الجمهورية الجزائرية الديمقراطية الشعبية

REPEBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

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

**UNIVERSITE BADJI MOKHTAR - ANNABA**
**FACULTE DES SCIENCES DE L'INGENIEUR**
**DEPARTEMENT D'ELECTRONIQUE**

جامعة باجي مختار – عنابة
كلية علوم الهندسة
قسم الإليكترونيك

# MEMOIRE

**Présenté en vue de l'obtention du diplôme de**

# MAGISTER

**Thème**

# *Optimisation des paramètres d'un régulateur PID flou par les algorithmes génétiques*

**Option :** Automatique Industrielle

**Par**
CHAIB Amina

**Directeur de mémoire : S. KERMICHE Maître de conférences U.ANNABA**

**DEVENANT LE JURY**

| | | | |
|---|---|---|---|
| **Président :** | **H. A ABBASSI** | **Professeur** | **U. ANNABA** |
| **Examinateurs :** | **M. LAFIFI** | **Maître de conférences** | **U. ANNABA** |
| | **M. L SAÏDI** | **Maître de conférences** | **U. ANNABA** |
| | **R. HAMDI** | **Maître de conférence** | **U. ANNABA** |

**Année: 2009 / 2010**

# Remerciements

Le plus grand merci à Dieu qui est le seul à guider dans le bon sens durant ma vie et qui a aidés à réalises ce modeste travail.

Je tiens à remercier vivement monsieur **KERMICHE SALEH**, mon directeur de mémoire, pour son esprit scientifique et compréhensif qui ma beaucoup aidé avec ses idées et ses conseils.

Je remercie, le professeur **ABBASSI HADJ AHMED** qui a accepté de présider le jury de soutenance.

Je tiens aussi à remercier les professeur **M. LAFIFI**, **M. L SAÏDI** et **R. HAMDI** pour avoir accepté d'examiner ce travail.

Finalement je tiens à remercier tous les membres de ma famille et en particulier mes chers parents, et un gros merci pour mes amis.

# ملخص:

العمل المقدم في هذا البحث يوضح كيفية تعديل و تحسين المحكم PID. إن فعالية و مردودية المحكم PID تكمن في العدد المختصر للعناصر القاعدية المشكلة له، مما يغنينا عن استعمال المحكمات ذات الرتبة العالية. من جهة أخرى، تعديل المحكم PID يكون بصفة سهلة و سريعة و لا يحتاج إلى معلومات كثيرة عن النظام المراد مراقبته.

تعديل المحكم PID يتم باستعمال عدة طرق. في هذا السياق سيتركز بحثنا المكون من أربعة أجزاء:

- الجزء الأول: مخصص لعرض شامل حول المحكم PID و بالأخص الطرق الكلاسيكية المستعملة في تعديله.
- الجزء الثاني: مخصص لشرح كيفية تعديل المحكم PID بواسطة المحكم الغامض.
- الجزء الثالث: مخصص لعرض كيفية استخدام الانقسامات الوراثية التطورية في تعديل و تحسين الهيكل العام للقواعد المشكلة للمحكم الغامض.
- الجزء الرابع: مخصص لتطبيق التعديل الوراثي على المحكم الغامض PID .



المفاتيح: المحكم PID ، المحكم الغامض، الانقسامات الوراثية التطورية.

## Abstract :

This work considers the adjustment of proportional-integral-derivative (PID) controllers. Because of their very simple structures, this kind of controllers are very appreciated, and used abundantly by the industrial community.

The restricted number of the controller parameters often implies moderate performances, compared to those that can be obtained by high order control structures. In compensation, it is highly desired to use simple and fast procedures for the tuning of PID controllers. In practice, the methods using minimal a priori information about the plant are largely preferred.

This thesis falls in this memory, the adjustment of the PID gains is fact by several methods (approved methods, command in fuzzy logic and algorithms genetics).

This memory is structured in four chapters; the first chapter presents a generality on the regulating standard PID and more precisely the different classic synthesis methods. The second chapter is interested in the regulation to basis of the fuzzy logic. The third chapter is dedicated to the usage of algorithms genetics in optimization and the adjustment of the fuzzy controllers. The performances of a fuzzy controller are enhanced by the adjustment of the parameters of basis of the functions of adherences. Finally in the fifth chapter, the results of the application of the genetic algorithms for the optimization of the fuzzy controllers, are exposed and discussed.

**Key words:** PID controllers, fuzzy controller, the fuzzy genetic approach.

## Résumé :

Le travail documenté dans ce mémoire considère la synthèse de régulateur proportionnel intégral dérivé (PID). De par leurs structures simples, ceux-ci sont très appréciés et abondamment utilisés par la communauté industrielle.

Le nombre restreint de paramètres d'ajustement implique des performances souvent plus modérées que celles pouvant potentiellement être obtenues en utilisant un régulateur d'ordre élevé. En contrepartie, la synthèse d'une commande PID doit pouvoir être réalisée de façon simple et rapide. En pratique, les méthodes ne requérant qu'un minimum d'informations du système à commander sont largement favorisées.

C'est dans ce contexte que se situe ce mémoire, l'ajustement des gains PID ce fait par plusieurs méthodes (méthodes classiques, commande en logique floue et les algorithmes génétique).

Ce mémoire est structuré en quatre chapitres, le premier chapitre présente une généralité sur le régulateur PID standard et plus précisément les différentes méthodes de synthèses classiques. Le deuxième chapitre s'intéresse à la régulation à base de la logique floue. Le troisième chapitre est consacré à l'utilisation des algorithmes génétique dans l'optimisation et l'ajustement des contrôleurs flous. Les performances d'un contrôleur flou sont améliorées par l'ajustement des paramètres de base des fonctions d'appartenances. Finalement dans le cinquième chapitre, les résultats de l'application des algorithmes génétiques pour l'optimisation des contrôleurs flous obtenus, sont exposée et discutés.

**Mot clefs :** régulateur PID, contrôleur flou, l'approche génétique flou.

# Liste des tableaux

I.1 Ajustage des gains de régulateurs P, PI et PID selon la 1ère méthode de ZN en BF……6
I.2 Ajustage modifié des gains de régulateurs P, PI et PID selon la 1ère méthode de Z-N..6
I.3 Ajustage des gains de régulateurs P, PI et PID selon la 2ème méthode de ZN en BO….7
I.4 Ajustage des gains de régulateurs P, PI et PID selon la méthode de Z-N fréquentielle.8
I.5 Ajustage des gains de régulateurs P, PI et PID selon la méthode de Chien Hrones Reswick pour le rejet de perturbation ……9
I.6 Ajustage des gains de régulateurs P, PI et PID selon la méthode de Chien- Hrones Reswick pour la réponse au changement de consigne……9
I.7 Ajustage des gains de régulateurs P, PI et PID selon la méthode de Cohn-Coon ……10
I.8 Les paramètres de régulateur PID des 4 modèles ajusté par la méthode de ZN en BF12
I.9 Les paramètres de régulateur PID des 4 modèles ajusté par la méthode ZN en BO…13
I.10 Les paramètres de régulateur PID des 4 modèles ajusté par la méthode de Z-N fréq..14
I.11 Les paramètres de régulateur PID des 4 modèles ajusté par la méthode des relais ….15
I.12 Les paramètres de régulateur PID des 4 modèles ajusté par la méthode de Chien Hrones Reswick ……...16
II.1 Tableau de règles flou ……33

# Liste des figures

I.1 Mise en oscillation d'un système par contre-réaction .....6
I.2 Réponse indicielle du système à régler seul : on mesure $T_u$ et $T_g$ .....7
I.3 Mise en osciation contrôlée d'un système au moyen d'un relais .....8
I.4 Réponse indicielle pour un procédé stable : on mesure le gain statique $G_0$, le retard L et la constante du temps T..... 10
I.5 Les réponses indicielles en boucle fermée des 4 modèles régulés par la méthode de Ziegler-Nichols en boucle fermée..... 13
I.6 Les réponses indicielles en boucle fermée des 4 modèles régulés par la méthode de Ziegler-Nichols en boucle ouverte ..... 14
I.7 Les réponses indicielles en boucle fermée des 4 modèles régulés par la méthode de Ziegler-Nichols fréquentielle ..... 15
I.8 Les réponses indicielles en boucle fermée des 4 modèles régulés par la méthode des Relais ..... 16
I.9 Les réponses indicielles en boucle fermée des 4 modèles régulés par la méthode de Chien-Hrones-Reswick..... 17
II.1 Classification des personnes en trois ensembles selon la logique classique .....22
II.2 Classification des personnes en trois ensembles selon la logique flou .....22
II.3 L'intersection (min), L'union (max) et Le complément .....25
II.4 Système flou .....27
II.5 Procédure de conception de contrôleur flou .....29
II.6 Structure d'un contrôleur flou .....30
II.7 Fonctions d'appartenance des deux entrées avec (err : erreur) et (Δerr : la variation de l'erreur)..... 32
II.8 Fonction d'appartenance de la sortie u .....32
II.9 La surface de décision du contrôleur flou .....34
II.10 Réponses indicielles des premiers et deuxièmes modèles régulés avec un régulateur PID classique (bleu) et par un régulateur PID flou (rouge)..... 34
II.11 Réponses indicielles des troisième et quatrième modèles régulés avec un régulateur 35

III.1 Principe général des algorithmes génétique..... 38
III.2 Opérateur de croisement à 1 point .....41
III.3 Opérateur de croisement à 2 points .....41
III.4 Principe de l'opérateur de mutation .....43
III.5 Deux manières différentes de définir une règle floue linguistique .....44
III.6 Codage du chromosome .....46
III.7 Deux genres d'ajustement de la forme de la fonction d'appartenance .....47
III.8 Opérateur de croisement .....48
III.1 La roue de sélection .....49
III.10 Intervalle de variation pour chaque paramètre de fonction d'appartenance .....49
IV.1 Les meilleures valeurs de fitness obtenus pour les 10 générations .....53
IV.2 Les fonctions d'appartenance ajustée par les algorithmes génétiques pour les deux entrées e(t) et Δe(t) (bleu) et la sortie u(t) (rouge) pour les 10 générations..... 54
IV.3 Les meilleures valeurs de fitness obtenus pour les 50 générations .....55
IV.4 Les fonctions d'appartenance ajustée par les algorithmes génétiques pour les deux entrées e(t) et Δe(t) (bleu) et la sortie u(t) (rouge) pour les 50 générations..... 55
IV.5 Les meilleures valeurs de fitness obtenus pour les 100 générations.....56
IV.6 Les fonctions d'appartenance ajustée par les algorithmes génétiques pour les deux entrées e(t) et Δe(t) (bleu) et la sortie u(t) (rouge) pour les 100 générations..... 56
IV.7 Les meilleures valeurs de fitness obtenus pour les 200 générations.....57
IV.8 Les fonctions d'appartenance ajustée par les algorithmes génétiques pour les deux entrées e(t) et Δe(t) (bleu) et la sortie u(t) (rouge) pour les 200 générations..... 57
IV.9 Réponses indicielle du 1ère et 2ème modèles régulés par un régulateur PID classique (bleu) et PID flou ajusté par un AG pour les 10 générations (rouge)..... 58
IV.10 Réponses indicielle du 3ème et 4ème modèles régulés par un régulateur PID classique (bleu) et PID flou ajusté par un AG pour les 10 générations (rouge)..... 59
IV.11 Réponses indicielles du 1ère et 2ème modèles régulés par un PID classique (bleu) et PID flou ajusté par un AG pour les 50 générations..... 60
IV.12 Réponses indicielle du 3ème et 4ème modèles régulés par un PID classique (bleu) et PID flou ajusté par un AG pour les 50 générations..... 61

IV.13 Réponses indicielles du 1ère et 2ème modèles régulés par un PID classique (bleu) et PID flou ajusté par un AG pour les 100 générations..... 62

IV.14 Réponses indicielles du 3ème et 4ème modèles régulés par un PID classique (bleu) et PID flou ajusté par un AG pour les 100 générations..... 63

VI.15 Réponses indicielles du 1ère et 2ème modèles régulés par un PID classique (bleu) et PID flou ajusté par un AG pour les 200 générations..... 64

VI.16 Réponses indicielles du 3ème et 4ème modèles régulés par un PID classique (bleu) et PID flou ajusté par un AG pour les 200 générations..... 65

# *Sommaire*

***Introduction générale*** ... ... ...

## *Chapitre II* *commande en logique floue*


II.1 Introduction .....20
II.2 Généralité sur la logique floue .....20
II.2.1 Principe fondamental de la logique floue .....21
II.2.2 Théorie des ensembles flous .....23
II.2.2.1 Sous ensemble flou .....23
II.2.2.2 Quelques définitions .....24
II.2.2.3 Opérations sur les ensembles flous .....25
II.2.3 Les règles floues .....26
II.3 Système flou.....27
II.3.1 Fuzzification.....27
II.3.2 Moteur d'inférence.....28
II.3.3 Défuzzification.....28
II.4 Synthèse de régulateur PID par un contrôleur flou.....28
II.4.1 Conception d'un contrôleur flou.....29
II.4.2 Résultats de simulation et discutions.....31
II.5 Conclusion.....35


## *Chapitre III* *les algorithmes génétiques*


III.1 Introduction .....37
III.2 Principe de fonctionnement d'un algorithme génétique .....37
III.3 Structure de l'algorithme génétique .....38
III.4 Optimisation d'un contrôleur flou par un algorithme génétique .....43
III.5 Processus d'ajustement génétique des structures profondes .....43
III.5.1 La modélisation floue .....44
III.5.2 Processus d'ajustement .....45
III.5.3 Composants génétiques .....47
III.6 Interprétabilité dans la méthode d'ajustement proposé .....49
III.1 Conclusion .....50

IV.1 Introduction .....52
IV.2 Synthèse du contrôleur flou par un algorithme génétique .....52
IV.3 Résultats de l'optimisation .....53
IV.4 Analyse des résultats .....66
IV.5 Conclusion .....66

**Conclusion générale** .....68

**Référence bibliographique**.....72

# Introduction Générale

## Introduction générale :

De nos jours, les systèmes physiques animés par une commande automatique sont très répandus dans l'industrie et ils nous côtoient également dans la vie quotidienne. Exprimé de manière laconique, la premier de l'automatique est l'étude et la conception d'algorithmes appropriés, afin que la sortie mesurés d'un processus, suivre une trajectoire désirée.

Malgré un éventail très large de processus existants et malgré les diverses formulations envers les exigences requises, le régulateurs proportionnel intégral dérivé (PID), dont l'architecture relativement dépouillé consiste en seulement trois termes distincts, s'est imposé dans plus de 90% des boucles de contre réaction. La prédominance incontestés de ce type de commande provient, outre la simplicité extrême, des performances qu'il peut offrir aux systèmes en boucle fermée, satisfaisant très souvent les cahiers des charges, si ses paramètres sont choisis judicieusement.

Dans ce travail, nous abordons le problème de synthèse du régulateur PID qui n'est plus qu'un problème de réglage des actions proportionnelle, intégrale et dérivée, c'est-à-dire $K_p$, $T_i$, $T_d$. Le réglage de ces paramètres est un problème quotidien pour l'automaticien : il est alors demandeur de méthodes simples à mettre en œuvre, rapides et suffisamment précises.

Afin de régler les paramètres de régulateur PID par des méthodes classiques, nous utilisons un contrôleur flou. Ce principe de régulation, basé sur des concepts relativement simples, permet de faire intervenir dans le contrôleur des connaissances acquises par un expert humain. L'intérêt principal de cette méthode est qu'elle ne nécessite pas de connaître le fonctionnement du système, mais simplement la façon de le commander. Les connaissances sont exploitées sous une forme linguistique par l'intermédiaire de réglages comme « si *condition* alors *action* ». Une méthode de raisonnement (inférence) utilise ces règles pour définir les commandes qui sont envoyée au régulateur PID.

Lorsque les règles de commande ne sont pas connues précisément, la conception du contrôleur flou peut être assimilée à un problème d'optimisation : comment choisir les paramètres du contrôleur afin d'assurer un fonctionnement optimale selon certain critères? De nombreuses techniques d'optimisation ont été décrites dans la littérature (méthodes linéaire, programmation dynamique…), qui ont trouvé des applications dans la plupart des domaines scientifiques. Pour cela nous avons choisi les algorithmes génétiques afin d'optimiser la structure du contrôleur flou.

Un algorithme génétique est une méthode d'exploration de l'espace des solutions possibles d'un problème exploitant des mécanismes similaires de la sélection naturelle. A chaque itération, l'algorithme génétique évalue un ensemble limité de solutions du problème, et retient les meilleures d'entre elle (phase de sélection). Ces solutions sont combinées (phase de croisement) et éventuellement légèrement modifiées (phase de mutation) avons d'être à nouveau évaluées. A l'issue de ce processus itératif, on aboutit sous certaines conditions à la solution optimale du problème.

Ce travail est organisé comme suit :

- **Introduction générale.**
- **Chapitre I :** nous présentons des généralités sur le régulateur PID et plus précisément les différentes méthode de synthèse des paramètres de ce régulateur avec des résultat de l'application de quelque méthodes.
- **Chapitre II :** il est consacré à la commande floue, nous commençons par énoncer les fondements de la logique floue. Ensuite, nous décrivons la structure générale d'un contrôleur flou et les différents sous ensembles qui le constituent. Enfin en applique cette méthode pour régler les paramètres de PID pour quatre modèles proposés.
- **Chapitre III :** nous décrivons le principe des algorithmes génétique. Nous analysons l'intérêt des différentes phases de traitement : sélection des individus, croisement et mutation. Ensuite, nous avons donné une description de la méthode de synthèse d'un contrôleur flou par algorithme génétique.
- **Chapitre VI :** dans ce chapitre, nous décrivons quatre exemples des application de la méthode de synthèse décrite précédemment. Dans les quatre cas, un algorithme génétique est utilisé pour définir un contrôleur flou assurant le réglage des paramètres d'un régulateur PID. Les résultats de cette application serons exposés et discutés.
- **Conclusion générale.**

# Chapitre I

## I.1 Généralité :

Le régulateur PID (Proportionnel, Intégral, Dérivé) est encore largement utilisé dans le milieu industriel malgré l'émergence d'autres méthodes de régulation. Ce régulateur linéaire est basé sur une structure très simple dont le fonctionnement ne dépend que de trois coefficients, qui sont les gains appliqués sur les signaux proportionnel ($K_p$), intégral ($K_i$) et dérivé ($K_d$). Donc il permet de régler à l'aide de ses trois paramètres les performances (amortissement, temps de réponse) d'une régulation d'un processus modélisé par un deuxième ordre. Nombreux sont les systèmes physiques qui, même en étant complexes, ont un comportement voisin de celui d'un deuxième ordre, dans une certaine échelle de temps. Par conséquent, le régulateur PID est bien adapté à la plupart des processus de type industriel et est relativement robuste par rapport aux variations des paramètres du procédé, quand on n'est pas trop exigeant pour les performances de la boucle fermée par rapport à celles de la boucle ouverte (par exemple, accélération très importante de la réponse ou augmentation très importante de l'amortissement en boucle fermée).

## I.2 Régulateur PID : [1]

Le régulateur PID est un système linéaire du second ordre à une entrée et une sortie, dont la fonction de transfert dans le domaine de Laplace est donnée par :

$$G(p) = K_p + \frac{K_i}{p} + K_d . p \qquad \textbf{I.1}$$

$$G(p) = K_p \left( 1 + \frac{1}{T_i . p} + T_d . p \right)$$

où $K_p$ est le gain proportionnel, $K_i$ le gain intégrant, $K_d$ le gain dérivé, $T_i$ la constante de temps intégrante et, $T_d$ la constante de temps dérivée. Les trois fonctionnalités du terme peuvent être exprimées comme suit:

- L'action proportionnelle P crée un signal de commande u(t) proportionnel au signal d'erreur e(t). Le facteur de proportionnalité $K_p$ est son gain statique. Elle permet de jouer sur la vitesse de réponse du procédé.

- L'action intégrale I crée un signal u(t) qui est l'intégrale du signal d'erreur e(t). Elle permet d'annuler l'erreur statique (écart entre la mesure et la consigne).
- L'action dérivée D crée un signal de commande u(t) qui est la dérivée du signal d'erreur e(t). Elle ne peut être utilisée seule.

## I.3 Méthodes de synthèses du régulateur PID :

Dans la pratique industrielle, on utilise directement des « régulateurs PID », construits par des spécialistes en la matière. Le problème de synthèse du correcteur n'est plus alors qu'un problème de réglage des actions proportionnelle, intégrale et dérivée, c'est-à-dire $K_p$, $T_i$, $T_d$. Le réglage de ces paramètres est un problème quotidien pour l'automaticien : il est alors demandeur de méthodes simples à mettre en œuvre, rapides et suffisamment précises.

### I.3.1 Méthode de Ziegler et Nichols :

Dans toutes les méthodes proposées pour calculer les paramètres de réglage des régulateurs PID, la mieux documentée reste encore celle de J.G. Ziegler et N.B. Nichols, en novembre 1942. Elle est basée sur l'observation de la réponse du processus et la connaissance de la structure du régulateur utilisé. Trois variantes ont été proposées dès cette époque, la première pour un réglage en boucle fermée, l'autre en boucle ouverte et la dernière pour un réglage fréquentiel. [2]

#### I.3.1.1 *Méthode de Ziegler-Nichols en boucle fermée :*

Cette méthode nécessite de boucler le système sur un simple régulateur proportionnel dont on augmente le gain jusqu'à amener le système à osciller de manière permanente (phénomène de pompage) (figure I.1) ; on se trouve ainsi à la limite de stabilité du système. Après avoir relevé le gain critique $K_{cr}$ et la période d'oscillation $T_{cr}$ de la réponse, on peut calculer les paramètres du régulateur choisi à l'aide du tableau I.1.

Les valeurs proposées par Ziegler et Nichols ont été testées dans de très nombreuses situations et il faut souligner qu'ici également elles conduisent à un temps de montée relativement court assorti d'un dépassement élevé.

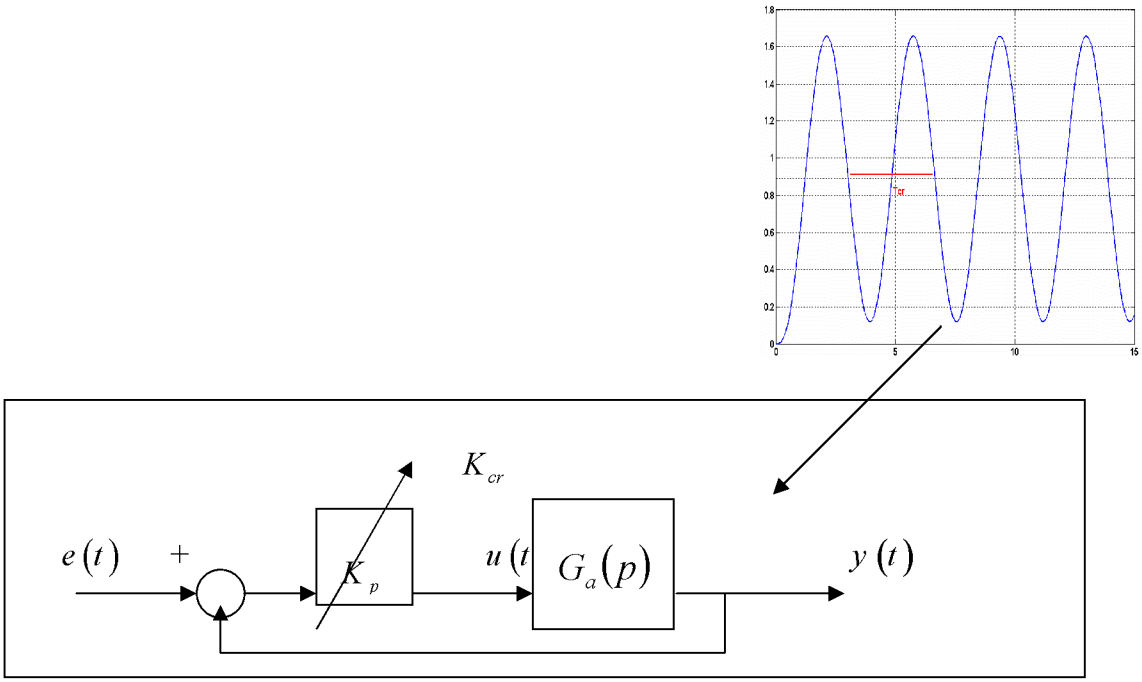



**Figure I.1** : Mise en oscillation d'un système par contre-réaction

| Paramètres / Type de régulateur | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | $0.5*K_{cr}$ | - | - |
| PI | $0.45* K_{cr}$ | $0.83*T_{cr}$ | - |
| PID | $0.6*K_{cr}$ | $0.5*T_{cr}$ | $0.125*T_{cr}$ |

**Tableau I.1** : Ajustage des gains de régulateurs P, PI et PID selon la première méthode de Ziegler-Nichols (méthode en BF).

Cette situation n'étant pas toujours satisfaisante, on est amené à corriger légèrement les coefficients proposés. Une modification possible est proposée par le tableau I-2. Il est important de remarquer que les paramètres $T_i$ et $T_d$ proposés dans les 2 méthodes de Ziegler-Nichols ont été fixés dans un rapport constant égal à 4.

| Paramètres / Type de régulateur | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | **$0.4*K_{cr}$** | - | - |
| PI | **$0.4* K_{cr}$** | **$0.4*T_{cr}$** | - |
| **PID** | **$0.4*K_{cr}$** | **$0.4*T_{cr}$** | **$0.1*T_{cr}$** |

**Tableau I.2** : Ajustage modifié des gains de régulateurs P, PI et PID selon la première méthode de Ziegler-Nichols

### I.3.1.2 *Méthode de Ziegler et Nichols en boucle ouverte :*

La plupart des méthodes de calcul des paramètres de réglage consistent à faire des calculs sur les courbes de réponse du processus en boucle ouverte, suite à l'application d'un échelon (figure I.2). Dans tous les cas, on considère la zone linéaire où la courbe de réponse présente la pente maximum, on trace une droite "épousant" cette zone linéaire et on s'intéresse au point d'intersection de cette droite avec l'axe des abscisses (axe des temps) : on définit ainsi le temps Tu. On définit ensuite le temps Tg, comme étant le temps qu'il faut à la variable contrôlée pour varier de la même amplitude que la sortie du régulateur, ceci à la vitesse de variation maximale (il faut donc reprendre la droite précédemment tracée).Ces deux paramètres permettent alors de définir les paramètres de réglage du régulateur.

