

877

République Algérienne Démocratique et Populaire  
Ministère de l'enseignement supérieur et de la recherche scientifique  
Université 8Mai 1945 – Guelma  
Faculté des sciences et de la Technologie  
Département d'Electronique et Télécommunications



**Mémoire de fin d'étude  
Pour l'obtention du diplôme de Master Académique**

Domaine : **Sciences et Technologie**  
Filière : **Génie Electrique**  
Spécialité : **Systemes Electronique**



---

**Commande prédictive généralisée sous contraintes d'un  
processus non linéaire.**

---

Présenté par : **LAYADA LAZHAR**

Sous la direction de :  
**Mr. SEDRAOUI Moussa**

Juin 2015

République Algérienne Démocratique et Populaire  
Ministère de l'enseignement supérieur et de la recherche scientifique  
Université 8Mai 1945 – Guelma  
Faculté des sciences et de la Technologie  
Département d'Electronique et Télécommunications



**Mémoire de fin d'étude**  
**Pour l'obtention du diplôme de Master Académique**

Domaine : **Sciences et Technologie**  
Filière : **Génie Electrique**  
Spécialité : **Systemes Electronique**

---

---

**Commande prédictive généralisée sous contraintes d'un  
processus non linéaire.**

---

---

Présenté par : **LAYADA LAZHAR**

Sous la direction de :  
**Mr. *SEDRAOUI Moussa***

Juin 2015

# Sommaire



Introduction générale.....	1
----------------------------	---

## Chapitre I *la commande prédictive généralisée sans contraintes*

I.1	Introduction .....	3
I.2	GPC mono-variable.....	3
I.2.1	Modèle CARIMA.....	3
I.2.2	Critère à minimiser.....	5
I.2.3	Prédicteur optima.....	6
I.2.4	Calcul de commande .....	9
I.3	Choix des paramètres de synthèse .....	11
I.4	Test de simulation.....	13
I.5	Conclusion.....	16

## Chapitre II *GPC sous contraintes*

II.1	Introduction.....	17
II.2	Les contraintes.....	18
II.3	Formulation des contraintes.....	19
II.3.1	Contraintes sur les incréments de commandes.....	20
II.3.2	Contraintes sur les amplitudes de commandes.....	21
II.3.3	Contraintes sur les signaux de sorties.....	22
II.3.4	Contraintes terminales de type égalité.....	23
II.3.5	Autres types de contraintes.....	23
II.4	Resultats de simulation.....	25
II.5	Conclusion.....	28

## Chapitre III *Application sur un système non linéaire*

III.1	Introduction.....	29
III.2	Le réacteur exothermique continu agité (CSTR).....	30
III.2.1	Description du procédé.....	30
III.3	Simulation.....	32
III.4	Conclusion.....	40
	Conclusion générale.....	41

INTRODUCTION  
GENERALE

## INTRODUCTION GENERALE

Dans les dernière, la stratégie de la commande prédictive à base de modèle *CARIMA Controlled Auto Regressive Intégral Moving Average* a attiré une grande attention tant sur le plan pédagogique que sur le plan industriel.

Cette méthodologie a été introduite pour la première fois par *Mohtadi*. Plus tard, plusieurs chercheurs ont développé d'autres algorithmes en se basant sur le même principe qui consiste à calculer une loi de commande en minimisant une fonction coût dans le but de faire atteindre la sortie du système à commander à la consigne. Le pilier de base de ces algorithmes est le prédicteur qui permet de prédire la sortie future du système dans un horizon du temps étendu. Ces algorithmes sont appliqués pour les systèmes linéaires invariants dans le temps. Pour les modèles linéaires variant dans le temps ces approches sont appliquées dans des structures auto adaptatives.

Parmi ces algorithmes, l'approche de *Clark* est appelée *Generalized predictive control GPC* qui représente l'algorithme le plus populaire où le prédicteur est donné sous une forme explicite et claire. Cet algorithme a été la base pour plusieurs auteurs qui ont développé plusieurs alternatives.

Dans le monde de l'industrie, la quasi-totalité des équipements sont soumis aux contraintes physiques qui se répercutent sur les boucles de commandes. Pour ces raison, les solutions analytique obtenues par les algorithmes qui ne tiennent pas compte des contraintes peuvent dégrader la performance de la commande voire même causer l'instabilité des systèmes.

Pour remédier ces inconvénients, les contraintes physique sont formulées mathématiquement avec la fonction coût de la stratégie prédictive et un problème d'optimisation linéaire sous contraintes est posé dont la solution est donnée par plusieurs techniques

Ces algorithmes considèrent l'utilisation des modèles linéaires, ces derniers ne sont pas toujours satisfaisant pour décrire avec fidélité la dynamique des procédés réels qui sont dans la plupart des cas non linéaires. Pour appliquer la commande GPC sur un système non linéaire, on doit le linéariser autour d'un point de fonctionnement. Puis un modèle linéaire flou, neuronal ou d'autre est utilisé pour le calcul de la commande. Dans ce travail, nous allons utiliser un modèle flou de *Takaggi Suggeno* (T-S) pour la linéarisation d'un processus industriel non linéaire. Ce dernier est une colonne de distillation dont sa dynamique est présentée par des équations différentielles non linéaires. Le modèle utilisé pour la commande sera développé on fait recourir la fonction *genfis2* du *Matlab*.

Ce travail a pour objectif de présenter une nouvelle approche qui se base sur le formalisme de GPC de *Clarke* où la solution au problème d'optimisation est obtenue par la méthode de *Quadprog* (quadratic programming) pour les fonctions convexes qu'on a adapté aux fonctions quadratique dans le but de réduire le temps de calcul et pour le rendre utilisable pour calculer la commande prédictive sous contraintes pour les systèmes mono-variables.

Ce travail est organisé comme suit:

- Chapitre1: nous allons présenter les concepts de base de la méthode GPC en absence des contraintes. Cette partie sera achevée par des résultats de simulation qui montre l'efficacité de cette méthode de commande.
- Chapitre2: décrit le formalisme de la commande prédictive en se basant sur le cas le plus populaire GPC où on présente la solution du problème sous contraintes à partir de l'application de la fonction *Quadprog* pour chaque instant du temps. Cette partie sera terminée par l'application de cette méthode de commande sur quelques systèmes mono variables en présence des contraintes linéaires.

- Chapitre3: dans cette partie, une extension de l'application de la commande prédictive généralisée GPC sous contraintes sur une colonne de distillation.
- Conclusion générale.
- Références.

CHAPITRE I  
LA COMMANDE  
PREDICTIVE GENERALISEE  
SANS CONTRAINTES



## I-1 Introduction

La commande prédictive généralisée GPC est une des derniers membres de la famille des commandes à horizon étendu LRPHC [1] *Long range horizon predictive control*. Elle présente une généralisation des algorithmes de cette famille. Elle se base sur les concepts suivants :

- Le modèle de prédiction utilisé est un modèle de *CARIMA Controlled Auto Regressive Intégral Moving Average* qui est une extension du modèle CARMA comportant une action intégrale dans le but d'éliminer l'écart permanent et l'effet des perturbations constantes [2].
  - L'utilisation de la prédiction à horizon étendu.
  - La résolution récursive de l'équation de Diophantienne (voir annexe).
  - L'introduction de la pondération sur les incréments de commande dans le critère à minimiser ;
- Le choix de l'horizon de commande à partir duquel tous les incréments sont pris nuls

La combinaison de ces concepts, issus de plusieurs algorithmes, a permis l'obtention d'une méthode de commande puissante et performante qui a donné des résultats très satisfaisants dans plusieurs applications industrielles.

## I.2. GPC mono-variable [2-5]:

### I.2.1. Modèle CARIMA:

Toute commande prédictive nécessite la connaissance d'un modèle afin de prédire le comportement futur du système. Dans la commande GPC, le modèle utilisé est celui de *CARIMA* donné par:

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

$A(q^{-1})$ ,  $B(q^{-1})$  et  $C(q^{-1})$  sont des polynômes dans l'opérateur retard  $q^{-1}$ , définis par :

$$\begin{aligned}
 A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a} \\
 B(q^{-1}) &= b_0 + b_1 q^{-1} + \dots + b_{n_b} q^{-n_b} \\
 C(q^{-1}) &= 1 + C_1 q^{-1} + \dots + C_{n_c} q^{-n_c}
 \end{aligned}
 \tag{1.2}$$

Où les degrés des polynômes  $A$ ,  $B$  et  $C$  sont respectivement  $n_a$ ,  $n_b$  et  $n_c$ .

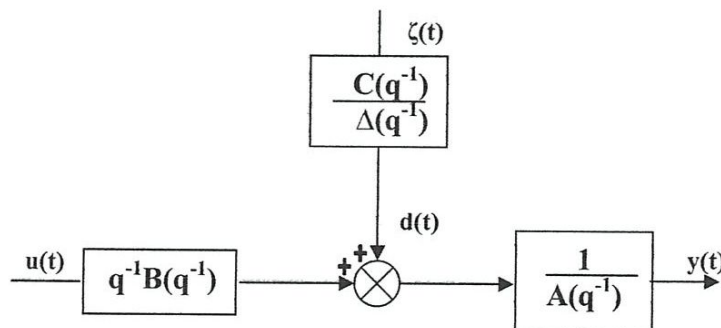
$u(t)$ ,  $y(t)$  et  $\zeta(t)$  sont respectivement l'entrée, la sortie et le signal de perturbation du système.

- $\zeta(t)$  est considéré aléatoire de moyenne nulle et de variance constante.
- $C$  est un polynôme qui modélise l'influence du bruit qui affecte sur le système à commander.

L'introduction de l'opérateur de retard  $\Delta(q^{-1}) = 1 - q^{-1}$  dans le modèle de bruit assure une action intégrale dans le correcteur et permet, donc, d'annuler toutes erreurs statiques vis-à-vis d'une entrée de perturbation en échelon.

La présence de perturbations de charge en échelon dans de nombreux processus industriels, celui-ci garantissant donc un comportement avec une erreur statique nulle face aux perturbations considérées. Ceci est une conséquence directe du principe du modèle interne.

Ce modèle *CARIMA* peut le représenter par la figure I.1 [6-7].



**Figure 1.1 :** Modèle de CARIMA

**I.2.2. Critère à minimiser:**

La loi de GPC est obtenue à partir d'une minimisation d'un critère quadratique portant sur les erreurs futures avec un terme de pondération sur les incréments de commande :

$$J = esp \left[ \sum_{j=N_1}^{N_2} [w(t+j) - \hat{y}(t+j)]^2 + \lambda \sum_{j=1}^{N_u} [\Delta u(t+j-1)]^2 \right] \quad (1.3)$$

Tenant compte la contrainte suivante :

$$\Delta u(t+j) = 0 \text{ pour } j \geq N_u.$$

Où  $N_1$  : représente l'horizon minimal de prédiction de sortie. Ce dernier est supposé dans la suite de ce travail, égal à 1.

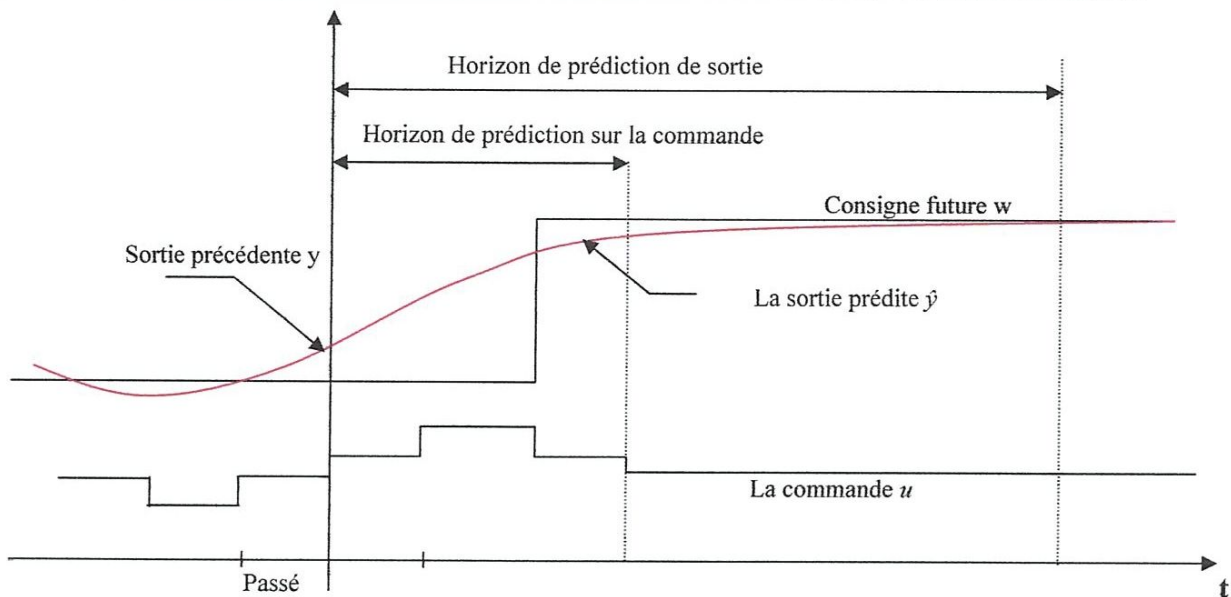
$N_2$  : est l'horizon maximal de prédiction de sortie.

$N_u$  : représente l'horizon de prédiction sur la commande.

$\lambda$  : est un facteur de pondération de l'énergie de commande.

$\Delta u(t+j-1)$  : est un incrément de commande appliqué à l'instant  $t+j-1$ .  $w(t+j)$ ,  $y(t+j)$  représentent, respectivement, l'entrée de consigne, la sortie prédite appliquées à l'instant  $t+j$ .

L'idée de base de cette loi de commande est de calculer une séquence de commande telle que l'écart entre la sortie prédite et la consigne désirée soit minimale par rapport au critère (1.3), et ce sur un horizon de prédiction  $N_2$  (voir la figure 1.2).



