

N° d'ordre : .....



جامعة محمد بوضياف - المسيلة  
Université Mohamed Boudiaf - M'sila  
كلية التكنولوجيا  
Faculté de Technologie  
قسم الإلكترونيك  
Département d'Electronique



## MEMOIRE DE MASTER

Présenté par : Roumair Asma & Kahoul Ranyia

DOMAINE : SCIENCES ET TECHNOLOGIES

FILIERE : ELECTRONIQUE

OPTION : SYSTEMES DE TELECOMMUNICATIONS

### Thème

## Modélisation et Identification Hybride des Systèmes Dynamiques en Utilisant la Logique floue Type 1 et l'Algorithme d'Optimisation PSO

Devant le jury composé de :

- |                        |     |             |                      |
|------------------------|-----|-------------|----------------------|
| ➤ BENNACER Hamza       | MCA | Univ.M'sila | <b>Président</b>     |
| ➤ OUALI Mohammed Assam | MCA | Univ.M'sila | <b>Rapporteur</b>    |
| ➤ LADJAL Mohamed       | MCA | Univ.M'sila | <b>Co-Rapporteur</b> |
| ➤ ATTALLAH Bilal       | MCA | Univ.M'sila | <b>Examineur</b>     |

2020/2021

## **REMERCIEMENTS**

*Nous remercions, spécialement **mon bon dieu** pour la volonté, la santé, le courage et la patience qu'il nous donnés pour mener ce travail à terme.*

*Nous adressons nos sincères remerciements à Monsieur **Mr. OUALI Mohammed Assam** et à **Mr. LADJAL Mohamed**, pour nous avoir permis de bénéficier de leurs grandes savoir dans la matière, pour leur pédagogiques, leurs compétences, leurs modesties et leur aide précieuse tout au long de ce projet même pendant les moments les plus difficiles. Vraiment merci pour une qualité d'accompagnement si sérieuse et si consistante.*

*Nous remercions très sincèrement monsieur le professeur **BENNACER Hamza** de nous avoir fait l'honneur d'accepter de présider le jury de cette mémoire.*

*Nous exprimons également nos remerciements aux membres de jury messieurs ; le docteur **ATTALLAH Bilal** pour avoir accepté d'examiner et de juger ce travail.*

*Nous remercions chaleureusement nos parents qui no sont soutenu tout au long de nos études, parfois au prix de quelques sacrifices, et sans qui nous n'aurais pas pu atteindre ce niveau.*

*Enfin, nous adresse nos remerciements à tous ceux qui ont contribué, de près ou de loin, à la réussite de ce travail.*

## **DEDICACES**

*C'est avec profonde gratitude et sincères mots, que nous dédie ce modeste travail de fin d'étude à nos chers parents ; qui ont sacrifié leur vie pour nos réussites et nos 'ont éclairé le chemin par leurs conseils judicieux.*

*Nos 'espères qu'un jour, nous pourrai leurs rendre un peu de ce qu'ils ont fait pour nous, que dieu leur prête bonheur et longue vie.*

*Nos 'espères que ce travail soit un exemple pour vous de persévérance, de courage et de générosité.*

*En fin, Nous tiens de dédies ce travail à toutes nos familles «**KAHOUL** » et «**ROUMAIR** » proches et collègues, qui nos 'ont toujours soutenu et en courgées au cours de la réalisation de ce mémoire. A tous ceux que nous ne puis citer ici mais qui savent combien d'avoir croisé leur route m'apporté.*

*« Le succès n'est pas la clé du bonheur. Le bonheur est la clé de succès. Si vous aimez ce que vous faites-vous réussirez. »*

*Lbert Schweitzer*

---

# *Sommaire*

---

# Sommaire

<b>REMERCIEMENTS</b> .....	I
<b>DEDICACES</b> .....	II
<b>Introduction générale</b> .....	1
<b>Chapitre I : Modélisation des Processus</b> .....	5
Introduction .....	6
Définition d'un processus et d'un modèle.....	14
Processus .....	14
Modèles .....	15
Qu'est-ce qu'un modèle ? .....	15
Modélisation statique et modélisation dynamique .....	15
Buts d'une modélisation.....	16
Classification des modèles .....	16
Choix du modèle d'un processus.....	18
Modèle à temps continu .....	18
Modèles à temps discontinu ou discret .....	19
Modèle linéaire.....	19
Modèle non linéaire.....	19
Identification des processus.....	21
Les structures d'identification.....	21
Identification parallèle.....	21
Identification série-parallèle.....	22
Conclusion.....	23
<b>Chapitre II : Logique floue type-1</b>	
Introduction .....	25
Logique floue type-1 .....	25
Définition .....	25
Historique et domaines d'application.....	26
Ensembles flous type-1 .....	27
Opérations de base sur les ensembles flous .....	28
Variables linguistiques .....	29
Système d'inférence floue.....	30

Exemple d'un système flou type-1 .....	31
Conclusion.....	32
<b>Chapitre III : L'optimisation par essaim particulaire</b>	
Introduction .....	34
Optimisation par essaim particulaire .....	34
Les différents types du PSO .....	35
PSO de l'optimum global .....	35
Principe de déplacement d'une particule dans PSO de l'optimum global .....	35
PSO de l'optimum local.....	38
Principe de déplacement d'une particule dans PSO de l'optimum local .....	39
Les Composants de vitesse.....	41
Les paramètres du PSO .....	42
La comparaison entre les deux versions du PSO .....	44
Test de l'efficacité du PSO .....	44
Les champs d'application du PSO .....	46
Conclusion.....	46
<b>Chapitre IV : Modélisation PSO-Floue Type-1des Systèmes Dynamiques</b>	
Introduction .....	49
Technique de modélisation proposée .....	49
Paramètres du modèle flou à ajuster .....	50
Fonction objectif .....	50
Identification du modèle primaire .....	51
Identification du processus d'erreur .....	52
Conception du modèle final.....	53
Résultats de simulation.....	54
Paramètres d'algorithme d'optimisation PSO.....	54
Modélisation du système I.....	54
Modélisation du système II .....	56
Modélisation du système III.....	54
Conclusion.....	58
<b>Conclusion générale .....</b>	<b>59</b>

## LISTE DES FIGURES

### Chapitre I

Figure I.1: Schéma synoptique d'un processus. ....	15
Figure I.2: Système à non linéarité séparable.....	20
Figure I.3: Structure d'identification parallèle. ....	22
Figure I.4 : Structure d'identification série-parallèle. ....	23

### Chapitre II

Figure II.1 : Comparaison d'un ensemble classique et d'un ensemble flou.....	27
Figure II.2 : Fonction d'appartenance : (a) Triangulaire, (b) Singleton, (c) Gaussienne. ....	28
Figure II.3 : Système d'inférence floue (SIF). ....	31

### Chapitre III

Figure III.1 : L'étoile topologie de voisinage. ....	35
Figure III.2 : déplacement d'une particule dans PSO de l'optimum global. ....	36
Figure III.3 : Organigramme de l'algorithme PSO de l'optimum global. ....	38
Figure III.4 : L'anneau topologie de voisinage.....	39
Figure III.5 : Principe de déplacement d'une particule dans PSO de l'optimum local.....	40
Figure III.6 : Organigramme de l'algorithme PSO de l'optimum local.....	41
Figure III.7 : Schéma vectorielle du déplacement d'une particule.....	42
Figure III.8 : Le diminue progressif en poids de l'inertie. ....	44
Figure III.9 : Fonction de Rosenbrock. ....	45
Figure III.10 : convergence de la fonction objective vers l'optimum global. ....	46

### Chapitre IV

Figure IV.1 : Ensemble flou type-2 Gaussien avec.....	50
Figure IV.2 : Identification du modèle primaire .....	51
Figure IV.3 : Processus d'erreur.....	52
Figure IV.4 : Identification du processus d'erreur .....	53
Figure IV.5 : Modèle final fF .....	54
Figure IV.6 : Résultats de modélisation du système dynamique non linéaire (Équation 8) basé sur PSO, (a) Modèle primaire, (b) Validation, (c) Modèle de l'erreur de modélisation, (d) Modèle final, (e) Erreur de Modélisation Final.....	56
Figure IV.7: Résultats de modélisation du système dynamique non linéaire (Equation. 11) ..	57
Figure IV.8 : Résultats de modélisation du système dynamique non linéaire (Equation. 14) .	58

---

# *Introduction Général*

---

## Introduction général :

Un système dynamique est un système qui se varie dans le temps d'une façon causale et déterministe, il désigne généralement la branche de recherche active des mathématiques, à la frontière de la topologie, de l'analyse, de la géométrie, de la théorie de la mesure et des probabilités. Les systèmes dynamiques sont néanmoins apparus assez tôt dans l'histoire scientifique puisqu'on peut les reconnaître dans les premiers travaux de la mécanique donnant lieu à des équations différentielles.

Les principales difficultés dans la théorie de la commande des systèmes dynamiques réels sont les non-linéarités et les incertitudes. Or la commande passe par l'élaboration d'un modèle mathématique du système en trouvant une relation entre les entrées et les sorties, ce qui suppose une bonne connaissance de la dynamique du système et ses propriétés. La problématique qui se pose est l'incapacité de prévenir le fonctionnement des systèmes dynamiques après une certaine période et pour le résoudre on peut remplacer cette fonction par la logique floue.

La logique floue a été Formalisée par Lotfi Zadeh en 1965, c'est un outil de l'intelligence artificielle, dont on peut utiliser dans des domaines aussi variés que :

- l'automatique (freins ABS, conduite de processus)
- la robotique (reconnaissance de formes)
- la gestion de la circulation routière (feux rouges)
- le contrôle aérien (gestion du trafic aérien)
- l'environnement (météorologie, climatologie, sismologie, analyse du cycle de vie)
- la médecine (aide au diagnostic)
- l'assurance (sélection et prévention des risques) et bien d'autres

L'application des techniques floues pour l'identification et le contrôle des systèmes non- linéaires peut fournir de nouvelles solutions pour ce problème, qui est l'identification floue et le contrôle flou, alors le mot « identification » est un ensemble de techniques dont l'objet est la détermination de modèle d'attitude d'un procédé physique à partir de mesures caractéristiques de son fonctionnement dynamique.

L'optimisation est un paradigme actuel dans chaque aspect de la vie. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation. Bien que les problèmes d'optimisation soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes complexe, stochastique, ou fortement non-linéaire et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données.

L'optimisation par essaim particulaire (PSO) est classée dans les techniques d'optimisation stochastiques à population introduite en 1995 par *James Kennedy* (psychologue social et statisticien) et *Russel Eberhart* (ingénieur) sous le nom Particle Swarm Optimization (PSO). PSO est inspiré par le comportement social des animaux, comme les essaims d'abeilles, les vols d'oiseaux et les bancs de poissons. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture, ou émigrent, Si un membre voit un chemin souhaitable pour l'exploration, le reste de cet essaim suivra rapidement. Raison pour laquelle, cette technique est fondée sur la notion de coopération et l'échange d'information entre les particules. Les individus de l'algorithme sont appelés particules et la population est appelée essaim.

Un modèle flou peut représenter ou modéliser n'importe quel système ou fonction inconnu,  $y = f(x)$ , en utilisant les données d'entrée/sortie. L'objectif de ce travail est d'offrir un modèle flou fiable et efficace pour représenter les systèmes dynamiques. Dans un premier temps un modèle primaire a été initialement conçu à l'aide des données réelles d'entrée/sortie du système dynamique en considération. Ensuite, l'erreur entre la sortie du système dynamique en considération et la sortie du modèle primaire a été modélisée pour créer le modèle d'erreur. Ce dernier modèle représente les incertitudes dans le modèle primaire qui peuvent être facilement supprimés par une simple soustraction de la sortie du modèle d'erreur de la sortie du modèle Primaire.

Les modèles flous utilisessont de type Takagi-Sugeno Kang (TSK) dont l'idée est de trouver une relation entre l'entrée et la sortie du modèle flou en ajustant au fur et à mesure ses paramètres en utilisant des mécanismes d'ajustement (algorithme d'optimisation) de telle façon une fonction objectif atteint son minimum. Pour avoir l'objectif voulu notre méthode sera composée de trois étapes

- *Étape 1 : identification du modèle primaire,*

- *Étape 2 : identification du processus d'erreur,*
- *Étape 3 : conception du modèle final.*

Le présent mémoire est organisé autour de quatre chapitres :

- **Le premier chapitre** : donne un aperçu général sur les processus et leurs modélisations.
- **Le deuxième chapitre** sera consacré à présenter les notions de base relatives à la logique floue
- **Dans Le troisième chapitre nous allons** présenter les notions de base relatives à l'optimisation par essaim particulaire (PSO),
- **Le Chapitre quatre** sera consacré donc à présenter une structure de modélisation des systèmes dynamiques à base de la logique type-1 et l'algorithme d'optimisation PSO, ainsi qu'aux résultats de simulation trouvés.

Nous terminerons notre mémoire par une conclusion générale, où nous rappellerons les principaux résultats obtenus dans cette étude.

---

# CHAPITRE I

## *Modélisation des Processus*

---

*La modélisation des systèmes dynamiques, elle a pour objectif de trouver une représentation mathématique, aussi simple et compacte que possible, qui permet de rendre compte les relations existantes entre leurs entrées et leurs sorties. Nous exposons dans ce chapitre les notions de base relatives à la modélisation des processus.*

### **Sommaire**

---

- 1. Introduction**
  - 2. Définition d'un processus et d'un modèle**
  - 3. Choix du modèle d'un processus**
  - 4. Identification des processus**
  - 5. Conclusion**
-

## 1. Introduction

Les systèmes réels sont difficiles à étudier, donc on est amené à les modéliser mathématiquement pour pouvoir les commander [1]. La modélisation d'un processus consiste à trouver une description mathématique de son fonctionnement, qui permet de rendre compte des relations existantes entre ses entrées et ses sorties, et qui sont habituellement représentées par des équations. Si ces équations sont algébriques, le modèle est dit statique. Si ces équations sont des équations différentielles ou des équations aux différences récurrentes, le modèle est dit dynamique, respectivement à temps continu ou à temps discret [2].

## 2. Définition d'un processus et d'un modèle

### Processus

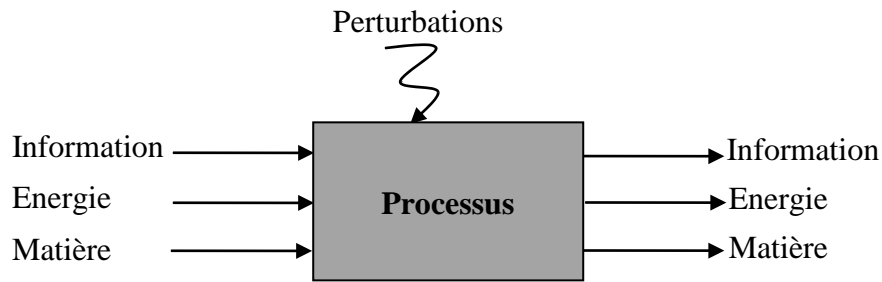
Un processus est un système dynamique qui évolue dans le temps. Du point de vue d'un observateur, un processus correspond à un système physique envisagé dans le cadre de l'évolution des échanges réalisés avec son environnement.

Un processus est caractérisé par :

- Une ou plusieurs grandeurs de sortie, mesurables, qui constituent le résultat du processus,
- Une ou plusieurs grandeurs d'entrée (ou facteurs), qui peuvent être de deux types :
  - des entrées sur lesquelles il est possible d'agir (entrées de commande),
  - des entrées sur lesquelles il n'est pas possible d'agir (perturbations) ; ces dernières peuvent être aléatoires ou déterministes, mesurables ou non mesurables.

