

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE FERHAT ABBAS SETIF-1

FACULTE DE TECHNOLOGIE

THESE

Présentée au Département d'électronique

Pour l'obtention du diplôme de

DOCTORAT EN SCIENCES

Filière : électronique

Option : électronique

Par

LEMITA Abdallah

THEME

**Méthodologie d'optimisation de la commande du procédé de
traitement des eaux usées par boues activées basée sur les algorithmes
évolutionnaires**

Soutenue le 17/11/2021 devant le Jury:

CHEMALI Hamimi	Professeur	Univ. F. Abbas Sétif 1	Présidente
BOULAHBEL Sebti	MCA	Univ. F. Abbas Sétif 1	Directeur de thèse
BOUTOUT Farid	Professeur	Univ. B.B.A	Examineur
AOUICHE A/Aziz	MCA	Univ. Tebessa	Examineur
KHELLAF Abdelhafid	Professeur	Univ. F. Abbas Sétif 1	Invité

عملية معالجة مياه الصرف الصحي الحمأة المنشطة هي نظام غير خطي معقد للغاية. يُظهر عدم خطية كبيرة واضطرابات وتغيرات غير متوقعة في المدخلات. تستخدم عملية معالجة مياه الصرف الصحي عدة مراحل من المعالجة ، ولكن أهمها المعالجة البيولوجية. يتمثل مبدؤها في استخدام كائنات دقيقة معينة (بكتيريا) قادرة على التطور عن طريق استهلاك المواد الملوثة. إن تطور هذه الكائنات الحية الدقيقة مشروط بوجود الأكسجين المذاب في الماء ، وبالتالي فإن التحكم في الأكسجين المذاب ضروري في معالجة مياه الصرف الصحي بواسطة الحمأة المنشطة. سيتم استخدام طريقتين للتحكم في تركيز الأكسجين المذاب في مفاعل حيوي لمعالجة مياه الصرف الصحي الحمأة المنشطة: تقنية التحكم الأولى هي مزيج بين منظم PI والشبكات العصبية الاصطناعية (تحكم PI التكيفي للبنية العصبية). تستخدم هذه التقنية منظم PI للتحكم في تركيز الأكسجين المذاب ، بينما تستخدم الشبكات العصبية لضبط معاملات المنظم PI k_p و k_i . تقنية التحكم الثانية هي مزيج من طريقة النسب المتدرجة والشبكات العصبية. سيتم مقارنة نتائج المحاكاة لتقنيتي التحكم مع منظم PI التقليدي لإظهار أدائهم.

الكلمات الدالة:

معالجة مياه الصرف الصحي باستخدام الحمأة المنشطة ، والشبكات العصبية الاصطناعية ، وطريقة الانحدار ، ومنظم PI.

Résumé:

Le procédé du traitement des eaux usées par boues activées est un système non linéaire très complexe. Il présente un grand non linéarité, des perturbations et des changements inattendus à l'entrée. Le procédé du traitement des eaux usées utilise plusieurs étapes du traitement, mais le traitement le plus important, c'est le traitement biologique. Son principe consiste à utiliser certains micro-organismes (bactéries) capables de se développer en nourrissant des matières polluantes. Le développement de ces micro-organismes est conditionné par la présence de l'oxygène dissous dans l'eau, donc, le contrôle de l'oxygène dissous est indispensable dans le traitement des eaux usées par boues activées. Deux techniques seront utilisées pour contrôler la concentration de l'oxygène dissous dans un bioréacteur de traitement des eaux usées par boues activées: la première technique de commande est une combinaison entre un régulateur PI et les réseaux de neurones artificiels (commande PI adaptative d'une structure neuronale). Cette technique utilise le régulateur PI pour commander la concentration d'oxygène dissous, tandis que les réseaux de neurones sont utilisés afin d'ajuster les paramètres du régulateur PI k_p et k_i . La deuxième technique de commande est une combinaison entre la méthode de descente du gradient et les réseaux de neurones. Les résultats de simulation des deux techniques de commande seront comparés avec un régulateur PI classique pour montrer leur performance.

Mots clés:

Traitement des eaux usées par boues activées, les réseaux de neurones artificiels, la méthode de descente du gradient, le régulateur PI.

Abstract:

The activated sludge wastewater treatment process is a very complex non-linear system. It presents a strong non-linearity, disturbances and unexpected changes at the input. The wastewater treatment process uses various steps of treatment, but the most important one is the biological treatment. Its principle consists by using certain microorganisms (bacteria) able of developing by feeding on polluting matter. The development of these microorganisms is very affected by the presence of dissolved oxygen in the water, therefore, the control of dissolved oxygen is essential in the treatment of wastewater by activated sludge. Two techniques will be used to control the dissolved oxygen concentration in an activated sludge wastewater treatment bioreactor: the first control technique is a combination between a PI regulator and artificial neural networks (adaptive PI control with a neural structure). This technique uses the PI regulator to control the dissolved oxygen concentration, while neural networks are used to adjust the PI parameters k_p and k_i . The second control technique is a combination of the gradient descent method and neural networks. The simulation results of the two control techniques will be compared with a conventional PI regulator to show their performance.

Keywords:

Wastewater treatment by activated sludge, artificial neural networks, the gradient descent method, the PI regulator.

Remerciements

Aux noms d'Allah, le Tout Miséricordieux, le Très Miséricordieux. Tout à d'abord, on remercie Allah le Tout Puissant Qui nous accordé la réussite d'accomplir ce travail, car sans Son aide on ne peut rien faire.

Je tiens à remercier Monsieur **BOULAHBEL Sebti** Maître de conférences à l'Université de Ferhat Abbas Sétif 1 d'avoir diriger ce travail ainsi que pour ses conseils et ses remarques constructifs.

Je tiens à remercier Monsieur **CHEMALI Hamimi** Professeur à l'Université de Ferhat Abbas Sétif 1, d'avoir accepter de présider le Jury de thèse.

Je suis très reconnaissant aux membres de Jury: Monsieur **BOUTOUT Farid** Professeur à l'Université Bordj Bou Arreridj, Monsieur **AOUICHE Abdelaziz** Maître de conférences à l'Université de Tébessa et le Monsieur **KHELLAF Abdelhafid** Professeur à l'Université de Ferhat Abbas Sétif 1, qui m'ont fait l'honneur d'avoir accepter d'examiner ce travail.

Je tiens particulièrement à remercier Monsieur **KAHLA Sami**, Maître de recherche dans le centre de recherche en technologie industrielle CRTI à Cheraga, qui m'a énormément aidé.

Je tiens également à remercier à toutes personnes qui m'ont aidés de près ou de loin.

Table des matières:

Table des Matières.....	iv
Liste des figures.....	viii
Liste des tableaux.....	xii
Liste des symboles et acronymes.....	xiii
Introduction générale	
Introduction générale	1
Chapitre I: Généralités sur le traitement des eaux usées	
I-1) Introduction	4
I-2) Définitions	5
I-2-1) Matières en suspension (MES).....	5
I-2-2) Matières colloïdales	5
I-2-3) Matières solubles	5
I-2-4) Microorganismes	5
I-2-5) La demande chimique en oxygène (DCO).....	6
I-2-6) La demande biologique en oxygène (DBO5).....	6
I-3) Définition des eaux usées	6
I-4) Les différents types des eaux usées	6
I-4-1) Les eaux usées domestiques	6
I-4-2) Les eaux usées industrielles	7
I-4-3) Les eaux usées agricoles	7
I-4-4) Les eaux usées pluviales	7

I-5) Définition d'une station d'épuration	7
I-6) Les grandes étapes du traitement des eaux usées	8
I-6-1) Le prétraitement	9
I-6-2) Le traitement primaire	10
I-6-3) Le traitement secondaire ou biologique	10
I-7) Procédé de traitement des eaux usées par boues activées	11
I-8) Conclusion	12
Chapitre II: La résolution numérique des équations différentielles	
II-1) Introduction	14
II-2) Les équations différentielles ordinaires (EDO).....	14
II-3) La résolution numérique des équations différentielles	15
II-4) La méthode d'Euler	15
II-4-1) La méthode d'Euler explicite	15
II-4-2) La méthode d'Euler implicite.....	18
II-4-3) La méthode d'Euler modifiée	18
II-5) Conclusion	22
Chapitre III: La méthode de descente du gradient	
III-1) Introduction	23
III-2) Méthode de descente du gradient	23
III-2-1) Méthode du gradient à pas fixe	24
III-2-2) Méthode du gradient adaptatif (AdaGrad).....	28
III-2-3) Méthode de gradient avec momentum	31
III-3) Conclusion	34

Chapitre IV: Les réseaux de neurones

IV-1) Introduction	36
IV-2) Neurone biologique	37
IV-3) Neurone formel	37
IV-4) Quelque exemple des fonctions d'activations	39
IV-4-1) La fonction d'activation binaire	39
IV-4-2) La fonction d'activation signe	39
IV-4-3) La fonction d'activation linéaire	40
IV-4-4) La fonction d'activation linéaire avec saturation	41
IV-4-5) La fonction d'activation sigmoïde	42
IV-4-6) La fonction d'activation tangente hyperbolique	43
IV-5) Structure des réseaux de neurones artificiels MLP (perceptron multicouche).	45
IV-6) Apprentissage	48
IV-7) Algorithme de rétro-propagation	49
IV-8) Le réseau de neurones RBF	56
IV-9) Conclusion	58

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

V-1) Introduction	59
V-2) Régulateur PI	59
V-3) La commande PI adaptative basée sur la technique des réseaux de neurones RBF	60
V-3-1) Approximation de l'expression par la technique des réseaux de neurones RBF	61

Table des matières

V-4) Principe de fonctionnement de la commande de gradient basée sur la technique des réseaux de neurones	63
V-4-1) La méthode du gradient	65
V-5) Modèle mathématique d'un bioréacteur de traitement des eaux usées par boues activées	68
V-6) Résultats de simulation	69
V-7) Conclusion	88
Conclusion générale	
Conclusion générale	89

Liste des figures:

Chapitre I: Généralités sur le traitement des eaux usées

Figure I.1: Les étapes du traitement de la station d'épuration	8
Figure I.2: Prétraitement	9
Figure I.3: Traitement primaire	10
Figure I.4: Traitement secondaire ou biologique	11
Figure I.5: Schéma général d'une station à boues activées	11

Chapitre II: La résolution numérique des équations différentielles

Figure II.1: Principe de la méthode d'Euler	16
Figure II.2: Méthode d'Euler explicite: schéma du rectangle à gauche	17
Figure II.3: Méthode d'Euler implicite: schéma du rectangle à droite	18
Figure II.4: Méthode d'Euler modifiée: schéma du rectangle à point milieu	19
Figure II.5: Méthode d'Euler explicite et modifiée	21

Chapitre III: La méthode de descente du gradient

Figure III.1: Principe de la méthode de la descente du gradient	24
Figure III.2: Méthode du gradient: $\rho = 0.05$	26
Figure III.3: Méthode du gradient: $\rho = 0.2$	27
Figure III.4: Méthode du gradient: $\rho = 1$	27
Figure III.5: Méthode du gradient à pas fixe: $\rho = 0.05$	29
Figure III.6: Méthode du gradient adaptatif	31
Figure III.7: Méthode du gradient sans momentum	32
Figure III.8: Méthode du gradient avec momentum.....	33

Liste des figures

Figure III.9: Méthode du gradient sans et avec momentum	34
Chapitre IV: Les réseaux de neurones	
Figure IV.1: Neurone biologique	37
Figure IV.2: Structure d'un neurone formel	38
Figure IV.3: La fonction binaire	39
Figure IV.4: La fonction d'activation sign	40
Figure IV.5: La fonction d'activation linéaire	41
Figure IV.6: La fonction d'activation linéaire avec saturation	42
Figure IV.7: La fonction d'activation sigmoïde	43
Figure IV.8: La fonction d'activation tangente hyperbolique	44
Figure IV.9: Architecture du réseau neuronal multicouche	45
Figure IV.10: Structure d'un réseau neurone avec trois neurones dans la couche cachée et un neurone dans la couche de sortie	47
Figure IV.11: Structure d'un réseau neurone avec trois neurones dans la couche cachée et deux neurones dans la couche de sortie.....	53
Figure IV.12: Structure d'un réseau de neurones RBF.....	57
Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones	
Figure V.1: Schéma général d'un régulateur PI.....	59
Figure V.2: Modélisation d'un système par la technique des réseaux de neurones.....	61
Figure V.3: La commande PI adaptative basée sur la technique des réseaux de neurones RBF.....	63
Figure V.4: Système non linéaire avec une seule entrée et une seule sortie.....	63

Liste des figures

Figure V.5: Courbes de l'équation (5.17) avec deux entrées de commande différentes $u(t) = u_0$ et $u(t) = u_1$	64
Figure V.6: Principe de fonctionnement la commande propose.....	65
Figure V.7: Modélisation d'un système par la technique des réseaux de neurones...	66
Figure V.8: Structure de la technique de commande de gradient basée sur les réseaux de neurones RBF.....	68
Figure V.9: Schéma général d'un bioréacteur de traitement des eaux usées par boues actives.....	68
Figure V.10: Concentration de la biomasse.....	70
Figure V.11: Concentration du substrat.....	71
Figure V.12: Concentration de l'oxygène dissous.....	71
Figure V.13: Concentration de la biomasse recycle.....	72
Figure V.14: Structure de la commande proposée.....	73
Figure V.15: Modèle neuronal d'un système non linéaire (bioréacteur).....	73
Figure V.16: La commande PI adaptative basée sur la technique des réseaux de neurones RBF.....	74
Figure V.17: Concentration du substrat S_{in}	75
Figure V.18: Taux de dilution.....	76
Figure V.19: Concentration de l'oxygène dissous.....	76
Figure V.20: L'erreur entre la référence et la concentration de l'oxygène dissous ..	78
Figure V.21: Le taux d'aération (variable de commande).....	79

Liste des figures

Figure V.22: Adaptation des poids $C_{1,j}$	80
Figure V.23: Adaptation des poids $C_{2,j}$	80
Figure V.24: Adaptation des poids b_j	81
Figure V.25: Adaptation des poids $W_{1,j}$	81
Figure V.26: Adaptation des poids $C_{1,j}$	82
Figure V.27: Adaptation des poids $C_{2,j}$	83
Figure V.28: Adaptation des poids b_j	83
Figure V.29: Adaptation des poids $W_{1,j}$	84
Figure V.30: Adaptation du gain k_p	85
Figure V.31: Adaptation du gain k_i	85
Figure V.32: Concentration du substrat.....	86

Liste des tableaux:

Chapitre II: La résolution numérique des équations différentielles

Tableau II.1: Comparaison de la performance..... 21

Chapitre IV: Les réseaux de neurones

Tableau IV.1: Comparaison entre le neurone biologique et formel..... 38

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient
& la technique des réseaux de neurones

Tableau V.1: Comparaison entre les différentes techniques de commande..... 87

Tableau V.2: Valeurs de: temps de montée, temps de réponse et le dépassement...
..... 87

Liste des symboles et acronymes:

Symboles:

MES	Matières En Suspension
h	Pas de discrétisation
ρ	Pas de descente ou le taux d'apprentissage
β	Facteur de momentum
$W_{j,i}$	Poids entre la couche d'entrée et la couche cachée
h_j	Vecteur de sortie de la couche cachée du réseau de neurones
$V_{k,j}$	Poids entre la couche la couche cachée et la couche de sortie
y_k	Vecteur de sortie du réseau de neurones
f	Fonction d'activation de la couche cachée
g	Fonction d'activation de la couche de sortie
y_{d_k}	Sortie désirée du réseau de neurones à l'instant k
y_{nn_k}	Sortie du réseau de neurones à l'instant k
E	Fonction de performance à minimiser
η	Taux d'apprentissage
e_k	Erreur entre la sortie désirée et la sortie du réseau de neurones
$C_{j,i}$	Centre
b_j	Largeur de base

Liste des symboles et acronymes

E_{RBF}	Fonction de performance à minimiser
e_{RBF}	Erreur entre la sortie désirée et la sortie du réseau de neurones RBF
α	Facteur moment
$\Delta C_{j,i}$	Changement de $C_{j,i}$
Δb_j	Changement de b_j
$\Delta V_{k,j}$	Changement de $V_{k,j}$
u	Entrée de commande du système
k_p	Paramètre proportionnel du régulateur PI
k_i	Paramètre intégral du régulateur PI
r	Référence
y	Sortie du système
e	Erreur entre la référence et la sortie du système
Δk_p	Changement de k_p
Δk_i	Changement de k_i
y_{u_k}	Trajectoire de la sortie du système pour une entrée de commande u_k
λ	Taux d'apprentissage
X	Concentration de la biomasse
S	Concentration du substrat

Liste des symboles et acronymes

DO	Concentration de l'oxygène dissous
X_r	Concentration de la biomasse recyclée
μ_{max}	Taux de la croissance maximale de la biomasse
k_S	Coefficient de demi-saturation de la biomasse
k_{DO}	Coefficient de demi-saturation d'oxygène
D	Taux de dilution
r	Taux de recyclage de la biomasse
Y	Facteur de rendement de la biomasse
S_{in}	Concentration du substrat entrante
K_0	Coefficient constant
KLa	Coefficient de transfert de l'oxygène
α	Taux de transfert de l'oxygène
β	Taux d'excès de la biomasse
W	Taux d'aération
DO_{max}	Concentration maximale de l'oxygène
DO_{in}	Concentration de l'oxygène entrante

Acronymes:

MES	Matières En Suspension
DCO	La Demande Chimique en Oxygène
DBO ₅	La Demande Biologique en Oxygène

Liste des symboles et acronymes

EDO	Equation Différentielle Ordinaire
GD	Gradient Descent
AdaGrad	Adaptive Gradient
Adadelta	Adaptive Delta
NAG	Nesterov Accelerated Gradient
RMSprop	Root Mean Square Propagation
Adam	Adaptive moment estimation
Nadam	Nesterov-Accelerated Adaptive moment estimation
MLP	Multi Layer Perceptron
RBF	Radial Base Function
PI	Proportional Integral
IAE	Integral of Absolute Error
ISE	Integral of Square Error

Introduction générale

Introduction générale

L'eau est une ressource précieuse et indispensable à l'existence de toute espèce humaine, végétale ou animale. Sans l'eau la vie serait impossible car tout dépend d'elle et elle est utilisée dans plusieurs domaines: elle permet de se laver, de faire la vaisselle, d'éteindre les feux, d'irriguer les champs et d'arroser les plantes et plus encore.

Cependant, à cause du développement qu'a connu l'être humain ces derniers siècles dans plusieurs domaines, en particulier, dans le domaine industriel et agricole, les eaux sont devenues altérées, car ce développement était accompagné par une utilisation intensive des pesticides et des engrais dans les activités agricoles pour augmenter le produit. Tout cela a conduit à la détérioration de l'air et du sol, et à la pollution de l'eau dans les eaux des mers et des rivières en provoquant un empoisonnement de la vie marine, sans oublier l'effet du réchauffement climatique qui a détérioré l'écosystème et la santé humaine. Donc, le traitement des eaux usées est devenu une obligation.

Le procédé le plus utilisé dans le traitement des eaux usées est celui de boues activées. Son principe consiste à ajouter aux eaux usées des bactéries ou micro-organismes capables de dégrader les matières polluantes. Ces micro-organismes ont besoin de l'oxygène dissous dans cette opération de dégradation. Donc, la présence de l'oxygène dissous dans un procédé de traitement des eaux usées par boues activées est un paramètre très important qui a un effet direct sur la croissance des micro-organismes et l'efficacité du traitement, sinon les micro-organismes ne seront pas capables de dégrader les matières polluantes [1-2]. Donc, le contrôle de l'oxygène dissous dans un procédé de traitement des eaux usées par boues activées est devenu une nécessité.

Parmi les techniques de commande les plus utilisées pour contrôler la concentration de l'oxygène dissous dans un procédé de traitement des eaux usées par boues activées est la commande ou le régulateur PID classique [3-6]. Malheureusement, un régulateur PID classique dont les paramètres qui sont fixes n'est pas toujours capable d'assurer les performances désirées, surtout quand le système à commander ou contrôlé présente des

non linéarités, des incertitudes, des perturbations et des changements inattendus comme c'est le cas avec le procédé de traitement des eaux usées par boues activées [7-8].

Afin d'améliorer les performances du contrôleur dans le procédé de traitement des eaux usées, la technique de commande ou bien le contrôleur utilisé doit être adaptatif. La commande adaptative diffère donc d'un régulateur PI classique du fait que ses paramètres sont variables et ajustés en temps réel en se basant sur l'état du système commandé [9-12].

Dans notre thèse, deux techniques de commande seront utilisées pour contrôler la concentration de l'oxygène dissous dans un bioréacteur de traitement des eaux usées par boues activées:

- * La première est la commande PI adaptative d'une structure neuronale. Elle est basée sur deux techniques: un régulateur PI classique et la technique intelligente des réseaux de neurones RBF. Le régulateur PI est utilisé pour commander la concentration de l'oxygène dissous, tandis que la technique des réseaux de neurones est utilisée pour auto-ajuster en permanence les paramètres du régulateur PI k_p et k_i . Cette combinaison permet d'améliorer les performances et l'adaptabilité du contrôleur PI contre les perturbations.

- * La deuxième est la commande proposée qui utilise la méthode de gradient avec la technique intelligente des réseaux de neurones RBF.

L'ensemble du travail est enchaîné comme suit:

Le premier chapitre sera consacré aux notions générales sur le traitement des eaux usées par boues activées y compris; les stations d'épurations, les différents types des eaux usées ainsi que leur origine, les différentes étapes de traitement des eaux usées; le prétraitement (le dégrillage, dessablage, déshuilage-dégraissage), traitement primaire (décantation, coagulation-floculation, flottation) et traitement secondaire ou biologique qui a pour objectif d'éliminer les substances qui sont échappées au traitement primaire en utilisant des micro-organismes.

Dans le deuxième chapitre et le troisième chapitre, deux méthodes seront expliquées:

Introduction générale

- La méthode d'Euler qui sert à résoudre numériquement les équations différentielles.
- La méthode d'optimisation de descente du gradient.

Ces deux algorithmes ou méthodes seront ensemble utilisées dans la commande proposée celle de la méthode de gradient avec la technique intelligente des réseaux de neurones RBF.

Le quatrième chapitre sera consacré aux notions théoriques des réseaux de neurones artificiels: définition, les fonctions d'activations les plus utilisées, la structure d'un réseau de neurones MLP, et un réseau de neurone RBF, l'algorithme d'apprentissage de rétro-propagation.

Le cinquième chapitre présentera une démonstration des deux techniques de commande utilisées; la commande PI adaptative d'une structure neuronale et la commande proposée celle du gradient avec la technique des réseaux de neurones RBF. Sous un environnement Matlab, ces deux techniques de commande seront utilisées pour commander la concentration de l'oxygène dissous dans un bioréacteur de traitement des eaux usées par boues activées. Nous comparerons les résultats de simulation obtenus par ces deux techniques de commande avec ceux d'un régulateur PI classique pour montrer leur intérêt dans la commande dans les systèmes non linéaires.

Enfin, nous finirons par une conclusion générale.

Chapitre I: Généralités sur le traitement des eaux usées

I-1) Introduction

Les eaux usées sont principalement des eaux qui ont perdu leurs puretés par la présence des polluants issus des différents usages: domestiques, industriels, agricoles ou autres. Cette contamination des eaux les a rendues impropres à l'utilisation par les êtres humains, les animaux et les plantes.

On peut distinguer trois types de pollution: pollution physique, chimique et organique. La pollution physique comprend les matériaux solides à savoir les pierres, les matières plastiques, etc. Sa source physique pourrait être les déchets ménagers. La pollution chimique comprend les matières chimiques et minérales et les métaux lourds. Sa source chimique vient des pesticides, des déchets agricoles et industriels. La pollution organique comprend les huiles industrielles, les matières organiques dissoutes et leur source peut être des déchets ménagers et industriels. Les risques qui en résultent sont la pollution de l'environnement, l'intoxication de l'eau et les maladies microbiennes. Quelque soit la nature des polluants, les eaux usées doivent être traitées dans une station d'épuration des eaux usées avant d'être jetées dans les milieux naturels qui sont généralement les mers, les rivières et les océans.

Le procédé de traitement des eaux usées est composé de plusieurs phases. Chaque phase traite un type particulier de pollution. Les étapes de traitement qui caractérisent chacune des phases sont : le prétraitement, les traitements primaire et secondaire.

* Le prétraitement est un traitement de type physique qui est une phase d'épuration grossière dans laquelle on élimine toutes les matières volumineuses et grossières (pierres, flacons, etc.) qui pourraient endommager les installations et perturber les étapes de traitement ultérieures.

* Le traitement primaire est de type physico-chimique qui permet d'éliminer les matières en suspension décantables en utilisant des réactifs chimiques.

* Le traitement secondaire est de type biologique qui permet d'éliminer les matières dissoutes. Cette phase de traitement biologique représente la phase clé de la chaîne

globale du traitement [13]. Son principe de fonctionnement repose sur le développement des populations bactériennes capables de dégrader les matières solubles biodégradables et les matières en suspension.

Ce chapitre est consacré aux notions générales du traitement des eaux usées. Il est introduit d'abord par une définition des eaux usées puis par la définition de la station d'épuration. Ensuite, on aborde les différents types des eaux usées ainsi que les différentes étapes de traitement (le prétraitement, le traitement primaire et secondaire). Enfin, on terminera par une conclusion.

I-2) Définitions

I-2-1) Matières en suspension (MES)

Les matières en suspension sont des particules solides et colloïdales qui sont visibles à l'œil. Elles peuvent être d'une taille allant de 0.1 à 1 *mm*. Leur élimination peut être effectuée par une opération de filtration ou centrifugation.

I-2-2) Matières colloïdales

Elles sont des particules d'une taille allant de 0.01 à 100 μm . Leur élimination est effectuée par une opération de coagulation.

I-2-3) Matières solubles

Elles sont des matières dont la taille est inférieure à 0.01 μm . Leur élimination est effectuée durant le traitement biologique par des microorganismes.

I-2-4) Microorganismes

Les microorganismes sont essentiellement des bactéries étant capables de dégrader les matières polluantes présentes dans les eaux usées. Elles sont nécessaires pour le traitement biologique des eaux usées.

I-2-5) La demande chimique en oxygène (DCO)

La demande chimique en oxygène est un paramètre qui mesure la quantité d'oxygène nécessaire pour la dégradation chimique des matières organiques (présentes sous forme solide, colloïdale et dissoute) contenant dans un liquide. Ce paramètre est exprimé en $mg\ O_2.l^{-1}$.

I-2-6) La demande biologique en oxygène (DBO₅)

La demande biologique en oxygène est un paramètre qui mesure la quantité d'oxygène nécessaire à la destruction des matières organiques (présentes sous forme dissoute) contenant dans un liquide par une voie biologique. Il est utilisé pour mesurer le degré de contamination après cinq jours de réaction.

La demande biologique en oxygène DBO₅ se résume à la réaction chimique suivante:
$$\text{Substrat} + \text{microorganisme} + O_2 \rightarrow CO_2 + H_2O + \text{énergie} + \text{biomasse}$$

I-3) Définition des eaux usées

Les eaux usées sont des eaux dont la qualité a été altérée par les différents produits polluants (des déchets domestiques, industriels, agricoles ou autres) et par les milieux dans lesquels elles sont déversées. Ces eaux usées ne doivent pas être jetées dans les milieux naturels avant qu'elles ne soient traitées pour éliminer ces polluants indésirables.

I-4) Les différents types des eaux usées

Les pollutions qui se trouvent dans l'eau peuvent être d'origines: domestique, industrielle agricole et pluviale.

I-4-1) Les eaux usées domestiques

Les eaux usées domestiques sont les eaux provenant des différents usages domestiques tels que; les eaux de cuisine, les eaux des appareils ménagers, les eaux de toilettes et de salle de bain, etc. Elles sont essentiellement caractérisées par la présence des matières minérales, des matières organiques azotées et phosphatées.

I-4-2) Les eaux usées industrielles

Les eaux usées industrielles sont des eaux rejetées par les industries. Beaucoup d'industries utilisent une grande quantité d'eau au cours de production (lavage des matières premières, refroidissement, fabrication des laines, etc.). Elles sont caractérisées par la présence des différentes substances qui se diffèrent d'une industrie à une autre selon l'activité industrielle.

I-4-3) Les eaux usées agricoles

L'agriculture peut être considérée comme une source importante de la pollution des eaux. Dans le domaine agricole, divers produits d'origine industrielle et agricole sont utilisés tels que les engrais, les pesticides, les fertilisants, les produits phytosanitaires (insecticides, herbicides, fongicides, etc.). Les eaux issues des terres agricoles sont chargées par ces différents produits non assimilées ni par le sol ni par les plantes. Ce qui entraînera la pollution du milieu qui s'y trouvent.

I-4-4) Les eaux usées pluviales

Les eaux de pluies qui s'écoulent sur le sol, pourraient se mélanger avec des différents impuretés; résidus de pesticides, les huiles et les carburants des véhicules, etc. Elles sont souvent jetées directement dans les rivières causant une dégradation du milieu aquatique.

I-5) Définition d'une station d'épuration

Une station d'épuration est une usine qui a pour objectif de dépolluer les eaux usées de toutes sortes des polluants pour obtenir des eaux épurées dont la qualité répond aux normes mondiales exigées [14]. La station d'épuration comporte généralement une phase de prétraitement dans laquelle on élimine; les matières solides de grandes tailles, les sables et les graisses par différents systèmes (dégrillage, dessablage et déshuilage). Puis vient une phase du traitement primaire qui permet d'éliminer une grande partie des matières en suspension. Ensuite vient la phase du traitement biologique qui vise à éliminer les matières en suspension et les matières organiques dissoutes.

Chapitre I: Généralités sur le traitement des eaux usées

Généralement, après le traitement secondaire, les eaux épurées sont prêtes à être rejetées dans les milieux naturels.

Les différentes étapes du traitement ainsi que les tâches effectuées sont schématisées dans la figure 1.

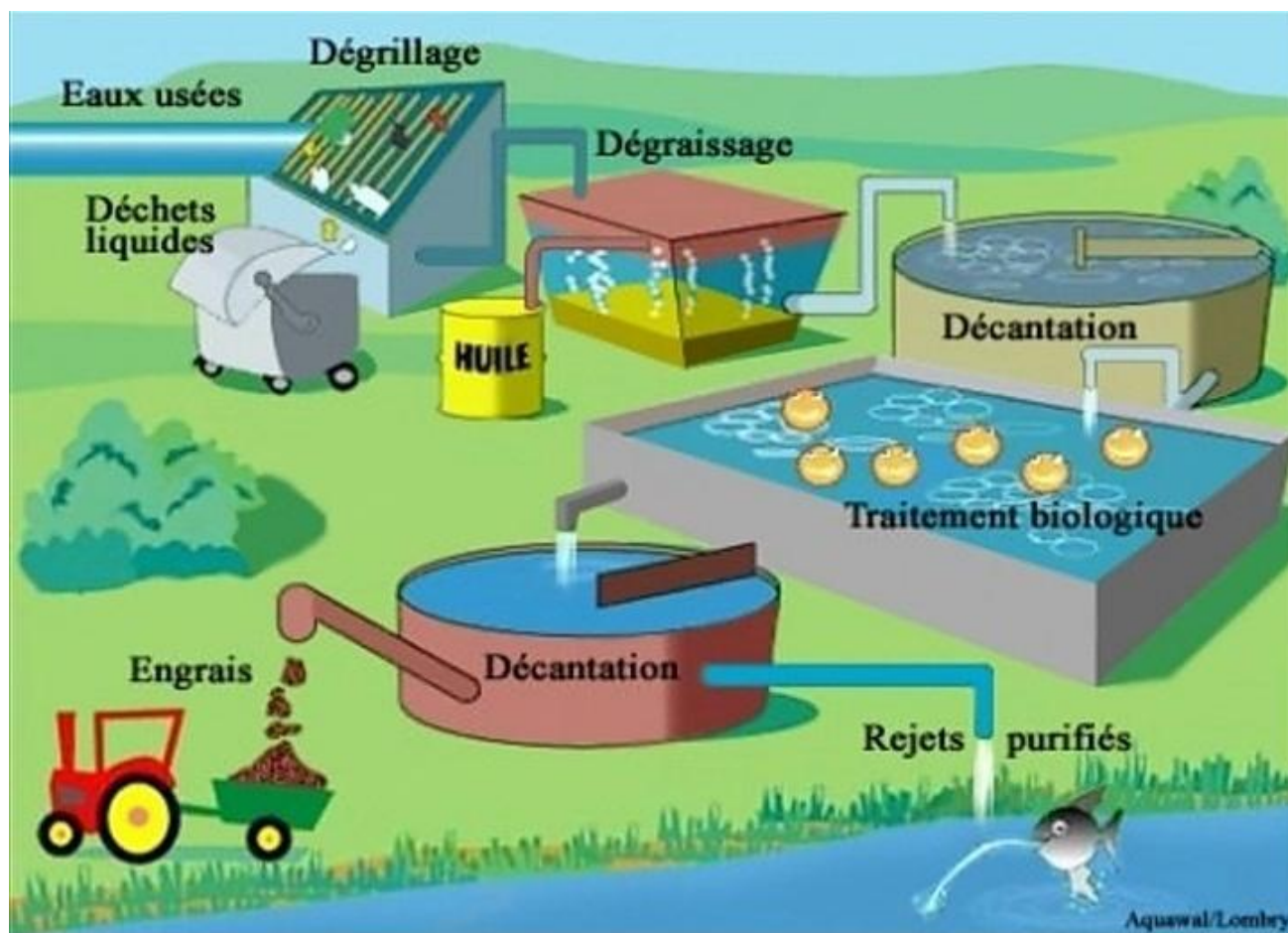


Figure I.1: Les étapes du traitement de la station d'épuration

I-6) Les grandes étapes du traitement des eaux usées

Les étapes du traitement des eaux usées peuvent être divisées en trois grandes étapes : le prétraitement, le traitement primaire et secondaire ou bien biologique. Ces différentes étapes du traitement ainsi que les différentes tâches effectuées sont présentées selon le degré d'élimination des matières polluantes et les procédés qui sont mis en œuvre.

I-6-1) Le prétraitement

Le prétraitement vise à éliminer les matières les plus grossières et volumineuses et les éléments susceptibles de gêner les autres étapes du traitement. Le prétraitement peut subdiviser en trois étapes successives liées les unes aux autres. Elles sont respectivement les suivantes:

* Le dégrillage: il permet d'éliminer les déchets volumineux tels que les flacons, les pierres, les plastiques, les tissus, ...etc. Le principe consiste à faire passer les eaux usées à travers des grilles métalliques afin d'en retirer les déchets volumineux [15]. Les déchets retirés sont déposés en décharge.

* Le dessablage: il a pour objectif de retirer les sables et les graviers pour protéger les organes mécaniques (pompes) et éviter de perturber les autres étapes du traitement. Les eaux usées venant de l'étape précédente (dégrillage) se retrouvent dans un bassin. Cependant, elles gardent des déchets tels que les sables et les graviers qui tomberont par la suite au fond du bassin par l'effet de leur poids, puis, ils sont récupérés par aspiration, ensuite, ils sont collectés pour être envoyés en décharges.

* Le dégraissage-déshuilage: il permet d'éliminer les graisses et les huiles pour éviter l'encrassement de la station d'épuration. Le principe consiste à injecter de l'air dans le bassin pour faire remonter les matières légères (les graisses et les huiles). La récupération des graisses et des huiles se fait à la surface. Les matières collectées seront incinérées ou recyclées pour la fabrication des savons ou détergents selon la qualité du lavage.

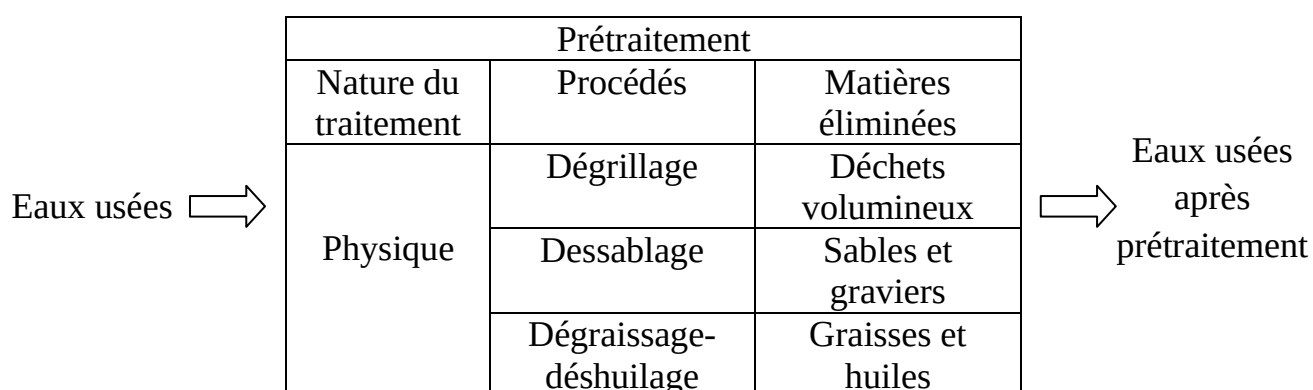


Figure I.2: Prétraitement

I-6-2) Le traitement primaire

Le traitement primaire permet d'éliminer le maximum des matières en suspension et de matières organiques facilement décantables [16]. Il comporte trois étapes du traitement:

* La décantation: elle permet d'éliminer jusqu'à 50 % des matières en suspension. Le principe consiste à séparer les matières solides des matières liquides (séparation solide-liquide). Les matières solides se déposent au fond du décanteur par l'effet de leur masse pour former les boues primaires [17].

* La flottation: elle permet d'éliminer les particules dont la masse volumique inférieure à celle du liquide qui les contient. Par opposition à la décantation, la flottation est un procédé de séparation solide-liquide ou liquide-liquide [18]. Les particules flottantes seront éliminées par raclage de surface.

