

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique  
Laboratoire de Commande des Processus



# Mémoire de Magister

Spécialité : **Automatique**

Option : **Automatique des Systèmes Industriels**

Présenté par

**ALLAD Mourad**

Ingénieur d'Etat en Automatique de l'UMMTO

**Thème**

**TECHNIQUES DE COMMANDE MODERNES APPLIQUEES AUX  
BIOREACTEURS**

Soutenu publiquement le 04/02/2006 devant le jury composé des membres :

Président :	M. S. BOUCHERIT	Professeur à l'ENP
Rapporteur :	R. ILLOUL	Chargé de Cours à l'ENP
Examineurs :	A. HELLAL	Professeur à l'ENP
	M. TADJINE	Maître de conférence à l'ENP
	H. CHEKIREB	Maître de conférence à l'ENP

## *Dédicaces*

*A Ma Famille*

*A Mes Proches*

*A Mes Amis(es)*

## *Remerciements*

*Le travail présenté dans ce mémoire a été mené au laboratoire de commande des processus du département de Génie Electrique de l'Ecole Nationale Polytechnique.*

*Je tiens à remercier Monsieur R. ILLOUL pour m'avoir apporté son aide à la réalisation de cette étude.*

*Je remercie les membres de jury, qui m'ont fait l'honneur de participer au jugement de ce travail.*

*Nous tenons également compte à remercier l'ensemble des enseignants d'automatique*

## ملخص

يرمي هذا المشروع إلى تسوية مشاكل تحكم في أجهزة التخمير شبه متواصلة المسماة (fed-batch) والمتواصلة. التحكم في الأجهزة الكيماوية تظهر صعوبات هائلة نظرا لميزتها الأخطية. إن استعمال الشبكات العصبونية الإصطناعية تسمح بعلاج المشكل المطروح أعلاه و التي تعتمد أساسا على تعلمها عن طريق خوارزمية الانتشار الخلفي للخطأ. الدراسة الأولى من هذه المذكرة تخص كيفية تشكيل النماذج لأجهزة التخمير، ثانيا تنص على تقديم التحكم العصبوني و نكرس المرحلة الأخيرة من عملنا لسلسلة من المحاكات أين نطبق التحكم الكلاسيكي باستعمال PI و من أجل إظهار التحكم العصبوني و هذا بهدف المقارنة بينهما والتحكم يكون في تركيز الجلوكوز.

**الكلمات المفتاحية:** أجهزة التخمير، كتلة حيوية، التخمير، أجهزة التخمير شبه متواصلة والمتواصلة، التحكم PI، التحكم العصبوني، خوارزمية الانتشار الخلفي.

## « Techniques de commande modernes appliquées aux bioréacteurs »

### Résumé

L'objet de ce mémoire concerne la commande des procédés semi-continus et continus de fermentation. La commande des systèmes chimiques ou biochimiques présentent de grandes difficultés à cause de leur grande complexité et leurs forte non linéarités. L'utilisation des réseaux de neurones artificiels permet de remédier à ce problème grâce à leur flexibilité, et leurs capacités d'apprentissage et d'approximation des fonctions non linéaires. Leur entraînement se base sur l'algorithme de rétropropagation de gradient.

Le premier chapitre du mémoire concerne les généralités sur les processus de fermentation. Le deuxième et le troisième traitent des réseaux de neurones et de leur utilisation dans la commande, enfin le dernier présente les simulations effectuées afin de comparer la commande classique PI et la commande par réseaux de neurones pour deux processus de fermentation, le premier est le bioréacteur fed-batch et le second le bioréacteur continu, la variable à commander étant la concentration du glucose.

### Mots clés

Bioréacteurs, biotechnologie, biomasse, fermentation, procédé semi-continu, procédé continu, régulation PI, régulateur neuronal, algorithme de rétropropagation

## «Modern control applied to bioreactors »

### Abstract

This memory deals with bioreactors control. These bioreactors are used for fermentation either in fed-batch mode or in continuous mode. The control of chemical or biochemical systems is rather difficult, because of their complexity and strong nonlinearities. The use of neural networks can solve the problem thanks to their flexibility, their learning capacities, and their ability to approximate any non linear function. Training of these networks is done using the backpropagation algorithm.

The first part of this memory presents the biochemical processes, especially the fermentation ones, the second and third part present neural networks and their use in control. Finally the last part presents our simulation results. We compare the performances of classic regulation using PI with neural regulation, in two cases; first with a fed-batch process, and second with a continuous process. The variable to control is the glucose concentration.

### Key words

Bioreactors, biotechnology, biomass, fermentation, fed- batch process, continuous process, PI regulation, neurocontroller, backpropagation algorithm.

# Sommaire

<b>Introduction générale</b> .....	1
 <b>Chapitre 1: Fermentation et caractéristiques des bioréacteurs</b>	
1.1 Introduction .....	3
1.2 Bioprocédés .....	4
1.2.1 Généralités sur le processus de fermentation .....	4
1.2.2 Enzymes .....	4
1.2.3 Facteurs de croissance microbienne .....	5
1.3 Fermenteur .....	6
1.4 Modes de fonctionnement .....	7
1.4.1 Fermentation en mode discontinu (batch) .....	7
1.4.2 Fermentation continue .....	7
1.4.3 Fermentation semi-continue (fed-batch) .....	9
1.4.4 Les avantages d'un procédé semi-continu de fermentation .....	9
1.5 Dynamique cellulaire .....	10
1.6 Modélisation des cinétiques de réaction .....	12
1.6.1 Vitesses de réaction .....	12
1.7 Modélisation des vitesses spécifiques .....	13
1.7.1 Taux de croissance de la biomasse .....	13
1.7.2 Taux de production de produit .....	15
1.7.3 Taux de dégradation du substrat .....	15
1.8 Modélisation des procédés .....	16
1.8.1 Conservation de la matière .....	16
1.8.2 Modèle de croissance de la biomasse .....	17
1.8.3 Modèle de consommation du substrat .....	17
1.8.4 Volume .....	18
1.8.5 Procédé discontinu type Batch .....	19
1.8.6 Procédé continu .....	19
1.8.7 Procédé discontinu type Fed-batch .....	20
1.8.8 Généralisation .....	21
1.9 Modes de conduite des bioréacteurs .....	21
1.9.1 Fermentation semi-continue (Fed-batch) .....	21
1.9.2 Fermentation continue .....	22
1.10 Régulation et automatisation des procédés .....	23
1.10.1 Régulateurs à action proportionnelle, intégrale et dérivée .....	23
1.10.2 Régulateurs neuronaux .....	24
1.11 Stratégies de commande d'un bioréacteur .....	24
1.12 Conclusion .....	24

## **Chapitre 2 : Théorie et apprentissage des réseaux de neurones**

2.1 Introduction .....	25
------------------------	----

2.2 Neurone biologique.....	26
2.3 Réseaux de neurones artificiels.....	27
2.3.1 Modèle formel d'un neurone.....	27
2.3.1.a Opérateur de sommation.....	29
2.3.1.b Fonction de seuil (ou d'activation) .....	29
2.4 Apprentissage des réseaux de neurones.....	30
2.4.1 Apprentissage supervisé.....	31
2.4.2 Apprentissage non supervisé.....	31
2.4.3 Surapprentissage.....	32
2.5 Mémoire associative.....	33
2.6 Architecture des réseaux de neurones artificiels.....	33
2.6.1 RNA à propagation directe.....	34
2.6.2 Réseaux de neurones bouclés (ou récurrents) .....	35
2.7 Algorithmes d'apprentissage.....	36
2.7.1 Algorithme de rétropropagation (Back-propagation BP) .....	36
2.7.2 Résumé de l'algorithme BP.....	37
2.7.3 Difficultés et limites de l'algorithme.....	40
2.7.4 Algorithme de Back- propagation avec momentum.....	41
2.7.5 Application de l'algorithme (BP).....	42
2.8 Optimisation des réseaux neuronaux.....	42
2.9 Réseaux de neurones et approximateurs universels .....	43
2.10 Conclusion.....	43

### **Chapitre 3 : Identification et commande par réseaux de neurones**

3.1 Introduction.....	44
3.2 Apprentissage en commande et commande adaptative .....	44
3.3 Quelques propriétés.....	45
3.4 Classification des méthodes de commande .....	46
3.4.1 Méthodes directes.....	46
3.4.1.1 Reproduction d'un contrôleur existant.....	46
3.4.1.2 Amélioration d'un système de commande linéaire.....	47
3.4.1.3 Identification directe d'un modèle inverse.....	48
3.4.1.4 Apprentissage direct par estimation de l'erreur.....	50
3.4.2 Méthodes indirectes.....	50
3.4.2.1 Identification des processus .....	50
3.4.2.2 Commande par modèle direct neuronal .....	52
3.4.3 Commande adaptative par réseaux de neurones.....	53
3.4.3.1 Régulateur auto-ajustable.....	53
3.4.3.2 Commande adaptative avec modèle de référence .....	55
3.5 Conclusion.....	57

### **Chapitre 4 : Commande d'un bioréacteur fed-batch et d'un bioréacteur continu**

4.1 Introduction.....	58
4.2 Description du problème de commande d'un bioréacteur et modélisation.....	58
4.3 Modèle d'un bioréacteur fed-batch.....	59
4.3.1 Comportement du système en boucle ouverte.....	60
4.4 Commande du bioréacteur en utilisant le régulateur classique.....	62
4.4.1 Description de la commande PID.....	62

---

4.4.2 Application d'un régulateur PI à un procédé semi- continu de fermentation.....	63
4.5 Commande neuronale du bioréacteur fed-batch.....	67
4.6 Commande d'un bioréacteur continu.....	73
4.6.1 Modèle d'un bioréacteur continu .....	73
4.6.2 Comportement du système en boucle ouverte.....	74
4.6.3 Commande du bioréacteur en utilisant le PI .....	76
4.6.4 Commande neuronale .....	78
4.7 Conclusion.....	81
<b>Conclusion générale.....</b>	<b>83</b>

## **Bibliographie**

# **Introduction générale**

# Introduction générale

---

---

La biotechnologie est un secteur en rapide croissance et les bioréacteurs constituent la principale application de cette technologie. Ils sont largement utilisés que ce soit dans l'industrie alimentaire ou pharmaceutique. Les bioréacteurs sont souvent fortement nonlinéaires, difficiles à commander et à modéliser. Ils constituent donc un domaine d'application très intéressant pour les techniques de commande modernes.

Pour l'analyse et la synthèse d'une loi de commande permettant de réaliser l'asservissement d'un processus, on doit passer le plus souvent par un modèle mathématique. Ce dernier peut être issu de la description mathématique d'un phénomène physique (modèle de connaissance) ou de l'expérience permettant de caractériser son comportement entrée-sortie (modèle d'observation). On peut noter qu'un modèle, aussi parfait soit-il, n'est jamais qu'une approximation de la réalité, et la commande désirée doit évidemment assurer des performances satisfaisantes en termes de stabilité, précision et rejet de perturbation dans tous les cas de figure.

Les réseaux de neurones artificiels (RNA) constituent un outil très efficace pour l'identification et la commande des systèmes nonlinéaires développé durant ces trois dernières décennies. Ils ont fait l'objet de nombreux travaux de recherche dans divers domaines scientifiques, donnant de très bons résultats. Les processus chimiques constituent un terrain d'application privilégié de ces techniques, car les problèmes rencontrés sont nombreux que ce soit dans leur modélisation, leur commande et même dans l'acquisition des données.

Des études ont été menées sur l'identification, la stabilité et la robustesse des réseaux de neurones, en identification et en commande des processus chimiques et les résultats obtenus sont satisfaisants.

L'objet de cette étude concerne la conception de lois de commande, pour la conduite de deux procédés de fermentation l'un semi continu (fed-batch) et l'autre continu. Le premier modèle a été extrait de l'article de M. Titica, D. Dochain et M. Guay [Tit03] où ils ont traité un problème d'optimisation. Le deuxième se trouve dans l'article de H.J.Tham, M.A.Hussain et K.B.Ramachandran [Tha00]. Ces derniers ont appliqué la commande par mode glissant

afin de contrôler la concentration du substrat dans le bioréacteur. Ces deux techniques de fermentation, sont très répandues dans la biotechnologie.

Dans cette optique, notre mémoire est divisé en quatre chapitres. Le premier chapitre est consacré à la description des phénomènes caractéristiques d'une fermentation à partir de laquelle dérivent la plupart des modèles de conduite. Les fermentations présentent des caractéristiques différentes à cause de la diversité des micro-organismes, des substrats utilisés et des produits formés. Après avoir décrit les différents modes de fonctionnement, les équations d'état des différents processus de fermentation sont présentées.

Le deuxième chapitre est consacré à la présentation des réseaux de neurones artificiels, certaines définitions de base et quelques notions sur la classification des RNA. On présentera aussi l'algorithme le plus utilisé pour l'apprentissage à savoir l'algorithme de rétropropagation du gradient (Gradient Descent Back-Propagation).

Le troisième chapitre nous présente les différentes méthodes directes et indirectes pour la commande et l'identification des systèmes par RNA, ainsi que les schémas correspondants.

Le quatrième chapitre expose nos résultats de simulation. En premier lieu, nous présentons une application sur un bioréacteur fed-batch, en utilisant la commande classique PI, puis en utilisant la commande neuronale. Ensuite, nous présentons une application sur un bioréacteur continu, en reprenant les mêmes lois de commande, ce qui permet de comparer les performances.

Nous terminons notre travail par une conclusion générale discutant les performances obtenues et donnant une perspective pour des travaux futurs.

# **Chapitre 1**

## **Fermentation et caractéristiques des bioréacteurs**

# Chapitre 1

## Fermentation et caractéristiques des bioréacteurs

---

### 1.1 Introduction

Les fermentations sont connues depuis la plus haute antiquité où elles étaient utilisées pour la fabrication des vins et toute sorte de liqueurs. La racine du mot fermentation provient du verbe latin ‘fermentare’ qui signifie transformer certaines denrées alimentaires à partir du substrat par le ferment.

Les dégagements gazeux accompagnant le plus souvent la réaction furent très longtemps considérés comme la caractéristique principale du phénomène. Plus tard, la notion même de fermentation fut étendue à d’autres réactions ne s’accompagnant pas nécessairement de dégagements gazeux, mais dont le caractère commun était d’obtenir à partir de la décomposition de matières premières des produits variés tels que l’acide lactique, l’éthanol...

Les fermentations alcooliques sont les plus anciennement étudiées. Elles correspondent à la transformation des sucres en alcool.

Les microorganismes responsables de ces réactions biologiques ne furent découverts qu’en 1680 par Loewenhock et ce n’est qu’au milieu de 19<sup>ème</sup> siècle qu’ils furent étudiés de manière scientifique par Pasteur, qui montra que ces transformations sont dues à des microorganismes divers qui utilisent pour leur survie et leur développement la matière et l’énergie produite par la dégradation de substance organique.

La recherche scientifique, par l’étude de la croissance microbienne, a permis la conception de modèles mathématiques nécessaires à la description de ce phénomène et facilitant la production à une échelle industrielle.

La fermentation industrielle utilise trois principaux types de microorganismes :

- Bactéries
- Champignons et moisissures

- Levures

L'étude de la fermentation se fait suivant trois approches :

- Une approche « microbiologique », qui consiste en la sélection des nutriments, la sélection des microorganismes, afin d'améliorer la production.
- Une approche « technologique » concernant le choix des bioréacteurs et de leurs modes de fonctionnement.
- Une approche « mathématique » dont les objectifs sont la modélisation, la surveillance et la commande du procédé de fermentation.

## 1.2 Bioprocédés

### 1.2.1 Généralités sur le processus de fermentation

Les objectifs de la fermentation sont les suivants :

- production de biomasse.
- production de produits de synthèse.
- dégradation de matières.

La fermentation est un procédé (figure 1.1) qui consiste, en général, à faire croître une population de micro-organismes (levures, bactéries, etc...) aux dépens de certains nutriments (substrats carbonés) à l'intérieur d'un bioréacteur, dans des conditions physico-chimiques (température, pH, aération, agitation, etc...) les plus favorables possibles [Que90].

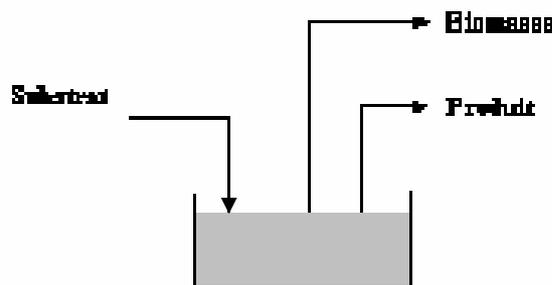


Figure 1.1 : Représentation simplifiée de la fermentation

### 1.2.2 Enzymes

Les processus de fermentation mettent en jeu de nombreuses réactions biochimiques cellulaires où intervient une classe particulière de protéines appelées enzymes, ce sont donc

des catalyseurs qui ont la propriété d'accélérer la vitesse de la réaction.

La théorie enzymatique fût développée par L. Michaelis et M. Menten, puis complétée par G. Briggs et J. Haldane. Elle postule que :

1. L'enzyme  $E$  forme avec le substrat  $S$  un complexe  $ES$  au cours d'une réaction réversible :



2. Ce complexe libère le produit  $P$ , ainsi que l'enzyme :



### 1.2.3 Facteurs de croissance microbienne

Il est difficile de tenir compte de l'ensemble des facteurs qui peuvent influencer sur les microorganismes.

Les microorganismes se différencient selon les conditions nécessaires à leurs croissances ainsi que leurs besoins en éléments nutritifs dont les principaux sont :

#### a- Eau

L'eau est indispensable aux microorganismes, car pour que la croissance de ceux-ci soit effective. Il faut qu'un seuil d'humidité relative, exprimé par l'activité de l'eau, soit atteint [Rae01].

#### b- Source d'énergie et carbone

Les microorganismes organotrophes, qui représentent la majorité des microorganismes utilisés industriellement, ont recours à la dégradation de composés organiques pour leurs besoins énergétiques, tels que l'amidon, sucres, etc.... Les microorganismes photosynthétiques et la plupart des chémolithotrophes sont capables d'utiliser directement le gaz carbonique de l'air pour leurs besoins énergétiques.

#### c- Azote

L'azote provient partiellement de substances organiques (protéines ou minérale).

#### d- Oxygène

L'oxygène, vital pour beaucoup de microorganismes, peut parfois se révéler toxique. En fait, il existe deux sortes de fermentation :

- Aérobie : où les microorganismes ont besoin d'oxygène pour leur développement.
- Anaérobie : le besoin en oxygène n'est pas nécessaire pour le développement (anaérobie facultatif), et peut être inhibiteur (anaérobie strict).

### **e- pH**

La croissance de microorganismes exige une valeur optimale de pH. La plupart des bactéries croissent en milieu neutre, alors que les champignons, levures et bactéries lactiques préfèrent un milieu acide.

### **f-Température**

Selon la température optimale de croissance, on distingue entre organismes:

Psychrophiles : de 5 à 20°C, mésophiles : de 20 à 45°C et les thermophiles : de 45 à 55°C.

## **1.3 Fermenteur**

Le fermenteur ou bioréacteur constitue le cœur d'une installation de fermentation. Il doit permettre d'établir les conditions environnementales (température, pH, oxygène, nutriments limitants, etc...) optimales pour l'obtention d'une croissance rapide, ainsi qu'une formation efficace de produits. La conception du fermenteur vise à optimiser les transferts de matière et de chaleur. Les exigences principales concernant les bioréacteurs sont [Rae01] :

- Mélange homogène des fluides
- Dissipation homogène de l'énergie
- Forces de cisaillement adéquates (ailettes de mélangeur).
- Turbulence suffisante pour le transfert de chaleur
- Bonne dispersion de l'air
- Durée de mélange optimale
- Pas de séparation des constituants du milieu
- Minimisation de la formation de mousse
- Hydrodynamique bien définie, absence de zones mortes
- Possibilité de stériliser et de maintenir les conditions de stérilité
- Isolation du contenu
- Durabilité
- Possibilité de mesurer les variables du procédé, ainsi que ses éléments-clés
- Manipulation pratique, entretien avantageux

## 1.4 Modes de fonctionnement

Les différents modes de fonctionnement représentent les différentes façons d'alimenter le bioréacteur en substrat frais. On dénombre trois principaux modes opératoires [Que90], [Duv88] :

### 1.4.1 Fermentation en mode discontinu (Batch)

Le substrat nécessaire à la formation d'une certaine quantité de produit est apporté en début de manipulation dans un réacteur préalablement rempli et ensemencé. Le fermenteur est ensemencé par une concentration en biomasse initiale  $X_0$  dans un milieu de concentration en substrat  $S_0$  et de volume  $V$ . La fermentation se déroule jusqu'à l'épuisement du substrat (figure 1.2).

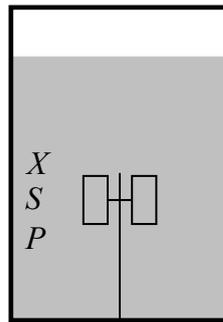


Figure 1.2 : Schéma d'un procédé discontinu (batch)

$P$  : Concentration en produit

### 1.4.2 Fermentation continue

Le fermenteur est soumis à un apport de substrat et de milieu frais et à un débit de soutirage (figure 1.3). Le flux d'alimentation et le flux de soutirage sont égaux à  $Q$ . Après une période transitoire qui dépend des conditions opératoires, le processus se stabilise en régime permanent.

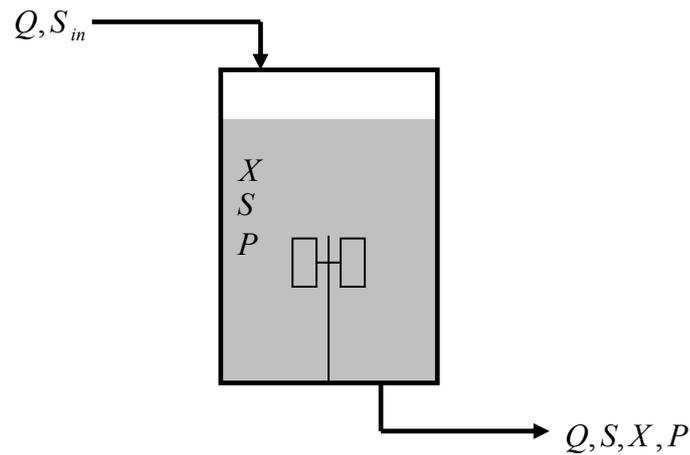


Figure 1.3 : Schéma d'un procédé continu

$S_{in}$  : Concentration du substrat d'alimentation

Depuis les travaux de Monod en 1949 et l'établissement expérimental de son équation, on sait pratiquer la culture des microorganismes en continu, en alimentant de façon ininterrompue le fermenteur avec un milieu nutritif, tout en soutirant en continu un volume égal du mélange biomasse-milieu liquide, de façon à maintenir le volume du bioréacteur constant.

Un tel dispositif permet, contrairement à la fermentation discontinue, de maintenir constant et à une valeur voulue.

- La concentration cellulaire.
- Le taux de croissance de la biomasse
- Les équilibres nutritifs du milieu
- La production en produit recherché

Deux dispositifs expérimentaux permettent de mettre en œuvre la fermentation continue :

Le chemostat, et le turbidostat. Dans le premier cas, le fermenteur est alimenté en continu avec un milieu nutritif complet contenant tous les composants en excès par rapport aux besoins des microorganismes sauf un, dit facteur limitant, qui va contrôler la quantité de biomasse à l'état d'équilibre stationnaire. Dans le deuxième cas, l'alimentation est asservie à la mesure de la concentration en biomasse par turbidimétrie (mesures optiques de la turbidité).

### 1.4.3 Fermentation semi-continue (Fed-batch)

Un procédé semi-continu peut être schématisé par la (figure 1.4). Le fermenteur est préalablementensemencé par une concentration en biomasse initiale  $X_0$  dans un milieu de concentration en sucre  $S_0$  et de volume  $V_0$ , il est alimenté ensuite en substrat de concentration  $S_{in}$  avec un débit d'entrée  $Q$ .

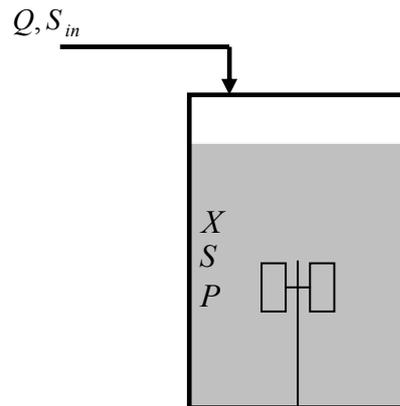


Figure 1.4 : Schéma d'un procédé semi- continu (Fed-batch)

Rien n'est soutiré au cours de la fermentation de sorte que le volume dans le fermenteur est variable. La fermentation se termine lorsque le volume obtenu est maximal.

### 1.4.4 Les avantages d'un procédé semi-continu de fermentation

Parmi les différents modes présentés, on trouve les procédés semi-continus de fermentation.

Cette technique est utilisée dans bon nombre de fermentations, le fait de pouvoir contrôler à tout instant la concentration en sucre dans le milieu de fermentation, constitue le principal avantage d'une fermentation semi-continue, car le rendement de la réaction est optimisé. Ce procédé permet d'éviter l'effet inhibiteur de concentration élevée de substrat. Comparons les temps de production d'éthanol des fermentations discontinu (Batch) et en semi continu (Fed-batch) [Que90].

Quantité d'Alcool Produit (G)	Temps (h)		Gain De Temps (%)
	Batch	Fed -Batch	
800	23.7	22.8	3.8
1000	26.4	24.9	5.7
1500	34.1	31.3	8.2

Table 1.1 : Comparaison des temps de production d'une quantité donnée d'alcool en Batch et en Fed-batch [Que90]

Nous constatons que le temps de production d'alcool est effectivement important dans le mode batch que dans le mode fed-batch.

### 1.5 Dynamique cellulaire

La liste des procédés de fermentations les plus importants exploités industriellement est [Bou] :

- La fermentation alcoolique, utilise les microorganismes suivants :  
Levures, bactéries zymomonas et bactéries zymosarcina, elle sert à la production d'alcool industriel.
- La fermentation homolactique, qui utilise les microorganismes suivants :  
Streptococcus et lactobacillus. Ce type de fermentation est très utilisable en industrie agroalimentaire.
- La fermentation acétonobutylique, ce type de fermentation est utilisable pour la Production d'acétate, CO<sub>2</sub>, butérate.

La croissance peut être vue sous deux angles [Bou], [Bur00], [Scr00], [Sim70] :

**Chimique** : La croissance est considérée comme un ensemble de réactions chimiques en chaîne. Le processus obéit au principe de conservation de la matière.

**Biologique** : L'augmentation de la population microbienne au cours du phénomène de croissance résulte de la reproduction qui peut se faire selon divers mode, selon le type de microorganisme mis en jeu (bactéries, levure).

Les levures se reproduisent par bourgeonnement : un gonflement apparaît à la surface d'une cellule, appelée alors cellule mère. Lorsque le bourgeon a pratiquement la même taille que la cellule mère, il s'en sépare pour devenir lui-même une nouvelle cellule.

L'étude de la croissance n'est valable que si l'on dispose de techniques appropriées d'évaluation. C'est pourquoi l'instrumentation joue un rôle important en permettant la mise au point de capteurs capables de travailler en temps réel en effet l'un des problèmes rencontrés dans cette industrie est qu'il faut d'abord prélever des échantillons puis les analyser, ce qui nécessite un temps relativement long et qui a comme conséquence l'altération des résultats.

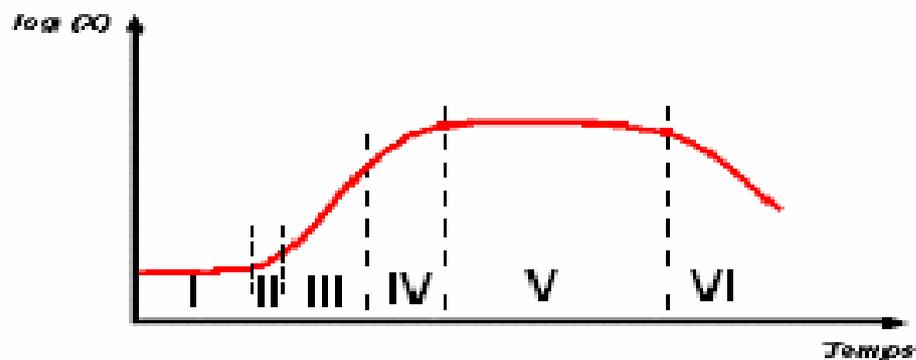


Figure 1.5 : Courbe de croissance microbienne

On distingue 6 phases successives dans la croissance de la biomasse (figure 1.5) définies par Buchanan (1918) :

- I** Phase de latence
- II** Phase d'accélération de croissance
- III** Phase de croissance exponentielle
- IV** Phase de décélération de croissance
- V** Phase stationnaire
- VI** Phase de déclin

- Les phases **I** et **II** correspondent à la phase d'adaptation au milieu. Le temps de latence est en fait une mesure de l'influence du milieu de culture sur la croissance.

