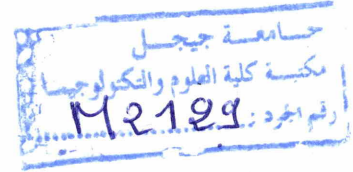


République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur

de la Recherche Scientifique



Université de Jijel

Faculté des Sciences et de la Technologie

Département d'Electronique

Mémoire de fin d'Etude pour l'Obtention du Diplôme de

Master II en Electronique

*Option :*

Electronique et Analyse des Systèmes

*Thème :*

***Conception de la base de règles Floues  
en utilisant l'Algorithme d'Optimisation  
Par Essaim Particulaire***

**Encadré par :**

Dr : Nesrine TALBI

**Présenté Par :**

M<sup>elle</sup> : Sarra MEGHRICHE

M<sup>elle</sup> : Fatima BOUNAAS

Promotion : Juin 2015

# Remerciements

Nous remercions *Allah* le tout puissant qui nous a donné la force, la volonté et le courage pour accomplir ce modeste travail.

Nous tenons à remercier *Dr N. Talbi* pour la proposition du sujet de ce mémoire ainsi pour les conseils bénéfiques en vue du bon déroulement du travail durant la réalisation de ce projet.

Nous remercions l'ensemble des membres de *jury* qui nous ont fait l'honneur d'accepter de lire et de juger ce mémoire.

Nous remercions ainsi l'ensemble des *collègues* et *amis* de leurs support et soutien durant nos études.

*\*\*Sarra et Fatima\*\**

# Dédicaces

Je dédie ce travail à tous ceux qui m'ont aidé à réaliser ce travail.

\* *A ma mère*

J'espère qu'elle le trouve récompensant de tous les sacrifices qu'elle a consentis pour moi.

\* *A mon père*

En témoignage de ma sincère gratitude pour ses sacrifices qu'il a consentis pour moi et pour toute la famille.

\* *A mon frère : Zidane.*

\* *A mes sœurs : Nahla et Chaima.*

\* *A mon binôme Fatima et sa famille.*

\* *A tous mes amis en témoignage de mon amitié et mon*

*attachement surtout : Mouna, Rafika et Loubna.*

\* *Sarra* \*

# Dédicaces

Je dédie ce travail à tous ceux qui m'ont aidé à réaliser ce travail.

\*A *mon père*

En témoignage de ma sincère gratitude pour ses sacrifices qu'il a consentis pour moi et pour toute la famille.

\*A *ma mère*

J'espère qu'elle le trouve récompensant de tous les sacrifices qu'elle a consentis pou moi.

\*A *mes frères : Adlan, Tarik, Ahmed et Omar.*

\* A *ma sœur : Chafia et son mari.*

\* A mon neveu : *Monsif.*

\* A mon binôme *Sarra et sa famille.*

\*A tous ceux que j'aime, et qui m'aime.

*\*Fatima\**

## TABLE DES MATIERES

<b>Introduction Générale.....</b>	<b>1</b>
<b>Chapitre I: La commande floue</b>	
<b>I.1 Introduction.....</b>	<b>4</b>
<b>I.2 La logique floue .....</b>	<b>5</b>
I.2.1 Définition .....	5
I.2.2 Historique .....	5
<b>I.3 Le formalisme de la logique floue .....</b>	<b>6</b>
I.3.1 Variables linguistiques .....	6
I.3.2 Fonction d'appartenance.....	7
I.3.3 Le concept d'ensemble floue .....	8
I.3.3.1 Caractéristiques d'un sous-ensemble floue .....	8
I.3.3.2 Les opérations sur les ensembles flous .....	9
<b>I.4 La Commande floue .....</b>	<b>11</b>
I.4.1 Introduction .....	11
I.4.2 Définition .....	12
I.4.3 Caractéristiques de la commande floue .....	12
I.4.4 Structure d'une commande floue .....	13
I.4.4.1 Interface la fuzzification .....	13
I.4.4.2 Interface d'inférence floue .....	14
I.4.4.3 Interface de défuzzification .....	14
I.4.5 Différents types de contrôleurs flous .....	15
I.4.5.1 Contrôleur flou type Sugeno .....	15
I.4.5.2 Contrôleur flou type Mamdani .....	16
I.4.6 Les domaines d'application .....	17
I.4.7 Avantages et inconvénients de la commande floue .....	17
I.4.7.1 Avantages .....	17
I.4.7.2 Inconvénients .....	18
<b>I.5. Conclusion .....</b>	<b>18</b>



---

**Chapitre II : L'optimisation par Essaim Particulaire**

<b>II.1 Introduction</b> .....	19
<b>II.2 Méta-heuristique</b> .....	20
II.2.1 Définition .....	20
II.2.2 Principe de méta-heuristique .....	21
II.2.3 Différents types de méta-heuristiques .....	21
II.2.3.1 Le méta-heuristique à population .....	22
II.2.3.2 Le méta-heuristique à parcours .....	22
<b>II.3 L'optimisation par essaim de particule</b> .....	23
II.3.1 Introduction .....	23
II.3.2 Définition .....	23
II.3.3 principe de déplacement d'une particule .....	24
II.3.4 Les caractéristiques de l'OEP .....	25
II.3.4.1 Nombre de particule .....	26
II.3.4.2 La position, la vitesse .....	26
II.3.4.3 Facteur d'inertie .....	26
II.3.5 Voisinage .....	27
II.3.5.1 Notion .....	27
II.3.5.2 Les différents types de topologie .....	28
II.3.6 Algorithme standard de L'OEP .....	30
II.3.7 Critères d'arrêt .....	31
II.3.7 Avantages et inconvénients .....	31
<b>II.4 Conclusion</b> .....	32

**Chapitre III : Simulation & Interprétation**

<b>III.1 Introduction</b> .....	33
<b>III.2 Le modèle mathématique du bain d'eau</b> .....	34
<b>III.3 La commande floue du bain d'eau</b> .....	35
III.3.1 Le type de Contrôleur flou utilisé .....	35
III.3.2 Structure de base du contrôleur flou .....	36

Liste des figures

<b>Fig I.1 : Variable linguistique</b> .....	6
<b>Fig I.2 : Différents types des fonctions d'appartenances</b> .....	8
<b>Fig I.3 : Les caractéristiques d'un ensemble flou</b> .....	9
<b>Fig I.4 : Exemple d'opérations sur des ensembles flous</b> .....	11
<b>Fig I.5: Structure générale d'une commande floue</b> .....	13
<b>Fig II.1 : Principe générale des méta-heuristiques</b> .....	21
<b>Fig II.2 : Classification des méthodes d'optimisation</b> .....	22
<b>Fig II.3 : Déplacement d'une particule</b> .....	25
<b>Fig II.4 : Le voisinage géographique</b> .....	27
<b>Fig II.5 : Le voisinage en cercle</b> .....	28
<b>Fig II.6 : Topologie en étoile</b> .....	29
<b>Fig II.7 : Topologie en anneau</b> .....	29
<b>Fig II.8 : Topologie en rayon</b> .....	30
<b>Fig III.1 : Structure du procédé</b> .....	34
<b>Fig III.2 : Structure du contrôleur flou à optimiser</b> .....	35
<b>Fig III.3 : Forme des fonctions d'appartenances</b> .....	37
<b>Fig III.4 : Evolution de la température et la référence en fonction du temps</b> .....	38
<b>Fig III.5: Evolution signal de contrôle, erreur et dérivé d'erreur</b> .....	38
<b>Fig III.6 : Test de robustesse en changeant la trajectoire de référence</b> .....	39
<b>Fig III.7: Structure du contrôleur floue de la température avec l'algorithme</b> .....	41
<b>Fig III.8: Organigramme de la méthode des essaims particulaires</b> .....	42
<b>Fig III.9: Evolution de la fonction cout</b> .....	43
<b>Fig III.10 : Disposition et formes des fonctions d'appartenance des prémisses et des conclusions après optimisation</b> .....	44
<b>Fig III.11 : Evolution de la température et la référence</b> .....	45

<b>Fig III.12</b> :	Evolution de la commande et l'erreur de poursuite de la température.....	45
<b>Fig III.13</b> :	Test de robustesse en changeant la trajectoire de référence.....	47
<b>Fig III.14</b> :	Evolution de la commande et l'erreur de poursuite de la température.....	46



**Liste des tableaux**

<b>Tab III.1 :</b> Paramètres du système.....	35
<b>Tab III.2 :</b> Table des règles du contrôleur flou (333).....	37
<b>Tab III.3 :</b> Paramètres des prémisses optimaux obtenus par l'OEP.....	43
<b>Tab III.4 :</b> Paramètres des conclusions de règles floues optimaux obtenus par l'OEP.....	43



## Introduction Générale

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 existants dans la littérature proposent des approches qui sont, généralement, limitées à des formes bien particulières de systèmes 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.

Les systèmes flous, basés sur la théorie de la logique floue, ont été utilisés comme alternative pour construire de telles structures multi-modèles. Ils présentent l'avantage de tolérer l'incertitude du modèle et compensent son effet, de traiter les non linéarités sans aucune hypothèse sur leur nature, de modéliser puis compenser les interactions entre les boucles et de réduire l'effet des perturbations externes. Leur qualité d'approximateurs universels a été démontrée, notamment pour les applications en commande et en identification floue. On distingue deux classes principales de modèles flous : les modèles flous de Mamdani et ceux de Takagi-Sugeno (TS). Ils diffèrent au niveau de la conclusion.

Les systèmes flous sont capables de modéliser des systèmes qui sont quelque

fois impossible à les décrire par un formalisme mathématique. Cependant, le choix optimal des paramètres du système flou (les fonctions d'appartenance, les règles floues, les gains, etc.....) représentent une tâche très difficile. Pour cela, les chercheurs ont développées plusieurs méthodes optimales et systématiques pour la conception des systèmes flous. Parmi ces méthodes, il y a les métaheuristiques. Ces méthodes d'optimisation sont des heuristiques à large domaine d'application par opposition aux heuristiques simples développées pour résoudre un problème particulier. Plusieurs méta-heuristiques s'inspirent des systèmes biologiques.

Les méta-heuristiques forment une famille d'algorithmes d'optimisation visant à résoudre des problèmes d'optimisation difficile, pour lesquels nous ne connaissons pas de méthodes classiques plus efficaces. Les méta-heuristiques peuvent être considérées comme des algorithmes stochastiques itératifs, où elles manipulent une ou plusieurs solutions à la recherche de l'optimum global.

Les méta-heuristiques sont souvent inspirées de processus naturels qui relèvent de la physique (l'algorithme du recuit simulé), de la biologie de l'évolution (les algorithmes génétiques) ou encore de l'éthologie (les algorithmes de colonies de fourmis et l'**optimisation par essaim particulaire**).

L'objectif de notre travail est la Conception de la base de règles Floues en utilisant l'Algorithme d'Optimisation par Essaim Particulaire. Pour ce faire, ce mémoire est organisé comme suit :

***Le chapitre I*** est consacré à décrire les bases de la logique floue, allant du concept des ensembles flous et les différents opérations nécessaire à la commande floue. Ensuite, la structure d'une commande floue (fuzzification, inférence floue et la défuzzification), ainsi que leurs caractéristiques sont établis, puis les deux types de contrôleurs flous les plus utilisé (Mamdani et Takagi-Sugeno) sont présentés, finalement le chapitre est clôturé par les avantages et les inconvénients de la logique floue.

*Le chapitre II* a pour objet de présenter une introduction aux méta-heuristiques et leurs principes, notions de bases, ensuite l'algorithme d'optimisation par essaim particulaire et ses différentes caractéristiques est détaillé. A la fin, les avantages et les inconvénients de la méthode sont mentionnés.

*Le chapitre III* est consacré dans un premier lieu, à la mise au point d'un régulateur flou de type Takagi-Sugeno d'ordre zéro pour la commande de la température d'un bain d'eau, la deuxième partie traite l'optimisation des paramètres du contrôleur flou de type Takagi-Sugeno d'ordre zéro par l'algorithme d'optimisation par essaim particulaire. Nous intéressons aux valeurs modales des fonctions d'appartenance des entrées du contrôleur flou ainsi les conclusions des règles floues, l'optimisation consistera donc à minimiser un critère d'erreur entre la consigne et la sortie du système. Le chapitre illustre les réponses du bain d'eau et le teste de robustesse de ceux-ci aux changements de leurs trajectoires de références de température désirée

Ce mémoire est clôturé par une conclusion générale.

# Chapitre I

## La Commande Floue

### I.1 Introduction [1]

A partir de concepts mathématiques relativement simples, la logique floue permet de résoudre des problèmes complexes là où les outils de résolution conventionnels sophistiqués peinent. Son succès réside principalement dans son potentiel à résoudre des problèmes du monde réel. L'industrie et les services peuvent bénéficier des atouts de la logique floue, la panoplie des applications à l'échelle planétaire en témoigne. Tous les problèmes de contrôle, de diagnostic ou d'aide à la décision sont des clients potentiels de la logique floue. Ainsi, La logique floue s'est imposée dans des domaines aussi variés que l'électrotechnique, l'industrie, l'automatisme, la robotique, la gestion de la circulation routière, le contrôle aérien, l'environnement, la médecine, afin de résoudre les problèmes d'identification, de régulation de processus, d'optimisation, de classification, de détection de défauts ou de prise de décision.

Le développement des algorithmes flous se fait à travers les méthodes par lesquelles l'homme essaye de copier la nature et de reproduire des modes de raisonnement et de comportement qui lui sont propres.

L'objectif de ce chapitre est de présenter les notions principales de la théorie de la logique floue, ainsi que de la commande floue. D'abord, quelques concepts de

base de la logique floue ainsi que leurs applications pour la synthèse d'une commande floue ont été présentés. Les éléments de bases d'un régulateur par logique floue, les différents types de régulateurs flous, ainsi que leurs avantages et leurs inconvénients.

## I.2 La logique floue

### I.2.1 Définition [2]

La logique floue apparaît comme une logique graduelle qui peut être très proche de notre perception nuancé du monde. Formellement, la théorie du flou définit une interface entre le qualitatifs /symbolique et le quantitatifs/numérique, pratiquement elle offre une approche élégante à la résolution de problèmes, multidimensionnels et complexes caractérisés par une forte interactivité des parties, faisant intervenir l'homme à la fois comme capteur et comme décideur/actionneur.

### I.2.2 Historique [3]

Les prémisses de la logique floue sont apparues en 1940 lorsque des chercheurs américains ont commencé à se pencher sur le concept d'incertitude. En 1965, Lofti Zadeh introduit la théorie des ensembles flous, (*fuzzy set theory*). En 1973 le professeur Zadeh publie un article qui mentionne pour la première fois le terme de valeurs linguistiques dont la valeur est un mot du langage naturel.

En 1974, Mamdani établit la première application industrielle de la logique floue : la régulation floue d'une chaudière à vapeur. Il introduit ainsi la commande floue dans la régulation industrielle.

