

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



ECOLE NATIONALE POLYTECHNIQUE



Département d'Automatique

Spécialité Automatique

Projet de fin d'études

Pour l'obtention du diplôme

D'ingénieur d'état en Automatique

Thème :

*Bioréacteur de fermentation : Applications d'une
commande neuro-floue et d'un PI calculé à base d'un
contrôleur à modèle interne.*

Réalisé et présenté par :

GACEM Hind

Proposé et dirigé par :

Dr : ILLOUL Rachid

Juin 2010

*Laboratoire de Commande des Processus
Ecole Nationale Polytechnique, 10, AV. Hassen Badi, El-Harrach, Algérie*

Dédicaces

Je dédie ce travail à mon père et à ma mère qui m'ont soutenue et aidée tout au long de mes études, ainsi qu'à mon frère et mes sœurs, Salim, Imene et Nadjet.

Je dédie ce travail, aussi à tous mes amis.

Hind

Remerciements

Je remercie, vivement, monsieur Illoul, docteur à l'école nationale polytechnique, d'avoir proposé et dirigé ce travail.

Je remercie également madame Hellal, professeur à l'école nationale polytechnique, pour son aide.

J'adresse mes vifs remerciements à tous mes enseignants, durant les années d'étude à l'ENP.

Je remercie le président et les membres du jury d'avoir accepté d'évaluer ce travail.

Je remercie mes parents, qui m'ont apporté aide et soutien pour la réussite de ce travail.

Chapitre I : Etude biologique de La fermentation

I.1	INTRODUCTION	(5)
I.2	La fermentation	(5)
I.2.1	Définition de la fermentation lactique	(5)
I.2.2	Le lactosérum	(5)
I.2.3	L'acide lactique	(6)
I.2.4	Les bactéries lactiques	(6)
I.2.5	Les facteurs de croissance	(6)
I.2.6	Les facteurs environnementaux influençant la fermentation	(7)
I.2.6.1	Influence du pH sur la croissance	(7)
I.2.6.2	L'influence de la température	(7)
I.2.6.3	L'influence des gaz	(7)
I.2.6.4	L'influence de l'humidité	(7)
I.3	Les Modes de fermentation	(7)
I.3.1	Le mode continu	(8)
I.3.2	Le mode semi-continu	(8)
I.3.3	Le mode discontinu	(9)
I.4	modélisation de la cinétique de croissance bactérienne	(10)
I.4.1	Description de la cinétique de croissance bactérienne	(10)
I.4.2	Les modèles déterministes décrivant la cinétique de croissance microbienne ...	(11)
a)	Modèles primaires de croissance	(11)
b)	Modèles secondaires	(12)
b.1)	La modélisation du taux de croissance	(13)
b.2)	Modélisation du temps de latence	(14)
I.5	bilan de matière	(14)

Chapitre II : Modélisation et étude de stabilité en boucle ouverte du bioréacteur

II.1	Introduction.....	(17)
II.2	Modélisation du bioréacteur de fermentation continue d'acide lactique	(17)
II.2.1	Modélisation de la cinétique de croissance des <i>Streptococcus thermophilus</i>	(17)
II.2.1.1	Modèle primaire de la cinétique de croissance	(17)
II.2.1.2	Modèle secondaire de la cinétique de croissance	(17)
II.2.1.3	Modélisation de la cinétique de production	(18)
II.2.1.4	Modélisation de la vitesse de consommation de substrat	(18)
II.2.2	L'établissement du modèle d'état	(19)
II.2.3	Les paramètres de fermentation	(20)
II.3	Etude de la stabilité du fermenteur continu d'acide lactique	(20)
II.3.1	Détermination des points d'équilibre.....	(21)
II.3.2	Calcul de la sortie en fonction de l'entrée du système à l'équilibre	(22)
II.3.3	Représentation de la sortie en fonction de l'entrée à l'équilibre	(23)
II.3.4	Etude du comportement réel (expérimental) du système à l'équilibre	(24)
II.3.5	Linéarisation du système non linéaire	(24)
II.3.6	Linéarisation du modèle du fermenteur continu d'acide lactique	(25)
II.3.7	Etude de la stabilité du système linéarisé	(25)
II.3.8	Conclusion sur la stabilité du système non linéaire	(26)
II.3.9	Interprétation biologique des résultats de la stabilité	(27)
II.4	Simulation en boucle ouverte du bioréacteur	(27)
II.4.1	Test de stabilité de P1	(28)
II.4.2	Test de stabilité de P2	(29)
II.4.3	Test de stabilité de P3.....	(30)
II.4.4	Test de stabilité de P4	(31)
II.4.5	Etude du degré de stabilité des points P1 et P2	(32)
II.4.6	Etudes des résultats de simulations 1, 2, 3, 4	(33)
II.4.7	Etudes des résultats de simulations 5	(34)

II.5	Conclusions	(35)
------	-------------------	------

Chapitre III : Commande du bioréacteur de fermentation par PI linéaire à base de contrôleur à modèle interne

III.1	Historique	(37)
III.2	Structure de la commande à modèle interne (IMC)	(37)
III.2.1	La structure IMC	(37)
III.2.2	Les avantages de cette structure	(38)
III.3	Structure Feedback équivalente	(39)
III.4	La fonction de transfert de la structure IMC	(40)
III.5	Propriétés de cette structure (IMC)	(41)
III.6	Le contrôleur IMC	(42)
III.6.1	Les conditions sur le contrôleur	(42)
III.6.2	Calcul du contrôleur	(42)
III.6.2.1	Erreur de modélisation	(43)
III.6.2.2	La factorisation de $G_m(s)$	(43)
III.6.2.3	La sélection du filtre $f(s)$	(44)
III.7	Utilisation du contrôleur à modèle interne pour la conception de PID	(46)
III.7.1	Calcul du régulateur PID	(46)
III.7.2	Elimination des paramètres négatifs du PID à base d'IMC	(47)
III.7.3	Exemple : Comparaison entre IMC et PID à base d'IMC, et entre PID à base d'IMC et PID Calculé par d'autres méthodes	(47)
III.8	Application du contrôleur PI basé IMC au processus de fermentation continue d'acide lactique	(48)
III.8.1	Linéarisation du système non linéaire autour des points de fonctionnement choisis	(48)
III.8.2	Calcul du régulateur PI à base d'IMC	(49)
III.8.2.1	Calcul du PI linéaire, pour le premier point d'équilibre	(49)
III.8.2.2	Calcul du PI linéaire, pour le deuxième point d'équilibre ...	(50)
III.8.2.3	Réalisation de la boucle de régulation	(50)
III.9	Simulations	(50)
III.10	Conclusions	(62)

Chapitre IV Réseaux de neurones

IV.1	Introduction	(64)
IV.2	Définition réseau de neurones artificiels	(64)
IV.3	Historique	(64)
IV.4	Applications	(65)
IV.5	Neurone biologique	(66)
IV.6	Réseau de neurones artificiels	(67)
IV.6.1	Neurone formel	(67)
IV.6.2	L'analogie avec le neurone biologique	(68)
IV.6.3	Les différentes fonctions d'activation	(68)
IV.7	L'architecture d'un réseau de neurones	(69)
IV.7.1	Les réseaux monocouches	(69)
IV.7.2	Réseaux multicouches	(69)
IV.7.3	Réseaux de Hopfield	(70)
IV.7.4	Réseaux récurrents	(71)
IV.8	L'apprentissage du réseau de neurones artificiels	(71)
IV.8.1	L'apprentissage supervisé	(71)
IV.8.2	L'apprentissage non supervisé	(72)
IV.8.3	L'optimisation des paramètres.....	(72)
IV.8.3.1	L'optimisation globale	(72)
IV.8.3.2	L'optimisation locale	(72)
IV.8.4	Algorithmes d'optimisations itératives	(73)
IV.8.4.1	Optimisation par descente du gradient	(74)
IV.8.4.1.1	Le choix du gain d'adaptation	(75)
a)	Gain d'adaptation fixe	(75)
b)	Gain d'adaptation adaptatif	(75)
IV.9	L'apprentissage du réseau de neurones	(75)
IV.9.1	Algorithme de rétro-propagation du gradient	(75)
IV.9.2	Dimension de la base d'apprentissage	(79)
IV.9.3	Critère d'arrêt de l'apprentissage	(79)
IV.10	Les inconvénients des réseaux de neurones	(80)

Chapitre V La logique floue

V.1 Introduction.....	(82)
V.2 Historique de la logique floue	(82)
V.3 Les applications de la logique floue	(83)
V.4 Systèmes flous	(83)
V.5 L'univers de discours	(84)
V.6 Les variables linguistiques.....	(84)
V.7 Les valeurs linguistiques	(85)
V.8 Les règles linguistiques	(85)
V.9 L'ensemble flou	(85)
V.10 Les fonctions d'appartenances	(85)
V.11 Les formes des fonctions d'appartenances	(85)
V.12 Les opérateurs de la logique floue	(86)
V.13 Le raisonnement d'un système flou	(87)
V.13.1 La fuzzification	(87)
V.13.2 L'inférence	(87)
V.13.3 La defuzzification	(87)
V.14 La commande par la logique floue	(88)
V.14.1 Types de contrôleurs flous	(88)
V.14.1.1 Le contrôleur de Mamdani	(88)
a)Le régulateur proportionnel flou de type Mamdani	(88)
b) Contrôleur PD flou de type Mamdani.....	(90)
V.14.1.2 Le contrôleur de Sugeno.....	(90)
V.14 .2 Les avantages de la commande floue	(91)
V.14 .3 Les inconvénients de la commande floue	(91)

Chapitre VI

Contrôle neuro-flou du bioréacteur de fermentation

VI.1	Introduction.	(93)
VI.2	Les systèmes neuro-flous	(93)
VI.3	Structure de base du neurone flou	(94)
VI.4	Les différentes architectures du système neuro-flou	(94)
VI.5	L'intérêt des systèmes neuro-flous dans le contrôle.....	(95)
VI.6	Le contrôleur neuro-flou basé sur le modèle de conclusions de Takagi-Sugeno.....	(96)
	VI.6.1 Structure du contrôleur neuro-flou basé sur le modèle de conclusions de Takagi-Sugeno	(96)
	VI.6.2 L'apprentissage du contrôleur neuro-flou se basant sur le modèle de Takagi Sugeno	(98)
VI.7	Application sur le procédé de fermentation continue d'acide lactique	(100)
	VI.7.1 Le choix des variables linguistiques du régulateur	(100)
	VI.7.2 Simulations.....	(101)
	VI.7.2.1 Teste de poursuite du régulateur neuro-flou	(101)
	VI.7.2.2 Teste de rejet de perturbation, sur Sin, du contrôleur neuro-flou	(107)
	VI.7.2.3 Teste du rejet de perturbation, sur la sortie du système, du contrôleur neuro-flou	(108)
	VI.7.2.4 test de robustesse du contrôleur neuro-flou	(110)
	VI.7.2.5 test sur l'effet des valeurs initiales, des paramètres, du contrôleur neuro-flou	(111)
	VI.7.3 Etude des résultats de simulations	(113)
VI.8	Conclusion	(113)

التحكم العصبوني- الغامض وPI الخطي عن طريق الية التحكم بالنموذج الداخلي، تطبيق على مفاعل بيولوجي.

الملخص: الهدف من هذه الدراسة هو التحكم في المفاعل البيولوجية المتواصلة، المعروفة بديناميكتها ألا خطية البطيئة.

في بادئ الأمر قدمنا بعض المفاهيم البيولوجية لعملية التخمر، ثم حصلنا على النموذج الرياضي لهذا التفاعل، حتى نتمكن من إيجاد آلية التحكم و استعمال المحاكاة.

اقترحنا أولاً، آلية تحكم خطية، وهي نسبية تكاملية باستعمال آلية تحكم بنموذج داخلي. بسبب عدم كفاءة هذه الأخيرة ، اقترحنا طريقة ثانية وهي التحكم الهجين العصبوني الغامض ، التي وافقت تماما لمتطلبات التحكم . نتائج المحاكاة فعالية هذه الأخيرة .

كلمات مفتاح : المفاعل البيولوجية، التخمر، حمض الأكتيك ، التحكم الغامض، التحكم العصبوني، خوارزمية الانتشار الخلفي، التحكم الخطي.

Commande neuro-floue et PI linéaire à base d'IMC, appliquées au bioprocédé de fermentation.

Résumé : l'Object de cette étude consiste en la détermination de la loi de commande, réalisant les performances exigées et la robustesse, à appliquer au processus de fermentation continue d'acide lactique.

Ce processus est caractérisé par sa dynamique très lente. Son état peut être entraîné facilement et de manière irréversible vers le lessivage. Il se caractérise, aussi, par l'obtention d'une même valeur de sortie (productivité), pour deux valeurs de commande (taux de dilution) différentes.

Dans ce travail, nous proposons deux lois de commandes, PI linéaire, PI neuro-flou. Pour chacune des lois de commande le test de performances et de robustesse se fait par simulation sur le modèle établi dans le chapitre II.

La première loi de commande, proposée, est le PI linéaire à base d'IMC. C'est la plus simple à calculer et à implémenter pratiquement. Mais les tests d'évaluations montrent que ce contrôleur n'est pas robuste. Il peut entraîner le système vers le lessivage, si le point de fonctionnement est différent du point d'équilibre, pour lequel ce PI a été calculé.

La deuxième loi de commande, calculée indépendamment du modèle du système, et basée sur l'hybridation de deux approches d'intelligence artificielle, est le contrôleur neuro-flou. C'est le plus complexe, et le plus exigeant en termes d'implémentation pratique. Les simulations d'évaluation, de ce contrôleur, montrent que celui-ci réalise les meilleures performances, et qu'il est très robuste aux variations paramétriques. La sortie considérée est la productivité commandée par le taux de dilution.

Mots clés : Fermentation, bioréacteur, PI linéaire, commande à modèle interne, réseaux de neurones, systèmes flous, systèmes neuro-flou.

Neural-fuzzy controller and linear PI based internal model control, applied on bioreactor.

Abstract: In this work, we are interested in the control of continuous bioreactor, in the first step; we try to find the mathematical model of this biological process. Then we proposed to use the linear PI based on the internal model control. The results of simulation have shown that this last one doesn't accomplish the performance, and robustness, and objective control.

The second control law proposed is the PI fuzzy-neural controller, which gave better response, and high degree of performance and robustness. The output considered in the productivity, which is controlled by the dilution rate.

Key words: Bioreactor, fermentation, linear PI, internal model control, neural networks, fuzzy controller, neural-fuzzy controller.

Abréviations

- Q_{in} : Débit d'alimentation en substrat. (l/h)
- Q_{out} : Débit de soutirage du métabolite produit. (l/h)
- V : volume du bioréacteur. (l)
- μ : taux spécifique de croissance. (h^{-1}).
- μ_{max} : Taux spécifique de croissance maximal. (h^{-1}).
- μ_0 : taux de croissance optimale de Monod. (h^{-1})
- K_s : Constante de demi-saturation du modèle de Monod.(g/l)
- K_i : Constante d'inhibition de la croissance. (g/l)
- S_{in} : Concentration d'alimentation en substrat.(g/l)
- λ : temps de latence. (h)
- $dX(t)$: Variation de la concentration de la biomasse à l'instant t.
- $dS(t)$: Variation de la concentration du substrat à l'instant t.
- $dP(t)$: Variation de la concentration du produit à l'instant t.
- $X(t)$: Concentration de la biomasse dans le bioréacteur à l'instant t. (g/l)
- $S(t)$: Concentration du substrat dans le bioréacteur à l'instant t.(g/l)
- $P(t)$: Concentration du produit à l'instant t.(g/l)
- $Q(t)$: Productivité, la quantité de produit soutirée par unité de temps.(g/lh⁻¹)
- $D(t)$: Taux de dilution.(h⁻¹)
- $Q_r(t)$: Référence.
- v_x : Vitesse de croissance.
- v_s : Vitesse de consommation du substrat.
- v_p : Vitesse de production.
- α, β : Constantes.
- Y_{xs} : Rendement de conversion substrat en biomasse.
- G_m : Modèle nominale du système.
- $Q(s)$: Contrôleur à modèle interne.
- $C(s)$: Contrôleur de la structure Feedback classique.

Abréviations

d : Perturbation sur la sortie.

R : référence.

$T(s)$: Fonction de sensibilité inverse.

$S(s)$: Fonction de sensibilité.

a_i^j : Sortie du neurone i de la couche j .

α_k^l : Coefficients de la variable l correspondant à la conclusion k .

c_i : Centre de la fonction d'appartenance i .

σ_i : largeur de la fonction d'appartenance i .

f_j^i : Fonction d'activation du neurone j de la couche i .

β : Taux d'apprentissage du contrôleur neuro-flou.

T : Température.

Introduction générale

Introduction générale

Le développement rapide des industries laitières, conduit à une production de plus en plus importante de lactosérum. Les quantités importantes de ce sous-produit, rejetées dans la nature, affectent, considérablement, l'environnement.

Dans le but de valoriser le lactosérum, la fermentation de ce dernier, en acide lactique, est l'une des solutions idéales, qui intéressent les industriels, vu la diversité et l'importance des applications industrielles de l'acide lactique.

Pour une production, en quantités importantes, d'acide lactique, le processus de fermentation doit être mené en mode continu. La forte sensibilité des paramètres de ce système envers les variations des conditions environnementales, et les perturbations sur le débit et la concentration du substrat d'alimentations, entraînent le processus de fermentation vers le lessivage.

Dans ce cas, une loi de commande robuste, qui réalise les performances, est nécessaire pour conduire la sortie de ce bioprocédé (la quantité d'acide lactique en sortie), vers la valeur désirée.

Une Etude bibliographique, décrivant le procédé biologique de fermentation, est présentée dans le chapitre 1, incluant les définitions, les descriptions de la fermentation, les éléments principaux intervenants dans cette réaction, les différents facteurs influençant et les différents modes de fermentations.

Dans ce chapitre, nous présentons, aussi, les modèles les plus importants établis par les biologistes décrivant les cinétiques du procédé de fermentation. Nous terminons par la présentation d'un modèle d'état généralisé du processus de fermentation, établi à partir des équations des bilans de matières pour un bioréacteur supposé parfaitement mélangé.

Le deuxième chapitre, est divisé en quatre parties. La première partie, concerne la modélisation du procédé de fermentation d'acide lactique en mode continu. L'étude du comportement et de la stabilité des points d'équilibre en boucle ouverte est réalisée dans la deuxième partie. Les simulations et conclusions sur le comportement du procédé de fermentations sont présentées dans les parties 3 et 4 de ce chapitre.

Le troisième chapitre est divisé en deux parties. Dans la première partie, nous présentons Les bases théoriques de conception du contrôleur à modèle interne, et le calcul du contrôleur PID linéaire équivalent.

Dans la deuxième partie nous calculons le régulateur PI à base du contrôleur à modèle interne et nous l'appliquons au modèle du bioréacteur obtenu dans la première partie du chapitre II.

Les chapitres 4 et 5 sont consacrés à l'étude théorique de deux approches d'intelligences artificielles.

Un bref historique et les notions de bases des réseaux de neurones, ainsi que l'algorithme d'apprentissage Backpropagation, sont présentés dans le quatrième chapitre, Que nous terminons par l'énumération des inconvénients et des avantages de cette approche.

Quelques notions théoriques de la logique floue, le raisonnement des systèmes flous, et quelques types de contrôleurs flous, sont présentés dans le cinquième chapitre.

Nous terminons ce chapitre par une énumération des avantages et inconvénients du contrôle flou.

Dans la première partie du chapitre VI, nous abordons le concept des systèmes neuro-flous résultants de l'hybridation des approches étudiées dans les deux chapitres précédents. Alors que dans la deuxième partie, nous calculons le régulateur neuro-flou de type Sugeno, que nous appliquons à notre procédé de fermentation.

On termine, le chapitre, par les discussions des résultats de simulations, et les conclusions sur ce contrôleur.

Nous terminons par une conclusion générale, sur le comportement du procédé de fermentation considéré, et sur l'étude comparative des lois de commandes calculées et appliquées à ce procédé.

Enfin, nous proposons des perspectives pour des travaux futures.

Chapitre I
Etude biologique de
La fermentation
Lactique

I.1 INTRODUCTION

Le développement rapide des industries fromagères, a conduit vers une production de plus en plus importante de lactosérum, considéré comme facteur important de pollution. Vu sa richesse en sucre fermentescible, les industriels se sont intéressés à trouver un moyen de valorisation de ce déchet. La solution que les chercheurs ont proposée est la fermentation de ce sous produit fromager, en acide lactique par l'intermédiaire de microorganismes.

Cette solution n'est pas nouvelle, car la fermentation existe depuis longtemps, mais le plus important est de trouver les moyens de l'optimiser, pour avoir un rendement de valorisation maximal du lactosérum, et une production importante de substance à valeur ajoutée.

La meilleure solution est de réaliser cette fermentation en continu, dans des bioréacteurs automatisés. Ceci permet d'avoir de plus importantes quantités de produits. Alors une étude, sur le métabolisme de fermentation et des différents modèles, réalisée par les biologistes est devenue nécessaire, dans le but de déterminer la loi de commande, qui permet d'avoir les caractéristiques sur le produit de fermentation imposées par l'utilisateur.

Dans le présent chapitre nous allons présenter les principaux constituants de fermentation d'acide lactique, les facteurs influençant ce métabolisme, les différents modes de fermentations, les principaux modèles établis pour la modélisation de la croissance microbienne, et la modélisation de la dynamique de fermentation (Le bilan des matières).

I.2 La fermentation :

I.2.1 Définition de la fermentation lactique :

La fermentation est l'application du métabolisme microbien, pour transformer une matière en produits à valeur ajoutée. Il suffit simplement de connaître le microorganisme adapté, de contrôler son métabolisme et sa croissance, et d'être en mesure de l'utiliser à grande échelle.

La fermentation lactique peut être définie comme un procédé de fermentation dans lequel interviennent un groupe des bactéries lactiques, qui croissent sous des conditions anaérobies et utilisent les sources de carbones pour produire de l'acide lactique, comme seul ou majeur acide organique.

I.2.2 Le lactosérum :

C'est un sous produit de l'industrie alimentaire, fromagerie et caséinerie, considéré comme un excellent milieu de culture, pour certains microorganismes. Utilisé principalement pour l'élevage industriel porcin, incorporé dans la ration des vaches laitières, (Agnes, 1986).

Vu les quantités énormes de lactosérum produites dont la plus grande partie est rejetée dans la nature, constituant un facteur de pollution (de 4 à 5 kg de lactosérum sont rejetés pour 1kg de fromage produit par l'unité ORLAC de Boudouaou, voir 6000 l/J de lactosérum rejetés), l'utilisation du lactosérum comme substrat de fermentation pour la production de produits de haute valeur est d'une importance capitale.

I.2.3 L'acide lactique :

L'acide lactique est un acide organique naturel intéressant. Il se présente sous deux formes optiquement active L(+) et D(-).

Cet acide trouve diverses applications courantes dans les industries pharmaceutiques, chimiques, alimentaires, et dans la production de polymères biodégradables.

Il est utilisé :

- Comme acidulant ou conservateur d'aliments, grâce à son léger goût acide qui n'altère pas les saveurs naturelles aromatiques (Messens et Vuyst, 2002).
- pour régler le pH de certains aliments.
- comme émulsifiant en boulangerie (Jarry, 1994).
- pour empêcher les bactéries pathogènes sur le bœuf, et la volaille, pendant le processus d'abattage.
- sous forme de polymères ou copolymères lactiques-glycoliques pour la libération contrôlée des médicaments dans des fibres creuses biodégradables (Jarry, 1994).

Le lactate de potassium, le lactate de sodium et le lactate de calcium sont des sels neutralisés de l'acide lactique, utilisés comme additifs alimentaires pour une plus longue conservation de produits frais, tout en réduisant la quantité d'agents conservateurs chimiques utilisés pour la conservation (chlorure de sodium).

I.2.4 les bactéries lactiques :

Les bactéries lactiques sont les microorganismes utilisés pour la fermentation d'acide lactique, qui nécessitent, des besoins complexes en facteurs de croissance. Parmi ces microorganismes on distingue les homolactiques (production d'un seul métabolite) et hétérolactiques (plusieurs métabolites), (Orla-Jensen .1919). On peut les classer suivant leur morphologie : les cocci (sphérique) les bacilles (bâtonnets).(Lenoir et Al .1992)

I.2.5 Les facteurs de croissance :

Toutes les bactéries ont besoin d'eau, d'une source d'énergie, d'une source de carbone, d'une source d'azote et d'éléments minéraux. Certaines bactéries qualifiées d'auxotrophes nécessitent, en plus des besoins élémentaires, la présence de facteurs de croissance.

Les facteurs de croissance sont des bases puriques ou pyrimidiques, des acides gras, des acides aminés, et des vitamines (coenzymes, précurseurs de coenzymes, groupements prosthétiques de diverses enzymes).

Les vitamines :

Les vitamines B1, B2, B3, B4, B5, B6, B8 sont indispensables pour la croissance, ils jouent le rôle de coenzyme et stimule fortement la croissance cellulaire.

Les ions métalliques :

Mg^{+2} joue le rôle d'activateur enzymatique intervenant dans les différentes phases du métabolisme réactionnel : division cellulaire, stabilisation d'acide nucléique ADN et ARN (Webb .1948, 1949, 1951a, b), peptide hydrolyses (Johnson et Berger .1942). Il est essentiel pour la phosphokynase (Shankar et Bard .1955).

L'ajout de ce cation, au milieu, stimule fortement la croissance des *Streptococcus thermophilus* et *Streptococcus lactis*. Et une insuffisance de Mg^{+2} retarde la division cellulaire (Webb,1951).

Ledesma et al (1977), ont prouvé qu'en absence de Mn^{++} ou Fe^{++} il n'y aura pas de croissance.

1.2.6 Les facteurs environnementaux influençant la fermentation : Les facteurs influençant la croissance sont classés en deux groupes :

Facteurs d'environnement biotique : compétition entre différentes espèces microbiennes, production d'inhibiteurs spécifiques, altération du milieu, et état physiologique de l'inoculum, etc.

Facteurs d'environnement abiotique : température, taux d'oxygène, structure de l'aliment, substance antimicrobienne, nitrite, acides organique, etc.

I.2.6.1 Influence du pH sur la croissance : Le pH affecte considérablement la croissance bactérienne. Chaque espèce se développe dans une gamme définie de pH et possède un pH optimum de croissance.

La majorité des bactéries et des protozoaires est neutrophile ($5,5 < \text{pH optimum} < 8$). Une forte variation de pH endommage les bactéries, en détruisant leurs membranes plasmiques, en inhibant l'activité des enzymes et en détruisant les protéines membranaires de transport.

I.2.6.2 L'influence de la température : La croissance des microorganismes, augmente avec l'augmentation de la température, jusqu'à la température optimale. À partir de cette valeur une augmentation de cette dernière inhibe la croissance.

I.2.6.3 L'influence des gaz : Certains gaz sont nécessaires à la croissance, alors que d'autres l'inhibent. Par exemple, l'oxygène inhibe la croissance des bactéries anaérobies.

I.2.6.4 L'influence de l'humidité : La majorité des microorganismes, nécessite une certaine valeur d'humidité, pour se développer.

I.3 Les Modes de fermentation :

Bioréacteur: terme générique dérivé du vocabulaire du génie chimique désignant des récipients dans lesquels se déroule une réaction (bio)-chimique.

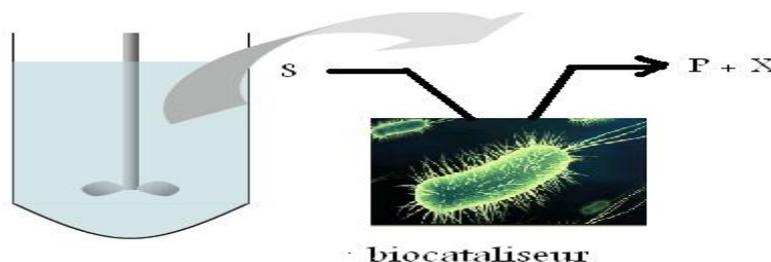


Figure (1.1) : réacteur biologique

S : substrat (contenant la source de carbone) ;
 P : métabolite de fermentation (produit) ;
 X : biomasse formée.

Les modes de fonctionnement des bioréacteurs sont généralement caractérisés par les échanges liquides, c'est-à-dire par le type d'alimentation en substrat et de soutirage du réacteur. On distingue trois modes principaux :

I.3.1 Le mode continu :

Caractérisé par une alimentation continue en substrat : le substrat(en solution) est introduit tout au long de la réaction, et un soutirage continu de son contenu.



Figure (1.2) : réacteur en mode continu

Avantage :

La transformation des substrats (production du produit) peut se poursuivre indéfiniment, et l'appareillage est utilisé à plein temps (intérêt économique). Il permet une production importante dans des réacteurs de taille réduite.

Inconvénients :

- Un peu difficile à mettre en œuvre (par rapport aux autres modes).
- Un risque accru de contamination, parce qu'un fonctionnement sur les temps longs peut engendrer la modification du matériel biologique (mutation de microorganisme, dénaturation, donc inactivation des protéines, ...)

I.3.2 Le mode semi-continu :

Le bioréacteur fonctionnant en semi-continu (ou en batch alimenté ou *Fed batch*) est partiellement alimenté : à partir d'un volume initial préalablement ensemencé, le substrat est introduit tout au long de la réaction, au fur et à mesure des besoins constatés des microorganismes, mais sans qu'il y ait soutirage du milieu



Figure (1.3): réacteur en mode semi continu

Avantage :

- permet d'augmenter le volume au fur et à mesure de la croissance de la biomasse, pas d'inhibition due à la concentration de substrat.

Inconvénients :

- débit contrôlé, à ne dépasser (risque de débordement).
- Arrêt de l'alimentation, lorsque le volume atteint une certaine valeur, ce qui limite son utilisation industrielle.
- Inhibition due à l'accumulation du métabolite et/ou celle due à l'encombrement cellulaire.

I.3.3 Le mode discontinu

Traitement par lot ("*batch*" en anglais), pour ce mode, le substrat (solide ou liquide) est introduit au démarrage de la réaction. Aucun apport ni prélèvement n'est, par la suite, réalisé, et le volume reste constant tout au long de la réaction. Seules les conditions d'environnement du milieu fermentaire subissent une action de l'opérateur (pH, température, vitesse d'agitation, débit d'aération,...) $Q_{in} = 0, Q_{out} = 0$ et V constant. (1.3)

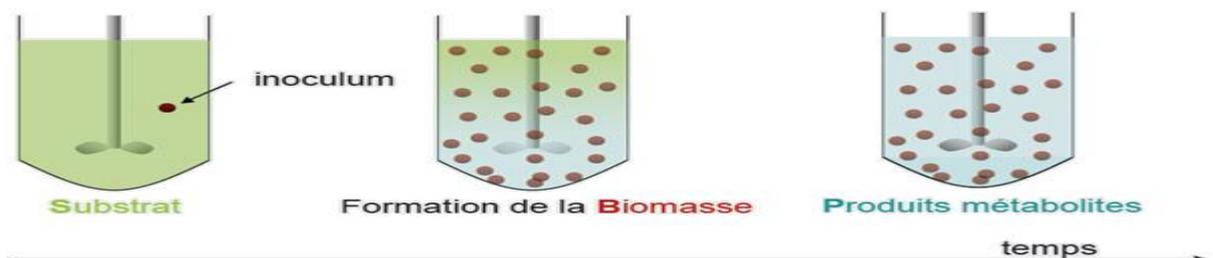


Figure (1.4) : réacteur en mode discontinu

Avantages :

- simple à mettre en œuvre (les seules actions se feront sur les variables d'environnement) et de moindre coût.
- Les risques de contamination de culture sont minimes.

Inconvénient :

- un apport élevé en substrat, au départ, inhibe, en générale, la croissance et rend le temps de latence très important, du fait qu'on introduit toute la quantité nécessaire en éléments nutritifs pour toute la période supposée de la réaction.

Remarque : Ce mode de culture est plus facile à mettre en œuvre et à conduire, mais les productivités obtenues sont plus faibles que celle obtenues par un mode continu de culture.

I.4 modélisation de la cinétique de croissance bactérienne :

La notion de croissance bactérienne recouvre deux aspects : la croissance de la cellule bactérienne (taille, masse, volume), et le phénomène de multiplication cellulaire. Pour simplifier, on assimile souvent la croissance à la multiplication cellulaire.

L'étude de la croissance bactérienne consiste en la détermination des paramètres de croissance pour une souche bactérienne donnée.

Dans un milieu liquide bien mélangé et lorsque les conditions de milieu sont favorables (présence de nutriments, température, activité de l'eau, pH, absence d'antiseptiques ou d'antibiotiques...), la croissance microbienne est homogène. Elle est décrite comme suit :

I.4.1 Description de la cinétique de croissance bactérienne:

Buchanan (1918), a donné la première description des différentes phases de croissance des cultures bactériennes, Ces phases sont caractérisées par certaines valeurs ou variations de la vitesse de multiplication de la culture. On reconnaît classiquement les sept phases successives suivantes, figure (1.5):

1. la phase stationnaire initiale (phase de latence) : Elle est caractérisée par une vitesse de multiplication nulle. Cette phase correspond à une période de biosynthèses et d'adaptation de l'inoculum à son nouvel environnement de croissance avant d'entrer en division (Robinson *et al*, 1998).
2. la phase transitoire entre la latence et la croissance exponentielle.
3. la phase de croissance exponentielle, caractérisée par une vitesse de croissance maximale, et constante, μ_{max} .
4. la phase de diminution de la vitesse de croissance qui, progressivement, devient nulle.
5. la phase stationnaire correspond à un arrêt de la croissance. Le ralentissement, puis l'arrêt de la croissance, sont dus à l'épuisement d'une substance nutritive ou à toutes autres modifications des propriétés du milieu de culture le rendant impropre à la croissance des microorganismes.
6. la phase du début de décroissance.
7. la phase de décroissance.

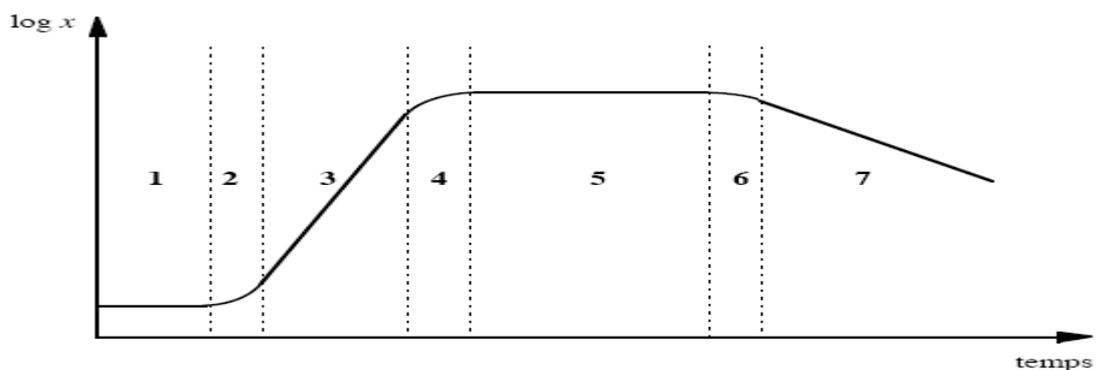


Figure (1.5) : Courbe de croissance d'une culture bactérienne et ses différentes phases (Buchanan, 1918).

I.4.2 Les modèles déterministes décrivant la cinétique de croissance microbienne :

Whiting et Buchanan (1993), ont proposé les classes de modèles suivants:

a) Modèles primaires de croissance :

Ces modèles sont valables pour les conditions environnementales fixes. Ils représentent l'évolution du nombre de micro-organismes en fonction du temps :

$$X = f(t, \theta)$$

X : concentration en cellules

t : temps

$\theta = \{ \text{temps de latence } \lambda, \text{ taux spécifique maximal de croissance } \mu_{\max} \}$, les principaux paramètres de croissance, définis par Jacques Monod en 1942.

Etablissement du modèle de croissance exponentielle :

Si l'on s'intéresse à des cellules, isolées, comme c'est le cas quasi systématiquement; et si leur tailles, masses et volumes moyens sont invariables et négligeables lors de la phase de croissance considérée, Alors l'étude de la croissance revient à l'évaluation de la biomasse bactérienne et à l'évaluation du nombre de bactéries (il existe alors une relation entre ce nombre et la biomasse). Dans ce cas, l'étude de la croissance est donc rendue possible grâce à l'augmentation de la biomasse bactérienne et aux divisions cellulaires qui se produisent lorsque les bactéries se trouvent dans un milieu favorable et dans des conditions physicochimiques (température, pression, nature des gaz ambiants, pH...) optimales.

Taux de croissance μ : c'est le nombre de divisions par unité de temps t (heures).

Temps de génération t_g : c'est l'intervalle de temps entre deux divisions.

Soit une population de concentration initiale X_0 à $t=t_0$, elle augmentera après chaque génération de la façon suivante :

$$\text{Après une génération} : X_1 = 2 * X_0 \quad \text{à } t_1$$

$$\text{Après 2 générations} : X_2 = 2 * X_1 = 2^2 * X_0$$

$$\text{Après n générations} : X_n = 2^n * X_0 \quad \text{à } t_n$$

Cette équation peut être exprimée en fonction du taux de croissance :

$$\ln(2) * n = (\ln(X_n) - \ln(X_0)), \quad \text{avec } t_n - t_0 = n * t_g$$

Posons $\mu_{\max} = \ln(2)/t_g$

$$\mu_{\max} = (\ln(X_n) - \ln(X_0)) / (t_n - t_0)$$

D'où on trouve le modèle de croissance exponentielle :

$$\frac{dX}{dt} = \mu * X(t) \quad (1.4)$$

$X(t)$: est la concentration en cellules à l'instant t .

dX : est la variation de la concentration de la biomasse au cours de dt .

μ : C'est la vitesse (taux) de croissance.

Ce modèle ajuste bien la phase de croissance exponentielle (mais il est mal adapté pour décrire les croissances que l'on observe dans la nature et qui sont généralement limitées.

ZAMORA et ZARITSKY ont proposé, en 1985 une extension simple du modèle exponentiel, tenant compte de la phase de latence :

$$\text{Ln}(X(t)) = \text{Ln}(X_0) \quad t \leq \lambda$$

$$\text{Ln}(X_0) + \mu * (t - \lambda) \quad t > \lambda$$

Ce modèle suppose que le démarrage de la croissance exponentielle se fait brutalement sans phase de transition, et ne prend en compte ni la saturation du milieu ni la décroissance.

BARANYI et AL (1993), ont proposé un modèle combinant la fonction de freinage logistique avec une cinétique de passage des cellules de la phase de latence à la phase de croissance exponentielle. ils tiennent compte de la forme sigmoïdes des courbes de croissance, et de la phase transitoire :

$$\text{Ln}(X) = \text{Ln}(X_0) - \text{Ln} \left(1 + \frac{\exp(\mu * A + 1)}{\frac{X_{max}}{X_0}} \right) + \mu * A(t)$$

$$A(t) = t + (1/\mu) * \text{Ln} (\exp(-\mu t) + \exp(-\mu * \lambda) - \exp(-\mu t - \mu * \lambda))$$

Cependant, c'est un modèle qui reste très peu utilisé vue sa grande complexité.

b) Modèles secondaires :

Les modèles secondaires permettent de décrire l'effet des facteurs environnementaux sur les paramètres de croissance des modèles primaires θ : taux de croissance maximum, temps de latence (λ), X_0 , X_{max} .

$$\theta = g(T, pH, \dots)$$

Des modèles ont, également, été développés afin de définir les conditions environnementales

permettant, ou non, la croissance des micro-organismes et s'attachant essentiellement à modéliser la frontière entre les conditions permettant la croissance et celles qui ne la permettent pas.

b.1) La modélisation du taux de croissance :**Modèle tenant compte de l'effet d'un seul facteur :**

RAYKOWSKY et AL (1983) ont proposé le modèle du taux de croissance, en fonction de température, le plus simple :

$$\sqrt{\mu(t)} = \begin{cases} 0 & T < T_{min} \\ b_1 * (T - T_{min}) & T > T_{min} \end{cases}$$

Modèle tenant compte de l'effet d'un seul facteur (modèle de gamma):

En utilisant le "gamma concept", le taux de croissance devient le produit de toutes les fonctions exprimant, chacune, l'effet d'un facteur d'environnement affectant la croissance et s'exprimant par l'équation mathématique de l'effet barrière (hurdle effect).

Soit le modèle suivant, qui tient compte de l'effet de la température, ph :

$$\mu = \gamma * \mu_{opt}$$

$$\gamma = \gamma(ph) * \gamma(T)$$

$$\gamma(ph) = (ph - ph_{min}) * \frac{(ph_{max} - ph)}{(ph_{opt} - ph_{min}) * (ph_{max} - ph_{opt})}$$

$$\gamma(T) = \left(\frac{T - T_{min}}{T_{opt} - T_{min}} \right)^2$$

$$0 < \gamma < 1$$

Les valeurs cardinales (max, opt, min) sont supposées constantes pour le microorganisme étudié.

Exemple : L'influence du taux de sel sur le taux de croissance peut être prise en compte par le modèle de gamma, en utilisant l'équation de Monod

$$\mu = \mu_{opt} * \gamma(S) = \mu_{opt} * (S / (K_s + S))$$

S : concentration en sel.

K_s : constante qui correspond à la concentration de μ_{opt} divisée par 2.

Selon le même principe, on peut compléter le modèle de gamma par d'autres facteurs : les substances inhibitrices (acides organiques) en supposant toujours que leurs effets sont multiplicatifs.

Inconvénients :

- Difficulté à estimer les croissances aux conditions limites.
- Ignorance des éventuelles interactions entre les facteurs.

