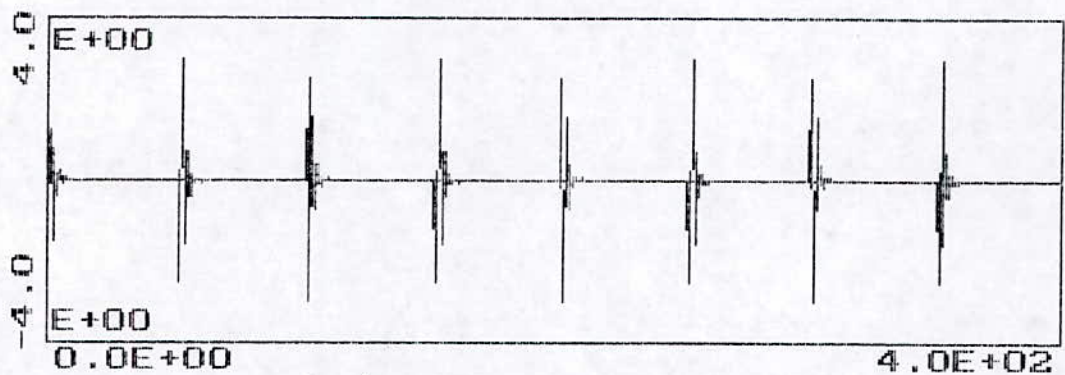


2. évolution de la commande.

Figure 2.3 Présentation sortie/ commande d'un système stable pour $P(q) = 1 - 0.05q^{-1} + 0.06q^{-2}$, $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

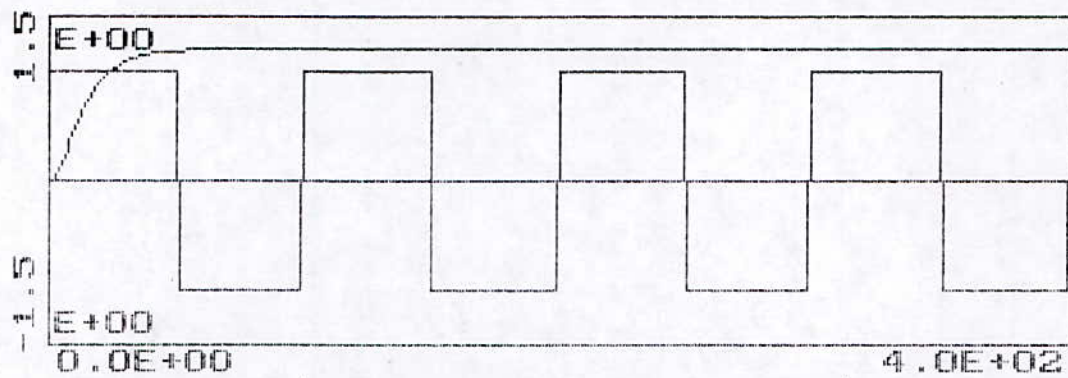


1. évolution de la sortie du système.

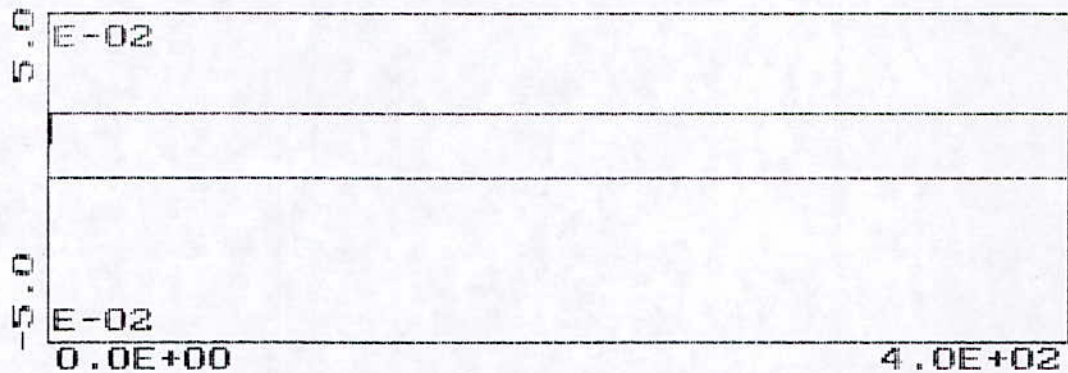


2. évolution de la commande.

Figure 2.4 Présentation sortie/ commande d'un système stable pour $P(q) = 1 - 0.001q^{-1}$, $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

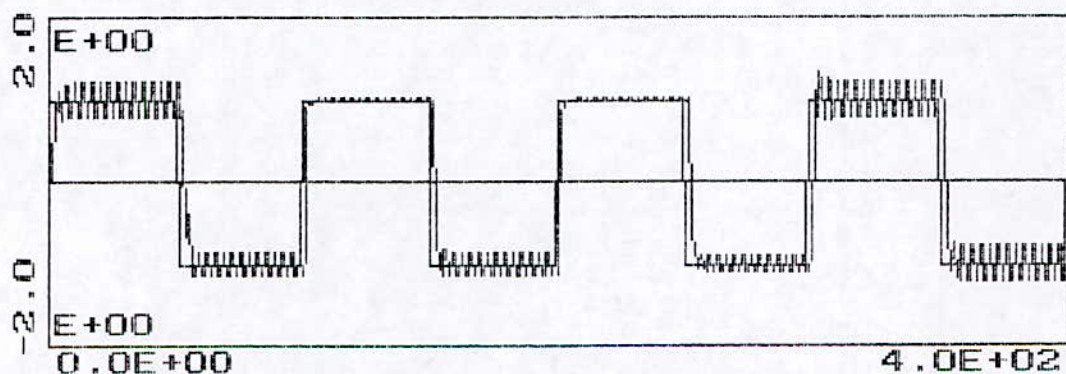


1. évolution de la sortie du système.

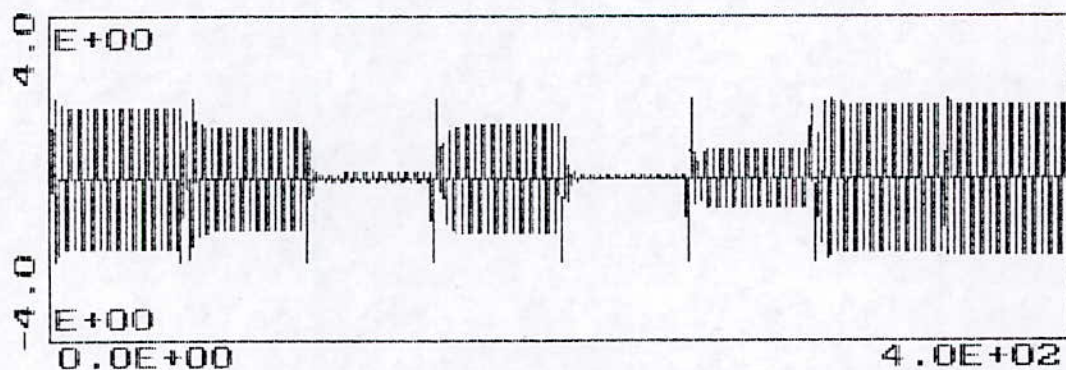


2. évolution de la commande.

Figure 2.5 Présentation sortie/ commande d'un système stable pour $P(q) = 1 - 2q^{-1} + 0.06q^{-2}$, $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

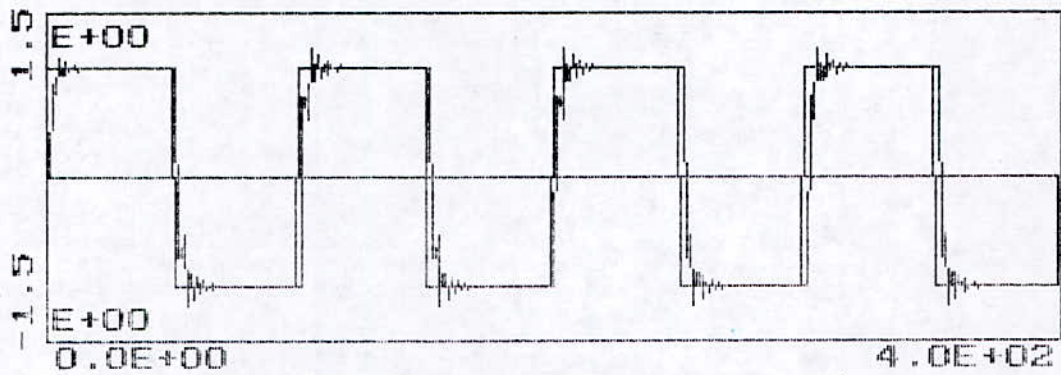


1. évolution de la sortie du système.

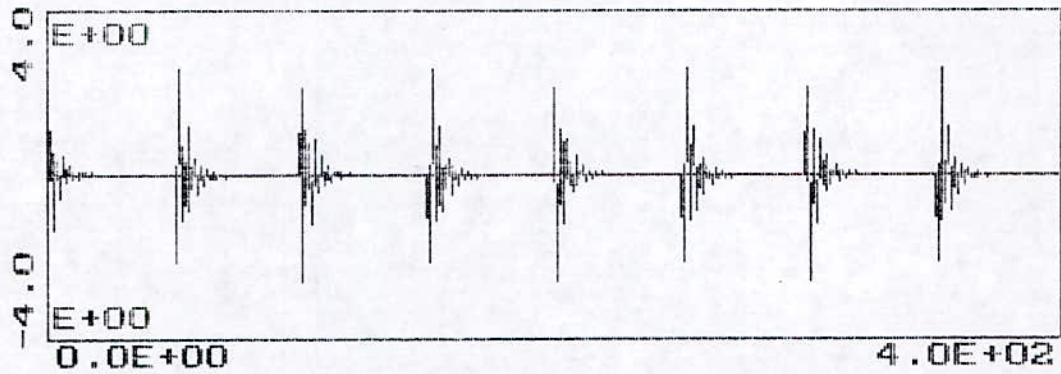


2. évolution de la commande.

Figure 2.6 Présentation sortie/ commande d'un système stable pour $P(q) = 1 + 0.5q^{-1} - 0.5q^{-2}$, $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

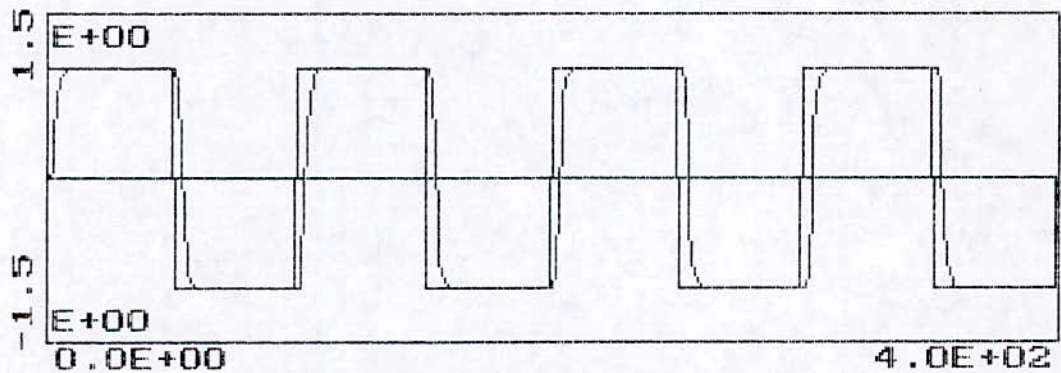


1. évolution de la sortie du système.

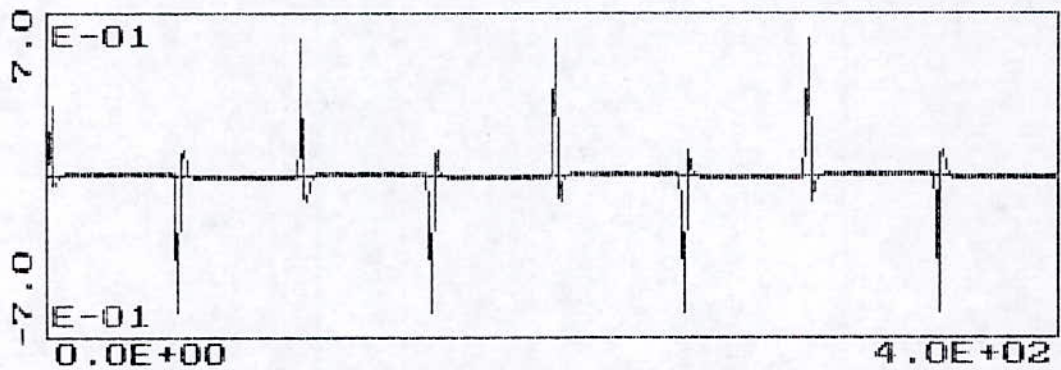


2. évolution de la commande.

Figure 2.7 Présentation sortie/ commande d'un système stable pour $P(q) = 1 + 0.5q^{-1} - 0.5q^{-2}$, $N1=1$ $N2=1$ $NU=1$ $\lambda=0.1$.

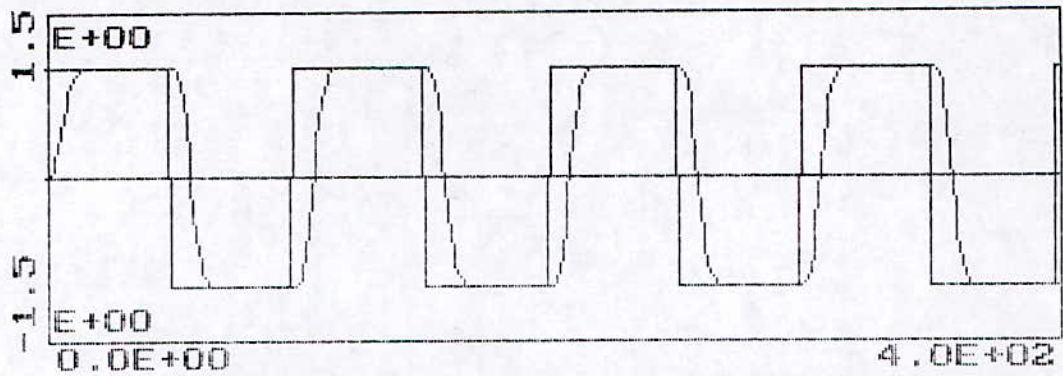


1. évolution de la sortie du système.

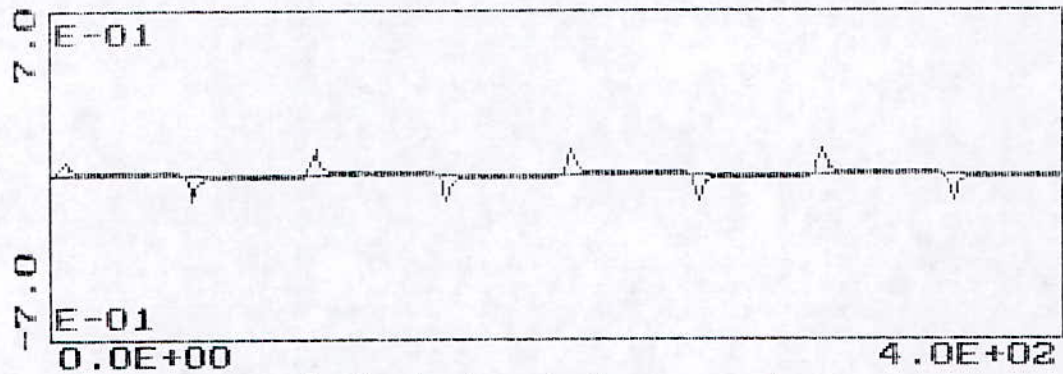


2. évolution de la commande.

Figure 2.8 Présentation sortie/ commande d'un système stable pour $P(q) = 1 + 0.5q^{-1} - 0.5q^{-2}$, $N1=1$ $N2=2$ $NU=1$ $\lambda=0$.

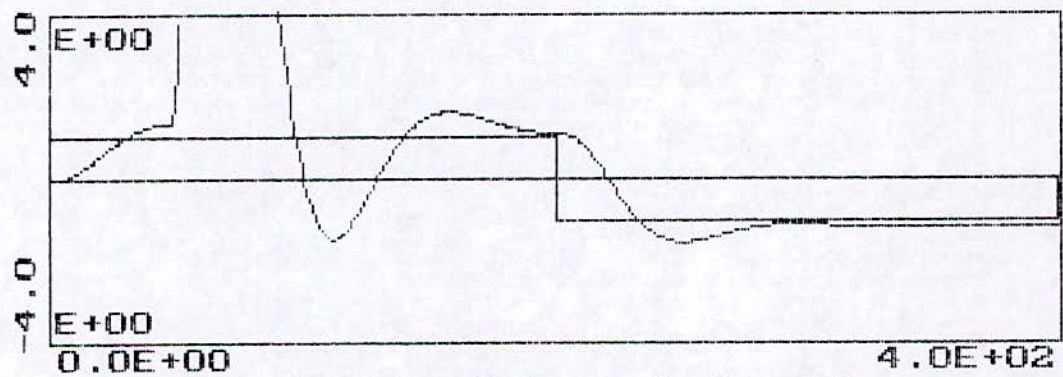


1. évolution de la sortie du système.

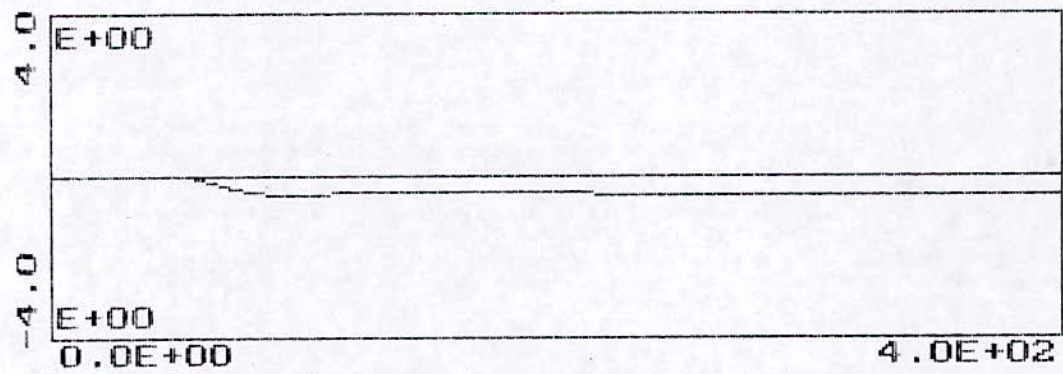


2. évolution de la commande.

Figure 2.9 Présentation sortie/ commande d'un système stable pour $P(q) = 1 + 0.5q^{-1} - 0.5q^{-2}$, $N1=1$ $N2=6$ $NU=1$ $\lambda=0$.



1. évolution de la sortie du système.



2. évolution de la commande.

Figure 2.10 Présentation sortie/ commande d'un système stable pour $P(q) = 1 + 0.05q^{-1} - 0.2q^{-2}$, $N1=1$ $N2=1$ $NU=1$ $\lambda=600$.

2.8.1 Modèle de perturbation.

Comme c'est illustré dans le chapitre 1, un processus ARIMA s'obtient par le passage d'un bruit blanc à travers un modèle (filtre) présentant des pôles et des zéros, tel que c'est illustré dans la figure (GPCE.1).

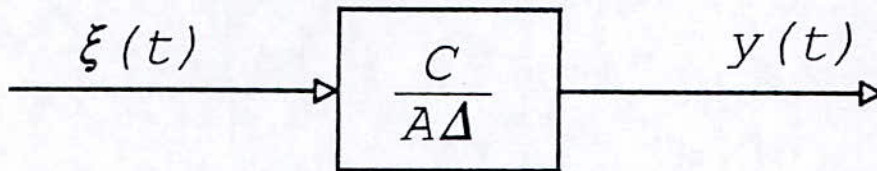


Figure (GPCE.1) Génération d'un processus ARIMA.

Le modèle CARIMA (procédé + perturbation) est utilisé pour représenter ensemble l'effet de la commande et des perturbations sur la sortie du procédé. La génération de ce processus est illustrée sur la figure (GPCE.2).

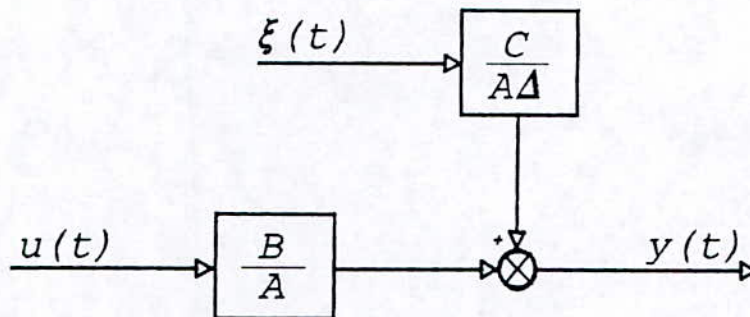


Figure (GPCE.2) Génération d'un processus aléatoire CARIMA.

Le développement de la loi de commande GPC a été élaborée pour un processus déterministe et stochastique. En effet, les cas $c(q^{-1})=1$ et $c(q^{-1}) \neq 1$ ont été développés avec et sans sortie auxiliaires $P(q^{-1})y(t)=\Psi(t)$. Avec un choix spécifique des paramètres de synthèse de la GPC et tenant compte de la similitude avec la méthode déterministe, le calcul du régulateur sera donc identique en choisissant $P(q^{-1})=c(q^{-1})$ (pôles désirés en BF). Il est à noter que le polynôme $C(q^{-1})$ n'apparaît pas dans la fonction de transfert en BF car le développement de l'algorithme a été fait sous la base d'un filtrage de la commande et de la sortie (équation (2.18)).

Remarque

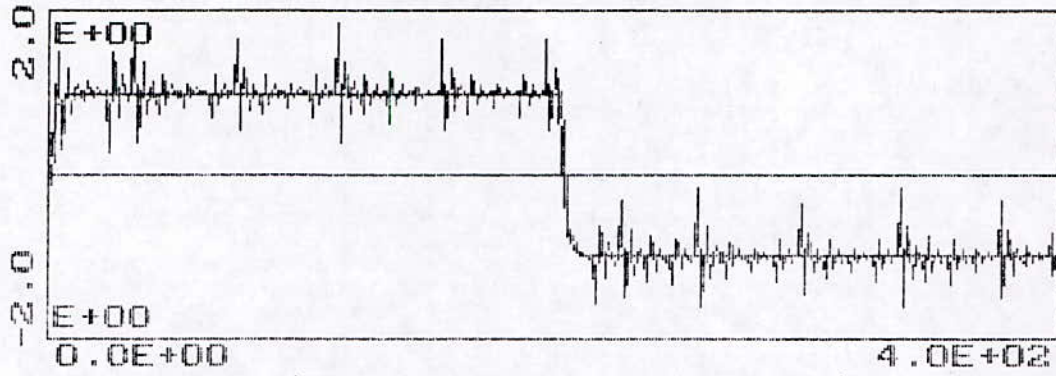
Notons que $C(q^{-1})$ doit avoir des zéros stables qui spécifieront les pôles en BF.

Pour illustrer l'effet de ce filtrage nous avons introduit un polynôme $C'(q^{-1})$ différent de celui du modèle de perturbation du système, dans le calcul des paramètres du régulateur, c'est à dire: $C'=E_j A \Delta + q^{-j} F_j$ au lieu de $C=E_j A \Delta + q^{-j} F_j$

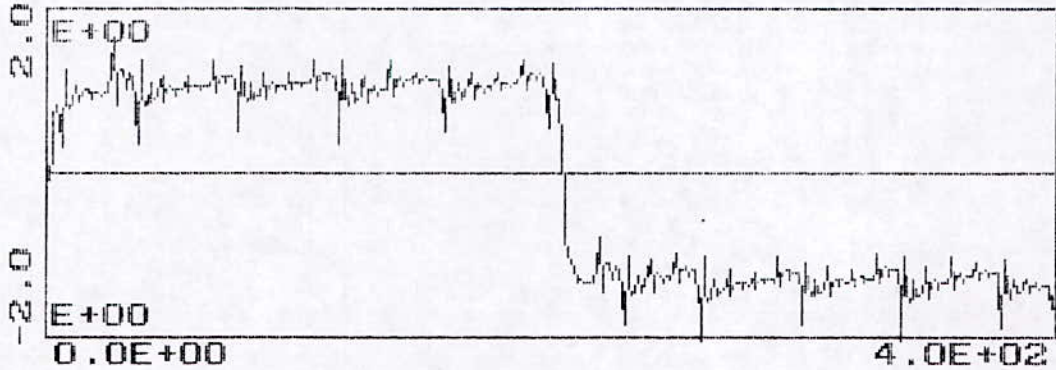
Pour la simulation

$$\begin{cases} C = 1 - 0.5q^{-1} - 0.05q^{-2} \\ C' = 1 + 0.2q^{-1} + 0.5q^{-2} \end{cases}$$

Les figures (2.11)(2.12) montrent une diminution de l'amplitude de la perturbation aléatoire et ceci en introduisant au système sont propre modèle de perturbation.

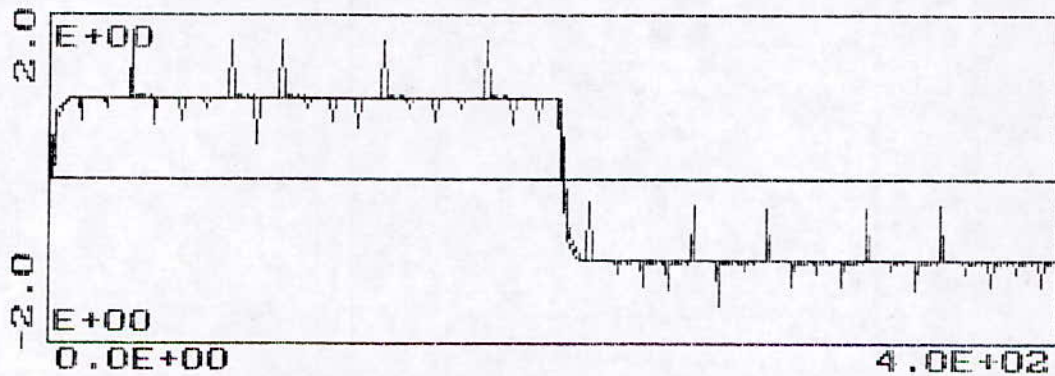


1. évolution de la sortie du système.

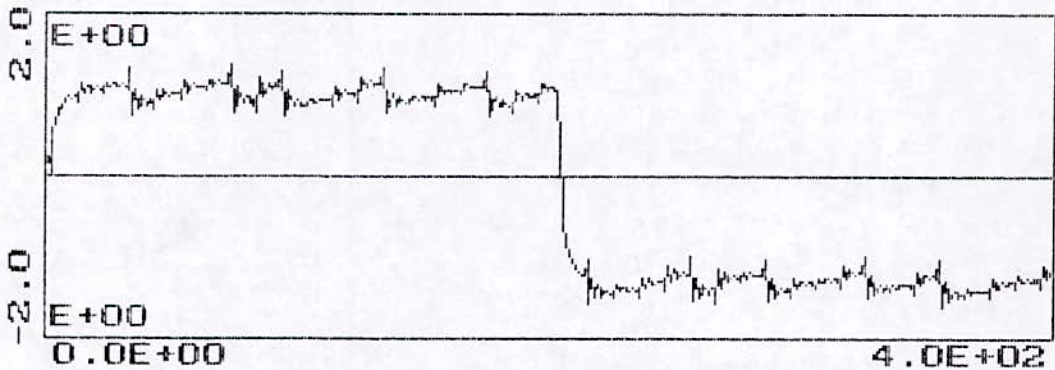


2. évolution de la commande.

Figure 2.11 Présentation sortie/ commande d'un système stable à une perturbation aléatoire pour $C(q^{-1}) \neq C'(q^{-1})$, $N1=1$ $N2=4$ $NU=1$ $\lambda=0$.



1. évolution de la sortie du système.



2. évolution de la commande.

Figure 2.12 Présentation sortie/ commande d'un système stable à une perturbation aléatoire pour $C(q^{-1}) = C'(q^{-1})$, $N1=1$ $N2=4$ $NU=1$ $\lambda=0$.

2.8.2 Influence du retard et de la surparamétrisation sur la GPC.

Cette partie consiste à tester la GPC sur quelques problèmes qu'on peut rencontrer lors de la commande des processus. Des problèmes relatifs à la variation au cours du temps du retard et de la surparamétrisation. La simulation a été réalisée pour $P(q^{-1})=C(q^{-1})=1$.

2.8.2.1 Influence du retard.

L'influence du retard sur la GPC a été testée sur un système à phase non-minimale et instable en boucle ouverte. Ce système présente un pôle et un zéro de 1.1 et 2 respectivement, qui s'écrit sous la forme suivante [2].

$$\frac{B(q^{-1})}{A(q^{-1})} = \frac{-0.1q^{-1} - 0.2q^{-2}}{1 - 1.1q^{-1}}$$

Le temps maximal de la simulation est de 400 dont sont réparties 4 périodes de créneau de référence d'amplitude 1 et -1 alternativement. Pour illustrer l'influence du retard sur les performances de la commande, on a introduit au cour de simulation, à $t=150$ un retard de 2, figure (2.13), puis le retard reprend son état initial à $t=300$. De bons résultats ont été obtenus par un choix des paramètres de synthèse de la GPC de $N1=NU=1, N2=10$. Ces résultats sont consignés sur les figures (2.14)(2.15). Il est à signaler que les oscillations obtenues entre $t=150$ et $t=300$ peuvent être éliminer en choisissant d'autre paramètres, en particulier $N2$. L'augmentation de ce retard au delà de deux risque de détériorer les performances, et de déstabiliser la sortie du système en BF.

2.8.2.2 Influence de la surparamétrisation.

L'idée principale de la surparamétrisation est de commander un système dont le nombre de paramètres est supposée supérieur aux paramètres réels.

Soit un système donner par la fonction de transfert suivante:[2] :

$$\frac{B(q^{-1})}{A(q^{-1})} = \frac{1}{1 - 0.9q^{-1}}$$

Pour la simulation du régulateur par GPC, la loi de commande est élaborée à partir des paramètres relatifs au système surparamétrisé équivalent à :

$$\frac{B(q^{-1})}{A(q^{-1})} = \frac{1 + 0.5q^{-1}}{1 - 0.4q^{-1} - 0.5q^{-2}}$$

Cette surparamétrisation est obtenue en introduisant un facteur commun au numérateur et dénominateur du système réel.

Dans le but d'une comparaison, les performances obtenues par la commande du système réel et du système surparamétrisé sont identique et consigné sur la figure (2.16). D'autres dynamiques de sortie peuvent être tirées en agissant sur les paramètres de synthèse, en particulier l'horizon maximal de prédiction.

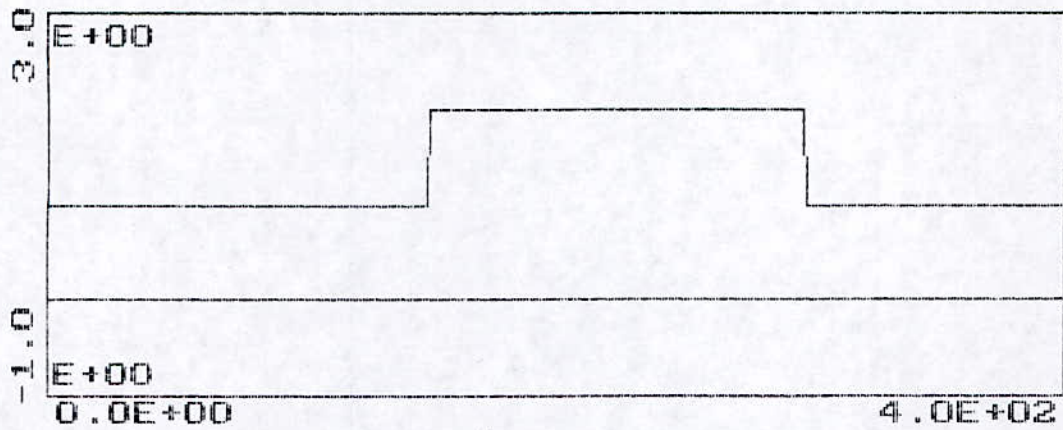


figure 2.13 évolution du retard

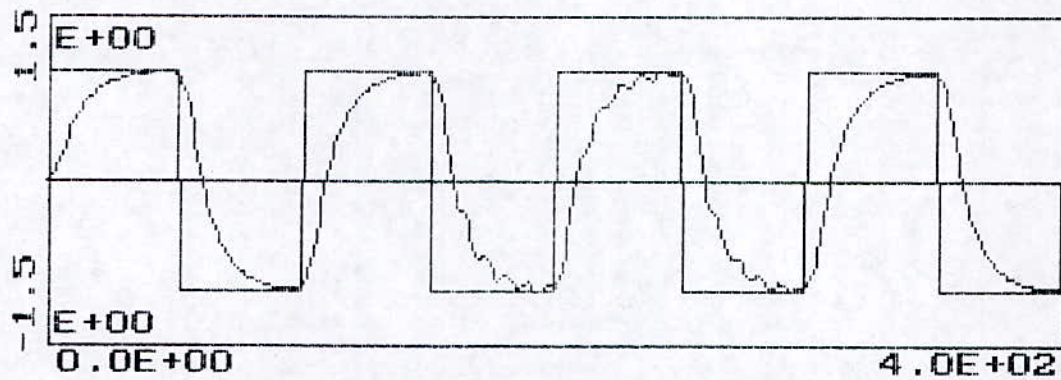


figure 2.14 évolution de la sortie pour une variation du retard.

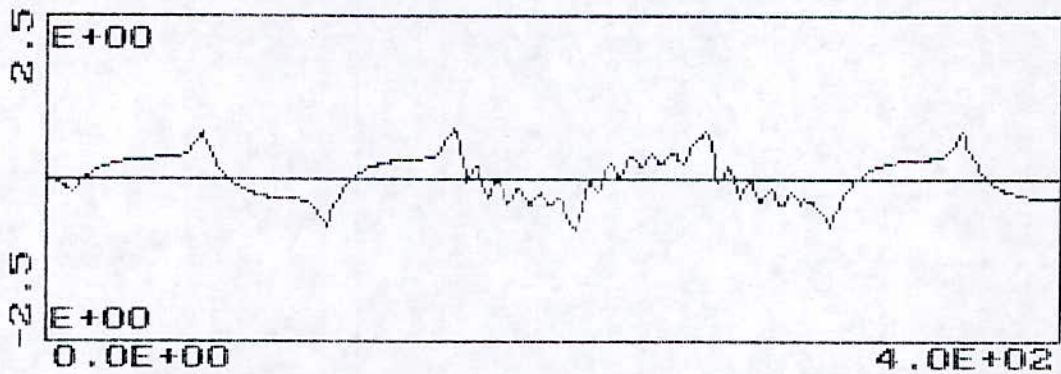
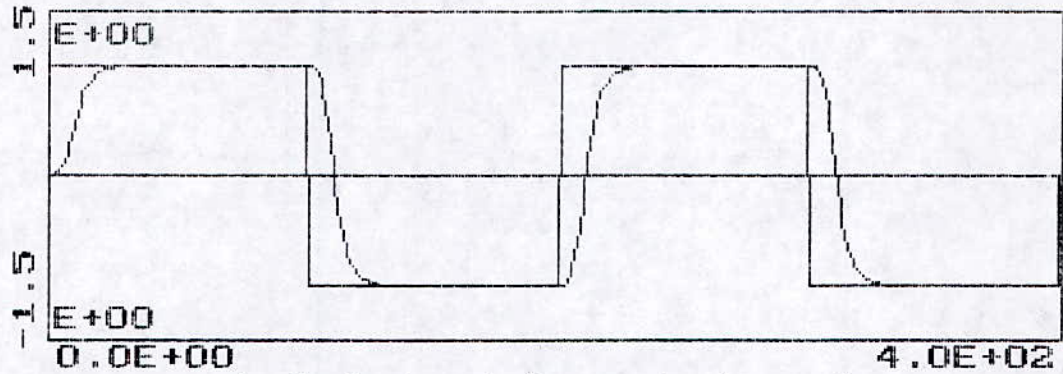
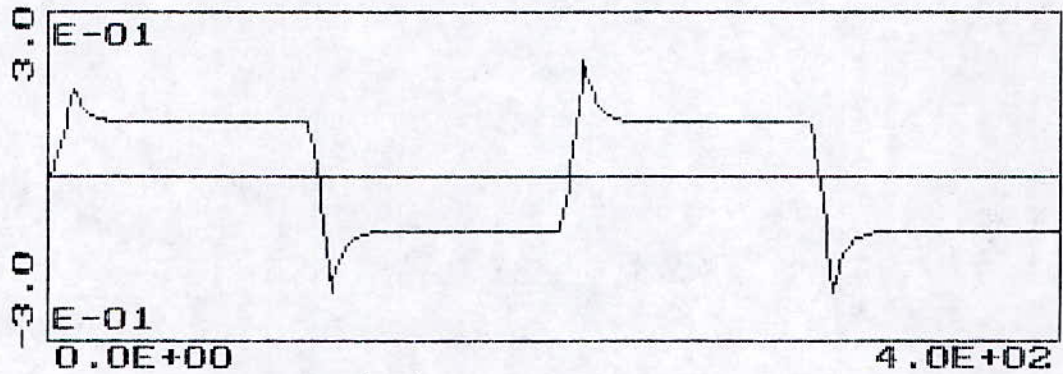


Figure 2.15 évolution de la commande pour une variation du retard.



1. évolution de la sortie du système.



2. évolution de la commande.

Figure 2.16 Présentation sortie/ commande d'un système Surparamétrisé de degrés ($na_s=2$ $nb_s=1$) pour $N1=1$ $N2=10$ $NU=1$ $\lambda=0$.

Conclusion.

L'extension de l'algorithme nous a permis d'augmenter le nombre de paramètres de synthèse de la commande prédictive généralisée. Cette augmentation donne à l'algorithme plus de degré de liberté ou de flexibilité concernant leur choix, facilitant ainsi à l'utilisateur d'obtenir des réponses appréciables et des commandes douces et réalisables. Cette extension offre aussi la possibilité de maîtriser la dynamique du système en boucle fermée qui sera fixée selon les 'désires' de l'utilisateur, mais tout reste dans le cadre des limites du système.

La stabilité et les performances du système sont gouvernées par les racines de l'équation du polynôme caractéristique. Cette équation dépend énormément du polynôme de pondération $P(q^{-1})$ et du polynôme d'observation $C(q^{-1})$ qui doivent être choisis de tel sorte que ses zéros soient à l'intérieur du cercle unité. Dans le cas déterministe le choix de $C(q^{-1})$ est arbitraire, en fonction des bruits de mesure des différentes variables nécessaires pour le calcul de la commande. Par contre dans le cas stochastique, ce polynôme est imposé par le modèle de représentation du système, défini par le modèle CARIMA. L'obtention de meilleurs performances dépend de la fidélité de la phase de modélisation, surtout concernant le modèle de perturbation.

L'étude par simulation de la GPC a montré l'efficacité de l'algorithme à faire face au problème de variation du retard et de la surparamétrisation. Les résultats obtenus montrent qu'à partir du choix des paramètres de la GPC à savoir les horizons de prédictions, de commande et des pondérations, on peut obtenir les performances qu'on désire et se ramener à quelques algorithmes comme celui du placement de pôles. La spécification de la dynamique en boucle fermée, peut être faite de différentes manières. En faisant un choix du polynôme $P(q^{-1})$, en fonction du comportement du système, on peut alors contrôler le comportement en boucle fermée. Les paramètres de synthèse de la GPC permettent aussi de spécifier cette dynamique, en faisant varier $N1, N2, NU$ et λ , sans changer $P(q^{-1})$.

Des variations structurelles du système à commander, tel que le retard, affectent grandement la commande et détériore le comportement en boucle fermée. Une variation du retard au delà de deux aboutit à une divergence de la sortie. Une variation paramétrique dans le système, déstabilise ce dernier et nécessite un nouveau calcul du régulateur en fonction de ces variations.

L'introduction d'un outil intermédiaire entre le système et le calcul de la commande est nécessaire. Cet outil se traduit par une loi d'adaptation qui permet un calcul conséquent des paramètres du régulateur en fonction des variations structurelles du système à commander. C'est l'objectif du prochain chapitre.

Chapitre 3

COMMANDE PREDICTIVE GENERALISEE ADAPTATIVE

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

BOILEAU

Chapitre 3

COMMANDE PREDICTIVE GENERALISEE ADAPTATIVE (GPCA)

Introduction.

Depuis une trentaine d'années, on note une activité toujours croissante dans le domaine de l'étude théorique de la commande adaptative (CA) et son application. Le mot en lui même peut être défini de la façon suivante:

La "commande adaptative" est un ensemble de techniques utilisées pour l'ajustement automatique en ligne des régulateurs des boucles de commande afin de réaliser ou maintenir un certain niveau de performances quand les paramètres du procédé à commander sont inconnus et/ou varient dans le temps [31].

Le sens généralement retenu par la commande adaptative consiste à dire qu'un régulateur est adaptatif, s'il peut modifier son comportement en réponse à des changements dans la dynamique du procédé à contrôler [32]. Cette modification doit naturellement se faire d'une manière automatique, sans intervention extérieur. Quant au comportement du régulateur, il est défini comme son aptitude à maintenir un certain niveau de performance, préalablement choisi par l'utilisateur.

Les tâches typiques pouvant être effectuées par un système de commande adaptative sont indiquées ci-dessous [31]:

- 1- Ajustement automatique des paramètres du régulateur à la mise en oeuvre, dont l'effet est la réduction du temps d'ajustement et l'amélioration des performances.
- 2- Détermination automatique des paramètres optimaux des régulateurs dans les divers point de fonctionnement.
- 3- Maintenir les performances du système de commande quand les caractéristiques du procédé changent.
- 4- Possibilité de mise en oeuvre des régulateurs plus complexes et plus performants que les régulateurs classiques et ceci comme conséquence de l'ajustement automatique.
- 5- Détection des variations anormales des caractéristiques des procédés. Ces variations se reflètent dans les valeurs des paramètres fournies par l'algorithme d'adaptation.

A partir des points énoncés précédemment, plusieurs techniques ont été proposées, pour garantir une capacité d'adaptation dans le système de commande. A cet effet il est nécessaire de les différencier, car l'appellation "adaptatif" est général et comprend plusieurs concepts distincts.

Dans ce chapitre nous présenterons un bref aperçu des différentes techniques

de commande adaptative. Nous passerons ensuite au développement de l'algorithme d'adaptation paramétrique (AAP). Deux techniques de commande prédictive généralisée adaptative seront présentées à savoir, la commande prédictive généralisée directe (GPCAD) et la commande prédictive généralisée indirecte (GPCAI). Pour clôturer ce chapitre nous présenterons des résultats de simulation.

3.1 Différentes techniques de commande adaptative. [31][33]

Il existe plusieurs techniques de commande adaptative:

- Commande adaptative à gain programmé (Gain Scheduling).
- Commande adaptative à modèle de référence (MRAC ou MRAS: model reference adaptive control ou system).
- Système à commande auto-ajustable (self-tuning control STC).
- Approximation des stratégies de commande optimal stochastique (commande duale).

Le point commun à toutes ces approches est la boucle classique de contre réaction schématisée sur la figure (GPCA.1).

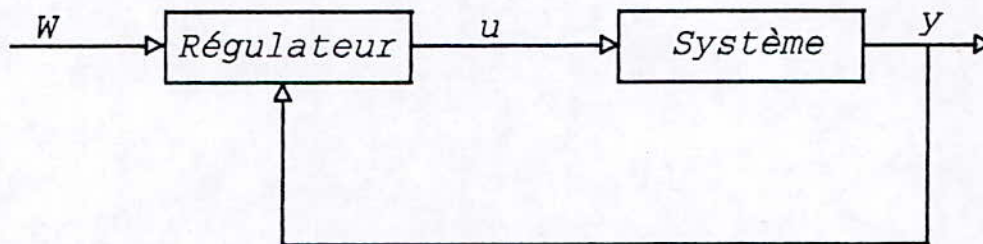


Figure (GPCA.1) Boucle classique de contre réaction.

Mais le problème principal est la recherche d'un algorithme adéquat qui change les paramètres du régulateur en fonction de l'évolution de ceux du processus et des perturbations aléatoires; c'est là où réside la différence dans les schémas choisis.

3.1.1 Commande adaptative à gain programmé.

Lorsque les paramètres du système sont variables ou inconnus, une approche naturelle consiste à trouver des variables auxiliaires mesurables dont la variation est une image acceptable de l'évolution des paramètres du système. On construit alors des lois de variation des paramètres du régulateur en fonction des variables auxiliaires mesurées. La structure de réglage de ce type de commande est présentée sur la figure (GPCA.2).

3.1.2 Commande à modèle de référence (MRAC).

Dans cette approche, les performances désirées sont réalisées sous la forme d'un modèle à suivre. L'écart entre la sortie du système et celle du modèle est utilisé par un algorithme d'adaptation pour faire varier les paramètres du régulateur, dans le but de réaliser la meilleure poursuite possible (minimiser l'écart). La figure (GPCA.3) illustre la structure de réglage de ce type de commande.

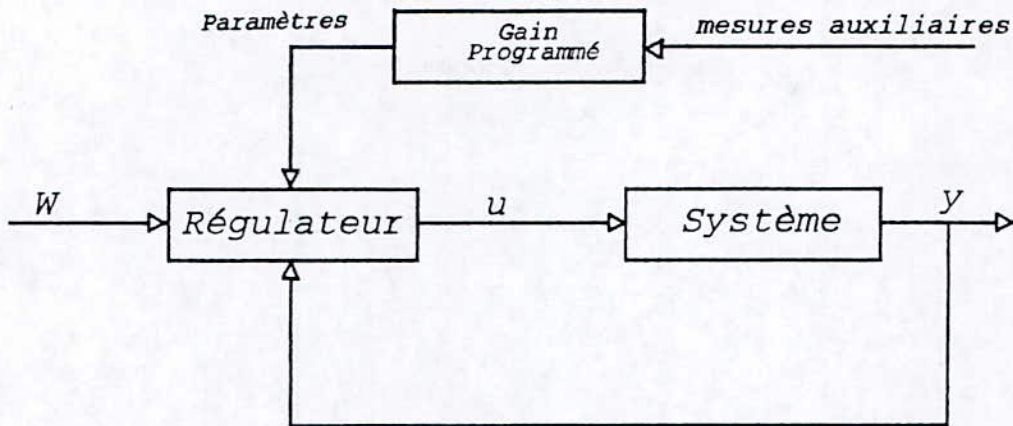


Figure (GPCA.2) Boucle de réglage adaptatif à gain programmé.

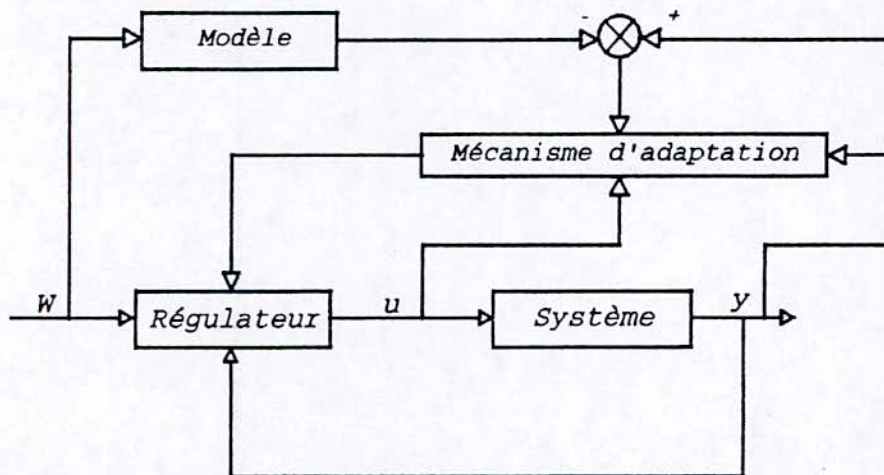


Figure 3.3 Boucle de réglage adaptatif à modèle de référence.

3.1.3 Réglage auto-ajustable.

Le principe de cette approche est l'ajustement des paramètres du régulateur à l'aide d'une boucle externe. Celle-ci est composée de deux blocs, figure (GPCA.4).

- bloc d'estimation récurrente.
- bloc de calcul de la commande à partir des paramètres estimés.