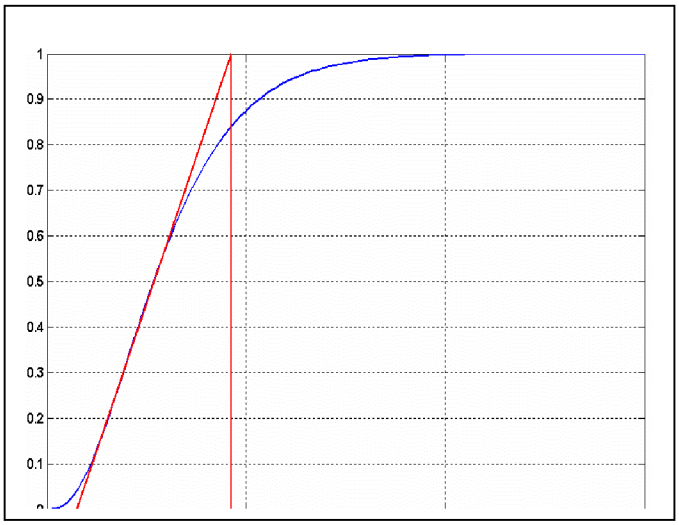



**Figure I.2** : Réponse indicielle du système à régler seul : on mesure les temps Tu et Tg

On peut alors calculer les coefficients du régulateur choisi à l'aide du tableau I.3 :

| Paramètres / Type de régulateur | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | Tg/Tu | - | - |
| PI | 0.9*Tg/Tu | 3.3*Tu | - |
| PID | 1.2*Tg/Tu | 2*Tu | 0.5*Tu |

**Tableau I.3** : Ajustage des gains de régulateurs P, PI et PID selon la méthode de Ziegler-Nichols en boucle ouverte.

### I.3.1.3 *Méthode de Ziegler-Nichols fréquentielle :*

Cette méthode est basée sur la connaissance du point critique du diagramme de Nyquist du procédé. Les paramètres du régulateur PID en fonction des caractéristiques $K_0$ et $T_0$ sont donnés dans le tableau I.4 où $K_0$ et $T_0$ sont respectivement le gain qui amène le système en oscillation et la période des oscillations.

| Paramètres / Type de régulateur | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | $0.5*K_0$ | - | - |
| PI | $0.45*K_0$ | $0.8*T_0$ | - |
| PID | $0.6*K_0$ | $0.5*T_0$ | $0.125*T_0$ |

**Tableau I.4** : Ajustage des gains de régulateurs P, PI et PID selon la méthode de Ziegler-Nichols fréquentielle.

## I.3.2 Méthode standard du relais :

La méthode standard du relais propose de remplacer simplement le régulateur par un élément tout-ou-rien. Ce montage en rétroaction est illustré en figure I.3. [3] $y(t)$

On peut utiliser un relais avec hystérésis [2] [4] ou avec retard [5]. Les deux types de relais donnent des résultats similaires, mais si on l'implémente sur un microprocesseur avec une petite longueur des mots (8 bits), le test du relais avec retard est en principe plus précis [6].

Différentes modifications de la méthode standard du relais ont permis d'appliquer cette méthodologie aux systèmes en boucle fermée dans le but d'en extraire les marges de phase [7] [8] et de gain [1] [7].

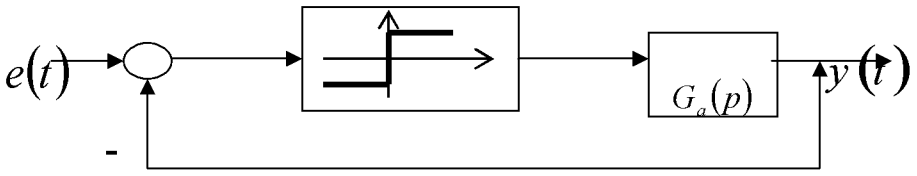



**Figure I.3** : Mise en oscillation contrôlée d'un système asservi au moyen d'un relais.

### I.3.3 Méthode de Chien-Hrones-Reswick :

Cette méthode représente une amélioration de la méthode de Ziegler-Nichols en boucle ouverte. Les paramètres du régulateur sont calculés séparément pour le rejet de perturbation de charge (tableau I.5) et pour la réponse au changement de consigne (tableau I.6). [10]

| Dépassement | 0% | | | 20% | | |
|---|---|---|---|---|---|---|
| Paramètres | $K_p$ | $T_i$ | $T_d$ | $K_p$ | $T_i$ | $T_d$ |
| P | 0.3/a | - | - | 0.7/a | - | - |
| PI | 0.6/a | 4L | - | 0.7/a | 2.3L | - |
| PID | 0.95/a | 2.4L | 0.42L | 1.2/a | 2L | 0.42L |

**Tableau I.5** : Ajustage des gains de régulateurs P, PI et PID selon la méthode de Chien-Hrones-Reswick pour le rejet de perturbations.

| Dépassement | 0% | | | 20% | | |
|---|---|---|---|---|---|---|
| Paramètres | $K_p$ | $T_i$ | $T_d$ | $K_p$ | $T_i$ | $T_d$ |
| P | 0.3/a | - | - | 0.7/a | - | - |
| PI | 0.35/a | 1.2T | - | 0.6/a | T | - |
| PID | 0.6/a | T | 0.5L | 0.95/a | 1.4T | 0.47L |

**Tableau I.6** : Ajustage des gains de régulateurs P, PI et PID selon la méthode de Chien-Hrones-Reswick pour la réponse au changement de consigne.

### I.3.4

## II.3.4 Méthode de Cohen-Coon :

Cette méthode est basée sur le modèle suivant :

$$G(p)=\frac{G_0 e^{-L.p}}{1+T.p} \qquad \textbf{I.2}$$

Elle pourrait être considérée dans la classe des méthodes de placement des pôles, car elle cherche à positionner les pôles dominants de la boucle fermée pour obtenir un rapport de décroissance de d = 1/4. Les paramètres du régulateur PID ont été déduits par des calculs analytiques et numériques et sont exprimés en fonction des caractéristiques du procédé :

$$a=\frac{G_0.L}{T} \quad et \quad \tau=\frac{L}{L+T} \qquad \textbf{I.3}$$

Comme dans le tableau I.7. [9]

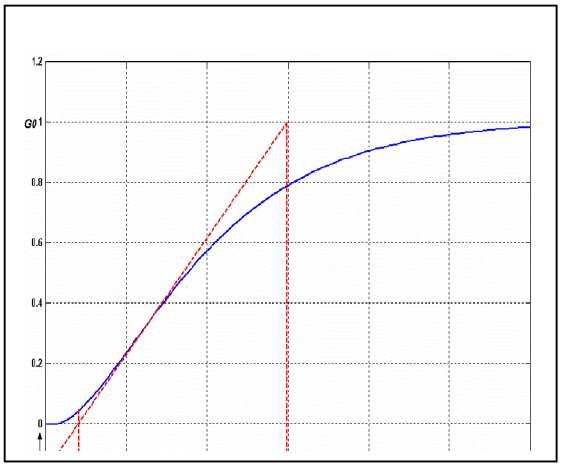



**Figure I.4** : Réponse indicielle pour un procédé stable : on mesure le gain statique $G_0$ le retard L et la constante du temps T.

| Paramètres / Type de régulateur | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| P | 1/a [1+ (0.35τ/1-τ)] | - | - |
| PI | 0.9/a [1+ (0.92τ/1-τ)] | [(3.3-3τ)/1+1.2τ]*L | - |
| PD | 1.24/a [1+ (0.13τ/1-τ)] | - | [0.27-0.36τ/1-0.87τ]*L |

| | | | |
|---|---|---|---|
| PID | 1.35/a[1+ (0.18τ/1-τ)] | [2.5-2τ/1-0.39τ]*L | [0.37-0.37τ/1-0.81τ]*L |

**Tableau I.5** : Ajustage des gains de régulateurs P, PI, PD et PID selon la méthode de Cohen-Coon.

## I.3.5 Méthodes qui simplifient la dynamique du procédé :

Ces méthodes sont basées sur l'idée d'obtenir la fonction de transfert du régulateur C(p) directement par calcul, à partir de la spécification de la fonction de transfert désirée en boucle fermée $H_{BF}$ et en connaissant un modèle du procédé G (p).

### I.3.5.1 *Méthode de Halman :[11]*

Pour les systèmes avec retard, Halman a suggéré de choisir : $H_{BO} = \frac{2}{3} L.p.e^{-L.p}$ **I.4**

Donc on obtient un régulateur PID pour un procédé modélisé par :

$$G(p) = \frac{G_0}{(1+T_1.p)(1+T_2.p)} e^{-L.p} \quad \textbf{I.5}$$

avec :

$$K_p = \frac{2.(T_1+T_2)}{3.G_0.L}, \quad T_i = (T_1+T_2) \quad et \quad T_d = \frac{T_1.T_2}{(T_1+T_2)} \quad \textbf{I.6}$$

### I.3.5.2 Méthode IMC:

La méthode IMC (Internal Model Control) adaptée aux régulateurs PID, permet seulement lorsque le processus est décrit par un modèle de premier ordre, de concevoir un régulateur garantissant une stabilité robuste à la boucle fermée. [12] [13]

## I.3.6 Placement de pôle :

Cette méthode calcule les paramètres du régulateur PID à partir de la spécification des pôles désirés en boucle fermée et en connaissant le modèle du procédé.

### *I.3.6.1 Optimum du module et optimum symétrique :*

Les approches par **optimum du module** (OM) « Betrags Optimum » et **optimum symétrique** (OS) « Symmetric Optimum » sont deux méthodes de calcul des paramètres du

PID qui ont comme but de calibrer la fonction de transfert de la boucle ouverte [14]. Ces méthodes se prêtent particulièrement bien au contexte industriel : elles sont par exemple largement utilisées dans le domaine des asservissements électriques.

#### *I.3.6.2 Méthode KLV (Kessler-Landau-Voda) :*

La synthèse de toutes les phases (théorie de base, calculs, hypothèses amplificatrices, domaine de validité et évaluation expérimentale) de la méthode de KLV est présenté dans [15]. Le principe de départ de la méthode KLV pour l'auto calibrage des PID est inspiré de la méthode d'optimum symétrique (OS).

### *I.3.7* Méthode de Kappa-Tau :

La méthode de Kappa-Tau, est de nature empirique, propose de simples lois d'ajustement qui attribuent à un système inconnu. [1]

## I.4 Résultat de simulation :

Afin d'illustrer l'importance de la connaissance du comportement du régulateur PID et de méthodes d'ajustement des paramètres $K_p$, $T_i$ et $T_d$, on présente ci-après quelques exemples de systèmes de régulation et l'allure de la réponse indicielle de ces systèmes trouvée par les méthodes de ziegler-Nichols, la méthode des relais et la méthode de Chien-Hrones-Reswick.

**Modèle 1:** $$G_1(p)=\frac{1}{(1+10.p)^2}e^{-0.8.p}$$

**Modèle 2:** $$G_2(p)=\frac{1}{(1+10.p)^2}$$

**Modèle 3:** $$G_3(p)=\frac{1-0.5.p}{(1+p)^3}$$

**Modèle 4:** $$G_4(p)=\frac{1}{(1+p)(1+\alpha.p)(1+\alpha^2.p)(1+\alpha^3.p)},\quad où\quad \alpha=0.2$$

### I.4.1 *Application de la méthode de Ziegler-NIchols en boucle fermée :*

Pour chaque exemple, nous avons ajusté les paramètres du régulateur PID par la méthode de Z-N en boucle fermée et nous présentons dans un tableau ces derniers (tableau I.8).

| paramètres / modèles | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| Modèle 1 | 15 | 6.145 | 1.536 |
| Modèle 2 | 4.8 | 1.875 | 0.4688 |
| Modèle 3 | 1.44 | 2.86 | 0.715 |
| Modèle 4 | 5.4 | 0.479 | 0.11 |

**Tableau I.8** : les paramètres de régulateur PID des 4 modèles ajusté par la méthode de Zigler-Nichols en boucle fermée.

Les réponses indicielles des systèmes en boucle fermées avec les paramètres du régulateur PID calculé d'après la méthode de Ziegler-Nichols en boucle fermée sont représentées ci-dessous (figure I.5) :

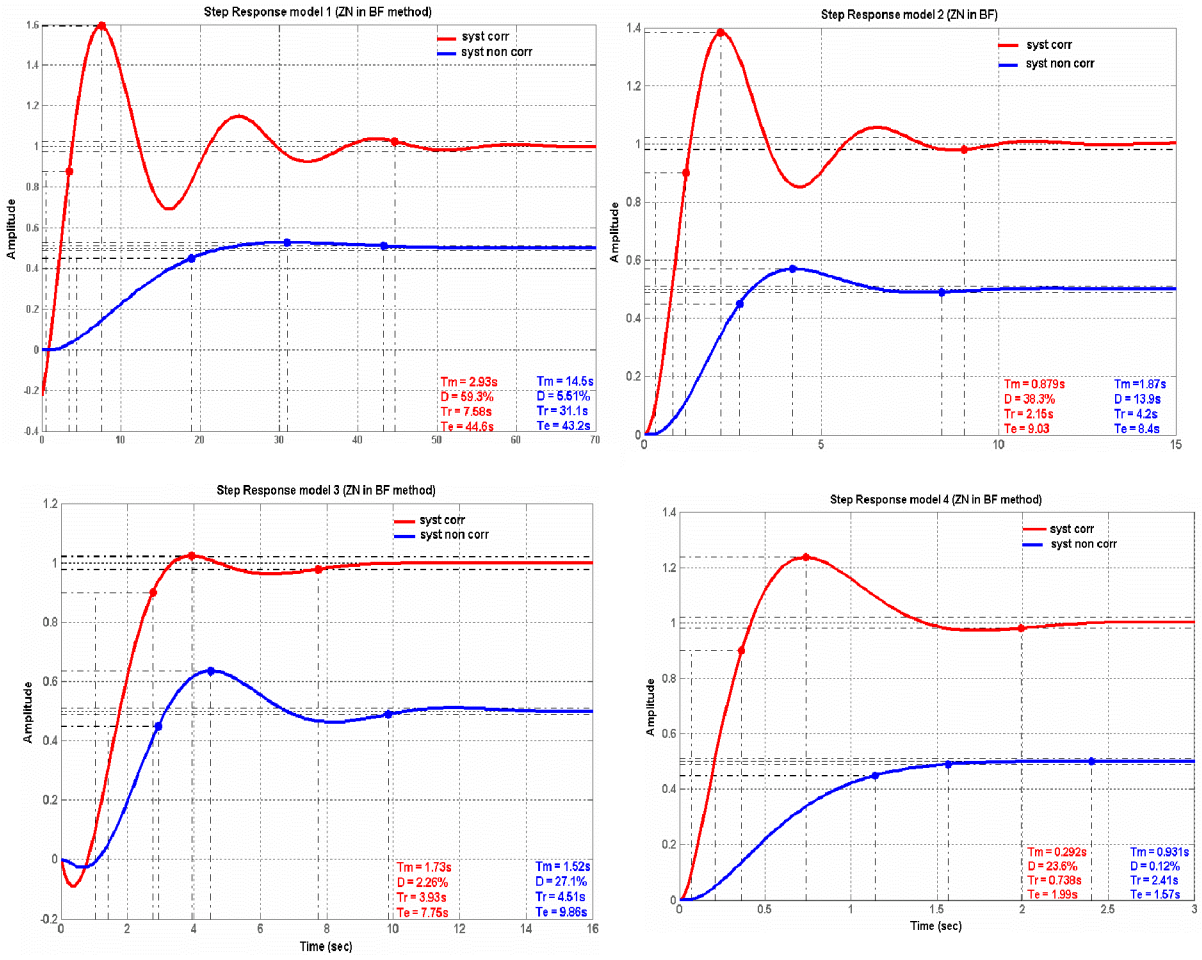



**Figure I.5** : Les réponses indicielles en boucle fermée des quatre modèles régulés par la méthode Ziegler-Nichols en Boucle Fermée

### I.4.2 *Application de la méthode de Ziegler et Nichols en boucle ouverte :*

On applique la méthode de Z-N en boucle ouverte et on détermine les paramètres du régulateur PID (tableau I.9

| Paramètres / modèles | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| Modèle 1 | 8.30 | 7.6 | 1.9 |
| Modèle 2 | 1.12 | 2.13 | 0.28 |
| Modèle 3 | 1 | 2.5 | 0.4 |
| Modèle 4 | 10.58 | 0.338 | 0.084 |

**Tableau I.9** : les paramètres de régulateur PID des 4 modèles ajusté par la méthode de Z-N en boucle ouverte.

Les réponses indicielles des systèmes en boucle fermées avec les paramètres du régulateur PID calculé d'après la méthode de Ziegler-Nichols en boucle ouverte sont représentées ci-dessous (figure I.6) :

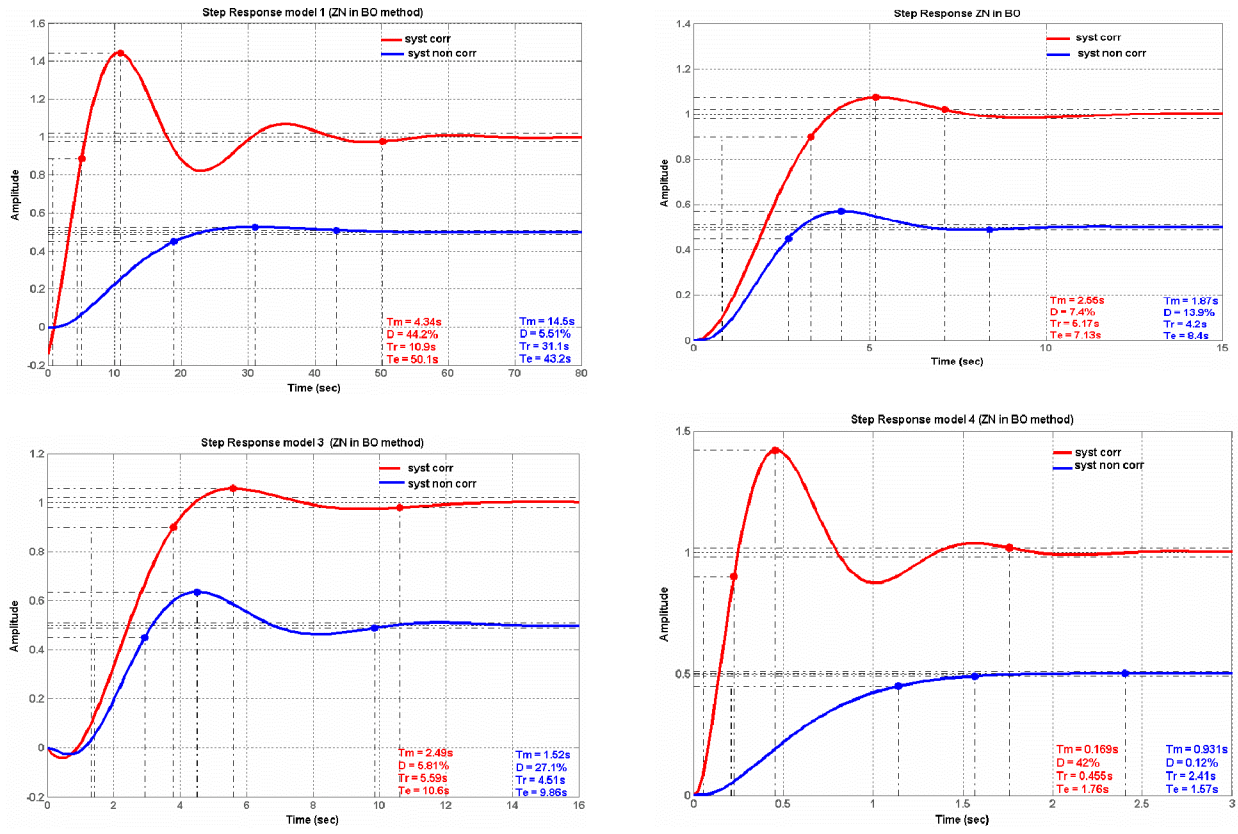



**Figure I.6** : Les réponses indicielles en boucle fermée des quatre modèles régulés par la méthode Ziegler-Nichols en Boucle Ouverte.

On applique la méthode de Ziegler et Nichols fréquentielle et on détermine les paramètres du régulateur PID (tableau I.10) :

| paramètres / modèles | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| Modèle 1 | 15 | 6.155 | 1.5388 |
| Modèle 2 | 4.8 | 1.8150 | 0.4537 |
| Modèle 3 | 1.44 | 2.66 | 0.665 |
| Modèle 4 | 5.4 | 0.28 | 0.07 |

**Tableau I.10** : Les paramètres de régulateur PID des 4 modèles ajusté par la méthode de Ziegler –Nichols fréquentielle.

Les réponses indicielles des systèmes en boucle fermées avec les paramètres du régulateur PID calculé d'après la méthode de Ziegler et NIchols fréquentielle sont représentées ci-dessous (figure I.7)

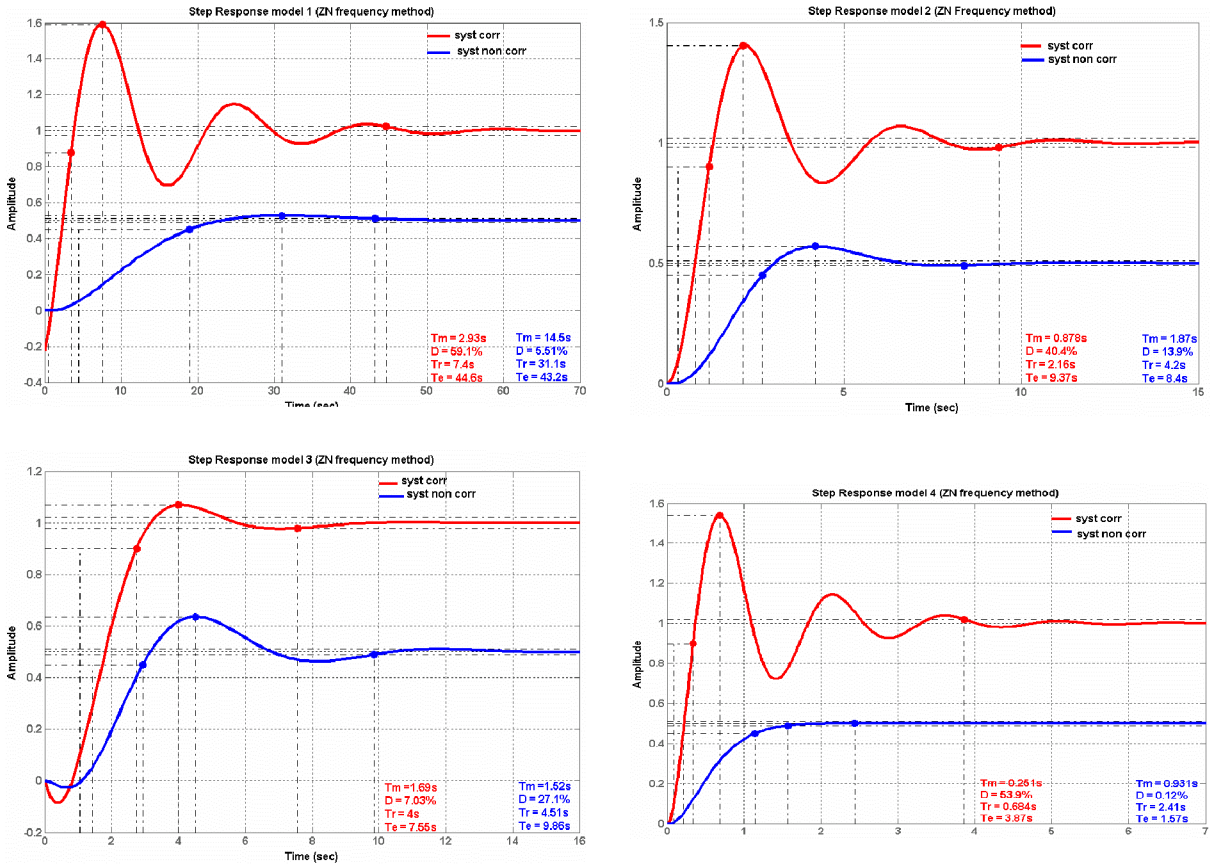



**Figure I.7** : Les réponses indicielles en boucle fermée des quatre modèles régulés par la méthode de Ziegler et Nichols Fréquentielles.

### I.4.4 *Application de la méthode des relais :*

On applique la méthode des relais et on détermine les paramètres du régulateur PID (tableau I.11) :

| Parameters / modèles | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| Modèle 1 | 9 | 9.725 | 2.431 |
| Modèle 2 | 3.084 | 1.85 | 0.462 |
| Modèle 3 | 1.59 | 2 | 0.5 |
| Modèle 4 | 7.64 | 0.295 | 0.073 |

**Tableau I.11** : Les paramètres de régulateur PID des 4 modèles ajusté par la méthode des relais.

Les réponses indicielles des systèmes en boucle fermées avec les paramètres du régulateur PID calculé d'après la méthode des relais sont représentées ci-dessous (figure I.8) :

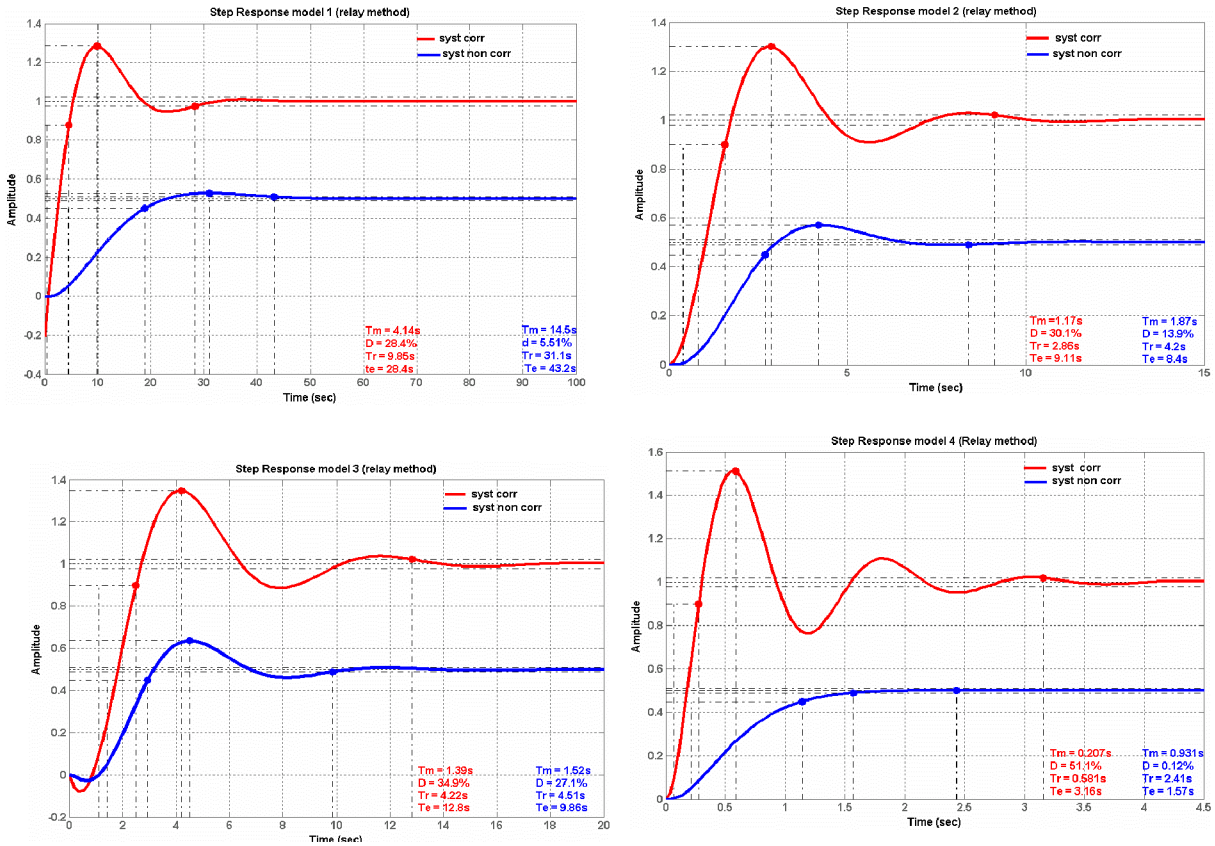



**Figure I.8** : Les réponses indicielles en boucle fermée des quatre modèles régulés par la méthode des relais

### I.4.5 Application de la méthode de Chien-Hrones-Reswick :

On applique la méthode de Chien-Hrones-Reswick et on détermine les paramètres du régulateur PID (tableau I.12) :

| paramètres / modèles | $K_p$ | $T_i$ | $T_d$ |
|---|---|---|---|
| Modèle 1 | 4.16 | 26.3 | 1.9 |
| Modèle 2 | 3.92 | 4.8 | 0.37 |
| Modèle 3 | 2.14 | 3.89 | 0.55 |
| Modèle 4 | 5.4 | 1.491 | 0.084 |

**Tableau I.12** : Les paramètres de régulateur PID des 4 modèles ajusté par la méthode de Chien-Hrones-Reswick.

Les réponses indicielles des systèmes en boucle fermées avec les paramètres du régulateur PID calculé d'après la méthode de Ziegler et Nichols fréquentielle sont représentées ci-dessous (figure I.9)

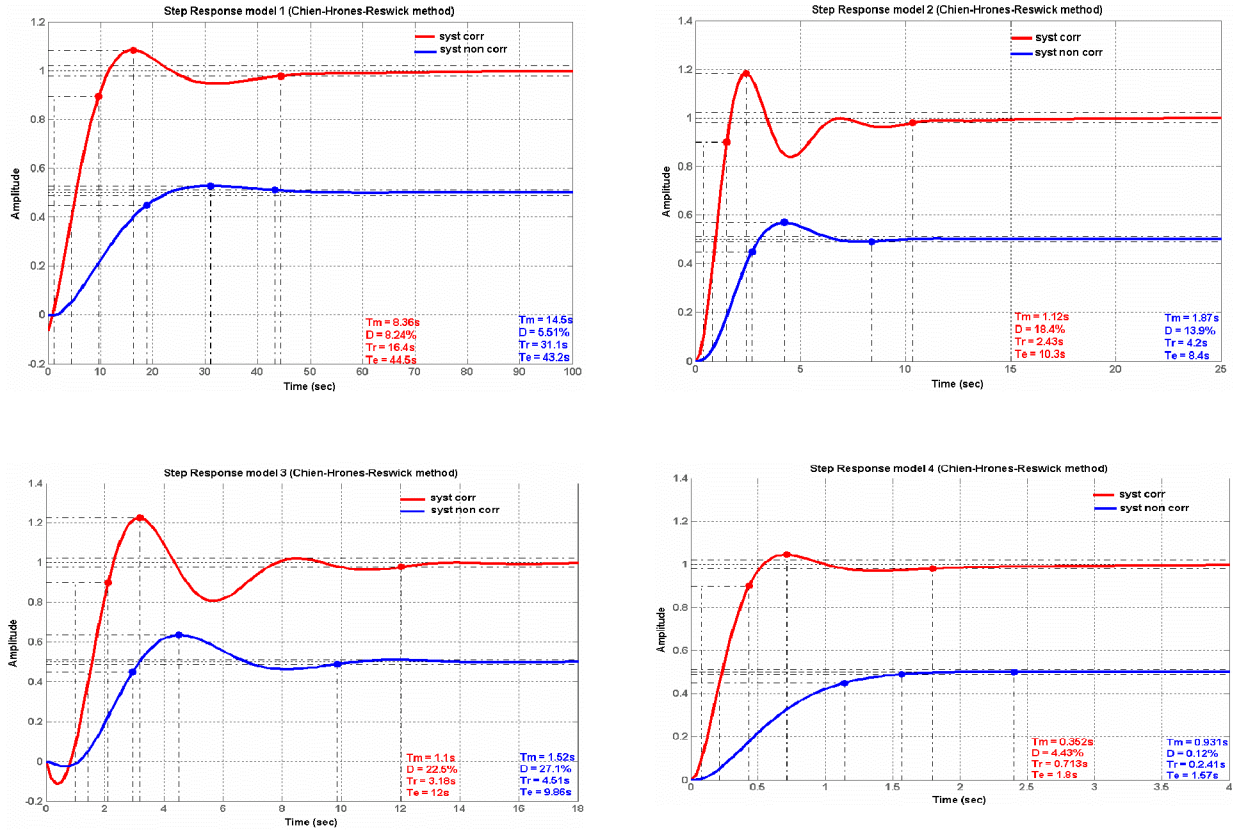



**Figure I.9** : Les réponses indicielles en boucle fermée des quatre modèles ajustés par la méthode de Chien-Hrones-Reswick.