* La coagulation et floculation: la coagulation sert à transformer les particules organiques liquide en matières solide. Elle consiste à déstabiliser les particules en injectant des produits chimiques pour faciliter leur agrégation [19]. La floculation permet d'agglomérer de substances colloïdales pour former des flocs plus gros afin de faciliter leur décantation. Les matières solides extraites forment les boues primaires.

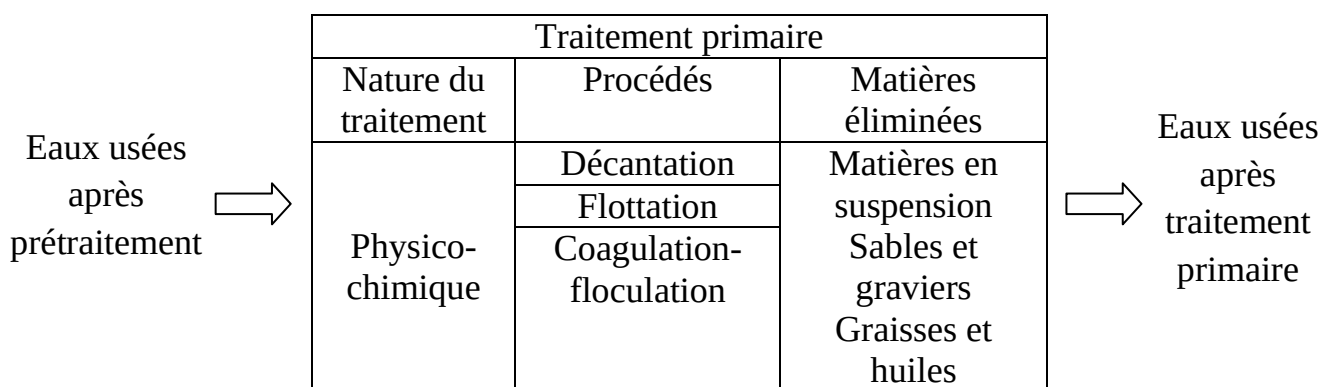


Figure I.3: Traitement primaire

I-6-3) Le traitement secondaire ou biologique

Le traitement biologique a pour objectif principal d'éliminer les matières organiques solubles qui sont échappées au traitement précédent (traitement primaire). Le principe

Chapitre I: Généralités sur le traitement des eaux usées

consiste à mettre en contact les eaux usées avec des micro-organismes capables de dégrader ou assimiler les matières organiques [20].

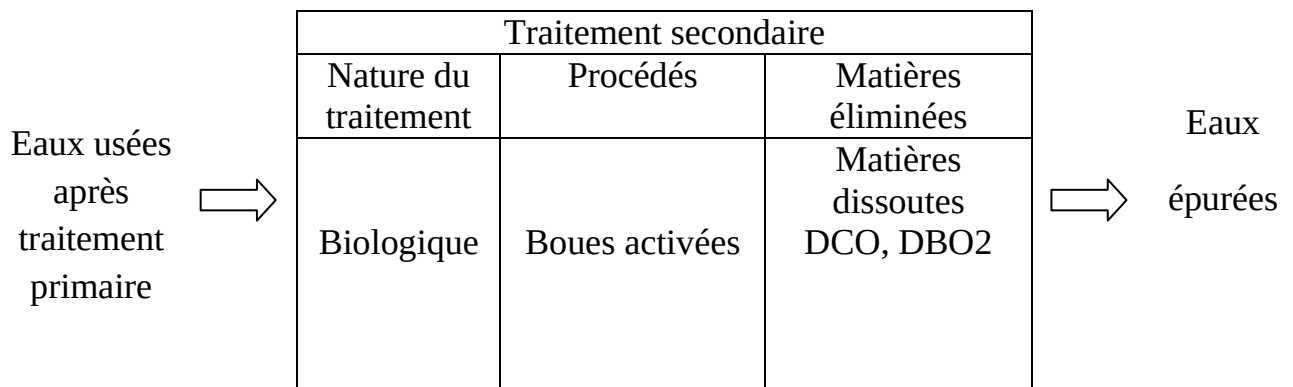


Figure I.4: Traitement secondaire ou biologique

Après que les eaux usées subissent ces étapes d'épuration, maintenant, elles sont prêtes à être jetées dans les milieux naturels (généralement des rivières).

I-7) Procédé de traitement des eaux usées par boues activées

Le but du traitement biologique est d'éliminer la pollution organique soluble (qui a échappé au traitement primaire) à l'aide des micro-organismes. La chaîne du traitement est composée d'un bioréacteur, d'un clarificateur et d'une boucle de recyclage des boues (figure 5) [21].

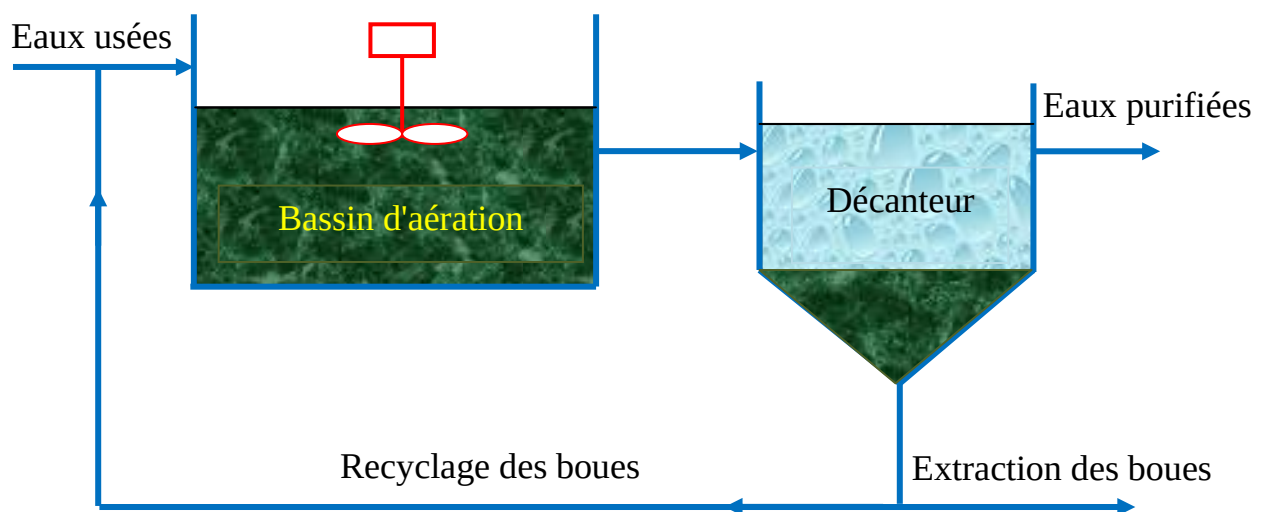


Figure I.5: Schéma général d'une station à boues activées

Les eaux usées venant du traitement primaire, seront dirigées vers un bassin d'aération où elles seront en contact avec des micro-organismes (biomasses) responsables de l'épuration.

Le bassin d'aération est un réacteur biologique alimenté en permanence en oxygène par des aérateurs de surface (turbine), suppresseur ou par un compresseur d'air. Les micro-organismes assimilent et transforment les matières organiques solubles en boues. En sortie du bioréacteur, on obtient un clarificateur qui a comme fonction de séparer les boues floculées de l'eau épurée par décantation.

Puisque la croissance des micro-organismes (biomasses) au sein du bioréacteur n'est pas suffisante, une partie des boues décantées est recyclée vers le bioréacteur afin de compenser la perte engendrée par le débit de sortie du bassin d'aération. Les boues en excès sont extraites pour rejoindre le système du traitement des boues. Les eaux épurées sont maintenant prêtes à être jetées dans la nature et les boues décantées seront déchargées.

Les eaux traitées après le traitement biologique contiennent des nutriments, donc, ils peuvent être utilisés dans l'irrigation des terres agricoles sans recours aux engrais artificiels qui sont coûteux [22-23]. Elles peuvent être utilisées aussi pour arroser les espaces verts et les jardins publics, éteindre les incendies, la récupération des sous-produits utiles tels que les papiers, les plastiques, etc.

I-8) Conclusion

Le traitement des eaux usées est un processus très important qui consiste à épurer les eaux usées de toutes sortes d'impuretés afin de préserver la santé des êtres humains, éviter les maladies microbiennes et protéger l'environnement.

Le traitement des eaux usées se fait dans une station d'épuration à travers plusieurs étapes du traitement et chaque étape traite un type particulier de pollution. Le traitement le plus important c'est le traitement biologique. Son principe de fonctionnement repose

sur le développement des micro-organismes qui se nourrissent des substances sous la présence de l'oxygène dissous.

Chapitre II: La résolution numérique des équations différentielles

II-1) Introduction

Les équations différentielles ont des applications nombreuses dans plusieurs domaines: physiques, chimiques, mécaniques et biologiques, etc [24], et il serait trop long d'en faire une liste. A la différence des équations différentielles linéaires, il n'existe pas des méthodes analytiques systématiques qui permettent de résoudre les équations différentielles non linéaires. Pour cette raison, les méthodes numériques sont existées afin de résoudre numériquement les équations différentielles non linéaires. Parmi les méthodes numériques les simples et les plus utilisées, c'est la méthode d'Euler.

La méthode d'Euler est une méthode numérique classique à un pas constant. Elle est utilisée pour résoudre par approximation des équations différentielles (linéaires ou non linéaires) du premier ordre avec une condition initiale [25].

Dans ce chapitre, on présente premièrement une introduction générale sur les équations différentielles ordinaires. Ensuite, une brève description sera donnée sur la résolution numérique des équations différentielles. Pour finir, on abordera la résolution numérique des équations différentielles par la méthode d'Euler (trois méthodes d'Euler seront expliquées: la méthode d'Euler explicite, implicite et modifiée).

II-2) Les équations différentielles ordinaires (EDO)

Une équation différentielle ordinaire (EDO) est une équation qui dépend d'une variable t et d'une fonction $y(t)$ et qui contient des dérivées de $y(t)$. La forme générale d'une équation différentielle d'ordre n s'écrit comme [26]:

$$F(t, y, \dot{y}, \ddot{y}, \dots, y^{(n)}) = 0 \quad (2.1)$$

L'ordre de cette équation différentielle est déterminé par l'ordre le plus élevé de ses dérivées.

y : représente une fonction d'une variable t , et $y, \dot{y}, \ddot{y}, \dots, y^{(n)}$ représentent ses dérivées successives.

Chapitre II: La résolution numérique des équations différentielles

On va se limiter aux équations différentielles du premier ordre, car n'importe quelle équation différentielle d'ordre $n > 1$ peut être ramenée à un système de n équations d'ordre 1.

II-3) La résolution numérique des équations différentielles

L'immense majorité des systèmes physiques, chimiques, mécaniques et biologiques, etc, pourraient être décrits par des équations différentielles non linéaires. Cependant, il faut noter que la résolution analytique ou explicite des équations différentielles non linéaires n'est pas souvent simple et dans la plupart du temps sera trop difficile, voire même impossible. Donc, on fait recours à des méthodes numériques.

Les méthodes numériques employées pour résoudre les équations différentielles sont des méthodes approximatives basées sur la discrétisation de la variable du temps t en un certain nombre de pas. On peut distinguer plusieurs méthodes numériques: des méthodes numériques à un pas fixe ou à un pas variable, des méthodes explicites ou implicites.

II-4) La méthode d'Euler

La méthode d'Euler est la première méthode de résolution numérique des équations différentielles. Elle fut introduite par Leonhard Euler en 1768 [27-28]. Son principe consiste à discrétiser l'intervalle de la variable du temps t en un certain nombre de pas fixe h .

II-4-1) La méthode d'Euler explicite

Soit une fonction f définie sur l'intervalle $I = [t_0, t_{k+1}]$, $k \in N$ et une fonction y dérivable sur I telle que: $y(t_0) = y_0$, ($y_0 \in R$) et $\dot{y}(t) = f(t, y(t))$ pour tout $t \in I$ est une primitive de f [29].

La résolution numérique de l'équation $\dot{y}(t) = f(t, y(t))$ consiste à discrétiser l'intervalle $I = [t_0, t_{k+1}]$ en $k + 1$ sous-intervalles de même longueur h avec $t_1 - t_0 = t_2 - t_1 = \dots = t_{k+1} - t_k = h$ (h est appelé le pas de discrétisation).

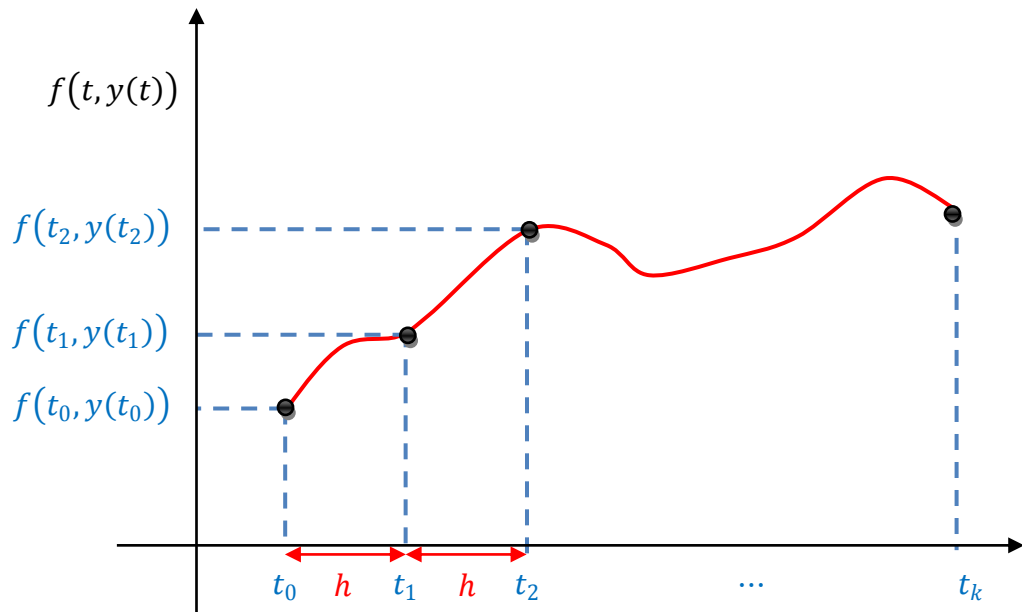


Figure II.1: Principe de la méthode d'Euler

Soit l'équation différentielle suivante:

$$\frac{dy}{dt} = f(t, y(t)), \quad y(t_0) = y_0 \quad (2.2)$$

Trouver la solution de cette équation différentielle (2.2) revient à calculer l'intégrale $f(t, y(t))$ entre les deux points t_0 et t_1 :

$$\int_{t_0}^{t_1} \frac{dy}{dt} dt = \int_{t_0}^{t_1} f(t, y(t)) dt \quad (2.3)$$

$$y(t_1) - y(t_0) = \int_{t_0}^{t_1} f(t, y(t)) dt \quad (2.4)$$

On peut donc calculer $y(t_1)$ en connaissant $y(t_0)$ (condition initiale) à condition qu'on doit savoir aussi calculer l'intégrale $\int_{t_0}^{t_1} f(t, y(t)) dt$.

Généralement, on n'arrive pas à calculer cette intégrale puisqu'on ne connaît pas la solution exacte. Par contre, on peut l'approcher à l'aide des méthodes d'intégration numériques.

L'approximation de l'intégrale $\int_{t_0}^{t_1} f(t, y(t)) dt$ peut être construite par différents schémas: schéma des rectangles à gauche, rectangles à droite, rectangles à point milieu.

Si on utilise le schéma des rectangles à gauche, on obtient la figure 2.

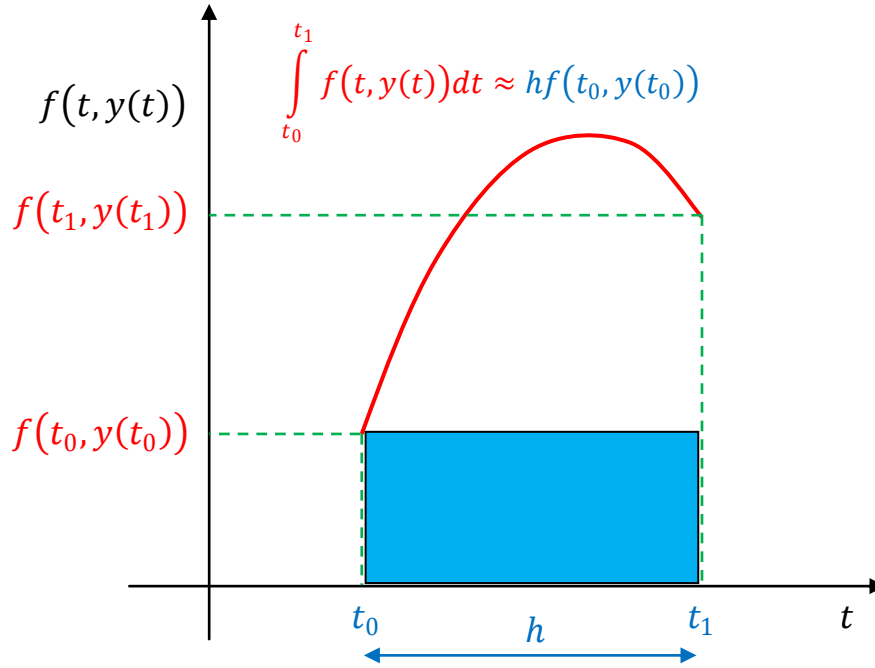


Figure II.2: Méthode d'Euler explicite: schéma du rectangle à gauche

L'approximation de l'intégrale $\int_{t_0}^{t_1} f(t, y(t)) dt$ par la méthode du rectangle à gauche revient à calculer l'aire de ce rectangle, on obtient:

$$\int_{t_0}^{t_1} f(t, y(t)) dt \approx hf(t_0, y(t_0)) \quad (2.5)$$

Si on remplace l'approximation de l'intégrale $\int_{t_0}^{t_1} f(t, y(t)) dt$ dans l'équation (2.4), on obtient alors la méthode d'Euler explicite [30]:

$$y(t_1) = y(t_0) + hf(t_0, y(t_0)) \quad (2.6)$$

Remplaçant maintenant la valeur exacte $y(t_1)$ par la valeur que l'on espère approchée y_1 , on aura:

$$y_1 = y_0 + hf(t_0, y_0), \quad t_1 = t_0 + h \quad (2.7)$$

Pour les valeurs $k + 1$ l'algorithme de la méthode d'Euler explicite s'écrit:

$$y_{k+1} = y_k + hf(t_k, y_k), \quad t_{k+1} = t_k + h \quad (2.8)$$

II-4-2) La méthode d'Euler implicite

En utilisant la méthode des rectangles à droite pour approximer l'intégrale $\int_{t_0}^{t_1} f(t, y(t)) dt$, ce qui donne le schéma numérique d'Euler implicite (figure 3).

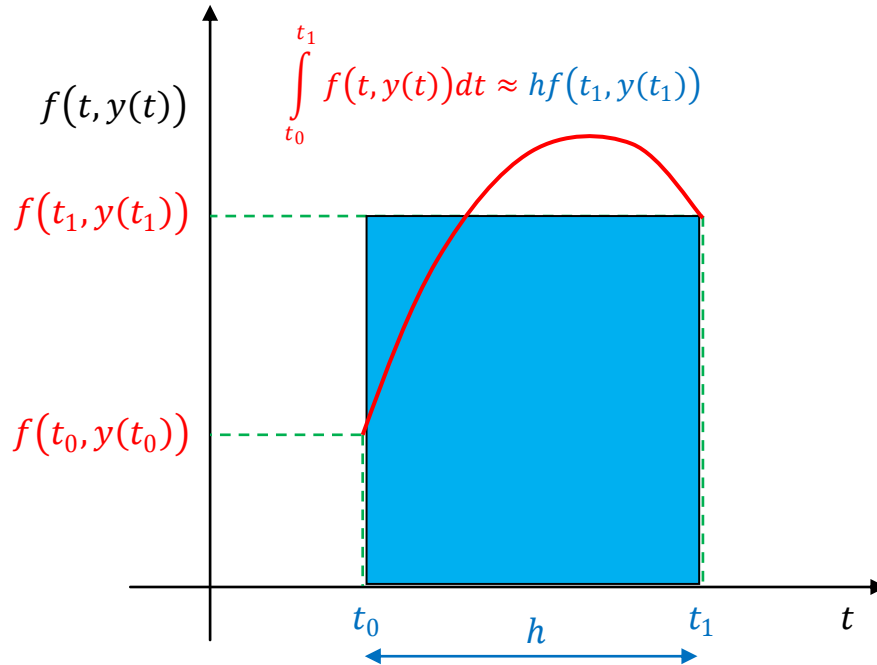


Figure II.3: Méthode d'Euler implicite: schéma du rectangle à droite

Le même procédé appliqué à la méthode des rectangles à gauche est utilisé avec celle des rectangles à droite. On obtient:

$$y(t_1) = y(t_0) + hf(t_1, y(t_1)) \quad (2.9)$$

Donc, l'algorithme de la méthode d'Euler implicite s'écrit comme:

$$y_{k+1} = y_k + hf(t_{k+1}, y_{k+1}), \quad t_{k+1} = t_k + h \quad (2.10)$$

Il s'agit d'une méthode implicite car elle ne permet pas dans tout les cas d'explicitier y_{n+1} en fonction de y_n .

II-4-3) La méthode d'Euler modifiée

On peut aussi approcher l'intégrale $\int_{t_0}^{t_1} f(t, y(t)) dt$ par la méthode des rectangles à point milieu, on trouve le schéma numérique d'Euler modifiée (figure 4).

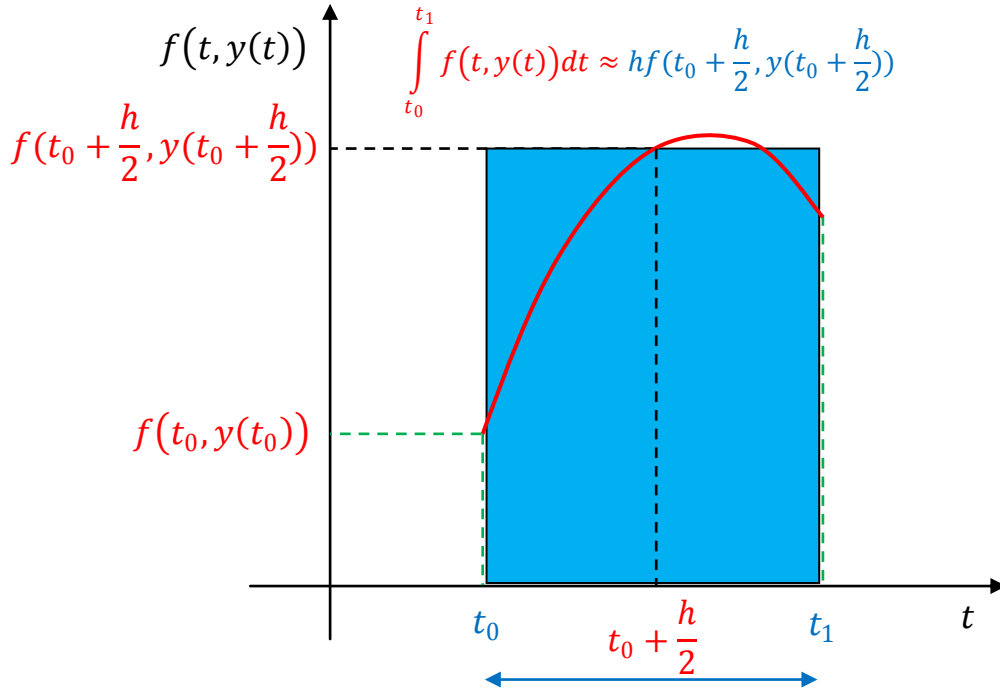


Figure II.4: Méthode d'Euler modifiée: schéma du rectangle à point milieu

L'algorithme de la méthode d'Euler modifiée est donné par l'équation suivante:

$$y(t_1) = y(t_0) + hf\left(t_0 + \frac{h}{2}, y\left(t_0 + \frac{h}{2}\right)\right) \quad (2.11)$$

Cette équation ne permet pas de construire directement un schéma numérique, puisque la valeur $y\left(t_0 + \frac{h}{2}\right)$ ne correspond pas à un point de discrétisation. Cependant, On peut utiliser le développement limité de série de Taylor pour approximer le terme $y\left(t_0 + \frac{h}{2}\right)$:

$$y\left(t_0 + \frac{h}{2}\right) \approx y(t_0) + \frac{h}{2}y'(t_0) = y(t_0) + \frac{h}{2}f(t_0, y(t_0)) \quad (2.12)$$

L'algorithme de la méthode d'Euler modifiée devient alors:

$$y(t_1) = y(t_0) + hf\left(t_0 + \frac{h}{2}, y(t_0) + \frac{h}{2}f(t_0, y(t_0))\right) \quad (2.13)$$

Remplaçant maintenant la valeur exacte $y(t_1)$ par des valeurs approchées y_1 , on obtient:

$$y_1 = y_0 + hf\left(t_0 + \frac{h}{2}, y_0 + \frac{h}{2}f(t_0, y_0)\right) \quad (2.14)$$

Chapitre II: La résolution numérique des équations différentielles

Pour les valeurs $k + 1$ l'algorithme de la méthode d'Euler modifiée s'écrit par les équations suivantes:

$$\begin{cases} k_1 = y_k + \frac{h}{2} f(t_k, y_k) \\ y_{k+1} = y_k + hf\left(t_k + \frac{h}{2}, k_1\right), \quad t_{k+1} = t_k + h \end{cases} \quad (2.15)$$

Exemple:

Soit une équation différentielle non linéaire dont on connaît la solution exacte pour pouvoir faire une comparaison avec les méthodes numériques celles d'Euler explicite et modifiée:

$$\frac{dy}{dt} = f(t, y(t)) = t - ty(t), \quad y(0) = 3.5 \quad (2.16)$$

L'algorithme de la méthode d'Euler explicite est donné par l'équation suivante:

$$y_{k+1} = y_k + hf(t_k, y_k) = y_k + h(t_k - t_k y_k), \quad t_{k+1} = t_k + h \quad (2.17)$$

L'algorithme de la méthode d'Euler modifiée est le suivant:

$$\begin{cases} k_1 = y_k + \frac{h}{2} f(t_k, y_k) \\ y_{k+1} = y_k + hf\left(t_k + \frac{h}{2}, \left(t_k + \frac{h}{2}\right) k_1\right), \quad t_{k+1} = t_k + h \end{cases} \quad (2.18)$$

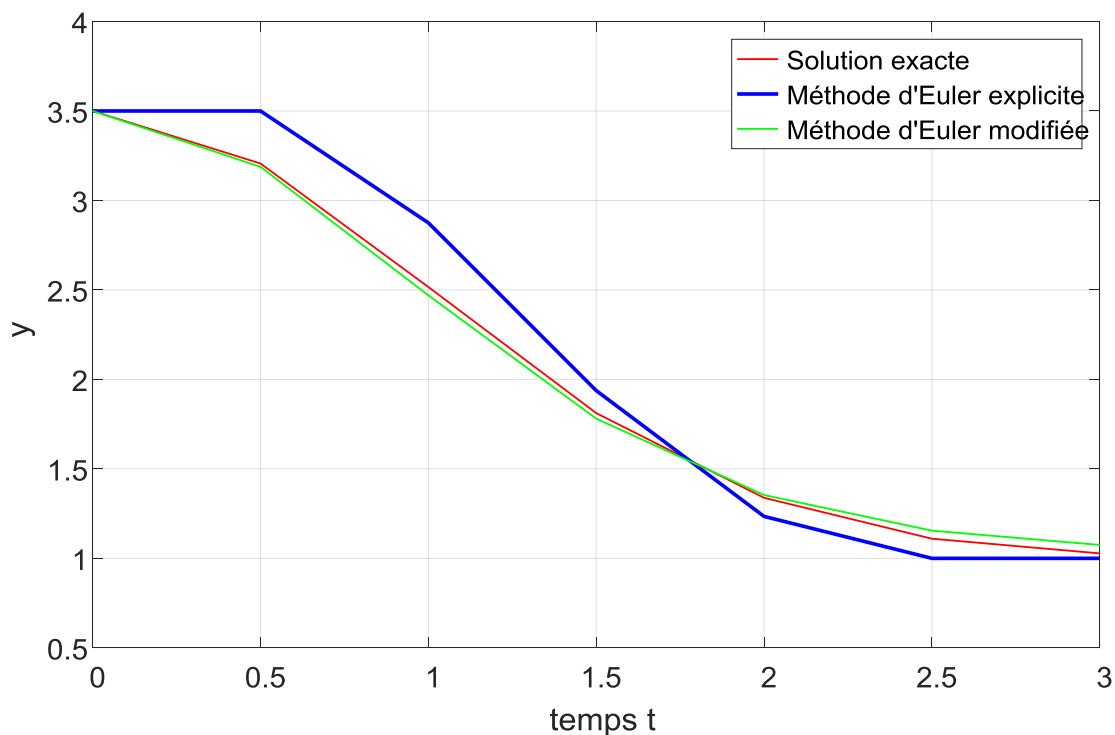


Figure II.5: Méthode d'Euler explicite et modifiée

Les valeurs numériques de la solution exacte de cette équation différentielle non linéaire ainsi que sa résolution numérique par la méthode d'Euler explicite et modifiée sont montrées dans le tableau 1.

Itérations k	La méthode d'Euler explicite	La méthode d'Euler modifiée	La solution exacte
1	3.5000	3.5000	3.5000
2	3.5000	3.1875	3.2062
3	2.8750	2.4697	2.5163
4	1.9375	1.7808	1.8116
5	1.2344	1.3538	1.3383
6	1.0000	1.1548	1.1098
7	1.0000	1.0750	1.0278

Tableau II.1: Comparaison de la performance

Le pas de discrétisation utilisé $h = 0.5$.

Les résultats donnés dans la figure 5 ou par le tableau montrent clairement que la méthode d'Euler modifiée est plus précise que celle explicite.

Remarque:

Les résultats peuvent être améliorés en utilisant un pas de discrétisation plus petit.

II-5) Conclusion

La résolution analytique des équations différentielles non linéaires n'est pas souvent faisable, voire même impossible. Le seul moyen qui existait pour résoudre ces équations différentielles non linéaire, c'est la résolution numérique.

Il existe deux types des méthodes numériques permettant de résoudre les équations différentielles non linéaire: les méthodes numériques qui utilisent un pas fixe et celles avec un pas variable. La méthode d'Euler est parmi les méthodes numériques qui utilisent un pas fixe.

Dans ce chapitre, trois algorithmes dérivés de la méthode d'Euler ont été expliqués; l'algorithme de la méthode d'Euler explicite, implicite et modifiée. Les résultats montrent que la méthode d'Euler modifiée présente moins d'erreur que celle explicite.

Chapitre III: La méthode de descente du gradient

III-1) Introduction

Les problèmes d'optimisation, particulièrement les problèmes d'optimisation numérique se posent dans plusieurs domaines: informatique, ingénierie, recherche opérationnelle, économie, théorie du contrôle et de la commande, etc.

L'optimisation est une branche des mathématiques qui a pour objectif de chercher la meilleure solution d'un problème (par rapport à un critère) en faisant changer certains paramètres. Parmi les méthodes d'optimisation les plus utilisées, on peut distinguer la méthode de descente du gradient. C'est une méthode d'optimisation numérique qui cherche à minimiser ou maximiser un problème.

La méthode de descente du gradient est utilisée dans l'entraînement des modèles en apprentissage tels que: la régression logistique, les machines Learning et notamment les réseaux de neurones artificiels.

Il existe plusieurs dérivés de l'algorithme de descente du gradient tels que : GD, GD avec momentum, Adagrad, Adadelata, NAG, RMSprop, Adam, Nadam [31-37]. Ici, on ne présente que les trois premiers algorithmes cités.

Ce chapitre s'intéresse sur la méthode de descente du gradient. On commence tout d'abord par donner une définition de la méthode de descente du gradient. Ensuite, trois algorithmes dérivés de la méthode du gradient seront expliqués: l'algorithme du gradient avec un taux d'apprentissage fixe, l'algorithme du gradient adaptatif, et l'algorithme du gradient avec un momentum. Pour chacun de ces algorithmes nous présenterons des exemples démonstratifs. On finira par une conclusion qui récapitule les avantages et les inconvénients de chaque algorithme utilisé.

III-2) Méthode de descente du gradient

L'algorithme de descente du gradient fait partie d'une famille des algorithmes à directions de descente. Le principe de cette méthode de descente est qu'à partir d'un point actuel, on effectue un déplacement dans la direction opposée par rapport au gradient afin

de trouver le point suivant dans lequel la fonction à minimiser prend une valeur inférieure à celle qu'elle a en point actuel. Il existe plusieurs méthodes permettant de déterminer la direction de descente, parmi les méthodes les plus utilisées, c'est la méthode du gradient.

III-2-1) Méthode du gradient à pas fixe

Soit f la fonction objectif à minimiser. Pour ce faire, on se donne un point de départ x_0 arbitrairement choisi pour déterminer l'itération suivante x_1 . Il faut penser qu'on veut se rapprocher du minimum de la fonction objectif f en utilisant un algorithme de descente. Pour que cet algorithme puisse converger vers la valeur minimale de la fonction objectif f , il doit satisfaire la condition: $f(x_1) \leq f(x_0)$.

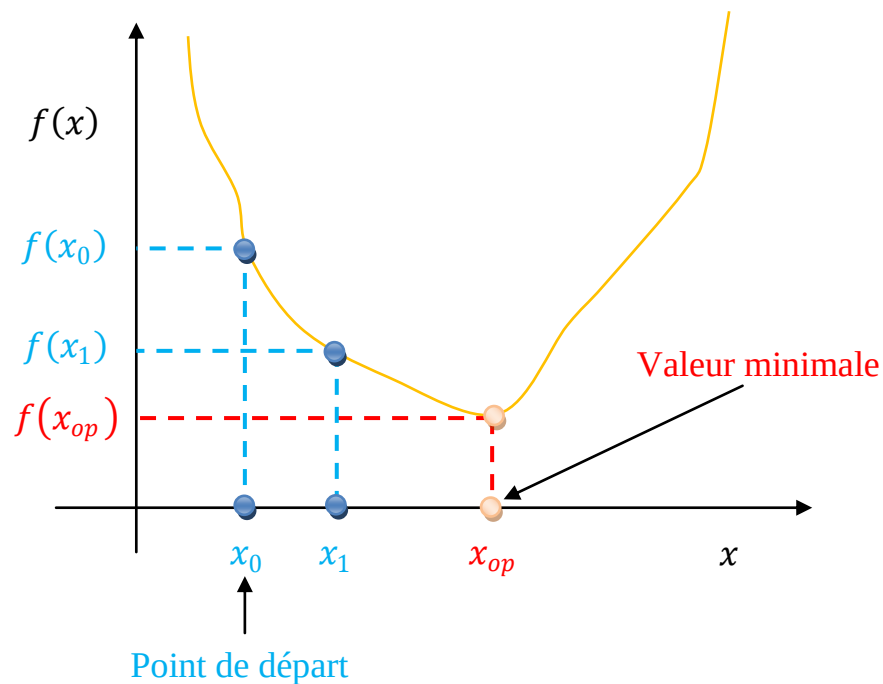


Figure III.1: Principe de la méthode de la descente du gradient

On cherche alors x_1 sous la forme $x_1 = x_0 + \rho_0 d_0$, où d_0 est un vecteur non nul de R^n et ρ_0 un réel strictement positif.

On appelle d_0 direction de descente et ρ_0 est le pas de descente ou bien le taux d'apprentissage.

La condition qui garantit la convergence de l'algorithme vers la valeur minimale de la fonction objectif f est: $f(x_1) \leq f(x_0)$.

Chapitre III: La méthode de descente du gradient

$$f(x_1) \leq f(x_0) \Rightarrow f(x_0 + \rho_0 d_0) \leq f(x_0) \quad (3.1)$$

Donc, on cherche à déterminer l'expression de la direction de descente d_0 qui satisfait la condition: $f(x_0 + \rho_0 d_0) \leq f(x_0)$.

On remplace la fonction $f(x_1)$ au voisinage de x_0 par son développement de Taylor de premier ordre:

$$f(x_1) = f(x_0 + \rho_0 d_0) = f(x_0) + \rho_0 f'(x_0) d_0 \quad (3.2)$$

$$f(x_1) \leq f(x_0) \Rightarrow f(x_0) + \rho_0 f'(x_0) d_0 \leq f(x_0). \text{ Donc: } \rho_0 f'(x_0) d_0 \leq 0.$$

Si on choisit $d_0 = -f'(x_0)$, on est sûr que $-\rho_0 (f'(x_0))^2 \leq 0$, donc, l'algorithme de la méthode du gradient s'écrit comme:

$$x_1 = x_0 + \rho_0 d_0 \quad (3.3)$$

$$\text{Avec : } d_0 = -f'(x_0).$$

Le schéma général d'une méthode de descente est le suivant :

$$x_{k+1} = x_k + \rho_k d_k \quad (3.4)$$

$$\text{Avec : } d_k = -f'(x_k).$$

On appelle méthode du gradient à pas fixe si $\rho_k = \rho$.

Exemple 1:

On veut minimiser la fonction $f(x) = (x - 1)^2 + 4.5$ en utilisant la méthode du gradient avec différentes valeurs du taux d'apprentissage ρ . Le point initial $x_0 = 5$.