Le bloc d'estimation est utilisé soit pour estimer les paramètres du système lui-même, c'est "l'approche indirecte" de la commande adaptative, soit pour estimer les paramètres du régulateur, on parle alors "d'approche directe". Plusieurs méthodes ont été utilisées pour réaliser cette estimation telles que les moindres carrés étendus ou généralisés, la variable instrumentale, le filtre de Kalman et le maximum de vraisemblance. Le bloc de calcul de la commande dépend de la méthode choisie: Variance minimale généralisée (GMV), placement de pôles (PP), poursuite et régulation (PR) ou la commande prédictive généralisée (GPC).

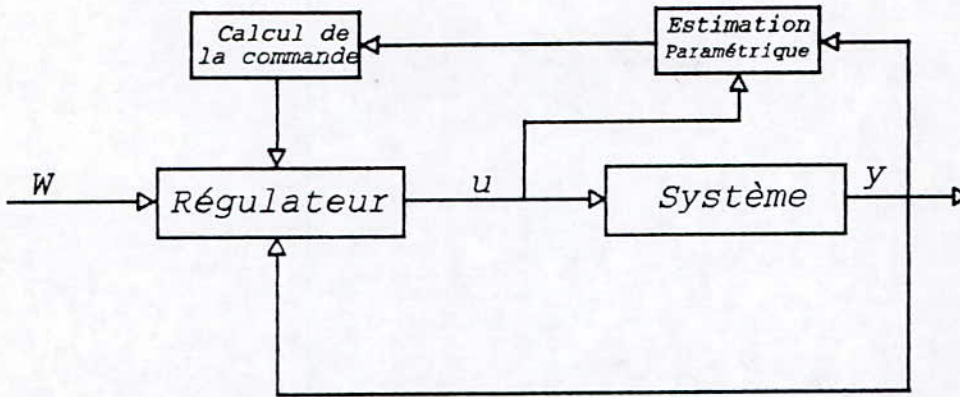


Figure (GPCA.4) structure de réglage auto-ajustable.

3.1.4 Commande duale.

Dans le cas de la commande duale, les paramètres inconnus sont traités comme des états additionnels du système. Ce qui transforme un très simple problème de commande linéaire en un problème de commande nonlinéaire stochastique. Le régulateur est composé d'un estimateur nonlinéaire suivi par le régulateur proprement dit dont la structure de réglage est présentée sur la figure (GPCA.5).

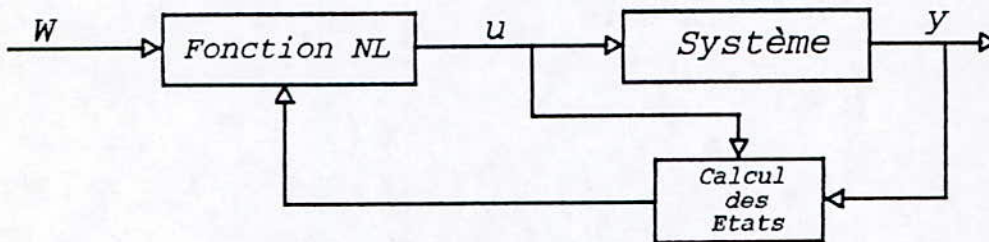


Figure (GPCA.5) Schéma bloc de la commande duale.

Après avoir présenté les différentes approches de la commande adaptative, nous nous sommes intéressés dans notre travail aux régulateurs auto-ajustables. Cette étude nécessite l'utilisation d'un modèle de représentation qui peut être obtenu par identification dont on présentera la principe et les différentes méthodes utilisées.

3.2 Identification.

Nous avons présenté dans les chapitres précédents une étude des performances de la GPC à paramètres connus basée sur un modèle dynamique linéaire CARIMA. Lorsque ce modèle est issu des lois physiques, on parlera de modèle de connaissance. Dans le cas où il est construit à partir d'une suite de mesures des entrées et des sorties, le modèle obtenu est appelé modèle de représentation. La détermination explicite des paramètres de ce modèle peut être faite à l'aide de l'identification [34].

L'identification consiste à déterminer un ensemble d'équations (ou modèle) décrivant le mieux possible le procédé, dont la connaissance est nécessaire pour la conception est la mise en oeuvre d'un système performant de commande. L'identification comporte quatre étape [35]:

étape 1: Acquisition des entrées et des sorties.

étape 2: Choix de la structure du modèle.

étape 3: Estimation des paramètres du modèle.

étape 4: Validation du modèle identifié.

Les valeurs numériques des paramètres du modèle obtenu à partir de ces quatre étapes sont déterminés de telle sorte que le comportement du modèle soit le plus proche de celui du système réel. Ceci est réalisé qu'après avoir choisi et minimisé un critère. Ce critère est formé à partir de la différence entre la sortie du système et du modèle. Le schéma du principe de l'identification est présenté sur la figure (GPCA.6).

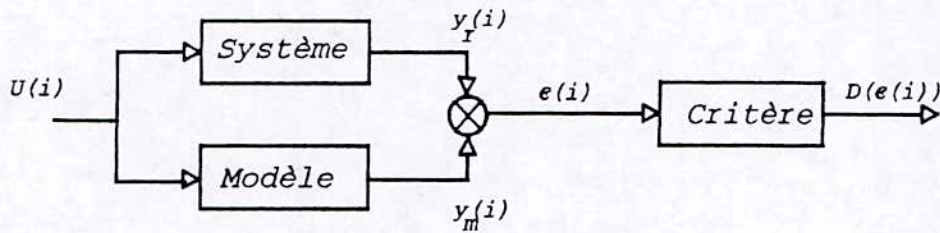


Figure (GPCA.6) Schéma du principe de l'identification.

avec:

$$J = D[\mathbf{e}(t)] = \sum_{i=0}^t [y_R(i) - y_M(i)]^2 = \sum_{i=0}^t [\mathbf{e}(i)]^2 \quad (3.1)$$

où J est le critère et D : Distance de sortie.

3.3 Type de modèle utilisé.

La classe de modèle utilisé dans différentes stratégies de commande adaptative est représentée par l'équation suivante [21]:

$$A(q^{-1})y(t) = B(q^{-1})u(t) + \omega(t) \quad (3.2)$$

En effet, ce modèle linéaire est à la base de la synthèse de la plupart des lois de commande linéaire. Sa nature discrète le rend facilement utilisable numériquement et il est capable de représenter le comportement de la majorité des procédés rencontrés en pratique [1].

Le principe de l'estimation des paramètres, dans le cas discret, est représenté par la figure (GPCA.7) [35].

3.4 Identification récurrente.

Une approche naturelle de l'identification est de recueillir toutes les mesures nécessaires et ensuite de déterminer par leur étude statistique les paramètres du modèle, l'identification est alors dite en temps différé (off-line). Cette approche est souvent mise en défaut lorsque les paramètres du système sont variables dans le temps de plus que le nombre de valeurs numériques traitées en même temps est important. Pour ce type de système [1], il faut à chaque instant réactualiser les

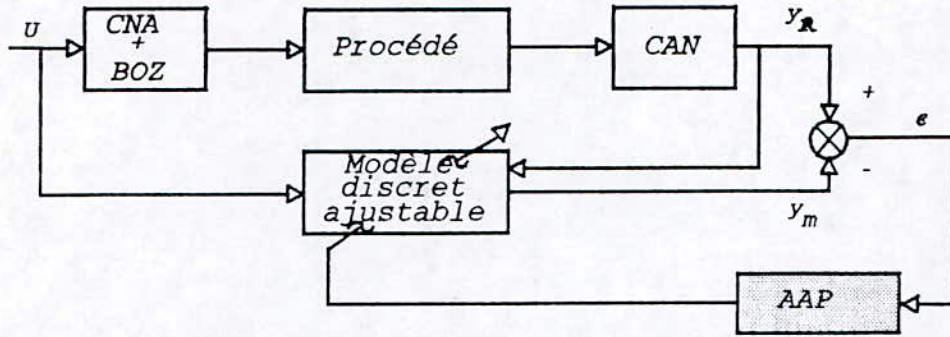


Figure (GPCA.7) Principe de l'identification dans le cas discret.

paramètres identifiés. Il faut donc faire une identification en temps réel (on-line). Un algorithme comme celui des moindres carrés récursif (MCR et ces extensions permet de résoudre ce problème.

3.4.1 Formulation du problème.

Nous présentons la synthèse des algorithmes d'adaptation paramétrique à partir de la minimisation d'un critère de type "moindres carrés" à un pas en terme de l'erreur procédé-modèle [35]. Considérons le modèle discrétisé, déterministe d'un procédé caractérisé par la fonction de transfert échantillonnée suivante:

$$H(q^{-1}) = q^{-d} \frac{B(q^{-1})}{A(q^{-1})}; \quad (d \geq 1 \text{ équivalent à } k) \quad (3.3)$$

La relation entrée-sortie s'exprime:

$$A(q^{-1})y_R(t) = q^{-d} B(q^{-1})u(t) \quad (3.4)$$

où *A* et *B* sont identiques à ceux définis dans les chapitres précédents. Sous une forme plus explicite, la sortie du procédé s'écrit:

$$y_R(t) = -\sum_{i=1}^n a_i y_R(t-i) + \sum_{i=0}^m b_i u(t-d-i) \quad (3.5)$$

ou sous une forme condensée:

$$y_R(t) = \theta^T \phi(t) \quad (3.6)$$

avec

$$\begin{cases} \theta^T(t) = [a_1 \dots a_n, b_0 \dots b_m] \\ \phi^T(t) = [-y_R(t-1) \dots -y_R(t-n), u(t-d) \dots u(t-d-m)] \end{cases} \quad (3.7)$$

sont respectivement les vecteurs des paramètres du procédé et des mesures. Dans le cas où la sortie $y_R(t)$ ne dépend que de l'entrée u , le système peut alors être modélisé par un modèle ARMA, figure (GPCA.8).

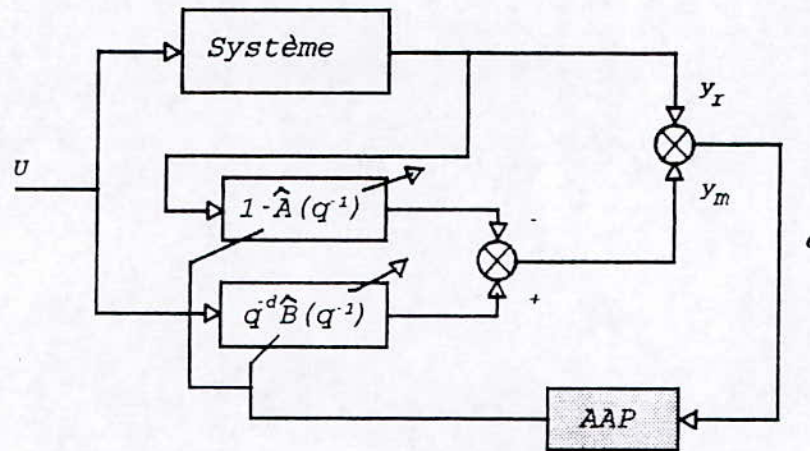


Figure (GPCA.8) Identification du modèle ARMA.

Avec \hat{d} est le retard estimé, $\hat{d} > 1$.

et
$$\begin{cases} \hat{B} = \hat{b}_0 + \hat{b}_1 q^{-1} + \dots + \hat{b}_m q^{-m} \\ 1 - \hat{A} = \hat{a}_1 q^{-1} + \dots + \hat{a}_n q^{-n} \end{cases}$$

Les valeurs \hat{d} , \hat{n} et \hat{m} peuvent être estimés au départ selon la connaissance que l'on a sur le système.

3.4.2 Expression du critère.

Exprimons le critère de l'équation (3.1) en fonction des paramètres identifiés, soit:

$$y_M(t) = - \sum_{i=1}^n \hat{a}_i y_R(t-i) + \sum_{i=0}^m \hat{b}_i u(t-\hat{d}-i) \tag{3.8}$$

En introduisant les vecteurs des paramètres et des observations θ et ϕ respectivement dans l'équation (3.6), on obtient:

$$y_M(t) = \hat{\theta}^T \phi(t) \tag{3.9}$$

avec:

$$\begin{cases} \hat{\theta}^T(t) = [\hat{a}_1 \dots \hat{a}_n, \hat{b}_0 \dots \hat{b}_m] \\ \phi^T(t) = [-y_R(t-1) \dots -y_R(t-n), u(t-\hat{d}) \dots u(t-\hat{d}-m)] \end{cases} \tag{3.10}$$

D'où:

$$J = \sum_{i=0}^t [y_R(i) - \hat{\theta}^T(i) \phi(i)]^2 \tag{3.11}$$

3.4.3 Algorithme des moindres carrés récursives.

La minimisation de $J(t)$ [35][36], donne l'algorithme récursif suivant [35][37]:

$$\begin{cases} \hat{\theta}(t+1) = \hat{\theta}(t) + F(t)\hat{\theta}(t)\epsilon(t+1) \\ F(t+1) = F(t) - \frac{F(t)\phi(t)\phi^T(t)F(t)}{1 + \phi^T(t)F(t)\phi(t)} \\ \epsilon(t+1) = \frac{y_R(t+1) - \hat{\theta}(t)\phi(t)}{1 + \phi^T(t)F(t)\phi(t)} \end{cases} \quad (3.12)$$

La matrice F , appelée matrice du gain d'adaptation, est construite à partir du vecteur d'observation.

$$F(t) = \sum_{i=1}^t \phi(i-1)\phi^T(i-1) \quad (3.13)$$

L'équation donnant $F(t+1)$ s'obtient en appliquant le lemme d'inversion matricielle [34][35] à l'expression suivante:

$$F^{-1}(t+1) = F^{-1}(t) + \phi(t)\phi^T(t) \quad (3.14)$$

3.4.4 Initialisation des paramètres.

La convergence étant assurée indépendamment des conditions initiales, on peut choisir, $\hat{\theta}^T(0) = [0 \dots 0]$, ou autres valeurs selon la commande utilisée. Les paramètres initiaux peuvent être éloignés des valeurs réelles, on choisit alors une matrice de gain d'adaptation F telle que $F(0) = g.I$ où g est un réel très grand ($g > 100$)

3.4.5 Cas stochastique.

En présence des perturbations aléatoires, le cas précédent converge vers des paramètres biaisés [35]. Pour tenir compte de l'influence des perturbations un changement de modèle d'identification est nécessaire. On obtient le schéma de la figure (GPCA.9).

Avec les modifications concernant les vecteurs des paramètres et d'observation (eqt 3.9)

$$\begin{cases} \hat{\theta}^T(t) = [\hat{a}_1 \dots \hat{a}_n, \hat{b}_0 \dots \hat{b}_m, \hat{c}_1 \dots \hat{c}_{nc}] \\ \phi^T(t) = [-y_R(t-1) \dots -y_R(t-n), u(t-d) \dots u(t-d-m), \\ \epsilon(t) \dots \epsilon(t-nc)] \end{cases} \quad (3.15)$$

L'algorithme donnée par l'équation (3.12) est appelé algorithme des moindres carrés étendu [34] et est utilisé pour l'identification des paramètres du modèle CARIMA. Dans le vecteur de mesure ϕ nous avons pris les écarts ϵ comme estimations du bruit blanc $\xi(t)$, en se basant sur le principe suivant:

L'estimation $\hat{\theta}(t)$ étant optimale, L'écart entre y_R et y_M ne peut être du qu'au bruit.

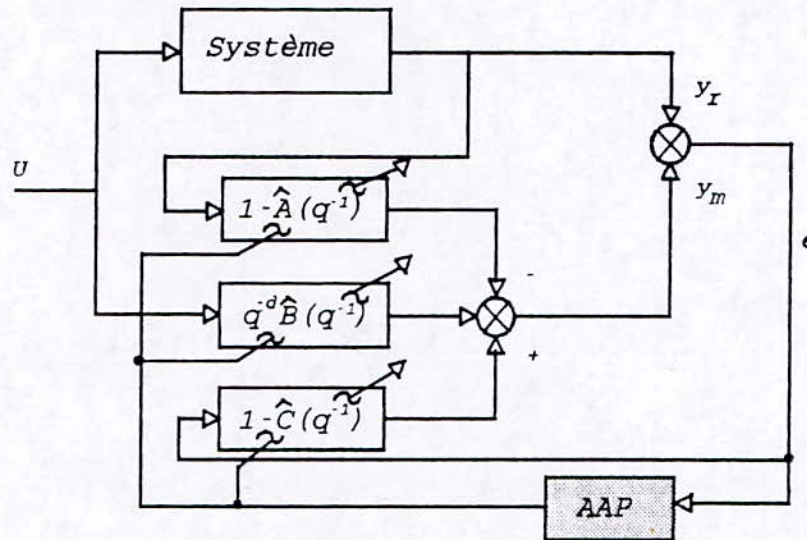


Figure (GPCA.9) Identification dans le cas stochastique

3.4.6 Variantes de l'algorithme des MCR.

Le gain d'adaptation $F(t)$ tend vers zéros lorsque l'horizon d'identification augmente, ce qui assure la convergence de l'algorithme lorsque les paramètres du système sont constants. Lorsque ceux-ci sont variables, l'identification devient biaisée. Il faut donc une mise à jour du gain d'adaptation en fonction de cette variation. Ceci peut être réalisé en écrivant l'équation (3.14) sous la forme (proposée par Landau) [35].

$$F^{-1}(t+1) = \lambda_1(t) F^{-1}(t) + \lambda_2(t) \phi(t) \phi^T(t) \tag{3.16}$$

où: λ_1 et λ_2 sont des coefficients de pondération variables tels que $0 < \lambda_1 < 1$ et $0 < \lambda_2 < 2$. Ces coefficients introduisent, comme le montre l'équation (3.16) un profil de réactualisation de F , qui peut être exploité pour l'identification des paramètres variables.

En prenant $\lambda_1 = \lambda_2 = 1$, on retrouve l'algorithme des moindres carrés précédent qu'on appelle aussi algorithme à gain décroissant. La deuxième expression de l'équation (3.12) devient:

$$F(t+1) = \frac{1}{\lambda_1(t)} \left[F(t) - \frac{F(t) \phi(t) \phi^T(t) F(t)}{\frac{\lambda_1(t)}{\lambda_2(t)} + \phi^T(t) F(t) \phi(t)} \right] \tag{3.17}$$

En fonction des valeurs données à λ_1 et λ_2 , on différencie plusieurs classes d'algorithmes.

a. Algorithmes à facteur d'oubli [35].

Il se caractérise par $\lambda_2 = 1$ et $0 < \lambda_1(t) < 1$. Le critère à minimiser est alors donné par:

$$J = \sum_{i=0}^t \lambda_1(i) e^2(i) \quad (3.18)$$

on voit que les données anciennes ont une pondération plus faible que les récentes qui sont ainsi privilégiées. C'est pour cette raison que $\lambda_1(t)$ est appelé facteur d'oubli.

Selon le choix de $\lambda_1(t)$, on distingue:

* L'algorithme à facteur d'oubli fixe, ou on pose: [35]

$\lambda_1(t) = \lambda_1 = \text{const}$ ($0.95 \leq \lambda_1 \leq 0.99$). Cette algorithme convient pour l'identification des systèmes lentement variables dans le temps.

* L'algorithme à facteur d'oubli variable, où on pose: [35]

$\lambda_1(0) = \lambda_0$, $\lambda_1(t) = \lambda_0 \lambda_1(t-1) + 1 - \lambda_0$ ($0.95 \leq \lambda_0 \leq 0.99$). Cet algorithme se comporte au départ comme le précédent, puis lorsque t augmente, son comportement devient celui à gain décroissant.

b. Algorithme à trace constante [35].

Cet algorithme à connu un grand succès dans le domaine de la commande adaptative. En effet, le maintien de la trace constante de $F(t)$ permet le regonflement permanent du gain d'adaptation et rend ainsi l'algorithme apte à suivre la variation des paramètres du système. Les valeurs de $\lambda_1(t)$ et $\lambda_2(t)$ sont calculées à chaque instant de manière à ce que l'on ait:

$$\begin{cases} \text{tr}[F(t+1)] = \text{tr}[F(t)] = \text{tr}[F(0)] \\ \frac{\lambda_1(t)}{\lambda_2(t)} = c \end{cases}$$

où c et $\text{tr}[F(t)]$ étant fixés à l'avance avec $c=0.97$.

3.5 Principe d'équivalence certaine.

Nous avons vu que pour la commande à modèle de référence et pour la commande auto-ajustable, les paramètres du régulateur ou du procédé sont estimés en ligne. Le principe d'équivalence certaine, qui revient à négliger les incertitudes sur les estimés [38], permet d'utiliser les estimés comme s'ils étaient les vrais paramètres. Il faut tout de même noter que ces incertitudes, si elles sont connues, peuvent être utilisées pour modifier le régulateur. On peut donc dire que la synthèse de l'AAP se fait indépendamment de la méthode de commande choisie, pourvue que les modèles utilisés par ces deux tâches soient compatibles.

L'AAP est utilisé soit pour l'ajustement des paramètres du régulateur dans les schémas de commande adaptative directe [11], soit pour l'ajustement des paramètres du modèle du procédé dans les schémas de commande adaptative indirecte [12].

Nous présenterons par la suite les concepts de commande adaptative directe et indirecte en liaison avec les stratégies de commande développées dans les chapitres précédents.

3.6 Commande prédictive généralisée adaptative: Approche Indirecte.

Dans cette approche, les paramètres du régulateurs sont obtenus à partir de l'estimation des paramètres du système. La GPCA indirecte consiste à utiliser la loi de commande développée dans les chapitres précédents, et remplacer les paramètres du modèle par leur estimation. La figure (GPCA.10) représente le schéma bloc de la GPCA indirecte [25];

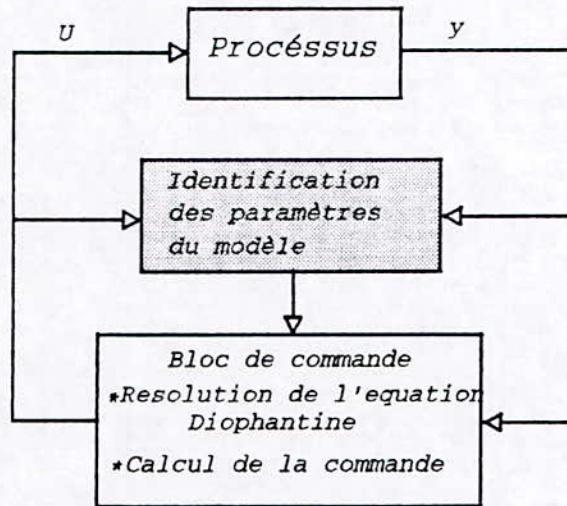


Figure (GPCA.10) Principe de la GPCA.

La commande prédictive généralisée indirecte obéit aux étapes suivantes [41]:

étape 1: Estimation en temps réel des paramètres du modèle du procédé.

étape 2: Calcul des paramètres du régulateur à partir des paramètres estimés du modèle du procédé.

Nous illustrerons cette approche pour le cas de la GPCA dont le schéma du principe est présenté sur la figure (GPCA.11).

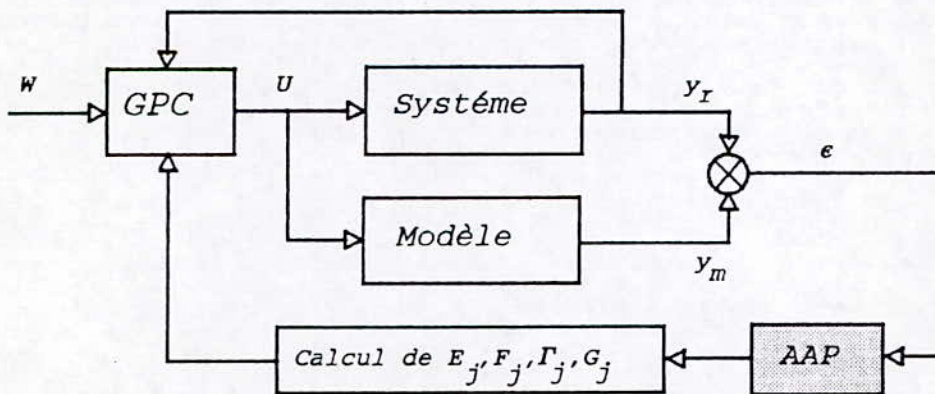


Figure (GPCA.11) Principe de la GPCA (Schéma représentatif).

Pour bien expliciter ce type de principe de commande ,on distingue deux étapes:

étape 1: Estimation des paramètres du modèle.

L'équation du procédé est donné par le modèle CARIMA qui peut s'écrire sous la forme:

$$y_R(t) = \theta^T \phi(t) \quad (3.19)$$

Le vecteur de paramètre θ peut être estimé en utilisant une méthode d'identification de type MCR à trace constante, on construit alors un estimateur ajustable de la forme:

$$y_M(t) = \hat{\theta}^T \phi(t) \quad (3.20)$$

et en utilisant l'AAP donnée par l'équation (3.12) où l'erreur d'adaptation est donnée par:

$$\varepsilon(t) = y(t) - \hat{\theta}^T(t) \phi(t) \quad (3.21)$$

à chaque instant on obtient de $\hat{\theta}(t)$ les polynômes estimés du modèle du procédé $\hat{A}(t, q^{-1})$ et $\hat{B}(t, q^{-1})$ donnés par:

$$\begin{cases} \hat{A}(t, q^{-1}) = 1 + \hat{a}_1 q^{-1} + \dots + \hat{a}_n q^{-n} \\ \hat{B}(t, q^{-1}) = \hat{b}_0(t) + \dots + \hat{b}_m q^{-m} \end{cases} \quad (3.22)$$

étape 2: Calcul des paramètres du régulateur ajustable.

Après avoir estimé les paramètres du système, on se propose de calculer les paramètres du régulateur. La structure du régulateur ajustable s'obtient du tableau (N°1) du chapitre 2 (page 43).
avec:

$$\Delta u(t) = f(\hat{R}, \hat{S}, \hat{T}) \quad (3.23)$$

les polynômes \hat{R}, \hat{S} et \hat{T} s'obtient à partir des polynômes E_j et F_j , dont A et B sont remplacés par leur estimés. On a alors à résoudre à chaque pas l'équation suivante [2]:

$$\begin{cases} 1 = F_j \hat{A} \Delta + q^{-j} F_j \\ \hat{G}_j = E_j \hat{B} \end{cases} \quad (3.24)$$

La résolution de l'équation (3.24) (réursive ou matricielle); à chaque pas, qui découle du principe de la commande adaptative indirecte a néanmoins deux inconvénients [35] [41]:

* Nécessite un temps de calcul important.

* Ne prend pas en compte les estimées précédente des paramètres du régulateur.

3.7 Commande prédictive généralisée adaptative: Approche Directe [39].

Dans cette partie, on s'intéresse à l'élaboration d'un schéma de commande adaptative directe concernant la GPC. En effet, au lieu d'identifier en permanence (temps réel) les paramètres du procédé et de calculer à chaque pas les paramètres du régulateur à partir des équations (3.24), (2.17) et (2.19), on cherche une structure particulière du modèle du processus où interviennent explicitement les paramètres de

la loi de commande [25]. Cette approche évite la résolution en temps réel de l'identité polynomiale de l'équation (3.24). C'est ce qu'on développera dans ce qui suit.

Reprenons l'équation (2.17) concernant le prédicteur optimale à j -pas obtenu précédemment.

$$\begin{cases} \Psi_j(t+j) = G_j \frac{\Delta u(t+j-1)}{C} + F_j \frac{y(t)}{C} \\ \Psi_j(t+j) = P y(t+j) \end{cases} \quad (3.25)$$

ou $y(t) = y_R(t)$ et G_j et F_j sont les résultats de la résolution de l'équation Diophantine.

donc:

$$\Psi_j(t+j) = -\sum_{i=1}^{nc} c_i \Psi_j(t+j-i) + G_j \Delta u(t+j-1) + F_j y(t) \quad (3.26)$$

Sachant que l'expression de la commande s'écrit en fonction des coefficients des polynômes G_j et F_j , et afin de déterminer directement les paramètres du régulateur GPC sans passer par l'estimation des paramètres du processus. L'identification de ces deux polynômes s'impose.

L'équation (3.26) s'écrit:

$$\Psi_j(t) = \theta_j(t) \phi_j(t) \quad (3.27)$$

et

$$\begin{cases} \theta_j^T(t) = [c_1 \dots c_{nc}, G_{(1,j)} \dots G_{(j-1+m,j)}, \\ \quad F_{(1,j)} \dots F_{(n,j)}] \\ \phi_j^T(t) = [-\Psi_j(t-1) \dots -\Psi_j(t-nc), \Delta u(t-1) \dots \Delta u(t-j+1-m), \\ \quad y(t-j) \dots y(t-j-n)] \end{cases} \quad (3.28)$$

La sortie auxiliaire Ψ_j , le vecteur d'observation ϕ et le vecteur de paramètres θ , constituent le modèle où interviennent directement les paramètres du régulateur à GPC. Il constitue donc le point de départ pour établir la forme implicite (directe) du régulateur auto-ajustable [25].

La méthode des MCR pourra être utilisée pour estimer le vecteur paramètres θ_j .

L'algorithme de commande GPCAD se résume en deux étapes:

étape 1: Identification.

- * Calculer la sortie $\Psi(t)$ à partir des mesures des variables $y(t)$, $w(t)$ et $u(t)$.
- * Former le vecteur d'observation $\phi_j(t)$.
- * Former les erreurs:

$$e_j(t) = P y(t) - \hat{\theta}_j(t) \phi_j(t)$$
- * Estimer le vecteur paramètre $\theta_j(t)$. en utilisant les équations (3.12) et (3.17) (algorithme à trace constante)

Cette étape se répète $N2-N1$ fois.

étape 2: Commande.

Utiliser l'estimation des vecteurs de paramètres $\hat{\theta}_j(t)$ et les observations pour calculer la commande $u(t)$ à partir de l'équation (2.19).

Ces deux étapes sont répétées à chaque pas de calcul, les deux schémas bloc relatifs à cette structure de commande sont représentés sur les figures (GPCA.12), (GPCA.13).

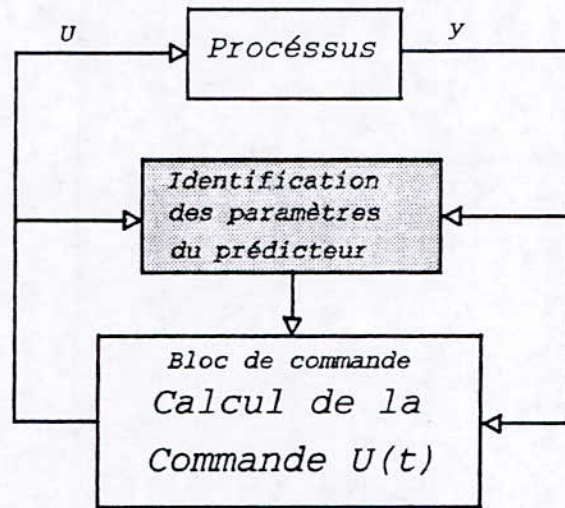


Figure (GPCA.12) Principe de la CA Directe.

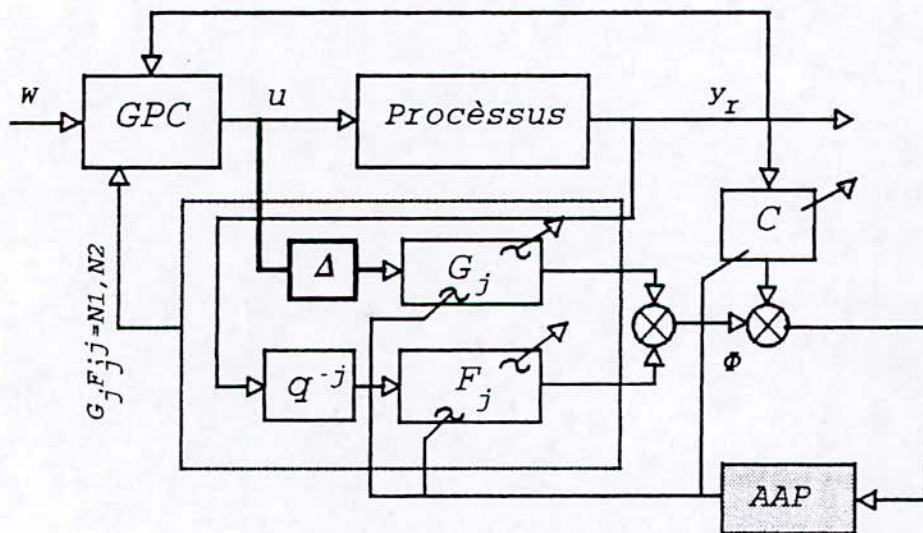


Figure (GPCA.13) Schéma bloc explicite et représentatif du régulateur GPCAI Directe.

Remarque.

Dans l'approche directe, on ne peut estimer les paramètres du polynôme d'observation qu'avec le vecteur paramètre θ_{N1} , car les autres vecteurs contiennent le même polynôme.

3.8 Résultats de simulation.

L'objectif de cette partie est de montrer l'aptitude de la GPC adaptative à faire face aux problèmes rencontrés lors de la commande des processus. En effet, les variations du retard, les variations paramétriques, la sousparamétrisation et la surparamétrisation (présence d'un facteur commun), peuvent affectés les performances de sortie du système à commander, néanmoins, ils peuvent être résolus d'une manière très simple par l'algorithme adaptatif de la GPC de base.

Pour prendre en compte d'éventuelles variations des paramètres du procédé pendant le déroulement de la commande, l'estimation doit être capable de suivre l'évolution des paramètres. Cette capacité d'adaptation peut être assurée par l'algorithme des moindres carrés récursif à trace constante avec un choix adéquat de la trace.

Les résultats, concernant la sortie et la commande sont représentés sur deux fenêtres différentes, simulé sur plusieurs périodes de 100, d'amplitude 1 et -1. La commande a été limitée à $U_{\max}=5$, les paramètres initiaux sont choisis égaux à 1. Cette partie est organisée de la façon suivante: après avoir étudié l'influence de la variation des polynômes A et B séparément puis A, B en même temps, nous verrons celles de la sous/surparamétrisation. La variation du retard sera traitée avant d'avoir exposé quelques résultats sur la GPCA directe.

3.8.1 Influence de la variation des paramètres.

Le système simulé est caractérisé par les polynômes suivants [25]:

$$\begin{cases} A(q^{-1})=1+a_1q^{-1}+a_2q^{-2}=1+0.8q^{-1}+0.15q^{-2} \\ B(q^{-1})=b_0+b_1q^{-1}=1+0.7q^{-1} \end{cases}$$

tel que A et B sont stables.

Pendant la durée de la variation nous avons fait varier les paramètres à partir de $t=120$ jusqu'à $t=300$ où le système reprend ces paramètres initiaux.

a) Variation du polynôme $A(q^{-1})$.

A l'instant $t=120$ le paramètre $a_1=0.8$ commute à 0.2 et $a_2=0.15$ à 0.5. Les figures (3.1)(3.2)(3.3) montrent que le régulateur obtenu par la GPCA réagit bien et maintient la poursuite de la référence après variation des paramètres pour $N1=1$ $N2=1$ $NU=1$, $N1=1$ $N2=10$ $NU=1$, $N1=1$ $N2=10$ $NU=5$ respectivement. Le cas $N1=1$ $N2=10$ $NU=10$ donne les mêmes performances que ceux obtenus sur la figure (3.3) qui est relatif à l'algorithme de base de la GPC avant l'introduction de NU .

b) Variation du polynôme $B(q^{-1})$.

En gardant le même système que précédemment, la même référence et le même algorithme d'identification ($tr=400$, $c=0.97$), nous avons varié les paramètres du polynôme B entre $t=120$ et $t=300$ tel que, le paramètre $b_0=1$ commute à 0.5 et

$b_1=0.7$ à 0.2 . Il est évident qu'à $t=120$ et $t=300$ le polynôme B reprend ces paramètres initiaux et ceci dans le but de faire subir au système et à la commande une autre variation dans le sens opposé.

Une bonne poursuite et bonne identification paramétrique ont été constatées, voir figure (3.4).

Une autre variation des paramètres du polynôme B a été réalisée de façon à rendre le système à phase non-minimale.

Les polynômes du système réel de présentent sous la forme suivante:

$$\begin{cases} A(q^{-1})=1+a_1q^{-1}+a_2q^{-2}=1-1.7q^{-1}+0.72q^{-2} \\ B(q^{-1})=b_0+b_1q^{-1}=0.8+0.4q^{-1} \end{cases}$$

La variation a été introduite comme précédemment entre 120 et 300 à savoir que $b_0=0.8$ commute à 0.4 et $b_1=0.4$ à 0.8 . Les résultats concernant la sortie, l'erreur entre le modèle et la sortie réel, le commande et les paramètres sont représentés sur la figure (3.5) pour $N1=1$ $N2=10$ $NU=1$. Nous tenons à signaler que pour $N1=N2=NU=1$ le système diverge entre $t=120$ et $t=300$ ce qui valide les résultats obtenus au premier chapitre pour les systèmes à PNM, dans le cas non adaptatif.

c) Variation des polynômes $A(q^{-1})$ et $B(q^{-1})$.

Cette étape est consacrée à la variation de tous les paramètres du système réel à savoir entre $t=120$ et $t=300$ nous avons:

$$A(q^{-1}) = \begin{cases} \text{de } a_1=0.8 \text{ commute à } 0.2 \\ \text{de } a_2=0.15 \text{ commute à } 0.5 \end{cases}$$

$$B(q^{-1}) = \begin{cases} \text{de } b_0=1 \text{ commute à } 0.5 \\ \text{de } b_1=0.7 \text{ commute à } 0.2 \end{cases}$$

Nous constatons que pour $N1=N2=NU=1$ et $N2=10$ figures(3.6)(3.7), de bons résultats ont été obtenus concernant la poursuite. Par contre une, une convergence paramétrique biaisée est observée pour $N2=10$ figure (3.7).

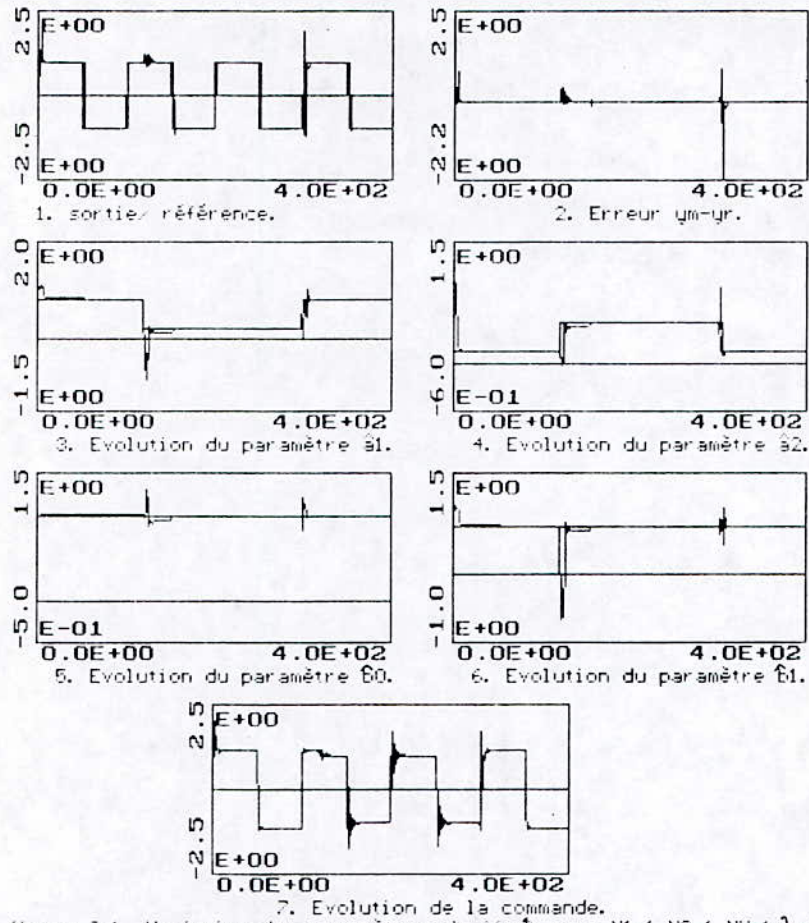


figure 3.1 Variation des paramètres de $\hat{h}(q^2)$ pour $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

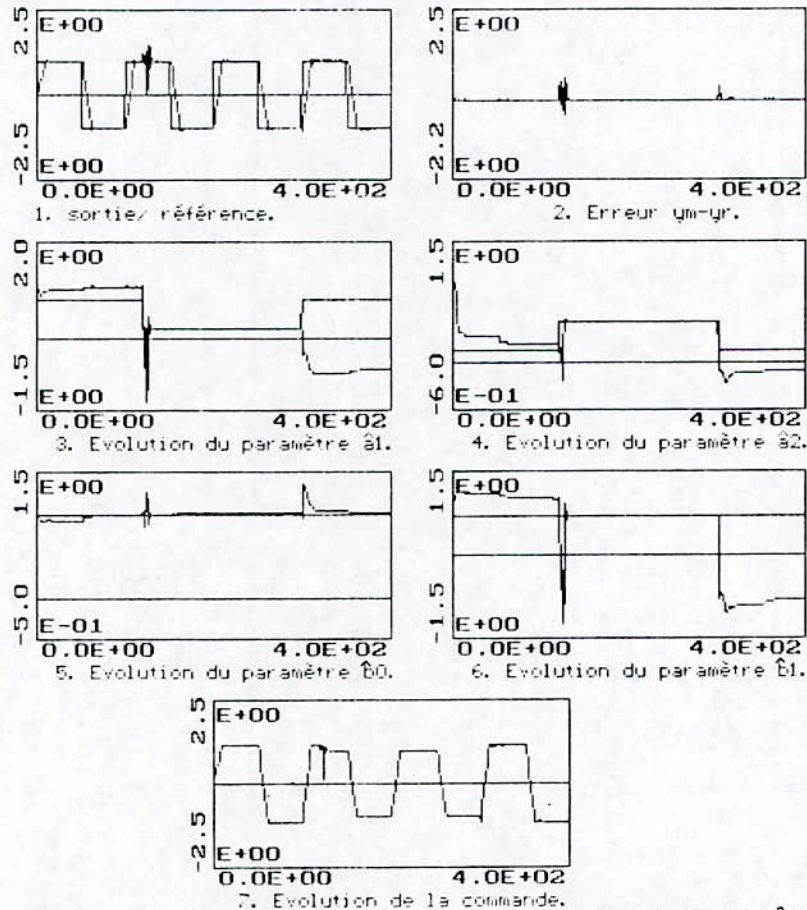


figure 3.2 Variation des paramètres de $\hat{h}(q^1)$ pour $N1=1$ $N2=10$ $NU=1$ $\lambda=0$.

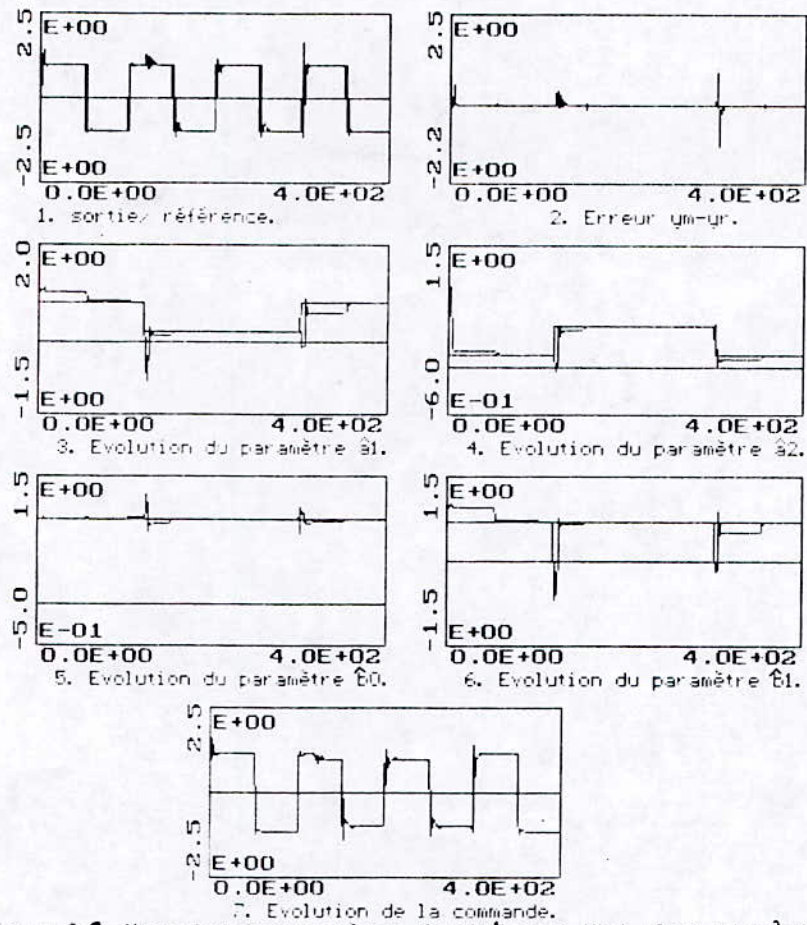


figure 3.3 Variation des paramètres de $\hat{B}q^1$ pour $N1=1$ $N2=10$ $NU=5$ $\lambda=0$.

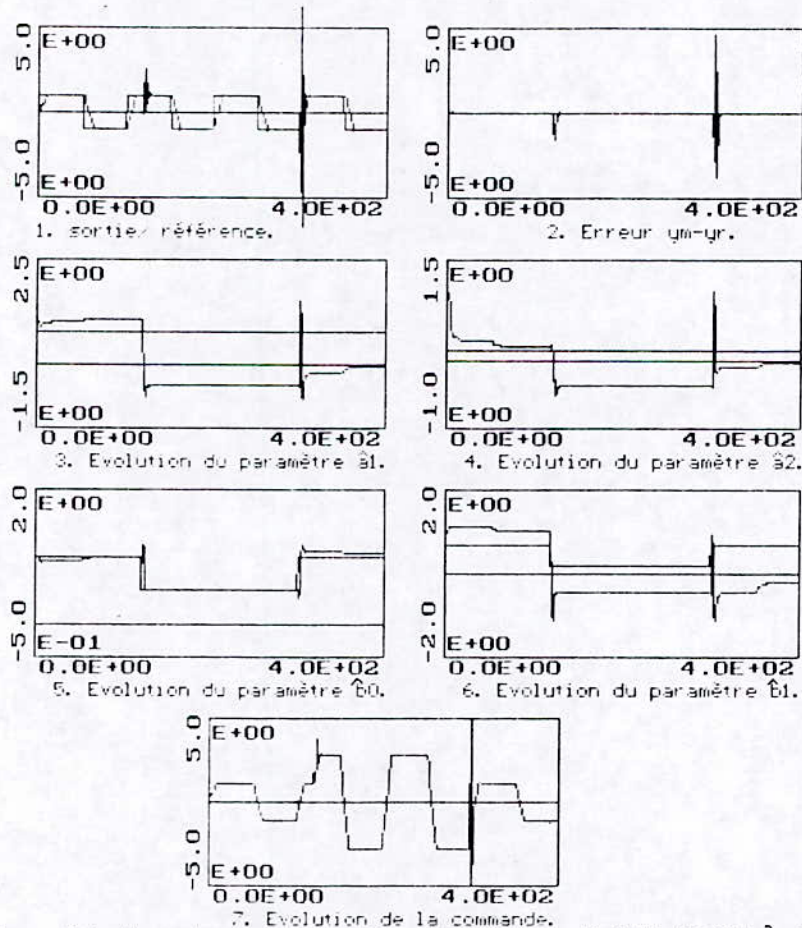


figure 3.4 Variation des paramètres de $\hat{B}q^2$ pour $N1=1$ $N2=10$ $NU=1$ $\lambda=0$.