- Dans la phase **III**, l'augmentation de la concentration en biomasse  $X$  est proportionnelle au temps et à la concentration en bactéries présentées :

$$dX = \mu X dt \quad (1.3)$$

$\mu$  est le taux de croissance. Pendant cette phase, la croissance est assurée par la consommation du substrat  $S$ , et le rendement de croissance est exprimé par :

$$D = \left| \frac{dX}{dS} \right| \quad (1.4)$$

- Les phases **IV**, **V**, **VI** correspondent à un ralentissement de croissance de la concentration de la biomasse.

## 1.6 Modélisation des cinétiques de réaction

### 1.6.1 Vitesses de réaction

Comme on l'a déjà dit, les levures se reproduisent par bourgeonnement, un gonflement apparaît à la surface d'une cellule, appelée alors cellule mère. Lorsque le bourgeon a pratiquement la même taille que la cellule mère, il s'en sépare pour devenir lui-même une nouvelle cellule. L'accroissement de population est proportionnel à la population elle-même, c'est à dire que la vitesse de croissance de la masse cellulaire  $r_x$  est proportionnelle à la concentration en microorganisme  $X$  à chaque instant, ce que l'on peut expliciter par la formule suivante [Que90], [Duv88] :

$$r_x = \mu X \quad (g \ l^{-1} \ h^{-1}) \quad (1.5)$$

On désigne par  $\mu$  le taux de croissance, On définit de la même manière respectivement  $v_s$  le taux de dégradation (le substrat consommé est utilisé pour la synthèse cellulaire, pour le maintien de la biomasse dans un état viable) et  $v_p$  le taux de production (production et croissance sont fortement corrélés). Ces trois paramètres représentent les vitesses spécifiques de la réaction relatives à la biomasse. On peut définir la vitesse de dégradation du substrat  $r_s$  et la vitesse de production de produit  $r_p$  par les équations suivantes :

$$r_s = v_s X \quad (1.6)$$

$$r_p = v_p X$$

## 1.7 Modélisation des vitesses spécifiques

### 1.7.1 Taux de croissance de la biomasse

La croissance microbienne dépend d'un grand nombre de facteurs physiques et chimiques (température, pH, agitation...) mais la prise en compte de toutes ces variables, dites d'environnement compliquerait le modèle. Pour cela la plupart des modèles empiriques ne dépendent que des concentrations de sucre, de la biomasse et produit formé.

Les modèles sont construits à partir des considérations sur la dynamique cellulaire. L'influence de la teneur en substrat sur  $\mu$  est schématisé sur la (figure 1.6) où l'on peut discerner trois phase [Que90], [Duv88]:

- A : limitation par manque de substrat.
- B : maximum du taux de croissance  $\mu$ .
- C : inhibition par excès de substrat.

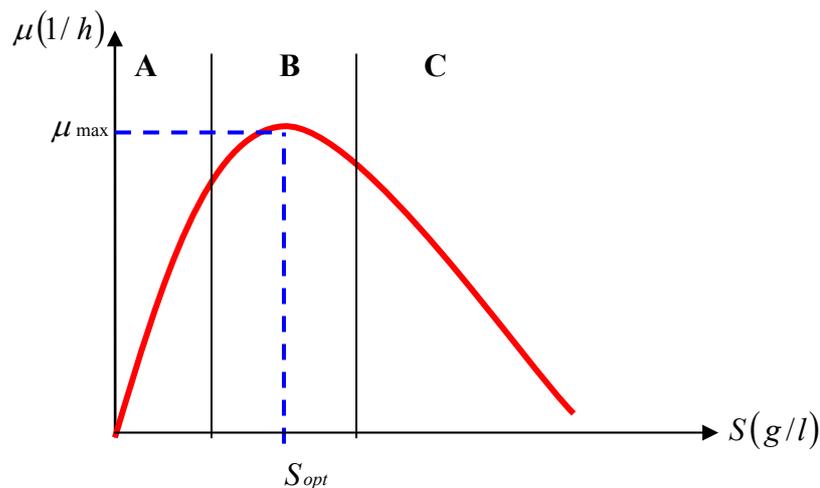


Figure 1.6 : Influence du substrat sur la croissance

Le modèle de Monod [Mon42], très largement répandu dans la littérature, décrit les deux premières phases de la croissance (figure 1.7).

$$\mu(S) = \mu_{max} \frac{S}{K_S + S} \quad (1.8)$$

Où  $\mu_{max}$  représente le taux de croissance maximum et  $K_S$  (constante de Monod) ou (coefficient de demi saturation) une concentration en substrat telle que le taux de croissance

soit demi-maximal (figure 1.7). Ce modèle est couramment utilisé dans tous les secteurs de la microbiologie. Plus précisément, 1800 publications scientifiques font explicitement référence aujourd'hui aux travaux de Monod.

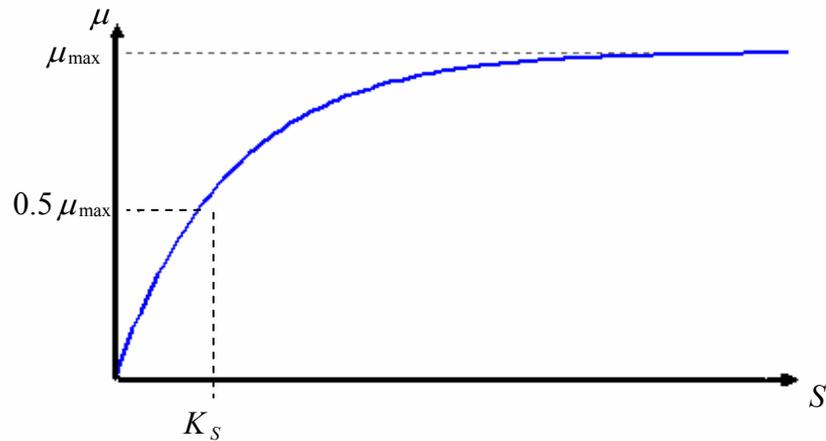


Figure 1.7 : Traduction graphique de l'équation de Monod

La représentation de Monod, est simple puisqu'elle ne fait intervenir que deux paramètres. Cependant il s'avère que cette expression n'est pas suffisante notamment pour la prise en compte de l'inhibition du substrat. Le modèle de Haldane fait intervenir l'inhibition due à l'excès de substrat (figure 1.6), son expression mathématique est :

$$\mu(S) = \mu_0 \frac{S}{\left( K_s + S + \frac{S^2}{K_I} \right)} \quad (1.9)$$

Avec :  $K_I$  constante d'inhibition de croissance bactérienne dû à la forte concentration du substrat.

$$\mu_0 = \mu_{\max} \left( 1 + 2 \sqrt{\frac{K_s}{K_I}} \right) \quad (1.10)$$

D'après la formule (1.9), on constate que le taux de croissance est optimal si  $S = S_{opt}$ .

$$S_{opt} = \sqrt{K_s K_I} \quad (1.11)$$

L'influence des substances produites (taux de production) sur la croissance de la biomasse est traduite par l'adjonction d'un terme multiplicatif aux précédentes expressions qui est symbolisé par  $\mu(P)$ . Dans la plupart des cas, les effets inhibiteurs du produit sont décrits par des relations linéaires, hyperboliques ou exponentielles. Le choix de l'une ou de l'autre de ces relations ne dépend que de la levure utilisée.

$$\text{-linéaire } \mu(P) = 1 - a P \quad [\text{Hin46}] \quad (1.12)$$

$$\text{-exponentiel } \mu(P) = e^{-K P} \quad [\text{Aib68}] \quad (1.13)$$

$$\text{-hyperbolique } \mu(P) = \frac{K_P}{P + K_P} \quad [\text{Ier69}] \quad (1.14)$$

L'association des effets du substrat et du produit nous amène à exprimer le taux spécifique de croissance par l'expression :

$$\mu = \mu(S) \mu(P) \quad (1.15)$$

### 1.7.2 Taux de production de produit

En considérant, d'une part, que production et croissance de la biomasse sont fortement corrélés pendant la première phase de la fermentation, que, d'autre part, les cellules sont proliférantes et produisent de l'alcool en fin de fermentation. Luedeking et Piret expriment le taux de production comme une fonction linéaire du taux de croissance.

$v_P = \alpha \mu + \beta$  ( $\alpha$  et  $\beta$  sont des constantes), cependant les modèles proposés plus récemment dissocient généralement croissance et production. Il traduisent cependant la corrélation croissance-production par une similitude des expressions de  $\mu$  et de  $v_P$ . Citons à ce sujet les modèles proposés par Ghose et Tyagi [Gho79].

### 1.7.3 Taux de dégradation du substrat

Le taux de dégradation est relatif à la quantité de substrat consommé, utilisée pour la production de l'éthanol et pour le maintien de la biomasse dans un état viable. On exprime alors le taux de dégradation par :

$$v_s = \frac{\mu}{y_X} + \frac{v_P}{y_P} + m \quad (1.16)$$

$y_X$  : Rendement de conversion du substrat en biomasse.

$y_p$  : Rendement de conversion du substrat en produit.

$m$  : Coefficient de maintenance, n'est généralement pas pris en considération dans les modèles, pouvant être négligé devant les 2 premiers termes de l'expression de  $v_s$ .

## 1.8 Modélisation des procédés

L'établissement du modèle mathématique d'un processus biotechnologique comprend deux phases bien distinctes [Duv88], [Que90], [Scr00] :

La première concerne la détermination des diverses vitesses de réaction qui rendent compte de la façon dont la biomasse se développe, le substrat se dégrade et le produit se fabrique.

La deuxième étape permet de déterminer les équations d'état du système. Les variables d'état considérées ici sont les concentrations de biomasse  $X$ , de substrat  $S$ , de produit  $P$  et le volume  $V$ .

Quel que soit le mode de fonctionnement des fermentations, la modélisation est faite à partir des équations établissant le bilan de matière relatif à la biomasse, au substrat et au produit, en écrivant que la variation des quantités correspondantes pendant  $dt$  est égale à la somme de ce qui est apporté et créé, diminuée de ce qui est dégradé et soutiré.

Nous allons donc établir, dans ce qui suit, les équations de fonctionnement d'un réacteur à alimentation et soutirage permanent par la suite celle d'un procédé semi-continu.

### 1.8.1 Conservation de la matière

D'après la loi fondamentale de Lavoisier sur la conservation de la matière, rien ne se crée et rien ne se perd, mais tout se transforme. Ainsi la masse exprimée en gramme se conserve en dépit des transformations [Scr00], [Sim70], [Lie98].

La variation de la quantité de masse dans le bioréacteur est résumée comme suit

$$\frac{dM}{dt} = M_e - M_s + R_g - R_c \quad (1.17)$$

$M$  : Quantité de matière relative à la biomasse et au substrat (g).

Relatif à la biomasse :

$$M = V \cdot X \quad (1.18)$$

Relatif au substrat :

$$M = V \cdot S \quad (1.19)$$

$M_e(g)$  : quantité de matière entrante ‘alimentation’.

$M_s(g)$  : quantité de matière sortante ‘soutirage’.

$R_g(g)$  : quantité de matière produite au cours de la réaction.

$R_c(g)$  : quantité de matière consommée au cours de la réaction.

Si nous nous limitons à  $X, S$  et nous ne prenons pas en compte  $P$  nous pourrions obtenir pour un fermenteur de volume variable  $V$  les équations suivantes :

### 1.8.2 Modèle de croissance de la biomasse

$$\frac{d[VX]}{dt} = \text{quantité de biomasse produite} - \text{quantité de biomasse soutirée}$$

D’après (1.3), la quantité de biomasse produite est égale à  $\mu VX$ .

$$\frac{d[VX]}{dt} = \mu VX - Q_s X \quad (1.20)$$

### 1.8.3 Modèle de consommation du substrat

$$\frac{d[VS]}{dt} = - \text{quantité de substrat consommée} - \text{quantité de substrat soutirée} + \text{quantité ajoutée de substrat (alimentation)}$$

On suppose que la quantité de substrat consommée est proportionnelle à la quantité de biomasse créée [Duv88], [Scr00].

$$\text{Soit } D \text{ le rendement de conversion } D = \left| \frac{dX}{dS} \right| = \left| \frac{\mu XV}{dS} \right|, \text{ alors } dS = -\frac{\mu XV}{D}.$$

$$\frac{d[VS]}{dt} = -\frac{\mu}{D} XV - Q_s S + Q_E S_{in} \quad (1.21)$$

### 1.8.4 Volume

$$\frac{dV}{dt} = \text{débit d'alimentation en substrat} - \text{débit de soutirage}$$

$$\frac{dV}{dt} = Q_E - Q_S \quad (1.22)$$

Si nous nous limitons à  $X, S$  et nous ne prenons pas en compte  $P$ , nous avons pour un fermenteur de volume variable  $V$ , le modèle suivant : [Que90], [Duv88], [Scr00]

$$\left\{ \begin{array}{l} \frac{d[VX]}{dt} = \mu VX - Q_S X \\ \frac{d[VS]}{dt} = -\frac{\mu}{D} X V - Q_S S + Q_E S_{in} \\ \frac{dV}{dt} = Q_E - Q_S \end{array} \right. \quad (1.23)$$

$Q_E$  : débit d'alimentation en substrat (1/h).

$Q_S$  : débit de soutirage (1/h).

$V$  : volume du mélange (biomasse+substrat) dans le bioréacteur (l).

$S$  : concentration en substrat (g/l).

$X$  : concentration de la biomasse (g/l)

$S_{in}$  : concentration de substrat à l'entrée de fermenteur (g/l).

$\mu$  : taux de croissance ( $h^{-1}$ )

$D$  : rendement de transformation ( Substrat  $\longrightarrow$  Biomasse)

$$D = \left| \frac{dX}{dS} \right| \quad (1.24)$$

La concentration de gaz  $y$  due à la réaction de fermentation est toujours proportionnelle à la quantité de biomasse produite, cette grandeur sera notre sortie [Que90], [Duv88], [Scr00].

$$y = k_2 \mu X \quad (1.25)$$

Le coefficient  $k_2$  étant évidemment positif.

### 1.8.5 Procédé discontinu type batch

Dans ce type des procédés de fermentation les débits d'entrée et de sortie sont nuls, le volume dans le bioréacteur reste constant,  $\frac{dV}{dt} = 0$ .

Le modèle (1.23) devient :

$$\left\{ \begin{array}{l} \frac{dX}{dt} = \mu X \\ \frac{dS}{dt} = -\frac{\mu}{D} X \\ y = k_2 \mu X \end{array} \right. \quad (1.26)$$

### 1.8.6 Procédé continu

La première étape consiste à ensemercer le fermenteur renfermant un volume du milieu qui reste constant. Une fois que la phase de croissance exponentielle est obtenue, notre bioréacteur est maintenu à l'état d'équilibre. Il sera décrit par les équations (1.27), qui sont obtenues par considération sur le bilan de masse et en supposant que la croissance de la biomasse est dans une phase exponentielle.

$$\left\{ \begin{array}{l} \frac{dX}{dt} = -\frac{Q}{V} X + \mu X \\ \frac{dS}{dt} = -\frac{\mu}{D} X + \frac{Q}{V} S_{in} - \frac{Q}{V} S \\ \frac{dV}{dt} = 0 \\ y = k_2 \mu X \end{array} \right. \quad (1.27)$$

Dans ce type de fonctionnement, les cellules sont approvisionnées en substrat de façon continue, tout en soutirant en continu un volume égal du mélange biomasse milieu liquide, de façon à maintenir le volume dans le bioréacteur constant. La croissance de la biomasse sera, de ce fait, toujours dans sa phase exponentielle [Bou], [Dan89], [Duv88], [Scr00].

**Remarque 1 :** Ce système d'équations est non linéaire sur l'état et linéaire sur la commande. Il s'exprime sous la forme :

$$\dot{x} = f(x) + g(x)u \quad \text{Avec :}$$

$x$  : vecteur d'état

$u$  : commande de système  $u = \frac{Q}{V}$

$f$  et  $g$  : fonctions nonlinéaires de l'état

### 1.8.7 Procédé discontinu type Fed-batch

La différence entre le mode continu et le mode fed-batch est qu'il n'y a pas de soutirage. Dans ce cas ( $Q_s = 0$ ), le volume est donc variable. Ce type de fonctionnement est similaire au fonctionnement en discontinu (batch), à la différence qu'une alimentation en substrat est présente. Ce qui rend le volume variable en fonction du débit d'alimentation [Bou], [Dan89], [Duv88], [Scr00].

En utilisant les équations (1.23), on aura le résultat suivant :

$$\left\{ \begin{array}{l} \frac{dX}{dt} = \mu X - \frac{Q}{V} X \\ \frac{dS}{dt} = -\frac{\mu}{D} X + \frac{Q}{V} S_{in} - \frac{Q}{V} S \\ \frac{dV}{dt} = Q \\ y = k_2 \mu X \end{array} \right. \quad (1.28)$$

La fermentation commence dans un petit volume initial du milieu de culture. Le débit d'alimentation ( $Q_E = Q$ ) est réglé de façon à ce que la production en biomasse soit optimale [Duv88].

### 1.8.8 Généralisation

En partant des équations (1.23), nous aboutissons aux équations unifiées qui peuvent être appliquées au trois modes déjà étudiés (1.29) :

$$\left\{ \begin{array}{l} \frac{dX}{dt} = \mu X - \frac{Q_E}{V} X - \frac{X}{V} (Q_E - Q_S) \\ \frac{dS}{dt} = -\frac{\mu}{D} X + \frac{Q_E}{V} S_{in} - \frac{Q_S}{V} S - \frac{S}{V} (Q_E - Q_S) \\ \frac{dV}{dt} = Q_E - Q_S \\ y = k_2 \mu X \end{array} \right. \quad (1.29)$$

Où :

- $Q_S$ : le débit sortant du bioréacteur « soutirage ».
- $Q_E$ : le débit entrant « alimentation ».
- Si  $Q_E = Q_S = 0$ , on se place dans le mode discontinu.
- Si  $Q_E = Q_S = Q$ , on se place dans le mode « continu ».
- Si  $Q_S = 0$ , on se place dans le mode Fed-batch

On notera, par  $u$  le taux de dilution qui est égal à  $\frac{Q}{V}$ .

$$u = \frac{Q}{V} \quad (1.30)$$

## 1.9 Modes de conduite des bioréacteurs

### 1.9.1 Fermentation semi-continue (Fed-batch)

La fermentation commence dans un petit volume de milieu de culture. La concentration initiale et le volume de milieu sont faibles. La fermentation démarre sans ajout de substrat, ce qui correspond à la fermentation discontinue. Lorsque la biomasse est en phase exponentielle de croissance, on introduit dans la cuve le substrat stérile. Le débit d'alimentation est réglé de

façon, à ce que la concentration en substrat atteigne une valeur optimale, et corresponde à une étape de la phase exponentielle de croissance cellulaire. Si le débit est trop faible (figure 1.8), la concentration cellulaire augmente selon une courbe plus ou moins proche de celle correspondant à l'évolution de la croissance en discontinu. Si au contraire le débit d'alimentation est trop grand, la concentration en biomasse a tendance à diminuer (figure 1.8). Dans ces conditions, le volume du milieu de fermentation augmente rapidement, et la quantité totale de biomasse  $VX$  plus faiblement car l'augmentation de volume  $V$  est plus importante que la diminution de la concentration en biomasse  $X$  [Scr00].

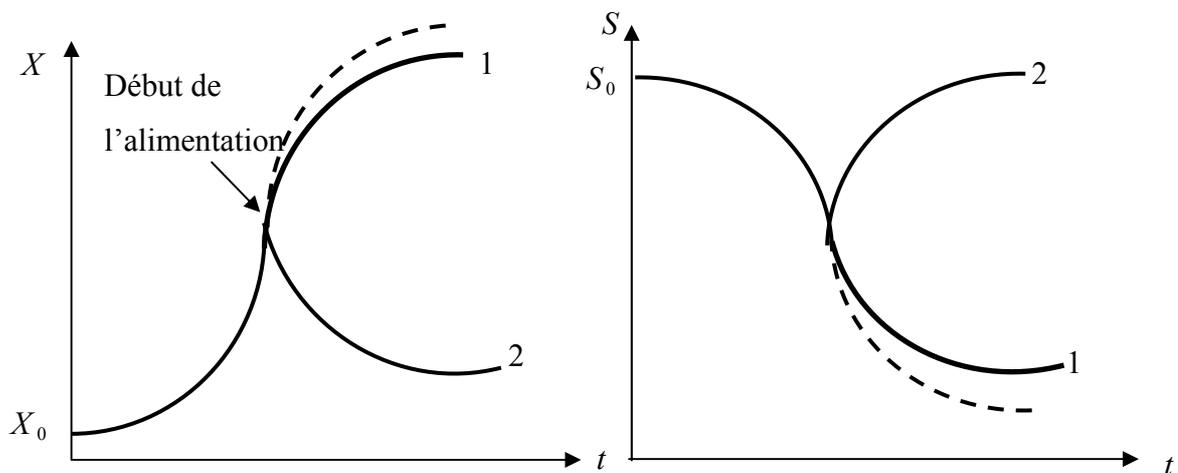


Figure 1.8 : Evolution des concentrations en biomasse et substrat dans le fermenteur  
 (1) débit d'alimentation faible  
 (2) débit d'alimentation élevé

### 1.9.2 Fermentation continue

On peut maintenir un état d'équilibre dans une cuve de fermentation alimentée et soutirée en continu. Ce mode de fonctionnement ne se conçoit pas sans une phase de croissance discontinue préalable à l'alimentation et au soutirage en continu. Il faut donc commencer par ensemençer dans une cuve de volume  $V$  de substrat stérile. L'alimentation et le soutirage aux mêmes débits commencent quand la concentration cellulaire  $X$  désirée est atteinte. Il lui correspond des concentrations  $S$  en substrat et  $P$  en produit formé constantes. Pour maintenir ces concentrations, il faut que le débit, soit suffisant pour apporter assez de substrat et maintenir la population microbienne dans un état physiologique éloigné de la phase stationnaire (c'est le chemostat).

D'après les équations (1.27) l'équilibre est atteint quand le taux de dilution est égal au taux de croissance ( $\mu = u$ ).

Si à partir de cet état d'équilibre, on augmente le débit, le volume restant constant, on aura donc une augmentation du taux de dilution, on a alors  $u > \mu$ . En conséquence, la concentration cellulaire dans la cuve tend à diminuer, ce qui entraîne une augmentation de la concentration en substrat, mais si  $S$  augmente, cela entraîne l'augmentation de  $\mu$  conformément à la relation de Monod. Le système va donc se stabiliser dans un nouvel état d'équilibre où  $\mu = u$  caractérisé par de nouvelles concentrations en biomasse, en substrat et en produit.

Si au contraire, toujours à partir de l'état d'équilibre, le taux de dilution diminue du fait d'une diminution du débit, alors  $u < \mu$ , la concentration cellulaire augmente ce qui provoque la diminution de la concentration en substrat et on aboutit à un nouvel état d'équilibre où  $\mu = u$ . En conclusion, on voit que dans la fermentation continue, on a pour chaque valeur de  $u$  une position d'équilibre stable correspondant à  $u = \mu$ . On cherchera donc à ajuster  $u$  de manière à avoir  $u = \mu_{opt}$ . Ces positions d'équilibre ne peuvent exister que si  $u < \mu_{max}$ . Si  $u > \mu_{max}$  le seul point d'équilibre possible est  $S = S_0$  et  $X = 0$ , cela correspond à une quantité totale de biomasse nulle puisque  $V$  est constant [Scr00].

## 1.10 Régulation et automatisation des procédés

Réguler c'est maintenir, pour un paramètre donné, un écart significatif minimal, et si possible nul, entre sa valeur mesurée (à l'aide d'un capteur) et celle désirée également appelée consigne. En fonction du type d'action qu'ils permettent, on distingue différents types de régulateurs [Scr00] :

### 1.10.1 Régulateur à action proportionnelle, intégrale et dérivée

Ils permettent d'obtenir rapidement une régulation très fine. Il permet de rassembler les principales exigences d'une bonne régulation : rapidité, précision, et stabilité. Ce genre de régulateur est le plus utilisé dans l'industrie.

### 1.10.2 Régulateurs neuronaux

Malgré l'existence de régulateurs performant dans le domaine des biotechnologies, la complexité du système est toujours un sérieux problème pour les industriels, car on veut une maîtrise toujours plus grande des procédés. Les réseaux de neurones sont très prometteurs à cause de leur aptitude à prendre en charge des systèmes fortement non linéaires comme c'est le cas des fermenteurs. Mais, ils sont difficiles à implémenter à cause de la complexité des calculs nécessaires à la mise au point du régulateur.

### 1.11 Stratégies de commande d'un bioréacteur

Les bioréacteurs nécessitent un réglage précis et rapide des paramètres car ces derniers sont sujets à beaucoup de variations au cours du temps. Ce qui rend leur régulation plus complexe que celui d'un réacteur chimique conventionnel.

Avant la réalisation d'une commande de bioréacteur, l'automaticien doit connaître les principales caractéristiques de son bioréacteur :

- Les meilleures conditions de culture (pH, agitation, température, etc...).
- Les différentes phases de croissance.
- Les éventuelles interactions entre les différents composants du milieu de culture

Ainsi, on pourra définir les objectifs à atteindre et les paramètres à optimiser.

### 1.12 Conclusion

L'objet de ce chapitre est de donner une idée générale sur les biotechnologies et en particulier sur les bioréacteurs. Nous définissons les différents modes de leurs fonctionnements et nous donnons leurs modèles mathématiques. Notre procédé de fermentation est décrit par un ensemble d'équations dynamiques où la commande utilisée sera le débit d'alimentation. Ensuite, nous donnons un bref exposé sur les techniques de commande utilisées afin d'améliorer la productivité et la rentabilité des procédés de fermentation.

# **Chapitre 2**

**Théorie et apprentissage des réseaux  
de neurones**

# Chapitre 2

## Théorie et apprentissage des réseaux de neurones

---

### 2.1 Introduction

Les réseaux de neurones artificiels, ou réseaux neuromémitiques présentent une alternative puissante à tout algorithme de calcul classique car ils ont la capacité de résoudre des problèmes mathématiques non linéaires. Ils sont constitués de cellules de calculs identiques appelées neurones formels, qui fonctionnent en parallèle. La mémorisation des informations est faite dans les poids des connexions de chaque neurone. Les RNA sont aussi caractérisés par leurs capacités d'apprentissage et de généralisation. Ces caractéristiques poussent les chercheurs de toutes les disciplines à étudier les RNA et à les appliquer dans leurs domaines de recherche.

L'étude des réseaux de neurones date des années 40. Le premier modèle de neurone est dû aux deux neurobiologistes américains, Warren McCulloch et Walter Pitts qui supposèrent que l'impulsion nerveuse était le résultat d'un calcul simple effectué par chaque neurone et c'est grâce à la mise en commun des neurones dans le réseau que naît la pensée. Au même moment Donald Hebb, psychologue canadien proposait une hypothèse qui associait l'activité des neurones à l'efficacité des synapses qui les relient [Jod 94].

Les travaux sur les réseaux de neurones formels ont connu des débuts prometteurs vers la fin des années 50, mais un manque de base théorique, et la concurrence de l'intelligence artificielle, ont conduit à une période de latence dans ce domaine. Il faut attendre la fin des années 70/milieu des années 80 pour que la recherche dans ce domaine redémarre, c'est en fait la période où l'approche « classique » de l'intelligence artificielle basée sur la logique binaire a montrée ses limites. Le physicien John Hopfield est l'un des chefs de file de cette reprise vers la fin des années 80. On peut dire que les réseaux de neurones sont devenus un

domaine de recherche attractif, dans différentes disciplines : Psychologie, biologie, informatique, automatique.

Dans ce chapitre, nous présentons brièvement quelques notions générales sur le neurone biologique, ensuite les caractéristiques du neurone artificiel, puis l'algorithme d'apprentissage pour un RNA.

## 2.2 Neurone biologique

Les neurones sont les cellules de base du système nerveux. Ces cellules sont responsables de la réception et de la transmission des influx nerveux (impulsions électriques), et forment donc un réseau. Un neurone est composé d'un corps cellulaire qui contient un noyau, d'un axone et d'un ou plusieurs dendrites (figure 2.1). Les dendrites sont les parties multi-ramifiées qui reçoivent les influx nerveux. Les axones sont les structures allongées qui transmettent les influx à partir du corps cellulaire. Les neurones transmettent leurs influx à travers des connexions nerveuses appelées synapses par l'intermédiaire d'un neuromédiateur (neurotransmetteur) chimique tel que l'acétylcholine. Cette substance chimique est délivrée par de minuscules vésicules qui se trouvent à l'extrémité de l'axone, quand l'influx nerveux les atteint, les neurotransmetteurs se diffusent alors dans la fente synaptique et excitent les dendrites de la cellule suivante pour créer un nouvel influx nerveux dans cette cellule. Le mécanisme se reproduit tout au long de la chaîne de neurones et l'influx nerveux se propage uniquement dans une seule direction.

