

**République algérienne démocratique et populaire**  
*Ministère de l'enseignement supérieur et de la recherche scientifique*

UNIVERSITE MENTOURI –CONSTANTINE-  
FACULTE DES SCIENCES DE L'INGENIEUR  
DEPARTEMENT D'ELECTRONIQUE

**THESE**

Présenté pour l'obtention du diplôme de  
**DOCTORAT EN SCIENCES**

Electronique

Option

CONTROLE DES SYSTEMES

Par

Moussa Sedraoui

N° d'ordre :

Série :

**Apport des principales méthodes d'optimisation  
dans la commande prédictive généralisée  
multivariable sous contraintes**

Soutenu publiquement le : / / 2008

Devant le jury :

|               |            |       |                           |
|---------------|------------|-------|---------------------------|
| M.L. Riabi :  | Président  | Prof. | Université de Constantine |
| S. Filali :   | Rapporteur | Prof. | Université de Constantine |
| S. Yahmedi :  | Examineur  | Prof. | Université d'Annaba       |
| I. Boutemina: | Examineur  | M.C.  | Université de Constantine |
| M.T. Khadir : | Examineur  | M.C.  | Université d'Annaba       |

Année 2008

# Avant-propos

Après de longues années d'étude on arrive aujourd'hui à la fin de celles-ci tout en souhaitant réussite, béni soit dieu le compassion et le miséricordieux qui nous a donné son aide et sa force afin qu'on puisse parvenir au terme de ce modeste travail.

Je voudrais saluer et remercier ici les personnes qui vont contribuer, toutes dans des manières très différentes, au déroulement original de mes six années de thèse.

Mes premiers sincères remerciements vont à mon directeur de recherche Monsieur S. Filali. Sa constante disponibilité, sa direction scientifique et ses qualités humaines m'ont été une aide précieuse pour mener à bien ce projet de doctorat.

Je remercie également messieurs, M.L. Riabi, S. Yahmedi, I. Boutemina et M.T. Khadir pour l'intérêt qu'ils ont accordé à mon travail en acceptant de l'examiner.

Pendant ces six années, j'ai été entouré par des personnes qui ont manifesté un intérêt tout particulier pour mes travaux, dont les critiques se sont révélées très précieuses. La liste est longue mais je tiens à remercier plus spécialement messieurs S. Gherbi, M.T. Maanani, M.C. Amara Corba, A. Louadjani, S.Derbel et A.M. Boumali.

Il sera difficile de trouver les mots pour remercier mes parents, ma femme, mes sœurs, mon frère et particulièrement Samer Mohamed El Habib qui ont pu me supporter, ont su m'encourager et m'aider tout au long de ce travail. Qu'ils trouvent ici toute ma gratitude et mon amour.

Enfin, nous remercions tous nos amis pour leur soutien moral.

# Sommaire

|   |    |
|---|----|
| Introduction générale .....   | 01 |
| <b>Chapitre1 : GPC multivariable algorithmique</b>  |    |
| 1. Introduction .....   | 05 |
| 1.2. GPC multivariable algorithmique multivariable.....   | 05 |
| 1.2.1. Introduction aux algorithmes prédictifs .....  | 05 |
| 1.2.1.1. Principe.....  | 06 |
| 1.2.2. Modèle prédictif .....   | 06 |
| 1.2.1.1. Factorisation d'une matrice de transfert .....   | 07 |
| 1.2.3. Critère à minimiser.....   | 08 |
| 1.2.4. Prédicteur optimal.....  | 10 |
| 1.2.5. Calcul de la solution optimale .....   | 12 |
| 1.2.6. Choix des paramètres de synthèse.....  | 16 |
| 1.2.6.1. Choix de $N_2$ : horizon de prédiction sur la sortie.....  | 16 |
| 1.2.6.2. Choix de $N_u$ : horizon de prédiction sur la commande.....  | 16 |
| 1.2.6.3. Choix de la matrice de pondération $\Lambda$ sur la commande.....  | 17 |
| 1.2.7. Exposé de la méthode pour des horizons $N_{2_i}$ de prédictions de la sortie différents sur chacune des voies..... | 17 |
| 1.3. Tests et simulations.....  | 19 |
| 1.3.1. Système1.....  | 19 |
| 1.3.2. Système2.....  | 22 |
| 1.4. Conclusion.....  | 25 |
| <b>Chapitre2 : GPC multivariable polynomial</b>   |    |
| 2. Introduction.....  | 26 |
| 2.3. Obtention du correcteur polynomial équivalent .....  | 27 |
| 2.4. GPC multivariable utilisant le filtre $C(q^{-1})$ .....  | 29 |
| 2.5. Analyse de robustesse.....   | 31 |
| 2.5.1. Introduction.....  | 31 |
| 2.5.2. Fonctions de sensibilité et sensibilité complémentaire.....  | 31 |
| 2.5.3. Théorème du faible gain.....   | 32 |

|  |    |
|--|----|
| 2.5.4. Robustesse « stabilité / performances » d'un système soumis à des incertitudes paramétriques..... | 34 |
| 2.5.5. Amélioration de la robustesse.....  | 34 |
| 2.5.6. Rôle de la matrice $C(q^{-1})$ .....  | 36 |
| 2.6. Tests et simulations.....   | 37 |
| 2.6.1. Application1 : « GPC polynomial ».....  | 37 |
| 2.6.2. Application2 : « Analyse de robustesse ».....   | 39 |
| 2.6.3. Application3 : « amélioration de robustesse ».....  | 43 |
| 2.7. Conclusion.....   | 47 |

### **Chapitre3 : GPC multivariable sous contraintes**

|   |    |
|---|----|
| 3. introduction.....                                    | 48 |
| 3.1. Les contraintes.....                               | 49 |
| 3.2. Formulation des contraintes.....                   | 50 |
| 3.2.1. Contraintes sur les incréments de commandes..... | 50 |
| 3.2.2. Contraintes sur les amplitudes de commandes..... | 51 |
| 3.2.3. Contraintes sur les signaux de sorties.....      | 51 |
| 3.2.4. Contraintes terminales de type inégalité .....   | 52 |
| 3.2.5. Autres types de contraintes .....                | 53 |
| 3.3. Principe de la fonction <i>quadprog</i> .....      | 54 |
| 3.3.1. Ensemble actif .....                             | 54 |
| 3.3.2. Point intérieur .....                            | 56 |
| 3.4. Région de faisabilité .....                        | 57 |
| 3.5. Tests et simulations.....                          | 62 |
| 3.6. Conclusion .....                                   | 67 |

### **Chapitre4 : GPC multivariable utilisant les méthodes d'optimisation**

|  |    |
|--|----|
| 4. Introduction.....   | 68 |
| 4.1. Méthodes d'optimisation utilisant les dérivées de la fonction objective.....      | 69 |
| 4.1.1. Méthode du gradient conjugué.....   | 70 |
| 4.2. Méthode d'optimisation n'utilisant pas les dérivées de la fonction objective..... | 71 |
| 4.2.1. Méthode de Dichotomie.....  | 71 |
| 4.2.1.1. Cas unidimensionnel.....  | 71 |
| 4.2.1.2. Cas pluridimensionnel.....  | 73 |
| 4.2.2. Amélioration de la méthode de dichotomie.....                                   | 75 |

|  |            |
|--|------------|
| 4.2.2.1. Méthode Tric_quad.....  | 75         |
| 4.2.2.2. Méthode de Dichotomie utilisant le paramètre $\Delta L$<br>(Dich_incr)..... | 77         |
| 4.3. GPC utilisant les méthodes d'optimisation proposées.....                        | 79         |
| 4.4. Tests et simulations.....   | 81         |
| 4.4.1. Application1.....   | 81         |
| 4.4.2. Application sur une colonne de distillation.....                              | 88         |
| 4.4.2.1. Description du procédé.....   | 88         |
| 4.4.2.2. Application de GPC utilisant les méthodes d'optimisation<br>proposées ..... | 90         |
| a). Cas nominal $K_1 = K_2 = 1$ .....  | 91         |
| b). Cas de $K_1 = K_2 = 1.2$ .....   | 93         |
| c). Cas de $K_1 = K_2 = 0.8$ .....   | 96         |
| d). Cas de $K_1 = 1.2, K_2 = 0.8$ .....  | 98         |
| 4.5. Interprétation de résultats et conclusion.....                                  | 100        |
| <b>Conclusion générale.....</b>  | <b>102</b> |
| <b>Perspectives.....</b>   | <b>104</b> |
| <b>Annexe A.....</b>   | <b>105</b> |
| <b>Références bibliographiques.....</b>  | <b>110</b> |

## Liste des figures

### Chapitre1 : GPC multivariable algorithmique

|   |    |
|---|----|
| Fig.1.1 : Principe de fonctionnement du GPC MIMO.....                   | 06 |
| Fig.1.2 : Fenêtre de prédiction de la sortie et la commande.....        | 09 |
| Fig.1.3 : Schéma représente le fonctionnement du réacteur chimique..... | 19 |
| Fig.1.4.1 : Sorties et Commandes du système corrigé.....                | 21 |
| Fig.1.4.2 : Incréments de commande.....                                 | 22 |
| Fig.1.5.1: Les sorties $y_i (i=1, 2, 3)$ .....                          | 23 |
| Fig.1.5.2 : Les commandes $u_i (i=1, 2, 3)$ .....                       | 24 |
| Fig.1.5.3 : Les incréments de commande $\Delta u_i (i=1, 2, 3)$ .....   | 24 |

### Chapitre2 : GPC multivariable polynomial

|   |    |
|---|----|
| Fig.2.1 : GPC sous forme algorithmique.....           | 26 |
| Fig.2.2 : GPC sous forme polynomiale équivalente..... | 26 |
| Fig.2.3 : Système bouclé classique.....               | 31 |

|   |    |
|---|----|
| Fig.2.4 : Schéma général d'étude de la robustesse.....  | 32 |
| Fig.2.5a: $G_p = G + \Delta_a$ .....  | 33 |
| Fig.2.5b: $G_p = (I_m + \Delta_s)G$ .....   | 33 |
| Fig.2.5c: $G_p = G(I_m + \Delta_e)$ .....   | 33 |
| Fig.2.5d: $G_p = (I_m + \Delta_{is})^{-1}G$ .....   | 33 |
| Fig.2.5e: $G_p = G(I_m + \Delta_{ie})^{-1}$ .....   | 33 |
| Fig.2.6 : Structure polynomiale au GPC pour $C(q^{-1})=I_m$ .....   | 34 |
| Fig.2.7 : Les valeurs singulières de la sensibilité directe et la sensibilité complémentaire.....               | 38 |
| Fig.2.8.1 : Sensibilité directe et condition de robustesse sur les performances (comparaison)...                | 41 |
| Fig.2.8.2 : Robustesse de stabilité (comparaison).....  | 41 |
| Fig.2.8.1 : Sorties/Commande obtenues par RST_GPC pour $w=(1\ 0)^T$ .....                                       | 42 |
| Fig.2.8.2 : Sorties/Commande obtenues par RST_GPC pour $w=(0\ 1)^T$ .....                                       | 42 |
| Fig.2.9.1 : Les entrées de perturbations injectées au système à commander.....                                  | 45 |
| Fig.2.9.2 : Les entrées de bruits agissant sur les entrées du système.....                                      | 45 |
| Fig.2.9.3 : Les sorties obtenues via le contrôleur RST avec ou sans filtre.....                                 | 46 |
| Fig.2.9.4 : Les commandes fournies par le contrôleur RST.....   | 46 |
| <b>Chapitre3 : GPC multivariable sous contraintes</b>   |    |
| Fig.3.1. : Classification des contraintes .....   | 49 |
| Fig.3.2.a. Sorties obtenues par le GPC multivariable utilisant la fonction <i>quadprog</i> .....                | 63 |
| Fig.3.2.b : Commandes fournies par le GPC multivariable utilisant la fonction <i>quadprog</i> .....             | 64 |
| Fig.3.2.c. Incréments de commandes fournis par le GPC multivariable utilisant la fonction <i>quadprog</i> ..... | 64 |
| Fig.3.3.a. Sorties obtenues par le GPC multivariable utilisant la fonction <i>quadprog</i> .....                | 65 |
| Fig.3.3.b. Commandes fournies par le GPC multivariable utilisant la fonction <i>quadprog</i> .....              | 66 |
| Fig.3.3.c. Incréments de commandes fournis par le GPC multivariable utilisant la fonction <i>quadprog</i> ..... | 66 |
| <b>Chapitre4 : GPC multivariable utilisant les méthodes d'optimisation</b>                                      |    |
| Fig. 4.1. Argument de l'optimum sans contraintes et de l'optimum avec contraintes .....                         | 70 |
| Fig.4.2. Comparaison entre les fonctions : $R_{dich\_incr}(N, \rho)$ , $R_{dichotomie}(N)$ .....                | 79 |
| Fig.4.3.a. Sorties obtenues par le GPC utilisant la méthode des gradients conjugués .....                       | 81 |
| Fig.4.3.b. Commandes fournies par le GPC utilisant la méthode des gradients conjugués .....                     | 82 |
| Fig.4.3.c. Incréments de commandes fournis par le GPC utilisant la méthode des gradients conjugués .....        | 82 |
| Fig.4.4.a. Sorties obtenues par le GPC utilisant la méthode de Dichotomie .....                                 | 83 |

|   |     |
|---|-----|
| Fig.4.4.b. Commandes fournies par le GPC utilisant la méthode de Dichotomie .....                                   | 83  |
| Fig.4.4.c. Incréments de commandes fournis par le GPC utilisant la méthode de Dichotomie ..                         | 84  |
| Fig.4.5.a. Sorties obtenues par le GPC utilisant la méthode de Tric_quad .....                                      | 84  |
| Fig.4.5.b. Commandes fournies par le GPC utilisant la méthode de Tric_quad .....                                    | 85  |
| Fig.4.5.c. Incréments de commandes fournis par le GPC utilisant la méthode des Tric_quad ...                        | 85  |
| Fig.4.6.a. Sorties obtenues par le GPC utilisant la méthode de Dich_incr .....                                      | 86  |
| Fig.4.6.b. Commandes fournies par le GPC utilisant la méthode de Dich_incr .....                                    | 86  |
| Fig.4.6.c. Incréments de commandes fournis par le GPC utilisant la méthode de Dich_incr ...                         | 87  |
| Fig.4.7 : Schéma bloc du système.....   | 88  |
| Fig.4.8.1 : Sorties (poursuite/interaction) du système : cas nominal et ( $w_1 = 1, w_2 = 0$ ).....                 | 91  |
| Fig.4.8.2 : Commande de la première voie du système : cas nominal et ( $w_1 = 1, w_2 = 0$ ).....                    | 91  |
| Fig.4.8.3 : Commande de la seconde voie du système : cas nominal et ( $w_1 = 1, w_2 = 0$ ).....                     | 92  |
| Fig.4.8.4 : Incrément de commande de la voie1 : cas nominal et ( $w_1 = 1, w_2 = 0$ ).....                          | 92  |
| Fig.4.8.5 : Incrément de commande de la voie2 : cas nominal et ( $w_1 = 1, w_2 = 0$ ).....                          | 93  |
| Fig.4.9.1 : Sorties (poursuite/interaction) : cas de ( $K_1 = K_2 = 1.2$ ) et ( $w_1 = 1, w_2 = 0$ ).....           | 93  |
| Fig.4.9.2 : Commande de voie1 : cas de ( $K_1 = K_2 = 1.2$ ) et ( $w_1 = 1, w_2 = 0$ ).....                         | 94  |
| Fig.4.9.3 : Commande de la voie2 : cas de ( $K_1 = K_2 = 1.2$ ) et ( $w_1 = 1, w_2 = 0$ ).....                      | 94  |
| Fig.4.9.4 : Incrément de commande de la voie1 : cas de ( $K_1 = K_2 = 1.2$ ) et ( $w_1 = 1, w_2 = 0$ )...95         | 95  |
| Fig.4.9.5 : Incrément de commande de la voie2 : cas de ( $K_1 = K_2 = 1.2$ ) et ( $w_1 = 1, w_2 = 0$ )...95         | 95  |
| Fig.4.10.1 : Sorties (interaction/poursuite) : cas de ( $K_1 = K_2 = 0.8$ ) et ( $w_1 = 0, w_2 = 1$ ).....          | 96  |
| Fig.4.10.2 : Commande de la voie1 : cas de ( $K_1 = K_2 = 0.8$ ) et ( $w_1 = 0, w_2 = 1$ ).....                     | 96  |
| Fig.4.10.3 : Commande de la voie2 : cas de ( $K_1 = K_2 = 0.8$ ) et ( $w_1 = 0, w_2 = 1$ ).....                     | 97  |
| Fig.4.10.4 : Incrément de commande de la voie1 : cas de ( $K_1 = K_2 = 0.8$ ) et ( $w_1 = 0, w_2 = 1$ )..97         | 97  |
| Fig.4.10.5 : Incrément de commande de la voie2 : cas de ( $K_1 = K_2 = 0.8$ ) et ( $w_1 = 0, w_2 = 1$ )...98        | 98  |
| Fig.4.11.1 : Sorties (interaction/poursuite) : cas de ( $K_1 = 1.2, K_2 = 0.8$ ) et ( $w_1 = 0, w_2 = 1$ )....98    | 98  |
| Fig.4.11.2 : Commande de la voie1 : cas de ( $K_1 = 1.2, K_2 = 0.8$ ) et ( $w_1 = 0, w_2 = 1$ ).....99              | 99  |
| Fig.4.11.3 : Commande de la voie2 : cas de ( $K_1 = 1.2, K_2 = 0.8$ ) et ( $w_1 = 0, w_2 = 1$ ).....99              | 99  |
| Fig.4.11.4: Incrément de commande de la voie1 : cas de ( $K_1 = 1.2, K_2 = 0.8$ ) et<br>( $w_1 = 0, w_2 = 1$ )..... | 100 |
| Fig.4.11.5: Incrément de commande de la voie2 : cas de ( $K_1 = 1.2, K_2 = 0.8$ ) et<br>( $w_1 = 0, w_2 = 1$ )..... | 100 |

**Contexte :**

Le but principal de la commande de systèmes, est de pouvoir satisfaire les objectifs définis par le cahier des charges, de manière la plus performante possible.

Il s'agit, par exemple, de contrôler certains signaux (de sortie) par actions sur d'autres signaux (d'entrée) de sorte qu'il soit possible de pouvoir satisfaire les objectifs en présence d'incertitudes et changements des caractéristiques du processus (modèle) et de restrictions sur les variables opérationnelles. Il s'agit là d'un problème pratique car bien des processus réels présentent ces caractéristiques.

Les actionneurs ont souvent un champ limité d'opération (position totalement ouverte ou fermée). Entrent en compte des raisons de sécurité, d'environnement ou de propres limitations des capteurs, des limites sur les variables des processus, (e.g. des niveaux, des flux, des températures et des pressions). Tout ceci rend nécessaire de considérer des contraintes dans l'étape de conception et d'implantation du contrôleur pour obtenir une meilleure performance du système.

La synthèse de la commande sans prendre en compte les restrictions mentionnées peut écartier la réponse du système commandé de celle désirée.

Les techniques de commande prédictive généralisée (GPC) (Generalized predictive Control) constituent des outils puissants pour affronter le problème de commande avec restrictions. Une synthèse sur ces méthodes où sont exposées les caractéristiques les plus représentatives peut être trouvée dans [CB99].

**Commande Prédictive Généralisée sans contraintes:**

Le GPC est une technique de commande pour des systèmes à dynamique relativement lente ou, du moins, compatible avec le fait qu'à chaque temps d'échantillonnage, le signal de commande découle de la résolution d'un problème d'optimisation quadratique sous contraintes.

La commande prédictive constitue un domaine ample et varié et intègre des disciplines comme la commande optimale, la commande par retour d'état ou par retour de sortie. Les avantages les plus importants que le GPC présente par rapport à d'autres méthodes sont les suivants:

- ✓ Il peut être employé pour contrôler une grande variété de processus, des systèmes avec un comportement relativement simple à d'autres qui présentent un comportement dynamique peu habituel comme ceux avec de grands retards, oscillant fortement, de phase non minimale ou

instables. Également les systèmes multivariables et non linéaires rentrent dans le domaine d'application d'une telle approche.

- ✓ Il possède intrinsèquement la compensation de retard et la compensation par anticipation (feedforward) de perturbations mesurables.
- ✓ Le traitement des restrictions peut être inclus systématiquement pendant le développement de la conception et de l'implantation du contrôleur. Les concepts manipulés sont intuitifs et peuvent être assimilés par un personnel sans connaissance profonde en commande.

A côté de tous ces avantages, restent quelques inconvénients associés. L'un d'eux est la charge nécessairement élevée de calcul: la résolution des algorithmes numériques en ligne nécessite un volume et un temps de calcul plus grands que ceux impliqués, par exemple, par la mise en œuvre de contrôleurs classiques de type PID.

Un autre inconvénient est dû au fait qu'il est nécessaire d'avoir un modèle approprié du processus. L'approche de commande prédictive est basée sur la connaissance d'un modèle et donc, les performances obtenues dépendront des écarts existant entre le vrai processus et le modèle utilisé.

Le GPC n'est pas une stratégie spécifique de commande mais c'est une méthodologie développée autour de certaines idées communes.

Les principes qui apparaissent à un degré plus ou moins élevé dans les classes de la commande prédictive sont basiquement les suivants:

- Utilisation du modèle du système pour prévoir la sortie du système à de futurs moments du temps.
- Calcul des actions optimales de commande basé sur la minimisation d'une ou plusieurs fonctions de coût ce qui peut inclure des restrictions sur les variables du processus.
- La stratégie de l'horizon mobile, c'est-à-dire, à chaque itération et en se servant d'un modèle du processus, des futures consignes sur un certain horizon d'une fonction objectif, les futurs changements de la commande sont calculés en prenant en compte des restrictions qui agissent sur le processus. Finalement, seulement le premier signal de commande est appliqué au système, rejetant le reste et déplaçant l'horizon vers le futur, répétant les calculs dans la période suivante.

### **Commande prédictive généralisée sous contraintes :**

Dans les applications industrielles, il y a plusieurs types de contraintes dont il faut tenir compte, telles que :

- ✓ Des restrictions physiques.
- ✓ Des spécifications technologiques.
- ✓ Des raisons de sécurité, de qualité...etc.

Un contrôleur doit être conçu pour satisfaire ce genre de spécifications technologiques, le rôle des contraintes dans un système de commande est double, ils doivent permettre une meilleure représentation des systèmes physiques, et d'autre part, être utilisées comme des paramètres de conception pour améliorer les performances du système, ces contraintes rendent le calcul de la commande assez difficile, ce qui nécessite l'inclusion des méthodes d'optimisation pour résoudre ce genre des problèmes.

Parmi les fonctions disponibles dans la bibliothèque d'optimisation du Matlab et les plus utilisées dans les derniers travaux celle de *Qaudprog* (Quadratic Programming), l'utilisation de cette fonction de le GPC offre des meilleurs résultats.

Toutefois pour un certain choix des paramètres de synthèse ( $N_2; N_u; \Lambda$ ) de GPC, un nombre des contraintes important, ainsi pour un mauvais choix des contraintes à satisfaire cette fonction présente plusieurs anomalies.

Notre approche se fixe comme objectif de palier ces inconvénients, ceci par proposition d'autres méthodes d'optimisations pluridimensionnelles permettant de résoudre le problème de la fonction de cout de GPC sous contraintes.

L'idée adoptée dans ce travail est de transformer de toutes contraintes à satisfaire à des bornes supérieures et inférieures (contraintes de bornes) avant d'appliquer l'optimisation. Avec cette transformation le nombre des contraintes à satisfaire devient très réduit ce qui peut faciliter le problème d'optimisation.

### **Organisation de la thèse:**

Ce travail est divisé en quatre chapitres, les deux premières développent la commande GPC sans contraintes de point de vue général et conceptuel, ils ont pour but de préciser les bases théoriques nécessaires pour la suite, le troisième chapitre est consacré à la commande prédictive généralisée multivariable sous contraintes, il est pour but de présenter, le principe de base de la fonction QP du

Matlab, la sensibilité de méthode aux choix des contraintes à satisfaire et le traitement des ces dernières pour les transformer sous forme des contraintes de bornes, ces contraintes sont utilisées par la suite pour définir la région de faisabilité des méthodes proposées dans le chapitre suivant. Le dernier chapitre présente les principales méthodes d'optimisation pour palier les inconvénients de la fonction *Quadprog* du Matlab. Le travail est organisé comme suit :

- ✓ Introduction générale
- ✓ Chapitre 1 : GPC multivariable algorithmique sans contraintes.
- ✓ Chapitre2 : GPC multivariable polynomial sans contraintes.
- ✓ Chapitre 3 : GPC multivariable sous contraintes.
- ✓ Chapitre 4 : GPC multivariable utilisant des méthodes d'optimisation multidimensionnelles.
- ✓ Conclusion générale.
- ✓ Annexe A.
- ✓ Références bibliographiques.
- ✓ Résumé.

## 1. Introduction

L'importance de la commande prédictive, introduite en 1978 par J.Richalet [Rich78], a crû au cours des années 80, et surtout après les travaux de D.W Clarke en 1984. La justification de la commande prédictive résulte de l'idée de tenir compte pour la conception du correcteur de maximum d'information dont l'automaticien dispose et tout premier lieu de la consigne.

De plus, la simplicité de l'algorithme, et sa facilité de mise en œuvre du moins en mono variable, lui ont permis d'être appliqué dans l'industrie avec efficacité.

Ce chapitre va donc décrire la commande prédictive généralisée dans sa version algorithmique pour des systèmes multivariables, cette dernière sera utilisée pour le chapitre suivant pour élaborer sa version polynomiale afin de traiter de robustesse de cette méthode de commande.

### 1.2. GPC algorithmique multivariable

#### 1.2.1. Introduction aux algorithmes prédictifs :

Comme pour le cas des algorithmes prédictifs monovariables, la mise en œuvre de la commande prédictive généralisée multivariable suppose que l'utilisateur dispose de trois types d'informations :

- La consigne  $W$  est connue sur un horizon plus ou moins long et choisi par l'utilisateur. Pour le cas multivariable, les horizons peuvent être choisis différemment sur chacune des voies.
- Un critère quadratique à minimiser incluant :
  - Un terme d'erreur entre la sortie et la consigne.
  - Un terme pondéré faisant intervenir la commande ou son incrément.
  - Un modèle MIMO linéaire du système à commander. Un tel modèle peut-être issu d'une identification (modèle de représentation) ou d'équations physiques du système (modèle de connaissances).

L'exigence de la connaissance du modèle apparaît dans le calcul de la prédiction de la sortie.

On se limitera dans la suite aux systèmes carrés de ' $m$ ' entrées, et ' $m$ ' sorties.

### 1.2.1.1. Principe :

Le schéma de principe de l'algorithme du GPC MIMO est représenté par la fig.1.1 :

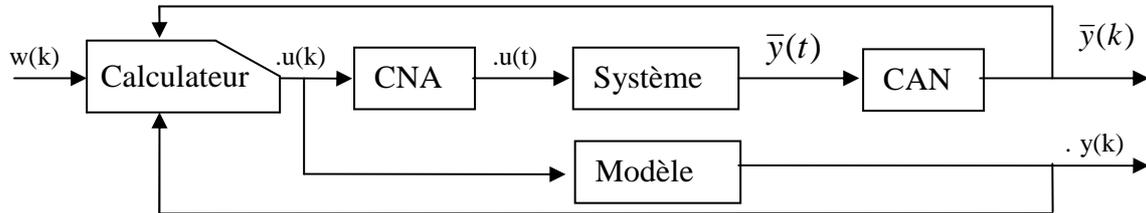


Fig.1.1 : Principe de fonctionnement du GPC MIMO

Comme on peut constater sur le schéma de la figure 1.1, une consigne spécifiant le comportement futur désiré de la sortie système est envoyée sur l'algorithme de minimisation du critère GPC.

Celui-ci renvoie une séquence de vecteur donnant la commande de l'instant  $t$  à  $t+Nu-1$  (cas MIMO), dont seul le premier vecteur donnant la commande à l'instant  $t$  est appliqué au système réel.

De plus, la commande ainsi calculée, sert à l'établissement de la prédiction  $\hat{Y}$ .

La valeur du vecteur de sortie système est également renvoyée vers l'algorithme GPC et permet d'assurer le recalage de la prédiction en cas d'erreur, ce qui se présente si le système subit des perturbations de sortie, par exemple, ou quand le modèle linéaire utilisé ne représente pas parfaitement le système réel (ex : erreurs paramétriques). C'est ce recalage qui fait du GPC un algorithme boucle fermée.

### 1.2.2. Modèle prédictif :

Le système est représenté sous la forme CARMA (Controlled Auto Regressive and Moving Average) suivante [CB93]:

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

$$A(q^{-1}) = I_{m \times m} + A_1 q^{-1} + \dots + A_{na} q^{-na}$$

$$\text{Avec : } B(q^{-1}) = B_0 + B_1 q^{-1} + \dots + B_{nb} q^{-nb}$$

$$C(q^{-1}) = I_{m \times m} + C_1 q^{-1} + \dots + C_{nc} q^{-nc}$$

On suppose que  $A(q^{-1})$  (respectivement  $\hat{C}(q^{-1})$ ) est inversible, soit  $\det A \neq 0$  (resp.  $\det \hat{C} \neq 0$ ) C'est-à-dire que  $\det A$  (resp.  $\det \hat{C}$ ) n'est pas identiquement nul. Les  $A_i, \hat{C}_i$  sont des matrices carrées réelles de dimension  $m$ .

Les  $B_i$  sont des matrices carrées réelles de dimensions  $m$ . De plus la séquence  $B_1, \dots, B_d$  ( $n_b \geq d \geq 1$ ) peut être nulle, dans ce cas  $d$  correspond au retard pur du système.

En outre, on a les données suivantes [CB93]:

$y(t)$  : Le vecteur de sortie à l'instant  $t$  (dimension  $m$ ).

$u(t-1)$  : La commande à l'instant  $t-1$  (dimension  $m$ ).

$\zeta(t)$  : représente une perturbation déterministe ou stochastique.

Une autre façon de représenter le système, et qui permet d'introduire une action intégrale dans la loi de commande, réside dans le modèle suivant [CB93]:

$$\Delta_m A(q^{-1})y(t) = B(q^{-1})\Delta_m u(t-1) + \hat{C}(q^{-1})\zeta(t) \quad (1.2)$$

Avec  $\Delta_m = \Delta I_m$ ,  $I_m$  est la matrice identité d'ordre  $m$ ,  $\Delta = 1 - q^{-1}$  est l'opérateur retard.

On obtient alors le modèle *CARIMA* (Controlled Auto Regressive Integrated Moving Average).

En introduisant  $\Delta_m$ , on demande au correcteur, pour chaque voie du système à commander, d'annuler complètement toutes erreurs statiques vis-à-vis d'une consigne constante. On peut aussi considérer  $\hat{C}(q^{-1})$ , apparaissant dans l'équation (1.1), comme un modèle particulier de perturbations  $\hat{C}(q^{-1}) = \Delta_m^{-1} C(q^{-1})$ . Dans ce chapitre, on prendra  $C(q^{-1}) = I_m$ . Le cas général  $C(q^{-1}) \neq I_m$  sera considéré au chapitre suivant.

### 1.2.2.1. Factorisation d'une matrice de transfert:

Il s'agit ici de transformer une matrice de transfert en un modèle *CARIMA* ceci pour étendre l'application de cette méthode de commande sur un système présenté par une matrice de transfert. Pour réaliser cette opération, on suppose que tous les coefficients de la matrice de transfert  $P(q^{-1})$  sont irréductibles sachant que :

$$P(q^{-1}) = \begin{bmatrix} P_{11}(q^{-1}) & P_{12}(q^{-1}) & \dots & P_{1m}(q^{-1}) \\ P_{21}(q^{-1}) & P_{22}(q^{-1}) & \dots & P_{2m}(q^{-1}) \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ P_{m1}(q^{-1}) & P_{m2}(q^{-1}) & \dots & P_{mm}(q^{-1}) \end{bmatrix} \quad (1.3)$$

Le lien de  $u(t)$  par  $y(t)$  sera donnée à l'aide de la matrice transfert  $P(q^{-1})$  sous la forme:

$$y(t) = P(q^{-1})u(t) \quad (1.4)$$

Utilisant les deux équations (1.4) et (1.3) pour construire le modèle *CARIMA* comme suit:

$$P(q^{-1}) = A^{-1}(q^{-1}).B(q^{-1}) \quad (1.5)$$

Le problème consiste à déterminer les deux matrices polynomiales  $A(q^{-1})$  et  $B(q^{-1})$ . La méthode la plus simple est de mettre la matrice  $A(q^{-1})$  sous forme diagonale dont chaque élément de cette dernière est égale au plus petit commun multiplicateur des éléments (*ppcm*) de la ligne correspondante de  $P(q^{-1})$ , on peut déduire:

$$B(q^{-1}) = A(q^{-1}).P(q^{-1}) \quad (1.6)$$

### 1.2.3. Critère à minimiser :

Le calcul de la commande dans le GPC multi variable résulte de la minimisation d'un critère proposé par Clarke et Al dans le cas mono variable en 1987 [CB93]

Le critère à minimiser est:

$$J = esp \sum_{i=1}^m \left[ \sum_{j=N_{1_i}}^{N_{2_i}} [w_i(t+j) - y_i(t+j)]^2 + \lambda_i \sum_{j=1}^{N_{u_i}} [\Delta u_i(t+j-1)]^2 \right] \quad (1.7)$$

Où  $N_{1_i}$  : Horizon minimal de prédiction de sortie.

$N_{2_i}$  : Horizon maximal de prédiction de sortie.

$N_{u_i}$  : Horizon de prédiction sur la commande.

$\lambda_i$  : facteur de pondération sur le  $i^{\text{ème}}$  incrément de commande.

$w_i(t+j)$  : Consigne appliquée à la  $i^{\text{ème}}$  voie à l'instant  $t+j$ .

$y_i(t+j)$  : Sortie prédite à l'instant  $t+j$  sur la voie  $i$ .

$\Delta u_i(t+j-1)$  : Incréments de commande sur la voie  $i$  à l'instant  $t+j-1$ .

Le critère quadratique (1.7) de la commande prédictive généralisée comprend deux termes de nature différente :

- l'un portant sur chacune des erreurs entre la consigne et la sortie prédite de la  $i^{\text{ème}}$  voie.
- L'autre est un terme proportionnel à « l'énergie » fournie par les commandes. Ce terme est pondéré par  $\lambda_i$ . Ainsi, plus  $\lambda_i$  est faible, et moins « l'énergie » de la commande est pénalisée dans la fonction de coût, et par conséquent, plus le correcteur GPC synthétisé est « énergique », et donc sa réponse est rapide. Ce terme permet d'éviter les signaux des commandes trop importants pouvant saturer le système. Dans ce travail, on ne considère que des réglages manuels de  $\lambda_i$ . (Noter que seule une optimisation sous contraintes du critère  $J$  permet d'assurer que les commandes ne dépassent pas des valeurs minimales et maximales imposées).

Comme il est possible de le constater à partir de (1.7), on peut théoriquement définir des horizons de prédiction de sortie  $N_{1_i}$  et  $N_{2_i}$  différents sur chacune des voies, de même pour l'horizon de commande  $N_{u_i}$ .

Dans la pratique, il faudrait avoir un comportement entrée-sortie véritablement différent sur chacune des voies pour appliquer le critère dans sa forme la plus générale. On prendra donc pour la suite des  $N_{1_i}$  égaux à  $N_1$  pour  $i=1, \dots, m$  (de même pour les  $N_{2_i}$  et les  $N_{u_i}$  égaux respectivement à  $N_2$  et  $N_u$ ).

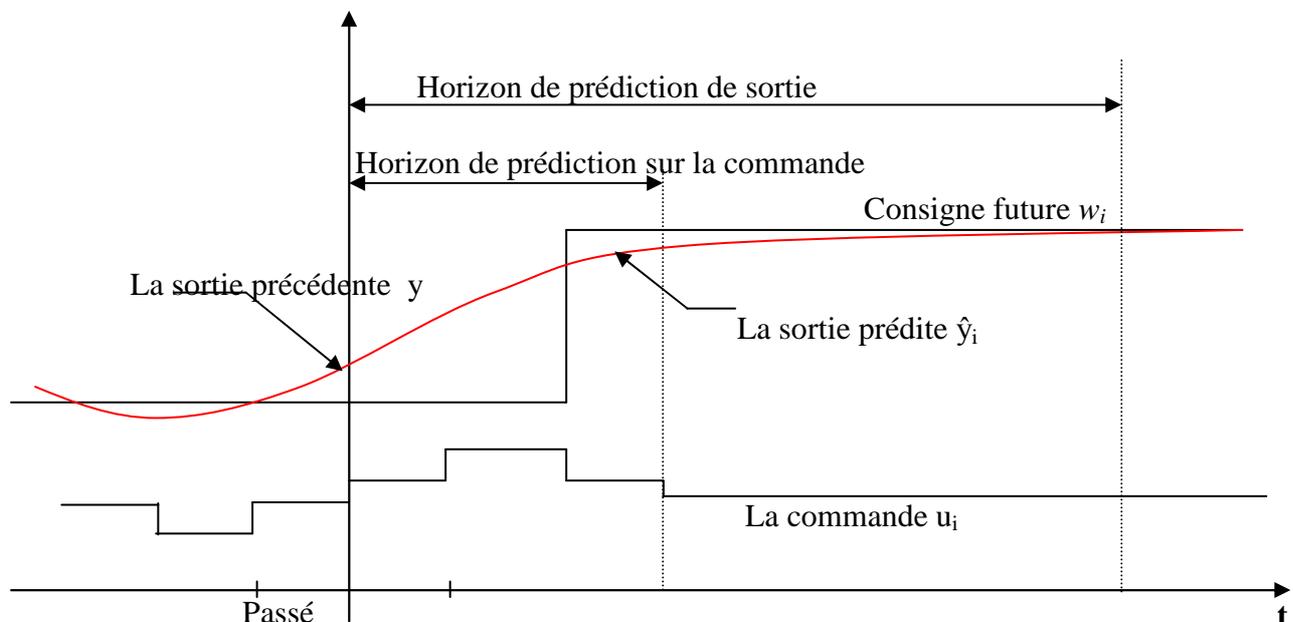


Fig.1.2 : Fenêtre de prédiction de la sortie et la commande

Une hypothèse supplémentaire est faite sur la commande :

$$\Delta u_i(t+j) = 0 \text{ pour } j \geq N_{u_i} \quad (1.8)$$

- On peut constater d'après la figure ci-dessus que  $\Delta u_i(t+j)$  est d'autant plus faible que la valeur de  $j$  est élevée donc, pour satisfaire la condition (1.8), il est préférable de prendre  $N_{u_i}$  élevée.

#### 1.2.4. Prédicteur optimal:

Pour simplifier les notations dans l'équation (1.7), on désigne par  $y(t+j)/_t$  les prédictions  $y(t+j)/_t$  de la sortie à l'instant  $t+j$  réalisées à l'instant  $t$ , de même pour les commandes future  $u(t+j)/_t$  que l'on notera par  $u(t+j)$ .

Le calcul de  $y(t+j)$ , la prédiction effectuée à l'instant  $t$  de la valeur  $y$  à  $j$  pas d'échantillonnage en avance, nécessite la résolution de deux équations Diophantiennes.

Soit  $(E_j, F_j)$  l'unique couple solution de l'équation Diophantienne [CB99]:

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

Avec :

$$E_j = E_0^j + E_1^j q^{-1} + \dots + E_{j-1}^j q^{-j+1}$$

$$F_j = F_0^j + F_1^j q^{-1} + \dots + F_{na}^j q^{-na}$$

Les matrices polynomiales  $E_j$  et  $F_j$  sont de dimension  $m \times m$ . La résolution de l'équation (1.9) de manière récursive et pour le cas général est décrite dans l'annexe A.

En opérant (1.2) par  $E_j q^j$  on obtient :

$$E_j A \Delta y(t+j) = E_j B \Delta u(t+j-1) + E_j \zeta(t+j) \quad (1.10)$$

La substitution de  $E_j A \Delta$  provenant de l'équation (1.9) donne alors :

$$(I_m - q^{-j} F_j) y(t+j) = E_j B \Delta u(t+j-1) + E_j \zeta(t+j) \quad (1.11)$$

On obtient après le développement:

$$y(t+j) = F_j y(t) + E_j B \Delta u(t+j-1) + E_j \zeta(t+j) \quad (1.12)$$

Dans cette dernière équation, toutes les références à des valeurs de la perturbation pour des instants d'échantillonnage passés et présents ont été supprimées.

Il ne subsiste plus que la combinaison linéaire  $E_j \zeta(t+j)$  de valeurs futures de la perturbation ou du bruit. Or ces valeurs sont par définition indépendantes de signaux mesurables à l'instant  $t$ . Il est donc clair que la prédiction optimale à l'instant  $t$  de  $y(t+j)$ , notée  $\hat{y}(t+j)$ , obtenue au sens du minimum de la variance de l'erreur et grâce à des grandeurs dont la valeur est connue à l'instant  $t$ , est donnée par :

$$\hat{y}(t+j) = F_j y(t) + E_j B \Delta u(t+j-1) \quad (1.13)$$

Dans l'équation (1.13), le terme  $E_j B \Delta u(t+j-1)$  est une combinaison linéaire de valeurs de  $\Delta u$  à des instants d'échantillonnage compris entre les instants  $t - n_b - d$  et  $t + j - 1 - d$ . Mais la fonction de coût (1.7) implique seulement les valeurs futures et la valeur présente de  $\Delta u$ . Ainsi va-t-on chercher sa séparer ces valeurs des valeurs passées dans l'équation (1.13). Cette séparation peut être obtenue grâce à la résolution d'une seconde équation Diophantienne, en  $G_j$  et  $H_j$  [CB99]:

$$E_j B = G_j + q^{-j} H_j \quad (1.14)$$

$$G_j = G_0^j + G_1^j q^{-1} + \dots + G_{j-1}^j q^{-j+1}$$

$$\text{Avec : } H_j = H_0^j + H_1^j q^{-1} + \dots + H_{n_h}^j q^{-n_h}$$

$$n_h = n_b + d - 1$$

Les matrices des polynômes  $G_j$  et  $H_j$  sont de dimension  $m \times m$ .

L'équation Diophantienne (1.14) peut se résoudre récursivement et dans le cas général à l'aide de la méthode rappelée en *annexe A*.

L'utilisation de (1.14) dans (1.13) conduit finalement à :

$$\hat{y}(t+j) = G_j \Delta u(t+j-1) + [H_j \Delta u(t-1) + F_j y(t)] \quad (1.15)$$

Le premier terme de la somme désigne la réponse « forcée » du système, qui est due aux commandes futures, et le deuxième terme, la réponse dite « libre ».

Dans le sens qu'elle est la conséquence, dans le futur, des commandes qui ont été appliquées jusqu'à l'instant  $t$ .

### 1.2.5. Calcul de la solution optimale :

On va supposer que les  $N_i$  sont égaux pour tous les  $i = 1, 2, \dots, m$  (de même pour les  $N_{2_i}$ , et  $N_{u_i}$ )

Notons ainsi que l'on envisagera des cas d'horizons distincts par la suite.

Posons:

$$Fc = \begin{bmatrix} Fc_{N_1} \\ Fc_{N_1+1} \\ \cdot \\ \cdot \\ \cdot \\ Fc_{N_2} \end{bmatrix} \text{ Vecteur colonne de dimension } m \times (N_2 - N_1 + 1) \text{ avec } Fc_k = \begin{bmatrix} fc_{1,k} \\ fc_{2,k} \\ \cdot \\ \cdot \\ fc_{m,k} \end{bmatrix}; k = N_1, \dots, N_2 \text{ et}$$

$$Fc = \begin{bmatrix} H_{N_1} \Delta u(t-1) + F_{N_1} y(t) \\ H_{N_1+1} \Delta u(t-1) + F_{N_1+1} y(t) \\ \cdot \\ \cdot \\ H_{N_2} \Delta u(t-1) + F_{N_2} y(t) \end{bmatrix} \quad (1.16)$$

$$\hat{Y} = \begin{bmatrix} \hat{y}(t + N_1) \\ \hat{y}(t + N_1 + 1) \\ \cdot \\ \cdot \\ \hat{y}(t + N_2) \end{bmatrix} \text{ Vecteur colonne de dimension } m \times (N_2 - N_1 + 1) \text{ des sorties futures estimées,}$$

$$\text{avec } \hat{y}(t+k) = \begin{bmatrix} \hat{y}_1(t+k) \\ \hat{y}_2(t+k) \\ \cdot \\ \cdot \\ \hat{y}_m(t+k) \end{bmatrix}; k = N_1, \dots, N_2$$

$$W = \begin{bmatrix} w(t + N_1) \\ w(t + N_1 + 1) \\ \cdot \\ \cdot \\ w(t + N_2) \end{bmatrix} \text{ Vecteur colonne de dimension } m \times (N_2 - N_1 + 1) \text{ de consigne, avec}$$

$$w(t + k) = \begin{bmatrix} w_1(t + k) \\ w_2(t + k) \\ \cdot \\ \cdot \\ w_m(t + k) \end{bmatrix}; k = N_1, \dots, N_2$$

$$\tilde{U} = \begin{bmatrix} \Delta u(t) \\ \Delta u(t + 1) \\ \cdot \\ \cdot \\ \Delta u(t + N_u - 1) \end{bmatrix} \text{ Vecteur colonne de dimension } m \times N_u \text{ des incréments de contrôles, avec}$$

$$\Delta u(t + k) = \begin{bmatrix} \Delta u_1(t + k) \\ \Delta u_2(t + k) \\ \cdot \\ \cdot \\ \Delta u_m(t + k) \end{bmatrix}; k = 0, \dots, N_u - 1$$

On peut alors écrire l'équation (1.15), pour  $N_1 \leq j \leq N_2$ , sous la forme:

$$\hat{Y} = G\tilde{U} + Fc \tag{1.17}$$

Avec  $G \in \mathfrak{R}^{m \times (N_2 - N_1 + 1) \times (m \cdot N_u)}$  définie comme suit :

$$\text{Avec : } G = \begin{bmatrix} G_{N_1-1} & \Lambda & G_0 & \Lambda & \Lambda & 0 \\ G_{N_1} & G_{N_1-1} & \Lambda & G_0 & \Lambda & 0 \\ M & M & O & O & O & M \\ G_{N_u-1} & G_{N_u-2} & G_{N_u-3} & \Lambda & \Lambda & G_0 \\ M & M & M & M & M & M \\ G_{N_2-1} & G_{N_2-2} & G_{N_2-3} & \Lambda & G_{N_2-N_u+1} & G_{N_2-N_u} \end{bmatrix}$$

Les termes  $G_i \in \mathfrak{R}^{m \times m}$  sont les matrices réelles représentent les coefficients de la matrice  $G$  avec :

$$G_i = \begin{bmatrix} g_{11} & \dots & \dots & g_{1m} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ g_{m1} & \dots & \dots & g_{mm} \end{bmatrix}_i \quad (1.18)$$

où  $(g_{ij})_k$  est le coefficient de la réponse indicielle à l'instant  $k$  pour le transfert entre la  $j^{\text{ème}}$  entrée et la  $i^{\text{ème}}$  sortie.

L'expression du critère (1.7) peut donc s'écrire sous forme matricielle. Compte tenu des notations précédentes, il vient [CB93] :

$$J = (\hat{Y} - W)^T (\hat{Y} - W) + \tilde{U}^T \Lambda \tilde{U} \quad (1.19)$$

$$\text{Avec : } \Lambda = \begin{bmatrix} \begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & \lambda_m \end{bmatrix}_1 & 0 \\ \vdots & \vdots \\ 0 & \dots & \dots & \begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & \lambda_m \end{bmatrix}_{N_u} \end{bmatrix} = \text{diag}(\text{diag}(\lambda_{i=1,2,\dots,m})_1, \dots, \text{diag}(\lambda_{i=1,2,\dots,m})_{N_u}) \quad (1.20)$$

En remplaçant  $\hat{y}$  par son expression (1.15), la valeur de  $J$  au sens des moindres carrés finalement obtenue pour  $\delta U = \delta U_{opt}$  avec :

$$\tilde{U}_{opt} = (G^T G + \Lambda)^{-1} G^T (W - Fc) = M \cdot (W - Fc) \quad (1.21)$$

D'après (1.21), on peut remarquer que la matrice  $(G^T G + \Lambda)^{-1}$  à inverser est de dimension  $m.N_u \times m.N_u$ . On aura donc tout intérêt à ne pas prendre une valeur de  $N_u$  trop élevée.

La matrice  $M \in \mathfrak{R}^{m.N_u \times m.(N_2 - N_1 + 1)}$  est la matrice du gain optimal du correcteur GPC, en se basant sur le principe de l'horizon glissant et seules les  $m$  premières lignes des vecteur de commande  $\tilde{U}_{opt}$  sont appliquées au système (ces  $m$  premières lignes correspondent au calcul de la commande à l'instant  $t$ , les  $m.(N_u - 1)$  autres correspondent aux instants  $t + 1, \dots, t + N_u - 1$  quand  $N_u \neq 1$ )

La commande réellement appliquée au système au temps  $t$  vaut :

$$u(t) = u(t-1) + \Delta u_{opt}(t)$$

$$\text{Avec : } \Delta u_{opt}(t) = M_1(W - Fc) \quad (1.22)$$

$M_1 \in \mathfrak{R}^{m \times m.(N_2 - N_1 + 1)}$  Est la matrice qui représente les  $m$  premières lignes de la matrice  $M$ .

La procédure est répétée à chaque période d'échantillonnage. Comme on peut le constater certains éléments de l'algorithme de GPC peuvent être calculés hors ligne, d'autre pas.

Le calcul de la commande par GPC peut donc se résumer de la façon suivante :

▪ *Calculs hors ligne :*

- Initialisation des matrices  $A(q^{-1})$  et  $B(q^{-1})$  du modèle.
- Choix des paramètres  $N_1, N_2, N_u, \Lambda$ .
- Calcul des coefficients de  $G_j$ .
- Calcul de  $M_1$ , i.e. des  $m$  premières lignes de  $(G^T G + \Lambda)^{-1} G^T$ , de dimension  $m \times m.(N_2 - N_1 + 1)$ .
- Calcul de la consigne.

▪ *Calculs en temps réel pour chaque période d'échantillonnage :*

- Calcul du vecteur  $Fc$ .
- Calcul de la commande minimisant le critère  $J$ .

### 1.2.6. Choix de paramètres de synthèse :

On peut constater, que la commande en GPC est extrêmement simple. Le seul point délicat concerne les degrés de liberté laissés à l'utilisateur, à savoir le choix des paramètres  $N_1, N_2, N_u, \Lambda$ , étant donné le nombre de réglages possibles.

Notons néanmoins que l'étendue des réglages est l'une des forces du GPC : celle-ci lui permet de s'adapter à toutes sorte de systèmes, depuis les systèmes à déphasage minimal jusqu'aux systèmes instables en boucle ouverte. Le but d'un « bon » réglage d'un système par le GPC est d'assurer des marges de robustesse en stabilité suffisantes pour un maintien des performances correct face à des perturbations, des variations paramétriques ou non du modèle.

On suppose les  $N_{2_i}$  égaux pour  $i=1,2,\dots,m$  (de même pour  $N_{1_i}$  et  $N_{u_i}$ ). Dans le cas contraire, le choix effectué pour l'un des paramètres d'indice 'i' doit tenir compte des caractéristiques propres à la  $i^{\text{ème}}$  voie.

De nombreuses études ont permis aux scientifiques d'énoncer le choix suivant [CB93],[Cama93]:

#### 1.2.6.1. Choix de $N_2$ : horizon de prédiction sur la sortie:

- Dans le cas multivariable, on prend  $N_1$  comme la valeur minimale des retards purs de chaque voie. Pour un système ne représentant de retard pur sur aucune voie, on prend  $N_1 = 1$ .
- L'horizon final (maximum) de prédiction  $N_2$  est choisi de façon que  $N_2 \times T_e$  est égal au maximum des temps de réponse en boucle ouverte pour chacune des voies. où  $T_e$  est le temps d'échantillonnage du contrôleur. Il est à noter que plus  $N_2$  est grand, plus le temps de calcul est long.

#### 1.2.6.2. Choix de $N_u$ : horizon de prédiction sur la commande:

Pour les processus simples, prendre l'horizon de commande  $N_u$  égal à 1 donne souvent de bons résultats, par contre, pour les processus complexes  $N_u$  doit être égal au moins au nombre des pôles instables ou mal amortis.

L'horizon de commande ne doit en aucun cas avoir une valeur supérieur à celle de l'horizon de prédiction maximal.