Les processus peuvent être de toutes natures : physique, chimique, biologique, écologique, financier, sociologique, .... etc [3].

Dans le cas général, un processus est un système dynamique traversé par des flux d'informations, d'énergie et de matière tout en étant soumis à des perturbations ayant l'une des trois formes précitées. La figure 1, fournit un exemple d'une telle représentation.



**Figure 1** : Schéma synoptique d'un processus.

L'étude des processus a généralement pour objectif de :

- ✓ Pouvoir comprendre le fonctionnement du système et prévoir son comportement et ses performances face à une variation des entrées ; on parle alors d'analyse du système,
- ✓ Chercher à maîtriser les sorties et les performances du système en agissant sur les entrées ; il s'agit, dans ce cas, de synthèse de lois de commande.

### Modèles

#### *Qu'est-ce qu'un modèle ?*

Un modèle est caractérisé par son domaine de validité, c'est-à-dire par le domaine de l'espace des entrées dans lequel l'accord entre les valeurs des sorties du processus calculées par le modèle, et leurs valeurs mesurées, est considéré comme satisfaisant compte tenu de l'utilisation que l'on fait du modèle. Un modèle est dit statique s'il est représenté par des équations mathématiques algébriques. Si ces équations sont des équations différentielles ou des équations aux différences récurrentes, le modèle est dit dynamique, respectivement à temps continu ou à temps discret [2].

#### *Modélisation statique et modélisation dynamique :*

- Un système statique a une sortie  $y(t)$  indépendante des valeurs antérieures de l'entrée  $u(T)$  avec  $T < t$ , pour tout  $t$ . La modélisation statique est un modèle qui réalise une relation algébrique entre ses entrées et ses sorties. Ce modèle est utilisé pour prédire les états stationnaires de la sortie d'un processus industriel ou pour relier des grandeurs qui sont indépendantes du temps.
- Un système dynamique a une sortie  $y(t)$  qui dépend des valeurs antérieures de l'entrée  $u(T)$  avec  $T < t$ . Pour la modélisation dynamique les entrées et les sorties sont reliées entre elles ; soit par des équations différentielles (modèle à temps continu) soit par des équations récurrentes ou aux différences (modèle à temps discret). Le pont diviseur de tension est donc un système statique, alors que le système masse-ressort est un système dynamique, qui nécessite de garder en mémoire les

valeurs antérieures de son entrée. Est également possible de classer les systèmes dynamiques dans deux catégories [2].

### ***Buts d'une modélisation***

Un modèle peut être utilisé soit :

- Pour simuler un processus : à des fins pédagogiques, de détection d'anomalies de fonctionnement, de diagnostic de pannes, de conception assistée par ordinateur, etc.,
- Pour effectuer la synthèse d'une loi de commande, ou pour être incorporé dans un dispositif de commande.

### ***Classification des modèles***

#### ***a. Classification selon le mode de conception***

On distingue trois sortes de modèles en fonction des informations mises en jeu pour leur conception.

- ✓ **Les modèles de connaissance** : les modèles de connaissance sont construits à partir d'une analyse physique, chimique, biologique (ou autre suivant le type du processus), en appliquant soit les lois générales, fondées sur des principes (lois de la mécanique, de l'électromagnétisme, de la thermodynamique, de la physique quantique, etc.), soit les lois empiriques (finance, économie), qui régissent les phénomènes intervenant au sein des processus étudiés. Ces modèles ne comportent généralement pas de paramètres ajustables, ou des paramètres ajustables en très petit nombre. Dans la pratique, il est toujours souhaitable d'établir un modèle de connaissance des processus que l'on étudie. Néanmoins, il arrive fréquemment que le processus soit trop complexe, ou que les phénomènes qui le régissent soient trop mal connus, pour qu'il soit possible d'établir un modèle de connaissance suffisamment précis pour l'application considérée. On est alors amené à concevoir des modèles purement empiriques, fondés exclusivement sur les résultats de mesures effectuées sur le processus.
- ✓ **Les modèles « boîte noire »** : les modèles "boîte noire" sont construits essentiellement sur la base de mesures effectuées sur les entrées et les sorties du processus à modéliser. La modélisation consiste alors à utiliser, pour représenter les relations entre les entrées et les sorties, des équations (algébriques, différentielles, ou récurrentes) paramétrées, et à estimer les

paramètres, à partir des mesures disponibles, de manière à obtenir la meilleure précision possible avec le plus petit nombre possible de paramètres ajustables. Le domaine de validité d'un tel modèle ne peut pas s'étendre au-delà du domaine des entrées qui est représenté dans les mesures utilisées pour l'apprentissage.

- ✓ **Les modèles « boîte grise »** : lorsque des connaissances, exprimables sous forme d'équations, sont disponibles, mais insuffisantes pour concevoir un modèle de connaissance satisfaisant, on peut avoir recours à une modélisation "boîte grise" (ou modélisation semi physique) qui prend en considération à la fois les connaissances et les mesures. Une telle démarche peut concilier les avantages de l'intelligibilité d'un modèle de connaissance avec la souplesse d'un modèle comportant des paramètres ajustables.

***b. Classification selon l'utilisation.***

Indépendamment de la classification précédente, on peut distinguer deux types de modèles en fonction de l'utilisation qui en est faite.

- ✓ **Les modèles de simulation (ou simulateurs)** : un modèle de simulation est utilisé de manière indépendante du processus qu'il représente. Il doit donc posséder un comportement aussi semblable que possible à celui du processus. De tels modèles sont utilisés pour valider la conception d'un système avant sa fabrication (conception assistée par ordinateur en mécanique, en microélectronique, ..), pour la formation de personnels (simulateurs de vols), pour la prévision à long terme, etc. Du point de vue de la structure du modèle, les sorties passées, mesurées sur le processus à modéliser, ne peuvent constituer des entrées du modèle. L'estimation des paramètres et l'utilisation du modèle constituent deux phases successives et distinctes (apprentissage non adaptatif).
- ✓ **Les modèles de prédiction (ou prédicteurs)** : un modèle de prédiction est utilisé en parallèle avec le processus dont il est le modèle. Il prédit la sortie du processus à une échelle de temps courte devant les constantes de temps du processus. Les prédicteurs sont utilisés pour la synthèse de lois de commande, ou dans le système de commande lui-même (commande avec modèle interne). Du point de vue de la structure du modèle, les sorties passées, mesurées sur le processus, peuvent constituer des entrées du modèle. L'estimation des paramètres et l'utilisation du modèle peuvent être effectuées simultanément si

nécessaire (apprentissage adaptatif, utile notamment si les caractéristiques du processus dérivent dans le temps).

### 3. Choix du modèle d'un processus

La détermination d'un modèle mathématique d'un processus nécessite en général diverses hypothèses simplificatrices afin de limiter sa complexité. Dans chaque application, il apparaît nécessaire de faire un compromis entre la finesse et la précision du modèle à mettre en œuvre d'une part, et la limite de complexité admissible, compte tenu des objectifs fixe, d'autre part. Nous nous intéressons ici essentiellement aux processus pour lesquels les variables caractéristiques sont susceptibles de prendre un ensemble continu de valeurs appartenant à des intervalles fixés. De plus nous ne traiterons que des modèles de processus à paramètres localisés, c'est-à-dire, décrits par des équations différentielles ordinaires ou des équations récurrentes.

Il est important de toujours se souvenir que le choix du modèle d'un processus dépend de l'utilisation prévue pour ce modèle.

#### Modèle à temps continu

Dans ce type de modèle, le temps est une variable qui évolue continûment sur un intervalle  $T \subset R$  prenant en croissant toutes les valeurs situées dans cet intervalle. C'est le cas par exemple d'un modèle dans l'espace d'état de la forme :

$$\begin{aligned} \dot{x} &= f(x, u, t, v) \\ y &= h(x, u, t, w) \end{aligned} \quad (1)$$

où  $x$  représente la dérivée totale  $dx/dt$  du vecteur d'état  $x \in R^n$  par rapport au temps,  $u \in R^l$  le vecteur d'entrée,  $t \in T$  le temps,  $v \in R^{n_v}$  un vecteur de perturbations, et  $y \in R^m$  le vecteur des sorties.

Dans cette représentation on a :

$$\begin{aligned} f &: R^n \times R^l \times T \times R^{n_v} \rightarrow R^n, \\ h &: R^n \times R^l \times T \times R^{n_v} \rightarrow R^m, \\ T &= [t_0, t_0 + T_e], t_0 \in R, t_0 \dots \infty, T_e \notin R^+ \end{aligned} \quad (2)$$

Dans beaucoup de problème, on prend  $t_0 = 0$  ou  $t_0 \neq 0$  mais cette restriction n'est en aucun nécessaire a priori.

### Modèles à temps discontinu ou discret

Ce type de modèle est utilisé soit lorsque le processus étudié a naturellement un mode d'évolution séquentiel, soit lorsque-on a adopté une représentation du processus qui correspond à une observation des variables d'état ou de sortie à des instants discrets  $t_k$  du temps avec  $k \in \mathbb{Z}$ .

On a dans ce cas une représentation dite échantillonnée, particulièrement bien adoptée aux divers types de calculs à effectuer dans les problèmes de commande de processus par calculateur numérique.

notant :

$$x_k = x(t_k). \quad (3)$$

Alors, la description continue devient :

$$\begin{aligned} x_{k+1} &= f(x_k, u_k, k, v_k), \\ y_k &= h(x_k, u_k, k, w_k). \end{aligned} \quad (4)$$

### Modèle linéaire

Un tel modèle caractérise un processus susceptible d'être décrit par une équation différentielle ordinaire ou une équation récurrente à coefficients constants (stationnaire) ou non constants (nonstationnaire) ou par un ensemble d'équation de ce type.

La propriété fondamentale des systèmes linéaires s'exprime par le principe de superposition : si  $y_1(t)$  et  $y_2(t)$  représentent respectivement l'évolution des sorties du processus pour les entrées  $u_1(t)$  et  $u_2(t)$  sur un horizon et pour un état initial donné, alors, à l'entrée  $u(t)$  :

$$u(t) = \alpha_1 u_1(t) + \alpha_2 u_2(t), \quad (5)$$

où  $\alpha_1$  et  $\alpha_2$  sont des constantes réelles, correspond la sortie :

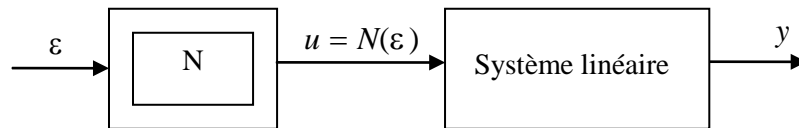
$$y(t) = \alpha_1 y_1(t) + \alpha_2 y_2(t). \quad (6)$$

sur le même horizon et pour les mêmes conditions initiales.

### Modèle non linéaire

Dans un modèle non linéaire, le théorème de superposition n'est plus valable. L'aspect non linéaire peut être intrinsèque et quasiment irréductible, comme dans la modélisation de la loi d'action de masse en chimie, ou correspondre des éléments à caractéristiques non linéaires à des systèmes linéaires comme dans le cas des systèmes à non linéarités séparables (figure .2).

Dans certains cas, lorsque le processus non linéaire est utilisé dans une plage de variation limitée de ses variables d'état, alors l'évolution sera effectuée autour d'une valeur  $x_0$  donnée, il est donc possible d'effectuer une linéarisation autour de ce point de fonctionnement en prenant comme nouvelle variable l'expression  $\Delta x = x - x_0$ .



**Figure. 2 :** Système à non linéarité séparable.

Dans ce cas linéaire, on néglige alors dans le développement limité de la fonction  $f(\cdot)$  les termes d'ordre supérieurs à 1.

Il vient par exemple pour la relation :

$$x = f(x, u, t), \quad (7)$$

si la fonction  $f$  admet des dérivées partielles continues au premier ordre, le modèle linéarisé (8) sera valable pour  $(x, u)$  évoluant dans un voisinage de  $(x_0, u_0)$  :

$$\begin{aligned} f(x, u) &= f(x_0, u_0) + (x - x_0) f'_x(x_0, u_0) + (u - u_0) f'_u(x_0, u_0) \\ &= f(x_0, u_0) + \Delta x F_x + \Delta u F_u \end{aligned} \quad (8)$$

avec :  $\Delta x = x - x_0$ ,  $\Delta u = u - u_0$ , et :

$$F_x = \left\{ \frac{\partial f}{\partial x} \right\}, F_u = \left\{ \frac{\partial f}{\partial u} \right\} \quad (9)$$

Sont les matrices jacobiennes de  $f$  par rapport à  $x$  et à  $u$ .

Pour certains processus non linéaires et/ou non stationnaires, on adopte aussi parfois une représentation multi modèle, chaque modèle étant représentatif de l'évolution du processus dans un domaine limite de l'espace et du temps.

Ainsi un système linéaire non stationnaire peut être représenté par un ensemble de modèle linéaires à coefficients constants se succédant au cours du temps.

#### 4. Identification des processus

L'identification est la détermination, sur la base de la connaissance d'un nombre fini d'entrées-sorties du système, d'un modèle appelé modèle d'identification, qui soumit aux mêmes entrées que le système fournit des sorties suffisamment proches de celui-ci. L'identification consiste à déterminer un ensemble d'équations –un modèle- décrivant le mieux possible le procédé. Il y a deux étapes d'identification, la première consiste à fixer la forme des équations, c'est l'étape qualitative, ou caractérisation, la seconde consiste à Trouver les valeurs numériques des coefficients qui interviennent dans ces équations, c'est l'étape quantitative, ou estimation des paramètres. Ces valeurs numériques sont déterminées pour que le comportement de modèle soit le plus proche de celui du système [1].

##### Les structures d'identification

La sortie d'un système dynamique dépend de son entrée et de son état ultérieur, c'est pour cela qu'il existe deux classes de modèle d'identification.

##### *Identification parallèle*

Dans le cas d'un système dynamique à temps discret, la sortie du modèle est calculée à partir de ses entrées et ses sorties passées :

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1); u(k), u(k-1), \dots, u(k-m+1)] \quad (10)$$

Certaines précautions doivent être prises lors de l'identification, la principale étant d'utiliser des entrées bornées et sorties bornées, le système reste stable, il est dit BIBO (*Bonded Input Bonded Output*).

L'inconvénient de l'identification parallèle est que même avec un système borné, rien ne garantit que les paramètres vont converger et que l'erreur tend vers zéro [1].

La structure d'un identificateur parallèle est donnée par la figure suivante :

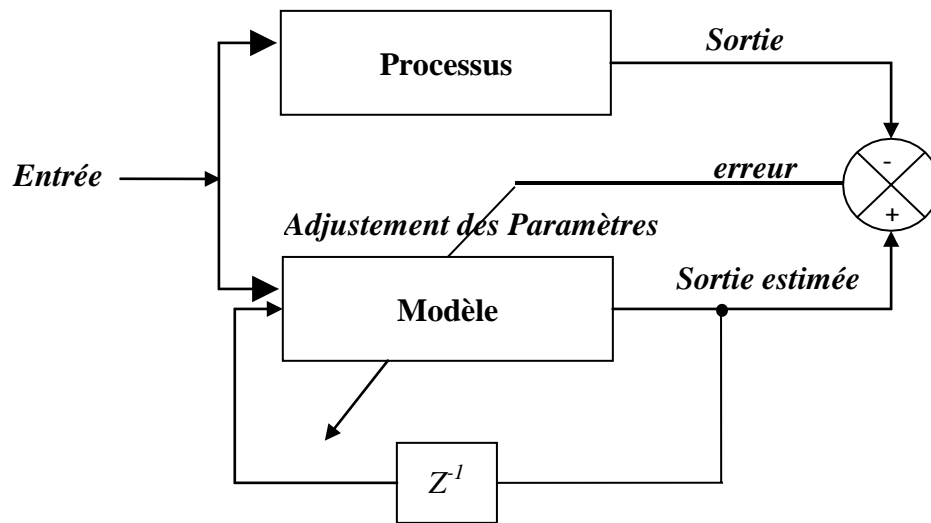


Figure. 3 : Structure d'identification parallèle.