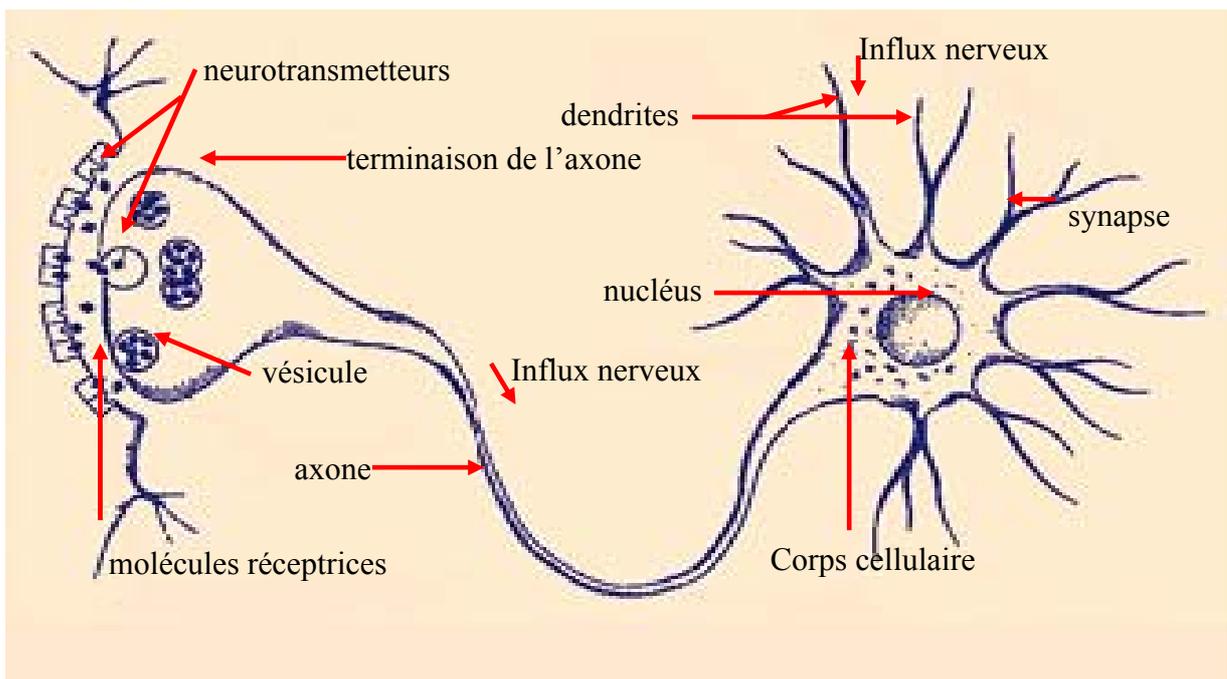


Figure 2.1 : Neurone biologique

Si l'on observe le sens de circulation de l'influx nerveux, on constate que l'information au niveau de chaque neurone est soumise à deux unités de traitements :

-Premièrement, la synapse où le signal transmis par les arborisations terminales des axones subit une modulation. Cette modulation est réalisée par le fait que l'excitation du neurone récepteur n'est effectuée que si la quantité du neurotransmetteur qui se fixe sur l'autre coté de l'espace synaptique (membrane post synaptique) dépasse un certain seuil. La synapse obéit donc à la loi du tout ou rien en réalisant une fonction seuil.

-Deuxièmement, le corps cellulaire effectue une sommation de tous les signaux reçus depuis les synapses via les dendrites.

### 2.3 Réseaux de neurones artificiels

Les réseaux de neurones artificiels sont des modèles informatiques inspirés de la structure et du comportement des neurones biologiques. Ils sont composés d'unités interconnectées que l'on appelle neurones formels ou artificiels, recevant des informations provenant du milieu extérieur ou d'autres neurones. Les neurones sont connectés entre eux au moyen de synapses dont les poids modulent les informations transmises à d'autres neurones ou vers la sortie. Ces poids sont déterminés par apprentissage.

#### 2.3.1 Modèle formel d'un neurone

C'est le résultat de l'analogie directe qu'ont faite MacCulloch et Pitts, en 1943, avec le neurone biologique. Ce fut la première modélisation mathématique d'un neurone. Il s'agit d'un corps cellulaire qui exécute une somme pondérée des signaux d'entrée qui lui parviennent (figure 2.2). Si cette somme dépasse un certain seuil, le neurone est activé, ou au niveau haut. Autrement le neurone est dit désactivé, ou au niveau bas [Her94], [Jod94], [Fre92].

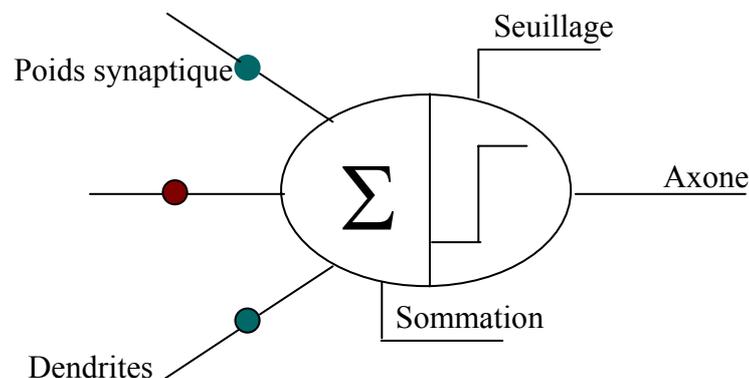


Figure 2.2 : Neurone formel

La (figure 2.3) montre la correspondance entre le neurone biologique et le neurone artificiel.

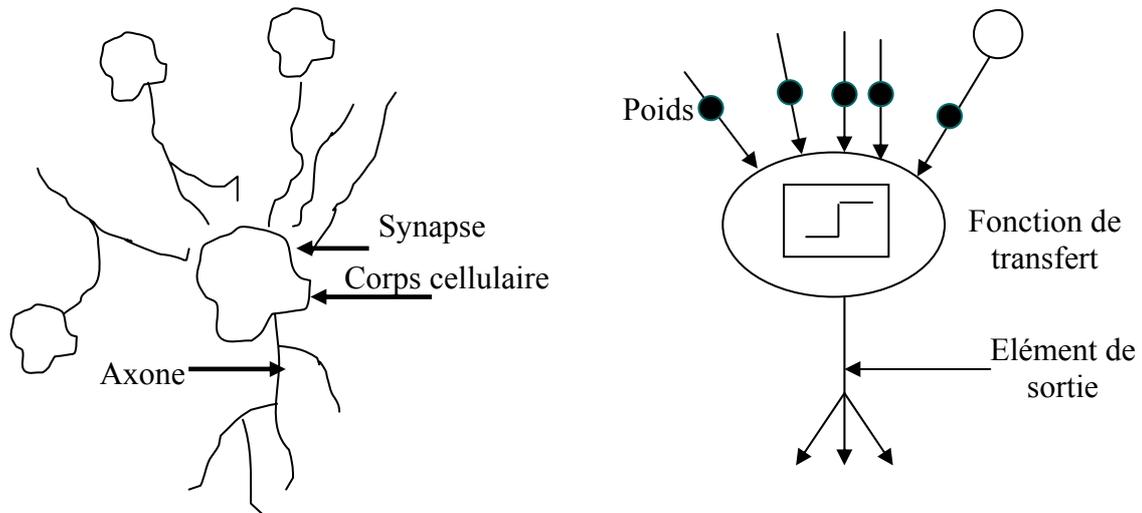


Figure 2.3 : Mise en correspondance neurone biologique/ neurone artificiel

Le neurone formel est modélisé par deux opérateurs comme la montre la (figure 2.4).

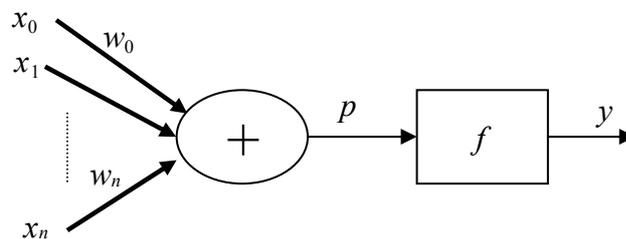


Figure 2.4 : Modèle formel d'un neurone

Les paramètres sont attachés aux entrées du neurone : La sortie du neurone est une fonction nonlinéaire d'une combinaison des entrées  $\{x_i\}$  pondérées par les paramètres  $\{w_i\}$ , qui sont souvent désignés sous le nom de « poids » ou, en raison de l'inspiration biologique des réseaux de neurones « poids synaptiques », conformément à l'usage (également inspiré par la biologie), cette combinaison linéaire sera appelé « potentiel ». Le potentiel  $p$  le plus fréquemment utilisé est la somme pondérée, à laquelle s'ajoute un terme constant ou « biais ».

### 2.3.1.a Opérateur de sommation

$$p = \sum_{i=1}^n w_i x_i + w_0 \quad (2.1)$$

Avec :

$p$  : potentiel somatique

$w_0$  : biais

$x_i$  : entrée du neurone  $i$

$w_i$  : poids synaptiques (Les connexions entre neurones ont des poids variables, ce qui rend les neurones plus ou moins influents sur d'autres neurones).

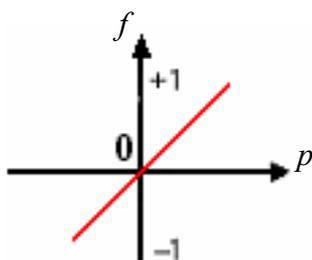
### 2.3.1.b Fonction de seuil (ou d'activation)

C'est un opérateur qui calcule la sortie  $y$  du neurone en fonction de son potentiel  $P$ .

$$y = f(p) \quad (2.2)$$

La fonction d'activation  $f$  doit être en générale saturable pour éviter des valeurs en sortie trop élevées qui peuvent déstabiliser le réseau. D'autre part, la fonction d'activation est en général à caractère nonlinéaire, ce qui rend les réseaux de neurones capables de simuler des fonctions complexes.

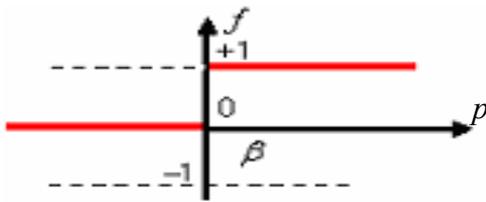
Les figures suivantes présentent des fonctions d'activation utilisées :



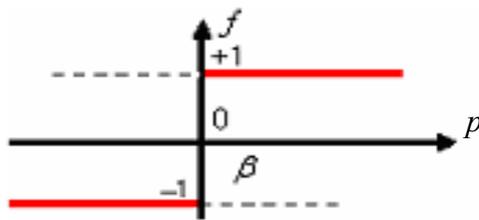
**Fonction identité**

Dans ce cas la sortie du neurone est égale au produit scalaire du vecteur d'entrée par le vecteur de poids.

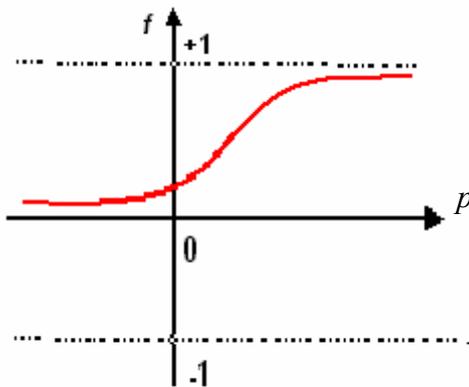
$$y = f(p) = p$$

**Fonction à seuil**

Ici la fonction  $f$  est booléenne, si le produit scalaire  $p$  dépasse un certain seuil  $\beta$  le neurone répond par 1, mais si  $p$  est inférieur ou égal à  $\beta$ , le neurone répond par 0.

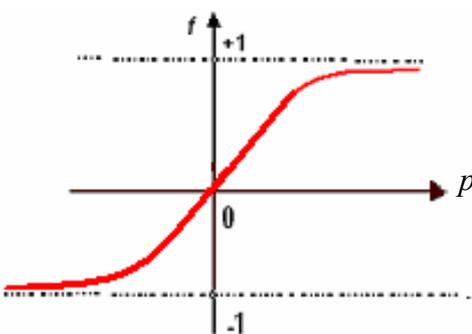
**Fonction signe**

Comme dans le cas précédent la fonction  $f$  est une fonction binaire, si  $p$  est supérieur à  $\beta$  la sortie est +1, autrement, si  $p$  est inférieur ou égal à  $\beta$  la sortie est -1.

**Fonction log-sigmoïde**

Dans ce cas  $f$  est une fonction sigmoïde prenant ses valeurs sur l'intervalle  $[0, 1]$ .

$$f(p) = \frac{1}{1 + \exp(-p)}$$

**Fonction tan-sigmoïde**

Il s'agit d'une fonction sigmoïde (tangente hyperbolique) prenant ses valeurs sur l'intervalle  $[-1, 1]$  suivant l'expression :

$$f(p) = \frac{\exp(p) - \exp(-p)}{\exp(p) + \exp(-p)}$$

## 2.4 Apprentissage des réseaux de neurones

On appelle apprentissage des réseaux de neurones la procédure qui consiste à estimer les paramètres des neurones du réseau, afin que celui-ci remplisse au mieux la tâche qui leur est affectée.

L'apprentissage est la phase qui permet de déterminer les poids et le seuil de chaque neurone dans un RNA. Une fois l'apprentissage terminé, le réseau est prêt à remplir sa fonction. Cette fonction peut être la mémorisation, la classification, ou l'optimisation.

Dans le cadre de cette définition, on peut distinguer deux types d'apprentissage : l'apprentissage supervisé et l'apprentissage non supervisé.

### 2.4.1 Apprentissage supervisé

Le processus d'apprentissage supervisé est réalisé à l'aide d'un expert qui fournit une sortie que le réseau doit approcher au mieux en réponse aux stimuli provenant de son environnement (figure 2.5). Pour ce faire, les deux réponses sont comparées pour définir un critère d'erreur. Les méthodes classiques d'optimisation sont utilisées pour minimiser ce critère de sorte et atteindre la précision désirée. Parmi les algorithmes qui utilisent ce type d'apprentissage, on trouve la rétropropagation du gradient.

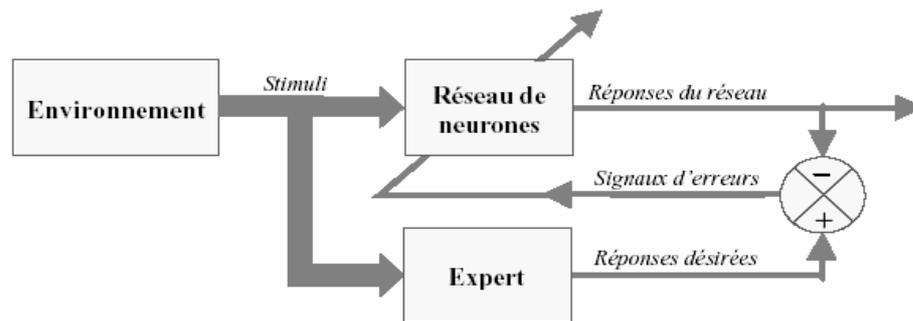


Figure 2.5 : Schéma d'un apprentissage supervisé.

### 2.4.2 Apprentissage non supervisé

L'apprentissage non supervisé se fait en l'absence d'expert. C'est le réseau lui-même qui détermine une classification en s'auto-organisant par association des réponses aux stimuli (figure 2.6). La qualité de représentation de la connaissance au sein du réseau est exprimée par un critère d'erreur mesuré dépendant uniquement des réponses. Le modèle de Kohonen, entre autres, utilise ce type d'apprentissage.

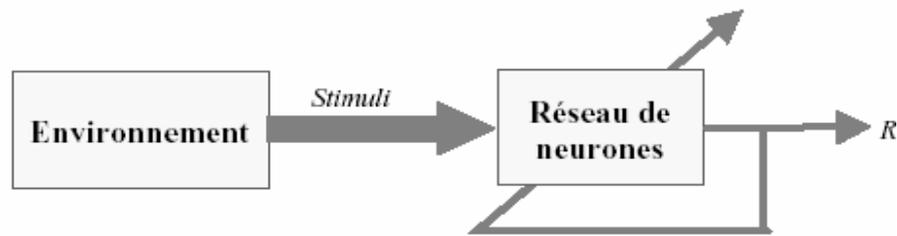


Figure 2.6: Schéma de l'apprentissage non supervisé.

L'apprentissage non supervisé est un outil à objectifs multiples :

-Dans le cas de l'optimisation il détermine les paramètres donnant la solution optimale au problème.

-Dans le cas de la classification il divise l'espace de représentation en régions de décisions dont les limites sont définies par les règles de décisions adoptées dans le RNA et les représentants de chaque classe seront codés dans les poids (modèle de Kohonen).

### 2.4.3 Surapprentissage

Les travaux expérimentaux menés sur les réseaux de neurones ont montré que, si on dépasse un certain nombre d'exemples d'entraînement, le réseau peut perdre sa capacité de généralisation, et donner ainsi des sorties non satisfaisantes pour des nouvelles entrées en gardant des réponses correctes pour les entrées déjà apprises, c'est ce que l'on appelle le surapprentissage [Ham93].

Pour éviter ce phénomène, on effectue un test de généralisation chaque fois que le réseau aura appris un certain nombre d'exemples. Si le test s'avère être négatif, on arrête l'introduction de nouveaux exemples, pour éviter le surapprentissage. Cette explication est résumée dans un schéma graphique (figure 2.7).

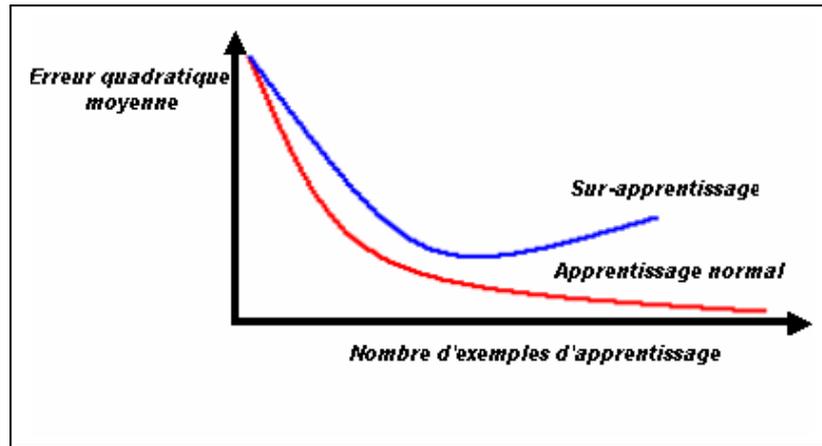


Figure 2.7 : Phénomène de surapprentissage

## 2.5 Mémoire associative

Un réseau neuronal n'est rien d'autre qu'une mémoire associative. Son travail consiste en premier lieu à stocker ou à mémoriser des données sous forme d'entrées/sorties durant la phase d'apprentissage, et en second lieu à se rappeler, un exemple déjà mémorisé durant la phase de généralisation même si l'exemple présenté est bruité. Si les entrées et sorties sont du même type, on parlera d'auto-association. Si les entrées et sorties ne sont pas du même type, on parlera d'hétéro-association.

Dans la phase de rappel, la notion la plus importante est celle de la décision. Le réseau doit décider si la donnée est plus proche de tel exemple appris plutôt que de tel autre. Cette capacité est tout simplement analogue au fonctionnement du cerveau humain, puisque celui-ci peut, après avoir vu le visage d'une personne, l'associer automatiquement au nom de celle-ci.

## 2.6 Architecture des réseaux de neurones artificiels

Le neurone lui-même, en tant qu'unité autonome élémentaire n'a aucun pouvoir. La force et l'efficacité du cerveau résident en effet, dans le regroupement de ces neurones et le partage des tâches entre eux. La connexion de plusieurs neurones entre eux, constitue ce qu'on appelle un champ de neurones, ces derniers forment à leur tour un réseau de neurones, où les unités intérieures travaillent ensemble pour accomplir une tâche bien déterminée. Il existe des réseaux de neurones qui sont organisés en couches: une couche d'entrée, des couches cachées, et une couche de sortie ; c'est ce que l'on appelle des réseaux multi-couches. Un autre type de réseaux sont les réseaux récurrents.

Aux cours des dernières années, différents modèles ont été créés et étudiés. Ces modèles diffèrent dans l'organisation de leurs neurones, les connexions entre ces derniers, et le type d'apprentissage. En se basant sur ces paramètres nous pourrions faire la classification d'architectures suivantes :

### 2.6.1 RNA à propagation directe

Dans ce cas les neurones sont organisés en couches (couche d'entrée, couches cachées et couche de sortie). Chaque couche ne communique sa sortie qu'à la couche suivante (figure 2.8). En effet, les neurones d'une couche ne sont connectés qu'aux neurones de la couche suivante à travers les poids de connexions et l'information se propage de l'entrée vers la sortie du RNA. D'où l'appellation propagation directe « feedforward ».

Les neurones sont reliés entre eux par des connexions pondérées. Ce sont les poids de ces connexions qui gouvernent le fonctionnement du réseau.

Durant l'apprentissage la propagation de l'information se fait dans les deux sens. Le signal d'entrée se propage vers la sortie, et le signal d'erreur se propage de la sortie vers l'entrée, ce qu'on appelle la rétropropagation de l'erreur qui provoque la modification des poids de connexion. Une fois l'apprentissage terminé, l'information reprend un sens unique de l'entrée vers la sortie.

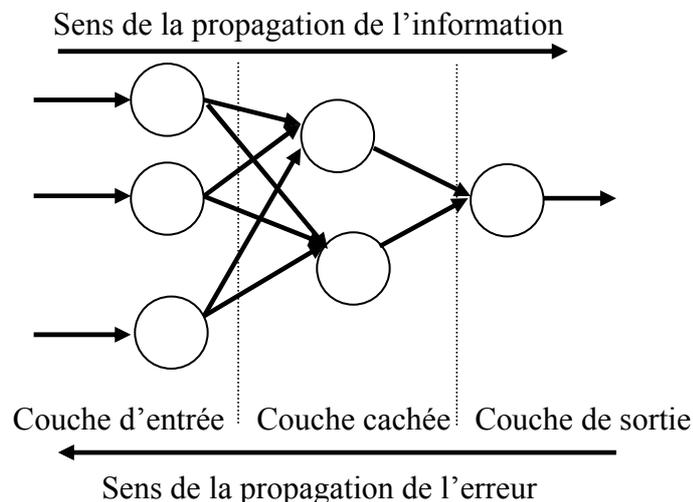


Figure 2.8 : RNA à propagation directe

### 2.6.2 Réseaux de neurones bouclés (ou récurrents)

L'information dans ce cas là se propage dans plusieurs sens. Les neurones possèdent une dynamique récurrente. En effet, le RNA est bouclé sur lui-même, d'où son nom de RNA récursif. Lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ (un tel chemin est désigné sous le terme de « cycle »). Ainsi, à chaque connexion d'un réseau de neurones bouclé (ou à chaque arête de son graphe) est attaché, outre un poids comme pour les réseaux non bouclés, un retard.

La (figure 2.9) représente un exemple de réseau de neurones bouclé. Les chiffres dans les carrés indiquent le retard attaché à chaque connexion, exprimé en multiple de l'unité de temps (ou période d'échantillonnage). Ce réseau contient un cycle, qui part du neurone 3 et revient à celui-ci en passant par le neurone 4. La connexion de 4 vers 3 ayant un retard de 1 unité de temps.

A l'instant  $kT$ , le neurone 3 reçoit en entrée les quantités  $u_1(kT)$ ,  $u_2(k-1)T$  et  $y_4(k-1)T$  (où  $k$  est un entier positif et  $y_4(kT)$  est la sortie du neurone 4 à l'instant  $kT$ ), et calcule sa sortie  $y_3(kT)$ . Le neurone 4 reçoit  $u_2(kT)$  et  $y_3(kT)$ , et calcule sa sortie  $y_4(kT)$ . Le neurone 5 reçoit  $y_3(kT)$ ,  $u_1(kT)$  et  $y_4(k-1)T$ , et calcule sa sortie, qui est la sortie du réseau de neurones  $g(kT)$ .

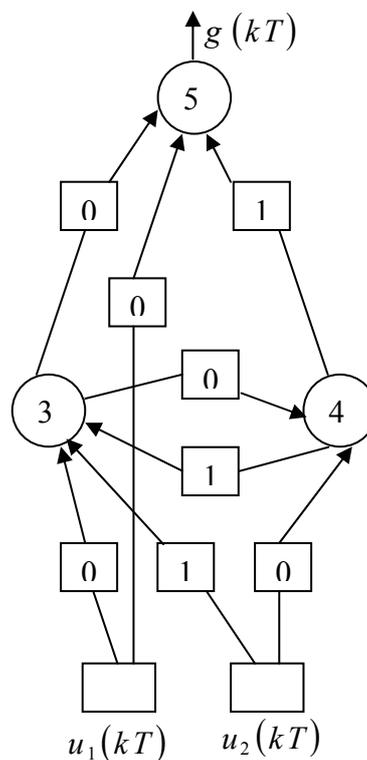


Figure 2.9 : Réseau de neurones bouclé à deux entrées

## 2.7 Algorithmes d'apprentissage

La phase d'apprentissage est une étape déterminante dans la conception du réseau de neurones. Pour cela, des algorithmes appropriés ont été élaborés et développés au fil des années chaque algorithme est développé pour un type spécifique de réseau. Ces méthodes sont basées sur des techniques mathématiques connues qui ont prouvée leurs efficacités.

### 2.7.1 Algorithme de rétropropagation (Back-Propagation BP)

Nous présentons l'algorithme le plus connu qui est celui de rétropropagation du gradient, découvert par Werbos, dans sa thèse de doctorat, et développé par Rumelhart, Hinton et Williams en 1986. C'est grâce à lui que l'on doit le renouveau d'intérêt pour les RNA au début des années 80. Il s'agit d'une méthode d'apprentissage supervisée.

Cet algorithme offre une procédure d'apprentissage commode. Il est basé sur une méthode numérique dite de relaxation qui effectue une descente de gradient sur la surface d'erreur quadratique moyenne. Il utilise donc les techniques de dérivées partielles (on peut représenter la descente de gradient comme une bille que l'on poserait sur la surface (figure 2.12), et qui descendrait le long de la pente), le gradient représente la pente instantanée en un point donné de la surface. Le principe de la méthode est résumé dans la (figure 2.10). Il s'agit de présenter au réseau un vecteur d'entrée et un vecteur de sortie désirée. Le réseau va calculer sa propre sortie par une propagation avant (forward). Une fois la sortie calculée, celle-ci est comparée à la valeur désirée (erreur). Cette erreur est utilisée pour l'adaptation des pondérations en faisant une propagation arrière (backward), de la couche de sortie vers la couche d'entrée, dans le but de minimiser l'erreur.