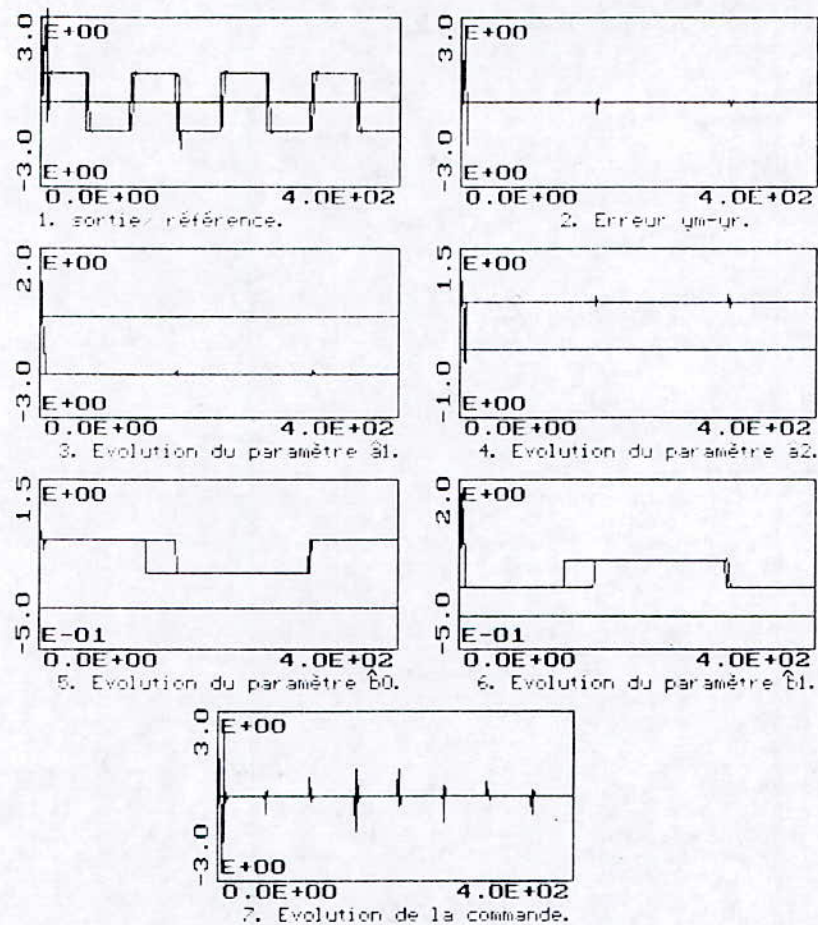


figure 3.5 Variation des paramètres de $B(q^2)(PMM)$ pour $N_1=1$ $N_2=2$ $NU=1$ $\lambda=0$.

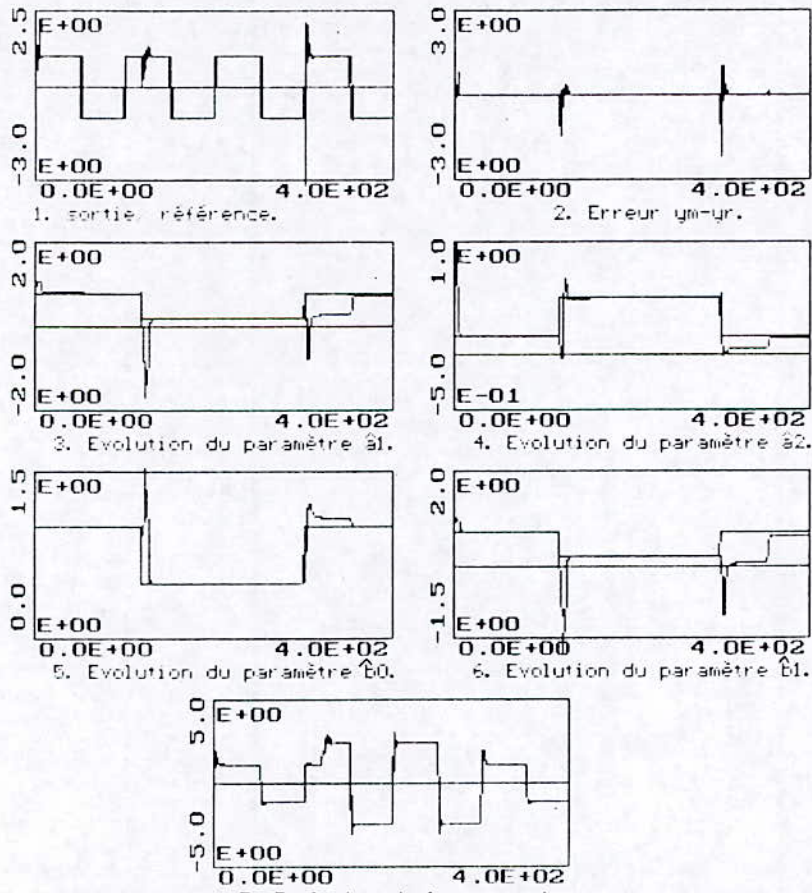


figure 3.6 Variation des paramètres de h et B pour $N_1=1$ $N_2=1$ $NU=1$ $\lambda=0$.

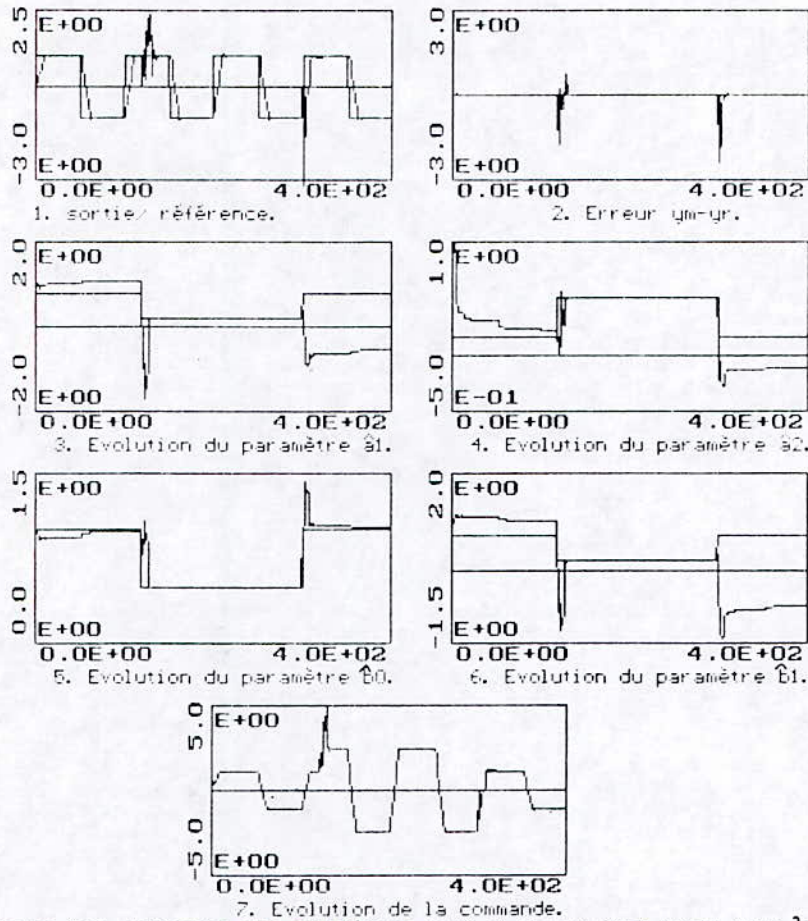


figure 3.7 Variation des paramètres de A et B pour N1=1 N2=10 NU=1 λ=0.

**3.8.2 Influence de la surparamétrisation
(Présence d'un facteur commun).**

En présence d'un facteur commun lors de modélisation, le système (simulé) peut s'écrire.

$$\begin{cases} \alpha(q^{-1})A(q^{-1}) = 1 + \tilde{a}_1 q^{-1} + \tilde{a}_2 q^{-2} = 1 - 0.4q^{-1} - 0.45q^{-2} \\ \alpha(q^{-1})B(q^{-1}) = \tilde{b}_0 + \tilde{b}_1 q^{-1} = 1 + 0.5q^{-1} \end{cases}$$

où $\alpha(q^{-1}) = 1 + 0.5q^{-1}$ et $\hat{n} = \hat{m} = 1$.

La figure (3.8) montre l'évolution de la sortie, de la commande et des paramètres du système réel. On peut dire que les paramètres obtenus représentent ceux du système réel après que l'identification s'est chargée de simplifier le facteur commun. La présence d'un facteur commun instable déstabilise le système.

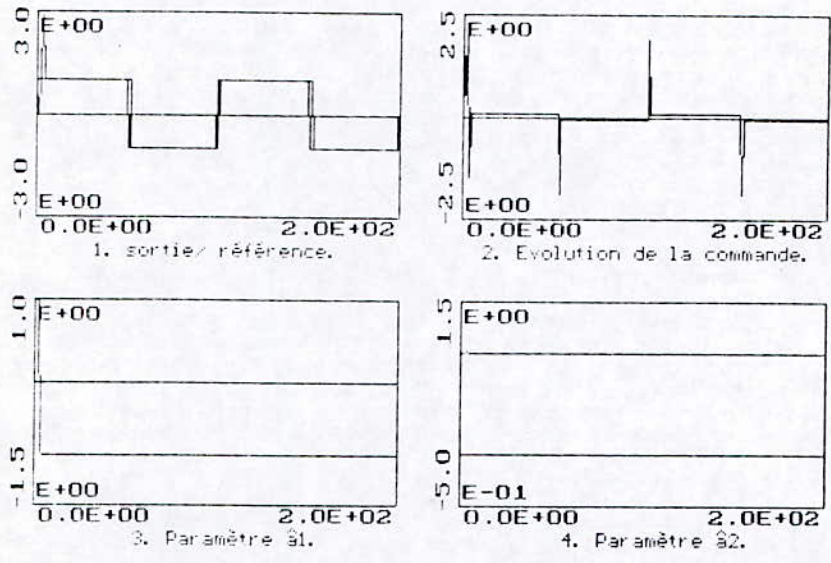


Figure 3.8 Evolution de la sortie, de la commande et des paramètres en Présence d'un facteur commun dans le système.

3.8.3 Influence de la sousparamétrisation [2].

Considérons le système continu de quatrième ordre suivant:

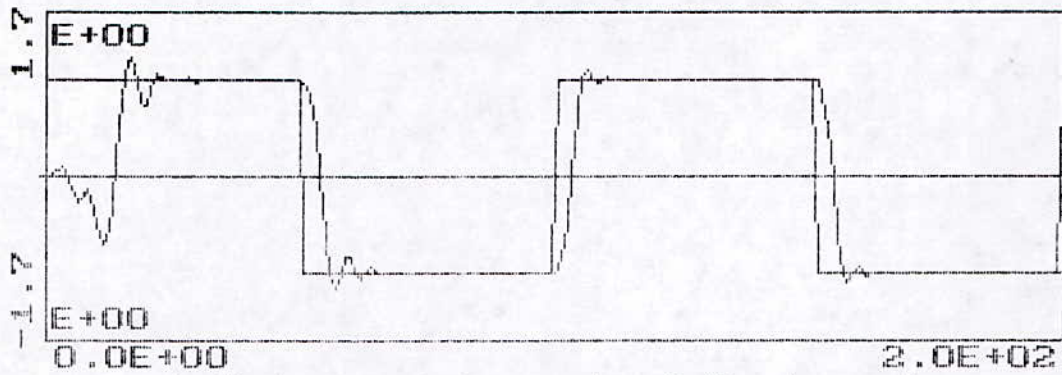
$$(1+s)^2(1+3s)^2y(t) = u(t)$$

Un modèle du deuxième ordre est estimé et le système est échantillonné à une période d'une seconde. Les polynômes obtenus après échantillonnage sont [2]:

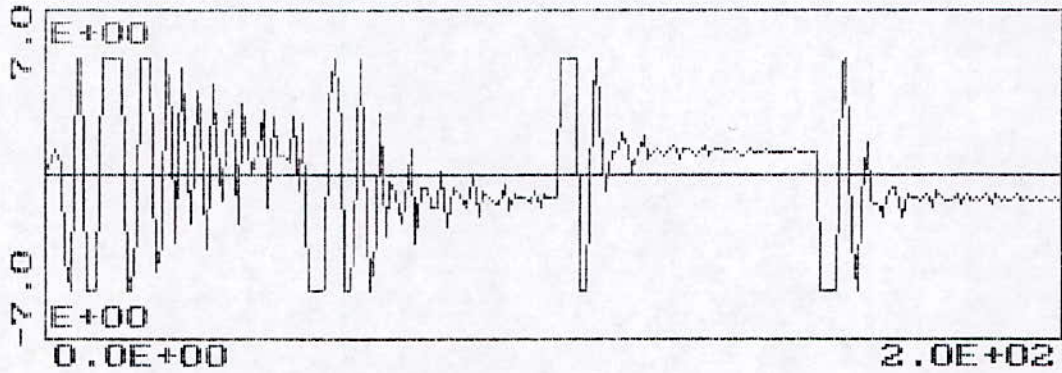
$$\begin{cases} A(q^{-1}) = 1 - 2.1688q^{-1} + 1.7031q^{-2} - 0.5717q^{-3} + 0.069483q^{-4} \\ B(q^{-1}) = 0.0027522 + 0.01815q^{-1} + 0.01065q^{-2} + 0.00055573q^{-3} \end{cases}$$

Dans ce cas deux paramètres du polynôme A et trois de B ont été estimé et le retard est imposé à un ($\hat{n}=2, \hat{m}=2, \hat{d}=1$).

Les figures (3.9)(3.10)(3.11), (3.12)(3.13)(3.14), (3.15)(3.16)(3.17) illustrent la sortie, la commande et les paramètres pour $\hat{n}=4, \hat{m}=3$ et $\hat{n}=2, \hat{m}=2$ respectivement pour $N1=N2=NU=1, N1=1 N2=5 NU=1, N1=1 N2=10 NU=1$.

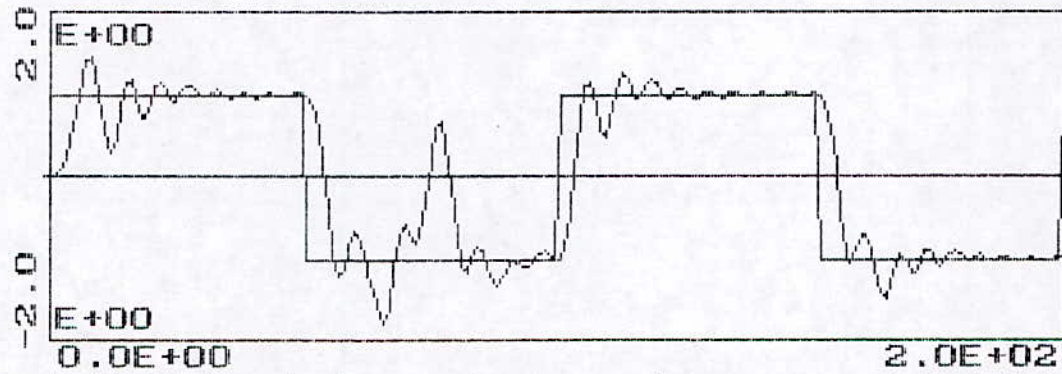


1. Evolution de la sortie réelle du système.

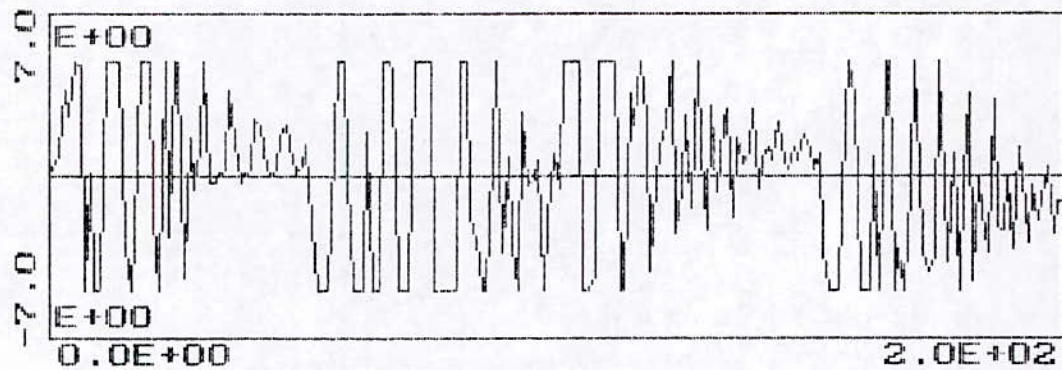


2. Evolution de la commande.

Figure 3.9 sortie et commande d'un système dont $\hat{n}=4$ $\hat{m}=3$ $\hat{d}=1$ pour $N1=1$ $N2=2$ $NU=1$ $\lambda=0$.



1. Evolution de la sortie du système sous-paramétrisée.



2. Evolution de la commande.

Figure 3.10 sortie et commande d'un système sous-paramétrisée dont $\hat{n}=2$ $\hat{m}=2$ $\hat{d}=1$ pour $N1=1$ $N2=2$ $NU=1$ $\lambda=0$.

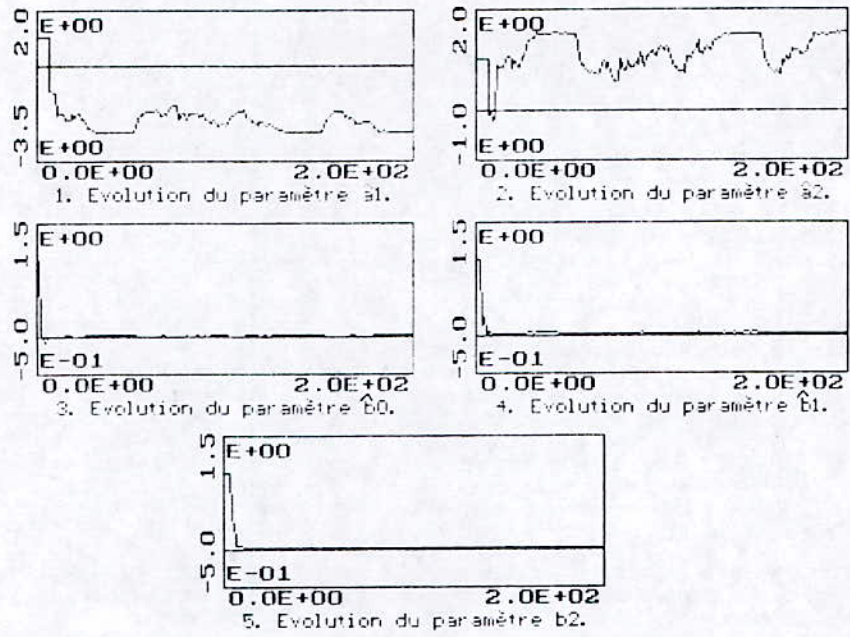


figure 3.11 Sous-Paramétrisation d'un système ($\hat{n}=2$ $\hat{m}=2$ $\hat{d}=1$) pour $N_1=1$ $N_2=2$ $NU=1$ $\lambda=0$.

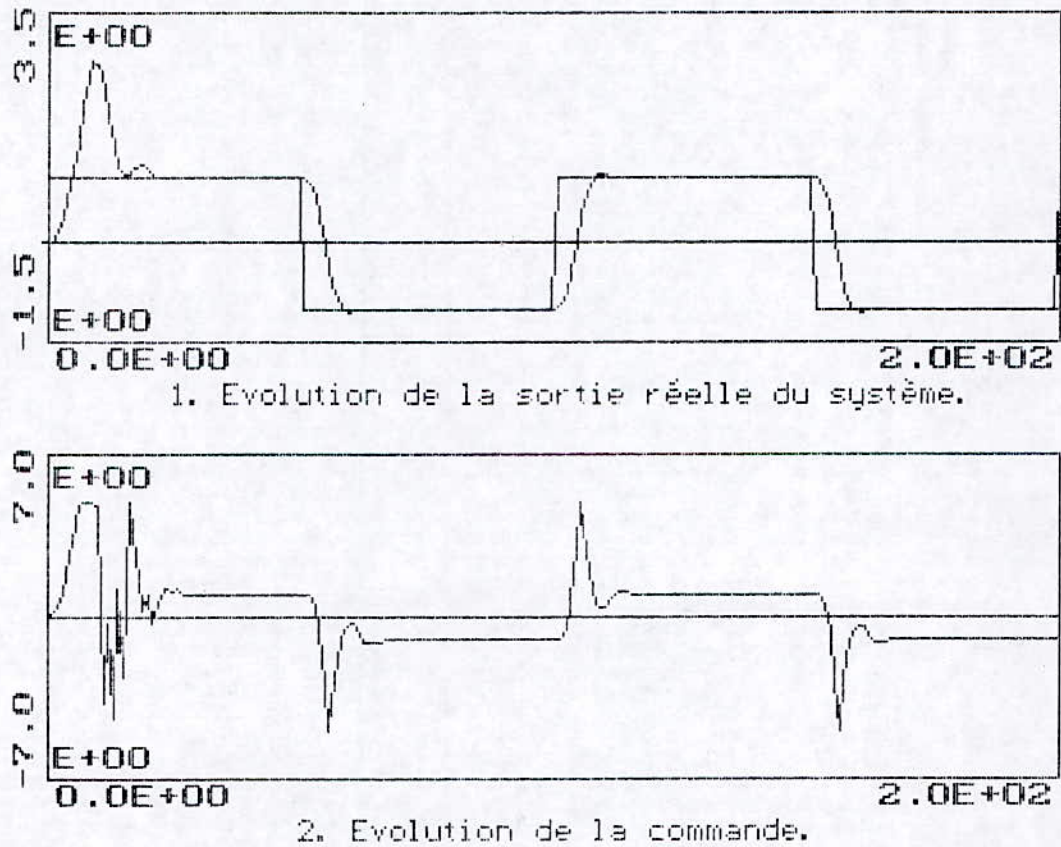
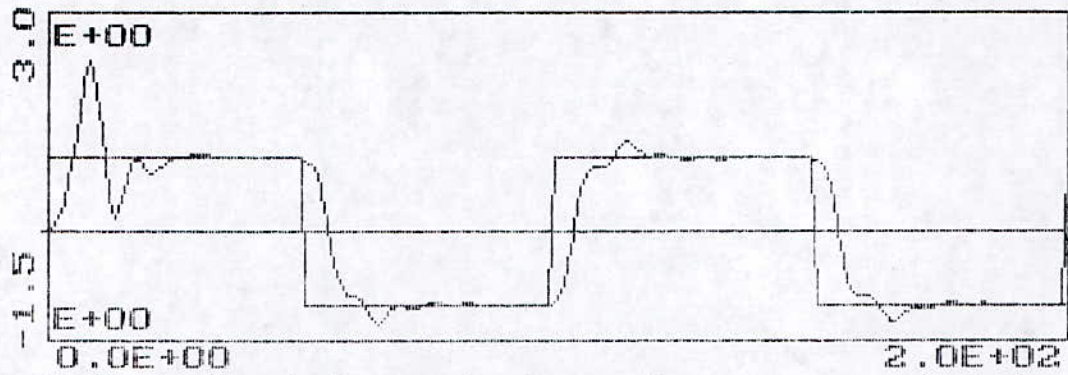
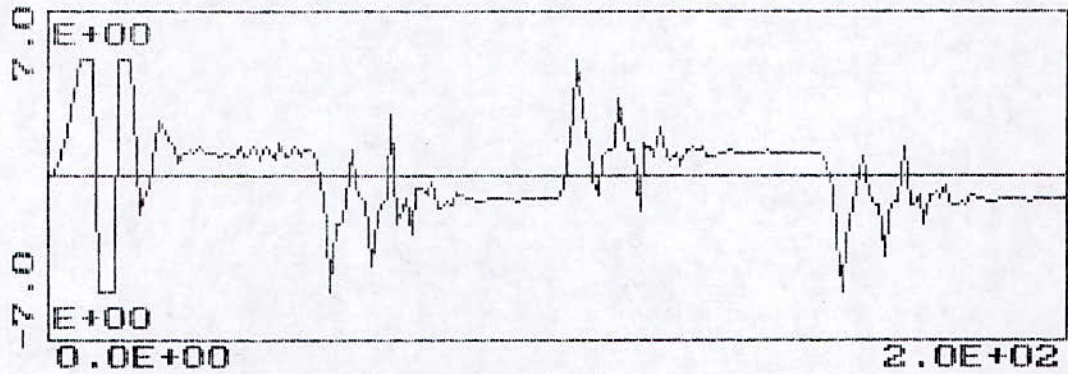


Figure 3.12 sortie et commande d'un système dont $\hat{n}=4$ $\hat{m}=3$ $\hat{d}=1$ pour $N_1=1$ $N_2=5$ $NU=1$ $\lambda=0$.



1. Evolution de la sortie du système sous-paramétrisée.



2. Evolution de la commande.

Figure 3.13 sortie et commande d'un système sous-paramétrisée dont $\hat{n}=2$ $\hat{m}=2$ $\hat{d}=1$ pour $N1=1$ $N2=5$ $NU=1$ $\lambda=0$.

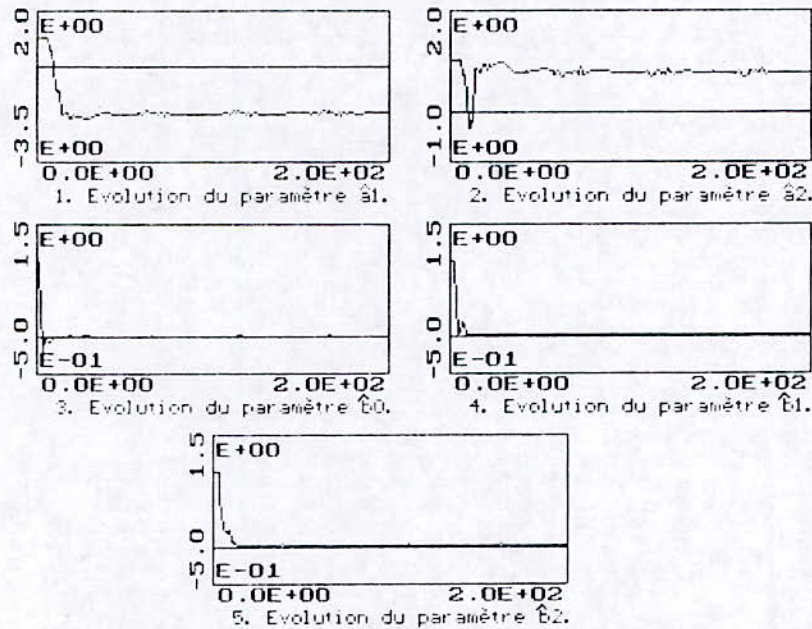
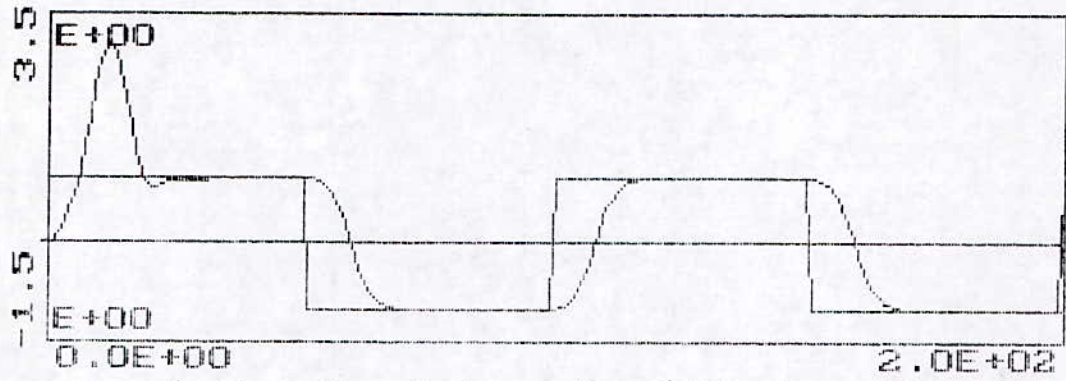
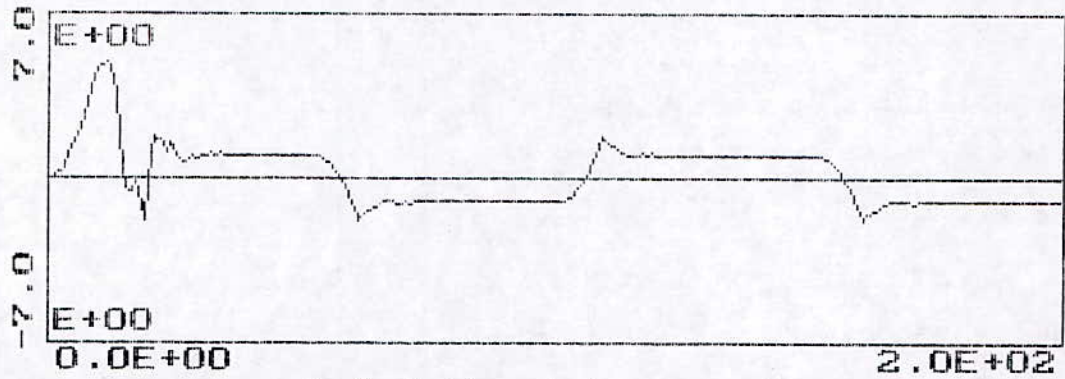


figure 3.14 Sous-Paramétrisation d'un système ($\hat{n}=2$ $\hat{m}=2$ $\hat{d}=1$) pour $N1=1$ $N2=5$ $NU=1$ $\lambda=0$.

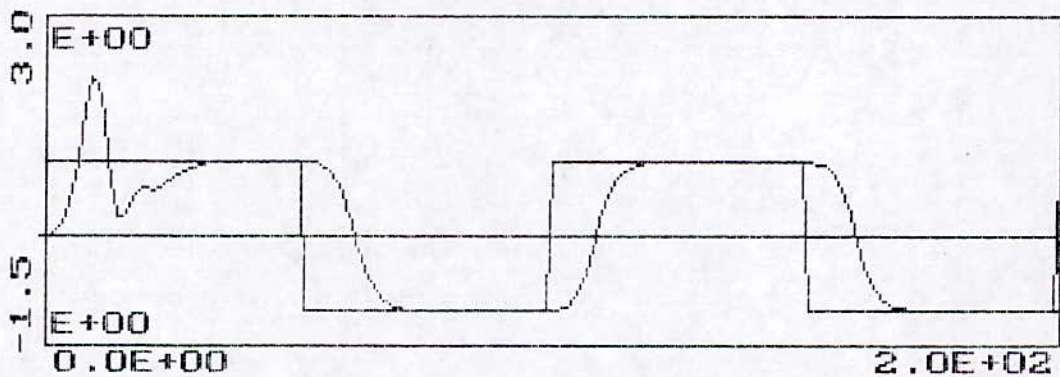


1. Evolution de la sortie réelle du système.

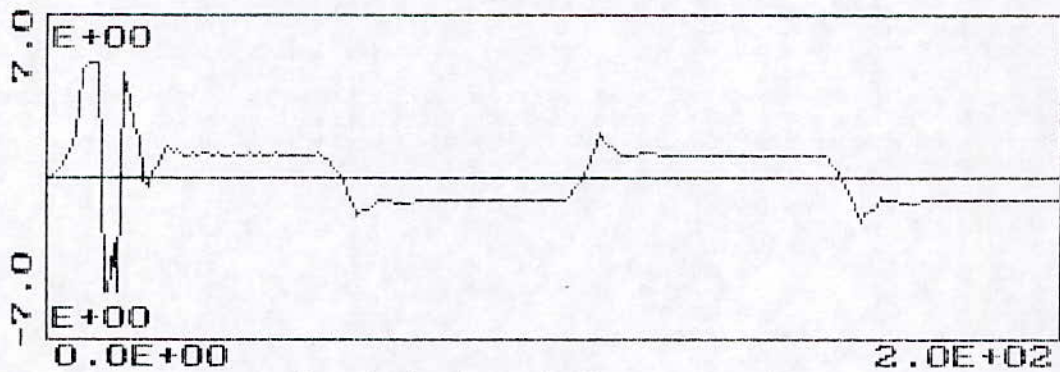


2. Evolution de la commande.

Figure 3.15 sortie et commande d'un système dont $\hat{n}=4$ $\hat{m}=3$ $\hat{d}=1$ pour $N1=1$ $N2=10$ $NU=1$ $\lambda=0$.



1. Evolution de la sortie du système sous-paramétrisée.



2. Evolution de la commande.

Figure 3.16 sortie et commande d'un système sous-paramétrisée dont $\hat{n}=2$ $\hat{m}=2$ $\hat{d}=1$ pour $N1=1$ $N2=10$ $NU=1$ $\lambda=0$.

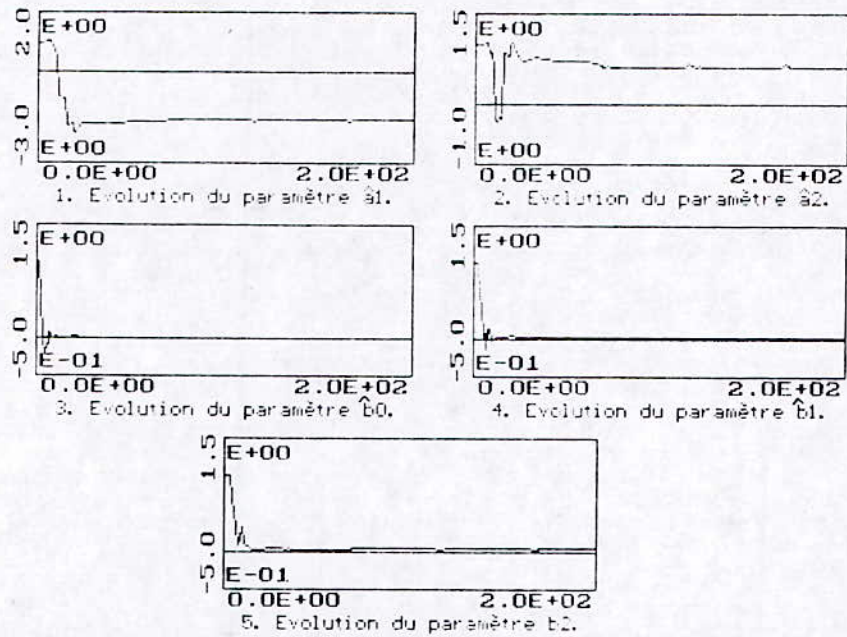


figure 3.27 Sous-Paramétrisation d'un système ($\hat{n}=2$ $\hat{m}=2$) pour $N1=1$ $N2=10$ $NU=1$ $\lambda=0$.

3.8.4 Influence du retard. (Variable ou inconnu)

Considérons le système instable et à PNM suivant [2]:

$$\frac{B}{A} = \frac{-0.1 - 0.2q^{-1}}{1 - 1.1q^{-1}}$$

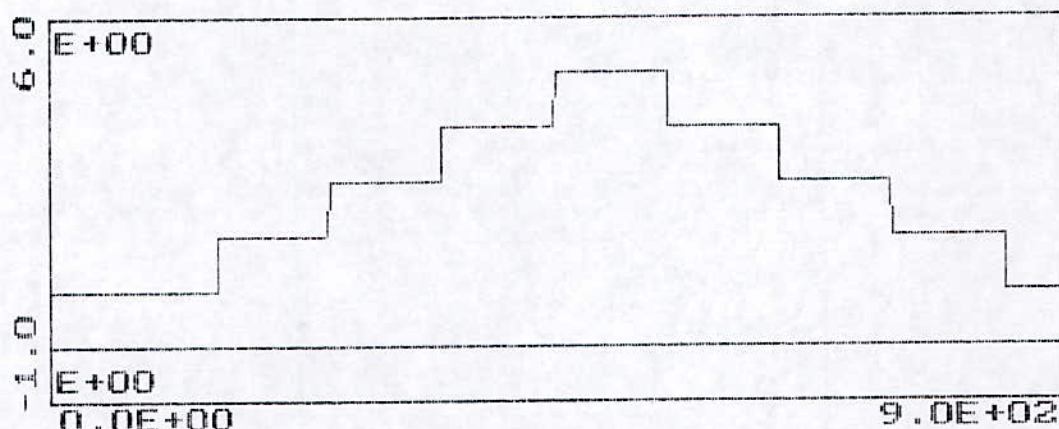
ou bien sous la forme entrées-sorties:

$$\begin{cases} (1-1.1q^{-1})y(t) = -(0.1 + 0.2q^{-1})u(t-d) \\ d = 1, 2, 3, 4, 5 \end{cases}$$

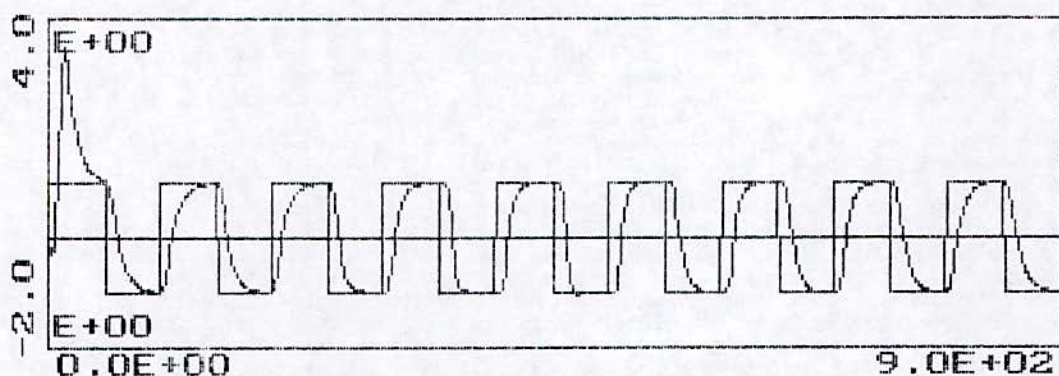
La variation du retard commence à $t=150$ jusqu'à 900 et cela de la valeur 2 jusqu'à 5 puis décrois de 5 à un.

Le régulateur adaptatif estime deux paramètres du polynôme \hat{A} et six paramètres de \hat{B} ($\hat{n}=2$ et $\hat{m}=5$).

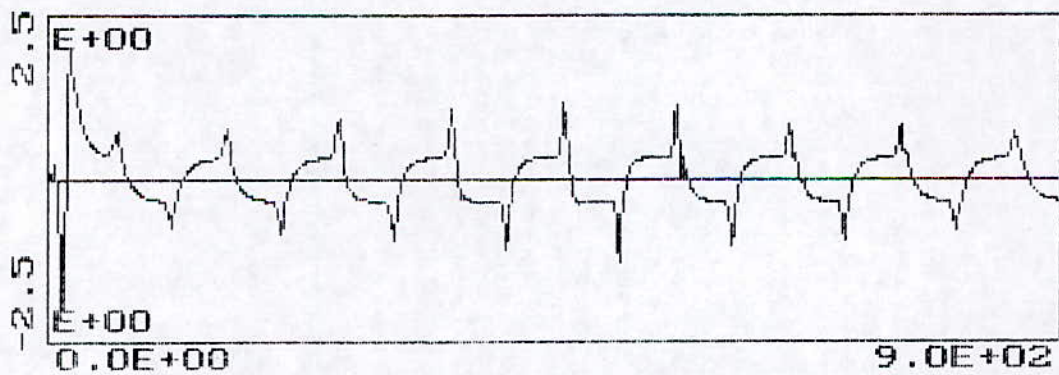
La figure (3.18)(3.19) illustre la sortie, la commande et les paramètres estimés pour $N1=1$ $N2=10$ $NU=1$.



1. Evolution du retard



2. Evolution sortie/référence.



3. Evolution de la commande.

figure 3.18 Comportement d'un système à des variations du retard pour $N1=1$ $N2=10$ $NU=1$ $\lambda=0$.

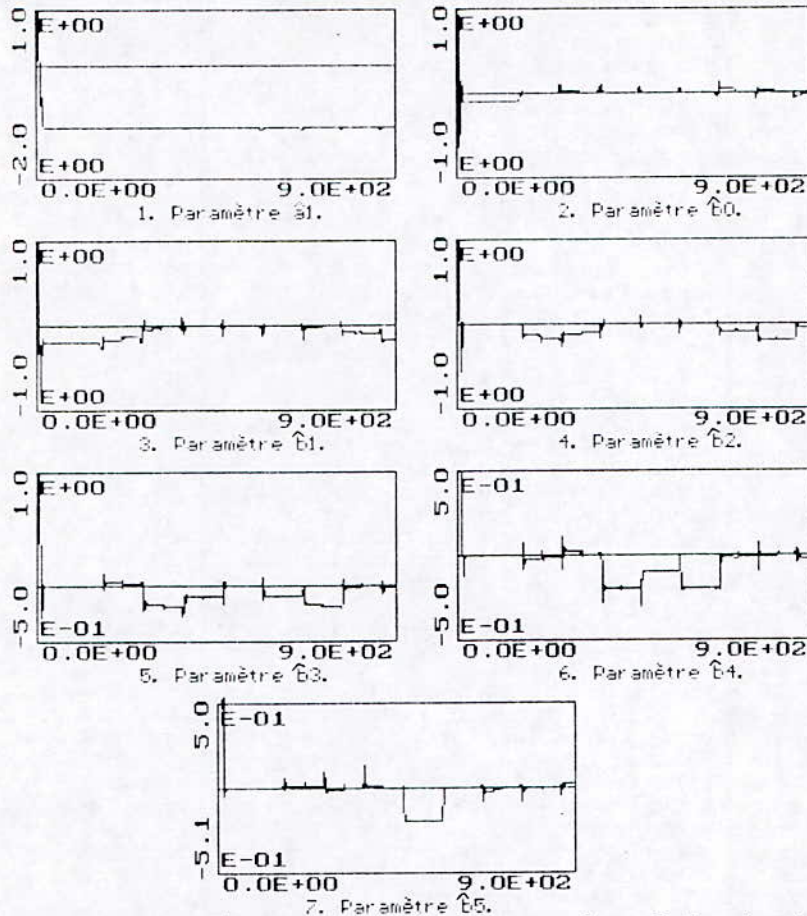


figure 3.19 Paramètres estimés du système pour la variation du retard.

3.8.5 Simulation de la GPC Directe. (Cas de la variation paramétrique)

L'approche directe de la GPCA est testée à une variation paramétrique des polynômes *A* et *B* relatif au paragraphe (3.8.1), le système s'écrit:

$$\frac{B}{A} = \frac{1 + 0.7q^{-1}}{1 + 0.8q^{-1} + 0.15q^{-2}}$$

pendant la variation, le système devient:

$$\frac{B}{A} = \frac{0.5 + 0.2q^{-1}}{1 + 0.2q^{-1} + 0.5q^{-2}}$$

Dans cette approche on s'intéresse à identifier directement les paramètres du régulateur à partir de l'équation du prédicteur optimale, contrairement au cas de la GPCAI où on estimait le paramètres du système puis calculer ceux du régulateur à chaque instant.

La figure (3.20) illustre la sortie, l'erreur, la commande et les paramètres G_j et F_j pour $N1=N2=NU=1$.

D'autres paramètres G_j et F_j seront identifiés pour $N2=2$ dont les performances de sortie et l'évolution des paramètres sont consignés sur les figures (3.21)(3.22) (3.23).

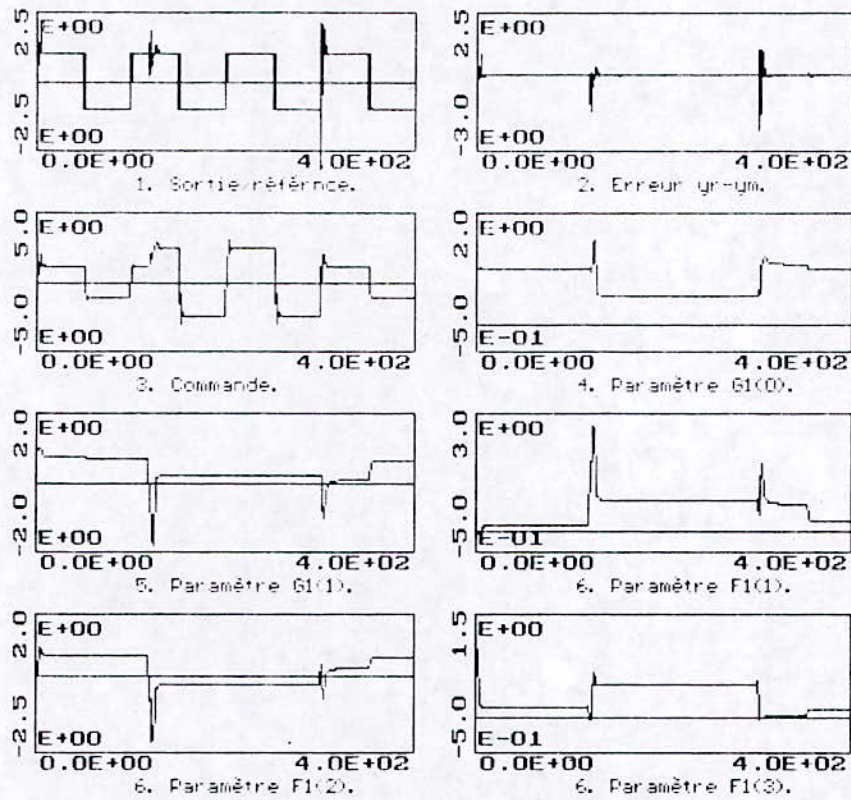


figure 3.20 Sortie/commande/erreur et évolution des paramètres (G1,F1) estimés du régulateur à une variation de A et B pour $N1=1$ $N2=1$ $NU=1$ $\lambda=0$.

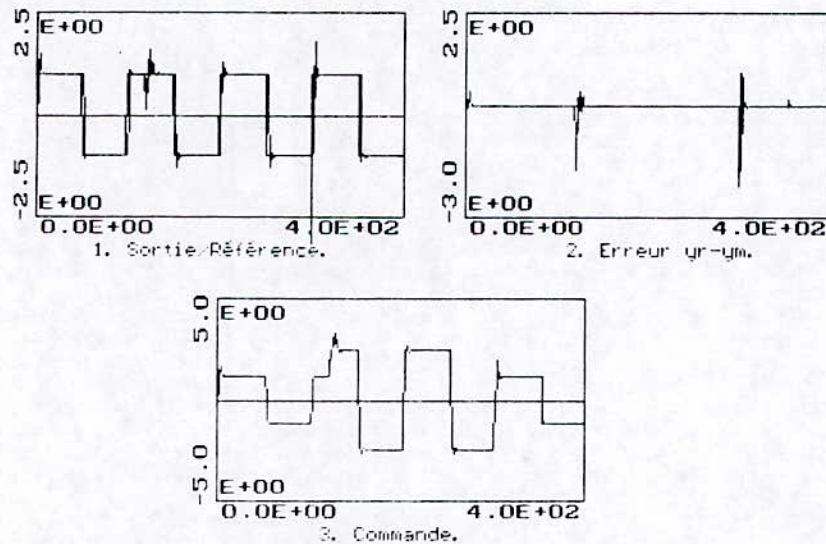


figure 3.21 Evolution de la sortie/commande/erreur de la GPC Direct à une variation paramétrique de A et B pour $N1=1$ $N2=2$ $NU=1$ $\lambda=0$.

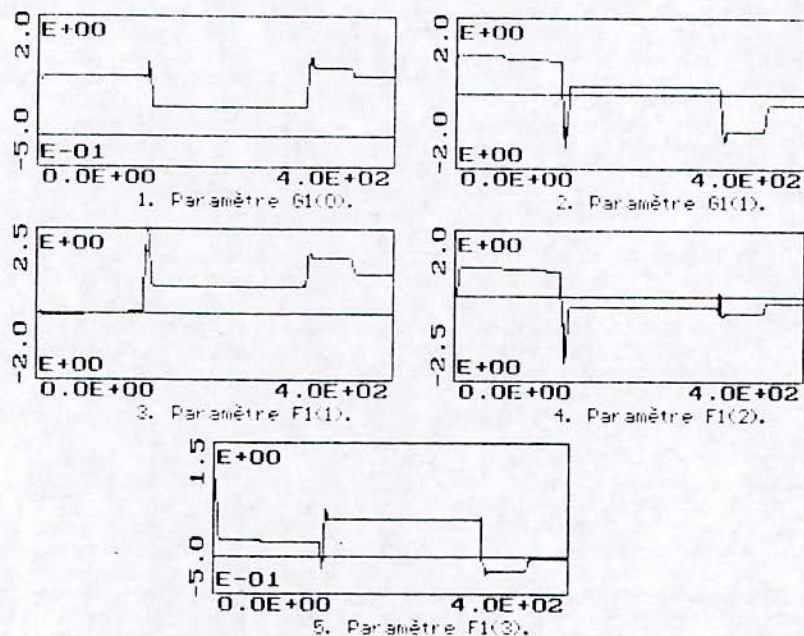


figure 3.22 Evolution des paramètres (G1,F1) estimés du régulateur à une variation de R et E pour N1=1 N2=2 NU=1 $\lambda=0$.

