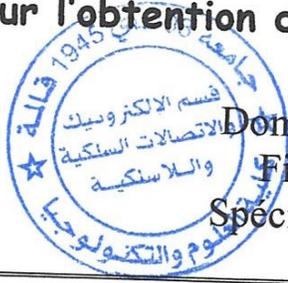


874

République Algérienne Démocratique et Populaire  
Ministère de l'enseignement supérieur et de la recherche scientifique  
Université 8Mai 1945 – Guelma  
Faculté des sciences et de la Technologie  
Département d'Electronique et Télécommunications



Mémoire de fin d'étude  
pour l'obtention du diplôme de Master Académique



Domaine : Sciences et Technologie  
Filière : Electronique  
Spécialité : Systèmes Electroniques

---

# MODELISATION FLOUE BASEE SUR LES METHODES DE CLUSTERING

---

Présenté par :

**Khezzane Abderrahmane**  
**Bouchmella Mohamed**

Sous la direction de :

Dr. Nemissi Mohamed

Juin 2015

*Dédicace*



*Je dédie ce mémoire*

*A mes parents pour leur amour inestimable, leur confiance, leur soutien, leurs sacrifices et toutes les valeurs qu'ils ont su m'inculquer.*

*A mes frères ainsi qu'à mes sœurs pour leur tendresse, leur complicité et leur présence malgré la distance qui nous sépare.*

*A mon superviseur Mr. Nemissi Mohamed*

*A toute ma famille ainsi qu'à mes amis.*

*Et à moi.*

*X. Abderrahmane*

## REMERCIEMENT

J'adresse mes remerciements aux personnes qui m'ont aidé dans la réalisation de ce mémoire.

En premier lieu, je remercie Mr. Nemissi Mohamed, Docteur à l'université de Guelma, Pour ses précieux conseils et ses encouragements lors de la réalisation de ce mémoire.

Je tiens ensuite à remercier mes parents pour le soutien inconditionnel dont ils ont fait preuve depuis que mon projet professionnel est défini. Merci pour le soutien financier, moral et psychologique. Si je suis ici aujourd'hui, c'est grâce à vous!

Je remercie également toutes les personnes qui, de près ou de loin, ont participé à l'élaboration de ce mémoire.

Enfin, je remercie mes amis et camarades de promotion pour ces années passées ensemble, dans les meilleurs moments comme dans les pires.

 **Abderrahmane**

## Résumé :

Dans ce travail on s'intéresse à la modélisation floue en se basant sur les données entré-sortie. Notre étude concerne les deux systèmes d'inférence floue les plus utilisés : le système de Mamdani et le système de Sugeno. Plus précisément, nous allons utiliser les méthodes de clustering pour la génération automatique des règles floues. Il s'agit, en premier lieu, de partitionner l'espace des entrées en régions formée par le clustering, puis, chaque région sera représentée par une règle floue.

### Mots clés :

Règles floues, modélisation floue, FCM, subtractive clustering, système de Mamdani, systèmes de Sugeno.

### Abstract:

This work deals with Input-output based-fuzzy modelling. Our study concerns the two most commonly used fuzzy inference systems: Mamdani and Sugeno. We use the clustering methods for automatic generation of fuzzy rules. First, the input space is partitioned into regions formed by the clusters, next each region is represented by a fuzzy rule.

### Keywords:

Fuzzy rules, fuzzy modeling, FCM, subtractive clustering, TSK fuzzy logic, Mamdani fuzzy logic.

# Table des matières

<b>RESUME :</b>	<b>V</b>
<b>TABLE DES MATIERES</b>	<b>VI</b>
<b>TABLEAU DE FIGURE</b>	<b>VIII</b>
<b>INTRODUCTION GENERALE</b>	<b>1</b>
<b>CHAPTER I : CONCEPT DE BASE DE LA LOGIQUE FLOUE</b>	<b>3</b>
I.1. INTRODUCTION	4
I.2. HISTORIQUE	4
I.3. DOMAINES D'APPLICATION	5
I.4. DEFINITION DE LA LOGIQUE FLOUE	6
I.5. EXEMPLE ILLUSTRATIF DE LA LOGIQUE FLOUE	7
I.5.1. LES SOUS-ENSEMBLES FLOUE	9
I.5.2. DEFINITION D'UN SOUS-ENSEMBLE FLOUE	9
a. Caractéristiques d'un sous-ensemble flou	10
i. Support	10
ii. La Hauteur	11
iii. Noyau	11
b. Les différentes formes de fonction d'appartenance	12
I.5.3. LES OPERATEURS FLOUES	14
a. L'opérateur Non	14
b. L'opérateur OU	14
c. L'opérateur Et	16
I.6. PROPRIETES DE FL	17
<b>CHAPTER II : SYSTEME D'INFERENCE FLOUES</b>	<b>20</b>
II.1. LES SYSTEMES D'INFERENCE FLOUS	21
II.1.1. FUZZIFICATION	22
II.1.2. INFERENCE FLOUE	22
a. Les méthodes des inférences	23
i. La méthode d'inférence max-min :	23
ii. La méthode d'inférence max-prod :	23
iii. La méthode d'inférence somme-prod :	23

b. Les règles Si-Alors floues	23
II.1.3. DEFUZZIFICATION	25
a. La méthode du maximum	25
b. La défuzzification par centre de gravité	26
II.2. SYSTEME D'INFERENCE FLOU DE MAMDANI	27
II.3. SYSTEME D'INFERENCE FLOU DE SUGENO ET TAKAGI	28
II.4. SYSTEMES FLOUS SOUS MATLAB	30
II.4.1. EXEMPLE DE MODELISATION AVEC LE SYSTEME DE MAMDANI	31
II.4.2. EXEMPLE DE MODELISATION AVEC LE SYSTEME DE SUGENO	35
<b>CHAPTER III : CLUSTERING</b>	<b>38</b>
III 1 INTRODUCTION	39
III.2. CLUSTERING PAR LA METHODE K -MEANS	39
III.2.1. L'ALGORITHME DE K-MEANS	40
III.3. CLASSIFICATION FLOUE FCM (FUZZY C- MEANS)	41
III.3.1. L'ALGORITHME DU FCM	43
a. Exemple :	43
III.4. LA METHODE DE SUBTRACTIVE CLUSTERING :	45
a. Exemple :	47
<b>CHAPTER IV : APPLICATION</b>	<b>49</b>
IV.1. MODELISATION FLOUE BASEE SUR LE CLUSTERING	50
IV.2. APPLICATION SUR LA BASE DE DONNEES <i>Mytripdata</i>	50
IV.3. MODELISATION BASE SUR MAMDANI	51
IV.4. MODELISATION BASE SUR SUGENO	55
<b>CONCLUSION GENERAL</b>	<b>60</b>
<b>BIBLIOGRAPHIE</b>	<b>61</b>

## Tableau de figure

Fig I.1 Classification des personnes en trois ensembles selon la logique classique.....	8
Fig. I.2 Classification des personnes en trois ensembles selon la logique floue .....	9
Fig. I.3 Classification des températures d'une pièce en deux ensembles.....	10
Fig. I.4 Le support d'un sous-ensemble flou.....	11
Fig. I.5 La hauteur du sous-ensemble flou.....	11
Fig. I.6 Le noyau d'un sous-ensemble flou .....	12
Fig. I.7 Exemples de fonction d'appartenance .....	13
Fig. I.8 L'opérateur NON de sous-ensemble flou A. ....	14
Fig. I.9 L'opérateur OU de deux sous-ensemble flou A et B.....	15
Fig. I.10 L'opérateur OU de deux sous-ensemble flou A et B.....	15
Fig. I.11 L'opérateur ET de deux sous-ensemble flou A et B. ....	16
Fig. I.12 L'opérateur ET de deux sous-ensemble flou A et B.....	17
Fig II.1. Système d'Inférence Flou (SIF).....	22
Fig II.2 Défuzzification par La méthode du maximum.....	26
Fig II.3 La défuzzification par centre de gravité.....	26
Fig II.4 Système d'inférence de Mamdani .....	28
Fig II.5 Système d'inférence de Sugeno et Takagi .....	30
Fig II.6 Système inférence floue.....	31
Fig II.7 l'inférence de pourboire par le système de Mamdani.....	32
Fig II.8 Les fonctions des appartenances de l'entrée plat. ....	33
Fig II.9 Les fonctions des appartenances de l'entrée service .....	33
Fig II.10 Les fonctions des appartenances de sortie pourboire.....	34

Fig II.11 Tableau des règles dans ce system par l'inférence Mamdani. ....	34
Fig II.12 Surface dans ce system par l'inférence Mamdani. ....	35
Fig II.13 L'exemple de pourboire par le system de Sugeno.....	36
Fig II.14 Tableau des règles dans ce system par l'inférence Sugeno.....	36
Fig II.15 Surface dans ce system par l'inférence Sugeno.....	37
Fig III.1 Exemple de classification avec K-Means.....	40
Fig III.2 Clustering de la base de données FCMDATA avec 2 clusters par FCM.....	44
Fig III.3 Clustering de la base de données FCMDATA avec 3 clusters par FCM.....	44
Fig III.4 Clustering de la base de données FCMDATA avec 2 clusters par subtractive	47
Fig III.5 Clustering de la base de données FCMDATA avec 3 clusters par subtractive. .....	48
Fig IV.1 Les entrées et le sortie des données d'apprentissage.....	51
Fig IV.2 Les fonctions des appartenances des entrées des données d'apprentissage. .....	52
Fig IV.3 Les fonctions des appartenances la sortie des données d'apprentissage par système de Mamdani.....	52
Fig IV.4 Les règles de système par le système de Mamdani. ....	53
Fig IV.5 La sortie calculée et la sortie désirée pour les données d'apprentissage par système de Mamdani.....	53
Fig IV.6 La sortie calculée et la sortie désirée pour les données de test par système de Mamdani. ....	54
Fig IV.7 Les fonctions des appartenances les entrées des données d'apprentissage par système de Sugeno.....	55
Fig IV.8 Les règles de système par le système de Sugeno.....	56
Fig IV.9 La sortie calculée et la sortie désirée pour les données d'apprentissage par système de Sugeno.....	56
Fig IV.10 La sortie calculée et la sortie désirée pour les données de test par système de Sugeno.....	57

Fig IV.11 L'erreur d'apprentissage et de test par le système de Mamdani. ....58

Fig IV.12 L'erreur d'apprentissage et de test par le système de Sugeno.....59

## Introduction générale

Le développement des modèles pour les systèmes réels complexes constitue un thème important dans beaucoup de domaines de la science et de l'ingénierie. Ces Modèles sont généralement utilisés dans la simulation, l'identification, la conception des contrôleurs, etc. les systèmes de Floue constitue un outil de représentation des connaissances d'une manière qui ressemble à la communication et au raisonnement humain. La modélisation floue est la tâche de trouver les paramètres du modèle floue de manière à obtenir le comportement souhaitée. Deux différentes approches sont utilisées pour la conception de modèles floue: conception basée sur les connaissances acquises par un expert et conception basée sur les données entrée-sortie.

Dans ce travail on s'intéresse à la deuxième approche, conception de modèle floue en se basant sur les données entré-sortie. On se base sur les deux systèmes d'inférence floue les plus utilisé : le système de Mamdani et le système de Sugeno. Plus précisément, nous allons utiliser les méthodes de clustering pour la génération automatique des règles floues. Dans cette approche chaque cluster est considéré comme une région qui sera représentée par une règle floue.

Le présent mémoire se compose de quatre chapitres :

Le premier chapitre constitue une introduction à la logique floue. Il décrit le cadre général de la logique floue, son historique et ses concepts de bases comme les sous-ensembles flous et les opérateurs flous.

Le deuxième chapitre décrit les systèmes d'inférences floues et leurs différentes étapes en mettant l'accent sur les deux systèmes les plus utilisés Mamdani et de Sugeno. Il présente également un exemple de modélisation en utilisant ces deux systèmes.

Le troisième chapitre décrit brièvement les méthodes de clustering. Il présente la méthode FCM et la méthode subtractive et un exemple en utilisant ces deux.

Dans le dernier chapitre nous appliquons les systèmes de Mamdani et de Sugeno pour modélisation de la base de données Mytripdata et nous évaluons les performances de ces deux systèmes.

# Chapter I : Concept de base de la logique floue

Ce chapitre constitue une introduction à la logique floue. Il décrit le cadre général de la logique floue, son historique et ses concepts de bases comme les sous-ensembles flous et les opérateurs flous.

## I.1. Introduction

Dans le raisonnement humain de nombreux concepts ne sont pas assez précis dans le sens d'être complètement vrai ou faux, plutôt ils peuvent être interprétés d'une manière plus qualitative. Dans la vie quotidienne, nous utilisons des concepts comme grand, petit, rapide, lent, bon, mauvais ... qui sont difficiles à traduire numériquement. La logique et l'inférence classiques ont été insuffisantes pour faire face à ces concepts vagues. Bien que l'homme résonne avec ces concepts de façon naturelle dans sa vie quotidienne, notre quête de connaissances scientifiques nous a conduits à aborder le problème de la représentation de ces concepts d'une manière plus systématique et plus précise. C'était Zadeh qui a formalisé pour la première fois le concept du flou, et plus tard le domaine de la logique floue a suscité l'intérêt d'une grande part de la communauté scientifique, la théorie des ensembles flous et la logique floue ont été appliquées dans presque tous les domaines.

## I.2. Historique

En 1965, le professeur Lotfi A. Zadeh, de l'université de Berkeley (Californie), a déclaré: «un contrôleur électromécanique doté d'un raisonnement humain serait plus performant qu'un contrôleur classique ». Il a publié un article où il introduit le terme « fuzzy » dans la littérature technique et donne naissance au concept d'ensemble flou. Le professeur tentait de modéliser les phénomènes sous forme floue afin de remédier aux limitations dues aux incertitudes des modèles classiques à équations différentielles. Il a réalisé depuis, des avancées majeures dans le domaine et a été rapidement accompagné par de nombreux chercheurs développant des travaux théoriques.

En 1974, le docteur Mamdani, de l'Université de Londres, a introduit la commande floue dans la régulation d'un processus industriel, en réalisant un contrôleur flou expérimental sur un moteur à vapeur. Le problème sur lequel

il s'était penché, avait la réputation d'être difficile et les résultats obtenus étaient très encourageants. En 1978, la société danoise F.L. Smidth & Co a réalisé la première véritable application industrielle de la logique floue : le contrôle d'un four à ciment.

A partir des années 80, les recherches se sont arrêtées en Europe et ont été reprises au Japon où plusieurs applications ont commencé à émerger.

En 1985, la logique floue a été introduite au Japon par le chercheur M. Sugeno. Les sociétés japonaises ont très vite compris son avantage à la fois technique et commercial. La recherche n'était pas uniquement théorique mais également appliquée. A la fin des années 80, ce fut une véritable 'explosion du flou'. Les Japonais ont mis sur le marché plus de 2000 produits flous destinés au grand public, notamment des appareils électroménagers et du matériel audio-visuel. Dans l'industrie, l'application de la logique floue s'est étendue au traitement des eaux, aux grues portuaires, aux métros, aux systèmes de ventilation et de climatisation ainsi qu'à d'autres domaines très différents tels que la finance ou le diagnostic médical.

A partir de 1990, apparaissent de nombreuses applications en Allemagne et à moindre échelle aux Etats Unis d'Amérique.

De nos jours, une intense activité de recherche a débuté afin de combiner la logique floue aux algorithmes génétiques et aux réseaux de neurones.

### **I.3. Domaines d'application**

Les applications pratiques déjà connues sont innombrables, nous en citons quelques-unes :

Contrôle :

- Aviation (Rockwell)
- Métro (Hitachi)

- Boîte de vitesses (Nissan, Subaru)
- Amarrage de navettes spatiales (NASA)

Ordonnancements :

- Ascenseurs (Mitsubishi, ...)

Traitement du signal :

- Reconnaissance de caractères (Palm Sony)
- Autofocus (Sanyo, Canon)
- Stabilisateur d'images (Panasonic)

La logique floue comporte des applications dans d'autres domaines tels que la recherche opérationnelle, la gestion financière, le contrôle de production, la robotique, la météorologie, l'informatique, la médecine, l'intelligence artificielle, la neuromimétique, la linguistique, le traitement des données, la communication homme-machine, la reconnaissance des formes, la fiabilité, la chimie industrielle....

Aujourd'hui, nous avons beaucoup de possibilités de travail avec des logiciels qui comportent la logique floue :

- C++ : "fuzzy-lite", "jFuzzyQt".
- Java : "jFuzzyLogic".
- MATLAB : "Fuzzy Logic Toolbox " (utilisé dans ce mémoire).
- Octave : Fuzzy Logic Toolkit.
- Python : pyfuzzy.

#### **I.4. Définition de la logique floue**

La logique floue est une généralisation de la logique booléenne, elle se base sur la théorie des ensembles flous, qui est, à son tour, une généralisation de la théorie des ensembles classiques. En introduisant la notion de degré dans la

vérification d'une condition, la logique floue permet à une condition d'être dans un autre état que vrai ou faux. La logique floue donne ainsi une flexibilité aux raisonnements en tenant compte des imprécisions et des incertitudes.

De nombreux auteurs tentent de relier le flou à des probabilités. Dans certains cas, le flou et les probabilités peuvent être traitée avec des règles similaires, mais il est très important de rappeler que le flou présente le degré d'appartenance d'un élément à un certain ensemble, tandis que les probabilités permettent de représenter de degré de certitude d'apparition de plusieurs éléments.