Grâce au chercheur Japonais M. Sugeno, la logique floue a été introduite au Japon dès 1985. Les sociétés japonaises comprirent l'avantage à la fois technique et commercial de cette logique :

- facilité d'implantation.
- solution de problèmes multi-variables complexes.

- robustesse vis à vis des incertitudes.
- possibilité d'intégration du savoir de l'expert.

L. Zadeh a établi en 1978 le lien entre la théorie des ensemble flous et la modélisation non probabiliste de l'incertitude grâce à la théorie des possibilités, les ensembles flous et la théorie des possibilités.

Les ensembles flous et la théorie des possibilités sont à la base d'une méthodologie de représentation et d'utilisation de connaissances imprécises, que zadeh appelle théories du raisonnement approché, plus connue sous le vocable de « la logique floue ».

### I.3 Le formalisme de la logique floue

#### I.3.1 Variables linguistiques [4]

Une variable linguistique sert à modéliser les connaissances imprécises ou vagues sur une variable dont la valeur précise peut être inconnue. La variable linguistique est caractérisée par un triplet  $(V; XV; TV)$  tel que :

1.  $V$  est le nom de la variable linguistique (âge, taille, masse, ... ) ;
2.  $XV$  est l'univers des valeurs pouvant être, éventuellement, prises par  $V$ .
3.  $TV$  est un ensemble de sous-ensembles flous de  $XV$  qui caractérise  $V$ .

La figure suivante illustre un exemple de la variable (température) avec une trois termes linguistique : Froide, Moyenne, Chaude.

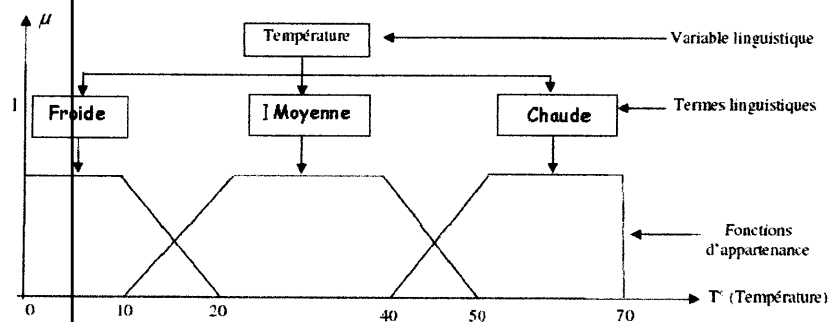


Fig I.1: Variable linguistique.

### I.3.2 Fonction d'appartenance [5]

Au lieu d'appartenir à l'ensemble « vrai » ou à l'ensemble « faux » de la logique binaire traditionnelle, la logique floue admet des degrés d'appartenance à un ensemble donné. Le degré d'appartenance à un ensemble flou est matérialisé par un nombre compris entre 0 et 1. Une valeur précise de la fonction d'appartenance liée à une valeur de la variable est notée  $\mu$  et appelée « facteur d'appartenance ».

Le plus souvent, la fonction d'appartenance est déterminée par l'une des fonctions suivantes :

➤ **Fonction triangulaire**

Elle est définie par trois paramètres  $\{a, b, c\}$ , qui déterminent les coordonnées des trois sommets (figure I.2-a)

$$\mu(x) = \max \left( \min \left( \frac{x-a}{b-a}, \frac{c-x}{c-b} \right), 0 \right) \quad (\text{I.1})$$

➤ **Fonction trapézoïdale**

Elle est définie par quatre paramètres  $\{a, b, c, d\}$  (figure I.2-b)

$$\mu(x) = \max \left( \min \left( \frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right) \quad (\text{I.2})$$

➤ **Fonction gaussienne**

Elle est définie par deux paramètres  $\{\sigma, m\}$  (figure I.2-c)

$$\mu(x) = \exp \left( -\frac{(x-m)^2}{2\sigma^2} \right) \quad (\text{I.3})$$

➤ **Fonction sigmoïde**

Elle est définie par deux paramètres  $\{a, c\}$  (figure I.2d)

$$\mu(x) = \frac{1}{1 + \exp(-a(x-c))} \quad (\text{I.4})$$



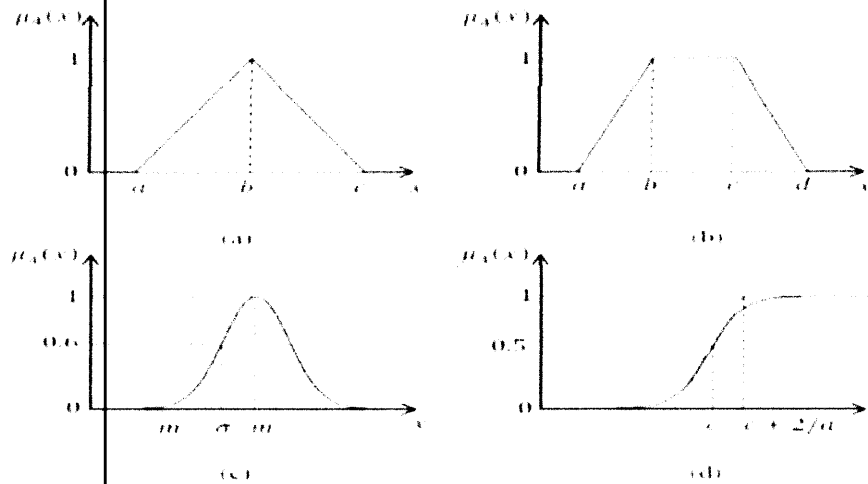


Fig I.2 : Différents types des fonctions d'appartenance.

### I.3.3 Le concept d'ensemble flou [6]

Un ensemble flou est défini comme suit : soit  $S$  un ensemble et  $x$  un membre de cet ensemble. Un sous-ensemble flou  $A$  de  $X$  est défini par une fonction d'appartenance  $A(x)$  qui mesure le degré auquel  $x$  appartient à  $A$ .

#### I.3.3.1 Caractéristiques d'un sous-ensemble flou [7]

Un sous-ensemble flou est complètement défini par la donnée de sa fonction d'appartenance. A partir d'une telle fonction, un certain nombre de caractéristiques du sous-ensemble flou peuvent être étudiées.

- **Le type** : les fonctions d'appartenance peuvent être triangulaire, gaussienne, trapézoïdales, sigmoïdale ou de type pic 'singleton', ....etc.
- **Support et Hauteur** : Ces deux caractéristiques montrent, dans quelle mesure un sous- ensemble flou  $A$  de  $x$  diffère d'un sous-ensemble classique de  $x$ .

a)- Le support d'un sous-ensemble flou  $A$  de  $x$ , noté  $supp(A)$ , est l'ensemble de tous les éléments qui lui appartiennent, la figure(I.3) représente les différents types de support. Formellement:

$$\text{supp}(A) = \{x \in X | \mu_A > 0\} \quad (\text{I.5})$$

b)- La hauteur d'un sous-ensemble flou  $A$  de  $x$ , notée  $h(A)$ , est la valeur maximale de la fonction d'appartenance, Formellement:

$$h(A) = \sup_{x \in X} \mu(x) \quad (\text{I.6})$$

➤ **Noyau** : Un sous-ensemble flou est normalisé si sa hauteur  $h(A) = 1$ . Le noyau d'un sous-ensemble flou  $A$  de  $X$ , noté  $\text{Noy}(A)$ , est l'ensemble de tous les éléments qui lui appartiennent totalement (avec un degré 1). Formellement :

$$\text{Noy}(A) = \{x \in X | \mu_A(x) = 1\} \quad (\text{I.7})$$

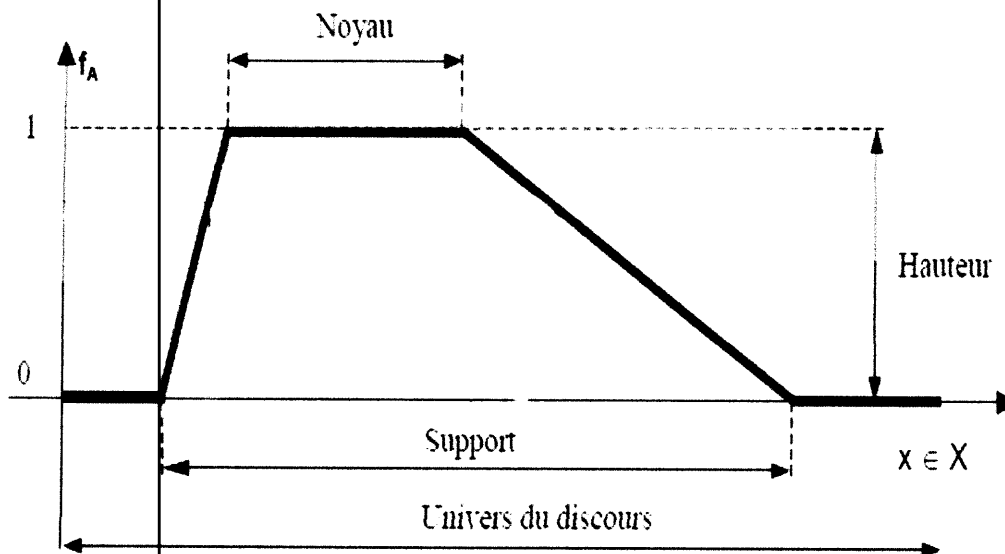


Fig I. 3 : Les caractéristiques d'un ensemble flou.

### I.3.3.2 Les opérations sur les ensembles flous [8]

Disposant d'une représentation formelle de classes floues sous forme de fonctions d'appartenance nuancées, il est naturel de chercher à faire avec des ensembles classiques.

Zadeh a aussi généralisé les opérations d'égalités, d'inclusion, de complémentation, d'union et d'intersection aux ensembles flous.

Supposons que  $A$  et  $B$  sont deux ensembles flous définis dans un univers de discours  $X$  par les fonctions d'appartenance  $\mu_A$  et  $\mu_B$ .

❖ **Egalité** :  $A$  et  $B$  sont dits égaux, propriété que l'on note  $A = B$ , si leurs fonctions d'appartenance prennent la même valeur en tout point de  $X$  :

$$\forall x \in X \mu_A(x) = \mu_B(x) \quad (\text{I.8})$$

❖ **Inclusion** :  $A$  est dit inclus dans  $B$ , propriété que l'on note  $A \subseteq B$ , si tout élément  $x$  de  $X$  qui appartient à  $A$  appartient aussi à  $B$  :

$$\forall x \in X \mu_A(x) \leq \mu_B(x) \quad (\text{I.9})$$

❖ **Intersection** : L'intersection de  $A$  et  $B$ , que l'on note  $A \cap B$ , est l'ensemble flou constitué des éléments de  $X$  affectés du plus petit des deux degrés d'appartenance  $\mu_A$  et  $\mu_B$  :

$$\forall x \in X \mu_{A \cap B}(x) = \min(\mu_A(x); \mu_B(x)) \quad (\text{I.10})$$

❖ **Union** : L'union de  $A$  et  $B$ , que l'on note  $A \cup B$ , est l'ensemble flou constitué des éléments de  $X$  affectés du plus grand des deux degrés d'appartenance  $\mu_A$  et  $\mu_B$  :

$$\forall x \in X \mu_{A \cup B}(x) = \max(\mu_A(x); \mu_B(x)) \quad (\text{I.11})$$

Dans ces définitions, min et max désignent, respectivement, l'opérateur de calcul du minimum et du maximum des deux valeurs.

❖ **Complément** : Le complément de  $A$  que l'on note  $\bar{A}$ , est l'ensemble flou de  $X$  constitué des éléments  $x$  lui appartenant d'autant plus qu'ils appartiennent peu à  $A$  :

$$\forall x \in X \mu_{\bar{A}} = 1 - \mu_A(x) \quad (\text{I.12})$$

La figure suivante résume ces opérations, de façon graphique.

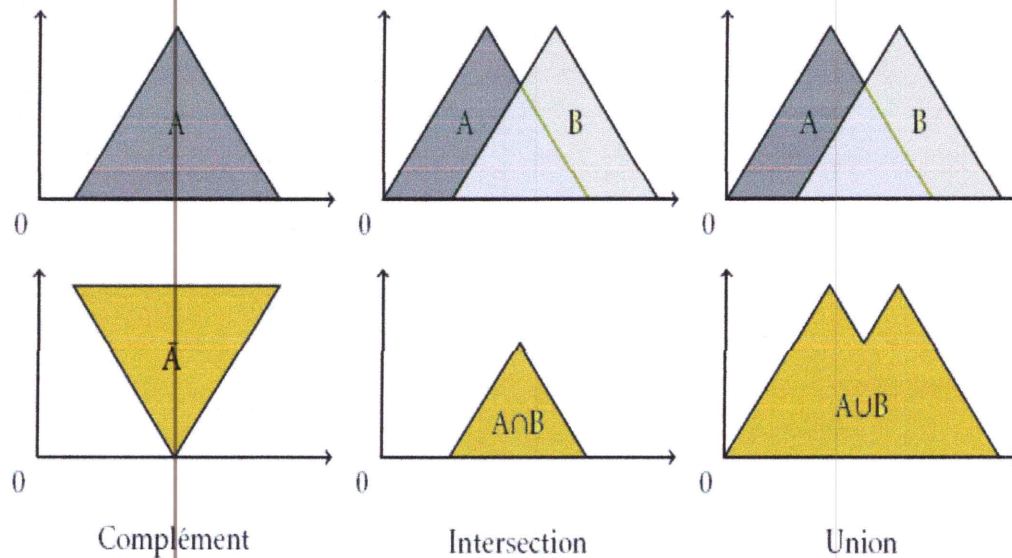


Fig I.4 : Exemple d'opérations sur des ensembles flous.

## I.4 La commande floue

### I.4.1 Introduction [9]

La commande floue est le domaine d'application de la logique floue le plus utilisé dans le cadre de la commande des systèmes non linéaires complexes. L'approche utilisée repose sur la propriété d'approximation universelle des systèmes flous. Cette approche de commande floue a pour objectif le traitement des problèmes de commande de processus qui sont, soit difficilement automatisables par une approche classique, soit les sources d'information sont jugées non précises ou incertaines, et cela à partir des connaissances des experts ou d'opérateurs qualifiés travaillant sur le processus.

La particularité de cette commande est de reproduire le comportement humain plutôt que de réaliser un modèle mathématique du système et le contrôleur flou peut ainsi être vu comme un algorithme qui peut convertir une stratégie formelle de

commande basée sur les connaissances d'un expert en une stratégie automatique de commande. Cet algorithme de commande se base sur une collection de règles floues appelée base de règles. L'ensemble des règles de la commande sont reliées par les concepts d'implication, de composition floue et des règles d'inférence floue.

#### **I.4.2 Définition [10]**

Une commande basée sur les techniques de la logique permet d'une part, de formaliser le traitement de l'inconsistance et de l'incertitude sur le modèle du système et d'autre part, d'établir une interprétabilité du comportement dynamique du système par un ensemble de règles linguistiques, traduites en langage simple et interprétable.