### 1.2.6.3. Choix de la matrice de pondération $\Lambda$ sur la commande:

Le problème lié à ce paramètre réside dans le fait qu'il détermine (avec  $N_2$ ) la dynamique du système en boucle fermée d'une manière peu précise.

En effet, si intuitivement, on peut dire que plus  $\lambda_i$  est élevé, et plus le temps de réponse du système est long sur la  $i^{\text{ème}}$  voie, en revanche, il n'existe pas de relation reliant directement ce paramètre au temps de réponse sur la  $i^{\text{ème}}$  voie.

Si, en monovariante, la valeur de  $\lambda$  est liée à celle du gain du système, il reste à étendre cette constatation au cas multivariable où l'horizon de prédiction de commande  $N_u$  est différent de « 1 » tout en conservant une bonne robustesse en performance.

On peut signaler également qu'augmenter  $\lambda_i$  revient à accroître la contrainte sur la commande de la  $i^{\text{ème}}$  voie, le cas  $\lambda_i = 0$  peu réaliste revient à ne mettre aucune contrainte.

Notons pour finir que la matrice  $\Lambda$  joue un rôle important dans le conditionnement numérique de la méthode puisqu'elle intervient dans la matrice  $G^T \cdot G + \Lambda$  que l'on doit inverser.

### 1.2.7. Exposé de la méthode pour des horizons $N_{2_i}$ de prédiction de la sortie différents sur chacune des voies [CB93]:

Pour simplifier les notations, on ne traitera que le cas  $m = 2$  et on supposera que  $N_{2_1} \neq N_{2_2}$ ,  $N_{1_1} = N_{1_2} = N_1$  et les  $N_{u_i}$  sont tous égaux à  $N_u$ , le cas des horizons  $N_{u_i}$  différents sur chacune des voies sera déduit à partir de ce qui va suivre.

- Le prédicteur optimal  $\hat{Y}(t+k)$  est utilisé entre  $N_1$  et  $N_{2_1}$ .
- Les notions utilisées au (1.2.5) deviennent :

$$\hat{Y} = [\hat{y}(t+N_1) \quad \dots \quad \hat{y}(t+N_{2_2}) \quad \dots \quad \hat{y}(t+N_{2_1})]^T \quad (1.23)$$

$$\text{Avec : } \hat{y}(t+k) = \begin{bmatrix} \hat{y}_1(t+k) \\ 0 \end{bmatrix} \text{ pour } k > N_{2_2}$$

$$W = [w(t+N_1) \quad \dots \quad w(t+N_{2_2}) \quad \dots \quad w(t+N_{2_1})]^T \quad (1.24)$$

$$\text{Avec : } w(t+k) = \begin{bmatrix} w_1(t+k) \\ 0 \end{bmatrix} \text{ pour } k > N_{2_2}$$

- $\tilde{U}$  est inchangée, puisque les  $N_u$  sont égaux et la matrice  $Fc$  devient :

$$Fc = \left[ Fc_{N_1} \quad \dots \quad Fc_{N_{2_2}} \quad \dots \quad Fc_{N_{2_1}} \right]^T \quad (1.25)$$

$$\text{Avec } Fc_k = \begin{bmatrix} fc_{1,k} \\ 0 \end{bmatrix} \text{ pour } k > N_{2_2}$$

- On peut alors réécrire la relation (1.20) Avec le changement suivant sur la matrice  $G$  :

$$G = \begin{bmatrix} G_{N_1-1} & \Lambda & G_0 & \Lambda & \Lambda & 0 \\ G_{N_1} & G_{N_1-1} & \Lambda & G_0 & \Lambda & 0 \\ M & M & O & O & O & M \\ G_{N_u-1} & G_{N_u-2} & G_{N_u-3} & \Lambda & \Lambda & G_0 \\ M & M & M & M & M & M \\ G_{N_{2_1}-1} & G_{N_{2_1}-2} & G_{N_{2_1}-3} & \Lambda & G_{N_{2_1}-N_u+1} & G_{N_{2_1}-N_u} \end{bmatrix} \quad (1.26)$$

- Notons que les  $G_j$  sont de la forme suivante :

$$G_j = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}_j \text{ pour } j = N_1, \dots, N_{2_2}. \quad (1.27)$$

$$G_j = \begin{bmatrix} g_{11} & g_{12} \\ 0 & 0 \end{bmatrix}_j \text{ pour } j = N_{2_2} + 1, \dots, N_{2_1}. \quad (1.28)$$

Avec les notations précédentes, le critère minimisé garde la forme écrite en (1.19) et la solution est celle développée en (1.21). La seule matrice invariante est  $\Lambda$ .

On peut généraliser ce qui vient d'être exposé au cas  $m \neq 2$ .

Il convient d'utiliser le prédicateur jusqu'au maximum des  $N_{2_i}$ , on met alors la prédiction sur la voie  $i$  à 0 au-delà de  $N_{2_i}$  (de même  $w_i(t+k) = 0$  pour  $k \neq N_{2_i}$ ).

### 1.3. Tests et simulations :

Dans cette partie on présente la simulation de la commande prédictive multi variable pour les deux cas suivants :

- pour des horizons  $N_{2_i}$  et  $N_{u_i}$  différents, la simulation est effectuée sur modèle mathématique que représente une dynamique d'un réacteur exothermique.
- Pour des horizons  $N_{2_i}$  et  $N_{u_i}$  identiques, la simulation est effectuée sur un système électromécanique représenté par un modèle prédictif de *CARIMA*.

#### 1.3.1. Système1:

Le processus multi variable suivant correspond à un système emprunté dans [CB99] à auquel il a été fait une légère modification pour pouvoir appliquer la méthode proposée. Le système consiste en un réacteur chimique avec chemise de refroidissement. La décomposition d'un produit A en un autre produit B se produit dans le réacteur (voir Fig.1.3). La réaction est exothermique et par conséquent la température doit être contrôlée par circulation d'eau à travers la chemise de refroidissement qui entoure les parois du réservoir. L'objectif est de régler la température dans le réservoir ( $T_R$ ) et la concentration du produit en sortie ( $C_A$ ).

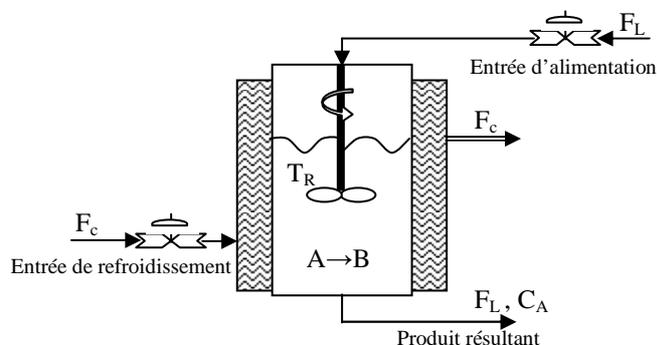


Fig.1.3. Schéma représente le fonctionnement du réacteur chimique

Le modèle est décrit par la matrice de transfert suivante [CB99] :

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{1+0.7 \cdot s} & \frac{5}{1+0.3 \cdot s} \\ \frac{1}{1+0.5 \cdot s} & \frac{2}{1+0.4 \cdot s} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

où les variables manipulées  $u_1$  et  $u_2$  sont les flux d'alimentation ( $F_L$ ) et le flux de refroidissement dans la chemise ( $F_c$ ) respectivement. Les variables contrôlées  $y_1$  et  $y_2$  sont la concentration du produit en sortie ( $C_A$ ) et la température dans le réacteur ( $T_R$ ) respectivement.

- Les constantes du temps de ce système sont exprimées en minutes.

- Pour la simulation, on a calculé la matrice de transfert discrète avec la période d'échantillonnage  $T_e = 0.03$  minutes on obtient :

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} \frac{0.0420 q^{-1}}{1 - 0.958 q^{-1}} & \frac{0.4758 q^{-1}}{1 - 0.9048 q^{-1}} \\ \frac{0.0582 q^{-1}}{1 - 0.9418 q^{-1}} & \frac{0.1445 q^{-1}}{1 - 0.9277 q^{-1}} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

La matrice  $A(q^{-1})$  peut être obtenue en construisant une matrice diagonale dont les éléments sont égaux au plus petit commun multiple des dénominateurs de chaque ligne, on obtient donc :

$$A(q^{-1}) = \begin{bmatrix} 1 - 1.8630q^{-1} + 0.8669q^{-2} & 0 \\ 0 & 1 - 1.8700q^{-1} + 0.8737q^{-2} \end{bmatrix}$$

En utilisant la relation (1.6) pour déterminer la matrice polynomiale  $B$  où :

$$B(q^{-1}) = \begin{bmatrix} 0.04195q^{-1} - 0.03796q^{-2} & 0.4758q^{-1} - 0.4559q^{-2} \\ 0.05824q^{-1} - 0.05403q^{-2} & 0.1445q^{-1} - 0.1361q^{-2} \end{bmatrix}$$

On applique dans l'intervalle du temps  $0 \leq t \leq 100$  les références suivantes :

- Pour la voie 1 :  $W_1 = \begin{cases} 0.5 & 0 \leq t \leq 50 \\ 0.4 & 50 \leq t \leq 100 \end{cases}$
- Pour la voie 2 :  $W_2 = 0.30 : 0 \leq t \leq 100$

Les paramètres de contrôle  $(5, 4, 3, 2, 0.1, 1)$  ont été choisis pour  $(N_{2_1}, N_{2_2}, N_{u_1}, N_{u_2}, \lambda_1, \lambda_2)$ .

Les allures des différentes sorties, commandes et incréments de commandes seront illustré par les figures 1.4.1 et Fig.1.4.2.

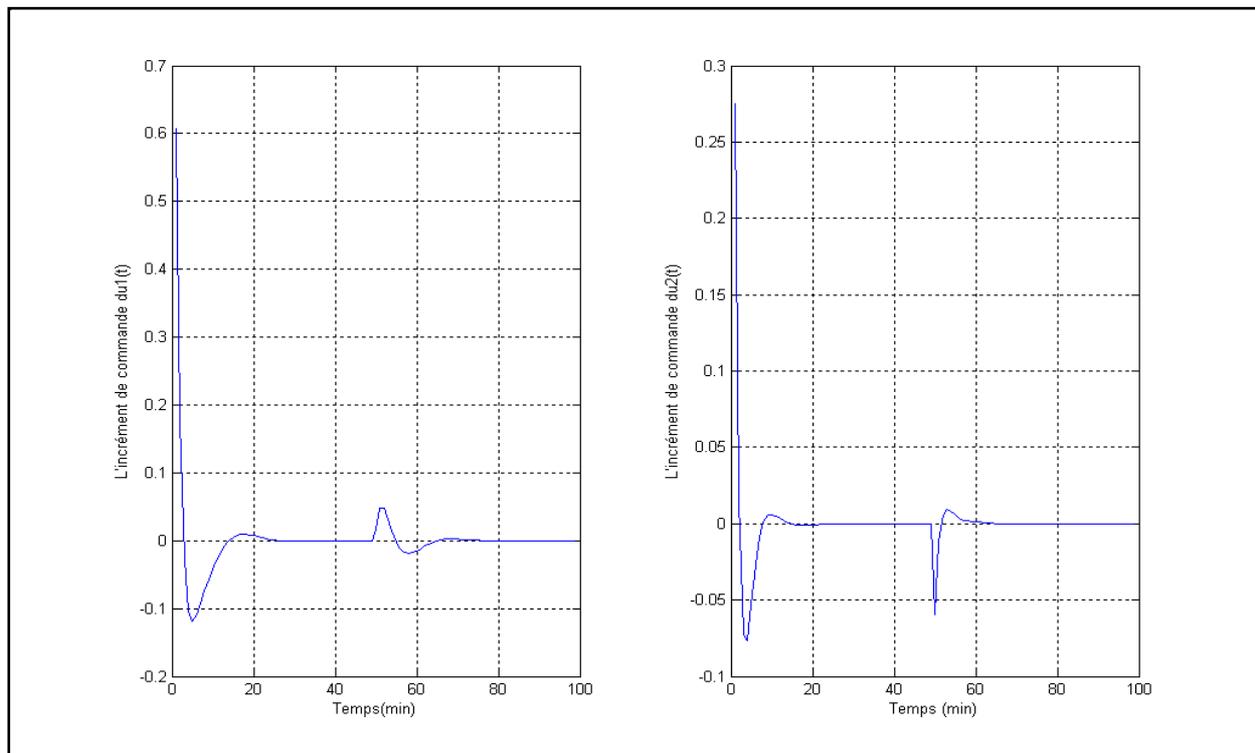
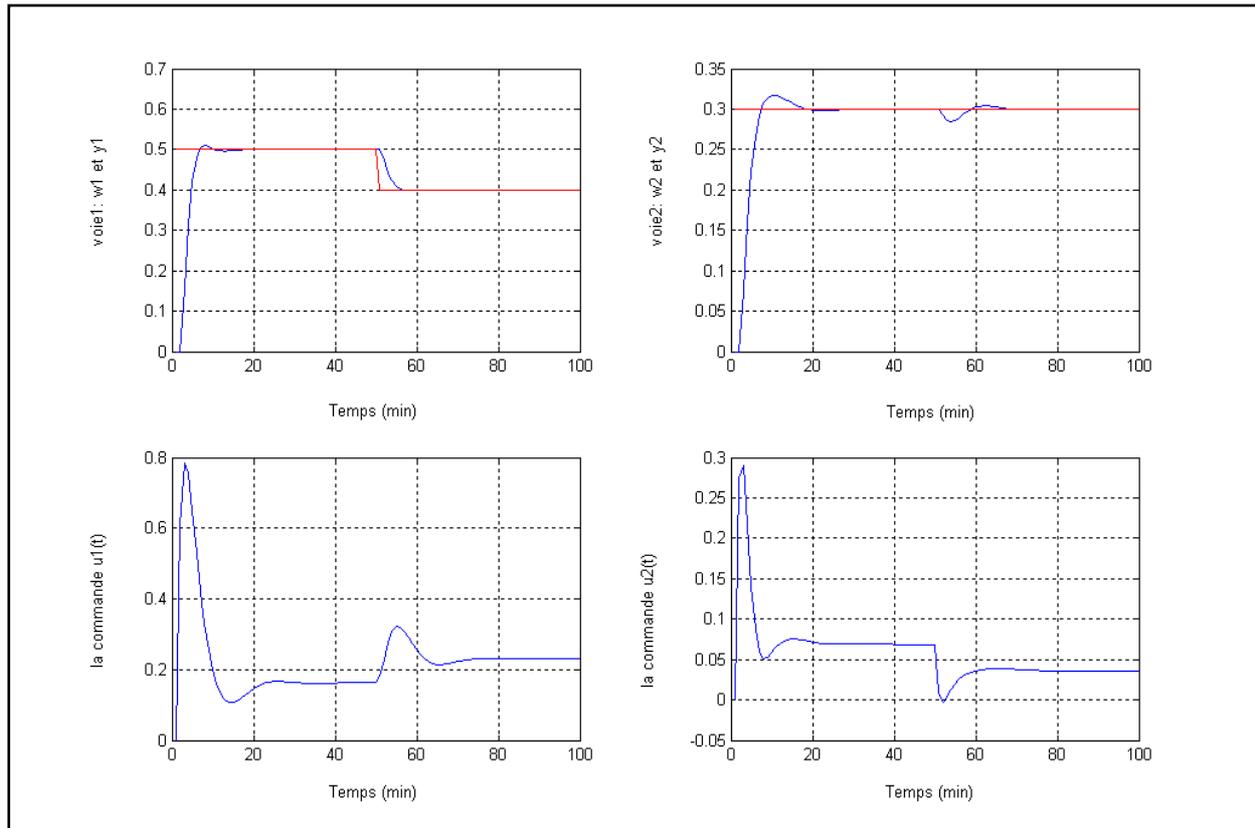


Fig.1.4.1 : Sorties et Commandes du système corrigé

Fig.1.4.2 : Incrément de commande

D'après les résultats obtenus, on peut remarquer notamment dans les régimes permanents :

- Des sorties tendent vers des valeurs constantes (valeurs de consignes)
- Des commandes stationnaires tendent vers les valeurs  $(0.2333 \quad 0.0333)^T$ , ces dernières représentent le produit de l'inverse du gain statique du système avec le vecteur de consignes  $w_{t \rightarrow 100} = (0.4 \quad 0.3)^T$ .

### 1.3.2. Système2 :

Le système considéré est celui décrit par le modèle de *CARIMA* suivant:

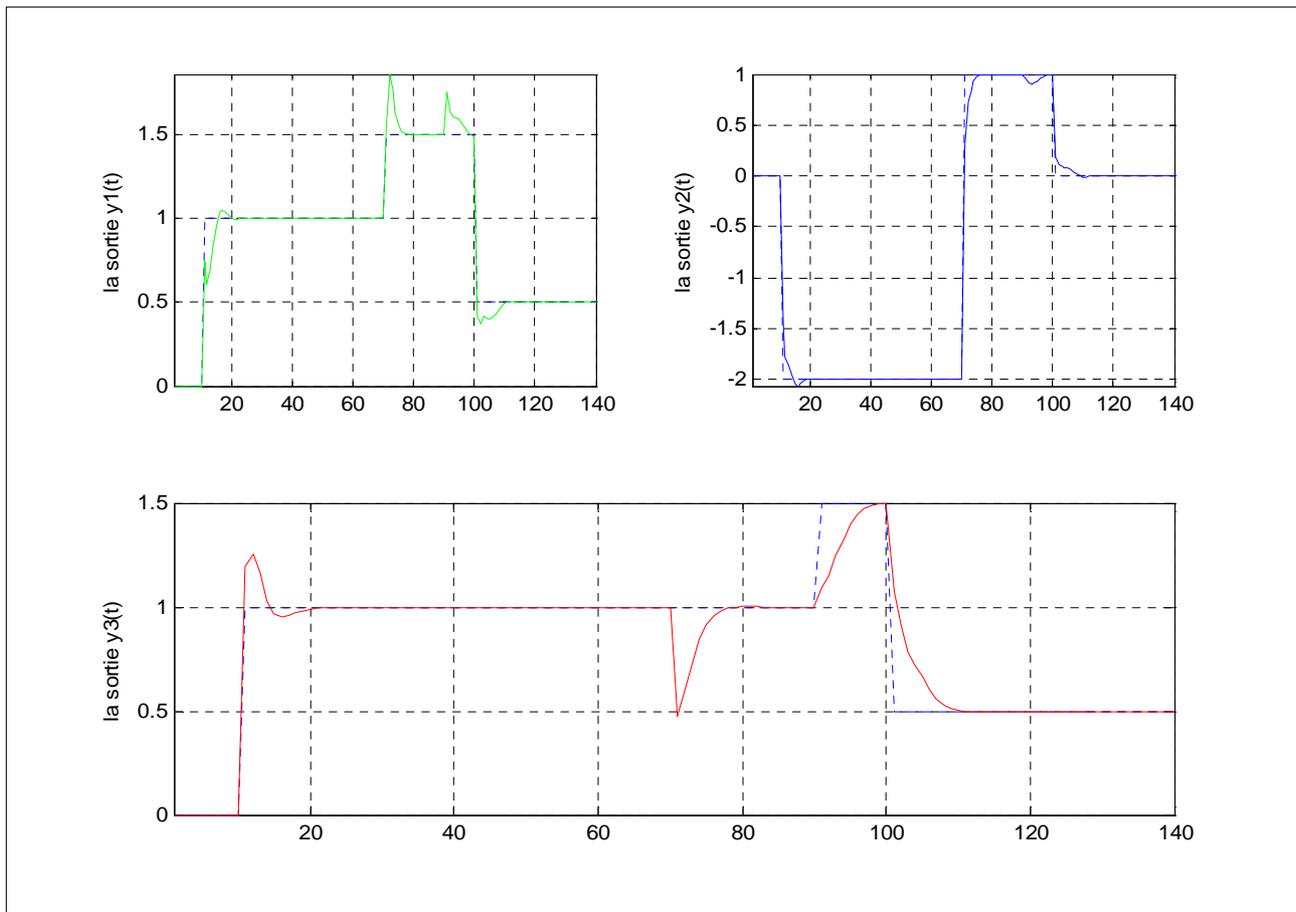
$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} -0.88 & -0.16 & 0 \\ 0.32 & -0.96 & -0.8 \\ 0 & 0 & -0.72 \end{bmatrix} q^{-1} + \begin{bmatrix} 0.112 & 0.08 & 0 \\ 0 & 0.136 & 0.08 \\ 0.16 & 0 & 0.112 \end{bmatrix} q^{-2} + \begin{bmatrix} 0.064 & 0.016 & 0 \\ -0.08 & 0.072 & 0.24 \\ 0.16 & 0 & 0.0192 \end{bmatrix} q^{-3}$$

$$B = \begin{bmatrix} 0.5 & 0.2 & -0.5 \\ 2 & 0 & 0.3 \\ 0 & 0.9 & -0.4 \end{bmatrix} q^{-1} + \begin{bmatrix} 1 & 2 & 1 \\ -0.8 & 0.6 & 0.5 \\ 1 & 0.3 & 0.5 \end{bmatrix} q^{-2}; \quad C(q^{-1}) = I_{3 \times 3}.$$

On applique dans l'intervalle  $0 \leq t \leq 140$ , les trois références suivantes :

- Pour la voie 1 :  $W_1 = \begin{cases} 0: & 0 \leq t \leq 10 \\ 1: & 10 \pi t \leq 70 \\ 1.5: & 70 \pi t \leq 100 \\ 0.5: & 100 \pi t \leq 140 \end{cases}$  \* Pour la voie 2 :  $W_2 = \begin{cases} 0: & 0 \leq t \leq 10 \\ -2: & 10 \pi t \leq 70 \\ 1: & 70 \pi t \leq 100 \\ 0: & 100 \pi t \leq 140 \end{cases}$
- Pour la voie 3 :  $W_3 = \begin{cases} 0: & 0 \leq t \leq 10 \\ 1: & 10 \pi t \leq 90 \\ 1.5: & 90 \pi t \leq 100 \\ 0.5: & 100 \pi t \leq 140 \end{cases}$

Les paramètres de contrôle  $(10, 7, 0.9, 0.8, 0.1)$  ont été choisis pour  $(N_2, N_u, \lambda_1, \lambda_2, \lambda_3)$ .

Fig.1.5.1: Les sorties  $y_i$  ( $i=1, 2, 3$ )

Le gain statique de ce système est obtenu, en appliquant le théorème de la valeur finale, on obtient :

$$G(q^{-1})_{q^{-1} \rightarrow 1} = A^{-1}(q^{-1})_{q^{-1} \rightarrow 1} \times B(q^{-1})_{q^{-1} \rightarrow 1}$$

$$\text{D'où, l'inverse de ce gain devient égal à: } G^{-1}(1) = \begin{bmatrix} 0.5159 & 0.2683 & 0.7384 \\ -0.1320 & -0.2302 & -0.1391 \\ -0.3748 & 0.0803 & -1.6033 \end{bmatrix}$$

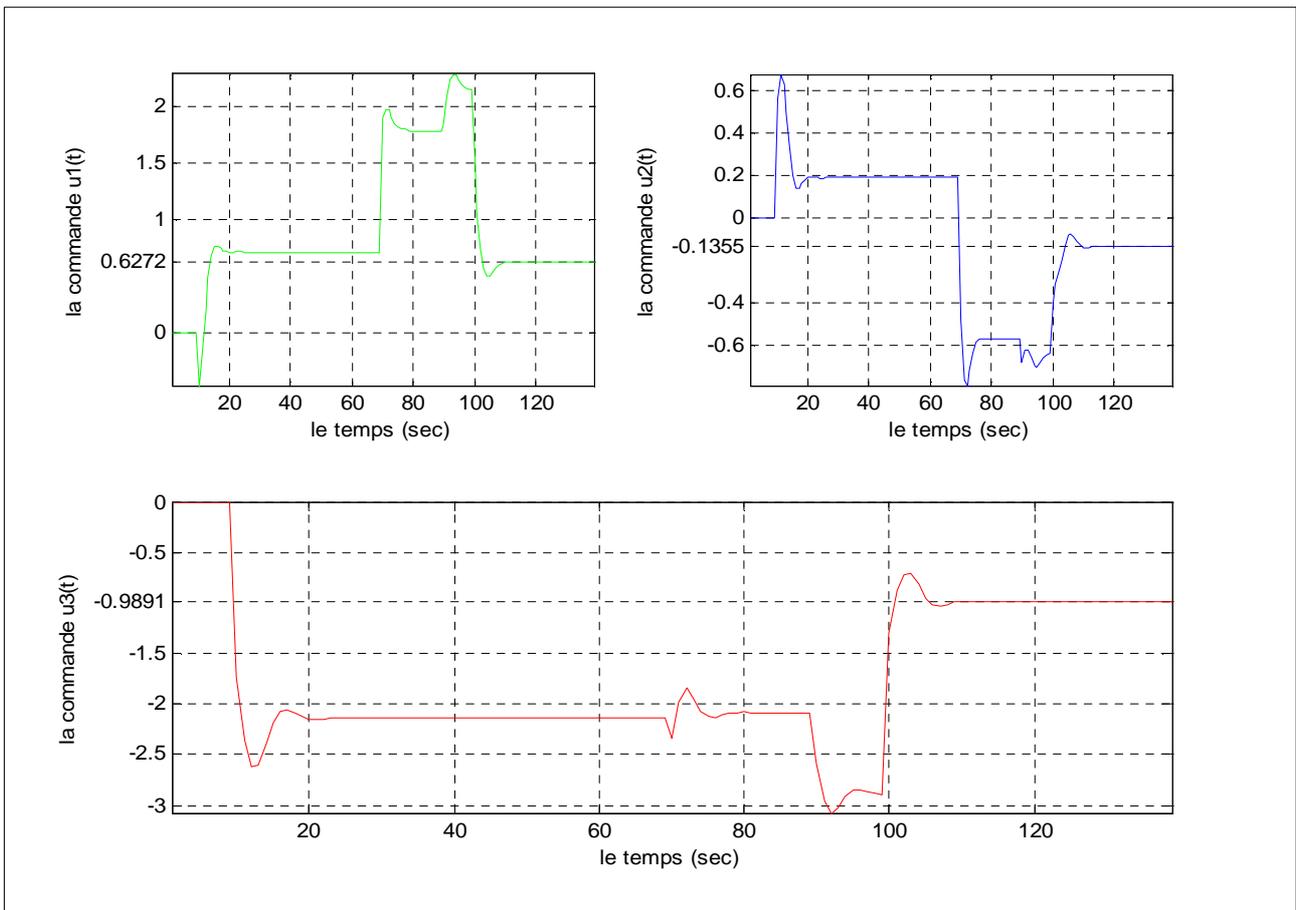


Fig.1.5.2 : Les commandes  $u_i$  ( $i=1, 2, 3$ )

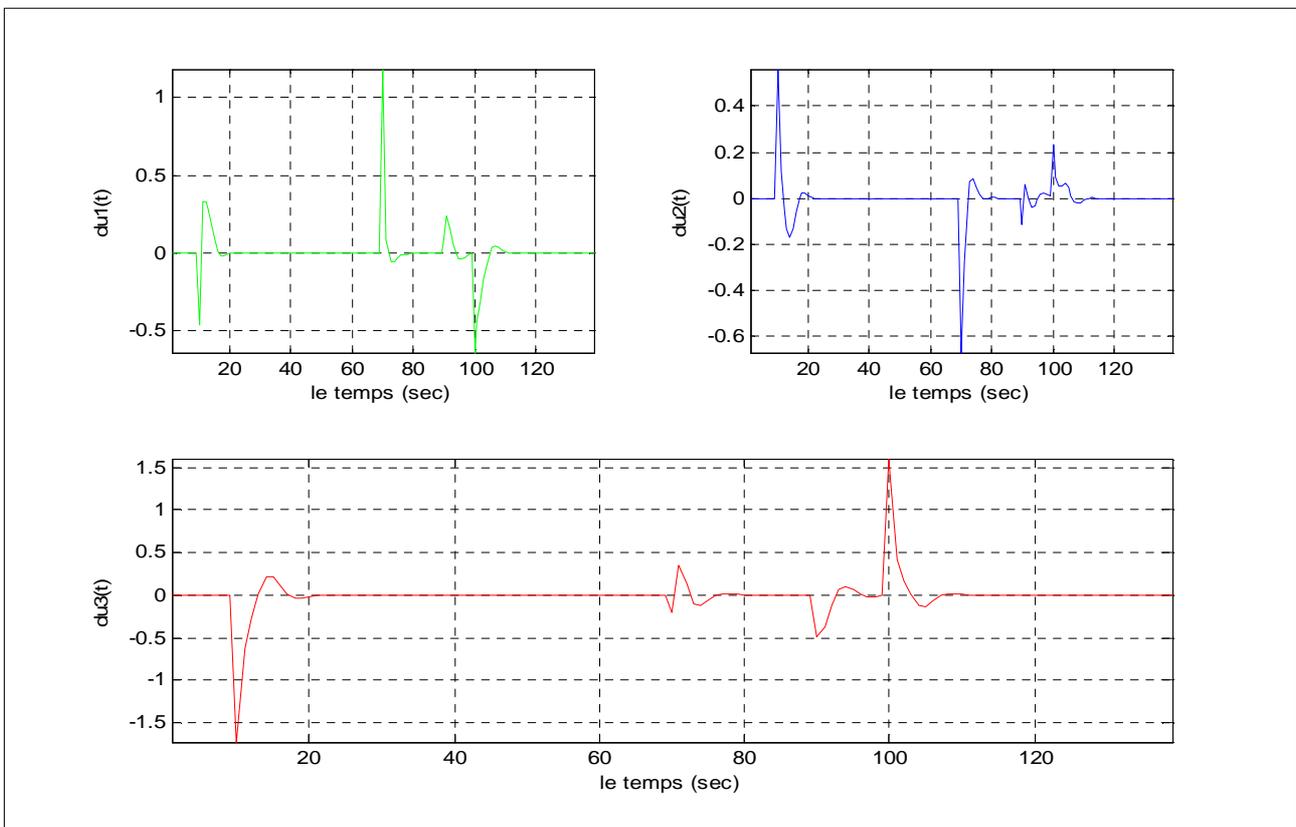


Fig.1.5.3 : Les incréments de commande  $\Delta u_i$  ( $i=1, 2, 3$ )

D'après les résultats obtenus, on peut remarquer, dans le régime stationnaire:

1. Des sorties tendent vers des valeurs constantes (valeurs de consignes).
2. Des commandes stables tendent vers les valeurs  $(0.6272 \quad -0.1355 \quad -0.9891)^T$ , ces dernières sont obtenues à partir de la multiplication de  $G^{-1}(1)$  par le vecteur de consignes

$$w_{t \rightarrow 100} = (0.5 \quad 0 \quad 0.5)^T$$

#### 2.4. Conclusion:

Dans ce chapitre, nous avons présenté l'algorithme de la commande prédictive généralisée multivariable algorithmique sans contraintes, cette dernière est appliquée sur deux systèmes multivariables.

Le premier système est présenté par une matrice de transfert d'ordre 2, l'application de cette méthode de commande nécessite à l'avance de la factorisation de cette matrice de transfert pour la rendre à un modèle *MISO* d'ordre 2 de type *CARIMA*. La dynamique de l'autre système est décrite par un modèle *MIMO* de *CARIMA* d'ordre 3 et les résultats obtenus montrent l'efficacité de cette méthode de commande, ceci grâce à un choix particulier des paramètres de synthèse (les horizons de prédiction minimal  $N_1$  et maximal  $N_2$ , les horizons de commande  $N_u$ , la matrice de pondération  $\Lambda$ ), ce choix nous a permis d'avoir des commandes stables et d'une bonne poursuite du signal de sortie à la consigne pour chaque voie. Cette méthode du fait qu'elle utilise des formes récursives, facilite considérablement son implémentation dans un calculateur numérique.

Cette commande présente aussi un avantage très intéressant qui réside dans l'élimination des effets de toutes perturbations en échelon de grandeur, de durée d'occurrence aléatoires grâce à l'action intégrale du modèle de *CARIMA*.

## 2. Introduction :

L'application de la loi de GPC pluridimensionnelle dans sa version algorithmique sur les systèmes multivariable présente un certain nombre de difficultés qui leurs sont inhérentes. Parmi celles-ci, nous mentionnons les lois de commande pour le cas des procédés représentés par des matrices de transfert d'ordre très élevé, ce qui nécessite un temps de calcul important de cette commande ; c'est pourquoi nous allons développer une autre version polynomiale de cette loi qui permet, d'une part, de réduire ce temps de calcul et, d'un autre part, de vérifier la stabilité du système corrigé dans le plan fréquentiel.

Cette partie a pour le but de développer la loi de commande obtenue avec le GPC algorithmique et que résume la fig.2.1, sous une forme équivalente linéaire à l'aide de trois matrices de polynômes  $R$ ,  $S$ ,  $T$  comme indiqué en fig.2.2.

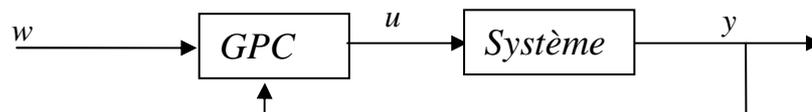


Fig.2.1 : GPC sous forme algorithmique

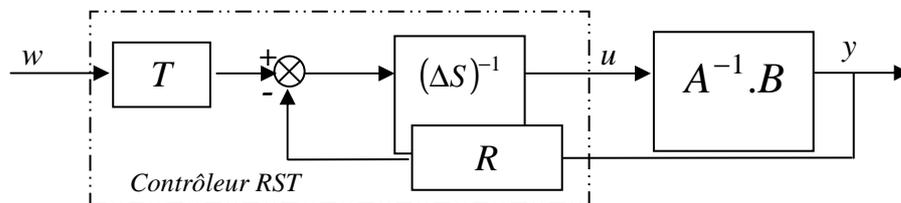


Fig.2.2 : GPC sous forme polynomiale équivalente

Les avantages procurés par l'obtention de la forme polynomiale  $RST$  équivalente au  $GPC$  sont multiples :

- L'utilisateur peut étudier a priori la stabilité du système en fonction des paramètres  $N_1$ ,  $N_2$ ,  $N_w$ ,  $\lambda$ . On peut alors garantir la stabilité du système nominal. Ce n'est pas le cas avec la version algorithmique de  $GPC$ , pour laquelle l'utilisateur n'a d'autres certitudes que celles qui lui sont données par le nombre restreint de théorèmes généraux de stabilité relatifs au  $GPC$ .

- Il est possible dans le cas multivariable de procéder à une étude de stabilité robuste par le tracé des valeurs singulières du système. On peut estimer le degré de robustesse en stabilité du système.
- Cette approche permet d'analyser le correcteur généré par le jeu de paramètres  $N_1, N_2, N_w, \lambda$  choisis par l'utilisateur : on peut donc obtenir les pôles et les zéros du correcteur, étudier son conditionnement...
- Les polynômes matriciels étant obtenus une fois pour toute hors ligne, l'approche polynomiale du *GPC* apparaît plus rapide à l'exécution que la version algorithmique. (Ce qui peut être intéressant dans le cas d'utilisation de périodes d'échantillonnage courtes).

### 2.1. Obtention du correcteur polynomial équivalent:

Par rapport à l'approche algorithmique, où l'on cherche à minimiser les calculs effectués en ligne, donc à ne pas résoudre les deux équations diophantiennes que peuvent nécessiter le calcul de la prédiction, l'approche polynomiale effectue le calcul de la prédiction à l'aide des équations de Diophantienne résolues hors ligne.

De plus des notations du paragraphe I.2.5, on utilise dans ce paragraphe [CB93]:

$$IF = \begin{bmatrix} F_{N_1}(q^{-1}) \\ F_{N_1+1}(q^{-1}) \\ \vdots \\ \vdots \\ F_{N_2}(q^{-1}) \end{bmatrix}, IH = \begin{bmatrix} H_{N_1}(q^{-1}) \\ H_{N_1+1}(q^{-1}) \\ \vdots \\ \vdots \\ H_{N_2}(q^{-1}) \end{bmatrix} \quad (2.1)$$

Explicitons ces notations :  $IF$  et  $IH$  sont des matrices blocs polynomiales telles que :

$$IF = \begin{bmatrix} F_0^{N_1} + F_1^{N_1} q^{-1} + \dots + F_{na}^{N_1} q^{-na} \\ \vdots \\ \vdots \\ F_0^{N_2} + F_1^{N_2} q^{-1} + \dots + F_{na}^{N_2} q^{-na} \end{bmatrix}, IH = \begin{bmatrix} H_0^{N_1} + H_1^{N_1} q^{-1} + \dots + H_{nh}^{N_1} q^{-nh} \\ \vdots \\ \vdots \\ H_0^{N_2} + H_1^{N_2} q^{-1} + \dots + H_{nh}^{N_2} q^{-nh} \end{bmatrix}$$

La mise sous forme matricielle de l'équation de prédiction (1.15), en utilisant les notions précédentes, soit alors:

$$\hat{Y} = G.\tilde{U} + IH.\Delta u(t-1) + IF.y(t) \quad (2.2)$$

Le critère (1.19) peut donc s'écrire cette fois :

$$J = \left[ G\tilde{U} + IH.\Delta u(t-1) + IF.y(t) - W \right]^T \left[ G\tilde{U} + IH.\Delta u(t-1) + IF.y(t) - W \right] + \tilde{U}^T \Lambda \tilde{U} \quad (2.3)$$

La loi de commande obtenue par minimisation du critère précédant vaut :

$$\tilde{U}_{opt} = -IM \left[ IF.y(t) + IH.\Delta u(t-1) - W \right] \quad (2.4)$$

$$\text{Avec } IM = \left( G^T G + \Lambda \right)^{-1} G^T = \begin{pmatrix} M_1 \\ \vdots \\ M_{N_u} \end{pmatrix}$$

Remarquons que nous avons  $W$  et non  $w(t)$ .

De la même façon qu'au §1.2.5 seules les  $m$  composantes de  $\tilde{U}_{opt}$  sont appliquées au système soit :

$$u(t) = u(t-1) + M_1 \left[ W - IF.y(t) - IH.\Delta u(t-1) \right] \quad (2.5)$$

Cette dernière équation peut s'écrire :

$$\left[ I_m + q^{-1}.M_1.IH \right] \Delta u(t) = -M_1.IF.y(t) + M_1.W \quad (2.6)$$

Le correcteur équivalent ayant pour équation :

$$S(q^{-1}).\Delta u(t) = T(q).w(t) - R(q^{-1}).y(t) \quad (2.7)$$

On a alors par identification des deux équations (2.6) et (2.7), les égalités :

$$\begin{aligned} S(q^{-1}) &= I_m + q^{-1}.M_1.IH \\ R(q^{-1}) &= M_1.IF \end{aligned} \quad (2.8)$$

$$T(q^{-1}) = M_1 \cdot \begin{bmatrix} q^{N_1} I_m \\ \vdots \\ q^{N_2} I_m \end{bmatrix}$$

Ces trois polynômes matriciels sont calculés hors ligne. Ainsi la consigne  $w(t)$  étant supposée connue (entre  $t+N_1$  et  $t+N_2$ ), on effectue l'acquisition de  $y(t)$ . Le calcul de la commande est alors immédiat d'après la relation (2.7).

## 2.2. GPC multivariable utilisant le filtre $C(q^{-1})$ [CB93] :

Nous avons pris pour la partie ci-dessus  $C(q^{-1}) = I_m$  dans l'équation du modèle de CARIMA. C'est-à-dire  $\hat{C}(q^{-1}) = \Delta_m^{-1}$ . Il se trouve qu'en procédant de cette façon, on cherche à annuler toute erreur statique vis-à-vis d'une perturbation constante en introduisant dans le modèle CARIMA un intégrateur. Cependant le modèle de perturbation choisi  $C(q^{-1}) = I_m$  n'est pas suffisamment représentatif de celui qui peut exister réellement sur un processus industriel pour lequel il peut exister des perturbations hautes fréquences mal connues ou complètement négligées.

Le but de ce qui suit est de prendre un modèle du processus sous forme CARIMA pour lequel on choisit un modèle explicite de perturbations  $C(q^{-1}) \neq I_m$  afin de mieux décrire la réalité. On verra par la suite qu'un choix possible pour  $C(q^{-1})$  est tel que son gain est élevé en haute fréquence : en général les dynamiques négligées ou mal connues le sont d'autant plus que la fréquence considérée est élevée.

Nous allons réutiliser les notations au cas de  $C(q^{-1}) = I_m$ . L'équation du modèle sous forme CARIMA s'écrit :

$$\Delta_m A(q^{-1})y(t) = B(q^{-1})\Delta_m u(t-1) + C(q^{-1})\zeta(t) \quad (2.9)$$

Il va être tenu compte de ce modèle dans l'estimation de la prédiction. Avec les notations du 1.2.4, l'équation (1.18) donnant la prédiction s'écrit :

$$C_j y(t+j) = C_j G_j \Delta u(t+j-1) + [F_j y(t) + H_j \Delta u(t-1)] + E_j C_j \zeta(t+j) \quad (2.10)$$

Or avec (2.9) égal à :

$$\Delta_m A(q^{-1})y(t+j) = B(q^{-1})\Delta_m u(t+j-1) + C(q^{-1})\zeta(t+j) \quad (2.11)$$

Et les deux équations de diophantiennes :

$$\begin{aligned} C &= E_j A \Delta + q^{-j} F_j \\ E_j B &= G_j C + q^{-j} H_j \end{aligned} \quad (2.12)$$

$$\text{Avec : } \begin{cases} E_j = E_0^j + E_1^j q^{-1} + \dots + E_{j-1}^j q^{-j+1} \\ F_j = F_0^j + F_1^j q^{-1} + \dots + F_{n_f^{(j)}}^j q^{-n_f^{(j)}} \\ n_f^{(j)} = \max(n_a, n_c - j) \end{cases}$$

$$\text{Et } \begin{cases} G_j = G_0^j + G_1^j q^{-1} + \dots + G_{j-1}^j q^{-j+1} \\ H_j = H_0^j + H_1^j q^{-1} + \dots + H_{n_h}^j q^{-n_h} \\ n_h = \max(n_c, n_b + d) - 1 \end{cases}$$

Le prédicteur optimal est donc :

$$C\hat{y}(t+j) = G_j \cdot C \cdot \Delta u(t+j-1) + [H_j \Delta u(t-1) + F_j y(t)] \quad (2.13)$$

Posons maintenant [CB93] :

$$\hat{Y}_F = \begin{bmatrix} C\hat{y}(t+N_1) \\ \vdots \\ C\hat{y}(t+N_2) \end{bmatrix}, W_F = \begin{bmatrix} Cw(t+N_1) \\ \vdots \\ Cw(t+N_2) \end{bmatrix} \text{ et } \tilde{U}_F = \begin{bmatrix} C\Delta u(t+N_1) \\ \vdots \\ C\Delta u(t+N_u-1) \end{bmatrix} \quad (2.14)$$

La minimisation analytique du critère conduit à la séquence optimale de commandes futures:

$$\tilde{U}_{Fopt} = -(G^T G + \Lambda)^{-1} G^T [IFy(t) + IH\Delta u(t-1) - W_F] \quad (2.15)$$

$$\text{Avec : } \tilde{U}_{Fopt} = \begin{bmatrix} \tilde{U}_{Fopt}(t) \\ \vdots \\ \tilde{U}_{Fopt}(t+N_u-1) \end{bmatrix} = \begin{bmatrix} C\tilde{U}_{opt}(t) \\ \vdots \\ C\tilde{U}_{opt}(t+N_u-1) \end{bmatrix} \quad (2.16)$$

La commande réellement appliquée est définie par :

$$u(t) = u(t-1) - C^{-1} \cdot M_1 [IF \cdot y(t) + IH \cdot \Delta u(t-1) - W_F] \quad (2.17)$$

Le correcteur se déduit alors comme suit :

$$(C + M_1.IH.q^{-1})\Delta u(t) = M_1.W_F - M_1.IF.y(t) \quad (2.18)$$

Par identification avec l'équation :

$$\Delta S'(q^{-1})u(t) = T'(q^{-1})W(t) - R'(q^{-1})y(t) \quad (2.19)$$

$$S'(q^{-1}) = C(q^{-1}) + M_1IHq^{-1}$$

$$\text{Il vient : } R'(q^{-1}) = M_1IF \quad (2.20)$$

$$T'(q^{-1}) = M_1 \begin{bmatrix} q^{N_1} I_m \\ \vdots \\ q^{N_2} I_m \end{bmatrix} C(q^{-1})$$

On peut constater que la matrice polynomiale  $C(q^{-1})$  se trouve au niveau du correcteur dans  $S'(q^{-1})$ ,  $T'(q^{-1})$  et dans  $R'(q^{-1})$  par l'intermédiaire de  $IF$ . Nous verrons plus loin que cette matrice polynomiale peut engendrer une amélioration de la robustesse tout en conservant un comportement d'entrées-sorties inchangé.

### 2.3. Analyse de la robustesse:

#### 2.3.1. Introduction :

Comme nous l'avons dit au paragraphe 2.1, la représentation de la version polynomiale du GPC est indispensable pour effectuer l'analyse de robustesse en stabilité du système. En effet, rappelons simplement que la version algorithmique ne fournit pas de façon explicite les correcteurs. Nous allons aborder un ensemble de théorèmes concernant des conditions de robustesse de stabilité d'un système bouclé. Nous pourrions ainsi remarquer que ces conditions font intervenir les fonctions de sensibilité  $S(w)$  et de sensibilité complémentaire  $S_c(w)$  caractérisant aussi la performance nominale et la robustesse aux bruits de mesure du système bouclé face à des incertitudes paramétriques du modèle.

#### 2.3.2. Fonctions de sensibilité et sensibilité complémentaire:

On considère le système bouclé discret multivariable de la fig.2.3 [CS99] :

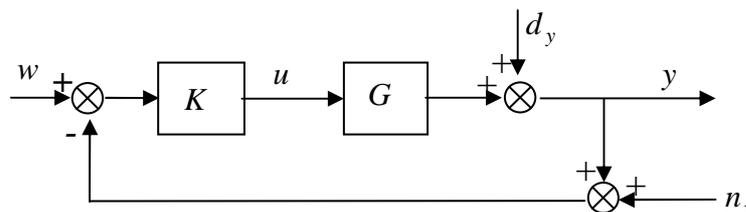


Fig.2.3 : Système bouclé classique

Avec  $n_y$  représente les entrées de bruits de mesure,  $d_y$  la perturbation agissant en sortie du système.

La sortie du système bouclé  $y$  peut s'écrire :

$$y = G.K(I_m + G.K)^{-1}.w - G.K(I_m + G.K)^{-1}.n_y + (I_m + G.K)^{-1}.d_y \quad (2.21)$$

L'erreur d'asservissement  $\varepsilon = w - y$ , compte tenu de l'identité :

$$G.K(I_m + G.K)^{-1} + (I_m + G.K)^{-1} = I_m \quad (2.22)$$

Est alors :

$$\varepsilon = (I_m + G.K)^{-1}.w + G.K(I_m + G.K)^{-1}.n_y - (I_m + G.K)^{-1}.d_y \quad (2.23)$$

On définit alors d'après les relations (2.22) et (2.23) [FD97]:

- La fonction de sensibilité :  $S_d = (I_m + G.K)^{-1}$
- La fonction de sensibilité complémentaire :  $S_c = G.K(I_m + G.K)^{-1}$

Ces deux matrices de transfert vérifient la relation [FD97] :

$$S_d + S_c = I_m \quad (2.24)$$

### 2.3.3. Théorème du faible gain [FD97] :

Ce théorème est d'une portée moins générale que le théorème de Nyquist puisqu'il ne donne qu'une condition suffisante de stabilité. Souvent énoncé à propos des systèmes continus, le théorème du faible gain est également valide en discret.

#### 2.3.3.1. Hypothèse:

Considérons  $\Delta$  une perturbation stable, et  $P$  la fonction de transfert du système entre l'entrée et la sortie de la perturbation supposée stable également.

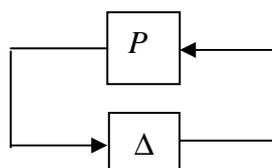


Fig.2.4 : Schéma général d'étude de la robustesse

$P$  est le transfert qui apparaît lorsqu'on isole dans un bloc  $\Delta$  toutes les perturbations ou incertitudes de modèle.

**2.3.3.2. Théorème [FD97]:**

- Le système bouclé est stable en présence de la perturbation  $\Delta$  si  $\|P.\Delta\|_{\infty} < 1$

où  $\|\cdot\|_{\infty}$  représente la norme  $H_{\infty}$ .

- Une autre condition de stabilité également suffisante mais plus conservatrice est la suivante :

$$\|P\|_{\infty} \cdot \|\Delta\|_{\infty} < 1 \tag{2.25}$$

**2.3.4. Robustesse « stabilité/performances » d'un système soumis à des incertitudes paramétriques:**

Nous rappelons brièvement ici les modélisations couramment employées des incertitudes pouvant exister sur un modèle représentant un processus physique. Le principal intérêt de telles modélisations est qu'elles sont très générales, et permettent donc de modéliser de façon très simple des incertitudes aussi diverses que des dynamiques mal connues d'actionneurs ou des variations paramétriques de modèle.

La représentation non structurée des incertitudes de modèle affectant un système peut être donnée par [CB93]:

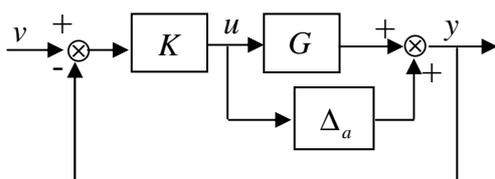


Fig.2.5a:  $G_p = G + \Delta_a$

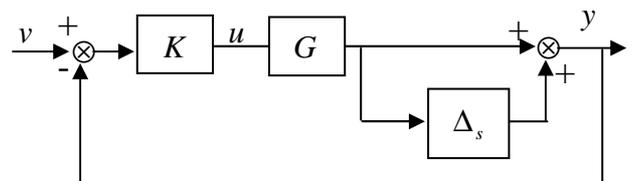


Fig.2.5b:  $G_p = (I_m + \Delta_s)G$

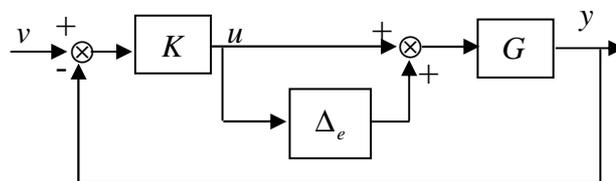


Fig.2.5c:  $G_p = G(I_m + \Delta_e)$

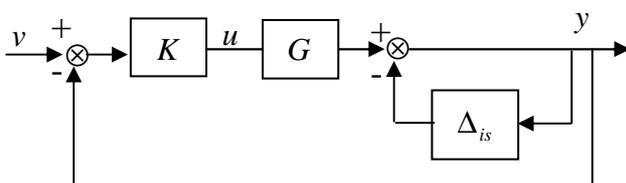


Fig.2.5d:  $G_p = (I_m + \Delta_{is})^{-1}G$

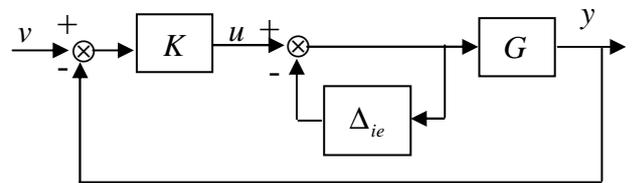


Fig.2.5e:  $G_p = G(I_m + \Delta_{ie})^{-1}$

Nous allons appliquer le théorème du faible gain aux diverses représentations obtenues au paragraphe précédent afin de déduire la condition de robustesse sur la stabilité. Nous notons :

$\bar{\sigma}(\cdot)$  : La valeur singulière maximale.

$\Delta_x$  : La matrice d'incertitudes obtenue à partir des figures 2.5, généralement sa structure est inconnue ; elle satisfait une certaine limite supérieure :

$$\forall \omega \in \left[ 0 \quad \frac{\pi}{T_e} \right]; \bar{\sigma}[\Delta_x(\omega)] \leq \bar{\sigma}[L_m(\omega)] \quad (2.26)$$

$L_m(\omega)$  : est une fonction scalaire qui limite la matrice de transfert du système perturbé au voisinage de nominale

- Avec les hypothèses ci-dessus, la condition de robustesse sur la stabilité est donnée par :

$$\begin{aligned} \forall \omega \in \left[ 0 \quad \frac{\pi}{T_e} \right]; \quad & \bar{\sigma}[L_m(\omega)] \bar{\sigma}[S_c(\omega)] \leq 1 \\ & \Rightarrow \bar{\sigma}[S_c(\omega)] \leq \frac{1}{\bar{\sigma}[L_m(\omega)]} \end{aligned} \quad (2.27)$$

- Les performances nominales (NP) sont ainsi définies comme le poids maximal de la matrice de sensibilité d'où la condition de robustesse sur les performances est donnée par:

$$\begin{aligned} \forall \omega \in \left[ 0 \quad \frac{\pi}{T_e} \right]; \quad & \bar{\sigma}[w_s(\omega)] \bar{\sigma}[S_d(\omega)] \leq 1 \\ & \Rightarrow \bar{\sigma}[S_d(\omega)] \leq \frac{1}{\bar{\sigma}[w_s(\omega)]} \end{aligned} \quad (2.28)$$

$w_s(\omega)$  : représente les spécifications sur les performances.