Lorsque les conditions de température sont défavorables, le microorganisme devient plus exigeant quant aux conditions d'autres facteurs (exemples ceux du pH : pH_{min} change).

Une extension du modèle de Gamma a été proposée par AUGUSTIN (1999), pour donner un modèle qui tient compte des interactions.

Pour l'exemple précédent, la détermination de pH_{min} ou t_{min} obéit la loi suivante :

$$X_{min} = X_{opt} - (X_{opt} - X_{min}^o) * (1 - ((y_{opt} - y) / (y_{opt} - y_{min}^o))^3)^{1/3}$$

X et Y sont les deux variables entre lesquelles il y a interaction.

Y_{min}^o , X_{min}^o obtenues lorsque l'autre variable est fixée et optimale.

b.2) Modélisation du temps de latence :

En ce qui concerne le temps de latence, il est généralement admis que quelque soient les conditions de croissance, celui-ci est inversement proportionnel au taux de croissance maximum (Zwietering et al. 1994 ; Rosso, 1995, 1998), obtenu dans les mêmes Conditions d'environnement. Delignette-Muller (1998) a montré que la relation suivante reste satisfaisante dans la majorité des cas et répond au critère de parcimonie :

$$\ln(\mu_{max} \times \lambda) = \ln(K)$$

K : est une constante dépendant de l'état physiologique de l'inoculum ;

λ : temps de latence.

Le temps de latence peut disparaître en agissant sur les conditions décrivant l'histoire de l'inoculum. Johannsen *et al.* (1994).

George et Lund (1992), ont observé que les cellules, en phase exponentielle de croissance, reprenaient immédiatement leur croissance, lorsqu'elles étaient transférées dans un milieu de culture frais.

I.5 bilan de matière :

Modèle permettant de décrire le comportement dynamique des différents composants de la réaction biologique, en exprimant, par des équations différentielles, la variation de la quantité d'un composé comme égale à la somme de ce qui est produit ou apporté, diminué de ce qui est consommé ou soutiré.

Les modèles de bilan matières sont constitués de deux types de termes représentant la conversion et le transport de matière. La conversion englobe la vitesse à laquelle se déroulent les diverses réactions biochimiques du procédé et les rendements de conversion des divers substrats en biomasse et produits. Pour sa part, la dynamique de transport regroupe le transit de la matière au sein du procédé.

Equations du bilan de matière :

Quel que soit le mode de fonctionnement (batch, fed batch, continu), le comportement dynamique des différents composants de la réaction biologique (biomasse, substrat, produits) découle directement de l'expression des bilans de matières. D'après la loi fondamentale sur la conservation de la matière, de Lavoisier, L'équation générale d'évolution de chacun de ces éléments sur un intervalle de temps déterminé est régie par:

$$\left[\begin{array}{c} \text{Variation de la quantité} \\ \text{dans le réacteur} \end{array} \right] = \left[\begin{array}{c} \text{Quantité} \\ \text{apportée} \end{array} \right] - \left[\begin{array}{c} \text{Quantité} \\ \text{soutirée} \end{array} \right] + \left[\begin{array}{c} \text{Quantité} \\ \text{produite} \end{array} \right] - \left[\begin{array}{c} \text{Quantité} \\ \text{consommée} \end{array} \right] \quad (1.5)$$

Dans le cas d'un réacteur infiniment mélangé, à alimentation et soutirage permanents et à volume variable, les équations de la dynamique de la croissance d'une population de micro-organismes sur un simple substrat sont représentées par les équations différentielles ordinaires suivantes SETNES (1998):

On applique la formule (1.5) pour chaque composant de la réaction.

1. L'équation différentielle de la croissance de la biomasse :

$$\frac{d(X*V)}{dt} = 0 - X * Q_{out} + V * v_x - 0 \quad (1.6)$$

2. L'équation différentielle de consommation du substrat (lactosérum) :

$$\frac{d(S*V)}{dt} = S_{in} * Q_{in} - S * Q_{out} + 0 - V * v_S \quad (1.7)$$

3. L'équation différentielle de production (l'acide lactique):

$$\frac{d(P*V)}{dt} = 0 - P * Q_{out} + V * v_p - 0 \quad (1.8)$$

4. L'équation de variation du volume :

$$\frac{dV}{dt} = Q_{in} - Q_{out} \quad (1.9)$$

S_{in} : concentration de substrat dans la solution d'alimentation (g/L),

Q_{in} : débit d'alimentation en substrat (L/h),

Q_{out} : débit de soutirage (L/h),

V : volume réactionnel (L),

X : concentration en micro-organismes (g/L),

S : concentration du substrat dans le réacteur (g/L),

P : concentration du métabolite dans le réacteur (g/L),

v_X : vitesse de croissance de la biomasse (g/L h),

v_S : vitesse de consommation du substrat (g/L h).

v_p : vitesse de production d'acide lactique (g/L h).

Chapitre II
Modélisation et étude de
stabilité en boucle
ouverte du bioréacteur

II.1 Introduction :

Dans le chapitre précédent nous avons étudié l'aspect biologique de la fermentation, les éléments pouvant influencer une telle réaction, ainsi que quelques modèles mathématiques établis pour représenter, le mieux, le comportement des variables qui interviennent dans le métabolisme réactionnel (chaque modèle a été établi sous certaines hypothèses pour répondre à des exigences bien précises).

Dans le présent chapitre nous allons nous intéresser à un type de fermentation bien défini : La fermentation d'acide lactique par *Streptococcus* en mode continu. Pour un but d'étude du comportement du système global (input/output) et de visualisation de l'évolution des variables les plus importantes intervenant dans le métabolisme de fermentation.

II.2 Modélisation du bioréacteur de fermentation continue d'acide lactique :

II.2.1 Modélisation de la cinétique de croissance des *Streptococcus thermophilus* :

II.2.1.1 Modèle primaire de la cinétique de croissance :

Etant donné que nous travaillons en mode continu, la phase de croissance exponentielle est celle qui nous intéresse, le modèle le plus représentatif et le plus simple est le modèle de croissance exponentielle.

Dans le modèle de croissance exponentielle, la vitesse de la croissance est proportionnelle à la quantité de biomasse présente dans le réacteur :

$$vX = \frac{dX}{dt} = \mu * X \quad (2.1)$$

μ : Taux spécifique de croissance (h^{-1}).

X : concentration en biomasse à l'instant t (g/l)

dX : variation de la concentration de la biomasse à l'instant t .

II.2.1.2 Modèle secondaire de la cinétique de croissance :

Dans cette modélisation nous ne tenons compte que de l'effet de la concentration du substrat sur la croissance bactérienne, sous l'hypothèse que la réaction s'effectue dans des conditions environnementales optimales et invariantes.

Le modèle utilisé est celui de Haldane Andrews (1968)

$$\mu(S) = \mu_0 * \frac{S}{K_S + S + \frac{S^2}{K_I}} \quad (2.2)$$

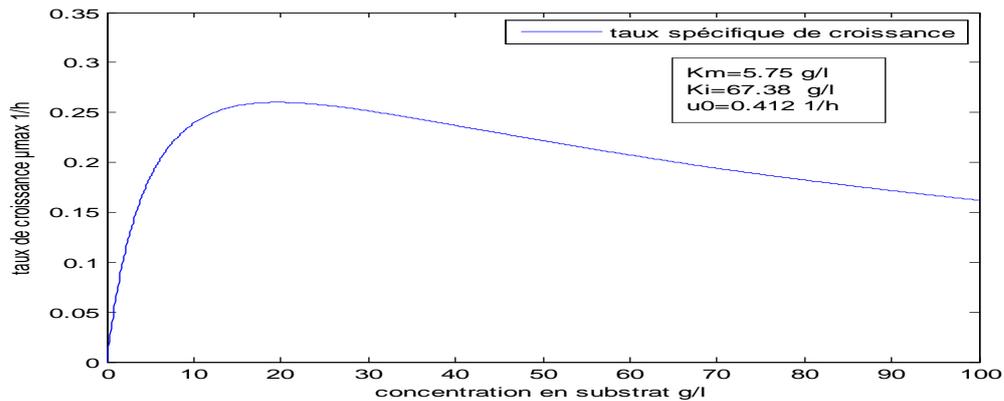


Figure (2.1) la représentation du modèle de Haldane du taux de croissance.

S : concentration du substrat dans le bioréacteur.

K_i : constante d'inhibition de la croissance liée à la forte concentration de S .

K_s : constante de demi-saturation (g/L) du modèle de Monod.

μ_0 : Taux spécifique de croissance optimale de Monod.

En l'absence de substrat ou au cas où l'un des nutriments commence à faire défaut, $S=0$ ce qui donne $\mu=0$, alors nous assistons à l'absence de la croissance.

Lorsque S devient très grand $\lim_{S \rightarrow \infty} \mu(S) = 0$ ce qui conduit à l'inhibition de croissance ($\mu=0$).

II.2.1.3 Modélisation de la cinétique de production :

Luedeking et Piret (1959a), ont montré que la production d'acide lactique était partiellement liée à la vitesse de croissance et à la concentration de la biomasse existante, déjà, dans le milieu réactionnel. La production accompagne la phase de multiplication et la phase de maintien. Ils proposent la formule suivante de la vitesse de production :

$$v_p = \frac{dP(t)}{dt} = \alpha * \frac{dX}{dt} + \beta * X = (\alpha\mu + \beta)X \quad (2.3)$$

$$\mu = \frac{dX}{Xdt}$$

α, β : constantes;

$P(t)$: concentration d'acide lactique.

$dP(t)$: variation de la concentration d'acide lactique à l'instant t .

II.2.1.4 Modélisation de la vitesse de consommation de substrat :

Selon Piret, (1975), la vitesse de consommation du substrat est la somme de deux termes, l'un est lié à la consommation nécessaire à la croissance (c) et l'autre au maintien (m).

$$v_s = \frac{dS(t)}{dt} = \frac{dS(t)}{dt}_c + \frac{dS(t)}{dt}_m \quad (2.4)$$

$dS(t)$: variation de la concentration du substrat à l'instant t .

Le terme lié à la croissance est proportionnel à la vitesse de croissance :

$$\frac{dS}{dt}_c = -\frac{1}{Y_{xs}} * \frac{dX}{dt}$$

En substituant (2.1) dans cette dernière équation on trouve :

$$\frac{dS}{dt}_c = -\frac{\mu}{Y_{xs}} * X$$

Le signe (-) est dû au fait qu'une croissance de la biomasse correspond à une consommation du substrat.

$Y_{xs} = -\frac{dX}{dS}$ Le rendement de conversion Substrat en Biomasse.

$$\frac{dS}{dt}_m = -ms * X$$

On trouve finalement :

$$v_s = \left(-ms - \frac{\mu}{Y_{xs}}\right) X \quad (2.5)$$

Le terme représentant le maintien est souvent négligeable, et la formule (2.5) devient :

$$v_s = \frac{\mu}{Y_{xs}} X \quad (2.6)$$

II.2.2 L'établissement du modèle d'état :

L'établissement de la représentation d'état du bioréacteur, consiste en la fusion des équations des cinétiques réactionnelles et les équations des Bilans de matières.

La substitution des expressions de v_s , v_p et v_x , données par les formules (2.1), (2.3) et (2.6), dans, le modèle du bioréacteur, représenté par les formules (1.6), (1.7), (1.8) et (1.9). Donne le modèle d'état suivant :

$$\frac{d(X*V)}{dt} = -X * Q_{out} + X * \mu * V \quad (2.7)$$

$$\frac{d(S*V)}{dt} = S_{in} * Q_{in} - S * Q_{out} - X * V * \mu / Y_{xs} \quad (2.8)$$

$$\frac{d(P*V)}{dt} = -P * Q_{out} + (\alpha * \mu + \beta) * X * V \quad (2.9)$$

$$\frac{dV}{dt} = Q_{in} - Q_{out} \quad (2.10)$$

Comme notre fermentation est en mode continue à volume constant, les équations (2.7) à (2.1) deviennent :

$$\frac{dX}{dt} = -X * D + X * \mu \quad (2.11)$$

$$\frac{dS}{dt} = S_{in} * D - S * D - X * \mu / Y_{xs} \quad (2.12)$$

$$\frac{dP}{dt} = -P * D + (\alpha * \mu + \beta) * X \quad (2.13)$$

Avec :

$$\mu(S) = \mu_0 * \frac{S}{K_s + S + \frac{S^2}{K_i}} \quad (2.14)$$

$D = Q/V$: taux de dilution, h^{-1} . La commande du système.

q : la sortie du système (la productivité), g/lh.

II.2.3 les paramètres de fermentation :

Le métabolisme de fermentation étudié est celui de la fermentation continue d'acide lactique par *Streptococcus thermophilus*, supposée réalisée sous des conditions environnementales optimales et fixes. le tableau suivant donne les valeurs des paramètres de la fermentation considérée :

Paramètres/conditions optimale	Valeur
PH	6.42
T	42°C
μ_0	0.421
K_s	5.75
K_i	67.38
α	5.1523
β	0.0821
Y_{xs}	0.39

Tableau (2.1) : les paramètres de la cinétique de fermentation d'acide lactique par *Streptococcus thermophilus*.(Boudjema, 2008).

$S_{in} = 20g/l$.

II.3 Etude de la stabilité du fermenteur continu d'acide lactique :

Pour l'étude de la stabilité du système (représentation d'état (2.11) à (2.13)), on utilise la méthode indirecte de Lyapunov, Qui nous permet de conclure sur la stabilité locale des systèmes non linéaires (autour des points d'équilibre).

La conclusion sur la stabilité locale du système non linéaire, passe par la détermination de celle du système linéaire, résultant du développement en série de Taylor, du système nonlinéaire, où

seul le premier ordre du développement est retenu. Les valeurs propres de la matrice de transition du système linéaire permettent de trancher, dans certain, cas sur le type de stabilité du système global.

Le théorème suivant donne les conclusions sur la stabilité du système non linéaire, à l'exclusion du cas où au moins une des valeurs propre est nulle et les autres sont strictement négatives, dans ce cas la stabilité est déterminée par les ordres supérieurs du développement.

Théorème : soit $\dot{x} = f(x)$, et on pose $\left. \frac{\partial f}{\partial x} \right|_{x_0}$ avec

$$\Lambda = \lambda(A)$$

$\forall \lambda \in \mathbb{A}, \operatorname{Re}(\lambda) < 0 \Rightarrow x = x_0$ est asymptotiquement stable

$\exists \lambda \in \mathbb{A}, \operatorname{Re}(\lambda) > 0 \Rightarrow x = x_0$ est instable

$\forall \lambda \in \mathbb{A}, \operatorname{Re}(\lambda) \leq 0 \Rightarrow x = x_0$ on ne peut pas conclure

II.3.1 Détermination des points d'équilibre :

Les points d'équilibre (\bar{x}, \bar{u}) d'un système dynamique sont les états ayant une variation temporelle nulle, déterminés par la formule suivante :

$$f(\bar{x}, \bar{u}) = 0 \quad (2.15)$$

x : vecteur d'état du système.

u : vecteur de commande.

Nous appliquons la formule (2.17) au système non linéaire de fermentation,

Pour $x=(x, s, p)$ $u=D$.

Ce qui donne:

$$\begin{cases} 0 = X(\mu - D) & (2.16. a) \\ 0 = (S_{in} - S)D - \mu * \frac{X}{Y_{xs}} & (2.16. b) \\ 0 = (\alpha\mu + \beta)X - PD & (2.16. c) \end{cases}$$

De la formule (2.16.a), deux solutions sont possibles :

$$\begin{cases} X = 0 & (2.17. a) \\ \mu = D & (2.17. b) \end{cases}$$

La solution (2.17.a) donne une concentration en biomasse nulle dans le bioréacteur. Absence de microorganisme → pas de métabolisme de fermentation. Cette solution est à rejeter, et le point d'équilibre correspondant est indésirable. Il correspond à un phénomène dit lessivage.

La solution $\mu = D = \bar{D}$ est retenue.

En substituant (2.17.b) dans l'expression de Haldane, Deux solutions d'équilibres sont possibles, figure (2.1) :

$$\bar{S}_1 < \sqrt{k_s k_i} \quad \text{et} \quad \bar{S}_2 > \sqrt{k_s k_i}$$

$\sqrt{k_s k_i}$ Concentration du substrat qui donne $\mu = \mu_{\max}$.

A l'équilibre, les états peuvent prendre les valeurs suivantes :

$$\begin{cases} \bar{S}_1 < \sqrt{k_s k_i} & , \quad \bar{X}_1 = (S_{in} - \bar{S}_1) * Y_{xs} & , \quad \bar{P}_1 = \bar{X}_1(\alpha \bar{D} + \beta) / \bar{D} \\ \bar{S}_2 > \sqrt{k_s k_i} & , \quad \bar{X}_2 = (S_{in} - \bar{S}_2) * Y_{xs} & , \quad \bar{P}_2 = \bar{X}_2(\alpha \bar{D} + \beta) / \bar{D} \end{cases}$$

Et la sortie (la productivité) prend les valeurs suivantes :

$$\begin{cases} Q_1 = \bar{P}_1 * \bar{D} \\ Q_2 = \bar{P}_2 * \bar{D} \end{cases}$$

La stabilité de ces points d'équilibre sera étudiée dans les parties (II.4.6 et II.3.8).

II. 3.2 Calcul de la sortie en fonction de l'entrée du système à l'équilibre :

De la reprise des équations (2.16.a) à (2.16.c), nous trouvons :

$$D^3 a_3 + D^2 a_2 + D a_1 + a_0 = 0 \quad (2.18)$$

$$a_3 = \alpha^2 \left(k_s + S_{in} \left(1 + \frac{S_{in}}{k_i} \right) \right)$$

$$a_2 = \alpha \left(2\beta \left(k_s + S_{in} \left(1 + \frac{S_{in}}{k_i} \right) \right) - \alpha \mu_0 S_{in} - \frac{Q}{Y_{xs}} \left(1 + 2 \frac{S_{in}}{k_i} \right) \right)$$

$$a_1 = \frac{Q^2}{Y_{xs}^2 * k_i} + \beta^2 \left(k_s + S_{in} \left(1 + \frac{S_{in}}{k_i} \right) \right) - 2\alpha\beta\mu_0 S_{in} + \frac{Q}{Y_{xs}} \left(\alpha\mu_0 - \beta \left(1 + 2 \frac{S_{in}}{k_i} \right) \right)$$

$$a_0 = -\mu_0 \left(\frac{Q}{Y_{xs}} + \beta^2 S_{in} \right)$$

En observant cette équation de troisième ordre, nous concluons directement que pour la même valeur de sortie Q on peut faire correspondre plusieurs valeurs d'entrées D comme solutions de cette équation.

Ce type de système est dit *système à entrées multiples (system with input multiplicities)*, Pour lequel plusieurs valeurs en entrée peuvent donner la même valeur en sortie. Figure (2.2).

Les systèmes avec input multiplicities sont généralement ceux qui présentent des effets compétitifs (exemple : réaction chimiques ...). Dans notre cas, cela est due au taux spécifique de croissance en fonction de la concentration du substrat figure (2.1), une augmentation de la concentration de substrat augmente le taux spécifique de croissance et favorise le métabolisme réactionnel. C'est cette augmentation de la concentration, lorsqu'elle dépasse la valeur $\sqrt{k_s k_i}$ correspondante à μ_{\max} , qui va réduire la cinétique de croissance et inhiber le métabolisme fermentaire.

II. 3.3 Représentation de la sortie en fonction de l'entrée à l'équilibre :

La figure suivante représente les résultats expérimentaux de la sortie (productivité) en fonction de l'entrée (taux de dilution) à l'équilibre:

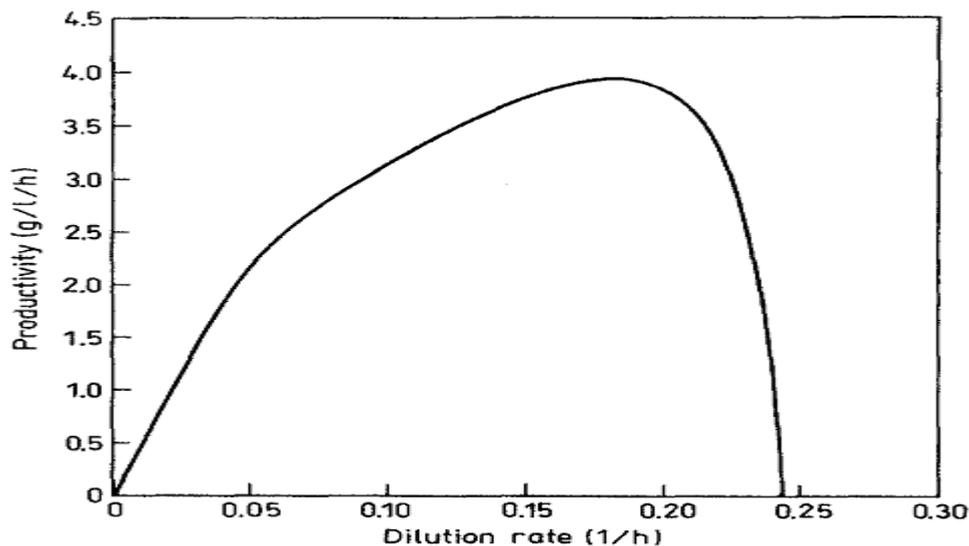


Figure (2.2) : la caractéristique expérimentale entrée-sortie de fermentation continue d'acide lactique. (Reddy ,1995)

D'après cette figure, on constate qu'une valeur de sortie désirée, à l'équilibre, peut être obtenue avec deux valeurs différentes en entrée, à l'exception d'un seul point, celui correspondant à une productivité maximale qui ne peut être obtenue qu'avec une seule entrée $D = \mu_{\text{optimal}}$.

En tenant compte de ces résultats expérimentaux, et ceux théorique de (II.3.2), nous concluons que deux solutions de l'équation (2.18) sont retenues, et la troisième doit être obligatoirement négative.

II.3.4 Etude du comportement réel (expérimental) du système à l'équilibre :

D'après la figure (2.2) nous observons que, plus le taux de dilution augmente, la productivité augmente jusqu'à $D = \mu_{\text{optimale}} < \mu_{\text{max}}$ correspondant à une productivité maximale.

Au-delà de cette valeur du taux de dilution, la productivité chute très rapidement et le système devient très sensible aux faibles variations de D , entraînant le système vers le point de lessivage indésirable. Cette interprétation sera appuyée par les résultats de l'étude de la stabilité dans les parties (II.3.8 et II.4.7) qui démontrent qu'au niveau de cette partie de la courbe tous les points d'équilibre sont moins stables. Le système va se trouver finalement sur le point d'équilibre indésirable pour de faibles perturbations.

Lorsque le taux de dilution atteint ou dépasse μ_{max} le système sera entraîné vers le lessivage, la figure (2.2) montre que aucune sortie ne corresponde à un taux de dilution $> \mu_{\text{max}}$.

Cette courbe Figure (2.2) exprime clairement la forte nonlinéarité du système. En plus de la non linéarité du gain statique nous remarquons que celui-ci change de signe. Il est positif dans la partie correspondante à des points d'équilibres les plus stables, et devient négatif pour les points d'équilibres les moins stables.

II.3.5 Linéarisation du système non linéaire :

La linéarisation d'un système d'équation de la forme $\dot{x} = f(x, u, t)$ autour d'un point d'équilibre (\bar{x}, \bar{u}) se fait par le développement de Taylor suivant :

$$\dot{x}_i = f_i(\bar{x}, \bar{u}, t) + \sum_{j=1}^n \frac{\partial f_i}{\partial x_j(\bar{x}, \bar{u})} * \Delta x_j + \sum_{k=1}^m \frac{\partial f_i}{\partial u_k(\bar{x}, \bar{u})} * \Delta u_k + o(\bar{x}, \bar{u}, t, \Delta x, \Delta u)$$

$$\lim_{(x,u) \rightarrow \bar{x}, \bar{u}} \|o(\bar{x}, \bar{u}, t, \Delta x, \Delta u)\| / \|\Delta x, \Delta u\| = 0$$

$$\Delta x_j = x_j - \bar{x}_j$$

$$\Delta u_j = u_j - \bar{u}_j$$

$$\dot{x}_i = \Delta \dot{x}_i + \dot{\bar{x}}_i$$

$$\dot{\bar{x}}_i = f_i(\bar{x}, \bar{u}, t)$$

$$\dot{\Delta x}_i = \sum_{j=1}^n \frac{\partial f_i}{\partial x(\bar{x}, \bar{u})} * \Delta x_j + \sum_{k=1}^m \frac{\partial f_i}{\partial u(\bar{x}, \bar{u})} * \Delta u_k \quad (2.19)$$

On obtient le système linéaire suivant :

$$\dot{\Delta \mathbf{x}} = \mathbf{A} * \Delta \mathbf{x} + \mathbf{B} * \Delta \mathbf{u}$$

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x(\bar{x}, \bar{u})} \\ \vdots \\ \frac{\partial f_n}{\partial x(\bar{x}, \bar{u})} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u(\bar{x}, \bar{u})} \\ \vdots \\ \frac{\partial f_n}{\partial u(\bar{x}, \bar{u})} \end{bmatrix}$$

\mathbf{A} : matrice jacobienne de $f(x, u, t)$.

$\mathbf{x} = (x_1 \dots x_n)^t$: vecteur d'état du système non linéaire.

$\mathbf{u} = (u_1 \dots u_m)^t$: vecteur de commande du système non linéaire.

$\Delta \mathbf{x} = (\Delta x_1 \dots \Delta x_n)^t$: vecteur d'état du système linéarisé.

$\Delta \mathbf{u} = (\Delta u_1 \dots \Delta u_m)^t$: vecteur de commande du système linéarisé.

II.3.6 Linéarisation du modèle du fermenteur continu d'acide lactique :

Pour le point d'équilibre suivant $(\bar{x}, \bar{s}, \bar{p}, \bar{D}, \bar{q})$, on a appliqué la formule (2.19) à la représentation d'état du système non linéaire (2.11) à (2.14). Le système linéarisé résultant est le suivant :

$$\dot{\Delta \mathbf{x}} = \mathbf{A} * \Delta \mathbf{x} + \mathbf{B} * \Delta \mathbf{u} \quad (2.20)$$

$$\mathbf{A} = \begin{pmatrix} 0 & -c & 0 \\ -\bar{D}/y_{xs} & -\bar{D} - c/y_{xs} & 0 \\ \alpha \bar{D} + \beta & \alpha c & -\bar{D} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} -\bar{x} \\ s_{in} - \bar{s} \\ -\bar{p} \end{pmatrix}, \quad c = \bar{x} * \mu_0 * \frac{k_s \frac{\bar{s}^2}{k_i}}{\left(k_s + \bar{s} + \frac{\bar{s}^2}{k_i}\right)^2}$$

$$\Delta \mathbf{x} = (\Delta x \ \Delta s \ \Delta p)^t$$

$$\Delta \mathbf{u} = \Delta D$$

\bar{q} : la productivité à l'équilibre.

II.3.7 Etude de la stabilité du système linéarisé :

La détermination de la stabilité du système linéarisé (2.20) est dépendante du signe de la partie réelle des valeurs propres de \mathbf{A} .

Si toutes les valeurs propres de A sont à partie réelle négative, le système linéaire (2.20) est exponentiellement stable.

Si au moins une valeur propre est à partie réelle positive, alors le système linéaire (2.20) est instable.

Pour cela nous avons calculé le polynôme caractéristique de A, selon la formule suivante :

$$P(\lambda) = \lambda I - A :$$

$$\text{Nous avons obtenu : } P(\lambda) = (\lambda + \bar{D}) * \left(\lambda \left(\frac{c}{y_{xs}} + \lambda + \bar{D} \right) + \frac{c\bar{D}}{y_{xs}} \right) \quad (2.21)$$

Les valeurs propres correspondantes sont :

$$\lambda_1 = -c/y_{xs}$$

$$\lambda_2 = -\bar{D} < 0$$

La valeur propre λ_2 est toujours réelle négative car nous ne pouvons pas avoir de taux de dilution négatif. C'est le signe de λ_1 qui va déterminer la stabilité du système (2.20). Le signe de λ_1 est celui de C, car y_{xs} est un rendement > 0 .

$$c = \bar{x} * \mu_0 * \frac{k_s \frac{s^2}{k_i}}{\left(k_s + s + \frac{s^2}{k_i} \right)^2}$$

$$\begin{cases} c > 0 \text{ si } s < \sqrt{k_s k_i} & \rightarrow \lambda_1 < 0 & (2.22.a) \\ c < 0 \text{ si } s > \sqrt{k_s k_i} & \rightarrow \lambda_1 > 0 & (2.22.b) \\ c = 0 \text{ si } s = \sqrt{k_s k_i} & \rightarrow \lambda_1 = 0 & (2.22.c) \end{cases}$$

Si la formule (2.22.a) est vérifiée, nous aurons un système linéaire exponentiellement stable.

Si la formule (2.22.b) est vérifiée, le système linéaire est instable.

Si la formule (2.22.c) est vérifiée, le système linéaire est asymptotiquement stable.

II.3.8 Conclusion sur la stabilité du système non linéaire :

D'après les conditions de Lyapunov, nous constatons que plusieurs cas se présentent :

- Le système non linéaire est asymptotiquement stable sur les points d'équilibre vérifiant $s < \sqrt{k_s k_i}$ et correspondants à $\mu < \mu_{max}$ (1.)
- Le système non linéaire est instable sur les points d'équilibres vérifiant $s > \sqrt{k_s k_i}$ correspondants à $\mu > \mu_{max}$.

- La méthode de Lyapunov ne permet pas de décider sur la stabilité du système non linéaire lorsque $s = \sqrt{k_s k_i}$. Pour décider sur la stabilité en ce point nous allons tracer la solution du système non linéaire (2.11) à (2.14) et voir son comportement.

II.3.9 Interprétation biologique des résultats de la stabilité:

Lorsque le taux de dilution est supérieur à la vitesse de croissance maximale, la vitesse de soutirage est plus importante à celle de la croissance, (multiplication) et après un certain temps (ce temps est proportionnelle à la vitesse de soutirage) il n'y aura que le substrat dans le réacteur. on dit que le réacteur a divergé vers un état indésirable appelé lessivage ($S=Sin$, $X=0$, $P=0$, $Q=0$).

Ce comportement (divergence du point d'équilibre, vérifiant $\bar{s} > \sqrt{k_s k_i}$, vers un autre point d'équilibre, attractif mais indésirable), explique bien l'instabilité conclue précédemment lorsque $\bar{s} > \sqrt{k_s k_i}$ qui correspond à un $\bar{D} > \mu_{max} = \sqrt{k_s k_i}$.

Lorsque le taux de dilution est inférieur à la vitesse de croissance maximale, les cellules auront le temps de se multiplier et de produire le métabolite à une vitesse $\mu = \bar{D}$ en régime permanent. Ceci explique la stabilité conclue précédemment, lorsque $\bar{s} < \sqrt{k_s k_i}$, correspondant à des taux de dilution $\bar{D} < \mu_{max} = \sqrt{k_s k_i}$.

II.4 Simulation en boucle ouverte du bioréacteur :

D'après la formule (2.2) :

$$\mu_{max} = \frac{\mu_0}{1+2*\sqrt{\frac{K_s}{K_i}}} \quad (2.23)$$

Nous appliquons les données du tableau (2.1) à la formule précédente pour trouver :

$$\mu_{max} = 0.26 \text{ h}^{-1} \quad (2.24)$$

Pour étudier le comportement du système en boucle ouverte, nous avons choisi quatre commandes, correspondantes à quatre points d'équilibre P, à appliquer sur le système non linéaire se trouvant dans un état initial x_0

$$\bar{D}_1 = 0.0927 \quad (\text{pour P1})$$

$$\bar{D}_2 = 0.2505 \quad (\text{pour P2})$$

$$\bar{D}_3 = 0.26 \quad (\text{pour P3, correspondant à } \mu_{max})$$

$$\bar{D}_4 = 0.3 \quad (\text{pour P4})$$

$$x_0 = \begin{cases} x_0 = 2 \text{ g/l} \\ S_0 = 5 \text{ g/l} \\ P_0 = 0.2 \text{ g/l} \end{cases}$$

II.4.1 test de stabilité de P1

Simulation 1 : application de la commande $\overline{D1}$, donnant les résultats suivants :

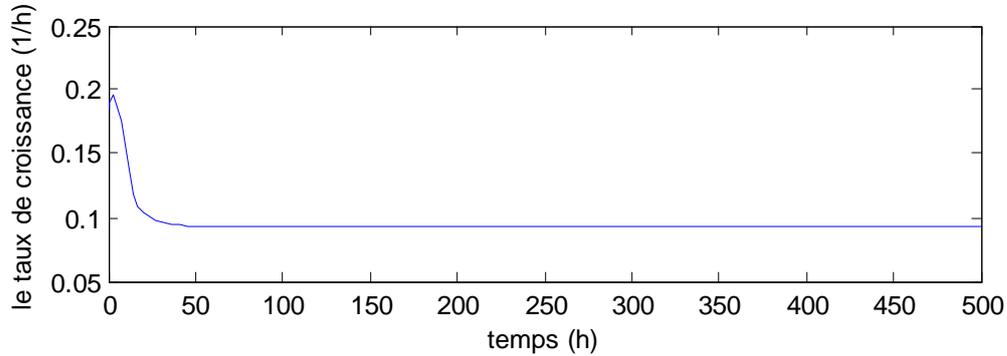


Figure (2.3) : taux de croissance correspond au point d'équilibre P1

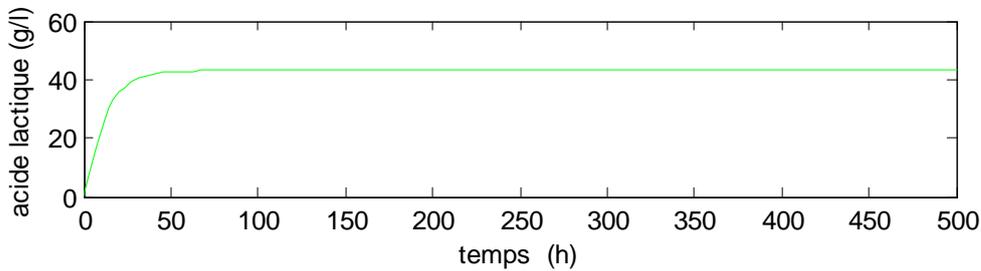


Figure (2.4) : concentration d'acide lactique, au point d'équilibre P1.

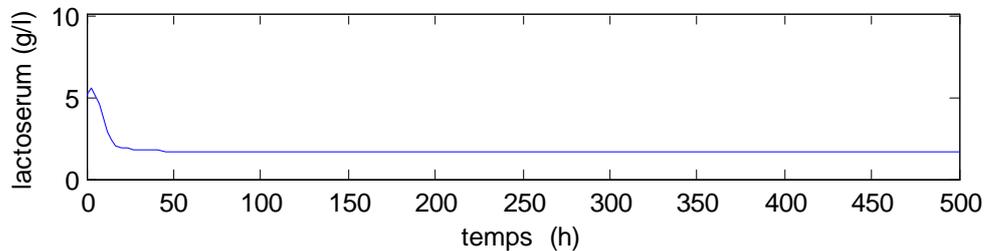


Figure (2.5) : la concentration du substrat au point d'équilibre P1

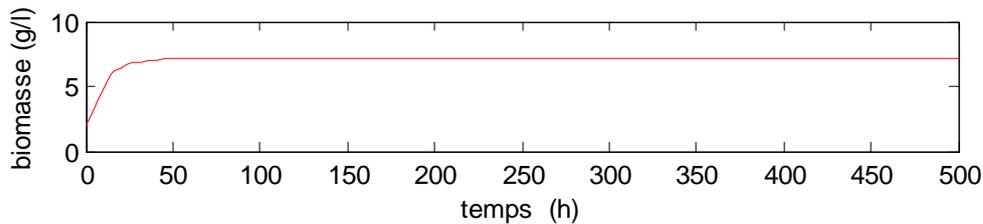


Figure (2.6) : la concentration en biomasse qui correspond au point d'équilibre P1

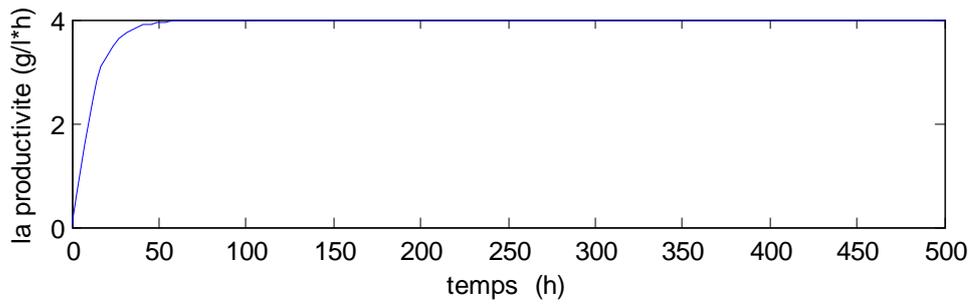


Figure (2.7) : productivité au point d'équilibre P1

II.4. 2 Test de stabilité de P2

Simulation 2 : application de la commande $\overline{D2}$, donnant les résultats suivants :

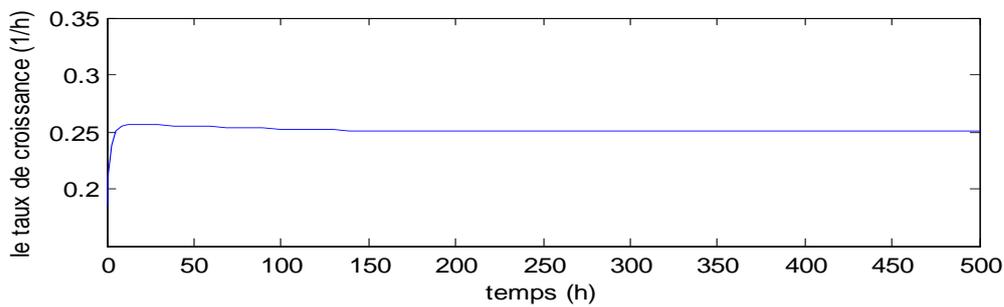


Figure (2.8) : taux de croissance pour le point P2.

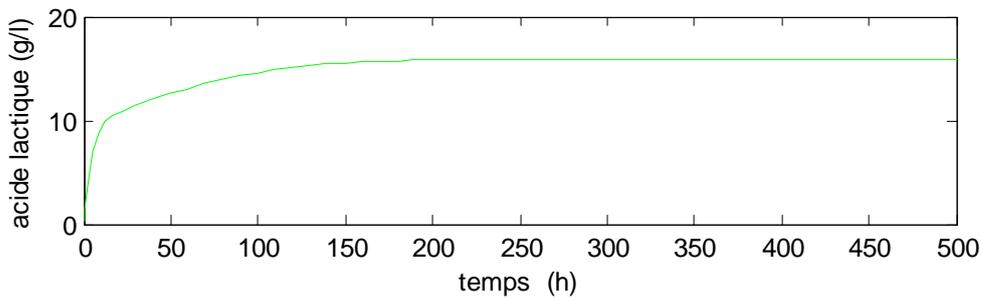


Figure (2.9) : la concentration du produit pour le point P2.

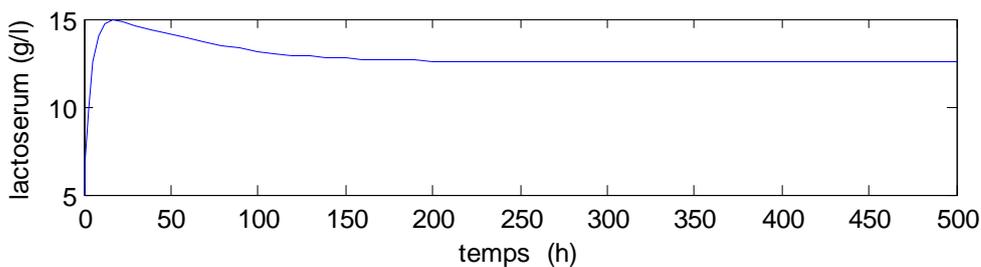


Figure (2.10) : la concentration de substrat dans le milieu réactionnel pour le point P2.

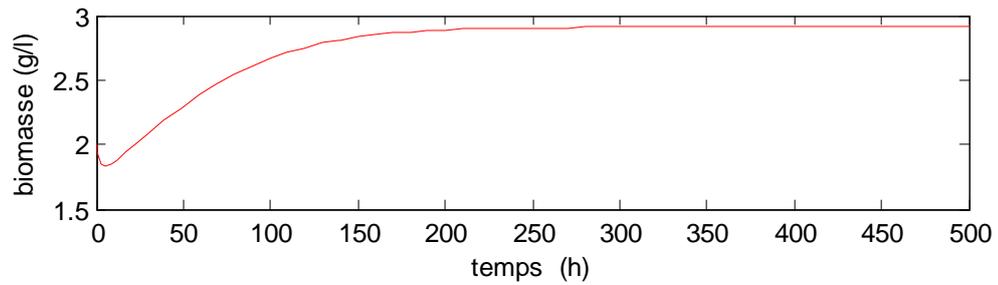


Figure (2.11) : la concentration de la biomasse dans le milieu réactionnel pour P2

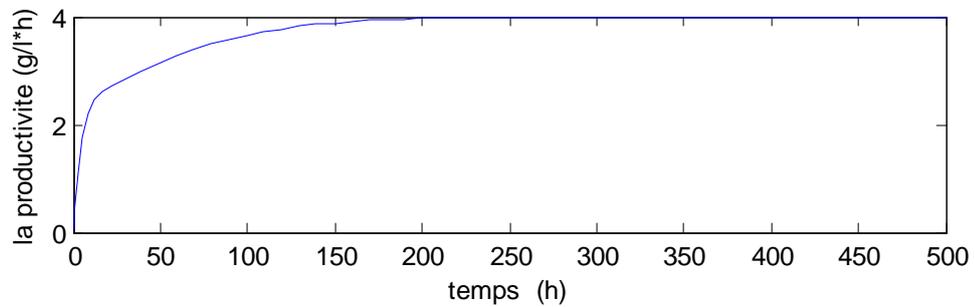


Figure (2.12) : la productivité du système en boucle ouverte pour P2.