*Identification série-parallèle*

La sortie du modèle est calculée à partir de ses entrées et la sortie du système à identifier :

$$y'(k + 1) = f[y(k), y(k - 1), \dots, y(k - n + 1); u(k), u(k - 1), \dots, u(k - m + 1)] \quad (11)$$

Ce modèle a plus de chances de converger, car tous les signaux utilisés lors de l'identification sont bornés [1]. La structure d'un identificateur série-parallèle est donnée par la figure suivante :

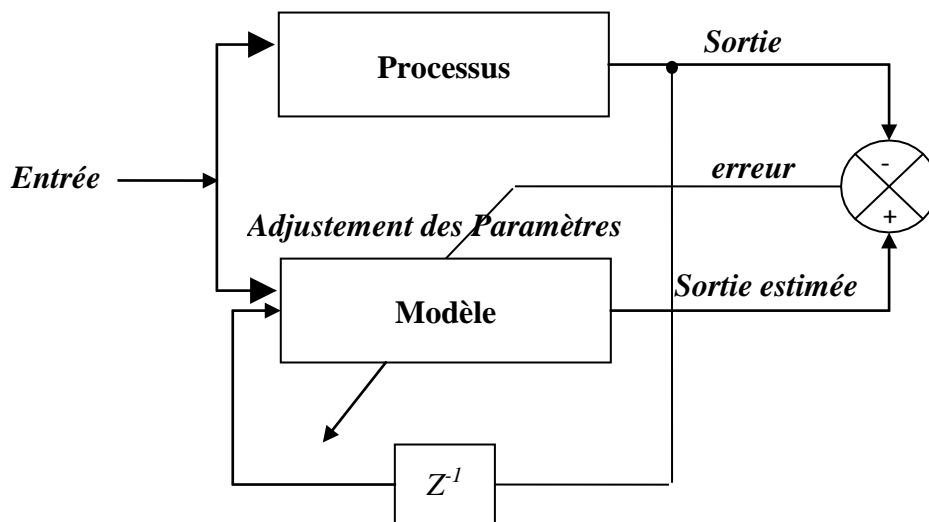


Figure. 4 : Structure d'identification série-parallèle.

## 5. Conclusion

Ce chapitre à été consacré aux notions de base relatives au processus et la modélisation, ou nous avons introduits quelques définitions, l'objectif de la modélisation, types des modèles et comment choisir un modèle. Le chapitre suivant sera consacré à la présentation des notions de base relatives aux algorithmes d'optimisations.

---

## CHAPITRE II

### *Logique Floue Type-1*

---

*La logique floue a été étudiée de manière approfondie dans la littérature dans le but de prouver son efficacité. Cet outil sera utilisé dans notre mémoire de fin d'étude pour la modélisation, l'identification, des systèmes dynamiques. Nous exposons dans ce premier chapitre les notions liées à la logique floue.*

#### **Sommaire**

---

- 1. Introduction**
  - 2. Logique floue type 1**
  - 3. Conclusion**
-

## 1. Introduction

Le terme d'ensemble flou fut paru la première fois en 1965 lorsque le professeur Lotfi A. Zadeh, de l'université de Berkeley aux USA,[4] publie un article intitulé « Ensembles flous (*Fuzzy sets*) ». Il a réalisé depuis de nombreuses avancées théoriques majeures dans le domaine et a été rapidement accompagné par de nombreux chercheurs développant des travaux théoriques. Parallèlement, certains chercheurs se sont penchés sur la résolution par logique floue des problèmes réputés difficiles. Ainsi en 1975,[5] le professeur Mamdani à Londres développe une stratégie pour le contrôle des procédés et présentées résultats très encourageants qu'il les a obtenus sur la conduite d'un moteur à vapeur. En 1978, la société danoise F. L. Smith réalise le contrôle d'un four à ciment.[6] C'est là, la première véritable application industrielle de la logique floue. La logique floue, dans la plupart des applications actuelles, permet de prendre en compte toutes sortes de connaissances quantitatives de concepteurs et d'opérateurs dans l'automatisation des systèmes. Dans le domaine des processus de production continue, et dans les automatismes les applications se sont également multipliées. La logique floue s'y développe car il s'agit d'une approche essentiellement pragmatique, efficace et générique. On dit parfois qu'elle permette de systématiser ce qui est du domaine de l'empirisme, et donc difficile à maîtriser. La théorie des ensembles flous fournit une méthode pertinente et facilement réalisable dans des applications temps réel [7] ; elle permet de transcrire et rendre dynamiques les connaissances des concepteurs ou des opérateurs.

## 2. Logique floue type-1

### Définition

Nous vivons dans un monde où très peu de notions sont binaires. Nous côtoyons l'imprécision au quotidien. Par exemple, quand nous disons « Pierre a la trentaine », la connaissance que nous apportons est imprécise.[8] Le besoin de modéliser ce type de connaissances s'est fait sentir et c'est ainsi que dans la seconde moitié du XXe siècle que le professeur Lofti Zadeh a introduit une logique permettant de formuler de tels énoncés. Cette logique est appelée logique floue et est basée mathématiquement sur la théorie des sous-ensembles flous. Le présent document que nous proposons vise à présenter l'idée de base de la logique floue. Ainsi le premier chapitre est destiné à montrer les types de raisonnements qu'il est possible de faire grâce à la logique floue. On montrera également pourquoi elle est indispensable pour formuler certaines connaissances. S'en suivra un historique. Le second chapitre sera consacré à la formalisation des notions données dans le chapitre 1. Ainsi nous

définirons les concepts de base de la théorie des sous-ensembles flous et expliquerons comment fonctionne le raisonnement flou.

### **Historique et domaines d'application :**

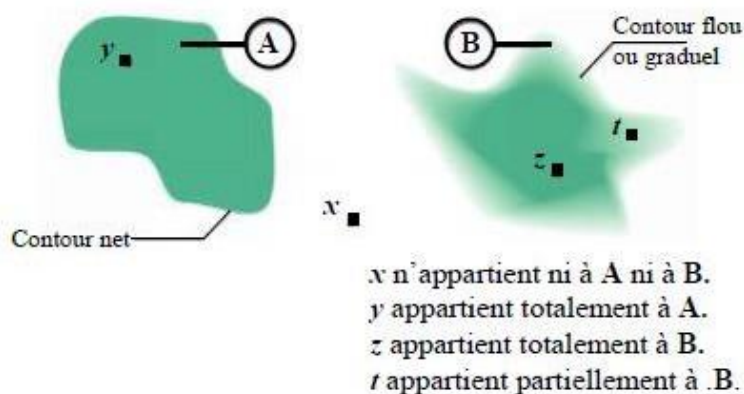
Les prémisses de la logique floue sont apparues en 1940 lorsque des chercheurs américains ont commencé à se pencher sur le concept d'incertitude. En 1965, Lofti Zadeh introduit la théorie des ensembles flous, (fuzzy set theory), il est alors professeur à l'université californienne de Berkeley et deviendra un automaticien de renommée internationale. 6 En 1973 le professeur Zadeh publie un article qui mentionne pour la première fois le terme de valeurs linguistiques, i.e. dont la valeur est un mot et non une lettre. En 1974 un chercheur 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 De nombreux produits grand public sont alors lancés sur le marché japonais avec la mention vendeuse : « fuzzylogicinside ». Comme par exemple dans la conception du lave-linge Aïsaïgo Day Fuzzy de Matsushita (Février 1990).[9] Il existe de nombreuses applications 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 contrôle/commande dans la plupart des domaines industriels de production, – transformation, traitement de produits et déchets. Terminons cette liste par une application plutôt impressionnante, à savoir la commande à distance d'un hélicoptère sans pilote, pour des opérations de sauvetage ou inspection de sites dangereux, travail développé par l'équipe du Professeur Sugeno. Ce système est caractérisé par 17 variables d'état et 4 commandes (vitesse et inclinaison des rotors), reçoit des ordres globaux comme : atterrissage, décollage, vol stabilisé, vol avant, vol arrière, virage à droite, virage à gauche,. . . Un système à base de logique floue, très structuré hiérarchiquement (2 niveaux de commande, une douzaine de blocs d'une soixantaine de règles chacun), traite ces ordres pour arriver aux actions sur les organes de commande de l'hélicoptère, qui vole déjà dans le ciel de Tokyo. Depuis longtemps

L'homme recherche à maîtriser les incertitudes et les imperfections inhérentes à sa nature. La première réelle manifestation de la volonté de formaliser la prise en compte des connaissances incertaines fut le développement de la théorie des probabilités à partir du XVII<sup>e</sup> siècle. Mais les probabilités ne peuvent maîtriser les incertitudes psychologiques et linguistiques. Puis la logique floue est apparue en 1965 à Berkeley dans le laboratoire de Lotfi Zadeh avec la théorie des sous-ensembles flous puis en 1978 avec la théorie des possibilités. Ces deux théories constituent aujourd'hui ce que l'on appelle Logique Floue. La logique floue permet la formalisation des imprécisions dues à une connaissance globale d'un système très complexe et l'expression du comportement d'un système par des mots. Elle permet donc la standardisation de la description d'un système et du traitement de données aussi bien numériques qu'exprimées symboliquement par des qualifications linguistiques.

### **Ensembles flous type-1**

Dans la théorie des ensembles, un élément appartient ou n'appartient pas à un ensemble. La notion d'ensemble est à l'origine de nombreuses théories mathématiques. Cette notion essentielle ne permet cependant pas de rendre compte de situations pourtant simples et rencontrées fréquemment. A titre d'exemple, si on considère l'univers des fruits, il est facile de définir l'ensemble des pommes. Par contre, il sera plus difficile de définir l'ensemble des pommes mûres. On conçoit bien que la pomme mûrit progressivement..., la notion de pomme mûre est donc graduelle.

C'est pour prendre en compte de telles situations qu'a été créé la notion d'ensemble flou. La théorie des ensembles flous repose sur la notion d'appartenance partielle : chaque élément appartient partiellement ou graduellement aux ensembles flous qui ont été définis. Les contours de chaque ensemble flou (cf., Figure 2.1) ne sont pas « nets », mais « flous » ou « graduels ». [10]



**Figure. 1 :** Comparaison d'un ensemble classique et d'un ensemble flou.

Un ensemble flou  $B$  est défini sur un ensemble de valeurs appelé l'univers du discours  $X$  par sa « fonction d'appartenance »  $\mu_B(x)$ , qui correspond à la notion de « fonction caractéristique » en logique classique. La fonction d'appartenance comprise généralement entre 0 et 1 qui quantifie le degré avec lequel chaque élément  $x$  de  $X$  appartient à  $B$ .

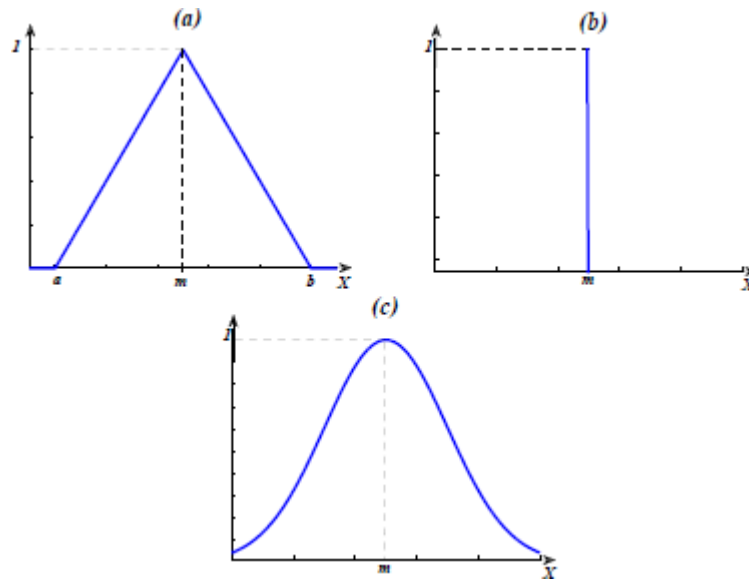
Un ensemble flou  $B$  dans l'univers du discours  $X$  peut s'écrire comme suit :

$$B = \{(x, \mu_B(x)) | x \in X\}, (1)$$

Où  $\mu_B(x)$  est le degré d'appartenance de  $x$  dans  $B$ .

Selon le type de la fonction d'appartenance, différents types d'ensembles flous seront obtenus. Les fonctions d'appartenance pourraient être classées en deux groupes : celles qui sont faites de lignes droites et celles dont la forme est Gaussienne. Les fonctions d'appartenance les plus couramment utilisées sont présentées dans la figure. 2 :

- **Fonction d'appartenance Triangulaire** : elle est définie par une valeur inférieure  $a$ , une valeur supérieure  $b$  et une valeur modale  $m$  de sorte que  $a < m < b$  (Figure 2(a)).
- **Fonction d'appartenance Singleton** : cette fonction d'appartenance prend la valeur 0 dans tout l'univers du discours sauf au point  $m$  où elle prend la valeur 1 (Figure. 2(b)).
- **Fonction d'appartenance Gaussienne** : elle est définie par sa moyenne  $m$  et son écart type  $\sigma$  (Figure. 2(c)).



**Figure. 2** : Fonction d'appartenance : (a) Triangulaire, (b) Singleton, (c) Gaussienne.

### Opérations de base sur les ensembles flous [11]

Soient  $A$  et  $B$  deux ensembles flous et  $\mu_A(x)$  et  $\mu_B(x)$  sont leurs fonctions d'appartenance respectivement sur l'univers du discours  $X$ . Comme dans la théorie des ensembles classique, on définit l'intersection, l'union des ensembles flous ainsi que le complément d'un ensemble flou. Ces relations sont traduites par les opérateurs « et », « ou » et « non ». De nouvelles fonctions d'appartenance liées à ces opérateurs sont établies :

- ✓  $x$  appartient à  $A$  et  $B \Leftrightarrow x \in A \cap B \Leftrightarrow \mu_{A \cap B}(x)$ .
- ✓  $x$  appartient à  $A$  ou  $B \Leftrightarrow x \in A \cup B \Leftrightarrow \mu_{A \cup B}(x)$ .
- ✓  $x$  appartient au complément de  $A \Leftrightarrow x \in \bar{A} \Leftrightarrow \mu_{\bar{A}}(x)$ .

L'opérateur « et » se définit par une norme triangulaire (*t-norme*)  $T$  :

$$T: [0 \ 1] \times [0 \ 1] \rightarrow [0 \ 1] \quad (2)$$

$$(x, y) \rightarrow z = xTy.$$

De même, l'opérateur « ou » se définit par une co-norme triangulaire ( $T^*$ ) qu'on appelle aussi *s-norme* ( $S$ ) :

$$T: [0 \ 1] \times [0 \ 1] \rightarrow [0 \ 1] \quad (3)$$

$$(x, y) \rightarrow z = xSy.$$

Les opérateurs les plus utilisés en logique floue sont :

- 1) L'opérateur « et » pour la *t-norme*, qui correspond à l'intersection de deux ensembles *A* et *B*. Il peut être réalisé par :
  - ✓ La fonction « Min » :  $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$ .
  - ✓ La fonction arithmétique « Somme » :  $\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x)$ .
- 2) L'opérateur « ou » pour la *s-norme*, qui correspond à l'union de deux ensembles *A* et *B*. Il peut être réalisé par :
  - ✓ La fonction « Max » :  $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$ .
  - ✓ La fonction « Somme » :  $\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x)$ .
- 3) L'opérateur « non » est réalisé par :  $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$ .