### 2.3.5. Amélioration de la robustesse :

On note  $RST$ ,  $R'S'T'$  les correcteurs obtenus pour  $C(q^{-1}) = I_m$  et  $C(q^{-1}) \neq I_m$  respectivement.

Il est aisé de constater que le schéma (2.2) donnant la structure polynomiale équivalente du GPC peut se mettre sous la forme :

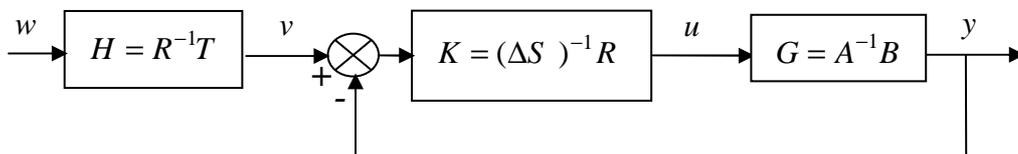


Fig.2.6 : Structure polynomiale au GPC pour  $C(q^{-1})=I_m$

- La matrice de transfert entre l'entrée de référence la sortie et est égale, avant introduction de  $C(q^{-1})$  à :

$$G_{bf} = G.K(I_m + G.K)^{-1}.H$$

Or d'après l'identité [CB93]:

$$G.K(I_m + G.K)^{-1} = [I_m + (G.K)^{-1}]^{-1}$$

Il vient :

$$G_{bf} = [I_m + (G.K)^{-1}]^{-1}.H$$

$$\text{D'où : } G_{bf} = \left[ I_m + (A^{-1}.B.(\Delta.S)^{-1}.R) \right]^{-1} .R^{-1}.T = \left[ R + \Delta.S.B^{-1}.A \right]^{-1} .T \quad (2.29)$$

- La dynamique de la boucle fermée est donc déterminée par  $P_c = [R + \Delta.S.B^{-1}.A]$  avant introduction de  $C(q^{-1})$ ; la dynamique est donc définie par les paramètres  $N_2, N_u, \Lambda$  définissant  $R(q^{-1})$  et  $S(q^{-1})$ .
- Après ajout de  $C(q^{-1})$ , nous avons constaté que les éléments  $R(q^{-1})$ ,  $S(q^{-1})$  et  $T(q)$  deviennent  $R'(q^{-1})$  ( $IF$  dépend de  $C(q^{-1})$ ),  $S'(q^{-1})$  et  $T'(q) = T(q).C(q^{-1})$ . La matrice de transfert entre la sortie et l'entrée vaut :

$$G'_{bf} = (R' + \Delta.S'.B^{-1}.A)^{-1}.T.C \quad (2.30)$$

Remarquons également que si l'on ne souhaite pas modifier le comportement entrée-sortie après l'introduction du filtre  $C(q^{-1})$ , il faut que  $C(q^{-1})$  soit scalaire puisque  $y(t)$  est remplacé par  $y_F(t)$  et  $w(t)$  par  $w_F(t)$  donc :

$$y_F = G'_{bf}.w_F \Leftrightarrow y = C^{-1}.G'_{bf}.C.w \Leftrightarrow y = G'_{bf}.w \Leftrightarrow G_{bf} = G'_{bf} \quad (2.31)$$

Donc la fonction de transfert (2.30) doit être égale à celle obtenue avant introduction de  $C(q^{-1})$  soit  $(R + \Delta.S.B^{-1}.A)^{-1}T$  ce qui impose que l'on ait :

$$R' + \Delta.S'.B^{-1}.A = CP_c \quad (2.32)$$

$$\text{Avec : } P_c = R + \Delta.S.B^{-1}.A$$

La condition de robustesse sur la stabilité, avant l'ajout de  $C(q^{-1})$ , est exprimé

En résumé, l'adjonction du filtre matriciel  $C(q^{-1})$  se décompose en deux actions :

- On modifie la dynamique de la boucle fermée qui passe de  $P_c = (R + \Delta.S.B^{-1}.A)$  à  $(R' + \Delta.S'.B^{-1}.A) = CP_c$
- On filtre la consigne  $W$  de façon à garder un comportement entrée sortie inchangé (avec  $C(q^{-1})$  scalaire pour assurer l'identité du comportement entrée-sortie).
- Les conditions de robustesse sur la stabilité sont exprimées en fonction de la matrice de transfert en boucle fermée, avant et après l'ajout de  $C(q^{-1})$ , par les équations (2.33) et (234) comme suit:

$$\forall \omega \in \left[ 0 \quad \frac{\pi}{T_e} \right]; \overline{\sigma}[H^{-1}.G_{bf}(\omega)] \pi \quad 1/\overline{\sigma}[L_m(\omega)] \quad (2.33)$$

$$\forall \omega \in \left[ 0 \quad \frac{\pi}{T_e} \right]; \overline{\sigma}[H'^{-1}.G_{bf}(\omega)] \pi \quad 1/\overline{\sigma}[L_m(\omega)] \quad (2.34)$$

$$\text{Avec : } H = R^{-1}.T \text{ et } H' = R'^{-1}.T.C.$$

D'après ces dernières équations, si l'on veut améliorer la robustesse du système, il faudra bien choisir la matrice des polynômes  $C(q^{-1})$ .

### 2.3.6. Rôle de la matrice $C(q^{-1})$ [Rodr03]:

On peut analyser le rôle de  $C(q^{-1})$  de deux façons. Dans le cas où l'on connaît  $C(q^{-1})$  du système ou une estimation du bruit agissant sur le système, le  $C(q^{-1})$  sert à réaliser une prédiction optimale de la sortie dans le sens de la minimisation de la variance. Dans ce cas, on peut voir  $C(q^{-1})$  comme un filtrage, permettant d'atténuer l'erreur de prédiction.

D'autre part, on peut se servir de  $C(q^{-1})$  comme un paramètre du système servant à rendre robuste la commande.  $C(q^{-1})$  Peut être vu, alors, comme un observateur ou un pré filtrage du système. Dans ce cas, on perd l'optimalité dans la prédiction, mais on augmente la robustesse.

#### 2.4. Tests et simulations :

Les systèmes considérés ensuite, sont ceux étudiés dans le chapitre précédent plus un autre système disponible dans la bibliothèque de Matlab/Robuste.

La première application est effectuée dans le but de voir le tracé des valeurs singulières maximales de la fonction de sensibilité et la sensibilité complémentaire dans le plan fréquentiel. L'application suivante exploite des données disponibles dans le logiciel de Matlab d'un système commandé par une méthode robuste, ceci dans l'objectif de vérifier la robustesse de la méthode GPC polynomiale face aux incertitudes paramétriques du modèle.

Dans la dernière application, on utilise pour le calcul des paramètres de GPC polynomial, la matrice des polynômes  $C(q^{-1})$  différente de l'unité celle-ci dans le but de voir l'impact des signaux de bruits ainsi de perturbations sur les commandes et les sorties du système bouclé. Les résultats obtenus sont ensuite comparé à ceux obtenus par le contrôleur RST non robustifié.

#### Application1: « GPC polynomial »

Le système considéré ici est celui présenté dans §1.3.1, on choisit les mêmes paramètres de synthèse de GPC algorithmique, cités dans le chapitre précédent, pour déterminer les paramètres du contrôleur RST équivalent on obtient :

- Pour  $R(q^{-1}) = \begin{bmatrix} R_{11}(q^{-1}) & R_{12}(q^{-1}) \\ R_{21}(q^{-1}) & R_{22}(q^{-1}) \end{bmatrix}$  :

$$R_{11}(q^{-1}) = -3.786 + 6.114q^{-1} - 2.495q^{-2} ; \quad R_{12}(q^{-1}) = 4.551 - 6.421q^{-1} + 2.465q^{-2}$$

$$R_{21}(q^{-1}) = 19.37 - 27.76q^{-1} + 10.69q^{-2} ; \quad R_{22}(q^{-1}) = -1.135 + 1.76q^{-1} - 0.7003q^{-2}$$

- Pour  $S(q^{-1}) = \begin{bmatrix} S_{11}(q^{-1}) & S_{12}(q^{-1}) \\ S_{21}(q^{-1}) & S_{22}(q^{-1}) \end{bmatrix}$  :

$$S_{11}(q^{-1}) = 1 - 0.5519q^{-1} ; \quad S_{12}(q^{-1}) = -0.06464q^{-1}$$

$$S_{21}(q^{-1}) = -0.3533q^{-1} ; \quad S_{22}(q^{-1}) = 1 - 1.187q^{-1}$$

- Pour  $T(q^{-1}) = \begin{bmatrix} T_{11}(q^{-1}) & T_{12}(q^{-1}) \\ T_{21}(q^{-1}) & T_{22}(q^{-1}) \end{bmatrix}$  :

$$T_{11}(q^{-1}) = -0.1561 - 0.1053q^{-1} - 0.04068q^{-2} + 0.0394q^{-3} + 0.09485q^{-4}.$$

$$T_{12}(q^{-1}) = 0.0647 + 0.09108q^{-1} + 0.1186q^{-2} + 0.1474q^{-3} + 0.1732q^{-4}.$$

$$T_{21}(q^{-1}) = 0.7722 + 0.6551q^{-1} + 0.5321q^{-2} + 0.3433q^{-3}.$$

$$T_{22}(q^{-1}) = -0.07226 - 0.037q^{-1} + 0.0003947q^{-2} + 0.03394q^{-3}.$$

Avec cette transformation polynomiale, on peut voir les caractéristiques fréquentielles du système bouclé, par le tracer des lieux des valeurs singulières maximales de la fonction de sensibilité directe et la sensibilité complémentaire (voir Fig.2.7).

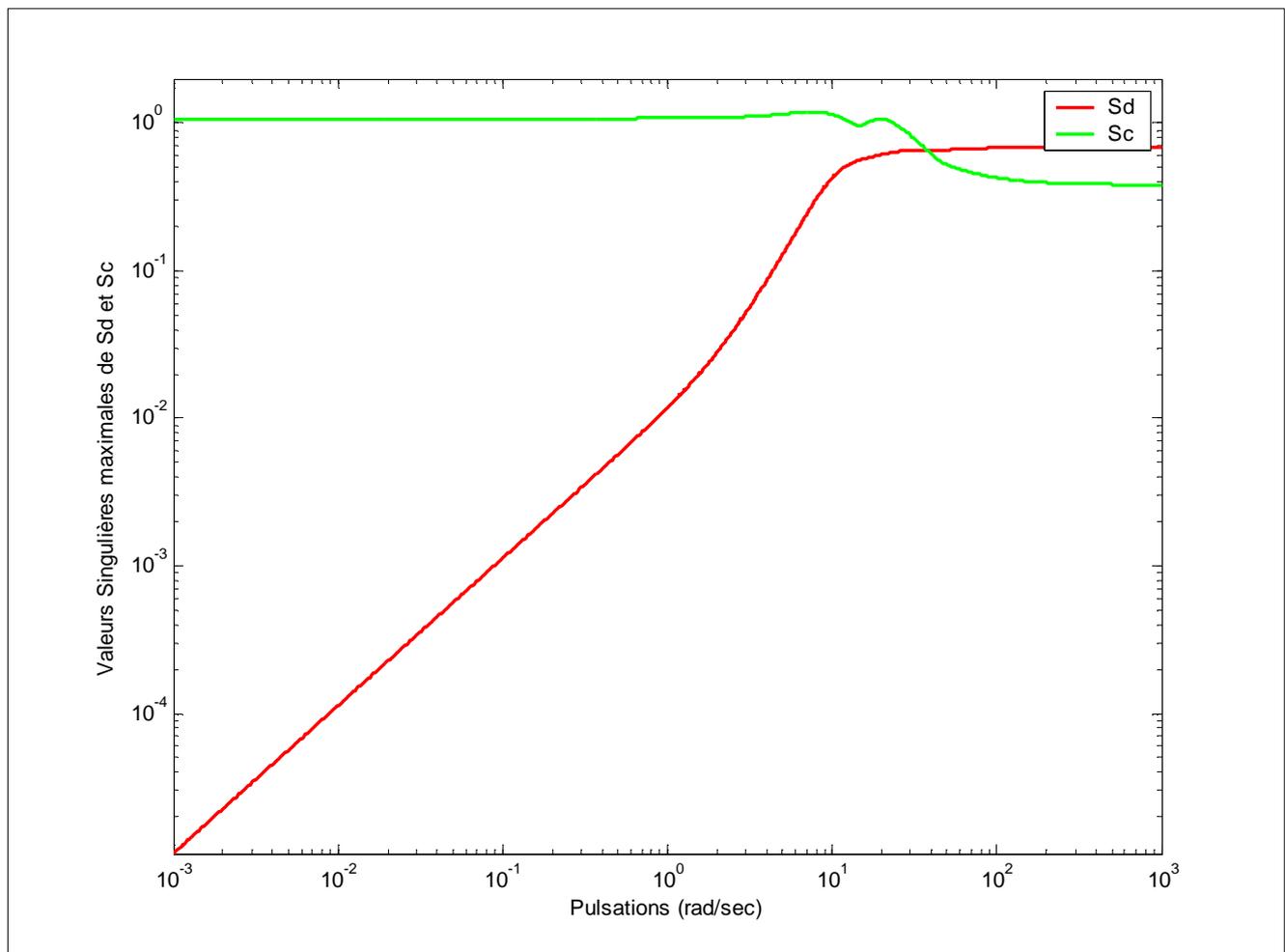


Fig.2.7 : les valeurs singulières de la sensibilité directe et la sensibilité complémentaire

D'après la figure ci-dessus on peut constater les remarques suivantes :

- Pour la fonction de sensibilité directe : Elle est faible pour les basses fréquences et proche de l'unité pour les hautes fréquences. Les valeurs faibles de la fonction de sensibilité directe pour les basses fréquences, qui sont nécessaires pour une bonne atténuation des erreurs de modèle,

assurent la protection contre les erreurs de modèle de l'inverse du transfert en boucle ouverte aux basses fréquences.

- Pour la fonction de sensibilité complémentaire: Elle proche de l'unité pour les basses fréquences et décroissante en haute fréquence. Il faut noter que, plus la fonction de sensibilité complémentaire décroît avec la fréquence, plus le système en boucle fermée devient protégé aux erreurs de modèle en haute fréquence. Ceci est important car du fait des dynamiques négligées- appelées aussi effet parasites. Dans notre cas, cette fonction ne décroît pas rapidement ceci traduit par la sensibilité du système bouclés aux incertitudes paramétriques du modèle.

- Aucune condition de robustesse examinée ici pour confirmer est ce que ce système est robuste en stabilité et/ou en performances. Pour vérifier cette robustesse, nous allons procéder, dans la partie suivante, à une comparaison avec un autre système commandé par la méthode robuste  $H_\infty$ .

#### 2.4.1. Application2: « analyse de robustesse »

Cette application à pour but de vérifier la robustesse du contrôleur RST\_GPC pour le cas d'un système multivariable décrit par un modèle incertain, pour cela, nous allons procéder à une étude comparative entre la méthode GPC polynomiale et la méthode  $H_\infty$ , on utilise pour cette étude les données disponibles dans le fichier *josedemo.m* de Matlab.

Notre objectif étant précisé ; nous n'allons pas s'attarder sur la conception d'un contrôleur par la méthode  $H_\infty$ , nous allons concentrer notre étude sur la vérification de la satisfaction de la condition de robustesse cités deux conditions de robustesse.

Le procédé sur lequel nous allons faire notre étude est décrit par les équations d'états et de mesures comme suit :

$$\begin{aligned}\frac{dx(t)}{dt} &= A_g \cdot x(t) + B_g u(t) \\ y(t) &= C_g x(t) + D_g u(t)\end{aligned}$$

$$\text{Avec : } A_g = \begin{bmatrix} -0.99005 & 4.7491 \times 10^{-4} & 0.48994 & 1.9219 \\ 9.0643 \times 10^{-4} & -0.98765 & 1.9010 & -0.49180 \\ -0.49605 & -1.9005 & -311.7 & 4.9716 \\ -1.9215 & 0.49069 & -7.7879 & -398.31 \end{bmatrix}, B_g = \begin{bmatrix} 0.78273 & -0.61398 \\ 0.61298 & 0.78260 \\ 0.78349 & 0.59597 \\ 0.60685 & -0.78784 \end{bmatrix};$$

$$C_g = \begin{bmatrix} 0.78291 & 0.61279 & -0.78163 & -0.6061 \\ -0.61238 & 0.782 & -5.9841 & -0.78842 \end{bmatrix}; D_g = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Les transferts de pondération  $W_1$  et  $W_3$  choisis pour la méthode  $H_\infty$  sont donnés par :

$$W_1^{-1} = \begin{bmatrix} \frac{0.01 \cdot \left(1 + \frac{s}{100}\right)^2}{\gamma \left(1 + \frac{s}{5000}\right)^2} & 0 \\ 0 & \frac{0.01 \cdot \left(1 + \frac{s}{100}\right)^2}{\gamma \left(1 + \frac{s}{5000}\right)^2} \end{bmatrix}, \quad W_3^{-1} = \begin{bmatrix} \frac{2000}{s} & 0 \\ 0 & \frac{2000}{s} \end{bmatrix}$$

Où  $\gamma$  est une pondération réalisée dans la méthode  $H_\infty$ , elle est choisie par l'utilisateur pour faire un compromis entre les robustesses stabilité-performances. L'analyse de cette dernière a été faite dans une plage fréquentielle définie par l'intervalle  $[10^{-2} \quad 10^{15}]$ .

En se basant sur le pas  $T_e = 0.43$  sec pour discrétiser la forme d'état donnée et on choisit pour les deux méthodes de commande présentées les paramètres de synthèse  $\gamma = 0.009$ ;  $N_{2_1} = 2$  ;

$N_{2_2} = 4$ ;  $N_{u_1} = N_{u_2} = 2$ , et  $\Lambda = \text{diag}(0.4, 1)$ , On obtient :

- Les figures (2.8.1), (2.8.2), représentent la comparaison des fonctions de sensibilités et de sensibilités complémentaires obtenues par le GPC polynomial et la méthode  $H_\infty$ .
- Les figures (2.8.3) et (2.8.4) représentent la réponse indicielle ainsi les commandes du système corrigé pour les deux vecteurs de consignes :  $w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  et  $w = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ .
- N.B : Les gains statiques du système à commander sont obtenus par l'application du théorème de la valeur finale avec la relation:

$$G^{-1}(s)_{s \rightarrow 0} = \left[ C_g \times (s \times I_{4 \times 4} - A_g)_{s \rightarrow 0}^{-1} \times B_g + D_g \right]^{-1} D \text{ où :}$$

$$G^{-1}(0) = \begin{bmatrix} 1.0029 & 0.00026 \\ 0.0036 & 1.0027 \end{bmatrix}$$

Pour une erreur de poursuite proche de zéros, les commandes doivent tendent vers :

$$u = \begin{bmatrix} 1.0029 & 0.00026 \\ 0.0036 & 1.0027 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1.0029 \\ 0.0036 \end{bmatrix} : \text{ Pour le vecteur de consignes } w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \text{ et}$$

$$u = \begin{bmatrix} 1.0029 & 0.00026 \\ 0.0036 & 1.0027 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.00026 \\ 1.0027 \end{bmatrix} : \text{ Pour le vecteur de consignes } w = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

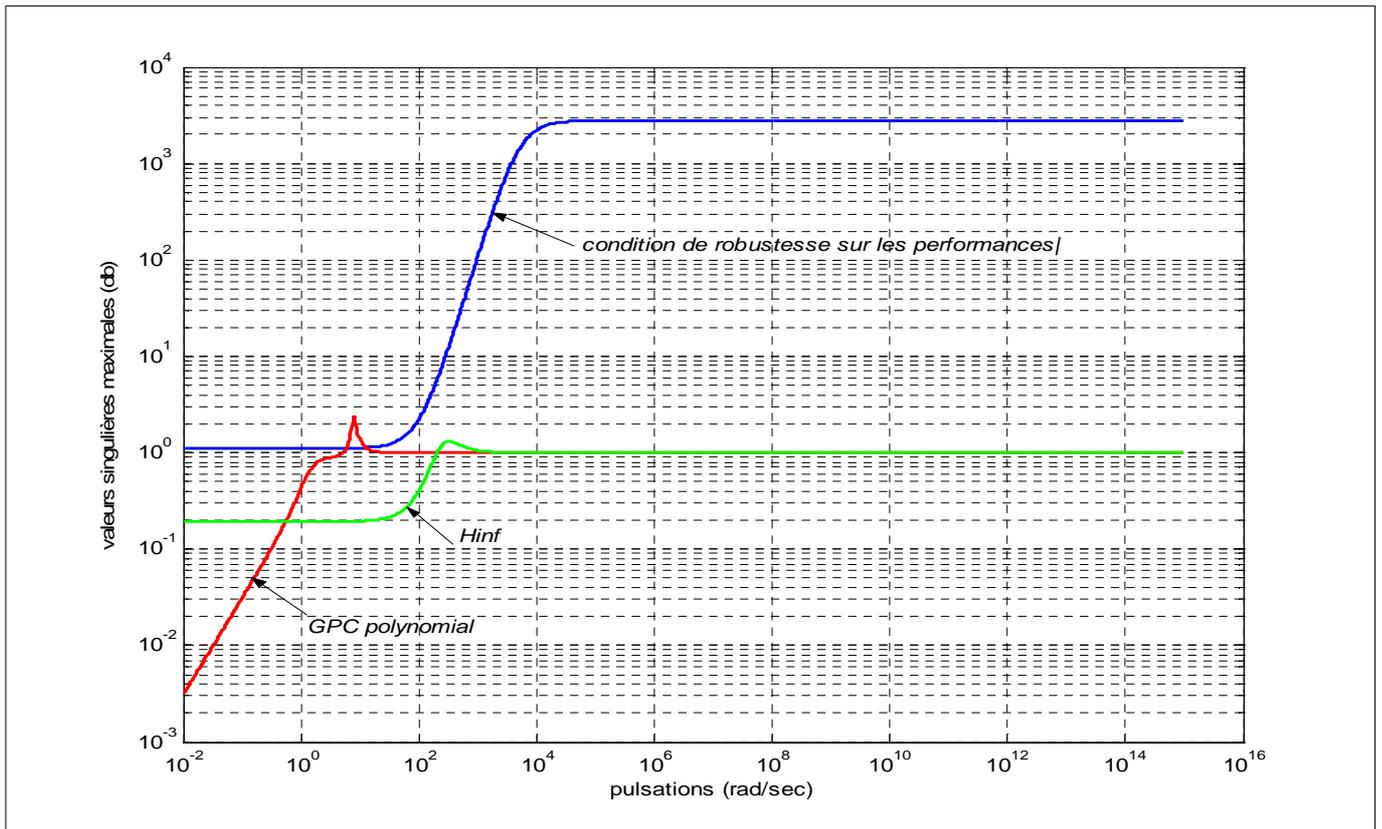


Fig.2.8.1 : sensibilité directe et condition de robustesse sur les performances (comparaison)

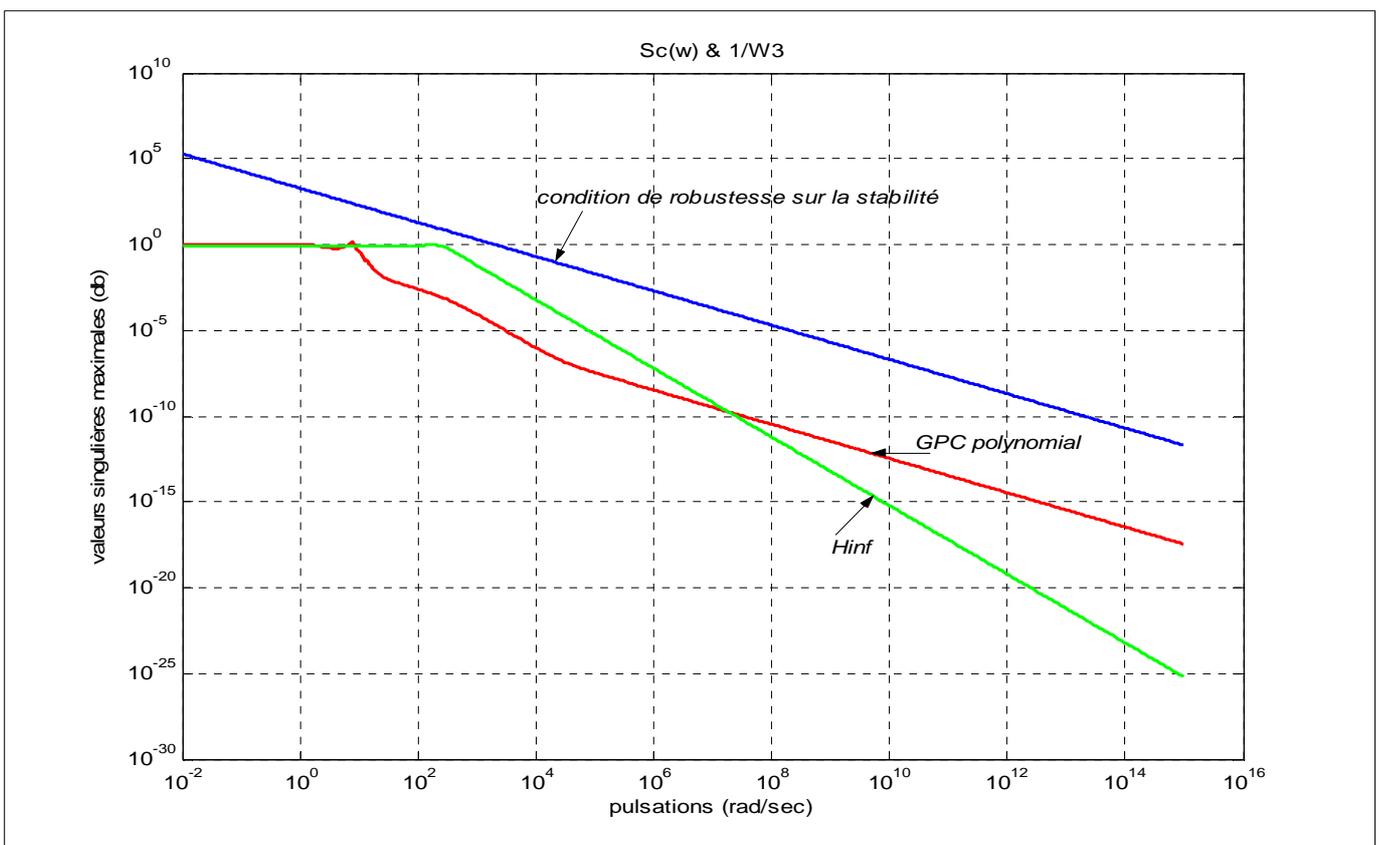


Fig.2.8.2 : Robustesse de stabilité (comparaison)

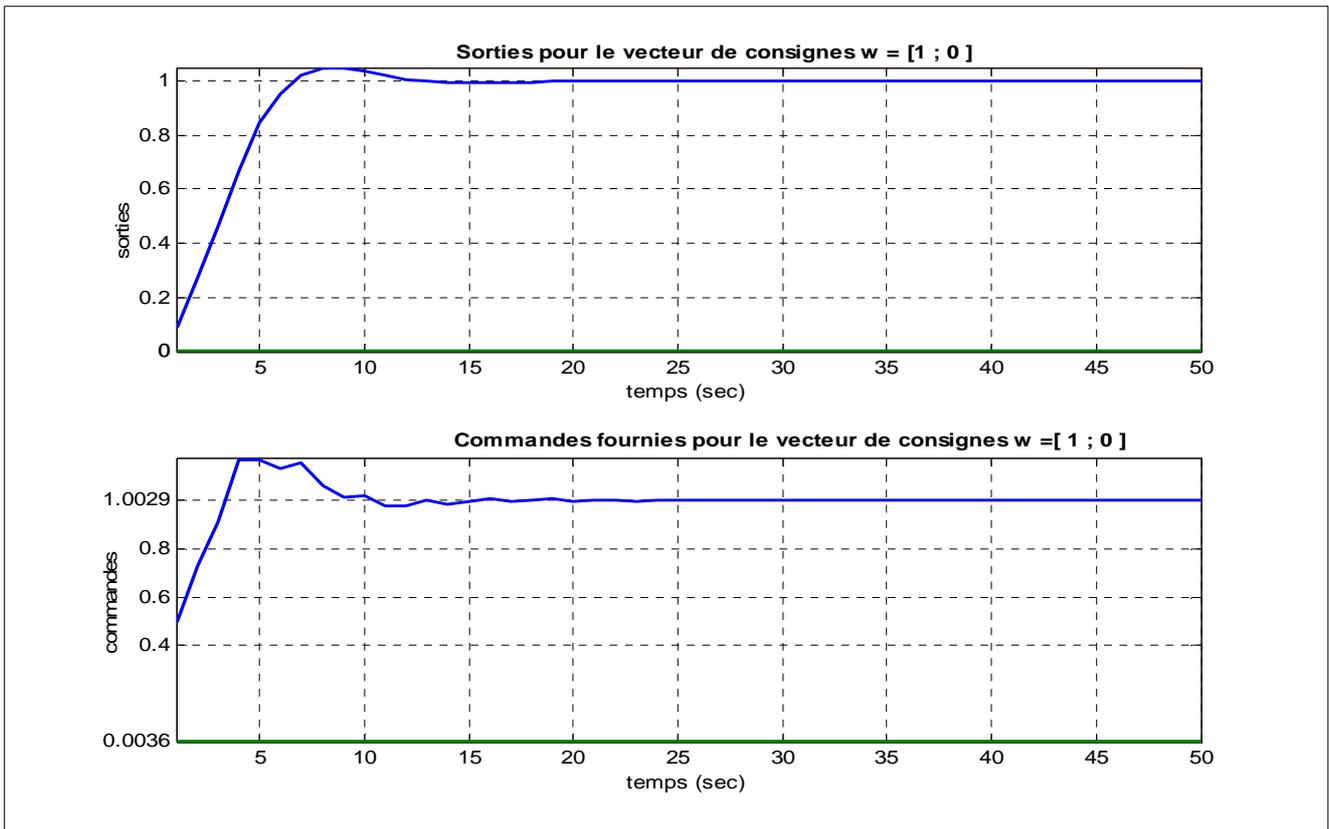


Fig.2.8.3 : Sorties/Commandes obtenues par RST\_GPC pour  $w=(1 \ 0)^T$

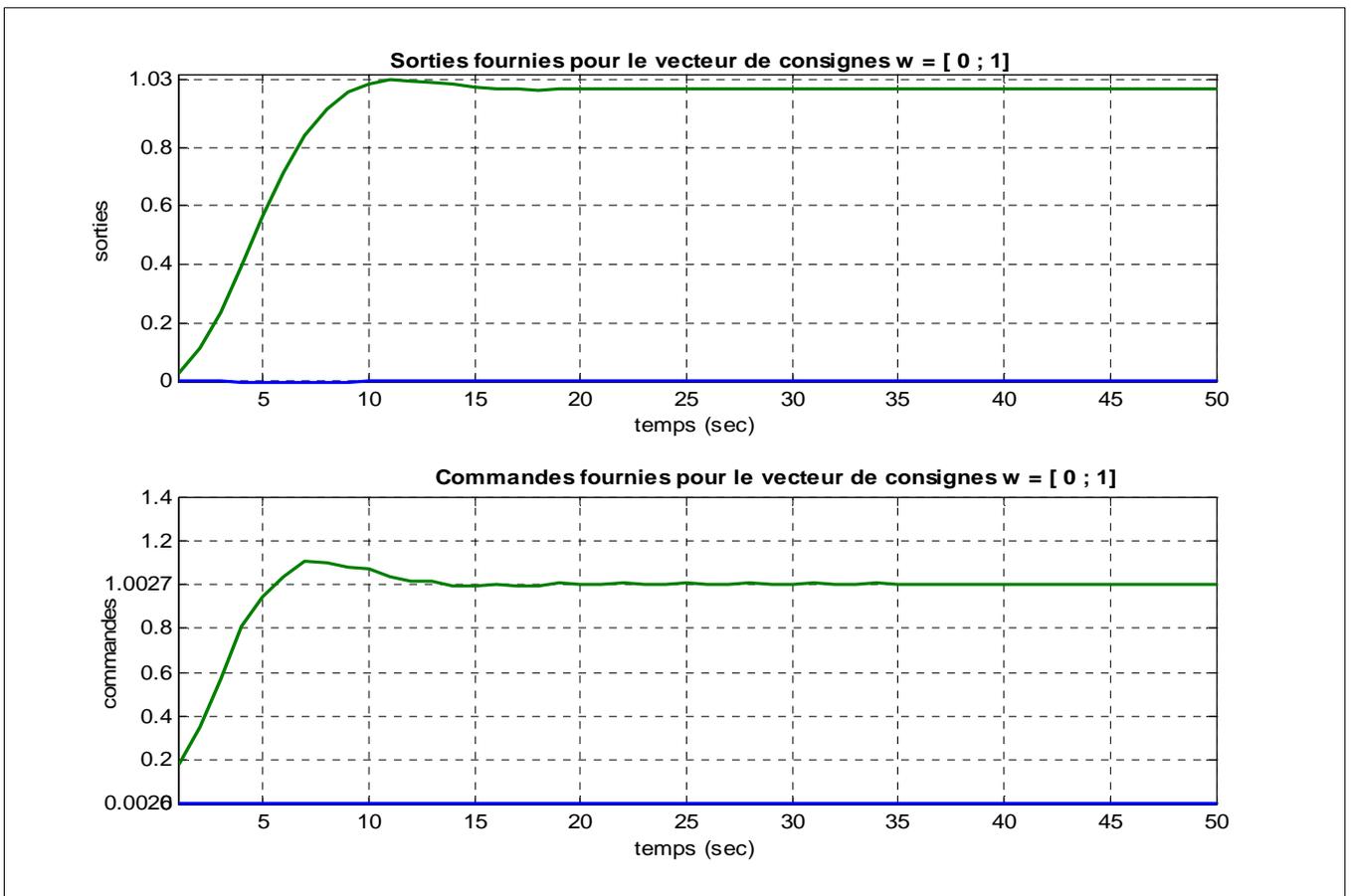


Fig.2.8.4 : Sorties/Commandes obtenues par RST\_GPC pour  $w=(0 \ 1)^T$

Dans le plan fréquentiel, d'après les figures Fig.2.8.1, et Fig. 2.8.2, on peut constater que les lieux des valeurs singulières maximales de la fonction de sensibilité se situent au dessous du lieux des valeurs singulières maximales de l'inverse de la spécification  $w_s$ , ainsi que les lieux des valeurs singulières maximales de la fonction de sensibilité complémentaires se situent au dessous de  $w_3^{-1}$  celles-ci sont traduits par la satisfaction de la robustesse de stabilité et de performances par la méthode  $H_\infty$ . Ces deux conditions de robustesses sont vérifiées de même par le GPC polynomial avec un degré plus pour la robustesse de performances en basses fréquences et par un degré moins pour la robustesse de stabilité en hautes fréquences.

Dans le plan temporel, d'après les figures Fig.2.8.3 et Fig.2.8.4 on peut remarquer, qu'avec un bon choix de  $(N_2, N_u, \Lambda)$ , on peut voir un bon comportement de poursuite/ interaction du système bouclé ceci traduit par une erreur statique presque nulle pour chaque voie. Les commandes fournies par le contrôleur RST\_GPC sont stationnaires et tendent vers des valeurs constantes.

### 2.4.3. Application3: « GPC polynomial utilisant le préfiltre $C(q^{-1})$ »

Dans le paragraphe ci-dessus, nous avons montré qu'avec la transformation polynomiale de la loi de GPC on peut voir le comportement du système bouclé dans le plan fréquentiel ceci par le tracé des valeurs singulières maximales de la fonction de sensibilité directe et de sensibilité complémentaire. Nous avons montré également que pour un choix des paramètres de synthèse de GPC, le contrôleur RST nécessite d'une robustification face aux erreurs de modélisation. L'un des moyens utilisé pour améliorer la robustesse de GPC polynomial, celle proposée, pour le cas monovarié, par [Podr03], l'idée de cette approche est de réaliser un préfiltre  $C(q^{-1}) \neq 1$ . Nous ce qui va suivre nous allons essayer d'étendre cette approche sur un cas multivariable dans l'objectif est d'améliorer les réponses du contrôleur RST.

Le processus considéré ici représente un système multivariable perturbé sa dynamique est décrite par le modèle incertain de *CARIMA* où les incertitudes de ce modèle sont obtenues sous forme des bruits de mesures en haute fréquence. Notre objectif est de contrôler ce processus par un RST en utilisant la matrice polynomiale  $C(q^{-1})$ , les résultats trouvés sont comparés par ceux obtenus sans la pris en compte de cette matrice polynomiale.

- Modèle *CARIMA*:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} -0.88 & -0.16 & 0 \\ 0.32 & -0.96 & -0.8 \\ 0 & 0 & -0.72 \end{bmatrix} q^{-1} + \begin{bmatrix} 0.112 & 0.08 & 0 \\ 0 & 0.136 & 0.08 \\ 0.16 & 0 & 0.112 \end{bmatrix} q^{-2} + \begin{bmatrix} 0.064 & 0.016 & 0 \\ -0.08 & 0.072 & 0.24 \\ 0.16 & 0 & 0.0192 \end{bmatrix} q^{-3}$$

$$B = \begin{bmatrix} 0.5 & 0.2 & -0.5 \\ 2 & 0 & 0.3 \\ 0 & 0.9 & -0.4 \end{bmatrix} q^{-1} + \begin{bmatrix} 1 & 2 & 1 \\ -0.8 & 0.6 & 0.5 \\ 1 & 0.3 & 0.5 \end{bmatrix} q^{-2}.$$

- Les paramètres de synthèse :  $N_{u_{1,2}} = 4$   $N_{2_{1,2}} = 5$ ;  $\Lambda = \text{dig}(0.0002, 0.0001, 0.0005)$

Durant l'intervalle du temps  $0 \leq t \leq 140$ , on applique, pour chaque voie du système, une entrée de perturbation sous forme d'un échelon retardé d'amplitude 0.2 à partir de l'instant  $t=60$  secondes, un bruit de variance égale 0.03 à partir de l'instant  $t=100$  secondes, et trois références différentes définies par :

- Les références de chaque voie  $m$  :

$$\text{- Voie 1 : } W_1 = \begin{cases} 0.0: & 0 \leq t \leq 10 \\ 1.0: & 10 \pi t \leq 90 \\ 1.5: & 90 \pi t \leq 100 \\ 0.5: & 100 \pi t \leq 140 \end{cases}$$

$$\text{- Voie 2 : } W_2 = \begin{cases} 0.0: & 0 \leq t \leq 10 \\ -2.0: & 10 \pi t \leq 70 \\ 1.0: & 70 \pi t \leq 100 \\ 0.0: & 100 \pi t \leq 140 \end{cases}$$

$$\text{- Voie 3 : } W_3 = \begin{cases} 0.0: & 0 \leq t \leq 10 \\ 1.5: & 10 \pi t \leq 70 \\ -1.0: & 70 \pi t \leq 100 \\ 0.0: & 100 \pi t \leq 140 \end{cases}$$

Comparons alors les régulateurs obtenus avec  $C(q^{-1}) = I_{3 \times 3}$  et  $C(q^{-1}) = (1 - 0.85q^{-1}).I_{3 \times 3}$  on obtient les figures suivantes :

- Figure (2.9.1) : Les entrées de perturbations injectées au système à commander
- Figure (2.9.2) : Les entrées de bruits agissant sur les entrées du système.
- Figure (2.9.3) : Les sorties obtenues via le contrôleur RST avec ou sans filtre.
- Figure (2.9.4) : Les commandes fournies par le contrôleur RST.

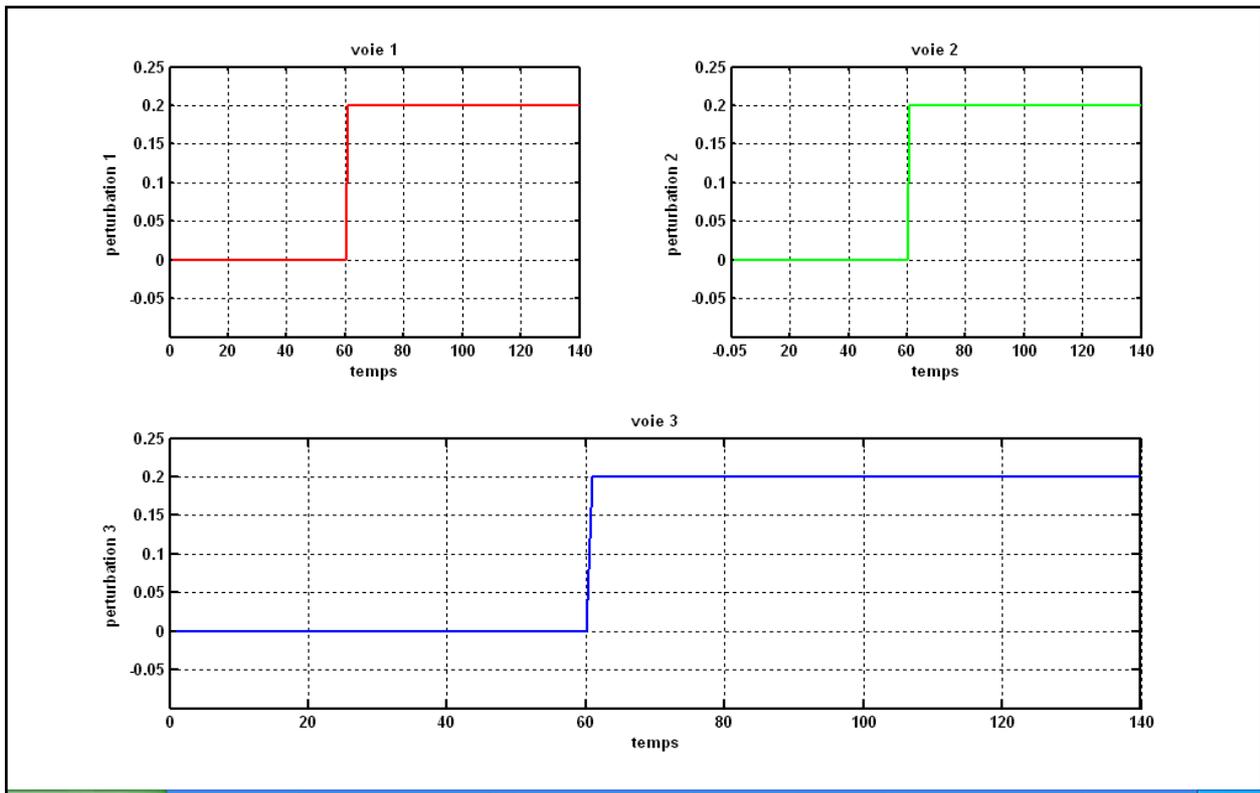


Fig.2.9.1: Les entrées de perturbations injectées au système à commander

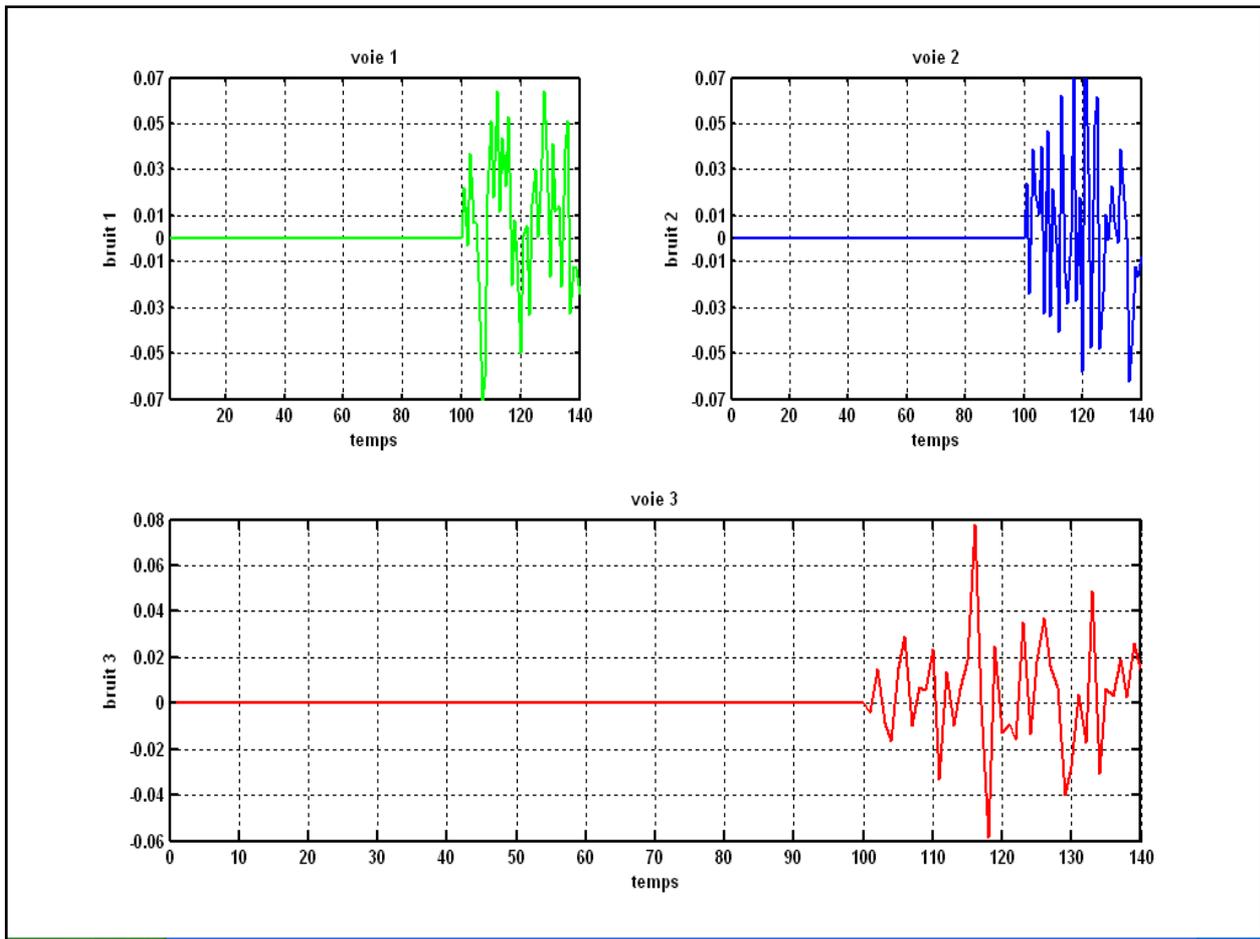


Fig.2.9.2: Les entrées de bruits agissant sur les entrées du système.

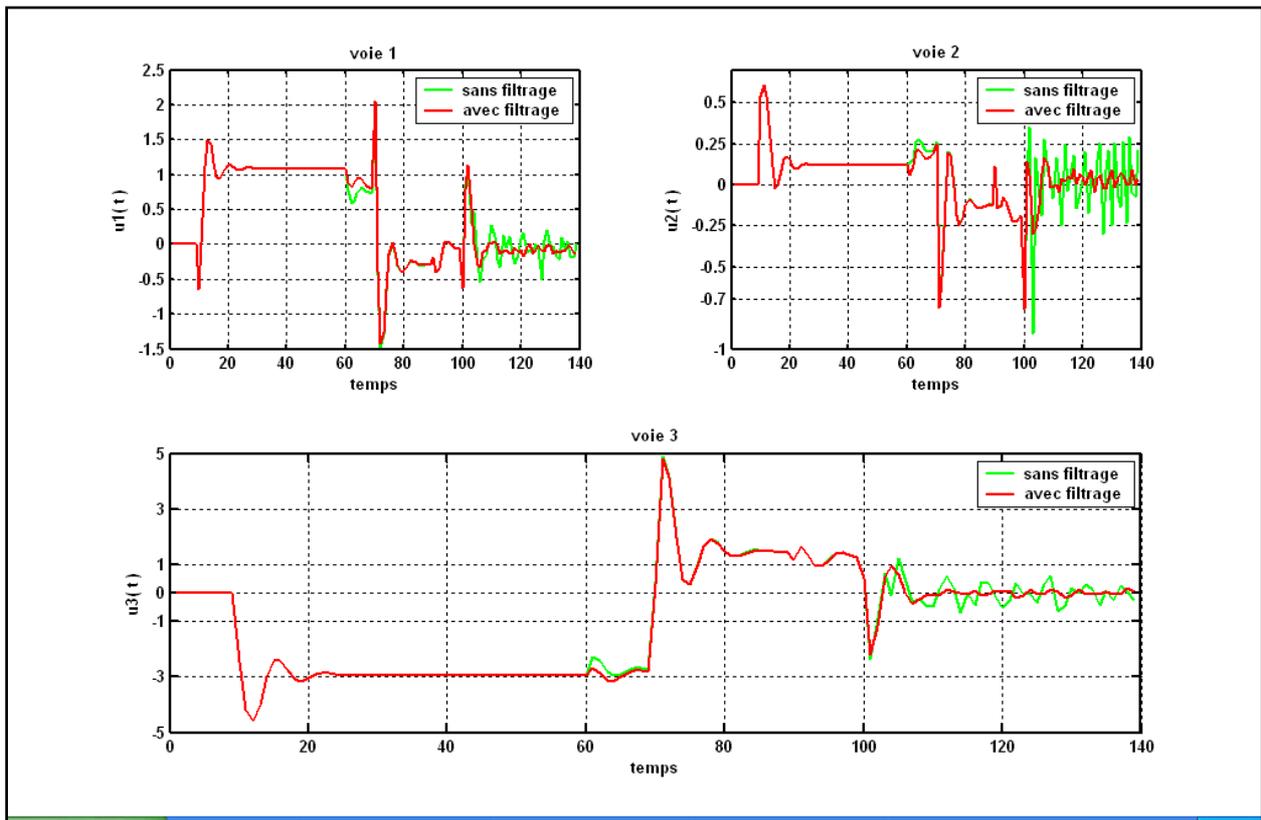


Fig.2.9.3: Les sorties obtenues via le contrôleur RST avec ou sans filtre.

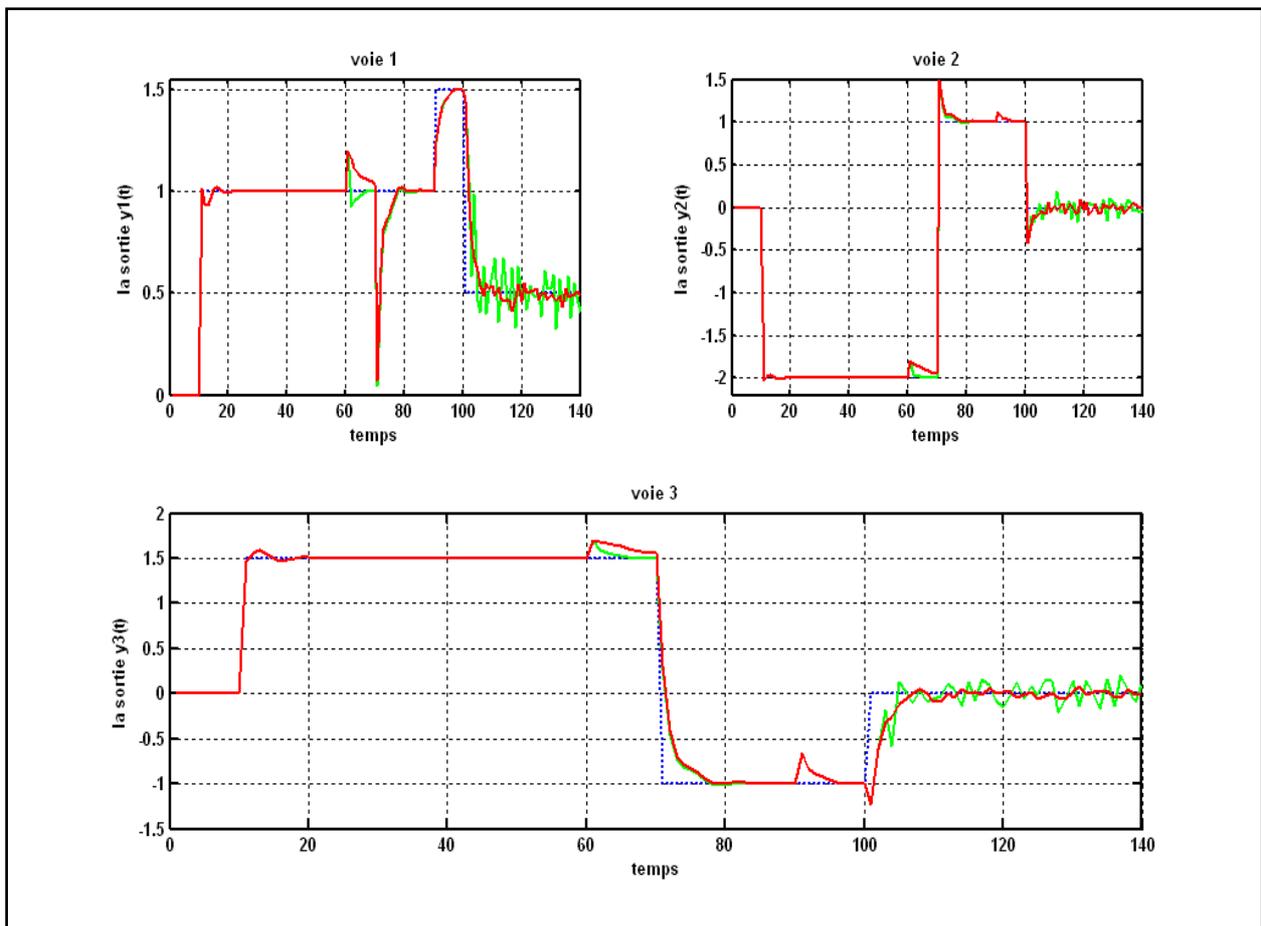


Fig.2.9.4: Les commandes fournies par le contrôleur RST.