**Figure 1.2:** Le principe de base de la commande prédictive à base de modèle

### I.2.3. Prédicteur optimal [8] :

On s'intéresse ici à prédire la sortie du modèle à l'instant  $j$ , en fonction des données connues, c'est à dire, la sortie jusqu'à l'instant  $t$  et la commande jusqu'à l'instant  $t-1$ . La démarche suivie est la suivante:

- La valeur de la sortie à l'instant  $t+j$  est compte tenu de (1.1) soit:

$$y(t+j) = \frac{B(q^{-1})}{A(q^{-1})} u(t+j-1) + \frac{C(q^{-1})}{A(q^{-1})\Delta(q^{-1})} \xi(t+j) \quad (1.4)$$

- Le terme  $\frac{C(q^{-1})}{A(q^{-1})\Delta(q^{-1})} \xi(t+j)$  peut être décomposé en une partie dépendante des valeurs futures de  $\xi$  et en une deuxième partie dépendante des valeurs passées de  $\xi$ , d'où:

$$y(t+j) = \frac{B(q^{-1})}{A(q^{-1})} u(t+j-1) + E_j(q^{-1}) \xi(t+j) + \frac{F_j(q^{-1})}{A(q^{-1})\Delta(q^{-1})} \xi(t) \quad (1.5)$$

➤ Les deux polynômes  $E_j$  et  $F_j$  sont donnés par la résolution de l'équation diophantienne suivante [9] :

$$C = E_j A \Delta + q^{-j} F_j \quad (1.6)$$

$$\text{Avec : } \begin{aligned} E_j &= e_0^j + e_1^j q^{-1} + \dots + e_{j-1}^j q^{-j+1} \\ F_j &= f_0^j + f_1^j q^{-1} + \dots + f_{nf}^j q^{-nf} \end{aligned} \quad (1.7)$$

Sachant que :  $nf = \max(na, nc-j)$ .

La résolution des équations (1.6) de manière récursive est décrite en *annexe*.

En opérant (1.1) par  $E_j q^j$ , on obtient:

$$E_j A \Delta y(t+j) = E_j B \Delta u(t+j-1) + E_j C \zeta(t+j) \quad (1.8)$$

La substitution de  $E_j A \Delta$  provenant de l'équation (1.6) donne alors :

$$(C - q^{-j} F_j) y(t+j) = E_j B \Delta u(t+j-1) + E_j C \zeta(t+j) \quad (1.9)$$

On obtient après le développement et en exploitant la forme particulière de  $C$  :

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

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 donc 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  $\hat{y}(t+j)$  au sens du minimum de la variance de l'erreur, obtenue grâce à des grandeurs dont la valeur est connue à l'instant  $t$  est donnée par:

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

On peut noter que ce résultat est classique pour la détermination de prédictions optimales. Il met clairement en évidence que le terme  $C$  permet de modifier la dynamique de l'observateur (le prédicteur de la sortie). Ainsi, l'ajustement de  $C$  dans le modèle (1.1) permet de conférer au correcteur des propriétés de filtrage du bruit.

Dans l'équation (1.11), le terme  $E_j B \Delta u(t+j-1)$  est une combinaison linéaire de valeurs définis à des instants d'échantillonnage compris entre les instants  $(t-nb+d)$  et  $(t+j-1-d)$ , ( $d$ :retard) mais la fonction de coût (1.3) implique seulement les valeurs futures et la valeur présente de  $\Delta u$ . Aussi allons-nous chercher à séparer ces valeurs des valeurs passées dans l'équation (1.11). Cette séparation peut être obtenue grâce à la résolution d'une seconde équation Diophantienne en  $G_j$  et  $H_j$  comme suite :

$$E_j B = C G_j + q^{-j} H_j \quad (1.12)$$

Avec :

$$\begin{aligned} G_j &= g_0^j + g_1^j q^{-1} + \dots + g_{j-1}^j q^{-(j-1)} \\ H_j &= h_0^j + h_1^j q^{-1} + \dots + h_{n_h}^j q^{-n_h} \end{aligned} \quad (1.13)$$

Sachant que :  $n_h = \max(n_c, n_b + d) + 1$

L'équation Diophantienne (1.12) peut la résoudre récursivement à l'aide de la méthode rappelée en *annexe*.

L'utilisation de (1.12) dans (1.11) conduit finalement à l'équation suivante:

$$\begin{aligned} \hat{y}(t+j) &= C^{-1} F_j y(t) + G_j \Delta u(t+j-1) + C^{-1} H_j \Delta u(t-1) \\ &= G_j \Delta u(t+j-1) + C^{-1} [H_j \Delta u(t-1) + F_j y(t)] \end{aligned} \quad (1.14)$$

Le premier terme de la somme désigne la réponse « forcée » du système, qui est due aux commandes future, et le second terme représente la réponse « libre ».

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

**I.2.4. Calcul de commande [10-11]:**

Soit  $f$  une matrice de dimension  $(N_2 - N_1 + 1) \times N_u$  constituée de la prédiction de la réponse libre du système pour les instants d'échantillonnage supérieurs à  $t + N_1$  (horizon d'initialisation) et ainsi, inférieurs à  $t + N_2$  (horizon de prédiction) :

$$f = \begin{bmatrix} C^{-1} [H_{N_1} \Delta u(t-1) + F_{N_1} y(t)] \\ C^{-1} [H_{N_1+1} \Delta u(t-1) + F_{N_1+1} y(t)] \\ : \\ : \\ C^{-1} [H_{N_2} \Delta u(t-1) + F_{N_2} y(t)] \end{bmatrix} \quad (1.15)$$

Soit  $\Delta U$  un vecteur de dimension  $N_u \times 1$  contenant les incréments de  $u$  sachant que  $\Delta u(t+j) = 0$  pour  $(j \geq N_u)$  avec :

$$\Delta U = [\Delta u(t) \quad \Delta u(t+1) \quad \dots \quad \Delta u(t + N_u - 1)]^T \quad (1.16)$$

De plus, soit  $\hat{y}$  le vecteur de dimension  $(N_2 - N_1 + 1) \times 1$  contenant les prédictions des sorties à partir de l'horizon d'initialisation défini par:

$$\hat{Y} = [\hat{y}(t + N_1) \quad \hat{y}(t + N_1 + 1) \quad \dots \quad \hat{y}(t + N_2)]^T \quad (1.17)$$

On peut donc écrire pour  $N_1 \leq j \leq N_2$  l'équation de prédicteur sous forme matricielle suivante :

$$\hat{Y} = G \cdot \Delta U + f \quad (1.18)$$

Avec :

$$G = \begin{bmatrix} G_{N_1} & \dots & \dots & \dots & 0 \\ : & : & : & : & : \\ : & : & : & : & G_{N_1} \\ : & : & : & : & : \\ G_{N_2} & \dots & \dots & \dots & G_{N_2} - G_{N_u-1} \end{bmatrix} \quad (1.19)$$

Les termes  $g_k$  avec  $0 \leq k \leq N_2 - 1$  sont les coefficients de la matrice  $G$  (solution de l'équation 1.12).

L'expression du critère (1.3) peut donc s'écrire sous forme matricielle, compte tenu des notations précédentes, il vient :

$$J = (\hat{Y} - W)^T (\hat{Y} - W) + \lambda \cdot \Delta U^T \cdot \Delta U \quad (1.20)$$

Où  $W$  est un vecteur de dimension  $(N_2 - N_1 + 1) \times 1$  contenant les signaux de référence futurs à partir de l'horizon d'initialisation avec :

$$W = [w(t + N_1) \quad w(t + N_1 + 1) \quad \dots \quad w(t + N_2)]^T \quad (1.21)$$

En utilisant les deux équations (1.18) et (1.20) pour réécrire le critère (1.3) sous la forme vectorielle suivante:

$$J = (G \cdot \Delta U + f - W)^T (G \cdot \Delta U + f - W) + \lambda \cdot \Delta U^T \cdot \Delta U \quad (1.22)$$

La loi de commande est obtenue par la minimisation de l'équation (1.22) par rapport à  $\Delta U$  d'où :

$$\Delta U_{opt} = (G^T G + \lambda I)^{-1} G^T (W - f) \quad (1.23)$$

On définit la matrice  $K \in \mathfrak{R}^{N_u \times (N_2 - N_1 + 1)}$  par la relation:

$$K = (G^T G + \lambda I)^{-1} G^T \quad (1.24)$$

La loi de la commande optimale  $\Delta U_{opt}$  peut la trouver sous forme d'un retour avec:

$$\Delta U_{opt} = -K(W - f) \quad (1.25)$$

De plus, seule la première ligne de commande  $\Delta U_{opt}$  est appliquée au système ( la première ligne correspondent au calcul de la commande à l'instant  $t$ , les  $N_u - 1$  autres



correspondants aux instants  $t+1, \dots, t+N_u-1$  quand  $N_u > 1$ ). La commande réellement appliquée au système au temps  $t$  vaut :

$$u(t) = u(t-1) + \Delta u_{opt}(t) \quad (1.26)$$

Avec :

$$\Delta u_{opt}(t) = K_1(W - f) \text{ et } K_1 \text{ représente la première ligne de } (G^T G + \lambda I)^{-1} G^T$$

### I.3. Choix des paramètres de synthèse [12]:

De nombreuses études ont permis aux utilisateurs de la méthode de GPC d'énoncer le choix suivant :

#### a) - Horizons de prédiction :

- l'horizon minimal de prédiction de la sortie du système

Le produit  $N_1 T_e$  ( $T_e$  période d'échantillonnage) est choisi égal au retard pur du système. De plus, pour un système ne présentant pas de retard ou un retard mal connu ou variable,  $N_1$  est choisi égal à 1. Cependant, dans le cas d'un système à déphasage non minimal,  $N_1$  est pris égal à la valeur  $N_0$  indiquée sur la figure (I.3).

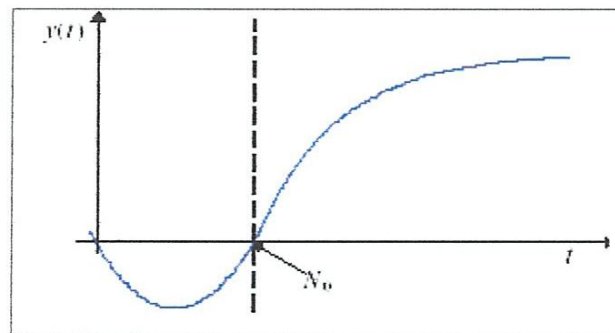
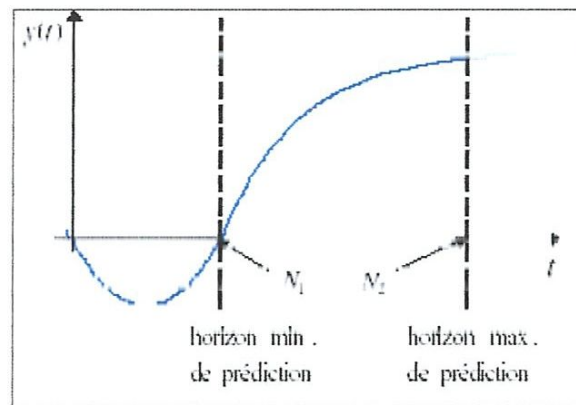


Figure I.3 : Valeur de  $N_1$  pour un système à déphasage non minimal

- **L'horizon maximal de prédiction de la sortie du système**

$N_2$  est choisi de sorte que le produit  $N_2 T_e$  soit limité par la valeur du temps de réponse souhaitée. En effet, augmenter la prédiction au delà du temps de réponse n'apporte aucune information supplémentaire et complexifie la résolution. Par ailleurs, plus  $N_1$  est grand, plus le système corrigé est stable et lent.

Notons sur le schéma de la figure (I.4) le choix de  $N_2$  dans le cas d'un système à déphasage non minimal.



**Figure I.4 :** Choix de  $N_2$  pour un système à déphasage non minimal

**b) Horizon de commande :**

Pour les processus simples, le choix de 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 supérieure à celle de l'horizon de prédiction maximal de sortie.

**c) La pondération :**

Le bon choix de la pondération ( $\lambda = 0$ ) donne généralement des résultats satisfaisants, mais des fois on prend  $\lambda = \text{eps}$  ( $\text{eps}$  très faible) pour améliorer la robustesse.

Si  $(\lambda)$  est suffisamment grand, le système en boucle fermée est instable si le processus en boucle ouverte est instable.