### Variables linguistiques

En logique floue les concepts des systèmes sont normalement représentés par des variables linguistiques. Une variable linguistique est une variable dont les valeurs sont des mots ou des phrases utilisées couramment dans une langue naturelle ou un langage artificiel. Une variable linguistique est définie par :

$$(X, U, T(X), \mu_x), \quad (4)$$

Où *X* désigne le nom de la variable, *U* est l'univers du discours associé à la variable *X* (appelé aussi référentiel),  $T(x) = \{T_1, T_2, \dots, T_n\}$  est l'ensemble des valeurs linguistiques de la variables *X* (appelé également termes linguistiques ou étiquettes linguistiques), et finalement  $\mu_x$  sont les fonctions d'appartenance associées à l'ensemble de termes linguistiques.

### Système d'inférence floue

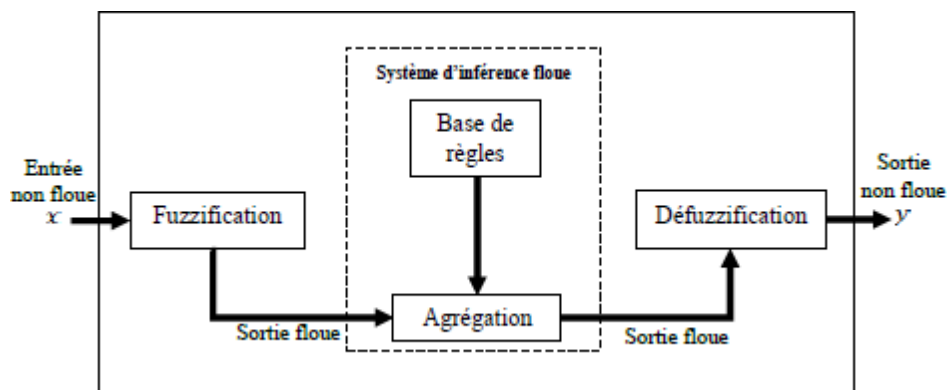
Un Système d'inférence floue (SIF) a comme but de transformer les données d'entrée en données de sortie à partir de l'évaluation d'un ensemble de règles. [12] Les entrées sont issues du processus de fuzzification et l'ensemble de règles normalement sont définies par le savoir-faire de l'expert. Un SIF (Figure. 3) est constitué de trois étapes :

- a) Fuzzification,
- b) Inférence,
- c) Défuzzification.

La première étape est la fuzzification, qui consiste à caractériser les variables linguistiques utilisées dans le système. Il s'agit donc d'une transformation des entrées réelles en une partie floue définie sur espace de représentation lié à l'entrée. Cet espace de représentation est normalement un sous-ensemble flou. Durant l'étape de la fuzzification, chaque variable d'entrée et de sortie est associée à des sous-ensembles flous.

La deuxième étape est le moteur d'inférence, qui est un mécanisme permettant décondenser l'information d'un système à travers un ensemble de règles définies pour la représentation d'un problème quelconque. Chaque règle délivre une conclusion partielle qui est ensuite agrégée aux autres règles pour fournir une conclusion (agrégation).

La troisième étape est la défuzzification, cette opération est l'inverse de la fuzzification et permet de transformer les sorties floues de l'inférence en une valeur non floue comme réponse finale du SIF.[13]



**Figure. 3 :** Système d'inférence floue (SIF).

### Exemple d'un système flou type-1

Le modèle flou « Takagi-Sugeno Kang (TSK) » est un appropiateur universel de fonctions réelles continues qui sont définies dans un sous-ensemble fermé et borné de nombre réel de  $n$ -dimensions ( $R^n$ ). Cette forte propriété du modèle TSK trouve plusieurs applications dans la modélisation des systèmes dynamiques. Un système flou de type TSK est décrit par un ensemble de règles *IF-THEN* qui représentent les relations entrée-sortie d'un système. Le système flou TSK du premier ordre est le plus couramment utilisé. Il a une base de  $M$  règles, chacune ayant  $p$  antécédents, ou la  $l^{ieme}$  règle est exprimée comme suite :

$$R^l: IF \underset{1}{x_1 \text{ is } F^l} \text{ and } \underset{2}{x_2 \text{ is } F^l} \text{ and, } \dots, \underset{p}{x_p \text{ is } F^l}, \quad (5)$$

$$THEN y^l = c_0^l + c_1^l x_1 + c_2^l x_2 + \dots + c_p^l x_p,$$

Où  $l = 1, 2, \dots, M$  ;  $c_j^l$  les paramètres de la conséquence, pour  $j = 0, 1, \dots, p$  ;  $x_j$  est l'entrée du système flou ;  $y^l$  est la sortie de la  $l^{ieme}$  règle ; et  $F_j^l$  est l'ensemble flou, pour  $j = 0, 1, \dots, p$ . La sortie finale du modèle TSK du premier ordre non complet est déduit comme suite :

$$r = \sum_{l=1}^M f^l y^l, \quad (6)$$

Où  $f^l$  sont les degrés d'activation des règles définies comme suit :

$$f^l = T_{j=1}^p \mu_{F_j}^l(x_j), \quad (7)$$

Où  $T$  dénote la  $t$ -norme.

Lorsque les fonctions d'appartenance sont Gaussiennes :

$$\mu_{F_j}^l(x_j) = \exp \left[ -\frac{1}{2} \left( \frac{x_j - m_j^l}{\sigma_j^l} \right)^2 \right]. \quad (8)$$

Et la  $t$ -norme est utilisée, l'Equation 6 peut être exprimée comme suit :

$$r = \sum_{l=1}^M y^l \prod_{j=1}^p \exp \left[ -\frac{1}{2} \left( \frac{x_j - m_j^l}{\sigma_j^l} \right)^2 \right], \quad (9)$$

Où  $m_j^l$  et  $\sigma_j^l$  sont les centre () moyennes et les écarts types (déviations standards) du  $l^{iem}$  ensemble flou, respectivement.

### 3. Conclusion

Ce chapitre a été dédié à la présentation d'une technique qui sera utilisée ultérieurement à savoir de la logique floue type-1 Le fondement théorique de cette méthode a été explicitement présenté dans le but d'ouvrir la voie à l'algorithme qui sera proposé dans le chapitre 4.

---

## CHAPITRE III

### *L'optimisation par essaim particulaire (PSO)*

---

*L'optimisation est un paradigme actuel dans chaque aspect de la vie. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation.*

#### **Sommaire**

---

1. Introduction
  2. Optimisation par essaim particulaire
  3. Les Composants de vitesse
  4. Les paramètres du PSO
  5. La comparaison entre les deux versions du PSO
  6. Test de l'efficacité du PSO
  7. Les champs d'application du PSO
  8. Conclusion
-

## 1. Introduction

L'optimisation est un paradigme actuel dans chaque aspect de la vie. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation [14]. Bien que les problèmes d'optimisation soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes complexe, stochastique, ou fortement non-linéaire et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données.

Dans ce chapitre, l'accent sera mis sur une méthode (algorithme) d'optimisation intelligente classée comme méta heuristique inspirée par analogie de l'éthologie. Il s'agit l'algorithme d'optimisation par essaim particulaire (OEP). Nous nous intéressons, par la suite, notamment à la présentation des différents types de l'algorithme d'optimisation par essaim particulaire, et l'efficacité du cet algorithme dans la résolution des problèmes non linéaires[15].

### 1. Optimisation par essaim particulaire

La bio-inspiration est un changement de paradigme qui amène les ingénieurs à s'inspirer de la nature pour développer de nouveaux systèmes artificiels. Depuis son apparition , L'optimisation par essaim particulaire a gagné la popularité croissante parmi des chercheurs et des praticiens comme une technique robuste et efficace pour résoudre des problèmes d'optimisation difficiles.

L'optimisation par essaim particulaire (OEP ) est classée dans les techniques d'optimisation stochastiques à population introduite en 1995 par *James Kennedy* [16](psychologue social et statisticien) et *Russel Eberhart* (ingénieur) sous le nom Particle Swarm Optimisation (PSO) . PSO est inspiré par le comportement social des animaux, comme les essaims d'abeilles, les vols d'oiseaux et les bancs de poissons. En effet, tout comme ces animaux se déplacent en groupe pour trouver de la nourriture, ou émigrent, Si un membre voit un chemin souhaitable pour l'exploration, le reste de cet essaim suivra rapidement. Raison pour laquelle, cette technique est fondée sur la notion de coopération et

l'échange d'information entre les particules. Les individus de l'algorithme sont appelés particules et la population est appelée essaim[17].

## 2. Les différents types du PSO

### PSO de l'optimum global

Dans cette version le voisinage pour chaque particule est l'essaim entier. Le réseau social utilisé par ce type d'algorithme est la topologie d'étoile, figure (3.1). Pour la topologie de voisinage d'étoile, le composant social de la mise à jour de vitesse de particules reflète l'information obtenue à partir de toutes les particules dans l'essaim. Dans ce cas-ci, l'information sociale est la meilleure position trouvée par l'essaim, désigné sous le nom de  $g$ [18].

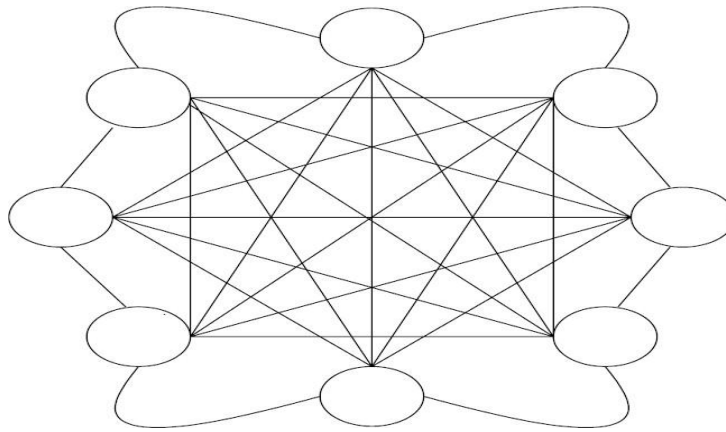


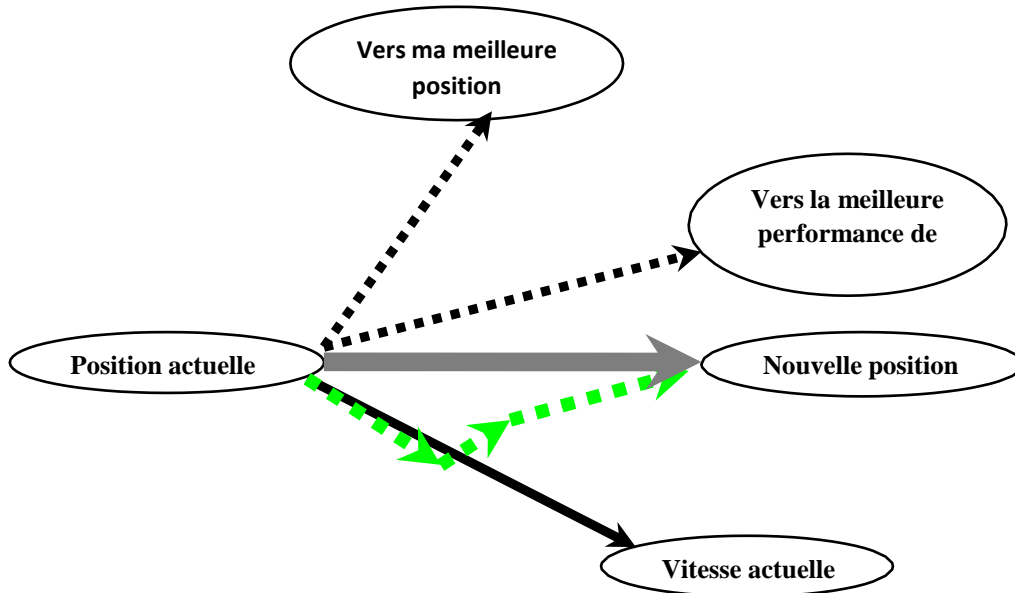
Figure.1 : L'étoile topologie de voisinage.

### Principe de déplacement d'une particule dans PSO de l'optimum global

L'algorithme PSO dans sa version globale est initialisé par une population de solutions potentielles aléatoires interprétées comme des particules se déplaçant dans l'espace de recherche. Toutes les particules étant considérées voisines et chacune d'elle dispose d'une vitesse et une petite mémoire, lui permettant de se souvenir de sa meilleure performance, en position et en valeur. La performance de chaque particule est mesurée selon une fonction objective prédéfinie relative au problème à résoudre[19].

La théorie de la socio-psychologie suggère que les individus sont influencés par leur comportement passé et par le succès de leurs voisins dans le réseau social et non pas nécessairement dans l'espace. Dans ce contexte, chaque particule interroge chaque congénère

de proximité sur sa meilleure performance. Sur la base des informations dont elle dispose, chaque particule modifie sa vitesse d'une manière stochastique en fonction de sa propre vitesse actuelle, sa meilleure performance et la meilleure performance de ses voisins[20].



**Figure.2** :Principe de déplacement d'une particule dans PSO de l'optimum global.

La figure (2) montre que la décision de déplacement d'une particule est basée sur une combinaison linéaire de trois tendances à savoir de suivre sa propre voie, revenir vers sa meilleure performance et aller vers la meilleure performance de ses voisins.

Chaque membre de l'essaim conserve une connaissance partielle de sa meilleure position  $p^i$ , et la meilleure position de toutes les particules de l'essaim  $g$  et ajuste dynamiquement sa propre vitesse en se basant sur ces informations. L'ajustement de vitesse est basé sur les comportements historiques des particules eux-mêmes aussi bien que leurs compagnons. La particule la mieux placée dans une région proche de l'endroit de la nourriture (optimum global) va motiver l'essaim d'explorer de proximité cette région et les régions adjacentes. De cette façon, les particules tendent à se rapprocher davantage de cet optimum d'une étape à l'autre durant le processus de recherche. Dans chaque étape on définit une nouvelle meilleure particule qui, à son tour, va attirer vers elle l'ensemble de l'essaim[21].

La particule ajuste dynamiquement sa propre vitesse en se basant sur les comportements historiques des particules eux-mêmes aussi bien que leurs compagnons[25].

La vitesse de chaque particule peut être mise à jour itérativement selon l'équation suivante :

$$v^i(k + 1) = wv^i(k) + c_1rand_1 (p^i - x^i(k)) + c_2rand_2 (g - x^i(k)) \quad (1)$$

Où

$k$  : est le nombre d'itérations,

$w$  : est le coefficient d'inertie,

$v^i(k)$  : est la vitesse courante de la particule  $i$  à l'itération  $k$ ,

$x^i(k)$  : est la position actuelle de la particule  $i$  à l'itération  $k$ ,

$c_1, c_2$  : sont les coefficients d'accélération,

$rand_1, rand_2$  : sont des nombres aléatoires de distribution uniforme sur [0,1].

Chaque particule rejoint sa nouvelle position dans l'espace de recherche en utilisant l'équation suivante :

$$x^i(k + 1) = x^i(k) + v^i(k + 1) \quad (2)$$

Après un certain nombre d'itérations, les particules se regroupent par la suite dans la région où les solutions sont les plus convenables (probables). Si une particule tend à sortir de l'espace de recherche, on lui attribue la valeur du point frontière le plus proche avec ralentissement voir annulation de sa vitesse.

