

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE**

**UNIVERSITE FERHAT ABBAS – SETIF 1-  
UFAS (ALGERIE)**

**MEMOIRE**

Présenté à la faculté de Technologie  
Département d'Electronique

Pour l'obtention du Diplôme de

**MAGISTER**

**Option : Contrôle**

**Présenté Par :**

**LEMITA ABDALLAH**

***Thème***

---

**Commande adaptative par structure  
neuronale récurrente pour la conduite  
d'un procédé de traitement d'eau usée**

---

Soutenu le : / /20 devant la commission d'examen :

*Président : Abdelhafid Khellaf Prof Université de Sétif*

*Rapporteur : Sebti Boulahbel MCA Université de Sétif*

*Examineur : Larbi Khouni MCA Université de Sétif*

*Examineur : Messous Ammar MCA Université de Sétif*

## *Remerciements*

*Je remercie Allah le tout puissant qui m'a donné la force et la volonté pour pouvoir finir ce mémoire de magister.*

*Ce travail a été réalisé au sein du département d'électronique de l'université Farhat Abbas de Sétif.*

*Ce mémoire de magister a été dirigé par Monsieur M. Boulahbel Sebti, maître de conférence à l'institut Universitaire de Technologie de Sétif, dont Je tiens à le remercier pour la confiance qu'il m'a accordée, ses encouragements, et ses précieux conseils.*

*Je tiens à exprimer aussi toute ma gratitude envers les membres du jury pour leur disponibilité et la qualité de leurs remarques dont ils m'ont fait.*

*Je tiens à remercier, tous ceux qui m'ont enseigné durant toutes mes études à l'université de Farhat Abbas.*

*Pour finir J'exprime ma profonde gratitude à mes parents pour leurs encouragements, leur soutien et pour les sacrifices qu'ils ont enduré.*

## Dédicace

*Je dédie ce travail*

*A mes chers parents pour tout les sacrifices infiniment, pour leur soutien durant toute mes années d'études.*

*A mon père Marouch et ma mère Hafssia.*

*A mes frères Farid, Mourad, Samir.*

*A ma sœur Inès.*

*A toute la famille.*

*A mes chers amis : Mohammad. Lotfi, Adel, ammar*

*A mes collègues de la promotion d'électronique de Tébessa : Sami, Atef, Riad, Walid, mohcen.*

*A mes collègues de la promotion d'électronique de Sétif : Abdeslem, Moustafa. Abdelkader. Rabia. Zidane .Fawaz .Hanane.*

*A tous mes amis d'enfance.*

## TABLE DE MATIERE

---

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

### CHAPITRE-I-traitement des eaux

I-1- Introduction .....	3
I-2- Définition.....	4
I-3- Les grandes étapes du traitement.....	4
➤ I-3-1- Le prétraitement.....	5
○ I-3-1-1- Le dégrillage et le tamisage.....	5
○ I-3-1-2- Le dessablage.....	5
○ I-3-1-3- Le dégraissage-déshuilage.....	5
➤ I-3-2- Le traitement primaire .....	6
○ I-3-2-1- La décantation.....	6
○ I-3-2-2- La coagulation et la floculation .....	6
○ I-3-2-3- La flottation.....	8
➤ I-3-3- Le traitement biologique.....	8
○ I-3-3-1- Elimination du carbone .....	8
○ I-3-3-2- Elimination de l'Azote.....	9
I-4- Présentation d'un réacteur biologique.....	10
I-5- Présentation du modèle ASM1.....	11
➤ I-5-1- Unité utilisée.....	12
➤ I-5-2- Les variables.....	12
➤ I-5-3- Description des cinétiques.....	15
➤ I-5-4- Les adaptations : ASM2, ASM2d, ASM3 .....	18
➤ I-5-5- Ecriture du modèle ASM1.....	22

Conclusion

**CHAPITRE -II - Les réseaux de neurones**

II-1- Introduction .....	25
II-2- Historique sur les réseaux de neurones.....	25
II-3- Réseaux de neurones .....	26
➤ II-3-1- Le neurone biologique.....	27
➤ II-3-2- Le neurone formel (Artificiel).....	28
➤ II-3-3- Modélisation d'un neurone formel .....	28
II-4- Architecture des réseaux de neurones.....	32
➤ II-4-1- Les réseaux non bouclés .....	32
➤ II-4-2- Les Réseaux Bouclés (récurrents) .....	33
II -5- Apprentissage des réseaux de neurones.....	34
➤ II-5-1- L'apprentissage supervisé .....	35
➤ II-5-2- L'apprentissage non- supervisé .....	35
II-6- Algorithme d'apprentissage .....	35
➤ II-6-1- adaptation des poids.....	35
II-7- Problèmes.....	39
II-8- Application des réseaux de neurones.....	41
➤ II-8-1- Identification de système .....	41

**Conclusion**

**CHAPITRE -II- La commande adaptative neuronale des systèmes non linéaires**

III-1- Introduction .....	50
III-2- L'identification neuronale.....	51
➤ III-2-1- L'identification du modèle direct.....	52

## TABLE DE MATIERE

---

➤ III-2-2- L'identification du modèle inverse .....	53
III-3- la commande neuronale .....	56
➤ III-3-1- Commande neuronale directe par modèle inverse.....	56
➤ III-3-2- Commande neuronale directe par modèle interne.....	56
III-4- la commande adaptative.....	57
➤ III-4-1- la commande adaptative directe.....	57
➤ III-4-2- la commande adaptative indirecte.....	58
III-5- Méthodes d'obtention des lois d'adaptation.....	59
➤ III-5-1- Méthodes de Lyapunov.....	59
➤ III -5-2- Méthode de descente du gradient.....	61
III -6- la commande adaptative neuronale.....	62
➤ III -6-1- La commande neuronale adaptative directe.....	62
➤ III -6-2- La commande neuronale adaptative indirecte.....	64

## CONCLUSION

## CHAPITRE - IV – Simulation

IV-1- Introduction.....	72
IV-2- Modèle du Bioréacteur.....	72
IV-3- La commande par PI classique .....	76
IV-4- La commande adaptative neuronale du Bioréacteur.....	78
➤ IV-4-1 La commande adaptative neuronale directe.....	78
➤ IV-4-2 La commande adaptative neuronale indirecte.....	81
➤ IV-4-3- La commande hybride .....	82
IV-5- Résultats de la simulation .....	84

## Conclusion

**Conclusion générale**

## Liste des figures

<b>Figure-I-1- : Réacteur Biologique.....</b>	<b>10</b>
<b>Figure-I-2- : Fractionnement de la DCO.....</b>	<b>14</b>
<b>Figure-I-3- : Fractionnement de l'azote.....</b>	<b>15</b>
<b>Figure-I-4- : Evolution des composants au cours du temps.....</b>	<b>17</b>
<b>Figure-I-5-:Représentations du concept mort-régénération et du processus de respiration endogène.....</b>	<b>20</b>
<b>Figure- II -1- : Le neurone biologique.....</b>	<b>27</b>
<b>Figure- II -2- : Le neurone formel.....</b>	<b>28</b>
<b>Figure- II -3- : Fonction Heaviside.....</b>	<b>30</b>
<b>Figure- II -4- : Fonction signe.....</b>	<b>30</b>
<b>Figure- II -5- : Fonction linéaire.....</b>	<b>31</b>
<b>Figure- II -6- : Fonction linéaire à seuil.....</b>	<b>31</b>
<b>Figure- II -7- : Fonction sigmoïde.....</b>	<b>32</b>
<b>Figure- II -8- : Réseau non bouclé.....</b>	<b>33</b>
<b>Figure- II-9- : Réseau récurrent.....</b>	<b>34</b>
<b>Figure- II-10- : schéma de réseau de neurone.....</b>	<b>36</b>
<b>Figure- II-11- : Algorithme de rétro propagation.....</b>	<b>38</b>
<b>Figure- II-12- : Identification par la structure parallèle.....</b>	<b>41</b>
<b>Figure- II-13- : Identification par la structure série-parallèle.....</b>	<b>42</b>
<b>Figure- II-14- : Les sorties <math>y(k)</math> du procédé et <math>\hat{y}(k)</math> du modèle.....</b>	<b>43</b>
<b>Figure – II-15-: Erreur d'identification.....</b>	<b>44</b>
<b>Figure – II-16- : Les sorties <math>y(k)</math> du procédé et <math>\hat{y}(k)</math> du modèle.....</b>	<b>45</b>
<b>Figure – II-17-: Erreur d'identification.....</b>	<b>45</b>
<b>Figure – II -18- : Les sorties <math>y_1(k)</math> du procédé et <math>\hat{y}_1(k)</math> du modèle.....</b>	<b>46</b>

## TABLE DE MATIERE

---

Figure – II -19-: Erreur d’identification.....	47
Figure –II -20- : Les sorties $y_2(k)$ du procédé et $\hat{y}_2(k)$ du modèle.....	47
Figure –II -21-: Erreur d’identification.....	48
Figure –III-1 : Identification du modèle direct.....	52
Figure –III-2 : Architecture générale d’apprentissage.....	54
Figure–III-3 : Architecture indirecte d’apprentissage.....	55
Figure–III-4 : Architecture spécialisée d’apprentissage.....	55
Figure–III-5 : Commande directe par modèle inverse.....	56
Figure–III-6 : Structure de la commande neuronale par modèle interne.....	57
Figure–III-7 : La commande adaptative directe.....	58
Figure–III-8 : La commande adaptative indirecte.....	59
Figure–III-9 : Structure de la commande neuronale adaptative directe.....	63
Figure–III-10 : Structure de la commande neuronale adaptative Indirecte.....	64
Figure–IV–1 : Schéma d’un bioréacteur biologique.....	67
Figure–IV–2: (a), (b) L’Evolution du substrat et de la biomasse du bioréacteur.....	69
Figure–IV–3: (a), (b) L’Evolution du substrat et de la biomasse du bioréacteur.....	70
Figure–IV–4: commande par PI.....	71
Figure–IV–5: Evolutions de la sortie $y(k)$ , de la consigne $y_{ref1}(k)$ et $y_{ref2}(k)$ généré par PI.....	72
Figure–IV–6 : Structure de la commande neuronale directe.....	74
Figure–IV–7 : Schéma du contrôleur neuronal.....	74
Figure–IV–8 : Structure de la commande neuronale indirecte .....	76
Figure–IV–9 : Structure d’une commande hybride .....	78
Figure–IV–10: (a) Evolutions du substrat $S(k)$ , de la consigne $S_{ref1}(k)$ , (b) signal de commande $u(k)$ , (c) l’erreur entre $S(k)$ et $S_{ref1}(k)$ .....	80
Figure–IV–11: (a) Evolutions du substrat $S(k)$ , de la consigne $S_{ref2}(k)$ , (b) signal de commande $u(k)$ , (c) l’erreur entre $S(k)$ et $S_{ref2}(k)$ .....	81



## TABLE DE MATIERE

---

<b>Figure–IV–12: (a) Evolutions du substrat <math>S(k)</math>, de la consigne <math>S_{ref1}(k)</math> , (b) signal de commande <math>u(k)</math>, (c) l’erreur entre <math>S(k)</math> et <math>S_{ref1}(k)</math>, (d) la sortie du modèle neuronal <math>S_m(k)</math> et <math>y_{ref1}(k)</math> .....</b>	<b>84</b>
<b>Figure–IV–13: (a) Evolutions de la sortie <math>y(k)</math>, de la consigne <math>y_{ref2}(k)</math>, (b) signal de commande <math>u(k)</math>, (c) et l’erreur entre <math>y(k)</math> et <math>y_{ref1}(k)</math> , (d) la sortie du modèle neuronal <math>y_m(k)</math> et <math>y_{ref2}(k)</math> .....</b>	<b>86</b>
<b>Figure–IV–14: (a) Evolutions de la sortie <math>y(k)</math>, de la consigne <math>y_{ref1}(k)</math> , (b) signal de commande <math>u(k)</math>, (c) l’erreur entre <math>y(k)</math> et <math>y_{ref1}(k)</math>.....</b>	<b>88</b>
<b>Figure–IV–15: (a) Evolutions de la sortie <math>y(k)</math>, de la consigne <math>y_{ref2}</math>, (b) signal de commande <math>u(k)</math>, (c) l’erreur entre <math>y(k)</math> et <math>y_{ref2}</math>.....</b>	<b>89</b>

## Liste des tableaux

<b>Table-I-1- : diamètres de particules polluantes.....</b>	<b>7</b>
<b>Table-I-2- : le temps de décantation des particules polluants.....</b>	<b>7</b>
<b>Table-I-3- Modèle ASM1.....</b>	<b>21</b>
<b>Tableau- II -1- : Analogie entre le neurone biologique et le neurone formel.....</b>	<b>29</b>
<b>Table-IV-1: Valeurs des paramètres du bioréacteur.....</b>	<b>68</b>
<b>Table-IV-2: Comparaison entre les différentes structures de commandes.....</b>	<b>90</b>

# Introduction générale

### Introduction générale

L'origine de la commande adaptative remonte aux années 1950. Les chercheurs se sont aperçus qu'un contrôleur avec les paramètres fixes n'était pas toujours capable d'assurer les performances désirées, surtout quand les paramètres du système varient avec le temps. La commande adaptative diffère donc d'une commande ordinaire du fait que ses paramètres sont variables et ajustés grâce à un mécanisme qui agit en temps réel en se basant sur l'état du système commandé.

Les réseaux de neurones sont connus comme des approximateurs ajustables et capables de reproduire le comportement complexe des systèmes non linéaires. Les premières applications des réseaux de neurones en commande n'étaient pas basées sur des analyses de stabilité rigoureuses. Plus tard, la théorie de Lyapunov a été introduite pour calculer des lois adaptatives garantissant la stabilité du système en boucle fermée. Les réseaux de neurones sont ensuite utilisés, soit pour approcher directement la loi de commande soit pour approcher les non linéarités formant ainsi une loi de commande neuronale adaptative.

Une application directe de cette technique sera faite sur un processus non linéaire comme le bioréacteur, élément essentiel dans le traitement des eaux usées.

L'ensemble du travail est regroupé en quatre chapitres :

Le premier chapitre sera consacré au système de bioréacteur biologique. Une étude détaillée sera donnée, qui traitera des différents phénomènes qui se passent dans un bioréacteur biologique, avec le modèle mathématique correspondant.

Dans le deuxième chapitre, nous présentons les notions de base de la théorie des réseaux de neurones.

Le travail effectué dans le troisième chapitre tourne autour des principales techniques de la commande à base de réseaux de neurones et la commande adaptative neuronale. Nous allons présenter deux structures de commande : la

## **INTRODUCTION GENERALE**

---

technique de commande neuronale adaptative directe et indirecte avec modèle de référence pour la poursuite de la trajectoire.

Le dernier chapitre traitera de l'utilisation de la commande neuronale adaptative pour une simulation appliquée au bioréacteur. Nous comparerons les résultats obtenus avec un régulateur PI et celles de la commande adaptative neuronale pour situer l'intérêt de cette technique de commande dans les systèmes non linéaires.

Enfin nous finirons par une conclusion générale.

# Chapitre I

## Traitement des eaux usées

## I-1- Introduction

L'épuration des eaux usées au moyen de boues activées est un procédé largement utilisé depuis les deux derniers siècles [Van Haandel et al., 1981], [Henze et al., 1987], [Gujer et al., 1999], [Olsson et Newell, 1999], [Dochain et Vanrolleghem, 2001]. Ce procédé chimique et biologique consiste à mettre en contact par brassage les eaux usées avec un mélange riche en bactéries pour dégrader et éliminer les composants polluants contenus dans l'eau, en suspension ou dissouts. L'activité des bactéries peut être liée à la présence d'oxygène ou d'autres sources d'oxydation, formant des boues activées où la pollution sert de nourriture.

Les traitements utilisés étaient basés sur l'injection d'oxygène dans le milieu polluant, pour enlever des matières organiques. Aujourd'hui, la rapidité d'expansion de la population et le développement industriel imposent de mettre au point de méthodes de dépollution plus efficaces. Durant les quatre dernières décennies, des méthodes pour éliminer des composants bio-chimiques, comme l'azote et le phosphore, ont été mises en œuvre, afin de respecter la réglementation établie pour la protection de l'environnement. Il existe différents processus, de nature biologique ou de nature physico-chimique, pour réaliser cette élimination. Dans le cas des processus par boues activées, si l'élimination de l'azote est souhaitée, différentes configurations sont possibles : bassins séparés ou bassin unique, différents types de réactions (aérée et non-aérée). Pour des considérations de nature économique, une configuration avec un seul bassin -où les deux phases aérobie et anaérobie alternent- a été développée [13].

Dans ce chapitre on citera les grandes étapes du traitement des eaux usées (le prétraitement, traitement primaire et secondaire), et chaque étape elle-même subdivisée en plusieurs parties, ensuite on présentera le modèle ASM1 et on y verra les principaux polluants qui se trouvent dans les eaux usées et les différentes réactions qui s'y trouvent, enfin on va voir les équations différentielles qui peuvent présenter ce modèle.

## I-2- Définition

- **Ammoniaque** : composé gazeux d'azote et d'hydrogène, très soluble dans l'eau.
- **Milieu anoxique** : milieu dans lequel l'oxygène n'est pas présent sous forme libre dissoute.
- **Milieu aérobique** : milieu dans lequel l'oxygène libre est présent en quantité suffisante et joue un certain rôle.
- **Biomasse hétérotrophe** : représente les organismes qui ne peuvent tirer de l'énergie, pour l'entretien de la vie, que de la décomposition des matières organiques, et qui ne peuvent utiliser les composés minéraux comme seule source d'énergie ou pour effectuer la synthèse de matières organiques.
- **Biomasse autotrophe** : représente les organismes qui peuvent utiliser le carbone minéral comme source principale de carbone et qui peuvent tirer de l'énergie pour l'entretien de la vie, de l'oxydation d'éléments minéraux (chimiotrophe) ou de l'énergie lumineuse (phototrophe).
- **Demande chimique en oxygène (D.C.O)** : quantité d'oxygène nécessaire à une dégradation purement chimique des débris et résidus contenus dans une eau polluée sans intervention de micro-organismes.
- **Hydrolyse** : fractionnement de grosses molécules en plus petites molécules.
- **Nitrification** : oxydation de l'azote ammoniacal en nitrates sous l'action de bactéries.
- **Substrat** : matière pouvant être consommée par les micro-organismes.

## I-3- Les grandes étapes du traitement

Les eaux polluées contiennent des matières qui peuvent être minérales ou organique, sous forme soluble ou insolubles. Ces matières subissent une première série de traitements physico-chimiques qui permettront de retirer de l'eau les polluants facilement éliminables. Celle qui est plus difficilement séparable (la pollution soluble) va subir des transformations qui permettront de l'éliminer plus facilement. Ces transformations sont donc de natures physico-chimiques et/ou biologiques [11], [12]. [13].



Selon le degré d'élimination de la pollution et les procédés mis en œuvre, trois niveaux de traitements sont définis.

### **I-3-1- Le prétraitement**

Les prétraitements ont pour objectif d'éliminer les éléments les plus grossiers, qui sont susceptibles de gêner les traitements ultérieurs et d'endommager les équipements. Il s'agit des déchets volumineux (dégrillage), des sables et graviers (dessablage) et des graisses (dégraissage-déshuilage).

#### **I-3-1-1- Le dégrillage et le tamisage**

Au cours du dégrillage, les eaux usées passent au travers d'une grille dont les barreaux, plus ou moins espacés, retiennent les matières les plus volumineuses. Ces éléments sont ensuite éliminés avec les ordures ménagères. Le tamisage, qui utilise des grilles dont l'espacement est plus réduit, peut compléter cette phase de prétraitement. Cependant, il génère beaucoup plus de déchets.

#### **I-3-1-2- Le dessablage**

Le dessablage débarrasse les eaux usées des sables et des graviers par sédimentation. L'écoulement de l'eau à une vitesse réduite dans un bassin appelé "dessableur" entraîne leur dépôt au fond de l'ouvrage. Ces particules sont ensuite aspirées par une pompe. Les sables récupérés sont essorés, puis lavés avant d'être soit envoyés en décharge, soit réutilisés, selon la qualité du lavage.

#### **I-3-1-3- Le dégraissage-déshuilage**

Le dégraissage vise à éliminer la présence de graisses dans les eaux usées, graisses qui peuvent gêner l'efficacité des traitements biologiques qui interviennent ensuite. Le dégraissage s'effectue par flottation. L'injection d'air au fond de l'ouvrage permet la remontée en surface des corps gras. Les graisses sont raclées à la surface, puis stockées avant d'être éliminées (mise en décharge ou incinération). Elles peuvent aussi faire l'objet d'un traitement biologique spécifique au sein de la station d'épuration. De nombreuses stations utilisent des dessaleurs-dégraisseurs combinés.

### **I-3-2- Le traitement primaire**

Le traitement s'effectue par voie physico-chimique avec pour but d'extraire le maximum de matières en suspension et de matières organiques facilement décantables. Trois voies de traitement sont possibles :

#### **I-3-2-1- La décantation**

La décantation primaire classique consiste en une séparation des éléments liquides et des éléments solides sous l'effet de la pesanteur. Les matières solides se déposent au fond d'un ouvrage appelé "décanteur" pour former les "boues primaires". Ces dernières sont récupérées au moyen d'un système de raclage. Ce traitement élimine 50 à 55 % des matières en suspension.

#### **I-3-2-2- La coagulation et la floculation**

Lorsque les particules sont trop petites ou trop légères pour être décantées ou filtrées (dites particules colloïdales), ces particules peuvent rester en suspension dans l'eau durant de très longues périodes, peuvent même traverser un filtre très fin. Par ailleurs, du fait de leur grande stabilité, elles n'ont pas tendance à s'accrocher les unes aux autres.

Pour éliminer ces particules, on a recours aux procédés de coagulation et de floculation. La coagulation a pour but principal de déstabiliser les particules en suspension, c'est-à-dire de faciliter leur agglomération. En pratique, ce procédé est caractérisé par l'injection et la dispersion de produits chimiques. La floculation a pour but de favoriser, à l'aide d'un mélange lent, les contacts entre les particules déstabilisées. Ces particules s'agglutinent pour former un floc qu'on pourra facilement éliminer par décantation.

#### **Quelques ordres de grandeurs**

- Quelle est la taille de ces si petites particules dites colloïdales?

Diamètre des particules (micromètre)	Etat	Exemple
< 0,001	Etat dissous	Argile Huiles et graisses Bactéries
0,001 à 1	Etat colloïdal	
> 1	Solides décantables	Sables

**Table-I-1- : diamètres de particules polluantes**

- Quel est le temps mis par ces particules pour chuter naturellement d'un mètre dans l'eau?

Type	Diamètre des particules (micromètre)	Temps de décantation
Gravier	10 000	1 s
Sables grossiers	1 000	10 s
Sables fins	100	2 min
Argile	10	2 h
Bactéries	1	8 j
Colloïdes	0,1 à 0,00	2 à 200 ans

**Table-I-2- : le temps de décantation des particules polluants**

### I-3-2-3- La flottation

La flottation est un procédé qui permet de séparer un solide ou un liquide de liquide, et qui s'applique à des particules qui ont une densité inférieure à celle de l'eau. La flottation peut être 'naturelle' pour les graisses par exemple, ou 'associée' si de particules qui flottent ont besoin d'une intervention c'est-à-dire par insufflation de l'air qui va accélérer le phénomène. les éléments flottants seront éliminés par raclage de surface.

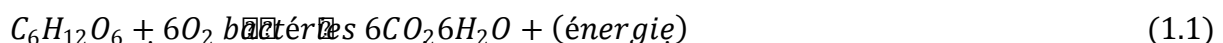
### I-3-3- Le traitement biologique

Les traitements biologiques sont fondés sur l'utilisation d'une biomasse bactérienne qui, pour sa croissance et sa multiplication, va utiliser les composés organiques oxydables et l'azote ammoniacal présents dans l'eau à épurer.

#### I-3-3-1- Elimination du carbone

La boue activée est constituée essentiellement de bactéries et de protozoaires, parfois de champignons, de rotifères et de nématodes. Les bactéries y constituent le groupement le plus important, responsable principalement de l'élimination de la matière organique contenue dans l'effluent d'une part et de la formation des flocons ou flocs d'autre part. La nature des composés organiques constituant la pollution influe naturellement sur le genre dominant, de même les conditions du milieu (pH, température, oxygène dissous).

Pour l'élimination du carbone dans les effluents la voie aérobie est utilisée car l'oxygène est associé aux réactions de dégradation et elles s'instaurent spontanément dans les eaux suffisamment aérées. Le carbone organique se retrouve sous forme de  $CO_2$  et de biomasse. On peut prendre pour exemple le bilan de la dégradation d'une molécule de glucose :

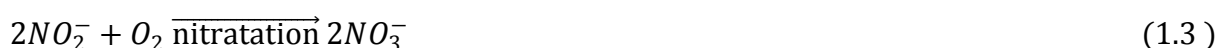
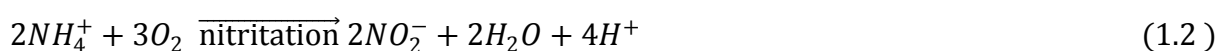


Les micros - organismes s'adaptent au milieu nutritif dans une phase de latence : la vitesse de croissance est alors nulle ou faiblement positive. L'absence de matière nutritive provoque une diminution de la masse des micros – organismes : c'est la phase de respiration endogène qui se produit à l'intérieur même de l'organisme bactérien, sans apport extérieur.

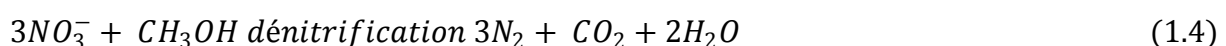
Cette réaction de dégradation du carbone se déroule dans un réacteur biologique à boues activées dont le rendement est intimement lié à la présence d'oxygène ; aussi lui associe-t-on des systèmes d'aération forcée que ce soit par le fond ou en surface.

### I-3-3-2- Elimination de l'Azote

Les eaux résiduaires contiennent différents composés azotés : protéines, urée et produits de décomposition ainsi que de l'azote sous forme minérale. L'élimination de cet azote va se dérouler en deux phases conduites par la biomasse bactérienne : la nitrification et la dénitrification, parallèlement auxquelles une partie de l'azote sera utilisée pour l'anabolisme bactérien. La nitrification consiste en l'oxydation de l'azote organique sous forme d'un ammoniac ( $\text{NH}_4^+$ ) en nitrite ( $\text{NO}_2^-$ ) puis en nitrate ( $\text{NO}_3^-$ ) par l'intermédiaire de micro - organismes autotrophes en milieu aérobie. L'équation de cette réaction peut s'écrire :



Ces réactions sont opérées par des bactéries du type Nitrosomonas, Nitrosospina ou Nitrosococcus pour la nitrification et Nitrobacter pour la nitrification. Elles s'accompagnent de l'accroissement de la biomasse bactérienne qui s'exprimera sur le plan pratique par la formation de boues. ensuit une réduction du nitrate en diazote gazeux opérée le plus souvent par des Pseudomonas hétérotrophes en milieu anaérobie :



#### I-4- Présentation d'un réacteur biologique

La chaîne de traitement est composée d'un bioréacteur, d'un clarificateur / décanteur et d'une boucle de recyclage des boues (Figure-I-1-).

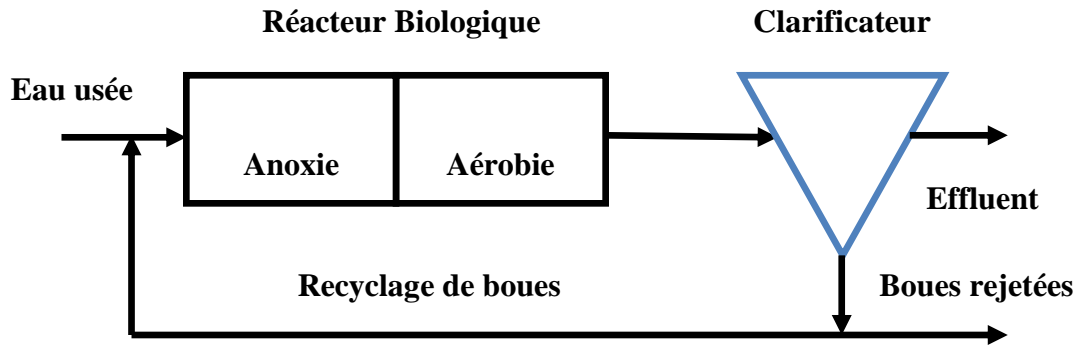


Figure-I-1- : Réacteur Biologique

##### ➤ Le Bioréacteur

L'eau polluée soumise préalablement à un traitement primaire provenant d'une source externe, circule dans le bassin d'aération bioréacteur dans lequel la biomasse bactérienne dégrade la matière organique. Les micro-organismes s'agglomèrent en floccs et produisent les boues.

Le mélange, appelé "liqueur mixte", se compose d'une phase solide (micro-organismes, débris organiques, matières minérales) et d'une phase liquide contenant les matières dissoutes. Pour éliminer la matière azotée, le réacteur biologique peut être séparé en deux zones distinctes : aérobie et anoxique. Dans la majorité des cas, la zone anoxique est placée en amont de la zone aérobie, juste à l'entrée de l'effluent dans le réacteur, pour que les bactéries aient suffisamment de matière organique biodégradable pour effectuer la réaction de dénitrification. Le recyclage de la liqueur mixte de la zone aérobie vers la zone anoxie permet l'élimination des polluants azotés oxydés.

➤ **Le Clarificateur**

En sortie de réacteur, est obtenue une liqueur mixte composée de boues flocculées et d'eau épurée prête à être rejetée dans le milieu naturel. Cette liqueur alimente un clarificateur qui a une fonction de séparation afin de ramener la boue la plus concentrée possible dans le réacteur biologique. Du fait de l'absence d'oxygène, le temps de passage des boues doit être le plus faible possible pour éviter qu'elles ne se trouvent en phase anaérobie. Ceci aurait pour conséquence l'apparition d'odeurs désagréables ainsi qu'une détérioration de la qualité mécanique de la boue perturbant tant le traitement biologique de l'eau

➤ **Le Recyclage de Boues**

La croissance de la biomasse au sein du bioréacteur n'est pas suffisante pour compenser la perte engendrée par le débit de sortie de la liqueur mixte. Pour conserver une concentration en biomasse constante dans le réacteur, une partie des boues du clarificateur est recyclée vers le réacteur. Des purges ont également lieu afin d'extraire la quantité de boues en excès dans le procédé. Les taux de recyclage et de purge permettent également de maîtriser l'âge des boues c'est-à-dire le temps au bout duquel la biomasse est entièrement renouvelée. Ce temps caractérise la forme physiologique de la biomasse dans le réacteur mais aussi la composition de celle-ci. Prenons l'exemple du traitement de la matière azotée : les bactéries autotrophes responsables de la réaction de nitrification ne seront absentes que si l'âge des boues est supérieur à 6 jours, leur temps de duplication étant de 4 jours.

### **I-5- Présentation du modèle ASM1**

Un système à boues activées comporte des phénomènes tels que l'oxydation du carbone, la nitrification et la dénitrification. La simulation de son comportement fait appel à de nombreuses réactions entre un grand nombre de composants. Pour être mathématiquement exploitables, tout en fournissant des prévisions réalistes, les réactions doivent représenter les processus fondamentaux du système.