II.4. 3 Test de stabilité de P3

Simulation 3 : application de la commande $\overline{D3}$, donnant les résultats suivants :

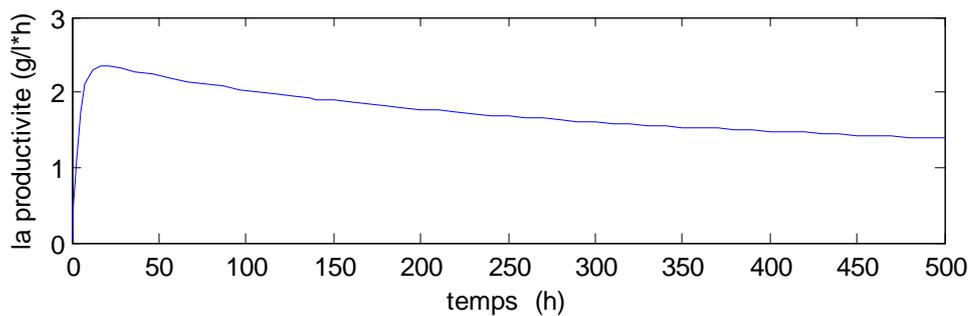


Figure (2.13) : la productivité pour le point P3, en boucle ouverte

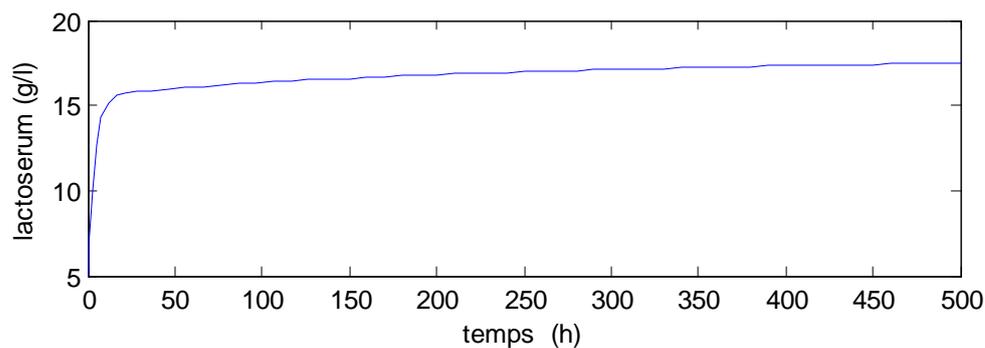


Figure (2.14) : La concentration du substrat pour le point P3.

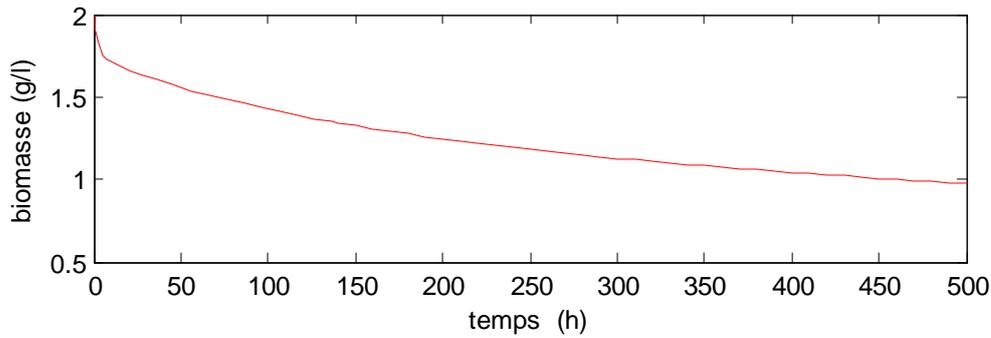


Figure (2.15) : la concentration de la biomasse pour le point P3

II.4. 4 Test de stabilité de P4

Simulation 4 : application de la commande $\overline{D4}$, donnant les résultats suivants :

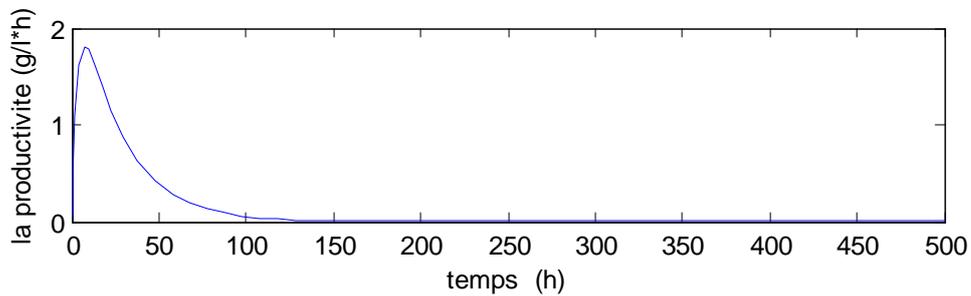


Figure (2.16) : la productivité pour le point P4

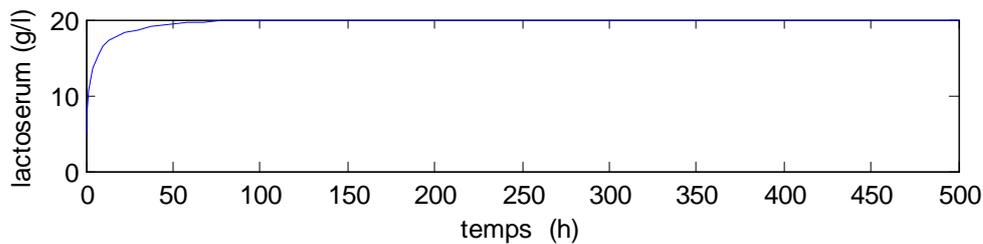


Figure (2.17) : la concentration du substrat pour le point P4

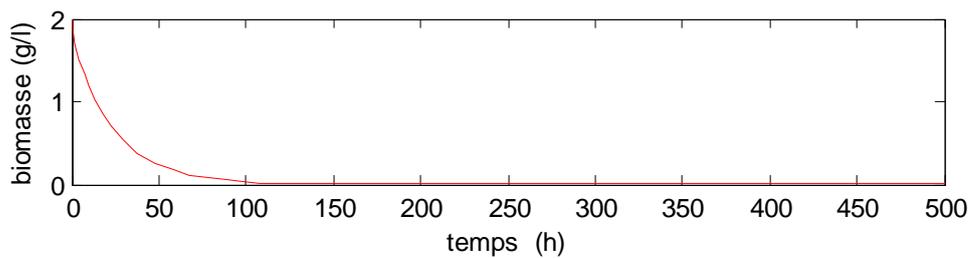


Figure (2.18) : la concentration de la biomasse pour le point P4

II.4.5 Etude du degré de stabilité des points P1 et P2

Simulation 5 :

Nous soumettons le système, à partir des deux points d'équilibres P1 et P2, à des perturbations de forme créneau sur la commande, valant 0.009 et 0.15 pendant 50h. cela nous donnera les résultats suivants :

I) pour une perturbation faible de $0.009h^{-1}$

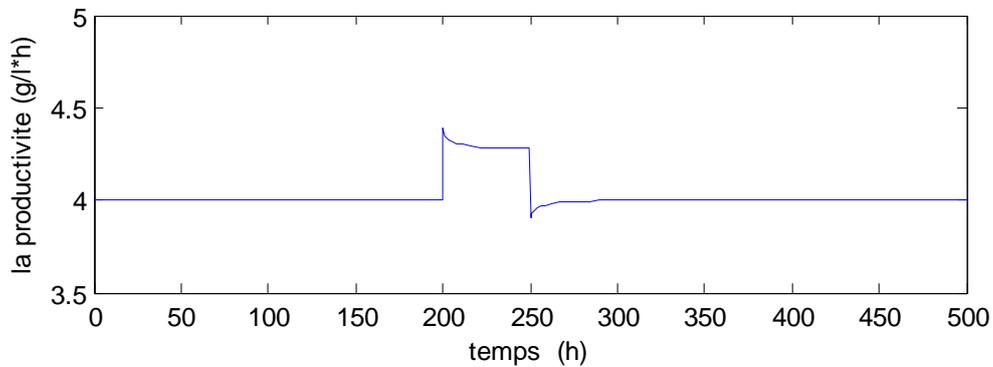


Figure (2.19) : la productivité pour une perturbation de 0.009 au point P1

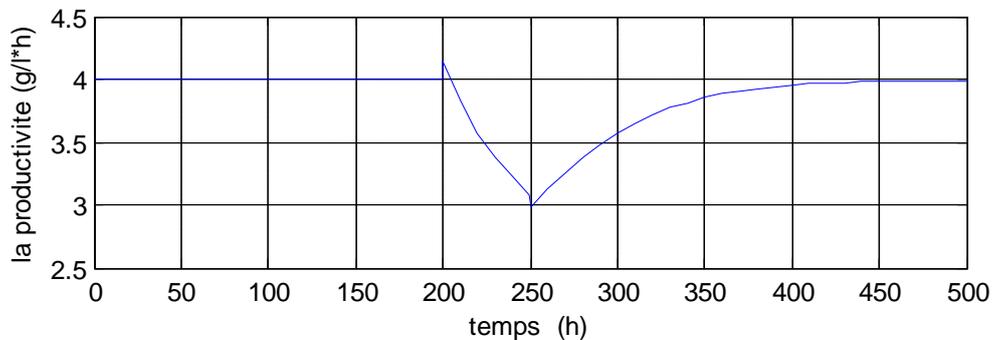


Figure (2.20) : la productivité pour une perturbation de 0.009 au point P2

II) pour une perturbation plus importante de $0.15h^{-1}$

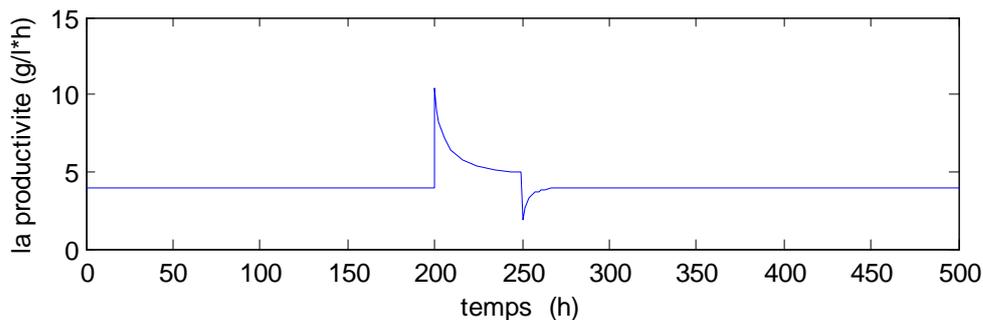


Figure (2.21) : la productivité pour une perturbation de 0.15 au point P1

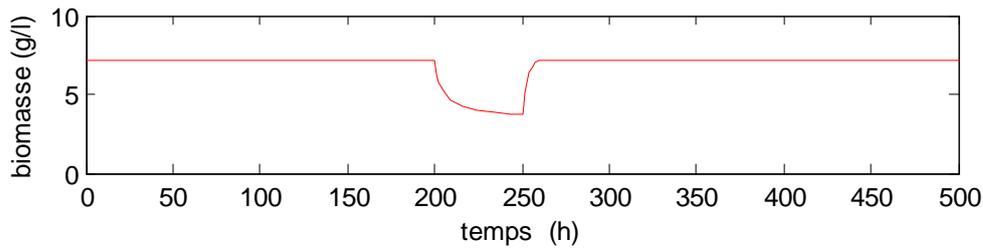


Figure (2.22) : la concentration de la biomasse pour une perturbation de 0.15 au point P1

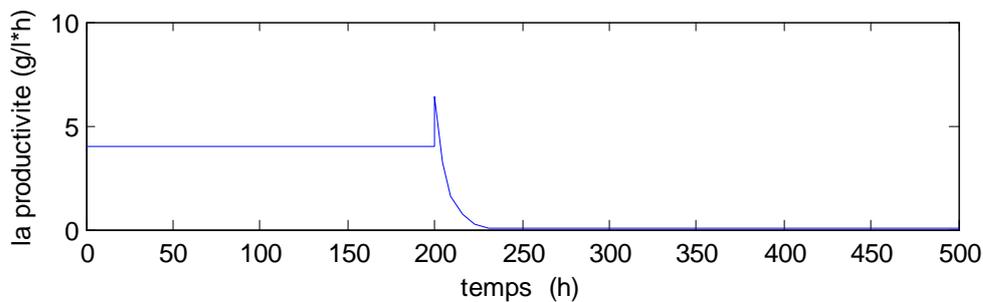


Figure (2.23) : la productivité pour une perturbation de 0.15 au point P2

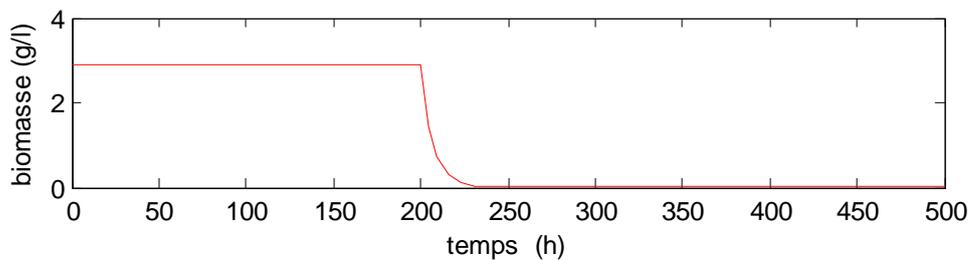


Figure (2.24) : la concentration de la biomasse pour une perturbation de 0.15 au point P2.

II.4.6 Etudes des résultats de simulations 1, 2, 3, 4:

Les figures (2.7) et (2.12) représentant la sortie du système, montrent que, pour deux commandes différentes, à partir d'un même état initial, nous obtenons deux points d'équilibre différents, ayant la même sortie Q , ceci s'explique par la caractéristique "input multiplicities" du système, étudiée précédemment dans (V.2 et V.3).

Nous remarquons que la dynamique du système diffère. elle est plus lente pour le point d'équilibre P2 (la sortie atteint sa valeur finale après 200h), que pour le point d'équilibre P1 (atteint la valeur finale de sortie après 50h).

D'après les résultats des simulations 1 et 2, correspondants respectivement à $\overline{D1}$ et $\overline{D2}$, nous concluons que les points d'équilibre P1 et P2 sont stables. Car, à partir d'un état initial x_0 ,

l'application des commandes $\overline{D1}$ et $\overline{D2}$, entraîne une convergence respective de l'état du système vers P1 et P2.

Les résultats de simulation 3 montrent que le système est instable au point P3, correspondant à la commande $D3=\mu_{\max}$. Car, à partir d'un état initial donné, la concentration de la biomasse dans le réacteur tend exponentiellement vers 0. figure (2.15) .Donc le système se trouvant sur P3 converge vers l'état de lessivage (état indésirable).

La simulation 4 montre que le point P4 qui correspond à la commande $\overline{D4}$ est instable. Les figures (2.16), de la productivité, et (2.18), de la concentration de biomasse, montrent que le système converge plus rapidement que P3 vers l'état de lessivage indésirable.

II.4.7 Etudes des résultats de simulations 5 :

Simulation 5.I :

Le retour au point d'équilibre, que ce soit pour P1 ou P2, après la disparition de la perturbation $d=0.009h^{-1}$ sur la commande, confirme les conclusions précédentes, concernant la stabilité des points d'équilibre ayant un taux de dilution $< \mu_{\max}$.

Mais les figures (2.19) et (2.20), relatives à la productivité, montrent un comportement totalement inverse du système :

- Pour P1, on observe qu'avec une augmentation du taux de dilution nous avons une augmentation de la productivité (sortie), qui s'explique par l'augmentation du taux spécifique de croissance ($\mu=D$ à l'équilibre). Car la croissance est proportionnelle à la quantité d'acide lactique produite, formule (2.3).
- Par contre pour P2, une augmentation du taux de dilution entraîne une diminution de la productivité. Ceci s'explique par l'effet inhibiteur de la concentration de la biomasse dans le milieu réactionnel. Formule de Haldane.

Simulation 5.II :

Les résultats de ces simulations montrent que, pour une perturbation importante sur la commande, le système revient à son point d'équilibre, si ce point est P1, mais diverge de P2 vers le point d'équilibre indésirable, le lessivage. Ceci s'explique par :

L'état de P2 est plus proche de l'instabilité que P1 car $\overline{D2}=0.2505$ est plus proche à $\mu_{\max}=0.26$. et une perturbation assez importante ramenant le taux de dilution à une valeur $> \mu_{\max}$, déstabilise le système, parce que dans ce cas on soutire à une vitesse plus importante que celle de la multiplication des bactéries. et à un certain moment on se retrouve avec 0 microorganismes dans le bioréacteur correspondant au lessivage. Figure (2.24).

Alors que P1 qui correspond à un taux de dilution $\overline{D1}=0.0972$, plus éloigné de μ_{\max} , accepte une plus grande marge de perturbation sur D sans que le système ne devienne instable.

II.5 Conclusions :

Pour chaque sortie $Q \neq 0$, correspondent deux valeurs de la commandes D , inférieurs à μ_{\max} , représentant deux points d'équilibres stables différents.

Les résultats et discussions précédentes, sont en conformité avec l'étude théorique de la stabilité par la méthode de Lyapunov indirecte.

- Le système est stable autour des points $P1$ et $P2$ lorsqu'ils ont comme taux de dilution $D1 = 0.0972$ ou $D2 = 0.2505$ inférieurs à $\mu_{\max}=0.26$.
- Le système est instable au point $P4$, ayant un taux de dilution $D=0.3 > \mu_{\max}=0.26$.

Pour le point $P=3$, qui correspond à un taux de dilution $D= \mu_{\max}=0.26$, La simulation a montré son instabilité. Cas indéterminé pour la méthode de Lyapunov indirect.

Le procédé de fermentation continue est fortement non linéaire, il a une dynamique de convergence variable, et son gain statique est très variable d'un point d'équilibre à un autre, et change de signe (augmentation sur $D \rightarrow$ augmentation de la productivité pour $P1$, et diminution de productivité pour $P2$).

Le point d'équilibre correspondant au plus faible taux de dilution est le plus stable.

Chapitre III
Commande du
bioréacteur de
fermentation par PI
linéaire à base de
contrôleur à modèle
interne

III.1 Historique :

La structure de base de la commande à modèle interne (internal model control IMC) a été proposée, pour la première fois, par Horwitz, puis développée et appliquée aux systèmes linéaires et stables en boucle ouverte, par Garcia et Morari (1982, 1985, b).

Le nom IMC découle de l'utilisation d'un modèle mathématique dans la boucle de régulation, comme le montre la figure (3.1). La complexité du régulateur IMC dépend directement du degré de complexité du modèle du régulateur.

Depuis des années, cette approche a fait l'objet d'un grand nombre de sujets recherches et de publications. Elle s'est imposée dans plusieurs applications, dont l'utilisation d'IMC, pour la conception d'un régulateur PID (Rivera et Morari .1986). Une amélioration du PID à base de IMC (obtenue par Rivera) a été apportée par D. Braatz (1996).

K. Yamada (1999) a étendu l'utilisation d'IMC aux systèmes instables.

D'autres chercheurs se sont intéressés à hybrider la structure IMC avec des outils d'intelligence artificielle, pour le contrôle des systèmes non linéaire et incertains. A. Aoyama (1995), Dang, Xu et Li (2009) ont proposés une approche IMC neuronale pour la commande des systèmes non linéaires.

III.2 structure de la commande à modèle interne (IMC) :

III.2.1 La structure IMC :

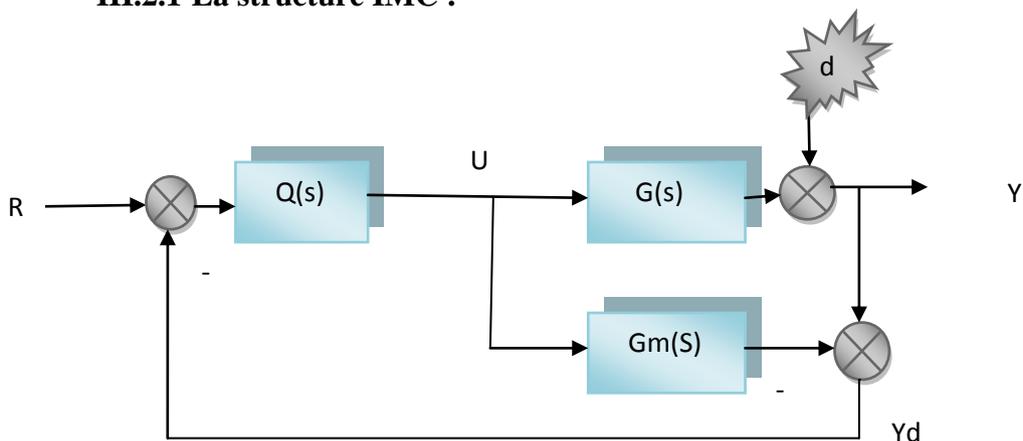


Figure (3.1) : la structure de la commande IMC

R : signal de référence.

U : commande.

Y : sortie commandée.

d : perturbation extérieure.

G : système réel.

G_m : modèle nominale du système.

III.2.2 Les avantages de cette structure :

Le but idéal de chaque processus de régulation est de réaliser la poursuite parfaite de la référence (rejection absolue de perturbation) pour toutes les fréquences.

Dans le cas de régulation de système stable, avec la parfaite connaissance de ce dernier (modèle mathématique exacte $G_m = G$), la solution la plus simple est la régulation en boucle ouverte avec un choix trivial de régulateur :

$$Q = G^{-1}.$$

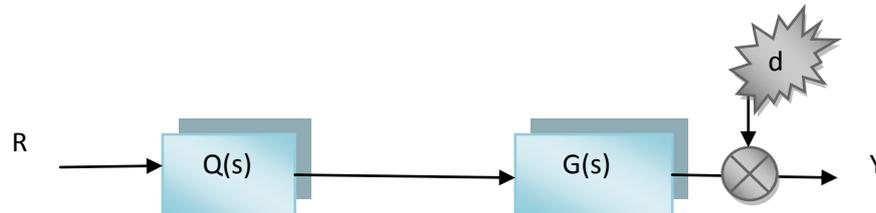


Figure (3.2) : structure de commande en boucle ouverte

$$Y = Q * G * R = R ; \text{ pour } d=0$$

- Cette commande, parfaite, ne peut être réalisée, sachant, qu'en pratique, on ne peut jamais avoir un modèle mathématique exact à 100%, et la sortie sera sensible aux erreurs de modélisation.

$$Y = G * Q * R = (1 + \Delta G) * R$$

G est représenté par son modèle nominale avec une erreur de modélisation multiplicative $G = G_m * (1 + \Delta G)$ $Q = G_m^{-1}$

- En plus, le processus est soumis à des perturbations extérieures inconnues. Supposant $G = G_m$ et $d \neq 0$:

$$Y = d + R$$

La sortie est biaisée par les perturbations extérieures, que le régulateur ne peut compenser, parce qu'on n'a pas de retour de sortie (feedback).

- Lorsque G est strictement propre ou il est à phase non minimale (régulateur instable), la réversibilité de G devient irréalisable.

Pour résoudre les problèmes liés à la sensibilité aux erreurs de modélisation et perturbations extérieures, la structure Feedback conventionnelle est utilisée. Le retour de la sortie permet au régulateur de compenser l'effet de ces perturbations sur la sortie, et la ramènera à la référence.

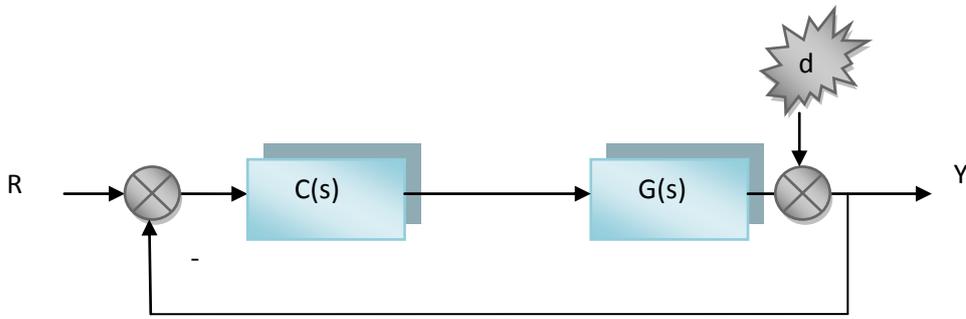


Figure (3.3) : structure de commande Feedback classique

- Le retour de sortie peut entrainer le système bouclé à une instabilité. alors la conception du correcteur stabilisant la boucle devient une tache difficile.

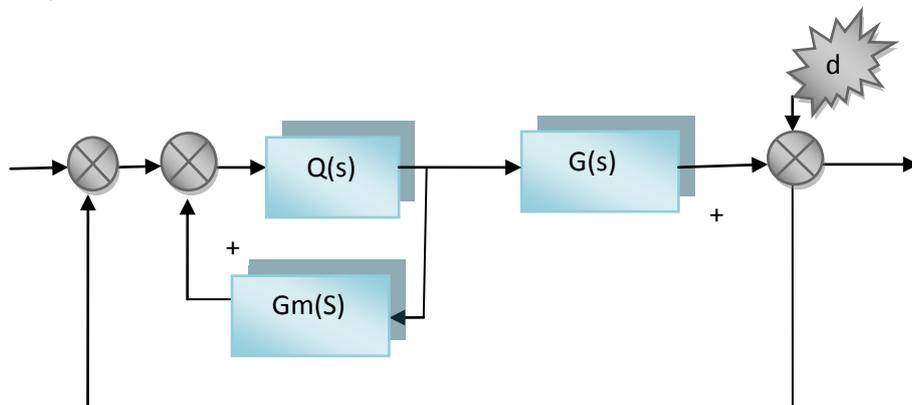
En effet la structure qui combine les avantages des deux structures précédentes est la structure IMC. En absence d'erreurs de modélisation, le signal de retour devient nul, et la structure IMC devient équivalente à une structure de régulation en boucle ouverte. Figure (3.1). Nous aurons un régulateur, simple à concevoir, qui découle directement du modèle de système. Le retour du signal d'erreur permet de compenser l'effet des perturbations et de réaliser la robustesse.

Saturation de la commande :

Dans le cas des commandes saturées, la structure IMC est plus adaptée que la structure Feedback classique, car la sortie du contrôleur (u) est appliquée au procédé et au modèle. Ce qui permet, malgré la saturation, de modifier la consigne, à l'aide de l'écart toujours représentatif de l'évolution entre la sortie du modèle et celle du procédé.

III.3 Structure Feedback équivalente :

Une équivalence peut être établie entre la structure IMC et la structure Feedback conventionnelle, suivant un certain nombre de modifications du schéma fonctionnel, comme suit :



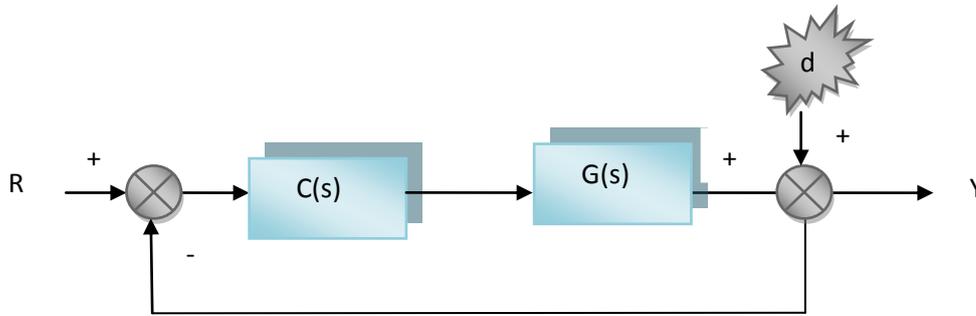


Figure (3.4) : structure Feedback équivalente

$$C=Q/ (1-Gm *Q) \quad (3.1)$$

L'inverse des transformations donne :

$$Q= C/ (1+Gm *Q) \quad (3.2)$$

Si la commande est saturée, la structure de la commande à modèle interne n'est plus équivalente à la structure Feedback conventionnelle.

III.4 La fonction de transfert de la structure IMC :

Pour la détermination de la fonction de transfert du système en boucle fermée, on peut utiliser directement la structure IMC ou la structure Feedback équivalente :

$$Y= d + G *U \quad (3.4)$$

$$U= Q*(R-Yd) \quad (3.5)$$

$$Yd=Gm *U -Y \quad (3.6)$$

Dans (3.5), nous remplaçons Yd par son égalité en (3.6). Ce qui nous donne une égalité de U qui remplacera U de (3.4). Nous obtenons :

$$Y=T(s) *R + S(s) *d \quad (3.7)$$

$$T(s) = \frac{Q*G}{1+Q (G-Gm)} \quad (3.8.a)$$

$$S(s) = \frac{1-Q*Gm}{1+Q (G-Gm)} \quad (3.8.b)$$

$$S(s)+T(s)=1 \quad (3.8.c)$$

T(s) est la fonction de sensibilité inverse.

S(s) est la fonction de sensibilité de la sortie par rapport aux perturbations.

III.5 Propriétés de cette structure (IMC) :

- **La stabilité :**

Si $G=G_m$ le système bouclé se comporte comme un système en boucle ouverte, et sa stabilité est assurée par la stabilité du modèle G et du contrôleur. Les équations de (3.7) à (3.8) donnent :

$$Y = Q * G * R + (1 - Q * G) * d$$

Pour un système stable en boucle ouverte, la structure IMC assure la stabilité de la boucle fermée, pourvue que Q soit stable. Ce qui n'est pas le cas avec la boucle feedback classique.

- **Le contrôle parfait :**

Le choix du contrôleur IMC correspondant à $Q = G_m^{-1}$ permet l'obtention des performances idéales, même en présence d'erreurs de modélisation. Les équations de (3.7) à (3.8) donnent :

$$T(s) = 1 \text{ et } S(s) = 0 \quad \forall s$$

$$Y = R$$

- **Système de type I :**

Si les équations (3.8) sont stables (système stable en boucle fermée), et le régulateur Q choisi a un gain en régime permanent égale à l'inverse de $G_m(0)$:

$$Q(0) = G_m^{-1}(0) \quad (3.9)$$

Soit $e(t)$ l'erreur entre la référence R et le signal de sortie Y :

$$e(s) = R - Y = \frac{1 - G_m * Q}{1 + Q(G - G_m)} * (R - d) = S(s) * (R - d) \quad (3.10)$$

Pour une entrée de R ou d de type échelon :

$$R(s) - d(s) = \frac{c}{s}$$

Par application du théorème de la valeur finale, d'après les équations (3.10) et (3.9)

$$\lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} s * e(s) = \lim_{s \rightarrow 0} c * S(s) = \lim_{s \rightarrow 0} c * \frac{1 - G_m * Q}{1 + Q(G - G_m)} = 0 \quad (3.11)$$

Nous constatons que la fonction de sensibilité devient nulle en régime permanent. Nous obtenons une poursuite parfaite. ($T(0) = 1 - S(0) = 1$). Dans ce cas, nous disons que le système en boucle fermée est de type 1.

- **Système de type 2:**

Si l'égalité (3.9), la condition de stabilité, et l'équation (3.12) sont vérifiées, la poursuite est parfaite, en régime permanent, pour une entrée de type rampe. Le système est dit de type 2. (Brosilow, 1983).

$$\left. \frac{dGm*Q}{ds} \right|_{s=0} = 0 \quad (3.12)$$

III.6 Le contrôleur IMC :

Dans la pratique La propriété du contrôle parfait de la structure IMC ne peut être réalisée, car $Q = G^{-1}m$ n'est pas réalisable dans la majorité des cas de régulation de processus réels, à cause de :

- Situation où G_m est à phase non minimale : les zéro instables de $G_m(s)$ seront des pôles instables de $Q(s)$, alors le système en boucle fermé devient instable (car sortie du régulateur U divergente).
- La majorité des systèmes physiques ou chimiques, ...etc., ont un temps mort t_m représenté dans le domaine de Laplace par : e^{-s*t_m} . Lorsqu'on inverse $G_m(s)$ on retrouve e^{+s*t_m} dans $Q(s)$, qui représente une prédiction dans le domaine temporel, celle-ci est irréalisable.
- Si G_m possède un degré relatif différent de zéro alors $Q(s)$ est impropre. Une faible perturbation en haute fréquence déstabilise le système $\lim_{s \rightarrow \infty} Q(s) = \infty$

III.6.1 Les conditions sur le contrôleur :

Vu les problèmes de divergence et de réalisabilité du régulateur, ce dernier doit obéir à certaines conditions :

Pour éviter les problèmes de divergences et de non réalisabilité du contrôleur, ce dernier doit être :

- stable de type BIBO (bounded input bounded output), c'est-à-dire il ne doit pas contenir des pôles dans le demi plan droit.
- propre ou strictement propre.
- Causal, et ne doit en aucun cas contenir de prédiction.

III.6.2 Calcul du contrôleur :

Le calcul du contrôleur IMC se fait en deux étapes. Dans la première étape on ne s'intéresse qu'aux performances, sans tenir compte des erreurs de modélisation. Nous aboutissons, à la fin de cette étape, à $Q' = G^{-1}m$. (Rivera et Morari. 1986).

Ce n'est qu'en deuxième étape qu'on introduit un filtre passe bas (f) dit filtre de robustesse. À cette étape, nous essayons de réaliser le meilleur compromis entre robustesse et performances, par le biais d'un seul paramètre. (Rivera et Morari. 1986).

Finalement, le contrôleur aura la forme suivante : $Q = f * Q' = f * G^{-1}m$.

III.6.2.1 Erreur de modélisation

Pour ce qui suit, l'erreur de modélisation utilisée est l'erreur de modélisation multiplicative

$$G = G_m * (1 + \Delta G) \quad (3.13)$$

Avec :

$$\begin{cases} |\Delta G| < B(w) \quad \forall w \\ G_m(s) \text{ contient tous les pôles instable du système réel} \end{cases}$$

Remarque : les erreurs de modélisation deviennent importantes en hautes fréquences.

La fonction de sensibilité et de sensibilité inverse sont obtenues avec la prise en considération des erreurs de modélisations, comme suit :

$$S(s) = \frac{1 - G_m * Q}{1 + Q(G - G_m)} = \frac{1 - G_{+m} * f}{1 + G_m^+ * (\Delta G) * f} \quad (3.14, a)$$

$$T(s) = 1 - S'(s) = 1 - \frac{1 - G_{+m} * f}{1 + G_{+m} * (\Delta G) * f} = \frac{G_{+m} * f * (1 + \Delta G)}{1 + G_{+m} * (\Delta G) * f} \quad (3.14, b)$$

III.6.2.2 la factorisation de G_m(s) :

La factorisation est une étape importante, car elle permet de réaliser les conditions énumérées dans (3.6.1) qui assurent la stabilité et la réalisabilité de Q(s). à cette étape, on considère que le modèle mathématique est exacte, $\Delta G = 0$.

On factorise G_m(s) en deux termes :

$$G_m(s) = G_{-m}(s) * G_{+m}(s) \quad (3.15)$$

G_{-m}(s) contient tous les termes à phase minimale de G_m.

G_{+m}(s) contient tous les éléments qui donnent une phase non minimale (temps mort, tous les zéros instables de G_m(s)). Les relations (3.8) donnent :

$$S(s) = 1 - G_m^+ * f \quad (3.16, a)$$

$$T(s) = G_{+m} * f \quad (3.16, b)$$

Le choix de G_{+m}(s) :

Pour réaliser les performances (but de cette phase), il faut que T(s)=1, ce qui donne $|G_{+m} * f|=1$

Morari et Holt (1984,1985) ont proposé un filtre unité f=1, sous hypothèse que G=G_m (on ne se préoccupe pas à la robustesse), et deux possibilités de G_{+m}, selon le critère choisi, pour minimiser l'erreur entre la référence et la sortie:

Pour minimiser le critère Integral square error (ISE), ils proposent la structure de G_{+m} suivante:

$$G_{+m} = e^{-Tm*s} * \prod_i \frac{-\beta_i*s+1}{\beta_i*s+1} \quad (3.17)$$

Dans ce cas, les pôles correspondant au demi-plan gauche, image des zéro du demi-plan droit, seront rajoutés à la dynamique du système en boucle fermée.

Pour minimiser le critère Integral Absolute error (IAE), ils proposent la structure de G_{+m} suivante:

$$G_{+m} = e^{-Tm*s} * \prod_i (-\beta_i * s + 1) \quad (3.18)$$

III.6.2.3 La sélection du filtre f(s) :

A cette étape, nous vérifions le reste des conditions auxquelles le contrôleur doit répondre. Pour rendre ce contrôleur propre, on multiplie $G_{+m}(s)$ par un filtre passe bas.

Selon le type d'entrée d et R (rampe ou échelon), et afin d'assurer une erreur statique nulle nous avons deux type de filtre.

Si les entrées d ou R sont de type échelon (poursuite parfaite de référence de type échelon) il faut vérifier la propriété système type I :

$$G_{+m}(0) * f(0) = 1 \text{ D'après (3.18) et (3.17) on a } G_{+m}(0) = 1 \text{ alors } f(0)=1.$$

Le filtre le plus simple satisfaisant les conditions précédentes est :

$$f = \frac{1}{(\lambda*s+1)^r} \quad (3.19, a)$$

$$\text{Le régulateur finale} \quad Q = \frac{G_{+m}}{(\lambda*s+1)^r} \quad (3.19,b)$$

Si les entrées d ou R sont de type rampe il faut vérifier la propriété système de type II

$$G_{+m}(0) * f(0) = 1$$

$$\left. \frac{dG_{+m} * Q}{ds} \right|_{s=0} = \left. \frac{dG_{+m}(s) * f(s)}{ds} \right|_{s=0} = 0$$

D'après (3.18) et (3.18) on a $G_{+m}(0) = 1$ alors $f(0)=1$.

Voici une forme de filtre satisfaisant ces conditions :

$$f = \frac{r*\lambda*s+1}{(\lambda*s+1)^r} \quad (3.20,a)$$

$$\text{Le régulateur finale} \quad Q = \frac{r*\lambda*s+1}{(\lambda*s+1)^r} * G_{+m}(s) \quad (3.20,b)$$

r : est choisi pour assurer la propreté du contrôleur Q .

λ : Est la constante de temps du filtre, pour réaliser un meilleur compromis entre performance et robustesses.

Jusqu'à cette étape de calcul du contrôleur IMC, nous n'avons pas tenu compte des erreurs de modélisation (modèle mathématique supposé exact). Pratiquement, on ne peut jamais aboutir à une telle certitude et on a toujours des erreurs de modélisation dues aux :

- Des paramètres variant en fonction de l'environnement (procédés biologiques, biochimique, ...).
- La réduction d'ordre du système.

Pour conserver la validité du contrôleur, conçu jusqu'à maintenant (sous hypothèse $\Delta G \approx 0$), il faut s'assurer qu'au moins le modèle est valide dans la plage de fréquence du fonctionnement du système qui nous intéresse.

C'est à ce moment qu'intervient le filtre de robustesse f , par un choix approprié de sa constante de temps λ . nous limitons la bande passante du système en boucle fermée, de telle sorte qu'elle soit à l'intérieure de la plage de fréquence où le modèle représente, le mieux, le système réel. Tout en signalant que cette constante de temps ralentit la dynamique du système bouclé.

En présence d'erreurs de modélisations, la fonction de sensibilité inverse devient :

$$T(s) = \frac{T'(s) * (1 + \Delta G)}{1 + T'(s) * (\Delta G)} \quad (3.21)$$

$T'(s)$ la fonction de sensibilité inverse déterminé en absence d'erreurs de modélisation.

Doyle et Stein (1981), ont montré, que la condition (3.23) doit être vérifiée, pour que le système en boucle fermée soit stable, en présence d'erreurs de modélisation.

$T(j\omega)$ est stable si :

$$|\Delta G| * |T'(j\omega)| < 1 \quad \forall \omega$$

Ce qui implique que $|G_m^+(j\omega) * f(j\omega)| = < 1/B(j\omega)$ (3.22)

Alors il faut que l'ordre du filtre et sa constante de temps réalise la condition (3.22), pour assurer une stabilité robuste.

$B(j\omega)$ devient très important dans les hautes fréquences. Alors il faut choisir une constante de temps suffisamment importante pour limiter la bande passante du système en boucle fermé à une pulsation inférieure à ω_e .

$$|B(j\omega)| \ll 1 \quad \forall \omega < \omega_e$$

Mais il ne faut pas que cette constante de temps soit très grande parce qu'elle introduit une dynamique lente dans la boucle fermée.

La meilleure constante de temps est celle qui réalise le meilleur compromis entre robustesse et performance

III.7 Utilisation du contrôleur à modèle interne pour la conception de PID :

Le développement des calculateurs et la rapidité des ordinateurs n'ont pas ralenti l'accroissement de l'utilisation des régulateurs PID en industrie, ce régulateur qui a fait l'objet de plusieurs recherches qui ont donné lieu à différentes méthodes de calculs : (Zigler et Nichols, 1943), (Cohen et Coon, 1953), (Smith et Corio, 1985), (Rivera et Al, 1986).

Morari et Zafiriou (1989), et Rivera et Al (1986), ont montré comment approximer un contrôleur IMC par un régulateur PI ou PID pour certaines classes de systèmes linéaires et stables.

L'algorithme utilisé est basé sur l'approximation du contrôleur IMC par un I, PI, PID en utilisant les séries de Maclaurin, les paramètres du régulateur obtenus sont fonction d'un seul paramètre ajustable : la constante du temps du filtre.

