

Chapitre V

SYSTEME HYBRIDE -ANFIS-

Système Hybride –ANFIS-

V.1 Introduction

Le cerveau est un réseau de neurones, et la logique floue n'est autre qu'une technique que le cerveau utilise afin de traiter les données. Le contrôleur flou peut être vu comme un système expert fonctionnant à partir d'une représentation des connaissances basées sur la théorie des ensembles flous [58]. L'avantage d'un tel système est que seules les connaissances sur le comportement du procédé à commander sont suffisantes pour la synthèse des lois de commandes [12] [58], mais le contrôleur flou est incapable d'apprendre l'état du système contrairement aux réseaux de neurones, d'un autre point de vue, les réseaux de neurones ont besoin d'un minimum d'informations pour effectuer l'apprentissage ; d'où l'utilité de l'hybridation de ces deux approches. L'approche hybride est une orientation plus pragmatique ayant pour objectif la fusion de plusieurs techniques de traitement par une formulation théorique unifiée, à la recherche d'une intelligence de degré supérieur. Ainsi les réseaux neuronaux flous, conçus pour réunir les avantages des deux techniques de l'intelligence artificielle, se présentent comme des outils de traitement très puissants intégrant les capacités d'apprentissage et du raisonnement approximatif [59, 60]. Une étude fouillée des différents aspects théoriques et pratiques des FNNs et de leurs champs d'application, nous a permis d'adopter les caractéristiques suivantes [55] [61, 62] :

A. *Efficacité*

Dans les applications pratiques des FNNs, l'efficacité peut être évaluée en termes de performances obtenues, d'un côté, et des exigences (mémoire, temps de calcul), d'un autre.

B. *Coût d'implémentation*

Dans le cas d'une implémentation hardware, le coût est lié au nombre de portes logiques nécessaires. Dans le cas d'une implémentation software, ceci revient à développer des modèles n'impliquant que des opérations arithmétiques simples et n'exigeant pas d'organes de calculs supplémentaires (coprocesseur arithmétique).

C. Représentation mathématique simple et explicite

Ceci se résume à la facilité de développement d'algorithmes d'apprentissage pour l'architecture en question.

D. Modèle général souple

Un tel modèle peut être obtenu par une formulation générale unifiant les différentes possibilités (différents opérateurs logiques, méthodes de fuzzification/défuzzification, etc.). Un ajustement empirique de certains paramètres du modèle permet un choix optimal de la structure du réseau et une meilleure adaptation à l'application pratique.

V.2 Le système Hybride – ANFIS –

ANFIS est un système d'inférence adaptatif utilisant la commande neuronal floue. La combinaison de ces deux techniques appelée commande neuro-floue est fondamentalement une commande floue augmentée par les réseaux de neurone et permet d'améliorer les caractéristiques comme la flexibilité, la possibilité de traitement de données et l'adaptabilité [63]. Le processus du raisonnement flou est réalisé par les réseaux neurones dont les poids correspondent aux paramètres du raisonnement flou. Le contrôleur flou peut être vu comme un système expert fonctionnant à partir d'une représentation des connaissances basées sur la théorie des ensembles flous. L'avantage d'un tel système est que seules les connaissances sur le comportement du procédé à commander sont suffisantes pour la synthèse des lois de commandes. Mais le contrôleur flou est incapable d'apprendre l'état du système contrairement aux réseaux de neurones, d'un autre point de vue, les réseaux de neurones ont besoin d'un minimum d'informations pour effectuer l'apprentissage, d'où l'utilité de l'hybridation de ces deux approches [11] [63].

V.3 Le neurone flou

A l'origine, notre travail a été motivé par les problèmes rencontrés dans les réseaux de neurones. Il s'agit essentiellement de : (1) problèmes de convergence et (2) le choix empirique de certains paramètres, tels que la structure du réseau.

C'est ainsi que nous avons envisagé les deux solutions suivantes :

1. Initialisation des poids synaptiques du réseau neuronal en utilisant les connaissances floues disponibles. Ceci réduit la durée d'apprentissage et la probabilité de tomber dans un minimum local.
2. Estimation de la structure du réseau par la partition floue des espaces d'entrée/sortie.

Il convient alors d'implémenter chaque élément dans un système flou par un élément correspondant dans un réseau neuronal [50]. La figure (V.1) représente la structure du neurone flou.