En outre, le modèle doit prendre en compte la cinétique (vitesse de réaction) et la stœchiométrie (rapport qu'un composant a sur les autres dans une réaction chimique) de chaque processus. Les principales tâches conceptuelles dans le développement du modèle mathématique sont l'identification des réactions, le choix de leurs expressions cinétiques et stœchiométriques appropriées. Associant une description simple des phénomènes biologiques et une représentation conforme de la réalité, elles expriment les vitesses de dégradation des substrats carbonés et azotés en fonction de l'état de pollution (nommé par des variables) et des caractéristiques de la biomasse assurant le traitement (nommé par des paramètres).

### **I-5-1- Unité utilisée**

Une question qui a jeté la confusion et qui dans une certaine mesure a empêché le développement de la théorie des boues activées est le choix du paramètre permettant d'évaluer le mieux la matière organique dans l'eau usée. Trois mesures reconnues sont largement répandues : la Demande Biochimique en Oxygène (**DBO**), le Carbone Organique Total (**COT**), et la Demande Chimique en Oxygène (**DCO**). De ces dernières, la **DCO** est assurément la meilleure mesure. En effet, elle seule fournit un lien entre les équivalents électrons dans le substrat organique, la biomasse et l'oxygène utilisé. En outre, le bilan massique peut être fait en termes de **DCO**. Ainsi, les concentrations de toutes les formes organiques, y compris la biomasse, sont dans des unités de **DCO** dans le modèle.

### **I-5-2- Les variables**

Nous allons présenter un modèle simplifié issu du modèle ASM1 comprenant 11 variables d'état. Les composés insolubles (ou particulaires) portent le symbole **X**, les composés solubles portent le symbole **S**, la biomasse est représentée par **b**, le substrat par **s** et l'oxygène par **O**.

La matière organique dans les eaux usées peut être subdivisée en un certain nombre de catégories.



- **La matière organique non biodégradable** est biologiquement inerte (ou réfractaire) et traverse le système à boues activées sans modification de concentration. Selon l'état physique, deux formes peuvent être identifiées :
  - **la matière organique soluble inerte** ( $S_i$ ) qui quitte le système avec la même concentration qu'à l'entrée.
  - **la matière organique particulaire inerte** ( $X_i$ ) qui se retrouve dans les boues pour être évacuée lors de leur extraction.
- **La matière organique biodégradable** peut également être divisée en deux fractions selon la rapidité d'utilisation par les bactéries :
  - **la fraction rapidement biodégradable** ( $S_S$ ) qui est considérée comme des molécules à structure simple qui peuvent être utilisées directement par les bactéries hétérotrophes et participent à la croissance de la biomasse. Pour la modélisation elle est traitée comme si elle était soluble.
  - **la fraction lentement biodégradable** ( $X_S$ ) qui est constituée de molécules complexes, qui doivent être transformées en substrat rapidement biodégradable avant d'être utilisées. Elle est traitée comme si elle était particulaire.
- **Les biomasses hétérotrophes** ( $X_{BH}$ ) et autotrophes ( $X_{BA}$ ) peuvent être présentes dans l'eau usée, ayant de ce fait un effet important sur l'exécution des réactions dans le système. Cependant, la prédominance et l'intensité de cet effet sont toujours inconnues et ceci n'a pas été considéré par le groupe de travail dans le développement du modèle. Cette biomasse est perdue par décomposition et devient à son tour une fraction de substrat lentement biodégradable ( $X_p$ ).

La matière organique est ainsi fractionnée en différentes parties dont les définitions se distinguent selon leur rôle dans la croissance de la biomasse. Elles interviennent dans le modèle sous forme de **variables** et caractérisent l'état de pollution de l'effluent brut, de la boue des bassins et de l'effluent traité. Ce qui s'illustre pour la DCO totale sous trois formes : la forme substrat, la forme inerte et la

biomasse vivante (hétérotrophe et autotrophe). Ces trois formes se subdivisent à leur tour et nous pouvons distinguer sept fractions dont la hiérarchie est présentée dans l'organigramme ci dessous:

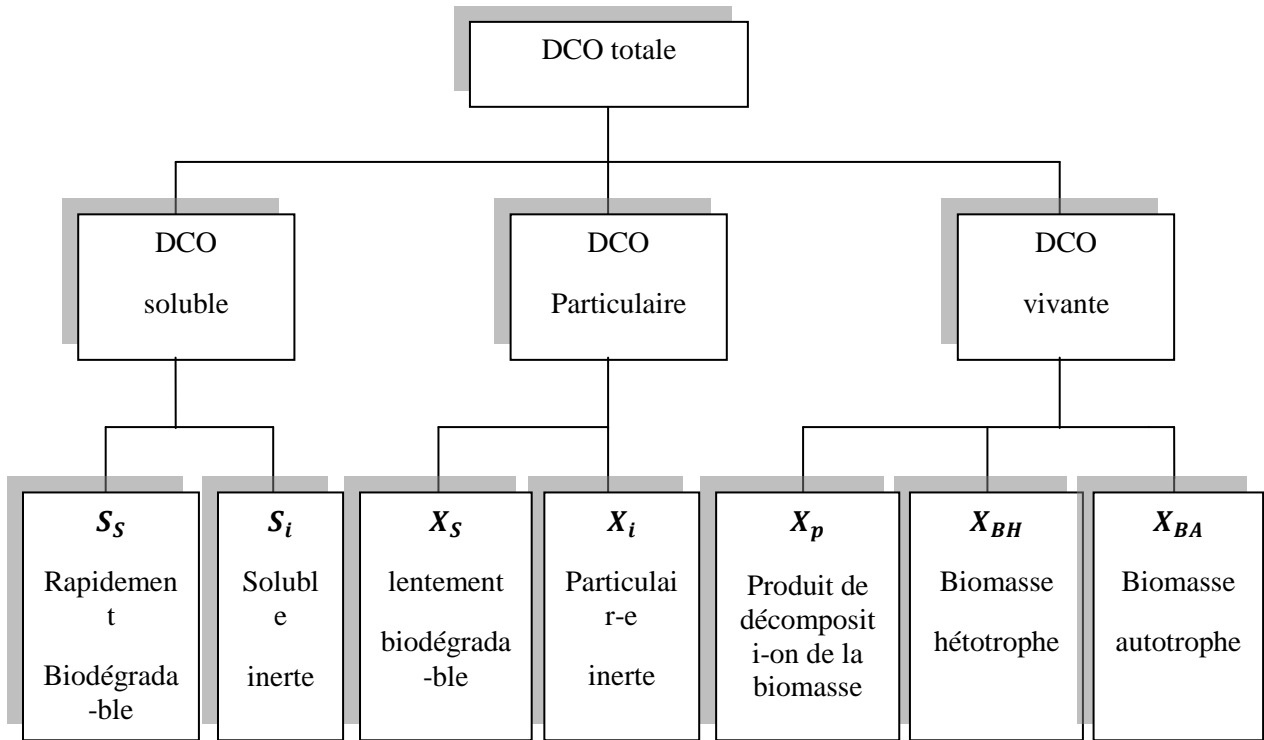


Figure-I-2- : Fractionnement de la DCO

Pour l'azote, nous retrouvons la même démarche de fractionnement. Il consiste à déterminer les fractions minérales ( $S_{NH}$  : azote ammoniacal et  $S_{NO}$  : nitrates), organique biodégradable (soluble :  $S_{ND}$  et particulaire :  $X_{ND}$ ) et réfractaires au traitement (soluble :  $S_{NI}$  et particulaire :  $X_{NI}$ ).

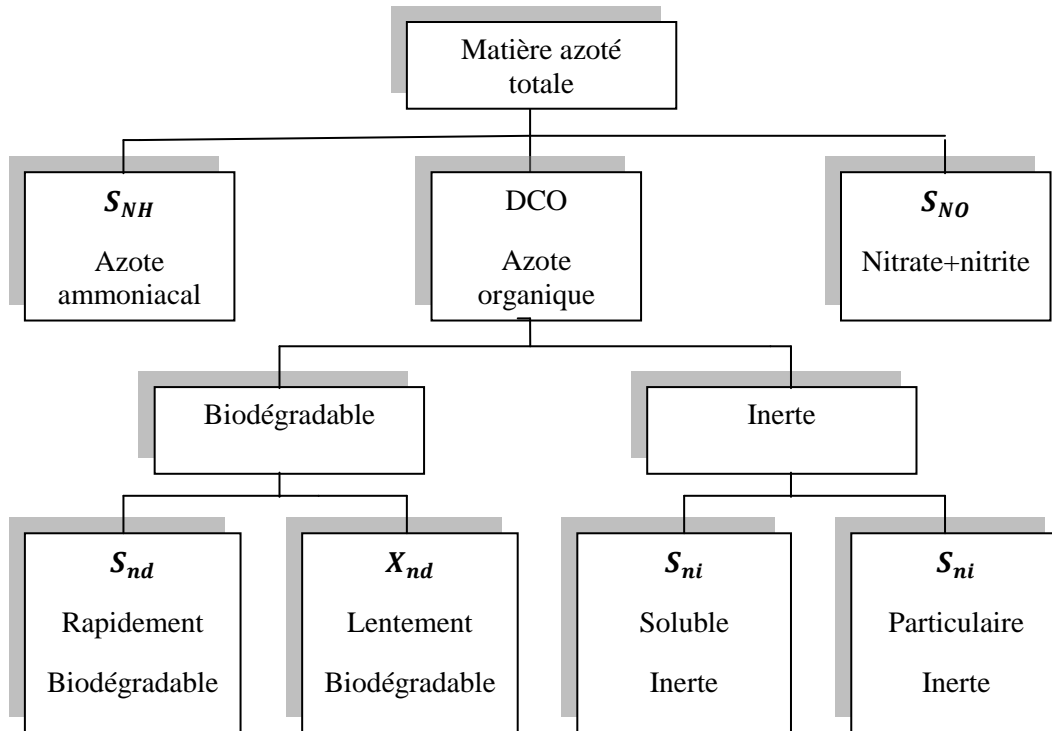


Figure-I-3- : Fractionnement de l'azote

### I-5-3- Description des cinétiques

Les variables d'état du modèle ASM1 sont liées par 8 processus de dégradation [13], [22], [24]. Une brève description de chacun de ces processus est donnée dans les points suivants.

#### 1. la croissance de la biomasse

- **Croissance aérobie des hétérotrophes ( $\rho_1$ )** : une fraction de substrat facilement biodégradable est utilisée pour la croissance de la biomasse et à l'élimination du substrat. il est ainsi naturellement limité par l'épuisement du substrat  $S_S$ , mais également par le manque d'oxygène dissous et d'ammoniaque.
- **Croissance anoxie des hétérotrophes ( $\rho_2$ )** : en l'absence d'oxygène, les organismes hétérotrophes utilisent le nitrate comme récepteur terminal

d'électrons, avec le substrat  $S_S$ , pour produire la biomasse et de l'azote gazeux (dénitrification). Un changement de l'alcalinité est associé à la réaction. L'ammoniaque sert de source d'azote pour la synthèse des micro-organismes.

- **Croissance aérobie des autotrophes ( $\rho_3$ )** : l'ammoniaque est oxydé en nitrate (nitrification) ce qui donne comme résultat une production de biomasse autotrophe et une augmentation de la demande d'oxygène. Il est aussi utilisé comme source d'énergie pour la synthèse. L'effet sur la quantité de biomasse produite est très faible. Par contre, l'alcalinité comme la consommation d'oxygène sont fortement affectées.

## 2. la mortalité de la biomasse

- **Mortalité hétérotrophes et autotrophes ( $\rho_4$ )** : ces processus sont modélisés selon l'hypothèse de mortalité- régénération. Les micro-organismes meurent à un taux déterminé, et cette biomasse devient une combinaison des produits particuliers et de substrat lentement biodégradable. L'azote organique associé à  $X_S$  est disponible comme azote organique particulaire. La DCO n'est pas modifiée. Les taux de mortalité sont supposés invariants en condition d'aérobie et d'anoxie

## 3. l'ammonification de l'azote organique soluble

- **Ammonification de l'azote organique soluble ( $\rho_1$ )** : l'azote organique biodégradable est transformé en ammonium par l'action de la biomasse hétérotrophe active. L'alcalinité est aussi modifiée.

## 4. l'hydrolyse des composants organiques absorbés

- **Hydrolyse des matières organique ( $\rho_7$ )**: le substrat lentement biodégradable correspondant aux substances polluantes est décomposé par un mécanisme

extracellulaire, en produisant du substrat facilement biodégradable utilisable par la biomasse. Ce processus a lieu en aérobie et en anoxie. Le taux d'hydrolyse en anoxie est réduit par un facteur  $\eta_h (< 0)$  par rapport aux conditions d'aérobie. Il est lié à la biomasse hétérotrophe par une fonction du premier ordre et sature quand la concentration de substrat absorbé devient grande par rapport à la biomasse.

- **Hydrolyse de l'azote organique ( $\rho_8$ )** : l'azote organique particulaire biodégradable est décomposé en azote organique soluble à un certain taux défini par la réaction d'hydrolyse précédemment écrite.

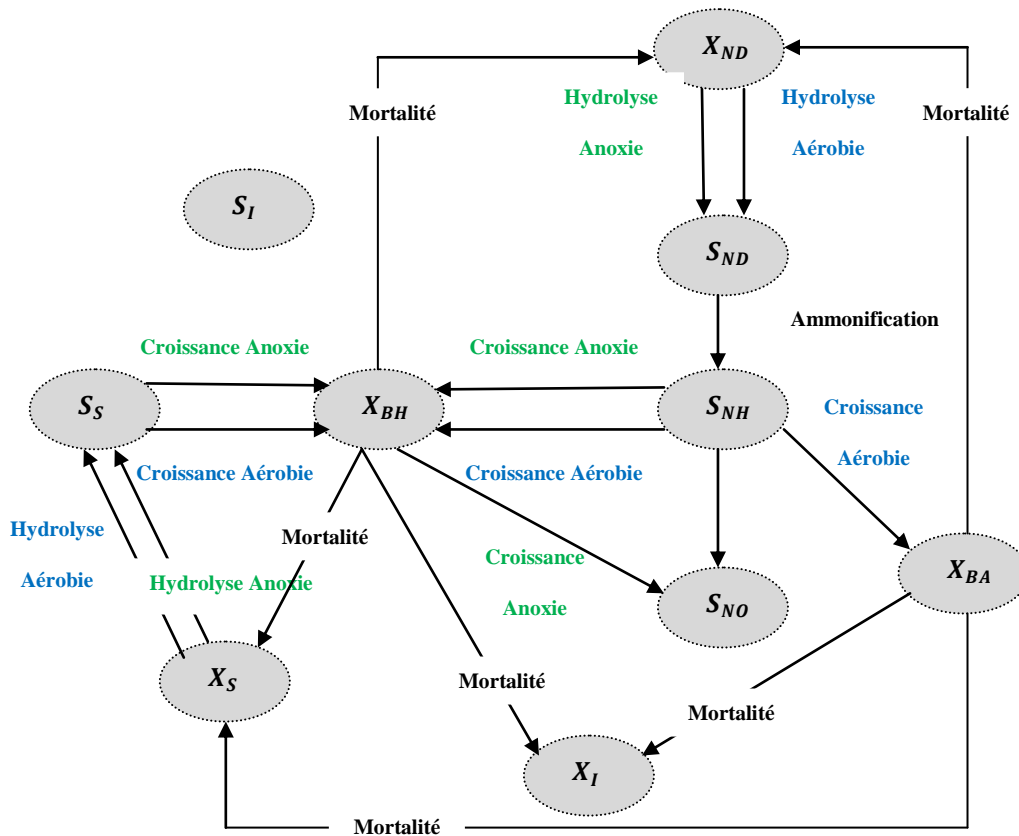


Figure-I-4- : Evolution des composants au cours du temps

### I-5-4- Les adaptations : ASM2, ASM2d, ASM3

- ASM2 et ASM2d :

Les modèles ASM2 (Henze et al., 1995) et ASM2d (Henze et al., 1999) sont des extensions du modèle ASM1. Ils incluent un certain nombre de variables d'état et de processus supplémentaires [23]. De cette manière, ils permettent une simulation de la déphosphatation biologique et physico-chimique.

Cependant, ces deux modèles ne sont pas adaptés à la présente étude (seulement les dégradations du carbone et de l'azote),

- ASM3 :

Le modèle ASM3 (Gujer et al., 1999) est une amélioration du modèle ASM1. Il permet, de simuler le traitement de la pollution carbonée et azotée, sans considération du traitement du phosphore.

A la différence d'ASM1, le modèle ASM3 considère le stockage de la matière organique avant sa consommation ("Ss" transformé en "Xsto", stocké, puis utilisé) au lieu d'une utilisation directe pour la croissance. Il distingue alors les phénomènes de stockage des polymères et des phénomènes de croissance bactérienne.

De plus, le processus de mortalité, est non plus décrit par le concept de mort-régénération, mais par celui de respiration endogène. Une comparaison de ces deux façons de représenter la mortalité est illustrée par la **Figure-I-5-**.

Les processus de conversion des biomasses hétérotrophe et autotrophe sont alors séparés. Le modèle ASM3 considère une consommation moindre d'oxygène, puisque seule la fraction "Xs" de l'affluent est hydrolysée. Il n'y a donc pas de lien avec la matière produite par la biomasse, celle-ci étant inerte dans son intégralité.

➤ **Au niveau des variables :**

- carbonées, la fraction stockée " $X_{sto}$ " a donc été introduite comme étant le nouveau substrat de croissance des hétérotrophes; la fraction " $X_p$ " n'est plus distinguée de la fraction " $X_i$ ";
- azotées, deux introductions ont été effectuées, celle de l'azote sous forme gazeux et celle de l'azote incorporé défini comme une fraction des composés carbonés (biomasse et substrat) et variant selon la DCO.

➤ **Au niveau des processus, quatre ajouts ont été effectués :**

- le stockage aérobie et anoxique de " $X_{sto}$ ", incluant un facteur correctif pour considérer une part des hétérotrophes non actifs pour la dénitrification ;
- la mortalité aérobie et anoxie ou respiration endogène ;
- la respiration aérobie et anoxie des produits stockés ;
- l'hydrolyse, phénomène avec moins d'ampleur dans ce modèle.

Le modèle ASM3 diffère principalement d'ASM1 par la considération du stockage. Cette caractéristique lui confère une structure plus complexe, qui devrait lui attribuer une meilleure représentativité de la réalité.

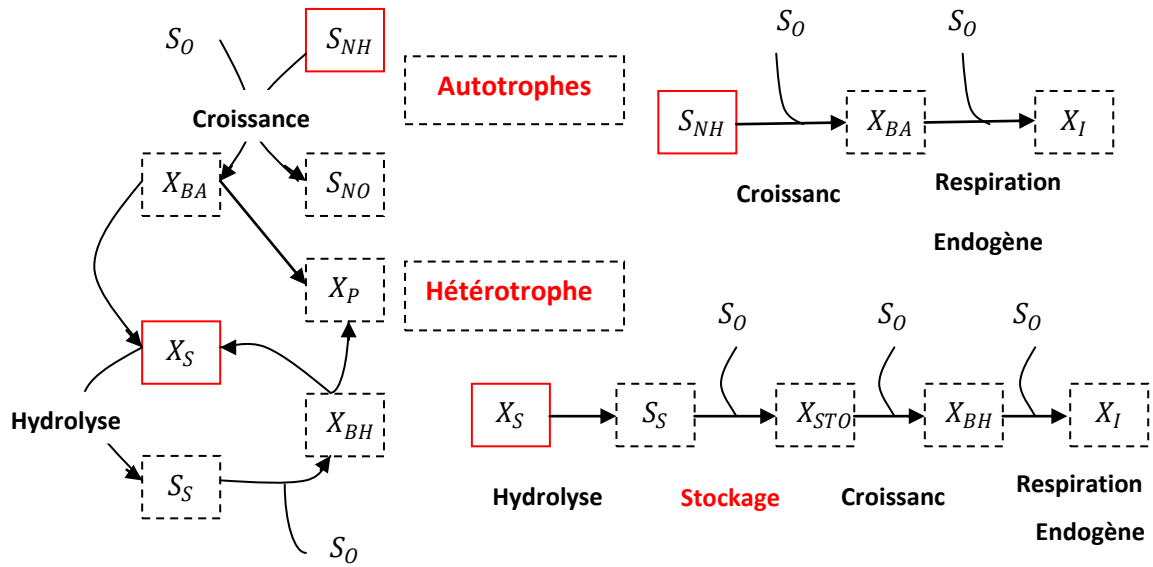


Figure-I-5- : Représentations du concept mort-régénération et du processus de respiration endogène.

### I-5-5- Ecriture du modèle ASM1

Le modèle ASM1 peut se représenter sous deux formes : une forme matricielle dans un tableau, ou sous forme d'équations différentielles

➤ La forme matricielle :

Concentrations j		1	2	3	4	5	6	7	8
i	Réaction $_i$	$S_I$	$S_S$	$X_I$	$X_S$	$X_{B,H}$	$X_{B,A}$	$S_{NO}$	$S_{NH}$
1	Croissance aérobie des hétérotrophes		$\frac{1}{Y_H}$			1			$-i_{NBM}$
2	Croissance anoxie des hétérotrophes		$\frac{1}{Y_H}$			1		$-\frac{1 - Y_H}{2.86Y_H}$	$-i_{NBM}$
3	Croissance aérobie des autotrophes						1	$\frac{1}{Y_A}$	$\frac{1}{Y_H}$ $-i_{NBM}$
4	Mortalité des hétérotrophes			$fr_{XI}$	$1 - fr_{XI}$	-1			
5	Mortalité des autotrophes			$fr_{XI}$	$1 - fr_{XI}$		-1		
6	Ammonification de l'azote organique soluble								1
7	Hydrolyse de la matière organique absorbée		1		-1				
8	Hydrolyse de l'azote organique								



absorbé							
9	10	11	Cinétique de la réaction				
$S_{ND}$	$X_{ND}$	$S_O$					
		$-\frac{1 - Y_H}{Y_H}$	$\rho_1 = \mu_H \frac{S_S}{S_S + K_S} \frac{S_{NH}}{S_{NH} + K_{NH,H}} \frac{S_O}{S_O + K_{O,H}} X_{B,H}$				
			$\rho_2 = \mu_H \frac{S_S}{S_S + K_S} \frac{S_{NH}}{S_{NH} + K_{NH,H}} \cdot \eta_{NO,g} \frac{K_{O,H}}{S_O + K_{O,H}} \frac{S_{NO}}{S_{NO} + K_{NO}} X_{B,H}$				
		$-4.57 \frac{1}{Y_A}$	$\rho_3 = \mu_A \frac{S_{NH}}{S_{NH} + K_{NH,A}} \frac{S_O}{S_O + K_{O,A}} X_{B,A}$				
	$-i_{NBM} - fr_{XI} \cdot i_{NXI}$		$\rho_4 = b_H X_{S,H}$				
	$-i_{NBM} - fr_{XI} \cdot i_{NXI}$		$\rho_5 = b_A X_{B,A}$				
-1			$\rho_6 = k_a X_{B,H} S_{ND}$				
	1		$\rho_7 = k_h \frac{X_S / X_{B,H}}{K_X + X_S / X_{B,H}} \left( \frac{S_O}{S_O + K_{O,H}} + \eta_{NO,h} \frac{K_{O,H}}{S_O + K_{O,H}} \frac{S_{NO}}{S_{NO} + K_{NO}} \right) X_{B,H}$				
1			$\rho_8 = k_h \frac{X_{ND} / X_{B,H}}{K_X + X_S / X_{B,H}} \left( \frac{S_O}{S_O + K_{O,H}} + \eta_{NO,h} \frac{K_{O,H}}{S_O + K_{O,H}} \frac{S_{NO}}{S_{NO} + K_{NO}} \right) X_{B,H}$				

Table-I-3- Modèle ASM1

➤ Les équations différentielles:

Nous donnons ici les équations différentielles à titre indicatif.

Chaque équation du modèle est écrite explicitement, afin de démontrer la complexité lorsqu'on utilise la forme matricielle [10].

Le comportement dynamique de la concentration de la biomasse hétérotrophe est affectée par trois procédés différents; croissance aérobie, anoxique de croissance et par décomposition.

$$\frac{dX_{B,H}}{dt} = \hat{\mu}_H \frac{S_S}{K_S + S_S} \frac{S_O}{K_{O,H} + S_O} X_{B,H} + \eta_g \hat{\mu}_H \frac{S_S}{K_S + S_S} \frac{K_{O,H}}{K_{O,H} + S_O} \frac{S_{NO}}{K_{NO} + S_{NO}} X_{B,H} - b_H X_{B,H} \quad (1.5)$$

La concentration de la biomasse autotrophe est plus simple puisque les autotrophes ne se développent pas dans un environnement anoxique.

$$\frac{dX_{B,A}}{dt} = \hat{\mu}_A \frac{S_{NH}}{K_{NH} + S_{NH}} \frac{S_O}{K_{O,A} + S_O} X_{B,A} - b_A X_{B,A} \quad (1.6)$$

La concentration de substrat facilement biodégradable est réduite par la croissance de bactéries hétérotrophes (dans des conditions aérobies et anoxiques) et est augmenté par hydrolyse du substrat lentement biodégradable.

$$\begin{aligned} \frac{dS_S}{dt} = & -\frac{\hat{\mu}_H}{Y_H} \left( \frac{S_S}{K_S + S_S} \right) \left\{ \left( \frac{S_O}{K_{O,H} + S_O} \right) + \eta_g \left( \frac{K_{O,H}}{K_{O,H} + S_O} \right) \left( \frac{S_{NO}}{K_{NO} + S_{NO}} \right) \right\} X_{B,H} \\ & + k_h \frac{\left( \frac{X_S}{X_{B,H}} \right)}{K_X + \left( \frac{X_S}{X_{B,H}} \right)} \left\{ \left( \frac{S_O}{K_{O,H} + S_O} \right) + \eta_h \left( \frac{K_{O,H}}{K_{O,H} + S_O} \right) \left( \frac{S_{NO}}{K_{NO} + S_{NO}} \right) \right\} X_{B,H} \end{aligned} \quad (1.7)$$

La concentration de substrat lentement biodégradable est augmentée par le recyclage des bactéries mortes selon l'hypothèse de mort-régénération et diminué par le procédé d'hydrolyse.

$$\begin{aligned} \frac{dX_S}{dt} = & (1 - f_p)(b_H X_{B,H} - b_A X_{B,A}) \\ & - k_h \frac{\left( \frac{X_S}{X_{B,H}} \right)}{K_X + \left( \frac{X_S}{X_{B,H}} \right)} \left\{ \left( \frac{S_O}{K_{O,H} + S_O} \right) + \eta_h \left( \frac{K_{O,H}}{K_{O,H} + S_O} \right) \left( \frac{S_{NO}}{K_{NO} + S_{NO}} \right) \right\} \end{aligned} \quad (1.8)$$

L'équation la plus courte est le modèle décrivant la concentration de matériaux inertes produits particuliers provenant de désintégration de la biomasse.

$$\frac{dX_p}{dt} = f_p (b_H X_{B,H} + b_A X_{B,A}) \quad (1.9)$$

La concentration de l'azote organique particulaire est augmentée par la décomposition de la biomasse et diminué par le procédé d'hydrolyse.

$$\begin{aligned} \frac{dX_{ND}}{dt} = & (i_{XB} - f_p i_{XP})(b_H X_{B,H} + b_A X_{B,A}) \\ & - k_h \frac{\left(\frac{X_{ND}}{X_{B,H}}\right)}{K_X + \left(\frac{X_S}{X_{B,H}}\right)} \left\{ \left(\frac{S_O}{K_{O,H} + S_O}\right) + \eta_h \left(\frac{K_{O,H}}{K_{O,H} + S_O}\right) \left(\frac{S_{NO}}{K_{NO} + S_{NO}}\right) \right\} X_{B,H} \end{aligned} \quad (1.10)$$

La concentration de l'azote organique soluble est affectée par ammonification et l'hydrolyse.

$$\begin{aligned} \frac{dS_{ND}}{dt} = & -k_a S_{ND} X_{B,H} \\ & + k_h \frac{\left(\frac{X_{ND}}{X_{B,H}}\right)}{K_X + \left(\frac{X_S}{X_{B,H}}\right)} \left\{ \left(\frac{S_O}{K_{O,H} + S_O}\right) + \eta_h \left(\frac{K_{O,H}}{K_{O,H} + S_O}\right) \left(\frac{S_{NO}}{K_{NO} + S_{NO}}\right) \right\} X_{B,H} \end{aligned} \quad (1.11)$$

La concentration d'ammoniac est affectée par la croissance de tous les micro-organismes comme l'ammoniac et utilisé comme source d'azote pour l'incorporation dans la masse cellulaire. La concentration est également diminuée par le processus de nitrification et augmenté à la suite d'ammonification de l'azote organique soluble. Cela conduit à l'équation différentielle suivante plutôt complexe.

$$\begin{aligned} \frac{dS_{NH}}{dt} = & \left[ -i_{XB} \hat{\mu}_H \left(\frac{S_S}{K_S + S_S}\right) \left\{ \left(\frac{S_O}{K_{O,H} + S_O}\right) + \eta_g \left(\frac{K_{O,H}}{K_{O,H} + S_O}\right) \left(\frac{S_{NO}}{K_{NO} + S_{NO}}\right) \right\} + k_a S_{ND} \right] X_{B,H} \\ & - \hat{\mu}_A \left(i_{XB} + \frac{1}{Y_A}\right) \left(\frac{S_{NH}}{K_{NH} + S_{NH}}\right) \left(\frac{S_O}{K_{O,A} + S_O}\right) X_{B,A} \end{aligned} \quad (1.12)$$

La concentration de nitrate est seulement impliquée dans deux procédés. Elle est augmentée par nitrification et diminué par dénitrification.

$$\begin{aligned} \frac{dS_{NO}}{dt} = & -\hat{\mu}_H \left( \frac{1 - Y_H}{2.86Y_H} \right) \left( \frac{S_S}{K_S + S_S} \right) \left( \frac{K_{O,H}}{K_{O,H} + S_O} \right) \left( \frac{S_{NO}}{K_{NO} + S_{NO}} \right) X_{B,H} \\ & + \frac{\hat{\mu}_A}{Y_A} \left( \frac{S_{NH}}{K_{NH} + S_{NH}} \right) \left( \frac{S_O}{K_{O,A} + S_O} \right) X_{B,A} \end{aligned} \quad (1.13)$$

Enfin, la concentration en oxygène dans les eaux usées est réduite par l'aérobie, la croissance de bactéries hétérotrophes et la biomasse des autotrophes.

$$\begin{aligned} \frac{dS_O}{dt} = & -\hat{\mu}_H \left( \frac{1 - Y_H}{Y_H} \right) \left( \frac{S_S}{K_S + S_S} \right) \left( \frac{S_O}{K_{O,H} + S_O} \right) X_{B,H} \\ & - \hat{\mu}_A \left( \frac{4.57 - Y_A}{Y_A} \right) \left( \frac{S_{NH}}{K_{NH} + S_{NH}} \right) \left( \frac{S_O}{K_{O,A} + S_O} \right) X_{B,A} \end{aligned} \quad (1.14)$$

## Conclusion

Ce chapitre a été consacré aux traitements des eaux usées, le modèle le plus utilisé pour expliquer ces réactions est le modèle ASM 1 dont on a fait deux représentations : la représentation matricielle sous un tableau et la représentation par des équations différentielles.