## I.5 Analyse des résultats :

- Tous d'abord, d'après les résultats obtenus, on peut constater que les paramètres de régulateur PID influencent les réponses des systèmes de la manière suivante :
  - ✓ $K_p$ : Lorsque $K_p$ augmente, le temps de montée est plus court mais il y a un dépassement plus important. Le temps d'établissement varie peu et l'erreur statique se trouve améliorée et vis vers ça.
  - ✓ $T_i$ : Lorsque $1/T_i$ augmente, le temps de montée est plus court mais il y a un dépassement plus important. Le temps d'établissement au régime stationnaire s'allonge mais dans ce cas on assure une erreur statique nulle.
  - ✓ $T_d$ : Lorsque $T_d$ augmente, le temps de montée change peu mais le dépassement diminue. Le temps d'établissement au régime stationnaire est meilleur. Pas d'influences sur l'erreur statique.
- Les réponses indicielles des systèmes sont assez proches pour les cinq méthodes qu'on a appliquées malgré des paramètres légèrement différents.
- L'analyse du système avec un PID est très simple mais sa conception peut être délicate, car il n'existe pas de manière unique pour résoudre ce problème. Il faut trouver des compromis, le régulateur idéal n'existe pas.

## I.6 Conclusion :

Ce chapitre permet d'expliquer pourquoi les régulateurs PID sont aussi employés :

- En premier lieu, les paramètres de réglages sont peu nombreux.
- En deuxième lieu, il faut signaler qu'il n'est pas toujours possible d'obtenir un modèle du procédé.
- La simplicité et la flexibilité de l'ajustement des PID sont à juste titre des paramètres importants
- Enfin, l'utilisation de régulateurs PID conduit à un comportement dit robuste. Cela signifie que le comportement reste satisfaisant même si des erreurs sont faites sur le modèle du procédé, ou encore que le procédé évolue en cours de fonctionnement pour diverses raisons (non linéarités, vieillissement…).

# Chapitre II

## II.1 Introduction :

Contrairement à l'automatique linéaire, l'automatique non linéaire ne dispose pas de solutions universelles ni pour l'analyse des systèmes ni pour la conception de leurs contrôleurs. L'analyse et la commande de ces systèmes ne sont pas, toujours, des tâches faciles. La plupart des travaux dans la littérature proposent des approches qui sont, généralement, limitées à des formes bien particulières de systèmes [16] [17]. De plus, les performances assurées sont, souvent, au prix de la complexité du schéma de commande et du développement théorique utilisé.

La plupart des approches de commande non linéaires exigent la disponibilité d'un modèle mathématique du système. Les performances assurées seront directement liées à l'exactitude du modèle utilisé. Pour résoudre ces problèmes, l'utilisation des contrôleurs basés sur l'expertise humaine peut être une alternative. Ils présentent l'avantage de tolérer l'incertitude du modèle et compensent son effet. Parmi ces approches, on trouve la commande par logique floue qui permet la commande des systèmes dont on dispose d'informations linguistiques qui peuvent porter aussi bien sur le modèle que sur la commande [18] [19]. Les incertitudes ainsi que les non linéarités négligées lors de la modélisation mathématique du processus peuvent être aussi compensées par les contrôleurs flous [20]. Ces contrôleurs ont connu beaucoup de succès et devenus un sujet dominant dans le domaine de la recherche des systèmes intelligents [21] [22] [23].

Un contrôleur flou peut avoir plusieurs structures, parmi lesquelles ceux qui sont similaires aux contrôleurs classiques. L'analogie existante entre les deux structures permet d'exploiter le support théorique des approches classiques, et de trouver des méthodes, plus ou moins, systématiques pour la conception du contrôleur flou et le réglage de ses paramètres.

## II.2 Généralité sur la logique floue :

Les bases théoriques de la logique floue (fuzzy Logic en anglais) ont été établies en 1965 par le professeur Lotfi A.Zadeh de l'université de Berkeley en Californie [24] [25]. La caractéristique principale de la logique floue est qu'elle est capable d'exprimer les données linguistiquement. Elle permet de décrire un système par une base de règles qui est une image de raisonnement humain. Par ailleurs, elle fournit des outils mathématiques pour traiter les problèmes d'incertitude et d'imprécision. Cette méthode de raisonnement linguistique permet de réduire, dans beaucoup de cas, le temps de calcul et d'améliorer la capacité de description

pour des procédés dont le modèle mathématique n'est pas connu, très compliqué ou quand l'expérience humaine peut jouer un rôle important.

De nos jours, les applications de la logique floue se sont multipliées. Entre autres, on peut citer, les contrôleurs flous, les systèmes experts flous, les systèmes de diagnostic, de classification et de reconnaissance des formes flous…etc.

Le contrôle flou, qui est probablement le domaine d'application le plus important de la logique floue, n'a été développé qu'à partir de 1975 par E Mamdani [26]. Depuis, elle a attiré beaucoup de chercheurs partout dans le monde.

Peu de temps après, en 1977, le danois Ostergaard [27] a appliqué la logique floue à la commande de tubes broyeurs pour la fabrication de ciment. A cette époque, la plupart des études concernant les systèmes de régulation exploitant la logique floue ont été réalisées en Europe [28] [29] [30]. A partir de 1985 environ, ce sont les Japonais [31] [32] [33] [34] qui commencent à utiliser largement la logique floue dans des produits industriels et de consommation pour résoudre des problèmes de réglage et de commande. Notons entre autres les exemples, le contrôle de programme de certaine machine à laver, dans des équipements électroniques et védio, dans le retraitement de eaux usées…etc. Les nombreux succès qu'a remporté le flou au pays du soleil levant, ont séduit l'Europe qui vient de redécouvrir ce qu'elle connaissait déjà.

## II.2.1 Principe fondamental de la logique floue :

Afin de mettre en évidence le principe fondamental de la logique floue, on présente un exemple simple, celui de la classification des personnes en trois ensembles « jeune », « entre deux âges » et « âgés ».

Selon la logique classique (logique de Boole), qui n'admet pour les variables que les deux valeurs 0 et 1, une telle classification pourrait se faire comme le montre la figure II.1. Toutes les personnes âgées de moins de 30 ans sont alors considérées comme appartenant à l'ensemble « jeune » et toutes les personnes âgées de plus de 50 ans comme appartenant à l'ensemble « âgé ». [35]

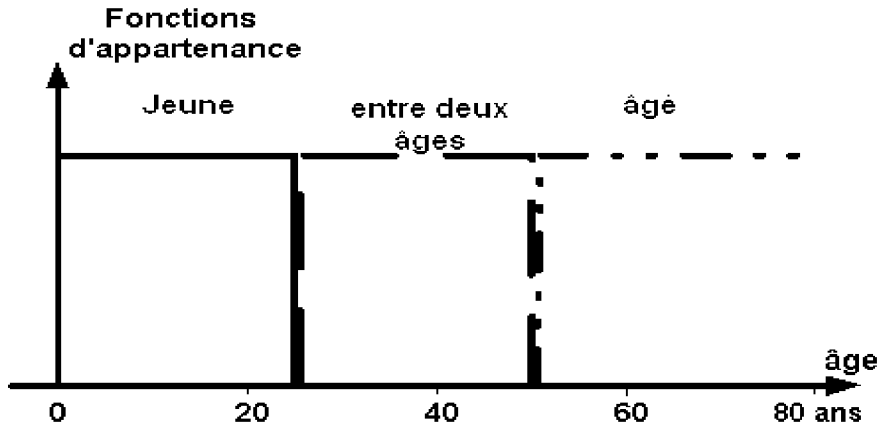



**Figure II.1** : classification des personnes en trois ensembles selon la logique classique

Cependant, une telle logique de classification n'est même pas logique. Pourquoi une personne, lorsqu'elle a eu 50 ans, doit-elle être considérée comme appartenant à l'ensemble « âgé » ? En réalité, un tel passage se fait progressivement et individuellement.

La logique floue, dont les variables peuvent prendre n'importe quelle valeur entre 0 et 1 (on parle alors de fonction d'appartenance $\mu$), qui permet de tenir compte de cette réalité. Les limites ne varient pas soudainement, mais progressivement. La figure II.2 montre une classification possible pour l'exemple précédent, cette fois-ci, à l'aide de la logique floue. Ainsi, une personne de 25 ans appartient à l'ensemble « jeune » avec une valeur $\mu$=0.75 de la fonction d'appartenance et à l'ensemble « entre deux âges » avec $\mu$=0.25. Par contre, une personne de 70 ans appartient avec une valeur $\mu$=1 de la fonction d'appartenance à l'ensemble « âgé ». Evidemment, la forme des fonctions d'appartenance n'est pas rigide et peut dépendre du contexte envisagé, à savoir professionnel, sportif ou autre.

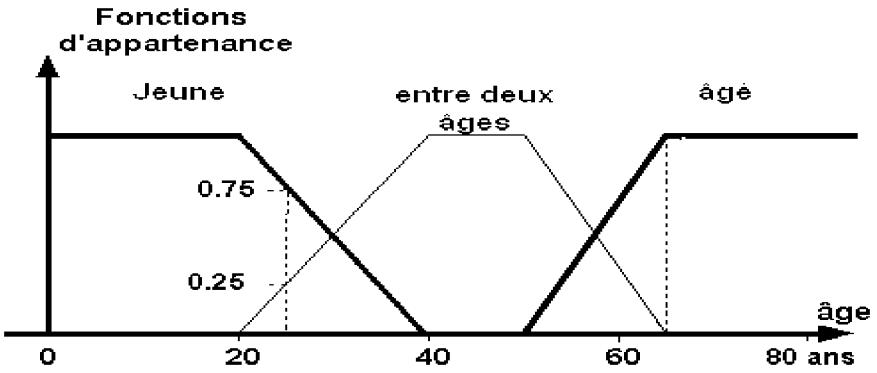



**Figure II.2** : classification des personnes en trois ensembles selon la logique floue

## II.2.2 Théorie des ensembles flous :

La théorie des ensembles flous [36] [37] [38] [39] est une théorie mathématique qui permet de modéliser des notions vagues du langage naturel pour pallier l'inadéquation de la théorie des ensembles classiques dans ce domaine et consiste de faire une interface entre les mondes linguistique et numérique. Elle a été introduite par Lotfi A.Zadeh et permet de définir toutes les opérations classique (conjonction, disjonction…etc).

### II.2.2.1 *Sous ensemble flou :*

En logique binaire, la fonction d'appartenance $\mu_A(x)$ caractérisant l'appartenance d'un élément x, de l'univers de discours X à un ensemble classique A prend uniquement les valeurs 0 et 1.

$$\mu_A(x) \qquad X \rightarrow \{0,1\} \qquad \text{II.1}$$

$$x \rightarrow \begin{cases} \mu_A\{x\}=1 \; si x \in A \\ \mu_A\{x\}=0 \; si x \notin A \end{cases}$$

Dans la théorie des ensembles flous, les valeurs de la fonction d'appartenance $\mu A(x)$ sont comprises dans l'intervalle [0,1] :

$$\mu_A(x) \qquad X \rightarrow \{0,1\} \qquad \text{II.2}$$

$$x \rightarrow \mu_A(x)$$

Un ensemble flou A est défini par un ensemble de paires ordonnées $\{\mu A(x), x\}$ :

$$A = \{(\mu_A(x), x / x \in X)\} \qquad \text{II.3}$$

Avec :

x : est l'élément générique appartenant à l'univers de discours X.

$\mu_A(x)$ : est le degré d'appartenance de cet élément à l'ensemble flou A.

Lorsque l'univers X est discret, l'ensemble flou A est noté :

$$A = \frac{\sum_i \mu_A(X_i)}{X_i} \qquad \textbf{II.4}$$

De façon analogue, si X est continu :

$$A = \int_x \frac{\mu_A(X)}{X} \qquad \textbf{II.5}$$

### II.2.2.2 *Quelques définitions :*

Soient A et B deux ensembles flous et x un élément de l'univers X :

- La hauteur : haut (A) = max { $\mu_A(x)$, $x \in X$ } **II.6**
- Une coupe de niveau $\alpha$ : $\alpha$ -coup (A) = { $x \in X$, $\mu_A(x) \geq \alpha$ } **II.7**
- Le support : supp (A) = { $x \in X$, $\mu_A(x) > 0$ } **II.8**
- Le noyau : nayau (A) = { $x \in X$, $\mu_A(x) = 1$ } **II.9**
- L'inclusion : $A \subset B \Leftrightarrow \{\forall x \in X, \mu_A(x) \leq \mu_B(x)\}$ **II.10**
- L'égalité: $A = B \Leftrightarrow \{\forall x \in X, \mu_A(x) = \mu_B(x)\}$ **II.11**
- Une T-norme est définie de [0,1] × [0,1] dans [0,1] et vérifie les propriétés suivantes :

$T(a,1) = a$

$T(a,b) \leq T(c,d)$ $(\forall a \leq c, b \leq d)$ **II.12**

$T(a,b) = T(b,a)$

$T(T(a,b), c) = T(a, T(b,c))$

- Une S-norme est définie de [0,1] × [0,1] dans [0,1] et vérifie les propriétés suivantes :

$S(a,0) = a$

$S(a,b) \leq S(c,d)$ $(\forall a \leq c, b \leq d)$ **II.13**

$S(a,b) = S(b,a)$

$S(S(a,b), c) = S(a, S(b,c))$

### II.2.2.3 *Opérations sur les ensembles flous :*

Les opérations les plus importantes dans la théorie des ensembles classiques sont l'union, l'intersection et la complémentation. Ces opérations sont également définies en logique floue grâce aux fonctions d'appartenances (figure II.3).

Soient A et B deux sous ensembles flous ayant respectivement $\mu_A$ et $\mu_B$ comme fonction d'appartenance. L'intersection de A et B est un ensemble flou $A \cap B$ tel que :

$$\mu_{A \cap B} = \min (\mu_A, \mu_B) \qquad \textbf{II.14}$$

L'union de A et A est un ensemble flou $A \cup B$ tel que :

$$\mu_{A \cup B} = \max (\mu_A, \mu_B) \qquad \textbf{II.15}$$

Le complément d'un ensemble flou A est un ensemble flou $A^c$ dont la fonction d'apparten ance $\mu_A{}^c$ vérifie:

$$\mu_A{}^c = 1-\mu_A \qquad \textbf{II.16}$$

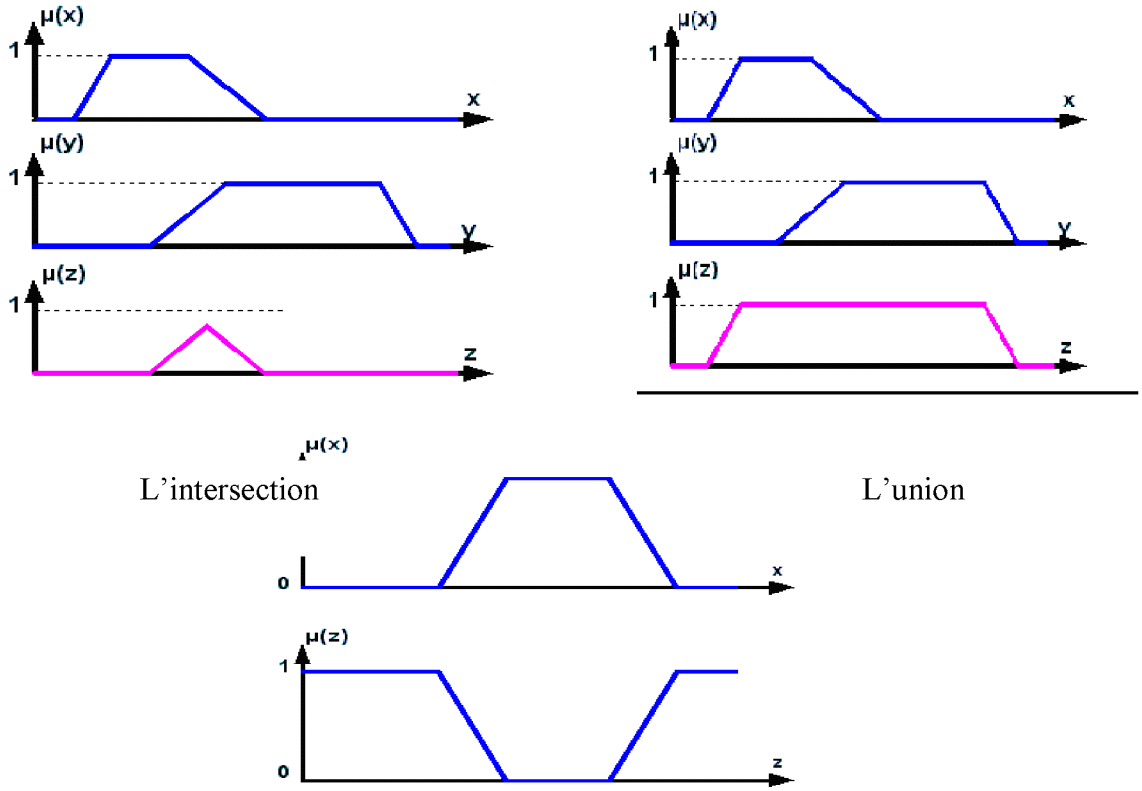



**Figure II.3 :** L'intersection (min), L'union (max) et Le complément

## II.2.3 Les règles floues :

Les règles floues sont des règles exprimées à travers des variables linguistiques, caractérisées par des sous ensembles flous. Les règles floues traduisent une relation entre deux prémisses floues A et B et une conclusion C. Elles s'expriment de la manière suivante :

Si (X est A) et (Y est B) Alors (Z est C)

**II.17**

Où A, B et C sont des sous ensembles flous ayant respectivement les fonctions d'appartenances $\mu_A$, $\mu_B$ et $\mu_C$.

De nombreuses implications floues ont été développées, correspondant chacune à une interprétation différente des règles floues. Pour le contrôle flou, les plus utilisées ont sans aucune doute les implications de Mamdani et Larsen. [40]

- Mamdani

$$\mu_{A\to B}(x,y) = (\mu_A(x), \mu_B(y)) \qquad \textbf{II.18}$$

- Larsen

$$\mu_{A\to B}(x,y) = \mu_A(x).\mu_B(y) \qquad \textbf{II.19}$$

Lorsque la règle contient plusieurs prémisses, de la forme Si $X_1$ est $A_1$ et $X_2$ est $A_2$, Alors Y est B, ou Si $X_1$ est $A_1$ ou $X_2$ est $A_2$, Alors Y est B, les conjonctions sont prises en compte par des opérateurs.

L'opérateur ET et OU s'écrivent de la manière suivante :

- Opérateur ET

$$\mu_A(x) = \mu_A(x_1,x_2) = \min(\mu_{A_1}(x_1), \mu_{A_2}(x_2)) \qquad \textbf{II.20}$$

- Opérateur OU

$$\mu_A(x) = \mu_A(x_1,x_2) = \max(\mu_{A_1}(x_1), \mu_{A_2}(x_2))$$

## II.3 Système flou: II.21

Les systèmes flous permettent d'exploiter et de manipuler efficacement les informations linguistiques émanant de l'expert humain grâce un arsenal théorique important [35] [41]. De plus, le système mis en oeuvre peut être intégré facilement dans une boucle de commande ou d'identification. La structure de base d'un système flou se divise en trois parties principales comme montré dans la figure II.4.

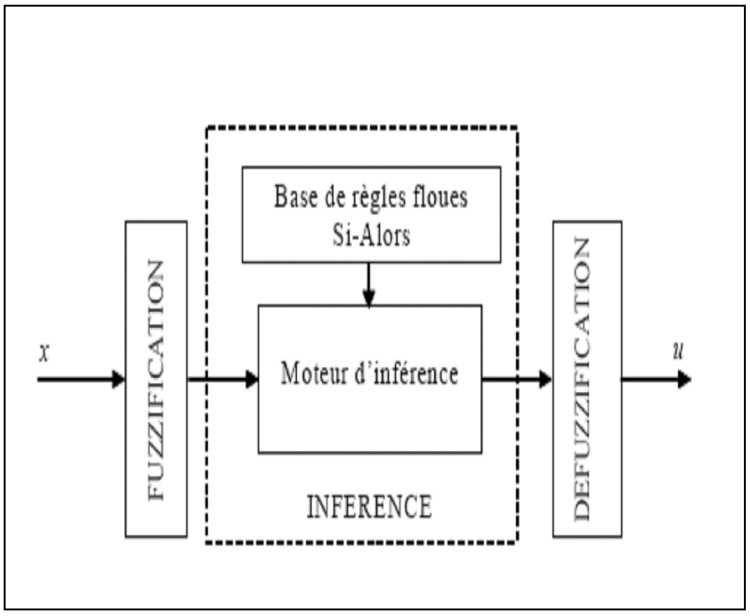



**Figure II.4 :** système flou

Classiquement, un régulateur flou se décompose en trois parties : Fuzzification, raisonnement flou, et défuzzification [42].

### II.3.1 Fuzzification :

La fuzzification est la première étape du traitement flou proprement dite. Elle consiste à définir des fonctions d'appartenances pour les différentes variables linguistiques. Ceci a pour but de la conversion d'une grandeur physique en une linguistique. Il s'agit d'une projection de la variable physique sur les ensembles flous caractérisant cette variable. Cette opération permet d'avoir une mesure précise sur le degré d'appartenance de la variable d'entrée à chaque ensemble flou [43]

### II.3.2 Moteur d'inférence :

L'évaluation des règles floues, autrement appelée inférence floue, consiste à mettre en œuvre les règles floues, établies au préalable, afin de calculer les sorties floues à partir des entrées issues de la fuzzification.

Une règle floue est une procédure qui permet d'établir un lien entre une ou plusieurs prémisses et une conclusion.

Il y a deux principales règles d'inférence utilisées pour le contrôle flou :

- L'inférence de Larsen, basée sur l'implication et l'opérateur produit.
- L'inférence de Mamdani, basée sur l'implication et l'opérateur minimum.

### II.3.3 Défuzzification :

Les sous ensembles flous obtenus en sortie du système d'inférence ne sont pas directement exploitables pour commander un processus. Les sorties d'un système de commande doivent être des grandeurs strictes, c'est-à-dire non floues, et il est alors nécessaire de convertir les sorties floues du système en valeurs réelles strictes.

La transformation en valeurs strictes et exploitables, appelée défuzzification, est effectuée de différentes manières. Parmi les méthodes publiées dans la littérature [35] [41], on peut citer :

- Le centre de gravité.
- La méthode de la hauteur.
- La méthode de la hauteur modifiée.
- La méthode de la valeur maximum.
- La méthode de la moyenne des centres.
- Bissecteur de surface

## II.4 Synthèse de régulateur PID par un contrôleur flou :

Après avoir optimiser les paramètres d'un régulateur PID par les méthodes classiques dans le premier chapitre en s'intéresse dans cette section de régler ces paramètres par un contrôleur flou.

## II.4.1 Conception d'un contrôleur flou :

La procédure générale de conception de contrôleurs flous [35] comporte les étapes montrées sur la figure II.5.