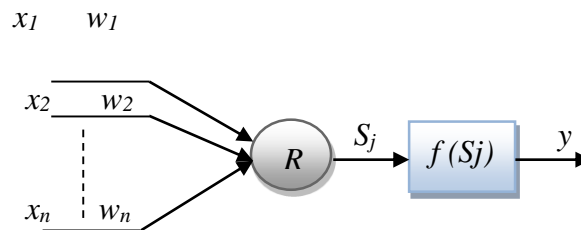


Fig V.1 : Structure du neurone flou

L'évaluation de la sortie se fait typiquement par la somme pondérée des entrées, et le passage du résultat à travers une non linéarité.

$$S = \sum_{i=0}^n w_i \cdot x_i \quad (5.1)$$

$$y = f(S) = \mu_{Ai}(x) \quad (5.2)$$

Où $f(S)$ représente la fonction d'activation du neurone. Le neurone flou est caractérisé par sa fonction d'activation qui doit être convexe et bornée. Cependant la fonction d'activation $\mu_{Ai}(x)$ peut être en forme de Gaussienne, triangle ou trapézoïdal. La fonction utilisée dans notre réseau est la fonction Gaussienne, la seule restriction sur le choix de cette fonction concerne la facilité de sa dérivé.

Les fonctions généralisées de ces formes sont représentées comme suit :

Triangle :
$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

Trapézoïdale :
$$\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right)$$

Gaussienne :
$$\mu(x) = \exp\left(-\frac{(x-c)^2}{\sigma^2}\right)$$

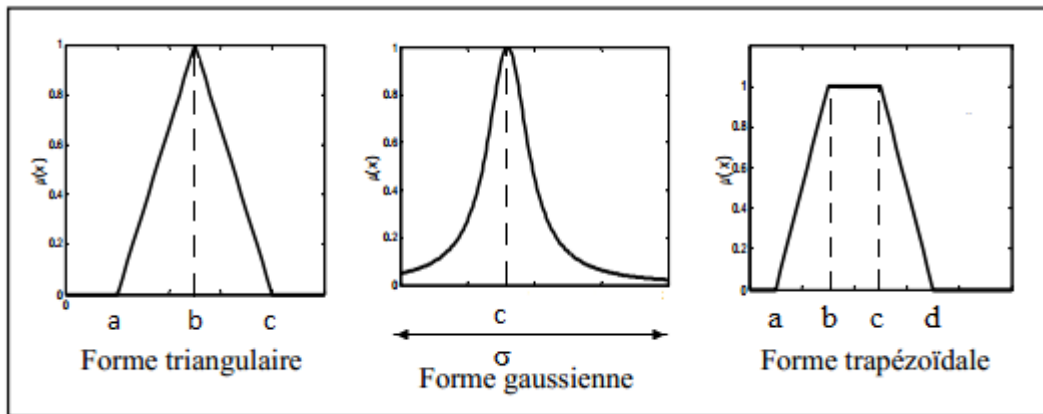


Fig V.2 : Les fonctions d'appartenance de l'ANFIS

Pendant que les valeurs des paramètres (a , b , c , σ) désignées sous le nom de paramètres de la fonction d'appartenance changent, les fonctions en forme précédente changent en conséquence [50] [59].

V.4 Extraction automatique des connaissances

Ces méthodes sont basées sur l'apprentissage pour déduire l'information suffisante pour traiter un problème quelconque dans un environnement complexe. Les deux approches les plus rencontrées dans la littérature sont l'approche à auto apprentissage ou directe et l'approche connexionniste.

V.4.1 Approche à auto – apprentissage ou directe

Les travaux de Mamdani et de Procyk ont conduit au développement du premier contrôleur SOC (*Self Organizing Controller*), il possède une structure hiérarchique constituée de deux bases de règles, la première est d'ordre général, la seconde est construite à partir de méta – règles qui crée et modifie la base de règle générale. Donc ce contrôleur expose la capacité humaine d'apprentissage et de modification. Les règles sont adaptées en cours de fonctionnement et peuvent donc suivre les modifications du processus à contrôler, cette stratégie est utilisée pour l'apprentissage des paramètres d'un FLC par application directe d'une méthode d'optimisation [36] [39].

V.4.2 Approche connexionniste

L'approche connexionniste consiste à combiner les deux méthodes, la théorie de la logique floue avec celle des réseaux de neurones [63]. Dans cette approche on construit un système d'inférence flou sous forme d'un réseau de neurones multicouches dans lequel les poids du réseau correspondent aux paramètres du système flou, dans notre cas les poids du réseau représentent les écarts types des fonctions d'appartenances. L'architecture du réseau dépend du type de règles et des méthodes d'inférence et de defuzzification choisies.