Les figures (2.9.4) et (2.9.3) illustrent, pour les deux choix de  $C(q^{-1})$ , le rejet de perturbations en échelon d'amplitude 0.2, et l'effet de bruits sur les signaux de commandes et les sorties du système d'où on peut constater les observations suivantes :

- De l'instant initial jusqu'à  $t = 100$  secondes : un meilleur comportement de poursuite et de rejection de perturbations est obtenu pour les deux configurations de  $C(q^{-1})$  ce qui traduit par un bon choix de paramètres de synthèse  $(N_2, N_u, \Lambda)$ . De plus, la réponse à échelon de consigne est la même pour chaque voie, car le transfert en boucle fermée n'est pas influencé par  $C(q^{-1})$ .
- Dès qu'après l'instant  $t = 100$  secondes, l'effet de bruits dans les commandes et les sorties du système devient important avec  $C(q^{-1}) = I_m$ .

## 2.5. Conclusion:

Ce chapitre a permis de développer la forme polynomiale de la commande prédictive généralisée multivariable, cette forme est ensuite utilisée pour l'étude de robustesse du système bouclé dans le plan fréquentiel, celle-ci par le tracé de la fonction de sensibilité et de la sensibilité complémentaire.

On a pu constater, d'après d'une étude comparative avec la méthode  $H_\infty$  appliquée sur un exemple disponible dans la bibliothèque du *Matlab/Robuste*, que le contrôleur de GPC polynomial multivariable nécessite d'une robustification en hautes fréquences face aux incertitudes du modèle.

L'un des moyens employés pour robustifier le contrôleur de GPC polynomial celle proposée dans le cas monovariante par [Podr03]. Dans la dernière application présentée dans ce chapitre, on a utilisé cette approche proposée et la adapter au cas multivariable, la robustification a été réalisée par un choix de  $C(q^{-1})$  diagonal différent de l'unité, et la comparaison avec le cas de  $C(q^{-1})$  égale à l'unité montre que le système bouclé par le RST robustifié devient moins sensible aux incertitudes paramétriques du modèle, moins sensible également aux bruits de mesures et la dynamique de poursuite du système en boucle fermée devient inchangée.

### 3. Introduction:

Les contraintes les plus souvent définies dans le domaine temporel interviennent sur toutes les parties d'un système de commande, des actionneurs aux capteurs, et, en général, comme contraintes sur les états du système à commander. Leur présence est une source de problèmes parfois complexes, non seulement pour les praticiens de l'automatique mais aussi pour le milieu académique, car ils constituent le type de non linéarité le plus classique affectant, en pratique, une loi de commande, avec des implications directes sur la stabilité, les performances et la sûreté de fonctionnement du système en boucle fermée (parfois avec des conséquences tragiques).

La commande prédictive des systèmes linéaires invariants dans le temps, restreinte au cas sans contraintes, ne requiert pas la résolution effective d'un problème d'optimisation en ligne, car le correcteur est à son tour linéaire invariant et sa description analytique peut être obtenue hors-ligne.

Un avantage primordial issu de cette caractéristique est que toute la théorie de la commande linéaire (sans doute la plus complète de l'automatique) peut être utilisée pour l'analyse et la synthèse de la loi prédictive sans contraintes.

Une fois le correcteur élaboré, les pôles de la boucle ouverte, la stabilité où l'erreur stationnaire sont faciles à déterminer. Clarke a montré que le réglage des performances d'une boucle de commande prédictive s'effectue de façon claire par le choix des paramètres de réglage de la loi prédictive, horizons de prédiction ou pondérations. Dans ce cas, choisir une méthode de commande plutôt qu'une autre est souvent lié au contexte.

On peut donc conclure à ce stade, en l'absence de contraintes, qu'une méthodologie très complète de commande prédictive linéaire existe, s'appliquant aux systèmes à déphasage non minimal, avec retards, instables, multi variables, comme le soulignent les études effectuées par [KBM96]. Le problème change cependant complètement lorsque des contraintes doivent être prises en compte.

Des avancées au niveau de la mise en forme des problèmes de commande prédictive ont été obtenues vers la fin des années 90 [KBM96], permettant l'application de la fonction « *quadprog* » (Quadratic Programming), devenue très populaire et qui offre même aujourd'hui des solutions intéressantes avec un plus concernant la facilité de traitement du cas robuste.

### 3.1. Les contraintes :

Le critère précédent (1.19) avec l'équation (1.17) peut être réécrit comme suit :

$$J = (G\tilde{U} + Fc - W)^T (G\tilde{U} + Fc - W) + \tilde{U}^T \Lambda \tilde{U} \quad (3.1)$$

Une première conclusion pour la formulation GPC dérivé de l'équation (3.1) est que la fonction de coût GPC possède une forme quadratique [Cama93]:

$$J = \tilde{U}^T Q_2 \tilde{U} + Q_1^T \tilde{U} + Q_0 \quad (3.2)$$

$$\text{Avec : } \begin{cases} Q_2 = G^T G + \Lambda \\ Q_1^T = 2(Fc - W)^T G \\ Q_0 = (Fc - W)^T (Fc - W) = \text{const} \end{cases}$$

Le critère (3.2) est quadratique en  $\tilde{U}$ , mais en réalité les solutions possibles de cette optimisation sont restreintes par l'existence des contraintes provenant de limitations portant sur l'amplitude de la commande, la vitesse de réaction de l'actionneur, le signal de sortie ou autres signaux critiques pour le fonctionnement du système, mais aussi sur des contraintes terminales permettant de renforcer la stabilité.

Pratiquement, les différentes contraintes qui existent peuvent être classées selon le schéma :

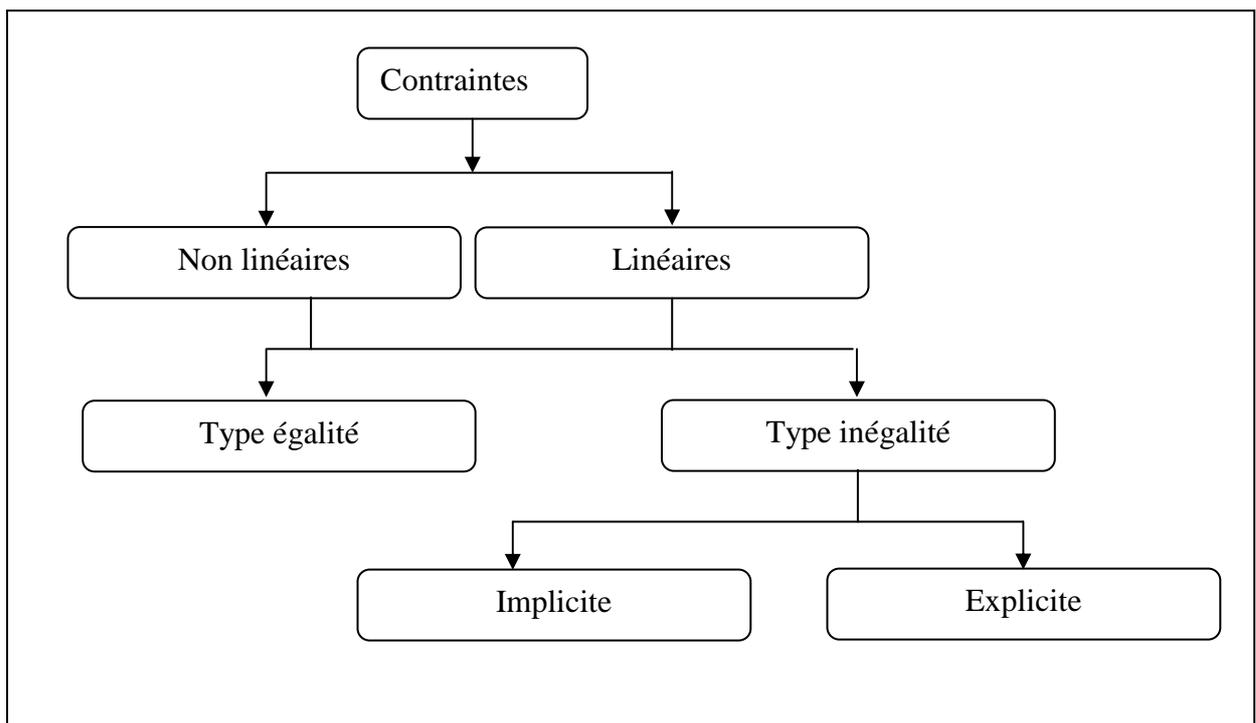


Figure 3.1. Classification des contraintes

### 3.2. Formulation des contraintes:

Les contraintes qui interviennent dans la formulation GPC peuvent être décrites par l'intermédiaire d'un formalisme unifié. Sachant que les incréments des commandes futures sont les variables qui subissent finalement ces contraintes ; il est intéressant d'exprimer toutes les limitations en se basant sur ce formalisme [cama93]:

$$\begin{aligned} v_{\min} \leq v(t+j) \leq v_{\max} \\ N_{s1} \leq j \leq N_{s2} \end{aligned} \quad (3.3)$$

Avec  $v(t+j)$  qui représente le signal contraint,  $(v_{\min}, v_{\max})$  les limitations à respecter,  $(N_{s1}, N_{s2})$  l'horizon de contraintes et pouvant être différents des horizons de prédiction  $(N_2, N_u)$ .

Afin d'illustrer le mécanisme de construction des contraintes pour un problème GPC, les paragraphes suivants proposent quelques formalismes de contraintes très souvent rencontrés en pratique.

#### 3.2.1. Contraintes sur les incréments de commandes:

On s'intéresse ici aux contraintes de la forme [CS91]:

$$\begin{aligned} \Delta u_{\min_i} \leq \Delta u_i(t+j) \leq \Delta u_{\max_i} \\ N_{s1} \leq j \leq N_{s2}, \quad i = 1, 2, \dots, m \end{aligned} \quad (3.4)$$

L'expression matricielle pour ce type de contraintes, en particulierisant les horizons  $N_{s1} = 0$ ,  $N_{s2} = N_u - 1$  est :

$$\begin{aligned} D_{\Delta U} \tilde{U} \leq \Delta \tilde{U}_{\max} \\ -D_{\Delta U} \tilde{U} \leq -\Delta \tilde{U}_{\min} \end{aligned} \quad (3.5)$$

Avec :

$$D_{\Delta U} = \begin{pmatrix} I_{mxm} & 0 & 0 & \dots & 0 \\ 0 & I_{mxm} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & I_{mxm} \end{pmatrix}_{m.N_u \times m.N_u}, \text{ et}$$

$$\Delta \tilde{U}_{\max} = \left( \begin{bmatrix} \Delta u_{\max 1} \\ \mathbf{M} \\ \Delta u_{\max m} \end{bmatrix} \mathbf{K} \begin{bmatrix} \Delta u_{\max 1} \\ \mathbf{M} \\ \Delta u_{\max m} \end{bmatrix} \right)^T, \quad \Delta \tilde{U}_{\min} = \left( \begin{bmatrix} \Delta u_{\min 1} \\ \mathbf{M} \\ \Delta u_{\min m} \end{bmatrix} \mathbf{K} \begin{bmatrix} \Delta u_{\min 1} \\ \mathbf{M} \\ \Delta u_{\min m} \end{bmatrix} \right)^T$$

### 3.2.2. Contraintes sur les amplitudes de commandes:

On s'intéresse ici aux contraintes de la forme :

$$\begin{aligned} u_{\min i} &\leq u_i(t+j) \leq u_{\max i} \\ N_{s1} &\leq j \leq N_{s2}, \quad i = 1, 2, \dots, m \\ i &= 1, 2, \dots, m \end{aligned} \quad (3.6)$$

Avec la relation :

$$u_i(t+j) = u_i(t-1) + \sum_{k=0}^j \Delta u_i(t+k) \quad (3.7)$$

On obtient l'expression matricielle :

$$\begin{aligned} D_U \tilde{U} &\leq \tilde{U}_{\max} \\ -D_U \tilde{U} &\leq -\tilde{U}_{\min} \end{aligned} \quad (3.8)$$

Avec :

$$D_U = \begin{bmatrix} I_{m \times m} & 0 & 0 & \dots & 0 \\ I_{m \times m} & I_{m \times m} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \vdots & \dots & \vdots \\ I_{m \times m} & I_{m \times m} & I_{m \times m} & \dots & I_{m \times m} \end{bmatrix}_{m.N_u \times m.N_u}, \text{ et}$$

$$\tilde{U}_{\max} = \left( \begin{bmatrix} u_{\max 1} - u_1(t-1) \\ \mathbf{M} \\ u_{\max m} - u_m(t-1) \end{bmatrix} \mathbf{K} \begin{bmatrix} u_{\max 1} - u_1(t-1) \\ \mathbf{M} \\ u_{\max m} - u_m(t-1) \end{bmatrix} \right)^T, \quad \tilde{U}_{\min} = \left( \begin{bmatrix} u_{\min 1} - u_1(t-1) \\ \mathbf{M} \\ u_{\min m} - u_m(t-1) \end{bmatrix} \mathbf{K} \begin{bmatrix} u_{\min 1} - u_1(t-1) \\ \mathbf{M} \\ u_{\min m} - u_m(t-1) \end{bmatrix} \right)^T$$

### 3.2.3. Contraintes sur les signaux de sorties:

On s'intéresse ici aux contraintes de la forme [CS91] :

$$\begin{aligned}
y_{\max_i} &\leq y_i(t+j) \leq y_{\min_i} \\
N_{s1} &\leq j \leq N_{s2}, \quad i = 1, 2, \dots, m
\end{aligned} \tag{3.9}$$

Utilisant la relation (1.17), d'où l'expression matricielle de (3.9) qui devient :

$$\begin{aligned}
G\tilde{U} &\leq \tilde{Y}_{\max} - Fc \\
-G\tilde{U} &\leq -\tilde{Y}_{\min} + Fc
\end{aligned} \tag{3.10}$$

Avec :

$$G = \begin{bmatrix} G_{N_1-1} & \Lambda & G_0 & \Lambda & \Lambda & 0 \\ G_{N_1} & G_{N_1-1} & \Lambda & G_0 & \Lambda & 0 \\ \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{M} \\ G_{N_u-1} & G_{N_u-2} & G_{N_u-3} & \Lambda & \Lambda & G_0 \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} \\ G_{N_2-1} & G_{N_2-2} & G_{N_2-3} & \Lambda & G_{N_2-N_u+1} & G_{N_2-N_u} \end{bmatrix}, \text{ et}$$

$$\tilde{Y}_{\max} - Fc = \begin{bmatrix} y_{1\max} - fc_{1,N_1} \\ y_{2\max} - fc_{2,N_1} \\ \mathbf{M} \\ y_{m\max} - fc_{m,N_1} \end{bmatrix} \mathbf{K} \begin{bmatrix} y_{1\max} - fc_{1,N_2} \\ y_{2\max} - fc_{2,N_2} \\ \mathbf{M} \\ y_{m\max} - fc_{m,N_2} \end{bmatrix}^T, \quad \tilde{Y}_{\min} - Fc = \begin{bmatrix} y_{1\min} - fc_{1,N_1} \\ y_{2\min} - fc_{2,N_1} \\ \mathbf{M} \\ y_{m\min} - fc_{m,N_1} \end{bmatrix} \mathbf{K} \begin{bmatrix} y_{1\min} - fc_{1,N_2} \\ y_{2\min} - fc_{2,N_2} \\ \mathbf{M} \\ y_{m\min} - fc_{m,N_2} \end{bmatrix}^T$$

### 3.2.4. Contraintes terminales de type égalité:

Les contraintes terminales de type égalité peuvent être exigées pour forcer la sortie prédite de suivre la référence prédite durant un certain nombre d'échantillonnages  $N_s$  après l'horizon  $N_2$ . Ces contraintes peuvent prendre la structure [CS91] :

$$\begin{aligned}
y_i(t+j) &= w_i(t+j) \\
j &= N_2 + 1, \dots, N_s, \quad i = 1, 2, \dots, m
\end{aligned} \tag{3.11}$$

Il s'agit encore une fois de contraintes sur la sortie. Le formalisme (3.11) peut être exprimé comme un ensemble de contraintes d'égalité sur les incréments de commandes future, en utilisant :

$$\tilde{Y}_{N_s} = \begin{bmatrix} y_1(t+N_2+1) \\ y_2(t+N_2+1) \\ \mathbf{M} \\ y_m(t+N_2+1) \end{bmatrix} \mathbf{K} \begin{bmatrix} y_1(t+N_2+N_s) \\ y_2(t+N_2+N_s) \\ \mathbf{M} \\ y_m(t+N_2+N_s) \end{bmatrix}^T \tag{3.12}$$

Avec la relation (1.17), le formalisme (3.11) mis sous forme matricielle devient :

$$G_{N_s} \tilde{U} = \tilde{Y}_{N_s} - Fc_{N_s} \quad (3.13)$$

### 3.2.5. Autres types de contraintes:

En plus des contraintes décrites précédemment, d'autres contraintes peuvent être rencontrées lors des applications, comme par exemple imposer un comportement monotone pour éviter les oscillations, ou encore imposer des contraintes agissant sur le comportement des systèmes à déphasage non minimal. En conclusion de ces développements sur les formulations des contraintes, il faut souligner que le but est d'exprimer les limitations en termes de degrés de liberté se rapportant à l'expression sur laquelle elles agissent, soit ici  $\tilde{U}$ . Toutes les limitations ont été représentées par des contraintes linéaires, comme on peut le constater à partir des relations (3.5), (3.8), (3.10) et (3.13). Parmi les différentes formes que ces contraintes peuvent prendre, il est toujours possible d'identifier la présence du vecteur correspondant à l'argument du problème d'optimisation  $\tilde{U}$ . En particulierisant ce vecteur qui demeure l'objectif du processus de décision, l'ensemble de toutes les contraintes peut s'écrire sous forme compacte :

$$\text{Contraintes GPC : } \begin{cases} L_{in} \tilde{U} \leq M_{in} \\ L_{\acute{e}q} \tilde{U} = M_{\acute{e}q} \end{cases} \quad (3.14)$$

$$\text{Où : } \begin{cases} L_{in} = [D_{\Delta U} & -D_{\Delta U} & D_U & -D_U & G & -G]^T \\ M_{in} = [\Delta \tilde{U}_{\max} & -\Delta \tilde{U}_{\min} & \tilde{U}_{\max} & -\tilde{U}_{\min} & \tilde{Y}_{\max} - Fc & -\tilde{Y}_{\min} + Fc]^T \\ L_{\acute{e}q} = G_{N_s} \\ M_{\acute{e}q} = \tilde{Y}_{N_s} - Fc_{N_s} \end{cases}$$

Finalement, en regroupant la formulation du critère de performance (3.2) avec l'ensemble des contraintes à satisfaire (3.14), on obtient le problème de la commande de GPC qui doit le résoudre à chaque pas d'échantillonnage:

$$\text{Problème : } \begin{cases} \underset{\tilde{U}}{\text{Min}} \left( \frac{1}{2} \tilde{U}^T Q_2 \tilde{U} + \tilde{U}^T Q_1 + Q_0 \right) \\ \text{subject à : } \begin{cases} L_{in} \tilde{U} \leq M_{in} \\ L_{\acute{e}q} \tilde{U} = M_{\acute{e}q} \\ L_B \leq \tilde{U} \leq L_U \end{cases} \end{cases} \quad (3.15)$$

Mentionnons avant tout que la complexité des problèmes de programmation quadratique de ce type dépend largement de la nature de la matrice  $Q_2$  qui compose ici le Hessien. La commande prédictive bénéficie d'un avantage important, à savoir le fait que les problèmes d'optimisation associés sont caractérisés par un Hessien défini positif grâce à la construction de la fonction de coût (3.2) et en conséquence de quoi les conditions d'optimalité du deuxième ordre seront vérifiées.

Le problème (3.15) fait partie de la classe des problèmes de programmation non-linéaire, plus spécifiquement de la catégorie des problèmes de programmation quadratique. Pour résoudre ce type de problème, on fait appel à la fonction *Quadprog* (*quadratic programming*) disponible dans le Toolbox/Optimisation/ Matlab avec l'utilisation de la commande :

$$\tilde{U}_{opt} = \text{quadprog}(H, f, A, b, A_{eq}, b_{eq}, L_B, L_U) \quad (3.16)$$

$$\text{Où : } \begin{cases} H = Q_2, & f = Q_1 \\ A = L_{in}, & b = M_{in} \\ A_{eq} = L_{eq}, & b_{eq} = M_{eq} \\ L_B = [], & L_U = [] \end{cases}$$

$L_B$ ,  $L_U$  Représentent les limites supérieures et inférieures de l'ensemble des contraintes de bornes.

Dans (3.16), cet ensemble est considéré vide.

Une fois  $\tilde{U}_{opt}$  obtenue, on applique le principe de l'horizon glissant et seules les  $m$  lignes de  $\tilde{U}_{opt}$  sont appliquées au système à commander.

### 3.3. Principe de la fonction quadprog :

Mentionnons avant tous que, cette fonction représente la mise à jour de l'ancienne fonction *QP*.

Les deux méthodes de base utilisées dans cette fonction sont appelées: méthode de l'ensemble actif et méthode du point intérieur. Chacune de ces deux méthodes est activée dans cette fonction selon la nature des contraintes utilisées.

#### 3.3.1. Ensemble actif :

L'idée de cette méthode est présentée ci-dessous,

Si l'on connaît un point initial faisable  $\tilde{U}_0$  pour les contraintes :

$$\begin{cases} L_{in} \cdot \tilde{U}_0 \leq M_{in} \\ L_{\acute{e}q} \cdot \tilde{U}_0 = M_{\acute{e}q} \end{cases},$$

on peut alors identifier les inégalités saturées et ensuite une matrice  $L_{sat}$  et un vecteur  $B_{sat}$  en regroupant toutes les contraintes égalité mais aussi les composant de  $\bar{L}_{in}, \bar{M}_{in}$  qui correspondent aux inégalités saturées. Ce qui donne :

$$L_{sat} \cdot \tilde{U}_0 = M_{sat} \quad (3.17)$$

Le problème d'optimisation quadratique :

$$P' : \begin{cases} \underset{\tilde{U}}{Min} & f(\tilde{U}) = \frac{1}{2} \tilde{U}^T Q_2 \tilde{U} + \tilde{U}^T Q_1 + Q_0 \\ \text{subject à :} & L_{sat} \cdot \tilde{U} = M_{sat} \end{cases}, \quad (3.18)$$

peut être résolu en supposant que la solution soit  $\tilde{U}_1$ . Un test d'optimalité de Kuhn- Tucker donné par (3.19) doit être réalisé. Une telle opération revient à la vérification de la positivité des multiplicateurs de Lagrange pour (3.18),  $\mu_i \geq 0$ .

$$\text{Condition d'optimalité de Kuhn-Tucker : } \begin{cases} Q_2 \tilde{U} + L_{sat}^T \cdot \mu = -Q_1 \\ L_{sat} \cdot \tilde{U} = M_{sat} \\ \mu \geq 0 \end{cases} \quad (3.19)$$

Si ces conditions ne sont pas accomplies, une contrainte parmi celles associées à un coefficient de Lagrange négatif est éliminée de l'ensemble actif et la procédure est réitérée.

Si pour le cas où  $\tilde{U}_1$  se trouve à l'extérieur du domaine faisable décrit par l'ensemble complet des contraintes (3.14), elle sera remplacée par la combinaison linéaire:

$$\tilde{U}_{10} = \tau_1 \cdot \tilde{U}_0 + \tau_2 \cdot \tilde{U}_1, \quad 0 \leq \tau_1 \leq 1, \quad 0 \leq \tau_2 \leq 1, \quad \tau_1 + \tau_2 = 1, \quad (3.20)$$

qui se trouve sur la frontière. Le point  $\tilde{U}_{10}$  est faisable mais il lui correspond certaines contraintes saturées qui étaient précédemment inactives. Celles-ci sont ajoutées à l'ensemble actif et la procédure est réitérée.

### 3.3.2. Point intérieur:

Le principe de cette méthode part du problème initial (3.15) et de la formulation des conditions d'optimalité KKT :

$$\begin{cases} Q_2 \tilde{U} + Q_1 + L_{eq}^T \cdot \vartheta + L_{in}^T \cdot \mu = 0 \\ L_{eq} \cdot \tilde{U} = M_{eq} \\ L_{in} \cdot \tilde{U} + \psi = M_{in} \\ \mu \geq 0; \Psi \geq 0; \mu^T \cdot \psi = 0 \end{cases} \quad (3.21)$$

Avec  $\vartheta$ ,  $\mu$  les vecteurs des multiplicateurs de Lagrange et  $\psi$  le vecteur des variables de relaxation qui permet la transformation des inégalités en égalités. Ensuite ces équations seront transcrites sous forme matricielle :

$$\begin{bmatrix} Q_2 & L_{eq}^T & L_{in}^T \\ -L_{eq} & 0 & 0 \\ -L_{in} & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \tilde{U} \\ \vartheta \\ \mu \end{bmatrix} + \begin{bmatrix} Q_1 \\ M_{eq} \\ M_{in} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \psi \end{bmatrix} \quad (3.22)$$

$$\mu \geq 0; \psi \geq 0; \mu^T \cdot \psi = 0$$

Avec la matrice dans le membre de gauche défini positif, la méthode débouche au final sur la résolution d'un système d'équations non linéaires :

$$F(\tilde{U}, \vartheta, \mu, \psi) = \begin{bmatrix} Q_2 \tilde{U} + Q_1 + L_{eq}^T \cdot \vartheta + L_{in}^T \cdot \mu \\ -L_{eq} \cdot \tilde{U} + M_{eq} \\ -L_{in} \cdot \tilde{U} - \psi + M_{in} \\ \Omega \cdot \Gamma \cdot \Xi \end{bmatrix} = 0 \quad (3.23)$$

$$\mu \geq 0; \Psi \geq 0$$

$$\Omega = \text{diag}(\psi_1 \quad \psi_2 \quad \dots \quad \psi_r)$$

$$\text{Avec : } \Gamma = \text{diag}(\mu_1 \quad \mu_2 \quad \dots \quad \mu_r)$$

$$\Xi = (1 \quad 1 \quad \dots \quad 1)^T$$

L'indice  $r$  représente le nombre d'inégalités dans l'ensemble de contraintes.

La méthode de Newton est ensuite utilisée pour résoudre, pour chaque itération, la fonction

$$F(\tilde{U}, \vartheta, \mu, \psi)$$

### 3.4. Région de faisabilité:

Dans ce qui suit, on s'intéresse uniquement aux contraintes de type inégalité, ces dernières peuvent être divisées en deux classes : contraintes implicites et contraintes explicites.

Pour un problème d'optimisation, elles sont formulées comme suit :

$$\underset{\tilde{U}}{\text{Min}} \left( \frac{1}{2} \tilde{U}^T Q_2 \tilde{U} + \tilde{U}^T Q_1 + Q_0 \right)$$

$$\text{Sujet à des contraintes explicites : } \gamma_{\min} \leq \tilde{U} \leq \gamma_{\max} \quad (3.24)$$

$$\text{Sujet à des contraintes implicites : } g_i(\tilde{U}) \leq 0, \quad i = 1, 2, \dots, p$$

Les contraintes explicites sont aussi appelées contraintes de réglage et elles représentent les limitations dans les paramètres d'optimisation par des bornes supérieures et des bornes inférieures. Dans les contraintes implicites, les variables implicites  $g_i(\tilde{U})$  sont des fonctions de toutes les variables (paramètres) d'optimisation. Dans la théorie de l'optimisation sous contraintes, ces deux classes sont traitées plus ou moins de la même façon. En général, les contraintes implicites rendent l'optimisation plus difficile à traiter [CB99]. Notre objectif étant précisé, nous allons transformer les contraintes implicites en contraintes explicites pour faciliter le problème d'optimisation.

La méthode de recherche proposée par [CB99] permet de d'identifier les bornes maximales et autres bornes minimales afin de limiter toutes les contraintes implicites, les rendant par conséquent explicites.

Les bornes trouvées sont ensuite combinées de celles définissant l'ensemble des contraintes explicites et conduisant à un espace contenant uniquement les contraintes des bornes.

L'avantage prévu de cette approche est, d'une part, de réduire l'ensemble des contraintes inactives, ce qui peut faciliter la résolution du problème et d'autre part d'utiliser des méthodes d'optimisation basées sur les contraintes de bornes comme : la Dichotomie, la méthode utilisant la suite de Fibonacci... etc. Le problème d'optimisation devient alors :

$$P'' : \begin{cases} \underset{\tilde{U}}{\text{Min}} f(\tilde{U}) = \frac{1}{2} \tilde{U}^T Q_2 \tilde{U} + \tilde{U}^T Q_1 + Q_0 \\ \gamma'_{\min} \leq \tilde{U} \leq \gamma'_{\max} \end{cases} \quad (3.25)$$

La résolution du problème (3.22) par la même fonction *quadprog* devient moyennement facile par rapport à celui résolu par (3.16), cette fonction est utilisée avec la particularité :

$$\tilde{U}_{opt} = \text{quadprog}(H, f, A, b, A_{eq}, b_{eq}, L_B, U_B) \quad (3.26)$$

$$\text{Où : } \begin{cases} H = Q_2, & f = Q_1 \\ A = [], & b = [] \\ A_{eq} = [], & b_{eq} = [] \\ L_B = \gamma'_{\min}, & U_B = \gamma'_{\max} \end{cases}$$

Le principe de cette approche est illustré pour le cas particulier suivant :

$N_1 = 1, N_u = 3, N_2 = 4$ , et  $m = 2$  on a donc :

$$\tilde{U} = \left( \tilde{U}(t) \quad \tilde{U}(t+1) \quad \tilde{U}(t+2) \right)^T \quad (3.27)$$

Avec  $\tilde{U}(t+k) = \left( \Delta u_1(t+k) \quad \Delta u_2(t+k) \right)^T$

Considérons les contraintes suivantes :

$$\begin{cases} c_0 \leq D_{\Delta U} \cdot \tilde{U} \leq c_1 \\ c_2 \leq D_U \cdot \tilde{U} \leq c_3 \\ \bar{c}_4 \leq G \cdot \tilde{U} \leq \bar{c}_5 \end{cases} \quad (3.28)$$

Avec :

$$D_{\Delta U} = \begin{pmatrix} I_{2 \times 2} & 0_{2 \times 2} & 0_{2 \times 2} \\ 0_{2 \times 2} & I_{2 \times 2} & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & I_{2 \times 2} \end{pmatrix}, D_U = \begin{pmatrix} I_{2 \times 2} & 0_{2 \times 2} & 0_{2 \times 2} \\ I_{2 \times 2} & I_{2 \times 2} & 0_{2 \times 2} \\ I_{2 \times 2} & I_{2 \times 2} & I_{2 \times 2} \end{pmatrix}, \text{ et } G = \begin{pmatrix} G_0 & 0_{2 \times 2} & 0_{2 \times 2} \\ G_1 & G_0 & 0_{2 \times 2} \\ G_2 & G_1 & G_0 \\ G_3 & G_2 & G_1 \end{pmatrix}$$

Sachant que :  $G_k = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}_{k=1,2,\dots,N_2}$

Les vecteurs  $c_{i(i=1,2,\dots,5)}$  sont définis tels que :

$$c_0 = \left( \begin{bmatrix} \Delta u_{1\min} \\ \Delta u_{2\min} \end{bmatrix} \quad \begin{bmatrix} \Delta u_{1\min} \\ \Delta u_{2\min} \end{bmatrix} \quad \begin{bmatrix} \Delta u_{1\min} \\ \Delta u_{2\min} \end{bmatrix} \right)^T = (c_{01} \quad c_{02} \quad c_{03})^T \quad (3.29)$$

$$c_1 = \left( \begin{bmatrix} \Delta u_{1\max} \\ \Delta u_{2\max} \end{bmatrix} \begin{bmatrix} \Delta u_{1\max} \\ \Delta u_{2\max} \end{bmatrix} \begin{bmatrix} \Delta u_{1\max} \\ \Delta u_{2\max} \end{bmatrix} \right)^T = (c_{11} \quad c_{12} \quad c_{33})^T \quad (2.30)$$

$$c_2 = \left( \begin{bmatrix} u_{1\min} - u_1(t-1) \\ u_{2\min} - u_2(t-1) \end{bmatrix} \begin{bmatrix} u_{1\min} - u_1(t-1) \\ u_{2\min} - u_2(t-1) \end{bmatrix} \begin{bmatrix} u_{1\min} - u_1(t-1) \\ u_{2\min} - u_2(t-1) \end{bmatrix} \right)^T = (c_{21} \quad c_{22} \quad c_{23})^T \quad (2.31)$$

$$c_3 = \left( \begin{bmatrix} u_{1\max} - u_1(t-1) \\ u_{2\max} - u_2(t-1) \end{bmatrix} \begin{bmatrix} u_{1\max} - u_1(t-1) \\ u_{2\max} - u_2(t-1) \end{bmatrix} \begin{bmatrix} u_{1\max} - u_1(t-1) \\ u_{2\max} - u_2(t-1) \end{bmatrix} \right)^T = (c_{31} \quad c_{32} \quad c_{33})^T \quad (2.32)$$

$$\bar{c}_4 = \left( \begin{bmatrix} y_{1\min} - fc_{1,1} \\ y_{2\min} - fc_{2,1} \end{bmatrix} \begin{bmatrix} y_{1\min} - fc_{1,2} \\ y_{2\min} - fc_{2,2} \end{bmatrix} \begin{bmatrix} y_{1\min} - fc_{1,3} \\ y_{2\min} - fc_{2,3} \end{bmatrix} \begin{bmatrix} y_{1\min} - fc_{1,4} \\ y_{2\min} - fc_{2,4} \end{bmatrix} \right)^T = (\bar{c}_{41} \quad \bar{c}_{42} \quad \bar{c}_{43} \quad \bar{c}_{44})^T \quad (2.33)$$

$$\bar{c}_5 = \left( \begin{bmatrix} y_{1\max} - fc_{1,1} \\ y_{2\max} - fc_{2,1} \end{bmatrix} \begin{bmatrix} y_{1\max} - fc_{1,2} \\ y_{2\max} - fc_{2,2} \end{bmatrix} \begin{bmatrix} y_{1\max} - fc_{1,3} \\ y_{2\max} - fc_{2,3} \end{bmatrix} \begin{bmatrix} y_{1\max} - fc_{1,4} \\ y_{2\max} - fc_{2,4} \end{bmatrix} \right)^T = (\bar{c}_{51} \quad \bar{c}_{52} \quad \bar{c}_{53} \quad \bar{c}_{54})^T \quad (2.34)$$

Considérons les deux vecteurs  $c_4, c_5$  donnés par :

$$c_4 = (G_0^{-1} \cdot \bar{c}_{41} \quad G_0^{-1} \cdot \bar{c}_{42} \quad G_0^{-1} \cdot \bar{c}_{43} \quad G_0^{-1} \cdot \bar{c}_{44})^T = (c_{41} \quad c_{42} \quad c_{43} \quad c_{44})^T \quad (2.35)$$

$$c_5 = (G_0^{-1} \cdot \bar{c}_{51} \quad G_0^{-1} \cdot \bar{c}_{52} \quad G_0^{-1} \cdot \bar{c}_{53} \quad G_0^{-1} \cdot \bar{c}_{54})^T = (c_{51} \quad c_{52} \quad c_{53} \quad c_{54})^T \quad (2.36)$$

Notre objectif est de définir la région de faisabilité en se basant sur la réduction des contraintes précédentes de telles sortes que :

$$\text{Région de faisabilité : } \gamma'_{i\min} \leq \tilde{U}(t+i) \leq \gamma'_{i\max} \quad (3.37)$$

On a pour  $i = 0$  :

$$\gamma'_{0\min} \leq \tilde{U}(t) \leq \gamma'_{0\max}, \text{ avec : } \gamma'_{0\min} = \begin{bmatrix} \gamma'_{01\min} \\ \gamma'_{02\min} \end{bmatrix}, \gamma'_{0\max} = \begin{bmatrix} \gamma'_{01\max} \\ \gamma'_{02\max} \end{bmatrix}, \text{ et } \tilde{U}(t) = \begin{bmatrix} \Delta u_1(t) \\ \Delta u_2(t) \end{bmatrix}$$

Les deux de vecteurs de bornes  $\gamma'_{0\min}, \gamma'_{0\max}$  sont définies comme suit:

$$\begin{cases} \tilde{U}(t) \leq c_{11} \\ \tilde{U}(t) \leq c_{31} \\ \tilde{U}(t) \leq c_{51} \end{cases} \rightarrow$$

$$\gamma'_{0\max} = \min(c_{11}, c_{31}, c_{51}) \quad (3.38)$$

Remarquons qu'on peut éliminer à la fois deux vecteurs de contraintes sur  $\tilde{U}(t)$ .

$$\begin{cases} \tilde{U}(t) \geq c_{01} \\ \tilde{U}(t) \geq c_{21} \\ \tilde{U}(t) \geq c_{41} \end{cases} \rightarrow$$

$$\gamma'_{0\min} = \max(c_{01}, c_{21}, c_{41}) \quad (3.39)$$

On a, pour  $i = 1$ , les deux vecteurs de bornes  $\gamma'_{1\min}$ ,  $\gamma'_{1\max}$  définies comme suit:

$$\begin{cases} \tilde{U}(t+1) \leq c_{12} \\ \tilde{U}(t+1) \leq c_{32} - \tilde{U}(t) \\ \tilde{U}(t+1) \leq c_{52} - G_0^{-1} \cdot G_1 \cdot \tilde{U}(t) \\ \tilde{U}(t) \leq \gamma'_{0\max} \end{cases} \rightarrow$$

$$\gamma'_{1\max} = \min(c_{12}, c_{32} - \gamma'_{0\max}, c_{52} - G_0^{-1} \cdot G_1 \cdot \gamma'_{0\max}) \quad (3.40)$$

De plus

$$\begin{cases} \tilde{U}(t+1) \geq c_{02} \\ \tilde{U}(t+1) \geq c_{22} - \tilde{U}(t) \\ \tilde{U}(t+1) \geq c_{42} - G_0^{-1} \cdot G_1 \cdot \tilde{U}(t) \\ \tilde{U}(t) \geq \gamma'_{0\min} \end{cases} \rightarrow$$

$$\gamma'_{1\min} = \max(c_{02}, c_{22} - \gamma'_{0\min}, c_{42} - G_0^{-1} \cdot G_1 \cdot \gamma'_{0\min}) \quad (3.41)$$

On a, pour  $i = 2$ , les deux vecteurs de bornes  $\gamma'_{2\min}$ ,  $\gamma'_{2\max}$  définies comme suit:

$$\left\{ \begin{array}{l} \tilde{U}(t+2) \leq c_{13} \\ \tilde{U}(t+2) \leq c_{33} - \tilde{U}(t) - \tilde{U}(t+1) \\ \tilde{U}(t+2) \leq c_{53} - G_0^{-1}G_2\tilde{U}(t) - G_0^{-1}G_1\tilde{U}(t+1) \\ \tilde{U}(t+2) \leq G_1^{-1}\bar{c}_{54} - G_1^{-1}G_3\tilde{U}(t) - G_1^{-1}G_2\tilde{U}(t+1) \\ \tilde{U}(t) \leq \gamma'_{0\max} \\ \tilde{U}(t+1) \leq \gamma'_{1\max} \end{array} \right. \rightarrow$$

$$\gamma'_{2\max} = \min(c_{13}, c_{33} - \gamma'_{0\max} - \gamma'_{1\max}, c_{53} - G_0^{-1}[G_2\gamma'_{0\max} - G_1\gamma'_{1\max}], G_1^{-1}[\bar{c}_{54} - G_3\gamma'_{0\max} - G_2\gamma'_{1\max}]) \quad (3.42)$$

Ainsi que :

$$\left\{ \begin{array}{l} \tilde{U}(t+2) \geq c_{03} \\ \tilde{U}(t+2) \geq c_{23} - \tilde{U}(t) - \tilde{U}(t+1) \\ \tilde{U}(t+2) \geq c_{43} - G_0^{-1}G_2\tilde{U}(t) - G_0^{-1}G_1\tilde{U}(t+1) \\ \tilde{U}(t+2) \geq G_1^{-1}\bar{c}_{44} - G_1^{-1}G_3\tilde{U}(t) - G_1^{-1}G_2\tilde{U}(t+1) \\ \tilde{U}(t) \geq \gamma'_{0\min} \\ \tilde{U}(t+1) \geq \gamma'_{1\min} \end{array} \right. \rightarrow$$

$$\gamma'_{2\min} = \max(c_{03}, c_{23} - \gamma'_{0\min} - \gamma'_{1\min}, c_{43} - G_0^{-1}[G_2\gamma'_{0\min} - G_1\gamma'_{1\min}], G_1^{-1}[\bar{c}_{44} - G_3\gamma'_{0\min} - G_2\gamma'_{1\min}]) \quad (3.43)$$

Lorsque les contraintes sévères sont utilisées, l'infaisabilité provient du fait que les contraintes sont mutuellement incompatibles. Cette méthode proposée ne peut garantir la détermination de la région de faisabilité ; dans ce cas, la méthode d'optimisation ne peut calculer le vecteur de commande et la loi de GPC devient caduque.

### 3.5. Test et simulation:

#### 3.5.1. Application de la commande prédictive multivariable sous contraintes sur un système électromécanique:

La matrice de transfert ci-dessous représente le modèle mathématique simulant la dynamique d'un système électromécanique:

$$G(q^{-1}) = \begin{pmatrix} G_{11}(q^{-1}) & G_{12}(q^{-1}) \\ G_{21}(q^{-1}) & G_{22}(q^{-1}) \end{pmatrix}$$

$$\text{Avec: } G_{11}(q^{-1}) = \frac{6.04q^{-1} - 11.81q^{-2} + 6.03q^{-3}}{1 - 2.7923q^{-1} + 2.6695q^{-2} - 0.8702q^{-3}}, \quad G_{12}(q^{-1}) = \frac{0.01q^{-1} - 0.05q^{-2} - 0.0013q^{-3}}{1 - 2.7923q^{-1} + 2.6695q^{-2} - 0.8702q^{-3}}$$

$$G_{21}(q^{-1}) = \frac{0.0023q^{-1} + 0.0025q^{-2} + 0.0012q^{-3}}{1 - 2.7923q^{-1} + 2.6695q^{-2} - 0.8702q^{-3}}, \quad G_{22}(q^{-1}) = \frac{-0.12q^{-1} - 0.21q^{-2} - 0.104q^{-3}}{1 - 2.7923q^{-1} + 2.6695q^{-2} - 0.8702q^{-3}}$$

Notre système possède deux grandeurs d'entrée notés par:  $u_1$ ,  $u_2$ , et deux grandeurs de sorties qui sont  $y_1$  et  $y_2$ .

On désire suivre deux consignes représentées par deux signaux carrés périodiques dans une plage du temps comprise entre 0 et 400 secondes. La première consigne est d'amplitude égale à 1.50 destinée à la première voie de ce système, et la seconde est d'amplitude égale à 1.00 destinée à la deuxième voie.

Les paramètres de synthèses sont choisis tels que les illustrés dans le tableau suivant :

|        | $N_2$ | $N_u$ | $\lambda$ |
|--------|-------|-------|-----------|
| Voie 1 | 2     | 2     | 1.00      |
| Voie 2 | 5     | 2     | 0.01      |

En absence de contraintes, la commande prédictive multivariable offre une dynamique de poursuite stable, cependant, les commandes ainsi que les incréments de commandes présentent des limites maximales non souhaitables, avec:  $-0.54 \leq \Delta u_1 \leq 0.5$ ,  $-9.22 \leq \Delta u_2 \leq 9.25$ ,  $-1.45 \leq u_1 \leq 1.96$  et  $-8.08 \leq u_2 \leq 7.91$ . Notre objectif est de limiter ces limites, celles-ci par le choix des contraintes suivantes :

$$\Delta u(t) : \begin{cases} -0.18 \leq \Delta u_1(t) \leq 0.17 \\ -0.30 \leq \Delta u_2(t) \leq 0.30 \end{cases} \quad (3.36)$$

$$u(t) : \begin{cases} -1.30 \leq u_1(t) \leq 1.47 \\ -2.5 \leq u_2(t) \leq 2.5 \end{cases} \quad (3.37)$$

Nous appliquons, pour chaque instant du temps  $t$ , l'approche de réduction citée précédemment pour déterminer la région de faisabilité. Ensuite nous utilisons, pour la phase d'optimisation, la fonction *quadprog* présentée sous le formalisme (3.25), ainsi nous obtenons les réponses suivantes:

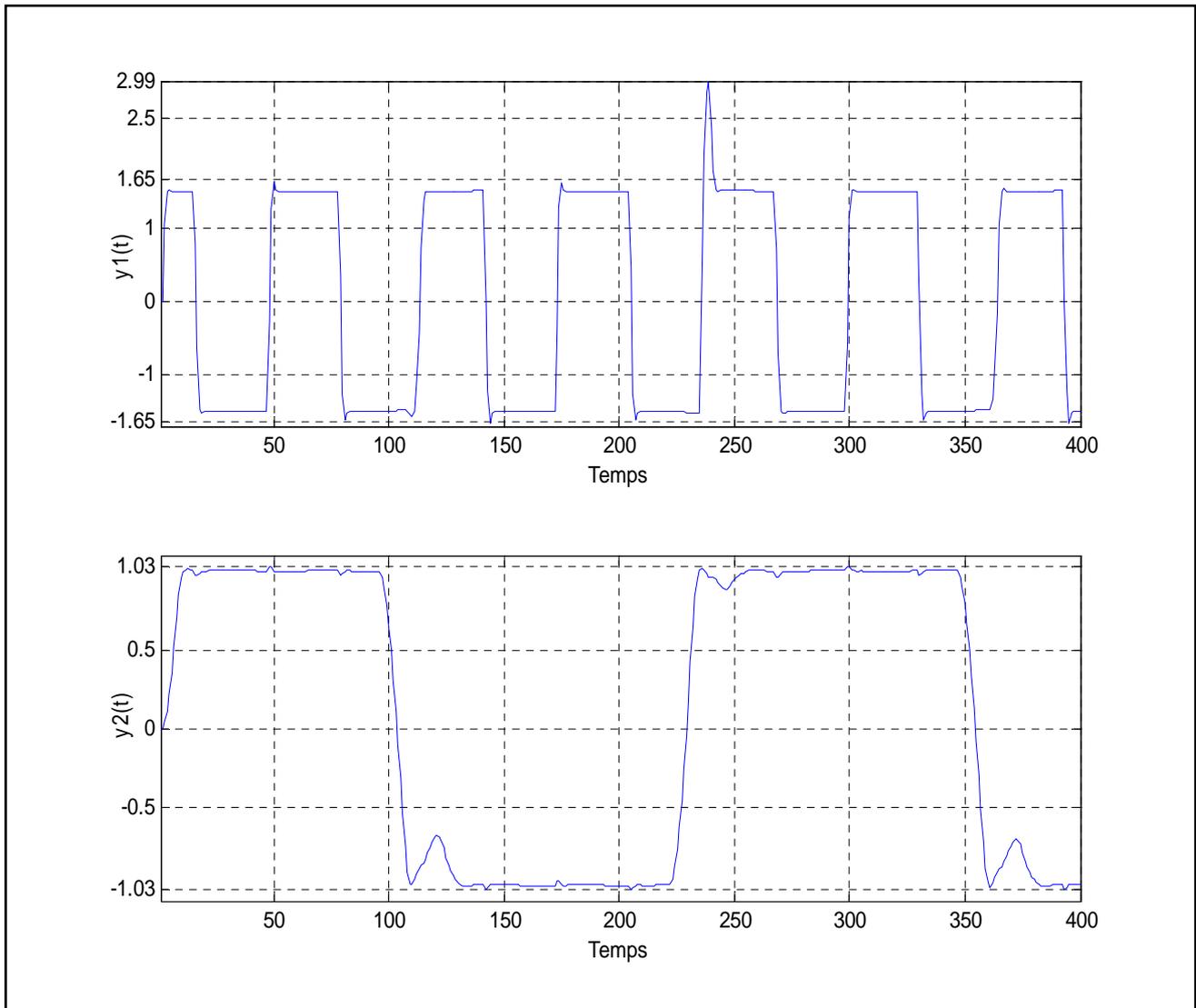


Fig.3.2.a. Sorties obtenues par le GPC multivariable utilisant la fonction *quadprog*

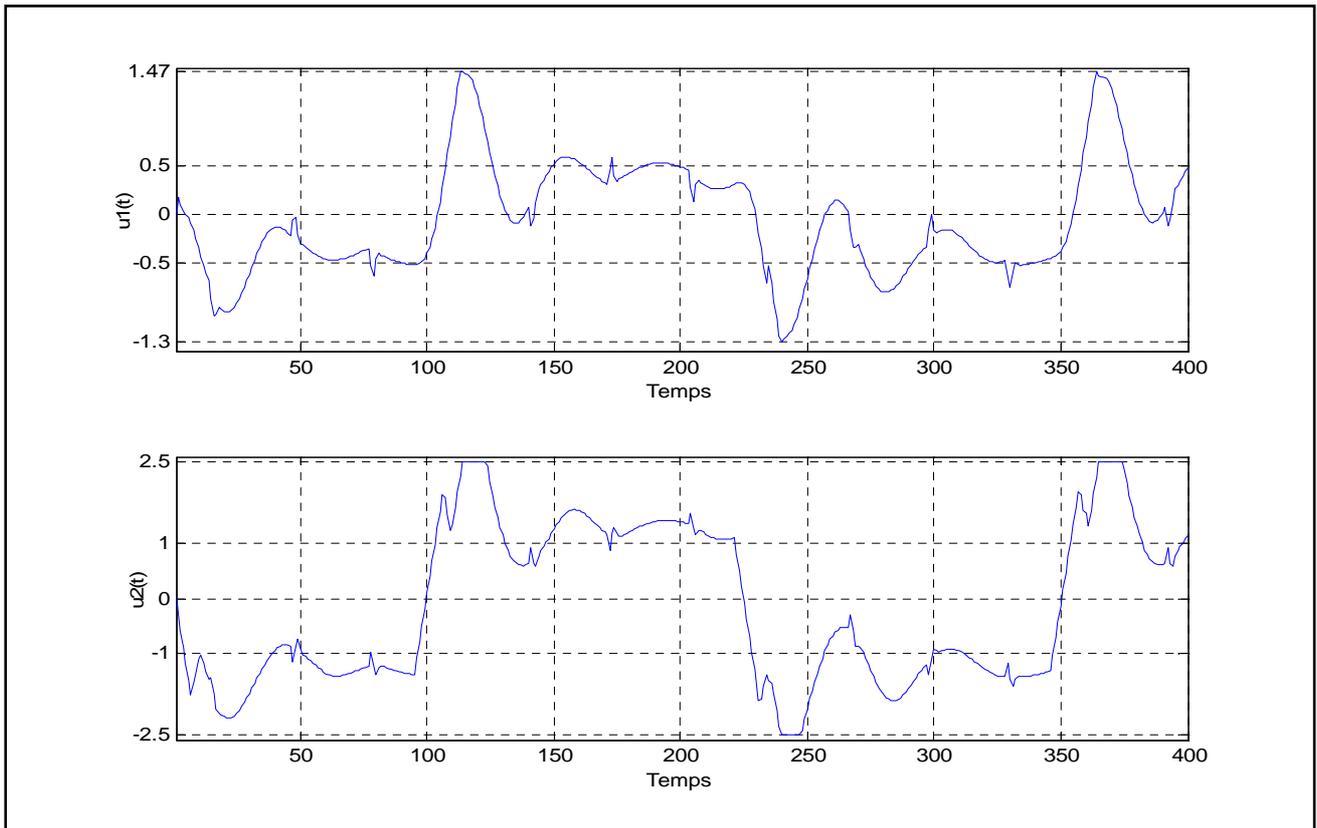


Fig.3.2.b. Commandes fournies par le GPC multivariable utilisant la fonction *quadprog*

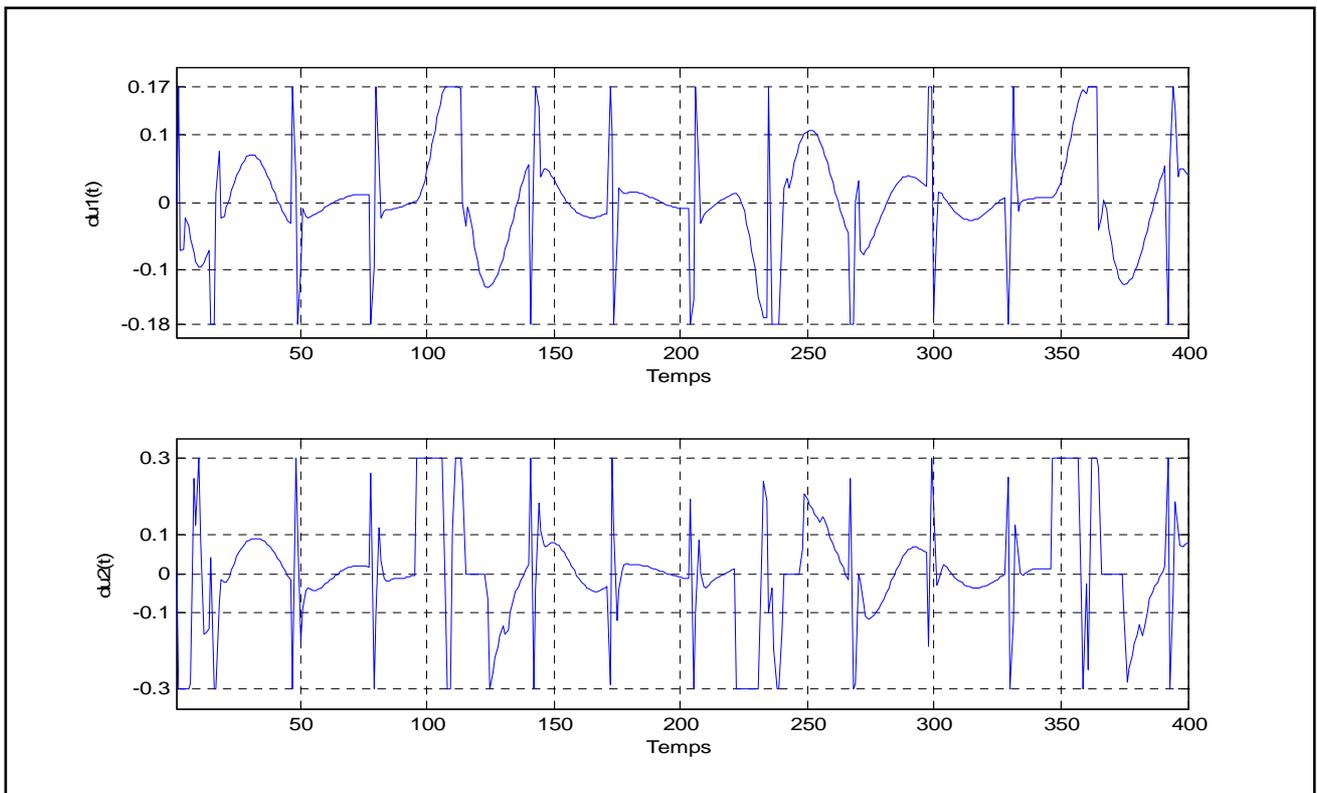


Fig.3.2.c. Incrément de commandes fournis par le GPC multivariable utilisant la fonction *quadprog*

D'après les figures 3.2, on peut conclure les résultats suivants :

- Une dynamique de poursuite stable obtenue dans les deux voies avec une satisfaction de toutes les contraintes considérées.
- Une présence d'un pic égal à 2.99 apparu à l'instant  $t = 239$  seconde dans la première voie, ce pic survient lorsque  $\Delta u_1 = \Delta \tilde{U}_{1\min}$  et  $u_1 = \tilde{U}_{1\min}$ . D'un point de vue pratique, si  $y_1(t)$  présente une vitesse, ce pic peut alors signifier une survitesse ou une vitesse critique ce qui se traduit par un phénomène mécanique non souhaité.