# Chapitre I

## Les réseaux de neurones

## II-1- Introduction

Les réseaux de neurones connaissent depuis quelques années un succès dans divers domaines de la science (physique, chimie, biologie, finance, etc.), notamment pour la modélisation de processus et la synthèse de lois de commandes, dans notre cas nous nous intéressons à l'application des réseaux de neurones aux domaines de procédés de traitement des eaux usées [8],[2].

Au départ nous citerons un bref historique sur les réseaux de neurones, puis nous rappellerons les définitions et les notations de base relatives aux réseaux de neurones, nous poursuivrons par les différentes architectures, et les méthodologies d'apprentissage.

Enfin nous terminerons ce chapitre par quelques applications des réseaux de neurones.

## II-2- Historique sur les réseaux de neurones

Les recherches menées dans le domaine du connexionnisme ont démarré avec la présentation en 1943 par W. MCCulloch et W. Pitts d'un modèle simplifié de neurone biologique communément appelé neurone formel. Ils montrèrent également théoriquement que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes.

En 1949, D. Hebb initie, dans son ouvrage "The Organization of Behavior", la notion d'apprentissage. Deux neurones entrant en activité simultanément vont être associés (c'est-à-dire que leurs contacts synaptiques vont être renforcés). On parle de loi de Hebb et d'associationnisme.

En 1958, F. Rosenblatt développe le modèle du Perceptron. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception (sert à recueillir les entrées) et une couche de décision. C'est le premier modèle pour lequel un processus d'apprentissage a pu être défini.

S'inspirant du perceptron, Widrow et Hoff, développent, dans la même période, le modèle de l'Adaline (Adaptive Linear Element). Ce dernier sera, par la suite, le modèle de base des réseaux de neurones multi-couches.

En 1969, Les recherches sur les réseaux de neurones ont été pratiquement abandonnées lorsque M. Minsky et S. Papert ont publié leur livre « Perceptrons » (1969) et démontré les limites théoriques du perceptron, en particulier, l'impossibilité de traiter les problèmes non linéaires par ce modèle.

En 1982, Hopfield développe un modèle qui utilise des réseaux totalement connectés basés sur la règle de Hebb pour définir les notions d'attracteurs et de mémoire associative. En 1984 c'est la découverte des cartes de Kohonen avec un algorithme non supervisé basé sur l'auto-organisation et suivi une année plus tard par la machine de Boltzman (1985).

Une révolution survient alors dans le domaine des réseaux de neurones artificiels : une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non-linéaires : le perceptron multicouche ne possède pas les défauts mis en évidence par Minsky. Proposé pour la première fois par Werbos, le Perceptron Multi-Couche apparaît en 1986 introduit par Rumelhart, et, simultanément, sous une appellation voisine, chez Le Cun(1985). Ces systèmes reposent sur la rétropropagation du gradient de l'erreur dans des systèmes à plusieurs couches, chacune de type Adaline de Bernard Widrow, proche du Perceptron de Rumelhart.

Dans nos jours, l'utilisation des réseaux de neurones dans divers domaines ne cesse de croître. Les applications en sont multiples et variées [7], [8].

### **II-3- Réseaux de neurones**

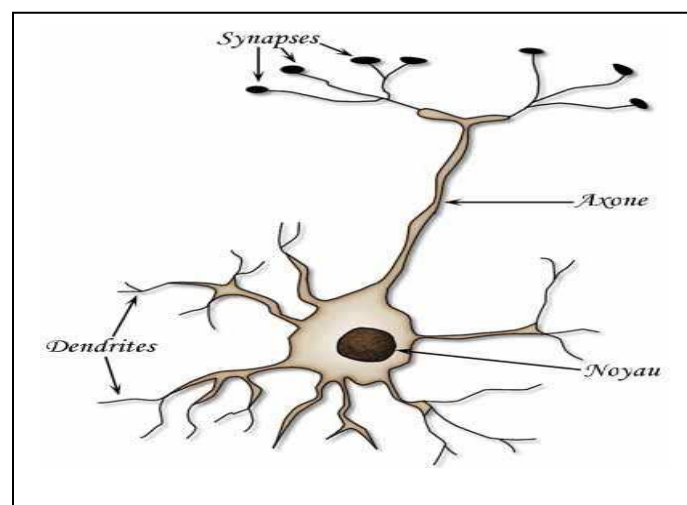
Les réseaux de neurones ont été développés pour résoudre des problèmes de contrôle, de reconnaissance de formes ou de mots, de décision, de mémorisation

comme une alternative à l'intelligence artificielle, et en relation plus ou moins étroite avec la modélisation de processus cognitifs (capable de connaître ou faire connaître) réels et des réseaux de neurones biologiques.

### II-3-1- Le neurone biologique

Le neurone biologique est une cellule vivante spécialisée dans le traitement des signaux électriques. Les neurones sont reliés entre eux par des liaisons appelées axones. Ces axones vont eux mêmes jouer un rôle important dans le comportement logique de l'ensemble. Ces axones conduisent les signaux électriques de la sortie d'un neurone vers l'entrée (synapse) d'un autre neurone. Les neurones font une sommation des signaux reçus en entrée et en fonction du résultat obtenu vont fournir un courant en sortie [9], [3]. (**Figure- II -1**) La structure d'un neurone se compose de trois parties :

- La soma : ou cellule d'activité nerveuse, au centre du neurone.
- L'axone : attaché au soma qui est électriquement actif, ce dernier conduit l'impulsion conduite par le neurone.
- Dendrites : électriquement passives, elles reçoivent les impulsions d'autres neurones.



**Figure- II -1- : Le neurone biologique**



### II-3-2- Le neurone formel (Artificiel)

Le neurone artificiel (ou cellule) est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones appartenant à un niveau situé en amont (on parlera de neurones "amont"). A chacune des entrées est associé un poids  $w$  représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones appartenant à un niveau situé en aval (on parlera de neurones "avals"). A chaque connexion est associé un poids (Figure- II -2).

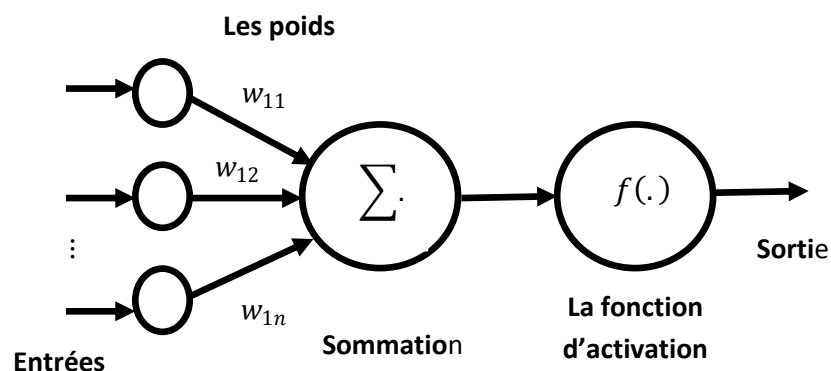


Figure- II -2- : Le neurone formel

### II-3-3- Modélisation d'un neurone formel

Les réseaux de neurones formels sont à l'origine d'une tentative de modélisation mathématique du cerveau humain. Les premiers travaux datent de 1943 et sont l'œuvre de MM. Mac Culloch et Pitts. Ils présentent un modèle assez simple pour les neurones et explorent les possibilités de ce modèle.

La modélisation consiste à mettre en œuvre un système de réseau neuronal sous un aspect non pas biologique mais artificiel, cela suppose que d'après le principe biologique on aura une correspondance pour chaque élément composant le neurone biologique, donc une modélisation pour chacun d'entre eux.

On pourra résumer cette modélisation par le **Tableau-II-1**, qui nous permettra de voir clairement la transition entre le neurone biologique et le neurone formel.

Neurone biologique	Neurone biologique
Synapses	Poids de connexions
Axones	Signal de sortie
Dendrite	Signal d'entrée
Somma	Fonction d'activation

**Tableau- II -1- : Analogie entre le neurone biologique et le neurone formel**

- **Les entrées**

Elles peuvent être :

- Booléennes.
- Binaires (0, 1) ou bipolaires (-1, 1).
- Réelles.

- **Fonction d'activation**

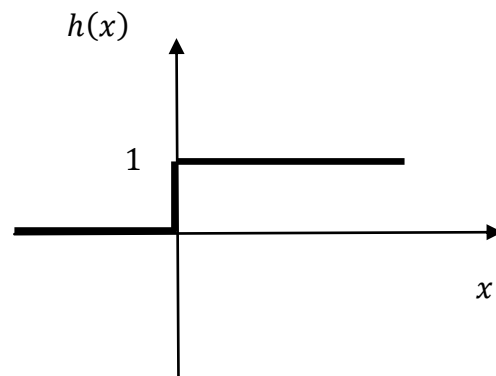
Cette fonction permet de définir l'état interne du neurone en fonction de son entrée totale [14], à titre d'exemple, on cite quelques fonctions souvent utilisées :

- **Fonction binaire à seuil**

Exemple :

Fonction Heaviside(**Figure-II-3**) définie par :

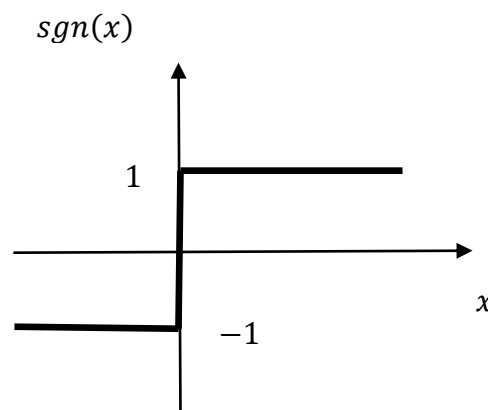
$$h(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases} \quad (2.1)$$



**Figure- II -3- : Fonction Heaviside**

Fonction Signe (**Figure-II-4**) définie par :

$$sgn(x) = \begin{cases} +1 & \text{si } x \geq 0 \\ -1 & \text{sinon} \end{cases} \quad (2.2)$$



**Figure- II -4- : Fonction signe**

Le seuil introduit une non-linéarité dans le comportement du neurone, cependant il limite la gamme des réponses possibles à deux valeurs.

- **Fonction linéaire**

C'est l'une des fonctions d'activations les plus simples (**Figure-II-5**), sa fonction est définie par :

$$f(x) = x \quad (2.3)$$

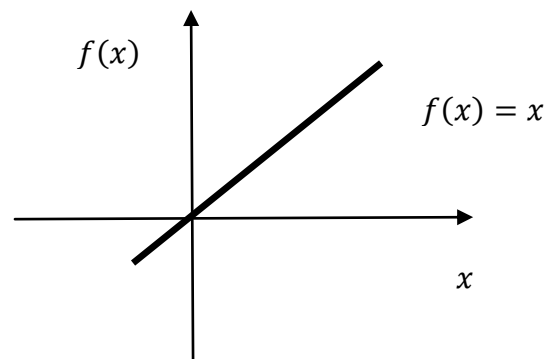


Figure- II -5- : Fonction linéaire

- Fonction linéaire à seuil ou multi-seuils

On peut la définir comme suit :

$$f(x) = \begin{cases} x & x \in [u, v] \\ v & \text{si } x \geq v \\ u & \text{si } x \leq u \end{cases} \quad (2.4)$$

Cette fonction représente un compromis entre la fonction linéaire et la fonction seuil (**Figure-II-6-**) : entre ses deux barres de saturation, elle confère au neurone une gamme de réponses possibles. En modulant la pente de la linéarité, on affecte la plage de réponse du neurone.

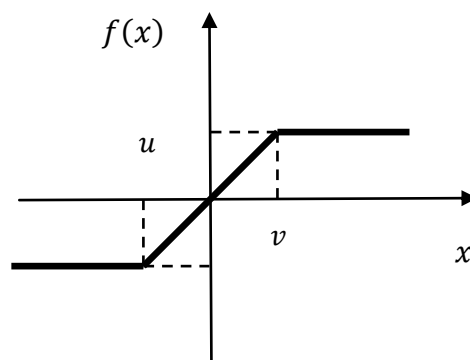
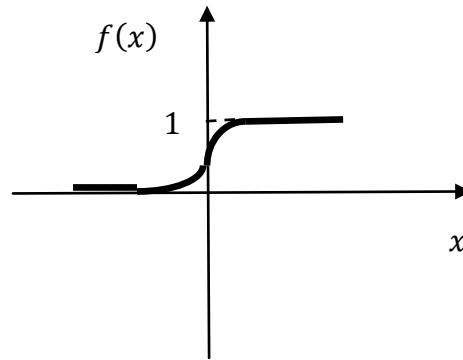


Figure- II -6- : Fonction linéaire à seuil

- Fonction sigmoïde

Elle est dérivable, et sa dérivée est simple à calculer (**Figure-II-7**) elle est définie par :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.5)$$



**Figure- II -7- : Fonction sigmoïde**

➤ **Fonction de sortie**

Elle calcule la sortie d'un neurone en fonction de son état d'activation. En général, cette fonction est considérée comme la fonction identité.

Elle peut être :

- Binaire (0, 1) ou bipolaire (-1, 1)
- Réelle.

## **II-4- Architecture des réseaux de neurones**

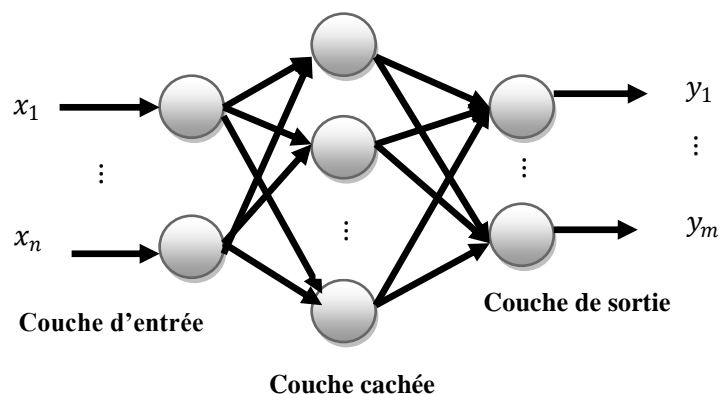
Les réseaux de neurones peuvent être classés en deux types en fonction de la circulation de l'information et les connexions qui relient les neurones entre eux :

- Les réseaux de neurones statiques (ou acycliques, ou non bouclés).
- Les réseaux de neurones dynamiques (ou récurrents, ou bouclés).

### **II-4-1- Les réseaux non bouclés**

Ils peuvent être organisés en plusieurs couches. Une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie. Les signaux d'entrées vont être propagés après entraînement vers la couche de sortie via les couches cachées, sans aucune connexion entre les neurones de la même couche. Ce sont des systèmes statiques, utilisés principalement pour la modélisation statique de processus [3], [8].

La **Figure-II-8** représente une structure d'un réseau de neurones statique avec une couche cachée.



**Figure- II -8- : Réseau non bouclé**

### Remarque

La fonction non linéaire réalisée par un réseau dépend des valeurs des poids. Le temps n'intervient pas comme variable fonctionnelle, le réseau n'a pas de mémoire et ses sorties ne dépendent pas de son passé.

### II-4-2- Les Réseaux Bouclés (récurrents)

Il s'agit de réseaux de neurones avec retour en arrière (récurrent). La **Figure-II-9** illustre un réseau récurrent.

Dans ce type de réseau de neurone, la sortie d'un neurone peut être connectée avec l'entrée du même neurone ou avec celles des autres neurones. L'importance de ces réseaux est qu'ils permettent d'apprendre la dynamique de systèmes, c'est-à-dire

qu'ils peuvent imiter le comportement temporel en insérant des délais dans les boucles, reliant l'entrée à la sortie du réseau ou dans des couches internes.

Ce type de réseau est généralement utilisé pour l'identification et le contrôle des systèmes non linéaires [3]. Malgré tous ces avantages du réseau de neurone récurrent, Ces connexions récurrentes peuvent rendre le réseau instable.

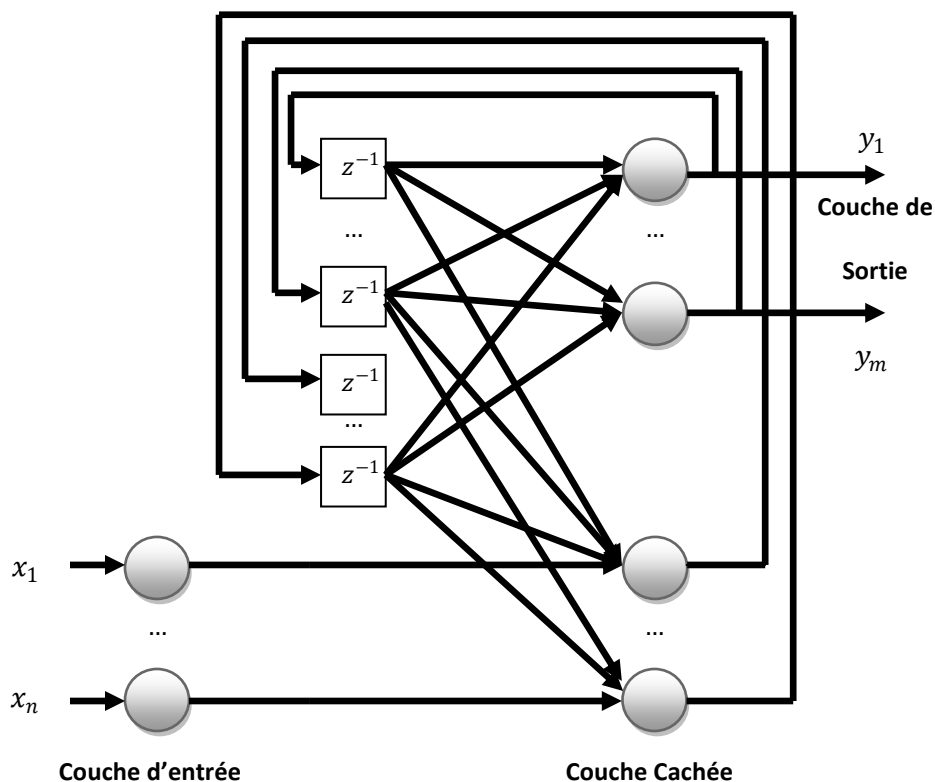


Figure- II-9- : Réseau récurrent

## II -5- Apprentissage des réseaux de neurones

L'apprentissage est la propriété la plus intéressante des réseaux neuronaux. L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré.

Au niveau des algorithmes d'apprentissage, on peut trouver deux grandes classes l'apprentissage supervisé et l'apprentissage non supervisé.

### **II-5-1- L'apprentissage supervisé**

Ce type d'apprentissage nécessite que la réponse désirée du système à entrainer soit connue à priori, et il est effectué de la façon suivante : on donne au réseau les valeurs d'entrées et on calcule sa sortie actuelle correspondante, ensuite les poids sont ajustés de façon à réduire l'écart entre la réponse désirée et celle du réseau (la sortie obtenue par le réseau de neurone). Cette procédure est répétée itérativement jusqu' à ce qu'un critère de performance soit satisfait [8], [6], [2].

### **II-5-2- L'apprentissage non- supervisé**

Dans ce type d'apprentissage, la connaissance à priori de la sortie désirée n'est pas nécessaire, et la procédure d'apprentissage est basée seulement sur les valeurs d'entrées. Le réseau s'auto-organise de façon à optimiser une certaine fonction de cout, sans qu'on lui donne la réponse désirée, cette propriété est appelée auto-organisation [8], [6], [2].

L'avantage de ce type d'apprentissage est sa grande capacité d'adaptation reconnue comme une auto-organisation.

L'apprentissage non-supervisé est utilisé surtout pour le traitement de signale et l'analyse factorielle.

## **II-6- Algorithme d'apprentissage**

L'algorithme d'apprentissage le plus utilisé est basée sur la méthode du gradient. Il cherche à minimiser, par des méthodes d'optimisation non linéaire, une fonction de coût qui constitue une mesure de l'écart entre les réponses réelles du réseau et ses réponses désirées.

### **II-6-1- Adaptation des poids**

---



L'adaptation des poids se fait par la méthode du gradient, qui est basée sur la minimisation d'une fonction de la performance  $J$ ,

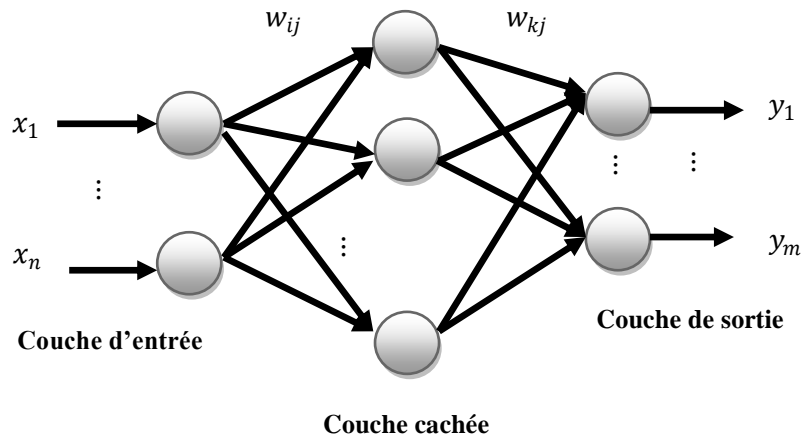


Figure- II-10- : schéma de réseau de neurone

$$J(k) = \frac{1}{2} \sum_{k=1}^N (y(k) - y_d(k))^2 \quad (2.6)$$

$$W_{ij}^l(k+1) = W_{ij}^l(k) - \Delta W_{ij}^l \quad (2.7)$$

$$W_{jk}^L(k+1) = W_{jk}^L(k) - \Delta W_{jk}^L \quad (2.8)$$

$$\Delta W_{ij}^l = \eta \frac{\partial J(W)}{\partial W_{ij}^l(k)} \quad (2.9)$$

$$\Delta W_{jk}^L = \eta \frac{\partial J(W)}{\partial W_{jk}^L(k)} \quad (2.10)$$

$\eta$  : une constante positive appelée taux d'apprentissage. Sa valeur est généralement choisie expérimentalement.

$W_{ij}$  : les poids de la connexion entre le  $j^{\text{ème}}$  neurone de la couche  $l$  et le  $i^{\text{ème}}$  neurone de la couche  $l - 1$ .

$W_{jk}$  : les poids de la connexion entre le  $k^{\text{ème}}$  neurone de la couche  $L$  et le  $j^{\text{ème}}$  neurone de la couche  $L - 1$ .

$i = 1, \dots, n + 1$  : La  $i^{\text{ème}}$  composante du vecteur d'entrée,

$j = 1, \dots, m + 1$  : La  $j^{\text{ème}}$  composante du vecteur de sortie,

$l = 1, \dots, L$  : L'ordre d'une couche dans le réseau de neurone,

Voici les différentes étapes à suivre lors de l'apprentissage d'un réseau de neurones à propagation avec l'algorithme de rétro propagation.

- étape (1) : Initialiser les poids des liens entre les neurones. Souvent une valeur entre 0 et 1, déterminée aléatoirement, est assignée à chacun des poids.
- étape (2) : Application d'un vecteur entrées-sorties à apprendre.
- étape (3) : Calcul des sorties du RNA à partir des entrées qui lui sont appliquées et calcul de l'erreur entre ces sorties et les sorties idéales à apprendre.
- étape (4) : Correction des poids des liens entre les neurones de la couche de sortie et de la première couche cachée selon l'erreur présente en sortie.
- étape (5) : Propagation de l'erreur sur la couche précédente et correction des poids des liens entre les neurones de la couche cachée et ceux en entrées.
- étape (6) : Boucler à la 2e étape avec un nouveau vecteur d'entrées-sorties tant que les performances du réseau de neurone (erreur sur les sorties) ne sont pas satisfaisantes.

### Exemple

Voici l'apprentissage d'un réseau de neurones à propagation avec l'algorithme de rétro propagation avec une couche cachée à trois neurones (**Figure-II-11**).

1 - on calcule  $y_1, y_2, y_3, y_4$  et  $y_5$  :

$$\begin{cases} y_1 = \sigma(x_1 w_{11} + x_2 w_{12} - w_{01}) \\ y_2 = \sigma(x_1 w_{21} + x_2 w_{22} - w_{02}) \\ y_3 = \sigma(x_1 w_{31} + x_2 w_{32} - w_{03}) \end{cases} \quad (2.11)$$

$$\begin{cases} y_4 = \sigma(y_1 \acute{w}_{11} + y_2 \acute{w}_{12} + y_3 \acute{w}_{13} - \acute{w}_{01}) \\ y_5 = \sigma(y_1 \acute{w}_{21} + y_2 \acute{w}_{22} + y_3 \acute{w}_{23} - \acute{w}_{02}) \end{cases} \quad (2.12)$$

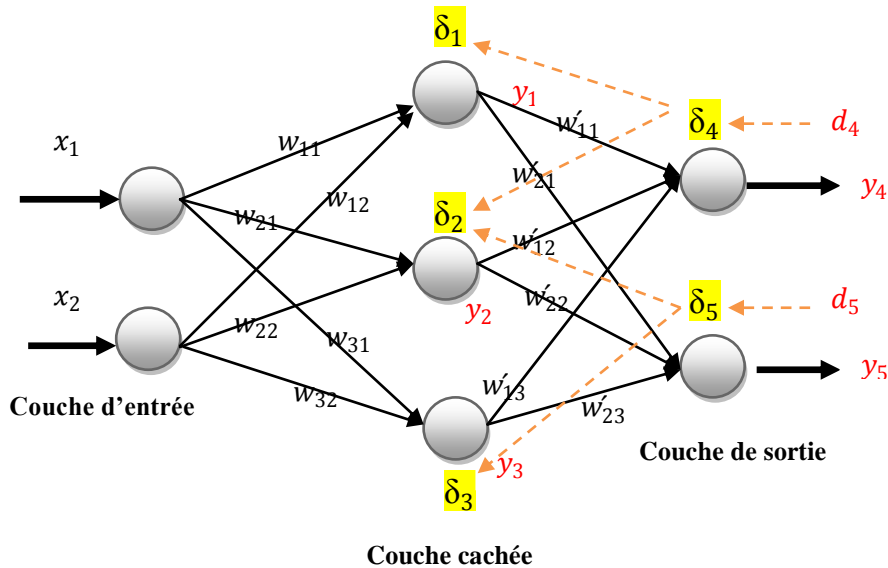


Figure- II-11- : Algorithme de rétro propagation

2 - après avoir calculé les sorties de chaque neurone, on calcule les erreurs de la couche de sortie:

$$\begin{cases} \delta_4 = (d_4 - y_4) \cdot \acute{\sigma}(y_1 \acute{w}_{11} + y_2 \acute{w}_{12} + y_3 \acute{w}_{13} - \acute{w}_{01}) \\ \delta_5 = (d_5 - y_5) \cdot \acute{\sigma}(y_1 \acute{w}_{21} + y_2 \acute{w}_{22} + y_3 \acute{w}_{23} - \acute{w}_{02}) \end{cases} \quad (2.13)$$

et les erreurs de la couche cachée:

$$\begin{cases} \delta_1 = (\acute{w}_{11} \cdot \delta_4 + \acute{w}_{21} \cdot \delta_5) \cdot \acute{\sigma}(x_1 w_{11} + x_2 w_{12} - w_{01}) \\ \delta_2 = (\acute{w}_{12} \cdot \delta_4 + \acute{w}_{22} \cdot \delta_5) \cdot \acute{\sigma}(x_1 w_{21} + x_2 w_{22} - w_{02}) \\ \delta_3 = (\acute{w}_{13} \cdot \delta_4 + \acute{w}_{23} \cdot \delta_5) \cdot \acute{\sigma}(x_1 w_{31} + x_2 w_{32} - w_{03}) \end{cases} \quad (2.14)$$

3 - on calcule la nouvelle valeur de chaque poids entre la couche d'entrée et la couche cachée:

$$\begin{cases} w_{11}(t+1) = w_{11}(t) + \eta(t) \cdot \delta_1(t) \cdot x_1 \\ w_{21}(t+1) = w_{21}(t) + \eta(t) \cdot \delta_1(t) \cdot x_2 \\ w_{12}(t+1) = w_{12}(t) + \eta(t) \cdot \delta_2(t) \cdot x_1 \\ w_{22}(t+1) = w_{22}(t) + \eta(t) \cdot \delta_2(t) \cdot x_2 \\ w_{13}(t+1) = w_{13}(t) + \eta(t) \cdot \delta_3(t) \cdot x_1 \\ w_{23}(t+1) = w_{23}(t) + \eta(t) \cdot \delta_3(t) \cdot x_2 \end{cases} \quad (2.15)$$

et entre la couche cachée et la couche de sortie:

$$\begin{cases} \dot{w}_{14}(t+1) = \dot{w}_{14}(t) + \eta(t) \cdot \delta_4(t) \cdot y_1 \\ \dot{w}_{24}(t+1) = \dot{w}_{24}(t) + \eta(t) \cdot \delta_4(t) \cdot y_2 \\ \dot{w}_{25}(t+1) = \dot{w}_{25}(t) + \eta(t) \cdot \delta_5(t) \cdot y_2 \\ \dot{w}_{35}(t+1) = \dot{w}_{35}(t) + \eta(t) \cdot \delta_5(t) \cdot y_3 \end{cases} \quad (2.16)$$

Il faut recommencer ces opérations pour tous les vecteurs d'apprentissage, puis tester la qualité de l'apprentissage avec les vecteurs de test qui n'ont pas servi à l'apprentissage: ce qui permet de tester les capacités de la généralisation du réseau.

On commence tout d'abord l'entraînement par un choix aléatoire des valeurs initiaux du poids. Une fois que la sortie du réseau soit calculée, l'erreur correspondante et le gradient de l'erreur par rapport à tous les poids sont calculés. Les poids sont alors ajustés. On refait la même procédure pour tous les exemples d'apprentissage. Ce processus est répété jusqu'à ce que les sorties du réseau soient suffisamment proches des sorties désirées.

## II-7- Problèmes

Le problème est qu'on ne connaît pas de méthode systématique permettant de donner à priori le nombre de neurones nécessaires dans la couche cachée pour une application donnée. Cependant il existe des méthodes qui permettent de faire évoluer la structure du réseau en cours d'apprentissage:

- Une méthode par élagage consiste à initialiser le réseau avec un très Grand nombre de neurones dans la couche cachée et à supprimer ceux d'entre eux

dont les poids synaptiques sont très faibles et n'influencent pas trop le comportement du réseau.

- Une méthode par construction consiste à rajouter des neurones dans la couche cachée lorsque l'erreur du réseau ne diminue plus.

➤ **Problème des valeurs initiales des poids du réseau**

Un autre problème est le temps de convergence de l'algorithme de rétro propagation. En effet, plus la somme pondérée des entrées d'un neurone est forte, plus le neurone se trouve dans la zone de saturation de sa fonction d'activation  $\sigma$  (tanh), donc plus la dérivée  $\sigma'$  est faible.

Il faut donc démarrer l'apprentissage en initialisant les poids du réseau à des valeurs suffisamment faibles qui placent la fonction d'activation dans sa zone linéaire: on choisit donc en général des valeurs initiales inférieures à 0.1.