V.5 La commande neuronale floue

Les réseaux de neurones tentent de modéliser l'architecture du cerveau en créant une modélisation de l'entité du cerveau, les systèmes flous eux modélisent le cerveau par son mode de fonctionnement. Dans les deux cas le principe est de stocker la connaissance et l'employer pour prendre des décisions (lois) de commande, et la combinaison de ces deux techniques appelée commande neuro-floue est fondamentalement une commande floue augmentée par les réseaux de neurones et permet d'améliorer les caractéristiques comme la flexibilité, la possibilité de traitement de données, et l'adaptation aux changements des paramètres internes et externes du procédé à contrôler. Le processus du raisonnement flou est réalisé par les réseaux de neurones dont les poids correspondent aux paramètres du raisonnement flou. En utilisant la méthode de rétro-propagation ou n'importe quel

algorithmes d'apprentissage, le système de commande neuro-floue peut identifier les règles de commande et adapter les poids synaptiques du réseau, ainsi que les paramètres des fonctions d'appartenance floues. Il convient de noter que le modèle flou de type Takagi–Sugeno est l'un des systèmes les plus employés pour réaliser la commande neuro-floue, cela revient à sa fonction qui lie les entrées avec la sortie sous forme d'une équation polynomiale [13]. Un des avantages principaux de la commande neuro-floue est qu'elle n'exige pas des informations sur le modèle mathématique du système à commander. Cette classe de commande offre une nouvelle voie dans la résolution de plusieurs problèmes de commande où le modèle mathématique du système pourrait être non disponible. Cependant, une de ses limitations principales est l'analyse systématique de la stabilité des systèmes de commande en boucle fermée et la convergence des algorithmes d'adaptation [64].

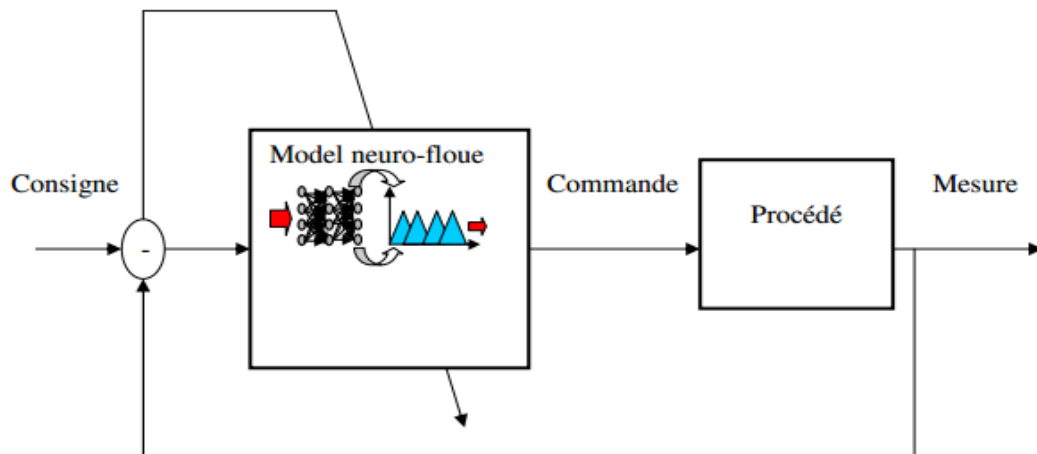


Fig V.3 : Structure générale d'une commande neuro-floue

V.6 Le Modèle ANFIS

De nombreux types de systèmes Neuro-flous ont été définis et développés ces dernières années, loin d'être uniformisés. Il y a plusieurs types pour combiner les réseaux de neurones et les systèmes flous tels que le type FALCON (Fuzzy Adaptive Learning Control Network), NEFCON (NEuro-Fuzzy CONtrol), ANFIS (Adaptif Fuzzy Inference System), et beaucoup d'autres. Ces types peuvent être classés suivant leurs architectures et la configuration de recherche entre le système d'inférence flou et les réseaux de

neurones [59]. Dans le cas où la sortie dépend des entrées cela exige une combinaison de type d'ANFIS.