Son principe a été introduit par Mamdani et Assilian sur l'exemple d'une machine à vapeur expérimentale. Depuis il a été exploité dans plusieurs domaines comme la commande d'une usine d'eau chaude, échangeur de chaleur, régulation de vitesse d'un moteur, four à ciment. La commande floue a été aussi largement développée au Japon au début des années 80.

#### **I.4.3 Caractéristiques de la commande floue [11]**

La spécificité de la commande floue réside dans les points suivants :

- La connaissance mathématique du fonctionnement du processus n'est pas nécessaire. C'est le savoir faire de l'opérateur qui est pris en compte.
- Des variables subjectives sont utilisables. Il est possible de modéliser les sens humains.

L'intérêt de la commande floue apparaît clairement dans le cas de système mal connu ou difficile à décrire. De même manière lorsque les variables sont décrites de façon imprécise ou en langage naturel. La commande floue:

☞ est simple à réaliser, facilement adaptable aux conditions de fonctionnement.

La plupart du temps un petit nombre de règle suffit à décrire le système.

☞ Permet La coordination de plusieurs objectifs (système multivariable).

☞ est reconnue comme robuste.

Le principe de la commande floue consiste à caractériser le mieux possibles les variables qui agissent sur le processus en fonction de tous les états du processus à commander et des avis et recommandations des experts. Elle traite des problèmes de commande de processus à partir uniquement de connaissances de comportement que les spécialistes du procédé doivent formuler sous forme linguistique (floue).

#### I.4.4 Structure d'une commande floue [12] [13]

La structure d'une commande floue, présentée sur la figure suivante peut être décomposée en trois grands blocs : La fuzzification, L'inférence floue et la défuzzification.

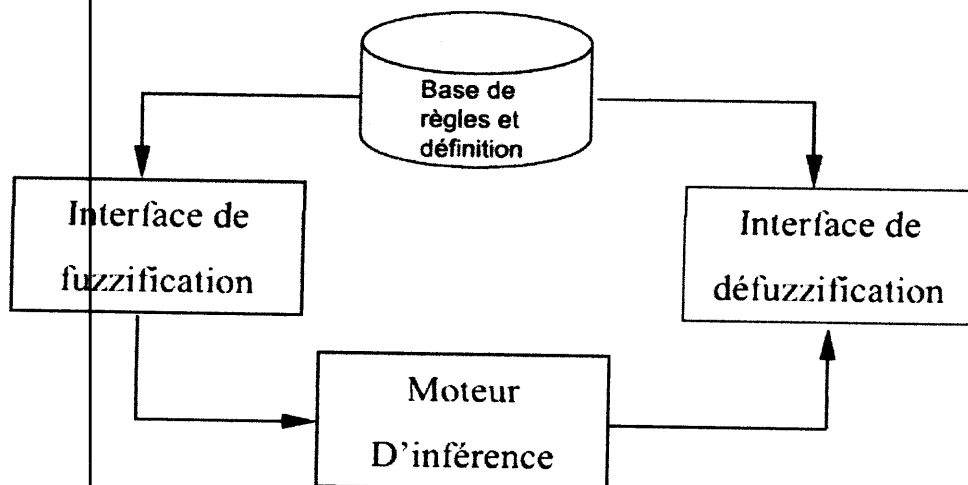


Fig I.5: Structure générale d'une commande floue.

**I.4.4.1 Interface de la fuzzification :** La fuzzification proprement dite consiste à définir des fonctions d'appartenance pour les différentes variables linguistiques. Ceci a pour but la conversion d'une grandeur physique en une grandeur linguistique.

D'une autre manière, l'entrée  $x$  varie dans l'univers de discours qui est partagé en un nombre fini d'ensembles flous de telle sorte que dans chaque zone il y ait une situation dominante. Afin de faciliter le traitement numérique et l'utilisation de ces

ensembles, on les décrit par les fonctions d'appartenance. Elles admettent comme argument la position de  $x$  dans l'univers de discours, et comme sortie le degré d'appartenance de  $x$  à la situation décrite par la fonction .

**I.4.4.2 Interface d'inférence floue :** L'interface d'inférence est formée de deux blocs :

- ❖ La base de règles, composé d'un ensemble de relations liant les variables d'entrées aux variables de sorties du système à régler. Chaque relation est composée d'une condition précédée du symbole **Si** appelée prémisses, et d'une conclusion (action, décision, opération ou commande) précédée du symbole **Alors**.
- ❖ Le moteur d'inférence réalise le traitement numérique des règles d'inférence, décrites par des opérateurs flous, pour obtenir la sortie linguistique ou floue du régulateur. Cette opération est faite par différentes méthodes, on cite principalement :

a) **La méthode d'inférence max-min :** L'opérateur **ET** est réalisé par la formation du minimum, l'opérateur **OU** est réalisé par la formation du maximum, et l'implication (**ALORS**) est réalisée par la formation du minimum.

b) **La méthode d'inférence max-produit :** L'opérateur **ET** est réalisé par la formation du produit, l'opérateur **OU** est réalisé par la formation du maximum, et l'implication (**ALORS**) est réalisée par la formation du produit.

c) **La méthode d'inférence somme-prod :** On réalise au niveau de la condition, l'opérateur **OU** par la formation de la somme (valeur moyenne), et l'opérateur **ET** par la formation du produit. Pour la conclusion, l'opérateur **ALORS** est réalisé par un produit.

**I.4.4.3 Interface de défuzzification :** La transformation d'une information floue en une information déterminée est la défuzzification. Pendant cette étape se fait la déduction de la grandeur de sortie numérique à partir de l'inférence floue. Il s'agit de calculer, à partir des degrés d'appartenance de tous les ensembles flous des

variables d'entrées et des ensembles flous de la variable de sortie, une valeur numérique de la variable de sortie en utilisant un ensemble de règles. Parmi les stratégies de défuzzification, on cite la méthode du centre de gravité, la méthode du maximum et la méthode de la moyenne des maximums.

a) **La technique du maximum** : elle est la plus simple. Elle consiste à considérer seulement, pour chaque sortie, la règle présentant le maximum de validité. Cette règle, ignore les règles secondaires qui peuvent néanmoins être importantes pour le fonctionnement et la stabilité du système. Elle est par conséquent peu employée

b) **La technique de la moyenne des maximums** : elle considère, comme valeur de sortie, la moyenne de toutes les valeurs pour lesquelles la fonction d'appartenance issues de l'inférence est maximale.

c) **La technique du centre de gravité** : plus performante, elle consiste à tracer, sur un même diagramme représentant les ensembles flous de sortie, les différentes zones correspondantes à chacune des règles et à calculer le centre de gravité de la zone consolidée.

#### **I.4.5 Différents types de contrôleurs flous [14]**

Les contrôleurs flous sont principalement composés de deux types :

- Contrôleur flou type Mamdani.
- Contrôleur flou type Sugeno.

##### **I.4.5.1 Contrôleur flou type Sugeno**

D'une manière générale, les contrôleurs de Sugeno forment une famille particulière de contrôleur, la forme générale d'un jeu de règles à deux entrées et une conclusion numérique polynomiale est décrit par la forme suivante :

$$\text{si } (x \text{ est } A_i) \text{ ET } (y \text{ est } B_i) \text{ Alors } (u_i = a_i + b_{iX} + c_{iY})$$



Le calcul de la commande engendrée par les entrées  $x_0$  et  $y_0$  est donné par l'expression :

$$u = \frac{\sum_i w_i \cdot u_i}{\sum_i w_i} \quad (\text{I.14})$$

Avec  $w_i = f_{\tau(A_i)}(x) \cdot f_{\tau(B_i)}(y)$

Sugeno a proposé une méthode d'inférence floue qui garantit la continuité de la sortie, cette méthode d'inférence s'avère très efficace dans des applications faisant intervenir à la fois des technique linéaires et adaptatives.

Les contrôleurs de Sugeno appartiennent à la classe des contrôleurs à conclusion numérique qui fournissent une partie floue de domaine numérique.

#### ***1.4.5.2 Contrôleur flou type Mamdani***

Dans la plupart des applications reportées dans la littérature, un contrôleur de ce type est conçu pour régler une variable de sortie d'un procédé, soit uniquement à partir de l'erreur  $e$  (entre la consigne et la sortie mesurée), soit à partir de l'erreur  $e$  et de sa variation  $\Delta e$ . En 1974, Mamdani avait présenté, pour la première fois, la technique de réglage par la logique floue. Celle-ci consiste à déterminer un ensemble de règles qui maîtrise le comportement dynamique du système à commander. L'obtention de ces règles est facile auprès des experts qui connaissent bien le système. Il avait utilisé des règles à prémises et conclusions symbolique, l'inférence (max, min).

Supposons que la base de connaissances est constituée de  $n$  règles d'inférence contenant chacune  $m$  prémises et une conclusion. Le processus d'inférence peut être décrit comme suit :

Règle 1 : *si*( $x_1$  est  $A_{11}$ )*ET* ... .. *ET*( $x_m$  est  $A_{1m}$ )*Alors*( $y$  est  $B_1$ )

Règle 2 : *si*( $x_1$  est  $A_{21}$ )*ET* ... .. *ET*( $x_m$  est  $A_{2m}$ )*Alors*( $y$  est  $B_2$ )

Règle  $n$  : *si*( $x_1$  est  $A_{n1}$ )*ET* ... .. *ET*( $x_m$  est  $A_{jm}$ )*Alors*( $y$  est  $B_j$ )

Dans lequel  $x_1, \dots, x_m$  sont des éléments des univers du discours  $X_1, \dots, X_m$  et  $A_{ij}$  ( $j = 1, \dots, m$ ), sont des quantités floues sur l'univers du discours  $X_i$ , et  $B_j$  ( $j = 1, \dots, m$ ), sont également des quantités floues sur l'univers du discours  $Y$ . Afin de définir une seule prémisse pour une règle  $i$ , les propositions « $x_j$  est  $A_{ij}$ », ( $j = 1, \dots, m$ ), sont combinées par l'opérateur minimum. La fonction d'appartenance de cette prémisse unique est donc donnée par :

$$\mu_{RAi}(X_1, \dots, X_m) = \mu_{Ai1}(X_1) \cap \dots \cap \mu_{Aim}(X_m) \quad (I.15)$$

#### I.4.6 Les domaines d'application [15]

Il existe de nombreuses applications la commande floue dans différents domaines, voici une liste non exhaustive de quelques-unes :

- Appareils électroménagers (lave-linge, aspirateur, autocuiseur,...etc.).
- Systèmes audio-visuels (appareil photo autofocus, caméscope à stabilisateur d'images, photocopieur,...).
- Systèmes automobiles embarqués (ABS, suspension, climatisation,...etc.).
- Systèmes de transport (train, métro, ascenseur,...).
- Systèmes de décision, diagnostic (médical, assurance).
- Systèmes de contrôles/commandes dans la plupart des domaines industriels de production, transformation, traitement de produit et déchet.
- Systèmes autonomes mobiles.

#### I.4.7 Avantages et inconvénients de la commande floue

##### I.4.7.1 Avantages

- La théorie est simple et s'applique à des systèmes complexes.
- Pas de modèles mathématiques requis du procédé à asservir.

- Robustesse de la commande floue vis à vis des incertitudes.
- Possibilités de commande auto-adaptative aux variations du procédé.

#### **I.4.7.2 Inconvénients**

- Technique de réglage essentiellement empirique.
- Performances dépendent de l'expertise.
- Il n'existe pas de théorie générale qui caractérise rigoureusement la stabilité, la robustesse des contrôleurs flous.

### **I.5 Conclusion**

Les contrôleurs flous (CF) dont les bases relèvent de la théorie des ensembles flous proposés par Zadeh, sont devenus très populaires. La commande floue aide à gérer des systèmes complexes de façon simple et facilement explicitable ' *l'expertise humaine*'. La difficulté réside dans sa mise au point d'une base de connaissances floues (structure & paramètres du CF). Ce problème peut être résolu par l'utilisation des méthodes d'extraction automatique des connaissances : Les méthodes analytiques, les techniques d'intelligence artificielle, les métaheuristiques...etc.

## Chapitre II

# L'Optimisation par Essaim Particulaire

### II.1 Introduction

Depuis que l'ordinateur existe, l'homme n'a cessé de le perfectionner pour qu'il soit le plus rapide et le plus performant, mais rien de cela ne serait possible sans la partie logiciel qui vient s'implanté sur ce support. A l'air où les logiciels les plus rapide du marché ne peuvent plus résoudre certains problèmes (de part leur complexité), l'homme s'inspire de plus en plus de la nature qui l'entoure pour mettre en place des algorithmes simulant le comportement des animaux, leurs actions ou réactions vis-à-vis d'un problème et même les méthodes qu'ils utilisent pour y faire face.

Même si ces méthodes sont sujette à de nombreuses controverses de part leur non-exactitude, ils permettent de trouver facilement et rapidement la solution la plus approchée de l'optimal si ce dernier existe, et reste un moyen très efficace pour traitement de problèmes complexes pouvant prendre des années de calculs sans résultats.

Les méta-heuristiques, comprenant notamment la méthode du recuit simulé, les algorithmes évolutionnaires, la méthode de recherche tabou, les algorithmes de colonies de fourmis, l'optimisation par essaim particulaire... sont apparues à partir des années 80. Ce sont des algorithmes d'optimisation de type stochastiques et progressant vers un optimum par échantillonnage d'une fonction objective dont le

but est la résolution de problèmes d'optimisation difficile en se basant sur plusieurs notions telles que le voisinage, la diversification et l'intensification

Ce chapitre est structuré comme suit : dans la section II.2, nous présentons les métaheuristiques d'optimisation ; dans la section II.3, nous détaillons l'algorithme d'Optimisation par Essaim Particulaire (OEP, *PSO en anglais*), qui constitue le sujet principal de notre mémoire.

## II.2 Méta-heuristique [16]

### II.2.1 Définition

Le mot méta-heuristique est dérivé de la composition de deux mots grecs:

- heuristique qui vient du verbe *heuriskein* et qui signifie 'trouver'
- Meta qui est un suffixe signifiant 'au-delà', 'dans un niveau supérieur'

Une **méta-heuristique** est un algorithme d'optimisation visant à résoudre des problèmes d'optimisation difficile (souvent issus des domaines de la recherche opérationnelle, de l'ingénierie ou de l'intelligence artificielle) pour lesquels on ne connaît pas de méthode classique plus efficace.

Les méta-heuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction, par échantillonnage d'une fonction objective. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation).

Il existe un grand nombre de méta-heuristiques différentes, partant d'une recherche locale simple à des algorithmes de recherche globale complexes, ces méthodes utilisent cependant un haut niveau d'abstraction, leur permettent d'être adaptées à une large gamme de problèmes différents.

### II.2.2 Principe de méta-heuristique

Le but d'utilisation d'une méta-heuristique est de résoudre un problème d'optimisation donné : elle cherche un objet mathématique (une permutation, un vecteur, etc.) minimisant (ou maximisant) une fonction *objectif*, qui décrit la qualité d'une solution du problème.

