

**UNIVERSITE SAAD DAHLEB DE BLIDA**

**Faculté des Sciences de l'Ingénieur**

Département d'Electronique

**MEMOIRE DE MAGISTER**

Spécialité : Signaux & Systèmes

**COMMANDE PREDICTIVE**

**A**

**MODELE NEURONAL**

Par

**Linda KAIBICHE**

Devant le jury composé de :

Mr A.GUESSOUM	Professeur, U.de Blida	Président
Mr M.HADJSADOK	Maître de conférences, U.de Blida	Examineur
Mr K.BENMANSOUR	Maître de conférences, U de.Media	Examineur
Mr K. KARA	Maître de conférences, U. de Blida	Rapporteur

Blida, septembre 2011

## **RESUME**

Ce travail envisage la mise en œuvre d'une loi de commande prédictive non linéaire à modèle neuronal. La stratégie adoptée, est celle qui consiste à utiliser un modèle de prédiction constitué d'une partie non linéaire, dépendante des conditions initiales disponibles à l'instant actuel, et d'une partie linéaire dépendante des incréments futures du signal de commande. Le système à commander est modélisé par un réseau de neurones multicouches statique. Ce modèle sert à générer les deux parties du modèle de prédiction. Cette stratégie de commande permet de fournir une solution, rapide et efficace, au problème d'optimisation de la fonction de coût sans recourir aux algorithmes d'optimisation numérique. En effet, elle permet d'établir une expression analytique pour la loi de commande.

Afin de mettre en évidence l'apport de la démarche développée, nous considérons sous forme de simulation, la commande du réacteur chimique CSTR. Ainsi qu'une étude comparative avec d'autres approches existantes sera présentée.

## **ABSTRACT**

This work deals with the development of a nonlinear predictive control based on a neural model: the Adopted strategy consists on using a prediction model which is a combination of a non linear and a linear part. The first part depends on the initial conditions, whereas the second part depends on the future increments of the control signal. The system to be controlled is modeled by a multi-layer static neural network. This model is used to generate the two parts of the prediction model. This control strategy allows to provide a rapid and effective solution to the problem of optimization, this without using other numerical optimization algorithms to minimize the cost function. In fact, it allows to set up an analytical expression for the control law. In order to highlight the contribution of the proposed methods a simulation to control a continuous-stirred tank reactor and a comparative study with other existing approaches were considered.

## ملخص

هذا العمل يعالج مسألة تصميم قانون تحكم تنبؤي غير خطي، ذو نموذج عصبوني. الخطة المتبناة تتمثل في استعمال نموذج تنبؤي متكون من جزء غير خطي، مرتبط بالشروط الأولية متوفرة في الوقت الحالي، و من جزء خطي مرتبط بالمتزايدات مستقبلية لإشارة التحكم. نتحصل على نموذج النظام المتحكم فيه بواسطة شبكة أعصاب ساكنة المتعددة الطبقات. هذا النموذج يستعمل في حساب جزئي النموذج التنبؤي. تمكن خطة التحكم هذه من إعطاء حل سريع وفعال لمشكلة تقليص دالة الكلفة بدون استعمال خوارزميات التقليص العددي. بالفعل، تمكن هذه الخطة من إيجاد عبارة تحليلية لقانون التحكم.

من أجل عرض نجاعة الخطة مقترحة، سنطبقها في حساب نظام التحكم لمفاعل كيميائي ذو تفاعل غير عكسي (CSTR)، و سنقوم بمقارنة النتائج المتحصل عليها مع نتائج خطط أخرى.

## REMERCIEMENTS

Tout d'abord nous remercions dieu le tout puissant de nous avoir permit d'arriver à bout de ce mémoire.

Je voudrais, exprimer ma plus grande reconnaissance à Monsieur kara kamel mon promoteur qui a suivi ce travail, en détail, avec une compétence et patience remarquable.

J'exprime mes remerciements aux membres de jury, d'avoir bien voulu consacrer du temps et de l'attention à mon travail.

Je remercie également mes proches pour leurs encouragements et leurs appuis.

Je tiens à saluer tous mes collègues ainsi que l'ensemble des personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Enfin, je ne saurais suffisamment remercier mon très cher mari, pour son soutien de tous les jours, sa patience, son aide précieuse. Qu'il trouve ici toute ma gratitude et mon amour.

# TABLE DES MATIERES

RESUME	1
REMERCIEMENTS	3
TABLE DES MATIERES	4
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX	8
INTRODUCTION	9
1. COMMANDE PREDICTIVE GENERALISEE (GPC)	12
1.1. Introduction	12
1.2. Principe	12
1.3. Mise en Œuvre de la commande GPC	14
1.3.1. Définition du modèle du système	14
1.3.2. Prédicteur optimal	15
1.3.3. Fonction de coût quadratique	17
1.3.4. Prédicteur optimal sous forme matricielle	18
1.3.5. Critère quadratique matriciel	18
1.3.6. Minimisation de la fonction de coût	19
1.4. Choix des paramètres de réglage	19
1.4.1. Choix de l'horizon minimal de prédiction $N1$	20
1.4.2. Choix de l'horizon maximal de prédiction $N2$	20
1.4.3. Choix de l'horizon de prédiction sur la commande $Nu$	20
1.4.4. Choix du facteur de pondération de la commande $\lambda$	21
1.5. La commande prédictive avec contraintes	21
1.5.1. Contraintes sur l'incrément de commande	22
1.5.2. Contraintes sur l'amplitude de commande	23
1.5.3. Contraintes sur la sortie	24
1.5.4. Contraintes sur le dépassement	24
1.5.5. Contraintes terminales de type égalité	25
1.6. Conclusion	26
2. IDENTIFICATION ET COMMANDE A BASE DE RESEAUX DE NEURONES	27
2.1. Introduction	27

2.2. Réseaux de neurones	27
2.2.1. Modèle d'un neurone	28
2.2.2. Fonctions d'activation	29
2.2.3. Réseau de neurones multicouches	29
2.2.4. Apprentissage	31
2.3. Identification des systèmes par réseaux de neurones	32
2.3.1. Définition	32
2.3.2. Acquisition des exemples d'apprentissage et de test	34
2.3.2.1. Choisir la fréquence d'échantillonnage	34
2.3.2.2. Concevoir le signal d'excitation	34
2.3.2.3. Prétraitement des mesures obtenues	35
2.3.3. Choix des régresseurs	35
2.3.4. Choix de la structure du modèle	36
2.3.5. Calcul des paramètres	37
2.3.6. Validation du modèle	37
2.3.7. Exemple	37
2.4. La commande neuronale	39
2.4.1. Commande directe	40
2.4.1.1. Commande supervisée	40
2.4.1.2. Commande directe par modèle inverse	41
2.4.1.3. Commande par modèle interne	41
2.4.1.4. Commande adaptative à modèle de référence	42
2.4.2. Commande indirecte	43
2.5. Conclusion	43
3. COMMANDE PREDICTIVE A MODELE NEURONALE	44
3.1. Introduction	44
3.2. Commande prédictive à modèle neuronal	44
3.2.1. Principe général	44
3.2.2. Approche par décomposition et linéarisation instantanée	46
3.2.2.1. Fonction de coût	47
3.2.2.2. Modèle de prédiction	47
3.2.2.3. Minimisation de la fonction de coût	50
3.2.2.4. Algorithme	51

3.3. Simulation	51
3.3.1. Décomposition et utilisation d'un modèle linéaire fixe	51
3.3.2. Linéarisation instantanée	53
3.3.3. Décomposition et linéarisation instantanée	54
3.4. Conclusion	56
4. APPLICATION A LA COMMANDE D'UN REACTEUR	57
4.1. Introduction	57
4.2. Modélisation du réacteur continu parfaitement mélangé	57
4.2.1. Définition et caractéristiques des réacteurs chimiques	57
4.2.2. Classification des réacteurs chimiques	58
4.2.3. Caractéristique du réacteur C.S.T.R	59
4.2.4. Bilan de matière et d'énergie dans les réacteurs chimiques	60
4.2.5. Modélisation	61
4.2.5.1. Modèle Mathématique	61
4.2.5.2. L'identification du CSTR	66
4.3. Mise en œuvre du contrôleur	69
4.3.1. Calcul de la loi de commande du CSRT avec la GPC	70
4.3.2. Calcul de la loi de commande du CSRT par décomposition	72
4.3.3. Etude comparative	78
4.4. Conclusion	79
CONCLUSIONS ET PERSPECTIVES	80
APPENDICE	82
A. Liste des abréviations	82
REFERENCES	84

## LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

Figure 1.1	Représentation temporelle du principe de la commande prédictive	13
Figure 1.2	Schéma de principe du GPC	14
Figure 1.3	Représentation <i>CARMA</i> du modèle numérique	14
Figure 1.4	Choix de $N_1$ pour un système à déphasage non minimal	20
Figure 1.5	Choix de $N_2$ pour un système à déphasage non minimal	20
Figure 2.1	Modèle d'un neurone artificiel.	28
Figure 2.2	Fonctions d'activation	29
Figure 2.3	Structure du réseau de neurones MLP	30
Figure 2.4	Les principales étapes de l'identification	33
Figure 2.5	Exemple de signaux d'excitation	34
Figure 2.6	Le modèle série-parallèle (NARX)	37
Figure 2.7	La réponse du système à l'entrée d'excitation	38
Figure 2.8	Validation du modèle obtenu	39
Figure 2.9	Structure de la commande supervisée	40
Figure 2.10	Commande directe par modèle inverse	41
Figure 2.11	Commande à modèle interne	41
Figure 2.12	Commande adaptative à modèle de référence	42
Figure 3.1	Schéma de principe de la commande prédictive à modèle neuronal	45
Figure 3.2	Prédicteur neuronal	45
Figure 3.3	Structure de commande	47
Figure 3.4	Performances obtenues du régulateur	52
Figure 3.5	Effort de commande	52
Figure 3.6	Performances obtenues du régulateur	53
Figure 3.7	Effort de commande	54
Figure 3.8	Performance obtenues du régulateur	55
Figure 3.9	Signal de commande	55
Figure 3.10	Performances obtenues dans les trois cas de commande	56
Figure 4.1	Schéma du réacteur chimique parfaitement mélangé	59

Figure 4.2	Variation des paramètres de réaction dans un CSTR	60
Figure 4.3	Stratégie de simulation d'un processus	61
Figure 4.4	Coupe du réacteur avec les différents échanges de chaleur	64
Figure 4.5	Evolution des paramètres $C_a$ , T et $T_c$ du réacteur quand $Q_c = 0$	66
Figure 4.6	Données d'apprentissage et de validation	67
Figure 4.7	Schéma de l'identification	68
Figure 4.8	Evolution de l'erreur quadratique moyenne pendant l'apprentissage	68
Figure 4.9	Validation du modèle	69
Figure 4.10	Trajectoire de référence	69
Figure 4.11a	Performances du contrôleur	70
Figure 4.11b	Vue agrandie de la sortie & référence de la figure 4.11a	71
Figure 4.12	Erreur de poursuite	71
Figure 4.13a	Performances du contrôleur	73
Figure 4.13b	Vue agrandie de la sortie & référence de la figure 4.13a	73
Figure 4.14	Signal de commande	74
Figure 4.15	Signal de perturbation	74
Figure 4.16	Performance du contrôleur avec perturbations, cas CPNMLPF	75
Figure 4.17	Performance du contrôleur avec perturbations, cas CPNMLPV	76
Figure 4.18	Erreur de poursuite avec perturbation cas CPNMLPF et CPNMLPV	76
Figure 4.19	Performances obtenues pour différentes valeurs de $N_2$ ( $S=10^{-5}$ )	77
Figure 4.20	Performances obtenues pour différentes valeurs de S ( $N_2=8$ )	77
Figure 4.21	Performances obtenues pour les quatre algorithmes de commande	78
Tableau 4.1	Classification des réacteurs chimiques	58
Tableau 4.2	Constantes thermodynamiques chimiques du CSTR	65

## INTRODUCTION

La commande prédictive est née à la fin des années 1970 et s'est considérablement développée depuis, à la fois dans la communauté de la recherche en automatique et dans l'industrie. La raison de ce succès peut être attribuée au fait que la commande prédictive est la façon la plus générale de poser le problème de la commande de procédés dans le domaine temporel. En effet, sa formulation intègre des concepts tirés de la commande optimale, la commande stochastique, la commande des systèmes à temps mort, la commande avec modèle interne, la commande multivariable et elle prend en compte les références futures lorsqu'elles sont disponibles [4].

Grâce à ses concepts intuitifs, la commande prédictive a été implantée dans un grand nombre d'applications industrielles, parmi lesquelles il faut noter les processus chimiques, qui ont été les premiers à utiliser ce type de commande, les processus de distillation, l'industrie pétrolière et les systèmes électromécaniques tels que la commande d'axes de robot. Ces applications industrielles ont toutes un dénominateur commun : la connaissance de la trajectoire à suivre par le système dans le futur, au moins sur un certain horizon.

Les réseaux de neurones artificiels ou réseaux de neurones formels sont développés depuis les années 40 par la volonté de l'homme de comprendre et d'imiter les capacités du cerveau (mémoire, apprentissage, intelligence, traitement parallèle massif d'informations et plasticité). Arrivés sur le marché industriel au milieu des années 80, les réseaux de neurones artificiels ont largement bénéficié des capacités de calcul offertes par les ordinateurs modernes [8][9].

Le regard que nous portons sur les caractéristiques des réseaux de neurones est celui des ingénieurs: le grand nombre de connexions entre neurones, la non linéarité des relations d'entrée-sortie des neurones, à leur faculté d'adaptabilité et leur parallélisme.

Les domaines d'application privilégiés, présentant les caractéristiques d'une résolution neuronale, concernent le regroupement et la classification des données, le

traitement du signal, la modélisation et l'identification de processus, la surveillance et la commande de processus, la prédiction et l'aide à la décision[8].

La commande par réseaux de neurones a fait l'objet de nombreux travaux de recherche depuis une vingtaine d'années, à cause de la capacité d'apprentissage, d'approximation et de généralisation des réseaux de neurones. En effet plusieurs applications ont vu le jour et ont trouvé leur place dans le marché.

### ***Motivations***

Les principales difficultés dans la théorie de la commande des systèmes dynamiques réels sont les non-linéarités et les incertitudes. Or la commande passe par l'élaboration d'un modèle mathématique du système pour trouver une relation entre les entrées et les sorties, ce qui suppose une bonne connaissance de la dynamique du système et ses propriétés. Dans le cas des systèmes non-linéaires, les techniques conventionnelles ont montré souvent leur insuffisance surtout quand les systèmes à étudier présentent de fortes non linéarités. Le manque de connaissances à priori nécessaires pour l'élaboration du modèle mathématique était en quelque sorte dans cet échec. Face à ce problème, le recours aux méthodes de commandes par apprentissage est devenu une nécessité car les systèmes de commande obtenus ainsi procèdent par collecter des données empiriques, stocker et extraire les connaissances contenues dans celle-ci et utiliser ces connaissances pour réagir à de nouvelles situations : on a passé alors à la commande intelligente [9] [10].

Ces dernières années la commande prédictive à modèle neuronal, que l'on peut rattacher à la famille de la commande par modèle, ne cesse de connaître un développement croissant. Par le faite que les réseaux de neurones sont des approximateurs universels et par leur capacité de s'adapter à une dynamique évoluant au cours du temps, plusieurs techniques utilisant l'approche neuronale et permettant de mettre en œuvre ce type de commande ont été proposées. Néanmoins, vu la structure complexe des réseaux de neurones, l'utilisation des modèles neuronaux conduit à un problème d'optimisation non linéaire et non convexe pour lequel une solution analytique n'est pas possible. Les méthodes itératives de l'optimisation non linéaire sont généralement très coûteuses en terme de temps de calcul, ceci est très contraignant dans le cas d'une application en temps réel portant sur des systèmes à dynamique rapide. L'objectif tracé pour notre travail, est de trouver une solution à cette difficulté. En fait, nous avons exploité l'idée proposée par E.

F.camacho et al [20] qui consiste à décomposer la réponse totale du système en deux parties : réponse libre et réponse forcée. Dans notre travail, la matrice des coefficients de la réponse indicielle est générée, à chaque période d'échantillonnage en utilisant un modèle linéaire. Ce dernier, est obtenu par linéarisation, à chaque période d'échantillonnage, du modèle neuronal du système. Nous rappelons que E. F.camacho et al. Ont utilisé dans leurs travaux un modèle linéaire fixe, obtenu par identification, pour générer cette matrice.

### ***Organisation du mémoire***

Ce mémoire est articulé autour de quatre chapitres: Les deux premiers introduisent les principes fondamentaux : de la commande prédictive généralisée (GPC : Generalized predictive control), des réseaux de neurones, de la modélisation et de la commande neuronale d'un point de vue général et conceptuel. Le troisième chapitre est consacré à la commande prédictive à modèle neuronal, il présente les différentes approches considérées dans ce travail. Enfin, le dernier chapitre contient l'application des algorithmes considérés sur un réacteur chimique parfaitement mélangé.

# CHAPITRE 1

## COMMANDE PREDICTIVE GENERALISEE (GPC)

### 1.1. Introduction

La Commande Prédicative est née à la fin des années 1970, mais n'a connu un réel essor en tant que technique de commande avancée que depuis le milieu des années 80. Cet essor s'est réalisé principalement selon deux axes privilégiés :

- Commande Prédicative Généralisée (G.P.C.) de D.W. Clarke: 1985 [1]
- Commande Prédicative Fonctionnelle (P.F.C.) de J. Richalet: 1987 [2]

Dans ce chapitre on présente le principe de la commande prédictive en général, puis on aborde de façon détaillée la commande prédictive généralisée.

### 1.2. Principe

Comme l'illustrent par exemple les articles de Clarke, et al[1], la commande prédictive s'avère une structure suffisamment complète proposée pour résoudre un problème très général, fournissant un système asservi stable pour un jeu de paramètres de réglage donné. Cette stratégie permet de commander des processus à déphasage non minimal, avec des pôles instables ou mal amorties, avec des temps morts, constants ou inconnus, ou avec un modèle d'ordre inconnu. Elle a montré son efficacité, sa flexibilité et son succès dans des applications industrielles, même pour des systèmes à faible période d'échantillonnage.

Le concept clé de la commande prédictive réside dans la création d'un effet anticipatif. On exploite pour cela les connaissances explicites sur l'évolution de la trajectoire à suivre dans le futur (connaissances nécessaires requises au moins sur un horizon de quelques points au delà de l'instant présent). Cette contrainte permettant de tirer partie de toutes les ressources de la méthode restreint nécessairement le domaine d'application à la commande de systèmes pour lesquels la trajectoire à suivre est

parfaitement connue et stockée point par point dans le calculateur. C'est le cas de la commande numérique de machine outils (découpe de pièces), de la commande de bras de robots, de suivis de profils de température d'un bâtiment ...

La mise en œuvre de ce concept nécessite la considération des points suivants :

- Définition d'un modèle numérique du système permettant de réaliser la prédiction du comportement futur du système. Ce modèle peut être obtenu par une discrétisation de la fonction de transfert continue du modèle (transformée en  $z$ ) ou par une identification préalable hors ligne du système. Cette particularité permet de classer la commande prédictive dans la grande famille des commandes à base de modèles, dite 'MBC' (Model Based Control).

- Élaboration d'une séquence de commandes futures. Elle est obtenue par minimisation d'une fonction de coût quadratique, sur un horizon fini, portant sur les erreurs de prédiction futures, (écarts entre la sortie prédite du système et la consigne ou une trajectoire de référence future), et sur un terme dépendant de la commande.

La figure 1.1 illustre le principe de la commande prédictive, où l'on observe les commandes  $u(k)$  à appliquer au système pour obtenir le ralliement autour de la consigne  $w(k)$ .

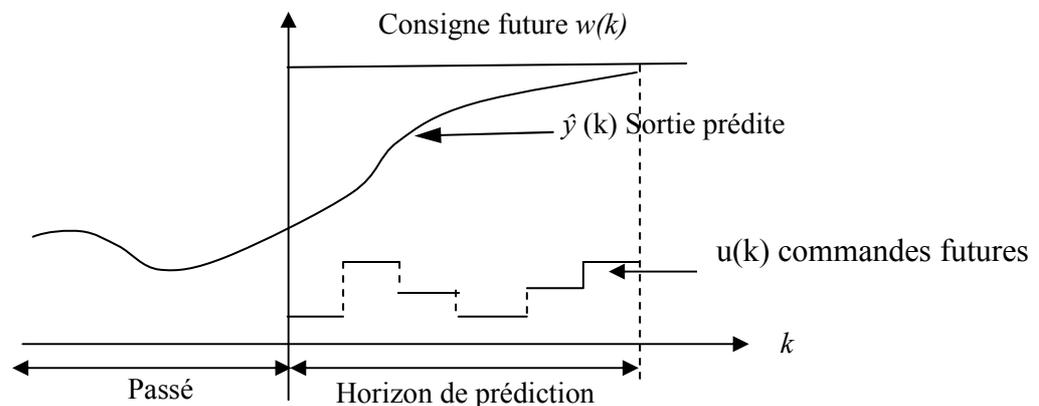


Figure 1.1 : Représentation temporelle du principe de la commande prédictive

- Seul le premier élément  $u(k)$  de la séquence optimale de commande est appliqué sur le système, tous les autres éléments de la séquence sont oubliés. A la période d'échantillonnage suivante, une nouvelle sortie est mesurée et la procédure complète est répétée. Ce procédé s'appuie sur le principe de l'horizon fuyant (receding horizon).

Les principes de la commande prédictive qui viennent d'être exposés permettent d'établir le schéma fonctionnel de la figure 1.2.

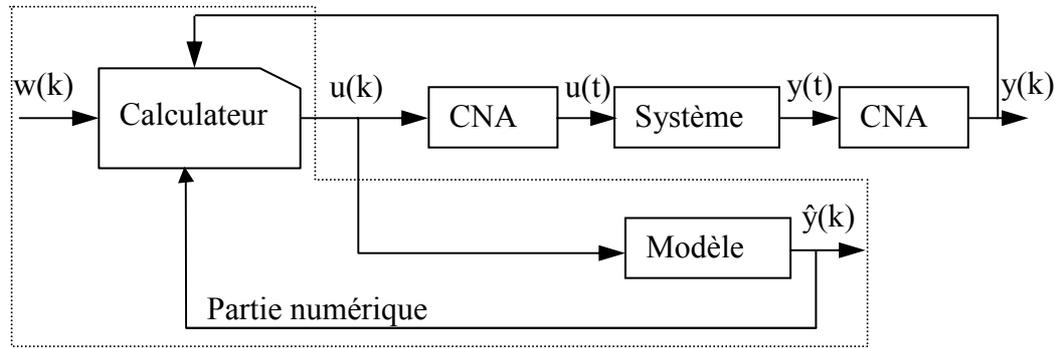


Figure 1.2: Schéma de principe du GPC

avec :  $w(k)$  consigne,  $y(k)$  sortie du processus,  $\hat{y}(k)$  sortie prédite,  $u(k)$  commande appliquée sur le système.

### 1.3. Mise en Œuvre de la commande GPC

#### 1.3.1. Définition du modèle du système

La commande prédictive nécessite, comme nous l'avons vu ci-dessus, un modèle numérique de représentation pour son fonctionnement. Ce modèle permet de calculer la sortie prédite sur un horizon fini.

La loi de commande prédictive n'est pas astreinte à un type de représentation particulier, mais historiquement adopte de préférence une forme de représentation du système de type polynomial externe (entrée-sortie). Le système est alors représenté sous la forme *CARMA* (Controlled AutoRegressive Moving Average), correspondant à la figure 1.3, dont l'expression est :

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

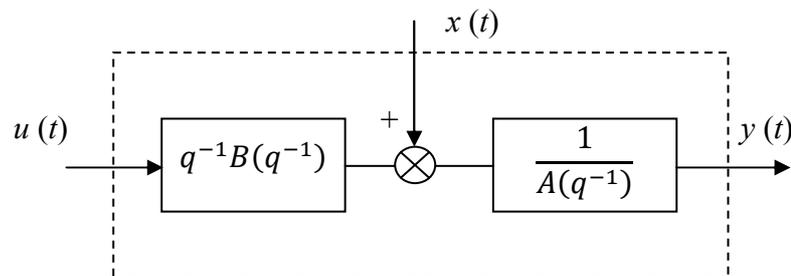


Figure 1.3 : Représentation *CARMA* du modèle numérique

avec :

$y(t)$  : sortie du processus

$u(t)$  : commande appliquée au système

$q^{-1}$  : opérateur retard

$x(t)$  : terme lié aux perturbations, choisi généralement sous la forme  $x(t) = C(q^{-1})\xi(t)$  avec  $\xi(t)$  est une séquence aléatoire non corrélée centrée.

Les polynômes  $A(q^{-1})$ ,  $B(q^{-1})$  et  $C(q^{-1})$  sont définis par :

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{na}q^{-na} \\ B(q^{-1}) &= b_0 + b_1q^{-1} + \dots + a_{nb}q^{-nb} \\ C(q^{-1}) &= 1 + c_1q^{-1} + \dots + c_{nc}q^{-nc} \end{aligned}$$

On obtient alors le modèle *CARMA* suivant :

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

Une autre possibilité consiste à raisonner sur un modèle incrémental du système, en introduisant une action intégrale dans le modèle :

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\frac{\xi(t)}{\Delta(q^{-1})} \quad (1.3)$$

avec :  $\Delta(q^{-1}) = 1 - (q^{-1})$  opérateur différence.

L'équation (1.3) constitue le modèle *CARIMA*. Le but recherché avec l'introduction d'un intégrateur est l'annulation de toute erreur statique vis-à-vis d'une consigne ou d'une perturbation constante. On retiendra alors pour la suite cette dernière structure *CARIMA*, avec  $C(q^{-1}) = 1$  en ne s'intéressant qu'aux fonctions de transfert entrée/sortie pour lesquelles ce polynôme n'a aucune influence.

### 1.3.2. Prédicteur optimal

La méthodologie prédictive requiert la définition d'un prédicteur optimal à  $j$ -pas qui permette d'anticiper le comportement du processus dans le futur sur un horizon fini. Pour cela, à partir de la forme du modèle (équation (1.3)), on élabore la sortie estimée à l'instant  $t + j$ , connaissant la sortie à l'instant  $t$ . La sortie prédite  $\hat{y}(t + j/t)$  est décomposée de

façon classique en réponse libre et réponse forcée, incluant une forme polynomiale pour mener à bien la synthèse polynomiale finale, sous la forme :

$$\hat{y}(t+j) = \underbrace{F_j(q^{-1})y(t) + H_j(q^{-1}) \Delta u(t-1)}_{\text{réponse libre}} + \underbrace{G_j(q^{-1}) \Delta u(t+j-1) + J_j(q^{-1})\xi(t+j)}_{\text{réponse forcée}} \quad (1.4)$$

Avec  $G_j$  représentant le futur,  $F_j, H_j$  correspondant au présent et au passé,  $J_j$  est lié aux perturbations. Dès lors, le premier terme représentant la réponse libre est dû aux incréments de commandes passées, le deuxième terme représentant la réponse forcée correspond aux incréments de commandes futures et présentes, et à l'influence des perturbations.

L'équation (1.3) du modèle *CARIMA*, combinée avec celle du prédicteur (équation (1.4)), fournit le système d'équations ci-dessous :

$$\begin{cases} A(q^{-1})\Delta(q^{-1})y(t+j) = B(q^{-1})\Delta u(t+j-1) + \xi(t+j) \\ [1 - q^{-j}F_j(q^{-1})]y(t+j) = [G_j(q^{-1}) + q^{-j}H_j(q^{-1})]\Delta u(t+j-1) + J_j(q^{-1})\xi(t+j) \end{cases} \quad (1.5)$$

Considérant l'égalité des fonctions de transfert, les quatre polynômes inconnus sont obtenus de façon explicite et unique par résolution des deux équations diophantiennes suivantes [4] :

$$\begin{cases} \Delta(q^{-1})(A(q^{-1})J_j(q^{-1})q^{-j}F_j(q^{-1})) = 1 \\ G_j(q^{-1}) + q^{-j}H_j(q^{-1}) = B(q^{-1})J_j(q^{-1}) \end{cases} \quad (1.6)$$

En faisant l'hypothèse que la meilleure prédiction du terme lié aux perturbations est sa moyenne (ici nulle dans le cas du bruit blanc centré), le prédicteur optimal est défini de façon unique, dès que les polynômes sont connus, par la relation :

$$\hat{y}(t+j) = F_j(q^{-1})y(t) + G_j(q^{-1})\Delta u(t+j-1) + H_j(q^{-1})\Delta u(t-1) \quad (1.7)$$

avec :

$$\begin{aligned} \text{degré}[J_j(q^{-1})] &= \text{degré}[G_j(q^{-1})] = j - 1 \\ \text{degré}[F_j(q^{-1})] &= \text{degré}[A(q^{-1})] \\ \text{degré}[H_j(q^{-1})] &= \text{degré}[B(q^{-1})] - 1 \end{aligned}$$

### 1.3.3. Fonction de coût quadratique

Une fois définie l'équation de prédiction, la méthode implique la minimisation d'un critère quadratique à horizon fini. La stratégie GPC minimise une somme pondérée des carrés des erreurs futures entre la sortie prédite et la consigne, et des incréments des commandes futures

$$J = \sum_{j=N_1}^{N_2} (\hat{y}(t+j) - w(t+j))^2 + \lambda \sum_{j=1}^{N_u} \Delta u(t+j-1)^2 \quad (1.8)$$

Sous l'hypothèse :

$$\Delta u(t+j) = 0 \quad \forall j \geq N_u \quad (1.9)$$

avec :

$w(t+j)$ : consigne appliquée à l'instant  $t+j$ ,

$\hat{y}(t+j)$ : sortie prédite à l'instant  $t+j$

$\Delta u(t+j-1)$  : incrément de commande à l'instant  $t+j-1$ .