### **I.5. Exemple illustratif de la logique floue**

Afin de mettre en évidence le principe fondamental de la logique floue, on présente un exemple simple, celui de la classification des personnes en trois ensembles « Jeune », « Homme », « Vieux ».

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

Cependant, une telle logique de classification n'est même pas logique. Pourquoi une personne, lorsqu'elle a eu 50 ans, doit-elle être considérée comme appartenant à l'ensemble «vieux»? En réalité, un tel passage se fait progressivement et individuellement. [1]

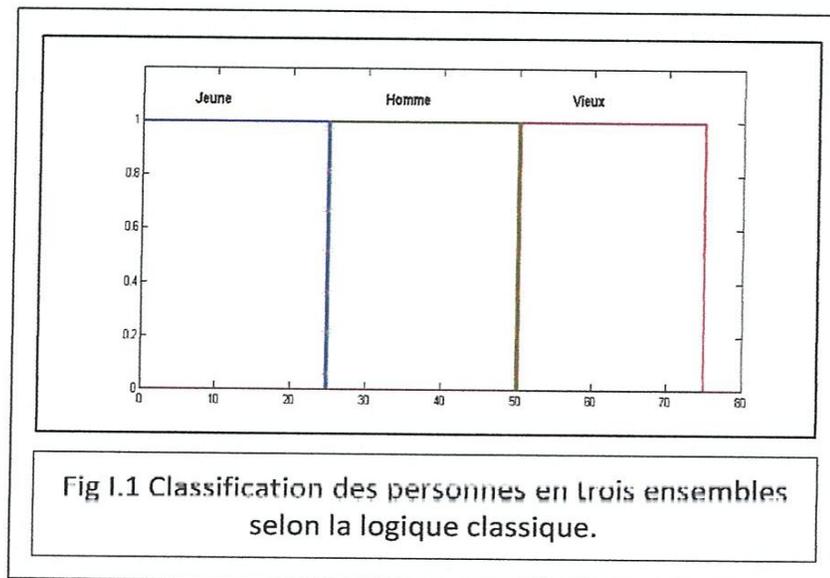


Fig I.1 Classification des personnes en trois ensembles selon la logique classique.

La logique floue, dont les variables peuvent prendre n'importe quelle valeur entre 0 et 1 (on parle alors de fonction d'appartenance  $\mu$ ), permet de tenir compte de cette réalité. Les limites ne varient pas soudainement, mais progressivement. La figure (I.2) montre une classification possible pour l'exemple précédent, cette fois-ci, à l'aide de la logique floue. Ainsi une personne de 25 ans appartient à l'ensemble «jeune» avec une valeur  $\mu=0,75$  de la fonction d'appartenance et l'ensemble «homme» avec une valeur  $\mu=0,25$ . Par contre, une personne de 70 ans appartient avec une valeur  $\mu=1$  de la fonction d'appartenance à l'ensemble «vieux». Evidemment la forme des fonctions d'appartenance n'est pas rigide et peut dépendre du contexte envisagé, à savoir professionnel, sportif ou autre. [1]

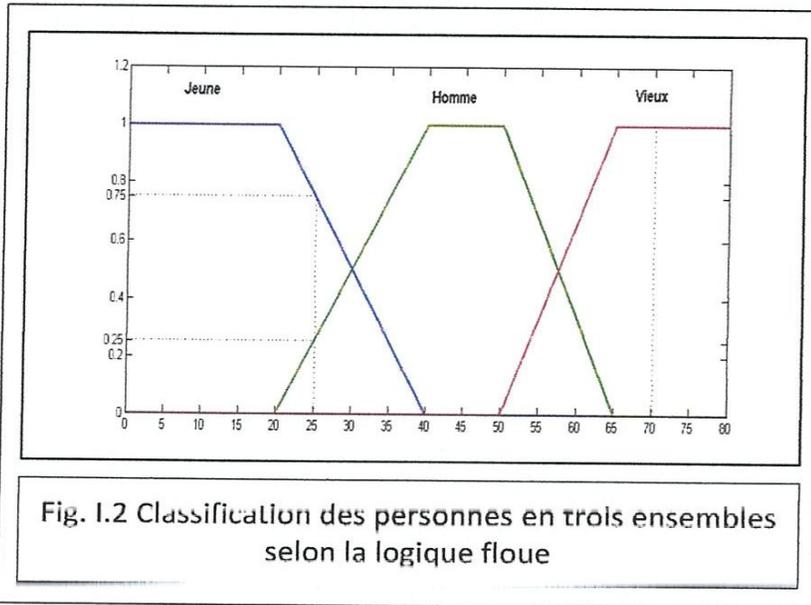


Fig. I.2 Classification des personnes en trois ensembles selon la logique floue

### I.5.1. Les sous-ensembles floue

Le concept des ensembles flous se pose comme une réponse aux problèmes des incertitudes et de l'absence de précision trouvés dans des ensembles classiques. Les ensembles flous sont plus inhérents à la nature que les ensembles classiques.

### I.5.2. Définition d'un sous-ensemble floue

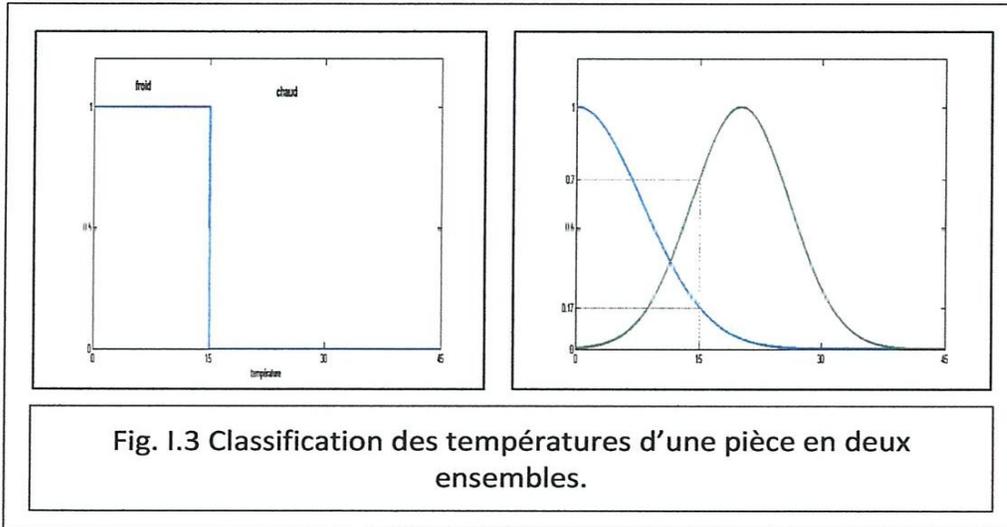
Dans la théorie des ensembles classique, l'appartenance d'un élément à un sous-ensemble est définie par une valeur logique standard (1 si l'élément appartient au sous-ensemble, 0 sinon), dans un univers de  $X$  est caractérisé par sa fonction d'appartenance  $\mu_A(x)$  qui associe à chaque élément  $x \in X$  une valeur dans l'intervalle des nombres réels  $[0, 1]$ :

$$\mu_A: X \rightarrow \{0,1\} \quad (1.1)$$

Dans la théorie floue, un élément peut appartenir en partie à un sous-ensemble: son degré d'appartenance est décrit par une valeur comprise entre 0 et 1. [2]

Un sous-ensemble flou  $A$  dans un univers de  $X$  est caractérisé par sa fonction d'appartenance  $\mu_A(x)$  qui associe à chaque élément  $x \in X$  une valeur dans l'intervalle des nombres réels  $[0, 1]$  :

$$\mu_A(X) \rightarrow [0,1] \quad (1.2)$$



*a. Caractéristiques d'un sous-ensemble flou*

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. [3]

*i. Support*

Le support d'un sous-ensemble flou de  $A$  de  $X$ , noté  $Supp(A)$ , est l'ensemble de tous les éléments qui lui appartiennent au moins un petit peu Formellement:

$$supp(A) = \{x \in X / \mu_A(x) \neq 0\} \quad (1.3)$$

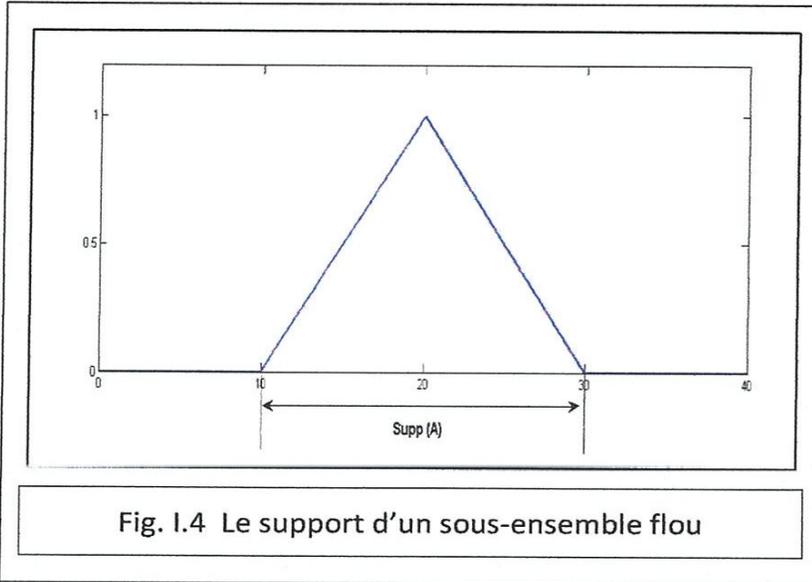


Fig. I.4 Le support d'un sous-ensemble flou

*ii. La Hauteur*

La hauteur du sous-ensemble flou  $A$  de  $X$ , notée  $h(A)$ , est le plus fort degré avec lequel un élément de  $X$  appartient à  $A$ . Formellement:

$$h(A) = \sup_{x \in X} \{\mu_A(x)\} \quad (1.4)$$

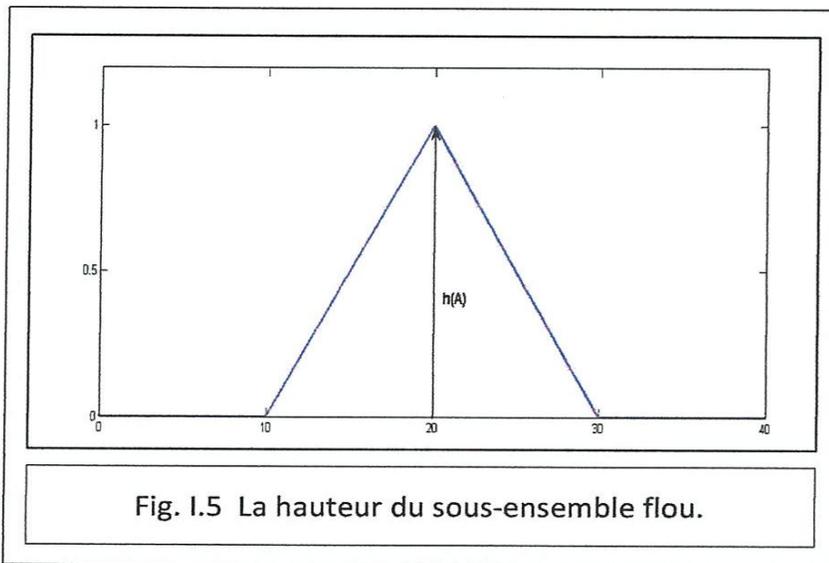


Fig. I.5 La hauteur du sous-ensemble flou.

*iii. Noyau*

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

$$\text{noy}(A) = \{x \in X / \mu_A(x) = 1\} \quad (1.5)$$

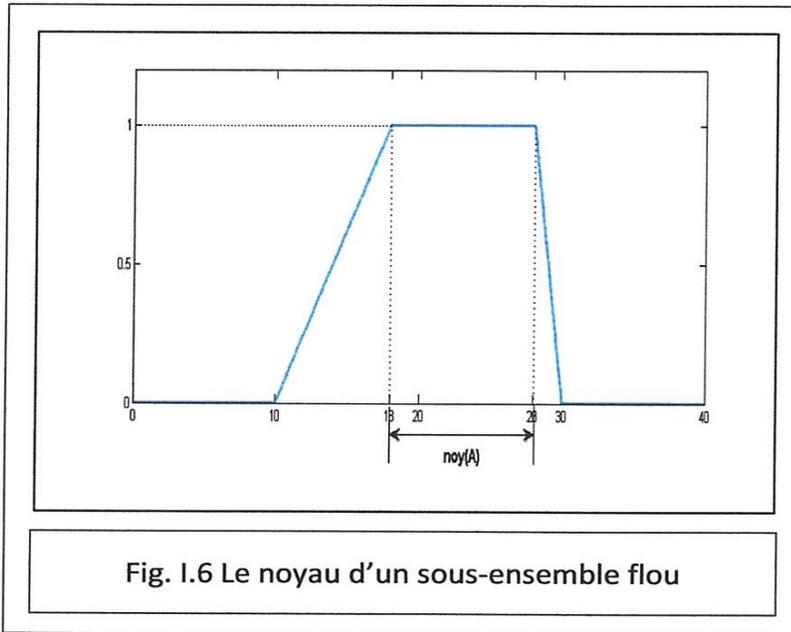


Fig. 1.6 Le noyau d'un sous-ensemble flou

**b. Les différentes formes de fonction d'appartenance**

La logique floue est basée sur la notion d'appartenance partielle ; un élément peut alors appartenir partiellement à plusieurs sous-ensembles. Un sous-ensemble flou  $B$  est défini par une fonction d'appartenance  $\mu_B$  qui associe à chaque élément  $x$  un degré  $\mu_B(x)$  compris entre 0 (non-appartenance) et 1 (appartenance totale).

La logique floue possède des règles de calcul bien précises permettant de combiner plusieurs fonctions d'appartenance et de tirer des conclusions pondérées :

Fonction d'appartenance trapézoïdale :

$$u(x, a, b, d) = \max\left(0, \min\left(\frac{x - a}{b - a}, 1, \frac{d - x}{d - c}\right)\right) \quad (1.6)$$

Fonction d'appartenance triangulaire :

$$u(x, a, b, d) = \max\left(0, \min\left(\frac{x - a}{b - a}, \frac{c - x}{c - b}\right)\right) \quad (1.7)$$

Fonction d'appartenance gaussienne :

$$f(x) = e^{-\frac{(x-u)^2}{2\sigma^2}} \quad (1.8)$$

Fonction d'appartenance cloche généralisée :

$$f(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (1.9)$$

Fonction d'appartenance sigmoïdale :

$$f(x) = \frac{1}{1 + e^{-a(x-c)}} \quad (1.10)$$

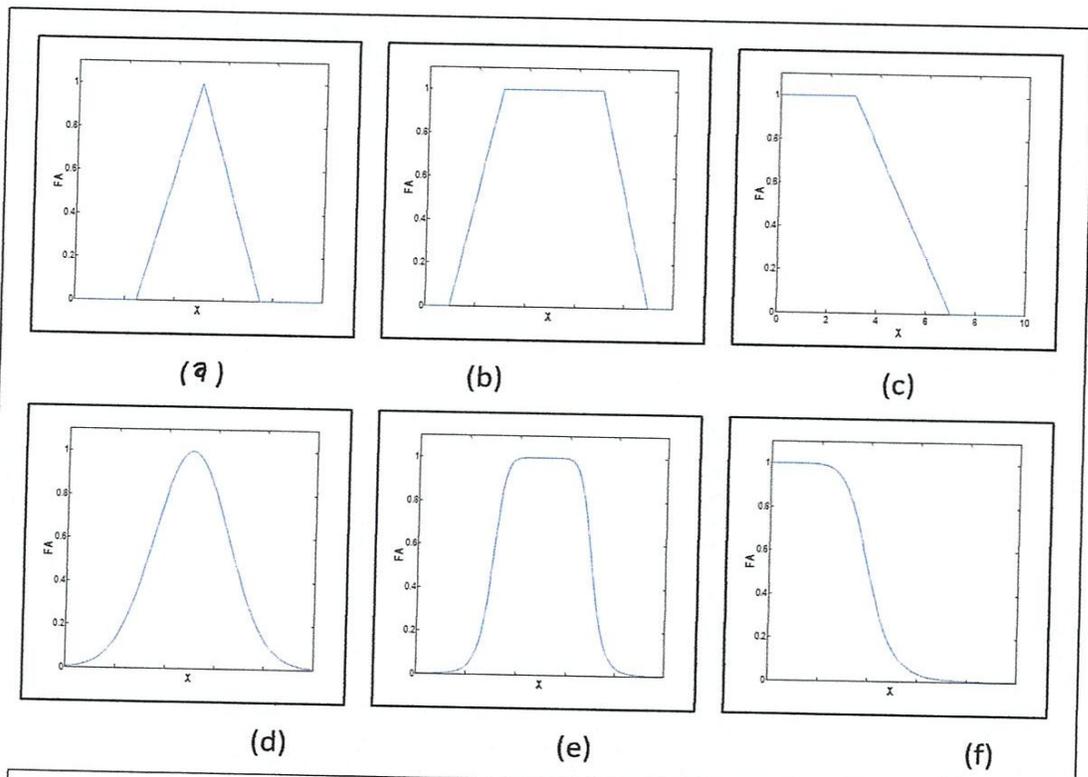


Fig. I.7 Exemples de fonction d'appartenance

- (a) Fonction triangulaire
- (b) Fonction trapézoïdale
- (c) Fonction linéaire saturable
- (d) Fonction gaussienne
- (e) Fonction cloche généralisée
- (f) Fonction sigmoïdale