Les méta-heuristiques manipulent une ou plusieurs solutions, à la recherche de l'optimum, la meilleure solution du problème. Les itérations successives doivent permettre de passer d'une solution de mauvaise qualité à la solution optimale. L'algorithme s'arrête après avoir atteint un critère d'arrêt, consistant généralement en l'atteinte du temps d'exécution imparti ou en une précision demandée.

L'ensemble des solutions possibles forme l'espace de recherche. L'espace de recherche est au minimum borné, mais peut être également limité par un ensemble de contraintes.

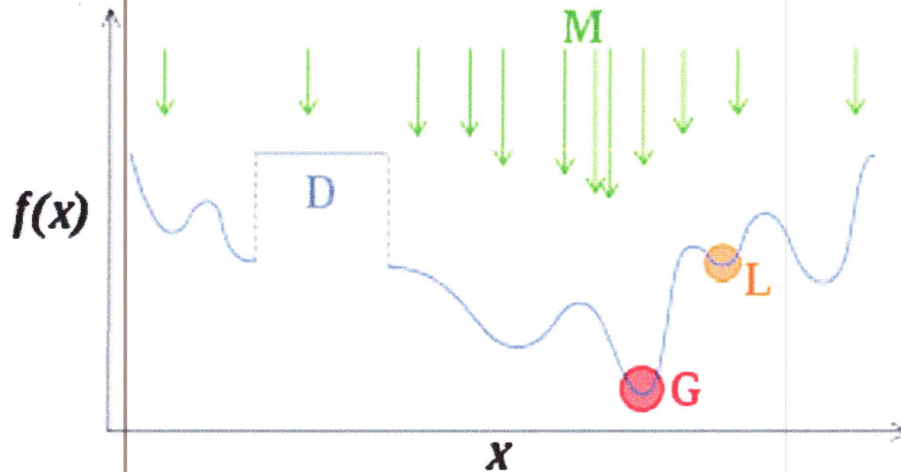


Fig II.1 : Principe générale des méta-heuristiques.

### II.2.3 Différents types de méta-heuristiques

Il existe un grand nombre de méta-heuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes peuvent être adaptées à une large gamme de problèmes différents. Elles peuvent être regroupées en deux grandes catégories; les méta-heuristiques à population, et les

méta-heuristiques à parcours. Les algorithmes présents dans chacune de ces catégories sont les plus classiques, entre ces deux classes on peut trouver des méta-heuristiques moins connus ceux qui sont conçus pour des problèmes particuliers.

### II.2.3.1 Les méta-heuristiques à population

Le principe des méta-heuristiques à population ou encore dite évolutives est de faire évoluer un ensemble de solutions vers l'optimum cherché. Cette évolution se fait à partir de transformations et de coopérations entre les individus qui représentent individuellement une solution de l'espace total du problème. Parmi ces méthodes, nous distinguons essentiellement les algorithmes génétiques, et les colonies de fourmis.

### II.2.3.2 Les méta-heuristiques à parcours

Les méta-heuristiques à parcours ne travaillent qu'avec une seule solution, cette dernière est modifiée à chaque itération de l'algorithme. L'espace des solutions est visité par la solution, et celle-ci est modifiée pour être améliorée au cours de ce parcours.

Les méta-heuristiques à parcours les plus classiques sont : le recuit simulé, la recherche Tabou. Les métaheuristiques peuvent être classées de nombreuses façons. Ce diagramme tente de présenter où se placent quelques méthodes les plus connues.

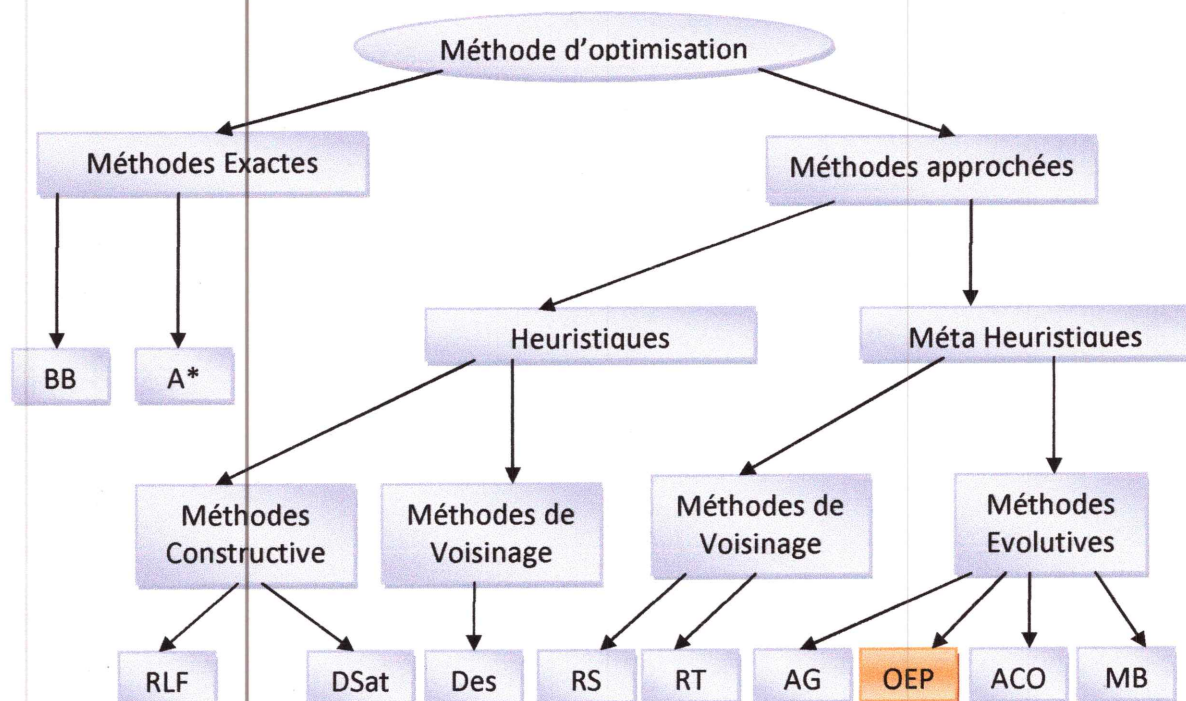


Fig. II.2 : Classification des méthodes d'optimisation.

BB : Branch & Bound.

RT : Recherche Tabou.

RLF : Récursive Largest First.

AG : Algorithmes Génétiques.

DSatur : Degré Saturation maximal.

OEP : Optimisation par essaim de particules

Desc : Méthode de Descente.

ACO : Optimisation par Colonie de Fourmis.

RS : Recuit Simulé.

MBO : Optimisation par Colonie d'abeilles.

### ➤ Domaines d'application

Les métaheuristiques ont été appliquées avec succès pour résoudre de larges gammes de problèmes, tels que :

- Problèmes de tournée de véhicules.
- Optimisation de réseaux mobiles UMTS.
- Gestion du trafic aérien.
- Optimisation des plans de chargement des cœurs de réacteur nucléaires,....etc.

## II.3 L'optimisation par essaim de particules [17] [18] [19] [20] [21] [22] [23]

### II.3.1 Introduction

En 1995, Russel Eberhart, ingénieur en électricité et James Kennedy, socio-psychologue, s'inspirent du monde du vivant pour mettre en place une méta heuristique : l'optimisation par essaim particulaire.

L'origine de cette méthode vient des observations faites lors des simulations informatiques de vols groupés d'oiseaux et de bancs de poissons .Ces simulations ont mis en valeur la capacité des individus d'un groupe en mouvement à conserver une distance optimale entre eux et à suivre un mouvement global par rapport aux mouvements locaux de leur voisinage.



### II.3.2 Définition

L'optimisation par essaim de particules (Particle Swarm Optimization (*PSO*)) est une méthode d'optimisation stochastique, pour les fonctions non-linéaires, basée sur la reproduction d'un comportement social. Elle repose sur un ensemble d'individus originellement disposés de façon aléatoire et homogène, que nous appellerons « des particules », qui se déplacent dans l'hyper-espace de recherche et constituent, chacune, une solution potentielle.

Chaque particule dispose d'une mémoire conservant sa meilleure solution visitée ainsi que la capacité de communiquer avec les particules constituant son entourage. À partir de ces informations, la particule va suivre une tendance faite, d'une part, de sa volonté à retourner vers sa solution optimale, et d'autre part, de son mimétisme par rapport aux solutions trouvées dans son voisinage.

À partir d'optimums locaux trouvés, l'ensemble des particules va converger vers la solution optimale globale du problème traité.

### II.3.3 Principe de déplacement d'une particule

Un essaim de particules, qui sont des solutions potentielles au problème d'optimisation, « survole » dans l'espace de recherche, à la recherche de l'optimum global. Le déplacement d'une particule est influencé par trois composantes :

- une composante d'inertie : la particule tend à suivre sa direction courante de déplacement.
- une composante cognitive : la particule tend à se fier à sa propre expérience et, ainsi à se diriger vers le meilleur site par lequel elle est déjà passée.
- une composante sociale : la particule tend à se fier à l'expérience de ses congénères et, ainsi à se diriger vers le meilleur site déjà atteint collectivement par l'essaim. La figure II.3 illustre la stratégie de déplacement d'une particule.

Dans un espace de recherche de dimension  $d$ , la particule  $i$  de l'essaim est modélisée par son vecteur de position  $x_i = (x_{i1} \dots x_{i2} \dots x_{id})^T$  et par son vecteur de

vitesse  $v_i = (v_{i1} \dots v_{i2} \dots v_{id})^T$ . La qualité de la position est déterminée par la valeur de la fonction objective en ce point. Cette particule garde en mémoire la meilleure position par laquelle elle est déjà passée, que l'on note  $p_i = (p_{i1} \dots p_{i2} \dots p_{id})^T$ . La meilleure position atteinte par ses particules voisines est notée  $g_i = (g_{i1} \dots g_{i2} \dots g_{id})^T$ . Au temps  $t$ , les vecteurs de vitesse et de position de chaque particule sont calculés successivement à partir des équations (III.1) (III.2).

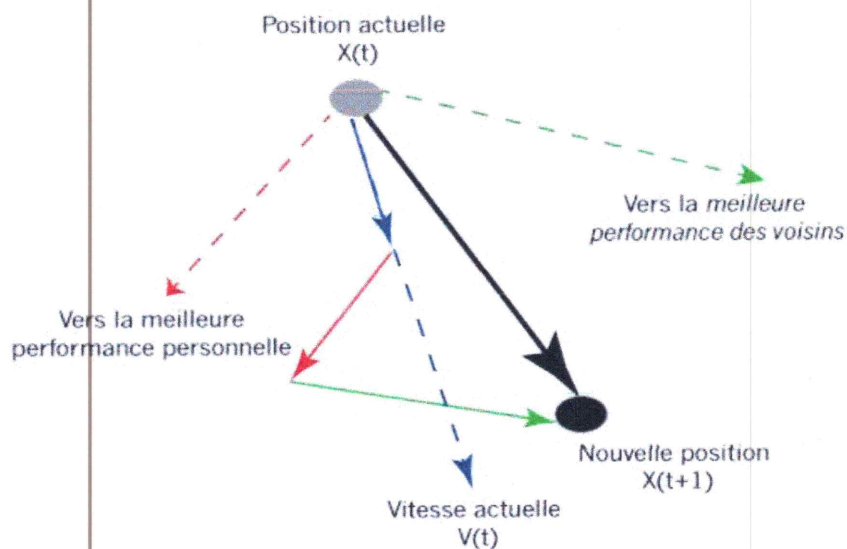


Fig. II.3 : Déplacement d'une particule.

$$v_{i,j}(t+1) = w v_{i,j}(t) + c_1 r_{1,i,j} [p_{i,j}(t) - x_{i,j}(t)] + c_2 r_{2,i,j} [g_i(t) - x_{i,j}(t)] \quad (\text{III.1})$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad (\text{III.2})$$

- $w$  : est une constante, appelée coefficient d'inertie.
- $c_1$  et  $c_2$  : sont deux constantes, appelées coefficients d'accélération.
- $r_{1,i,j}$  et  $r_{2,i,j}$  : sont deux nombres aléatoires tirés uniformément dans  $[0, 1]$ , à chaque itération  $t$  et pour chaque dimension  $j$ .

### II.3.4 Les caractéristiques de l'OEP

Un essaim de particules est caractérisé par : le nombre de particules, la vitesse, la position, l'inertie d'une particule, les coefficients de confiance.....etc.

#### II.3.4.1 nombre de particules

La quantité de particules allouées à la résolution du problème dépend essentiellement de deux paramètres : la taille de l'espace de recherche, le rapport entre les capacités de calcul de la machine et le temps maximum de recherche. Il n'y a pas de règle pour déterminer ce paramètre, donc il faut faire de nombreux essais pour permettre de se doter de l'expérience nécessaire à l'appréhension de ce paramètre.

#### II.3.4.2 La position, la vitesse

Dans un espace de recherche, la particule  $i$  de l'essaim est modélisée par son :

➤ Vecteur de position:  $\vec{X}_i = (x_{i1}, x_{i2}, \dots, x_{id})$

➤ Vecteur de vitesse:  $\vec{V}_i = (v_{i1}, v_{i2}, \dots, v_{id})$

La qualité de sa position est déterminée par la valeur de la fonction objective en ce point. Cette particule garde en mémoire la meilleure position par laquelle elle est déjà passée, que l'on note :

$$\vec{P}_{best} = (p_{i1}, p_{i2}, \dots, p_{id}) \quad (\text{III.3})$$

La meilleure position atteinte par les particules de l'essaim est notée :

$$\vec{G}_{best} = (g_1, g_2, \dots, g_d) \quad (\text{III.4})$$

#### II.3.4.3 Facteur d'inertie

Le facteur d'inertie  $w$  qui permet de définir la capacité d'exploration de chaque particule en vue d'améliorer la convergence.

- Une grande valeur de  $w (> 1)$  est synonyme d'une grande amplitude et donc, d'exploration globale.
- Une faible valeur de  $w (< 1)$  est synonyme de faible amplitude de mouvement et donc, d'exploration locale.

### II.3.5 Voisinage

#### II.3.5.1 Notion

Le voisinage d'une particule est le sous-ensemble de particules de l'essaim avec lequel il a une communication directe. Ce réseau de rapports entre toutes les particules est connu comme la sociométrie, ou la topologie de l'essaim.

Il existe deux principaux types de voisinage :

- **Les voisinages géographiques** : les voisins sont considérés comme les particules les plus proches. Cependant, à chaque itération, les nouveaux voisins doivent être recalculés à partir d'une distance prédéfinie dans l'espace de recherche. C'est donc un voisinage dynamique.