#### I.4. Test de simulation

Dans cette section on présente l'étude en simulation de la méthode d GPC pour commander un système instable linéaire invariant dans le temps. L'objectif est de montrer l'efficacité de cet algorithme notamment pour un système linéaire invariant dans le temps. Le model de CARIMA considéré ici est celui donné par :

$$A(q^{-1}) = 1 - 0.72q^{-1}, \quad B(q^{-1}) = 1.5 + 0.32q^{-1}$$

Le polynôme de filtrage  $C(q^{-1})$  est supposé égal à 1. L'entrée de référence est

$$\text{supposée ici un signal carre définie par : } W(t) = \begin{cases} 0 : 1 \leq t \leq 25 \\ 2 : 25 \leq t \leq 75 \\ 0 : 75 \leq t \leq 125 \\ 1 : 125 \leq t \leq 150 \end{cases}$$

L'application de la méthode GPC sur le système ci-dessus donne les résultats donnés par les figures suivantes:

- La figure (I.5) représente la dynamique de poursuite du système bouclé pour des différents choix des paramètres de réglage.
- La figure (I.6) est un agrandissement (zoom) des réponses présentées dans la figure précédente.
- La figure (I.7), (I.8) et (I.9) représentent les différents signaux de commandes fournies par la loi de GPC pour des différents choix des paramètres de réglages  $N_u, N_2$  et  $\lambda$

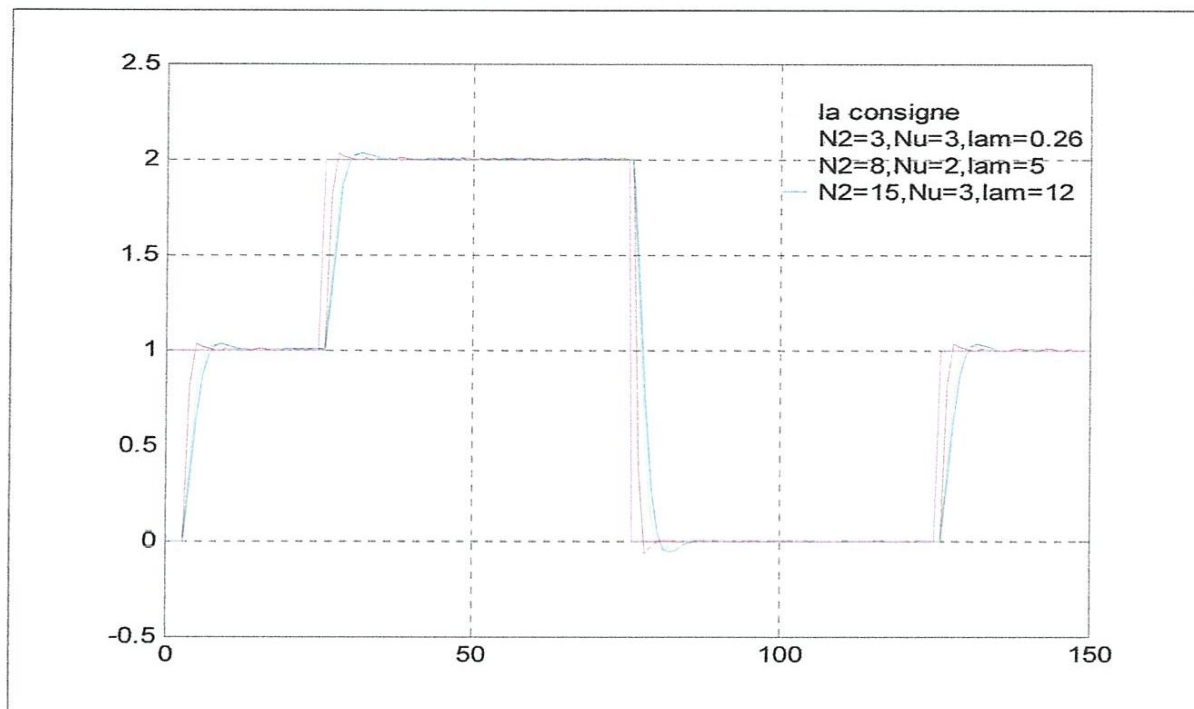


Figure I.5. : Les sorties obtenues par la méthode GPC

D'après la figure (I.5) on peut constater que la dynamique de poursuite dépend directement par un bon choix des paramètres de réglage de la méthode de GPC. La bonne dynamique est jugée par d'une erreur statique presque nulle avec un temps de montée très court.

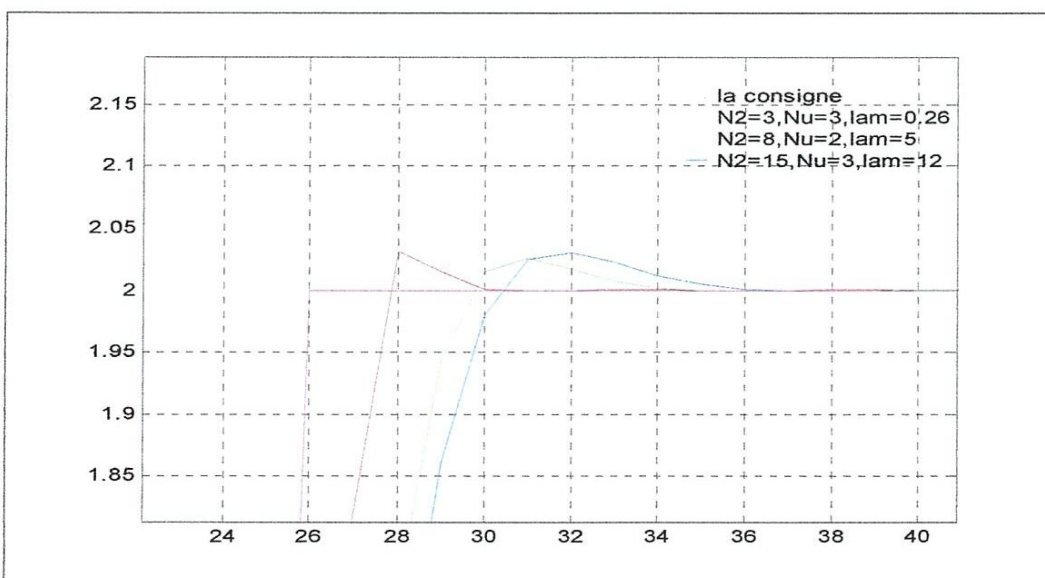
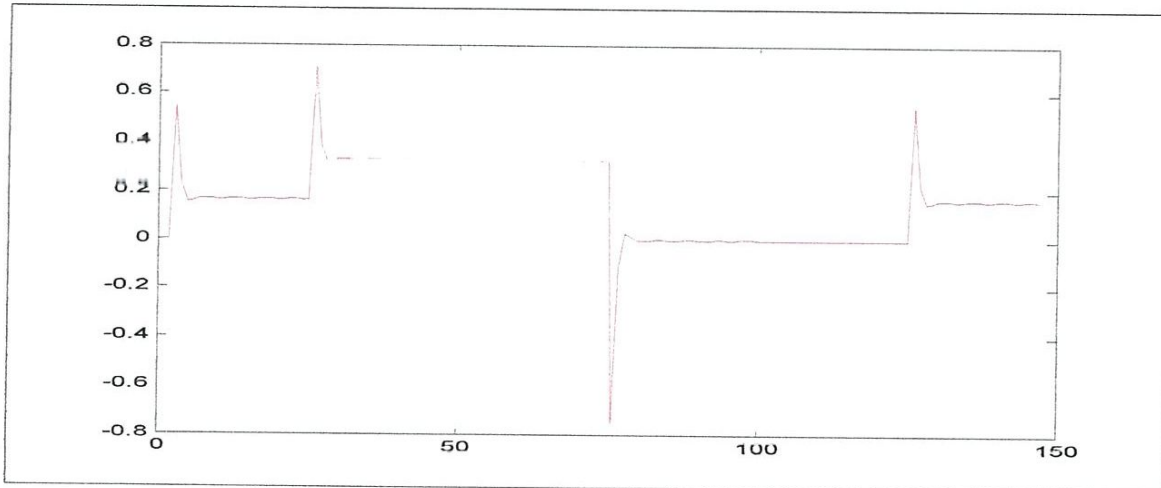


Figure I.6. Agrandissement d'une partie de la sortie

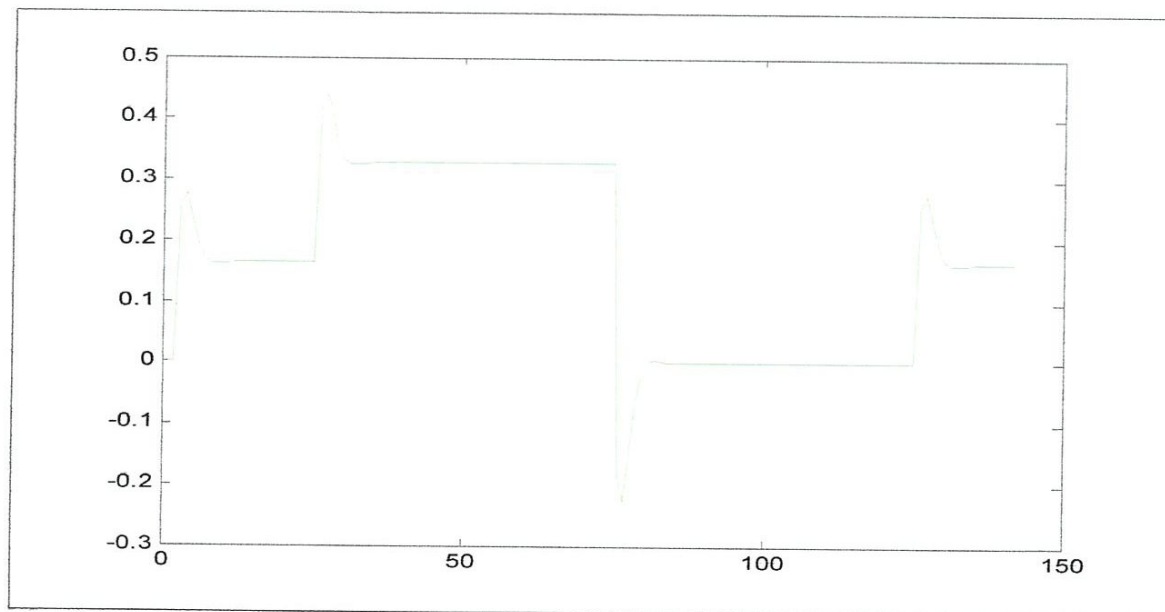
D'après la figure (I.6) on peut constater les remarques suivantes :

- les performances obtenues sont variées en fonction des paramètres de synthèse  $N_u, N_2$  et  $\lambda$ .
- Plus que  $N_2$  devient grand, plus que le système corrigé devient lent.

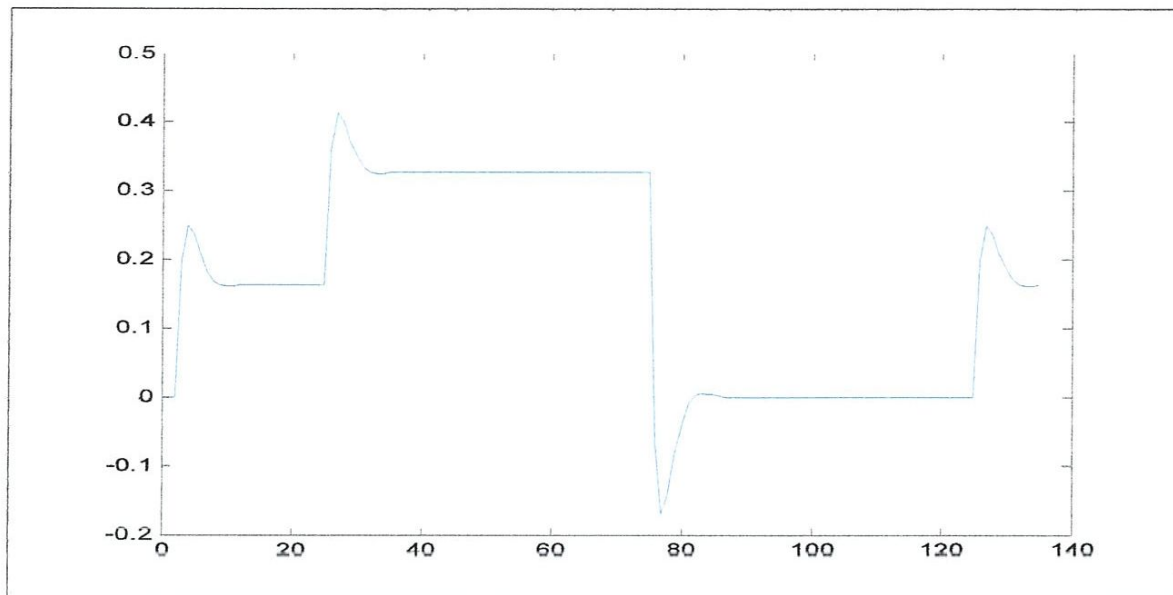
Pour les signaux de commande on obtient les résultats suivants :



**Figure I.7.** : Commande fournie par la loi de GPC pour ( $\lambda = 0.26, N_u = 3, N_2 = 3$ )



**Figure I.8.** : Commande fournie par la méthode GPC ( $\lambda = 5, N_u = 2, N_2 = 8$ )



**Figure I.9.** : Commande fournie par la méthode GPC ( $\lambda = 12, N_u = 3, N_2 = 15$ )

D'après les figures précédentes, on peut constater :

- plus la valeur de  $\lambda$  devient petite, la loi de GPC offre une commande très économique.
- Pour toutes valeurs de  $\lambda$ , on obtient des commandes stables.

### I.5. Conclusion

Dans ce chapitre, nous avons présenté la démarche de la commande prédictive généralisée mono-variable sans contraintes qui donne des résultats de simulation très encourageants, ainsi que les performances obtenus sont liées directement par les paramètres de synthèse ( $N_2, N_u, \lambda$ ).

CHAPITRE II  
GPC SOUS CONTRAINTES

## II.1. 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 fondamental 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 les erreurs stationnaires 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 à savoir, l'horizon de prédiction de sortie et de commande ainsi le coefficient de pondération qui pénalise les signaux de commandes. A cet effet, le choix d'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, à retards et/ou les systèmes instables. Comme le soulignent les études effectuées par [13]. 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 [14], permettant l'application de la fonction « *quadprog* » (Quadratic Programming), devenue très populaire et qui offre même aujourd'hui des solutions intéressantes avec un avantage considérable concernant la facilité de traitement du cas robuste.

Dans ce qui va suivre, nous allons exposer la formulation du problème de la loi de GPC en présence des contraintes à savoir sur les incréments de commandes, sur les commandes ainsi sur la sortie du système à commander. Ces contraintes sont ensuite reformulées sous forme matricielle et la fonction *Quadprog*, qui est disponible dans le *Toolbox/Matlab*, offre la possibilité de développer une loi de commande qui assure une bonne minimisation du critère avec une bonne satisfaction de toutes les contraintes imposées au préalable par le constructeur. Des résultats de simulation sont ensuite insérés à la fin de ce chapitre afin de confirmer l'efficacité de cette méthode d'optimisation sous contraintes.