➤ **Problème de sur-apprentissage**

Il faut aussi donner suffisamment d'exemples bien répartis (i.e. représentatifs) pour que le réseau généralise correctement, mais pas trop pour qu'il ne fasse pas de sur apprentissage (i.e. de l'apprentissage par cœur) au détriment des capacités de généralisation.

Un moyen simple de vérifier qu'il n'y a pas sur apprentissage consiste à comparer l'erreur quadratique globale du réseau qui décroît toujours, et l'erreur faite par le réseau sur la base de test qui diminue puis augmente lorsqu'il y a sur apprentissage. La base de test ne doit jamais servir à l'apprentissage.

➤ **Problème du réglage du pas d'Apprentissage**

Le réglage du pas d'apprentissage  $\eta$  joue aussi un rôle important dans la vitesse de convergence.

## **II-8- Application des réseaux de neurones**

### **II-8-1- Identification de système**

L'identification de système ou l'identification paramétrique est une technique de l'automatique consistant à obtenir un modèle mathématique d'un comportement dynamique d'un système à partir de mesures.

Ces derniers ont prouvé leur efficacité dans différents domaines tels que : la reconnaissance de formes, le traitement du signal, le filtrage, la reconnaissance vocale...etc., l'aide à la décision ainsi que la simulation .Ils ont été aussi utilisés dans le domaine de l'automatique, soit pour la commande de système dont nous donnerons quelques détails dans les chapitres suivants, soit pour l'identification de procédés.

L'identification par réseau de neurone peut se faire par deux structures :

#### **➤ Identification parallèle**

La structure parallèle utilise la chaîne de retour directe des sorties du réseau (**Figure-II-12-**).Elle estime les sorties et emploie ces évaluations pour prévoir les futures sorties, cependant, Cette structure ne garantie pas la stabilité en raison de la chaîne de retour.

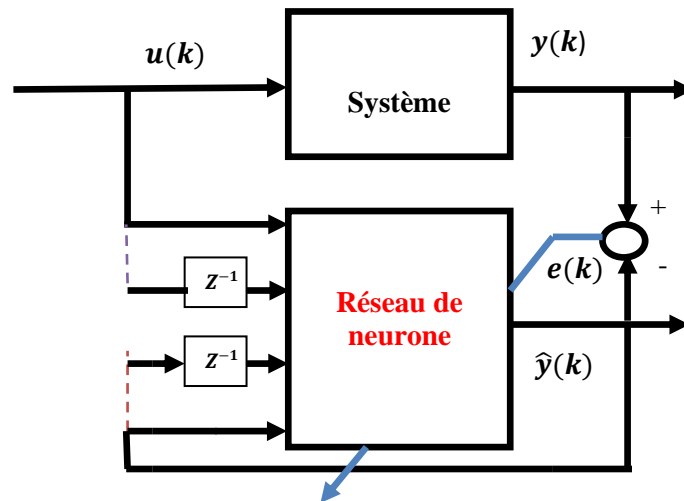


Figure- II-12- : Identification par la structure parallèle

➤ **Identification série-parallèle**

La structure série parallèle n'emploie pas la chaîne de retour. Par contre, elle emploie la sortie réelle pour prévoir les futures sorties. La rétro propagation statique est employée généralement pour assurer la convergence de cette méthode.

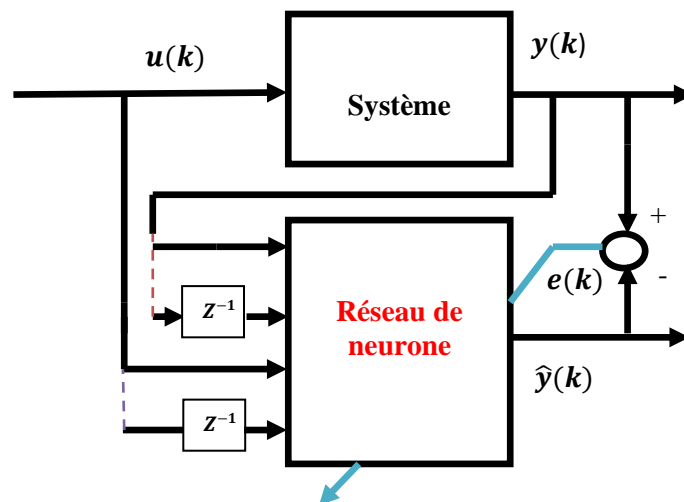


Figure- II-13- : Identification par la structure série-parallèle

**Exemple :**

➤ **Cas des systèmes non linéaires SISO**

1) On a un système d'ordre 2 représenté par l'équation aux différences suivantes:

$$y(k+1) = f[y(k)] + u(k) \quad (1.17)$$

Où la fonction  $f$  est donnée par:

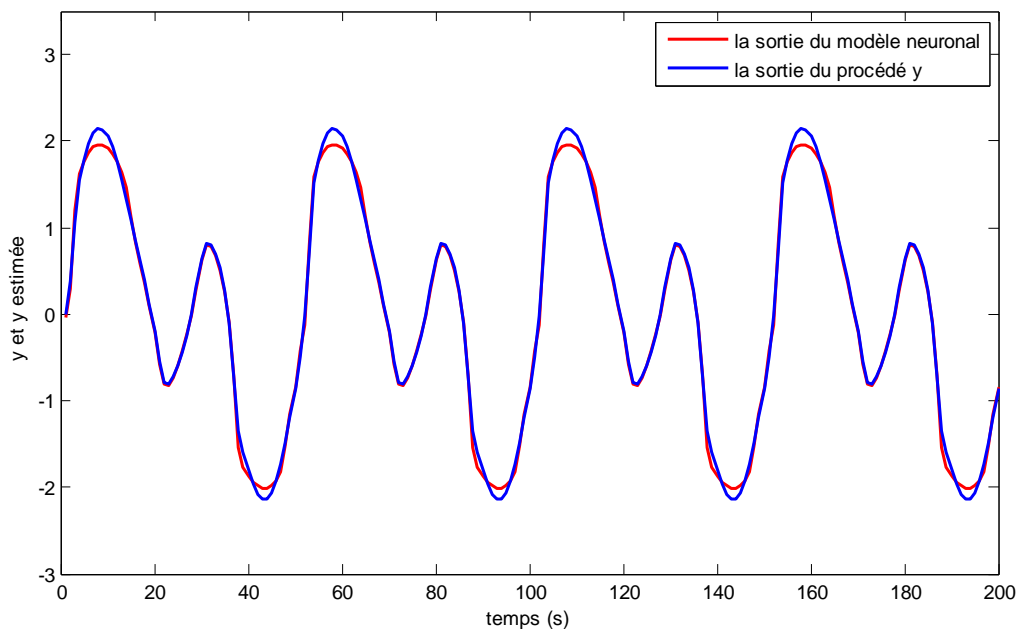
$$f[y(k)] = \frac{y(k)}{1 + y^2(k)} \quad (1.18)$$

Le modèle d'identification neuronal est décrit par :

$$\hat{y}(k+1) = \hat{f}[y(k)] + u(k) \quad (1.19)$$

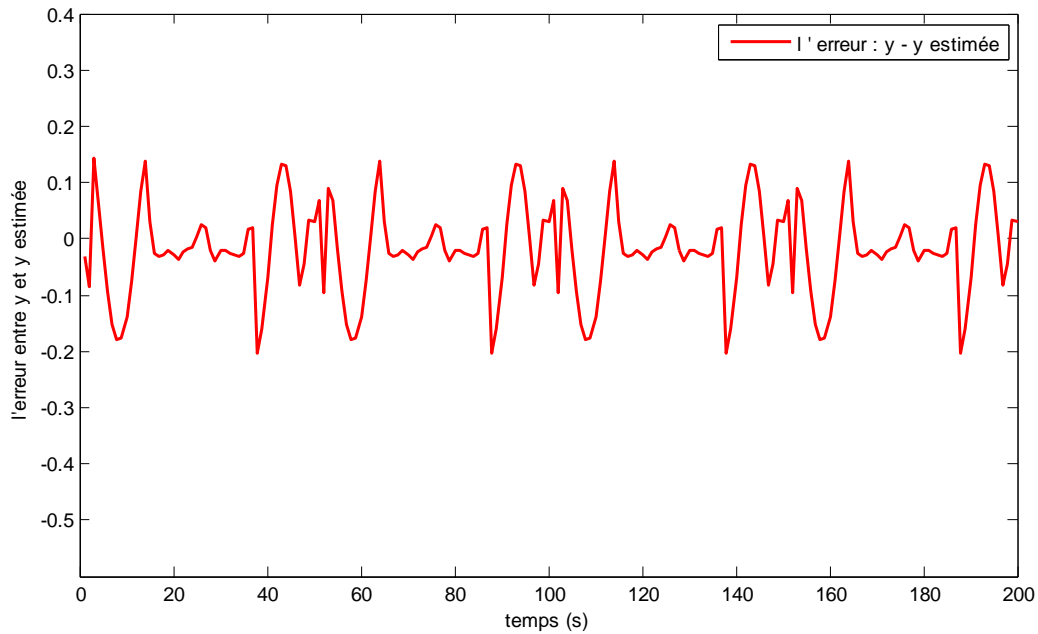
Où l'entrée du système et du modèle est donné par :

$$u(k) = \sin\left(\frac{2\pi k}{25}\right) + \sin\left(\frac{2\pi k}{50}\right) \quad (1.20)$$



**Figure- II-14- : Les sorties  $y(k)$  du procédé et  $\hat{y}(k)$  du modèle**





**Figure – II-15-: L'erreur d'identification**

2) Soit un système d'ordre 2 décrit par l'équation aux différences suivante

$$y(k + 1) = f[y(k), y(k - 1)] + u(k) \quad (1.21)$$

Où la fonction inconnue  $f$  est donnée par:

$$f[y(k), y(k - 1)] = \frac{y(k)y(k - 1)[y(k) + 2.5]}{1 + y^2(k) + y^2(k - 1)} \quad (1.22)$$

Le modèle d'identification neuronal est décrit par :

$$\hat{y}(k + 1) = \hat{f}[y(k), y(k - 1)] + u(k) \quad (1.23)$$

L'entrée du système et du modèle est donné par :

$$u(k) = \sin\left(\frac{2\pi k}{25}\right) \quad (1.24)$$

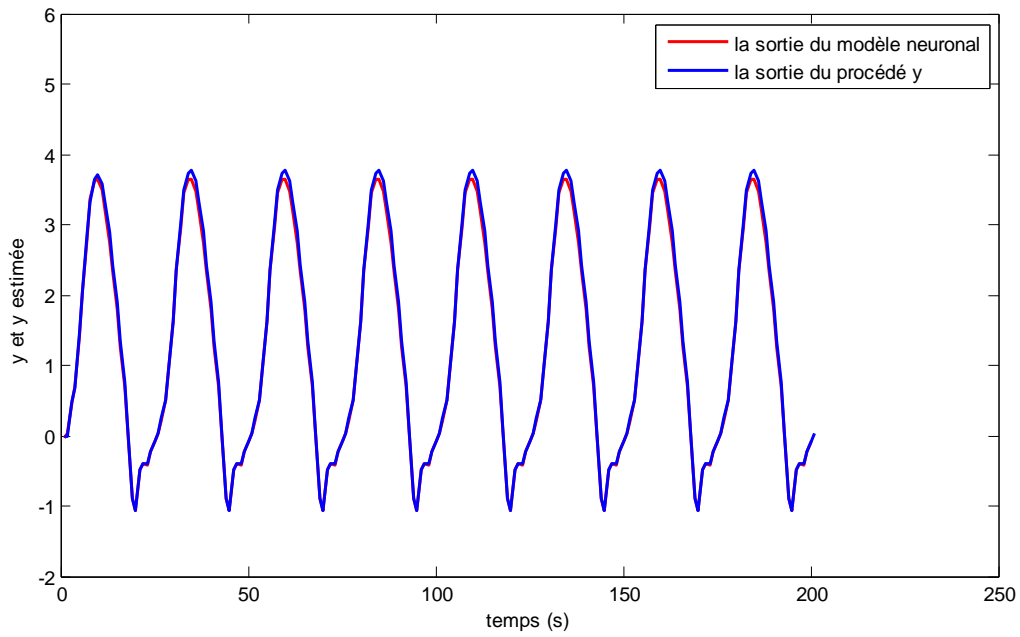


Figure – II-16- : Les sorties  $y(k)$  du procédé et  $\hat{y}(k)$  du modèle

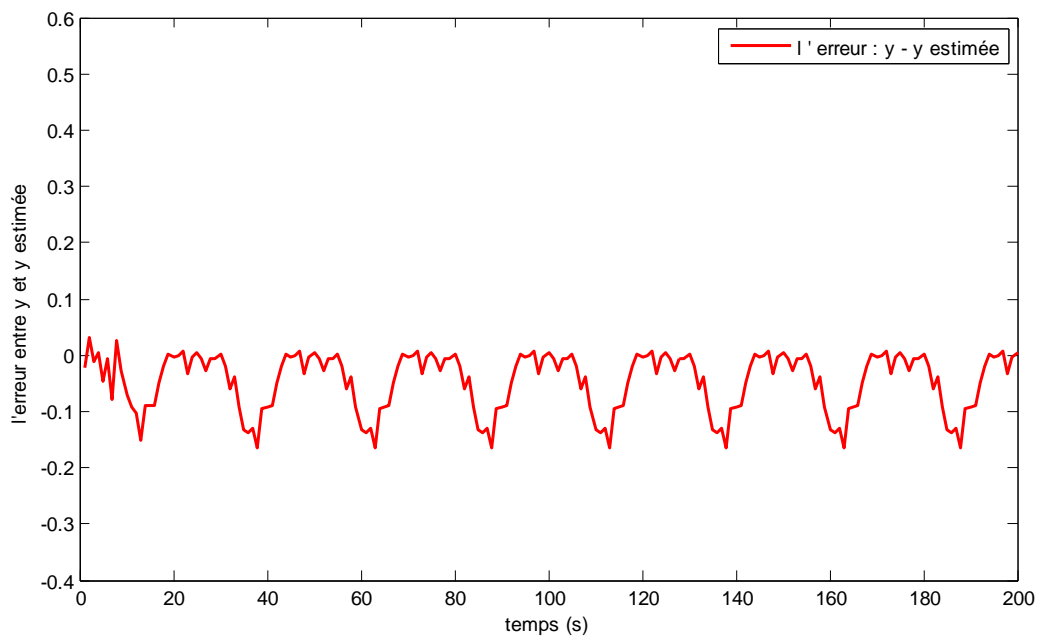


Figure – II-17-: L'erreur d'identification

➤ Cas des systèmes non linéaires MIMO

On a choisi le système MIMO décrit par l'équation aux différences :

$$\begin{pmatrix} y_1(k+1) \\ y_2(k+1) \end{pmatrix} = \begin{pmatrix} \frac{y_1(k)}{1+y_2^2(k)} \\ \frac{y_1(k)y_2(k)}{1+y_2^2(k)} \end{pmatrix} + \begin{pmatrix} u_1(k) \\ u_2(k) \end{pmatrix} \quad (1.25)$$

Le modèle d'identification neuronal est ainsi donné décrit par :

$$\begin{pmatrix} y_1(k+1) \\ y_2(k+1) \end{pmatrix} = \begin{pmatrix} \hat{f}^1[y_1(k), y_2(k)] \\ \hat{f}^2[y_1(k), y_2(k)] \end{pmatrix} + \begin{pmatrix} u_1(k) \\ u_2(k) \end{pmatrix} \quad (1.26)$$

L'entrée du système est donné par :

$$[u_1(k), u_2(k)] = \left[ \sin\left(\frac{2\pi k}{25}\right), \cos\left(\frac{2\pi k}{25}\right) \right] \quad (1.27)$$

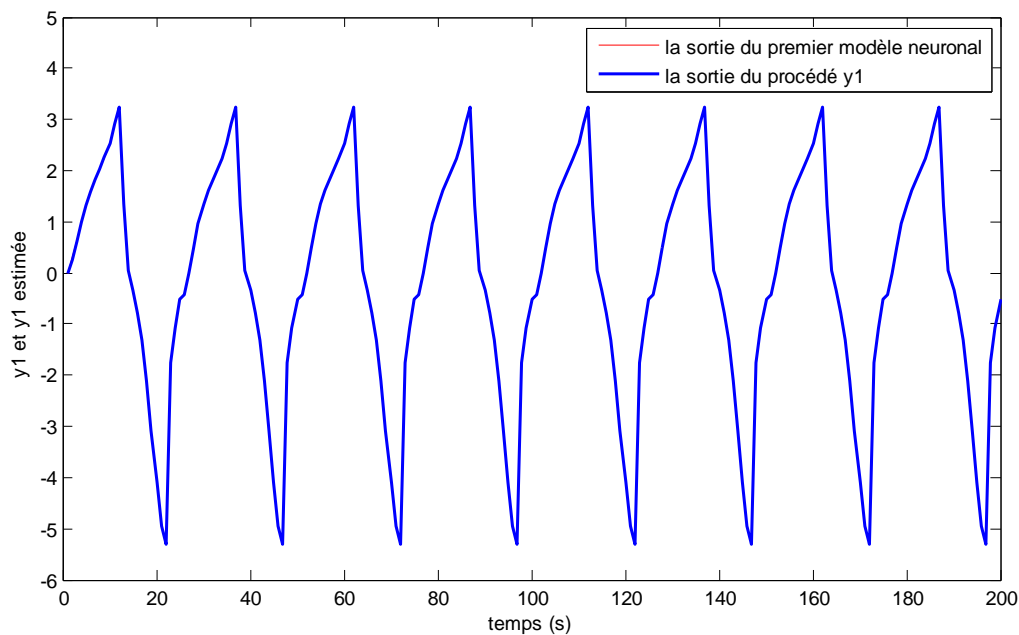


Figure – II -18- : Les sorties  $y_1(k)$  du procédé et  $\hat{y}_1(k)$  du modèle

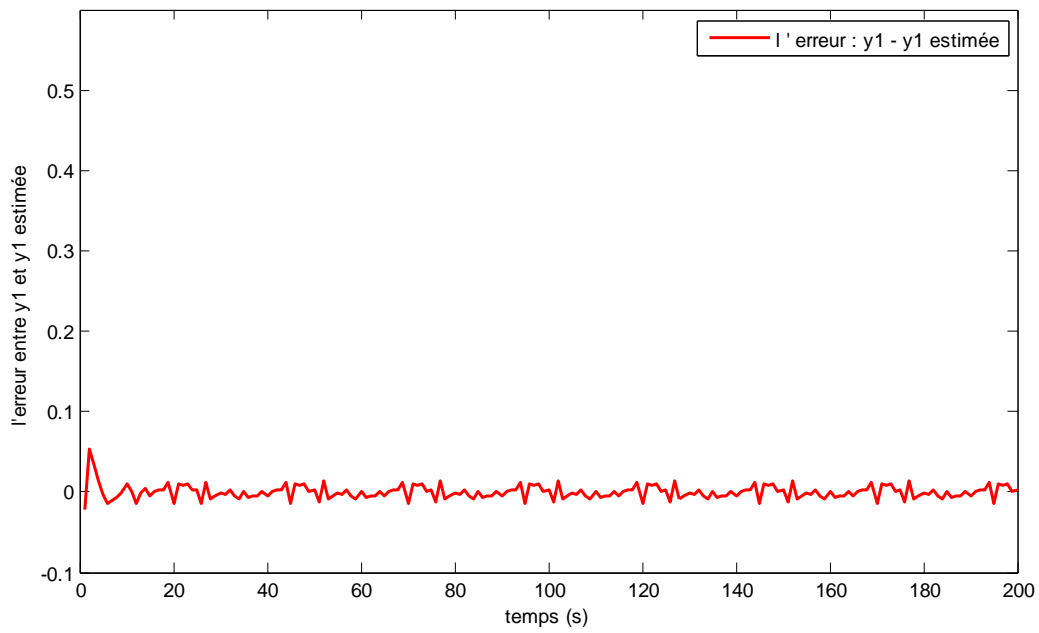


Figure – II -19-: L'erreur d'identification

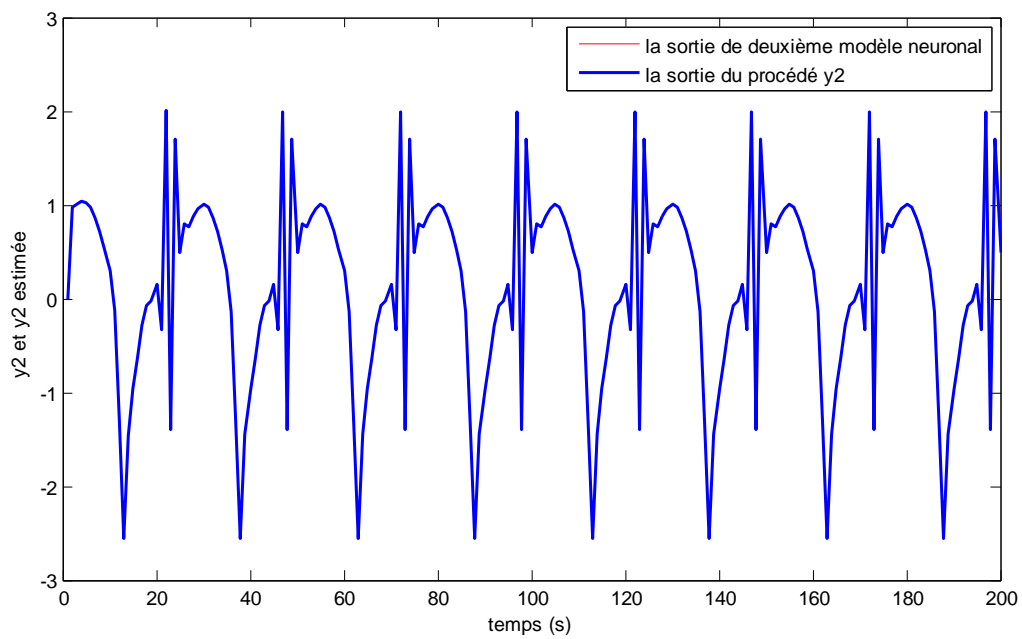
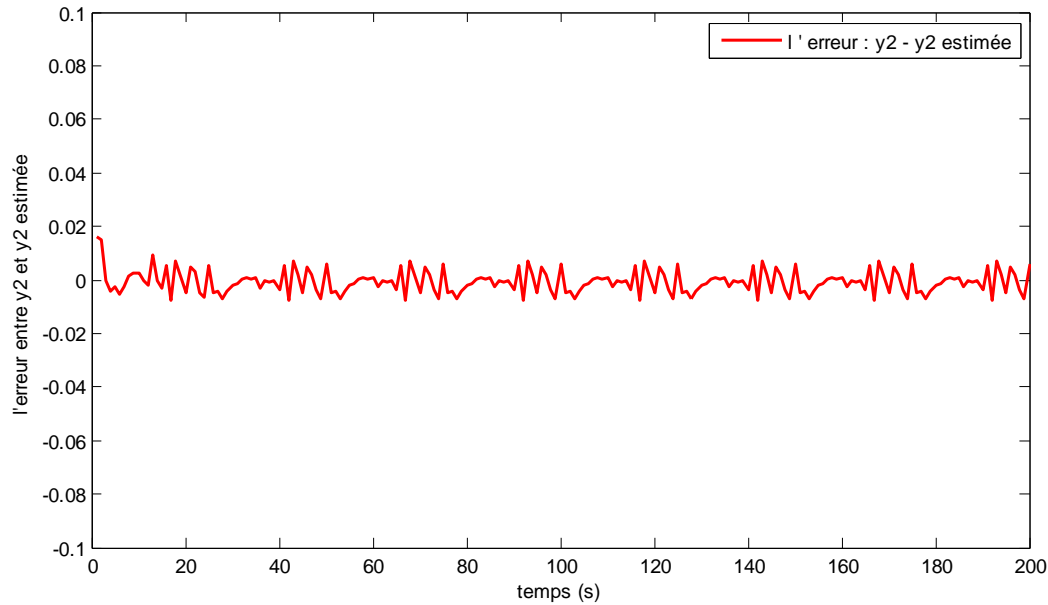


Figure – II -20- : Les sorties  $y_2(k)$  du procédé et  $\hat{y}_2(k)$  du modèle



**Figure –II -21-: L'erreur d'identification**

## Conclusion

Les réseaux de neurones possèdent des propriétés remarquables qui sont à l'origine de leurs utilisations dans beaucoup de domaines.

Dans ce chapitre nous avons donné les définitions essentielles relatives aux réseaux de neurones, nous avons fait une distinction entre les réseaux de neurones non bouclés (statiques), qui réalisent les équations algébriques, et les réseaux de neurones bouclés (dynamiques), qui réalisent des équations aux différences non linéaires.

Nous avons vu que le nombre de neurones cachés est très important, parce qu'il détermine la capacité de calcul du réseau. Un nombre insuffisant de neurones cachés peut compromettre la capacité du réseau à résoudre le problème.

Inversement, trop de neurones peuvent rendre le réseau incapable de résoudre le problème. Malheureusement il n'y a pas de méthode systématique pour déterminer le nombre de neurones, le nombre de couches ou le choix des paramètres internes

de l'algorithme comme l'erreur quadratique qu'on veut atteindre et le nombre d'itérations. Donc la détermination des paramètres du réseau de neurone reste une affaire délicate.

Enfin, nous avons décrit quelques applications des réseaux de neurones.

# Chapitre III

## La commande adaptative neuronale des systèmes non linéaire

### III-1- Introduction

Dans la théorie de la commande des systèmes dynamiques non linéaires, les techniques conventionnelles ont montré souvent leur faiblesse et insuffisance surtout quand les systèmes à étudier présentent de forts non linéarités et des incertitudes, donc le recours aux méthodes de commandes par apprentissage qui peuvent surmonter des difficultés engendrées par les forts non linéarités est devenu une nécessité, Des recherches avancées ont permis de développer de nouvelles techniques de commande non linéaire telles que la commande non linéaire adaptative.

Dans ces techniques de commande adaptative non linéaire, l'incertitude du système non linéaire est identifiée en ligne. Un inconvénient majeur de ces techniques de commande non linéaire adaptative est que la dynamique inconnue possède une structure connue avec des paramètres inconnus entrant linéairement dans la dynamique.

La paramétrisation linéaire de la dynamique inconnue du système pose des problèmes sérieux pour l'utilisation des algorithmes de commande adaptative dans des applications pratiques, parce qu'il est difficile de fixer la structure des non linéarités inconnues. Donc on a besoin d'un approximateur non linéaire pour estimer les non linéarités inconnues, c'est le cas de réseaux de neurones artificiels qui ont trouvé une large utilisation dans le domaine de la commande intelligente. Ceci est dû à leur propriété d'approximer n'importe quel système non linéaire.

On verra dans ce chapitre au premier temps, le principe de l'identification neuronale des systèmes non linéaires et La commande par réseau de neurone. Un modèle direct et inverse doit être accompli hors ligne avant d'être utilisé pour la commande neuronale.



On poursuivra par la commande adaptative en partant de la commande adaptative neuronale directe, puis la commande adaptative indirecte.

### **III-2- L'identification neuronale**

L'identification d'un système consiste principalement à trouver un modèle mathématique pour représenter le comportement dynamique d'un système à partir des mesures de ses entrées/sorties. Si l'identification est à la base de réseaux de neurones, on parle alors de l'identification neuronale qui permet d'approximer un système quelconque par un modèle neuronal.

L'identification se fait en général en deux étapes: l'étape qualitative (caractérisation) et l'étape quantitative (estimation des paramètres) [1][8][12].

- l'étape qualitative permet de déterminer la structure du modèle. Il s'agit de la connaissance de l'ordre du système, ses conditions initiales, l'amplitude maximale de ses entrées et sorties, et la forme des équations qui décrivent le système.

Deux sortes de modèles peuvent être dénombrées :

- Les modèles paramétriques comme les équations différentielles ou les fonctions de transfert.
- Les modèles non-paramétriques comme les réponses impulsionnelles ou fréquentielles.
- Etape quantitative, quant à elle consiste à déterminer les valeurs numériques des paramètres (poids des connexions) qui interviennent dans le modèle d'identification.

L'identification passe par quatre étapes :

- choix de la structure du modèle : ordre et retard du système à identifier.
- choix de l'entrée à appliquer au système.

- algorithme d'apprentissage.
- Validation le modèle obtenu: réalisation de plusieurs tests de vérification en utilisant des valeurs différentes de celles utilisées lors de l'identification.

### III-2-1- L'identification du modèle direct

L'identification neuronale du modèle direct utilise un réseau d'anticipation qui estime la sortie du système à partir des anciennes valeurs de son entrée et de sa sortie. Le modèle neuronal direct du système à commander sera utilisé pour la commande par modèle interne. Le schéma d'apprentissage du modèle neuronal direct est donné par la Figure –III-1.

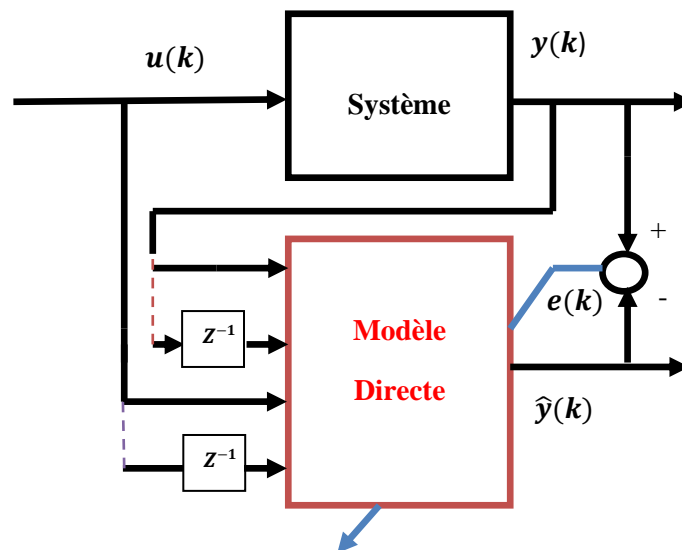


Figure –III-1 : Identification du modèle direct

Avec :

$u(k)$  : l'entrée de commande,

$y(k)$  : la sortie désirée,

$\hat{y}(k)$ : La sortie du modèle neuronal,

$e(k)$  : l'erreur d'apprentissage.

### III-2-2- L'identification du modèle inverse

Le modèle inverse de système joue un rôle très important dans la théorie de la commande, mais malheureusement l'accomplissement de sa forme analytique n'est pas toujours disponible. Un système dynamique peut être décrit par l'équation (3.1) reliant ces entrées aux sorties :

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

En général, le modèle inverse de ce système peut être écrit sous la forme suivante :

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

L'identification de ce modèle inverse commence par la détermination du vecteur d'entrée, à savoir le nombre des retards en sorties et en entrées, ceci est lié à l'ordre du système [2]. [19]. Puis on choisit l'architecture du réseau c'est-à-dire le nombre de couches cachées et le nombre de neurones dans chaque couche. Pour ce faire, trois architectures peuvent être données :

#### ➤ Architecture d'apprentissage général

Dans cette architecture, l'apprentissage hors ligne du modèle inverse du système à commander sera utilisé en minimisant l'erreur  $e(k)$  entre l'entrée du système  $u(k)$  et la sortie du réseau de neurone  $\hat{u}(k)$  [43].