$N_1$ : horizon de prédiction minimal sur la sortie;

$N_2$ : horizon de prédiction maximal sur la sortie;

$N_u$ : horizon de prédiction sur la commande;

$\lambda$ : coefficient de pondération sur la commande.

La relation (1.9) signifie que lorsque le pas de prédiction  $j$  atteint la valeur fixée pour l'horizon de commande  $N_u$ , la variation de commande s'annule et donc que la commande future va se stabiliser. Cette hypothèse permettra par la suite de simplifier le calcul de la commande.

Le critère comprend donc un terme quadratique sur l'erreur et l'incrément de la commande. Sa minimisation analytique fournit la séquence des commandes futures dont seule la première sera effectivement appliquée. L'aspect incrémental du modèle se retrouve dans le critère par l'intermédiaire de  $\Delta u$ . Enfin le coefficient  $\lambda$  permet de donner plus ou moins de poids à la commande par rapport à la sortie, de façon à assurer la convergence lorsque le système de départ présente un risque d'instabilité.

Ce critère nécessite la définition de quatre paramètres de réglage :  $N_1$ ,  $N_2$ ,  $N_u$  et  $\lambda$ .

#### 1.3.4. Prédicteur optimal sous forme matricielle

La relation (1.7) donnant le prédicteur optimal est utilisée dans le critère (1.8) entre les horizons  $N_1$  et  $N_2$ . Pour simplifier les notations, il est possible d'utiliser une représentation matricielle de ce prédicteur.

Posons pour cela :

$$\begin{aligned}
 \mathbf{if}(q^{-1}) &= [F_{N_1}(q^{-1}) \dots F_{N_2}(q^{-1})]^T \\
 \mathbf{ih}(q^{-1}) &= [H_{N_1}(q^{-1}) \dots H_{N_2}(q^{-1})]^T \\
 \tilde{\mathbf{u}} &= [\Delta u(t) \dots \Delta u(t + N_u - 1)]^T \\
 \tilde{\mathbf{y}} &= [\Delta y(t + N_1) \dots \Delta y(t + N_2)]^T \\
 \mathbf{w} &= [\Delta w(t + N_1) \dots \Delta w(t + N_2)]^T
 \end{aligned}$$

$$\mathbf{G} = \begin{bmatrix} g_{N_1}^{N_1} & g_{N_1-1}^{N_1} & \dots & \dots \\ g_{N_1+1}^{N_1+1} & g_{N_1}^{N_1+1} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ g_{N_2}^{N_2} & g_{N_2-1}^{N_2} & \dots & g_{N_2-N_u-1}^{N_2} \end{bmatrix} \quad (1.10)$$

La matrice  $\mathbf{G}$  formée à partir des coefficients des polynômes  $G_j$  s'identifie en fait aux coefficients de la réponse indicielle du modèle  $g_i = g(i.T_e)$ ,  $T_e$  étant la période d'échantillonnage. Avec ces notations, le prédicteur optimal à  $j$ -pas peut s'écrire sous forme matricielle :

$$\hat{\mathbf{y}} = \mathbf{G}\tilde{\mathbf{u}} + \mathbf{if}(q^{-1})y(t) + \mathbf{ih}(q^{-1})\Delta u(t - 1) \quad (1.11)$$

#### 1.3.5. Critère quadratique matriciel

Le critère quadratique donné par l'équation (1.8) peut se combiner avec la relation (1.11) pour obtenir l'expression matricielle de ce critère :

$$\begin{aligned}
 J &= [\mathbf{G}\tilde{\mathbf{u}} + \mathbf{if}(q^{-1})y(t) + \mathbf{ih}(q^{-1})\Delta u(t - 1) - \mathbf{w}]^T \\
 &\quad [\mathbf{G}\tilde{\mathbf{u}} + \mathbf{if}(q^{-1})y(t) + \mathbf{ih}(q^{-1})\Delta u(t - 1) - \mathbf{w}] + \lambda \tilde{\mathbf{u}}^T \tilde{\mathbf{u}} \quad (1.12)
 \end{aligned}$$

### 1.3.6. Minimisation de la fonction de coût

La commande optimale s'obtient enfin par minimisation analytique du critère sous forme matricielle. La dérivée  $\frac{\partial J}{\partial u} = 0$  donne :

$$\tilde{\mathbf{u}} = \mathbf{M}[\mathbf{w} - \mathbf{if}(q^{-1})y(t) + \mathbf{ih}(q^{-1}) \Delta u(t - 1)] \quad (1.13)$$

avec les matrices:

$$\begin{aligned} \mathbf{M} &= \mathbf{QG}^T \text{ de dimension } N_u \times (N_2 - N_1 + 1) \\ \mathbf{Q} &= [\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}_{N_u}], \text{ de dimension } N_u \times N_u \end{aligned} \quad (1.14)$$

où  $\mathbf{I}_{N_u}$  est une matrice identité de dimension  $N_u \times N_u$

De façon classique en commande prédictive, seule la première valeur de la séquence  $\tilde{\mathbf{u}}$ , (équation (1.13)), est finalement appliquée au système en accord avec la stratégie de l'horizon fuyant, l'ensemble de la procédure étant effectué de nouveau à la période d'échantillonnage suivante.

$$\Delta u_{opt}(t) = -\mathbf{m}_1^T [\mathbf{if}(q^{-1})y(t) + \mathbf{ih}(q^{-1}) \Delta u(t - 1) - \mathbf{w}] \quad (1.15)$$

avec  $\mathbf{m}_1^T$  première ligne de la matrice  $\mathbf{M}$ .

La commande à appliqué sur le système est donnée par :

$$u(t) = u(t - 1) + \Delta u_{opt}(t) \quad (1.16)$$

### 1.4. Choix des paramètres de réglage

La définition du critère quadratique donnée par l'équation (1.8) a montré que l'utilisateur doit fixer quatre paramètres de réglage  $N_1$ ,  $N_2$ ,  $N_u$ , et  $\lambda$ . Ce choix des paramètres s'avère cependant délicat, car il n'existe pas de relations empiriques permettant de relier ces paramètres à des 'indicateurs' classiques en Automatique, tels que marges de stabilité ou bande passante ; tout au plus peut-on indiquer les grandes lignes permettant une approche d'un "bon réglage" [4].

### 1.4.1. Choix de l'horizon minimal de prédiction $N_1$

Le produit  $N_1 T_e$  est choisi égal au retard pur du système. Ainsi, 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  $N_0$  indiqué sur la figure 1.4.

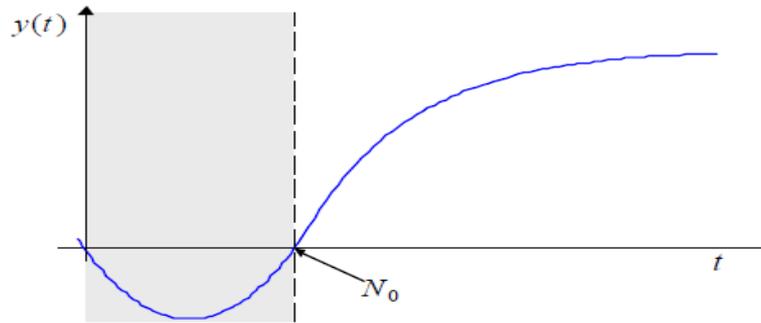


Figure 1.4 : choix de  $N_1$  pour un système à déphasage non minimal

### 1.4.2. Choix de l'horizon maximal de prédiction $N_2$

$N_2$  est choisi de sorte que le produit  $N_2 T_e$  soit limité par la valeur du temps de réponse souhaité. En effet augmenter la prédiction au delà du temps de réponse n'apporte aucune information supplémentaire. Par ailleurs, plus  $N_2$  est grand, plus le système corrigé est stable et lent. Notons sur le schéma de la figure 1.5 le choix de  $N_2$  dans le cas d'un système à déphasage non minimal.

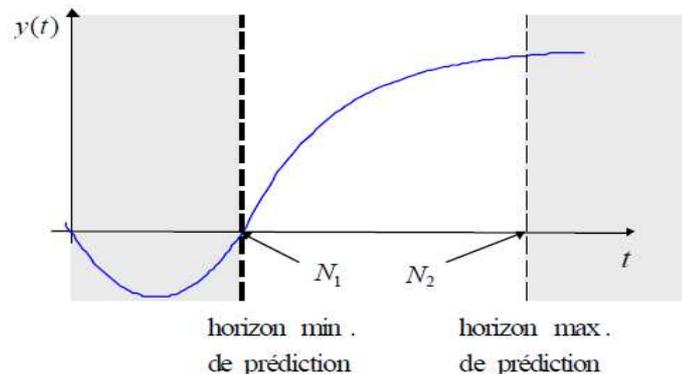


Figure 1.5 : Choix de  $N_2$  pour un système à déphasage non minimal

### 1.4.3. Choix de l'horizon de prédiction sur la commande $N_u$

Très souvent la valeur  $N_u = 1$  est suffisante pour beaucoup d'applications relativement simples. Dans ce dernier cas, le calcul de la séquence de commandes futures

se réduit au simple calcul du scalaire  $\tilde{u}_{opt}$ . En effet,  $N_u$  fixe la dimension des matrices à inverser dans le calcul du régulateur. De plus, pour des valeurs importantes de cet horizon de commande  $N_u$ , la marge de phase diminue. On choisira donc classiquement  $N_u$  égal à 1.

#### 1.4.4. Choix du facteur de pondération de la commande $\lambda$

On peut interpréter le facteur de pondération  $\lambda$  comme ‘l’équilibre de la balance’. En effet, si  $\lambda = 0$ , on minimise uniquement dans le critère quadratique, la différence entre la consigne et la sortie prédite. Il peut donc en résulter une commande très forte pouvant faire diverger le processus réel. D’autre part, si  $\lambda$  est très élevé, on pondère alors excessivement la commande qui n’est plus assez ‘dynamique’ pour obtenir le ralliement à la consigne. Dans le cadre monovarié, partant donc de la constatation que plus le gain d’un système est grand, plus la commande doit être pondérée (c’est à dire plus  $\lambda$  est important, et vice et versa), il apparaît alors clairement que  $\lambda$  est lié au gain du système. Une relation permettant de déterminer rapidement la valeur de  $\lambda$  apportant au système le maximum de stabilité est donnée ci-dessous :

$$\lambda_{opt} = trace(\mathbf{G}^T \mathbf{G}) \quad (1.17)$$

où  $\mathbf{G}$  est la matrice décrite par la relation (1.10).

#### 1.5. La commande prédictive avec contraintes

Le critère (1.12) 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 des contraintes terminales permettant de renforcer la stabilité [5].

Les contraintes qui interviennent dans la formulation GPC peuvent être décrites par l’intermédiaire d’un formalisme unifié [6][7]. Sachant que les incréments des commandes futures sont les variables qui subissent finalement les contraintes, il est intéressant d’exprimer toutes les limitations en se basant sur des modèles de structure similaire à la relation (1.2) afin de faciliter la construction des prédictions des signaux sous contraintes :

$$\begin{cases} A_\gamma(q^{-1})\gamma(t) = B_\gamma(q^{-1})u(t-1) \\ \gamma_{\min} \leq \hat{\gamma}(t+j) \leq \gamma_{\max}, \quad N_{c1} \leq j \leq N_{c2} \end{cases} \quad (1.18)$$

Avec  $A_\gamma, B_\gamma$  des polynômes (de degrés  $n_a^\gamma$  et  $n_b^\gamma$ ) en l'opérateur retard  $q^{-1}$ ,  $\gamma(t)$  le signal contraint,  $\gamma_{\min}, \gamma_{\max}$  les limitations à respecter.  $N_{c1}$  et  $N_{c2}$  représentent l'horizon de contrainte et peuvent être différents des horizons de prédiction ( $N_1, N_2$ ) ou de l'horizon de commande choisi  $N_u$ .

En utilisant  $(A_\gamma, B_\gamma)$ , il devient possible de construire les prédictions des signaux contraints  $\hat{\gamma}(t+j)$  comme la superposition d'une réponse libre et d'une réponse forcée :

$$\hat{\gamma}(t+j) = \underbrace{F_{\gamma_j}(q^{-1})\gamma(t) + H_{\gamma_j}(q^{-1})\Delta u(t-1)}_{\text{réponse libre}} + \underbrace{G_{\gamma_j}(q^{-1})\Delta u(t+j-1)}_{\text{réponse forcées}} \quad (1.19)$$

ou sous forme vectorielle :

$$\hat{\gamma} = \mathbf{f}_{\gamma_j}(q^{-1})\gamma(t) + \mathbf{h}_{\gamma_j}(q^{-1})\Delta u(t-1) + \mathbf{G}_{\gamma_j}\tilde{u} \quad (1.20)$$

Les vecteurs  $\mathbf{f}_{\gamma_j}, \mathbf{h}_{\gamma_j}, \mathbf{G}_{\gamma_j}$ , contiennent les coefficients des polynômes  $F_{\gamma_j}(q^{-1}), H_{\gamma_j}(q^{-1}), G_{\gamma_j}(q^{-1})$  solutions d'équations diophantiennes.

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

### 1.5.1. Contraintes sur l'incrément de commande

On s'intéresse ici aux contraintes de la forme :

$$\Delta u_{\min} \leq u(t+j) - u(t+j-1) \leq \Delta u_{\max}, \quad N_{c1} \leq j \leq N_{c2}$$

Le modèle de prédiction pour le signal contraint peut se construire à partir de (1.18) avec  $A_\gamma = 1, B_\gamma = \Delta$

$$\begin{cases} \hat{\gamma}(t+1) = \Delta u(t) \\ \Delta u_{\min} \leq \hat{\gamma}(t+1) \leq \Delta u_{\max} \end{cases} \quad N_{c1} \leq j \leq N_{c2} \quad (1.21)$$

Les solutions des équations diophantiennes sont alors :

$$F_{\gamma_j}(q^{-1}) = 0, \quad H_{\gamma_j}(q^{-1}) = 0, \quad G_{\gamma_j}(q^{-1}) = q^{j-1}$$

soit, en les regroupant sous forme matricielle :

$$G_\gamma = \mathbf{I}_{N_u \times N_u}, \text{if}_\gamma = 0, \text{ih}_\gamma = 0$$

L'expression matricielle pour ce type de contraintes :

$$\mathbf{I}\Delta u_{\min} \leq G\tilde{u} \leq \mathbf{I}\Delta u_{\min}$$

avec  $\mathbf{I} \in \mathfrak{R}^{N_u \times 1}$  vecteur composés d'éléments unitaires.

### 1.5.2. Contraintes sur l'amplitude de commande

On s'intéresse ici aux contraintes de la forme :

$$u_{\min} \leq u(t+j) \leq u_{\max}, \quad N_{c1} \leq j \leq N_{c2}$$

Le modèle de prédiction pour le signal contraint peut se construire à partir de (1.18) avec  $A_\gamma = 1, B_\gamma = 1$  :

$$\begin{cases} \gamma(t) = u(t) \\ u_{\min} \leq \hat{\gamma}(t+j) \leq u_{\max}, \quad N_{c1} \leq j \leq N_{c2} \end{cases}$$

Les solutions des équations diophantiennes sont alors

$$H_{\gamma_j} = 0, F_{\gamma_j} = 1, G_{\gamma_j} = 1 + q^{-1} + q^{-2} + \dots + q^{-j+1}$$

soit, en les regroupant sous forme matricielle :

$$G_\gamma = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{bmatrix}; \text{if}_\gamma = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \text{ih}_\gamma = 0$$

l'expression matricielle pour ce type de contraintes :

$$\mathbf{I}\Delta u_{\min} \leq [\text{if}_\gamma \ G_\gamma] \begin{bmatrix} \gamma(t) \\ \tilde{u} \end{bmatrix} \leq \mathbf{I}\Delta u_{\min} \quad ; \quad \gamma(t) = u(t-1) \quad (1.22)$$

### 1.5.3. Contraintes sur la sortie

On s'intéresse ici aux contraintes de la forme :

$$y_{min} \leq y(t+j) \leq y_{max}, \quad N_{c1} \leq j \leq N_{c2}$$

Les prédictions pour le signal contraint sont construites avec le même modèle utilisé lors de la construction du critère, donc  $A_\gamma = A$ ,  $B_\gamma = B$

$$\begin{cases} A(q^{-1})\gamma(t) = B(q^{-1})u(t-1) \\ y_{min} \leq \hat{\gamma}(t+j) \leq y_{max}, \quad N_{c1} \leq j \leq N_{c2} \end{cases} \quad (1.23)$$

Les solutions des équations diophantiennes de sorte que la formulation compacte des contraintes sur la sortie est donnée par :

$$y_{min} \leq [\mathbf{if}_\gamma \mathbf{ih}_\gamma \mathbf{G}_\gamma] \begin{bmatrix} \boldsymbol{\gamma}(t) \\ \Delta \mathbf{u}(t-1) \\ \tilde{\mathbf{u}} \end{bmatrix} \leq y_{max}; \quad \gamma(t) = y(t) \quad (1.24)$$

### 1.5.4. Contraintes sur le dépassement

On s'intéresse ici aux contraintes de la forme :

$$y(t+j) \leq w(t+j), \quad N_{c1} \leq j \leq N_{c2}$$

Il s'agit encore une fois de contraintes sur la sortie, et donc le modèle est donné par  $A_\gamma = A$ ,  $B_\gamma = B$ , avec la particularité que la consigne à suivre intervient également :

$$\begin{cases} \gamma(t) = y(t) \\ \hat{\gamma}(t+j) \leq w(t+j), \quad N_{c1} \leq j \leq N_{c2} \end{cases} \quad (1.25)$$

Les solutions des équations diophantiennes est donnée par :

$$[\mathbf{if}_\gamma \mathbf{ih}_\gamma \mathbf{G}_\gamma - \mathbf{I}_{(N_{c2}-N_{c1}+1) \times (N_{c2}-N_{c1}+1)}] \begin{bmatrix} \mathbf{y}(t) \\ \Delta \mathbf{u}(t-1) \\ \tilde{\mathbf{u}} \\ \mathbf{w} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} w(t+N_{c1}) \\ \vdots \\ w(t+N_{c2}) \end{bmatrix} \quad (1.26)$$

### 1.5.5. Contraintes terminales de type égalité

On s'intéresse ici aux contraintes de la forme :

$$y(t + j) \leq w(t + N_2), \quad N_2 < N_{c1} \leq j \leq N_{c2}$$

Les prédictions des sorties sont toujours construites à partir du modèle nominal  $A_\gamma = A$ ,  $B_\gamma = B$  en étendant cependant la portée de la prédiction au delà de l'horizon de prédiction nécessaire pour le critère de coût ( $N_2 < N_{c1} \leq j \leq N_{c2}$ )

$$\begin{cases} \gamma(t) = y(t) \\ \hat{\gamma}(t + j) = w_{N_2}, \quad N_2 < N_{c1} \leq j \leq N_{c2} \end{cases} \quad (1.27)$$

Les prédictions sont construites à partir des polynômes  $F_{\gamma_j}(q^{-1})$ ,  $H_{\gamma_j}(q^{-1})$ ,  $G_{\gamma_j}(q^{-1})$ , mais pour  $j > N_2$ . Par exemple la matrice  $\mathbf{G}_\gamma$  pour  $N_2 < j$  s'exprime par :

$$\mathbf{G}_\gamma = \begin{bmatrix} \mathbf{g}_{N_{c1}} & \cdots & \mathbf{g}_{N_{c1}-N_u+1} \\ \vdots & \ddots & \vdots \\ \mathbf{g}_{N_{c2}} & \cdots & \mathbf{g}_{N_{c2}-N_u+1} \end{bmatrix} \in \mathfrak{R}^{(N_{c2}-N_{c1}+1) \times N_u}$$

Avec ces éléments, les contraintes de type égalité à la fin de l'intervalle de prédiction se reformulent par :

$$[\mathbf{if}_\gamma \ \mathbf{ih}_\gamma \ \mathbf{G}_\gamma - \mathbf{J}] \begin{bmatrix} y(t) \\ \Delta u(t-1) \\ \tilde{u} \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbf{J} = \begin{bmatrix} \mathbf{0} \in \mathfrak{R}^{(N_{c2}-N_{c1}+1) \times (N_{c2}-N_{c1})} \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w(t + N_{c1}) \\ \vdots \\ w(t + N_{c2}) \end{bmatrix} \quad (1.28)$$

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 sur l'expression sur lesquelles elles agissent, soit ici  $\tilde{u}$ .

## 1.6. Conclusion

Puisque toutes les dimensions des vecteurs et matrices précédents sont liées au modèle du procédé, les exigences de connaissance requises pour la mise en œuvre de la structure prédictive sont les suivantes :

- Degrés des numérateur et dénominateur  $B, A$  fixés
- Retard pur et temps de réponse du procédé connus
- Période d'échantillonnage choisie préalablement

Ainsi les lois de commande prédictive généralisée permettent de garantir les performances requises pour une classe d'erreurs de modélisation aussi large que possible. Dans les chapitres suivants nous allons voir comment utiliser le principe de la commande prédictive généralisé pour la commande des systèmes non linéaires.

## CHAPITRE 2

# IDENTIFICATION ET COMMANDE A BASE DE RESEAUX DE NEURONES

### 2.1. Introduction

Les réseaux de neurones, constituent une approche permettant d'aborder les problèmes de perception, de mémoire, d'apprentissage et de raisonnement. Grâce à leur traitement parallèle de l'information et à leurs mécanismes inspirés des cellules nerveuses (neurones), ils infèrent des propriétés émergentes permettant de solutionner des problèmes complexes.

Dans ce chapitre, nous présentons les éléments de base constituant les réseaux de neurones, en particulier nous considérons les réseaux statiques multicouches utilisés dans notre travail. Ensuite nous exposons d'une manière détaillée les différentes étapes de l'identification des systèmes en utilisant les réseaux de neurones. La dernière partie de ce chapitre est consacrée aux méthodes de commande à base des réseaux de neurones.

### 2.2. Réseaux de neurones

Les réseaux de neurones artificiels, sont composés d'un ensemble structuré d'unités de traitement, appelées *neurones*, qui fonctionnent en parallèle et qui sont fortement interconnectées. Ces neurones réalisent chacun une fonction non linéaire de leurs entrées, qui est déterminée par un ensemble de paramètres dont les valeurs sont établies à la suite d'un apprentissage par l'exemple. Les non-linéarités présentes dans ces systèmes connexionnistes, ainsi que l'apprentissage discriminant qu'ils subissent, rendent ces derniers particulièrement bien adaptés pour la modélisation et le contrôle des systèmes non linéaires.

### 2.2.1. Modèle d'un neurone

Le modèle mathématique d'un neurone artificiel est illustré à la figure 2.1. Un neurone est essentiellement constitué d'un intégrateur qui effectue la somme pondérée de ses entrées. Le résultat  $s$  de cette somme est ensuite transformée par une fonction d'activation  $f$  qui produit la sortie  $y$  du neurone, les  $n$  entrées du neurones correspondent au vecteur  $X^T = [x_1 \ x_2 \ \dots \ x_n]$ , alors que  $W^T = [w_1 \ w_2 \ \dots \ w_n]$  représente le vecteur des poids du neurone. La sortie  $s$  de l'intégrateur est donnée par l'équation suivante :

$$s = \sum_{j=1}^n w_j x_j - b \quad (2.1)$$

$x_j$  : la valeur de la  $j^{eme}$  entrée du neurone,

$w_j$  : le poids synaptique correspondant au neurone  $j$ ,

$b$  : désigne le biais,

$f(.)$  : la fonction d'activation. La fonction la plus utilisée est de type sigmoïde.

Que l'on peut aussi écrire sous forme matricielle :

$$s = w^T x - b \quad (2.2)$$

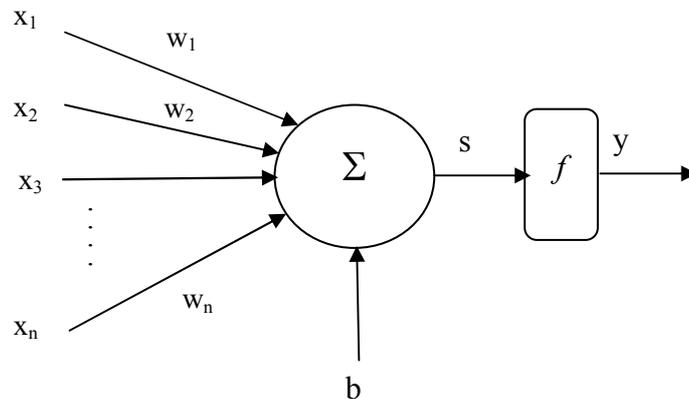


Figure 2.1: Modèle d'un neurone artificiel

Cette sortie correspond à une somme pondérée des poids et des entrées moins ce qu'on nomme le biais  $b$  du neurone. Le résultat  $s$  de la somme pondérée s'appelle le niveau d'activation du neurone. Le biais  $b$  s'appelle aussi le seuil d'activation du neurone. Lorsque le niveau d'activation atteint ou dépasse le seuil  $b$ , alors l'argument de  $f$  devient positif (ou nul). Sinon, il est négatif.

Pour pouvoir simuler un réseau de neurones, nous allons rendre le temps discret dans nos équations. Autrement dit, nous allons supposer que tous les neurones sont synchrones, c'est-à-dire qu'à chaque temps  $t$ , ils vont simultanément calculer leur somme pondérée et produire une sortie  $y(t) = f(s(t))$ . Dans les réseaux biologiques, tous les neurones sont en fait asynchrones.

Revenons donc à notre modèle tel que formulé par l'équation (2.2) et ajoutons la fonction d'activation  $f$  pour obtenir la sortie du neurone:

$$y = f(s) = f(\mathbf{w}^T \mathbf{x} - b) \quad (2.3)$$

### 2.2.2. Fonctions d'activation

Il existe plusieurs fonctions de d'activation, dans notre travail nous avons utilisé les fonctions, «linéaire» et «tangente hyperbolique».

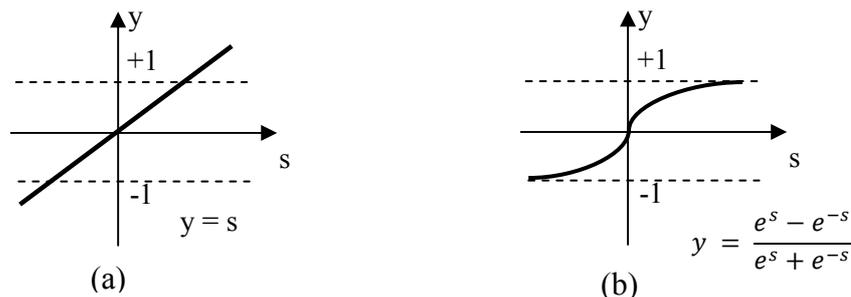


Figure 2.2 : Fonction d'activation : (a) «linéaire», (b) «tangente hyperbolique».

### 2.2.3. Réseau de neurones multicouches

Les réseaux de neurones artificiels regroupent en réseaux un certain nombre de neurones connectés entre eux de diverses manières : réseaux totalement interconnectés, réseaux à couches, réseaux statiques, réseaux récurrents (dynamique)...etc.

Un réseau multicouche MLP (Multi Layer Perceptron) est un réseau dont l'architecture est basée sur l'idée de grouper des neurones dans une couche, ensuite plusieurs couches sont placées bout à bout et les neurones de deux couches adjacentes sont connectés. Ces couches sont organisées tel que :

**La couche d'entrée :** Elle recevra les données source que l'on veut utiliser pour l'analyse. Sa taille est donc directement déterminée par le nombre de variables d'entrées.

**Les couches cachées :** Le MLP peut comporter une ou plusieurs couches cachées, en ce sens qu'elle n'a pas de contact direct avec l'extérieur. Les fonctions d'activations sont en général non linéaires sur cette couche mais il n'y a pas de règle à respecter. Le choix de sa taille n'est pas implicite et elle doit être ajustée.

**La couche de sortie :** Elle donne le résultat obtenu après compilation par le réseau des données entrées dans la première couche. Sa taille est directement déterminée par le nombre de variables qu'on veut en sortie.