Cette constante comme indiqué précédemment est choisie en fonction de la gamme de fréquence pour laquelle le système est bien connu, ou selon les l'amplification maximale tolérée du bruit dans le cas où $\Delta G = 0$.

III.7.1 Calcul du régulateur PID :

Le passage de la structure IMC à la structure feedback classique équivalente a permis d'exprimer le contrôleur IMC équivalent C :

$$C = Q / (1 - G_m * Q)$$

Le calcul de Q pour la poursuite parfaite d'entrée de type échelon comme indiqué dans (3.20,a) donne :

$$Q = G_{-m}(s) / (s * \lambda + 1)^r$$

En substituant cette dernière expression dans celle du contrôleur feedback classique C, la formule (3.1), on trouve :

$$C = \frac{G_{-m}(s)^{-1}}{(s * \lambda + 1)^r - G_{+m}(s)}$$

De () on a $G_{+m}(0) = 1$ alors

$$D(s)|_{s=0} = (s * \lambda + 1)^r - G_{+m}(s)|_{s=0} = 0$$

Du résultat précédent on conclut que C a un pôle à l'origine et peut être exprimée de la façon suivante :

$$C(s) = f(s) / s$$

Le développement de C(s) en série de Maclaurin donne :

$$C(s) = (1/s) * (f(0) + f'(0) * s + f''(0) * s^2/2 + \dots + o(S^n))$$

$$C(s) = K_c * (1 + 1/(\tau_i * s) + s * \tau_d)$$

Par identification avec la structure d'un régulateur PID, on détermine la forme analytique des paramètres du régulateur PID, qui ne dépend que du modèle et la constante de temps du filtre.

$$f(0) = \frac{Kp}{D(0)}$$

$$f'(0) = -\frac{G'_{-m}(0)D(0) + KpD'(0)}{(KpD(0))^2}$$

$$f''(0) = f'(0) * \left(\frac{G''_{-m}(0)D(0) + 2G'_{-m}(0)D'(0) + KpD''(0)}{G'_{-m}(0)D(0) + KpD'(0)} + \frac{2f'(0)}{f(0)} \right)$$

$Kp = G_{-m}(0)$

III.7.2 Elimination des paramètres négatifs du PID à base d'IMC :

Dans le cas des systèmes ayant des zéros dominant la méthode de calcul précédente donne des constantes d'intégration τ_i et de dérivation τ_d négatives, pour éviter cela on approxime le contrôleur IMC équivalent par un PID en cascade avec un filtre passe bas $1/h(s)$.

$$h(s) = 1 + \gamma * s$$

On a
$$C(s) = \frac{f(s)}{s} = f(s) * \frac{h(s)}{h(s)*s}$$

Le développement donne :

$$C(s) = \frac{f(0) + (f'(0) + \gamma f(0))s + \frac{(f''(0) + 2\gamma f'(0))s^2}{2} + \frac{(f'''(0) + 3\gamma f''(0))s^3}{6} + O(s^4)}{s(\gamma s + 1)}$$

γ Prend la valeur qui annule le coefficient du terme d'ordre 3 :
$$\gamma = -\frac{f'''(0)}{3f''(0)}$$

III.7.3 Exemple Comparaison entre IMC et PID à base d'IMC, et entre PID à base d'IMC et PID Calculé par d'autres méthodes.

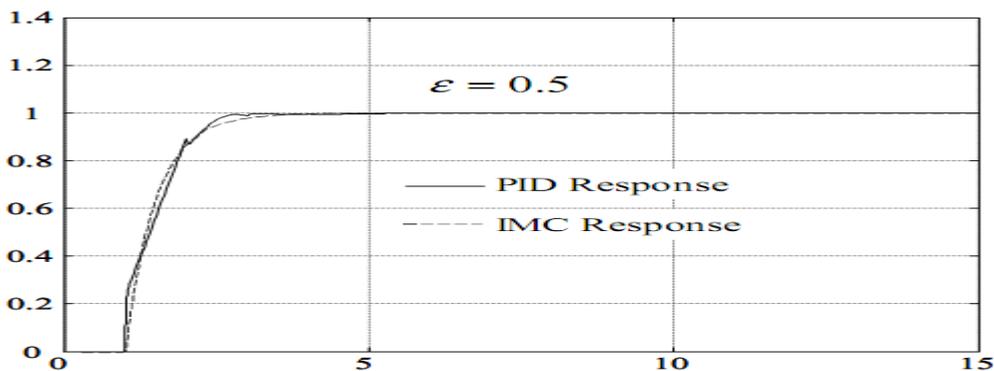


Figure (3.5) : comparaison entre régulateur PID basé sur IMC et implémentation directe de l'IMC.

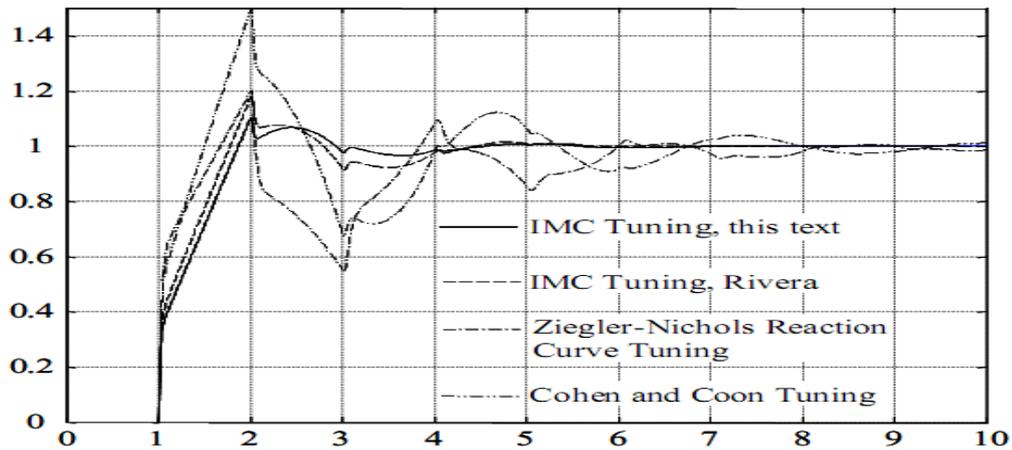


Figure (3.6) comparaison de performances des PID conçu par des méthodes différentes. (Rivera et Morari, 1986).

Il n'y a pas une différence importante entre les performances du système bouclé avec implémentation d'IMC ou PID basé sur IMC pour une constante de temps suffisamment importante. Le développement en série de Maclaurin devient plus précis.

Il est claire d'après les simulations précédentes que le PID à base IMC est meilleur que ceux obtenu par d'autres méthodes.

III.8 Application du contrôleur PI basé IMC au processus de fermentation continue d'acide lactique:

Pour ce type de système à "input multiplicities", il est intéressant de tester, le contrôleur PI linéaire à base du contrôleur à modèle interne, sur les deux points d'équilibres différents correspondants à la même sortie, et voir comment évoluera le système bouclé, si on utilise, un contrôleur, calculé autour du point de fonctionnement correspondant au plus faible taux de dilution, sur le système se trouvant à l'état correspondant au plus grand taux de dilution, et vis versa.

Pour la sortie $\bar{Q} = 4g/l \cdot h$, correspondons deux valeurs de commande $\bar{D1} = 0.0927h^{-1}$, $\bar{D2} = 0.2505 h^{-1}$, obtenus par la résolution de l'équation (2.18), qui exprime la relation non linéaire entre l'entrée et la sortie du processus de fermentation, à l'équilibre.

La commande d'un système non linéaire par un régulateur PI linéaire se fait selon les étapes suivantes :

111.8 1 Linéarisation du système non linéaire autour des points de fonctionnement choisi.

Dans cette étape on approxime le comportement du système non linéaire, au tour d'un point de fonctionnement (\bar{D}, \bar{Q}) , par un système linéaire. Nous utilisons, la représentation d'état linéaire (2.22), trouvée dans le chapitre précédent.

On obtient l'approximation linéaire suivante :

$$\frac{\Delta Q}{\Delta D} = F(s)$$

Pour le premier point d'équilibre noté : $P1 = (\overline{D1}, \overline{Q}) = (0.0927 \text{ h}^{-1}, 4 \text{ g/l}^*\text{h})$, le modèle linéarisé obtenu est le suivant :

$$F1(s) = \frac{43.134 * (S + 0.8781) * (S^2 - 0.0138 * S + 0.07506)}{(S + 0.7716) * (S + 0.0927)^2}$$

Pour le deuxième point d'équilibre, noté : $P2 = (\overline{D2}, \overline{Q}) = (0.2505 \text{ h}^{-1}, 4 \text{ g/l}^*\text{h})$, le modèle linéarisé obtenu est le suivant :

$$F2(s) = \frac{-15.9507 * (S + 0.3668) * (S + 0.1128) * (-S + 0.2043)}{(S + 0.02474) * (S + 0.2505)^2}$$

Nous remarquons que le gain statique, de la fonction de transfert du système linéarisé, change de signe, il est positif autour de P1, et il devient négatif autour de P2.

L'interprétation de ces résultats est donnée dans la partie étude du comportement réel du système à l'équilibre, dans le chapitre II.

III.8.2 Calcul du régulateur PI à base d'IMC

On calcule le régulateur PI linéaire, pour le système linéarisé, obtenu dans l'étape 1, comme l'indique l'approche présentée dans (III.7).

III.8.2.1 Calcul du PI linéaire, pour le premier point d'équilibre :

La factorisation $F1(s)$ est faite comme suit :

$$F1_+(s) = (13.32 * S^2 - 0.183 * S + 1)$$

$$F1_-(s) = \frac{428.76 * (1.13 * S + 1)}{(1.296 * S + 1) * (10.288 * S + 1)^2}$$

Le filtre de robustesse est choisi, pour rendre le contrôleur propre, et pour vérifier la propriété du système bouclé, dite système de type 1, car pour ce type de processus c'est la régulation qui est réalisée :

$$f(s) = \frac{1}{(10*s+1)^2}$$

L'expression du régulateur, obtenue est :

$$C1(s) = 0.0414 * \left(1 + \frac{1}{S * 46.1705}\right)$$

On calcul le régulateur PI linéaire, pour le système linéarisé, obtenu dans l'étape 1, comme l'indique l'approche présentée dans (III.7).

III.8.2.2 Calcul du PI linéaire, pour le deuxième point d'équilibre :

La factorisé $F2(s)$ est faite comme suit :

$$F2_+(s) = (-4.89 * S + 1)$$

$$F2_-(s) = \frac{-86.85 * (2.7262 * S + 1) * (8.8652 * S + 1)}{(40.42 * S + 1) * (3.992 * S + 1)^2}$$

Le filtre de robustesse est choisit, pour rendre le contrôleur propre, et pour vérifier la propriété du système bouclé, dite système de type 1, car pour ce type de processus c'est la régulation qui est réalisée :

$$f(s) = \frac{1}{(10*s+1)}$$

L'expression du régulateur, obtenue est :

$$C2(s) = -0.0285 * \left(1 + \frac{1}{S * 36.82}\right)$$

III.8.2.3 Réalisation de la boucle de régulation

On applique ce régulateur au système non linéaire, comme l'indique la structure suivante :

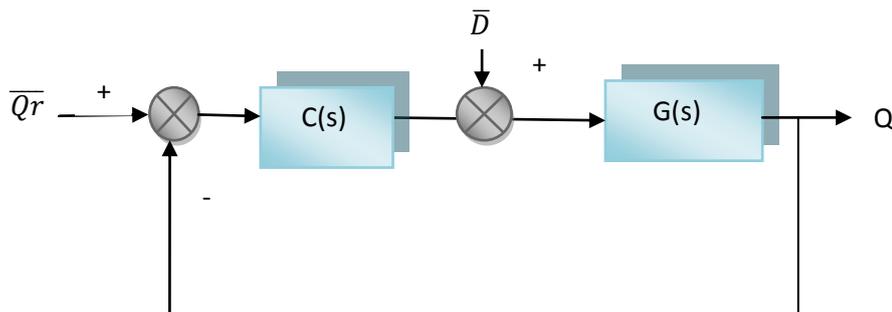


Figure (3.7) : structure de commande du système non linéaire par PI linéaire.

\bar{Y} et \bar{U} Représente le point de fonctionnement autour de lequel on veut travailler.

III.9 Simulations :

Dans cette partie de simulation, nous avons testé, la robustesse, et les performances, de chaque contrôleur C_i , sur le système se trouvant autour du point de fonctionnement P_i , pour lequel il a été calculé, simulations 1, 2, 3, 4, 5, 6, et 7.

Puis, nous nous sommes intéressées, au comportement du système, en boucle fermée avec un PI, calculé pour le taux de dilution le plus faible $\bar{D}1$, et un point de fonctionnement correspondant au taux de dilution le plus fort. Et vis versa.

III.9.1 Testes de robustesse, et de performance du PI linéaire :

Pour tester la robustesse et les performances de ce régulateur, pour les deux points de fonctionnement P1 et P2, nous avons réalisé les simulations suivantes :

III.9.1.1 Simulation 1 : test du rejet de perturbation sur Sin de 50% du PI calculé et appliqué, autour de P1 ($D1=0.0927 \text{ h}^{-1}$).

Une perturbation de + 50% a été appliquée à l'instant $t = 50 \text{ h}$, les résultats obtenus sont les suivants :

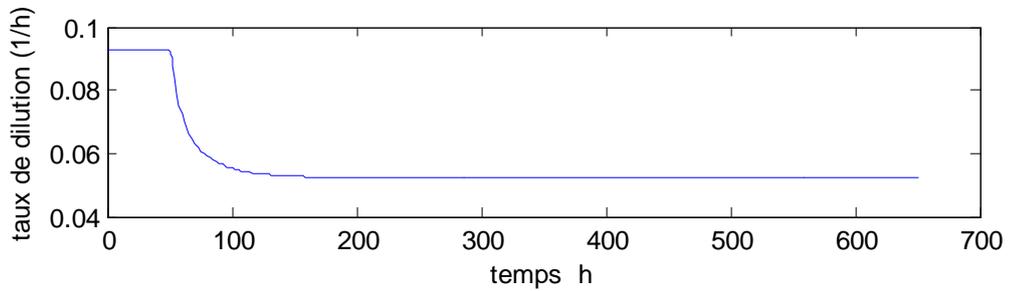


Figure (3.8) : le signal de commande du bioréacteur.

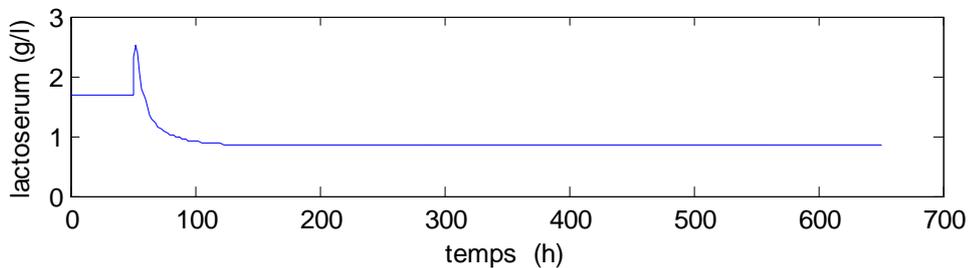


Figure (3.9) : la concentration du substrat dans le bioréacteur.

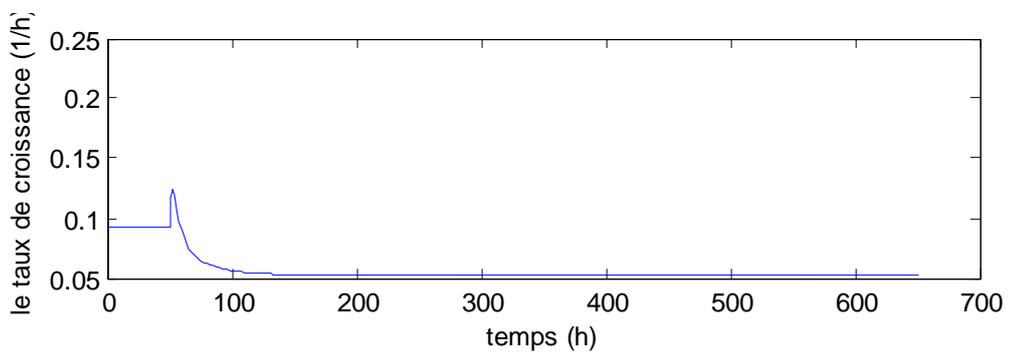


Figure (3.10) : le taux de croissance

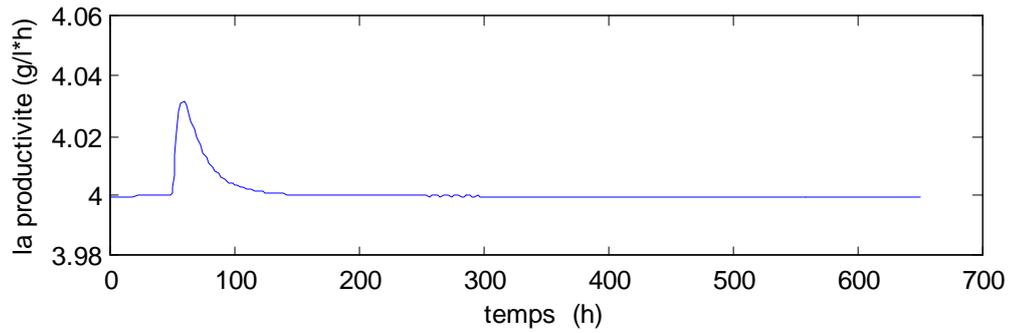


Figure (3.11) : la sortie du bioréacteur

III.9.1.2 Simulation 2 : test du rejet de perturbation sur Sin de 50% du PI calculé et appliqué, autour de $P2$ ($D = 0.2505 \text{ h}^{-1}$). Nous obtenons les résultats suivants :

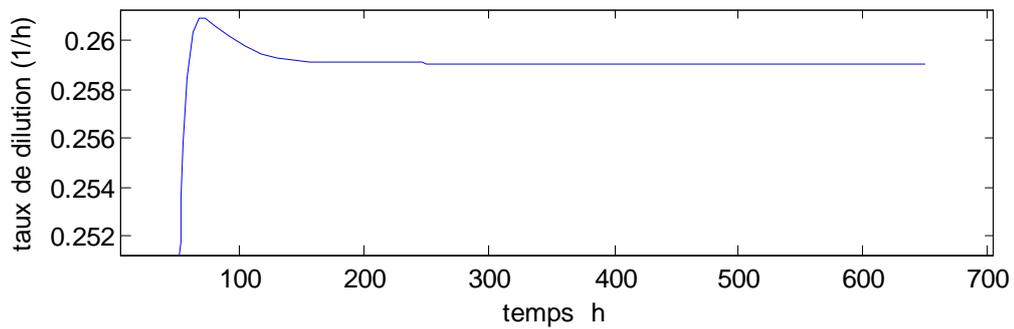


Figure (3.12) : le signal de commande du bioréacteur.

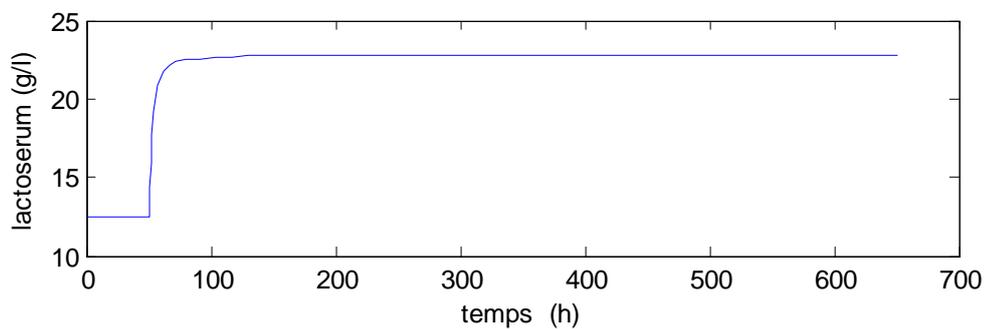


Figure (3.13) : la concentration du substrat dans le bioréacteur

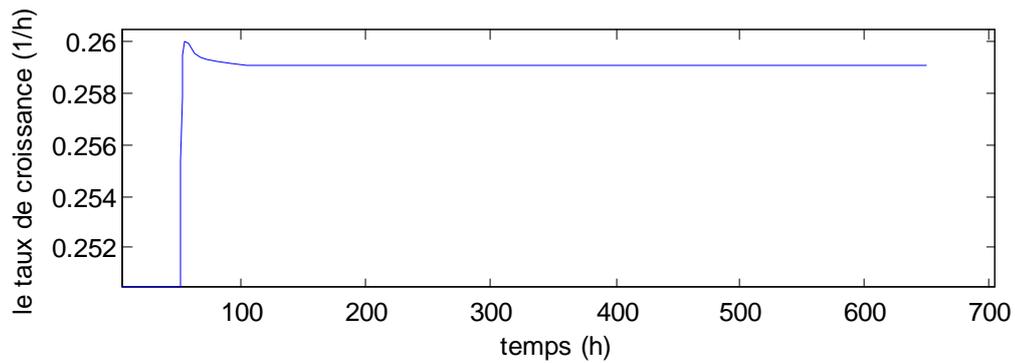


Figure (3.14) : le taux de croissance, pour une perturbation.

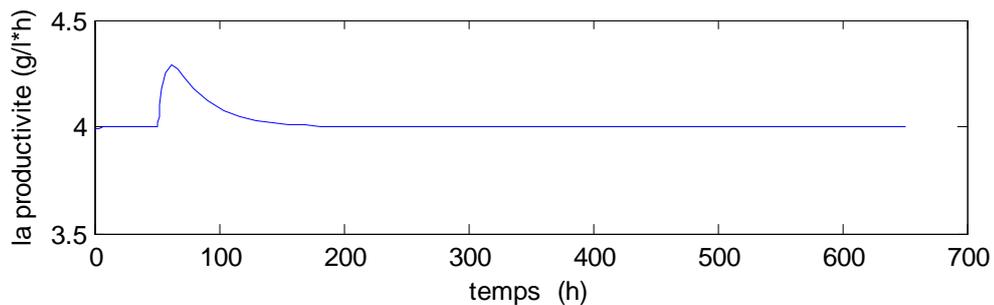


Figure (3.15) : la sortie du bioréacteur.

III.9.1.3 discussions et interprétations des tests de rejets de perturbations sur S_{in} :

Les simulations précédentes, 1 et 2, montrent que :

- le PI linéaire à base d'IMC permet le rejet des perturbations sur S_{in} , mais avec une dynamique lente (environs, 100h pour que la sortie atteigne sa valeur de référence). Figure (3.11) et figure (3.15).
- la commande (taux de dilution) diminue, figure (3.8), lorsque la productivité devient supérieure à la valeur désiré, pour le point de fonctionnement P1.
- la commande (taux de dilution) augmente, figure (3.12), lorsque la productivité devient supérieure à la valeur désiré, pour le point de fonctionnement P2.

Interprétation des simulations 1 et 2 :

- d'après la caractéristique présentée sur, la figure (2.2), la diminution du taux de dilution entraîne, la diminution de la productivité, lorsque le point de fonctionnement du système correspond au taux de dilution le plus faible.
- La figure (2.2) montre aussi, que pour diminuer la productivité, lorsque le point de fonctionnement correspond au taux d dilution le plus fort, il faut augmenter le taux de dilution (commande).
- L'augmentation de la productivité, pour les deux points de fonctionnement, au moment de l'augmentation de S_{in} , est le résultat de l'augmentation du taux de croissance.

Le tracé du modèle de Haldane figure (2.1), montre qu'une augmentation de la concentration du substrat, engendre une augmentation du taux de croissance, si $S < \sqrt{K_s * K_i}$ = 19.68 g/l . Et l'inverse si $S > \sqrt{K_s * K_i}$.

III.9.1.4 Simulation 3 : test du rejet de perturbation, sur la sortie du système, du PI calculé et appliqué, autour de P1.

Une perturbation a été appliquée, sur la sortie du système, égale à 0.1 g/lh à l'instant t=250h. Les résultats de simulations, obtenus sont les suivants

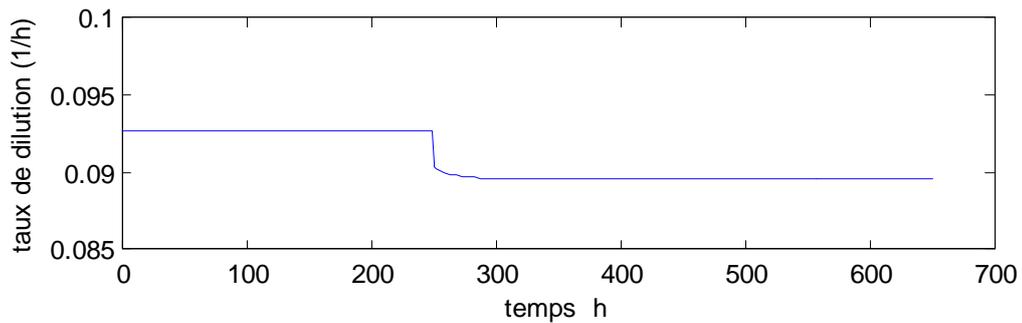


Figure (3.16) : le signal de commande du bioréacteur.

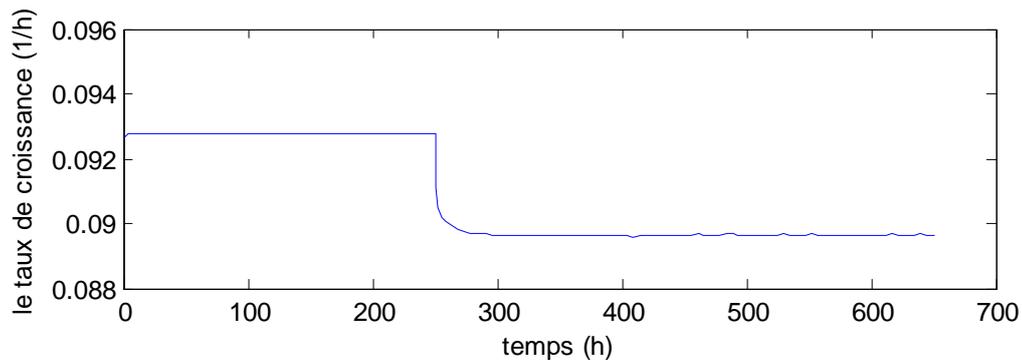


Figure (3.17) : le taux de croissance.

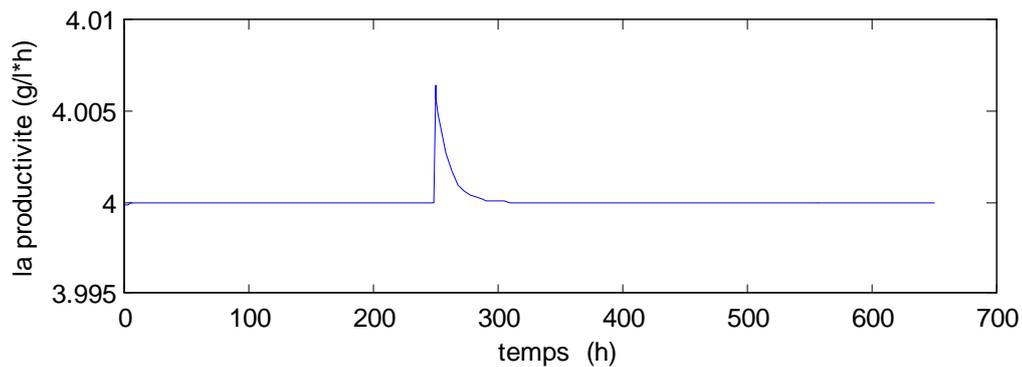


Figure (3.18) : la sortie du bioréacteur.

III.9.1.5 Simulation 4 : test du rejet de perturbation, sur la sortie du système, du PI calculé et appliqué, autour de P2.

Une perturbation a été appliquée, sur la sortie du système, égale à 0.1 g/lh à l'instant $t=250h$. Les résultats de simulations, obtenus sont les suivants :

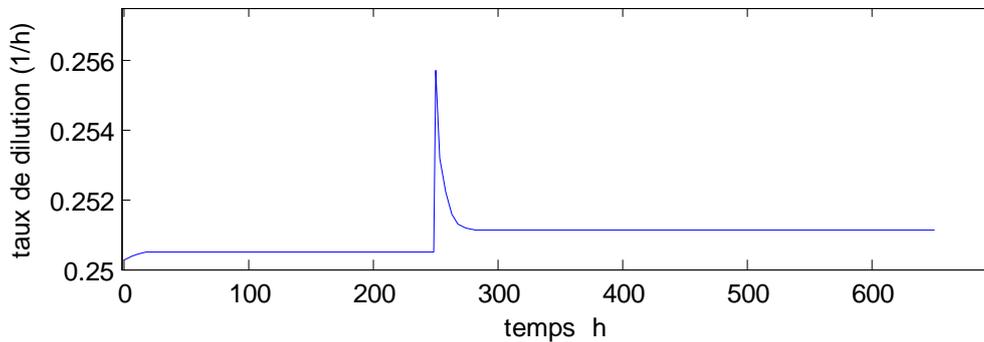


Figure (3.19) : le signal de commande du bioréacteur.

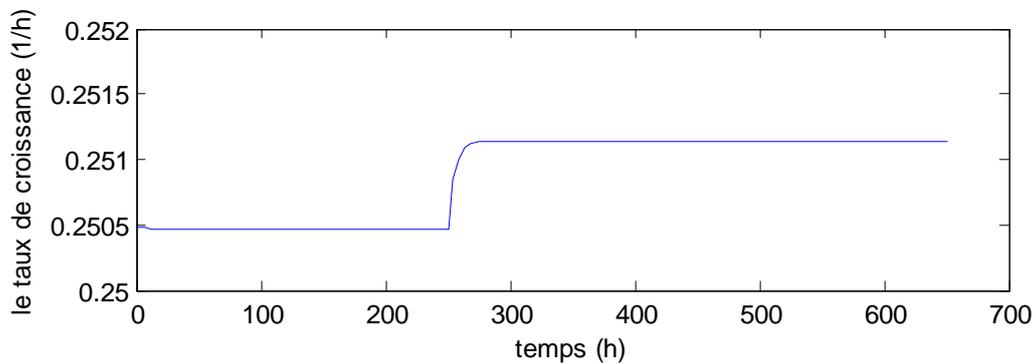


Figure (3.20) : le taux de croissance.

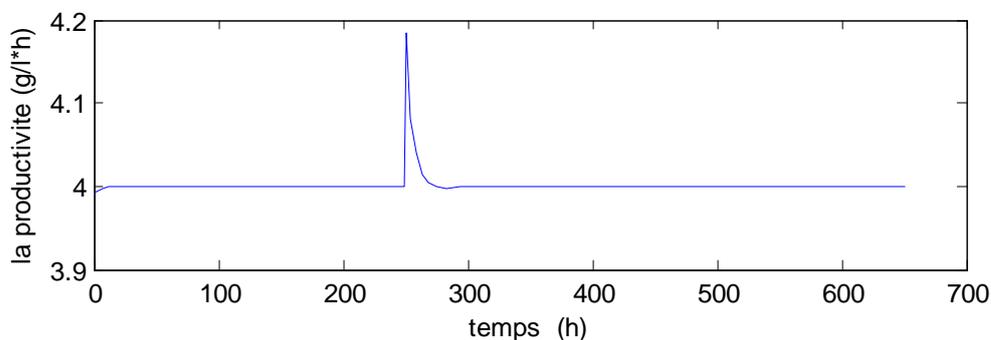


Figure (3.21) : la sortie du bioréacteur.

III.9.1.6 discussions des tests de rejets de perturbations sur Q:

- Les simulations 3, et 4, montrent que le contrôleur PI linéaire, réalise le rejet de perturbation présente sur la sortie du système avec une dynamique lente.
- La figures (3.18), et la figure (3.16), montrent que Lorsque, le système bouclé, dont le point de fonctionnement correspond au plus faible taux de dilution, est soumis à une

perturbation de + 0.1 g/l*h sur la sortie Q, le contrôleur diminue le taux de dilution, pour diminuer la valeur de la sortie, et la ramène à la valeur de référence.

- La figures (3.19), et la figure (3.21), montrent que Lorsque, le système bouclé, dont le point de fonctionnement correspond au plus fort taux de dilution, est soumis à une perturbation de + 0.1 g/l*h sur la sortie Q, le contrôleur augmente le taux de dilution, pour diminuer la valeur de la sortie, et la ramené à la valeur de référence.

III.9.1.7 Simulation 5 : test de poursuite de la référence, du PI calculé et appliqué, pour le taux de dilution $D = 0.0927 \text{ h}^{-1}$.

$$\circ \text{ La sortie désirée} = \begin{cases} 4 & \text{g/lh } 0 < t < 50\text{h} \\ 6 & \text{g/lh } 50\text{h} < t < 400\text{h} \\ 3 & \text{g/lh } t > 400\text{h} \end{cases}$$

Les résultats obtenus sont les suivants :

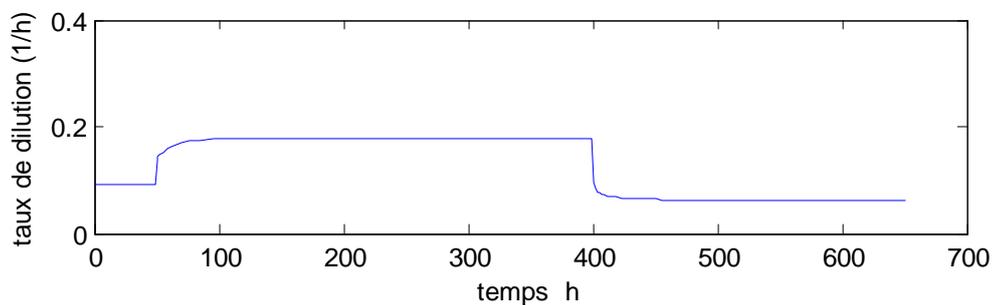


Figure (3.22) : le signal de commande du bioréacteur.

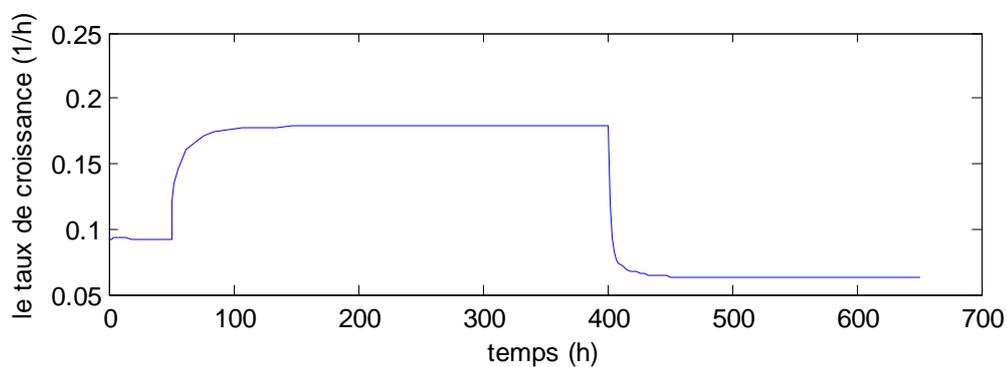


Figure (3.23) : le taux de croissance.

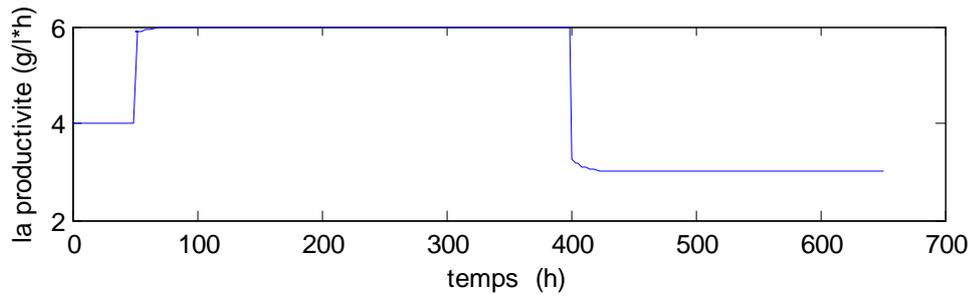


Figure (3.24) : la sortie du bioréacteur..

III.9.1.8 Simulation 6 : test de poursuite de la référence, du PI calculé et appliqué, pour le taux de dilution $D = 0.2505 \text{ h}^{-1}$.

$$\circ \text{ La sortie désirée} = \begin{cases} 4 & \text{g/lh } 0 < t < 50\text{h} \\ 6 & \text{g/lh } 50\text{h} < t < 400\text{h} \\ 3 & \text{g/lh } t > 400\text{h} \end{cases}$$

Les résultats obtenus sont les suivants :

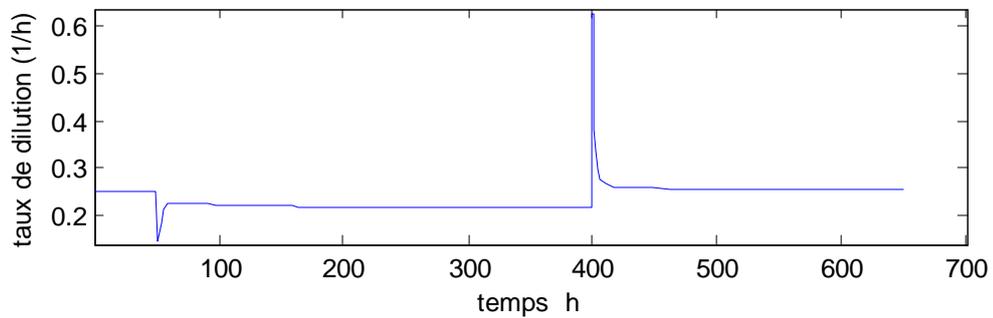


Figure (3.25) : le signal de commande du bioréacteur..

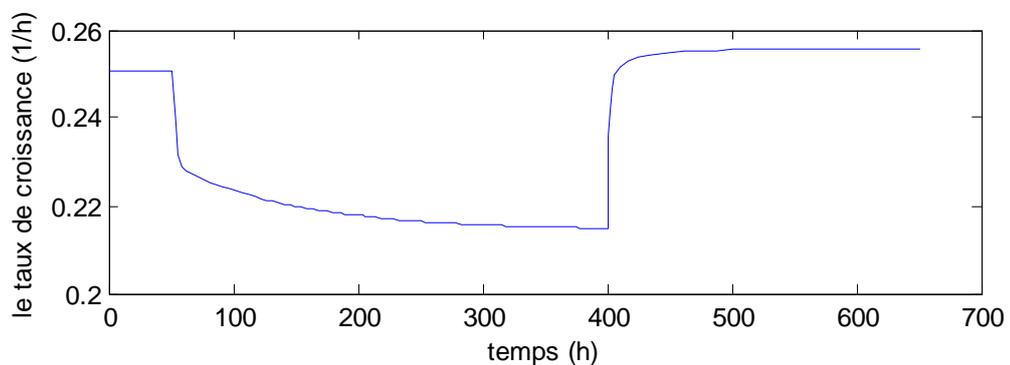


Figure (3.26) : le taux de croissance

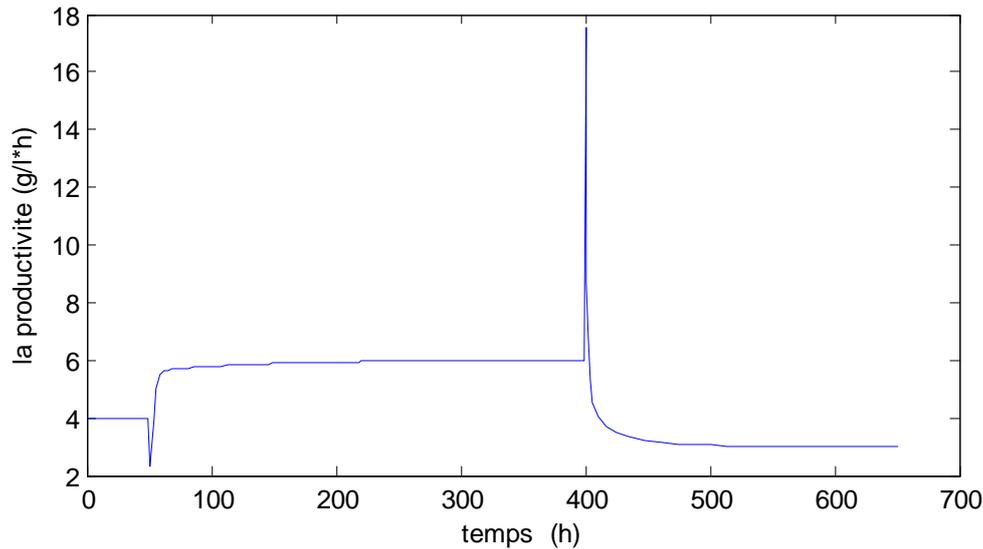


Figure (3.27) : la sortie du bioréacteur..

III.9.1.9 discussions des résultats, des tests de poursuite du PI calculé et appliqué, autour du même point de fonctionnement:

La figure (3.27) montre que le PI linéaire (C2), réalise la poursuite de référence, mais avec des dépassements importants aux moments de variation de la référence. Lorsqu'il est appliqué autour de P2.

La figure (3.25) montre, que le PI linéaire (C1), réalise la poursuite parfaite de référence, lorsqu'il est appliqué autour de P1.

Simulation 7 : test de robustesse vis-à-vis les variations paramétriques, du PI calculé et appliqué, pour le taux de dilution $D = 0.0927 \text{ h}^{-1}$.