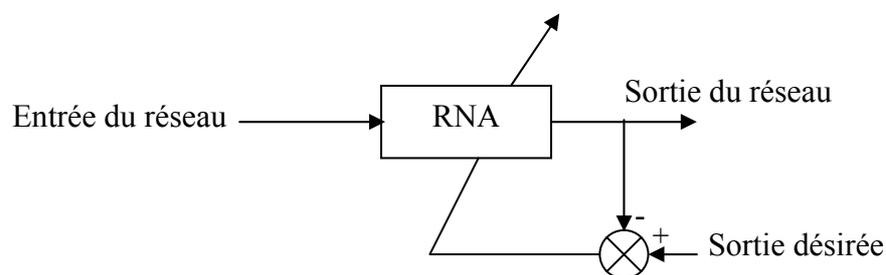


Figure 2.10: Principe de la rétropropagation

### 2.7.2 Résumé de l'algorithme BP

Soit le réseau de la (figure 2.11) à  $n_0$  neurones en entrées,  $n_1$  neurones dans la couche cachées,  $n_2$  neurones en sortie,  $\eta$  s'appelle le pas du gradient, taux d'apprentissage ou bien le gain de l'algorithme  $0 \leq \eta \leq 1$

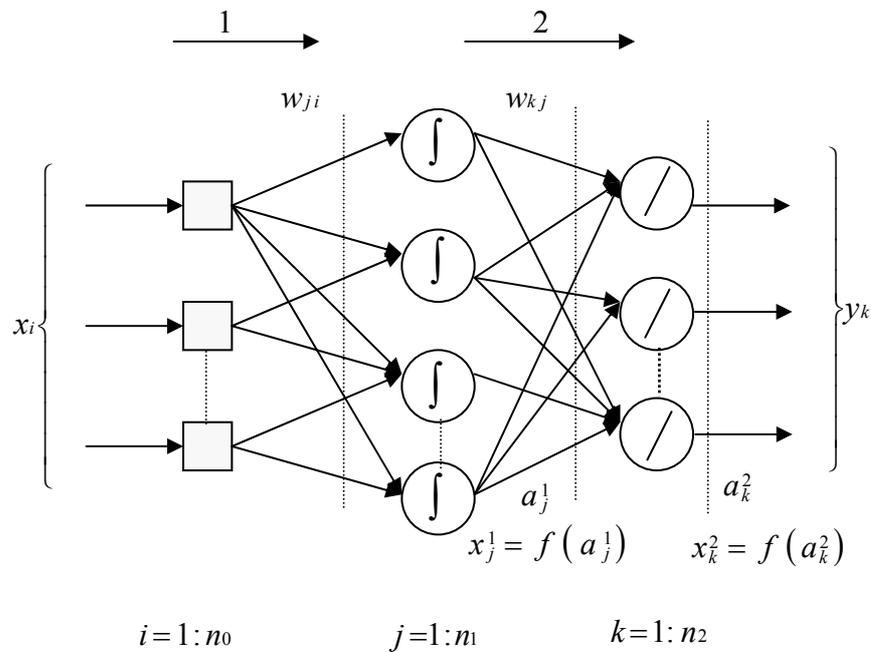


Figure 2.11 : Réseau multi-couches

#### 1-Propagation

On calcule les sorties du réseau en propageant les valeurs de  $x$ , de la couche d'entrée vers la couche de sortie.  $a$  représente l'entrée de chaque couche de neurone.

$$a_j^1 = \sum_{i=1}^{n_0} w_{ji} x_i \quad (2.3)$$

$$x_j^1 = f(a_j^1) \quad (2.4)$$

$$a_k^2 = \sum_{j=1}^{n_1} w_{kj} x_j^1 \quad (2.5)$$

$$y_k = g(a_k^2) = a_k^2 \quad (2.6)$$

## 2-Calcul de l'erreur

-On présente un exemple  $x=[x_1 \dots x_n]$  (avec  $y^{des}$ , sortie désirée)

-On calcule la sortie correspondante  $y=[y_1 \dots y_n]$ , l'erreur est :  $e_k = y_k^{des} - y_k$

-Le coût associé à l'exemple est :  $J(exemple) = \frac{1}{2} \sum_{k=1}^{n_2} e_k^2$  (Somme des erreurs quadratiques observées).

## 3- Calcul du gradient

La correction des poids de neurones de  $w_{ji}$  et  $w_{kj}$  sont calculés par :

$$\Delta w = -\eta \frac{\partial J}{\partial w} \quad (2.7)$$

### 3.1- Couche de sortie

Calcul de  $\frac{\partial J}{\partial w_{kj}}$  pour un exemple fixé

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial y_k} \frac{\partial y_k}{\partial a_k^2} \frac{\partial a_k^2}{\partial w_{kj}} \quad (2.8)$$

$$J = \frac{1}{2} \sum_{k=1}^{n_2} (y_k^{des} - y_k)^2 \Rightarrow \frac{\partial J}{\partial y_k} = -(y_k^{des} - y_k) : \text{La forme connue pour } J \text{ est l'erreur}$$

quadratique de la sortie  $y$  par rapport à la sortie désirée  $y_d$ .

$$y_k = g(a_k^2) \Rightarrow \frac{\partial y_k}{\partial a_k^2} = g'(a_k^2) = 1 \quad (2.9)$$

$$a_k^2 = \sum_{j=1}^{n_1} w_{kj} x_j^1 \Rightarrow \frac{\partial a_k^2}{\partial w_{kj}} = x_j^1 \quad (2.10)$$

$$\text{Posons : } S_k = \frac{\partial J}{\partial a_k^2} = -(y_k^{des} - y_k) g'(a_k^2) \quad (2.11)$$

$$\Rightarrow \frac{\partial J}{\partial w_{kj}} = S_k x_j^1 \quad (2.12)$$

### 3. 2- Couche cachée

Calcul de  $\frac{\partial J}{\partial w_{ji}}$  pour un exemple fixé

$$\frac{\partial J}{\partial w_{ji}} = \frac{\partial J}{\partial x_j^1} \frac{\partial x_j^1}{\partial a_j^1} \frac{\partial a_j^1}{\partial w_{ji}} \quad (2.13)$$

$$\frac{\partial J}{\partial x_j^1} = \sum_{k=1}^{n_2} \frac{\partial J}{\partial a_k^2} \frac{\partial a_k^2}{\partial x_j^1} = \sum_{k=1}^{n_2} S_k w_{kj} \quad (2.14)$$

$$y_j^1 = f(a_j^1) \Rightarrow \frac{\partial x_j^1}{\partial a_j^1} = f'(a_j^1) \quad (2.15)$$

$$a_j^1 = \sum_{i=0}^{n_0} w_{ji} x_i \Rightarrow \frac{\partial a_j^1}{\partial w_{ji}} = x_i \quad (2.16)$$

Posons :  $S_j = \frac{\partial J}{\partial a_j^1}$ , Alors :  $S_j = \left( \sum_{k=1}^{n_2} S_k w_{kj} \right) f'(a_j^1)$  (2.17)

$$\frac{\partial J}{\partial w_{ji}} = S_j x_i \quad (2.18)$$

**Remarque 1** [Mar04] : Pour corriger l'erreur observée, il s'agit de modifier les poids  $w$  dans le sens opposés au gradient  $\frac{\partial J}{\partial w}$  de l'erreur. Cette dérivée partielle représente également la fonction de sensibilité (figure 2.12).

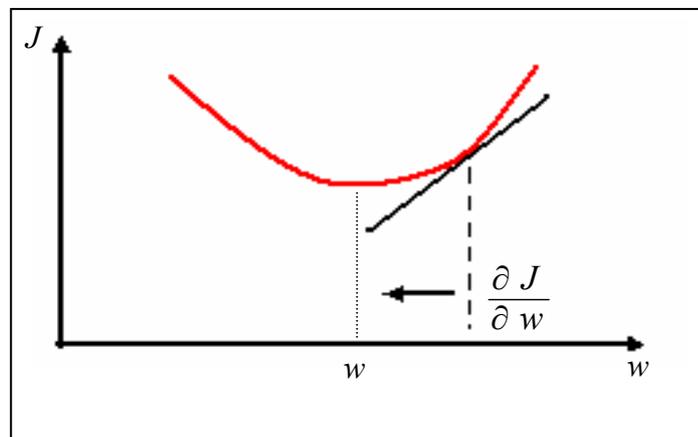


Figure 2.12 : Gradient de l'erreur totale

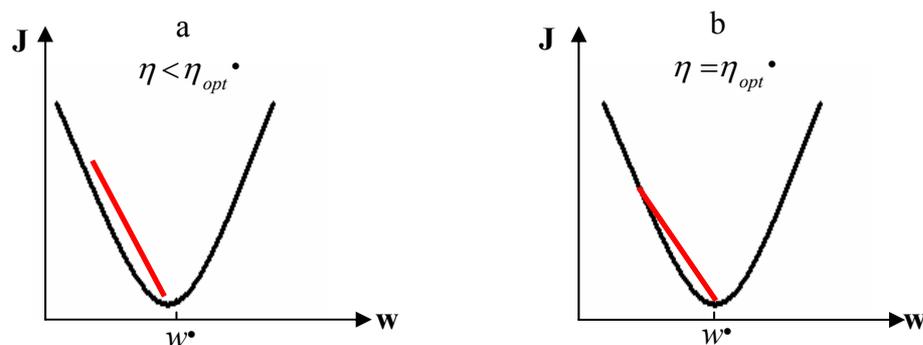
## Remarque 2

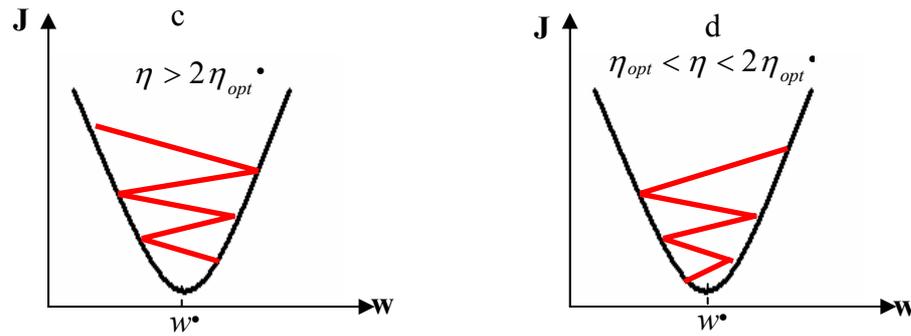
- L'initialisation des poids synaptiques  $w$  est aléatoire.
- L'expérience a montré qu'il est généralement, préférable d'initialiser les poids à des valeurs comprises entre 0 et 1 pour éviter de déstabiliser le réseau dès le départ [Fre92], [Shu90].
- Le pas du gradient  $\eta$  est choisi entre 0 et 1, et on le diminuera au fur et à mesure pour atteindre une valeur fixe.
- Le choix des exemples d'apprentissage est très important, un choix judicieux de ces derniers pourra faciliter la généralisation sur des exemples non appris. Ceci a poussé les chercheurs à faire l'étude statistique des données d'entraînement pour trier celles qui sont le plus représentatives du phénomène qu'on souhaite avoir [Ham93].
- L'algorithme de rétropropagation du gradient a pu être étendu à certaines classes de réseaux récurrents c'est ce que l'on appelle la rétropropagation dynamique.

### 2.7.3 Difficultés et limites de l'algorithme

Des questions restent posées quant à l'utilisation de cet algorithme, à savoir :

- Combien faut-il utiliser des couches, et combien doit-il y avoir de neurones par couche ?
- Les réponses ne sont données que par expérience, et il n'existe aucune règle théorique précise qui puisse fixer le nombre de neurones ou de couches dans le réseau.
- Le problème de minimisation n'est pas facile à résoudre. En effet, la surface d'erreur peut présenter des caractéristiques peu satisfaisantes, telles que des minima locaux qui empêchent la convergence vers le minimum global, ainsi que des plateaux où les pentes sont très faibles.
- Le pas du gradient (figure 2.13) peut être difficile à choisir. S'il est faible, la convergence peut être très lente. S'il est élevé, l'erreur risque d'osciller sans pour autant converger.





a- Convergence lente    b- Convergence rapide    c- Divergence  
d- Convergence oscillatoire

Figure 2.13 : Convergence et choix de  $\eta$

#### 2.7.4 Algorithme de back-propagation avec momentum

Comme nous l'avons mentionné précédemment, le pas du gradient peut être difficile à choisir. S'il est faible, la convergence peut être très lente. S'il est élevé, l'erreur risque d'osciller sans pour autant converger.

Une solution proposée par D.E.Rumelhart, consiste à introduire un terme supplémentaire pour la réadaptation des poids.

L'équation d'adaptation devient alors :

$$\Delta w_{ji}(t) = -\eta E_{rj} x_i + \alpha \Delta w_{ji}(t-1) \quad (2.19)$$

Le terme ajouté est appelé momentum (élan, quantité de mouvement), en analogie avec la mécanique classique, ou un objet en mouvement garde l'élan acquis grâce à la quantité de mouvement qui lui a été communiquée précédemment. Le paramètre  $\alpha$  est utilisé pour pondérer l'effet de ce terme. Sa valeur est généralement prise entre 0.8 et 0.9. Des recherches plus poussées ont montré que l'adaptation de sa valeur, comme c'est le cas pour le taux d'apprentissage  $\eta$ , donne des résultats très intéressants.

L'utilisation de cette méthode permet de faire sortir les poids des minimums locaux, (figure 2.14) afin de chercher d'autres optimums. Ce qui donne beaucoup de chance d'aboutir à un minimum global.

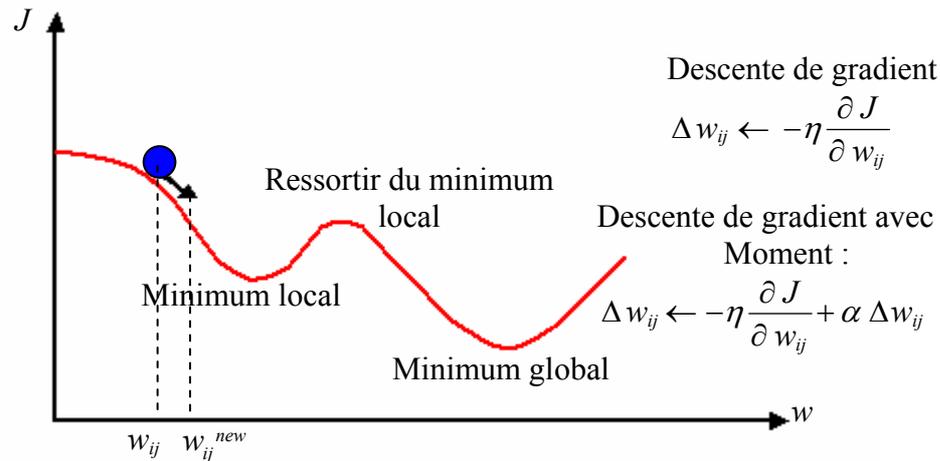


Figure 2.14 : Effet de momentum pour échapper d'un minimum local

### 2.7.5 Application de l'algorithme (BP)

Malgré les difficultés citées, l'algorithme de rétropropagation s'est révélé très performant dans la résolution de plusieurs problèmes, tels que [Fre92]:

- La reconnaissance des formes géométrique.
- Le traitement et l'analyse des signaux.
- La commande des processus.
- La classification
- Le filtrage de bruit
- Le diagnostic médical.

## 2.8 Optimisation des réseaux neuronaux

On dispose d'un réseau neuronal, d'un échantillon de données empiriques pour entraîner le réseau et d'un algorithme d'apprentissage approprié. La question posée est quelle doit être la taille du réseau pour obtenir la meilleure généralisation possible?

L'approximation de fonctions complexes nécessite un réseau de grande taille avec de nombreuses couches cachées. Cependant, outre l'inconvénient qu'il présente : temps de calcul élevé, un réseau surdimensionné risque de conduire à une mauvaise généralisation ou surapprentissage. L'amélioration des capacités de généralisation est essentiellement expérimentale. Aucune méthode ne permet de déterminer précisément la structure du réseau en fonction des exemples à prendre pour fournir une généralisation optimale. Les travaux

présentés par le mathématicien Vapnik [Vap82] offrent un cadre théorique sur l'application de la minimisation structurelle aux réseaux neuronaux. Même si ces travaux ne sont pas encore validés pratiquement, ils constituent une avancée remarquable sur le plan théorique, et motivent actuellement beaucoup de recherches pour expliquer les observations expérimentales [Bau89].

## 2.9 Réseaux de neurones et approximateurs universels

Cybenko et Hornik (1989) considèrent les réseaux de neurones comme une classe d'approximateurs universels. On a ainsi démontré qu'un perceptron multicouche avec une seule couche cachée pourvue d'un nombre suffisant de neurones, peut approximer n'importe quelle fonction avec la précision souhaitée. Néanmoins, cette propriété ne permet pas de choisir, pour un type de fonction donné, le nombre de neurones optimal dans la couche cachée.

## 2.10 Conclusion

Dans ce chapitre, nous avons introduit différentes notions théoriques concernant les réseaux de neurones tels que l'architecture, l'activation et l'apprentissage. Chaque réseau est caractérisé par un certain nombre de paramètres (fonction d'activation, connexions des différents neurones du réseau, etc...), en tenant compte de cela, nous aboutissons au réseau de neurones multicouche.

De manière générale les réseaux de neurones présentent un moyen efficace pour résoudre des problèmes pour lesquels, les méthodes classiques ont montré leurs limites, que cela soit en identification ou en commande, et les résultats obtenus sont appréciables.

Bien que la détermination de l'architecture optimale du réseau n'est pas cernée par des règles théoriques précises, l'expérience acquise durant les diverses applications nous permet d'améliorer les performances en jouant sur plusieurs paramètres, tels que le nombre de couche, le nombre de neurones par couche, le type de fonction d'activation, les coefficients d'apprentissage, etc...

# **Chapitre 3**

**Identification et commande par  
réseaux de neurones**

# **Chapitre 3**

## **Identification et commande par réseaux de neurones**

---

### **3.1 Introduction**

La commande des systèmes (processus) non linéaires, pouvant être approchées par des systèmes linéaires, est aujourd'hui un domaine bien maîtrisé par les Automaticiens.

Les outils de l'algèbre linéaire permettent d'obtenir pour ces systèmes des contrôleurs possédant de bonnes propriétés en stabilité, en temps de réponse et en amortissement. Cependant dans le monde réel, un grand nombre de processus sont caractérisés par un comportement dynamique non linéaire complexe, et cela rend difficile l'utilisation des outils classiques de l'Automatique. Il en est de même pour les systèmes pour lesquels les modèles mathématiques connus sont incomplets ou défaillants. Il n'existe pas aujourd'hui de théorie systématique applicable à la commande de tels processus. Pour résoudre ce problème une des solutions proposées consiste à avoir recours à un apprentissage pour reconnaître le modèle du processus ou du contrôleur.

Le terme apprentissage désigne ici la modification de structure et /ou des paramètres du système, de manière à améliorer ses performances futures, en se basant sur des observations expérimentales passées [Bak92].

Les RNA permettent d'approcher n'importe quelle fonction nonlinéaire [Hay94] car se sont des approximateurs universels de fonctions. C'est cette propriété qui motive leur utilisation pour la réalisation de systèmes de commande non linéaire par apprentissage.

### **3.2 Apprentissage en commande et commande adaptative [Gau99]**

De nombreuses méthodes de commande adaptative ont été développées par les

Automaticiens pour permettre au contrôleur d'évoluer (de s'adapter) en fonction de l'évolution de l'environnement. Ces méthodes permettent, une fois la structure du contrôleur choisie, de régler les paramètres de ce dernier.

Si le principe général mis en œuvre dans ces algorithmes est similaire à l'apprentissage réalisé par les RNA, l'adaptation se résume à un réglage de coefficients de la boucle de commande.

L'utilisation de systèmes ayant des possibilités d'apprentissage importante tels que les RNA, permet de gagner dans l'automatisation des processus, comme la prise de décision ou la reconnaissance de situation [Cos97], on parle dans ce cas d'apprentissage en commande plutôt que de commande adaptative.

Un système d'apprentissage en commande mémorise les paramètres associés aux différents points de fonctionnement du processus. Alors qu'un système de commande adaptative doit réajuster ses paramètres chaque fois que les conditions de fonctionnement changent.

Les méthodes de commande adaptative nécessitent de plus la connaissance a priori de la structure du système. Alors que les RNA peuvent se passer de cette connaissance.

### 3.3 Quelques propriétés [Gau99]

Les RNA possèdent plusieurs caractéristiques intéressantes pour la réalisation d'un système de commande :

-La parallélisation du traitement leur confère une grande rapidité de calcul et les rend très adaptés aux applications temps réel.

-La bonne résistance aux pannes internes. Certains auteurs se sont intéressés à l'étude de l'évolution de leur performance lorsque certains neurones ou certaines connexions sont supprimées arbitrairement. R. Velazco étudie par exemple les possibilités d'utilisation des RNA dans des applications spatiales embarquées pour lesquelles les circuits électroniques sont dégradés par les rayonnements solaire [Ass96].

-Leurs capacités de généralisation leur confèrent une bonne résistance aux bruits. Ceci est particulièrement important lorsque les capteurs permettant de mesurer l'état ou la sortie de processus commandé sont sujets à des perturbations.

### 3.4 Classification des méthodes de commande

Dans cette section nous présentons une classification des principales méthodes de commande neuronales. La plupart d'entre elles sont indépendantes du type de processus commandé. Les modèles seront présentés pour des processus ne comportant qu'une seule entrée et une seule sortie, mais peuvent s'appliquer dans le cas multivariables.

La grande majorité des architectures rencontrées utilisent des réseaux multicouches entraînés par rétropropagation du gradient, mais d'autres modèles sont possibles. K.Narendra [Nar90] note en particulier que les réseaux récurrents, en raison de leur aspect dynamique, sont plus à même de contrôler les systèmes dynamique.

Nous séparons ces approches en trois classes principales:

- Méthodes directes: On calcule directement la commande sans passer par l'identification du système.
- Méthodes indirectes: On passe par l'identification du système
- Commande adaptative

#### 3.4.1 Méthodes directes

##### 3.4.1.1 Reproduction d'un contrôleur existant

La première méthode utilisée pour la réalisation d'un système de commande neuronal consiste à reproduire le fonctionnement d'un contrôleur existant, même si cette approche semble au premier abord peu intéressante puisqu'elle nécessite l'existence d'un autre contrôleur, elle peut s'avérer utile si ce dernier est trop complexe ou trop lent pour être utilisé en temps réel, ou encore s'il utilise des données qui ne sont pas disponibles en permanence.

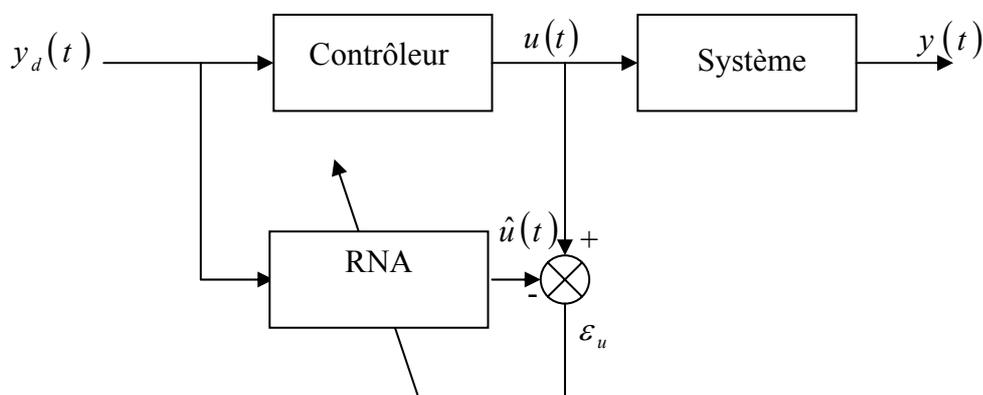


Figure 3.1 : Apprentissage d'un système de commande neuronal pour reproduction d'un contrôleur existant

L'architecture générale est représentée sur la (figure 3.1). Elle consiste à apprendre au réseau à reproduire la commande  $\hat{u}(t)$  préconisée par le premier contrôleur à partir de la sortie désirée  $y_d(t)$  et éventuellement de la sortie précédente.

Un des premiers exemples de système de commande neuronal, proposé par B. Widrow et F. W. Smith [Wid64] en 1964, a utilisé cette technique pour résoudre le problème de commande d'un pendule inversé.

**Remarque 1 :** On peut noter que cette approche demande de parcourir, lors de la phase d'apprentissage, tous les modes de fonctionnement du système commandé. Il est donc nécessaire de posséder une bonne connaissance a priori des conditions d'utilisation du contrôleur.

### 3.4.1.2 Amélioration d'un système de commande linéaire

Cette approche consiste à utiliser conjointement un contrôleur linéaire classique et un contrôleur neuronal. L'idée principale est de réaliser une somme de commandes issues de deux contrôleurs, en augmentant progressivement l'importance donnée à la commande  $\hat{u}(t)$  préconisée par le réseau neuronal, au fur et à mesure de l'apprentissage de ce dernier. M. Kawato [Kaw90] propose d'utiliser un contrôleur de type « feedback » classique (CFC). Dans sa version la plus simple un tel contrôleur se contente d'appliquer un gain sur l'erreur  $\varepsilon_y$  mesurée en sortie du processus. La commande est donc proportionnelle à cette erreur, on obtient donc l'architecture décrite par la (figure 3.2). Le réseau reçoit en entrée la sortie désirée  $y_d(t)$  et on utilise comme signal d'erreur la sortie de CFC. Lorsque l'apprentissage avance, le réseau apprend à minimiser par ses commandes la sortie du CFC et par conséquent l'erreur en sortie du processus. On trouve parfois pour cette architecture le nom « feedback error learning ». H. Myamoto [Mya88] a utilisé cette approche pour la commande d'un bras de robot.

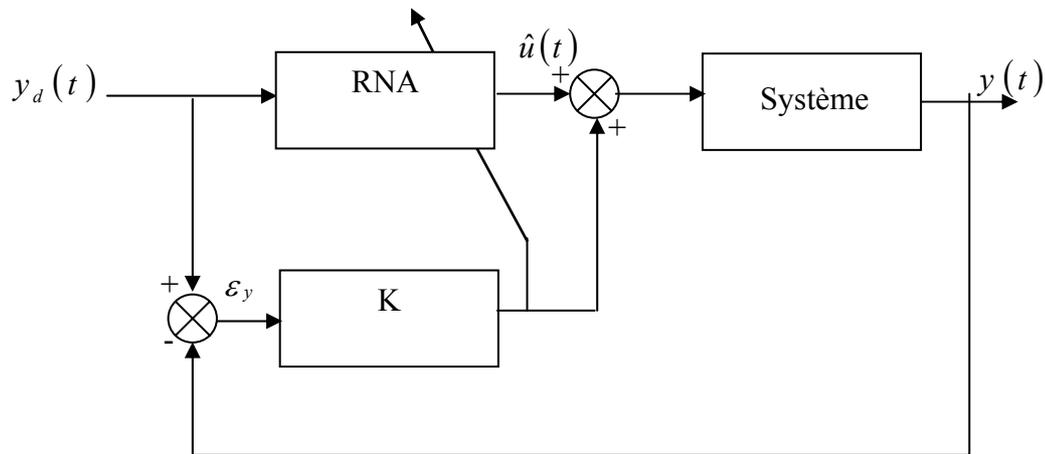
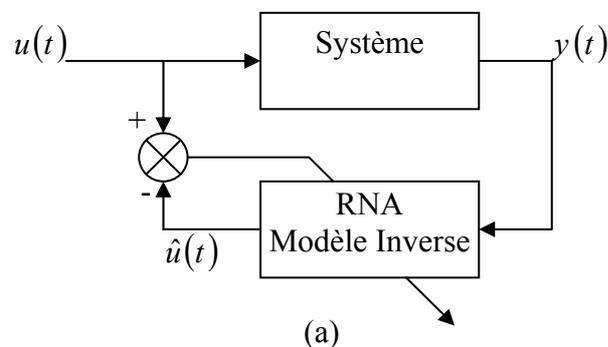


Figure 3.2 : Amélioration d'un contrôleur feedback classique

### 3.4.1.3 Identification directe d'un modèle inverse

Cette approche nécessite deux phases séparées pour l'apprentissage, puis pour l'utilisation de réseau. Durant l'apprentissage, le réseau et le processus sont placés en parallèle. Un échantillon de commande  $u$  est fourni au processus. On utilise alors les sorties  $y$  de ce dernier comme entrée du réseau qui est entraîné à retrouver en sortie les commandes  $u$ . Le réseau apprend ainsi un modèle inverse du processus, c'est à dire une fonction donnant la commande appliquée  $u(t)$  à partir de la sortie actuelle du processus  $y(t)$  et éventuellement de sa sortie  $y(t-1)$ . Après cette phase d'apprentissage le réseau est donc théoriquement capable de fournir la commande  $u(t)$  nécessaire pour obtenir une sortie  $y_d(t)$  qui lui est donnée en entrée. Il est donc placé directement en série avec le système commandé (figure 3.3).



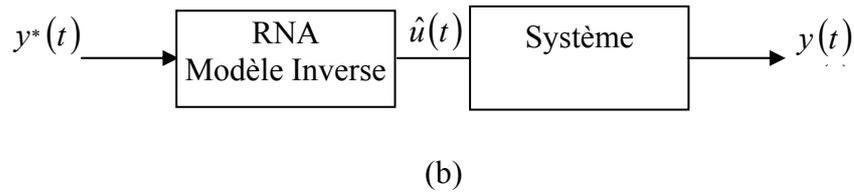


Figure 3.3 : a) Phase d'apprentissage d'identification du modèle inverse  
b) Phase d'utilisation

La limitation majeure de cette approche vient du fait que le processus doit évidemment être inversible pour que le système de commande fonctionne.

Cette approche a cependant été utilisée avec succès par de nombreux auteurs. On peut citer par exemple les travaux de Y. Kuroe [Kur94] qui l'utilisent pour la commande d'un bras de robot.