L'algorithme PSO de l'optimum global a été décrite par plusieurs auteurs. L'organigramme de la figure (3) montre comment PSO de l'optimum global opère [22].

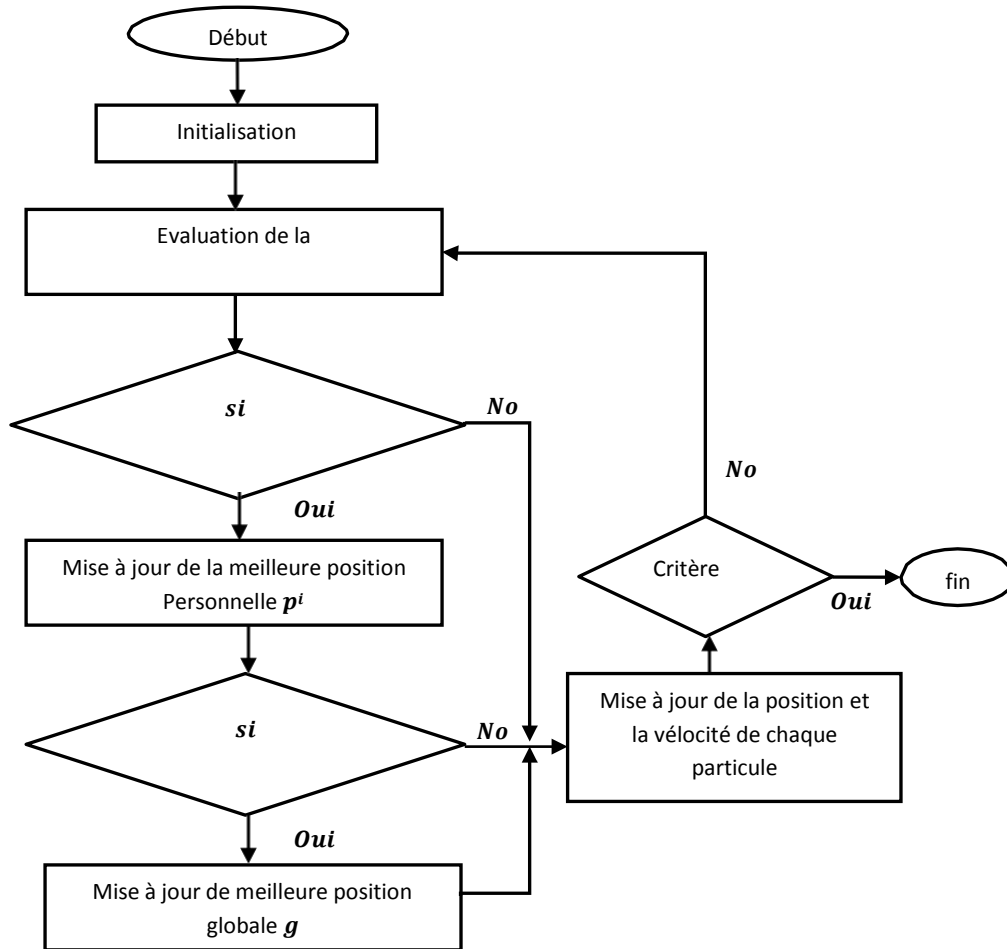


Figure.3 : Organigramme de l’algorithme PSO de l’optimum global.

### PSO de l’optimum local

Dans cette version de PSO, utilisations d’une topologie de réseau sociale d’anneau où des plus petits voisinages sont définis pour chaque particule, figure (4). Le composant social reflète l’information échangée dans le voisinage de la particule, reflétant la connaissance locale de l’environnement. La contribution sociale à la vitesse de particules est proportionnelle à la distance entre une particule et la meilleure position a trouvé par le voisinage des particules.

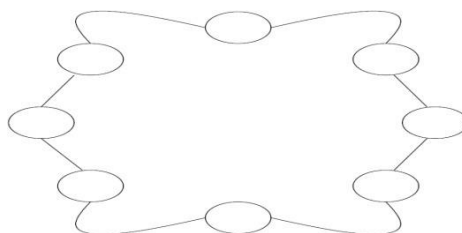
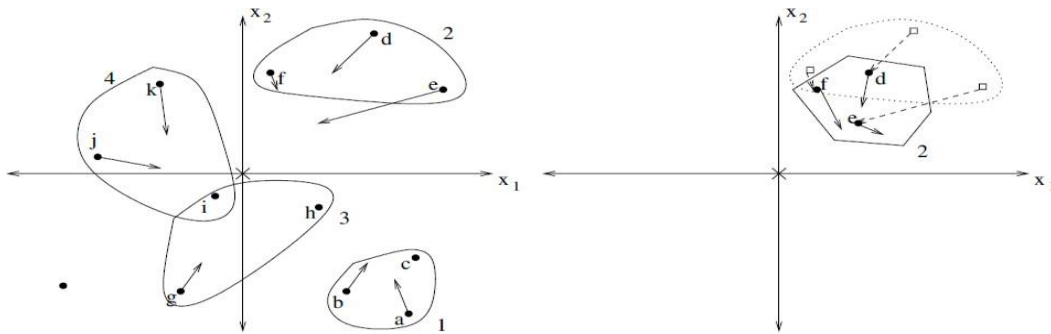


Figure .4 : L’anneau topologie de voisinage.

**Principe de déplacement d'une particule dans PSO de l'optimum local**

L'algorithme PSO dans sa version locale est initialisé par une population de solutions potentielles aléatoires interprétées comme des particules se déplaçant dans l'espace de recherche. comme illustré sur la figure (3.5), montre comment des particules sont influencées par leurs voisins immédiats. Pour maintenir le graphique lisible, seulement certains des mouvements sont illustrés, et seulement la direction globale de vitesse est indiquée. Le *voisinage 1*, les particules *a* et *b* se déplacent vers la particule *c*, qui est la meilleure solution dans ce voisinage. Considérant le *voisinage 2*, les mouvements des particules *d* vers *f*, fait ainsi *e*. Pour la prochaine itération, *e* sera la meilleure solution pour le *voisinage 2*. Mouvement maintenant de *d* et de *f* vers *e* comme illustré sur la figure (3.5) (b) (seulement une partie de l'espace de solution est illustrée). Les blocs représentent les positions précédentes. Noter que *e* reste la meilleure solution pour le *voisinage 2*. Noter également le mouvement général vers le minimum.



**Figure .5 :** Principe de déplacement d'une particule dans PSO de l'optimum local.

La particule ajuste dynamiquement sa propre vitesse en se basant sur les comportements historiques des particules eux-mêmes aussi bien que leurs voisins immédiats.

La vitesse de chaque particule peut être mise à jour itérativement selon l'équation suivante :

$$v^i(k + 1) = wv^i(k) + c_1rand_1(p^i - x^i(k)) + c_2rand_2(g_n - x^i(k)) \quad (3)$$

Où

$g_n$  : est la meilleure position découverte par les voisins immédiats .

la meilleure position découverte par les voisins immédiats  $N_i$  est défini comme suit :

$$g_n(k + 1) \in \{N_i / f(g_n(k + 1)) = \min\{f(x(k))\}, \forall x \in N_i\} \quad (4)$$

Il est important de noter que pour la PSO de base, les particules dans un voisinage  $N_i$  n'ont aucun rapport entre eux. Le choix des voisinages est fait à base sur des index de particules. Cependant, des stratégies ont été développées où des voisinages sont formés ont basé sur la similitude spatiale[23].

L'algorithme PSO de l'optimum local a été décrite par plusieurs auteurs. L'organigramme de la figure (6) montre comment PSO de l'optimum local opère.

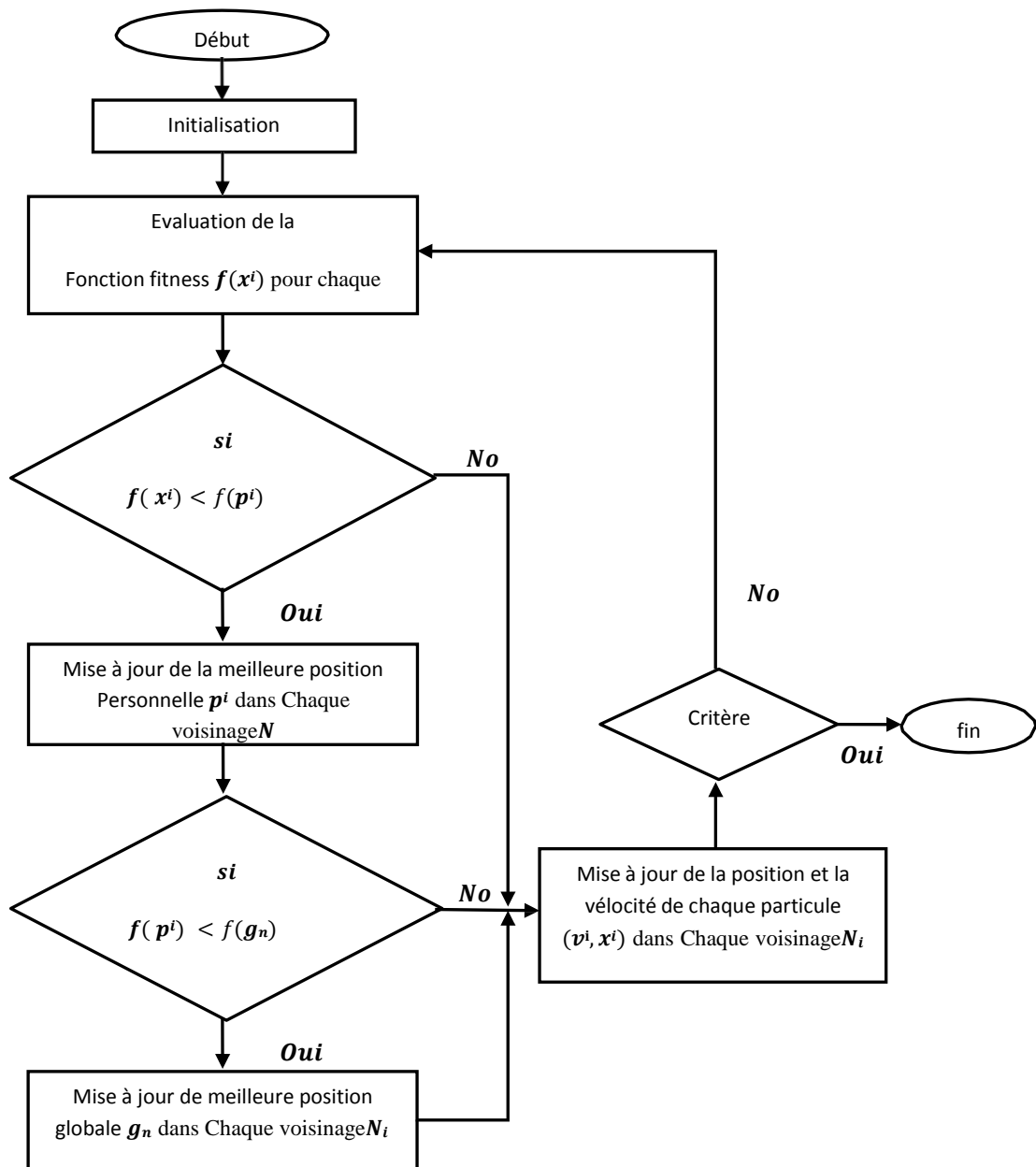


Figure.6 : Organigramme de l'algorithme PSO de l'optimum local.

### 3. Les Composants de vitesse

Le calcul de vitesse du  $i^{eme}$  particule comme donné dans les équations (3.1) et (3.2) compose de trois termes :

- **La vitesse précédente**,  $v^i(k)$ , qui sert de mémoire de la direction précédente de vol, c.-à-d. mouvement dans le passé immédiat. Cette terme de mémoire peut être vue comme élan, qui empêche la particule de la direction rigoureusement changeante, et à la polarisation vers la direction courante. Ce composant désigné également sous le nom du composant d'inertie[24].
- **Le composant cognitif**,  $c_1 rand_1 (p^i - x^i(k))$ , qui mesure la performance de la particule  $i$  relativement à sa performance passée. Dans une certaine mesure, le composant cognitif ressemble la mémoire individuelle du position qui était la meilleure pour la particule. L'effet de cette terme est que ces particules sont attirées à leurs propres meilleures positions, ressemblant la tendance des individus de retourner aux situations ou aux endroits qui les ont satisfaits plus dans le passé. *Kennedy et Eberhart* se sont également référés au composant cognitif comme « nostalgie » de la particule[25].
- **Le composant social**,  $c_2 rand_2 (g - x^i(k))$ , dans la version globale de PSO ou ,  $c_2 rand_2 (g_n - x^i(k))$ , dans la version local , qui mesure la performance de la particule  $i$  relativement à un groupe de particules, ou des voisins. Conceptuellement, le composant social ressemble à une norme ou à une norme de groupe que les individus cherchent à atteindre. L'effet du composant social est que chaque particule est attirée vers la meilleure position trouvée par son voisinage[26].

La figure (7) présente une illustration géométrique des mises à jour de vitesse et de position:

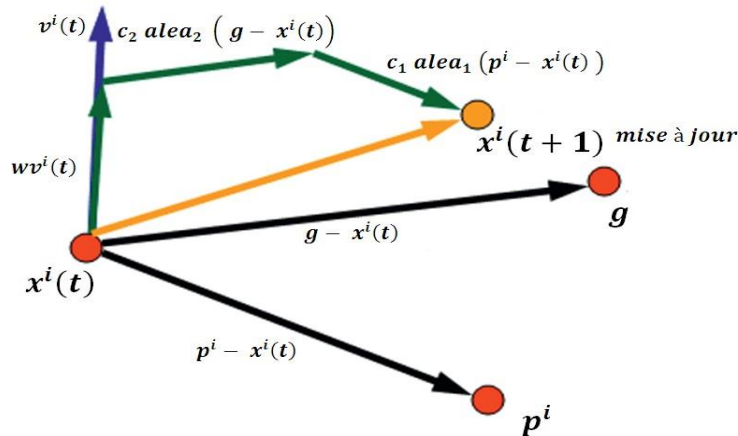


Figure .7 :Schéma vectorielle du déplacement d'une particule.

#### 4. Les paramètres du PSO

Le PSO de base est influencé par un certain nombre de paramètres de contrôle, à savoir la dimension du problème, nombre de particules, coefficients d'accélération, poids d'inertie, voisinage taille, nombre d'itérations, et les valeurs aléatoires qui influencent la contribution des composants cognitifs et sociaux[27].

- **Taille de l'essaim**: On lui a montré dans un certain nombre d'études empiriques que le PSO a la capacité de trouver les solutions optimales avec de petites tailles d'essaim de 10 à 30 particules.
- **Taille de voisinage** : La taille de voisinage définit l'ampleur de l'interaction sociale dans l'essaim. Plus les voisinages sont petits, moins l'interaction se produit. Tandis que de plus petits voisinages sont plus lents dans la convergence, ils ont une convergence plus fiable aux solutions optimales. De plus petites tailles de voisinage sont moins susceptibles des minimum locaux. Pour profiter des avantages de petites et grandes tailles de voisinage, commencer la recherche avec de petits voisinages et augmenter la taille de voisinage proportionnellement à l'augmentation en nombre des itérations. Cette approche assure une première diversité élevée avec une convergence plus rapide pendant que les particules se déplacent vers un secteur prometteur de recherche.
- **Nombre d'itérations** : Le nombre d'itérations pour atteindre une bonne solution est également dépendante à la problème. Trop peu d'itérations peuvent terminer la recherche prématurément. Un nombre trop grand d'itérations a la conséquence de la

complexité du calcul supplémentaire inutile (à condition que le nombre d'itérations est le seul état d'arrêt).