La **Figure -III-2** montre le principe de cette méthode.

$$e(k) = \frac{1}{2} \sum_{k=1}^N (u(k) - \hat{u}(k))^2 \quad (3.3)$$

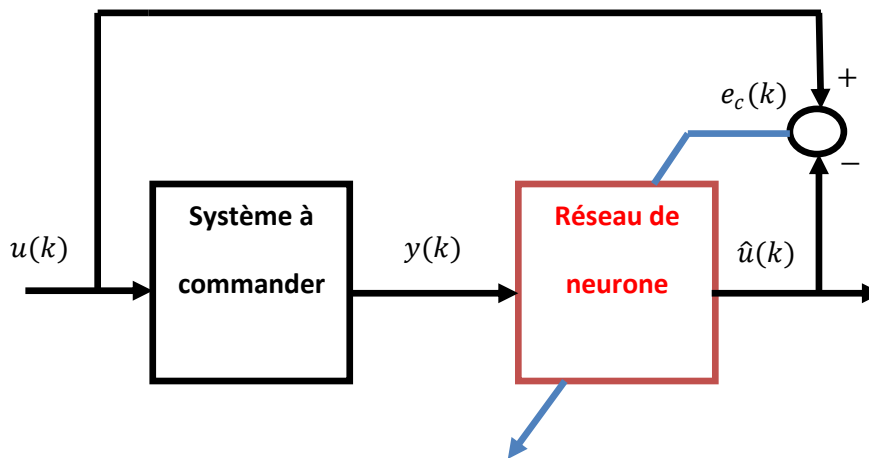


Figure –III-2 : Architecture générale d'apprentissage

L'inconvénient de cette méthode est que la sortie  $y(k)$  du système utilisé dans l'apprentissage ne garantit pas que les sorties du modèle neuronal  $\hat{u}(k)$  vont être dans des régions voulues pour le succès de son utilisation dans la commande.

### ➤ Architecture d'apprentissage indirecte

L'idée de cette architecture est que le modèle inverse en cours d'apprentissage sert aussi à commander le système. La consigne  $y_{ref}(k)$  fournie au premier réseau qui produit une commande  $u(k)$  au système, la sortie de celui-ci devient une consigne pour le deuxième modèle inverse, celui-ci produit alors une commande  $\hat{u}(k)$  (Figure-III-3).

La différence entre  $u(k)$  et  $\hat{u}(k)$  sert de signal d'erreur afin d'effectuer l'apprentissage des paramètres du modèle par rétro-propagation. Cette architecture est proposée par [38].

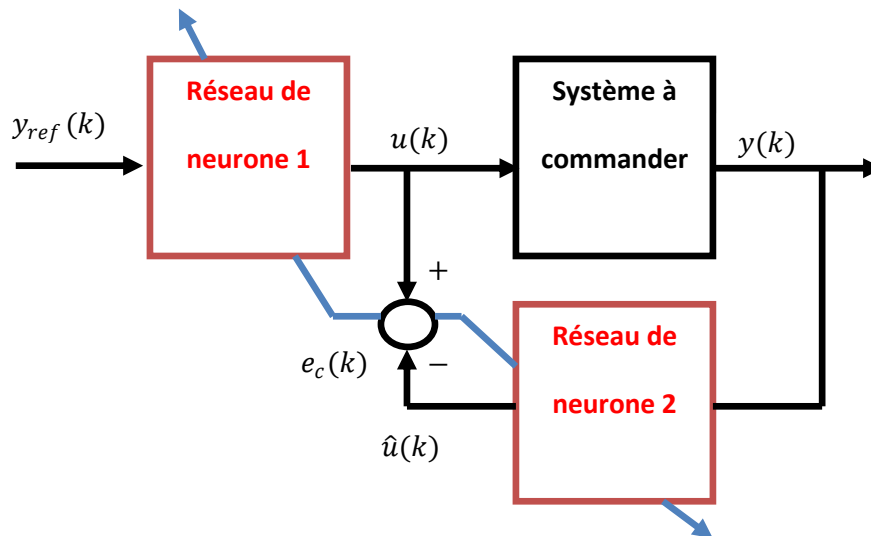


Figure-III-3 : Architecture indirecte d'apprentissage

L'inconvénient de cette architecture, c'est que, une erreur nulle sur la commande ne provoque pas nécessairement que la sortie du système  $y(k)$  soit égale au signal de référence.

➤ **Architecture d'apprentissage spécialisée**

Contrairement aux deux précédentes architectures, le fonctionnement de l'architecture spécialisée consiste à conditionner le modèle inverse en se basant sur l'écart entre la consigne  $y_{ref}(k)$  et la sortie du système  $y(k)$ , **Figure-III-4**.

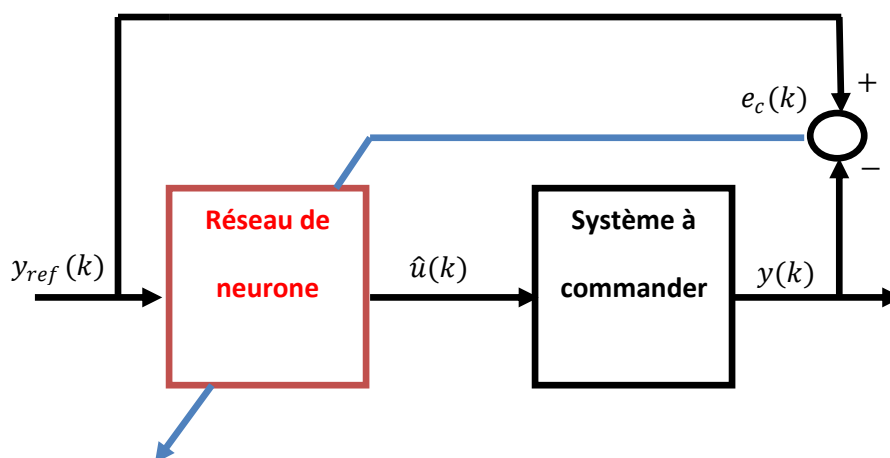


Figure-III-4 : Architecture spécialisée d'apprentissage

### III-3- la commande neuronale

#### III-3-1- la commande neuronale directe par modèle inverse

Cette commande est basée sur l'apprentissage du modèle inverse du système. Cette commande a été utilisée par [38], [41], [42]. Le principe extrêmement simple, est réalisé en plaçant le modèle inverse en amont comme contrôleur pour commander le système [2], [15], [18] (Figure-III-5). Si le modèle neuronal est exactement l'inverse du système commandé, alors il conduit la sortie du ce dernier à suivre la consigne  $y_{ref}(k)$ . Mais elle est très peu mise en œuvre, parce que les performances étant relativement limitées, à cause de l'inconvénient majeur de l'estimation de l'inverse du système, qui n'est pas toujours inversible.

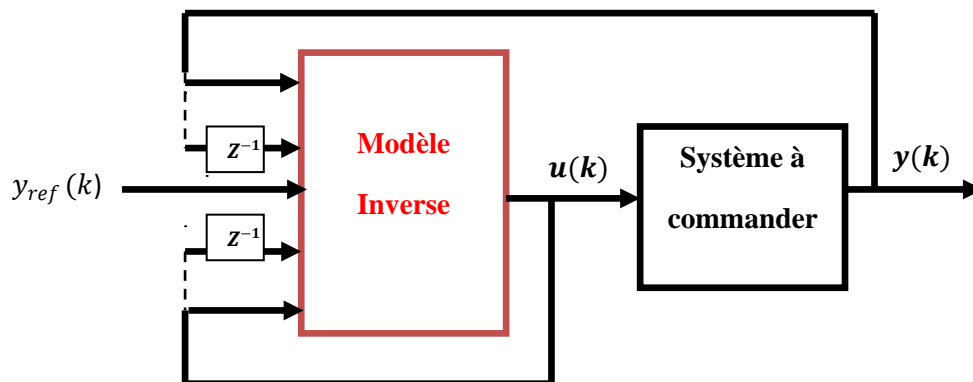
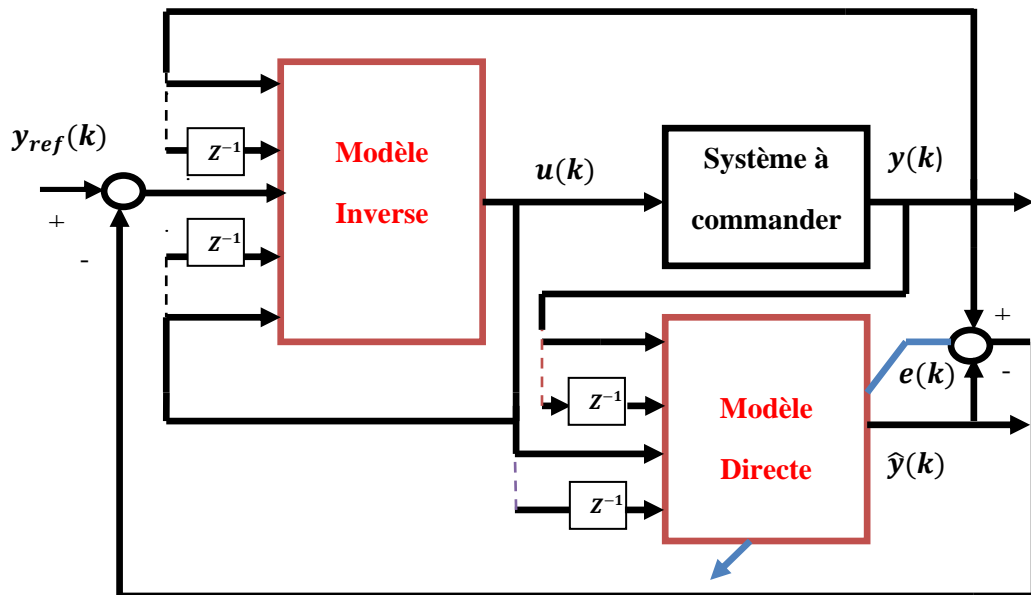


Figure-III-5 : Commande directe par modèle inverse

#### III -3-2- Commande neuronale par modèle interne

Cette structure est similaire à la structure de la commande par modèle inverse. Elle utilise le modèle inverse comme contrôleur et le modèle direct du système placé en parallèle avec le système comme un modèle interne [32].

Le schéma de principe d'une telle commande est donné à la Figure- III -6.



Figure–III-6 : Structure de la commande neuronale par modèle interne

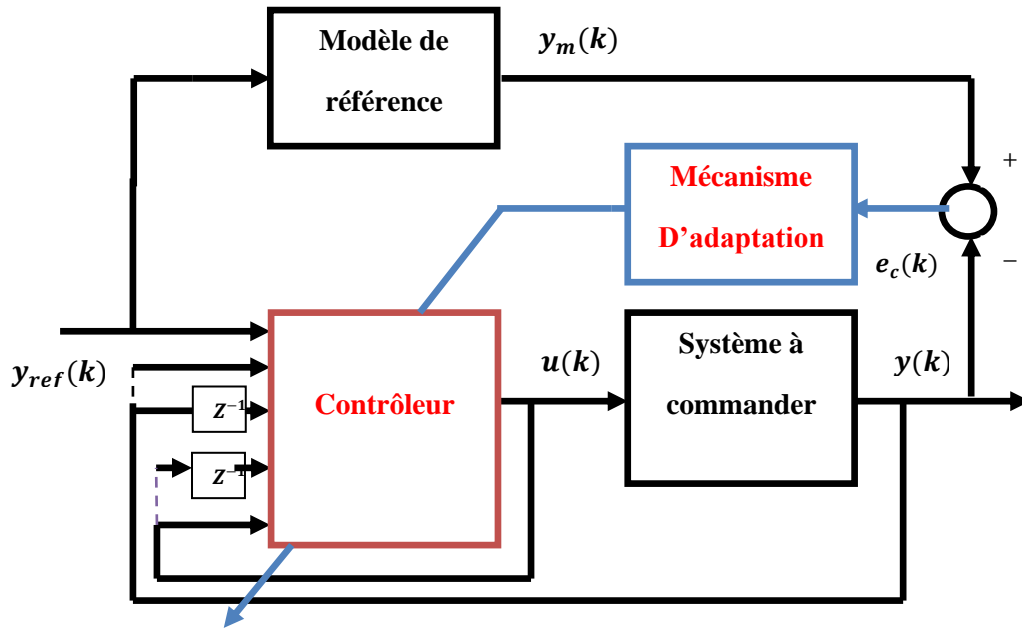
### III-4- la commande adaptative

Il existe deux approches différentes dans la littérature de la commande adaptative, ce sont : la commande adaptative directe et la commande adaptative indirecte [46].

#### III-4-2- la commande adaptative directe

L'objectif est de déterminer une entrée de commande  $u(k)$  à appliquer au système pour que la sortie du système suive celle du modèle de référence (Figure-III-7).

Donc l'erreur  $e_c(k) = y(k) - y_m(k)$  doit être nulle [44], [50], [53].



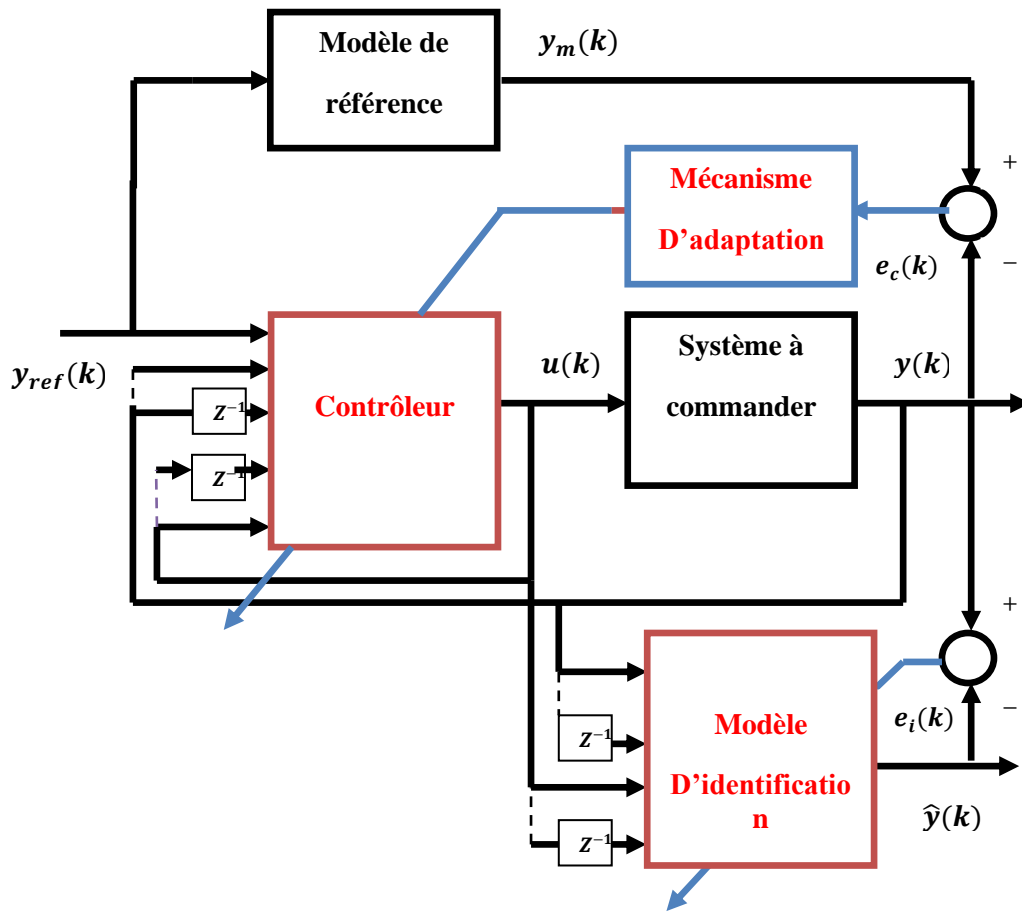
Figure–III-7 : La commande adaptative directe

### III-4-2- la commande adaptative indirecte

Une approche indirecte de la commande adaptative est constituée d'un modèle d'identification (approximateur), il a pour but d'identifier le comportement du système à contrôler, et un contrôleur placé en série avec le système. Cette commande adaptative indirecte est illustrée par la **Figure-III-8**.

Le rôle du contrôleur est de commander la sortie du système de façon à minimiser l'écart  $e_c(k)$  entre la sortie du système  $y(k)$  et la sortie du modèle de référence  $y_m(k)$ .





Figure–III-8 : La commande adaptative indirecte

### III-5- Méthodes d'obtention des lois d'adaptation

Parmi les méthodes utilisées pour la détermination des lois d'adaptation, nous citons la méthode de Lyapunov et l'approche de descente du gradient.

#### ➤ III-5-1- Méthodes de Lyapunov

La méthode de Lyapunov [54] permet d'étudier la stabilité des systèmes complexes qui sont décrits par des systèmes différentiels. La stabilité d'un système est la capacité de ce dernier à revenir à sa position d'équilibre lorsqu'il en est ponctuellement écarté. Il existe deux méthodes pour démontrer la stabilité du système, directe et indirecte.

- **Méthode directe de Lyapunov**

L'étude de la stabilité des systèmes différentiels  $\dot{x} = f(x, t)$  est énoncée par le théorème suivant.

**Théorème :**

Soit  $V(x, t)$  une fonction candidate de Lyapunov qui satisfait les conditions suivantes:

a)  $\forall x \neq 0, V(x) > 0, V(0, t) = 0$  et  $V(x)$  est de classe  $C^1$ .

b) quand  $\|x\| \rightarrow \infty$ , alors  $V(x) \rightarrow \infty$ , fonctions indéfiniment croissantes.

On définit la dérivée de la fonction de Lyapunov par :

$$\dot{V}(x, t) = \frac{\partial V}{\partial t}(x, t) + \left[ \frac{\partial V}{\partial x}(x, t) \right]^T f(x, t) \quad (3.4)$$

$$\text{où } \left[ \frac{\partial V}{\partial x}(x, t) \right]^T \text{ est le gradient de } \frac{\dot{V}(x, t)}{x} = \begin{pmatrix} \frac{\partial V}{\partial x_1}(x, t) \\ \vdots \\ \frac{\partial V}{\partial x_n}(x, t) \end{pmatrix}$$

Si  $\dot{V}(x) < 0$ , alors le système est asymptotiquement stable.

- **Méthode indirecte de Lyapunov**

Soit le système décrit par la représentation d'état suivante:

$$\dot{x} = Ax(t) + Bu(t) \quad (3.5)$$

On considère une fonction candidate de Lyapunov quadratique  $V(x) = x^T Px$ , alors

$$\begin{aligned} \dot{V}(x) &= \dot{x}^T Px + x^T P \dot{x} = (x^T A^T + u^T B^T) Px + x^T P (Ax + Bu) \\ &= x^T (A^T P + PA)x + u^T B^T Px + x^T P Bu \end{aligned} \quad (3.6)$$

Soi la loi  $u(t) = -R^{-1}B^T Px(t)$  qui stabilise le système par retour d'état et minimise le critère  $J$  :

$$J = \int_0^{\infty} x^T Qx + u^T R u dt \quad (3.7)$$

$$\begin{aligned} \dot{V}(k) &= x^T (A^T P + PA)x - x^T P^T B R^{-1} B^T P x - x^T P B R^{-1} B^T P x \\ &= x^T (A^T P + PA - (P^T B R^{-1} + P B R^{-1}) B^T P)x \end{aligned} \quad (3.8)$$

$$P^T = P \text{ alors } \dot{V}(k) = x^T (A^T P + PA - 2P B R^{-1} B^T P)x \quad (3.9)$$

**Théorème :**

Une condition nécessaire et satisfaisante pour que un système  $\dot{x} = Ax(t) + Bu(t)$  soit asymptotiquement stable est que  $\forall Q = Q^T > 0$  et  $R > 0$ , il existe une solution unique  $P$  de l'équation de Riccati, symétrique et définie positive

$$A^T + PA - 2P B R^{-1} B^T P + Q = 0 \quad (3.10)$$

**Remarque :**

Une autre forme de l'équation de Riccati est comme suit :

$$A^T P + PA - 2P A \left( \frac{1}{r} - \frac{1}{2\rho^2} \right) B^T P + Q = 0 \quad (3.11)$$

Avec  $Q = Q^T > 0$  et  $\left( \frac{2}{r} - \frac{1}{\rho^2} \right) \geq 0$ , où  $r$  est une constante positive et  $\rho$  représente un niveau d'atténuation donné. Cette équation est utilisée généralement dans la robustification par  $H_{\infty}$ .

### III-5-2- Méthode de descente du gradient

La méthode de descente de gradient a été utilisée pour la première fois par Whitaker dans son travail original, L'application de cette approche pour l'adaptation revient à utiliser la règle de MIT.

#### Règle de MIT

Soit  $e$  l'erreur entre  $y_m$  et  $y$  ( $e = y_m - y$ ) et  $\theta$  le vecteur des paramètres à ajuster. Un critère à minimiser est proposé comme suit :

$$J(\theta) = \frac{1}{2} e^2 \quad (3.12)$$

Par conséquent, pour que  $J$  soit petit, il est raisonnable de changer les paramètres dans le sens négatif du gradient de  $J$ .

$$\begin{cases} \frac{d\theta}{dt} = -\gamma \cdot \frac{\partial J}{\partial \theta} \\ \frac{d\theta}{dt} = -\gamma \cdot e \cdot \frac{\partial e}{\partial \theta} \end{cases} \quad (3.13)$$

$\frac{\partial e}{\partial \theta}$  : représente la sensibilité du système,  $\gamma$  détermine la vitesse d'adaptation des paramètres.

### III -6- la commande adaptative neuronale

La commande adaptative des systèmes non linéaires inconnus nécessite l'utilisation d'un autre outil tel que les réseaux neurones afin d'appliquer les deux techniques conventionnelles de la commande adaptative, on parle alors de la commande adaptative neuronale.

Deux structures de contrôle adaptatif neuronal à modèle de référence ont été proposées par Narendra et Parthasarathy [47]; la structure de contrôle direct et la structure de contrôle indirect [2] [15] [16] [17].

### III-6-1- La commande neuronale adaptative directe

La commande neuronale adaptative directe utilise un système neuronal comme contrôleur, placé en cascade avec le système à commander.

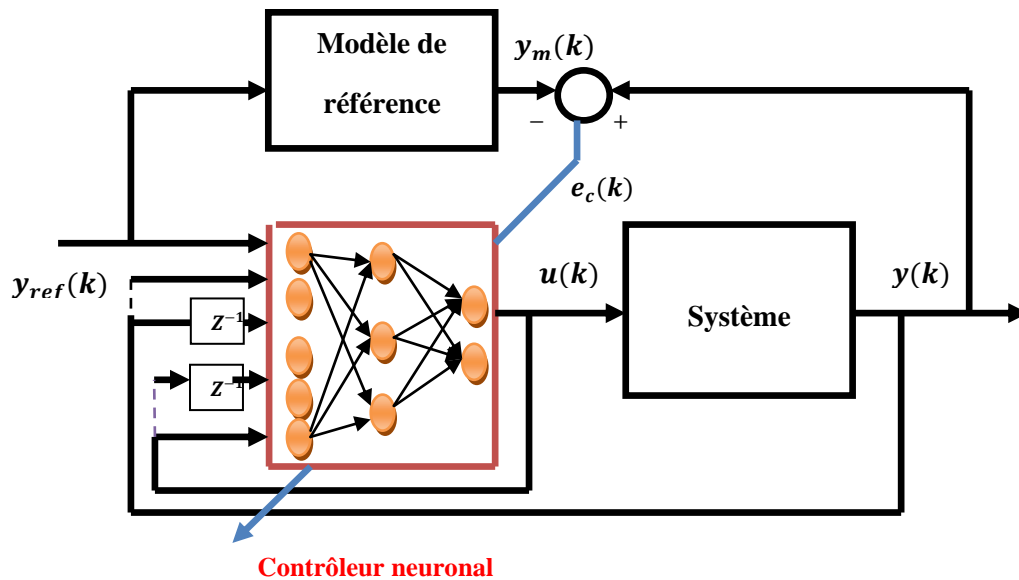


Figure-III-9 : Structure de la commande neuronale adaptative directe

Prenons par exemple un système qu'on veut commander qui s'écrit sous la forme suivante:

$$y(k + 1) = f(y(k), \dots, y(k - n), u(k), \dots, u(k - m)) \quad (3.14)$$

Avec un modèle de référence:

$$y_m(k + 1) = f(y_m(k), r(k)) \quad (3.15)$$

La sortie du contrôleur neuronal représente une commande à appliquer à l'entrée du système, celle-ci est donnée par:

$$u(k + 1) = NN(y_m(k), y(k), u(k)) \quad (3.16)$$

Les paramètres du contrôleur neuronal sont directement ajustés en se basant sur l'erreur  $e_c$  :

$$e_c(k + 1) = y(k + 1) - y_m(k + 1) \quad (3.17)$$

Cette méthode de commande présente un inconvénient quand le système à contrôler est inconnu, puisque on ne peut pas calculer les changements des paramètres du contrôleur.

### III -6-2- La commande neuronale adaptative indirecte

Ce type de commande utilise une combinaison de deux réseaux de neurones **Figure-III-10**. Le premier réseau est entraîné pour identifier le modèle dynamique du système. Le deuxième réseau de neurones est entraîné pour générer une commande nécessaire à appliquer au système pour que sa sortie suive celle du modèle de référence.

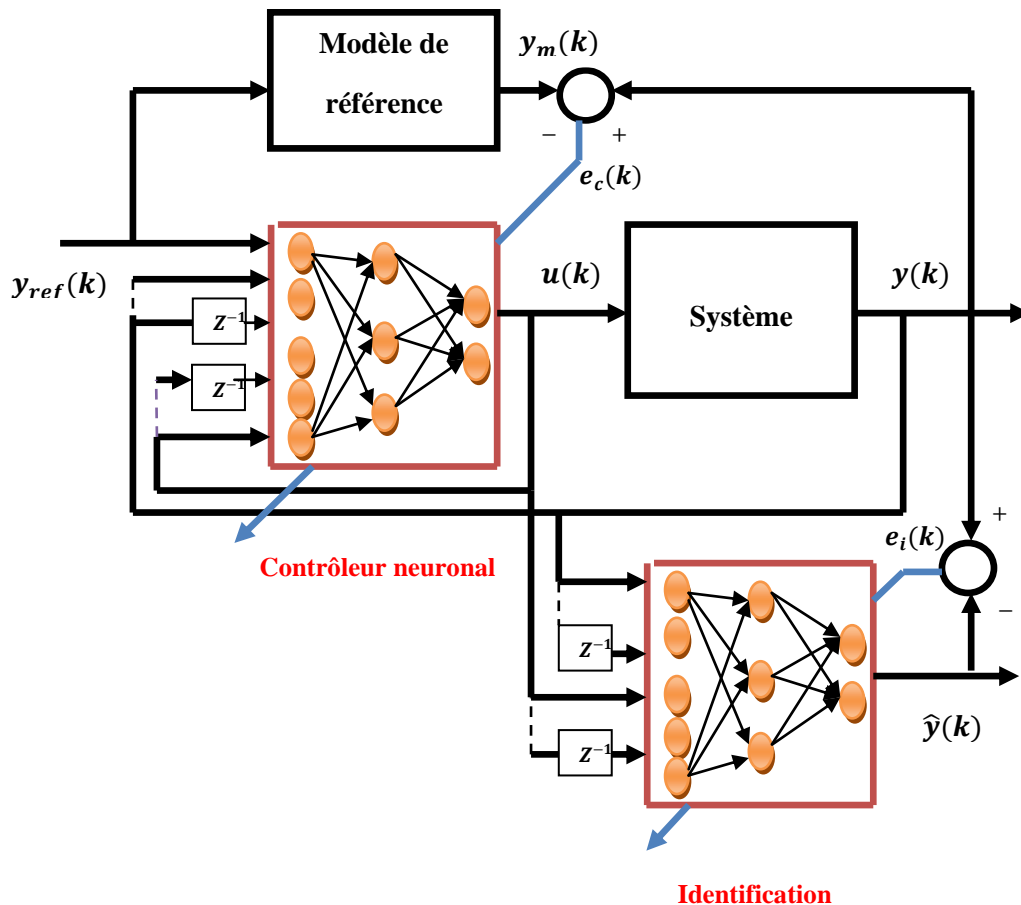


Figure-III-10 : Structure de la commande neuronale adaptative Indirecte

Le signal de commande généré par le contrôleur neuronal  $u(k)$  est utilisé comme une entrée pour le modèle neuronal et le système commandé. L'erreur  $e_i(k)$  entre la sortie du système  $y(k)$  et celle du modèle  $\hat{y}(k)$  sera utilisée pour ajuster les paramètres du modèle neuronal d'une part, et d'autre part, l'erreur  $e_c(k)$  entre la sortie du modèle de référence  $y_m(k)$  et la sortie du système commandé sera utilisée pour adapter les paramètres du contrôleur neuronal. Ce qui signifie que la qualité de la poursuite dépend largement de la convergence du modèle identifié.

L'inconvénient de cette technique est la complexité de la structure de commande avec deux réseaux et le temps nécessaire aux calculs et aux ajustements des paramètres.

Si on prend un système non linéaire sous la forme :

$$y(k + 1) = f(y(k), \dots, y(k - n), u(k), \dots, u(k - m)) \quad (3.18)$$

La structure du modèle d'identification neuronal sera donnée par l'équation suivante:

$$\hat{y}(k + 1) = NN(y(k), \dots, y(k - n), u(k), \dots, u(k - m)) \quad (3.19)$$

Avec  $\hat{y}(k)$  la sortie du modèle d'identification.

L'ajustement des paramètres du modèle neuronal est basé sur l'erreur d'identification.

Les paramètres du contrôleur neuronal sont ajustés en se basant sur l'erreur  $e_c(k + 1)$  :

Avec

$$e_c(k + 1) = y(k + 1) - y_m(k + 1) \quad (3.20)$$

Nous utilisons alors le modèle neuronal identifié, quand le système à commander est inconnu, dont l'expression de l'erreur est:

$$e_c(k + 1) = \hat{y}(k + 1) - y_m(k + 1)$$

## **Conclusion**

Les réseaux de neurones ont été souvent appliqués dans l'identification et la commande du système. Ils possèdent la propriété d'approximation universelle pour tout système dynamique non linéaire.

Deux structures de commande neuronales sont étayées : la commande par modèle inverse et la commande par modèle interne. Pour l'apprentissage du modèle inverse, plusieurs architectures ont été données.

Ensuite nous avons donné un aperçu sur le principe général de la commande adaptative neuronale.

En général, les contrôleurs neuronaux adaptatifs sont de deux types : la commande neuronale adaptative directe et la commande neuronale adaptative indirecte. Dans la commande neuronale adaptative directe, le système neuronal est utilisé comme un contrôleur adaptatif. Cependant, dans la commande neuronale adaptative indirecte, Nous avons besoin d'identifier le système avant d'être commandé.



# Chapitre IV

## Simulation

## IV-1- Introduction

La commande du bioréacteur est l'une des préoccupations majeures des recherches dans le traitement des eaux usées. En effet, un bioréacteur est caractérisé par un comportement purement non linéaire.

Des recherches avancées ont permis de développer de nouvelles techniques de commande non linéaire telle que la commande neuronale adaptative pour les applications aux bioréacteurs.