Un réseau de neurones MLP avec une seule couche cachée dotée de  $n_h$  nœuds et ayant comme vecteur des signaux d'entrée  $X$  de  $n_e$  valeurs et comme sortie  $y$  est schématisé dans la figure 2.3. La valeur analytique de sa sortie peut être calculée par l'équation suivante:

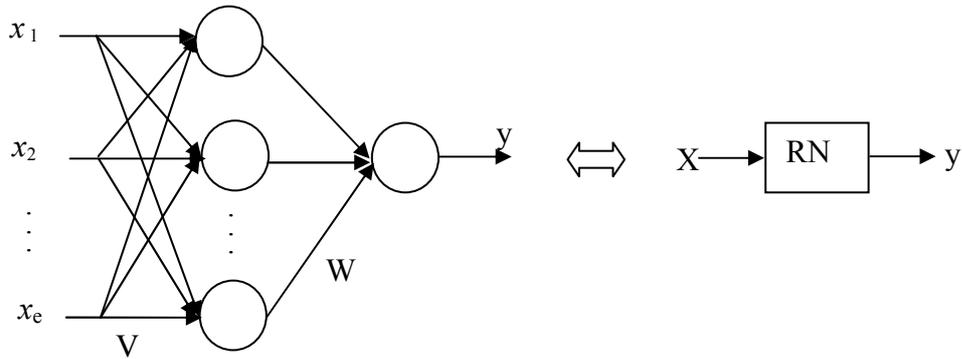


Figure 2.3 : Structure du réseau de neurones MLP

$$y = W^T \cdot f(V^T \cdot X) \quad (2.4)$$

Avec  $V$  une matrice de poids du réseau reliant la couche d'entrée à la couche cachée unique et  $W$  un vecteur de poids du réseau reliant la couche cachée à la sortie. La fonction d'activation peut être de type hyperbolique, l'équation (2.4) devient :

$$y = \sum_{j=1}^{n_h} w_j \tanh \left[ \sum_{k=1}^{n_e} v_{jk} x_k + b_{j0} \right] + b_0 \quad (2.5)$$

Il a été démontré qu'un réseau MLP avec seulement une couche cachée dont la fonction d'activation est de type sigmoïde et une sortie avec une fonction d'activation linéaire peut approximer n'importe quelle fonction continue pourvu qu'un nombre

suffisant de neurones soit fourni [8][9][13]. Cette propriété est dite propriété d'approximation universelle des réseaux de neurones.

#### 2.2.4. Apprentissage

Parmi les propriétés désirables pour un réseau de neurones, la plus fondamentale est sûrement la capacité d'apprendre de son environnement, d'améliorer sa performance à travers un processus d'apprentissage.

L'apprentissage se traduit par une modification de l'efficacité synaptique, c'est-à-dire par un changement dans la valeur des poids qui relient les neurones d'une couche à l'autre, ceux-ci provoquent dans le futur une réponse nouvelle vis-à-vis de l'environnement. Ainsi, le réseau peut s'améliorer avec le temps.

Les différentes méthodes d'apprentissage sont des procédures d'estimation itératives. L'ajustement des paramètres s'opère à chaque itération en fonction de leur participation à une fonction de l'erreur  $E$ , de sorte à minimiser l'écart entre la sortie calculée  $\hat{y}$  et une sortie mesurée  $y$ . Pour notre part, nous sommes limités à l'emploi du critère quadratique, sans doute le plus répandu dans le domaine de l'identification

$$E = 1/2N \sum_{t=1}^N (y(t) - \hat{y}(t))^2 \quad (2.6)$$

Au niveau des réseaux multicouches, il a été défini deux grandes classes d'algorithmes d'apprentissage [9] :

- Dans la première, les algorithmes sont basés sur le calcul de dérivée du critère (2.6), tel que l'algorithme du gradient (une méthode du premier ordre), les méthodes de Gauss-Newton (du second ordre), et l'algorithme de Levenberg-Marquardt (il représente un compromis entre les méthodes du premier ordre et celles du second ordre).

- Dans la deuxième classe, on trouve les méthodes d'apprentissage utilisant les algorithmes d'optimisation ne nécessitant pas le calcul de la dérivée. Les procédures d'apprentissage peuvent se subdiviser, elles aussi, en trois grandes catégories: apprentissage supervisé, apprentissage par renforcement et apprentissage non supervisé.

- ***apprentissage supervisé:*** Dans ce cas, un superviseur (ou expert humain) fournit une valeur ou un vecteur de sortie (appelé cible ou sortie désirée), que le réseau de neurones doit associer à un vecteur d'entrée. L'apprentissage consiste dans ce cas à ajuster

les paramètres du réseau afin de minimiser l'erreur entre la sortie désirée et la sortie réelle du réseau.

- ***apprentissage semi-supervisé ou apprentissage par renforcement***: Ce mode d'apprentissage, suppose qu'un comportement de référence n'est pas disponible, mais en revanche, il est possible d'obtenir des indications qualitatives (vrai, faux, ...) sur les performances du réseau.

- ***apprentissage non supervisé***: Dans cet apprentissage, les données ne contiennent pas d'information sur la sortie désirée. Il n'y a pas de superviseur. La tâche du réseau consiste, par exemple dans ce cas, à créer des regroupements de données selon des propriétés communes (classification).

### 2.3. Identification des systèmes par réseaux de neurones

La construction des lois de commande prédictive nécessite l'existence d'un modèle qui représente le mieux possible le comportement dynamique du processus à commander. Pour atteindre ce but, les réseaux de neurones représentent une bonne solution, vu leur capacités d'approximation universelle des fonctions non linéaires.

L'identification des systèmes non linéaires par réseaux de neurones a fait l'objet de nombreuse recherche depuis plusieurs années. Cette nouvelle approche fournit une description mathématique très proche de la réalité des systèmes non linéaires[10][11].

#### 2.3.1. Définition

L'identification, est la détermination du modèle dynamique d'un système à partir des données entrées /sorties, autrement dis si  $Z(t)$  est l'ensemble de mesures entrées/sortie :

$$Z(t) = \{y(1), u(1), \dots \dots \dots \dots \dots, y(t), u(t)\} \quad (2.7)$$

Identifier un système revient à trouver une application  $f$  de la forme :

$$\hat{y}(t) = f(Z(t - 1)) \quad (2.8)$$

où  $\hat{y}(t)$  est la sortie estimée de la sortie réel  $y(t)$ .

Si on dispose d'un système dont on connaît les entrées et les sorties quantitatives, à partir d'un certain nombre d'exemples fournis aux réseaux, ceux-ci apprennent à se

comporter comme le système. L'apprentissage de ces réseaux peut ensuite être aisément renforcé sur de nouveaux exemples. Autrement dit l'objectif de l'identification neuronale des systèmes non linéaires est d'approcher la fonction  $f$  dans la relation (2.8) en utilisant un modèle paramétrique de la forme :

$$\hat{y}(t/\theta) = f(\varphi(t)/\theta) \quad (2.9)$$

où  $\theta$  : est le vecteur des paramètres à déterminer. Dans notre cas, sont les poids synaptiques du réseau de neurone utilisé.

$\varphi(t)$  : est le vecteur des mesures entrées/sortie de dimension fixe, il est appelé vecteur des régresseurs, il est donnée par :

$$\varphi(t) = [y(t-1), \dots, y(t-n), u(t-1), \dots, u(t-m)] \quad (2.10)$$

Les principales étapes à suivre pour concevoir un modèle sont données par la figure 2.4.

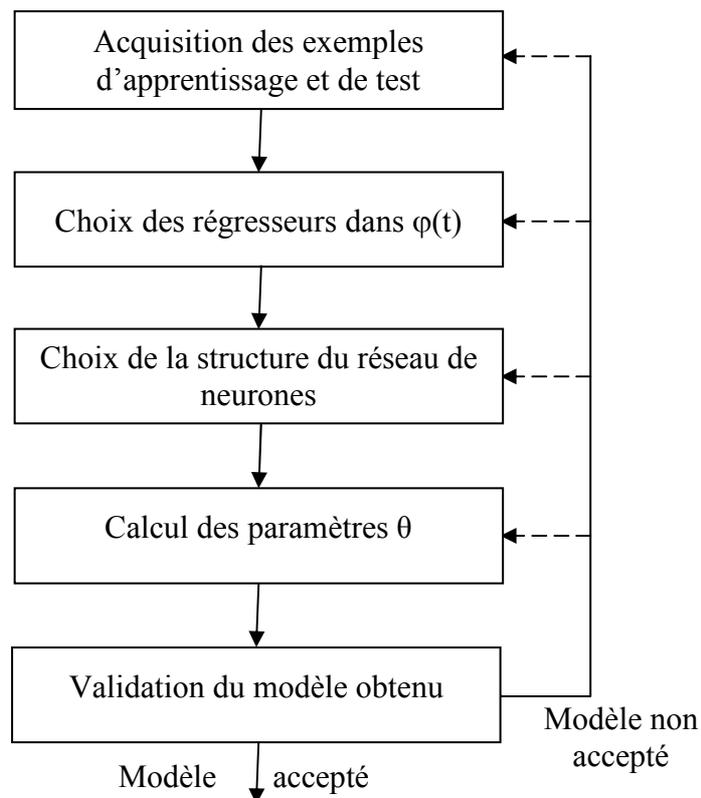


Figure 2.4: Les principales étapes de l'identification

### 2.3.2. Acquisition des exemples d'apprentissage et de test

La précision et la robustesse du modèle à identifier est conditionnée par la richesse et la consistance des variables expérimentales qui caractérisent le système. Pour cela, on doit appliquer un signal d'entrée riche en amplitude et en fréquences. Dans cet étape on a besoin de :

#### 2.3.2.1. Choisir la fréquence d'échantillonnage

Le choix de la période d'échantillonnage repose à la fois sur des considérations pratiques et sur une comparaison des spectres des signaux obtenus par la transformation de Fourier. Le spectre du signal échantillonné doit être proche de celui du signal brut et les spectres secondaires engendrés par l'échantillonnage, atténués et décalés, doivent respecter le théorème de Shannon (un signal de fréquence  $f_0$  est parfaitement défini par ses valeurs échantillonnées si  $f_e$  vérifie l'inégalité :  $f_e \geq 2f_0$ , où  $f_e$  est la fréquence d'échantillonnage)

#### 2.3.2.2. Concevoir le signal d'excitation

Le signal d'entrée choisi ne doit pas exciter les modes dynamiques du système à ne pas inclure dans le modèle. Un exemple des signaux d'entrée utilisés en identification est donné par la figure 2.5.

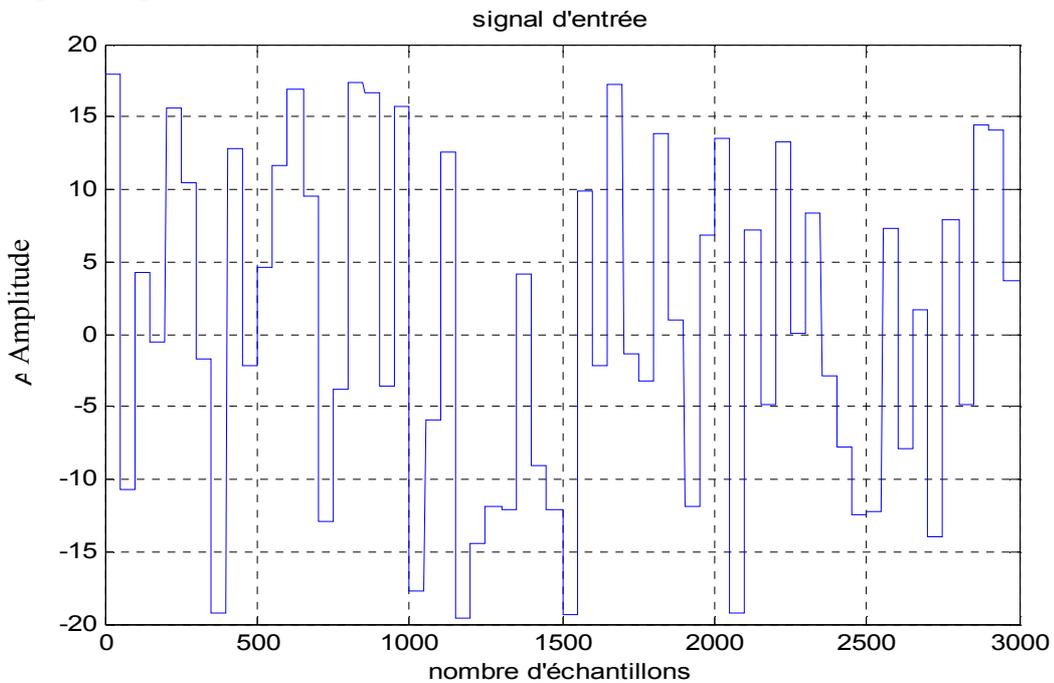


Figure 2.5: Exemple de signaux d'excitation

### 2.3.2.3. Prétraitement des mesures obtenues

Pour éliminer d'éventuelles perturbations ou bruit, des prétraitements des données sont nécessaires (filtrage, normalisation, interpolation..).

### 2.3.3. Choix des régresseurs

Il existe plusieurs techniques pour guider le choix des régresseurs[9][12]. Par contre, il n'a pas une méthode générale qui permet leur construction optimale. Les étapes principales de la procédure de choix des régresseurs sont : la sélection des régresseurs candidats puis l'évaluation de ces régresseurs sélectionnés.

- ***La sélection des régresseurs*** : Deux grandes classes des méthodes de sélection des régresseurs candidats parmi l'ensemble de tous les régresseurs possibles: les méthodes heuristiques et les méthodes pseudo aléatoires. Dans les méthodes heuristiques on peut envisager deux approches différentes :

- Dans la première approche, le modèle est construit en ajoutant à chaque fois une nouvelle entrée; cette approche est adoptée dans le cas des réseaux de neurones à cause de la taille réduite du modèle d'évaluation.

- Dans la deuxième approche, le modèle est construit en utilisant toute les entrées possibles, ensuite les régresseurs dont leur impact sur les performances du modèle est faible, seront éliminés.

Dans les méthodes pseudo aléatoires les régresseurs candidats sont obtenus en utilisant les méthodes de la sélection génétique.

- ***L'évaluation des régresseurs sélectionnés***: pour évaluer la qualité des régresseurs sélectionnés, deux types des méthodes sont distingués:

- Les méthodes basées sur le modèle: c'est-à-dire de construire d'abord le modèle pour pouvoir ensuite évaluer la qualité des régresseurs;

- Les méthodes indépendantes du modèle: dans ce cas la qualité des régresseurs est évaluée en se basant seulement sur les mesures entrées/sorties.

Les méthodes de sélection et d'évaluation des régresseurs sont données d'une manière détaillée dans [9][14].

#### 2.3.4. Choix de la structure du modèle

L'identification par réseaux de neurones requiert à priori la sélection des entrées du réseau, ainsi que sa topologie interne. Une des formes les plus simples d'une telle topologie, est le Perceptron Multicouches (MLP), sa capacité d'approximation universelle a fait de lui un bon choix pour faire la modélisation des systèmes. Plusieurs modèles ont été proposés :

- NFIR (Non linear Finite Impulse Response): la régression est composée uniquement des entrées passées :

$$\hat{y}(t) = f(u(t-1), \dots, u(t-n)) \quad (2.11)$$

- NARX (Non linear AutoRegressive with eXternal input): dans ce cas la régression est composée des entrées et sorties passées (c'est le modèle série parallèle):

$$\hat{y}(t) = f(u(t-1), \dots, u(t-n), y(t-1), \dots, y(t-m)) \quad (2.12)$$

- NOE (Non linear Output Error): la régression est composée des entrées passées et des sorties estimées passées (c'est le modèle parallèle):

$$\hat{y}(t) = f(u(t-1), \dots, u(t-n), \hat{y}(t-1), \dots, \hat{y}(t-m)) \quad (2.13)$$

- NARMAX (Non linear AutoRegressive, Moving Average with eXternal input): la régression est composée des sorties et entrées passées ainsi que des erreurs d'estimation:

$$\hat{y}(t) = f(u(t-1), \dots, u(t-n), y(t-1), \dots, y(t-m), e(t-1), \dots, e(t-l)) \quad (2.14)$$

Le choix d'un tel modèle dépend de l'application, des informations disponibles et de la complexité du modèle. Le modèle NARX, appelé aussi modèle série-parallèle est le plus utilisé vu sa simplicité et sa structure non récursive (Figure 2.6). Il peut aussi représenter une large classe des systèmes non linéaires discrets [15].

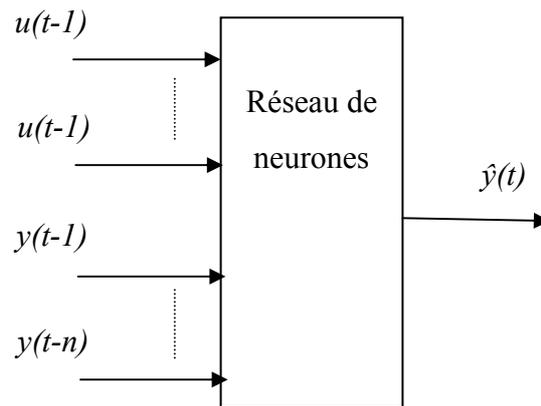


Figure 2.6: Le modèle série-parallèle (NARX)

### 2.3.5. Calcul des paramètres

Le calcul des paramètres du modèle revient à déterminer les poids des connexions du réseau de neurones utilisé. Ils sont déterminés par apprentissage de telle sorte à minimiser un critère de performance.

Les algorithmes d'apprentissage pour les réseaux multicouches sont nombreux et le choix de l'un de ces algorithmes dépend de la nature d'identification. L'algorithme de Levenberg-Marquardt, même s'il ne fournit aucune garantie d'atteindre le minimum global, est cependant largement recommandé [9].

### 2.3.6. Validation du modèle

L'objectif de cette étape est de déterminer si le modèle obtenu représente convenablement le système non linéaire considéré. On distingue plusieurs critères de validation; la plupart de ces critères utilisent un ensemble de données (données de validation) qui n'ont pas été utilisées dans la phase d'apprentissage. Le critère typique le plus utilisé est celui de l'erreur de prédiction :  $\varepsilon(t) = y(t) - \hat{y}(t/\theta)$  sa valeur doit être au dessous d'un certain seuil.

### 2.3.7. Exemple

On considère l'identification du système non linéaire suivant :

$$\ddot{y}(t) + \dot{y}(t) + y(t) + y^3(t) = u(t) \quad (2.12)$$

Le réseau de neurones utilisé est de type MLP à une seule couche cachée, les neurones de cette couche ont comme fonction d'activation la tangente hyperbolique. Les neurones de la couche de sortie possèdent une fonction d'activation linéaire. L'apprentissage, de type supervisé, est effectué en utilisant l'algorithme de Levenberg-Marquardt et un ensemble de 2000 paires de mesures entrée/sortie. Nous avons utilisé un autre ensemble de 1000 paires de mesures entrée/sortie pour la validation du modèle. Les données d'apprentissages et de validation sont générées par le signal d'entrée donnée par la figure 2.5. La réponse du système à ce signal est présentée sur la figure 2.7

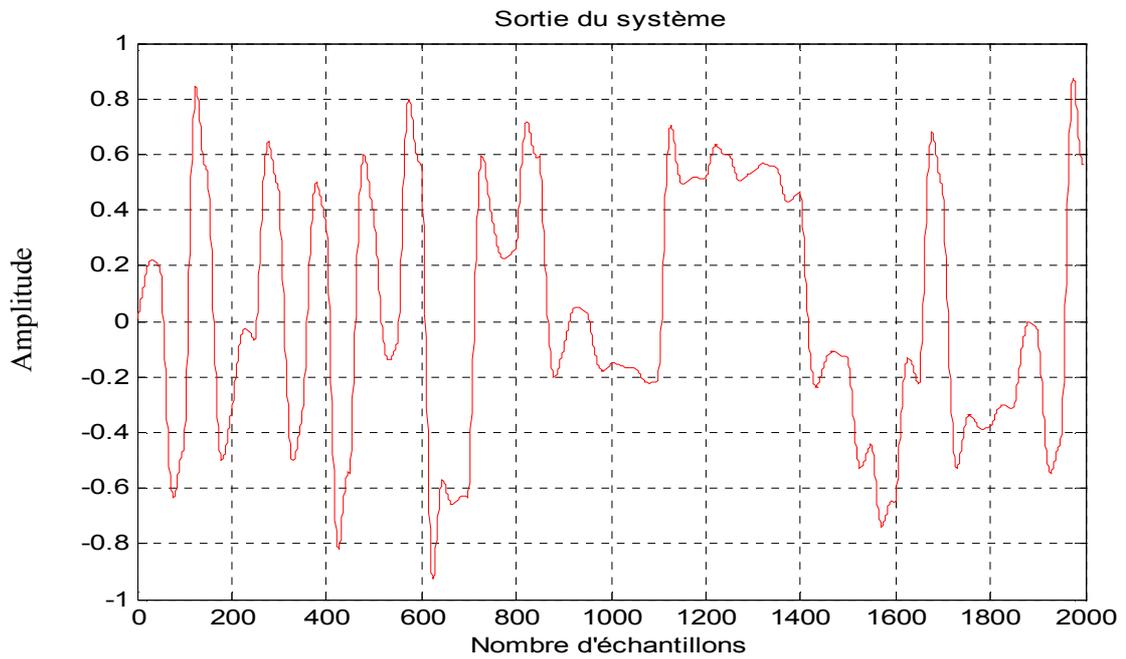


Figure 2.7 : La réponse du système à l'entrée d'excitation

Le modèle d'identification utilisé est donné par l'équation suivante :

$$\hat{y}(t) = NN(u(t-1), u(t-2), y(t-1), y(t-2)) \quad (2.13)$$

où  $NN$  comporte 10 neurones sur sa couche cachée

La réponse du modèle obtenu à l'entrée de teste ainsi que la réponse réelle du système sont présentées sur la figure 2.8. Les deux courbes sont presque superposées et l'écart entre eux est très faible.

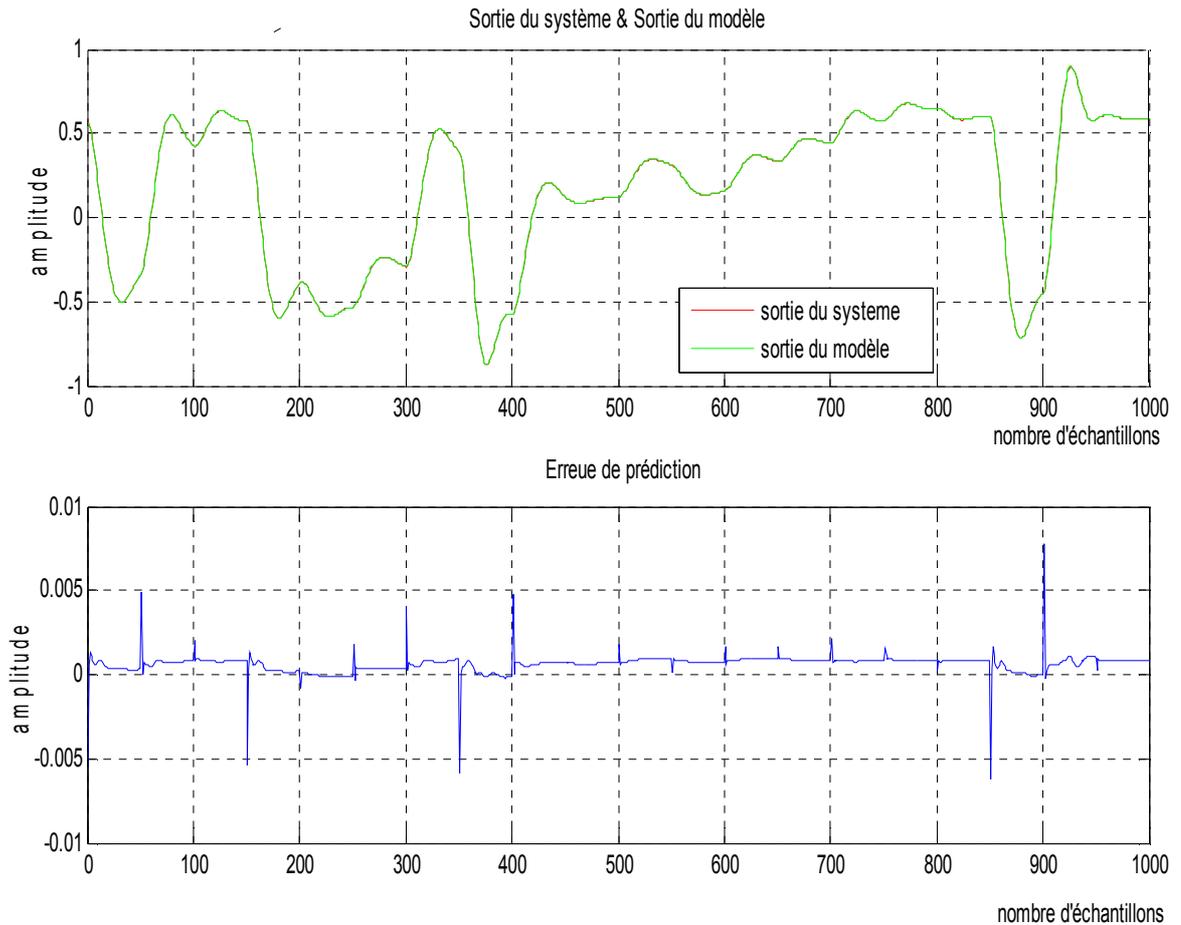


Figure 2.8 : Validation du modèle obtenu

#### 2.4. La commande neuronale

Commander un système, c'est faire en sorte qu'il se comporte d'une manière désirée en réponse à une sollicitation précise. Cette tâche nécessite la surveillance continue d'un ensemble de variables, ainsi que le respect d'un certain nombre de contraintes opératoires.

Le recours aux méthodes de commandes par apprentissage est devenu une nécessité car les systèmes de commande obtenus ainsi procèdent par collecter des données empiriques, stocker et extraire les connaissances contenues dans celle-ci et utiliser ces connaissances pour réagir à de nouvelles situations: on est passé à la commande intelligente. Dans ces dernières années, un grand intérêt a été porté à la commande intelligente utilisant les réseaux de neurones ; ceci est dû, principalement, à la capacité d'apprentissage et de généralisation des réseaux de neurones ainsi qu'à leur parallélisme.

On peut distinguer deux classes de structures de commande : la commande directe et la commande indirecte.

### 2.4.1. Commande directe

Directe signifie que le contrôleur est le réseau de neurone lui-même. Plusieurs structures à base de cette idée ont été proposées. Parmi elles nous citons les structures suivantes:

#### 2.4.1.1. Commande supervisée

Cette technique de commande consiste à imiter un contrôleur déjà existant (peut être un opérateur humain). La sortie du contrôleur à imiter sera une fonction à approcher par un réseau de neurone (figure 2.9.a). Une fois cette tâche accomplie, le réseau sera implanté à la place de l'ancien contrôleur pour conduire le système en imitant son comportement (figure 2.9.b). Mais pourquoi nous voudrions copier un contrôleur existant qui fait déjà le travail ? La plus part des contrôleurs classiques sont calculés autour d'un point de fonctionnement donné. Ceci signifie que les contrôleurs peuvent fonctionner correctement autour d'un certain point et ils échoueront s'il y a n'importe quelle sorte d'incertitude ou de changement. L'avantage de la commande neuronale est si une incertitude se produit le réseau de neurones pourra adapter ces paramètres de telle sorte à préserver les performances de commande.