## II.2. Les contraintes [15-18]

Sachant que le critère, à minimiser, est donné par :

$$J = (GU + f - W)^T (GU + f - W) + \lambda U^T U \quad (2.1)$$

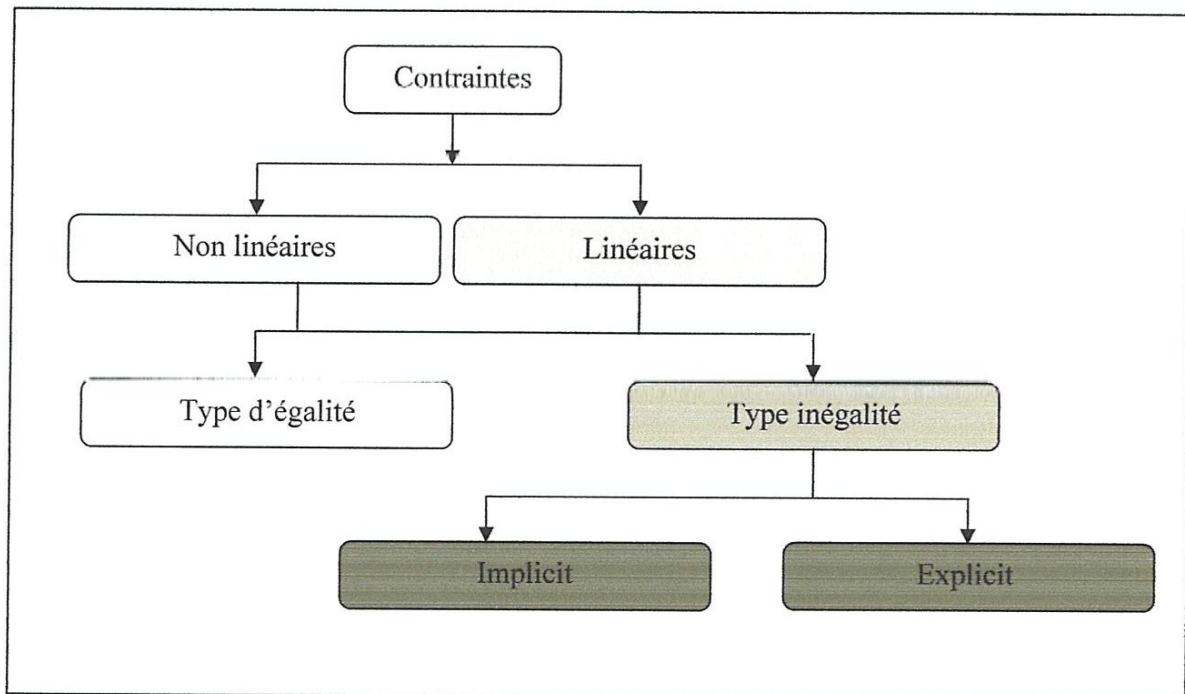
Une première conclusion pour la formulation GPC dérivée de l'équation (2.1) est que la fonction de coût GPC possède une forme quadratique [19] donnée sous la forme générale suivante :

$$J = \frac{1}{2} U^T Q_2 U + Q_1^T U + Q_0 \quad (2.2)$$

$$\text{Avec : } \begin{cases} Q_2 = 2(G^T G + \lambda I) \\ Q_1^T = 2(f - W)^T G \\ Q_0 = (f - W)^T (f - W) = \text{constant} \end{cases}$$

Le critère (2.2) est quadratique en  $U$ , 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 suivant [20]:



**Figure 2.1.** Classification des contraintes

### II.3. Formulation des contraintes [21-22]:

Les contraintes qui interviennent dans la formulation de la loi de 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 de les exprimer sous la forme générale suivante :

$$\begin{aligned} v_{\min} &\leq v(t+j) \leq v_{\max} \\ N_{\min} &\leq j \leq N_{\max} \end{aligned} \quad (2.3)$$

Avec  $v(t+j)$  représente le signal contraint,  $(v_{\min}, v_{\max})$  sont les limitations à respecter,  $(N_{\min}, N_{\max})$  représentent les horizons 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 de GPC, les paragraphes suivants proposent quelques formalismes de contraintes très souvent rencontrés en pratique. Parmi lesquels on peut citer les suivants :

### II.3.1. Contraintes sur les incréments de commandes

Ces contraintes sont appelées « contraintes de saturations ». Dans ce travail, on s'intéresse ici par les contraintes linéaires d'inégalité explicite. Ces dernières sont données sous forme[22]:

$$\begin{aligned} \Delta u_{\min} &\leq \Delta u(t+j) \leq \Delta u_{\max} \\ N_{\min} &\leq j \leq N_{\max} \end{aligned} \quad (2.4)$$

L'expression matricielle pour ce type de contraintes, en particulier les horizons de prédiction de commande  $N_{\min} = 0$  et  $N_{\max} = N_u - 1$ , est donnée comme suit:

$$\begin{aligned} \Delta u_{\min} &\leq \Delta u(t+j) \leq \Delta u_{\max} \\ N_{\min} &\leq j \leq N_{\max} \end{aligned} \quad (2.5)$$

On obtient :

$$\begin{bmatrix} \Delta u_{\min} \\ \Delta u_{\min} \\ \vdots \\ \Delta u_{\min} \end{bmatrix} \leq \begin{bmatrix} \Delta u(t+0) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N_u-1) \end{bmatrix} \leq \begin{bmatrix} \Delta u_{\max} \\ \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix}$$

D'où :

$$\left\{ \begin{array}{l} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \Delta U \leq \begin{bmatrix} \Delta u_{\max} \\ \Delta u_{\max} \\ \vdots \\ \Delta u_{\max} \end{bmatrix} \\ \\ \begin{bmatrix} -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta u_{\min} \\ -\Delta u_{\min} \\ \vdots \\ -\Delta u_{\min} \end{bmatrix} \end{array} \right.$$

### II.3.2. Contraintes sur les amplitudes de commandes:

On s'intéresse ici par les contraintes linéaires d'inégalité implicite où ces dernières sont imposées comme suit :

$$\begin{aligned} u_{\min} &\leq u(t+j) \leq u_{\max} \\ N_{\min} &\leq j \leq N_{\max} \end{aligned} \quad (2.6)$$

L'expression matricielle pour ce type de contraintes, en particulier les horizons de prédiction de commande  $N_{\min} = 0$  et  $N_{\max} = N_u - 1$ , est donnée comme suit:

$$\begin{bmatrix} u_{\min} \\ u_{\min} \\ \vdots \\ u_{\min} \end{bmatrix} \leq \begin{bmatrix} u(t+0) \\ u(t+1) \\ \vdots \\ u(t+N_u-1) \end{bmatrix} \leq \begin{bmatrix} u_{\max} \\ u_{\max} \\ \vdots \\ u_{\max} \end{bmatrix} .$$

Tenant-compte l'expression suivante :

$$\begin{aligned} u(t+j) &= u(t-1) + \sum_{k=0}^j \Delta u(t+k) \\ j &= 0, \dots, N_u - 1 \end{aligned} \quad (2.7)$$

On obtient :

$$\left\{ \begin{array}{l} \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \Delta U \leq \begin{bmatrix} -u(t-1) + u_{\max} \\ -u(t-1) + u_{\max} \\ \vdots \\ -u(t-1) + u_{\max} \end{bmatrix} \\ \\ \begin{bmatrix} -1 & 0 & \dots & 0 \\ -1 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -1 & -1 & \dots & -1 \end{bmatrix} \Delta U \leq \begin{bmatrix} u(t-1) - u_{\min} \\ u(t-1) - u_{\min} \\ \vdots \\ u(t-1) - u_{\min} \end{bmatrix} \end{array} \right. \quad (2.8)$$

### II.3.3. Contraintes sur les signaux de sorties:

On s'intéresse ici par les contraintes linéaires d'inégalité implicite [22] :

$$\begin{array}{l} y_{\min} \leq y(t+j) \leq y_{\max} \\ N_{\min} \leq j \leq N_{\max} \end{array} \quad (2.9)$$

L'expression matricielle pour ce type de contraintes, en particulier les horizons de prédiction de sortie  $N_{\min} = N_1 - 1$  et  $N_{\max} = N_2 + N_1 - 1$ , est donnée comme suit:

$$\begin{bmatrix} y_{\min} \\ y_{\min} \\ \vdots \\ y_{\min} \end{bmatrix} \leq \begin{bmatrix} y(t+j) \\ y(t+j) \\ \vdots \\ y(t+j) \end{bmatrix} \leq \begin{bmatrix} y_{\max} \\ y_{\max} \\ \vdots \\ y_{\max} \end{bmatrix}$$

En utilisant la relation (1.18) (i.e.  $\hat{Y} = G \cdot \Delta U + f$ ) d'où l'expression (2.9) devient :

$$\left\{ \begin{array}{l} G \cdot \Delta U \leq \begin{bmatrix} -f(t + N_1 - 1) + y_{\max} \\ -f(t + N_1) + y_{\max} \\ \vdots \\ -f(t + N_2 + N_1 - 1) + y_{\max} \end{bmatrix} \\ -G \cdot \Delta U \leq \begin{bmatrix} f(t + N_1 - 1) - y_{\min} \\ f(t + N_1) - y_{\min} \\ \vdots \\ f(t + N_2 + N_1 - 1) - y_{\min} \end{bmatrix} \end{array} \right. \quad (2.10)$$

### II.3.4. 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  $N_s$  l'horizon  $N_2$ . Ces contraintes peuvent prendre la structure [22] :

$$\begin{aligned} y(t + j) &= w(t + j) \\ j &= N_2 + 1, \dots, N_s \end{aligned} \quad (2.11)$$

Il s'agit encore une fois de contraintes linéaires d'égalité implicite. Le formalisme (2.11) peut être exprimé comme un ensemble de contraintes d'égalité sur les incréments de commandes future, en utilisant l'expression suivante :

$$\begin{cases} G_{N_s} \cdot \Delta U = y_{\max_{N_s}} - f \\ -G_{N_s} \cdot \Delta U = -y_{\min_{N_s}} + f \end{cases} \quad (2.12)$$

### II.3.5. Autres types de contraintes [23-24]:

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  $\Delta U$ . Toutes les limitations ont été représentées par des contraintes linéaires, comme on peut le constater à partir des relations (2.4), (2.6), (2.9) et (2.11).

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  $\Delta U$ . En particulier ce vecteur qui demeure l'objectif du processus de décision, l'ensemble de toutes les contraintes peut s'écrire sous forme compacte suivante :

$$\begin{cases} A_{inq} \cdot \Delta U = B_{inq} \\ A_{eq} \cdot \Delta U = B_{eq} \end{cases} \quad (2.13)$$

Où :

$$\begin{cases} A_{inq} = \left[ \begin{array}{cccc} [1 & 0 & \dots & 0] \\ [0 & 1 & \dots & 0] \\ \vdots & \vdots & \vdots & \vdots \\ [0 & 0 & \dots & 1] \end{array} \begin{array}{cccc} [-1 & 0 & \dots & 0] \\ [0 & -1 & \dots & 0] \\ \vdots & \vdots & \vdots & \vdots \\ [0 & 0 & \dots & -1] \end{array} \begin{array}{cccc} [1 & 0 & \dots & 0] \\ [1 & 1 & \dots & 0] \\ \vdots & \vdots & \vdots & \vdots \\ [1 & 1 & \dots & 1] \end{array} \begin{array}{cccc} [-1 & 0 & \dots & 0] \\ [-1 & -1 & \dots & 0] \\ \vdots & \vdots & \vdots & \vdots \\ [-1 & -1 & \dots & -1] \end{array} \right]^T, G_s, -G \\ B_{inq} = \left[ \begin{array}{cccc} [\Delta u_{\max}] & [-\Delta u_{\min}] & [-u(t-1) + u_{\max}] & [u(t-1) - u_{\min}] & [-f(t+N_1-1) + y_{\max}] & [f(t+N_1-1) - y_{\min}] \\ [\Delta u_{\max}] & [-\Delta u_{\min}] & [-u(t-1) + u_{\max}] & [u(t-1) - u_{\min}] & [-f(t+N_1) + y_{\max}] & [f(t+N_1) - y_{\min}] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ [\Delta u_{\max}] & [-\Delta u_{\min}] & [-u(t-1) + u_{\max}] & [u(t-1) - u_{\min}] & [-f(t+N_2+N_1-1) + y_{\max}] & [f(t+N_2+N_1-1) - y_{\min}] \end{array} \right]^T \\ A_{eq} = [G_{N_s}, -G_{N_s}]^T \\ B_{eq} = [y_{\max_{N_s}} - f, -y_{\min_{N_s}} + f]^T \end{cases}$$

Finalement, en regroupant la formulation du critère de performance (2.2) avec l'ensemble des contraintes (2.13), on obtient le problème de la loi de GPC sous contraintes suivant :

$$\text{Problème : } \begin{cases} \text{Min}_{\Delta U}(J) = \text{Min}_{\Delta U} \left\{ \frac{1}{2} U^T Q_2 U + Q_1^T U + Q_0 \right\} \\ \text{sujet à :} \\ \begin{cases} A_{inq} \cdot \Delta U = B_{inq} \\ A_{eq} \cdot \Delta U = B_{eq} \end{cases} \end{cases} \quad (2.14)$$

Le problème (2.14) 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 suivante:

$$\Delta U_{opt} = \text{quadprog}(Q_2, Q_1, A_{inq}, B_{inq}, A_{eq}, B_{eq}) \quad (2.15)$$

Dans ce travail, on ne considère que les contraintes linéaires d'inégalité, d'où l'expression (2.15) devient comme suit :

$$\Delta U_{opt} = \text{quadprog}(Q_2, Q_1, A_{inq}, B_{inq}) \quad (2.16)$$

Une fois  $\Delta U_{opt}$  obtenue, on applique le principe de l'horizon glissant et seule la première ligne de  $\Delta U_{opt}$  est appliquée au système à commander.