D. Psaltis [Psa87] proposent une variante de cette architecture, appelée « Indirect learning architecture », décrite par la (figure 3.4) Dans cette architecture les réseaux partagent les mêmes poids (c'est à dire le deuxième réseau est une copie de premier). La valeur de sortie désirée  $y_d$  est propagée à travers le premier réseau pour produire une commande  $\hat{u}_1$  qui est appliquée au système. La valeur de sortie  $y$  mesurée est alors utilisée comme entrée du deuxième réseau pour obtenir une deuxième consigne  $\hat{u}_2$ . C'est la différence entre  $\hat{u}_1$  et  $\hat{u}_2$  qui est utilisée pour corriger les réseaux. L'idée exploitée par cette architecture est que si l'erreur tend vers zéro, la différence  $y - y_d$  tend également vers zéro, on constate alors que le système converge souvent vers une solution triviale donnant toujours la même commande, ce qui rend l'architecture peut utilisable.

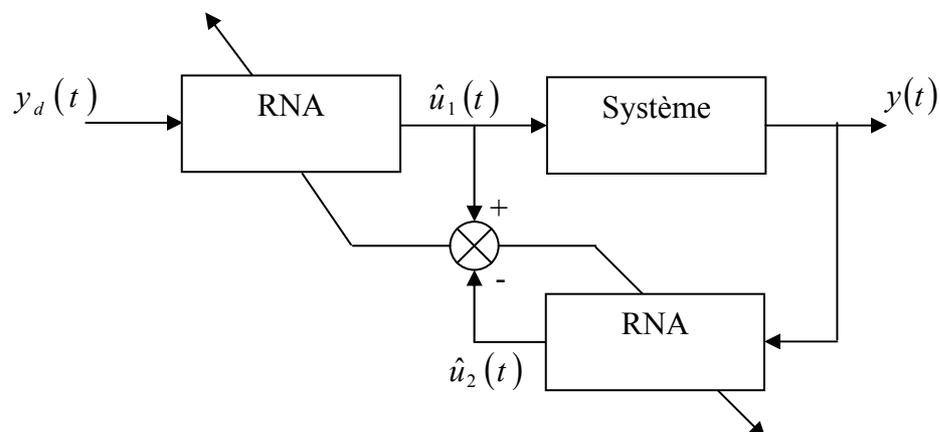


Figure 3.4: Indirect learning architecture

**Remarque2 :** Le terme Indirect n'a ici aucun rapport avec la notion de méthode de commande directe ou indirecte.

### 3.4.1.4 Apprentissage direct par estimation de l'erreur

Il s'agit ici de réaliser l'apprentissage d'un modèle de commande en boucle ouverte du procédé. On dispose à cet effet en série le modèle connexionniste ou bien le réseau de neurone (figure 3.5), et le procédé à commander. Le signal d'erreur correspond à l'erreur de consigne  $\varepsilon$  et est utilisée afin de réaliser l'apprentissage des paramètres du modèle connexionniste. On peut remarquer que la valeur utilisée pour l'apprentissage des paramètres du modèle de commande est l'erreur en sortie du procédé. Une utilisation adéquate de la rétropropagation imposerait portant que l'on rétropropage l'erreur en sortie du modèle, qui est naturellement inconnue.

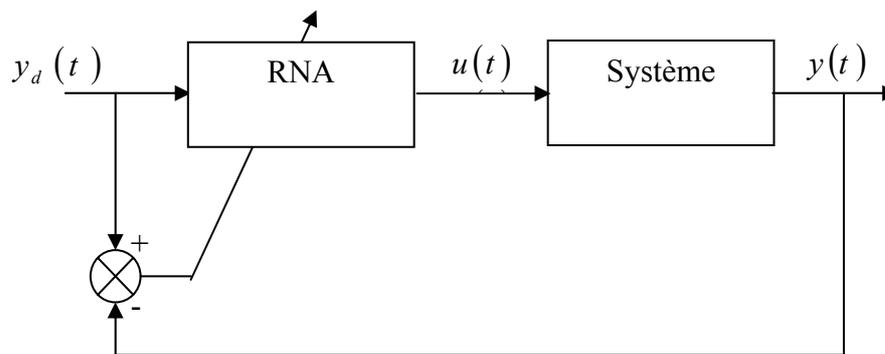


Figure 3.5 : Identification du modèle de commande

## 3.4.2 Méthodes indirectes

### 3.4.2.1 Identification des processus

Nous supposons que le système à identifier est de la forme générale :

$$y(k) = f(y(k-1), y(k-2), \dots, y(k-n), u(k-1), u(k-2), \dots, u(k-m)) \quad (3.1)$$

Où  $f(\cdot)$  est une fonction nonlinéaire.

Nous entamons cette classification par une présentation des méthodes d'identification de processus utilisant des RNA. Par identification d'un processus nous entendons l'entraînement d'un réseau, pour reproduire une fonction donnant les sorties ou l'état du processus à partir des entrées qui lui sont appliquées.

Le principe général est simple, consiste à placer en parallèle le réseau et le processus à identifier, comme indiqué sur la (figure 3.6). Le réseau reçoit en entrée la commande  $u(t)$  appliquée, les commandes passées ( $u(t-1), \dots, u(t-m)$ ) et les sorties précédentes du processus. Il est entraîné à produire la nouvelle sortie (ou le nouvel état)  $y(t)$  du processus. Cette méthode d'identification est souvent appelée méthode série- parallèle. Dans la méthode parallèle, le réseau ne reçoit pas en entrée la sortie réelle  $y(t-1)$  du processus mais la sortie  $\hat{y}(t-1)$  (figure 3.7) qui est la sortie du réseau au temps précédent.

La fonction à approximer par le réseau dans ce cas est :

$$\hat{y}(k) = f(\hat{y}(k-1), \hat{y}(k-2), \dots, \hat{y}(k-n), u(k-1), u(k-2), \dots, u(k-m)) \quad (3.2)$$

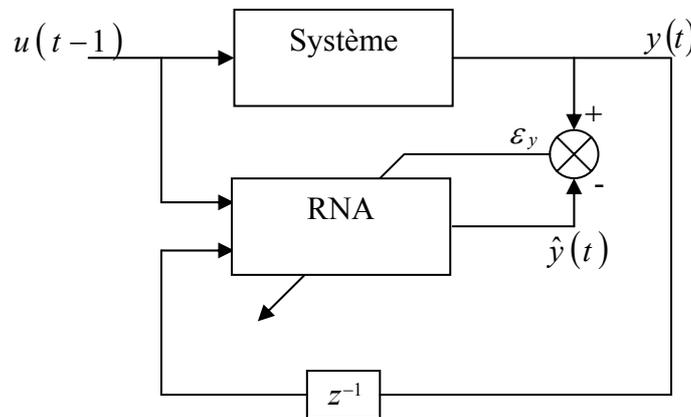


Figure 3.6 : Identification de processus par la méthode série- parallèle

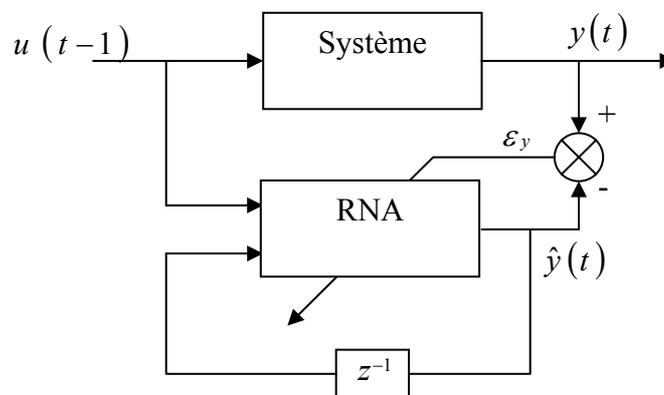


Figure 3.7 : Identification de processus par la méthode parallèle

L'approche série-parallèle est plus stable [Ren95] car le réseau est régulièrement « recalé en utilisant l'état réel du processus ». La méthode parallèle ne garantit pas la convergence.

### 3.4.2.2 Commande par modèle direct neuronal

Cette approche consiste à utiliser un modèle neuronal direct du processus. En supposant que le modèle coïncide parfaitement avec le processus réel, une stratégie de commande est établie en s'appuyant sur le réseau estimateur (figure 3.8).

La stratégie de commande se base souvent sur des méthodes d'optimisation d'un critère portant sur les sorties prédites afin qu'elles soient le plus proche possible d'un comportement de référence et éventuellement sur les variables de commande afin d'assurer une conduite sans à-coups.

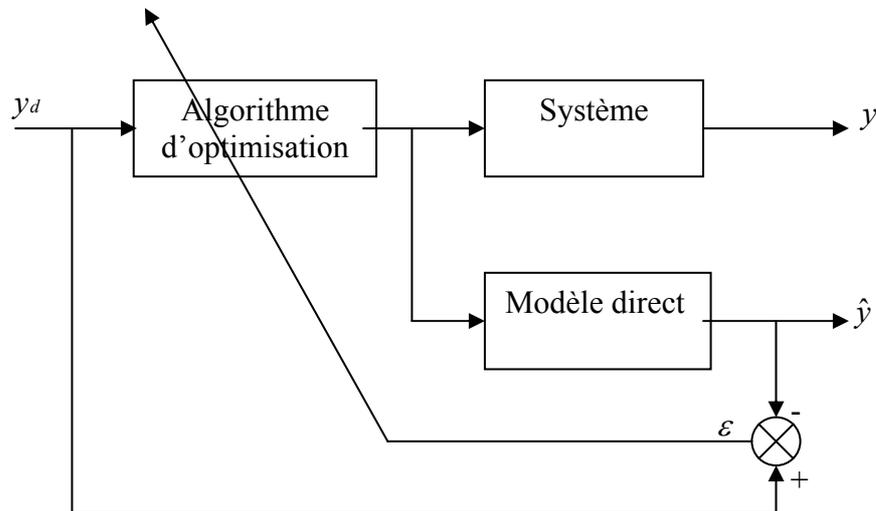


Figure 3.8 : Commande par modèle direct neuronal

Les méthodes d'optimisation utilisées sont par exemple :

- Un algorithme génétique.
- Une descente de gradient :

$$\Delta u = -\eta \frac{\partial J}{\partial u} = -\eta \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial u} \quad (3.3)$$

$$\text{Si } J = \frac{1}{2} [y - \hat{y}]^2 \quad (3.4)$$

$$\text{On } \Delta u = \eta (y_d - \hat{y}) \frac{\partial \hat{y}}{\partial u}$$

$$u(k+1) = u(k) + \Delta u \quad (3.5)$$

### 3.4.3 Commande adaptative par réseaux de neurones

M'Saad a montré qu'une telle commande peut nous libérer de la nécessité d'établissement d'un modèle paramétrique de représentation pour le système, et de l'estimation de ses paramètres. Ceci, grâce à la capacité d'estimation des réseaux de neurones, ce qui la rend adaptée pour la commande des bras de robot [Saa94].

Cette commande adaptative neuronale a été réalisée sous l'aspect de régulateurs auto-ajustables et de commande avec modèle de référence.

#### 3.4.3.1 Régulateur Auto-ajustable

Un régulateur auto-ajustable (figure 3.9) à base de réseaux de neurones, exploite les capacités d'approximation de ses réseaux, pour l'estimation des paramètres définissant le système, afin de les utiliser, pour calculer la commande.

Tokita [Fre92] propose un régulateur auto-ajustable pour la commande d'un bras de robot manipulateur, où un réseau de neurones est utilisé pour l'estimation des paramètres, définissant le système utilisé. Ces paramètres sont utilisés pour identifier la dynamique du système, et prédire ses sorties futures. Le régulateur est basé sur la minimisation de l'erreur de prédiction.

[Che90] propose, d'une manière générale, un processus nonlinéaire commandé par un régulateur auto-ajustable, utilisant les réseaux de neurones. Le régulateur est « ordinaire », et le réseau sert à estimer les paramètres utilisés par ce régulateur.

Le système, pour lequel cette stratégie est utilisée est nonlinéaire et est sous la forme compagne.

$$y(k+1) = f(y(k), y(k-1), \dots, y(k-n), u(k-1), u(k-2), \dots, u(k-n)) + g(y(k), y(k-1), \dots, y(k-n), u(k), u(k-1), \dots, u(k-n))u(k) \quad (3.6)$$

Si les fonctions  $f$  et  $g$  sont connues, pour que la sortie du système  $y(k+1)$  suive la sortie désirée  $d(k+1)$ , il faut utiliser une commande de la forme :

$$u(k) = -\frac{f(y(k))}{g(y(k))} + \frac{d(k+1)}{g(y(k))} = \frac{d(k+1) - f(y(k), u(k-1))}{g(y(k), u(k-1))} \quad (3.7)$$

Comme les fonctions  $f$  et  $g$  ne sont pas connues en général, on utilise deux réseaux de neurones pour les estimer. L'un  $\hat{f}$  avec des poids synaptiques  $W$ , et l'autre  $\hat{g}$  avec des poids synaptiques  $V$ .

$$\hat{y}(k+1) = \hat{f}(y(k), W(k)) + \hat{g}(y(k), V(k))u(k) \quad (3.8)$$

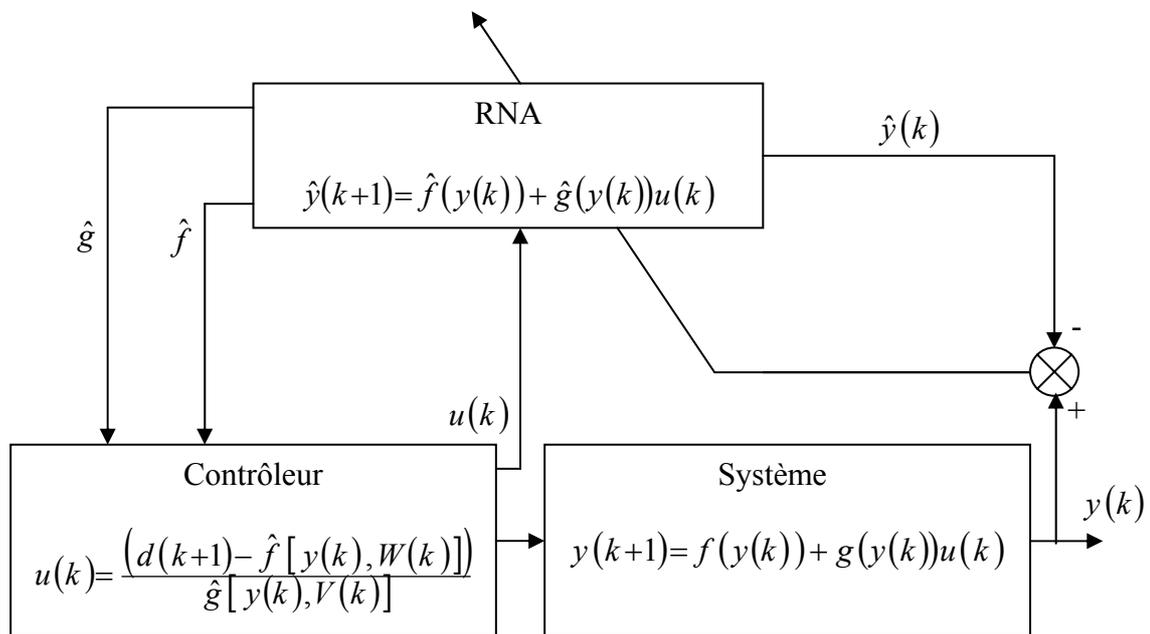


Figure 3.9 : Régulateur Auto-ajustable utilisant les réseaux de neurones

Si on suppose que l'identification a été correcte, pour que le système poursuive la sortie désirée  $d(k)$ , la commande à appliquer au système et au modèle doit, donc être de la forme :

$$u(k) = \frac{\hat{f}[y(k), W(k)]}{\hat{g}[y(k), V(k)]} + \frac{d(k+1)}{\hat{g}[y(k), V(k)]} \quad (3.9)$$

A partir de (3.6) et (3.7), la sortie du système sera de la forme

$$y(k+1) = f(y(k)) + g(k) \left\{ \frac{\hat{f}[y(k), W(k)]}{\hat{g}[y(k), V(k)]} + \frac{d(k+1)}{\hat{g}[y(k), V(k)]} \right\} \quad (3.10)$$

Les poids synaptiques  $V(k)$  et  $W(k)$  peuvent être réadaptés en minimisant l'erreur :

$$E(k) = (d(k+1) - y(k+1))^2 \quad (3.11)$$

Pour cela, l'algorithme de rétropropagation est utilisé. Dans la réadaptation des poids, la fonction  $g(k)$  inconnue est remplacée par son signe pour le calcul des dérivées [Che90].

### 3.4.3.2 Commande adaptative avec modèle de référence [Yed98]

Dans cette stratégie, la commande a pour rôle d'obliger le système à suivre un modèle de référence défini. Pour une commande par réseaux de neurones, ce modèle est défini par des couples entrées-sorties.

Comme dans le cas de la commande adaptative classique, deux approches peuvent être utilisées.

#### 1. Commande adaptative indirecte

Dans ce cas, les paramètres du système sont estimés à chaque instant. Les paramètres du régulateur sont calculés en supposant que les paramètres du modèle estimé coïncident avec ceux du système réel (figure 3.10). C'est ce qui est appelé en commande adaptative classique le principe d'équivalence certaine [Ast89].

La réussite d'une commande dépend étroitement de la précision avec laquelle le réseau de neurones arrive à estimer les paramètres du système

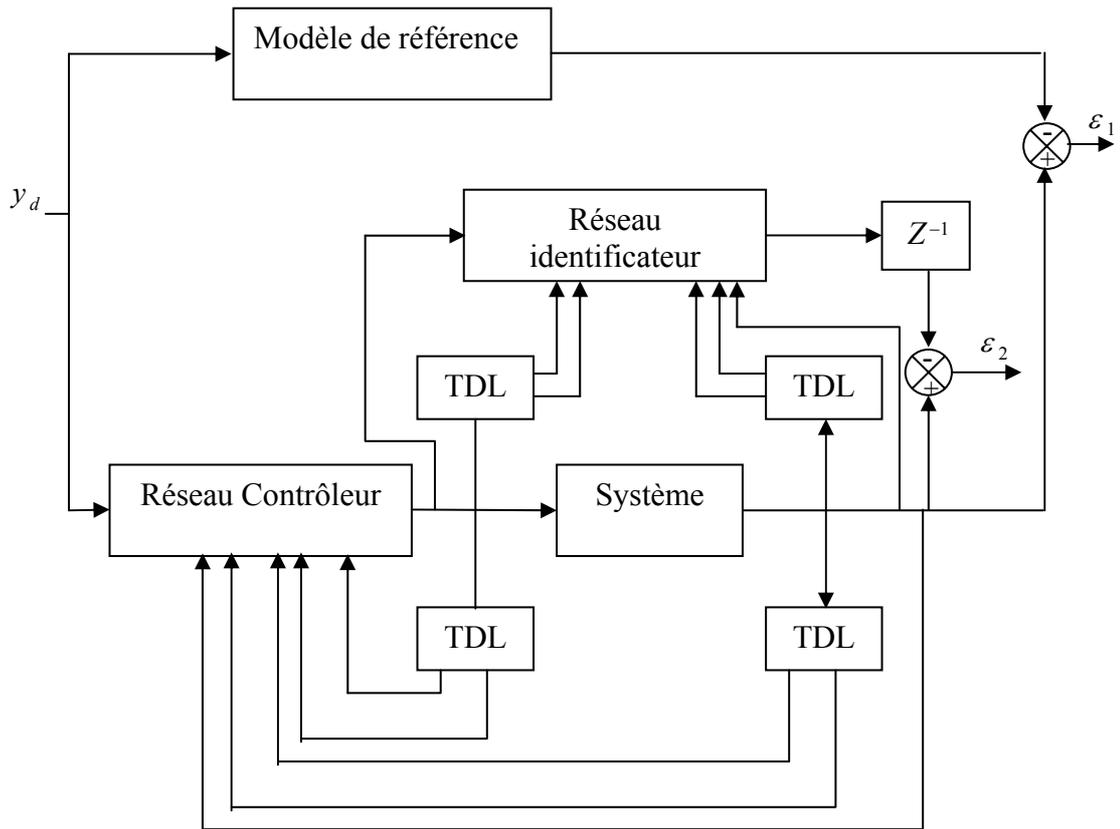


Figure 3.10 : Commande adaptative indirecte par réseaux de neurones

## 2. Commande adaptative directe

En commande adaptative directe, les paramètres du réseau contrôleur sont directement adaptés, en minimisant l'erreur en sortie entre le modèle de référence et le système (figure 3.11).

Dans le contexte neuronal, deux conditions importantes doivent être vérifiées :

- Le système doit être stable
- Les poids initiaux du réseau ne doivent pas être éloignés des valeurs qui donnent lieu à une poursuite parfaite du modèle de référence.
- La vitesse d'adaptation des poids de réseau, qui dépend du taux d'apprentissage, doit être suffisamment faible, afin de ne pas créer d'effet déstabilisateur.

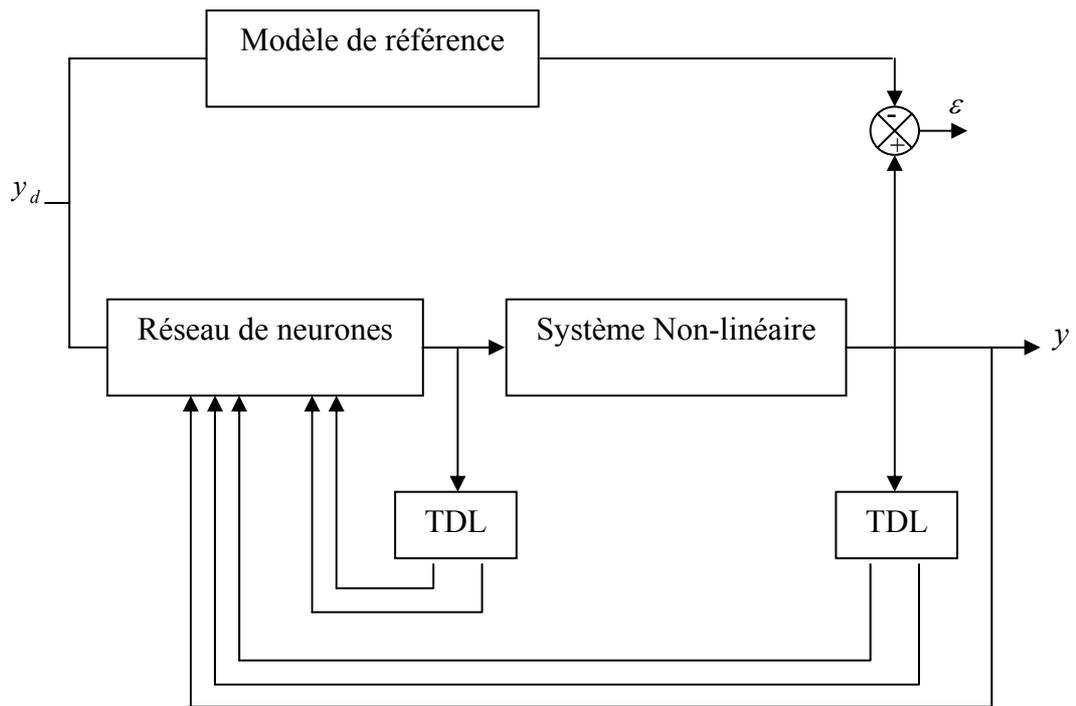


Figure 3.11 : Commande adaptative directe par réseau de neurones

### 3.5 Conclusion

Au cours de ce chapitre nous avons présenté une synthèse des différentes approches de commande utilisant les réseaux de neurones artificiels.

D'autres méthodes de commande neuronale existent, et qui n'ont pas été présentée ici, car elles sont moins utilisées et sont encore dans une étape d'exploration et d'expérimentation. Nous citerons par exemple : la commande par modèle interne.

Les techniques de commande neuronale présentent une certaine souplesse quant à la connaissance du modèle de système à commander. En effet, l'utilisation des RNA nous épargne la connaissance du modèle de système. Il suffit donc d'identifier le comportement de celui-ci à l'aide de structures d'identification appropriées, ou alors d'acquérir le savoir faire nécessaire à l'obtention des performances désirées durant la phase d'apprentissage. Le problème principal est que l'erreur en sortie du modèle est différente de l'erreur en sortie du procédé dans la commande par réseaux de neurones.

# **Chapitre 4**

**Commande d'un bioréacteur fed-batch  
et d'un bioréacteur continu**

# **Chapitre 4**

## **Commande d'un bioréacteur fed-batch et d'un bioréacteur continu**

---

### **4.1 Introduction**

Les procédés biotechnologiques sont complexes à mettre en œuvre et leur compréhension est difficile à cause de la nature de la matière vivante. Les modèles de connaissance sont en général non stationnaires, font intervenir de nombreux paramètres et sont fortement non linéaires. L'essor des biotechnologies imposent néanmoins de développer des stratégies de commande permettant de contrôler efficacement ces procédés afin d'améliorer leurs productivités.

Les applications que nous allons étudier concerne deux procédés de fermentation industrielle importants l'un est le bioréacteur semi-continu (fed-batch) et l'autre le bioréacteur continu. Une production quasi optimale de biomasse peut être obtenue en imposant de suivre une consigne de concentration en substrat, ici le glucose.

Ce chapitre est consacré à l'application et la comparaison de différentes lois de commandes, d'abord une commande classique telle que la régulation PI, ensuite une technique de commande moderne à base de réseaux de neurones artificiels (RNA), utilisant l'algorithme de retropropagation du gradient pour ajuster les paramètres de commande.

Pour sa complexité la commande d'un bioréacteur permet de tester et de comparer l'efficacité des performances de la commande classique et de la commande par RNA.

### **4.2 Description du problème de modélisation et de commande d'un bioréacteur**

Le bioréacteur est un système difficile à commander à cause de sa forte non linéarité. Comme on l'a vu au premier chapitre, sous sa forme la plus simple le bioréacteur est un

réservoir contenant de l'eau et des cellules (bactéries ou levures) qui consomment des éléments nutritifs (substrat), produisent certains produits (désirés et non désirés) et de la biomasse. Les cellules peuvent changer leurs taux de croissance et de production radicalement, suivant la température, l'humidité, etc... La dynamique du bioréacteur sera très différente suivant que l'on est en mode fed-batch (figure 4.1) ou continu (figure 4.19).

### 4.3 Modèle d'un bioréacteur fed-batch

On rappelle les équations du bioréacteur

$$\left\{ \begin{array}{l} \frac{dX}{dt} = \mu X - Xu \\ \frac{dS}{dt} = -k_1 \mu X + u(S_{in} - S) \\ \frac{dV}{dt} = uV \\ y = k_2 \mu X \end{array} \right. \quad (4.1)$$

$$\text{Avec : } \mu(S) = \frac{\mu_0 S}{K_S + S + \frac{S^2}{K_I}} \quad (4.2)$$

$k_1, k_2$  : Coefficients positifs ( $k_1 = \frac{1}{D}$ )

On considère les variables d'état comme étant, la concentration en biomasse  $X$ , la concentration en substrat  $S$ , et le volume  $V$  dans le bioréacteur. La sortie du système est la concentration de gaz  $y$  qui s'échappe du bioréacteur (ici du  $CO_2$ ). La commande appliquée au système est le taux de dilution  $u$  et la variable à réguler est la concentration en glucose  $S$ .

La fermentation est réalisée dans un bioréacteur fonctionnant en mode semi-continu [Tit03], alimenté en glucose à la concentration  $S_{in} = 20 \text{ g/l}$  (figure 4.1).

- Les conditions initiales sont:

$$X_0 = 7.2 \text{ g/l}$$

$$S_0 = 2 \text{ g/l}$$

$$V_0 = 0.031l$$

- Les paramètres sont:

$$\mu_0 = 0.53h^{-1}$$

$$K_S = 1.2 \text{ g/l}$$

$$K_I = 0.22 \text{ g/l}$$

$$k_1 = 0.4$$

$$k_2 = 1$$

La concentration nécessaire pour une productivité optimale est :

$$S_{opt} = \sqrt{K_S K_I} = \sqrt{1.2 * 0.22} = 0.5 \text{ g/l} \quad (4.3)$$

Avec :

$$\mu_0 = \mu_{max} \left( 1 + 2 \sqrt{\frac{K_S}{K_I}} \right) \Rightarrow \mu_{max} = \frac{0.53}{\left( 1 + 2 \sqrt{\frac{1.2}{0.22}} \right)} = 0.09 h^{-1} \quad (4.4)$$

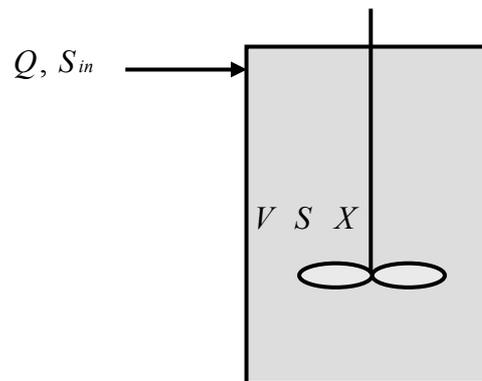


Figure 4.1 : Schéma représentatif de bioréacteur semi-continu

### 4.3.1 Comportement du système en boucle ouverte

Avant d'entamer la commande du bioréacteur, nous avons jugé utile de présenter le comportement de ce système en boucle ouverte.