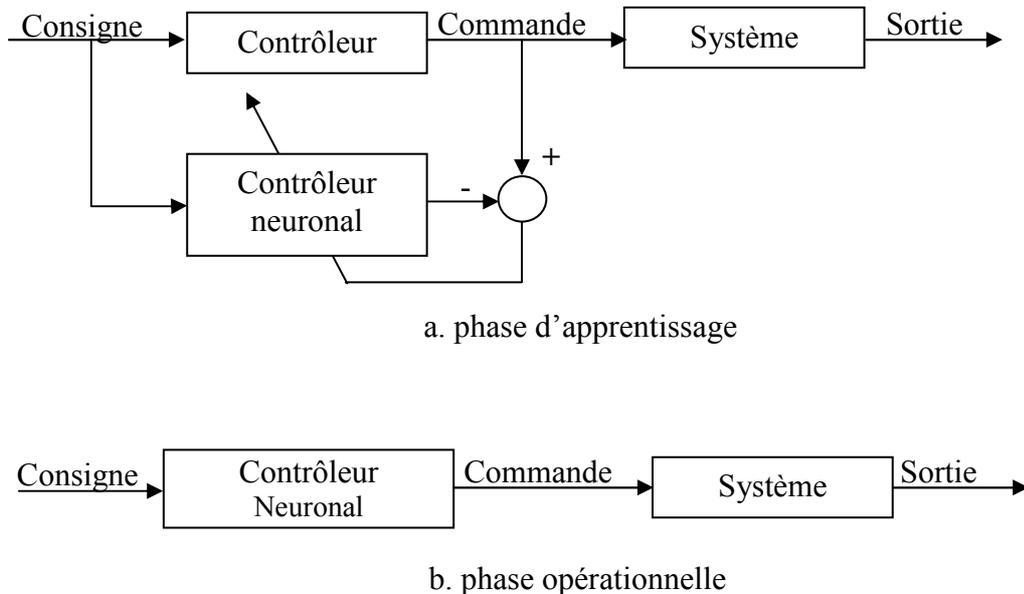


Figure 2.9 : Structure de la commande supervisée

### 2.4.1.2. Commande directe par modèle inverse

La plus simple méthode pour construire un système de commande neuronal, à partir d'un modèle du système identifié sous forme d'un réseau de neurones en boucle ouverte est l'inversion directe du modèle. Le système de commande est alors simplement l'inverse du modèle du processus. La sortie du système est employée comme une entree au réseau. La sortie du réseau de neurone est comparée au signal de commande et l'erreur, est ensuite utilisée comme moyen pour ajuster les poids du réseau (figure 2.10). Cette méthode force le réseau de neurones à apprendre le modèle inverse du système

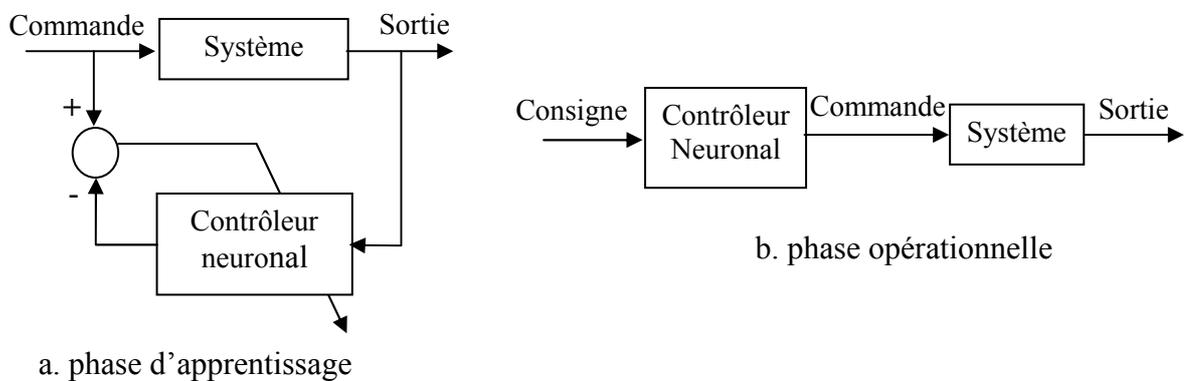


Figure 2.10 : Commande directe par modèle inverse

### 2.4.1.3. Commande par modèle interne (interna model control IMC)

Comme son nom l'indique, la commande par modèle interne met en jeu, autre que le contrôleur, un modèle du système, dit modèle interne. Les deux réseaux dont l'un est le modèle directe et l'autre le modèle inverse du système sont placés, après la phase d'apprentissage, dans de la boucle de régulation de la manière indiquée par la figure 2.11. Ceci, permettra d'assurer la réjection de certaines perturbations.

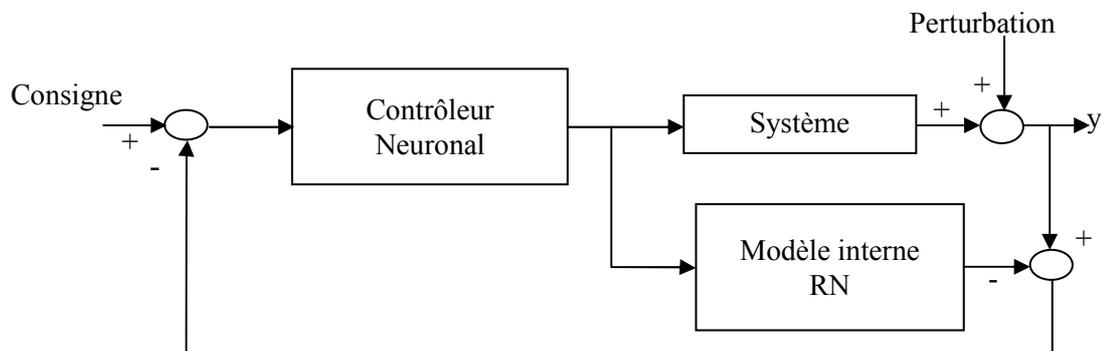
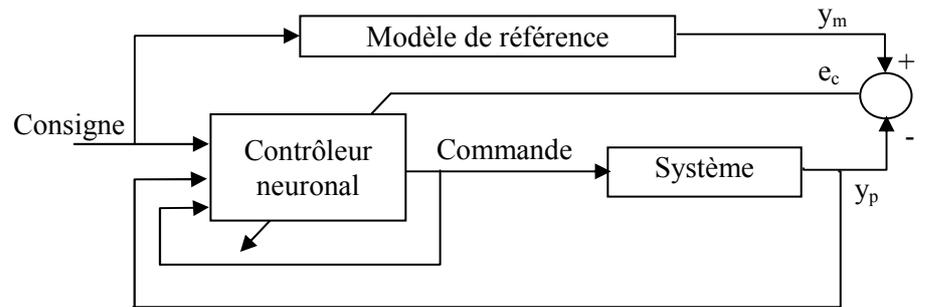


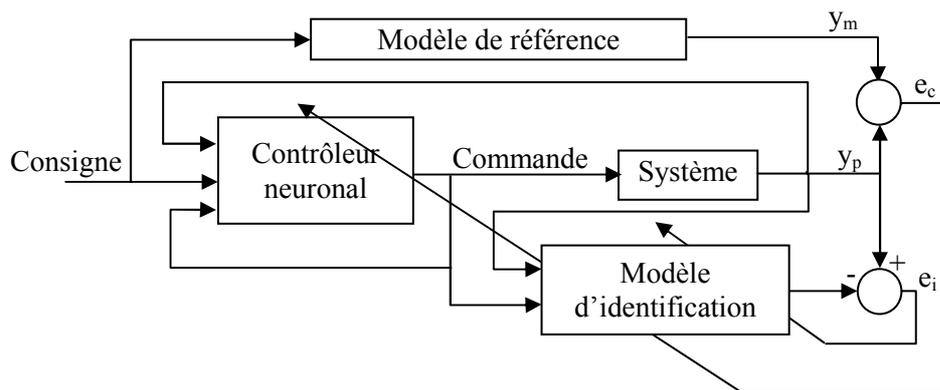
Figure 2.11 : Commande à modèle interne

#### 2.4.1.4. Commande adaptative à modèle de référence

Dans cette stratégie de commande, un modèle de référence est utilisé pour spécifier le comportement désiré du système à commander. Le contrôleur, quant à lui, est utilisé pour forcer la sortie du système à suivre celle du modèle de référence. Deux structures de commande ont été proposées, l'une directe et l'autre indirecte (figure 2.12).



a. méthode directe



b. méthode indirecte

Figure 2.12 : Commande adaptative à modèle de référence

Dans la structure directe (figure 2.12a), les paramètres du contrôleur sont directement adaptés pour réduire l'erreur en sortie du système. Une fois la phase d'apprentissage est achevée, l'action de contrôle est activée de telle sorte que la sortie du système suive celle du modèle de référence. Puisque le système est placé entre le réseau de neurones et l'erreur de sortie, on ne peut pas ajuster directement les poids du réseau contrôleur. Cette difficulté peut être surmontée en utilisant la structure indirecte.

Dans la structure indirecte (figure 2.12.b), un réseau de neurones est utilisé pour identifier la dynamique directe du système (modèle d'identification), les poids de ce réseau

peuvent être adaptés en ligne pour suivre en permanence le comportement dynamique du système. L'erreur utilisée pour adapter les poids du contrôleur neuronal est obtenue en rétropropageant l'erreur de commande (l'écart entre la sortie du modèle de référence et celle du système) dans le modèle d'identification. Cette méthode utilise deux réseaux de neurones, le premier, monté en parallèle du système, assure la fonction d'identification, le deuxième réseau assure la fonction de commande de forçant la sortie du système à suivre celle du modèle de référence.

#### 2.4.2. Commande indirecte

Dans la commande indirecte, la conception de la loi de commande est basée sur l'utilisation d'un modèle neuronal du processus. On peut citer parmi ces méthodes la commande prédictive à modèle neuronal: Cette stratégie de commande fera l'objet du chapitre suivant.

#### 2.5. Conclusion

Nous avons rappelé, dans ce chapitre, les éléments de base des réseaux de neurones, les différentes étapes de l'identification des systèmes en utilisant ces réseaux ainsi que quelques méthodes de commande à base des réseaux de neuronaux.

La propriété d'approximation universelle et d'adaptation que possèdent les réseaux de neurones leur permettent de résoudre avec une bonne précision le problème de modélisation et de commande des systèmes non linéaires.

La compréhension des notions exposées dans ce chapitre sont nécessaires pour mener à bien les développements qui seront envisagés dans le chapitre suivant.

## CHAPITRE 3

### COMMANDE PREDICTIVE A MODELE NEURONALE

#### 3.1. Introduction

Le principe de base d'une loi de commande prédictive comme il a été illustré dans le premier chapitre s'appuie sur le modèle du processus pour déterminer par anticipation la valeur optimale des entrées de commande à appliquer, au sens d'un critère de performance préalablement défini qui caractérise le but que l'on souhaite atteindre (régulation, poursuite, ou tout autre critère de qualité).

Le modèle du système joue un rôle central dans la commande. Le modèle choisi doit être capable de rendre compte de la dynamique du processus pour prédire précisément les sorties futures et il doit aussi être simple à implanter. Tous les algorithmes de commande prédictive ne diffèrent entre eux que par le modèle utilisé pour représenter le procédé et par la fonction de coût à minimiser.

La commande prédictive neuronal utilise un réseau de neurones pour la prédiction des sorties futures. Les réseaux de neurones permettent de modéliser n'importe quel système non linéaire avec une bonne précision. Cependant, leur utilisation dans les algorithmes de commande prédictive présente plusieurs difficultés. Dans ce chapitre, nous présentons le principe général de la commande prédictive à modèle neuronal et les problèmes liés à ce type de commande. Ensuite, nous exposons quelques solutions présentées qui ont été proposées pour contourner les difficultés rencontrées, et nous présentons l'approche développée.

#### 3.2. Commande prédictive à modèle neuronal

##### 3.2.1. Principe général

La commande prédictive à base d'un modèle neuronale est un cas particulier de la famille de la commande par modèle, sa particularité réside dans la nature du prédicteur

utilisé. En fait, cette stratégie de commande utilise un réseau de neurone comme modèle pour prédire les sorties futures du système (figure 3.1). Ensuite, les incréments futurs de la commande sont calculés en minimisant une fonction de coût [16] [9].

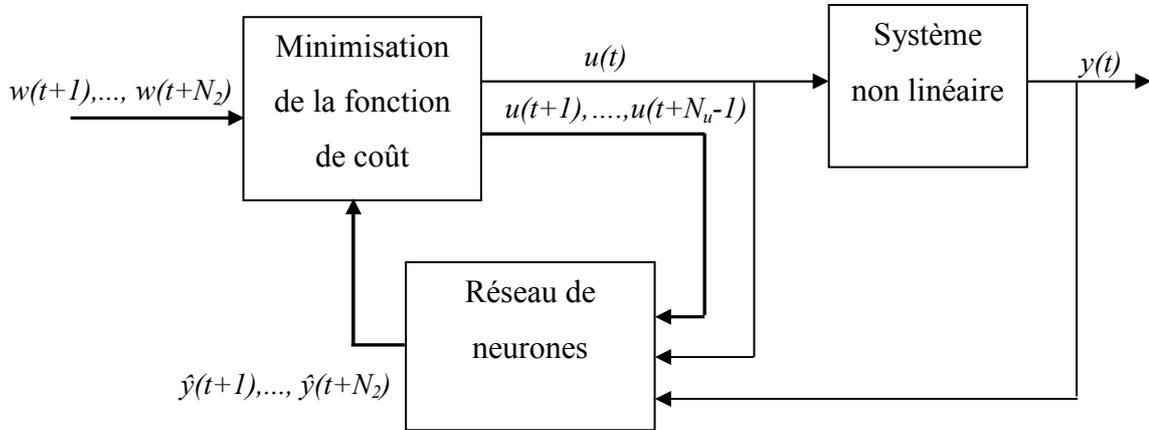


Figure 3.1 : Schéma de principe de la commande prédictive à modèle neuronal

La prédiction  $j$  pas en avant de la sortie du système, en utilisant un réseau de neurones est illustrée par la figure suivante.

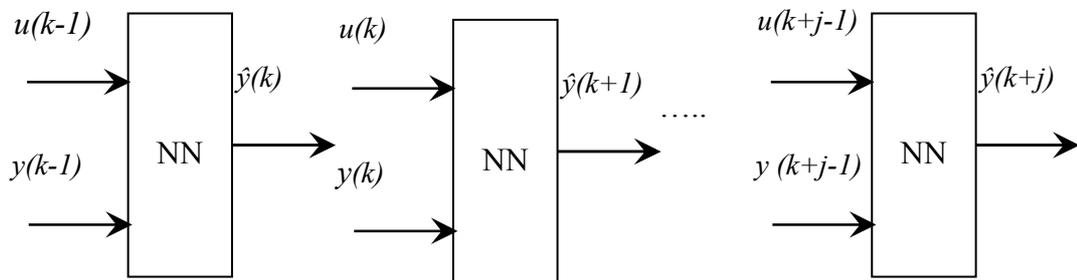


Figure 3.2 : Prédicteur neuronal

La loi de commande est obtenue en minimisant une fonction de coût. Cette dernière est en général, la somme de l'erreur quadratique entre les prédictions des sorties du système et la trajectoire de référence, et de l'énergie du signal de commande.

A ce niveau, on se trouve face à un problème de minimisation non linéaire et non convexe qui est difficile à résoudre. Plusieurs approches basées sur les méthodes de minimisation numérique ont été proposées pour résoudre ce problème. Ces méthodes itératives largement expliqués dans [9], demande un temps de calcul important. Leur utilisation dans le cas des systèmes à dynamique rapide n'est pas donc, recommandée.

Les autres approches, visant à obtenir une forme convexe à la fonction de coût, utilisent un modèle de prédiction de structure particulière [17][18]. Nous en citons plus particulièrement, l'approche basée sur l'idée de décomposition de la réponse totale du système en réponse libre et réponse forcée [19][20], et l'approche utilisant le principe de la linéarisation instantanée du modèle de prédiction [9][12][21].

Dans les travaux de F. Camacho et al [17][18], la réponse libre (réponse du système lorsque l'entrée de commande est maintenue constante durant les horizons de commande et de prédiction) est estimée à partir du modèle non linéaire du système, et la réponse forcée (la réponse due aux incréments futurs de la commande) est calculée à partir d'un modèle linéaire incrémental du système. Le modèle de prédiction est obtenu en combinant les deux réponses. Cette technique, permet de donner de meilleures performances par rapport aux méthodes utilisant un modèle de prédiction obtenu par linéarisation autour d'un point de fonctionnement. Cependant, il a été montré qu'elle est sensible aux perturbations et au changement du point de fonctionnement.

Dans notre travail, nous avons adopté cette technique de décomposition et à la place d'utiliser le modèle linéaire incrémental du système pour calculer la réponse forcée, nous avons préféré d'utiliser un modèle linéaire issu de la linéarisation instantanée du modèle du système.

### 3.2.2. Approche par décomposition et linéarisation instantanée

Le but de décomposer la réponse totale du système en réponse libre et réponse forcée, est d'appliquer la méthodologie de la commande prédictive généralisée linéaire. En effet, cette décomposition permet de préserver les non linéarités du système dans la réponse libre d'une part, et de calculer analytiquement la loi de commande, d'autre part. Le principe de cette technique, est illustré par le schéma bloc de la figure 3.3.

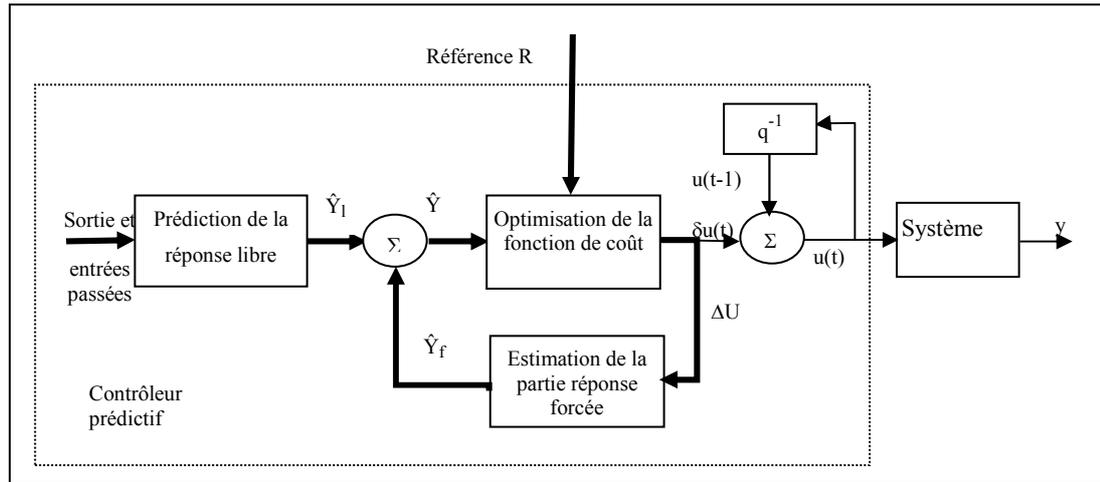


Figure 3.3 : Structure de commande

### 3.2.2.1. Fonction de coût

La loi de commande est obtenue par minimisation d'une fonction de coût quadratique portant sur les erreurs futures et incréments de commande de la forme suivante :

$$J = \sum_{j=N_1}^{N_2} (r(t+j) - \hat{y}(t+j))^T Q (r(t+j) - \hat{y}(t+j)) + \sum_{j=1}^{N_u} \delta u(t+j-1)^T S \delta u(t+j-1) \quad (3.1)$$

où  $N_1$  est l'horizon minimal de prédiction,  $N_2$  l'horizon maximal de prédiction,  $N_u$  l'horizon de commande,  $r(t)$  la trajectoire de référence,  $\hat{y}(t+j)$  les sorties prédites,  $\delta u(t) = u(t) - u(t-1)$  l'incrément de commande et  $Q$  et  $S$  sont deux matrices définies positive et semi positive respectivement.

### 3.2.2.2. Modèle de prédiction

Le système à commander est représenté par le modèle neuronal NARX suivant :

$$\begin{aligned} y(t) &= NN[Y(t-1), U(t-1)] \\ Y(t-1) &= [y(t-1) \dots y(t-m)] \\ U(t-1) &= [u(t-d) \dots u(t-d-n+1)] \end{aligned} \quad (3.2)$$

où  $d$  est le temps mort du système et  $NN[]$  est le modèle neuronal du système.

En utilisant l'idée de décomposition, le modèle de prédiction est donnée par:

$$\hat{y}(t+k/t) = \hat{y}_f(t+k/t) + \hat{y}_l(t+k/t) \quad (3.3)$$

où :  $\hat{y}_f(t + k/t)$ : sont les valeurs futures de la réponse forcée dues aux incréments futurs de la commande, et  $\hat{y}_l(t + k/t)$ : sont les valeurs futures de la réponse libre dues aux conditions initiales disponible à l'instant  $t$ .

Par la suite, nous utilisons pour modéliser le système un réseau de neurones MLP à une seule couche cachée et une seule sortie linéaire. Sa sortie, à l'instant  $t + j$ , est donnée par les équations suivantes :

$$y(t + j) = \sum_{i=1}^{n_h} W_i h_i(t + j) + W_0 \quad j = N_1, \dots, N_2 \quad (3.4)$$

$$h_i(t + j) = f \left( \sum_{l=1}^{n_e} w_{il} \varphi_l(t + j) + w_{i0} \right) \quad (3.5)$$

où  $f(\dots)$  est la fonction tangente hyperbolique et  $\varphi(t)$  le vecteur d'entrée du réseau de neurones,  $n_e, n_h$  sont le nombre d'entrées et le nombre de neurones de la couche cachée, respectivement.  $\varphi(t)$  il est donné par :

$$\varphi(t) = [Y(t - 1), U(t - 1)] \quad (3.6)$$

- **Calcul de  $\hat{y}_f(t + k/t)$ :**

Cette partie est calculée en utilisant un modèle linéaire pour le système. Ce modèle, de la forme donnée par l'équation (3.7), est obtenu en linéarisant à chaque période d'échantillonnage le modèle neuronal du système autour du point de fonctionnement actuel.

$$y(t) = (1 - A(q^{-1}))y(t) + q^{-d}B(q^{-1})u(t) + \zeta(\tau) \quad (3.7)$$

où : les polynômes :

$$\begin{cases} A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a} \\ B(q^{-1}) = 1 + b_1 q^{-1} + \dots + b_{m_b} q^{-m_b} \end{cases}$$

à l'instant  $t = \tau$ , sont obtenus de la manière suivante :

$$\begin{cases} a_i = \left. \frac{\partial y(t)}{\partial y(t-i)} \right|_{\varphi(t)=\varphi(\tau)} \\ b_i = \left. \frac{\partial y(t)}{\partial u(t-d-i)} \right|_{\varphi(t)=\varphi(\tau)} \end{cases} \quad (3.8)$$

et le terme du biais  $\zeta(\tau)$  est donné par :

$$\zeta(\tau) = y(\tau) + a_1 y(\tau - 1) + \dots + a_{n_a} y(\tau - n_a) - b_0 u(\tau - d) - \dots - b_{m_b} u(\tau - d - m_b) \quad (3.9)$$

Le modèle donné par l'équation (3.7) est considéré comme étant un modèle linéaire affecté par une perturbation constante dépendante du point de fonctionnement actuel.

Dans le cas du réseau de neurones utilisé, la dérivée de la sortie par rapport aux composantes  $\varphi_i(t)$  du vecteur d'entrée est donnée par:

$$\frac{\partial y(t)}{\partial \varphi_i(t)} = \sum_{j=1}^{n_h} W_j w_{jk} \left[ 1 - f \left( \sum_{k=1}^{n_e} w_{jk} \varphi_k(t) + w_{j0} \right)^2 + W_0 \right] \quad (3.10)$$

Le modèle linéaire obtenu sert dans le calcul de la matrice  $G(t)$  des coefficients de la réponse indicielle. Ces coefficients sont déterminés en résolvant les deux équations diophantiennes, de la manière indiquée dans le premier chapitre.

- **Calcul de  $\hat{y}_l(t + k/t)$**

Le terme  $\hat{y}_l(t + k/t)$  représente, dans notre cas, la réponse du modèle neuronal aux conditions initiales disponibles à l'instant  $t$ , en gardant l'entrée de commande constante et égale à sa dernière valeur  $u(t - 1)$  pendant tout l'intervalle de prédiction.

Les sorties prédites  $\hat{y}_l(t + k/t)$  sont générées d'une manière récursive en utilisant le modèle neuronal du système :

$$\begin{aligned} \hat{y}_l(t + k/t) &= NN[Y(t + k - 1), U(t + k - 1)] & k = N_1, \dots, N_2 \\ Y(t + k - 1) &= [y(t + k - 1), \dots, y(t + j), \dots, y(t + k - m)] \\ U(t + k - 1) &= [u(t + k - d), \dots, u(t + i), \dots, u(t + k - d - n + 1)] \end{aligned} \quad (3.11)$$

$$y(t + j) = \begin{cases} y(t + j) & \text{si } j < 0 \\ \hat{y}_l(t + j) + \hat{y}_f(t + j) & \text{si } j \geq 0 \end{cases} \quad j = k - m, \dots, k - 1$$

$$u(t + i) = \begin{cases} u(t + i) & \text{si } i < 0 \\ u(t - 1) & \text{si } i \geq 0 \end{cases} \quad i = k - d - n + 1, \dots, k - d$$

### 3.2.2.3. Minimisation de la fonction de coût

En utilisant la représentation matricielle, il est possible d'écrire l'expression (3.3) sous la forme suivante :

$$\hat{Y}(t) = G(t)\Delta U(t) + \hat{Y}_l(t) \quad (3.12)$$

où :

$$\hat{Y}(t) = \begin{bmatrix} \hat{y}(t + N_1/t) \\ \hat{y}(t + N_1 + 1/t) \\ \vdots \\ \hat{y}(t + N_2/t) \end{bmatrix} \quad G(t) = \begin{bmatrix} g_{N_1-1} & g_{N_1-2} & \dots & g_0 & 0 & \dots & \dots & \dots & 0 \\ g_{N_1} & g_{N_1-1} & \dots & g_0 & 0 & \dots & \dots & \dots & 0 \\ & & & \vdots & & & & & \\ g_{N_2-1} & g_{N_2-2} & \dots & \dots & \dots & \dots & \dots & \dots & g_{N_2-N_u} \end{bmatrix}$$

$$\Delta U(t) = \begin{bmatrix} \delta u(t/t) \\ \delta u(t + 1/t) \\ \vdots \\ \delta u(t + N_u - 1/t) \end{bmatrix} \quad \hat{Y}_l = \begin{bmatrix} \hat{y}_l(t + N_1/t) \\ \hat{y}_l(t + N_1 + 1/t) \\ \vdots \\ \hat{y}_l(t + N_2/t) \end{bmatrix} \quad R(t) = \begin{bmatrix} r(t + N_1) \\ r(t + N_1 + 1) \\ \vdots \\ r(t + N_2) \end{bmatrix}$$

La fonction de coût devient:

$$\begin{aligned} J &= (R(t) - \hat{Y}(t))^T Q (R(t) - \hat{Y}(t)) + \Delta U(t)^T S \Delta U(t) \\ &= (R(t) - \hat{Y}_l(t) - G(t)\Delta U(t))^T Q (R(t) - \hat{Y}_l(t) - G(t)\Delta U(t)) + \Delta U(t)^T S \Delta U(t) \end{aligned} \quad (3.13)$$

En dérivant  $J$  par rapport à  $\Delta U(t)$ , nous obtenons :

$$\frac{\partial J}{\partial \Delta U(t)} = 2G^T(t)Q (\hat{Y}_l(t) - R(t)) + 2(G^T(t)QG(t) + S)\Delta U(t) \quad (3.14)$$

Puisque la séquence optimale des incréments de commande correspond à  $\frac{\partial J}{\partial \Delta U(t)} = 0$ , nous obtenons alors :

$$\Delta U(t) = (G^T(t)QG(t) + S)^{-1}G^T(t)Q (R(t) - \hat{Y}_l(t)) \quad (3.15)$$

La valeur optimale de la commande à l'instant  $t$  est donnée par :

$$u(t) = u(t - 1) + \delta u(t) \quad (3.16)$$

où  $\delta u(t)$  est le premier élément de  $\Delta U(t)$

### 3.2.2.4. Algorithme

Les étapes principales de l'algorithme de calcul de la commande sont résumées par les points suivants :

- 1- Spécifier la trajectoire de référence et choisir les paramètres de réglage ;
- 2- Linéarisation instantanée du modèle du système à l'instant d'échantillonnage  $t$  (équation (3.7)) ;
- 3- Calculer la matrice  $G(t)$  par la résolution des équations diophantiennes;
- 4- En utilisant le modèle du système, calculer les valeurs futures  $\hat{y}_l(t + k/t)$  (équation (3.11)) ;
- 5- Calculer le vecteur des incréments de commande  $\Delta U(t)$  (équation (3.15)) ;
- 6- La valeur de la commande  $u(t)$  à l'instant d'échantillonnage  $t$  est donnée par l'équation (3.16) ;
- 7- Aller à l'étape 2 pour calculer une nouvelle valeur de  $u(t)$  ;

### 3.3. Simulation

Nous considérons la commande du système non linéaire décrit par l'équation différentielle suivante :

$$\ddot{y}(t) + \dot{y}(t) + y(t) + y^3(t) = u(t) \quad (3.17)$$

Le résultat d'identification de ce système a été donné dans le chapitre précédent. Nous rappelons ici le modèle utilisé :

$$\hat{y}(t) = NN(u(t-1), u(t-2), y(t-1), y(t-2)) \quad (3.18)$$

où  $NN$  est un réseau de neurones MLP possédant 10 neurones sur sa couche cachée.

#### 3.3.1. Décomposition et utilisation d'un modèle linéaire fixe

Dans ce cas, nous utilisons un modèle linéaire fixe pour générer la réponse forcée. Ce modèle est obtenu par identification du système. Il est donné par :

$$H(z) = \frac{0,0092z^{-1} + 0,00044z^{-2}}{1 - 1,898z^{-1} + 0,90951z^{-2}} \quad (3.19)$$

La réponse libre est estimée à partir du modèle neuronal du système.