L'algorithme de la méthode de descente du gradient à pas fixe peut s'écrire:

$$x_{k+1} = x_k + \rho d_k.$$

$$\text{Avec: } d_k = -\frac{\partial f(x)}{\partial x_k} = -2(x_k - 1).$$

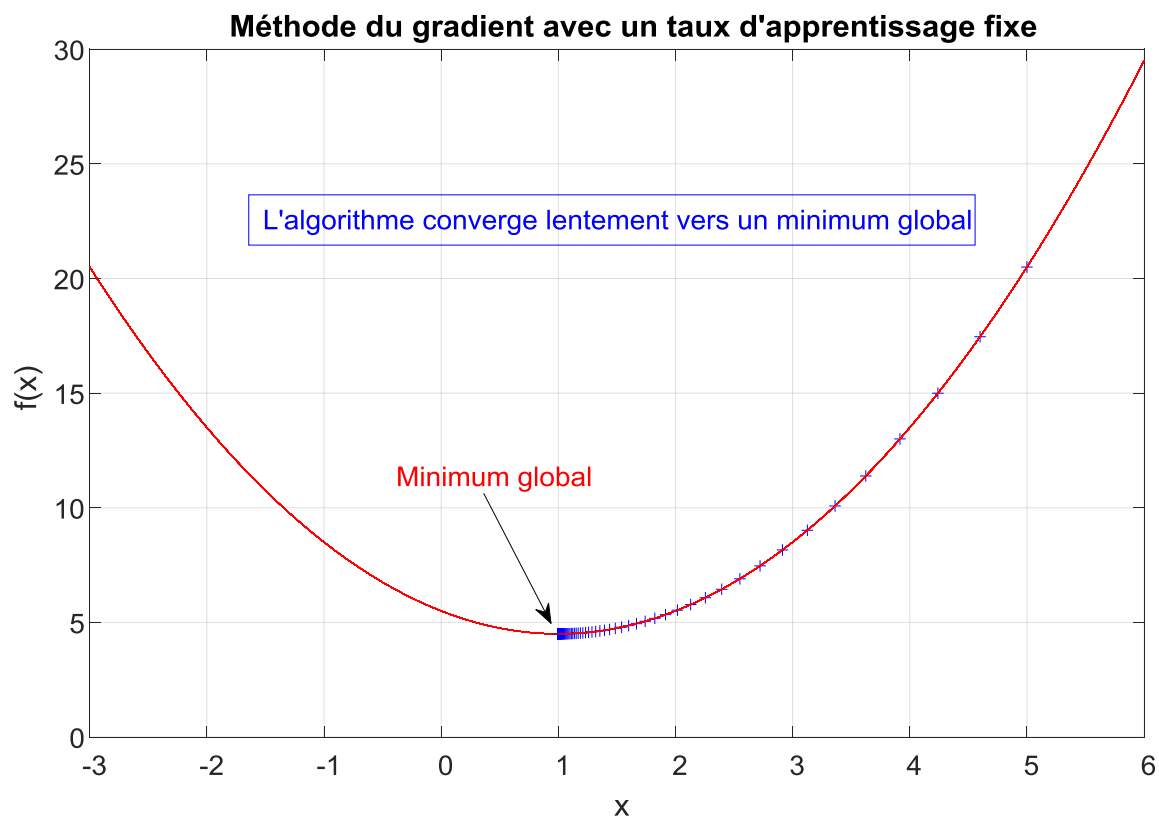


Figure III.2: Méthode du gradient: $\rho = 0.05$

Avec un pas $\rho = 0.05$, l'algorithme a besoin des nombreuses itérations (108 itérations) pour converger vers un minimum global (la valeur $x = 1$). Si on remplace cette valeur dans la fonction f , on aura donc, le minimum de la fonction f : $\min f(x) = 4.5$.

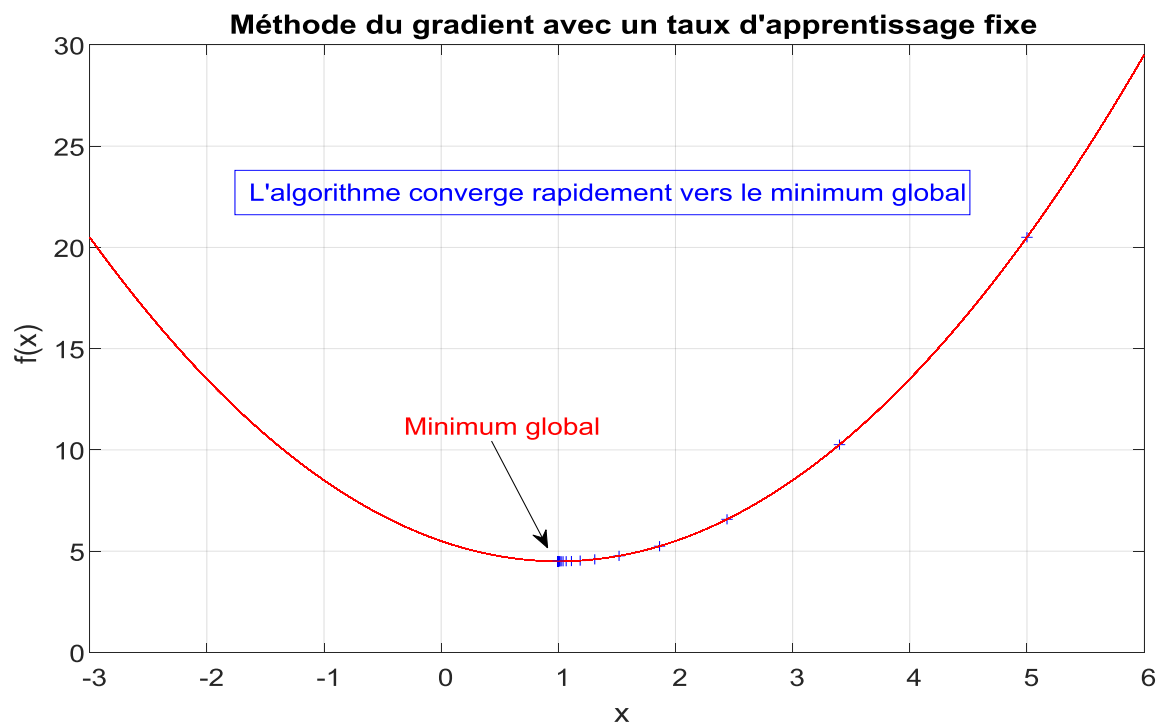


Figure III.3: Méthode du gradient: $\rho = 0.2$

Avec un pas $\rho = 0.2$, l'algorithme converge vers un minimum global après seulement 23 itérations.

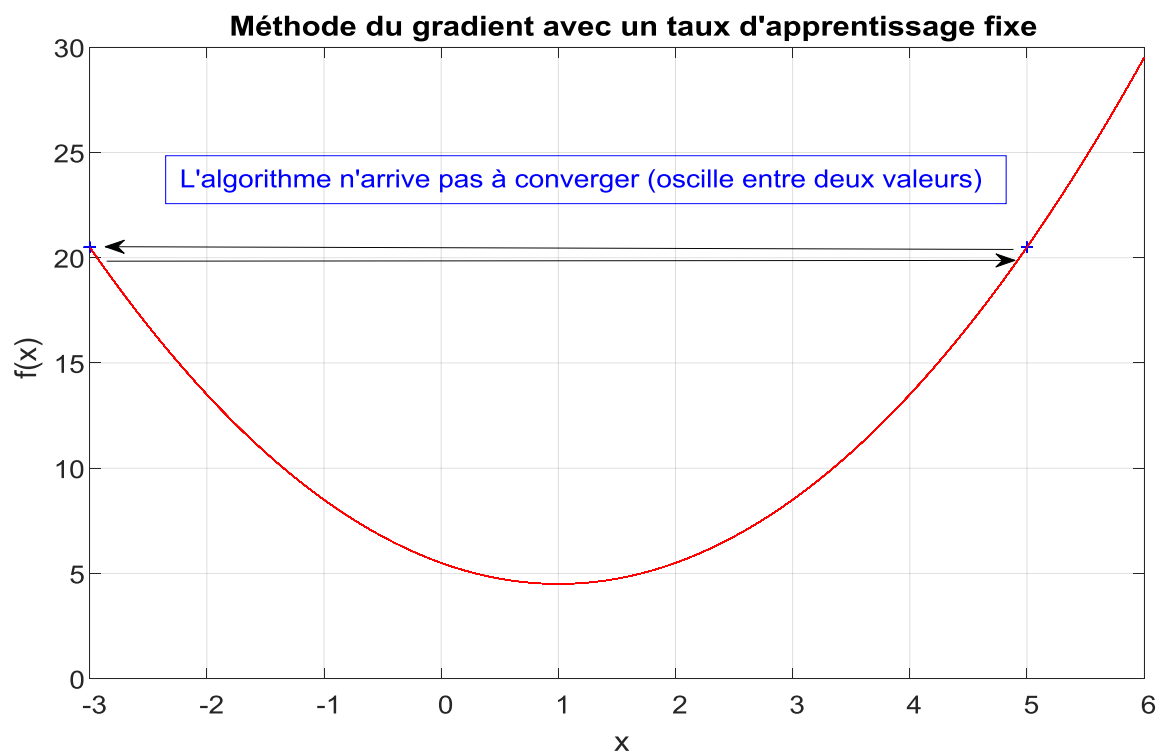


Figure III.4: Méthode du gradient: $\rho = 1$

Chapitre III: La méthode de descente du gradient

On remarque bien que l'algorithme oscille indéfiniment entre les deux valeurs -3 et 5, donc, l'algorithme ne peut pas converger.

Conclusion:

Le taux d'apprentissage ρ a une influence très importante sur la vitesse de convergence de la méthode du gradient:

- * Si on utilise un très petit taux d'apprentissage, l'algorithme convergera lentement.
- * Si on utilise un grand taux d'apprentissage, l'algorithme convergera rapidement.
- * Cependant, si on choisit un trop grand taux d'apprentissage, on aura un problème d'oscillation ou de zigzag et l'algorithme peut même diverger.

III-2-2) Méthode du gradient adaptatif (AdaGrad)

La méthode du gradient adaptatif ou AdaGrad [31] est une méthode d'optimisation qui modifie le taux d'apprentissage à chaque itération en divisant le taux d'apprentissage par la racine carrée de G qui est la somme accumulée du gradient actuel et les gradients passés au carrés. Son algorithme est donné par les équations suivantes:

$$x_{k+1} = x_k - \frac{\rho}{\sqrt{G_{k+1} + \varepsilon}} f'(x_k) \quad (3.5)$$

$$G_{k+1} = G_k + f'^2(x_k) \quad (3.6)$$

G_{k+1} : il permet d'accumuler la somme du gradient actuel et les gradients passés au carrés (chaque gradient ajouté doit être positif) [38].

La raison pour laquelle on a utilisé l'opérateur racine dans le dénominateur est pour compenser le carré des gradients accumulés en G_{k+1} .

ε : est un constant de petite valeur (valeur par défaut $\varepsilon = 10^{-7}$) qui a été ajouté au dénominateur afin d'éviter le problème de la division sur zéro quand G_{k+1} devient nul $G_{k+1} = 0$.

Chapitre III: La méthode de descente du gradient

L'avantage de la méthode du gradient adaptatif est qu'il élimine le besoin de modifier le taux d'apprentissage manuellement.

Son inconvénient majeur est l'accumulation des gradients carrés dans le dénominateur. La somme accumulée ne cesse pas de croître au cours de l'apprentissage, ce qui fait que le taux d'apprentissage diminue et devient infiniment petit.

Exemple 2:

On cherche à minimiser les deux fonctions $f_1(x) = 0.3(x - 1)^2 + 4.5$ et $f_2(x) = 1.5(x - 1)^2 + 4.5$ avec la méthode du gradient avec un taux d'apprentissage fixe et la méthode du gradient adaptatif.

L'algorithme de la méthode de descente du gradient avec un taux d'apprentissage fixe:

* Pour la fonction $f_1(x)$: $x_{k+1} = x_k - \rho f_1'(x_k) = x_k - 0.6\rho(x_k - 1)$

* Pour la fonction $f_2(x)$: $x_{k+1} = x_k - \rho f_2'(x_k) = x_k - 3\rho(x_k - 1)$

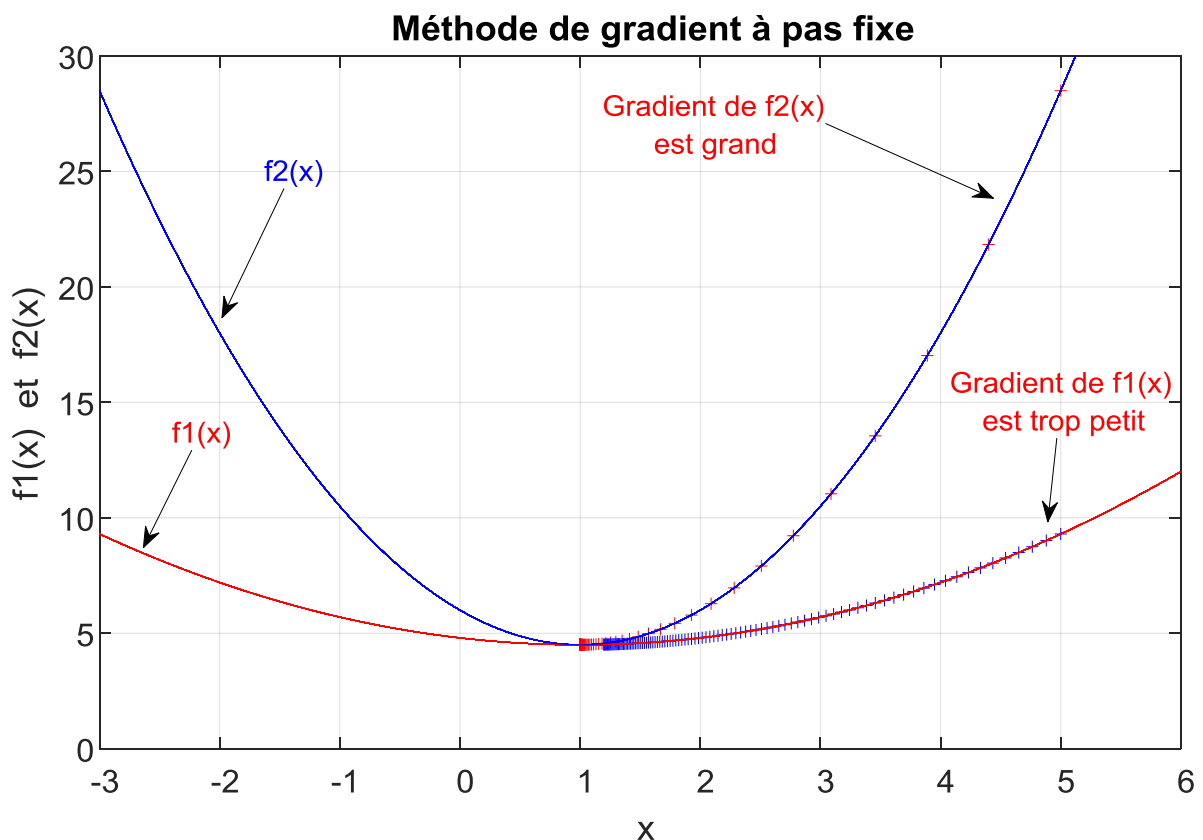


Figure III.5: Méthode du gradient à pas fixe: $\rho = 0.05$

Chapitre III: La méthode de descente du gradient

La figure 5 montre clairement l'inconvénient de la méthode de gradient avec un taux d'apprentissage fixe.

La courbe de la fonction f_2 montre bien que l'algorithme converge rapidement, parce que les valeurs des gradients de la fonction f_2 commencent grandes et diminuent légèrement. Contrairement pour la fonction f_1 , l'algorithme converge lentement, parce que les valeurs des gradients de la fonction f_1 commencent très petites et deviennent infiniment petites.

L'algorithme de la méthode du gradient adaptatif peut remédier à ce problème: Quand les valeurs des gradients commencent très petites (courbe de la fonction f_1), elles seront accumulées en G_{k+1} et le taux d'apprentissage $\frac{\rho}{\sqrt{G_{k+1} + \varepsilon}}$ devient alors élevé, il est ensuite multiplié par la composante du gradient actuelle $f'(x_k)$, ce qui augmente la vitesse de convergence de l'algorithme et vice versa. Quand les valeurs des gradients commencent très grandes (l'algorithme peut avoir un problème d'oscillation ou risque de divergence), elles seront accumulées en G_{k+1} , ce qui donne un taux d'apprentissage petit.

Maintenant, si on utilise l'algorithme de la méthode du gradient adaptatif:

Pour la fonction $f_1(x)$:

$$\begin{cases} G_{k+1} = G_k + f_1'^2(x_k) = G_k + 0.36(x_k - 1)^2 \\ x_{k+1} = x_k - \frac{\rho}{\sqrt{G_{k+1} + \varepsilon}} f_1'(x_k) = x_k - 0.6 \frac{\rho}{\sqrt{G_{k+1} + \varepsilon}} (x_k - 1) \end{cases}$$

Pour la fonction $f_2(x)$:

$$\begin{cases} G_{k+1} = G_k + f_2'^2(x_k) = G_k + 9(x_k - 1)^2 \\ x_{k+1} = x_k - \frac{\rho}{\sqrt{G_{k+1} + \varepsilon}} f_2'(x_k) = x_k - 3 \frac{\rho}{\sqrt{G_{k+1} + \varepsilon}} (x_k - 1) \end{cases}$$

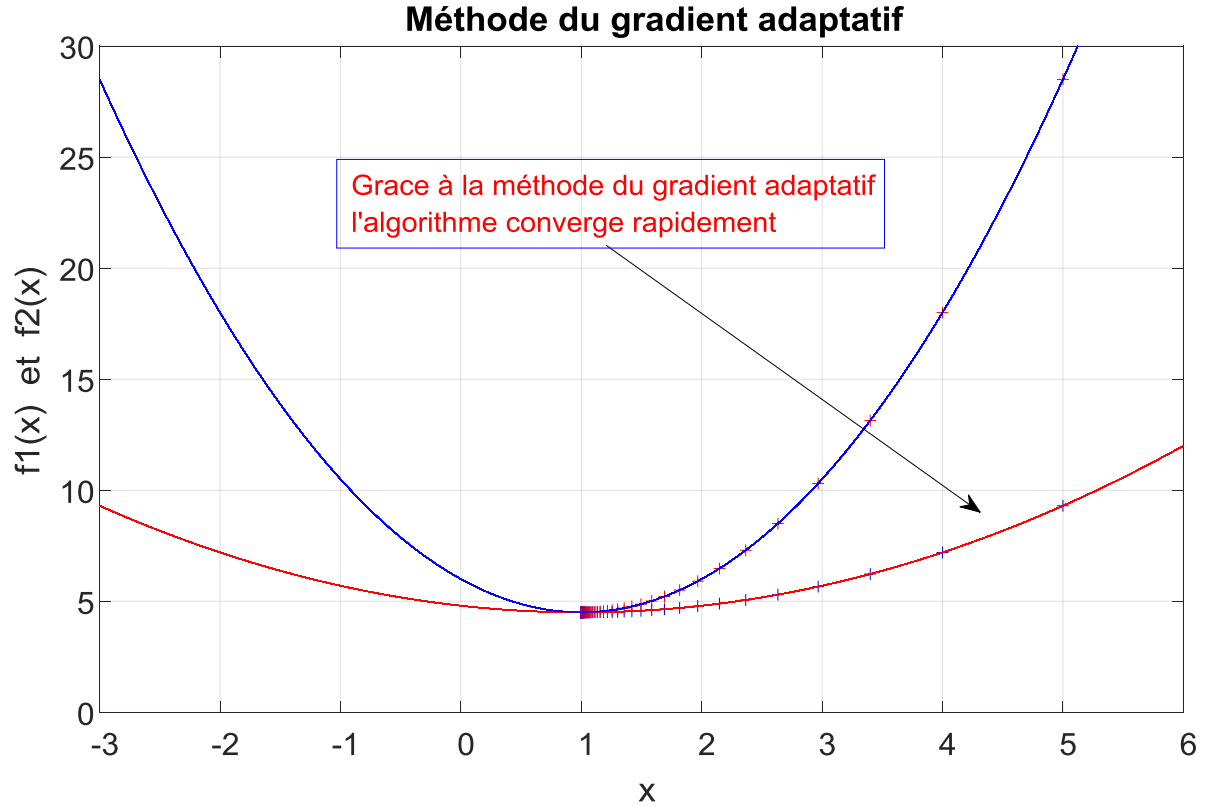


Figure III.6: Méthode du gradient adaptatif

Plus les valeurs des gradients accumulés en G_{k+1} sont petites, plus la valeur de G_{k+1} sera petite, ce qui entraîne un taux d'apprentissage plus élevé $\frac{\rho}{\sqrt{G_{k+1} + \varepsilon}}$.

III-2-3) Méthode de gradient avec momentum

La méthode du gradient avec momentum [32] [39] est similaire à celle de gradient avec un taux d'apprentissage fixe. Au lieu d'utiliser uniquement la valeur du gradient actuel dans l'algorithme, un autre terme m dit d'inertie (momentum) sera ajouté qui est un agrégat des gradients actuels et passés.

$$x_{k+1} = x_k - \rho m_k$$

$$m_k = \beta x_{k-1} + (1 - \beta) \frac{\partial f(x)}{\partial x_k}$$

La valeur par défaut commune: $\beta = 0.9$.

Chapitre III: La méthode de descente du gradient

La méthode du gradient avec momentum possède deux intérêts:

- * Elle aurait plus de chance de converger vers un minimum global.
- * Elle réduit le phénomène d'oscillation et accélère la vitesse de convergence [40].

L'inconvénient de la méthode de gradient avec momentum reste dans le choix du taux d'apprentissage (doit être réglé manuellement).

Exemple 3:

On veut minimiser la fonction $f = (x - 2)^2(x^2 + 4x + 2)$ en utilisant la méthode de descente du gradient avec et sans momentum. Le pas de descente: $\rho = 0.05$ et $x_0 = 3.5$.

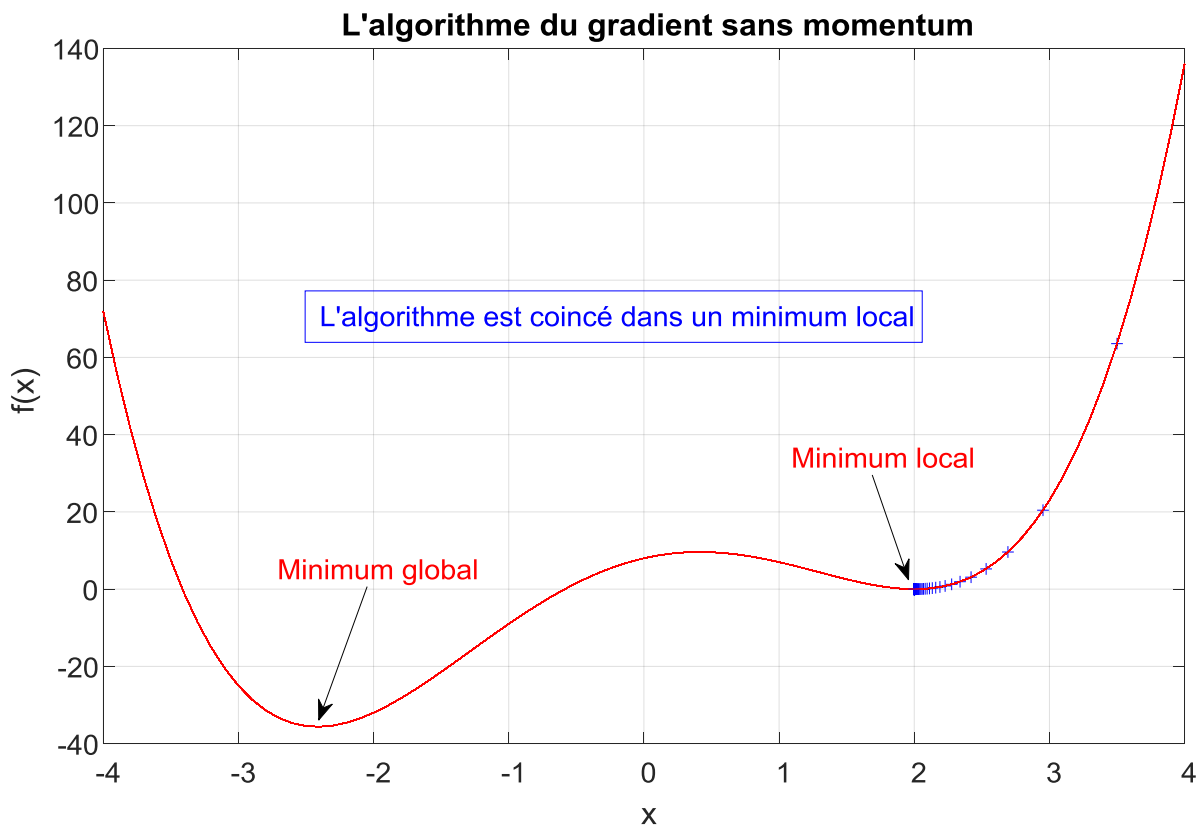


Figure III.7: Méthode du gradient sans momentum

Selon la figure ci-dessus, on remarque dès que l'algorithme s'approche du minimum local (le point $x = 2$), le gradient devient très faible et après certaines itérations, il s'y

coince (le gradient devient nul $\frac{\partial f(x)}{\partial x} = 0$). Dans ce cas, le gradient disparaît en ce point, et l'algorithme du gradient converge vers un minimum local [41-42].

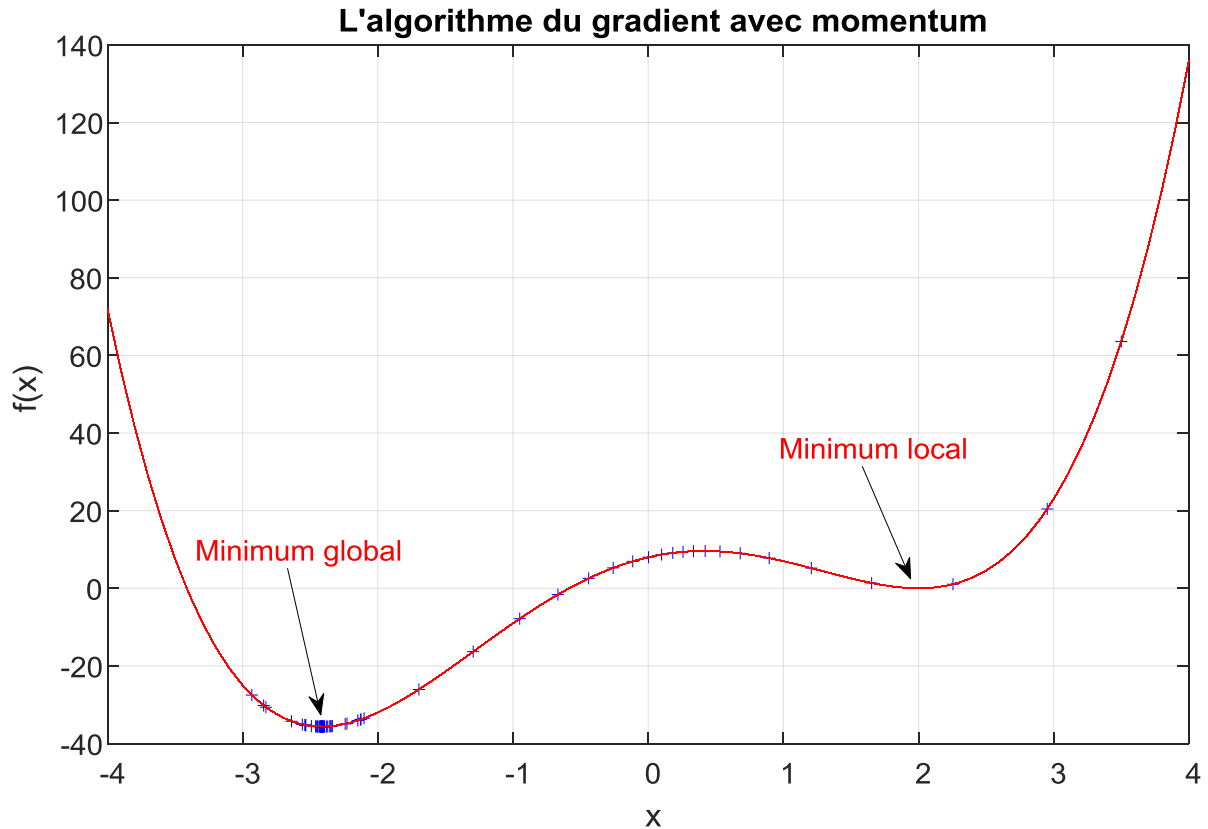


Figure III.8: Méthode du gradient avec momentum

D'après la figure ci-dessus, on remarque dès que l'algorithme s'approche du minimum local, le gradient s'annule $\frac{\partial f(x)}{\partial x} = 0$, mais, grâce aux gradients accumulés à partir des itérations passées, l'algorithme du gradient pourrait éventuellement dépasser le minimum local et convergerait vers un minimum global.

Remarque:

La méthode du gradient avec momentum n'assure pas la convergence vers un minimum global.

Exemple 4:

On veut minimiser la fonction à deux variables $f(x_1, x_2) = 0.1x_1^2 + 2x_2^2 - 1.5x_1 - 1.5x_2$ en utilisant la méthode de descente du gradient sans et avec momentum.

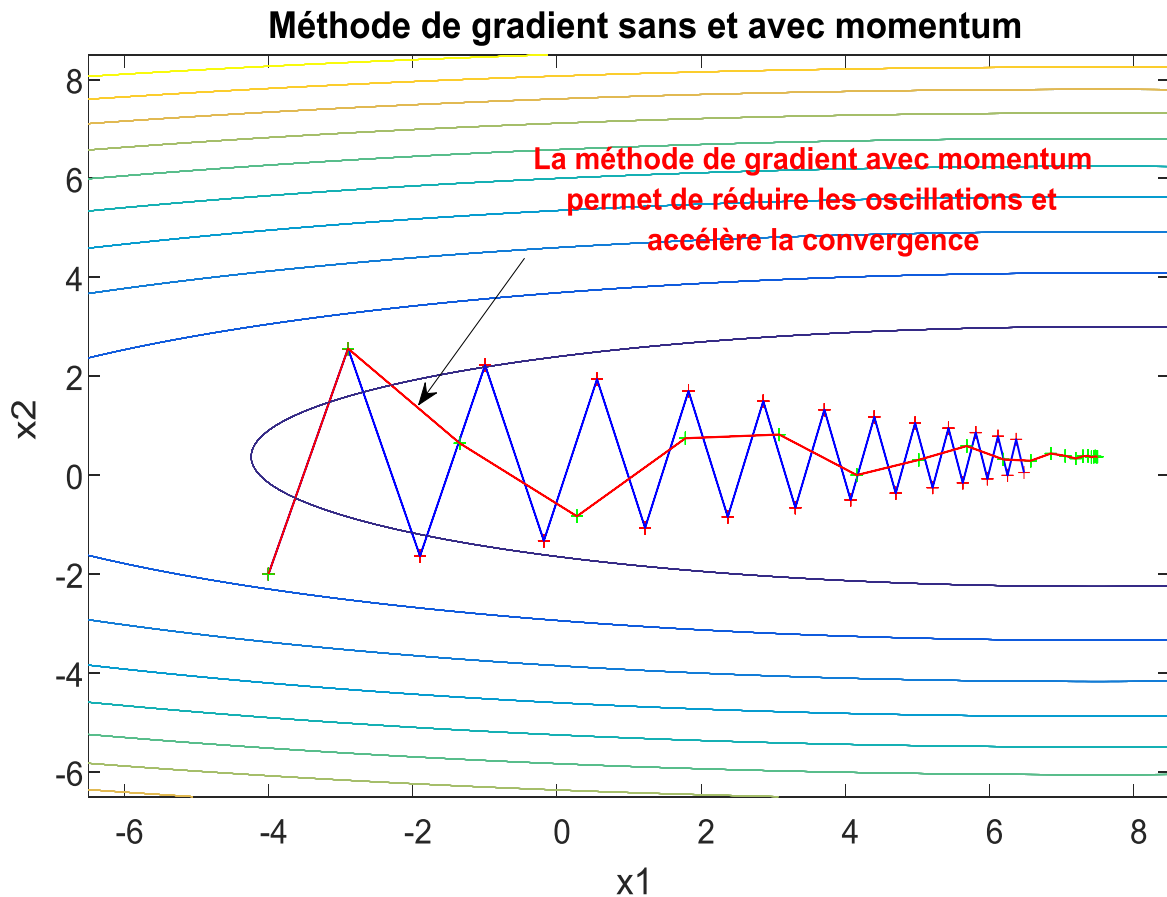


Figure III.9: Méthode du gradient sans et avec momentum

L'intérêt de la méthode du gradient avec momentum (la courbe en rouge) apparaît clairement sur la figure 9. Elle permet de réduire les oscillations et accélère la convergence, tandis que la méthode du gradient sans momentum converge lentement à cause de la présence de beaucoup d'oscillations.

III-3) Conclusion

Dans ce chapitre, trois dérivées de la méthode de descente du gradient ont été citées : la méthode du gradient avec un pas constant, la méthode du gradient adaptatif et la méthode du gradient avec momentum.

Chapitre III: La méthode de descente du gradient

La méthode du gradient standard utilise un taux d'apprentissage constant qui doit être choisi manuellement et il peut varier considérablement en fonction du problème à résoudre. Cependant, un mauvais choix du taux d'apprentissage peut conduire à des mauvais résultats: un taux d'apprentissage trop grand peut faire diverger l'algorithme tandis qu'un taux d'apprentissage trop petit ralentit la convergence de l'algorithme ou bloque l'algorithme dans un minimum local indésirable [43].

La méthode du gradient adaptatif est une amélioration de la méthode du gradient standard. Contrairement à la méthode du gradient standard qui utilise un taux d'apprentissage fixe, celle du gradient adaptatif utilise un taux d'apprentissage adaptatif afin d'améliorer la vitesse de convergence [44].

La méthode du gradient avec un momentum présente plusieurs intérêts: elle réduit les oscillations, accélère la convergence et ne resterait pas coincé dans des minima locaux. Comme suite de ce travail, on envisagera d'utiliser la méthode du gradient avec un momentum pour la commande d'un système non linéaire du traitement des eaux usées par boues activées.

Chapitre IV: Les réseaux de neurones

IV-1) Introduction

Les réseaux de neurones artificiels ont été inspirés du fonctionnement des neurones biologiques du cerveau humain. L'objectif des chercheurs était de construire une machine capable de simuler le comportement du cerveau humain.

La première tentative de la modélisation mathématique du neurone biologique en neurone formel était en 1943 par Mac Culloch et Pitts [45]. En 1958, F. Rosenblatt [46] construit le premier neuro-ordinateur basé sur un modèle d'une couche d'entrée et une couche de sortie et l'applique au domaine de la reconnaissance des formes. En 1960, Widrow [47] s'appuie sur le neurone formel de McCulloch et Pitts pour développer un réseau de neurones artificiels avec une simple couche où la différence avec McCulloch et Pitts était dans la loi d'apprentissage. Les recherches et les développements ne cessent pas de continuer, mais, les réseaux de neurones jusqu'à maintenant n'étaient pas capables de résoudre des problèmes non linéaires jusqu'à l'apparition de l'algorithme de la rétro-propagation du gradient en 1974 par Paul Werbos et mis au point plus tard, en 1986 par David Rumelhart [48]. Dès cette découverte, les réseaux de neurones étaient capables de résoudre des problèmes non linéaires avec une bonne approximation [49-50].

Grâce à leur capacité d'apprentissage, les réseaux de neurones artificiels ont des applications dans plusieurs domaines parmi lesquelles:

- * Le traitement d'image (compression d'images, reconnaissance de formes et de caractères, classification, etc.).
- * Le traitement de signal (traitement de la parole, filtrage, classification, etc.).
l'optimisation régulation du trafic, gestion, finance, etc.).
- * Le contrôle (diagnostic des défauts, commande des systèmes, robotique, etc.).
- * La modélisation (approximation d'une fonction inconnue ou modélisation d'une fonction connue mais complexe à calculer avec précision [51]).

Chapitre IV: Les réseaux de neurones

On va présenter dans ce chapitre les principes de base des réseaux de neurones artificiels: définition d'un neurone biologique et formel, les fonctions d'activations les plus utilisées, la structure d'un réseau de neurones MLP, l'algorithme d'apprentissage de rétro-propagation du gradient, la structure d'un réseau de neurones RBF et on finira par une conclusion.

IV-2) Neurone biologique

Le neurone biologique est une cellule nerveuse spécialisée dans le traitement des signaux électriques. Elle se compose en quatre parties: dendrites, synapses, noyau et des axones (figure 1).

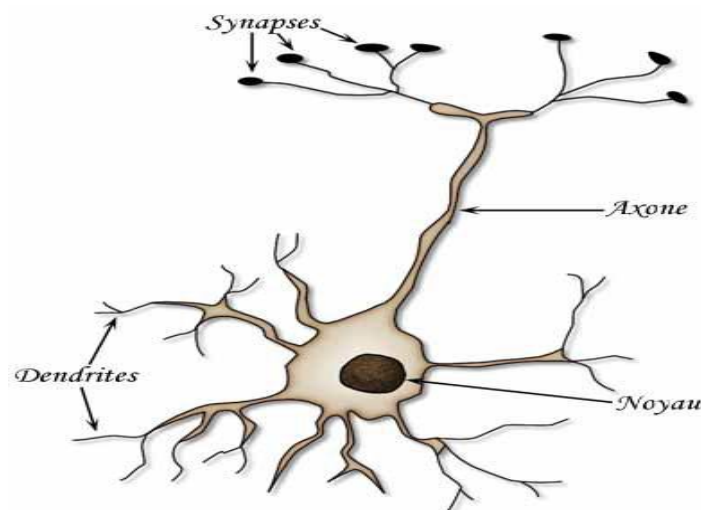


Figure IV.1: Neurone biologique

La réception des signaux électriques d'entrées (informations) se fait à travers les dendrites pour être acheminées vers le noyau du neurone qui est l'unité de traitement. Ensuite, l'information traitée par le neurone sera transmise aux autres neurones par les axones.

IV-3) Neurone formel

Un neurone formel est un processus simple qui imite la structure et le principe de fonctionnement d'un neurone biologique [45]. Chaque neurone reçoit un certain nombre d'entrées. Chacune de ces entrées est multipliée par un facteur multiplicatif (poids). Par la

Chapitre IV: Les réseaux de neurones

suite, la somme des produits entre les entrées et les poids passe dans une fonction qui s'appelle "fonction d'activation" pour donner la valeur de sortie du neurone. La figure ci-dessous montre la structure d'un neurone formel.