- **Les coefficients d'accélération:** influencent la valeur maximale du déplacement qu'une particule peut prendre durant une seule itération.  $c_1, c_2$  sont désignés également sous le nom de paramètres cognitif et social, respectivement. Les nombres aléatoires  $rand_1, rand_2$  sont employés pour affecter la nature stochastique de l'algorithme.
- **Le coefficient (poids) d'inertie:** a été développée par *Eberhart et al*, régit la quantité de vitesse actuelle qui devrait être maintenue pour influencer le calcul de la nouvelle vitesse. De cette façon, le paramètre  $w$  règle les capacités d'exploration globale et locale de l'essaim et influence le comportement de convergence de l'algorithme. Un grand poids d'inertie facilite l'exploration globale (exploration de nouveaux secteurs), alors qu'une petite valeur tend à faciliter l'exploration locale (exploration fine du secteur courant de recherche). Une valeur appropriée du poids de l'inertie fournit habituellement l'équilibre entre les capacités globales et locales d'exploration et par conséquent une réduction du nombre d'itérations nécessaires pour localiser la solution optimale. En règle générale, il est souhaitable de commencer le poids d'inertie à une grande valeur, afin de faire une meilleure exploration globale de l'espace de recherche, et le diminue graduellement pour affiner l'exploration locale.

La figure (3.8) illustre le diminution progressive en poids de l'inertie:

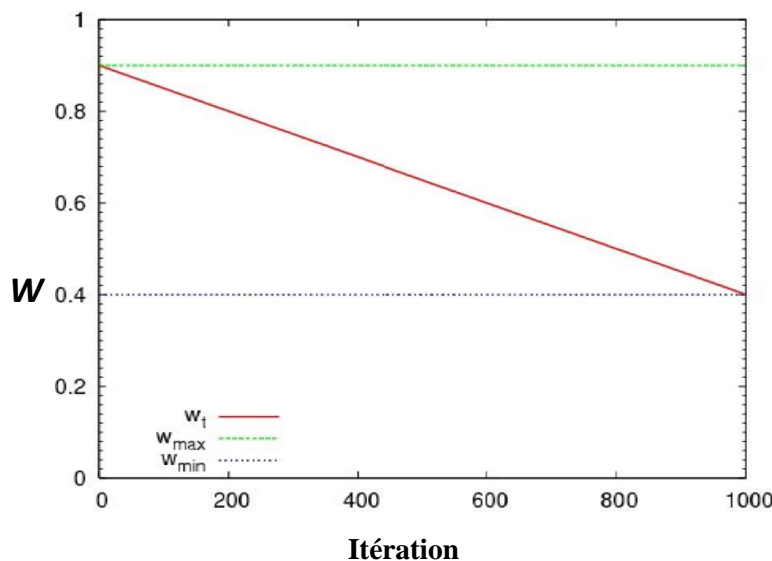


Figure. 8 : Le diminution progressive en poids de l'inertie.

## 5. La comparaison entre les deux versions du PSO

Il y a deux différences principales entre les deux approches en ce qui concerne leurs caractéristiques de convergence :

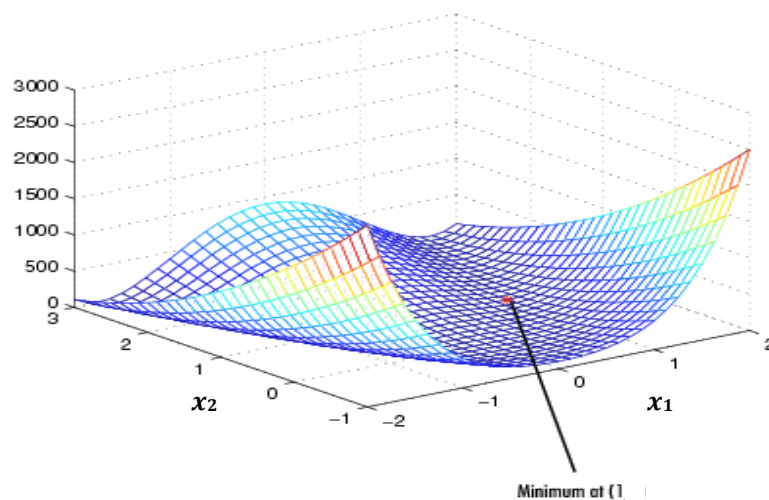
- En raison de la très grande interconnectivité des particules du PSO de l'optimum global, il converge plus rapidement que le PSO de l'optimum local. Cependant, cette convergence plus rapide vient au coût de moins de diversité que le PSO de l'optimum local.
- Comme conséquence de la très grande diversité du PSO de l'optimum local (qui a comme conséquence de plus grandes parties de l'espace de recherche étant couvert), le PSO de l'optimum local est moins susceptible de l'emprisonnement dans des minimums locaux. En général (selon le problème), les structures de voisinage telles que la topologie d'anneau utilisée dans le PSO de l'optimum local améliore les performances[28].

## 6. Test de l'efficacité du PSO

Dans le but de tester les capacités du PSO, nous avons choisi de chercher l'optimum global de la fonction Rosenbrock définie par :

$$f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \quad (5)$$

Cette fonction, comme le montre la figure (9), a plusieurs optimums locaux et n'admet qu'un seul optimum global défini au point (1,1) tel que  $f(1,1) = 0$ .



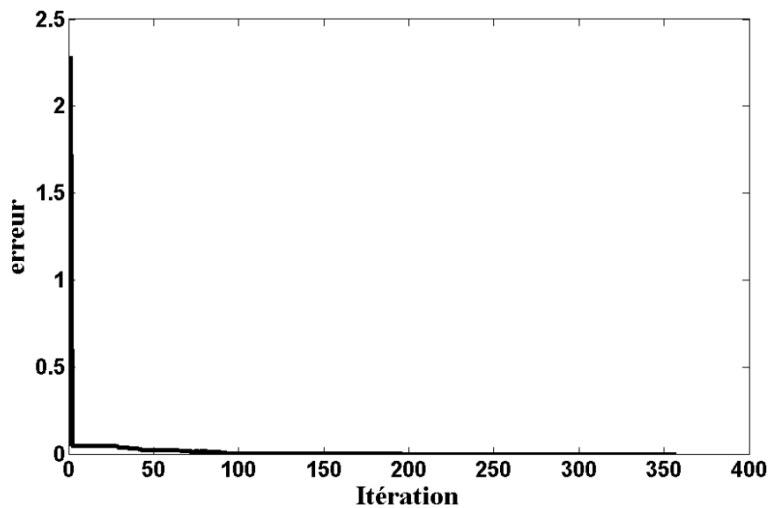
**Figure.9** : Fonction de Rosenbrock.

On prend :

- $S = 20$  ; taille de l'essaim.
- $k = 1000$  ; nombre d'itérations.
- $c_1 = c_2 = 1.8$
- $w$  ; d'après *Eberhart et al*
- Condition d'arrêt est ( $k = 1000$  Ou la valeur du fonction objective "erreur" =0)

Le résultat obtenu est :  $x_1 = 1$  ,  $x_2 = 1$  , *erreur*  $\cong 0$  après 357 itirations

La figure (3.10) montre la convergence de la fonction objective vers l'optimum global.



**Figure. 10** : convergence de la fonction objective vers l'optimum global.

## 7. Les champs d'application du PSO

Cet algorithme se révèle efficace pour les problèmes d'optimisation non linéaires à variables continue ou discrète[25]. Il a fait sa preuve dans des domaines variés entre autres comme l'optimisation de:

- l'apprentissage de réseaux de neurones,
- la commande par logique floue,
- les régulateurs PID,
- régulation et optimisation de l'écoulement de puissance,
- l'identification et le diagnostic des machines électriques

- la commande des convertisseurs de puissance.
- ...etc.

## **8. Conclusion**

Parce qu'elle est simple à comprendre, à programmer et à utiliser, PSO se révèle efficace pour les problèmes d'optimisation non linéaires. La puissance de cet algorithme découle de la nature même de son principe fondé sur la notion de coopération entre les particules. PSO semble très performant en termes de la précision des solutions trouvées, du temps de calcul et du nombre de paramètres à ajuster. Pour notre système, nous avons opté pour l'algorithme PSO de l'optimum global car il semble bien adapté au problème posé en termes de temps de calcul.

L'analyse et l'étude des performances de ces méthodes choisies appliquées au domaine de contrôle et de surveillance des eaux propres. Constituent notre principal objectif. Une étude en simulation ayant pour but d'évolution des performances de ces techniques proposées, fera l'objet du chapitre suivant. L'évaluation des résultats, reflétant les performances obtenues, nous conduira au meilleur choix de la méthode hybride la mieux adaptée à l'application. L'architecture du système de contrôle et de surveillance proposé est aussi présentée.

---

## CHAPITRE IV

### *Modélisation PSO-Floue Type-1 des Systèmes Dynamiques*

---

*Dans ce chapitre nous allons discuter une technique de modélisation des systèmes dynamiques à base de la logique type-1 et l'algorithme d'optimisation PSO. Cette technique intègre la capacité d'apprentissage de la logique floue avec l'algorithme d'optimisation PSO pour approximer les systèmes dynamiques.*

#### **Sommaire**

---

<b>1. Introduction.....</b>	<b>1</b>
<b>2. Méthode proposée pour la modélisation des systèmes dynamique ..</b>	<b>2</b>
<b>3. Résultats de simulation.....</b>	<b>3</b>
<b>4. Conclusion .....</b>	<b>4</b>

---

## 1. Introduction

Modéliser les systèmes complexes est un véritable défi. Pour y parvenir, beaucoup de chercheurs ont eu l'idée d'avoir recours à l'intelligence artificielle. Dans ce chapitre nous allons présenter une technique de modélisation des systèmes dynamiques à base de la logique floue type-1 et l'algorithme d'optimisation PSO. Cette technique comprend une interconnexion parallèle de deux sous modèles flou type-1. Le premier sous modèle flou est le modèle primaire, qui représente un modèle ordinaire à faible résolution. Pour surmonter le problème de la résolution et obtenir un modèle avec une résolution plus élevée, nous avons introduit un deuxième sous-modèle flou appelé le modèle d'erreur, qui représente la modélisation d'erreur entre la sortie du modèle primaire et la sortie réelle du système dynamique. Le modèle d'erreur représente l'incertitude dans le modèle principal, cette incertitude est minimisée par une simple soustraction de la sortie du modèle d'erreur de la sortie du modèle primaire, ce qui entraîne une interconnexion parallèle entre les deux modèles. Cette interconnexion nous emmène à un seul modèle final unique et complet possédant une résolution plus élevée.

## 2. Technique de modélisation proposée

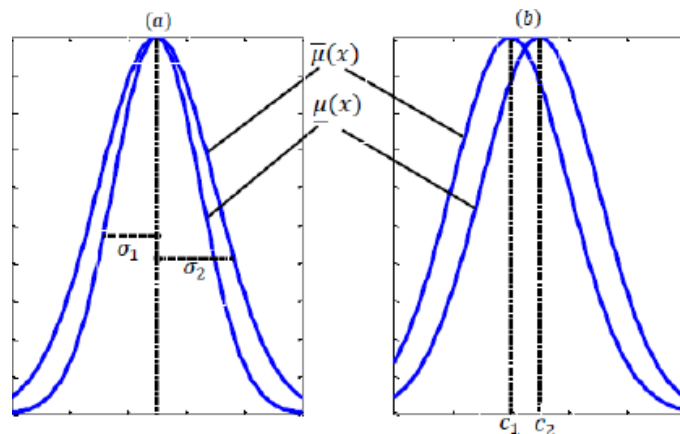
Notre objectif est d'offrir un modèle floue fiable et efficace pour représenter les systèmes dynamiques. Un modèle primaire est initialement conçu à l'aide des données réelles d'entrée/ sortie du système dynamique en considération. Ensuite, l'erreur entre la sortie du système dynamique en considération et la sortie du modèle primaire est modélisée pour créer le modèle d'erreur. Ce dernier modèle représente les incertitudes dans le modèle primaire qui peuvent être facilement supprimés par une simple soustraction de la sortie du modèle d'erreur de la sortie du modèle Primaire. Les détails de cette méthode seront présentés dans les sections suivantes.

Un modèle flou peut représenter ou modéliser n'importe quel système ou fonction inconnu,  $y = f(x)$ , en utilisant les données d'entrée/sortie. Le modèle flou utilisé est de type Takagi-Sugeno Kang (TSK) discutée dans la section.2.7 du chapitre 2, dont l'idée est de trouver une relation entre l'entrée et la sortie du modèle flou en ajustant au fur et à mesure ses paramètres en utilisant des mécanismes d'ajustement (algorithme d'optimisation) de telle façon fonction objectif atteint son minimum. Pour avoir l'objectif voulu notre méthode sera composée de trois étapes :

- *Étape 1 : identification du modèle primaire,*
- *Étape 2 : identification du processus d'erreur,*
- *Étape 3 : conception du modèle final.*

### Paramètres du modèle flou à ajuster

La conception du système flou type-2 pour la modélisation comprend la détermination des paramètres inconnus qui sont les paramètres de l'antécédent et les parties consécutives des règles floues. Dans les parties antécédentes, l'espace d'entrée est divisé en un ensemble de régions floues, puis dans les parties consécutives, le comportement du système dans ces régions conçu d'une manière automatique. Dans les ensembles flous types-2 Gaussiens, l'incertitude peut être associée à la moyenne (centre) et à la déviation standard (STD). Dans ce travail, les fonctions d'appartenance Gaussienne ont été choisies en raison de leur capacité d'approximation universelle et de leur aptitude à estimer uniformément des fonctions continues. Les figures.1 (a) et (b) représentent des ensembles flous type-2 Gaussiens avec des incertitudes dans la déviation standard (STD) et la moyenne (centre), respectivement.