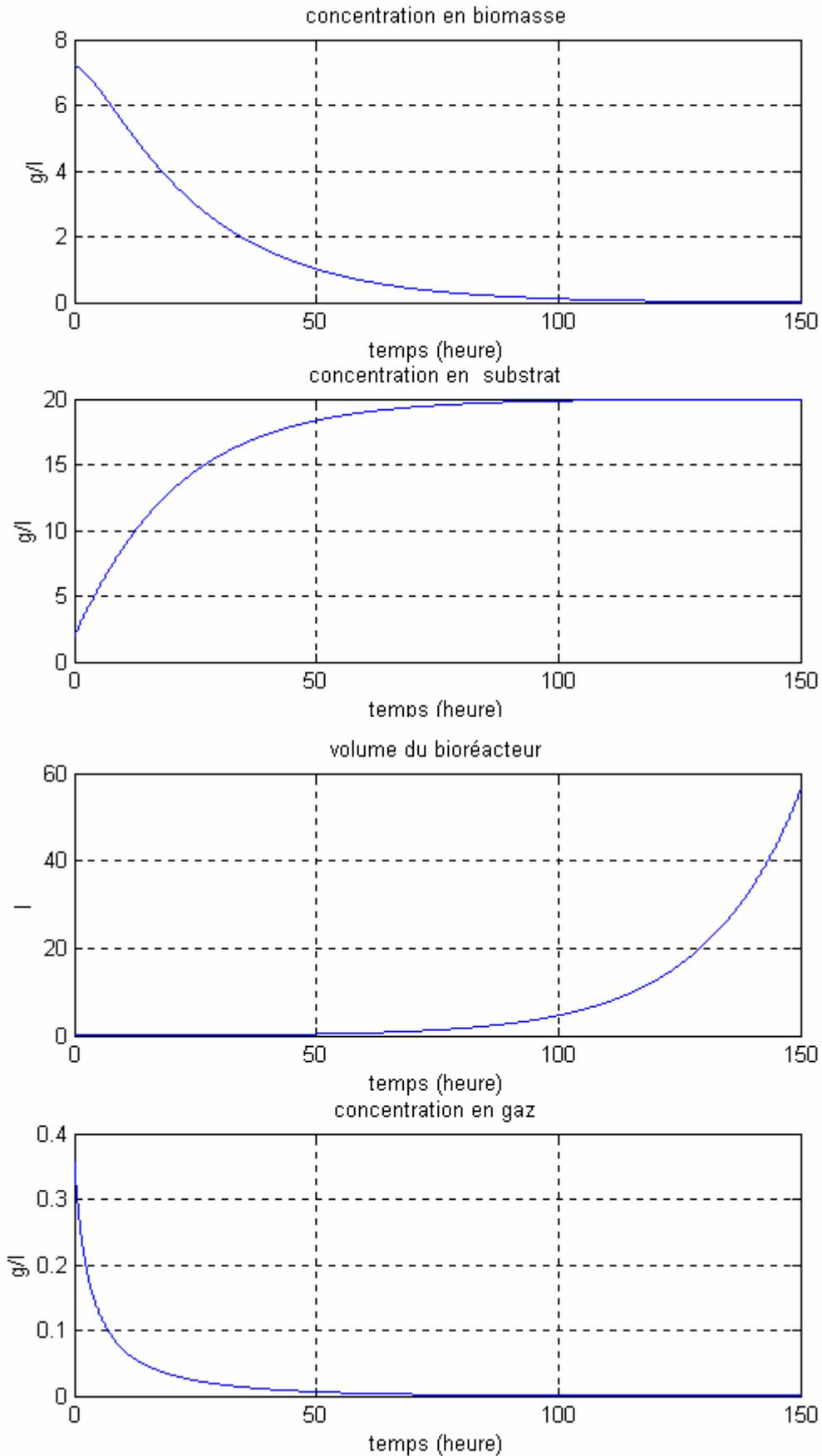


Figure 4.2 : Evolution des variables d'état pour  $u = 0.05 \text{ h}^{-1}$

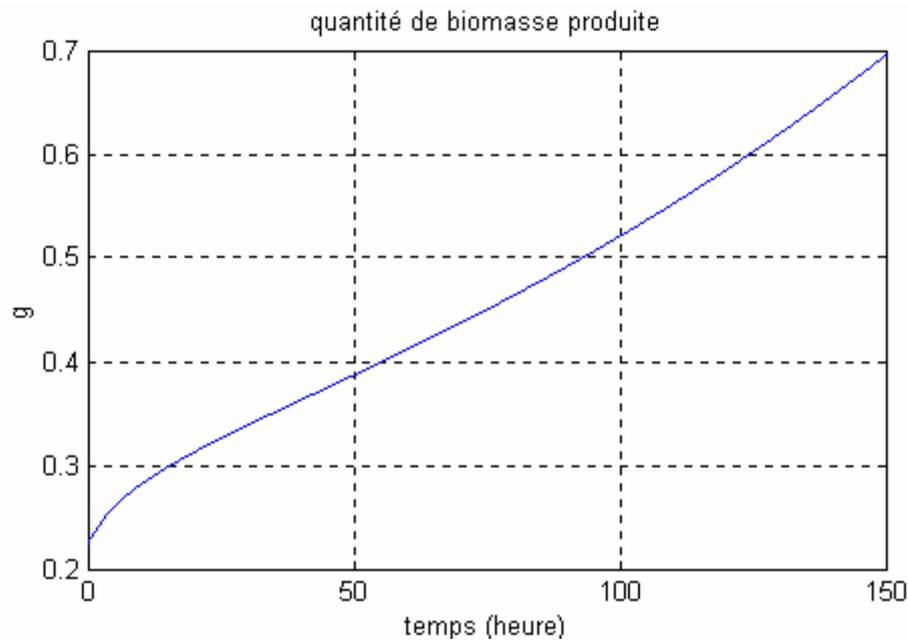


Figure 4.3 : Evolution de la quantité de biomasse  $X.V$

La (figure 4.2) montre les états du système pour une commande égale à  $0.05 h^{-1}$ . Comme  $\mu \neq u$  en général, le seul état d'équilibre stable possible est obtenu si  $\mu < u$  (4.1), l'état d'équilibre correspondant est  $X=0$  et  $S = S_{in}$ .

Le produit  $X.V$  qui représente la quantité de biomasse augmente (figure 4.3), on aura donc bien une formation de biomasse mais comme  $S = S_{in}$  correspond à une faible productivité, la quantité de biomasse augmente bien moins vite que le volume de bioréacteur et sa concentration tend vers zéro.

#### 4.4 Commande du bioréacteur en utilisant le régulateur classique

##### 4.4.1 Description de la commande PID

De nombreux problèmes de commande peuvent être traités à l'aide d'une commande PID (proportionnelle, intégrale, dérivée), tant que les exigences de performances ne sont pas élevées. L'algorithme PID est un régulateur standard pour la commande des procédés. Il est aussi la base de nombreux systèmes de commande. La version classique de l'algorithme peut s'écrire sous la forme (4.5).

$$u(t) = K_p \varepsilon(t) + K_i \int \varepsilon(t) dt + K_v \frac{d\varepsilon(t)}{dt} \quad (4.5)$$

dans laquelle  $u$  est la variable de commande,  $\varepsilon$  représente l'erreur définie par :

$$\varepsilon = y_d - y \quad (4.6)$$

$y_d$  et  $y$  étant respectivement la valeur désirée de la sortie et la valeur réelle.

Le diagramme fonctionnel pour ce type de commande est représenté par la (figure 4.4).

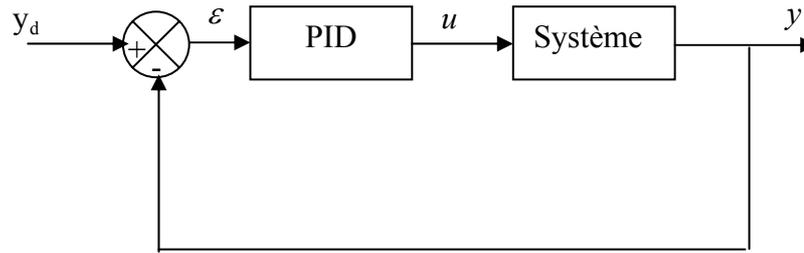


Figure 4.4 : Schéma fonctionnel de la commande PID

De nombreuses méthodes de réglage des paramètres de régulateur PID sont basées sur la réponse transitoire. La règle de Ziegler et Nichols est une méthode empirique qui permet de déterminer les paramètres du régulateur à partir de la réponse soit à un échelon de consigne, soit lors du phénomène de pompage. Dans notre cas les coefficients du régulateur sont ajustés au vu des résultats obtenus en simulation.

#### 4.4.2 Application d'un régulateur PI à un procédé semi-continu de fermentation

Pour la commande d'un procédé semi-continu, nous avons considéré un régulateur PI de la forme :

$$u(t) = K_p \varepsilon(t) + K_i \int \varepsilon(t) dt \quad (4.7)$$

Une action dérivée ne nous a pas semblé nécessaire car la réponse n'est pas oscillatoire.

Le choix des paramètres de PI va nous permettre d'obtenir les réponses désirées. Pour cela, nous avons opté pour les valeurs suivantes :

$$K_p = 6,$$

$$K_i = 0.01$$

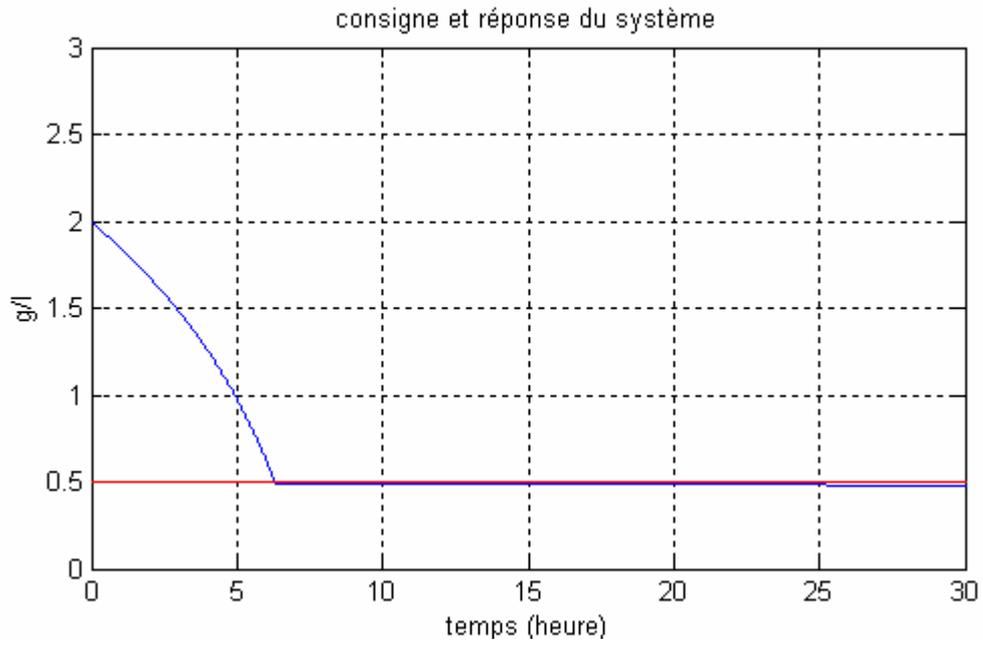


Figure 4.5 : Réponse du système pour une consigne égale 0.5 g / l

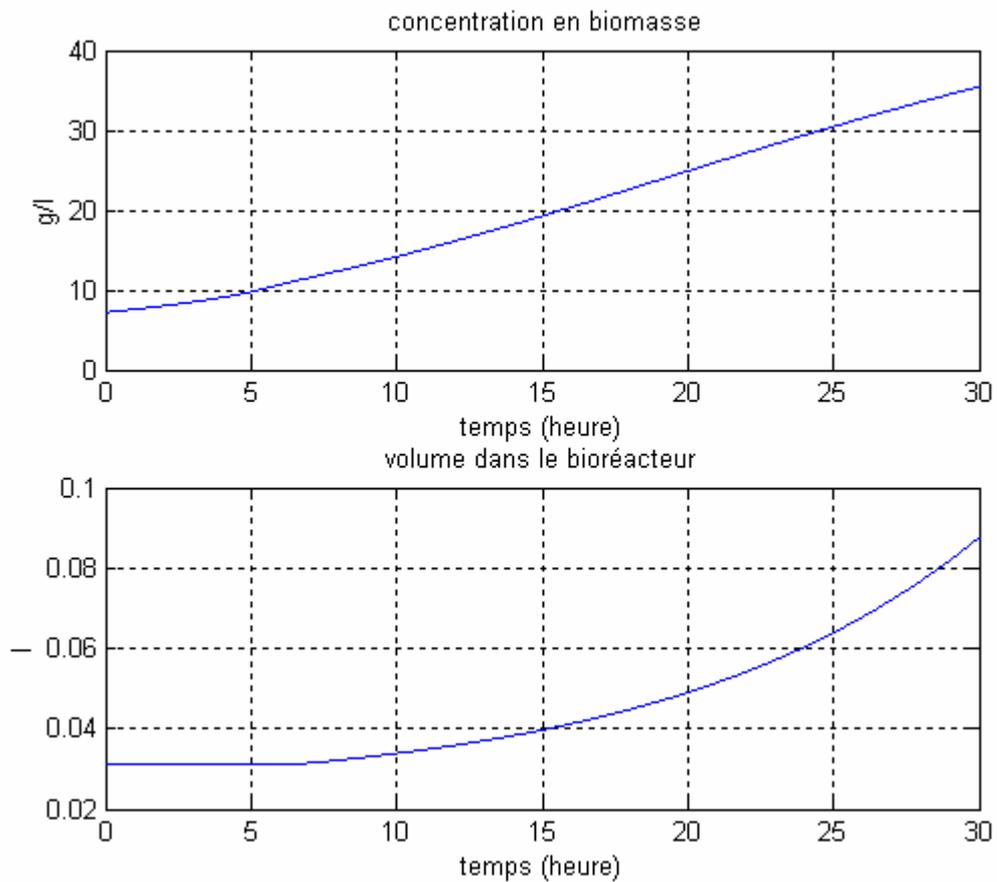


Figure 4.6: Evolution de la concentration en biomasse et du volume du bioréacteur

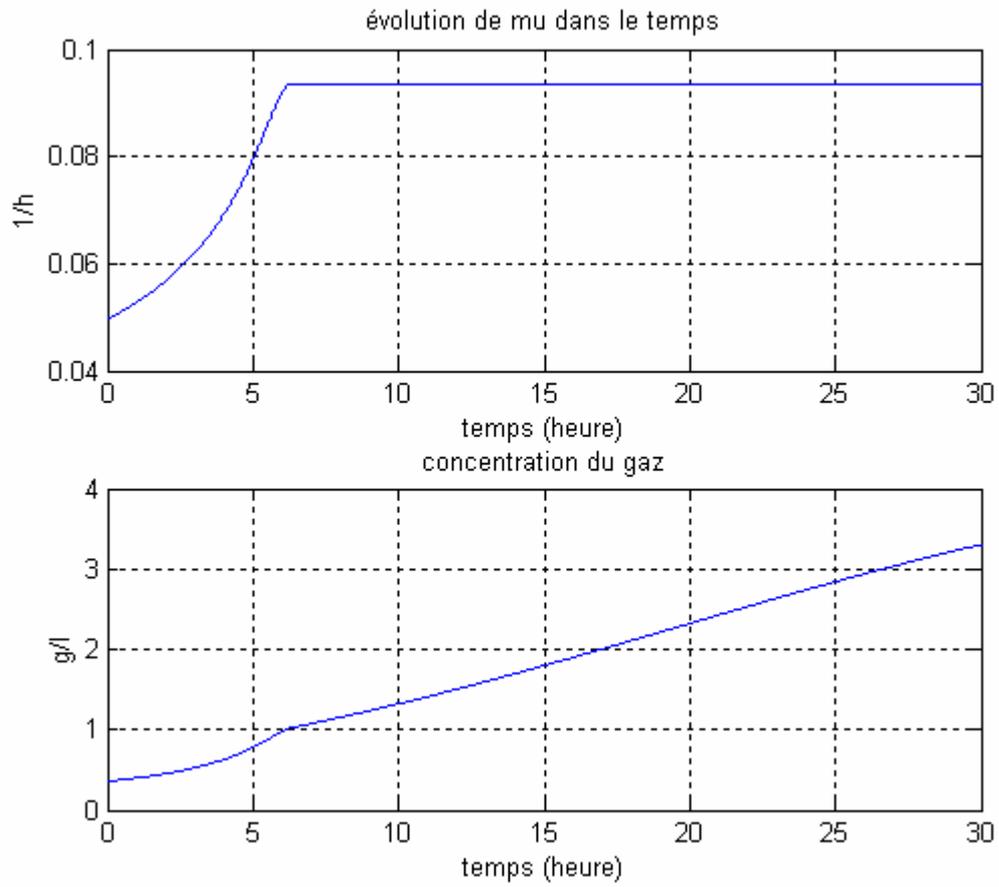


Figure 4.7: Evolution du taux de croissance et de la concentration en gaz

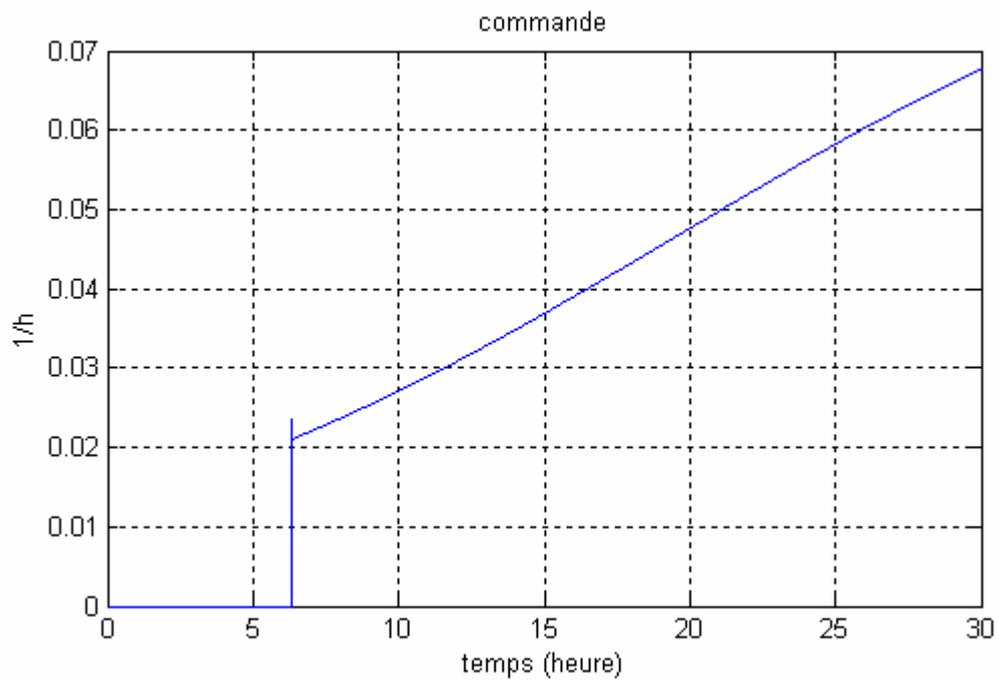


Figure 4.8: Evolution du taux de dilution

La (figure 4.5) montre l'évolution de la concentration de substrat pour une valeur de consigne égale à  $0.5 \text{ g/l}$ . On remarque que la sortie tend vers la consigne, et se stabilise autour de sa valeur après  $6 \text{ h}$ .

La (figure 4.6) présente l'évolution de la concentration en biomasse, on voit qu'elle croît de manière exponentielle. La référence en substrat atteignant rapidement une valeur qui correspond à un taux d'accroissement  $\mu$  optimal d'après le modèle de Haldane, nous pouvons dire que l'accroissement de biomasse est effectivement optimal.

Le volume du bioréacteur reste constant au début car, la commande  $u$  est nulle on n'alimente pas en substrat la biomasse afin de diminuer rapidement la concentration en substrat  $S$  pour obtenir  $S = S_{opt}$  (figure 4.8), on alimente ensuite le bioréacteur en substrat et on aura ainsi une augmentation exponentielle du volume.

La (figure 4.7) nous montre la concentration du gaz en sortie. Elle croît d'une façon quasi-linéaire à partir de  $6 \text{ h}$  quand le taux de croissance  $\mu$  est constant.

La (figure 4.9) montre la réponse à une variation de consigne de  $0.5 \text{ g/l}$  à  $1 \text{ g/l}$ , on remarque la décroissance de la concentration de substrat de la valeur initiale  $2 \text{ g/l}$  pour atteindre la valeur désirée et ce en à peu près  $5 \text{ h}$ .

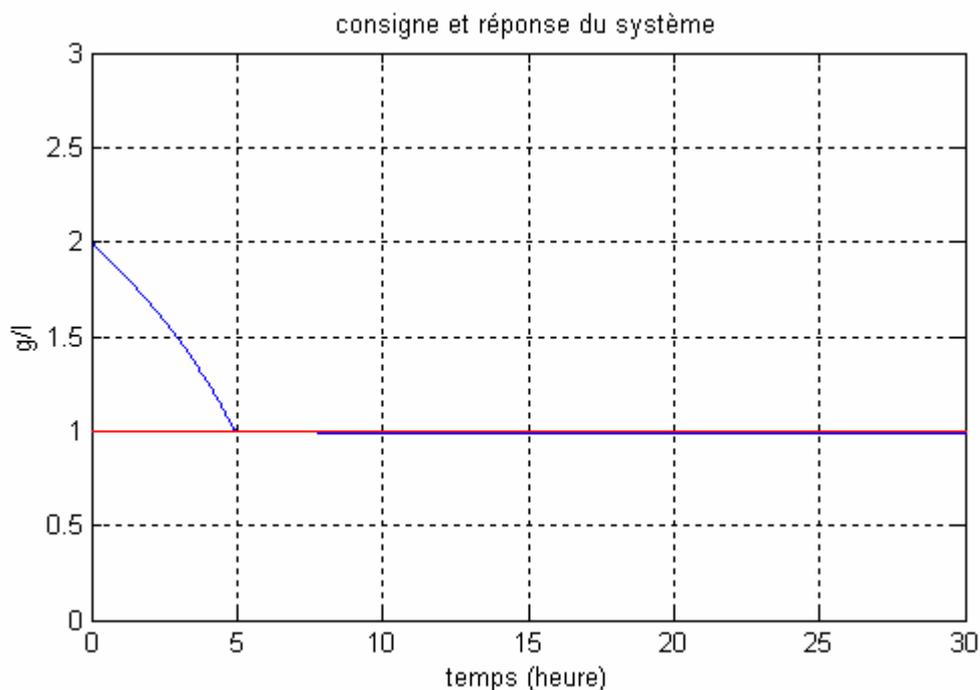


Figure 4.9 : Réponse du système pour une consigne égale à  $1 \text{ g/l}$

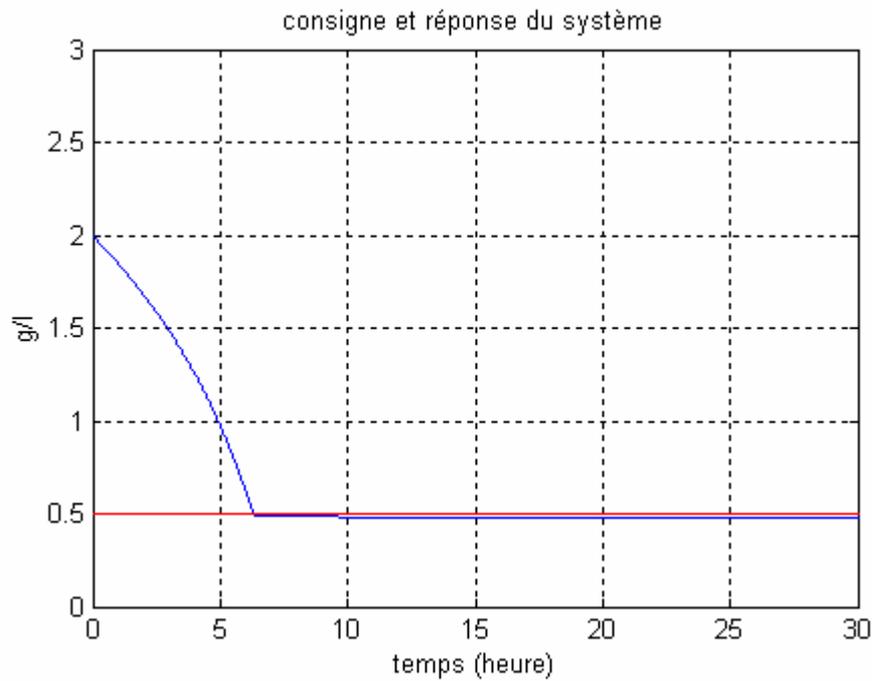


Figure 4.10 : Réponse du système pour  $S_{in}$  égal à 10 g/l

Afin de tester la robustesse de la commande par rapport aux variations de paramètres. On fait passer la concentration du substrat d'alimentation  $S_{in}$  de 20g/l à 10g/l. On constate d'après la (figure 4.10) un bon suivi de consigne.

#### 4.5 Commande neuronale du bioréacteur fed-batch

La commande neuronale consiste en l'introduction d'un réseau de neurone. Ce réseau est le neurocontrôleur. La stratégie d'apprentissage du contrôleur est celle de l'apprentissage direct par estimation de l'erreur que nous avons présenté en détail dans le chapitre 3. Cette commande se prête bien à la résolution de ce problème.

Le neurocontrôleur fournit l'action de commande  $u$  nécessaire pour obtenir la valeur désirée du substrat.

L'architecture d'apprentissage du neurocontrôleur est présentée dans la (figure 4.11). Une fois l'apprentissage effectué, on gèlera les poids synaptiques du neurocontrôleur (figure 4.12).

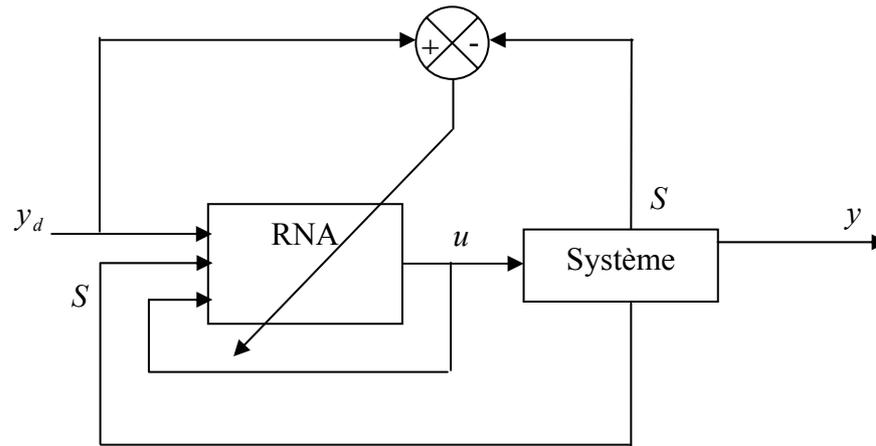


Figure 4.11 : Architecture d'apprentissage du neurocontrôleur pour la commande du bioréacteur

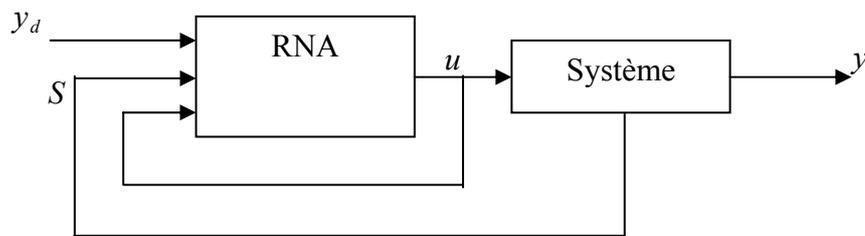


Figure 4.12 : Phase opérationnelle

Chaque réseau a une performance différente, en fonction du nombre de couches cachées et du nombre de neurones par couches. Le problème est de trouver celui qui donne les meilleurs résultats. Pour cela, plusieurs essais doivent être effectués afin de déterminer l'architecture optimale du réseau.