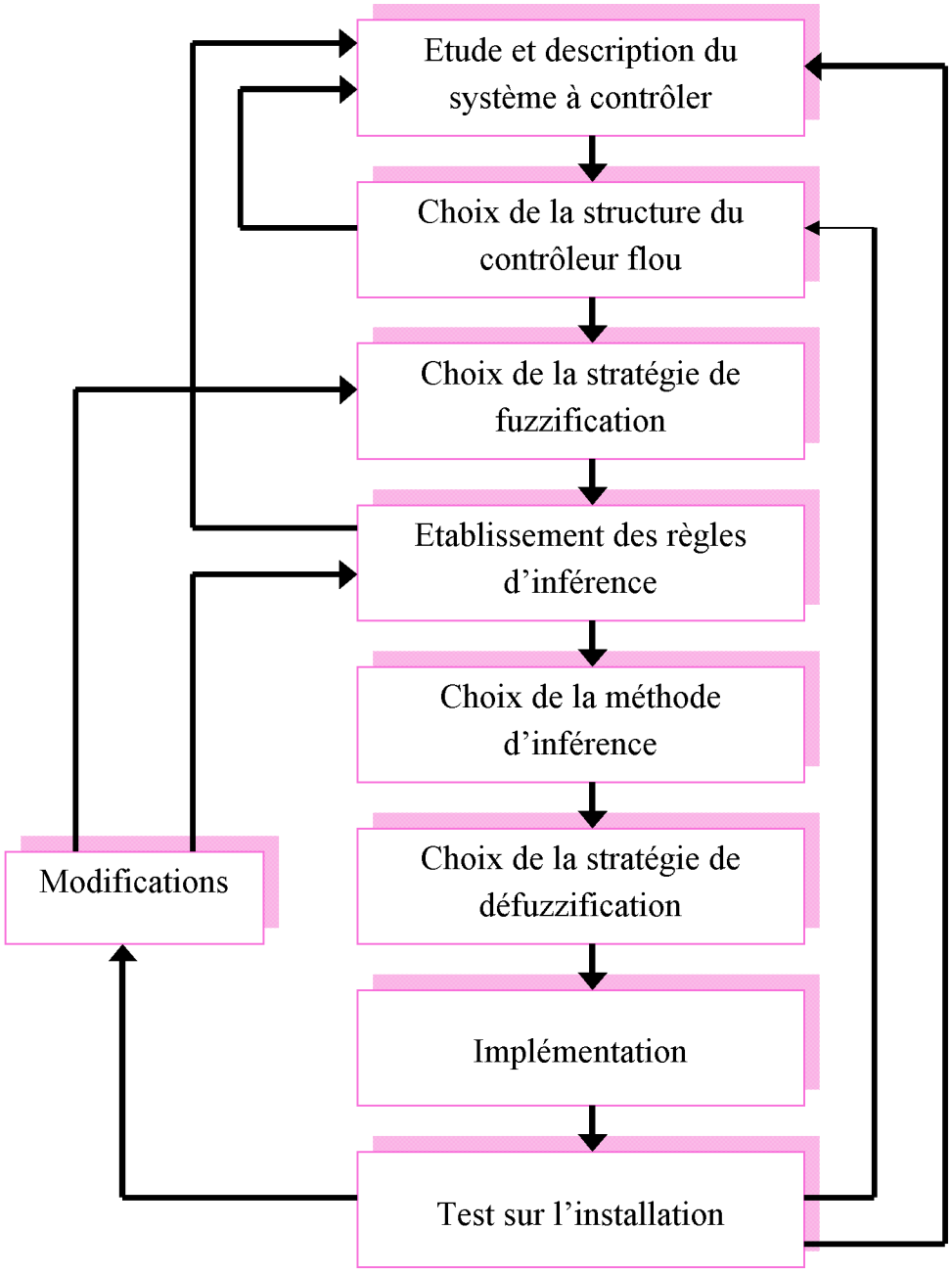



**Figure II.5 :** procédure de conception de contrôleur flou

Cette procédure de conception de régulateurs flous présente l'avantage qu'il ne faut pas établir un modèle pour le système à régler.

➥ ***Etape 1 : <u>Choix de la structure du contrôleur :</u>***

Nous avons choisi la structure du contrôleur proposée par Mamdani, qui est un contrôleur à deux entrées : l'erreur (err) et la variation de l'erreur (Δerr). La sortie du contrôleur est la commande du processus (figure II.6)

L'usage d'un système d'inférence flou peut être adopté pour déterminer la valeur du poids du signal de commande pendant le régime transitoire pour diminuer le temps de montée et au même réduire le dépassement, selon la valeur actuelle de l'erreur du système (e(t)) et sa dérivé (Δe(t)) .Le schéma du contrôle total est montré dans la figure (II.6).

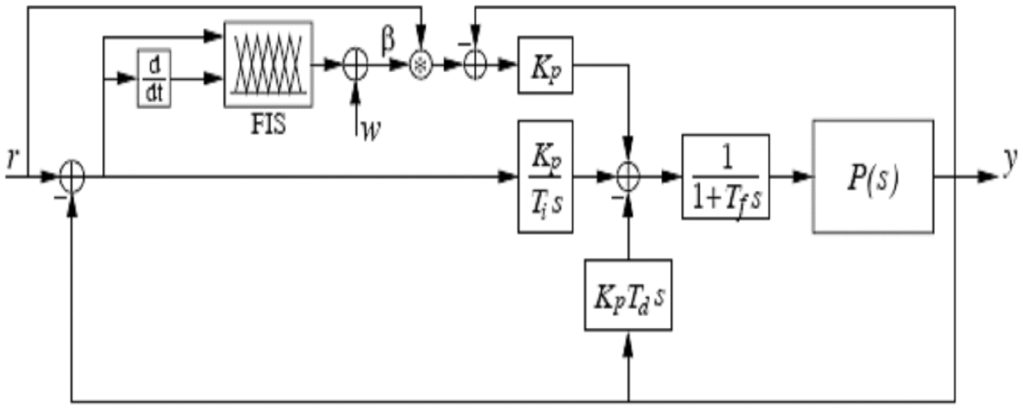



**Figure II.6:** structure d'un contrôleur flou proposée

***Etape 2 : <u>Choix de la stratégie de fuzzification :</u>***

Il s'agit du choix des formes d'appartenance, et leur répartition sur l'univers de discours. Dans notre cas, nous avons choisi de travailler sur des univers de discours normalisé [-1, 1], partitionnés en cinq classes (sous ensembles flous) pour les entrées et neuf classes pour la sortie. La forme la plus utilisée en commande floue et la forme triangulaire, pour cela nous utilisons des fonctions d'appartenance triangulaires.

➥ ***Etape 3 : <u>Etablissement des règles d'inférence :</u>***

Sur la base de la description du système à régler avec des variables linguistiques et de la définition des fonctions d'appartenance pour les variables d'entrée et de sortie, on peut établir les règles d'inférence. Nous avons utilisé une base de règle que l'on trouve dans la plupart des applications.

➥ ***Etape 4 : <u>Choix de la méthode d'inférence :</u>***

Les opérateurs min et max tels qu'ils sont définis par Zadeh sont utilisés pour inférer les règles.

➥ Et***ape 5 : <u>choix de la stratégie de défuzzification</u> :***

La méthode la plus utilisée en commande floue est celle du centre de gravité, mais nous avons opté pour la méthode du bissecteur qui consiste à calculer la valeur de l'abscisse séparant verticalement la surface en deux parts égales.

## II.4.2 <u>Résultat de simulation et discutions :</u>

Dans cette partie, nous allons tester les performances du contrôleur flou sur un régulateur PID pour plusieurs systèmes plus une étude comparative avec le PID classique.

**Modèle 1:** $G_1(p) = \dfrac{1}{(1+10.p)^2} e^{-0.8.p}$

**Modèle 2:** $G_2(p) = \dfrac{1}{(1+p)^3}$

**Modèle 3:** $G_3(p) = \dfrac{1-0.5.p}{(1+p)^3}$

**Modèle 4:** $G_4(p) = \dfrac{1}{(1+p)(1+\alpha.p)(1+\alpha^2.p)(1+\alpha^3.p)}$, *où* $\alpha = 0.2$

Pour chaque exemple, nous avons optimisé les paramètres du régulateur PID avec un contrôleur flou. Pour comparer les performances, nous représentons les réponses des deux couples (PID + système) et (PID flou + système) pour un même signal d'entré constitué un échelon passant de 0 à 1.

Les figures (II.7) et (II.8) présentent les fonctions d'appartenances des entrées et de la sortie du contrôleur flou. Quelques règles floues générées à partir des couples d'entrée-sortie désirés sont données sur le tableau (II.1). La surface de décision (contrôle) du contrôleur est donnée sur la figure (II.9), elle représente la non linéarité et la souplesse de la commande floue.

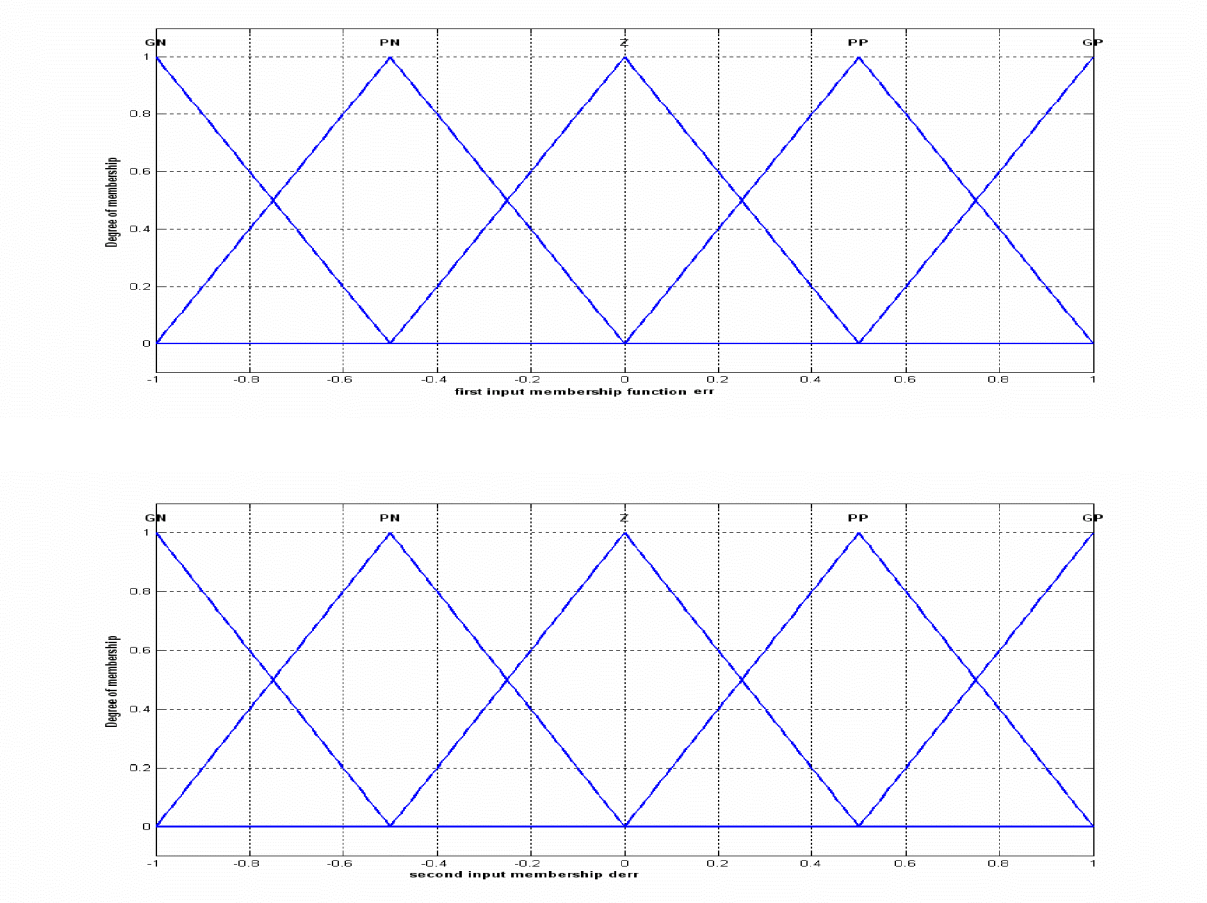



**Figure II.7 :** fonctions d'appartenance des deux entrées avec (err : erreur) et (Δerr : la variation de l'erreur)

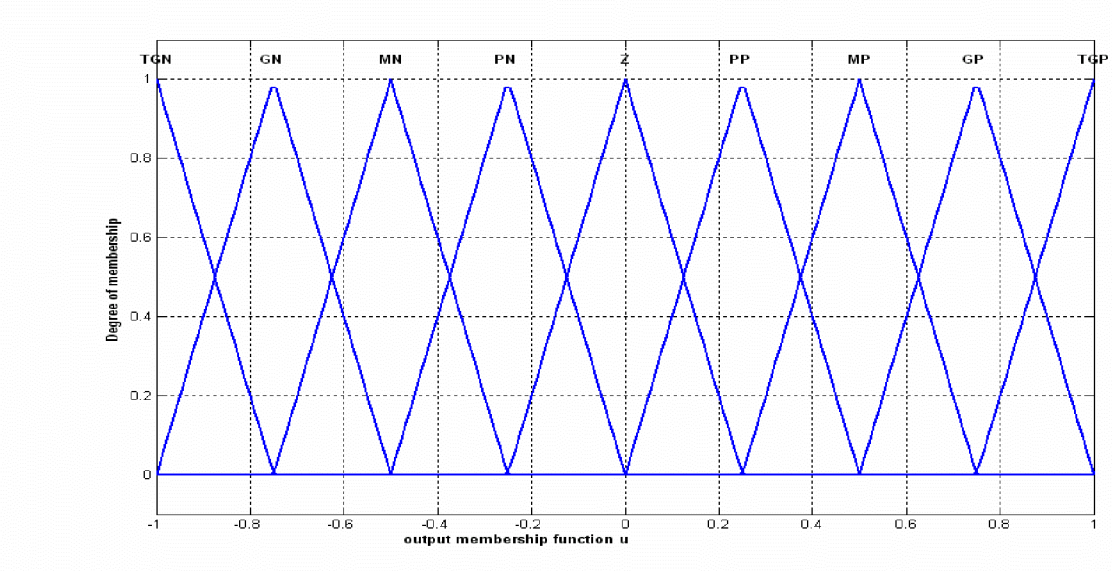



**Figure II.8 :** fonction d'appartenance de la sortie u

| La sortie (u) | | La variation de l'erreur (de) | | | | |
|---|---|---|---|---|---|---|
| L'erreur (e) | | GN | PN | Z | PP | GP |
| | GN | **TGN** | **GN** | **MN** | **PN** | **Z** |
| | PN | **GN** | **MN** | **PN** | **Z** | **PP** |
| | Z | **MN** | **PN** | **Z** | **PP** | **PM** |
| | PP | **PN** | **Z** | **PP** | **MP** | **GP** |
| | GP | **Z** | **PP** | **MP** | **GP** | **TGP** |

Tableau II.1 : tableau de règles floues

La base de règles consiste en (25) règles, chaqu' une à la forme suivante :

***R*$_j$ : If** e **is** $E_j$ **and** $\Delta e$ **is** $DE_j$ **Then** u is $U_j$, pour j = 1…25. Où

- ✓ $R_j$ est la $j^{ème}$ règle.
- ✓ $E_j$ est l'ensemble flou de e dans l'univers de discours $U_e \subset R$ pour la $j^{ème}$ règle qui prend les valeurs linguistique suivantes (**GN** : grande négative, **PN** : petite négative, **Z** : zéros, **PP** : petite positive, **GP** : grande positive).
- ✓ $DE_j$ est l'ensemble flou $\Delta e$ dans l'univers de discours $U_{\Delta e} \subset R$ pour la $j^{ème}$ règle qui prend les valeurs linguistique suivantes (**GN** : grande négative, **PN** : petite négative, **Z** : zéros, **PP** : petite positive, **GP** : grande positive).
- ✓ $U_j$ est l'ensemble flou de e dans l'univers de discours $U_u \subset R$ pour la $j^{ème}$ règle qui prend les valeurs linguistique suivantes (**TGN** : très grande négative, **GN** : grande négative, **MN** : moyenne négative, **PN** : petite négative, **Z** zéros, **PP** : petite positive, **MP** : grande positive, GP : grande positive, **TGP** : très grande positive).

Nous avons noté quelques règles:

$R_1$ : **Si** e est GN **ET** $\Delta e$ est GN **Alors** u est TGN

$R_2$: **Si** e est GN est $\Delta e$ est PN **Alors** u est GN

$R_3$: **Si** e est ET $\Delta e$ est Z Alors u est MN

…

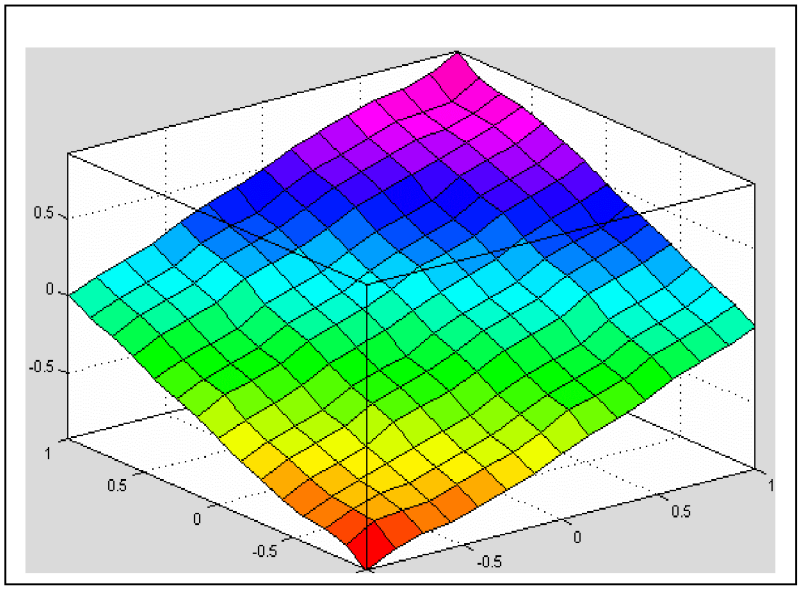



**Figure II.9 :** la surface de décision du contrôleur flou

Les réponses du premier et deuxième modèle sont présentées sur la figure (II.10). Pour le PID classique pour le premier modèle, les paramètres sont fixés aux valeurs suivantes : Kp = 9; Ti = 9.725; Td =2.431. Pour le deuxième modèle les paramètres sont fixés aux : Kp = 1.12; Ti = 2.13; Td = 0.28 On constate que le PID flou dans les deux cas le plus performant par rapport au PID classique.

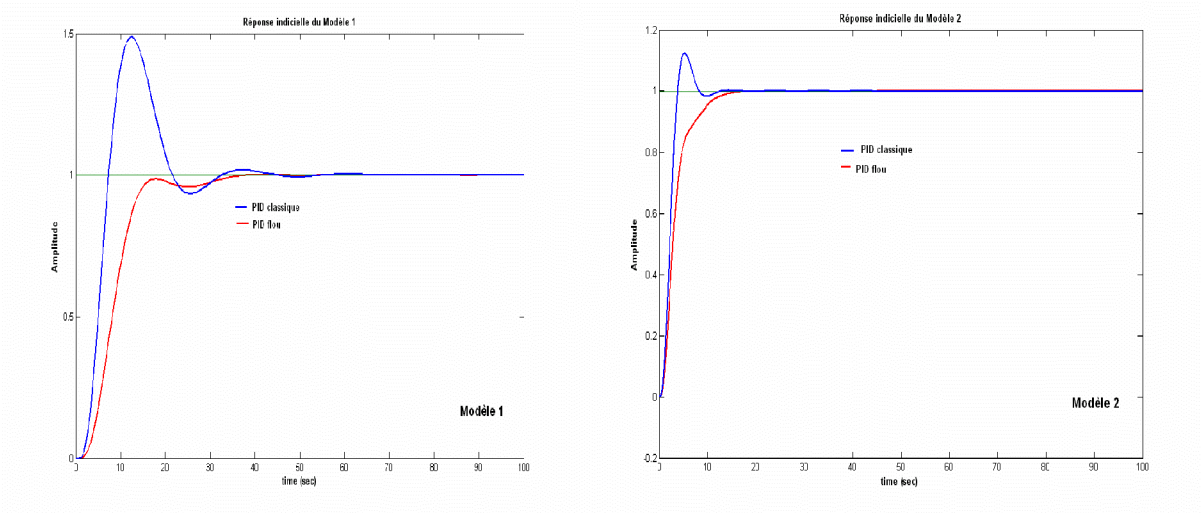



**Figure II.10 :** Réponse indicielle du premier et deuxième modèles réglés avec un régulateur PID classique (bleu) et par un régulateur PID flou (rouge).

Les réponses du troisième et quatrième modèle sont présentées sur la figure (II.11). Pour le PID classique pour le premier modèle, les paramètres sont fixés aux valeurs suivantes: Kp = 1.44; Ti = 2.86; Td = 0.715. Pour le deuxième modèle les paramètres sont fixés aux Kp = 5.4; Ti = 1.491; Td = 0.084. On constate que le PID flou dans les deux cas le plus performant par rapport au PID classique.

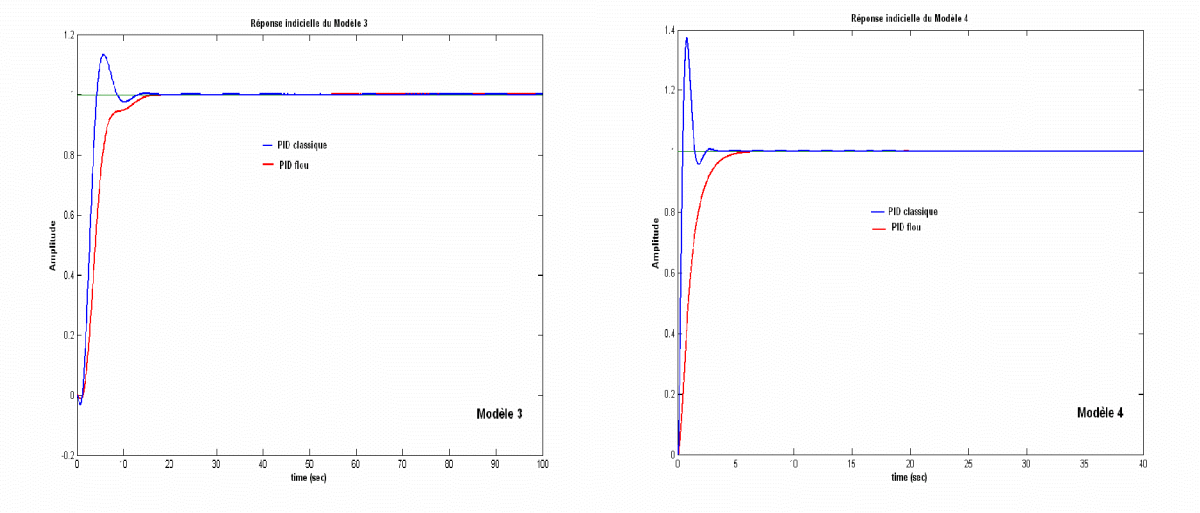



**Figure II.11 :** Réponse indicielle des troisièmes et quatrièmes modèles réglés avec un régulateur PID classique (bleu) et par un régulateur PID flou (rouge)

## II.5 Conclusion :

Nous avons vu dans ce chapitre l'intérêt de la logique floue dans le domaine du contrôle de processus. Cette approche permet de tenir compte les connaissances d'un expert humain. En fait, le fonctionnement d'un contrôleur flou dépend d'un nombre très important de paramètres (fonctions d'appartenance, règle floues, fuzzification, défuzzification). Ensuite on utilise cette stratégie de synthèse pour la régulation des paramètres d'un régulateur PID.

Une étude comparative avec le régulateur PID classique à été abordée dans plusieurs modèles. Cette étude a permis de montrer les avantages du contrôleur proposé en terme de performances, de robustesse et de flexibilité.

# Chapitre III

## III.1 Introduction :

Parmi les algorithmes évolutionnistes, les algorithmes génétiques (AG) sont les méthodes les plus répandues. Ils sont basés sur le processus d'évolution génétique des individus biologiques à travers des générations selon la théorie de Darwin et les règles de Mendel. Leur but est d'obtenir une solution approchée, en un temps correct, à un problème d'optimisation, lorsqu'il n'existe pas (ou qu'on ne connaît pas) de méthode exacte pour le résoudre en un temps raisonnable.

Tout d'abord, les algorithmes génétiques peuvent traiter des problèmes pour lesquels les solutions potentielles sont situées dans un espace de grande dimension, ce qui ne permet pas d'utiliser des méthodes standard reposant sur une exploration systématique. Ensuite, la recherche d'une solution optimale à un problème peut être réalisée par un algorithme génétique sans nécessiter de connaissance a priori sur la répartition des solutions dan l'espace.

L'utilisation, dans la résolution de problèmes, d'algorithmes génétiques est à l'origine du fruit des recherches de John Holland et de ses collègues et élèves de l'Université du Michigan qui ont, dès 1960, travaillé sur ce sujet. La nouveauté introduite par ce groupe de chercheurs a été la prise en compte de l'opérateur de « crossing over » en complément des mutations. Et c'est cet opérateur qui permet le plus souvent de se rapprocher de l'optimum d'une fonction en combinant les gènes contenus dans les différents individus de la population. Le premier aboutissement de ces recherches a été la publication en 1975 [45]. En 1989, Goldberg [46] a publié un livre sur les AG qui est devenu une référence.

De nos jours, les AG ont trouvé des applications dans des domaines très variés, allant de la biologie jusqu'aux applications liées plus directement à l'informatique (traitement d'images, optimisation, placement de formes).

## III.2 Principe de fonctionnement d'un algorithme génétique :

Le principe général du fonctionnement d'un algorithme génétique est représenté sur la figure III.1 : on commence par générer une population d'individus de façon aléatoire. Pour passer d'une génération k à la génération k+1, les trois opérations (sélection, croisement et mutation) sont répétées pour tous les éléments de la population k. Des couples de parents P1 et P2 sont sélectionnés en fonction de leurs adaptations. L'opérateur de croisement est appliqué avec une probabilité $P_c$ et génère des couples d'enfants $E_1$ et $E_2$. L'opérateur de mutation est appliqué avec la probabilité $P_m$ et génère des individus mutés E. Le niveau

d'adaptation des enfants ($E_1$, E2) et des individus mutés E sont ensuite évalués avant insertion dans la nouvelle population. Différents critères d'arrêt de l'algorithme peuvent être choisis :

- Le nombre de générations que l'on souhaite exécuter peut être fixé à priori. C'est ce que l'on est tenté de faire lorsque l'on doit trouver une solution dans un temps limité.
- L'algorithme peut être arrêté lorsque la population n'évolue plus ou plus suffisamment rapidement.

Nous allons maintenant détailler chacun de ces points.