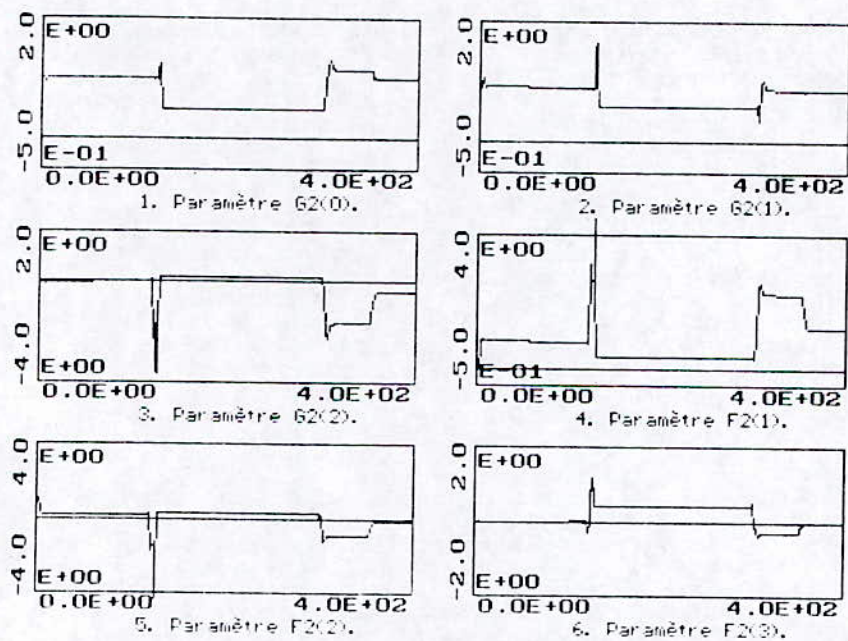


figure 3.23 Evolution des paramètres (G2,F2) estimés du régulateur à une variation de R et E pour N1=1 N2=2 NU=1 $\lambda=0$.

Conclusion.

Le but de ce chapitre a été consacré à l'étude de quelques schémas de commande adaptative. Nous nous sommes attachés en particulier à étudier la loi de commande prédictive généralisée adaptative directe et indirecte. Ces deux approches convergent vers des résultats identiques. Seulement, du point de vue calcul numérique, l'approche indirecte effectue quelques calculs supplémentaires par rapport à l'approche directe, à savoir, la résolution de l'équation Diophantine. Cette méthode identifie directement les paramètres du régulateur sans se soucier de ceux du système. Contrairement à l'approche indirecte où l'identification des paramètres du système est nécessaire pour le calcul de la commande. Sachant que nous faisons de la simulation le temps de calcul ne nous "dérange" pas mais si on est amené à une implémentation réelle de la commande, l'approche directe est la mieux adaptée. Ceci en faisant un choix adéquat de l'horizon maximal de prédiction puisqu'il est en relation directe avec le nombre de paramètres du régulateur à identifier.

Pour assurer l'adaptation des paramètres du modèle utilisé dans la synthèse des lois de commande auto-ajustables, nous avons étudié l'algorithme des moindres carrés récursif et ces variantes et nous avons opté pour celui à trace constante qui est capable d'assurer une poursuite et une convergence des paramètres dans le cas où ces derniers varient dans le temps.

La simulation a été consacrée à la commande de quelques systèmes présentant les caractéristiques suivantes:

- Systèmes à paramètres variables.
- Systèmes surparamétrisés (facteur commun).
- Systèmes sousparamétrisés.
- Systèmes à retard variable ou inconnu.

En ce qui concerne la variation des paramètres, la sortie du système suit très bien la référence même si les paramètres convergent vers des valeurs biaisées. Un léger transitoire est constaté au début de chaque simulation, dû à la méconnaissance des véritables paramètres. Cette variation ne cause aucun problème pour la GPCA si au cours de la simulation le système devient à phase non-minimale et ceci sans introduire une pondération sur la commande.

La convergence des paramètres est grandement influencée par le choix des paramètres de synthèse de la GPC. En effet, on a eu une convergence vers des paramètres réels du système pour de faibles valeurs de l'horizon de commande, contrairement à d'autres horizons plus élevés. Cela est dû au fait que la commande qui attaque le système et le modèle n'est pas trop excitée. Une solution à ce problème peut être aussi apportée par l'augmentation de l'horizon de commande.

Les résultats concernant la surparamétrisation et la sousparamétrisation montrent que la GPCA surmonte le problème de présence d'un facteur commun et de la commande d'un système avec un nombre de paramètres réduit (ordre du modèle réduit). La présence d'un facteur commun instable déstabilise le système, ceci est dû à la présence de ce même facteur dans le numérateur et le dénominateur de la fonction de

transfert en boucle fermée. Ce qui confirme que lors de la résolution de l'équation Diophantine, les polynômes constituant le système doivent être premiers entre eux. En ce qui concerne un facteur commun stable, l'identification s'en charge de la simplification.

Les performances obtenues dans le cas de la sousparamétrisation, pour plusieurs valeurs de l'horizon de prédiction, montrent que pour une valeur élevée, nous obtenons des dynamiques de sortie presque identiques, contrairement à une valeur inférieure où la dynamique change mais la référence est atteinte.

La version adaptative de la GPC a montré qu'elle est capable de garder de bonne performance au problème de variation du retard même important, et ceci à condition de choisir un modèle d'identification surparamétré. L'observation des paramètres montrent qu'au cours des variations du retard imposé au système, quelques paramètres oscillent autour de zéro et d'autres non. Ce qui montre que le retard a été bien identifié. Dans notre travail, nous avons améliorés les résultats obtenus par Clarke [2]. En effet, nous avons utilisé un algorithme d'identification à trace constante qui permet d'obtenir un modèle plus précis.

Finalement, ces travaux montrent que les résultats obtenus permettent de dire que la version adaptative de la GPC directe et indirecte nous permet de résoudre un grand nombre de problèmes. Une attention particulière doit être portée au choix des paramètres de synthèse de la GPC.

Chapitre 4

MODELISATION DE DEUX ROBOTS MANIPULATEURS

*"La pensée n'est qu'un éclair au milieu d'une longue nuit,
mais c'est cet éclair qui est tout".*

HENRI POINCARRE

Chapitre 4

MODELISATION DE DEUX ROBOTS MANIPULATEURS (PUMA, SCARA).

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 ses grandeurs de sorties. L'ensemble de ces relations constituent le modèle mathématique du système considéré. 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, et du modèle dynamique inverse, en utilisant le principe de D'Alembert (algorithme de Newton-Euler (NE)).

4.1 Présentation des robots.

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 uns par rapport aux autres. Cet ensemble a pour objectif de mener l'organe terminal vers un lieu géométrique imposé par la tâche" [42].

Les robots manipulateurs que nous nous proposons de modéliser sont le PUMA et le SCARA [43][44]. Le robot PUMA est caractérisé par deux articulations rotationnelles θ_1 et θ_2 , présenté sur la figure (MOD.1). Le robot SCARA est caractérisé par les mêmes articulations mais évoluant sur un espace horizontal présenté sur la figure (MOD.2).

Pour pouvoir établir les différents modèles, plusieurs hypothèses doivent être prises en considération [45]:

- Les frottements sont de nature visqueuse, et linéaire par rapport à la

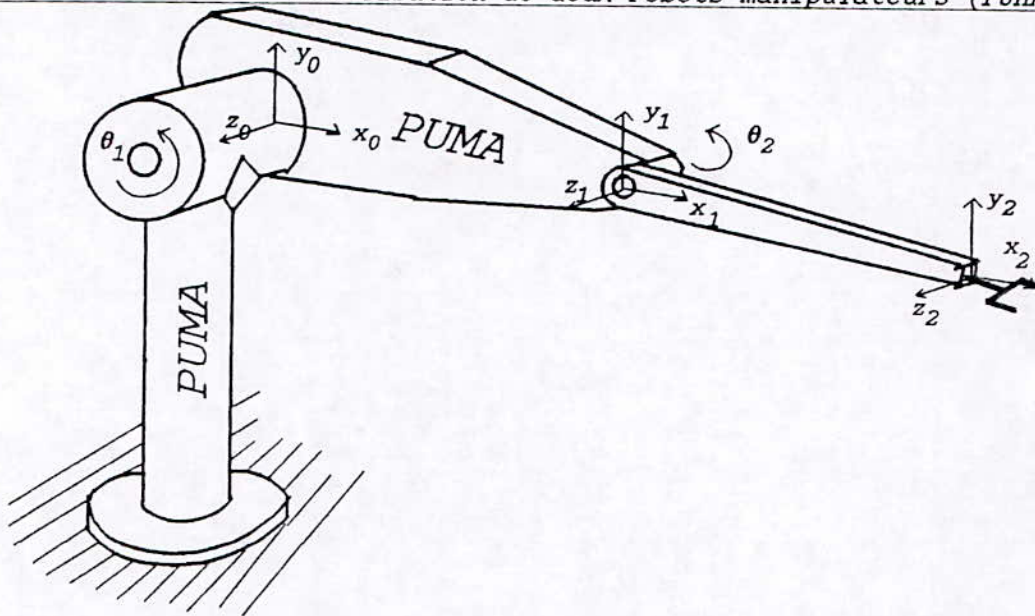


Figure (MOD.1) Présentation du robot PUMA.

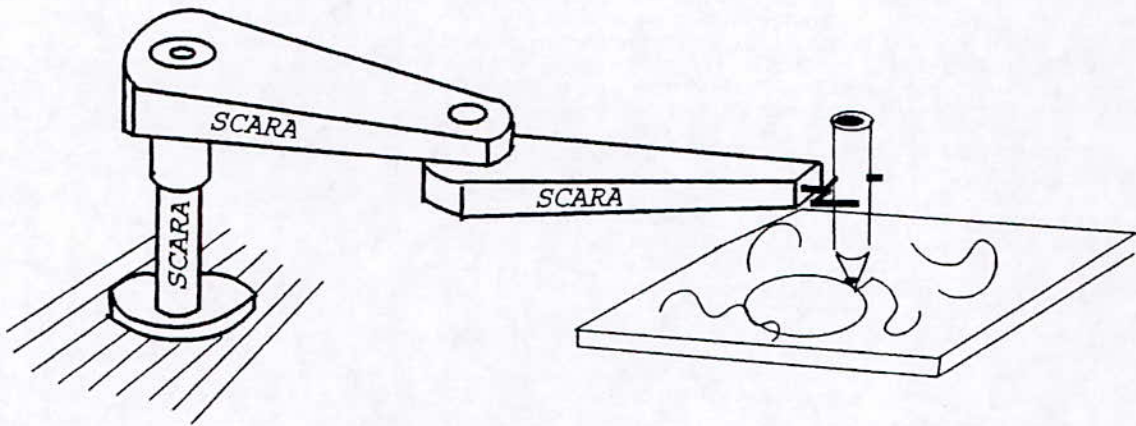


Figure (MOD.2) Présentation du robot SCARA.

vitesse généralisée.

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

4.2 Modélisation cinématique des deux robots.

Tout manipulateur peut être considéré comme une chaîne de liaisons connectées par des articulations. Chaque liaison est caractérisée 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 [46] 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 la transformation de Denavit et Hartenberg, est présentée ci-dessous.

Pour l'application de cette transformation, il faut d'abord fixer les différents

repères dans chaque articulation d'une façon successive, en procédant suivant les étapes suivantes:

étape 1: Numéroté chaque liaison et articulation, en commençant de la base, notée liaison 0, et l'élément terminal, noté liaison n voir figure (MOD.3). La liaison i se déplace, soit autour (rotative) ou le long (prismatique) par rapport à $i-1$,

étape 2: Etablir le repère de chaque articulation, en suivant les règles suivantes:

- L'axe Z_{i-1} est choisi le long de l'axe de l'articulation i .
- L'axe X_i est choisi perpendiculaire à Z_{i-1} , dont le sens peut être choisi arbitrairement ($X_i = Z_{i-1} \times Z_i$).
- L'axe Y_i est choisi, de telle sorte à former un trièdre droit.

étape 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 .

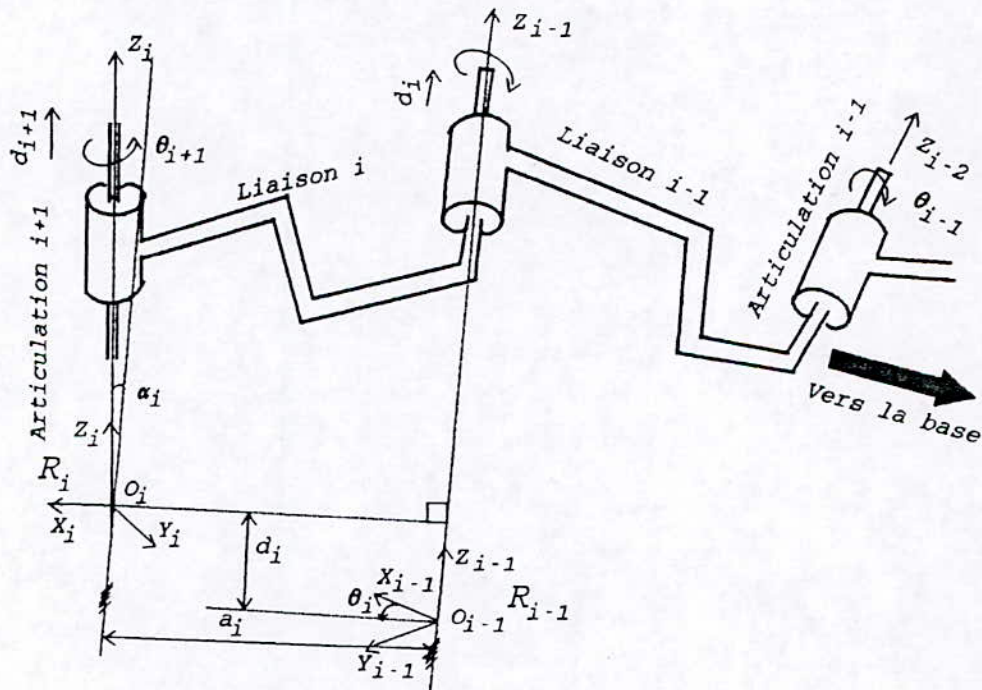


Figure (MOD.3) Représentation des repères dans les articulations.

4.2.1 Le robot PUMA.

L'implementation de cette algorithme dans le cas de robots manipulateurs à deux degrés de liberté est présenté sur la figure (MOD.4).

Tous les axes de rotation des articulations sont autour de l'axe Z. Cet axe est perpendiculaire au plan de la feuille.

On définit les paramètres suivants:

- θ_1, θ_2 : variables articulaires.
- m_1, m_2 : masses des liaisons.
- $\alpha_1, \alpha_2, d_1, d_2, a_1, a_2$: paramètres de liaison.

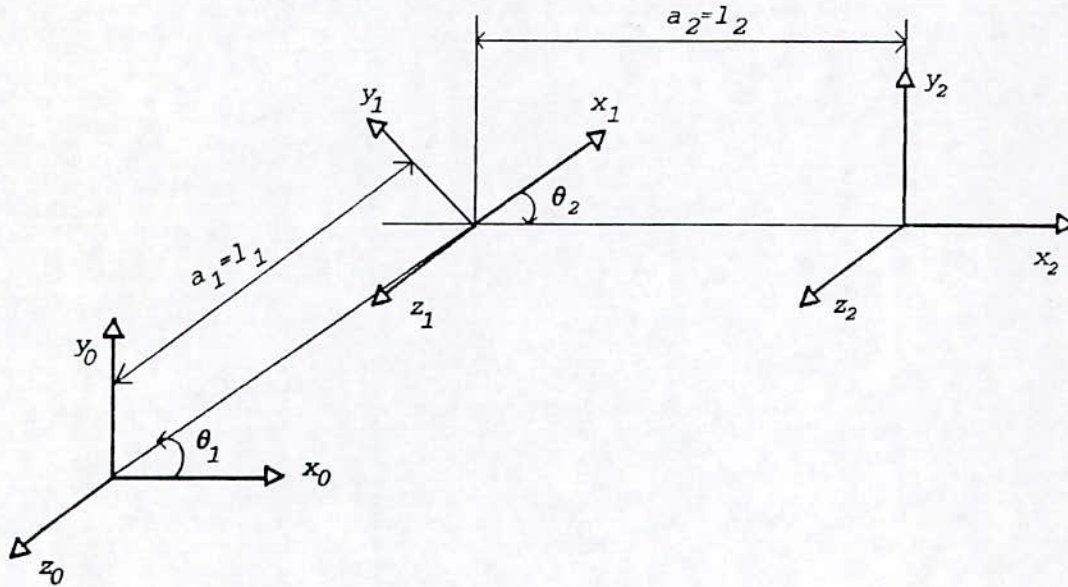


Figure (MOD.4) Détermination des paramètres de liaisons.

On construit alors le tableau suivant:

i	Variables	α_i	a_i	d_i	θ_i
1	θ_1	0	l_1	0	θ_1
2	θ_2	0	l_2	0	θ_2

Posons $l_1 = l_2 = l$ et ${}_{i-1}T^i$ matrice de transformation de coordonnées homogènes (ou matrice de passage) qui relie le repère R_i au repère R_{i-1} , elle s'exprime sous la forme suivante.

$${}_{i-1}T^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

Ainsi nous tirons les matrices de transformation, représentant les deux rotations, d'un repère à un autre.

$${}^0T^1 = \begin{bmatrix} C_1 & -S_1 & 0 & lC_1 \\ S_1 & C_1 & 0 & lS_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; \quad {}^1T^2 = \begin{bmatrix} C_2 & -S_2 & 0 & lC_2 \\ S_2 & C_2 & 0 & lS_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} ;$$

la matrice de transformation du repère R_0 à R_2 est:

$${}^0T^2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & l(C_{12}+C_1) \\ S_{12} & C_{12} & 0 & l(S_{12}+S_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ;$$

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

4.2.2 Le robot SCARA.

Pour le robot SCARA, il suffit de tourner les repères des articulations du robot PUMA, dans le cas où sa configuration est horizontale ($\theta_1 = \theta_2 = 0^\circ$), d'un angle de -90° , autour de l'axe x_i [44].

4.3 Modélisation géométrique des deux robots.

Les transformations précédentes permettent d'exprimer la position de l'organe terminal par rapport au repère R_0 sous la forme d'une fonction $\bar{F}(R_0)$ à partir des variables articulaires q_i .

$$\bar{F}(R_0) = F(\bar{q}) \tag{4.2}$$

$$\text{avec: } \begin{cases} \bar{q}^T = [q_1 \ q_2] \\ \bar{F}^T = [x \ y \ z \ 1] \\ F(\bar{q}) = {}^0T^2 \ {}_2\bar{F}^T \end{cases}$$

et ${}_2\bar{F}^T$: coordonnées de l'élément terminal dans R_2 .

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

Dans le cas des deux robots on a: ${}_2\bar{F}^T = [0 \ 0 \ 0 \ 1]$

$$\text{d'où } \begin{cases} r = {}^0T^2 \ {}_2\bar{F}^T \\ \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} l(C_{12}+C_1) \\ l(S_{12}+S_1) \\ 0 \\ 1 \end{bmatrix} \end{cases}$$

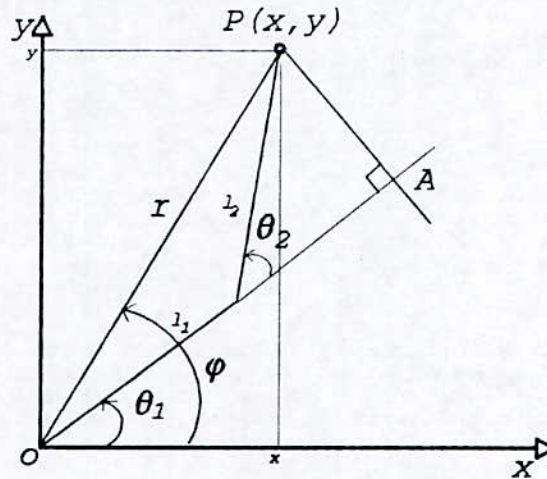
avec: $\bar{q}^T = [\theta_1, \theta_2]$

Cette équation est appelée modèle géométrique du robot. C'est à partir de cette

équation qu'on peut tirer les coordonnées généralisées q_i . On déduit l'équation suivante:

$$\begin{cases} x=l(C_{12}+C_1) \\ y=l(S_{12}+S_1) \end{cases} \quad (4.3)$$

On peut remarquer qu'il est très difficile de tirer les coordonnées généralisées en fonction des coordonnées cartésiennes. On présente alors une méthode purement géométrique pour déterminer θ_1 et θ_2 en fonction respectivement des coordonnées cartésiennes x, y ou polaire r, φ .



de la figure ci-dessus on trouve:

$$\begin{cases} x=l_1 \cos \theta_1 + l_2 \cos (\theta_1 + \theta_2) \\ y=l_1 \sin \theta_1 + l_2 \sin (\theta_1 + \theta_2) \end{cases}$$

On remarque que si $l_1=l_2=l$ on retrouve la même équation que l'équation (4.3). Dans le triangle OAP on a:

$$r^2 = (l_1 + l_2 \cos \theta_2)^2 + (l_2 \sin \theta_2)^2$$

d'où l'on tire dans le cas $l_1=l_2=l$ et $r < 2l$:

$$\theta_2 = \arccos \left[\frac{1}{2} \left(\frac{r}{l} \right)^2 - 1 \right] \quad \text{où} \quad r^2 = x^2 + y^2 \quad (4.4)$$

Sachant que:

$$\frac{l_2}{\sin(\varphi - \theta_1)} = \frac{r}{\sin(\pi - \theta_2)}$$

d'où l'on tire dans le cas $l_1=l_2=l$ et $|l_2 \cos \theta_2| < r$:

$$\theta_1 = \varphi - \arcsin \left[\frac{l}{r} \sin(\pi - \theta_2) \right] \quad \text{où} \quad \varphi = \arctan \left(\frac{y}{x} \right) \quad (4.5)$$

on déduit alors les conditions d'existence de θ_1 et θ_2 ;

$$\begin{cases} \theta_2: \text{ existe toujours} \\ \theta_1: \text{ existe si } |l_2 \cos \theta_2| \leq r \end{cases} \quad (4.7)$$

La détermination des coordonnées généralisées a été établie à partir d'un seul point dans l'espace cartésien, qu'on est-t-il pour plusieurs points?

4.4 Génération de trajectoire.

En se basant sur la définition des robots donnée précédemment au paragraphe (4.1), à savoir que l'organe terminal du robot accomplit un certain mouvement dans l'espace établi au préalable par l'opérateur. Le mouvement ainsi réalisé est appelé trajectoire. En effet, à chaque point de coordonnées cartésiennes, on détermine les coordonnées généralisées correspondantes. Suivant la trajectoire décrite par l'élément terminal, on peut trouver une relation entre les coordonnées généralisées (q_i) et les paramètres introduits pour décrire cette même trajectoire, ceci est illustré sur la figure (MOD.5).

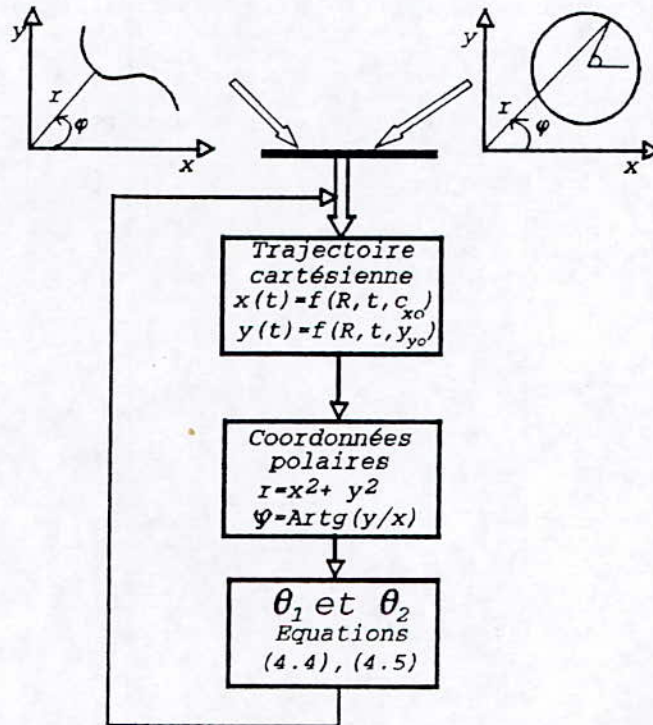


Figure (MOD.5) Schéma récapitulatif relatif au modèle géométrique directe et inverse.

La trajectoire désirée pour que le robot l'exécute est un cercle dans le plan $(0, x, y)$ situé à une distance (C_{x_0}, C_{y_0}) du repère R_0 du robot et de rayon R voire figure (MOD.6)

Les équations du cercle dans le repère $(0, x, y)$ sont:

$$\begin{cases} x = C_{x_0} + R \cos \omega t \\ y = C_{y_0} + R \sin \omega t \end{cases} \quad \text{avec} \quad \begin{cases} C_{x_0} = (l_1 + l_2 - R) \cos\left(\frac{\pi}{4}\right) \\ C_{y_0} = (l_1 + l_2 - R) \sin\left(\frac{\pi}{4}\right) \end{cases}$$

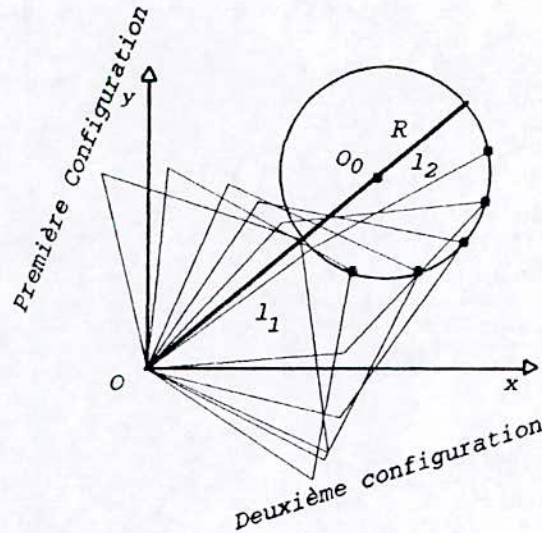


Figure (MOD.6) Evolution de l'élément terminal pour une trajectoire cercle

$$\rightarrow \begin{cases} C_{x_0} = (2l-R) \cos\left(\frac{\pi}{4}\right) \\ C_{y_0} = (2l-R) \cos\left(\frac{\pi}{4}\right) \end{cases}$$

A partir des coordonnées généralisées établies pour un point quelconque dans l'espace, on détermine leurs évolutions en fonction du temps pour une trajectoire cercle, ce qui revient à calculer θ_1 et θ_2 pour chaque point du cercle.

Etant donné que nous avons θ_1 et θ_2 , équations (4.4)(4.5), proposons-nous de déterminer les vitesses et accélérations de la trajectoire cercle à partir des coordonnées généralisées du modèle géométrique inverse.

$$\begin{cases} \theta_2 = q_{d_2} = \arccos\left[\frac{1}{2}\left(\frac{r}{l}\right)^2 - 1\right] \\ \theta_1 = q_{d_1} = \varphi - \arcsin\left[\frac{l}{r} \sin(\pi - q_{d_2})\right] \end{cases} \quad (4.8)$$

où l'indice d veut dire désirer.

Sachant que $\begin{cases} \dot{x} = -R\omega \sin\omega t \\ \dot{y} = R\omega \cos\omega t \end{cases}$

on détermine après quelques calculs de dérivé simple et double les vitesses et accélérations généralisées \dot{q}_{d_2} et \ddot{q}_{d_2}

avec: $\begin{cases} q_{d_2} = \arccos(\mu) \\ \mu = \frac{1}{2}\left(\frac{l}{r}\right)^2 - 1 \\ \dot{\mu} = \frac{r\dot{r}}{l^2} \end{cases}$

$$\begin{cases} \dot{q}_{d_2} = -\frac{\dot{\mu}}{\sqrt{1-\mu^2}} \\ \ddot{q}_{d_2} = -\frac{(\dot{r}^2 + \ddot{r}r)(1-\mu^2) + \mu\dot{\mu}r\dot{r}}{(1-\mu^2)\sqrt{1-\mu^2}} \end{cases} \quad (4.9)$$

La même méthode est appliquée pour θ_1 . on obtient alors:

$$\begin{cases} \dot{q}_{d_1} = \dot{\phi} - \dot{\phi}_1 \\ \ddot{q}_{d_1} = \ddot{\phi} - \ddot{\phi}_1 \end{cases} \quad (4.10)$$

avec:

$$\begin{cases} \phi = \arctan\left(\frac{x}{y}\right) & \phi_1 = \arcsin\left[\frac{l}{r} \sin(\pi - q_{d_2})\right] \\ \dot{\phi} = \frac{(\dot{y}x - \dot{x}y)r - 2\dot{r}(\dot{y}x - \dot{x}y)}{r^3} \\ \ddot{\phi}_1 = \frac{\ddot{\theta}\sqrt{1-\theta} + \dot{\theta}\phi_1\theta}{1-\theta^2} \\ \theta = \frac{l}{r} \sin(\pi - q_{d_2}) \end{cases}$$

Remarque.

La génération de trajectoire pour les deux robot est la même. Pour le robot PUMA le cercle est dans le plan vertical tandis que pour le SCARA, sur le plan horizontal.

Le comportement dynamique du robot exige d'imposer des trajectoires réalisables. Le choix d'une trajectoire est reliée à l'évolution de la position, la vitesse et l'accélération. En effet, la continuité en position, vitesse et accélération offre au robot la possibilité de poursuivre cette trajectoire avec des commandes réalisables. La décomposition de la tâche en plusieurs points intermédiaires, nécessite une continuité du premier et second type [47].

La connaissance des conditions aux limites de différent type, permet les calculs directes des paramètres des polynômes d'interpolation, figure (MOD.7).

L'imposition d'une consigne constante, nécessite l'utilisation d'un polynôme du troisième degré permettant une continuité en position et en vitesse. Cette tâche se traduit par le passage d'un état d'équilibre à un autre état d'équilibre [47].

Le polynôme choisi est:

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

les condition aux limite sont:

$$\begin{cases} P(0) = \theta_0 ; & \dot{P}(t) = 0 \\ P(t_f) = \theta_f ; & \dot{P}(t_f) = 0 \end{cases}$$

à partir de ces conditions on calcul différent les paramètres a_0, a_1, a_2, a_3 [47]:

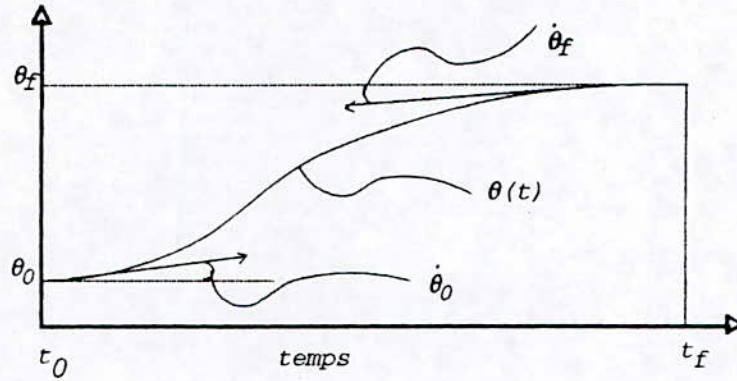


Figure (MOD.7) Génération d'un polynôme d'interpolation

$$\begin{cases} a_0 = \theta_0 \\ a_1 = \dot{\theta}_0 \\ a_2 = \frac{3}{t_f^3} (\theta_f - \theta_0) - \frac{2}{t_f} \dot{\theta}_0 - \frac{1}{t_f} \dot{\theta}_f \\ a_3 = -\frac{2}{t_f^3} (\theta_f - \theta_0) + \frac{1}{t_f^2} (\dot{\theta}_f + \dot{\theta}_0) \end{cases}$$

4.5 Modélisation dynamique des robots [43][48].

Après avoir défini la modélisation cinématique du robot et la spécification de ses tâches, on s'intéresse dans cette partie, à l'élaboration du modèle dynamique des robots, 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 à partir des lois de la mécanique Newtonienne et Lagrangienne. Les formalismes d'Euler-Lagrange et Nexton-Euler, nous permettent d'aboutir aux équations du mouvement des robots.

4.5.1 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 directe 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: [43]

$$\frac{d}{dt} \left(\frac{\delta L}{\delta \dot{q}_i} \right) - \left(\frac{\delta L}{\delta q_i} \right) + \left(\frac{\delta E_D}{\delta \dot{q}_i} \right) = \tau_i \quad i=1,2 \quad (4.11)$$

où L désigne le Lagrangien qui est défini par:

$$L = E_c - E_p = K - P \quad (4.12)$$

tels que:

- E_c : Energie cinétique.
- E_p : Energie potentielle.
- E_D : Energie de dissipation.
- τ_i : Force généralisée.

Pour la calcul de l'énergie cinétique, l'expression de la vitesse est:

$${}^0V^i = \frac{d}{dt} ({}^0R^i) = \frac{d}{dt} ({}^0T^i \quad {}^iR^i) \quad (4.13)$$

${}^iR^i$: coordonnées du point i dans le repère R_i .

D'où

$${}^0V^i = \sum_{j=1}^i [U_{ij} \dot{q}_j] \quad {}^iR^i \quad (4.14)$$

tel que:

$$U_{ij} = \begin{cases} {}^0T^{j-1} Q_j & j-1 < i \\ 0 & j \geq i \end{cases} \quad \text{et} \quad Q_j = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

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

$$dK_i = \frac{1}{2} \text{tr}(V_i V_i^T) dm = \frac{1}{2} \text{tr} \left[\sum_{p=1}^i \sum_{r=1}^i U_{ip} [{}^iR^i dm \quad {}^iR^{iT}] U_{ir}^T \dot{q}_p \dot{q}_r \right] \quad (4.15)$$

Donc l'énergie cinétique de la liaison i est:

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

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:

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

I_l :moment d'inertie de l'actionneur (toutes les articulation sontrotatives).

$$K = \sum_{i=1}^n K_i + E_{c_s} \quad (4.16)$$

L'énergie potentielle est:

$$P = \sum_{i=1}^n -m_i g^T {}_0R^i = \sum_{i=1}^n -m_i g^T ({}_0T^i {}_iR^i) \quad (4.17)$$

avec:

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

Et l'énergie de dissipation est:

$$E_D = \frac{1}{2} \sum_{i=1}^n b_i \dot{q}_i^2$$

où b_i : coefficient de frottement visqueux.

En déduit alors les forces généralisées: [43] [48]

$$\begin{aligned} \tau_i = & \sum_{k=i}^n \sum_{p=1}^k tR[U_{kp} J_k U_{ki}^T] \ddot{q}_p - \sum_{k=i}^n m_k g^T U_{ki} {}_kR^k \\ & + \sum_{k=i}^n \sum_{p=1}^k \sum_{r=1}^k tR[U_{kpr} J_k U_{ki}^T] \dot{q}_r \dot{q}_p + I_i \ddot{q}_i + b_i \dot{q}_i; \quad i=1, n \end{aligned} \quad (4.18)$$

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

Dans le cas du bras manipulateur PUMA et SCARA, les calculs en utilisant les équations (4.18) sont long (voir annexe 2), on préfère calculer d'abord le Lagrangien ensuite procédé par des dérivations en utilisant (4.11).

sachant que [43]:

$$U_{ij} = \begin{cases} {}_0T^{j-1} Q_j {}_{j-1}T^i & j \leq i \\ 0 & j > i \end{cases} \quad (4.19)$$

on a alors

$$\begin{cases} U_{11} = {}_0T^0 Q_1 {}_0T^1 \\ U_{22} = {}_0T^1 Q_2 {}_1T^2 \end{cases} \quad \begin{cases} U_{12} = 0 \\ U_{21} = {}_0T^1 Q_1 {}_0T^2 \end{cases} \quad (4.20)$$

L'expression des U_{ij} sont exposées en annexe 2.

On a:

$$\begin{cases} L = \frac{1}{2} \sum_{i=1}^2 \sum_{j=1}^i \sum_{k=1}^i \text{tr}(U_{ij} J_j U_{ik}^T) \dot{q}_j \dot{q}_k + \sum_{i=1}^2 m_i g ({}^0 T^i {}_i \bar{r}^i) \\ \tau_i = \frac{d}{dt} \left(\frac{\delta L}{\delta \dot{q}_i} \right) - \left(\frac{\delta L}{\delta q_i} \right) \end{cases} \quad (4.21)$$

donc :

$$\begin{cases} \text{pour } i=1 \\ L_1 = \frac{1}{2} \sum_{j=1}^1 \text{tr}(U_{1j} J_j U_{11}^T) \dot{q}_j \dot{q}_1 \\ \text{pour } i=2 \\ L_2 = \frac{1}{2} \sum_{j=1}^2 \text{tr}(U_{2j} J_j U_{21}^T) \dot{q}_j \dot{q}_1 + \text{tr}(U_{2j} J_j U_{22}^T) \dot{q}_j \dot{q}_2 \end{cases}$$

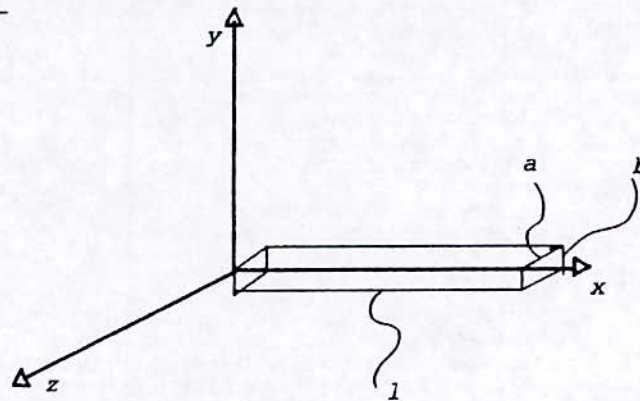
Le 2^{eme} terme du lagrangien (energie potentielle) devient :

$$\sum_{i=1}^2 m_i g {}^0 T^i {}_i \bar{r}^i = m_1 g ({}^0 T^1 {}_1 \bar{r}^1) + m_2 g ({}^0 T^2 {}_2 \bar{r}^2)$$

la matrice d'inertie devient

$$J_1 = \begin{bmatrix} \frac{1}{3} m_1 l^2 & 0 & 0 & -\frac{1}{2} m_1 l \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2} m_1 l & 0 & 0 & m_1 \end{bmatrix} ;$$

tel que $\begin{cases} \int x^2 dm = \frac{ml^2}{3} \\ \int y^2 dm = \frac{mb^2}{3} \end{cases}$ suivant le schéma ci-dessous.



avec $b \ll l$ et $a \ll l$.

Sachant que :

$$\begin{cases} \bar{x}_1 = -\frac{l}{2} & \text{et } \bar{y}_1 = \bar{z}_1 = 0 \\ \bar{x}_2 = -\frac{l}{2} & \text{et } \bar{y}_2 = \bar{z}_2 = 0 \end{cases}$$

Et après quelques manipulations mathématiques, l'expression du Lagrangien devient:

$$\begin{aligned} L = & \left(\frac{1}{6} m_1 l^2 + \frac{2}{3} m_2 l^2 \right) \dot{q}_1^2 + \frac{1}{2} m_2 l^2 C_2 \dot{q}_1^2 + \frac{1}{3} m_2 l^2 \dot{q}_1 \dot{q}_2 \\ & + \frac{1}{2} m_2 l^2 C_2 \dot{q}_1 \dot{q}_2 + \frac{1}{6} m_2 l^2 \dot{q}_2^2 - (m_1 g \bar{x}_1 + m_1 g l) S_1 \\ & - m_2 g \bar{x}_2 S_{12} - m_2 g l (S_{12} + S_1) + \frac{1}{2} b_1 \dot{q}_1^2 + \frac{1}{2} b_2 \dot{q}_2^2 \end{aligned}$$

Le Lagrangien ainsi obtenu, on peut procéder par le calcul directe en utilisant (4.11), donc les forces généralisées sont:

$$\begin{cases} \tau_1 = \left(\frac{1}{3} m_1 l^2 + \frac{4}{3} m_2 l^2 + m_2 l^2 C_2 \right) \ddot{q}_1 + \frac{1}{3} m_2 l^2 + \frac{1}{2} m_2 l^2 C_2 \ddot{q}_2 \\ \quad - m_2 l^2 S_2 \dot{q}_1 \dot{q}_2 - \frac{1}{2} m_2 l^2 S_2 \dot{q}_2^2 + (m_1 g \frac{l}{2} + m_2 g l) C_1 \\ \quad \quad \quad \frac{1}{2} m_2 g l C_{12} + (\text{frottement}) \dot{q}_1 \\ \tau_2 = \left(\frac{1}{m_2} l^2 + \frac{1}{m_2} l^2 C_2 \ddot{q}_1 \right) + \frac{1}{3} m_2 l^2 \ddot{q}_2 + \frac{1}{2} m_2 l^2 S_2 \dot{q}_1^2 \\ \quad \quad \quad + \frac{1}{2} m_2 g l C_{12} + (\text{frottement}) \dot{q}_2 \end{cases}$$

Les commandes du robot sont obtenues en faisant: $\tau_1 = k_1 u_1$ et $\tau_2 = k_2 u_2$ avec k_1 et k_2 sont les coefficients de proportionnalités des actionneurs. (troisième hypothèse du paragraphe (4.1))

En ce qui concerne le robot SCARA, les calculs sont exactement les mêmes, seulement que l'effet gravitationnel n'existe pas. En Effet puisque le robot évolue dans un environnement horizontal donc il n'y a pas de variation d'énergie potentielle ($g=0$).

4.5.2 Résultats de simulation.

Pour voir l'évolution des deux robots en position, vitesse et accélération, il est nécessaire de simuler le modèle de connaissance établi par le formalisme de LE. La mise sous forme d'état des équations différentielles nonlinéaires trouvées nous permettent d'utiliser la méthode de Runge-Kutta (RK) d'ordre quatre sous les conditions initial suivantes:

Le vecteur d'état X s'écrit: $X^T = [x_1 \ x_2 \ x_3 \ x_4] = [q_1 \ \dot{q}_1 \ q_2 \ \dot{q}_2]$ tel que initialement on a: $X^T = [0 \ 0 \ 0 \ 0]$.

La mise sous forme matricielle du modèle des deux robots donne le système suivant:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} + \begin{bmatrix} t_{g1} \\ t_{g2} \end{bmatrix} + [b_1 \ b_2] \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

avec:

$$\begin{cases} \alpha = \frac{1}{3} l^2 (m_1 + 4m_2 + 3m_2 \cos(q_2)) \\ \beta = m_2 l^2 \left(\frac{1}{3} + \frac{1}{2} \cos(q_2) \right) \\ \gamma = \frac{1}{3} m_2 l^2 \end{cases} \begin{cases} f_1 = -m_2 l^2 \dot{q}_2 \sin(q_2) \left(\frac{1}{2} \dot{q}_2 + \dot{q}_1 \right) \\ f_2 = \frac{1}{2} m_2 \sin(q_2) l^2 \dot{q}_1^2 \\ g_1 = \frac{1}{2} m_1 g l \cos(q_1) + m_2 g l \left[\frac{1}{2} \cos(q_1 + q_2) + \cos(q_1) \right] \\ g_2 = \frac{1}{2} m_2 g l \cos(q_1 + q_2) \end{cases}$$

et b_1 et b_2 sont les termes de frottement visqueux. On détermine alors la forme d'état suivante:

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{\left[\frac{(k_1 u_1 - f_1 - t_{g1} - b_1 \dot{q}_1)}{\alpha} + (-k_2 u_2 + f_2 + t_{g2} + b_2 \dot{q}_2) \frac{\beta}{\alpha \gamma} \right]}{\left[1 - \frac{\beta}{\alpha \gamma} \beta \right]} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{\left[\frac{(k_2 u_2 - f_2 - t_{g2} - b_2 \dot{q}_2)}{\gamma} + (-k_1 u_1 + f_1 + t_{g1} + b_1 \dot{q}_1) \frac{\beta}{\alpha \gamma} \right]}{\left[1 - \frac{\beta}{\alpha \gamma} \beta \right]} \end{array} \right. \quad (4.22)$$

On s'est intéressé à la simulation de deux types de robot à deux degrés de liberté présentant deux articulations rotative. Le premier type PUMA met en évidence l'effet de la gravitation agissant sur différentes liaisons. Tandis que le second type SCARA se déplace dans un espace planaire horizontal négligeant ainsi cette effet. Les valeurs numériques des différents paramètres des deux robots sont:

pour le bras manipulateur PUMA [45] $\left\{ \begin{array}{l} l_1 = l_2 = l = 1m; \ m_1 = m_2 = 10kg \\ b_1 = 75 \text{ N/rd/s}; \ b_2 = 10 \text{ N/rd/s} \\ k_1 = 40 \text{ Nm/v}; \ k_2 = 20 \text{ Nm/v} \\ g = 9.81 \text{ ms}^{-2} \end{array} \right.$

pour le bras manipulateur SCARA [44] $\left\{ \begin{array}{l} l_1 = l_2 = l = 0.432m; \\ m_1 = 15.91 \text{ kg}; \ m_2 = 11.36 \text{ kg} \\ b_1 = b_2 = 7 \cdot 10^{-4} \text{ N/rd/s} \\ k_1 = k_2 = 0.18 \text{ Nm/v}; \end{array} \right.$

avec:

b_1, b_2 : frottements visqueux des actionneurs.

k_1, k_2 : constantes du couple moteur.

l_1, l_2 : longueurs des deux liaisons.

m_1, m_2 : masses des deux liaisons.

La figure (MOD 4.8) montre la disposition des actionneurs qui sont disposés de telle sorte à négliger leurs poids dans l'élaboration du modèle de LE. La transmission du mouvement est assurés par des courroies (pour le robot PUMA).

Les réponses des deux robots en position, vitesse et accélération pour une entrée échelon, sur un horizon de temps suffisant, sont consignés sur les figures (4.1)(4.2).

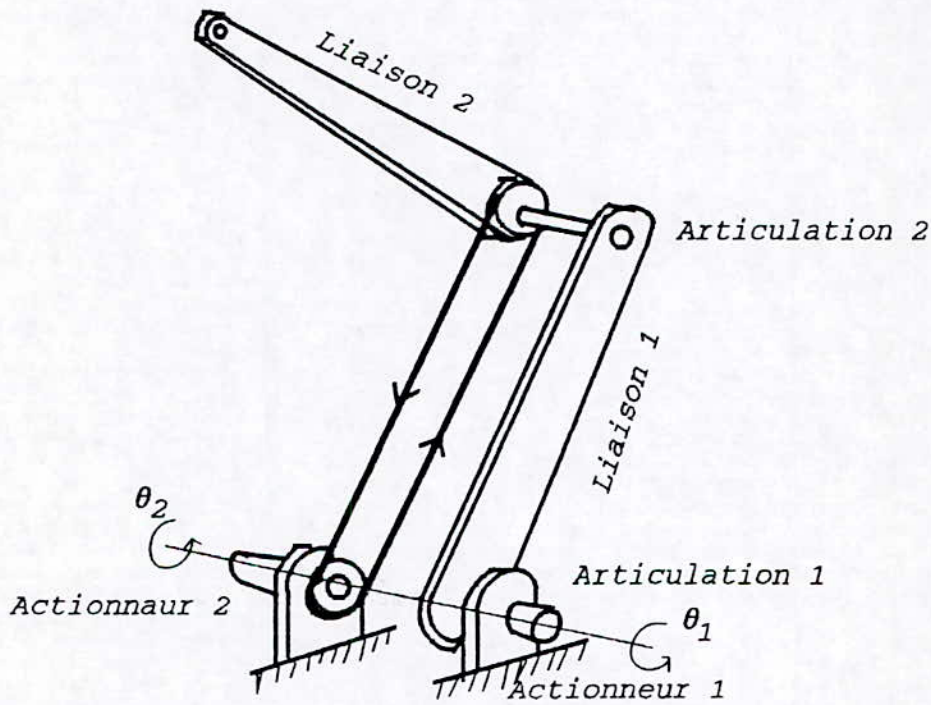


Figure (MOD.8) Disposition des actionneurs.