Dans ce chapitre, nous appliquons la technique de commande neuronale adaptative pour la commande du substrat d'un bioréacteur à suivre certain référence.

## IV-2- Modèle du Bioréacteur

Le procédé de traitement à boues activées reste le système le plus utilisé dans le cas du traitement des eaux usées. L'épuration biologique des eaux usées a lieu dans les réacteurs biologiques. Dans cette section, le modèle réduit d'un réacteur à boues activées est présenté dans l'équation (4.1). L'élimination biologique de la pollution carbonée s'effectue au sein de deux unités principales: un réacteur biologique et un décanteur (Figure-IV-1).

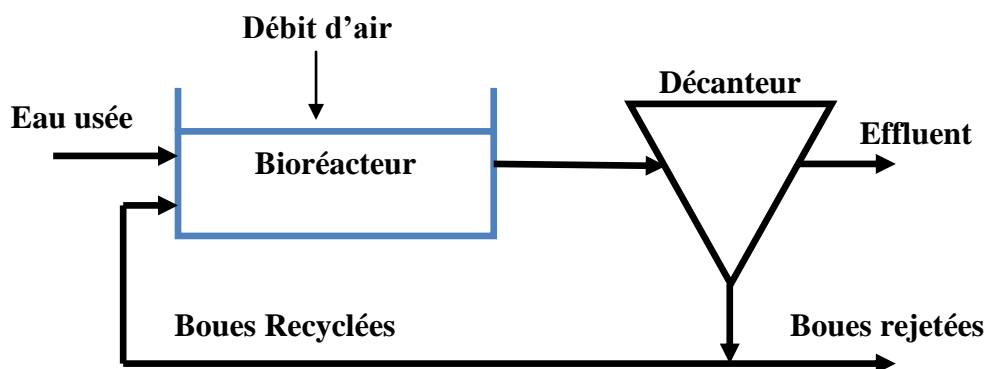


Figure-IV-1 : Schéma d'un bioréacteur biologique

On considère un modèle de bioréacteur avec un volume constant  $V$  [25]. Les équations dynamiques qui déterminent le modèle [26], [27] peuvent être écrites sous la forme :

$$\begin{cases} \dot{S} = -\frac{1}{Y} \cdot \mu_{max} \frac{S}{S+K_S} X + \frac{Q}{V} \cdot (S_{in} - S) \\ \dot{X} = \mu_{max} \frac{S}{S+K_S} \cdot X - \frac{Q}{V} \cdot X \end{cases} \quad (4.1)$$

Le volume  $V$  est constant dans toutes ses phases dans lequel une réaction microbiologique a lieu là où la biomasse  $X$  croit sur un substrat  $S$ .

$$\mu(S) = \mu_{max} \frac{S}{S + K_S} \quad (4.2)$$

Avec  $\mu(S)$  est la loi de Monod.

$Y$  : Rendement.

$Q(k)$  : Débit d'entrée du bioréacteur.

$\mu_{max}$  : Le taux maximum de croissance.

$K_S$  Coefficient de demi-saturation du substrat facilement biodégradable

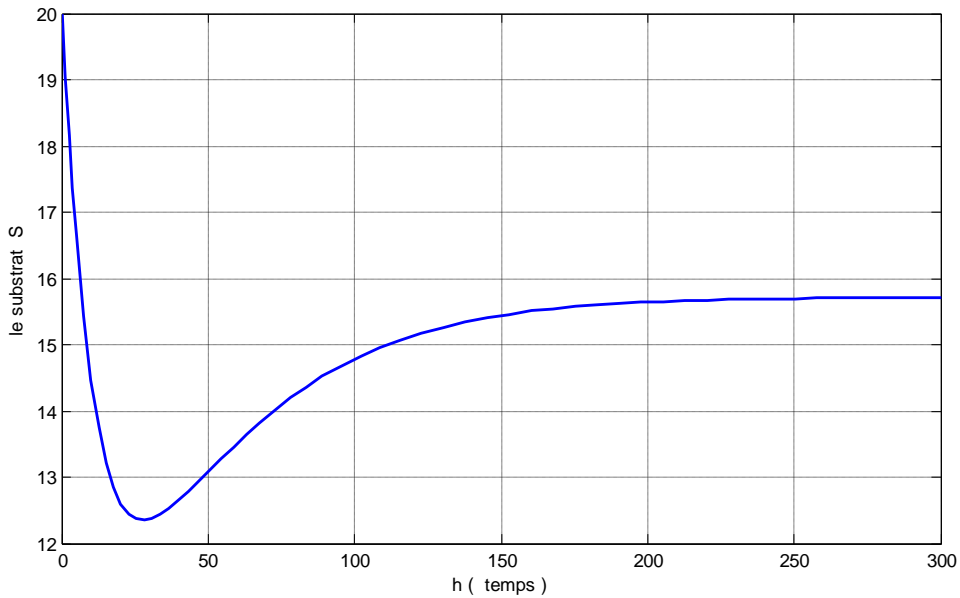
$D(k) = \frac{Q(k)}{V} \in [0, D_{max}]$  . On le considère ici comme variable d'entrée pour le bioréacteur [25].

Paramètres	valeurs
$S_{in}$	$100 \text{ gl}^{-1}$
$Y$	0.1
$V$	40 l
$\mu_{max}$	0.045
$K_S$	$10 \text{ gl}^{-1}$

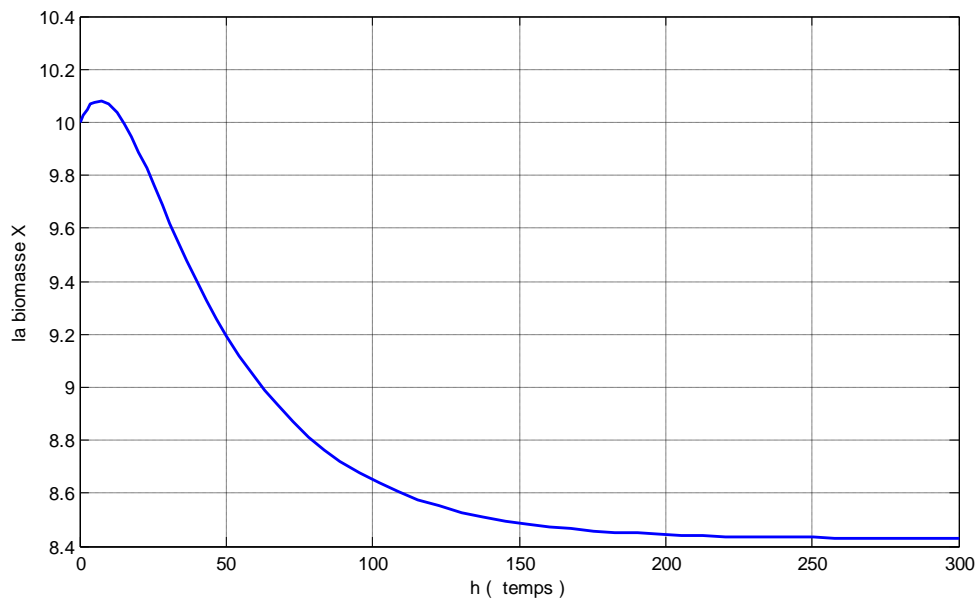
**Table-IV-1: Valeurs des paramètres du bioréacteur**

On supposera que la concentration initiale du substrat varie entre  $(10 \leq S(k_0) \leq 20)$  pour cela, on considère deux conditions initiales différentes  $S(k_0) = 20 \text{ gl}^{-1}$  et  $S(k_0) = 10 \text{ gl}^{-1}$  , et une condition initiale de micro-organismes  $X(k_0) = 8. \text{ gl}^{-1}$ .

La figure -IV-2 montre l'évolution de la sortie du bioréacteur  $S(k)$  et  $x(k)$  pour une entrée constante  $D(k) = 0.0275 \text{ h}^{-1}$ .



(a)

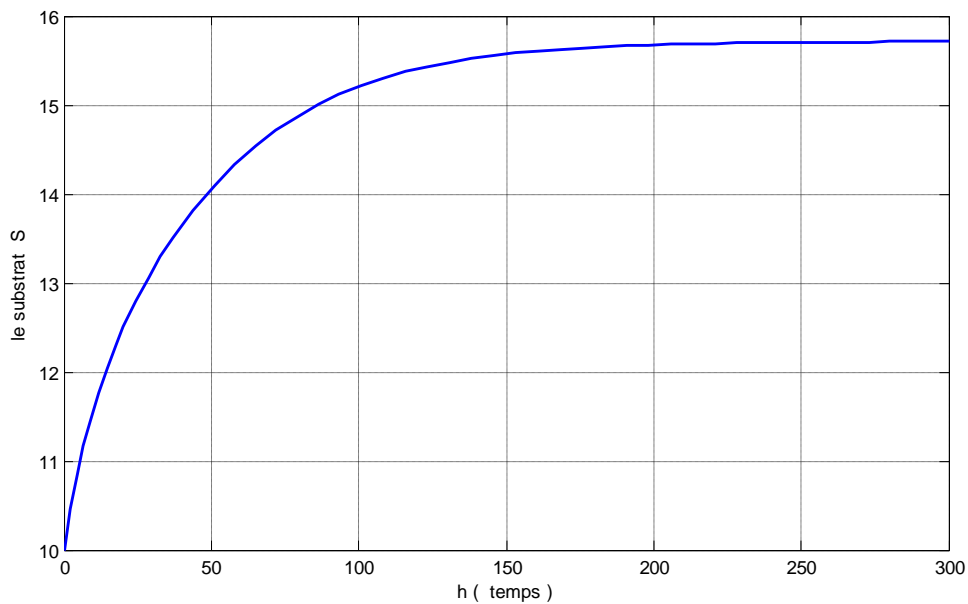


(b)

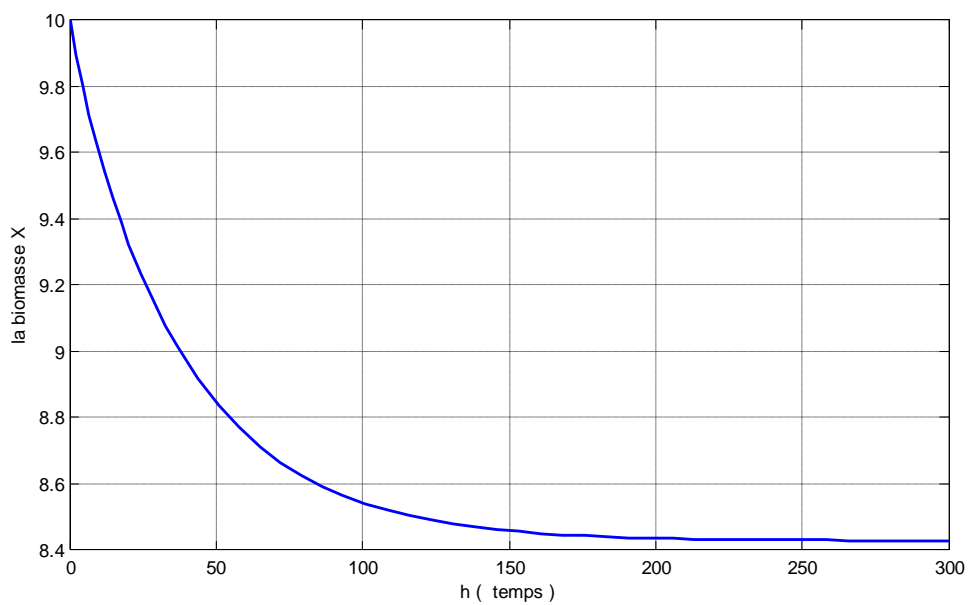
Figure-IV-2: (a), (b) L'Evolution du substrat et de la biomasse du bioréacteur

La figure montre la concentration du substrat et de la biomasse par passage d'une valeur initiale à une valeur finale stabilisée.

Pour la conditions initiale  $S(k_0) = 10 \text{ gl}^{-1}$  et  $X(k_0) = 8 \text{ gl}^{-1}$ . L'évolution de la sortie du bioréacteur  $S(k)$  et  $x(k)$  est alors donnée à la **Figure -IV-3** :



(a)



(b)

**Figure-IV-3: (a), (b) L'Evolution du substrat et de la biomasse du bioréacteur**

Selon la **Figure -IV-3** on remarque que la concentration du substrat augmente à partir de la valeur initiale vers la valeur finale.

Nous avons le même constat pour la stabilisation des paramètres  $S(k)$  et  $X(k)$ .

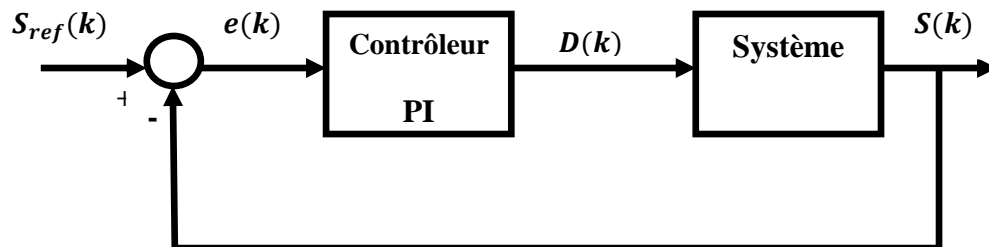
### IV-3- La commande par PI classique

Une commande classique par PI est élaborée afin de comparer ultérieurement les résultats avec ceux relatifs aux commandes adaptative neuronales (**Figure -IV-4**).

La loi de commande est définie dans ce cas comme suit :

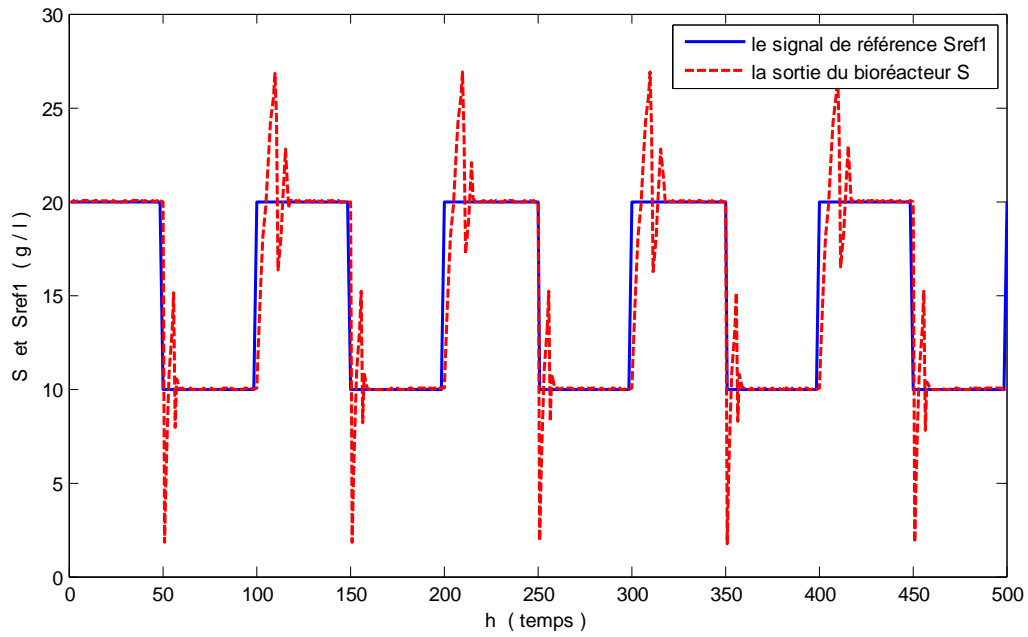
$$u_1(k) = u_1(k - 1) + k_i |y_{ref}(k) - y(k)| - k_p (y(k) - y(k - 1)) \quad (4.3)$$

Dans un premier temps nous allons considérer que le bioréacteur dans un intervalle déterminé se comporte comme un système linéaire, Dans ce cas la commande classique par PI peut être utilisée.

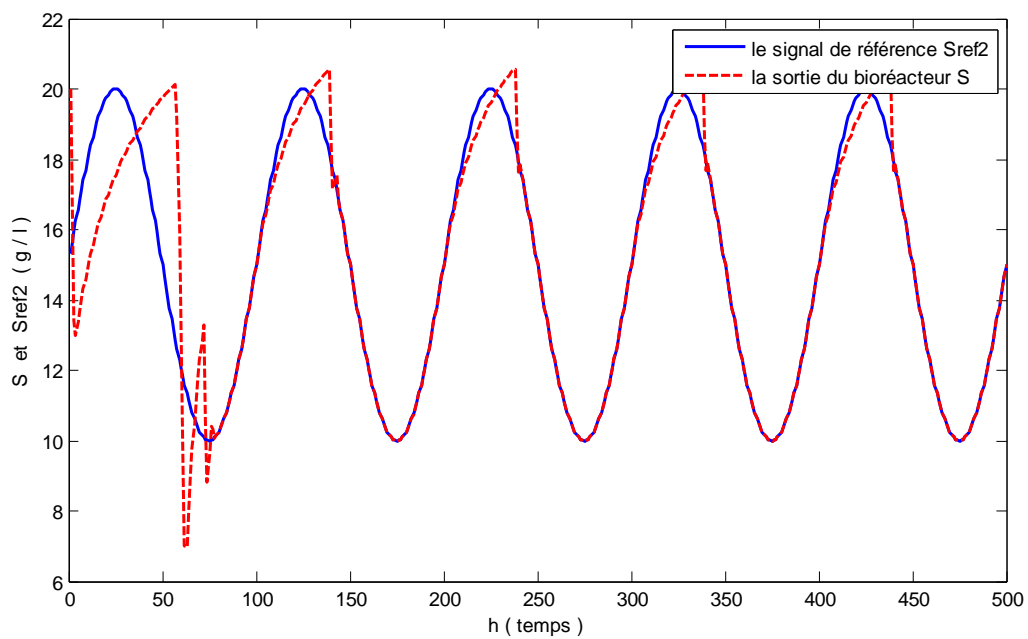


**Figure-IV-4: commande par PI**

La **Figure -IV-5** montre l'évolution du substrat dans le bioréacteur en utilisant la commande Pi avec deux signaux de référence.



(a)



(b)

Figure-IV-5: (a), (b) Evolution de la sortie  $S(k)$ , de la consigne  $S_{ref1}(k)$  et  $S_{ref2}(k)$  g n r  par PI

Le critère utilisé pour comparer les différentes stratégies de commande est le suivant :

$$E_C = \frac{1}{N} \sum_{k=1}^N (S_{ref} - S(k))^2 \quad (4.4)$$

#### IV-4- La commande adaptative neuronale du Bioréacteur

Les régulateurs classiques ne sont pas adaptés aux systèmes non linéaires comme le bioréacteur, ils sont conçus pour des systèmes linéaires sans retard. On doit donc utiliser une stratégie de commande à base de réseaux de neurones. Ces derniers sont des techniques puissantes de traitement non linéaire de données et qui ont fait leurs preuves dans de nombreux domaines d'application.

##### IV-4-1- La commande adaptative neuronale directe

Cette approche est basée sur la commande neuronale directe avec modèle de référence. La commande neuronale peut être référencée ou pas selon qu'elle se base ou non sur un modèle de référence. Sur la **Figure-IV-6**, le schéma de commande n'est pas référencé par un modèle de comportement.

Dans ce schéma, les paramètres du contrôleur neuronal, sont les poids et les biais du réseau de neurones directement ajustés grâce à une erreur calculée entre la sortie du signal de référence et celle du système,  $e(k) = y_{ref}(k) - y(k)$ .

Ce contrôleur neuronal génère une commande  $u(k)$  qui sera appliqué au bioréacteur. La sortie doit suivre une certaine référence



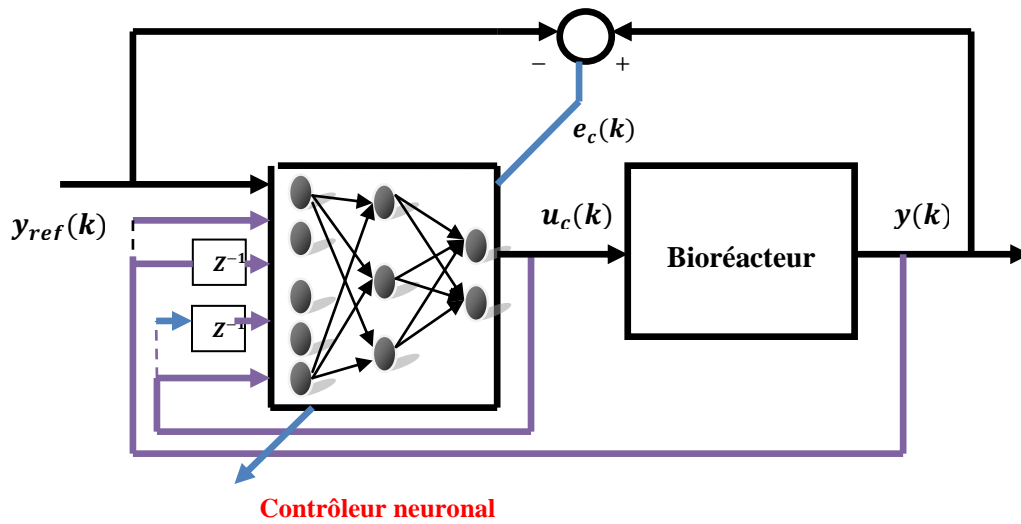


Figure-IV-6 : Structure de la commande neuronale directe

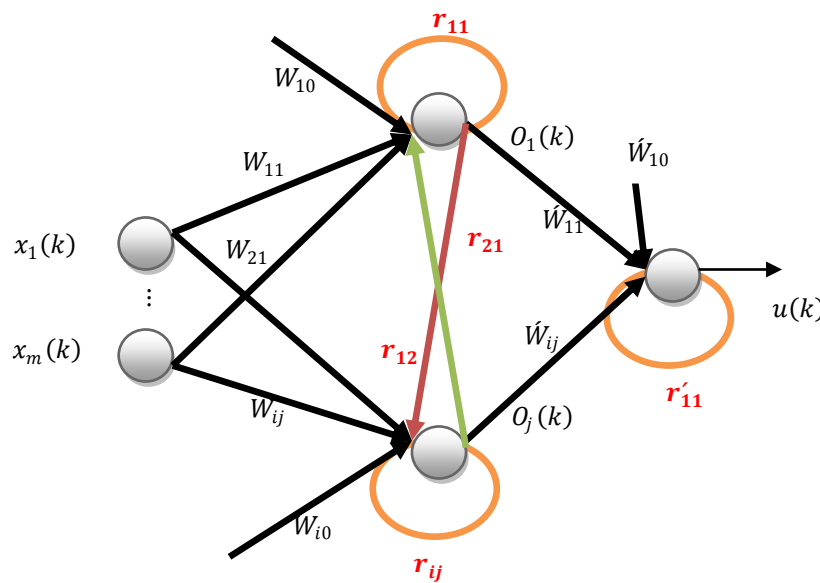


Figure-IV-7 : Schéma du contrôleur neuronal

Les connexions des poids sont ajustés par les équations suivantes :

$$J(k) = \frac{1}{2} \sum_{k=1}^N (y_{ref}(k) - y(k))^2 \tag{4.5}$$

$$\Delta W_{ij}(k) = \Delta W_{ij}(k-1) - \eta \frac{\partial J(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial W_{j,i}} \tag{4.6}$$

$$\Delta \hat{W}_{ij}(k) = \Delta \hat{W}_{ij}(k-1) - \eta \frac{\partial J(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial \hat{W}_{ij}} \tag{4.7}$$

$$\Delta r_{ij}(k) = \Delta r_{ij}(k-1) - \eta \frac{\partial J(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial r_{ij}} \quad (4.8)$$

$$\Delta \dot{r}_{ij}(k) = \Delta \dot{r}_{ij}(k-1) - \eta \frac{\partial J(k)}{\partial y(k)} \cdot \frac{\partial y(k)}{\partial u(k)} \cdot \frac{\partial u(k)}{\partial \dot{r}_{ij}} \quad (4.9)$$

$$\frac{\partial J(k)}{\partial y(k)} = -e_c(k) \text{ avec } \frac{\partial y(k)}{\partial u(k)} = J_c(k)$$

Donc, les équations (4.6) (4.7) (4.8) et (4.9) deviennent :

$$\Delta W_{ij}(k) = \Delta W_{ij}(k-1) - \eta e_c(k) \cdot J_c(k) \cdot \frac{\partial u(k)}{\partial W_{j,i}} \quad (4.10)$$

$$\Delta \dot{W}_{ij}(k) = \Delta \dot{W}_{ij}(k-1) - \eta e_c(k) \cdot J_c(k) \cdot \frac{\partial u(k)}{\partial \dot{W}_{ij}} \quad (4.11)$$

$$\Delta r_{ij}(k) = \Delta r_{ij}(k-1) - \eta e_c(k) \cdot J_c(k) \cdot \frac{\partial u(k)}{\partial r_{ij}} \quad (4.12)$$

$$\Delta \dot{r}_{ij}(k) = \Delta \dot{r}_{ij}(k-1) - \eta e_c(k) \cdot J_c(k) \cdot \frac{\partial u(k)}{\partial \dot{r}_{ij}} \quad (4.13)$$

$$J_c(x) = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (4.14)$$

Où  $f$  est la sortie et  $x$  est l'entrée est au système. Dans notre cas, parce que nous traitons avec une entrée et une sortie le Jacobien est un scalaire. Une façon d'approcher cette durée est en prenant le rapport de la différence entre le courant de sortie  $y(k)$  et la sortie précédente du système  $y(k-1)$  par la différence entre le courant d'entrée  $u(k)$  et l'entrée précédente du système  $u(k-1)$ . Cette approximation devient plus précise, si le temps d'échantillonnage est suffisamment faible.

$$J_c(K) = \frac{\partial y(k)}{\partial u(k)} \approx \frac{y(k) - y(k-1)}{u(k) - u(k-1)} \quad (4.15)$$

L'algorithme de commande utilisé est le suivant :

- Initialisation du procédé,  $k = 1$ ,

- Acquisition de la sortie  $y(k)$ ,
- Calcul du vecteur d'entrée du contrôleur neuronal,  $[y_{ref}(k), y(k), y(k-1), u(k-1)]$ ,
- Calcul de la loi de commande (sortie du contrôleur),  $u_c(k)$ ,
- Application de la commande  $u_c(k)$  sur l'entrée du système,
- Calcul de la sortie  $y(k)$ ,
- $k = k + 1$ , retour à l'étape deux.

#### IV-4-2- La commande adaptative neuronale indirecte

Cette structure est similaire à la structure de la commande directe. Elle utilise le contrôleur neuronal et le modèle neuronal direct placé en parallèle avec le système commandé comme un modèle interne [32]. Ce principe est illustré sur la Figure-IV-8.

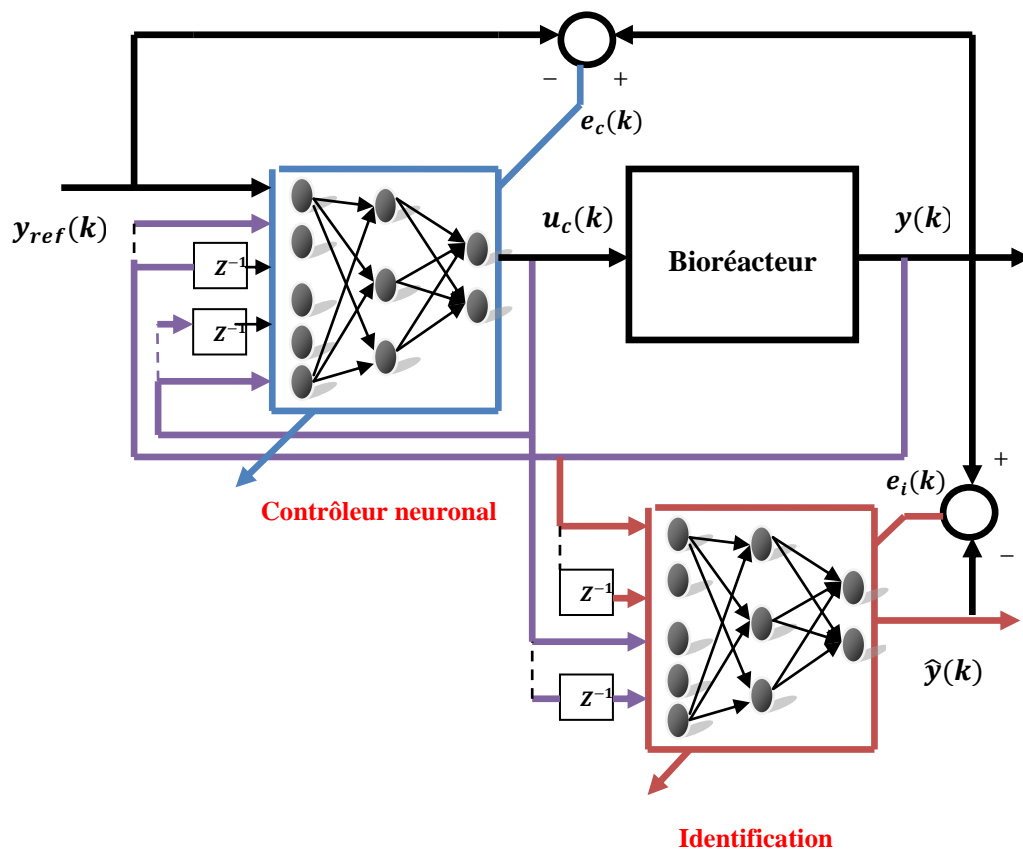


Figure-IV-8 : Structure de la commande neuronale indirecte

Les deux modèles connexionnistes présents dans ce schéma ont des rôles bien distincts. L'un estime le système en se basant sur les entrées et les sorties de ce dernier, le second est utilisé en tant que contrôleur pour commander le système.

Les paramètres du contrôleur neuronal sont ajustés de façon à ce que le système réel suive le comportement du signal de référence. [20]. Le signal de commande  $u(k)$  généré par le contrôleur neuronal est utilisé comme une entrée pour le modèle neuronal direct et le système commandé, d'une part, l'erreur  $e_i(k)$  entre la sortie du système  $y(k)$  et celle du modèle  $\hat{y}(k)$  sera utilisée pour ajuster les paramètres du modèle neuronal. D'autre part, l'erreur  $e_c(k)$  entre la sortie du signal de référence  $y_{ref}(k)$  et la sortie du système commandé  $y(k)$  sera utilisée pour adapter les paramètres du contrôleur neuronal.

L'algorithme de commande utilisé est le suivant :

- Initialisation du procédé,  $k = 1$ ,
- Acquisition de la sortie  $y(k)$ ,
- Calcul du vecteur d'entrée du modèle direct,  $[u(k), u(k-1), y(k), y(k-1), ]$ ,
- Calcul de la sortie du modèle direct  $y_m(k)$ ,
- Calcul de l'erreur  $(k) = y(k) - y_m(k)$ ,
- Calcul du vecteur d'entrée du contrôleur neuronal,  $[y_{ref}(k), y_m(k), y_m(k-1), u(k-1)]$ ,
- Calcul de la loi de commande (sortie du contrôleur),  $u(k)$ ,
- Application de la commande  $u(k)$ ,
- $k = k + 1$ , retour à l'étape deux.