Les figures 3.4 et 3.5 présentent les performances obtenues en choisissant pour les paramètres de réglage les valeurs suivantes :

$$T_e = 0.2 \text{ s}, N_1 = 1, N_2 = 8, N_u = 1, Q = I_{N_2 \times N_2} \text{ et } S = 10^{-3}$$

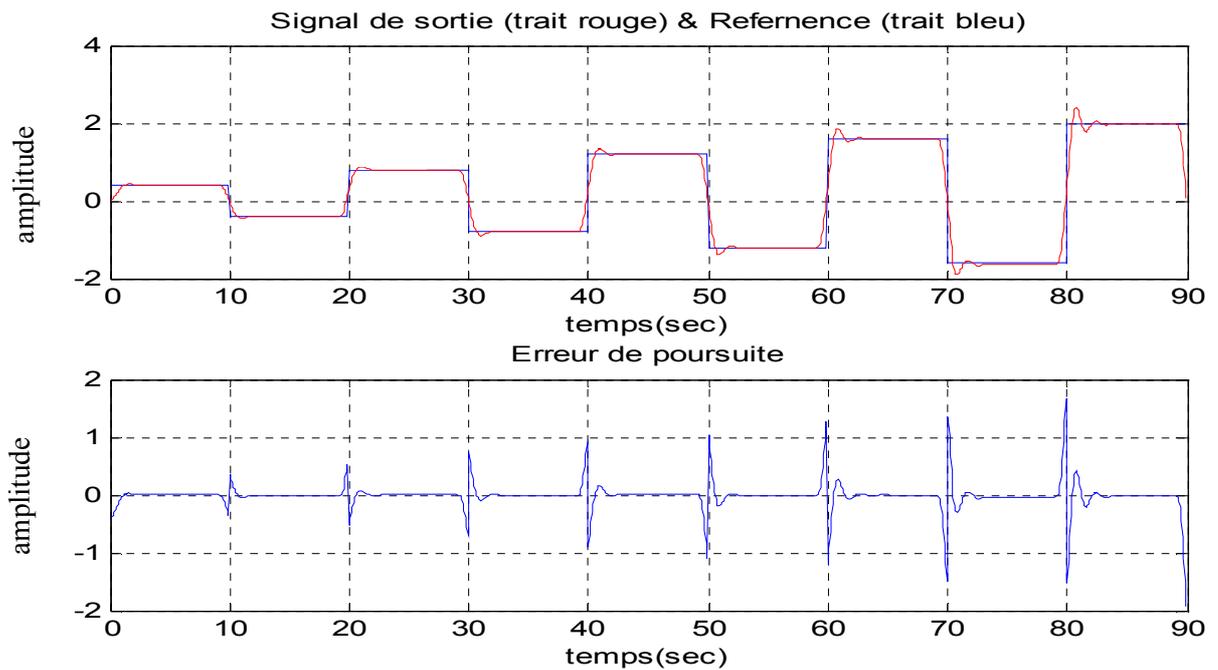


Figure 3.4: Performances obtenues du régulateur

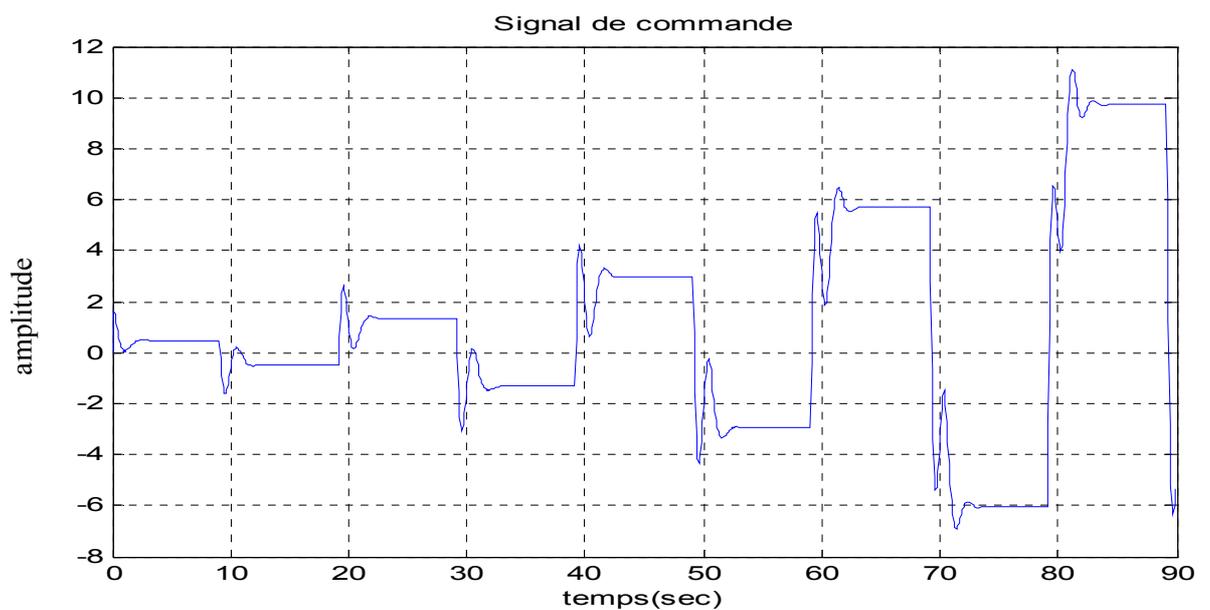


Figure 3.5: Effort de commande

Nous obtenons une bonne poursuite de la trajectoire de référence, mais on remarque que lorsque l'amplitude du signal de référence augmente, l'erreur de poursuite de la trajectoire de référence augmente aussi.

Dans le but d'améliorer le résultat obtenu par l'approche précédente, nous allons cette fois-ci procéder à la linéarisation instantanée du modèle de prédiction.

### 3.3.2. Linéarisation instantanée

Dans ce cas, la loi de commande est déterminée en utilisant la méthodologie de la GPC linéaire. Nous utilisons un modèle linéaire de la forme donnée par l'équation (3.7). Les coefficients de ce modèle sont calculés à chaque période d'échantillonnage par linéarisation du modèle neuronal du système autour du point de fonctionnement actuel. Les performances obtenues du régulateur pour les mêmes valeurs des paramètres de réglage utilisées dans l'exemple précédent sont données par les figures 3.6 et 3.7.

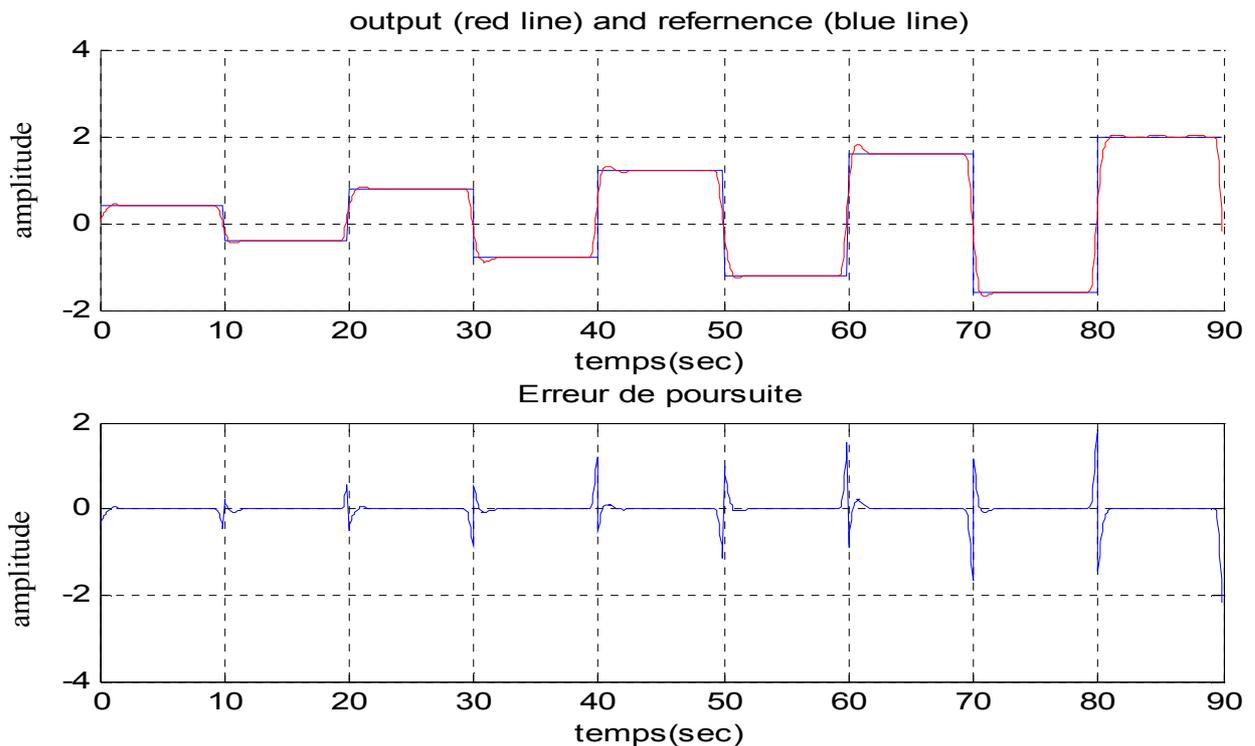


Figure 3.6: Performances obtenues du régulateur

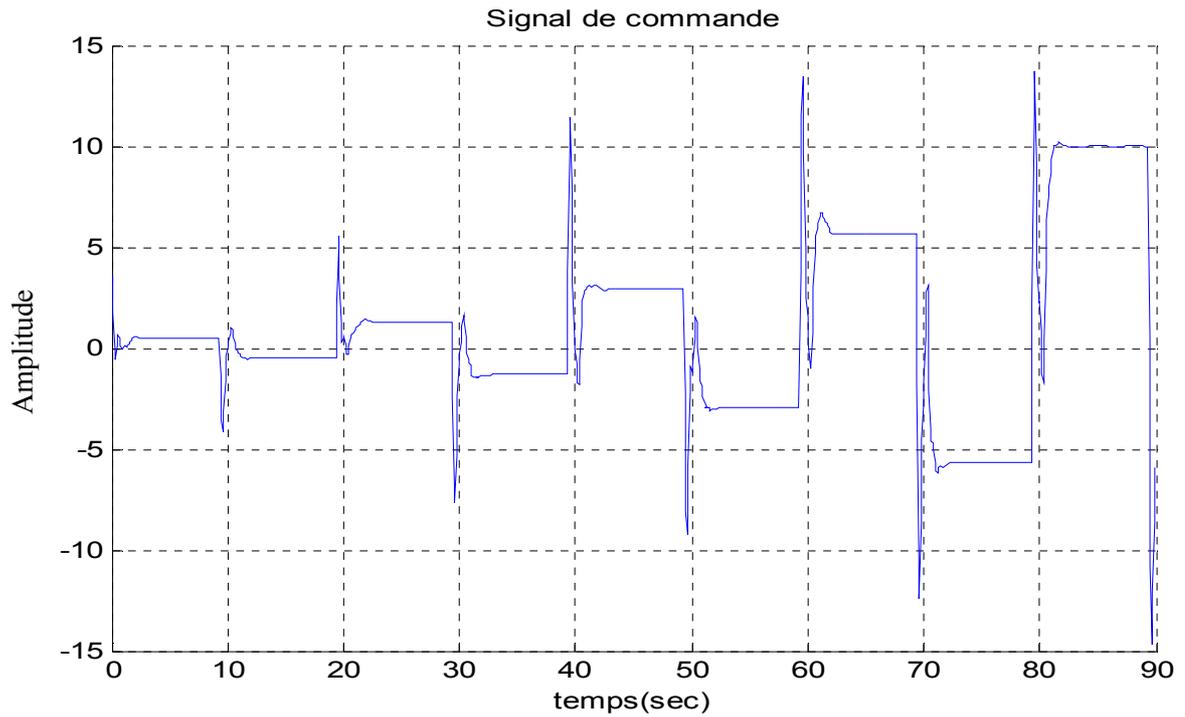


Figure 3.7: Effort de commande

On remarque que le signal de la sortie suit mieux la consigne malgré que l'amplitude de celle-ci augmente.

### 3.3.3. Décomposition et linéarisation instantanée

La matrice  $G(t)$  des coefficients de la réponse indicielle est cette fois-ci, calculée à chaque période d'échantillonnage en utilisant le modèle linéaire donné par l'équation (3.7). Nous rappelons que les coefficients de ce modèle sont évalués à chaque période d'échantillonnage, par linéarisation du modèle neuronal du système, et que la réponse libre est estimée à partir de ce dernier.

Les performances de commande obtenues pour les mêmes valeurs des paramètres de réglage utilisées précédemment sont données par les figures (3.8) et (3.9).

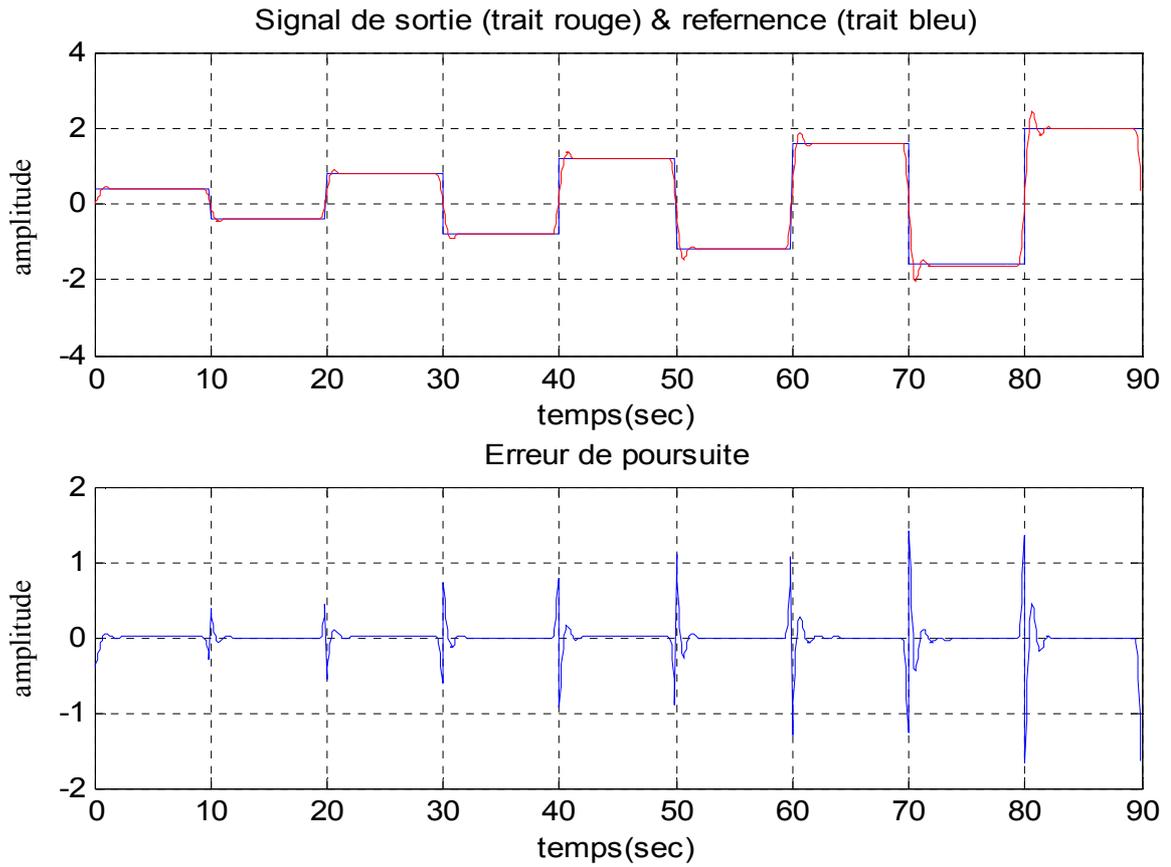


Figure 3.8 : Performances obtenues du régulateur

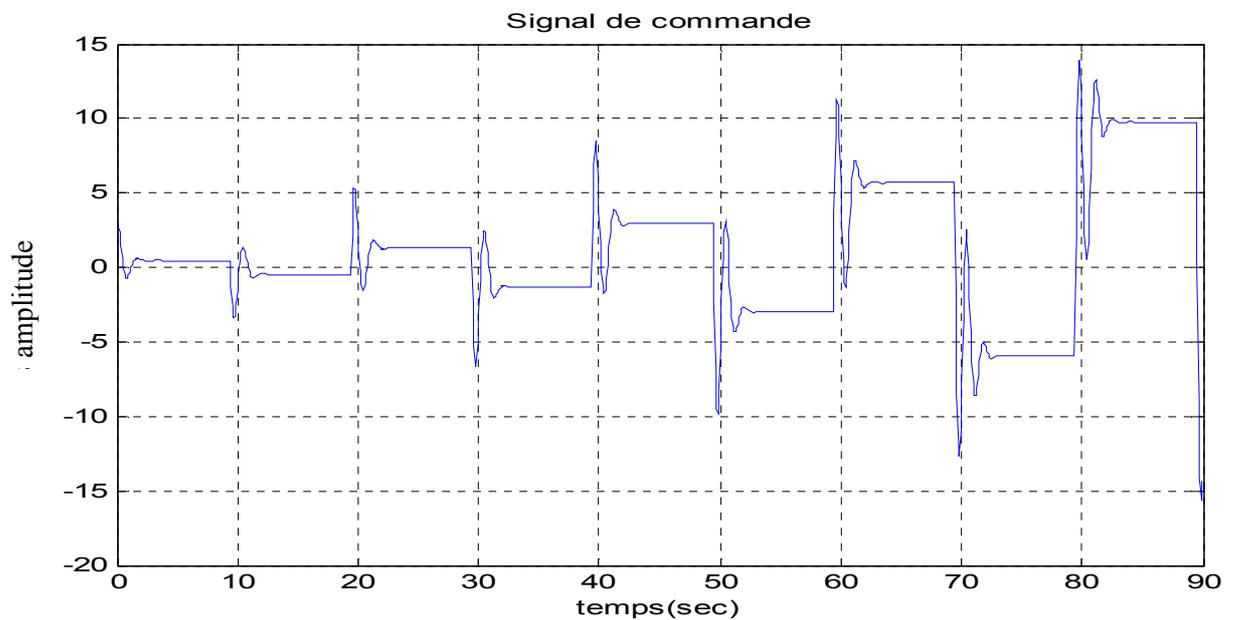


Figure 3.9 : Signal de commande

Nous obtenons une bonne poursuite de la trajectoire de référence avec une commande plus ou moins lisse.

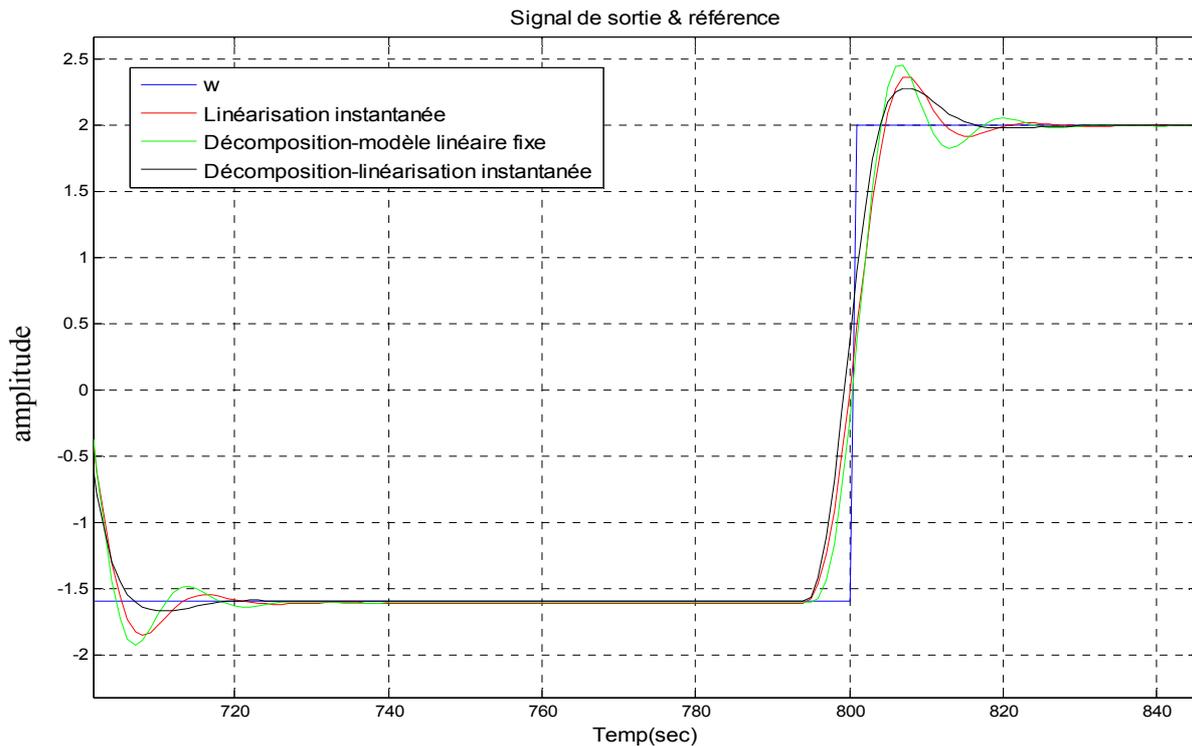


Figure 3.10 : Performances obtenues dans les trois cas de commande

Dans le but de comparer les trois méthodes, la figure 3.10 illustre la vue agrandie d'une partie des sorties du système obtenues dans les trois cas, avec les mêmes valeurs des paramètres de réglage données précédemment. On remarque qu'il y a une amélioration de la poursuite de la trajectoire de référence dans la méthode de la composition avec la linéarisation instantané, où on a obtenu la plus petite erreur quadratique moyenne, elle égale à : 0.0414 par rapport à la méthode de la linéarisation instantané, l'erreur de poursuite été à 0.0540 et 0.0707 pour la méthode de décomposition avec modèle linéaire fixe.

### 3.4. Conclusion

Nous avons considéré dans ce chapitre, la commande prédictive non linéaire à base d'un modèle neuronal. Nous avons exposé les problèmes liés à l'utilisation d'un modèle neuronal et les solutions proposées dans la littérature. En particulier, nous avons considéré la technique basée sur la décomposition de la réponse total en réponse libre et réponse forcée et celle basée sur la linéarisation instantanée du modèle du système. Ensuite, nous avons proposé de combiner l'idée de décomposition et la linéarisation instantanée du modèle du système. Les performances des trois techniques ont été étudiées en considérant un exemple de simulation. Dans, le chapitre suivant nous allons envisager l'application des ces techniques à la commande d'un réacteur continu parfaitement agité.

# CHAPITRE 4

## APPLICATION A LA COMMANDE D'UN REACTEUR CHIMIQUE

### 4.1. Introduction

Ce chapitre envisage l'application des structures de commande prédictive à modèle neuronal, considérées lors des chapitres précédents, à un procédé chimique souvent utilisé, comme exemple pour évaluer les performances des nouveaux algorithmes de commande. Ce procédé est un réacteur continu parfaitement mélangé (CSTR : Continus Stirred Tank Reactor). L'approche utilisée se sert de l'idée de décomposition de la réponse totale qui permet de préserver les non linéarités du système dans la réponse libre d'une part, et de calculer analytiquement la loi de commande, d'autre part. Dans le but d'améliorer les performances de cette technique, nous avons utilisé un modèle linéaire, issu de la linéarisation instantanée du modèle neuronal du système, pour calculer la réponse forcée.

Pour mettre en évidence l'approche développée, nous considérons aussi la commande du réacteur en utilisant la GPC linéaire, la technique basée sur la décomposition avec un modèle linéaire à paramètres fixes et celle basée sur la linéarisation instantanée du modèle neuronal du système.

### 4.2. Modélisation du réacteur continu parfaitement mélangé

#### 4.2.1. Définition et caractéristiques des réacteurs chimiques

On appelle réacteur chimique tout appareil permettant de réaliser une réaction chimique qui permet de transformer des espèces moléculaires en d'autres espèces moléculaires dans le but de produire une ou plusieurs espèces chimiques, ou d'éliminer d'un mélange un ou plusieurs composés.

Généralement un réacteur présente des caractéristiques de fonctionnement qui conditionnent les installations placées en amont (préparation des charges de réactifs, choix

de la température et la pression) et les dispositifs placés en aval (pour la séparation des produits), c'est un élément intermédiaire qui participe énormément à l'amélioration du coût d'investissement et de la consommation en matière d'énergie. En ce sens il est véritablement « le cœur du procédé » qui requiert toute l'attention de l'ingénieur.

Un réacteur chimique exige des caractéristiques très diverses. Toutefois tous les réacteurs doivent répondre généralement aux conditions principales suivantes [22] :

- assurer un rendement élevé.
- fournir un taux de transformation le plus grand possible.
- consommer le moins d'énergie pour le transport et l'agitation des réactifs.
- être suffisamment simple et pas cher.
- utiliser au mieux la chaleur des réacteurs exothermiques.
- avoir un fonctionnement sûr et permettre un contrôle automatique du procédé.

#### 4.2.2. Classification des réacteurs chimiques

On distingue plusieurs types de réacteurs qui dépendent essentiellement des principaux critères suivants [22] :

- La circulation du mélange réactionnel.
- L'évolution dans le temps.
- L'état thermique.
- Le degré de mélange des substances en réaction à l'intérieur du réacteur.

Le tableau 4.1 résume la classification et les types de réacteurs les plus utilisés

Critère	Type de réacteur
Circulation du mélange réactionnel	Réacteur fermé Réacteur semi- fermé Réacteur ouvert
Évolution dans le temps	En régime transitoire En régime permanent
État thermique	Réacteur adiabatique. Réacteur isothermique.
Degré de mélange des substances en réaction	Réacteur parfaitement agité Réacteur à écoulement piston

Tableau 4.1 : Classification des réacteurs chimiques

Le réacteur de notre application est un réacteur continu parfaitement agité connu sous la dénomination anglo-saxonne : Continuous Stirred Tank Reactor (CSTR). Une description assez succincte sera présentée pour faciliter sa modélisation.

#### 4.2.3. Caractéristique du réacteur C.S.T.R

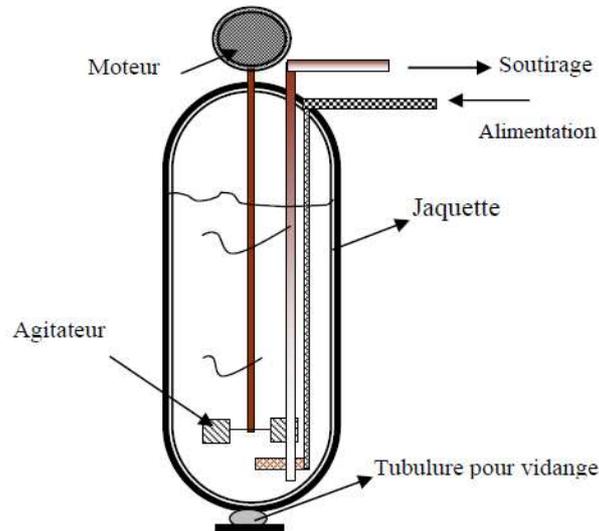


Figure 4.1 : Schéma du réacteur chimique parfaitement mélangé

Ce type de réacteur étant continu, ouvert et parfaitement agité, il appartient donc à la famille des réacteurs homogènes. L'ensemble du volume des réactifs à l'intérieur du réacteur, possède une concentration identique avec une température uniforme. Comme le montre la figure 4.1, il est alimenté par un réactif frais, et le produit de la réaction est soutiré de manière continue.

Ce réacteur présente une réaction exothermique, car la conversion du réactif de A vers B s'accompagne d'un dégagement de chaleur. Celle-ci augmente progressivement jusqu'à ce que la température d'activation relative à la réaction chimique désirée ( $A \rightarrow B$ ) soit atteinte. Mais dès que la température dépasse un certain seuil  $T_a$ , le réactif résultant B désiré va se décomposer pour donner un troisième réactif C indésirable. Une autre conséquence plus grave est qu'elle peut engendrer l'explosion de l'installation dû à l'emballement thermique.

D'après la figure 4.2, on peut remarquer que la température et le taux de conversion des réactifs sont uniformément croissants pour un réacteur CSTR. De même, la vitesse de conversion passe par un maximum, puis elle tend vers zéro tel que le taux de conversion

s'approche de 100%. Delà, on voit l'importance du point maximum de la vitesse de conversion sur le rendement de la réaction, et la nécessité de le prendre en considération, et cela en agissant sur la température d'une manière très prudente pour que le système fonctionne dans les conditions voulues. Un refroidissement performant doit être intégré au bon fonctionnement de l'installation.