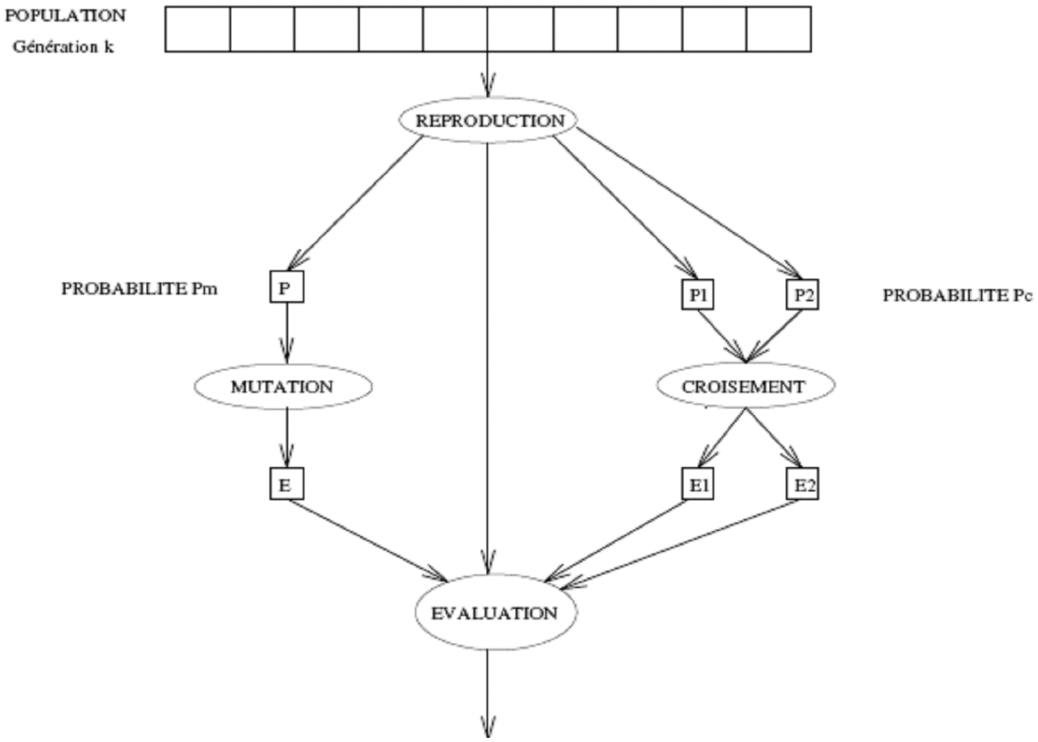



**Figure III.1 :** Principe générale des algorithmes génétiques

## III.3 Structure de l'algorithme génétique:

A l'inverse d'autres techniques d'optimisation, les AG ne requièrent pas d'hypothèse particulière sur la régularité de la fonction objective. L'algorithme génétique n'utilise pas ses dérivées successives, ce qui rend très vaste son domaine d'application. Aucune hypothèse sur la continuité n'est non plus requise. Néanmoins, dans la pratique, les AG sont sensible à la régularité des fonctions qu'ils optimisent. Le peu d'hypothèses requises permet de traiter des problèmes très complexes. L'algorithme génétique comprend les six étapes suivantes :

***Etape 1 : Codage des données :***

Historiquement le codage utilisé par les algorithmes génétiques était représenté sous forme de chaînes de bits contenant toute l'information nécessaire à la description d'un point dans l'espace d'état. Ce type de codage a pour intérêt de permettre de créer des opérateurs de croisement et de mutation simples. C'est également en utilisant ce type de codage que les premiers résultats de convergence théorique ont été obtenus.

Cependant, ce type de codage n'est pas toujours bon comme le montrent les deux exemples suivants :

- Pour des problèmes d'optimisation dans des espaces de grande dimension, le codage binaire peut rapidement devenir mauvais. Généralement, chaque variable est représentée par une partie de la chaîne de bits et la structure du problème n'est pas bien reflétée, l'ordre des variables ayant une importance dans la structure du chromosome alors qu'il n'en a pas forcément dans la structure du problème.
- Les algorithmes génétiques utilisant des vecteurs réels pour éviter ce problème, en conservant les variables du problème dans le codage de l'élément de population sans passer par le codage binaire intermédiaire. La structure du problème est conservée dans le codage.

***Etape 2 : Génération aléatoire de la population initiale :***

Le choix de la population initiale d'individus conditionne fortement la rapidité de l'algorithme. Si la position de l'optimum dans l'espace d'état est totalement inconnue, il est naturel de générer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace d'état en veillant à ce que les individus produits respectent les contraintes . Si par contre, des informations à priori sur le problème sont disponibles, il parait bien évidemment naturel de générer les individus dans un sous domaine particulier afin d'accélérer la convergence. Dans l'hypothèse où la gestion des contraintes ne peut se faire directement, les contraintes sont généralement incluses dans le critère à optimiser sous forme de pénalités. Il est clair qu'il vaut mieux, lorsque c'est possible ne générer que des éléments de population respectant les contraintes.

***Etape 3 : Fonction d'évaluation (Fitness) :***

La fonction d'évaluation joue le rôle de l'environnement où évolue la population. Elle permet d'avantager ou de désavantager un individu vis-à-vis des règles de sélection et de reproduction. De plus, l'efficacité d'un AG s'appuie pour une grande part, sur la qualité de

cette fonction. La mise au point d'une bonne fonction d'évaluation (Fitness) doit respecter plusieurs critères qui se rapportent à sa complexité et à la satisfaction des contraintes du problème. Lorsque la fonction d'évaluation apparaît excessivement complexe, consommant une importante puissance de calcul, la recherche d'une approximation plus simple est souhaitable.

La Fitness est donc donnée par une fonction à valeurs positives réelles :

- Dans le cas d'un codage binaire, nous utiliserons souvent une fonction de décodage $d$ qui permettra de passer d'une chaîne binaire à un chiffre à valeur réelle : $d : \{0,1\}^l \rightarrow \Re$ (où $l$ est la longueur de la chaîne). La fonction de fitness est alors choisie telle qu'elle transforme cette valeur en valeur positive, soit $f : d(\{0,1\}^l) \rightarrow \Re^*_+$. Le but d'un algorithme génétique est alors simplement de trouver la chaîne qui maximise cette fonction $f$. Bien évidemment, chaque problème particulier nécessitera ses propres fonctions $d$ et $f$.

***Etape 4 : <u>Principe de sélection :</u>***

Cet opérateur est peut-être le plus important puisqu'il permet aux individus d'une population de survivre, de se reproduire ou de mourir. En règle générale, la probabilité de survie d'un individu sera directement reliée à son efficacité relative au sein de la population.

La sélection permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais. On trouve dans la littérature un nombre important de principes de sélection plus ou moins adaptés aux problèmes qu'ils traitent. Dans le cadre de notre travail, les deux principes de sélection suivants ont été testés et évalués:

- Roulette Wheel sélection [47]
- Selection par tournoi [47]

***Etape 5 : <u>Opérateur de croisement:</u>***

Le croisement a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes. Classiquement, les croisements sont envisagés avec deux parents et génèrent deux enfants. Initialement, le croisement associé au codage par chaînes de bits est le croisement à découpage de chromosomes (slicing crossover). Pour effectuer ce type de croisement sur des chromosomes constitués de $M$ gènes, on tire aléatoirement une position dans chacun des parents. On échange ensuite les deux sous chaînes terminales de chacun des deux chromosomes, ce qui produit deux enfants $C_1$ et $C_2$ (voir figure III.2).

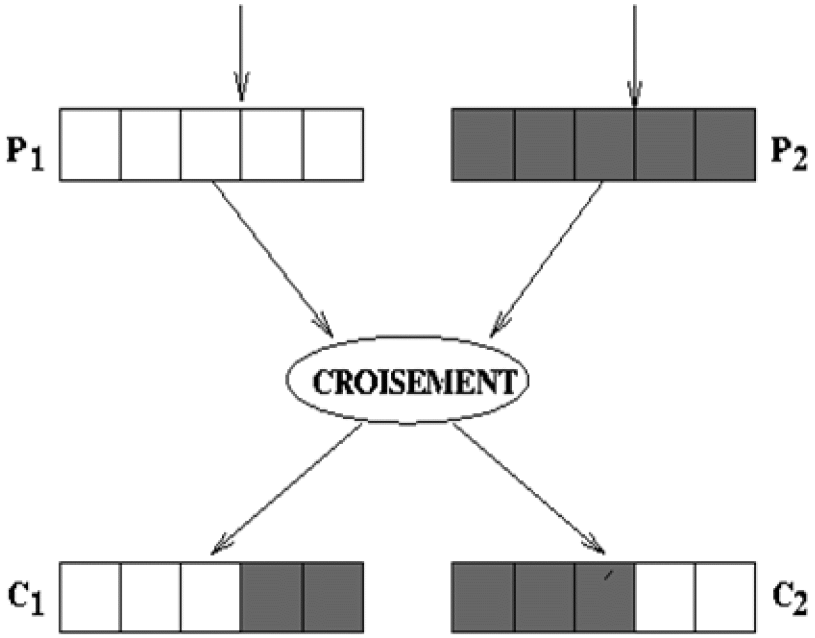



**Figure III.2** : Opérateur de croisement à 1 point

On peut étendre ce principe en découpant le chromosome non pas en 2 sous chaînes mais en 3, 4, etc. (voir figure III.3).

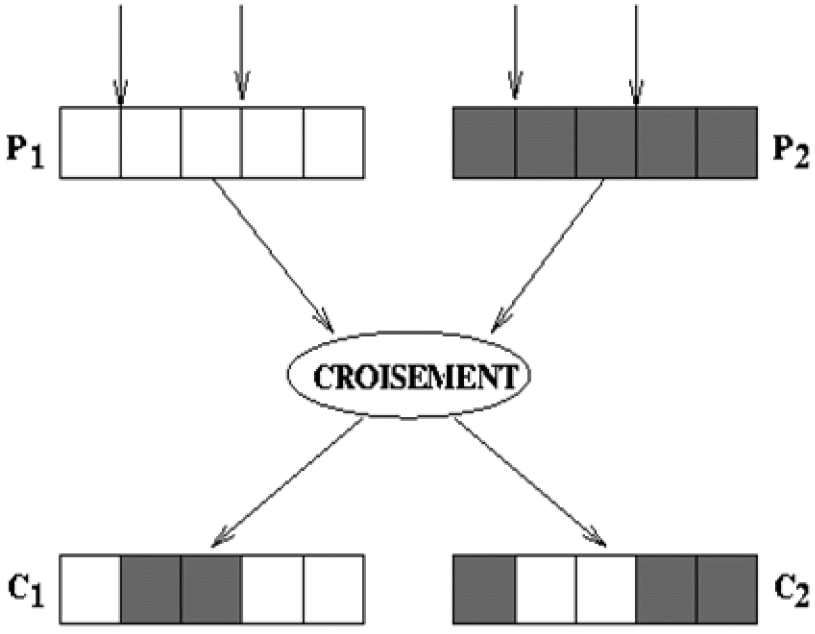



**Figure III.3 :** Opérateur de croisement à 2 points

Ce type de croisement à découpage de chromosomes est très efficace pour les problèmes discrets.

Pour les problèmes continus, un croisement ``barycentrique'' est souvent utilisé : deux gènes $P_1$ $(i)$ et $P_2$ $(i)$ sont sélectionnés dans chacun des parents à la même position $i$. Ils définissent deux nouveaux gènes $C_1$ $(i)$ et $C_2$ $(i)$ par combinaison linéaire :

$$\begin{cases} C_1(i) = \alpha P_1(i) + (1-\alpha) P_2(i) \\ C_2(i) = (1-\alpha) P_1(i) + \alpha P_2(i) \end{cases} \quad \textbf{(III.1)}$$

où $\alpha$ est un coefficient de pondération aléatoire adapté au domaine d'extension des gènes (il n'est pas nécessairement compris entre 0 et 1, il peut par exemple prendre des valeurs dans l'intervalle [-0.5,1.5] ce qui permet de générer des points entre, ou à l'extérieur des deux gènes considérés).

***Etape 6 : <u>Opérateur de mutation :</u>***

Dans les algorithmes génétiques, le processus de sélection peut parfois faire disparaître complètement certains gènes. Par exemple, suite à une phase de sélection, il peut arriver que tous les chromosomes contiennent la même sous chaîne binaire dans une position donnée. On constate facilement que cette évolution est irréversible, puisque cette sous chaîne ne peut plus être modifiée par les opérations de sélection et de croisement.

C'est à ce niveau qu'intervient l'opération de mutation. Elle permet de modifier de façon tout à fait aléatoire le chromosome d'un individu. Toutefois, pour ne pas trop perturber l'évolution globale de la population, la probabilité d'apparition d'une mutation doit rester très faible.

Pour les problèmes discrets, l'opérateur de mutation consiste généralement à tirer aléatoirement un gène dans le chromosome et à le remplacer par une valeur aléatoire (voir figure 4). Si la notion de distance existe, cette valeur peut être choisie dans le voisinage de la valeur initiale.

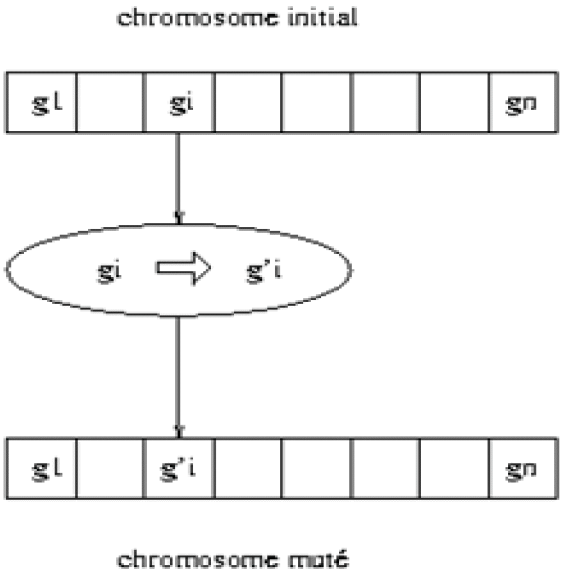



**Figure III.4 :** principe de l'opérateur de mutation

Dans les problèmes continus, on procède un peu de la même manière en tirant aléatoirement un gène dans le chromosome, auquel on ajoute un bruit généralement gaussien. L'écart type de ce bruit est difficile à choisir a priori.

## III.4 Optimisation d'un contrôleur flou par un algorithme génétique :

Comme nous l'avons vu dans le chapitre 2, les contrôleurs flous permettent de piloter des systèmes complexes ou difficilement modélisables en utilisant une méthode de raisonnement de type «***Si*** condition ***Alors*** action».

Le fonctionnement d'un contrôleur flou dépend principalement des caractéristiques de trois sous-systèmes : la fuzzification, les règles floues et la défuzzification. Plusieurs méthodes de conception, permettant de spécifier les différents paramètres dont dépend le fonctionnement d'un contrôleur flou, ont été décrites dans la littérature. Elles sont pour la plupart basées sur la définition du meilleur jeu de paramètres pour une structure donnée de contrôleur, donc on a les types d'optimisations suivantes :

- Optimisation des fonctions d'appartenance,
- Optimisation des règles floues,
- Optimisation simultanée des fonctions d'appartenance et des règles floues.

## III.5 Processus d'ajustement génétique des structures profondes :

Un processus d'ajustement basé sur les algorithmes génétiques AGs a été suggéré pour adapter les fonctions d'appartenance en changeant leurs paramètres de base et additionnels.

L'ajustement commence à partir d'une base de connaissance prédéfinie (base de données + base de règles) dérivée d'une méthode d'apprentissage ou fournie par des experts.

Les sections suivantes présentes une introduction sur la modélisation floue, le processus génétique proposé, ses composants, et finalement quelques dissertations au sujet de l'intèrprétabilité des modèles ajustés.

## III.5.1 La modélisation floue :

La modélisation floue (MF), signifie modéliser un processus en représentant son comportement dynamique à l'aide d'un système basé sur les règle floues, en se servant d'un langage descriptif qui est la logique floue.

Un des domaines les plus importants est la MF linguistique, où l'intèrprétabilité du modèle obtenu est l'objectif principal. Cette tâche est développée au moyen de systèmes flous linguistiques, qui utilisent des règles floues composées de variables linguistiques. Ainsi, le modèle linguistique flou se compose d'un ensemble de description linguistique concernant le comportement du système ainsi modélisé. Chacune de ces règles floues linguistiques peut être représentée par deux niveaux de description en définissant deux structures différentes.

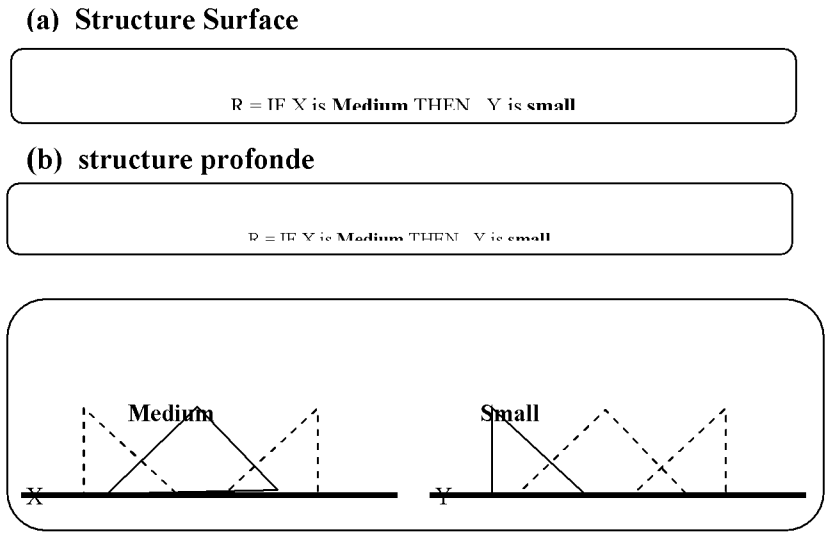



**Figure III.5 :** deux manières différentes de définir une règle floue linguistique.

(a) *Structure surface* = représentation symbolique

(b) *Structure profonde* = représentation symbolique + formes des fonctions d'appartenances

☙ ***Structure surface :*** c'est une description moins spécifique et implique de définir la règle sous sa forme symbolique comme une relation entre les variables linguistiques d'entrée et de sortie.

- ***Structure profonde :*** c'est une description plus spécifique et comprend la structure surface ainsi que les définitions des fonctions d'appartenance associées aux termes linguistiques des variables.

Un des problèmes les plus importants dans les applications de la logique floue est la dérivation automatique des ces structure à partir d'informations numériques (les données d'entrées-sorties) représentant le comportement du système réel.

Des méthodes automatiques nombreuses, basées sur différentes techniques telles que les résaux neuro-floue [48] et les algorithmes génétiques (49), ont été développées pour accomplir cette tâche. Un des buts cruciaux émergent pendant la conception d'un modèle flou, est d'obtenir un modèle précis et compréhensible.

Actuellement, une des matières les plus prometteuses de recherches dans MF se relie avec la recherche d'un compromis entre l'intèrprétabilité et la précision [50]

## III.5.2 Processus d'ajustement :

La proposition de l'ajustement génétique est caractérisée comme suit : [51]

- L'objectif (fonction d'évaluation « fitness ») sera de réduire au minimum *l'erreur moyenne quadratique* (MES) : $MSE = \frac{1}{2.N}\sum_{l=1}^{N}\left(F\left(x^l\right)-y^l\right)^2$ **III.2**

Avec $N$ est la taille des données, $F(x^l)$ est la sortie obtenue à partir du système flou considéré pour l'exemple $l$, $y^l$ est la sortie désirée.

- Un triple code $(CS_p + CS_A)$ est employé. $CS_p$ *(coding sheme)* codera les paramètres de base de la fonction d'appartenance, et $CS_A$ les paramètres $\alpha$ de chaque fonction d'appartenance (c'est-à-dire, les facteurs d'échelle non linéaires). Par conséquent, $CS_p$ et $CS_A$ sont employés pour ajuster la sémantique des structures profondes. La figure (III.5) montre graphiquement cet arrangement.

  - Pour la branche $CS_p$, un triplet des valeurs réelles pour chaque fonction triangulaire d'appartenance est employé, donc la base de données est codée dans un chromosome construit par des valeurs réelles en liant les fonctions d'appartenance impliquées dans chaque partition floue. Un intervalle de variation est défini pour chaque paramètre de base.

- Pour la branche $CS_A$, un chromosome des valeurs réelles qui code la valeur du paramètre additionnel $\alpha$ pour chaque fonction d'appartenance est employé. Chaque gène peut prendre n'importe quelle valeur dans l'intervalle [-1,1] avec la relation suivante entre les allèles et la valeur réelle :

$$c^A_{ij} \in [-1,0] \longleftrightarrow \alpha \in [min, 1]$$

$$c^A_{ij} \in [0,1] \longleftrightarrow \alpha \in [1, max]$$

**Avec** $$\min = \frac{\log(0.5)}{\log(s_\alpha)} \qquad \max = \frac{\log(0.5)}{\log(1-s_\alpha)} \qquad \textbf{III.3}$$

$c^A_{ij}$ est le gène associe à la fonction d'appartenance pour la $j^{ème}$ terme linguistique de la $i^{ème}$ variable, et $S_\alpha \in [0, 0.5]$ est un paramètres qui définit le degré de flexibilité accordé à l'ajustement des fonctions d'appartenances ($S_\alpha = 0$ pour une flexibilité maximum et $S_\alpha = 0.5$ pour une flexibilité minimum, c'est-à-dire pas d'ajustement). Les formes extrêmes de fonctions d'appartenance permises par cette valeur sont présentées sur la figure (III.6)

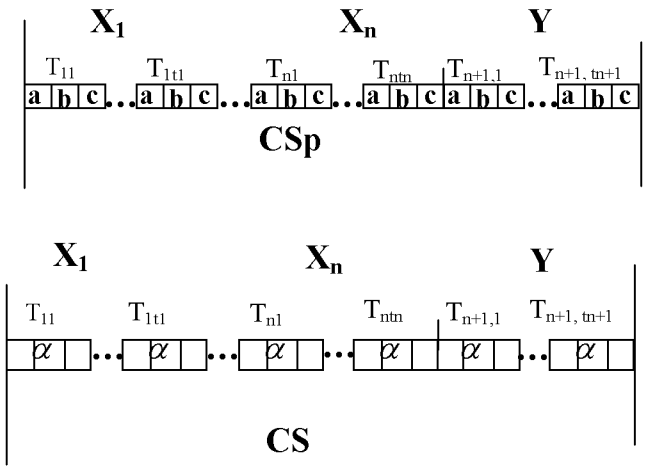



**Figure III.6 :** codage du chromosome avec *n* est le nombre de variables d'entrée, $T_{ij}$ étant le $j^{ème}$ terme linguistique de la $i^{ème}$ variable (avec n + 1 variable de sortie). $T_i$ le nombre de termes linguistiques de la $i^{ème}$ variable, et m le nombre de règles floues linguistiques.

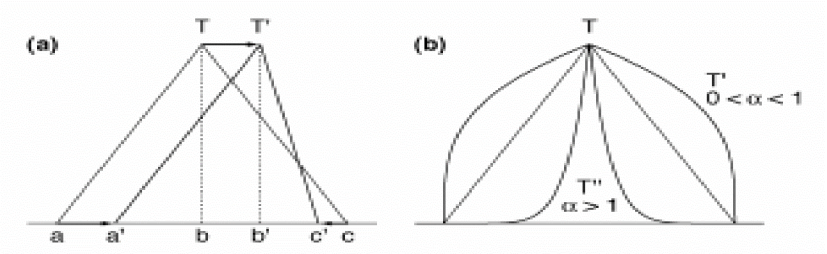



**Figure III.7 :** deux genres d'ajustement de la forme de la fonction d'appartenance.

(a) Ajustement en changeant les paramètres de base de la fonction d'appartenance.

(b) Ajustement en employant des facteurs d'échelle non-linéaire .

### III.5.3 Composants génétiques :

La méthode d'ajustement génétique a les composants suivants :

- ***Population initiale :***

Dans la population initiale, une partie d'information originale dans la base de connaissance initiale sera mélangée avec des valeurs aléatoires.

- Dans les parties $CS_p$, les valeurs réelles (originales) seront directement incluses.
- Pour la branche $CS_A$, les valeurs originales dépendront de l'utilisation de ces paramètres dans la base de connaissance initiale ou non. Si oui, les paramètres $\alpha$ seront codés ; sinon, le gène 0 (qui signifie que $\alpha = 1$) sera employé. Dans notre travail nous avons employés le gène 0.
- Les trois étapes suivantes sont considérées pour l'initialisation de la population.

**a.** Un chromosome où les gènes dans $CS_p$ et $CS_A$ coderont directement les valeurs correspondant à la base de connaissance originale.

**b.** Un tiers de la population est produit avec la branche $CS_p$ codée aléatoirement dans l'intervalle de variation de chaque gène.

**c.** Un autre tiers de la population est produit avec des allèles aléatoires (dans l'intervalle [-1,1]) pour $CS_A$.

- ***Opérateur de croisement :***

L'opérateur de croisement dépendra du codage de chromosome et l'endroit où il est appliqué.

- Pour $CS_p$ et $CS_A$, le croisement max-min arithmétique [48] est considéré.

Si $C^t_v = (c_1, \ldots, c_k, \ldots, c_H)$ et $C^t_w = (c_1, \ldots, c_k, \ldots, c_H)$ vont être croisé, les quatres fils suivants sont produits :

$$C_1^{t+1} = aC_w^t + (1\text{-}a)C_v^t$$

$$C_2^{t+1} = aC_v^t + (1\text{-}a)C_w^t$$

$$C_3^{t+1} \text{ avec } c_{3,k}^{t+1} = min\{c_k, c_k'\} \qquad \text{(III.4)}$$

$$C_4^{t+1} \text{ avec } c_{4,k} = max\{c_k, c_k'\}$$

Le paramètre $\alpha \in [0, 0.5]$ est défini. Notant que l'intervalle de variation de chaque gène ne sera jamais excédé en raison de $min\{c_k, c_k'\} \leq max\{c_k, c_k'\}, \forall i \in \{1, 2, 3,4\}$ (III.5)

Après la recombinaison de chaque partie, les deux meilleurs chromosomes parmi quatre obtenus seront choisis pour remplacer leurs parents. La figure (III.7) montre l'opération graphiquement.

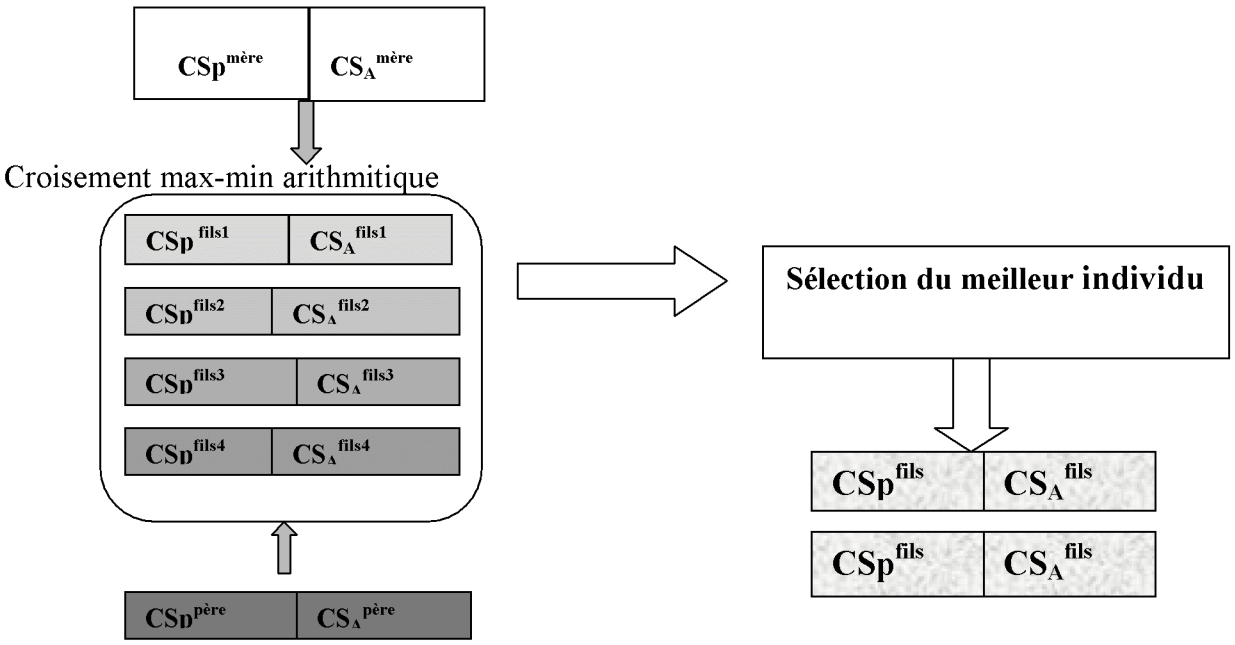



**Figure III.8 :** opérateur de croisement

- ***Opérateur de mutation :***

L'opérateur de mutation dépendra également de la branche de chromosome où il est appliqué.