Pour une variation de 50% des valeurs des paramètres, du modèle nominale utilisé pour le calcul du contrôleur PI linéaire. Nous obtenons les résultats suivants :

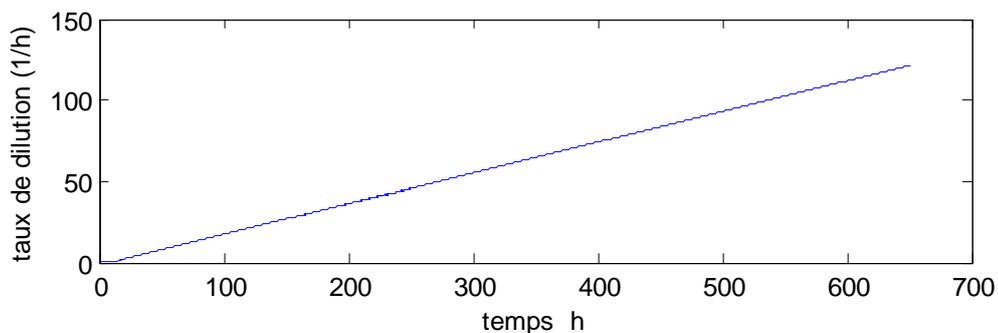


Figure (3.28) : le signal de commande du bioréacteur.

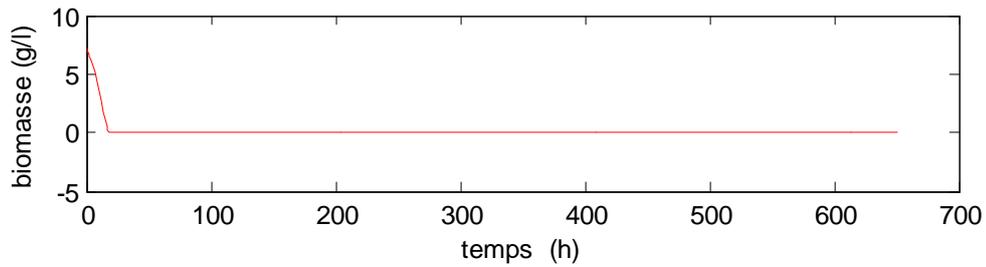


Figure (3.29) : la concentration de la biomasse dans le bioréacteur.

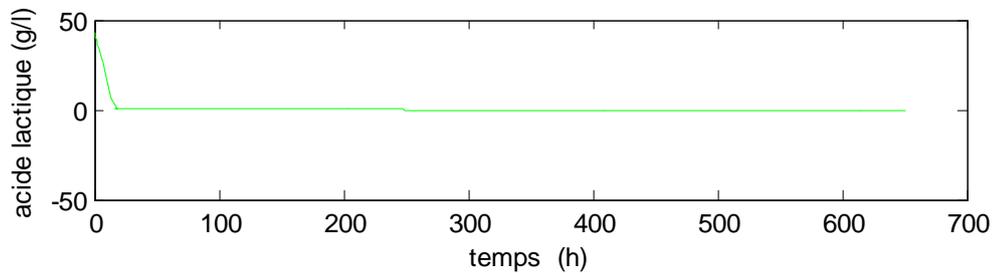


Figure (3.30) : la concentration du produit dans le bioréacteur

III.9.1.10 discussions des résultats du test de robustesse du contrôleur PI linéaire :

Les figures (3.30) et (3.29), montrent qu'une variation paramétrique, entraîne le système bouclé vers le lessivage.

Interprétation :

Le contrôleur PI linéaire à base d'IMC, dépend des paramètres du modèle nominale, ce qui explique, sa sensibilité, aux variations paramétriques.

III.9.2 test de rejet de perturbation sur S_{in} , du régulateur PI linéaire, calculé pour le taux de dilution le plus faible, et appliqué au point de fonctionnement correspondant au taux de dilution le plus fort, Et vis versa.

III.9.2.1 Simulation 8 : test du rejet de perturbation, du PI linéaire, calculé pour le taux de dilution $D = 0.0927 \text{ h}^{-1}$, et appliqué autour du point de fonctionnement P2. Avec une perturbation de + 50% sur S_{in}

Le choix d'une perturbation positive sur S_{in} , appliquée au système commandé par C1, et dont le point de fonctionnement est P2, pour montrer les limites de l'utilisation du régulateur PI linéaire.

Nous avons obtenus les résultats suivants :

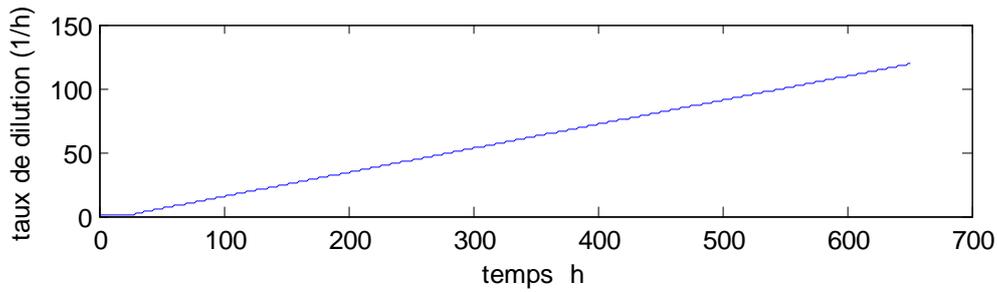


Figure (3.31) : le signal de commande du bioréacteur.

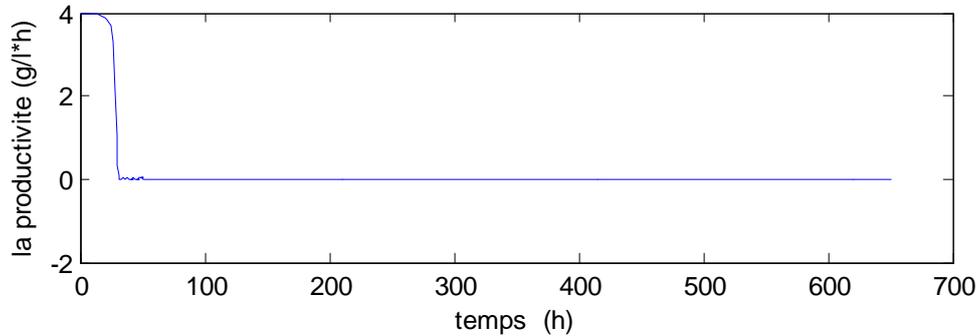


Figure (3.32) : la sortie du processus.

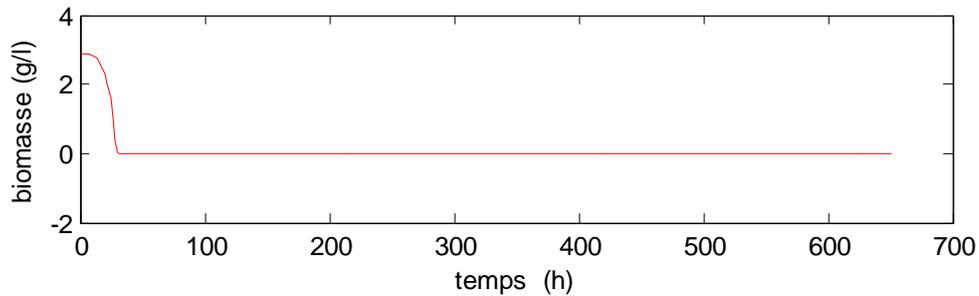


Figure (3.33) : la concentration de la biomasse dans le bioréacteur.

III.9.2.2 Simulation 9 : test du rejet de perturbation, du PI linéaire, calculé pour le taux de dilution $D = 0.2505 \text{ h}^{-1}$, et appliqué pour le taux de dilution $D=0.0927\text{h}^{-1}$. Avec une perturbation de -50% de S_{in}

Le choix d'une perturbation négative sur S_{in} , appliquée au système commandé par C2, et dont le point de fonctionnement est P1, pour montrer les limites de l'utilisation du régulateur PI linéaire.

Nous avons obtenu les résultats suivants :

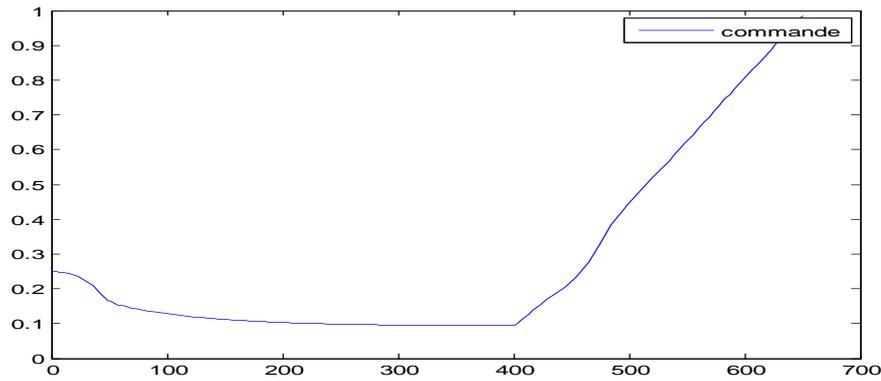


Figure (3.34) : le signal de commande du bioréacteur.

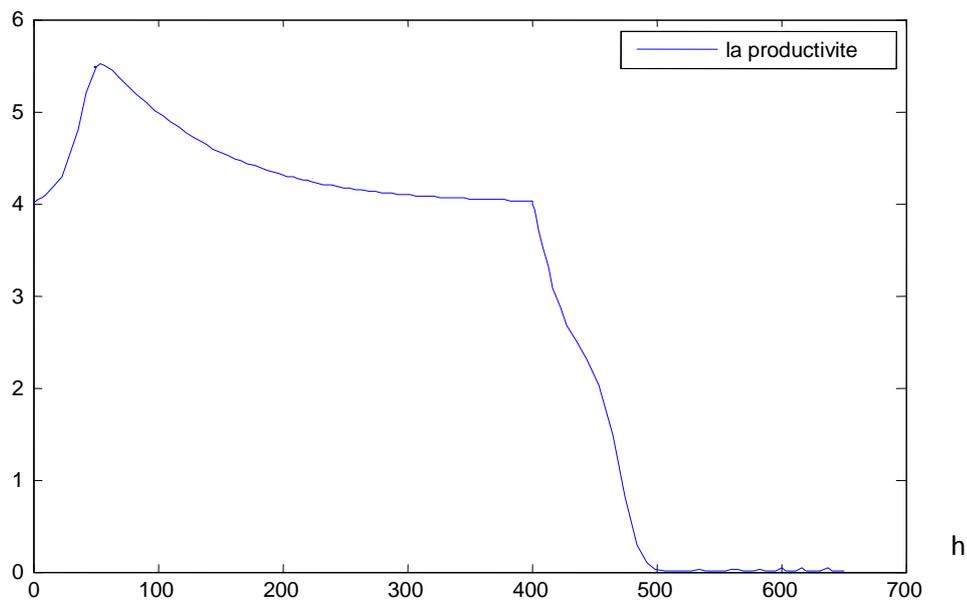


Figure (3.35) : la sortie du bioréacteur. Pour une perturbation sur S_{in} de -50%.

III.9.2.3 discussions et interprétations, des résultats :

La simulation 8, montre, que si le système est ramené au point P1, le régulateur C2, déstabilise la sortie, en présence de perturbation positive sur S_{in} .

Interprétation :

Lorsque le système se trouve dans la partie de la courbe (2.2) correspondante aux taux de dilutions les plus faible, une augmentation de S_{in} , engendre une augmentation de la productivité ($e = Q_r - Q < 0$). Le contrôleur C2 ayant un gain négatif, sa sortie positive s'ajoute à D_r (taux de dilution augmente), dans ce cas la productivité augmente au lieu de diminuer, et l'erreur devient de plus en plus importante. Le système devient instable.

Les résultats de la simulation 9, montre que si le système est ramené au point P2, le régulateur C1, déstabilise la sortie, en présence de perturbation négative sur S_{in} .

Interprétation :

Lorsque le système se trouve dans la partie de la courbe (2.2) correspondante aux taux de dilutions les plus fort, une diminution de S_{in} , engendre une diminution de la productivité ($e = Q_r - Q > 0$). Le contrôleur C1 ayant un gain positif, sa sortie positive s'ajoute à D_r , et la productivité diminue encore figure (3.35), au lieu d'augmenter pour atteindre Q_r . Et le système atteint, le lessivage.

III.10 Conclusions :

Nous avons conclu, à partir des discussions, que :

- Le PI linéaire calculé à base d'IMC, réalise le rejet des perturbations présentes sur la sortie du système, si ce contrôleur, est calculé, et est appliqué autour du même point de fonctionnement.
- Le PI linéaire calculé à base d'IMC, réalise le rejet des perturbations présentes la concentration du substrat d'alimentation du bioréacteur, si ce contrôleur, est calculé, et est appliqué autour du même point de fonctionnement. Mais avec une dynamique très lente.
- Le rejet de perturbations en sortie, se fait avec une dynamique beaucoup plus rapide, que le rejet de perturbations, sur la concentration du substrat d'alimentation du bioréacteur.
- Le PI linéaire n'est pas robuste, aux erreurs de modélisations. Le système bouclé devient instable, en présence de variations paramétrique.
- Lorsque le système est bouclé, avec un contrôleur conçu autour du point de fonctionnement correspondant au taux de dilution le plus faible, et que des perturbations ou variations paramétriques, entraînent l'état du système vers les états correspondants, au taux de dilution les plus fort. Le système bouclé devient instable.
- Lorsque le système est bouclé, avec un contrôleur conçu autour du point de fonctionnement correspondant au taux de dilution le plus fort, et que des perturbations ou variations paramétriques, entraînent l'état du système vers les états correspondants, au taux de dilution les plus faibles. Le système bouclé devient instable.

Vu que ce système est souvent sujet de variations paramétriques, causées par les changements des conditions environnementales, et l'état du système peut facilement basculer entre, les états correspondants aux taux de dilutions les plus faibles, et les états de dilutions les plus fort.

Nous concluons, finalement, que le contrôle de fermentation continue d'acide lactique, par PI linéaire, n'est pas une bonne solution, car il peut déstabiliser le système bouclé et l'entraîner vers le lessivage, de façon irréversible.

Chapitre IV
Réseaux de neurones

IV.1 Introduction :

Le cerveau humain est constitué de plusieurs milliers de milliards d'unités élémentaires, effectuant des tâches très élémentaires (transmission ou inhibition de signaux électriques), appelées neurones. La puissance de cet organe résulte de la façon dont sont organisés et connectés ces neurones entre eux (dix milles connexions par neurone), ces connexions massives, parallèles et non linéaires de neurones lui permettent d'accomplir des tâches très élaborées, de mémoriser, d'apprendre, d'utiliser les connaissances acquises et de s'adapter, ...etc.

La question qui se pose : comment tirer profit du comportement et de la structure du cerveau, pour construire des calculateurs intelligents, qui apprennent et qui peuvent résoudre n'importe quel problème, quelque soit sa complexité ? Tout en étant construit à partir d'unités simples élémentaires et qui effectuent des tâches élémentaires ?

La réponse à cette question, a donné naissance aux réseaux de neurones artificiels **RNA**.

IV.2 Définition réseau de neurones artificiels :

C'est un outil informatique (mathématique) composé d'éléments de calcul interconnectés qui a des capacités d'apprentissage. Les connaissances qu'il acquiert pendant son apprentissage sont stockées de façon distribuée dans le réseau (les poids synaptiques), Ceci lui permet une résistance à la dégradation, une grande capacité de généralisation.

Haykin (1994), a proposé la définition suivante:

« Un réseau de neurones est un processus distribué de manière massivement parallèle, qui a une propension naturelle à mémoriser des connaissances de façon expérimentale et de les rendre disponibles pour utilisation. Il ressemble au cerveau en deux points:

- 1. la connaissance est acquise au travers d'un processus d'apprentissage;*
- 2. les poids des connexions entre les neurones sont utilisés pour mémoriser la connaissance. »*

IV.3 Historique:

La première étude systématique du neurone artificiel a été réalisée par McCulloch et Pitts, en 1943, ils ont réussi à proposer, pour la première fois, un modèle mathématique simplifié du neurone biologique. C'est le neurone formel, ayant la fonction seuil comme fonction d'activation.

Les travaux du physiologiste américain Donald Hebb (1949), sur l'apprentissage, en 1949, décrits dans son ouvrage 'The Organisation of Behaviour', ont permis d'avancer les travaux de MP en introduisant les coefficients synaptiques évolutifs dans les réseaux de neurones, les neurones formels.

Mais ce n'est qu'à la fin des années 50 que le premier modèle opérationnel des réseaux de neurones a eu lieu, par Franck Rosenblatt (1958), en inventant le perceptron. Qui a connu beaucoup de succès et a permis d'attirer un grand nombre de chercheurs à travailler sur les réseaux de neurones.

En parallèle, avec les travaux de ce dernier, Benrad Widrow et ted Hoff (1960), ont introduit un algorithme d'apprentissage du perceptron.

Le succès des recherches, qui ont été menées sur les neurones artificiels et leurs réseaux s'est très rapidement atténué, après les publications de Marvin Minsky et Seymour Papert ,(1969). Qui ont montré que le perceptron était limité à ne résoudre que les problèmes dont les variables sont linéairement séparables. Figure (4.1, A), les limites du perceptron monocouche figure (4.1, A, B)

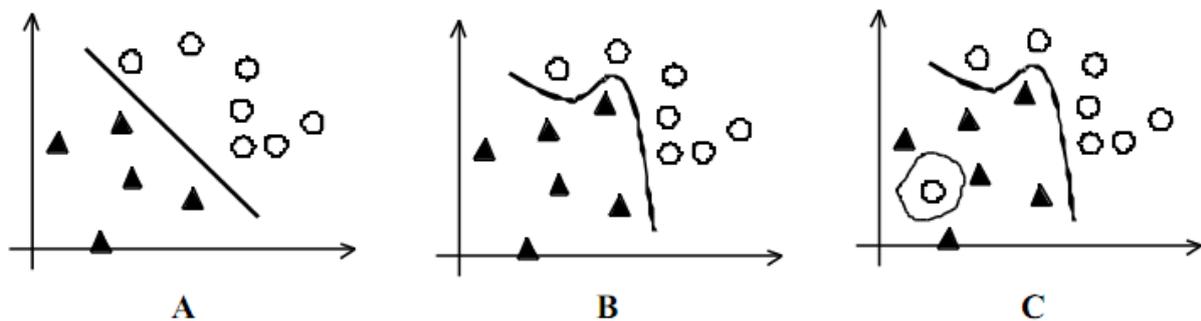


Figure (4.1) A discrimination linéaire, B la discrimination non linéaire, C discrimination discontinus.

Malgré que pas mal de gens ont cru que c'était la fin de cet outil de traitement, et que la plus part des sujets de recherche sur les neurones artificiels étaient suspendus, il y a eu quelques travaux très importants initiés par Tenno Kohonen (1972), et James Anderson (1972), qui ont réussi à développer les réseaux de neurones mémoires associatives.

Au début des années 80, les réseaux de neurones ont connu un essor considérable, pour les raisons suivantes :

L'introduction des mémoires auto-associatives par Hopfield (1982).qui avaient la capacité de restituer une donnée à partir d'informations incomplètes et/ou bruitées.

L'invention de l'algorithme de Backpropagation, permettant l'entraînement des réseaux de neurones multicouches par David Rumelhart et James McClelland (1986), Et le développement des calculateurs, ont donné un nouveau souffle aux recherches sur les réseaux de neurones

IV.4 Applications :

L'intégration des réseaux de neurones dans divers domaines devient de plus en plus large. en voici quelques applications:

- L'identification, la modélisation et la commande des processus.
- En robotique, on trouve les systèmes de vision, le contrôle de la trajectoire, les robots autonomes, ... etc.
- Le traitement du signal, la compression des images et des signaux, la reconnaissance automatique de caractères manuscrits.
- Dans la médecine, la prévision des maladies, et les diagnostics.
- Le guidage automatique d'engins automobiles.
- La reconnaissance de formes (reconnaissance de caractères, signatures, visages, billets, ... etc.), reconnaissance vocale... etc.
- la prédiction (par exemple des mouvements de bourse, par reconnaissance de tendances).

IV.5 Neurone biologique :

Le corps cellulaire : contient le noyau de la cellule nerveuse; c'est en cet endroit que prend naissance l'influx nerveux, qui représente l'état d'activité du neurone.

Les dendrites: ramifications tubulaires courtes, formant une espèce d'arborescence autour du corps cellulaire; ce sont les entrées principales du neurone, qui captent l'information venant d'autres neurones.

L'axone : longue fibre nerveuse qui se ramifie à son extrémité; c'est la sortie du neurone et le support de l'information vers les autres neurones.

La synapse : qui communique l'information entre deux neurones, en la pondérant par un poids synaptique, à un autre neurone.

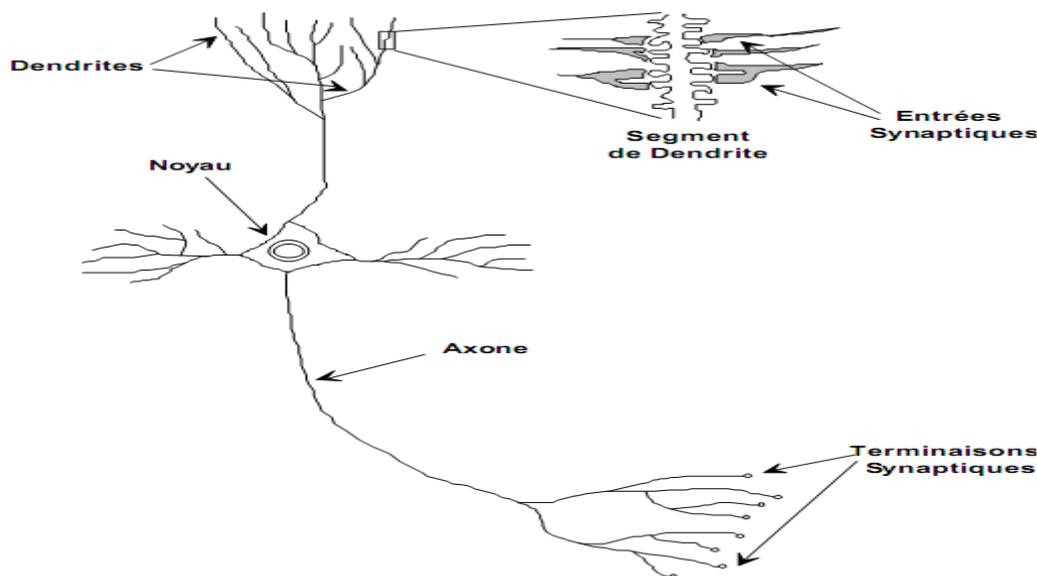


Figure (4.2) neurone biologique

IV.6 Réseau de neurones artificiels:

C'est un processeur parallèle de traitement d'informations distribuées, caractérisé par sa capacité d'adaptation, de généralisation issue de la phase d'apprentissage et sa mémoire distribué sur ces paramètres internes.

Inspiré du cerveau humain, il présente la même structure que ce dernier, mais avec un nombre plus faible d'unités de traitement et de connexions que ceux présents dans le cerveau (dans le cerveau les neurones sont massivement connectés entre eux, environs 10 0 milliards de neurones, et 10 milles connexion par neurone).

L'unité de traitement est la composante élémentaire du réseau, c'est le résultat de la projection d'un neurone biologique dans le domaine des mathématiques, appelée neurone formel.

IV.6.1 Neurone formel :

Le neurone formel est un processeur élémentaire, qui réalise la somme pondérée des signaux qui lui parviennent en entrée additionnés au biais. La sortie du neurone est la transformation de cette somme par la fonction de transfert f appelée fonction d'activation du neurone.

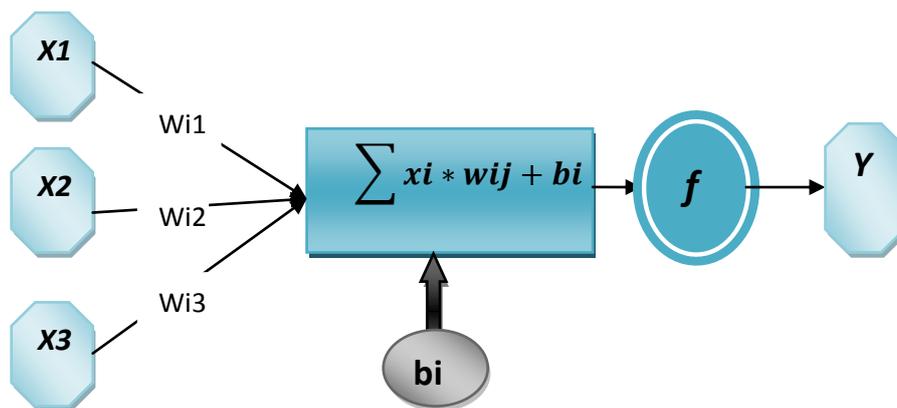


Figure (4.3) modèle du neurone formel.

n^i : est le potentiel d'activation du neurone i .

$$n^i = \sum_{j=1}^n x_j * w_{ij} + b$$

W_{ij} : poids de pondération de la connexion de l'entrée j avec le neurone i ;

x_j : L'entrée j du neurone i .

n : Nombre d'entrées du neurone.

b : seuil (biais).

f : fonction d'activation du neurone.

w_{ij} et b_i sont des paramètres ajustables qui représentent les connaissances mémorisées du neurone. Leurs valeurs nominales sont obtenues pendant la phase d'apprentissage du neurone, ils sont initialisés par le concepteur, puis mis à jour par un algorithme d'adaptation.

IV.6.2 L'analogie avec le neurone biologique:

Le corps cellulaire est modélisé par la sommation et la fonction d'activation.

La sortie du neurone est le signal porté par l'axone.

Les poids w_{ij} , définissant les paramètres du modèle interne du réseau, représentent l'intensité synaptique présente dans le point de jonction de deux neurones biologique.

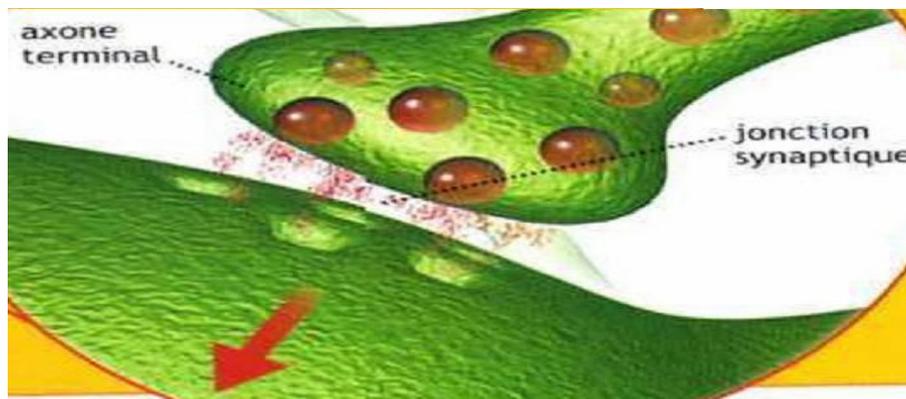


Figure (4.4) : la jonction synaptique entre deux neurones biologique.

Une autre analogie, entre le réseau de neurone artificiel et le cerveau, apparaît au niveau de la jonction synaptique. Pour le cerveau l'intensité de la jonction synaptique est évolutive, elle se renforce si l'activation du premier neurone active le deuxième, et devient de plus en plus faible dans le cas contraire Hubb (1949). Pour les réseaux de neurones artificiels, on trouve le même phénomène, et c'est les poids qui évoluent pour renforcer ou dénier une connexion entre deux neurones formels pendant la phase d'apprentissage du réseau.

IV.6.3 Les différentes fonctions d'activation :

Le choix de la fonction d'activation dépend de la problématique et de l'algorithme d'entraînement utilisé, pour celui qui nécessite des dérivés il faut que la fonction soit continue. En pratique toute fonction croissante, impaire, peut être utilisée, mais le plus souvent on fait appel à des fonctions ramenant les potentiels de sortie à l'intérieur de bornes prédéfinies.

Parmi les fonctions couramment utilisées : Echelon, Signe, Sigmoidale, linéaire,...etc.

Voici les fonctions d'activation les plus utilisées :

IV.7 L'architecture d'un réseau de neurones :

IV.7.1 Les réseaux monocouches :

Ce qu'on appelle couche de neurones est un ensemble de neurones disposés en parallèle, et qui travaillent en même temps. Une couche à 2 neurones est représentée dans la figure (5.7), les neurones de la même couche ne sont pas reliés entre eux dans un réseau feedforward.

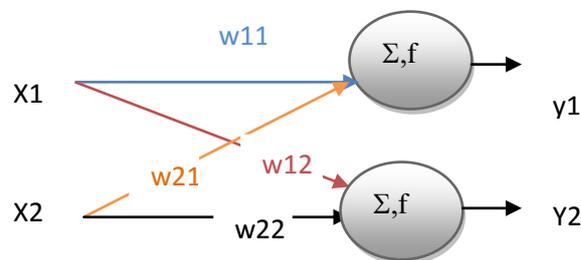


Figure (4.5) : réseau de neurones artificiel monocouche.

IV.7.2 Réseaux multicouches :

C'est des réseaux de neurones formés par la mise en cascade d'un ensemble de couches, où les sorties de chaque couche sont connectées partiellement ou totalement aux entrées de la couche suivante, les entrées de la première couche sont les variables externes liées au problème à résoudre.

Les neurones de la même couche i ne peuvent communiquer qu'avec les neurones de la couche $i+1$ ou ceux de la couche $i-1$. Ce sont des réseaux de type feedforward.

La figure suivante représente un réseau à P couches :

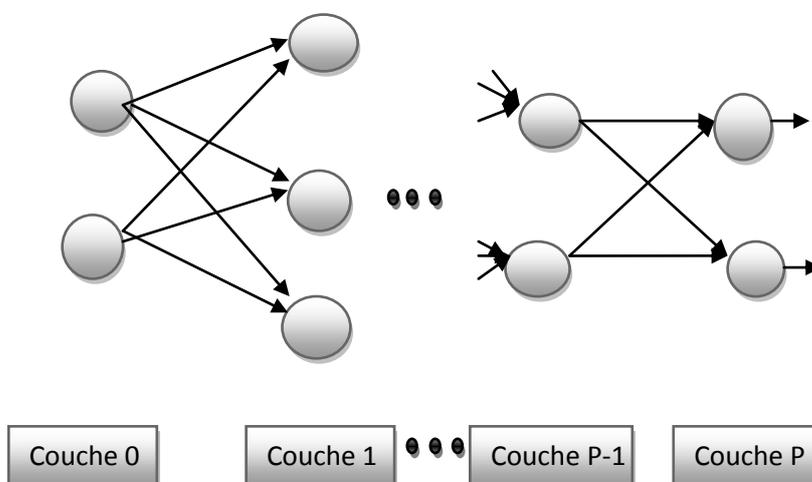


Figure (4.6) : réseau de neurone multi couches.

La couche 0, celle qui reçoit les variables externes, est appelée couche d'entrée du réseau. Elle ne comporte pas de neurones.

La couche P, celle qui fournit les sorties du réseau, est appelée couche de sortie. Les autres couches sont appelées couches cachées.

Le nombre de neurones par couche n'est pas obligatoirement le même, ainsi que le type de fonctions d'activation. Dans la représentation précédente on a des couches à 2 neurones et d'autres à 3.

Les réseaux de neurones multicouches sont plus puissants que ceux avec une seule couche, et peuvent résoudre n'importe quel problème, quelque soit sa complexité ou sa non linéarité, pourvue que le choix du nombre de couches et celui des neurones par couche soit approprié.

Le problème du choix du nombre de neurones dans les couches d'entrée et de sortie ne se pose pas. Il est déterminé par les variables externes à manipuler. Par contre, il n'y a pas de méthodes systématiques pour déterminer le nombre de couches cachées où le nombre de neurones par couche qu'il faut pour une problématique donnée, cette tâche est le résultat de l'expérience et de l'intuition du concepteur.

Dans la pratique les réseaux à 2 et à 3 couches sont les plus utilisés.

IV.7.3 Réseaux de Hopfield (la mémoire auto-associative), (Hopfield,1982):

Ce réseau est une mémoire dite « associative » capable de mémoriser spécifiquement certains états du réseau, grâce à un algorithme d'entraînement. Et Lorsque le réseau est initialisé dans un état différent de ceux qui ont été mémorisés, il évoluera vers l'état mémorisé le plus proche de l'état initial

Pour ce type de réseau mémoire l'évolution se fait de la manière suivante : à chaque itération, l'état de chacun des neurones est recalculé comme étant une fonction d'une somme pondérée de l'état de tous les autres neurones à l'itération précédente. L'état d'un neurone du réseau à un instant donné dépend de l'état de tous les autres.

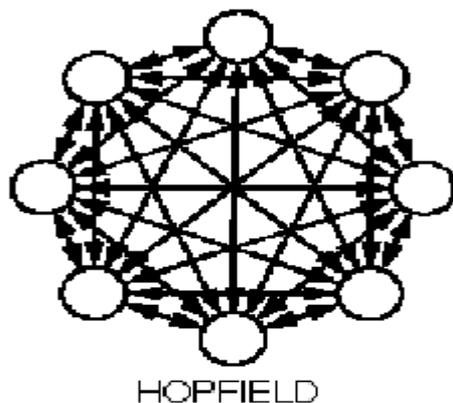


Figure (4.7) : réseau de Hopfield.

IV.7.4 Réseaux récurrents :

Ça peut être un réseau monocouche ou multicouche, avec la particularité de permettre la réinjection des sorties (feedback) des neurones de couches supérieures vers les entrées des couches inférieures, Ou vers les neurones de la même couche. Pour ce type d'architecture chaque neurone peut communiquer avec n'importe quel autre neurone du réseau.

Il est plus puissant que le réseau feedforward, le retard qu'entraîne le feedback lui permet de présenter la notion de temps, et de ce fait de modéliser et résoudre des problèmes qui dépendent explicitement du temps. (Haykin 1994).

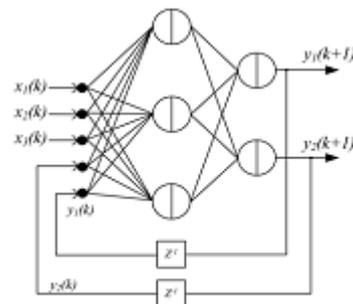


Figure (4.8.a) : structure d'un réseau de neurones bouclé.

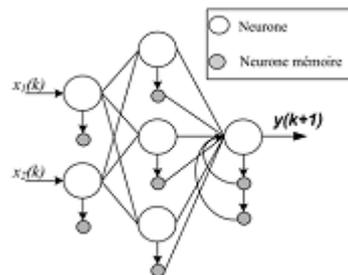


Figure (4.8.b) : structure d'un réseau de neurones à mémoire.

Les sorties du réseau qui retournent doivent être initialisées.

IV.8 L'apprentissage du réseau de neurones artificiels :

L'apprentissage ou l'entraînement du réseau de neurones artificiels peut se faire selon deux modes : supervisé ou non supervisé, selon la disponibilité de l'expert (les sorties désirées) et l'architecture du réseau.

Pendant la phase d'apprentissage, c'est la mémoire du réseau qui évolue, pour permettre à ce dernier de répondre convenablement aux entrées qu'il n'a pas encore rencontrées pendant la phase d'entraînement.

IV.8.1 L'apprentissage supervisé :

Dans ce mode d'apprentissage, on fournit les vecteurs d'entrées et les vecteurs de sorties souhaitées. L'adaptation des paramètres internes du réseau se fait suivant un algorithme

d'optimisation, pour minimiser la fonction coût, qui représente l'erreur entre les sorties du réseau et celles désirées pour un vecteur d'entrée donné.

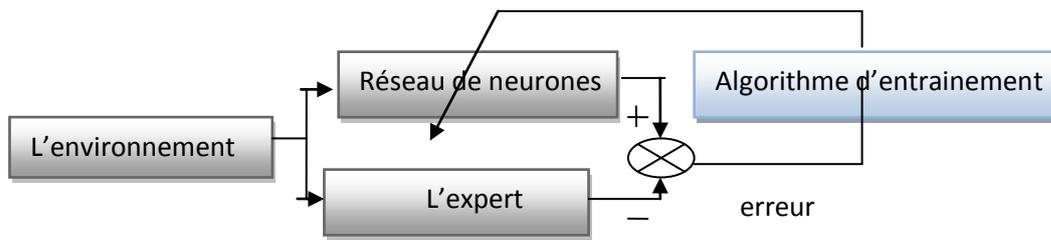


Figure (4.9) : la structure d'apprentissage supervisé

L'expert peut être la fonction à modéliser, ou le système à identifier...etc.

IV.8.2 L'apprentissage non supervisé :

L'algorithme vise à converger vers une solution, c'est-à-dire, la détermination des valeurs de poids qui assurent de bonnes propriétés aux sorties, en accord avec le problème posé. Il est utilisé pour entraîner les réseaux mémoires, la carte auto-organisatrice...etc.

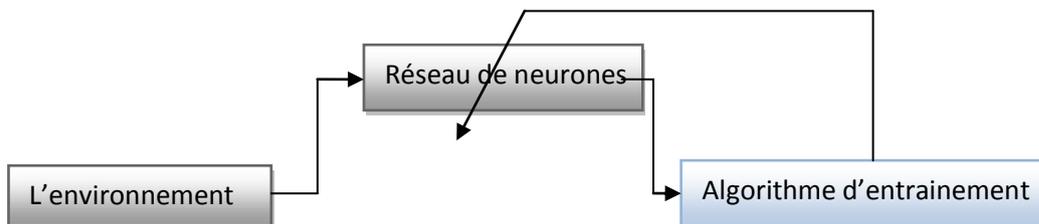


Figure (4.10) : la structure d'apprentissage non supervisé.

IV.8.3 L'optimisation des paramètres:

IV.8.3.1 L'optimisation globale : elle consiste à l'utilisation d'un algorithme itératif d'optimisation, pour adapter tous les poids du réseau en vue de Minimiser un critère choisi selon la fonction que doit réaliser ce réseau, classification, modélisation, reconnaissance...etc.

IV.8.3.2 L'optimisation locale: consiste en l'adaptation d'un nombre réduit de poids des connexions pour chaque entrée. Seuls les poids des connexions des neurones les plus activés par le signal d'entrée sont adaptés. Ex: "apprentissage compétitif".

Exemple : Hebb (1949), Augmentation des poids si co-activité des neurones pré et post-Synaptique.

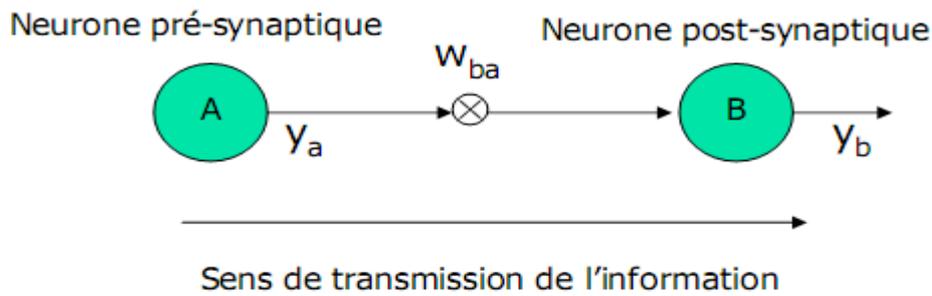


Figure (4.11) : la connexion des neurones dans le modèle de Hebb.

IV.8.4 Algorithmes d'optimisations itératives :

Elle consiste en la recherche et l'exploration, à chaque itération, d'une direction descendante de l'hyper surface de l'erreur, conduisant à un minimum. La figure suivante représente l'hyper surface de l'erreur possédant un minimum globale, en fonction de deux paramètres.

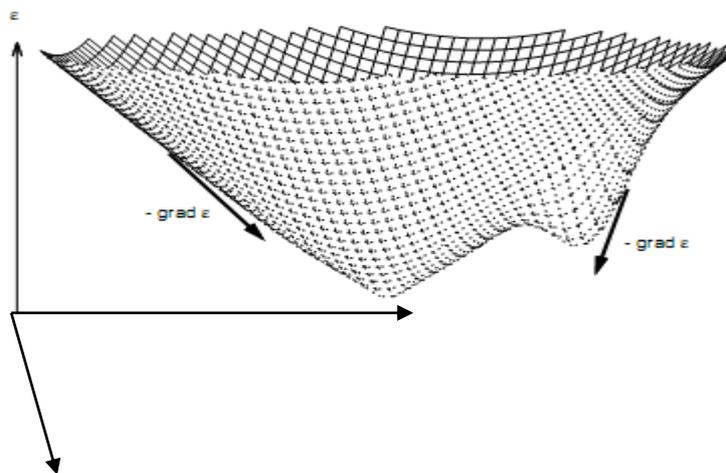


Figure (4.12) : l'hyper surface de l'erreur possédant un minimum globale, en fonction de deux paramètres.

Dans la littérature, il existe un grand nombre de méthodes d'optimisations itératives, parmi lesquelles on trouve la descente du gradient, c'est l'un des algorithmes les plus largement utilisés, car il ne fait intervenir que la première dérivée, mais il se caractérise par la lenteur et n'assure pas la convergence vers le minimum global, lorsque les locaux sont présents. La méthode de Newton assure la convergence vers l'optimum en une seule itération, pour les critères ayant une forme quadratique, mais nécessite beaucoup de calculs et d'espace mémoire à cause de la manipulation de la jacobienne du critère.

L'autre méthode d'optimisation, qui a l'avantage de convergence en nombre fini d'itérations vers l'optimum global, pour des fonctions quadratique (avantage de Newton) mais ne nécessite pas le

calcul de la deuxième dérivée (avantage de la méthode de gradient), est l'algorithme du gradient conjugué.