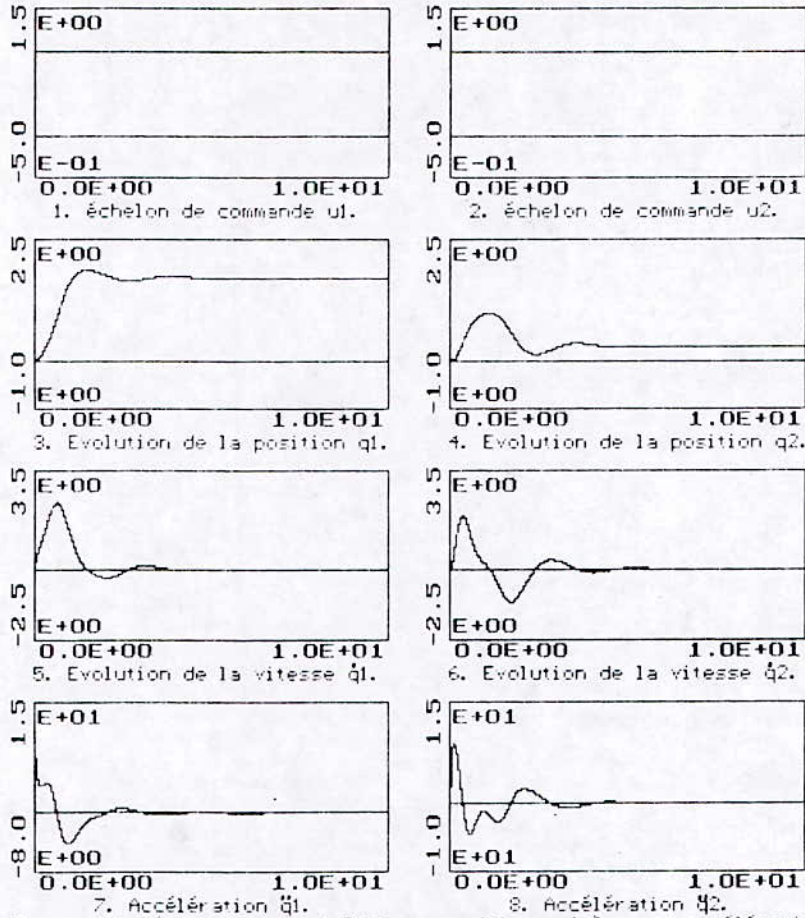


figure 4.1 Réponse du robot PUMA en position, vitesse et accélération.

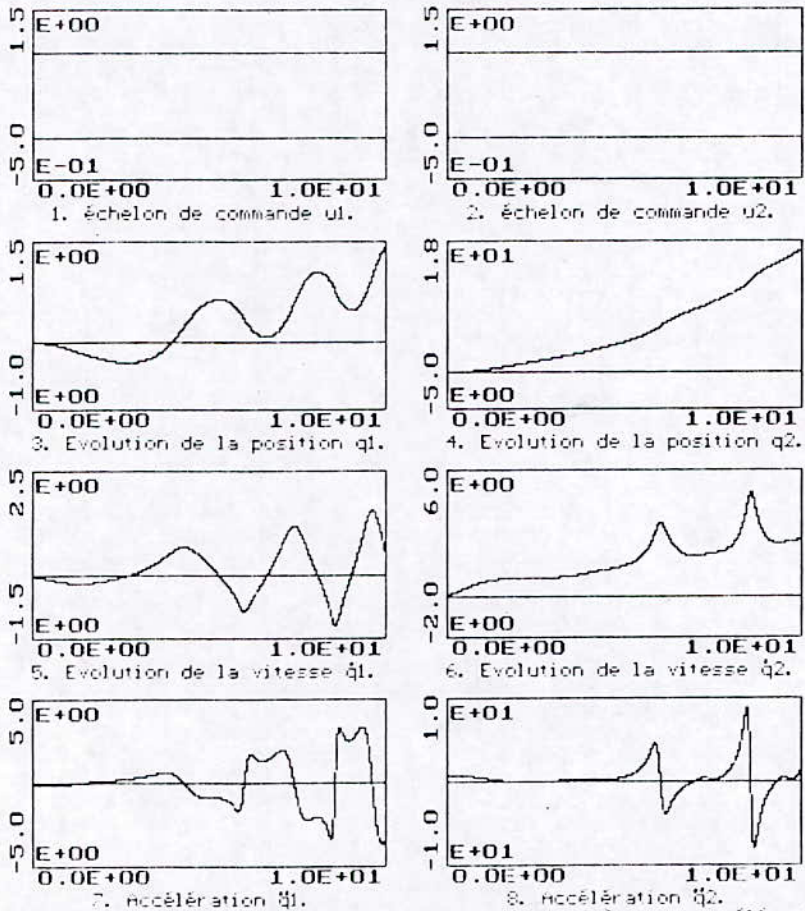


figure 4.2 Réponse du robot SCARA en position, vitesse et accélération.

4.5.3 Formalisme de Newton-Euler.

Dans la partie précédente nous avons élaboré un ensemble d'équations différentielles nonlinéaires du second ordre à partir du formalisme de Lagrange qui décrivent le comportement dynamique des robots. L'utilisation du même formalisme pour le calcul des forces généralisées à partir de la position, de la vitesse et de l'accélération d'une certaine trajectoire, exige un temps de calcul en temps réel excessive pour la commande en boucle ouverte. Le problème est dû à l'utilisation des matrices de transformation de DH, 4x4.

Pour réduire ce temps de calcul, la formulation de Newton-Euler permet d'utiliser un algorithme récursif basé sur le principe de D'Alambert. L'algorithme est composé de deux étapes [43].

1^{ere} étape: "FORWARD EQUATIONS" pour $i=1,2$.

Dans cette étape on calcule les vitesses et les accélérations linéaires v_i, \dot{v}_i et angulaires $\omega_i, \dot{\omega}_i$ de chaque liaison, exprimées dans le repère lié à la liaison considérée, commençant de la base jusqu'à l'élément terminal.

Remarque

Dans la suite du raisonnement on ne représentera que les équations relatives aux rotations (les deux robots SCARA et PUMA présentent des liaisons ritatives).

$${}_iR^0\omega_i = {}_iR^{i-1}({}_{i-1}R^0\omega_{i-1}) \tag{4.23}$$

$${}_iR^0\dot{\omega}_i = {}_iR^{i-1}({}_{i-1}R^0\dot{\omega}_{i-1}) \tag{4.24}$$

$$\begin{aligned} {}_iR^0\dot{v}_i &= {}_iR^{i-1}(Z_0\ddot{q}_i + {}_{i-1}R^0\dot{v}_{i-1}) + ({}_iR^0\dot{\omega}_i) \wedge ({}_iR^0P_i^*) \\ &\quad + 2({}_iR^0\omega_i) \wedge ({}_iR^{i-1}Z_0\dot{q}_i) \\ &\quad + ({}_iR^0\dot{\omega}_i) \wedge [({}_iR^0\omega_i) \wedge ({}_iR^0P_i^*)] \end{aligned} \tag{4.25}$$

$${}_iR^0\bar{a}_i = ({}_iR^0\dot{\omega}_i) \wedge ({}_iR^0\bar{S}_i) + ({}_iR^0\omega_i) \wedge [({}_iR^0\omega_i) \wedge ({}_iR^0\bar{S}_i)] + {}_iR^0\dot{v}_i \tag{4.25}$$

2^{eme} étape: "BACKWARD EQUATIONS" pour $i=2,1$.

$${}_iR^0f_i = {}_iR^{i+1}({}_{i+1}R^0f_{i+1}) + m_i {}_iR^0\bar{a}_i \tag{4.27}$$

$$\begin{aligned} {}_iR^0\eta_i &= {}_iR^{i+1}[{}_{i+1}R^0\eta_{i+1} + ({}_{i+1}R^0P_i^*) \wedge ({}_{i+1}R^0f_{i+1})] + ({}_iR^0P_i^* + {}_iR^0\bar{S}_i) \wedge ({}_iR^0f_i) \\ &\quad + ({}_iR^0I_i \text{ } {}_0R^i) ({}_iR^0\dot{\omega}_i) + ({}_iR^0\omega_i) \wedge [({}_iR^0I_i \text{ } {}_0R^i) ({}_iR^0\omega_i)] \end{aligned} \tag{4.28}$$

\wedge : produit vectoriel.

$$\tau_i = ({}_iR^0 f_i)^T ({}_iR^{i-1} Z_0) + b_i \dot{q}_i \quad (4.29)$$

avec b_i frottement visqueux de l'articulation i . Et

- m_i : masse total de la liaison i .
- \bar{S}_i : position du centre de masse de la liaison i par rapport au repère R_i .
- P_i^* : distance du repère R_i par rapport à R_{i-1} .
- V_i : vitesse linéaire du centre de masse de la liaison i .
- a_i : accélération linéaire du centre de masse de la liaison i .
- F_i : force totale exercée sur la liaison i au centre de gravité.
- I_i : matrice d'inertie de la liaison i du centre de masse par rapport à R_0 .
- f_i : force exercée sur la liaison i par la liaison $i-1$ (force déduite de la séparation des liaisons).
- η_i : moment exercé sur la liaison i par la liaison $i-1$ dans le repère R_{i-1} .

Avec les conditions initiales suivantes:

$$\omega_0 = \dot{\omega}_0 = V_0 = 0; \quad \dot{V}_0 = (g_x, g_y, g_z)^T \quad |g| = 9.81 \text{ ms}^{-2}, \quad z_0 = (0 \ 0 \ 1)^T.$$

Soit ${}_{i-1}R^i$ la transformation 3x3 qui exprime le passage du repère R_i à un repère R_{i-1} c'est une matrice extraite de la matrice (4.1)

$${}_{i-1}R^i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \quad (4.30)$$

$$\text{avec: } ({}_{i-1}R^i)^{-1} = {}_iR^{i-1} = ({}_{i-1}R^i)^T$$

L'intérêt de cette matrice de transformation est qu'elle nous évite le calcul de $\omega_i, \dot{\omega}_i, \dot{V}_i, a_i, P_i^*, \bar{S}_i, f_i, \eta_i$ et τ_i dans le repère R_0 , cependant on calcul ${}_i T^0 \omega_i, {}_i T^0 \dot{\omega}_i, \dots, {}_i T^0 \tau_i$ qui sont référenciés au repère lié à la liaison.

L'application de l'algorithme exige de définir les matrices et les vecteurs suivants:

$${}_0R^1 = \begin{bmatrix} C_1 & -S_1 & 0 \\ S_1 & C_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad {}_1R^2 = \begin{bmatrix} C_2 & -S_2 & 0 \\ S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad {}_0R^2 = \begin{bmatrix} C_{12} & -S_{12} & 0 \\ S_{12} & C_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix};$$

$${}_1R^0 P_1^* = [1 \ 0 \ 0]^T; \quad {}_2R^0 P_2^* = [1 \ 0 \ 0]^T; \quad {}_1R^0 \bar{S}_1 = \left[-\frac{l}{2} \ 0 \ 0\right]^T; \quad {}_2R^0 \bar{S}_2 = \left[-\frac{l}{2} \ 0 \ 0\right]^T; \\ {}_2R^0 P_1^* = {}_2R^1 {}_1R^0 P_1^* = [l C_1 \ -l S_2 \ 0]^T;$$

$${}_iR^0 I_i \quad {}_0R^i = \begin{bmatrix} I_{xx_i} & 0 & 0 \\ 0 & I_{yy_i} & 0 \\ 0 & 0 & I_{zz_i} \end{bmatrix}$$

$$\text{avec } I_{xx} = \frac{m(a^2 + b^2)}{12}; \quad I_{yy} = \frac{ml^2}{12}; \quad I_{zz} = \frac{ml^2}{2}$$

les produits d'inerties sont nuls car le repère R_i est parallèle aux axes principaux de la liaison et d'origine le centre de gravité de cette liaison.

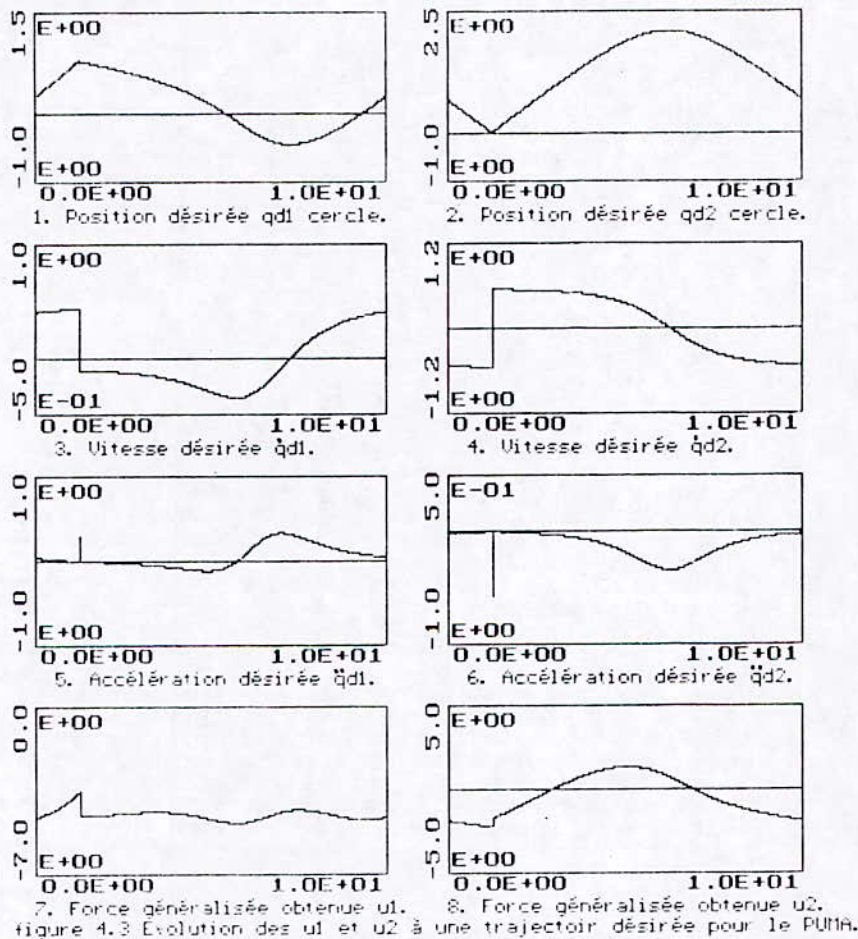
4.5.4 Résultats de simulation.

L'algorithme de Newton-Euler permet le calcul des forces généralisées pour une consigne circulaire dans l'espace et polynomiale de troisième ordre dans le temps. Pour le robot PUMA on applique la consigne circulaire tandis que la consigne polynomiale est appliqué pour le robot SCARA.

La figure (4.3) montre l'évolution des positions, des vitesses, accélérations et des forces généralisées dans le cas du robot PUMA et la figure (4.4) pour le robot SCARA.

Remarque.

la simulation a été réalisée avec un couple et une force appliqués à l'élément terminal nuls.



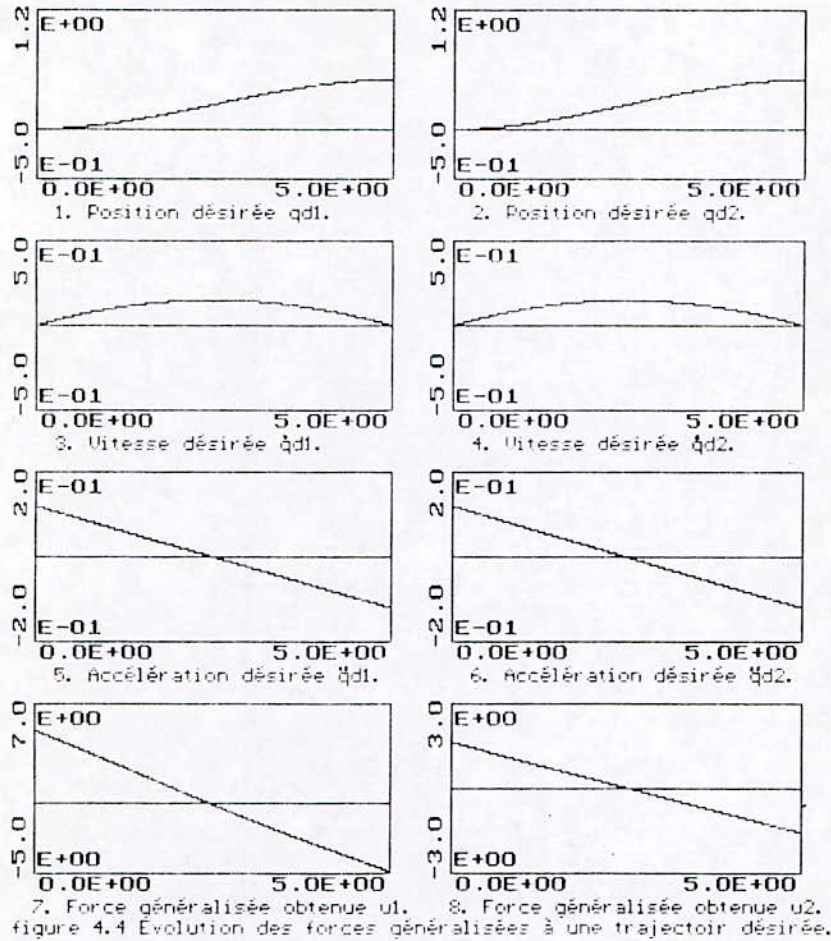


figure 4.4 Evolution des forces généralisées à une trajectoire désirée.

4.6 Validation des deux modèles. [49]

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 différentes de celles utilisées pour le modèle initial.

Dans notre cas, on s'est intéressé à modéliser les robots, en utilisant deux formalismes, celui du principe du moindre action (LE) et celui du principe de D'Alembert (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.

Donc, le but cette partie est de confirmer que les deux modèle, obtenus par les deux formalismes, sont identiques.

D'après la figure présentée, les deux modèle se valident si les erreurs définies par:

$$e_{1,2} = q_{d_{1,2}} - q_{1,2}; \quad \dot{e}_{1,2} = \dot{q}_{d_{1,2}} - \dot{q}_{1,2}; \quad \ddot{e}_{1,2} = \ddot{q}_{d_{1,2}} - \ddot{q}_{1,2} \text{ soient très faibles.}$$

Les figures (4.5)(4.6)(4.7)(4.8) montrent les vitesses, les positions et les

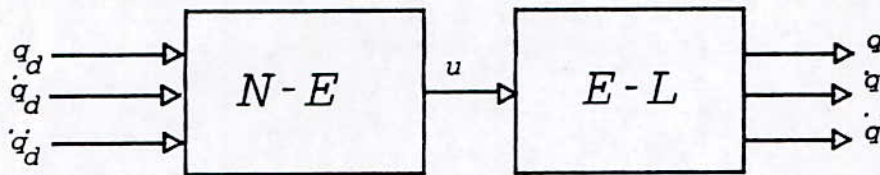


Figure I.11 Couplages des deux modèles NE-EL.

accélérations pour chaque trajectoire désirée et celles générées par le modèle dynamique d'EL. On présente alors à côté de chaque caractéristique cinématique leurs erreurs.

Remarque.

Toute les simulations on été réalisées avec un pas de calcul de $dt=10^{-3}s$.

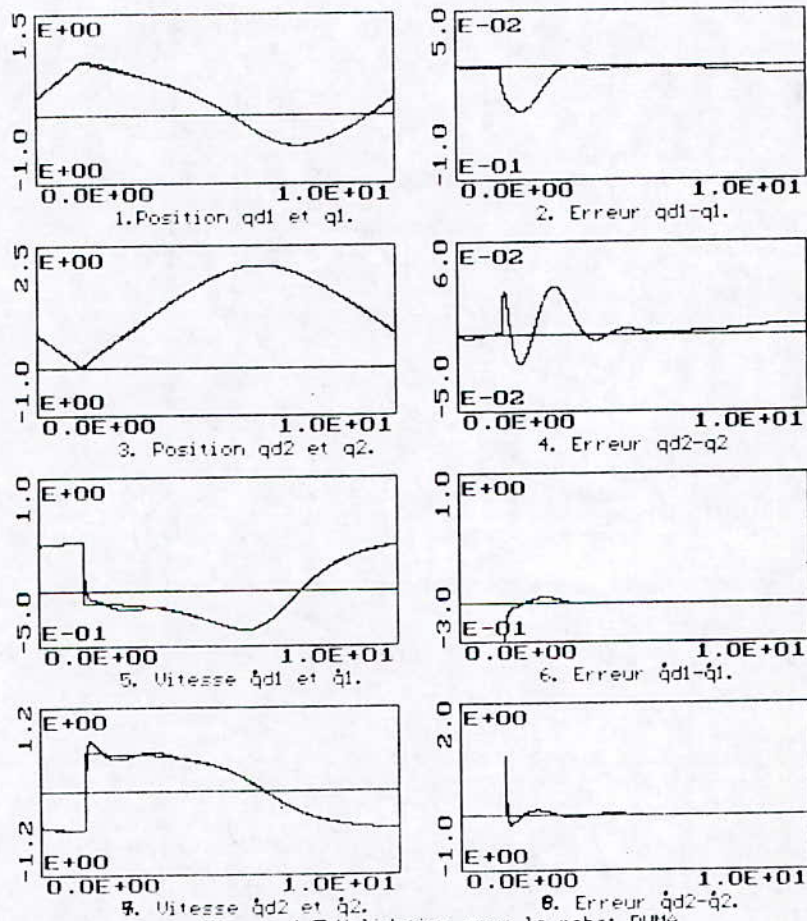


Figure 4.5 Validation pour le robot PUMA

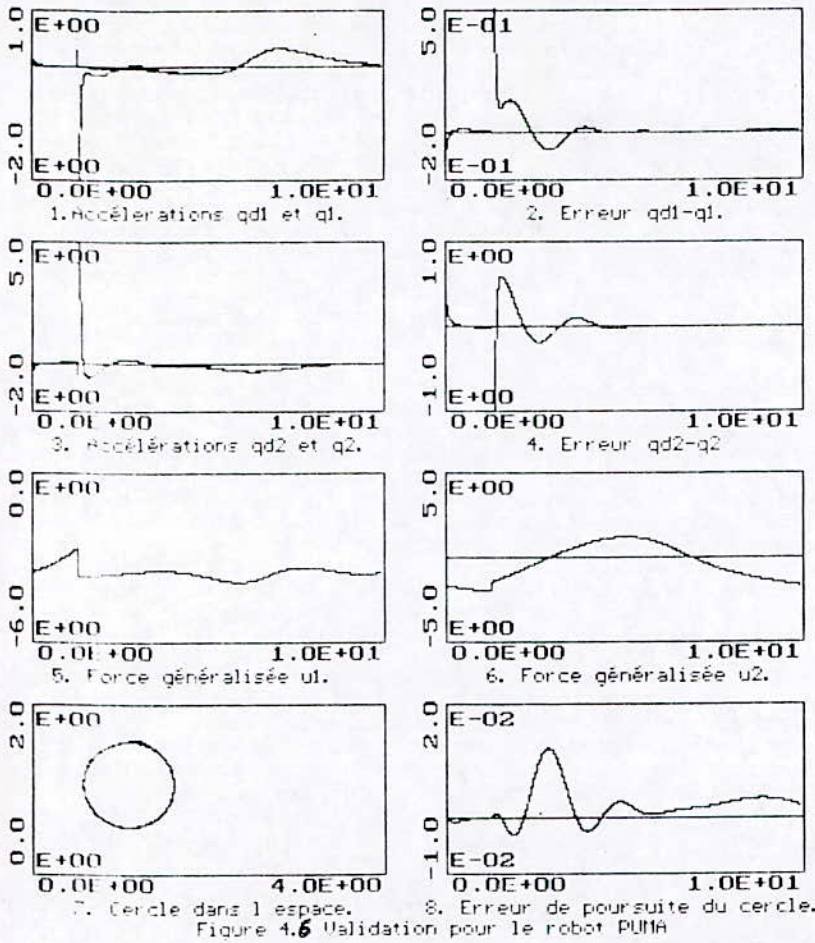


Figure 4.6 Validation pour le robot PUMA

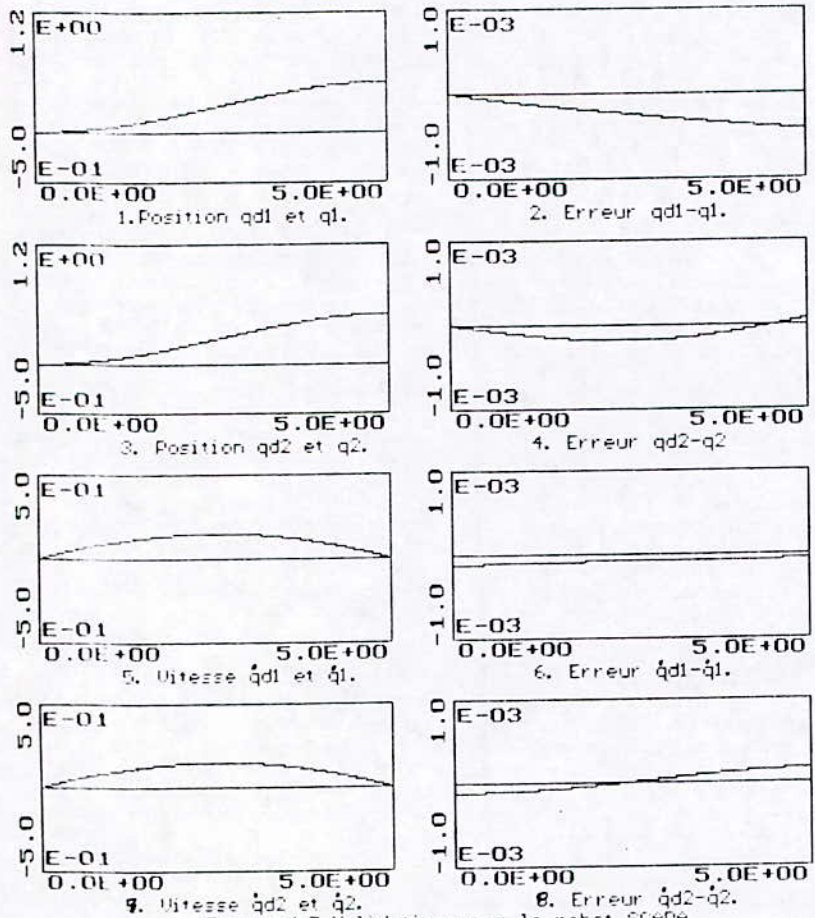


Figure 4.7 Validation pour le robot SCARA

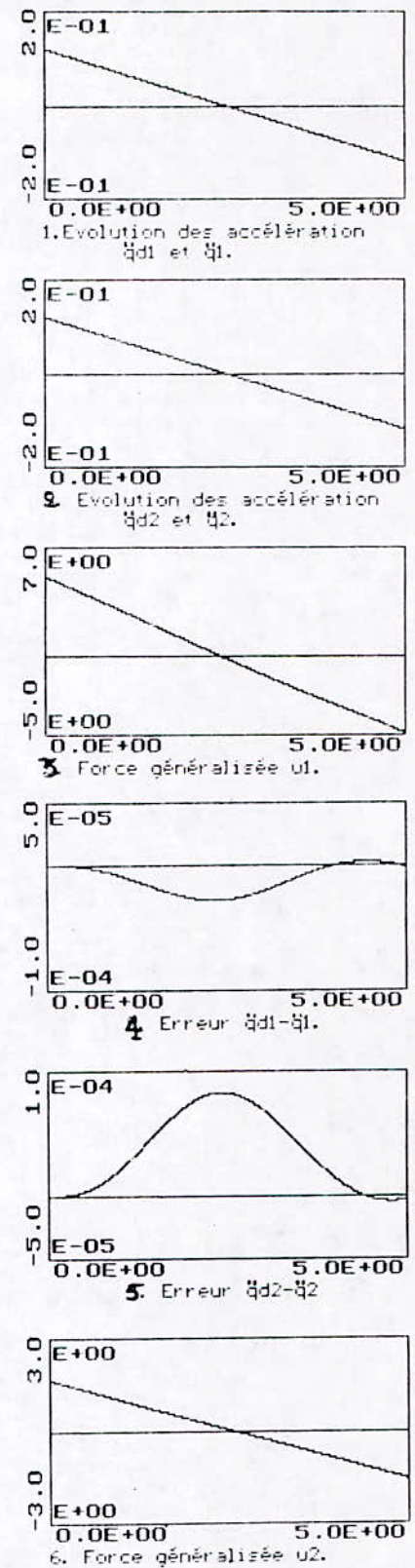


Figure 4.8 Validation pour le robot SCARA

Conclusion.

Les formalismes de Newton-Euler et Lagrange-Euler permettent d'établir un système d'équations différentielles reliant les coordonnées généralisées aux forces généralisées, obtenant ainsi le modèle dynamique du robot.

Le modèle cinématique inverse est un modèle spécifiant la relation entre les coordonnées cartésiennes dans l'espace opérationnel et les coordonnées généralisés.

Le comportement dynamique du robot exige une trajectoire spécifique (continue en position et en vitesse), permettant ainsi le contrôle du robot avec des commandes réalisables.

Le modèle dynamique obtenu à partir du formalisme de Lagrange-Euler couplé avec celui de Newton-Euler aboutit à une commande en boucle ouverte. Ce couplage permet une validation du calcul du modèle dynamique du robot. La continuité de la trajectoire désiré dans l'espace opérationnel (cercle), n'implique pas une continuité des trajectoires des coordonnées généralisées, ce qui est illustré par notre calcul dans le cas d'une trajectoire cercle. Pour avoir une continuité des trajectoires désirées généralisées, on procède à un lissage. Ce lissage peut être fait de différentes manières. Une qui consiste à assurer une continuité de la position et de la vitesse, entre deux points considérés. Une autre, qui nécessite une continuité en position , vitesse et accélération suivant l'application considérée.

Chapitre 5

APPLICATION DE LA GPC AUX BRAS MANIPULATEURS PUMA ET SCARA

*"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 5

APPLICATION DE LA GPC AUX BRAS MANIPULATEURS PUMA ET SCARA

Introduction.

Plusieurs chercheurs se sont penchés sur la commande adaptative des systèmes monovariables (SISO: Single input Single output). Ce type de commande a connu un essor considérable tout au long de ces dernières années dont les premières investigations ont été dédiées aux schémas de commandes adaptatives directes. Des résultats théoriques fort intéressants sur la stabilité, la convergence et la robustesse de ces algorithmes, sont disponibles dans la littérature. C'est grâce à l'article de base d'Åstrom et Wittenmark [50] que toutes les méthodes de synthèse associées reposent, telles que ceux qui se basent sur le principe de variance minimale [7][8], placement de pôles [51][52], commande prédictive étendue [2][10] et d'autres. Le développement de ces méthodes a tenté d'autres chercheurs tels que Borisson [53], Koivo[54], Dion [55][56] pour étendre les méthodes citées au cas multivariables (MIMO: Multinput multioutput).

Des versions de la GPC MIMO ont été mises au point par plusieurs auteurs. Toutes les méthodes développées se différencient les unes des autres par le type de modèle de représentation utilisé, par le domaine d'étude fréquentiel ou temporel [57] et par l'application industrielle (robotique).

La version continue de la GPC (CGPC: Continuous GPC) a été développées par Demircioğlu et Gaawthoop [58]. Quelques années après la version discrète de Clarke est apparue [2]. Une étude comparative entre les deux versions a montré qu'elles ont les mêmes propriétés [10]. En 1992, les mêmes auteurs ont mis au point l'extension en multivariable de la CGPC [59], la méthode peut être utilisée dans les deux cas adaptative et nonadaptative appliquées à des systèmes de dimension $p \times m$.

M'Saad et ces collaborateurs ont développés une technique de la commande prédictive généralisée adaptative multivariable qui impose une reparamétrisation du modèle à commander. Cette reparamétrisation peut s'effectuer par l'adjonction d'un modèle de référence basé sur l'état partiel. Le concept de modèle^{de} référence sur l'entrée et la sortie repose sur une interprétation adéquate du principe de placement de pôles. Il permet, plus particulièrement d'incorporer une capacité de poursuite convenable dans la synthèse des lois de commande basées sur la minimisation d'un critère quadratique, comme la commande prédictive généralisée [60][61][62]. L'utilisation de la GPC comme algorithme de commande dans cette approche a été motivée par l'utilisation d'un modèle CARIMA pour éliminer l'écart permanent, d'un prédicteur multipas optimale de l'erreur de poursuite et sa robustesse à commander plusieurs types de système.

La matrice retard ou interacteur des systèmes multivariables est une simple généralisation du terme retard associé au système SISO. La méconnaissance ou la variation du retard ou de l'intégrateur dans le cas de la commande à GMV détériore les performances du système [56]. Une solution à ce problème est donnée par Shah [63]

où il utilise la technique de commande à horizon étendu (GPC MIMO) qui n'a pas besoin de la connaissance exacte de la matrice interacteur. Ceci grâce à l'utilisation d'un critère multipas et aux choix adéquats des paramètres de synthèse de la commande.

La version multivariable de la GPC que nous vous proposons est similaire à celle de la version présentée par Bouzouia [64]. Elle est basée sur un modèle de représentation (obtenu par identification) découplé. Ce type de modèle facilite l'extension de l'algorithme de commande GPC monovariable présenté dans les chapitres précédents, au cas multivariable. Ceci revient à considérer plusieurs boucles de commandes indépendantes, chacune ayant une entrée et une sortie indépendante des autres [1]. Dans le cas de commande indépendante, il est clair qu'aucun terme de couplage ne peut être pris en compte, sinon sous forme de perturbation. Ceci implique donc que la méthode de commande doit être capable de rejeter ces perturbations, tout en assurant la stabilité de la boucle fermée. Le comportement non linéaire du modèle obtenu dans le chapitre modélisation, nécessite l'utilisation d'un algorithme d'identification à trace constante qui permet d'obtenir à chaque pas de calcul un modèle linéaire du manipulateur, vu que le modèle utilisé par la commande est linéaire.

5.1 Modèle paramétrique des systèmes multivariable.

Le modèle obtenu dans le chapitre 4 est un modèle multivariable à deux entrées et deux sorties non linéaire et couplé. Il se présente sous la forme d'état suivante:

$$\begin{cases} \dot{x} = f(x, u, t) \\ y = h(x) \end{cases} \quad (5.1)$$

où

- x : vecteur d'état constitué de la position, de la vitesse de dimension 2.
- y : vecteur de sortie de dimension 2.
- u : vecteur d'entrée ou de commande de dimension 2.
- t : temps.

$$f^T(x, u, t) = [f_1 \dots f_m] \quad (m=2)$$

$$h^T(x) = [h_1(x) \dots h_m(t)] \quad (m=2)$$

La commande par ordinateur de ce type de système nécessite la connaissance du modèle de représentation discret, échantillonné à une période d'échantillonnage T_e , adéquatement choisie. Pour cela la discrétisation de l'équation (5.1) est nécessaire. [65]. Après des manipulations mathématiques aboutissant au modèle paramétrique discret, on obtient [65].

$$\begin{bmatrix} A_{11}(q^{-1}) & \dots & A_{1m}(q^{-1}) \\ \cdot & \dots & \cdot \\ \cdot & \dots & \cdot \\ \cdot & \dots & \cdot \\ A_{m1}(q^{-1}) & \dots & A_{mn}(q^{-1}) \end{bmatrix} \begin{bmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_m \end{bmatrix} = \begin{bmatrix} B_{11}(q^{-1}) & \dots & B_{1m}(q^{-1}) \\ \cdot & \dots & \cdot \\ \cdot & \dots & \cdot \\ \cdot & \dots & \cdot \\ B_{m1}(q^{-1}) & \dots & B_{mn}(q^{-1}) \end{bmatrix} \begin{bmatrix} u_1 \\ \cdot \\ \cdot \\ \cdot \\ u_m \end{bmatrix} \quad (5.2)$$

avec :

$$A_{ii}(q^{-1}) = 1 + \sum_{l=1}^{n_{ii}} a_{ii_l} q^{-l} \quad i = 1, \dots, m$$

$$A_{ij}(q^{-1}) = \sum_{l=1}^{n_{ij}} a_{ij_l} q^{-l} \quad \begin{cases} i = 1, \dots, m \\ j = 1, \dots, m \\ i \neq j \end{cases}$$

$$B_{ij}(q^{-1}) = q^{-d_{ij}} \sum_{l=0}^{m_{ij}} b_{ij_l} q^{-l} \quad \begin{cases} i = 1, \dots, m \\ j = 1, \dots, m \\ d_{ij} \geq 1 \end{cases}$$

Le modèle ainsi obtenu est le modèle multivariable discret linéaire du robot manipulateur spécifiant le comportement du robot autour d'une trajectoire. Ce modèle peut être interprété comme une linéarisation du modèle nonlinéaire autour de cette trajectoire.

5.2 Modèle de représentation.

L'identification des systèmes multivariables constitue actuellement un axe de recherche important dans le domaine de l'automatique. En effet, lorsque les systèmes sont multivariables il est nécessaire d'adopter des méthodes appropriées pour aboutir à un modèle de représentation. On s'intéresse donc à l'extension des méthodes récursives monomariables pour l'identification des systèmes multivariables, La figure (APP.1) illustre ce concept.

Le modèle de représentation choisi est le modèle Diagonal-Diagonal [66] en entrée et en sortie. Il se présente sous la forme suivante:

$$\begin{bmatrix} A_1(q^{-1}) & & 0 \\ & \ddots & \\ & & A_m(q^{-1}) \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} B_1(q^{-1}) & & 0 \\ & \ddots & \\ 0 & & B_m(q^{-1}) \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \quad (5.3)$$

$$+ \begin{bmatrix} C_1(q^{-1}) & & 0 \\ & \ddots & \\ 0 & & C_m(q^{-1}) \end{bmatrix} \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_m \end{bmatrix} + \begin{bmatrix} h_1 \\ \vdots \\ h_m \end{bmatrix}$$

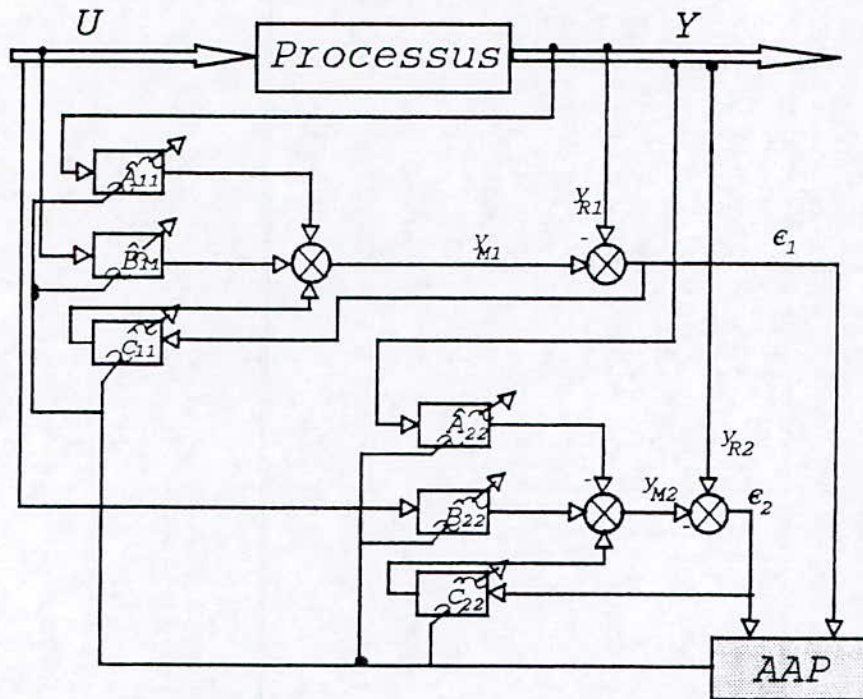


Figure (APP.1) Identification d'un système multivariable découplée.

Dans notre cas $m=2$.
avec:

$$A_i(q^{-1}) = 1 + \sum_{l=1}^{n_i} a_{i_l} q^{-l} \quad i = 1, \dots, m$$

$$B_i(q^{-1}) = q^{-d_i} \sum_{l=1}^{m_i} b_{i_l} q^{-l} \quad i = 1, \dots, m$$

ξ_i : bruit blanc de moyenne nulle, de variance constante.

\hat{n}_i, \hat{m}_i et \hat{d}_i sont estimés par une identification de structure ($d_i \geq 1$).

$h_i(t)$: paramètre représentant le couplage s'exprimant sous la forme:(eqt 5.2)

$$h_i(t) = - \sum_{\substack{j=1 \\ i \neq j}}^m A_{ij} y_j(t) + \sum_{\substack{j=1 \\ i \neq j}}^m B_{ij} u_j(t) \quad i=1 \dots m$$

Pour une sortie i choisie on a:

$$y_i(t) = - \sum_{l=1}^{n_i} a_{i_l} y_i(t-l) + \sum_{l=0}^{m_i} b_{i_l} u_i(t-d_i-l) + h_i(t) + \sum_{l=1}^{nc_i} c_{i_l} \xi(t-l) \quad (5.4)$$

Dans le cas de la recherche du modèle d'identification, l'équation (5.4) peut s'écrire sous forme vectorielle suivante:

$$\hat{y}_i(t) = \hat{\theta}_i^T \phi_i(t) \tag{5.5}$$

$$\text{où } \begin{cases} \hat{\theta}_i^T(t) = [\hat{a}_{i1} \dots \hat{a}_{i\hat{n}_i}, \hat{b}_{i0} \dots \hat{b}_{i\hat{m}_i}, \hat{c}_{i1} \dots \hat{c}_{i\hat{n}c_i}, h_i(t)] \\ \phi_i^T(t) = [-y_1(t-1) \dots -y_1(t-\hat{n}_i), u_1(t-\hat{d}_i) \dots u_1(t-\hat{d}_i-\hat{m}_i), \\ \epsilon_i(t) \dots \epsilon_i(t-\hat{n}c_i) 1] \end{cases}$$

et l'algorithme d'identification peut s'écrire de la manière suivante relatif aux équations (3.12) et (3.17).

$$\begin{cases} \hat{\theta}_i(t+1) = \hat{\theta}_i(t) + F_i(t) \hat{\theta}_i(t) \epsilon_i(t+1) \\ F_i(t+1) = \frac{1}{\lambda_{i1}} \left[F_i(t) - \frac{F_i(t) \phi_i(t) \phi_i^T(t) F_i(t)}{\left(\frac{\lambda_{i1}}{\lambda_{i2}} \right) + \phi_i^T(t) F_i(t) \phi_i(t)} \right] \\ \epsilon_i(t+1) = \frac{y_{iR}(t) - \hat{y}_i(t)}{\frac{\lambda_{i1}}{\lambda_{i2}} + \phi_i^T(t) F_i(t) \phi_i(t)} \end{cases} \tag{5.6}$$

tel que $tr[F_i(t+1)] = tr[F_i(t)] = tr[F_i(0)]$.

L'algorithme d'identification choisi est celui des MCR Etendus à trace constante à partir duquel on a pu représenter un système multivariable couplé par un autre totalement découplé. Ce couplage est absorbé par le paramètre h et la variation des paramètres identifiés, ainsi que le terme de perturbations aléatoires

5.3 Commande multivariable découplée.

La commande multivariable découplée est définie par opposition à la commande multivariable "complète". Cette dernière est basée sur un modèle de représentation tenant compte de tous les couplages entre les entrées $u_i(t)$ et les sorties $y_i(t)$. En supposant que le nombre d'entrée est égale au nombre de sorties et égal à m . L'expression du modèle est:

$$A(q^{-1})y_i(t) = B_i(q^{-1})u(t-d-1) + \omega_i(t) \tag{5.7}$$

où A et B sont des matrices polynomiales de dimension $(m \times m)$.

5.4 Algorithme à GPC multivariable.

Ce sont les termes non-diagonaux des matrices A et B qui représente l'effet des couplages. A partir de ce schéma, notre approche consiste à considérer que ces éléments non-diagonaux sont nuls [1] [64] [67] et à identifier seulement les éléments \hat{A}_i et \hat{B}_i . Ainsi le modèle de représentation peut être considéré comme plusieurs modèles monodimensionnels. La synthèse de la commande est alors menée sur chacun de ces modèles qui sont décrits par:

$$\hat{A}_i(q^{-1})y_i(t) = \hat{B}_i(q^{-1})u(t-d-1) + \omega_i(t) \tag{5.8}$$

avec: $\omega_i(t) = h_i(t) + \frac{c_i(q^{-1})}{\Delta(q^{-1})} \xi(t)$

où A_i, B_i et C_i sont décrit précédemment.

$\xi(t)$: bruit blanc de variance constante et de moyenne nul.

L'introduction de l'opérateur Δ , dans le modèle défini par (5.8) est introduit dans le but de faciliter la synthèse de la GPC

Par ce choix on est libre de donner des spécifications différentes pour chaque modèle i .

Considérons le critère de performance suivant:[68]

$$J = E \left[\sum_{j=N1}^{N2} (y(t+j) - W(t+j))^T P(j) (y(t+j) - W(t+j)) + \sum_{j=1}^{Nu} (\Delta u(t+j-1))^T Q(j) (\Delta u(t+j-1)) \right] \quad (5.9)$$

où:

$P(j)$: est une matrice diagonale de polynômes de pondération $P(J) = \text{diag}[P_i^2(J)]$.

$Q(j)$: est une matrice diagonale de pondération de la commande $Q(J) = \text{diag}[Q_i(J)]$.

$N1, N2, Nu$: sont les paramètres de synthèse de la GPC.

y, W : vecteur de sorties et de références.

Le coefficient de la commande est un intégrateur utilisé dans le but de garantir une erreur de régulation nulle pour une référence et une perturbation constante.

Après quelques manipulations mathématiques équivalentes à celles du premier chapitre on aboutit à l'équation algébrique suivante et ceci pour chaque sous-système (eqt 2.6)

$$P_i C_i = E_{j_i} A_i \Delta + q^{-j} F_{j_i} \quad i=1 \dots m \quad (m=2) \quad (5.10)$$

avec

j : intervalle de prédiction.

E_{j_i}, F_{j_i} : résultats de résolution de l'équation Diophantine (5.10) dont l'algorithme est présenté dans le premier et le deuxième chapitre.

$P_i(J) = P_i; \forall J$.

A partir de l'équation (5.10) on détermine le prédicteur optimal j -pas pour chaque sous système (eqt 2.8)

$$\Psi_i(t+j/t) = G_{j_i} \Delta u_i^f(t+j-1) + F_{j_i} y_i^f(t) \quad (5.11)$$

on aboutit alors, après avoir écrit l'équation du prédicteur sous forme matricielle à la minimisation du critère (5.9). La minimisation de ce dernier donne (eqt 2.19)

$$\Delta u_i = [G_i^T G_i + \lambda_j]^{-1} G_i^T (w_i - f_i) \quad (5.12)$$

où G_i, W , et f_i sont identiques à ceux définis dans le deuxième chapitre. avec: $Q_i(J) = \lambda_j; \forall J$.

L'algorithme d'identification estime directement les paramètres du régulateur pour chaque sous-système: c'est l'approche multivariable directe.

Le schéma de commande correspondant à la GPC multivariable (GPCM) directe et indirecte sont présentés respectivement sur les figures (APP.2) et (APP.3).

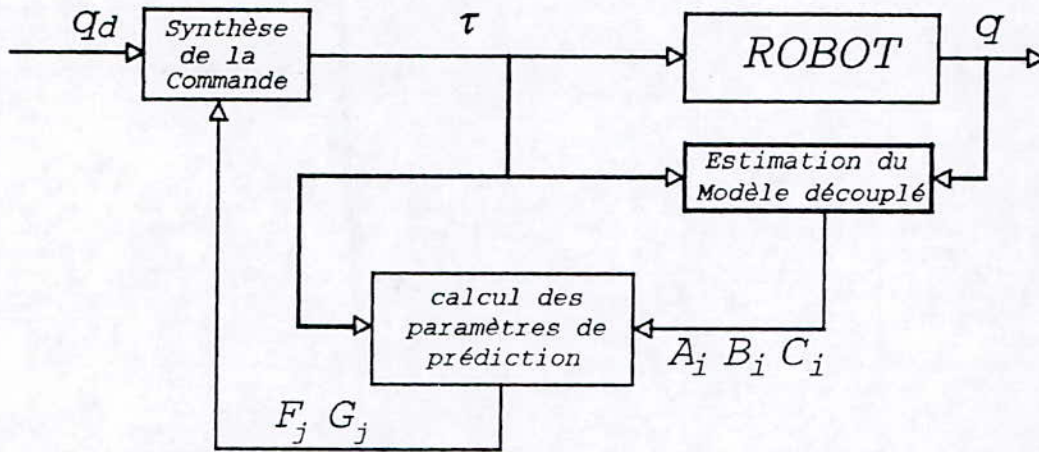


Figure (APP.2) Schéma bloc de la GPCM (Approche indirecte).