- Dans les parties $CS_p$ et $CS_A$, un opérateur de mutation uniforme est considéré. Il implique de changer la valeur du gène choisi par autre aléatoirement produite dans l'intervalle correspondant.

- ***Opérateur de sélection :***

Un AG avec le procédé de « Roulette Wheel Selection » [52] ainsi l'élitisme (qui assure le choix du meilleur individu de la génération précédente) est considéré.

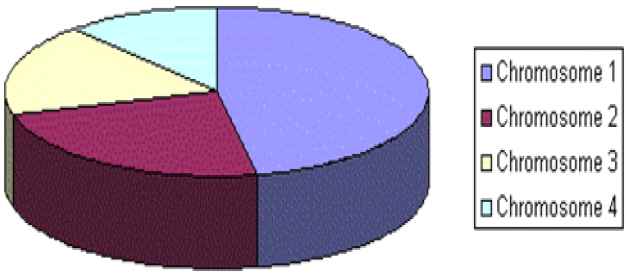



**Figure III.9 :** la roue de sélection

## III.6 Interprétabilité dans la méthode d'ajustement proposée :

Le processus d'ajustement est conçu pour rendre la définition rigide des modèles flous linguistiques plus flexible dans le but d'augmenter leur précision. Néanmoins, quelques aspects sur l'interprétabilité doivent être considérés dans le but de préserver une lisibilité appropriée des modèles linguistiques ajustés.

- ***Compréhensibilité des fonctions d'appartenances :***

Une fonction d'appartenance est compréhensible tant que la signification du terme linguistique associée est facilement compréhensible, cette mesure est fortement subjective. Nous pourrions dire que l'utilisation de la fonction d'appartenance avec une forme correcte (vérifiant la contrainte $a \leq b \leq c$) et une similitude raisonnable avec la définition initiale (avec ajustement) assure leur compréhensibilité. La méthode d'ajustement proposée aborde cette question contraignant l'optimisation de chaque gène dans les branches $CS_p$ et $CS_A$. Les paramètres de base de la fonction d'appartenance (partie $CS_p$) sont contraints par des intervalles de courtes variations. La figure III.9 montre un exemple de l'intervalle considéré pour chaque paramètre.

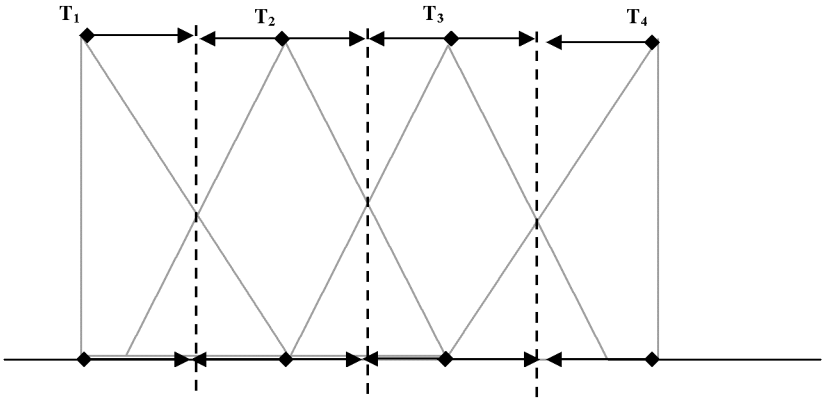



**Figure III.10 :** intervalle de variation pour chaque paramètres de fonction d'appartenance.

## III.7 Conclusion :

Dans ce chapitre, nous avons décrit brièvement le principe des algorithmes génétiques, qui permettent d'aborder la majorité des problèmes d'optimisation en utilisant un formalisme simple. Ensuite, nous donnons comment les algorithmes génétiques peuvent être utilisés pour choisir les paramètres d'un contrôleur flou.

# Chapitre IV

## IV.1 Introduction :

L'étude sera concentrée sur l'application du processus d'ajustement à un modèle flou pour l'ajustement des paramètres d'un régulateur PID ($K_p$, $T_i$, $T_d$).

L'objectif est la conception d'un contrôleur flou capable d'évaluer le signal de commande pendant le régime transitoire pour diminuer le temps de montée et au même réduire le dépassement.

## VI.2 Synthèse du contrôleur flou par algorithme génétique :

Afin d'optimiser la structure du contrôleur flou montré dans le chapitre 3, nous décrivons tout d'abord le codage utilisé pour représenter les différentes fonctions d'appartenance du contrôleur puis nous présentons le résultat de la synthèse pour les quatre modèles proposés.

- ***Codage des fonctions d'appartenance :***

Dans le contrôleur flou proposé, les entrées e (t) et Δe (t) (l'erreur et sa variation) et la sortie u (t) prennent leurs valeurs dans l'univers du discours [-1, 1] et sont partitionnés en cinq classes (sous ensembles flous) pour les entrées et neuf classes pour la sortie. Les termes linguistiques décrivant les entrées et la sortie sont représentés par des vecteurs réels et chaque vecteur comporte trois variables.

- ***Paramètres de l'algorithme génétique :***

Dans ce chapitre, nous avons vérifie que le choix des paramètres n'est pas critique. On peut simplement fixer les valeurs de ces paramètres et les conserver constats durant tout le processus d'optimisation.

- Une taille de population de 50 individus.
- 10, 50, 100, 200 évaluations.
- Méthode de sélection : Roulette Wheel.
- Méthode de croisement : Max-Min Arithmétique.
- Méthode de mutation : Uniforme.

## VI.3 Résultat de l'optimisation :

Nous présentons les résultats obtenus à l'issus du processus d'optimisation du contrôleur flou pour les mêmes systèmes proposés dans le 1er chapitre. Les fonctions de transfert de ces systèmes sont données par :

Modèle 1 : $$G_1(p) = \frac{1}{(1+10.p)^2} e^{-0.8.p}$$

Modèle 2 : $$G_2(p) = \frac{1}{(1+p)^3}$$

Modèle 3 : $$G_3(p) = \frac{1-0.5.p}{(1+p)^3}$$

Modèle 4: $$G_4(p) = \frac{1}{(1+p)(1+\alpha.p)(1+\alpha^2.p)(1+\alpha^3.p)}, \quad où \quad \alpha = 0.2$$

La figure VI. 1 montre les meilleures valeurs du fitness obtenues durant les 10 générations. Les fonctions d'appartenances ajusté des différents termes linguistiques qualifiant les entrées e(t) et Δe(t) et la sortie u(t) sont représentés sur la figure VI. 2.

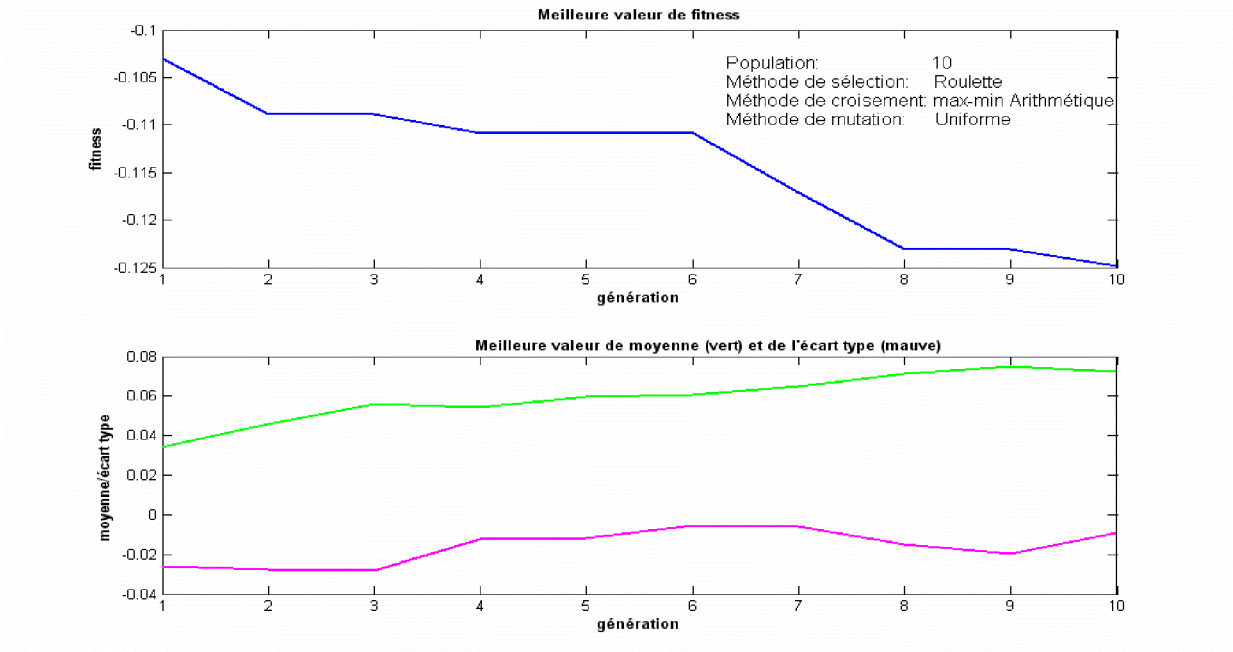



**Figure VI. 1 :** Les meilleures valeurs de fitness obtenus pour les 10 générations

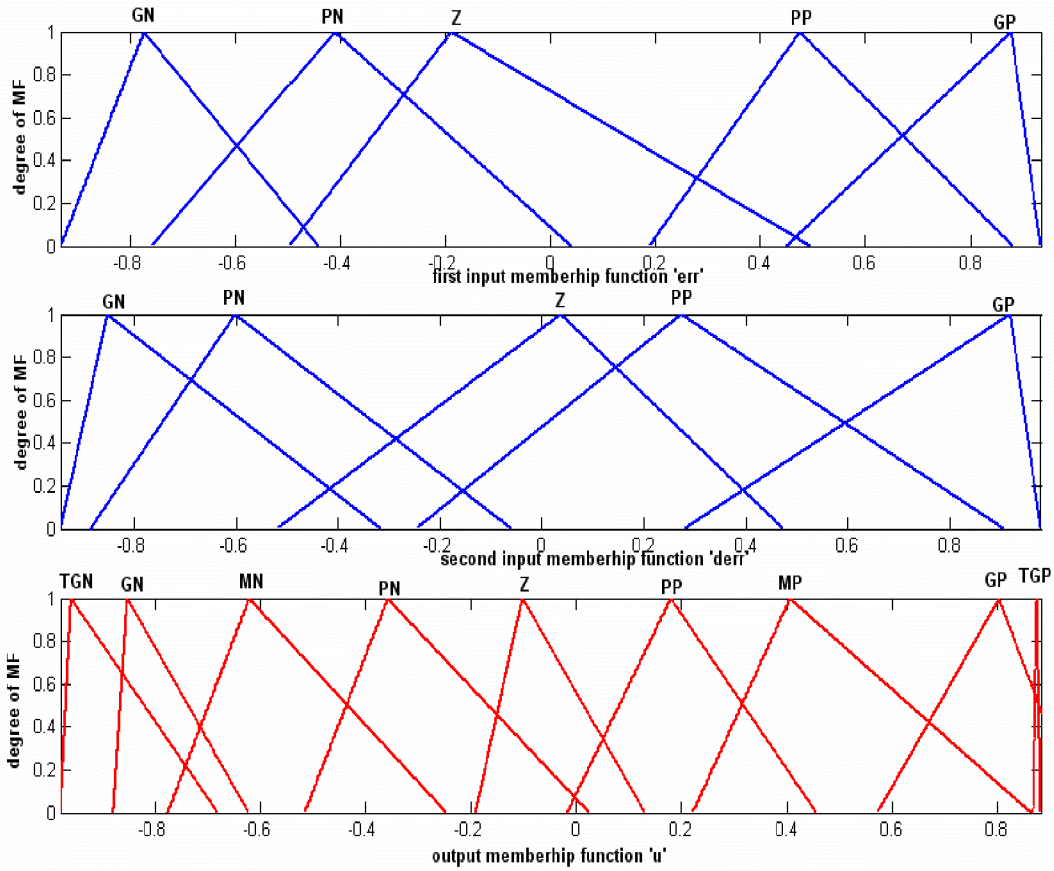



**Figure VI. 2 :** Les fonctions d'appartenance ajustée par les algorithmes génétiques pour les deux entrées e (t) et Δe (t) (bleu) et la sortie u (t) (rouge) pour les 10 générations.

La figure VI. 3 montre les meilleures valeurs du fitness obtenues durant les 50 générations. Les fonctions d'appartenances ajusté des différents termes linguistiques qualifiant les entrées e(t) et Δe(t) et la sortie u(t) sont représentés sur la figure VI. 4.

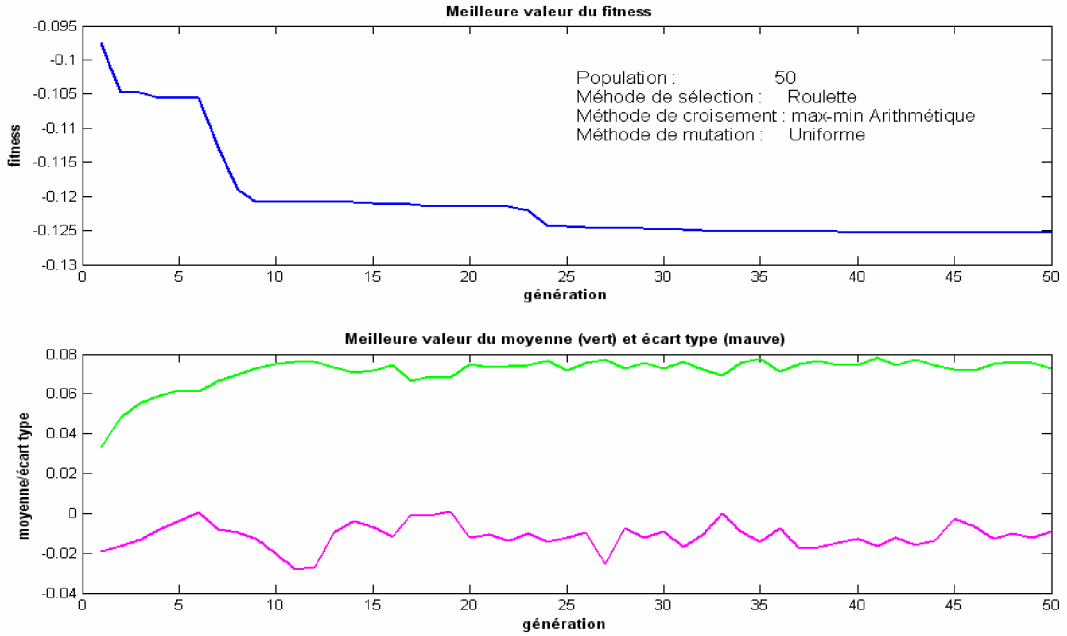



**Figure VI.3 :** Les meilleures valeurs de fitness obtenus pour les 50 générations.

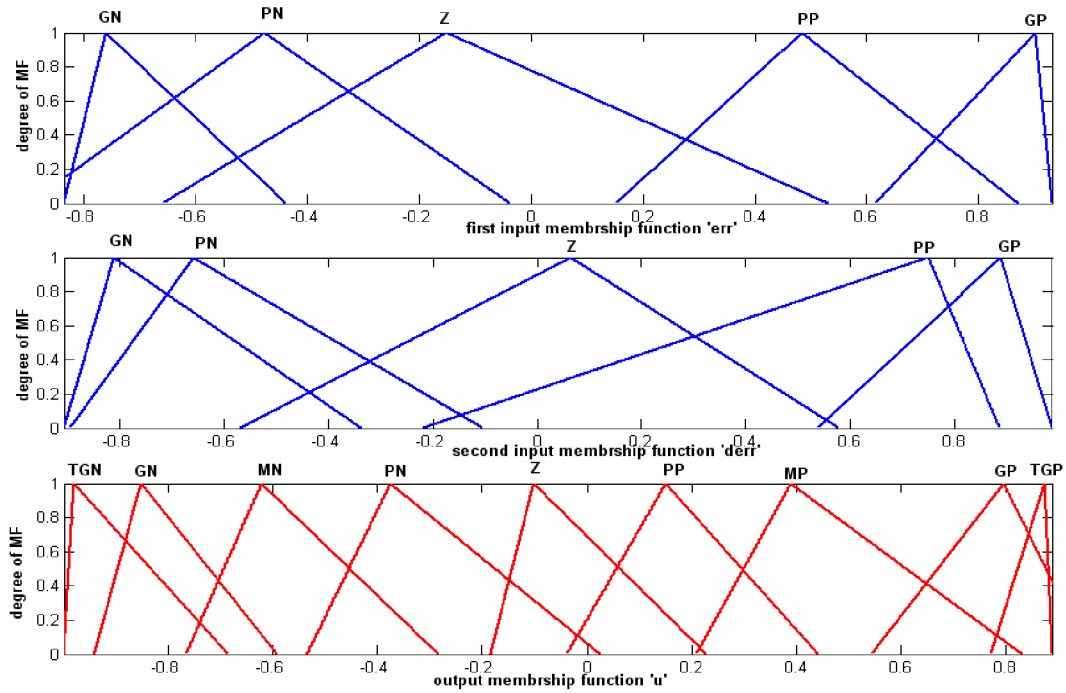



**Figure VI. 4:** Les fonctions d'appartenance ajustée par les algorithmes génétiques pour les deux entrées e (t) et Δe (t) (bleu) et la sortie u (t) (rouge) pour les 50 générations.

La figure VI. 5 montre les meilleures valeurs du fitness obtenues durant les 100 générations. Les fonctions d'appartenances ajusté des différents termes linguistiques qualifiant les entrées e(t) et Δe(t) et la sortie u(t) sont représentés sur la figure VI. 6.

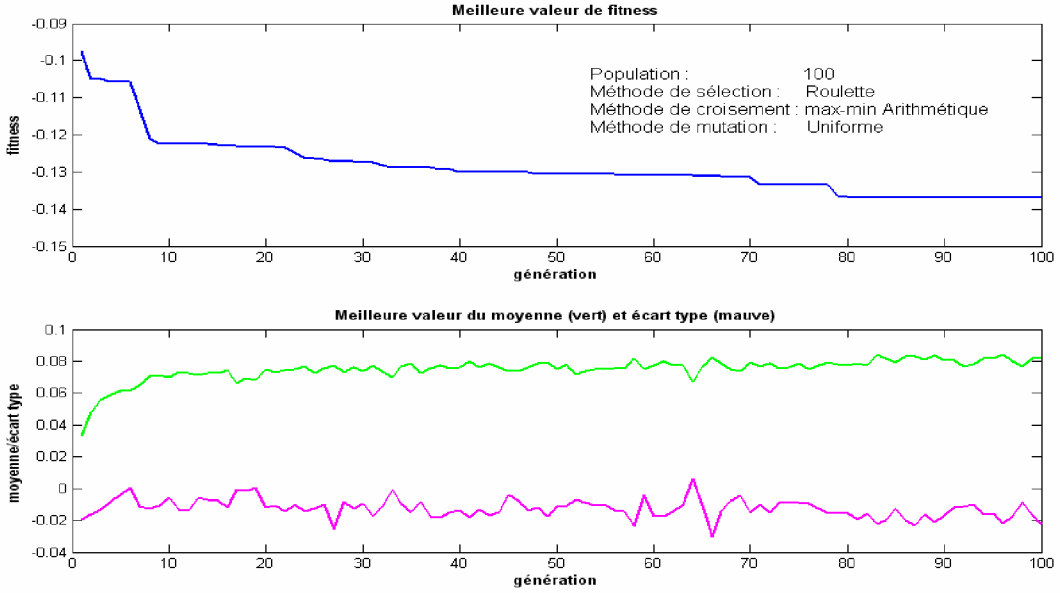



**Figure VI.5 :** Les meilleures valeurs de fitness obtenus pour les 100 générations.

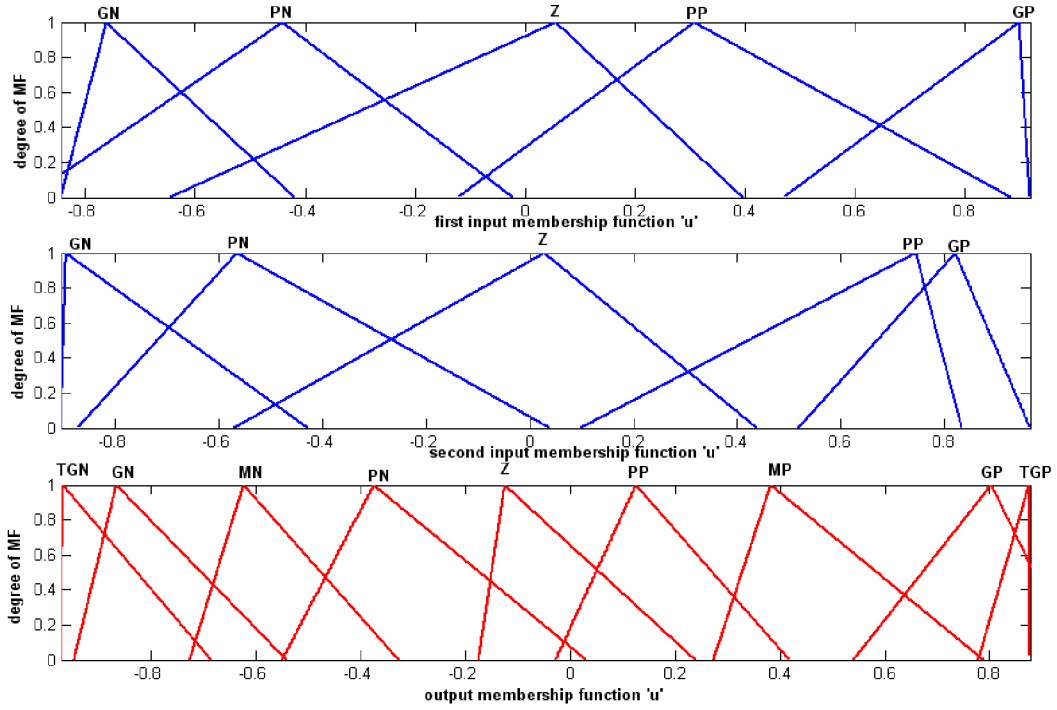



**Figure VI. 6 :** Les fonctions d'appartenance ajustée par les algorithmes génétiques pour les deux entrées e (t) et Δe (t) (bleu) et la sortie u (t) (rouge) pour les 100 générations.

La figure VI. 7 montre les meilleures valeurs du fitness obtenues durant les 200 générations. Les fonctions d'appartenances ajusté des différents termes linguistiques qualifiant les entrées e(t) et Δe(t) et la sortie u(t) sont représentés sur la figure VI. 8.

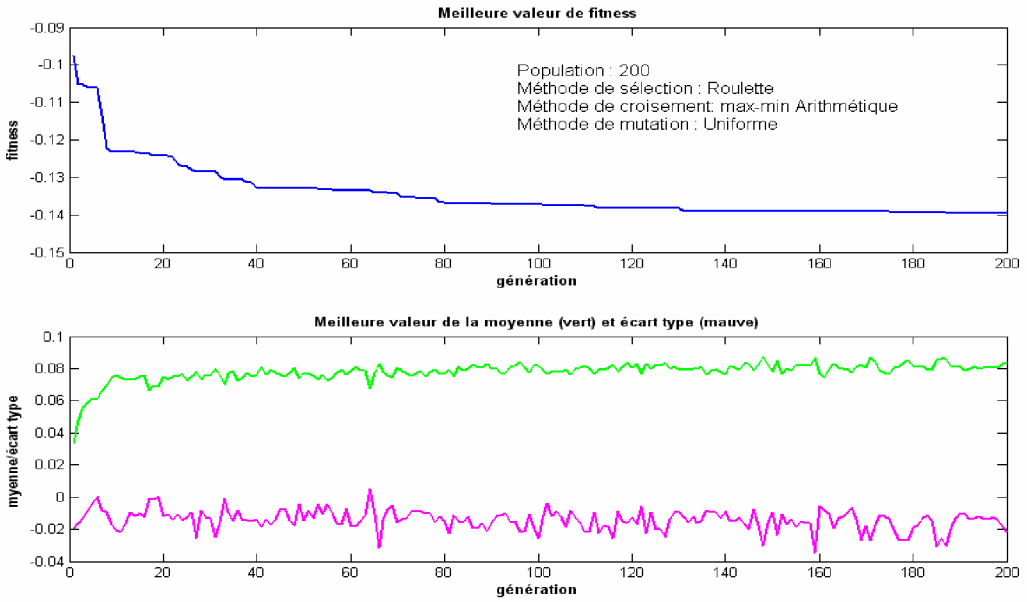



**Figure VI.7 :** Les meilleures valeurs de fitness obtenus pour les 200 générations

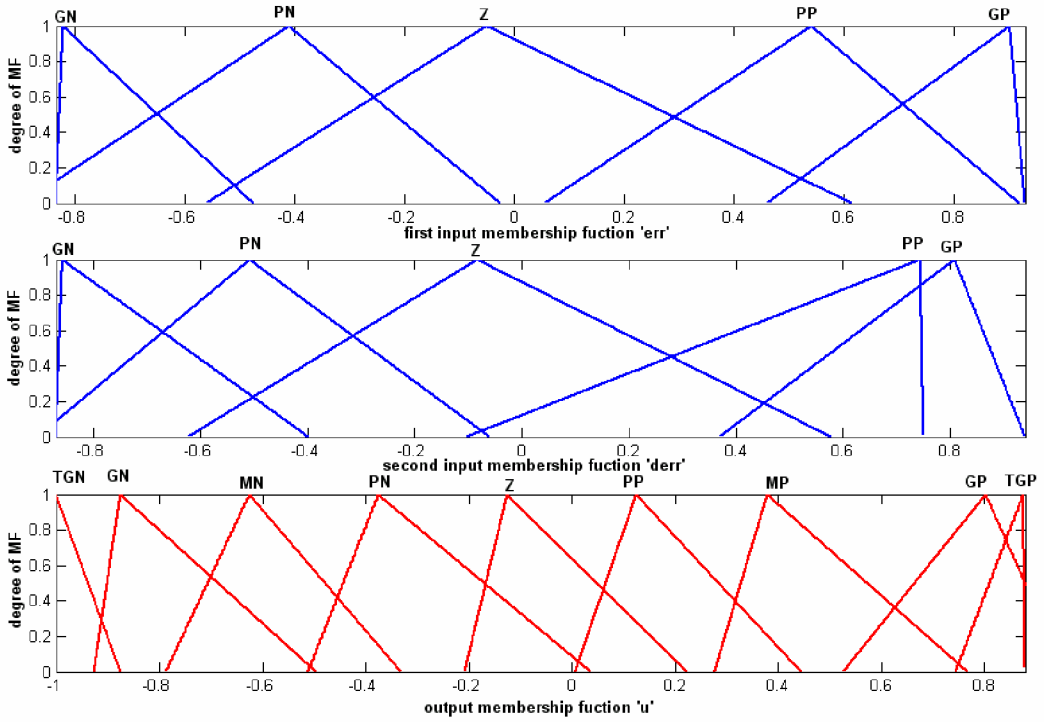



**Figure VI. 8 :** Les fonctions d'appartenance ajustée par les algorithmes génétiques pour les deux entrées e (t) et Δe (t) (bleu) et la sortie u (t) (rouge) pour les 200 générations.

Afin de définir les différentes formes des fonctions d'appartenances ajustées pour évaluer le signal de commande utilisé pour diminuer le temps de montée et au même réduire le dépassement, et pour montrer l'efficacité du PID flou ajusté dans lequel le contrôleur flou résulte d'un processus d'optimisation, nous avons comparé ses performances à celle d'un autre régulateur PID classique. Mais avons de déterminer les réponses indicielles à un échelon nous avons donnés les signaux de commande pour chaque modèle.