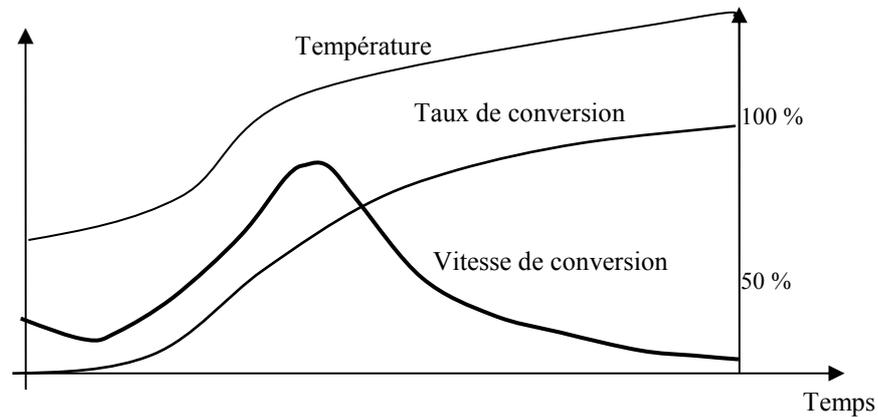


Figure 4.2: Variation des paramètres de réactions dans un CSTR

#### 4.2.4. Bilan de matière et d'énergie dans les réacteurs chimiques

En général, un bilan de matière ou d'énergie doit être exprimé dans une zone telle que le volume réactionnel (siège d'une transformation chimique), effectue un échange de matière et d'énergie avec le milieu extérieur.

**Bilan de la matière :** Pour établir un bilan de la matière dans un réacteur, il faut qu'il existe une variation des paramètres opératoires. Dans le cas d'une cuve parfaitement agitée la conservation de la matière exige qu'à chaque instant :

$$\left( \text{Flux de } A_i \text{ entrant} \right) + \left( \text{débit de production} \right) = \left( \text{flux de } A_i \text{ sortant} \right) + \left( \text{débit d'accumulation de } A_i \right)$$

**Bilan énergétique :** Dans le cas d'un réacteur ouvert en région quelconque, et en appliquant le premier principe de la thermodynamique (conservation d'énergie) la forme générale se réduit à une balance d'enthalpie de la forme :

$$\left( \text{Flux d'énergie interne entrant} \right) + \left( \text{Chaleur ajoutée au système par réaction} \right) = \left( \text{Flux d'énergie interne sortant} \right) + \left( \text{énergie accumulée dans le réacteur} \right)$$

Un réacteur est dit instable lorsqu'une légère augmentation de la température produira un accroissement de la vitesse de production de la chaleur plus importante que la vitesse d'élimination de la chaleur.

#### 4.2.5. Modélisation

La modélisation et la simulation des procédés sont devenues dans de nombreux secteurs industriels, un moyen efficace et souvent indispensable dans la représentation et l'analyse d'un processus. Ceci dans le but de concevoir un équipement ou une installation devant répondre aux exigences de sécurité et de productivité optimales. Parmi les avantages de la simulation, on peut citer, la minimisation de l'effort expérimental souvent coûteux et parfois inadéquat. D'autre part, elle permet de prédire le mauvais fonctionnement et les risques les plus divers. On peut résumer la stratégie de simulation comme le schématise la figure 4.3 en plusieurs étapes importantes.

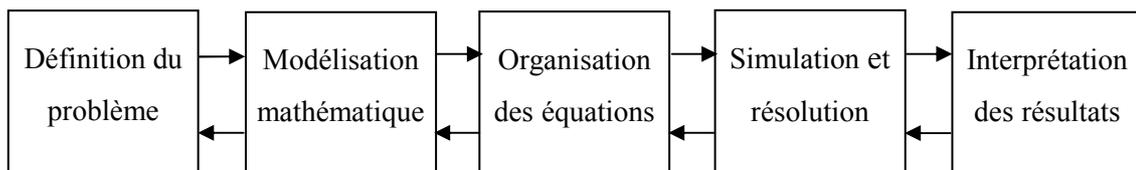


Figure 4.3 : Stratégie de simulation d'un processus

##### 4.2.5.1. Modèle Mathématique

Comme la réaction est du premier ordre et irréversible  $A \longrightarrow B$ , et en prenant le cas où la conversion d'un réactif A est unique en milieu indilatable et si  $x$  est son taux de conversion, et en utilisant la loi Arrhenius, la vitesse de conversion  $r$  de la réaction sera [22]:

$$r = K C_a \quad (4.1)$$

comme

$$K = K_0 \exp^{-E/RT} \quad \text{et} \quad C_a = C_{a0}(1 - x)$$

par conséquent

$$r = K_0 \exp^{-E/RT} C_a \quad (4.2)$$

où

$C_a$  : concentration du réactif.

$K_0$  : facteur de fréquence constant.

$C_{a0}$  : concentration initiale du réactif à l'instant  $t = 0$ .

$K$  : constante de vitesse.

$T$  : température en °Kelvin.

$E$  : énergie d'activation.

$R$  : constante des gaz parfaits.

a) En faisant intervenir le bilan de la matière qui se traduit par la conservation du débit molaire c'est à dire,

$$F_{entrant} + F_{produit} = F_{sortant} + F_{accumulé} \quad (4.3)$$

alors en remplaçant chaque débit molaire par son expression dans (4.3) on obtient :

$$Q C_{a0}(t) + \gamma rV = Q C_a(t) + d\eta_a/dt \quad (4.4)$$

$\gamma$  : coefficient stœchiométrique égale à -1

$\eta_a$  : molarité.

$Q$  : débit volumique du réactif.

$V$  : volume du réactif.

$F$  : débit molaire en (mole/min).

puisque  $\eta_a = C_a V$  et en utilisant 1.1, 1.2 et 1.4 alors

$$Q C_{a0}(t) - K C_a(t) V = Q C_a(t) + \frac{dC_a(t)V}{dt} \quad (4.5)$$

Comme le volume de la cuve est constant, l'équation (4.5) devient:

$$Q (C_{a0} - C_a(t)) - K_0 (\exp^{-E/RT}) C_a(t) V = V \frac{dC_a(t)}{dt} \quad (4.6)$$

d'où

$$\frac{dC_a(t)}{dt} = \frac{Q}{V} (C_{a0} - C_a(t) - K_0 C_a(t) (\exp^{-E/RT(t)}) \quad (4.7)$$

b) Le bilan d'énergie se traduit maintenant par la conservation de l'énergie au niveau du réacteur, l'équation est de la forme:

$$F_e \Delta H_e + \gamma r V \Delta H_r = F_s \Delta H_s + q + \frac{dU}{dt} \quad (4.8)$$

avec :

$F_e, F_s$  : débits volumiques des réactifs entrant et sortant égales à  $Q$

$\Delta H_e$  : flux d'enthalpie entrant

$\gamma$  : coefficient stœchiométrique égale à -1

$\Delta H_r$  : flux d'enthalpie de réaction

$\Delta H_s$  : flux d'enthalpie sortant

$V$  : Volume du réactif

$q$  : quantité d'énergie reçue par le réfrigérant (ou l'extérieur)

$\frac{dU}{dt}$  : accumulation d'énergie

en remplaçant chaque terme par son expression dans (4.8), on aura :

$$Q \rho C_p (T_0 - T_r) - K C_a(t) V \Delta H_r = Q \rho C_p (T_s(t) - T_r) + q + \frac{d(m C_p T(t))}{dt} \quad (4.9)$$

$$Q \rho C_p (T_0 - T_r) - C_a(t) \Delta H_r V K_0 \exp^{-\frac{E}{RT_s}} = Q \rho C_p (T_s(t) - T_r) + q + \rho V C_p \frac{dT(t)}{dt}$$

$$\frac{dT(t)}{dt} = \frac{Q}{V} (T_0 - T(t)) - \frac{\Delta H}{\rho C_p} C_a(t) K_0 (\exp^{-\frac{E}{RT_s(t)}}) - \frac{q}{\rho V C_p} \quad (4.10)$$

$Q$  : débit du réactif.

$\rho$  : masse volumique du réactif.

$C_p$  : chaleur spécifique du réactif.

$T_r = 0^\circ K$  : température de référence.

$T_0$  : température initiale du réactif.

$T_s(t) = T(t)$  : température de sortie du réactif.

c) En faisant maintenant intervenir le bilan énergétique au niveau de la jaquette et du milieu réactionnel, qui se traduit simplement par un échange de chaleur entre le réactif et le réfrigérant (figure 4.4) :

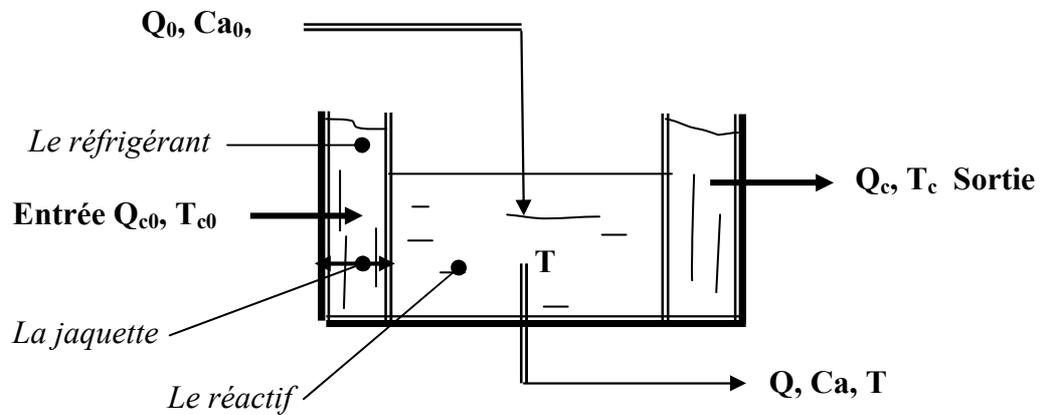


Figure 4.4 : Coupe du réacteur avec les différents échanges de chaleur

Puisque le réacteur est parfaitement agité alors les concentrations et les températures sont uniformément réparties dans le réacteur, et en supposant que les débits dans le réacteur sont constants alors  $Q_0 = Q$  et  $Q_{c0} = Q_c$

De même, l'équation de transfert de chaleur entre le réfrigérant et le milieu réactionnel est :

$$q = hA(T - T_c) \quad (4.11)$$

avec :

$h$  : coefficient de transfert de chaleur.

$A$  : surface d'échange.

$T$  : température du milieu réactionnel.

$T_c$  : température du réfrigérant.

Or, le bilan d'énergie au niveau de l'échangeur (réfrigérant et réactif) (figure 4.4), est:

$$F_e \Delta H_e + q = F_s \Delta H_s + dU/dt \quad (4.12)$$

avec :  $F_e$  et  $F_s$  : débits volumiques du réfrigérant entrant et sortant égales à  $Q_c$

et en remplaçant chaque terme dans l'expression (4.12), on obtient :

$$Q_c[\rho_c C_{pc} (T_{c0} - T_r)] + q = Q_c[\rho_c C_{pc} (T_c(t) - T_r)] + \rho_c V_c C_{pc} \frac{dT_c(t)}{dt} \quad (4.13)$$

avec :

$q$  : quantité d'énergie reçue par le réactif.

$Q_c$  : débit volumique du réfrigérant.

$T_c$  : température de sortie du réfrigérant.

$T_{c0}$  : température d'entrée du réfrigérant.

$V_c$  : volume du réfrigérant.

En utilisant l'équation (4.11) avec  $h_a = hA$ , l'équation (4.13) devient :

$$\frac{dT_c(t)}{dt} = \frac{Q_c}{V_c} (T_{c0} - T_c(t)) + \frac{h_a}{\rho_c V_c C_{pc}} (T(t) - T_c(t)) \quad (4.14)$$

Enfin le modèle mathématique du réacteur CSTR est donné par trois équations différentielles non linéaires du premier ordre (4.7), (4.10) et(4.14) :

$$\begin{cases} \frac{dC_a(t)}{dt} = \frac{Q}{V} (C_{a0} - C_a(t)) - K_0 C_a(t) \exp^{-E/RT(t)} \\ \frac{dT(t)}{dt} = \frac{Q}{V} (T_0 - T(t)) - \frac{\Delta H_r}{\rho C_p} C_a(t) K_0 \exp^{-E/RT} - \frac{h_a}{\rho V C_p} (T(t) - T_c(t)) \\ \frac{dT_c(t)}{dt} = \frac{Q_c}{V_c} (T_{c0} - T_c(t)) + \frac{h_a}{\rho_c V_c C_{pc}} (T(t) - T_c(t)) \end{cases} \quad (4.15)$$

Les valeurs des paramètres du système considéré sont résumées dans le tableau 4.2

Paramètre	Description	Valeur Nominale
$Q$	Débit du réactif	100 l/min
$V$	Volume du réacteur	100 l
$K_0$	Constante de vitesse	$7.2 \cdot 10^{10} \text{ min}^{-1}$
$E/R$	Énergie d'activation	$1.10^4 \text{ K}$
$T_0$	Température d'alimentation	$350^\circ \text{ K}$
$T_{c0}$	Température d'alimentation du réfrigérant	$350^\circ \text{ K}$
$\Delta H_r$	Chaleur de réaction	$-2.10^5 \text{ cal/mol}$
$C_p, C_{pc}$	Constante de chaleur spécifique	$1 \text{ cal/g}^\circ\text{K}$
$\rho_c, \rho$	Densité des liquides	$1.10^3 \text{ g/l}$
$h_a$	Coefficient de transfert	$7.10^5 \text{ cal/min } ^\circ\text{K}$

Tableau 4.2 : Constantes thermodynamiques chimiques du CSTR

A l'équilibre et pour une concentration nominale voulue de  $C_a = 0.1 \text{ mol/l}$ , en utilisant l'équation (4.15), on peut déduire les paramètres du régime permanent du réacteur chimique qui sont :  $T = 438.54 \text{ }^\circ\text{K}$ ,  $T_c = 425,48 \text{ }^\circ\text{K}$  et  $Q_c = 121,1 \text{ l/min}$ .

Dans le cas où le dispositif de refroidissement n'est pas actionné ( $Q_c = 0 \text{ l/min}$ ), on voit d'après la figure 4.5, que la réaction est exothermique, et elle présente une croissance rapide de la température  $T$  du réactif autour de la concentration nominale  $C_a = 0.1 \text{ mol/l}$  du réacteur. Il en résulte un réactif indésirable.

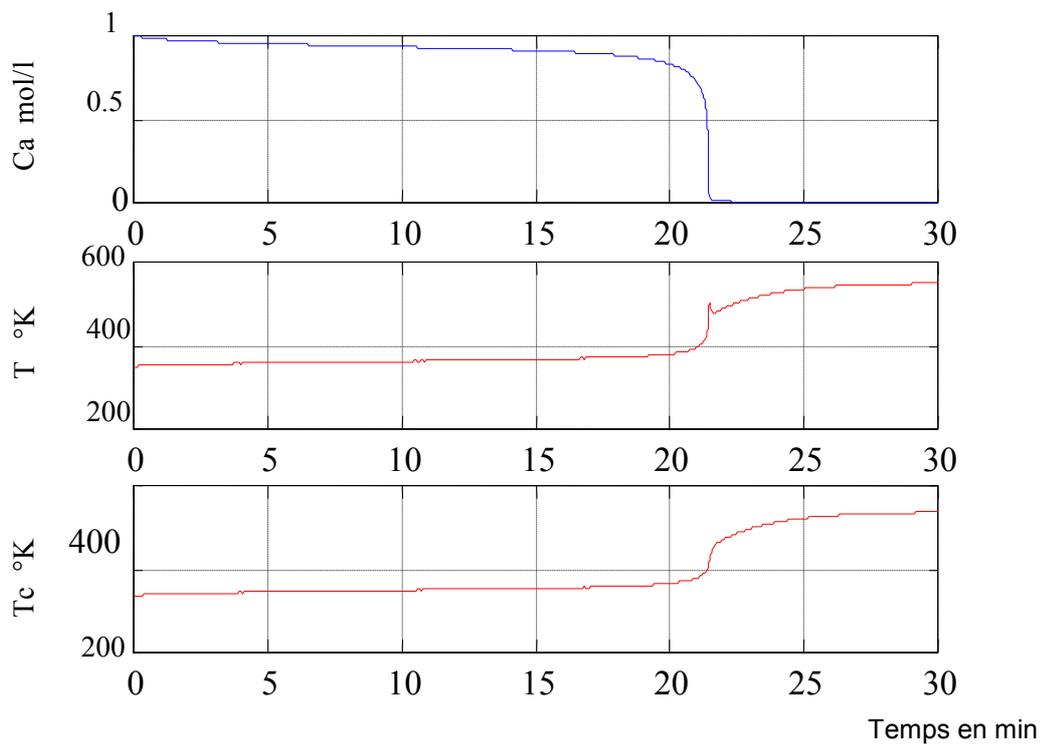


Figure 4.5 : Evolution des paramètres  $C_a$ ,  $T$  et  $T_c$  du réacteur quand  $Q_c = 0$

#### 4.2.5.2. L'identification du CSTR

Pour l'identification du CSTR, nous avons utilisé :

- un ensemble de données de 7500 échantillons avec une période d'échantillonnage  $T_s = 0.1 \text{ min}$  (figure 4.6). Cet ensemble est divisé en deux parties: 4000 échantillons pour faire l'apprentissage au réseau et 3500 pour valider le modèle. Les échantillons de l'apprentissage sont centrés et réduits à la même valeur de la variance (figure 4.7).

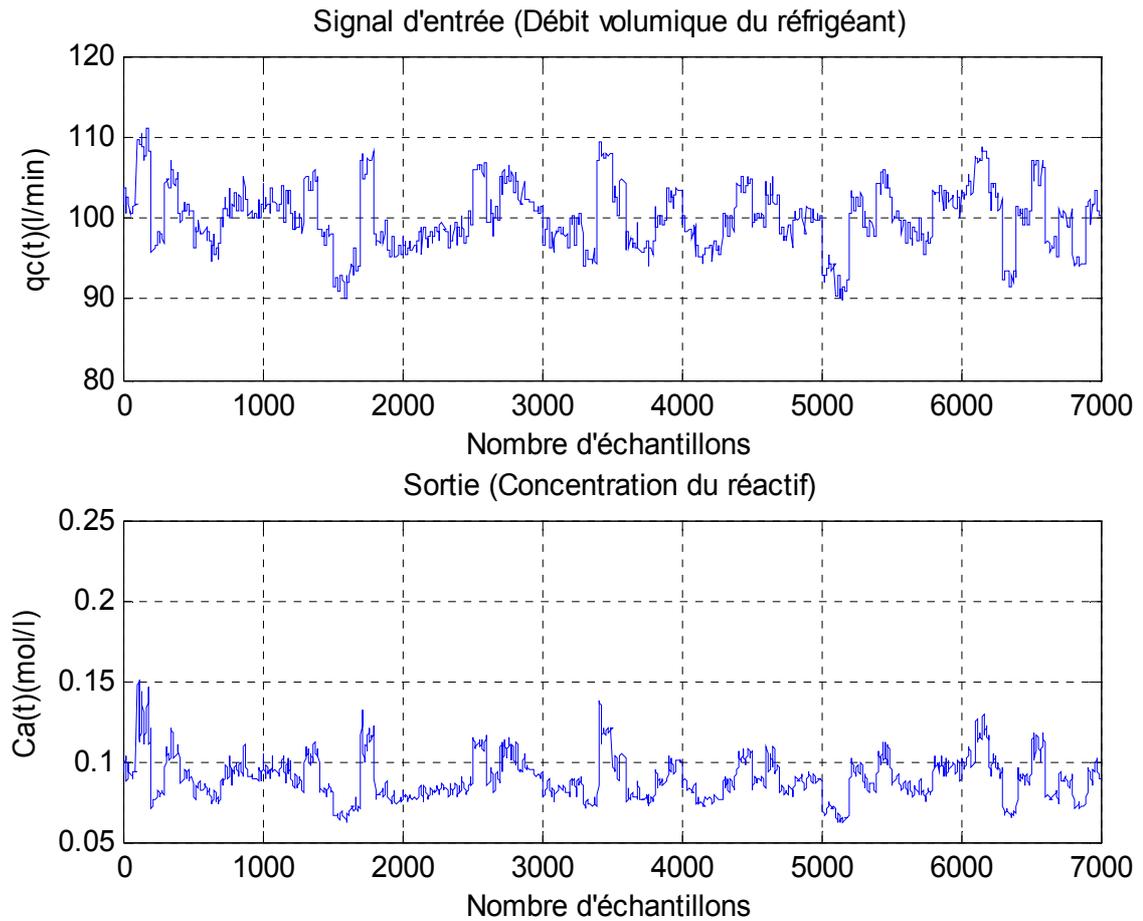


Figure 4.6 : Données d'apprentissage et de validation

- un réseau de neurones MLP à deux couches (une couche cachée et une couche de sortie), avec 8 neurones sur la couche cachée. La fonction d'activation des neurones cachés est la tangente hyperbolique et celle du neurone de sortie est linéaire.
- un modèle neuronal NARX, dont la structure (figure 4.7) est donnée par l'équation ci-après :

$$\hat{C}_a(k) = NN(C_a(k-1), C_a(k-2), q_c(k-2), q_c(k-1)) \quad (4.2)$$

où :  $NN()$  est le réseau de neurone utilisé ;

- l'algorithme de Levenberg-Marquardt pour l'apprentissage du réseau.

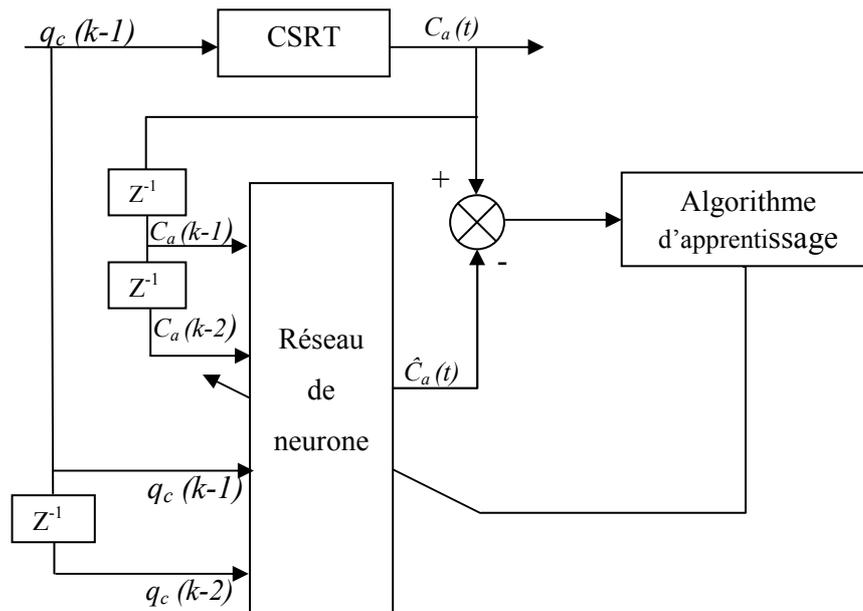


Figure 4.7 : Schéma de l'identification

La figure 4.8 montre l'évolution de l'erreur quadratique moyenne pendant la phase d'apprentissage. Pratiquement une valeur de l'ordre de  $10^{-9}$  est obtenue au bout de 100 itérations. Les sorties du système et du modèle sont représentées sur la figure 4.9, les deux courbes sont superposées, et l'erreur de prédiction est très faible.

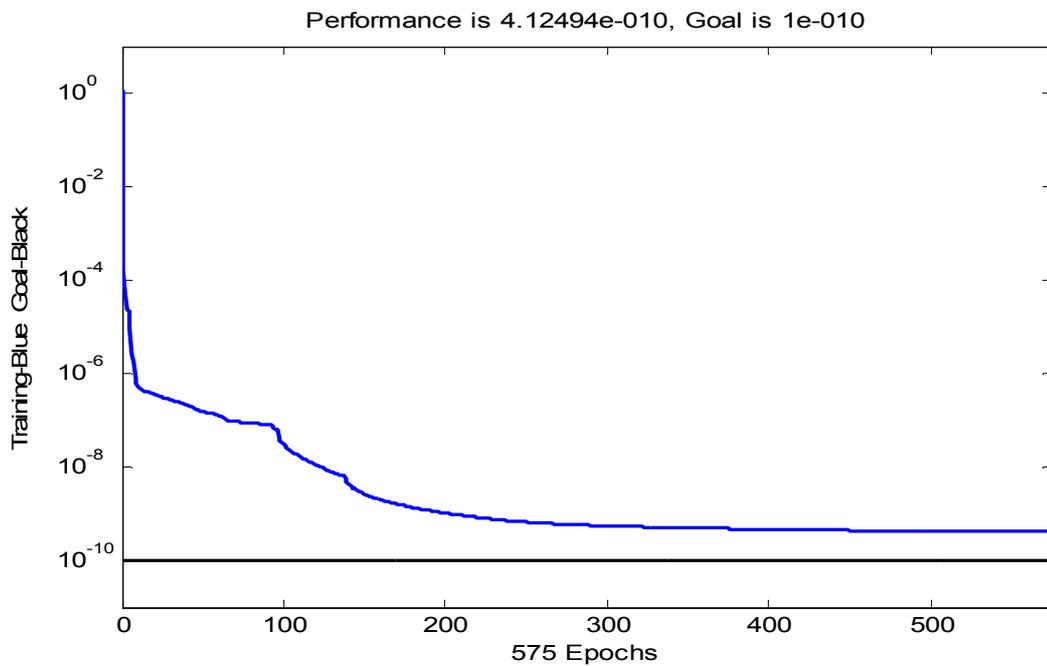


Figure 4.8 : Evolution de l'erreur quadratique moyenne pendant l'apprentissage

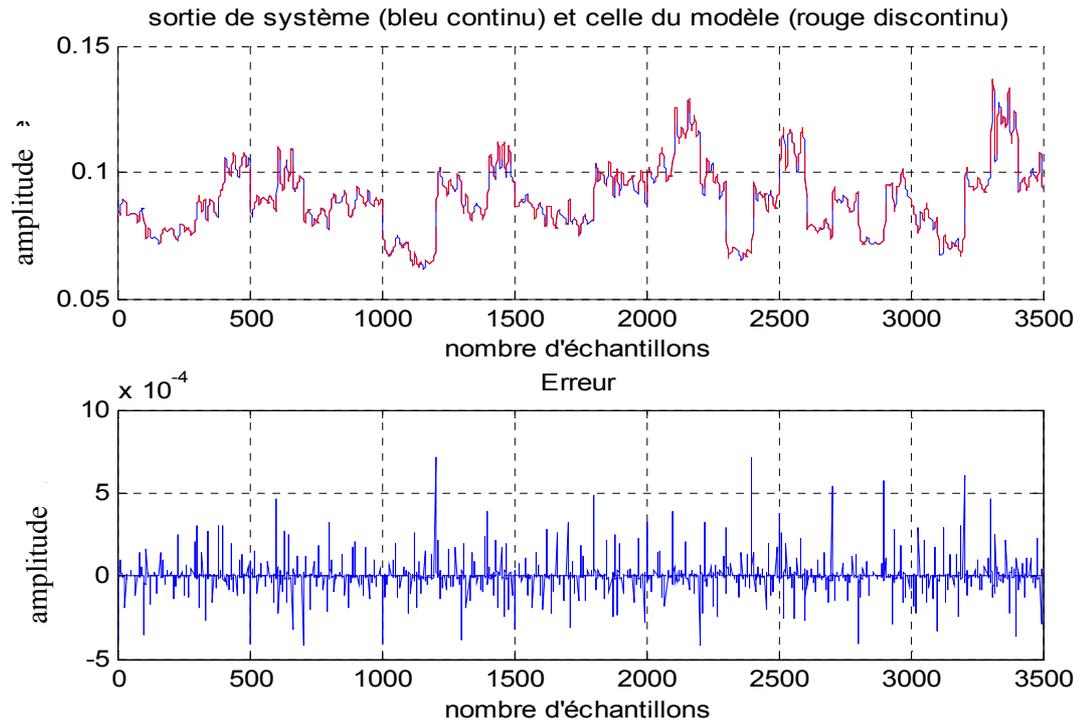


Figure 4.9 : Validation du modèle

#### 4.3. Mise en œuvre du contrôleur

Nous avons utilisé une trajectoire de référence en forme d'escalier (figure 4.10) pour mettre en évidence le caractère non linéaire du système. La durée de chaque pas est de 10 min et les valeurs minimale et maximale de son amplitude sont respectivement 0.08 mol/l et 0.145 mol/l. La loi de commande prédictive a été calculée en utilisant les approches citées plus haut.