Nous essayons, dans ce qui suit, d'étendre l'intervalle de  $\Delta u_1$  tel que :  $-0.5 \leq \Delta u_1(t) \leq 0.5$  et de maintenir les limites citées précédemment sur les contraintes restantes. Les réponses obtenues sont comparées aux celles données par la figure 3.2.a, ainsi on obtient:

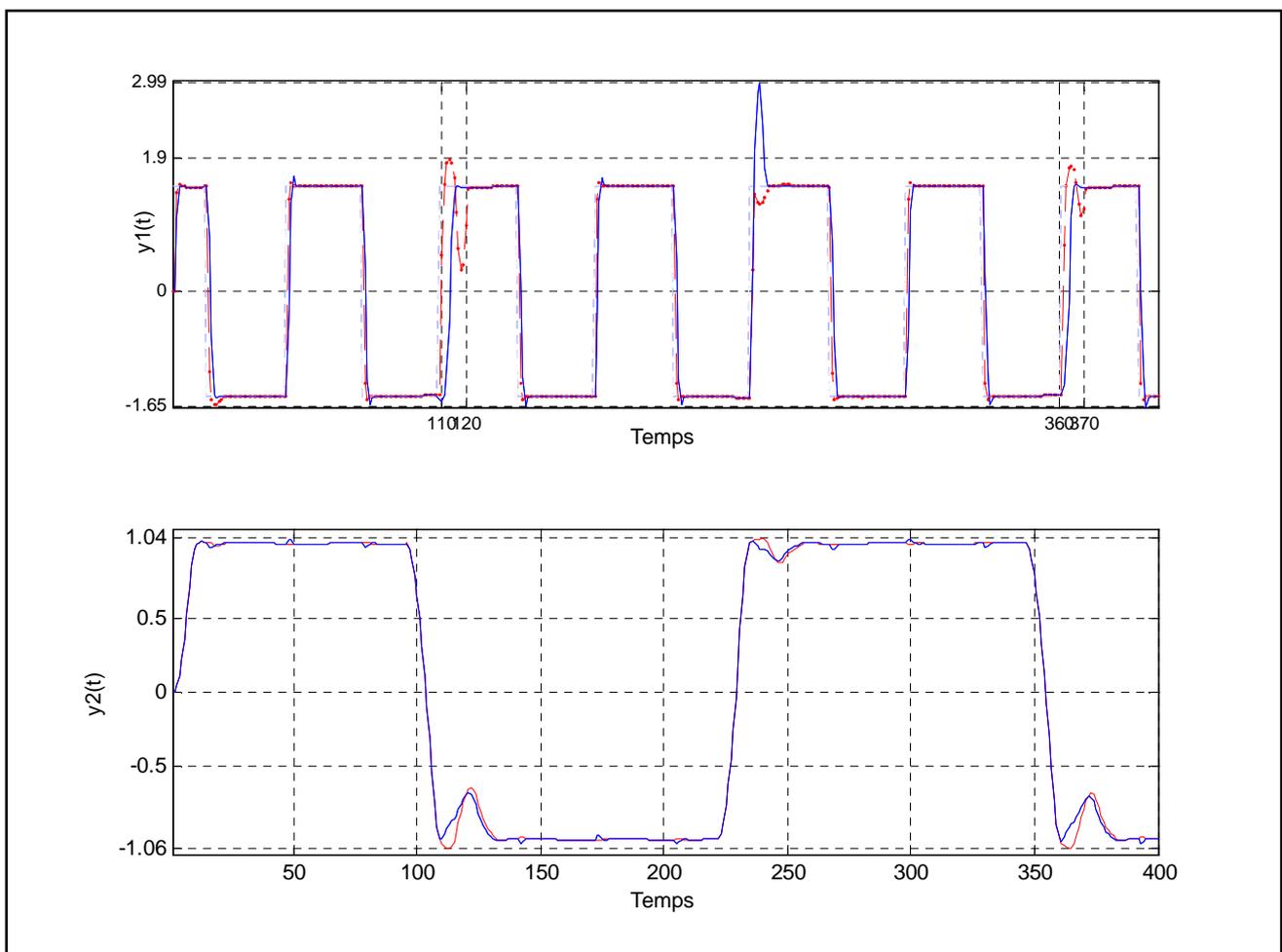


Fig.3.3.a. Sorties obtenues par le GPC multivariable utilisant la fonction *quadprog*

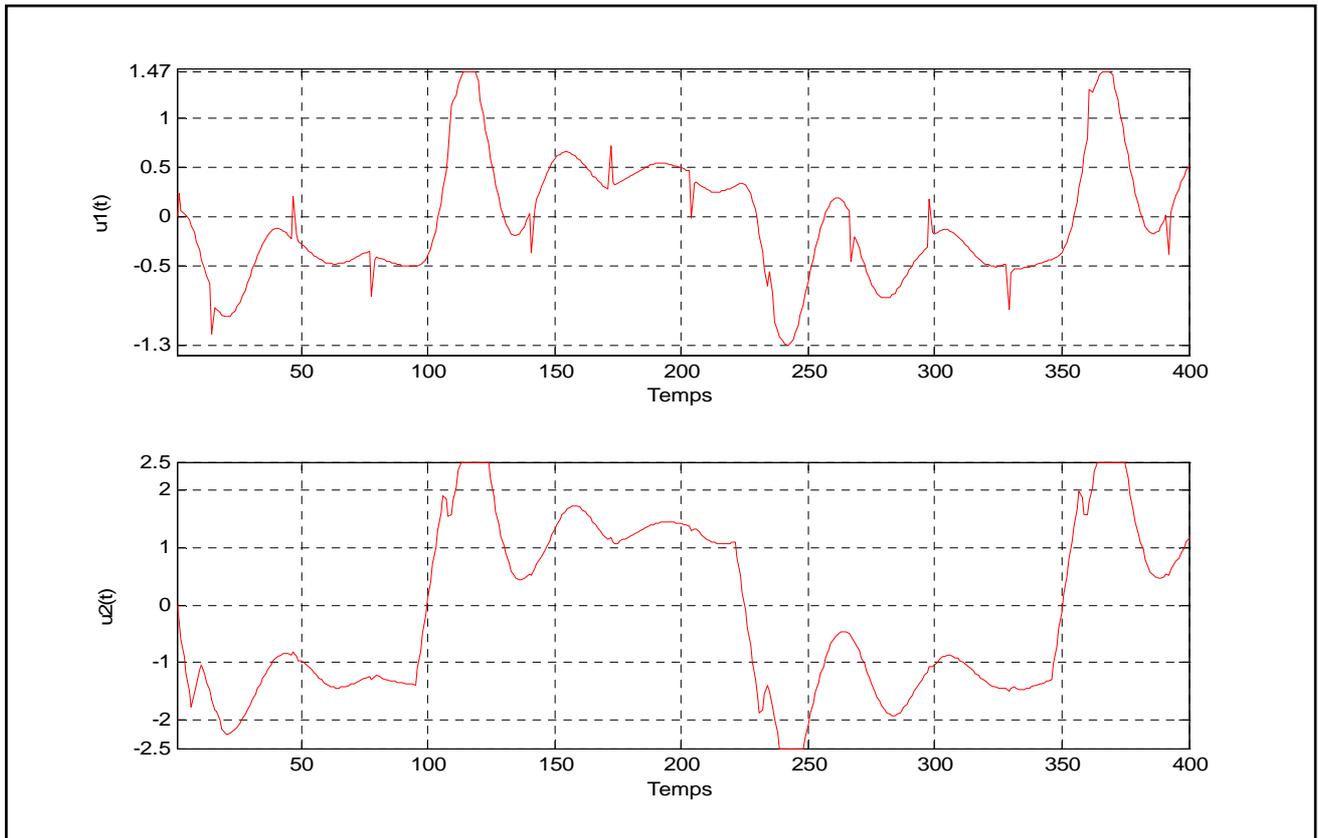


Fig.3.3.b. Commandes fournies par le GPC multivariable utilisant la fonction *quadprog*

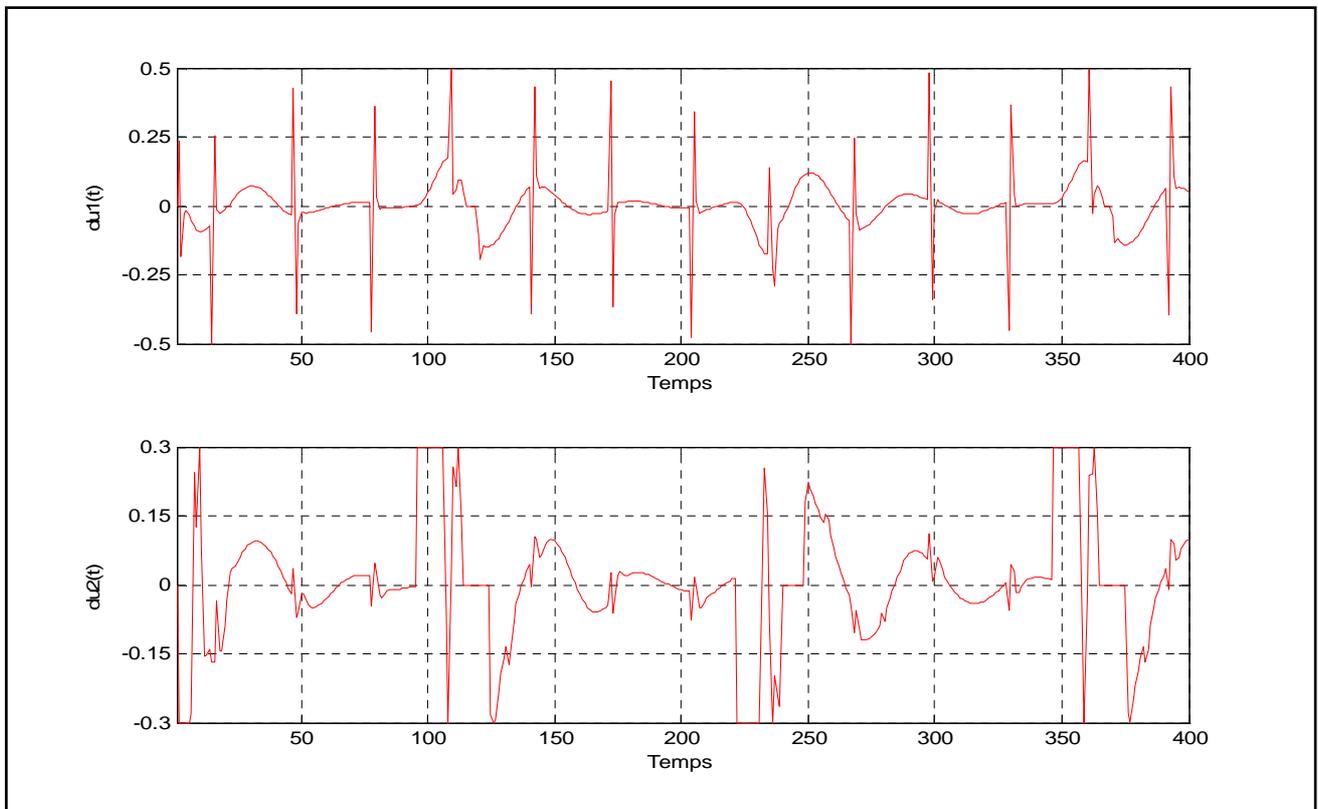


Fig.3.3.c. Incrément de commandes fournis par le GPC multivariable utilisant la fonction *quadprog*

Dans les figures 3.3, on peut remarquer que :

- Le pic du premier cas est annulé complètement, en revanche, d'autres limites maximales et minimales non souhaitables surviennent dans des plages du temps très réduites.
- L'effet de la commande  $u_1(t)$  sur la réponse  $y_2(t)$ . Ceci dénote le couplage des sorties aux entrées du système.

Comme résultat, on peut conclure qu'avec ce choix de contraintes, la fonction *quadprog* restreinte l'application de la commande prédictive multivariable sur ce processus. Nous verrons, dans le chapitre suivant, qu'avec d'autres méthodes d'optimisation, on peut aboutir à des résultats proches où mieux à ceux obtenus par le GPC utilisant cette fonction.

### 3.6. Conclusion:

- Dans ce chapitre, nous avons présenté les différents types des contraintes agissant sur les signaux d'entrées et de sorties du système, nous avons également présenté le principe de la fonction utilisée dans la bibliothèque de Matlab pour résoudre le problème quadratique utilisé dans la commande prédictive multivariable sous contraintes.
- Puisque le problème quadratique sous contraintes de type inégalité est très difficile à résoudre grâce aux contraintes implicites lors de quelques applications, à cet effet, nous avons présenté une technique permettant de transformer ce type des contraintes et les combiné à celles implicites pour reconstruire un domaine de faisabilité permettant de faire élargir l'application des autres méthodes d'optimisation qui font l'objet de la partie suivante du travail.
- Nous avons montré à partir des résultats d'une simulation effectuée sur un processus multivariable que la fonction *quadprog* est inefficace puisque elle présente plusieurs inconvénients, et en conséquence de quoi, nous proposerons, dans le chapitre suivant, d'autres méthodes d'optimisation permettant de pallier à ces inconvénients.

#### 4. Introduction :

L'application de la commande prédictive multivariable utilisant la fonction *quadprog* sur un système électromécanique, cité précédemment, donne une meilleure dynamique de poursuite, cette dernière dépend directement du choix donné avec précaution de l'ensemble des contraintes. Un tel choix peut limiter cette application du point de vue pratique.

Dans la suite de ce travail, on essaiera d'élargir le domaine d'application de la commande prédictive, ceci par l'utilisation de deux types principaux:

- Méthodes utilisant la dérivée de la fonction objective ;
- Méthodes n'utilisant pas la dérivée de la fonction objective.

L'avantage de ces dernières réside dans le fait qu'elles ne nécessitent que le seul calcul de la fonction en certains nombres de points dits supports. La plupart d'entre elles ne supposent ni la continuité ni la dérivabilité de la fonction à minimiser.

Cependant, dans tous les cas, on cherche à déterminer le minimum d'une fonction quadratique à l'intérieur d'une région de départ limitée par le domaine  $[\gamma'_{\min} \quad \gamma'_{\max}]$ , cette région dépend du choix des contraintes sur signaux d'entrées- sorties du système, ce choix peut restreindre le domaine de variation de la solution et il est concevable que si  $[\gamma'_{\min} \quad \gamma'_{\max}]$  est mal choisi, il ne contient pas la solution globale du problème.

##### 4.1. Méthodes utilisant les dérivées de la fonction objective:

Le principe général de ce type de méthodes consiste à partir d'un point faisable  $\tilde{U}_j$  de trouver une direction faisable  $d_j$  telle qu'il existe un pas de déplacement  $\alpha$  pour que  $\tilde{U}_j + \alpha.d_j$  soit faisable et la fonction objectif pour  $\tilde{U}_j + \alpha.d_j$  devient meilleure que celle correspondant à  $\tilde{U}_j$ .

Une fois la direction faisable construite, une optimisation unidimensionnelle est résolue pour déterminer la valeur optimale  $\alpha_{opt}$  de déplacement le long de  $d_j$ . Ce choix conduit à un nouveau point  $\tilde{U}_{j+1}$  et la procédure est réitérée.

Une de ces méthodes utilisée dans ce travail est celle de la « méthode des gradients conjugués ». Les détails de cette méthode sont trouvés dans [MB93].

**4.1.1. Méthode des gradients conjugués [MB93]:**

Partant du problème d'optimisation quadratique (3.25):

$$P'' : \begin{cases} \text{Min } f(\tilde{U}) = \frac{1}{2} \tilde{U}^T Q_2 \tilde{U} + \tilde{U}^T Q_1 + Q_0 \\ \gamma_{\min} \leq \tilde{U} \leq \gamma_{\max} \end{cases} \quad (4.1)$$

La fonction  $f(\tilde{U})$  est alors une fonctionnelle strictement convexe deux fois continument différentiable, son gradient donne :

$$\nabla f(\tilde{U}_{opt}) = Q_2 \tilde{U} + Q_1 \quad (4.2)$$

Où  $\nabla^2 f = Q_2$  représente le Hessien de  $f$  qui est défini positif.

En absence des contraintes, l'idée de cette méthode est de construire progressivement des directions  $d_0, d_1, \dots, d_j$  mutuellement conjuguées par rapport à la matrice  $Q_2$ , ces directions doivent vérifier la condition :

$$\left. \begin{array}{l} \forall i (0 \leq i \leq n-1) \\ \forall j (0 \leq j \leq n-1) \\ i \neq j \end{array} \right\} \Rightarrow d_i^T Q_2 d_j = 0 \quad (4.3)$$

à chaque étape  $j$ , la direction  $d_k$  est obtenue par combinaison linéaire du gradient  $-\nabla f(\tilde{U}_j)$  en  $\tilde{U}_j$ , et des directions précédentes  $d_0, d_1, \dots, d_{j-1}$ , les coefficients de la combinaison linéaire étant choisis de telle sorte que  $d_j$  soit conjuguée par rapport à toutes les directions précédentes.

Cette méthode présente l'avantage de converger vers une solution globale grâce à la convexité de la fonction objective et la nature des contraintes utilisées. Parfois et pour certains cas, le calcul des composantes de  $\tilde{U}_{j+1}$  peut coïncider avec des valeurs situées à l'extérieure de son limites  $[\gamma_{\min}, \gamma_{\max}]$  conduisant à des points infaisables. Il s'agit pour ce cas de remplacer les composantes de l'optimum sans contraintes par leurs limites.

Des tels choix s'avèrent parfois très éloignés (voir la figure ci-dessous) du véritable optimum avec contraintes

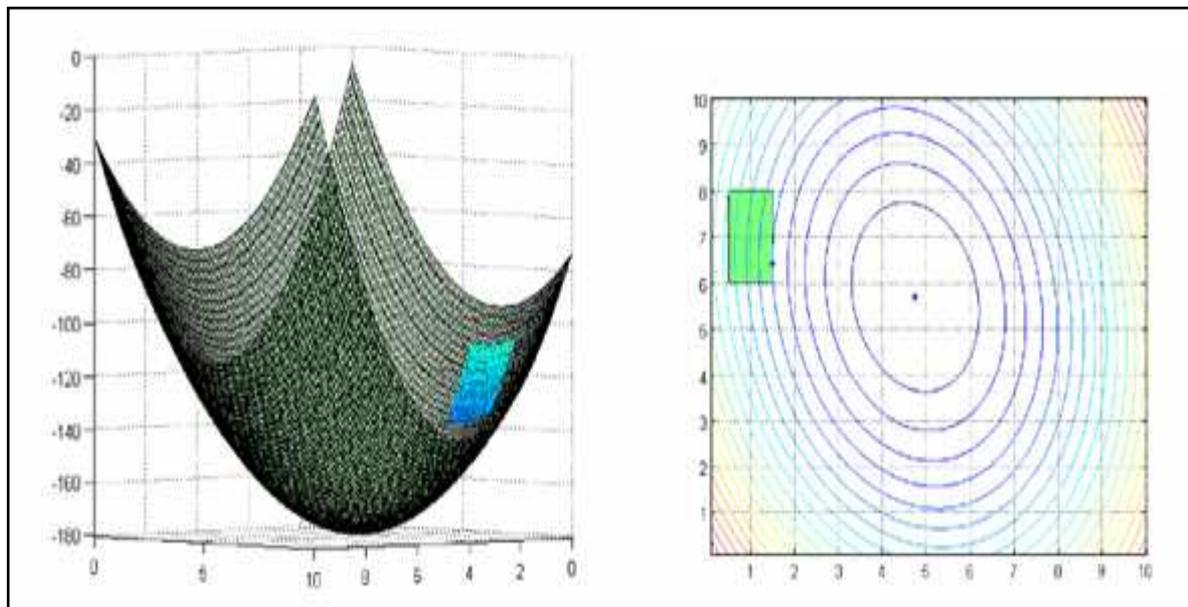


Fig. 4.1. Argument de l'optimum sans contraintes et de l'optimum avec contraintes

#### 4.1.2. Algorithme de la méthode des gradients conjugués:

L'algorithme de la méthode des gradients conjugués peut être résumé en quatre étapes :

##### Etape 0: Initialisation

- Soit choisir, pour  $k=1$ , le point de départ  $\tilde{U}_k$ .

- Le gradient de  $f$  en  $\tilde{U}_k$  est obtenu par :  $\nabla f(\tilde{U}_k) = Q_2 \tilde{U}_k + Q_1$  (4.4)

- Pour, initialiser la direction de recherche  $d_k$  telle que :  $d_k = -\nabla f(\tilde{U}_k)$  (4.5)

##### Etape 1 : Calcul du point suivant

- A l'itération  $k$ , on est au point  $\tilde{U}_k$ , le point devint sera égal  $\tilde{U}_{k+1} = \tilde{U}_k + \alpha_k d_k$

où  $\alpha_k = -\frac{\nabla f(x_k)^T d_k}{d_k^T Q_2 d_k}$  (4.6)

##### Etape 2 : Calcul de la direction $d$

Déterminer :  $d_{k+1} = -\nabla f(\tilde{U}_{k+1}) + \beta_k d_k$  (4.7)

Avec :  $\beta_k = \frac{\nabla f(x_{k+1})^T Q_2 d_k}{d_k^T Q_2 d_k}$  (4.8)

##### Etape 3 : test d'arrêt

Si  $f(\tilde{U}_{k+1}) \approx f(\tilde{U}_k)$  : Remplacer  $k$  par  $k+1$  et retourner à l'étape 1

(Noter que, le dénominateur  $d_k^T Q_2 d_k$  ne peut être nul).

## 4.2. Méthodes d'optimisation n'utilisant pas les dérivées de la fonction objective:

Considérons le problème d'optimisation obtenu à partir de (3.22):

$$P'' : \begin{cases} \text{Min } f(\tilde{U}) = \frac{1}{2} \tilde{U}^T Q_2 \tilde{U} + \tilde{U}^T Q_1 + Q_0 \\ \gamma'_{\min} \leq \tilde{U} \leq \gamma'_{\max} \end{cases}$$

Ainsi l'intervalle de départ est initialement de longueur :

$$L^1 = x_{\min}^1 - x_{\max}^1 \quad (4.9)$$

$$\text{Avec : } \begin{bmatrix} x_{\min}^1 & x_{\max}^1 \end{bmatrix} = \begin{bmatrix} \gamma'_{\min} & \gamma'_{\max} \end{bmatrix}$$

L'objet de ce type de méthodes est de procéder à la réduction de cet intervalle en deçà d'une limite finale  $L^n$ . Cette dernière définit la précision avec laquelle on localise la solution  $\tilde{U}_{opt}$ .

Un type de ces méthodes utilisant ce principe est celui de la « méthode de Dichotomie », son principe est décrit de la manière suivante :

### 4.2.1. Méthode de Dichotomie :

On commence par le cas unidimensionnel pour illustrer le principe de Dichotomie, ensuite on essaiera de le généraliser pour le cas multidimensionnel.

#### 4.2.1.1. Cas unidimensionnel [GB91]:

Cette méthode permet, à chaque pas, de diviser par deux la longueur de l'intervalle contenant l'optimum en calculant la fonction  $f$  en deux points déterminés. En acceptant  $N$  évaluations de la fonction  $f(\tilde{U})$ , on peut ainsi réduire la longueur initiale  $L^1$ , dans un rapport de réduction:

$$\frac{L^n}{L^1} = \left(\frac{1}{2}\right)^{n-1} \quad (4.10)$$

Où  $n$  représente le nombre nécessaire pour réduire l'intervalle initial de  $L^1$  à  $L^n$

Les démarches à suivre pour localiser une solution d'un problème quadratique sous contraintes des bornes peuvent être résumées comme suit :

- a) Au début (pour l'itération  $k = 1$ ), on part de l'intervalle initiale  $[x_{\min}^k \quad x_{\max}^k]$ . En considérant le point milieu  $c^k$  et les deux points de support  $x_1^k$  et  $x_2^k$  tels que :

$$c^k = \frac{x_{\max}^k + x_{\min}^k}{2} \quad (4.11)$$

$$x_1^k = \frac{x_{\min}^k + c_1^k}{2} \quad (4.12)$$

$$x_2^k = \frac{c_1^k + x_{\max}^k}{2} \quad (4.13)$$

- b) En évaluant la fonction  $f$  dans les trois points ci-dessus et en utilisant l'unimodalité, il est aisé de voir que l'on peut toujours éliminer deux des quatre sous intervalles (parce que l'optimum ne peut y être situé) et qu'il subsiste seulement deux sous intervalles contigus :

$$\left\{ [x_{\min}^k \quad x_1^k], [x_1^k \quad c^k] \right\} \text{ où } \left\{ [c^k \quad x_2^k], [x_2^k \quad x_{\max}^k] \right\}.$$

- c) Le nouvel intervalle  $[x_{\min}^{k+1} \quad x_{\max}^{k+1}]$  doit être choisit à partir de ces deux conditions:

$$x_{\min}^{k+1} \leftarrow c^k; \quad c^{k+1} \leftarrow x_2^k \quad \text{Si : } f(x_1^k) \phi f(x_2^k) \quad (4.14)$$

$$x_{\max}^{k+1} \leftarrow c^k; \quad c^{k+1} \leftarrow x_1^k \quad \text{Si : } f(x_1^k) \pi f(x_2^k) \quad (4.15)$$

- On se ramène ainsi au même problème sur le segment  $[x_{\min}^{k+1} \quad x_{\max}^{k+1}]$  de longueur moitié et de point milieu  $c^{k+1}$ . Pour l'itération suivante, on aura uniquement deux évaluations de la fonction objective dans les points de support  $x_1^{k+1}$  et  $x_2^{k+1}$ .
- On réitère ce processus de réduction  $(n-1)$  fois, la longueur de l'intervalle final sera alors:

$$L^n = x_{\max}^n - x_{\min}^n \quad (4.16)$$

Cette dernière peut être exprimée en fonction de l'intervalle initial avec :

$$L^n = \left(\frac{1}{2}\right)^{n-1} (x_{\max}^1 - x_{\min}^1) \quad (4.17)$$

On réalise trois évaluations de la fonction objective à la première itération, puis deux expériences à chaque itération suivante, soit au total :

$$N = 2(n-1)+1 \quad (4.18)$$

La relation entre le nombre d'évaluation de  $f$  et le rapport de réduction  $R_{dichotomie} = \frac{L^n}{L^1}$  s'écrit :

$$N = 1 + 2 \times \frac{\log(R_{dichotomie})}{\log\left(\frac{1}{2}\right)} \quad (4.19)$$

$$\text{Soit : } N = 1 - 2.89 \times \log(R_{dichotomie}) \quad (4.20)$$

Le tableau suivant, donne le rapport de réduction de l'intervalle initiale en fonction de nombre  $N$  de d'évaluations de la fonction  $f$

| $R_{dichotomie}$ | $N$ |
|------------------|-----|
| $10^{-2}$        | 17  |
| $10^{-3}$        | 23  |
| $10^{-4}$        | 29  |

#### 4.2.1.2. Cas pluridimensionnel:

Le principe de la recherche multidimensionnelle de la solution du problème quadratique sous contraintes explicites par la méthode de Dichotomie est illustré par le cas tridimensionnel  $\mathfrak{R}^3$  ci-dessous. Ce principe est généralisé, à chaque fois, pour l'ensemble de l'espace  $\mathfrak{R}^n$  ( $n = N_u.m$ ).

Considérons le problème quadratique sous contraintes de bornes dont les variables de la fonction objective  $f$  sont données par le vecteur  $x = (x_1 \quad x_2 \quad x_3)^T$ .

Supposons que l'on peut limiter la région de faisabilité par l'ensemble des contraintes :

$$\begin{cases} \gamma'_{1\min} \leq x_1 \leq \gamma'_{1\max} \\ \gamma'_{2\min} \leq x_2 \leq \gamma'_{2\max} \\ \gamma'_{3\min} \leq x_3 \leq \gamma'_{3\max} \end{cases} \quad (4.21)$$

Au départ, on suppose, pour  $k = 1$ , les intervalles  $\left[ x_i^k \min \quad x_i^k \max \right]_{i=1, \dots, n}$  donnés par:

$$\begin{bmatrix} x_{i \min}^k & x_{i \max}^k \end{bmatrix} = \begin{bmatrix} \gamma'_{i \min} & \gamma'_{i \max} \end{bmatrix} \quad (4.22)$$

Ces derniers peuvent être partitionnés tel que :

$$voie_i \rightarrow x_{i \min}^k \leq x_{i1}^k \leq c_i^k \leq x_{i2}^k \leq x_{i \max}^k \quad (4.23)$$

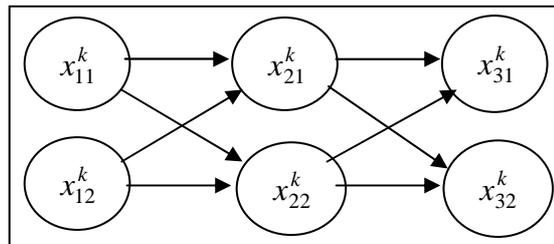
Avec :

$$c_i^k = \frac{x_{i \min}^k + x_{i \max}^k}{2} \quad (4.24)$$

$$x_{i1}^k = \frac{x_{i \min}^k + c_i^k}{2} \quad (4.25)$$

$$x_{i2}^k = \frac{c_i^k + x_{i \max}^k}{2} \quad (4.26)$$

- A partir de ces points de supports, on peut construire  $s = 2^n$  vecteurs  $y_{s(s=1, \dots, n)}$  selon la structure :



Avec :

$$y_1 = (x_{11}^k \ x_{21}^k \ x_{31}^k)^T, \ y_2 = (x_{11}^k \ x_{21}^k \ x_{32}^k)^T, \ y_3 = (x_{11}^k \ x_{22}^k \ x_{31}^k)^T, \ y_4 = (x_{11}^k \ x_{22}^k \ x_{32}^k)^T \quad (4.27)$$

$$y_5 = (x_{12}^k \ x_{21}^k \ x_{31}^k)^T, \ y_6 = (x_{12}^k \ x_{21}^k \ x_{32}^k)^T, \ y_7 = (x_{12}^k \ x_{22}^k \ x_{31}^k)^T, \ y_8 = (x_{12}^k \ x_{22}^k \ x_{32}^k)^T \quad (4.28)$$

- On évalue la fonction  $f$  pour chaque vecteur  $y_s$ , et on détermine celui qui fournit la plus petite valeur de  $f$ . noter le par  $y_{r(1 \leq r \leq s)}$ .

- A partir des composantes de  $y_r$ , on peut construire un nouveau intervalle  $\begin{bmatrix} x_{\min}^{k+1} & x_{\max}^{k+1} \end{bmatrix}$  telles que les deux conditions suivantes:

- Si  $x_{ij(r)}^k \ \pi \ c_i^k \Rightarrow \begin{bmatrix} x_{i \min}^{k+1} & x_{i \max}^{k+1} \end{bmatrix} = \begin{bmatrix} x_{i \min}^k & c_i^k \end{bmatrix} \quad (4.29)$

- Sinon  $\begin{bmatrix} x_{i \min}^{k+1} & x_{i \max}^{k+1} \end{bmatrix} = \begin{bmatrix} c_i^k & x_{i \max}^k \end{bmatrix} \quad (4.30)$

- Si l'on suppose, par exemple, que  $y_s = y_7$  le nouvel intervalle accepté sera:

$$\left[ \begin{array}{cc} x_{\min}^{k+1} & x_{\max}^{k+1} \end{array} \right] : \begin{cases} x_{12}^k \phi c_1^k \rightarrow \left[ \begin{array}{cc} x_{1 \min}^{k+1} & x_{1 \max}^{k+1} \end{array} \right] = \left[ \begin{array}{cc} c_1^k & x_{1 \max}^k \end{array} \right] \\ x_{22}^k \phi c_2^k \rightarrow \left[ \begin{array}{cc} x_{2 \min}^{k+1} & x_{2 \max}^{k+1} \end{array} \right] = \left[ \begin{array}{cc} c_2^k & x_{2 \max}^k \end{array} \right] \\ x_{31}^k \pi c_3^k \rightarrow \left[ \begin{array}{cc} x_{3 \min}^{k+1} & x_{3 \max}^{k+1} \end{array} \right] = \left[ \begin{array}{cc} x_{3 \min}^k & c_3^k \end{array} \right] \end{cases} \quad (4.31)$$

- On se ramène ainsi au même problème sur les segments  $\left[ x_{i \min}^{k+1} \quad x_{i \max}^{k+1} \right]$  de longueur égale à

$$L_i^{k+1} = \frac{1}{2} (x_{i \max}^k - x_{i \min}^k) \quad (3.32)$$

- On réitère ce processus de réduction  $(n-1)$  fois, la longueur obtenue dans toutes les voie est égale à  $L^n$  et le nombre d'évaluation de la fonction objective devient alors:

$$N = 3^{m \times N_u} + 2^{m \times N_u} \cdot (n-1) \quad (3.33)$$

$$\text{Avec : } n = \max(n_1, n_2, \dots, n_n) \quad (3.34)$$

$$\text{Et : } n_i = 1 + \frac{\log\left(\frac{L^n}{L_i^1}\right)}{\log\left(\frac{1}{2}\right)} \quad (3.35)$$

#### 4.2.2. Amélioration de la méthode de Dichotomie :

Cette méthode n'est pas très efficace, surtout, d'une part, si le rapport de réduction  $R_{dich}$  est très petit et si l'évaluation de la fonction objective requiert un effort de calcul important, et d'autre part par le choix d'un nouvel intervalle, pour l'itération suivante, si la grandeur de  $f$  devient égale pour les deux points supports  $x_1^k$  et  $x_2^k$ . Notre objectif est alors précis, nous allons essayer d'améliorer le rapport de réduction de cette méthode, ainsi de diminuer le nombre d'évaluation de  $f$ .

Dans ce qui est suit, nous présenterons, deux autres versions de la Dichotomie permettant de palier les inconvénients de la méthode précédente. Ces méthodes sont : la méthode Tric\_quad et la méthode de Dich\_incr.

##### 4.2.2.1. Méthode de Tric-quad:

Cette méthode combinant deux principes différents : principe de Quadritomie et celui de Trichotomie.

- Le premier principe est utilisé pour partitionner, pour chaque passage de réduction, l'intervalle de recherche  $[x_{\min}^k \quad x_{\max}^k]$  quatre segments équidistants tels que :

$$[x_{\min}^k \quad x_{\max}^k] = \left\{ [x_{\min}^k \quad x_1^k], [x_1^k \quad x_2^k], [x_2^k \quad x_3^k], [x_3^k \quad x_{\max}^k] \right\} \quad (4.36)$$

- Ces segments sont utilisés, par la suite, pour construire trois sous intervalles, en utilisant le principe de Trichotomie tels que les suivants:

$$[x_{\min}^k \quad x_1^k \quad x_2^k], [x_1^k \quad x_2^k \quad x_3^k], [x_2^k \quad x_3^k \quad x_{\max}^k] \quad (4.37)$$

- La fonction objective est évaluée, par la suite, dans les trois points de milieu de chaque sous intervalle précédent.
- On peut éliminer à la fois, suivant l'unimodalité de  $f$ , deux sous intervalles parmi les trois donnés par (4.37) et on accepte uniquement l'intervalle où son point de milieu mène vers la plus petite valeur de  $f$ . Le nouvel intervalle  $[x_{\min}^{k+1} \quad x_{\max}^{k+1}]$  sera donc :

$$[x_{\min}^{k+1} \quad x_{\max}^{k+1}] = \begin{cases} [x_{\min}^k \quad x_2^k]: & \text{si } f_{\min} = f(x_1^k) \\ [x_1^k \quad x_3^k]: & \text{si } f_{\min} = f(x_2^k) \\ [x_2^k \quad x_{\max}^k]: & \text{si } f_{\min} = f(x_3^k) \end{cases} \quad (4.38)$$

- On se ramène ainsi au même problème sur le segment  $[x_{i \min}^{k+1} \quad x_{i \max}^{k+1}]$  de longueur moitié. On réitère ce procédé à partir du nouvel intervalle et on le réduit jusqu'à ce dernier soit en deçà de la précision choisie pour la détermination de la solution du problème.

D'après (4.38), deux avantages principaux peuvent être cités à partir de la condition  $f_{\min} = f(x_2^k)$ . Le premier avantage est représenté par la réduction simultanée des deux limites (supérieure et inférieure) de l'intervalle de recherche.

L'autre avantage permet le choix d'un nouvel intervalle pour l'opération suivante où la valeur de  $f$  devient égale dans les deux points supports  $x_1^k$  et  $x_2^k$ .

Cette méthode possède le même nombre d'itération que la méthode précédente, cependant, le nombre d'évaluation de la fonction objective devient relativement grand par rapport à la méthode de Dichotomie, soit pour le cas général:

$$N = 3^{N_u \times m} \cdot n \quad (4.39)$$

**4.2.2.2. Méthode de Dichotomie utilisant le paramètre  $\Delta L$  (Dich\_incr) [GB91] :**

Elle peut être considérée comme une variante de la Dichotomie. Pour une recherche unidimensionnelle le principe de la méthode Dich\_incr est décrit comme suit: (Etant donnée la longueur finale  $L^n$  pour laquelle on désire réduire la longueur initiale  $L^0$  en  $L^n$ )

Soit  $\Delta L$  un paramètre, qui peut être considéré comme une incertitude sur  $\Delta L$ , choisi de tel sorte que :

$$\Delta L = \rho \times L^n \quad (4.40)$$

Où  $0 < \rho < \frac{1}{2}$

Au début, on situe les deux points de support  $x_1$  et  $x_2$  éloignés respectivement par  $-\Delta L$  et  $+\Delta L$  par rapport au centre de l'intervalle  $[x_{\min}^k \quad x_{\max}^k]$  comme suit:

$$x_1^k = \frac{1}{2}(x_{\max}^k + x_{\min}^k) - \Delta L \quad (4.41)$$

$$x_2^k = \frac{1}{2}(x_{\max}^k + x_{\min}^k) + \Delta L \quad (4.42)$$

- Il faut noter que les points supports  $x_1$  et  $x_2$  de cette méthode sont complètement différents de ceux trouvés par les deux méthodes précédentes. C'est pourquoi on ne trouve pas les mêmes résultats.

- A partir de ces deux points, on peut construire les deux intervalles équidistants:

$$[x_{\min}^k \quad x_2^k] \text{ et } [x_1^k \quad x_{\max}^k] \quad (4.43)$$

On évalue la fonction  $f$  dans les deux points supports  $x_1^k$  et  $x_2^k$ , on utilise ensuite l'unimodalité de  $f$ .

On obtient, par conséquent, le nouvel intervalle  $[x_{\min}^{k+1} \quad x_{\max}^{k+1}]$  de longueur  $L^{k+1} = \frac{L^k}{2} + \Delta L$ , ce dernier devient égal à l'un de ces deux intervalles:

$$[x_{\min}^k \quad c^k + \Delta L] \text{ où } [c^k - \Delta L \quad x_{\max}^k] \quad (4.44)$$

$$\text{Avec : } c^k = \frac{1}{2}(x_{\max}^k + x_{\min}^k)$$

- On réitère ce processus  $(n-1)$ , l'intervalle final devient :

$$L^n = \frac{L^1}{2^{n-1}} + \Delta L \left( 1 + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^{n-2}} \right) \quad (4.45)$$

$$\text{D'où : } L^n = \frac{L^1}{2^{n-1}} + \left( 1 - \frac{1}{2^{n-1}} \right) \cdot \Delta L \quad (4.46)$$

- pour chaque itération, on réalise uniquement deux évaluations de  $f$ , d'où le nombre total d'évaluation pour le cas général:

$$N = 2^{N_u \times m} \cdot (n-1) \quad (4.47)$$

- Ce dernier est lié au rapport de réduction par :

$$N = -2.89 \times \log \left( \frac{L^n - 2 \cdot \Delta L}{L^1 - 2 \cdot \Delta L} \right) \quad (4.48)$$

- Par l'utilisation de (4.40) dans l'expression (4.48), le rapport de réduction de cette méthode devient :

$$R_{dich\_inc} = \frac{\exp\left(-\frac{N}{2.89}\right)}{1 - 2 \times \rho \cdot \left[ 1 - \exp\left(-\frac{N}{2.89}\right) \right]} \quad (4.49)$$

- Remarquons que, le rapport de réduction de cette méthode dépend d'un choix donné sur  $\rho$ . Un bon choix réalisé sur ce paramètre peut rendre ce rapport meilleur que celui fournis par la Dichotomie.
- A partir de l'équation (4.20) on peut tirer le rapport de réduction de la méthode de Dichotomie en fonction du nombre d'évaluation de  $f$  avec :

$$R_{dichotomie} = \exp\left(\frac{1-N}{2.89}\right) \quad (4.50)$$

- La figure suivante représente une comparaison entre le rapport de réduction de la méthode de Dichotomie et celle utilisant le paramètre d'incertitude  $\Delta L$ .

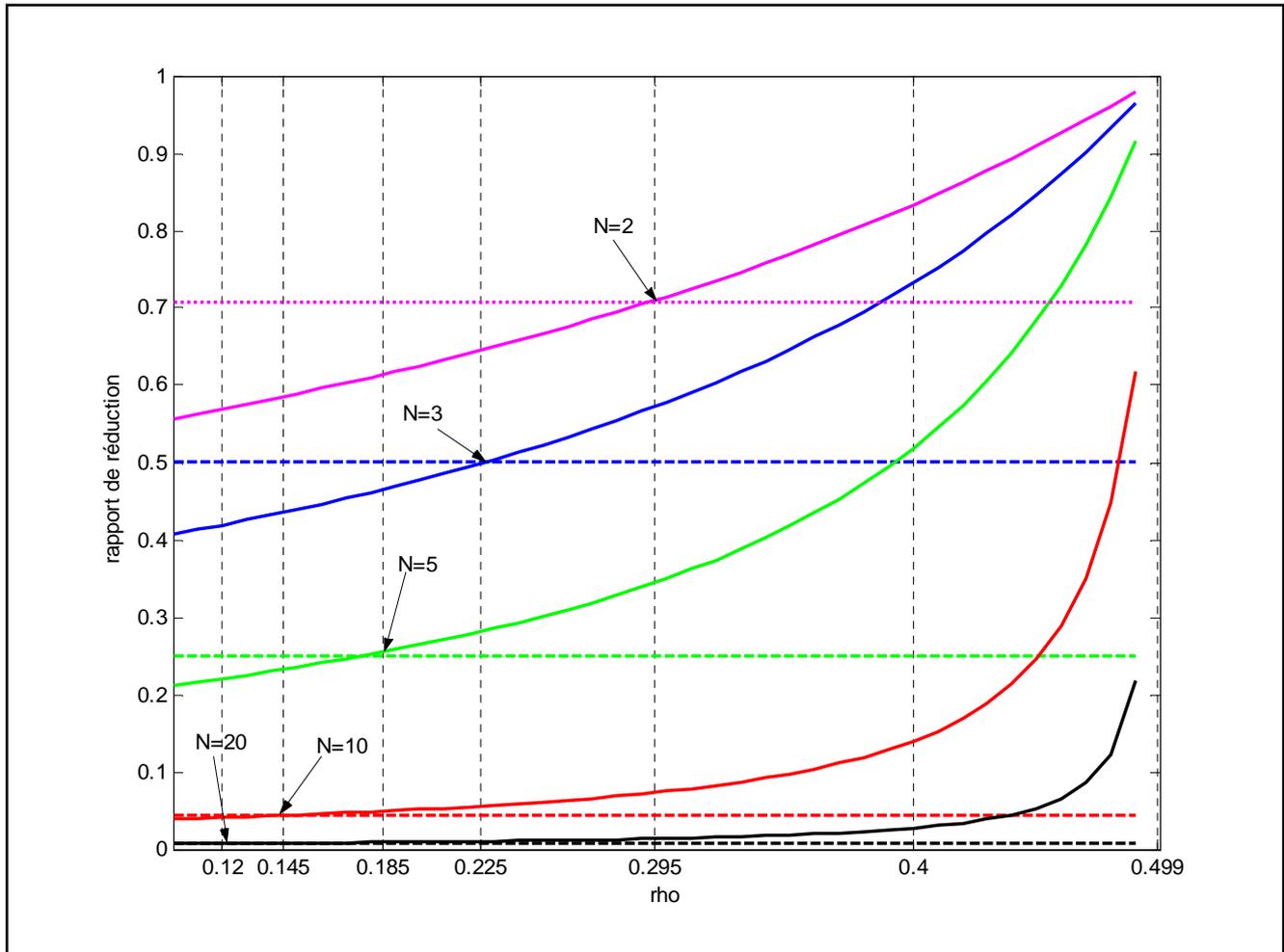


Fig.4.2. Comparaison entre les fonctions :  $R_{dich\_incr}(N, \rho)$ ,  $R_{dichotomie}(N)$

Les lieux discontinus et continus représentent respectivement les fonctions :  $R_{dichotomie}(N)$  et  $R_{dich\_incr}(N, \rho)$ .

- D'après la figure 4.2, on peut remarquer que, à partir d'un choix  $\Delta L \geq 29.5\%$  de la limite finale  $L^n$  (Celle qui correspond  $\rho \geq 0.295$ ), le rapport de réduction fournis par la méthode Dich\_incr, pour  $N \geq 2$  évaluations de  $f$  sera meilleur que celui donné par la Dichotomie. Ce rapport de réduction devient plus en plus meilleur si ce choix est proche de la longueur finale  $L^n$ , ceci quelque soit le nombre d'évaluation  $N$ .

### 4.3. GPC utilisant les méthodes d'optimisation proposées:

On utilise ces méthodes proposées pour le but de résoudre le problème de commande de la loi de commande prédictive multivariable sous contraintes de bornes, pour cela, nous commençons, tout d'abord, par la factorisation la matrice de transfert du système à commander (si sa dynamique est décrite par une matrice de transfert) afin de calculer les deux matrices polynomiales  $A, B$  de CARIMA.

Choisissons, dans la suite, les meilleurs paramètres de synthèse  $N_{2_i}, N_{u_i}, \lambda_i$  de la méthode GPC et finalement de mettre toutes limites des signaux contraints. Le calcul suivant passe par deux phases:

**Phase1** : *Calcul hors ligne* :

Pour la méthode GPC :

- On détermine les matrices des polynômes  $E, F$  et  $G$  à partir de résolutions de deux équations Diophantiennes.
- On initialise les vecteurs de commandes et de sorties.

Pour les méthodes d'optimisation proposées :

- On choisi la précision qui définit la longueur finale  $L^n$ .
- On calcul, pour les méthodes : Dichotomie, Tric\_quad et Dich\_inc, le nombre maximal d'itérations qui correspond cette longueur donnée. On peut choisir, pour la méthode des gradients conjugués, une précision  $\zeta$  égale la valeur de la limite finale  $L^n$ .