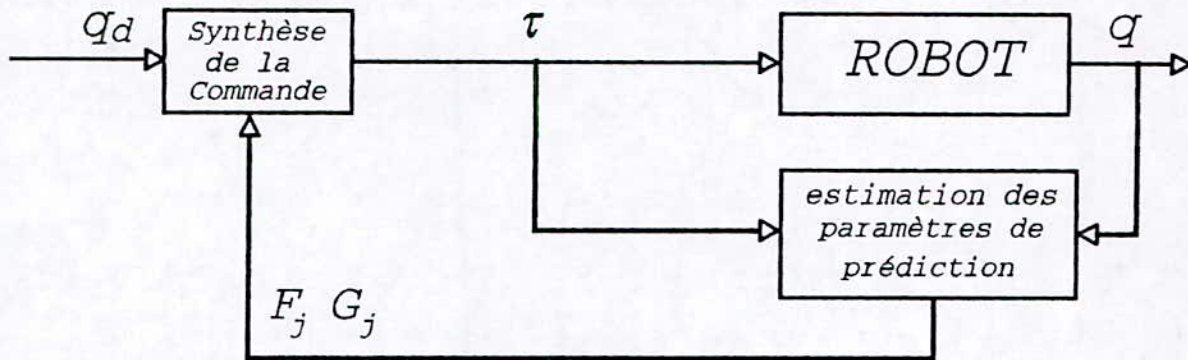


Figure (APP.3) Schéma bloc de la GPCM (Approche directe).

5.5 Avantage de la commande multivariable découplée.

La commande multivariable découplée présente les avantages suivants:

- Simplicité de mise en oeuvre. En effet, l'identification de m modèles monovariables, conduit à un nombre moins élevé de paramètres que l'identification d'un modèle multivariable complet.
- Chaque boucle de commande peut recevoir des spécifications différentes. Cette caractéristique s'avère essentielle dans le cas où les boucles ont des dynamiques différentes.
- Le modèle de représentation ainsi choisi permet une spécification, des paramètres de synthèse de la GPC pour chaque sous-système. Ceci présente un avantage de commande spécifique pour chaque ss avec des dynamiques désirées.

5.6 Résultats de simulation.

Après avoir étudié en détail le développement et les performances de la GPC, on se propose dans cette partie d'appliquer l'algorithme de base au modèles des deux robots PUMA et SCARA, présentés dans le quatrième chapitre.

Le modèle des deux robots ont été simulés avec un pas de calcul de $dt=10^{-3}s$, en utilisant la méthode de RK du quatrième ordre, et l'algorithme de commande adaptative a été appliqué avec une période d'échantillonnage de $T_e=10ms$.

La structure nonlinéaire et les dynamiques des modèles de connaissance des deux robots nécessite l'application de l'algorithme d'identification MCR à trace constante de gain égal 100.

Les trajectoires imposées aux deux robots sont les même que celles présentées dans le chapitre précédent, a savoir:

- Trajectoire polynomiale du troisième degré dans le temps.
- Trajectoire cercle dans l'espace.

L'étude par simulation nous montrera la puissance et les capacités de l'algorithme de commande qui donne une bonne poursuite des trajectoires.

Tous les résultats sont présentés sur des figures dont chacun comporte, les sorties et consignes, les erreurs et les commandes pour chaque variable généralisée (θ_1, θ_2) .

Les résultats obtenus par les deux robots ont été séparés en deux parties, ceux obtenus par le robot PUMA dans la partie une et ceux du robot SCARA dans la partie deux.

Remarque.

- tous les résultats ont été réalisés pour $N1[1]=N1[2]=1$.
- $N1[i], N2[i], NU[i], \lambda_i$ et $U_{max}[i]$ pour $i=1,2$: représentent les paramètres de synthèse de la commande a savoir.
 - $N1[i]$: est l'horizon initial de prédiction.
 - $N2[i]$: est l'horizon maximal de prédiction.
 - $NU[i]$: est l'horizon de commande.
 - λ_i : est la pondération de la commande.
 - $U_{max}[i]$: est la limitation de la commande.

PARTIE I: Résultats de simulation du robot PUMA.

A partir du choix des paramètres de synthèse de la GPC, les résultats de simulation concernant le robot PUMA pour une trajectoire polynomiale sont présentés.

La figure (5.1) montre l'évolution des variables généralisées θ_1 et θ_2 , les erreurs et les commandes correspondantes pour les paramètres suivant:

$$\left\{ \begin{array}{l} \lambda_1 = 7.10^{-4} \quad \lambda_2 = 10^{-3} \\ N2[1] = N2[2] = 10 \\ NU[1] = NU[2] = 10 \\ U_{max}[1] = U_{max}[2] = 10 \end{array} \right.$$

Une augmentation de $U_{max}[1] = U_{max}[2] = 15$, figure (5.2), améliore nettement les performances en comparaison à la figure (5.1). La possibilité de spécifier pour chaque bras manipulateur ses propres paramètres nous permet d'agir sur la limitation du 1^{er} bras et du 2^{eme} bras séparément, a savoir: $U_{max}[1] = 15$ et $U_{max}[2] = 10$, ce qui correspond au résultat consigné sur la figure (5.3).

Une légère augmentation de l'horizon de prédiction, $N2[1] = N2[2] = 11$, figure (5.4), améliore l'évolution de la commande en comparaison aux figures (5.1)(5.2)(5.3).

Sans agir sur la limitation de la commande de bon résultats peuvent être obtenus en changeant légèrement la valeur de $\lambda_2 = 7.10^{-5}$, figure (5.5).

Les résultats précédents montrent l'influence de la limitation de la commande envers la dynamique de poursuite et des oscillations de la commande.

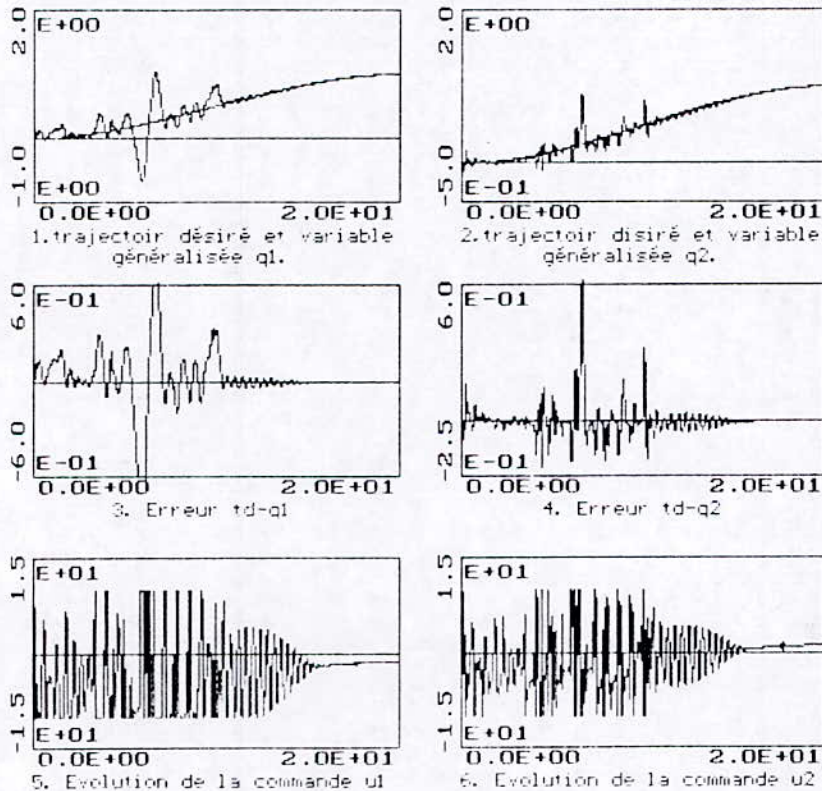


Figure 5.1 trajectoire désiré (td), position (q1, q2) pour le robot Puma pour $N1[1]=1$ $N2[1]=10$ $NU[1]=5$ $N1[2]=1$ $N2[2]=10$ $NU[2]=5$ $\lambda_1 = 0.0007$ $\lambda_2 = 0.001$ $U_{max}[1]=10$ $U_{max}[2]=10$.

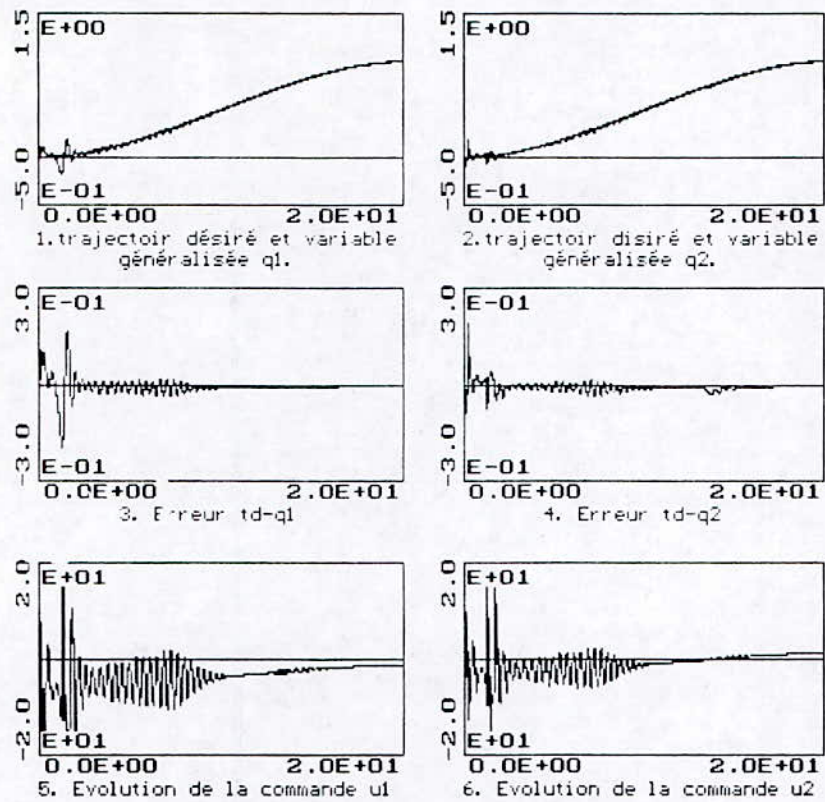


Figure 5.2 trajectoire désiré (td), position (q1,q2) pour le robot Puma pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=5$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=5$ $\lambda_1=0.0007$ $\lambda_2=0.001$ $U_{max}[1]=15$ $U_{max}[2]=15$.

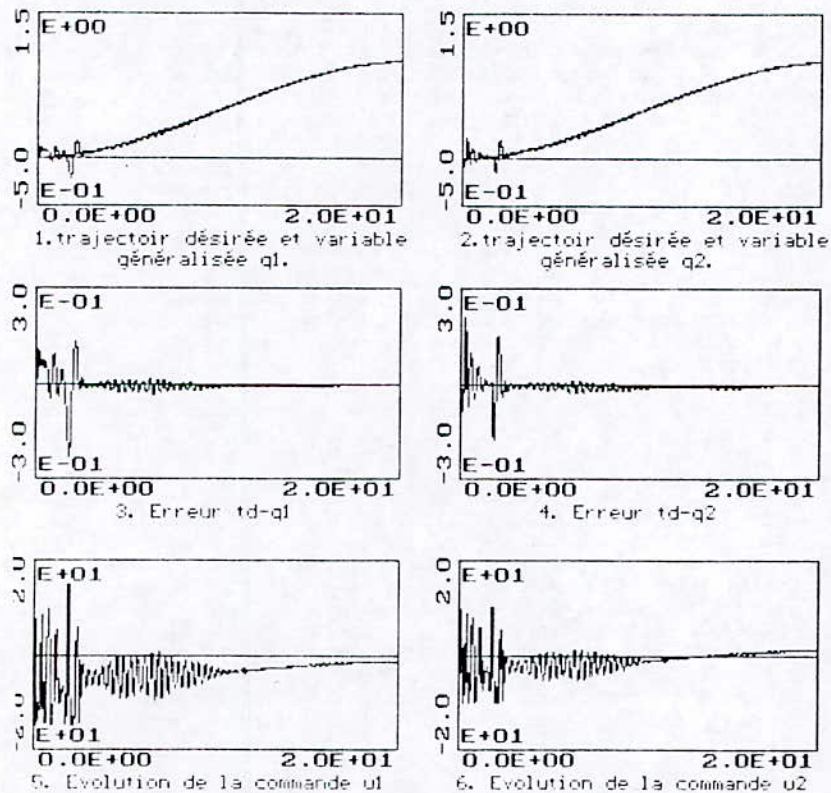


Figure 5.3 trajectoire désirée (td), position (q1,q2) pour le robot Puma pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=5$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=5$ $\lambda_1=0.0007$ $\lambda_2=0.001$ $U_{max}[1]=15$ $U_{max}[2]=10$.

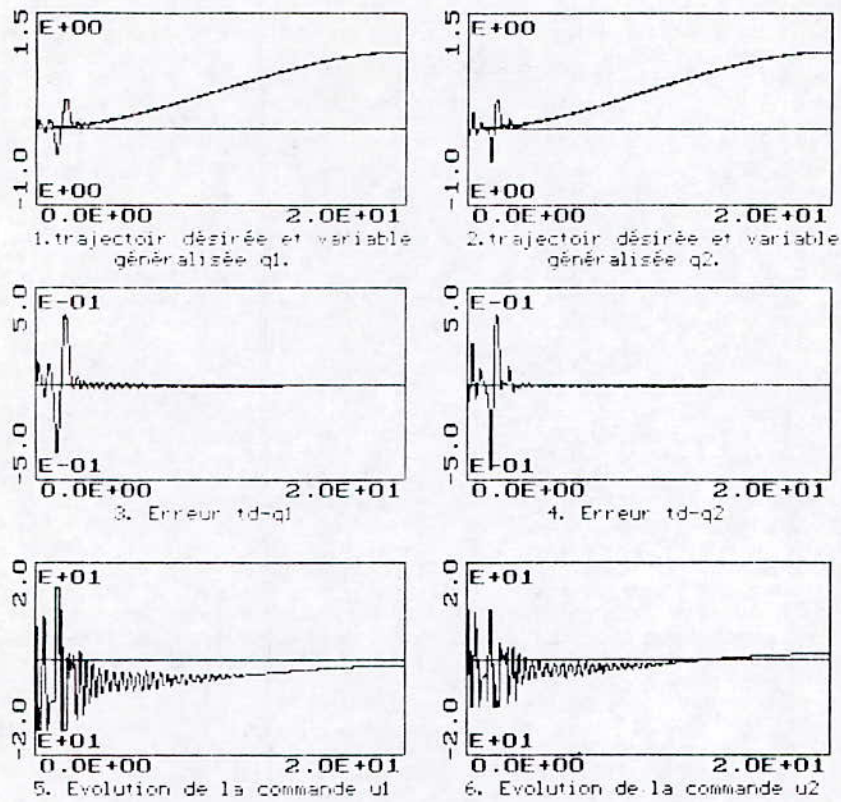


Figure 5.4 trajectoire désirée (td), position (q1,q2) pour le robot Puma pour $N1[1]=1$ $N2[1]=11$ $NU[1]=5$ $N1[2]=1$ $N2[2]=11$ $NU[2]=5$ $\lambda_1=0.00007$ $\lambda_2=0.001$ $U_{max}[1]=15$ $U_{max}[2]=10$.

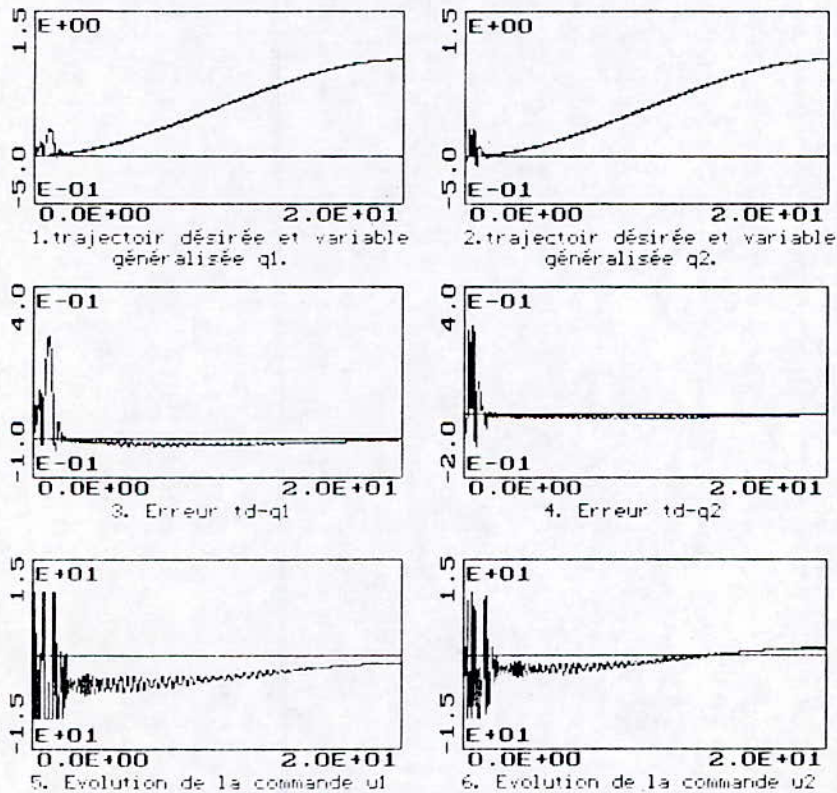


Figure 5.5 trajectoire désirée (td), position (q1,q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=10$ $NU[1]=5$ $N1[2]=1$ $N2[2]=10$ $NU[2]=5$ $\lambda_1=0.00007$ $\lambda_2=0.001$ $U_{max}[1]=10$ $U_{max}[2]=10$.

En gardant les mêmes pondération que précédemment, c'est à dire $\lambda_1=7.10^{-4}$ $\lambda_2=10^{-3}$, et pour une limitation à dix, l'effet de l'augmentation de $N2[1]=N2[2]=5$ à $N2[1]=N2[2]=10$ est présenté sur les figures (5.6)(5.7). Ces résultats ont été réalisés pour un horizon de commande égal à un. Si nous comparons la figure(5.7) à la figure(5.1) on peut dire que l'augmentation de $NU[i]$ rend les commandes plus excitantes. Il convient donc, pour obtenir une bonne poursuite de la trajectoire, pour des pondérations fixes, soit d'agir sur la limitation de la commande soit sur l'horizon de commande, la figure (5.8) montre l'effet d'augmenter $U_{max}[1]=U_{max}[2]=15$, ceci peut être comparé à la figure (5.2) où $U_{max}[1]=U_{max}[2]=5$.

La figure(5.8) peut être améliorée soit en agissant sur $\lambda_2=10^{-2}$, figure (5.9), soit sur la limitation $max[i]=10$ et $\lambda_2=10^{-2}$, figure(5.10).

Les figures(5.11)(5.12)(5.13) illustrent la possibilité d'améliorer les performances en agissant sur différents paramètres, ceci pour:

$$\left\{ \begin{array}{l} \lambda_1=7.10^{-6} \quad \lambda_2=10^{-5} \\ N2[1]=N2[2]=5 \\ NU[1]=NU[2]=5 \\ U_{max}[1]=15 \quad U_{max}[2]=10 \end{array} \right. \quad \left\{ \begin{array}{l} \lambda_1=7.10^{-6} \quad \lambda_2=10^{-1} \\ N2[1]=N2[2]=10 \\ NU[1]=NU[2]=1 \\ U_{max}[1]=U_{max}[2]=10 \end{array} \right. \quad \left\{ \begin{array}{l} \lambda_1=7.10^{-6} \quad \lambda_2=10^{-5} \\ N2[1]=N2[2]=10 \\ NU[1]=NU[2]=1 \\ U_{max}[1]=U_{max}[2]=10 \end{array} \right.$$

Les résultats précédents figure (5.1) à (5.13), ont été réalisés pour des pondérations de commande non nulles. Des résultats très satisfaisant ont été obtenus pour $\lambda_1=\lambda_2=0$. Les figures (5.14)(5.15)(5.16)(5.17) illustrent l'évolution de l'erreur des variables généralisées et de la commande pour les deux articulations, respectivement pour les paramètres de synthèse suivants:

$$\left\{ \begin{array}{l} N2[i]=5 \\ NU[i]=1 \\ U_{max}[i]=10 \end{array} \right. \quad \left\{ \begin{array}{l} N2[i]=10 \\ NU[i]=1 \\ U_{max}[i]=10 \end{array} \right. \quad \left\{ \begin{array}{l} N2[i]=5 \\ NU[i]=1 \\ U_{max}[i]=20 \end{array} \right. \quad \left\{ \begin{array}{l} N2[i]=10 \\ NU[i]=1 \\ U_{max}[i]=20 \end{array} \right.$$

En comparant les quatre figures nous pouvons en déduire que l'augmentation de $N2$ améliore nettement la poursuite pour $U_{max}[i]=10$, figure(5.14)(5.15), tandis qu'elles restent presque inchanger pour $U_{max}[i]=20$, figure(5.16)(5.17). Nous tenons à signaler que la valeur de $U_{max}[i]=10$ est mieux adapté que $U_{max}[i]=20$, figures(5.14)(5.16) et figures(5.15)(5.17).

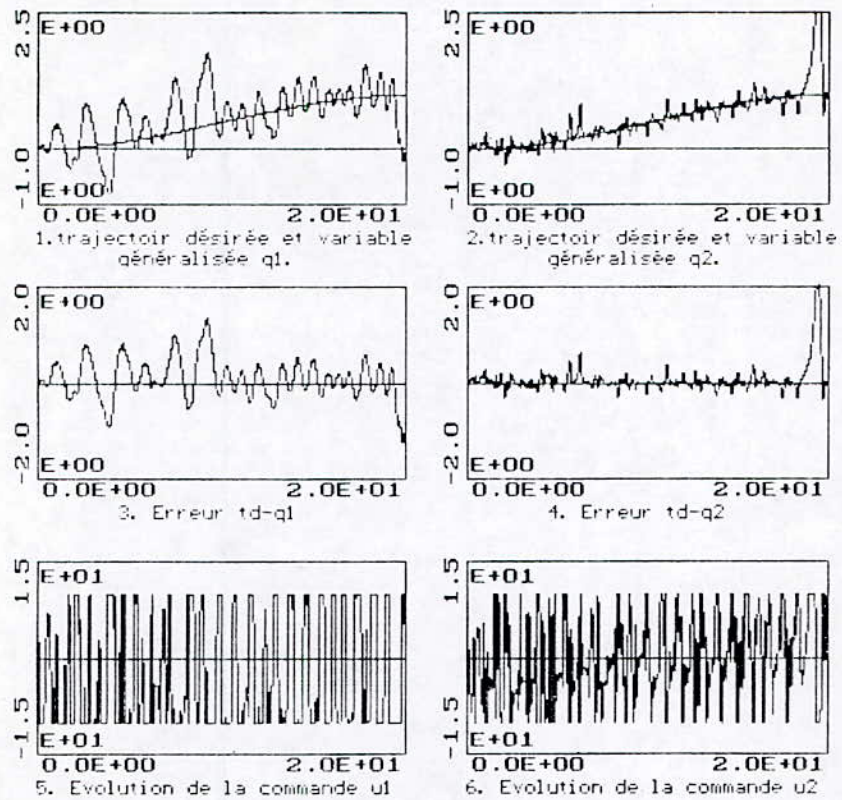


Figure 5.6 trajectoire désirée (td), position (q1,q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=5$ $NU[1]=1$ $N1[2]=1$ $N2[2]=5$ $NU[2]=1$ $\lambda_1=0.0007$ $\lambda_2=0.001$ $U_{max}[1]=10$ $U_{max}[2]=10$.

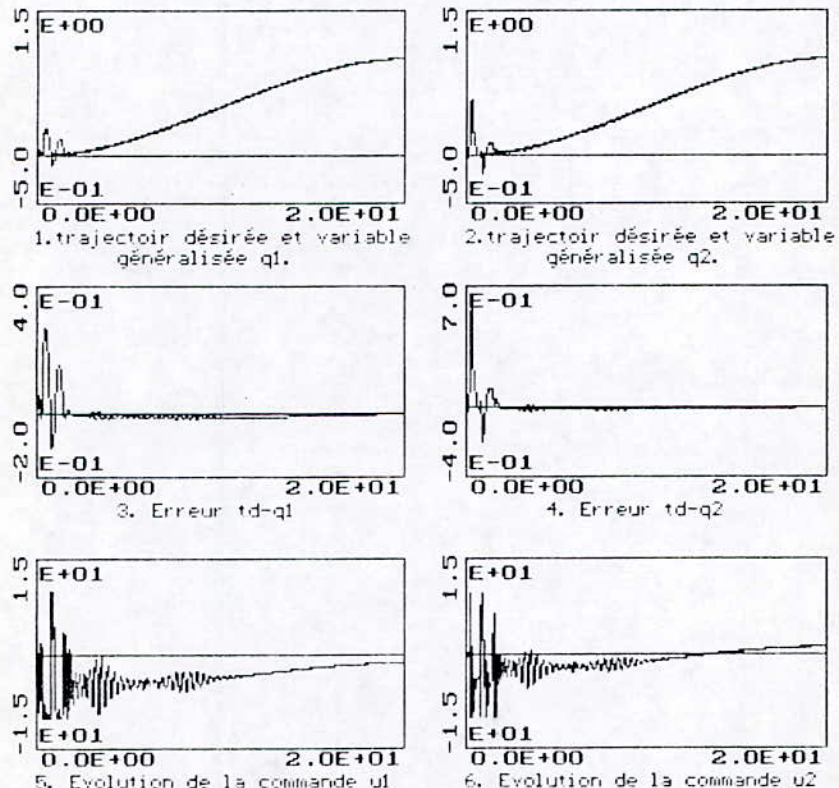


Figure 5.7 trajectoire désirée (td), position (q1,q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=10$ $NU[1]=1$ $N1[2]=1$ $N2[2]=10$ $NU[2]=1$ $\lambda_1=0.0007$ $\lambda_2=0.001$ $U_{max}[1]=10$ $U_{max}[2]=10$.

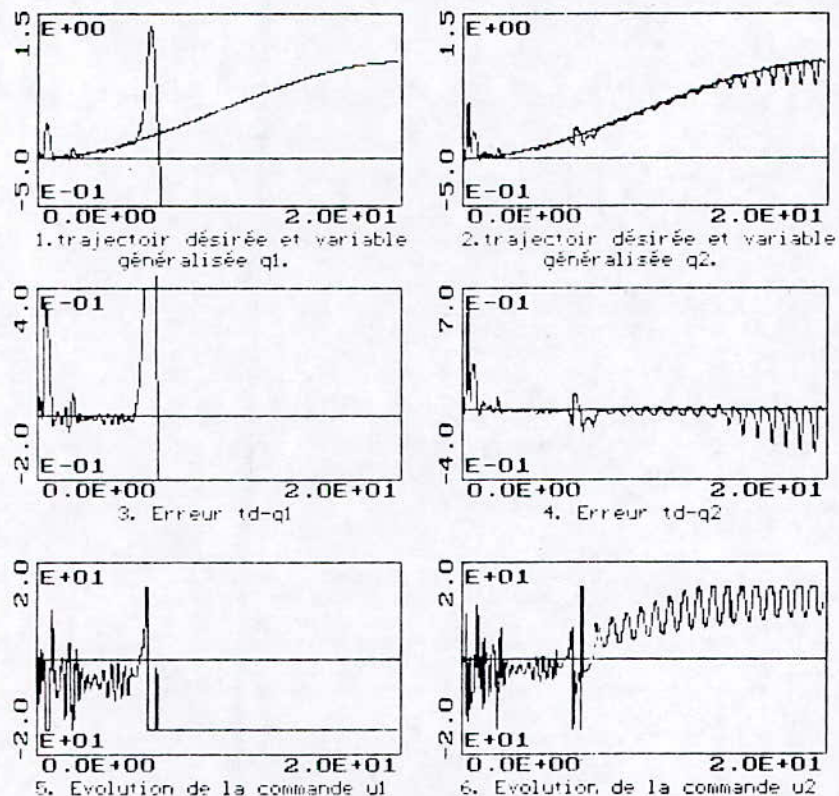


Figure 5.8 trajectoire désirée (td), position (q1, q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=10$ $NU1[1]=1$ $N1[2]=1$ $N2[2]=10$ $NU2[2]=1$ $\lambda_1=0.0007$ $\lambda_2=0.001$ $U_{max}[1]=15$ $U_{max}[2]=15$.

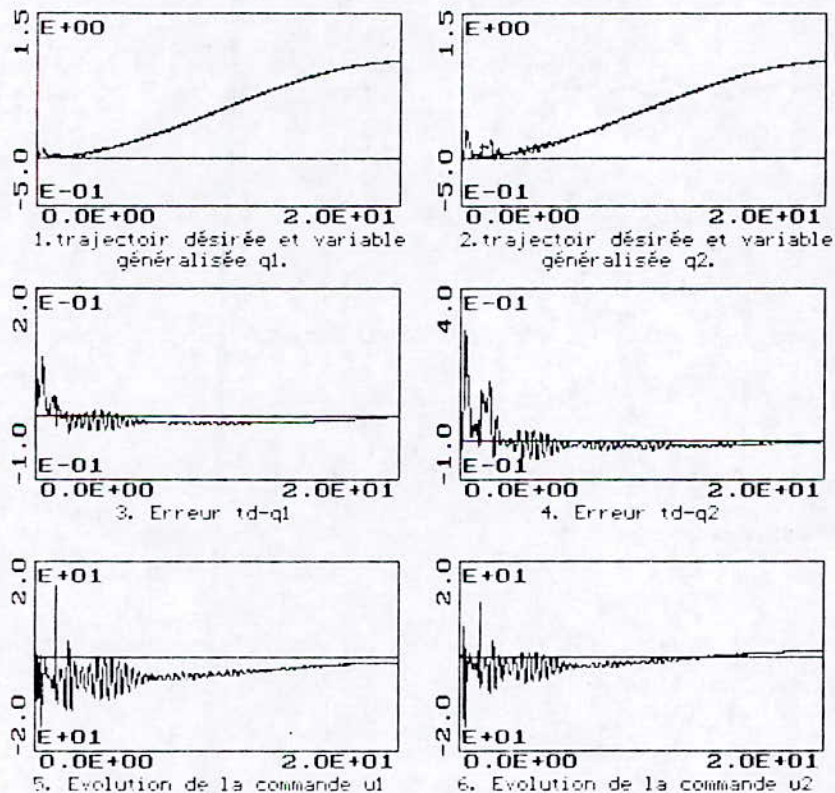


Figure 5.9 trajectoire désirée (td), position (q1, q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=10$ $NU1[1]=1$ $N1[2]=1$ $N2[2]=10$ $NU2[2]=1$ $\lambda_1=0.0007$ $\lambda_2=0.01$ $U_{max}[1]=15$ $U_{max}[2]=15$.

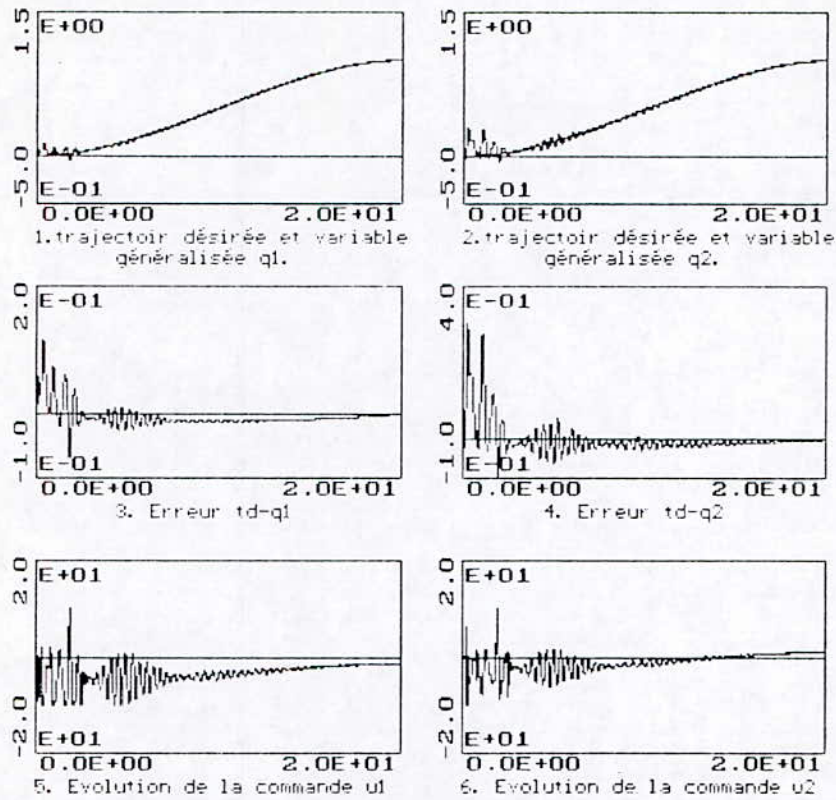


Figure 5.10 trajectoire désirée (td), position (q1, q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=1$ $\lambda_1=0.00007$ $\lambda_2=0.01$ $U_{max}[1]=10$ $U_{max}[2]=10$.

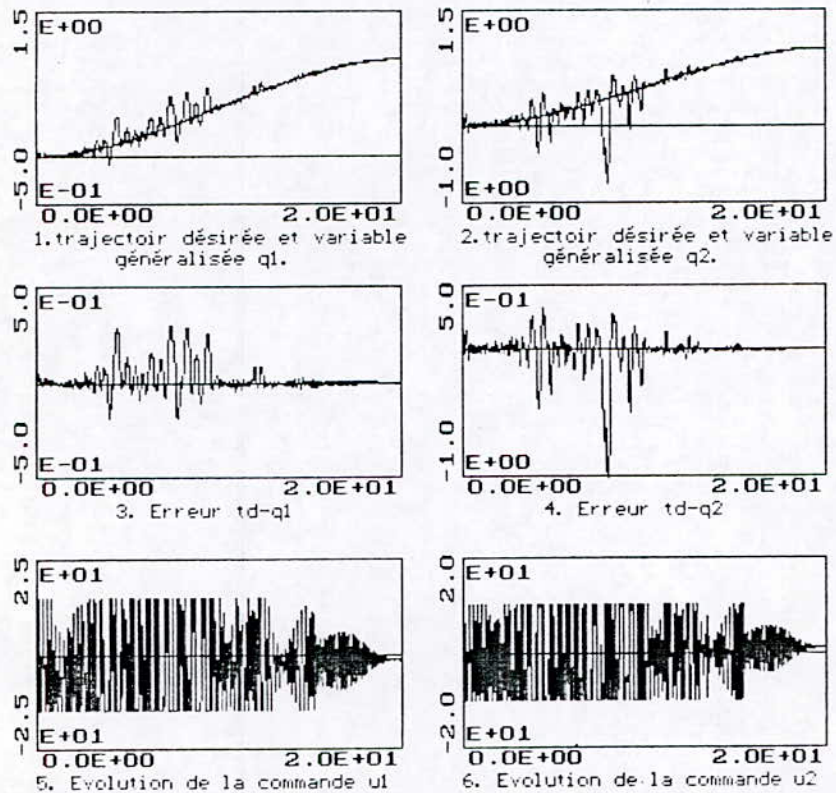


Figure 5.11 trajectoire désirée (td), position (q1, q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=5$ $NUC[1]=5$ $N1[2]=1$ $N2[2]=5$ $NUC[2]=5$ $\lambda_1=0.000007$ $\lambda_2=0.00001$ $U_{max}[1]=15$ $U_{max}[2]=10$.

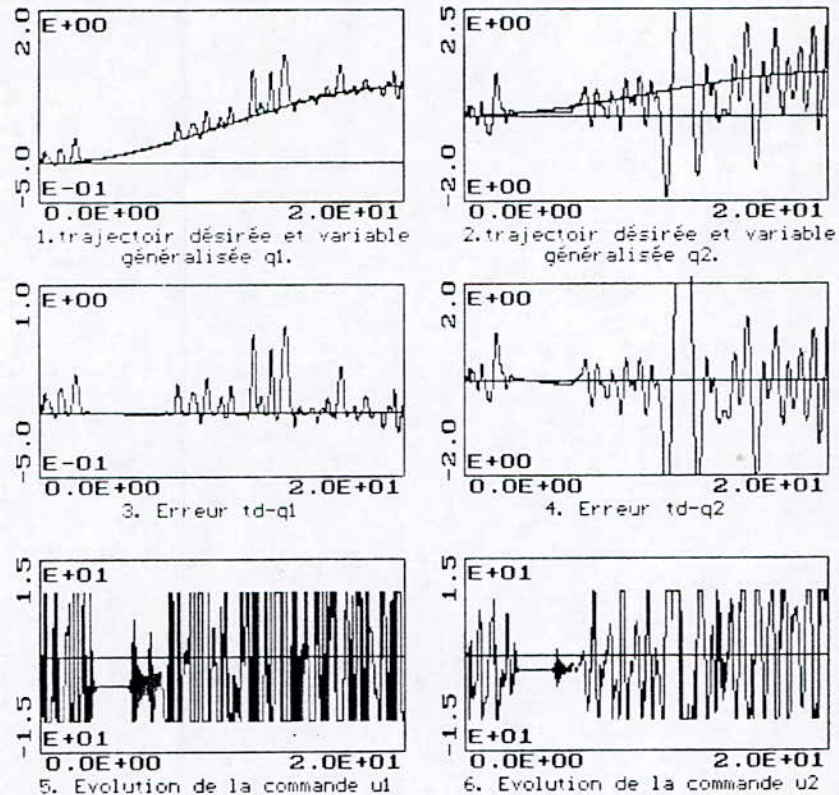


Figure 5.12 trajectoire désirée (td), position (q1,q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=1$ $\lambda_1=0.000007$ $\lambda_2=0.1$ $Umax[1]=10$ $Umax[2]=10$.

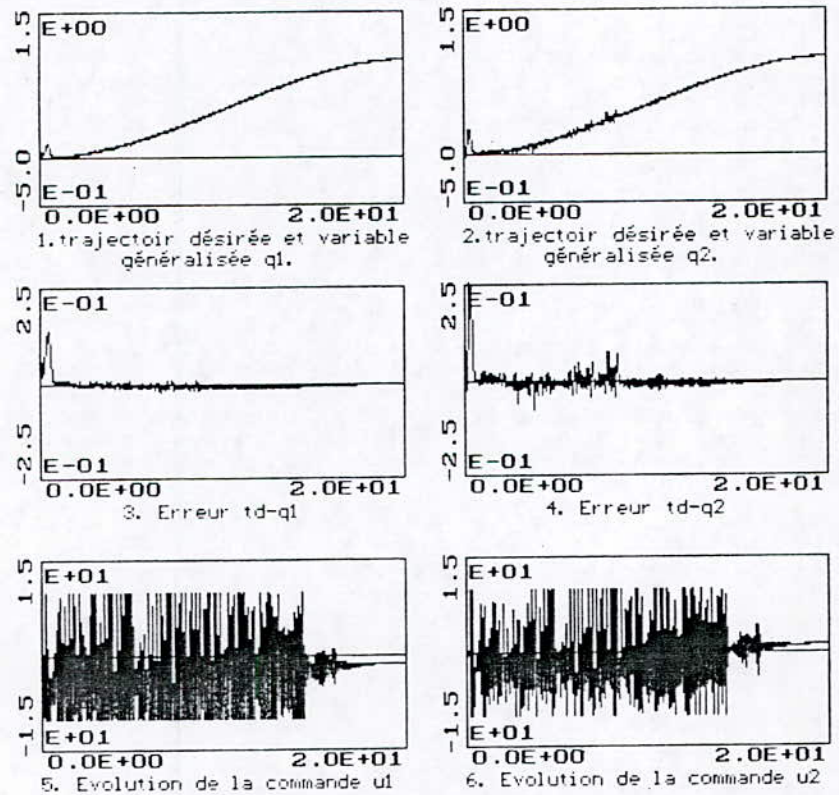


Figure 5.13 trajectoire désirée (td), position (q1,q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=1$ $\lambda_1=0.000007$ $\lambda_2=0.00001$ $Umax[1]=10$ $Umax[2]=10$.

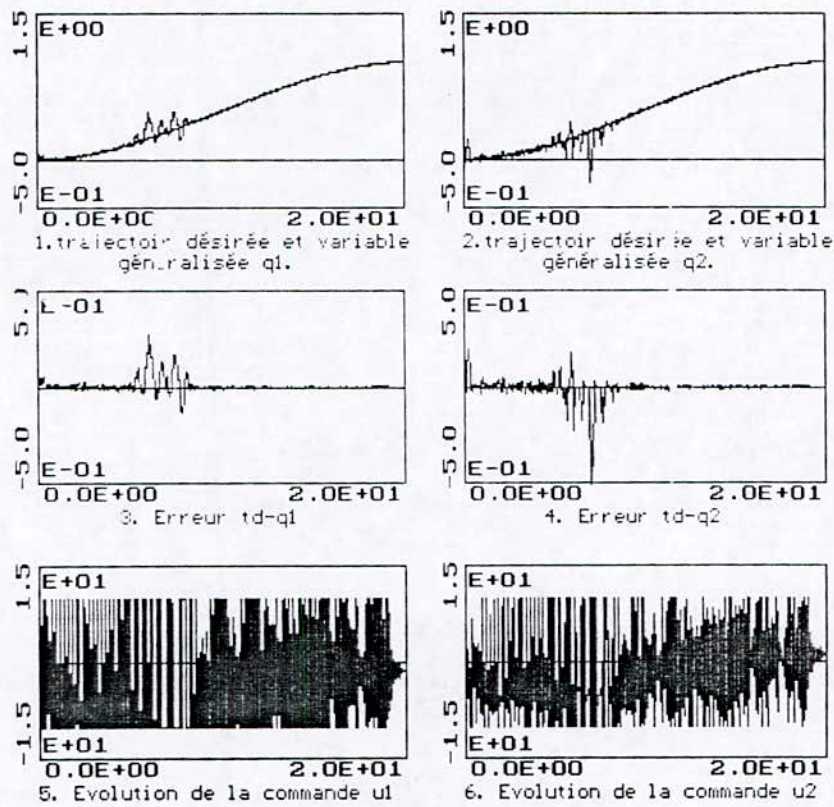


Figure 5.14 trajectoire désirée (td), position (q1,q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=5$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=5$ $NUC[2]=1$ $\lambda_1=0$ $\lambda_2=0$ $Umax[1]=10$ $Umax[2]=10$.

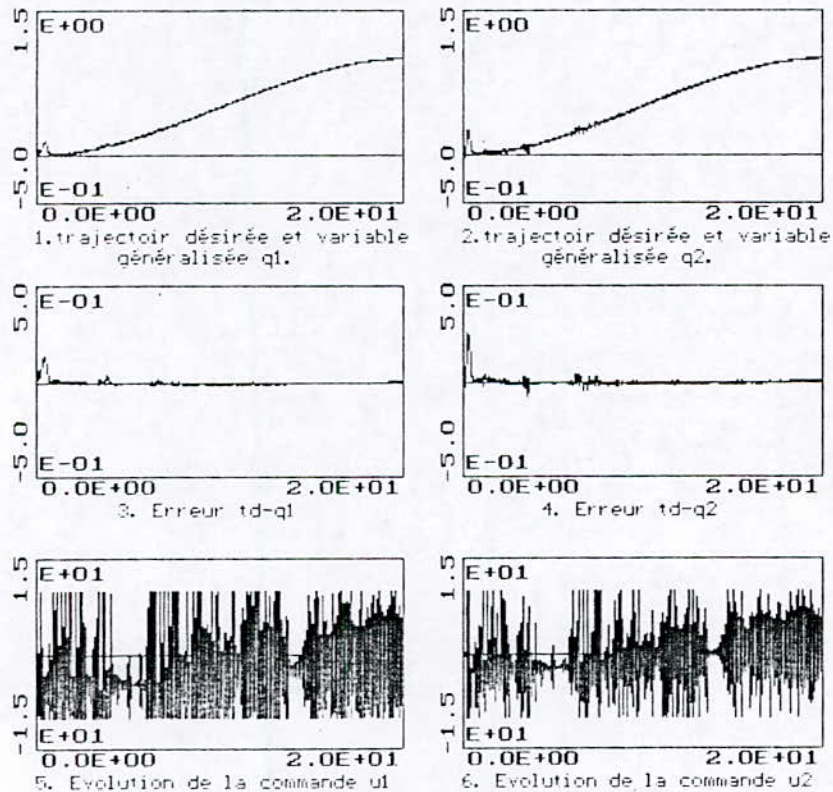


Figure 5.15 trajectoire désirée (td), position (q1,q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=1$ $\lambda_1=0$ $\lambda_2=0$ $Umax[1]=10$ $Umax[2]=10$.

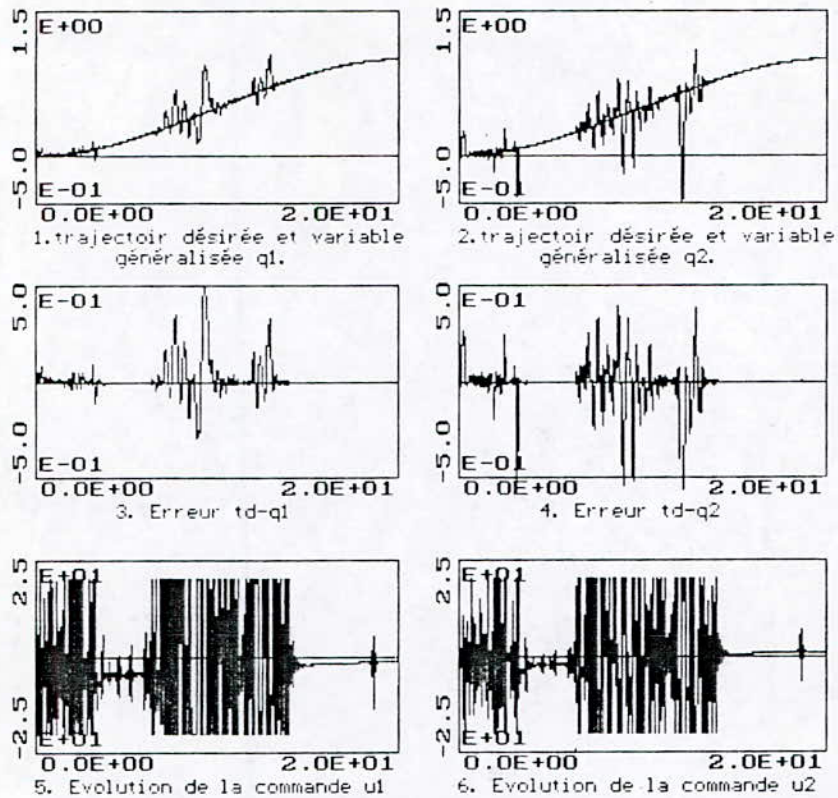


Figure 5.16 trajectoire désirée (td), position (q_1, q_2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=5$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=5$ $NUC[2]=1$ $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=20$ $U_{max}[2]=20$.