#### II.4. Resultats de simulation

Dans cette partie, on désire appliquer la loi de GPC en présence des contraintes sur un système linéaire invariant dans le temps. Ce dernier a été modélisé par un modèle mathématique de CARIMA suivant :

$$\begin{cases} A(q^{-1}) = 1 - 1.6003q^{-1} + 0.789q^{-2}. \\ B(q^{-1}) = 0.1012 + 0.0852q^{-1}. \end{cases}$$

Pour les contraintes à satisfaire par la fonction *quadprog*, on considère les contraintes sur le signal de sortie comme suit :



$$-0.25 \leq y(t+j) \leq 2.3 \text{ où } j = 0, \dots, N_2 - 1.$$

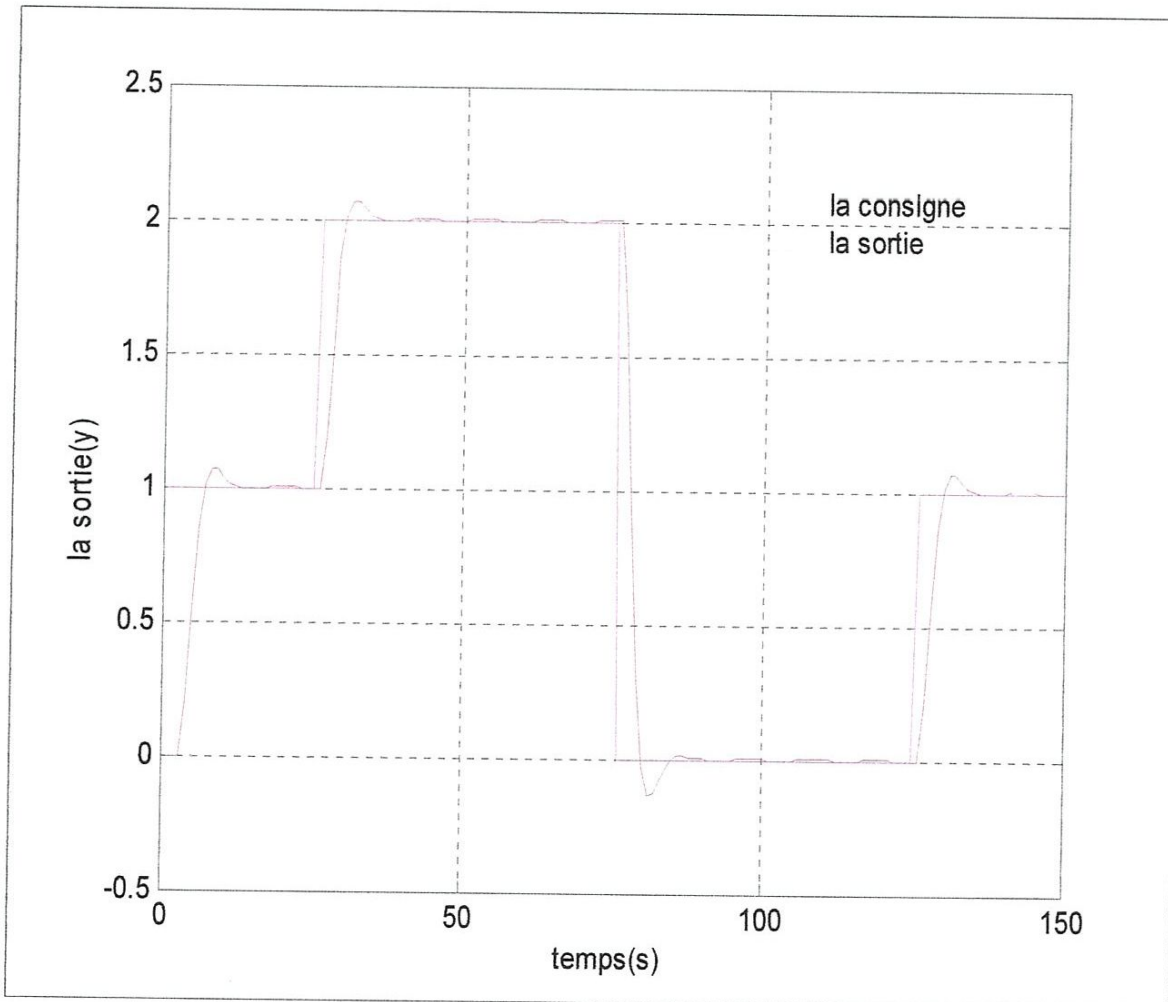
De plus, la contrainte sur la commande est donnée par :

$$-5 \leq u(t+j) \leq 5 \text{ où } j = 0, \dots, N_u - 1.$$

Les paramètres de réglage de la méthode GPC sont fixés comme suit :

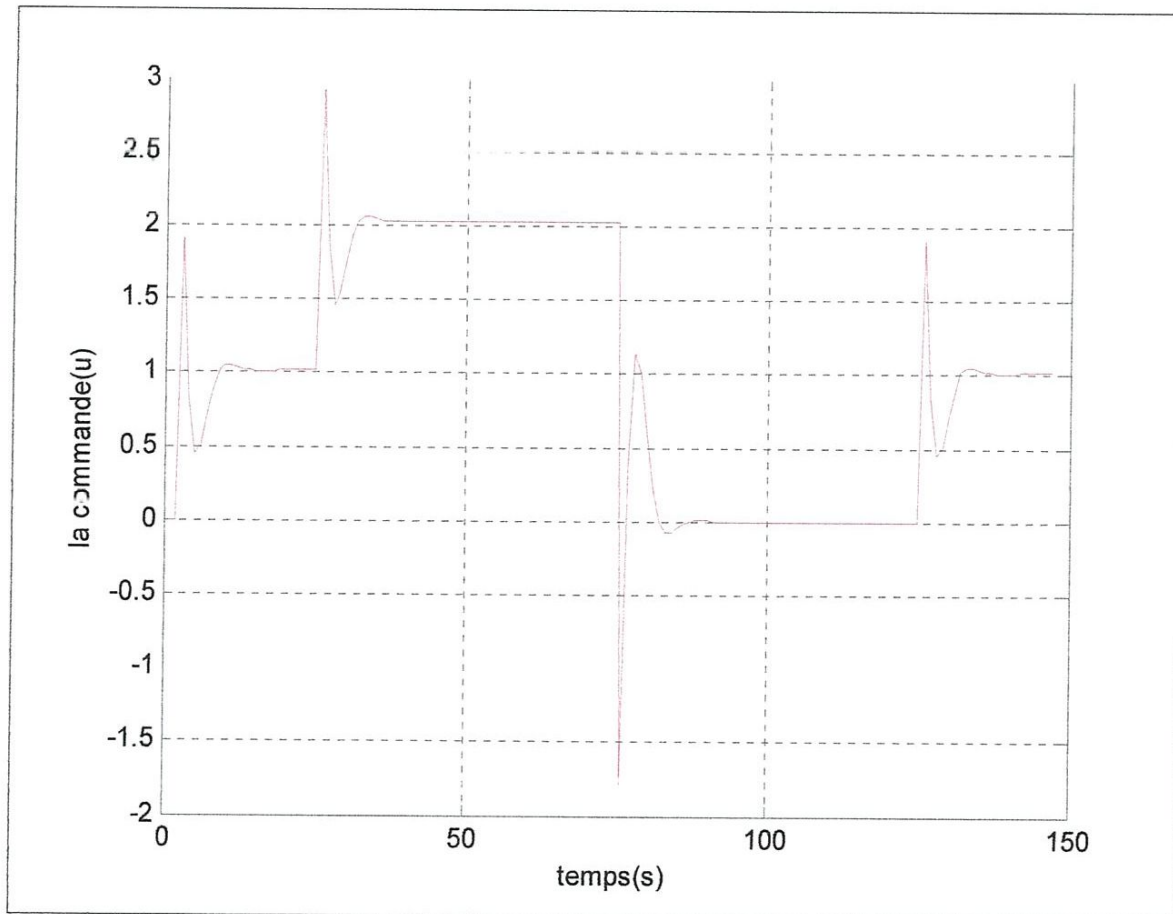
$$\begin{cases} N_2 = 5 \\ N_u = 2 \\ \lambda = 0.02 \end{cases}$$

L'application de la fonction Quadprog pour chaque pas de discrétisation donne les résultats présentés par les figures suivantes.



**Figure 2.2.** Dynamique de poursuite fournie par la loi de GPC sous contraintes.

D'après cette figure, on a pu constater que toutes les contraintes, imposées par l'utilisateur, sont très satisfaisantes. de plus, la loi de GPC assure une bonne dynamique d poursuite de la trajectoire de référence. La qualité de cette dynamique est interprétée par un meilleur temps de réponse, temps de montée très rapide et une erreur statique presque nulle. Maintenant, pour l'effort d commande fourni par cette loi, on obtient :



**Figure 2.3.** Commande fournie par la loi d GPC en présence des contraintes

**II.5. Conclusion :**

Dans ce chapitre, nous avons présenté la commande prédictive généralisée en présence des contraintes. Ces contraintes sont classées suivant la nature exigée par l'utilisateur.

Ce type de commande peut être appliqué à des procédés difficiles à commander, à savoir, instable en boucle ouverte ou /et à phase minimale. Ceci grâce à un choix particulier des paramètres de réglage de GPC à savoir l'horizon de prédiction maximal du signal de sortie ( $N_y$ ), l'horizon de prédiction de commande ( $N_u$ ) et finalement, le coefficient de pondération  $\lambda$ .

Des résultats de simulation obtenus via l'application de la loi de GPC sous contraintes confirment que cette méthode de commande assure une bonne dynamique de poursuite par une commande très satisfaisante de point de vue pratique.

CHAPITRE III  
APPLICATION SUR UN  
SYSTEME NON LINEAIRE

### III.1. Introduction :

Dans toute industrie, nous obtenons des produits finis en associant plusieurs techniques qui agissent simultanément à l'amélioration de notre produit, cela nous oblige à perfectionner une bonne méthode de commande; l'automatique fait des progrès immenses dans ce sens.

La plupart des processus réels, pour ne pas dire tous, décrivent par des dynamiques non linéaires peuvent poser de sérieuses difficultés pour la synthèse d'un contrôleur, puisque la plupart des méthodes de commande les plus avancées opèrent avec un modèle linéaire. Parmi les méthodes les plus récentes et les plus performantes on trouve la méthode de GPC qui nécessite un modèle linéaire afin de développer une loi de commande satisfaisante [25-28].

Le domaine d'exploitation de cette méthode de commande dans l'industrie est étroit à cause de non linéarité des processus à commander. L'avancée de la recherche a donné naissance à des méthodes de modélisation permettant de linéariser le système à commander autour d'un ou plusieurs points de fonctionnement. Parmi les plus utilisées on cite: la modélisation à base d'un modèle flou, modélisation par un réseau de neurone...etc. Leurs objectifs sont de représenter un processus non linéaire par un ensemble de modèles linéaires pour pouvoir appliquer la méthode de GPC dans tous les domaines industriels [29].

La première phase d'une modélisation consiste à rassembler les connaissances dont on dispose sur le comportement du processus, à partir d'expériences et/ou d'une analyse théorique des phénomènes physiques mis en jeu. Ces connaissances conduisent à faire plusieurs hypothèses de modèles. Chacun de ces modèles dynamiques, appelés *modèle-hypothèse*, réalise des fonctions non linéaires entre ses variables de commande. Dans le cas où ces fonctions sont inconnues, on parle de modèle boîte noire. Si certaines fonctions peuvent être fixées à partir de

L'analyse théorique, on parle alors de modèle de connaissance. L'approche boîte noire utilise, le plus souvent, des modèles *entrée-sortie*, alors qu'une représentation d'état est adoptée pour les modèles de connaissance [30].

La seconde phase de modélisation, aussi appelée identification, consiste à estimer les paramètres des modèles concurrents. Pour chacune des deux phases citée précédemment, on met en œuvre un système d'apprentissage constitué de la forme prédicateur du modèle-hypothèse, et un algorithme d'estimation est effectué afin d'estimer les paramètres de ce modèle. Cette estimation est achevée en minimisant une fonction de coût définie à partir de l'écart entre la sortie mesurée à partir du processus réel et les valeurs prédites obtenues via le modèle de synthèse (erreur de prédiction). La qualité de cette estimation dépend de la richesse des séquences d'apprentissage et l'efficacité de l'algorithme utilisé.

Pour concrétiser les notions théoriques déjà vue précédemment, nous allons appliquer la loi de GPC sous contraintes sur un processus industriel qui représente ici un réacteur chimique exothermique continu agité CSTR (Continued Stirred Tank Reactor CSTR). Sa dynamique non linéaire est modélisée par un modèle flou de Takagi Sugino (TS) en utilisant la version on-line dans la phase d'identification ainsi la phase de commande.

## **III.2 Le réacteur exothermique continu agité (CSTR):**

### **III.2.1 Description du procédé:**

Le réacteur considéré ici est un réacteur exothermique continu agité, sa dynamique est décrite par les équations différentielles non linéaires suivantes [31-36] :

$$\begin{aligned} \dot{C}_a(t) &= \frac{q_c(t)}{v} (C_{a_0} - C_a(t)) - K_0 \cdot C_a(t) \cdot e^{\frac{E}{R \cdot T(t)}} \\ \dot{T}(t) &= \frac{q_c(t)}{v} (T_0 - T(t)) - K_1 \cdot C_a(t) \cdot e^{\frac{E}{R \cdot T(t)}} + K_2 \cdot q_c(t) \cdot \left(1 - e^{\frac{K_3}{q_c(t)}}\right) \cdot (T_{c_0} - T(t)) \end{aligned} \quad (3.1)$$

Le procédé représente une réaction exothermique irréversible,  $A \leftrightarrow B$ , pour un volume constant, le refroidissement de ce réacteur est fait par la circulation de vapeur d'eau dans l'enveloppe (le serpent), la concentration du produit notée par ' $C_a(t)$ ' est contrôlée par la manipulation sur le débit du refroidissant ' $q_c(t)$ '. La température du mélange notée par  $T(t)$  représente la chaleur générée par la réaction, elle fait aussi ralentir la réaction.  $C_{a_0}$  : c'est la concentration initiale et ' $q_c(t)$ ' représente le taux de débit du processus,  $T_0$  et  $T_{c_0}$  sont les températures initiales d'alimentation et de refroidissement à l'entrée du réacteur. Les paramètres  $K_0, \frac{E}{R}, K_1, K_2$  et  $K_3$  sont des constantes chimiques et thermodynamiques données par les relation suivantes:

$$K_1 = \frac{\Delta H K_0}{p C_p}, \quad K_2 = \frac{p c \cdot C_{pc}}{p \cdot C_p \cdot v}, \quad K_3 = \frac{h a}{p c C_{pc}} \quad (3.2)$$

Les conditions nominales pour un produit de concentration de  $C_a(t) = 0.1 \text{ mol/L}$  sont :  $T = 438.54 \text{ K}$  et  $q_c = 103.41 \text{ L/min}$ . Les valeurs numériques de ces constantes sont données par le tableau 3.1 [L197]. On obtient :

paramètre	Description	Les valeurs nominales
$q$	Débit d'entre de produits	100 l/min
$v$	Volume de réaction	100 l
$K_0$	Vitesse de réaction	$7.2 \times 10^{10} \text{ min}^{-1}$
$E/R$	Energie d'activation à l'entrée	$1 \times 10^4 \text{ k}$
$T_0$	Température du produit à l'entrée	350 k
$T_{c0}$	Température de refroidissant	350 k
$\Delta H$	Chaleur de la réaction	$-2 \times 10^5 \text{ cal/mol}$
$C_p, C_{pc}$	Chalcura spécifiques	1 cal/g/k
$\rho, \rho_c$	Densités du liquide	$1 \times 10^3 \text{ g/l}$
$h_a$	Coefficient du transfert de chaleur	$7 \times 10^5 \text{ cal/min/k}$
$C_{a0}$	Concentration du produit à l'entrée	1 mol/l

Tab.3.1 : Les constantes du CSTR

### III.3 Simulation :

La simulation du *CSTR* est effectuée par la résolution de deux équations différentielles citées précédemment par l'équation (3.1) en utilisant la méthode de *Rang Kutta* d'ordre 04 et un pas de discrétisation  $dt = 6$  secondes.