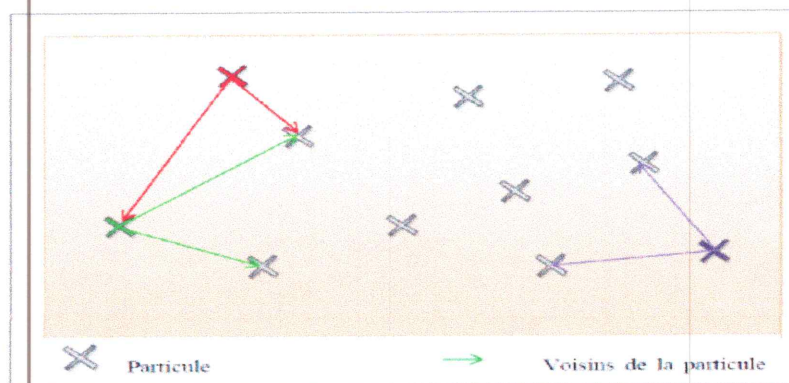


Fig II.4 : Le voisinage géographique.

- **Les voisinages sociaux** : les voisins sont définis à l'initialisation et ne sont pas modifiés ensuite. C'est le voisinage le plus utilisé, pour plusieurs raisons :
  - ✚ Il est plus simple à programmer.
  - ✚ Il est moins coûteux en temps de calcul.
  - ✚ En cas de convergence, un voisinage social tend à devenir un voisinage géographique.

Pour ce faire, on dispose les particules en cercle puis, pour la particule étudiée, on inclut progressivement dans ses informatrices, d'abord elle-même, puis les plus proches à sa droite et à sa gauche, jusqu'à atteindre la taille voulue. On peut aussi choisir les informatrices au hasard.

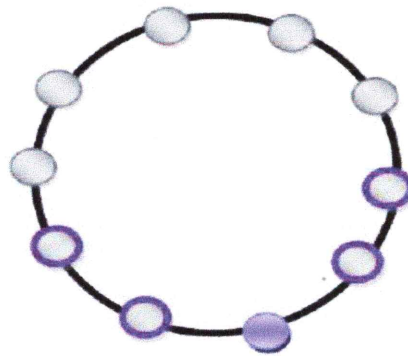


Fig II.5 : Le voisinage en cercle.

### II.3.5.2 Les différents types de topologie

Le voisinage constitue la structure du réseau social. Les particules à l'intérieur d'un voisinage communiquent entre-elles. En général, pour une nuée d'oiseaux, le voisinage suit trois types de topologies :

- **Topologie en étoile:** le réseau social est complet, donc une communication complète et une attirance vers la meilleure particule (fig II.6).

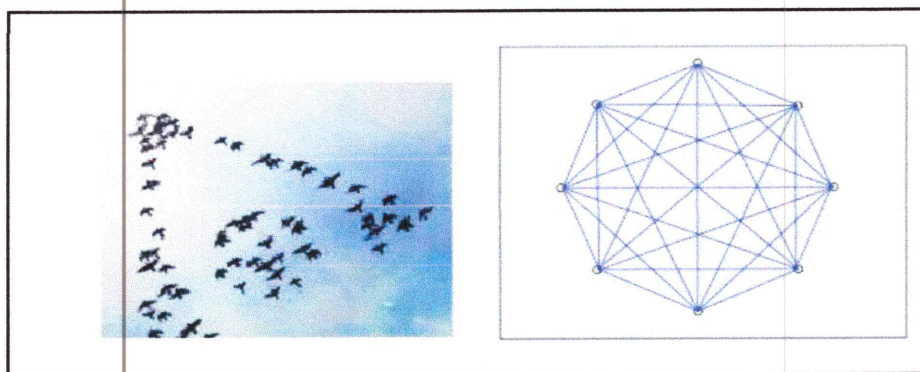


Fig II.6 : Topologie en étoile.

- **Topologie en anneau:** chaque particule communique avec  $n$  voisines immédiates. Chaque particule tend à se déplacer vers la meilleure dans son voisinage local (Fig II.7).

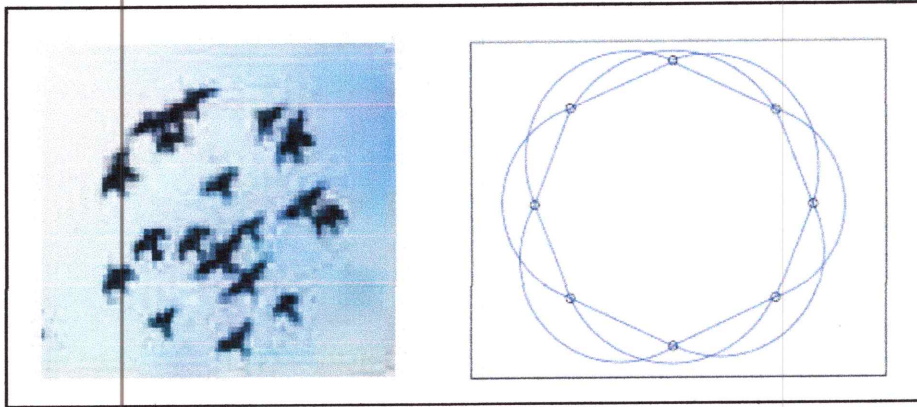


Fig II.7 : Topologie en anneau.

- **Topologie en rayon:** une particule "centrale" est connectée à toutes les autres. Seule cette particule centrale ajuste sa position vers la meilleure, si cela provoque une amélioration l'information est propagée aux autres (Fig II.8).

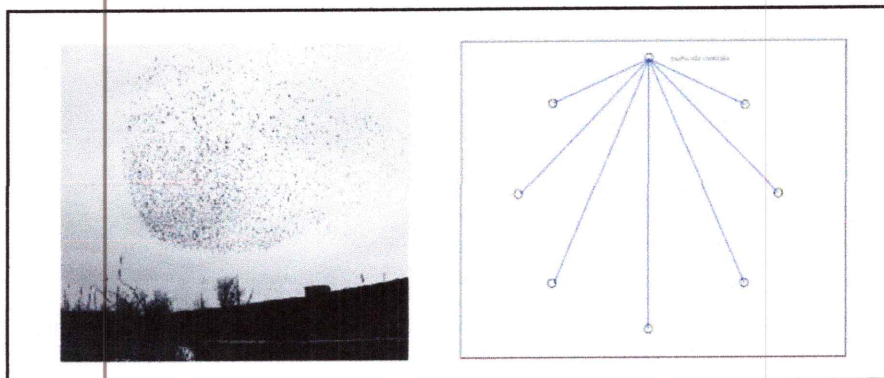


Fig II.8 : Topologie en rayon.

Le choix de la topologie de voisinage a un grand effet sur la propagation de la meilleure solution trouvée par l'essaim. En utilisant le modèle étoile la propagation est très rapide (c-à-d toutes les particules dans l'essaim seront affectées par la

meilleure solution trouvée dans l'itération  $t$ , immédiatement dans l'itération  $(t+1)$ . Cependant, en utilisant les topologies anneaux et rayon la progression sera ralentit.

### II.3.6 Algorithme standard de L'OEP

Comme indiqué précédemment, l'OEP simule les comportements de flocage des oiseaux. Supposons que le scénario suivant: un groupe d'oiseaux sont à la recherche alimentaire au hasard dans une zone. Il y a seulement un morceau de nourriture dans la zone recherchée. Tous les oiseaux ne savent pas où est la nourriture mais ils savent à quel point la nourriture est à chaque itération. Alors, quelle est la meilleure stratégie pour trouver la nourriture? L'une est efficace pour suivre l'oiseau qui est le plus proche de l'aliment.

L'OEP est initialisé avec un groupe de particules aléatoires (solutions), puis la recherche des optima en mettant à jour les positions et les vitesses est effectuée. Dans chaque itération, chaque particule est mise à jour en suivant deux valeurs «meilleures»: Sa meilleure solution déjà visitée et la meilleure solution trouvée par l'essaim.

A chaque itération :

- Chaque particule est capable d'évaluer la qualité de sa position qu'elle à atteinte jusqu'ici et sa qualité (la valeur en cette position de la fonction à optimiser).
- Chaque particule est capable d'interroger un certain nombre de ses voisins et d'obtenir de chacune d'entre elles sa propre expérience.
- A chaque pas de temps, chaque particule choisit la meilleure performance dont elle à connue, modifie sa vitesse en fonction de cette information par conséquence vers une meilleure performance.

L'algorithme standard de l'OEP est le suivant :

- **Initialisation** aléatoire des positions et des vitesses de chaque particule
- **Pour** chaque  $i$ ,
- **Tant que** le critère d'arrêt n'est pas atteint **faire**

**Pour**  $i = 1$  à  $N$  **faire**

**Déplacement** de la particule  $i$  à l'aide de (III.1) et (III.2)

**Évaluer** sa nouvelle position

**Si**  $f(x_i) < f(Pbest_i)$  **Alors**

$Pbest_i = x_i$

**Fin Si**

**Si**  $f(x_i) < f(Gbest_i)$  **Alors**

$Gbest_i = Pbest_i$

**Fin Si**

**Fin Pour**

**Fin Tant que**

### II.3.7 critères d'arrêt

- Si l'optimum global est connu a priori, on peut définir une erreur acceptable.
- un nombre maximum d'évaluations de la fonction objectif.
- nombre maximum d'itérations.

### II.3.8 Avantages et limitations

Contrairement à d'autres méthodes, l'optimisation par essaim de particules permet en général de converger rapidement vers une solution globale de bonne qualité. On peut ajouter qu'elle est relativement simple et facile à implémenter. A l'initialisation de l'algorithme, il est seulement nécessaire de bien décrire le problème à optimiser, les contraintes du problème, la fonction coût que l'on veut minimiser.

Par contre, il faut noter que parmi les inconvénients de la méthode il y a le



réglage des paramètres, citons  $c_1$ ,  $c_2$ ,  $w$  le nombre d'itérations et le nombre de particules. Le choix de ces paramètres reste toujours un problème posé selon le problème à résoudre.

#### II.4 Conclusion

Les méta-heuristiques constituent une stratégie de résolution de plus en plus privilégiée. Une des particularités importantes des méta-heuristiques, réside dans l'absence d'hypothèses particulière sur la régularité de la fonction objective. Aucune hypothèse sur la continuité de cette fonction n'est requise, ses dérivées successives ne sont pas nécessaires, ce qui rend très vaste le domaine d'application.

L'optimisation par essaims de particules présente un fort potentiel d'application pratique, mais le choix de paramètres reste l'un des problèmes de l'optimisation par OEP, c'est très difficile de trouver des bons paramètres adaptés à la structure du problème.

## Chapitre III

### Simulation & Interprétations

#### III.1 Introduction

Devant les problèmes délicats de la commande de systèmes complexes, les outils utilisés deviennent de plus en plus pointus. L'un des axes de recherche concerne la commande de systèmes non linéaires mono variable. Dans le but de tester et évaluer les performances de l'algorithme d'optimisation par essaim particulaire, nous avons effectué une application en simulation sur un système non linéaire monovarié qui est le système de bain d'eau. Le problème de commande du bain d'eau est vraisemblablement un des systèmes les plus utilisés pour tester différentes méthodes de commandes et d'optimisations. L'objectif de commande est de permettre aux états du système de suivre une trajectoire de référence prédéterminée.

Afin de mettre en évidence l'ensemble des caractéristiques des contrôleurs flous, nous avons procédé, dans la première partie de ce chapitre, à l'élaboration d'un régulateur flou de type Takagi-Sugeno d'ordre zéro pour commander la température du bain d'eau.

Nous proposons, dans la deuxième partie, une stratégie d'optimisation basée sur l'algorithme d'optimisation par essaim particulaire, qui utilise la notion de voisinage et de déplacement de particule selon une composante cognitive, sociale et d'inertie pour diriger une exploration hautement intelligente dans l'espace de recherche. Cette optimisation concerne les paramètres des prémisses et des conclusions de règles floues.

### III.2 Le modèle mathématique du bain d'eau

La structure du procédé de contrôle de la température de bain d'eau est donnée par la figure suivante :

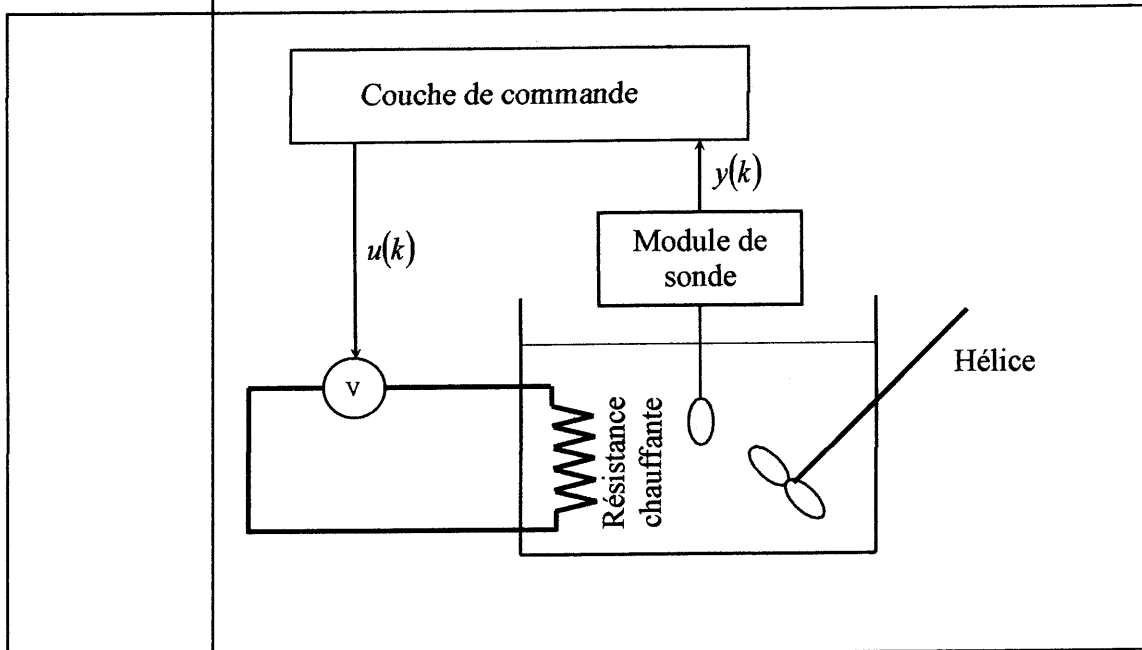


Fig III.1 : Structure du procédé.

La dynamique du système est caractérisée par l'expression suivante [24]:

$$y(k+1) = Q_1(T_s) \cdot y(k) + \frac{Q_2(T_s)}{1 + \exp(0.5y(k) - \gamma)} u(k) + (1 - Q_1(T_s)) \cdot Y_0 \quad (\text{III.1})$$

$$Q_1(T_s) = \exp(-\alpha T_s) \quad (\text{III.2})$$

$$Q_2(T_s) = \frac{\beta}{\alpha} \exp(1 - \exp(-\alpha T_s)) \quad (\text{III.3})$$

Avec :

$y(k)$  : La température du système.