La figure VI. 9 montres les signaux de commande pour les quatre modèles.

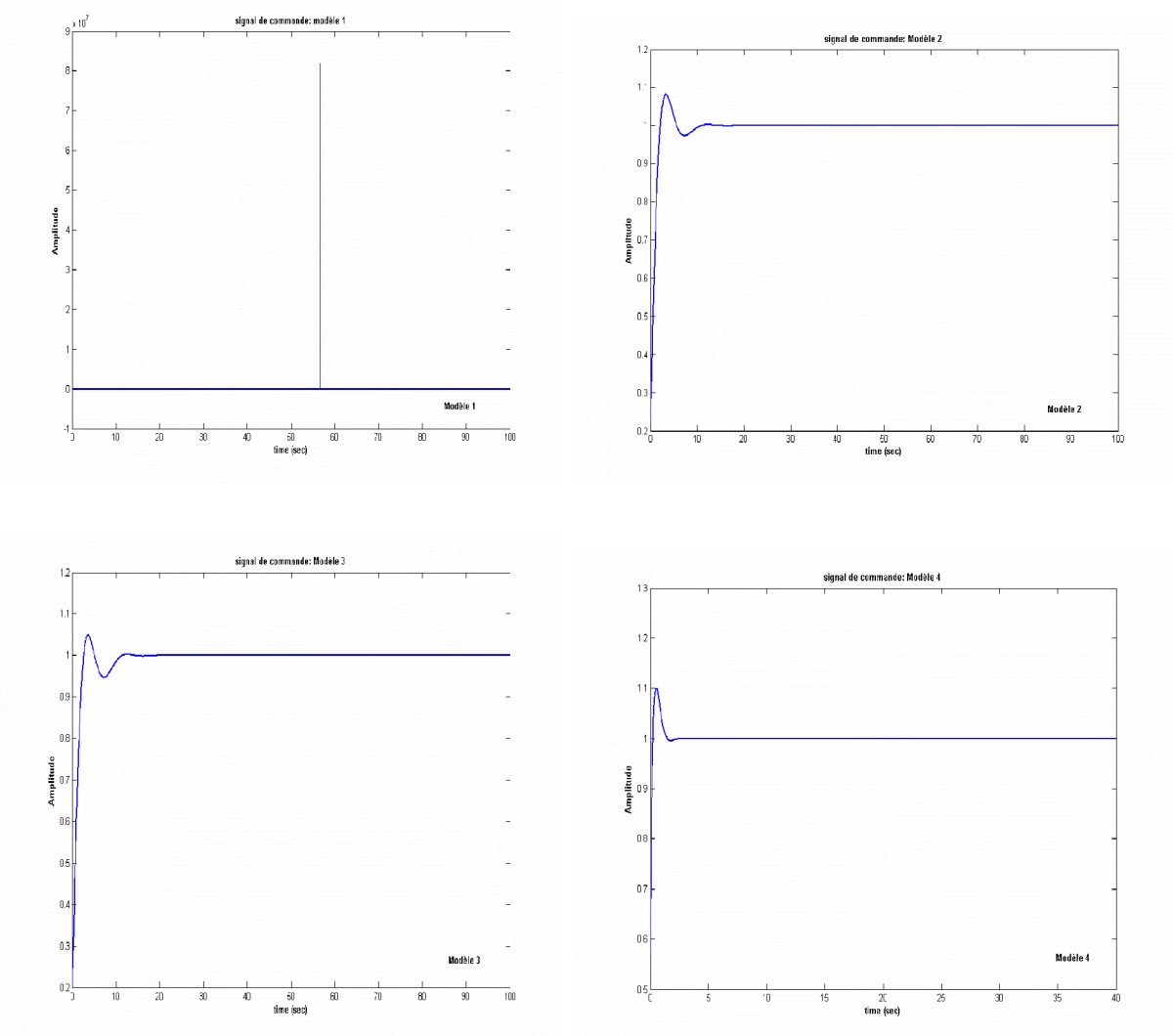
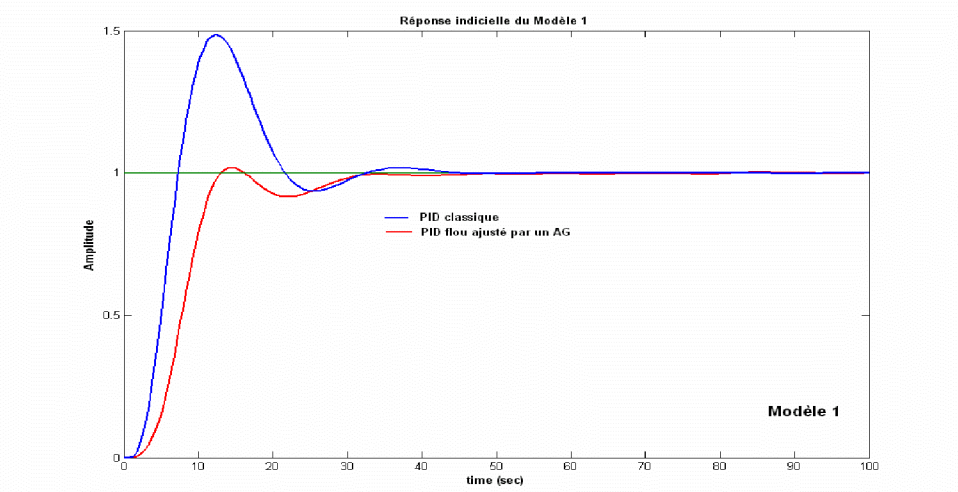



**Figure IV. 9 :** signaux de commandes des quatre modèles

Les figures IV.10 et IV.11 montre les réponses indicielles du première et deuxième modèle réglés par un régulateur PID classique (bleu) et un régulateur PID flou ajusté par un algorithme génétique après les 10 générations



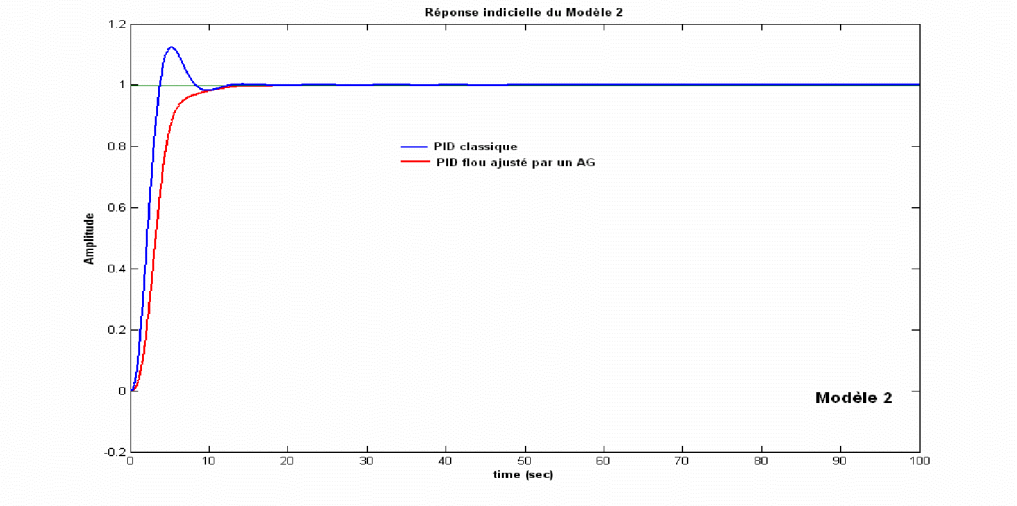
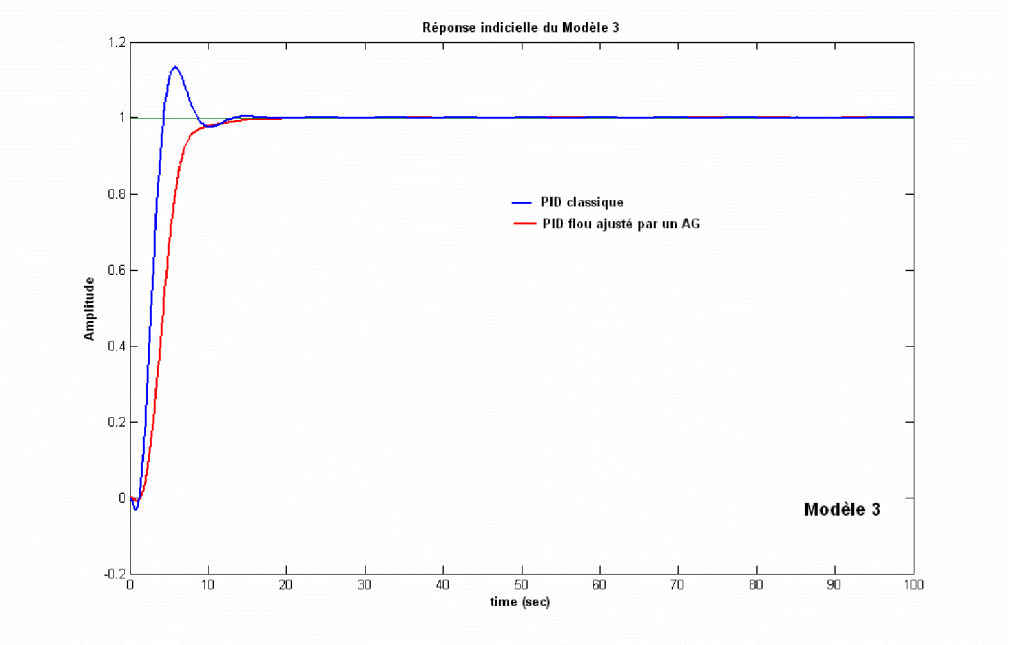




**Figure II.10 :** Réponse indicielle du premier et deuxième modèles réglés avec un régulateur PID classique (bleu) et par un régulateur PID flou ajusté par un AG pour 10 générations (rouge).



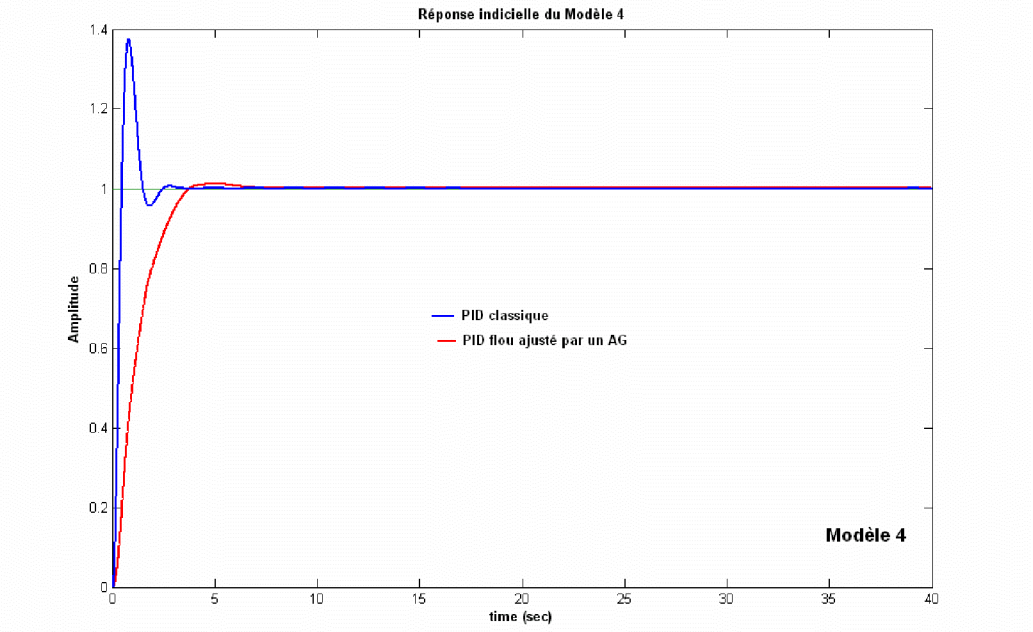
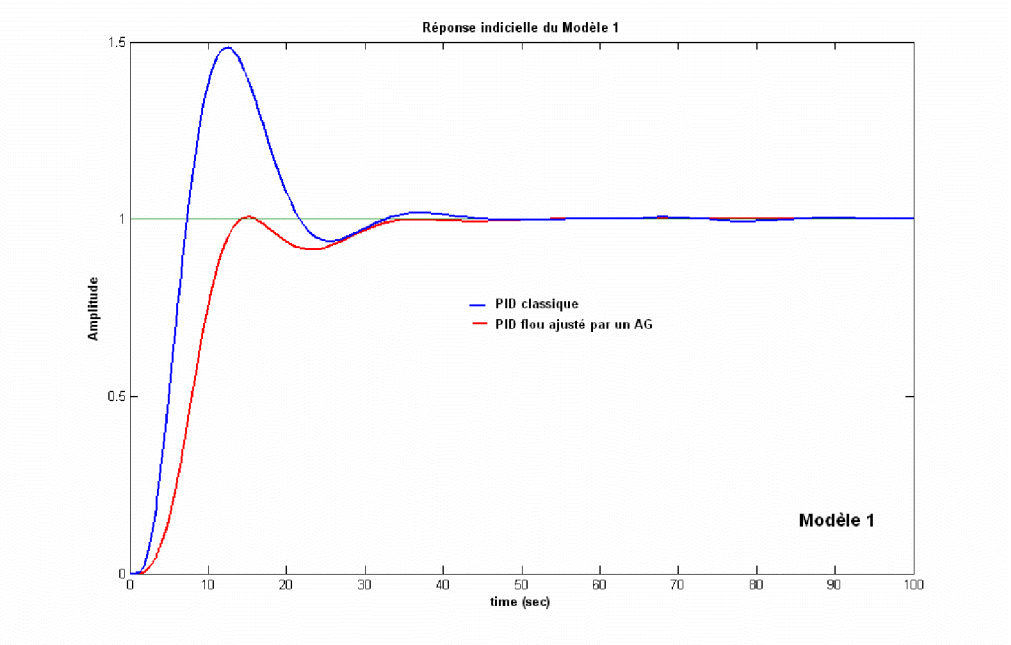




**Figure II.11 :** Réponse indicielle du troisième et quatrième modèles réglés avec un régulateur PID classique (bleu) et par un régulateur PID flou ajusté par un AG pour 10 générations (rouge).

Les figures VI. 12 et VI.13 montre les réponses indicielles du première et deuxième modèles réglés par un régulateur PID classique (bleu) et un régulateur PID flou ajusté par un algorithme génétique après les 50 générations.



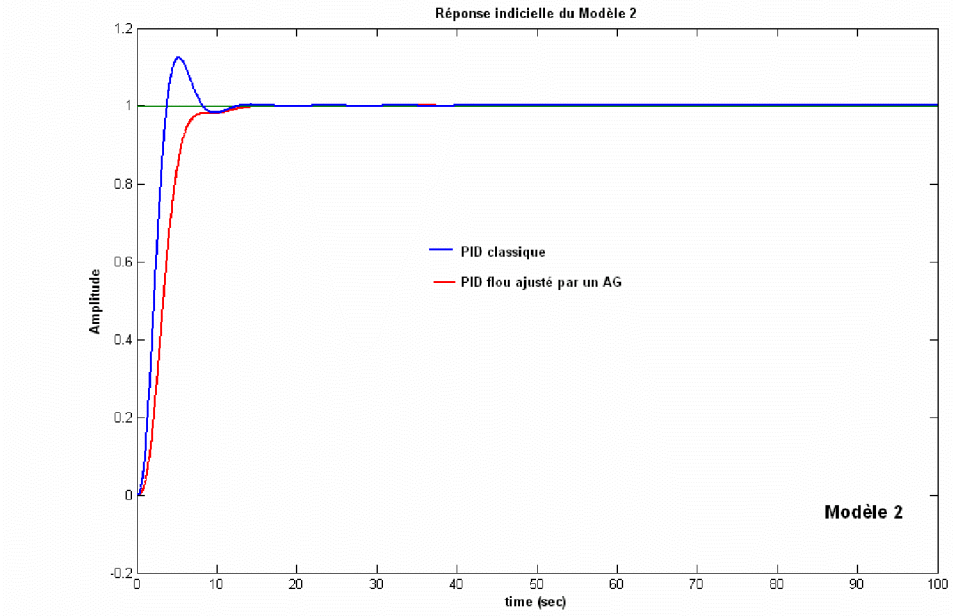
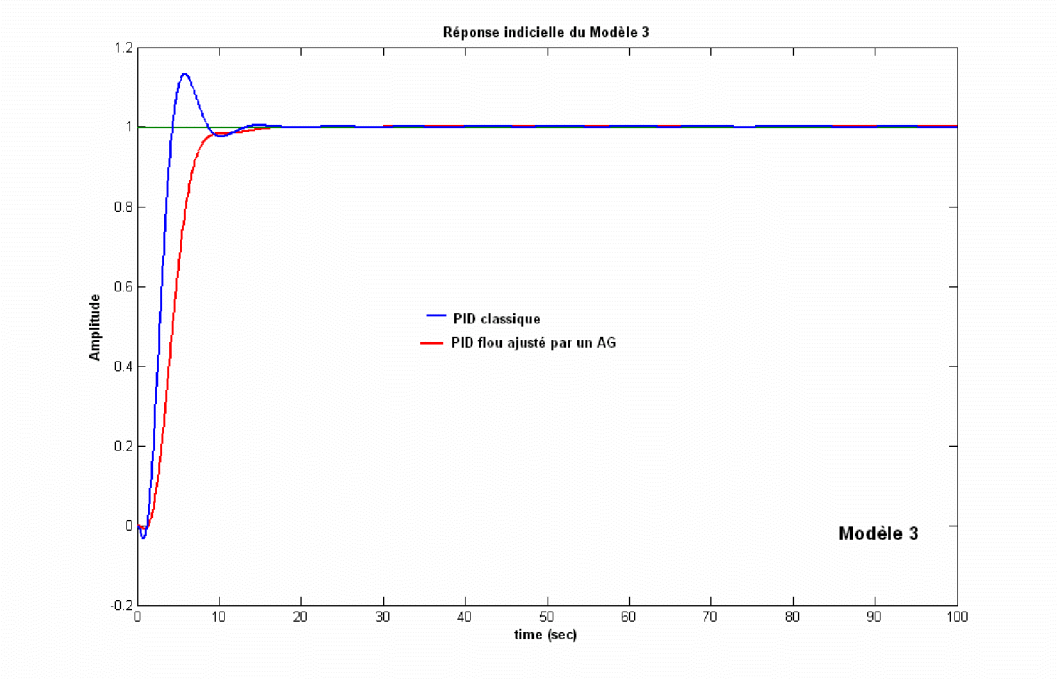




**Figure II.12 :** Réponse indicielle du première et deuxième modèles réglés avec un régulateur PID classique (bleu) et par un régulateur PID flou ajusté par un AG pour 50 générations (rouge).



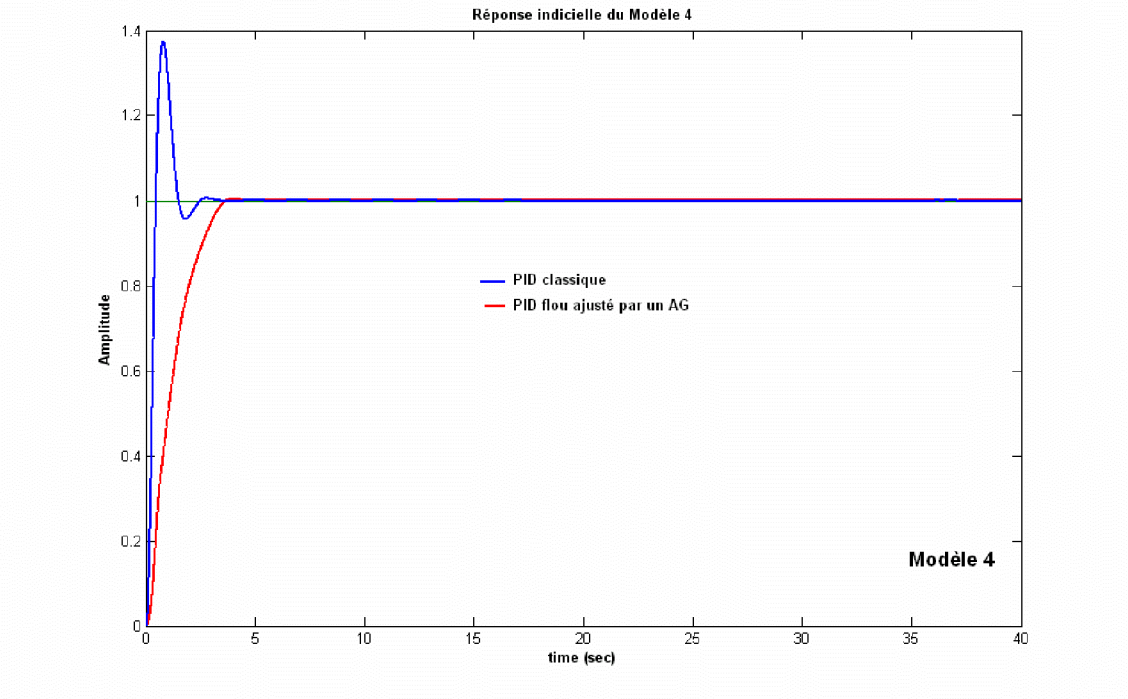
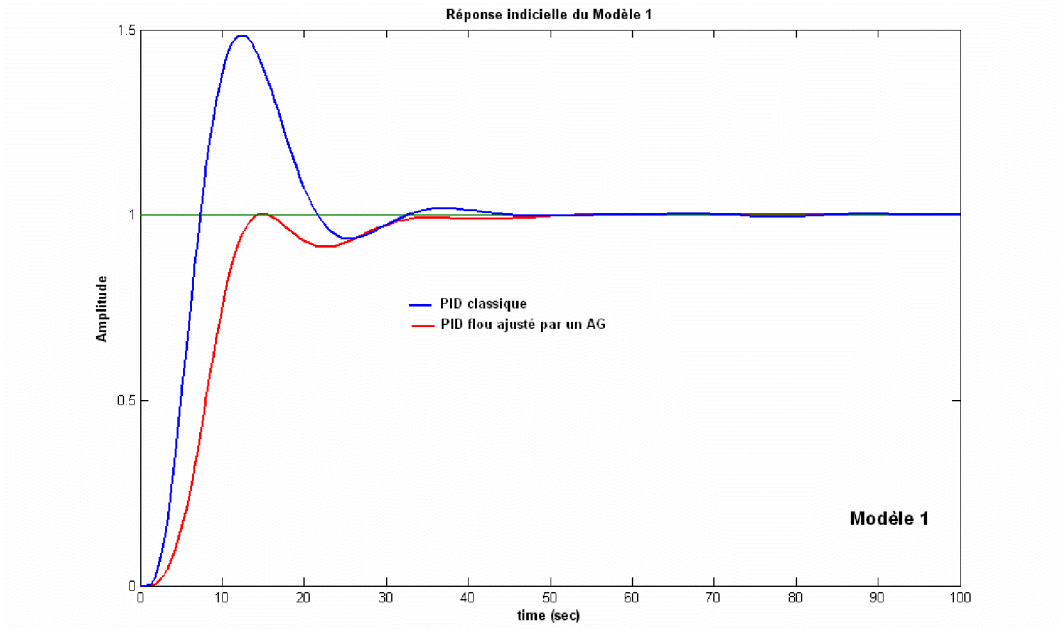




**Figure II.13 :** Réponse indicielle du troisième et quatrième modèles réglés avec un régulateur PID classique (bleu) et par un régulateur PID flou ajusté par un AG pour 50 générations (rouge).

Les figures VI. 14 et VI.15 montre les réponses indicielles du première et deuxième modèles réglés par un régulateur PID classique (bleu) et un régulateur PID flou ajusté par un algorithme génétique après les 100 générations.



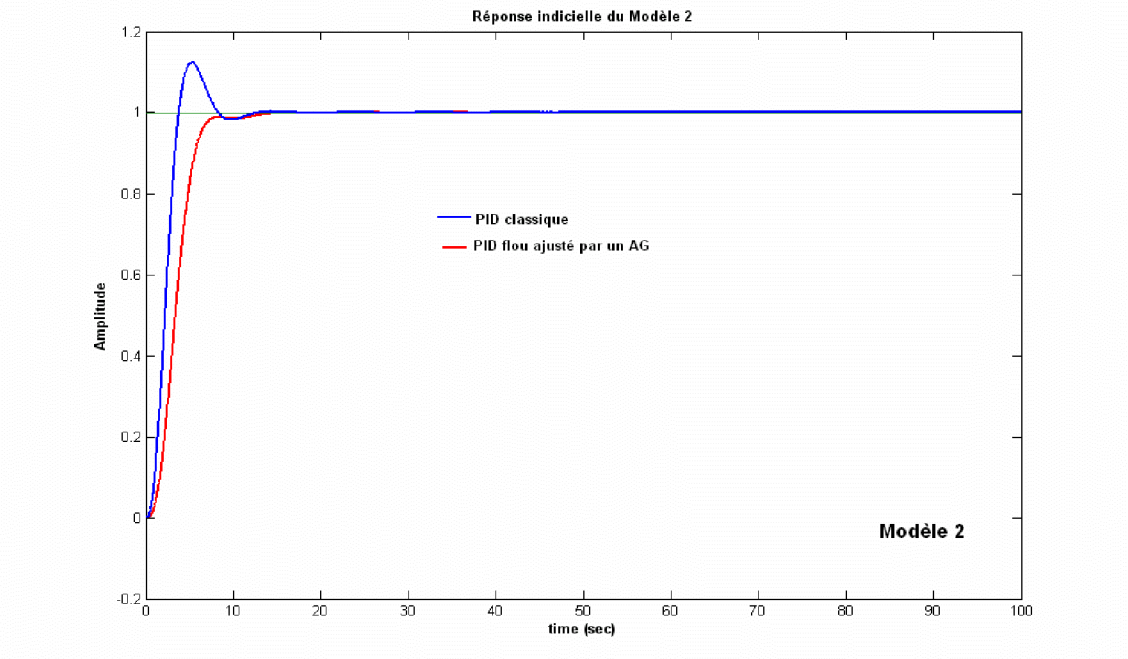




**Figure II.14 :** Réponse indicielle du première et deuxième modèles réglés avec un régulateur PID classique (bleu) et par un régulateur PID flou ajusté par un AG pour 100 générations (rouge).