V.6.1 Architecture de l'ANFIS

ANFIS est un système d'inférence adaptatif neuro-flou qui consiste à utiliser un réseau neurone de type MLP à 5 couches pour lequel chaque couche correspond à la réalisation d'une étape d'un système d'inférence flou de type Takagi-Sugeno [39]. Pour la simplicité, nous supposons que le système d'inférence flou à deux entrées x et y , et à comme une sortie f . Supposons que la base de règle contient deux règles floues de type Takagi-Sugeno.

Règle 1 : SI x est A_1 et y est B_1 ALORS $f_1 = p_1 x + q_1 y + r_1$

Règle 2 : SI x est A_2 et y est B_2 ALORS $f_2 = p_2 x + q_2 y + r_2$

L'ANFIS à une architecture posée par cinq couches comme représenté sur la figure (V.4).

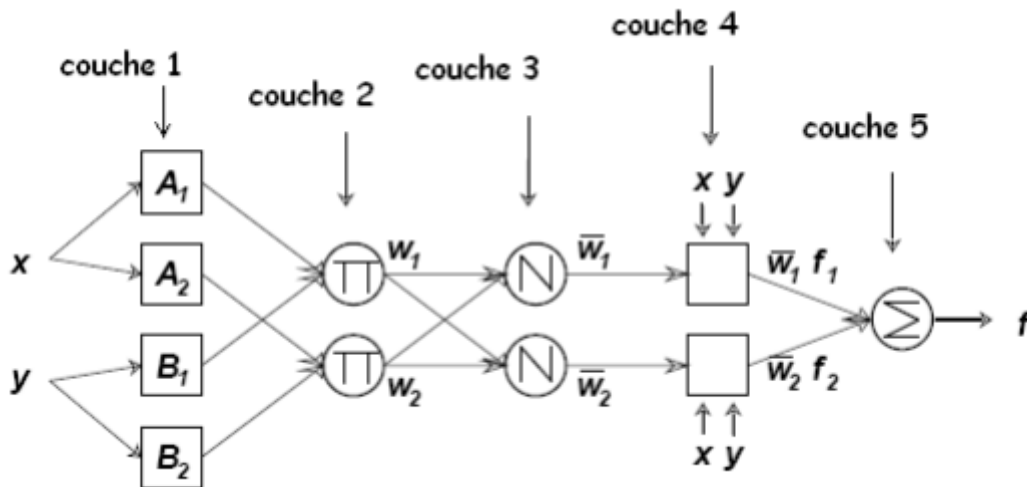


Fig V.4 : L'Architecture de l'ANFIS

Une architecture classique peut être décrite de la manière suivante :

1. La première couche d'une architecture de type ANFIS comporte un nombre de neurones qui correspond au nombre de sous-ensembles flous dans le système d'inférence représenté. Chaque neurone calcule le degré de vérité d'un sous ensemble flou particulier par sa fonction de d'appartenance qui est elle-même la fonction d'activation du neurone.

La fonction d'activation des neurones i de la première couche est :

$$f_i^l = \mu_{A_i}(x) \quad (5.3)$$

Tel que x est l'entrée au neurone i , et A_i est un sous ensemble flou correspondant à la variable x . En d'autres termes, f_i^l est la fonction d'appartenance du A_i et il indique le degré auquel la donnée x satisfait le quantifier A_i .

2. La deuxième couche cachée sert à calculer le degré d'activation des prémisses où chaque neurone représente la prémisse d'une règle. Ils reçoivent en entrée le degré de vérité des différents sous-ensembles flous composant cette prémisse. Les fonctions d'activation utilisées pour ces neurones peuvent être exprimées par l'implication de Mamdani ou celle de Larsen. La sortie de chaque neurone est déterminée par la fonction suivante

$$W_k = \mu_{A_i}(x) \times \mu_{B_j}(y) \quad (5.4)$$

Où k représente le nombre de règle, i représente le nombre de partition de x , et j le nombre de partition de y .

3. La troisième couche cachée normalise le degré d'activation des règles, où le i^{eme} neurone de cette couche calcule le rapport entre le i^{eme} poids de la couche précédente et la somme de tous les poids de cette couche. Cette opération est appelée *normalisation des poids*.

$$\overline{W}_k = \frac{W_k}{\sum W_i} \quad (5.5)$$

L'ensemble des sorties de cette couche seront appelées les poids normalisé.