$u(k)$  : Signal de commande à appliquer.

$Y_0$  : La température ambiante.

$T_s$  : La période d'échantillonnage.

$\alpha$ ,  $\beta$  et  $\gamma$  sont des constantes.

Les paramètres du système sont donnés par la table suivante :

Coefficient	Valeur
$\alpha$	$1.0015.10^{-4}$
$\beta$	$8.6793.10^{-3}$
$\gamma$	40
$T_s$ [s]	30
$Y_0$ [°C]	25

Tab III.1 : Paramètres du système.

Dans la première partie de simulation, l'objectif de contrôle est de générer une séquence d'actions  $0 \leq u(k) \leq 5$  permettant au système de suivre le trajet suivant :

$$y_{ref}(k) = \begin{cases} 35^\circ\text{C} & \text{si } k \leq 40 \\ 55^\circ\text{C} & \text{si } 40 \leq k \leq 80 \\ 75^\circ\text{C} & \text{si } 80 \leq k \leq 120 \end{cases} \quad (\text{III.4})$$

Dans la deuxième partie de simulation, le test de robustesse est effectué en faisant changer le modèle de référence.

Pour réaliser cette tâche on développe une loi de commande floue : 333 ; càd : 3 ensembles flous pour l'entrée  $e$ , 3 ensembles flous pour l'entrée  $\Delta e$  et 3 singletons flous pour la sortie.

### III.3 La commande floue du bain d'eau

#### III.3.1 Le type de Contrôleur flou utilisé

Le but de ce travail est de commander la température d'un bain d'eau. Le contrôleur utilisé est de type PI flou où la sortie du contrôleur flou est la variation de la commande, nous obtenons alors le régulateur flou, qui réalise la loi de commande suivante:

$$\Delta u(k) = G. (e(k) + \frac{1}{T_i} \cdot \int_{-\infty}^t e(k). dk) \quad (\text{III.5})$$

$$u(k) = u(k-1) + G_{\Delta u} \cdot \Delta u(k) \quad (\text{III.6})$$

### III.3.2 Structure de base du contrôleur flou

On considère un contrôleur flou de type TS d'ordre zéro à deux entrées, l'erreur  $e(t)$  et sa variation  $\Delta e(t)$  et une sortie  $u(t)$ , la commande appliquée au système (fig III.2):

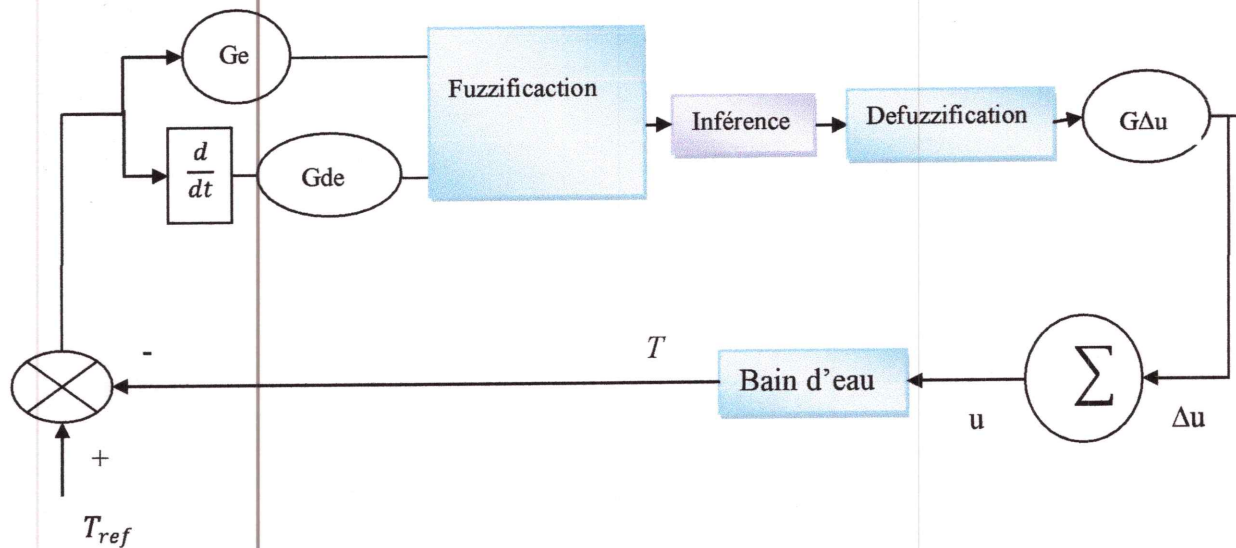


Fig III.2 : Structure de commande de la température du bain d'eau .

Où  $e(t) = T_{ref} - T =$  est la différence entre la sortie désirée  $T_{ref}$  et la sortie mesurée du système commandé  $T$ . Pour avoir une flexibilité dans l'implantation du régulateur, les univers de discours des entrées et de sortie sont limités à un intervalle  $[-1, 1]$  déterminé par la normalisation des entrées et de sortie. Pour ce faire, des gains d'adaptations (facteurs d'échelles) sont utilisés pour avoir la dynamique désirée.

Pour l'optimisation paramétrique d'un contrôleur flou, on doit alors définir :

- Le type et le nombre des ensembles flous pour chaque variable d'entrée et de Sortie.
- l'ensemble des paramètres des fonctions d'appartenance qui constituent la partition floue de la variable linguistique.
- La structure des règles floues.

- Le type de mécanisme d'inférence, les opérateurs de connections et la méthode de Défuzzification.

Nous avons utilisé pour le contrôleur flou :

- Pour la Fuzzification : des fonctions d'appartenances trapézoïdales (TPE) pour l'erreur  $e$  et sa dérivée  $\Delta e$ .
- La base de règles utilisée est celle de Macvicar et Whelan.
- Pour le mécanisme d'inférence : la méthode (min-max).
- Pour la défuzzification : la méthode du centre de gravité.
- $G_e, G_{\Delta e}, G_{\Delta u}$  sont les facteurs d'échelles de l'erreur, dérivée de l'erreur, et la variation de commande respectivement.

La figure III.3 et le tableau III.2 nous donnent respectivement, la forme des fonctions d'appartenance, on utilise trois fonctions d'appartenance de type triangulaire-trapézoïdale pour chaque variables d'entrée, ainsi la base de règles utilisée est composée de neuf règles pour un contrôleur flou de type Takagi-Sugeno à conclusions constantes.

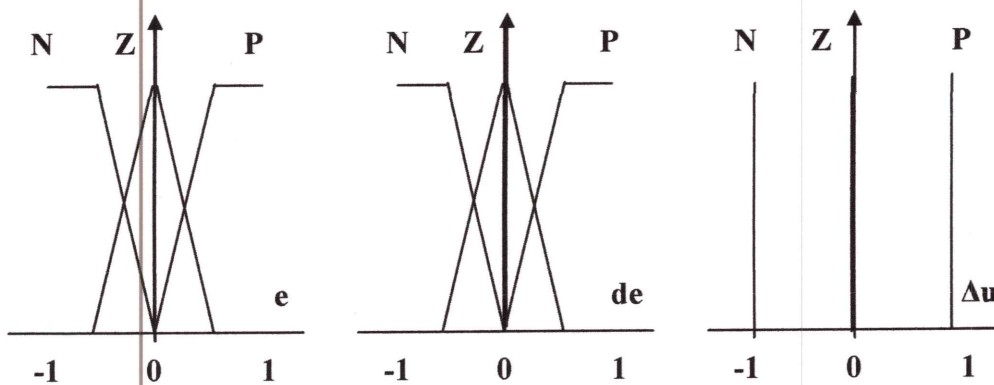


Fig III.3. Forme des fonctions d'appartenances.

U		E		
		N	Z	P
Δe	N	N	N	Z
	Z	N	Z	P
	P	Z	P	P

Tab III.2 : Table des règles du contrôleur floue (333).

III.3.3 Résultats de la simulation

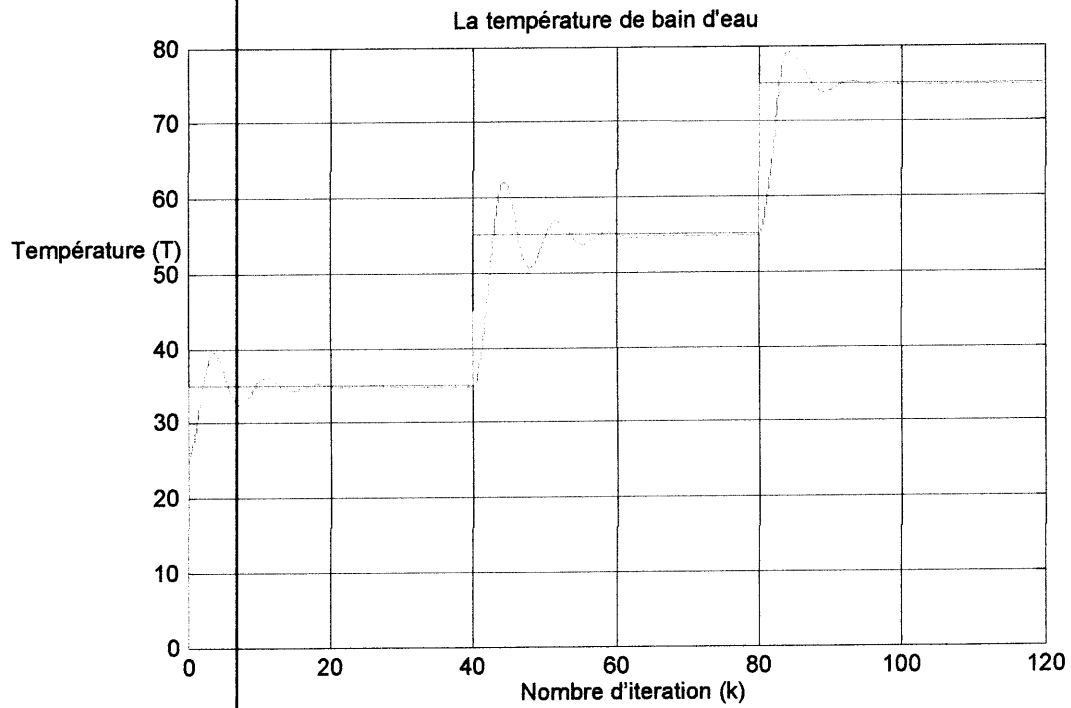


Fig III.4 : Evolution de la température et la référence en fonction du temps.

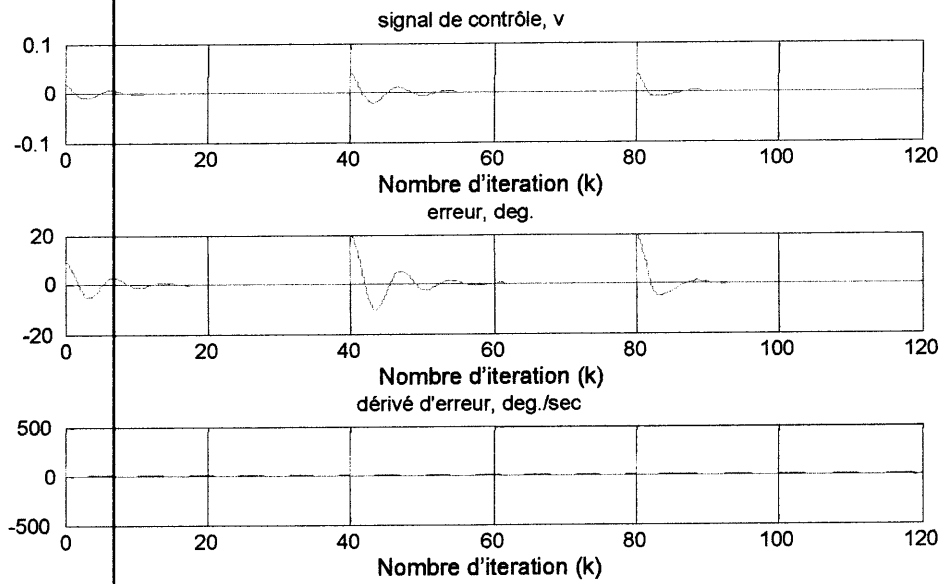


Fig III.5: Evolution signal de contrôle, erreur et dérivé d'erreur.

### ✓ Test de robustesse

La robustesse du contrôleur est justifiée par sa capacité de bien réagir lorsqu'on s'écarte des conditions de fonctionnement nominales. Dans ce cas, la robustesse est testée par le changement de la trajectoire de référence. La nouvelle trajectoire est définie comme suit:

$$y_{ref}(k) = \begin{cases} 34^{\circ}C & \text{si } k \leq 30 \\ (34 + 0.5(k - 30))^{\circ}C & \text{si } 30 < k \leq 50 \\ (44 + 0.8(k - 50))^{\circ}C & \text{si } 50 < k \leq 70 \\ (60 + 0.5(k - 70))^{\circ}C & \text{si } 70 < k \leq 90 \\ 70^{\circ}C & \text{si } 90 < k \leq 120 \end{cases} \quad (\text{III.7})$$

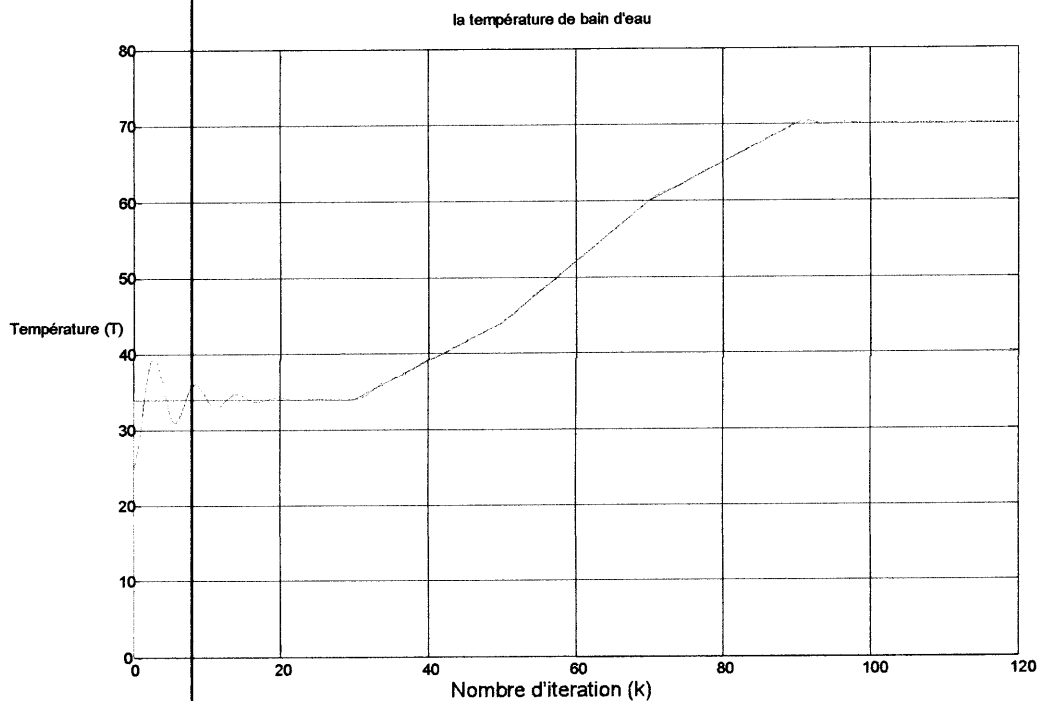


Fig III.6 : Test de robustesse en changeant la trajectoire de référence.