Après plusieurs essais, le réseau neuronal est choisi comme suit :

- Il possède trois entrées  $(S, u, y_d)$
- Il possède une seule couche cachée qui contient 9 neurones. Chaque neurone possède une fonction d'activation du type tangente hyperbolique.
- Un neurone de sortie qui possède une fonction d'activation linéaire.
- Le neurocontrôleur a une période d'échantillonnage égale à  $T=0.01h$ .

### 4.5.1 Phase de test

Il est nécessaire de valider le neurocontrôleur avec une base de données différentes de celle utilisée pour l'apprentissage et ce afin d'évaluer les performances de notre système. Pour cela nous avons testé notre réseau de neurone avec deux consignes différentes, la première égale à  $0.5 \text{ g/l}$  et la deuxième égale à  $1 \text{ g/l}$ . Les figures ci-dessous montre les différents états du système, ainsi que la commande  $u$ .

### 4.5.2 Consigne égale à $0.5 \text{ g/l}$

Nous testons notre neurocontrôleur pour une consigne égale à  $0.5 \text{ g/l}$ .

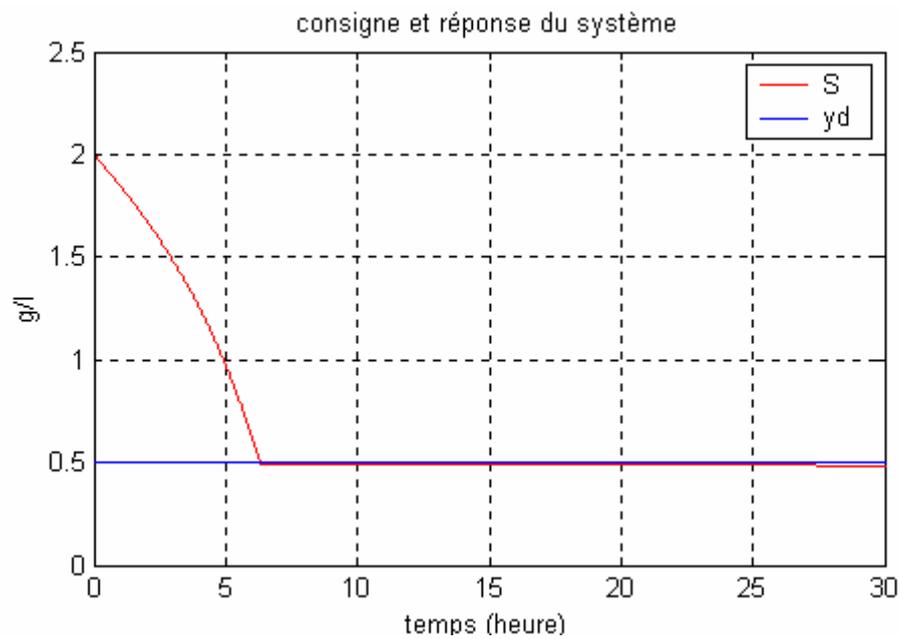


Figure 4.13 : Réponse du système pour une consigne égale à  $0.5 \text{ g/l}$

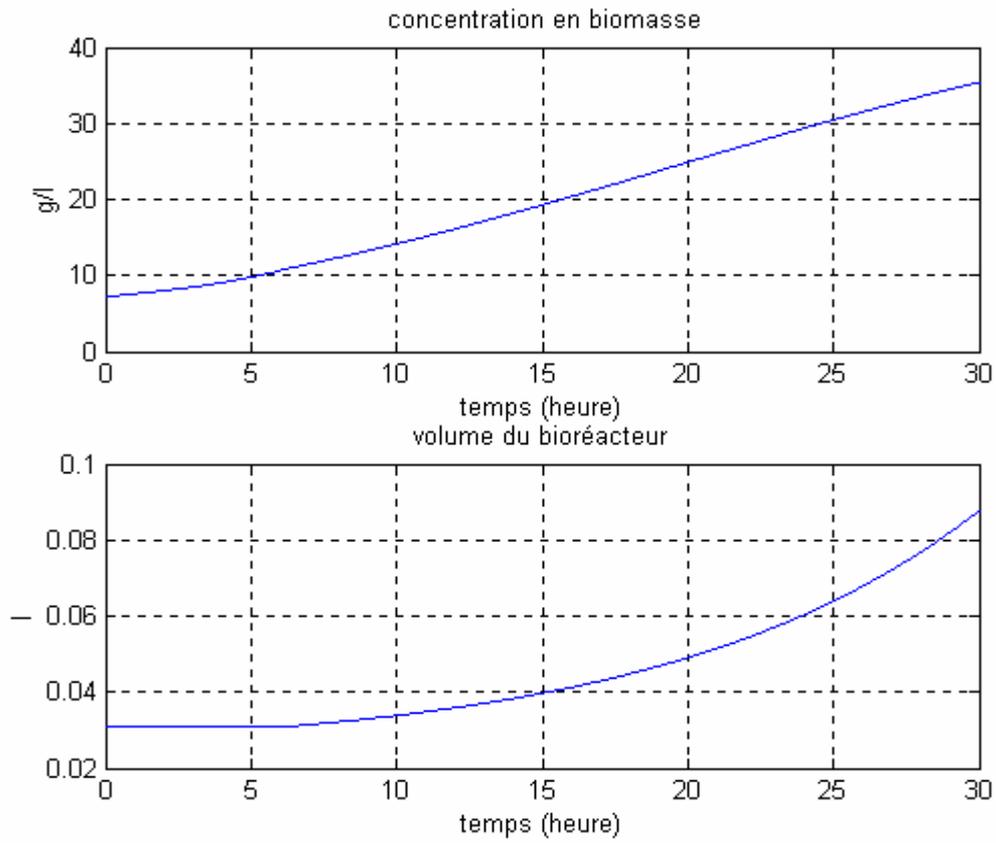


Figure 4.14 : Evolution de la concentration en biomasse et du volume du bioréacteur

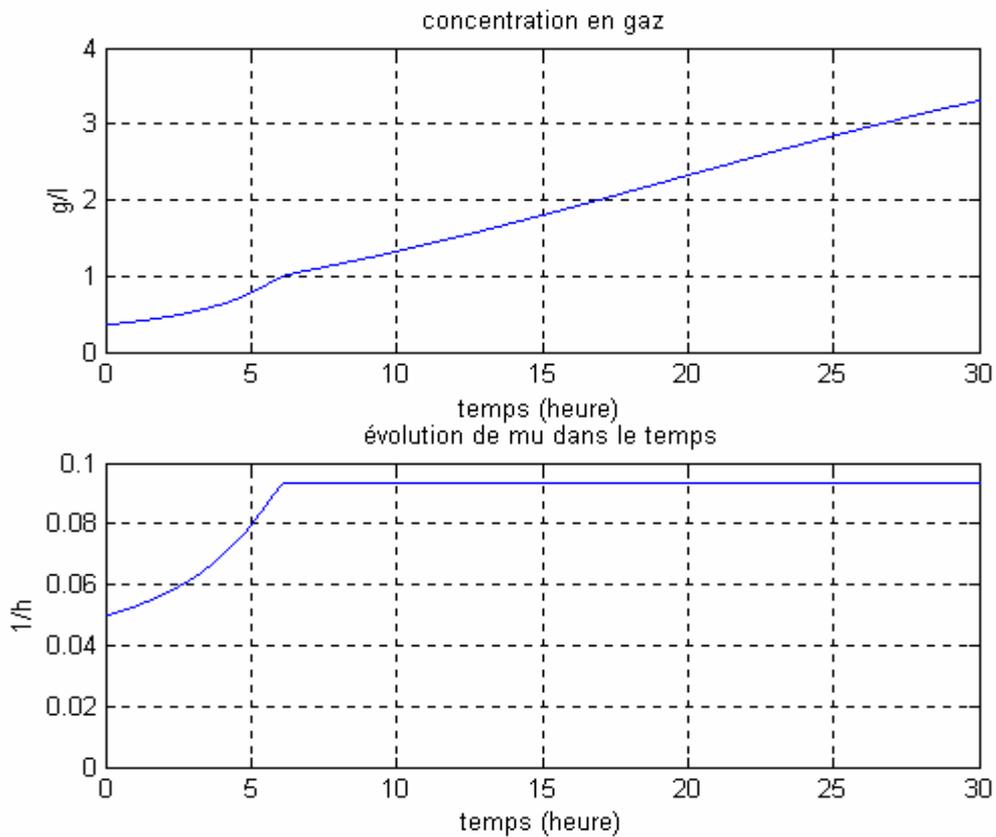


Figure 4.15 : Evolution de la concentration en gaz et du taux de croissance

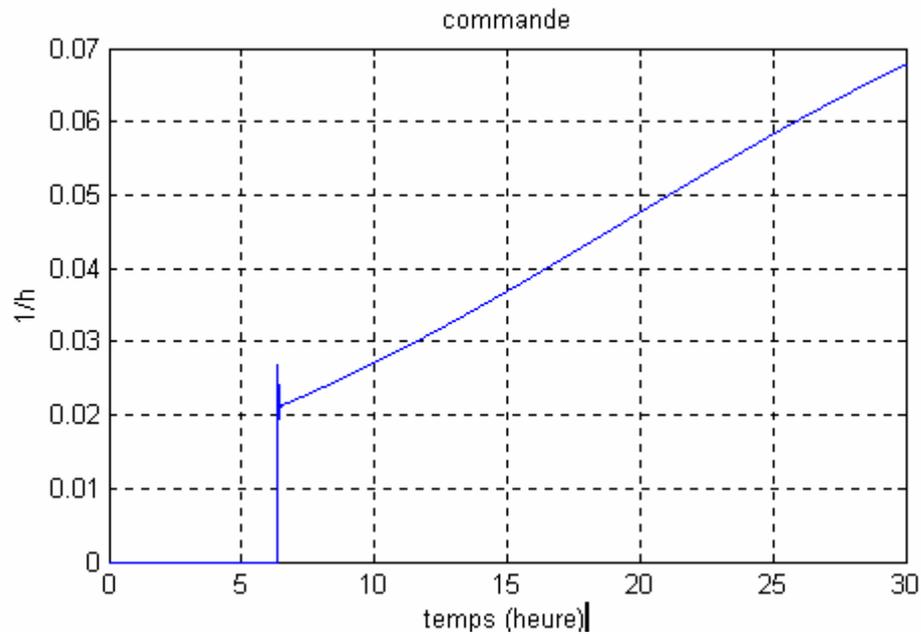


Figure 4.16 : Evolution du taux de dilution

La (figure 4.13) montre le bon suivi de la consigne imposée au système. En particulier on note que la consigne est atteinte (après 6h).

La concentration en biomasse augmente d'une manière quasi-linéaire (figure 4.14). On peut constater que la commande neuronale va permettre de produire au bout de 30h une concentration en biomasse égale à 35.4035 g/l.

La concentration en substrat du milieu étant maintenue fixe grâce à la régulation de substrat, le taux de croissance (figure 4.15), reste constant égal à  $0.0933h^{-1}$  et la concentration du gaz évolue de façon quasi-linéaire (figure 4.15).

#### 4.5.3 Consigne égale à 1 g/l

Afin de vérifier l'efficacité de cette commande, un autre test est effectué sur le contrôleur neuronal en lui imposant une consigne égale à 1 g/l.

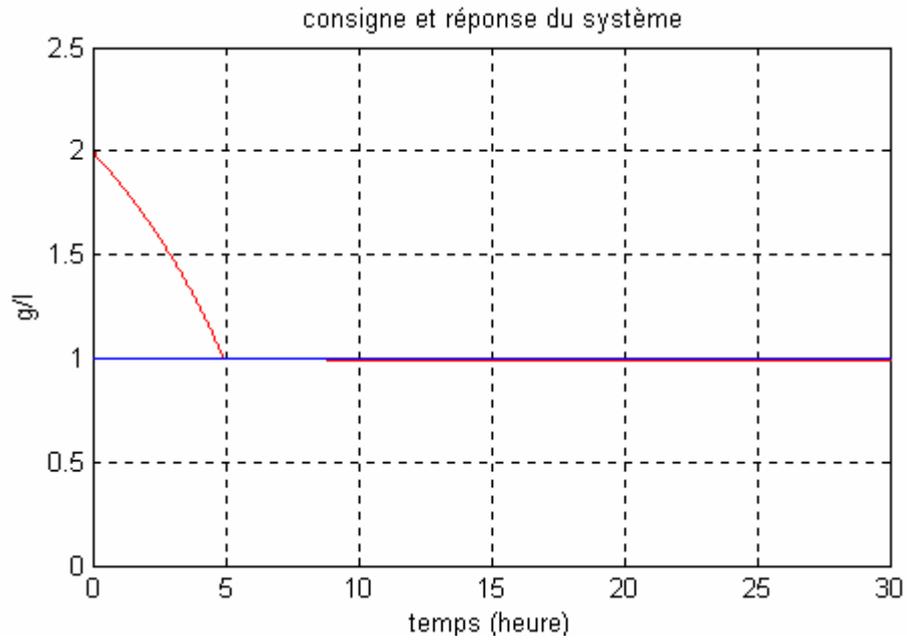


Figure 4.17 : Réponse à un échelon unité

La (figure 4.17) montre la réponse pour une consigne égale à 1 g/l. On constate qu'après 5 h la concentration de substrat atteint sa valeur désirée.

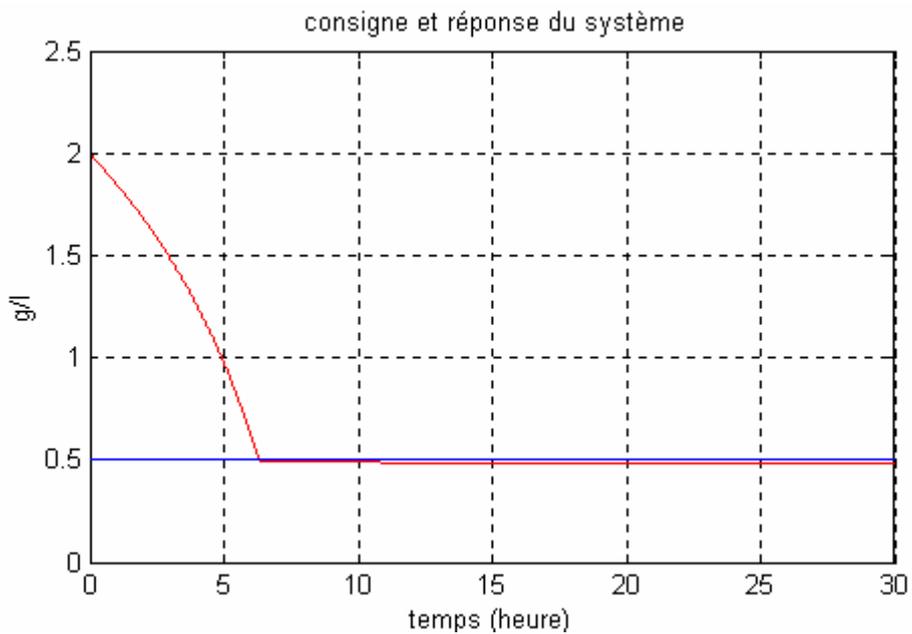


Figure 4.18 : Réponse du système pour  $S_{in}$  égal à 10 g/l

Pour effectuer un test de robustesse, on modifie la concentration en substrat d'alimentation  $S_{in}$  de 20g/l à 10g/l. La (figure 4.18) montre que le suivi de consigne est respecté.

#### 4.6 Commande d'un bioréacteur continu

La version la plus simple du problème est un réservoir à flux continu dans lequel la croissance de la biomasse dépend uniquement des éléments nutritifs apportés au système. La grandeur à contrôler est la concentration du substrat dans le réservoir (figure 4.19). Un tel bioréacteur est régi par le système d'équations différentielles (4.8).

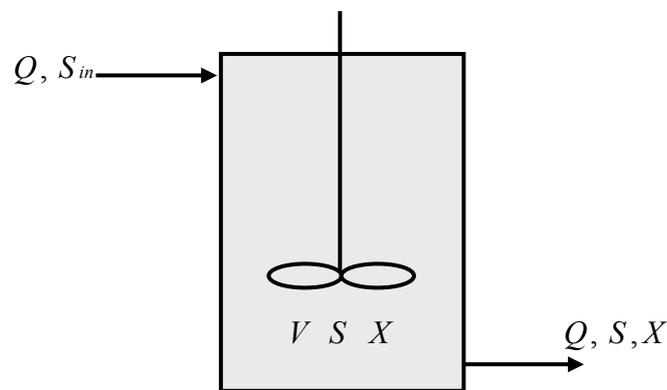


Figure 4.19 : Schéma représentatif du bioréacteur continu

##### 4.6.1 Modèle d'un bioréacteur continu

On rappelle le modèle du bioréacteur. Le modèle [Tha00] est donné par les équations suivantes :

$$\begin{cases} \frac{dX}{dt} = \mu X - X u \\ \frac{dS}{dt} = -k \mu X + u (S_{in} - S) \end{cases} \quad (4.8)$$

$$\text{Avec : } \mu(S) = \frac{\mu_0 S}{K_S + S + \frac{S^2}{K_I}}$$

- Les conditions initiales sont:

$$X_0 = 10 \text{ g/l}$$

$$S_0 = 0.2 \text{ g/l}$$

- Les paramètres sont:

$$\mu_0 = 0.3 \text{ h}^{-1}, K_S = 0.1 \text{ g/l}$$

$$K_I = 50 \text{ g/l}$$

$$S_{in} = 20 \text{ g/l}$$

$$k = 2$$

La concentration optimale de substrat  $S_{opt}$  est donnée par:

$$S_{opt} = \sqrt{K_S K_I} = \sqrt{0.1 * 50} = 2.23 \text{ g/l} \quad (4.9)$$

Avec:

$$\mu_0 = \mu_{\max} \left( 1 + 2 \sqrt{\frac{K_S}{K_I}} \right) \Rightarrow \mu_{\max} = \frac{0.3}{\left( 1 + 2 \sqrt{\frac{0.1}{50}} \right)} = 0.27 \text{ h}^{-1} \quad (4.10)$$

#### 4.6.2 Comportement du système en boucle ouverte

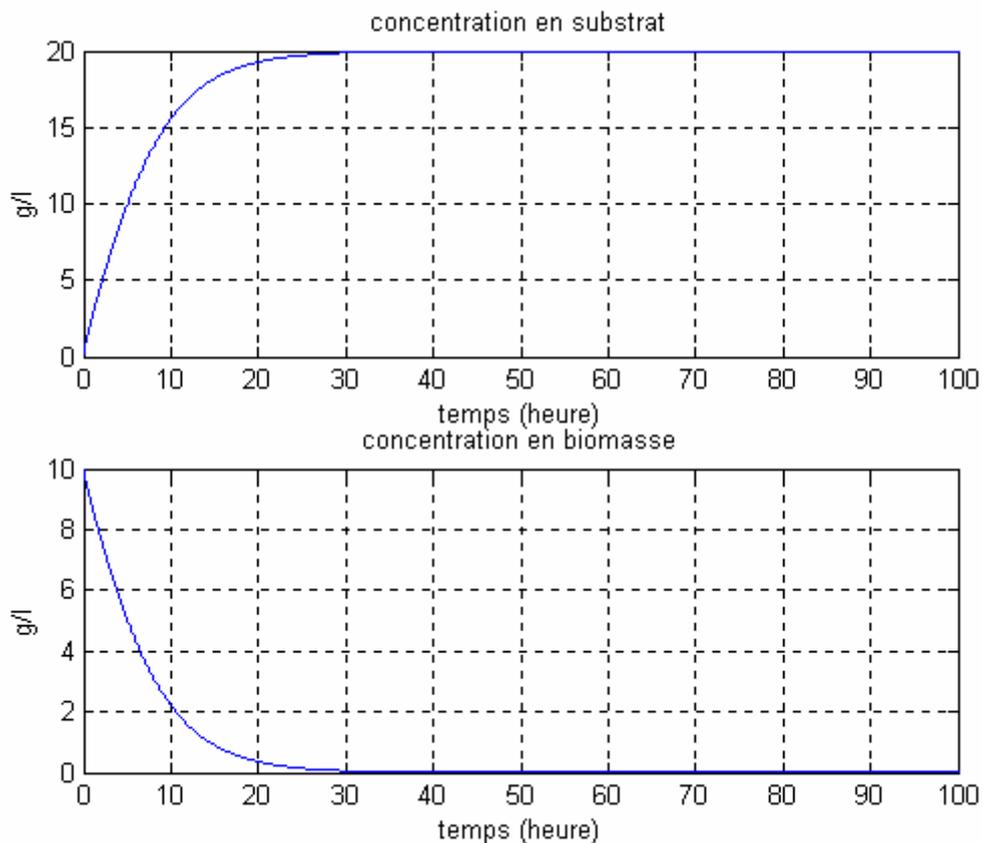


Figure 4.20: Evolution des différentes variables d'état pour  $u = 0.4 \text{ h}^{-1}$

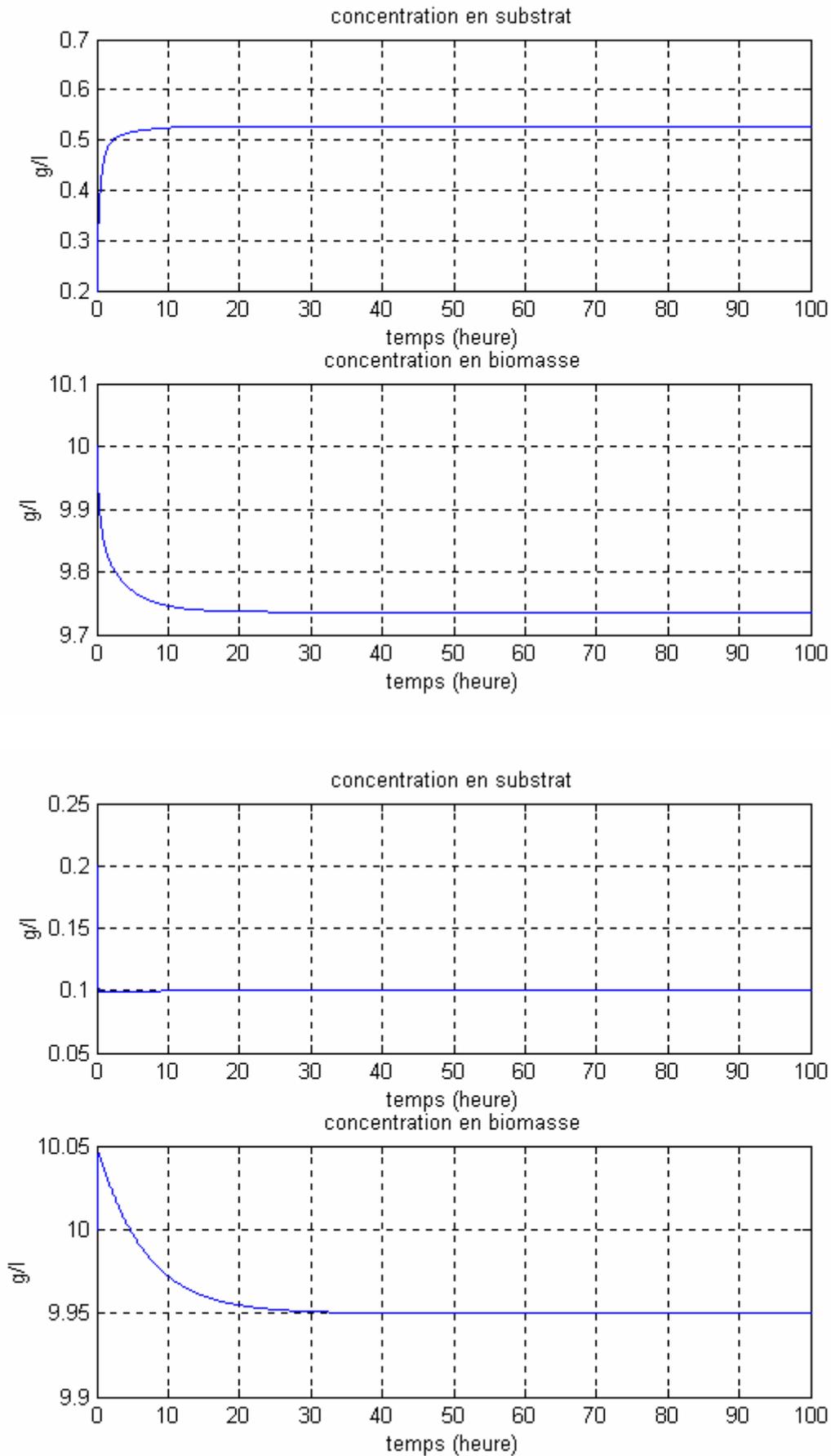


Figure 4.22: Evolution des différentes variables d'état pour  $u = 0.15 \text{ h}^{-1}$

Pour une commande  $u$  supérieure au taux de croissance ( $\mu_{\max}$ ) (figure 4.20), le seul point d'équilibre possible est  $S = S_{in}$  et  $X=0$ , comme dans le cas de bioréacteur fed-batch. La concentration en biomasse tend donc vers zéro et la concentration en substrat vers  $S_{in}$ .

Si  $u < \mu_{\max}$ , on voit qu'on a deux points d'équilibres possibles  $S_1, S_2$  d'après le modèle de Haldane (figure 4.23). En se basant sur la discussion menée en (1.9.2) on constate que seul le premier point  $S_1$  situé sur la partie montante de la caractéristique correspond à un équilibre stable.

Les (figures 4.21 et 4.22) montrent que la valeur de  $X$  et de  $S$  varie en fonction de  $u$ .

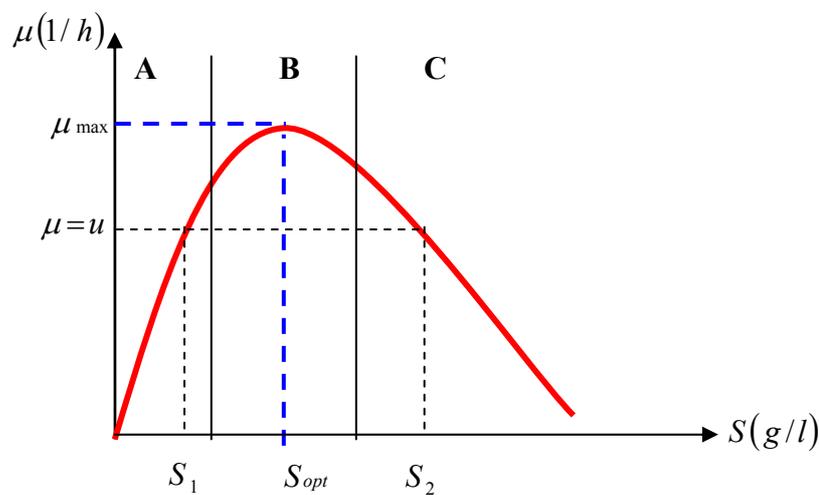


Figure 4.23 : Influence du substrat sur la croissance

#### 4.6.3 Commande du bioréacteur en utilisant le PI

Le choix des paramètres de PI va nous permettre d'obtenir les réponses désirées. Pour cela on a opté pour les valeurs suivantes :

$$K_p = 0.3$$

$$K_I = 1.5$$

La commande et la grandeur à réguler sont les mêmes que pour le bioréacteur fed-batch.