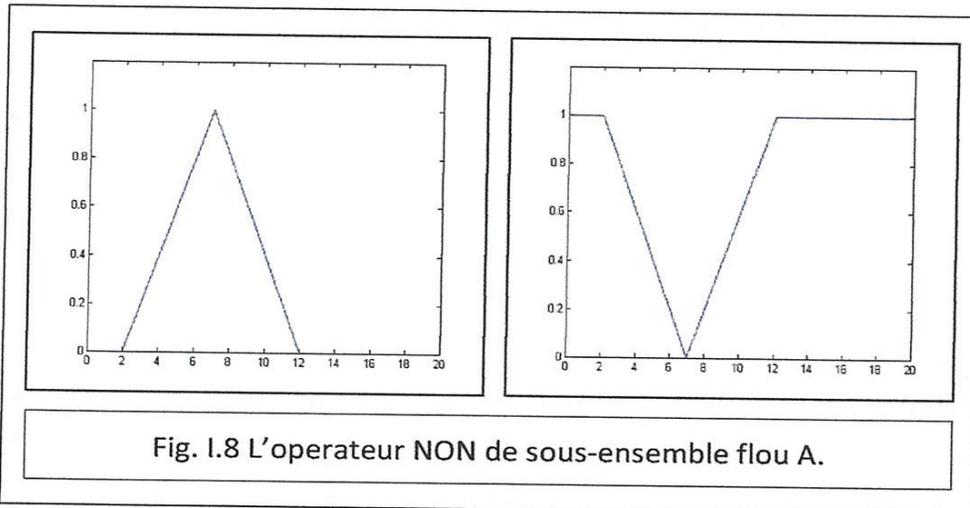
### I.5.3. Les Operateurs floues

Les règles d'inférences font appel aux opérateurs « et », « ou » et « non », qui s'appliquent aux variables floues. Dans le cas de la logique binaire ces opérateurs sont définis de façon simple et univoque. Dans le cas de la logique floue, la définition de ces opérateurs n'est plus univoque. [4]

#### a. L'opérateur Non

La fonction de membre du complément d'un ensemble flou B avec la fonction d'appartenance  $\mu(A)$  est définie comme la négation de la fonction d'appartenance spécifiée. Ceci est appelé le critère de négation.

$$\mu(B) = 1 - \mu(A) \quad (1.)$$



#### b. L'opérateur OU

La fonction de membre de l'intersection de deux ensembles flous A et B avec des fonctions d'appartenance  $u_A$  et  $u_B$  respectivement est défini comme le minimum des deux fonctions individuelles d'adhésion. Ceci est appelé le critère minimum.

$$u_{A \cap B} = \min(u_A, u_B) \quad (1.)$$

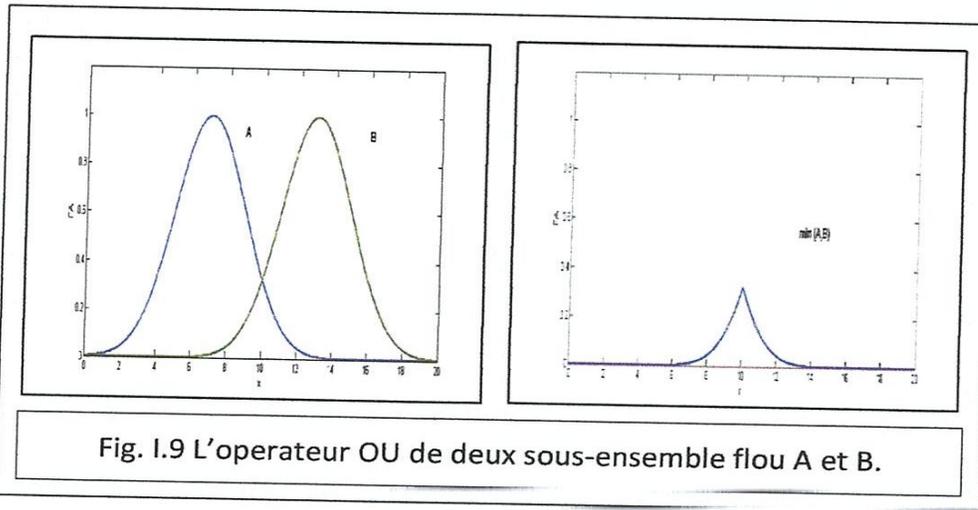


Fig. I.9 L'opérateur OU de deux sous-ensemble flou A et B.

Méthode de Produit algébrique

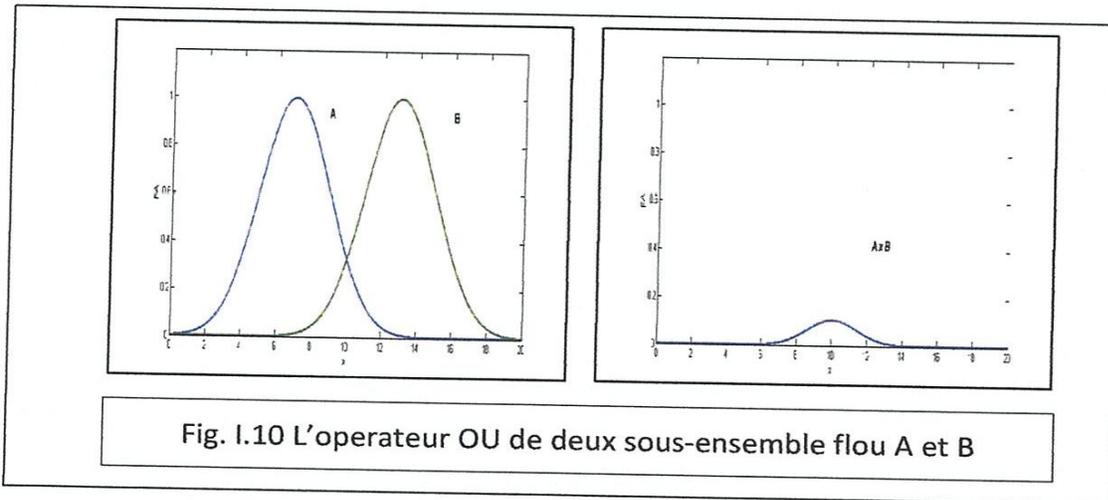


Fig. I.10 L'opérateur OU de deux sous-ensemble flou A et B

Cet opérateur peut être réalisé de plusieurs façons. Cependant les solutions les plus simples et les plus utilisées pour caractériser la satisfaction simultanée de deux propriétés sont [5]:

Opérateur minimum (de Zadeh):

$$u_{A \cap B}(x) = \min \{ u_A(x), u_B(x) \}; \quad (1.12)$$

Produit algébrique (représentation probabilistique) :

$$u_{A \cap B}(x) = u_A(x) \times u_B(x) \quad (1.13)$$

Produit borné (d'après Lukasiewicz) :

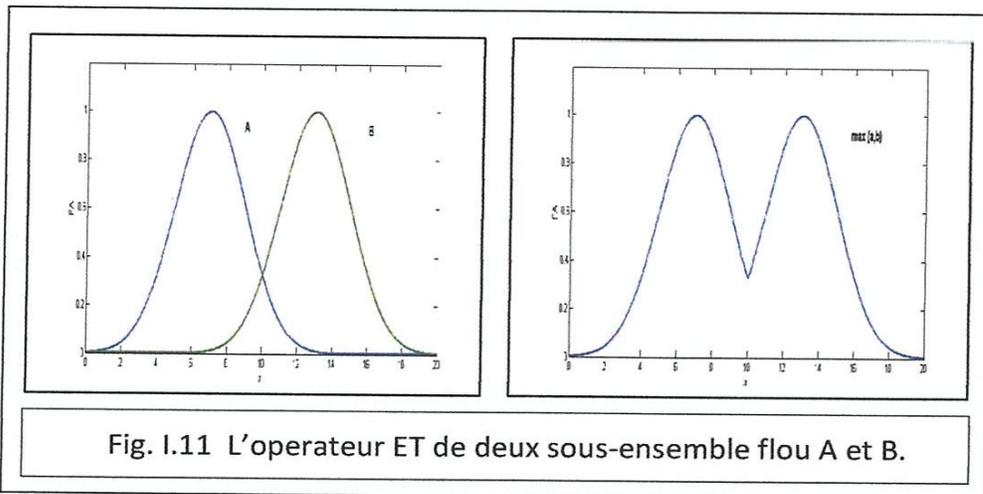
$$u_{A \cap B}(x) = \max \{ u_A(x) + u_B(x) - 1, 0 \} \quad (1.14)$$

Il est à noter qu'il existe plusieurs d'autres approche pour représenter le OU flou comme : le Produit d'Hamasher, le produit d'Einstein et le produit drastique de Weber.

*c. L'opérateur Et*

La fonction de membre de l'Union des deux ensembles flous A et B avec des fonctions d'appartenance  $\mu(A)$  et  $\mu(B)$  respectivement est définie comme le maximum des deux fonctions individuelles d'adhésion. Ceci est appelé le critère maximale.

$$u_{A \cup B} = \max(u_A, u_B) \quad (1.15)$$



Méthode de Produit algébrique

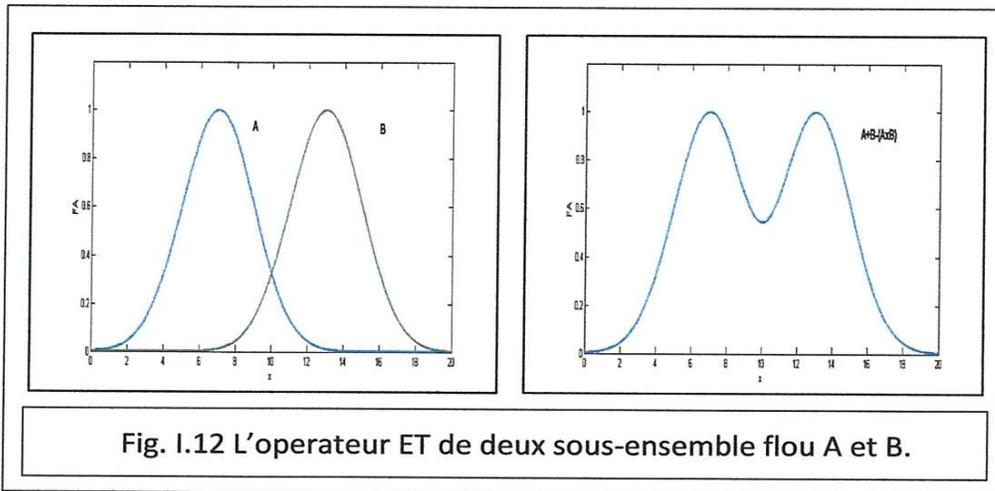


Fig. 1.12 L'opérateur ET de deux sous-ensemble flou A et B.

Tout comme pour l'opérateur ET, il existe de nombreuses possibilités pour représenter l'opérateur OU. Les plus répandus sont [5]:

Opérateur minimum (de Zadeh):

$$u_{A \cup B}(x) = \max \{ u_A(x), u_B(x) \}; \quad (1.16)$$

Produit algébrique (représentation probabilistique) :

$$u_{A \cup B}(x) = u_A(x) + u_B(x) - u_A(x)u_B(x) \quad (1.17)$$

Produit borné (d'après Lukasiewicz) :

$$u_{A \cup B}(x) = \min \{ u_A(x) + u_B(x), 1 \} \quad (1.18)$$

Il est à noter qu'il existe plusieurs d'autres approche pour représenter le ET flou comme : le Produit d'Hamasher, le produit d'Einstein et le produit drastique de Weber.

## I.6. Propriétés de FL

La logique floue admet les propriétés suivantes :

- La logique floue est conceptuellement facile à comprendre :

Les concepts mathématiques en raisonnement flou sont très simples. La logique floue est une approche plus intuitive.

- La logique floue est flexible.

Pour un système donné, il est facile d'intégrer plus de fonctionnalités sans repartir de zéro.

- La logique floue est tolérante des données imprécises.

La logique floue constitue une meilleure représentation des données imprécises.

- La logique floue peut modéliser des fonctions non linéaires de complexité arbitraire.

On peut créer un système flou pour correspondre à tout ensemble de données d'entrée-sortie. Ce processus est particulièrement facile par des techniques adaptatives.

- La logique floue peut être construite en se basant sur l'expérience des experts.

Contrairement aux les réseaux de neurones, qui utilisent seulement les données d'apprentissage et génèrent des modèles non interprétable. La logique floue permet d'exploiter l'expérience des gens qui ont déjà de l'expérience sur le système.

- La logique floue peut être combinée avec les techniques de contrôle classiques.

Les systèmes flous ne remplacent pas nécessairement les méthodes de contrôle classiques. Dans de nombreux cas, les systèmes flous les améliore et simplifier leur mise en œuvre.

- La logique floue est basée sur le langage naturel.

Le flou est à la base de la communication humaine. La logique floue est construite autour des structures de description qualitative utilisée dans le langage courant, la logique floue est donc facile à utiliser.

Ce dernier point est peut-être le plus important. En effet, le langage naturel, qui est utilisé quotidiennement par des gens ordinaires, a été façonné par des milliers d'années d'histoire humaine pour être pratique et efficace. Les phrases écrites dans les langues ordinaires représentent un triomphe de la communication efficace. [6]

# Chapter II : Système D'inférence Floues

Ce chapitre décrit les systèmes d'inférences floues et leurs différentes étapes en mettant l'accent sur les deux systèmes les plus utilisés Mamdani et de Sugeno. Il présente également un exemple de modélisation en utilisant ces deux systèmes

## II.1. Les systèmes d'inférence flous

Les systèmes d'inférence flous (SIF), appelés aussi les systèmes experts flous (Fuzzy expert systems), les modèles flous (Fuzzy Models), ou simplement systèmes flous (Fuzzy systems), sont des structures de calcul basées sur le concept de la théorie des ensembles flous, des règles si-alors floues et du raisonnement flou. En raison de leur caractère pluridisciplinaire, ils ont été appliqués avec succès dans une grande variété d'applications, tels que l'automatique, la robotique, la reconnaissance des formes ainsi que les systèmes d'aide à la décision.

Un SIF se compose essentiellement de trois éléments conceptuels figure (II.1): une base de règles qui contient une collection de règles floues, une base de données qui définit les fonctions utilisées dans les règles floues, et un mécanisme de raisonnement qui effectue la procédure d'inférence en se basant sur les règles. Ce processus s'effectue en trois phases:

**Fuzzification :** Cette étape consiste à déterminer le degré d'appartenance de chaque variable d'entrée aux sous-ensembles flous utilisés dans les prémisses des règles.

**Inférence :** Il s'agit de la détermination des degrés d'appartenances des règles par la combinaison des propositions de leurs prémisses et du calcul des sorties.

**Défuzzification :** Le rôle de cette étape est de fournir des valeurs précises des variables de sorties du système afin qu'elle soit exploitable. Par exemple dans le cas de la commande industriel, la sortie du système constitue le signal envoyé au processus.

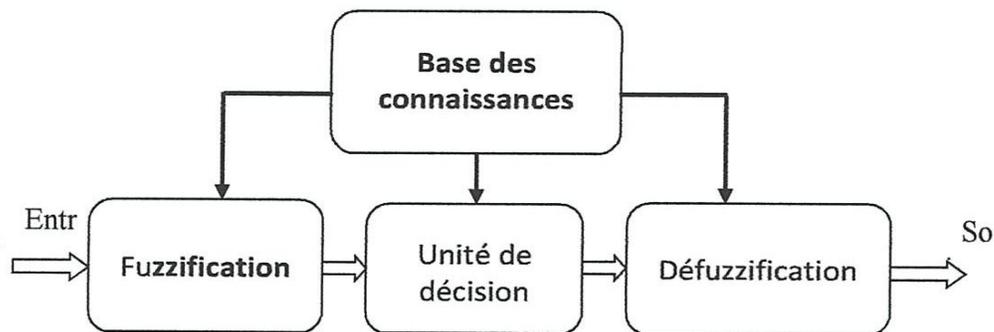


Fig II.1. Système d'Inférence Flou (SIF)

### II.1.1. Fuzzification

La fuzzification est réalisée dans l'interface d'entrée du système flou. Cette étape consiste à déterminer le degré d'appartenance de chaque variable d'entrée aux sous-ensembles flous utilisés dans les prémisses des règles.

Les entrées dans un système flou sont en général mesurées à l'aide d'organes de mesures qui sont le plus souvent de type analogique. Etant donné que, l'implémentation du système flou se fait toujours en numérique, il faut d'abord convertir les entrées analogiques en digital, puis on procède à la fuzzification. L'adaptation des entrées permet de garantir l'appartenance de celles-ci à l'espace caractéristique choisi. La dernière étape de fuzzification est la génération des degrés d'appartenance à chaque valeur linguistique définie par sa fonction d'appartenance.

### II.1.2. Inférence floue

L'inférence floue est le processus de formulation de la relation entre les entrées et les sorties par logique floue. Cette relation offre une base avec laquelle la décision est prise par le système flou. L'inférence floue fait appel alors aux concepts de fonctions d'appartenance, d'implication floue et les règles floues.

$$\underbrace{XestA}_{\text{prémise}} \rightarrow \underbrace{YestB}_{\text{conclusion}} \quad (2.1)$$

### *a. Les méthodes des inférences*

Il existe plusieurs méthodes pour définir les degrés d'appartenance de la variable de sortie à ses sous-ensembles flous. Elles se différencient essentiellement par la manière dont vont être réalisés les opérateurs Et et OU utilisés dans les règles d'inférence.