### III.3.4 Interprétation des résultats de simulation

Les résultats de simulation illustrent les réponses du bain d'eau soumis au régulateur de TS d'ordre zéro sont présentés respectivement sur la figure (III.4) et (III.5). Les figures montrent, respectivement, l'évolution de la température du bain



d'eau et la commande appliquée pour le système contrôlé. Le test de robustesse en changeant la trajectoire de référence est représenté dans la figure (III.6).

Les figures précédentes nous montrent une bonne convergence de la sortie même en changeant la trajectoire de la température désirée. Notons que le choix des paramètres du contrôleur flou aura des conséquences sur les performances du système contrôlé. Il est possible de régler ces paramètres manuellement, mais pour des systèmes plus complexes, une optimisation s'impose car un nombre plus important de paramètres rendra très problématique l'obtention d'un régulateur optimal par tâtonnement.

Nous allons donc développer dans la partie suivante une méthode d'extraction automatique des paramètres des prémisses et de conclusions de la base de règles floues d'un contrôleur flou de type TS d'ordre zéro pour la commande de température du bain d'eau, en gardant la même structure du contrôleur flou (333).

### **III.4 Optimisation du contrôleur flou par l'algorithme d'optimisation par essaim particulaire (OEP)**

Nous allons commencer cette partie par la présentation de l'algorithme d'optimisation par essaim particulaire (OEP) dont le but est l'optimisation d'un régulateur flou de type Takagi Sugeno d'ordre Zéro en vue de la commande de la température du bain d'eau.

#### **III.4.1 Stratégie d'optimisation du contrôleur flou**

Pour décrire le processus d'optimisation par essaim particulaire (OEP), considérant le schéma fonctionnel donné dans la figure (III.7) où  $e$  et  $\Delta e$  désignent respectivement l'erreur et la variation de l'erreur.

L'objectif global du système de commande est de minimiser l'erreur à chaque

instant  $k$  entre la réponse réelle du système et le point de consigne, l'indice de performance,  $f$  est choisi comme suit :

$$f = \frac{1}{n} \sum_{k=1} e^2(k)$$

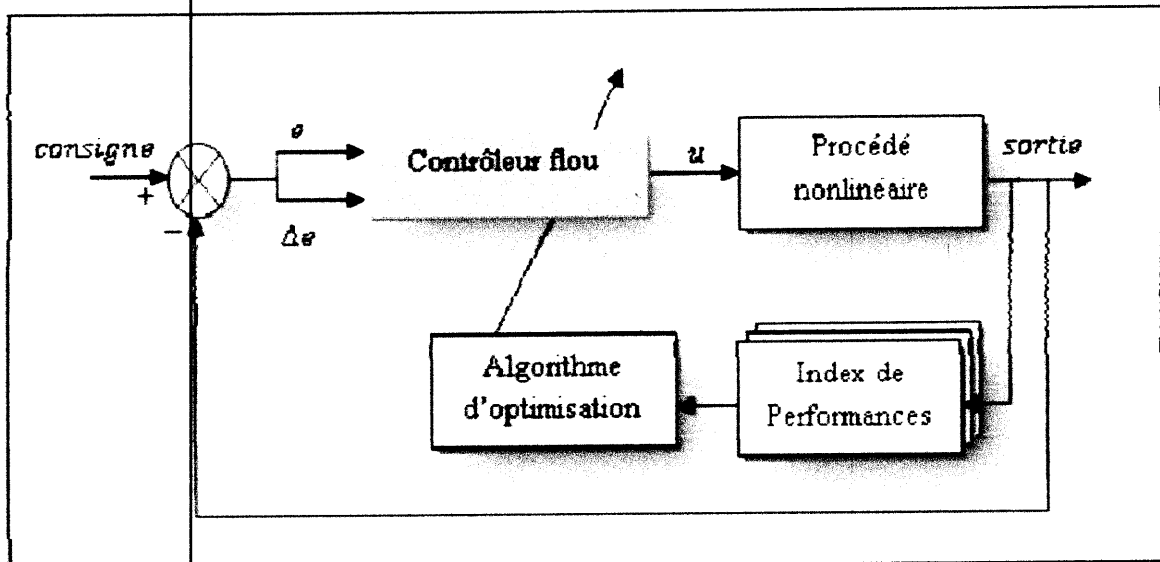


Fig III.7: Structure d'optimisation du contrôleur flou de la température.

Les paramètres déterminés par le processus d'exploitation, et qui garantissent un contrôle optimal, sont :

- D'une part, les paramètres des prémisses (valeurs modales).
- D'autre part, les paramètres des conclusions des règles (dans notre cas des singletons)

Sachant que la base de règles est constituée de neuf règles sur le tableau (III.2)

#### III.4.2 Organigramme général de l'algorithme d'optimisation par essaim particulière (OEP)

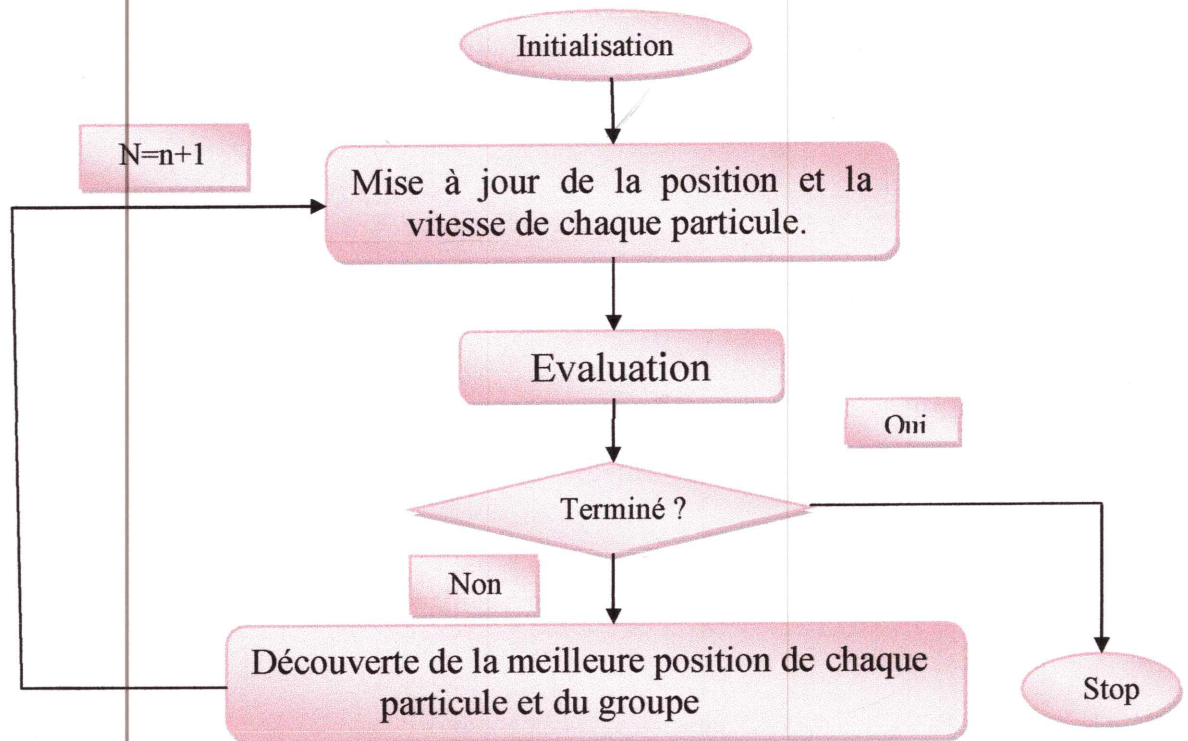


Fig III.8: Organigramme de l'algorithme d'optimisation par essais particulaires.

### III.4.3 Optimisation de la commande de la température du bain d'eau

Le contrôleur flou initial est un contrôleur flou (CF) type Takagi Sugeno d'ordre zéro, à neuf règles. Les paramètres introduits dans l'algorithme d'optimisation par essaim particulaire son comme suit :

- Les valeurs modales des fonctions d'appartenance TPE respectivement  $a_1, a_2, a_3$ , pour l'erreur et  $b_1, b_2, b_3$ , pour la variation de l'erreur sont choisis entre  $[-1, 1]$ , tout en respectant la contrainte ( $a_1 < a_2 < a_3$  et  $b_1 < b_2 < b_3$ ).
- Les valeurs de conclusions des règles respectivement  $N, Z, P$ , sont choisies entre  $[-1, 1]$ , tout en respectant la contrainte ( $N < Z < P$ ).
- Le nombre de particules : 30.
- Le nombre d'itérations (nombre de pas de l'essaim) : 10.
- $c1=c2=1.5$ .
- $w=0.9$ .
-

❖ Résultats de simulation

L'évolution de la fonction coût est donnée par la figure (III.9).

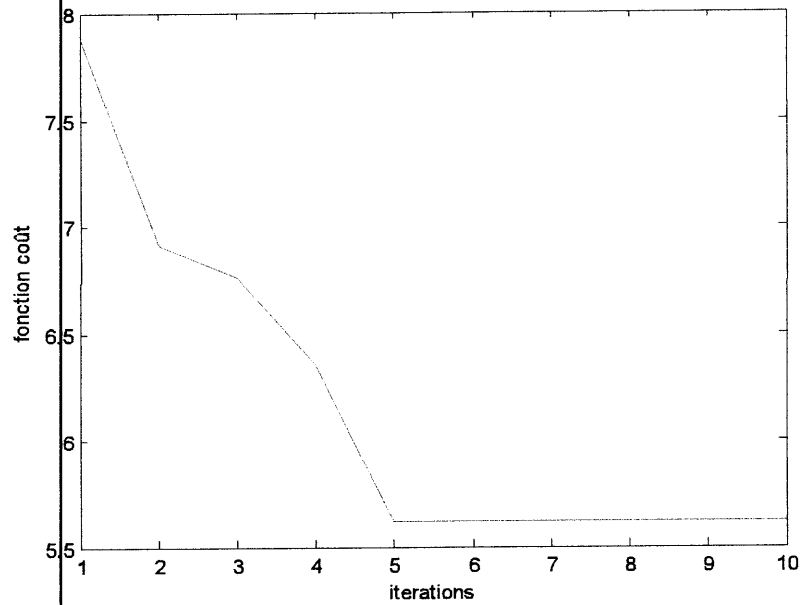


Fig III.9: Evolution de la fonction coût.

Les tableaux suivants donnent les paramètres optimaux obtenus par l'algorithme de l'optimisation par essaim particulaire (OEP) après 10 itérations :

Les points modaux des fonctions d'appartenance			
<i>e</i>	-0.7658 ( <i>a</i> <sub>1</sub> )	-0.4121 ( <i>a</i> <sub>2</sub> )	0.9520 ( <i>a</i> <sub>3</sub> )
$\Delta e$	-0.5872 ( <i>b</i> <sub>1</sub> )	-0.2572 ( <i>b</i> <sub>2</sub> )	0.6500 ( <i>b</i> <sub>3</sub> )

Tab III.3 : Paramètres des prémisses optimaux obtenus par l'OEP.

Les conclusions des règles		
N	Z	P
-0.7179 ( <i>r</i> <sub>1</sub> )	0.1872 ( <i>r</i> <sub>2</sub> )	0.9300 ( <i>r</i> <sub>3</sub> )

Tab III.4 : Paramètres des conclusions de règles floues optimaux obtenus par l'OEP.

La figure (III.10) représente les formes des fonctions d'appartenance des entrées et de la sortie du contrôleur trouvées après optimisation :

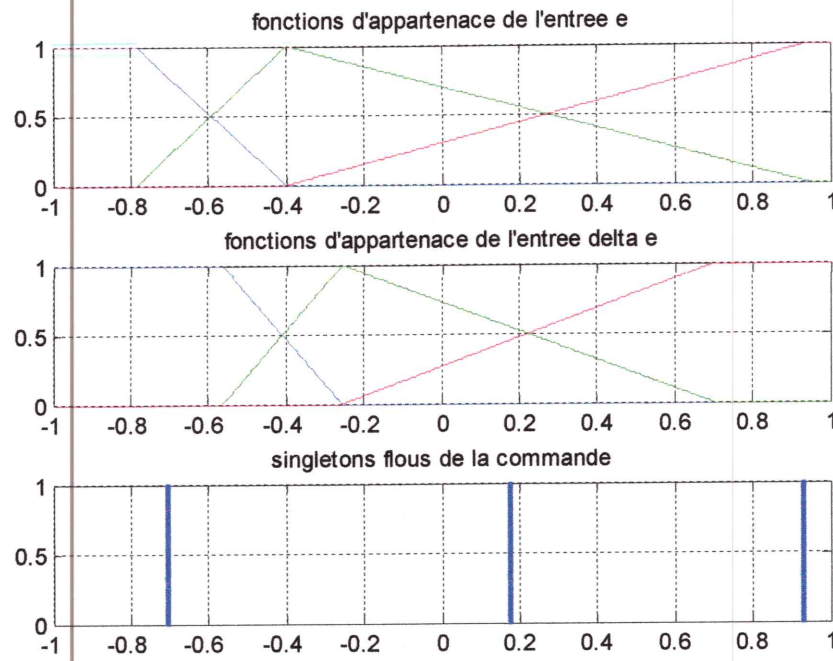


Fig III.10 : Disposition et formes des fonctions d'appartenance des prémisses et des conclusions après optimisation.

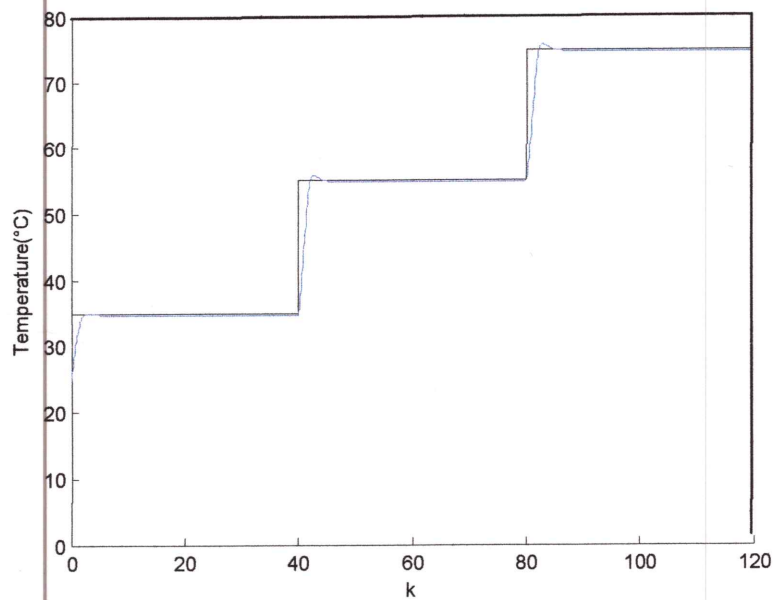


Fig III.11 : Evolution de la température.