La grandeur de commande  $q_c$  est injectée d'une façon aléatoire dans l'intervalle  $q_c \in [1.5091 \ 1.838]$ . Cette dernière donne des mesures entre l'intervalle  $C_o \in [0.065 \ 0.17]$ .

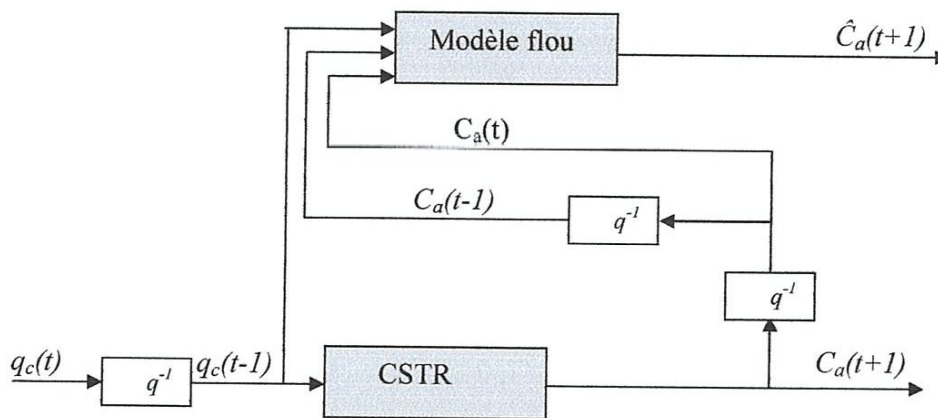
Une séquence de 9000 échantillons a été produite, les 7500 premiers échantillons ont été utilisés pour l'entraînement et les 2500 restantes sont utilisés pour la validation du modèle flou.



Le modèle flou de type T-S a été utilisé comme un modèle de prédiction, il a été construit à partir des ensembles d'entrées/sorties (E/S) du processus. Ses fonctions d'appartenances et ses règles floues sont déterminées en utilisant la fonction *genfis2* qui est disponible dans le Toolbox de Matlab, en utilisant un rayon de voisinage (*radii*)  $\rho=0.32$ .

Les paramètres du modèle de T-S sont obtenus, pour chaque pas de discrétisation, à partir de ses sorties polynomiales. La structure de ce dernier est choisie sous la forme donnée par Fig.3.1 afin de réaliser la fonction suivante:

$$\hat{C}_a(t+1) = f(C_a(t), C_a(t-1), q_c(t-1)) \quad (3.3)$$



**Figure3.1.** Choix de modèle de T-S

La figure 3.2 présente le signal d'entrée utilisé pour la phase d'identification. Pour la validation du modèle de T-S, on a utilisé les 2500 échantillons restants, d'où on obtient les figures 3.3 et 3.4 qui représentent, respectivement, les signaux de sorties fournis par le processus réel et le modèle de T-S. De plus, la comparaison de deux sorties est présentée par la figure 3.5. Cette comparaison donne la possibilité de tracer le signal d'erreur de modélisation qui est présenté par la figure 3.6. Sa valeur est obtenue via l'expression (3.4) avec :

$$\xi_R = \frac{1}{3000} \sum_{t=7501}^{9000} [Ca_m(t) - Ca(t)]^2 = 4.2094 \times 10^{-8} \quad (3.4)$$