Soit la fonction $F(x)$ à minimiser, suivant x

La valeur de x à chaque itération $k+1$ est égale :

$$x_{k+1} = x_k + \Delta x_k \quad (4.1, a)$$

$$\Delta x_k = \alpha_k * p_k \quad (4.1, b)$$

p_k est la direction de recherche du minimum.

$0 < \alpha_k < 1$ le gain d'adaptation du paramètre x à l'itération $k+1$.

Δx_k : est le terme de la mise à jour de la variable x à l'itération $k+1$.

IV.8.4.1 Optimisation par descente du gradient :

Soit une fonction $F(x)$ qu'on veut minimiser par rapport à x , pour cela il faut assurer qu'à chaque itération $k+1$ on :

$$F(x_{k+1}) < F(x_k) \quad (4.2)$$

Le développement en série de Taylor du premier ordre de F évalué autour de x_k :

$$F(x_{k+1}) = F(x_k + \Delta x_k) = F(x_k) + g_k * \Delta x_k \quad (4.3)$$

$$g_k \triangleq \nabla F(x)|_{x=x_k}$$

g_k : est le gradient de F évalué à x_k .

Pour vérifier la condition (4.2) il faut que :

$$g_k * \Delta x_k = g_k \alpha_k * p_k < 0 \Rightarrow g_k * p_k < 0$$

Notre but, est de déterminer, à chaque itération la direction p_k qui fait que le terme $g_k * p_k$ soit le plus négatif pour une descente plus raide. Etant donné que nous avons un produit entre un gradient et une direction, le résultat sera le plus négatif lorsque la direction et le gradient sont colinéaire de sens opposé :

$$-g_k = p_k \quad (4.4)$$

La valeur du paramètre à l'instant $k+1$ devient :

$$x_{k+1} = x_k - g_k \alpha_k \quad (4.6)$$

IV.8.4.1.1 Le choix du gain d'adaptation :

Le choix du bon gain d'adaptation doit se faire pour réaliser le compromis entre temps de convergence et stabilité de l'algorithme d'optimisation.

a) Gain d'adaptation fixe :

C'est la solution la plus simple, mais pas la meilleure, surtout lorsqu'on a des hyper surfaces d'erreur (non régulier), où l'algorithme d'optimisation nécessite différentes valeurs de α_k selon la variation du gradient.

Un choix du gain important accélère la convergence, mais dans la majorité des cas il déstabilise l'algorithme d'optimisation, et l'optimum ne sera jamais atteint.

Un choix du gain faible ralentit l'algorithme, et favorise la convergence vers des optimums locaux.

b) Gain d'adaptation adaptatif :

Le choix du gain d'adaptation va se faire à chaque itération, pour satisfaire la relation suivante :

$$\frac{dF(x_{k+1})}{d\alpha_k} \text{ Soit minimale.} \quad (4.7)$$

$$\text{Ce qui donne :} \quad \alpha_k = -\nabla F^T(x)|_{x=x_k} * \frac{p_k}{p_k^T * \nabla^2 F(x)|_{x=x_k} * p_k} = -\frac{g_k^T p_k}{p_k^T A_k p_k} \quad (4.8)$$

$\nabla^2 F(x)|_{x=x_k} = A_k$ La jacobienne de $F(x)$.

Cette méthode est peu pratique à cause de la manipulation de A à chaque itération.

IV.9 L'apprentissage du réseau de neurones :

L'apprentissage des réseaux de neurones est la détermination des paramètres internes optimales du réseau afin d'adapter le traitement effectué par le réseau à une tâche bien déterminée. Man et Halang (1997).

IV.9.1 Algorithme de rétro-propagation du gradient :

Établi par David Rumhart et James McClelland (1986).

Le perceptron multicouche entraîné par Backpropagation est le réseau de neurones le plus largement utilisé. C'est le résultat de la généralisation de l'algorithme (least mean square) établi par Widrow et Ted (1960), pour l'entraînement du perceptron monocouche.

Le perceptron multicouche :

C'est un réseau feedforward, contenant des réseaux perceptron en cascade. Il a mis fin aux limitations des perceptrons monocouche qui ne peuvent résoudre que des problèmes à variables linéairement séparables.

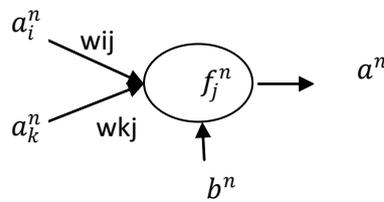


Figure (4.13) : neurone j de la couche cachée n d'un réseau multicouche.

a_i^{n-1} : la sortie du neurone i de la couche n-1.

W_{ij} : le poids synaptique entre neurone i de la couche n-1 et neurone j de la couche n.

f_j^n : La fonction d'activation des neurones de la couche n.

La sortie du réseau à trois couches est la suivante:

$$a^3 = f^3(W^3(f^2(W^2(f^1(W^1 * R + b^1) + b^2) + b^3)) \quad (4.11)$$

W^n : représentation matricielle des poids synaptiques de la couche n.

a^n : la représentation vectoriel des sorties de la couches n.

On constate que la sortie dépend implicitement des paramètres internes du réseau, Et que la sortie de chaque couche dépend de la sortie de la couche précédente. On peut dire, de manière plus générale, que :

$$a^{m+1} = f^{m+1}(n^{m+1}) \quad m=0 \dots M-1$$

$$n^{m+1} = W^{m+1} * a^m + b^{m+1}$$

M c'est le nombre de couches du réseau.

a^{m+1} C'est le vecteur de sortie de la couche m+1.

a^m C'est le vecteur sortie de la couche m, et celui d'entrée de la couche m+1.

$a^0 = R$ Vecteur d'entrée du réseau.

$y = a^M$ Vecteur de sortie du réseau.

Le critère de performance de Backpropagation est l'approximation de la moyenne des carrés des erreurs, résultants de la comparaison de la sortie du réseau (y) par rapport à celle désirée (y_d), par le ½ carré des erreurs.

Ce cas d'algorithme utilise la descente du gradient avec un gain d'apprentissage fixe comme moyen d'optimisation de ce critère.

La fonction à minimiser est :

$$E = e^2 = 0.5 * (y - yd)^2 \quad (4.12)$$

Les valeurs des paramètres à l'itération k+1 en fonction de celui à l'itération k:

$$w_{ij}^m(k+1) = w_{ij}^m(k) - \alpha \frac{\partial E}{\partial w_{ij}^m} \quad (4.13, a)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial E}{\partial b_i^m} \quad (4.13, b)$$

α : C'est le gain d'adaptation paramétrique, ou appelé taux d'apprentissage.

i: indice du neurone i de la couche m.

j : indice du neurone j de la couche m-1.

w_{ij}^m : Le poids de la connexion entre le neurone i de la couche m et le j ème neurone de la couche m-1.

b_i^m Le biais du neurone i de la couche m.

On a vu précédemment que la sortie du neurone y n'est pas fonction explicite des w et b, alors E n'est pas aussi fonction explicite de w et b.

Pour calculer le gradient de E, on doit utiliser le concept suivant :

Soit f une fonction explicite de x, la dérivée de f par rapport à y s'obtient comme suit :

$$\frac{df(x)}{dy} = \frac{df(x)}{dx} * \frac{dx(y)}{dy} \quad (4.14)$$

Et comme le potentiel d'activation (n_i^m) du neurone i de la couche m dépend explicitement de w_{ij}^m (les poids synaptique de la couches m) on a :

$$\frac{\partial E}{\partial w_{ij}^m} = \frac{\partial E}{\partial n_i^m} * \frac{\partial n_i^m}{\partial w_{ij}^m} \quad (4.15,a)$$

$$\frac{\partial E}{\partial b_i^m} = \frac{\partial E}{\partial n_i^m} * \frac{\partial n_i^m}{\partial b_i^m} \quad (4.15,b)$$

avec :

$$n_i^m = \sum_{j=1}^{s^{m-1}} W_{ij}^m * a_j^{m-1} + b_i^m \quad i=1 \dots s^m \quad (4.16)$$

s^{m-1} Nombre de neurones dans la couche m-1.

La dérivée de (4.16) par rapport aux paramètres donnés :

$$\frac{\partial n_i^m}{\partial w_{ij}^m} = a_j^{m-1} \quad (4.17.a)$$

$$\frac{\partial n_i^m}{\partial b_i^m} = 1 \quad (4.17.b)$$

Posons $\mathbf{S}_i^m = \frac{\partial E}{\partial n_i^m}$ la fonction de sensibilité de l'erreur aux variations du potentiel d'activation du neurone i de la couche m .

(4.15,a) et (4.15,b) deviennent :

$$\frac{\partial E}{\partial w_{ij}^m} = \mathbf{S}_i^m * a_j^{m-1} \quad (4.18.a)$$

$$\frac{\partial E}{\partial b_i^m} = \mathbf{S}_i^m \quad (4.18 .b)$$

Le calcul de \mathbf{S}^m : le vecteur sensibilité de la couche m :

La variation du potentiel d'activation de la couche $m+1$ par rapport à celui de m est donnée par :

$$\frac{\partial n^{m+1}}{\partial n^m} = \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \dots & \frac{\partial n_1^{m+1}}{\partial n_{s^m}^m} \\ \vdots & \ddots & \vdots \\ \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_1^m} & \dots & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_{s^m}^m} \end{bmatrix}$$

$$\frac{\partial n_i^{m+1}}{\partial n_j^m} = \frac{\partial (\sum_{l=1}^{s^{m+1}} W_{il}^{m+1} * a_l^m + b_i^{m+1})}{\partial n_j^m} = W_{ij}^{m+1} * \frac{\partial a_j^m}{\partial n_j^m}$$

Posons :

$$\frac{\partial a_j^m}{\partial n_j^m} = \frac{\partial f_j^m(n_j^m)}{\partial n_j^m} = \dot{f}^m(n_j^m)$$

On obtient :

$$\frac{\partial n^{m+1}}{\partial n^m} = W^{m+1} * \dot{F}^m(n^m) \quad (4.19)$$

$$\dot{F}^m(n^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \dot{f}^m(n_{s^m}^m) \end{bmatrix}$$

W^{m+1} Représentation matricielle des poids synaptiques à l'entrée de la couche $m+1$.

Alors la sensibilité \mathbf{S}^m écrit par la formule de récurrence suivante :

$$\mathbf{S}^m = \frac{\partial E}{\partial n^m} = \dot{F}^m(n^m) * W^{m+1T} * \mathbf{S}^{m+1} \quad (4.20.a)$$

$$m = M-1 \dots 1$$

la sensibilité de la couche de sortie est:

$$\mathbf{S}^M = \frac{\partial E}{\partial n^M} = \frac{\partial E}{\partial y} * \frac{\partial y}{\partial n^M} = 2 * (y - yd) * \dot{F}^M(n^M) \quad (4.20.b)$$

le terme d'adaptation par rétro propagation est :

$$\frac{\partial E}{\partial w_{ij}^m} = \mathbf{S}^m * \frac{\partial n_i^m}{\partial w_{ij}^m} \quad (4.21,a)$$

$$\frac{\partial E}{\partial b_i^m} = \mathbf{S}^m * \frac{\partial n_i^m}{\partial b_i^m} \quad (4.21,b)$$

On remarque que la sensibilité se propage dans le sens inverse de l'information du réseau d'où le nom Backpropagation.

IV.9.2 Dimension de la base d'apprentissage :

Elle dépend de la complexité du problème ou du nombre de points d'inflexion que contient la fonction à représenter par le réseau de neurones. Cette tâche est généralement laissée à l'intuition du concepteur et son expérience.

Widrow et Hoff ont proposé une méthode empirique pour déterminer cette dimension en se basant sur l'erreur tolérée et le nombre de paramètres :

$$N = \dim(W)/\varepsilon ;$$

N exemples d'apprentissage, W paramètres libres, ε erreur

Si on accepte une erreur de 10%, on doit présenter 10 fois plus d'exemples que le nombre de paramètres libres.

IV.9.3 Critère d'arrêt de l'apprentissage :

Il faut arrêter la phase d'apprentissage avant que le réseau n'entre dans la phase de spécialisation, pour qu'il puisse garder sa caractéristique importante de généralisation, la figure suivante montre le compromis entre minimisation d'erreur et généralisation :

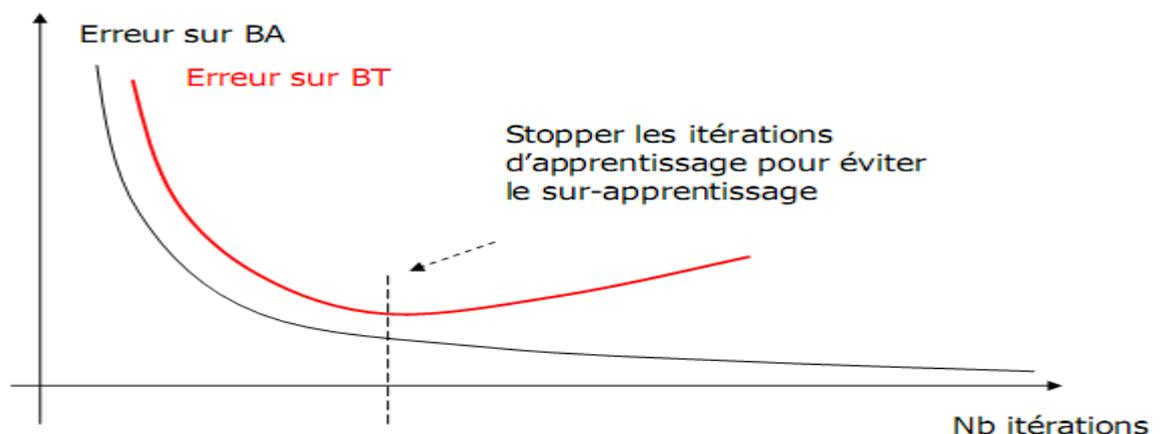


Figure (4.16) : Courbe typiques d'apprentissage et de test

BA : base d'apprentissage.

IV.10 Les inconvénients des réseaux de neurones :

A l'exception de quelques algorithmes aidant le constructeur, soit en pas à pas à construire les couches cachées, soit à éliminer les liaisons inutiles dans le cas d'un réseau surdimensionné (Refenes,1995) , il n'existe pas de méthode systématique pour l'établissement de l'architecture du réseau, du nombre de couches cachées, et du nombre de neurones contenus dans chacune de ces dernières.

Il n'y a pas de solution analytique pour le choix des poids et des biais, et l'utilisation d'algorithmes itératifs, descente du gradient, peut entraîner :

- Convergence vers un minimum local de la fonction coût.
- Une phase d'apprentissage longue.
- Différentes solutions obtenues à partir des différentes conditions initiales.

L'aspect boîte noire du réseau de neurones ne permet pas d'extraire les relations entre les variables du réseau ni d'interpréter ou de retracer le cheminement du résultat.

Difficulté dans la sélection de la base d'apprentissage pour qu'elle soit la plus représentative.

L'appel à l'intuition du constructeur, en ce qui concerne le choix initial des paramètres d'apprentissage, le gain d'adaptation et le point d'arrêt de la phase d'apprentissage.

Chapitre V
La logique floue

V.1 Introduction:

La principale caractéristique de la logique classique introduite par George Cantor (1845-1918), est l'appartenance absolue ou l'exclusion absolue d'un élément de l'univers de discours à un sous ensemble de cet univers. C'est-à-dire l'élément possède la caractéristique à 100% ou à 0%. Dans cette logique les bornes des sous ensembles sont claires et bien définies. Figure (5.1.a).

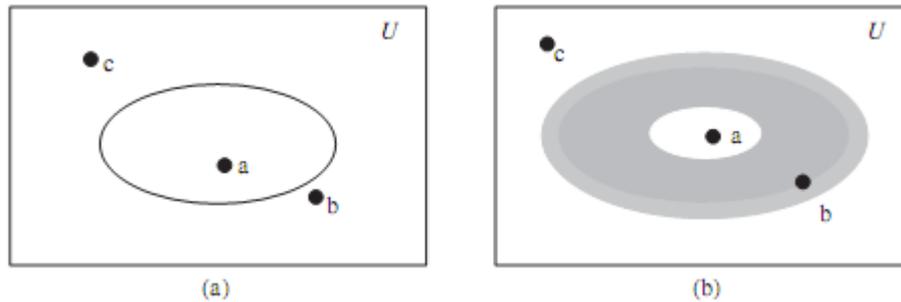


Figure (5.1) (a) sous ensemble classique. (b) sous ensemble flou.

Cette façon de raisonner trouve rapidement ses limites, lorsqu'elle est face à des problèmes physiques incertains, dont la majorité des descriptions de leur fonctionnement est basée sur le raisonnement humain imprécis ou possédant des variables dont la mesure est impossible ou qui ne peut être précise. Ce type de problèmes ne pouvait être résolu par des ordinateurs, jusqu'à l'arrivée de la logique floue, qui permet de traiter des problèmes dont les bornes définissant les caractéristiques de leurs éléments ne sont pas claires et/ou flexibles. Figure (5.1.b).

La logique floue a permis de généraliser la propriété d'appartenance à un sous ensemble donné, par l'introduction des fonctions d'appartenance, qui fournit le degré de vérité avec lequel un élément possède la caractéristique, définie par cette fonction d'appartenance. Ce degré varie de 0 à 1, lorsqu'il est égale à une de ces limites, l'appartenance est équivalente à celle de la logique de Cantor.

Dans le présent chapitre, nous allons présenter les applications de la logique floue, ces principaux constituants, la façon de raisonnement d'un système flou, quelques types de régulateurs flous, et nous terminons par la présentation des principaux avantages et inconvénients de cette logique.

V.2 Historique de la logique floue :

Avant le 19^{ème} siècle, la science considérait que les imprécisions sont des états indésirables, qu'il faut omettre à tout prix, mais lorsque les physiciens se sont rendu compte que les lois de la physique classique, newtonienne, n'étaient pas applicables à l'échelle atomique, il a fallu développer des méthodes qui permettent de manipuler des entités microscopiques dont le comportement et les caractéristiques étaient presque inconnus (électrons, atome...etc.). Ceci a conduit au développement de nouvelles approches qui sont basées principalement sur les lois de probabilités (Klir et Yuan, 1995).

Le 20^{ème} siècle a vu la naissance de plusieurs alternatives à la théorie des probabilités. Jan Lukasiewicz a développé multivalued discrete logic. Dans les années 60 Artur Dempster, (1967) a développé la théorie de l'évidence, qui permettait et pour la première fois d'évaluer l'ignorance et l'absence de l'information.

En 1965, Lotfi Zadeh a introduit une nouvelle logique, qui permet de répondre à tout type d'imprécisions, associé avec des informations linguistiques et intuitives.

La logique floue s'est imposée dans les différents domaines et applications, vu sa puissance de tirer la bonne information d'un certain nombre de descriptions humaines linguistiques. Dans le domaine du contrôle des processus, Mamdani, en 1970, a construit le premier contrôleur flou, qui a donné une poussée au développement et à l'utilisation de la logique floue, des années plus tard Takagi et Sugeno proposèrent un autre type de contrôleur flou. Les recherches et l'application de la logique floue se poursuivent jusqu'à présent, et les chercheurs durant cette dernière décennie se sont intéressés à hybrider cette logique avec d'autres approches, dont le neurone flou (Kosk, 1992)...etc.

V.3 Les applications de la logique floue :

Depuis l'introduction du concept du raisonnement flou, ce dernier s'est imposé dans divers domaines d'applications. Parmi ces domaines en trouve : la biologie, la médecine, l'économie, l'écologie, la technique, la défense, ...etc.

Voici quelques domaines techniques utilisant cette logique :

- La commande de processus, l'identification des processus, la robotique, la programmation, l'optimisation, les décisions multicritères, la classification,...etc.
- Beaucoup d'applications en électronique de puissance. (Song, 1998).
- La modélisation des systèmes nonlinéaires, complexes, incertains. (Wang, 1996).
- Dans les différentes applications industrielles (l'électroménager au Japon...etc).

V.4 Systèmes flous :

La structure standard d'un système flou est donnée par la figure (5.2). Les entrées-sorties de n'importe quel système flou sont des nombres réelles.

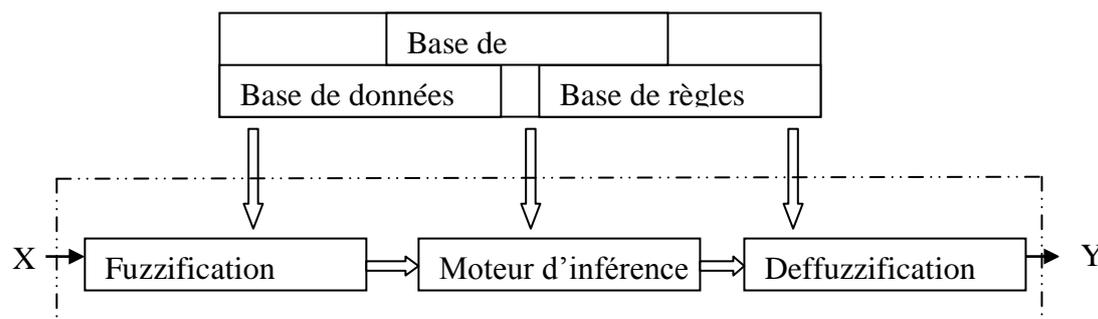


Figure (5.2) : structure standards d'un système flou.

Le bloc de fuzzification transforme les données réelles en données floues.

Le mécanisme d'inférence utilise la base des règles floues pour produire les conclusions.

Le bloc de defuzzification transforme ces conclusions floues en sorties numériques réelles.

V.5 L'univers de discours :

Exemple : la position x peut prendre les valeurs entre $-4m$ à $+5m$, $[-4,5]$ est l'univers de discours de x .

V.6 Les variables linguistiques:

L'établissement de la base de règles se fait par l'utilisation des expressions linguistiques, nécessitant l'expression des variables d'entrées/sorties par des variables linguistiques.

Ce sont des descriptions symboliques constantes décrivant les entrées/sortie des systèmes flous qui sont variant dans le temps.

Cette variable est définie par :

- son nom.
- l'ensemble des valeurs qu'elle prend, appelées termes linguistiques.
- son univers de discours.

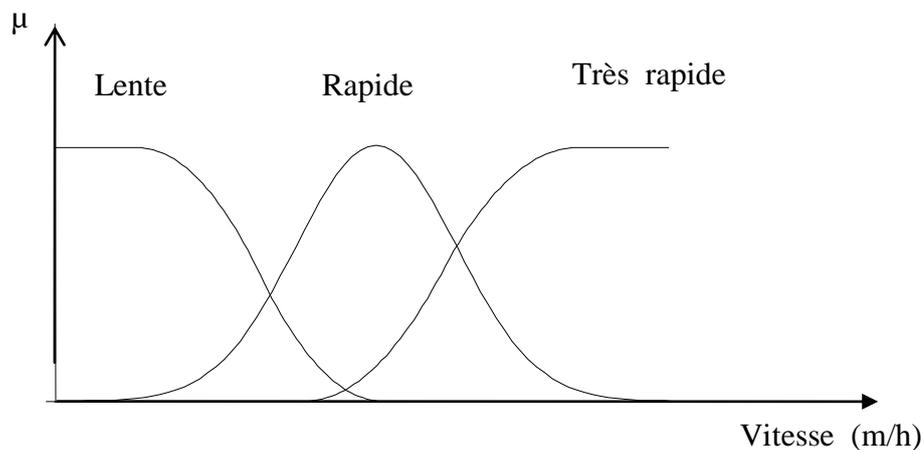


Figure (5.3) : représentation de la variable floue vitesse

Le nom de la variable : vitesse

Univers de discours $[0, \infty[$

Les valeurs linguistiques de cette variable = {lente, Rapide très rapide}

V.7 Les valeurs linguistiques :

Comme les variables réelles prennent leurs valeurs de leurs domaines de définition (univers de discours), les variables linguistiques puisent leurs valeurs linguistiques de leur univers de discours contenant les termes de description de ces variables.

Exemple: pour la variable linguistique du débit, son univers de discours peut être {faible, fort, très fort}.

V.8 Les règles linguistiques :

La liaison entre les entrées/sorties d'un système flou établi par l'intermédiaire de base de règles floues, où chaque règle se présente sous la forme suivante :

Condition (prémisse) → conclusion.

C'est l'établissement de ces règles qui va indiquer au système flou son comportement.

V.9 L'ensemble flou (sous-ensemble flou):

Utilisé pour quantifier et donner un sens numérique aux variables linguistiques. Ce concept est introduit par la définition de fonction d'appartenance et l'intervalle des valeurs de la variable réelle pour lesquelles le degré de vérité n'est pas nul.

Notation : soit E un univers de discours de la variable x, A est dit univers de discours

$$A = \{(x, \mu_A(x)), x \in E\}$$

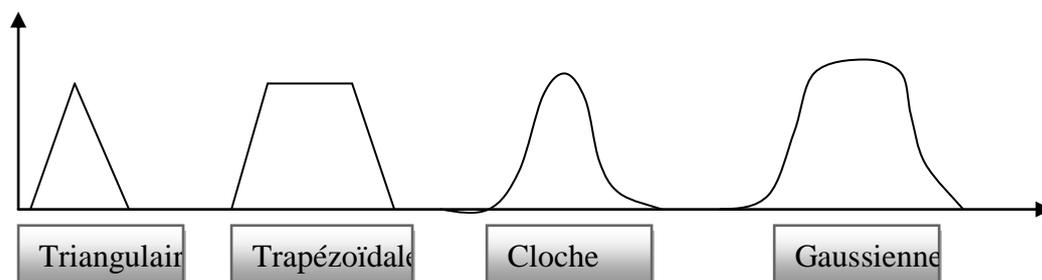
V.10 Les fonctions d'appartenances :

C'est la représentation analytique et continue (ou discrète) de chaque terme linguistique de la variable linguistique. Cette fonction peut être continue si l'univers de discours est continu ou discrète si ce dernier est discret.

Elle donne le degré de vérité qu'une variable xi vérifie la caractéristique Li, où Li est une valeur l'linguistique de l'univers de discours de xi.

Les fonctions d'appartenances permettent le traitement mathématique des variables linguistiques, dans le but de traiter les déductions floues par ordinateur.

V.11 Les formes des fonctions d'appartenances :



Les caractéristiques des fonctions d'appartenance sont leurs centres et leurs largeurs sur l'univers de discours.

V.12 Les opérateurs de la logique floue :

Le traitement des variables floues s'effectue par ce qu'on appelle les opérateurs flous, ces derniers agissent sur les fonctions d'appartenance des variables linguistiques. Voici les principaux opérateurs :

L'inclusion : Soit A et B deux sous ensembles flous, et x un élément de l'univers de discours E.

A est dit inclus dans B si $\mu_A(x) < \mu_B(x)$ pour tout $x \in E$.

Le complément :

$$\bar{\mu}_A(x) = 1 - \mu_A(x)$$

Intersection : il existe plusieurs méthodes de calcul de l'intersection.

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$$

$$\mu_{A \cap B}(x) = \mu_A(x) * \mu_B(x)$$

$$\mu_{A \cap B}(x) = \gamma \min(\mu_A(x), \mu_B(x)) + \frac{1-\gamma}{2} * (\mu_A(x) + \mu_B(x)) \quad 0 < \gamma < 1$$

Disjonction : il existe plusieurs méthodes de calcul de la disjonction.

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$$

$$\mu_{A \cup B}(x) = \frac{\mu_A(x) + \mu_B(x)}{2}$$

$$\mu_{A \cup B}(x) = \gamma \max(\mu_A(x), \mu_B(x)) + \frac{1-\gamma}{2} * (\mu_A(x) + \mu_B(x)) \quad 0 < \gamma < 1$$

Min-max : réalise le passage progressif entre ET et le OU flous selon la valeur de γ

$$\mu_{\min-max}(x) = \gamma \min(\mu_A(x), \mu_B(x)) + (1 - \gamma) * \max(\mu_A(x), \mu_B(x)) \quad 0 < \gamma < 1$$

Produit cartésien : les opérateurs utilisés jusqu'à maintenant s'appliquent sur des sous ensembles flous appartenant au même univers de discours, par contre celui là s'applique sur des ensembles flous appartenant à des univers de discours différents.

C'est le produit utilisé dans la partie prémisse des règles floues dans le bloc d'inférence.

V.13 Le raisonnement d'un système flou :**V.13.1 La fuzzification :**

C'est la transformation des variables numériques (les entrées du contrôleur), en variable linguistique (floue).

Pour chaque variable d'entrée, il fournit, sur sa sortie, les degrés d'appartenances de cette variable à chacune des fonctions d'appartenances qui lui sont associées.

V.13.2 L'inférence :

Ce bloc réalise la liaison entre les variables floues, par l'intermédiaire de la base des règles linguistiques, établies par le constructeur. Et donne en sortie la valeur floue (ensemble flou) de la commande Y.

La matrice d'inférence est la représentation graphique du moteur d'inférence. Table (5.1). La lecture à partir de cette représentation se fait comme suit :

Si (X1 est N ET X2 est N) alors (Y est Z)

Y :

X1\X2	N	Z
N	Z	
Z		N

Table (5.1) : la matrice d'inférence des variables floues

Les méthodes d'inférences :

Max-min

Max-prod

Somme-prod

V.13.3 La defuzzification :

C'est l'opération inverse de la fuzzification, elle transforme l'ensemble flou de la sortie (la commande) en variable numérique.

Etant donné que les systèmes physiques ne peuvent accepter que des variables réelles numériques, le passage de la représentation floue de la variable de sortie vers une représentation numérique est nécessaire.

Méthodes de defuzzification :**Defuzzification par centre de gravité :**

C'est la méthode de defuzzification la plus utilisée, la valeur résultante de cette méthode est obtenue par le calcul du centre de gravité de l'ensemble flou issu de l'étape d'inférence des variables floues.

Defuzzification par valeur maximum :

Elle est plus simple que la précédente. Le résultat de defuzzification est la valeur de la sortie correspondante à un degré d'appartenance maximal à l'ensemble flou issu du bloc d'inférence.

Defuzzification par la moyenne des maximas :

La sortie est égale à la moyenne des valeurs, correspondants à un degré d'appartenance maximal à l'ensemble flou issu du bloc d'inférence.

V.14 La commande par la logique floue :

L'idée d'introduire les techniques floues dans le domaine de la commande a été proposée pour la première fois par Chang et Zadeh, (1972).

V.14.1 Types de contrôleurs flous :

La différence entre les différents types de régulateurs flous se manifeste dans le mécanisme d'inférence utilisé, et plus précisément dans la forme des conclusions des règles floues. On en distingue plusieurs types, les plus connus sont le contrôleur de Mamdani et celui de Sugeno.

V.14.1.1 Le contrôleur de Mamdani :

Ce contrôleur a la structure standard des systèmes flous, caractérisé par des règles d'inférences ayant des prémisses et des conclusions manipulant des variables floues. Dans ce régulateur la sortie, ou le signal de commande, est une variable floue qui doit subir une defuzzification. Voici quelques exemples de régulateurs floue de type Mamdani.

a) Le régulateur proportionnel flou de type Mamdani :

Les sous ensembles flous associés à l'entrée et la sortie du régulateur Proportionnel, le contrôleur est ajusté par le changement des valeurs de A et/ou de B.

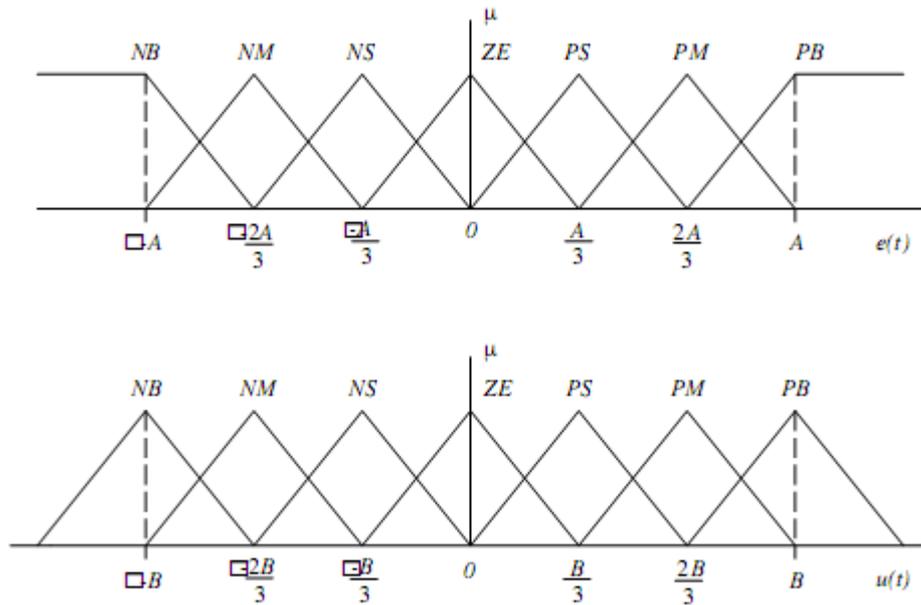


Figure (5.4) : fonction d'appartenances pour $e(t)$ et $u(t)$, (Jenkins ,1996)

Les règles floues d'un contrôleur Proportionnel d'un système SISO, sont construites de façon symétrique. Voici la base des règles de ce contrôleur (Passino, 1998)

1. **If e is NB Then u is NB**
2. **If e is NM Then u is NM**
3. **If e is NS Then u is NS**
4. **If e is ZE Then u is ZE**
5. **If e is PS Then u is PS**
6. **If e is PM Then u is PM**
7. **If e is PB Then u is PB**

Le mécanisme de fuzzification est singleton et la méthode du centre de gravité est utilisée pour la defuzzification.

On peut modifier cette caractéristique nonlinéaire et son effet sur le système en modifiant A et B.

Ce régulateur possède une nonlinéarité statique entre l'entrée et la sortie figure (5.5), cette dernière dépend de la base des règles floues, de la stratégie de fuzzification et de defuzzification utilisées.

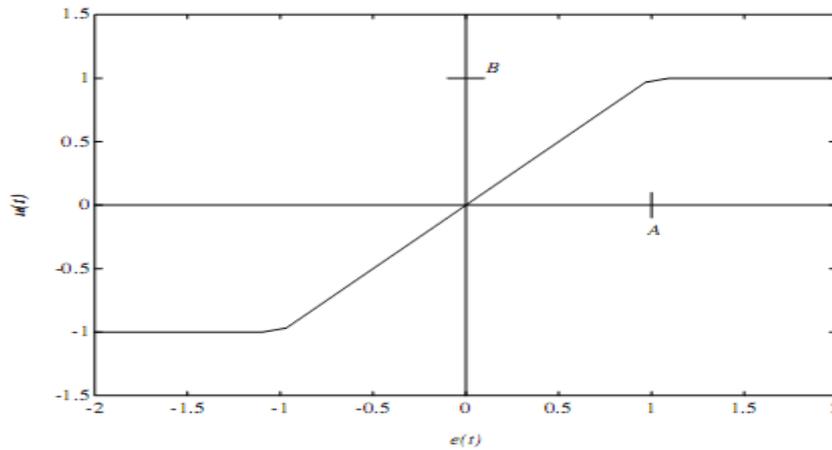


Figure (5.5) : la non linéarité du régulateur flou PD T-S (Jenkins ,1996).

b) Contrôleur PD flou de type Mamdani:

Il ressemble au contrôleur précédent avec une entrée en plus, et les mêmes sous-ensembles flous que la figure (5.5) avec ajustement des paramètres A et B, le paramètre d’ajustement de la deuxième variable d’entrée est D, utilisant le produit et centre de gravité comme outil de fuzzification et de defuzzification sur la base des règles présentée dans la table suivante :

"output" <i>u</i>		"change-in-error" <i>e</i>						
		NB	NM	NS	ZE	PS	PM	PB
"error" <i>e</i>	NB	NB	NS	PS	PB	PB	PB	PB
	NM	NB	NM	ZE	PM	PM	PB	PB
	NS	NB	NM	NS	PS	PM	PB	PB
	ZE	NB	NM	NS	ZE	PS	PM	PB
	PS	NB	NB	NM	NS	PS	PM	PB
	PM	NB	NB	NM	NM	ZE	PM	PB
	PB	NB	NB	NB	NB	NS	PS	PB

Tableau (5.2) : table de régulateur PD T-S, tirée de (Passino, 1998)

V.14.1.2 Le contrôleur de Sugeno

Proposé par Takagi et Sugeno en (1985), ce régulateur est un cas particulier de la classe des systèmes flous fonctionnels.

Ce type de système se différencie des systèmes flous standards par la forme de ses conclusions. Au lieu qu’il ait des conclusions floues, ses conclusions sont des fonctions dont les arguments sont généralement les variables d’entrées du système

La partie prémisse de ses règles est de même forme que celle du contrôleur de Mamdani.

La forme des règles du régulateur de Sugeno :

Soit X_i l'entrée du contrôleur flou, et P_i sont les fonctions d'appartenances associées à chaque variable d'entrée.

Y est la sortie du contrôleur.

Si (X_1 est P_1 ET... ET X_n est P_n) **alors** $Y_i = F_i(X)$

Selon $F(X)$, on peut distinguer plusieurs variantes du contrôleur de Sugeno :

$F(x)$, constante on parle de contrôleur de Sugeno d'ordre zéro.

$F(x)$ est un polynôme en fonction des X_i , le contrôleur de Sugeno est dit d'ordre un.

La defuzzification du contrôleur de Sugeno :

La sortie du contrôleur est calculée par la formule suivante :

$$Y = \sum \mu_i * \frac{F_i(X)}{\sum \mu_i}$$

μ_i Degré d'activation de la règle i ayant pour sortie $Y_i = F_i(X)$

On peut dire que le contrôleur de Sugeno d'ordre zéro est équivalent à un système flou standard utilisant la defuzzification par centre de gravité avec des fonctions d'appartenances des variables de sorties de type singleton.

V.14.2 Les avantages de la commande floue :

- Le contrôleur flou ne dépend pas du modèle mathématique du système à commander, ce qui facilite la commande des systèmes complexes et non linéaires, et a favorisé l'émergence de ce type de contrôleur.
- Kosko (1994) a montré que les systèmes flous sont des approximations universelles.
- Convient pour le contrôle des systèmes incertains, vu sa robustesse en vers les erreurs de modélisation, et aux perturbations.
- La possibilité d'implémenter des connaissances de l'opérateur de processus.

V.14.3 Les inconvénients de la commande floue :

- Manque de directives précises pour la conception du régulateur.
- Cohérences des inférences non garanties.
- Absence de méthodes standards pour la transformation de la connaissance ou de l'expérience vers la base des règles linguistiques d'un système flou.
- L'impossibilité de la démonstration de la stabilité en toute généralité.

Chapitre VI
Contrôle neuro-flou du
bioréacteur de fermentation

VI.1 Introduction :

Nous avons abordé, dans les chapitres précédents, deux approches différentes d'intelligence artificielles, permettant le traitement et la résolution des systèmes complexes, nonlinéaires. Chacune d'elles a été développée, au départ, pour réaliser certains objectifs, et destinée à résoudre des problèmes de natures différentes. La logique floue, créée par Zadeh, avait pour but de résoudre les lacunes de la logique classique de Cantor, et de représenter les systèmes dont les frontières, entre les différentes caractéristiques que peuvent avoir ses éléments, sont floues.

Chacune des deux approches citées ci-dessus a ses propres avantages et ses propres inconvénients. Mais ce qui est bien à remarquer, c'est que les inconvénients de l'une sont les avantages de l'autre et vice-versa, on peut dire qu'elles se complètent. Au moment où les réseaux de neurone ne permettent pas une utilisation de l'expert linguistique, donc une interprétation du fonctionnement interne du réseau et la détermination de l'enchaînement du résultat est impossible, la logique floue trouve sa puissance dans sa capacité descriptive élevée, vue qu'elle manipule des variables linguistiques et que son raisonnement épuise ses directives des descriptions linguistiques données par l'expert, traduites sous forme de règles linguistiques.

Le point faible de la logique floue est l'absence de méthode ou d'algorithme permettant le bon choix des paramètres du système flou, et leur détermination se fait, en général, par tâtonnement, le temps de leur détermination est fonction de l'expérience, de l'intuition, et du nombre de paramètres à déterminer. Par contre, les réseaux de neurones ne souffrent pas de ce problème, grâce à leur capacité d'apprentissage, ils peuvent non seulement déterminer les paramètres qui conviennent, mais de les optimiser, pour réaliser la fonction pour laquelle ils ont été développés.

Vue la complémentarité entre ces deux approches, les chercheurs se sont intéressés à les combiner, pour aboutir à un système facile à manier et à interpréter, utilisant les expressions linguistiques proche du langage humain et dont les paramètres internes seront déterminées automatiquement par un algorithme d'apprentissage. Alors ils ont abouti à ce qu'on appelle les systèmes neuro-flous.

VI.2 Les systèmes neuro-flous :

C'est le résultat de la combinaison de deux approches d'intelligence artificielle, pour la construction de systèmes flexibles et robustes.

Gupta et al., (1991), ont proposé trois modèles de neurones flous, ainsi que les idées principales sur les mécanismes d'apprentissage et d'adaptation, mais sans aucun détail permettant l'implémentation, ils affirment que les neurones flous sont un outil très puissant pour la modélisation des phénomènes associés avec la réflexion humaine et la perception.

La plus part des modèles neuro-flous proposés sont des réseaux de neurones, dont la fonctionnalité est équivalente à celles des contrôleurs flous, et utilisent l'algorithme d'apprentissage du Backpropagation pour leurs entraînements. Jang(1995,1998), Berenji,

(1992,1993) et Horikawa,(1992) ont proposé des modèles généralisés de réseaux de neurones qui peuvent répondre à une large gamme de systèmes flous.

Ils ont une capacité descriptive élevée qui résulte de l'utilisation des connaissances d'expert et de la manipulation de variables linguistiques.