Les trois méthodes d'inférence les plus usuelles sont : Max-min, Max-produit et Somme-produit.

#### *i. La méthode d'inférence max-min :*

L'implication floue et l'opérateur ET sont réalisés par la fonction minimum et l'opérateur logique OU est réalisé par la fonction maximum. Cette méthode est également dite "implication de Mamdani". La fonction d'appartenance résultante correspond au maximum des deux fonctions d'appartenance partielles puisque les règles sont liées par l'opérateur OU.

#### *ii. La méthode d'inférence max-prod :*

La seule différence avec la méthode précédente est la réalisation de l'implication floue par le produit. Les opérateurs logiques ET et OU sont toujours réalisés respectivement par les fonctions minimum et maximum. La méthode max-prod est dite implication de Larsen.

#### *iii. La méthode d'inférence somme-prod :*

La méthode d'inférence somme-prod réalise, au niveau de la condition, l'opérateur OU par la fonction somme, et l'opérateur ET par la fonction produit. Au niveau de la conclusion, elle réalise l'implication par la fonction produit.

### *b. Les règles Si-Alors floues*

Les règles Si-Alors floues (appelées aussi règles floues) admettent généralement la forme suivante :

$$\text{Si } x \text{ est } A \text{ et } y \text{ est } B \text{ Alors } z \text{ est } C \quad (2.2)$$

Où  $A, B$  et  $C$  sont des variables linguistiques définies par des sous-ensembles flous.

Une règle floue se compose de deux parties :

La prémisse : formée par une combinaison de prépositions liées entre elles par les opérateurs Et, Ou et Non. Dans l'exemple précédent, la proposition «  $x \text{ est } A \text{ et } y \text{ est } B$  » constitue la prémisse de cette règle.

La conclusion : c'est la partie qui définit la conséquence de la règle. Dans la règle précédente, la conclusion est la proposition «  $z \text{ est } C$  ».

La forme de ces règles est très employée dans notre expression quotidienne, par exemple :

$$\text{Si la Résistance est Grande Alors le Courant est Faible} \quad (2.3)$$

Où *le Courant* et *la Résistance* sont des variables floues et *Faible* et *Grande* sont des termes linguistiques caractérisés par des fonctions d'appartenances.

Une autre forme de règles Si-Alors floues, proposée par Sugeno et Takagi, comporte une conclusion numérique. Dans ces règles, les sous-ensembles flous sont introduits seulement dans les prémisses. Ce type de règles admet la forme :

$$\text{Si } x \text{ est } A \text{ et } y \text{ est } B \text{ Alors } z = f(x, y) \quad (2.4)$$

La fonction  $f(x, y)$  est généralement un polynôme de variables d'entrée  $x$  et  $y$ , mais elle peut être n'importe quelle fonction tant qu'elle permet de décrire convenablement la sortie du modèle en se basant sur les régions floues définie par les prémisses des règles. Un exemple de telles règles est donné par :

*Si la Fréquence est Grande Alors la Tension est 220 V* (2.5)

La conclusion (*Tension*) est numérique.

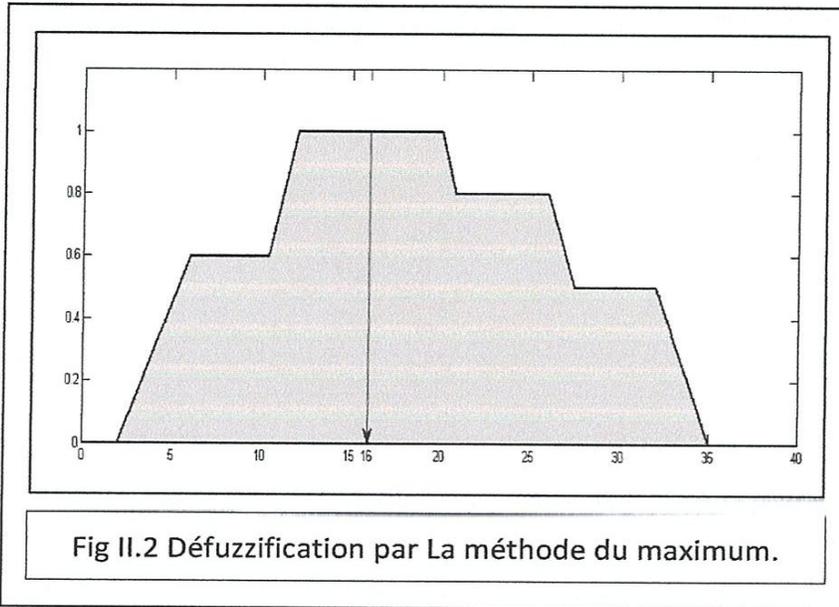
En se basant sur un ensemble de règles floues, un SIF met en œuvre une application non linéaire de l'espace d'entrée à l'espace de sortie. Chacune des règles utilisées décrit le comportement local de cette application ; sa prémisse définit une région de l'espace d'entrée et sa conclusion précise la sortie dans cette région.

### II.1.3. Défuzzification

Le résultat de l'inférence en utilisant une des méthodes d'implication floue, comme Max-min, est une valeur floue. Cette information ne peut être utilisée directement. Une transformation doit être prévue à la sortie du bloc d'inférence pour la convertir en grandeur fixe, cette transformation étant connue par le terme défuzzification (concrétisation). Plusieurs méthodes de défuzzification existent en logique floue, les plus utilisées sont les deux suivant :

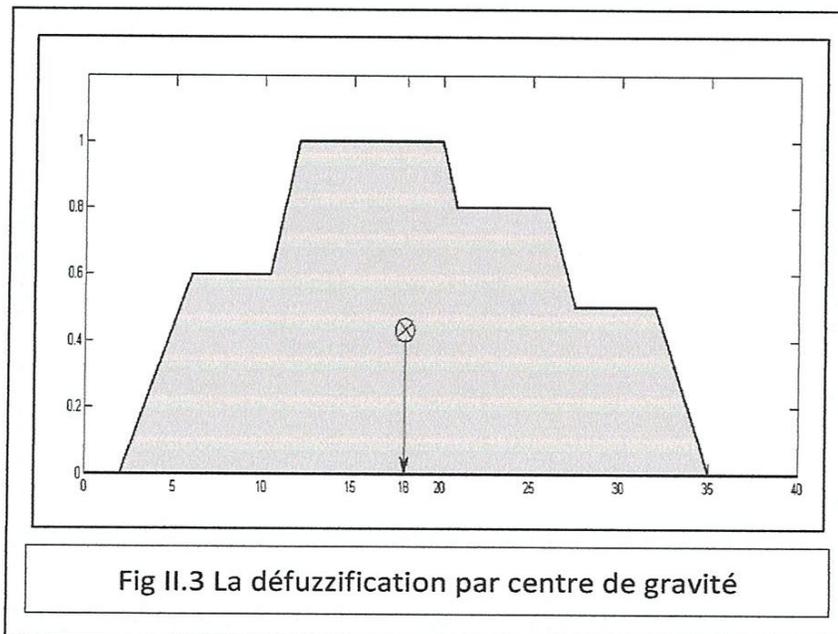
#### *a. La méthode du maximum*

Dans cette méthode, la valeur de sortie est estimée par l'abscisse du point correspondant au centre de l'intervalle pour lequel la fonction d'appartenance est maximale.



*b. La défuzzification par centre de gravité*

Il s'agit de calculer le centre de gravité de la fonction d'appartenance de la variable de sortie.



Le calcul du centre de gravité permet bien d'obtenir une seule valeur pour la grandeur de sortie. Son calcul est cependant relativement complexe puisqu'il nécessite le calcul d'une intégrale.

## II.2. Systeme d'Inférence Flou de Mamdani

Le Systeme d'Inférence Flou de Mamdani a été proposé en premier lieu comme une tentative de contrôle d'une machine à vapeur et de chaudières. Ce système est basé sur l'utilisation d'un ensemble de règles linguistiques obtenue à partir de l'expérience humaine.

Ce système s'effectue en se basant sur six étapes :

1. Détermination d'un ensemble des règles floues.
2. Fuzzification des entrées en utilisant les fonctions d'appartenances des entrées correspondantes.
3. Combinaison des entrées fuzzifiées correspondantes aux règles floues.
4. Détermination des conséquences des règles.
5. Combinaison des conséquences pour avoir la distribution de la sortie.
6. Défuzzification de la sortie.

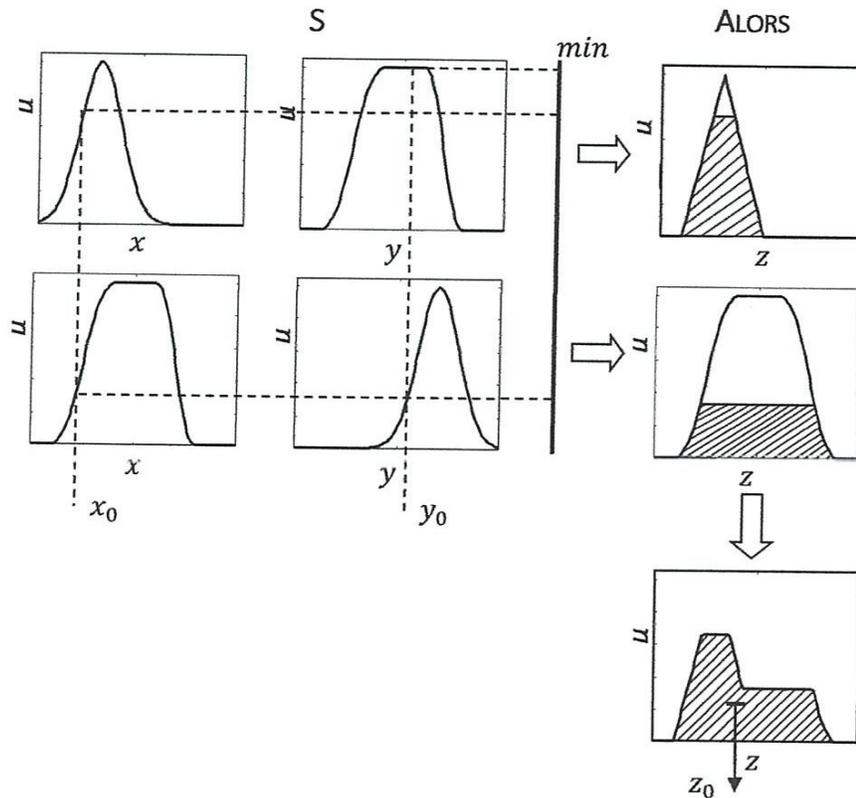


Fig II.4 Système d'inférence de Mamdani

La figure (II.4) illustre un exemple de ce système avec deux variables d'entrée  $x$  et  $y$ . La fuzzification de la variable d'entrée  $x$  s'effectue en utilisant deux sous-ensembles flous  $A_1$  et  $A_2$ , alors que celle de  $y$  s'effectue en utilisant  $B_1$  et  $B_2$ .

La sortie est déterminée en calculant le centre de gravité de la distribution de sortie déterminée par la somme des conséquences de règles.

### II.3. Système d'Inférence Flou de Sugeno et Takagi

Le système de Sugeno et Takagi a été proposé afin de concevoir un système de génération des règles floues en se basant sur les entrée/sortie. Ce système emploie des règles de la forme donnée par l'Equation 2.4. Il se diffère ainsi du système de Mamdani sur la façon de détermination de la sortie. Le système de Sugeno et Takagi détermine les conséquences des règles par une

combinaison linéaire des entrées avec des constantes, la sortie finale sera ensuite calculée par la moyenne pondérée des conséquences des règles.

Un exemple à deux variables d'entrée et deux règles est illustré sur la figure (II.5). Les deux règles sont données par :

$$\begin{cases} \mathcal{R}_1: \text{Si } x \text{ est } A_1 \text{ et } y \text{ est } B_1 \text{ Alors } z_1 = p_1x + q_1y + c_1 \\ \mathcal{R}_2: \text{Si } x \text{ est } A_2 \text{ et } y \text{ est } B_2 \text{ Alors } z_2 = p_2x + q_2y + c_2 \end{cases} \quad (2.6)$$

Les degrés d'activation de ces deux règles sont donnés par :

$$\begin{cases} \alpha_1 = \mu_{A_1}(x_0) \text{ ET } \mu_{B_1}(y_0) \\ \alpha_2 = \mu_{A_2}(x_0) \text{ ET } \mu_{B_2}(y_0) \end{cases} \quad (2.7)$$

Les sorties sont :

$$\begin{cases} z_1 = p_1x + q_1y + c_1 \\ z_2 = p_2x + q_2y + c_2 \end{cases} \quad (2.8)$$

La sortie finale du système est donnée par :

$$z = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2} \quad (2.9)$$

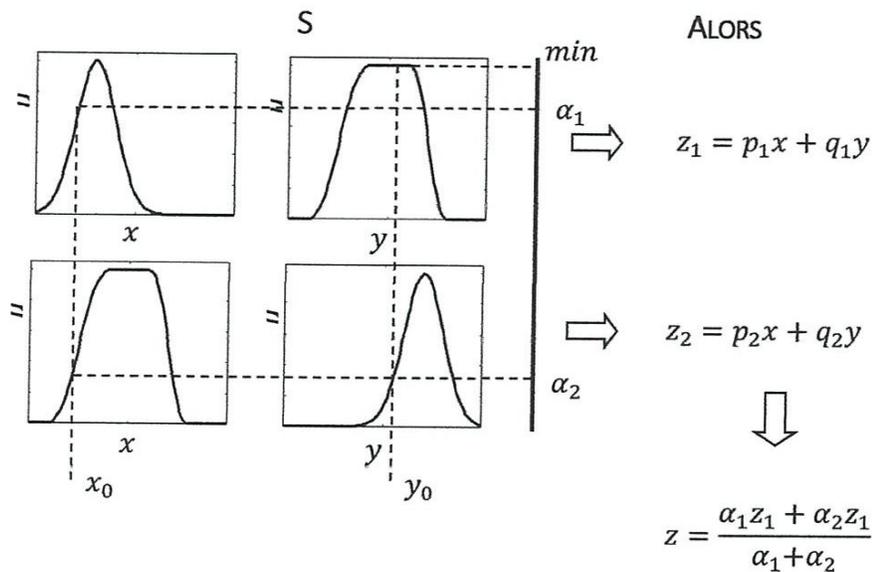


Fig II.5 Système d'inférence de Sugeno et Takagi

## II.4. Systèmes flous sous MATLAB

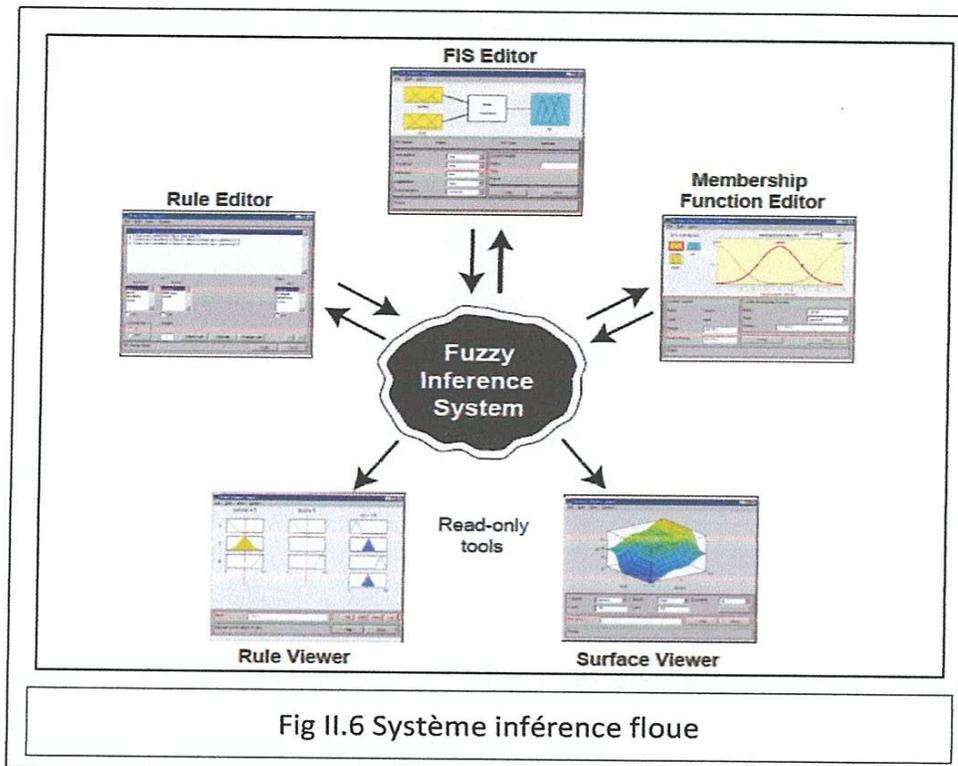
Le MATALB offre la possibilité d'implémenter les systèmes d'inférence flous avec une interface graphique qui contient les outils suivant (fig. 2.6):

Editeur fonction d'appartenance: Pour définir les formes des fonctions d'appartenances associées à chaque variable d'entrée.

Editeur de Règles: pour éditer la liste des règles qui définit le comportement du système.

Visualisation de Règles : Pour visualiser le diagramme de l'inférence floue. Il permet de montrer, pour une entrée, quelle règle est active et comment les fonctions d'appartenance influencent la forme du résultat.