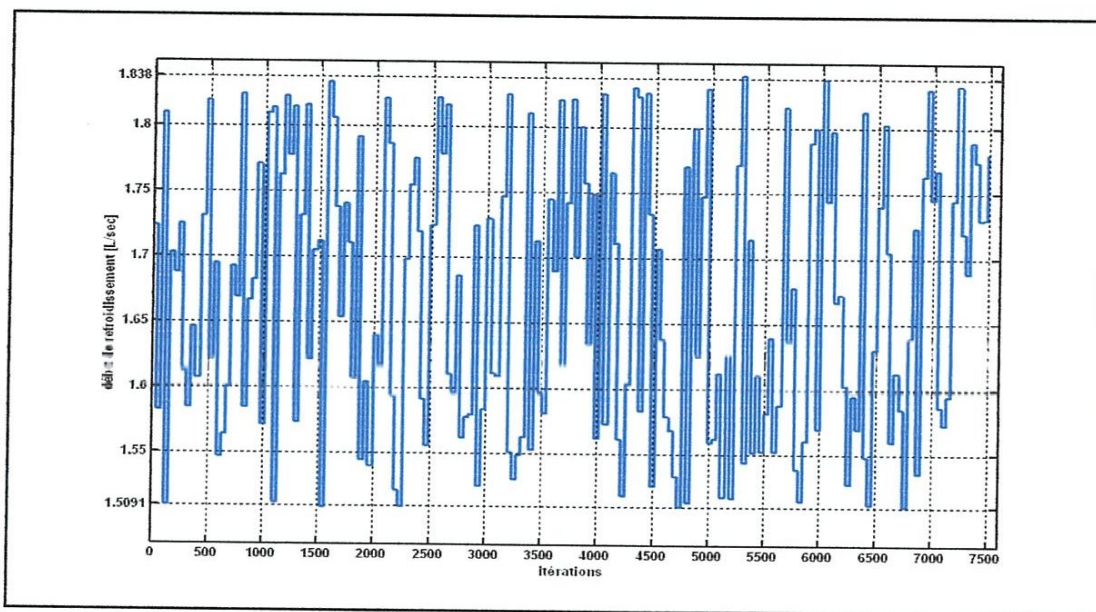


Figure 3.2: Signal d'entrée pour l'identification

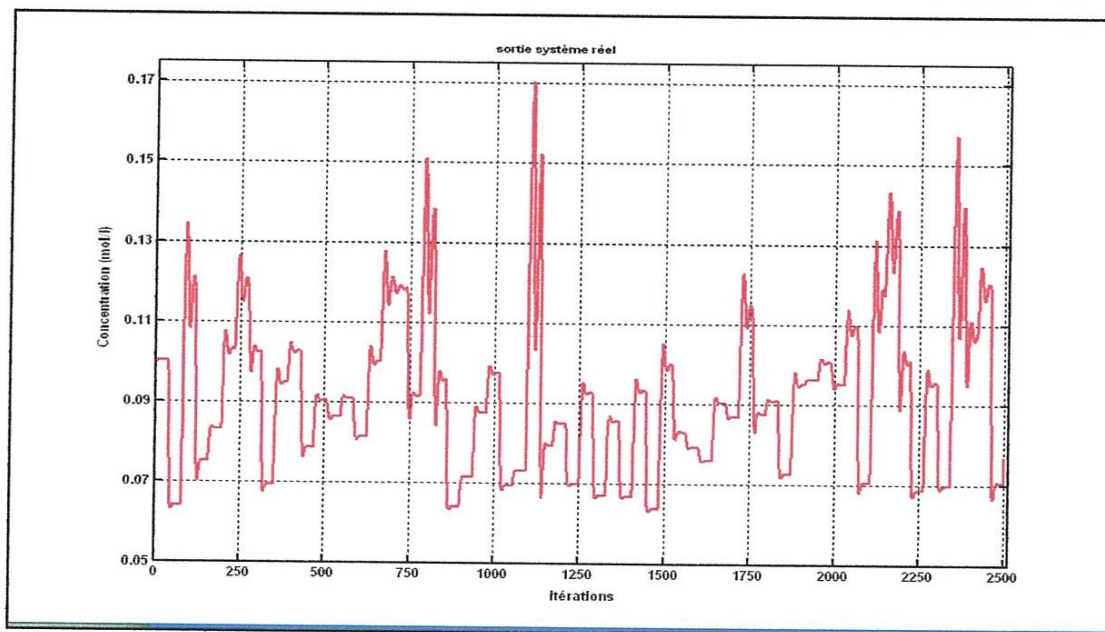


Figure 3.3: Signal de sortie du système réel lors la phase de validation

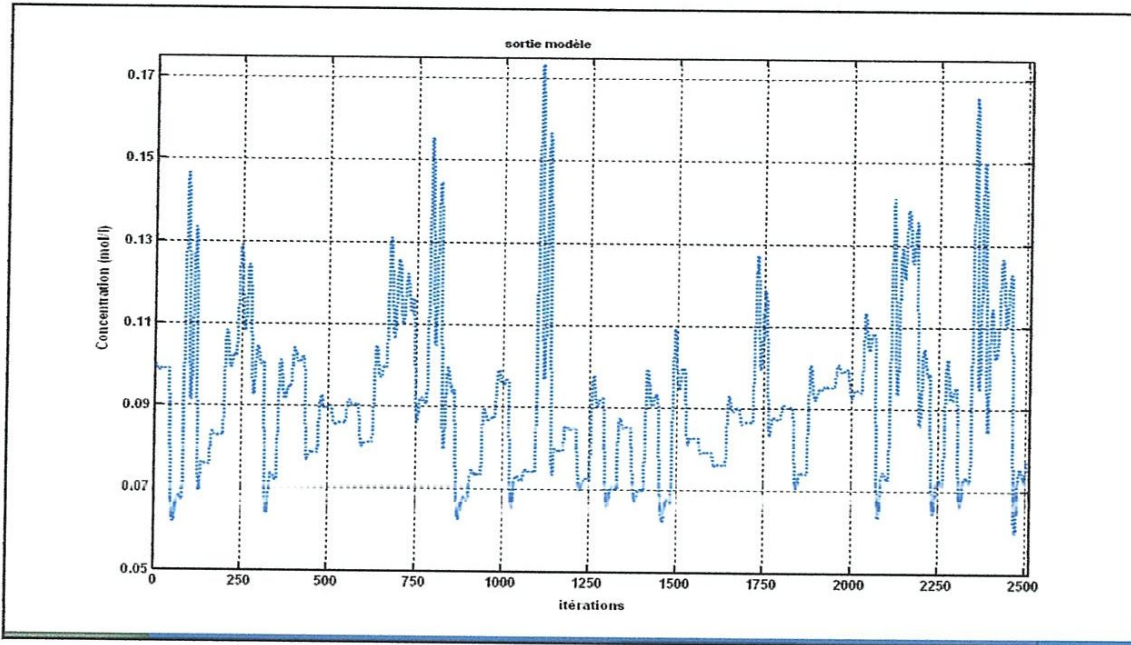


Figure3.4 : Sortie fournie par le modèle flou de T-3 lors la phase de validation

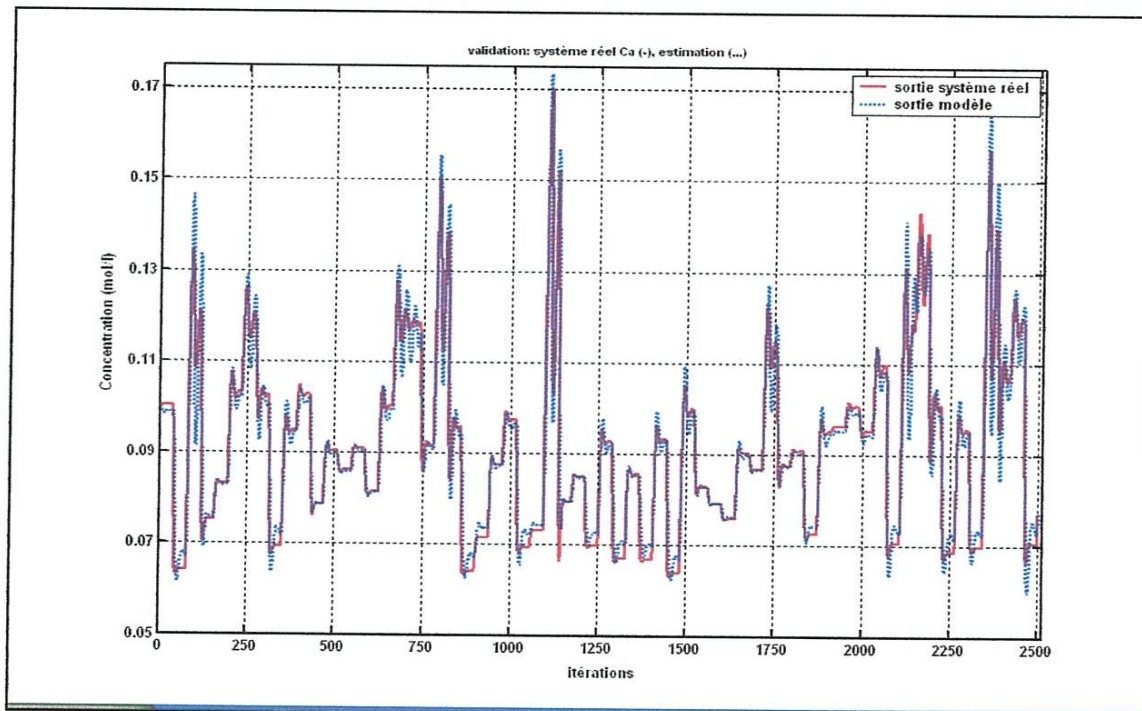


Figure3.5: Comparaison de deux sorties : réel et modèle

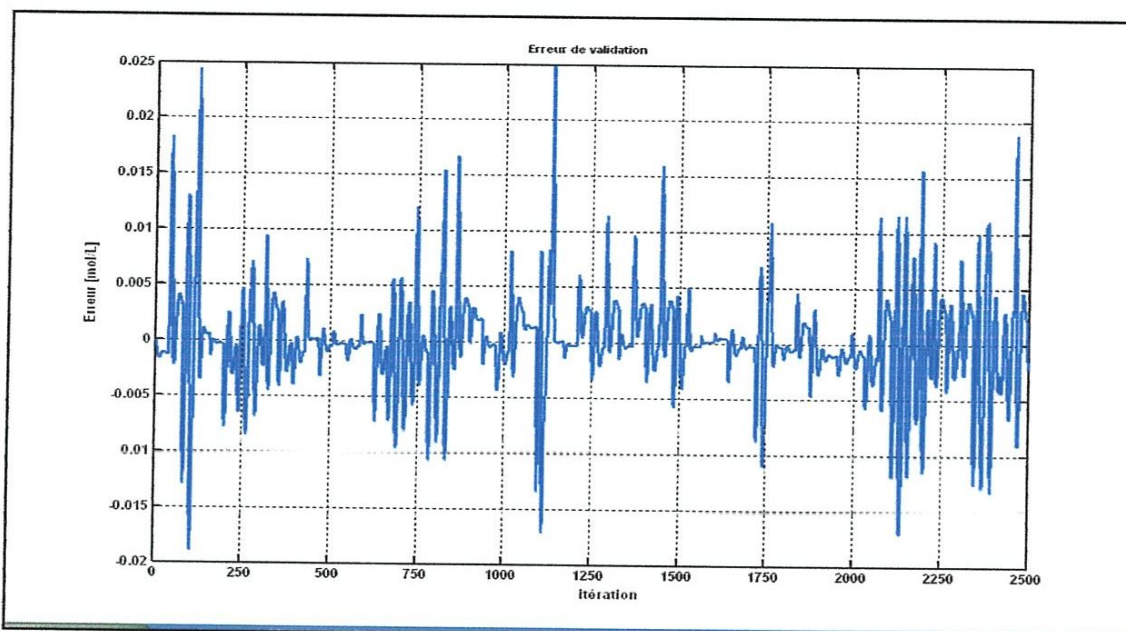


Figure 3.6: L'erreur de validation

Pour le modèle flou fourni par la fonction `genfis2`, on obtient 06 règles floues avec des séquences linéaires dont ses fonctions d'appartenance sont représentées par les figures 3.7, 3.8 et la figure 3.9.

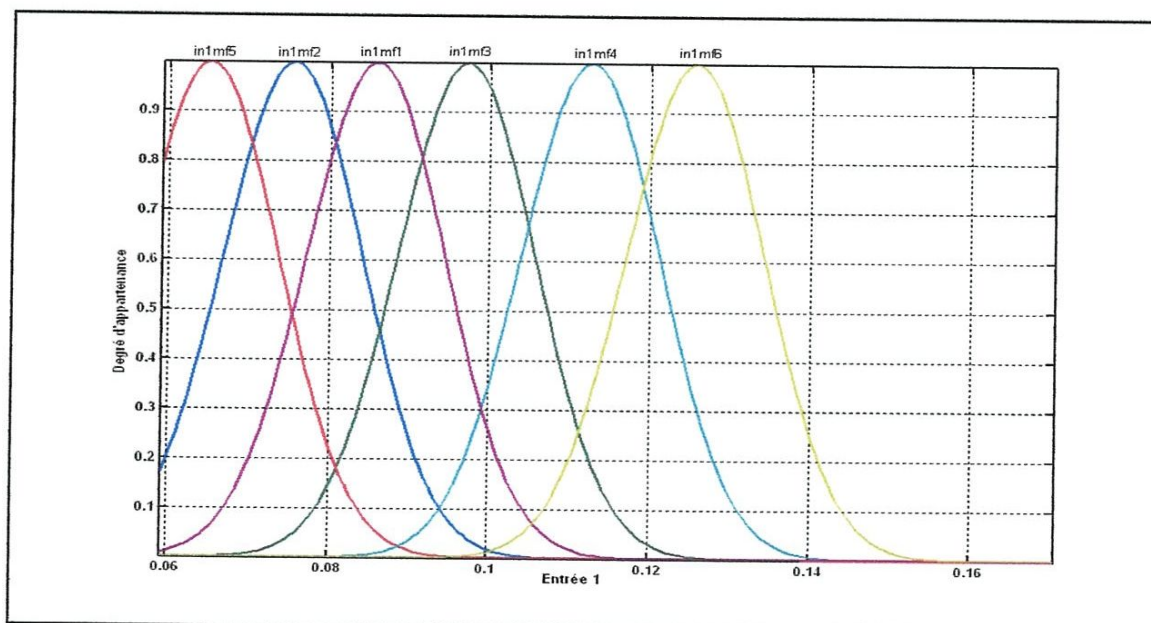


Figure 3.7: La fonction d'appartenance de l'entrée  $Ca(t)$

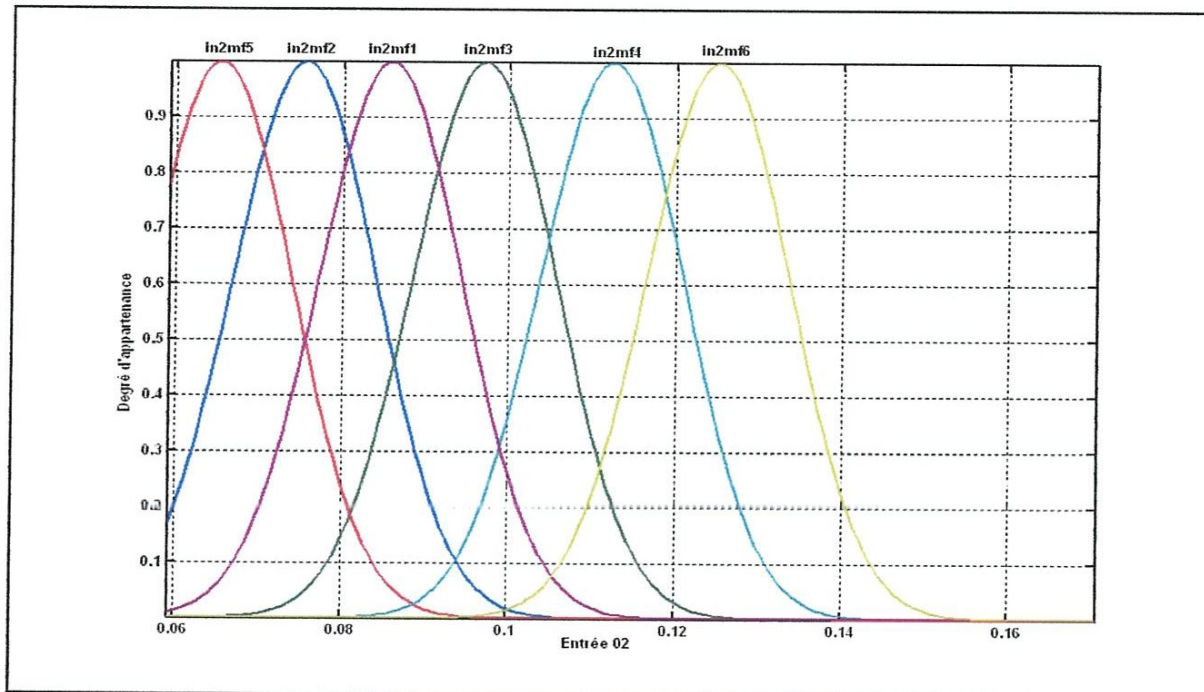


Figure 3.8: La fonction d'appartenance de l'entrée  $\bar{C}a(t-1)$

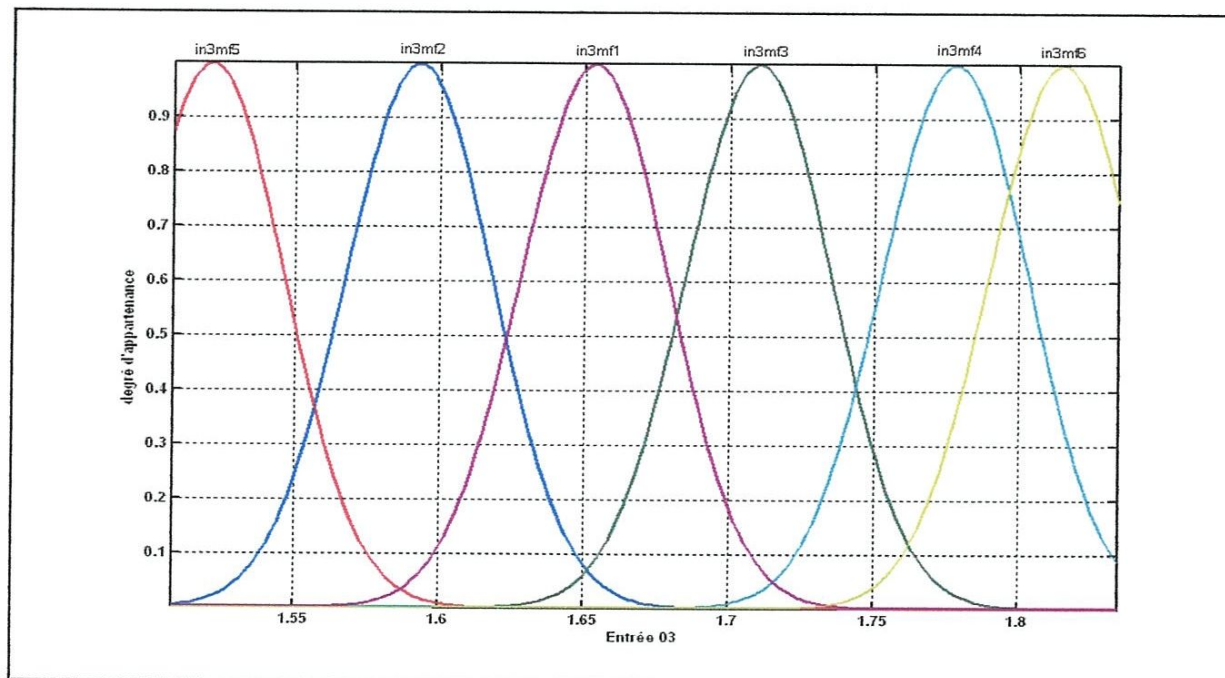


Figure 3.9: La fonction d'appartenance de l'entrée  $qc(t-1)$

La loi de GPC sous contraintes est développée à partir du modèle flou ainsi ses paramètres de réglage sont à priori fixés comme suit :

$$N_1 = 1 ; N_2 = 10 ; N_u = 5 ; \lambda = 1 \quad (3.5)$$

Les contraintes imposées sur le signal d'incrément de commande sont données par :

$$-0.1 \leq \Delta q_c(t+j) \leq 0.1, \text{ avec } j = 0,1 \quad (3.6)$$

Les contraintes imposées sur le signal de commande sont données par :

$$1.5101 \leq q_c(t+j) \leq 1.8368, \text{ avec } j = 0,1 \quad (3.7)$$

L'application de la loi de GPC sous contraintes sur la colonne de distillation cité précédemment avec les paramètres de réglage (3.5) tenant comptes les contraintes (3.6) et (3.7), on obtient les réponses données par les figures (3.10), (3.11) et (3.12) qui représentent respectivement les signaux de sortie réelle, l'effort de commande et les incrément de commande.