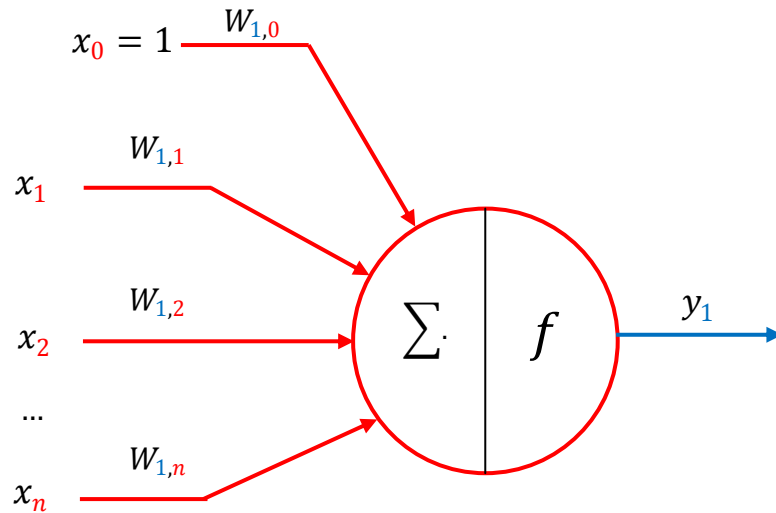


Figure IV.2: Structure d'un neurone formel

$$y_1 = f(x_1 W_{1,1} + x_2 W_{1,2} + \dots + x_n W_{1,n}) = f\left(\sum_{i=1}^n W_{1,i} x_i\right) \quad (4.1)$$

$X = [x_1, x_2, \dots, x_n]^T$: vecteur d'entrée.

$W_{1,1}, W_{1,2}, \dots, W_{1,n}$: poids.

f : représente la fonction d'activation.

y_1 : représente la sortie du neurone.

Neurone biologique	Neurone formel
Dendrite	Signal d'entrée
Synapses	Poids
Soma	Fonction d'activation
Axone	Signal de sortie

Tableau IV.1: Comparaison entre le neurone biologique et formel

IV-4) Quelque exemple des fonctions d'activations

La fonction d'activation est une fonction qui définit l'état interne du neurone en fonction de son entrée. Elle peut être une fonction linéaire, non linéaire, continue ou discontinue.

IV-4-1) La fonction d'activation binaire

La fonction binaire est une fonction linéaire dont la sortie est limitée à deux valeurs. Elle est égale à " 1 " pour les valeurs des signes positives et à " 0 " pour les valeurs des signes négatives.

$$f(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x < 0 \end{cases} \quad (4.2)$$

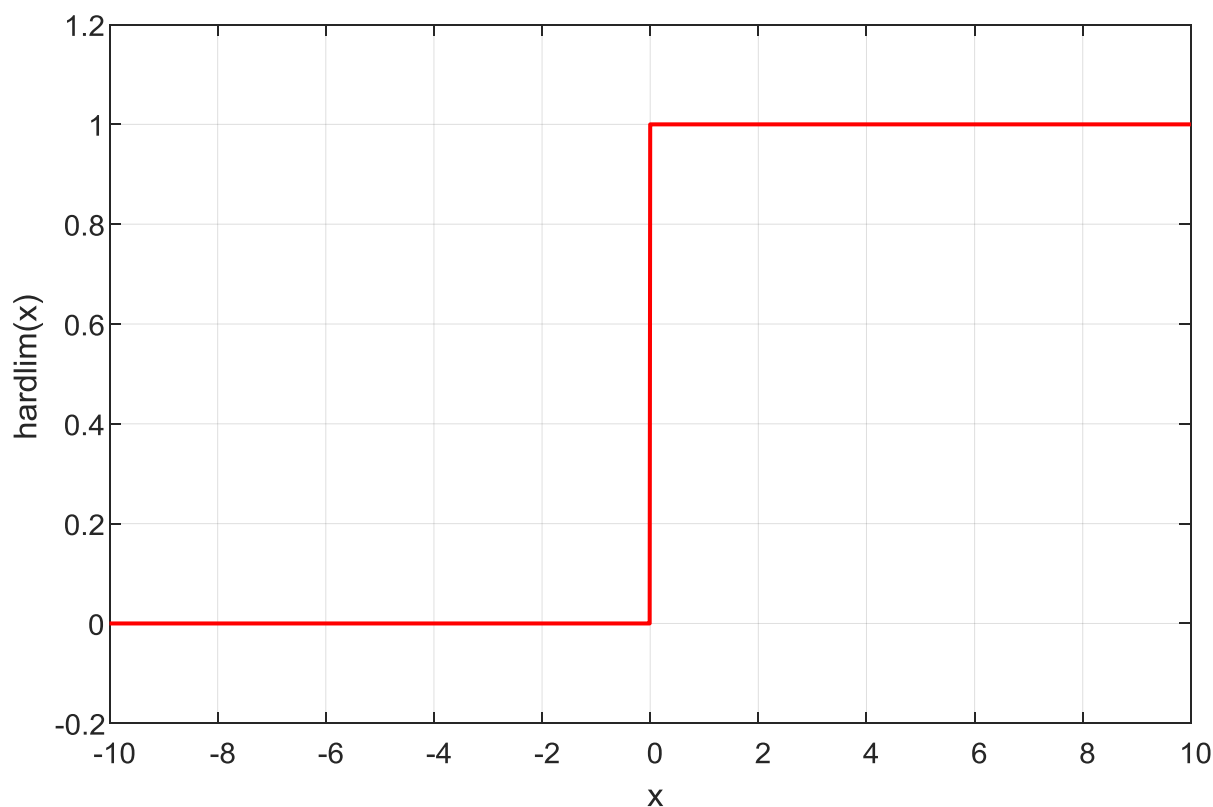


Figure IV.3: La fonction binaire

IV-4-2) La fonction d'activation signe

La fonction signe est une fonction linéaire dont la valeur de sortie est égale à " 1 " pour les valeurs des signes positives et à " -1 " pour les valeurs des signes négatives.

$$f(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{si } x < 0 \end{cases} \quad (4.3)$$

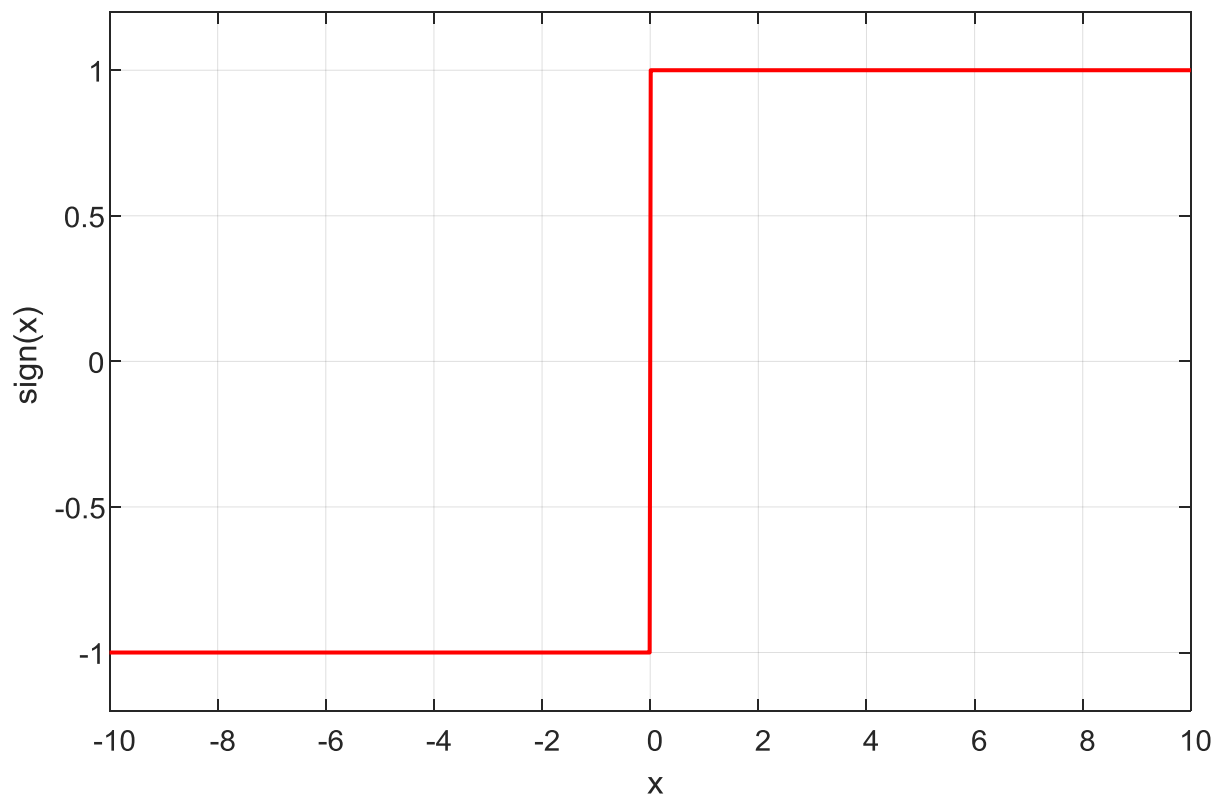


Figure IV.4: La fonction d'activation sign

IV-4-3) La fonction d'activation linéaire

La fonction d'activation linéaire est très utilisée dans la couche de sortie des réseaux de neurones. Son expression mathématique est donnée comme suit:

$$f(x) = x \quad (4.4)$$

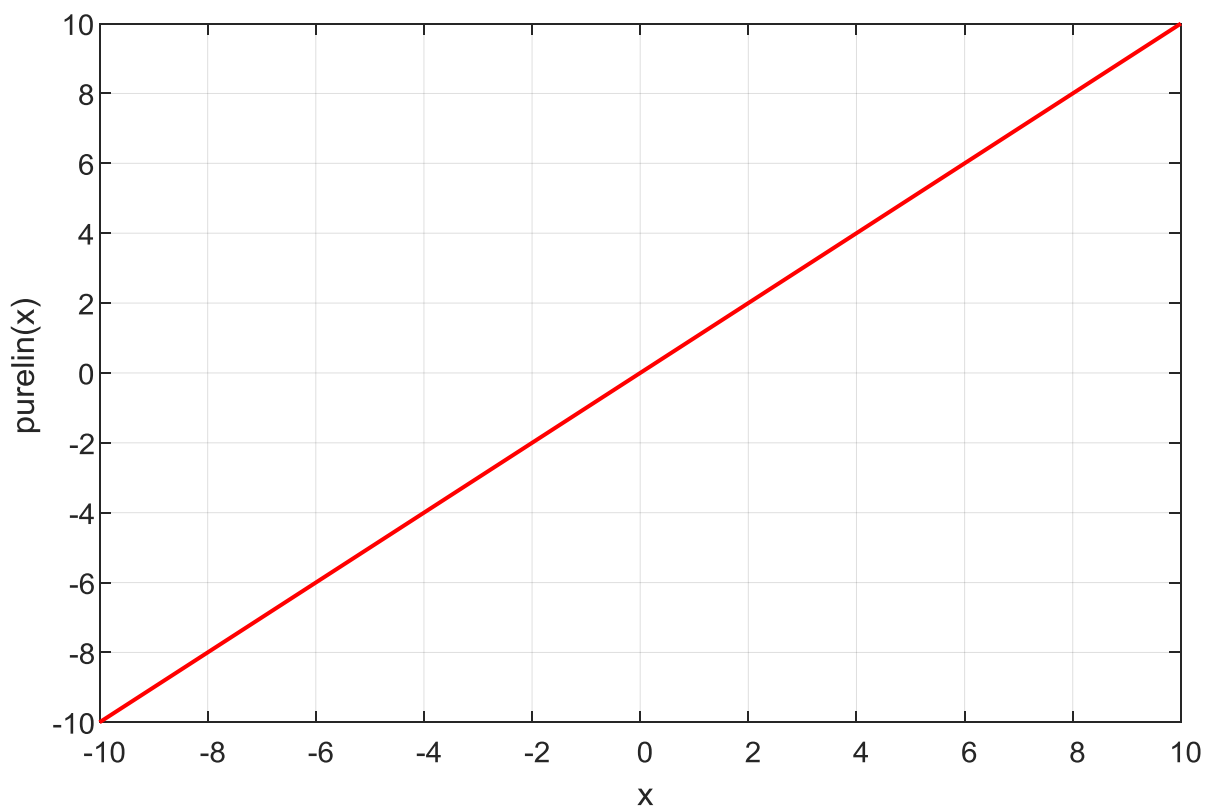


Figure IV.5: La fonction d'activation linéaire

IV-4-4) La fonction d'activation linéaire avec saturation

L'expression mathématique de la fonction d'activation linéaire avec saturation est donnée comme suit:

$$f(x) = \begin{cases} 1 & \text{si } x \geq a \\ \frac{1}{a}x & \text{si } -a < x < a \\ -1 & \text{si } x \leq 0 \end{cases} \quad (4.5)$$

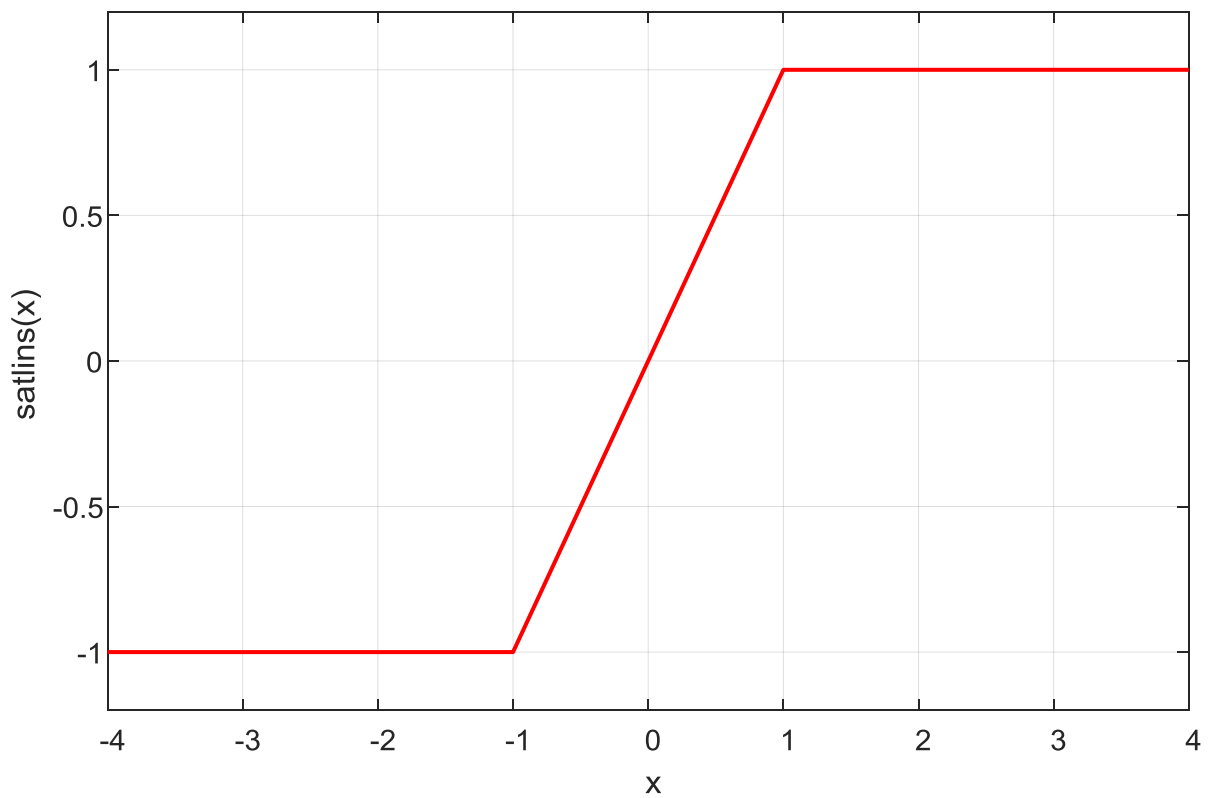


Figure IV.6: La fonction d'activation linéaire avec saturation

IV-4-5) La fonction d'activation sigmoïde

La fonction sigmoïde est une fonction non linéaire dont la valeur varie entre "1" et "0". Son modèle mathématique s'écrit comme:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4.6)$$

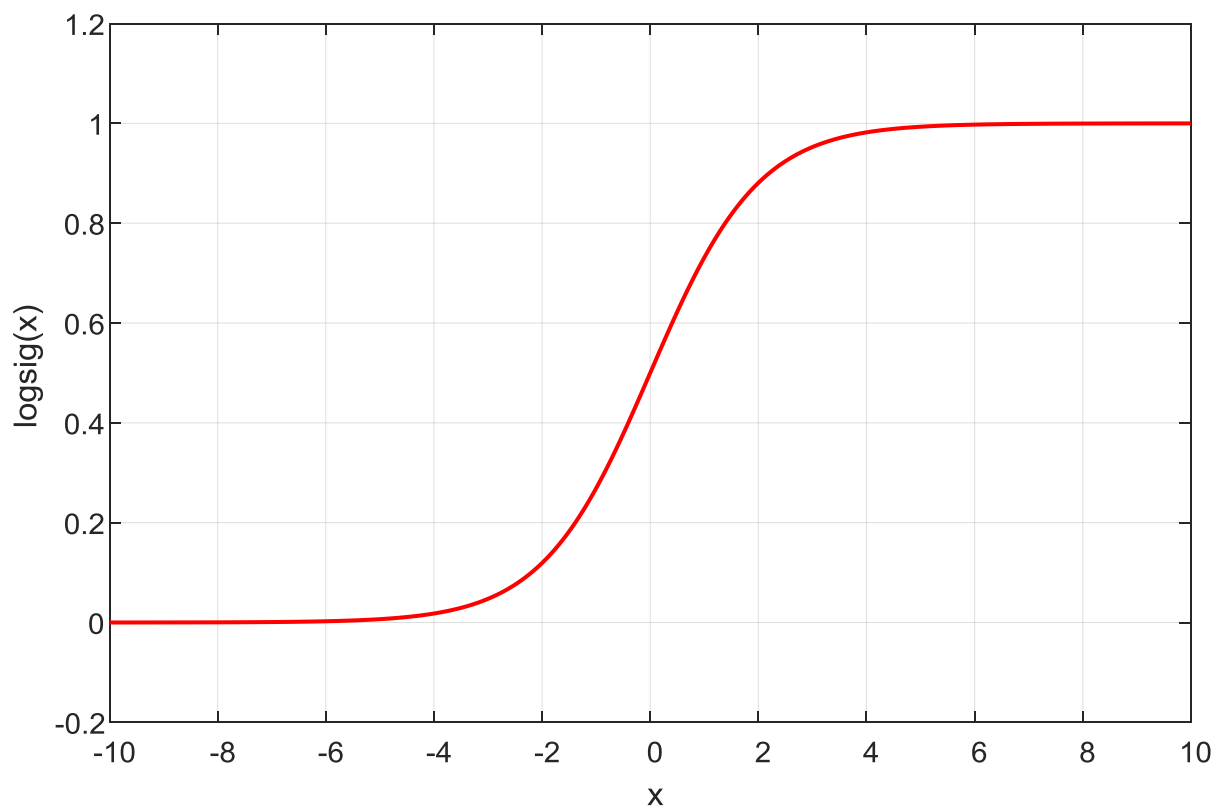


Figure IV.7: La fonction d'activation sigmoïde

IV-4-6) La fonction d'activation tangente hyperbolique

La fonction tangente hyperbolique est aussi une fonction non linéaire dont la valeur varie entre "+1" et "-1". Son expression mathématique est donnée par:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.7)$$

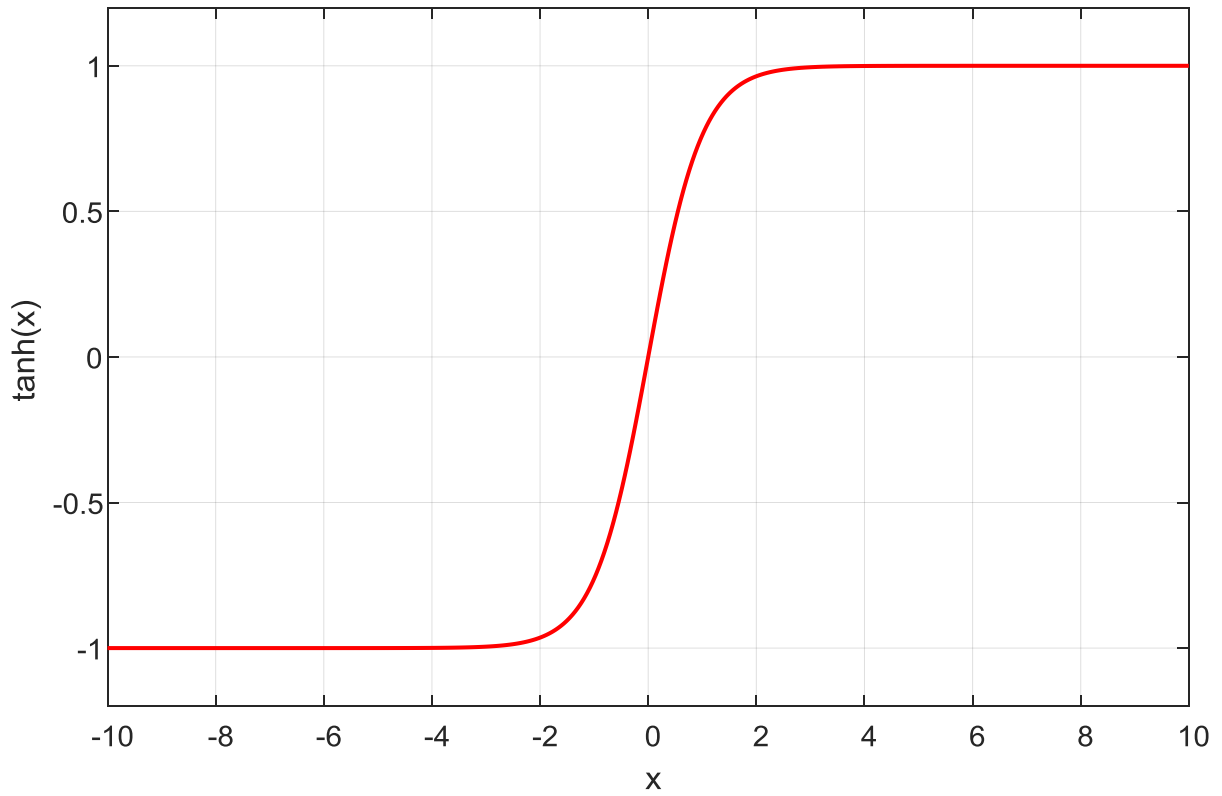


Figure IV.8: La fonction d'activation tangente hyperbolique

* *Dérivée de la fonction sigmoïde et tangente hyperbolique:*

On cherche à exprimer la dérivée de la fonction d'activation sigmoïde ainsi que la fonction tangente hyperbolique en fonction de leurs fonctions elles-mêmes.

$$f(x) = \frac{1}{1 + e^{-ax}} \Rightarrow f'(x) = \frac{ae^{-ax}}{(1 + e^{-ax})^2} = a \left(\frac{e^{-ax}}{1 + e^{-ax}} \right) \frac{1}{1 + e^{-ax}} \quad (4.8)$$

$$f'(x) = a \left(1 - \frac{1}{1 + e^{-ax}} \right) \frac{1}{1 + e^{-ax}} = a(1 - f(x))f(x) \quad (4.9)$$

Donc, la dérivée de la fonction sigmoïde peut être donnée par la relation suivante:

$$f' = a(1 - f)f \quad (4.10)$$

$$f(x) = \frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}} \Rightarrow f'(x) = a \frac{(e^{ax} + e^{-ax})^2 - (e^{ax} - e^{-ax})^2}{(e^{ax} + e^{-ax})^2} \quad (4.11)$$

$$f'(x) = a \left(1 - \frac{(e^{ax} - e^{-ax})^2}{(e^{ax} + e^{-ax})^2} \right) = a(1 - f^2(x)) \quad (4.12)$$

La dérivée de la fonction tangente hyperbolique peut être donnée par la relation suivante:

$$f' = a(1 - f^2) \quad (4.13)$$

Le choix de la fonction d'activation dépend de l'application. S'il faut avoir des sorties supérieures à " 1 ", c'est la fonction linéaire que l'on choisit habituellement.

IV-5) Structure des réseaux de neurones artificiels MLP (perceptron multicouche)

La structure d'un réseau MLP (Multi Layer Perceptron) ou le perceptron multicouche est un réseau de neurones artificiels qui se compose: d'une couche d'entrée, une couche de sortie et une ou plusieurs couches intermédiaires appelées couches cachées. La figure 9 donne l'exemple d'un réseau de neurones contenant n entrées dans la couche d'entrée, m neurones dans la couches cachées et r neurones dans la couche de sortie. Le nombre de couches cachées et le nombre de neurones dans chaque couche dépend de la complexité du problème résolu par le réseau.

Cette structure est couramment utilisée dans les applications d'identification des systèmes et d'approximation des fonctions non linéaires.

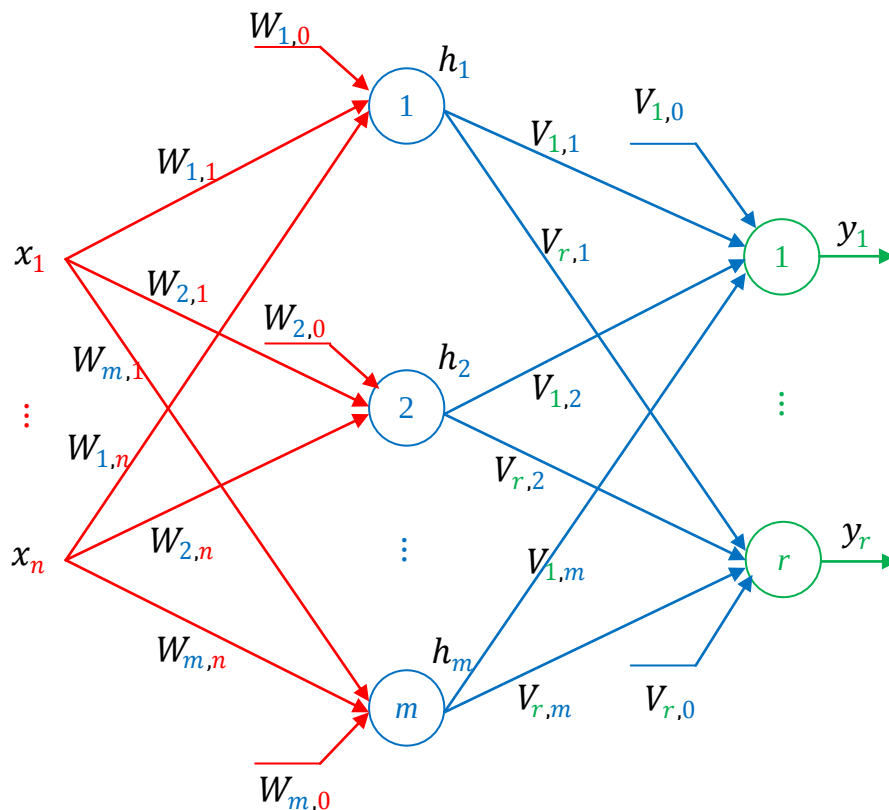


Figure IV.9: Architecture d'un réseau neuronal multicouche

Chapitre IV: Les réseaux de neurones

L'expression mathématique de la sortie du premier neurone dans la couche cachée est donnée par:

$$h_1 = f(W_{1,1}x_1 + W_{1,2}x_2 + \dots + W_{1,n}x_n + W_{1,0}) = f\left(\sum_{i=0}^n W_{1,i}x_i\right) \quad (4.14)$$

L'expression mathématique de la sortie du deuxième neurone dans la couche cachée est donnée par:

$$h_2 = f(W_{2,1}x_1 + W_{2,2}x_2 + \dots + W_{2,n}x_n + W_{2,0}) = f\left(\sum_{i=0}^n W_{2,i}x_i\right) \quad (4.15)$$

L'expression mathématique de la sortie du $j^{\text{ème}}$ neurone dans la couche cachée est donnée par:

$$h_m = f(W_{m,1}x_1 + W_{m,2}x_2 + \dots + W_{m,n}x_n + W_{m,0}) = f\left(\sum_{i=0}^n W_{m,i}x_i\right) \quad (4.16)$$

Le vecteur de sortie de la couche cachée peut être décrit comme suit:

$$h_j = f\left(\sum_{i=0}^n W_{j,i}x_i\right) \quad (4.17)$$

L'expression de la sortie du premier neurone dans la couche de sortie est donnée par:

$$y_1 = g(V_{1,1}h_1 + V_{1,2}h_2 + \dots + V_{1,m}h_m + V_{1,0}) = g\left(\sum_{j=0}^m V_{1,j}h_j\right) \quad (4.18)$$

L'expression de la sortie du deuxième neurone dans la couche de sortie est donnée par:

$$y_2 = g(V_{2,1}h_1 + V_{2,2}h_2 + \dots + V_{2,m}h_m + V_{2,0}) = g\left(\sum_{j=0}^m V_{2,j}h_j\right) \quad (4.19)$$

L'expression de la sortie du $k^{\text{ème}}$ neurone dans la couche de sortie est donnée par:

$$y_r = g(V_{r,1}h_1 + V_{r,2}h_2 + \dots + V_{r,m}h_m + V_{r,0}) = g\left(\sum_{j=0}^m V_{r,j}h_j\right) \quad (4.20)$$

Le vecteur de sortie du réseau de neurones:

$$y_k = g \left(\sum_{j=0}^m V_{k,j} h_j \right) = g \left(\sum_{j=0}^m V_{k,j} f \left(\sum_{i=0}^n W_{j,i} x_i \right) \right) \quad (4.21)$$

f : représente la fonction d'activation de la couche cachée.

g : représente la fonction d'activation de la couche de sortie.

Exemple 1:

Soit le réseau de neurones montré dans la figure 10, qui possède : deux entrée x_1 et x_2 , trois neurones dans la couche cachée et un neurone dans la couche de sortie.

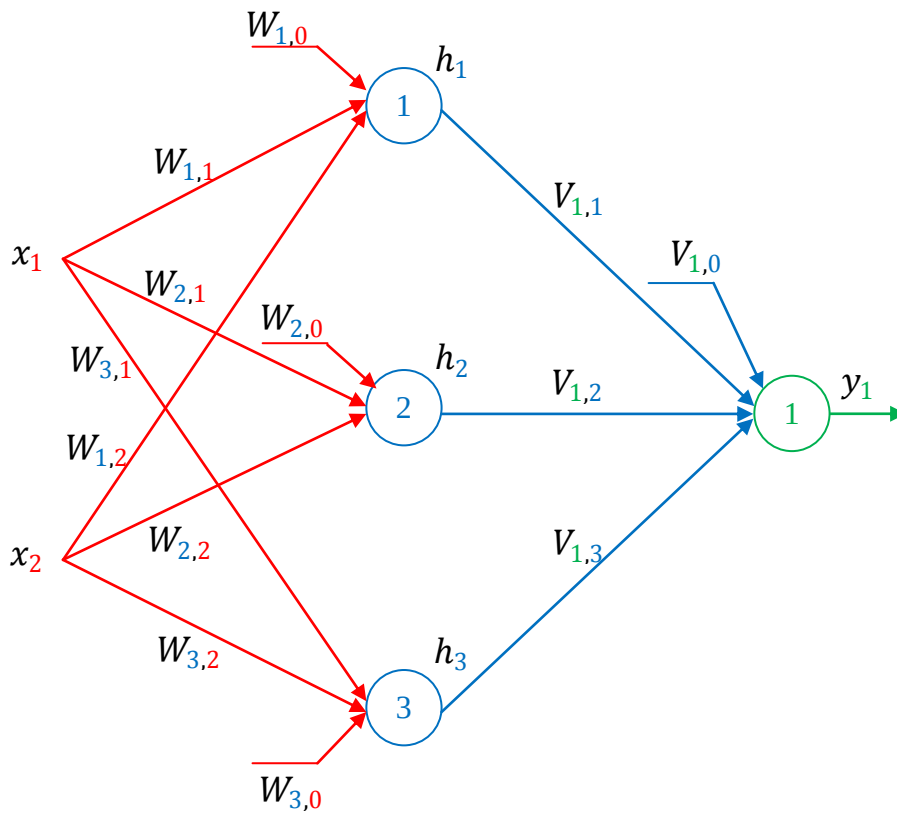


Figure IV.10: Structure d'un réseau de neurones avec trois neurones dans la couche cachée et un neurone dans la couche de sortie

On donne : $x_1 = 0.2, x_2 = 0.6$

La matrice des poids ainsi que les biais entre la couche d'entrée et la couche cachée sont:

$$W = \begin{pmatrix} W_{1,1} & W_{1,2} \\ W_{2,1} & W_{2,2} \\ W_{3,1} & W_{3,2} \end{pmatrix} = \begin{pmatrix} 1 & 1.5 \\ 0.5 & 2 \\ 2.5 & 3 \end{pmatrix}, \begin{pmatrix} W_{1,0} \\ W_{1,0} \\ W_{1,0} \end{pmatrix} = \begin{pmatrix} 1 \\ -0.5 \\ 1.5 \end{pmatrix}$$

Chapitre IV: Les réseaux de neurones

La matrice des poids ainsi que les biais entre la couche cachée et la couche de sortie sont :

$$V = (V_{1,1} \quad V_{1,2} \quad V_{1,3}) = (3 \quad 2 \quad 1), V_{1,0} = -4$$

La fonction d'activation pour la couche cachée et la couche de sortie est la fonction sigmoïde.

La sortie du premier neurone de la couche cachée:

$$h_1 = f(W_{1,1}x_1 + W_{1,2}x_2 + W_{1,0}) = f(1 * 0.2 + 1.5 * 0.6 + 1) = f(2.1) = 0.8909$$

La sortie du deuxième neurone de la couche cachée:

$$h_2 = f(W_{2,1}x_1 + W_{2,2}x_2 + W_{2,0}) = f(0.5 * 0.2 + 2 * 0.6 - 0.5) = f(0.8) = 0.6900$$

La sortie du troisième neurone de la couche cachée:

$$h_3 = f(W_{3,1}x_1 + W_{3,2}x_2 + W_{3,0}) = f(2.5 * 0.2 + 3 * 0.6 + 1.5) = f(3.8) = 0.9781$$

Le vecteur de sortie de la couche cachée:

$$\begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = \begin{pmatrix} 0.8909 \\ 0.6900 \\ 0.9781 \end{pmatrix}$$

La sortie du réseau de neurones:

$$y_1 = g(V_{1,1}h_1 + V_{1,2}h_2 + V_{1,3}h_3 + V_{1,0})$$

$$y_1 = g(3 * 0.8909 + 2 * 0.6900 + 1 * 0.9781 - 4) = g(1.0308) = 0.7371$$

IV-6) Apprentissage

L'apprentissage est la propriété la plus importante des réseaux neurones. C'est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention d'un comportement désiré. L'apprentissage d'un réseau de neurones signifie qu'il change son comportement de façon à lui permettre de se rapprocher d'un but défini.

IV-7) Algorithme de rétro-propagation

Le principe utilisé par la rétro-propagation du gradient est la minimisation d'une fonction dépendante de l'erreur. La méthode de la rétro-propagation permet de faire propager l'erreur à travers les couches.

On calcule d'abord l'erreur commise par chaque neurone. Pour les neurones de la couche de sortie, l'erreur est la différence entre la sortie désirée et la sortie du réseau de neurones, pour les neurones de la couche cachée, l'erreur n'est pas directement calculable alors on considère la somme pondérée des erreurs de la couche suivante.

La fonction à minimiser est la somme de l'erreur carrée entre la sortie désirée et la sortie du réseau de neurones, son expression s'écrit comme suit:

$$E(k) = \frac{1}{2} \sum_{k=1}^{\infty} (e_k(k))^2 = \frac{1}{2} \sum_{k=1}^{\infty} (y_{d_k}(k) - y_{nn_k}(k))^2 \quad (4.22)$$

Avec y_{d_k} et y_{nn_k} sont respectivement la sortie désirée et la sortie du réseau de neurones.

L'algorithme du gradient s'écrit par la relation suivante:

$$X(k+1) = X(k) - \eta \frac{\partial E(k)}{\partial X(k)} \quad (4.23)$$

Avec η est le taux d'apprentissage.

* La mise à jour des poids entre la couche cachée et la couche de sortie:

$$V_{k,j}(k+1) = V_{k,j}(k) - \eta \frac{\partial E(k)}{\partial V_{k,j}(k)} = V_{k,j}(k) - \eta \frac{\partial E(k)}{\partial y_{nn_k}(k)} \frac{\partial y_{nn_k}(k)}{\partial V_{k,j}(k)} \quad (4.24)$$

$$V_{k,j}(k+1) = V_{k,j}(k) + \eta e_k \frac{\partial y_{nn_k}(k)}{\partial V_{k,j}(k)} \quad (4.25)$$

La dérivée de l'expression $\frac{\partial y_{nn_k}(k)}{\partial V_{k,j}(k)}$ dépend de la fonction d'activation utilisée pour la couche de sortie.

Le vecteur de sortie du réseau de neurones:

$$y_{nn_k} = g \left(\sum_{j=0}^m V_{k,j} h_j \right) \quad (4.26)$$

On va se limiter par les fonctions d'activations suivantes: fonction linéaire, fonction sigmoïde et la fonction tangente hyperbolique.