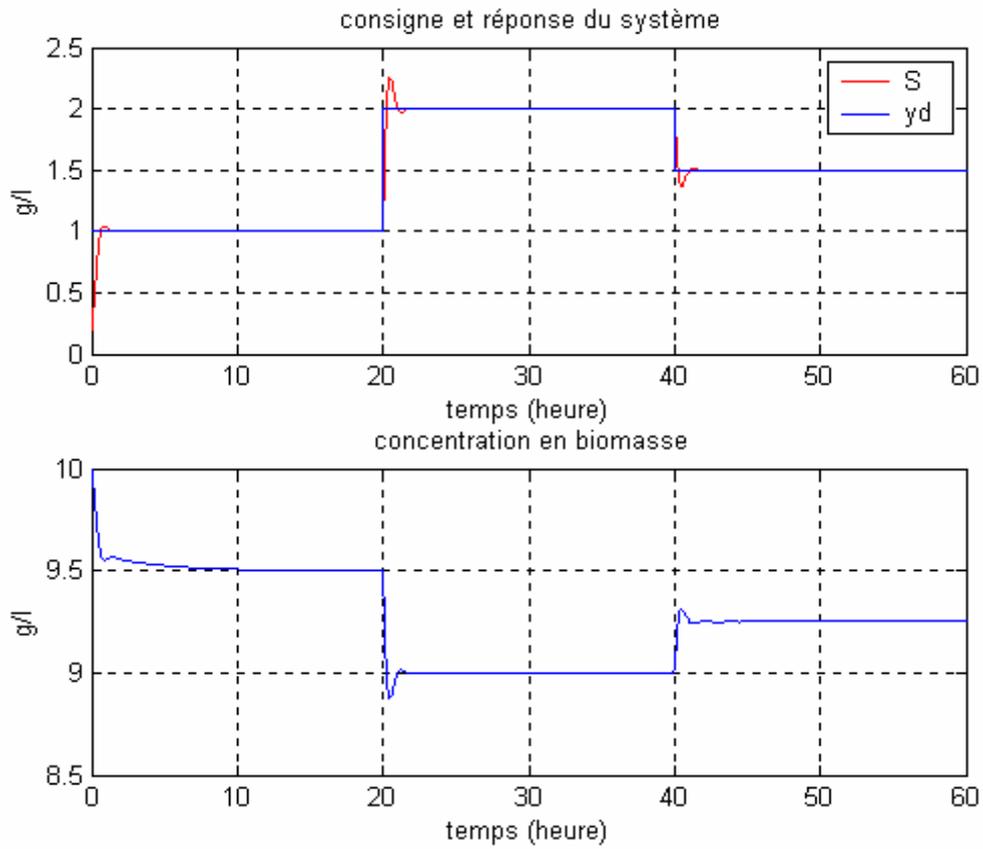


Figure 4.24 : Réponse du système et l'évolution de la concentration en biomasse

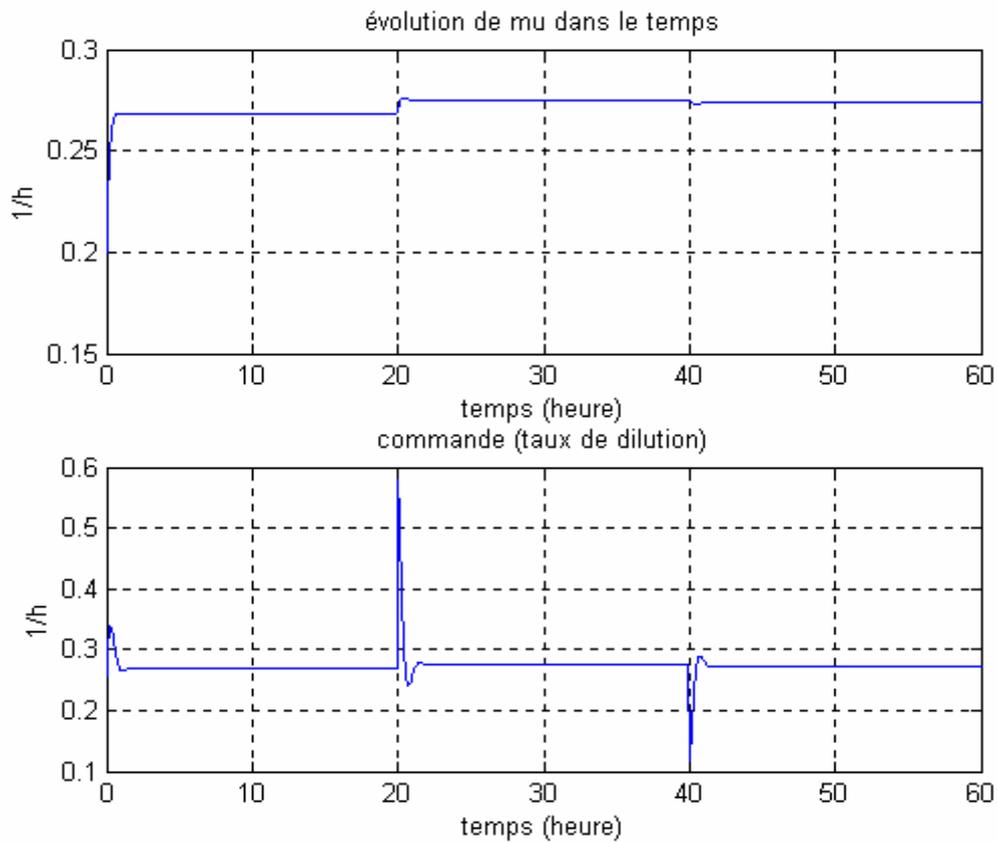


Figure 4.25 : Evolution du taux de croissance et la commande délivrée par le PI

Nous pouvons remarquer que le contrôleur PI stabilise la concentration du substrat à sa valeur désirée avec une bonne précision (figure 4.24). La concentration de la biomasse (figure 4.24) et la le taux de croissance (figure 4.25) tendent vers des valeurs constantes pour chaque consigne.

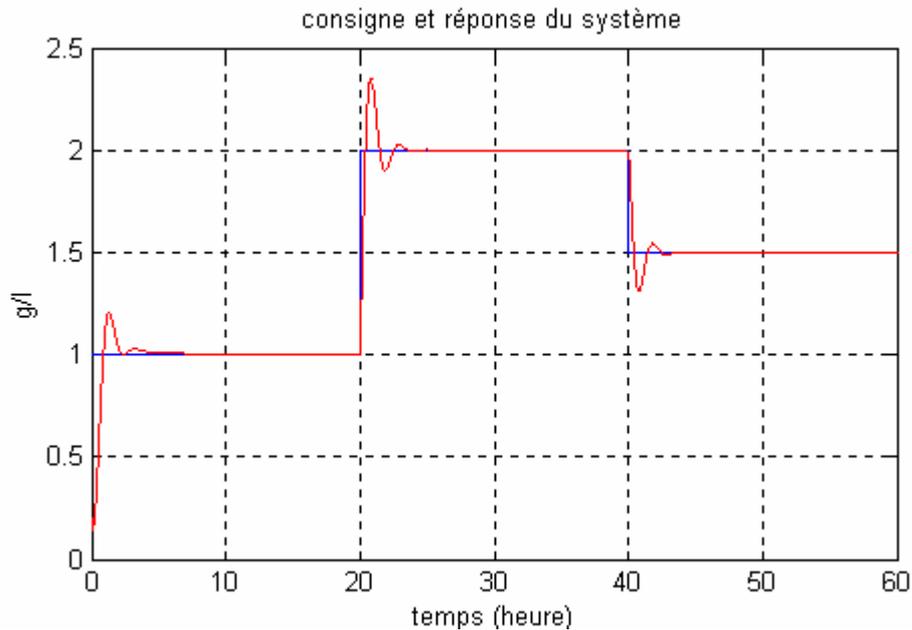


Figure 4.26 : Réponse du système pour  $S_{in}$  égal à 10 g/l

Afin de tester la robustesse du système. On fait varier  $S_{in}$  de 10 g/l à 20 g/l (figure 4.26). Les performances du système demeurent toujours acceptables.

#### 4.6.4 Commande neuronale

Le schéma de commande choisie est celui de l'apprentissage direct comme dans le cas du bioréacteur fed-batch, mais on a changé ici les grandeurs d'entrées du neurocontrôleur. On utilise toujours  $S$  et  $y_d$  mais on supprime  $u$  et on utilise  $S(t-1)$ .

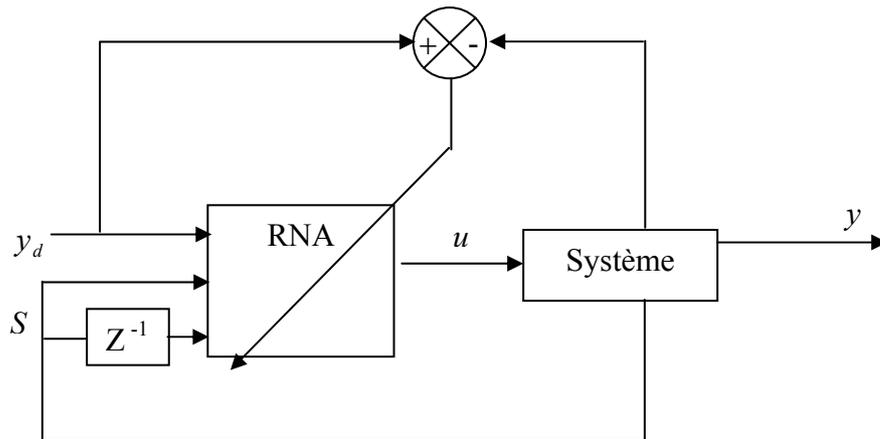


Figure 4.27 : Architecture d'apprentissage du neurocontrôleur

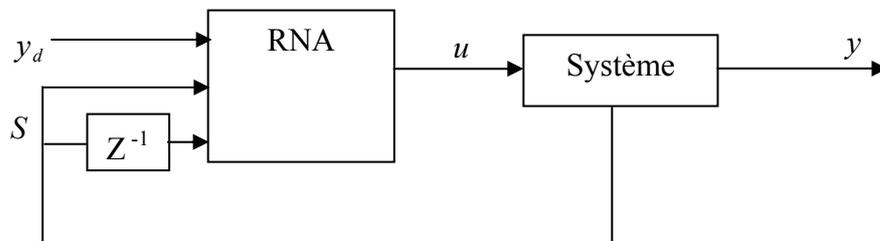


Figure 4.28 : Phase opérationnelle

Le réseau neuronal est choisi comme suit :

- Il possède trois entrées ( $S$ ,  $S(t-1)$ ,  $y_d$ )
- Il possède une seule couche cachée qui contient 9 neurones. Chaque neurone possède une fonction d'activation du type tangente hyperbolique.
- Un neurone de sortie qui possède une fonction d'activation linéaire.
- Le neurocontrôleur doit agir à la fin de chaque période  $T=0.01h$ .

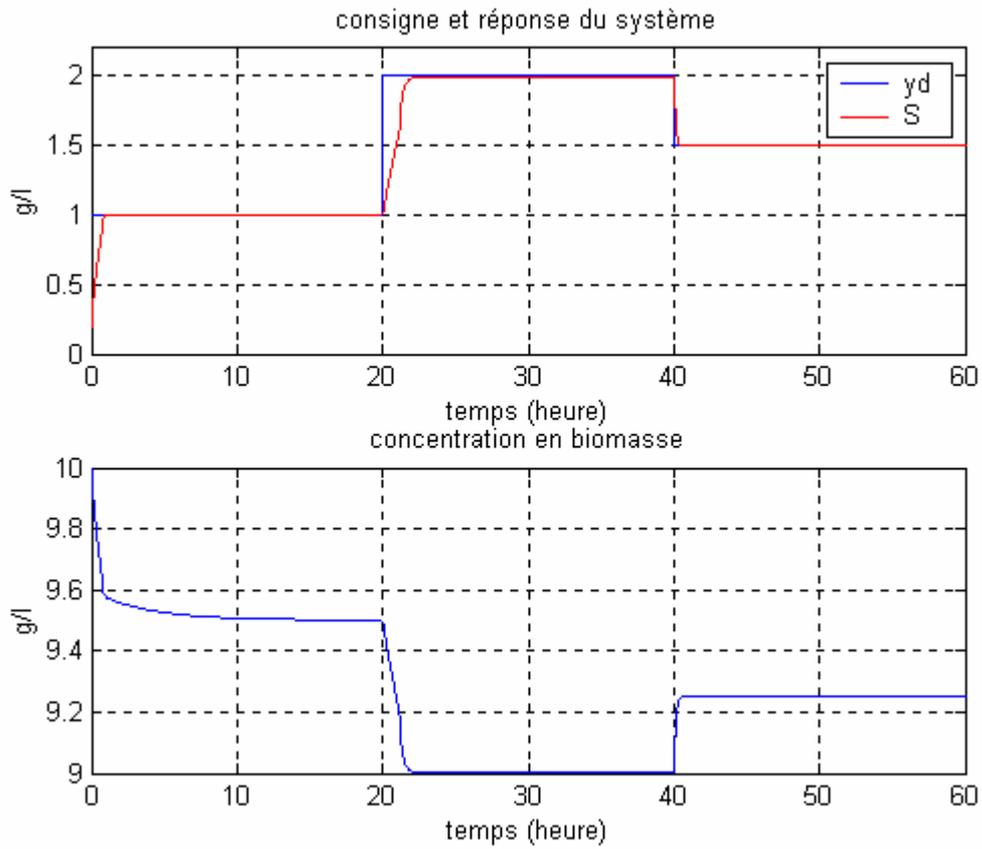


Figure 4.29 : Suivi de consigne et l'évolution de la concentration en biomasse

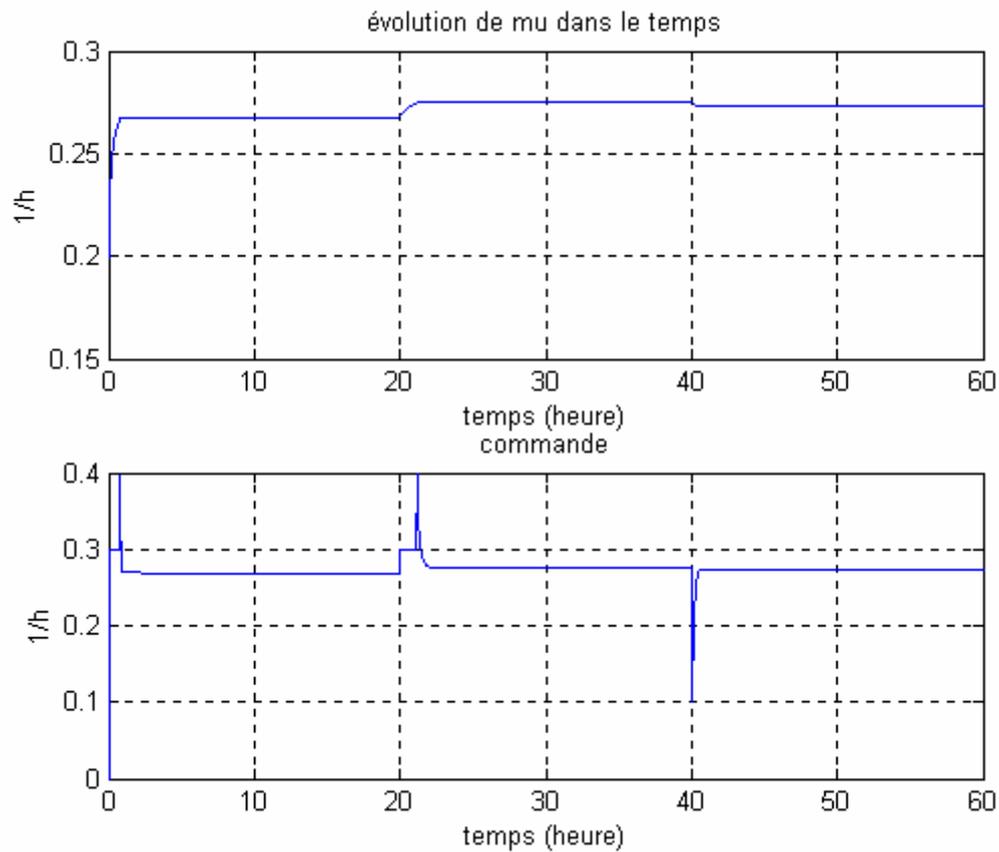


Figure 4.30 : Evolution du taux de croissance et de la commande

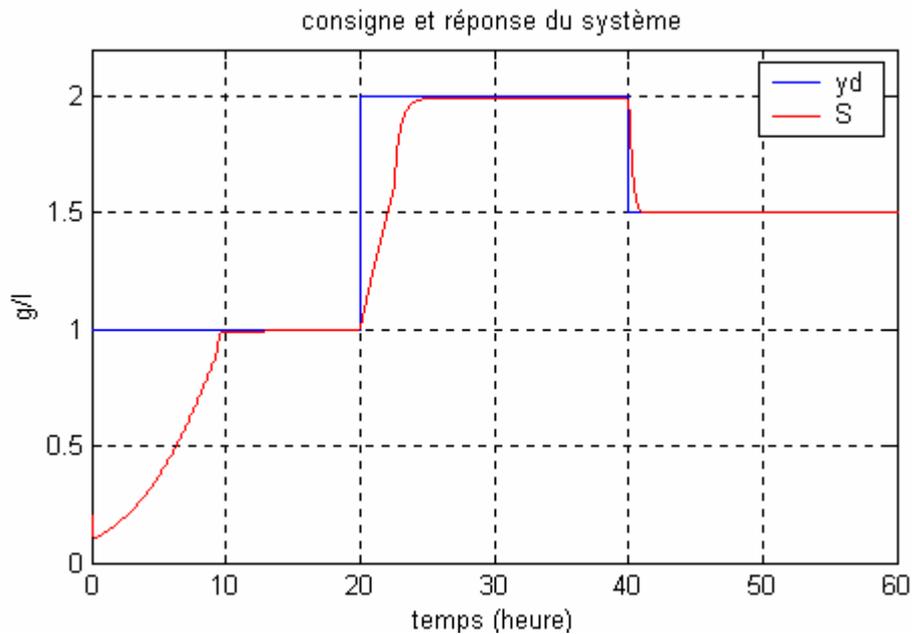


Figure 4.31 : Réponse du système pour  $S_{in}$  égal à 10 g/l

En premier lieu, on a testé l'aptitude du RNA à conduire le système vers la valeur de consigne. La (figure 4.29) montre les réponses du bioréacteurs pour trois créneaux différents d'amplitudes successivement égales à 1, 2 et 1.5 g/l. Dans les trois cas la concentration du substrat se stabilise à la valeur désirée, même chose pour la concentration en biomasse (figure 4.29) et le taux de croissance (figure 4.30).

En ce qui concerne la commande, elle présente un pic à chaque changement de consigne et se maintient ensuite (figure 4.30).

La (figure 4.31) illustre la réponse du système pour différents créneaux avec une variation de la concentration du substrat d'entrée,  $S_{in}$  à 10 g/l. On constate que la régulation est prend en charge cette variation paramétrique. Les résultats obtenus montrent l'intérêt de la commande neuronale car les réponses sont ici mieux amorties.

#### 4.7 Conclusion

Dans ce chapitre, nous avons présenté les modèles des bioréacteurs à commander (fed-batch et continu) qui sont caractérisés par trois variables d'état pour le premier et par deux variables pour le second : la concentration en substrat, concentration de biomasse et le volume pour le fed-batch .

Après avoir effectué une régulation classique, puis une commande par RNA nous constatons que la commande par RNA permet une stabilisation de la consigne, donc donne des performances satisfaisantes. Pour le bioréacteur continu, la commande par RNA les réponses sont mieux amorties.

# **Conclusion générale**

## Conclusion générale

---

---

L'objectif de notre travail a été la mise en œuvre de deux techniques de commande l'une est la commande classique PI et l'autre est la commande par réseaux de neurones artificiels, plus moderne et qui actuellement fait l'objet de nombreux travaux de recherche. Nous nous sommes penchés sur les systèmes biochimiques et plus particulièrement les bioréacteurs, un domaine qui suscite beaucoup d'intérêt à cause de leur complexité relative.

Nous avons mené dans le cadre de ce travail, une large recherche bibliographique afin d'avoir une meilleure idée sur ces nouveaux outils dont le principal intérêt réside dans la qualité d'approximateurs universels de fonctions nonlinéaires, et de connaître les différentes approches existantes concernant la commande neuronale. .

Nous avons pris deux exemples comme application, le premier est un bioréacteur fonctionnant en continu et le second en fed-batch. A partir des résultats de simulation nous avons constaté que la commande neuronale donne des résultats de régulation satisfaisants. Elle constitue donc un outil très intéressant grâce à ses propriétés d'approximation universelle de fonctions et ses capacités d'apprentissage et d'adaptation.

Ce mémoire ne constitue qu'une première étude dans le domaine des bioréacteurs qui demande à être poursuivie, en effet de nombreuses structures de commande peuvent être utilisées comme nous l'avons vu dans le chapitre 3. Les modélisations présentées dans ce mémoire laissent de côté de nombreux aspects qui n'ont pas pu être étudiés comme nous l'aurions désiré (productivité, état unique de la biomasse). Néanmoins, les systèmes de commande étudiés démontrent l'intérêt des réseaux de neurones artificiels dans le domaine de la commande des bioréacteurs.

Les systèmes de commande proposés dans ce mémoire ne pourront être réellement validés qu'une fois testés sur des bioréacteurs expérimentaux et / ou sur simulateur, mais les résultats obtenus nous rendent cependant optimiste à ce sujet.

On mentionnera finalement la possibilité d'étudier ces systèmes en utilisant la deuxième approche importante en identification et commande des systèmes non linéaires à savoir la commande floue et neuro-floue.

# *Bibliographie*

## Bibliographie

- [Aib 68] S. Aiba , M. Shoda, et M. Nagatani, *Kinetics of product inhibition in alcohol fermentation*, Biotechnol. Bioeng, Vol.10, pp.846-865, (1968).
- [And 81] J. Andrews , A mathematical model for the continuous culture of micro-organism utilizing inhibitory substrate, Biotechnol. Bioeng, Vol.23, (1981).
- [Ass 96] A. Assoum, N.E. Radi, R. Velazko, F. Elie, and R. Escoffet, *Robustness against SEU of an artificial neural space application*. IEEE Trans. on Nuclear Science, Vol.43, No.3, pp.973-978, (1996).
- [Ast 89] K. J. Astrom and W. Henmark, *Adaptive Control*, Adison Wisley Publishing Company, (1989).
- [Bak 92] W. L. Baker and J. A. Farrell, *An introduction to connectionist learning control systems*. In D. A. White and D. A. Sofge, editors, Handbook of intelligent Control, pp.35-64. Van Nostrand Reinhold, (1992).
- [Bau 89] E.B. Baum, and D. Haussier, *What Size Net Gives Valid Generalization*, Neural Computation, Vol.1, pp151-160, (1989).
- [Bou] G.Bourat, *Fermentations : propriétés et utilisations des microorganismes*, Document Technique d'ingénieur J 6002.
- [Bur 00] C. Burstein, *Biotechnologie enzymatique mode d'emploi*, Ed Polytechnica, (2000).
- [Che 90] F.Chen, *Back-propagation Neural Networks for Non linear Self Tuning Adaptive Control*, IEEE, Control Systems Magazine Vol.10, No.3, pp.44-48, April (1990).
- [Cos 97] M. Costaftis, *Comportement et contrôle des systèmes complexes. Introduction aux méthodes algébriques, qualitatives et fonctionnelles*. Diderot Editeur, arts et sciences, (1997).
- [Dan 89] P. Dantigny , Cinétique, modélisation de la croissance de saccharomyces cerevisiae, Thèse de doctorat, INPL, Nancy, (1989).
- [Duv 88] E.Duvivier, *Optimisation des procédés semi-continus de fermentation par programmation dynamique*, thèse de doctorat de l'université de Toulouse, (1988).
- [Fre 92] Freeman, *Neural Networks*, Ed Addison-Wesley, (1992).
- [Fre 92] J.A. Freeman and D.M Skapura, *Neural Networks Algorithms, Applications and programming Techniques*, Addison-Wesley Publishing Company, (1992).
- [Gau 99] E. Gauthier, *Utilisation des réseaux de neurones artificiels pour la commande d'un véhicule autonome*, thèse de l'université de Grenoble, (1999).
- [Gho 79] T.K. Ghose et R.D. Tyagi, *Rapid fermentation of cellulose hydrolyse. Product and substrate inhibition and optimization of fermentor design*, Biotechnol. Bioeng. Vol. 21, pp.1401-1420, (1979).
- [Ham 93] D. Hammerstrom, *Working With Neural Networks*, IEEE Spectrum, Vol.30, N°7, pp.46-53, (1993).

- [Hay 94] S. Haykin, *Neural Networks, A comprehensive Foundation.*, Macmillan College Publisher Corp, (1994).
- [Her 94] J. Hérault, *Réseaux Neuronaux et Traitement de Signal*, Traité des Nouvelles Technologies, Ed Technip Paris, (1994).
- [Hin 46] C.N. Hinselwood, *In chemicals kinetics of becterial cell*. Oxford Univ.Press, Londres, (1946).
- [Ier 69] N.D. Ierusalimsky, *In continuous cultivation of microorganism*, Maleck I.ed, Academic Press, Londres, (1969).
- [Jod 94] J.F. Jodouin, *Les Réseaux de Neurones, Principes et Définitions*, Ed Hermes, (1994).
- [Kaw 99] M. Kawato, *Schemes and models for control of arm trajectory*. In W.T. Miller, R.S. Sutton, and P.J. Werbos, editors, *Neural networks for control*, pp.197.228. MIT Press, (1999).
- [Kur 94] Y. Kuroe, Y. Nakai, and T. Mori, *A new neural network learning of inverse Kinematics of robot manipulator*. In Proc. of the Int. Conf. On Neural Network, Orlando (FL), (1994).
- [Lie 98] G. Lieto, *Le génie chimique à l'usage des chimistes*, Ed Tec et Doc Lavoisier, (1998).
- [Lin] G. linden, *transformation des produits alimentaires par les enzymes*, Document Techniques de l'ingénieur, F3700.
- [Mar 04] P. Marc, *Le perceptron Multicouche et Son Algorithme de Rétropropagation des Erreurs*, Université de Laval, (2004).
- [Mon 42] J. Monod, *Recherche sur la croissance des cultures microbiennes*, Ed. Herman.Paris, (1942).
- [Mya 88] H. Myamoto, M. Kawato, T. Setoyama, and R. Suzuki, *Feedback-error learning for trajectory control of a robotic manipulator*. *Neural Networks*, 1:251-265 (1988).
- [Nar 89] K.S Narendra and A.M. Annaswamy, *Stable Adaptive Systems*. Prentice Hall, Englewood, NJ, (1989).
- [Nar 90] K. S. Narendra and K. Parthasarathy, *Identification and control of dynamical systems using neural networks*, *IEEE Transactions on Neural Networks*, Vol.1, No.1, pp.4-27, (1990).
- [Non] M.Nonus, *Fermenteurs industriels conception et réalisation*. Document Techniques de l'ingénieur, J2006.
- [Que 90] I. Queinn, *automatisation d'un procédé de fermentation semi-continu : optimisation et commande*, thèse de doctorat de l'université de Toulouse, (1990).
- [Psa 87] D. Psaltis, A. Sideris and A. Yamamura, *Neural Controllers*, In Proc. of the Int. Conf. On Neural Networks, Vol.4, pp.551-558, San Diego, CA, (1987).
- [Psa 88] D. Psaltis, A. Sideris and A. Yamamura. *A multi layer neural network controller*, *IEEE, Control Systems Magazine*, Vol.8, No.2, pp.17-21, (1988)

- [Pus 93] G. V. Puskorius and L. A. Feldkamp, Implicit State Observation and Control with Recurrent Neural Networks for the Bioreactor Benchmark Problem, International Joint Conference on Neural Networks, (1993).
- [Rae 01] E. Raetz et W. Bauer, *Fermentation et technologie enzymatique*, Lausanne Semestre d'été, (2001).
- [Ren 95] J.M. Renders, *Algorithmes génétiques et réseaux de neurones*. Hermes, (1995).
- [Scr 00] R. Scriban, *biotechnologie*, Ed Tec et Doc. Lavoisier, (2000).
- [Saa 94] M. M'Saad, A. Dessaint, P. Bigros and K. Haddad, *Adaptive Versus Neural Adaptive Control: Application to robotics*. International Journal of Adaptive Control and Signal Processing. (1994).
- [Shu 90] Shun-Ichi, *A Mathematical Foundations of Neurocomputing*, Proc of the IEEE, Vol. 78, n° 9, p 1443-1463, (1990).
- [Sim 70] R. Simon, R. Meunier, *microbiologie industrielle et génie biochimique*. Ed Masson et Cie, (1970).
- [Tit 03] M. Titica, D. Dochain, M. Guay, *Real time optimisation of fed- batch bioreactors via adaptive extremum-seeking control*, 9<sup>ème</sup> congrès de la SFGP; Saint Nazaire, (2003)
- [Tha 00] H.J.Tham, M.A.Hussain, K.B.Ramachandran, *Variable Structure Control for a Continuous Bioreactor*, IEEE Tencn 2000 Proceedings, Vol.1, pp. 433-436, (2000).
- [Vap 82] V.N. Vapnik, *Estimation of Dependences Based on Empirical Data*. Springer Series in Statistics, Springer-Verlag, (1982).
- [Yam 90] T. Yamada and T. Yabuta, *Non linear neural network controller for dynamic system*, 16<sup>th</sup> Annual Conference of IEEE Vol.2, pp.1244-1249, (1990).
- [Yam 92] T. Yamada and T. Yabuta, *Neural Network Controller Using Autotuning Method For Nonlinear Functions*, IEEE Transactions on Neural Networks, Vol.3, No.4, (1992).
- [Yam 84] T. Yamane et S. Shimizu, *Fed-batch techniques in microbial process* Advances in bioch. Eng, Vol.30, pp.147-191, (1984).
- [Yed 98] Y. Yeddou, Etude de Synthèse sur les Réseaux de Neurones et Leurs Applications, Thèse de Magister, ENP, (1998).
- [Wid 64] S.B.W Widrow and F.W. Smith. Pattern, *Recognizing control systems*, In Proceedings of Computer and Information Sciences, (1964).