### IV-4-3- La Commande hybride

Cette structure de commande comporte deux contrôleurs placés en parallèle, un contrôleur neuronal et un contrôleur PI classique. Le signal de commande est

constitué de la somme des sorties des deux contrôleurs. Le contrôleur neuronal sert à modéliser les dynamiques inverses du système. Le contrôleur PI sert à compenser l'effet des perturbations et des incertitudes. La structure fonctionnelle de la stratégie de commande est donnée à la Figure-IV-9.

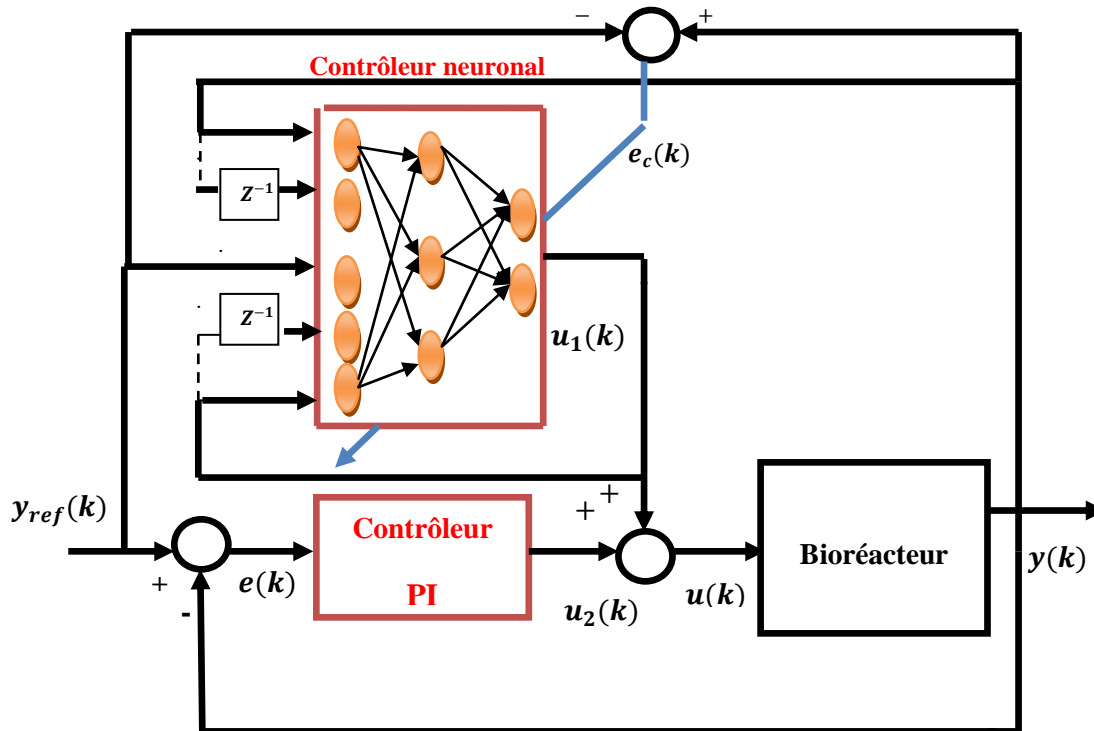


Figure-IV-9 : Structure d'une commande hybride

L'algorithme de commande utilisé est le suivant :

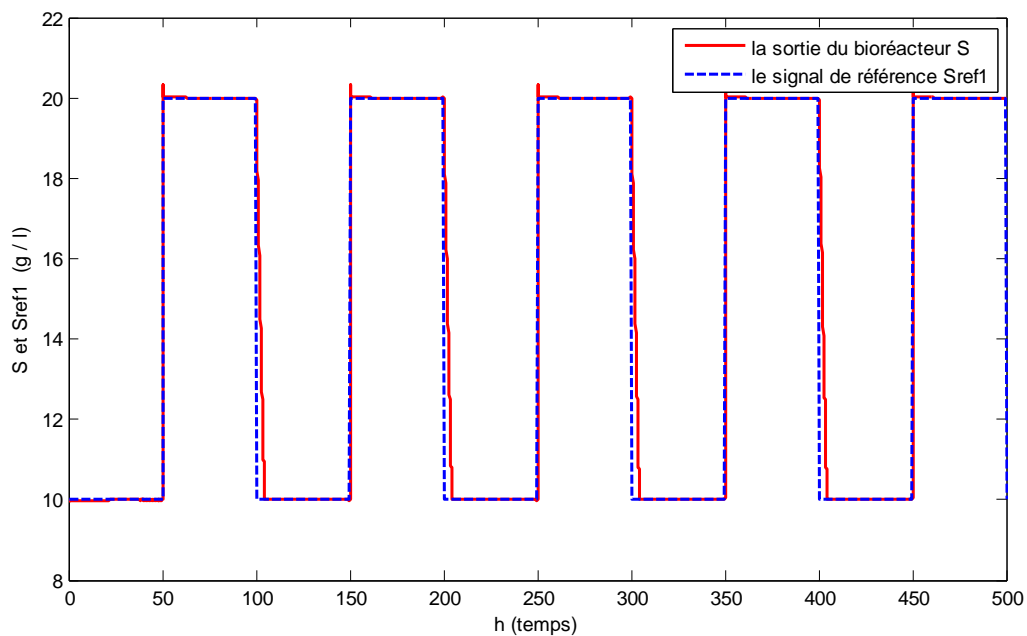
- Initialisation du système,  $k = 1$ ,
- calcul la sortie,  $y(k)$
- Calcul de l'entrée du PI :  $e(k) = y_{ref}(k) - y(k)$
- Calcul de la sortie du PI
 
$$u_1(k) = u_1(k-1) + k_i e(k) - k_p (y(k) - y(k-1))$$
- Calcul le vecteur d'entrée du contrôleur neuronal.
- Calcul de la loi de commande :  $u(k) = u_1(k) + u_2(k)$

- Application de la commande  $u(k)$
- $k = k + 1$ , retour à la deuxième étape.

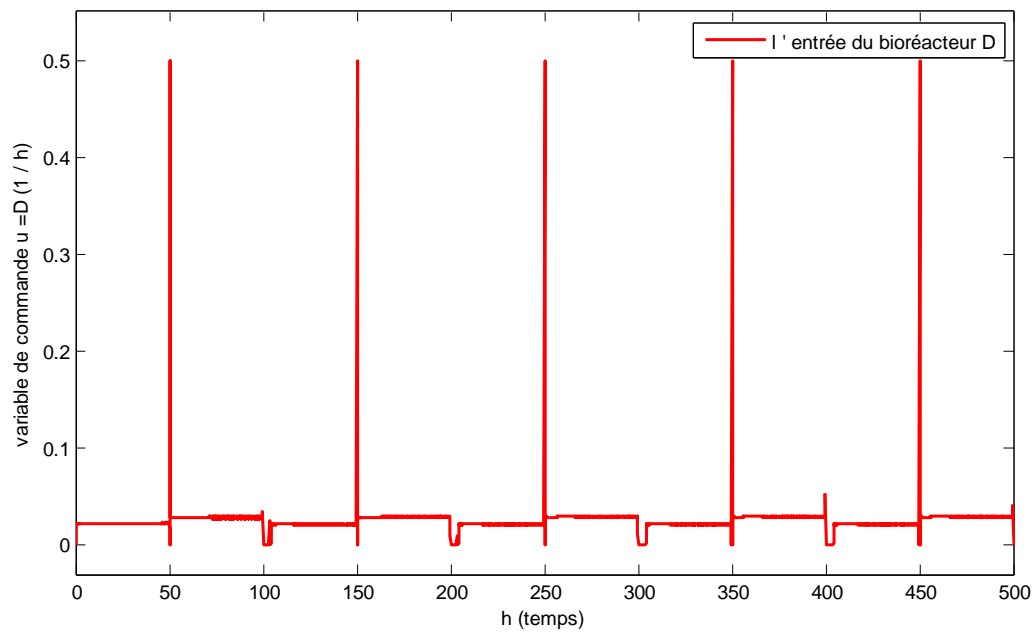
#### IV-5- Résultats de la simulation

##### ➤ La commande neuronale adaptative directe

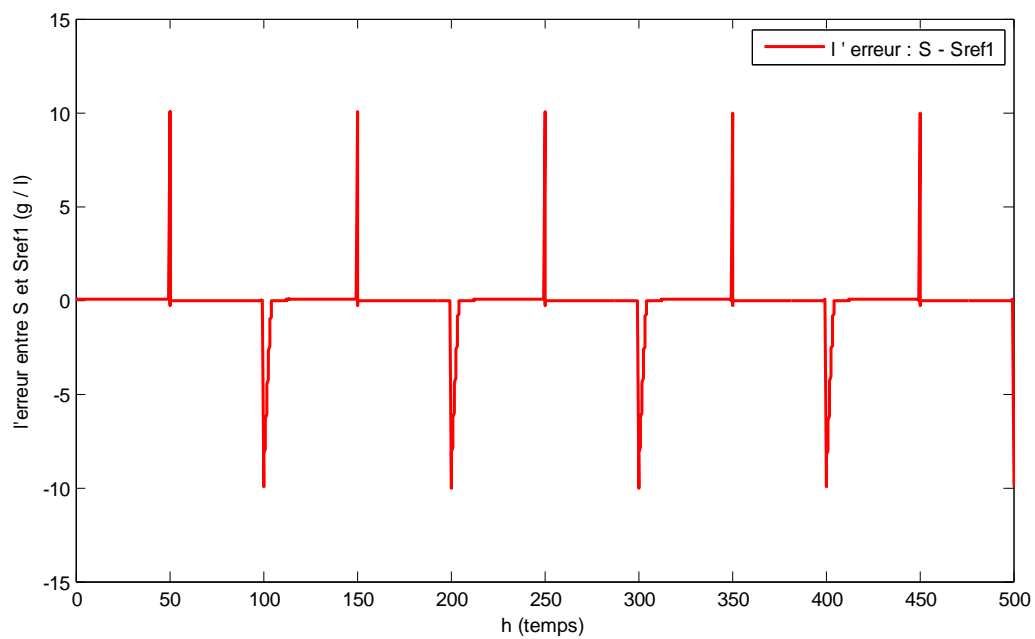
Les résultats obtenus par l'approche directe sont donnés dans La **Figure -IV-10**. Qui comporte l'évolution de la sortie réelle du bioréacteur, de la consigne et du signal de commande. Nous avons donc utilisé deux signaux de référence ref1 et ref2.



(a)

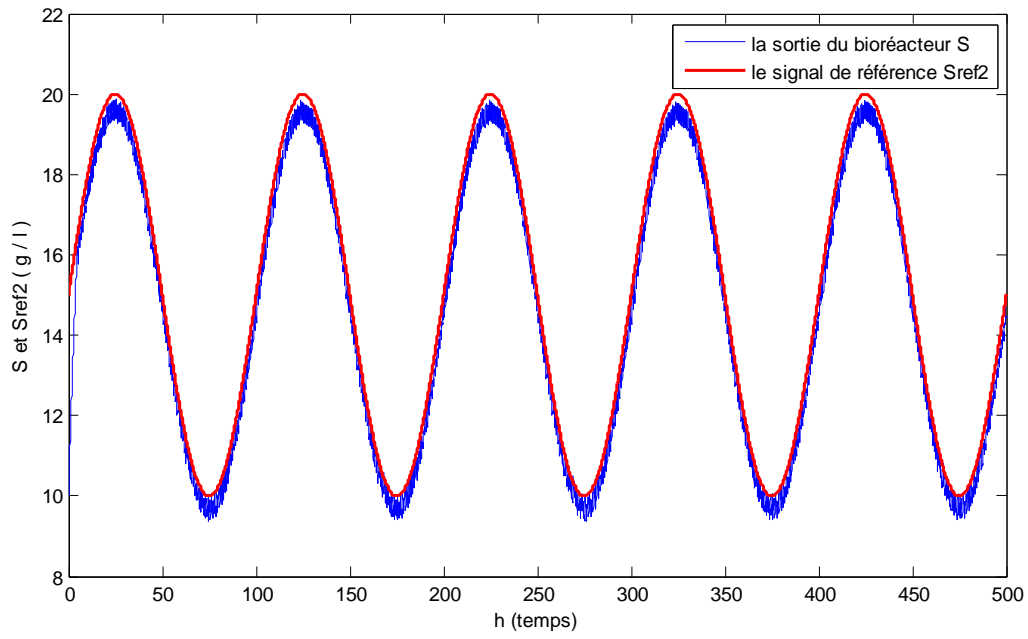


(b)

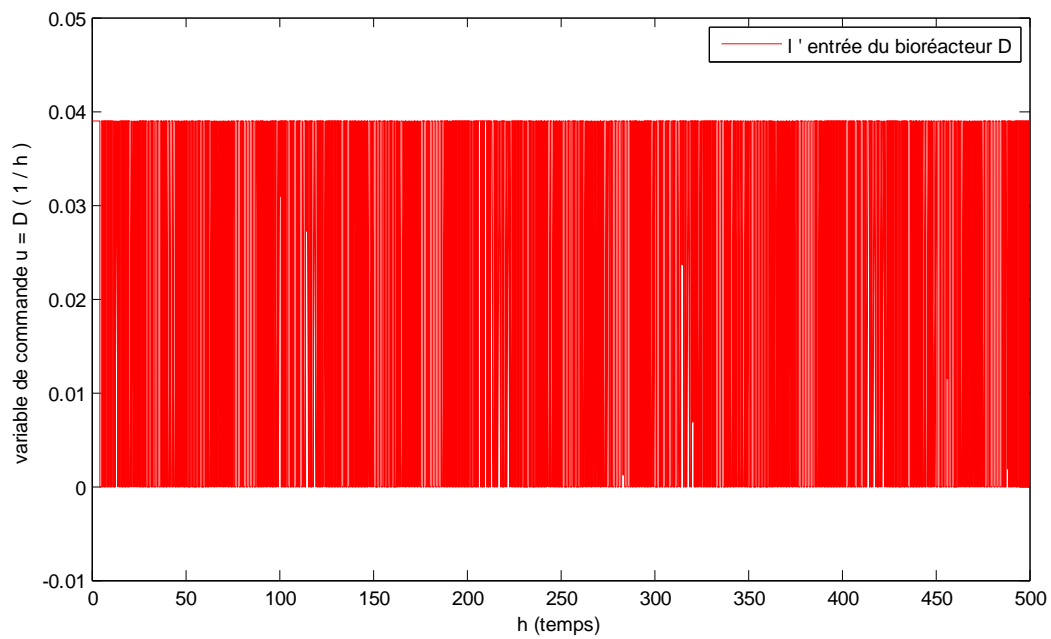


(c)

Figure-IV-10: (a) Evolutions du substrat  $S(k)$ , de la consigne  $S_{ref1}(k)$ , (b) signal de commande  $u(k)$ , (c) l'erreur entre  $S(k)$  et  $S_{ref1}(k)$

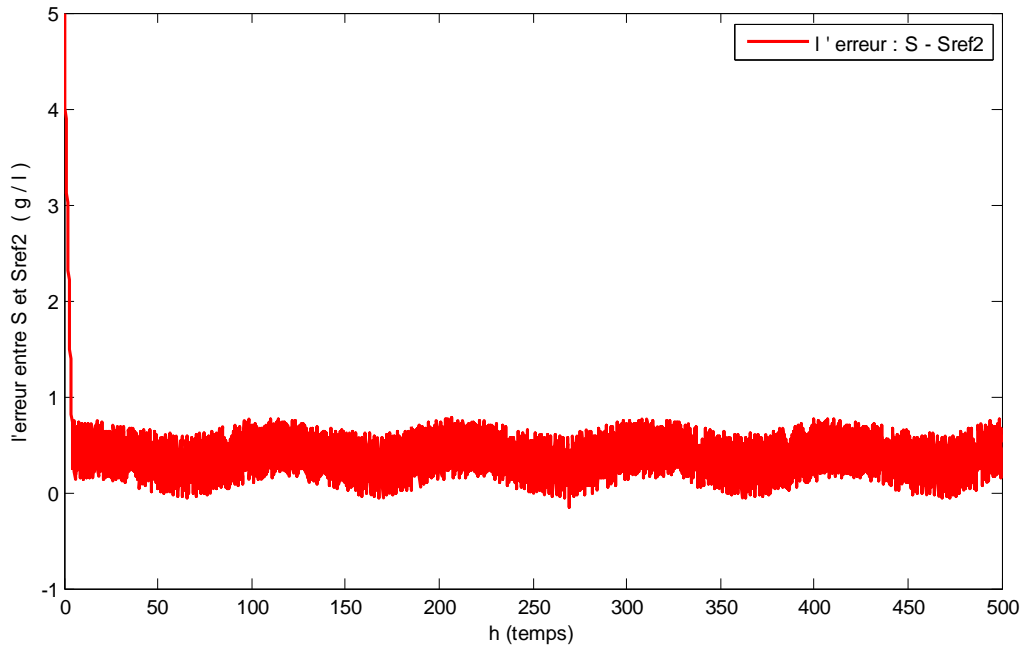


(a)



(b)



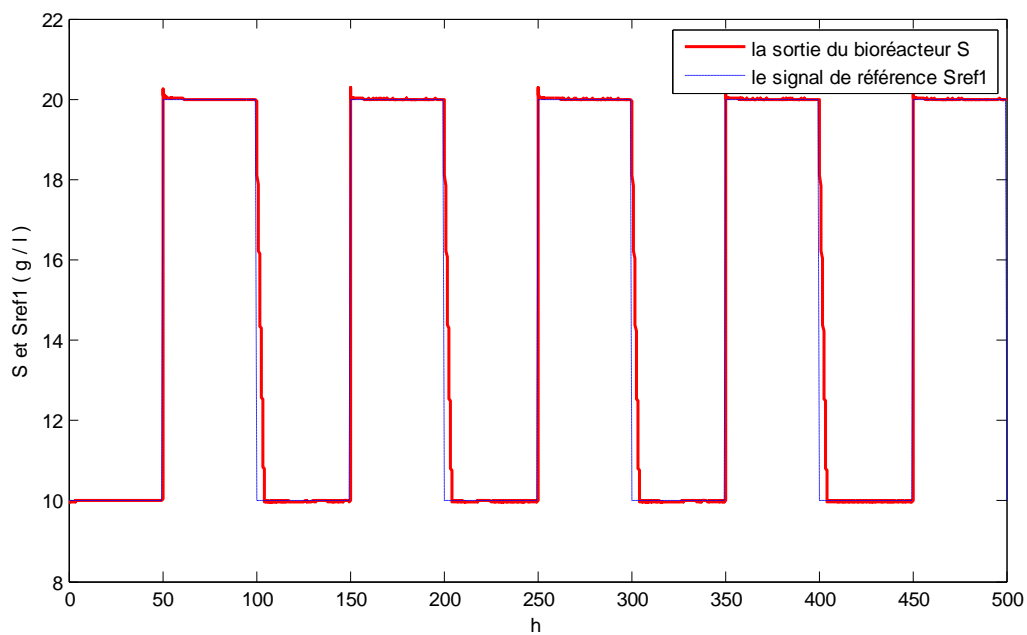


(c)

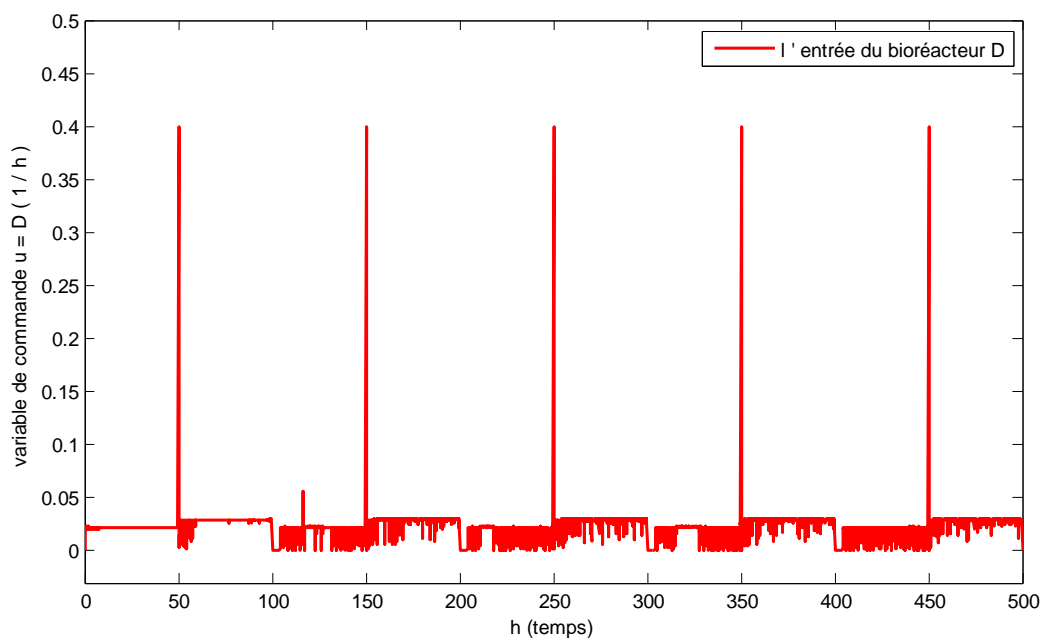
Figure-IV-11: (a) Evolutions du substrat  $S(k)$ , de la consigne  $S_{ref2}(k)$ , (b) signal de commande  $u(k)$ , (c) l'erreur entre  $S(k)$  et  $S_{ref2}(k)$

➤ **La commande neuronale adaptative indirecte**

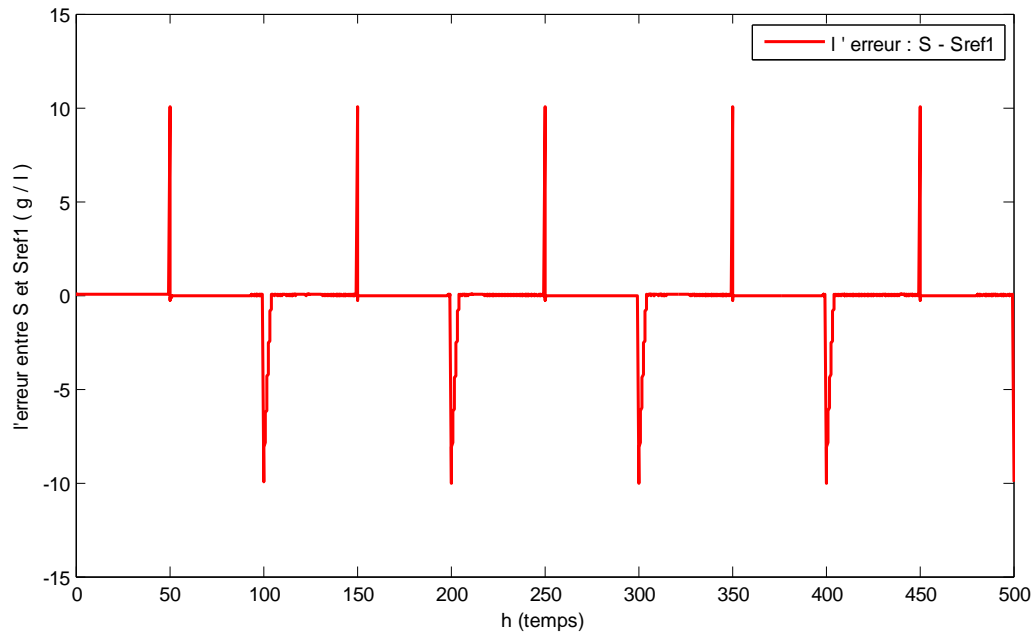
La Figure -IV-12 donne les évolutions de la sortie réelle du bioréacteur, de la consigne et du signal de commande.



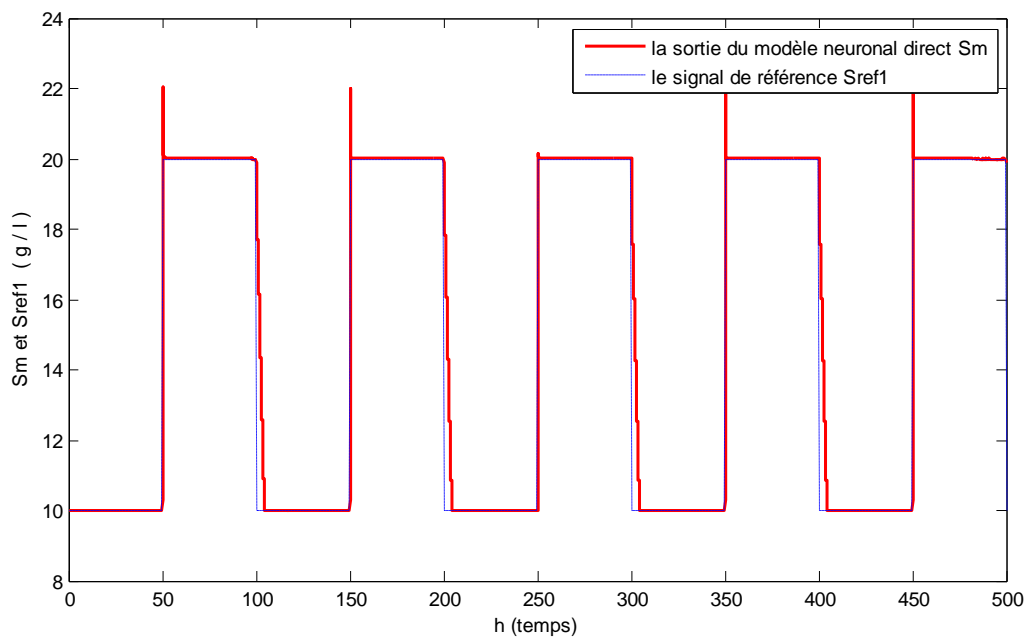
(a)



(b)

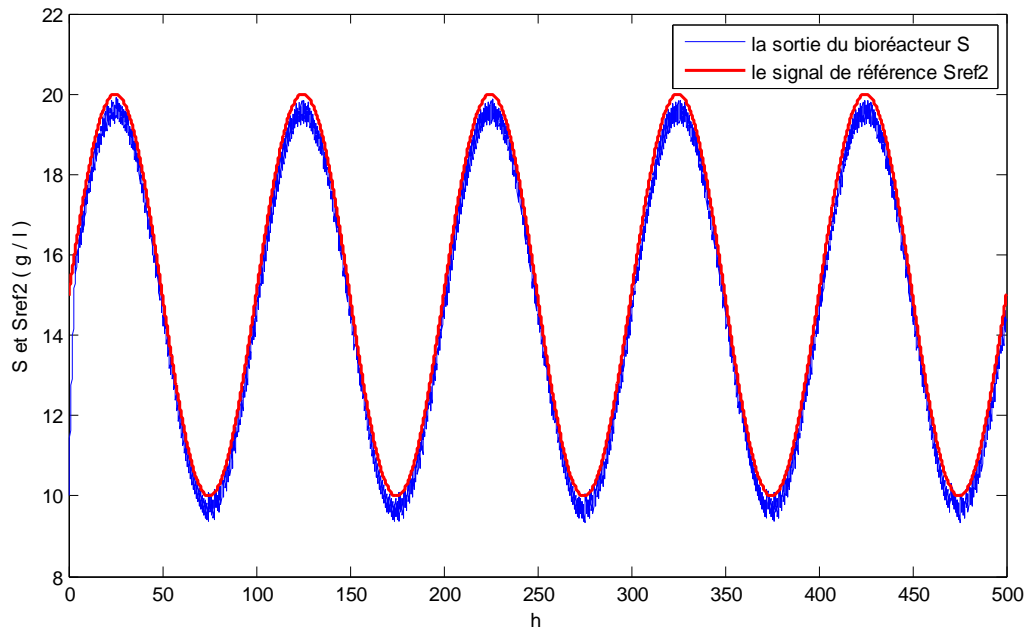


(c)

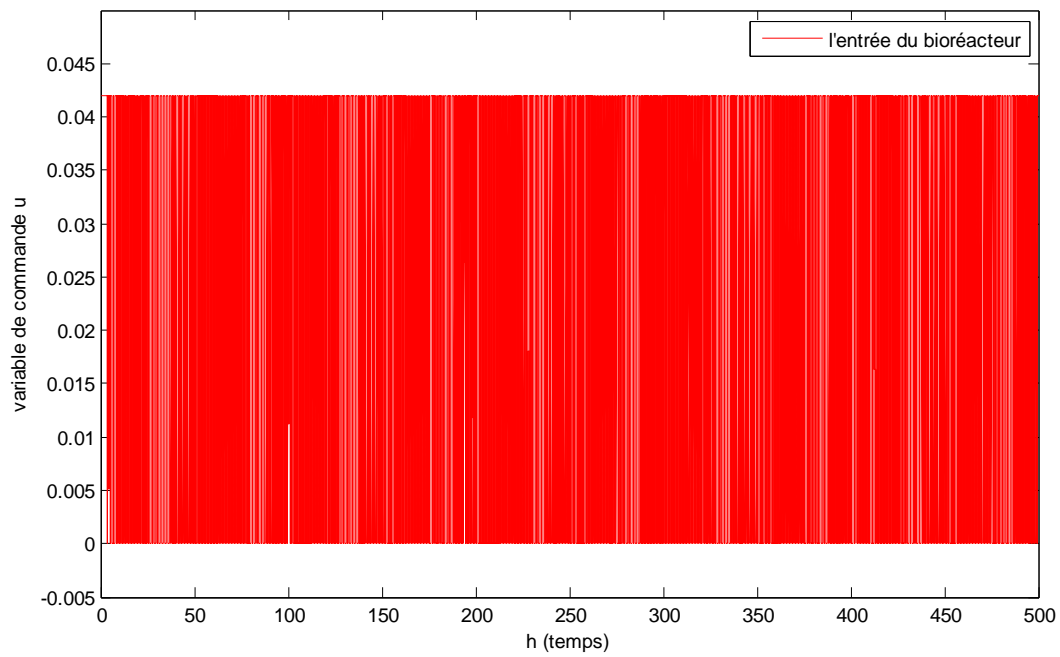


(d)

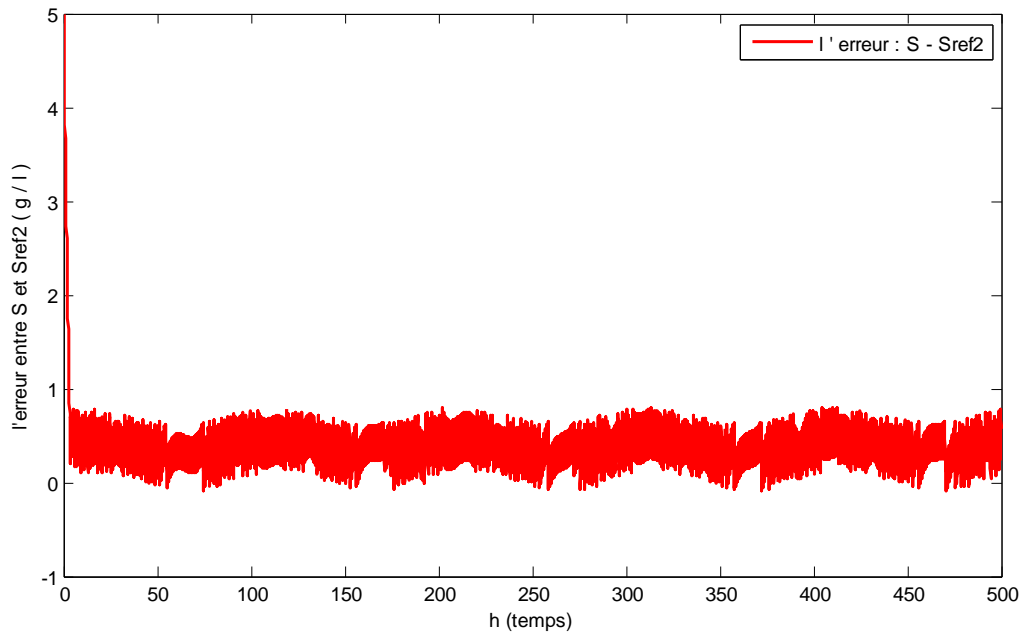
Figure–IV–12: (a) Evolutions du substrat  $S(k)$ , de la consigne  $S_{ref1}(k)$ , (b) signal de commande  $u(k)$ , (c) l'erreur entre  $S(k)$  et  $S_{ref1}(k)$ , (d) la sortie du modèle neuronal  $S_m(k)$  et  $S_{ref1}(k)$