Visualisation de la surface: permettant de visualiser la dépendance de l'une des sorties avec n'importe quelle deux entrées. Donc, il génère et visualise une carte de la surface de sortie du système. [6]



#### II.4.1. Exemple de modélisation avec le système de Mamdani

Pour présenter le système de Mamdani nous considérons l'exemple du pourboire « Basic Tipping Problem ». Cet exemple utilise deux entrées et une seule sortie pour modéliser les pratiques de pourboire aux États-Unis.

Etant Donné un certain nombre entre 0 et 10 qui représente la qualité du service dans un restaurant (où 10 est excellent), et un autre numéro entre 0 et 10 qui représente la qualité du plat (encore une fois que, 10 est excellent), combien serait-il le pourboire?

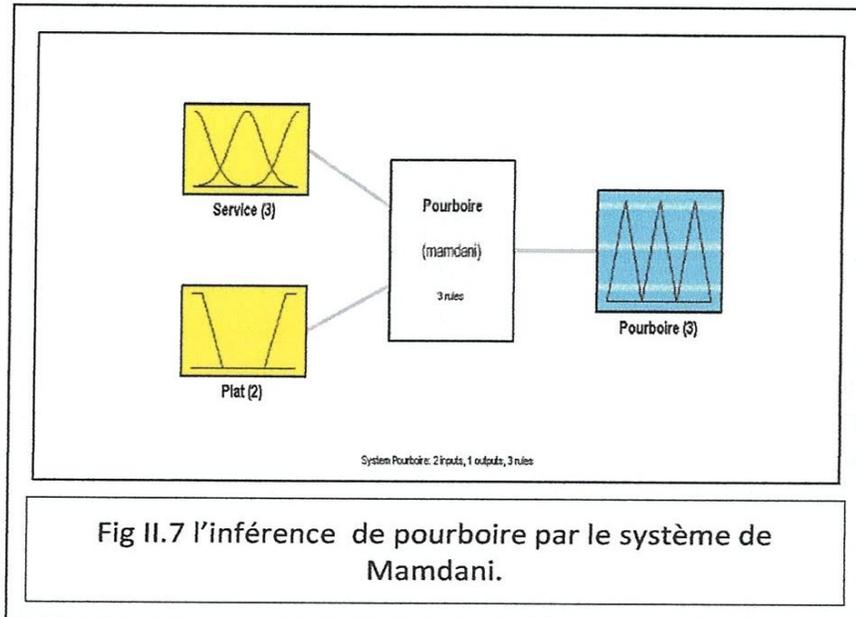
Le point de départ est d'écrire les trois règles d'or de pourboire:

*Si le service est mauvais OU le plat est mauvais,  
Alors le pourboire est peu.*

*Si le service est bon, Alors le pourboire est moyen.*

*Si le service est excellent OU le plat est délicieux, Alors le pourboire est généreux*

Supposant qu'un pourboire moyen est de 15%, généreux est de 25%, et peu est de 5%.



Pour la première entrée (le plat) elle fuzzifiée avec deux variables linguistiques, le premier est *mauvais* et le deuxième est *délicieux*, nous choisissons la fonction d'appartenance trapézoïdale pour les deux. Sa fonction d'appartenance est illustrée sur la figure (II.8)

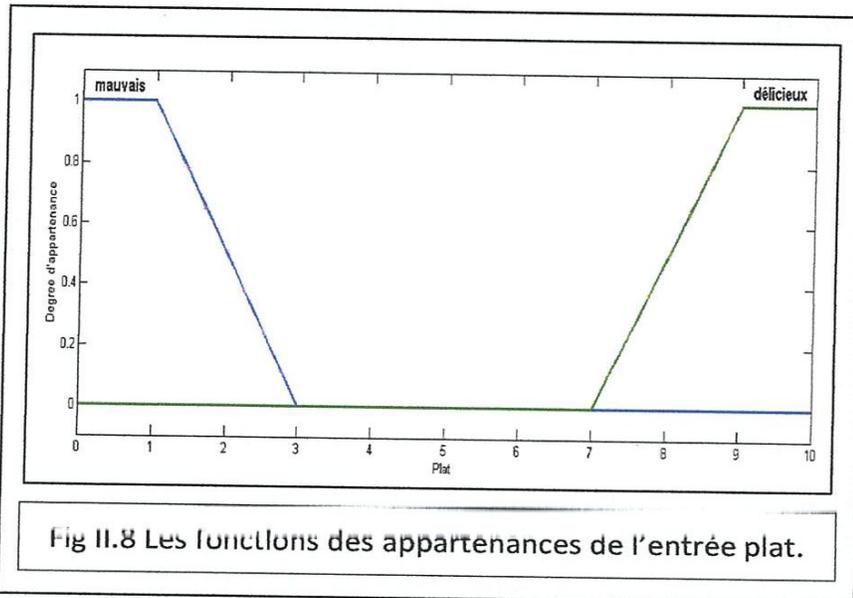


Fig II.8 Les fonctions des appartenances de l'entrée plat.

La deuxième entrée (le service) dispose de trois variables linguistiques le premier est *mauvais* et la seconde est *bon* et le troisième est *excellent*, les fonctions d'appartenance choisies pour la fuzzification de cette entrée sont des Gaussiennes. Sa fonction d'appartenance est illustrée sur la figure (II.9)

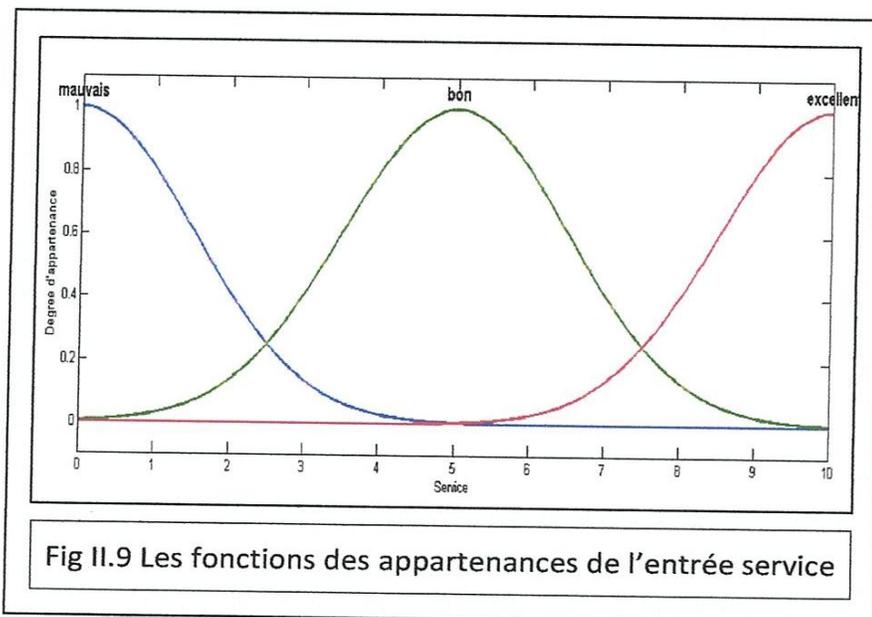
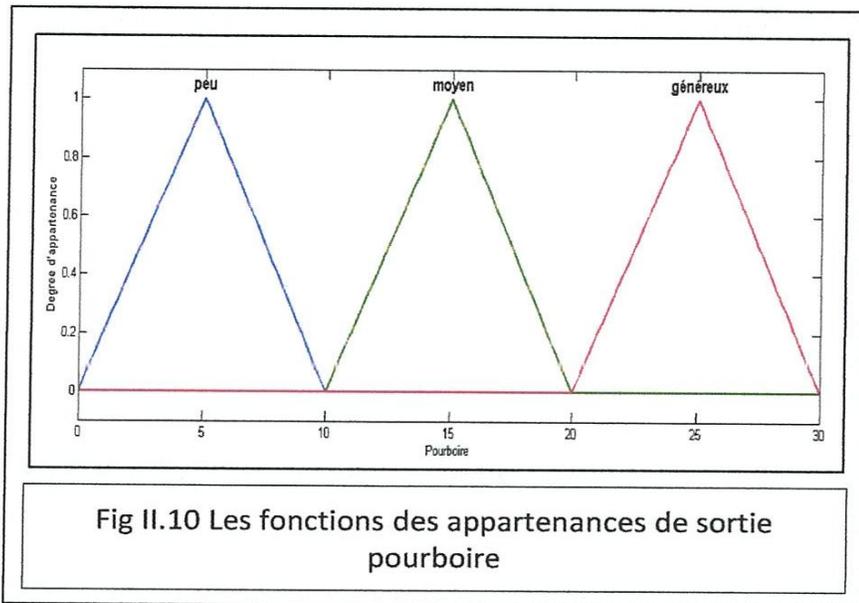


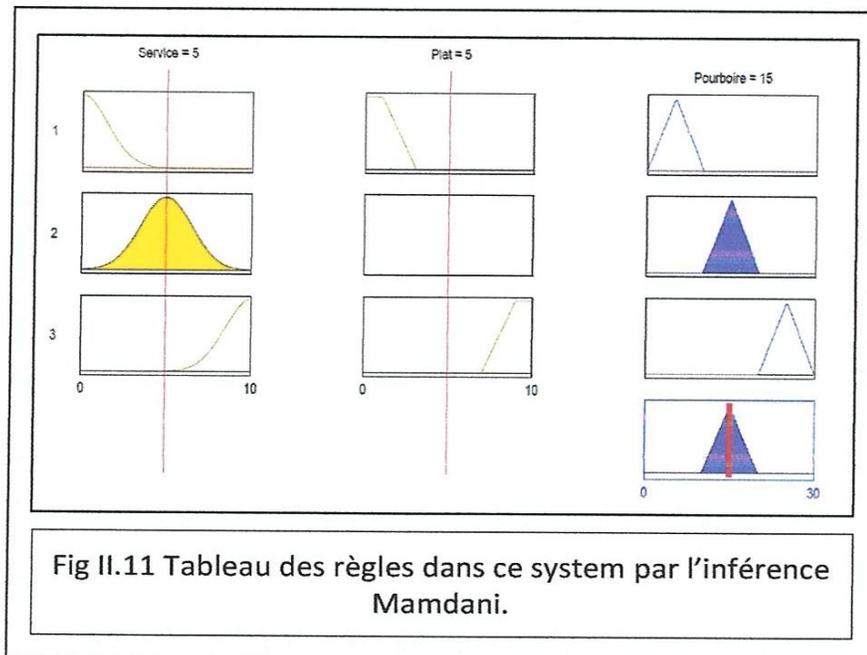
Fig II.9 Les fonctions des appartenances de l'entrée service

Nous utilisons le système de Mamdani c'est-à-dire que même la sortie est représentée par des fonctions d'appartenance. Dans cet exemple les trois variables linguistique de la sortie sont comme suit : le premier est *peu*, le deuxième est *moyen* et le dernier *généreux*.

Nous choisissons des fonctions d'appartenance triangulaire pour représenter cette sortie figure (II.10).



La figure (II.11) permet de visualiser les trois règles de cet exemple.

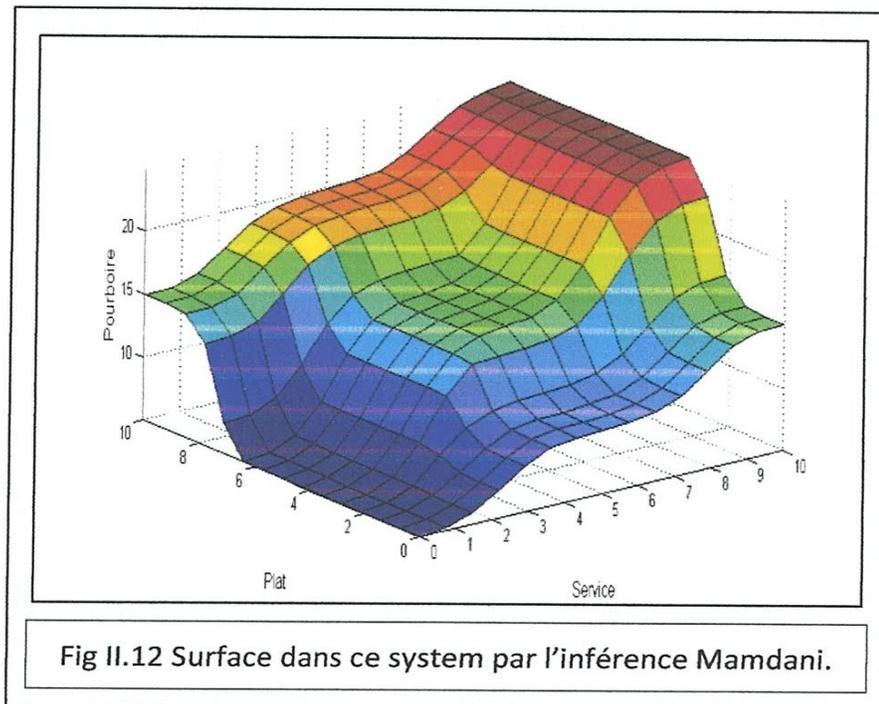


Les deux premières colonnes représentent les fonctions d'appartenance des prémisses de toutes les règles (les parties SI des règles).

La troisième colonne représente les fonctions d'appartenance des conséquences de toutes les règles (les parties ALORS des règles).

Le quatrième graphe de la troisième colonne représente la décision globale de ce système d'inférence. Dans l'exemple de cette figure pour l'entrée (service =5 et plat=5).

La figure (II.12) permet de visualiser la forme de la sortie (pourboire) en fonction des deux entrées (plat et service).



#### **II.4.2. Exemple de modélisation avec le système de Sugeno**

Pour présenter le système de Sugeno nous considérons également l'exemple précédent du pourboire « Basic Tipping Problem ».

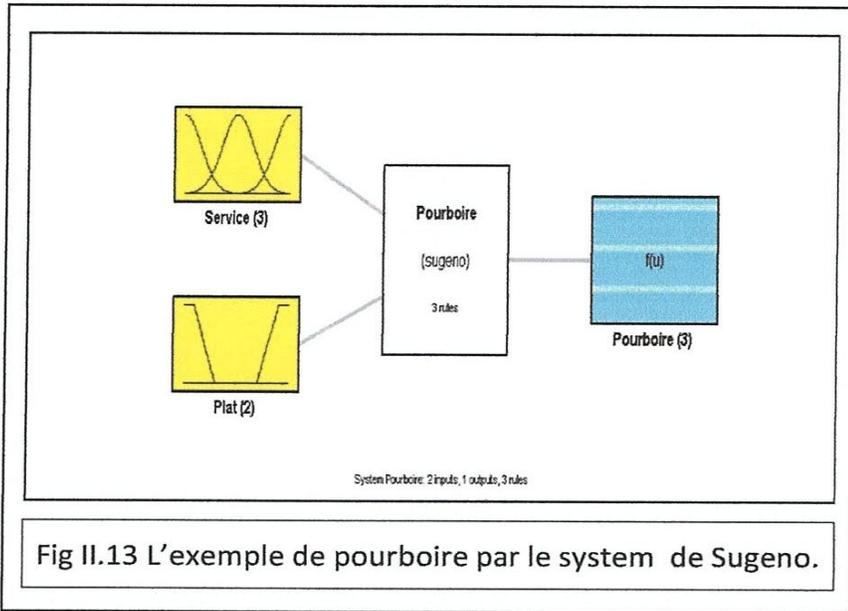


Fig II.13 L'exemple de pourboire par le system de Sugeno.

Nous utilisons les mêmes les fonctions d'appartenance pour la fuzzification des entrées. Cet exemple comporte aussi les mêmes règles. La différence réside dans la sortie, puisque dans le système de Sugeno les sorties sont des fonctions des entrées. Dans cet exemple nous utilisons un système d'ordre « 0 » où les sorties sont des constantes. La figure (II.14) permet de visualiser les trois règles de cet exemple.