* g : est une fonction linéaire.

$$y_{nn_k} = g \left(\sum_{j=0}^m V_{k,j} h_j \right) = \sum_{j=0}^m V_{k,j} h_j \quad (4.27)$$

$$\frac{\partial y_{nn_k}(k)}{\partial V_{k,j}(k)} = h_j \quad (4.28)$$

* g : est une fonction sigmoïde.

$$y_{nn_k} = g \left(\sum_{j=0}^m V_{k,j} h_j \right) = \frac{1}{1 + e^{-(\sum_{j=0}^m V_{k,j} h_j)}} \quad (4.29)$$

$$\frac{\partial y_{nn_k}(k)}{\partial V_{k,j}(k)} = (1 - y_{nn_k}) y_{nn_k} h_j \quad (4.30)$$

* g : est une fonction tangente hyperbolique.

$$y_{nn_k} = g \left(\sum_{j=0}^m V_{k,j} h_j \right) = \frac{e^{(\sum_{j=0}^m V_{k,j} h_j)} - e^{-(\sum_{j=0}^m V_{k,j} h_j)}}{e^{(\sum_{j=0}^m V_{k,j} h_j)} + e^{-(\sum_{j=0}^m V_{k,j} h_j)}} \quad (4.31)$$

$$\frac{\partial y_{nn_k}(k)}{\partial V_{k,j}(k)} = (1 - y_{nn_k}^2) h_j \quad (4.32)$$

Finalement, on pourrait écrire:

$$V_{k,j}(k+1) = V_{k,j}(k) + \eta \delta_k h_j \quad (4.33)$$

Avec:

$$\delta_k = \begin{cases} e_k (1 - y_{nn_k}) y_{nn_k} \\ e_k (1 - y_{nn_k}^2) \end{cases} \quad (4.34)$$

* La mise à jour des poids entre la couche d'entrée et la couche cachée:

$$W_{j,i}(k+1) = W_{j,i}(k) - \eta \frac{\partial E(k)}{\partial W_{j,i}(k)} = W_{j,i}(k) - \eta \frac{\partial E(k)}{\partial y_{nn_k}(k)} \frac{\partial y_{nn_k}(k)}{\partial h_j} \frac{\partial h_j}{\partial W_{j,i}(k)} \quad (4.35)$$

$$W_{j,i}(k+1) = W_{j,i}(k) + \eta e_k \frac{\partial y_{nn_k}(k)}{\partial h_j} \frac{\partial h_j}{\partial W_{j,i}(k)} \quad (4.36)$$

La dérivée de l'expression $\frac{\partial y_{nn_k}(k)}{\partial h_j}$ dépend de la fonction d'activation utilisée pour la couche de sortie.

Le vecteur de sortie du réseau de neurones:

$$y_{nn_k} = g \left(\sum_{j=0}^m V_{k,j} h_j \right) \quad (4.37)$$

* g : est une fonction linéaire.

$$y_{nn_k} = g \left(\sum_{j=0}^m V_{k,j} h_j \right) = \sum_{j=0}^m V_{k,j} h_j \quad (4.38)$$

$$\frac{\partial y_{nn_k}(k)}{\partial h_j} = V_{k,j} \quad (4.39)$$

* g : est une fonction sigmoïde.

$$y_{nn_k} = g \left(\sum_{j=0}^m V_{k,j} h_j \right) = \frac{1}{1 + e^{-(\sum_{j=0}^m V_{k,j} h_j)}} \quad (4.40)$$

$$\frac{\partial y_{nn_k}(k)}{\partial h_j} = (1 - y_{nn_k}) y_{nn_k} V_{k,j} \quad (4.41)$$

* g : est une fonction tangente hyperbolique.

$$y_{nn_k} = g \left(\sum_{j=0}^m V_{k,j} h_j \right) = \frac{e^{(\sum_{j=0}^m V_{k,j} h_j)} - e^{-(\sum_{j=0}^m V_{k,j} h_j)}}{e^{(\sum_{j=0}^m V_{k,j} h_j)} + e^{-(\sum_{j=0}^m V_{k,j} h_j)}} \quad (4.42)$$

$$\frac{\partial y_{nn_k}(k)}{\partial h_j} = (1 - y_{nn_k}^2) V_{k,j} \quad (4.43)$$

La dérivée de l'expression $\frac{\partial h_j}{\partial W_{j,i}(k)}$ dépend de la fonction d'activation utilisée pour la couche cachée.

Le vecteur de sortie de la couche cachée:

$$h_j = f \left(\sum_{i=0}^n w_{j,i} x_i \right) \quad (4.44)$$

* f : est une fonction linéaire.

$$h_j = f \left(\sum_{i=0}^n w_{j,i} x_i \right) = \sum_{i=0}^n w_{j,i} x_i \quad (4.45)$$

$$\frac{\partial h_j}{\partial w_{j,i}(k)} = x_i \quad (4.46)$$

* f : est une fonction sigmoïde.

$$h_j = f \left(\sum_{i=0}^n w_{j,i} x_i \right) = \frac{1}{1 + e^{-(\sum_{i=0}^n w_{j,i} x_i)}} \quad (4.47)$$

$$\frac{\partial h_j}{\partial w_{j,i}(k)} = (1 - h_j) h_j x_i \quad (4.48)$$

* f : est une fonction tangente hyperbolique.

$$h_j = f \left(\sum_{i=0}^n w_{j,i} x_i \right) = \frac{e^{(\sum_{i=0}^n w_{j,i} x_i)} - e^{-(\sum_{i=0}^n w_{j,i} x_i)}}{e^{(\sum_{i=0}^n w_{j,i} x_i)} + e^{-(\sum_{i=0}^n w_{j,i} x_i)}} \quad (4.49)$$

$$\frac{\partial h_j}{\partial w_{j,i}(k)} = (1 - h_j^2) x_i \quad (4.50)$$

Finalement, on pourrait écrire:

$$w_{j,i}(k+1) = w_{j,i}(k) + \eta \delta_j x_i \quad (4.51)$$

Avec:

$$\delta_j = \begin{cases} \sum_{k=1}^r \delta_k v_{k,j} \\ (1 - h_j) h_j \sum_{k=1}^r \delta_k v_{k,j} \\ (1 - h_j^2) \sum_{k=1}^r \delta_k v_{k,j} \end{cases} \quad (4.52)$$

Exemple 2:

Soit le réseau de neurones montré dans la figure 12, qui possède : deux entrée x_1 et x_2 , trois neurones dans la couche cachée et deux neurones dans la couche de sortie.

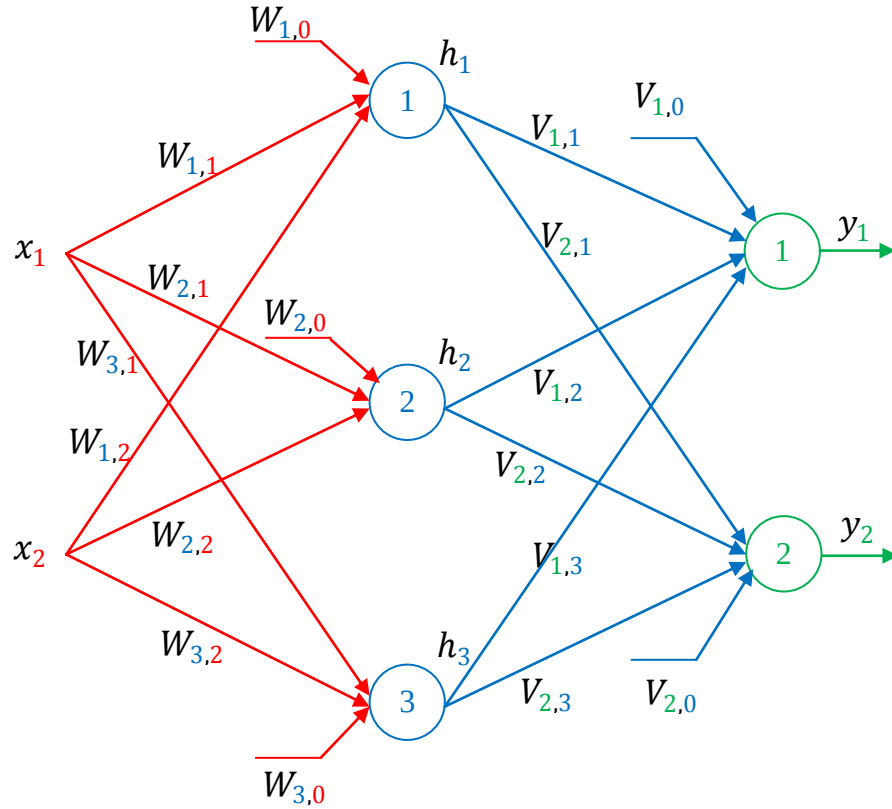


Figure IV.11: Structure d'un réseau de neurones avec trois neurones dans la couche cachée et deux neurones dans la couche de sortie

On donne : $x_1 = 0.3, x_2 = 0.5$

La matrice des poids ainsi que les biais entre la couche d'entrée et la couche cachée sont:

$$W = \begin{pmatrix} W_{1,1} & W_{1,2} \\ W_{2,1} & W_{2,2} \\ W_{3,1} & W_{3,2} \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 3 \\ 2 & 1 \end{pmatrix}, \begin{pmatrix} W_{1,0} \\ W_{2,0} \\ W_{3,0} \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \\ 1 \end{pmatrix}$$

La matrice des poids ainsi que les biais entre la couche cachée et la couche de sortie sont:

$$V = \begin{pmatrix} V_{1,1} & V_{1,2} & V_{1,3} \\ V_{2,1} & V_{2,2} & V_{2,3} \end{pmatrix} = \begin{pmatrix} -3 & -2 & 1 \\ 1 & 2 & 3 \end{pmatrix}, \begin{pmatrix} V_{1,0} \\ V_{2,0} \end{pmatrix} = \begin{pmatrix} 3 \\ -3 \end{pmatrix}$$

Les fonctions d'activation pour la couche cachée et la couche de sortie sont respectivement: la fonction sigmoïde et la fonction linéaire.

On veut que les sorties du réseau de neurones y_1 et y_2 soient égales aux sorties désirées $y_{d1} = 2$ et $y_{d2} = 2$. On donne le taux d'apprentissage $\eta = 0.5$.

Chapitre IV: Les réseaux de neurones

Les sorties du réseau de neurones y_1 et y_2 :

La sortie du premier neurone de la couche cachée:

$$h_1 = f(W_{1,1}x_1 + W_{1,2}x_2 + W_{1,0}) = f(1 * 0.3 + 2 * 0.5 + 2) = f(3.3000) = 0.9644$$

La sortie du deuxième neurone de la couche cachée:

$$h_2 = f(W_{2,1}x_1 + W_{2,2}x_2 + W_{2,0}) = f(3 * 0.3 + 3 * 0.5 - 2) = f(0.4000) = 0.5987$$

La sortie du troisième neurone de la couche cachée:

$$h_3 = f(W_{3,1}x_1 + W_{3,2}x_2 + W_{3,0}) = f(2 * 0.3 + 1 * 0.5 + 1) = f(2.1000) = 0.8909$$

Le vecteur de sortie de la couche cachée:

$$\begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = \begin{pmatrix} 0.9644 \\ 0.5987 \\ 0.8909 \end{pmatrix}$$

La sortie du premier neurone de la couche de sortie:

$$y_1 = g(V_{1,1}h_1 + V_{1,2}h_2 + V_{1,3}h_3 + V_{1,0})$$

$$y_1 = g(-3 * 0.9644 - 2 * 0.5987 + 1 * 0.8909 + 3) = g(-0.1998) = -0.1998$$

La sortie du deuxième neurone de la couche de sortie:

$$y_2 = g(V_{2,1}h_1 + V_{2,2}h_2 + V_{2,3}h_3 + V_{2,0})$$

$$y_2 = g(1 * 0.9644 + 2 * 0.5987 + 3 * 0.8909 - 3) = g(1.8345) = 1.8345$$

Le vecteur de sortie de la couche cachée:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} -0.1998 \\ 1.8345 \end{pmatrix}$$

Les nouvelles valeurs des poids et des biais:

* La mise à jour des poids entre la couche cachée et la couche de sortie:

$$V_{k,j}(k+1) = V_{k,j}(k) + \eta \delta_k h_j$$

$$\text{Avec: } \delta_k = e_k \Rightarrow \begin{cases} \delta_1 = e_1 = (y_{d1} - y_1) \\ \delta_2 = e_2 = (y_{d2} - y_2) \end{cases}$$

$$* \delta_1 = e_1 = (y_{d1} - y_1) = (2 - (-0.1998)) = 2.1998$$

$$* \delta_2 = e_2 = (y_{d2} - y_2) = (2 - 1.8345) = 0.1655$$

$$V_{1,1} = V_{1,1} + \eta \delta_1 h_1 = -3 + 0.5 * 2.1998 * 0.9644 = -1.9392$$

$$V_{1,2} = V_{1,2} + \eta \delta_1 h_2 = -2 + 0.5 * 2.1998 * 0.5987 = -1.3415$$

$$V_{1,3} = V_{1,3} + \eta \delta_1 h_3 = 1 + 0.5 * 2.1998 * 0.8909 = 1.9799$$

$$V_{1,0} = V_{1,0} + \eta \delta_1 = 1 + 0.5 * 2.1998 = 4.0999$$

$$V_{2,1} = V_{1,1} + \eta \delta_2 h_1 = 1 + 0.5 * 0.1655 * 0.9644 = 1.0798$$

$$V_{2,2} = V_{1,2} + \eta \delta_2 h_2 = 2 + 0.5 * 0.1655 * 0.5987 = 2.0495$$

$$V_{2,3} = V_{1,3} + \eta \delta_2 h_3 = 3 + 0.5 * 0.1655 * 0.8909 = 3.0737$$

$$V_{2,0} = V_{2,0} + \eta \delta_2 = 1 + 0.5 * 0.1655 = -2.9173$$

Donc, les nouveaux poids entre la couche cachée et la couche de sortie sont:

$$V = \begin{pmatrix} -1.9392 & -1.3415 & 1.9799 \\ 1.0798 & 2.0495 & 3.0737 \end{pmatrix}, \begin{pmatrix} V_{1,0} \\ V_{2,0} \end{pmatrix} = \begin{pmatrix} 4.0999 \\ -2.9173 \end{pmatrix}$$

* La mise à jour des poids entre la couche d'entrée et la couche cachée:

$$W_{j,i}(k+1) = W_{j,i}(k) + \eta \delta_j x_i$$

$$\delta_j = (1 - h_j) h_j \sum_{k=1}^2 \delta_k V_{kj} = (1 - h_j) h_j (\delta_1 V_{1,j} + \delta_2 V_{2,j})$$

$$* \delta_1 = (1 - h_1) h_1 (\delta_1 V_{1,1} + \delta_2 V_{2,1})$$

$$\delta_1 = (1 - 0.9644) 0.9644 (2.1998 * (-1.9392) + 0.1655 * 1.0798) = -0.1402$$

$$* \delta_2 = (1 - h_2) h_2 (\delta_1 V_{1,2} + \delta_2 V_{2,2})$$

$$\delta_2 = (1 - 0.5987) 0.5987 (2.1998 * (-1.3415) + 0.1655 * 2.0495) = -0.6275$$

$$* \delta_3 = (1 - h_3) h_3 (\delta_1 V_{1,3} + \delta_2 V_{2,3})$$

$$\delta_3 = (1 - 0.8909) 0.8909 (2.1998 * 1.9799 + 0.1655 * 3.0737) = 0.4727$$

$$W_{1,1} = W_{1,1} + \eta \delta_1 x_1 = 1 + 0.5 * (-0.1402) * 0.3 = 0.9790$$

$$W_{1,2} = W_{1,2} + \eta \delta_1 x_2 = 2 + 0.5 * (-0.1402) * 0.5 = 1.9649$$

$$W_{1,0} = W_{1,0} + \eta \delta_1 = 2 + 0.5 * (-0.1402) = 1.9299$$

$$W_{2,1} = W_{2,1} + \eta \delta_2 x_1 = 3 + 0.5 * (-0.6275) * 0.3 = 2.9059$$

$$W_{2,2} = W_{2,2} + \eta \delta_2 x_2 = 3 + 0.5 * (-0.6275) * 0.5 = 2.8431$$

$$W_{2,0} = W_{2,0} + \eta \delta_2 = -2 + 0.5 * (-0.6275) = -2.3138$$

$$W_{3,1} = W_{3,1} + \eta \delta_3 x_1 = 2 + 0.5 * 0.4727 * 0.3 = 2.0709$$

$$W_{3,2} = W_{3,2} + \eta \delta_3 x_2 = 1 + 0.5 * 0.4727 * 0.5 = 1.1182$$

$$W_{3,0} = W_{3,0} + \eta \delta_3 = 1 + 0.5 * 0.4727 = 1.2364$$

Donc, les nouveaux poids entre la couche d'entrée et la couche cachée sont:

$$W = \begin{pmatrix} 0.9790 & 1.9649 \\ 2.9059 & 2.8431 \\ 2.0709 & 1.1182 \end{pmatrix}, \begin{pmatrix} W_{1,0} \\ W_{1,0} \\ W_{1,0} \end{pmatrix} = \begin{pmatrix} 1.9299 \\ -2.3138 \\ 1.2364 \end{pmatrix}$$

Remarque:

Le réseau de neurones ne converge vers les sorties désirées $y_{d1} = 2$ et $y_{d2} = 2$ qu'après 48 itérations.

IV-8) Le réseau de neurones RBF

Un réseau de neurones RBF (Radial Base Function) est introduit dans le réseau de neurones par Lowe et Broomhead. Il comporte trois couches: une couche d'entrée, une couche cachée qui utilise la fonction gaussienne comme fonction d'activation et une couche de sortie avec une fonction d'activation linéaire. Les réseaux RBF ont de nombreuses utilisations, notamment l'approximation des fonctions [52-54], la classification et le contrôle des systèmes non linéaires. Ils présentent l'avantage d'une vitesse d'apprentissage rapide et permet d'éviter le problème du minimum local [55-57].

La structure du réseau de neurones RBF est illustrée à la figure 13.

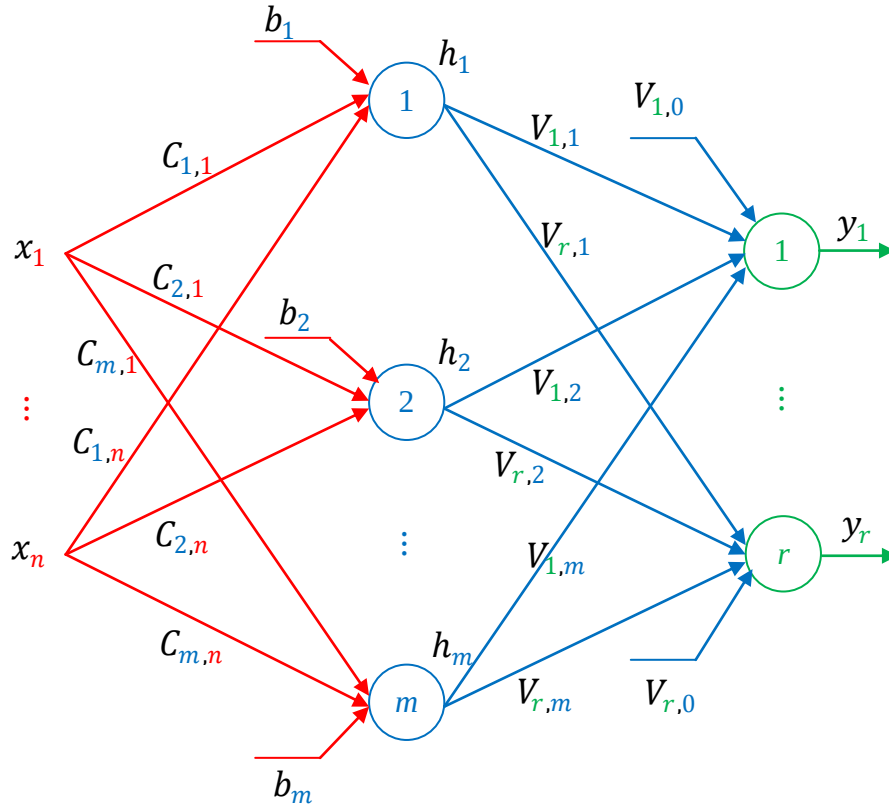


Figure IV.12: Structure d'un réseau de neurones RBF

Le vecteur d'entrée du réseau de neurones RBF est: $X = [x_1, x_2, \dots, x_n]^T$.

La sortie des neurones cachés avec centre $C_{j,i}$ et paramètre b_j est décrite comme suit:

$$h_j = \exp\left(-\frac{\|X - C_{j,i}\|^2}{2b_j^2}\right) \quad (4.53)$$

La sortie du réseau de neurone RBF peut être décrite par l'équation suivante:

$$y_k(k) = \sum_{j=1}^m V_{k,j} \cdot h_j = \sum_{j=1}^m V_{k,j} \cdot \exp\left(-\frac{\|X - C_{j,i}\|^2}{2b_j^2}\right) \quad (4.54)$$

Où $V_{k,j}$ est le poids entre la couche cachée et la couche de sortie.

Le centre $C_{j,i}$, le paramètre du largeur de base b_j et les poids $V_{k,j}$ du réseau de neurones RBF sont ajustés à l'aide de l'algorithme de descente du gradient afin de minimiser la somme de l'erreur carrée E_{RBF} (l'erreur entre la sortie système y et la sortie du réseau neuronal RBF y_{RBF}).

L'expression de E_{RBF} est donnée par:

$$E_{RBF}(k) = \frac{1}{2} \sum_{k=1}^r (e_{RBF}(k))^2 = \frac{1}{2} \sum_{k=1}^r (y(k) - y_{RBF}(k))^2 \quad (4.55)$$

Le changement de $C_{j,i}$, b_j et $V_{k,j}$ peut être exprimé comme suit:

$$C_{j,i}(k+1) = C_{j,i}(k) + \Delta C_{j,i} + \alpha (C_{j,i}(k) - C_{j,i}(k-1)) \quad (4.56)$$

$$b_j(k+1) = b_j(k) + \Delta b_j + \alpha (b_j(k) - b_j(k-1)) \quad (4.57)$$

$$V_{k,j}(k+1) = V_{k,j}(k) + \Delta V_{k,j} + \alpha (V_{k,j}(k) - V_{k,j}(k-1)) \quad (4.58)$$

Où α est le facteur moment, et η est le taux d'apprentissage.

$$\Delta C_{j,i} = -\eta \frac{\partial E_{RBF}(k)}{\partial C_{j,i}(k)} = -\eta \frac{\partial E_{RBF}(k)}{\partial y_{RBF}(k)} \frac{\partial y_{RBF}(k)}{\partial h_j} \frac{\partial h_j}{\partial C_{j,i}(k)} = \eta e_{RBF}(k) V_{k,j} h_j \frac{X - C_{j,i}}{b_j^2} \quad (4.59)$$

$$\Delta b_j = -\eta \frac{\partial E_{RBF}(k)}{\partial b_j(k)} = -\eta \frac{\partial E_{RBF}(k)}{\partial y_{RBF}(k)} \frac{\partial y_{RBF}(k)}{\partial h_j} \frac{\partial h_j}{\partial b_j(k)} = \eta e_{RBF}(k) V_{k,j} h_j \frac{\|X - C_{j,i}\|^2}{b_j^3} \quad (4.60)$$

$$\Delta V_{k,j} = -\eta \frac{\partial E_{RBF}(k)}{\partial V_{k,j}(k)} = -\eta \frac{\partial E_{RBF}(k)}{\partial y_{RBF}(k)} \frac{\partial y_{RBF}(k)}{\partial V_{k,j}(k)} = \eta e_{RBF}(k) h_j \quad (4.61)$$

IV-9) Conclusion

Ce chapitre a été consacré aux notions théoriques des réseaux de neurones artificiels. Les principales qualités des réseaux de neurones sont leur capacité d'approximer des problèmes non linéaires grâce à l'algorithme de la rétro-propagation du gradient. Ces qualités ont permis aux réseaux de neurones d'être utilisés dans plusieurs applications: le traitement d'image, le traitement de signal, la modélisation et la commande des systèmes non linéaires, etc.

Deux types des réseaux de neurones ont été présentés dans ce chapitre: les réseaux de neurones MLI et RBF. Par rapport aux réseaux de neurones MLI, les réseaux de neurones RBF présentent d'une vitesse d'apprentissage est plus rapide et ils permettent aussi d'éviter le problème du minimum local (l'algorithme de la rétro-propagation du gradient pourrait converger vers un minimum global).

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

V-1) Introduction

La concentration de l'oxygène dissous dans un procédé du traitement des eaux usées est un élément très essentiel qui a un effet direct sur la croissance des micro-organismes. Si le niveau d'oxygène dissous baisse, les micro-organismes ne seront pas capables de dégrader les matières polluantes. Pour ce faire, trois techniques de commande sont utilisées pour contrôler la concentration de l'oxygène dissous dans un bioréacteur du traitement des eaux usées par boues activées; la première; est un régulateur PI classique, la deuxième est la commande PI adaptative qui est basée aussi sur la technique des réseaux de neurones RBF et la troisième, est la commande proposée basée sur la méthode du gradient et la technique des réseaux de neurones RBF.

Dans ce chapitre, les performances de ces techniques de commande utilisées sont présentées. Les tests de simulation sont basés sur un modèle mathématique d'un bioréacteur du traitement des eaux usées par boues activées. Les simulations ont été réalisées sur le logiciel Matlab/simulink. Les performances de ces techniques de commande y compris l'adaptabilité, la robustesse, sont discutées.

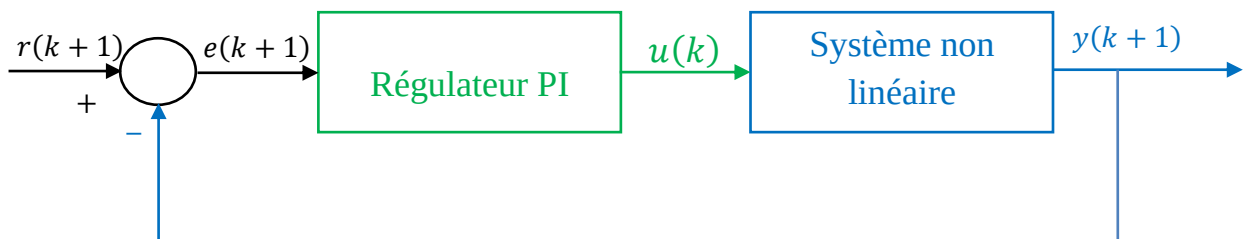
V-2) Régulateur PI

L'expression mathématique d'un régulateur PI (Proportional Integral) est donnée par l'équation suivante:

$$u(k) = u(k - 1) + k_p(e(k) - e(k - 1)) + k_i e(k) \quad (5.1)$$

Avec $u(k)$ est la sortie du régulateur PI (l'entrée de commande pour le système) et $e(k)$ est l'erreur entre la référence $r(k)$ et la sortie actuelle du système $y(k)$.

$$e(k + 1) = r(k + 1) - y(k + 1) \quad (5.2)$$



Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

Figure V.1: Schéma général d'un régulateur PI

Il est bien connu que les performances du régulateur PI dépendent de la valeur des paramètres k_p et k_i . Donc, le régulateur PI peut avoir d'excellentes performances avec des paramètres appropriés, sinon le régulateur ne peut pas répondre aux exigences souhaitées.

Pour pouvoir remédier à ce problème et améliorer les performances du régulateur PI, les paramètres k_p et k_i doivent être auto-ajustés en permanence.

V-3) La commande PI adaptative basée sur la technique des réseaux de neurones RBF

L'expression mathématique du régulateur PI adaptatif (Proportional Integral) est donnée par l'équation suivante:

$$u(k) = u(k-1) + k_p(k)(e(k) - e(k-1)) + k_i(k)e(k) \quad (5.3)$$

Les paramètres k_p et k_i peuvent être ajustés en utilisant la méthode de descente du gradient en minimisant la fonction de performance $E(k)$ qui est la somme de l'erreur au carrée entre la référence $r(k)$ et la sortie du système $y(k)$

La fonction de performance $E(k)$ est définie par l'expression suivante:

$$E(k) = \frac{1}{2} \sum_{k=1}^N (e(k))^2 = \frac{1}{2} \sum_{k=1}^N (r(k) - y(k))^2 \quad (5.4)$$

La mise à jour des paramètres du régulateur PI k_p et k_i sont ajustés en utilisant la méthode du gradient:

$$k_p(k) = k_p(k-1) + \Delta k_p(k) \quad (5.5)$$

$$k_i(k) = k_i(k-1) + \Delta k_i(k) \quad (5.6)$$

Avec:

$$\Delta k_p(k) = -\eta \frac{\partial E(k)}{\partial k_p(k)} = -\eta \frac{\partial E(k)}{\partial y(k)} \frac{\partial y(k)}{\partial u(k)} \frac{\partial u(k)}{\partial k_p(k)} = \eta e(k) \frac{\partial y(k)}{\partial u(k)} (e(k) - e(k-1)) \quad (5.7)$$

$$\Delta k_i(k) = -\eta \frac{\partial E(k)}{\partial k_i(k)} = -\eta \frac{\partial E(k)}{\partial y(k)} \frac{\partial y(k)}{\partial u(k)} \frac{\partial u(k)}{\partial k_i(k)} = \eta e(k) \frac{\partial y(k)}{\partial u(k)} e(k) \quad (5.8)$$

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

Dans le cas général, il est difficile de déterminer l'expression du terme $\frac{\partial y}{\partial u}$, mais, il est bien connu que la technique des réseaux de neurones est capable d'approximer n'importe quelle fonction non linéaire, donc, elle peut être utilisée pour approximer l'expression $\frac{\partial y}{\partial u}$.

V-3-1) Approximation de l'expression $\frac{\partial y}{\partial u}$ par la technique des réseaux de neurones RBF

Le réseau de neurones RBF est utilisé pour déterminer l'expression de la dérivée de sortie du système par rapport à son entrée $\frac{\partial y}{\partial u}$.

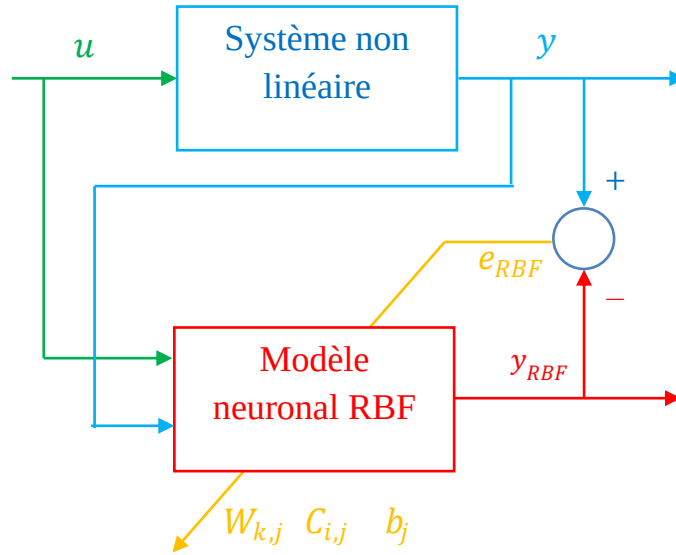


Figure V.2: Modélisation d'un système par la technique des réseaux de neurones

La dérivée de la sortie du système y par rapport son entrée u à l'instant k est approchée par la dérivée de la sortie du réseau de neurones RBF y_{RBF} par rapport à son entrée x_1 [58,59].

$$\frac{\partial y}{\partial u} \approx \frac{\partial y_{RBF}}{\partial u} = \frac{\partial y_{RBF}}{\partial x_1} = \frac{\partial}{\partial x_1} \sum_{j=1}^m W_{1,j} \exp\left(-\frac{\|X - C_{i,j}\|^2}{2b_j^2}\right) \quad (5.9)$$

$$\frac{\partial y}{\partial u} = \sum_{j=1}^m \frac{\partial}{\partial x_1} W_{1,j} \exp\left(-\frac{\|X - C_{i,j}\|^2}{2b_j^2}\right) \quad (5.10)$$

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

$$\frac{\partial y}{\partial u} = \sum_{j=1}^m W_{1,j} \frac{\partial}{\partial x_1} \exp \left(- \frac{X^T X - X^T C_{i,j} - C_{i,j}^T X + C_{i,j}^T C_{i,j}}{2b_j^2} \right) \quad (5.11)$$

$$\text{Avec: } X = [u, y]^T = [x_1, x_2]^T, \quad C_{i,j} = \begin{bmatrix} C_{1,1} & C_{1,2} & \cdots & C_{1,m} \\ C_{2,1} & C_{2,2} & \cdots & C_{2,m} \end{bmatrix}$$

$$\frac{\partial y}{\partial u} = \sum_{j=1}^m W_{1,j} \frac{\partial}{\partial x_1} \exp \left(- \frac{[x_1 \quad x_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - [x_1 \quad x_2] C_{i,j} - C_{i,j}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + C_{i,j}^T C_{i,j}}{2b_j^2} \right) \quad (5.12)$$

$$\frac{\partial y}{\partial u} = \sum_{j=1}^m W_{1,j} \frac{\partial}{\partial x_1} \exp \left(- \frac{x_1^2 + x_2^2 - 2x_1 C_{1,j} - 2x_2 C_{2,j} + C_{i,j}^T C_{i,j}}{2b_j^2} \right) \quad (5.13)$$

$$\frac{\partial y}{\partial u} = \sum_{j=1}^m W_{1,j} \left(\frac{C_{1,j} - x_1}{b_j^2} \right) \exp \left(- \frac{x_1^2 + x_2^2 - 2x_1 C_{1,j} - 2x_2 C_{2,j} + C_{i,j}^T C_{i,j}}{2b_j^2} \right) \quad (5.14)$$

$$\frac{\partial y}{\partial u} = \sum_{j=1}^m W_{1,j} \left(\frac{C_{1,j} - x_1}{b_j^2} \right) h_j = \sum_{j=1}^m W_{1,j} \left(\frac{C_{1,j} - u}{b_j^2} \right) h_j \quad (5.15)$$

Donc, l'expression de la dérivée de la sortie du système y par rapport à son entrée u est:

$$\frac{\partial y(k)}{\partial u(k)} = \sum_{j=1}^m W_{1,j} \left(\frac{C_{1,j} - x_1}{b_j^2} \right) h_j \quad (5.16)$$

La figure 3 montre la structure du régulateur ou contrôleur PI adaptatif basé sur la technique des réseaux de neurones RBF. Le régulateur PI est utilisé pour commander le système non linéaire afin de satisfaire des exigences souhaitées. Puisque les performances du contrôleur PI dépendent des paramètres k_p et k_i , alors, la technique des réseaux de neurones est utilisée pour ajuster automatiquement ces paramètres k_p et k_i .

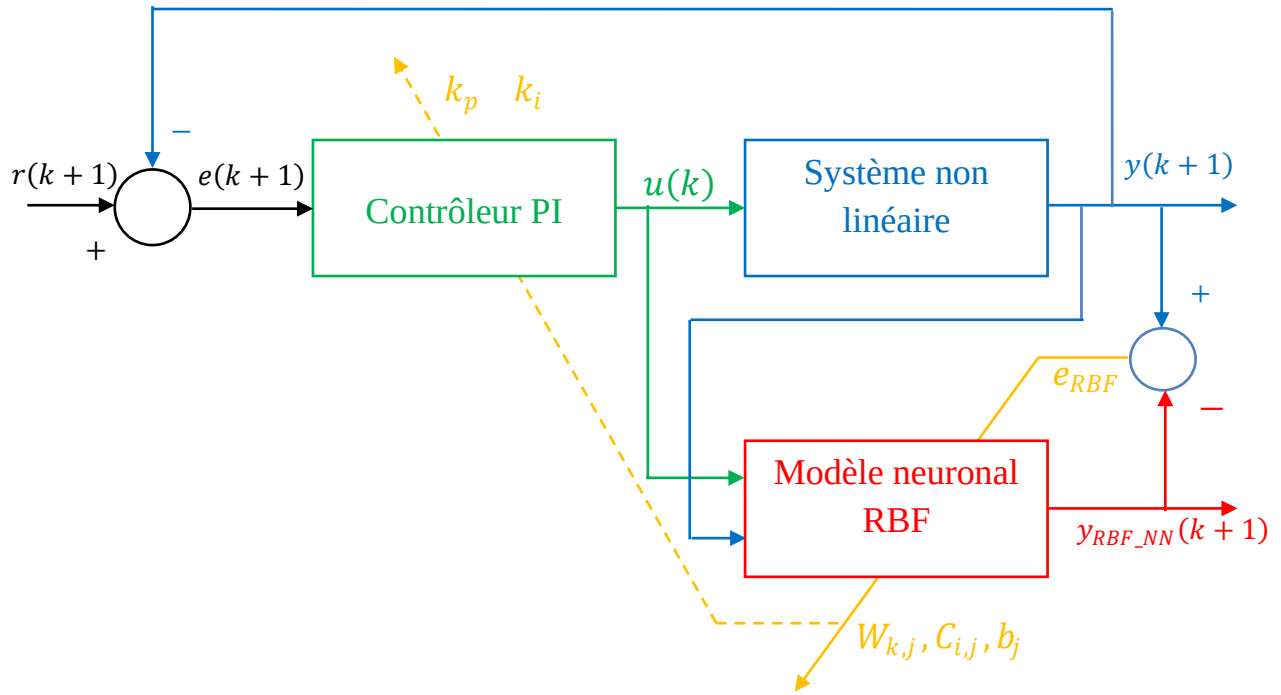


Figure V.3: La commande PI adaptative basée sur la technique des réseaux de neurones RBF

Cette combinaison entre le régulateur PI et la technique intelligente des réseaux de neurones permet d'augmenter les performances du régulateur classique PI et d'améliorer son adaptabilité contre les perturbations et les variations au niveau du système à commander.