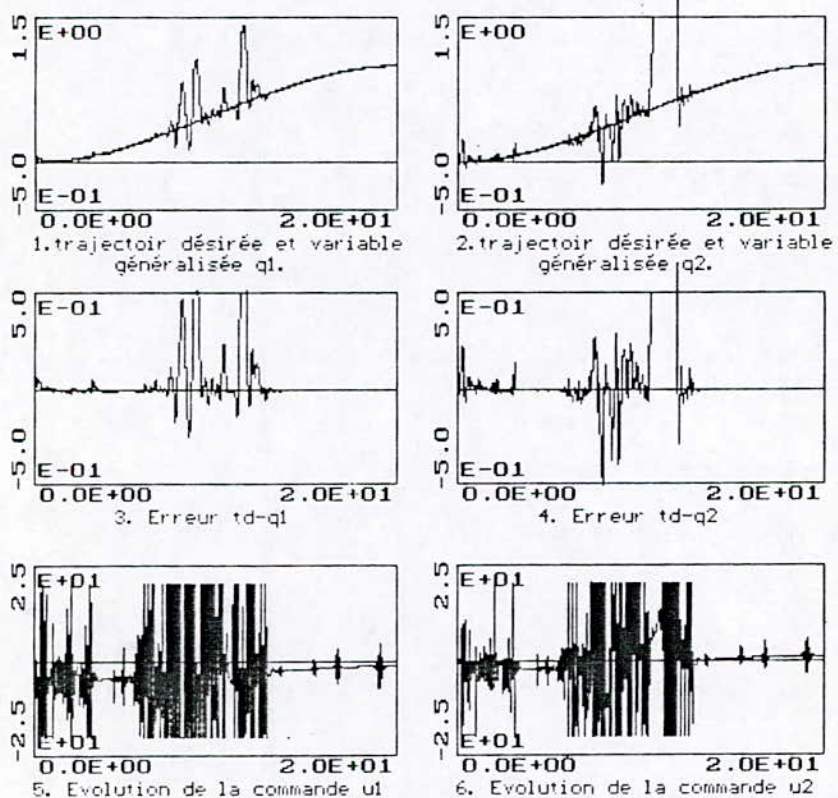


Figure 5.17 trajectoire désirée (td), position (q_1, q_2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=1$ $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=20$ $U_{max}[2]=20$.

Dans cette partie la poursuite de la trajectoire cercle dans le temps concernant l'évolution des variables généralisées et dans l'espace, est considérée. La simulation a été réalisée sur 20s, le temps que le robot effectue deux tours complet. Les figures(5.18)(5.19) pour $\lambda_i=0, N2[i]=5, NU[i]=1$ et $U_{max}[i]=10$ montrent l'évolution des variables généralisées θ_1 et θ_2 , de l'erreur et de la commande respectivement pour:

$$\begin{cases} x[1]=0 ; & x[3]=0 \\ x[1]=0.295 ; & x[3]=0.676 \end{cases}$$

où x est le vecteur d'état du robot tel que $x[1]$ et $x[3]$ représentent les positions initiales du robot avant d'entamer la tâche.

Les figures(5.18.a)(5.18.b) et les figures(5.19.a)(5.19.b) illustrent la poursuite du cercle dans l'espace pour les conditions initiales respectivement nulles et non nulles.

La même simulation a été réalisée pour d'autres paramètres de synthèse $N2[i]=10$ et $NU[i]=5$. Les résultats sont consignés sur les figures(5.20)(5.21) pour l'évolution des variables généralisées et les figures(5.20.a)(5.20.b), (5.21.a)(5.21.b) pour l'évolution du robot dans l'espace cartésien.

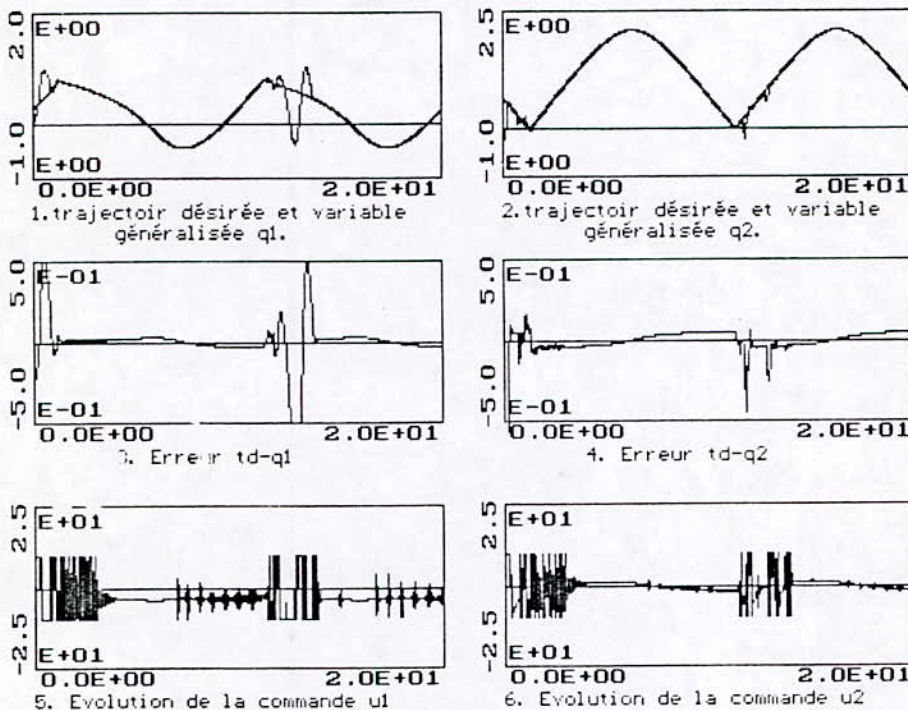


Figure 5.18 trajectoir désirée cercle (td), position (q1, q2) pour le robot PUMA pour $NI[1]=1$ $N2[1]=5$ $NU[1]=1$ $NI[2]=1$ $N2[2]=5$ $NU[2]=1$ $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0$ $x[3]=0$.

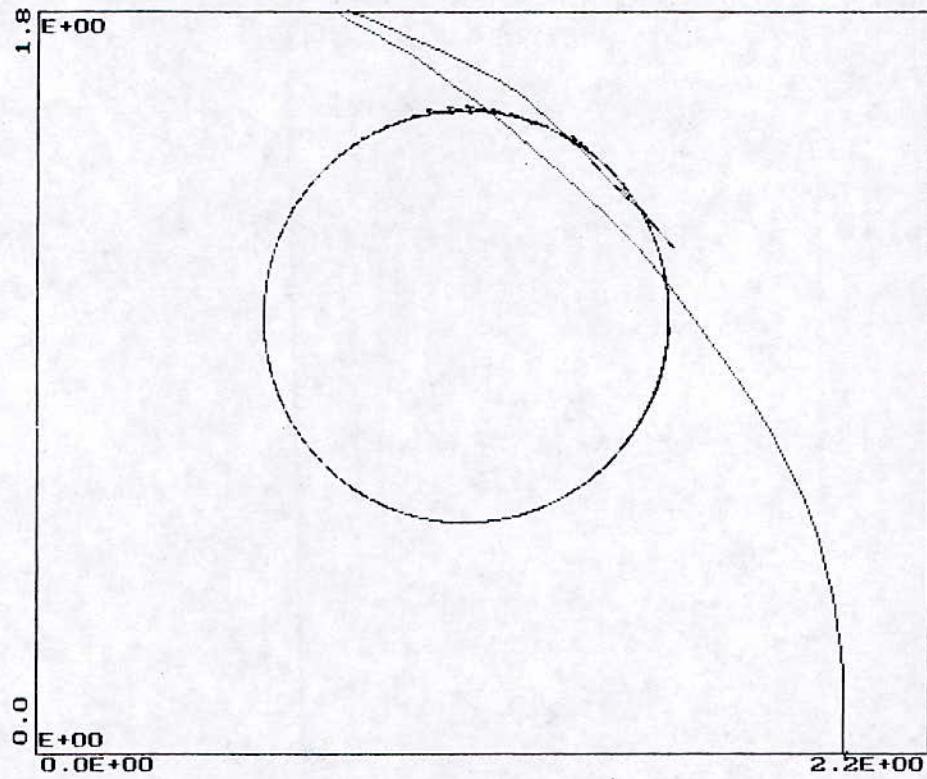


Figure 5.18.a Evolution du robot PUMA dans l'espace cartésien poursuivant une trajectoire cercle pour $N1[1]=1$ $N2[1]=5$ $NU[1]=1$ $N1[2]=1$ $N2[2]=5$ $NU[2]=1$, premier tour, $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0$ $x[3]=0$.

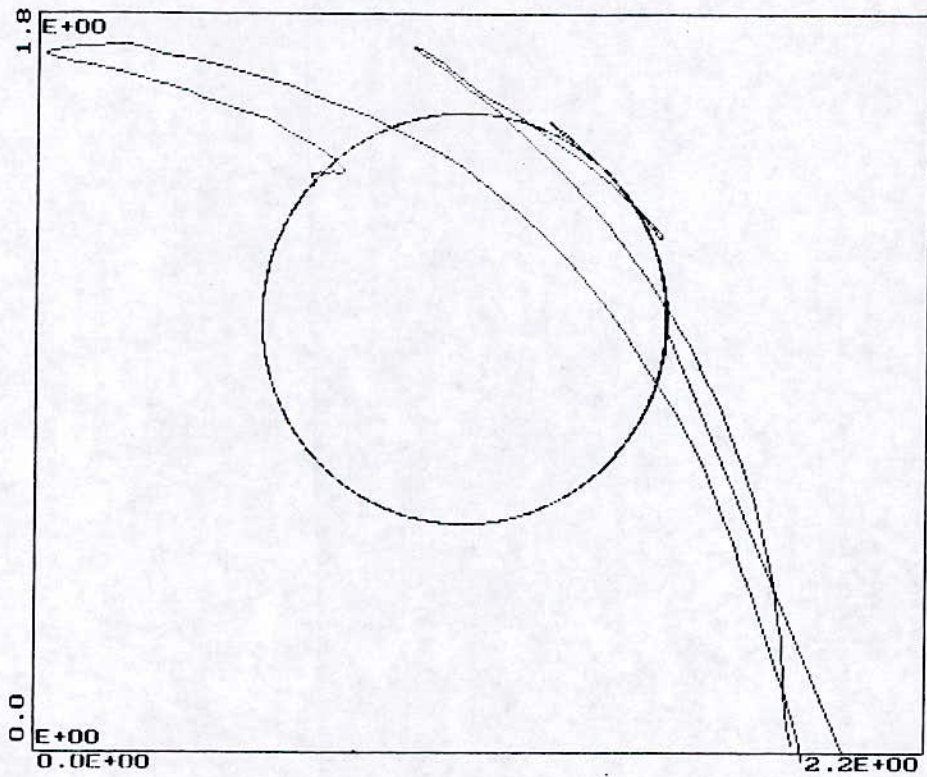


Figure 5.18.b Evolution du robot PUMA dans l'espace cartésien poursuivant une trajectoire cercle pour $N1[1]=1$ $N2[1]=5$ $NU[1]=1$ $N1[2]=1$ $N2[2]=5$ $NU[2]=1$, 2ème tour, $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0$ $x[3]=0$.

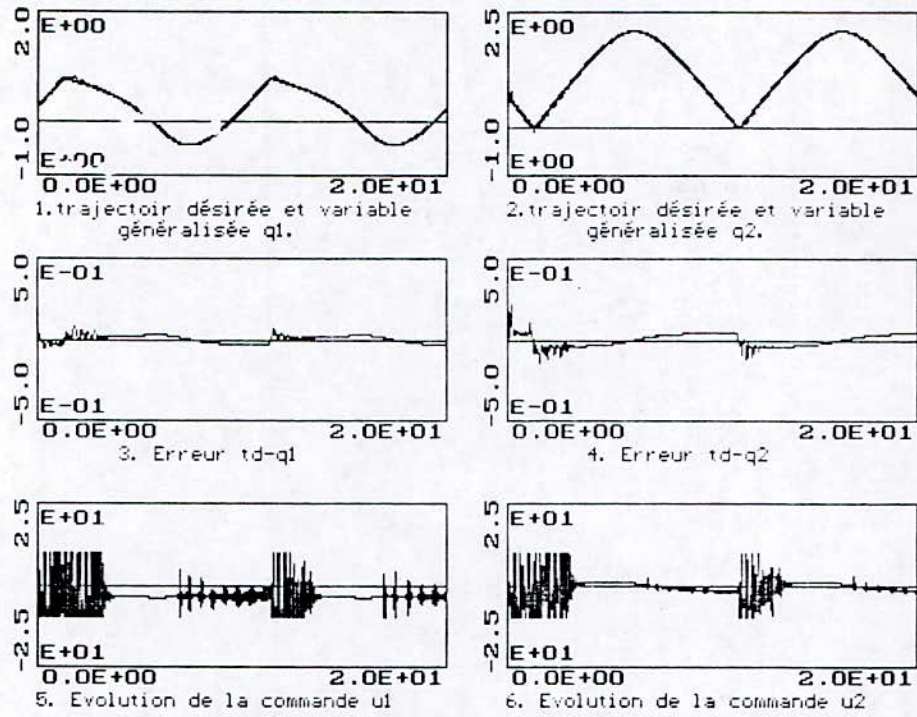


Figure 5.19 trajectoire désirée cercle (td), position (q_1, q_2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=5$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=5$ $NUC[2]=1$ $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0.295$ $x[3]=0.676$.

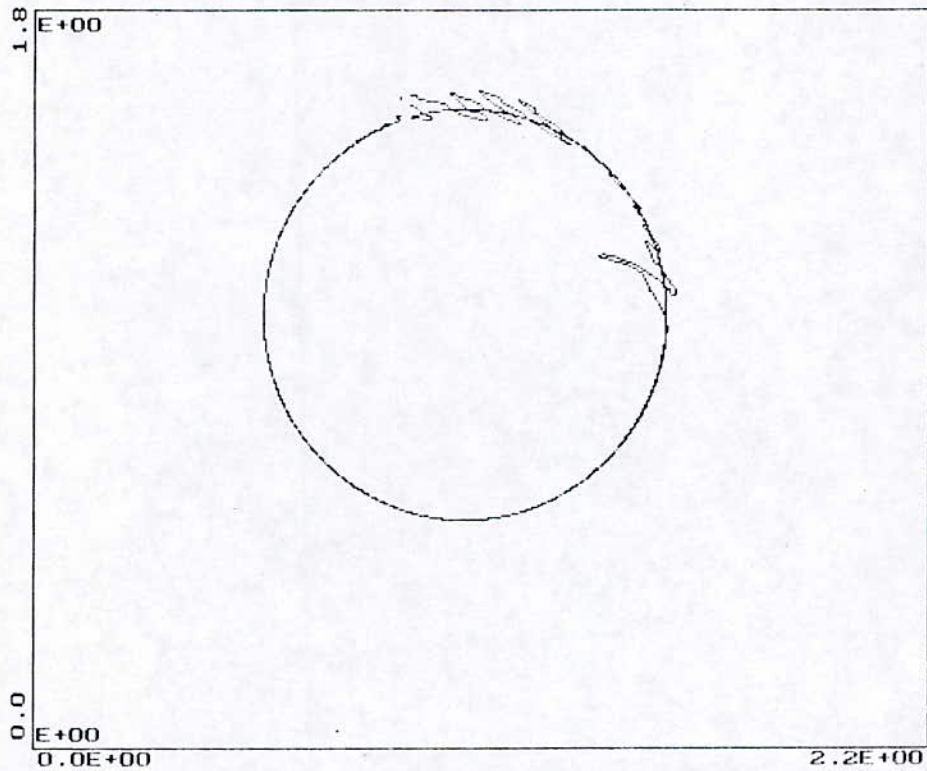


Figure 5.19.a Evolution du robot PUMA dans l'espace cartésien pour une trajectoire cercle pour $N1[1]=1$ $N2[1]=5$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=5$ $NUC[2]=1$, premier tour, $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0.295$ $x[3]=0.676$.

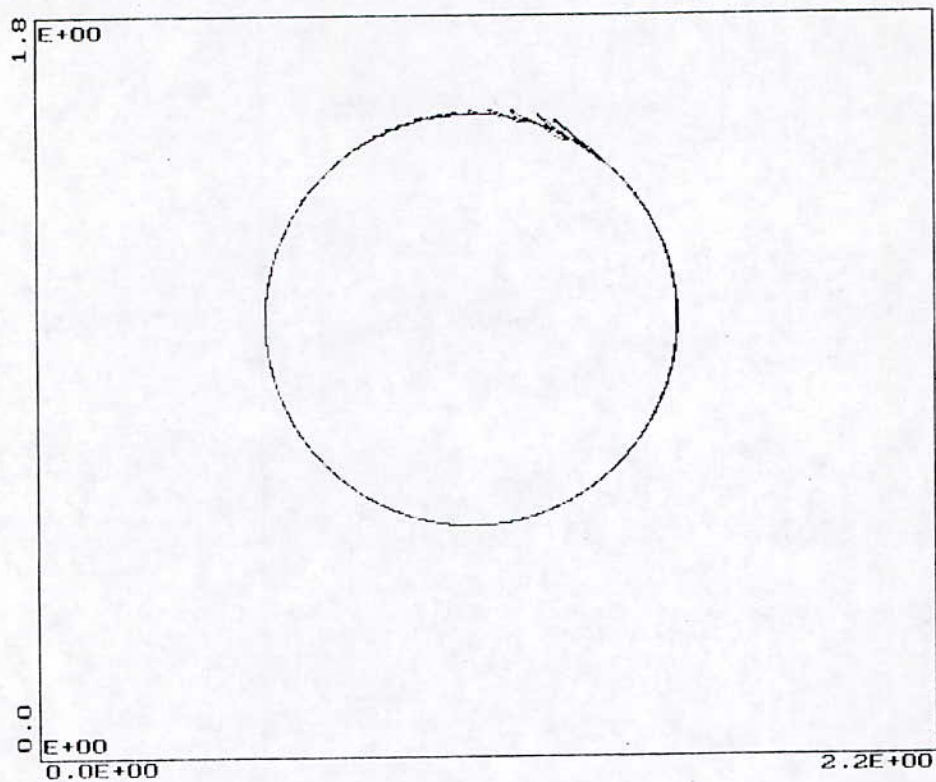


Figure 5.19.b Evolution du robot PUMA dans l'espace cartésien poursuivant une trajectoire cercle pour $N1[1]=1$ $N2[1]=5$ $NU[1]=1$ $N1[2]=1$ $N2[2]=5$ $NU[2]=1$, 2ème tour, $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0.295$ $x[3]=0.676$.

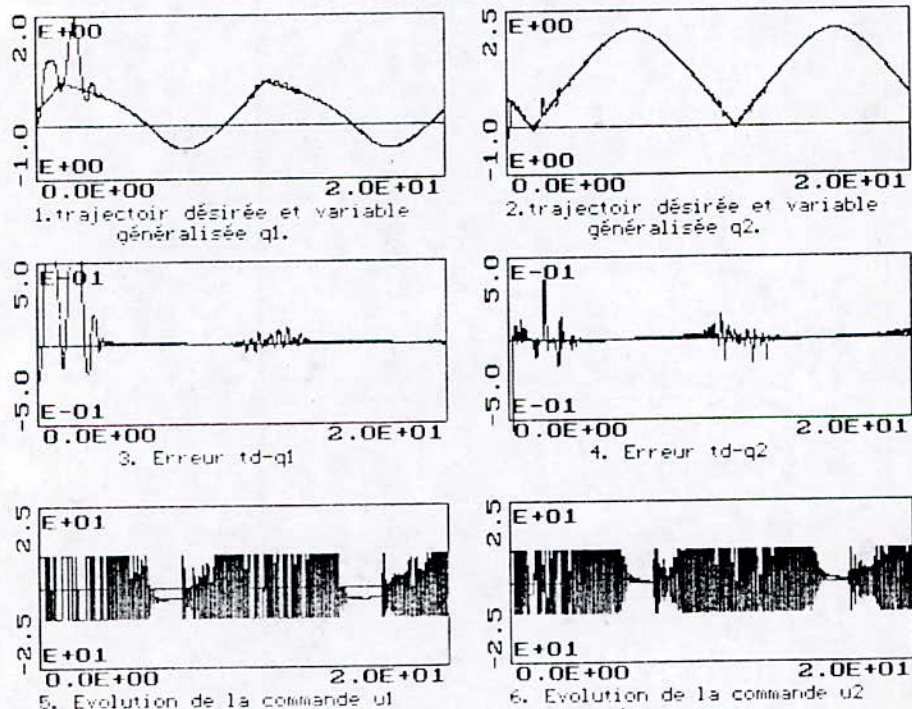


Figure 5.20 trajectoire désirée cercle (td), position (q1,q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=10$ $NU[1]=5$ $N1[2]=1$ $N2[2]=10$ $NU[2]=5$ $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0$ $x[3]=0$.

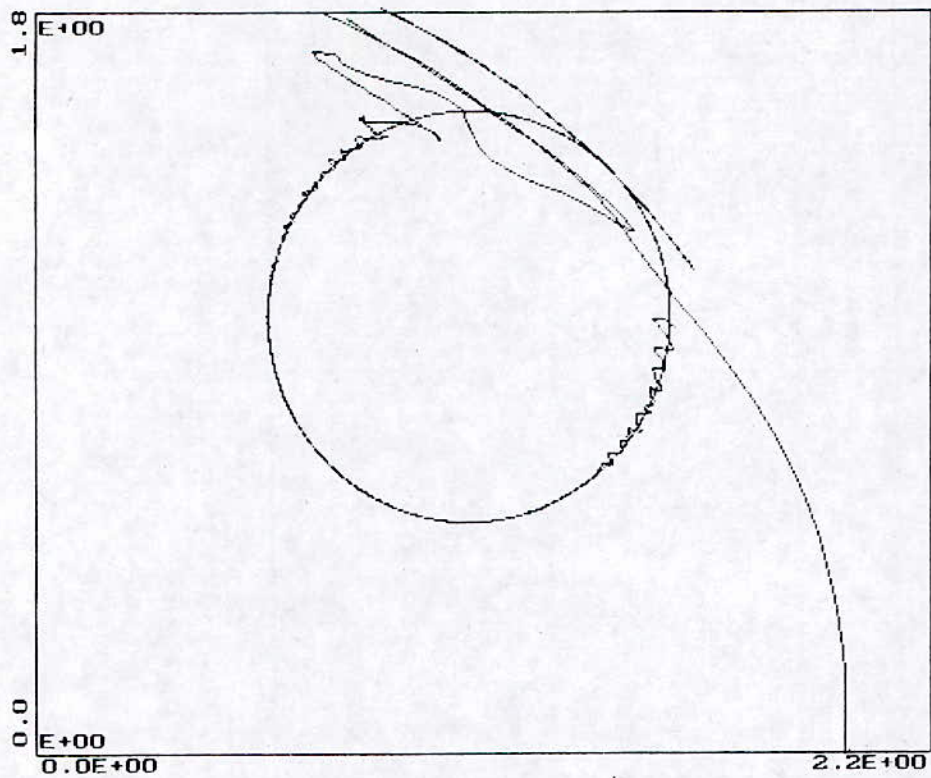


Figure 5.20.a Evolution du robot PUMA dans l'espace cartésien poursuivant une trajectoire cercle pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=5$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=5$, premier tour, $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0$ $x[3]=0$.

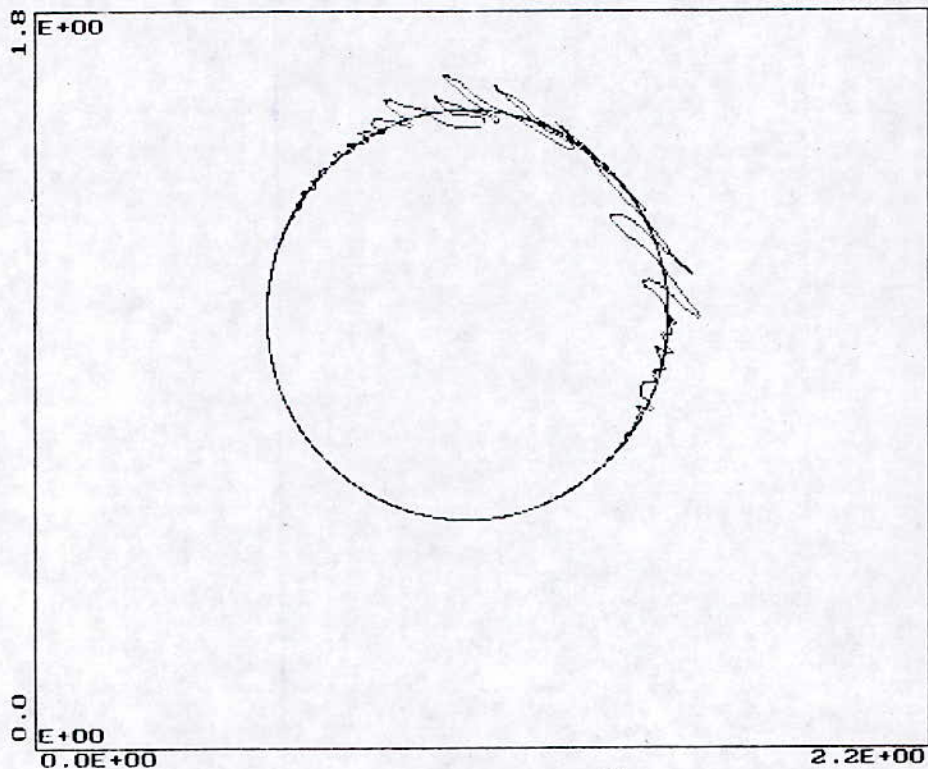


Figure 5.20.b Evolution du robot PUMA dans l'espace cartésien poursuivant une trajectoire cercle pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=5$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=5$, 2ème tour, $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0$ $x[3]=0$.

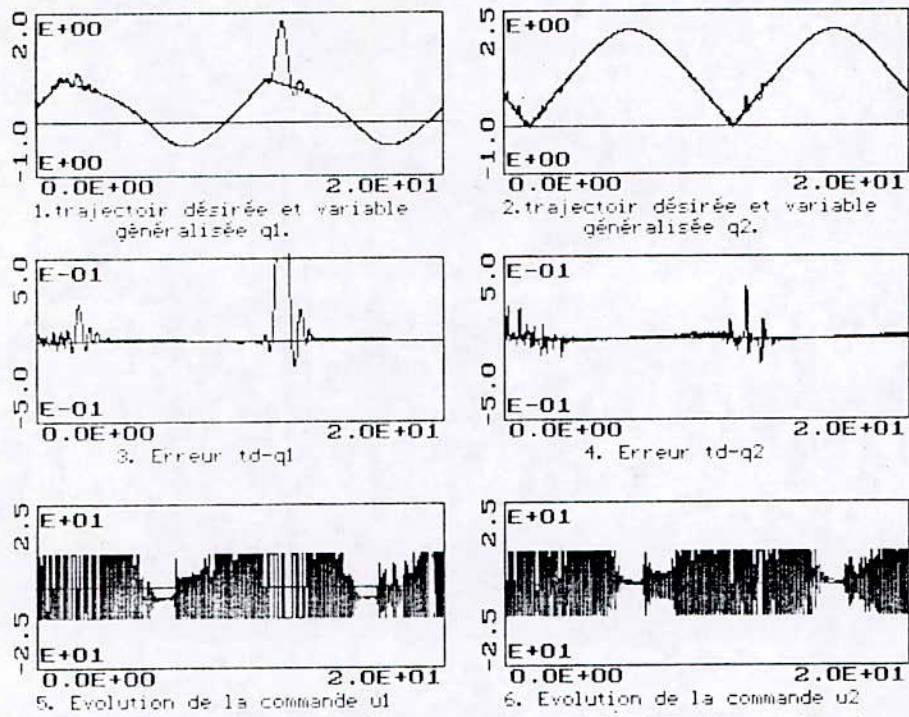


Figure 5.21 trajectoire désirée cercle (td), position (q1, q2) pour le robot PUMA pour $N1[1]=1$ $N2[1]=10$ $NUC1]=5$ $N1[2]=1$ $N2[2]=10$ $NUC2]=5$ $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0.295$ $x[3]=0.676$.

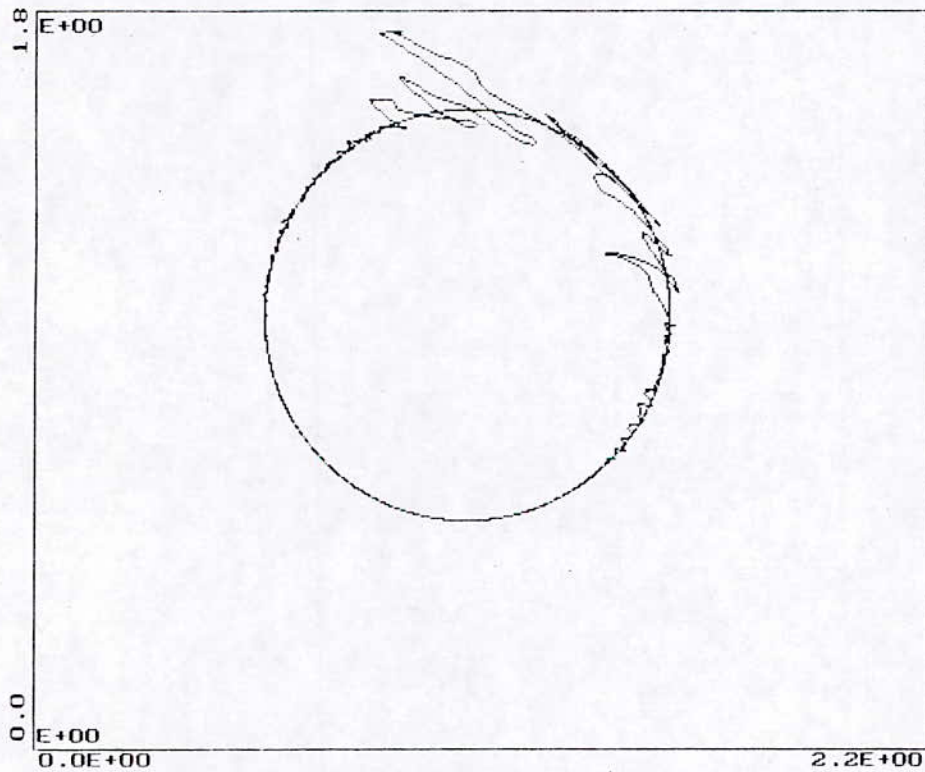


Figure 5.21.a Evolution du robot PUMA dans l'espace cartésien poursuivant une trajectoire cercle pour $N1[1]=1$ $N2[1]=10$ $NUC1]=5$ $N1[2]=1$ $N2[2]=10$ $NUC2]=5$, premier tour, $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0.295$ $x[3]=0.676$.

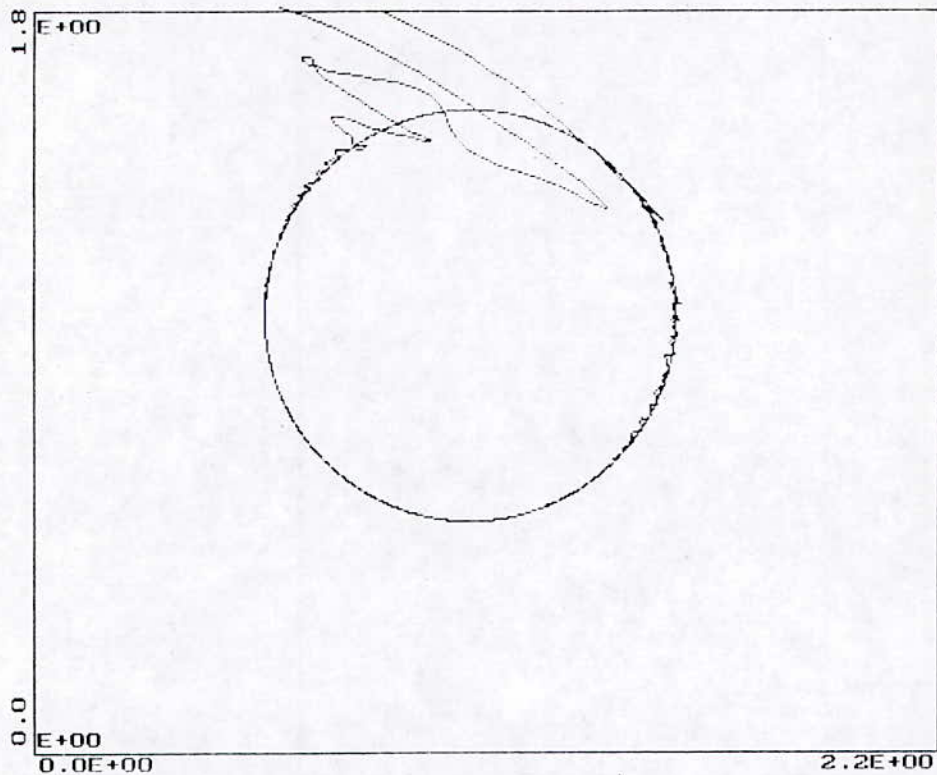


Figure 5.21.b Evolution du robot PUMA dans l'espace cartésien poursuivant une trajectoire circulaire pour $N1[1]=1$ $N2[1]=10$ $NU[1]=5$ $N1[2]=1$ $N2[2]=10$ $NU[2]=5$, 2ème tour, $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$ $x[1]=0.295$ $x[2]=0.676$.

PARTIE II: Résultats de simulation du robot SCARA.

Dans cette partie les résultats de simulation concernant le robot SCARA, qui est un robot à deux degrés de liberté évoluant sur un espace planaire, est considéré. La trajectoire désirée, imposée pour le robot est un polynôme de 3^{ème} degré puis un cercle.

En premier lieu nous avons testé la capacité de poursuite pour différentes valeurs de $N2[i]$ à savoir:

$$\left\{ \begin{array}{l} \text{pour } \lambda_i=0, NU[i]=1, U_{max}[i]=10 \\ N2[i]=1 \text{ relatif à la figure(5.22)} \\ N2[i]=5 \text{ relatif à la figure(5.23)} \\ N2[i]=10 \text{ relatif à la figure(5.24)} \end{array} \right.$$

Les résultats de la figure (5.22) ont été réalisés à partir des paramètres de synthèse de la GPC qui rendent ce dernier équivalent à l'algorithme de commande à VM. Nous tenons à signaler, que le paramètre $N2[i]$ affecte énormément les performances du robot et ceci en observant les erreurs et les commandes obtenues.

En second lieu, ce que réalise un changement de NU et U_{max} pour l'évolution du robot. En gardant, $\lambda_1=0$, $N2[i]=10$ et $NU[i]=5$ les figures (5.25)(5.26) montrent que le passage de $U_{max}[i]=10$ à $U_{max}[i]=20$ ne réalise pas un grand changement sur les performances. Tandis que nous obtenons des commandes appréciables pour $N2[1]=5$ $N2[2]=1$ $U_{max}[1]=20$ $U_{max}[2]=10$, figure (5.27) d'où la nécessité, dans quelque cas, d'assigner des valeurs différentes des paramètres pour chaque bras. L'introduction d'une pondération $\lambda_1=7.10^{-8}$ $\lambda_2=10^{-5}$ sur la commande oblige une poursuite oscillatoire de la trajectoire avec une limitation de $U_{max}[i]=50$, figures (5.28)(5.29) respectivement pour:

$$\begin{cases} N2[i]=10 \\ NU[i]=5 \end{cases} \quad \begin{cases} N2[i]=10 \\ NU[1]=10 \quad NU[2]=5 \end{cases}$$

La poursuite d'une trajectoire cercle est présenté sur la figure (5.30), dans le cas de l'évolution des variables généralisées et les figures (5.30.a)(5.30.b) concernant la poursuite dans l'espace cartésien respectivement pour le premier et deuxième tour. ceci a été réalisé dans les conditions initiales suivantes: $x[1]=0.295$ $x[3]=0.676$.

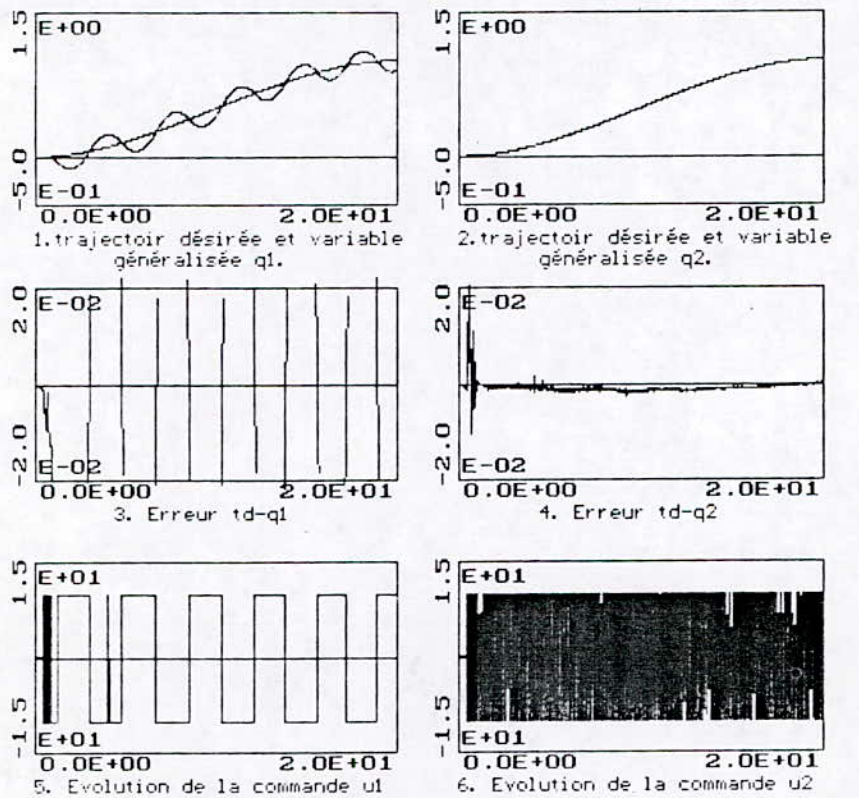


Figure 5.22 trajectoir désirée (td), position (q1,q2) pour le robot SCARA pour $NI[1]=1$ $N2[1]=1$ $NU[1]=1$ $NI[2]=1$ $N2[2]=1$ $NU[2]=1$ $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$.

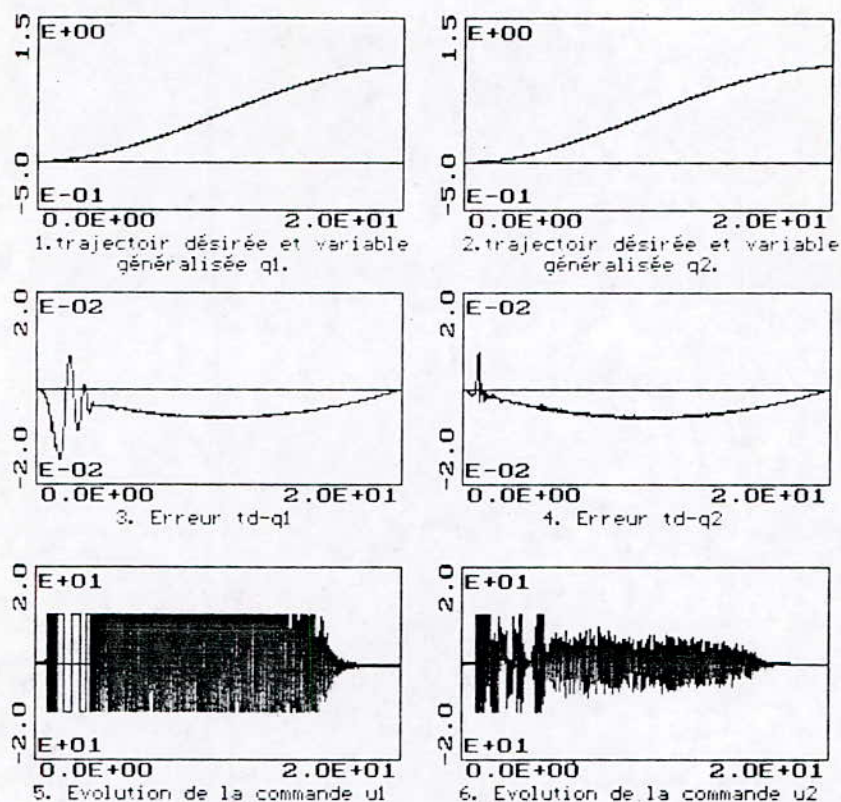


Figure 5.23 trajectoire désirée (td), position (q1, q2) pour le robot SCARA pour $N1[1]=1$ $N2[1]=5$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=5$ $NUC[2]=1$ $\lambda_1=0$ $\lambda_2=0$ $Umax[1]=10$ $Umax[2]=10$.

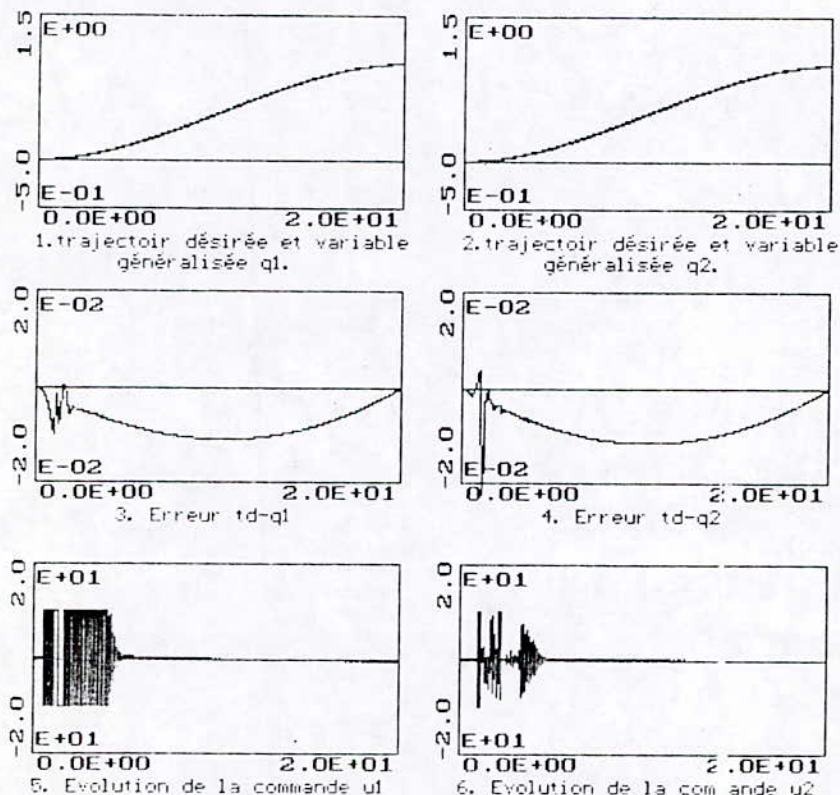


Figure 5.24 trajectoire désirée (td), position (q1, q2) pour le robot SCARA pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=1$ $\lambda_1=0$ $\lambda_2=0$ $Umax[1]=10$ $Umax[2]=10$.

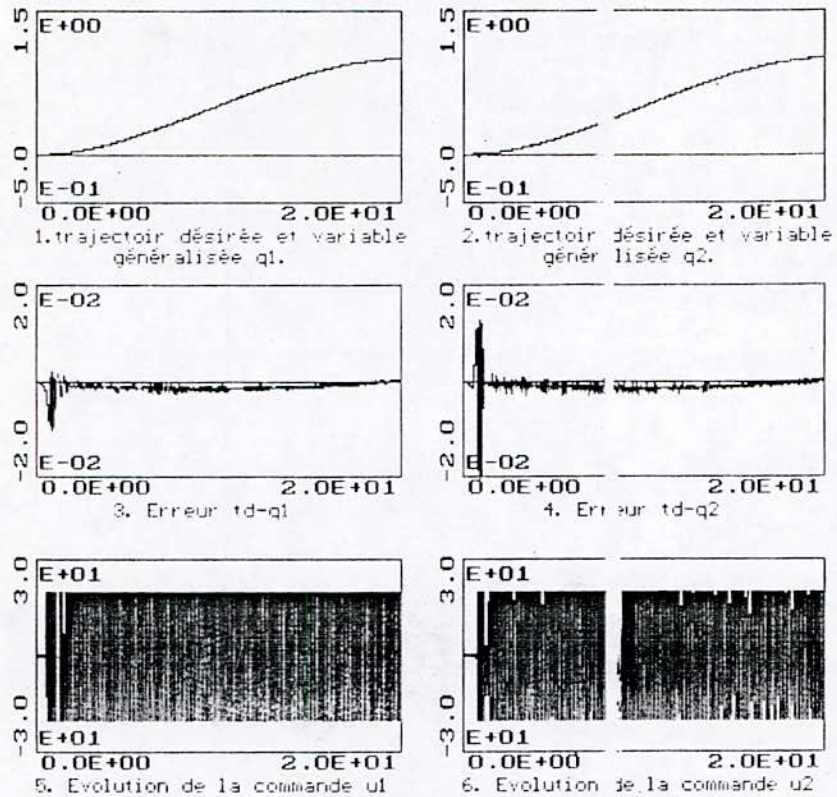


Figure 5.25 trajectoire désirée (td), position (q_1, q_2) pour le robot SCARA pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=5$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=5$ $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=20$ $U_{max}[2]=20$.

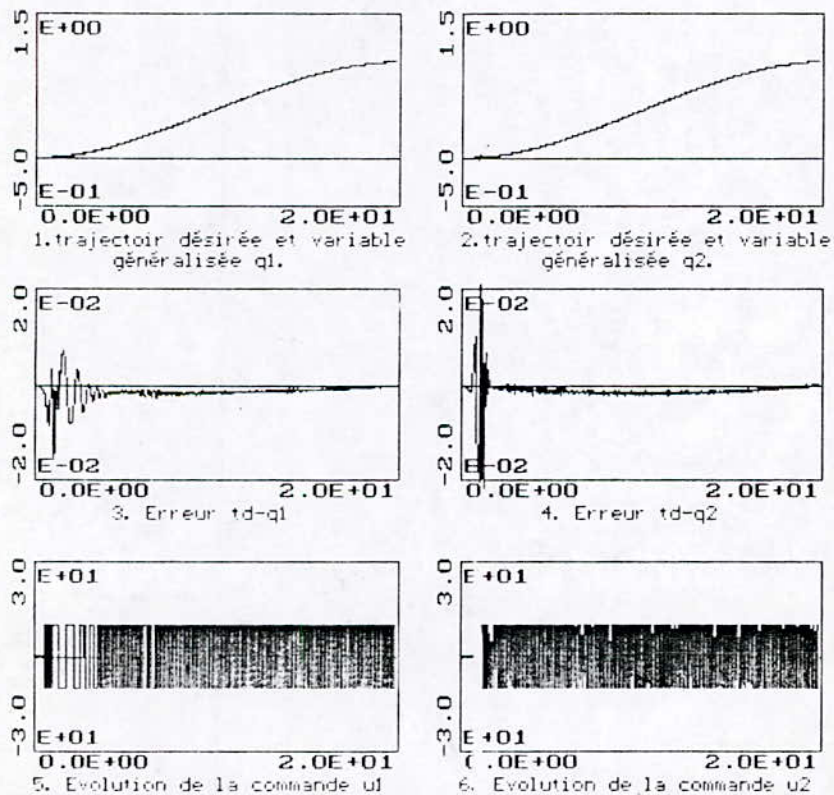


Figure 5.26 trajectoire désirée (td), position (q_1, q_2) pour le robot SCARA pour $N1[1]=1$ $N2[1]=10$ $NUC[1]=5$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=5$ $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=10$ $U_{max}[2]=10$.

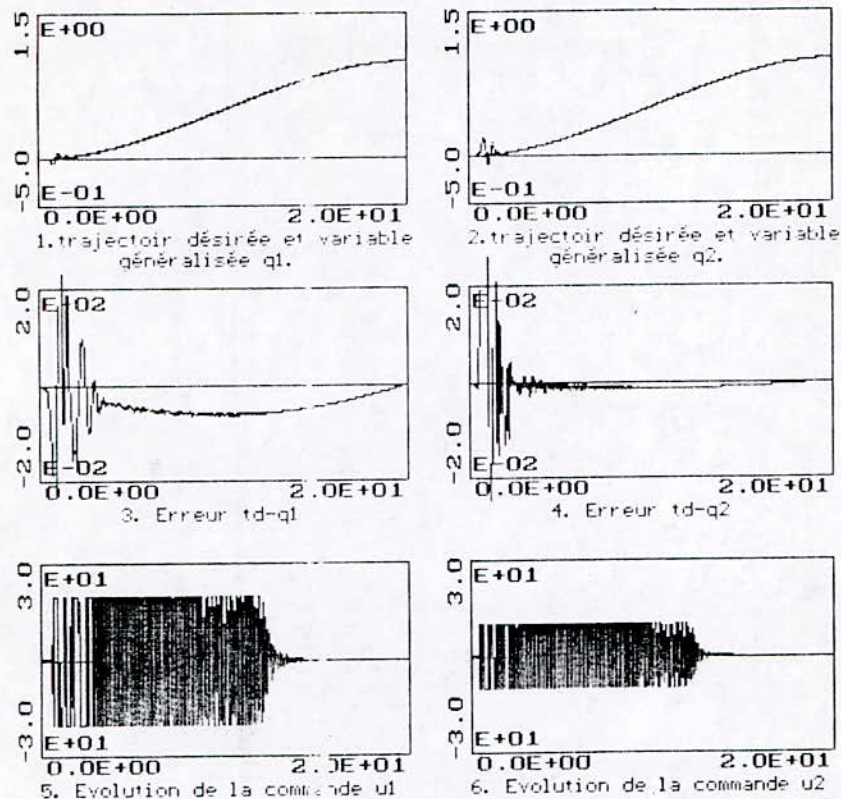


Figure 5.27 trajectoire désirée (td), position (q1,q2) pour le robot SCARA pour $NIC[1]=42[1]=5$ $NUC[1]=1$ $NIC[2]=1$ $N2[2]=1$ $NUC[2]=1$ $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=20$ $U_{max}[2]=10$.