4. La quatrième couche cachée sert à déterminer les paramètres de la partie conséquence des règles (p_k, q_k, r_k). La fonction de chaque neurone dans cette couche est la suivante :

$$f_k^4 = \overline{W}_k \times f_k = \overline{W}_k \times (p_k x + q_k y + r_k) \quad (5.6)$$

L'ensemble des paramètres (p_i, q_i, r_i) désigner sous le nom paramètres conséquents représentent les paramètres de la fonction de Takagi , Sugeno [37] .

5. La couche de sortie contient un seul neurone qui calcul la sortie globale comme addition de tous les signaux entrants, c'est-à-dire :

$$f^5 = \sum \overline{W}_k \times f_k \quad (5.7)$$

La méthode de fuzzification adaptée à cette architecture est celle des moyennes pondérées, et la procédure d'apprentissage utilisée peut être celle utilisée dans un réseau RBF (*Radial Basis Function*) traditionnel [65].

V.6.2 Algorithme d'apprentissage de l'ANFIS

Le système ANFIS applique le mécanisme d'apprentissage des réseaux de neurones sur les techniques d'inférence floues. D'un autre terme L'ANFIS est un système d'inférence floue dont les paramètres des fonctions d'appartenances sont ajustés en utilisant l'algorithme d'apprentissage de rétro-propagation. Dans l'architecture ANFIS proposée dans la figure (V.4), la sortie globale peut être exprimée en tant que des combinaisons linéaires des paramètres conséquents [60] [11]. Avec plus de précision, elle peut être réécrite comme suit :

$$f = \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_1+w_2} f_2 \quad (5.8)$$

$$f = (\overline{w}_1 x) p_1 + (\overline{w}_1 y) q_1 + (\overline{w}_1) r_1 + (\overline{w}_2 x) p_2 + (\overline{w}_2 y) q_2 + (\overline{w}_2) r_2$$

La sortie est une fonction linéaire des paramètres conséquents (p, q, r). La mesure de l'erreur E_k pour la K^{ime} entrée de donnée est donnée par :

$$E_k = \sum_{i=1}^{N(L)} (d_i - x_{L,i})^2 \quad (5.9)$$

Mesure d'erreur globale E :

$$E = \sum_{k=1}^K E_k \quad (5.10)$$

$N(L)$: Nombre de neurones dans la couche L

d_i : I^{em} composant du vecteur de sortie désiré

$x_{L,i}$: I^{em} composant de vecteur de sortie réel (de l'ANFIS)

K : Le nombre d'exemple ou d'itérations pendant la phase d'apprentissage.

V.6.3 Les avantages de l'ANFIS

- Exploitation de la connaissance disponible, grâce à la base de règles.
- Réduction de la taille de la base de règles : il suffit d'avoir des règles générales, les détails seront fournis par le réseau de neurones.
- Réduction de la complexité de l'apprentissage : le réseau de neurones doit simplement apprendre un cas particulier du procédé, pas le problème complet vis-à-vis au changements internes et externes du procédé à contrôler.
- Efficacité immédiate dès le début de l'apprentissage.

V.7 Conclusion

Au début de ce chapitre, nous avons commencé par l'exposé de certaines caractéristiques définissant la performance d'un modèle neuronal flou. Ceci pour justifier l'importance de l'utilisation de cette commande proposée. En effet, le développement des systèmes hybrides réunit les capacités d'apprentissage des réseaux de neurones avec la théorie très puissante de la logique floue qui permet d'obtenir des conclusions et de générer des réponses à partir des informations vagues ambiguës, imprécises et incomplètes. Cet outil permet d'intégrer une connaissance partielle issue de l'expertise et une connaissance issue de données. La connaissance experte est utilisée avec les relations causales dans une première modélisation assez simple et grossière mais suffisante. La connaissance experte est ensuite exprimée sous formes de règles floues et de contraintes sur la fuzzyfication des entrées, alors que la phase d'apprentissage permet d'ajuster les paramètres non définis (paramètres conséquents) à l'aide de données. L'utilisation de connaissances expertes permet ainsi de résoudre en partie les problèmes d'exhaustivité des comportements représentés par un ensemble de données. En revanche, les capacités d'apprentissage de l'ANFIS permettent de combler le manque de précision issue de l'expertise. C'est donc un système gagnant où les deux sources de connaissance (règles et données) permettent de combler les lacunes de l'autre. Ce système d'inférence adaptatif est très performant et largement utilisé dans les réalisations pratiques comme il est confirmé dans le chapitre suivant.