**Figure 1:** ensemble flou type-2 Gaussien avec : (a) déviation standard incertaine, (b) moyenne incertaine (centre).

### Fonction objectif

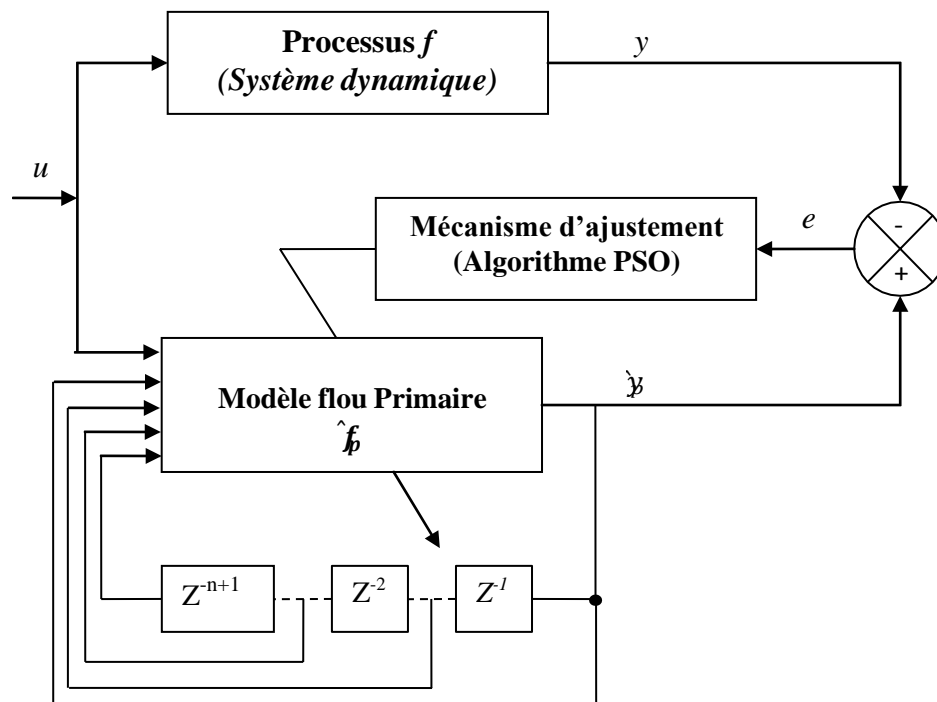
Le terme fonction objectif ou la fonction de coût est utilisé en optimisation mathématique pour désigner une fonction qui sert de critère pour déterminer la meilleure solution à un problème d'optimisation. La fonction objectif utilisée est le critère d'erreur quadratique moyenne (MSE : *Mean Square Error*) qui combine les valeurs réelles et estimées comme suit :

$$MSE = \frac{\sum_{k=1}^N (y_k - \hat{y}_k)^2}{N} = \frac{\sum_{k=1}^N e^2}{N} \quad (1)$$

Avec  $y_k$  et  $\hat{y}_k$  sont la mesure réelle et son estimation, respectivement, et  $N$  est la taille des données.

### Identification du modèle primaire

Pour cette étape, l'ensemble de données d'entrée/sortie est utilisé pour déterminer le modèle flou primaire  $\hat{f}_p$  pour le processus qui est le système dynamique en considération (voir Figure. 2).



**Figure 2:** Identification du modèle primaire.

Avec :

$f$ : fonction inconnue représentant la série temporelle que nous souhaitons identifier, ayant  $y$  comme sortie.

$u$  : entrée du système

$\hat{y}$  : la sortie du modèle primaire.

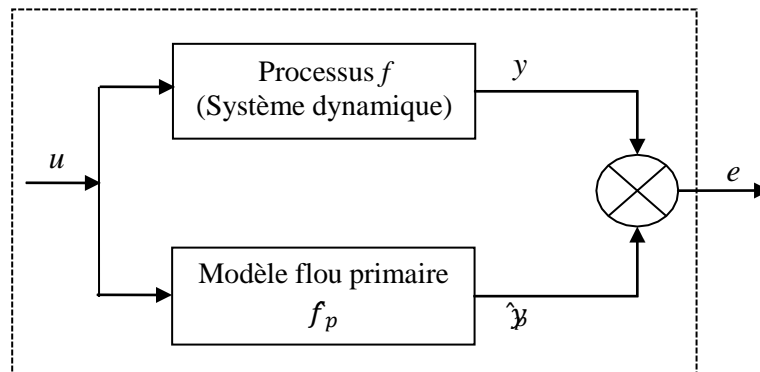
$e = \hat{y} - y$  : erreur de modélisation, utilisée dans la fonction du coût (1) de l'algorithme d'optimisation (pour exciter le mécanisme d'ajustement).

Le principe de l'identificateur flou consiste en une adaptation en temps réel de  $\hat{f}_p$ . Le bloc mécanisme d'ajustement représenté dans la Figure. 2, est l'algorithme d'optimisation PSO discuté dans le chapitre 3 qui va ajuster les paramètres du modèle flou  $f_p$  de telle sorte que l'erreur  $e$  entre la sortie du processus  $y$  et la sortie du modèle primaire  $\hat{y}$  atteigne sa valeur minimale.

Pour identifier  $\hat{f}_p$ , on a utilisé un modèle autorégressif et moyenne mobile (ARMA) flou, qui est une technique de modélisation qui fait partie d'un groupe de formules de prédiction qui tentent de prédire la sortie d'un système en se basant sur les résultats précédents et l'entrée du système (ancienne sortie devient nouvelle entrée).

### Identification du processus d'erreur

Le processus d'erreur ( $E_p$ ) est défini comme une interconnexion parallèle entre le processus (système dynamique considéré) et le modèle flou primaire comme le montre la figure suivante :



**Figure 3:** processeur d'erreur.

La sortie  $e$  du processus d'erreur ( $E_p$ ) est définie comme suit :

$$e = \hat{y} - y. \quad (2)$$

Une fois que la sortie du processus d'erreur ( $E_p$ ) est obtenue (Equation. 2), un deuxième modèle flou peut être conçu pour la modélisation de l'erreur  $e$ . Ce modèle sera étiqueté par le modèle flou d'erreur. L'erreur  $e$  est considérée comme une série temporelle, donc il convient de concevoir son modèle en utilisant un modèle flou

autorégressif (AR), qui tente de prédire la nouvelle sortie en fonction des résultats précédents.

La structure de l'étape 2 est illustré dans la Figure. 4, où  $\hat{e}$  est la sortie du modèle flou d'erreur et  $e_1$  est l'erreur de modélisation entre la sortie du processus d'erreur et la sortie du modèle flou d'erreur.

La tache maintenant est d'ajuster les paramètres du modèle flou d'erreur jusqu'à ce que l'erreur  $e_1$  atteint son minimum.

### Conception du modèle final

Dans cette dernière étape, le modèle primaire  $\hat{f}_p$  et le modèle d'erreur  $\hat{f}_e$  sont interconnectés en une structure parallèle (voir figure. 5) afin d'obtenir le modèle final  $f_F$ . Cette configuration nous permettra de compenser l'erreur résiduelle (erreur de modélisation) obtenu dans la première étape (modèle primaire), ce qui a comme

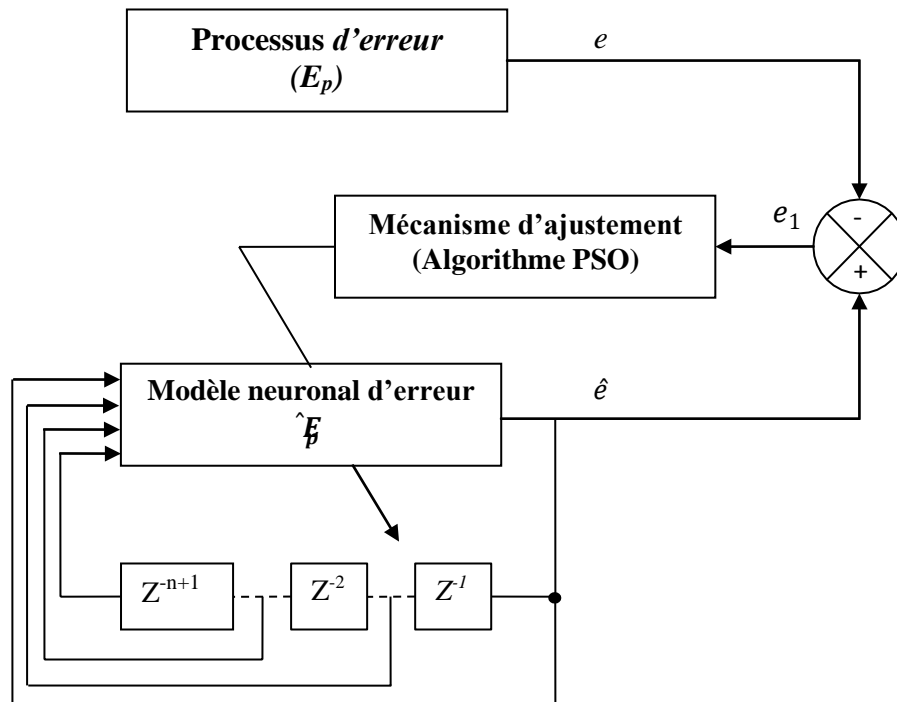


Figure 4: identification du processus d'erreur.

Conséquence la minimisation de l'erreur de modélisation globale. La sortie du modèle final peut être décrite comme suit :

$$\hat{y} = \hat{y} - \hat{e}. \quad (3)$$

Où  $\hat{y}$  et  $\hat{e}$  sont les sorties du modèle neuronal primaire et la sortie du modèle neuronal d'erreur, respectivement.

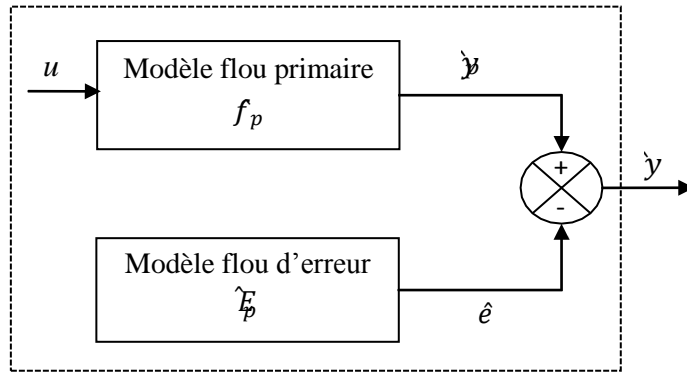


Figure 5: modèle final  $\hat{f}_p$

### 3. Résultats de simulation

Dans cette section nous allons présenter et discuter les résultats de simulation de notre méthode appliquée pour la modélisation des systèmes dynamiques. Trois systèmes dynamiques [1] décrits ci-dessous vont être considérer pour modélisation :

#### 1. Modèle I :

$$y_p(k + 1) = f[y_p(k), y_p(k - 1), \dots, y_p(k - n + 1)] + \sum_{i=0}^{m-1} \beta_i u(k - 1) \quad (4)$$

#### 2. Modèle II

$$y_p(k + 1) = \sum_{i=0}^{n-1} \alpha_i y_p(k - 1) + g[u(k), u(k - 1), \dots, u(k - m + 1)] \quad (5)$$

#### 3. Modèle III

$$y_p(k + 1) = f[y_p(k), y_p(k - 1), \dots, y_p(k - n + 1)] + g[u(k), u(k - 1), \dots, u(k - m + 1)] \quad (6)$$

#### Paramètres d'algorithme d'optimisation PSO

La taille de la population utilisée dans les algorithmes optimisations est un facteur important dans la détermination de la solution optimale. Lorsque la taille de la population augmente, la solution dans l'espace d'exploration est améliorée ; cependant cela augmentée temps de calcul. Dans ce travail, la taille de population correspondante à l'algorithme PSO a été fixé à 200. Les différents paramètres d'algorithme d'optimisation PSO sont choisis comme suit : les constantes d'accélération  $C_1 = C_2 = 2$  et le coefficient d'inertie  $\omega = 0.75$

#### Modélisation du système I

Dans cette sous-section nous allons examiner l'efficacité de la technique de modélisation floue sur un système compliqué (Équation 4), ou la non linéarité est associée seulement aux sorties, en considérant le cas particulier régi par l'équation aux différences suivante :

$$y_p(k + 1) = f[y_p(k), y_p(k - 1)] + u(k), \quad (7)$$

Avec :

$$f[y_p(k), y_p(k - 1)] = \frac{y_p(k)y_p(k-1)[y_p(k)+2.5]}{1+y_p^2(k)+y_p^2(k-1)}, \quad (8)$$

et:

$$u(k) = \sin \frac{2\pi k f}{25} \text{ for } 1 \leq k \leq 5 \text{ \& } 150 \leq k \leq 200. \quad (9)$$

$$u(k) = 0.5 \sin \frac{2\pi k}{10} + 0.5 \sin \frac{2\pi k f}{5} \text{ for } 50 \leq k \leq 150. \quad (10)$$

La technique de modélisation des systèmes dynamiques discutée précédemment sera appliquée pour modéliser le système dynamique donné par les Equations 7, 8, 9 et 10. Les résultats de simulation sont illustrés dans les Figures 6-a, 6-b et 6-c, respectivement, telle que :

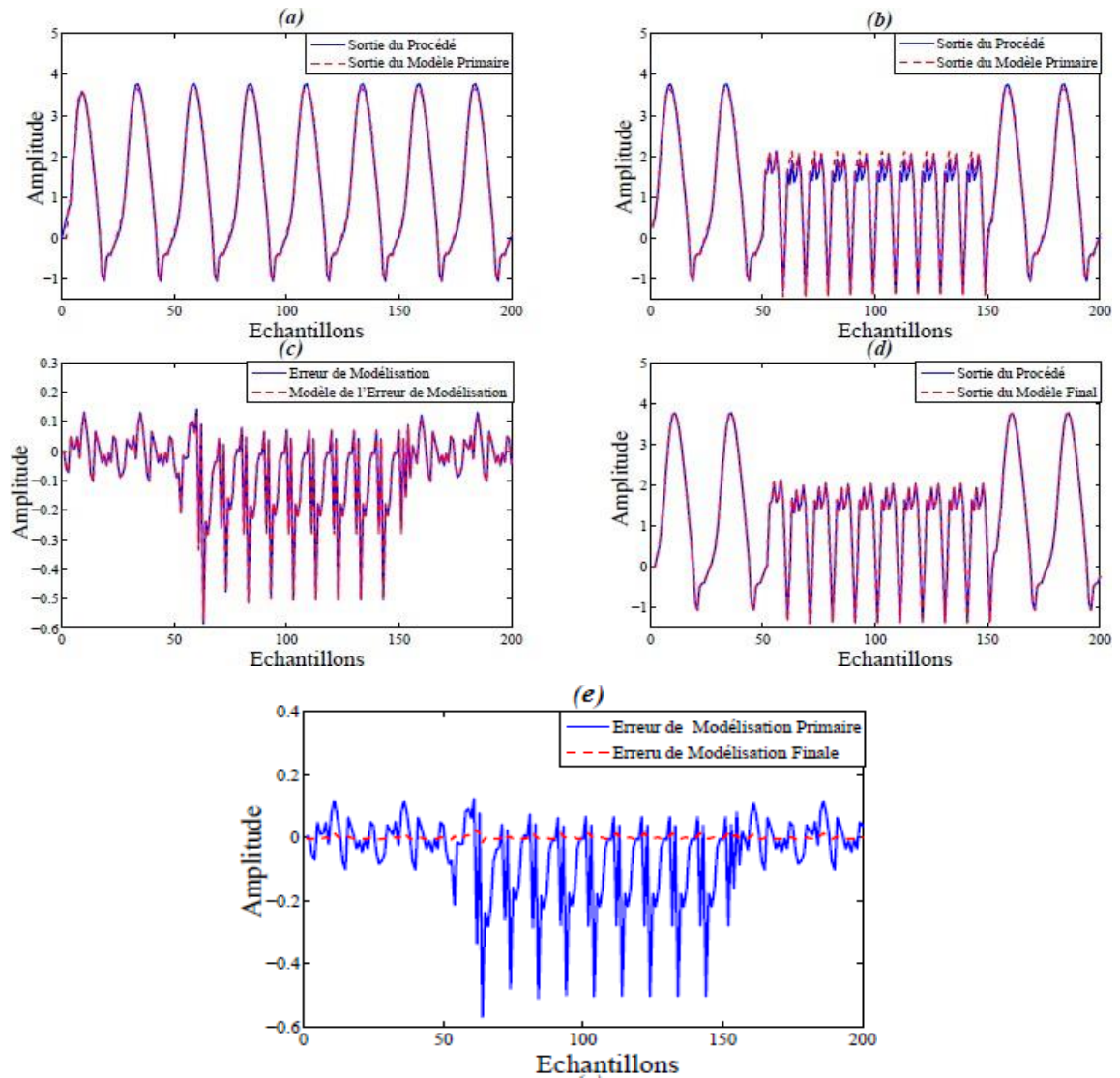
- ✓ **Figure. 6-a** : représente une superposition de la sortie du système dynamique et la sortie du modèle neuronal primaire.
- ✓ **Figure. 6-b** : représente une superposition de la sortie du procédé et la sortie du modèle primaire en utilisant une deuxième entrée (généralisation) :

$$u(k) = \sin \frac{2\pi k f}{25} \text{ for } 1 \leq k \leq 50 \text{ \& } 150 \leq k \leq 200$$

$$u(k) = 0.5 \sin \frac{2\pi k}{10} + 0.5 \sin \frac{2\pi k f}{5} \text{ for } 50 \leq k \leq 150$$

- ✓ **Figure. 6-c** : représente une superposition de l'erreur de modélisation et le modèle de l'erreur de modélisation.
- ✓ **Figure. 6-d** : représente une superposition de la sortie du procédé et la sortie du modèle final.
- ✓ **Figure. 6-e** : représente une superposition de l'erreur de modélisation primaire de l'erreur de modélisation finale.