V-4) Principe de fonctionnement de la commande de gradient basée sur la technique des réseaux de neurones

On considère un système non linéaire décrit par l'équation différentielle suivante:

$$\frac{\partial y_u(t)}{\partial t} = f(t, y_u(t), u(t)) \quad (5.17)$$

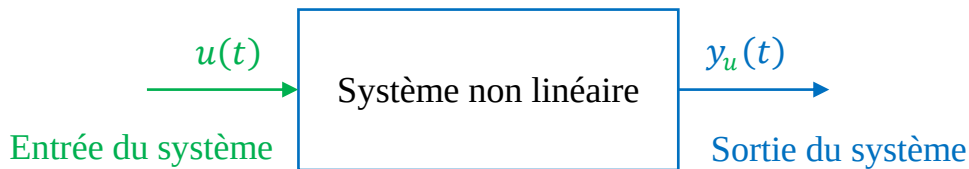


Figure V.4: Système non linéaire avec une seule entrée et une seule sortie

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

La trajectoire de la sortie du système $y_u(t)$ dépend de l'entrée de commande $u(t)$. Si on considère les deux entrées $u(t) = u_0$ et $u(t) = u_1$, alors l'équation (5.17) pourra devenir:

$$\frac{\partial y_{u_0}(t)}{\partial t} = f(t, y_{u_0}(t), u_0) \quad (5.18)$$

$$\frac{\partial y_{u_1}(t)}{\partial t} = f(t, y_{u_1}(t), u_1) \quad (5.19)$$

Les solutions numériques des équations (équation (5.18) et équation (5.19)) en utilisant la méthode d'Euler explicite sont:

$$y_{u_0}(t_{k+1}) = y_{u_0}(t_k) + hf(t_k, y_{u_0}(t_k), u_0) \quad (5.20)$$

$$y_{u_1}(t_{k+1}) = y_{u_1}(t_k) + hf(t_k, y_{u_1}(t_k), u_1) \quad (5.21)$$

La figure 5 montre les courbes des équations (5.20) et (5.21).

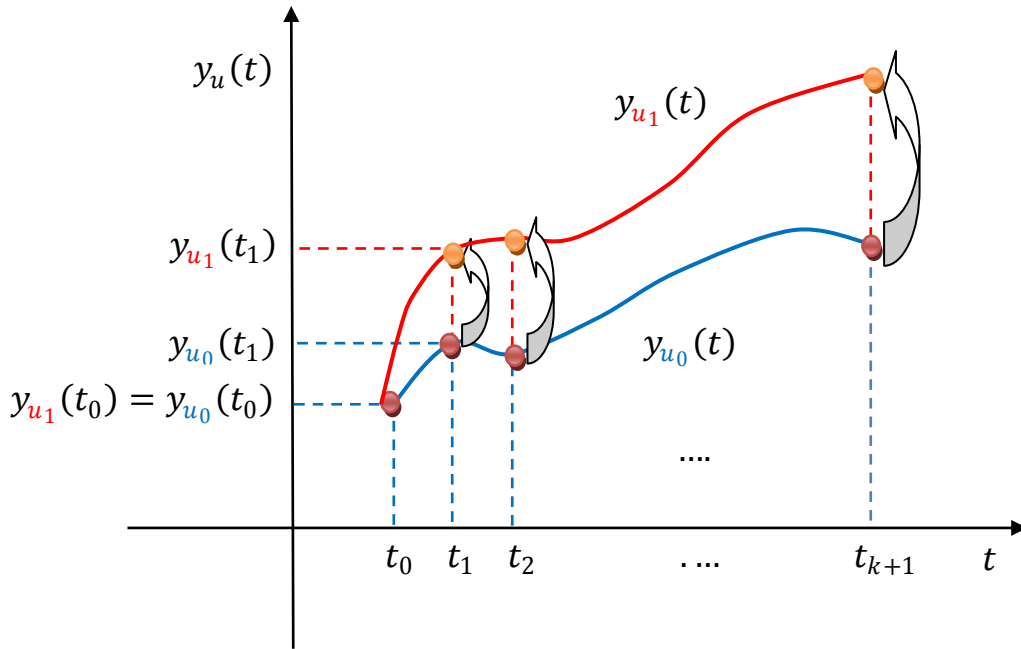


Figure V.5: Courbes de l'équation (5.17) avec deux entrées de commande différentes

$$u(t) = u_0 \text{ et } u(t) = u_1$$

La figure 5 montre, si on applique deux entrées différentes $u(t) = u_0$ et $u(t) = u_1$ sur le système présenté dans la figure 4, on aura deux trajectoires différentes sur la sortie du système.

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

On suppose qu'on sait l'entrée de commande $u(t) = u_0$, si on veut que la trajectoire de la sortie du système $y_{u_0}(t_{k+1})$ (la courbe en bleue) suive la trajectoire $y_{u_1}(t_{k+1})$ (la courbe en rouge), on doit trouver la valeur de la commande u_1 qui fait que y_{u_0} suit bien y_{u_1} .

L'objectif de cette stratégie de commande est de contrôler la sortie du système $y_{u_0}(t_{k+1})$ à suivre une référence $r(t_{k+1})$ en faisant changer à chaque instant t_k la valeur de la commande u_k .

V-4-1) La méthode du gradient

On considère le système non linéaire suivant:

$$\frac{\partial y_{u_0}(t)}{\partial t} = f(t, y_{u_0}(t), u_0) \quad (5.22)$$

La solution numérique de cette équation différentielle en utilisant la méthode d'Euler explicite est:

$$y_{u_0}(t_1) = y_{u_0}(t_0) + hf(t_0, y_{u_0}(t_0), u_0) \quad (5.23)$$

A l'instant t_1 , on doit chercher la valeur de l'entrée de la commande u_1 pour avoir $y_{u_0}(t_1) = r(t_1)$.

$$y_{u_1}(t_1) = y_{u_0}(t_0) + hf(t_0, y_{u_0}(t_0), u_1) \quad (5.24)$$

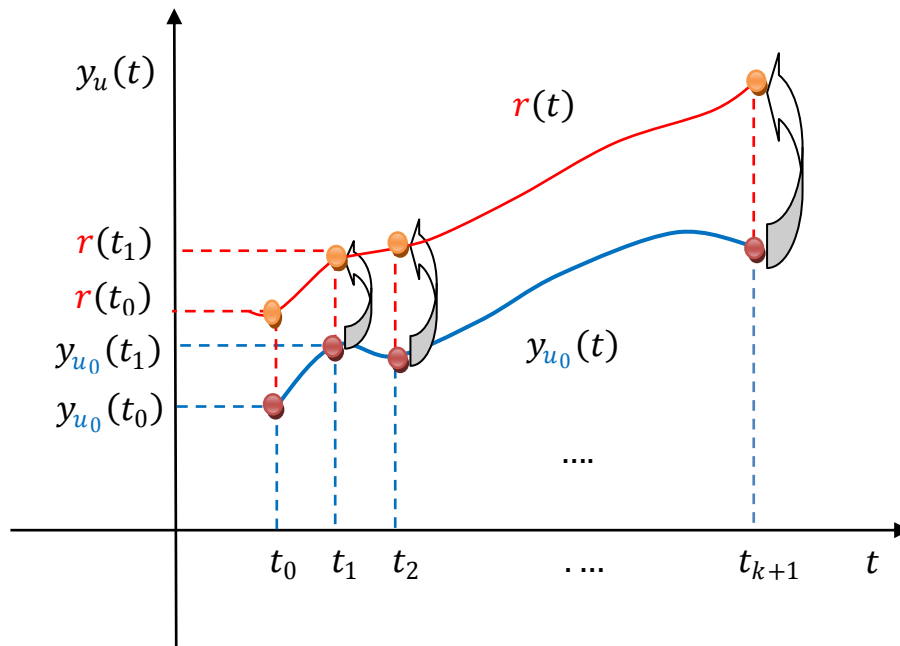


Figure V.6: Principe du fonctionnement de la commande proposée

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

L'entrée de commande u_1 est ajustée en utilisant la méthode de descente du gradient en minimisant la fonction de performance en respectant u_0 . La fonction de performance choisie est l'erreur carrée $E(t_1)$ entre $y_{u_1}(t_1)$ et $y_{u_0}(t_1)$

$$E(t_1) = \frac{1}{2} (e(t_1))^2 = \frac{1}{2} (y_{u_1}(t_1) - y_{u_0}(t_1))^2 = \frac{1}{2} (r(t_1) - y_{u_0}(t_1))^2 \quad (5.25)$$

La mise à jour de la commande u_1 par la méthode de descente du gradient:

$$u_1 = u_0 + \Delta u_0 \quad (5.26)$$

$$\Delta u_0 = -\lambda \frac{\partial E(t_1)}{\partial u_0} \quad (5.27)$$

alors:

$$u_1 = u_0 - \lambda \frac{\partial E(t_1)}{\partial u_0} \quad (5.28)$$

Avec: λ le taux d'apprentissage.

$$u_1 = u_0 - \lambda \frac{\partial E(t_1)}{\partial y_{u_0}(t_1)} \frac{\partial y_{u_0}(t_1)}{\partial u_0} \quad (5.29)$$

$$u_1 = u_0 + \lambda e(t_1) \frac{\partial y_{u_0}(t_1)}{\partial u_0} \quad (5.30)$$

La dérivée de la sortie du système y_{u_0} par rapport à son entrée u_0 à l'instant t_1 est approchée par la dérivée de la sortie du réseau de neurones RBF y_{RBF} par rapport à son entrée x_1 .

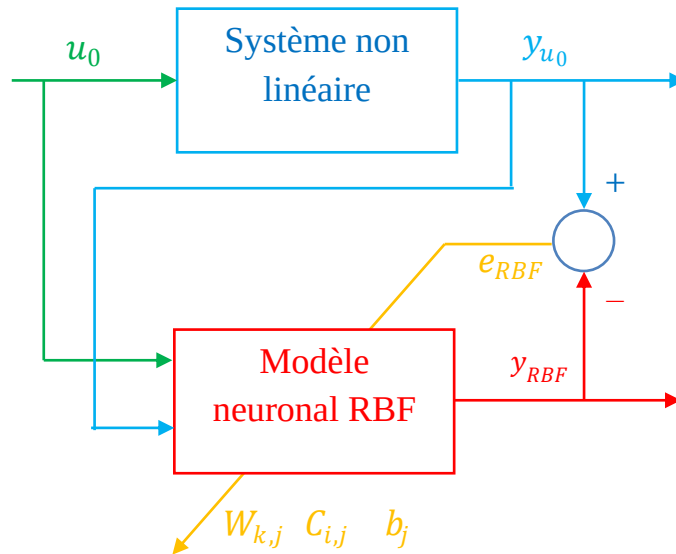


Figure V.7: Modélisation d'un système par la technique des réseaux de neurones

A partir de l'équation (4.16), l'expression du terme $\frac{\partial y_{u_0}(t_1)}{\partial u_0}$ est:

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

$$\frac{\partial y_{u_0}(t_1)}{\partial u_0} = \sum_{j=1}^m W_{1,j} \left(\frac{C_{1,j} - u_0}{b_j^2} \right) h_j \quad (5.31)$$

Finalement, l'expression de la loi de commande s'écrit:

$$u_1 = u_0 + \lambda e(t_1) \sum_{j=1}^m W_{1,j} \left(\frac{C_{1,j} - x_1}{b_j^2} \right) h_j \quad (5.32)$$

Les étapes de la stratégie de commande proposée sont les suivantes:

- 1) On choisit u_0 (le choix de u_0 dépend du système étudié).
- 2) Initialisation des paramètres du réseau de neurones RBF: le nombre de neurones dans la couche d'entrée, la couche cachée et la couche de sortie, $C_{i,j}$, b_j et $W_{k,j}$ et le taux d'apprentissage.
- 3) Pour $k \geq 0$
 - * On détermine la sortie du système $y_{u_k}(t_{k+1})$ ainsi que la sortie du réseau de neurones RBF $y_{RBF}(t_{k+1})$.
 - * On calcule l'erreur $e_{RBF}(t_{k+1})$ entre la sortie du système et celle du réseau de neurones RBF.
 - * On fait régler les paramètres du réseau de neurones: $C_{i,j}$, b_j et $W_{k,j}$ en utilisant les équations (4.56) à (4.61).
 - * On calcule l'erreur $e(t_{k+1})$ entre la référence $r(t_{k+1})$ et la sortie du système $y_{u_k}(t_{k+1})$.
 - * On calcule la nouvelle valeur via l'expression suivante:

$$u_{k+1} = u_k + \lambda e(t_{k+1}) \sum_{j=1}^m W_{1,j} \left(\frac{C_{1,j} - x_k}{b_j^2} \right) h_j \quad (5.33)$$

- * On remplace la valeur de la commande trouvée u_{k+1} dans le système afin d'avoir $y_{u_k}(t_{k+1}) = r(t_{k+1})$.

$$y_{u_k}(t_{k+1}) = y_{u_k}(t_k) + hf(t_k, y_{u_k}(t_k), u_{k+1}) \quad (5.34)$$

La structure de la commande proposée est montrée dans la figure 8.

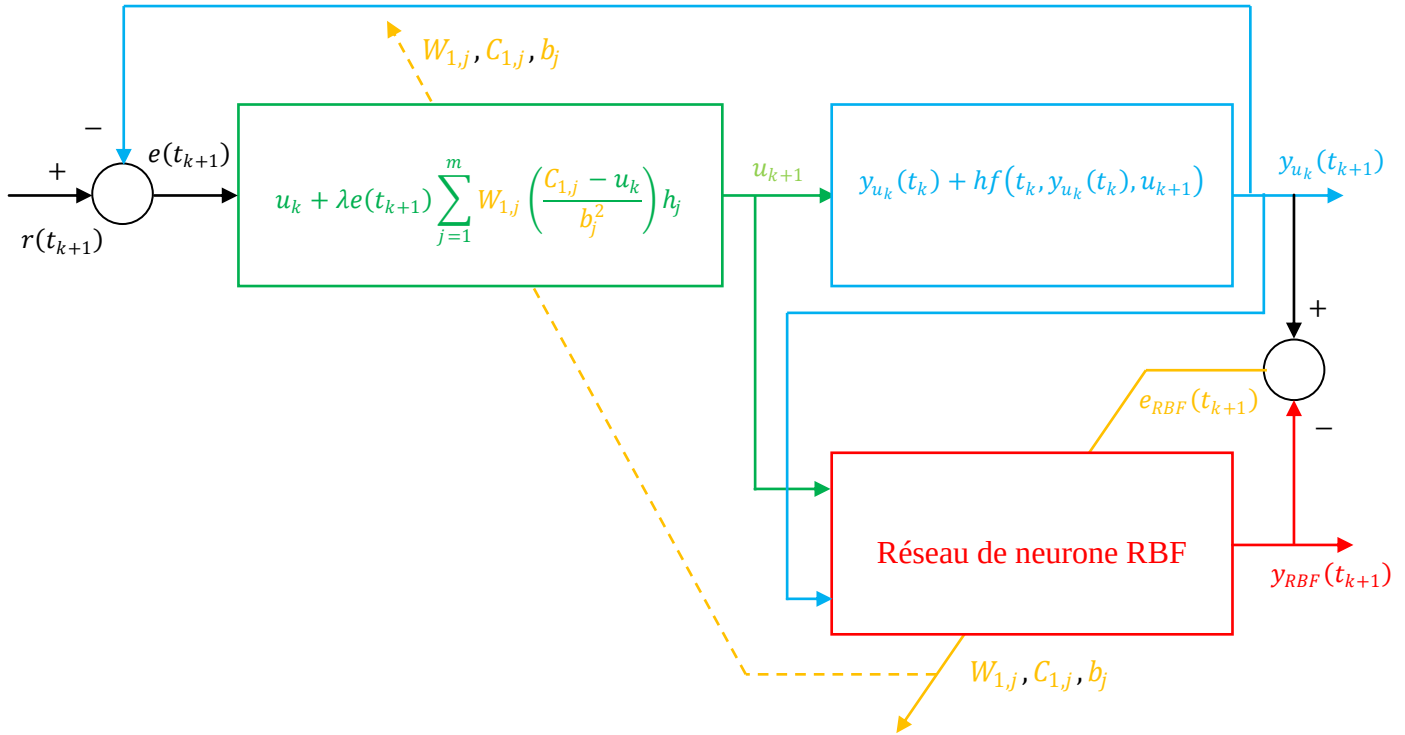


Figure V.8: Structure de la technique de commande du gradient basée sur les réseaux de neurones RBF

V-5) Modèle mathématique d'un bioréacteur de traitement des eaux usées par boues activées

Le processus de boues activées est un traitement biologique qui utilise les micro-organismes (biomasse) pour éliminer la matière organique, l'azote et le phosphore, mais l'élimination organique et l'azote sont les plus utilisés dans le traitement des eaux usées. Le schéma du processus du traitement des eaux usées est illustré à la figure 9.

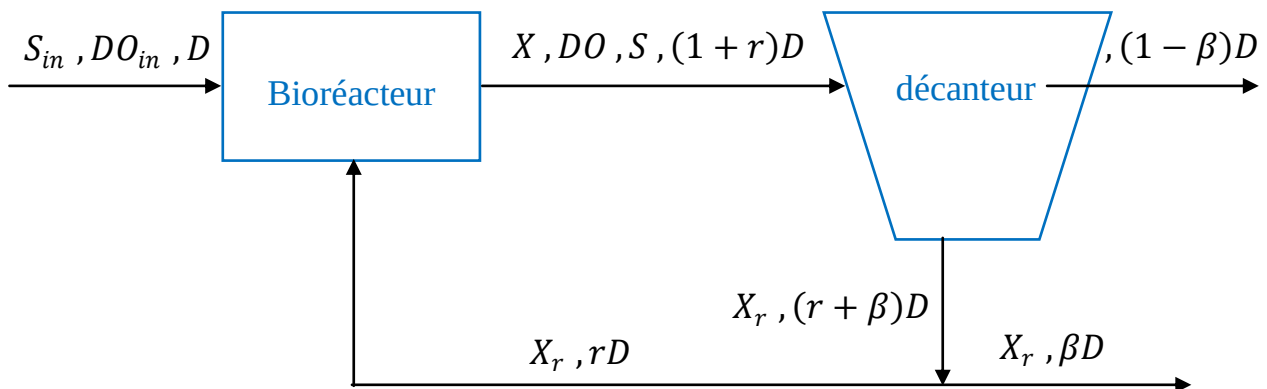


Figure V.9: Schéma général d'un bioréacteur du traitement des eaux usées par boues activées

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

Il se compose d'un réacteur biologique (cuve d'aération) où se développe la population de micro-organismes (biomasse) visant à retirer le substrat du réacteur, et d'un décanteur. Dans le décanteur, les solides sont séparés des eaux usées. Une partie des boues éliminées est recyclée vers le bassin d'aération.

Le modèle mathématique considéré dans ce chapitre possède quatre équations différentielles, la concentration en biomasse X , la concentration en substrat S , la concentration en oxygène dissous DO et la concentration en biomasse recyclée X_r . Le modèle est donné par les équations suivantes [60].

$$\frac{\partial X(t)}{\partial t} = \mu(t)X(t) - D(1+r)X(t) + rDX_r(t) \quad (5.35)$$

$$\frac{\partial S(t)}{\partial t} = -\frac{1}{Y}\mu(t)X(t) - D(1+r)S(t) + DS_{in} \quad (5.36)$$

$$\frac{\partial DO(t)}{\partial t} = -\frac{K_0}{Y}\mu(t)X(t) - D(1+r)DO(t) + KLa(DO_{max} - DO(t)) + DDO_{in} \quad (5.37)$$

$$\frac{\partial X_r(t)}{\partial t} = D(1+r)X(t) - D(\beta + r)X_r(t) \quad (5.38)$$

Avec:

$$\mu(t) = \mu_{max} \left(\frac{S(t)}{k_s + S(t)} \right) \left(\frac{DO(t)}{k_{DO} + DO(t)} \right) \quad (5.39)$$

Avec: $KLa = \alpha W(k)$

W : le taux d'aération.

Le taux d'aération est considéré comme la variable de commande pour contrôler la concentration de l'oxygène dissous.

Les valeurs des paramètres de ce modèle sont données dans l'annexe.

V-6) Résultats de simulation

La technique de commande proposée basée sur la méthode du gradient et la technique intelligente des réseaux de neurones RBF est utilisée pour contrôler la concentration du substrat S via le contrôle de l'oxygène dissous DO dans un bioréacteur aérobic. La concentration du substrat doit être inférieure à 20 mg.l^{-1} .

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

Les figures 10-14 montrent respectivement (en boucle ouverte): la concentration de la biomasse X , le substrat S , l'oxygène dissous DO et la concentration de la biomasse recyclée X_r .

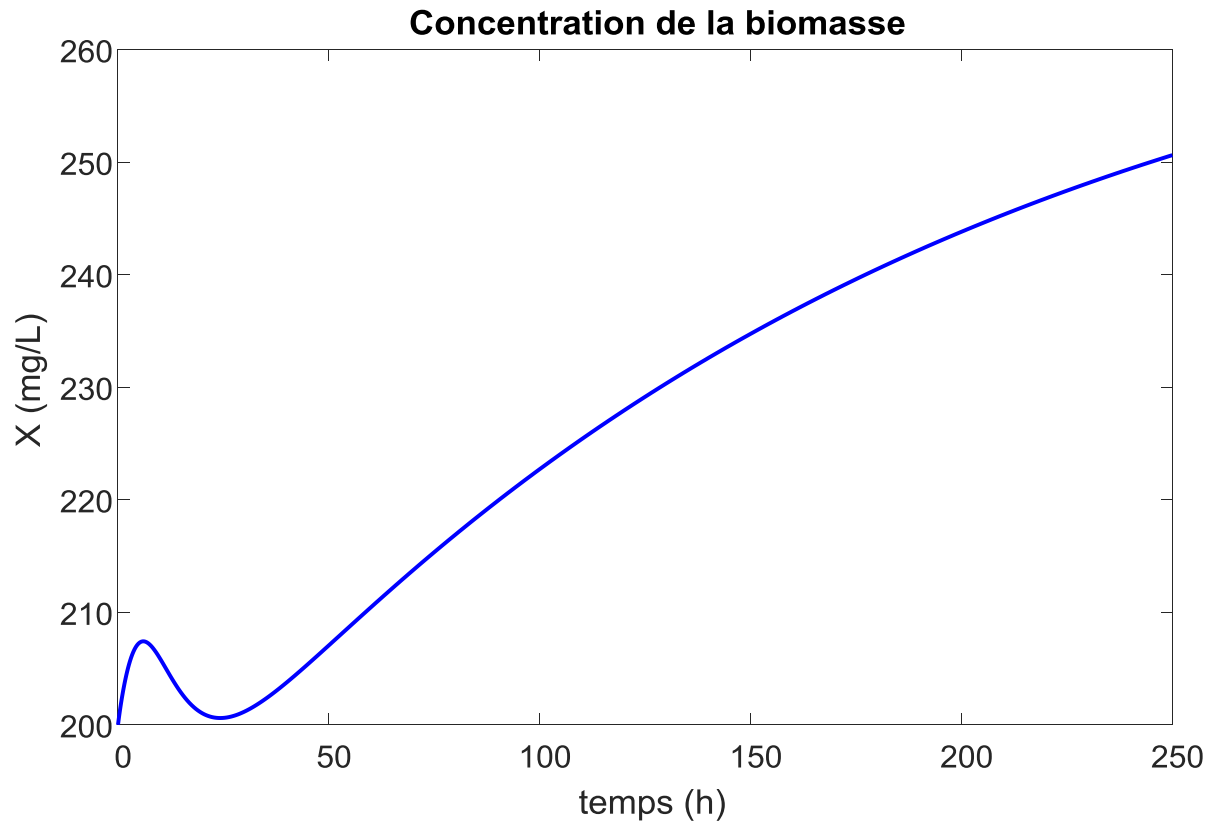


Figure V.10: Concentration de la biomasse

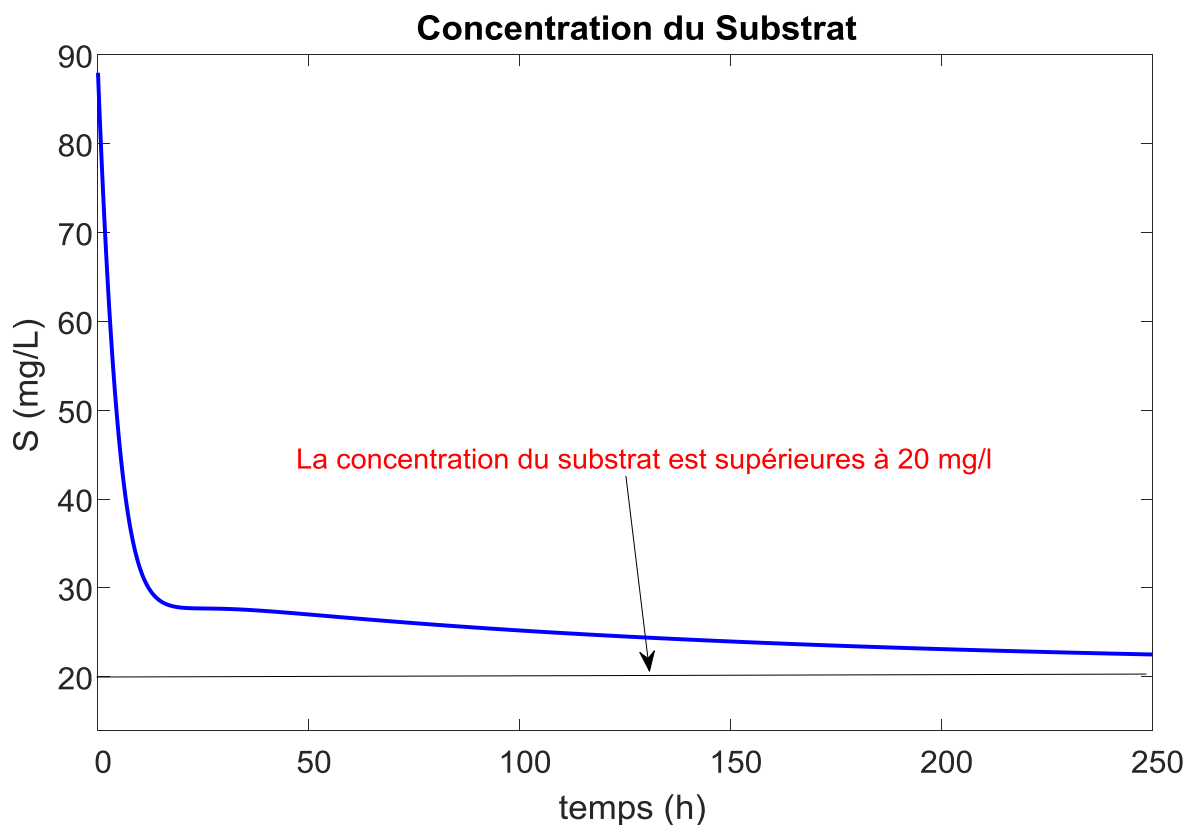


Figure V.11: Concentration du substrat

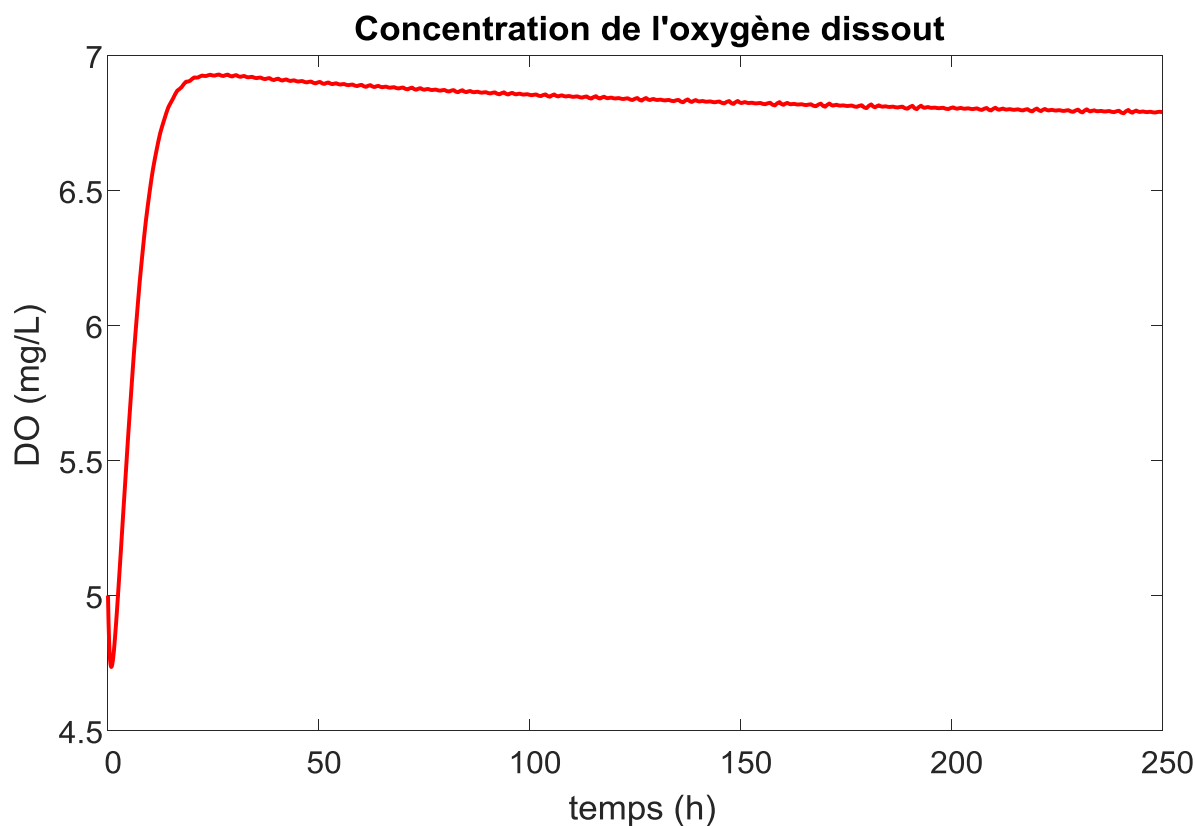


Figure V.12: Concentration de l'oxygène dissout

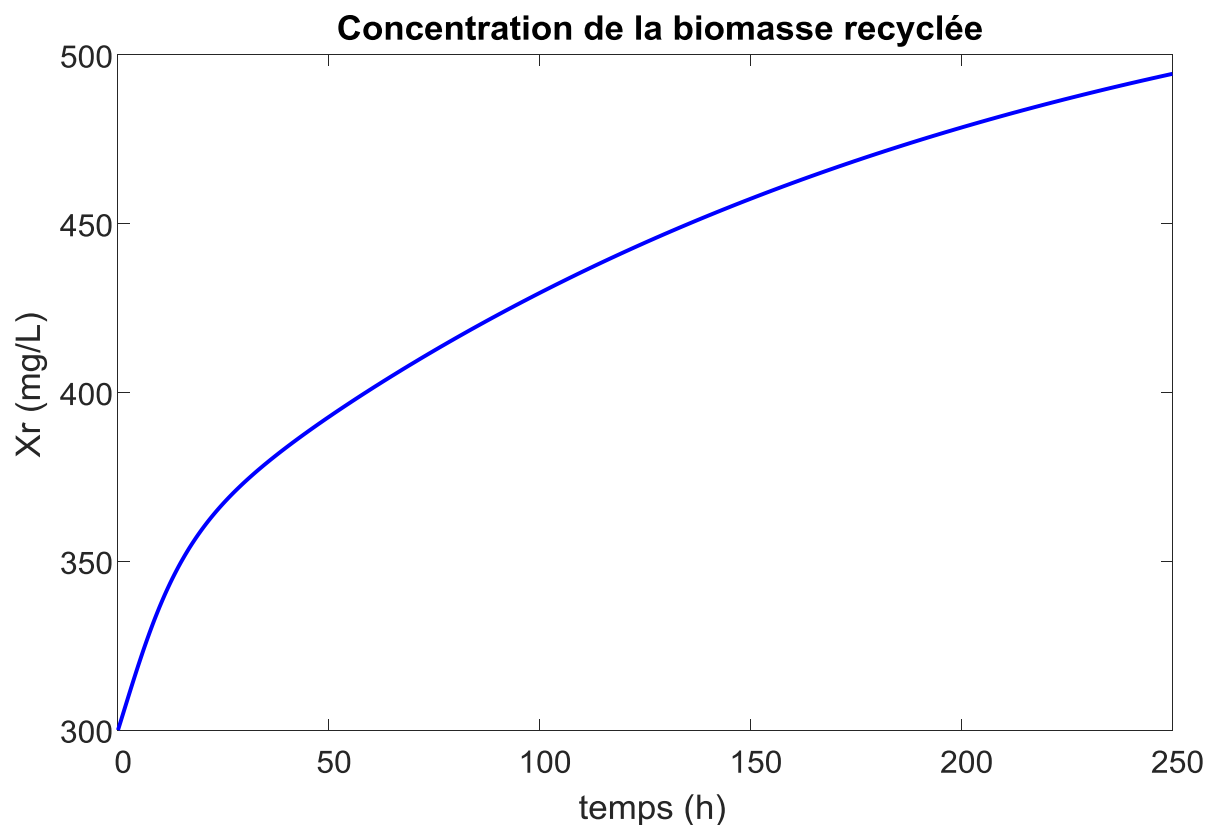


Figure V.13: Concentration de la biomasse recyclée

On peut clairement remarquer que la concentration du substrat S dépasse la concentration maximale autorisée 20 mg.l^{-1} . Donc, le contrôle de la concentration de l'oxygène dissous devient une obligation afin d'avoir une concentration du substrat inférieure à la valeur standard 20 mg.l^{-1} .

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

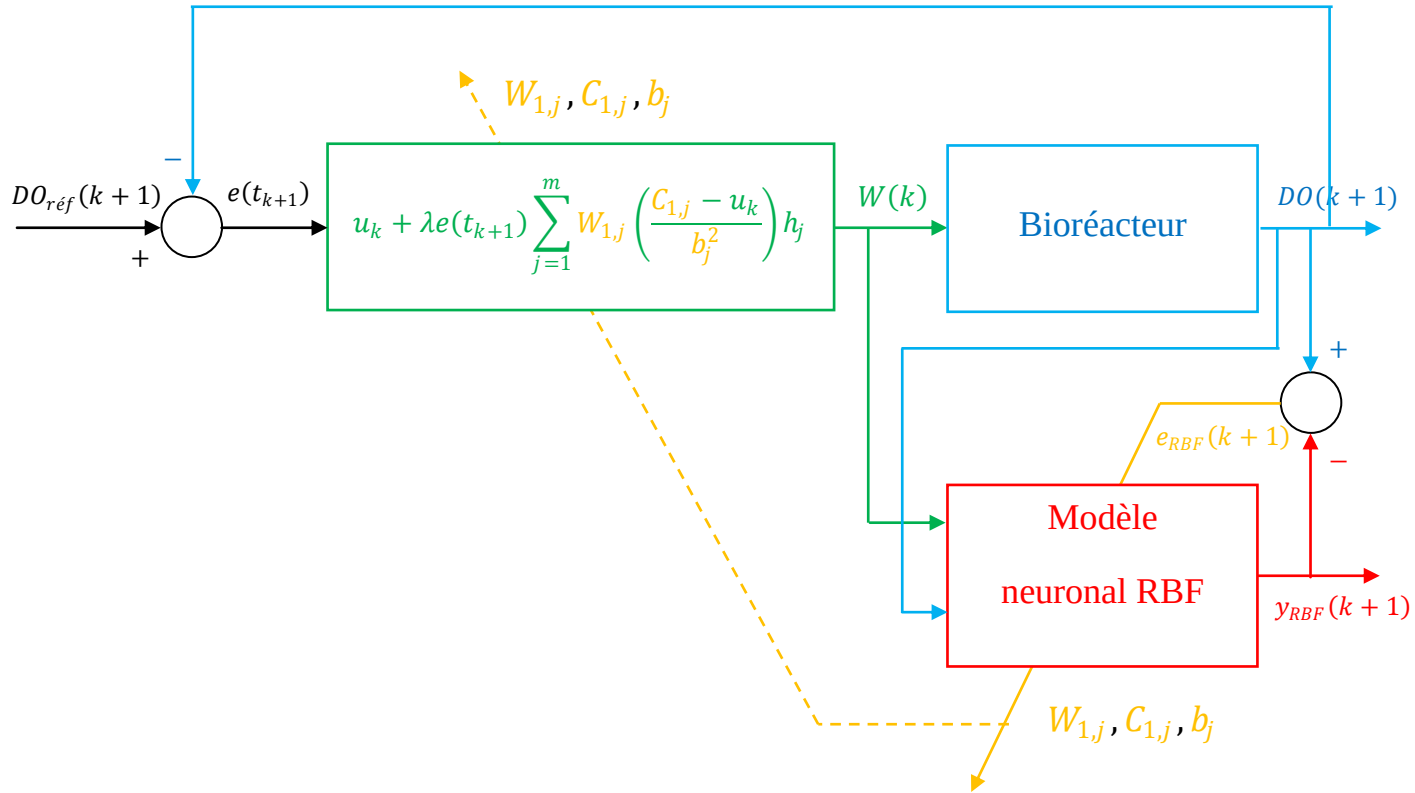
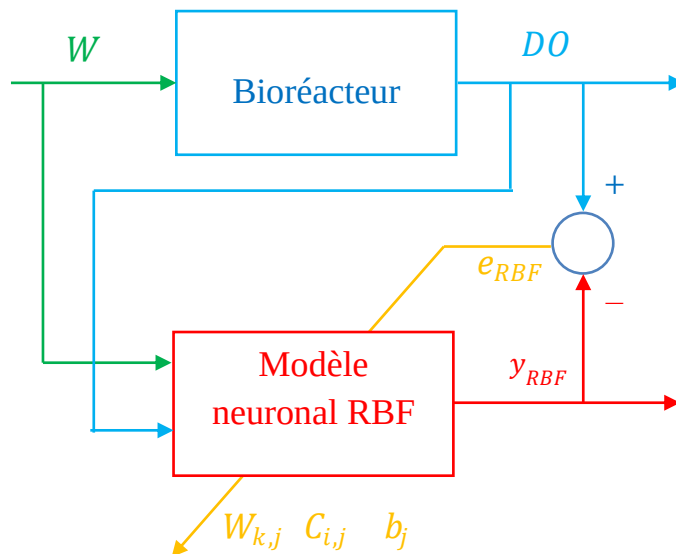


Figure V.14: Structure de la commande proposée

La structure du modèle neuronal RBF utilisée pour les deux techniques de commande: la commande du gradient basée sur la technique des réseaux de neurones et la commande PI adaptative (après plusieurs essais) est:

- * Deux entrées dans la couche d'entrée $[W, DO]^T$.
- * Quinze neurones dans la couche cachée.
- * Un seul neurone dans la couche de sortie.



Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

Figure V.15: Modèle neuronal d'un système non linéaire (bioréacteur)

Les paramètres du réseau de neurones RBF utilisées (après plusieurs essais) sont:

- * $C_{i,j} = C_{2,15}$ est initialisé entre l'intervalle $[60,70]$.
- * $b_j = b_{15}$ est initialisé entre l'intervalle $[30,40]$.
- * $W_{k,j} = W_{1,15}$ est initialisé entre l'intervalle $[0,10]$.
- * Le taux d'apprentissage $\eta = 0.09$.
- * Le facteur de momentum $\alpha = 0.5$.

La commande PI adaptative basée sur la technique des réseaux de neurones RBF a été utilisé afin de pouvoir faire une comparaison avec la technique de commande proposée.

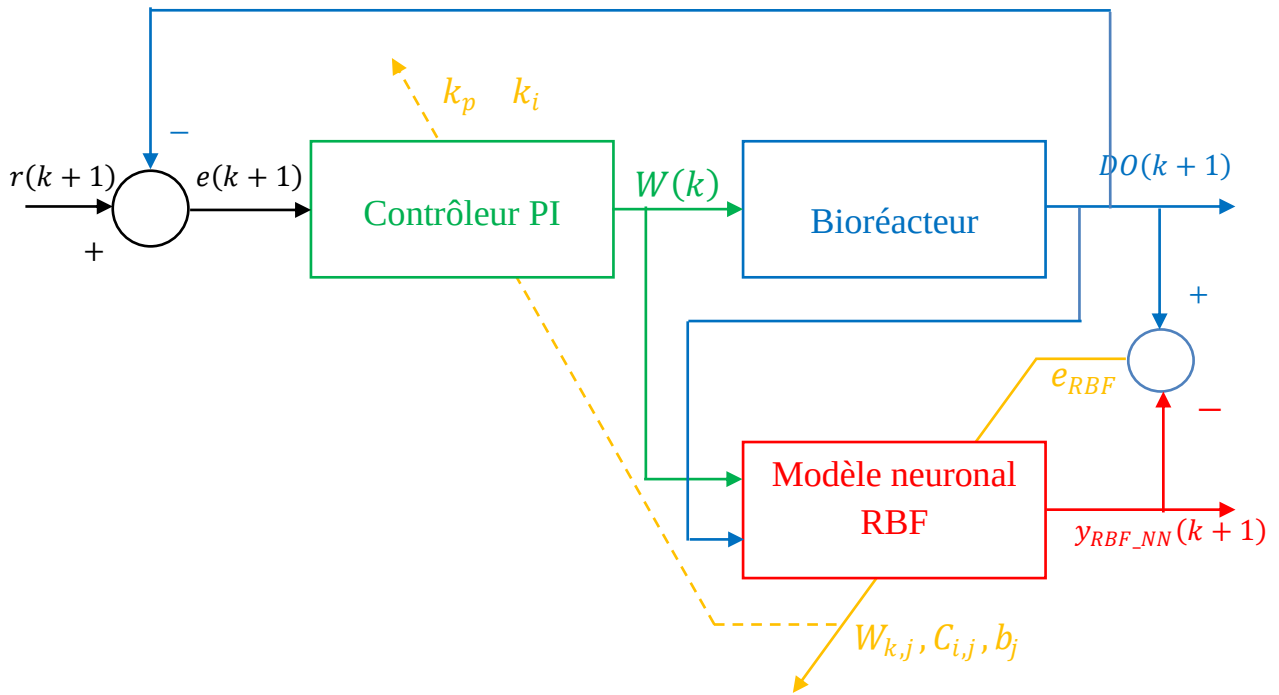


Figure V.16: La commande PI adaptative basée sur la technique des réseaux de neurones RBF

Le taux de dilution est considéré comme variable (le débit d'eau entrant dans le bioréacteur n'est pas constant dans un système du traitement des eaux usées) afin de couvrir des différents régimes: un taux élevé $D = 0.035 \text{ h}^{-1}$, un taux normal $D = 0.025 \text{ h}^{-1}$ et un taux faible $D = 0.015 \text{ h}^{-1}$.

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

La concentration du substrat de l'entrée S_{in} est considérée aussi avec différentes valeurs pour assurer une véritable étude du système du traitement des eaux usées.

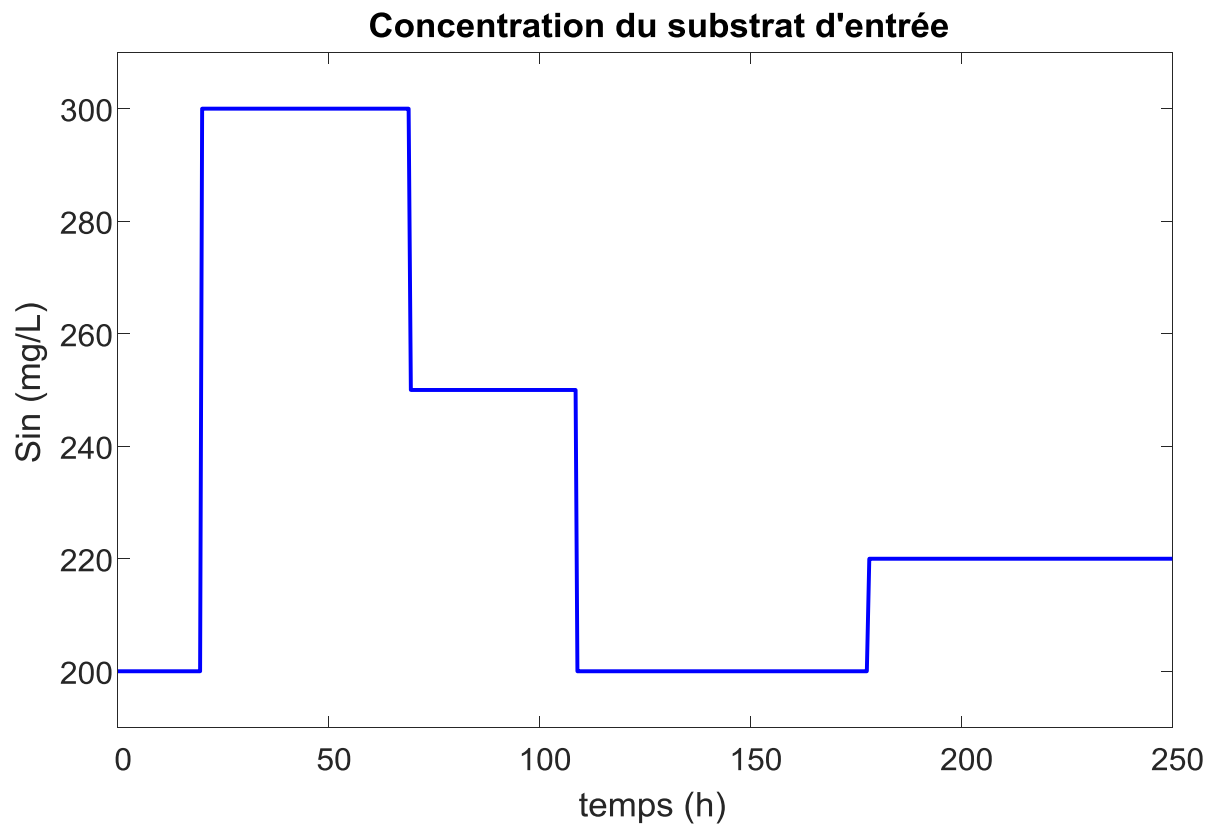


Figure V.17: Concentration du substrat S_{in}

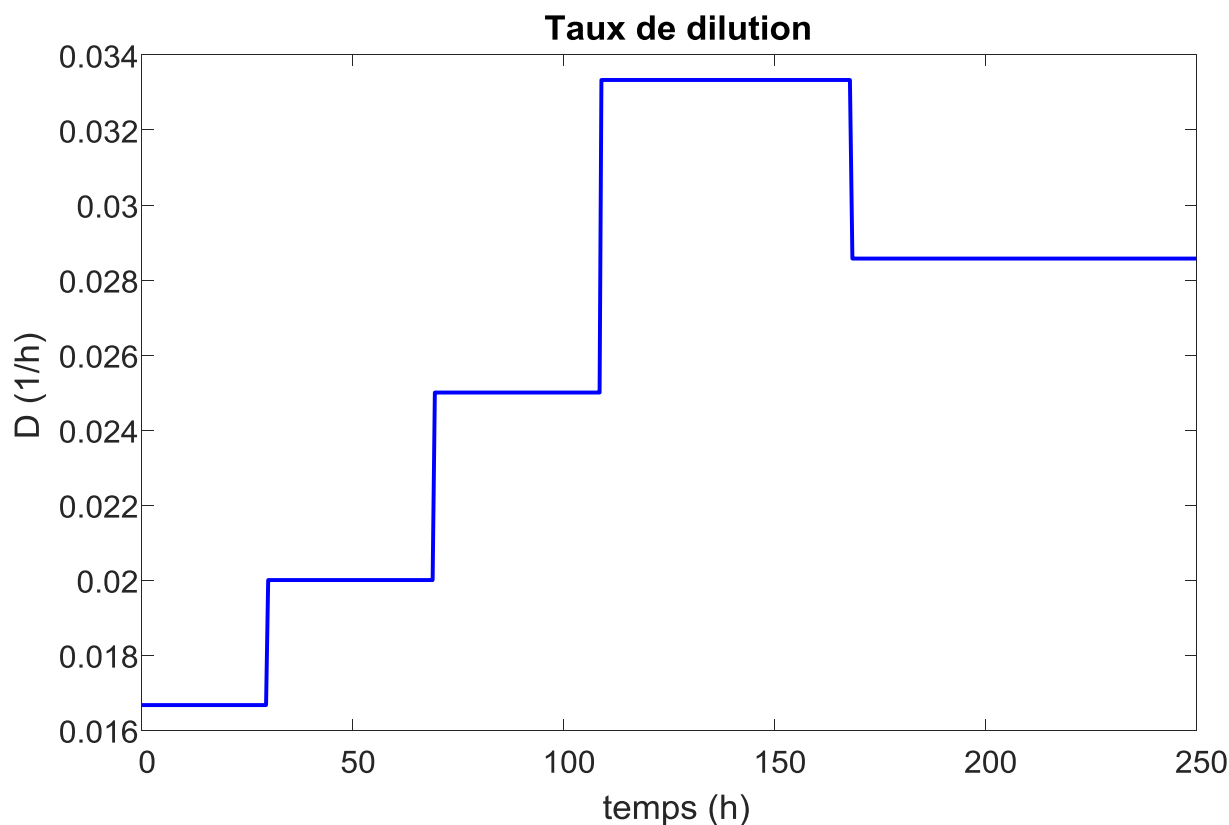


Figure V.18: Taux de dilution

La figure 17 montre la concentration de l'oxygène dissous dans un bioréacteur aérobie contrôlée par trois techniques:

- * La commande proposée basée sur la méthode du gradient de la technique des réseaux de neurones RBF.
- * Le contrôleur PI classique avec des paramètres fixes: $k_p = 3$ et $k_i = 0.9$.
- * La commande PI adaptative basée sur la technique des réseaux de neurones RBF.

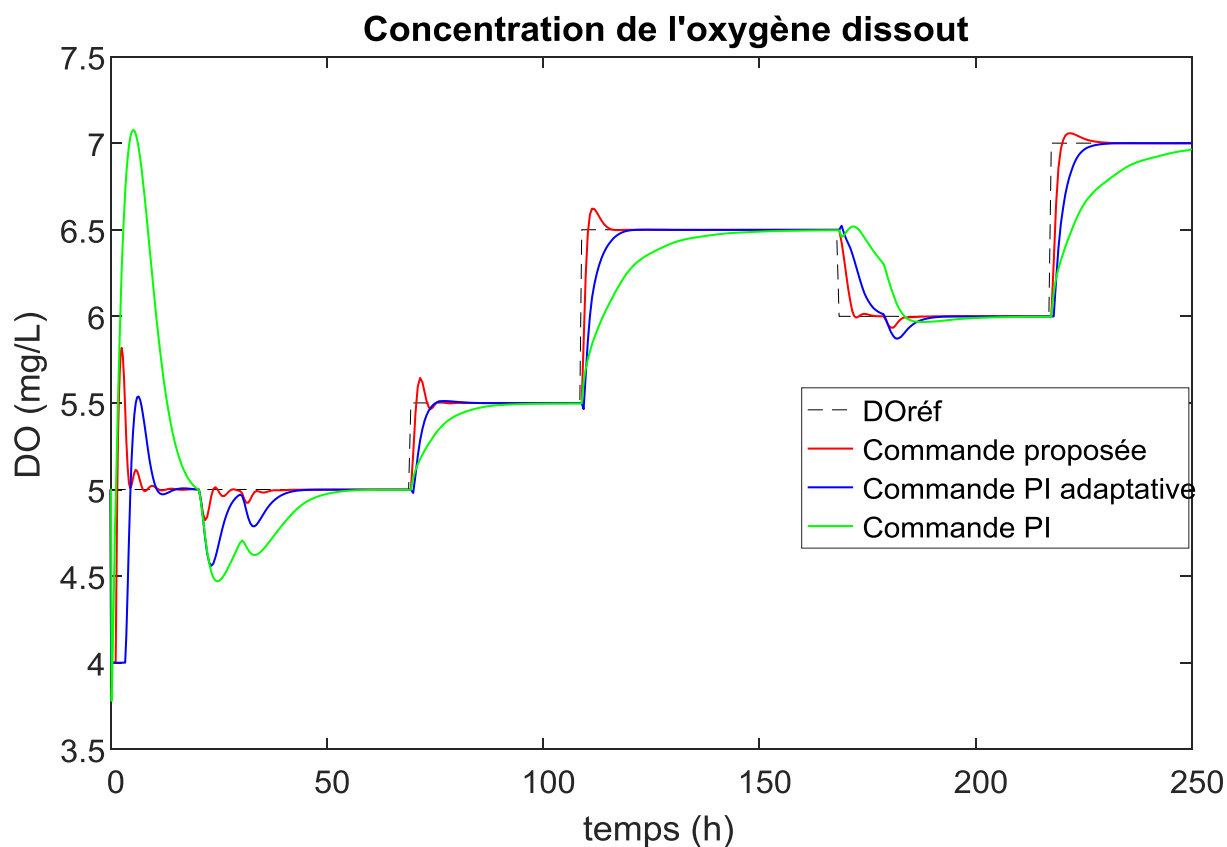


Figure V.19: Concentration de l'oxygène dissout

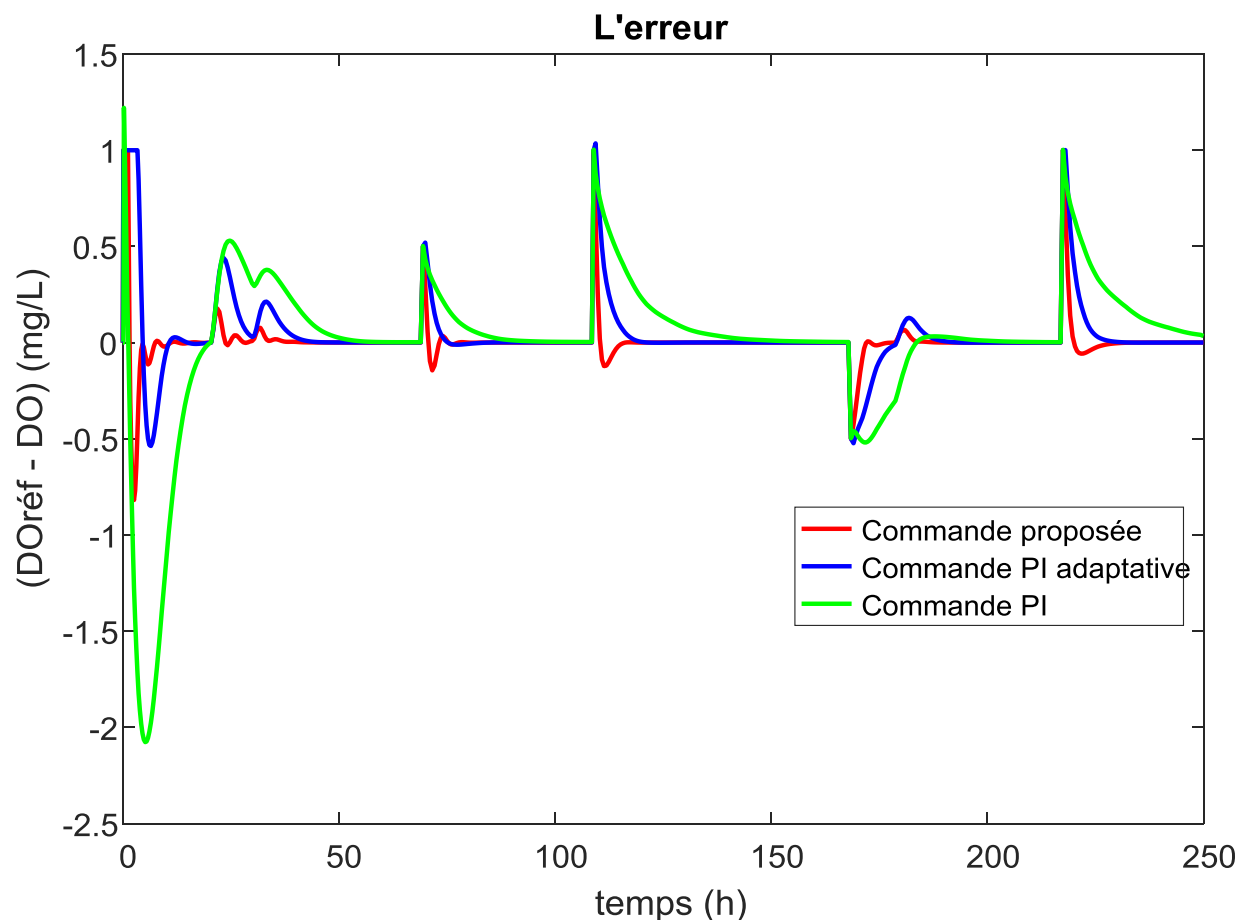


Figure V.20: L'erreur entre la référence et la concentration de l'oxygène dissous

La figure 19 et 20 montre clairement que la commande proposée donne des résultats meilleurs (la concentration de l'oxygène dissout suit bien la référence DO_{ref}) que la commande PI classique et la commande PI adaptative.

Donc, la commande proposée rejette bien les perturbations générées par la concentration du substrat S_{in} et le taux de dilution D .

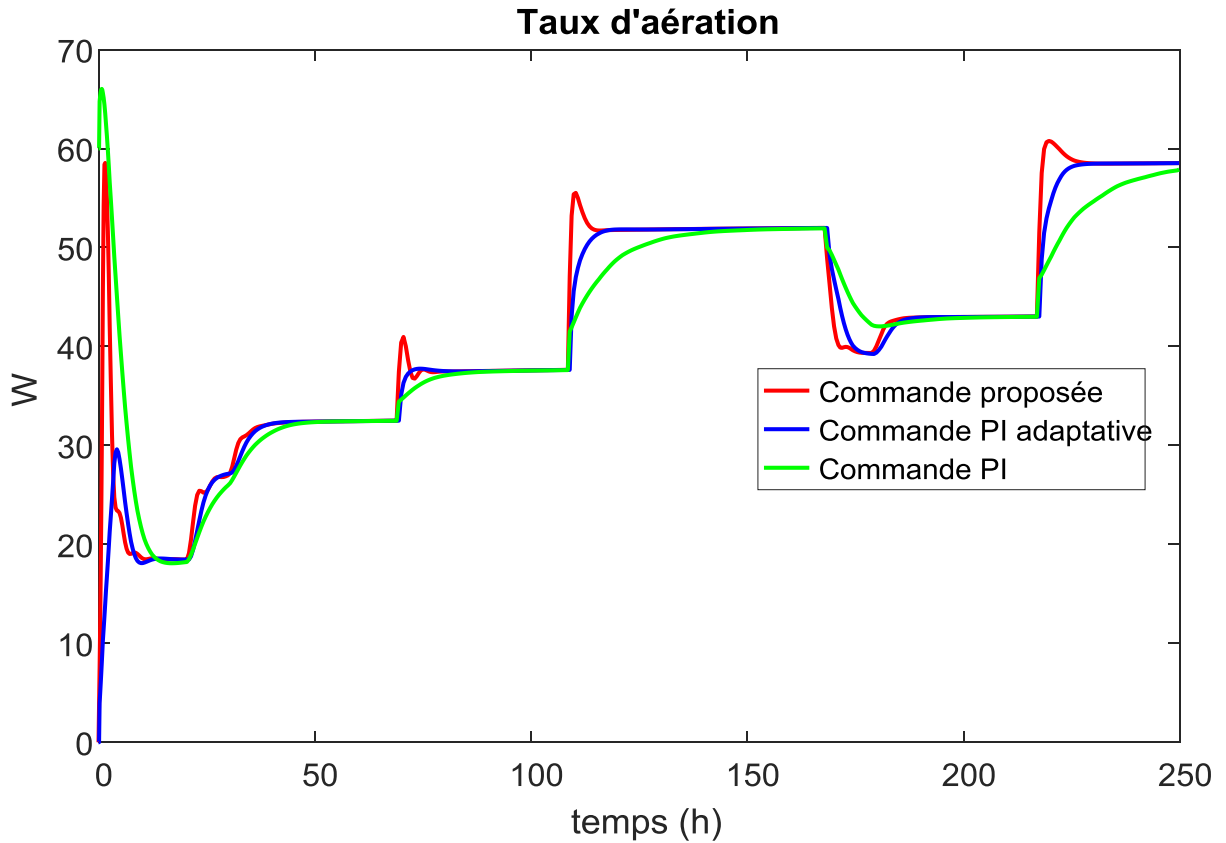


Figure V.21: Le taux d'aération (variable de commande)

La variable de commande W a des valeurs comprises entre 20 et 60 $m^3 \cdot h^{-1}$. Cette augmentation est due au changement du taux de dilution, si le taux de dilution D augmente, on devra augmenter le taux d'aération (pour permettre aux micro-organismes bactériennes de se développer pour pouvoir dégrader les pollutions).

Les figures 22-25 montrent l'évolution des paramètres du modèle neuronal $C_{i,j}$, b_j et $W_{k,j}$ pour la technique de commande proposée.

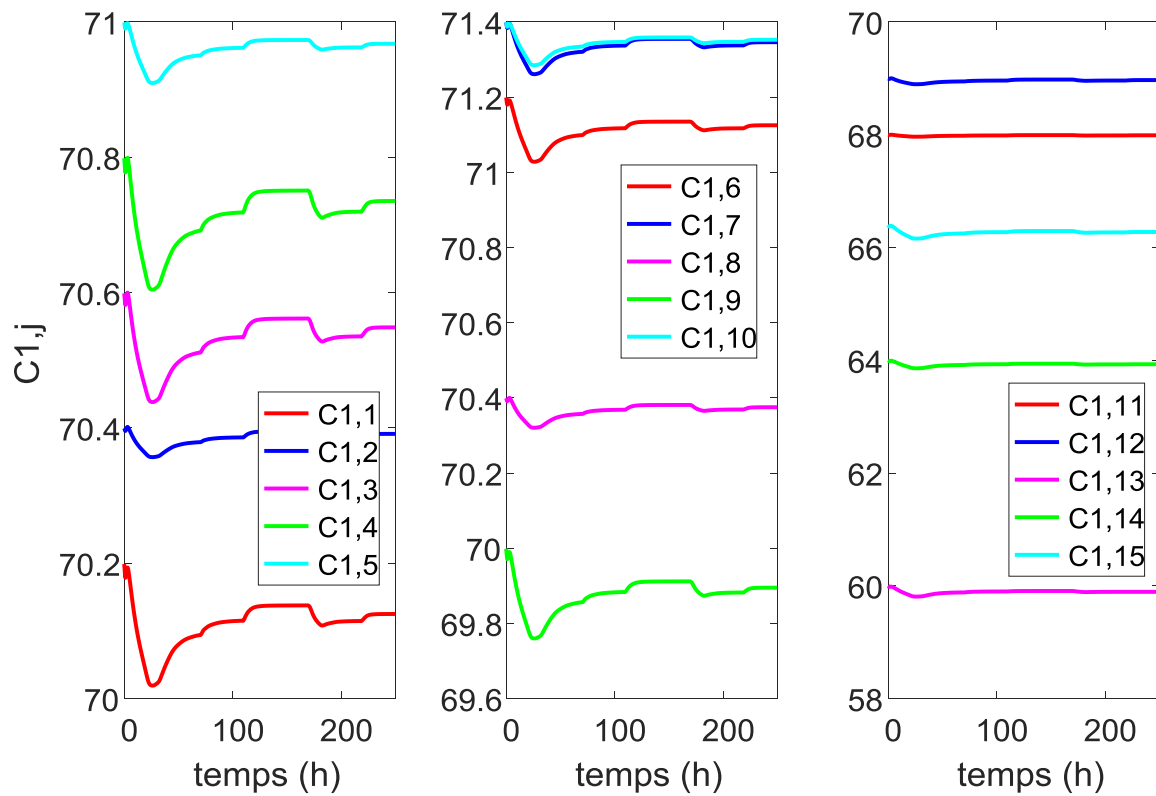


Figure V.22: Adaptation des poids $C_{1,j}$

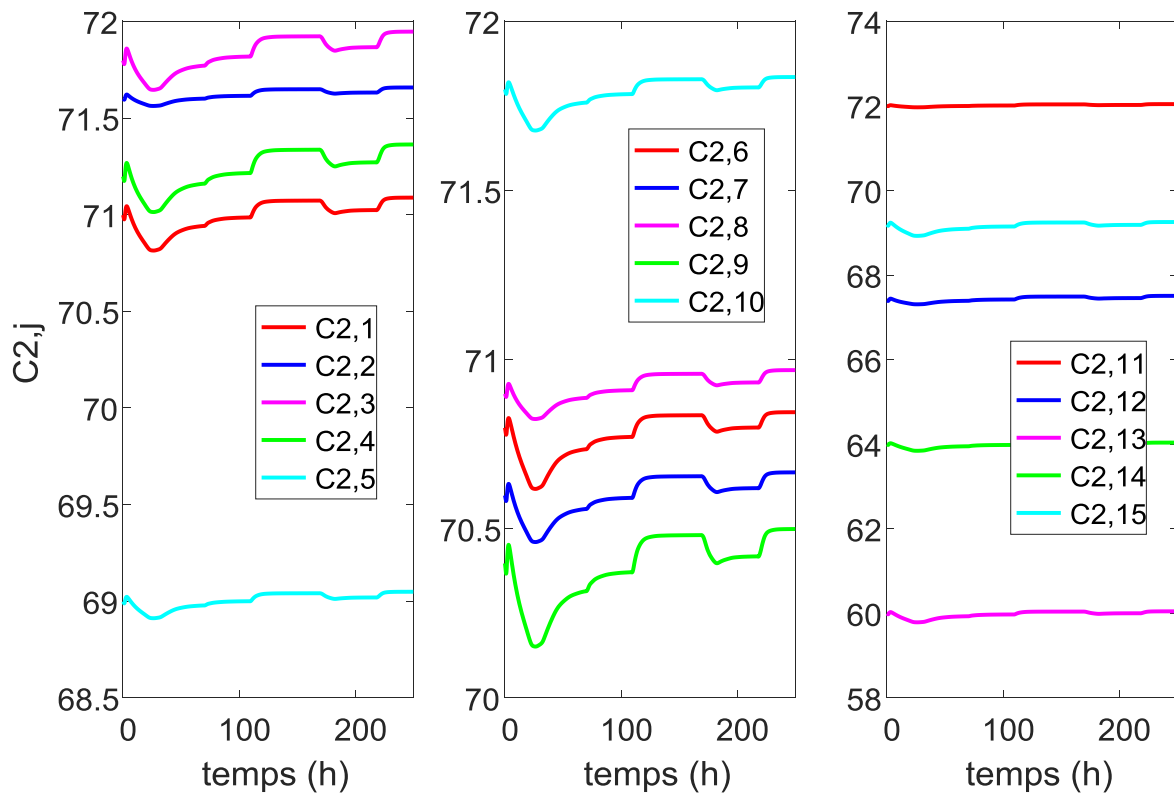


Figure V.23: Adaptation des poids $C_{2,j}$

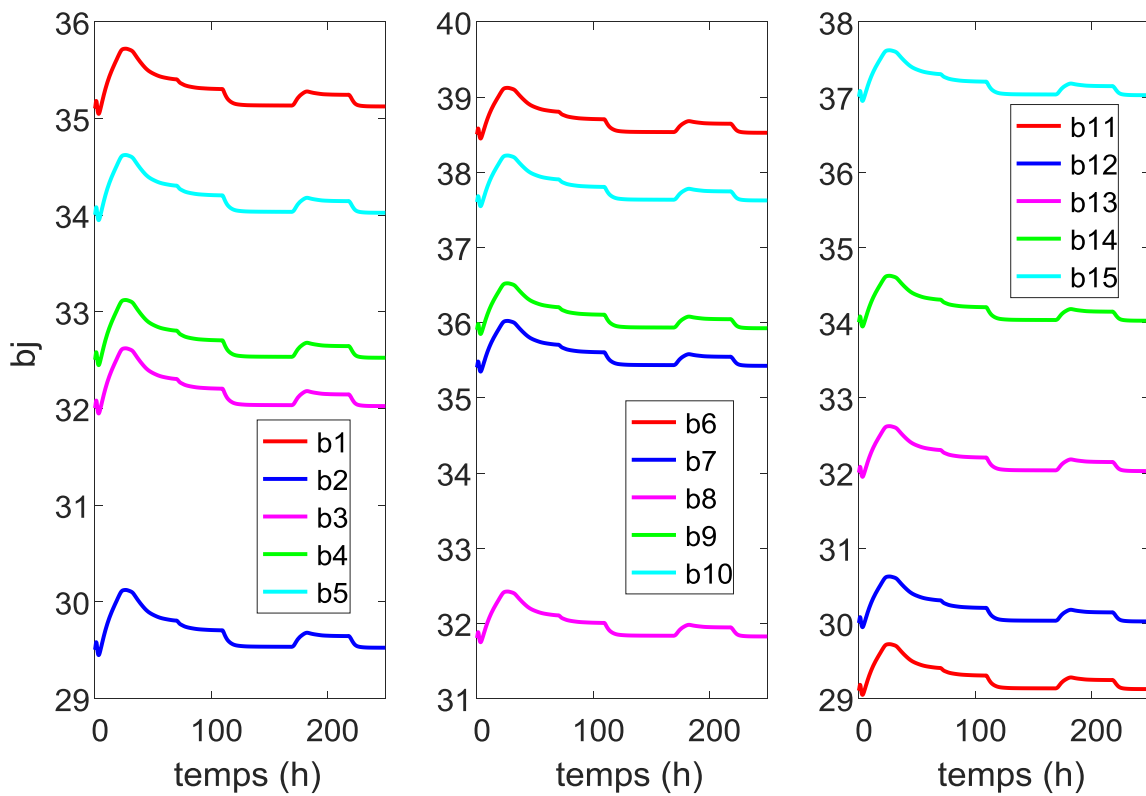


Figure V.24: Adaptation des poids b_j

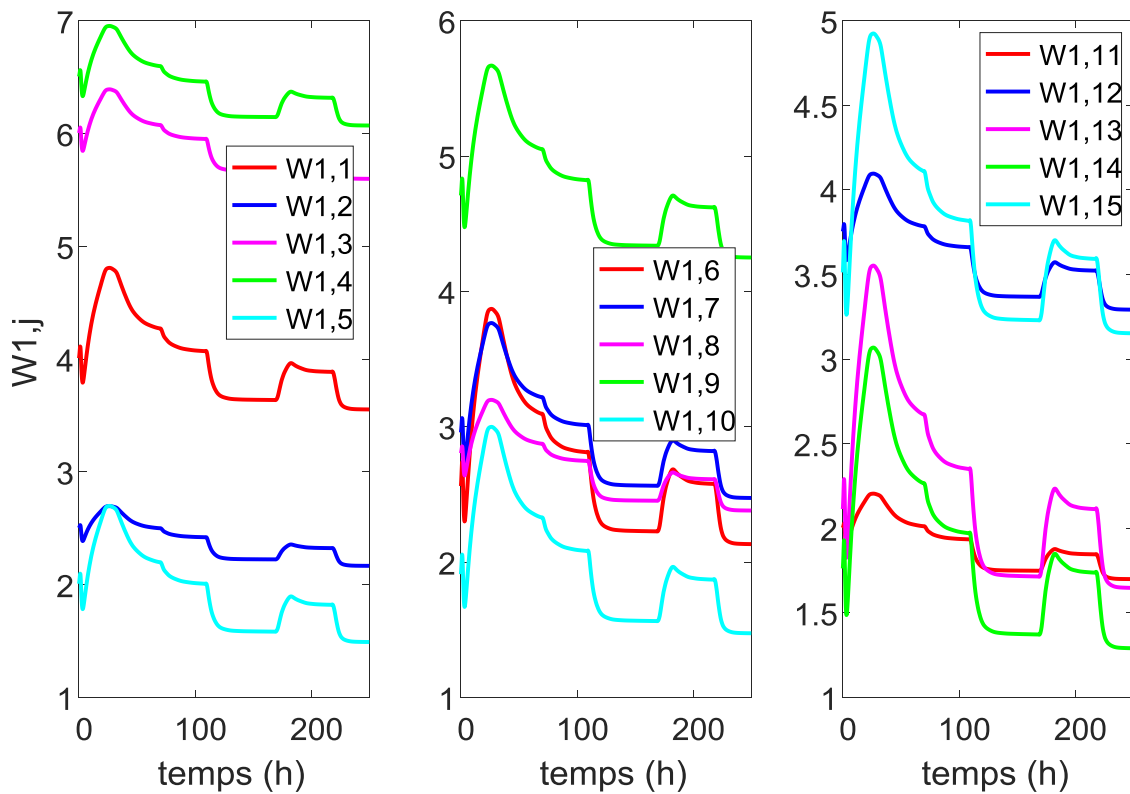


Figure V.25: Adaptation des poids $W_{1,j}$

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

A chaque fois que la référence $DO_{réf}$ change, les paramètres du réseau de neurones RBF $C_{i,j}$, b_j et $W_{k,j}$ changent afin de minimiser la fonction de performance (la somme de l'erreur au carrée E_{RBF} entre la concentration de l'oxygène dissous DO et la sortie du réseau de neurones y_{RBF}).

Les nouvelles valeurs des paramètres du modèle neuronal RBF $C_{i,j}$, b_j et $W_{k,j}$ seront utilisés pour contrôler la concentration de l'oxygène dissous DO .

Les figures 26-29 montrent l'évolution des paramètres du modèle neuronal $C_{i,j}$, b_j et $W_{k,j}$ pour la commande PI adaptative basée sur la technique des réseaux de neurones RBF.