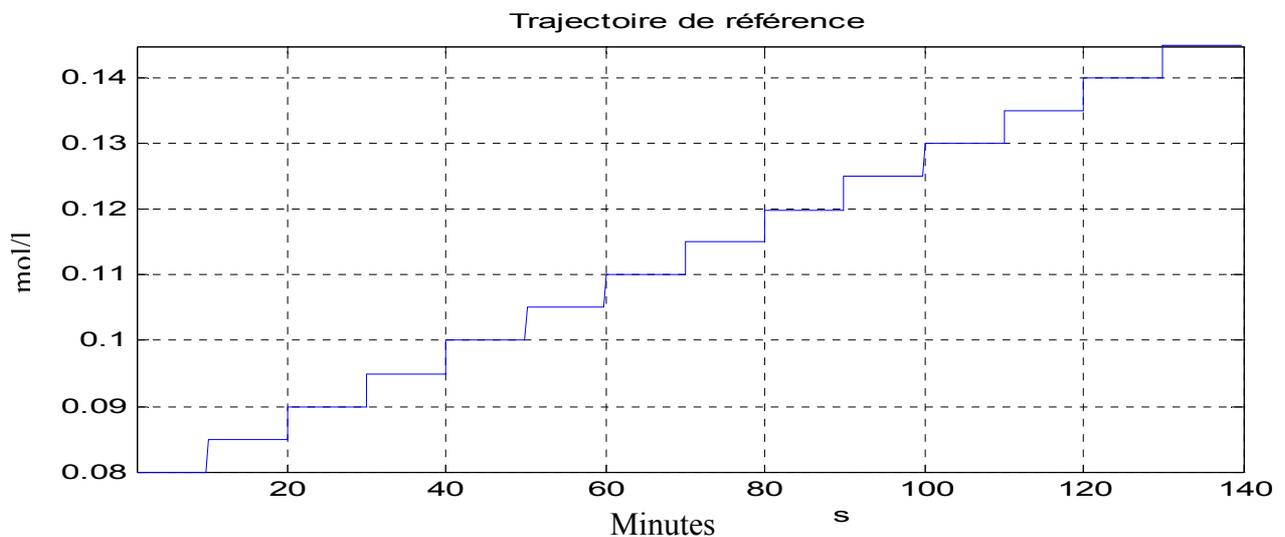


Figure 4.10 : Trajectoire de référence

#### 4.3.1. Calcul de la loi de commande du CSRT avec la GPC

Dans ce paragraphe nous envisageons la mise en place de la commande prédictive généralisée (GPC) pour le réacteur CSRT. Nous considérons deux cas :

- **1<sup>er</sup> cas : GPC(linéaire)**

Nous utilisons un modèle numérique linéaire, à paramètres fixes, obtenu par identification du CSTR. Il est donné par [14]:

$$H(z) = \frac{1.653 \times 10^{-4} z^{-1}}{1 - 2.43z^{-1} + 2.40 z^{-2} - 1.189z^{-3} + 0,269z^{-4}} \quad (4.3)$$

- **2<sup>ème</sup> cas : GPCN (GPC Neuronale).**

Nous utilisons un modèle linéaire à paramètres variables. Ce modèle est obtenu par linéarisation instantanée, à chaque période d'échantillonnage ( $T_s = 0.1min$ ) du modèle neuronal du CSTR.

Les figures 4.11a et 4.11.b montrent les performances obtenues, pour les valeurs des paramètres de réglage  $N_1 = 1$ ,  $N_2 = 10$ ,  $N_u = 2$ ,  $\lambda = S = 10^{-4}$ .

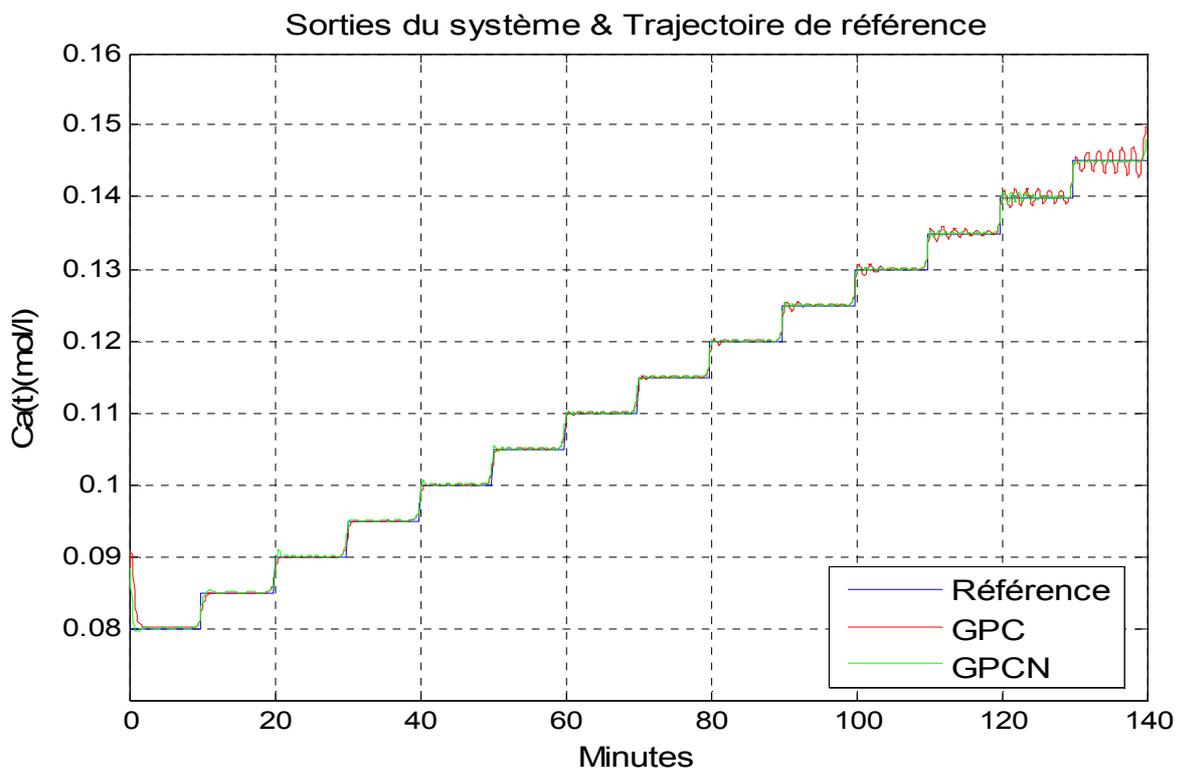


Figure 4.11a: Performances du contrôleur

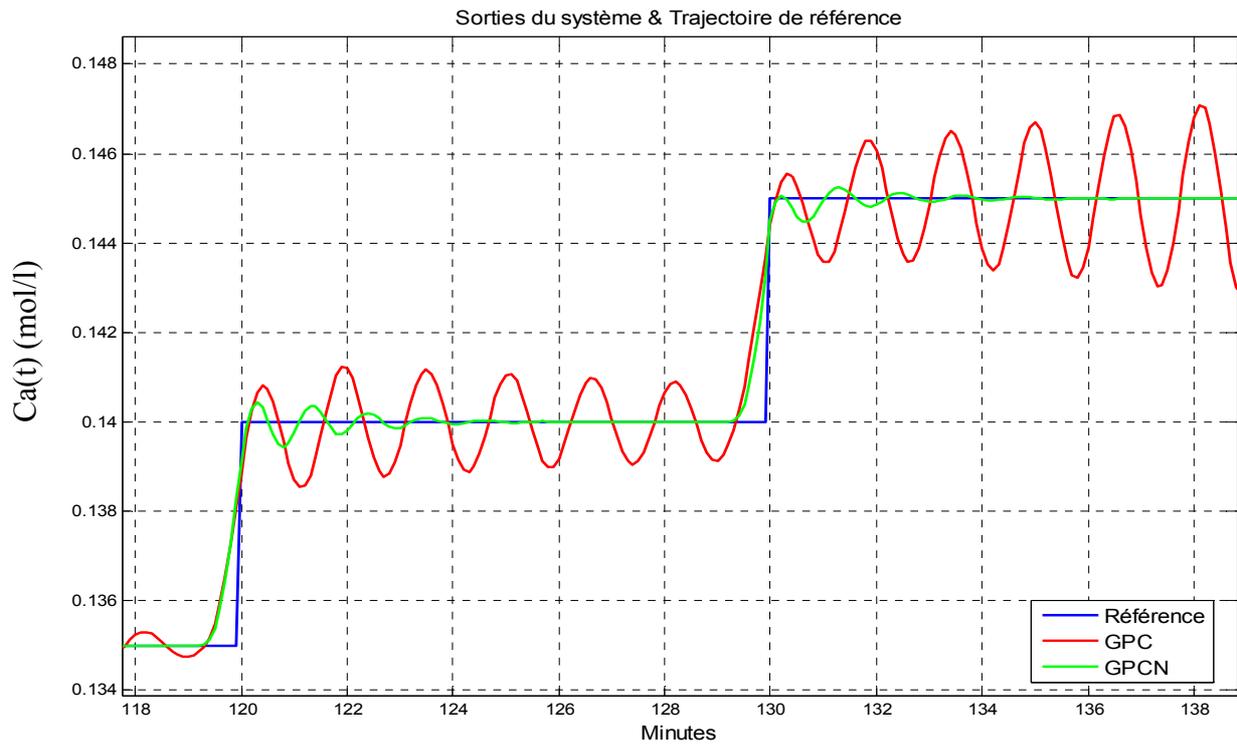


Figure 4.11b : Vue agrandie de la sortie & référence de la figure 4.11a

Nous remarquons que la GPCN a donné des performances plus meilleures que la GPC, le signal de sortie obtenu par cette dernière ne converge pas lorsque l'amplitude du signal de référence augmente. Donc il y a une dégradation des performances du régulateur GPC lorsqu'on s'éloigne du point de fonctionnement (Figure 4.12).

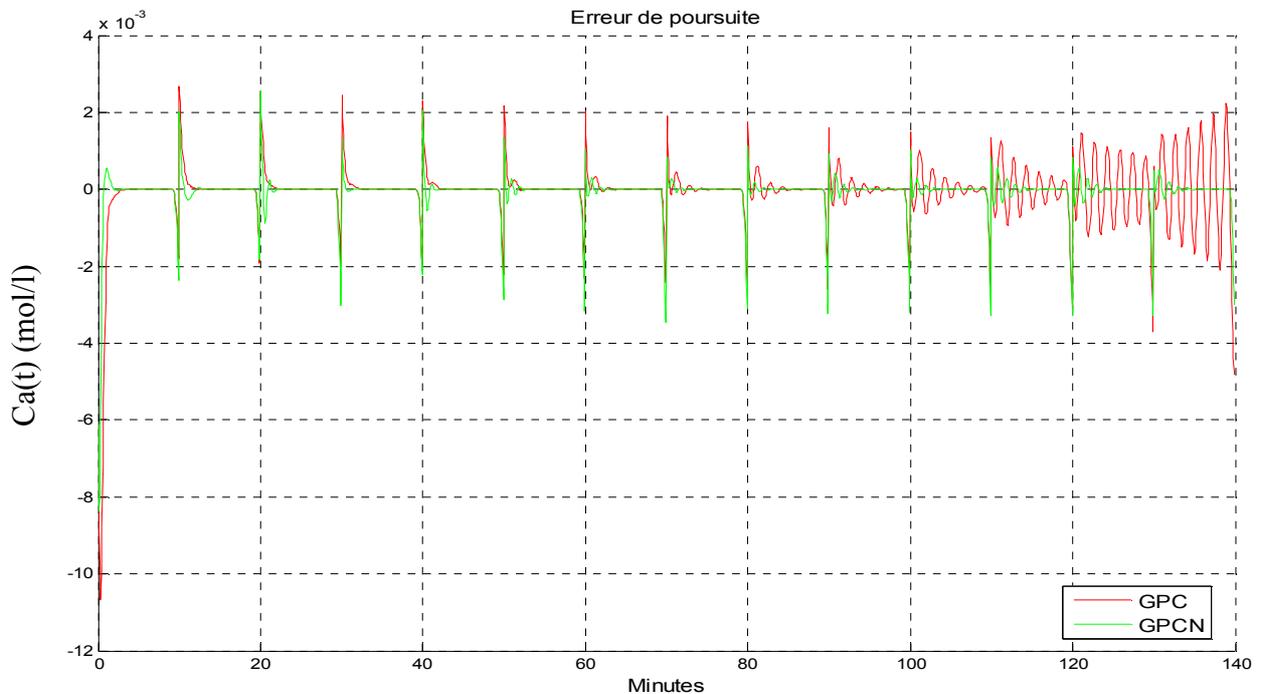


Figure 4.12 : Erreur de poursuite

#### 4.3.2. Calcul de la loi de commande en utilisant l'approche par décomposition

Maintenant, nous allons calculer la loi commande en utilisant l'approche par décomposition et nous considérons aussi deux cas :

- **1<sup>er</sup> cas : CPNMLPF (Commande Prédicative Neuronale avec Modèle Linéaire à Paramètres Fixes).**

La partie réponse libre est calculée en utilisant le modèle neuronal du CSTR tandis que la partie réponse forcée est calculée en utilisant le modèle linéaire à paramètres fixes du CSTR obtenu par identification.

- **2<sup>ème</sup> cas : CPNMLPV (Commande Prédicative Neuronale avec Modèle Linéaire à Paramètres variables).**

La partie réponse libre est calculée en utilisant le modèle neuronal du CSTR tandis que la partie réponse forcée est calculée en utilisant le modèle linéaire à paramètres variables obtenu par linéarisation du modèle neuronal du CSTR à chaque période d'échantillonnage.

Les paramètres de réglage dans les deux cas sont les suivants:  $Q = I^{N2 \times N2}$ ,  $S = 10^{-4} I^{Nu \times Nu}$ ,  $N1 = 1$ ,  $N2 = 8$ ,  $Nu = 1$ . Les performances des contrôleurs obtenues sont représentées sur la figure 4.13a, 4.13b et 4.14.

Nous remarquons que le signal de sortie obtenu dans le cas de CPNMLPV a un dépassement d'amplitude plus petite que le signal de sortie obtenu dans le cas de la CPNMLPF. La figure 4.13b montre bien que l'approche que nous avons développée (CPNMLPV) permet de stabiliser la sortie à sa valeur désirée plus rapidement que l'approche utilisant un modèle linéaire à paramètres fixes (CPNMLPF).

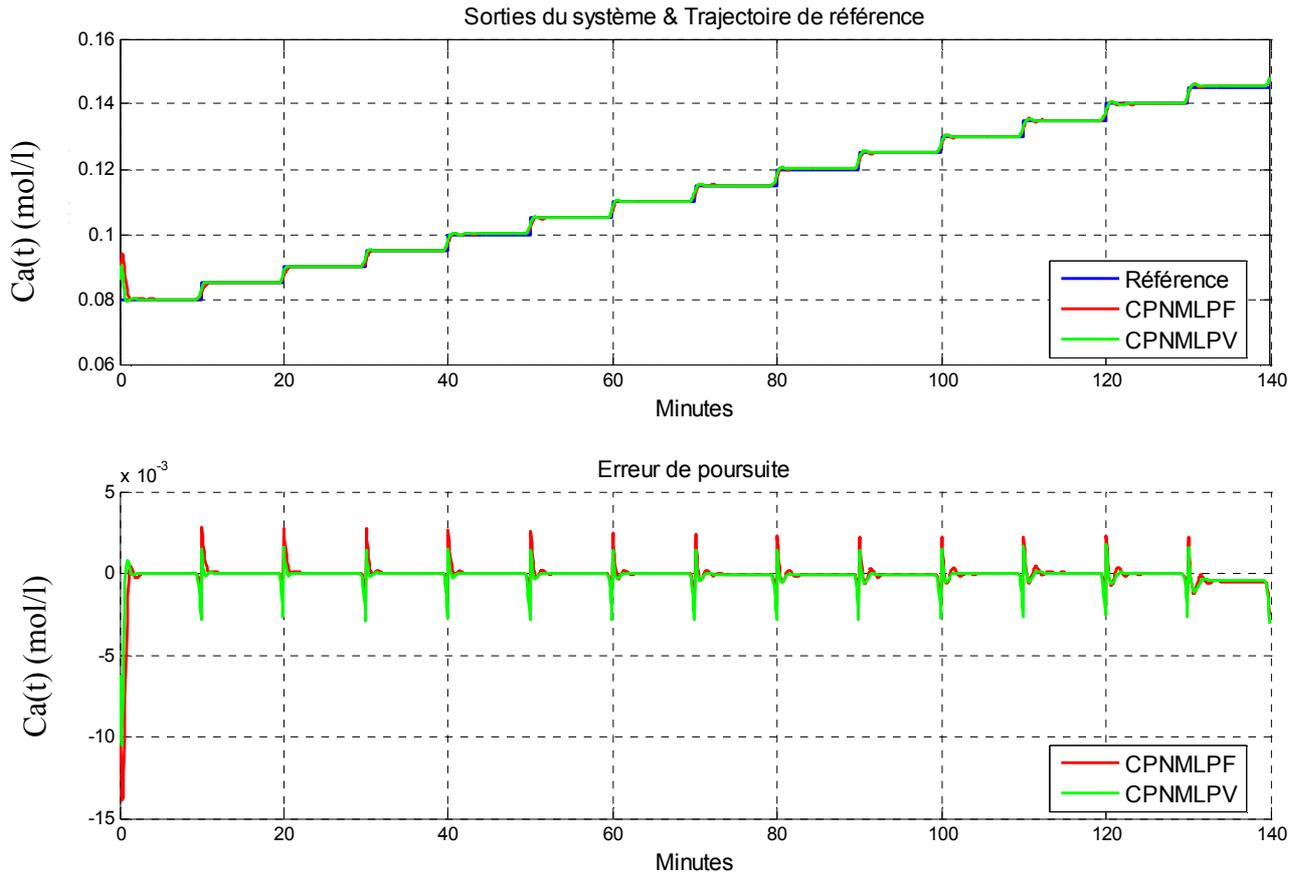


Figure 4.13a : Performances du contrôleur

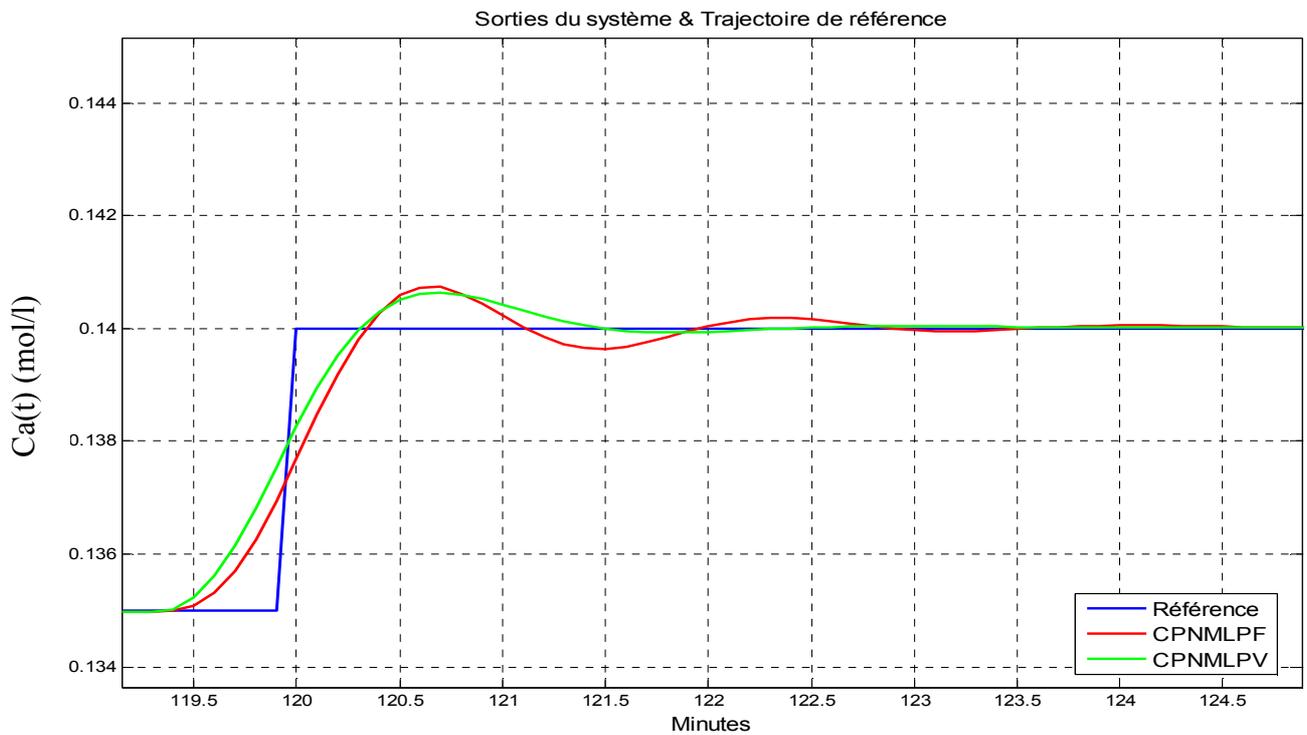


Figure 4.13b : Vue agrandie de la sortie &amp; référence de la figure 4.13a

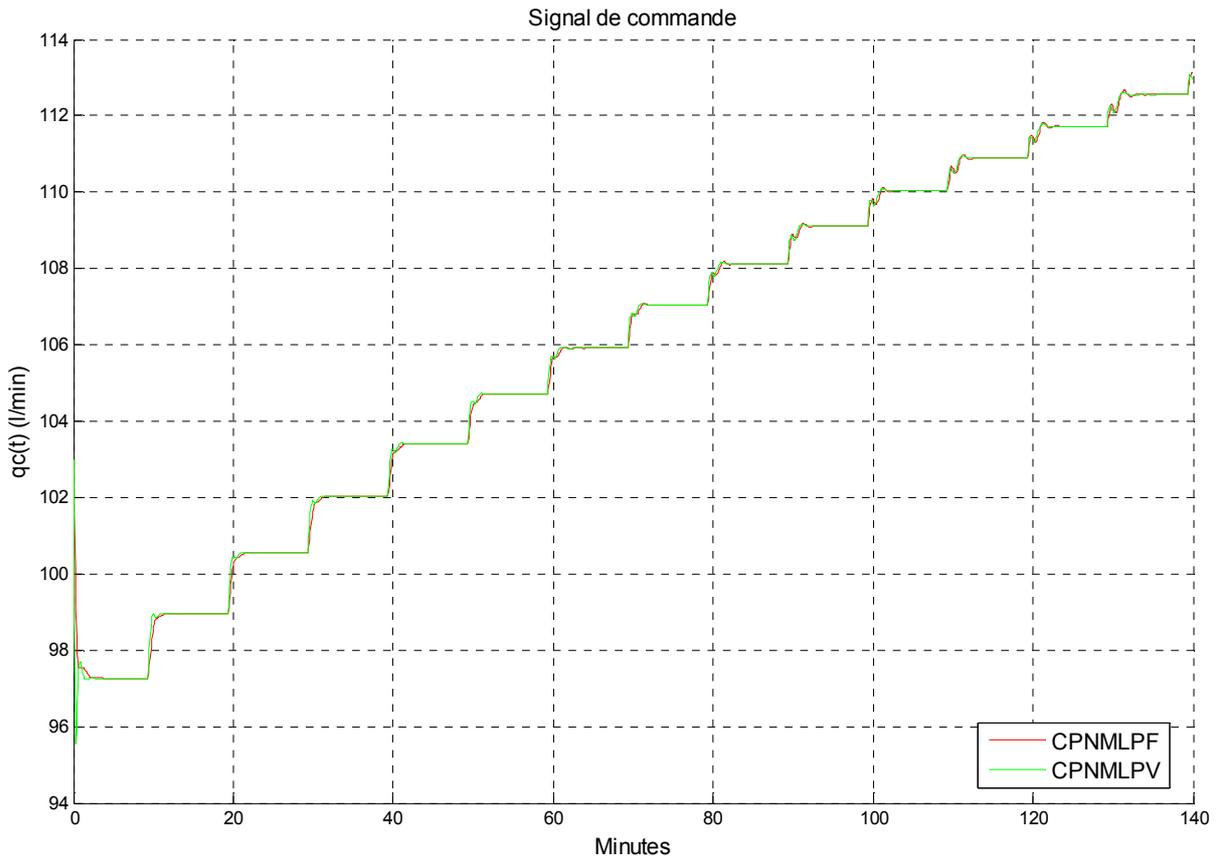


Figure 4.14: Signal de commande

Ensuite, pour les mêmes valeurs des paramètres définies ci-dessus, nous avons appliqué au milieu de chaque pas une perturbation d'amplitude égale à  $\pm 0.005$  mol/l (figure 4.15).

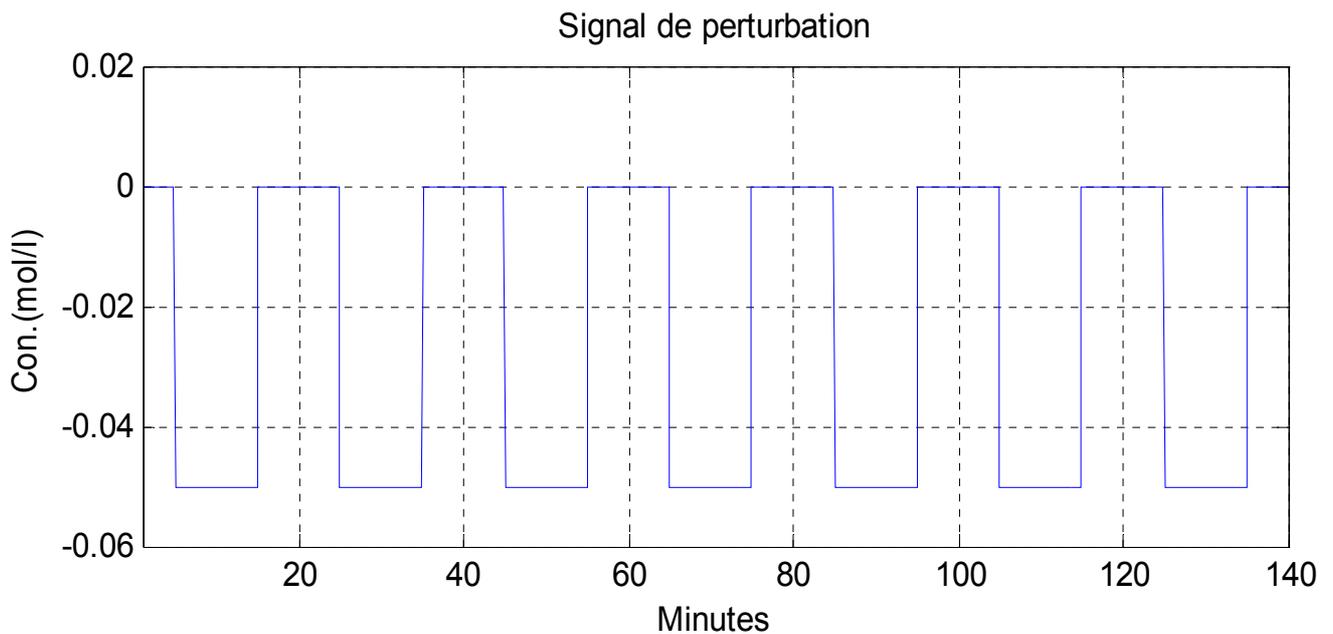


Figure 4.15: Signal de perturbation

Les performances obtenues des deux régulateurs (CPNMLPF, CPNMLPV) sont représentées sur les figures 4.16 et 4.17. Elles montrent que le contrôleur compense rapidement l'effet de cette perturbation par des pics d'amplitude raisonnable pour les deux algorithmes, mais l'erreur de poursuite est plus faible dans le cas de CPNMLPV (Figure 4.18)