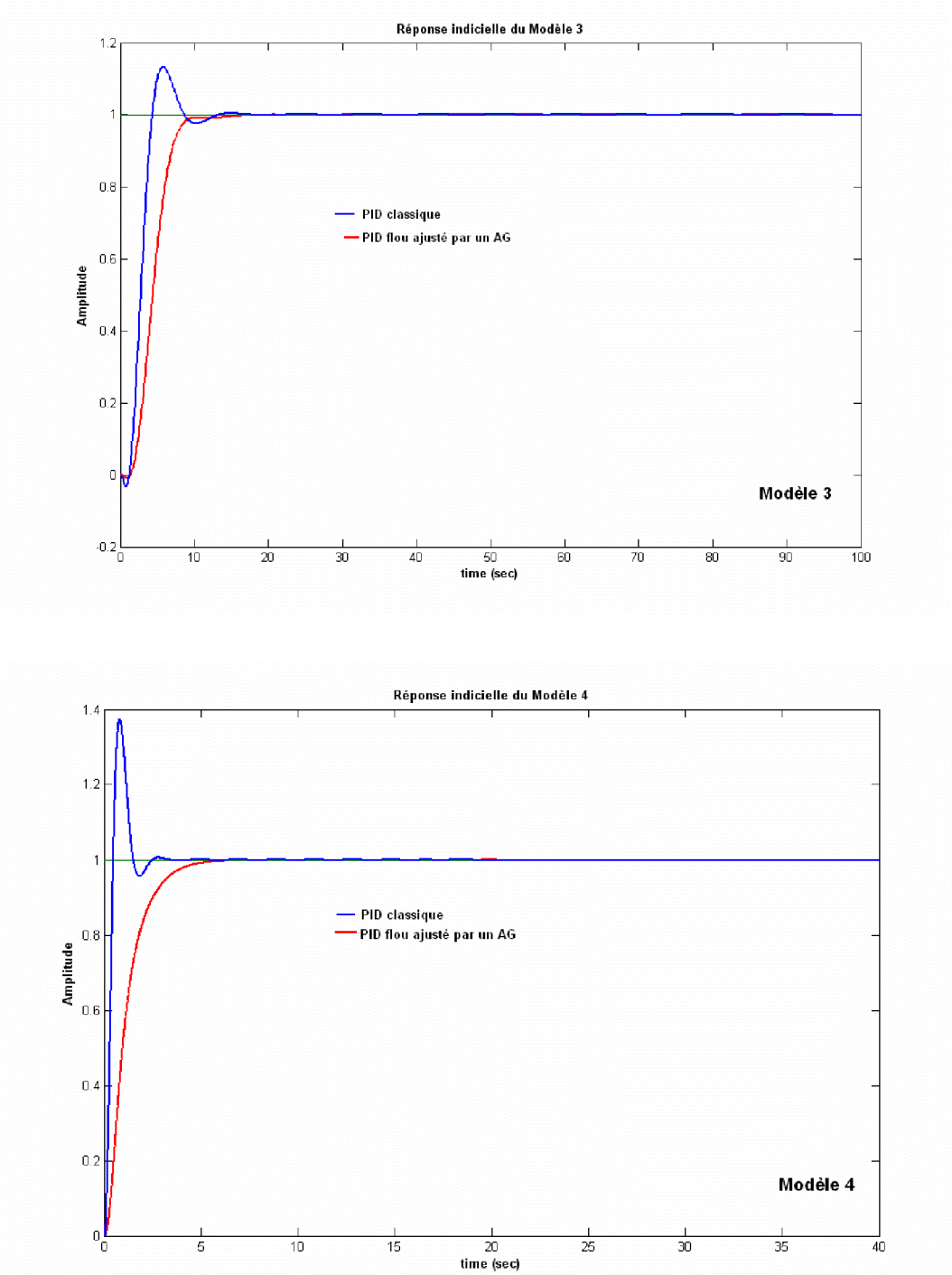
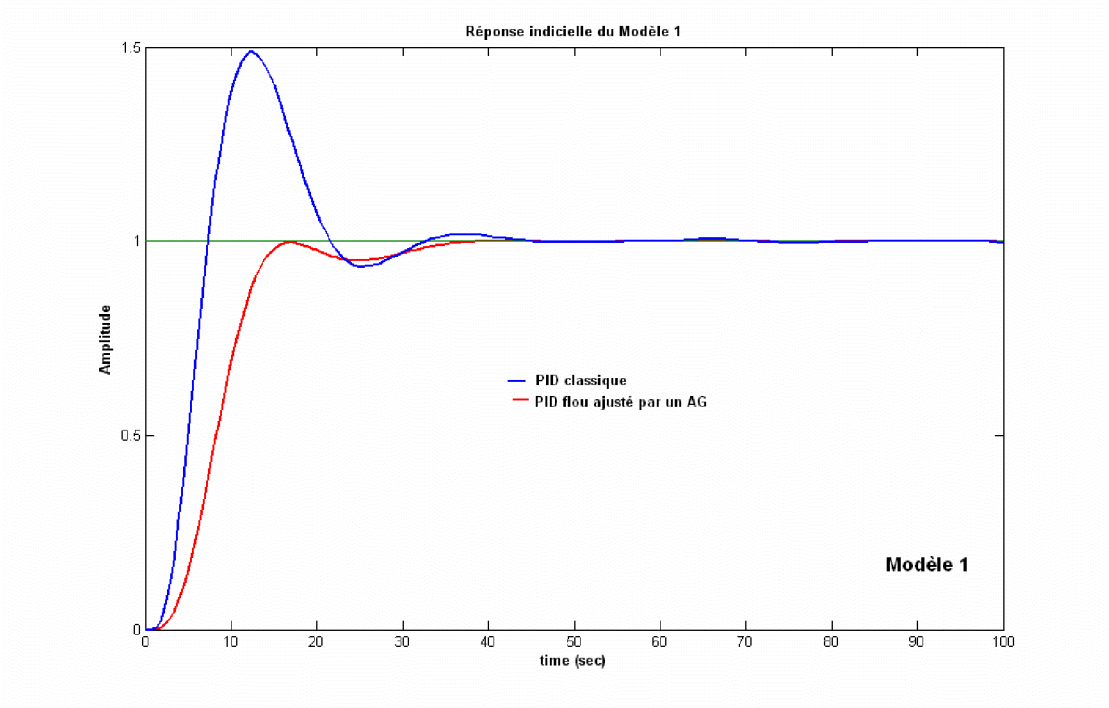



**Figure II.15:** Réponse indicielle du troisième et quatrième modèles réglés avec un régulateur PID classique (bleu) et par un régulateur PID flou ajusté par un AG pour 100 générations (rouge).

Les figures VI. 16 et VI.17 montre les réponses indicielles du première et deuxième modèles réglés par un régulateur PID classique (bleu) et un régulateur PID flou ajusté par un algorithme génétique après les 200 générations.



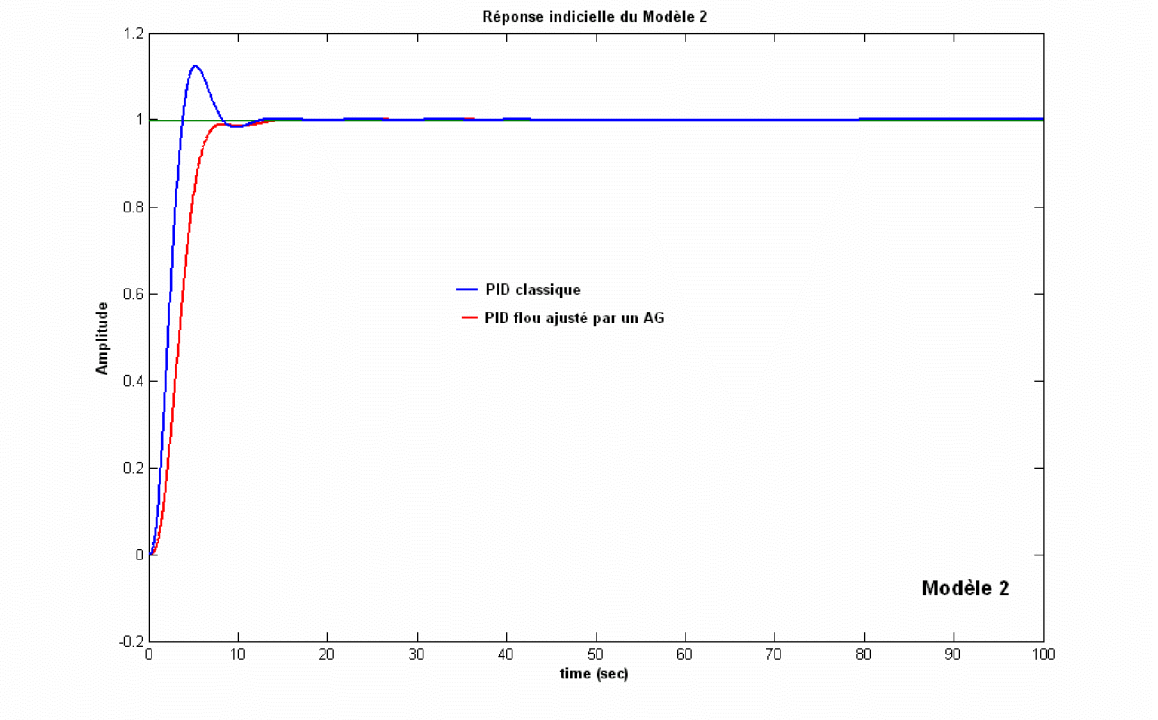




**Figure II.16:** Réponse indicielle du première et deuxième modèles réglés avec un régulateur PID classique (bleu) et par un régulateur PID flou ajusté par un AG pour 200 générations (rouge).

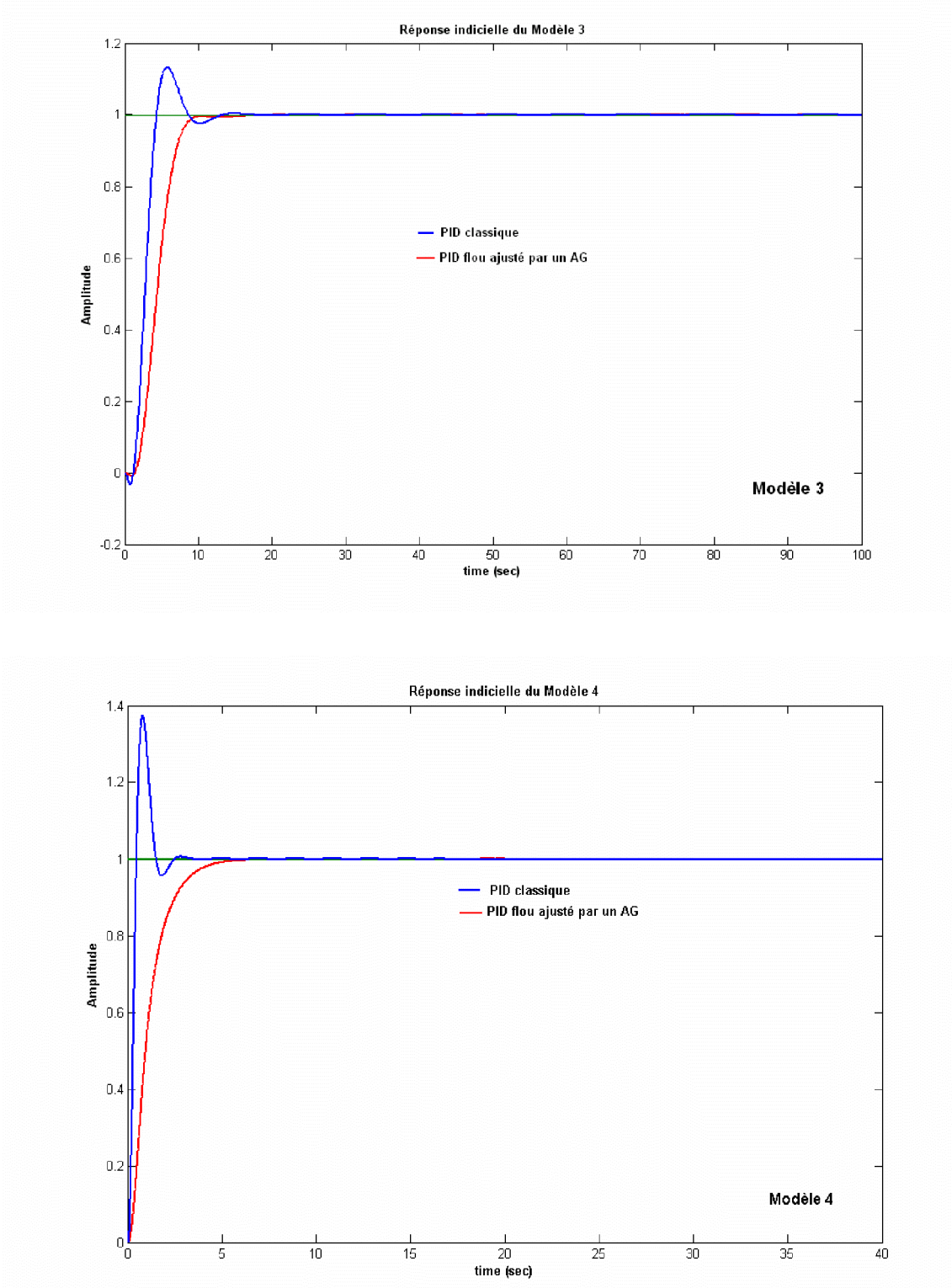



**Figure II.17:** Réponse indicielle du troisième et quatrième modèles réglés avec un régulateur PID classique (bleu) et par un régulateur PID flou ajusté par un AG pour 200 générations (rouge).

## VI.4 Analyse des résultats :

- Tout d'abord, d'après les résultats, on peut conclure que l'ajustement de la structure des fonctions d'appartenance des contrôleurs flous améliore les performances des systèmes et que le PID flou ajusté est plus performent que le PID classique.

- Dans une vue générale, quand l'ajustement des structures des fonctions d'appartenance est effectué, des contrôleurs flous plus précis sont obtenus.

- La comparaison entre le PID classique et le PID flou ajusté montre que le PID flou ajusté par les algorithmes génétiques est plus performant que le PID flou lui-même qui est aussi le plus performant que le PID réglée par une méthode classique.

- D'après les résultat, on peut remarque aussi que le PID ajusté par les algorithme génétiques a beaucoup de réponses plus rapide qu'utiliser une méthode classique. Mais on peur utiliser ses méthodes comme le point de départ de ce qui est les gains de régulateur PID.

- On remarque, que les résultat étaient bons et que le nombre de génération joue un rôle très important de point de vie diminution de dépassement, mais on remarque aussi que le MSE est très proche pour les 10, 50, 100 et 200 générations.

- L'inconvénient majeur qu'il faut noté est le temps consommé par la processus d'ajustement, mais on peut l'accepter puisque l'ajustement est effectué une seul fois pour chaque évaluation et ensuite on utilise les résultats obtenus pour la commande.

## VI.5 Conclusion :

Les résultats obtenus montrent, que même pour des différentes nombres de générations, les contrôleur flou obtenus par une méthode de génération des règles floues à partir d'une base de données numériques et optimisées ensuite par un processus basé sur les algorithmes génétiques, est plus performent par rapport a un régulateur PID ajusté par des méthodes classique et même par un méthode de synthèse à base de la logique floue.

# Conclusion Générale

## Conclusion générale :

Le travail mené dans ce mémoire a porté sur l'ajustement des paramètres de régulateurs PID flou. Afin d'assurer une description meilleures du régulateur PID est de divers méthodes de synthèse classique, nous avons proposé une autre méthode de synthèse basé sur les algorithmes génétiques.

Les algorithmes génétiques sont des algorithmes d'exploration robustes et convergent vers une solution satisfaisante, lorsque leurs paramètres (taille de la population, nombre d'itérations,…) sont choisis d'une manière adéquate. Cependant, le choix des paramètres ainsi que le codage des données est encore plus un art qu'une science. Il n'y a aucun jeu de paramètres qui serait universel pour tous les problèmes considérés du fait que ces valeurs dépendent étroitement du type de problème à résoudre.

Le troisième chapitre de ce mémoire est réservé à la description de la méthode de synthèse des régulateurs PID par un algorithme génétique. Nous avons tout d'abord décrit le principe des algorithmes génétiques, ensuite nous avons donnés en détaille la processus d'ajustement proposés pour l'optimisation des contrôleur flou. Durant la phase de synthèse, les différentes parties du contrôleur flou sont spécifiées grâce au procédé d'optimisation par algorithme génétique décrit dans le chapitre trois. Pour ce faire, on doit tout d'abord définir les codages utilisés pour représenter tous les paramètres du contrôleur que l'on souhaite optimiser. Ensuite, nous avons définir la fonction d'évaluation utilisé dans les algorithmes génétique.

Le quatrième chapitre est consacré à l'utilisation de cette méthode de synthèse d'un contrôleur flou. L'application décrite est le réglage des gains d'un régulateur standard PID. L'algorithme génétique nous a permis de définir au mieux la structure du contrôleur flou qui assure ce réglage. L'objectif de l'application des algorithmes génétiques sera de réduire au minimum l'erreur moyenne quadratique.

D'après les résultats obtenus, nous pouvons noter que le processus d'ajustement proposé améliore la précision, la robustesse et la flexibilité des contrôleurs PID flou, il a été montré aussi que l'usage cette technique qui représente en effet une solution précieuse pour la mise en ouvre d'un contrôleur performant en retenant en même temps le facilité totale d'usage d'un contrôleur PID.

## Perspectives :

Nous pensons que le travail décrit dans cette thèse peut se poursuivre principalement selon deux directions. Une première voie de recherche possible est l'amélioration de la méthode de synthèse du contrôleur flou. D'autre part, afin de valider complètement cette méthode, on doit tenter de l'appliquer pour résoudre d'autres problèmes de réglage dynamique des paramètres.

### Amélioration de la méthode de synthèse :

Plusieurs étapes de la méthode de synthèse décrite dans le chapitre trois peuvent certainement être améliorées. Par exemple, lors de la définition du contrôleur flou, de nombreuses caractéristiques doivent être choisies de façon intuitive : nombre d'entrées), nombre de termes linguistiques qualifiant les entrées ou les sorties.

Concernant ce dernier point, nous avons prévu dans le codage des paramètres du contrôleur flou une méthode pour supprimer éventuellement des termes linguistiques, ce qui permet théoriquement d'éliminer des termes inutiles. Sur les différents exemples que nous avons traités, cette possibilité n'a jamais été exploitée par le processus d'optimisation. Nous avons cependant souvent constaté que certains termes linguistiques deviennent inutiles dans la pratique car leur fonction d'appartenance prend des valeurs non nulles sur un intervalle très limité de l'univers du discours. Leur élimination totale simplifierait la structure du contrôleur flou sans incidence majeure sur les performances.

Nous pensons qu'il faudrait également préciser la façon de faire intervenir des connaissances a priori dans le processus de synthèse. Lorsque des règles sont connues par avance, nous n'en tenons pas compte dans le codage, ce qui fait que le processus optimise uniquement les règles initialement inconnues. Parfois, on aboutit à des contradictions entre les règles définies a priori et les règles issues de la synthèse. Pour éviter ce problème, on pourrait par exemple faire intervenir les règles connues dans la fonction d'évaluation, ce qui permettrait de défavoriser les règles contradictoires durant le processus d'optimisation.

### Autres applications :

La méthode de synthèse décrite dans ce travail peut être utilisée dans de nombreuses applications dans lesquelles les traitements dépendent du choix de nombreux paramètres. En particulier, dans le domaine du traitement des images numériques, on sait que les algorithmes

utilisent souvent des valeurs fixées par l'opérateur ou le programmeur : seuils, coefficients de pondération, etc.

# Référence Bibliographique

**Bibliographie:**

**[1]** K.J. Åström et T. Hägglund, *"PID Controllers: Theory, Design, and Tuning"*. Research Triangle Park, NC: Instrum. Soc. Amer. 1995.

**[2]** K.J. Åström and T. Hägglund, *"Automatic Tuning of PID Controllers"*. Research Triangle Park, NC: Instrum. Soc. Amer.1988.

**[3]** K.J. Åström and T. Hägglund, "*Automatic tuning of simple regulators with specifications on phase and amplitude margins*," Automatic*a*, vol. 20, pp. 645–651, 1984.

**[4]** L. Schmitt, *"L'Autoréglage des Asservissements en Position dans les Systèmes de Commande d'Axe Industriels "*, PhD thesis Institut Polytechnique de Grenoble, 1990.

**[5]** A. BESANÇON-VODA et I.D. LANDAU, *"Procédé et dispositif d'ajustement d'un régulateur PID"*, French patent No 95/05364, 1995.

**[6]** A. BESANÇON-VODA et H. ROUX-BUISSON, *"Another version for the feedback relay experiment"*, Journ. of Process Control, vol. 2, p. 240-246, 1997

**[7]** R. Longchamp et Y. Piguet. *"Closed-loop estimation of robustness margins by the relay method". IEEE ACC*, pages 2687–2691, 1995.

**[8]** T. S. Schei. *"Automatic tuning of PID controllers based on transfer function estimation". Automatica*, 30(12):1983–1989, 1994.

**[9]** G.H. Cohen and G.A. Coon, *"Theoretical considerations of retarded control*," *Trans. ASME,* vol. 75, pp. 827–834, 1953).

**[10]** K. Åström et T. Hägglund *"PID Controllers: Theory, Design, and Tuning"* (3rd ed) (2005).

**[11]** A. Leva et A. M. Colombo. *"On the IMC-based synthesis of the feedback block of the ISA PID regulators"*, Transactions of the Institute of Measurement and Control, 26(5):417–440, 2004.

**[12]** M. Morari and E. Zafirou. *"Robust Process Control"*. Prentice-Hall, Inc, Englewood Cliffs, N.J., 1989

**[13]** A. B. Voda et S. Gentil, *" régulateur PID analogiques et numériques"*, laboratoire d'automatique de Grenoble, R7 416,

**[14]** A. Voda et I. D. Landau, *"A method for the auto calibration of PID controllers"*, Automatica, vol. 31, p. 41-53, 1995.

**[15]** J.-J. E. Slotine and W. Li, *"Applied nonlinear control"*, Prentice-Hall, N.J., 1991.

**[16]** H.K. Khalil, *"Nonlinear Systems"*, Prentice Hall, 1996.

**[17]** A. El Hajjaji, A. Ciocan and D. Hamad, *"Four Wheel steering control by fuzzy*

*Approach",* Journal of intelligent and robotics systems, pp. 141-156, 2005.

**[18]** O. Pagès and A. El Hajjaji, *"Two fuzzy multiple reference model tracking control* designs *with an application to vehicle lateral dynamics control",* IEEE Conf. on Decision and Control, and the European Control Conference, pp. 3267-3272, 2005.

**[19]** Y. V. Kolokolov, S. L. Koschinsky, A. Hamzaoui, *"Comparative study of the dynamics and overall performance of boost converter with conventional and fuzzy control in application to PFC",* Power Elect. Spec. Conf., pp. 2165 – 2171, 2004.

**[20]** H. –X. Li, and H. B. Gatland, *"Conventional fuzzy control and its enhancement*", IEEE Trans. on sys. Man and Cyber, pp. 791-796, 1996.

**[21]** V. S. C. Raviraj and P. C. Sen, *"Comparative study of proportional-integral, sliding mode, and fuzzy logic controllers for power converters".* IEEE Trans. on Ind. App., pp. 518-524, 1997.

**[22]** J. Carvajal, G. Chen and H. Ogmen, *"Fuzzy PID controller: design performance evaluation and stability analysis",* Inter. Jour, of Information Sciences, pp. 249-170, 2000.

**[23]** R.R. Yager, S. Ovchinnikov, R.M. Tong, H.T. Nguyen (Ed), "*Fuzzy Sets and Applications* " : Selected Papers by L.A. Zadeh, J. Wiley, New York, 1987

**[24]** L.A. Zadeh, "*Fuzzy Sets, Information and control",* Vol. 8, pp. 29-44, 1965.

**[25]** E. Mamdani, " *An Expriment in Linguistic Synthesis with a Fuzzy Logic Controller*", International Journal on Man-Machine Studies, Vol. 7, pp. 1-13, 1975.

**[26]** J.J. Ostergaad. " *Fuzzy logic control of a heat exchange process".* In Fuzzy Automata and Decision Processes, M.M. Gupta, G.N. Saridis, and B.R. Gaines, Eds., pages 285–320, 1977.

**[27]** D.Willaeys, N. Malvache, et P. Hammad. " *Utilization of fuzzy sets for systems modelling and control* ". Dans IEEE International Conference on Decision and Control, 1977.

**[28]** D. Willaeys, P. Mangin, et N. Malvache. " *Use of fuzzy sets for systems modelling and control : Application to the speed control of a strongly perturbed motor*". Dans IFAC/IFIP International Conference on Digital Computer Applications to Process Control, 1997.

**[29]** D. Willaeys et N. Malvache. " *Use of fuzzy model for process control*". Dans IEEE International Conference on Cybernetics and Society, 1978.

**[30]** M. Sugeno et K. Murakami. " *Fuzzy parking control of model car* ". In 23rd IEEE Conf. on Decision and Control, 1984.

**[31]** J.B. Kiszka, M.M. Gupta, et P.N. Nikiforuk. " *Energetic stability of fuzzy dynamic systems*". IEEE Trans. on Syst. Man Cybern., SMC-15(5) :783–792, 1985.

**[32]** M. Togai et H. Watanabe. " *Expert system on a chip : An engine for real-time approximate reasoning"*. IEEE Expert Syst. Mag., 1 :55–62, 1986.

**[33]** T. Tamakawa. " *High speed fuzzy controller hardware system*". Proc. 2nd Fuzzy System Symp., pages 122–130, 1986.

**[34]** H.Bûchler. *"Réglage par logique floue"*, Presses Polytechniques et universitaires Romandes, Losanne, p :1-2, 1994

**[35]** B. Bouchon-Meunier. *"La logique floue"*. Presses Universitaires de FRANCE, 1993.

**[36]** D. Dubois et H. Prade. *"Principes de base de la théorie des ensembles flous et enjeux liés à la logique floue"*. Dans Les Perspectives et Applications Industrielles de la Logique Floue en Europe, 1994.

**[37]** B. Bouchon-Meunier. *"La logique floue et ses applications"*. Addison-Wesley France, 1995.

**[38]** J.K. George et B. Yuan. *"Fuzzy Sets and Fuzzy Logic: Theory and Applications."* Englewood Cliffs, N.J. Prentice-Hall, 1995.

**[39]** N. Cordier, *" development et evaluation de strategie de contrôle de ventilation appliquées aux locaux de grandes dimensions"*, Institut National Des Sciences Appliquées de Lion, p. 69-71, 2007.

**[40]** H. Ying, "*Fuzzy control and modeling , analytical fundations and applications"*, IEEE Press, NJ, 2000.

**[41]** C.C. Lee. *"Fuzzy logic in control systems : fuzzy logic controller"* part i and part ii. IEEE Transactions on Systems, Man and Cybernetics, vol 20:404–418, 419–435, 1990.

**[42]** K. Guesmi, *" Contribution a la commande floue d'un convertisseur statique"*, université de Reims Champagne Ardenne, p : 79, 2006.

**[44]** A. Rubaai, J. Castro-Sititiche et A. Rofoli. *"Design and Implementation of parallel fuzzy PID controller for higt performance brushless motor drives: An environment for rapid control prototyping"*. IEEE Transactions on industry applications, vol. 44, N. 4, pp 10901098. July/August 2008.

**[45]** M. H. Hung, L.S. Sun, S.J. Ho, S.F. Hwang, S.Y. Ho. *"A novel intelligent multiobjective simulated annealing algorithm for designing robust PID controllers"*, IEEE transactions on systems, MAN and cybernetics PART A: systems and Humants, vol. 38, N.2, pp. 319-330, March 2008.

**[46]** K.A. De Jong, *"An analysis of the behavior of a class of genetic adaptive system"*, ph, D.Dissertation 76-9381, Univercity of Michigan, 1975.

**[47]** D.E. Golderg, *" Genetic Algorithms in Search, optimization, and machine learning"*, Addison-Wesley, new york, 1989.

**[48]** D. Nauuck, F. Klawonn, et R. Kruse, *"fundations of neuro-fuzzy systems"*, new York: Wiley, 1997

**[49]** O. Cordon, F. Herrera, F. Hoffman, et L. Magdalena, *"genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge base*s", Singapore; world scientific,2001.

**[50]** J. Casillas, O. Cordon, F. Herrera et L. Magdalene, Eds., *"intereprebility issues in fuzzy modelling"*. Heidelberg, Germany: Springer- verlag, 2003.

**[51]** S. Aouaouda, *"Optimisation du contrôleur flou par algorithms génétiques: Application à un robot mobile"*. Mémoire de magister en automatique industrielle, LASA, Annaba, 2006.

**[52]** M. Kumar, P.G. Devendra, *"intelligent learning of fuzzy logic controllers via neural network and genetic algorithm"*, proceeding of symposium on flexible automation usa, 2004.