VI.3 Structure de base du neurone flou :

La structure générale d'un neurone flou est très proche du modèle du neurone classique, à différence que celui-là manipule des variables floues caractérisées par leurs valeurs linguistiques, alors que l'autre manipule des variables réelles.

VI.4 Les différentes architectures du système neuro-flou :

Wang et Lee, (2001), ont classé les systèmes neuro-flous hybrides en trois groupes principaux, et ceci en fonction de la forme des règles floues utilisées :

- **Groupe 1** : les Conclusions des règles sont de type Mamdani. La figure (6.1.a) représente le système Falcon et la figure (6.1.b) représente le système Garic.

Ces deux systèmes ont été développés par Lin et Lee, (1991). Ils ont une structure composée de 5 couches, et des poids synaptiques égaux à l'unité. Chaque couche de ce réseau est la projection d'une tache réalisée par le système flou de Mamdani.

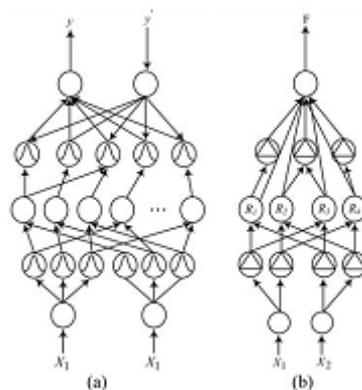


Figure (6.1) réseaux neuro flou du groupe 1

Couche 1 : réalise la fuzzification de X_1 et X_2 en les connectant avec leurs termes linguistiques (neurone de la couche 2).

Couche 2 : chaque neurone de cette couche calcule le degré d'appartenance de la variable linguistique présente sur son entrée.

Couche 3 : chaque neurone représente la prémisse d'une règle, donc le nombre de neurones est égale au nombre de règles. Utilise les T-normes pour calculer le degré d'activation de chaque règle.

Couche 4 : détermine les ensembles flous de la sortie y' pour chaque conclusion.

Couche 5 : c'est la couche de defuzzification par le moyen des Co-normes.

- **Groupe 2** : les signaux flous sont combinés avec des poids flous, les conclusions sont de type fonction linéaire des variables d'entrées.

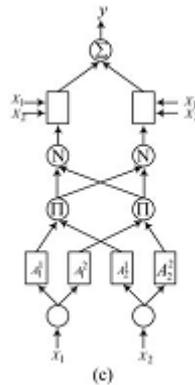


Figure (6.2) réseau neuro-flou du groupe 2(ANFIS)

- **Groupe 3** : NEFCON, réseau de neurones décrit par des équations logiques (si/alors) les conclusions sont de type singleton.

Caractérisé par une rapidité plus importante que celle des groupes I et II, vu le nombre de couches inférieur qu'il contient, et l'absence des couches de defuzzification. Il est plus adapté pour les problèmes de classification.

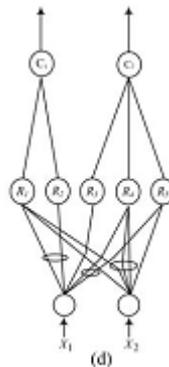


Figure (6.3) réseau neuro-flou du groupe 3

VI.5 L'intérêt des systèmes neuro-flous dans le contrôle :

Le contrôleur neuro-flou, permet une exploitation maximale des connaissances linguistiques, et facilite la tâche du constructeur en ayant des paramètres ajustables automatiquement pour réaliser le meilleur compromis entre robustesse et performance.

Ce régulateur évite au constructeur de chercher le modèle mathématique du système, et ne demande aucun effort à la détermination des paramètres du régulateur, il faut juste spécifier le nombre de paramètres et ils vont converger vers leurs valeurs nominales par la façon la plus optimale selon l'algorithme d'apprentissage utilisé.

Il nous permet la réalisation du contrôle en temps réel, et le rejet de tout type de perturbations. La dynamique du rejet dépend de l'algorithme d'apprentissage, dans le cas du Backpropagation il dépend du gain d'apprentissage.

VI.6 Le contrôleur neuro-flou basé sur le modèle de conclusions de Takagi-Sugeno :

VI.6.1 Structure du contrôleur neuro-flou basé sur le modèle de conclusions de Takagi-Sugeno :

Ce contrôleur à la structure d'un réseau de neurones à 5 couches feedforward, dont chaque couche est la projection d'une partie de fonctionnement du contrôleur flou de type Takagi-Sugeno, les fonctions de chaque neurone et leurs nombre dans chaque couche dépendent des fonctions d'appartenances définies pour le contrôleur.

Le schéma suivant représente ce type de contrôleur neuro-flou pour un vecteur d'entrée de dimension n :

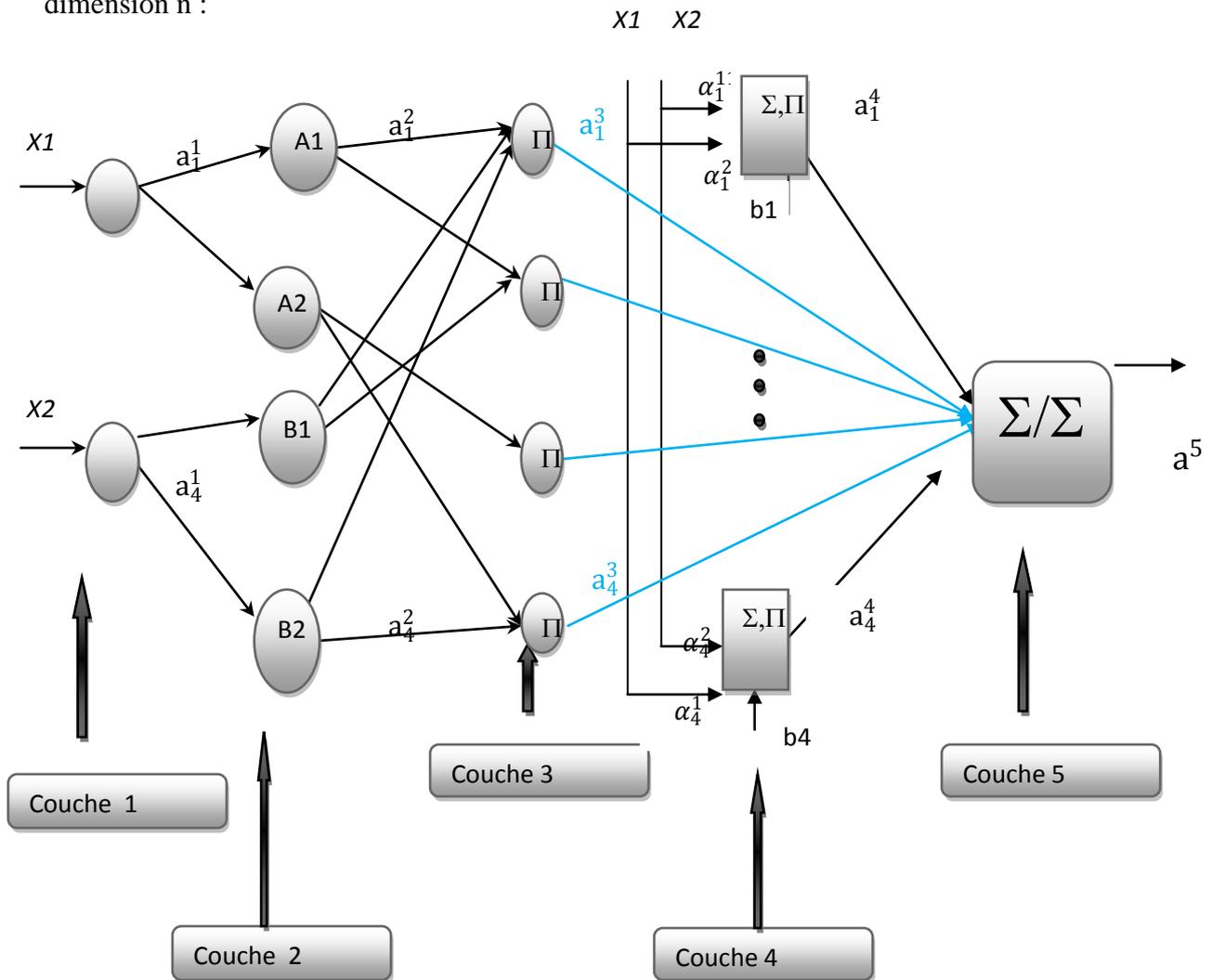


Figure (6.4) structure du contrôleur neuro-flou basé sur le modèle de règles de Sugeno

a_i^j La sortie du neurone i de la couche j .

A_i, B_i , sont les fonctions d'appartenance de x_1 et x_2 .

α_j^i Les poids synaptique w_j^i

B_i les seuils.

La couche 1 (la couche d'entrée/couche fuzzification) :

Le nombre de neurones dans cette couche est égal à la dimension du vecteur d'entrée. Cette couche lie chaque variable d'entrée X_i aux différentes fonctions d'appartenance la caractérisant (les neurones de la couche suivante) F_{ij} , avec $j=1$ à n_i . n_i est la dimension de l'ensemble de termes linguistiques associés à X_i . On dit que cette couche réalise la fuzzification des variables d'entrées.

Les neurones de cette couche sont caractérisés par un vecteur de pondération unité et fonction d'activation $f_1=1$.

La sorties de cette couche : $a_i^1 = x_i$ (6.1)

$i=1$: dimension(\mathbf{x})

La couche 2 :

Chaque neurone i de cette couche représente un terme linguistique, et il n'est connecté qu'à l'entrée caractérisée par ce terme, sa fonction d'activation f_i^2 égale à la fonction d'appartenance liée à ce terme.

Le nombre de neurones dans cette couche est égal au nombre total de termes décrivant toutes les variables d'entrées

L'entrée de chaque neurone i est donnée par :

$$u_i^2 = \begin{cases} a_1^1 & i = 1:n_1 \\ \vdots & \\ a_n^1 & n_{n-1} + 1:n \end{cases} \quad (6.2)$$

n_i : Le nombre de fonction de répartition associée à la variable X_i .

La sortie du neurone i est égale au degré d'appartenance de la variable floue x_i à f_i^2 donnée par la relation suivante :

$$a_i^2 = \exp\left(-\frac{u_i^2 - c_i^2}{2 * \sigma_i^2}\right) \quad (6.3)$$

σ_i et c_i Représentent la largeur et le centre de la fonction de répartition f_i^2 .

La couche 3 (la couche d'inférence) :

Le nombre de neurones dans cette couche est égal à $k = n_1 * n_2 * \dots * n_n$, et la fonction d'activation est la même pour tous les neurones de cette couche, c'est le produit cartésien flou. La sortie de chaque neurone est le degré d'activation de la règle dont il représente sa prémisse.

Sa sortie s'exprime comme suit :

$$a_i^3 = \prod_j u_j^3 = \prod_j a_j^2 \quad i=1 : k \quad (6.4)$$

J : prend les valeurs des indices des neurones de la couche précédente qui donnent, sur leurs sorties, les degrés d'appartenance des variables présentes dans la partie prémisse de la règle i.

La Couche 4 :

Calcule la valeur de la partie conclusion de chaque règle i, le nombre de neurones est identique à celui de la couche précédente.

A la différence des réseaux de perceptrons multicouches, les entrées de cette couche ne sont pas les sorties de la couche précédente, mais ce sont, en général, les mêmes entrées de la couche 1. Cela est dû à la forme des conclusions du système flou de type Takagi-Sugeno.

C'est la seule couche qui a des poids synaptiques ajustables, ils représentent les coefficients de la fonction polynomiale présente dans la partie conclusion des règles floues.

$$a_i^4 = bi + Wi * X = \alpha_i^0 + x_1 * \alpha_i^1 + \dots + \alpha_i^n * x_n \quad (6.5)$$

b_i le seuil de chaque neurone i

W_i le vecteur de pondération des connexions du neurone i avec le vecteur X.

Couche 5 (couche de defuzzification)

Composée d'un seul neurone pour les systèmes neuro-flous à une seule sortie, elle réalise la moyenne pondérée des sorties de chaque règle i (sortie de la couche 4).

$$a^5 = \frac{\sum_{j=1}^k a_j^3 * a_j^4}{\sum_{l=1}^k a_l^3} \quad (6.6)$$

VI.6.2 L'apprentissage du contrôleur neuro-flou se basant sur le modèle de Takagi-Sugeno :

L'apprentissage de ce contrôleur va permettre de déterminer les paramètres optimaux du régulateur neuro-flou de type Sugeno, pour que ce dernier puisse fournir sur sa sortie la commande nécessaire permettant au système à commander de suivre correctement, et de la meilleure façon, la référence désirée.

L'algorithme de Backpropagation avec la descente du gradient est utilisé pour entrainer le régulateur, les relations de mise à jour des paramètres sont obtenues comme suit :

Le critère à minimiser est fonction de la déviation entre la sortie désirée et la sortie réelle du système à commander :

$$J = 0.5 * (y - yd)^2 \quad (6.7)$$

y la sortie réelle du système.

yd la référence à imposer au système.

D'après la formule (4.13) on a

$$W_j^i(t + 1) = W_j^i(t) - \beta * \frac{\partial J}{\partial W_j^i(t)} \quad (6.8.a)$$

$$b_j(t + 1) = b_j(t) - \beta * \frac{\partial J}{\partial b_j(t)} \quad (6.8.b)$$

$$\frac{\partial J}{\partial W_j^i(t)} = \frac{\partial J}{\partial y} * \frac{\partial y}{\partial a^5} * \frac{\partial a^5}{\partial W_j^i(t)} \quad (6.9.a)$$

$$\frac{\partial J}{\partial b_j(t)} = \frac{\partial J}{\partial y} * \frac{\partial y}{\partial a^5} * \frac{\partial a^5}{\partial b_j(t)} \quad (6.9.b)$$

$$\frac{\partial J}{\partial y} = (y - yd) \quad (6.10)$$

$\frac{\partial y}{\partial a^5}$ C'est la variation de la sortie du système par rapport à la variation du signal de commande, qui n'est rien d'autre que la sortie du contrôleur neuro-flou.

$$\frac{\partial a^5}{\partial W_j^i(t)} = \frac{\partial f^5}{\partial W_j^i(t)} = \frac{\partial \left(\frac{\sum_{i=1}^k a_i^3 * a_i^4}{\sum_{i=1}^k a_i^3} \right)}{\partial W_j^i(t)}$$

$$\frac{\partial a^5}{\partial b_j(t)} = \frac{\partial f^5}{\partial b_j(t)} = \frac{\partial \left(\frac{\sum_{i=1}^k a_i^3 * a_i^4}{\sum_{i=1}^k a_i^3} \right)}{\partial b_j(t)}$$

On substitue la valeur de a_i^4 de l'équation (6.5) dans les deux formules de dérivation précédentes on trouve :

$$\frac{\partial a^5}{\partial W_j^i(t)} = \frac{a_j^3 * x_j}{\sum_{i=1}^k a_i^3} \quad (6.11.a)$$

$$\frac{\partial a^5}{\partial b_j(t)} = \frac{a_j^3}{\sum_{i=1}^k a_i^3} \quad (6.11.b)$$

Les nouvelles valeurs de paramètres du régulateur neuro-flou à l'instant $t+1$ sont :

$$W_j^i(t+1) = W_j^i(t) - \beta * (y - yd) * \frac{\partial y}{\partial a^5} * \frac{a_j^3 * x_j}{\sum_{i=1}^k a_i^3} \quad (6.12.a)$$

$$b_j(t+1) = b_j(t) - \beta * (y - yd) * \frac{\partial y}{\partial a^5} * \frac{a_j^3}{\sum_{i=1}^k a_i^3} \quad (6.12.b)$$

J : indice de la variable d'entrée.

I : indice de la règle.

W_j^i et b_j : Sont les coefficients de la représentation polynomiale des conclusions des règles du contrôleur de Takagi-Sugeno.

VI.7 Application sur le procédé de fermentation continue d'acide lactique :

On veut tester les performances et la robustesse de ce régulateur sur le système étudié dans le chapitre II.

Comme précisé dans le chapitre II, la commande du bioréacteur est le taux de dilution noté D , et sa sortie est la productivité noté Q , donc le régulateur neuro-flou doit fournir au système le taux de dilution D , pour que Q devienne et reste égale à Q_d , sortie désirée.

Les perturbations, peuvent se manifester sur la sortie, ou sur S_{in} (concentration d'alimentation du bioréacteur).

Le régulateur qu'on propose est le contrôleur neuro-flou. Utilisant l'algorithme d'apprentissage du Backpropagation à gain fixe, minimisant le critère :

$$J = 0.5 * (Q - Q_r)^2$$

Q : Sortie réelle du processus de fermentation.

Q_r : Sortie désirée.

VI.7.1 Le choix des variables linguistiques du régulateur :

Les entrées réelles du régulateur sont l'erreur ($e(t)$) et la dérivée de l'erreur ($de(t)$), ces mêmes notations seront utilisées pour représenter les variables floues, l'erreur et la dérivée de l'erreur.

Les fonctions d'appartenance de la variable floue, erreur sont présentées sur la figure suivante :

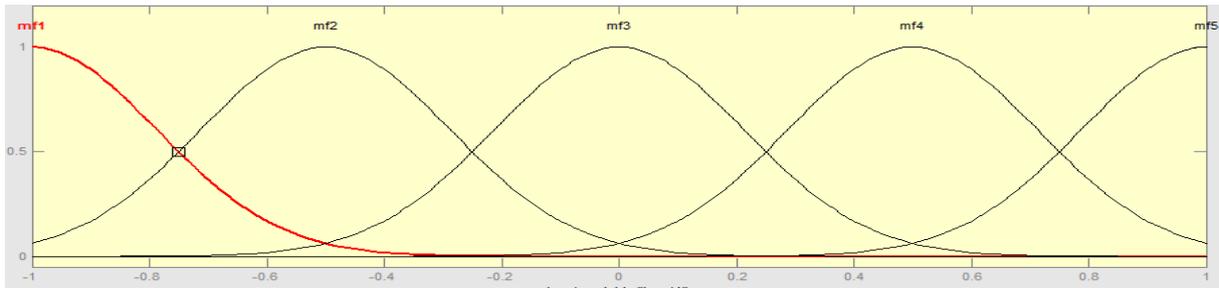


Figure (6.5) : Fonction d'appartenance de la variable floue normalisée, erreur.

Le gain de normalisation = 2/75.

Les fonctions d'appartenances de la variable floue dérivée de l'erreur sont présentées dans la figure suivante :

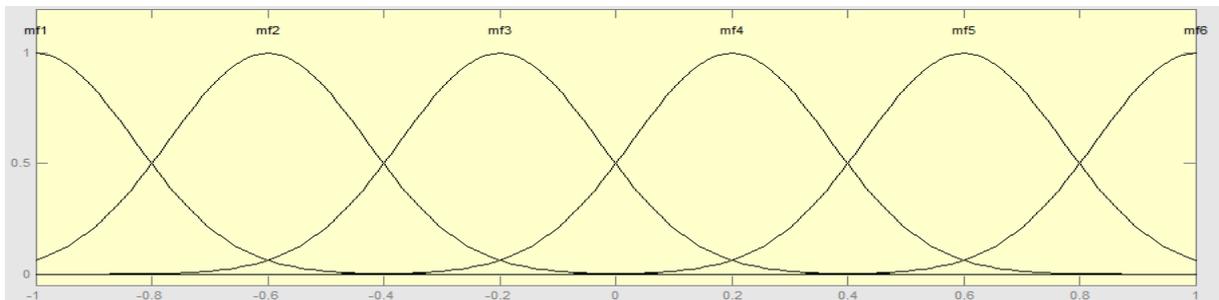


Figure (6.6) : les fonctions d'appartenance de la variable floue, dérivée de l'erreur normalisée.

Avec un gain de normalisation = 1/50.

La forme des conclusions est la suivante :

$$a_i^4 == \alpha_i^0 + e(t) * \alpha^1 + \alpha^2 * de(t)$$

$$\alpha_j^i(t + 1) = \alpha_j^i(t) - \beta * (y - yd) * \frac{\partial y}{\partial a^5} * \frac{a_j^3 * x_j}{\sum_{i=1}^k a_i^3}$$

$$\alpha_i^0(t + 1) = \alpha_i^0(t) - \beta * (y - yd) * \frac{\partial y}{\partial a^5} * \frac{a_j^3}{\sum_{i=1}^k a_i^3}$$

$x = (e(t), de(t))$

$a^5 = D$: commande appliquée au processus de fermentation, c'est aussi la sortie, du contrôleur neuro-flou.

VI.7.2 Simulations

VI.7.2.1 Teste de poursuite du régulateur neuro-flou :

Simulation1 :

Cette simulation est réalisée sous les contraintes suivantes :

- Un signal de référence égal à :
$$\begin{cases} 4 \frac{g}{l} \text{ pour } t \in [0,15]h \\ 2 \frac{g}{l} \text{ pour } t \in [15,30]h \end{cases}$$
- Une initialisation aléatoire des paramètres, ajustables, du régulateur neuro-flou , dans $[0,0.001]$.
- Un taux d'apprentissage fixe = 0.004

Les résultats obtenus sont les suivants :

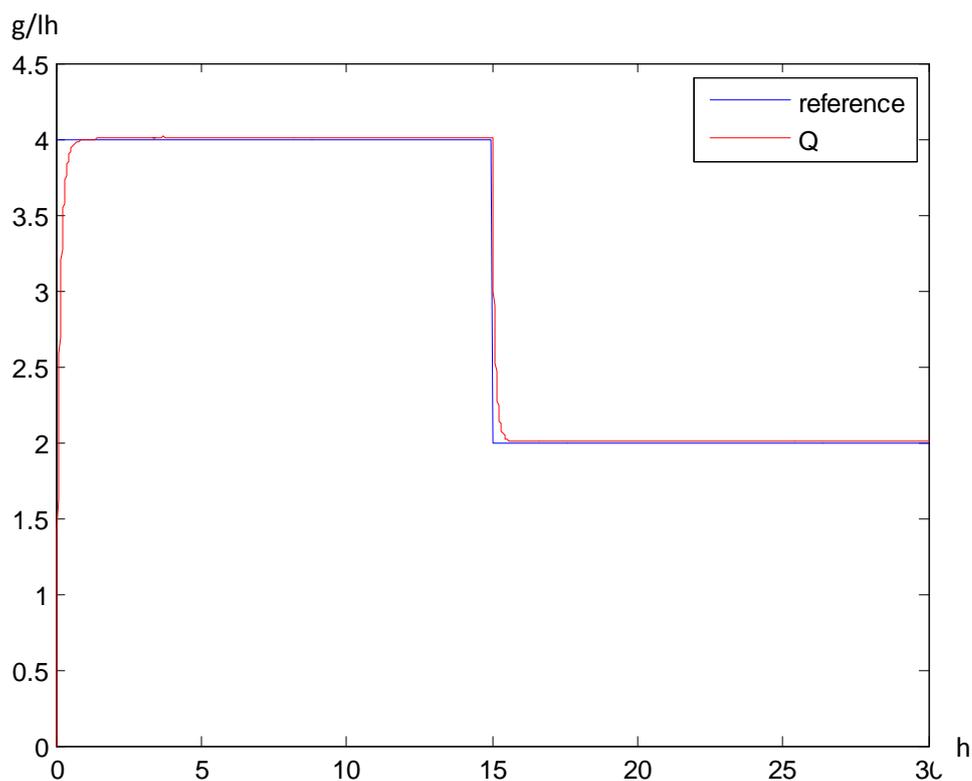


Figure (6.7) : test1 de la poursuite du régulateur neuro-flou

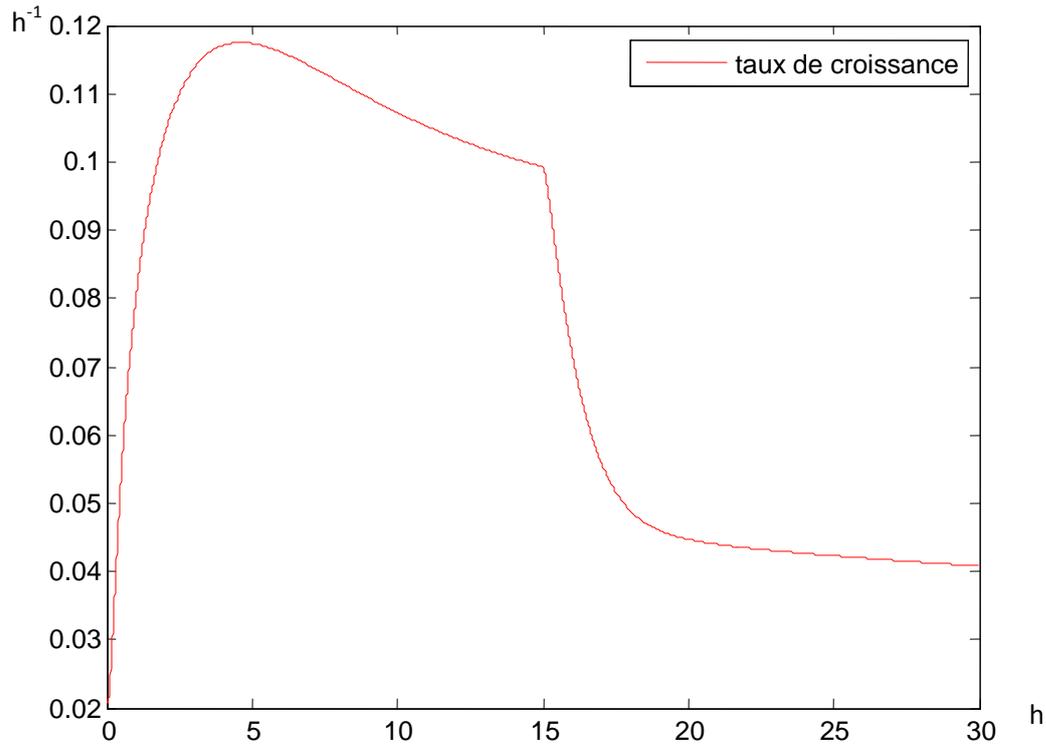


Figure (6.8) : taux de croissance (h^{-1}), pour le test1 de la poursuite.

Simulation2 :

Cette simulation est réalisée sous les contraintes suivantes :

- Un signal de référence y_r égal à :

$$\frac{dy_r}{dt} + 0.33 * y_r(t) = 0.33 * U(t)$$

$$U(t) = \begin{cases} 4 \frac{g}{l} & \text{pour } t \in [0,15]h \\ 2 \frac{g}{l} & \text{pour } t \in [15,30]h \end{cases}$$

- Une initialisation aléatoire des paramètres, ajustables, du régulateur neuro-flou , dans $[0,0.001]$.
- Un taux d'apprentissage fixe = 0.004

Les résultats obtenus sont les suivants :

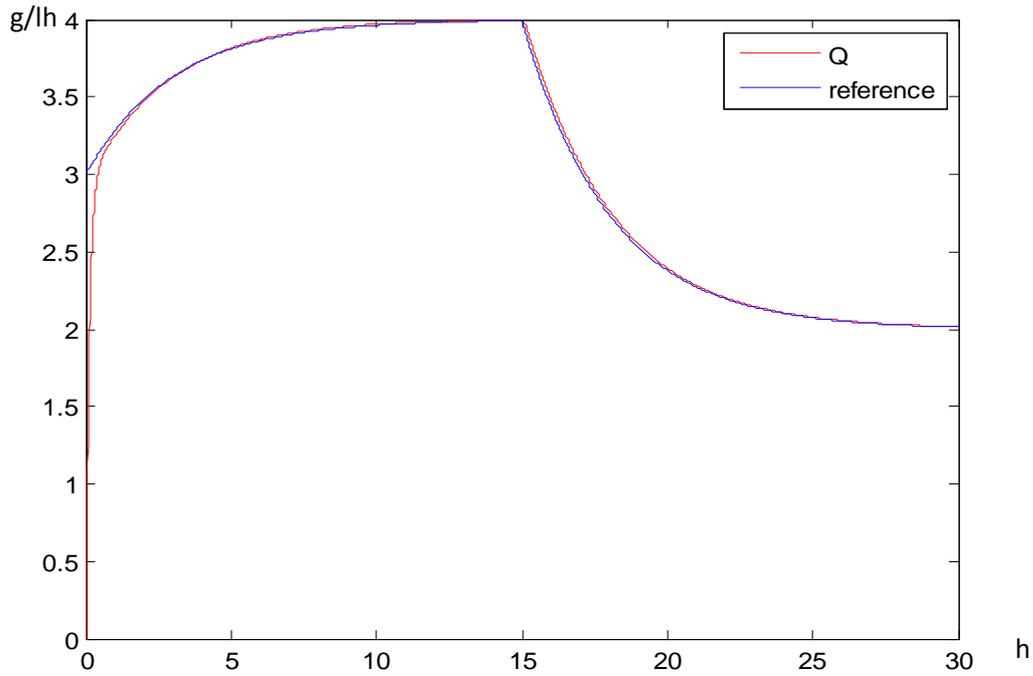


Figure (6.9) : test2 de la poursuite du régulateur neuro-flou

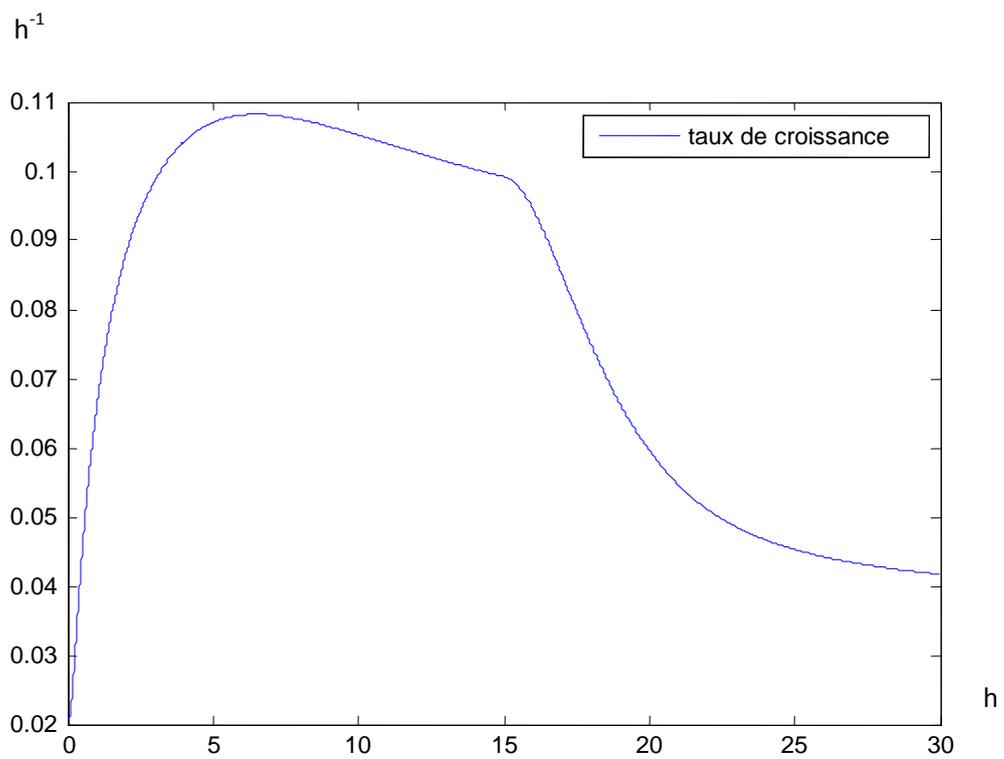


Figure (6.10) : taux de croissance (h^{-1}), pour le test 2 de la poursuite.

Simulation3 :

Cette simulation est réalisée sous les contraintes suivantes :

- Un signal de référence y_r égal à : $y_r(t) = 1 + \sin(t)$
- Une initialisation aléatoire des paramètres, ajustables, du régulateur neuro-flou, dans $[0, 0.001]$.
- Un taux d'apprentissage fixe = 0.004

Les résultats obtenus sont les suivants :

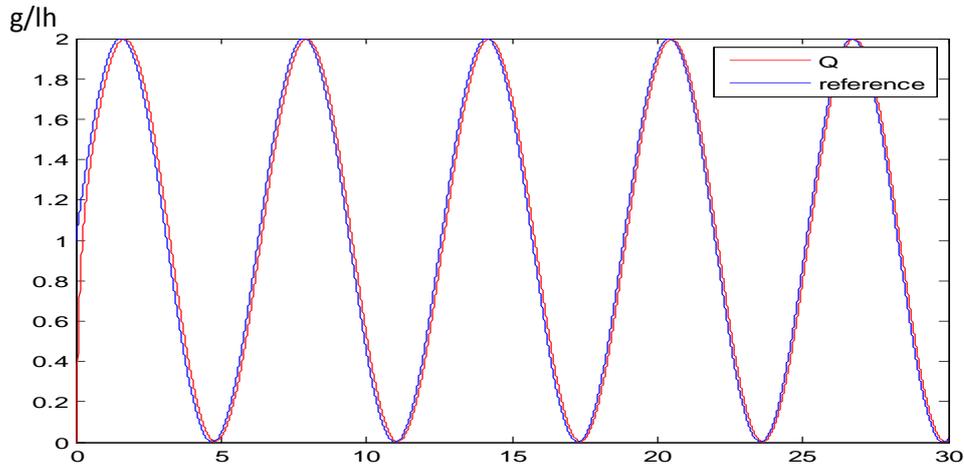


Figure (6.11) : test 3 de la poursuite du régulateur neuro-flou

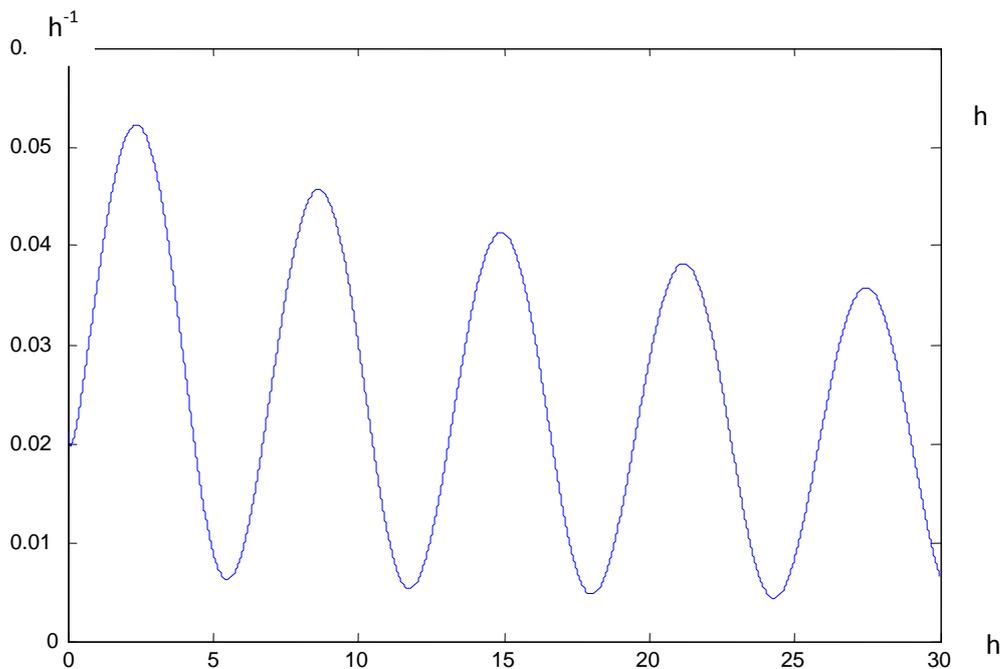


Figure (6.12) : taux de croissance (h^{-1}), pour le test3 de la poursuite.

Simulation 4 :

Cette simulation est réalisée sous les contraintes suivantes :

- Un signal de référence y_r égal à : $y_r(t) = 1 + \sin(t)$
- Une initialisation aléatoire des paramètres, ajustables, du régulateur neuro-flou , dans $[0, 0.001]$.
- Un taux d'apprentissage fixe = **0.02**
- Les résultats obtenus sont les suivants :

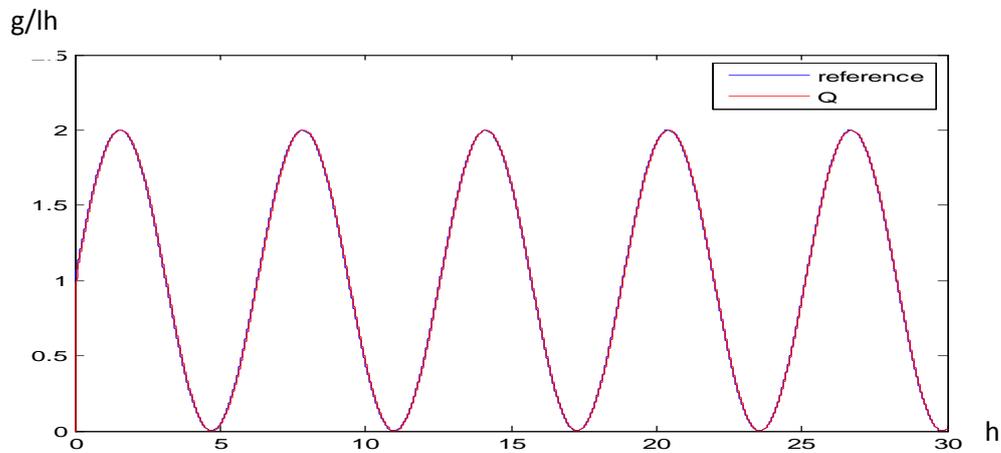


Figure (6.13) : test 4 de la poursuite du régulateur neuro-flou

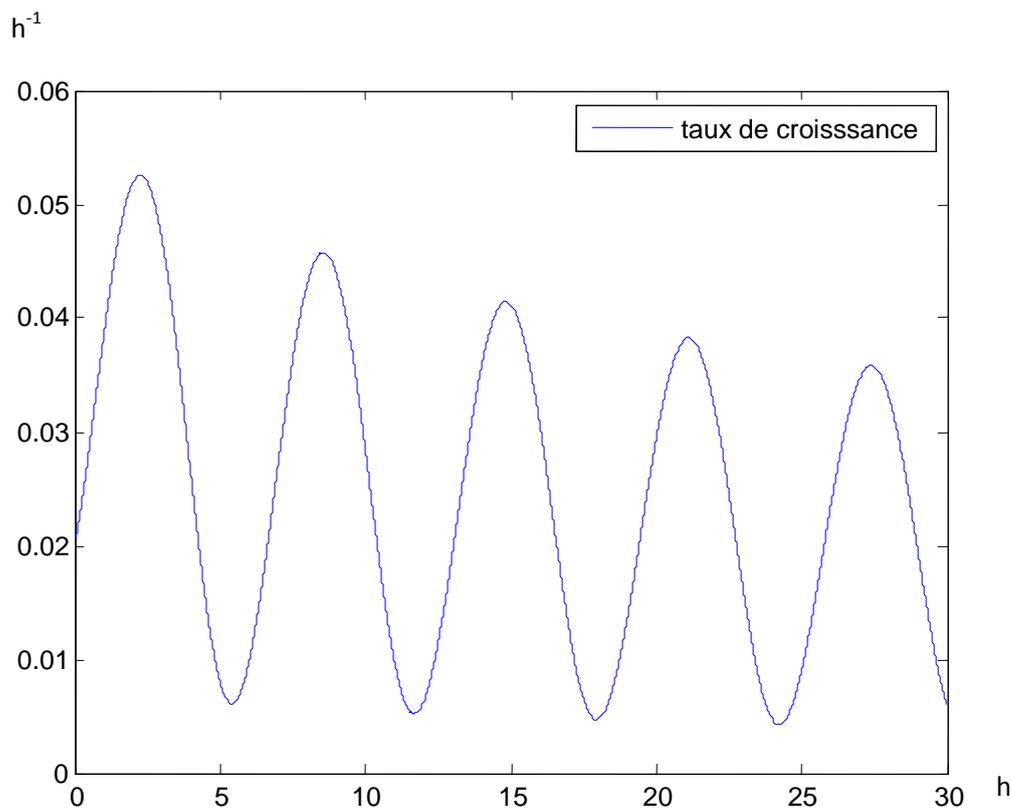


Figure (6.14) : taux de croissance (h^{-1}), pour le test3 de la poursuite.

VI.7.2.2 Teste de rejet de perturbation, sur Sin, du régulateur neuro-flou :

Simulation 5 test de performance (rejet de perturbations)

Cette simulation est réalisée sous les contraintes suivantes :

- Un signal de référence y_r égal à : $y_r(t) = 4 \text{ g/lh}^{-1}$
- Une initialisation aléatoire des paramètres, ajustables, du régulateur neuro-flou , dans $[0, 0.001]$.
- Un taux d'apprentissage fixe = 0.004
- Une perturbation de + 100% sur la concentration de Sin, à partir de l'instant $t=5\text{h}$.