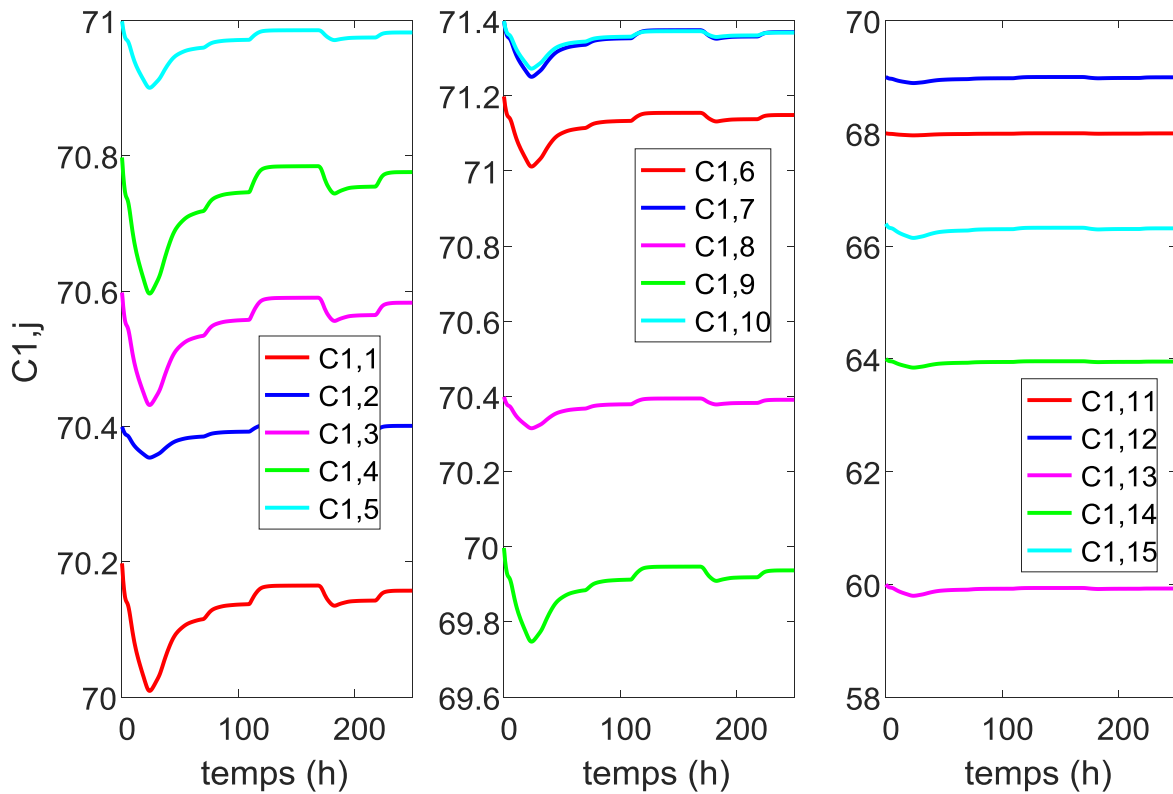


Figure V.26: Adaptation des poids $C_{1,j}$

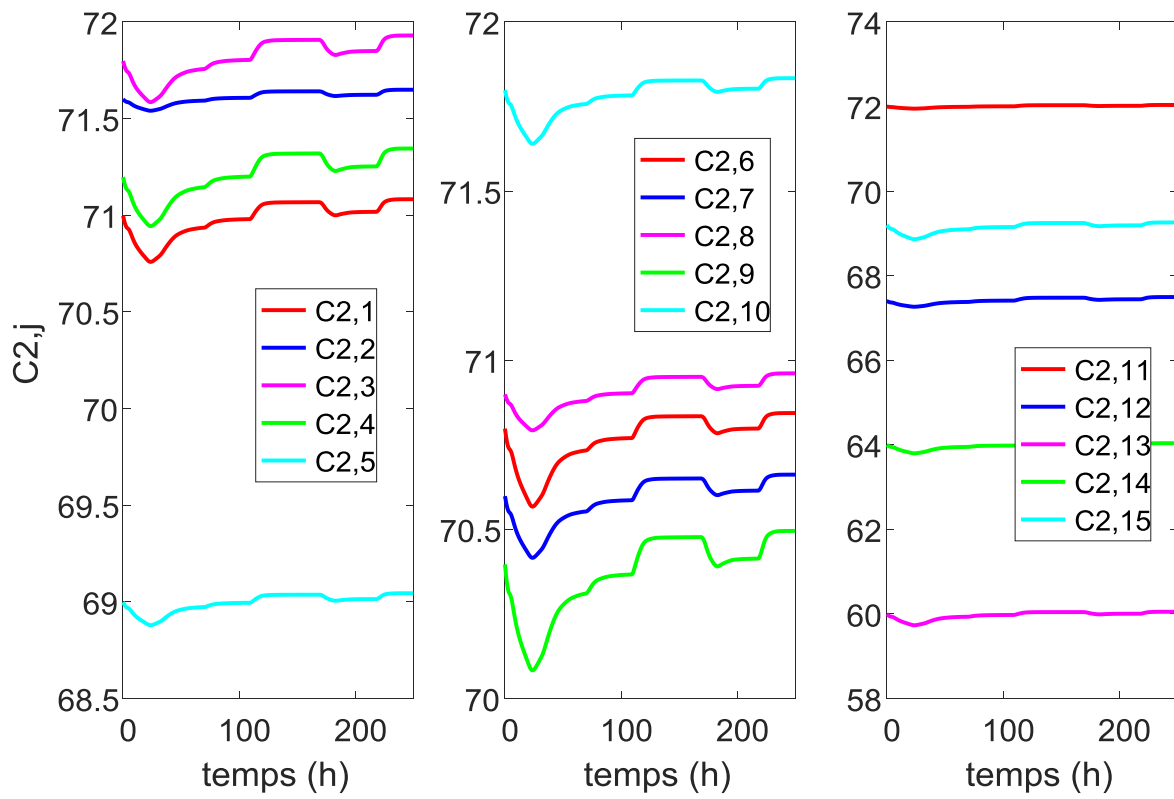


Figure V.27: Adaptation des poids $C_{2,j}$

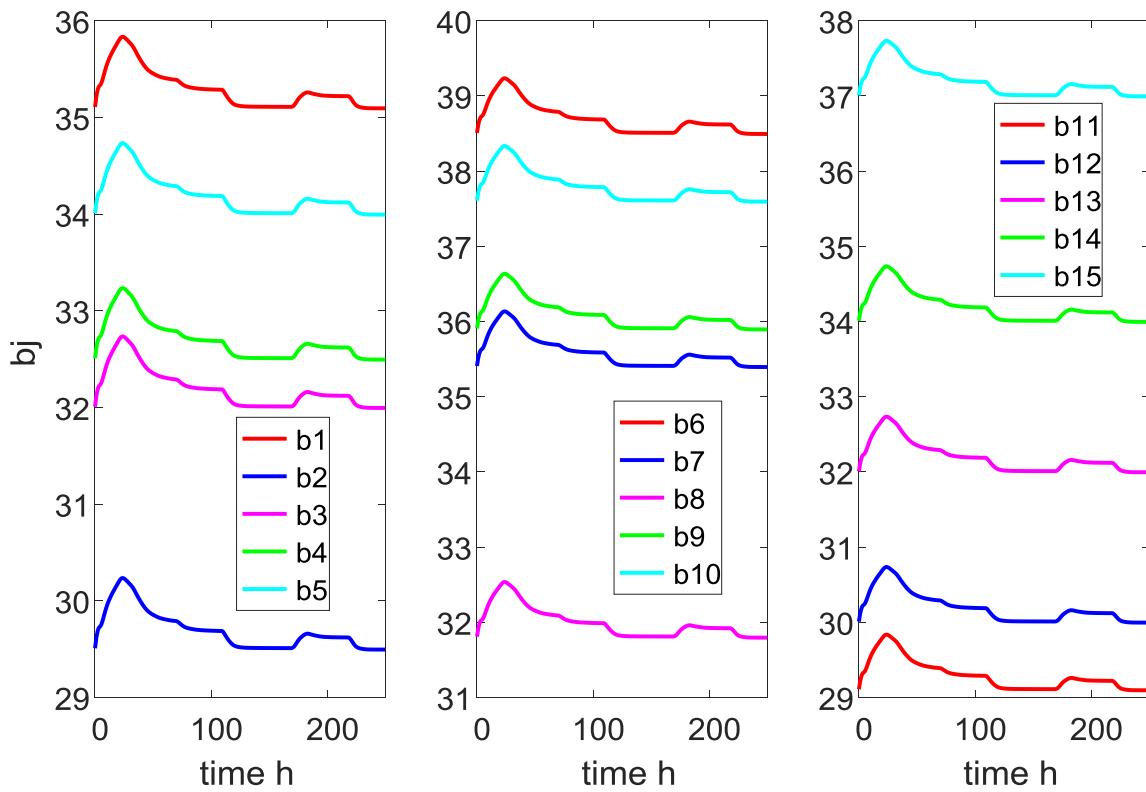


Figure V.28: Adaptation des poids b_j

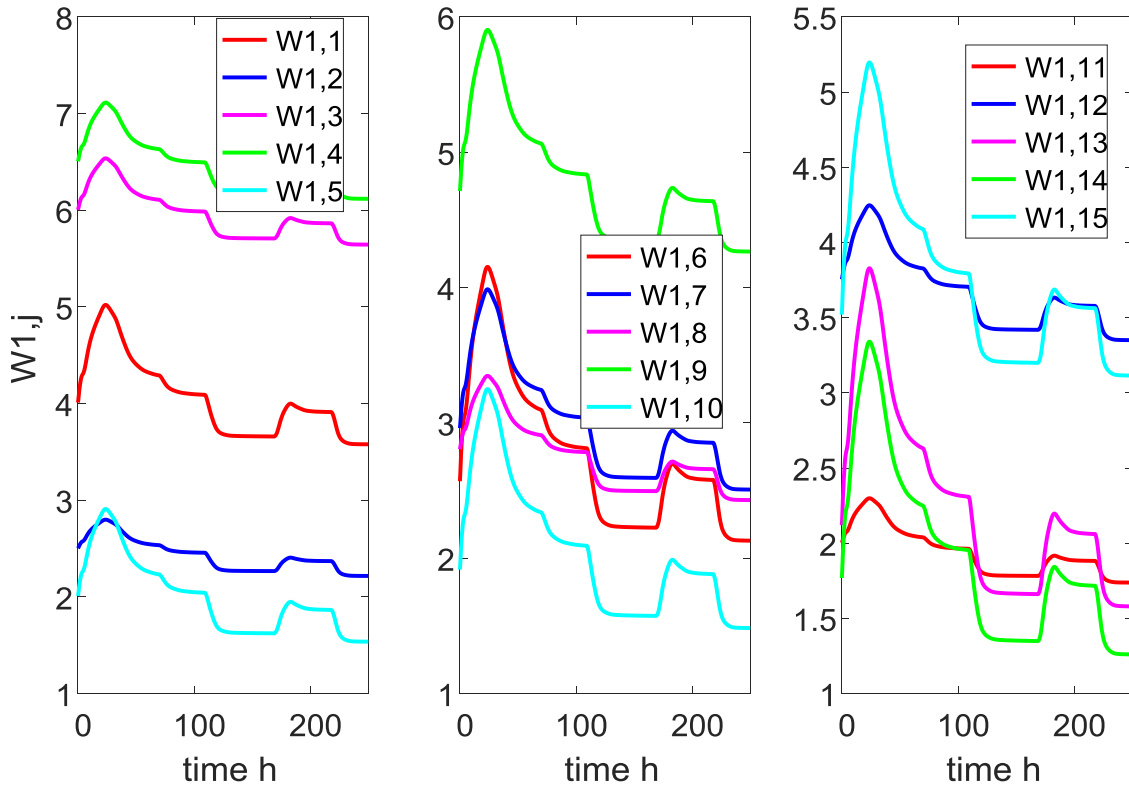


Figure V.29: Adaptation des poids $W_{1,j}$

A chaque changement de la référence $DO_{réf}$, les paramètres du réseau de neurones RBF $C_{i,j}$, b_j et $W_{k,j}$ changent afin de minimiser la fonction de performance.

Les nouvelles valeurs des paramètres du modèle neuronal RBF $C_{i,j}$, b_j et $W_{k,j}$ seront utilisés pour ajuster les paramètres du régulateur PI: k_p et k_i .

Les deux figures ci-dessous 30 et 31 montrent l'évolution des deux paramètres k_p et k_i pour la commande PI adaptative basée sur les réseaux de neurones RBF.

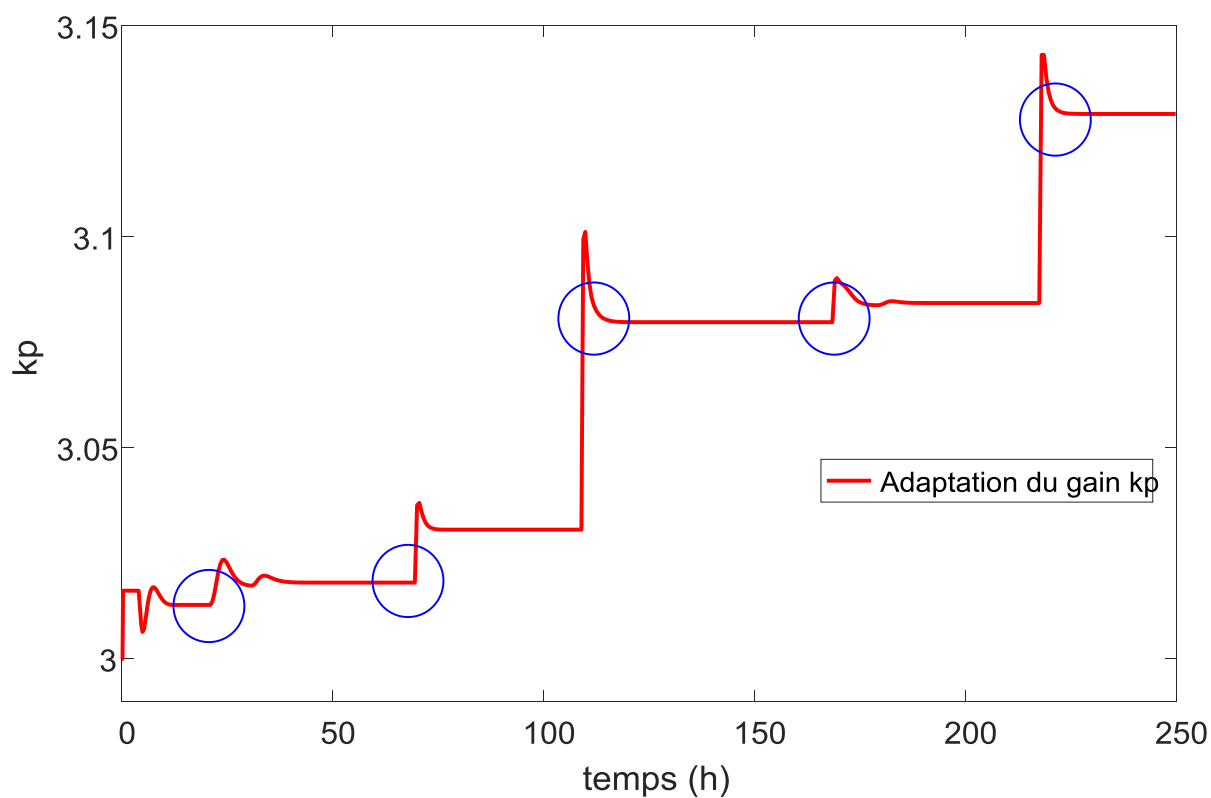


Figure V.30: Adaptation du gain k_p

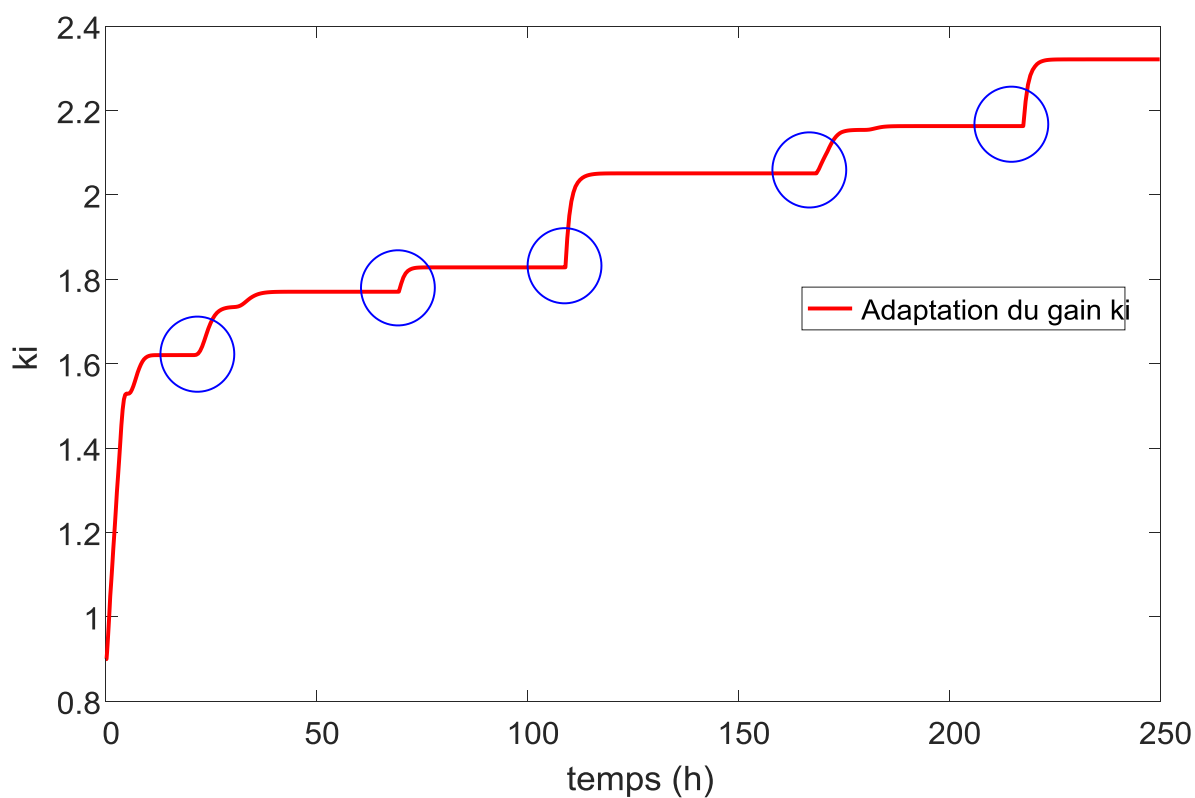


Figure V.31: Adaptation du gain k_i

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

L'ajustement automatique des paramètres du régulateur PI peut être observé dans les figures 30 et 31. Les paramètres k_p et k_i continuent à changer en se basant sur la technique des réseaux de neurones RBF. Les réseaux de neurones sont donc, capables d'ajuster de façon automatique les paramètres du régulateur PI afin d'améliorer l'adaptabilité du contrôleur PI contre les perturbations dans le système (variations du taux de dilution et la concentration du substrat à l'entrée du bioréacteur).

La figure 32 montre clairement que la concentration du substrat est biologiquement dégradée au dessous de la valeur 20 mg.l^{-1} .

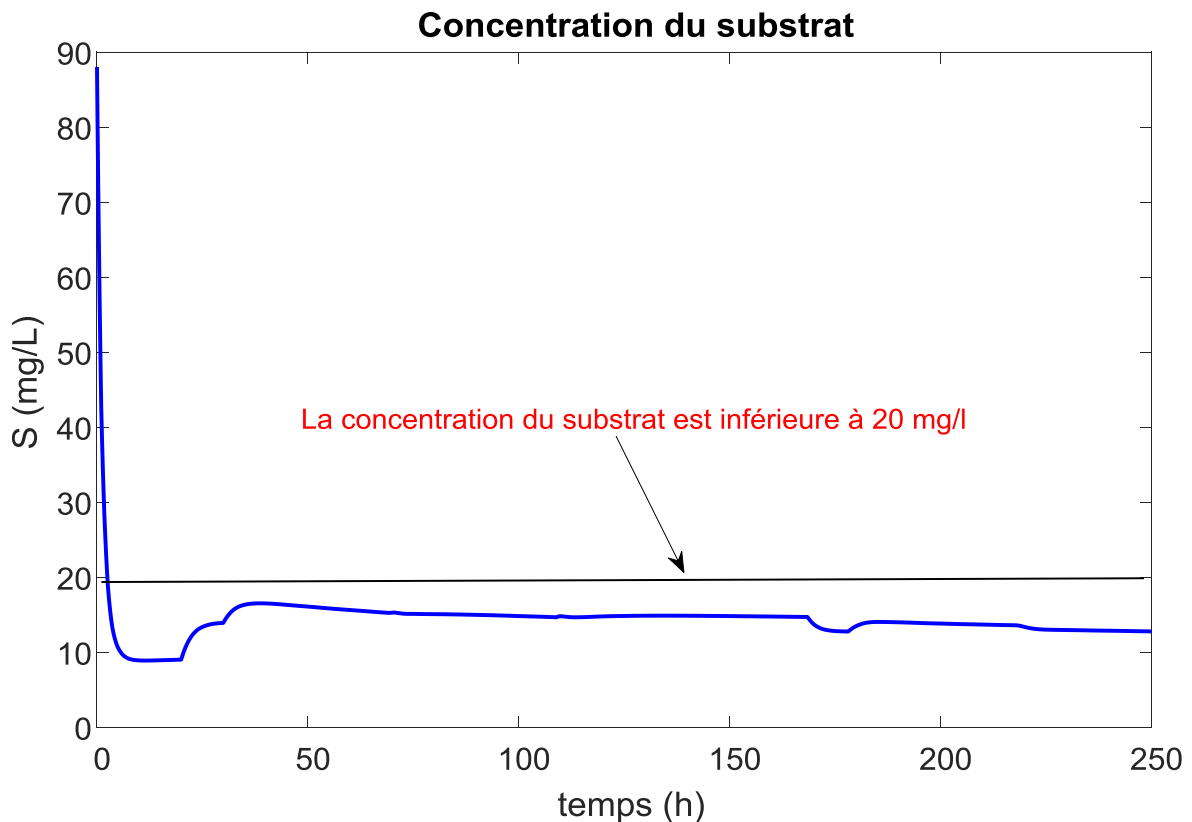


Figure V.32: Concentration du substrat

L'efficacité de la commande proposée est validée en la comparant au contrôleur PI classique et à la commande PI adaptative. Afin de comparer ces techniques de commande, leurs performances doivent être évaluées à l'aide de IAE (Integral of Absolute Error) et de ISE (Integral of Square Error). Ces critères sont calculés comme suit:

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

$$ISE = \int_0^{\infty} e^2(t) dt \quad (5.40)$$

$$ISE = \int_0^{\infty} |e(t)| dt \quad (5.41)$$

$e(t)$ est le signal d'erreur (la référence moins la sortie mesurée). Les résultats de l'évaluation des performances obtenus sont données dans le tableau ci-dessous:

Techniques de commande utilisées	IAE	ISE
Régulateur PI	0.2323	0.2111
La commande PI adaptative par structure neuronale	0.1017	0.0649
La commande proposée	0.0497	0.0326

Tableau V.1: Comparaison entre les différentes techniques de commande

Techniques de commande utilisées	Temps de montée (h)	Temps de réponse (h)	Dépassement (%)
Régulateur PI	16.3976	65.3565	0
La commande PI adaptative par structure neuronale	3.9798	45.9566	0.0211
La commande proposée	1.3658	44.0606	0.5204

Tableau V.2: Valeurs de: temps de montée, temps de réponse et le dépassement

Le tableau V.1 montre la somme de l'erreur absolue et l'erreur au carrée entre la référence (la concentration de l'oxygène dissous désirée $DO_{réf}$) et la concentration de l'oxygène dissous mesurée à la sortie du bioréacteur DO . Les résultats donnés par la commande proposée et la commande PI adaptative sont meilleurs par rapport à la commande PI classique, ce qui montre que ces deux techniques de commande suivent mieux la référence désirée $DO_{réf}$, et qu'elles sont robustes contre les perturbations dans le système générées par le taux de dilution D et la concentration du substrat S_{in} .

Le temps de montée ainsi que le temps de réponse de la commande proposée et la commande PI adaptative sont meilleurs que ceux donnés par la commande PI classique, ce qui montre leur rapidité, sauf que la commande proposée présente un dépassement plus grand que les autres techniques de commande.

Chapitre V: Commande adaptative basée sur la méthode de descente du gradient & la technique des réseaux de neurones

V-7) Conclusion

Dans ce chapitre, une technique de commande basée sur la méthode du gradient et la technique des réseaux de neurones RBF a été établie pour contrôler la concentration de l'oxygène dissous dans un procédé de traitement des eaux usées par boues activées. L'efficacité de cette méthode a été évaluée par une comparaison avec une commande PI adaptative basée sur les réseaux de neurones RBF et un contrôleur PI classique. Une référence variable pour la concentration de l'oxygène dissous a été conçue. Les résultats indiquent que la commande proposée basée sur la méthode de gradient et les réseaux de neurones peut atteindre une meilleure performance.

Conclusion générale

Conclusion générale

Toute espèce, qu'elle soit humaine, animale ou végétale ont besoin de l'eau pour vivre, sinon leur existence serait impossible. Cependant, à cause du développement qu'a connu l'être humain dans plusieurs domaines particulièrement, dans le domaine industriel et agricole, les eaux sont devenues polluées ce qui a conduit à la détérioration de l'environnement, l'écosystème et la santé humaine. Donc, le traitement des eaux polluées est devenu indispensable.

Le procédé le plus utilisé dans le traitement des eaux usées est celui de boues activées. Son principe consiste à ajouter aux eaux usées des bactéries capables de se développer en dégradant les matières polluantes. Ces bactéries ont besoin de l'oxygène dissous dans cette opération de dégradation. Donc, la présence de l'oxygène dissous dans un procédé de traitement des eaux usées par boues activées est indispensable. Donc, maintenir la concentration de l'oxygène dissous dans un procédé de traitement des eaux usées par boues activées permet aux bactéries de mieux dégrader les matières polluantes.

Dans cette thèse, trois techniques de commande ont été utilisées pour contrôler la concentration de l'oxygène dissout dans un bioréacteur de traitement des eaux usées par boues activées: Un régulateur PI classique, une commande PI adaptative d'une structure neuronale et la troisième est une combinaison entre la méthode du gradient et la technique des réseaux de neurones RBF.

* Un régulateur PI: les performances d'un régulateur PI dépendent essentiellement de ses paramètres k_p et k_i , sinon le régulateur ne peut pas assurer des performances désirées. Malheureusement, les résultats donnés par le régulateur PI ne sont pas satisfaisants, surtout quand le système du traitement des eaux usées présente des perturbations (le débit d'eau entrant dans le bioréacteur n'est pas constant). Donc, le régulateur PI n'est pas capable de rejette les perturbations générées par le taux de dilution D et la concentration du substrat S_{in} .

* Une commande PI adaptative d'une structure neuronale: cette technique de commande

Conclusion générale

permet d'améliorer les performances du régulateur PI, c'est-à-dire les paramètres k_p et k_i vont être auto-ajustés en permanence grâce à la technique des réseaux de neurones. Les résultats donnés par cette commande adaptative sont satisfaisants.

* Une commande basée sur la méthode de descente du gradient et les réseaux de neurones: cette technique de commande proposée est une commande adaptative qui permet de changer ses paramètres grâce à la technique des réseaux de neurones. Les résultats donnés par cette technique de commande sont meilleurs que ceux donnés par le régulateur PI classique et la commande PI adaptative.

Annexes

Annexes

Annexes:

Les valeurs des paramètres du modèle sont données dans le tableau ci-dessous.

Description	Paramètres	Unités	Valeurs
Facteur de rendement de la biomasse	Y	-	0.65
Taux de la croissance maximale de la biomasse	μ_{max}	h^{-1}	0.15
Coefficient de demi-saturation de la biomasse	k_S	$mg.l^{-1}$	100
Coefficient de demi-saturation d'oxygène	k_{DO}	$mg.l^{-1}$	2
Concentration maximale de l'oxygène	DO_{max}	$mg.l^{-1}$	10
Coefficient constant	K_0	-	0.5
Taux de transfert de l'oxygène	α	-	0.018
Taux de recyclage de la biomasse	r	-	0.6
Taux d'excès de la biomasse	β	-	0.2
Concentration du substrat entrante	S_{in}	$mg.l^{-1}$	200
Concentration de l'oxygène entrante	DO_{in}	$mg.l^{-1}$	0.5
Coefficient de transfert de l'oxygène	KLa	h^{-1}	-
Taux de dilution	D	h^{-1}	-
Taux d'aération	W	$m^3.h^{-1}$	-

Tableau 1: Paramètres du modèle

Les valeurs initiales des concentrations du modèle sont données dans le tableau ci-dessous.

Concentrations	Symboles	Unités	Valeurs
Concentration de la biomasse	X	$mg.l^{-1}$	88
Concentration du substrat	S	$mg.l^{-1}$	20
Concentration de l'oxygène dissous	DO	$mg.l^{-1}$	4
Concentration de la biomasse recyclée	X_r	$mg.l^{-1}$	320

Tableau 2: Valeurs initiales

Bibliographie

Bibliographie:

- [1] Bo, Y.C.; Zhang, X. (2017). Online adaptive dynamic programming based on echo state networks for dissolved oxygen control. *Appl. Soft Comput.*
- [2] Holenda, B.; Domokos, E.; Rédey, Á.; Fazakas, J. (2008). Dissolved oxygen control of the activated sludge wastewater treatment process using model predictive control. *Comput. Chem. Eng.*, 32, 1270–1278.
- [3] M. Yong, P. Yongzhen, and U. Jeppsson. (2006). Dynamic evaluation of integrated control strategies for enhanced nitrogen removal in activated sludge processes,” *Control Engineering Practice*, vol. 14, no. 11, pp. 1269–1278.
- [4] R. Tzoneva. (2007). Optimal PID control of the dissolved oxygen concentration in the wastewater treatment plant,” in *AFRICON 2007*, Windhoek, South Africa.
- [5] Wahab, N.A.; Katebi, R.; Balderud, J. (2009). Multivariable PID control design for activated sludge process with nitrification and denitrification. *Biochem. Eng. J.* 2009, 45, 239–248.
- [6] Luo, F.; Hoang, B.L.; Tien, D.N.; Nguyen, P.H. (2015). Hybrid PI controller design and hedge algebras for control problem of dissolved oxygen in the wastewater treatment system using activated sludge method. *Int. Res. J. Eng. Technol.* 2015, 2, 733–738.
- [7] Yu, K.; Muhetaer, A.; Wei, L. (2006). Evaluation indexes of sewage stabilization from municipal wastewater treatment plant. *China Water Wastewater* 2016, 5, 93–97.
- [8] Chen, C.S. (2011). Robust self-organizing neural-fuzzy control with uncertainty observer for MIMO nonlinear systems. *IEEE Trans. Fuzzy Syst.* 2011, 19, 694–706.
- [9] Nascu, I.; Vlad, G.; Folea, S.; Buzdugan, T. (2008). Development and application of a PID auto-tuning method to a wastewater treatment process. In *Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2008)*, Cluj-Napoca, Romania, 22–25 May 2008.
- [10] Ye, H.T.; Li, Z.Q.; Luo, W.G. (2013). Dissolved oxygen control of the activated sludge wastewater treatment process using adaptive fuzzy PID control. In *Proceedings of the 32nd Chinese Control Conference (CCC2013)*, Xi'an, China, 26–28 July 2013.

Bibliographie

- [11] Goldar, A.; Revollar, S.; Lamanna, R.; Vega, P. (2014). Neural-MPC for N-removal in activated-sludge plants. In Proceedings of the IEEE European Control Conference (ECC 2014), Strasbourg, France, 24–27 June 2014.
- [12] Goldar, A.; Revollar, S.; Lamanna, R.; Vega, P. (2016). Neural NLMPC schemes for the control of the activated sludge process. In Proceedings of the 11th IFAC Symposium on Dynamics and Control of Process Systems Including Biosystems, Trondheim, Norway, 6–8 June 2016.
- [13] Eckenfelder, W.W. (1991). Gestion des eaux usées urbaines et Industrielles - Caractérisation, Technique d'épuration, aspects économiques. Tech & Doc, Lavoisier.
- [14] Institut de recherche et de sécurité pour la prévention des accidents du travail et des maladies professionnelles, Novembre 2004.
- [15] Alexandre, O, Boutin, C., Duchène, Ph., Lagrange C., Lakel, A., Liénard, A. and Orditz, D., (1998). Filières d'épuration adaptées aux petites collectivités. -Technique et documentation Lavoisier (FNDAE N22), Paris, France.
- [16] CARDOT Claude. (1999). Les traitements de l'eau – Procédés physico-chimiques et biologiques. Paris : Ellipses, 1999, 247 p.
- [17] Bürger, R and Wendland, W. (2001). Sedimentation and suspension flows: Historical perspective and some recent developments. -Journal of Engineering Mathematics Vol 41, pp 101-116.
- [18] Norström, A. (2005). Treatment of domestic wastewater using microbiological processes and hydroponics in Sweden-Department of Biotechnology, Division of Applied Environmental Microbiology, Royal Institute of Technology, AlbaNova University Center, Stockholm, Sweden, Printed at Intellecta DocuSys AB, Nacka, Sweden.
- [19] Berland, J. M., Boutin, C., Molle, P. and Cooper, P. (2001). Procédés extensifs d'épuration des eaux usées. -Office des publications des communautés européennes, Luxemburg.
- [20] Alexandre, O, Boutin, C., Duchène, Ph., Lagrange C., Lakel, A., Liénard, A. and Orditz, D. (1998). Filières d'épuration adaptées aux petites collectivités. -Technique et documentation Lavoisier (FNDAE N22), Paris, France.
- [21] B. Chachuat, N. Roche et M. A. Latifi, (2003). Réduction du modèle ASM 1 pour la

Bibliographie

commande optimale des petites stations d'épuration à boues activées, Revue des sciences de l'eau / Journal of Water Science, Marseille, FRANCE, n1, 2003.

[22] BERKOUCHE. chaabane. (2010). Modélisation et identification pour la simulation et l'observation d'états des bioprocédé. Université Ferhat Abbes-Sétif, Mémoire master 2010.

[23] Maria Elena Hernandez Rojas. (2006). Bioréacteur a membranes immergées pour le traitement d'eaux usées domestiques, influence des conditions biologiques sur les performances du procédé. Thèse doctorat 2006.

[24] Chicone, C. (2006). Equations différentielles ordinaires avec applications (Vol. 34). Springer Science & Business Media.

[25] DEVRIES, Paul L. ; HASBUN, Javier E. (2011). A first course in computational physics. Second edition. Jones and Bartlett Publishers: 2011. p. 215.

[26] Simmons, George F. (1972). Differential Equations with Applications and Historical Notes, New York: McGraw- Hill, LCCN 75173716.

[27] Butcher, John C. (2003). Numerical Methods for Ordinary Differential Equations. New York: John Wiley & Sons. ISBN 978-0-471-96758-3. p. 45.

[28] Hairer, Ernst; Nørsett, Syvert Paul; Wanner, Gerhard. (1993). Solving ordinary differential equations I: Nonstiff problems. Berlin, New York: Springer-Verlag. ISBN 978-3-540- 56670-0. p. 35.

[29] Bradie, Brian. (2006). Une introduction amicale à l'analyse numérique . Upper Saddle River, New Jersey: Pearson Prentice Hall. ISBN 978-0-13-013054-9. pp. 533-655.

[30] Atkinson, Kendall A. (1989). An Introduction to Numerical Analysis (2nd ed.). New York: John Wiley & Sons. ISBN 978-0-471-50023-0. p. 343.

[31] Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, vol.12,2011,p. 2121–2159.

[32] Qian, N. (1999). On the momentum term in gradient descent learning algorithms. Neural Networks : The Official Journal of the International Neural Network Society, 12(1), 145–151.

Bibliographie

- [33] Zeiler, Matthew D. (2012). Adadelta: An adaptive learning rate method. arXiv preprint arXiv:1212.5701, 2012.
- [34] Yurii Nesterov. (1983). A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In Soviet Mathematics Doklady, vol 27, p 372–376, 1983.
- [35] Tieleman, T. and Hinton, G. Lecture 6.5 - RMSProp, COURSERA: Neural Networks for Machine Learning. Technical report, 2012.
- [36] Diederik Kingma and Jimmy Ba. Adam. (2014). A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
- [37] Timothy Dozat. Incorporating Nesterov Momentum into Adam. (2016). In Proceedings of 4th International Conference on Learning Representations, Workshop Track, 2016.
- [38] Chris Dyer: Notes on AdaGrad. <http://www.ark.cs.cmu.edu/cdyer/adagrad.pdf>
- [39] David E. Rumelhart, Hinton, Geoffrey E. et Williams, Ronald J. (1986). Learning representations by backpropagating errors. Nature, vol. 323, no 6088, 8 octobre 1986, p. 533–536.
- [40] Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. International conference on machine learning. pp. 1139–1147.
- [41] Léon Bottou. (1998). Online Learning and Neural Networks, Cambridge University Press. ISBN 978-0-521-65263-6.
- [42] Krzysztof C. Kiwiel. (2001). Convergence and efficiency of subgradient methods for quasiconvex minimization. Mathematical Programming (Series A), Berlin, Heidelberg, Springer, vol. 90, no 1, 2001, p. 1–25. ISSN 0025-5610.
- [43] Buduma, Nikhil; Locascio, Nicholas (2017). Principes de base de l'apprentissage profond: conception d'algorithmes d'intelligence artificielle de nouvelle génération . O'Reilly. p. 21. ISBN 978-1-4919-2558-4.
- [44] Patterson, Josh; Gibson, Adam. (2017). Comprendre les taux d'apprentissage". Deep Learning: l'approche d'un praticien . O'Reilly. 258-263. ISBN 978-1-4919-1425-0.
- [45] McCulloch W. S. Et Pitts W. (1943). A logical calculus of the ideas imminent in nervous activity, Bulletin of Mathematical Biophysics, Vol. 5, 1943, pp. 115-133.

Bibliographie

- [46] Rosenblatt F. (1958). The perceptron: a probabilistic model for information storage and organisation in the brain, *Psych. Rev.*, Vol. 65 1958, pp. 386-408.
- [47] Widrow B. (1960). Modèle ADALINE (Adaptive Linear Element) inspiré du Perceptron.
- [48] Rumelhart, D. E., Hinton, McClelland, and Williams, R. J. (1986). Learning
- [49] R.L. Watrous. (1987). Learning Algorithms for Connectionist Networks: Applied Gradient Methods of Nonlinear Optimization», *IEEE First International Conference on Neural Networks*, 2, pp 619-627.
- [50] Anand, Mehrotra, Mohan et Ranka. (1992). Intelligent Control Using Neural Networks», *IEEE Control System mag*; Avril, pp 11-18.
- [51] C. TOUZET. (1992). Les réseaux de neurones artificiels, introduction au connexionnisme. cours, exercice et travaux pratiques, paris, juillet 1992.
- [52] X. Dong, Y. Zhao, H. R. Karimi, and P. Shi. (2013). Adaptive variable structure fuzzy neural identification and control for a class of MIMO nonlinear system. *Journal of the Franklin Institute*, vol. 350, no. 5, pp. 1221–1247.
- [53] C. C. Chuang, J. T. Jeng, and P. T. Lin. (2004). Annealing robust radial basis function networks for function approximation with outliers. *Neurocomputing*, vol. 56, no. 1–4, pp. 123–139.
- [54] J.K. Sing, D. K. Basu, M. Nasipuri, and M. Kundu. (2007). Face recognition using point symmetry distance-based RBF network. *Applied Soft Computing Journal*, vol. 7, no. 1, pp. 58–70.
- [55] H. Liu, S.Wang, and P. Ouyang. (2006). Fault diagnosis in a hydraulic position servo system using RBF neural network. *Chinese Journal of Aeronautics*, vol. 19, no. 4, pp. 346–353.
- [56] Z. A. Bashir and M. E. El-Hawary. (2009). Applying wavelets to short term load forecasting using PSO-based neural networks. *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 20–27.
- [57] C. Zhang, M. Lin, and M. Tang. (2008). BP neural network optimized with PSO algorithm for daily load forecasting. *Proceedings of the International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII*

Bibliographie

'08), vol. 56, pp. 82–85, December 2008.

[58] Mustafa, M.W., Abdullah J. H. Al Gizi and, Malik A. Alsaedi. (2012). Adaptive PID Controller Based on Real Base Function Network Identification, and Genetic Algorithm in Automatic Voltage Regulator System. *International Journal of Scientific and Research Publications*, 2(11) 2250-3153.

[59] Song, Y. (2014). Intelligent PID controller based on fuzzy logic control and neural network technology for indoor environment quality improvement. Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy.

[60] Nejari, F. et al. (1999). Non-linear multivariable adaptive control of an activated sludge wastewater treatment process. *International Journal of Adaptive Control and Signal Processing*, Vol. 13, pp. 347-365.