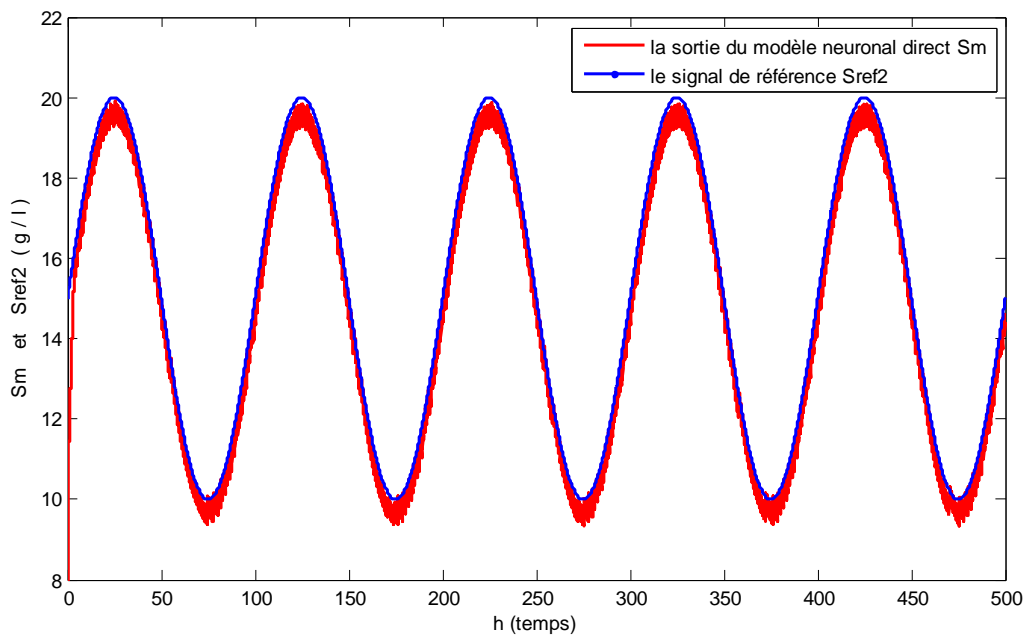
(a)



(b)



(c)

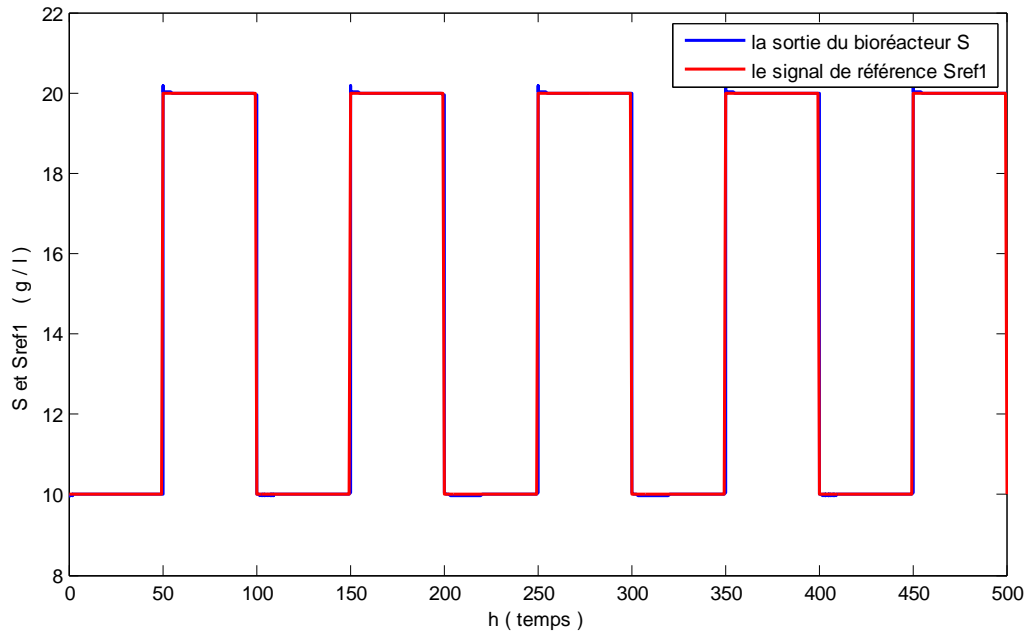


(d)

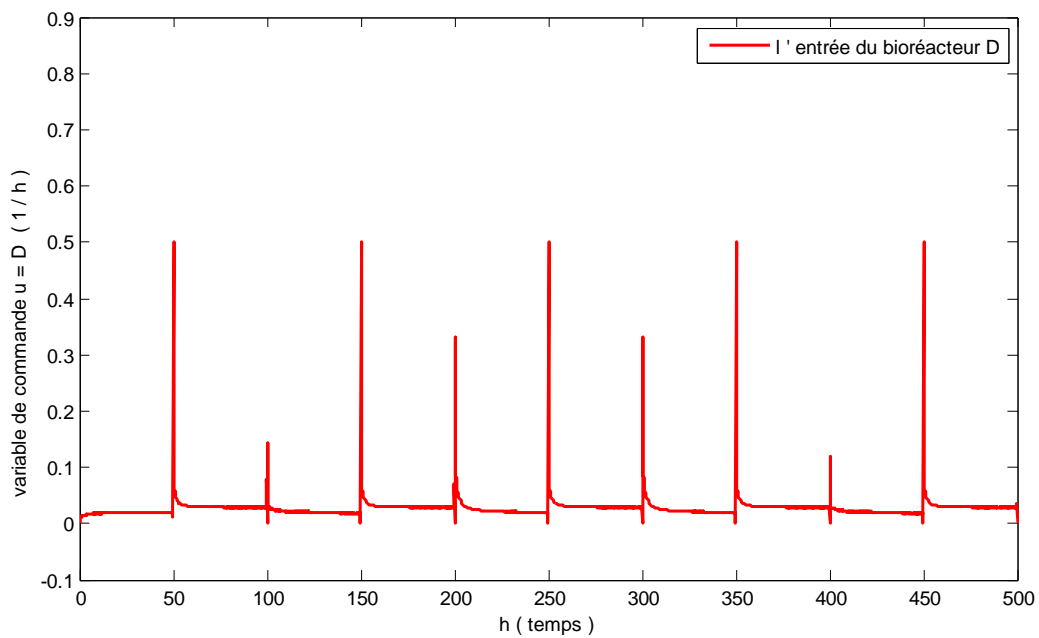
**Figure-IV-13:** (a) Evolutions de la sortie  $S(k)$ , de la consigne  $S_{ref2}(k)$ , (b) signal de commande  $u(k)$ , (c) et l'erreur entre  $S(k)$  et  $S_{ref1}(k)$ , (d) la sortie du modèle neuronal  $S_m(k)$  et  $S_{ref2}(k)$

➤ La commande hybride

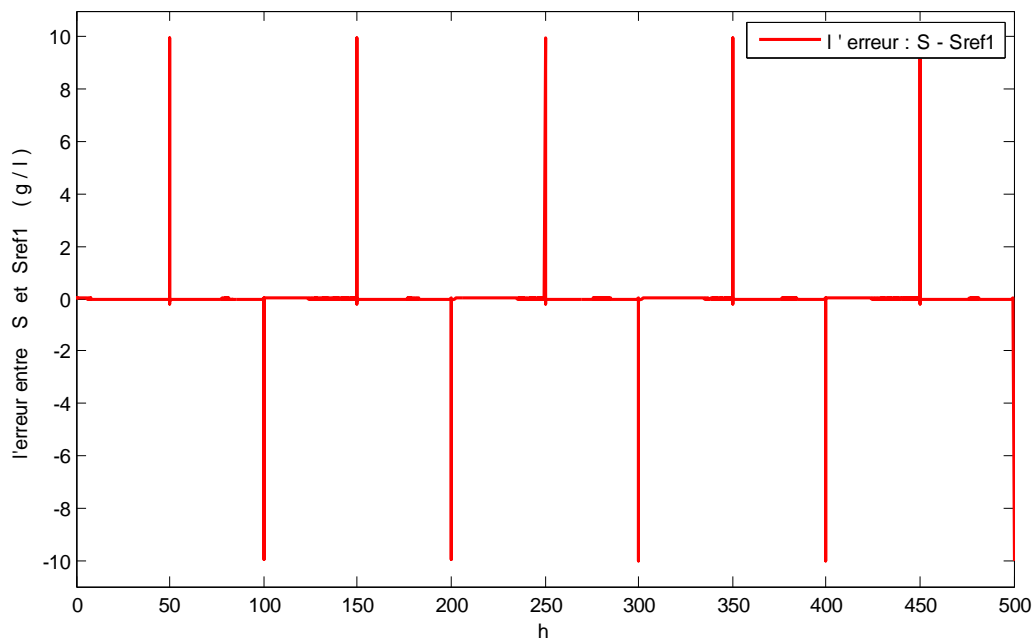
Quant à la commande hybride les résultats obtenus sont donnés à la Figure -IV-15.



(a)

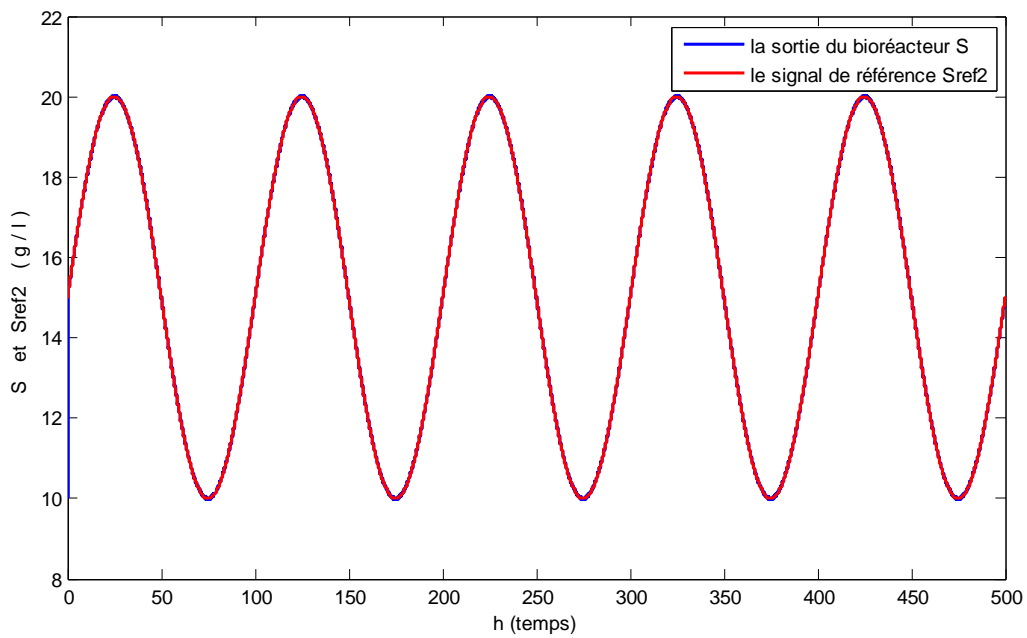


(b)

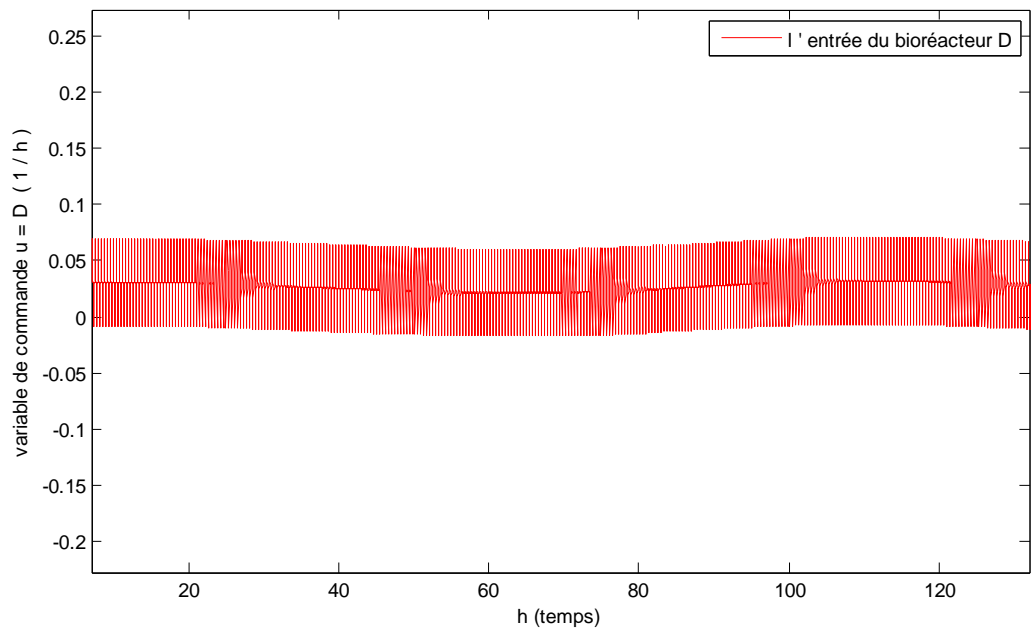


(c)

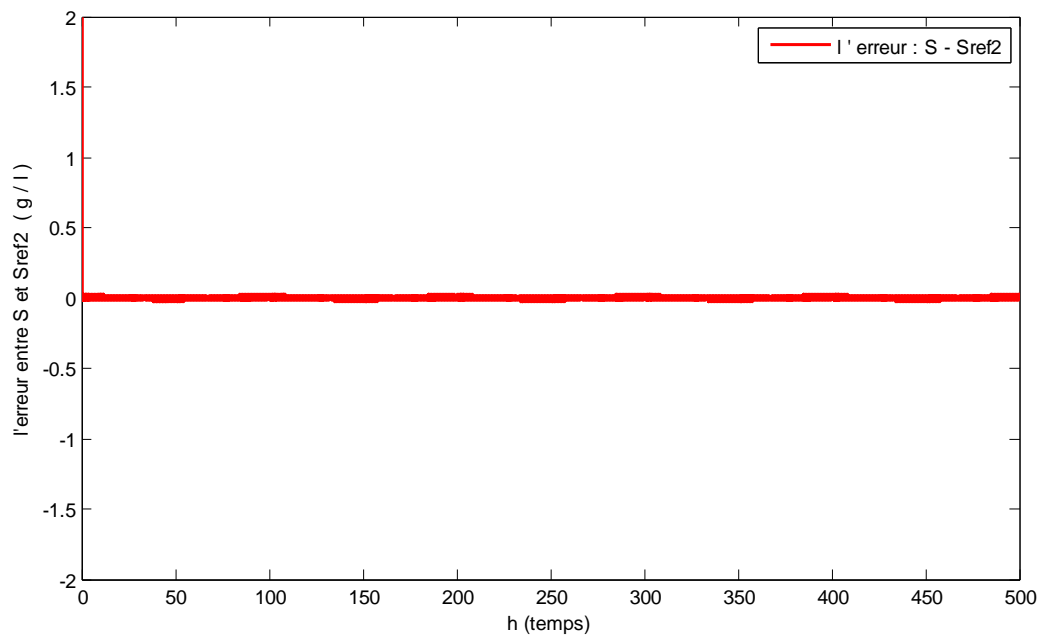
Figure-IV-14: (a) Evolutions de la sortie  $S(k)$ , de la consigne  $S_{ref1}(k)$ , (b) signal de commande  $u(k)$ , (c) l'erreur entre  $S(k)$  et  $S_{ref1}(k)$



(a)



(b)



(c)

Figure-IV-15: (a) Evolutions de la sortie  $S(k)$ , de la consigne  $S_{ref2}(k)$ , (b) signal de commande  $u(k)$ , (c) l'erreur entre  $S(k)$  et  $S_{ref2}(k)$

Ce tableau résume les résultats obtenus suivant le critère  $E_C$  :



$$E_C = \frac{1}{N} \sum_{k=1}^N (S_{ref} - S(k))^2$$

Méthode	L'erreur $E_C$ (Référence carré)	L'erreur $E_C$ (Référence sinusoïdale)
<b>PI</b>	4.9314	1.7370
<b>directe</b>	0.3944	0.2473
<b>indirecte</b>	0.8247	0.1417
<b>hybride</b>	0.1617	0.0048

**Table-IV-2: Comparaison entre les différentes structures de commandes**

### Conclusion

Le travail effectué et présenté dans ce dernier chapitre a pour objectif d'appliquer la technique de la commande neuronale adaptative sur un système non linéaire, comme le bioréacteur biologique.

Cette commande permet une adaptation directe des poids du réseau de neurones. Même si la règle d'adaptation des poids est relativement simple à implémenter, la détermination du nombre de couches et de neurones dans chaque couche cachée dépend de la nature et de la complexité du système à traiter, le réglage des paramètres reste toujours très délicat.

Après plusieurs tentatives, la meilleure architecture obtenue est celle d'une couche cachée de 8 neurones, utilisant l'algorithme de rétro propagation du gradient.

Les structures de commande neuronales adaptatives (directe, indirecte et la commande hybride) ont nettement donné des meilleures performances par rapport aux PI classique (**Table-IV-2**).

Par comparaison des trois structures de commande, on voit bien que la commande hybride présente de meilleurs résultats.

# Coclusion générale

### Conclusion générale

Les propriétés d'apprentissage des réseaux de neurones, nous ont conduits à étudier les techniques neuronales dans les problèmes d'identification, et de la commande des systèmes dynamiques non linéaires. Grâce à ces propriétés, les réseaux de neurones sont devenus des outils de modélisation puissants dont les domaines d'applications sont multiples, nous nous sommes intéressés au bioréacteur biologique.

A fin de commander un système par la technique neuronale, un modèle inverse du système à commander est entraîné hors ligne. Par la suite ce modèle inverse est utilisé comme contrôleur neuronal. Cependant, cette technique présente un inconvénient majeur, car l'identification du modèle neuronal est construite hors ligne avec des paramètres fixes, ce qui produit un modèle neuronal entraîné seulement pour le point de fonctionnement considéré. Il ne peut rien faire face au changement du comportement du système ou à des perturbations externes. Les paramètres doivent donc être modifiés et adaptés en ligne. Pour remédier à ce problème, la commande neuronale adaptative a été utilisée.

D'une manière générale, les contrôleurs neuronaux adaptatifs sont divisés en deux types : la commande neuronale adaptative directe et la commande neuronale adaptative indirecte.

Dans la commande neuronale adaptative directe, le système neuronal est utilisé comme un contrôleur adaptatif.

Cependant ce type de commande présente un inconvénient lorsque le système est inconnu ou mal défini. Nous faisons appel alors à la commande neuronale adaptative indirecte, car ce type de commande utilise deux réseaux de neurones : le premier pour identification, et le deuxième pour la commande.

## CONCLUSION GENERALE

---

Les bons résultats obtenus par simulation ont donné pleine satisfaction et ont montré l'avantage d'utiliser ces techniques dans la commande des systèmes non linéaires.

Une perspective à ce travail sera entamée par la suite. Nous envisageons d'étudier le PID adaptatif basé sur les réseaux de neurones avec un estimateur par la logique floue.

# Bibliographie

## BIBLIOGRAPHIE

---

- [1] M. Boumahraz, "Identification et contrôle avec réseaux de neurones", Thèse de Magister, Université de Sétif 1993.
- [2] M.Melakhessoi "Contrôle et identification des systèmes non linéaires par les techniques neuronales" mémoire Magister, Université de Batna.
- [3] M. el Kebir "Modélisation et commande par modèle inverse neuronale d'un four ventilé", Thèse de Magister, Université d'Oran 2009.
- [4] Dimitrios FRAGKOULI "Détection et localisation des défauts provenant des actionneurs et des capteurs : application sur un système non linéaire" Thèse de doctorat de l'université de Toulouse 2008..
- [5] Gaël Peiffier "Impact environnementale des effluents d'abattoirs : actualité technique et réglementaires" Thèse de doctorat Université de Paul-Sabatier de Toulouse en 2002.
- [6] M. Guitani Issam "Commande adaptative neuronale par retour de sortie des systèmes non linéaires", Thèse de Magister, université Mentouri de Constantine en 2007.
- [7] M. Abdelaziz Aouiche "rejection des perturbation dans les systèmes non linéaires", Thèse de Magister, Université de M'SILA 2006.
- [8] Mohamed Yessin Ammar "Mise en œuvre Mise en œuvre de réseaux de neurones pour la modélisation de cinétique réactionnelle en Batch/continu" Thèse de Doctorat de l'Université de Toulouse 2007.
- [9] Karima Rabah "Modélisation et Contrôle des systèmes à comportement chaotique : Cas des arythmies cardiaques" Thèse de Magister, Université de Constantine 2006.
- [10] U. Jeppsson .A General Description of the Activated Sludge Model No. 1 (ASM1) "Modelling aspects of wastewater treatment processes". These PhD Lund Institute of Technology, ISBN 91-88934-00-4. (1996).
- [11] "Modélisation et commande floues de type Takagi-Sugeno appliquées à un bioprocédé de traitement des eaux usées" Thèse de Doctorat de l'Université Paul Sabatier - Toulouse III et l'Université de los Andes, Colombie 2007.
- [12] Cindy Bassompierre "Procédé à boues activée pour le traitement LE d'effluents papetiers : de la conception d'un pilote à la validation de modèles" Thèse de Doctorat de l'Université de Grenoble en 2007.
- [13] R. Pörtner and C. Giese An Overview on Bioreactor Design, "Prototyping and Process Control for Reproducible Three-Dimensional Tissue Culture". WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim. (2007)

## BIBLIOGRAPHIE

---

[14] John Wiley & Sons, 'Chemical Reactor Design and control'. Université de New Jersey en 2007

[15] MELAKHESSOU LAKHDAR " Contrôle et identification des systèmes non linéaires par les techniques neuronales", L'Habilitation à Diriger Des Recherches université de Haute Alsace en 2009.

[16] Mohamed Barkaoui " Etude et développement d'une commande neuronale adaptative " (M. Sc.) UNTVERSITÉ LAVAL en 1998.

[17] Djaffar ould Abdeslam, "Logique floue et réseaux de neurones pour la commande dans les systèmes électriques : Filtrage actif parallèle", Université de Haute Alsace, Cours école Doctorale SBA 2008/2009.

[18] Hicham Chaoui " Implantation sur FPGA d'une loi de commande adaptative neuronale supervisé pour une articulation flexible", (M.Sc.) université du Québec en Outaouais en 2005.

[19] Tsurng jehng Shen, " Les réseaux de neurones affines et leur application à la commande automatique du vol", thèse de doctorat de l'université de Toulouse en 1995.

[20] MOHAMED CHEMACHEMA, " Contribution à la Commande Adaptative Neuronale Floue des Systèmes Non Linéaires", Thèse de Doctorat de l'Université de Constantine en 2007.

[21] Patrice WIRA, " Approches neuromimétiques pour l'identification et la commande", L'Habilitation à Diriger Des Recherches, université de Haute Alsace en 2009.

[22] Paula Porras, "Protocole de fractionnement de la DCO des eaux résiduaires urbaines et impact sur les résultats de simulation d'une station à boues activées par le modèle ASM1", mémoire de DEA, université de Louis Pasteur Strasbourg en 2003.

[23] Johanna Mouchard, 'réalisation d'un modèle biologique décrivant le fonctionnement dynamique des bioréacteurs à membrane immergée" Mémoire de CES / Mastères Ecole Nationale du Génie de l'Eau et de l'Environnement de Strasbourg en 2005-2006

[24] Geneviève PRADES, "Optimisation d'une installation de traitement par boues activées lors de surcharges de pollution en période de vendanges", école nationale du génie de l'eau et de l'environnement de Strasbourg en 2003.

[25] Fourati, Baklouti, Moalla et Cthourou, " Control of a bioreactor for wastewater treatment ", International Journal of Research and Reviews in Soft and Intelligent Computing (IJRRSIC) Vol. 1, No. 2, June 2011

[26] J. Harmand "Contribution à l'analyse et au contrôle des systèmes biologiques application aux bioprocédés de dépollution". Habilitation à diriger des recherches, Université de C. Bernard (Lyon). 2004

## BIBLIOGRAPHIE

---

- [27] G.P. Reddy, V. R. Kumar and B. Spandana “ Pseudo dynamic model reference nonlinear control of a continuous bioreactor with input multiplicities”. IMECS 2009, March 18-20, Hong Kong. (2009)
- [28] Pham, D. T. and X. Liu (1995) Neural Networks for Identification, Prediction and Control. Springer-Verlag, London.
- [29] K. J. Hunt, D. Sbarbaro; R. Zbikowski and P. J. Gawthrop (1992) Neural networks for control systems – a survey. Automatica, vol. 28, pp. 1083-1112.
- [30] Abderrazak Chatti, Tarek Gallah, Imen Ayari, “Modélisation et commande neuronales d’une unité de régulation de niveau d’eau”, Lara, Ecole Nationale d’Ingénieurs de Tunis, BP. 37, Le Belvédère ,1002 Tunis, Tunisie. LAGIS, Ecole Centrale de Lille, BP. 48, F 59651 Villeneuve d’Ascq Cedex, France.
- [31] D. Psaltis, A. Sideris and A. A. Yamamura (1988) A multilayer neural network controller. IEEE Control Systems Magazine, vol. 8, pp. 17-21
- [32] I.Rivals, (1995), «Modélisation et commande de processus par réseaux de neurones; application au pilotage d’un véhicule autonome», Thèse de Doctorat, Université de Paris 6.
- [33] W. Li and J.J.E. Slotine ,”Neural network control of unknown nonlinear systems”. American Control Conference, Pittsburgh, vol.2, pp. 1136-1141.
- [34] J. B. D. Cabera and K. S. Narendra (1999) “Issues in the application of neural networks for tracking based on inverse control”. IEEE Transactions on Automatic Control, vol. 44, pp. 2007-2027.
- [34]. I. Abdelmalek, "Identification et commande floues des systèmes non linéaires", Thèse de Magister, Université de Batna 1999.
- [36] G. Karsai, “learning to control, some practical experiments with neural networks”, In International Joint Conference on Neural Networks, vol. II, Seattle, 1991, pp. 701-707.
- [37] B. Widrow, “Adaptive inverse control”, In Proceedings of the IFAC Adaptive Systems in Control and Signal Processing Conference, Lund, 1988, pp.1-5.
- [38] D. Psaltis, A.Sideris and A. Yamamura, “Neural controllers”, In International Neural Networks Conference, vol.IV, 1987, pp.551-558.



## BIBLIOGRAPHIE

---

- [39] T. Yabuta and T. Yamada, "Possibility of neural networks controller for robot manipulators, Robotics and Automation", Proceedings, IEEE International Conference, vol.3, 1990, pp. 1686-1691.
- [40] C. Goutte and C. Ledoux, "Synthèse des techniques de commande connexionniste", IBP-Laforia 1995/02: Rapport de Recherche Laforia, 1995.
- [41] G.A. Pugh, "Ship directional control by synthetic neural network", Proc of the American Control conference ACC90, San Diego, 1990, pp.3028-3029.
- [42] Y. Ichikawa and T.Sawa, "neural network application for direct feedforward controller", IEEE Trans on Neural Networks, vol.3 No2, 1992, pp.224-231.
- [43] G.W. Ng, "Application of neural networks to adaptive control of nonlinear systems", Research Studies Press, Taunton, Somerset, England, 1997.
- [44] 1. Zhao, "System modeling, identification and control using fuzzy logic", Thèse de Doctorat en sciences appliquées, UCL, CESAME, Belgique 1995.
- [45] L.X Wang, "Adaptive fuzzy systems and control: Design and stability analysis", Englewood Cliffs, NJ: Prentice Hall, 1994.
- [46] K.J. Åström and B. Wittenmark, "Adaptive control", Addison-Wesley, 1989.
- [47] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," IEEE Trans. Neural Networks, vol. 1, no. 1, pp. 4-27, Mar. 1990. C. W. Anderson.
- [48] K.J.Hunt, D. Sbarbaro, R. Zbikowski and P. J. Gawthrop, "Neural network for control systems - A survey," Automatica, vol. 28, no. 6, pp 1083-1112, Dec 1992.
- [49] G.Dreyfus, "Réseaux de neurones, méthodologie et application", 2ème édition Eyrolles, 416p, 2004.
- [50] K S. Narendra and K Parthasarathy, "Identification and control of dynamical systems using neural networks", IEEE Trans. On Neural Networks., vol. 1, no. 1, pp.4-26, Mar 1990.
- [51] D.Arar, A.Ghadhbane et A.Bensaadi, "Identification et contrôle des systèmes non linéaires
- [52] N. Golea, "Commande par logique floue " Thèse de Magister, Institut. D'Electronique, Université de Sétif, 1994.
- [53] KS. Narendra, "Neural networks for identification and control", Center for Systems Science, Yale University, Dec. 1998.
- [54] P. Ioannou, J. Sun, "Robust Adaptive Control", Prentice-Hall, Inc., Englewood Cliffs, NJ, 1996

## Résumé

Dans ce mémoire, on utilise la commande adaptative par structure neuronale appliquée à un système non linéaire du bioréacteur pour le traitement des eaux. Nous comparerons les résultats des commandes adaptatives neuronales directe et indirecte utilisées avec celles de la commande classique de type PI.

Les résultats sont validés par des simulations sous l'environnement MATLAB.

**Mots clés** : systèmes non linéaires, bioréacteur, réseaux de neurones, La commande adaptative neuronale.

## Abstract

In this specification, the adaptive control by using the neural network structure applied to a nonlinear system of the bioreactor for the wastewater treatment. We will compare the results of the adaptive neural control used with those of conventional PI control.

The results are validated by simulation in MATLAB environment.

**Keywords:** nonlinear system, bioreactor, neural network, adaptive neural control.

## ملخص

في هذه المواصفات، والسيطرة على التكيف باستخدام بنية الشبكة العصبية تطبيقها على نظام غير الخطية من مفاعل حيوي لمعالجة مياه الصرف الصحي. التقليدية PI سنقارن نتائج التحكم العصبي على التكيف مع تلك المستخدمة من السيطرة

يتم التحقق من صحة النتائج عن طريق المحاكاة في بيئ

MATLAB

نظام غير الخطية، مفاعل حيوي، الشبكة العصبية، السيطرة العصبية التكيفية:الكلمات الرئيسية