Les résultats obtenus sont les suivants :

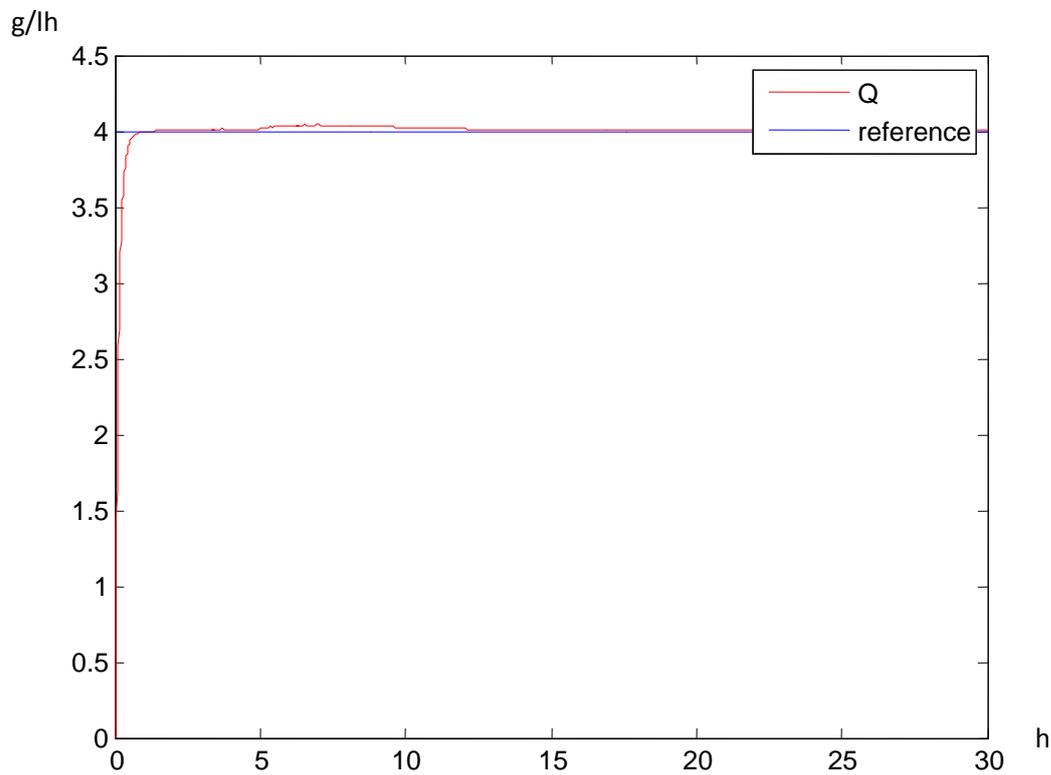


Figure (6. 15) : test de rejet de perturbations sur Sin, par le contrôleur neuro-flou

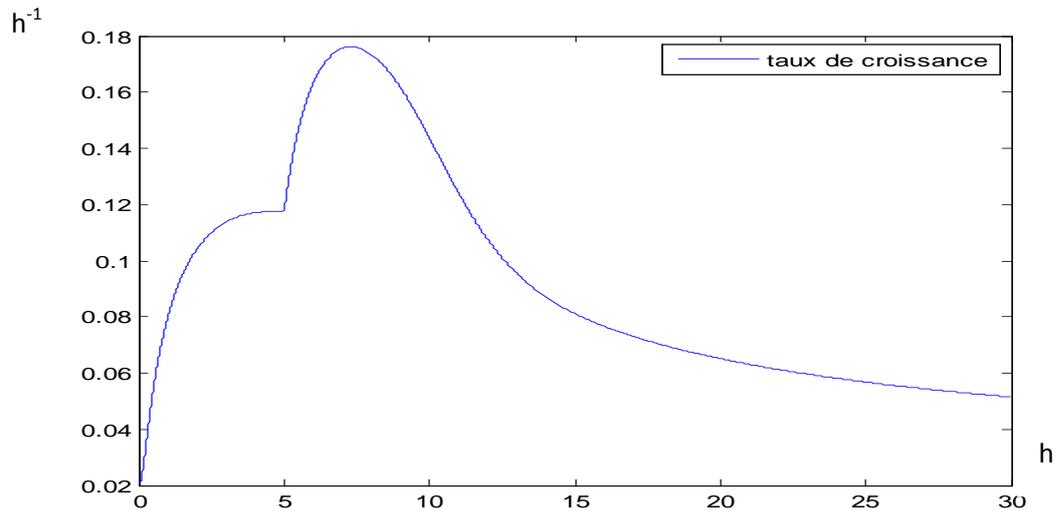


Figure (6.16) : taux de croissance, en présence de perturbation sur Sin

VI.7.2.3 Teste du rejet de perturbation, sur la sortie du système, du contrôleur neuro-flou :

Simulation 6 test de performance (rejet de perturbations)

Cette simulation est réalisée sous les contraintes suivantes :

- Un signal de référence y_r égal à : $y_r(t) = 4 \text{ g/lh}^{-1}$
- Une initialisation aléatoire des paramètres, ajustables, du régulateur neuro-flou , dans $[0, 0.001]$.
- Un taux d'apprentissage fixe = 0.004
- Une perturbation créneau égale à $+0.5 \text{ g/l h}$ sur la sortie du système pendant l'intervalle $[1, 10] \text{ h}$.

Les résultats obtenus sont les suivants :

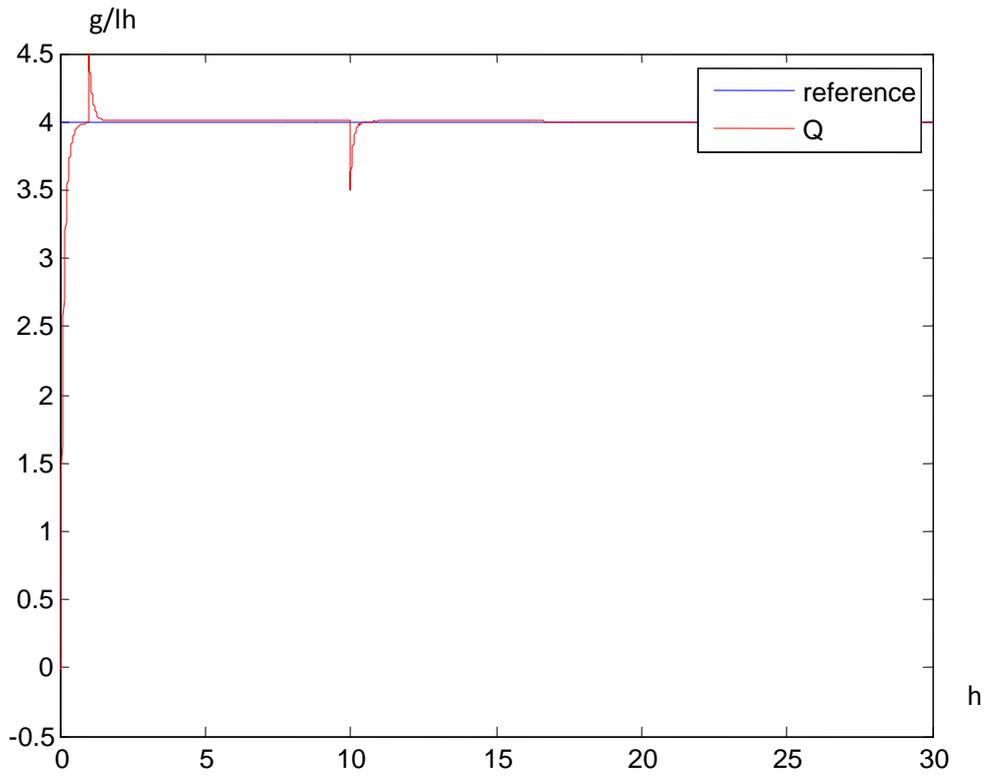


Figure (6.17) : test de rejet de perturbation sur la sortie du système.

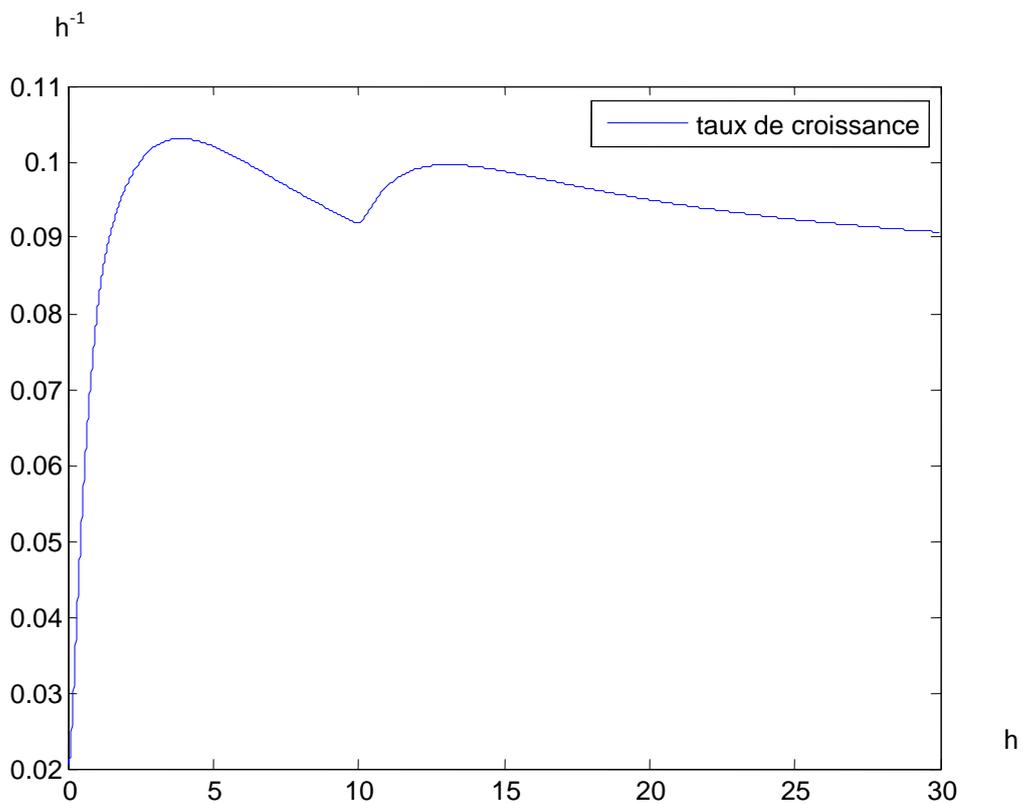


Figure (6.18) : taux de croissance, en présence de perturbations sur la sortie du système.

VI.7.2.4 test de robustesse du contrôleur neuro-flou :**Simulation7 :**

Pour une référence de type échelon $Q=4\text{g/l}\cdot\text{h}^{-1}$, sous les contraintes suivantes :

- Des erreurs de modélisations de 50%, introduites à l'instant $t=10\text{ h}$, sur les paramètres : K_m, K_i, μ_0, Y_{xs} .
- Une initialisation aléatoire des paramètres, ajustables, du régulateur neuro-flou, dans $[0, 0.001]$.
- Un taux d'apprentissage fixe $\lambda=0.004$.

Nous obtenons les résultats suivants :

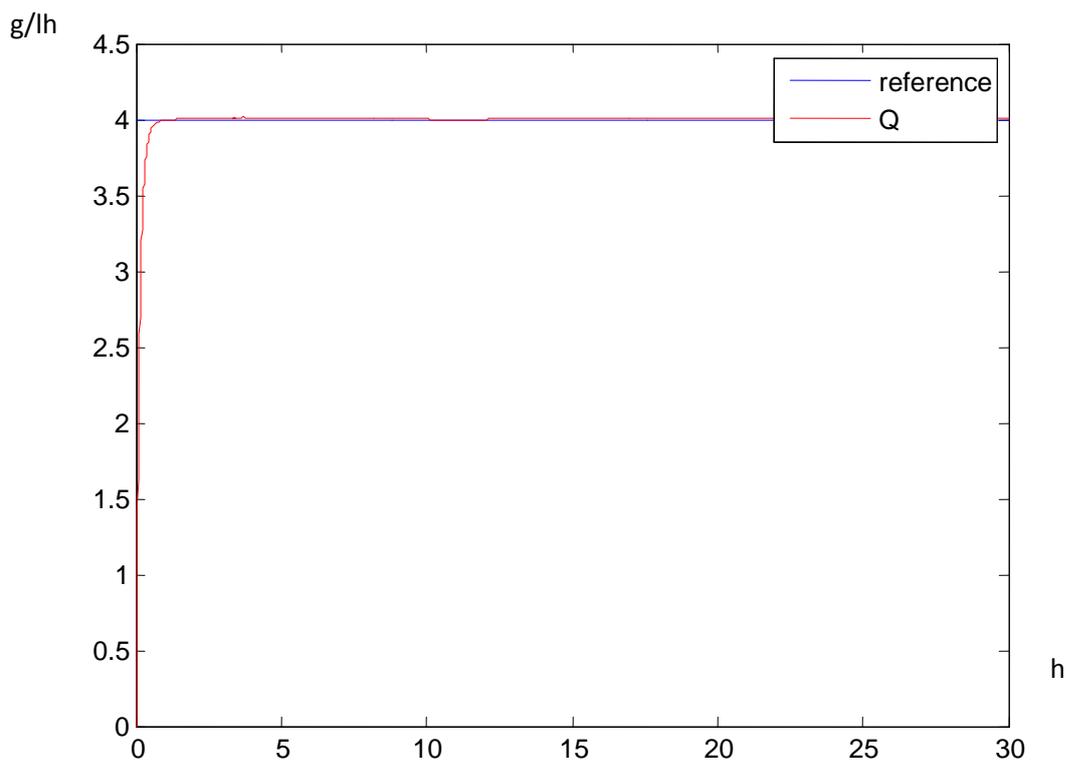


Figure (6.19) : teste de robustesse du régulateur neuro-flou

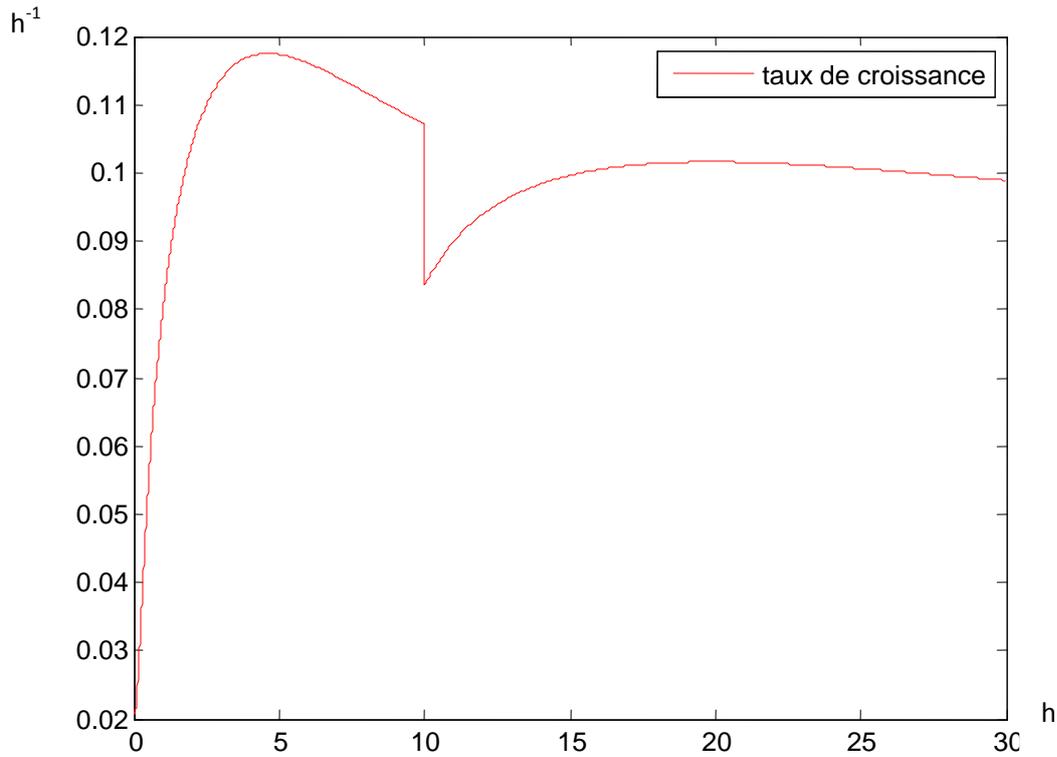


Figure (6.20) : taux de croissance, en présence d'erreur de modélisation.

VI.7.2.5 test sur l'effet des valeurs initiales, des paramètres, du contrôleur neuro-flou :

Simulation8 : test de l'effet, d'un choix de valeurs initiales, grandes, des paramètres du contrôleur neuro-flou.

Pour une référence de type échelon $Q=4g/l \cdot h^{-1}$, sous les contraintes suivantes :

- Une initialisation aléatoire, dans l'intervalle **[0,1]**, des paramètres ajustables, du régulateur neuro-flou
- Un taux d'apprentissage fixe $\lambda = 0.004$.

Nous obtenons les résultats suivants :

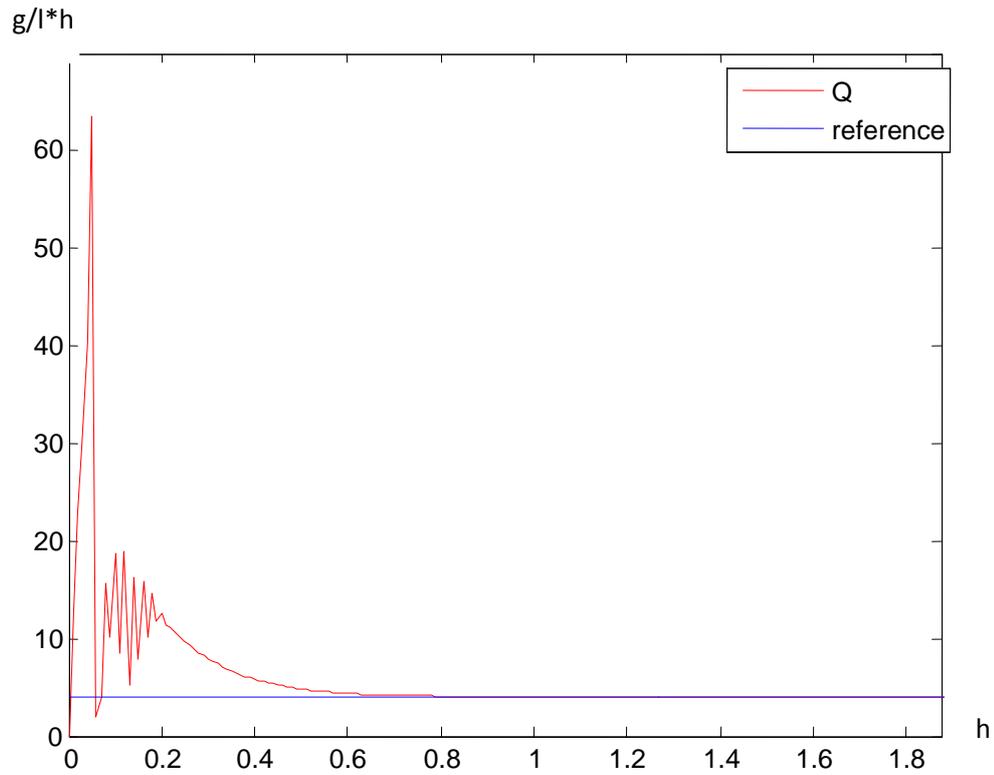


Figure (6.21) : test de l'effet d'un choix de valeurs initiales importantes des paramètres du contrôleur neuro-flou.

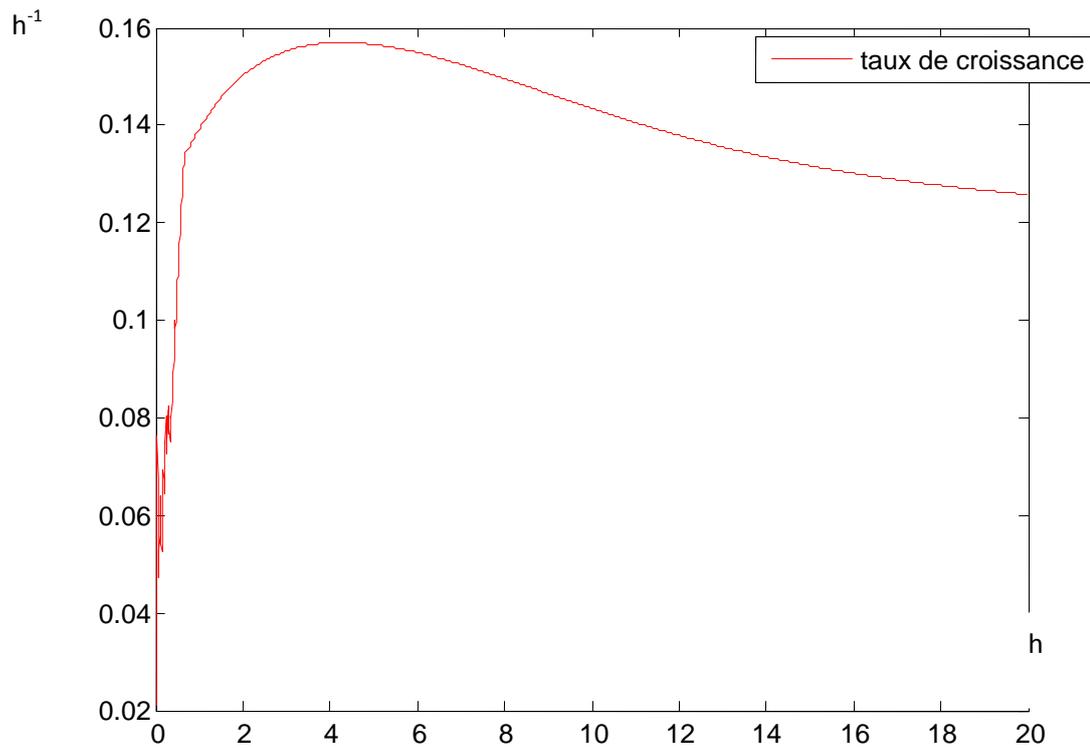


Figure (6.22) : taux de croissance, pour des valeurs initiales importantes des paramètres du contrôleur neuro-flou.

VI.7.3 Etude des résultats de simulations :

La Figure (6.7) et la Figure (6.9), montrent que :

- le contrôleur neuro-flou réalise la poursuite parfaite, lorsque le signal de référence est constant ou présente une dynamique souple (test 1 pendant les intervalles de temps où la référence est constante, test 2).
- Une erreur apparaît au moment des variations brusque du signal de référence (test 1 à $t=15h$).

La comparaison de la figures (6.13) et la figure (6.11), montrent que :

- Le système suit la dynamique de référence avec un retard (test 3), si le contrôleur neuro-flou a un gain d'apprentissage très faible, et la référence varie rapidement. Un taux d'apprentissage faible, entraîne des adaptations lentes des paramètres, ne permettant pas au système de suivre la référence si cette dernière a une dynamique trop rapide.
- Pour le même signal de référence (test 4 et test 3), mais avec un taux d'apprentissage plus important, la sortie du système suit parfaitement la référence.

La simulation 5, figure (6.15), montre, qu'en présence de perturbation sur la concentration d'alimentation en substrat, une erreur de l'ordre de 1% apparaît sur la sortie du système, qui devient nulle après certain temps.

La simulation 6, figure (6.17) montre que le régulateur neuro-flou permet un rejet parfait des perturbations sur la sortie du système. Nous remarquons, d'après La figure (6.18), que le taux de croissance diminue lorsque la perturbation apparaît sur la sortie. Ceci s'explique par le rôle du contrôleur, qui a diminué la valeur de la commande (taux de dilution).

Aucune déviation sur la sortie n'est observée sur la Figure (6.19), simulation 7, en présence de variations paramétriques de 50%. Ceci s'explique par la robustesse du contrôleur neuro-flou liée à sa capacité d'adaptation en fonction de l'erreur.

La simulation 8, figure (6.22) montre que lorsque les valeurs initiales sont trop grandes, la sortie du régulateur devient oscillatoire. Ce qui entraîne de fortes oscillations sur la sortie du système, et la convergence, si elle a lieu, devient lente, figure (6.21).

Les résultats de simulations ont montré que l'approche neuro-floue est idéale pour réaliser les performances désirées. Elle est robuste via les incertitudes du système, sous certaines conditions de choix des valeurs initiales des paramètres et du taux d'apprentissage.

VI.8 Conclusion :

Nous concluons, à partir de l'étude des simulations réalisée dans la partie précédente que :

- La stabilité du contrôleur neuro-flou, et sa rapidité de convergence sont relatives aux choix des valeurs initiales des paramètres ajustables du régulateur.

- Un choix des valeurs initiales trop éloignées de leurs valeurs nominales peut rendre le contrôleur neuro-flou oscillatoire ou le déstabilise.
- Le choix du taux d'apprentissage d'un contrôleur neuro-flou, à taux d'apprentissage fixe, est un facteur important affectant la rapidité de convergence, et la poursuite des références qui varient rapidement.
 - Pour un taux d'apprentissage trop faible, la dynamique de convergence devient lente, et la sortie présentera un retard par rapport aux références à dynamique très rapide.
 - Un taux d'apprentissage important, rend la convergence plus rapide, et assure une poursuite parfaite des références qui varient rapidement, mais, dans la majorité des cas, déstabilise ce contrôleur neuro-flou.
 - Le bon choix de ce paramètre, est celui réalisant le meilleur compromis entre la rapidité de convergence de la sortie du système et la stabilité du contrôleur neuro-flou.
- L'utilisation d'un taux d'apprentissage adaptatif pour ce contrôleur, est une solution des problèmes d'un choix du taux d'apprentissage fixe, lorsque l'utilisateur n'a pas assez d'expérience sur le système.
- Le régulateur neuro-flou, appliqué à notre système, permet un rejet parfait des erreurs, sur la sortie du système, et sur la concentration d'alimentation en substrat.
- Un régulateur neuro-flou est très robuste vis-à-vis des variations paramétriques.

On conclut, finalement, que ce régulateur neuro-flou, à taux d'apprentissage fixe, est parfait pour le contrôle de la fermentation étudiée. Il réalise un rejet parfait des perturbations et une robustesse vis-à-vis des variations paramétriques (phénomène souvent rencontré dans ce type de réaction, ayant des paramètres du modèle très sensibles aux variations des conditions environnementales).

Seul le choix approprié, des valeurs initiales des paramètres de ce régulateur, est nécessaire pour éviter les oscillations en sortie du système.

Un taux d'apprentissage adaptatif n'est pas nécessaire pour notre cas. La recherche du meilleur taux d'apprentissage fixe n'est pas importante, car la dynamique de notre système est de nature lente, et le choix du taux d'apprentissage assez faible, pour éviter les oscillations pendant la convergence, devient suffisant, étant donné que les références à appliquer à notre système sont de type échelon, et la dynamique de convergence est très rapide pour des taux d'apprentissage très fiables (simulation 1, Figure (6.7)).

Conclusion générale
Et
Perspectives

Conclusion générale

Dans ce projet de fin d'études, notre but consiste à déterminer un contrôleur en réalisant, une commande qui permet la robustesse et la performance d'un réacteur biotechnologique de fermentation continue d'acide lactique. Nous avons proposé deux approches de contrôle différentes, de la plus simple à la plus complexe, au fur et à mesure des limites rencontrées pour chaque contrôleur.

Le contrôle des processus nécessite une étude technologique préalable, du processus à commander.

Nous avons conclu, de l'étude bibliographique du métabolisme fermentaire, que les paramètres de ce dernier sont très sensibles aux variations des facteurs environnementaux (PH, T°,...etc.).

L'Etude des états d'équilibres du modèle de fermentation continue d'acide lactique, que nous avons établi dans le chapitre II, a permis de conclure que ce type de procédé présente une caractéristique intéressante, dite "input multiplicities".

Cette caractéristique s'explique de la manière suivante : Pour la même valeur de sortie désirée (la productivité), deux valeurs de commandes (taux de dilution) peuvent être appliquées en entrée.

Nous concluons, à partir de l'étude de la stabilité du système non linéaire, que le plus faible taux de dilution correspond à l'état le plus stable, et que le plus grand taux de dilution correspond à l'état le moins stable.

La première loi de commande, mise en œuvre, est le PI linéaire, conçu à base du contrôleur à modèle interne. Les simulations réalisées sur le système en boucle fermée, ont montré que ce contrôleur reste stable, s'il est calculé et utilisé autour du même point d'équilibre.

Le régulateur PI linéaire devient instable s'il est utilisé, autour du point de fonctionnement correspondant au taux de dilution le plus petit, quand il est calculé autour du point de fonctionnement correspondant au taux de dilution le plus grand, et vis versa.

Ceci nous a permis de conclure, que le plus simple régulateur ne réalise pas les objectifs de commande, et nous a conduits à rechercher un contrôleur qui réalise la robustesse et la performance, complexe qu'il soit.

Le calcul du contrôleur PI linéaire est dépendant du modèle mathématique représentant le procédé. Et comme ce dernier est à paramètres variables, une variation d'un facteur influençant la fermentation ne permettra pas à ce régulateur de réaliser les objectifs de commande.

Nous nous sommes intéressés à la conception d'un contrôleur intelligent et indépendant du modèle mathématique.

L'étude Bibliographique, menée sur les systèmes flous et les réseaux de neurones, a permis de conclure que l'hybridation de ces deux approches d'intelligence artificielle, donne un système neuro-flou combinant les avantages des deux approches.

Nous avons décrit, par la suite, la structure du contrôleur neuro-flou, de type Sugeno, utilisé, et les fonctions d'appartenances de la partie prémisse, et l'algorithme d'apprentissage du contrôleur.

D'après les résultats de simulations, nous avons conclu, que le contrôleur neuro-flou, réalise les meilleures performances comparées à celles réalisées par le régulateur PI linéaire, lorsque ce dernier, est appliqué au même point de fonctionnement, pour lequel il a été calculé.

Le contrôleur neuro-flou est très robuste vis-à-vis des variations paramétriques du processus de fermentation.

Nous concluons, aussi, que le contrôleur neuro-flou est meilleur, pour la commande de fermentation, car le contrôleur le plus simple (PI linéaire) peut déstabiliser le système bouclé, et il est sensible aux variations paramétriques du processus de fermentation.

Perspectives :

Notre première perspective est l'application des lois de commandes développées, sur un processus réel de fermentation, et la validation des résultats théoriques obtenus.

L'autre perspective consiste à optimiser, les fonctions d'appartenances des variables d'entrées, et le nombre de neurones des couches 4 et 3 (correspondant au nombre des règles du contrôleur flou), par l'utilisation d'algorithmes génétiques.

Références bibliographiques

- Agnes N., "Production des protéines à partir de lactosérum brut". Thèse de 3^{ème} cycle, université de Lyon, France. 1986.
- Anderson J.A., "A simple neural network generating an interactive memory," *Mathematical Biosciences*, vol. 14, pp. 197-220, 1972.
- Augustin J.C. "Modélisation de la dynamique de croissance des populations de *Listeria monocytogenes* dans les aliments". Thèse Doctorat Université Claude Bernard Lyon I, 1999
- ATSUSHI .A (1995), "Internal Model Control Framework Using Neural Networks for the Modeling and Control of a Bioreactor" *Engng Applic. Artif. Intell.* Vol. 8, No. 6, pp. 689-701, 1995
- Baranyi J., Mc Clure P.J., Sutherland J.P., Roberts T.A. "Modeling bacterial growth responses". *J. Ind. Microbiol.*, 1993, 12, 190-194.
- Bernard GOSSELIN, thèse de doctorat : "application de reseaux de neurones artificiels a la reconnaissance automatique de caracteres manuscrits", Faculté Polytechnique de Mons, 1996
- Berenji, H.R. et P. Khedkar (1992). "Learning and tuning fuzzy logic controllers through reinforcements". *IEEE Transactions on Neural Networks* 3(5), 724-740
- Boudjema Khaled. "Optimisation de la production d'acide lactique sur lactosérum par *Streptococcus thermophilus*", thèse de magistère. 2008.
- Braatz.D.R., "Improved Filter Design in Internal Model Control". *Ind. Eng. Chem. Res.* 1996, 35, 3437-3441.
- Brosilow, C.. "personal communication", Case Western Reserve University, which implies Cleveland, OH, 1983
- Brosilow, C.; Markale, N. "Model Predictive Cascade Control and its Implications for Classical and IMC Cascade Control". AICHE Annual Meeting, Miami Beach, FL, 1992
- Buchanan, R.E. 1918. "Life phases in a bacterial culture". *J. Infect. Dis.* 98, 1271-1273
- Chen, D. E. Seborg, "PI/PID controller design based on direct synthesis and disturbance rejection". *Ind. Eng. Chem. Res.*, 41, 2002, 4807-4822.
- Delignette-Muller, M.L. 1998. "Relation between the generation time and the lag time of bacterial growth kinetics". *Int. J. Food Microbiol.* 43, 97-104.
- Dempster A.P (1967). "Upper and Lower Propabilities Induced by a Multivalued Mapping". *Annals of Mathematical Statistics* 38, pp. 325-339.
- Deng, Xu et Li (2009). "A novel neural internal model control for multi-input multi-output nonlinear discrete-time processes". *Journal of Process Control* 19 (2009) 1392-1400
- Dochain, D. (2001). "Automatique des bioprocédés". Hermes Science.
- Doyle, J.; Stein, G. *IEEE Trans. Autom. Control* 1981, AC-26, 4. Morari et Holt (1984, 1985)

- Garcia, C. E., Morari, M. "Internal Model Controls 1. A Unifying Review and Some New Results". *Ind. Eng. Chem. Process Des. Dev.* 1982, 21, 308
- Garcia, C.E. et M. Morari. "Internal Model Control-3. Multivariable control law computation and tuning guidelines,". *Ind. Eng. Chem. Process Des. & Dev.*, 24, pp.484-494.(1985).
- George, S.M., Lund, B.M. 1992. "The effect of culture medium and aeration on growth of *Listeria monocytogenes* at pH 4.5". *Lett. Appl. Microbiol.* 15, 49-52
- Hayken, S. "Neural Networks-a Comprehensive Foundation", Macmillan College Publishing Company Inc., 1994
- Hebb, D. O., "The organization of Behavior", Wiley, New York, 1949.
- Holt, B. R.; Morari, M. *Chem. Eng. Sci.* 1985, 40, 59
- Hopfield J.J., "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National Academy of Sciences of the USA*, vol. 79 no. 8 pp. 2554-2558, April 1982.
- Horowitz, I.M. (1963). "Synthesis of feedback systems," Academic Press
- Jang, J.-S.R. (1993). "Anfis: adaptive-network-based fuzzy inference system". *IEEE Transaction on System Man and Cybernetics* 23(3), 665-685
- Jarry A. "Production industrielle d'acide lactique in : les bactéries lactiques". T1, ED Loriga, Lavoisier, Paris, 604 p. 1994.
- Jenkins D. L. and K. M. Passino. "Introduction to nonlinear analysis of fuzzy control systems". Unpublished paper, 1996.
- Johansen, C., Gram, L., Meyer, A.S. 1994. "The combined inhibitory effect of lysozyme and low pH on growth of *Listeria monocytogenes*". *J. Food Prot.* 57, 561-566
- Johnson M.J. & Berger J. (1942). "The enzymatic properties of peptidases". *Adv. Enzymol.* 2, 69-89
- Kevin M. Passino et Stephen Yurkovich, "Fuzzy Control", 1998.
- Kim, H. M. and Wang, F. Y.: "Design of adaptive neuro-fuzzy controllers", *Proc. of 1994 IEEE Int. Conf. on Systems, Man, and Cybernetics*, 1994
- Wang, F. Y. and Kim, H. M.: "Implementing adaptive fuzzy logic controllers with neural networks: a design paradigm", *Journal of Intelligent and Fuzzy Systems* 3(2) (1995), 165-180
- Klir, G. and Yuan, B (1995). "Fuzzy sets and Fuzzy Logic : Theory and Applications", Prentice Hall, Upper Saddle River, NJ.
- Stengel. F. "Toward intelligent flight control". *IEEE Trans. on Systems, Man, and Cybernetics*, 23(6):1699-1717, Nov./Dec. 1993.
- Kohonen T. "The Self-Organising Map proc". of the *IEEE*, vol 78, n°9, pp 1464-1480, Septembre 1990

Kosko, B. (1992). 'Neural networks and fuzzy systems'. Prentice Hall, Englewood Cliffs, Nj

Ledesma O.V., de Ruitz Hoigado AP., Oliver G., De Giori G.S., Raiband P. & Galpin J.V. (1977) "A synthetic medium for comparative nutritional studies of lactobacilli". *J. Appl. Bacteriol.* 42, 123-133

Lin, C.T. et C.S.G. Lee (1991). Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on Computers* 40(12), 1320.

Lenoir J., Hermier J., Weber. F., 1992. "Les groupes microbiens d'intérêt laitiers"; Ed Ciplil, p30-50

Lippmann.R "An Introduction To Computing with Neural Nets". *IEEE ASSP Magazine*, volume 4, n° 2, pp 4-22, Avril 1987

Luedeking R and Piret E. L., 1959 (a). "A Kinetic study of the lactic acid fermentation batch process at controlled pH". *Journal of biochemical, microbiology and technology engineering*, 1 : 393-412.

MESSENS W., DE VUYST L. "Inhibitory substances produced by Lactobacilli isolated from sourdoughs-a review". *Int. J. Food Microbiol.*, 2002, 72, 31-43.

Minsky.M and S. Papert, 'Perceptrons', Cambridge, MA: MIT Press, 1969.

Orla- Jensen, 1919. "The lactic bacteria Hosled son.CopenHagen". 74 : 131-142

RAYKOWSKY et AL (1983) Ratkowsky D.A., Lowry R.K, Mc Meekin T.A.,Stokes A.N., Chandler RE. "Model forbacterial culture growth rate throughout theentire biokinetic temperature range.*Journal of bacteriol*"., 1983, 154:1222-1226

Robinson, T.P., Ocio, M.J., Kaloti, A., Mackey, B.M. 1998. "The effect of the growth environment on the lag phase of *Listeria monocytogenes*". *Int. J. Food Microbiol.* 44, 83-92

Johnson M.J. & Berger J. (1942) , "The enzymatic properties of peptidases" *Adv. Enzymol.* 2, 69-89

Refenes, A.-P. and Zaidi, A.(1995), "Managing Exchange-Rate Prediction Strategies with Neural Networks, in Refenes A.-P. (ed), "Neural Networks in the Capital Markets", Wiley Finance Edition, New York, pp. 213-219.

Rivera D.E , M. Morari, S. Skogestad, "Internal model control. 4. PID controller design," *Ind. Eng. Chem. Process Des. Dev.*, 25, 1986, 252-265.

Rosenblatt .F, "The Perceptron: A probabilistic model for information storage and organization in the brain , "Psychological Review, vol. 65, pp. 386-408, 1958.

Rosso, L. 1995. "Modélisation et Microbiologie Prévisionnelle : Elaboration d'un Nouvel Outil pour l'Agroalimentaire". Thèse de Doctorat, Université Claude Bernard - Lyon I, n° 95-197.

Rosso, L. 1998. "Quantification et prévision du comportement dynamique des populations microbiennes : Applications à la sécurité des aliments". Mémoire d'Habilitation à Diriger les Recherches, Université de Paris-Sud.

Rumelhart, D. E., and McClelland, J L.(eds.), "Parallel distributed Processing", vol.I, M.I.T.Press, Cambridge,MA, 1986.

- Scales L.E., "Introduction to non-linear optimization", New York :Springer-Verlag, 1985
- SETNES, M., BABUŠKA, R., VERBRUGGEN, H.B., "Rule-based modelling: precision and transparency", *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*, vol.28, n1, pp.165-169, 1998.
- Shankar K. & Bard R.C. (1955) "Effect of metallic ion in the growth, morphology, and metabolism of *Clostridium perfringens*". 69, 436-443
- Song, Y., Johns, A.T.: "Applications of Fuzzy Logic in Power Systems, Part 2", *IEE Power Engineering Journal*, Aug. 1998, pp. 185–190
- Song, Y., Johns, A.T.: "Applications of Fuzzy Logic in Power Systems, Part 1", *IEE Power Engineering Journal*, Oct. 1997
- Takagi .T and M. Sugeno. "Fuzzy identification of systems and its applications to modeling and control". *IEEE Trans. on Systems, Man, and Cybernetics*, 15(1):116–132, January 1985
- Wang H.O, K. Tanaka, and M. F. Griffin, "An Approach to Fuzzy Control of Nonlinear Systems: Stability and Design Issues," *IEEE Trans. Fuzzy Syst.*, Vol. 4. No. 1, pp. 1423 1996
- Webb M. (1948). "The influence of magnesium on cell division". 1. The growth of *Clostridium welchii* in complex media deficient in magnesium". *J. Gen. Microbiol.* 2, 275-287
- Webb M. (1949). "The influence of magnesium on cell division". II. The effect of magnesium on the growth and cell division of various bacterial species in complex media". *J. Gen. Microbiol.* 3, 410-417
- Webb M. (1951a). "The influence of magnesium on cell division". IV. The specificity of magnesium". *J. Gen. Microbiol.* 5, 480-484
- Webb M. (1951b). "The influence of magnesium on cell division". V. The effect of magnesium on the growth of bacteria in chemically defined media of varying complexity". *J. Gen. Microbiol.* 5,484-495.
- Whiting et Buchanan (1993) Whiting RC., Buchanan R.L. "A classification of models for predictive microbiology". *Food Microbiol.*, 1993, 10, 175-177
- Widrow, B., and Hoff, M.E., Jr., "Adaptativ Switcing Circuits," *IRE WESCON Convention Record*, pt. 4, 96-104, Septembre 1960.
- Yamada Kou" Modified Internal Model Control for unstable systems". *Proceedings of the 7th Mediterranean Conference on Control and Automation (MED99) Haifa, Israel - June 28-30, 1999*
- Rivera, D. E., S. Skogestad, et M. Morari. 1986. "Internal Model Control 4.PID Controller Design," *I&EC Chem. Proc. Des. & Dev.* 25, 252 – 265.
- Zadeh, L. (1965). "Fuzzy set", *Inf. Control*, vol. 8, pp. 535-537.
- Zamora M.C., Zaritzky N.E. " Modeling microbial growth in refrigerated packaged beef". *J. of Food Sci.*, 1985, 50: 1003-1012.
- Zimmerman. H. J. "Fuzzy Set Theory—and Its Applications". *KluwerAcademic Press*, Boston, 2nd edition, 1991

Zwietering, M.H., Cuppers, H.G.A.M., de Wit, J.C., van't Riet, K. 1994. 'Evaluation of data transformations and validation of a model for the effect of temperature on bacterial growth'. *Appl. Environ. Microbiol.* 60, 195-203.