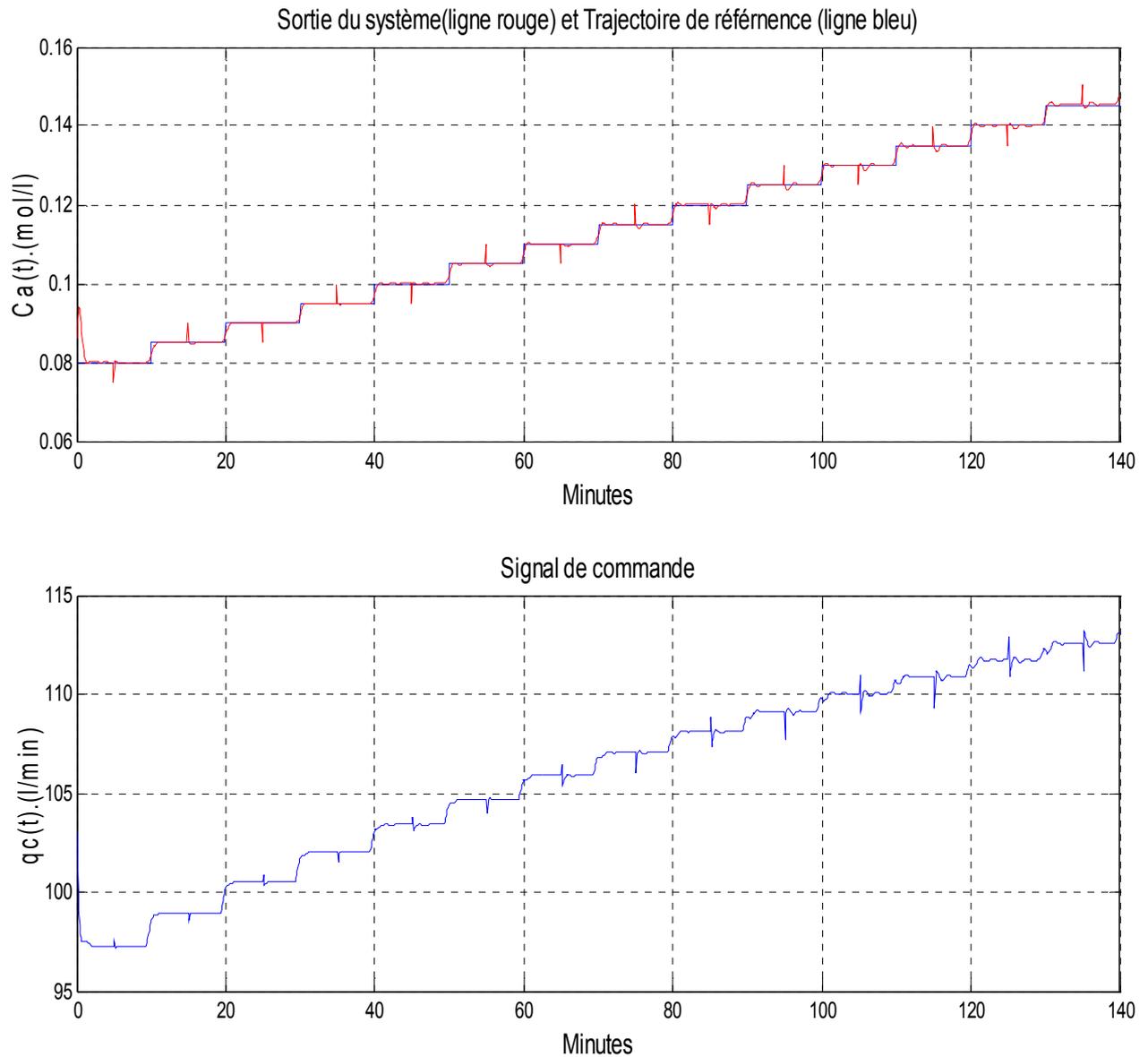


Figure 4.16 : Performance du contrôleur avec perturbations, cas CPNMLPF

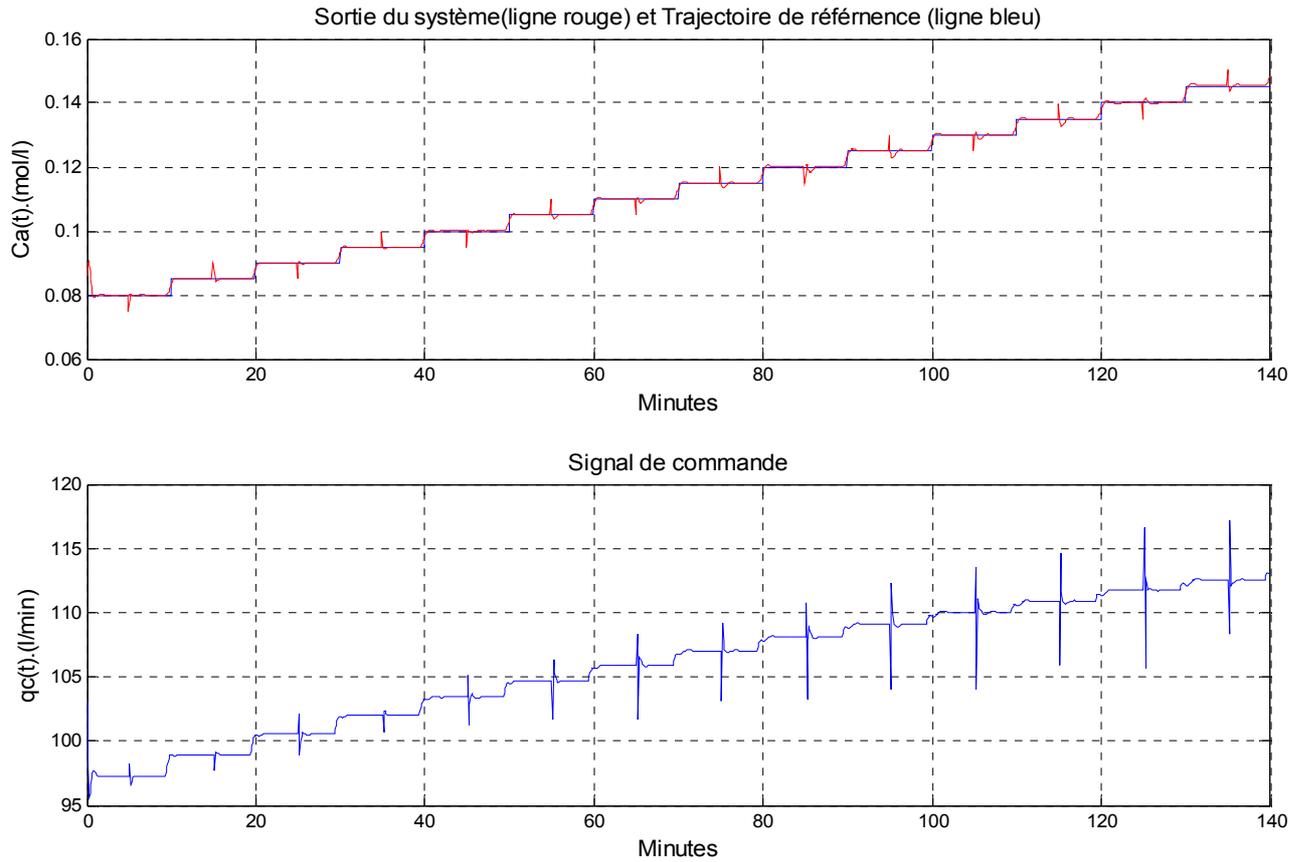


Figure 4.17 : Performance du contrôleur avec perturbations, cas CPNMLPV

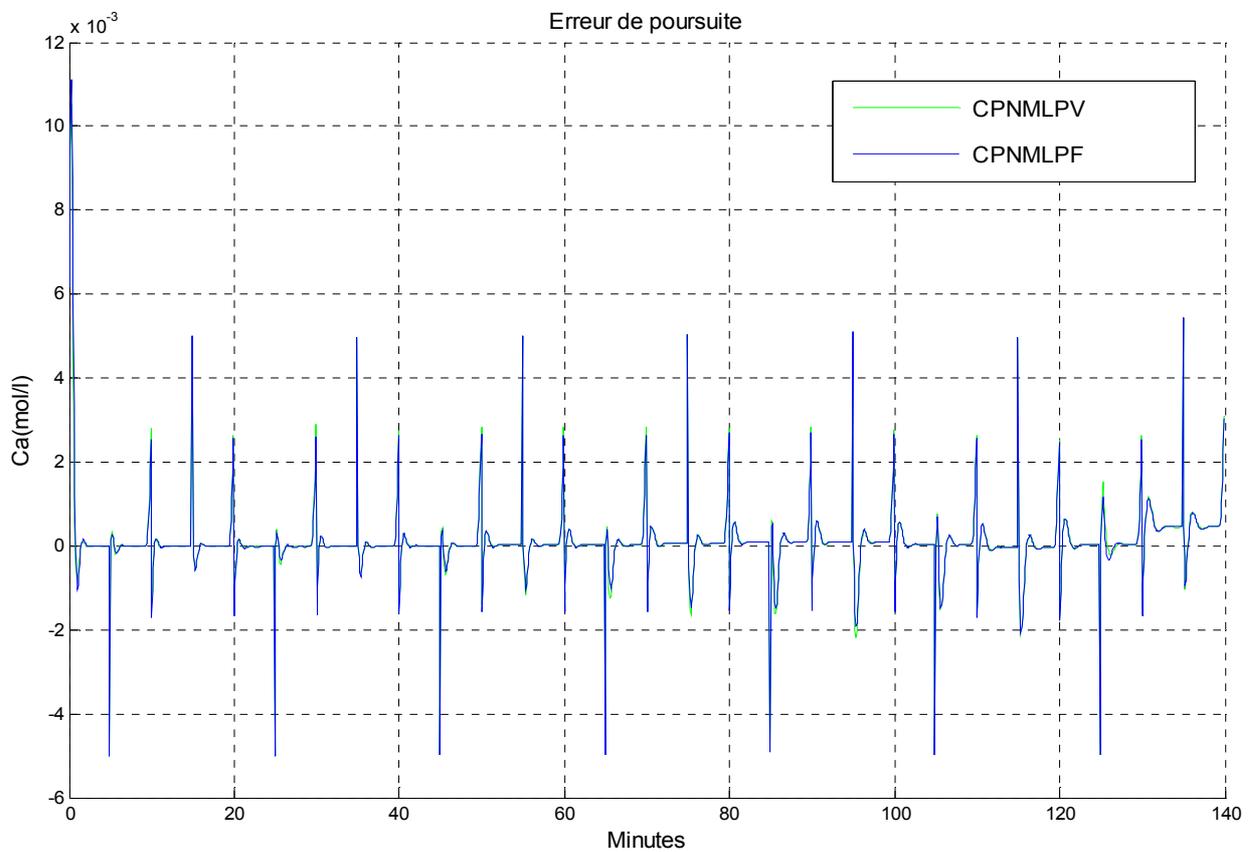


Figure 4.18 : Erreur de poursuite avec perturbation, cas CPNMLPF et CPNMLPV

Maintenant nous allons montrer l'influence des paramètres de réglage ( $N_2, S$ ) sur les performances de l'approche développée (CPNMLPV). Les résultats obtenus sont illustrés dans les figures 4.19 et 4.20

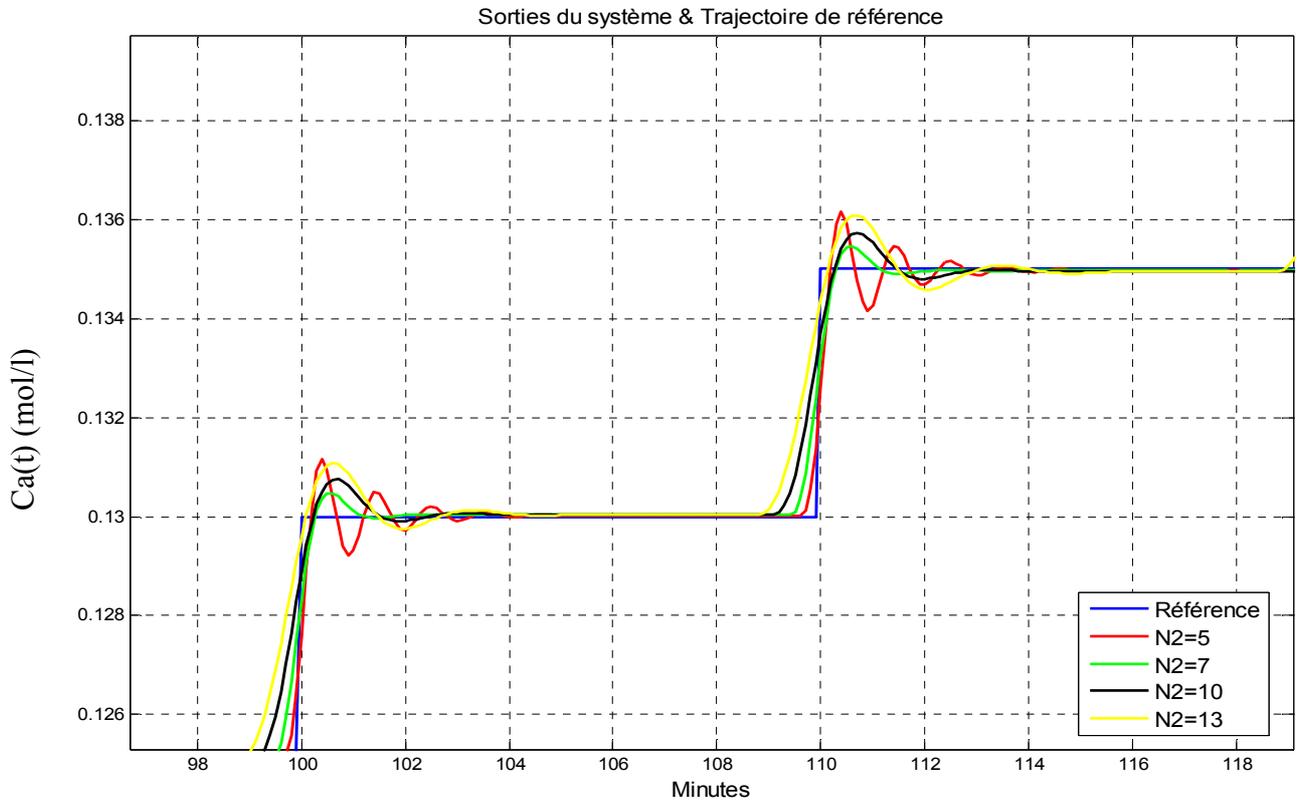


Figure 4.19 : Performances obtenues pour différentes valeurs de  $N_2$  ( $S=10^{-5}$ )

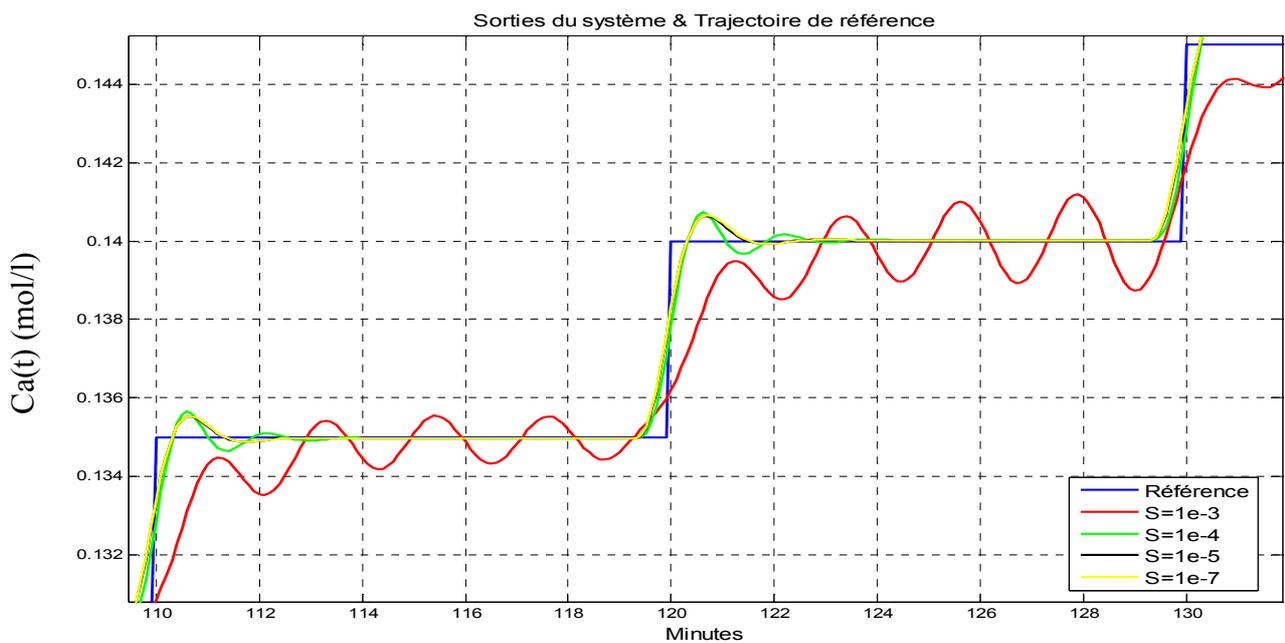


Figure 4.20 : Performances obtenues pour différentes valeurs de  $S$  ( $N_2=8$ )

### 4.3.3. Etude comparative

Sur la figure 4.21 nous avons représenté une partie des sorties du système obtenues en utilisant les quatre algorithmes : GPC, GPCN, CPNMLPF et CPNMLPV. Nous remarquons que les régulateurs GPC et GPCN n'arrivent pas à stabiliser la sortie à sa valeur désirée contrairement aux régulateurs CPNMLPF, CPNMLPV. Ceci justifié l'utilisation des modèles non linéaires et en particulier les réseaux de neurones. Avec l'approche CPNMLPV nous avons obtenu la meilleure poursuite de la trajectoire de référence par rapport autres régulateurs. Ces résultats montrent bien l'apport de l'approche que nous avons mis en œuvre.

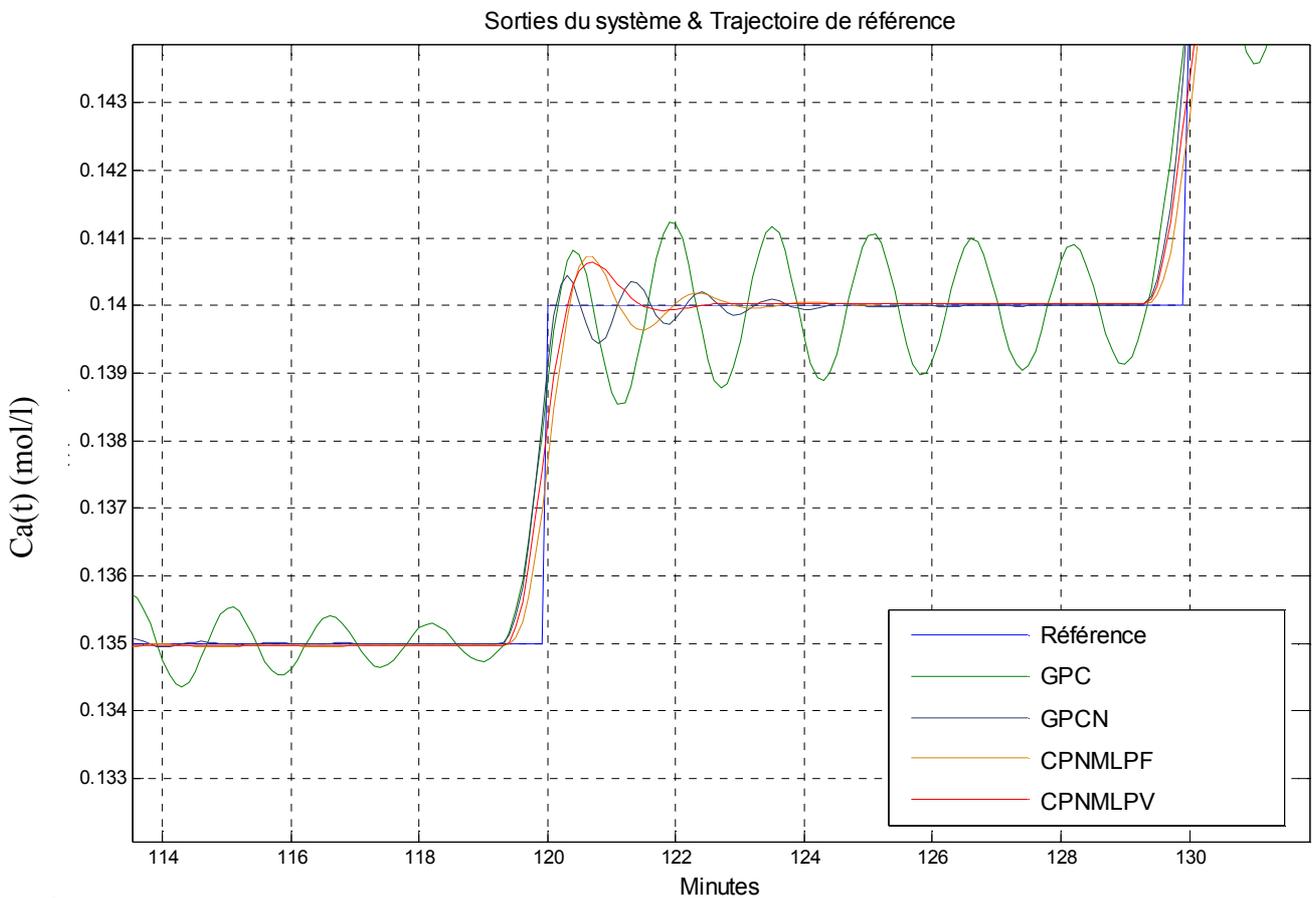


Figure 4.21: Performances obtenues les quatre algorithmes de commande

### 4.4. Conclusion

Dans ce chapitre, nous avons mis en évidence les performances des régulateurs prédictifs étudiés dans les chapitres précédents en considérant la commande du réacteur parfaitement mélangé. Le CSTR est un système complexe fortement non linéaire, souvent considéré comme étant un système de référence. Cette application a permis de montrer

l'efficacité des réseaux de neurones dans la modélisation des systèmes non linéaires et l'efficacité de la loi de la commande prédictive à modèle neuronal basée sur la stratégie de la décomposition de la réponse totale en ses deux composantes libre et forcé. Plus précisément nous avons montré par simulation que le régulateur CPNMLPV donne les meilleures performances.

## CONCLUSIONS ET PERSPECTIVES

Dans le cadre de la conception d'une loi de commande prédictive pour les système non linéaire, notre travail part de l'élaboration d'un modèle non linéaire, qui approche le mieux possible le comportement dynamique du système à contrôler, et s'étend à la recherche d'une solution rapide et efficace au problème d'optimisation non linéaire qui consiste à trouver la loi de commande en minimisant une fonction de coût donnée. Pour cela, nous avons utilisé les réseaux de neurones, en raison de leurs propriétés d'approximation universelle, de généralisation et d'adaptation. En particulier, nous avons choisi les réseaux MLP statiques à cause de la simplicité de leurs algorithmes d'apprentissage.

Bien que les réseaux de neurones constituent un outil, simple et puissant, de modélisation des systèmes non linéaires, leur utilisation dans le cadre de la commande prédictive conduit à un problème d'optimisation, non linéaire, complexe et dont la solution par les méthodes itératives est généralement coûteuse en terme du temps de calcul. Il est donc important de développer d'autres techniques qui permettent de surmonter cette contrainte. Le travail que nous avons réalisé, dans ce contexte, a permis de trouver une expression analytique à la loi de commande en minimisant une fonction de coût quadratique.

L'approche que nous avons considérée est basée sur l'utilisation d'un modèle de prédiction non linéaire de structure particulière. Ce modèle, constitué d'une partie non linéaire dépendante des conditions initiales disponibles à l'instant actuel et d'une partie linéaire dépendante des incréments futurs du signal de commande, est établi à partir du modèle neuronal du système. L'utilisation de tel modèle, pour prédire les valeurs futures de la sortie du système, nous a permis d'appliquer la technique de la commande prédictive généralisée linéaire pour mettre au point la loi de commande. Les performances de cette approche sont mises en évidence en considérant la commande d'un réacteur continu parfaitement mélangé. Nous avons obtenu des résultats satisfaisants, en particulier la bonne poursuite de la trajectoire de référence avec une action de commande suffisamment lisse.

Enfin, l'étude comparative que nous avons effectuée montre l'apport de l'approche envisagée.

Plusieurs perspectives à ce travail peuvent être envisagées. Une première démarche consiste à l'ajustement automatique, en ligne et en temps réel, des paramètres de réglages de la commande prédictive qui permet de maintenir un même niveau de performances.

L'une des raisons que fait de la méthodologie prédictive une technique de commande puissante et très utilisée dans le domaine industriel, est la prise en compte des contraintes pour la mise au point de la loi de commande. Il est donc recommandable de considérer le problème de la commande prédictive à modèle neuronal avec contraintes. La prise en compte des contraintes et l'utilisation d'un modèle neuronal posent un véritable problème d'optimisation non linéaire et non convexe. Un travail devrait donc se faire dans ce contexte pour développer des méthodes permettant de trouver une solution rapide et efficace à ce problème. Les méthodes d'optimisation par point intérieur (interior point methods) pourraient être considérées.

Une autre perspective qui mérite d'être étudiée consiste à envisager le problème de stabilité. En effet, Il n'existe aucune théorie générale permettant de démontrer d'une façon systématique, la stabilité des systèmes non linéaires. Toutes les méthodes développées sont basées sur le principe de l'hyperstabilité de Lyapunov pour lequel le choix de la fonction de Lyapunov n'est pas toujours évident. Ce problème de stabilité devient beaucoup plus complexe pour les systèmes de commande à base des réseaux de neurones.

Enfin, il apparaît nécessaire de sortir du cadre de la simulation, au sein duquel on a pu montrer l'intérêt du travail effectué, pour tester de façon expérimentale les bénéfices réellement attendus.

## APPENDICE

### LISTE DES SYMBOLES ET DES ABREVIATIONS

$\hat{y}(t + j)$  : Sortie prédite à l'instant  $t + j$ .

$w(t + j)$  : Consigne appliquée à l'instant  $t + j$ .

$\Delta u(t + j - 1)$  : Incrément de commande à l'instant  $t + j - 1$ .

$y(t)$  : Sortie du processus.

$u(t)$  : Commande appliquée au système.

$q^{-1}$  : Opérateur retard.

$\Delta(q^{-1}) = 1 - (q^{-1})$  Opérateur différence.

$T_s$ : Période d'échantillonnage

$N_1$  : Horizon de prédiction minimal sur la sortie.

$N_2$  : Horizon de prédiction maximal sur la sortie.

$N_u$  : Horizon de prédiction sur la commande.

$\lambda$  : Coefficient de pondération sur la commande.

*CARMA*: Controlled AutoRegressive Moving Average.

MLP : Multi Layer Perceptron, un réseau multicouche.

RN : Réseau de Neurones.

*NN//* : Modèle Neuronal du système.

CSTR : Continus Stirred Tank Reactor, réacteur continu parfaitement mélangé.

GPC: Commande Prédictive Généralisée.

GPCN : GPC Neuronale.

CPNMLPF: Commande Prédictive Neuronale avec Modèle Linéaire à Paramètres Fixes.

CPNMLPV: Commande Prédictive Neuronale avec Modèle Linéaire à Paramètres variables.

## BIBLIOGRAPHIE

1. Clarke, D.W. et al., "Generalized Predictive Control -Part 1&2", *Automatica*, V. 23, (1987), n°2, 137-160,(1987).
2. Richalet ,J. et al., "Predictive Functional Control. Application to Fast and Accurate Robots", In: *Proceedings 10th IFAC World Congress, Munich, (1987)*.
3. Clarke, D.W., "Application of Generalized Predictive Control to Industrial Processes", *IEEE Control Systems Magazine*, (Avril 1988), 49-55.
4. Boucher, P., and D. Dumur, "La Commande Prédictive", Technip Editions, Paris, (1996).
5. Ehrlinger, A., Boucher, P., Dumur, D., "Unified Approach of Equality and Inequality Constraints in G.P.C", *Proceedings of the 5th IEEE Conference on Control Applications, Dearborn, (1996)*, 893-899.
6. Sockaert, P. O. M., Clarke, D., "Stabilising properties of constrained predictive control", *IEE Proceedings Control Theory Applications*, V. 141, n°5, (1994), 295-304.
7. Olaru. S., " Contribution a l'Etude de la Commande Prédictive sous Contraintes par Approche Géométrique", thèse doctorat, l'université paris xi Orsay, (Septembre 2005).
8. Parizeau, M., "Réseaux de Neurones", Université Lava, (2004).
9. Norgaard M., Ravn O., Poulsen N. K., Hansen L. K., "Neural Networks for Modelling and Control of Dynamic Systems", *A Practitioner's Handbook*, Springer-Verlag, London Limited (2000).
10. Narendra, K. S., "Neural Networks for Identification and Control", Center for systems science, Yale University.

11. Kara, K., "Application des Réseaux de Neurones à l'Identification des Systèmes Non Linéaires", Thèse de Magister, Université de Constantine, (1995).
12. Kara, K., "Contrôle Intelligent et Systèmes Autonomes (Application à la Robotique)", thèse de doctorat, Université de stif, (2006).
13. Hornik, K., Stinchcombe, M., White, H., "Multilayer Feed forward Networks are Universal Approximators", *Neural Networks*, V. 2, (1989), 359-366.
14. Espinosa Oviedo Jairo J., "Fuzzy Modeling and Control", Thèse de Doctorat, Katholieke Universiteit Leuven, Belgique, (April. 2001).
15. Narendra, K. S. and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE Trans. Neural Networks*, V. 1, n°1, (Mar 1990), 4-27.
16. Merabet, A., "Commande Non Linéaire à Modèle Prédicatif pour une Machine Asynchrone", Thèse doctorat, L'université du Québec.
17. Botto, M. A., Van Den Boom Ton, J. J., Krijgsman, A., José Sà Da Costa, "Predictive Control Based on Neural Network Models with I/O Linearization", *Int. Journal of Control*, V. 72, n° 17, (1999), 1538-1554.
18. Liu, G. P., Kadiramanathan, V., Billings, S. A., "Predictive Control for Non-Linear Systems Using Neural Networks", *Int. Journal of Control*, V. 71, n° 6, (1998), 1119-1132.
19. Camacho, E.F., Berenguel, M., "Application of Generalized Predictive Control to a Solar Power Plant", *Proc. of the 3rd IEEE conference on control applications*, Glasgow, (U.K. 1994), 1657-1662.
20. E.F. Camacho and Carlos Bordons, "Model Predictive Control", Springer-Verlag, London, (1999).
21. Nørgaard, M., Sørensen, P. H., Poulsen, N. K., Ravn, O., Hansen, L. K., "Intelligent Predictive Control of Nonlinear Processes Using Neural Networks", *Proceedings of the 1996 IEEE International Symposium on intelligent Control*, (1996), 301-306.
22. BENOUAËD, A., "Utilisation d'un Contrôleur Hybride Flou - H.G.A. pour le Contrôle d'un Réacteur Chimique du Type C.S.T.R.", Université Blida, (Juillet 2007).