**Phase2** : *Calcul en ligne* :

Pour chaque instant du temps  $t$  :

- On applique l'approche, proposée dans le chapitre précédent, pour transformer toutes contraintes considérées en des contraintes de bornes. On obtient pour chaque voie ( $i = 1, 2, \dots, m$ ) la limite :

$$\gamma'_{i \min} \leq \Delta u_1(t) \leq \gamma'_{i \max} \quad (4.51)$$

- On détermine la matrice des polynômes  $Fc$  puis, on détermine  $Q_2, Q_1$  et  $Q_0$ .
- On construit la matrice  $L$  et le vecteur  $M$ .
- On applique selon le choix de l'utilisation, une méthode d'optimisation parmi lesquelles proposées pour déterminer la solution  $\tilde{U}_{opt}$ .
- Cette solution est utilisée pour le calcul de commandes, en utilisant le principe de l'horizon glissant, ainsi les sorties du système

#### 4.4. Tests et simulations :

##### 4.4.1. Application1:

Le système à commander est celui donné par §3.5. On utilise les mêmes contraintes citées dans (3.36), (3.37) et mêmes paramètres de synthèse de GPC. Cette application a pour but d'améliorer la réponse  $y_1(t)$  qui comporte un dépassement très élevé fournit via le GPC utilisant la fonction *quadprog*. Pour cela, cette dernière est remplacée, chaque fois, par l'une de ces quatre approches proposées précédemment, on effectue le choix sur  $L^n$  avec  $L^n = 10^{-3}$  et pour la méthode *dich\_incr* on choisit  $\rho = 2 \times 10^{-3}$ , les réponses, les incréments et les commandes sont présentés par les figures suivantes:

- **GPC multivariable utilisant la méthode des gradients conjugués :**

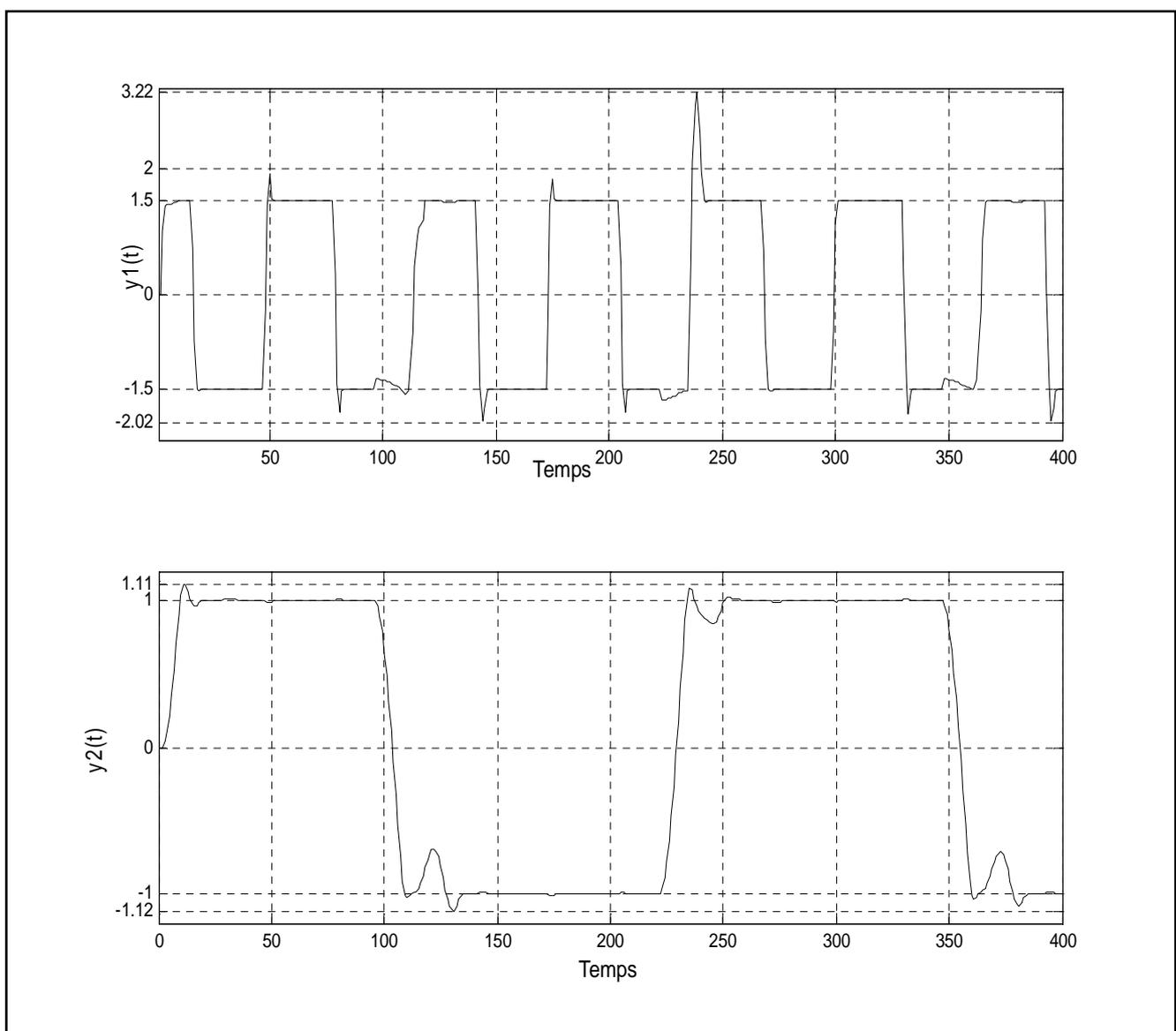


Fig.4.3.a. Sorties obtenues par le GPC utilisant la méthode des gradients conjugués

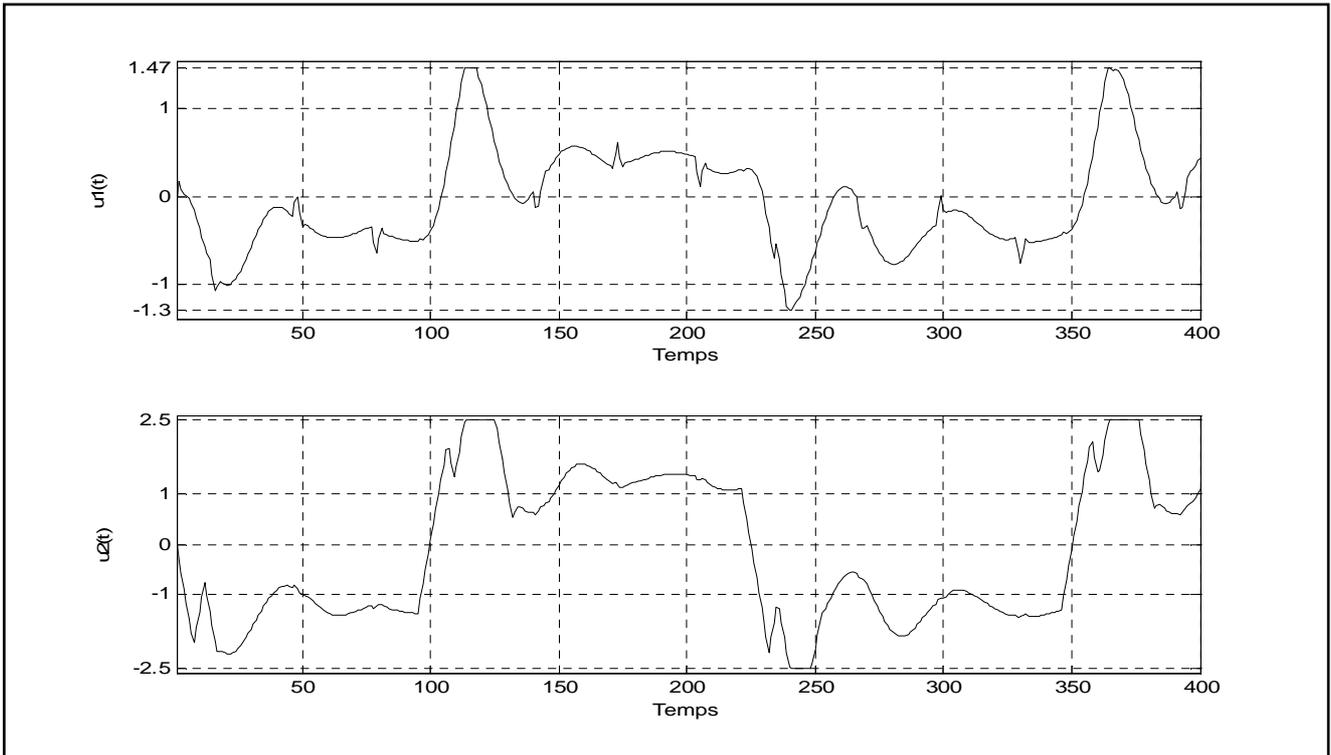


Fig.4.3.b. Commandes fournies par le GPC utilisant la méthode des gradients conjugués

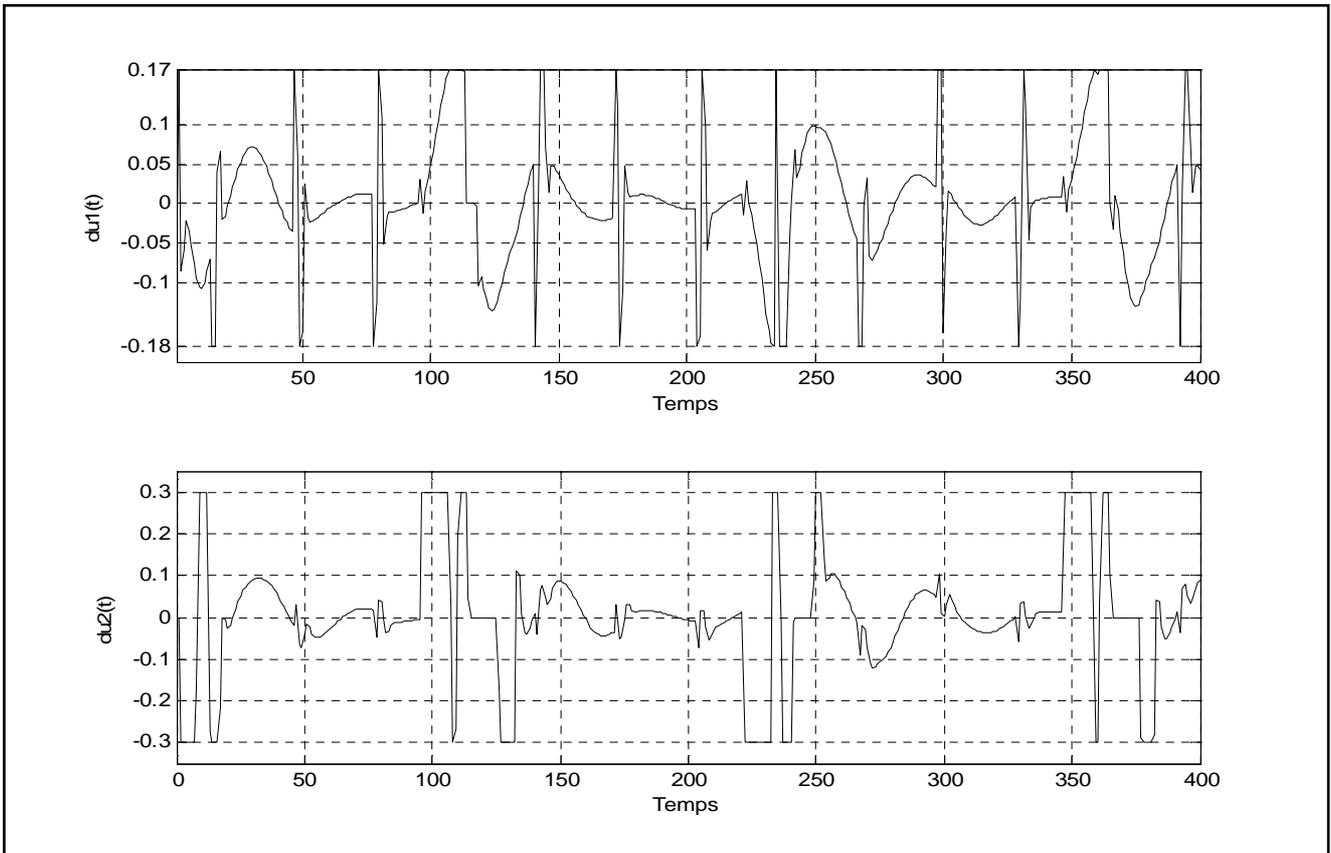


Fig.4.3.c. Incréments de commandes fournis par le GPC utilisant la méthode des gradients conjugués

- **GPC multivariable utilisant la méthode de Dichotomie:**

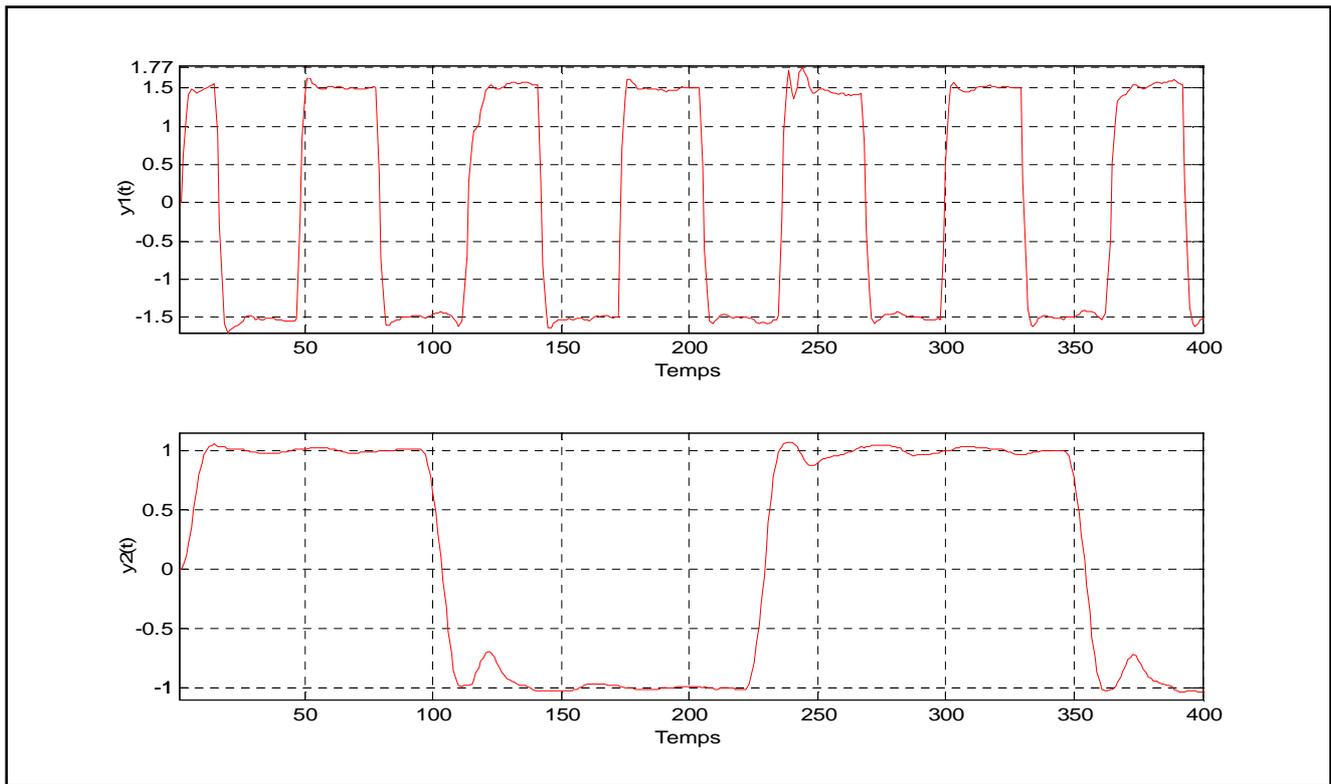


Fig.4.4.a. Sorties obtenues par le GPC utilisant la méthode de Dichotomie

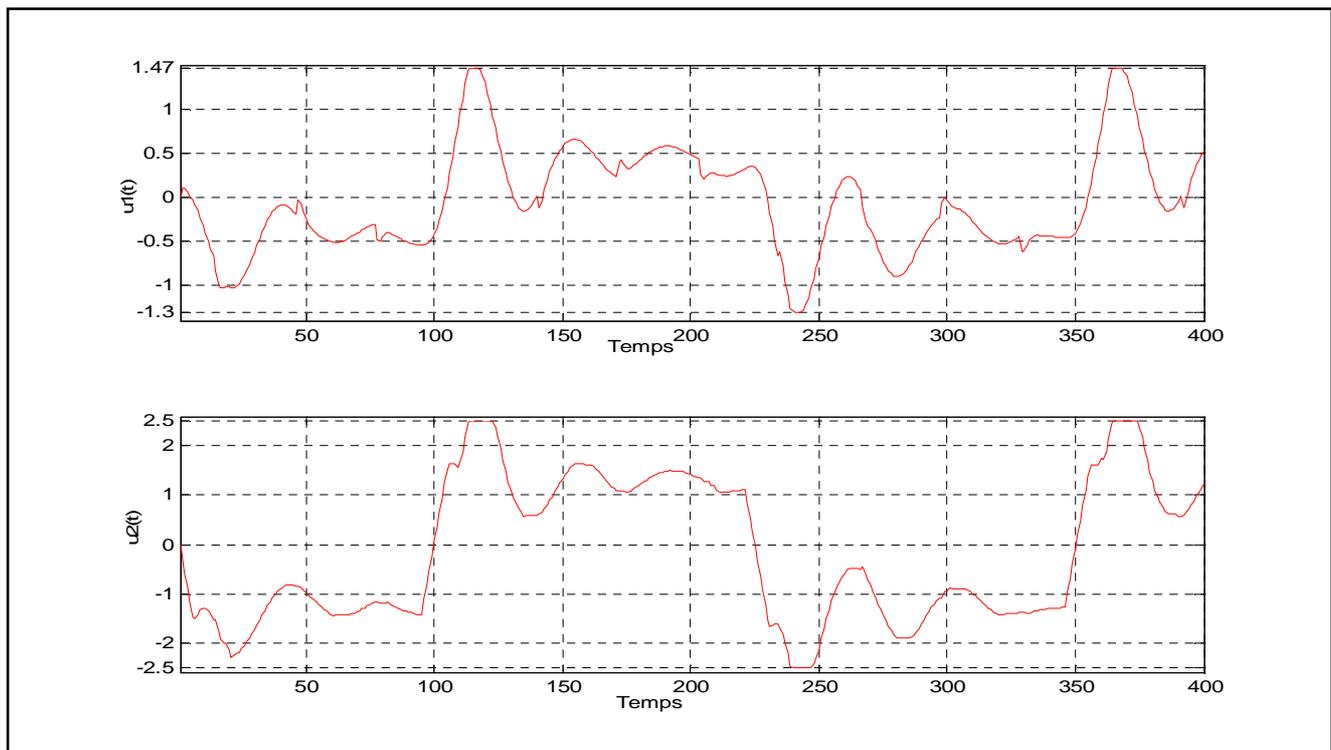


Fig.4.4.b. Commandes fournies par le GPC utilisant la méthode de Dichotomie

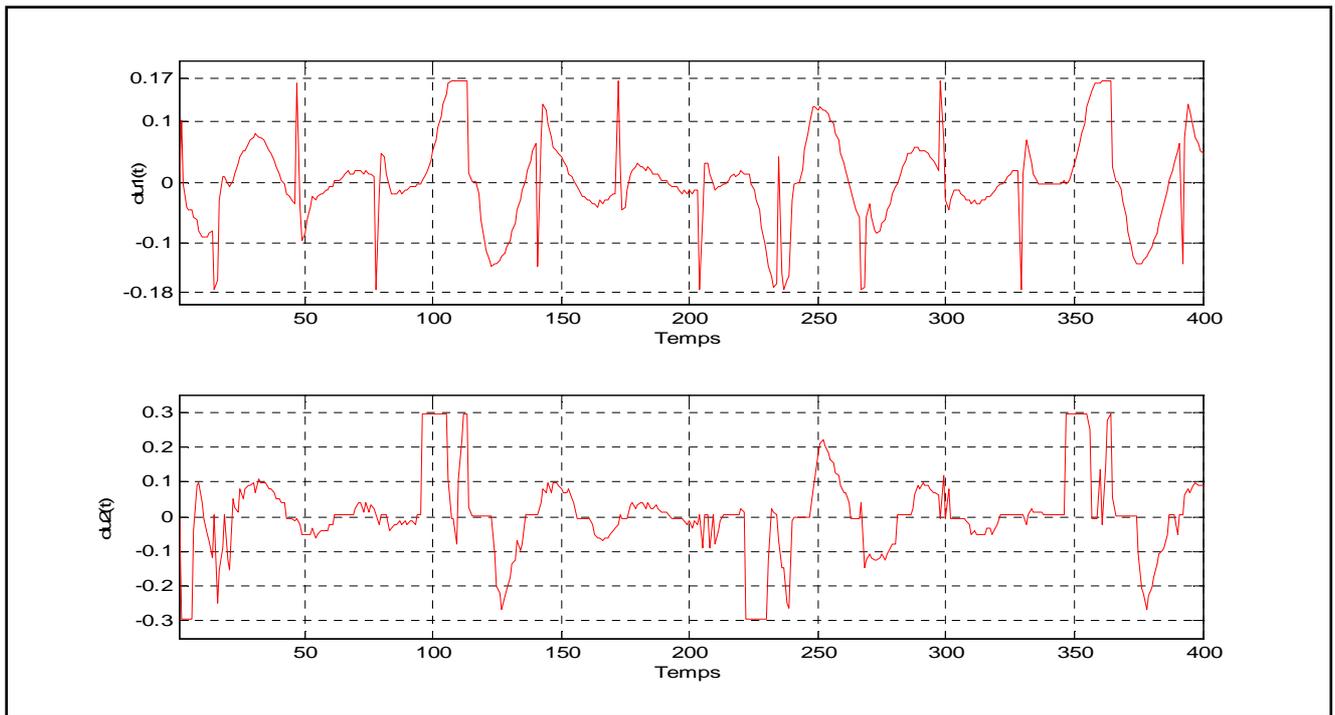


Fig.4.4.c. incréments de commandes fournis par le GPC utilisant la méthode de Dichotomie

- **GPC multivariable utilisant la méthode Tric\_quad :**

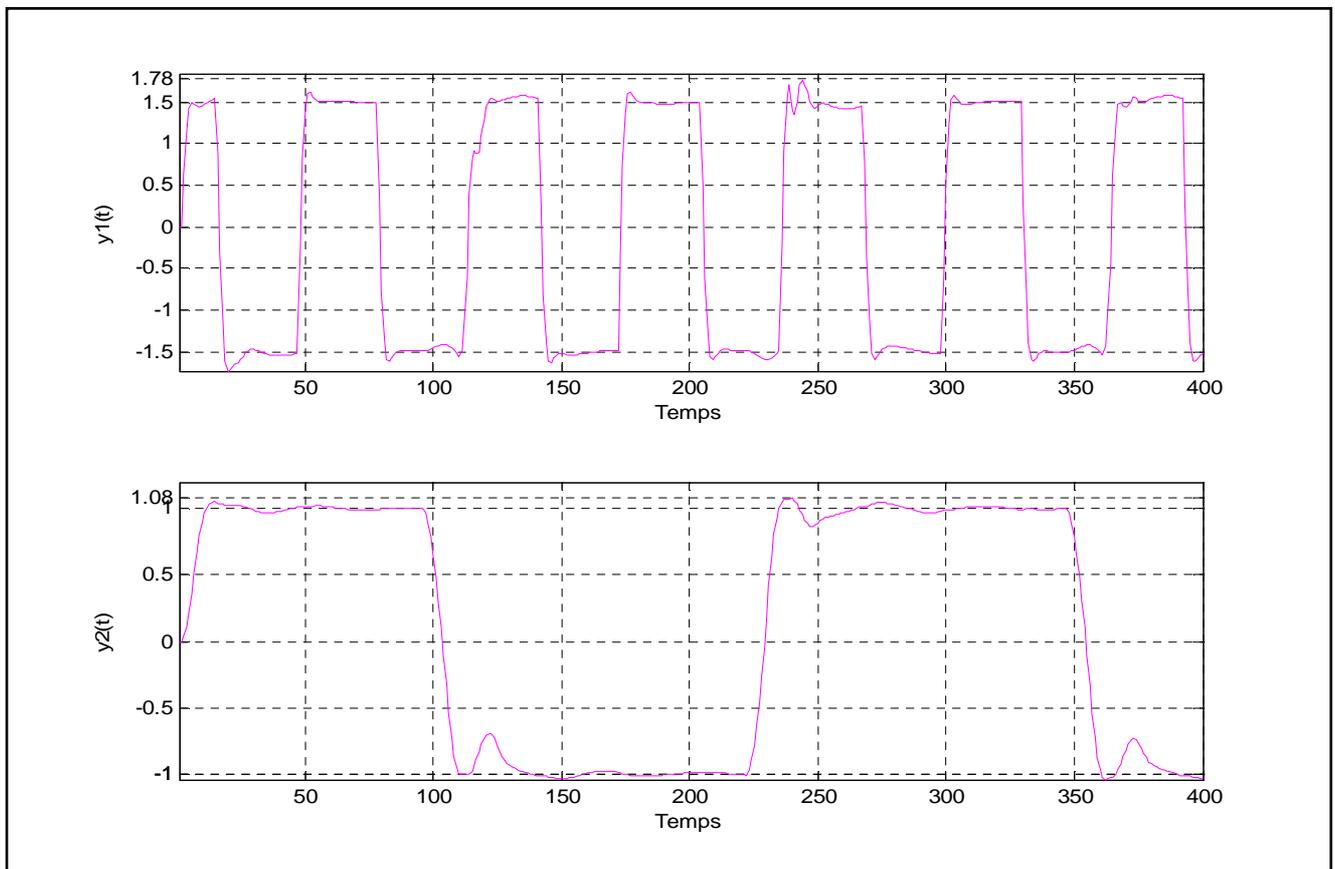


Fig.4.5.a. Sorties obtenues par le GPC utilisant la méthode de Tric\_quad

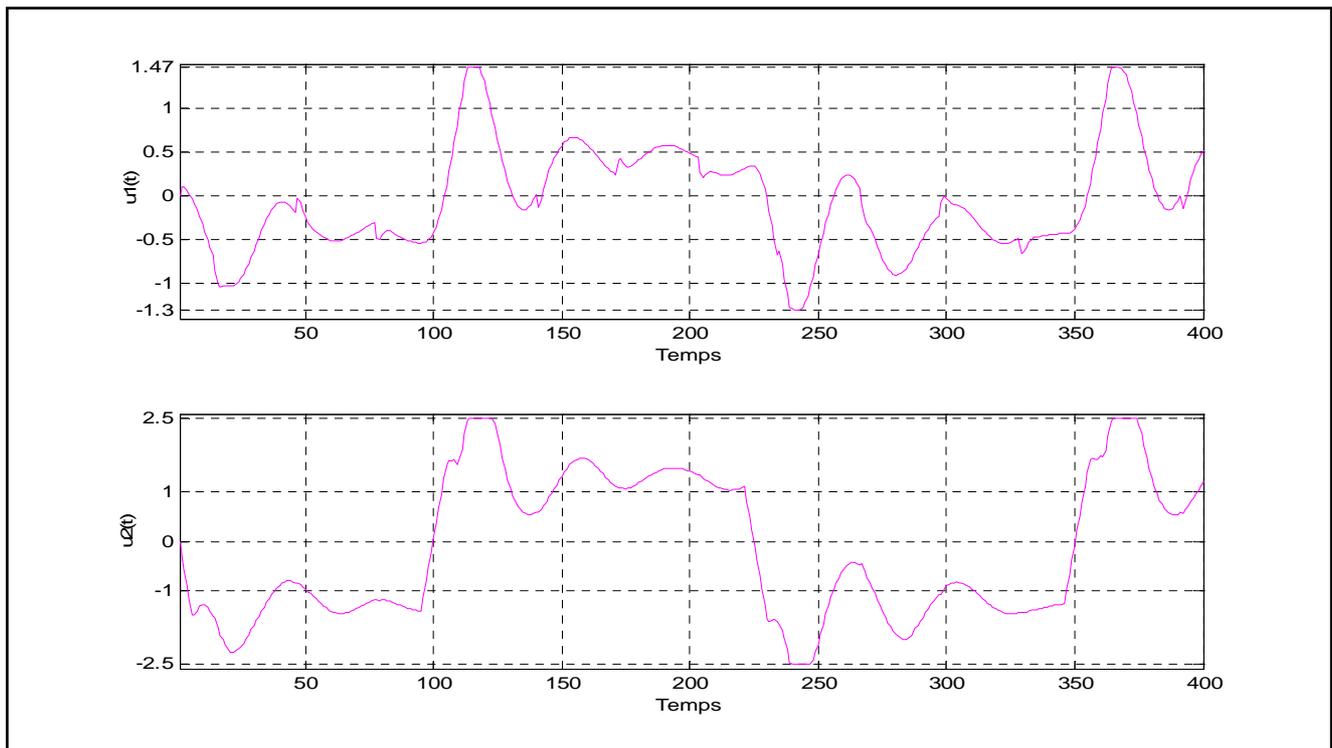


Fig.4.5.b. Commandes fournies par le GPC utilisant la méthode de Tric\_quad

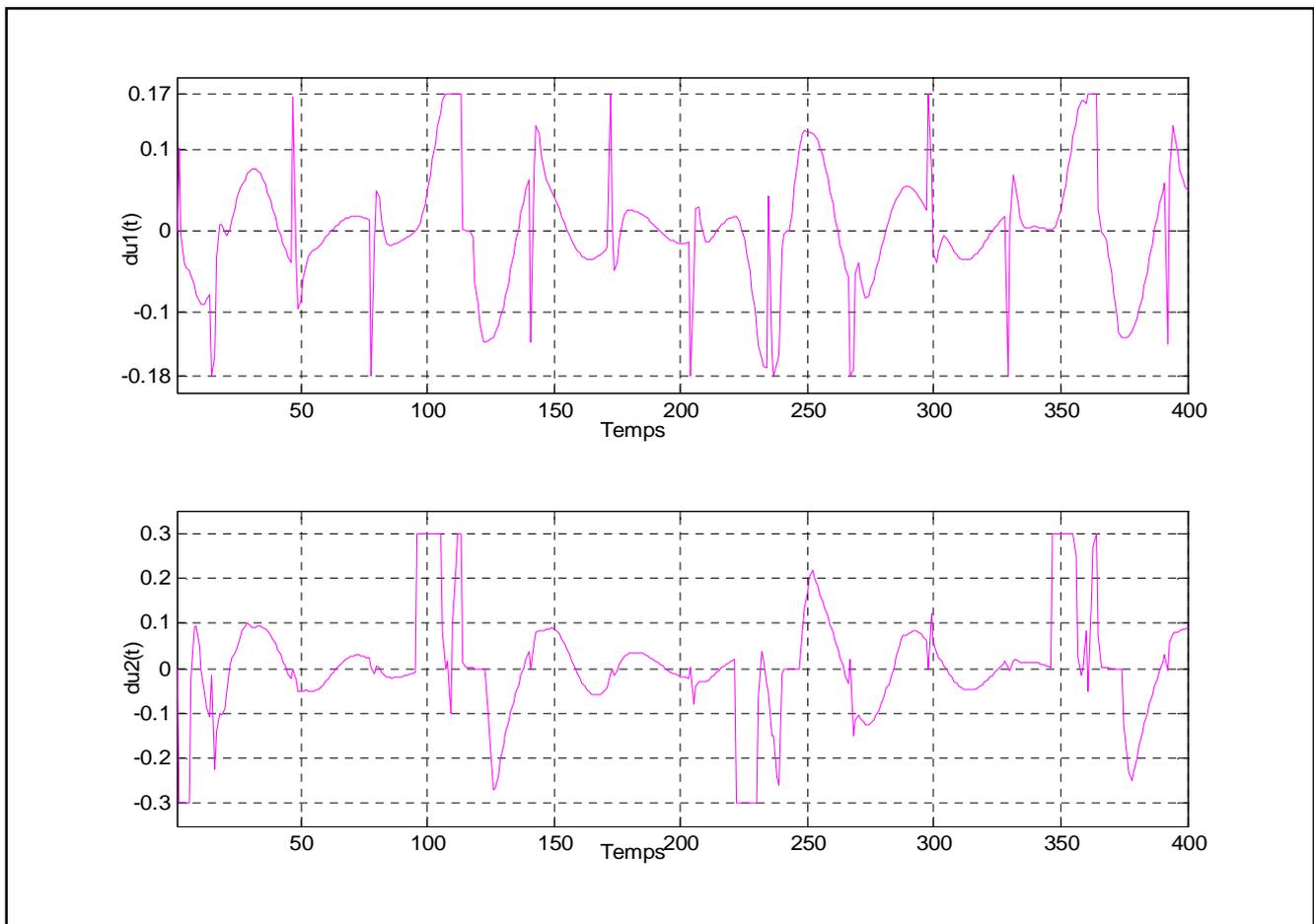


Fig.4.5.c. Incréments de commandes fournis par le GPC utilisant la méthode de Tric\_quad

- GPC multivariable utilisant la méthode Dich\_incr :

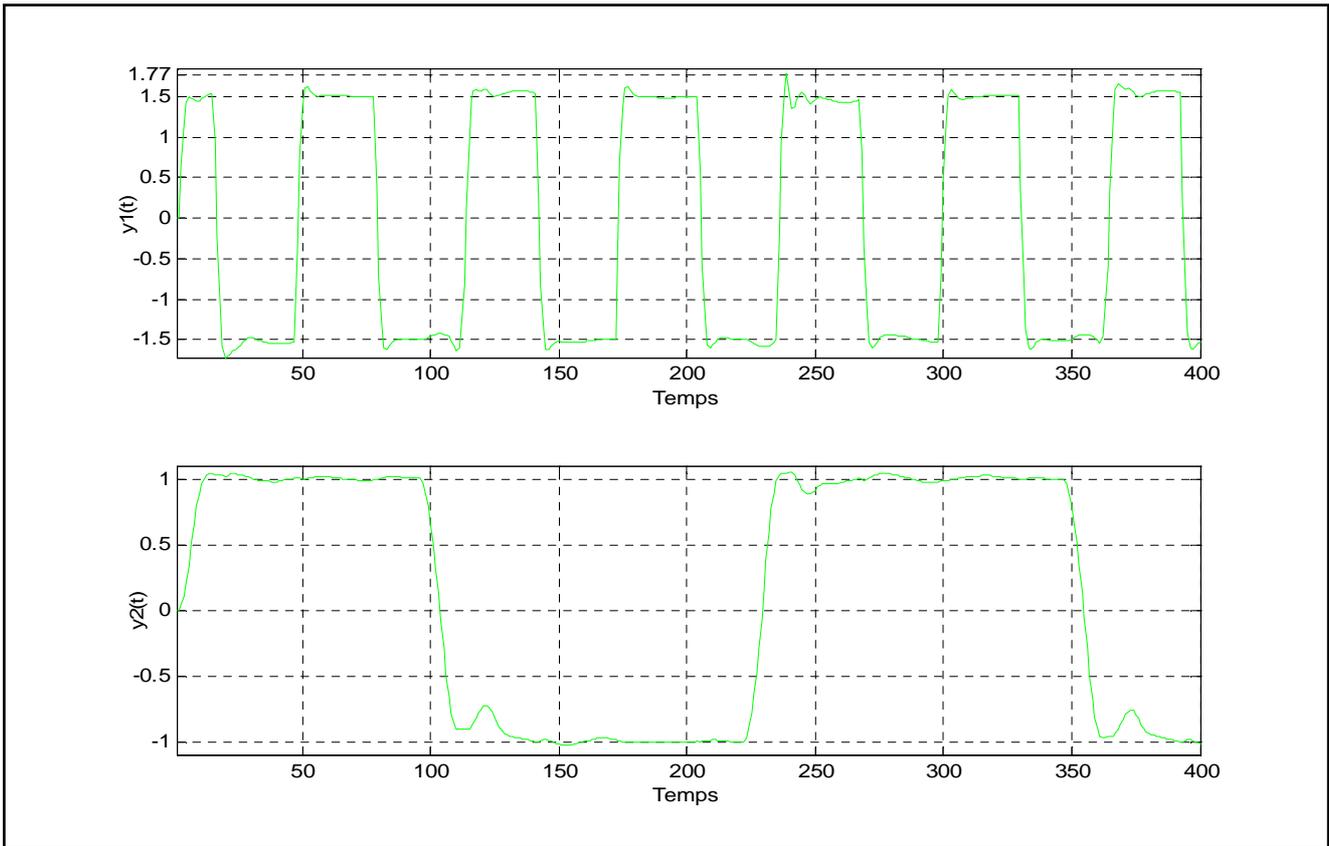


Fig.4.6.a. Sorties obtenues par le GPC utilisant la méthode de Dich\_incr

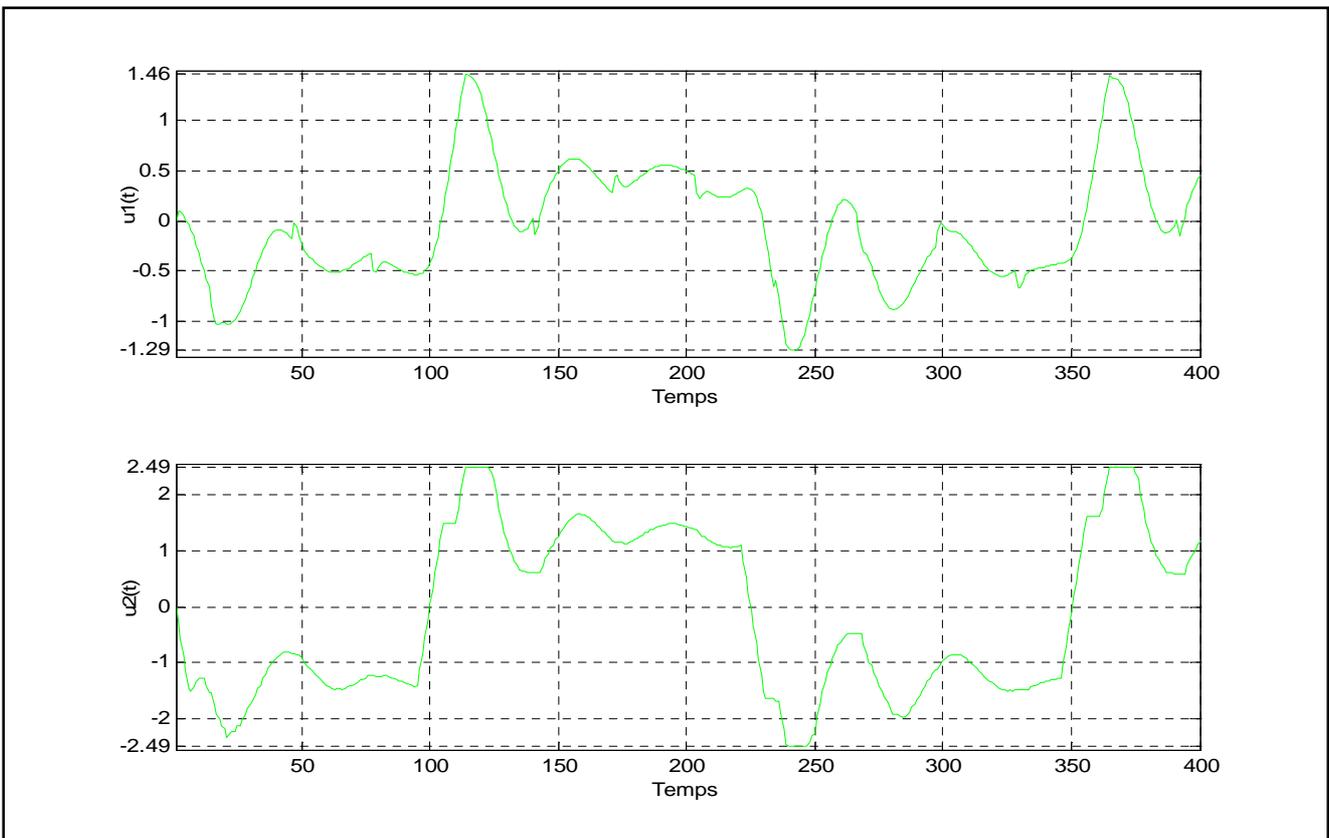


Fig.4.6.b. Commandes fournies par le GPC utilisant la méthode de Dich\_incr

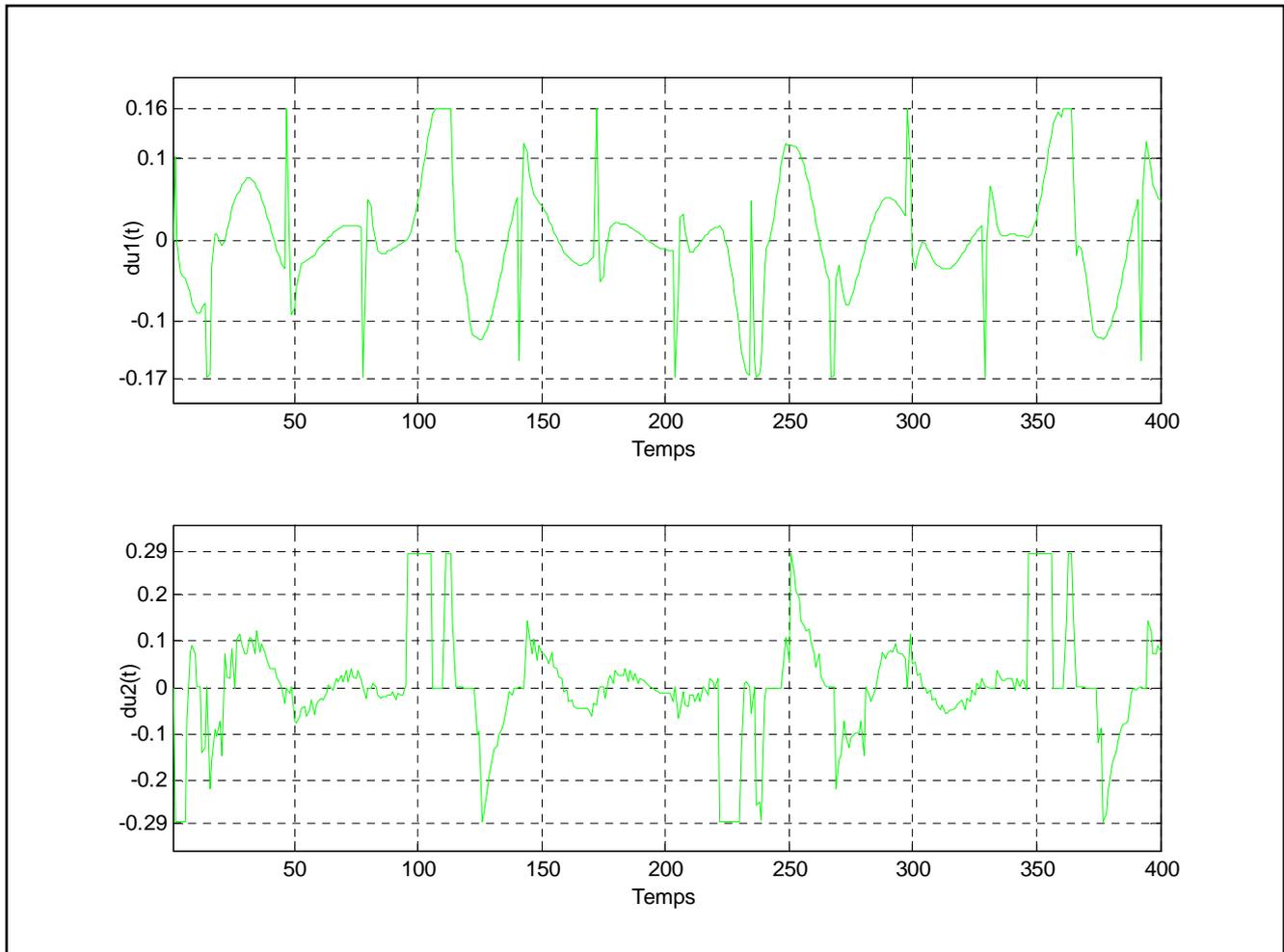


Fig.4.6.c. Incréments de commandes fournis par le GPC utilisant la méthode de Dich\_incr

D'après les résultats présentés par les figures 4.3, 4.4, 4.5 et 4.6, on peut constater que toutes les contraintes considérées sont satisfaisantes par les quatre méthodes proposées, celles-ci pour les deux voies du système.

La méthode des gradients conjugués offre, dans les deux voies, d'une meilleure dynamique de poursuite par rapport aux autres méthodes proposées, offre également, dans la deuxième voie, d'une meilleure réduction de l'erreur de poursuite par rapport à la fonction *quadprog*.

Un inconvénient considérable de cette méthode c'est que la sortie de la première voie présente un pic non souhaité ceci dégrade les performances de cette méthode ainsi de limiter leur application.

Pour les trois méthodes restantes, on obtient presque les mêmes résultats dont on peut perfectionner la méthode Dich\_incr. Cette dernière nécessite, pour le calcul de la commande, moins des limites maximales et minimales de commandes et des incréments de commandes par rapport aux autres méthodes.

#### 4.4.2. Application sur une colonne de distillation :

Cette partie envisage l'application des structures de commande prédictive sous contraintes développées lors des chapitres précédents à un système chimique. Le problème choisi consiste à commander une colonne de distillation binaire mal conditionnée (ill conditioned). L'approche se sert de modèle existant pour aboutir à la construction de loi de type GPC de faible complexité.

##### 4.4.2.1. Description du procédé :

La distillation est une opération de base dans les industries pétrochimiques et pharmaceutiques, c'est un procédé qui permet de séparer les constituants d'un mélange liquide en se basant sur la différence de leurs points d'ébullition. La colonne de distillation étudiée consiste en un courant de vapeur liquide dans un but d'effectuer un transfert de masse entre les deux phases. Le contacte des deux phases est généralement effectué dans une série de plateaux. Il y a une quantité de liquide sur chaque plateau, la vapeur monte en passant à travers le liquide. Le mélange à séparer est introduit dans la colonne, le résidu est chauffé au niveau du rebouilleur situé au bas de la colonne jusqu'à l'évaporation. La vapeur condensée en tête de colonne peut être évacuée comme distillat ou retournée en partie dans la colonne comme reflux. Le modèle du procédé utilise ainsi les performances de conception de la commande qui ont été proposées par [Lime90].

Le modèle mathématique peut se présenter par le schéma bloc ci-après:

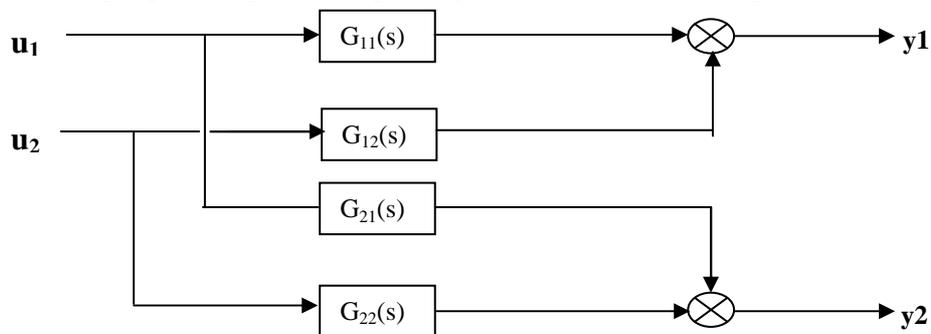


Fig.4.7 : Schéma bloc du système

Avec :

- $G_{11}(s) = \frac{0.878K_1 e^{-\theta_1}}{75s + 1}$  ;  $G_{12}(s) = \frac{-0.864K_2 e^{-\theta_2}}{75s + 1}$
- $G_{21}(s) = \frac{1.082K_1 e^{-\theta_1}}{75s + 1}$  ;  $G_{22}(s) = \frac{-1.096K_2 e^{-\theta_2}}{75s + 1}$  ;
- $K_i \in [0.8 \ 1.2]$ ; and  $\theta_i \in [0.0 \ 1.0]$

D'où, les  $(G_{ij}(s))_{(i=1,2; j=1,2)}$  représentent les fonctions de transfert et forment ainsi la matrice de transfert :

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \text{ Avec : } G(s) = \frac{1}{75s+1} \begin{bmatrix} 0.878 & -0.864 \\ 1.082 & -1.096 \end{bmatrix} \begin{bmatrix} K_1 e^{-\theta_1} & 0 \\ 0 & K_2 e^{-\theta_2} \end{bmatrix}$$

Physiquement, ceci équivalent à des gains incertains de 20% et un retard de plus d'une minute sur chaque direction.

Ce procédé possédant deux entrées qui sont le flux et le débit de vapeur notées par  $u = [u_1 \quad u_2]^T$ , et deux sorties qui représentent les deux compositions, distillat et résidu notées par  $y = [y_1(t) \quad y_2(t)]^T$

Pour la simulation, on a calculé le modèle discret avec un bloqueur d'ordre zéros et une période d'échantillonnage égale à 1 minute.

Pour notre système, nous devons remplir les spécifications [Lime90]:

1- voir une stabilité en boucle fermée.

2- Pour le couple d'entrées de consigne  $w = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ , la sortie de poursuite  $y_1(t)$  et la sortie

d'interaction  $y_2(t)$  doivent satisfaire :

a- Pour :  $t \geq 30 \text{ min} \Rightarrow y_1(t) \geq 0.9$ .

b-  $\forall t \geq 0 \Rightarrow y_1(t) \leq 1.1$  : il faut voir un dépassement maximal de 10% sur la première voie.

c-  $0.99 \leq y_1(t) \leq 1.1$  dans le régime permanent.

d-  $\forall t \geq 0 \Rightarrow y_2(t) \leq 0.5$ .

e-  $-0.01 \leq y_2(t) \leq 0.01$  dans le régime permanent.

f- Les commandes sont limitées à 200

3- Pour le couple d'entrées de consigne  $w = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ , la sortie de poursuite  $y_2(t)$  et la sortie

d'interaction  $y_1(t)$  doivent satisfaire les mêmes spécifications a), b), c), d), e) et f).

4- Les contraintes sont choisissés comme suit:

$$-12 \leq \Delta u_1(t) \leq 12$$

$$-12 \leq \Delta u_2(t) \leq 12$$

#### 4.4.2.2. Application de GPC utilisant les méthodes d'optimisation proposées :

À la longue de la boucle du temps  $t$ , nous devons, tout d'abord, déterminer la région de faisabilité à partir de ces contraintes ci-dessus. En fixant la précision  $\zeta$  ( $\zeta = 10^{-4}$ ) demandée pour chaque méthode d'optimisation, afin de déterminer le nombre d'itération  $n$  pour chaque méthode.

Pour les paramètres de GPC, on choisit :

|        | $N_2$ | $N_u$ | $\lambda$ |
|--------|-------|-------|-----------|
| Voie 1 | 7     | 2     | 0.00001   |
| Voie 2 | 6     | 2     | 0.00001   |

Pour le cas nominal où  $K_1 = K_2 = 1$  ainsi pour le cas de  $K_1 = K_2 = 1.2$ , on applique au ce système deux entrées de consignes, la première représente l'entrée de poursuite  $w_1 = 1$  et la seconde représente l'entrée d'interaction  $w_2 = 0$ . Maintenant pour le cas  $K_1 = K_2 = 0.8$  et  $K_1 = 1.2, K_2 = 0.8$ , on considère l'inverse du cas précédent c'est-à-dire la première entrée de consigne considérée comme étant d'interaction cependant la seconde, est considérée comme une entrée de poursuite.