Par une simple inspection visuelle des Figure. 6-a, 6-b, 6-c, 6-d et 6-e, on peut remarquer que le modèle final est beaucoup mieux que le modèle primaire, où on constat que l'erreur de modélisation finale a été réduite ce qui permet d'améliorer efficacement le modèle final.



**Figure 6: résultats de modélisation du système dynamique non linéaire (équation 8) basé sur PSO (a) modèle primaire ,(b) validation ,(c) modèle de l'erreur de modélisation ,(d) modèle final ,(e) erreur de modélisation finale**

### Modélisation du système II

Dans ce qui suit nous examinons l'efficacité de notre méthode sur un deuxième système plus compliqué (Équation. 5), où la non linéarité est associée seulement à l'entrée en prenant en considération l'équation aux différences suivante :

$$y_p(k + 1) = 0.3y_p(k) + 0.6y_p(k - 1) + f[u(k)], \quad (11)$$

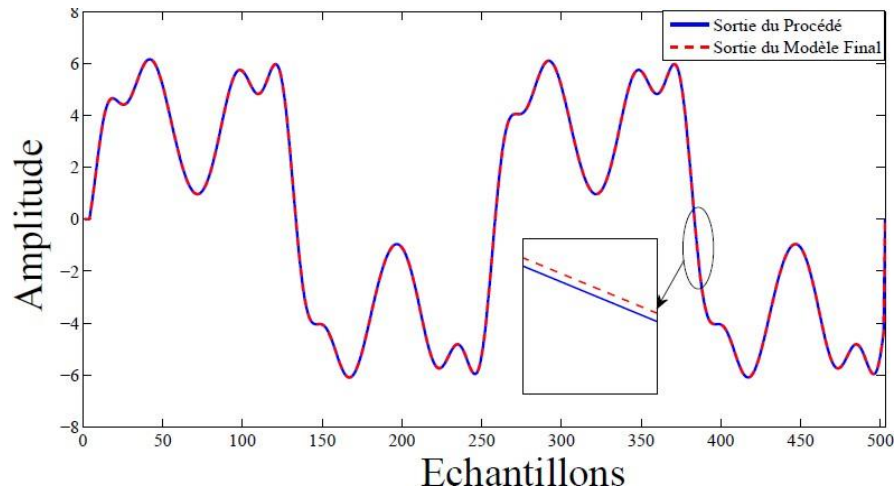
Avec :

$$f(u) = 0.6 \sin(\pi u) + 0.3 \sin(3\pi u) + 0.1 \sin(5\pi u). \quad (12)$$

et :

$$u(k) = \sin \frac{2\pi k}{250} \quad (13)$$

La Figure. 7 , représente le résultat final de l'identificateur flou. En inspectant la zoom, nous confirmons l'efficacité de notre méthode pour ce type de problèmes.



**Figure 7:** résultats de modélisation du système dynamique non linéaire (équation.11).

### Modélisation du système III

Le modèle III (Équation. 6), pour lequel l'entrée et la sortie du système sont tous les deux compris dans la non linéarité, va être utilisé pour confirmer l'efficacité de notre méthode pour ce type de problèmes, en considérant le cas particulier décrit par l'équation aux différences suivante :

$$y_p(k+1) = f_p(y_p(k), u(k)) = \frac{y_p(k)}{1+y_p(k)^2} + u^3(k), \quad (14)$$

où la fonction inconnue  $f_a$  à identifier a une forme plus générale du fait que ses variables indépendantes sont les signaux d'entrée et de sortie.

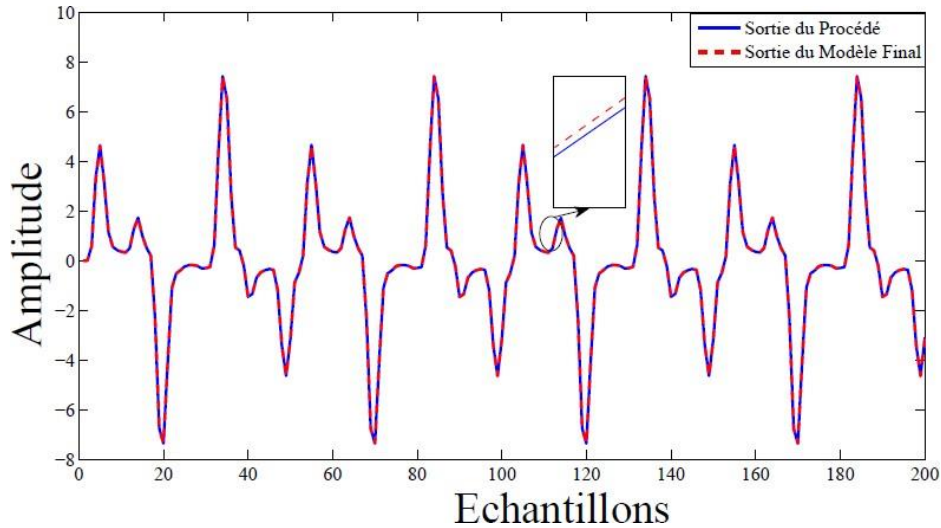
Le signal d'entrée  $u$  est choisi comme suit :

$$u(k) = \sin\left(\frac{2\pi k}{25}\right) + \sin\left(\frac{2\pi k}{10}\right). \quad (15)$$

La Figure. 8 , représente le résultat final de l'identificateur flou. En inspectant les zooms, nous confirmons toujours l'efficacité de notre méthode pour ce type de problèmes.

De tous ces résultats de simulation on peut remarquer l'efficacité de la structure proposée pour la modélisation du système dynamique. La particularité ici est l'introduction d'un nouveau module s'appelle le modèle d'erreur qui permet de mieux laisser le modèle final du système dynamique considéré. La deuxième particularité est l'utilisation de l'algorithme d'optimisation PSO pour ajuster les paramètres de la

structure proposée à base de la logique floue type1, ce qui élimine le problème du minimum local rencontré chaque fois en utilisant des méthodes d'optimisation du gradient.



**La Figure. 8 :** Résultats de modélisation du système dynamique non linéaire (Equation. 14)

#### 4. Conclusion

Dans ce chapitre, nous avons présenté une méthode de modélisation des systèmes dynamiques à base de la logique floue type-1 et l'algorithme d'optimisation PSO. L'approche permet de généraliser la notion d'identification en ajoutant un nouveau module d'identification, appelé modèle d'erreur. Le modèle flou d'erreur introduit a été utilisé comme un complément du modèle primaire pour améliorer la qualité du modèle, ce qui a fourni une bonne précision. L'optimisation des paramètres de la structure proposée a été assurée en utilisant l'algorithme d'optimisation PSO. Plusieurs résultats de simulations ont été introduits pour vérifier l'efficacité de notre méthode.

---

## *Conclusion Général*

---

## Conclusion général:

Le travail de recherche présenté dans ce mémoire de fin d'étude, avait pour but principal de :

- *développer une structure floue pour la modélisation et l'identification des systèmes dynamiques.*

Le chapitre 1 du présent mémoire a été dédié à présenter les notions de base relatives aux systèmes dynamiques. Dans les chapitres 2 et 3 nous avons présenté les techniques et les outils de base qui ont été utilisés pour avoir l'objectif voulu du présent mémoire à savoir la logique floue type 1 et l'algorithme d'optimisation PSO

Enfin, dans le chapitre 4, la technique de modélisation développée a été présentée en détail. Cette technique comprend une interconnexion parallèle de deux sous-modèles flous type-1. Le premier sous-modèle flou est le modèle primaire, qui représente un modèle ordinaire à faible résolution du système dynamique en considération. Pour surmonter le problème de la résolution et obtenir un modèle avec une résolution plus élevée, nous avons introduit un deuxième sous-modèle flou appelé le modèle d'erreur, qui représente la modélisation d'erreur entre la sortie du modèle primaire et la sortie réelle du système dynamique. Le modèle d'erreur représente l'incertitude dans le modèle principal, cette incertitude est minimisée par une simple soustraction de la sortie du modèle d'erreur de la sortie du modèle primaire, ce qui entraîne une interconnexion parallèle entre les deux modèles. Cette interconnexion nous mène à un seul modèle final unique et complet possédant une résolution plus élevée.

L'approche développée dans le cadre de ce mémoire de fin d'étude permet de généraliser la notion d'identification en ajoutant un nouveau module d'identification, appelé modèle d'erreur. Le modèle flou d'erreur introduit a été utilisé comme un complément du modèle

primaire pour améliorer la qualité du modèle, ce qui a fourni une bonne précision. L'optimisation des paramètres de la structure proposée a été assurée en utilisant l'algorithme d'optimisation PSO. Plusieurs résultats de simulations ont été introduits pour vérifier l'efficacité de la structure développée.

Comme perspectives, nous proposons d'appliquer d'autre technique d'optimisation pour améliorer la qualité du résultat obtenu tel que :

- L'algorithme d'optimisation baptisée IWO (*Invasive Weed Optimization*) ;
- L'algorithme d'optimisation baptisée CMA-ES (*Covariance Matrix Adaptation Evolution Strategy*)

---

# *Références*

---

**Références:**

- [1] Habchi, F. (2013). Identification et commande des systèmes non linéaires par les techniques neuronales (Doctoral dissertation, Université abderrahmane mira béjaia).
- [2] Merzouka, N. (2018). Etude des performances des réseaux de neurones dynamiques à représenter des systèmes réel: une approche dans l'espace d'état (Doctoral dissertation).
- [3] Hammami, S. (2009). Sur la stabilisation de systèmes dynamiques continus non linéaires exploitant les matrices de formes en flèche: application à la synchronisation de systèmes chaotiques (Doctoral dissertation, Ecole centrale de Lille).
- [4] Zadeh, L. A. 1994. Soft computing and fuzzy logic. *IEEE Software*, 11(6), pp 48-56.
- [5] Mamdani, E.H., Assilian, S .1975 . An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man Mach. Studies*, 7(1) : 1–13,
- [6] Ostergaad, J.J. Fuzzy logic control of a heat exchange process. Dans *Fuzzy Automata and Decision Processes*, M.M. Gupta, G.N. Saridis, and B.R. Gaines, Eds., pp 285–320.
- [7] Willaeyts, D., Malvache, N. 1978. Use of fuzzy model for process control. *IEEE International Conference on Cybernetics and Society*.
- [8] IEC 61131-7 Programmable Controllers part 7 Fuzzy Control Programming.
- [9] James C. BEZDEK & Sanker K. PAL. 1992. *Fuzzy models for pattern recognition*, , IEEE Press.
- [10] D. DUBOIS, H. PRADE, Academic Press 1980. *Fuzzy sets and systems : Theory and applications*, , Mathematics in Sciences and Engineering vol. 144.
- [11] Bernadette Bouchon-Meunier 2003.- *Logique floue, principes, aide à la décision*, éditions Lavoisier.
- [12] Joao M. C. Sousa, Uzay Kaymar, 2002. *Fuzzy Decision Making in Modeling and Control*, World Scientific.
- [13] George J. Klir, 1995. Bo Yuan, *Fuzzy Sets and Fuzzy Logic, Theory and Applications*, Prentice Hall.
- [14] M Clerc - Hermès-Lavoisier, février, 2005 - particleswarm.info L'optimisation par essaim
- [15] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University, Press, 1996. tculaire Tutoriel pour OEP 2003 Maurice Clerc France Télécom R&D
- [16] Riccardo Poli, James Kennedy, and Tim Blackwell. 2007. Particle swarm optimization. *Swarm Intelligence*, 1(1) :33–57.

- [17] M.T Huynh 2009 «Application de l'Optimisation par Essaim Particulaire à la représentation de matrice de préférence valuées» Université Libre De Bruxelles.
- [18] D.M.Vinod Kumar, «Intelligent Controllers for Automatic Generation Control», Proc. of IEEE region 10 International conference on global connectivity in Energy, Computer, Communication and Control, 1998, p557-574.
- [19] A. S. Fraser. 1957. "Simulation of genetic systems by automatic digital computers". Australian Journal of Biological Science, Vol. 10, pp. 484–491,
- [20] Y. Cooren, "Perfectionnement d'un algorithme adaptatif d'optimisation par essaim
- [21] R. Eberhart, Y. Shi, 2000. "Comparing inertial weights and constriction factor in particle swarm optimization," Proceeding of the International Congress on Evaluationing Computation, San Diego, California NJ, pp. 84 88,
- [22] X. Hu et R.C. Eberhart, 2002. Adaptive particle swarm optimization: detection and response to dynamic systems, Congress on Evolutionary Computation, Hawaii,.
- [23] CHALANE Idir. OUARI Tarek 2017, Optimisation des paramètres d'un PID par essaim particulaires (PSO). Université A. MIRA de Bejaia
- [24] M.T Huynh 2009 .«Application de l'Optimisation par Essaim Particulaire à la représentation de,matrice de préférence valuées» Université Libre De Bruxelles
- [25]: KHEDAM .R, HAMDOUN .H, MILOUDI .S et BELHADJ-AISSA .A, 2007 ,, "L'algorithme d'optimisation par Essaims Particulaires Appliqué a la Classification d'Images Satellitaires"" . Première Journée Nationale sur les Applications des Métaheuristiques 2007 (JNAM 07). Alger .Algérie
- [26] Zwe-Lee Gaing , 2004. "A Particle swarm optimization approach for optimum design in AVR system" , IEEE Transactions on Energy Conversion , Vo.19, No.2,
- [27] Ziegler J .G., Nichols N .B., novembre 1942.Optimum setting for automatic controllers, ASME, vol. 64
- [28] C. Zhang, J. Ning, S. Lu, D. Ouyang, and T. Ding. "A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization". Oper. Res. Lett., Vol. 37, No. 2, pp. 117–122, March 2009

## Résumé

L'objectif de ce mémoire de fin d'étude est développer une structure floue pour la modélisation et l'identification des systèmes dynamiques. Plusieurs résultats de simulations ont été introduits pour vérifier l'efficacité de la structure développée.

Cette interconnexion nous emmène à un seul modèle final unique et complet possédant une résolution plus élevée.

## Abstract

The objective of this final thesis is to develop a fuzzy structure for the modeling and identification of dynamic systems. Several simulation results were introduced to verify the efficiency of the developed structure.

This interconnection takes us to a single, complete and unique final model with higher resolution.

## ملخص

الهدف من هذه الأطروحة النهائية هو تطوير هيكل غامض لنمذجة وتحديد الأنظمة الديناميكية. تم تقديم العديد من نتائج المحاكاة للتحقق من كفاءة الهيكل المطور.

يأخذنا هذا الترابط إلى نموذج نهائي واحد وكامل وفريد من نوعه بدقة أعلى.