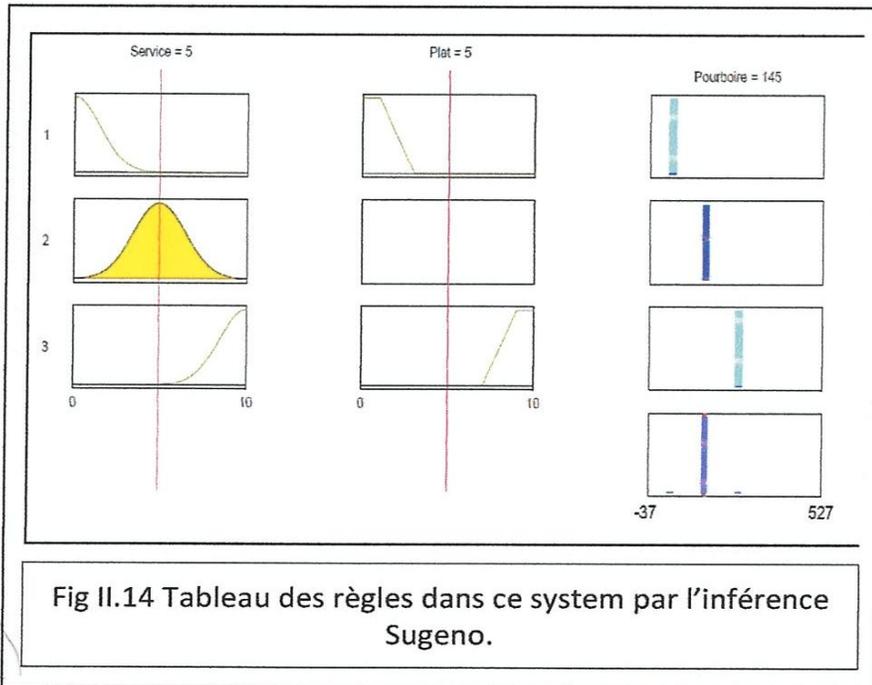
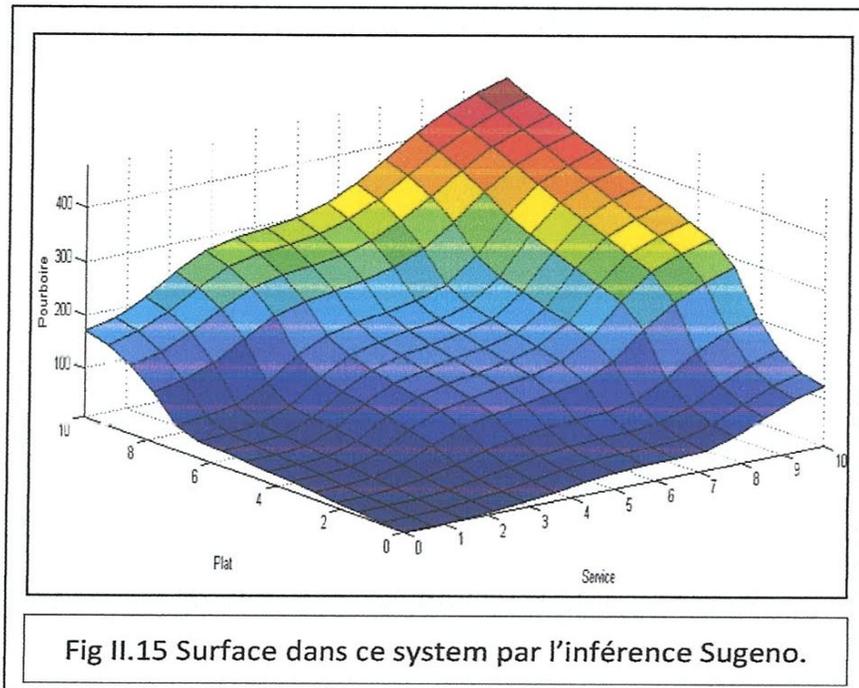


Fig II.14 Tableau des règles dans ce system par l'inférence Sugeno.

La figure (II.15) permet de visualiser la forme de la sortie (pourboire) en fonction des deux entrées (plat et service).



# Chapter III : Clustering

Ce chapitre décrit brièvement les méthodes de clustering. Il présente la méthode FCM et la méthode subtractive et un exemple en utilisant ces deux méthodes

### III.1. Introduction

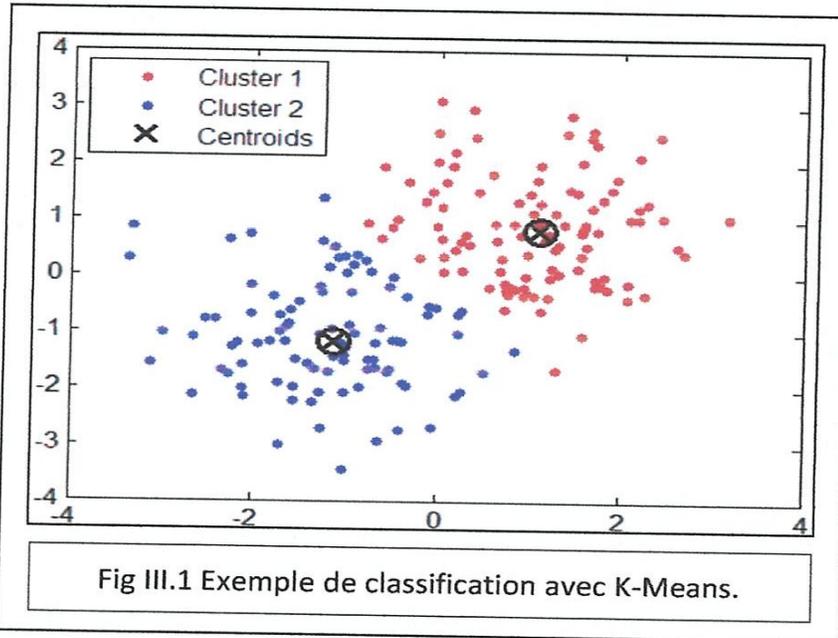
L'intérêt des méthodes de classification non supervisées (clustering) est qu'elles ne nécessitent aucune tâche préalable d'étiquetage. La seule intervention de l'expert se situe à la fin du processus pour évaluer les classes obtenues et les comparer avec la classe réelle. Les algorithmes non supervisés les plus répandus tendent à minimiser une fonction coût, dépendant de la distance de chaque point aux prototypes (ou noyaux) des classes. Le prototype d'une classe étant un point connu dont l'appartenance à la classe est garantie et où chaque point est assigné à la classe qui lui est la plus proche.

L'objectif général de la classification est donc de pouvoir étiqueter des données en leur associant une classe. L'apprentissage automatique se propose de construire automatiquement une telle procédure de classification en se basant sur des exemples, c'est-à-dire sur un ensemble limité de données disponibles. Si les classes possibles sont connues et si les exemples sont fournis avec l'étiquette de leur classe, on parle d'*apprentissage supervisé*. Au contraire, l'*apprentissage non supervisé* où seuls des exemples sans étiquette sont disponibles et où les classes sont inconnues. L'apprentissage se ramène alors à regrouper les exemples de la manière la plus naturelle possible. Cette volonté de regrouper naturellement est bien sûr ambiguë et le plus souvent formalisée par l'objectif de définir des groupes d'exemples tels que la distance entre exemples d'un même groupe soit minimale et que la distance entre groupes soit maximale (ces deux contraintes vont dans des sens opposés et c'est le meilleur compromis qui doit être trouvé).

### III.2. Clustering par la méthode K -Means

K-means est un des plus simples algorithmes de classification automatique des données. L'idée principale est de choisir aléatoirement un ensemble de centres fixé a priori et de chercher itérativement la partition optimale.

Chaque individu (également appelé centroïde ou centroid en anglais) est affecté au centre le plus proche, après l'affectation de toutes les données la moyenne de chaque groupe est calculé, elle constitue les nouveaux représentants des groupes, lorsqu'on abouti à un état stationnaire (aucune donnée ne change de groupe) l'algorithme est arrêté [7].



### III.2.1. L'algorithme de k-means

Le nom de cet algorithme dérive du fait que pour représenter chacune des  $K$  classes  $C_k$ , on utilise la moyenne  $\pi_k$  de ses points appelée centroides (ou centre de masse).

Chacune des  $C$  composantes du vecteur  $\pi_k$  est calculée par:

$$\pi_{kj} = \frac{1}{|C_k|} \sum_{O_i \in C_k} P_{ij} \quad (3.1)$$

Dans le cas de données numériques, cela donne un sens géométrique et statistique à la méthode. L'inertie intra-classe constitue le critère à optimiser. Elle est définie comme la moyenne des carrés des distances des objets de la

classe au centre de gravité de celle-ci. On cherche ainsi à construire des classes compactes.

L'inertie intra-classe associée à la classe  $C_k$  s'écrit formellement :

$$I_k = \frac{1}{|C_k|} \sum_{O_i \in C_k} d^2(O_i, \pi_{kj}) \quad (3.2)$$

L'objectif est alors de minimiser la somme de l'inertie intra-classe sur l'ensemble des classes. L'algorithme procède en deux étapes : dans la première phase, on réassigne tous les objets au centroïde le plus proche, et dans la deuxième phase, on recalcule les centroïdes des classes qui ont été modifiées. Pour mesurer la proximité entre un centroïde et un objet, on calculera une distance entre ces deux vecteurs. On pourra utiliser, par exemple, la distance euclidienne, calculée de la manière suivante :

$$d(O_i, \pi_{kj}) = \sqrt{\sum_j^c (P_{ij} - \pi_{kj})^2} \quad (3.3)$$

### III.3. Classification floue FCM (Fuzzy C- Means)

La modélisation de l'imprécision s'effectue en considérant des frontières graduelles au lieu de frontières nettes entre les classes. L'incertitude s'exprime par le fait qu'un point possède aussi bien des attributs qui l'assignent à une classe qu'à une autre. La classification floue assigne donc, non pas à un point une étiquette relative à une classe unique, mais son degré d'appartenance à chacune des classes. Ces valeurs expriment l'appartenance incertaine d'un point à un cluster sont appelées degrés d'appartenance. Le degré d'appartenance se situe dans l'intervalle  $[0, 1]$  et les classes obtenues ne sont pas forcément disjointes.

L'algorithme des c-moyennes floues (fuzzy c-means FCM) est une extension directe de l'algorithme classique des c-moyennes (c-means), où l'on a introduit la notion d'ensemble flou dans la définition des classes. Cet algorithme a été développé essentiellement par Bezdek [8].

Le principe de base est de former à partir des exemples non étiquetés c-groupes qui soient les plus homogènes et naturels possible. "Homogène" et "naturel" signifient que les groupes obtenus doivent contenir des individus les plus semblables possible, tandis que des individus de groupes différents doivent être les plus dissemblables possibles.

Ceci se traduit par le critère suivant à minimiser :

$$J_m = \sum_{j=0}^c \sum_{k=0}^l (\mu_{jk})^m d_j^2(x^k, v_j) \quad (3.4)$$

$v_j$  désigne le centroïde du cluster  $j$ , et  $x^k$  l'échantillon numéro  $k$ .

$d_j(x^k, v_j)$  est la distance entre l'échantillon  $x^k$  et le centroïde du cluster  $j$ , définie par

$$d_j^2(x^k, v_j) = (x^k - v_j)^T A_j (x^k - v_j) \quad (3.4)$$

Si  $A_j$  est la matrice identité, le cluster  $j$  sera sphérique, sinon il sera ellipsoïdal. Dans la version habituelle des FCM,  $A_j$  est la matrice identité.

$\mu_{jk}$  est le degré d'appartenance de l'échantillon à la classe (cluster). Ils forment une matrice, dite matrice d'appartenance.

$m$  est un index de *fuzziness*, qui règle le degré de *fuzziness* des clusters. Si  $m$  tend vers 1, on tend vers des clusters "nets", c'est-à-dire que la matrice  $M$  comportera uniquement des 0 et des 1. Par contre, plus  $m$  est grand, plus les

clusters sont "flous", c'est-à-dire que leur fonction d'appartenance est très étendue.

### III.3.1. L'algorithme du FCM

Etant donné un ensemble de vecteurs de données  $Z$  :

Fixer le nombre de *clusters*  $1 < c < N$

Fixer le degré de flou  $m > 1$  et l'erreur de fin d'algorithme  $\epsilon > 0$

Initialiser la matrice  $U = [\mu_{ik}]$  de partition floue

Répéter

Pas 1. Calculer les prototypes (centres) des clusters

$$v_i = \frac{\sum_{k=1}^N (\mu_{ik})^m \cdot Z_k}{\sum_{k=1}^N (\mu_{ik})^m} \quad (3.5)$$

Pas 2. Calculer les distances pour chaque cluster

$$D_{jkA}^2 = \|z_k - v_i\|_A^2 = (z_k - v_i)^T A (z_k - v_i) \quad (3.6)$$

Pas 3. Mettre à jour la matrice U de partition floue

$$\mu_{ik} = \frac{1}{\sum_{i=1}^c (D_{ikA}^2 / D_{jkA}^2)^{1/(m-1)}} \quad (3.7)$$

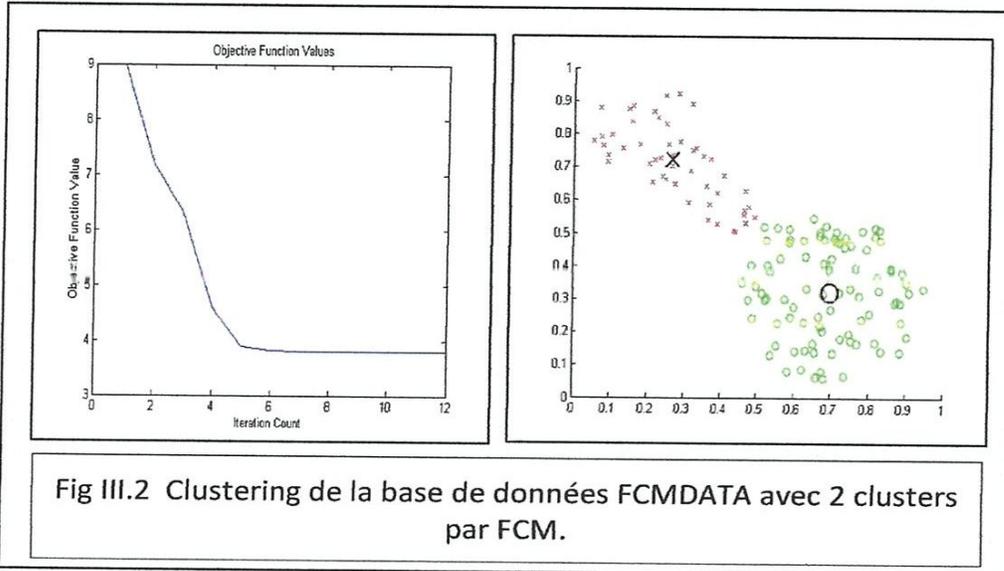
Jusqu'à obtenir la stabilité de la partition  $\|U^{(t)} - U^{(t-1)}\| < \epsilon$  [3]

#### a. Exemple :

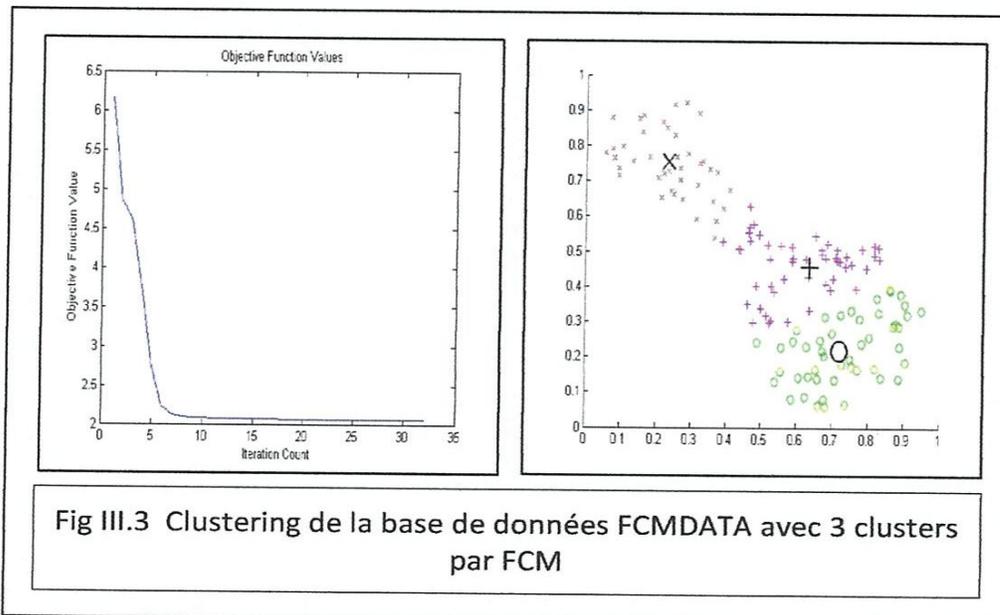
Nous allons appliquer la méthode FCM sur une base de données dans le MATLAB (FCMDATA). Nous effectuons plusieurs tests en utilisant de

déférentes valeurs de nombre de clusters. Les figures (2, 3) illustrent les résultats de clustering

Pour deux clusters :



Pour trois clusters



### III.4. La méthode de subtractive clustering :

La méthode de clustering " subtractive " permet, à la fois, de déterminer le nombre de cluster et leurs centres. Donc c'est contrairement à méthode FCM où on doit fixer a priori le nombre de clusters et les positions initiales de leurs centres. La méthode de clustering " subtractive " est une méthode itérative basée sur un calcul de potentiel des exemples à traiter. Plus un point a de voisins proches, plus son potentiel sera élevé.

Pour expliquer le principe de cette méthode, Considérons un ensemble de  $n$  exemples dans un espace  $M$  dimensionnel. Sans perdre le sens de généralité, on assume que les exemples sont normalisés dans chaque dimension. On considère chaque point de données  $x_i$  comme un centre de cluster et on définit une mesure de potentiel comme suit:

$$p_1 = \sum_{j=1}^n e^{-\alpha \|x_i - x_j\|^2} \quad (3.8)$$

Ou

$$\alpha = \frac{4}{r_a^2} \quad (3.9)$$

$r_a$  est une constante positive, donc la mesure de potentiel pour un point de données est une fonction de ses distances à tous les autres points. Un point de donnée avec beaucoup de point de donnée voisins aura une haute valeur de potentiel. Le constant  $r_a$  constitue le rayon qui définit le voisinage, les points de données on dehors de ce rayon ont une faible influence sur le potentiel. L'influence d'un point de données voisin se diminue exponentiellement avec le carré de la distance au lieu de la distance elle-même [9].