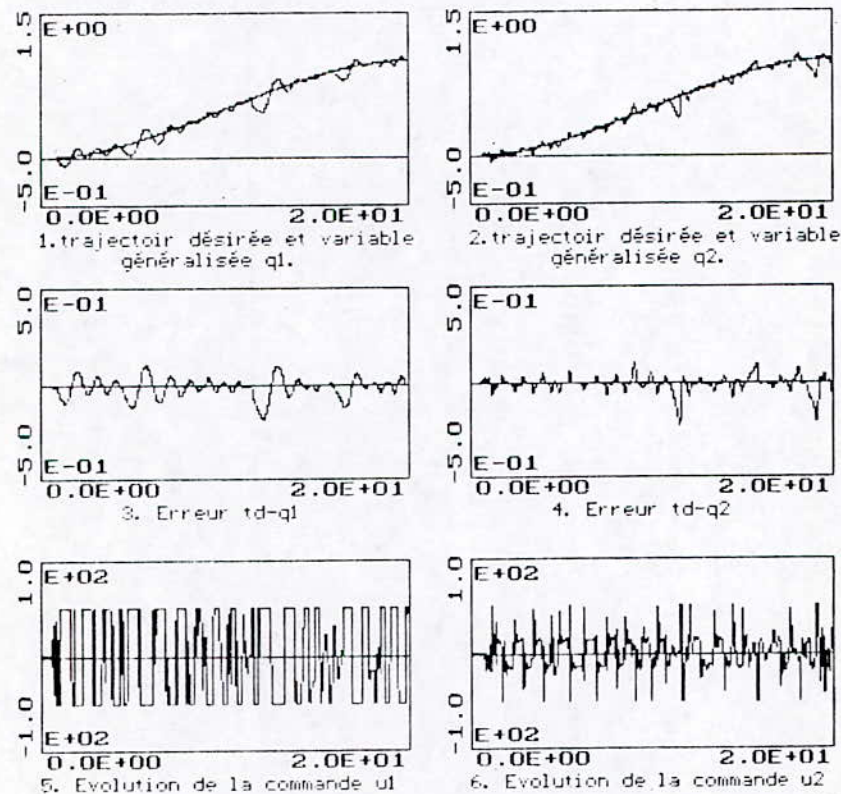


Figure 5.28 trajectoire désirée (td), position (q1,q2) pour le robot SCARA pour $NIC[1]=1$ $N2[1]=10$ $NUC[1]=5$ $NIC[2]=1$ $N2[2]=10$ $NUC[2]=5$ $\lambda_1=0.00000007$ $\lambda_2=0.00001$ $U_{max}[1]=50$ $U_{max}[2]=50$.

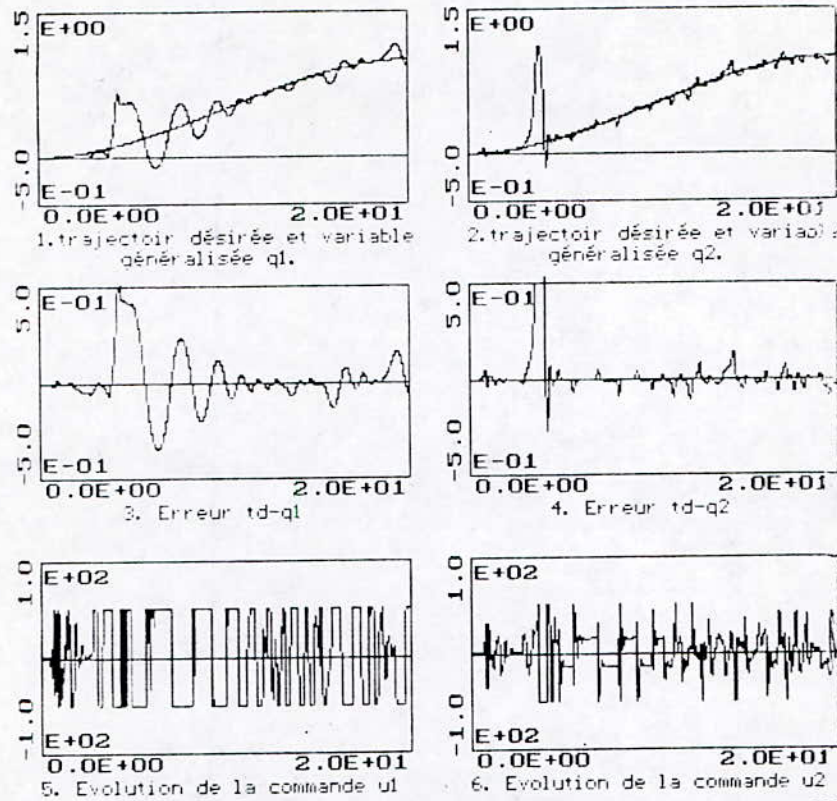


Figure 5.29 trajectoire désirée (td), position (q_1, q_2) pour le robot SCARA pour $NI[1]=1$ $N2[1]=10$ $NU[1]=10$ $NI[2]=1$ $N2[2]=10$ $NU[2]=5$ $\lambda_1=0.000000007$ $\lambda_2=0.00001$ $U_{max}[1]=50$ $U_{max}[2]=50$.

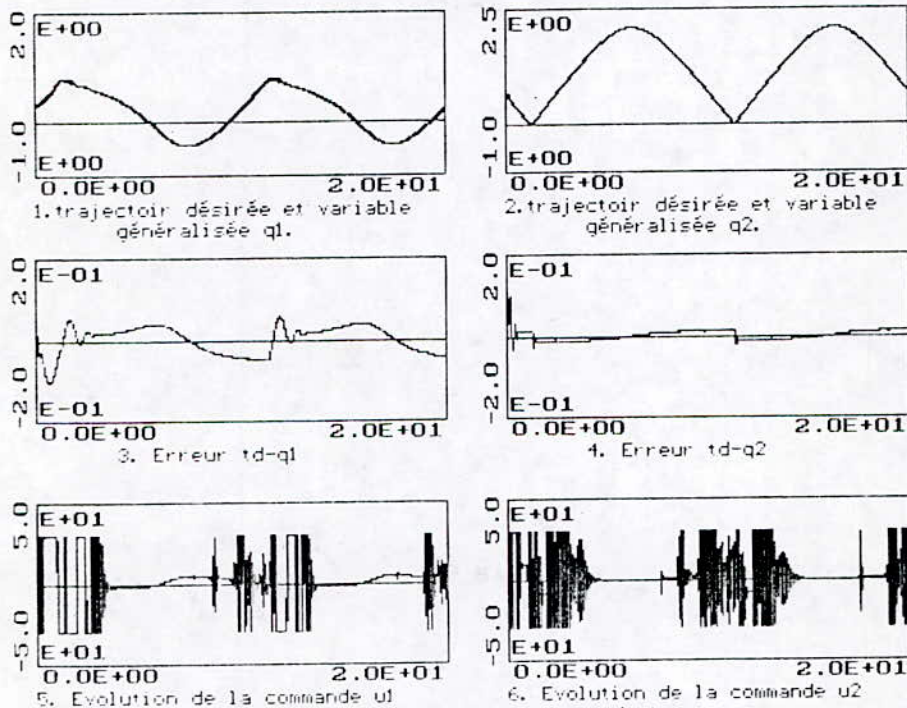


Figure 5.30 trajectoire désirée cercle (td), position (q_1, q_2) pour le robot SCARA pour $NI[1]=1$ $N2[1]=7$ $NU[1]=1$ $NI[2]=1$ $N2[2]=10$ $NU[2]=5$ $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=30$ $U_{max}[2]=30$.

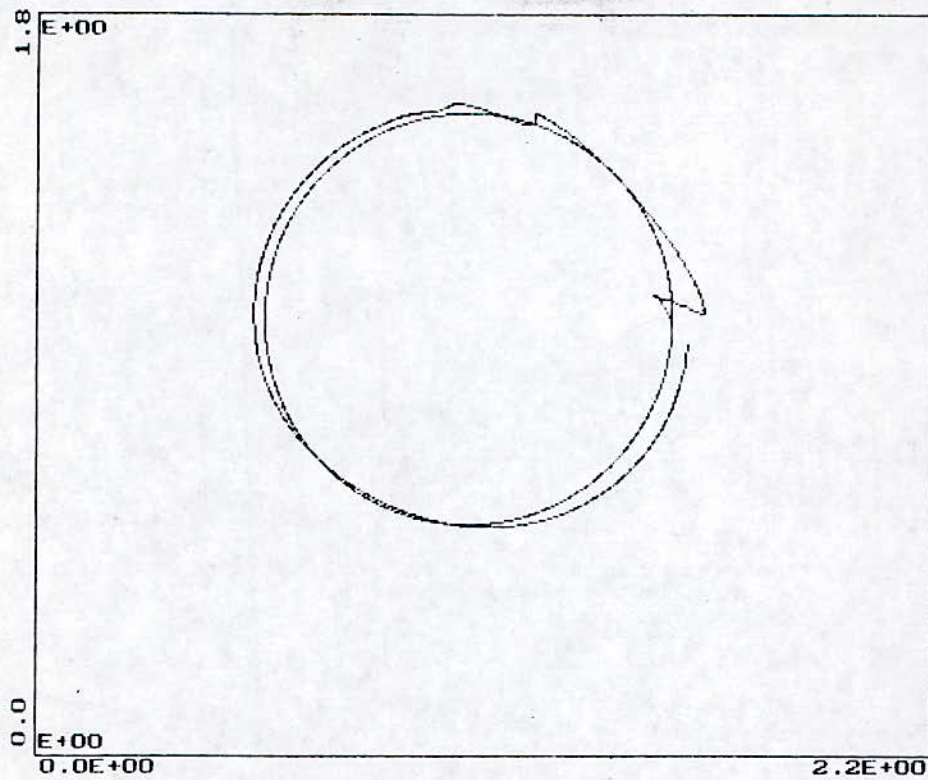


Figure 5.30.a Evolution du robot SCARA dans l'espace cartésien poursuivant une trajectoire cercle pour $N1[1]=1$ $N2[1]=7$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=5$ premier tour $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=30$ $U_{max}[2]=30$ $x[1]=0.295$ $x[3]=0.676$.

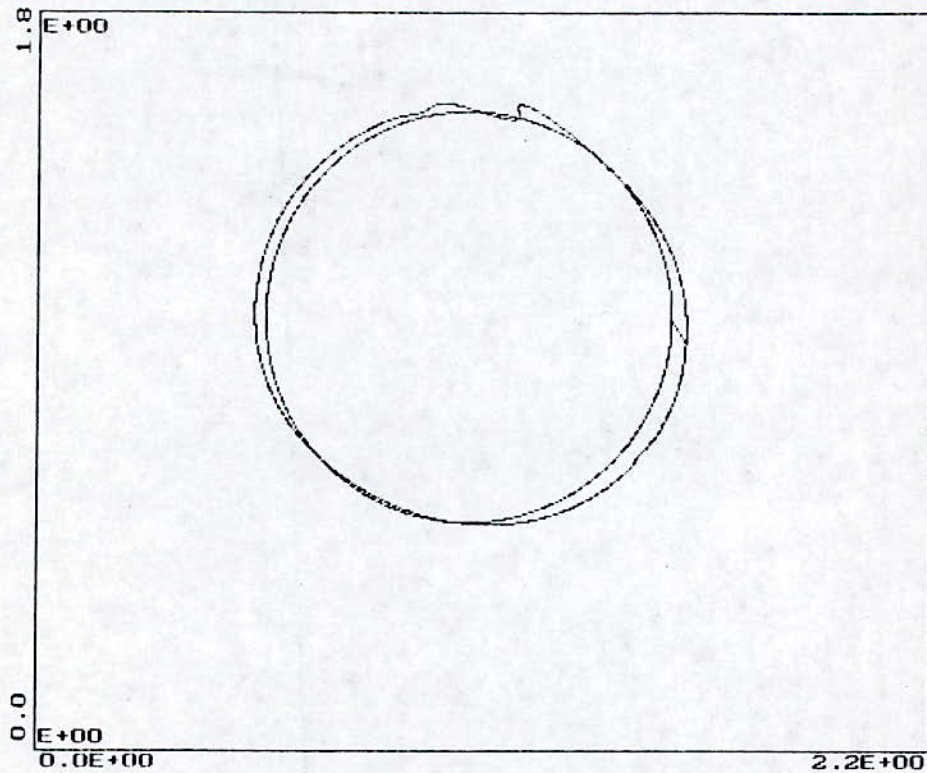


Figure 5.30.b Evolution du robot SCARA dans l'espace cartésien poursuivant une trajectoire cercle pour $N1[1]=1$ $N2[1]=7$ $NUC[1]=1$ $N1[2]=1$ $N2[2]=10$ $NUC[2]=5$ 2^{ème} tour $\lambda_1=0$ $\lambda_2=0$ $U_{max}[1]=30$ $U_{max}[2]=30$ $x[1]=0.295$ $x[3]=0.676$.

Conclusion.

Ce chapitre a été consacré à l'application de la loi de commande prédictive généralisées à deux types de robot manipulateurs. En effet, le premier type est le Robot PUMA qui évolue dans un espace opérationnel à trois dimensions, tandis que le deuxième est le type SCARA qui se déplace dans un plan horizontal, d'où l'effet gravitationnel qui n'intervient pas.

Après un bref aperçu sur les différents travaux effectués jusqu'à présent par plusieurs chercheurs dans le domaine multivariable, nous décrivons le modèle de représentation échantillonné multivariable à partir d'un modèle sous forme d'état, nonlinéaire et couplé.

Les caractéristiques nonlinéaires des deux robots démontrées lors de l'analyse en boucle ouverte, ont montré la nécessité d'une adaptation en temps réel des lois de commande linéaire monovariable, en faisant un bon choix du modèle de représentation. Le choix d'une structure de modèle de représentation de type Diagonal-Diagonal en entrée et sortie, a permis l'implémentation directe des algorithmes d'identification et de commande monovariables. L'effet des nonlinéarités et du couplage du modèle de connaissance sont introduits dans le modèle de perturbation.

L'algorithme d'identification choisi est celui des MCR Étendus à trace constante, sachant qu'il est le mieux adapté pour l'identification des paramètres variant dans le temps.

L'extension de l'algorithme à GPC au cas multivariable et de son application aux deux types de robots manipulateurs ne pose aucun problème vu le choix du modèle de représentation.

Il est clair, d'après les résultats obtenus, qu'une valeur supérieure ou égale à cinq de l'horizon de prédiction est nécessaire. Nous tenons à signaler qu'un compromis existe entre l'horizon de commande et la limitation de commande. En effet, l'augmentation de NU rend les commandes plus actives d'où la nécessité d'introduire une limitation à une valeur supérieure autrement on peut observer une saturation. Les coefficients de pondération supérieurs, ou égaux à un ou légèrement inférieurs à un, rendent les commandes insuffisantes à subvenir au besoin des robots pour la poursuite, d'où la nécessité d'assigner à λ_i des valeurs très petites par rapport à un où 'carrément' égales à zéro.

Enfin nous pouvons conclure qu'une bonne poursuite des trajectoires désirées pour le SCARA et le PUMA ne peut être réalisée que par un choix adéquat des paramètres de synthèse de la GPC.

*CONCLUSION
GENERALE*

"Tout est dit ... tout reste à faire".

LUDWIG VAN BEETHOVEN

CONCLUSION GENERALE.

Dans ce mémoire, on s'est intéressé à l'étude des performances de la commande prédictive généralisée. L'étude par simulation de quatre types de procédé à paramètres connus, présentant chacun une particularité spécifique, a montré que l'algorithme peut faire face aux problèmes de commande des systèmes instables et à phase non-minimale. Ceci grâce aux paramètres de synthèse de la GPC, qui constituent la puissance de l'algorithme.

Par un choix aléatoire de ces paramètres de synthèse, on aboutit à un placement de pôles aléatoire, ceci veut dire qu'on ne peut pas spécifier d'avance une dynamique de sortie vu qu'il n'y a pas de lien concret entre les paramètres de synthèse et l'équation caractéristique.

Dans le but de pouvoir gouverner les racines de l'équation caractéristique et d'imposer par la même occasion au système une dynamique désirée, une extension de l'algorithme s'impose. Cette extension se traduit par l'adjonction dans le critère de commande des paramètres supplémentaire facilitant ainsi l'analyse en boucle fermée. Pour un choix spécifique de ces paramètres la GPC peut se ramener à quelques algorithmes comme celui à placement de pôles, à variance minimale généralisée et à horizon étendu de Ydstie.

Les résultats de simulation ont été réalisés sur des systèmes à paramètres connus avec une variation du retard d'une unité. La détérioration des performances dans le cas de variations paramétriques imposent l'introduction d'un outil intermédiaire entre le système et le calcul de la commande.

Cet outil, qui est l'identification, consiste à identifier directement les paramètres du régulateur, on aboutit alors à l'approche directe de la commande adaptative et les paramètres du système pour le calcul du régulateur dans l'approche indirecte. Dans les deux approches la GPC adaptative, utilisant un algorithme d'identification à trace constante, offre la possibilité d'une bonne poursuite du modèle et des paramètres. Dans le cas de la variation du retard, même très importante, une surestimation de l'ordre du modèle s'impose. Les résultats obtenus s'avèrent très satisfaisants, surtout aux problèmes de sur et sousparamétrisation.

L'application de la GPC aux de deux robots manipulateurs a montré que malgré la dynamique imposée par les deux modèles en boucle ouverte, de très bons résultats ont été obtenus et ceci par un choix spécifique des paramètres de synthèse.

Un autre paramètre paraît très important pour l'obtention d'une bonne poursuite de la trajectoire, c'est la limitation de la commande. Cette limitation est introduite pour réduire les efforts de commande et de la maintenir dans une amplitude fixe. Une très grande ou très petite limitation risque de dévier au manipulateur sa trajectoire. Cette limitation est en relation directe avec le type d'actionneur utilisé. Donc pour un robot réels, il faut prendre en considération la puissance nominale des actionneurs, pour pouvoir simulé réellement la commande du robots.

Dans le but d'obtenir de très bon résultats de simulation soit dans le cas

adaptatif ou non adaptatif, une attention particulière doit être donnée au choix des paramètres de synthèse, puisqu'ils représentent la puissance de la GPC.

La recherche de ces paramètres est une tâche pénible. L'utilisation d'un algorithme, se basant sur un critère optimal, permet de trouver systématiquement les paramètres optimaux. La spécification du critère et la méthode d'optimisation offrent de nouvelles voies pour la synthèse d'un algorithme à GPC dont l'utilisateur spécifie un double critère, celui de la commande et un autre pour les paramètres de synthèse.

Une étude des performances de la GPC a été présentée dans ce travail. L'introduction de nouvelle pondération, dans le critère de la commande, permet de spécifier la dynamique en boucle fermée. Cette dynamique affecte directement la sortie du processus.

L'utilisation d'un modèle sur l'état partiel, permet de spécifier deux dynamiques, celle de la sortie et celle de la commande. Ainsi on formule le critère de commande d'une manière particulière. On aboutit alors à la commande prédictive généralisée à double modèle de référence en spécifiant la dynamique de la commande et de la sortie (Partial State Reference Model). Ceci peut faire l'objet d'un autre thème d'étude.

Le modèle de représentation utilisé par la synthèse de la commande, est linéaire. L'utilisation d'un autre type de modèle s'avère une idée intéressante. Le modèle Hamerstein est un modèle pouvant représenter des nonlinéarités du type autre que du premier ordre. Le problème qui se pose alors est la méthode de prédiction d'un modèle nonlinéaire et l'utilisation de ce modèle pour synthétiser la commande GPC. En effet, le couplage d'un modèle nonlinéaire tel que d'Hamerstein, et le critère de commande GPC, aboutit à un axe de recherche dont on doit trouver des équations équivalentes.

ANNEXES

"Autant j'apprends, autant je deviens bête".

VICTOR HUGO

Annexe 1

ANALYSE SUR LE CHOIX DES PARAMETRES DE SYNTHESE (N1 N2 NU λ)

Dans le chapitre un nous avons montré qu'il est très difficile de faire une analyse détaillée sur l'étude de la stabilité du système en boucle fermée. En effet, cette étude met en évidence l'équation caractéristique du système qui est fonction des polynômes R , S et T de la structure RST du régulateur. L'étude est délicate vu la complexité de cette équation. Cette complexité augmente au fur et à mesure qu'on change les paramètres de synthèse de la GPC. Cette annexe examine l'effet des paramètres de la GPC et du retard du système sur la stabilité de la sortie en boucle fermée.

Soit le système écrit sous la forme suivante:

$$y(t) = \frac{B}{A} u(t-1)$$

avec $A = 1 + a_1 q^{-1} + \dots + a_n q^{-n}$
 $B = b_0 + b_1 q^{-1} + \dots + b_m q^{-m}$

Sachant que la résolution de l'équation Diophantine (1.9) donne:

$$E_j = 1 + e_1 q^{-1} + \dots + e_{j-1} q^{-(j-1)}$$
$$F_j = f_{0,j} + f_{1,j} q^{-1} + \dots + f_{n,j} q^{-n}$$

Suivant la valeur de $N1$, $N2$ et NU la matrice G est:

$$G = \begin{bmatrix} g_0 & 0 & 0 & \dots & 0 \\ g_1 & g_0 & 0 & \dots & 0 \\ g_2 & g_1 & g_0 & \dots & 0 \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ g_{N2-1} & g_{N2-2} & \cdot & \dots & g_0 \end{bmatrix} \quad \text{pour } N1=1 \quad NU=N2 \quad \text{et } N2 \text{ quelconque}$$

$$G = \begin{bmatrix} g_{N1-1} & 0 & 0 & \dots & 0 \\ g_{N1} & g_{N1-1} & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{N2-1} & g_{N2-2} & \cdot & \cdot & \cdot & g_{N1-1} \end{bmatrix} \quad \text{pour } NU=N2, \quad N2 \text{ et } N1 \text{ quelconques}$$

$$G = \begin{bmatrix} g_{N1-1} & g_{N1-2} & \cdot & \cdot & \cdot & 0 \\ g_{N1} & g_{N1-1} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ g_{N2-1} & g_{N2-2} & \cdot & \cdot & \cdot & g_{N2-NU} \end{bmatrix} \quad \text{pour } N1 \quad N2 \text{ et } NU \text{ quelconques}$$

Le remplissage de cette matrice pour différentes valeurs de N1 N2 et NU est:

$$\text{pour } N1=N2=NU=1 \quad G = [g_0]$$

$$\text{pour } N1=1 \quad N2=2 \quad NU=1 \quad G = \begin{bmatrix} b_0 \\ b_1 + b_0 e_1 \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \end{bmatrix}$$

$$\text{pour } N1=1 \quad N2=2 \quad NU=2 \quad G = \begin{bmatrix} b_0 & 0 \\ b_1 + b_0 e_1 & b_0 \end{bmatrix} = \begin{bmatrix} g_0 & 0 \\ g_1 & g_0 \end{bmatrix}$$

$$\text{pour } N1=N2=2 \quad NU=1 \quad G = [g_1]$$

$$\text{pour } N1=N2=NU=2 \quad G = [g_0]$$

1.1 Effet de NU.

Si $N1=1 \quad N2=2$ et $NU=2$ on a:

$$G = \begin{bmatrix} g_0 & 0 \\ g_1 & g_0 \end{bmatrix}$$

Le calcul de la commande aboutit à l'expression suivante:

$$\Delta u = (G^T G + \lambda I)^{-1} G^T (W - f)$$

Ann1.1

avec $(G^T G + \lambda I)^{-1} = \begin{bmatrix} g_0^2 + g_1^2 + \lambda & g_0 g_1 \\ g_0 g_1 & g_0^2 + \lambda \end{bmatrix}^{-1}$ Ann1.2

si $\lambda=0$ et $b_0=0$ cette matrice est non-inversible d'où la choix de $NU=1$.

1.2 Effet de N2 avec $NU=1$.

Cas où $N1=N2=1$.

On a $G = [g_0]$ donc $\Delta u(t) = \frac{g_0}{g_0^2 + \lambda} [w - f(t+1)]$ Ann1.3

les polynômes R, S et T sont obtenus à partir de la forme suivante:

$$(g_0^2 + \lambda + q^{-1} g_{fin}) \Delta u(t) = g_0 [w - F_1 y(t)]$$
 Ann1.4

avec g_{fin} sont les derniers élément du polynôme G_j

et $\begin{cases} R=F \\ S=g_0^2 + \lambda + q^{-1} g_{fin} \\ T=g_0 \end{cases}$

le système en boucle fermée donne:

$$H_{BF} = \frac{BT}{SA\Delta + q^{-1}BR}$$
 Ann1.5

Le polynôme caractéristique est:

$$P_c = SA\Delta + q^{-1}BR = Bg_0 + \lambda A\Delta$$
 Ann1.6

d'où $y_{BF} = \frac{Bg_0}{g_0 B + \lambda A\Delta} w$ Ann1.7

si $\lambda=0$ et le système est à phase non-minimale l'équation caractéristique du système en BF est instable.

Cas où $N1=1$ $N2=2$.

L'expression de la commande devient:

$$\Delta u(t) = \frac{1}{g_0^2 + g_1^2 + \lambda} (g_0 \ g_1) \begin{pmatrix} w-f(t+1) \\ w-f(t+2) \end{pmatrix} \quad \text{Ann1.8}$$

on obtient l'expression du polynôme R,S et T suivant:

$$\begin{cases} R = g_0 F_1 + g_1 F_1 \\ S = g_0^2 + g_1^2 + \lambda + q^{-1} (g_0 g_{fin1} + g_1 g_{fin2}) \\ T = g_0 + g_1 \end{cases}$$

L'équation caractéristique est:

$$P_c = SA\Delta + q^{-1}BR = B(g_0 q^{-1} + g_1) - g_1 g_0 A\Delta + q^{-1} \lambda A\Delta \quad \text{Ann1.9}$$

$$\text{d'où } y_{BF} = \frac{q^{-1} B(g_0 + g_1)}{B(g_0 q^{-1} + g_1) - g_0 g_1 A\Delta + q^{-1} \lambda A\Delta} w \quad \text{Ann1.10}$$

De cette manière on peut faire l'étude de la stabilité, pour différent choix de λ.

Cas où $N1=1 \ N2=3$.

La forme de la commande est:

$$\Delta u(t) = \frac{1}{g_0^2 + g_1^2 + g_2^2 + \lambda} (g_0 \ g_1 \ g_2) \begin{pmatrix} w-f(t+1) \\ w-f(t+2) \\ w-f(t+3) \end{pmatrix} \quad \text{Ann1.11}$$

dans les polynômes R,S et T sont:

$$\begin{cases} R = g_0 F_1 + g_1 F_1 + g_2 F_3 \\ S = g_0^2 + g_1^2 + g_2^2 + \lambda + q^{-1} (g_0 g_{fin1} + g_1 g_{fin2} + g_2 g_{fin3}) \\ T = g_0 + g_1 + g_2 \end{cases} \quad \text{Ann1.12}$$

$$P_c = B(g_0 q^{-2} + g_1 q^{-1} + g_2) - g_0 g_1 A\Delta - g_2 (g_0 + g_1 q^{-1} A\Delta) + q^{-2} \lambda A\Delta \quad \text{Ann1.13}$$

$$\text{d'où } y_{BF} = \frac{q^{-2} B(g_0 + g_1 + g_2)}{P_c} w \quad \text{Ann1.14}$$

Pour avoir une forme générale de la fonction de transfert en BF nous examinons le cas $N2=4$. Tel que la commande, l'équation caractéristique et la sortie en BF sont respectivement:

$$\Delta u(t) = \frac{1}{\left[\sum_{i=0}^3 g_i^2 + \lambda \right]} (g_0 \ g_1 \ g_2 \ g_3) \begin{pmatrix} w-f(t+1) \\ w-f(t+2) \\ w-f(t+3) \\ w-f(t-4) \end{pmatrix} \quad \text{Ann1.15}$$

$$P_c = B(g_0 q^{-3} + g_1 q^{-2} + g_2 q^{-1} + g_3) - g_0 g_1 A \Delta - g_2 (g_0 + g_1 q^{-1} A \Delta) - g_3 A \Delta (g_0 + g_1 q^{-1} + g_2 q^{-2}) + q^{-3} \lambda A \Delta \quad \text{Ann1.16}$$

$$y_{BF} = \frac{q^{-3} B (g_0 + g_1 + g_2 + g_3)}{P_c} w$$

Sous une forme générale la fonction de transfert du système en BF est:

$$y_{BF} = \frac{q^{-(N2-1)} B \sum_{i=1}^{N2} g_{i-1}}{B \left[\sum_{i=1}^{N2} g_{i-1} q^{-(N2-1)} \right] - \left[\sum_{i=1}^{N2} g_i \sum_{k=1}^i q^{-(k-1)} g_{k-1} \right] A \Delta + \lambda A \Delta q^{-(N2-1)}} w \quad \text{Ann1.17}$$

1.3 Effet de N1.

cas où $N1=2$ $N2=2$ $NU=1$.

la matrice $(G^T G + \lambda I)^{-1} G^T = (g_1^2 + \lambda)^{-1} g_1$

La commande s'écrit:

$$\Delta u(t) = \frac{g_1}{g_1^2 + \lambda} [w - f(t+2)] \quad \text{Ann1.18}$$

avec $f(t+2) = g_1 F_2 y(t)$

donc:

$$\begin{cases} R = g_1 F_2 \\ S = g_1^2 + \lambda + q^{-1} g_1 g_{fin2} \\ T = g_1 \end{cases} \quad \text{Ann1.19}$$

L'équation caractéristique donne:

$$P_c = S A \Delta + q^{-1} B R = B g_1 - g_1 g_0 A \Delta + q^{-1} \lambda A \Delta \quad \text{Ann1.20}$$

$$\text{La sortie est } y_{BF} = \frac{Bg_1}{P_c} w$$

Ann1.21

cas où $N1=3$ $N2=3$ $NU=1$.

l'expression de la commande:

$$(g_2^2 + \lambda) \Delta u(t) = g_2 w - g_2 f(t+3)$$

Ann1.22

la sortie en BF:

$$y_{BF} = \frac{g_2 q^{-2} B}{g_2 [B - (g_0 + g_1 q^{-1}) A \Delta] + q^{-2} \lambda A \Delta} w$$

Ann1.23

Sous une forme plus générale dans le cas où $N1=N2 \geq \text{retard}$ la fonction de transfert est:

$$y_{BF} = \frac{g_{N2-1} B}{g_{N2-1} \left[B - A \Delta \sum_{i=1}^{N2} q^{-(i-2)} g_{i-2} \right] + q^{-(N2-1)} \lambda A \Delta} w$$

Ann1.24

1.4 Effet du retard.

Dans le cas où $N1=N2=2$ on introduit un retard de un dans la système, c'est

à dire $b_0=0$.

Sachant que:

$$y_{BF} = \frac{q^{-1} g_1 B}{g_1 (B - g_0 A \Delta) + q^{-1} \lambda A \Delta}$$

Ann1.25

et $G_1 = g_0 + q^{-1} g_{fin1} = E_1 B$ donc $g_0=0$

$$\text{et } y_{BF} = \frac{g_1 B}{g_1 B + q^{-1} \lambda A \Delta} w$$

Ann1.26

Si $\lambda=0$ et le système est à phase non-minimale la sortie du système est instable. Par contre si $N1=N2=3$ en gardant le même retard la fonction de transfert en BF donne:

$$y_{BF} = q^{-1} \frac{g_2 B}{g_2 (B - g_1 q^{-1} \lambda A \Delta)} w$$

Ann1.27

qui peut être stable suivant la position des pôles de l'équation caractéristique.

Ainsi, on a présenté l'effet du choix de $N1, N2$ et NU sur la polynôme caractérisant la dynamique en boucle fermée. Cette analyse permet de choisir ces paramètres, en fonction de la complexité de calcul, pour permettre de choisir l'outil de calcul dans le cas d'une implémentation pratique.

Annexe 2

CALCUL DU MODELE DYNAMIQUE DU ROBOT PAR LE FORMALISME D'EULER-LAGRANGE.

Les matrices de transformations sont:

$${}^0T^1 = \begin{bmatrix} C_1 & -S_1 & 0 & lC_1 \\ S_1 & C_1 & 0 & lS_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ; \quad {}^1T^2 = \begin{bmatrix} C_2 & -S_2 & 0 & lC_1 \\ S_2 & C_2 & 0 & lS_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} ;$$

$${}^0T^2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & l(C_{12}+C_1) \\ S_{12} & C_{12} & 0 & l(S_{12}+S_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} ;$$

avec $Q_j = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} ;$

On obtient les matrices U_{ij} en utilisant:

$$U_{ij} = \begin{cases} {}^0T^{j-1}Q_j \quad {}^{j-1}T^i & j \leq i \\ 0 & j > i \end{cases}$$

on obtient alors $U_{11} = \begin{bmatrix} -S_1 & -C_1 & 0 & -lS_1 \\ C_1 & -S_1 & 0 & lC_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} ;$

$$U_{21} = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -l(S_{12}+S_1) \\ C_{12} & -S_{12} & 0 & l(C_{12}+C_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} ; \quad U_{22} = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -lS_{12} \\ C_{12} & -S_{12} & 0 & lC_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} ;$$

$$\tau_i = \frac{d}{dt} \left(\frac{\delta L}{\delta \dot{q}_i} \right) - \left(\frac{\delta L}{\delta q_i} \right) \text{ avec } L \text{ établi est le Lagrangien.}$$

$$\tau_i = \sum_{j=1}^2 \sum_{k=1}^j \text{tr}(U_{jk} I_j U_{ji}^T) \ddot{q}_k + \sum_{j=1}^2 \sum_{k=1}^j \sum_{m=1}^j \text{tr}(U_{jkm} I_j U_{ji}^T) \dot{q}_k \dot{q}_m - \sum_{j=1}^2 m_j g U_{ji} j r^j$$

en mettant $\tau_i = D(q(t)) \ddot{q}(t) + h(q(t), \dot{q}(t)) + c(q(t))$
on obtient τ_i sous la forme:

$$\tau_i = \sum_{j=1}^2 D_{ik} \ddot{q}_k + \sum_{k=1}^2 \sum_{m=1}^2 h_{ikm} \dot{q}_k \dot{q}_m + c_i$$

Sachant que:

$$U_{ijk} = \begin{cases} {}_0 T^{k-1} Q_{k-1} T^{j-1} Q_j j^{-1} T^i & ; k \leq j \leq i. \\ {}_0 T^{j-1} Q_j j^{-1} T^{k-1} Q_k k^{-1} T^i & ; j \leq k \leq i. \\ 0 & ; j \leq i \leq k. \end{cases}$$

on obtient:

$$\begin{cases} D_{11} = \text{tr}(U_{11} I_1 U_{11}^T) + \text{tr}(U_{21} I_2 U_{21}^T) = \frac{1}{3} m_1 l^2 + \frac{4}{3} m_2 l^2 + m_2 C_2 l^2 \\ D_{12} = D_{21} = \text{tr}(U_{22} I_2 U_{21}^T) = \frac{1}{3} m_2 l^2 + \frac{1}{2} m_2 l^2 C_2 \\ D_{22} = \text{tr}(U_{22} I_2 U_{22}^T) = \frac{1}{3} m_2 l^2 \end{cases}$$

$$\begin{cases} h_1 = \sum_{k=1}^2 \sum_{m=1}^2 h_{1km} \dot{\theta}_k \dot{\theta}_m = -\frac{1}{2} m_2 S_2 l^2 \dot{\theta}_2^2 - m_2 S_2 l^2 \dot{\theta}_1 \dot{\theta}_2 \\ h_2 = \sum_{k=1}^2 \sum_{m=1}^2 h_{2km} \dot{\theta}_k \dot{\theta}_m = \frac{1}{2} m_2 S_2 l^2 \dot{\theta}_1^2 \\ h_{ikm} = \sum_{j=\max(1, k, m)}^2 \text{tr}(U_{jkm} I_j U_{ji}^T) \end{cases}$$

$$\begin{cases} c_1 = -(m_1 g U_{11} {}_1 r^1 + m_2 g U_{21} {}_2 r^2) \\ c_2 = -m_2 \left(\frac{1}{2} g l C_{12} - g l C_{12} \right) \\ \text{avec } {}_1 r^1 = {}_2 r^2 = \begin{bmatrix} -\frac{1}{2} & 0 & 0 & 1 \end{bmatrix} \end{cases}$$

On obtient donc le même résultats que celui (§ 4.5.1)

$$\begin{aligned}
 \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} &= \begin{bmatrix} \frac{1}{3}m_1l^2 + \frac{4}{3}m_2l^2 + m_2C_2l^2 & \frac{1}{3}m_2l^2 + \frac{1}{2}m_2l^2C_2 \\ \frac{1}{3}m_2l^2 + \frac{1}{2}m_2l^2C_2 & \frac{1}{3}m_2l^2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \\
 &+ \begin{bmatrix} -\frac{1}{2}m_2S_2l^2\theta_2^2 - m_2S_2l^2\theta_1\theta_2 \\ \frac{1}{2}m_2S_2l^2\theta_1^2 \end{bmatrix} \\
 &+ \begin{bmatrix} \frac{1}{2}m_1glC_1 + \frac{1}{2}m_2glC_{12} + m_2glC_1 \\ \frac{1}{2}m_2glC_{12} \end{bmatrix}
 \end{aligned}$$

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] J.Bordeneuve, "Commande prédictive adaptative application à des procédés thermique" Thèse de doctorat, UPS, Toulous, N°797, 1990.
- [2] D.W.Clarke & al, "Generalized predictive control-Part I: the basic algorithm. Part II: Extensions and interpretation" *Automatica*, Vol.23, N°2, pp 137-160, 1987.
- [3] K.J.Aström, "Introduction to stochastic control theory" Academic press, 1970.
- [4] K.W.Lim & K.V.Ling, "Generalized predictive control of a heat exchanger" *IEEE, C.S.M.*, pp 9-12, 1989.
- [5] D.W.Clarke, "Implementation of self tuning controllers" H.Nicholson & B.H.Swanick, "Self tuning and adaptive control: teory and applications" *IEE, C.E.S.*, 15, 1981.
- [6] D.W.Clarke, "Self tuning control of nonminimum phase systems" *Automatica*, Vol.20, N°5, pp 501-517, 1984.
- [7] D.W.Clarke & al, "Self tuning controller" *Proc. IEE*, Vol.122, N°9, pp929-934, 1975.
- [8] D.W.Clarke & P.J.Gawthrop, "Self tuning cotrol" *Proc. IEE*, Vol. 126, N°6, pp 633- 640,1979.
- [9] R.R.Bitmead & al, "Adaptive optimal control: The thinking Man's GPC" Prentice Hall, 1990.
- [10] J.Richalet & al, "Model predictive heuristique control: Application to industrial process" *Atomatica*, Vol.14, pp 413-428, 1980.
- [11] C.R.Cutler & B.L.Ramaker, "Dynamic matrix control: computer control algorithm" *AICHE, national Mtg*, 1979.
- [12] D.W.Clarke & L.Zhang, "Long-rang predictive control using weighting-sequence models" *IEE Proc.*, Vol.134, Pt.D, N°3, 1987.
- [13] E.Carlos & al, "Model predictive control: Theory and practice-a survey" *Automatica*, Vol.25, N°3, pp 335-348, 1989.
- [14] D.W.Clarke, "Application of généralized predictive control to industrial process" *IEEE C.S.M.*, pp 49-55, 1988.
- [15] M.M'Saad & al, "A suitable generalized adaptive controller case study: Control of a flexible arm" *Automatica*, Vol.29, N°3, pp 589-608, 1993.
- [16] D.W.Clarke & C.Mohtadi, "properties of generalized predictive control" *Automatica*, Vol.25, pp 859-875, 1989.
- [17] B.E.Ydstie, "Extended horizon adaptive control" *IFAC 9th*, World congress, 1989.
- [18] DE Keyser & Van Cauwenbaghe, "Extended self adaptive control" *IFAC Symp.*, 1984.
- [19] K.Najim & G.Muratet, "Pratique de la régulation numériques processus industriels" Masson, 1983.
- [20] K.J.Aström & B.Wittenmark, "Computer controlled systems: Theory and design" Prentice Hall, 1990.
- [21] G.C.Goodwin & K.S.Sin, "Adaptive filtering prediction and control" Prentice Hall, 1984.
- [22] D.W.Clarke & L.Zhang, "Does long range predictive control work?" *IEE Conf. Publ.* 252, pp 13-18, 1985.

- [23] P.E.Wellstead & M.B.Zarrop, "Self tuning systems: Control and processing" John Wily, 1991.
- [24] K.Najim, "Commande adaptative des processus industriels" Masson, 1982.
- [25] K.Najim & G.Muratet, "Optimisation et commande en génie des procédés" Masson, 1987.
- [26] K.Djebara, "Commande adaptative monovariante d'un four de récuil rapide" Thèse de magister, HCR, 1992.
- [27] D.Dumur, "Commande prédictive et machine outil" thèse de Doctorat, ESE-SUPELEC, Orsay 1992.
- [28] P.Boucher & al, "Parameters automatic design of prediction cascaded controllers" IFAC, ENSIEG, 1992.
- [29] F.Benmiloud, "Simulation de la commande prédictive généralisée d'un actionneur asynchrone" Thèse de Magistère, ENP Alger, 1994.
- [30] K.J.Aström & B.Wittenmark, "Adaptive control" Addison-Wesley, 1989.
- [31] I.D.Landau & L.Dugard, "Commande adaptative: Aspect pratique et théoriques" Masson, 1986.
- [32] O.L.R.Jacobs, "Introductionh to adaptive control" H.Nicholson & B.H.Swanick, "Self tuning and adaptive control: theory and applications" IEE, C.E.S., 15, 1981.
- [33] V.V.Chalam, "Adaptive control systems: Techniques and applications" Marcel-Dekker
- [34] C.Foulard & Al, "Commande et régulation par calculateur numérique: de la theorie aux applications" Eyrolles, 1982.
- [35] I.D.Landau, "Identification et commande des systèmes: a l'aide des progiciels P.I.M. et PC-Reg" Hermes, 1988.
- [36] K.Najim & G.Muratet, "Pratique de la régulation numérique des processus industriels" Masson, 1983
- [37] L.Dugard & I.D.Landau, "Commande adaptative des systèmes: Méthodologie, Application" Ecole d'été d'automatique de Grenoble, 1990.
- [38] K.J.Aström & Al, "Theory and application of self tuning controllers" Automatica, Vol;20, N°5, pp 645-651, 1977.
- [39] R.Ortega & G.G.Sanchez, "Globaly convergence multistep receding horizon adaptive controller" Int.J.Contrl, Vol.49, N°5, pp 1655-1664, 1989.
- [40] C.Mohtadi & D.W.Clarke, "GPC, LQ or Pole-placement: A unified approach" Proc.of 25th conf. on Decision and control, 1986.
- [41] K.Warwick,, "Implementation of self tuning controllers" Peter Peregrinus, IEE control Engineering, seris 35, 1988.
- [42] P.Lopez & J.N.Foulc, "Introduction à la robotique" 2 tomes, Editest, 1984.
- [43] K.S.Fu & Al, "Robotics: Control, Sensing, Vision and intelligence" Mc Graw Hill, 1987.
- [44] M.Benallegue, "Contribution à la commande dynamique adaptative des robots manipulateurs rapides" Thèse de Doctorat de l'université de Paris 6, UPMC, 1991.
- [45] D.Stoten, "Modelling and control of elastic joint robots" ASME, Journal of dynamics system Measurement and control, Vol.109, pp 310-319, 1987.
- [46] J.Denavit & Al, "A kinematic notation for lower pair mechanisms based on matrices" Journal of applied mechanics, pp 215-221, 1955.
- [47] J.J.Craig, "Introduction to robotics: Mechanics and control" Addison-wesly, 1986.

- [48] R.P.Paul, "Robot manipulators: Mathematics, Programming and control" MIT press 1981.
- [49] H.Asada & J.E.Slotine, "Robot analysis and control" MIT, John Wiley, 1986.
- [50] K.J.Aström & B.Wittenmark, "On self tuning regulators" *Automatica*, Vol.9, pp 195-199, 1973.
- [51] P.E.Wellstead & Al, "Pole assignement self tuning regulator" *Proc. IEE*, Vol.126, N°8, pp 781-789, 1989.
- [52] K.J.Aström & B.Wittenmark, "Self tuning contollers: based on pole-zero placement" *IEE Proc. Pt.D*, Vol.127, N°3, pp 120-130, 1980.
- [53] Ulf Borison, "Self tuning regulators for class of multivariable systems" *Automatica*, Vol.15, pp 209-215, 1979.
- [54] H.N.Koivo, "A multivariable self tuning contoller" *Automatica*, Vol.16, pp 351-366, 1980.
- [55] J.M.Dion & Al, " MIMO adaptive constrained predictive control casa study: an environment test chamber" *Automatica*, Vol;27, N°4, pp 611-626, 1991.
- [56] J.M.Dion & L.Dugard, "Commande adaptative multivariable: quelques résultats théorique et pratique" *APII*, Vol.20, pp 337-356, 1986.
- [57] D.J.Cloud & B.Kouvaritakis, "Characteristic decomposition and the multivariable généralisation of predictive self tuning control" *IEE Pro.*, Vol.135, Pt.D, N°3, 1988.
- [58] H.Demircioğlu & P.J.Gawthrop, "Continus time generalized predictive control CGPC" *Automatica*, Vol.27, pp 55-74, 1991
- [59] H.Demircioğlu & P.J.Gawthrop, "Multivariable Continus time generalized predictive control " *Automatica*, Vol.28, N°8, pp 697-713, 1992
- [60] M.M'Saad & al, "A suitable generalized adaptive controller case study: Control of a flexible arm" *Automatica*, Vol.29, N°3, pp 589-608, 1993.
- [61] K.Najim & Al, "Commande predictive adaptative: theory et application" Rapport interne, CNRS, Toulous.
- [62] M.M'Saad & G.Sanchez, "Partial state reference model adaptive contol of multivariable systems" *Automatica*, Vol.28, N°6, pp 1189-1197, 1992.
- [63] S.L.Shah & Al, " Multivariable adaptive control without prior knowledge of the delay matrix" *System and control letters*, Vol.9, pp 295-306, 1987.
- [64] B.Bouzouia, "Adaptive control for robot manipulator: the self tuning approach" Rappot interne, LAAS, Toulouse, 1991.
- [65] Mei-Hua Liu & Wei lin, "Pole assignemnet self tuning controller for robotic" *Int.J.Control*, Vol.46, N°4, pp 1307-1317, 1987.
- [66] L.Guenfaf, N.Bali & M.S.Boucherit, "Identification réursive des systèmes multivariables: Etude de différentes structures" 2nd international meeting on components and Electronic systems, Sidi Bel Abbes, IMCES 2, 1994.
- [67] Mei-Hua Liu & Wei lin, "Multivariable self tuning control with decoupling for robotic manipulators" *IEE Proc.*, Vol.135, Pt.D, N°1, 1988.
- [68] R.Scatolini & N.Schianovi, "A not of multivariable généralized predictive control" *INT.J.Control*, Vol.56, N°1, pp 253-257, 1992.
- [69] M.S.Boucherit, "Sur l'application de l'automatique moderne dans la commande des machines electriques" Thèse de Doctorat d'état, ENP, (A paraître).