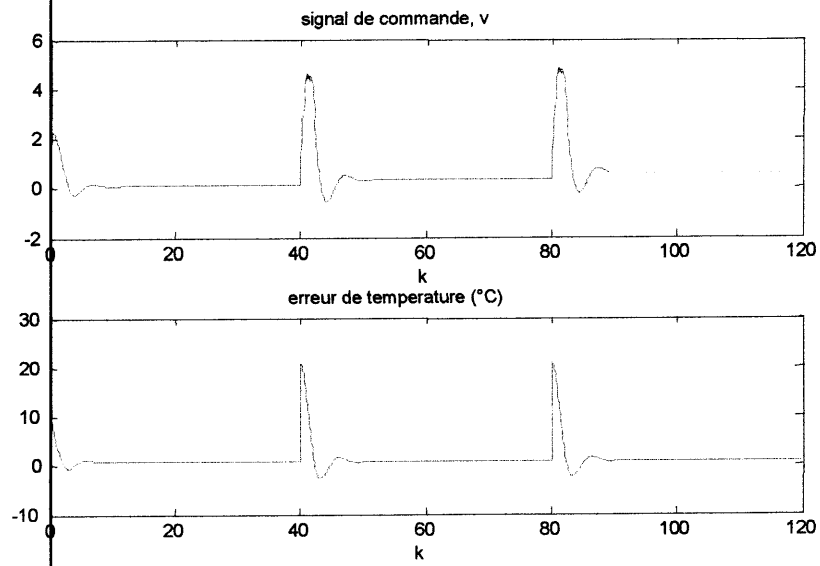


Fig III.12 : Evolution de la commande et l'erreur de poursuite de la température.

❖ Test de robustesse

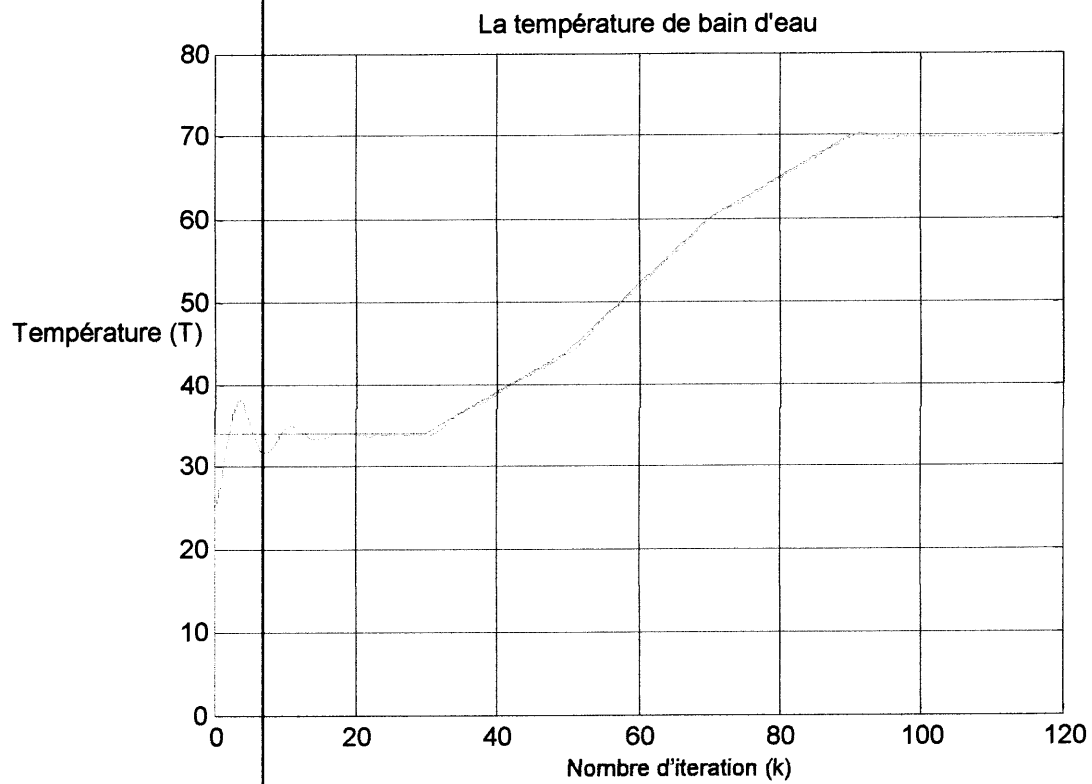


Fig III.13 : Test de robustesse en changeant la trajectoire de référence.

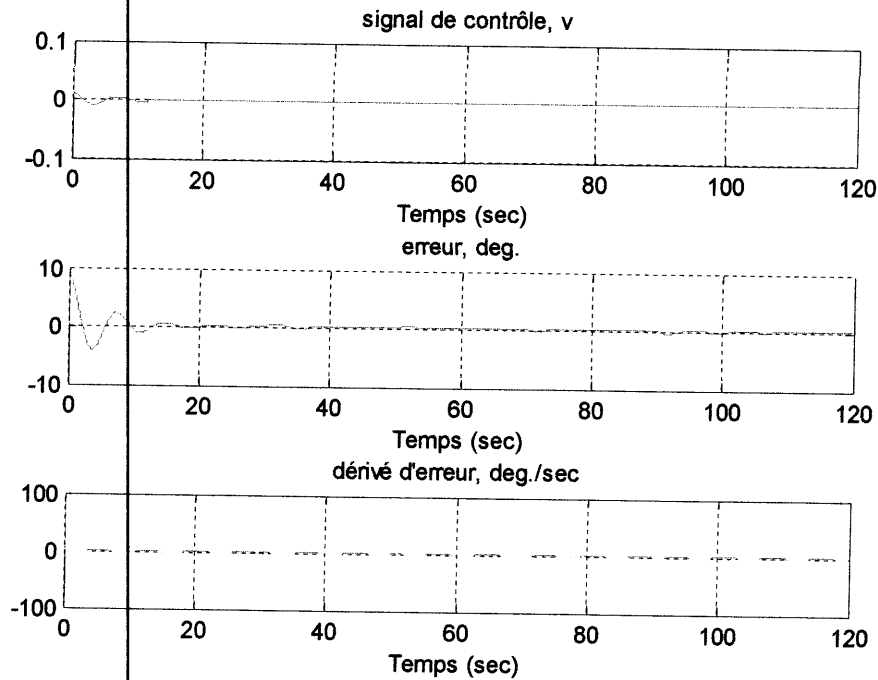


Fig III.14 : Evolution de la commande et l'erreur de poursuite de la température.

#### III.4.4 Interprétation des résultats

Dans cette partie du chapitre on a réalisé l'optimisation des paramètres du contrôleur flou de type Takagi-Sugeno à conclusion constante. L'optimisation consiste à construire automatiquement les paramètres des prémisses (les valeurs modales des fonctions d'appartenance triangulaires), et les paramètres des conclusions de règles floues (dans notre cas des singletons). Les résultats obtenus des paramètres optimaux sont présentés respectivement sur les figures (III.10).

Les figures (III.11) et (III.12) montrent, respectivement, l'évolution de la température et la commande appliquée pour le système nominal contrôlé. Le test de robustesse illustre la bonne poursuite de la température commandée par le contrôleur flou optimisé par l'OEP. Ainsi, les bonnes performances sont obtenues; en particulier, le temps de réponse réduit et dépassement minimum noté au niveau de la trajectoire de sortie ainsi un nombre réduit d'itérations ce qui minimise le temps de calcul.

De ce qui précède, on peut conclure que l'utilisation de l'algorithme d'optimisation par essaim particulaire pour optimiser les paramètres du contrôleur flou donne de meilleurs résultats du côté précision et de bonnes performances du côté asservissement.

### III.5 Conclusion

Les résultats obtenues par l'OEP sont très satisfaisant et confirment bien la validité de l'algorithme, sa simplicité d'implémentation lui donne un avantage conséquent, néanmoins par l'expérimentation, on a remarqué que la qualité des solutions ne peut être garantie même en augmentant le nombre d'itérations.



## Conclusion Générale

Le travail présenté dans ce mémoire concerne la commande d'un système non linéaire monovariante en utilisant les techniques de l'automatique telles que la logique floue et les outils d'optimisation tels que l'optimisation par essaim particulaire (OEP).

Les algorithmes de réglage se basent sur des réglages linguistiques de la forme *SI ... Alors*. Cependant, L'obtention d'un contrôleur flou optimal par un réglage manuel n'étant pas systématique, nous avons donc développé une méthode d'optimisation des caractéristiques d'un contrôleur flou basée sur l'utilisation de l'OEP.

Le *premier chapitre*, nous avons présenté les concepts fondamentaux de la logique floue, ainsi que de la commande floue. La structure d'un régulateur par la logique floue est analysée, ainsi que les deux types de régulateurs les plus connus ceux de Sugeno et de Mamdani sont présentés.

Le *deuxième chapitre*, nous avons présenté les différentes méthodes de métaheuristiques, parmi eux, on a présenté la méthode de l'OEP. On a décrit l'algorithme de calcul qui se base sur la notion de voisinage, pour trouvé l'optimum global d'un problème donné.

Le *dernier chapitre* est divisé en deux parties : Dans la première partie, une commande de la température du bain d'eau par logique floue à été synthétisée. Elle a permis la poursuite de trajectoire de référence de la température désirée.

Dans la deuxième partie, nous avons établi une stratégie d'optimisation des paramètres du contrôleur flou basé sur l'utilisation de l'OEP, on s'intéresse à optimiser les paramètres des prémisses et des conclusions de règles. Les résultats de simulation on montrés que l'OEP a amélioré les performances du régulateurs flou dans un temps de calcul minimal et avec une bonne précision.

L'approche d'optimisation basée sur l'utilisation l'algorithme par essaim particulaire présente de meilleures performances, elle est également plus robuste et plus systématique et facile à implémenter.

A l'issue de ce mémoire, notre travail ouvre de nouvelles perspectives de recherche dont nous citons :

- ✓ L'étude d'une approche hybride entre les métaheuristiques pour l'optimisation des contrôleurs flous avec l'établissement d'une comparaison avec la méthode présentée dans ce mémoire.
- ✓ L'application de la méthode présentée sur des systèmes réels, tels que le pendule inversé, le robot mobile, les machines électriques, .... etc.
- ✓ Optimisation des contrôleurs flous par d'autres algorithmes évolutionnaires cités dans le deuxième chapitre (ACO, BCO, etc).

## Bibliographie

- [1] : C.C. Lee, « Fuzzy logic in contrôl systèmes : fuzzy logic controller.part 1 », 1990.
- [2] : Claude Rsental, « Histoire de la logique floue », une approche sociologique de pratique de démonstration, revue de synthèse, octobre- décembre 1998.
- [3] : P.Clara, S.Akim and C.Simon, « HE03 introduction à la logique floue », Automne 2011, Université de technologie compiègne.
- [4] : Y.Morere, « Les réseaux neuro-flous », 17 mai 2001.
- [5] : N.Oudah, O.Azouaoui and M.hamerlain, « Implémentation d'un controleur flou pour la navigation d'un robot mobile de type voiture », 17 novembre 2005.
- [6] : L.Baghli, « contribution à la commande de la machine asynchrone, utilisation de la logique floue, des réseaux de neurons et des algorithmes génétiques », Thèse de doctorat de l'université Henri Poincaré, Nancy.I en Génie Electrique, 14 janvier 1999.
- [7] : A.Samuel, « Théorie des ensembles flous : application à la mesure de la pauvreté au congo », 16-2003.
- [8] : Hamid.Ouakka, « Contibution à l'indentification et la commande floue d'une classe de système non linéaire », Thèse de doctorat, 27 juin 2009.
- [9] : O.Guencounou, «éthodologie de conception de contrôleurs intelligents par l'approche génétique-application à un bioprocédé », Thèse de doctorat, Université Toulouse, Le 22 avril 2009.
- [10] : El-Hadi.Debaa, « Génération automatique et optimisation de système à inférence floue », Département Informatique et Robotique, Université paris XII, 18 décembre 2003.
- [11] : J.Faucher, « les plans d'expérience pour le réglage de commande à la base de logique floue », en Génie Electrique et Automatique, 26 septembre 2006.
- [12] : Y.Nakoula, « Apprentissage des modèles linguistique flous, par jeu de règles ponderees », Thèse de doctorat, en Electronique-Electrotechnique-Automatique, Université Paul Sabatier, Toulouse, 30 mars 1992.

- [13] : Projet de recherche, accord de coopération + école d'été en automatique à Bucarest cours « logique floue et commande », Université polytechnique de Bucarest, 21 mai 2012.
- [14] : S.Galichet, « Contrôleur de l'interpolation numérique au codage de l'expertise », En Université de Savoie, 21 décembre 2001.
- [15] : Jean Henri Gwèt, « Contribution à l'analyse des données floue », Thèse de doctorat d'état, En Mathématique, Université de Yaound , 13 février 1998.
- [16] : V. Angel, « La rugosité des paysages : une théorie pour la difficulté des problèmes d'optimisation combinatoire relativement aux métaheuristiques », Thèse de doctorat de l'université de Paris-Sud, Orsay, 1998.
- [17] : Abbas El Dor, « Perfectionnement des algorithmes d'optimisation par essaim particulaire : applications en segmentation d'images et en électronique ». Other. Université Paris-Est, 2012.
- [18] : G.Berthiau and P. Siarry, « Etat de l'art des méthodes d'optimisation globale », RAIRO Operations Research, Vol. 35, N°3, p. 329365, 2001.
- [19]: E.Bonabeau, M. Dorigo and G. Theraulaz, « Swarm Intelligence: from natural to artificial systems », Oxford University Press, 1999.
- [20]: M.Clerc and J. Kennedy, « The particle swarm explosion, stability, and convergence in a multidimensionnel complexscape », IEEE Transactions on Evolutionary Computation, Vol. 6, p. 5873, 2002.
- [21]: Maurice Clerc and Patrick Siarry, « Une nouvelle métaheuristique pour l'optimisation difficile », la méthode des essaims particulaires.
- [22]: Guillaume CALA, « Optimisation par essaim de particules ».
- [23] : EUDES et RIOLAND, « Optimisation par essaim particulaire pour un problème d'ordonnement et d'affectation de ressources », 2007.
- [24] : C.F. Juang, "A TSK-Type Recurrent Fuzzy Network for Dynamic Systems Processing by Neural Network and Genetic Algorithms", *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 2, April 2002, pp. 155-170.