Après que le potentiel de chaque point de données a été calculé on sélectionne le point avec la plus haut potentiel comme le premier centre de

cluster. Soit  $x_1^*$  la position du premier centre de cluster et  $p_1^*$  son potentiel, puis le potentiel de chaque point de données  $x_i$  sera modifié par la formule :

$$p_i \Leftarrow p_i - p_1^* e^{-\beta \|x_i - x_1^*\|^2} \quad (3.10)$$

Ou

$$\beta = \frac{4}{r_b^2} \quad (3.11)$$

Ainsi, on soustrait une quantité de potentiel de chaque point de données en fonction de sa distance du centre du premier cluster. Alors on réduit le potentiel de chaque point de données selon sa distance au centre de cluster. Lorsque les potentiels de tous les points de données sont modifiés, on sélectionne le point de données avec le potentiel restant le plus élevé comme le deuxième centre de cluster et ainsi de suite jusqu'à l'obtention des centres de tous les clusters. Ce processus sera répéter selon les critères suivant :

$$\text{Si } p_k^* > \bar{\varepsilon} p_1^*$$

Accepte  $x_k^*$  comme centre de cluster et continue

$$\text{Sinon Si } p_k^* < \underline{\varepsilon} p_1^*$$

Rejeter  $x_k^*$  et terminer processus de clustering.

$$\text{Sinon Si } \frac{d_{min}}{r_a} + \frac{p_k^*}{p_1^*} \gg 1$$

Accepte  $x_k^*$  comme centre de cluster et continue.

Sinon

Rejeter  $x_k^*$  et définir le nouveau  $x_k^*$  (le plus haut potentiel) Ou :  $d_{min}$  = la plus courte des distances entre  $x_k^*$  et tous les centres de cluster précédents.

$\bar{\varepsilon}$  représente le seuil min, généralement =0.5

$\underline{\varepsilon}$  représente le seuil max, généralement =0.15

D'autre part,  $r_a$  est généralement choisie = 0.5, et  $r_b = 1.5 r_b$

*a. Exemple :*

Nous appliquons la méthode subtractive au même exemple précédent. Nous effectuons plusieurs tests en utilisant de différentes valeurs du rayon.

Les valeurs de rayon (0.6, 0.7, 0.8, 0.9) nous donner deux clusters, la valeur de rayon (0.5) nous donner trois clusters.

Pour  $r = 0.6$ , deux clusters

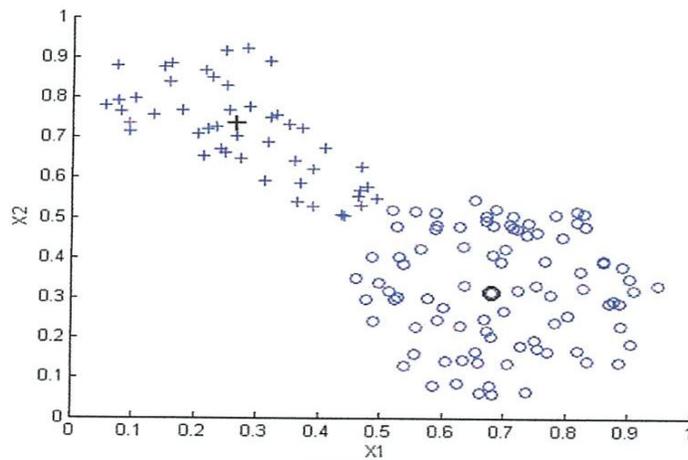
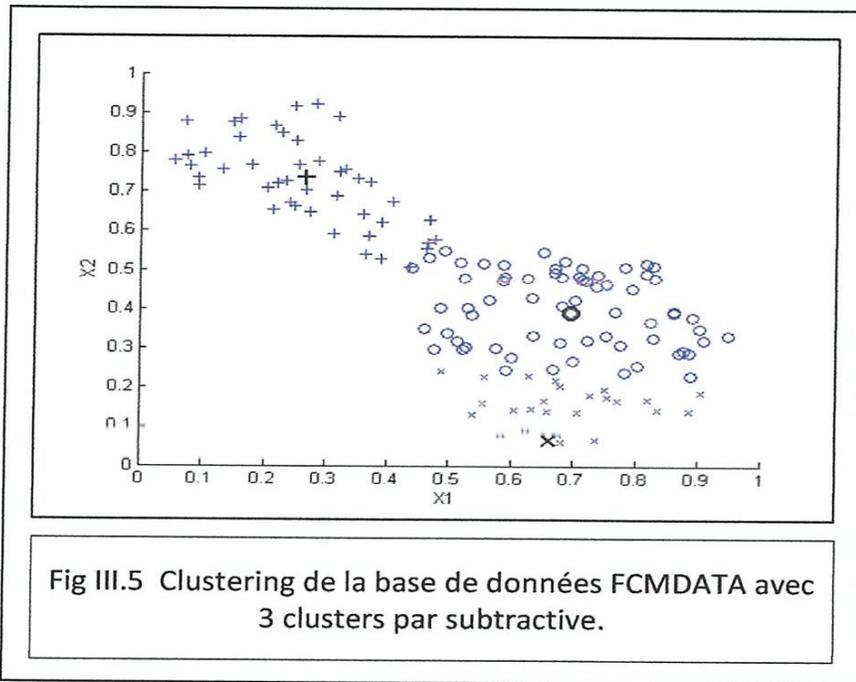


Fig III.4 Clustering de la base de données FCMDATA avec 2 clusters par subtractive

Pour  $r = 0.5$ , trois clusters



# Chapter IV : Application

Dans ce chapitre nous appliquons les systèmes de Mamdani et de Sugeno pour modélisation de la base de données Mytripdata et nous évaluons les performances de ces deux systèmes.

## IV.1. Modélisation floue basée sur le clustering

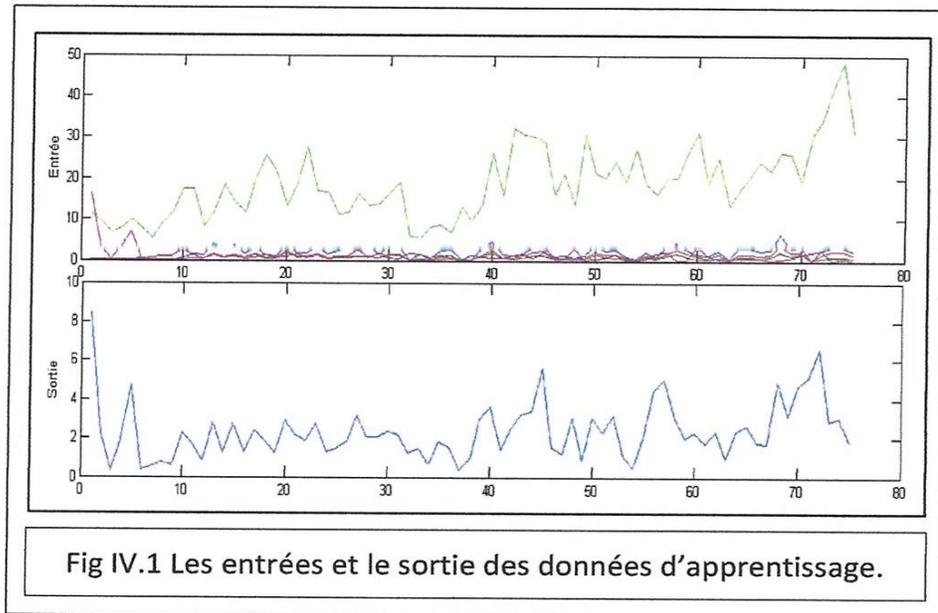
Dans un système d'inférence floue, les antécédents des règles floues partitionne l'espace des entrées en un certain nombre de régions locales, tandis que les conséquents décrivent le comportement dans une région donnée via une règle Si-Alors floue.

Le partitionnement de l'espace d'entrée s'effectue généralement selon la méthode de division en grilles « Grid partition ». Dans cette méthode, l'espace d'entrées est partitionné en un réseau de hyper-cubes. Dans ce cas, le nombre de règles nécessaires pour couvrir l'ensemble du domaine est une fonction exponentielle de la dimension de l'espace d'entrée et des ensembles flous utilisés pour chaque variable. Par exemple, un modèle flou avec 5 entrées et 3 fonctions d'appartenance sur chaque entrée se traduirait par  $3^5 = 243$  règles floues. Cette méthode de partitionnement est donc choisie seulement dans la conception des systèmes flous qui comportent quelques variables d'entrée et un petit nombre de fonctions d'appartenance. Ce problème, habituellement appelé la malédiction de la dimensionnalité, peut être atténué par les méthodes de clustering. Dans cette méthode chaque cluster est considéré comme une région représenté par une règle floue. Cette stratégie permet donc de limiter le nombre de règles utilisées.

## IV.2. Application sur la base de données *Mytripdata*

Cet exemple montre comment modéliser la relation entre le nombre de déplacement en automobile générés dans une région et les données démographiques de cette région. Ces données démographiques et de déplacement des véhicules ont été enregistrés dans 100 régions dans le comté de New Castle, Delaware. Cinq facteurs démographiques sont considérés: la population, le nombre de logements, nombre de propriétaire de véhicule, le revenu moyen par logement et le nombre d'employés. Par conséquent, le modèle a cinq variables d'entrée et une variable de sortie.

Cette base de données comporte 100 données, pour évaluer les performances de généralisation du modèle établie, la base de données est divisée en deux parties : une partie de 75 exemples pour l'apprentissage et 25 pour le test. La figure (IV.1) montre les données d'apprentissage.



### IV.3. Modélisation basé sur Mamdani

Dans notre base de donnée on a cinq entrée chaque entrée contient trois fonctions d'appartenance, nous choisissons la fonction d'appartenance de Gaussien pour présenter cette entrée. Les fonctions d'appartenances obtenues après le clustering sont illustrées sur la figure (IV.2).

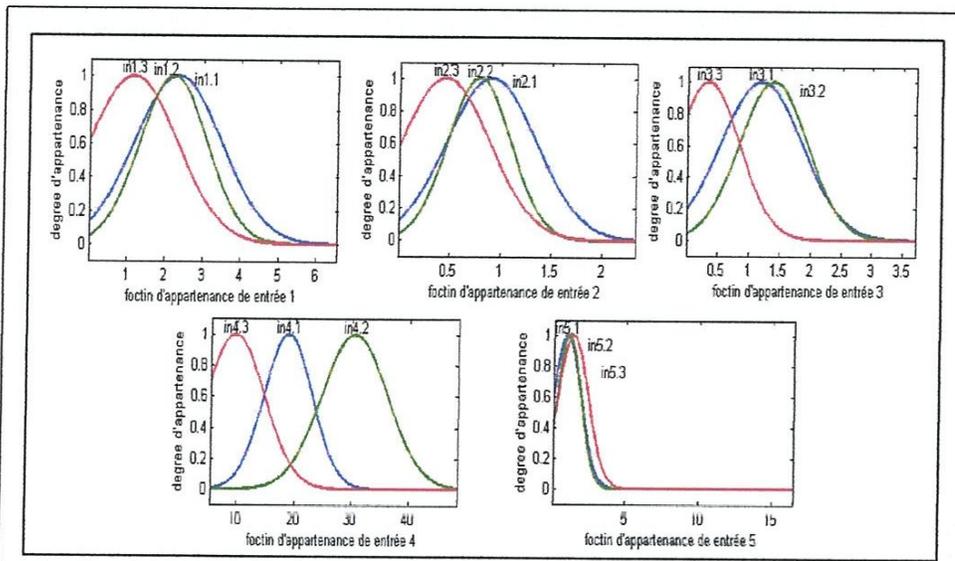


Fig IV.2 Les fonctions des appartenances des entrées des données d'apprentissage.

Cette base de données comporte une seule sortie qui sera représentée par trois fonctions d'appartenance de type de Gaussien.

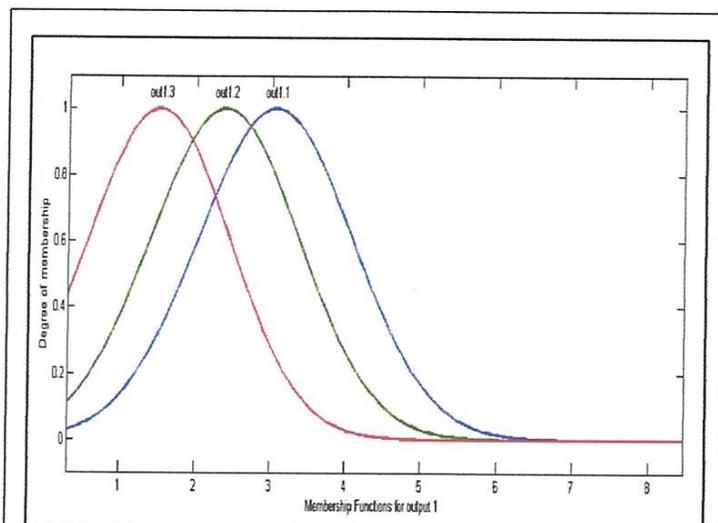


Fig IV.3 Les fonctions des appartenances la sortie des données d'apprentissage par système de Mamdani.

Du fait que le clustering a généré 3 sous-classes, donc cet exemple possède trois règles. Les règles de ce system sont illustrées sur la figure (IV.4).

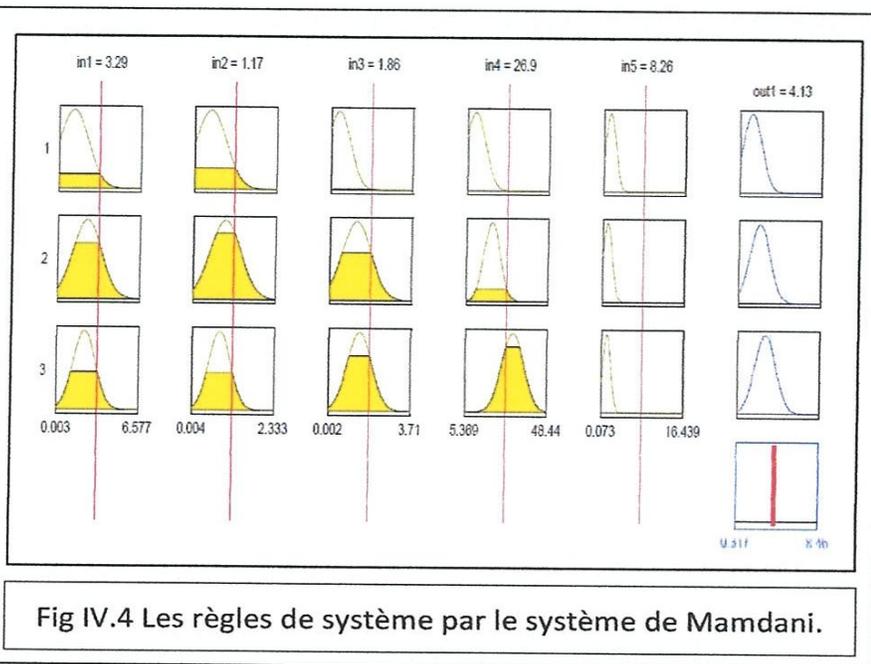


Fig IV.4 Les règles de système par le système de Mamdani.

La figure (IV.5) illustre la sortie calculée et la sortie désirée pour les données d'apprentissage pour un nombre de règles égale à 20, la sortie désirée et La sortie calculée sont présentées sous forme de cercles et ligne bleue, respectivement.

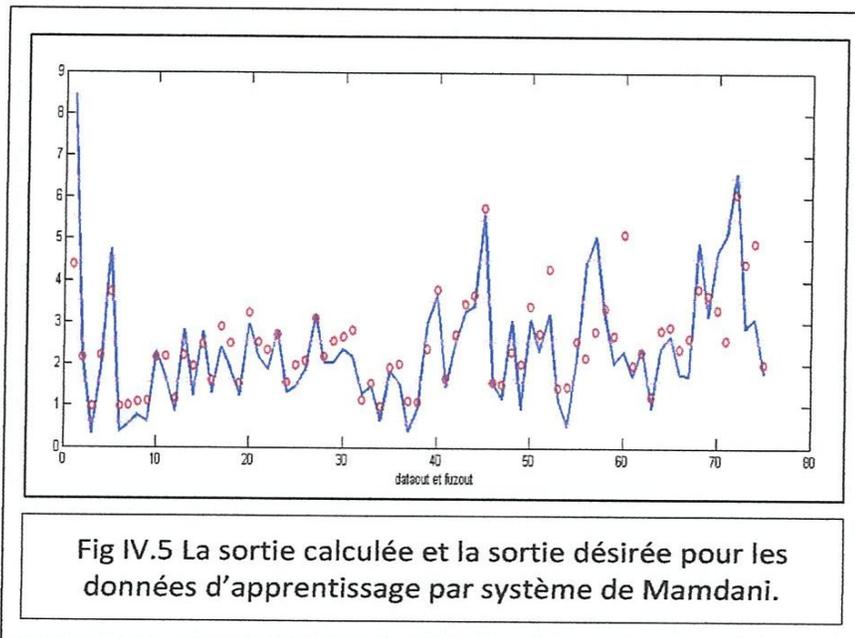


Fig IV.5 La sortie calculée et la sortie désirée pour les données d'apprentissage par système de Mamdani.

La figure (IV.6) illustre la sortie calculée et la sortie désirée pour les données de test pour un nombre de règles égale à 20, La sortie du modèle et les

données de test sont présentées sous forme de cercles et ligne bleue, respectivement.