- Pour les commandes fournies par la loi de GPC, on remarque qu'elles sont dépendent, en régime statique, par les valeurs des gains incertains  $K_1$  et  $K_2$ . Pour une bonne dynamique de poursuite/ interaction, les commandes du système doivent être tendent vers à des valeurs fournies par la relation suivante :

$$\begin{pmatrix} u_1(s) \\ u_2(s) \end{pmatrix}_{s \rightarrow 0} = G^{-1}(0) \times \begin{pmatrix} w_1(s) \\ w_2(s) \end{pmatrix}_{s \rightarrow 0} \quad \text{d'où on obtient :}$$

1.  $\begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} 39.9417 \\ 39.4315 \end{pmatrix}$ , pour  $K_1 = K_2 = 1$  et  $w = (1 \ 0)^T$ .
2.  $\begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} 33.2847 \\ 32.8596 \end{pmatrix}$ , pour  $K_1 = K_2 = 1.2$  et  $w = (1 \ 0)^T$ .
3.  $\begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} -39.3586 \\ -39.9964 \end{pmatrix}$ , pour  $K_1 = K_2 = 0.8$  et  $w = (0 \ 1)^T$ .
4.  $\begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} -26.2391 \\ -39.9964 \end{pmatrix}$ , pour  $K_1 = K_2 = 0.8$  et  $w = (0 \ 1)^T$ .

a). Cas nominal  $K_1 = K_2 = 1$  :

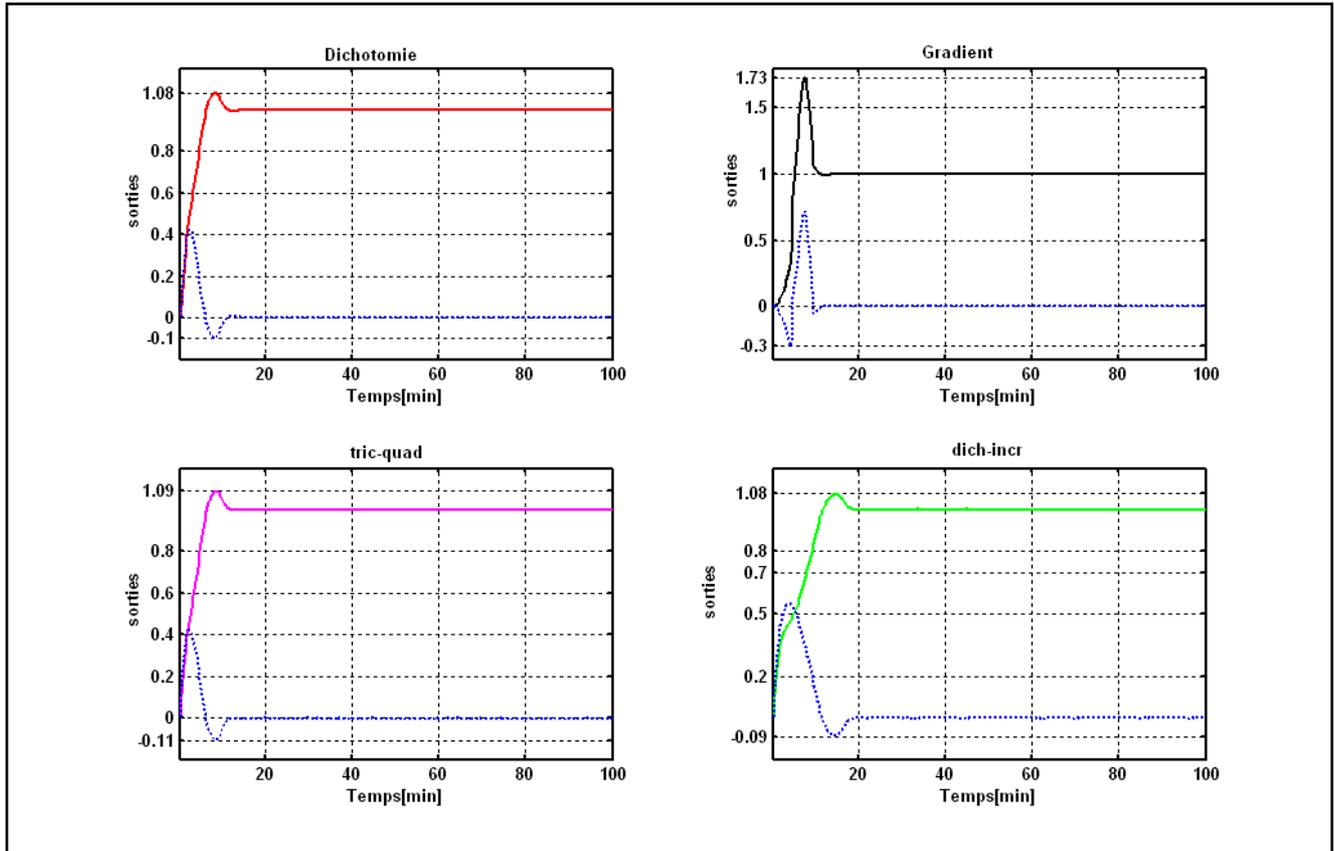


Fig.4.8.1 : Sorties (poursuite/interaction) : cas nominal et ( $w_1 = 1, w_2 = 0$ )

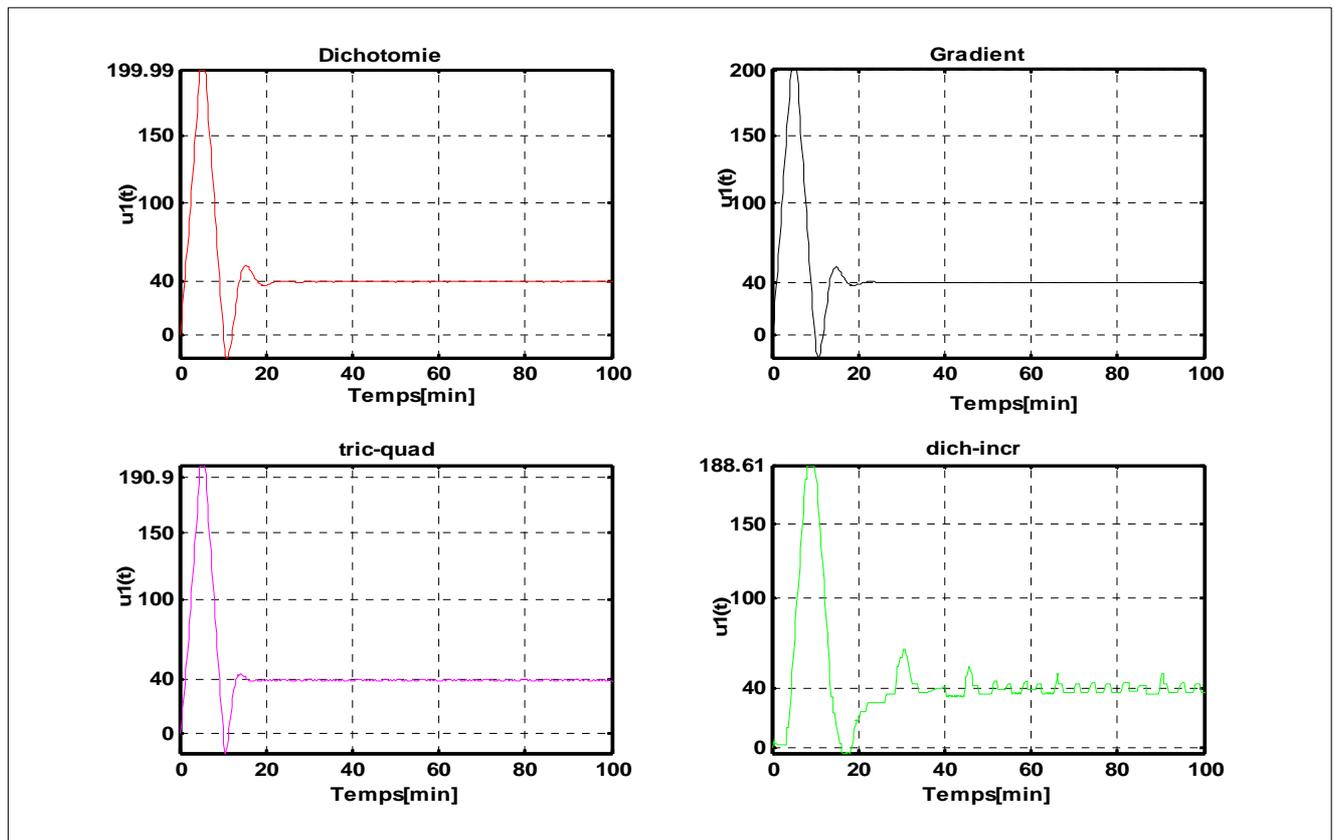


Fig.4.8.2 : Commande de la voie1 : cas nominal et ( $w_1 = 1, w_2 = 0$ )

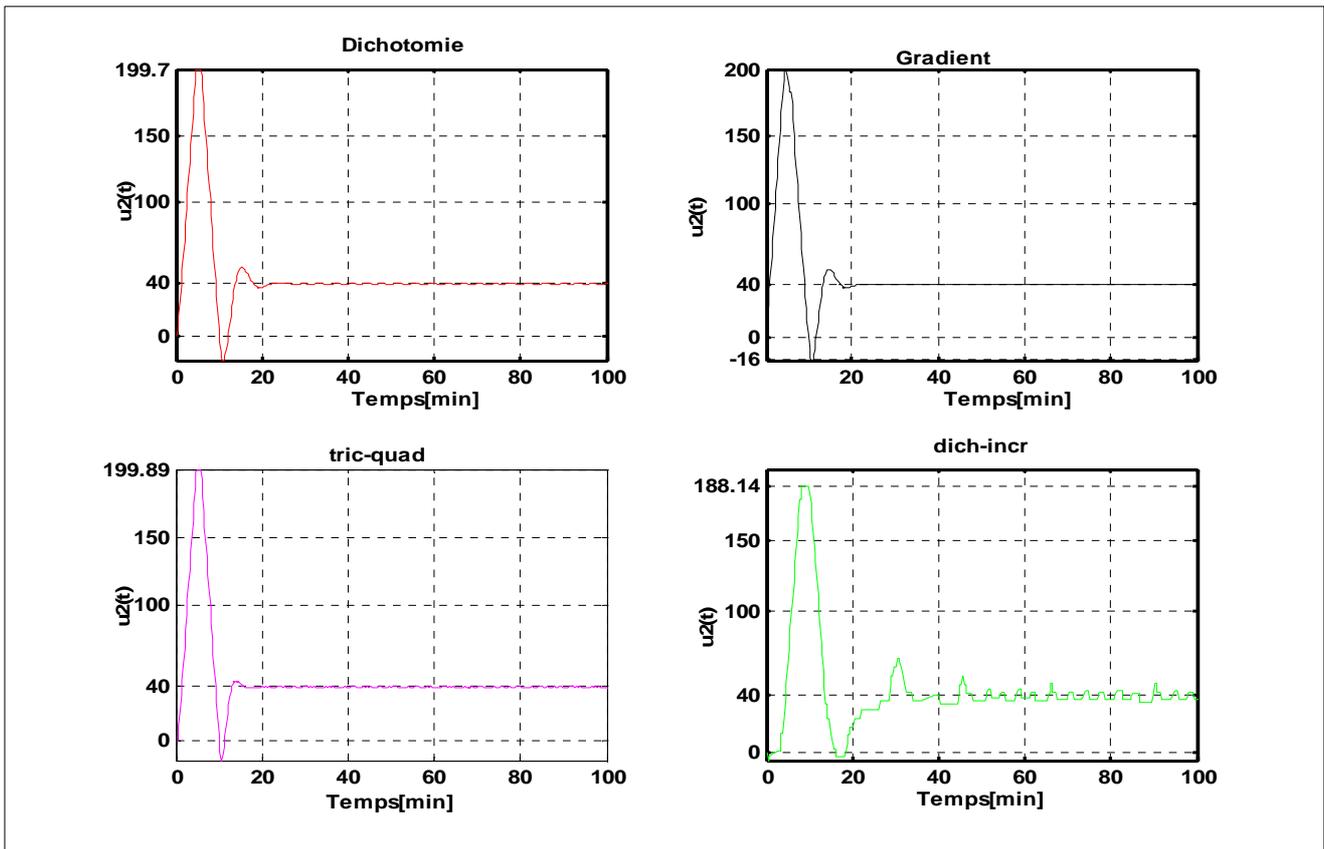


Fig.4.8.3 : Commande de la voie2 : cas nominal et ( $w_1 = 1, w_2 = 0$ )

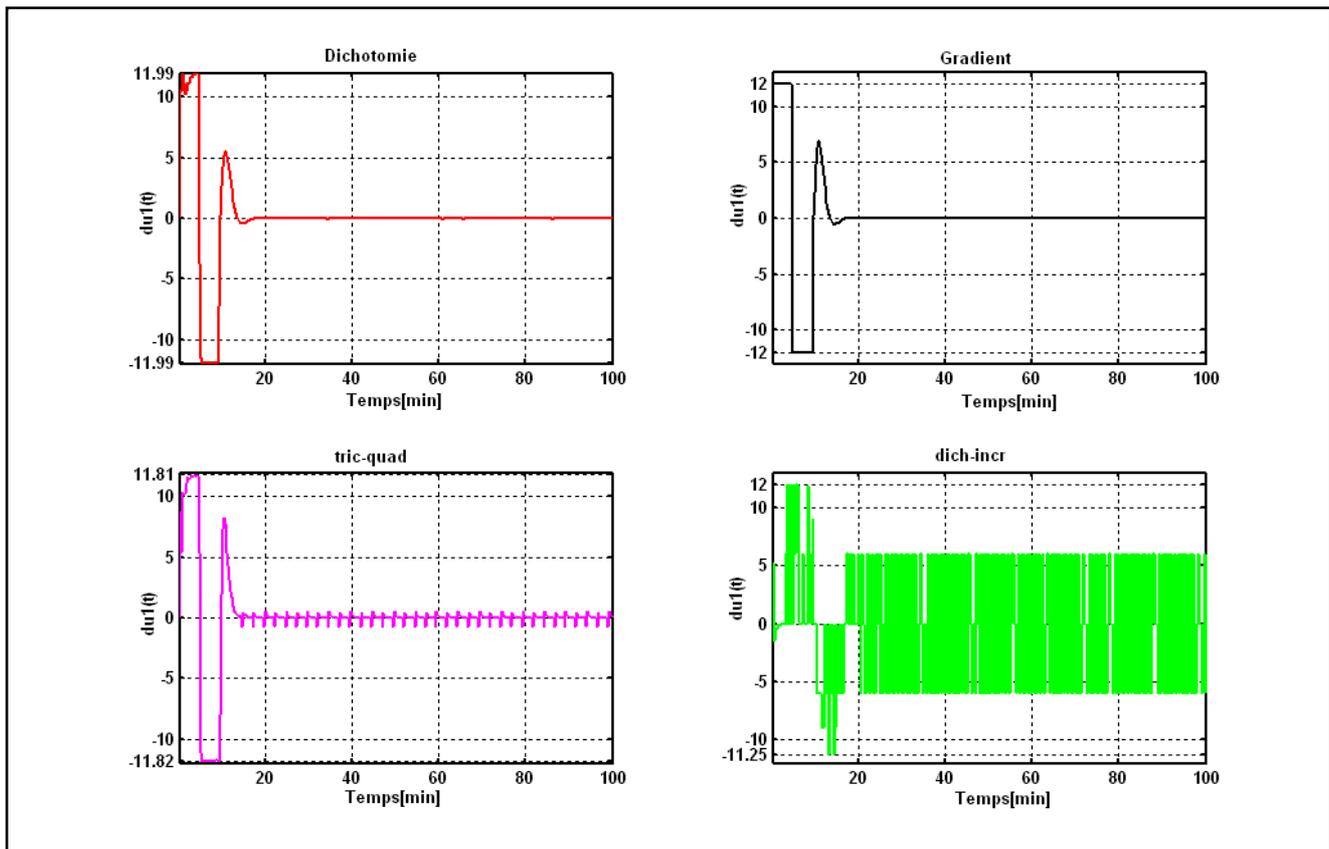


Fig.4.8.4 : Incrément de commande de la voie1 : cas nominal et ( $w_1 = 1, w_2 = 0$ )

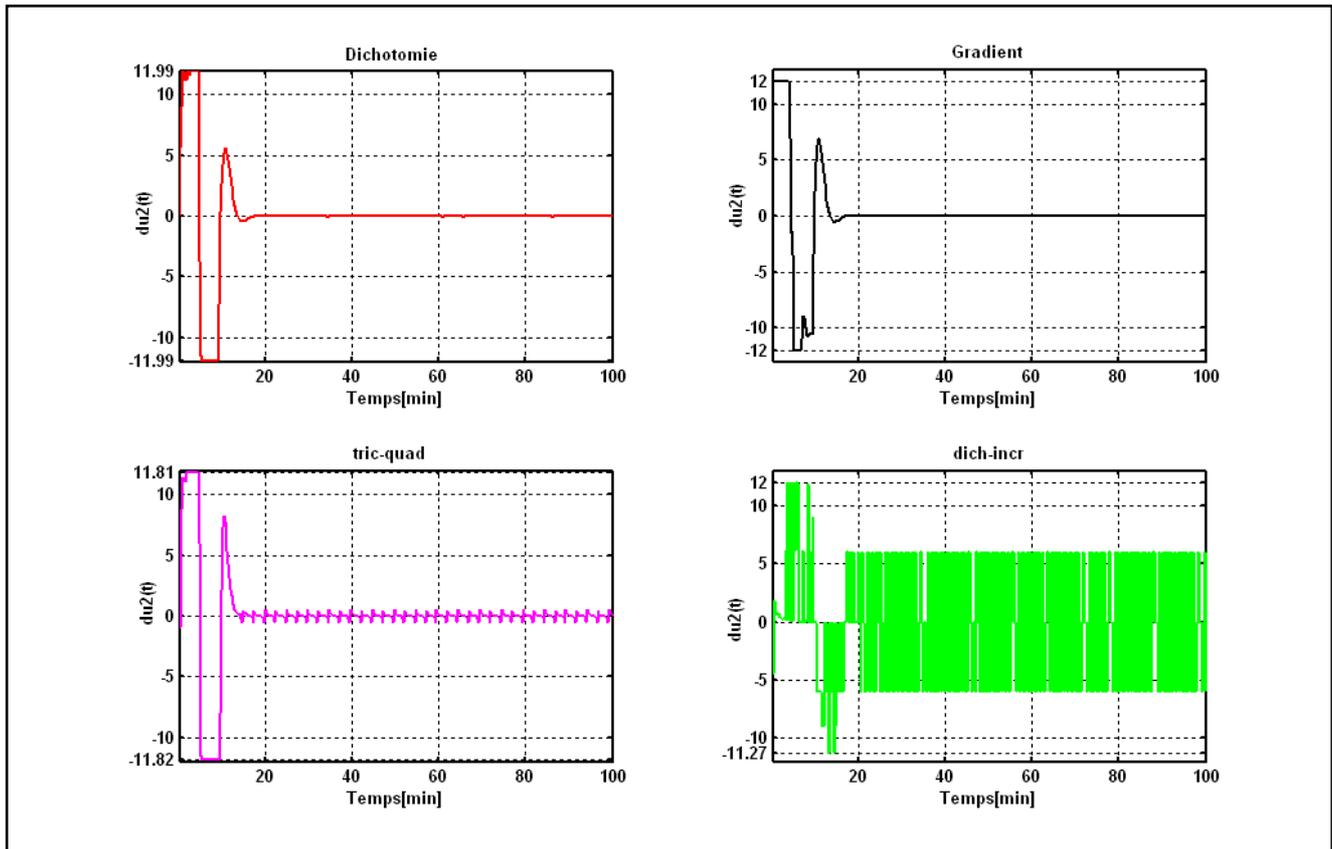


Fig.4.8.5 : Incrément de commande de la voie2 : cas nominal et ( $w_1 = 1, w_2 = 0$ )

**b). Cas de  $K_1 = K_2 = 1.2$**

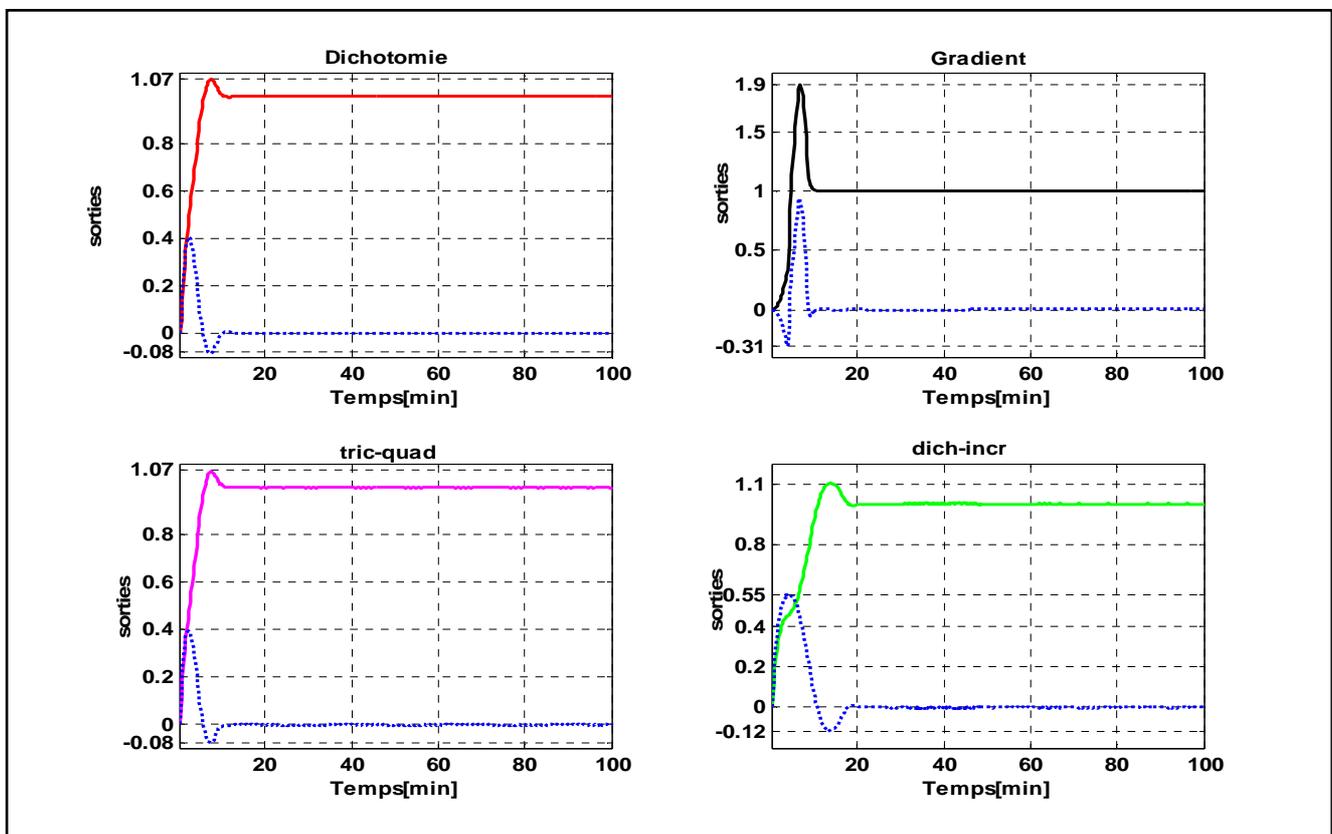


Fig.4.9.1 : Sorties (poursuite/interaction) : cas de ( $K_1 = K_2 = 1.2$ ) et ( $w_1 = 1, w_2 = 0$ ).

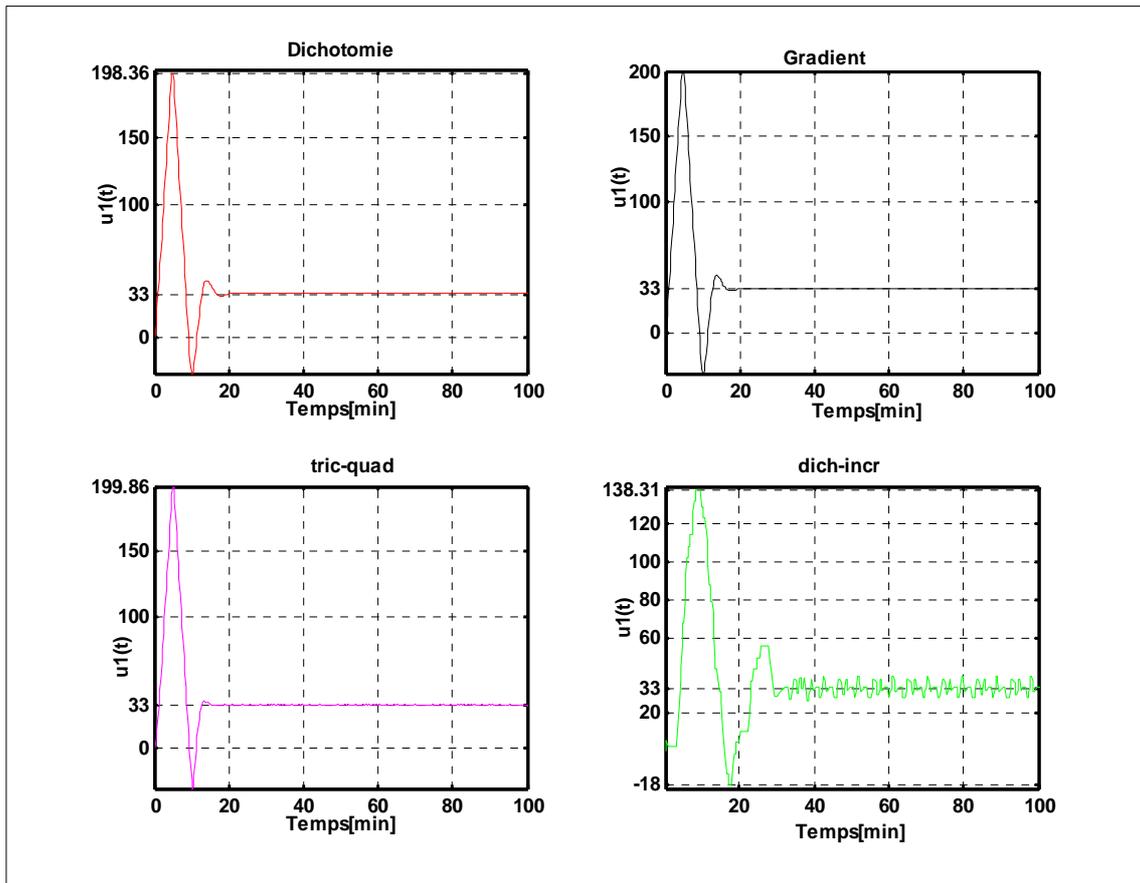


Fig.4.9.3 : Commande de la voie2 : cas de  $(K_1 = K_2 = 1.2)$  et  $(w_1 = 1, w_2 = 0)$ .

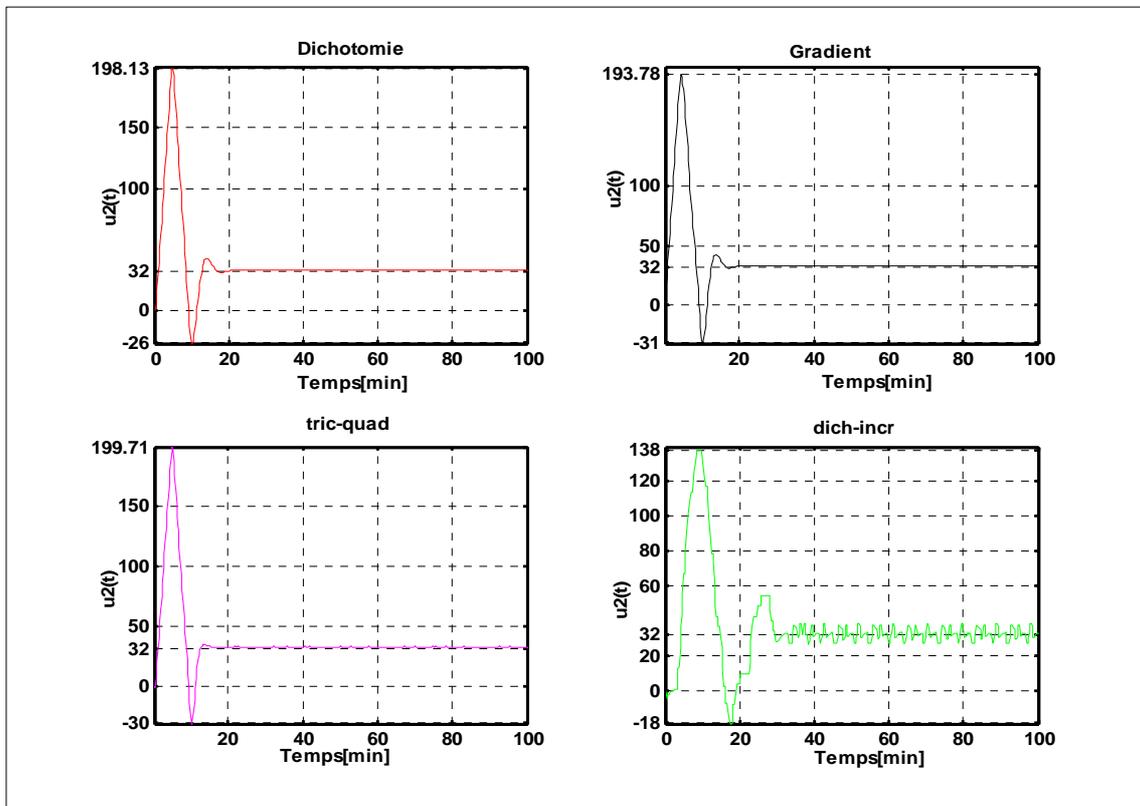


Fig.4.9.3 : Commande de la voie2 : cas de  $(K_1 = K_2 = 1.2)$  et  $(w_1 = 1, w_2 = 0)$ .

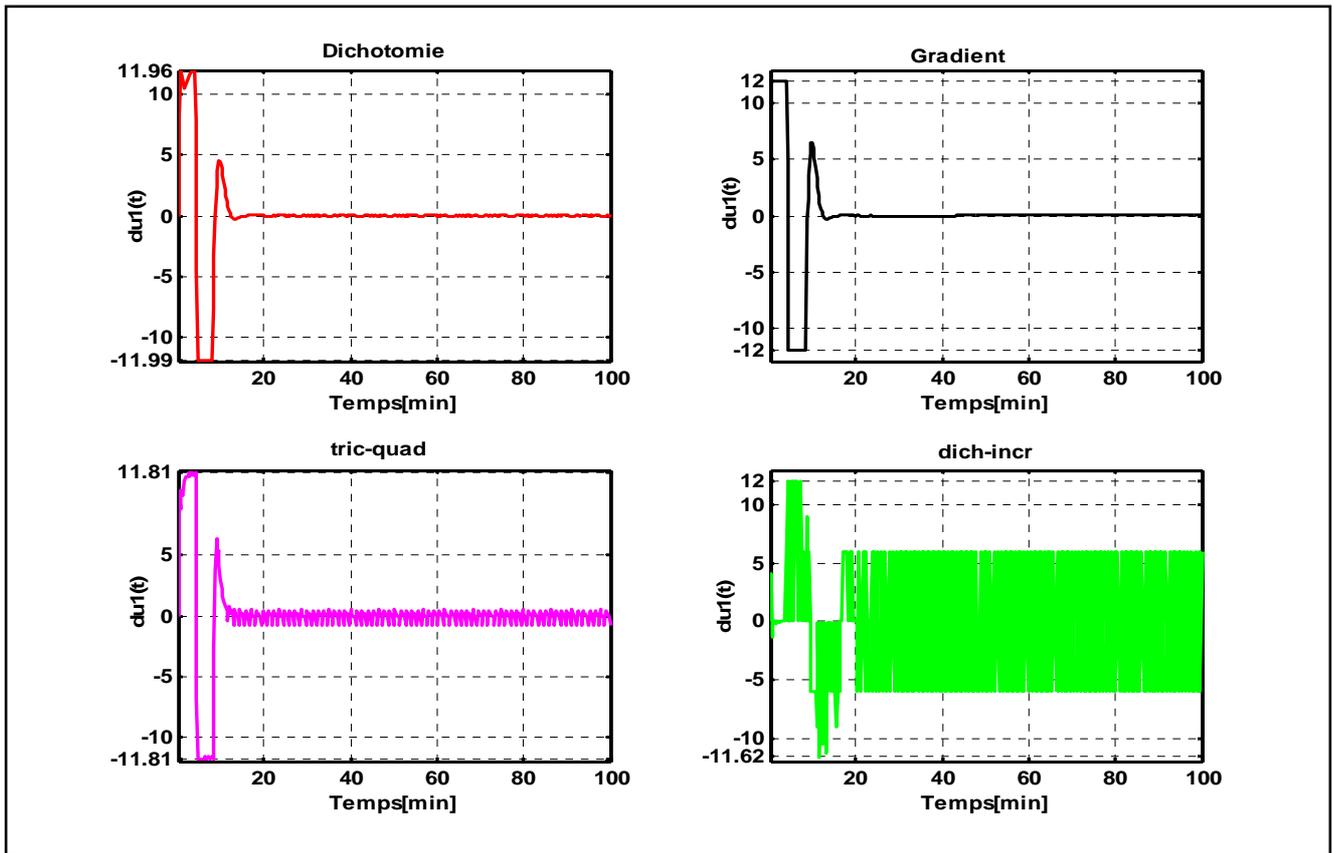


Fig.4.9.4 : Incrément de commande de la voie1 : cas de  $(K_1 = K_2 = 1.2)$  et  $(w_1 = 1, w_2 = 0)$ .

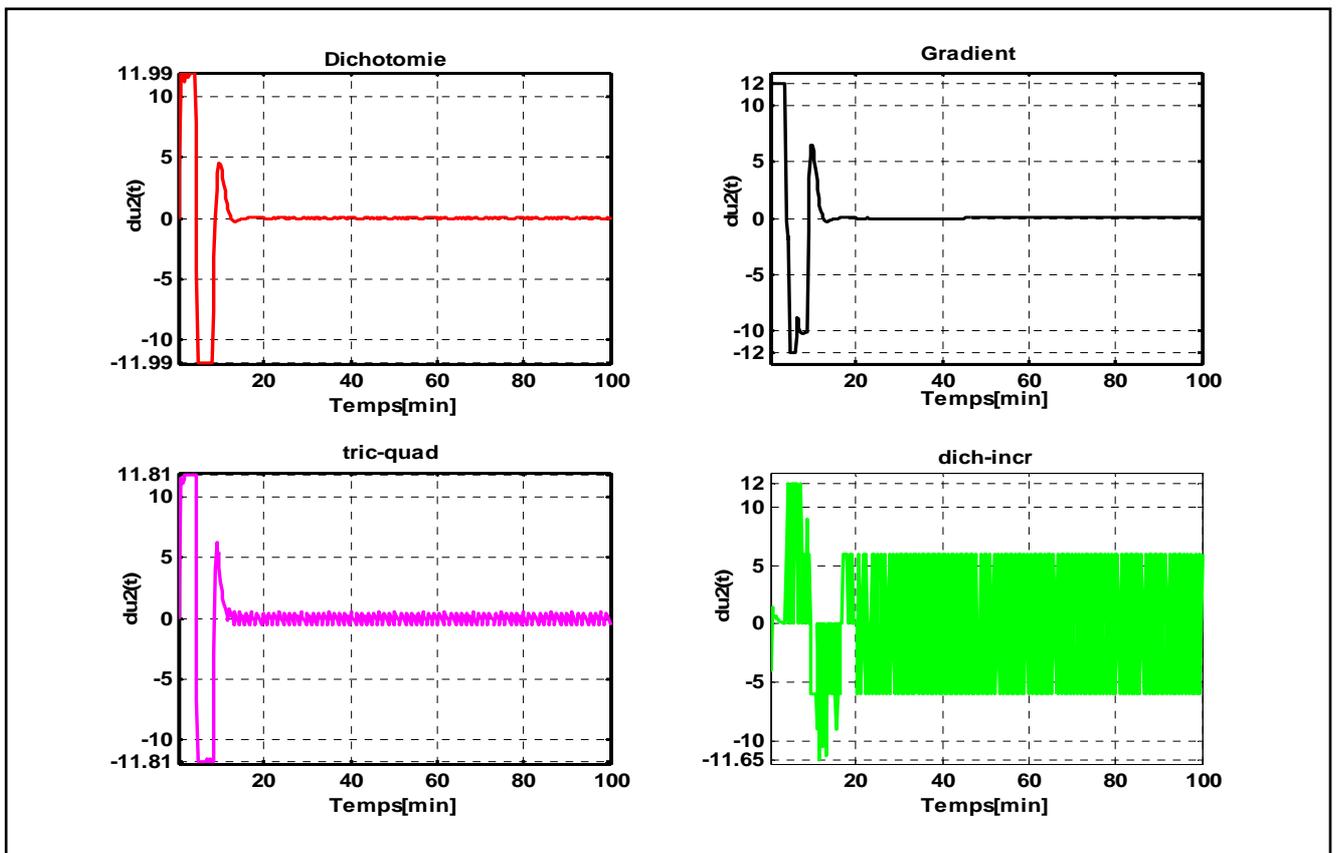


Fig.4.9.5 : Incrément de commande de la voie2 : cas de  $(K_1 = K_2 = 1.2)$  et  $(w_1 = 1, w_2 = 0)$ .

c). Cas de  $K_1 = K_2 = 0.8$  :

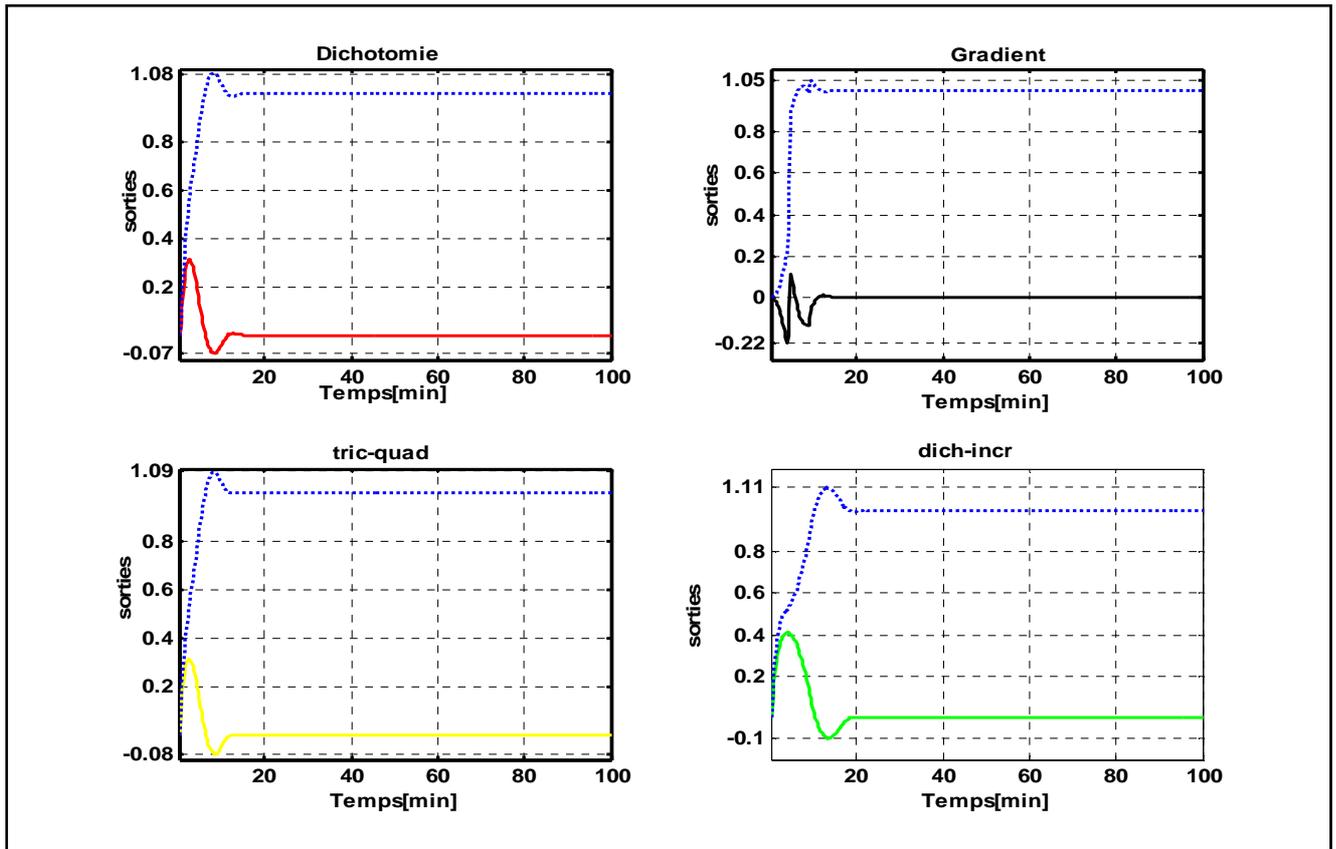


Fig.4.10.1 : Sorties (interaction/poursuite) : cas de ( $K_1 = K_2 = 0.8$ ) et ( $w_1 = 0, w_2 = 1$ ).

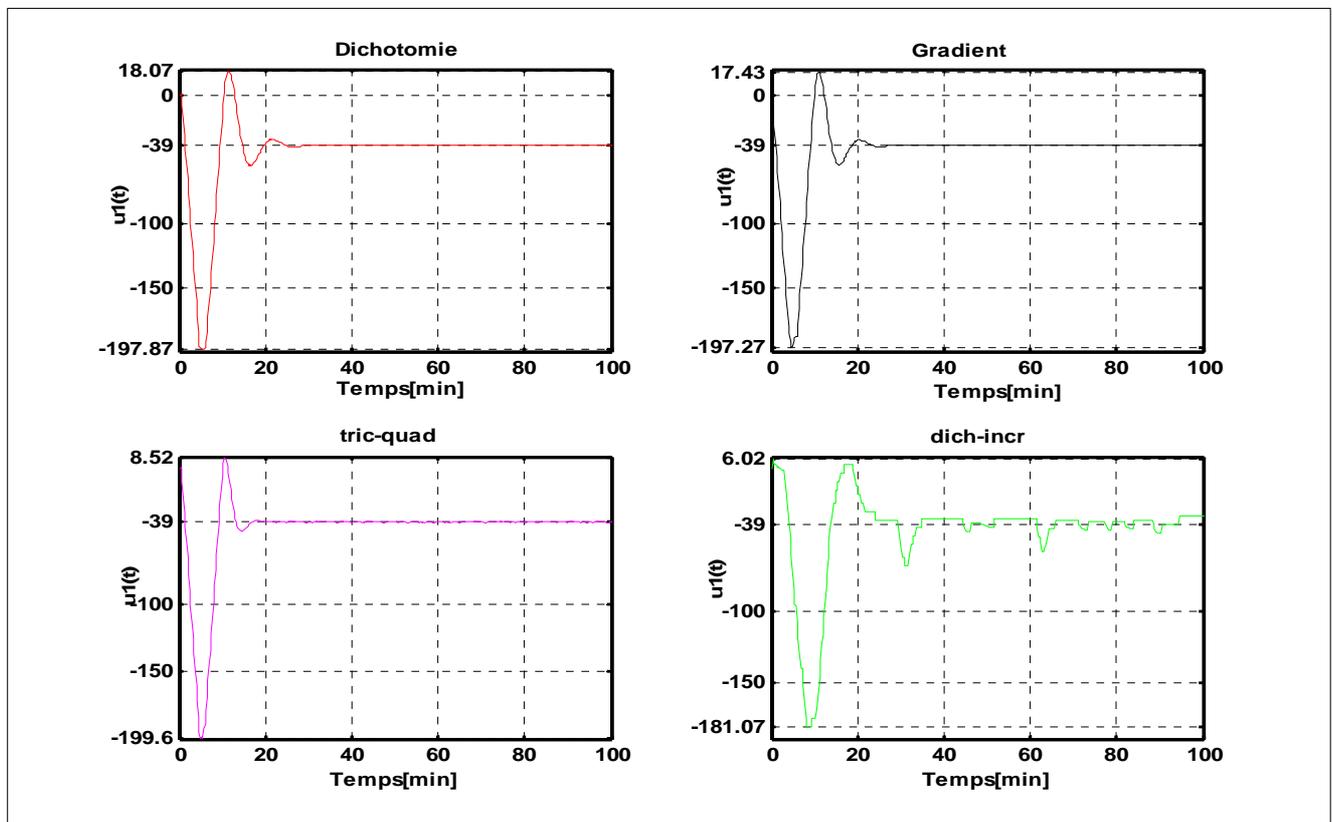


Fig.4.10.2 : Commande de la voie1 : cas de ( $K_1 = K_2 = 0.8$ ) et ( $w_1 = 0, w_2 = 1$ ).

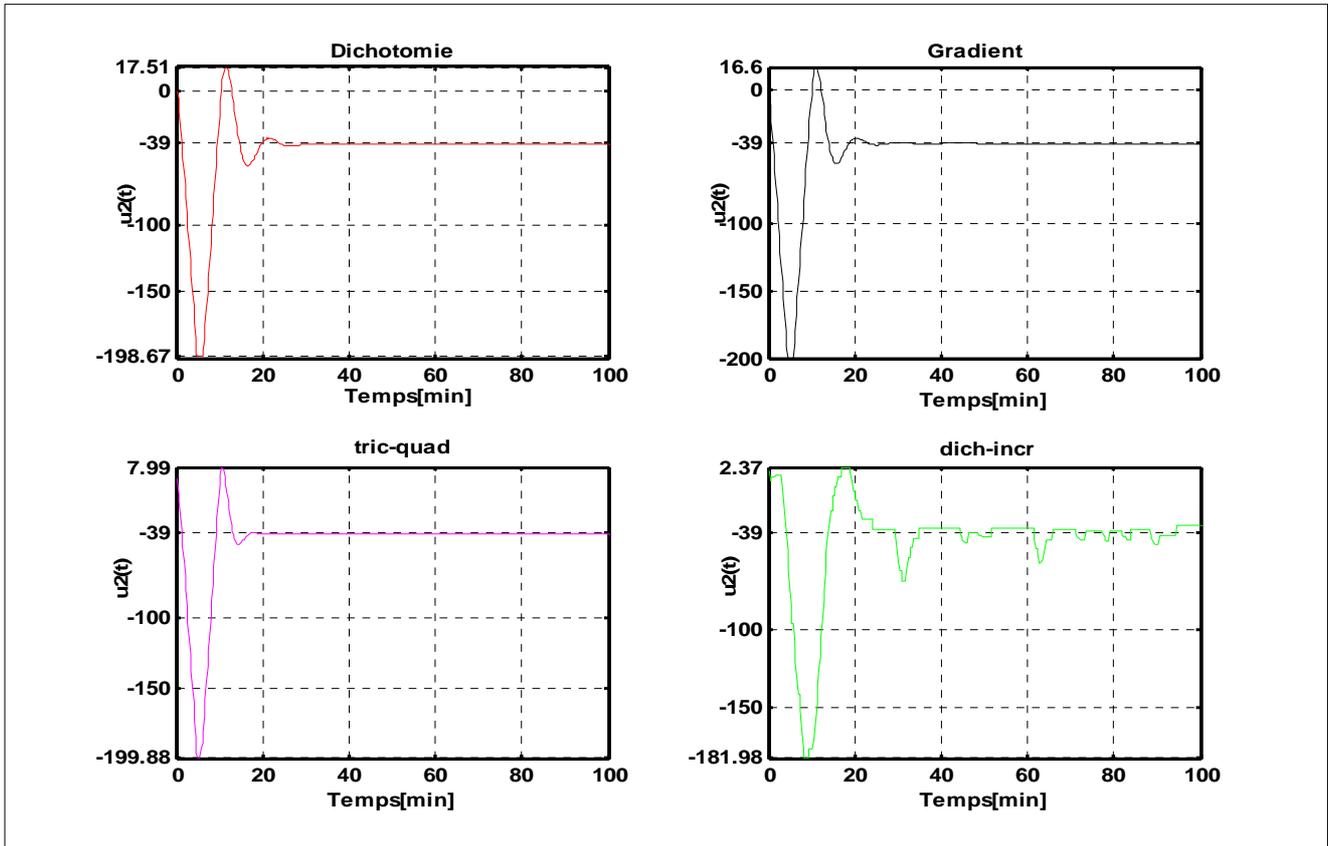


Fig.4.10.3 : Commande de la voie2 : cas de  $(K_1 = K_2 = 0.8)$  et  $(w_1 = 0, w_2 = 1)$ .