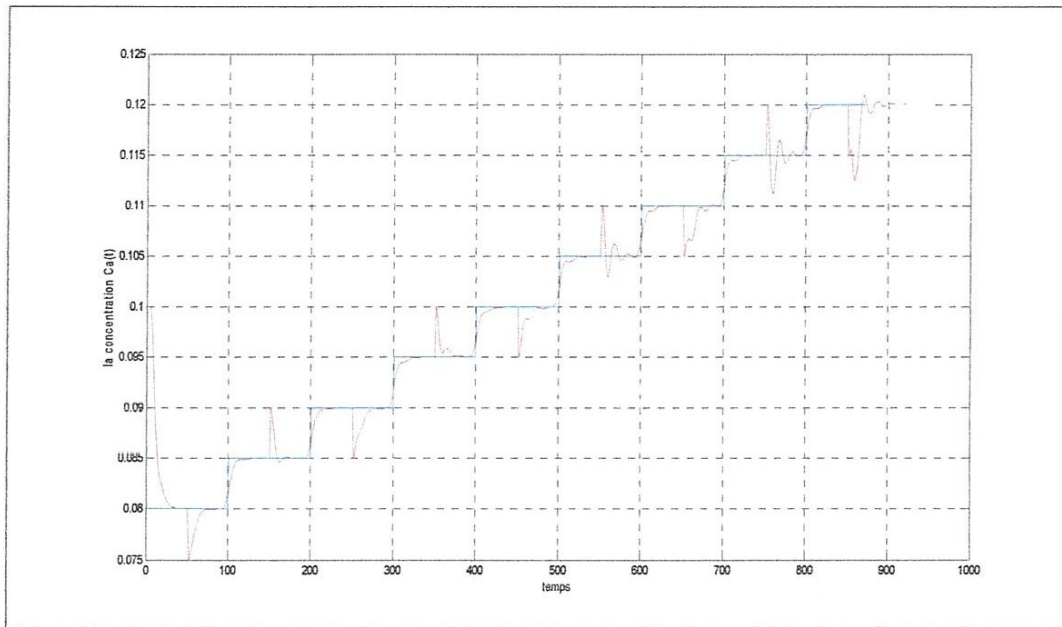


Figure 3.10 : Dynamique de poursuite assurée par la loi de GPC sous contraintes

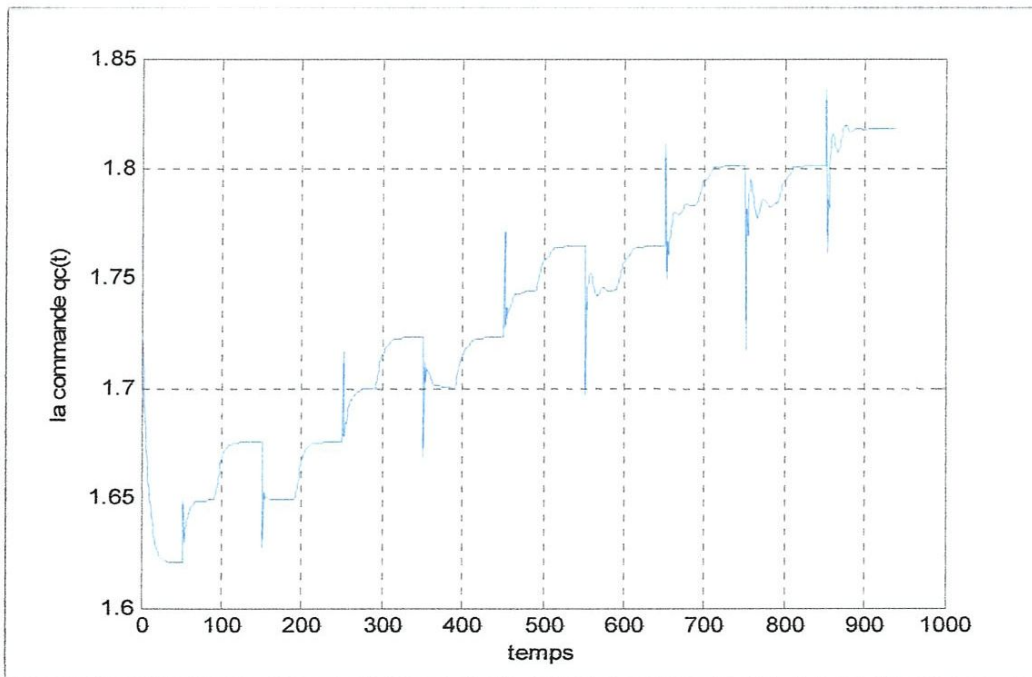


Figure 3.11 : Effort de commande fourni par la loi de GPC sous contraintes

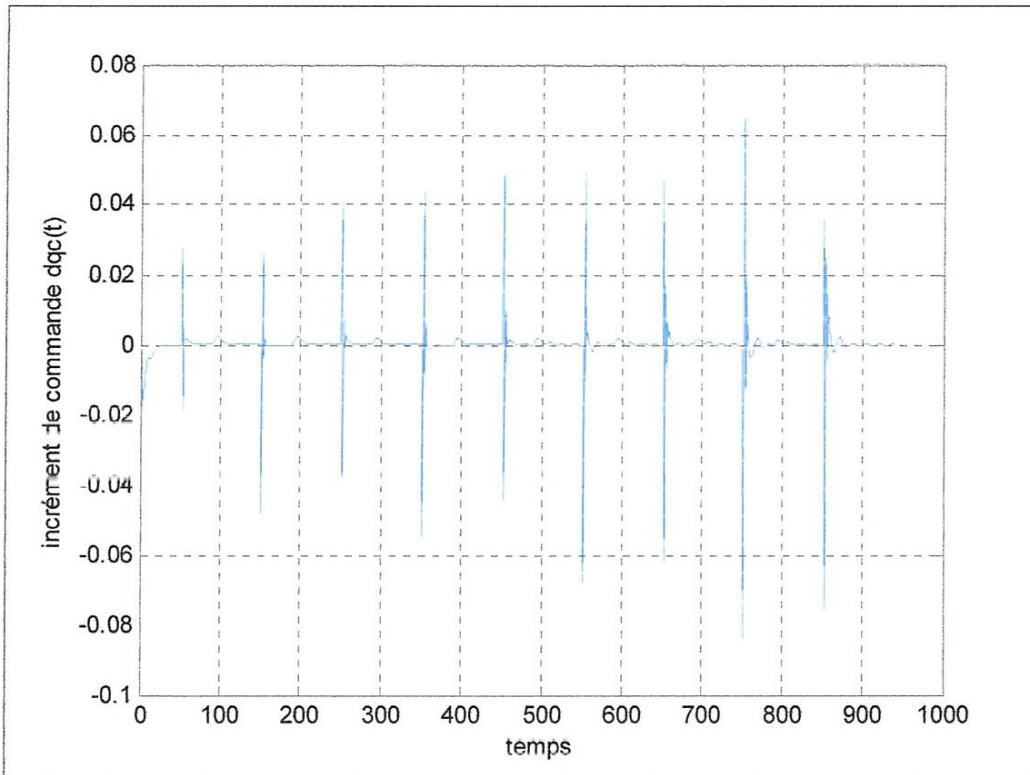


Figure 3.12 : Incrément de commande fournis par la loi de GPC sous contraintes

### III.4. Conclusion

D'après les figures précédentes, on a pu constater les remarques suivantes :

- Pour les paramètres de réglage ( $Nu$ ,  $N2$ ,  $\lambda$ ), la méthode d'optimisation *quadprog* assure une bonne dynamique de poursuite avec une meilleure rejection de perturbation. Ces comportements sont obtenus avec une bonne satisfaction de toutes les contraintes imposées par le cahier de charges.
- Comme résultat final, on peut confirmer que cette méthode de commande reste efficace pour les systèmes non linéaires modélisés par des modèles de T-S.



# CONCLUSION GENERALE

## CONCLUSION

On a présenté dans ce travail les démarches de la commande prédictive généralisée sous contraintes. Cette dernière assure des bonnes performances non seulement pour les systèmes linéaires invariants dans le temps mais aussi pour le cas des processus non linéaires modélisés par des modèles flous de type Takaggi Suggeno.

On a utilisé, dans la phase d'optimisation, la méthode *Quadrprog* pour la minimisation de la fonction coût de GPC. Cette dernière est donnée sous forme quadratique où sa minimisation assure une solution optimale.

On a présenté, à la fin de ce mémoire, une application de la commande prédictive généralisée à base d'un modèle linéaire et paramétrique qui est développé via le modèle flou de T S, et ce pour chaque pas de discrétisation issue d'une moyenne pondérée des sous modèles locaux qui ne sont que les point de fonctionnement associé du système.

# ANNEXE

### A.1. Calcul de prédicteur

La quantité  $\hat{y}(t + j/t)$  fera la référence au prédicteur de la sortie du processus calculée à l'instant courante  $t$ , depuis l'instant  $t + N_1$  jusqu'à l'instant  $t + N_2$ .

On pose que :

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

Cette dernière équation (1) est appel : équation de DIOPHANTINE, et puisque :

$C(q^{-1}) = 1$ , on a :

$$1 = E_j A(q^{-1}) \cdot \Delta + q^{-j} F_j(q^{-1}) \quad (2)$$

Pour obtenir la quantité  $y(t + 1)$ , on multiple l'équation (1.7) par la quantité

$E_j \cdot \Delta \cdot q^j$  On aura :

$$A \cdot E_j \cdot \Delta \cdot y(t + j) = B \cdot E_j \cdot \Delta \cdot u(t + j - 1) + e(t) \cdot E_j \cdot q^j$$

Sachant que  $e(t) \cdot q^j = e(t + j)$ , on obtient :

$$A \cdot E_j \cdot \Delta \cdot y(t + j) = B \cdot E_j \cdot \Delta \cdot u(t + j - 1) + E_j \cdot e(t + j) \quad (3)$$

D'après l'équation (2), on a :  $E_j \cdot A \cdots \Delta + 1 - q^{-j} \cdot F_j$

Alors :

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

D'où :

$$y(t + j) = F_j \cdot y(t) + B \cdot E_j \cdot \Delta \cdot u(t + j - 1) + e(t + j) \cdot E_j \quad (4)$$

Avec :

$$\Delta \cdot u(t + j - 1) = u(t + j - 1) - u(t + j - 2)$$

Le prédicteur

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

$$G_j = B \cdot E_j$$

$$y(t + j/t) = G_j \Delta u(t + j - 1) + F_j y(t) \quad (5)$$

Où :

$$G_j = g_{j0} + g_{j1} \cdot q^{-1} + g_{j2} q^{-2} + \cdots \cdots g_{jnb+j-1} \cdot q^{-(nb+j-1)}$$

$$E_j = e_{j0} + e_{j1} q^{-1} + \cdots \cdots + e_{jj} z^{-(j-1)} \quad (e_{j0} = 1)$$

$$F_j = f_{j0} + f_{j1}q^{-1} + \dots + f_{jna}q^{-na}.$$

$$y(t+1/t) = G_1 \Delta u(t) + F_1 y(t)$$

$$= g_{10} \Delta u(t) + (G_1 - g_{10}) \Delta u(t) + F_1 y(t) \quad (6)$$

$$= g_{10} \Delta u(t) + f(t+1) \quad (7)$$

$$y(t+2/t) = G_2 \Delta u(t+1) + F_2 y(t) \quad (8)$$

$$= g_{20} \Delta u(t+1) + g_{21} \Delta u(t) + (G_2 - g_{20} - g_{21}q^{-1}) \Delta u(t+1) + F_2 y(t) \quad (9)$$

$$= g_{20} \Delta u(t+1) + g_{21} \Delta u(t) + f(t+2) \quad (10)$$

$$\hat{y} = f + G \cdot u \quad (11)$$

$$\hat{y} = [\hat{y}(t+1), \hat{y}(t+2), \dots, \hat{y}(t+N)] \quad (12)$$

$$u^T = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N-1)] \quad (13)$$

$$f_{co} = [f(t+1), f(t+2), \dots, f(t+N)] \quad (14)$$

$$J = E[(\hat{y} - w)^T \cdot (\hat{y} - w) + \lambda \cdot u^T \cdot u] \quad (15)$$

$$J = (G \cdot u + f - w)^T (G \cdot u + f - w) + \lambda \cdot u^T \cdot u \quad (16)$$

$$u = [G^T \cdot G + \lambda \cdot I]^{-1} \cdot G^T \cdot (w - f)$$

# REFERENCES

## Références:

- [1] E.F.Camacho, “*Constrained Generalised Predictive Control*”, IEEE Transaction. on Automatic .Control, AC-38, pp.327-332, (2)1993.
- [2] E.F.Camacho, C. Bordons, “ *Model Predictive Control* ”, edition Springer-Verlag London,GB,1999
- [3] P.J. Campo, M. Morari, “*infinity-norm formulation of model predictive control problems*”, in proc. American Control conf., Seattle, WA, pp.339-342, 1986.
- [4] Clarck, D. W., C. Mohtadi, ” *Properties of generalized predictive control*”. Automatica, pp.859-875, (26)1989.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] D.W.Clark,F.R.Scattolini, “*Constrained Receding Horizon Predictive Control* ”, (CRHPC),*IEE Proceedings*, .vol. 138, July, (4)1991.
- [10] R.Y. Chiang and M. G. Safonov, “*Robust control tool Box: Users Guides*”, version 2, 1999.
- [11] 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.
- [12] 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.

- [13] 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.
- [14] B.A Francis, J.C Doyle "Linear control theory with year  $H_{\infty}$  optimally criterion", *control opt*, vol.25, pp.815-844, 1997.
- [15] R. Fletcher, "Practical method of optimisation", (John Wiley and Sons, 1990).
- [16] A.Gourdin, M.Boumahrat, «Méthode numériques appliquées », seconde édition, O.P.U, pp.87-94, 1991
- [17] 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
- [18] 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.
- [19] M. Kothare, V. Balakrishnan, et M. Morari « Robust Constrained Model Predictive Control using Linear Matrix Inequalities". *Automatica*, vol. 32, pp.1361-1379, 1996.
- [20] 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.
- [21] R.M.C.de Keyzer, A.R.van Cauwenberghe, "Extended prediction self-adaptive control". *IEEE Transactions on Automatic Control*, pp.831-847, (34)1989.
- [22] 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.
- [23] D.R Lewing, "Stable robust performance specifications for uncertain SISO systems". *Int. J. Control*, pp.1263-1281, (53)1989.
- [24] 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.
- [25] 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.



- [26] S.Mokhtar, H.Shetty, “*Non linear programming- theory and algorithms-*”, second edition, John Wiley& Sons.Inc, pp 274-282, pp420- 433, 1993.
- [27] M.M'saad, J.Chebassier. “*Adaptive Order of the systems*”, techniques of the engineer, labo.d' car. S7 426\_1 - S7 426-23, 1999
- [28] M.Minoux, «*Programmation mathématique – théorie et algorithmes-* » édition Dunod. 1988.
- [29] K.Ogata “*Discrete time control systems*” university of Minnesota, USA, flight II, pp.578 – 600, 1993.
- [30] G. Ramond, “*Contribution A La Commande Prédictive Généralisée Adaptative Directe Et Application* ”, Thèse De Doctorat ,Université Paris XI Orsay. pp08-33, 2001.
- [31] J.Richalet, “Application to Industrial Processes”, *Automatica* 14, pp.413-428, 1978
- [32] N.L. Ricker, “*The use of quadratic programming for constrained internal model control*”, ind. Eng. Commun,pp.73-87, (46)1986.
- [33] J. Richalet “*Predictive order* ”, techniques of the engineer, R7 423\_1 – R7 423\_17, 1999.
- [34] P .Rodriguez Ayerbe “*Robustification De Lois De Commande Prédictive Par la Paramétrisation De Youla* ”,Thèse Docteur ,Université Paris XI Orsay (Supélec), pp14-36, 2003.
- [35] 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.
- [36] N.Sugie, “*An extension of Fibonacci searching to multidimensional cases*”, IEEE. Trans. Autom. Control, pp.105-108, (9)1964.