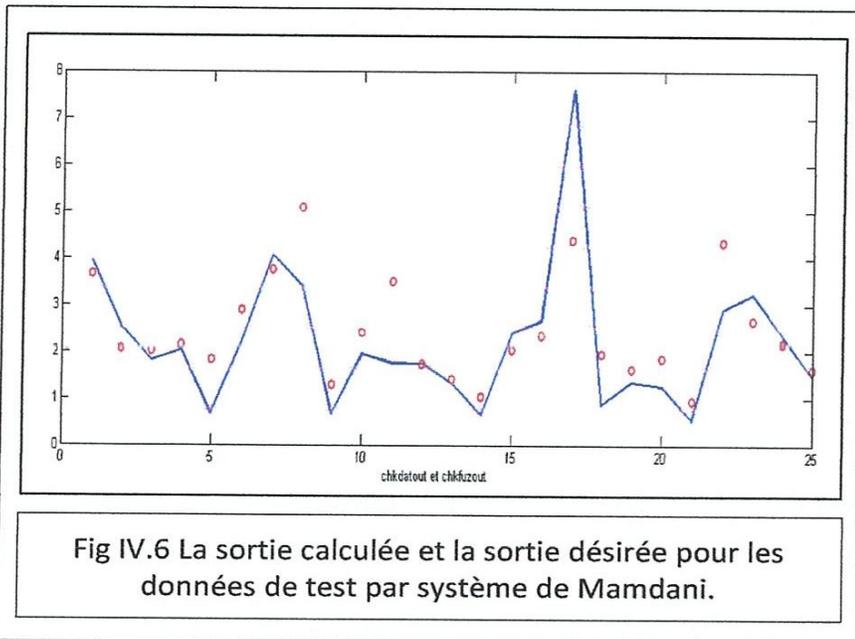


Tableau illustrée les erreurs d'apprentissage et les erreurs de test en fonction nombre des règles

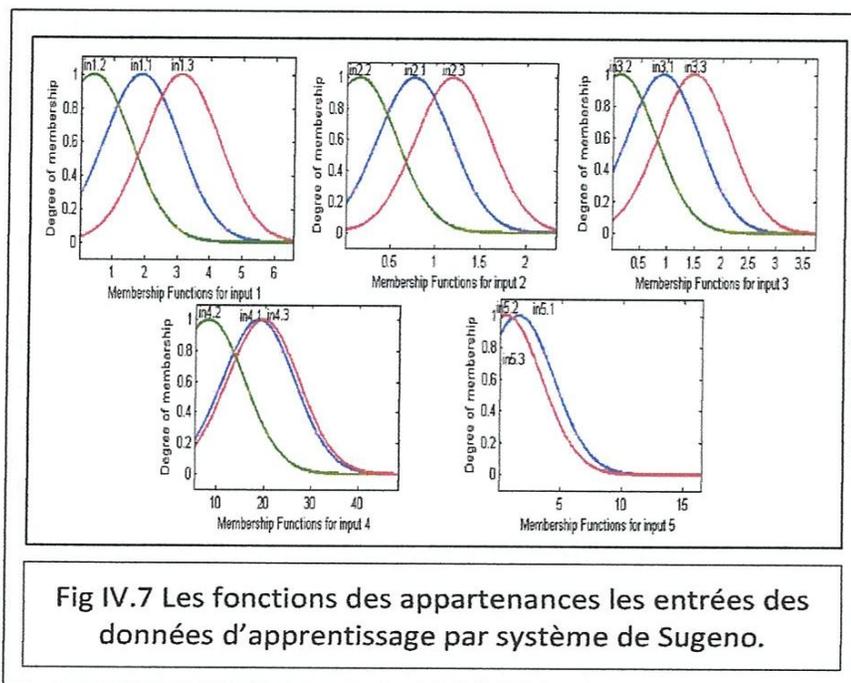
Les règles	Erreur d'apprentissage	Erreur de test
01	0	0
02	1.3655	1.4818
03	1.2178	1.3836
04	1.1729	1.3757
05	1.1628	1.3523
06	1.1695	1.3604
07	1.1554	1.2309
08	1.0563	1.2263
09	1.1382	1.2082

10	1.0573	1.2332
15	1.0226	1.0408
20	0.9890	0.9325
50	0.7377	1.1775

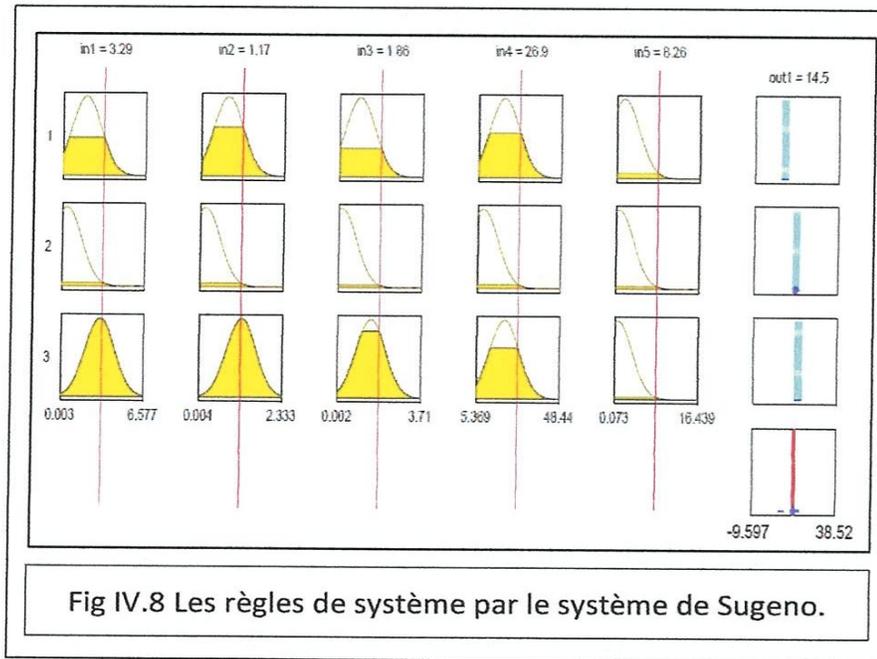
Tableau 4.1. Résultat de modélisation de système Mamdani

### IV.4. Modélisation basé sur Sugeno

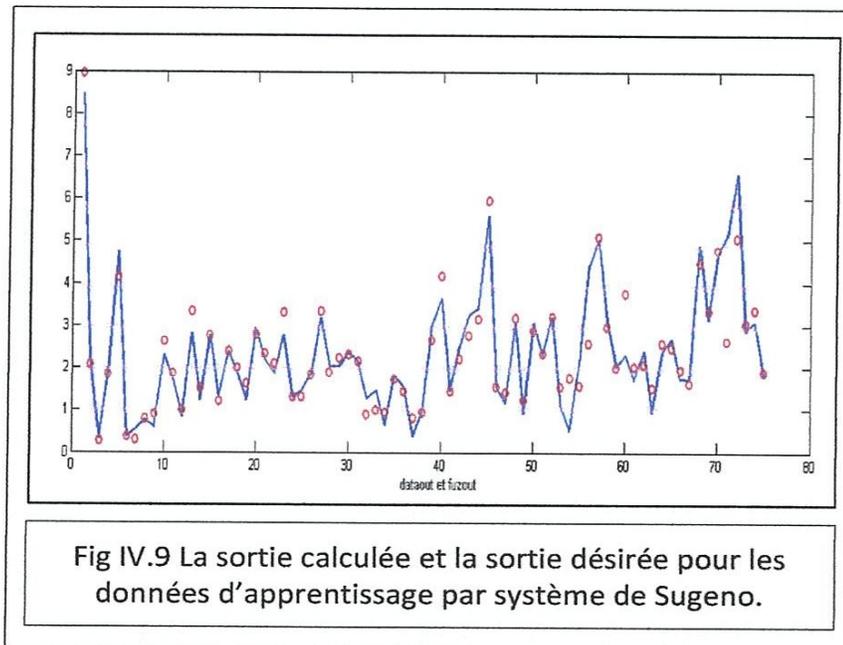
Dans notre base de donnée on a cinq entrée chaque entrée contienne trois fonctions d'appartenance, nous choisissons la fonction d'appartenance de Gaussien pour présenter cette entrée. Les fonctions d'appartenances obtenues après le clustering sont illustrées sur la figure (IV.7).



Du fait que le clustering a généré 3 sous-classes, donc cet exemple possède trois règles. Les règles de ce system sont illustrées sur la figure (IV.8).



La figure (IV.9) illustre la sortie calculée et la sortie désirée pour les données d'apprentissage pour un rayon 0.5, la sortie désirée et La sortie calculée sont présentées sous forme de cercles et ligne bleue, respectivement.



La sortie du modèle et le test des données sont présentées sous forme de cercles et ligne bleue, respectivement. Le graphique montre que le modèle ne

fonctionne pas bien sur les données de test. Après l'apprentissage terminé, valider le modèle.

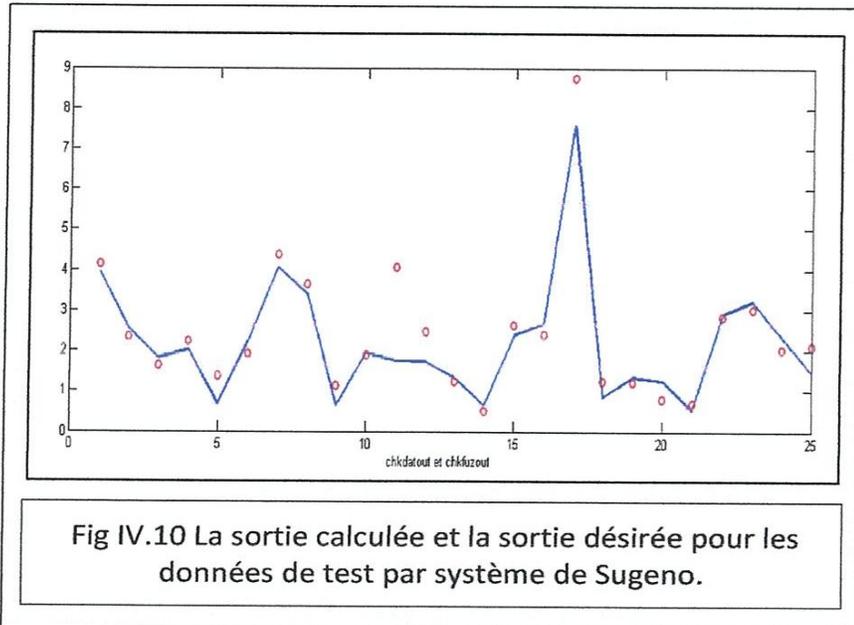


Tableau illustrée les erreurs d'apprentissage et les erreurs de test en fonction nombre des règles

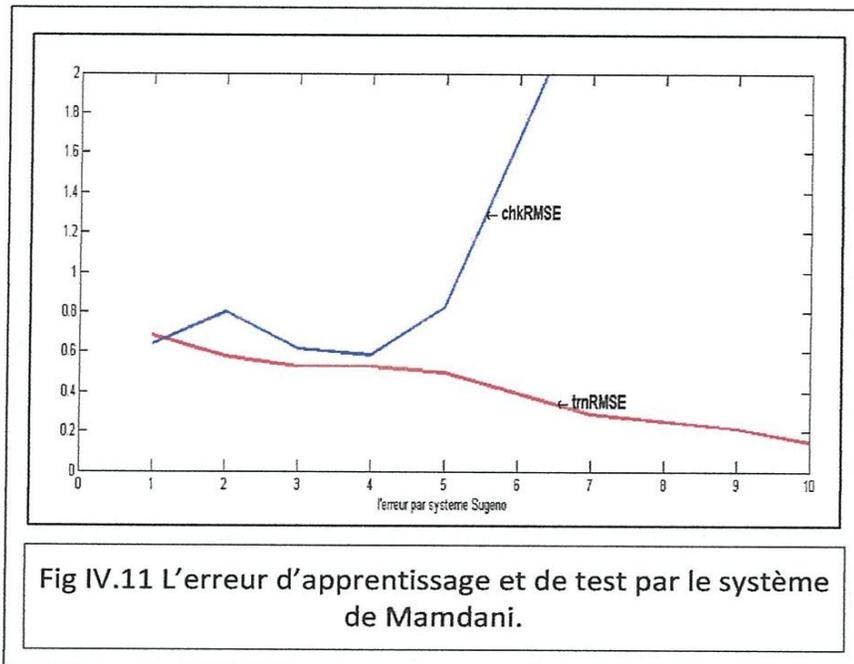
Les règles	Erreur d'apprentissage	Erreur de test
09	0.2179	5.2302
08	0.2788	2.5085
07	0.2895	2.5218
05	0.4998	0.8276
04	0.5276	0.5856
03	0.5276	0.6179
02	0.5796	0.8014
01	0.6858	0.6393

Tableau.4.2 résultat de modélisation de système Sugeno

La figure (IV.11) et la figure (IV.12) représentent les erreurs d'apprentissage et de test en fonction de nombre des règles pour le système de Mamdani et le système de Sugeno, respectivement.

Nous pouvons noter que les erreurs d'apprentissage et de test se diminuent lorsque le nombre des règles augmente, mais à partir d'un certain nombre des règles les d'apprentissage continue à diminuer et les erreurs de test commencent à augmenter. Ceci peut être définie comme le problème de surparamétrage (overfitting), où les performances de généralisation du système se diminuent.

Il est à noter également que le nombre des règles nécessaire pour le système de Sugeno sont moins de celui de Mamdani. En effet, le système de Sugeno nécessite seulement 3 règles tandis que de Mamdani nécessite 20.



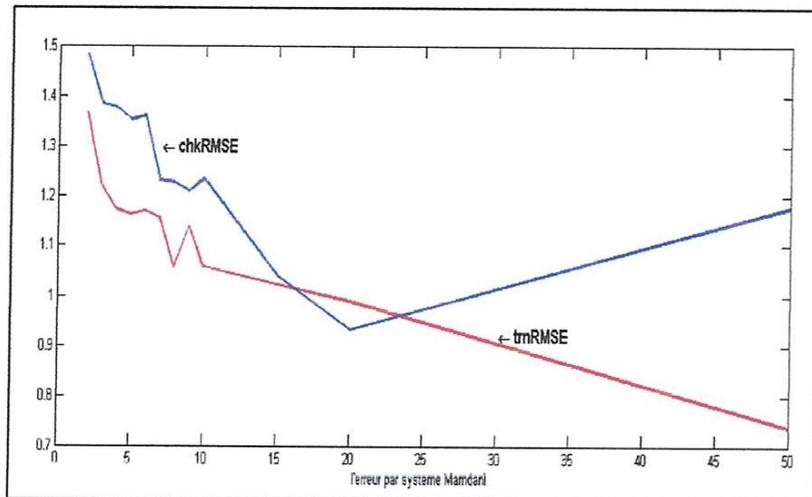


Fig IV.12 L'erreur d'apprentissage et de test par le système de Sugeno.

## Conclusion général

Dans ce travail on s'est intéressé à la modélisation floue en se basant sur les données entré-sortie. Nous avons présenté les concepts de base de la logique flous et les systèmes d'inférence flous ainsi que les méthodes de clustering. On s'est basé sur la méthode « fuzzy C-means » et la méthode « subtractive clustering ».

Dans notre étude on s'est concentré sur les deux systèmes d'Inférence floue les plus utilisés : le système de Mamdani et le système de Sugeno et nous avons s utilisée les méthodes de clustering pour la génération automatique des règles floues. Il s'agit, en premier lieu, de partitionner l'espace des entrées en régions formée par le clustering, puis, chaque région sera représentée par une règle floue.

Nous avons appliqué ces méthodes sur la base de données « Mytripdata » qui modélise la relation entre le nombre de déplacement en automobile générés dans une région et les données démographiques de cette région. Nous avons constaté que les deux systèmes, Mamdani et Sugeno, permettent de modéliser cette base de données. Nous avons noté que les erreurs d'apprentissage et de test se diminue lorsque le nombre des règles augmente, mais à partir d'un certain nombre des règles les d'apprentissage continue à diminuer et les erreurs de test commencent à augmenter. Ceci peut être définie comme le problème de sur paramétrage (overfitting), où les performances de généralisation du système se diminue. Nous avons noté également que le nombre des règles nécessaire pour le système de Sugeno sont moins de celui de Mamdani.

## Bibliographie

- [1] H. Buhler, Réglage par logique floue, Lausanne: Presses polytechnique romandes, 1994.
- [2] M. Halim, «Etude des systèmes flous à intervalle,» Université Mentouri de Constantine, Constantine, 2009.
- [3] S. Ambapour, «Théorie des ensembles flous : application à la mesure de la pauvreté au Congo,» Théorie des ensembles flous application à la mesure de la pauvreté au Congo., Congo, 2009.
- [4] P. Clara, «Introduction à la logique floue,» UTC University de Technologie Compiègne, France, 2011.
- [5] N. A. & C. Bekhaled, Application des systèmes hybrides neuro-flous au diagnostic des huiles de transformateurs, Algeria: Ecole nationale polytechnique, juin 2005.
- [6] MathWorks, Fuzzy Logic Toolbox™ User's Guide, New Work: The MathWorks, Inc., 2015.
- [7] C. G, Classification Automatique des Données, Paris: Bordas, 1989.
- [8] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, 1981.
- [9] L. Stephen, «Fuzzy Model Identification Based on Cluster Estimation,» *Rockwell Scionce Center*, pp. 268-278, 1994.
- [10] N. Ragot, Reconnaissance de formes par modélisation mixte intrinsèque/discriminante à base de systèmes d'inférence floue hiérarchisés, France: Université Rennes, 2003.