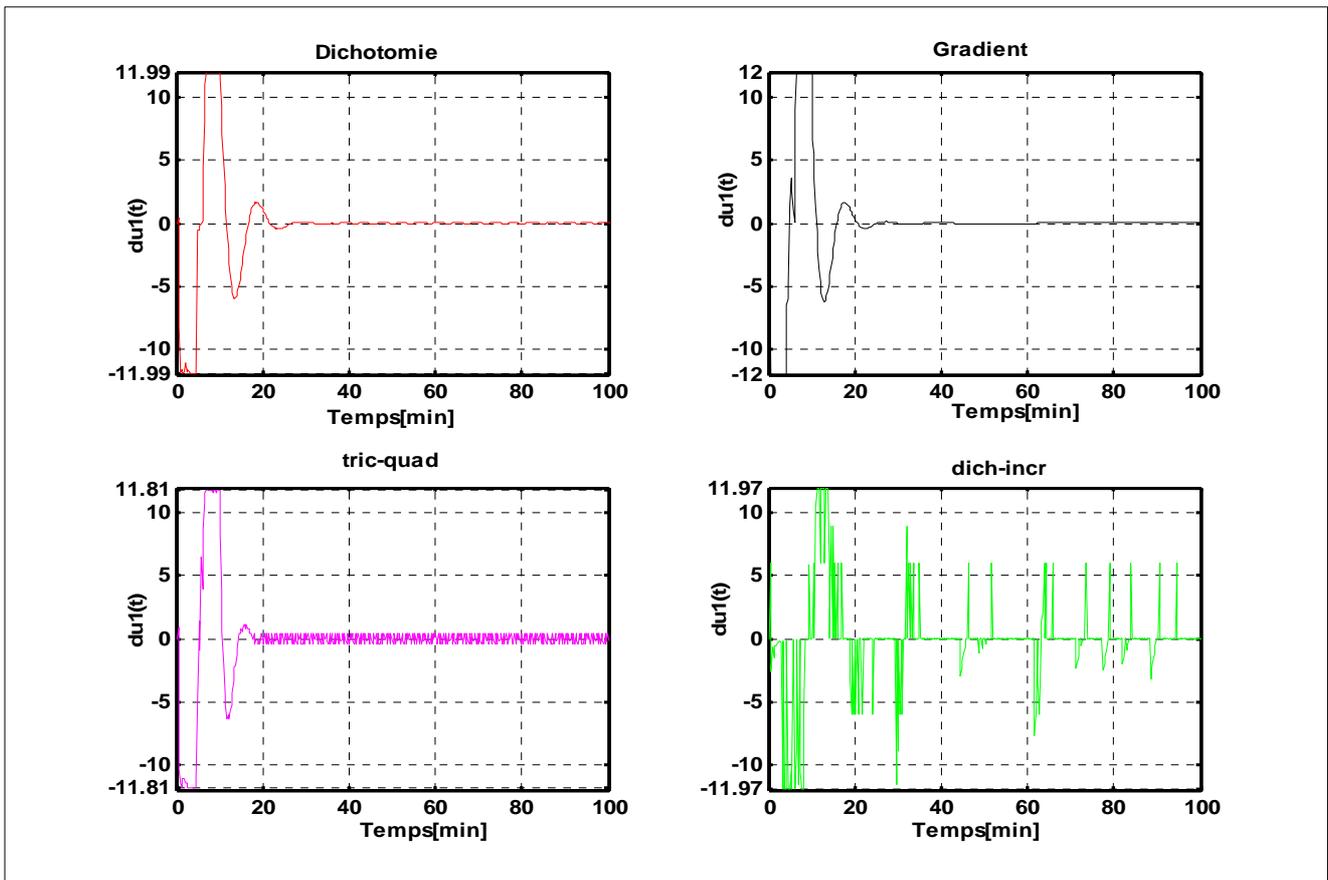


Fig.4.10.4 : Incrément de commande de la voie1 : cas de  $(K_1 = K_2 = 0.8)$  et  $(w_1 = 0, w_2 = 1)$

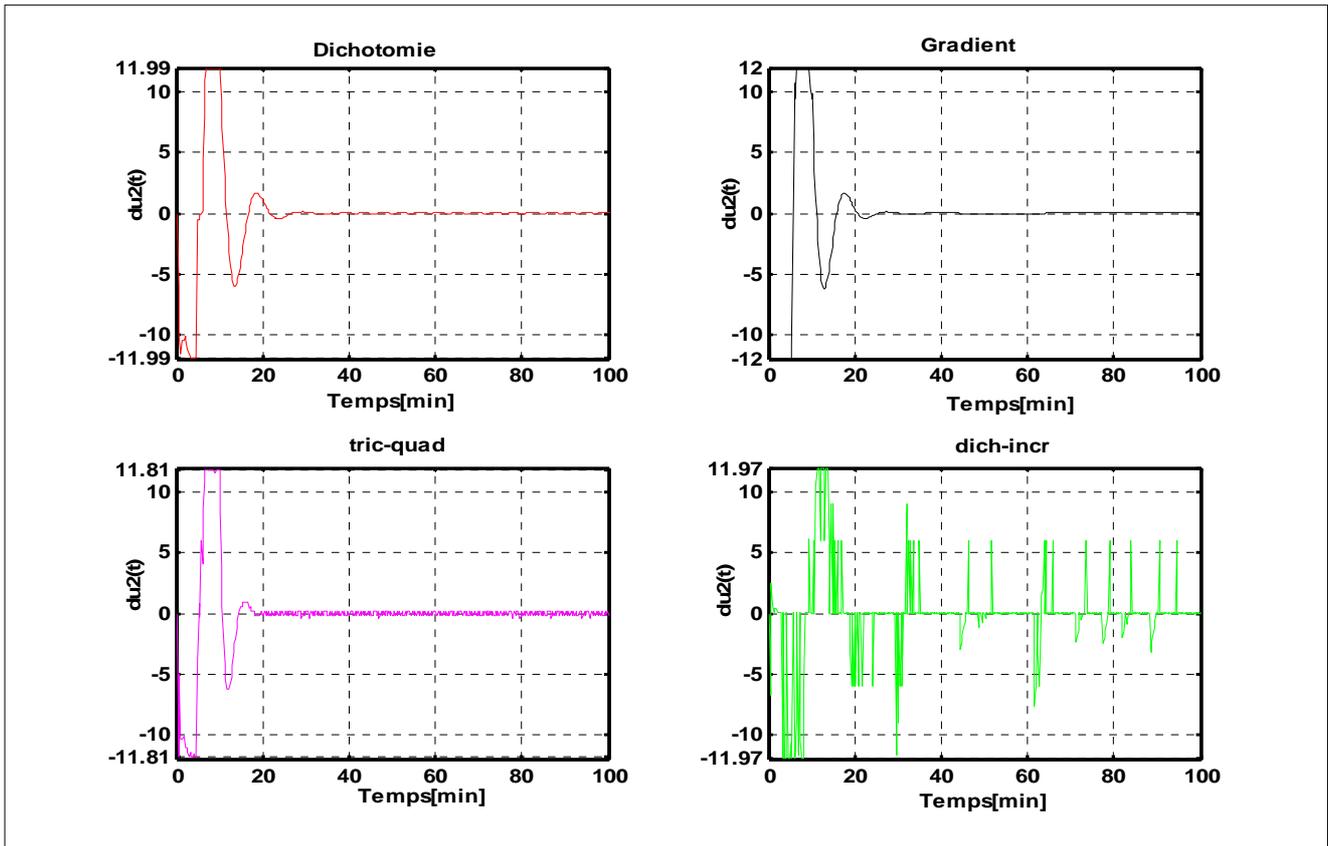


Fig.4.10.5 : Incrément de commande de la voie2 : cas de  $(K_1 = K_2 = 0.8)$  et  $(w_1 = 0, w_2 = 1)$

d). Cas de  $K_1 = 1.2, K_2 = 0.8$

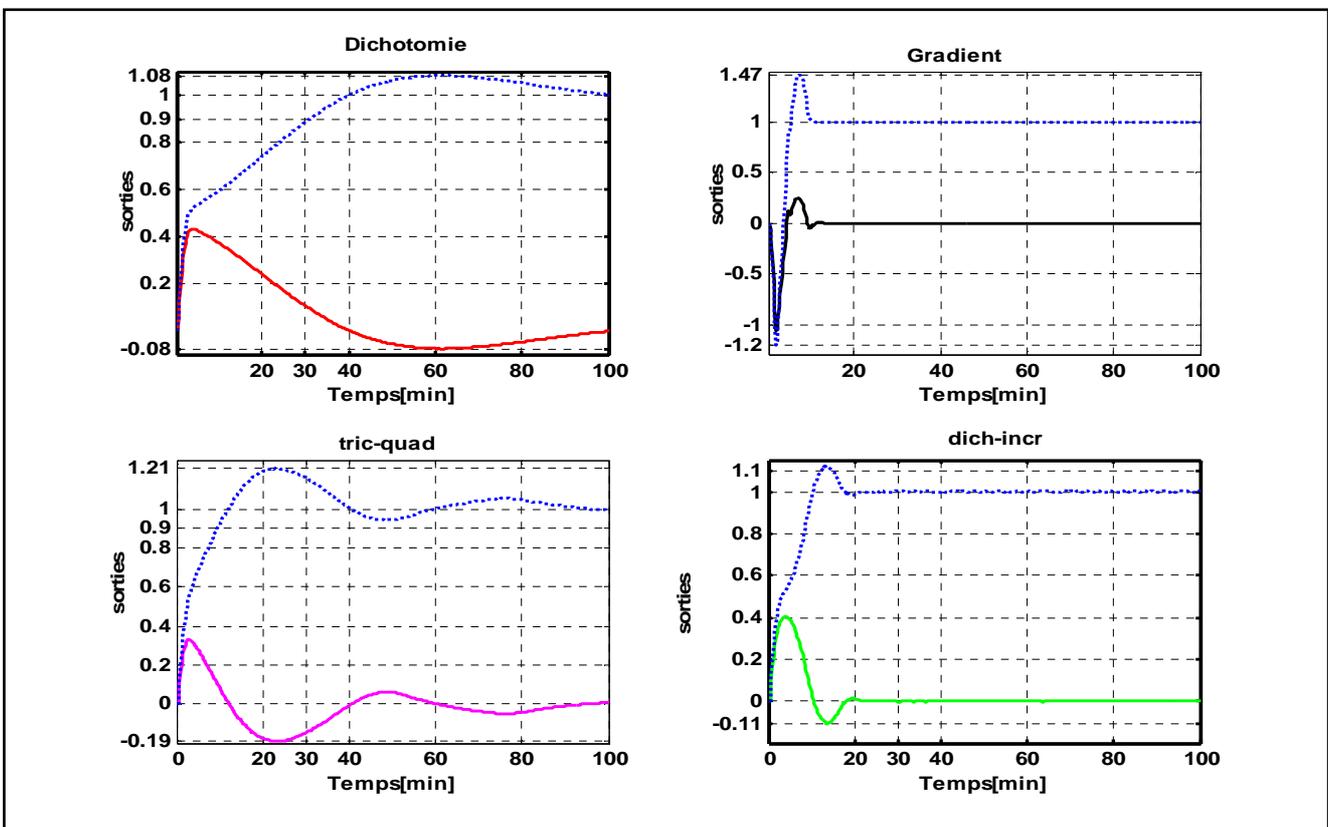


Fig.4.11.1 : Sorties (interaction/poursuite) : cas de  $(K_1 = 1.2, K_2 = 0.8)$  et  $(w_1 = 0, w_2 = 1)$

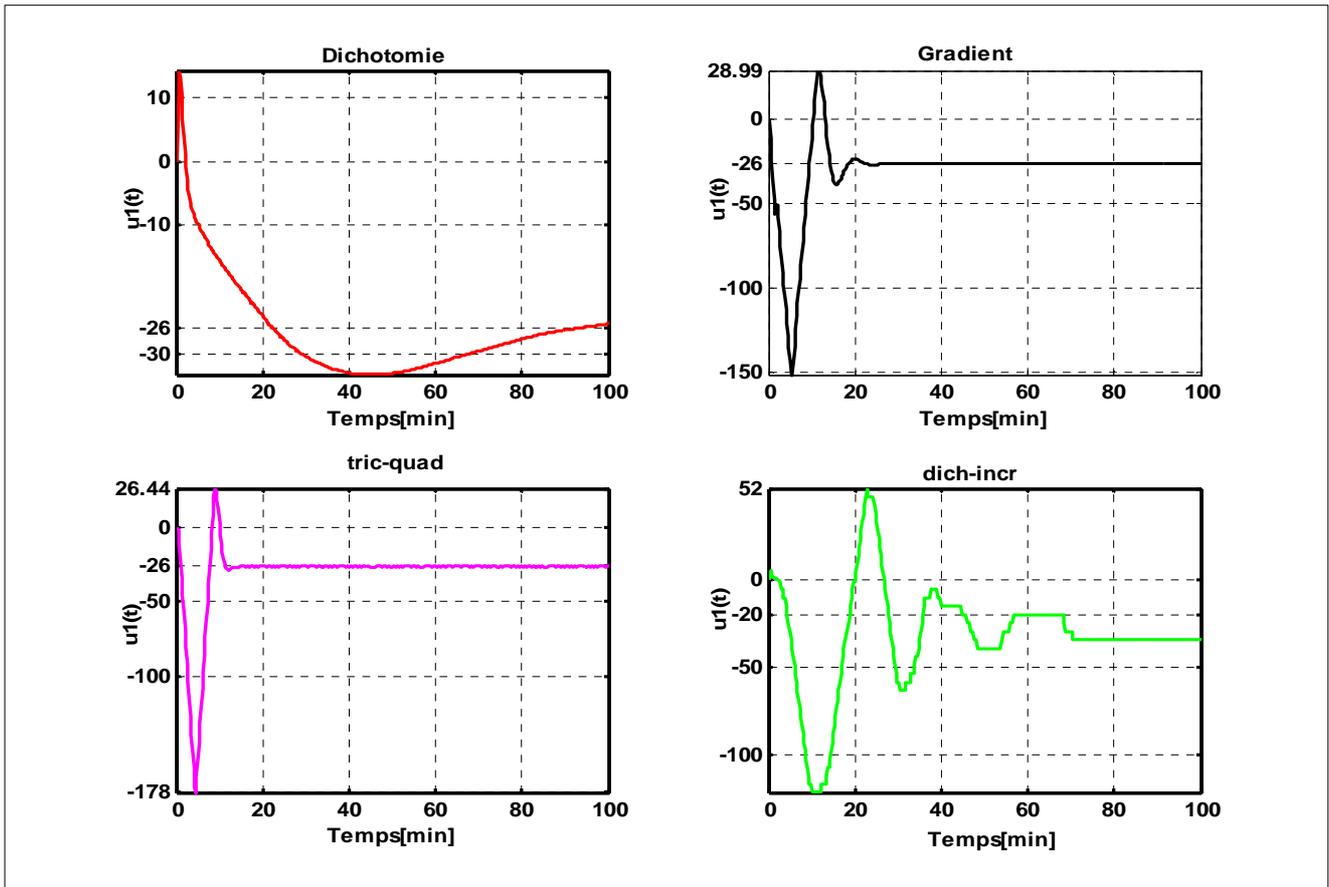


Fig.4.11.2 : Commande de la voie1 : cas de  $(K_1 = 1.2, K_2 = 0.8)$  et  $(w_1 = 0, w_2 = 1)$

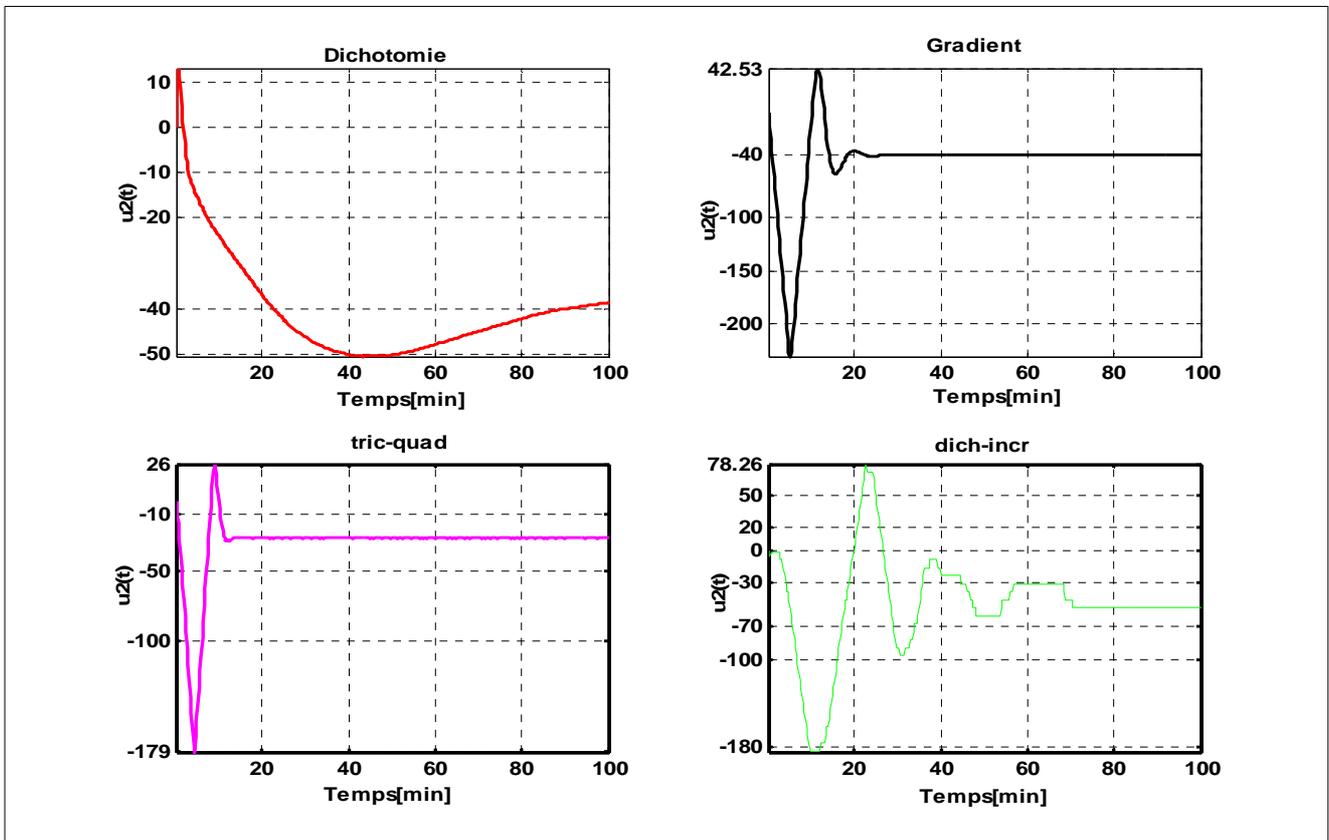


Fig.4.11.3 : Commande de la voie2 : cas de  $(K_1 = 1.2, K_2 = 0.8)$  et  $(w_1 = 0, w_2 = 1)$

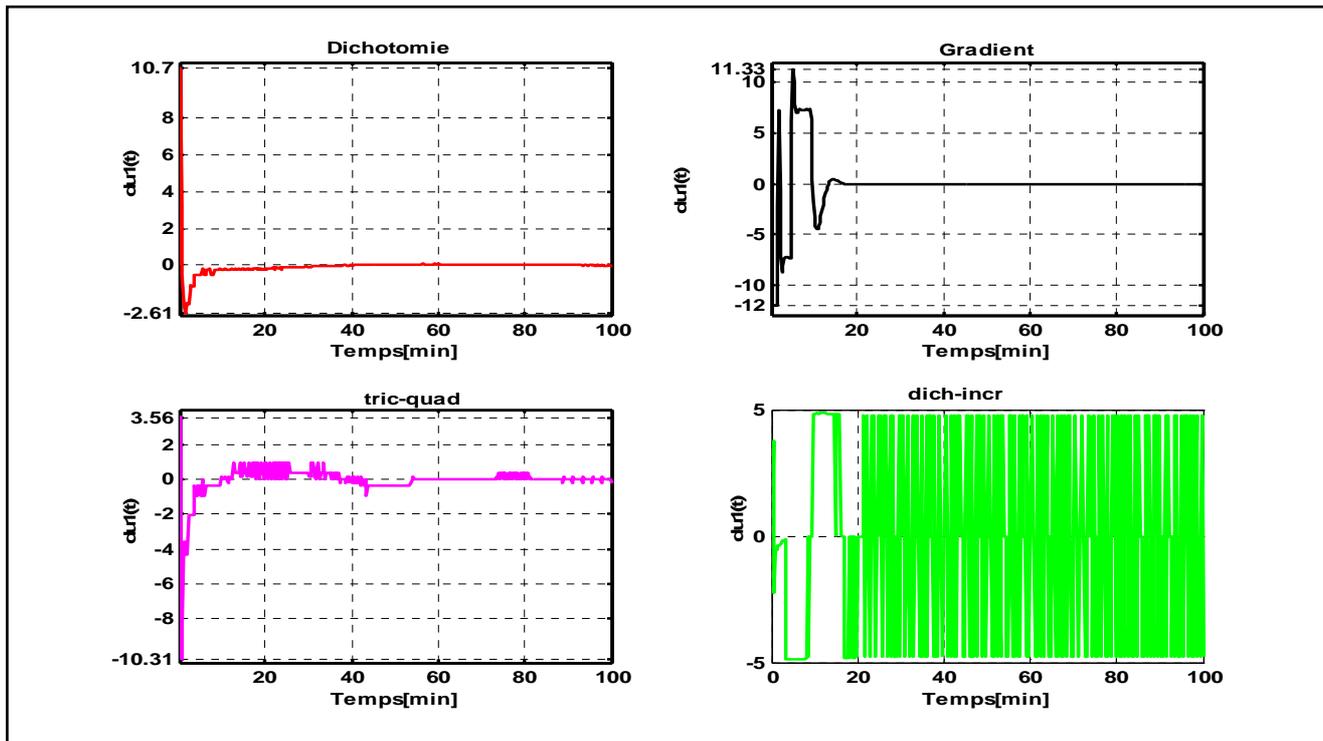


Fig.4.11.4: Incrément de commande de la voie1 : cas de  $(K_1 = 1.2, K_2 = 0.8)$  et  $(w_1 = 0, w_2 = 1)$

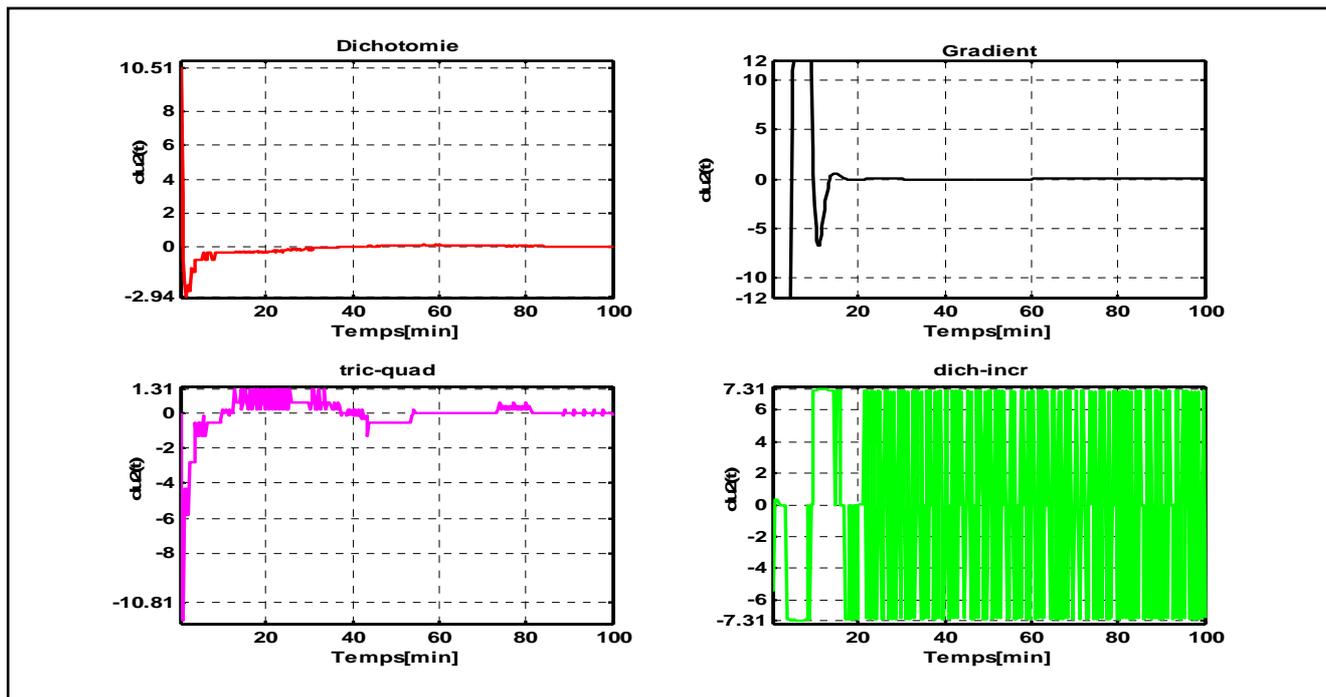


Fig.4.11.5 : Incrément de commande de la voie2 : cas  $(K_1 = 1.2, K_2 = 0.8)$  et  $(w_1 = 0, w_2 = 1)$

#### 4.5. Interprétation de résultats et conclusion :

D'après les figures ci-dessous, on a pu distinguer les remarques suivantes :

- Pour les paramètres donnés  $(N_u, N_2, \Lambda)$  le GPC utilisant les méthodes proposées satisfaisant la plupart des performances données dans le cahier de charge.

▪ Pour les méthodes proposées : la méthode du gradient est la plus rapide par rapport aux autres méthodes avec une convergence néanmoins satisfaisante pour une variation des gains dans le même sens et dans le sens inversé. Les sorties obtenues par cette méthode sont données avec un dépassement très élevé, ce qui dégrade les performances de cette méthode ainsi de limiter leur application. Pour la dynamique de poursuite obtenue par cette méthode, elle est la plus efficace par rapport aux autres méthodes d'optimisation, cependant, l'inconvénient majeur est que l'obtention de la commande se fait sans prendre en compte les contraintes imposées, autrement dit, le minimum de la fonction objective peut être trouvé hors de la région constituée par ces contraintes, dans ce cas, le minimum doit prendre, soit la valeur maximale, soit la valeur minimale de cette région. Un autre inconvénient de cette méthode c'est que le calcul de la sortie, pour chaque voie, nécessite à une fourchette de commande ainsi d'incrément de commande importante par rapport aux autres méthodes proposées. La convergence de cette méthode est ainsi conditionnée par un choix critique du vecteur de départ  $(\tilde{U}_0)$  un mauvais choix sur ce dernier peut conduire à la divergence rapide de la méthode.

▪ Pour les deux méthodes : dichotomie et tric\_quad, pour une variation des gains dans le même sens, l'erreur statique obtenue est très réduite dans le régime stationnaire, les sorties obtenues sont très satisfaisantes pour chaque voie, les performances obtenues variées selon les valeurs des gains, cependant, pour une variation de ces derniers dans un sens inversé, on peut voir nettement une dégradation des performances ainsi ces deux méthodes (avec la méthode de dich\_incr) sont conditionnées par la détermination de la région de faisabilité.

▪ Pour la méthode dich\_incr, les résultats obtenus sont très encourageants, la convergence est assurée avec un temps de calcul réduit, les signaux de commandes et les incréments de commande sont limités par des bornes maximales et minimales très réduites. Pour les différentes valeurs des gains, cette méthode offre à chaque fois des meilleurs résultats par rapport aux autres méthodes d'optimisation et ceux quelques soit le sens de variations de  $K_1, K_2$ .

- Pour toutes les méthodes proposées, les résultats obtenus sont acceptables du point vue poursuite (erreur statique nulle) et du point vue robustesse (sensibilité aux variations des gains incertains). Le choix de chaque méthode parmi lesquelles proposées dépend de la nature du problème à résoudre, par exemple, en utilisant la méthode du gradient dans le cas des systèmes dont les paramètres évoluent rapidement et si la solution obtenue reste toujours à l'intérieure de la région de faisabilité. Même chose pour les méthodes dichotomie, tric\_quad et la méthode dich\_incr elles sont conditionnées par la possibilité de détermination de cette région.

## Conclusion générale :

Nous avons présenté, dans ce travail, les démarches de la commande prédictive multi variable dans sa version algorithmique. Cette méthode de commande a été vérifiée sur trois systèmes multi variables décrits soit par un modèle prédictif de *CARIMA* où par une matrice de transfert carrée. Nous avons montré qu'une bonne performance obtenue dépend d'un meilleur choix des paramètres de synthèse  $N_u, N_2, \Lambda$  de GPC.

A la différence des méthodes de commande robuste proprement dites ( $H_2, H_\infty \dots$ ), la commande prédictive que nous avons étudiée ne prennent pas en compte de façon explicite la robustesse en stabilité ; néanmoins, nous avons montré comment on peut parvenir par une transformation simple à une autre version dite : polynomiale, cette dernière nous a permis, d'un côté d'éviter le calcul en ligne de la commande est nécessite un fardeau de calcul énorme, d'un autre côté, pour faire l'analyse de la robustesse de stabilité face aux incertitudes paramétriques du modèle dans le plan fréquentiel. Nous avons montré qu'à partir une étude comparative entre le GPC polynomial et la méthode  $H_\infty$  sur un benchmark disponible dans la bibliothèque de Matlab que cette la loi de commande nécessite une robustification. A cet effet, nous avons été dans l'obligation d'introduire le filtre  $C(q^{-1})$  dans le modèle *CARIMA* pour le but d'améliorer la dynamique de rejection des perturbations.

L'avantage considérable de GPC algorithmique c'est qu'on peut aussi l'appliquer en cas de présence des contraintes, celles-ci nous ont conduits à l'application, en ligne, d'une méthode d'optimisation pour la détermination de la commande.

Plusieurs ouvrages montrent que la fonction *quadprog* devient très efficace pour résoudre les problèmes quadratiques sous contraintes linéaires, cependant, pour quelques choix de contraintes et quelques paramètres de synthèse de la commande prédictive, cette fonction présente quelques inconvénients, par conséquence de quoi, on a proposé d'autres approches d'optimisation au GPC pour palier ces inconvénients.

Dans la suite de notre étude, nous avons pu apprécier les avantages et les inconvénients de multiples algorithmes d'optimisation et cela grâce à de nombreuses simulations aboutissant à des approches efficaces qui sont : le gradient conjugué, la dichotomie et ses dérivées. Nous avons examiné les

performances de chaque méthode proposée en testant les lois obtenues sur deux systèmes différents. Le premier est un système électromécanique sa dynamique est décrite par une matrice de transfert d'ordre 2; l'application de la commande prédictive utilisant la fonction *quadprog* sur ce processus présente un comportement mécanique non souhaité conduit par conséquence à l'inefficacité de cette fonction sur ce type des systèmes, le problème à été réglé par d'autres approches d'optimisation. Des allers-retours permettraient alors de perfectionner la méthode *Dic\_incr* qui donne de meilleurs résultats avec un domaine de variation des incréments et des commandes moins réduit par rapport aux autres approches proposées.

Le second procédé utilisé pour la simulation représente une colonne de distillation binaire mal conditionnée sa dynamique est décrite par un modèle incertain proposé par Limebeer. Malgré les variations paramétriques de ce modèle proposé, le GPC utilisant les approches proposées garantit la robustesse en stabilité et relativement la robustesse en performances et les résultats obtenus ont été à la hauteur de nos espérances.

Vu que le GPC qui utilise ces approches proposées est relativement facile à manipuler, elle souffre cependant de la détermination de la région de faisabilité qui est considérée comme une base de départ de ces algorithmes et qui limite aussi leurs applications en cas d'un mauvais choix des contraintes. De plus, le choix critique du nombre d'itération qui dépend directement de la précision de calcul peut modifier la convergence de ces approches proposées vers la solution du problème.

La détermination de la région de faisabilité pour des contraintes redondantes ainsi que l'amélioration de ces approches d'optimisation proposées, laissent un champ libre de recherches pour d'éventuels candidats ultérieurement, nous conseillons les futurs chercheurs d'orienter leurs recherches sur le perfectionnement des contraintes nécessaires parmi lesquelles redondantes pour la détermination de cette région et d'essayer d'améliorer les directions de recherche des méthodes directives.

## Perspectives :

Les perspectives à citer pour le GPC multi variable sont :

- 1- En l'absence des contraintes, la perspective la plus directe est de mettre une méthodologie permettant de faire un meilleur choix du filtre  $C(q^{-1})$  conduisant à robustifier le GPC polynomial face à différentes incertitudes paramétriques du modèle. Dans ce cas, une approche basée sur la paramétrisation de Youla avec des techniques d'optimisation sous un formalisme LMI semble la plus adaptée.
- 2- Obtention de la forme polynomiale RST de GPC multivariable sous contraintes pour faire l'analyse de robustesse du système bouclé.
- 3- Extension des approches d'optimisation proposées sur le cas de  $C(q^{-1}) \neq I_m$
- 4- Extension de cette étude sur les matrices de transfert non carrées.

**Introduction :**

Nous montrons dans cette partie une méthode de résolution des équations diophantiennes données dans le chapitre I, cette dernière donnée pour le cas où  $C(q^{-1}) \neq I$  du modèle de *CARIMA*.

**A.1 – Résolution récursive de l'équation diophantienne:**

Soit l'équation diophantienne donnée par (2.12) suivante :

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

Avec :

$$A(q^{-1}) = I + A_1 q^{-1} + \dots + A_{na} q^{-na} \quad (A.2)$$

$$\Delta = 1 - q^{-1} \quad (A.3)$$

$$C(q^{-1}) = I_{mxm} + C_1 q^{-1} + \dots + C_{nc} q^{-nc} \quad (A.4)$$

$$E_j = E_0^{(j)} + E_1^{(j)} q^{-1} + \dots + E_{j-1}^{(j)} q^{-j+1} \quad (A.5)$$

$$F_j = F_0^{(j)} + F_1^{(j)} q^{-1} + \dots + F_{nf^{(j)}}^{(j)} q^{-nf^{(j)}} \quad (A.6)$$

Les coefficients des matrices  $A, C, E_j$  et  $F_j$  sont des polynômes en  $q^{-1}$  qui ont pour degrés respectifs  $n_a, n_c, j-1$  et  $n_f^{(j)}$ .

Les matrices  $A_b, C_b, E_i^{(j)}$  et  $F_i^{(j)}$  sont à coefficients réels et regroupent les coefficients de ces polynômes.

Le couple des matrices de polynôme  $(E_j, F_j)$  est l'unique solution de cette équation, que l'on peut encore mettre sous la forme :

$$\begin{aligned} I_{mxm} + C_1 q^{-1} + \dots + C_{nc} q^{-nc} &= [E_0^{(j)} + E_1^{(j)} q^{-1} + \dots + E_{j-1}^{(j)} q^{-j+1}] \\ [I + (A_1 - I)q^{-1} + (A_2 - A_1)q^{-2} + \dots + (A_{na} - A_{na-1})q^{-na} - A_{na} q^{-na-1}] &+ F_0^{(j)} q^{-j} + F_1^{(j)} q^{-j-1} + \dots + F_{nf^{(j)}}^{(j)} q^{-j-nf^{(j)}} \end{aligned} \quad (A.7)$$

Le degré de la matrice  $E_j A \Delta$  vaut  $n_a + j$  et le degré de  $q^{-j} F_j$  est  $n_f^{(j)} + j$ .

Etant donné que les ordres des deux membres de l'équation (A.7) doivent être identiques, on déduit par identification l'ordre  $n_f^{(j)}$  du polynôme  $F_j$ :

$$\begin{cases} n_c \leq n_a + j \Rightarrow n_a + j = n_f^{(j)} + j \\ n_c \neq n_a + j \Rightarrow n_f^{(j)} + j = n_c \end{cases} \quad (A.8)$$

D'où :

$$n_f^{(j)} = \max(n_a, n_c - j) \quad (A.9)$$

Pour l'horizon de prédiction  $j-1$ , l'équation (A.1) s'écrit :

$$C = E_{j-1}A\Delta + q^{-j+1}F_{j-1} \quad (A.10)$$

A partir de (A.1) et (A.10), on obtient :

$$(E_j - E_{j-1})A\Delta + q^{-j+1}(q^{-1}F_j - F_{j-1}) = 0 \quad (A.11)$$

Etant donné la forme de  $\Delta$  (équation (A.4)), on peut récrire (A.11) de la manière suivante :

$$\begin{aligned} & [(E_0^{(j)} - E_0^{(j-1)}) + (E_1^{(j)} - E_1^{(j-1)})q^{-1} + \dots + (E_{j-2}^{(j)} - E_{j-2}^{(j-1)})q^{-j+2} + E_{j-1}^{(j)}q^{-j+1}]_x \\ & [I + (A_1 - I)q^{-1} + (A_2 - A_1)q^{-2} + \dots + (A_{na} - A_{na-1})q^{-na} - A_{na}q^{-na-1}] - q^{-j+1}[F_0^{(j-1)} + (F_1^{(j-1)} - F_0^{(j)})q^{-1} \\ & + \dots + (F_{nf}^{(j-1)} - F_{nf}^{(j)})q^{-nf} - F_{nf}^{(j)}q^{-nf-1}] = 0 \quad (A.12) \end{aligned}$$

En identifiant les membres de gauche et de droite, on trouve que :

$$E_k^{(j)} - E_k^{(j-1)} = 0 \quad \forall k = 0, \dots, j-2 \quad (A.13)$$

Par conséquent, on a :

$$E_j = E_{j-1} + E_{j-1}^{(j)}q^{-j+1} \quad (A.14)$$

On peut simplifier les notations en écrivant que  $\forall j, E_k^{(j)} = E_k$  Ainsi l'équation (A.12) devient :

$$\begin{aligned} & E_{j-1}z^{-j+1} \left[ \overbrace{I + (I + A_1 - I)q^{-1} + \dots + (A_{na} - A_{na-1})q^{-na} - A_{na}q^{-na-1}}^{A\Delta} \right] = \\ & q^{-j+1} \left[ \underbrace{F_0^{(j-1)} + (F_1^{(j-1)} - F_0^{(j)})q^{-1} + \dots + (F_{nf}^{(j-1)} - F_{nf}^{(j)})q^{-nf} - F_{nf}^{(j)}q^{-nf-1}}_{F_{j-1} - q^{-1}F_j} \right] \end{aligned}$$

Et on déduit que :

$$q^{-1}F_j = F_{j-1} - E_{j-1}A\Delta \quad (A.15)$$

A partir de ces relations, on peut finalement définir l'algorithme de calcul récursif suivant :

*Initialisation de la récurrence :*

Pour  $j=1$ , l'équation (A.7) s'écrit :

$$\begin{aligned} & I + C_1q^{-1} + C_2q^{-2} + \dots + C_{nc}q^{-nc} = \\ & E_0 \left[ I + (I + A_1 - I)q^{-1} + \dots + (A_{na} - A_{na-1})q^{-na} - A_{na}q^{-na-1} \right] \\ & + F_0^{(1)}q^{-1} + F_1^{(1)}q^{-2} + \dots + F_{nf}^{(1)}q^{-nf-1} \end{aligned}$$

Avec  $n_f^{(1)} = \max(n_a, n_c - 1)$ . On en déduit les initialisations suivantes :

$$\begin{cases} E_0 = I \\ F_i^{(j)} = C_{i+1} - E_0(A_{i+1} - A_i) \end{cases} \quad 0 \leq i \leq n_f^{(1)} \quad (A.16)$$

Calcul récursif de  $E$  et  $F$  :

On utilise l'équation (A.15), on déduit la récurrence pour  $j \geq 2$  :

$$\begin{cases} E_{j-1} = F_0^{(j-1)} \\ F_i^{(j)} = F_{i+1}^{(j-1)} - E_{j-1}(A_{i+1} - A_i) \end{cases} \quad 0 \leq i \leq n_f^{(1)} \quad (A.17)$$

Résolution récursive de l'équation diophantienne :

Soit l'équation diophantienne que l'on rappelle ici :

$$E_j B = C G_j + q^{-j} H_j \quad (A.18)$$

Dans laquelle :

$$B = B_0 + B_1 q^{-1} + \dots + B_{nb} q^{-nb} \quad (A.19)$$

$$E_j = E_0 + E_1 q^{-1} + \dots + E_{j-1} q^{-j+1} \quad (A.20)$$

$$C = I_{m \times m} + C_1 q^{-1} + C_2 q^{-2} + \dots + C_{nc} q^{-nc} \quad (A.21)$$

$$\Delta = \delta = 1 - q^{-1} \quad (A.22)$$

$$G_j = G_0^{(j)} + G_1^{(j)} q^{-1} + \dots + G_{j-1}^{(j)} q^{-j+1} \quad (A.23)$$

$$H_j = H_0^{(j)} + H_1^{(j)} q^{-1} + \dots + H_{nh}^{(j)} q^{-nh} \quad (A.24)$$

Les coefficients des matrices  $B, E_j, C, G_j$  et  $H_j$  sont des polynômes en  $q^{-1}$  dont les coefficients sont regroupés dans les matrices réelles  $B_i, E_i, C_i, G_i$  et  $H_i$ . Les degrés respectifs de ces polynômes sont  $nb, j-1, nc, j-1$  et  $nh$ .

On suppose que les matrices  $C$  s'écrit sous forme diagonale, i.e.,  $C(q^{-1}) = c(q^{-1}) I_{m \times m}$  avec  $c(q^{-1})$  est un polynôme de degré  $nc$ .

Le couple des matrices  $(G_j, H_j)$  est l'unique solution de cette équation, que l'on peut encore mettre sous forme suivante :

$$\begin{aligned} & \underbrace{(E_0 + E_1 q^{-1} + \dots + E_{j-1} q^{-j+1})}_{S_1} \underbrace{(B_0 + B_1 q^{-1} + \dots + B_{nb} q^{-nb})}_{S_1} = \\ & \underbrace{(I + C_1 q^{-1} + C_2 q^{-2} + \dots + C_{nc} q^{-nc})}_{S_2} \underbrace{(G_0^{(j)} + G_1^{(j)} q^{-1} + \dots + G_{j-1}^{(j)} q^{-j+1})}_{S_2} + \\ & \underbrace{(H_0^{(j)} + H_1^{(j)} q^{-1} + \dots + H_{nh}^{(j)} q^{-j-nh})}_{S_3} \end{aligned} \quad (A.25)$$

L'ordre des polynômes des deux membres de (A.25) doit être identique. D'où les relations suivantes :

- Si ordre ( $S_1$ )  $\leq$  ordre ( $S_2$ ) alors ordre ( $S_3$ ) = ordre ( $S_2$ )
- ordre ( $S_1$ )  $>$  ordre ( $S_2$ ) alors ordre ( $S_3$ )  $\leq$  ordre ( $S_1$ ) (A.26)

Ce qui se traduit par :

$$\begin{cases} nb \leq nc \Rightarrow nh = nc - 1 \\ nb \neq nc \Rightarrow nh = nb - 1 \end{cases} \quad (A.27)$$

On en déduit l'ordre  $nh$  du polynôme  $H$  :

$$nh = \max(nc, nb) - 1 \quad (A.28)$$

On récrit l'équation (A.18) pour l'horizon de prédiction  $j-1$  :

$$E_{j-1}B = CG_{j-1} + q^{-j+1}H_{j-1} \quad (A.29)$$

En faisant la différence entre (A.18) et (A.29), on obtient :

$$C(G_j - G_{j-1}) + q^{-j+1}[q^{-1}H_j - H_{j-1} - q^{j-1}(E_j - E_{j-1})B] = 0 \quad (A.30)$$

Or (d'après l'équation (A.14)) :

$$E_j - E_{j-1} = E_{j-1}q^{-j+1} \quad (A.30)$$

D'où :

$$C(G_j - G_{j-1}) - q^{-j+1}[H_{j-1} - q^{-1}H_j + E_{j-1}B] = 0 \quad (A.31)$$

En récrivant cette équation sous forme étendue on arrive à :

$$\begin{aligned} & (I + C_1q^{-1} + C_2q^{-2} + \dots + C_{nc}q^{-nc}) \cdot [(G_0^{(j)} - G_0^{(j-1)}) + (G_1^{(j)} - G_1^{(j-1)})q^{-1} + (G_2^{(j)} - G_2^{(j-1)})q^{-2} + \dots \\ & + (G_{j-2}^{(j)} - G_{j-2}^{(j-1)})q^{-j+2} + G_{j-1}^{(j)}q^{-j+1}] - q^{-j+1}[H_0^{(j-1)} + \dots + (H_{nh}^{(j-1)} - H_{nh-1}^{(j)})q^{-nh} - H_{nh}^{(j)}q^{-nh-1} + \\ & E_{j-1}(B_0 + B_1q^{-1} + \dots + B_{nb})] = 0 \quad (A.32) \end{aligned}$$

Etant donné que les polynômes de  $C$  ne peuvent pas être factoriser par  $q^{-j+1}$ , on trouve que :

$$G_k^{(j)} - G_k^{(j-1)} = 0 \quad \forall k = 0, 1, \dots, j-2 \quad (A.33)$$

Et par conséquent, on a :

$$G_j - G_{j-1} = G_{j-1}^{(j)}q^{-j+1} \quad (A.34)$$

L'égalité (A.33) permet de simplifier les notations et d'écrire que  $\forall j, G_k^{(j)} = G_k$ , ainsi l'équation (A.32) devient :

$$CG_{j-1}q^{-j+1} - q^{-j+1}[H_{j-1} - q^{-1}H_j + E_{j-1}B] = 0 \quad (A.35)$$

D'où l'on peut déduire l'équation réursive suivante :

$$q^{-1}H_j = H_{j-1} - CG_{j-1} + E_{j-1}B \quad (A.36)$$

Ces relations permettent de définir l'algorithme de calcul réursif suivant :

*Initialisation de la récurrence :*

Pour  $j=1$ , l'équation (A.25) s'écrit :

$$E_0(B_0 + B_1q^{-1} + \dots + B_{nb}q^{-nb}) = (I + C_1q^{-1} + \dots + C_{nc}q^{-nc})G_0 + (H_0^{(1)}q^{-1} + \dots + H_{nh}^{(1)}q^{-1-nh}) \quad (A.37)$$

On déduit que :

$$\begin{cases} G_0 = E_0B_0 \\ H_i^{(1)} = E_0B_{i+1} - C_{i+1}G_0 \quad 0 \leq i \leq nh \end{cases} \quad (A.38)$$

*Calcul réursif de G et H :*

En récrivant l'équation (A.36) de manière étendue, on obtient :

$$\begin{aligned} H_0^{(j)}q^{-1} + \dots + H_{nh}^{(j)}q^{-nh-1} &= H_0^{(j-1)} + H_1^{(j-1)}q^{-1} + \dots + H_{nh}^{(j-1)}q^{-nh} - (I + C_1q^{-1} + \dots + C_{nc}q^{-nc})G_{j-1} \\ &+ E_{j-1}(B_0 + B_1q^{-1} + \dots + B_{nb}q^{-nb}) \end{aligned} \quad (A.40)$$

Pour  $j \geq 2$  on a donc :

$$\begin{cases} G_{j-1} = H_0^{(j-1)} + E_{j-1}B_0 \\ H_i^{(j)} = H_{i+1}^{(j-1)} - C_{i+1}G_{j-1} + E_{j-1}B_{i+1} \quad 0 \leq i \leq nh \end{cases} \quad (A.41)$$

## Références:

- [Cama93] E.F.Camacho, “*Constrained Generalised Predictive Control*”, IEEE Transaction. on Automatic .Control, AC-38, pp.327-332, (2)1993.
- [CB99] E.F.Camacho, C. Bordons, “ *Model Predictive Control* ”, edition Springer-Verlag London,GB,1999
- [CM86] P.J. Campo, M. Morari, “*infinity-norm formulation of model predictive control problems*”, in proc. American Control conf., Seattle, WA, pp.339-342, 1986.
- [CM89] Clarck, D. W., C. Mohtadi, ” *Properties of generalized predictive control*”. Automatica, pp.859-875, (26)1989.
- [CMT87] D. W.Clarck, , C. Mohtadi, and P.S. Tuffs, “*Generalized Predictive Control –Part I : The basic algorithm – Part II: extensions and interpretation*”, Automatica, pp.137-160, (23)1987.
- [CMT95] D.W. Clarke, C.Mohtadi, P.S. Tuffs. “*Predictive application of constrained receding horizon control to the benchmark problem*” European control conf. Rome, Italy, pp. 22-24, Sep. 1995.
- [CR80] C.R.Cutler, B.L.Ramaker, “*Dynamic matrix control: a computer control algorithm*”. Proc. of the Joint American Control Conference.San Francisco, California, USA, pp 36-38, 1986.
- [CS83] T.S.Chang, D.E.Seborg, “*A linear approach programming for multivariables feedback control with inequality constraints*”, Int. J. of control, pp.583-597, (37)1983.
- [CS91] D.W.Clark,F.R.Scattolini, “*Constrained Receding Horizon Predictive Control* ”, (CRHPC),*IEE Proceedings*, .vol. 138, July, (4)1991.
- [CS99] R.Y. Chiang and M. G. Safonov, “*Robust control tool Box: Users Guides*”, version 2, 1999.

- [Cuv06] L.Cuvillon, “*Compensation Du Battement Cardiaque en Chirurgie Robotisée : Asservissement Visuel D’un Robot Médical Avec Flexibilités*”, Thèse présentée pour obtenir le grade de Docteur de l’Université Louis Pasteur Strasbourg I, 2006.
- [DDN90] J.M Dion, L. Dugard, T. Nguyenminh. “*Some developments on the order adaptive multi-variable*”, lab. of car. Grenoble, (Uacnrs 228), 24-145-170, pp.144 – 168, 1990.
- [DWKPD97] P. Dave, D.A. Willig, G.K. Kudva, J.F. Pekny, F.J. Doyle, “*LP methods in MPC of large-scale systems:application to paper-machine CD control*”, AIChE Journal 43, pp.1016-1031, (4)1997.
- [FD97] B.A Francis, J.C Doyle “*Linear control theory with year  $H_{\infty}$  optimally criterion*”, control opt, vol.25, pp.815-844, 1997.
- [Flech90] R. Fletcher,” *Practical method of optimisation*”, (John Wiley and Sons, 1990).
- [GB91] A.Gourdin, M.Boumahrat, «*Méthode numériques appliquées* », seconde édition, O.P.U, pp.87-94, 1991
- [GPM86] C.E. Garcia, D.M. Prett, A.M. Morshedi, “*Quadratic programming Solution of Dynamic matrix control (QDMC)*”, Chem, Eng. Commun. pp.73-87, (46)1986
- [IOT97] A.Ishikawa, M. Ohshima, M. Tanigaki, “*A practical method of removing ill-conditioning in industrial constrained predictive control*”, Computer chem Engng, Vol. 21, Suppl, pp. S1093-S1098, Elsevier Science Ltd Printed in Great Britain,1997.
- [KBM96] M. Kothare, V. Balakrishnan, etM. Morari «*Robust Constrained Model Predictive Control using Linear Matrix Inequalities*”. *Automatica*, vol. 32, pp.1361-1379, 1996.
- [KC84] A.G. Kuznetsov, D. W. Clarke, “*Application of constrained GPC for improving performance of controlled plants*”, in D.W. Clarke (Ed.), *Advances in model-Based Predictive Control*, Oxford Science. pp.318-329, 1984.
- [KC89] R.M.C.de Keyzer, A.R.van Cauwenberghé,” *Extended prediction self-adaptive control*”. *IEEE Transactions on Automatic Control*, pp.831-847, (34)1989.
- [KRC92] B. Kouvaritakis, J.A. Rossiter, A.O.T.Chang, “*Stable GPC: n algorithm with guaranteed stability*”, *IEE Proceeding*, part D, vol.139, pp.349-362, (4)1992.

- [Lew89] D.R Lewing, “*Stable robust performance specifications for uncertain SISO systems*”, Int. J. Control, pp.1263-1281, (53)1989.
- [LI97] G.Lightbody,G.W.Irwin, “*Non linear Control Structures Based on Embedded Neural System Models*”, *IEEE Transaction.on Neural Networks*, pp.553-567, (8)1997.
- [Lime90]D.J.N. Limebeer, “*The specification and purpose of a controller design case study*”, in Proc. IEEE Conf. Decision Contr., Brighton, U.K., pp.1579-1580, 1990.
- [MB93] S.Mokhtar, H.Shetty, “*Non linear programming- theory and algorithms-*”, second edition, John Wiley& Sons.Inc, pp 274-282, pp420- 433, 1993.
- [MC99] M.M'saad, J.Chebassier. “*Adaptive Order of the systems*”, techniques of the engineer, labo.d' car. S7 426\_1 - S7 426-23, 1999
- [Mino88] M.Minoux, «*Programmation mathématique – théorie et algorithmes-* » édition Dunod, 1988.
- [Oga93] K.Ogata “*Discrete time control systems*” university of Minnesota, USA, flight II, pp.578 – 600, 1993.
- [Ramo01]G. Ramond, “*Contribution A La Commande Prédictive Généralisée Adaptative Directe Et Application* », Thèse De Doctorat ,Université Paris XI Orsay. pp08-33, 2001.
- [Rich78] J.Richalet, “*Application to Industrial Processes*”, *Automatica* 14, pp.413-428, 1978
- [Rick86] N.L. Ricker, “*The use of quadratic programming for constrained internal model control*”, ind. Eng. Commun,pp.73-87, (46)1986.
- [Rich99] J. Richalet “*Predictive order* ”, techniques of the engineer, R7 423\_1 – R7 423\_17, 1999.
- [Rodr03]P .Rodriguez Ayerbe “*Robustification De Lois De Commande Prédictive Par la Paramétrisation De Youla* »,Thèse Docteur ,Université Paris XI Orsay (Supélec), pp14-36, 2003.
- [SSB93] C.M. Setty, H. D. Sherali, S. Bazaraa. “*Non linear programming theory and algorithms*” Second edition, pp. 129- 167, pp. 256 – 280, pp. 408 – 488, 1993.
- [Sug64] N.Sugie, “*An extension of Fibonacci searching to multidimensional cases*”, IEEE. Trans. Autom. Control, pp.105-108, (9)1964.

## المخلص

يقترح هذا العمل خوارزمية قانون التحكم التنبؤي المتعدد الأبعاد و بدون قيود و هذا لمراقبة الأنظمة العديدة المتغيرات كما أنه يقترح شكل كثير الحدود لهذا القانون و هذا: من جهة لتقليل الوقت المستغرق لحساب التحكم و من جهة أخرى يسمح بتحليل و

دراسة صلابة الاستقرار و المزايا من أجل ارتياب عوامل الأنظمة، هذا الشكل طبق على نظام مرجعي موجود في

Toolbox Robust Matlab النتائج المتحصل عليها قورنت مع الطريقة الصلبة  $H_{\infty}$  وهذا للتحقق من صلابة

الاستقرار و المزايا وفي وجود القيود: أربعة طرق أخرى سهلة الاستعمال أدرجت و طبقت من أجل حل مسألة التحكم التنبؤي

و هم Dich incr و طريقة Tric-quad, dichotomie, Gradient conjugue و

قانون التحكم التنبؤي المقيد المعتمد على هذه الطرق المقترحة طبق على أنظمة متعددة المتغيرات

و النتائج بالمقارنة مع طريقة البرمجة المربعة (*quadprog*) كانت مبهرة مع سعة تحكم و قيود مضبوطة.

## Abstract:

This work presents a low of the multivariable predictive control in the algorithmic version for the control of multivariable systems without constraints. Other version of this low (polynomial version) is proposed in this study for minimizing the computation time to calculate the control low; it permits also to analyse the stability and the performances robustness face of the parametric uncertainties of model. The polynomial version proposed is applied in the benchmark “*josedemo*” available in the Toolbox Robust Matlab and the obtained results are compared by those obtained by the  $H_{\infty}$  robustness method for verifying the stability robustness condition and the performances robustness condition.

For the constrained case, four optimisation methods requiring modest computational resources and which are easy to implement have been introduced and applied to solve on line the control low problem of GPC multivariable under constrained. These methods are: The conjugate gradient, dichotomy, tric\_quad and the dich\_incr method.

The GPC multivariable using these optimization methods is applied in the some multivariable systems, the obtained results by the proposed methods are compared with those obtained with the GPC multivariable using the *quadprog* function (Quadratic Programming available in the Toolbox Optimization of Matlab) which give by the proposed methods an appreciable results with the satisfactory imposed constraints.

**Keywords:** Multivariable Predictive Control,  $H_{\infty}$  method, RST controller, Quadratic Programming (*quadprog*), Conjugate Gradient, Dichotomy, Tric\_quad, Dich\_incr, distillation column.

## Résumé:

Ce travail présente la version algorithmique de la loi de la commande prédictive multi variable sans contraintes, celle-ci pour commander les systèmes multidimensionnels. Une autre version polynomiale de cette loi de commande est aussi proposée pour, d'un coté, minimiser le temps de calcul de commande et d'autre coté pour faire l'analyse de robustesse de stabilité et de performances face aux incertitudes paramétriques du modèle. La version polynomiale proposée est appliquée sur un modèle de référence « *josedemo* » disponible dans la bibliothèque Robuste/Matlab et les résultats obtenus sont comparés à ceux trouvés par la méthode robuste  $H_\infty$ , celle-ci pour vérifier la condition de robustesse sur la stabilité et sur les performances.

En présences des contraintes, quatre méthodes d'optimisation, qui sont faciles à manipulées, sont introduites et appliquées pour résoudre en ligne le problème de GPC multivariable sous contraintes, ces méthodes sont : gradients conjugués, dichotomie, tric\_quad et la méthode dic\_incr.

Le GPC multivariable utilisant ces dernières est appliqué sur deux systèmes multivariables et les résultats obtenus sont comparés à ceux utilisant la fonction *quadprog* (programmation quadratique qui est disponible dans la bibliothèque d'optimisation de Matlab) où les résultats trouvés sont très appréciables avec satisfaction des contraintes imposées.

**Mots clés :** commande prédictive multi variable, contrôleur RST, la méthode  $H_\infty$ , programmation quadratique, le gradient conjugué, dichotomie, tric\_quad, dich\_incr, colonne de distillation.